



#### 概要

このリファレンスマニュアルは、アプリケーション開発者を対象としています。STM32F75xxx および STM32F74xxx マイクロコントローラメモリとペリフェラルを使用する方法について、詳しく説明しています。

STM32F75xxx および STM32F74xxx は、さまざまなメモリサイズ、パッケージ、およびペリフェラルを持つマイクロコントローラファミリ製品になります。

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

ARM<sup>®</sup> FPU コア搭載 Cortex<sup>®</sup>-M7 については、FPU 搭載 Cortex<sup>®</sup>-M7 テクニカルリファレンスマニュアルを参照してください。

#### 関連ドキュメント

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

- STM32F75xxx および STM32F74xxx データシート

## 目次

<b>1</b>	<b>このマニュアルにおける表記の規則</b> .....	<b>59</b>
1.1	レジスタに関する略記 .....	59
1.2	用語 .....	60
1.3	使用可能なペリフェラル .....	61
<b>2</b>	<b>システムおよびメモリの概要</b> .....	<b>62</b>
2.1	システムアーキテクチャ .....	62
2.1.1	マルチ AHB バスマトリックス .....	63
2.1.2	AHB-APB ブリッジ (APB) .....	64
2.1.3	CPU AXIM バス .....	64
2.1.4	ITCM バス .....	64
2.1.5	DTCM バス .....	64
2.1.6	CPU AHBS バス .....	64
2.1.7	AHB ペリフェラルバス .....	65
2.1.8	DMA メモリバス .....	65
2.1.9	DMA ペリフェラルバス .....	65
2.1.10	Ethernet DMA バス .....	65
2.1.11	USB OTG HS DMA バス .....	65
2.1.12	LCD-TFT コントローラ DMA バス .....	65
2.1.13	DMA2D バス .....	65
2.2	メモリ構成 .....	66
2.2.1	概要 .....	66
2.2.2	メモリマップとレジスタ境界アドレス .....	66
2.3	内蔵 SRAM .....	70
2.4	フラッシュメモリの概要 .....	70
2.5	ブート設定 .....	71
<b>3</b>	<b>内蔵フラッシュメモリ (フラッシュ)</b> .....	<b>73</b>
3.1	概要 .....	73
3.2	フラッシュの主な機能 .....	73
3.3	フラッシュの機能詳細 .....	74
3.3.1	フラッシュメモリの構成 .....	74
3.3.2	読み出しアクセスの遅延 .....	75

3.3.3	フラッシュのプログラムおよび消去操作	77
3.3.4	フラッシュ制御レジスタのアンロック	78
3.3.5	プログラム/消去の並列処理	78
3.3.6	フラッシュ消去シーケンス	79
3.3.7	フラッシュプログラミングシーケンス	79
3.3.8	フラッシュ 割り込み	80
3.4	フラッシュオプションバイト	81
3.4.1	オプションバイトの説明	81
3.4.2	オプションバイトのプログラミング	84
3.5	フラッシュメモリの保護	85
3.5.1	読み出し保護 (RDP)	85
3.5.2	書き込み保護	87
3.6	OTP (One-time programmable) バイト	88
3.7	フラッシュレジスタ	89
3.7.1	フラッシュアクセス制御レジスタ (FLASH_ACR)	89
3.7.2	フラッシュキーレジスタ (FLASH_KEYR)	90
3.7.3	フラッシュオプションキーレジスタ (FLASH_OPTKEYR)	90
3.7.4	フラッシュステータスレジスタ (FLASH_SR)	91
3.7.5	フラッシュ制御レジスタ (FLASH_CR)	92
3.7.6	フラッシュオプション制御レジスタ (FLASH_OPTCR)	93
3.7.7	フラッシュオプション制御レジスタ (FLASH_OPTCR1)	94
3.7.8	フラッシュインタフェースレジスタマップ	96
<b>4</b>	<b>電源コントローラ (PWR)</b>	<b>97</b>
4.1	電源	97
4.1.1	独立した A/D コンバータ用電源と基準電圧	98
4.1.2	独立した USB トランシーバの電源	99
4.1.3	バッテリーバックアップドメイン	100
4.1.4	電圧レギュレータ	102
4.2	電源供給スーパバイザ	105
4.2.1	パワーオンリセット (POR) / パワーダウンリセット (PDR)	105
4.2.2	ブラウンアウトリセット (BOR)	106
4.2.3	プログラム可能な電圧検出器 (PVD)	106
4.3	低電力モード	107
4.3.1	デバッグモード	111
4.3.2	RUN モード	111



4.3.3	低電力モード	111
4.3.4	SLEEP モード	113
4.3.5	STOP モード	114
4.3.6	STANDBY モード	118
4.3.7	STOP および STANDBY モードからデバイスをウェイクアップさせるための RTC 代替機能のプログラミング	119
4.4	電源制御レジスタ	122
4.4.1	PWR 電源制御レジスタ (PWR_CR1)	122
4.4.2	PWR 電源制御/ステータスレジスタ (PWR_CSR1)	125
4.4.3	PWR 電源制御/ステータスレジスタ 2 (PWR_CR2)	126
4.4.4	PWR 電源制御レジスタ 2 (PWR_CSR2)	128
4.5	PWR レジスタマップ	130
<b>5</b>	<b>リセットおよびクロック制御 (RCC)</b>	<b>131</b>
5.1	リセット	131
5.1.1	システムリセット	131
5.1.2	電源リセット	131
5.1.3	バックアップドメインリセット	132
5.2	クロック	132
5.2.1	HSE クロック	135
5.2.2	HSI クロック	136
5.2.3	PLL	137
5.2.4	LSE クロック	138
5.2.5	LSI クロック	138
5.2.6	システムクロック (SYSCLK) の選択	138
5.2.7	クロックセキュリティシステム (CSS)	139
5.2.8	RTC/AWU クロック	139
5.2.9	ウォッチドッグクロック	140
5.2.10	クロック信号出力	140
5.2.11	TIM5/TIM11 を使用した内部/外部クロックの測定	141
5.2.12	ペリフェラルクロック有効レジスタ (RCC_AHBxENR、RCC_APBxENRy)	142
5.3	RCC レジスタ	143
5.3.1	RCC クロック制御レジスタ (RCC_CR)	143
5.3.2	RCC PLL 設定レジスタ (RCC_PLLCFGR)	145
5.3.3	RCC クロック設定レジスタ (RCC_CFGR)	147
5.3.4	RCC クロック割り込みレジスタ (RCC_CIR)	150

5.3.5	RCC AHB1 ペリフェラルリセットレジスタ (RCC_AHB1RSTR) .....	153
5.3.6	RCC AHB2 ペリフェラルリセットレジスタ (RCC_AHB2RSTR) .....	155
5.3.7	RCC AHB3 ペリフェラルリセットレジスタ (RCC_AHB3RSTR) .....	156
5.3.8	RCC APB1 ペリフェラルリセットレジスタ (RCC_APB1RSTR) .....	156
5.3.9	RCC APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR) .....	160
5.3.10	RCC AHB1 ペリフェラルクロックレジスタ (RCC_AHB1ENR) .....	162
5.3.11	RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2ENR) .....	164
5.3.12	RCC AHB3 ペリフェラルクロック有効レジスタ (RCC_AHB3ENR) .....	165
5.3.13	RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR) .....	166
5.3.14	RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2ENR) .....	169
5.3.15	低電力モードにおける RCC AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1LPENR) .....	171
5.3.16	低電力モードにおける RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2LPENR) .....	174
5.3.17	低電力モードにおける RCC AHB3 ペリフェラルクロック有効レジスタ (RCC_AHB3LPENR) .....	175
5.3.18	低電力モードにおける RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1LPENR) .....	176
5.3.19	低電力モードにおける RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2LPENR) .....	179
5.3.20	RCC バックアップドメイン制御レジスタ (RCC_BDCR) .....	181
5.3.21	RCC クロック制御およびステータスレジスタ (RCC_CSR) .....	183
5.3.22	RCC スペクトル拡散クロック生成レジスタ (RCC_SSCGR) .....	184
5.3.23	RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR) .....	185
5.3.24	RCC PLLSAI 設定レジスタ (RCC_PLLSAICFGR) .....	188
5.3.25	RCC 専用クロック設定レジスタ (RCC_DKCFGR1) .....	190
5.3.26	RCC 専用クロック設定レジスタ (DCKCFGR2) .....	192
5.3.27	RCC レジスタマップ .....	194
<b>6</b>	<b>汎用 I/O (GPIO) .....</b>	<b>197</b>
6.1	概要 .....	197
6.2	GPIO の主な機能 .....	197
6.3	GPIO の機能説明 .....	197
6.3.1	汎用 I/O (GPIO) .....	199
6.3.2	I/O ピンオルタネート機能マルチプレクサと配置 .....	200
6.3.3	I/O ポート制御レジスタ .....	201
6.3.4	I/O ポートデータレジスタ .....	201
6.3.5	I/O データのビット単位の操作 .....	201



6.3.6	GPIO ロック機構	202
6.3.7	I/O オルタネート機能の入力/出力	202
6.3.8	外部割り込み/ウェイクアップライン	202
6.3.9	入力設定	202
6.3.10	出力設定	203
6.3.11	オルタネート機能設定	204
6.3.12	アナログ設定	205
6.3.13	HSE または LSE オシレータのピンを GPIO として使用	205
6.3.14	GPIO ピンをバックアップ供給ドメインで使用	205
6.4	GPIO レジスタ	206
6.4.1	GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ K)	206
6.4.2	GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A ~ K)	207
6.4.3	GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A ~ K)	207
6.4.4	GPIO ポートプルアップ/プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ K)	208
6.4.5	GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ K)	208
6.4.6	GPIO ポート入力データレジスタ (GPIOx_ODR) (x = A ~ K)	209
6.4.7	GPIO ポートビットセット/クリアレジスタ (GPIOx_BSRR) (x = A ~ K)	209
6.4.8	GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ K)	210
6.4.9	GPIO オルタネート機能下位レジスタ (GPIOx_AFR1) (x = A ~ K)	211
6.4.10	GPIO オルタネート機能上位レジスタ (GPIOx_AFR2) (x = A ~ J)	211
6.4.11	GPIO レジスタマップ	212
<b>7</b>	<b>システム設定コントローラ (SYSCFG)</b>	<b>214</b>
7.1	I/O 補正セル	214
7.2	SYSCFG レジスタ	214
7.2.1	SYSCFG メモリ再配置レジスタ (SYSCFG_MEMRMP)	214
7.2.2	SYSCFG ペリフェラルモード設定レジスタ (SYSCFG_PMC)	215
7.2.3	SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1)	216
7.2.4	SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2)	217
7.2.5	SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG_EXTICR3)	218
7.2.6	SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4)	219
7.2.7	補正セル制御レジスタ (SYSCFG_CMPCR)	219
7.2.8	SYSCFG レジスタマップ	220

<b>8</b>	<b>ダイレクトメモリアクセスコントローラ (DMA)</b> .....	<b>221</b>
8.1	DMA の概要 .....	221
8.2	DMA の主な機能 .....	221
8.3	DMA の機能説明 .....	223
8.3.1	概要 .....	223
8.3.2	DMA トランザクション .....	224
8.3.3	チャンネル選択 .....	224
8.3.4	アービタ .....	225
8.3.5	DMA ストリーム .....	226
8.3.6	転送元、転送先、および転送モード .....	226
8.3.7	ポインタのインクリメント .....	229
8.3.8	サーキュラモード .....	230
8.3.9	ダブルバッファモード .....	230
8.3.10	プログラム可能なデータ幅、パッキング/アンパッキング、 エンディアン形式 .....	231
8.3.11	シングル転送とバースト転送 .....	233
8.3.12	FIFO .....	233
8.3.13	DMA 転送の完了 .....	236
8.3.14	DMA 転送の中断 .....	237
8.3.15	フローコントローラ .....	237
8.3.16	実現可能な DMA 設定の概要 .....	238
8.3.17	ストリーム設定手順 .....	239
8.3.18	エラー管理 .....	240
8.4	DMA 割り込み .....	241
8.5	DMA レジスタ .....	241
8.5.1	DMA ロー割り込みステータスレジスタ (DMA_LISR) .....	241
8.5.2	DMA ハイ割り込みステータスレジスタ (DMA_HISR) .....	242
8.5.3	DMA ロー割り込みフラグクリアレジスタ (DMA_LIFCR) .....	243
8.5.4	DMA ハイ割り込みフラグクリアレジスタ (DMA_HIFCR) .....	244
8.5.5	DMA ストリーム x 設定レジスタ (DMA_SxCR) (x=0 ~ 7) .....	245
8.5.6	DMA ストリーム x データ数レジスタ (DMA_HIFCR) (x=0 ~ 7) .....	248
8.5.7	DMA ストリーム x ペリフェラルアドレスレジスタ (DMA_SxPAR) (x=0 ~ 7) .....	249
8.5.8	DMA ストリーム x メモリ 0 アドレスレジスタ (DMA_SxM0AR) (x=0 ~ 7) .....	249
8.5.9	DMA ストリーム x メモリ 1 アドレスレジスタ (DMA_SxM1AR) (x=0 ~ 7) .....	250
8.5.10	DMA ストリーム x FIFO 制御レジスタ (DMA_SxFCR) (x=0 ~ 7) .....	250



8.5.11	DMA レジスタマップ	252
<b>9</b>	<b>Chrom-Art Accelerator™ コントローラ (DMA2D)</b>	<b>256</b>
9.1	DMA2D の概要	256
9.2	DMA2D の主な機能	257
9.3	DMA2D の機能詳細	257
9.3.1	概要	257
9.3.2	DMA2D 制御	258
9.3.3	DMA2D フォアグラウンドおよびバックグラウンド FIFO	258
9.3.4	DMA2D フォアグラウンドおよびバックグラウンドの ピクセルフォーマットコンバータ (PFC)	259
9.3.5	DMA2D フォアグラウンドおよびバックグラウンド CLUT インタフェース	261
9.3.6	DMA2D ブレンド	262
9.3.7	DMA2D 出力 PFC	263
9.3.8	DMA2D 出力 FIFO	263
9.3.9	DMA2D AHB マスタポートタイマ	264
9.3.10	DMA2D トランザクション	264
9.3.11	DMA2D 設定	264
9.3.12	DMA2D 転送制御 (開始、サスペンド、アポルト、完了)	267
9.3.13	ウォーターマーク	267
9.3.14	エラー管理	267
9.3.15	AHB デッドタイム	267
9.4	DMA2D 割り込み	268
9.5	DMA2D レジスタ	269
9.5.1	DMA2D 制御レジスタ (DMA2D_CR)	269
9.5.2	DMA2D 割り込みステータスレジスタ (DMA2D_ISR)	271
9.5.3	DMA2D 割り込みフラグクリアレジスタ (DMA2D_IFCR)	272
9.5.4	DMA2D フォアグラウンドメモリアドレスレジスタ (DMA2D_FGMAR)	273
9.5.5	DMA2D フォアグラウンドオフセットレジスタ (DMA2D_FGOR)	273
9.5.6	DMA2D バックグラウンドメモリアドレスレジスタ (DMA2D_BGMAR)	274
9.5.7	DMA2D バックグラウンドオフセットレジスタ (DMA2D_BGOR)	274
9.5.8	DMA2D フォアグラウンド PFC 制御レジスタ (DMA2D_FGPFCCR)	275
9.5.9	DMA2D フォアグラウンドカラーレジスタ (DMA2D_FGCOLR)	277
9.5.10	DMA2D バックグラウンド PFC 制御レジスタ (DMA2D_BGPFCCR)	278
9.5.11	DMA2D バックグラウンドカラーレジスタ (DMA2D_BGCOLR)	280
9.5.12	DMA2D フォアグラウンド CLUT メモリアドレスレジスタ (DMA2D_FGCMAR)	280

9.5.13	DMA2D バックグラウンド CLUT メモリアドレスレジスタ (DMA2D_BGCMAR) .....	281
9.5.14	DMA2D 出力 PFC 制御レジスタ (DMA2D_OPFCCR) .....	281
9.5.15	DMA2D 出力カラーレジスタ (DMA2D_OCOLR) .....	282
9.5.16	DMA2D 出力メモリアドレスレジスタ (DMA2D_OMAR) .....	283
9.5.17	DMA2D 出力オフセットレジスタ (DMA2D_OOR) .....	283
9.5.18	DMA2D ライン数レジスタ (DMA2D_NLR) .....	284
9.5.19	DMA2D ラインウォーターマークレジスタ (DMA2D_LWR) .....	284
9.5.20	DMA2D AHB マスタタイマ設定レジスタ (DMA2D_AMTGR) .....	285
9.5.21	DMA2D レジスタマップ .....	285
<b>10</b>	<b>ネスト化されたベクタ割り込みコントローラ (NVIC) .....</b>	<b>287</b>
10.1	NVIC の機能 .....	287
10.1.1	SysTick 較正值レジスタ .....	287
10.1.2	割り込みベクタと例外ベクタ .....	287
<b>11</b>	<b>拡張割り込み/イベントコントローラ (EXTI) .....</b>	<b>292</b>
11.1	EXTI の主な機能 .....	292
11.2	EXTI ブロック図 .....	292
11.3	ウェイクアップイベント管理 .....	293
11.4	機能詳細 .....	293
11.5	ハードウェア割り込みの選択 .....	293
11.6	ハードウェアイベントの選択 .....	293
11.7	ソフトウェア割り込み/イベントの選択 .....	294
11.8	外部割り込み/イベントラインの配置 .....	294
11.9	EXTI レジスタ .....	295
11.9.1	割り込みマスクレジスタ (EXTI_IMR) .....	295
11.9.2	イベントマスクレジスタ (EXTI_EMR) .....	296
11.9.3	立ち上がりトリガ選択レジスタ (EXTI_RTSR) .....	296
11.9.4	立ち下がりトリガ選択レジスタ (EXTI_FTSR) .....	297
11.9.5	ソフトウェア割り込みイベントレジスタ (EXTI_SWIER) .....	297
11.9.6	ペンディングレジスタ (EXTI_PR) .....	298
11.9.7	EXTI レジスタマップ .....	298



<b>12</b>	<b>巡回冗長検査計算ユニット (CRC)</b> .....	<b>299</b>
12.1	概要 .....	299
12.2	CRC の主な機能 .....	299
12.3	CRC の機能説明 .....	300
12.4	CRC レジスタ .....	301
12.4.1	データレジスタ (CRC_DR) .....	301
12.4.2	独立型データレジスタ (CRC_IDR) .....	302
12.4.3	制御レジスタ (CRC_CR) .....	302
12.4.4	CRC の初期値 (CRC_INIT) .....	303
12.4.5	CRC 多項式 (CRC_POL) .....	303
12.4.6	CRC レジスタマップ .....	304
<b>13</b>	<b>フレキシブル メモリコントローラ (FMC)</b> .....	<b>305</b>
13.1	FMC の主な機能 .....	305
13.2	ブロック図 .....	306
13.3	AHB インタフェース .....	307
13.3.1	サポートされるメモリおよびトランザクション .....	307
13.4	外部デバイスアドレスマッピング .....	308
13.4.1	NOR/PSRAM アドレスマッピング .....	310
13.4.2	NAND フラッシュメモリアドレスマッピング .....	310
13.4.3	SDRAM アドレスマッピング .....	311
13.5	NOR 型フラッシュ / PSRAM コントローラ .....	314
13.5.1	外部メモリインタフェース信号 .....	316
13.5.2	サポートされるメモリおよびトランザクション .....	318
13.5.3	一般的なタイミング規則 .....	319
13.5.4	NOR フラッシュ / PSRAM コントローラ非同期トランザクション .....	319
13.5.5	同期トランザクション .....	336
13.5.6	NOR/PSRAM コントローラレジスタ .....	343
13.6	NAND 型フラッシュコントローラ .....	349
13.6.1	外部メモリインタフェース信号 .....	350
13.6.2	NAND 型フラッシュによってサポートされるメモリとトランザクション .....	351
13.6.3	NAND フラッシュメモリのタイミング図 .....	351
13.6.4	NAND フラッシュ動作 .....	352
13.6.5	NAND フラッシュのプリウエイト機能 .....	353
13.6.6	エラー訂正コード (ECC) の計算 NAND 型フラッシュメモリ .....	354
13.6.7	NAND フラッシュコントローラレジスタ .....	355

13.7	SDRAM コントローラ	360
13.7.1	SDRAM コントローラの主な機能	360
13.7.2	SDRAM 外部メモリアンタフェース信号	360
13.7.3	SDRAM コントローラの機能説明	361
13.7.4	低電力モード	368
13.7.5	SDRAM コントローラレジスタ	371
13.8	FMC レジスタマップ	378
<b>14</b>	<b>Quad SPI インタフェース (QUADSPI)</b>	<b>380</b>
14.1	概要	380
14.2	QUADSPI の主な機能	380
14.3	QUADSPI の機能説明	380
14.3.1	QUADSPI ブロック図	380
14.3.2	QUADSPI コマンドシーケンス	382
14.3.3	QUADSPI シングルインタフェースプロトコルモード	384
14.3.4	QUADSPI インダイレクトモード	386
14.3.5	QUADSPI ステータスフラグポーリングモード	388
14.3.6	QUADSPI メモリマップドモード	388
14.3.7	QUADSPI フラッシュメモリの設定	389
14.3.8	QUADSPI 遅延データサンプリング	389
14.3.9	QUADSPI の設定	389
14.3.10	QUADSPI の使用	390
14.3.11	1 回限りの命令の送信	392
14.3.12	QUADSPI エラー管理	392
14.3.13	QUADSPI の BUSY ビットおよびアボート機能	393
14.3.14	nCS の動作	393
14.4	QUADSPI 割り込み	395
14.5	QUADSPI レジスタ	396
14.5.1	QUADSPI 制御レジスタ (QUADSPI_CR)	396
14.5.2	QUADSPI デバイス設定レジスタ (QUADSPI_DCR)	399
14.5.3	QUADSPI ステータスレジスタ (QUADSPI_SR)	400
14.5.4	QUADSPI フラグクリアレジスタ (QUADSPI_FCR)	401
14.5.5	QUADSPI データ長レジスタ (QUADSPI_DLR)	402
14.5.6	QUADSPI 通信設定レジスタ (QUADSPI_CCR)	403
14.5.7	QUADSPI アドレスレジスタ (QUADSPI_AR)	405
14.5.8	QUADSPI オルタネットバイトレジスタ (QUADSPI_ABR)	405

14.5.9	QUADSPI データレジスタ (QUADSPI_DR) .....	406
14.5.10	QUADSPI ポーリングステータスマスクレジスタ (QUADSPI_PSMKR) .....	407
14.5.11	QUADSPI ポーリングステータス一致レジスタ (QUADSPI_PSMAR) .....	407
14.5.12	QUADSPI ポーリングインターバルレジスタ (QUADSPI_PIR) .....	408
14.5.13	QUADSPI 低電力タイムアウトレジスタ (QUADSPI_LPTR) .....	408
14.5.14	QUADSPI レジスタマップ .....	409
<b>15</b>	<b>アナログデジタルコンバータ (ADC) .....</b>	<b>410</b>
15.1	ADC の概要 .....	410
15.2	ADC の主な機能 .....	410
15.3	ADC の機能詳細 .....	411
15.3.1	ADC のオン/オフ制御 .....	412
15.3.2	ADC1/2 および ADC3 接続性 .....	413
15.3.3	ADC クロック .....	416
15.3.4	チャンネル選択 .....	416
15.3.5	シングル変換モード .....	417
15.3.6	連続変換モード .....	417
15.3.7	タイミング図 .....	418
15.3.8	アナログウォッチドッグ .....	418
15.3.9	スキャンモード .....	419
15.3.10	インジェクトチャンネルの管理 .....	419
15.3.11	不連続モード .....	421
15.4	データの配置 .....	422
15.5	チャンネル単位でプログラム可能なサンプリング時間 .....	423
15.6	外部トリガによる変換およびトリガ極性 .....	423
15.7	高速変換モード .....	425
15.8	データ管理 .....	425
15.8.1	DMA の使用 .....	425
15.8.2	DMA を使用しない変換シーケンスの管理 .....	426
15.8.3	DMA およびオーバーラン検出を使用しない変換 .....	426
15.9	マルチ ADC モード .....	426
15.9.1	インジェクト同時モード .....	430
15.9.2	レギュラ同時モード .....	431
15.9.3	インタリーブモード .....	433
15.9.4	オルタネートトリガモード .....	434
15.9.5	レギュラ/インジェクト同時モードの組み合わせ .....	437

15.9.6	レギュラ同時 + オルタネートトリガモードの組み合わせ	437
15.10	温度センサ	438
15.11	バッテリー充電監視	440
15.12	ADC 割り込み	440
15.13	ADC レジスタ	441
15.13.1	ADC ステータスレジスタ (ADC_SR)	441
15.13.2	ADC 制御レジスタ 1 (ADC_CR1)	442
15.13.3	ADC 制御レジスタ 2 (ADC_CR2)	444
15.13.4	ADC サンプル時間レジスタ 1 (ADC_SMPR1)	447
15.13.5	ADC サンプル時間レジスタ 2 (ADC_SMPR2)	447
15.13.6	ADC インジェクトチャネルデータオフセットレジスタ x (ADC_JOFRx) (x=1..4)	448
15.13.7	ADC ウォッチドッグ高閾値レジスタ (ADC_HTR)	448
15.13.8	ADC ウォッチドッグ低閾値レジスタ (ADC_LTR)	449
15.13.9	ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)	449
15.13.10	ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)	450
15.13.11	ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)	450
15.13.12	ADC インジェクトシーケンスレジスタ (ADC_JSQR)	451
15.13.13	ADC インジェクトデータレジスタ x (ADC_JDRx) (x= 1..4)	452
15.13.14	ADC レギュラデータレジスタ (ADC_DR)	452
15.13.15	ADC 共通ステータスレジスタ (ADC_CSR)	453
15.13.16	ADC 共通制御レジスタ (ADC_CCR)	454
15.13.17	デュアルおよびトリプルモード用 ADC 共通レギュラデータレジスタ (ADC_CDR)	457
15.13.18	ADC レジスタマップ	457
<b>16</b>	<b>D/A コンバータ (DAC)</b>	<b>460</b>
16.1	DAC の概要	460
16.2	DAC の主な機能	460
16.3	DAC の機能詳細	462
16.3.1	DAC チャネルイネーブル	462
16.3.2	DAC 出力バッファイネーブル	462
16.3.3	DAC データフォーマット	462
16.3.4	DAC 変換	463
16.3.5	DAC 出力電圧	464
16.3.6	DAC トリガ選択	464
16.3.7	DMA リクエスト	465



16.3.8	ノイズ生成	465
16.3.9	三角波生成	466
16.4	デュアル DAC チャンネル変換	467
16.4.1	波形生成なしの独立トリガ	467
16.4.2	1 つの LFSR 生成による独立トリガ	468
16.4.3	異なる LFSR 生成による独立トリガ	468
16.4.4	1 つの三角波生成による独立トリガ	468
16.4.5	異なる三角波生成による独立トリガ	469
16.4.6	同時ソフトウェア開始	469
16.4.7	波形生成なしの同時トリガ	469
16.4.8	1 つの LFSR 生成による同時トリガ	470
16.4.9	異なる LFSR 生成による同時トリガ	470
16.4.10	1 つの三角波生成による同時トリガ	470
16.4.11	異なる三角波生成による同時トリガ	471
16.5	DAC レジスタ	472
16.5.1	DAC 制御レジスタ (DAC_CR)	472
16.5.2	DAC ソフトウェアトリガレジスタ (DAC_SWTRIGR)	475
16.5.3	DAC チャンネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R1)	476
16.5.4	DAC チャンネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L1)	476
16.5.5	DAC チャンネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R1)	477
16.5.6	DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2)	477
16.5.7	DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L2)	478
16.5.8	DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R2)	478
16.5.9	デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD)	479
16.5.10	デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD)	479
16.5.11	デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD)	480
16.5.12	DAC チャンネル 1 データ出力レジスタ (DAC_DOR1)	480
16.5.13	DAC チャンネル 2 データ出力レジスタ (DAC_DOR2)	481
16.5.14	DAC ステータスレジスタ (DAC_SR)	481
16.5.15	DAC レジスタマップ	482

<b>17</b>	<b>デジタルカメラインタフェース (DCMI) .....</b>	<b>483</b>
17.1	DCMI の概要 .....	483
17.2	DCMI の主な機能 .....	483
17.3	DCMI ピン .....	483
17.4	DCMI クロック .....	483
17.5	DCMI 機能概要 .....	484
17.5.1	DMA インタフェース .....	485
17.5.2	DCMI 物理インタフェース .....	485
17.5.3	同期 .....	487
17.5.4	キャプチャモード .....	490
17.5.5	クロップ機能 .....	491
17.5.6	JPEG フォーマット .....	492
17.5.7	FIFO .....	492
17.6	データフォーマットの説明 .....	493
17.6.1	データフォーマット .....	493
17.6.2	モノクロフォーマット .....	493
17.6.3	RGB フォーマット .....	494
17.6.4	YCbCr フォーマット .....	494
17.6.5	YCbCr フォーマット-Yのみ .....	495
17.6.6	ハーフ解像度画像抽出 .....	495
17.7	DCMI 割り込み .....	495
17.8	DCMI レジスタの説明 .....	496
17.8.1	DCMI 制御レジスタ (DCMI_CR) .....	496
17.8.2	DCMI ステータスレジスタ (DCMI_SR) .....	499
17.8.3	DCMI raw 割り込みステータスレジスタ (DCMI_RIS) .....	500
17.8.4	DCMI 割り込み有効レジスタ (DCMI_IER) .....	501
17.8.5	DCMI マスク済み割り込みステータスレジスタ (DCMI_MIS) .....	502
17.8.6	DCMI 割り込みクリアレジスタ (DCMI_ICR) .....	503
17.8.7	DCMI 埋め込み同期コードレジスタ (DCMI_ESCR) .....	504
17.8.8	DCMI 埋め込み同期マスク解除レジスタ (DCMI_ESUR) .....	505
17.8.9	DCMI クロップウィンドウ開始 (DCMI_CWSTRT) .....	506
17.8.10	DCMI クロップウィンドウサイズ (DCMI_CWSIZE) .....	506
17.8.11	DCMI データレジスタ (DCMI_DR) .....	507
17.8.12	DCMI レジスタマップ .....	508



<b>18</b>	<b>LCD-TFT コントローラ (LTDC) .....</b>	<b>509</b>
18.1	概要 .....	509
18.2	LTDC の主な機能 .....	509
18.3	LTDC の機能詳細 .....	510
18.3.1	LTDC のブロック図 .....	510
18.3.2	LTDC のリセットおよびクロック .....	510
18.3.3	LCD-TFT ピンと信号インタフェース .....	511
18.4	LTDC のプログラム可能なパラメータ .....	511
18.4.1	LTDC グローバル設定パラメータ .....	511
18.4.2	レイヤのプログラム可能なパラメータ .....	514
18.5	LTDC 割り込み .....	518
18.6	LTDC プログラミング手順 .....	519
18.7	LTDC レジスタ .....	520
18.7.1	LTDC 同期サイズ設定レジスタ (LTDC_SSCR) .....	520
18.7.2	LTDC バックポーチ設定レジスタ (LTDC_BPCR) .....	521
18.7.3	LTDC アクティブ幅設定レジスタ (LTDC_AWCR) .....	522
18.7.4	LTDC 全幅設定レジスタ (LTDC_TWCR) .....	523
18.7.5	LTDC グローバル制御レジスタ (LTDC_GCR) .....	524
18.7.6	LTDC シャドウ再ロード設定レジスタ (LTDC_SRCR) .....	525
18.7.7	LTDC バックグラウンド色設定レジスタ (LTDC_BCCR) .....	526
18.7.8	LTDC 割り込みイネーブルレジスタ (LTDC_IER) .....	527
18.7.9	LTDC 割り込みステータスレジスタ (LTDC_ISR) .....	528
18.7.10	LTDC 割り込みクリアレジスタ (LTDC_ICR) .....	529
18.7.11	LTDC ライン割り込み位置設定レジスタ (LTDC_LIPCR) .....	529
18.7.12	LTDC 現在位置ステータスレジスタ (LTDC_CPSR) .....	530
18.7.13	LTDC 現在表示ステータスレジスタ (LTDC_CDSR) .....	530
18.7.14	LTDC レイヤ x 制御レジスタ (LTDC_LxCR) (x=1..2) .....	531
18.7.15	LTDC レイヤ x ウィンドウ水平位置設定レジスタ (LTDC_LxWHPCR) (x=1..2) .....	532
18.7.16	LTDC レイヤ x ウィンドウ垂直位置設定レジスタ (LTDC_LxWVPCR) (x=1..2) .....	533
18.7.17	LTDC レイヤ x カラーキーイング設定レジスタ (LTDC_LxCCKCR) (x=1..2) .....	534
18.7.18	LTDC レイヤ x ピクセルフォーマット設定レジスタ (LTDC_LxPFCR) (x=1..2) .....	534
18.7.19	LTDC レイヤ x 定数アルファ設定レジスタ (LTDC_LxCACR) (x=1..2) .....	535
18.7.20	LTDC レイヤ x デフォルト色設定レジスタ (LTDC_LxDCCR) (x=1..2) .....	536

18.7.21	LTDC レイヤ x ブレンディング係数設定レジスタ (LTDC_LxBFRCR) (x=1..2) .....	537
18.7.22	LTDC レイヤ x カラーフレームバッファアドレスレジスタ (LTDC_LxCFBAR) (x=1..2) .....	538
18.7.23	LTDC レイヤ x カラーフレームバッファ長レジスタ (LTDC_LxCFBLR) (x=1..2) .....	539
18.7.24	LTDC レイヤ x カラーフレームバッファライン数レジスタ (LTDC_LxWVPCR) (x=1..2) .....	540
18.7.25	LTDC レイヤ x CLUT 書き込みレジスタ (LTDC_LxCLUTWR) (x=1..2) ...	541
18.7.26	LTDC レジスタマップ .....	542
<b>19</b>	<b>乱数発生器 (RNG) .....</b>	<b>545</b>
19.1	概要 .....	545
19.2	RNG の主な機能 .....	545
19.3	RNG の機能説明 .....	545
19.3.1	動作 .....	546
19.3.2	エラー管理 .....	546
19.4	RNG レジスタ .....	547
19.4.1	RNG 制御レジスタ (RNG_CR) .....	547
19.4.2	RNG ステータスレジスタ (RNG_SR) .....	547
19.4.3	RNG データレジスタ (RNG_DR) .....	548
19.4.4	RNG レジスタマップ .....	549
<b>20</b>	<b>暗号プロセッサ (CRYP) .....</b>	<b>550</b>
20.1	CRYP の概要 .....	550
20.2	CRYP の主な機能 .....	550
20.3	CRYP の機能詳細 .....	552
20.3.1	DES / トリプル DES 暗号コア .....	552
20.3.2	AES 暗号コア .....	557
20.3.3	データ型 .....	567
20.3.4	初期化ベクタ CRYP_IV0...1(L/R) .....	570
20.3.5	CRYP ビジーステート .....	571
20.3.6	暗号化または復号化を実施する手順 .....	572
20.3.7	コンテキストスワッピング .....	573



20.4	CRYP 割り込み	575
20.5	CRYP DMA インタフェース	575
20.6	CRYP レジスタ	576
20.6.1	CRYP 制御レジスタ (CRYP_CR)	576
20.6.2	CRYP ステータスレジスタ (CRYP_SR)	578
20.6.3	CRYP データ入力レジスタ (CRYP_DIN)	579
20.6.4	CRYP データ出力レジスタ (CRYP_DOUT)	580
20.6.5	CRYP DMA 制御レジスタ (CRYP_DMACR)	581
20.6.6	CRYP 割り込みマスクセット/クリアレジスタ (CRYP_IMSCR)	581
20.6.7	CRYP raw 割り込みステータスレジスタ (CRYP_RISR)	582
20.6.8	CRYP マスク済み割り込みステータスレジスタ (CRYP_MISR)	582
20.6.9	CRYP キーレジスタ (CRYP_K0...3(L/R)R)	583
20.6.10	CRYP 初期化ベクタレジスタ (CRYP_IV0...1(L/R)R)	585
20.6.11	CRYP コンテキストスワップレジスタ (CRYP_CSGCMCCM0..7R および CRYP_CSGCM0..7R)	586
20.6.12	CRYP レジスタマップ	587
<b>21</b>	<b>ハッシュプロセッサ (HASH)</b>	<b>589</b>
21.1	HASH の概要	589
21.2	HASH の主な機能	589
21.3	HASH の機能詳細	590
21.3.1	処理時間	591
21.3.2	データ型	591
21.3.3	メッセージダイジェストの計算	593
21.3.4	メッセージのパディング	594
21.3.5	ハッシュ操作	595
21.3.6	HMAC 操作	595
21.3.7	コンテキストスワッピング	596
21.3.8	HASH 割り込み	597
21.4	HASH レジスタ	598
21.4.1	HASH 制御レジスタ (HASH_CR)	598
21.4.2	HASH データ入力レジスタ (HASH_DIN)	601
21.4.3	HASH スタートレジスタ (HASH_STR)	602
21.4.4	HASH ダイジェストレジスタ (HASH_HR0..4/5/6/7)	603
21.4.5	HASH 割り込み有効レジスタ (HASH_IMR)	605
21.4.6	HASH ステータスレジスタ (HASH_SR)	606
21.4.7	HASH コンテキストスワップレジスタ (HASH_CSRx)	607

	21.4.8	HASH レジスタマップ	608
<b>22</b>		<b>高機能制御タイマ (TIM1/TIM8)</b>	<b>609</b>
	22.1	TIM1/TIM8 の概要	609
	22.2	TIM1/TIM8 の主な機能	609
	22.3	TIM1/TIM8 の機能説明	611
	22.3.1	タイムベースユニット	611
	22.3.2	カウンタモード	613
	22.3.3	繰り返しカウンタ	624
	22.3.4	外部トリガ入力ブロック	626
	22.3.5	クロック選択	627
	22.3.6	キャプチャ/比較チャンネル	631
	22.3.7	入力キャプチャモード	634
	22.3.8	PWM 入力モード	635
	22.3.9	強制出力モード	636
	22.3.10	出力比較モード	636
	22.3.11	PWM モード	638
	22.3.12	非対称 PWM モード	641
	22.3.13	組み合わせ PWM モード	642
	22.3.14	組み合わせ 3 相 PWM モード	643
	22.3.15	相補出力とデッドタイム挿入	644
	22.3.16	ブレーク機能の使用	646
	22.3.17	外部イベントによる OCxREF 信号のクリア	652
	22.3.18	6 ステップ PWM 生成	654
	22.3.19	ワンパルスモード	655
	22.3.20	再トリガ可能なワンパルスモード (OPM)	656
	22.3.21	エンコーダインタフェースモード	657
	22.3.22	UIF ビットの再配置	659
	22.3.23	タイマ入力 XOR 機能	660
	22.3.24	ホールセンサとのインタフェース	660
	22.3.25	タイマの同期	663
	22.3.26	ADC の同期	666
	22.3.27	DMA バーストモード	667
	22.3.28	デバッグモード	668
	22.4	TIM1/TIM8 レジスタ	669
	22.4.1	TIM1/TIM8 制御レジスタ 1 (TIMx_CR1)	669
	22.4.2	TIM1/TIM8 制御レジスタ 2 (TIMx_CR2)	671



22.4.3	TIM1/TIM8 のスレーブモード制御レジスタ (TIMx_SMCR) .....	673
22.4.4	TIM1/TIM8 DMA / 割り込み有効レジスタ (TIMx_DIER) .....	676
22.4.5	TIM1/TIM8 のステータスレジスタ (TIMx_SR) .....	677
22.4.6	TIM1/TIM8 のイベント生成レジスタ (TIMx_EGR) .....	679
22.4.7	TIM1/TIM8 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) .....	680
22.4.8	TIM1/TIM8 のキャプチャ/比較モードレジスタ 2 (TIMx_CCMR2) .....	685
22.4.9	TIM1/TIM8 のキャプチャ/比較有効レジスタ (TIMx_CCER) .....	686
22.4.10	TIM1/TIM8 のカウンタ (TIMx_CNT) .....	690
22.4.11	TIM1/TIM8 プリスケアラ (TIMx_PSC) .....	690
22.4.12	TIM1/TIM8 自動再ロードレジスタ (TIMx_ARR) .....	690
22.4.13	TIM1/TIM8 繰り返しカウンタレジスタ (TIMx_RCR) .....	691
22.4.14	TIM1/TIM8 キャプチャ/比較レジスタ 1 (TIMx_CCR1) .....	691
22.4.15	TIM1/TIM8 キャプチャ/比較レジスタ 2 (TIMx_CCR2) .....	692
22.4.16	TIM1/TIM8 キャプチャ/比較レジスタ 3 (TIMx_CCR3) .....	692
22.4.17	TIM1/TIM8 キャプチャ/比較レジスタ 4 (TIMx_CCR4) .....	693
22.4.18	TIM1/TIM8 ブレークおよびデッドタイムレジスタ (TIMx_BDTR) .....	693
22.4.19	TIM1/TIM8 DMA 制御レジスタ (TIMx_DCR) .....	698
22.4.20	フル転送のための TIM1/TIM8 DMA アドレス (TIMx_DMAR) .....	699
22.4.21	TIM1/TIM8 のキャプチャ/比較モードレジスタ 3 (TIMx_CCMR3) .....	699
22.4.22	TIM1/TIM8 キャプチャ/比較レジスタ 5 (TIMx_CCR5) .....	700
22.4.23	TIM1/TIM8 キャプチャ/比較レジスタ 6 (TIMx_CCR6) .....	701
22.4.24	TIM1 レジスタマップ .....	701
22.4.25	TIM8 レジスタマップ .....	703
<b>23</b>	<b>汎用タイマ (TIM2/TIM3/TIM4/TIM5) .....</b>	<b>706</b>
23.1	TIM2/TIM3/TIM4/TIM5 の概要 .....	706
23.2	TIM2/TIM3/TIM4/TIM5 の主な機能 .....	706
23.3	TIM2/TIM3/TIM4/TIM5 の機能説明 .....	708
23.3.1	タイムベースユニット .....	708
23.3.2	カウンタモード .....	710
23.3.3	クロック選択 .....	721
23.3.4	キャプチャ/比較チャンネル .....	725
23.3.5	入力キャプチャモード .....	727
23.3.6	PWM 入力モード .....	728
23.3.7	強制出力モード .....	729
23.3.8	出力比較モード .....	729
23.3.9	PWM モード .....	731

23.3.10	非対称 PWM モード	734
23.3.11	組み合わせ PWM モード	734
23.3.12	外部イベントによる OCxREF 信号のクリア	736
23.3.13	ワンパルスモード	737
23.3.14	エンコーダインタフェースモード	738
23.3.15	UIF ビットの再配置	740
23.3.16	タイマ入力 XOR 機能	740
23.3.17	タイマと外部トリガの同期	741
23.3.18	タイマの同期	745
23.3.19	DMA バーストモード	749
23.3.20	デバッグモード	750
23.4	TIM2/TIM3/TIM4/TIM5 レジスタ	751
23.4.1	TIMx 制御レジスタ 1 (TIMx_CR1)	751
23.4.2	TIMx 制御レジスタ 2 (TIMx_CR2)	753
23.4.3	TIMx スレーブモード制御レジスタ (TIMx_SMCR)	754
23.4.4	TIMx DMA / 割り込み有効レジスタ (TIMx_DIER)	757
23.4.5	TIMx ステータスレジスタ (TIMx_SR)	758
23.4.6	TIMx イベント生成レジスタ (TIMx_EGR)	760
23.4.7	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	761
23.4.8	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	765
23.4.9	TIMx キャプチャ/比較有効レジスタ (TIMx_CCER)	767
23.4.10	TIMx カウンタ (TIMx_CNT)	769
23.4.11	TIMx プリスケアラ (TIMx_PSC)	769
23.4.12	TIMx 自動再ロードレジスタ (TIMx_ARR)	770
23.4.13	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCR1)	770
23.4.14	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCR2)	771
23.4.15	TIMx キャプチャ/比較モードレジスタ 3 (TIMx_CCR3)	771
23.4.16	TIMx キャプチャ/比較モードレジスタ 4 (TIMx_CCR4)	772
23.4.17	TIMx DMA 制御レジスタ (TIMx_DCR)	773
23.4.18	完全転送の TIMx DMA アドレス (TIMx_DMAR)	773
23.4.19	TIM2 オプションレジスタ 1 (TIM2_OR)	774
23.4.20	TIM2 オプションレジスタ 1 (TIM5_OR)	774
23.4.21	TIM3 オプションレジスタ 1 (TIM3_OR1)	775
23.4.22	TIMx レジスタマップ	775

<b>24</b>	<b>汎用タイマ (TIM9 から TIM14)</b>	<b>778</b>
24.1	TIM9 から TIM14 の概要	778
24.2	TIM9 から TIM14 の主な機能	778
24.2.1	TIM9/TIM12 の主な機能	778
24.2.2	TIM10/TIM11 および TIM13/TIM14 の主な機能	779
24.3	TIM9 から TIM14 の機能説明	781
24.3.1	タイムベースユニット	781
24.3.2	カウンタモード	783
24.3.3	クロック選択	786
24.3.4	キャプチャ/比較チャネル	788
24.3.5	入力キャプチャモード	790
24.3.6	PWM 入力モード (TIM9/12 の場合のみ)	791
24.3.7	強制出力モード	792
24.3.8	出力比較モード	792
24.3.9	PWM モード	794
24.3.10	ワンパルスモード	795
24.3.11	TIM9/12 外部トリガ同期	797
24.3.12	タイマ同期 (TIM9/12)	800
24.3.13	デバッグモード	800
24.4	TIM9 および TIM12 レジスタ	800
24.4.1	TIM9/12 制御レジスタ 1 (TIMx_CR1)	800
24.4.2	TIM9/12 のスレーブモード制御レジスタ (TIMx_SMCR)	802
24.4.3	TIM9/12 割り込み有効レジスタ (TIMx_DIER)	803
24.4.4	TIM9/12 ステータスレジスタ (TIMx_SR)	804
24.4.5	TIM9/12 のイベント発生レジスタ (TIMx_EGR)	805
24.4.6	TIM9/12 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	806
24.4.7	TIM9/12 のキャプチャ/比較有効レジスタ (TIMx_CCER)	809
24.4.8	TIM9/12 のカウンタ (TIMx_CNT)	810
24.4.9	TIM9/12 プリスケアラ (TIMx_PSC)	810
24.4.10	TIM9/12 自動再ロードレジスタ (TIMx_ARR)	810
24.4.11	TIM9/12 キャプチャ/比較モードレジスタ 1 (TIMx_CCR1)	811
24.4.12	TIM9/12 キャプチャ/比較モードレジスタ 2 (TIMx_CCR2)	811
24.4.13	TIM9/12 レジスタマップ	812
24.5	TIM10/11/13/14 レジスタ	814
24.5.1	TIM10/11/13/14 制御レジスタ 1 (TIMx_CR1)	814
24.5.2	TIM10/11/13/14 割り込み有効レジスタ (TIMx_DIER)	815

24.5.3	TIM10/11/13/14 ステータスレジスタ (TIMx_SR) .....	815
24.5.4	TIM10/11/13/14 のイベント発生レジスタ (TIMx_EGR) .....	816
24.5.5	TIM10/11/13/14 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) ....	816
24.5.6	TIM10/11/13/14 のキャプチャ/比較有効レジスタ (TIMx_CCER) .....	819
24.5.7	TIM10/11/13/14 のカウンタ (TIMx_CNT) .....	820
24.5.8	TIM10/11/13/14 プリスケーラ (TIMx_PSC) .....	820
24.5.9	TIM10/11/13/14 自動再ロードレジスタ (TIMx_ARR) .....	820
24.5.10	TIM10/11/13/14 キャプチャ/比較レジスタ 1 (TIMx_CCR1) .....	821
24.5.11	TIM11 オプションレジスタ 1 (TIM11_OR) .....	821
24.5.12	TIM10/11/13/14 レジスタマップ .....	822
<b>25</b>	<b>基本タイマ (TIM6/TIM7) .....</b>	<b>824</b>
25.1	TIM6/TIM7 の概要 .....	824
25.2	TIM6/TIM7 の主な機能 .....	824
25.3	TIM6/TIM7 の機能詳細 .....	825
25.3.1	タイムベースユニット .....	825
25.3.2	カウントモード .....	827
25.3.3	UIF ビットの再配置 .....	830
25.3.4	クロックソース .....	830
25.3.5	デバッグモード .....	831
25.4	TIM6/TIM7 レジスタ .....	831
25.4.1	TIM6/TIM7 制御レジスタ 1 (TIMx_CR1) .....	831
25.4.2	TIM6/TIM7 制御レジスタ 2 (TIMx_CR2) .....	833
25.4.3	TIM6/TIM7 DMA / 割り込み有効レジスタ (TIMx_DIER) .....	833
25.4.4	TIM6/TIM7 のステータスレジスタ (TIMx_SR) .....	834
25.4.5	TIM6/TIM7 のイベント生成レジスタ (TIMx_EGR) .....	834
25.4.6	TIM6/TIM7 のカウンタ (TIMx_CNT) .....	835
25.4.7	TIM6/TIM7 プリスケーラ (TIMx_PSC) .....	835
25.4.8	TIM6/TIM7 の自動再ロードレジスタ (TIMx_ARR) .....	835
25.4.9	TIM6/TIM7 レジスタマップ .....	836

<b>26</b>	<b>低電力タイマ (LPTIM) .....</b>	<b>837</b>
26.1	概要 .....	837
26.2	LPTIM の主な機能 .....	837
26.3	LPTIM の実装 .....	837
26.4	LPTIM の機能詳細 .....	838
26.4.1	LPTIM ブロック図 .....	838
26.4.2	LPTIM のリセットとクロック .....	838
26.4.3	グリッチフィルタ .....	839
26.4.4	プリスケーラ .....	840
26.4.5	トリガマルチプレクサ .....	840
26.4.6	動作モード .....	841
26.4.7	タイムアウト機能 .....	842
26.4.8	波形生成 .....	843
26.4.9	レジスタの更新 .....	844
26.4.10	カウンタモード .....	844
26.4.11	タイマ有効 .....	845
26.4.12	エンコーダモード .....	845
26.5	LPTIM 割り込み .....	846
26.6	LPTIM レジスタ .....	847
26.6.1	LPTIM 割り込みおよびステータスレジスタ (LPTIMx_ISR) .....	847
26.6.2	LPTIM 割り込みクリアレジスタ (LPTIMx_ICR) .....	848
26.6.3	LPTIM 割り込み有効レジスタ (LPTIMx_IER) .....	849
26.6.4	LPTIM 設定レジスタ (LPTIMx_CFGR) .....	850
26.6.5	LPTIM 制御レジスタ (LPTIMx_CR) .....	853
26.6.6	LPTIM 比較レジスタ (LPTIMx_CMP) .....	854
26.6.7	LPTIM 自動再ロードレジスタ (LPTIMx_ARR) .....	854
26.6.8	LPTIM カウンタレジスタ (LPTIMx_CNT) .....	855
26.6.9	LPTIM1 オプションレジスタ (LPTIM1_OR) .....	855
26.6.10	LPTIM2 オプションレジスタ (LPTIM2_OR) .....	856
26.6.11	LPTIM レジスタマップ .....	857
<b>27</b>	<b>独立型ウォッチドッグ (IWDG) .....</b>	<b>858</b>
27.1	概要 .....	858
27.2	IWDG の主な機能 .....	858
27.3	IWDG の機能説明 .....	858
27.3.1	IWDG ブロック図 .....	858

27.3.2	ウィンドウオプション	859
27.3.3	ハードウェアウォッチドッグ	860
27.3.4	低電力凍結	860
27.3.5	レジスタのアクセス保護	860
27.3.6	デバッグモード	860
27.4	IWDG レジスタ	861
27.4.1	キーレジスタ (IWDG_KR)	861
27.4.2	プリスケアラレジスタ (IWDG_PR)	862
27.4.3	再ロードレジスタ (IWDG_RLR)	863
27.4.4	ステータスレジスタ (IWDG_SR)	864
27.4.5	ウィンドウレジスタ (IWDG_WINR)	865
27.4.6	IWDG レジスタマップ	866
<b>28</b>	<b>システムウィンドウ型ウォッチドッグ (WWDG)</b>	<b>867</b>
28.1	概要	867
28.2	WWDG の主な機能	867
28.3	WWDG の機能説明	867
28.3.1	ウォッチドッグの有効化	868
28.3.2	ダウンカウンタの制御	868
28.3.3	高度なウォッチドッグ割り込み機能	869
28.3.4	ウォッチドッグタイムアウトをプログラムする方法	869
28.3.5	デバッグモード	870
28.4	WWDG レジスタ	871
28.4.1	制御レジスタ (WWDG_CR)	871
28.4.2	設定レジスタ (WWDG_CFR)	872
28.4.3	ステータスレジスタ (WWDG_SR)	872
28.4.4	WWDG レジスタマップ	873
<b>29</b>	<b>リアルタイムクロック (RTC)</b>	<b>874</b>
29.1	概要	874
29.2	RTC の主な機能	874
29.3	RTC の機能説明	875
29.3.1	RTC ブロック図	875
29.3.2	RTC によって制御される GPIO	876
29.3.3	クロックとプリスケアラ	878
29.3.4	リアルタイムクロックとカレンダー	879

29.3.5	プログラム可能なアラーム	880
29.3.6	周期的自動ウェイクアップ	880
29.3.7	RTC の初期化と設定	881
29.3.8	カレンダーの読み出し	882
29.3.9	RTC のリセット	883
29.3.10	RTC の同期	884
29.3.11	RTC リファレンスクロック検出	884
29.3.12	RTC の高精度デジタル較正	885
29.3.13	タイムスタンプ機能	887
29.3.14	タンパ検出	888
29.3.15	較正クロック出力	890
29.3.16	アラーム出力	890
29.4	RTC 低電力モード	890
29.5	RTC 割り込み	891
29.6	RTC レジスタ	892
29.6.1	RTC 時刻レジスタ (RTC_TR)	892
29.6.2	RTC 日付レジスタ (RTC_DR)	893
29.6.3	RTC 制御レジスタ (RTC_CR)	894
29.6.4	RTC 初期化とステータスレジスタ (RTC_ISR)	897
29.6.5	RTC プリスケアラレジスタ (RTC_PRER)	900
29.6.6	RTC ウェイクアップタイムレジスタ (RTC_WUTR)	901
29.6.7	RTC アラーム A レジスタ (RTC_ALRMAR)	902
29.6.8	RTC アラーム B レジスタ (RTC_ALRMBR)	903
29.6.9	RTC 書き込み保護レジスタ (RTC_WPR)	904
29.6.10	RTC サブセカンドレジスタ (RTC_SSR)	904
29.6.11	RTC シフト制御レジスタ (RTC_SHIFTR)	905
29.6.12	RTC タイムスタンプ時刻レジスタ (RTC_TSTR)	906
29.6.13	RTC タイムスタンプ日付レジスタ (RTC_TSDR)	907
29.6.14	RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)	908
29.6.15	RTC 較正レジスタ (RTC_CALR)	909
29.6.16	RTC タンパ設定レジスタ (RTC_TAMPCR)	910
29.6.17	RTC アラーム A サブセカンドレジスタ (RTC_ALRMASR)	913
29.6.18	RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)	914
29.6.19	RTC オプションレジスタ (RTC_OR)	915
29.6.20	RTC バックアップレジスタ (RTC_BKPxR)	916
29.6.21	RTC レジスタマップ	917

<b>30</b>	<b>I2C (Inter-integrated circuit) インタフェース</b>	<b>919</b>
30.1	概要	919
30.2	I2C の主な機能	919
30.3	I2C の実装	920
30.4	I2C の機能詳細	920
30.4.1	I2C ブロック図	921
30.4.2	I2C クロックの要件	922
30.4.3	モード選択	922
30.4.4	I2C の初期化	924
30.4.5	ソフトウェアリセット	928
30.4.6	データ転送	929
30.4.7	I2C スレーブモード	931
30.4.8	I2C マスタモード	940
30.4.9	I2C_TIMINGR レジスタの設定例	952
30.4.10	SMBus 固有の機能	952
30.4.11	SMBus 初期化	955
30.4.12	SMBus : I2C_TIMEOCTR レジスタの設定例	957
30.4.13	SMBus スレーブモード	957
30.4.14	エラー条件	965
30.4.15	DMA リクエスト	966
30.4.16	デバッグモード	967
30.5	I2C 低電力モード	967
30.6	I2C 割り込み	968
30.7	I2C レジスタ	969
30.7.1	制御レジスタ 1 (I2C_CR1)	969
30.7.2	制御レジスタ 2 (I2C_CR2)	973
30.7.3	Own Address 1 レジスタ (I2C_OAR1)	976
30.7.4	Own Address 2 レジスタ (I2C_OAR2)	977
30.7.5	タイミングレジスタ (I2C_TIMINGR)	978
30.7.6	タイムアウトレジスタ (I2C_TIMEOCTR)	979
30.7.7	割り込みおよびステータスレジスタ (I2C_ISR)	980
30.7.8	割り込みクリアレジスタ (I2C_ICR)	982
30.7.9	PEC レジスタ (I2C_PECR)	983
30.7.10	受信データレジスタ (I2C_RXDR)	984
30.7.11	送信データレジスタ (I2C_TXDR)	984
30.7.12	I2C レジスタマップ	985



<b>31</b>	<b>USART</b> <b>(Universal synchronous asynchronous receiver transmitter) . . . . .</b>	<b>987</b>
31.1	概要 . . . . .	987
31.2	USART の主な機能 . . . . .	987
31.3	USART の拡張機能 . . . . .	988
31.4	USART の実装 . . . . .	989
31.5	USART の機能詳細 . . . . .	989
31.5.1	USART キャラクタの説明 . . . . .	992
31.5.2	トランスミッタ . . . . .	993
31.5.3	レシーバ . . . . .	996
31.5.4	ボーレート生成 . . . . .	1002
31.5.5	クロック偏差に対する USART レシーバの許容誤差 . . . . .	1005
31.5.6	自動ボーレート検出 . . . . .	1006
31.5.7	マルチプロセッサ通信 . . . . .	1007
31.5.8	Modbus 通信 . . . . .	1009
31.5.9	パリティ制御 . . . . .	1010
31.5.10	LIN (Local Interconnection Network) モード . . . . .	1011
31.5.11	USART 同期モード . . . . .	1013
31.5.12	単線半二重通信 . . . . .	1016
31.5.13	スマートカードモード . . . . .	1016
31.5.14	Ir SIR ENDEC ブロック . . . . .	1021
31.5.15	DMA を使用した連続通信 . . . . .	1023
31.5.16	RS232 ハードウェアフロー制御および RS485 ドライバ有効 . . . . .	1025
31.6	USART 低電力モード . . . . .	1027
31.7	USART 割り込み . . . . .	1028
31.8	USART レジスタ . . . . .	1030
31.8.1	制御レジスタ 1 (USARTx_CR1) . . . . .	1030
31.8.2	制御レジスタ 2 (USARTx_CR2) . . . . .	1033
31.8.3	制御レジスタ 3 (USARTx_CR3) . . . . .	1037
31.8.4	ボーレートレジスタ (USARTx_BRR) . . . . .	1040
31.8.5	ガード時間およびプリスケアラレジスタ (USARTx_GTPR) . . . . .	1041
31.8.6	レシーバタイムアウトレジスタ (USARTx_RTOR) . . . . .	1042
31.8.7	リクエストレジスタ (USARTx_RQR) . . . . .	1043
31.8.8	割り込みおよびステータスレジスタ (USARTx_ISR) . . . . .	1044
31.8.9	割り込みフラグクリアレジスタ (USARTx_ICR) . . . . .	1048
31.8.10	受信データレジスタ (USARTx_RDR) . . . . .	1049

31.8.11	送信データレジスタ (USARTx_TDR) .....	1049
31.8.12	USART レジスタマップ .....	1050
<b>32</b>	<b>シリアルペリフェラルインタフェース / I2S (SPI/I2S) .....</b>	<b>1051</b>
32.1	概要 .....	1051
32.2	SPI の主な機能 .....	1051
32.3	I2S の主な機能 .....	1052
32.4	SPI/I2S の実装 .....	1052
32.5	SPI の機能説明 .....	1053
32.5.1	概要 .....	1053
32.5.2	マスタとスレーブの 1 対 1 の通信 .....	1054
32.5.3	標準マルチスレーブ通信 .....	1056
32.5.4	スレーブ選択 (NSS) ピンの管理 .....	1057
32.5.5	通信フォーマット .....	1059
32.5.6	SPI の設定 .....	1061
32.5.7	SPI を有効にする手順 .....	1062
32.5.8	データの送受信手順 .....	1062
32.5.9	SPI ステータスフラグ .....	1072
32.5.10	SPI エラーフラグ .....	1073
32.5.11	NSS パルスモード .....	1074
32.5.12	TI モード .....	1075
32.5.13	CRC 計算 .....	1076
32.6	SPI 割り込み .....	1077
32.7	I <sup>2</sup> S の機能説明 .....	1078
32.7.1	I <sup>2</sup> S の概要 .....	1078
32.7.2	サポートされるオーディオプロトコル .....	1079
32.7.3	起動に関する説明 .....	1086
32.7.4	クロックジェネレータ .....	1087
32.7.5	I <sup>2</sup> S マスタモード .....	1089
32.7.6	I <sup>2</sup> S スレーブモード .....	1091
32.7.7	I <sup>2</sup> S ステータスフラグ .....	1093
32.7.8	I <sup>2</sup> S エラーフラグ .....	1094
32.7.9	DMA の機能 .....	1094
32.8	I <sup>2</sup> S 割り込み .....	1095
32.9	SPI および I <sup>2</sup> S レジスタ .....	1096
32.9.1	SPI 制御レジスタ 1 (SPIx_CR1) .....	1096



32.9.2	SPI 制御レジスタ 2 (SPIx_CR2) .....	1098
32.9.3	SPI ステータスレジスタ (SPIx_SR) .....	1101
32.9.4	SPI データレジスタ (SPIx_DR) .....	1102
32.9.5	SPI CRC 多項式レジスタ (SPIx_CRCPR) .....	1103
32.9.6	SPI Rx CRC レジスタ (SPIx_RXCRCR) .....	1103
32.9.7	SPI Tx CRC レジスタ (SPIx_TXCRCR) .....	1104
32.9.8	SPIx_I <sup>2</sup> S 設定レジスタ (SPIx_I2SCFGR) .....	1104
32.9.9	SPIx_I <sup>2</sup> S プリスケアラレジスタ (SPIx_I2SPR) .....	1106
32.9.10	SPI/I2S レジスタマップ .....	1107
<b>33</b>	<b>シリアルオーディオインタフェース (SAI) .....</b>	<b>1108</b>
33.1	概要 .....	1108
33.2	SAI の主な機能 .....	1109
33.3	SAI の機能詳細 .....	1110
33.3.1	SAI ブロック図 .....	1110
33.3.2	主要な SAI モード .....	1111
33.3.3	SAI 同期モード .....	1112
33.3.4	オーディオデータサイズ .....	1113
33.3.5	フレーム同期 .....	1113
33.3.6	スロットの設定 .....	1116
33.3.7	SAI クロックジェネレータ .....	1118
33.3.8	内部 FIFO .....	1120
33.3.9	AC'97 リンクコントローラ .....	1122
33.3.10	SPDIF 出力 .....	1124
33.3.11	特有の機能 .....	1126
33.3.12	エラーフラグ .....	1131
33.3.13	SAI の無効化 .....	1134
33.3.14	SAI DMA インタフェース .....	1135
33.4	SAI 割り込み .....	1136
33.5	SAI レジスタ .....	1137
33.5.1	グローバル設定レジスタ (SAI_GCR) .....	1137
33.5.2	設定レジスタ 1 (SAI_ACR1 / SAI_BCR1) .....	1137
33.5.3	設定レジスタ 2 (SAI_ACR2 / SAI_BCR2) .....	1141
33.5.4	フレーム設定レジスタ (SAI_AFRCR / SAI_BFRCR) .....	1143
33.5.5	スロットレジスタ (SAI_ASLOTR / SAI_BSOLOTR) .....	1145
33.5.6	割り込みマスクレジスタ 2 (SAI_AIM / SAI_BIM) .....	1146
33.5.7	ステータスレジスタ (SAI_ASR / SAI_BSR) .....	1148

33.5.8	クリアフラグレジスタ (SAI_ACLRFR / SAI_BCLRFR)	1151
33.5.9	データレジスタ (SAI_ADR / SAI_BDR)	1152
33.5.10	SAI レジスタマップ	1152
<b>34</b>	<b>SPDIF 受信インタフェース (SPDIFRX)</b>	<b>1154</b>
34.1	SPDIFRX インタフェースの概要	1154
34.2	SPDIFRX の主な機能	1154
34.3	SPDIFRX の機能詳細	1154
34.3.1	S/PDIF プロトコル (IEC-60958)	1155
34.3.2	SPDIFRX デコーダ (SPDIFRX_DC)	1158
34.3.3	クロック偏差に対する SPDIFRX の許容誤差	1161
34.3.4	SPDIFRX の同期	1162
34.3.5	SPDIFRX 処理	1164
34.3.6	データ受信管理	1166
34.3.7	専用制御フロー	1168
34.3.8	受信エラー	1169
34.3.9	クロック供給の方法	1172
34.3.10	DMA インタフェース	1172
34.3.11	割り込みの生成	1173
34.3.12	レジスタの保護	1174
34.4	プログラミング手順	1175
34.4.1	初期化フェーズ	1175
34.4.2	SPDIFRX からの割り込みの処理	1176
34.4.3	DMA からの割り込みの処理	1177
34.5	SPDIFRX インタフェースレジスタ	1178
34.5.1	制御レジスタ (SPDIFRX_CR)	1178
34.5.2	割り込みマスクレジスタ (SPDIFRX_IMR)	1180
34.5.3	ステータスレジスタ (SPDIFRX_SR)	1181
34.5.4	割り込みフラグクリアレジスタ (SPDIFRX_IFCR)	1183
34.5.5	データ入力レジスタ (SPDIFRX_DR)	1184
34.5.6	データ入力レジスタ (SPDIFRX_DR)	1185
34.5.7	データ入力レジスタ (SPDIFRX_DR)	1186
34.5.8	チャンネルステータスレジスタ (SPDIFRX_CSR)	1187
34.5.9	デバッグ情報レジスタ (SPDIFRX_DIR)	1188
34.5.10	SPDIFRX インタフェースレジスタマップ	1189



<b>35</b>	<b>SD/SDIO/MMC カードホストインタフェース (SDMMC) . . . . .</b>	<b>1190</b>
35.1	SDMMC の主な特長 . . . . .	1190
35.2	SDMMC バストポロジ . . . . .	1190
35.3	SDMMC 機能詳細 . . . . .	1192
35.3.1	SDMMC アダプタ . . . . .	1194
35.3.2	SDMMC APB2 インタフェース . . . . .	1205
35.4	カード機能詳細 . . . . .	1207
35.4.1	カード識別モード . . . . .	1207
35.4.2	カードリセット . . . . .	1207
35.4.3	動作電圧範囲の検証 . . . . .	1207
35.4.4	カード識別プロセス . . . . .	1207
35.4.5	ブロック書き込み . . . . .	1208
35.4.6	ブロック読み出し . . . . .	1209
35.4.7	ストリームアクセス、ストリーム書き込み、およびストリーム読み出し (MultiMediaCard のみ) . . . . .	1209
35.4.8	消去：グループ消去とセクタ消去 . . . . .	1211
35.4.9	ワイドバス選択または選択解除 . . . . .	1211
35.4.10	保護管理 . . . . .	1211
35.4.11	カードステータスレジスタ . . . . .	1215
35.4.12	SD ステータスレジスタ . . . . .	1218
35.4.13	SD I/O モード . . . . .	1222
35.4.14	コマンドおよびレスポンス . . . . .	1223
35.5	レスポンスフォーマット . . . . .	1226
35.5.1	R1 (ノーマルレスポンスコマンド) . . . . .	1226
35.5.2	R1b . . . . .	1227
35.5.3	R2 (CID、CSD レジスタ) . . . . .	1227
35.5.4	R3 (OCR レジスタ) . . . . .	1228
35.5.5	R4 (高速 I/O) . . . . .	1228
35.5.6	R4b . . . . .	1229
35.5.7	R5 (割り込みリクエスト) . . . . .	1229
35.5.8	R6 . . . . .	1230
35.6	SDIO I/O カード固有の操作 . . . . .	1230
35.6.1	SDMMC_D2 信号による SDIO I/O 読み出しウェイト操作 . . . . .	1230
35.6.2	SDMMC_CK の停止による SDIO 読み出しウェイト操作 . . . . .	1231
35.6.3	SDIO サスペンド/レジューム操作 . . . . .	1231
35.6.4	SDIO 割込み . . . . .	1231

35.7	HW フロー制御	1232
35.8	SDMMC レジスタ	1232
35.8.1	SDMMC 電源制御レジスタ (SDMMC_POWER)	1232
35.8.2	SDMMC クロック制御レジスタ (SDMMC_CLKCR)	1233
35.8.3	SDMMC 引数レジスタ (SDMMC_ARG)	1234
35.8.4	SDMMC コマンドレジスタ (SDMMC_CMD)	1235
35.8.5	SDMMC コマンドレスポンスレジスタ (SDMMC_RESPCMD)	1235
35.8.6	SDMMC レスポンス 1 ~ 4 レジスタ (SDMMC_RESPx)	1236
35.8.7	SDMMC データタイマレジスタ (SDMMC_DTIMER)	1237
35.8.8	SDMMC データ長レジスタ (SDMMC_DLEN)	1238
35.8.9	SDMMC データ制御レジスタ (SDMMC_DCTRL)	1238
35.8.10	SDMMC データカウンタレジスタ (SDMMC_DCOUNT)	1240
35.8.11	SDMMC ステータスレジスタ (SDMMC_STA)	1240
35.8.12	SDMMC 割り込みクリアレジスタ (SDMMC_ICR)	1242
35.8.13	SDMMC マスクレジスタ (SDMMC_MASK)	1243
35.8.14	SDMMC FIFO カウンタレジスタ (SDMMC_FIFOCNT)	1246
35.8.15	SDMMC データ FIFO レジスタ (SDMMC_FIFO)	1246
35.8.16	SDMMC レジスタマップ	1247
<b>36</b>	<b>コントローラエリアネットワーク (bxCAN)</b>	<b>1249</b>
36.1	概要	1249
36.2	bxCAN の主な機能	1249
36.3	bxCAN の概要	1250
36.3.1	CAN 2.0B アクティブコア	1250
36.3.2	制御レジスタ、ステータスレジスタ、設定レジスタ	1250
36.3.3	送信メールボックス	1250
36.3.4	受信フィルタ	1251
36.4	bxCAN 動作モード	1252
36.4.1	初期化モード	1252
36.4.2	通常モード	1252
36.4.3	スリープモード (低電力)	1253
36.5	テストモード	1254
36.5.1	サイレントモード	1254
36.5.2	ループバックモード	1254
36.5.3	サイレントモードとループバックモードの組み合わせ	1255

36.6	デバッグモードでの挙動	1255
36.7	bxCAN 機能の説明	1255
36.7.1	送信処理	1255
36.7.2	タイムトリガ通信モード	1257
36.7.3	受信処理	1257
36.7.4	識別子フィルタリング	1259
36.7.5	メッセージストレージ	1263
36.7.6	エラー管理	1265
36.7.7	ビットタイミング	1265
36.8	bxCAN の割込み	1268
36.9	CAN レジスタ	1269
36.9.1	レジスタのアクセス保護	1269
36.9.2	CAN 制御/ステータスレジスタ	1269
36.9.3	CAN メールボックスレジスタ	1280
36.9.4	CAN フィルタレジスタ	1287
36.9.5	bxCAN レジスタマップ	1291
<b>37</b>	<b>USB On-The-Go フルスピード/ハイスピード (OTG_FS/OTG_HS)</b>	<b>1295</b>
37.1	概要	1295
37.2	USB_OTG の主な機能	1296
37.2.1	標準機能	1296
37.2.2	ホストモード機能	1297
37.2.3	ペリフェラルモード機能	1297
37.3	USB_OTG の実装	1297
37.4	USB OTG の機能詳細	1298
37.4.1	USB OTG ブロック図	1298
37.4.2	OTG コア	1299
37.4.3	フルスピード OTG PHY	1300
37.4.4	内蔵フルスピード OTG PHY	1300
37.4.5	ハイスピード OTG PHY	1301
37.4.6	I2C インタフェースを使用する外部フルスピード OTG PHY	1301
37.5	OTG デュアルロールデバイス (DRD)	1301
37.5.1	ID ラインの検出	1302
37.5.2	HNP デュアルロールデバイス	1302
37.5.3	SRP デュアルロールデバイス	1302

37.6	USB ペリフェラル	1303
37.6.1	SRP 対応ペリフェラル	1304
37.6.2	ペリフェラルの状態	1304
37.6.3	ペリフェラルエンドポイント	1305
37.7	USB ホスト	1307
37.7.1	SRP 対応ホスト	1308
37.7.2	USB ホストの状態	1308
37.7.3	ホストチャンネル	1310
37.7.4	ホストスケジューラ	1311
37.8	SOF トリガ	1312
37.8.1	ホストの SOF	1312
37.8.2	ペリフェラルの SOF	1313
37.9	電力オプション	1313
37.10	OTG_HFIR レジスタの動的更新	1314
37.11	USB データ FIFO	1314
37.11.1	ペリフェラル FIFO アーキテクチャ	1315
37.11.2	ホスト FIFO アーキテクチャ	1316
37.11.3	FIFO RAM の割り当て	1317
37.12	OTG_FS システムのパフォーマンス	1318
37.13	OTG_FS/OTG_HS 割り込み	1319
37.14	OTG_FS/OTG_HS 制御およびステータスレジスタ	1320
37.14.1	CSR メモリマップ	1320
37.15	OTG_FS/OTG_HS レジスタ	1326
37.15.1	OTG 制御およびステータスレジスタ (OTG_GOTGCTL)	1326
37.15.2	OTG 割り込みレジスタ (OTG_GOTGINT)	1329
37.15.3	OTG AHB 設定レジスタ (OTG_GAHBCFG)	1330
37.15.4	OTG USB 設定レジスタ (OTG_GUSBCFG)	1332
37.15.5	OTG リセットレジスタ (OTG_GRSTCTL)	1335
37.15.6	OTG コア割り込みレジスタ (OTG_GINTSTS)	1337
37.15.7	OTG 割り込みマスクレジスタ (OTG_GINTMSK)	1342
37.15.8	OTG_FS 受信ステータスデバッグ読み出し / OTG ステータス読み出し およびポップレジスタ (OTG_GRXSTSR/OTG_GRXSTSP)	1345
37.15.9	OTG 受信 FIFO サイズレジスタ (OTG_GRXFSIZ)	1348
37.15.10	OTG ホスト非周期的送信 FIFO サイズレジスタ (OTG_HNPTXFSIZ) / エンドポイント 0 送信 FIFO サイズ (OTG_DIEPTXF0)	1348
37.15.11	OTG 非周期的送信 FIFO / キューステータスレジスタ (OTG_HNPTXSTS)	1349

37.15.12	OTG I <sup>2</sup> C アクセスレジスタ (OTG_GI2CCTL)	1350
37.15.13	OTG 一般コア設定レジスタ (OTG_GCCFG)	1351
37.15.14	OTG コア ID レジスタ (OTG_CID)	1352
37.15.15	OTG コア LPM 設定レジスタ (OTG_GLPMCFG)	1352
37.15.16	OTG ホスト周期的送信 FIFO サイズレジスタ (OTG_HPTXFSIZ)	1356
37.15.17	OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_DIEPTXFx)(x = 1..5[FS] / 7[HS]、ここで x は FIFO_number)	1357
37.15.18	ホストモードレジスタ	1357
37.15.19	OTG ホスト設定レジスタ (OTG_HCFG)	1358
37.15.20	OTG ホストフレームインターバルレジスタ (OTG_HFIR)	1359
37.15.21	OTG ホストフレーム番号/残りフレーム時間 レジスタ (OTG_HFNUM)	1360
37.15.22	OTG ホスト周期的送信 FIFO / キューステータスレジスタ (OTG_HPTXSTS)	1361
37.15.23	OTG ホスト全チャンネル割り込みレジスタ (OTG_HAINT)	1362
37.15.24	OTG ホスト全チャンネル割り込みマスクレジスタ (OTG_HAINTMSK)	1362
37.15.25	OTG ホストポート制御およびステータスレジスタ (OTG_HPRT)	1363
37.15.26	OTG ホストチャンネル x 特性レジスタ (OTG_HCCHARx) (x = 0..15[HS] / 11[FS]、ここで x = Channel_number)	1365
37.15.27	OTG ホストチャンネル x スプリット制御レジスタ (OTG_HCSPLTx) (x = 0..15、ここで x = Channel_number)	1367
37.15.28	OTG ホストチャンネル x 割り込みレジスタ (OTG_HCINTx) (x = 0..15[HS] / 11[FS]、ここで x = Channel_number)	1368
37.15.29	OTG ホストチャンネル x 割り込みマスクレジスタ (OTG_HCINTMSKx) (x = 0..15[HS] / 11[FS]、ここで x = Channel_number)	1369
37.15.30	OTG ホストチャンネル x 転送サイズレジスタ (OTG_HCTSIZx) (x = 0..15[HS] / 11[FS]、ここで x = Channel_number)	1371
37.15.31	OTG ホストチャンネル x DMA アドレスレジスタ (OTG_HCDMAx) (x = 0..15、ここで x = Channel_number)	1372
37.15.32	デバイスモードレジスタ	1372
37.15.33	OTG デバイス設定レジスタ (OTG_DCFG)	1372
37.15.34	OTG デバイス制御レジスタ (OTG_DCTL)	1374
37.15.35	OTG デバイスステータスレジスタ (OTG_DSTS)	1376
37.15.36	OTG デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG_DIEPMSK)	1377
37.15.37	OTG デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG_DOEPMSK)	1379
37.15.38	OTG デバイス全エンドポイント割り込みレジスタ (OTG_DAINT)	1380
37.15.39	OTG 全エンドポイント割り込みマスクレジスタ (OTG_DAINTMSK)	1381
37.15.40	OTG デバイス V <sub>BUS</sub> 放電時間レジスタ (OTG_DVBUSDIS)	1381
37.15.41	OTG デバイス V <sub>BUS</sub> パルシング時間レジスタ (OTG_DVBUSPULSE)	1382

37.15.42	OTG デバイス閾値制御レジスタ (OTG_DTHRCTL) .....	1382
37.15.43	OTG デバイス各エンドポイント割り込みレジスタ (OTG_DEACHINT) ...	1383
37.15.44	OTG デバイス IN エンドポイント FIFO エンプティ割り込みマスクレジスタ (OTG_DIEPEMPMSK) .....	1384
37.15.45	OTG デバイス各エンドポイント割り込みレジスタマスク (OTG_DEACHINTMSK) 1384	
37.15.46	OTG デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_DIEPCTL0) .....	1385
37.15.47	OTG デバイスエンドポイント x 制御レジスタ (OTG_DIEPCTLx) (x = 1..5[FS] / 0..7[HS]、ここで x = Endpoint_number) .....	1386
37.15.48	OTG デバイス制御 OUT エンドポイント 0 制御レジスタ (OTG_DOEPCTL0) .....	1389
37.15.49	OTG デバイスエンドポイント x 制御レジスタ (OTG_DOEPCTLx) (x = 1..5[FS] / 7[HS]、ここで x = Endpoint_number) .....	1390
37.15.50	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DIEPINTx) (x = 0..5[FS] / 7[HS]、ここで x = Endpoint_number) .....	1392
37.15.51	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DOEPINTx) (x = 0..5[FS] / 7[HS]、ここで x = Endpoint_number) .....	1394
37.15.52	OTG デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_DIEPTSIZ0) .....	1395
37.15.53	OTG デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG_DOEPTSIZ0) .....	1396
37.15.54	OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG_DIEPTSIZx) (x = 1..5[FS] / 7[HS]、ここで x = Endpoint_number) ..	1397
37.15.55	OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_DTXFSTSx) (x = 0..5[FS] / 7[HS]、ここで x = Endpoint_number) ..	1398
37.15.56	OTG デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_DOEPTSIZx) (x = 1..5[FS] / 7[HS]、ここで x = Endpoint_number) .	1398
37.15.57	OTG パワーおよびクロックゲーティング制御レジスタ (OTG_PCGCTL) .....	1400
37.15.58	OTG_FS/OTG_HS レジスタマップ .....	1401
37.16	OTG_FS/OTG_HS プログラミングモデル .....	1411
37.16.1	コアの初期化 .....	1411
37.16.2	ホストの初期化 .....	1412
37.16.3	デバイスの初期化 .....	1412
37.16.4	DMA モード .....	1413
37.16.5	ホストプログラミングモデル .....	1413
37.16.6	デバイスプログラミングモデル .....	1445
37.16.7	最悪ケースの応答時間 .....	1464
37.16.8	OTG プログラミングモデル .....	1465

<b>38</b>	<b>イーサネット (ETH) : メディアアクセス制御 (MAC)、DMA コントローラ付き</b> .....	<b>1471</b>
38.1	イーサネットの概要 .....	1471
38.2	イーサネットの主な機能 .....	1471
38.2.1	MAC コア機能 .....	1471
38.2.2	DMA の機能 .....	1473
38.2.3	PTP の機能 .....	1473
38.3	イーサネットピン .....	1474
38.4	イーサネット機能の説明 : SMI、MII、RMII .....	1475
38.4.1	ステーション管理インタフェース : SMI .....	1475
38.4.2	メディア独立インタフェース : MII .....	1479
38.4.3	減少メディア独立インタフェース : RMII .....	1481
38.4.4	MII/RMII の選択 .....	1482
38.5	イーサネット機能の説明 : MAC 802.3 .....	1483
38.5.1	MAC 802.3 フレームフォーマット .....	1483
38.5.2	MAC フレームの送信 .....	1487
38.5.3	MAC フレームの受信 .....	1494
38.5.4	MAC 割り込み .....	1499
38.5.5	MAC フィルタリング .....	1500
38.5.6	MAC ループバックモード .....	1503
38.5.7	MAC 管理カウンタ : MMC .....	1503
38.5.8	電源管理 : PMT .....	1504
38.5.9	高精度時間プロトコル (IEEE1588 PTP) .....	1507
38.6	イーサネット機能の説明 : DMA コントローラ動作 .....	1513
38.6.1	DMA を使用した転送の初期化 .....	1514
38.6.2	ホストバスバーストアクセス .....	1515
38.6.3	ホストデータバッファ整列 .....	1515
38.6.4	バッファサイズの計算 .....	1516
38.6.5	DMA アービタ .....	1516
38.6.6	DMA へのエラー応答 .....	1516
38.6.7	Tx DMA の設定 .....	1517
38.6.8	Rx DMA の設定 .....	1528
38.6.9	DMA 割り込み .....	1539
38.7	イーサネット割り込み .....	1540
38.8	イーサネットレジスタの説明 .....	1541
38.8.1	MAC レジスタの説明 .....	1541

38.8.2	MMC レジスタの説明	1560
38.8.3	IEEE 1588 タイムスタンプレジスタ	1566
38.8.4	DMA レジスタの説明	1573
38.8.5	イーサネットレジスタマップ	1586
<b>39</b>	<b>HDMI-CEC コントローラ (HDMI-CEC)</b>	<b>1590</b>
39.1	概要	1590
39.2	HDMI-CEC コントローラの主な機能	1590
39.3	HDMI-CEC の機能詳細	1591
39.3.1	HDMI-CEC ピン	1591
39.3.2	メッセージの説明	1592
39.3.3	ビットタイミング	1593
39.4	アービトレーション	1594
39.4.1	SFT オプションビット	1595
39.5	エラー処理	1596
39.5.1	ビットエラー	1596
39.5.2	メッセージエラー	1596
39.5.3	ビット立ち上がりエラー (BRE)	1596
39.5.4	ショートビット周期エラー (SBPE)	1596
39.5.5	ロングビット周期エラー (LBPE)	1597
39.5.6	送信エラー検出 (TXERR)	1598
39.6	HDMI-CEC 割り込み	1600
39.7	HDMI-CEC レジスタ	1601
39.7.1	CEC 制御レジスタ (CEC_CR)	1601
39.7.2	CEC 設定レジスタ (CEC_CFGR)	1602
39.7.3	CEC Tx データレジスタ (CEC_TXDR)	1605
39.7.4	CEC Rx データレジスタ (CEC_RXDR)	1605
39.7.5	CEC 割り込みおよびステータスレジスタ (CEC_ISR)	1606
39.7.6	CEC 割り込み有効レジスタ (CEC_IER)	1608
39.7.7	HDMI-CEC レジスタマップ	1610
<b>40</b>	<b>デバッグサポート (DBG)</b>	<b>1611</b>
40.1	概要	1611
40.2	ARM® リファレンス資料	1612
40.3	SWJ デバッグポート (シリアルワイヤと JTAG)	1612
40.3.1	JTAG-DP または SW-DP の選択メカニズム	1613



40.4	ピン名とデバッグポートピン	1614
40.4.1	SWJ デバッグポートピン	1614
40.4.2	柔軟性の高い SWJ-DP ピンの割り当て	1614
40.4.3	JTAG ピンでの内部プルアップ/プルダウン	1615
40.4.4	シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法	1616
40.5	STM32F75xxx および STM32F74xxx JTAG デバッグポートの接続	1616
40.6	ID コードとロック機構	1618
40.6.1	MCU デバイス ID コード	1618
40.6.2	バウンダリスキャンデバッグポート	1618
40.6.3	FPU 搭載 Cortex <sup>®</sup> -M7 のデバッグポート	1618
40.6.4	FPU 搭載 Cortex <sup>®</sup> -M7 の JEDEC-106 ID コード	1619
40.7	JTAG デバッグポート	1619
40.8	SW デバッグポート	1621
40.8.1	SW プロトコルの概要	1621
40.8.2	SW プロトコルシーケンス	1621
40.8.3	SW-DP ステートマシン (リセット、アイドル状態、ID コード)	1623
40.8.4	DP と AP の読み出し/書き込みアクセス	1623
40.8.5	SW-DP レジスタ	1624
40.8.6	SW-AP レジスタ	1624
40.9	AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効	1625
40.10	コアデバッグ	1626
40.11	システムリセット中のデバッグホスト接続機能	1627
40.12	FPB (フラッシュパッチブレイクポイント)	1627
40.13	DWT (データウォッチポイントトリガ)	1628
40.14	ITM (計測トレースマクロセル)	1628
40.14.1	概要	1628
40.14.2	タイムスタンプパケット、同期およびオーバーフローパケット	1628
40.15	ETM (組み込みトレースマクロセル)	1630
40.15.1	概要	1630
40.15.2	信号プロトコル、パケットタイプ	1630
40.15.3	主な ETM レジスタ	1631
40.15.4	設定例	1631
40.16	MCU デバッグコンポーネント (DBGMCU)	1631
40.16.1	低電力モードのデバッグサポート	1631
40.16.2	タイマ、ウォッチドッグ、bxCAN、および I <sup>2</sup> C のデバッグサポート	1632

40.16.3	デバッグ MCU 設定レジスタ	1632
40.16.4	DBGMCU_CR レジスタ	1633
40.16.5	デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ)	1634
40.16.6	デバッグ MCU APB2 フリーズレジスタ (DBGMCU_APB2_FZ)	1636
40.17	ペリカン TPIU (トレースポートインタフェースユニット)	1637
40.17.1	概要	1637
40.17.2	TRACE ピンの割当て	1638
40.17.3	TPIU フォーマッタ	1640
40.17.4	TPIU フレーム同期パケット	1640
40.17.5	同期フレームパケットの送信	1640
40.17.6	同期モード	1641
40.17.7	非同期モード	1641
40.17.8	STM32F75xxx および STM32F74xxx 内の TRACECLKIN 接続	1641
40.17.9	TPIU レジスタ	1642
40.17.10	設定例	1642
40.18	DBG レジスタマップ	1643
<b>41</b>	<b>デバイス電子署名</b>	<b>1644</b>
41.1	ユニークデバイス ID レジスタ (96 ビット)	1644
41.2	フラッシュサイズ	1645
41.3	パッケージデータレジスタ	1645
<b>42</b>	<b>改版履歴</b>	<b>1651</b>

## 表の一覧

表 1.	STM32F75xxx および STM32F74xxx レジスタ境界アドレス	66
表 2.	ブートモード	71
表 3.	フラッシュメモリの構成	75
表 4.	CPU クロック (HCLK) 周波数によるウェイトステート数	75
表 5.	プログラム/消去の並列処理	78
表 6.	フラッシュ割り込みリクエスト	81
表 7.	オプションバイトの構成	81
表 8.	アクセスと読み出し保護レベル	86
表 9.	OTP エリアの構成	88
表 10.	フラッシュレジスタマップとリセット値	96
表 11.	電圧レギュレータ設定モードとデバイス動作モード	103
表 12.	低電力モードの概要	108
表 13.	すべてのモードでの機能	109
表 14.	Sleep-now への移行と終了	113
表 15.	Sleep-on-exit への移行と終了	114
表 16.	STOP モードの動作モード	115
表 17.	STOP モードへの移行と終了 (STM32F75xxx および STM32F74xxx)	117
表 18.	STANDBY モードへの移行と終了	119
表 19.	PWR - レジスタマップとリセット値	130
表 20.	RCC レジスタマップとリセット値	194
表 21.	ポートビット設定表	199
表 22.	GPIO レジスタマップとリセット値	212
表 23.	SYSCFG レジスタマップとリセット値	220
表 24.	DMA1 リクエストマッピング	224
表 25.	DMA2 リクエストマッピング	225
表 26.	転送元および転送先アドレス	226
表 27.	ダブルバッファモード (DBM=1) における転送元および転送先アドレスレジスタ	231
表 28.	パッキング/アンパッキングおよびエンディアン動作 (ビット PINC=MINC=1)	232
表 29.	PSIZE と MSIZE に対する NDT に関する制限事項	232
表 30.	FIFO 閾値設定	234
表 31.	実現可能な DMA 設定	238
表 32.	DMA 割り込みリクエスト	241
表 33.	DMA レジスタマップとリセット値	252
表 34.	入力でサポートされているカラーモード	259
表 35.	メモリ内のデータ順序	260
表 36.	アルファモード設定	261
表 37.	サポートされている CLUT カラーモード	262
表 38.	メモリ内の CLUT データ順序	262
表 39.	出力でサポートされているカラーモード	263
表 40.	メモリ内のデータ順序	263
表 41.	DMA2D 割り込みリクエスト	268
表 42.	DMA2D レジスタマップとリセット値	285
表 43.	STM32F75xxx および STM32F74xxx ベクタテーブル	287
表 44.	外部割り込み/イベントコントローラのレジスタマップとリセット値	298
表 45.	CRC レジスタマップとリセット値	304
表 46.	NOR/PSRAM バンク選択	310
表 47.	NOR/PSRAM 外部メモリアドレス	310
表 48.	NAND メモリマッピングおよびタイミングレジスタ	310
表 49.	NAND バンク選択	311

表 50.	SDRAM バンク選択	311
表 51.	SDRAM アドレスマッピング	311
表 52.	8 ビットデータバス幅での SDRAM アドレスマッピング	312
表 53.	16 ビットデータバス幅での SDRAM アドレスマッピング	313
表 54.	32 ビットデータバス幅での SDRAM アドレスマッピング	313
表 55.	プログラム可能な NOR/PSRAM のアクセスパラメータ	315
表 56.	非マルチプレクス I/O NOR フラッシュメモリ	316
表 57.	16 ビットマルチプレクス I/O NOR フラッシュメモリ	316
表 58.	非マルチプレクス I/O PSRAM/SRAM	317
表 59.	16 ビットマルチプレクス I/O PSRAM	317
表 60.	NOR フラッシュ/PSRAM: サポートされているメモリおよびトランザクションの例	318
表 61.	FMC_BCRx ビットフィールド	321
表 62.	FMC_BTRx ビットフィールド	321
表 63.	FMC_BCRx ビットフィールド	323
表 64.	FMC_BTRx ビットフィールド	323
表 65.	FMC_BWTRx ビットフィールド	324
表 66.	FMC_BCRx ビットフィールド	326
表 67.	FMC_BTRx ビットフィールド	326
表 68.	FMC_BWTRx ビットフィールド	327
表 69.	FMC_BCRx ビットフィールド	328
表 70.	FMC_BTRx ビットフィールド	329
表 71.	FMC_BWTRx ビットフィールド	329
表 72.	FMC_BCRx ビットフィールド	331
表 73.	FMC_BTRx ビットフィールド	331
表 74.	FMC_BWTRx ビットフィールド	332
表 75.	FMC_BCRx ビットフィールド	333
表 76.	FMC_BTRx ビットフィールド	334
表 77.	FMC_BCRx ビットフィールド	339
表 78.	FMC_BTRx ビットフィールド	340
表 79.	FMC_BCRx ビットフィールド	341
表 80.	FMC_BTRx ビットフィールド	342
表 81.	プログラ可能な NAND フラッシュのアクセスパラメータ	349
表 82.	8 ビット NAND 型フラッシュ	350
表 83.	16 ビット NAND 型フラッシュ	350
表 84.	サポートされるメモリおよびトランザクション	351
表 85.	ECC 結果関連ビット	359
表 86.	SDRAM 信号	360
表 87.	FMC レジスタマップ	378
表 88.	QUADSPI 割り込みリクエスト	395
表 89.	QUADSPI レジスタマップとリセット値	409
表 90.	ADC ピン	412
表 91.	アナログウォッチドッグチャネル選択	419
表 92.	トリガ極性の設定	423
表 93.	レギュラチャネルの外部トリガ	424
表 94.	インジェクトチャネルの外部トリガ	424
表 95.	ADC 割り込み	440
表 96.	ADC グローバルレジスタマップ	457
表 97.	ADC レジスタマップと各 ADC のリセット値	458
表 98.	ADC レジスタマップとリセット値 (共通 ADC レジスタ)	459
表 99.	DAC ピン	461
表 100.	外部トリガ	464
表 101.	DAC レジスタマップ	482



表 102.	DCMI ピン	483
表 103.	DCMI 信号	485
表 104.	32 ビットワード (8 ビット幅) でキャプチャされたデータバイトの配置	486
表 105.	32 ビットワード (10 ビット幅) でキャプチャされたデータバイトの配置	486
表 106.	32 ビットワード (12 ビット幅) でキャプチャされたデータバイトの配置	487
表 107.	32 ビットワード (14 ビット幅) でキャプチャされたデータバイトの配置	487
表 108.	モノクロプログレッシブビデオフォーマットでのデータ保存	493
表 109.	RGB プログレッシブビデオフォーマットでのデータ保存	494
表 110.	YCbCr プログレッシブビデオフォーマットでのデータ保存	494
表 111.	YCbCr プログレッシブビデオフォーマットでのデータ保存 - Y 抽出モード	495
表 112.	DCMI 割り込み	495
表 113.	DCMI レジスタマップとリセット値	508
表 114.	LCD-TFT ピンと信号インタフェース	511
表 115.	ピクセルデータマッピングとカラーフォーマット	515
表 116.	LTDC 割り込みリクエスト	518
表 117.	LTDC レジスタマップとリセット値	542
表 118.	RNG レジスタマップとリセット値	549
表 119.	各 128 ビットブロックの処理に必要なサイクル数	550
表 120.	データ型	568
表 121.	CRYP レジスタマップとリセット値	587
表 122.	HASH レジスタマップとリセット値	608
表 123.	タイマ出力と BRK/BRK2 入力の動作	651
表 124.	カウント方向とエンコーダ信号	658
表 125.	TIMx 内部トリガ接続	675
表 126.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能	689
表 127.	TIM1 レジスタマップとリセット値	701
表 128.	TIM8 レジスタマップとリセット値	703
表 129.	カウント方向とエンコーダ信号	739
表 130.	TIMx 内部トリガ接続	756
表 131.	標準 OCx チャンネルの出力制御ビット	768
表 132.	TIM2/TIM3/TIM4/TIM5 レジスタマップとリセット値	775
表 133.	TIMx 内部トリガ接続	803
表 134.	標準 OCx チャンネルの出力制御ビット	810
表 135.	TIM9/12 レジスタマップとリセット値	812
表 136.	標準 OCx チャンネルの出力制御ビット	819
表 137.	TIM10/11/13/14 レジスタマップとリセット値	822
表 138.	TIM6/TIM7 レジスタマップとリセット値	836
表 139.	STM32F75xxx および STM32F74xxx LPTIM の機能	837
表 140.	プリスケーラの分周比	840
表 141.	エンコーダのカウントシナリオ	845
表 142.	LPTIM 外部トリガ接続	852
表 143.	LPTIM レジスタマップとリセット値	857
表 144.	IWDG レジスタマップとリセット値	866
表 145.	WWDG レジスタマップとリセット値	873
表 146.	RTC ピン PC13 の設定	876
表 147.	RTC ピン PI8 の設定	877
表 148.	RTC ピン PC2 の設定	878
表 149.	RTC 機能 (モード共通)	878
表 150.	低電力モードが RTC に与える影響	890
表 151.	割り込み制御ビット	891
表 152.	RTC レジスタマップとリセット値	917
表 153.	STM32F75xxx および STM32F74xxxI2C の実装	920

表 154.	I2C-SMBUS 仕様のデータのセットアップおよびホールド時間	927
表 155.	I2C 設定表	931
表 156.	I2C-SMBUS 仕様のクロックタイミング	941
表 157.	SMBus タイムアウト仕様	954
表 158.	SMBUS の PEC 設定	955
表 159.	さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{TIMEOUT} = 25\text{ ms}$ )	957
表 160.	さまざまな I2CCLK 周波数での TIMEOUTB の設定例	957
表 161.	さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{IDLE} = 50\text{ ms}$ )	957
表 162.	低電力モード	967
表 163.	I2C 割り込みリクエスト	968
表 164.	I2C レジスタマップとリセット値	985
表 165.	STM32F75xxx および STM32F74xxx USART の機能	989
表 166.	サンプリングされたデータからのノイズ検出	1001
表 167.	$f_{CK} = 216\text{ MHz}$ でプログラムされたボーレートのエラー計算 (8 倍のオーバーサンプリング (OVER8 = 1) の場合)	1004
表 168.	$f_{CK} = 216\text{ MHz}$ でプログラムされたボーレートのエラー計算 (16 倍のオーバーサンプリング (OVER8 = 0) の場合)	1004
表 169.	BRR [3:0] = 0000 のときの USART レシーバの許容誤差	1005
表 170.	BRR [3:0] が 0000 でないときの USART レシーバの許容誤差	1005
表 171.	フレームフォーマット	1010
表 172.	低電力モードが USART に与える影響	1027
表 173.	USART 割り込みリクエスト	1028
表 174.	USART レジスタマップとリセット値	1050
表 175.	STM32F75xxx および STM32F74xxx SPI の実装	1052
表 176.	SPI 割り込みリクエスト	1077
表 177.	標準 8 MHz HSE を使用した場合のオーディオ周波数精度	1088
表 178.	I <sup>2</sup> S 割り込みリクエスト	1095
表 179.	SPI レジスタマップとリセット値	1107
表 180.	外部同期の選択	1113
表 181.	オーディオサンプリング周波数範囲の例	1119
表 182.	SOPD パターン	1125
表 183.	パリティビットの計算	1125
表 184.	オーディオサンプリング周波とシンボルレート (SHARK)	1126
表 185.	SAI 割り込みソース	1136
表 186.	SAI レジスタマップとリセット値	1152
表 187.	プリアンプルの遷移シーケンス	1161
表 188.	SPDIFRX_CLK 最小周波数とオーディオサンプリングレート	1172
表 189.	ビットフィールドの特性と SPDIFRX の状態	1174
表 190.	SPDIFRX インタフェースレジスタマップとリセット値	1189
表 191.	SDMMC I/O 定義	1193
表 192.	コマンドフォーマット	1198
表 193.	ショートレスポンスフォーマット	1199
表 194.	ロングレスポンスフォーマット	1199
表 195.	コマンドバスステータスフラグ	1199
表 196.	データトークンフォーマット	1202
表 197.	DPSM フラグ	1203
表 198.	送信 FIFO ステータスフラグ	1204
表 199.	受信 FIFO ステータスフラグ	1204
表 200.	カードステータス	1215
表 201.	SD ステータス	1218



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

表 202.	速度クラスコードフィールド	1219
表 203.	移動パフォーマンスフィールド	1220
表 204.	AU_SIZE フィールド	1220
表 205.	最大 AU サイズ	1220
表 206.	消去サイズフィールド	1221
表 207.	消去タイムアウトフィールド	1221
表 208.	消去オフセットフィールド	1221
表 209.	ブロック指向書き込みコマンド	1224
表 210.	ブロック指向書き込み保護コマンド	1224
表 211.	消去コマンド	1225
表 212.	I/O モードコマンド	1225
表 213.	カードのロック	1225
表 214.	アプリケーション固有コマンド	1226
表 215.	R1 レスポンス	1226
表 216.	R2 レスポンス	1227
表 217.	R3 レスポンス	1228
表 218.	R4 レスポンス	1228
表 219.	R4b レスポンス	1229
表 220.	R5 レスポンス	1229
表 221.	R6 レスポンス	1230
表 222.	レスポンスタイプと SDMMC_RESPx レジスタ	1236
表 223.	SDMMC レジスタマップ	1247
表 224.	送信メールボックスの配置	1264
表 225.	受信メールボックスの配置	1264
表 226.	bxCAN レジスタマップとリセット値	1291
表 227.	STM32F75xxx の USB_OTG 実装	1297
表 228.	コアグローバル制御およびステータスレジスタ (CSR)	1320
表 229.	ホストモード制御およびステータスレジスタ (CSR)	1321
表 230.	デバイスモード制御およびステータスレジスタ	1323
表 231.	データ FIFO (DFIFO) アクセスレジスタマップ	1325
表 232.	パワーおよびクロックゲーティング制御およびステータスレジスタ	1325
表 233.	ソフト切断のための最短時間	1376
表 234.	OTG_FS/OTG_HS レジスタマップとリセット値	1401
表 235.	オルタネート機能配置	1474
表 236.	管理フレームのフォーマット	1476
表 237.	クロック範囲	1478
表 238.	TX インタフェース信号のエンコーディング	1480
表 239.	RX インタフェース信号のエンコーディング	1480
表 240.	フレームのステータス	1496
表 241.	デスティネーションアドレスフィルタリング	1502
表 242.	ソースアドレスフィルタリング	1503
表 243.	受信ディスクリプタ 0-ビット 7、5、0 のエンコーディング (通常ディスクリプタフォーマットのみ、EDFE=0)	1534
表 244.	タイムスタンプスナップショットのレジスタビットに対する依存性	1568
表 245.	イーサネットレジスタマップとリセット値	1586
表 246.	HDMI ピン	1591
表 247.	エラー処理タイミングパラメータ	1597
表 248.	TXERR タイミングパラメータ	1599
表 249.	HDMI-CEC 割り込み	1600
表 250.	HDMI-CEC レジスタマップとリセット値	1610
表 251.	SWJ デバッグポートピン	1614
表 252.	柔軟性の高い SWJ-DP ピンの割り当て	1614

表 253.	JTAG デバッグポートのデータレジスタ	1619
表 254.	シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ	1621
表 255.	パケットリクエスト (8 ビット)	1622
表 256.	ACK 応答 (3 ビット)	1622
表 257.	データ転送 (33 ビット)	1622
表 258.	SW-DP レジスタ	1624
表 259.	FPU 搭載 Cortex <sup>®</sup> -M7 AHB-AP レジスタ	1625
表 260.	コアデバッグレジスタ	1626
表 261.	主な ITM レジスタ	1629
表 262.	非同期 TRACE ピンの割当て	1638
表 263.	同期 TRACE ピンの割当て	1638
表 264.	柔軟性の高い TRACE ピン割り当て	1639
表 265.	重要な TPIU レジスタ	1642
表 266.	DBG レジスタマップとリセット値	1643
表 267.	文書改版履歴	1651

## 図の一覧

図 1.	STM32F75xxx および STM32F74xxx デバイスのシステムアーキテクチャ	62
図 2.	システムアーキテクチャ内のフラッシュメモリインタフェース接続 (STM32F75xxx および STM32F74xxx)	73
図 3.	RDP レベル	86
図 4.	電源の概要	98
図 5.	VDDUSB を VDD 電源へ接続した場合	99
図 6.	VDDUSB を独立した外部電源に接続した場合	100
図 7.	バックアップドメイン	102
図 8.	パワーオンリセット/パワーダウンリセット波形	105
図 9.	BOR の閾値	106
図 10.	PVD の閾値	107
図 11.	簡略化されたリセット回路図	132
図 12.	クロックツリー	133
図 13.	HSE/LSE クロックソース	136
図 14.	入力キャプチャモードにおける TIM5 を使用した周波数測定	141
図 15.	入力キャプチャモードにおける TIM11 を使用した周波数測定	142
図 16.	I/O ポートビットの基本構成	198
図 17.	5 V トレラント I/O ポートビットの基本構成	198
図 18.	入力フローティング/プルアップ/プルダウン設定	203
図 19.	出力設定	204
図 20.	オルタネート機能設定	204
図 21.	高インピーダンスアナログ設定	205
図 22.	DMA ブロック図	223
図 23.	チャンネル選択	224
図 24.	“ペリフェラルからメモリ”モード	227
図 25.	“メモリからペリフェラル”モード	228
図 26.	メモリ間モード	229
図 27.	FIFO の構造	234
図 28.	DMA2D ブロック図	258
図 29.	外部割り込み/イベントコントローラのブロック図	292
図 30.	外部割り込み/イベント GPIO の配置	294
図 31.	CRC 計算ユニットのブロック図	300
図 32.	FMC ブロック図	306
図 33.	FMC メモリバンク	309
図 34.	モード 1 読み出しアクセス波形	320
図 35.	モード 1 書き込みアクセス波形	320
図 36.	モード A 読み出しアクセス波形	322
図 37.	モード A 書き込みアクセス波形	322
図 38.	モード 2 およびモード B 読み出しアクセス波形	324
図 39.	モード 2 書き込みアクセス波形	325
図 40.	モード B 書き込みアクセス波形	325
図 41.	モード C 読み出しアクセス波形	327
図 42.	モード C 書き込みアクセス波形	328
図 43.	モード D 読み出しアクセス波形	330
図 44.	モード D 書き込みアクセス波形	330
図 45.	マルチプレクス読み出しアクセス波形	332
図 46.	マルチプレクス書き込みアクセス波形	333
図 47.	読み出しアクセス中の非同期ウェイト波形	335
図 48.	書き込みアクセス中の非同期ウェイト波形	336

図 49.	待ち設定波形	338
図 50.	同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)	339
図 51.	同期マルチプレクス書き込みモード波形 - PSRAM (CRAM)	341
図 52.	共通メモリアクセスの NAND フラッシュコントローラ波形	352
図 53.	CE が無視されない NAND 型フラッシュへのアクセス	353
図 54.	バースト書き込み SDRAM アクセス波形	362
図 55.	バースト読み出し SDRAM アクセス	363
図 56.	RBURST ビットセット時の読み出しアクセスの論理図 (CAS=1、RPIPE=0)	364
図 57.	行境界を跨ぐ読み出しアクセス	366
図 58.	行境界を跨ぐ書き込みアクセス	367
図 59.	セルフリフレッシュモード	369
図 60.	パワーダウンモード	370
図 61.	QUADSPI ブロック 図 (デュアルフラッシュモードが無効のとき)	380
図 62.	QUADSPI ブロック 図 (デュアルフラッシュモードが有効のとき)	381
図 63.	クワッドモードにおける読み出しコマンドの例	382
図 64.	クワッドモードにおける DDR コマンドの例	385
図 65.	nCS - CKMODE = 0 (T = CLK 周期) のとき	393
図 66.	nCS - SDR モード (T = CLK 周期) において CKMODE = 1 のとき	394
図 67.	nCS - DDR モード (T = CLK 周期) において CKMODE = 1 のとき	394
図 68.	nCS - アポルトが発生し (T = CLK 周期)、CKMODE = 1 のとき	394
図 69.	1 つの ADC ブロック 図	411
図 70.	ADC1 接続性	413
図 71.	ADC2 接続性	414
図 72.	ADC3 接続性	415
図 73.	タイミング図	418
図 74.	アナログウォッチドッグによって保護される領域	418
図 75.	インジェクト変換の遅延時間	420
図 76.	12 ビットデータの右詰め	422
図 77.	12 ビットデータの左詰め	422
図 78.	6 ビットデータの左詰め	422
図 79.	マルチ ADC ブロック 図 <sup>(1)</sup>	427
図 80.	4 チャンネルのインジェクト同時モード: デュアル ADC モード	431
図 81.	4 チャンネルのインジェクト同時モード: トリプル ADC モード	431
図 82.	16 チャンネルのレギュラ同時モード: デュアル ADC モード	432
図 83.	16 チャンネルのレギュラ同時モード: トリプル ADC モード	432
図 84.	連続変換モードにおける 1 チャンネルのインタリーブモード: デュアル ADC モード	433
図 85.	連続変換モードにおける 1 チャンネルのインタリーブモード: トリプル ADC モード	434
図 86.	オルタネートトリガ: 各 ADC のインジェクトグループ	435
図 87.	オルタネートトリガ: 不連続モードにおける 4 つのインジェクトチャンネル (各 ADC)	436
図 88.	オルタネートトリガ: 各 ADC のインジェクトグループ	436
図 89.	オルタネート + レギュラ同時	437
図 90.	インジェクト変換中にトリガが発生した場合	438
図 91.	温度センサおよび VREFINT チャンネルのブロック 図	439
図 92.	DAC チャンネルブロック 図	461
図 93.	DAC 出力バッファ接続	462
図 94.	シングル DAC チャンネルモードのデータレジスタ	463
図 95.	デュアル DAC チャンネルモードのデータレジスタ	463
図 96.	トリガ無効 (TEN = 0) 時の変換タイミング 図	464
図 97.	DAC LFSR レジスタ計算アルゴリズム	465
図 98.	LFSR 波形生成による DAC 変換 (SW トリガ有効)	466
図 99.	DAC 三角波生成	466
図 100.	三角波生成による DAC 変換 (SW トリガ有効)	467

図 101.	DCMI のブロック図	484
図 102.	最上位ブロック図	485
図 103.	DCMI 信号の波形	486
図 104.	タイミング図	488
図 105.	スナップショットモードにおけるフレームキャプチャ波形	490
図 106.	連続グラブモードにおけるフレームキャプチャ波形	491
図 107.	クロップ後のウィンドウの座標とサイズ	491
図 108.	データキャプチャ波形	492
図 109.	ピクセルラスタースキャンの順序	493
図 110.	LTDC のブロック図	510
図 111.	LCD-TFT 同期タイミング	512
図 112.	レイヤウィンドウのプログラム可能なパラメータ	514
図 113.	2つのレイヤとバックグラウンドとのブレンディング	517
図 114.	割り込みイベント	518
図 115.	ブロック図	545
図 116.	ブロック図	552
図 117.	DES / トリプル DES-ECB モードの暗号化	554
図 118.	DES / トリプル DES-ECB モードの復号化	554
図 119.	DES / トリプル DES-CBC モードの暗号化	556
図 120.	DES / トリプル DES-CBC モードの復号化	557
図 121.	AES-ECB モードの暗号化	558
図 122.	AES-ECB モードの復号化	559
図 123.	AES-CBC モードの暗号化	560
図 124.	AES-CBC モードの復号化	561
図 125.	AES-CTR モードの暗号化	562
図 126.	AES-CTR モードの復号化	562
図 127.	カウンタモードでの初期カウンタブロック構造	563
図 128.	DATATYPE に応じた 64 ビットブロックの構成	569
図 129.	トリプル DES-CBC 暗号化における初期化ベクタの使用	571
図 130.	ブロック図	590
図 131.	ビット、バイト、およびハーフワードスワッピング	592
図 132.	HASH 割り込みマッピング図	597
図 133.	高機能制御タイマのブロック図	610
図 134.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	612
図 135.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	612
図 136.	内部クロック分周比が 1 の場合のカウンタのタイミング図	614
図 137.	内部クロック分周比が 2 の場合のカウンタのタイミング図	614
図 138.	内部クロック分周比が 4 の場合のカウンタのタイミング図	615
図 139.	内部クロック分周比が N の場合のカウンタのタイミング図	615
図 140.	ARPE = 0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図	616
図 141.	ARPE = 1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図	616
図 142.	内部クロック分周比が 1 の場合のカウンタのタイミング図	618
図 143.	内部クロック分周比が 2 の場合のカウンタのタイミング図	618
図 144.	内部クロック分周比が 4 の場合のカウンタのタイミング図	619
図 145.	内部クロック分周比が N の場合のカウンタのタイミング図	619
図 146.	繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図 (不使用)	620
図 147.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	621
図 148.	内部クロック分周比が 2 の場合のカウンタのタイミング図	622
図 149.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	622

図 150.	内部クロック分周比が N の場合のカウンタのタイミング図	623
図 151.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図	623
図 152.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	624
図 153.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	625
図 154.	外部トリガ入力ブロック	626
図 155.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	627
図 156.	TI2 外部クロックの接続例	628
図 157.	外部クロックモード 1 の制御回路	629
図 158.	外部トリガ入力ブロック	629
図 159.	外部クロックモード 2 の制御回路	630
図 160.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	631
図 161.	キャプチャ/比較チャンネル 1 メイン回路	632
図 162.	キャプチャ/比較チャンネル (チャンネル 1、同じくチャンネル 2 および 3) の出力ステージ	632
図 163.	キャプチャ/比較チャンネル (チャンネル 4) の出力ステージ	633
図 164.	キャプチャ/比較チャンネル (チャンネル 5、同じくチャンネル 6) の出力ステージ	633
図 165.	PWM 入力モードタイミング	635
図 166.	出力比較モード、OC1 の反転	637
図 167.	エッジアライン PWM 波形 (ARR=8)	639
図 168.	センターアライン PWM 波形 (ARR=8)	640
図 169.	50% デューティサイクルの 2 位相シフトされた PWM 信号の生成	641
図 170.	チャンネル 1 および 3 における組み合わせ PWM モード	643
図 171.	周期ごとの複数トリガパルスを持つ組み合わせ 3 相 PWM 信号	644
図 172.	デッドタイム挿入のある相補出力	645
図 173.	負のパルスより長い遅延があるときのデッドタイムの波形	645
図 174.	正のパルスより長い遅延があるときのデッドタイムの波形	646
図 175.	ブレーク および ブレーク 2 回路の概要	648
図 176.	BRK (OSSI = 1) でのブレークイベントに対するさまざまな出力の動作	650
図 177.	BRK および BRK2 ピンのアサート後の PWM 出力状態 (OSSI=1)	651
図 178.	BRK アサート後の PWM 出力状態 (OSSI=0)	652
図 179.	TIMx OCxREF のクリア	653
図 180.	6 ステップ生成 COM の例 (OSSR=1)	654
図 181.	ワンパルスモードの例	655
図 182.	再トリガ可能なワンパルスモード	657
図 183.	エンコーダインタフェースモードにおけるカウンタの動作例	658
図 184.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	659
図 185.	3 つの信号上のエッジ間の時間間隔の測定	660
図 186.	ホールセンサインタフェースの例	662
図 187.	リセットモードの制御回路	663
図 188.	ゲートモードの制御回路	664
図 189.	トリガモードの制御回路	665
図 190.	外部クロックモード 2 + トリガモードの制御回路	666
図 191.	汎用タイマのブロック図	707
図 192.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	709
図 193.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	709
図 194.	内部クロック分周比が 1 の場合のカウンタのタイミング図	710
図 195.	内部クロック分周比が 2 の場合のカウンタのタイミング図	711
図 196.	内部クロック分周比が 4 の場合のカウンタのタイミング図	711
図 197.	内部クロック分周比が N の場合のカウンタのタイミング図	712
図 198.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図	712
図 199.	ARPE=1 (TIMx_ARR はプリロードされる) のときの更新イベント時のカウンタのタイミング図	713

図 200.	内部クロック分周比が 1 の場合のカウンタのタイミング図	714
図 201.	内部クロック分周比が 2 の場合のカウンタのタイミング図	715
図 202.	内部クロック分周比が 4 の場合のカウンタのタイミング図	715
図 203.	内部クロック分周比が N の場合のカウンタのタイミング図	716
図 204.	繰り返しカウンタが使用されていないときの更新イベント時のカウンタのタイミング図	716
図 205.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	718
図 206.	内部クロック分周比が 2 の場合のカウンタのタイミング図	718
図 207.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	719
図 208.	内部クロック分周比が N の場合のカウンタのタイミング図	719
図 209.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図	720
図 210.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	721
図 211.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	722
図 212.	TI2 外部クロックの接続例	722
図 213.	外部クロックモード 1 の制御回路	723
図 214.	外部トリガ入力ブロック	724
図 215.	外部クロックモード 2 の制御回路	725
図 216.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	726
図 217.	キャプチャ/比較チャンネル 1 メイン回路	726
図 218.	キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ	727
図 219.	PWM 入力モードタイミング	729
図 220.	出力比較モード、OC1 の反転	730
図 221.	エッジアライン PWM 波形 (ARR=8)	732
図 222.	センターアライン PWM 波形 (ARR=8)	733
図 223.	50% デューティサイクルの 2 位相シフトされた PWM 信号の生成	734
図 224.	チャンネル 1 および 3 における組み合わせ PWM モード	735
図 225.	TIMx_OCxREF のクリア	736
図 226.	ワンパルスモードの例	737
図 227.	エンコーダインタフェースモードにおけるカウンタの動作例	739
図 228.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	740
図 229.	リセットモードの制御回路	741
図 230.	ゲートモードの制御回路	742
図 231.	トリガモードの制御回路	743
図 232.	外部クロックモード 2 + トリガモードの制御回路	744
図 233.	マスタ/スレーブタイマの例	745
図 234.	TIM3 の OC1REF による TIM のゲート操作	746
図 235.	TIM3 の有効化による TIM のゲート操作	747
図 236.	TIM3 の更新による TIM のトリガ	747
図 237.	TIM3 の有効化による TIM のトリガ	748
図 238.	TIM3 と TIM2 を TIM3 TI1 入力でトリガします	749
図 239.	汎用タイマのブロック図 (TIM9 および TIM12)	779
図 240.	汎用タイマのブロック図 (TIM10/11/13/14)	780
図 241.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	782
図 242.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	782
図 243.	内部クロック分周比が 1 の場合のカウンタのタイミング図	783
図 244.	内部クロック分周比が 2 の場合のカウンタのタイミング図	784
図 245.	内部クロック分周比が 4 の場合のカウンタのタイミング図	784
図 246.	内部クロック分周比が N の場合のカウンタのタイミング図	785
図 247.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図	785
図 248.	ARPE = 1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図	786
図 249.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	787

図 250.	T12 外部クロックの接続例	787
図 251.	外部クロックモード 1 の制御回路	788
図 252.	キャプチャ／比較チャンネル (例: チャンネル 1 入力ステージ)	789
図 253.	キャプチャ／比較チャンネル 1 メイン回路	789
図 254.	キャプチャ／比較チャンネル (チャンネル 1) の出力ステージ	790
図 255.	PWM 入力モードタイミング	792
図 256.	出力比較モード、OC1 のトグル	793
図 257.	エッジアライン PWM 波形 (ARR=8)	795
図 258.	ワンパルスモードの例	796
図 259.	リセットモードの制御回路	798
図 260.	ゲートモードの制御回路	799
図 261.	トリガモードの制御回路	799
図 262.	基本タイマブロック図	824
図 263.	プリスケーラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	826
図 264.	プリスケーラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	826
図 265.	内部クロック分周比が 1 の場合のカウンタのタイミング図	827
図 266.	内部クロック分周比が 2 の場合のカウンタのタイミング図	828
図 267.	内部クロック分周比が 4 の場合のカウンタのタイミング図	828
図 268.	内部クロック分周比が N の場合のカウンタのタイミング図	829
図 269.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	829
図 270.	ARPE = 1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタの タイミング図	830
図 271.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	831
図 272.	低電力タイマのブロック図	838
図 273.	グリッチフィルタのタイミング図	839
図 274.	LPTIM 出力波形、シングルカウントモードの設定	841
図 275.	LPTIM 出力波形、シングルカウントモードの設定およびセットワンスモードの アクティブ化 (WAVE ビットをセット)	841
図 276.	LPTIM 出力波形、連続カウントモードの設定	842
図 277.	波形生成	843
図 278.	エンコーダモードのカウントシーケンス	846
図 279.	独立型ウォッチドッグのブロック図	858
図 280.	ウォッチドッグのブロック図	868
図 281.	ウィンドウ型ウォッチドッグのタイミング図	869
図 282.	RTC ブロック図	875
図 283.	I2C ブロック図	921
図 284.	I2C バスプロトコル	923
図 285.	セットアップおよびホールドタイミング	925
図 286.	I2C 初期化フローチャート	928
図 287.	データ受信	929
図 288.	データ送信	930
図 289.	スレーブ初期化フローチャート	933
図 290.	I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0)	935
図 291.	I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1)	936
図 292.	I2C スレーブトランスミッタの転送バス図	937
図 293.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0)	938
図 294.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1)	939
図 295.	I2C スレーブレシーバの転送バス図	939
図 296.	マスタクロック生成	941
図 297.	マスタ初期化フローチャート	943
図 298.	HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス	943

図 299.	HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス	944
図 300.	N≤255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート	945
図 301.	N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート	946
図 302.	I2C マスタトランスミッタの転送バス図	947
図 303.	N≤255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート	949
図 304.	N>255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート	950
図 305.	I2C マスタレシーバの転送バス図	951
図 306.	t <sub>LOW:SEXT</sub> 、t <sub>LOW:MEXT</sub> のタイムアウト間隔	954
図 307.	N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート	958
図 308.	SMBus スレーブトランスミッタの転送バス図 (SBC=1)	958
図 309.	N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート	960
図 310.	SMBus スレーブレシーバのバス転送図 (SBC=1)	961
図 311.	SMBus マスタトランスミッタのバス転送図	962
図 312.	SMBus マスタレシーバのバス転送図	964
図 313.	I2C 割り込みマッピング図	969
図 314.	USART のブロック図	991
図 315.	ワード長のプログラミング	993
図 316.	設定可能なストップビット	994
図 317.	送信時の TC/TXE の動作	996
図 318.	16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出	997
図 319.	データサンプリング (16 倍のオーバーサンプリング)	1000
図 320.	データサンプリング (8 倍のオーバーサンプリング)	1001
図 321.	アイドルライン検出を使用したミュートモード	1008
図 322.	アドレスマーク検出を使用したミュートモード	1009
図 323.	LIN モードでのブレーク検出 (11 ビットブレーク長、LBDL=1)	1012
図 324.	LIN モードでのブレーク検出とフレーミングエラー検出	1013
図 325.	USART の同期送信の例	1014
図 326.	USART データクロックタイミング図 (M ビット = 00)	1014
図 327.	USART データクロックタイミング図 (M ビット = 01)	1015
図 328.	RX データセットアップ/ホールド時間	1015
図 329.	ISO 7816-3 非同期プロトコル	1017
図 330.	ストップビット 1.5 個を使用したパリティエラー検出	1018
図 331.	IrDA SIR ENDEC- ブロック図	1022
図 332.	IrDA データ変調 (3/16) - 通常モード	1022
図 333.	DMA を使用した送信	1024
図 334.	DMA を使用した受信	1025
図 335.	2 つの USART 間のハードウェアフロー制御	1025
図 336.	RS232 RTS フロー制御	1026
図 337.	RS232 CTS フロー制御	1027
図 338.	USART 割り込みマッピング図	1029
図 339.	SPI ブロック図	1053
図 340.	全二重シングルマスタ/シングルスレーブアプリケーション	1054
図 341.	半二重シングルマスタ/シングルスレーブアプリケーション	1055
図 342.	単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ)	1056
図 343.	マスタと 3 つの独立したスレーブ	1057
図 344.	ハードウェア/ソフトウェアスレーブ選択管理	1058
図 345.	データクロックのタイミング図	1060
図 346.	データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置	1061
図 347.	FIFO での送受信用のデータのバッキング	1065
図 348.	マスタの全二重通信	1068
図 349.	スレーブの全二重通信	1069

図 350.	CRC のあるマスタの全二重通信	1070
図 351.	パックされたモードでのマスタの全二重通信	1071
図 352.	モトローラ SPI マスタモードでの NSSP パルス生成	1074
図 353.	TI モードでの転送	1075
図 354.	I <sup>2</sup> S ブロック図	1078
図 355.	I <sup>2</sup> S フィリップスプロトコルの波形 (16/32 ビットフル精度)	1080
図 356.	I <sup>2</sup> S フィリップス規格の波形 (24 ビットフレーム)	1080
図 357.	0x8EAA33 の送信	1081
図 358.	0x8EAA33 の受信	1081
図 359.	I <sup>2</sup> S フィリップス規格 (32 ビットパケットフレームに拡張された 16 ビット)	1081
図 360.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例	1081
図 361.	MSB 詰め 16 ビットまたは 32 ビット長フル精度	1082
図 362.	MSB 詰め 24 ビットフレーム長	1082
図 363.	32 ビットパケットフレームに拡張された MSB 詰め 16 ビット	1082
図 364.	LSB 詰め 16 ビットまたは 32 ビットフル精度	1083
図 365.	LSB 詰め 24 ビットフレーム長	1083
図 366.	0x3478AE を送信するために必要な動作	1083
図 367.	0x3478AE の受信に必要な動作	1084
図 368.	32 ビットパケットフレームに拡張された LSB 詰め 16 ビット	1084
図 369.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例	1084
図 370.	PCM 規格の波形 (16 ビット)	1085
図 371.	PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)	1085
図 372.	マスタモードでの開始シーケンス	1086
図 373.	オーディオサンプリング周波数の定義	1087
図 374.	I <sup>2</sup> S クロックジェネレータのアーキテクチャ	1087
図 375.	機能ブロック図	1110
図 376.	オーディオフレーム	1113
図 377.	FS の機能は、フレーム開始 + チャネルサイド識別 (FSDEF = TRIS = 1)	1115
図 378.	FS の機能はフレーム開始 (FSDEF = 0)	1116
図 379.	スロットサイズ設定 (SAI_xSLOTR で FBOFF = 0 の場合)	1117
図 380.	先頭ビットオフセット	1117
図 381.	オーディオブロッククロックジェネレータの概要	1118
図 382.	AC'97 オーディオフレーム	1122
図 383.	2 つ以上の内蔵 SAI を搭載したデバイス (3 個の外部 AC'97 デコーダ) の AC'97 設定の 代表例	1123
図 384.	SPDIF フォーマット	1124
図 385.	SAI_xDR レジスタの順序	1125
図 386.	SAI オーディオブロックのデータコンパニングハードウェア	1128
図 387.	非アクティブスロットにおける SD 出力ラインのトライステート化	1130
図 388.	I <sup>2</sup> S のようなプロトコルにおける出力データラインのトライステート化	1131
図 389.	オーバーラン検出エラー	1132
図 390.	FIFO アンダーランイベント	1132
図 391.	SPDIFRX ブロック図	1155
図 392.	S/PDIF サブフレームフォーマット	1156
図 393.	S/PDIF ブロックフォーマット	1156
図 394.	S/PDIF プリアンブル	1157
図 395.	チャンネルのコード化の例	1158
図 396.	SPDIFRX_DC デコーダ	1159
図 397.	ノイズフィルタリングおよびエッジ検出	1159
図 398.	閾値	1160
図 399.	同期のフローチャート	1163
図 400.	同期プロセスのスケジューリング	1164



図 401.	SPDIFRX の状態	1165
図 402.	SPDIFRX_DR レジスタ フォーマット	1167
図 403.	チャンネル/ユーザデータフォーマット	1168
図 404.	S/SPDIF オーバーランエラー (RXSTEO = 0 の場合)	1170
図 405.	S/SPDIF オーバーランエラー (RXSTEO = 1 の場合)	1171
図 406.	SPDIFRX インタフェース割り込みマッピング図	1173
図 407.	「no response」と「no data」操作	1191
図 408.	(複数) ブロック読み出し操作	1191
図 409.	(複数) ブロック書き込み操作	1191
図 410.	連続読み出し操作	1192
図 411.	連続書き込み操作	1192
図 412.	SDMMC ブロック図	1192
図 413.	SDMMC アダプタ	1194
図 414.	制御ユニット	1195
図 415.	SDMMC_CK クロック反位相 (BYPASS = 0)	1196
図 416.	SDMMC アダプタコマンドパス	1196
図 417.	コマンドパスステートマシン (SDMMC)	1197
図 418.	SDMMC コマンド転送	1198
図 419.	データパス	1200
図 420.	データパスステートマシン (DPSM)	1201
図 421.	CAN ネットワークのトポロジ	1250
図 422.	デュアル CAN ブロック図	1251
図 423.	bxCAN 動作モード	1253
図 424.	bxCAN (サイレントモード)	1254
図 425.	bxCAN (ループバックモード)	1254
図 426.	bxCAN (組み合わせモード)	1255
図 427.	送信メールボックスの状態	1257
図 428.	受信 FIFO の状態	1258
図 429.	フィルタバンクのスケール設定 - レジスタ構造	1261
図 430.	フィルタ番号の例	1262
図 431.	フィルタのメカニズム - 例	1263
図 432.	CAN エラー状態図	1264
図 433.	ビットタイミング	1266
図 434.	CAN フレーム	1267
図 435.	イベントフラグと割り込み生成	1268
図 436.	CAN メールボックスレジスタ	1280
図 437.	OTG フルスピードブロック図	1298
図 438.	OTG ハイスピードブロック図	1299
図 439.	OTG_FS A-B デバイス接続	1301
図 440.	USB_FS ペリフェラル専用接続	1303
図 441.	USB_FS ホスト専用接続	1308
図 442.	SOF 接続性 (TIM および ITR1 接続への SOF トリガ出力)	1312
図 443.	OTG_HFIR の動的更新	1314
図 444.	デバイスモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング	1315
図 445.	ホストモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング	1316
図 446.	割り込みの階層	1319
図 447.	送信 FIFO 書き込みタスク	1415
図 448.	受信 FIFO 読み出しタスク	1416
図 449.	通常のバルク/制御 OUT/SETUP	1417
図 450.	バルク/制御 IN トランザクション	1421
図 451.	通常割り込み OUT	1424
図 452.	通常割り込み IN	1428

図 453.	アイソクロナス OUT トランザクション	1430
図 454.	アイソクロナス IN トランザクション	1433
図 455.	通常のバルク／制御 OUT/SETUP トランザクション - DMA	1435
図 456.	通常のバルク／制御 IN トランザクション - DMA	1437
図 457.	通常の割り込み OUT トランザクション - DMA モード	1439
図 458.	通常の割り込み IN トランザクション - DMA モード	1440
図 459.	通常のアイソクロナス OUT トランザクション - DMA モード	1441
図 460.	通常のアイソクロナス IN トランザクション - DMA モード	1442
図 461.	受信 FIFO パケット読み出し	1448
図 462.	SETUP パケットの処理	1450
図 463.	バルク OUT トランザクション	1456
図 464.	TRDT 最大タイミングのケース	1465
図 465.	A デバイスの SRP	1466
図 466.	B デバイスの SRP	1467
図 467.	A デバイスの HNP	1468
図 468.	B デバイスの HNP	1469
図 469.	ETH のブロック図	1475
図 470.	SMI インタフェース信号	1476
図 471.	MDIO のタイミングとフレーム構造 - 書き込みサイクル	1477
図 472.	MDIO のタイミングとフレーム構造 - 読み出しサイクル	1478
図 473.	メディア独立インタフェース信号	1479
図 474.	MII クロックソース	1481
図 475.	減少メディア独立インタフェース信号	1481
図 476.	RMII クロックソース	1482
図 477.	クロック構成	1482
図 478.	アドレスフィールドのフォーマット	1484
図 479.	MAC フレームフォーマット	1486
図 480.	タグ付き MAC フレームのフォーマット	1486
図 481.	送信ビットの順序	1493
図 482.	衝突がない場合の送信	1493
図 483.	衝突がある場合の送信	1494
図 484.	MMI および RMII モードでのフレーム送信	1494
図 485.	受信ビットの順序	1498
図 486.	エラーなしでの受信	1498
図 487.	エラーありでの受信	1499
図 488.	偽キャリア表示付きの受信	1499
図 489.	MAC コア割り込みマスク構成	1500
図 490.	ウェイクアップフレームフィルタレジスタ	1505
図 491.	ネットワーク時間の同期	1508
図 492.	密補正法を使用したシステム時間の更新	1510
図 493.	TIM2 ITR1 への PTP トリガ出力	1512
図 494.	PPS 出力	1513
図 495.	ディスクリプタのリングおよびチェーン構造	1514
図 496.	デフォルトモードでの TxDMA の動作	1518
図 497.	OSF モードでの TxDMA の動作	1520
図 498.	通常送信ディスクリプタ	1521
図 499.	拡張送信ディスクリプタ	1527
図 500.	受信 DMA の動作	1529
図 501.	通常 Rx DMA ディスクリプタの構造	1531
図 502.	IEEE1588 タイムスタンプが有効な拡張受信ディスクリプタフィールドのフォーマット	1537
図 503.	割り込み方式	1540

図 504.	イーサネット MAC リモートウェイクアップフレームフィルタレジスタ (ETH_MACRWUFR) .....	1551
図 505.	ブロック図 .....	1591
図 506.	メッセージの構造 .....	1592
図 507.	ブロック .....	1592
図 508.	ビットタイミング .....	1593
図 509.	信号フリータイム .....	1594
図 510.	アービトレーションフェーズ .....	1594
図 511.	3つの公称ビット周期の SFT .....	1594
図 512.	エラービットタイミング .....	1596
図 513.	エラー処理 .....	1597
図 514.	TXERR 検出 .....	1598
図 515.	STM32 MCU と FPU 搭載 Cortex <sup>®</sup> -M7 レベルのデバッグサポートのブロック図 .....	1611
図 516.	SWJ デバッグポート .....	1613
図 517.	JTAG デバッグポートの接続 .....	1617
図 518.	TPIU ブロック図 .....	1637

## 1 このマニュアルにおける表記の規則

### 1.1 レジスタに関する略記

レジスタの説明では、次の略記が使用されます。

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

## 1.2 用語

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

- CPU コアは、2つのデバッグポートを内蔵しています。
  - JTAG デバッグポート (**JTAG-DP**) には、Joint Test Action Group (JTAG) のプロトコルに基づいた 5 ピンの標準インタフェースを搭載しています。
  - SWD デバッグポート (**SWD-DP**) には、Serial Wire Debug (SWD) プロトコルに基づいた 2 ピン (クロックとデータ) のインタフェースを搭載しています。  
JTAG および SWD のプロトコルについては、Cortex<sup>®</sup>-M7 テクニカルリファレンスマニュアルを参照してください。
- **ワード** : 32 ビット長のデータ。
- **ハーフワード** : 16 ビット長のデータ。
- **バイト** : 8 ビット長のデータ。
- **ダブルワード** : 64 ビット長のデータ。
- **IAP (アプリケーション内プログラミング)** : IAP は、ユーザプログラム実行中にマイクロコントローラのフラッシュメモリを再プログラムする機能です。
- **ICP (インサーキットプログラミング)** : ICP は、ユーザアプリケーションボードにデバイスが搭載された状態で、JTAG プロトコルや SWD プロトコルまたはブートローダを用いて、マイクロコントローラのフラッシュメモリをプログラムする機能です。
- **オプションバイト** : フラッシュメモリに格納された製品設定ビット。
- **AHB** : アドバンスドハイパフォーマンスバス。
- **AHBS** : AHB スレーブバス。
- **AXIM** : AXI マスタバス。
- **ITCM** : 命令密結合メモリ。
- **DTCM** : データ密結合メモリ。
- **CPU** : FPU コア搭載の Cortex<sup>®</sup>-M7 を参照。

## 1.3 使用可能なペリフェラル

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

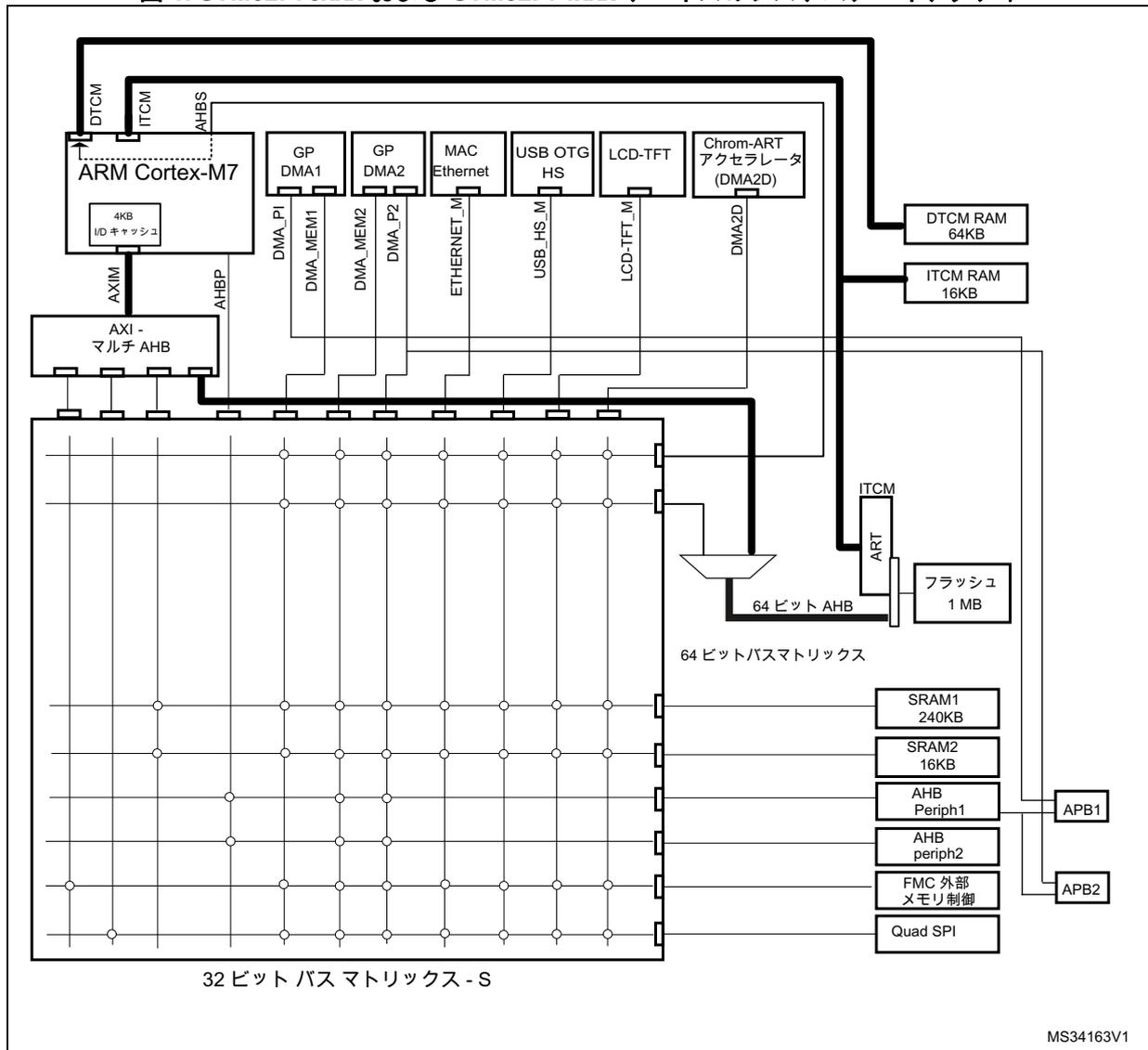
## 2 システムおよびメモリの概要

### 2.1 システムアーキテクチャ

メインシステムのアーキテクチャは2つのサブシステムに基づいています。

- AXI4 プロトコルを AHB-Lite プロトコルに変換する AXI からマルチ AHB へのブリッジ
  - 内蔵フラッシュに接続されている 1x AXI から 64 ビット AHB へのブリッジ
  - AHB バスマトリックスに接続されている 3x AXI から 32 ビット AHB へのブリッジ
- マルチ AHB バスマトリックス

図 1. STM32F75xxx および STM32F74xxx デバイスのシステムアーキテクチャ



マルチ AHB バスマトリックスはすべてのマスタおよびスレーブを相互接続し、以下のもので構成されています。

- 32 ビットマルチ AHB バスマトリックス
- 64 ビットマルチ AHB バスマトリックス: AXI-AHB ブリッジによる CPU からの 64 ビット AHB バスおよび GP DMA および 64 ビットに拡大されたペリフェラル DMA からの 32 ビット AHB バスを内部フラッシュへ相互接続します。

マルチ AHB バスマトリックスは以下のものを相互接続します。

- 12 個のバスマスタ：
  - 3x32 ビット AHB バス Cortex<sup>®</sup>-M7 AXI マスタバス 64 ビット、AXI-AHB ブリッジによって分割された 4 個のマスタ
  - 内蔵フラッシュに接続されている 1 個の 64 ビット AHB バス
  - Cortex<sup>®</sup>-M7 AHB ペリフェラルバス
  - DMA1 メモリバス
  - DMA2 メモリバス
  - DMA2 ペリフェラルバス
  - Ethernet DMA バス
  - USB OTG HS DMA バス
  - LCD コントローラ DMA バス
  - Chrom-Art Accelerator™ (DMA2D) メモリバス
- 8 個のバススレーブ：
  - AHB バス上の内蔵フラッシュ（フラッシュ読み出し／書き込みアクセス用、コード実行およびデータアクセス用）
  - DTCM RAM 上の DMA データ転送専用 Cortex<sup>®</sup>-M7 AHBS スレーブインタフェース
  - メイン内蔵メモリ SRAM1 (240 KB)
  - 補助内蔵メモリ SRAM2 (16 KB)
  - AHB-APBブリッジおよび APB ペリフェラルを含む AHB1 ペリフェラル
  - AHB-APBブリッジおよび APB ペリフェラルを含む AHB2 ペリフェラル
  - FMC
  - Quad SPI

## 2.1.1 マルチ AHB バスマトリックス

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

複数の高速ペリフェラルが同時動作中であっても、バスマトリックスによって、マスタからスレーブへのアクセスが可能になるとともに、同時アクセスや効率的な動作を行うことが可能になっています。DTCM および ITCM RAM（密に結合されたメモリ）は、バスマトリックスの一部ではありません。データ TCM RAMは、GP-DMA およびペリフェラル DMA によって、CPU の特定の AHB スレーブバス経由でアクセスできます。

命令 TCM RAM は CPU 専用で、0ウエイトステートのCPU クロックスピードでアクセスされます。アーキテクチャを [図 1](#) に示します。

## 2.1.2 AHB-APB ブリッジ (APB)

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

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

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

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

## 2.1.3 CPU AXIM バス

このバスは、FPU 搭載 Cortex<sup>®</sup>-M7 コアの命令およびデータバスを、AXI-AHB ブリッジ経由でマルチ AHB バスマトリックスに接続します。4 通りの AXI バスアクセスがあります。

- CPU AXI バスアクセス 1：この AXI バスの対象は、コードまたはデータを含む外部メモリ FMC です。アドレス 0x8000 0000 から 0x8FFF FFFF までに配置される NAND バンクについては、この空間の MPU メモリ属性をソフトウェアによってデバイスに再設定する必要があります。
- CPU AXI バスアクセス 2：この AXI バスの対象は、コードまたはデータを含む外部メモリ Quad SPI です。
- CPU AXI バスアクセス 3：この AXI バスの対象は、コードまたはデータを含む内部 SRAM (SRAM1 および SRAM2) です。
- CPU AXI バスアクセス 4：この AXI バスの対象は、AXI インタフェースに配置されたコードまたはデータを含む内蔵フラッシュです。

## 2.1.4 ITCM バス

このバスは、Cortex<sup>®</sup>-M7 が、ITCM インタフェースに配置された内蔵フラッシュの命令フェッチとデータアクセス、および ITCM RAM の命令フェッチのみを行うために使用されます。

## 2.1.5 DTCM バス

このバスは、Cortex<sup>®</sup>-M7 が、DTCM RAM のデータアクセスを行うために使用されます。また、命令フェッチにも使用できます。

## 2.1.6 CPU AHBS バス

このバスによって、Cortex<sup>®</sup>-M7 の AHB スレーブバスがバスマトリックスに接続されます。このバスは、DMA およびペリフェラル DMA が、DTCM RAM のデータ転送のみを行うために使用されます。

ITCM バスは AHBS ではアクセスできません。従って、DMA と ITCM RAM 間のデータ転送はサポートされません。DMA の ITCM インタフェース上のフラッシュに対するデータ転送は、強制的に AHB バス経由で行われます。

## 2.1.7 AHB ペリフェラルバス

このバスによって、Cortex<sup>®</sup>-M7 の AHB ペリフェラルがバスマトリックスに接続されます。このバスは、コアが、ペリフェラルへのすべてのデータアクセスを行うために使用されます。

このバスの対象は、APB ペリフェラルを含むAHB1ペリフェラル、および AHB2 ペリフェラルです。

## 2.1.8 DMA メモリバス

このバスによって、DMA メモリバスマスタインタフェースが BusMatrix に接続されます。このバスは、DMA がメモリとの間の転送に使用します。このバスの対象は、次のデータメモリです。内部 SRAM1、SRAM2、および DTCM (Cortex<sup>®</sup>-M7 の AHBS バス経由)、内部フラッシュメモリ、および FMC またはQuad SPI を経由した外部メモリ。

## 2.1.9 DMA ペリフェラルバス

このバスによって、DMA ペリフェラルマスタバスインタフェースが BusMatrix に接続されます。このバスは、DMA が AHB ペリフェラルへのアクセス、またはメモリ間転送を行うために使用します。このバスの対象は、AHB および APB ペリフェラル、および次のデータメモリです。内部 SRAM1、SRAM2、および DTCM (Cortex<sup>®</sup>-M7 の AHBS バス経由)、内部フラッシュメモリ、および FMC またはQuad SPI を経由した外部メモリ。

## 2.1.10 Ethernet DMA バス

このバスによって、Ethernet DMA マスタインタフェースが BusMatrix に接続されます。このバスは、Ethernet DMA がメモリに対してデータを読み書きするために使用します。このバスの対象は、次のデータメモリです。内部 SRAM1、SRAM2、および DTCM (Cortex<sup>®</sup>-M7 の AHBS バス経由)、内部フラッシュメモリ、および FMC またはQuad SPI を経由した外部メモリ。

## 2.1.11 USB OTG HS DMA バス

このバスによって、USB OTG HS DMA マスタインタフェースが BusMatrix に接続されます。このバスは、USB OTG DMA がメモリに対してデータを読み書きするために使用します。このバスの対象は、次のデータメモリです。内部 SRAM1、SRAM2、および DTCM (Cortex<sup>®</sup>-M7 の AHBS バス経由)、内部フラッシュメモリ、および FMC またはQuad SPI を経由した外部メモリ。

## 2.1.12 LCD-TFT コントローラ DMA バス

このバスによって、LCD コントローラ DMA マスタインタフェースが BusMatrix に接続されます。このバスは、LCD-TFT DMA がメモリに対してデータを読み書きするために使用します。このバスの対象は、次のデータメモリです。内部 SRAM1、SRAM2、および DTCM (Cortex<sup>®</sup>-M7 の AHBS バス経由)、FMC またはQuad SPI を経由した外部メモリ、および内部フラッシュメモリ。

## 2.1.13 DMA2D バス

このバスによって、DMA2D マスタインタフェースが BusMatrix に接続されます。このバスは、DMA2D グラフィックアクセラレータがメモリに対してデータを読み書きするために使用します。このバスの対象は、次のデータメモリです。内部 SRAM1、SRAM2、および DTCM (Cortex<sup>®</sup>-M7 の AHBS バス経由)、FMC または Quad SPI を経由した外部メモリ、および内部フラッシュメモリ。



## 2.2 メモリ構成

### 2.2.1 概要

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

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

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

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

### 2.2.2 メモリマップとレジスタ境界アドレス

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

次の表に、デバイスで使用可能なペリフェラルの境界アドレスを示します。

表 1. STM32F75xxx および STM32F74xxx レジスタ境界アドレス

境界アドレス	ペリフェラル	バス	レジスタマップ
0xA000 1000 - 0xA000 1FFF	Quad SPI 制御レジスタ	AHB3	<a href="#">セクション 14.5.14 : QUADSPI レジスタマップ (409 ページ)</a>
0xA000 0000 - 0xA000 0FFF	FMC 制御レジスタ		<a href="#">セクション 13.8 : FMC レジスタマップ (378 ページ)</a>
0x5006 0800 - 0x5006 0BFF	RNG	AHB2	<a href="#">セクション 19.4.4 : RNG レジスタマップ (549 ページ)</a>
0x5006 0400 - 0x5006 07FF	HASH		<a href="#">セクション 21.4.8 : HASH レジスタマップ (608 ページ)</a>
0x5006 0000 - 0x5006 03FF	CRYP		<a href="#">セクション 20.6.12 : CRYP レジスタマップ (587 ページ)</a>
0x5005 0000 - 0x5005 03FF	DCMI		<a href="#">セクション 17.8.12 : DCMI レジスタマップ (508 ページ)</a>
0x5000 0000 - 0x5003 FFFF	USB OTG FS		<a href="#">セクション 37.15.58 : OTG_FS/OTG_HS レジスタマップ (1401 ページ)</a>

表 1. STM32F75xxx および STM32F74xxx レジスタ境界アドレス (続き)

境界アドレス	ペリフェラル	バス	レジスタマップ
0x4004 0000 - 0x4007 FFFF	USB OTG HS	AHB1	<a href="#">セクション 37.15.58 : OTG_FS/OTG_HS レジスタマップ (1401 ページ)</a>
0x4002 B000 - 0x4002 BBFF	Chrom ART (DMA2D)		<a href="#">セクション 9.5.21 : DMA2D レジスタマップ (285 ページ)</a>
0x4002 8000 - 0x4002 93FF	ETHERNET MAC		<a href="#">セクション 38.8.5 : イーサネットレジスタマップ (1586 ページ)</a>
0x4002 6400 - 0x4002 67FF	DMA2		<a href="#">セクション 8.5.11 : DMA レジスタマップ (252 ページ)</a>
0x4002 6000 - 0x4002 63FF	DMA1		
0x4002 4000 - 0x4002 4FFF	BKPSRAM		<a href="#">セクション 5.3.27 : RCC レジスタマップ (194 ページ)</a>
0x4002 3C00 - 0x4002 3FFF	フラッシュインタフェースレジスタ		<a href="#">セクション 3.7.8 : フラッシュインタフェースレジスタマップ</a>
0x4002 3800 - 0x4002 3BFF	RCC		<a href="#">セクション 5.3.27 : RCC レジスタマップ (194 ページ)</a>
0x4002 3000 - 0x4002 33FF	CRC		<a href="#">セクション 12.4.6 : CRC レジスタマップ (304 ページ)</a>
0x4002 2800 - 0x4002 2BFF	GPIOK		<a href="#">セクション 6.4.11 : GPIO レジスタマップ (212 ページ)</a>
0x4002 2400 - 0x4002 27FF	GPIOJ		
0x4002 2000 - 0x4002 23FF	GPIOI		
0x4002 1C00 - 0x4002 1FFF	GPIOH		
0x4002 1800 - 0x4002 1BFF	GPIOG		
0x4002 1400 - 0x4002 17FF	GPIOF		
0x4002 1000 - 0x4002 13FF	GPIOE		
0x4002 0C00 - 0x4002 0FFF	GPIOD		
0x4002 0800 - 0x4002 0BFF	GPIOC		
0x4002 0400 - 0x4002 07FF	GPIOB		
0x4002 0000 - 0x4002 03FF	GPIOA		

表 1. STM32F75xxx および STM32F74xxx レジスタ境界アドレス (続き)

境界アドレス	ペリフェラル	バス	レジスタマップ
0x4001 6800 - 0x4001 6BFF	LCD-TFT	APB2	<a href="#">セクション 18.7.26 : LTDC レジスタマップ (542 ページ)</a>
0x4001 5C00 - 0x4001 5FFF	SAI2		<a href="#">セクション 33.5.10 : SAI レジスタマップ (1152 ページ)</a>
0x4001 5800 - 0x4001 5BFF	SAI1		<a href="#">セクション 33.5.10 : SAI レジスタマップ (1152 ページ)</a>
0x4001 5400 - 0x4001 57FF	SPI6		<a href="#">セクション 32.9.10 : SPI/I2S レジスタマップ (1107 ページ)</a>
0x4001 5000 - 0x4001 53FF	SPI5		
0x4001 4800 - 0x4001 4BFF	TIM11		<a href="#">セクション 24.5.12 : TIM10/11/13/14 レジスタマップ (822 ページ)</a>
0x4001 4400 - 0x4001 47FF	TIM10		
0x4001 4000 - 0x4001 43FF	TIM9		<a href="#">セクション 24.4.13 : TIM9/12 レジスタマップ (812 ページ)</a>
0x4001 3C00 - 0x4001 3FFF	EXTI		<a href="#">セクション 11.9.7 : EXTI レジスタマップ (298 ページ)</a>
0x4001 3800 - 0x4001 3BFF	SYSCFG		<a href="#">セクション 7.2.8 : SYSCFG レジスタマップ (220 ページ)</a>
0x4001 3400 - 0x4001 37FF	SPI4		<a href="#">セクション 32.9.10 : SPI/I2S レジスタマップ (1107 ページ)</a>
0x4001 3000 - 0x4001 33FF	SPI1		<a href="#">セクション 32.9.10 : SPI/I2S レジスタマップ (1107 ページ)</a>
0x4001 2C00 - 0x4001 2FFF	SDMMC1		<a href="#">セクション 35.8.16 : SDMMC レジスタマップ (1247 ページ)</a>
0x4001 2000 - 0x4001 23FF	ADC1 - ADC2 - ADC3		<a href="#">セクション 15.13.18 : ADC レジスタマップ (457 ページ)</a>
0x4001 1400 - 0x4001 17FF	USART6		<a href="#">セクション 31.8.12 : USART レジスタマップ (1050 ページ)</a>
0x4001 1000 - 0x4001 13FF	USART1		
0x4001 0400 - 0x4001 07FF	TIM8		<a href="#">セクション 22.4.24 : TIM1 レジスタマップ (701 ページ)</a>
0x4001 0000 - 0x4001 03FF	TIM1		

表 1. STM32F75xxx および STM32F74xxx レジスタ境界アドレス (続き)

境界アドレス	ペリフェラル	バス	レジスタマップ
0x4000 7C00 - 0x4000 7FFF	UART8	APB1	セクション 31.8.12 : USART レジスタマップ (1050 ページ)
0x4000 7800 - 0x4000 7BFF	UART7		
0x4000 7400 - 0x4000 77FF	DAC		セクション 16.5.15 : DAC レジスタマップ (482 ページ)
0x4000 7000 - 0x4000 73FF	PWR		セクション 4.4.4 : PWR 電源制御レジスタ 2 (PWR_CSR2) (128 ページ)
0x4000 6C00 - 0x4000 6FFF	HDMI-CEC		セクション 39.7.7 : HDMI-CEC レジスタマップ (1610 ページ)
0x4000 6800 - 0x4000 6BFF	CAN2		セクション 36.9.5 : bxCAN レジスタマップ (1291 ページ)
0x4000 6400 - 0x4000 67FF	CAN1		
0x4000 6000 - 0x4000 63FF	I2C4		セクション 30.7.12 : I2C レジスタマップ (985 ページ)
0x4000 5C00 - 0x4000 5FFF	I2C3		セクション 30.7.12 : I2C レジスタマップ (985 ページ)
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF	UART5		セクション 31.8.12 : USART レジスタマップ (1050 ページ)
0x4000 4C00 - 0x4000 4FFF	UART4		
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 4000 - 0x4000 43FF	SPDIFRX		
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3		セクション 32.9.10 : SPI/I2S レジスタマップ (1107 ページ)
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2		セクション 27.4.6 : IWDG レジスタマップ (866 ページ)
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		セクション 28.4.4 : WWDG レジスタマップ (873 ページ)
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		セクション 29.6.21 : RTC レジスタマップ (917 ページ)
0x4000 2400 - 0x4000 27FF	LPTIM1		セクション 26.6.11 : LPTIM レジスタマップ (857 ページ)
0x4000 2000 - 0x4000 23FF	TIM14		セクション 24.5.12 : TIM10/11/13/14 レジスタマップ (822 ページ)
0x4000 1C00 - 0x4000 1FFF	TIM13		
0x4000 1800 - 0x4000 1BFF	TIM12		セクション 24.4.13 : TIM9/12 レジスタマップ (812 ページ)
0x4000 1400 - 0x4000 17FF	TIM7		セクション 25.4.9 : TIM6/TIM7 レジスタマップ (836 ページ)
0x4000 1000 - 0x4000 13FF	TIM6		
0x4000 0C00 - 0x4000 0FFF	TIM5		
0x4000 0800 - 0x4000 0BFF	TIM4		セクション 23.4.22 : TIMx レジスタマップ (775 ページ)
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		



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

## 2.3 内蔵 SRAM

STM32F75xxx および STM32F74xxx の機能：

- 64 KB の データTCM RAMを含む最大 320 KB のシステム SRAM
- 16 KB の命令 RAM (ITCM-RAM)
- 4 KB のバックアップ SRAM (セクション 5.1.2 のバッテリーバックアップドメインを参照)

内蔵 SRAM は、最大 4 つのブロックに分割されます。

- システム SRAM
  - アドレス 0x2001 0000 に配置された、AHB バスマトリックスのすべての AHB マスタによるアクセスが可能な SRAM1
  - アドレス 0x2004 C000 に配置された、AHB バスマトリックスのすべての AHB マスタによるアクセスが可能な SRAM2
  - アドレス 0x2000 0000 に配置され、CPU のある特定の AHB スレーブバスを除く AHB バスマトリックスのすべての AHB マスタによるアクセスが可能な、TCM インタフェース (密に結合されたメモリインタフェース) 上の DTCM-RAM
- 命令 SRAM
  - アドレス 0x0000 0000 に配置され、CPU によってのみアクセス可能な命令 RAM (ITCM-RAM)

SRAM1 および SRAM2 は、バイト、ハーフワード (16 ビット)、またはフルワード (32 ビット) によるアクセスが可能です。一方、DTCM および ITCM RAM は、バイト、ハーフワード (16 ビット)、フルワード (32 ビット)、またはダブルワード (64 ビット) によるアクセスが可能です。これらのメモリは、最大システムクロック周波数 (ウェイトステートなし) でアドレス指定できます。

AHB マスタは、Ethernet または USB OTG HS からの SRAM への同時アクセスをサポートしています。たとえば、Ethernet MAC は、CPU が SRAM1 との読み出し/書き込みをしている間に、SRAM2 との読み出し/書き込みができます。

## 2.4 フラッシュメモリの概要

フラッシュメモリインタフェースは、フラッシュメモリへの CPU AXI および TCM アクセスを管理します。消去を実行し、フラッシュメモリ操作そして読み出し/書き込み保護メカニズムをプログラムします。また、TCM インタフェース上の ART または AXIM インタフェース上の L1 キャッシュによりコードの実行を加速します。

フラッシュメモリは、次のように構成されています。

- セクタに分割されたメインメモリブロック
- 情報ブロック：
  - システムメモリブートモードでデバイスがブートするシステムメモリ
  - ユーザデータ用の 1024 バイト OTP (one-time programmable)
  - デバイスが STANDBY または STOP モードのときに読み出し/書き込み保護、BOR レベル、ウォッチドッグソフトウェア/ハードウェアおよびリセットを設定するオプションバイト

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

## 2.5 ブート設定

STM32F75xxx および STM32F74xxxでは、表 2に示すように、2つのブート領域が、BOOT ピンと BOOT\_ADD0 および BOOT\_ADD1 オプションバイトでプログラムされたブートのベースアドレスによって選択できます。

表 2. ブートモード

ブートモード選択		ブート領域
BOOTピン	ブートアドレスのオプションバイト	
0	BOOT_ADD0[15:0]	ユーザオプションバイト BOOT_ADD0[15:0] で定義されたブートアドレスST のプログラム値 : アドレス 0x0020 0000 の ITCM 上のフラッシュ
1	BOOT_ADD1[15:0]	ユーザオプションバイト BOOT_ADD1[15:0]で定義されたブートアドレスST のプログラム値 : アドレス 0x0010 0000 のシステムブートルoader

BOOT ピンの値は、リセット解除後、SYSCLK の 4 番目の立ち上がりエッジでラッチされます。リセット後にBOOT ピンをセットするかどうかはユーザ次第です。

BOOT ピンは、デバイスが STANDBY モードを終了するときにも再度サンプリングされます。したがって、これらのピンはデバイスが STANDBY モードのときでも必要なブートモード設定に保たれる必要があります。

起動遅延時間後、プロセッサのリセットが解除される前に、ブート領域の選択が行われます。

BOOT\_ADD0 および BOOT\_ADD1 のアドレスオプションバイトを使用すると、下記を含む 0x0000 0000 から 0x3FFF FFFF までのすべてのブートメモリアドレスをプログラムすることができます。

- ITCM または AXIM インターフェースに配置されたすべてのフラッシュアドレス空間
- すべての RAM アドレス空間 : AXIM インタフェースに配置された ITCM、DTCM RAM、および SRAM
- システムメモリブートルoader

BOOT\_ADD0/BOOT\_ADD1の オプションバイトは、次のリセット後に他のブートアドレスからブートするために、リセット後に変更できます。

プログラムされたブートメモリアドレスがメモリ配置範囲外、または予約済み領域である場合、デフォルトのブートフェッチアドレスが次のようにプログラムされます。

- ブートアドレス 0 : アドレス 0x0020 0000 の ITCM 上のフラッシュ
- ブートアドレス 1 : アドレス 0x0000 0000 の ITCM 上の RAM

フラッシュのレベル 2 保護が有効の場合、(ITCM または AXIM インタフェース上の) フラッシュまたはシステムブートルoaderからのブートのみで使用できます。すでにBOOT\_ADD0 または BOOT\_ADD1 のオプションバイトにプログラムされたブートアドレスがメモリ範囲または (ITCM または AXIM 上の) RAM アドレスの範囲外である場合、デフォルトのフェッチはアドレス 0x00200000 の ITCM インタフェース上のフラッシュから強制的に行なわれます。

## 内蔵ブートローダ

内蔵ブートローダコードは、システムメモリに配置されます。これは生産時に ST によってプログラムされています。次のシリアルインタフェースのいずれかを使用して、フラッシュメモリを再プログラムするために使用します。

- PA9/PA10 ピンの USART1 と PB10/PB11 および PC10/PC11 ピンの USART3
- PB5/PB13 ピンの CAN2
- デバイスモードでの USB OTG FS (PA11/PA12 ピン) (DFU : デバイスファームウェアアップグレード)
- PB6/9 ピンの I2C1、PF0/PF1 ピンの I2C2、および PA8/PC9 ピンの I2C3

詳細については、アプリケーションノート AN2606 を参照してください。

デフォルトでシステムブートローダからのブートが選択されている場合、コードは TCM インタフェースから実行されます。これは、オプションバイトによって再プログラム可能です。

## 3 内蔵フラッシュメモリ (フラッシュ)

### 3.1 概要

フラッシュメモリインタフェースは、Cortex<sup>®</sup>-M7 AXI および TCMのフラッシュメモリへのアクセスを管理します。消去を実行し、フラッシュメモリ操作そして読み出し／書き込み保護メカニズムをプログラムします。

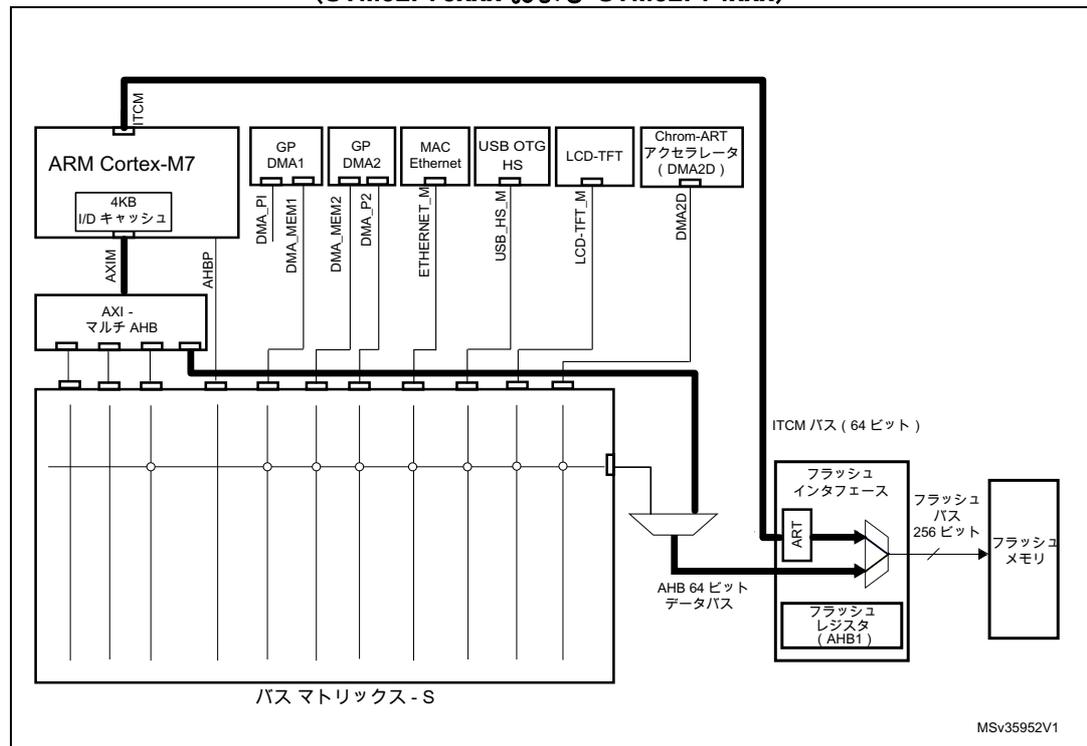
フラッシュメモリインタフェースは、ITCM インタフェース上の命令プリフェッチおよびキャッシュラインのシステムでコードの実行を加速します (ART Accelerator<sup>™</sup>)。

### 3.2 フラッシュの主な機能

- フラッシュメモリ読み出し操作
- フラッシュメモリプログラム／消去操作
- 読み出し／書き込み保護
- ITCM インタフェース上の 256 ビットのキャッシュライン 64 本 (ART Accelerator<sup>™</sup>)
- TCM 命令コードでのプリフェッチ

図 2 は、システムアーキテクチャ内のフラッシュメモリインタフェース接続を示しています。

図 2. システムアーキテクチャ内のフラッシュメモリインタフェース接続 (STM32F75xxx および STM32F74xxx)



## 3.3 フラッシュの機能詳細

### 3.3.1 フラッシュメモリの構成

フラッシュメモリは、以下のような主な特徴を備えています。

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

フラッシュメモリは、次のように構成されています。

- 32 KB のセクタ 4 つ、128 KB のセクタ 1 つ、256 KB のセクタ 3 つに分割されたメインメモリブロック
- 情報ブロックには次のものが含まれます。
  - システムメモリブートモードでデバイスがブートするシステムメモリ
  - ユーザーデータ用の 1024 バイト OTP (one-time programmable)
  - OTP エリアには、該当する OTP データブロックをロックするための追加の 16 バイトがあります。
  - 読み出し/書き込み保護、BOR レベル、ソフトウェア/ハードウェアウォッチドッグ、ブートメモリアドレス、およびデバイスが STANDBY または STOP モードのときのリセットを設定するオプションバイト

内蔵フラッシュには次の 3 つの主なインタフェースがあります。

- 64 ビット ITCM インタフェース：
  - Cortex-M7 の ITCM バスに接続され、命令実行およびデータ読み出しアクセスに使用します。
  - 書き込みアクセスは ITCM インタフェースではサポートされません。
  - 256 ビットのユニファイドキャッシュライン 64 本をサポートします (ART アクセラレータ)。
- 64 ビット AHB インタフェース：
  - AHB バスマトリックス経由で Cortex-M7 の AXI バス に接続され、コード実行、読み出しおよび書き込みアクセスに使用します。
  - フラッシュでの DMA およびペリフェラル DMA のデータ転送は、アドレス指定されたフラッシュインタフェースが TCM または AHB のいずれかにかかわらず、AHB インタフェースによって行われます。
- 32 ビット AHB レジスタインタフェース：
  - 制御およびステータスレジスタのアクセスに使用します。

表3 にメインメモリおよび情報ブロックの構成を示します。

表 3. フラッシュメモリの構成

ブロック	名前	AXIM インタフェース上の ブロックベースアドレス	ICTM インタフェース上の ブロックベースアドレス	セクタ サイズ
メインメモリ ブロック	セクタ 0	0x0800 0000 - 0x0800 7FFF	0x0020 0000 - 0x0020 7FFF	32 KB
	セクタ 1	0x0800 8000 - 0x0800 FFFF	0x0020 8000 - 0x0020 FFFF	32 KB
	セクタ 2	0x0801 0000 - 0x0801 7FFF	0x0021 0000 - 0x0021 7FFF	32 KB
	セクタ 3	0x0801 8000 - 0x0801 FFFF	0x0021 8000 - 0x0021 FFFF	32 KB
	セクタ 4	0x0802 0000 - 0x0803 FFFF	0x0022 0000 - 0x0023 FFFF	128 KB
	セクタ 5	0x0804 0000 - 0x0807 FFFF	0x0024 0000 - 0x0027 FFFF	256 KB
	セクタ 6	0x0808 0000 - 0x080B FFFF	0x0028 0000 - 0x002B FFFF	256 KB
	セクタ 7	0x080C 0000 - 0x080F FFFF	0x002C 0000 - 0x002F FFFF	256 KB
情報ブロック	システムメモリ	0x1FF0 0000 - 0x1FF0 EDBF	0x0010 0000 - 0x0010 EDBF	60 KB
	OTP	0x1FF0 F000 - 0x1FF0 F41F	0x0010 F000 - 0x0010 F41F	1024 バイト
	オプションバイト	0x1FFF 0000 - 0x1FFF 001F	-	32 バイト

### 3.3.2 読み出しアクセスの遅延

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

ウェイトステートと CPU クロック周波数との対応は、表 13 および 表 4 に記載されています。

- 注：
- VOS[1:0] = "0x01" のとき、 $f_{HCLK}$  の最大値 = 144 MHz
  - VOS[1:0] = "0x10" のとき、 $f_{HCLK}$  の最大値 = 168 MHz  
オーバードライブモードをアクティブにすると、180 MHz まで拡大することができます。
  - VOS[1:0] = "0x11" のとき、 $f_{HCLK}$  の最大値 = 180 MHz  
オーバードライブモードをアクティブにすると、216 MHz まで拡大することができます。
  - $V_{DD}$  の範囲が 1.8 ~ 2.1 V のときは、オーバードライブモードは使用できません。

オーバードライブモードを有効にする方法の詳細については、セクション 4.1.4 : 電圧レギュレータを参照してください。

表 4. CPU クロック（HCLK）周波数によるウェイトステート数

ウェイトステート (WS) (LATENCY)	HCLK (MHz)			
	電圧範囲 2.7 V - 3.6 V	電圧範囲 2.4 V - 2.7 V	電圧範囲 2.1 V - 2.4 V	電圧範囲 1.8 V - 2.1 V
0 WS (1 CPU サイクル)	0 < HCLK ≤ 30	0 < HCLK ≤ 24	0 < HCLK ≤ 22	0 < HCLK ≤ 20
1 WS (2 CPU サイクル)	30 < HCLK ≤ 60	24 < HCLK ≤ 48	22 < HCLK ≤ 44	20 < HCLK ≤ 40
2 WS (3 CPU サイクル)	60 < HCLK ≤ 90	48 < HCLK ≤ 72	44 < HCLK ≤ 66	40 < HCLK ≤ 60
3 WS (4 CPU サイクル)	90 < HCLK ≤ 120	72 < HCLK ≤ 96	66 < HCLK ≤ 88	60 < HCLK ≤ 80
4 WS (5 CPU サイクル)	120 < HCLK ≤ 150	96 < HCLK ≤ 120	88 < HCLK ≤ 110	80 < HCLK ≤ 100
5 WS (6 CPU サイクル)	150 < HCLK ≤ 180	120 < HCLK ≤ 144	110 < HCLK ≤ 132	100 < HCLK ≤ 120
6 WS (7 CPU サイクル)	180 < HCLK ≤ 210	144 < HCLK ≤ 168	132 < HCLK ≤ 154	120 < HCLK ≤ 140



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

表 4. CPU クロック（HCLK）周波数によるウェイトステート数（続き）

ウェイトステート (WS) (LATENCY)	HCLK (MHz)			
	電圧範囲 2.7 V - 3.6 V	電圧範囲 2.4 V - 2.7 V	電圧範囲 2.1 V - 2.4 V	電圧範囲 1.8 V - 2.1 V
7 WS (8 CPU サイクル)	210 < HCLK ≤ 216	168 < HCLK ≤ 192	154 < HCLK ≤ 176	140 < HCLK ≤ 160
8 WS (9 CPU サイクル)	-	192 < HCLK ≤ 216	176 < HCLK ≤ 198	160 < HCLK ≤ 180
9 WS (10 CPU サイクル)	-	-	198 < HCLK ≤ 216	-

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

CPU 周波数でフラッシュメモリにアクセスするためにウェイトステート数を調整するには、以下のソフトウェアシーケンスを使用することを推奨します。

### CPU 周波数の増加

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

### CPU 周波数の減少

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

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

## 命令プリフェッチ

各フラッシュの読み出し操作では、起動されるプログラムによって、32 ビットの命令 8 個から 16 ビットの命令 16 個までを表す 256 ビットが提供されます。したがって連続コードの場合は、その前に読み出された命令ラインの実行に 8CPU サイクル以上が必要となります。ITCM バスでのプリフェッチを使用すると、CPU によって現在の命令ラインが要求されている間にフラッシュの次の連続命令ラインを読み出すことができます。FLASH\_ACR レジスタの PRFTEN ビットをセットすることによって、プリフェッチを有効にできます。この機能は、フラッシュのアクセスに 1 つ以上のウェイトステートが必要な場合に有用です。コードが連続でない場合（分岐）、現在使用されている命令ラインやプリフェッチされた命令ラインには命令がないことがあります。この場合、ペナルティはサイクル数換算でウェイトステート数以上となります。

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

独自仕様の適応型リアルタイム（ART）メモリアクセラレータは、FPU プロセッサのある STM32 の業界標準 ARM® Cortex®-M7 用に最適化されています。これはフラッシュメモリ技術における FPU 搭載 ARM® Cortex®-M7 固有のパフォーマンス上の利点（通常、動作周波数が高くなると、プロセッサはフラッシュメモリを待つ必要性が発生します）を調整します。

プロセッサのパフォーマンスを最大限に引き出すため、アクセラレータはユニファイド命令キャッシュと、256 ビットのフラッシュメモリからのプログラム実行速度を上げる分岐キャッシュを実装しています。CoreMark ベンチマークによると、ART アクセラレータによって達成されるパフォーマンスは、最大 CPU 周波数 216 MHz におけるフラッシュメモリからの 0 ウェイトステートプログラム実行と同等となります。

ART アクセラレータは、ITCM インタフェース上でのフラッシュのアクセスにのみ使用できます。

ジャンプによる時間のロスを減らすため、ART アクセラレータ内に 256 ビットのライン 64 本を維持することができます。この機能は、FLASH\_CR レジスタの ARTEN ビットをセットすることによって有効にできます。ART アクセラレータは統合されており、命令だけでなく、データリテラルプールも含まれます。ミス（現在使用しているデータラインまたは命令キャッシュメモリに要求されたデータがない）が発生するたびに、読み出されるラインが ART の命令キャッシュメモリにコピーされます。命令キャッシュメモリに含まれるデータが CPU によって要求されると、遅延を挿入することなくそのデータが提供されます。すべてのキャッシュメモリラインが満たされると、LRU（最も長い時間使われていない）ポリシーを使用してメモリキャッシュの中で置換するラインを決定します。この機能は、ループを含むコードの場合に特に有用です。

**注：** ユーザー設定セクタのデータは、キャッシュできません。

### 3.3.3 フラッシュのプログラムおよび消去操作

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

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

### 3.3.4 フラッシュ制御レジスタのアンロック

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

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

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

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

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

### 3.3.5 プログラム／消去の並列処理

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

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

表 5 は、正しい PSIZE 値を記載しています。

表 5. プログラム／消去の並列処理

	電圧範囲 2.7~3.6 V、外部 V <sub>pp</sub>	電圧範囲 2.7 - 3.6 V	電圧範囲 2.4 - 2.7 V	電圧範囲 2.1 - 2.4 V	電圧範囲 1.8 V - 2.1 V
並列処理サイズ	x64	x32	x16		x8
PSIZE(1:0)	11	10	01		00

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

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

## 3.3.6 フラッシュ消去シーケンス

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

### セクタ消去

セクタを消去するには、次の手順に従います。

1. FLASH\_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH\_CR レジスタで SER ビットをセットし、メインメモリブロックで 8 個あるセクタの中から消去するセクタ（SNB）を選択します。
3. FLASH\_CR レジスタの STRT ビットをセットします。
4. BSY ビットがクリアされるのを待ちます。

### 全体消去

全体消去の実施には、次のシーケンスを推奨します。

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

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

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

## 3.3.7 フラッシュプログラミングシーケンス

### 標準プログラミング

フラッシュメモリのプログラミングシーケンスは、次のようになっています。

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

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

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



## プログラミングエラー

エラーが発生した場合、フラッシュ操作（プログラミングまたは消去）は以下のエラーのいずれかにより中止されます。

- **PGAERR**：配置プログラミングエラー  
128 ビットの列境界を超えるようなデータをフラッシュメモリにプログラムすることはできません。そのような場合、書き込み操作は実施されず、FLASH\_SR レジスタのプログラム配置エラーフラグ（PGAERR）がセットされます。
- **PGPERR**：プログラミング並列処理エラー  
書き込みアクセスタイプ（バイト、ハーフワード、ワード、ダブルワード）は、選択した並列処理タイプ（x8、x16、x32、x64）に対応している必要があります。対応していない場合、書き込み操作は実施されず、FLASH\_SR レジスタのプログラム並列処理エラーフラグ（PGPERR）がセットされます。
- **ERSERR**：消去シーケンスエラー  
制御レジスタが正しく設定されていない時にコードによってフラッシュへの消去操作が実施されると、ERSERR エラーフラグがセットされます。
- **WRPERR**：書き込み保護エラー  
次のいずれかの条件が発生した場合、WRPERR がセットされます。
  - 書き込み保護領域（WRP）にプログラミングまたは消去を試みた場合
  - システムメモリ領域にプログラミングまたは消去を試みた場合
  - 既にロックされている OTP エリアに書き込みした場合
  - 読み出し保護（RDP）レベルがセットされている状態で、オプションバイトの変更を試みた場合
  - フラッシュメモリが読み出し保護されており、侵入が検出された場合

フラッシュメモリ内の消去操作が ART アクセラレータのデータにも関連している場合は、コード実行中にこれらのデータにアクセスする前に、これらのデータが再度書き込まれることを確認する必要があります。これが安全に実施できない場合には、FLASH\_CR レジスタで ARTRST または ARTEN ビットをセットして、ART アクセラレータの一掃または機能停止を行うことを推奨します。

**注：** *ART キャッシュは、ART アクセラレータが無効（ARTEN = 0）のときのみ一掃できます。*

### 3.3.8 フラッシュ 割り込み

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

この場合、FLASH\_SR レジスタで操作終了（EOP）ビットがセットされます。プログラム、消去、または読み出し操作リクエスト中にエラーが発生すると、FLASH\_SR レジスタで次のいずれかのエラーフラグがセットされます。

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

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

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

表 6. フラッシュ割り込みリクエスト

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

## 3.4 フラッシュオプションバイト

### 3.4.1 オプションバイトの説明

オプションバイトは、アプリケーション要件によってエンドユーザが設定します。表 7 は、情報ブロック内のこれらのバイトの構成を示しています。

オプションバイトは、ユーザ設定メモリ位置またはオプションバイトレジスタから読み出しできます。

- [フラッシュオプション制御レジスタ \(FLASH\\_OPTCR\)](#)
- [フラッシュオプション制御レジスタ \(FLASH\\_OPTCR1\)](#)

表 7. オプションバイトの構成

AXI アドレス	[63:16]	[15:0]
0x1FFF 0000	予約済み	ROP およびユーザオプションバイト (RDP & USER)
0x1FFF 0008	予約済み	IWDG_STOP、IWDG_STBY、書き込み保護 nWRP (セクタ 0 から 7)、ユーザオプションバイト
0x1FFF 0010	予約済み	BOOT_ADD0
0x1FFF 0018	予約済み	BOOT_ADD1

## ユーザオプションバイトおよび読み出し保護オプションバイト

メモリアドレス：0x1FFF 0000

ST のプログラム値：0x5500AAFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP								nRST_S TDBY	nRST_S TOP	IWDG_S W	WWDG_ SW	BOR_LEV[1:0]		Res.	Res.
r								r	r	r	r	r	r		

ビット 31:13 未使用

ビット 15:8 **RDP**：読み出し保護

読み出し保護は、フラッシュメモリに格納されているソフトウェアコードの保護に役立ちます。

0xAA：レベル 0、保護なし

0xCC：レベル 2、チップ保護（デバッグと RAM からのブート機能は無効）

上記以外：レベル 1、メモリの読み出し保護（デバッグ機能は制限付き）

ビット 7 **nRST\_STDBY**

0：STANDBY モードに移行するときにリセットを生成します。

1：リセットは生成されません。

ビット 6 **nRST\_STOP**

0：STOP モードに移行するときにリセットを生成します。

1：リセットは生成されません。

ビット 5 **IWDG\_SW**：独立型ウォッチドッグの選択

0：ハードウェアによる独立型ウォッチドッグ

1：ソフトウェアによる独立型ウォッチドッグ

ビット 4 **WWDG\_SW**：ウィンドウ型ウォッチドッグの選択

0：ハードウェアによるウィンドウ型ウォッチドッグ

1：ソフトウェアによるウィンドウ型ウォッチドッグ

ビット 3:2 **BOR\_LEV**：BOR リセットレベル

このビットには、リセットをアクティブにしたリセットを解除したりする供給レベル閾値が含まれています。フラッシュメモリに新しい BOR レベルをプログラムするように書き込むことができます。

00：BOR レベル 3 (VBOR3)、ブラウンアウト閾値レベル 3

01：BOR レベル 2 (VBOR2)、ブラウンアウト閾値レベル 2

10：BOR レベル 1 (VBOR1)、ブラウンアウト閾値レベル 1

11：BOR オフ、POR/PDR リセット閾値レベルが適用されます。

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

ビット 1:0 未使用

## ユーザオプションバイトおよび書き込み保護オプションバイト

メモリアドレス : 0x1FFF 0008

ST のプログラムされた値 : 0x0000FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG_STOP	IWDG_STDBY	Res.	Res.	Res.	Res.	Res.	Res.	nWRP <i>i</i>							
r	r							r	r	r	r	r	r	r	r

ビット 31:16 未使用

ビット 15 **IWDG\_STOP** : STOP モードでの独立型ウォッチドッグカウンタの凍結

0 : STOP モードでの IWDG カウンタを凍結します。

1 : STOP モードで IWDG カウンタはアクティブになっています。

ビット 14 **IWDG\_STDBY** : STANDBY モードでの独立型ウォッチドッグカウンタの凍結

0 : STANDBYモードでのIWDGカウンタを凍結します。

1 : STANDBY モードで IWDG カウンタはアクティブになっています。

ビット 13:8 未使用

ビット 7:0 **nWRP*i*** : セクタ *i* の書き込み保護なし

0 : セクタ *i* で書き込み保護がアクティブになっています。

1 : セクタ *i* で書き込み保護がアクティブになっていません。

## ブートアドレスオプションバイト (BOOT ピン = 0 のとき)

メモリアドレス : 0x1FFF 0010

ST のプログラム値 : 0xFF7F 0080 (ITCM-FLASH ベースアドレス)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADD0[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 未使用

ビット 15:0 **BOOT\_ADD0[15:0]** : BOOT ピン = 0 のときのブートメモリベースアドレス

BOOT\_ADD0[15:0] はアドレス [29:14] に対応します。

ブートベースアドレスは、0x0000 0000 から 0x2004 FFFF までのアドレス範囲のみを 16 KB単位でサポートします。

例 :

BOOT\_ADD0 = 0x0000 : ITCM RAM (0x0000 0000) からのブート

BOOT\_ADD0 = 0x0040 : システムメモリブートローダ (0x0010 0000) からのブート

BOOT\_ADD0 = 0x0080 : ITCM インタフェース上のフラッシュ (0x0020 0000) からのブート

BOOT\_ADD0 = 0x2000 : AXIM インタフェース上のフラッシュ (0x0800 0000) からのブート

BOOT\_ADD0 = 0x8000 : DTCM RAM (0x2000 0000) からのブート

BOOT\_ADD0 = 0x8004 : SRAM1 (0x2001 0000) からのブート

BOOT\_ADD0 = 0x8013 : SRAM2 (0x2004 C000) からのブート



## ブートアドレスオプションバイト（BOOT ピン = 1 のとき）

メモリアドレス：0x1FFF 0018

ST のプログラム値：0xFFBF0040（システムメモリブートローダのアドレス）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADD1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **未使用**

ビット 15:0 **BOOT\_ADD1[15:0]**：BOOT ピン = 1 のときのブートメモリベースアドレス

BOOT\_ADD1[15:0] はアドレス [29:14] に対応します。

ブートベースアドレスは、0x0000 0000 から 0x2004 FFFF までのアドレス範囲のみを 16 KB 単位でサポートします。

例：

BOOT\_ADD1 = 0x0000：ITCM RAM (0x0000 0000) からのブート

BOOT\_ADD1 = 0x0040：システムメモリブートローダ (0x0010 0000) からのブート

BOOT\_ADD1 = 0x0080：ITCM インタフェース上のフラッシュ (0x0020 0000) からのブート

BOOT\_ADD1 = 0x2000：AXIM インタフェース上のフラッシュ (0x0800 0000) からのブート

BOOT\_ADD1 = 0x8000：DTCM RAM (0x2000 0000) からのブート

BOOT\_ADD1 = 0x8004：SRAM1 (0x2001 0000) からのブート

BOOT\_ADD1 = 0x8013：SRAM2 (0x2004 0000) からのブート

### 3.4.2 オプションバイトのプログラミング

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

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

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

#### ユーザオプションバイトの変更

ユーザオプション値を変更するには、次のシーケンスに従います。

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

**注：** *まず情報ブロックを消去し、次にすべてのオプションバイトを FLASH\_OPTCR レジスタに含まれている値でプログラムすると、オプションの値が自動的に変更されます。*

## 3.5 フラッシュメモリの保護

### 3.5.1 読み出し保護（RDP）

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

- レベル 0：読み出し保護なし

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

- レベル 1：読み出し保護有効

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

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

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

- レベル 2：デバッグ／チップ読み出し無効

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

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

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

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

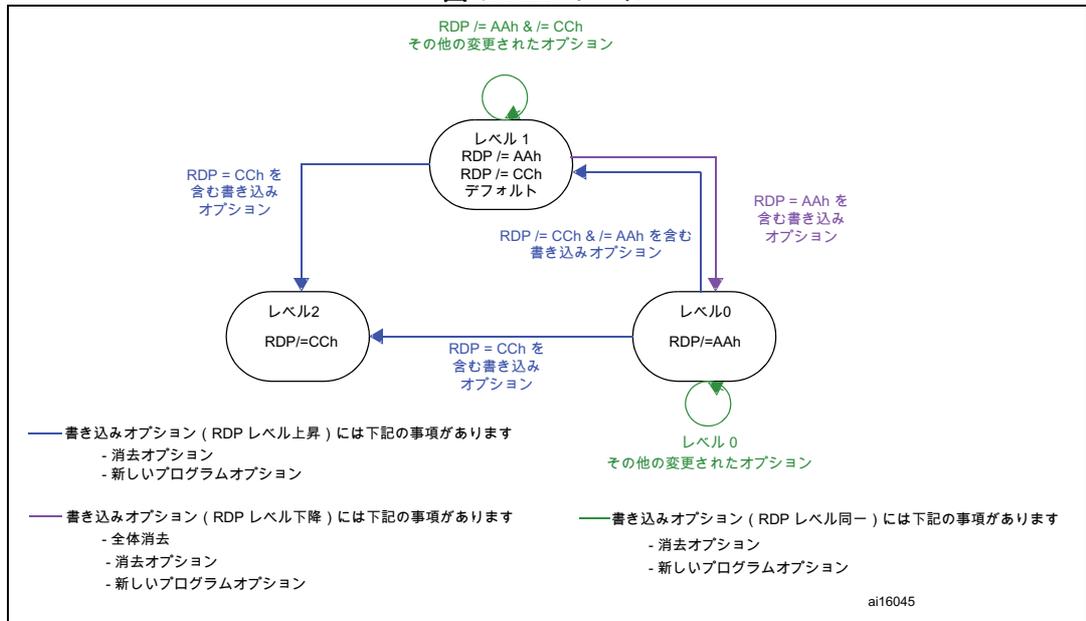
表 8. アクセスと読み出し保護レベル

メモリ領域	保護レベル	デバッグ機能、RAM またはシステム メモリブートローダからのブート			フラッシュメモリからのブート		
		読み出し	書き込み	消去	読み出し	書き込み	消去
メインフラッシュメモリ およびバックアップ SRAM	レベル 1	X		X <sup>(1)</sup>		O	
	レベル 2		X			O	
オプションバイト	レベル 1		O			O	
	レベル 2		X			X	
OTP	レベル 1	X		該当なし	O		該当なし
	レベル 2	X		該当なし	O		該当なし

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

図 3 は、ある RDP レベルから別の RDP レベルへの遷移を示しています。

図 3. RDP レベル



## 3.5.2 書き込み保護

フラッシュメモリでは、最大 8 個までのユーザセクタをプログラムカウンタコンテキストの損失による不要な書き込み操作から保護できます。FLASH\_OPTCR または FLASH\_OPTCR1 レジスタの非書き込み保護  $nWRP_i$  ビット ( $0 \leq i \leq 7$ ) がローのときは、該当するセクタを消去したりプログラムしたりすることができません。結果として、セクタが 1 つ書き込み保護されていると全体消去が実施できません。

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

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

### 書き込み保護エラーフラグ

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

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

- セクタ消去がリクエストされ、セクタ番号 SNB フィールドが有効でない場合
- オプションビットによって 1 つ以上のユーザセクタが書き込み保護されている（FLASH\_OPTCR レジスタで、MER および  $nWR_i = 0$ 、ただし  $0 \leq i \leq 7$  ビット）ときに、全体消去がリクエストされた場合
- 書き込み保護されているセクタでセクタ消去がリクエストされた場合（FLASH\_OPTCR レジスタで、SER = 1、SNB =  $i$  および  $nWRP_i = 0$ 、ただし  $0 \leq i \leq 7$  ビット）
- フラッシュメモリが読み出し保護されており、侵入が検出された場合

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

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

## 3.6 OTP（One-time programmable）バイト

表 9 は、OTP エリアの構成を示しています。

表 9. OTP エリアの構成

OTP ブロック	[255:224]	[223:193]	[192:161]	[160:128]	[127:96]	[95:64]	[63:32]	[31:0]	アドレス バイト 0
0	OTP0	OTP0	OTP0	OTP0	OTP0	OTP0	OTP0	OTP0	0x1FF0 F000
	OTP0	OTP0	OTP0	OTP0	OTP0	OTP0	OTP0	OTP0	0x1FF0 F020
1	OTP1	OTP1	OTP1	OTP1	OTP1	OTP1	OTP1	OTP1	0x1FF0 F040
	OTP1	OTP1	OTP1	OTP1	OTP1	OTP1	OTP1	OTP1	0x1FF0 F060
-	-	-	-	-	-	-	-	-	-
14	OPT14	OPT14	OPT14	OPT14	OPT14	OPT14	OPT14	OPT14	0x1FF0 F380
	OPT14	OPT14	OPT14	OPT14	OPT14	OPT14	OPT14	OPT14	0x1FF0 F3A0
15	OPT15	OPT15	OPT15	OPT15	OPT15	OPT15	OPT15	OPT15	0x1FF0 F3C0
	OPT15	OPT15	OPT15	OPT15	OPT15	OPT15	OPT15	OPT15	0x1FF0 F3E0
ロック ブロック	予約済み	予約済み	予約済み	予約済み	LOCK15... LOCKB12	LOCK11... LOCKB8	LOCK7... LOCKB4	LOCK3... LOCKB0	0x1FF0 F400

OTP エリアは、64 バイトの OTP データブロック 16 個と、16 バイトのロック OTP ブロック 1 個に分割されています。OTP データブロックおよびロックブロックは、消去できません。ロックブロックには、該当する OTP データブロック（ブロック 0 から 15）をロックする、16 バイトの LOCKBi (0 ≤ i ≤ 15) が含まれています。各 OTP データブロックは、該当する OTP ロックバイトで値 0x00 がプログラムされるまでプログラムできます。ロックバイトには、0x00 および 0xFF の値が含まれている必要があります。これらの値が含まれていない場合、OTP バイトが正しく考慮されないことがあります。

## 3.7 フラッシュレジスタ

### 3.7.1 フラッシュアクセス制御レジスタ（FLASH\_ACR）

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

アドレスオフセット：0x00  
リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

ビット 31:12 予約済み、クリア状態を保つ必要があります。

ビット 11 **ARTRST** : ART アクセラレータのリセット

- 0 : ART アクセラレータはリセットされません。
- 1 : ART アクセラレータはリセットされます。

ビット 10 予約済み、クリア状態を保つ必要があります。

ビット 9 **ARTEN** : ART アクセラレータ有効化

- 0 : ART アクセラレータは無効です。
- 1 : ART アクセラレータは有効です。

ビット 8 **PRFTEN** : プリフェッチ有効化

- 0 : プリフェッチは無効です。
- 1 : プリフェッチは有効です。

ビット 7:4 予約済み、クリア状態を保つ必要があります。

ビット 3:0 **LATENCY[3:0]** : 遅延

これらのビットは、フラッシュメモリアクセス時間に対する CPU クロック周期の割合を表します。

- 0000 : ウェイトステート 0 個
- 0001 : ウェイトステート 1 個
- 0010 : ウェイトステート 2 個
- 
- 
- 
- 1110 : ウェイトステート 14 個
- 1111 : ウェイトステート 15 個

### 3.7.2 フラッシュキーレジスタ（FLASH\_KEYR）

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

アドレスオフセット：0x04

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワードアクセス

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

ビット 31:0 **FKEYR**：FPEC キー

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

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

### 3.7.3 フラッシュオプションキーレジスタ（FLASH\_OPTKEYR）

フラッシュオプションキーレジスタは、情報ブロックでのプログラムおよび消去操作を許可するために使用します。

アドレスオフセット：0x08

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワードアクセス

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

ビット 31:0 **OPTKEYR**：オプションバイトキー

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

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

### 3.7.4 フラッシュステータスレジスタ (FLASH\_SR)

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

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

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

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

ビット 31:17 予約済み、クリア状態を保つ必要があります。

ビット 16 **BSY** : ビジー

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

- 0 : 進行中のフラッシュメモリ操作はありません。
- 1 : 進行中のフラッシュメモリ操作があります。

ビット 15:8 予約済み、クリア状態を保つ必要があります。

ビット 7 **ERSERR** : 消去シーケンスエラー

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

ビット 6 **PGPERR** : プログラミング並列処理エラー

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

ビット 5 **PGAERR** : プログラミング配置エラー

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

ビット 4 **WRPERR** : 書き込み保護エラー

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

ビット 3:2 予約済み、クリア状態を保つ必要があります。

ビット 1 **OPERR** : 操作エラー

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

ビット 0 **EOP** : 操作終了

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

## 3.7.5 フラッシュ制御レジスタ（FLASH\_CR）

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

アドレスオフセット：0x10

リセット値：0x8000 0000

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

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

### ビット 31 LOCK：ロック状態

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

ビット 30:26 予約済み、クリア状態を保つ必要があります。

### ビット 25 ERRIE：エラー割り込み有効化

このビットは、FLASH\_SR レジスタの OPERR ビットが 1 にセットされると割り込み生成を有効にします。

0：エラー割り込み生成は無効です。

1：エラー割り込み生成は有効です。

### ビット 24 EOPIE：操作終了割り込み有効化

このビットは、FLASH\_SR レジスタの EOP ビットが 1 になると割り込み生成を有効にします。

0：割り込み生成は無効です。

1：割り込み生成は有効です。

ビット 23:17 予約済み、クリア状態を保つ必要があります。

### ビット 16 STRT：Start

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

ビット 15:10 予約済み、クリア状態を保つ必要があります。

### ビット 9:8 PSIZE：プログラムサイズ

このビットは、プログラムの並列処理を選択します。

00 プログラム x8

01 プログラム x16

10 プログラム x32

11 プログラム x64

ビット 7 予約済み、クリア状態を保つ必要があります。

ビット 6:3 **SNB[3:0]** : セクタ番号  
 このビットは、消去するセクタを選択します。  
 0000 セクタ 0  
 0001 セクタ 1  
 ...  
 0111 セクタ 7  
 他は許可されません

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

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

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

### 3.7.6 フラッシュオプション制御レジスタ（FLASH\_OPTCR）

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

アドレスオフセット : 0x14

リセット値 : 0xC0FFAAFD. オプションバイトには、リセット解除時にフラッシュメモリからの値がロードされます。

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IWDG_S TOP	IWDG_S TDBY	Res.	Res.	Res.	Res.	Res.	Res.	nWRP[7:0]							
rw	rw							rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP[7:0]								nRST_S TDBY	nRST_S TOP	IWDG_S W	WWDG_ SW	BOR_LEV[1:0]		OPTSTR T	OPTLOC K
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs	rs

ビット 31 **IWDG\_STOP** : STOP モードでの独立型ウォッチドッグカウンタの凍結  
 0 : STOP モードで IWDG カウンタを凍結します。  
 1 : STOP モードで IWDG カウンタはアクティブになっています。

ビット 30 **IWDG\_STDBY** : STANDBY モードでの独立型ウォッチドッグカウンタの凍結  
 0 : STANDBY モードで IWDG カウンタを凍結します。  
 1 : STANDBY モードで IWDG カウンタはアクティブになっています。

ビット 29:24 予約済み、クリア状態を保つ必要があります。

ビット 23:16 **nWRP[7:0]** : 書き込み保護なし  
 これらのビットには、リセット後のセクタ 0 から 7 の書き込み保護オプションバイトの値が含まれています。これらのビットを書き込むことで、フラッシュメモリに新しい書き込み保護をプログラムすることができます。  
 0 : セクタ i で書き込み保護がアクティブになっています。  
 1 : セクタ i で書き込み保護がアクティブになっていません。



**ビット 15:8 RDP[7:0] : 読み出し保護**

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

0xAA : レベル 0、読み出し保護はアクティブではありません。

0xCC : レベル 2、チップ読み出し保護がアクティブです。

その他 : レベル 1、メモリの読み出し保護がアクティブです。

**ビット 7:4 USER : ユーザオプションバイト**

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

ビット 7 : nRST\_STDBY

ビット 6 : nRST\_STOP

ビット 5 : IWDG\_SW

ビット 4 : WWDG\_SW

**ビット 3:2 BOR\_LEV[1:0] : BOR リセットレベル**

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

00 : BOR レベル 3 (VBOR3)、ブラウンアウト閾値レベル 3

01 : BOR レベル 2 (VBOR2)、ブラウンアウト閾値レベル 2

10 : BOR レベル 1 (VBOR1)、ブラウンアウト閾値レベル 1

11 : BOR オフ、POR/PDR リセット閾値レベルが適用されます。

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

**ビット 1 OPTSTRT : オプション開始**

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

**ビット 0 OPTLOCK : オプションロック**

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

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

**注 :** *IWDG\_SW、IWDG\_STOP、または IWDG\_STDBY オプションバイトを変更する場合、変更内容を有効にするにはシステムリセットが必要です。*

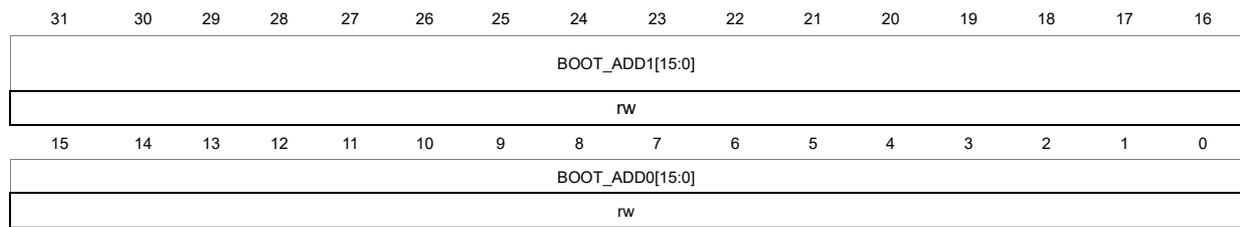
### 3.7.7 フラッシュオプション制御レジスタ (FLASH\_OPTCR1)

FLASH\_OPTCR1 レジスタは、ユーザオプションバイトの変更に使用します。

アドレスオフセット : 0x18

リセット値 : 0xFF7F 0080 (ITCM-FLASH) オプションバイトには、リセット解除時にフラッシュメモリからの値が入力されます。

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



ビット 31:16 **BOOT\_ADD1[15:0]** : BOOT ピン = 1 のときのブートベースアドレス

BOOT\_ADD1[15:0] はアドレス [29:14] に対応します。

ブートメモリアドレスは、16 KB 単位で 0x0000 0000 から 0x2004 FFFF の範囲にあるすべてのアドレスに対してプログラムすることができます。

例 :

BOOT\_ADD1 = 0x0000 : ITCM RAM (0x0000 0000) からのブート

BOOT\_ADD1 = 0x0040 : システムメモリブートローダ (0x0010 0000) からのブート

BOOT\_ADD1 = 0x0080 : ITCM インタフェース上のフラッシュ (0x0020 0000) からのブート

BOOT\_ADD1 = 0x2000 : AXIM インタフェース上のフラッシュ (0x0800 0000) からのブート

BOOT\_ADD1 = 0x8000 : DTCM RAM (0x2000 0000) からのブート

BOOT\_ADD1 = 0x8004 : SRAM1 (0x2001 0000) からのブート

BOOT\_ADD1 = 0x8013 : SRAM2 (0x2004 C000) からのブート

ビット 15:0 **BOOT\_ADD0[15:0]** : BOOT ピン = 0 のときのブートベースアドレス

BOOT\_ADD0[15:0] はアドレス [29:14] に対応します。

ブートベースアドレスは、16 KB 単位で 0x0000 0000 から 0x2004 FFFF の範囲にあるすべてのアドレスに対してプログラムすることができます。

例 :

BOOT\_ADD0 = 0x0000 : ITCM RAM (0x0000 0000) からのブート

BOOT\_ADD0 = 0x0040 : システムメモリブートローダ (0x0010 0000) からのブート

BOOT\_ADD0 = 0x0080 : ITCM インタフェース上のフラッシュ (0x0020 0000) からのブート

BOOT\_ADD0 = 0x2000 : AXIM インタフェース上のフラッシュ (0x0800 0000) からのブート

BOOT\_ADD0 = 0x8000 : ITCM RAM (0x2000 0000) からのブート

BOOT\_ADD0 = 0x8004 : SRAM1 (0x2001 0000) からのブート

BOOT\_ADD0 = 0x8013 : SRAM2 (0x2004 C000) からのブート

## 3.7.8 フラッシュインタフェースレジスタマップ

表 10. フラッシュレジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARTRST	Res.	ARTEN	PRFTEN	Res.	Res.	Res.	Res.	LATENCY[3:0]			
	リセット値																							0		0	0					0	0	0
0x04	FLASH_KEYR	KEY[31:16]															KEY[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	FLASH_OPTKEYR	OPTKEYR[31:16]															OPTKEYR[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERSERR	PGPERR	PGAERR	WRPERR	Res.	Res.	OPERR	EOP
	リセット値																0											0	0	0	0			0
0x10	FLASH_CR	LOCK	Res.	Res.	Res.	Res.	Res.	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STRT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSIZE[1:0]	Res.	SNB[3:0]			MER	SER	PG		
	リセット値	1						0	0								0									0	0	0	0	0	0	0	0	0
0x14	FLASH_OPTCR	IWDG_STOP	IWDG_STDBY	Res.	Res.	Res.	Res.	Res.	Res.	nWRP[7:0]							RDP[7:0]							nRST_STDBY	nRST_STOP	IWDG_SW	WWDG_SW	BOR_LEV[1:0]		OPTSTR1	OPTLOCK			
	リセット値	1	1								1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	
0x18	FLASH_OPTCR1	BOOT_ADD1[15:0]															BOOT_ADD0[15:0]																	
	リセット値	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

## 4 電源コントローラ (PWR)

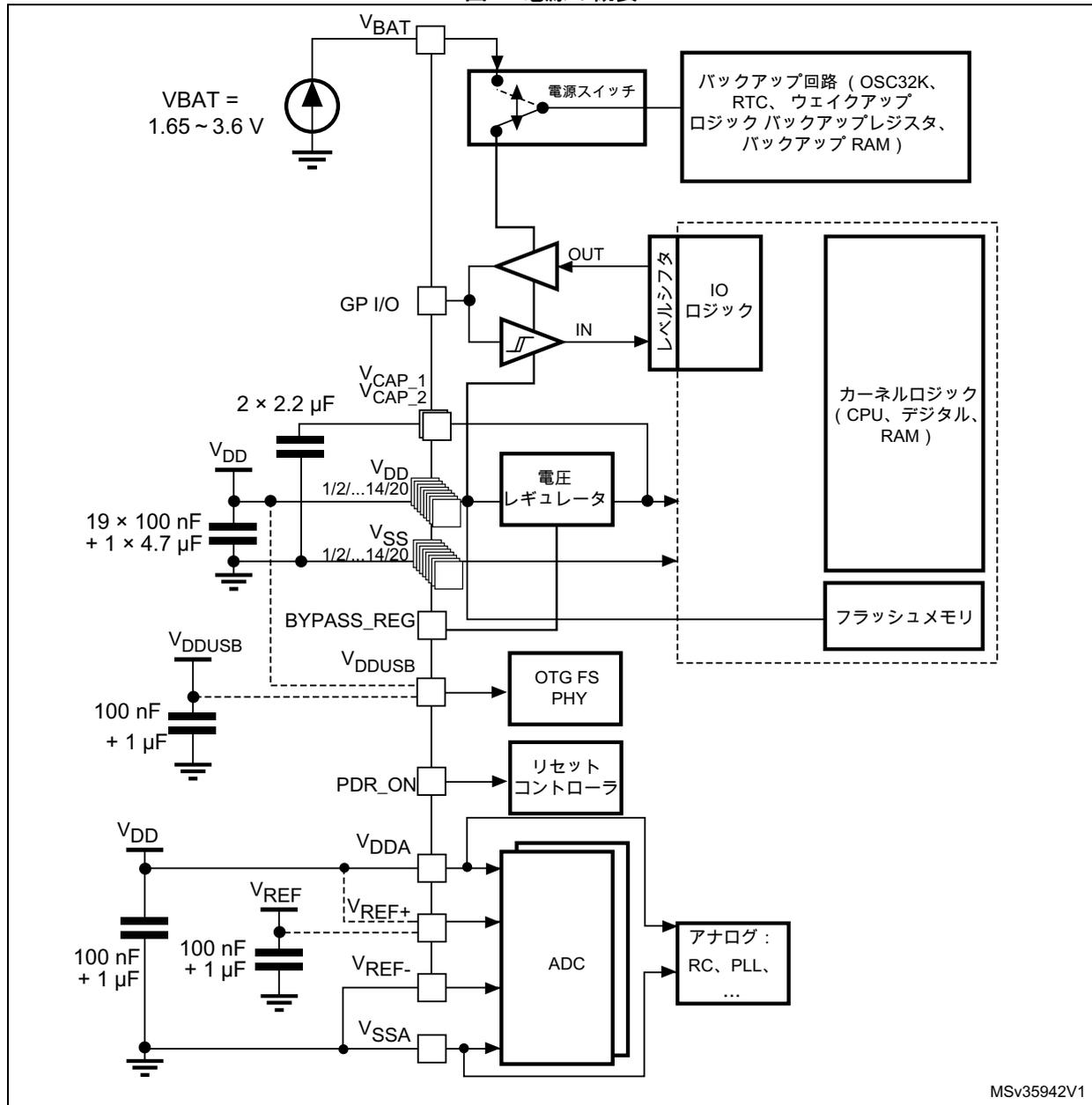
### 4.1 電源

このデバイスには、1.8~3.6 V の動作電源 ( $V_{DD}$ ) が必要です。内部の 1.2 V デジタル電源を供給するために、リニア電圧レギュレータが搭載されています。

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

**注：** *動作電源範囲に応じて、一部のペリフェラルは限定された機能と性能で使用されます。詳細については、STM32F75xxx および STM32F74xxx データシートの一般動作条件のセクションを参照してください。*

図 4. 電源の概要



MSV35942V1

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

## 4.1.1 独立した A/D コンバータ用電源と基準電圧

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

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

低電圧入力時の精度を確保するため、独立した外部基準電圧を ADC の  $V_{REF}$  入力に接続できます。 $V_{REF}$  の電圧は、1.8 V から  $V_{DDA}$  の範囲になります。

## 4.1.2 独立した USB トランシーバの電源

$V_{DDUSB}$  はフルスピードトランシーバ用の独立した USB 電源 (USB OTG FS および FS モード時の USB OTG HS) です。 $V_{DD}$  または USB トランシーバ用の独立した外部電源 (3.0~3.6 V) に接続できます (図 5 および 図 6 を参照)。たとえば、デバイスに 1.8 V が供給されている場合、独立した 3.3 V の電源を  $V_{DDUSB}$  に接続することができます。 $V_{DDUSB}$  は、独立した電源に接続されている場合は  $V_{DD}$  や  $V_{DDA}$  から独立していますが、最後に供給され最初に切断される電源でなければなりません。 $V_{DDUSB}$  に関する次の条件を遵守してください。

- パワーオンフェーズ ( $V_{DD} < V_{DD\_MIN}$ )の間は、 $V_{DDUSB}$  は常に  $V_{DD}$  より低くなければならない。
- パワーダウンフェーズ ( $V_{DD} < V_{DD\_MIN}$ )の間は、 $V_{DDUSB}$  は常に  $V_{DD}$  より低くなければならない。
- $V_{DDUSB}$  の立ち上がりおよび立ち下がり時間レートの仕様を遵守すること。
- 動作モードフェーズでは、以下のように、 $V_{DDUSB}$  が  $V_{DD}$  より低い場合も高い場合もあり得る。
  - USB (USB OTG\_HS/OTG\_FS) を使用する場合、 $V_{DDUSB}$  によって電源が供給される関連 GPIO は、 $V_{DDUSB\_MIN}$  と  $V_{DDUSB\_MAX}$  の間で動作する。
  - $V_{DDUSB}$  が両方の USB トランシーバ (USB OTG\_HS および USB OTG\_FS) に電源供給する。アプリケーションで片方の USB トランシーバのみを使用する場合でも、もう一方の USB トランシーバに関連する GPIO は  $V_{DDUSB}$  により電源供給される。
  - USB (USB OTG\_HS/OTG\_FS) を使用しない場合、 $V_{DDUSB}$  によって電源供給されている関連の GPIO は、 $V_{DD\_MIN}$  と  $V_{DD\_MAX}$  の間で動作する。

図 5.  $V_{DDUSB}$  を  $V_{DD}$  電源へ接続した場合

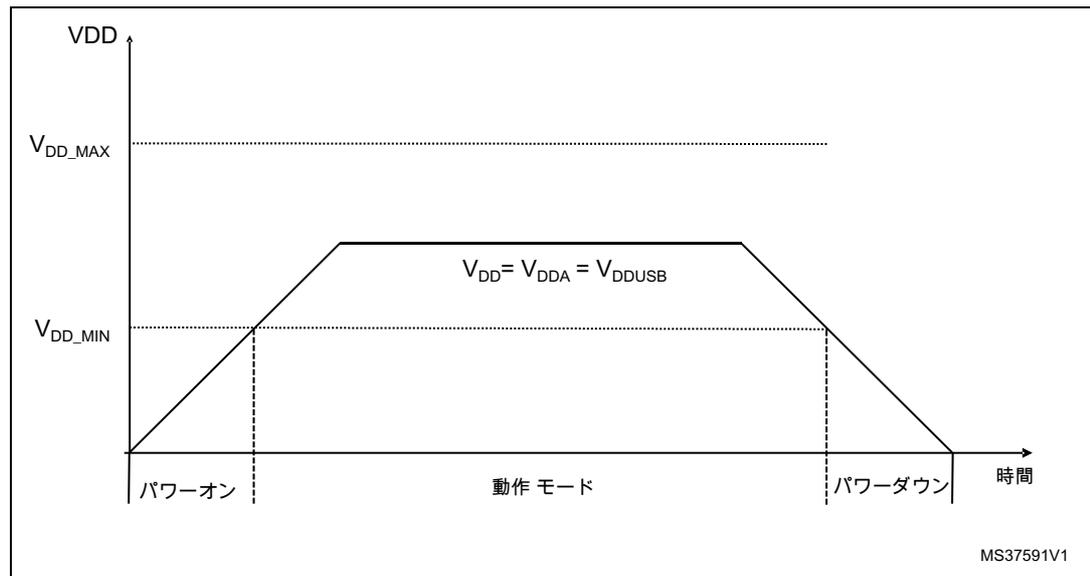
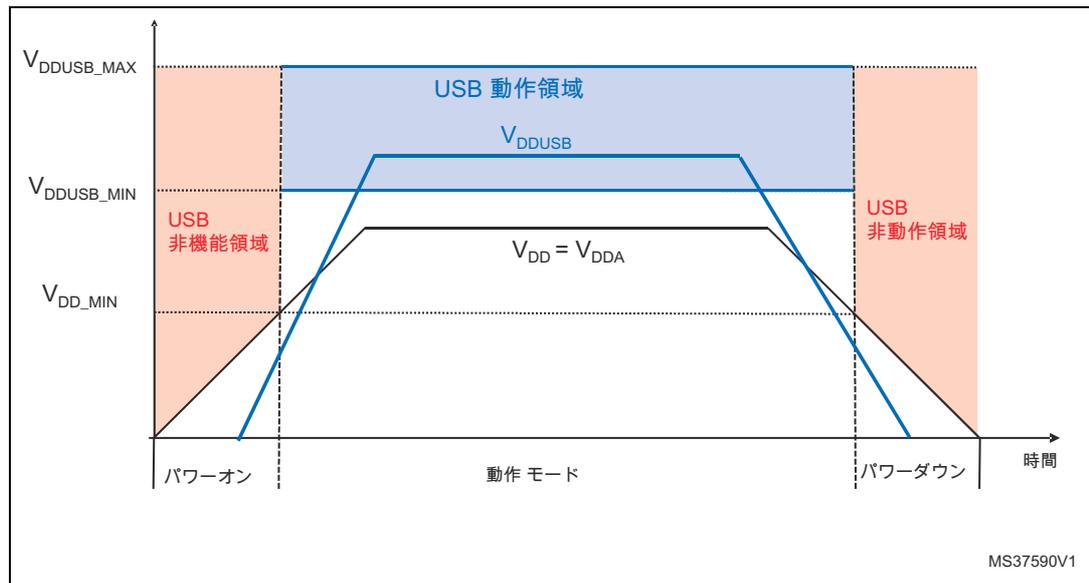


図 6.  $V_{DDUSB}$  を独立した外部電源に接続した場合



## 4.1.3 バッテリバックアップドメイン

### バックアップドメインの説明

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

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

- RTC
- LSE オシレータ
- 低電力バックアップレギュレータが動作状態のときのバックアップ SRAM
- PC13 から PC15 の I/O と PI8 の I/O (利用可能な場合)

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

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

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

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

- PC14 と PC15 は、GPIO または LSE ピンとして使用できます。
- PC13 と PI8 は GPIO ピンとして使用できます (これらのピンの設定に関する詳細は、[表 146 : RTC ピン PC13 の設定](#) および [表 147 : RTC ピン PI8 の設定](#) を参照)。

**注 :** **スイッチに流すことができる電流は限られているので (3 mA)、GPIO の PI8 および PC13 から PC15 までの使用には制限があります。出力として使用できるのは同時に 1 つの I/O のみで、最大負荷 30 pF で最大速度 2 MHz に制限する必要があります。また、これらの I/O を電流ソースとして使用することはできません (たとえば、LED を駆動するなど)。**

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

- PC14 と PC15 は、LSE ピンとしてのみ使用できます。
- PC13 はタンパピン (TAMP1) として使用できます。
- PI8 はタンパピン (TAMP2) として使用できます。

## バックアップドメインアクセス

リセット後、バックアップドメイン (RTC レジスタ、RTC バックアップレジスタ、バックアップ SRAM) は不要な書き込みアクセスの危険から保護されています。バックアップドメインへのアクセスを有効にするには、次の手順に従います。

- RTC および RTC バックアップレジスタへのアクセス
  1. RCC\_APB1ENR レジスタの PWREN ビットをセットして、電源インタフェースクロックを有効にします ([セクション 5.3.13](#)を参照)。
  2. [PWR 電源制御レジスタ \(PWR\\_CR1\)](#) の DBP ビットをセットして、バックアップドメインへのアクセスを有効にします。
  3. RTC クロックソースを選択します : [セクション 5.2.8 : RTC/AWU クロック](#)を参照。
  4. [RCC バックアップドメイン制御レジスタ \(RCC\\_BDCR\)](#) の RTCEN [15] ビットをプログラムすることで、RTC クロックを有効にします。
- バックアップ SRAM へのアクセス
  1. RCC\_APB1ENR レジスタの PWREN ビットをセットして、電源インタフェースクロックを有効にします ([セクション 5.3.13](#)を参照)。
  2. [PWR 電源制御レジスタ \(PWR\\_CR1\)](#) の DBP ビットをセットして、バックアップドメインへのアクセスを有効にします。
  3. [RCC APB1 ペリフェラルクロック有効レジスタ \(RCC\\_APB1ENR\)](#) の BKPSRAMEN ビットをセットして、バックアップ SRAM クロックを有効にします。

## RTC および RTC バックアップレジスタ

本リアルタイムクロック (RTC) は、独立した BCD タイマ/カウンタです。この RTC には、時刻クロック/カレンダー、プログラム可能なアラーム割り込みを 2 本、割り込み機能を備えたプログラム可能な周期的ウェイクアップフラグを搭載しています。この RTC には、32 個のバックアップデータレジスタ (128 バイト) を搭載しており、タンパ検出イベントが発生するとリセットされます。詳細については、[セクション 29 : リアルタイムクロック \(RTC\)](#) を参照してください。

## バックアップ SRAM

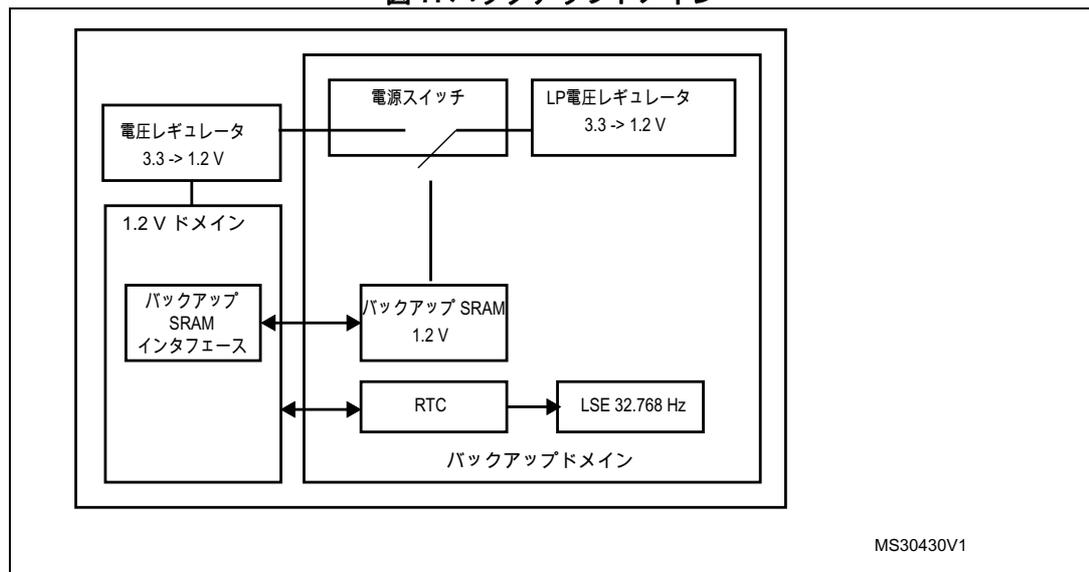
バックアップドメインには、32 ビット、16 ビット、8 ビットモードのいずれかでアドレス指定される 4 KB のバックアップ SRAM があります。低電力バックアップレギュレータが有効になっている場合、STANDBY モードや  $V_{BAT}$  モードであっても、その内容は保持されます。 $V_{BAT}$  が常に印加されている限り、バックアップ SRAM は、内蔵 EEPROM とみなすことができます。

バックアップドメインが  $V_{DD}$  から電力供給される場合（アナログスイッチが  $V_{DD}$  に接続された状態）、バッテリー寿命を延ばすため、バックアップ SRAM は、 $V_{BAT}$  の代わりに  $V_{DD}$  から電源供給されます。

バックアップドメインが  $V_{BAT}$  から電力供給を受けている場合（ $V_{DD}$  が印加されないため、アナログスイッチが  $V_{BAT}$  に接続された状態）、バックアップ SRAM は専用の低電力レギュレータにより電力供給されます。このレギュレータは、アプリケーションが STANDBY モードおよび  $V_{BAT}$  モードでバックアップ SRAM の機能を必要とするか否かに応じてオン/オフを切り替えることができます。このレギュレータのパワーダウンは、専用ビットである PWR\_CSR1 レジスタ（[セクション 4.4.2 : PWR 電源制御/ステータスレジスタ \(PWR\\_CSR1\)](#)を参照）の BRE 制御ビットにより制御されます。

バックアップ SRAM はタンパイイベントによって一括消去されることはありません。また、暗号の秘密鍵などの機密データへのアクセスを防ぐため、読み出し保護されています。バックアップ SRAM は、レベル 1 からレベル 0 への保護レベルの変更が要求されたときフラッシュインタフェースを介してのみ消去することができます。読み出し保護 (RDP) オプションバイトの説明を参照してください。

図 7. バックアップドメイン



### 4.1.4 電圧レギュレータ

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

この電圧レギュレータでは、2 つの外付けコンデンサを専用ピン  $V_{CAP\_1}$  と  $V_{CAP\_2}$  に接続する必要があり、これらのピンはすべてのパッケージに備わっています。電圧レギュレータを有効にしたり無効にしたりするには、ある特定のピンを  $V_{SS}$  または  $V_{DD}$  に接続する必要があります。これらのピンはパッケージによって異なります。

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

- **RUN モード**では、メインレギュレータは 1.2 V ドメイン (コア、メモリ、デジタルペリフェラル) にフル電力を供給します。このモードでは、レギュレータの出力電圧 (約 1.2 V) をソフトウェアによって、いくつかの異なる電圧値にスケールリングできます (スケール 1、スケール 2、スケール 3 が PWR\_CR1 レジスタの VOS[1:0] ビットによって設定可能)。この電圧スケールは、PLL がオフで HSI または HSE クロックソースがシステムクロックソースとして選択されている場合にのみ変更することができます。新しくプログラムされた電圧値は、PLL がオンの場合にのみアクティブになります。PLL がオフの場合、自動的に電圧スケール 3 が選択されます。この電圧スケールリングによって、デバイスが最大システム周波数より低速のクロックで動作しているときの消費電力を最適化することができます。STOP モードから復帰後、自動的に電圧スケール 3 が選択されます ([セクション 4.4.1 : PWR 電源制御レジスタ \(PWR\\_CR1\)](#) を参照)。2 つの動作モードがあります。
  - **通常モード** : CPU とコアロジックが所定の電圧スケールリング (スケール 1、スケール 2、スケール 3 のいずれか) にて最大周波数で動作します。
  - **オーバードライブモード** : このモードでは、CPU とコアロジックが所定の電圧スケールリング (スケール 1、スケール 2 またはスケール 3) にて通常モードより高い周波数で動作することができます。
- **STOP モード**では、メインレギュレータまたは低電力レギュレータが 1.2 V ドメインに低電力の電圧を供給して、レジスタと内蔵 SRAM の内容を保持します。電圧レギュレータは、メインレギュレータモード (MR) か、低電力モード (LPR) のいずれかに設定できます。両方のモードとも次のようにソフトウェアで設定できます。
  - **通常モード** : 1.2 V ドメインは通常リーク電流モードに置かれます。これは、メインレギュレータ (MR) または低電力レギュレータ (LPR) が有効になっているときのデフォルトモードです。
  - **アンダードライブモード** : 1.2 V ドメインは低リーク電流モードに置かれます。このモードは、メインレギュレータまたは低電力レギュレータが低電圧モードのときにのみ使用できます ([表 11](#)を参照)。
- **STANDBY モード**では、レギュレータは停止状態になります。STANDBY 回路とバックアップドメインを除き、レジスタと SRAM の内容は失われます。

**注 :** レギュレータがバイパスされたときにはオーバードライブモードとアンダードライブモードは使用できません。

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

表 11. 電圧レギュレータ設定モードとデバイス動作モード<sup>(1)</sup>

電圧レギュレータ設定	RUN モード	SLEEP モード	STOP モード	STANDBY モード
通常モード	MR	MR	MR または LPR	-
オーバードライブモード <sup>(2)</sup>	MR	MR	-	-
アンダードライブモード	-	-	MR または LPR	-
パワーダウンモード	-	-	-	あり

1. “-” は対応する設定が利用できないことを意味します。
2. V<sub>DD</sub> = 1.8~2.1 V のときオーバードライブモードは使用できません。

## オーバードライブモードへの移行

アプリケーションが重要なタスクの実行ではなく、システムクロックソースが HSI または HSE のいずれかであるときに、オーバードライブモードに切り替えることを推奨します。設定時間を最適化するには、PLL ロックフェーズ中にオーバードライブモードを有効にします。

オーバードライブモードを開始するには、次の手順に従います。

1. システムクロックとして HSI または HSE を選択します。
2. RCC\_PLLCFGR レジスタを設定し、RCC\_CR レジスタの PLLON ビットをセットします。
3. PWR\_CR1 レジスタの ODEN ビットをセットしてオーバードライブモードを有効にし、PWR\_CSR1 レジスタの ODRDY フラグがセットされるのを待ちます。
4. PWR\_CR1 レジスタの ODSW ビットをセットして、電圧レギュレータを通常モードからオーバードライブモードに切り替えます。システムは切り替え時に一時停止しますが、PLL クロックシステムはロックフェーズ中でも動作し続けます。
5. PWR\_CSR1 の ODSWRDY フラグがセットされるのを待ちます。
6. 必要なフラッシュ待ち時間および AHB プリスケラと APB プリスケラを選択します。
7. PLL ロックを待ちます。
8. システムクロックを PLL に切り替えます。
9. システム PLL によって生成されていないペリフェラルクロック (I2S クロック、LCD-TFT クロック、SAI1 クロック、USB\_48 MHz クロックなど) を有効にします。

**注：** *PLLI2S と PLLSAI はシステム PLL と同時に設定することができません。*

*オーバードライブスイッチのアクティブ化中は、どのペリフェラルクロックも有効にしてはなりません。オーバードライブモードを有効にした後、ペリフェラルクロックを有効にする必要があります。*

*STOP モードに入ると、オーバードライブモードと PLL が無効になります。STOP モードからの復帰後は、アプリケーションソフトウェアによって、オーバードライブモードと PLL を再設定する必要があります。*

## オーバードライブモードの終了

アプリケーションが重要なタスクの実行中ではなく、システムクロックソースが HSI または HSE のいずれかであるときに、オーバードライブモードを終了することを推奨します。オーバードライブモードを終了する手順は 2 つあります。

- PWR\_CR1 レジスタの ODEN と ODSW ビットを同時にリセットします (手順 1)。
- まず ODSW ビットをリセットして電圧レギュレータを通常モードへ切り換え、その後、ODEN ビットをリセットしてオーバードライブモードを無効にします (手順2)。

手順 1 の例

1. システムクロックソースとして HSI または HSE を選択します。
2. システム PLL によって生成されていないペリフェラルクロック (I2S クロック、LCD-TFT クロック、SAI1 クロック、USB\_48 MHz クロックなど) を無効にします。
3. PWR\_CR1 レジスタの ODEN ビットと ODSW ビットを同時にリセットして、電圧レギュレータを通常モードに戻し、オーバードライブモードを無効にします。
4. PWR\_CSR1 レジスタの ODWRDY フラグがリセットされるのを待ちます。

手順 2 の例

1. システムクロックソースとして HSI または HSE を選択します。
2. システム PLL によって生成されていないペリフェラルクロック (I2S クロック、LCD-TFT クロック、SAI1 クロック、USB\_48 MHz クロックなど) を無効にします。
3. PWR\_CR1 レジスタの ODSW ビットをリセットして、電圧レギュレータを通常モードに戻します。電圧切り替え中、システムクロックは一時停止します。
4. PWR\_CSR1 レジスタの ODWRDY フラグがリセットされるのを待ちます。
5. PWR\_CR1 レジスタの ODEN ビットをリセットして、オーバードライブモードを無効にします。

**注:** ステップ 3 の実行中、ODEN ビットはセットされたままになり、オーバードライブモードは有効なままになりますが、アクティブではありません (ODSW ビットがリセット状態)。ODEN ビットが代わりにリセットされると、オーバードライブモードが無効になり、電圧レギュレータは初期電圧に戻ります。

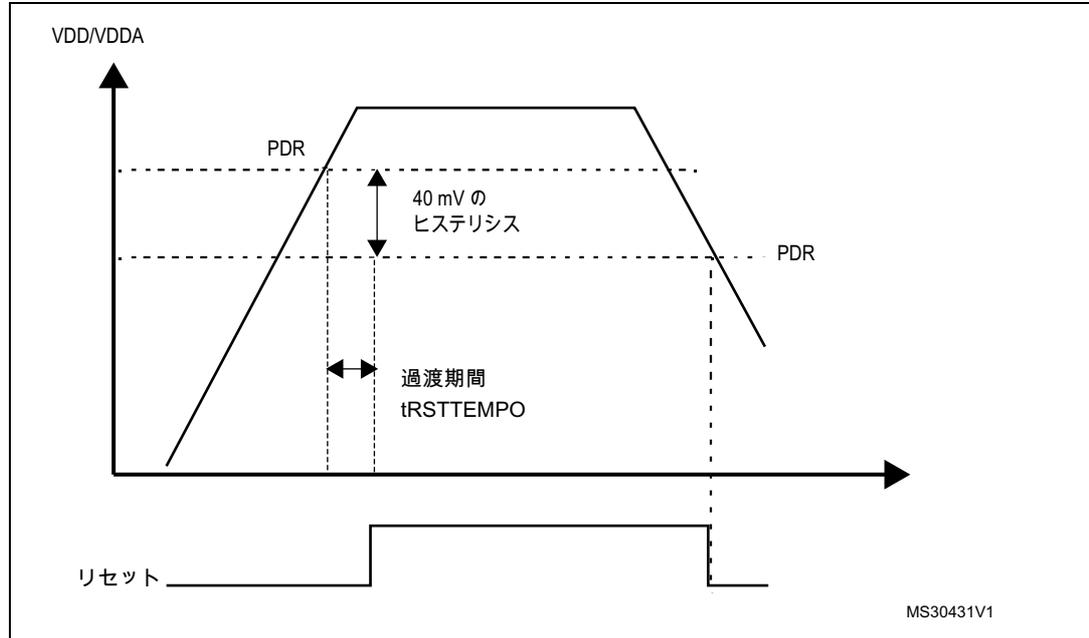
## 4.2 電源供給スーパバイザ

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

このデバイスには、1.8 V 以上での正常な動作を可能にする POR/PDR 統合回路が搭載されています。

デバイスは、 $V_{DD}/V_{DDA}$  が規定の閾値  $V_{POR/PDR}$  を下回っている間は、外部のリセット回路を必要とせず、リセットモードを維持します。POR/PDR 閾値の詳細については、データシートの電氣的特性の項を参照してください。

図 8. パワーオンリセット / パワーダウンリセット波形



## 4.2.2 ブラウンアウトリセット (BOR)

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

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

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

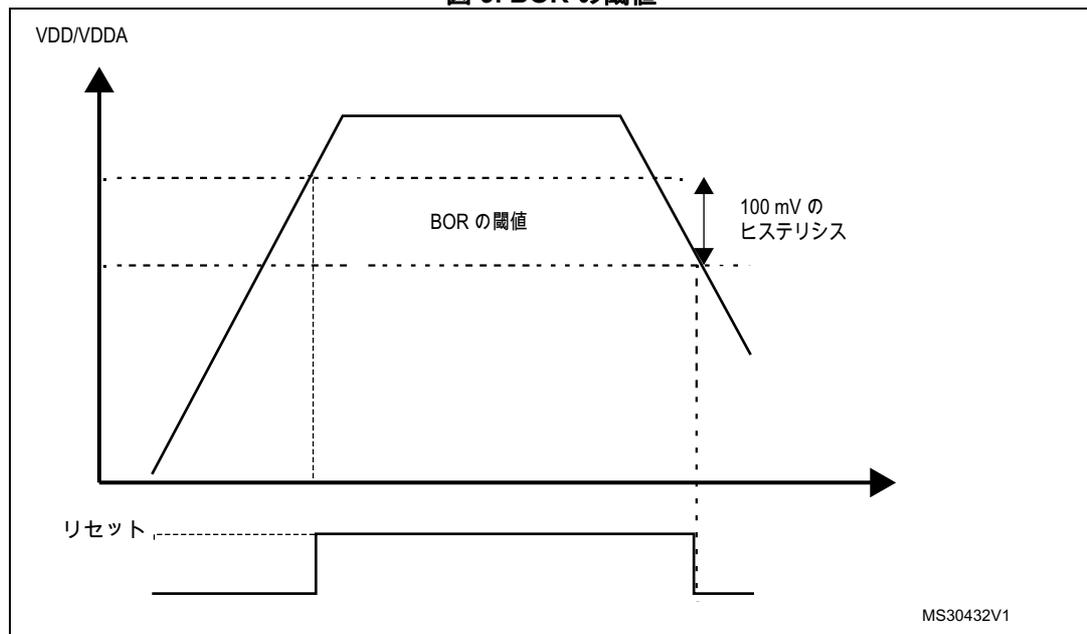
**注：** **BOR 特性の詳細については、デバイスデータシートの「電気的特性」のセクションを参照してください。**

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

BOR は、デバイスのオプションバイトをプログラミングすることで無効にすることができます。この場合、パワーオンとパワーダウンは POR/PDR によって監視されます ([セクション 4.2.1：パワーオンリセット \(POR\) /パワーダウンリセット \(PDR\)](#) を参照)。

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

図 9. BOR の閾値



## 4.2.3 プログラム可能な電圧検出器 (PVD)

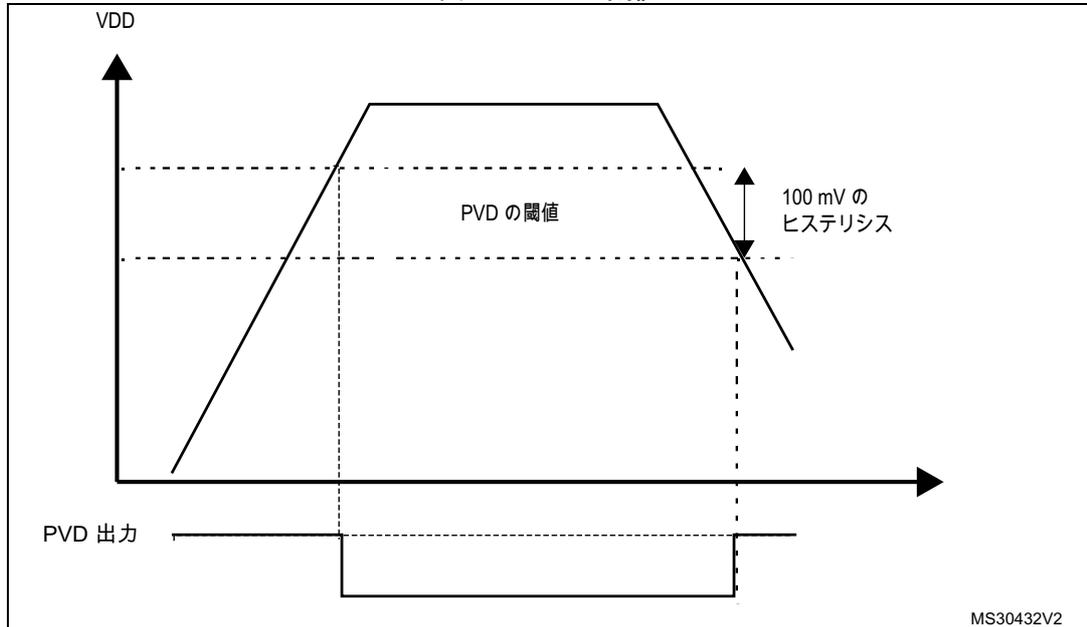
PVD を使用した  $V_{DD}$  電源の監視が可能です。[PWR 電源制御レジスタ \(PWR\\_CR1\)](#) の PLS[2:0] ビットで選択した閾値と比較することで監視します。

PVD は PVDE ビットをセットすることで有効になります。

[PWR 電源制御/ステータスレジスタ \(PWR\\_CSR1\)](#) の PVDO フラグを使用して、 $V_{DD}$  が PVD の閾値よりも高いか低いかを示すことができます。このイベントは EXTI のライン 16 に内部接続され、EXTI レジスタで有効な場合は割り込みリクエストを生成させることができます。EXTI ライン 16 の立ち上がり/立ち下がりエッジの設定に応じて、 $V_{DD}$  が PVD の閾値を下回るか、上回ったとき、あ

るいはその両方で、PVD 出力割り込みを生成させることができます。たとえば、サービスルーチンで、緊急停止処理を実行することなどが可能です。

図 10. PVD の閾値



## 4.3 低電力モード

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

このデバイスは、次の 3 つの低電力モードを備えています。

- SLEEP モード (Cortex<sup>®</sup>-M7 コアは停止、ペリフェラルは動作状態を維持)
- STOP モード (すべてのクロックが停止)
- STANDBY モード (1.2 V ドメインの電源オフ)

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

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

### 低電力モードへの移行

低電力モードへの移行は、MCU が WFI (Wait For Interrupt) または WFE (Wait For Event) 命令を実行することにより、または Cortex<sup>®</sup>-M7 システム制御レジスタの SLEEPONEXIT ビットが割り込みサービスルーチン (ISR) 終了時にセットされている時点で実行されます。

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

## 低電力モードの終了

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

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

– NVIC IRQ 割り込み :

Cortex<sup>®</sup>-M7 システム制御レジスタの SEVONPEND = 0 の場合 : ペリフェラル制御レジスタおよび NVIC にて割り込みを有効にすることによって行います。MCU が WFE からリスタートするときには、(NVIC 割り込みクリアペンディングレジスタの) ペリフェラル割り込みペンディングビットと NVIC ペリフェラル IRQ チャネルペンディングビットをクリアする必要があります。十分な優先度の NVIC 割り込みのみが MCU をウェイクアップし、割り込みを生成します。

Cortex<sup>®</sup>-M7 システム制御レジスタの SEVONPEND = 1 の場合 : ペリフェラル制御レジスタ (および任意で NVIC の割り込み) にて割り込みを有効にすることによって行います。MCU が WFE からリスタートするときには、(NVIC 割り込みクリアペンディングレジスタの) ペリフェラル割り込みペンディングビットと、有効であれば NVIC ペリフェラル IRQ チャネルペンディングビットをクリアする必要があります。すべての NVIC 割り込み (無効化されているものも含む) が MCU をウェイクアップします。十分な優先度の有効化された NVIC 割り込みのみが MCU をウェイクアップし、割り込みを生成します。

– イベント

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

MCU は、外部リセット (NRST ピン)、IWDG リセット、有効な WKUPx ピンのうちの 1 つの立ち上がりエッジ、または RTC イベントのいずれかが発生すると、低電力の STANDBY モードを終了します (図 282 : RTC ブロック図を参照)。

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

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

表 12. 低電力モードの概要

モード名	エントリ	ウェイクアップ	1.2 V ドメインロックへの影響	V <sub>DD</sub> ドメインクロックへの影響	電圧レギュレータ
SLEEP (Sleep-now または Sleep-on-exit)	WFI	割り込み	CPU クロックオフ 他のクロック およびアナログ クロックソース への影響なし	なし	ON
	WFE	ウェイクアップイベント			

表 12. 低電力モードの概要 (続き)

モード名	エントリ	ウェイクアップ	1.2 V ドメイン ロックへの影響	V <sub>DD</sub> ドメイン クロック への影響	電圧レギュレータ
STOP	SLEEPDEEP ビット + WFI または WFE	任意の EXTI ライン (EXTI レジスタで設定、内部 ラインおよび外部ライン)	すべての 1.2 V ドメインクロック オフ	HSI および HSE オシ レータオフ	メインレギュレータ または低電力 レギュレータ (PWR 電源制御レジスタ (PWR_CR1) に 応じて)
STANDBY	PDDS ビット + SLEEPDEEP ビット + WFI または WFE	WKUP ピンの立ち上がり / 立ち下がりエッジ、 RTC アラーム (アラーム A またはアラーム B)、RTC ウェイクアップイベント、 RTC タンパイイベント、RTC タイムスタンプイベント、 NRST ピンによる外部 リセット、IWDG リセット			オフ

表 13. すべてのモードでの機能 (1)

ペリフェラル	RUN	SLEEP	STOP		STANDBY		VBAT
				ウェイクアップ		ウェイクアップ	
CPU	Y	-	-		-		-
フラッシュアクセス	Y	Y	-		-		-
DTCM RAM	Y	Y	Y		-		
ITCM RAM	Y	Y	Y		-		
SRAM1	Y	Y	Y		-		-
SRAM2	Y	Y	Y		-		-
FMC	O	O	-		-		-
Quad SPI	O	O	-		-		-
バックアップレジスタ	Y	Y	Y		Y		Y
バックアップ RAM	Y	Y	Y		Y		Y
ブラウンアウトリセット (BOR)	Y	Y	Y	Y	Y	Y	
プログラム可能な電圧検出器 (PVD)	O	O	O	O	-		-
高速内部 (HSI)	O	O	(2)		-		-
高速外部 (HSE)	O	O	-				-
低速内部 (LSI)	O	O	O		O		-
低速外部 (LSE)	O	O	O		O		O

表 13. すべてのモードでの機能 (続き)<sup>(1)</sup>

ペリフェラル	RUN	SLEEP	STOP		STANDBY		VBAT
				ウェイクアップ		ウェイクアップ	
RTC	0	0	0	0	0	0	0
RTC タンパピンの数	3	3	3	3	3	3	2
CRC 計算ユニット	0	0	-		-		-
GPIO	Y	Y	Y	Y		ピン : 6本	タンパ : 2本
DMA	0	0	-		-		-
Chrom-ART アクセラレータ (DMA2D)	0	0	-		-		-
LCD-TFT	0	0	-	-	-		-
DCMI	0	0	-		-		-
USARTx (x = 1~8)	0	0	-		-		-
I2Cx (x = 1, 2, 3, 4)	0	0	-		-		-
SPIx (x = 1~6)	0	0	-		-		-
SAIx (x = 1, 2)	0	0	-		-		-
SPDIFRX	0	0	-		-		-
ADCx (x = 1, 2, 3)	0	0	-		-		-
DACx (x = 1, 2)	0	0	-		-		-
温度センサ	0	0	-		-		-
タイマ (TIMx)	0	0	-		-		-
低電力タイマ 1 (LPTIM1)	0	0	0	0	-		-
独立型ウォッチドッグ (IWDG)	0	0	0	0	0	0	-
ウィンドウ型ウォッチドッグ (WWDG)	0	0	-		-		-
SysTick タイマ	0	0	-		-		-
乱数発生器 (RNG)	0	0	-		-		-
暗号プロセッサ (CRYP)							
ハッシュプロセッサ (HASH)	0	0	-		-		-
SDMMC	0	0	-		-		-
CANx (x = 1, 2)	0	0	-		-		-
USB OTG FS	0	0	-		-		-
USB OTG HS	0	0	-		-		-
Ethernet	0	0	-		-		-
HDMI-CEC	0	0	-		-		-

1. 凡例 : Y = あり (有効) 0 = 任意(デフォルトでは無効だが、ソフトウェアで有効化できる) - = 該当なし

- STOP モードからのウェイクアップ機能を持つ一部のペリフェラルは、HSI の有効化をリクエストできます。この場合、HSI はペリフェラルによってウェイクアップされ、HSI をリクエストしたペリフェラルだけにクロックを供給します。ペリフェラルが HSI を必要としなくなると、HSI は自動的にオフされます。

## 4.3.1 デバッグモード

デフォルトでは、デバッグ機能が使用されているときにデバイスが STOP または STANDBY モードに移行すると、デバッグ接続は失われます。これは、Cortex<sup>®</sup>-M7 コアにクロックが供給されなくなるためです。

ただし、DBGMCU\_CR レジスタの設定ビットをセットすることによって、低電力モードを多用しているときでも、ソフトウェアのデバッグを行うことができます。詳細については、[セクション 40.16.1: 低電力モードのデバッグサポート](#)を参照してください。

## 4.3.2 RUN モード

### システムクロックの低速化

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

詳細については、[セクション 5.3.3: RCC クロック設定レジスタ \(RCC\\_CFGR\)](#) を参照してください。

### ペリフェラルクロックゲーティング

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

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

ペリフェラルへのクロックゲーティングは、AHB1 ペリフェラルクロック有効レジスタ (RCC\_AHB1ENR)、AHB2 ペリフェラルクロック有効レジスタ (RCC\_AHB2ENR)、AHB3 ペリフェラルクロック有効レジスタ (RCC\_AHB3ENR) によって制御されます ([セクション 5.3.10: RCC AHB1 ペリフェラルクロックレジスタ \(RCC\\_AHB1ENR\)](#)、[セクション 5.3.11: RCC AHB2 ペリフェラルクロック有効レジスタ \(RCC\\_AHB2ENR\)](#)、および [セクション 5.3.12: RCC AHB3 ペリフェラルクロック有効レジスタ \(RCC\\_AHB3ENR\)](#) を参照)。

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

## 4.3.3 低電力モード

### 低電力モードへの移行

低電力モードへの移行は、MCU が WFI (Wait For Interrupt) または WFE (Wait For Event) 命令を実行することにより、または Cortex<sup>®</sup>-M7 システム制御レジスタの SLEEPONEXIT ビットが割り込みサービスルーチン (ISR) 終了時にセットされている時点で実行されます。

## 低電力モードの終了

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

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

- NVIC IRQ 割り込み

- Cortex<sup>®</sup>-M7 システム制御レジスタの SEVEONPEND = 0 の場合

ペリフェラル制御レジスタおよび NVIC にて割り込みを有効にすることで行います。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) NVIC ペリフェラル IRQ チャネルペンディングビットをクリアする必要があります。

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

- Cortex<sup>®</sup>-M7 システム制御レジスタの SEVEONPEND = 1 の場合

ペリフェラル制御レジスタ (および 任意で NVIC の割り込み) にて割り込みを有効にすることで行います。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと、有効であれば (NVIC 割り込みクリアペンディングレジスタの) NVIC ペリフェラル IRQ チャネルペンディングビットをクリアする必要があります。

すべての NVIC 割り込み (無効化されているものも含む) によって MCU をウェイクアップします。

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

- イベント

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

MCU は、外部リセット (NRST ピン)、IWDG リセット、有効な WKUPx ピンのうちの 1 つの立ち上がりエッジ、または RTC イベントのいずれかが発生すると、低電力の STANDBY モードを終了します ( [図 282 : RTC ブロック図](#) を参照)。

## 4.3.4 SLEEP モード

### SLEEP モードにおける I/O の状態

SLEEP モードで、すべての I/O ピンは RUN モードと同じ状態を保持します。

### SLEEP モードへの移行

WFI (Wait For Interrupt) 命令または WFE (Wait for Event) 命令を実行することにより、SLEEP モードに移行します。SLEEP モードへの移行方法として、Cortex<sup>®</sup>-M7 システム制御レジスタの SLEEPONEXIT ビットによって、2つのオプションが選択できます。

- Sleep-now : SLEEPONEXIT ビットがクリアされている場合、MCU は、WFI または WFE 命令の実行直後に SLEEP モードに移行します。
- Sleep-on-exit : SLEEPONEXIT ビットがセットされている場合、MCU は、最も優先度の低い ISR の終了直後に SLEEP モードに移行します。

SLEEP モードへの移行方法の詳細については、表 14 および表 15 を参照してください。

### SLEEP モードの終了

WFI 命令によって SLEEP モードに移行した場合、ペリフェラルの割り込みがネスト化されたベクタ割り込みコントローラ (NVIC) によって認識されると、デバイスは SLEEP モードを終了することができます。

WFE 命令によって SLEEP モードに移行した場合、MCU はイベントの発生直後に SLEEP モードを終了します。ウェイクアップイベントは、次のいずれかによって生成できます。

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、Cortex<sup>®</sup>-M7 システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードに設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャネルペンディングビットをクリアする必要はありません。

このモードは、割り込みの入口と出口で余分な時間を必要としないため、ウェイクアップ時間が最も短くなります。

SLEEP モードの終了方法の詳細については、表 14 および表 15 を参照してください。

表 14. Sleep-now への移行と終了

Sleep-now モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) - SLEEPDEEP = 0、および - ペンディング状態の割り込み (WFI) やイベント (WFE) なし Cortex <sup>®</sup> -M7 システム制御レジスタを参照。
	次の条件下での割り込みサービスルーチン (ISR) 終了時 - SLEEPDEEP = 0 および - SLEEPONEXIT = 1、 - ペンディング状態の割り込みなし Cortex <sup>®</sup> -M7 システム制御レジスタを参照。

表 14. Sleep-now への移行と終了 (続き)

Sleep-now モード	説明
モードの終了	<p>WFI または割り込みサービスルーチン (ISR) 終了を使用して移行した場合                      割り込み: <a href="#">表 43: STM32F75xxx および STM32F74xxx ベクタテーブル</a> を参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 0 の場合：                      ウェイクアップイベント: <a href="#">セクション 11.3: ウェイクアップイベント管理</a> を参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 1 の場合：                      割り込みイベント (NVIC で無効化されている場合も含む) (<a href="#">表 43: STM32F75xxx および STM32F74xxx ベクタテーブル</a> を参照) およびウェイクアップイベント (<a href="#">セクション 11.3: ウェイクアップイベント管理</a> を参照)</p>
ウェイクアップ遅延時間	なし

表 15. Sleep-on-exit への移行と終了

Sleep-on-exit	説明
モードへの移行	<p>次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event)</p> <ul style="list-style-type: none"> <li>- SLEEPDEEP = 0、および</li> <li>- ペンディング状態の割り込み (WFI) やイベント (WFE) なし</li> </ul> <p>Cortex<sup>®</sup>-M7 システム制御レジスタを参照。</p> <p>次の条件下での割り込みサービスルーチン (ISR) 終了時</p> <ul style="list-style-type: none"> <li>- SLEEPDEEP = 0、</li> <li>- SLEEPONEXIT = 1、および</li> <li>- ペンディング状態の割り込みなし</li> </ul> <p>Cortex<sup>®</sup>-M7 システム制御レジスタを参照。</p>
モードの終了	割り込み: <a href="#">表 43: STM32F75xxx および STM32F74xxx ベクタテーブル</a> を参照。
ウェイクアップ遅延時間	なし

### 4.3.5 STOP モード

STOP モードは、ペリフェラルクロックゲーティングと組み合わせられた Cortex<sup>®</sup>-M7 のディープスリープ (deepsleep) モードに準拠しています。電圧レギュレータは、通常モードまたは低電力モードのいずれかに設定できます。STOP モードでは、1.2 V ドメインのすべてのクロックが停止し、PLL、HSI、HSE RC オシレータが停止します。内蔵 SRAM とレジスタの内容は保持されます。

STOP モードでは、PWR\_CR1 レジスタの追加設定を使用することにより、消費電力をさらに低減することができます。ただし、STOP モードからのウェイクアップの際に起動時間が長くなる原因になります ([表 16](#)を参照)。

表 16. STOPモードの動作モード

電圧レギュレータモード		UDEN[1:0] ビット	MRUDS ビット	LPUDS ビット	LPDS ビット	FPDS ビット	ウェイクアップ遅延時間
通常 モード	STOP MR (メインレギュ レータ)	-	0	-	0	0	HSI RC 起動時間
	STOP MR-FPD	-	0	-	0	1	HSI RC 起動時間 + パワーダウンモードからの フラッシュのウェイクアップ時間
	STOP LP	-	0	0	1	0	HSI RC 起動時間 + LP モードからのレギュレータの ウェイクアップ時間
	STOP LP-FPD	-	-	0	1	1	HSI RC 起動時間 + パワーダウンモードからの フラッシュのウェイクアップ時間+ LP モードからのレギュレータの ウェイクアップ時間
アンダー ドライブ モード	STOP UMR-FPD	3	1	-	0	-	HSI RC 起動時間 + パワーダウンモードからの フラッシュのウェイクアップ時間+ アンダードライブモードからの メインレギュレータの ウェイクアップ時間 + コアロジック の通常モードへの遷移時間
	STOP ULP-FPD	3	-	1	1	-	HSI RC 起動時間 + パワーダウンモードからの フラッシュのウェイクアップ時間+ LP アンダードライブモードからの レギュレータのウェイクアップ時間+ コアロジックの通常モードへの 遷移時間

## STOP モードにおける I/O の状態

STOP モードでは、すべての I/O ピンは RUN モードと同じ状態を保持します。

## STOP モードへの移行

STOP モードへの移行は、Cortex<sup>®</sup>-M7 のシステム制御レジスタの SLEEPDEEP ビットがセットされている場合に、*低電力モードへの移行*に従って実行されます。

STOP モードへの移行方法の詳細については、表 17 を参照してください。

マイクロコントローラが STOP モードに移行すると、自動的に電圧スケール 3 が選択されます。STOP モードで消費電力をさらに低減するには、内蔵電圧レギュレータを低電力モードまたは低電圧モードにします。この設定は *PWR 電源制御レジスタ (PWR\_CR1)* の LPDS、MRUDS、LPUDS、および UDEN ビットで行います。

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

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



STOP モードに入る前にオーバードライブモードを有効にした場合でも、STOP モードが有効である間、オーバードライブモードは自動的に無効になります。

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

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。 [セクション 27 : 独立型ウォッチドッグ \(IWDG\) のセクション 27.3](#) を参照してください。
- リアルタイムクロック (RTC) : この設定は [RCC バックアップドメイン制御レジスタ \(RCC\\_BDCR\)](#) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は [RCC クロック制御およびステータスレジスタ \(RCC\\_CSR\)](#) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は [RCC バックアップドメイン制御レジスタ \(RCC\\_BDCR\)](#) の LSEON ビットで行います。

ADC や DAC は、STOP モードに移行する前に停止させない限り、STOP モード中にも電力を消費します。ADC および DAC を停止させるには、ADC\_CR2 レジスタの ADON ビットと DAC\_CR レジスタの ENx ビットの両方に 0 を書き込む必要があります。

**注 :** *外部オシレータ (HSE) 障害による内蔵 MCU の動作への影響を防ぐため、STOP モードに入る前に、クロックセキュリティシステム (CSS) 機能を有効にすることを推奨します。*

## STOP モードの終了

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

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

アンダードライブモードを有効にした場合でも、STOP モードの終了後、アンダードライブモードは自動的に無効になります。

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

電圧レギュレータがアンダードライブモードで動作している場合、STOP モードからのウェイクアップの際、更なる起動時間が必要になります。

表 17. STOP モードへの移行と終了 (STM32F75xxx および STM32F74xxx)

STOP モード	説明
モードへの移行	<p>次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event)</p> <ul style="list-style-type: none"> <li>- ペンディング状態の割り込み (WFI) やイベント (WFE) なし</li> <li>- Cortex®-M7 システム制御レジスタの SLEEPDEEP ビットをセット</li> <li>- 電源制御レジスタ (PWR_CR) の PDDS ビットをクリア</li> <li>- PWR_CR レジスタの LPDS、MRUDS、LPUDS、UDEN ビットを設定することによって電圧レギュレータのモードを選択 (表 16 : STOP モードの動作モードを参照)</li> </ul>
	<p>次の条件下での割り込みサービスルーチン (ISR) 終了時</p> <ul style="list-style-type: none"> <li>- ペンディング状態の割り込みなし、</li> <li>- Cortex®-M7 システム制御レジスタの SLEEPDEEP ビットをセット、</li> <li>- SLEEPONEXIT = 1、および</li> <li>- PWR_CR1 の PDDS をクリア</li> </ul>
	<p><b>注:</b> STOP モードに移行するには、すべての EXTI ラインのペンディングビット (ペンディングレジスタ (EXTI_PR) 内)、すべてのペリフェラル割り込みのペンディングビット、RTC アラーム (アラーム A およびアラーム B)、RTC ウェイクアップ、RTC タンパ、RTC タイムスタンプフラグをリセットする必要があります。そうしないと、STOP モード移行手順が無視され、プログラムが実行され続けます。</p>
モードの終了	<p>WFI または Return from ISR を使用して移行した場合</p> <p>割り込みモードに設定されているすべての EXTI ライン (対応する EXTI 割り込みベクタが NVIC で有効になっている必要があります)。割り込みソースは、外部割り込みまたはウェイクアップ機能を備えたペリフェラルになることがあります。表 43 : STM32F75xxx および STM32F74xxx ベクタテーブル (287 ページ) を参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 0 の場合 :</p> <p>イベントモードに設定されているすべての EXTI ライン <a href="#">セクション 11.3 : ウェイクアップイベント管理 (293 ページ)</a> を参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 1 の場合 :</p> <ul style="list-style-type: none"> <li>- 割り込みモードに設定されている任意の EXTI ライン (対応する EXTI 割り込みベクタが NVIC で無効になっている場合も含む)。割り込みソースは、ウェイクアップ機能を備えた外部割り込みまたはペリフェラルになることがあります。表 43 : STM32F75xxx および STM32F74xxx ベクタテーブル (287 ページ) を参照してください。</li> <li>- ウェイクアップイベント : <a href="#">セクション 11.3 : ウェイクアップイベント管理 (293 ページ)</a> を参照。</li> </ul>
ウェイクアップ遅延時間	表 16 : STOP モードの動作モードを参照してください。

## 4.3.6 STANDBY モード

STANDBY モードでは、消費電力を最も少なくできます。このモードは、電圧レギュレータを無効にした状態の Cortex<sup>®</sup>-M7 のディープスリープ (deepsleep) モードに基づきます。結果として、1.2 V ドメインの電力がオフになります。PLL、HSI オシレータ、HSE オシレータもオフになります。バックアップドメインのレジスタ (RTC レジスタ、RTC バックアップレジスタ、バックアップ SRAM) と STANDBY 回路を除いて、SRAM とレジスタの内容は失われます (図 9 を参照)。

### STANDBY モードへの移行

STANDBY モードへの移行は、低電力モードへの移行に従って、Cortex<sup>®</sup>-M7 システム制御レジスタの SLEEPDEEP ビットがセットされている場合に実行されます。

STANDBY モードへの移行方法の詳細については、表 18 を参照してください。

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

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。参照: [セクション 27.3](#)、[セクション 27: 独立型ウォッチドッグ \(IWDG\)](#)
- リアルタイムクロック (RTC) : この設定は、バックアップドメイン制御レジスタ (RCC\_BDCR) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は、制御/ステータスレジスタ (RCC\_CSR) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は、バックアップドメイン制御レジスタ (RCC\_BDCR) の LSEON ビットで行います。

### STANDBY モードの終了

マイクロコントローラは、外部リセット (NRST ピン)、IWDG リセット、WKUP ピンの立ち上がり/立ち下がリエッジ、RTC アラーム、タンパイイベント、タイムスタンプイベントのいずれかが検出されると、STANDBY モードを終了します。[PWR 電源制御/ステータスレジスタ \(PWR\\_CSR1\)](#) を除くすべてのレジスタは、STANDBY モードからのウェイクアップ後にリセットされます。

STANDBY モードからのウェイクアップ後、プログラム実行はリセット後と同様にリスタートされます (ブートピン信号のサンプリング、ベクタリセットのフェッチなど)。[PWR 電源制御/ステータスレジスタ \(PWR\\_CSR1\)](#) の SBF ステータスフラグは MCU が STANDBY モードにあったことを示します。

STANDBY モードの終了方法の詳細については、表 18 を参照してください。

表 18. STANDBY モードへの移行と終了

STANDBY モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) - Cortex <sup>®</sup> -M7 システム制御レジスタの SLEEPDEEP ビットをセット、 - 電源制御レジスタ (PWR_CR) の PDDS ビットをセット、 - ペンディング状態の割り込み (WFI) やイベント (WFE) なし、 - 電源制御レジスタ (PWR_CR) の WUF ビットをクリア、 - 選択されたウェイクアップソース (RTC アラーム A、RTC アラーム B、RTC ウェイクアップ、タンパ、タイムスタンプフラグ) に対応する RTC フラグをクリア
	次の条件下での割り込みサービスルーチン (ISR) 終了時 - Cortex <sup>®</sup> -M7 システム制御レジスタの SLEEPDEEP ビットをセット、 - SLEEPONEXIT = 1、 - 電源制御レジスタ (PWR_CR) の PDDS ビットをセット、 - ペンディング状態の割り込みなし、 - 電源制御/ステータスレジスタ (PWR_SR) の WUF ビットをクリア、 - 選択されたウェイクアップソース (RTC アラーム A、RTC アラーム B、RTC ウェイクアップ、タンパ、タイムスタンプフラグ) に対応する RTC フラグをクリア
モードの終了	WKUP ピンの立ち上がり/立ち下がりエッジ、RTC アラーム (アラーム A およびアラーム B)、RTC ウェイクアップ、タンパイベント、タイムスタンプイベント、NRST ピンによる外部リセット、IWDG リセット
ウェイクアップ遅延時間	リセットフェーズ

## STANDBY モードにおける I/O の状態

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

- リセットパッド (有効)
- タンパ、タイムスタンプ、RTC アラーム出力、RTC クロック較正出力として設定されている場合の PC13
- 有効な場合、WKUP ピン (PA0/PA2/PC1/PC13/PI8/PI11)

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

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

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

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

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

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



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

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

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

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

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

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

## RTC 代替機能の安全なウェイクアップフラグクリアシーケンス

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

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

## 4.4 電源制御レジスタ

### 4.4.1 PWR 電源制御レジスタ (PWR\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000 C000 (STANDBY モードからのウェイクアップによってリセット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDEN[1:0]		ODSWEN	ODEN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VOS[1:0]		ADDC1	Res.	MRUDS	LPUDS	FPDS	DBP	PLS[2:0]			PVDE	CSBF	Res.	PDDS	LPDS
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rc_w1		rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 **UDEN[1:0]** : STOP モードにおけるアンダードライブ有効化

これらのビットは、ソフトウェアによってセットされます。これにより、STOP モードの消費電力を減らすことができますが、ウェイクアップ時間は長くなります。

このビットがセットされると、デバイスが STOP モードに移行したときのデジタル領域の漏れ電流が少なくなります。

00 : アンダードライブは無効

01 : 予約済み

10 : 予約済み

11 : アンダードライブは有効

ビット 17 **ODSWEN** : オーバードライブ切り替え有効化

このビットはソフトウェアでセットされます。このビットは、ODEN ビットがリセットされたとき、または STOP モード終了後にハードウェアによって自動的にクリアされます。このビットがセットされると、オーバードライブモードに切り替えることができます。

ODSWEN ビットをセットまたはリセットするには、HSI または HSE をシステムクロックとして選択する必要があります。

ODSWEN ビットは、オーバードライブモードに切り替えるため ODRDY フラグがセットされたときのみセットしなければなりません。

0 : オーバードライブ切り替えは無効

1 : オーバードライブ切り替えは有効

**注 :** *オーバードライブ切り替え時 (有効化であろうと無効化であろうと)、内部電圧の設定中は、システムクロックが一時停止します。*

ビット 16 **ODEN** : オーバードライブ有効化

このビットはソフトウェアでセットされます。このビットは、STOP モード終了後ハードウェアによって自動的にクリアされます。より高速に動作させることを目的として、オーバードライブモードを有効にするために使用します。

ODEN ビットをセットまたはリセットするには、HSI または HSE をシステムクロックとして選択する必要があります。ODEN ビットをセットした場合、アプリケーションは ODSWEN ビットをセットする前に、まずオーバードライブレディフラグ (ODRDY) がセットされるのを待つ必要があります。

0 : オーバードライブは無効

1 : オーバードライブは有効

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

このビットによって、メイン内蔵電圧レギュレータの出力電圧を制御し、デバイスが最大周波数で動作していないときに、パフォーマンスと消費電力のトレードオフを実現します（詳細は、STM32F75xxx および STM32F74xxx のデータシートを参照）。

これらのビットを変更できるのは、PLL がオフのときだけです。新しくプログラムされた電圧値は、PLL がオンの場合にのみアクティブになります。PLL がオフの場合、自動的に電圧スケール 3 が選択されます。

00 : 予約済み（スケール 3 モードが選択）

01 : スケール 3 モード

10 : スケール 2 モード

11 : スケール 1 モード（リセット値）

**ビット 13 ADCDC1** :

0 : 影響はありません。

1 : このビットの使い方の詳細は、AN4073 を参照してください。

**注 :** 電源電圧範囲 2.7~3.6 V で動作するときのみ、このビットをセットすることができます。

ビット 12 予約済みであり、リセット値に保持する必要があります。

**ビット 11 MRUDS** : ディープスリープアンダードライブモード時のメインレギュレータ

このビットは、ソフトウェアによってセット/クリアされます。

0 : デバイスが STOP モードのとき、メインレギュレータがオンになります。

1 : デバイスが STOP アンダードライブモードのとき、メインレギュレータはアンダードライブモードに入り、フラッシュメモリはパワーダウンとなります。

**ビット 10 LPUDS** : ディープスリープアンダードライブモード時の低電力レギュレータ

このビットは、ソフトウェアによってセット/クリアされます。

0 : デバイスが STOP モードのとき LPDS ビットがセットされていれば低電力レギュレータはオンになります。

1 : デバイスが STOP アンダードライブモードのとき LPDS ビットがセットされフラッシュメモリがパワーダウンの場合、低電力レギュレータはアンダードライブモードになります。

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

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

0 : デバイスが STOP モードのときフラッシュメモリはパワーダウンにはならない

1 : デバイスが STOP モードのときフラッシュメモリはパワーダウン

**ビット 8 DBP** : バックアップドメイン書き込み保護の無効化

リセット状態において、RCC\_BDCR レジスタ、RTC レジスタ（バックアップレジスタを含む）、PWR\_CSR1 レジスタの BRE ビットは、不要な書き込みアクセスから保護されています。これらのレジスタへの書き込みアクセスを可能にするには、このビットをセットする必要があります。

0 : RTC、RTC バックアップレジスタ、バックアップ SRAM へのアクセスは無効

1 : RTC、RTC バックアップレジスタ、バックアップ SRAM へのアクセスは有効

ビット 7:5 **PLS[2:0]** : PVD レベル選択

これらのビットは、電源電圧検出器によって検出される電圧閾値を選択するため、ソフトウェアで書き込みます。

000 : 2.0 V  
001 : 2.1 V  
010 : 2.3 V  
011 : 2.5 V  
100 : 2.6 V  
101 : 2.7 V  
110 : 2.8 V  
111 : 2.9 V

**注：** 詳細については、データシートの電気的特性を参照してください。

ビット 4 **PVDE** : 電源電圧検出器有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : PVD は無効です。  
1 : PVD は有効です。

ビット 3 **CSBF** : STANDBY フラグのクリア

このビットは常に 0 として読み出されます。

0 : 影響なし。  
1 : SDF STANDBY フラグのクリア (書き込み)

ビット 2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **PDDS** : パワーダウンドீプスリープ

このビットは、ソフトウェアによってセット/クリアされます。LPDS ビットとの組み合わせで動作します。

0 : CPU がディープスリープに移行すると、STOP モードに移行します。レギュレータの状態は LPDS ビットによって変化します。  
1 : CPU がディープスリープに移行すると、STANDBY モードに移行します。

ビット 0 **LPDS** : 低電力ディープスリープ

このビットは、ソフトウェアによってセット/クリアされます。このビットは PDDS ビットとの組み合わせで動作します

0 : メイン電圧レギュレータは STOP モードの間もオン状態を保ちます。  
1 : 低電力電圧レギュレータは STOP モードの間もオン状態を保ちます。

## 4.4.2 PWR 電源制御/ステータスレジスタ (PWR\_CSR1)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000 (STANDBY モードからのウェイクアップによってリセットされない)

このレジスタを読み出すには、通常の APB 読み出しに対して、追加の APB サイクルが必要です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDRDY[1:0]		ODSWRDY	ODRDY
												rc_w1	rc_w1	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VOSRDY	Res.	Res.	Res.	Res.	BRE	Res.	Res.	Res.	Res.	Res.	BRR	PVDO	SBF	WUIF
	r					rw						r	r	r	r

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 **UDRDY[1:0]** : アンダードライブレディフラグ

これらのビットは、STOP モードにおいてアンダードライブモードが有効なときにハードウェアによってセットされ、1 にプログラムすることでクリアされます。

00 : アンダードライブは無効

01 : 予約済み

10 : 予約済み

11 : STOP モード時、アンダードライブモードが作動します。

ビット 17 **ODSWRDY** : オーバードライブモード切り替えレディ

0 : オーバードライブモードは非アクティブ

1 : オーバードライブモードは、1.2 V ドメインにおけるデジタル領域でアクティブ

ビット 16 **ODRDY** : オーバードライブモードレディ

0 : オーバードライブモードはレディ状態ではない

1 : オーバードライブモードはレディ状態

ビット 14 **VOSRDY** : レギュレータ電圧スケール出力選択レディビット

0 : レディでない状態

1 : レディ状態

ビット 13:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **BRE** : バックアップレギュレータ有効化

このビットがセットされると、バックアップレギュレータ (STANDBY モードおよび V<sub>BAT</sub> モードでバックアップ SRAM の内容を保持するために使用) が有効になります。BRE がリセットされるとバックアップレギュレータはオフになります。STANDBY モードおよび V<sub>BAT</sub> モードではバックアップ SRAM はそのまま使用できますが、その内容は失われることとなります。いったんセットすると、アプリケーションは、バックアップレギュレータレディフラグ (BRR) が設定され、RAM に書き込まれたデータが STANDBY モードおよび V<sub>BAT</sub> モードにおいて保持されることが示されるのを待つ必要があります。

0 : バックアップレギュレータは無効

1 : バックアップレギュレータは有効

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

ビット 8:4 予約済みであり、リセット値に保持する必要があります。

- ビット 3 BRR** : バックアップレギュレータレディ  
 ハードウェアによってセットされ、バックアップレギュレータが動作可能状態であることを示します。  
 0 : バックアップレギュレータは動作可能状態ではありません。  
 1 : バックアップレギュレータは動作可能状態です。  
**注** : システムリセットや電源リセットによって、デバイスがSTANDBY モードからウェイクアップするとき、このビットはリセットされません。
- ビット 2 PVDO** : PVD 出力  
 このビットは、ハードウェアによってセット/クリアされます。PVDE ビットによって PVD が有効化されている場合のみ有効です。  
 0 : V<sub>DD</sub> は PLS[2:0] ビットで選択された PVD 閾値より高いです。  
 1 : V<sub>DD</sub> は PLS[2:0] ビットで選択された PVD 閾値より低いです  
**注** : PVD は STANDBY モードによって停止します。したがって、このビットは STANDBY またはリセット後、PVDE ビットがセットされるまで 0 になります。
- ビット 1 SBF** : STANDBY フラグ  
 このビットはハードウェアによってセットされ、POR/PDR (パワーオンリセット/パワーダウンリセット)、または PWR 電源制御レジスタ (PWR\_CR1) の CSBF ビットをセットすることによってのみクリアされます。  
 0 : デバイスは STANDBY モードではありません。  
 1 : デバイスは STANDBY モードです。
- ビット 0 WUIF** : ウェイクアップ割り込みフラグ  
 このビットは、STANDBY モードで内部ウェイクアップラインにウェイクアップが検出されたときにセットされます。また、すべての内部ウェイクアップソースがクリアされたときにクリアされます。  
 0 : ウェイクアップ割り込みイベントは発生していません。  
 1 : RTC アラーム (アラーム A またはアラーム B)、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップのいずれかによりウェイクアップイベントが検出されました。

## 4.4.3 PWR 電源制御/ステータスレジスタ 2 (PWR\_CR2)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000 (STANDBY モードからのウェイクアップによってリセットされない)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	WUPP6	WUPP5	WUPP4	WUPP3	WUPP2	WUPP1	Res	Res	CWUPF6	CWUPF5	CWUPF4	CWUPF3	CWUPF2	CWUPF1
		r/w	r/w	r/w	r/w	r/w	r/w			r	r	r	r	r	r

ビット 31:14 予約済み、常に 0 として読み出されます。

**ビット 13 WUPP6** : PI11 のウェイクアップピン極性ビット

これらのビットは、イベントの検出に使用される外部ウェイクアップピン PI11 の極性を定義します。  
 0 : 立ち上がりエッジで検出します。  
 1 : 立ち下がりエッジで検出します。

**ビット 12 WUPP5** : PI8 のウェイクアップピン極性ビット

これらのビットは、外部ウェイクアップピン PI8 のイベントの検出に使用される極性を定義します。  
 0 : 立ち上がりエッジで検出します。  
 1 : 立ち下がりエッジで検出します。

- ビット 11 **WUPP4** : PC13 のウェイクアップピン極性ビット  
これらのビットは、外部ウェイクアップピン PC13 のイベントの検出に使用される極性を定義します。  
0 : 立ち上がりエッジで検出します。  
1 : 立ち下がりエッジで検出します。
- ビット 10 **WUPP3** : PC1 のウェイクアップピン極性ビット  
これらのビットは、外部ウェイクアップピン PC1 のイベントの検出に使用される極性を定義します。  
0 : 立ち上がりエッジで検出します。  
1 : 立ち下がりエッジで検出します。
- ビット 9 **WUPP2** : PA2 のウェイクアップピン極性ビット  
これらのビットは、外部ウェイクアップピン PA2 のイベントの検出に使用される極性を定義します。  
0 : 立ち上がりエッジで検出します。  
1 : 立ち下がりエッジで検出します。
- ビット 8 **WUPP1** : PA0 のウェイクアップピン極性ビット  
これらのビットは、外部ウェイクアップピン PA0 のイベントの検出に使用される極性を定義します。  
0 : 立ち上がりエッジで検出します。  
1 : 立ち下がりエッジで検出します。
- ビット 7:6 予約済み。常に 0 として読み出されます。
- ビット 5 **CWUPF6** : PI11 のウェイクアップピンフラグをクリアします。  
このビットは常に 0 として読み出されます。  
0 : 影響なし。  
1 : WUPF ウェイクアップピンフラグをシステムクロックの 2 サイクル後にクリアします。
- ビット 4 **CWUPF5** : PI8 のウェイクアップピンフラグをクリアします。  
このビットは常に 0 として読み出されます。  
0 : 影響なし。  
1 : WUPF ウェイクアップピンフラグをシステムクロックの 2 サイクル後にクリアします。
- ビット 3 **CWUPF4** : PC13 のウェイクアップピンフラグをクリアします。  
このビットは常に 0 として読み出されます。  
0 : 影響なし。  
1 : WUPF ウェイクアップピンフラグをシステムクロックの 2 サイクル後にクリアします。
- ビット 2 **CWUPF3** : PC1 のウェイクアップピンフラグをクリアします。  
このビットは常に 0 として読み出されます。  
0 : 影響なし。  
1 : WUPF ウェイクアップピンフラグをシステムクロックの 2 サイクル後にクリアします。
- ビット 1 **CWUPF2** : PA2 のウェイクアップピンフラグをクリアします。  
このビットは常に 0 として読み出されます。  
0 : 影響なし。  
1 : WUPF ウェイクアップピンフラグをシステムクロックの 2 サイクル後にクリアします。
- ビット 0 **CWUPF1** : PA0 のウェイクアップピンフラグをクリアします。  
このビットは常に 0 として読み出されます。  
0 : 影響なし。  
1 : WUPF ウェイクアップピンフラグをシステムクロックの 2 サイクル後にクリアします。

## 4.4.4 PWR 電源制御レジスタ 2 (PWR\_CSR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000 (STANDBY モードからのウェイクアップによりリセット。RESET ピンでリセットされるウェイクアップフラグを除く)

このレジスタを読み出すには、通常の APB 読み出しに対して、追加の APB サイクルが必要です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	EWUP6	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1	Res.	Res.	WUPF6	WUPF5	WUPF4	WUPF3	WUPF2	WUPF1
		rW	rW	rW	rW	rW	rW			r	r	r	r	r	r

ビット 31:14 予約済み、常に 0 として読み出されます。

ビット 13 **EWUP6** : PI11 のウェイクアップピンを有効にします。

このビットは、ソフトウェアによってセット/クリアされます。

0 : この WKUP ピン PI11 でイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : この WKUP ピン PI11 の立ち上がり/立ち下がりエッジにより、システムを STANDBY モードからウェイクアップさせます。

ビット 12 **EWUP5** : PI8 のウェイクアップピンを有効にします。

このビットは、ソフトウェアによってセット/クリアされます。

0 : この WKUP ピン PI8 でイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : この WKUP ピン PI8 で立ち上がり/立ち下がりエッジが発生し、システムを STANDBY モードからウェイクアップさせます。

ビット 11 **EWUP4** : PC13 のウェイクアップピンを有効にします。

このビットは、ソフトウェアによってセット/クリアされます。

0 : この WKUP ピン PC13 でイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : この WKUP ピン PC13 で立ち上がり/立ち下がりエッジが発生し、システムを STANDBY モードからウェイクアップさせます。

ビット 10 **EWUP3** : PC1 のウェイクアップピンを有効にします。

このビットは、ソフトウェアによってセット/クリアされます。

0 : この WKUP ピン PC1 でイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : この WKUP ピン PC1 で立ち上がり/立ち下がりエッジが発生し、システムを STANDBY モードからウェイクアップさせます。

ビット 9 **EWUP2** : PA2 のウェイクアップピンを有効にします。

このビットは、ソフトウェアによってセット/クリアされます。

0 : この WKUP ピン PA2 でイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : この WKUP ピン PA2 で立ち上がり/立ち下がりエッジが発生し、システムを STANDBY モードからウェイクアップさせます。

ビット 8 **EWUP1** : PA0 のウェイクアップピンを有効にします。

このビットは、ソフトウェアによってセット/クリアされます。

0 : この WKUP ピン PA0 でイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : この WKUP ピン PA0 で立ち上がり/立ち下がりエッジが発生し、システムを STANDBY モードからウェイクアップさせます。

ビット 7:6 予約済み。常に 0 として読み出されます。

ビット 5 **WUPF6** : PI11 のウェイクアップピンフラグ

このビットは、ハードウェアによってセットされ、リセットピンまたは PWR 電源制御レジスタ 2 (PWR\_CR2) の CWUPF6 ビットをセットすることのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP PI11 でウェイクアップイベントが検出されました。

**注 :** *WKUP ピン PI11 のレベルがすでにハイであっても、(EWUP6 ビットをセットすることによって) WKUP ピンが有効化されている場合、追加のウェイクアップイベントが検出されます。*

ビット 4 **WUPF5** : PI8 のウェイクアップピンフラグ

このビットは、ハードウェアによってセットされ、リセットピンまたは PWR 電源制御レジスタ 2 (PWR\_CR2) の CWUPF5 ビットをセットすることのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP PI8 でウェイクアップイベントが検出されました。

**注 :** *WKUP ピン PI8 のレベルがすでにハイであっても、(EWUP5 ビットをセットすることによって) WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。*

ビット 3 **WUPF4** : PC13 のウェイクアップピンフラグ

このビットは、ハードウェアによってセットされ、リセットピンまたは PWR 電源制御レジスタ 2 (PWR\_CR2) の CWUPF4 ビットをセットすることのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP PC13 でウェイクアップイベントが検出されました。

**注 :** *WKUP ピン PC13 のレベルがすでにハイであっても、(EWUP4 ビットをセットすることによって) WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。*

ビット 2 **WUPF3** : PC1 のウェイクアップピンフラグ

このビットは、ハードウェアによってセットされ、リセットピンまたは PWR 電源制御レジスタ 2 (PWR\_CR2) の CWUPF3 ビットをセットすることのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP PC1 でウェイクアップイベントが検出されました。

**注 :** *WKUP ピン PC1 のレベルがすでにハイであっても、(EWUP3 ビットをセットすることによって) WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。*

ビット 1 **WUPF2** : PA2 のウェイクアップピンフラグ

このビットは、ハードウェアによってセットされ、リセットピンまたは PWR 電源制御レジスタ 2 (PWR\_CR2) の CWUPF2 ビットをセットすることのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP PA2 でウェイクアップイベントが検出されました。

**注 :** *WKUP ピン PA2 のレベルがすでにハイであっても、(EWUP2 ビットをセットすることによって) WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。*

ビット 0 **WUPF1** : PA0 のウェイクアップピンフラグ

このビットは、ハードウェアによってセットされ、リセットピンまたは PWR 電源制御レジスタ 2 (PWR\_CR2) の CWUPF1 ビットをセットすることのみクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP PA0 でウェイクアップイベントが検出されました。

**注 :** *WKUP ピン PA0 のレベルがすでにハイであっても、(EWUP1 ビットをセットすることによって) WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。*

## 4.5 PWR レジスタマップ

次の表に PWR レジスタの一覧を示します。

表 19. PWR - レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	PWR_CR1	Res.	UDEN[1:0]	ODSMEN	ODEN	VOS[1:0]	ADCDC1	Res.	Res.	MRUDS	LPUDS	FPDS	DBP	Res.	PLS[2:0]	Res.	PVDE	CSBF	Res.	PDDS	LPDS												
	リセット値														1	1	1	1	1	0		1	1	0	0	0	0	0	0	0	0	0	0
0x004	PWR_CSR1	Res.	UDRDY[1:0]	ODSWRDY	ODRDY	Res.	VOSRDY	Res.	Res.	Res.	Res.	BRE	Res.	Res.	Res.	Res.	Res.	BRR	PVDO	SBF	WUJF												
	リセット値														0	0	0	0	0					0	0	0	0	0	0	0	0	0	0
0x008	PWR_CR2	Res.	Res.	Res.	Res.	Res.	WUJPF6	WUJPF5	WUJPF4	WUJPF3	WUJPF2	WUJPF1	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	PWR_CSR2	Res.	Res.	Res.	Res.	Res.	EWJPF6	EWJPF5	EWJPF4	EWJPF3	EWJPF2	EWJPF1	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 5 リセットおよびクロック制御 (RCC)

### 5.1 リセット

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

#### 5.1.1 システムリセット

システムリセットは、クロックコントローラ CSR レジスタのリセットフラグとバックアップドメインのレジスタを除き、すべてのレジスタをリセット値にセットします (図 11 を参照)。

システムリセットは、次のイベントのいずれかの発生時に生成されます。

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

#### ソフトウェアリセット

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

デバイス上でソフトウェアリセットを実行するには、Cortex®-M7 のアプリケーション割り込みおよびリセット制御レジスタの SYSRESETREQ ビットをセットする必要があります。詳細については、Cortex®-M7 Technical Reference Manual を参照してください。

#### 低電力管理リセット

低電力管理リセットを生成するには、2つの方法があります。

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

### 5.1.2 電源リセット

電源リセットは、次のいずれかのイベントが発生したときに生成されます。

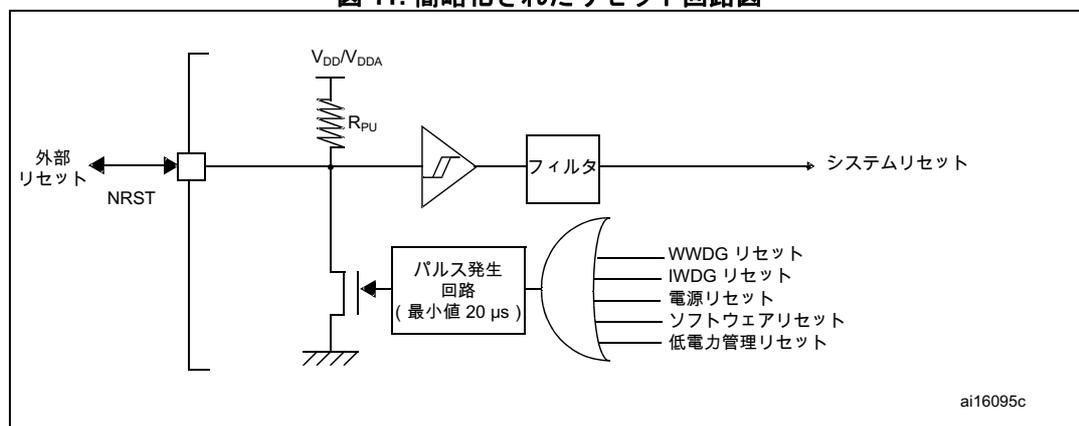
1. パワーオン/パワーダウンリセット (POR/PDR リセット) またはブラウンアウト (BOR) リセット
2. STANDBY モードを終了するとき

電源リセットは、バックアップドメインを除くすべてのレジスタをリセット値にセットします (図 11 を参照)。

これらのソースは NRST ピンに作用し、遅延フェーズの間、ピンをローに保ちます。リセットサービスルーチンベクタは、メモリマップのアドレス 0x0000\_0004 に固定されています。

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

図 11. 簡略化されたリセット回路図



バックアップドメインには、バックアップドメインだけに影響する 2 種類の特定のリセットがあります (図 11 を参照)。

### 5.1.3 バックアップドメインリセット

バックアップドメインリセットは、すべての RTC レジスタと RCC\_BDCR レジスタをリセット値にセットします。BKPSRAM は、このリセットの影響を受けません。BKPSRAM をリセットする唯一の方法は、フラッシュインタフェースを使用して保護レベルの 1 から 0 への変更を要求することです。

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

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

## 5.2 クロック

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

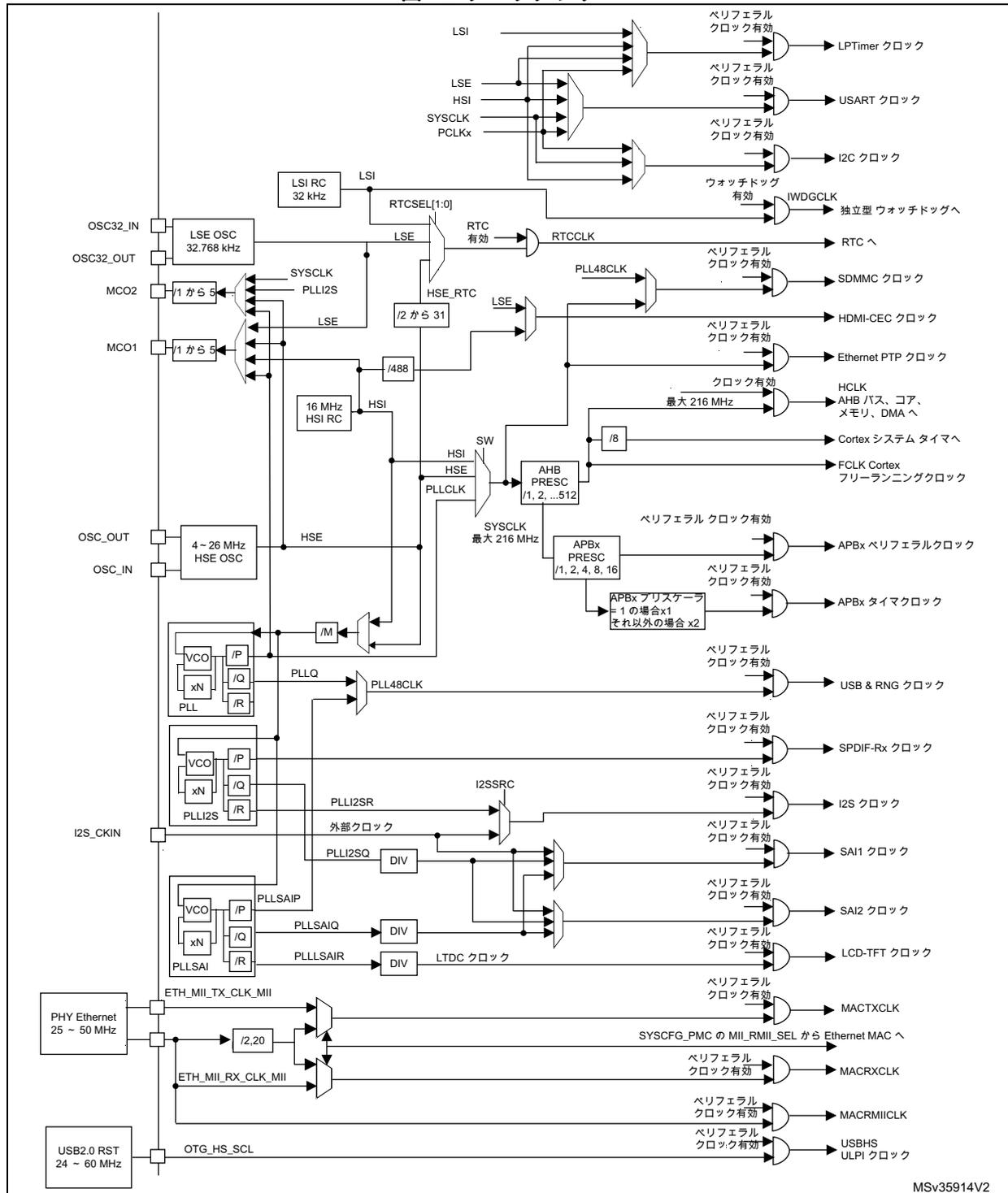
- HSI オシレータクロック
- HSE オシレータクロック
- メイン PLL (PLL) クロック

デバイスには、次の 2 つの 2 次クロックソースがあります。

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

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

図 12. クロックツリー



1. 内部および外部クロックソース特性の詳細については、デバイスのデータシートの「電気特性」のセクションを参照してください。
2. RCC\_DKCFGR1レジスタのTIMPREビットがリセットされた場合、APBxプリスケラが1であればTIMxCLK = PCLKx、それ以外ではTIMxCLK = 2x PCLKxとなります。
3. RCC\_DKCFGR1レジスタのTIMPREビットがセットされた場合、APBxプリスケラが1、2、4であればTIMxCLK = HCLK、それ以外ではTIMxCLK = 4x PCLKxとなります。

クロックコントローラは、外部クリスタルまたはオシレータの選択においてアプリケーションに高度な柔軟性を提供しており、コアおよびペリフェラルを最大周波数で実行します。

すべてのペリフェラルクロックは、それぞれのバスクロック (HCLK、PLCK1、PCLK2) から生成されます。ただし、次のものを除きます。

- USB OTG FS、SDMMC、および RNG に使用される 48MHz クロックは、次のうちのいずれかのソースから生成されます。
  - メイン PLL VCO (PLLQ クロック)
  - PLLSAI VCO (PLLSAI クロック)
- U(S)ART クロックは、次のうちのいずれかのソースから生成されます。
  - システムクロック (SYSCLK)
  - HSI クロック
  - LSE クロック
  - APB1 または APB2 クロック (U(S)ART に配置されている APB に応じて、PCLK1 または PCLK2)
- I2C クロックは、次のうちのいずれかのソースから生成されます。
  - システムクロック (SYSCLK)
  - HSI クロック
  - APB1 クロック (PCLK1)
- I2S クロック  
高品質のオーディオ性能を実現するため、I2S クロックを特定の PLL (PLLI2S) または I2S\_CKIN ピン上に配置された外部クロックから得ることができます。I2S クロック周波数と精度に関する詳細については、[セクション 32.7.4 : クロックジェネレータ](#)を参照してください。
- SAI1 および SAI2 クロックは、次のうちのいずれかのソースから生成されます。
  - PLLSAI VCO (PLLSAIQ)
  - PLLI2S VCO (PLLI2SQ)
  - I2S\_CKIN ピンに配置された外部クロック
- LTDC クロック  
LTDC クロックは、特定の PLL (PLLSAI) から生成されます。
- 低電カタイマ (LPTIM1) クロックは、次のうちのいずれかのソースから生成されます。
  - LSI クロック
  - LSE クロック
  - HSI クロック
  - APB1 クロック (PCLK1)
  - LPTIM1\_IN1 ピンに配置された外部クロック
- USB OTG HS (60 MHz) クロック。外部 PHY から供給されます。
- Ethernet MAC クロック (TX、RX、および RMII)。外部 PHY から供給されます。Ethernet構成の詳細については、Ethernetペリフェラル説明書の [セクション 38.4.4 : MII/RMII の選択](#)を参照してください。Ethernet が使用される際には、AHB クロック周波数は少なくとも 25 MHz でなければなりません。
- SPDIFRX クロック : PLLI2SP VCO から生成されます。
- HDMI-CEC クロックは、次のうちのいずれかのソースから生成されます。
  - LSE クロック
  - 488 分周された HSI クロック

- RTC クロックは、次のうちのいずれかのソースから生成されます。
  - LSE クロック
  - LSI クロック
  - 32 分周された HSE クロック
- IWDG クロックは、常に LSI クロックから生成されます。
- タイマクロック周波数は、ハードウェアによって自動的に設定されます。RCC\_CFGR レジスタの TIMPRE ビットの値により、2 つのケースがあります。
  - RCC\_DKCFGR1 レジスタの TIMPRE ビットがリセットされた場合：  
APB プリスケーラが分周比 1 に設定されると、タイマクロック周波数 (TIMxCLK) は PCLKx に設定されます。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 2 倍に設定されます。TIMxCLK = 2xPCLKx
  - RCC\_DKCFGR1 レジスタの TIMPRE ビットがセットされた場合：  
APB プリスケーラが分周比 1、2、または 4 に設定されると、タイマクロック周波数 (TIMxCLK) は HCLK に設定されます。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 4 倍に設定されます。TIMxCLK = 4xPCLKx

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

FCLK は Cortex<sup>®</sup>-M7 のフリーランニングクロックとして動作します。詳細については、Cortex<sup>®</sup>-M7 Technical Reference Manual を参照してください。

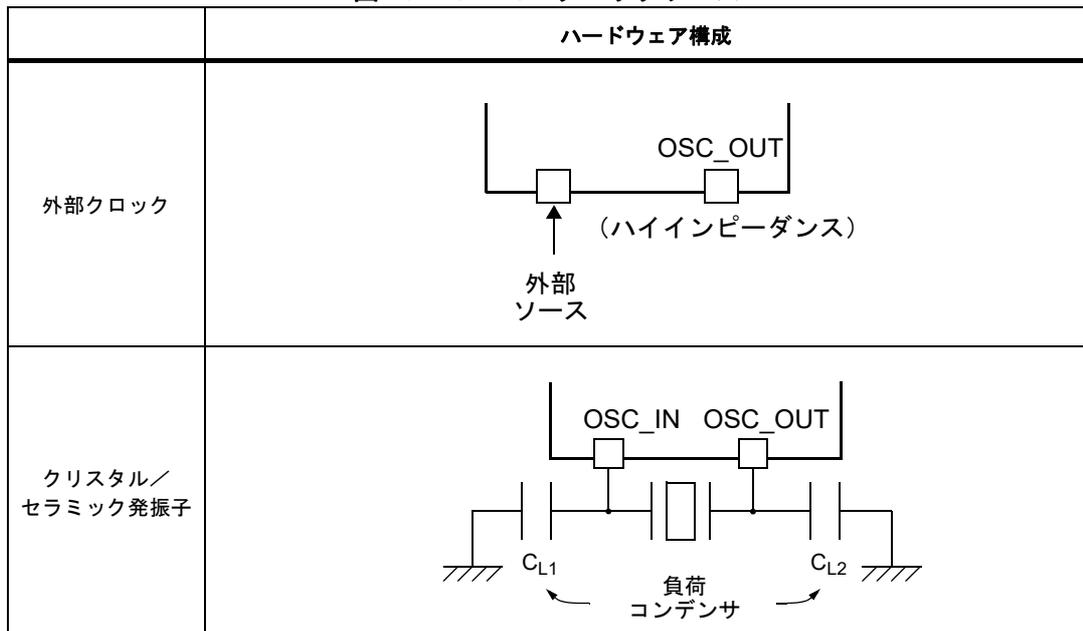
## 5.2.1 HSE クロック

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

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

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

図 13. HSE/LSE クロックソース



### 外部ソース (HSE バイパス)

このモードでは、外部クロックソースが必要です。このモードを選択するには、[RCC クロック制御レジスタ \(RCC\\_CR\)](#) HSEBYP および HSEON の <317 ビットをセットします。約 50 % のデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC\_IN ピンを駆動する必要があり、その間、OSC\_OUT ピンはハイインピーダンスに保つ必要があります ([図 13](#) を参照)。

### 外部クリスタル／セラミック発振子 (HSE クリスタル)

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

関連するハードウェア構成を [図 13](#) に示します。詳細については、データシートの電気的特性のセクションを参照してください。

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

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

## 5.2.2 HSI クロック

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

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

## 較正

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

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

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

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

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

HSI 信号は、HSE クリスタルオシレータに障害がある場合のバックアップソース（補助クロック）としても使用できます。[セクション 5.2.7: クロックセキュリティシステム \(CSS\) \(139 ページ\)](#) を参照してください。

## 5.2.3 PLL

デバイスは、次の 3 つの PLL を搭載しています。

- HSE または HSI オシレータで駆動されるメイン PLL (PLL) で、2 種類の異なる出力クロックを備えています。
  - 最初の出力は、高速システムクロック（最大 216 MHz）の生成に使用します。
  - 2 つ目の出力は、USB OTG FS、SDMMC、および RNG に使用される 48 MHz クロックの生成に使用します。
- PLLI2S は、I2S、SAI、および SPDIFRX インタフェースで高品質オーディオ性能を実現するための正確なクロックの生成に使用します。
- PLLSAI は、SAI インタフェース用のクロック、LCD-TFT クロック、および USB OTG FS、SDMMC、および RNG で選択できる 48MHz クロック (PLLSAI48CLK) の生成に使用します。

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

PLLI2S および PLLSAI は、PLL と同じ入力クロックを使用します（PLLM[5:0] と PLLSRC ビットはどちらの PLL にも共通）。PLLI2S および PLLSAI には有効/無効および分周比（N および R）を設定する専用のビットがあります。いったん PLLI2S および PLLSAI が有効になると、設定パラメータは変更できません。

STOP および STANDBY モードに入るときや、HSE または PLL（HSE で駆動される）がシステムクロックとして使用されているときに HSE で障害が発生した場合、3 つの PLL はハードウェアによって無効になります。[RCC PLL 設定レジスタ \(RCC\\_PLLCFGR\)](#)、[RCC クロック設定レジスタ \(RCC\\_CFR\)](#)、および [RCC 専用クロック設定レジスタ \(RCC\\_DKCFGR1\)](#) を使用して、PLL、PLLI2S、PLLSAI を設定できます。

## 5.2.4 LSE クロック

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

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

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

### 外部ソース (LSE バイパス)

このモードでは、外部クロックソースが必要です。最大 1 MHz までの周波数を使用する必要があります。このモードを選択するには、[RCC バックアップドメイン制御レジスタ \(RCC\\_BDCR\)](#) の LSEBYP および LSEON ビットをセットします。約50%のデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC32\_IN ピンを駆動する必要があり、その間、OSC32\_OUT ピンはハイインピーダンスに保つ必要があります (図 13 を参照)。

## 5.2.5 LSI クロック

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

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

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

## 5.2.6 システムクロック (SYSCLK) の選択

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

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

## 5.2.7 クロックセキュリティシステム (CSS)

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

HSE クロックに異常が検出された場合、このオシレータは自動的に無効になり、クロック異常イベントが TIM1 および TIM8 高機能制御タイマのブレーク入力に送られます。また、ソフトウェアに異常を通知する割り込み (クロックセキュリティシステム割り込み (CSSI)) が生成されるので、MCU は対応処理を行うことができません。CSSI は、Cortex<sup>®</sup>-M7 NMI (ノンマスカブル割り込み) 例外ベクタにリンクされています。

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

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

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

## 5.2.8 RTC/AWU クロック

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

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

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

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

- LSE が RTC クロックとして使用されている場合：
  - $V_{DD}$  供給がオフになった場合でも、 $V_{BAT}$  の供給が保たれている限り、RTC は動作を続けます。
- LSI が自動ウェイクアップユニット (AWU) クロックとして選択されている場合：
  - $V_{DD}$  供給がオフになった場合、AWU の状態は保証されません。LSI 較正の詳細については、[セクション 5.2.5 : LSI クロック \(138 ページ\)](#) を参照してください。
- HSE クロックが RTC クロックとして使用されている場合：
  - $V_{DD}$  供給がオフになった場合や、内蔵の電圧レギュレータがオフになり (1.2 V ドメインへの電力供給が停止) STOP モードに移行した場合、RTC の状態は保証されません。

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

## 5.2.9 ウォッチドッグクロック

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

## 5.2.10 クロック信号出力

2 つのマイクロコントローラクロック出力 (MCO) ピンが利用できます。

- MCO1
  - 設定可能なプリスケアラ (1 から 5) を使用して、MCO1 ピン (PA8) から 4 種類のクロックソースを出力できます。
    - HSI クロック
    - LSE クロック
    - HSE クロック
    - PLL クロック
  - [RCC クロック設定レジスタ \(RCC\\_CFGR\)](#) の MCO1PRE[2:0] および MCO1[1:0] ビットを使用して、希望するクロックソースが選択されます。
- MCO2
  - 設定可能なプリスケアラ (1 から 5) を使用して、MCO2 ピン (PC9) から 4 種類のクロックソースを出力できます。
    - HSE クロック
    - PLL クロック
    - システムクロック (SYSCLK)
    - PLLI2S クロック
  - [RCC クロック設定レジスタ \(RCC\\_CFGR\)](#) の MCO2PRE[2:0] および MCO2 ビットを使用して、希望するクロックソースが選択されます。

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

## 5.2.11 TIM5/TIM11 を使用した内部/外部クロックの測定

図 14 および 図 15 に示すように、TIM5 チャンネル 4 および TIM11 チャンネル 1 の入力キャプチャを使用することにより、ボード上のすべてのクロックソース発生回路の周波数を間接的に測定することができます。

### TIM5 チャンネル 4 を使用した内部/外部クロックの測定

TIM5 には、入力キャプチャが I/O によってトリガされるのか内部クロックによってトリガされるのかを選択できる入力マルチプレクサがあります。この選択は、TIM5\_OR レジスタの TI4\_RMP [1:0] ビットで行います。

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

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

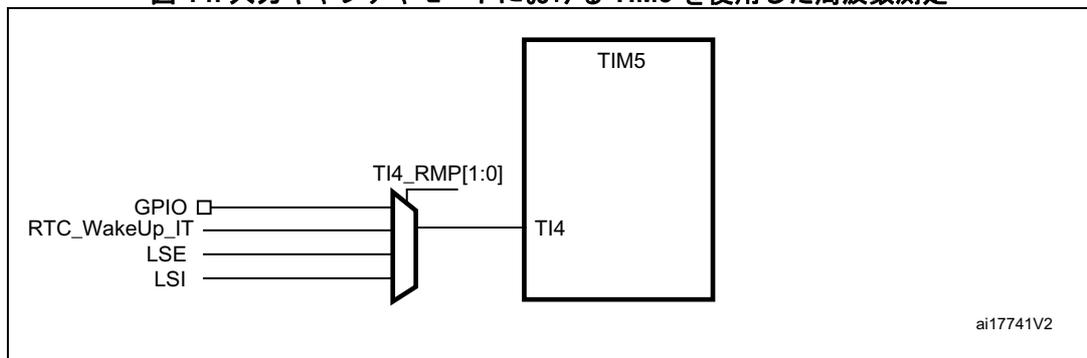
この基本概念は、相対測定（HSI/LSE 比など）ができることにあり、そのため、精度は 2 つのクロックソース間の比に密接につながっています。この比が大きければ大きいほど、測定精度は向上します。

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

LSI 周波数を測定するには、次の手順に従ってください。

1. TIM5 タイマを有効にして、チャンネル 4 を入力キャプチャモードに設定します。
2. TIM5\_OR レジスタの TI4\_RMP ビットを 0x01 にセットして、較正目的で、LSI クロックを TIM5 チャンネル 4 入力キャプチャに内部接続します。
3. TIM5 キャプチャ/比較 4 イベントまたは割り込みを使用して、LSI クロックの周波数を計測します。
4. 計測した LSI 周波数を使用して、必要な時間基準に従って RTC のプリスケアラを更新したり、IWDG タイムアウトを計算したりします。

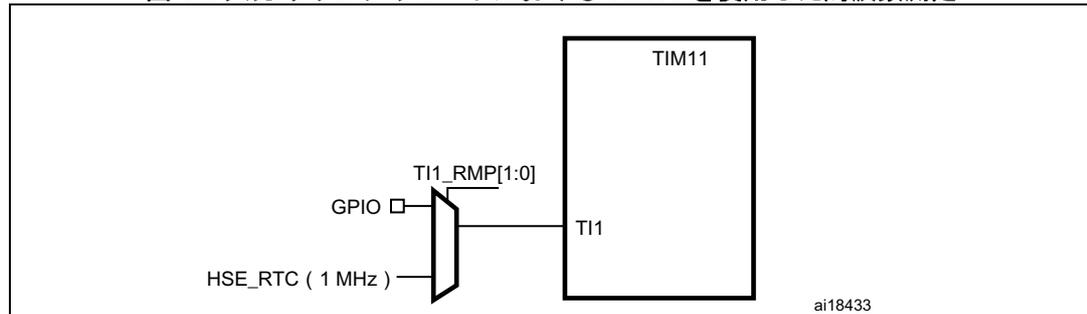
図 14. 入力キャプチャモードにおける TIM5 を使用した周波数測定



## TIM11 チャンネル 1 を使用した内部/外部クロックの測定

TIM11 には、入力キャプチャが I/O によってトリガされるのか内部クロックによってトリガされるのかを選択できる入力マルチプレクサがあります。この選択は、TIM11\_OR レジスタの TI1\_RMP [1:0] ビットで行います。HSE\_RTC クロック (HSE をプログラマブルプリスケーラで分周) は、チャンネル 1 の入力キャプチャに接続されており、およその外部クリスタル周波数を知ることができます。このとき、HSI がシステムクロックソースである必要があります。これは、たとえば高調波周波数または低調波周波数 (-50/+100 % の偏差) が判断できることを求めている IEC 60730/IEC 61335 規格に準拠していることを保証するのに有用です。

図 15. 入力キャプチャモードにおける TIM11 を使用した周波数測定



### 5.2.12 ペリフェラルクロック有効レジスタ (RCC\_AHBxENR、RCC\_APBxENRy)

各ペリフェラルクロックは、RCC\_AHBxENR または RCC\_APBxENRy レジスタの xxxxEN ビットによって有効にできます。

ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの読み出し/書き込みアクセスはサポートされません。ペリフェラルイネーブルビットには、グリッチのないペリフェラルのクロックを生成する同期メカニズムが含まれています。

イネーブルビットがセットされてからクロックがアクティブになるまでに、2 ペリフェラルクロックサイクルの遅延があります。

**注意：** ペリフェラルのクロックを有効にした直後、ソフトウェアは 2 ペリフェラルクロックサイクルだけ待ってから、ペリフェラルレジスタにアクセスする必要があります。

## 5.3 RCC レジスタ

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

### 5.3.1 RCC クロック制御レジスタ (RCC\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 XX83 (X は未定義)。

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PLLSAI RDY	PLLSAI ON	PLLI2SRDY	PLLI2SON	PLLRDY	PLLON	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
		r	rw	r	rw	r	rw					rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 29 **PLLSAIRDY** : PLLSAI クロックレディフラグ

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

- 0 : PLLSAI アンロック
- 1 : PLLSAI ロック

ビット 28 **PLLSAION** : PLLSAI 有効化

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

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

- 0 : PLLSAI オフ
- 1 : PLLSAI オン

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

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

- 0 : PLLI2S アンロック
- 1 : PLLI2S ロック

ビット 26 **PLLI2SON** : PLLI2S 有効化

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

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

- 0 : PLLI2S オフ
- 1 : PLLI2S オン

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

ハードウェアによってセットされ、PLL がロック状態であることを示します。

- 0 : PLL アンロック
- 1 : PLL ロック

ビット 24 **PLLON** : メイン PLL (PLL) 有効化

PLL を有効にするために、ソフトウェアによってセット/クリアされます。

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

- 0 : PLL オフ
- 1 : PLL ON

ビット 23:20 予約済みであり、リセット値に保持する必要があります。



**ビット 19 CSSON** : クロックセキュリティシステム有効化

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

0 : クロックセキュリティシステムオフ (クロック検出回路オフ)

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

**ビット 18 HSEBYP** : HSE クロックバイパス

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

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

0 : HSE オシレータはバイパスされません。

1 : HSE オシレータはバイパスされ、外部クロックが使用されます。

**ビット 17 HSERDY** : HSE クロックレディフラグ

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

0 : HSE オシレータはレディ状態ではありません。

1 : HSE オシレータはレディ状態です。

**ビット 16 HSEON** : HSE クロック有効化

ソフトウェアでセット/クリアされます。

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

0 : HSE オシレータオフ

1 : HSE オシレータオン

**ビット 15:8 HSICAL[7:0]** : 内部高速クロック較正

これらのビットは、スタートアップ時に自動的に初期化されます。

**ビット 7:3 HSITRIM[4:0]** : 内部高速クロックトリミング

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

ビット 2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 HSIRDY** : 内部高速クロックレディフラグ

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

0 : HSI オシレータはレディ状態ではありません。

1 : HSI オシレータはレディ状態です。

**ビット 0 HSION** : 内部高速クロック有効化

ソフトウェアでセット/クリアされます。

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

0 : HSI オシレータオフ

1 : HSI オシレータオン

## 5.3.2 RCC PLL 設定レジスタ (RCC\_PLLCFGR)

アドレスオフセット : 0x04

リセット値 : 0x2400 3010

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

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

- $f$  (VCO クロック) =  $f$  (PLL クロック入力) \* (PLLN / PLLM)
- $f$  (PLL 汎用クロック出力) =  $f$  (VCO クロック) / PLLP
- $f$  (USB OTG FS、SDMMC、RNG クロック出力) =  $f$  (VCO クロック) / PLLQ

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

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **PLLQ[3:0]** : USB OTG FS、SDMMC、乱数発生回路のクロックのためのメイン PLL (PLL) 分周比  
 USB OTG FS クロック、乱数発生回路クロック、SDMMC クロックの周波数を制御するためにソフトウェアによってセット/クリアします。これらのビットに書き込めるのは、PLL が無効なときだけです。

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

USB OTG FS クロック周波数 = VCO 周波数 / PLLQ、ここで  $2 \leq PLLQ \leq 15$

0000 : PLLQ = 0、誤った設定

0001 : PLLQ = 1、誤った設定

0010 : PLLQ = 2

0011 : PLLQ = 3

0100 : PLLQ = 4

...

1111 : PLLQ = 15

ビット 23 予約済みであり、リセット値に保持する必要があります。

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

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

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

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

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

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

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

**注意:** ソフトウェアは、このドメインで 180 MHz を超えないように、これらのビットを正しくセットする必要があります。

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

00 : PLLP = 2

01 : PLLP = 4

10 : PLLP = 6

11 : PLLP = 8

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

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

**注意:** これらのビットは、VCO 出力周波数が 192~432 MHz の間になるように、ソフトウェアで正しく設定する必要があります。以下の例は、PLL 入力  $F_{PLL\_IN} = 1 \text{ MHz}$  に等しい場合に、禁止されている PLLN ビットフィールドの値です。

VCO 出力周波数 = VCO 入力周波数 \* PLLN、ここで  $192 \leq PLLN \leq 432$

000000000 : PLLN = 0、誤った設定

000000001 : PLLN = 1、誤った設定

...

011000000 : PLLN = 192

...

110110000 : PLLN = 432

110110001 : PLLN = 433、誤った設定

...

111111111 : PLLN = 511、誤った設定

ビット 5:0 **PLLM[5:0]** : メイン PLL (PLL、PLLI2S、および PLLSAI) 入カクロックのための分周比

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

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

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

000000 : PLLM = 0、誤った設定

000001 : PLLM = 1、誤った設定

000010 : PLLM = 2

000011 : PLLM = 3

000100 : PLLM = 4

...

111110 : PLLM = 62

111111 : PLLM = 63

## 5.3.3 RCC クロック設定レジスタ (RCC\_CFGR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : 0 ≤ ウェイトステート ≤ 2、ワード、ハーフワード、およびバイトアクセス

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2		MCO2 PRE[2:0]			MCO1 PRE[2:0]			I2SSCR	MCO1		RTCPRE[4:0]				
rw		rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPRE2[2:0]			PPRE1[2:0]			Res.	Res.	HPRE[3:0]				SWS1	SWS0	SW1	SW0
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	r	r	rw	rw

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

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

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

### ビット 27:29 MCO2PRE : MCO2 プリスケーラ

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

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

### ビット 24:26 MCO1PRE : MCO1 プリスケーラ

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

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

### ビット 23 I2SSRC : I2S クロック選択

ソフトウェアでセット/クリアされます。このビットによって、PLLI2S クロックと外部クロックから I2S クロックソースを選択できます。このビットは、リセット後に I2S モジュールを有効にする前に変更することを強く推奨します。

- 0 : I2S クロックソースとして PLLI2S クロックが使用されます。
- 1 : I2S クロックソースとして I2S\_CKIN ピンに割り当てられている外部クロックが使用されます。



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

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

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

ビット 20:16 **RTCPRE** : RTC クロックのための HSE 分周比

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

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

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

ビット 15:13 **PPRE2** : APB 高速プリスケアラ (APB2)

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

**注意 :** ソフトウェアは、このドメインで 90 MHz を超えないように、これらのビットを正しくセットする必要があります。クロックは、PPRE2 に書き込んでから 1~16 AHB サイクルの間の新しい分周比で分周されます。

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

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

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

**注意 :** ソフトウェアは、このドメインで 45 MHz を超えないように、これらのビットを正しくセットする必要があります。クロックは、PPRE1 に書き込んでから 1~16 AHB の間の新しい分周比で分周されます。

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

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

**ビット 7:4 HPRE** : AHB プリスケーラ

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

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

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

0xxx : システムクロックは分周されません。

1000 : システムクロックは 2 分周されます。

1001 : システムクロックは 4 分周されます。

1010 : システムクロックは 8 分周されます。

1011 : システムクロックは 16 分周されます。

1100 : システムクロックは 64 分周されます。

1101 : システムクロックは 128 分周されます。

1110 : システムクロックは 256 分周されます。

1111 : システムクロックは 512 分周されます。

**ビット 3:2 SWS** : システムクロックスイッチステータス

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

00 : HSI オシレータがシステムクロックとして使用されています。

01 : HSE オシレータがシステムクロックとして使用されています。

10 : PLL がシステムクロックとして使用されています。

11 : 適用されません。

**ビット 1:0 SW** : システムクロックスイッチ

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

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

00 : HSI オシレータがシステムクロックとして選択されます。

01 : HSE オシレータがシステムクロックとして選択されます。

10 : PLL がシステムクロックとして選択されます。

11 : 設定禁止

## 5.3.4 RCC クロック割り込みレジスタ (RCC\_CIR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	PLLSAIRDYC	PLLI2SRDYC	PLLRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLLSAIRDYIE	PLLI2SRDYIE	PLLRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	PLLSAIRDYF	PLLI2SRDYF	PLLRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

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

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

0 : 影響なし。

1 : CSSF フラグをクリアします。

ビット 22 **PLLSAIRDYC** : PLLSAI レディ割り込みクリア

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

PLLSAIRDYF がクリアされると、ハードウェアによってリセットされます。

0 : PLLSAIRDYF はクリアされません。

1 : PLLSAIRDYF がクリアされます

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

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

0 : 影響なし。

1 : PLLI2SRDYF がクリアされます。

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

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

0 : 影響なし。

1 : PLLRDYF がクリアされます。

ビット 19 **HSERDYC** : HSE レディ割り込みクリア

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

0 : 影響なし。

1 : HSERDYF がクリアされます。

ビット 18 **HSIRDYC** : HSI レディ割り込みクリア

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

0 : 影響なし。

1 : HSIRDYF がクリアされます。

ビット 17 **LSERDYC** : LSE レディ割り込みクリア

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

0 : 影響なし。

1 : LSERDYF がクリアされます。

**ビット 16 LSIRDYC** : LSI レディ割り込みクリア

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

0 : 影響なし。

1 : LSIRDYF がクリアされます。

ビット 15 予約済みであり、リセット値に保持する必要があります。

**ビット 14 PLLSAIRDYIE** : PLLSAI レディ割り込み有効化

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

0 : PLLSAI ロック割り込み無効

1 : PLLSAI ロック割り込み有効

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

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

0 : PLLI2S ロック割り込み無効

1 : PLLI2S ロック割り込み有効

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

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

0 : PLL ロック割り込みは無効です。

1 : PLL ロック割り込みは有効です。

**ビット 11 HSERDYIE** : HSE レディ割り込み有効化

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

0 : HSE レディ割り込みは無効です。

1 : HSE レディ割り込みは有効です。

**ビット 10 HSIRDYIE** : HSI レディ割り込み有効化

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

0 : HSI レディ割り込み無効

1 : HSI レディ割り込み有効

**ビット 9 LSERDYIE** : LSE レディ割り込み有効化

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

0 : LSE レディ割り込みは無効です。

1 : LSE レディ割り込みは有効です。

**ビット 8 LSIRDYIE** : LSI レディ割り込み有効化

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

0 : LSI レディ割り込みは無効です。

1 : LSI レディ割り込みは有効です。

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

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

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

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

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

**ビット 6 PLLSAIRDYF : PLLSAI レディ割り込みフラグ**

このビットは、PLLSAI がロックされ、PLLSAIRDYDIE がセットされているときに、ハードウェアによってセットされます。

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

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

1 : PLLSAI ロックによるクロックレディ割り込みが発生しました。

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

このビットは、PLLI2S がロックされ、PLLI2SRDYDIE がセットされているときに、ハードウェアによってセットされます。

PLLI2SDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

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

1 : PLLI2S ロックによるクロックレディ割り込みが発生しました。

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

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

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

0 : PLL ロックによるクロックレディ割り込みは発生していません。

1 : PLL ロックによるクロックレディ割り込みが発生しました。

**ビット 3 HSERDYF : HSE レディ割り込みフラグ**

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

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

0 : HSE オシレータによるクロックレディ割り込みは発生していません。

1 : HSE オシレータによるクロックレディ割り込みが発生しました。

**ビット 2 HSIRDYF : HSI レディ割り込みフラグ**

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

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

0 : HSI オシレータによるクロックレディ割り込みは発生していません。

1 : HSI オシレータによるクロックレディ割り込みが発生しました。

**ビット 1 LSERDYF : LSE レディ割り込みフラグ**

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

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

0 : LSE オシレータによるクロックレディ割り込みは発生していません。

1 : LSE オシレータによるクロックレディ割り込みが発生しました。

**ビット 0 LSIRDYF : LSI レディ割り込みフラグ**

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

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

0 : LSI オシレータによるクロックレディ割り込みは発生していません。

1 : LSI オシレータによるクロックレディ割り込みが発生しました。

## 5.3.5 RCC AHB1 ペリフェラルリセットレジスタ (RCC\_AHB1RSTR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	OTGHS RST	Res.	Res.	Res.	ETHMAC RST	Res.	DMA2D RST	DMA2 RST	DMA1 RST	Res.	Res.	Res.	Res.	Res.
		r/w				r/w		r/w	r/w	r/w					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCRS T	Res.	GPIOK RST	GPIOJ RST	GPIOI RST	GPIOH RST	GPIOGG RST	GPIOF RST	GPIOE RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST
			r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **OTGHSRST** : USB OTG HS モジュールリセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : USB OTG HS モジュールをリセットしません。

1 : USB OTG HS モジュールをリセットします。

ビット 28:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **ETHMACRST** : Ethernet MAC リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : Ethernet MAC をリセットしません。

1 : Ethernet MAC をリセットします。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **DMA2DRST** : DMA2D リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : DMA2D をリセットしません。

1 : DMA2D をリセットします。

ビット 22 **DMA2RST** : DMA2 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : DMA2 をリセットしません。

1 : DMA2 をリセットします。

ビット 21 **DMA1RST** : DMA1 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : DMA1 をリセットしません。

1 : DMA1 をリセットします。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCRST** : CRC リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : CRC をリセットしません。

1 : CRC をリセットします。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **GPIOKRST** : IO ポート K リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : IO ポート K をリセットしません。

1 : IO ポート K をリセットします。

- ビット 9 GPIOJRST** : IO ポート J リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート J をリセットしません。  
1 : IO ポート J をリセットします。
- ビット 8 GPIOIRST** : IO ポート I リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート I をリセットしません。  
1 : IO ポート I をリセットします。
- ビット 7 GPIOHRST** : IO ポート H リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート H をリセットしません。  
1 : IO ポート H をリセットします。
- ビット 6 GPIOGRST** : IO ポート G リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート G をリセットしません。  
1 : IO ポート G をリセットします。
- ビット 5 GPIOFRST** : IO ポート F リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート F をリセットしません。  
1 : IO ポート F をリセットします。
- ビット 4 GPIOERST** : IO ポート E リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート E をリセットしません。  
1 : IO ポート E をリセットします。
- ビット 3 GPIODRST** : IO ポート D リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート D をリセットしません。  
1 : IO ポート D をリセットします。
- ビット 2 GPIOCRST** : IO ポート C リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート C をリセットしません。  
1 : IO ポート C をリセットします。
- ビット 1 GPIOBRST** : IO ポート B リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート B をリセットしません。  
1 : IO ポート B をリセットします。
- ビット 0 GPIOARST** : IO ポート A リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート A をリセットしません。  
1 : IO ポート A をリセットします。

## 5.3.6 RCC AHB2 ペリフェラルリセットレジスタ (RCC\_AHB2RSTR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

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

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTGFSRST** : USB OTG FS モジュールリセット  
ソフトウェアでセット/クリアされます。  
0 : USB OTG FS モジュールをリセットしません。  
1 : USB OTG FS モジュールをリセットします。

ビット 6 **RNGRST** : 乱数発生回路モジュールリセット  
ソフトウェアでセット/クリアされます。  
0 : 乱数発生回路モジュールをリセットしません。  
1 : 乱数発生回路モジュールをリセットします。

ビット 5 **HASHRST** : ハッシュモジュールリセット  
ソフトウェアでセット/クリアされます。  
0 : ハッシュモジュールをリセットしません。  
1 : ハッシュモジュールをリセットします。

ビット 4 **CRYPST** : 暗号モジュールリセット  
ソフトウェアでセット/クリアされます。  
0 : 暗号モジュールをリセットしません。  
1 : 暗号モジュールをリセットします。

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **DCMIRST** : カメラインタフェースリセット  
ソフトウェアでセット/クリアされます。  
0 : カメラインタフェースをリセットしません。  
1 : カメラインタフェースをリセットします。

## 5.3.7 RCC AHB3 ペリフェラルリセットレジスタ (RCC\_AHB3RSTR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

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

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **QSPIRST** : Quad SPI メモリコントローラリセット

ソフトウェアでセット/クリアされます。

0 : Quad SPI メモリコントローラをリセットしません。

1 : Quad SPI メモリコントローラをリセットします。

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

ソフトウェアでセット/クリアされます。

0 : FMC モジュールをリセットしません。

1 : FMC モジュールをリセットします。

## 5.3.8 RCC APB1 ペリフェラルリセットレジスタ (RCC\_APB1RSTR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8RST	UART7RST	DACRST	PWR RST	CECRST	CAN2 RST	CAN1 RST	I2C4RST	I2C3 RST	I2C2 RST	I2C1 RST	UART5 RST	UART4 RST	UART3 RST	UART2 RST	SPDIFRX RST
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Res.	Res.	WWDG RST	Res.	LPTIM1RST	TIM14 RST	TIM13 RST	TIM12 RST	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31 **UART8RST** : UART8 リセット  
ソフトウェアでセット/クリアされます。  
0 : UART8 をリセットしません。  
1 : UART8 をリセットします。
- ビット 30 **UART7RST** : UART7 リセット  
ソフトウェアでセット/クリアされます。  
0 : UART7 をリセットしません。  
1 : UART7 をリセットします。
- ビット 29 **DACRST** : DAC リセット  
ソフトウェアでセット/クリアされます。  
0 : DACインタフェースをリセットしません。  
1 : DACインタフェースをリセットします。
- ビット 28 **PWRRST** : 電源インタフェースリセット  
ソフトウェアでセット/クリアされます。  
0 : 電源インタフェースをリセットしません。  
1 : 電源インタフェースをリセットします。
- ビット 27 **CECRST** : HDMI-CEC リセット  
ソフトウェアでセット/クリアされます。  
0 : HDMI-CEC をリセットしません。  
1 : HDMI-CEC をリセットします。
- ビット 26 **CAN2RST** : CAN2 リセット  
ソフトウェアでセット/クリアされます。  
0 : CAN2 をリセットしません。  
1 : CAN2 をリセットします。
- ビット 25 **CAN1RST** : CAN1 リセット  
ソフトウェアでセット/クリアされます。  
0 : CAN1 をリセットしません。  
1 : CAN1 をリセットします。
- ビット 24 **I2C4RST** : I2C4 リセット  
ソフトウェアでセット/クリアされます。  
0 : I2C4 をリセットしません。  
1 : I2C4 をリセットします。
- ビット 23 **I2C3RST** : I2C3 リセット  
ソフトウェアでセット/クリアされます。  
0 : I2C3 をリセットしません。  
1 : I2C3 をリセットします。
- ビット 22 **I2C2RST** : I2C2 リセット  
ソフトウェアでセット/クリアされます。  
0 : I2C2 をリセットしません。  
1 : I2C2 をリセットします。
- ビット 21 **I2C1RST** : I2C1 リセット  
ソフトウェアでセット/クリアされます。  
0 : I2C1 をリセットしません。  
1 : I2C1 をリセットします。
- ビット 20 **UART5RST** : UART5 リセット  
ソフトウェアでセット/クリアされます。  
0 : UART5 をリセットしません。  
1 : UART5 をリセットします。

- ビット 19 **UART4RST** : USART4 リセット  
ソフトウェアでセット/クリアされます。  
0 : USART4 をリセットしません。  
1 : USART4 をリセットします。
- ビット 18 **USART3RST** : USART3 リセット  
ソフトウェアでセット/クリアされます。  
0 : USART3 をリセットしません。  
1 : USART3 をリセットします。
- ビット 17 **USART2RST** : USART2 リセット  
ソフトウェアでセット/クリアされます。  
0 : USART2 をリセットしません。  
1 : USART2 をリセットします。
- ビット 16 **SPDIFRXRST** : SPDIFRX リセット  
ソフトウェアでセット/クリアされます。  
0 : SPDIFRX をリセットしません。  
1 : SPDIFRX をリセットします。
- ビット 15 **SPI3RST** : SPI3 リセット  
ソフトウェアでセット/クリアされます。  
0 : SPI3 をリセットしません。  
1 : SPI3 をリセットします。
- ビット 14 **SPI2RST** : SPI2 リセット  
ソフトウェアでセット/クリアされます。  
0 : SPI2 をリセットしません。  
1 : SPI2 をリセットします。
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGRST** : ウィンドウ型ウォッチドッグリセット  
ソフトウェアでセット/クリアされます。  
0 : ウィンドウ型ウォッチドッグをリセットしません。  
1 : ウィンドウ型ウォッチドッグをリセットします。
- ビット 10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **LPTIM1RST** : 低電力タイマ 1 リセット  
ソフトウェアでセット/クリアされます。  
0 : LPTMI1 をリセットしません。  
1 : LPTMI1 をリセットします。
- ビット 8 **TIM14RST** : TIM14 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM14 をリセットしません。  
1 : TIM14 をリセットします。
- ビット 7 **TIM13RST** : TIM13 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM13 をリセットしません。  
1 : TIM13 をリセットします。
- ビット 6 **TIM12RST** : TIM12 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM12 をリセットしません。  
1 : TIM12 をリセットします。

- ビット 5 **TIM7RST** : TIM7 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM7 をリセットしません。  
1 : TIM7 をリセットします。
- ビット 4 **TIM6RST** : TIM6 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM6 をリセットしません。  
1 : TIM6 をリセットします。
- ビット 3 **TIM5RST** : TIM5 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM5 をリセットしません。  
1 : TIM5 をリセットします。
- ビット 2 **TIM4RST** : TIM4 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM4 をリセットしません。  
1 : TIM4 をリセットします。
- ビット 1 **TIM3RST** : TIM3 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM3 をリセットしません。  
1 : TIM3 をリセットします。
- ビット 0 **TIM2RST** : TIM2 リセット  
ソフトウェアでセット/クリアされます。  
0 : TIM2 をリセットしません。  
1 : TIM2 をリセットします。

## 5.3.9 RCC APB2 ペリフェラルリセットレジスタ (RCC\_APB2RSTR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	LTDCR ST	Res.	Res.	SAI2R ST	SAI1 RST	SPI6 RST	SPI5 RST	Res.	TIM11 RST	TIM10 RST	TIM9 RST
					rw			rw	rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SYSCFG RST	SPI4 RST	SPI1 RST	SDMMC1R ST	Res.	Res.	ADC RST	Res.	Res.	USART6 RST	USART1 RST	Res.	Res.	TIM8 RST	TIM1 RST
	rw	rw	rw	rw			rw			rw	rw			rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **LTDCRST** : LTDC リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : LCD-TFT をリセットしません。

1 : LCD-TFT をリセットします。

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **SAI2RST** : SAI2 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAI2 をリセットしません。

1 : SAI2 をリセットします。

ビット 22 **SAI1RST** : SAI1 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAI1 をリセットしません。

1 : SAI1 をリセットします。

ビット 21 **SPI6RST** : SPI6 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : SPI6 をリセットしません。

1 : SPI6 をリセットします。

ビット 20 **SPI5RST** : SPI5 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : SPI5 をリセットしません。

1 : SPI5 をリセットします。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11RST** : TIM11 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : TIM11 をリセットしません。

1 : TIM11 をリセットします。

ビット 17 **TIM10RST** : TIM10 リセット

このビットは、ソフトウェアによってセット/クリアされます。

0 : TIM10 をリセットしません。

1 : TIM10 をリセットします。

- ビット 16 **TIM9RST** : TIM9 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM9 をリセットしません。  
1 : TIM9 をリセットします。
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **SYSCFGRST** : システム設定コントローラリセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : システム設定コントローラをリセットしません。  
1 : システム設定コントローラをリセットします。
- ビット 13 **SPI4RST** : SPI4 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPI4 をリセットしません。  
1 : SPI4 をリセットします。
- ビット 12 **SPI1RST** : SPI1 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPI1 をリセットしません。  
1 : SPI1 をリセットします。
- ビット 11 **SDMMC1RST** : SDMMC1 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SDMMC1 モジュールをリセットしません。  
1 : SDMMC1 モジュールをリセットします。
- ビット 10:9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **ADCRST** : ADC インタフェースリセット (すべての ADC に共通)  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ADC インタフェースをリセットしません。  
1 : ADC インタフェースをリセットします。
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **USART6RST** : USART6 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USART6 をリセットしません。  
1 : USART6 をリセットします。
- ビット 4 **USART1RST** : USART1 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USART1 をリセットしません。  
1 : USART1 をリセットします。
- ビット 3:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **TIM8RST** : TIM8 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM8 をリセットしません。  
1 : TIM8 をリセットします。
- ビット 0 **TIM1RST** : TIM1 リセット  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM1 をリセットしません。  
1 : TIM1 をリセットします。

## 5.3.10 RCC AHB1 ペリフェラルクロックレジスタ (RCC\_AHB1ENR)

アドレスオフセット : 0x30

リセット値 : 0x0010 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OTGHS ULPIEN	OTGHSE N	ETHMA CPTPEN	ETHMA CRXEN	ETHMA CTXEN	ETHMAC EN	Res.	DMA2DE N	DMA2 EN	DMA1 EN	DTCMRAM EN	Res.	BKPSRA MEN	Res.	Res.
	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC EN	Res.	GPIOK EN	GPIOJ EN	GPIOIE N	GPIOHE N	GPIOGE N	GPIOF EN	GPIOE EN	GPIODE N	GPIOCE N	GPIOB EN	GPIOA EN
			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **OTGHSULPIEN** : USB OTG HSULPI クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : USB OTG HS ULPI クロック無効

1 : USB OTG HS ULPI クロック有効

ビット 29 **OTGHSEN** : USB OTG HS クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : USB OTG HS クロック無効

1 : USB OTG HS クロック有効

ビット 28 **ETHMACPTPEN** : Ethernet PTP クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : Ethernet PTP クロック無効

1 : Ethernet PTP クロック有効

ビット 27 **ETHMACRXEN** : Ethernet 受信クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : Ethernet 受信クロック無効

1 : Ethernet 受信クロック有効

ビット 26 **ETHMACTXEN** : Ethernet 送信クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : Ethernet 送信クロック無効

1 : Ethernet 送信クロック有効

ビット 25 **ETHMACEN** : Ethernet MAC クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : Ethernet MAC クロック無効

1 : Ethernet MAC クロック有効

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **DMA2DEN** : DMA2D クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : DMA2D クロック無効

1 : DMA2D クロック有効

- ビット 22 **DMA2EN** : DMA2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : DMA2 クロック無効  
1 : DMA2 クロック有効
- ビット 21 **DMA1EN** : DMA1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : DMA1 クロック無効  
1 : DMA1 クロック有効
- ビット 20 **DTCMRAMEN** : DTCM データ RAM クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : DTCM データ RAM クロック無効  
1 : DTCM データ RAM クロック有効
- ビット 19 予約済みであり、リセット値に保持する必要があります。
- ビット 18 **BKPSRAMEN** : バックアップ SRAM インタフェースクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : バックアップ SRAM インタフェースクロック無効  
1 : バックアップ SRAM インタフェースクロック有効
- ビット 17:13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **CRCEN** : CRC クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : CRC クロック無効  
1 : CRC クロック有効
- ビット 11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **GPIOKEN** : IO ポート K クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート K クロック無効  
1 : IO ポート K クロック有効
- ビット 9 **GPIOJEN** : IO ポート J クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート J クロック無効  
1 : IO ポート J クロック有効
- ビット 8 **GPIOEN** : IO ポート I クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート I クロック無効  
1 : IO ポート I クロック有効
- ビット 7 **GPIOHEN** : IO ポート H クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート H クロック無効  
1 : IO ポート H クロック有効
- ビット 6 **GPIOGEN** : IO ポート G クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート G クロック無効  
1 : IO ポート G クロック有効
- ビット 5 **GPIOFEN** : IO ポート F クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート F クロック無効  
1 : IO ポート F クロック有効

- ビット 4 **GPIOEEN** : IO ポート E クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート E クロック無効  
1 : IO ポート E クロック有効
- ビット 3 **GPIODEN** : IO ポート D クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート D クロック無効  
1 : IO ポート D クロック有効
- ビット 2 **GPIOCEN** : IO ポート C クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート C クロック無効  
1 : IO ポート C クロック有効
- ビット 1 **GPIOBEN** : IO ポート B クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート B クロック無効  
1 : IO ポート B クロック有効
- ビット 0 **GPIOAEN** : IO ポート A クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : IO ポート A クロック無効  
1 : IO ポート A クロック有効

### 5.3.11 RCC AHB2 ペリフェラルクロック有効レジスタ (RCC\_AHB2ENR)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

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

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

- ビット 7 **OTGFSEN** : USB OTG FS クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USB OTG FS クロック無効  
1 : USB OTG FS クロック有効
- ビット 6 **RNGEN** : 乱数発生回路クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : 乱数発生回路クロック無効  
1 : 乱数発生回路クロック有効
- ビット 5 **HASHEN** : ハッシュモジュールクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ハッシュモジュールクロック無効  
1 : ハッシュモジュールクロック有効

ビット 4 **CRYPEN** : 暗号モジュールクロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 暗号モジュールクロック無効  
 1 : 暗号モジュールクロック有効

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **DCMIEN** : カメラインタフェース有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : カメラインタフェースクロック無効  
 1 : カメラインタフェースクロック有効

### 5.3.12 RCC AHB3 ペリフェラルクロック有効レジスタ (RCC\_AHB3ENR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

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

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **QSPIEN** : Quad SPI メモリコントローラクロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : Quad SPI コントローラクロック無効  
 1 : Quad SPI コントローラクロック有効

ビット 0 **FMCEN** : フレキシブルメモリコントローラクロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : FMC クロック無効  
 1 : FMC クロック有効

## 5.3.13 RCC APB1 ペリフェラルクロック有効レジスタ (RCC\_APB1ENR)

アドレスオフセット : 0x40

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8EN	UART7EN	DACEN	PWREN	CECEN	CAN2EN	CAN1EN	I2C4EN	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	SPDIFRXEN
rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3EN	SPI2EN	Res.	Res.	WWWDGEN	Res.	LPTIM1EN	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
rw	rw			rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **UART8EN** : UART8 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : UART8 クロック無効

1 : UART8 クロック有効

ビット 30 **UART7EN** : UART7 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : UART7 クロック無効

1 : UART7 クロック有効

ビット 29 **DACEN** : DAC インタフェースクロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC インタフェースクロック無効

1 : DAC インタフェースクロック有効

ビット 28 **PWREN** : 電源インタフェースクロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : 電源インタフェースクロック無効

1 : 電源インタフェースクロック有効

ビット 27 **CECEN** : HDMI-CEC クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : HDMI-CEC クロック無効

1 : HDMI-CEC クロック有効

ビット 26 **CAN2EN** : CAN 2 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : CAN 2 クロック無効

1 : CAN 2 クロック有効

ビット 25 **CAN1EN** : CAN 1 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : CAN 1 クロック無効

1 : CAN 1 クロック有効

ビット 24 **I2C4** : I2C4 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : I2C4 クロック無効

1 : I2C4 クロック有効

- ビット 23 **I2C3EN** : I2C3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : I2C3 クロック無効  
1 : I2C3 クロック有効
- ビット 22 **I2C2EN** : I2C2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : I2C2 クロック無効  
1 : I2C2 クロック有効
- ビット 21 **I2C1EN** : I2C1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : I2C1 クロック無効  
1 : I2C1 クロック有効
- ビット 20 **UART5EN** : UART5 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : UART5 クロック無効  
1 : UART5 クロック有効
- ビット 19 **UART4EN** : UART4 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : UART4 クロック無効  
1 : UART4 クロック有効
- ビット 18 **USART3EN** : USART3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USART3 クロック無効  
1 : USART3 クロック有効
- ビット 17 **USART2EN** : USART2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USART2 クロック無効  
1 : USART2 クロック有効
- ビット 16 **SPDIFRXEN** : SPDIFRX クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPDIFRX クロック無効  
1 : SPDIFRX クロック有効
- ビット 15 **SPI3EN** : SPI3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPI3 クロック無効  
1 : SPI3 クロック有効
- ビット 14 **SPI2EN** : SPI2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPI2 クロック無効  
1 : SPI2 クロック有効
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGEN** : ウィンドウ型ウォッチドッグクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ウィンドウ型ウォッチドッグクロック無効  
1 : ウィンドウ型ウォッチドッグクロック有効
- ビット 10 予約済みであり、リセット値に保持する必要があります。

- ビット 9 **LPTMI1EN** : 低電力タイマ 1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : LPTIM1 クロック無効  
1 : LPTIM1 クロック有効
- ビット 8 **TIM14EN** : TIM14 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM14 クロック無効  
1 : TIM14 クロック有効
- ビット 7 **TIM13EN** : TIM13 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM13 クロック無効  
1 : TIM13 クロック有効
- ビット 6 **TIM12EN** : TIM12 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM12 クロック無効  
1 : TIM12 クロック有効
- ビット 5 **TIM7EN** : TIM7 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM7 クロック無効  
1 : TIM7 クロック有効
- ビット 4 **TIM6EN** : TIM6 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM6 クロック無効  
1 : TIM6 クロック有効
- ビット 3 **TIM5EN** : TIM5 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM5 クロック無効  
1 : TIM5 クロック有効
- ビット 2 **TIM4EN** : TIM4 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM4 クロック無効  
1 : TIM4 クロック有効
- ビット 1 **TIM3EN** : TIM3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM3 クロック無効  
1 : TIM3 クロック有効
- ビット 0 **TIM2EN** : TIM2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM2 クロック無効  
1 : TIM2 クロック有効

## 5.3.14 RCC APB2 ペリフェラルクロック有効レジスタ (RCC\_APB2ENR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	LTDC EN	Res.	Res.	SAI2EN	SAI1EN	SPI6EN	SPI5EN	Res.	TIM11 EN	TIM10 EN	TIM9 EN
					rw			rw	rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SYSCFG EN	SPI4 EN	SPI1 EN	SDMMC1E N	ADC3 EN	ADC2 EN	ADC1 EN	Res.	Res.	USART6 EN	USART1 EN	Res.	Res.	TIM8 EN	TIM1 EN
	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **LTDCEN** : LTDC クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : LTDC クロック無効

1 : LTDC クロック有効

ビット 27 : 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **SAI2EN** : SAI2 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAI2 クロック無効

1 : SAI2 クロック有効

ビット 22 **SAI1EN** : SAI1 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAI1 クロック無効

1 : SAI1 クロック有効

ビット 21 **SPI6EN** : SPI6 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SPI6 クロック無効

1 : SPI6 クロック有効

ビット 20 **SPI5EN** : SPI5 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SPI5 クロック無効

1 : SPI5 クロック有効

ビット 18 **TIM11EN** : TIM11 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : TIM11 クロック無効

1 : TIM11 クロック有効

ビット 17 **TIM10EN** : TIM10 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : TIM10 クロック無効

1 : TIM10 クロック有効

- ビット 16 **TIM9EN** : TIM9 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TIM9 クロック無効  
1 : TIM9 クロック有効
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **SYSCFGEN** : システム設定コントローラクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : システム設定コントローラクロック無効  
1 : システム設定コントローラクロック有効
- ビット 13 **SPI4EN** : SPI4 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPI4 クロック無効  
1 : SPI4 クロック有効
- ビット 12 **SPI1EN** : SPI1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SPI1 クロック無効  
1 : SPI1 クロック有効
- ビット 11 **SDMMC1EN** : SDMMC1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SDMMC1 モジュールクロック無効  
1 : SDMMC1 モジュールクロック有効
- ビット 10 **ADC3EN** : ADC3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ADC3 クロック無効  
1 : ADC3 クロック有効
- ビット 9 **ADC2EN** : ADC2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ADC2 クロック無効  
1 : ADC2 クロック有効
- ビット 8 **ADC1EN** : ADC1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ADC1 クロック無効  
1 : ADC1 クロック有効
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **USART6EN** : USART6 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USART6 クロック無効  
1 : USART6 クロック有効
- ビット 4 **USART1EN** : USART1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : USART1 クロック無効  
1 : USART1 クロック有効

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TIM8EN** : TIM8 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : TIM8 クロック無効

1 : TIM8 クロック有効

ビット 0 **TIM1EN** : TIM1 クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : TIM1 クロック無効

1 : TIM1 クロック有効

### 5.3.15 低電力モードにおける RCC AHB1 ペリフェラルクロック有効レジスタ (RCC\_AHB1LPENR)

アドレスオフセット : 0x50

リセット値 : 0x7EF7 B7FFh

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OTGHS ULPI LPEN	OTGHS LPEN	ETHPTP LPEN	ETHRX LPEN	ETHTX LPEN	ETHMAC LPEN	Res.	DMA2D LPEN	DMA2 LPEN	DMA1 LPEN	DTCML PEN	Res.	BKPSR AM LPEN	SRAM2 LPEN	SRAM1 LPEN
	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW		rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLITF LPEN	Res.	AXI LPEN	CRC LPEN	Res.	GPIOK LPEN	GPIOJ LPEN	GPIOI LPEN	GPIOH LPEN	GPIOGG LPEN	GPIOF LPEN	GPIOE LPEN	GPIOD LPEN	GPIOC LPEN	GPIOB LPEN	GPIOA LPEN
rW		rW	rW		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **OTGHSULPIPEN** : SLEEP モード時 USB OTG HS ULPI クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 USB OTG HS ULPI クロック無効

1 : SLEEP モード時 USB OTG HS ULPI クロック有効

ビット 29 **OTGHSLPEN** : SLEEP モード時 USB OTG HS クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 USB OTG HS クロック無効

1 : SLEEP モード時 USB OTG HS クロック有効

ビット 28 **ETHMACPTPLPEN** : SLEEP モード時 Ethernet PTP クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 Ethernet PTP クロック無効

1 : SLEEP モード時 Ethernet PTP クロック有効

ビット 27 **ETHMACRXLPEN** : SLEEP モード時 Ethernet 受信クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 Ethernet 受信クロック無効

1 : SLEEP モード時 Ethernet 受信クロック有効

ビット 26 **ETHMACTXLPEN** : SLEEP モード時 Ethernet 送信クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 Ethernet 送信クロック無効

1 : SLEEP モード時 Ethernet 送信クロック有効



- ビット 25 **ETHMACLPEN** : SLEEP モード時 Ethernet MAC クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 Ethernet MAC クロック無効  
1 : SLEEP モード時 Ethernet MAC クロック有効
- ビット 24 予約済みであり、リセット値に保持する必要があります。
- ビット 23 **DMA2DLPEN** : SLEEP モード時 DMA2D クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 DMA2D クロック無効  
1 : SLEEP モード時 DMA2D クロック有効
- ビット 22 **DMA2LPEN** : SLEEP モード時 DMA2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 DMA2 クロック無効  
1 : SLEEP モード時 DMA2 クロック有効
- ビット 21 **DMA1LPEN** : SLEEP モード時 DMA1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 DMA1 クロック無効  
1 : SLEEP モード時 DMA1 クロック有効
- ビット 20 **DTCMLPEN** : SLEEP モード時 DTCM RAM インタフェースクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 DTCM RAM インタフェースクロック無効  
1 : SLEEP モード時 DTCM RAM インタフェースクロック有効
- ビット 19 予約済みであり、リセット値に保持する必要があります。
- ビット 18 **BKPSRAMLPEN** : SLEEP モード時バックアップ SRAM インタフェースクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時バックアップ SRAM インタフェースクロック無効  
1 : SLEEP モード時バックアップ SRAM インタフェースクロック有効
- ビット 17 **SRAM2LPEN** : SLEEP モード時 SRAM 2 インタフェースクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SRAM2 インタフェースクロック無効  
1 : SLEEP モード時 SRAM2 インタフェースクロック有効
- ビット 16 **SRAM1LPEN** : SLEEP モード時 SRAM 1 インタフェースクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SRAM1 インタフェースクロック無効  
1 : SLEEP モード時 SRAM1 インタフェースクロック有効
- ビット 15 **FLITFLPEN** : SLEEP モード時フラッシュインタフェースクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時フラッシュインタフェースクロック無効  
1 : SLEEP モード時フラッシュインタフェースクロック有効
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **AXILPEN** : SLEEP モード時 AXI/AHB ブリッジクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 AXI/AHB ブリッジクロック無効  
1 : SLEEP モード時 AXI/AHB ブリッジクロック有効
- ビット 12 **CRCLPEN** : SLEEP モード時 CRC クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 CRC クロック無効  
1 : SLEEP モード時 CRC クロック有効

- ビット 11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **GPIOKLPEN** : SLEEP モード時 IO ポート K クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート K クロック無効  
1 : SLEEP モード時 IO ポート K クロック有効
- ビット 9 **GPIOJLPEN** : SLEEP モード時 IO ポート J クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート J クロック無効  
1 : SLEEP モード時 IO ポート J クロック有効
- ビット 8 **GPIOILPEN** : SLEEP モード時 IO ポート I クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート I クロック無効  
1 : SLEEP モード時 IO ポート I クロック有効
- ビット 7 **GPIOHLPEN** : SLEEP モード時 IO ポート H クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート H クロック無効  
1 : SLEEP モード時 IO ポート H クロック有効
- ビット 6 **GPIOGLPEN** : SLEEP モード時 IO ポート G クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート G クロック無効  
1 : SLEEP モード時 IO ポート G クロック有効
- ビット 5 **GPIOFLPEN** : SLEEP モード時 IO ポート F クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート F クロック無効  
1 : SLEEP モード時 IO ポート F クロック有効
- ビット 4 **GPIOELPEN** : SLEEP モード時 IO ポート E クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : SLEEP モード時 IO ポート E クロック無効  
1 : SLEEP モード時 IO ポート E クロック有効
- ビット 3 **GPIODLPEN** : SLEEP モード時 IO ポート D クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート D クロック無効  
1 : SLEEP モード時 IO ポート D クロック有効
- ビット 2 **GPIOCLPEN** : SLEEP モード時 IO ポート C クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート C クロック無効  
1 : SLEEP モード時 IO ポート C クロック有効
- ビット 1 **GPIOBLPEN** : SLEEP モード時 IO ポート B クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート B クロック無効  
1 : SLEEP モード時 IO ポート B クロック有効
- ビット 0 **GPIOALPEN** : SLEEP モード時 IO ポート A クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 IO ポート A クロック無効  
1 : SLEEP モード時 IO ポート A クロック有効

## 5.3.16 低電力モードにおける RCC AHB2 ペリフェラルクロック有効レジスタ (RCC\_AHB2LPENR)

アドレスオフセット : 0x54

リセット値 : 0x0000 00F1

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

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTGFSLPEN** : SLEEP モード時 USB OTG FS クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 USB OTG FS クロック無効

1 : SLEEP モード時 USB OTG FS クロック有効

ビット 6 **RNGLPEN** : SLEEP モード時乱数発生回路クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時乱数発生回路クロック無効

1 : SLEEP モード時乱数発生回路クロック有効

ビット 5 **HASHLPEN** : SLEEP モード時ハッシュモジュールクロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時ハッシュモジュールクロック無効

1 : SLEEP モード時ハッシュモジュールクロック有効

ビット 4 **CRYPLPEN** : SLEEP モード時暗号モジュールクロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時暗号モジュールクロック無効

1 : SLEEP モード時暗号モジュールクロック有効

ビット 3:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **DCMILPEN** : SLEEP モード時カメラインタフェース有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時カメラインタフェースクロック無効

1 : SLEEP モード時カメラインタフェースクロック有効

## 5.3.17 低電力モードにおける RCC AHB3 ペリフェラルクロック有効レジスタ (RCC\_AHB3LPENR)

アドレスオフセット : 0x58

リセット値 : 0x0000 0003

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

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **QSPILPEN** : SLEEP モード時Quad SPI メモリコントローラクロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時Quad SPI コントローラクロック無効

1 : SLEEP モード時Quad SPI コントローラクロック有効

ビット 0 **FMCLPEN** : SLEEP モード時フレキシブルメモリコントローラモジュールクロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 FMC モジュールクロック無効

1 : SLEEP モード時 FMC モジュールクロック有効

## 5.3.18 低電力モードにおける RCC APB1 ペリフェラルクロック有効レジスタ (RCC\_APB1LPENR)

アドレスオフセット : 0x60

リセット値 : 0xFFFF CBFFh

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8 LPEN	UART7 LPEN	DAC LPEN	PWR LPEN	CEC LPEN	CAN2 LPEN	CAN1 LPEN	I2C4 LPEN	I2C3 LPEN	I2C2 LPEN	I2C1 LPEN	UART5 LPEN	UART4 LPEN	USART3 LPEN	USART2 LPEN	SPDIFRX LPEN
rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 LPEN	SPI2 LPEN	Res.	Res.	WWDG LPEN	Res.	LPTMI1 LPEN	TIM14 LPEN	TIM13 LPEN	TIM12 LPEN	TIM7 LPEN	TIM6 LPEN	TIM5 LPEN	TIM4 LPEN	TIM3 LPEN	TIM2 LPEN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31 **UART8LPEN** : SLEEP モード時 UART8 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 UART8 クロック無効  
 1 : SLEEP モード時 UART8 クロック有効
- ビット 30 **UART7LPEN** : SLEEP モード時 UART7 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 UART7 クロック無効  
 1 : SLEEP モード時 UART7 クロック有効
- ビット 29 **DACLPEN** : SLEEP モード時 DAC インタフェースクロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 DAC インタフェースクロック無効  
 1 : SLEEP モード時 DAC インタフェースクロック有効
- ビット 28 **PWRLPEN** : SLEEP モード時電源インタフェースクロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時電源インタフェースクロック無効  
 1 : SLEEP モード時電源インタフェースクロック有効
- ビット 27 **CECLPEN** : SLEEP モード時 HDMI-CEC クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 HDMI-CEC クロック無効  
 1 : SLEEP モード時 HDMI-CEC クロック有効
- ビット 26 **CAN2LPEN** : SLEEP モード時 CAN 2 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 CAN 2 クロック無効  
 1 : SLEEP モード時 CAN 2 クロック有効
- ビット 25 **CAN1LPEN** : SLEEP モード時 CAN 1 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 CAN 1 クロック無効  
 1 : SLEEP モード時 CAN 1 クロック有効
- ビット 24 **I2C4LPEN** : SLEEP モード時 I2C4 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 I2C4 クロック無効  
 1 : SLEEP モード時 I2C4 クロック有効

- ビット 23 **I2C3LPEN** : SLEEP モード時 I2C3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 I2C3 クロック無効  
1 : SLEEP モード時 I2C3 クロック有効
- ビット 22 **I2C2LPEN** : SLEEP モード時 I2C2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 I2C2 クロック無効  
1 : SLEEP モード時 I2C2 クロック有効
- ビット 21 **I2C1LPEN** : SLEEP モード時 I2C1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 I2C1 クロック無効  
1 : SLEEP モード時 I2C1 クロック有効
- ビット 20 **UART5LPEN** : SLEEP モード時 UART5 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 UART5 クロック無効  
1 : SLEEP モード時 UART5 クロック有効
- ビット 19 **UART4LPEN** : SLEEP モード時 UART4 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 UART4 クロック無効  
1 : SLEEP モード時 UART4 クロック有効
- ビット 18 **USART3LPEN** : SLEEP モード時 USART3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 USART3 クロック無効  
1 : SLEEP モード時 USART3 クロック有効
- ビット 17 **USART2LPEN** : SLEEP モード時 USART2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 USART2 クロック無効  
1 : SLEEP モード時 USART2 クロック有効
- ビット 16 **SPDIFRXLPEN** : SLEEP モード時 SPDIFRX クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SPDIFRX クロック無効  
1 : SLEEP モード時 SPDIFRX クロック有効
- ビット 15 **SPI3LPEN** : SLEEP モード時 SPI3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SPI3 クロック無効  
1 : SLEEP モード時 SPI3 クロック有効
- ビット 14 **SPI2LPEN** : SLEEP モード時 SPI2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SPI2 クロック無効  
1 : SLEEP モード時 SPI2 クロック有効
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGLPEN** : SLEEP モード時ウィンドウ型ウォッチドッグクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時ウィンドウ型ウォッチドッグクロック無効  
1 : SLEEP モード時ウィンドウ型ウォッチドッグクロック有効
- ビット 10 予約済みであり、リセット値に保持する必要があります。

- ビット 9 **LPTIM1LPEN** : SLEEP モード時の低電力タイマ 1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 LPTIM1 クロック無効  
1 : SLEEP モード時 LPTIM1 クロック有効
- ビット 8 **TIM14LPEN** : SLEEP モード時 TIM14 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM14 クロック無効  
1 : SLEEP モード時 TIM14 クロック有効
- ビット 7 **TIM13LPEN** : SLEEP モード時 TIM13 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM13 クロック無効  
1 : SLEEP モード時 TIM13 クロック有効
- ビット 6 **TIM12LPEN** : SLEEP モード時 TIM12 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM12 クロック無効  
1 : SLEEP モード時 TIM12 クロック有効
- ビット 5 **TIM7LPEN** : SLEEP モード時 TIM7 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM7 クロック無効  
1 : SLEEP モード時 TIM7 クロック有効
- ビット 4 **TIM6LPEN** : SLEEP モード時 TIM6 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM6 クロック無効  
1 : SLEEP モード時 TIM6 クロック有効
- ビット 3 **TIM5LPEN** : SLEEP モード時 TIM5 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM5 クロック無効  
1 : SLEEP モード時 TIM5 クロック有効
- ビット 2 **TIM4LPEN** : SLEEP モード時 TIM4 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM4 クロック無効  
1 : SLEEP モード時 TIM4 クロック有効
- ビット 1 **TIM3LPEN** : SLEEP モード時 TIM3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM3 クロック無効  
1 : SLEEP モード時 TIM3 クロック有効
- ビット 0 **TIM2LPEN** : SLEEP モード時 TIM2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM2 クロック無効  
1 : SLEEP モード時 TIM2 クロック有効

## 5.3.19 低電力モードにおける RCC APB2 ペリフェラルクロック有効レジスタ (RCC\_APB2LPENR)

アドレスオフセット : 0x64

リセット値 : 0x04F7 7F33h

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	LTDC LPEN	Res.	Res.	SAI2 LPEN	SAI1 LPEN	SPI6 LPEN	SPI5 LPEN	Res.	TIM11 LPEN	TIM10 LPEN	TIM9 LPEN
					rw			rw	rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SYSCFG LPEN	SPI4 LPEN	SPI1 LPEN	SDMMC1 LPEN	ADC3 LPEN	ADC2 LPEN	ADC1 LPEN	Res.	Res.	USART6 LPEN	USART1 LPEN	Res.	Res.	TIM8 LPEN	TIM1 LPEN
	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **LTDC LPEN** : SLEEP モード時 LTDC クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 LTDC クロック無効  
 1 : SLEEP モード時 LTDC クロック有効

ビット 25:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **SAI2 LPEN** : SLEEP モード時 SAI2 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 SAI2 クロック無効  
 1 : SLEEP モード時 SAI2 クロック有効

ビット 22 **SAI1 LPEN** : SLEEP モード時 SAI1 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 SAI1 クロック無効  
 1 : SLEEP モード時 SAI1 クロック有効

ビット 21 **SPI6 LPEN** : SLEEP モード時 SPI6 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 SPI6 クロック無効  
 1 : SLEEP モード時 SPI6 クロック有効

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

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11 LPEN** : SLEEP モード時 TIM11 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 TIM11 クロック無効  
 1 : SLEEP モード時 TIM11 クロック有効

ビット 17 **TIM10 LPEN** : SLEEP モード時 TIM10 クロック有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : SLEEP モード時 TIM10 クロック無効  
 1 : SLEEP モード時 TIM10 クロック有効



- ビット 16 **TIM9LPEN** : SLEEP モード時 TIM9 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 TIM9 クロック無効  
1 : SLEEP モード時 TIM9 クロック有効
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **SYSCFGLPEN** : SLEEP モード時システム設定コントローラクロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時システム設定コントローラクロック無効  
1 : SLEEP モード時システム設定コントローラクロック有効
- ビット 13 **SPI4LPEN** : SLEEP モード時 SPI4 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SPI4 クロック無効  
1 : SLEEP モード時 SPI4 クロック有効
- ビット 12 **SPI1LPEN** : SLEEP モード時 SPI1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SPI1 クロック無効  
1 : SLEEP モード時 SPI1 クロック有効
- ビット 11 **SDMMC1LPEN** : SLEEP モード時 SDMMC1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 SDMMC1 モジュールクロック無効  
1 : SLEEP モード時 SDMMC1 モジュールクロック有効
- ビット 10 **ADC3LPEN** : SLEEP モード時 ADC 3 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 ADC 3 クロック無効  
1 : SLEEP モード時 ADC 3 クロック有効
- ビット 9 **ADC2LPEN** : SLEEP モード時 ADC2 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 ADC2 クロック無効  
1 : SLEEP モード時 ADC2 クロック有効
- ビット 8 **ADC1LPEN** : SLEEP モード時 ADC1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 ADC1 クロック無効  
1 : SLEEP モード時 ADC1 クロック有効
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **USART6LPEN** : SLEEP モード時 USART6 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 USART6 クロック無効  
1 : SLEEP モード時 USART6 クロック有効
- ビット 4 **USART1LPEN** : SLEEP モード時 USART1 クロック有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : SLEEP モード時 USART1 クロック無効  
1 : SLEEP モード時 USART1 クロック有効

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

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

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 TIM8 クロック無効

1 : SLEEP モード時 TIM8 クロック有効

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

このビットは、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 TIM1 クロック無効

1 : SLEEP モード時 TIM1 クロック有効

## 5.3.20 RCC バックアップドメイン制御レジスタ (RCC\_BDCR)

アドレスオフセット : 0x70

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

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

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

**RCC バックアップドメイン制御レジスタ (RCC\_BDCR)** の LSEON、LSEBYP、RTCSEL、および RTCEN ビットは、バックアップドメインにあります。このため、リセット後、これらのビットは書き込み保護されるので、これらを変更するには、**PWR 電源制御レジスタ (PWR\_CR1)** の DBP ビットをセットする必要があります。詳細については、[セクション 5.1.1 : システムリセット \(131 ページ\)](#) を参照してください。これらのビットは、バックアップドメインリセット後にのみリセットされます ([セクション 5.1.3 : バックアップドメインリセット](#) を参照)。内部または外部リセットは、これらのビットに影響しません。

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

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

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

このビットは、ソフトウェアによってセット/クリアされます。

0 : リセットは有効ではありません。

1 : バックアップドメイン全体をリセットします。

**注 :** **BKPSRAM** はこのリセットに影響されません。**BKPSRAM** をリセットするには、フラッシュインタフェースを介して、保護レベル 1 から 0 への変更を要求する必要があります。

ビット 15 **RTCEN** : RTC クロック有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : RTC クロック無効

1 : RTC クロック有効

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

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

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

00 : クロックなし。

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

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

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

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

**ビット 4:3 LSEDRV[1:0] : LSE オシレータの駆動能力**

LSE オシレータの駆動能力を変調するためにソフトウェアによってセットされます。

00 : 低駆動

01 : 中低駆動

10 : 中高駆動

11 : 高駆動

**ビット 2 LSEBYP : 外部低速オシレータバイパス**

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

0 : LSE オシレータはバイパスされません。

1 : LSE オシレータはバイパスされます。

**ビット 1 LSERDY : 外部低速オシレータレディ**

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

0 : LSE クロックはレディ状態ではありません。

1 : LSE クロックはレディ状態です。

**ビット 0 LSEON : 外部低速オシレータ有効化**

このビットは、ソフトウェアによってセット/クリアされます。

0 : LSE クロックオフ

1 : LSE クロックオン

## 5.3.21 RCC クロック制御およびステータスレジスタ (RCC\_CSR)

アドレスオフセット : 0x74

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

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

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

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

### ビット 31 LPWRRSTF : 低電力リセットフラグ

このビットは、低電力管理リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : 低電力管理リセットは発生していません。

1 : 低電力管理リセットが発生しました。

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

### ビット 30 WWDGRSTF : ウィンドウ型ウォッチドッグリセットフラグ

このビットは、ウィンドウ型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : ウィンドウ型ウォッチドッグリセットは発生していません。

1 : ウィンドウ型ウォッチドッグリセットが発生しました。

### ビット 29 IWDGRSTF : 独立型ウォッチドッグリセットフラグ

このビットは、V<sub>DD</sub> ドメインから独立型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : ウォッチドッグリセットは発生していません。

1 : ウォッチドッグリセットが発生しました。

### ビット 28 SFTRSTF : ソフトウェアリセットフラグ

このビットは、ソフトウェアリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : ソフトウェアリセットは発生していません。

1 : ソフトウェアリセットが発生しました。

### ビット 27 PORRSTF : POR/PDR リセットフラグ

このビットは、POR/PDR リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : POR/PDR リセットは発生していません。

1 : POR/PDR リセットが発生しました。

### ビット 26 PINRSTF : PIN リセットフラグ

このビットは、NRST ピンからリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : NRST ピンからのリセットは発生していません。

1 : NRST ピンからリセットが発生しました。

ビット 25 **BORRSTF** : BOR リセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってクリアされます。  
 このビットは、POR/PDR リセットまたは BOR リセットが発生したときに、ハードウェアによってセットされます。  
 0 : POR/PDR リセットまたは BOR リセットは発生していません。  
 1 : POR/PDR リセットまたは BOR リセットが発生しました。

ビット 24 **RMVF** : リセットフラグ解除

このビットは、リセットフラグをクリアするために、ソフトウェアによってセットされます。  
 0 : 影響なし。  
 1 : リセットフラグをクリアします。

ビット 23:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **LSIRDY** : 内部低速オシレータレディ

このビットは、内部 RC 40 kHz オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSION ビットがクリアされた後、LSIRDY は、LSI オシレータクロックの 3 サイクル後にローになります。  
 0 : LSI RC オシレータはレディ状態ではありません。  
 1 : LSI RC オシレータはレディ状態です。

ビット 0 **LSION** : 内部低速オシレータ有効化

このビットは、ソフトウェアによってセット/クリアされます。  
 0 : LSI RC オシレータオフ  
 1 : LSI RC オシレータオン

## 5.3.22 RCC スペクトル拡散クロック生成レジスタ (RCC\_SSCGR)

アドレスオフセット : 0x80

リセット値 : 0x0000 0000

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

スペクトル拡散クロック生成は、メイン PLL に対してのみ利用可能です。

RCC\_SSCGR レジスタは、メイン PLL が有効になる前またはメイン PLL が無効になった後に書き込む必要があります。

**注 :** **PLL スペクトル拡散クロック生成 (SSCG) 特性の詳細については、デバイスのデータシートの電気特性のセクションを参照してください。**

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

ビット 31 **SSCGEN** : スペクトル拡散変調有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : スペクトル拡散変調無効 (書き込みは クリア後 : CR[24]=PLLON ビット)

1 : スペクトル拡散変調有効 (書き込みは セット前 : CR[24]=PLLON ビット)

ビット 30 **SPREADSEL** : 拡散選択

このビットは、ソフトウェアによってセット/クリアされます。

CR[24]=PLLON ビットをセットする前に書き込みます。

0 : センタースプレッド

1 : ダウンスプレッド

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:13 **INCSTEP** : インクリメントステップ

これらのビットは、ソフトウェアによってセット/クリアされます。CR[24]=PLLON ビットをセットする前に書き込みます。

変調プロファイルの振幅の設定入力です。

ビット 12:0 **MODPER** : 変調周期

これらのビットは、ソフトウェアによってセット/クリアされます。CR[24]=PLLON ビットをセットする前に書き込みます。

変調プロファイルの周期の設定入力です。

## 5.3.23 RCC PLLI2S 設定レジスタ (RCC\_PLLI2SCFGR)

アドレスオフセット : 0x84

リセット値 : 0x2400 3000

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

このレジスタは、次の式に従って PLLI2S クロック出力を設定するために使用します。

$$f(VCO \text{ クロック}) = f(PLLI2S \text{ クロック入力}) * (PLLI2SN / PLLM)$$

$$f(PLLI2S\_P) = f(VCO \text{ クロック}) / PLLI2SP$$

$$f(PLLI2S\_Q) = f(VCO \text{ クロック}) / PLLI2SQ$$

$$f(PLLI2S\_R) = f(VCO \text{ クロック}) / PLLI2SR$$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PLLI2S[2:0]			PLLI2SQ[0:3]				Res.	Res.	Res.	Res.	Res.	Res.	PLLI2SP[1:0]	
	r/w	r/w	r/w	r/w	r/w	r/w	r/w							r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PLLI2SN[8:0]									Res.	Res.	Res.	Res.	Res.	Res.
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w						

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **PLLI2SR[2:0]** : I2S クロックのための PLLI2S 分周比

これらのビットは、I2S クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。標準クリスタル使用時の誤差が 0.3 %、オーディオクリスタル使用時の誤差が 0 % となるよう、分周比は I2S ペリフェラル内のプリスケアラ値に従って選択する必要があります。I2S クロック周波数と精度に関する 詳細については、I2S の章の [セクション 32.7.3 : 起動に関する説明](#) を参照してください。

**注意 :** I2S が正しく動作するには、192 MHz 以下の周波数が必要です。

$I2S \text{ クロック周波数} = VCO \text{ 周波数} / PLLR$ 、ここで  $2 \leq PLLR \leq 7$

000 : PLLI2SR = 0、誤った設定

001 : PLLI2SR = 1、誤った設定

010 : PLLI2SR = 2

...

111 : PLLI2SR = 7

ビット 27:24 **PLLI2SQ[3:0]** : SAI クロックのための PLLI2S 分周比

これらのビットは、SAI クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットに書き込めるのは、PLLI2S が無効なときです。

$SAI \text{ クロック周波数} = VCO \text{ 周波数} / PLLI2SQ$ 、ここで  $2 \leq PLLI2SQ \leq 15$

0000 : PLLI2SQ = 0、誤った設定

0001 : PLLI2SQ = 1、誤った設定

0010 : PLLI2SQ = 2

0011 : PLLI2SQ = 3

0100 : PLLI2SQ = 4

0101 : PLLI2SQ = 5

...

1111 : PLLI2SQ = 15

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **PLLI2SP[1:0]** : SPDIFRX クロックのための PLLI2S 分周比

これらのビットは、SPDIFRX クロックを制御するために、ソフトウェアによってセット/クリアされます。これらの

ビットに書き込めるのは、PLLI2S が無効なときだけです。

SPDIF モードに応じて 44 または 48 kHz に近いオーディオクロックが得られるよう、分周比は SPDIF 内のプリスケアラ値に従って選択する必要があります。

$SPDIF \text{ クロック周波数} = VCO \text{ 周波数} / PLLI2SP$ 、ここで  $PLLI2SP = 2, 4, 6, \text{ または } 8$

00 : PLLI2SP = 2

01 : PLLI2SP = 4

10 : PLLI2SP = 6

11 : PLLI2SP = 8

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:6 **PLLI2SN[8:0]** : VCOのための PLLI2S 通倍数

これらのビットは、VCO の通倍数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。これらのビットの書き込みでは、ハーフワードおよびワード単位でのアクセスのみが可能です。

**注意：** これらのビットは、VCO 出力周波数が 49~432 MHz の間になるように、ソフトウェアで正しく設定する必要があります。

VCO 出力周波数 = VCO 入力周波数 \* PLLI2SN、ここで  $49 \leq \text{PLLI2SN} \leq 432$

000000000 : PLLI2SN = 0、誤った設定

000000001 : PLLI2SN = 1、誤った設定

...

000110000 : PLLI2SN = 48、誤った設定

000110001 : PLLI2SN = 49

...

011000000 : PLLI2SN = 192

011000001 : PLLI2SN = 193

...

110110000 : PLLI2SN = 432

110110000 : PLLI2SN = 433、誤った設定

...

111111111 : PLLI2SN = 511、誤った設定

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

## 5.3.24 RCC PLLSAI 設定レジスタ (RCC\_PLLSAICFGR)

アドレスオフセット : 0x88

リセット値 : 0x2400 3000

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

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

- $f(VCO \text{ クロック}) = f(\text{PLLSAI クロック入力}) * (\text{PLLSAIN} / \text{PLLM})$
- $f(\text{PLLISAI\_P}) = f(VCO \text{ クロック}) / \text{PLLSAIP}$
- $f(\text{PLLISAI\_Q}) = f(VCO \text{ クロック}) / \text{PLLSAIQ}$
- $f(\text{PLLISAI\_R}) = f(VCO \text{ クロック}) / \text{PLLSAIR}$

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **PLLSAIR[2:0]** : LCD クロックのための PLLSAI 分周比

LCD クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットは、PLLSAI が無効なときに書き込むことができます。

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

000 : PLLSAIR = 0、誤った設定

001 : PLLSAIR = 1、誤った設定

010 : PLLSAIR = 2

...

111 : PLLSAIR = 7

ビット 27:24 **PLLSAIQ[3:0]** : SAI クロックのための PLLSAI 分周比

SAI クロックの周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットは、PLLSAI が無効なときに書き込むことができます。

SAI1 クロック周波数 = VCO 周波数 / PLLSAIQ、ここで  $2 \leq \text{PLLSAIQ} \leq 15$

0000 : PLLSAIQ = 0、誤った設定

0001 : PLLSAIQ = 1、誤った設定

...

0010 : PLLSAIQ = 2

0011 : PLLSAIQ = 3

0100 : PLLSAIQ = 4

0101 : PLLSAIQ = 5

...

1111 : PLLSAIQ = 15

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **PLLSAIP[1:0]** : 48MHz クロックのための PLLSAI 分周比

PLLSAI 出力クロック (PLLSAI48CLK) の周波数を制御するために、ソフトウェアによってセット/クリアされます。この出力は、USB、RNG、SDMMC (48 MHz クロック)で選択できます。これらのビットに書き込めるのは、PLLSAI が無効なときだけです。

これらのビットの書き込みでは、ハーフワードおよびワード単位でのアクセスのみが可能です。

PLLSAI48 出力クロック周波数 = VCO 周波数 / PLLSAIP、ここで PLLSAI P = 2、4、6、または8

00 : PLLSAIP = 2

01 : PLLSAIP = 4

10 : PLLSAIP = 6

11 : PLLSAIP = 8

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:6 **PLLSAIN[8:0]** : VCO のための PLLSAI 分周比

VCO の逡倍数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットは、PLLSAI が無効なときに書き込むことができます。

これらのビットの書き込みでは、ハーフワードおよびワード単位でのアクセスのみが可能です。

VCO 出力周波数 = VCO 入力周波数 \* PLLSAIN、ここで  $49 \leq \text{PLLSAIN} \leq 432$

000000000 : PLLSAIN = 0、誤った設定

000000001 : PLLSAIN = 1、誤った設定

.....

000110000 : PLLSAIN = 48、誤った設定

000110001 : PLLSAIN = 49

...

011000000 : PLLSAIN = 192

011000001 : PLLSAIN = 193

...

110110000 : PLLSAIN = 432

110110000 : PLLSAIN = 433、誤った設定

...

111111111 : PLLSAIN = 511、誤った設定

ビット 5:0 予約済みであり、リセット値のままにしておかなければなりません。

## 5.3.25 RCC 専用クロック設定レジスタ (RCC\_DKCFGR1)

アドレスオフセット : 0x8C

リセット値 : 0x0000 0000

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

次の式に従い、このレジスタを使用して、タイマクロックプリスケラ、および SAI、LTDC ペリフェラルのための PLLSAI、および PLLI2S の 出力クロック分周比を設定できます。

$$f_{\text{(PLLSAIDIVQ クロック出力)}} = f_{\text{(PLLSAI\_Q)}} / \text{PLLSAIDIVQ}$$

$$f_{\text{(PLLSAIDIVR クロック出力)}} = f_{\text{(PLLSAI\_R)}} / \text{PLLSAIDIVR}$$

$$f_{\text{(PLLI2SDIVQ クロック出力)}} = f_{\text{(PLLI2S\_Q)}} / \text{PLLI2SDIVQ}$$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMPRE	SAI2SEL[1:0]			SAI1SEL[1:0]		Res.	Res.	PLLSAIDIVR[1:0]	
							rW	rW	rW	rW	rW			rW	rW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	Res.	PLLSAIDIVQ[4:0]					Res.	Res.	Res.	PLLI2SDIVQ[4:0]					
			rW	rW	rW	rW	rW				rW	rW	rW	rW	rW	

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TIMPRE** : タイマクロックプリスケラ選択

このビットは、APB1 ドメインおよび APB2 ドメインに接続されているすべてのタイマのクロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

0 : APB プリスケラ (RCC\_CFGR レジスタのPPRE1、PPRE2) が分周比 1 に設定されている場合、TIMxCLK = PCLKx となります。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 2 倍に設定されます :

$$\text{TIMxCLK} = 2 * \text{PCLKx}$$

APB プリスケラ (RCC\_CFGR レジスタのPPRE1、PPRE2) が分周比 1、2、4 のいずれかに設定されている場合、TIMxCLK = HCLK となります。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 4 倍に設定されます :

$$\text{TIMxCLK} = 4 * \text{PCLKx}$$

ビット 23:22 **SAI2SEL[1:0]** : SAI2 クロックソースの選択 :

これらのビットは、SAI2 クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットに書き込めるのは、PLLSAI および PLLI2S が無効なときです。

$$00 : \text{SAI2 クロック周波数} = f_{\text{(PLLSAI\_Q)}} / \text{PLLSAIDIVQ}$$

$$01 : \text{SAI2 クロック周波数} = f_{\text{(PLLI2S\_Q)}} / \text{PLLI2SDIVQ}$$

$$10 : \text{SAI2 クロック周波数} = \text{オルタネート機能入力周波数}$$

11 : 誤った設定

ビット 21:20 **SAI1SEL[1:0]** : SAI1 クロックソースの選択

これらのビットは、SAI1 クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットに書き込めるのは、PLLSAI および PLLI2S が無効なときです。

$$00 : \text{SAI1 クロック周波数} = f_{\text{(PLLSAI\_Q)}} / \text{PLLSAIDIVQ}$$

$$01 : \text{SAI1 クロック周波数} = f_{\text{(PLLI2S\_Q)}} / \text{PLLI2SDIVQ}$$

$$10 : \text{SAI1 クロック周波数} = \text{オルタネート機能入力周波数}$$

11 : 誤った設定

ビット 19 : 18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **PLLSAIDIVR[1:0]** : LCD\_CLK のための分周比

これらのビットは、LCD\_CLK の周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットに書き込めるのは、PLLSAI が無効なときだけです。

$\text{LCD\_CLK 周波数} = f(\text{PLLSAI\_R}) / \text{PLLSAIDIVR}$ 、ここで  $2 \leq \text{PLLSAIDIVR} \leq 16$

00 : PLLSAIDIVR = /2

01 : PLLSAIDIVR = /4

10 : PLLSAIDIVR = /8

11 : PLLSAIDIVR = /16

ビット 15 : 13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **PLLSAIDIVQ[4:0]** : SAI1 クロックのための PLLSAI 分周比

これらのビットは、SAI1 クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットに書き込めるのは、PLLSAI が無効なときだけです。

$\text{SAI1 クロック周波数} = f(\text{PLLSAI\_Q}) / \text{PLLSAIDIVQ}$ 、ここで  $1 \leq \text{PLLSAIDIVQ} \leq 31$

00000 : PLLSAIDIVQ = /1

00001 : PLLSAIDIVQ = /2

00010 : PLLSAIDIVQ = /3

00011 : PLLSAIDIVQ = /4

00100 : PLLSAIDIVQ = /5

...

11111 : PLLSAIDIVQ = /32

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **PLLI2SDIV[4:0]** : SAI1 クロックのための PLLI2S 分周比

これらのビットは、SAI1 クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。

これらのビットに書き込めるのは、PLLI2S が無効なときだけです。

$\text{SAI1 クロック周波数} = f(\text{PLLI2S\_Q}) / \text{PLLI2SDIVQ}$ 、ここで  $1 \leq \text{PLLI2SDIVQ} \leq 32$

00000 : PLLI2SDIVQ = /1

00001 : PLLI2SDIVQ = /2

00010 : PLLI2SDIVQ = /3

00011 : PLLI2SDIVQ = /4

00100 : PLLI2SDIVQ = /5

...

11111 : PLLI2SDIVQ = /32

## 5.3.26 RCC 専用クロック設定レジスタ (DCKCFGR2)

アドレス : 0x90h

リセット値 : 0x0000 0000h

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

このレジスタを使用して、48MHz、SDMMC、HDMI-CEC、LPTIM1、UART、USART、および I2C クロックのソースクロックを選択できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SDMMC SEL	CK48MS EL	CECSEL	LPTIM1SEL		I2C4SEL		I2C3SEL		I2C2SEL		I2C1SEL	
			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART8SEL		UART7SEL		USART6SEL		UART5SEL		UART4SEL		UART3SEL		UART2SEL		UART1SEL	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **SDMMCSSEL** : SDMMC クロックソース選択

ソフトウェアでセット/クリアされます。

0 : 48 MHz クロックが SDMMC クロックとして選択されます。

1 : システムクロックが SDMMC クロックとして選択されます。

ビット 27 **CK48MSSEL** : 48MHz クロックソース選択

ソフトウェアでセット/クリアされます。

0 : 48MHz クロックが PLL から選択されます。

1 : 48MHz クロックが PLLSAI から選択されます。

ビット 26 **CECSEL** : HDMI-CEC クロックソース選択

ソフトウェアでセット/クリアされます。

0 : LSE クロックが HDMI-CEC クロックとして選択されます。

1 : 488 分周された HSI クロックが HDMI-CEC クロックとして選択されます。

ビット 25:24 **LPTIM1SEL** : 低電力タイマ 1 クロックソース選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が LPTILM1 クロックとして選択されます。

01 : LSI クロックが LPTILM1 クロックとして選択されます。

10 : HSI クロックが LPTILM1 クロックとして選択されます。

11 : LSE クロックが LPTILM1 クロックとして選択されます。

ビット 23:22 **I2C4SEL** : I2C4 クロックソースの選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が I2C4 クロックとして選択されます。

01 : システムクロックが I2C4 クロックとして選択されます。

10 : HSI クロックが I2C4 クロックとして選択されます。

11 : 予約済み

ビット 21:20 **I2C3SEL** : I2C3 クロックソースの選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が I2C3 クロックとして選択されます。

01 : システムクロックが I2C3 クロックとして選択されます。

10 : HSI クロックが I2C3 クロックとして選択されます。

11 : 予約済み

**ビット 19:18 I2C2SEL** : I2C2 クロックソースの選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が I2C2 クロックとして選択されます。

01 : システムクロックが I2C2 クロックとして選択されます。

10 : HSI クロックが I2C2 クロックとして選択されます。

11 : 予約済み

**ビット 17:16 I2C1SEL** : I2C1 クロックソースの選択

ソフトウェアでセット/クリアされます。

00 : APB クロック (PCLK1) が I2C1 クロックとして選択されます。

01 : システムクロックが I2C1 クロックとして選択されます。

10 : HSI クロックが I2C1 クロックとして選択されます。

11 : 予約済み

**ビット 15:14 UART8SEL[1:0]** : UART 8 クロックソース選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が UART 8 クロックとして選択されます。

01 : システムクロックが UART 8 クロックとして選択されます。

10 : HSI クロックが UART 8 クロックとして選択されます。

11 : LSE クロックが UART 8 クロックとして選択されます。

**ビット 13:12 UART7SEL[1:0]** : UART 7 クロックソース選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が UART 7 クロックとして選択されます。

01 : システムクロックが UART 7 クロックとして選択されます。

10 : HSI クロックが UART 7 クロックとして選択されます。

11 : LSE クロックが UART 7 クロックとして選択されます。

**ビット 11:10 USART6SEL[1:0]** : USART 6 クロックソース選択

ソフトウェアでセット/クリアされます。

00 : APB2 クロック (PCLK2) が USART 6 クロックとして選択されます。

01 : システムクロックが USART 6 クロックとして選択されます。

10 : HSI クロックが USART 6 クロックとして選択されます。

11 : LSE クロックが USART 6 クロックとして選択されます。

**ビット 9:8 UART5SEL[1:0]** : UART 5 クロックソース選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が UART 5 クロックとして選択されます。

01 : システムクロックが UART 5 クロックとして選択されます。

10 : HSI クロックが UART 5 クロックとして選択されます。

11 : LSE クロックが UART 5 クロックとして選択されます。

**ビット 7:6 UART4SEL[1:0]** : UART 4 クロックソース選択

ソフトウェアでセット/クリアされます。

00 : APB1 クロック (PCLK1) が UART 4 クロックとして選択されます。

01 : システムクロックが UART 4 クロックとして選択されます。

10 : HSI クロックが UART 4 クロックとして選択されます。

11 : LSE クロックが UART 4 クロックとして選択されます。

ビット 5:4 **USART3SEL[1:0]** : USART 3 クロックソース選択

- ソフトウェアでセット/クリアされます。
- 00 : APB1 クロック (PCLK1) が USART 3 クロックとして選択されます。
- 01 : システムクロックが USART 3 クロックとして選択されます。
- 10 : HSI クロックが USART 3 クロックとして選択されます。
- 11 : LSE クロックが USART 3 クロックとして選択されます。

ビット 3:2 **USART2SEL[1:0]** : USART 2 クロックソース選択

- ソフトウェアでセット/クリアされます。
- 00 : APB1 クロック (PCLK1) が USART 2 クロックとして選択されます。
- 01 : システムクロックが USART 2 クロックとして選択されます。
- 10 : HSI クロックが USART 2 クロックとして選択されます。
- 11 : LSE クロックが USART 2 クロックとして選択されます。

ビット 1:0 **USART1SEL[1:0]** : USART 1 クロックソース選択

- ソフトウェアでセット/クリアされます。
- 00 : APB2 クロック (PCLK2) が USART 1 クロックとして選択されます。
- 01 : システムクロックが USART 1 クロックとして選択されます。
- 10 : HSI クロックが USART 1 クロックとして選択されます。
- 11 : LSE クロックが USART 1 クロックとして選択されます。

### 5.3.27 RCC レジスタマップ

表 20 に、レジスタマップとリセット値を示します。

表 20. RCC レジスタマップとリセット値

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	RCC_CR	Res.	Res.	Res.	PLL SAIRDY	PLL SAION	PLL I2SRDY	PLL I2SON	PLL RDY	PLL ON	Res.	Res.	Res.	Res.	CSSON	HSEBYP	HSERDY	HSEON	HSICAL 7	HSICAL 6	HSICAL 5	HSICAL 4	HSICAL 3	HSICAL 2	HSICAL 1	HSICAL 0	HSITRIM 4	HSITRIM 3	HSITRIM 2	HSITRIM 1	HSITRIM 0	予約済み	HSIRDY	HSION	
0x04	RCC_PLLCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x08	RCC_CFGR	MCO2 1	MCO2 0	MCO2PREZ	MCO2PRE1	MCO2PRE0	MCO1PRE2	MCO1PRE1	MCO1PRE0	I2SSRC	MCO1 1	MCO1 0	RTCPRE 4	RTCPRE 3	RTCPRE 2	RTCPRE 1	RTCPRE 0	PPRE2 2	PPRE2 1	PPRE2 0	PPRE1 2	PPRE1 1	PPRE1 0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0C	RCC_CIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	PLLSAIRDYC	PLLI2SRDYC	PLLRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x10	RCC_AHB1RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x14	RCC_AHB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x18	RCC_AHB3RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x1C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x20	RCC_APB1RSTR	UART8RST	UART7RST	DACRST	PWRRST	CECRST	CAN2RST	CAN1RST	I2C4RST	I2C3RST	I2C2RST	I2C1RST	UART5RST	UART4RST	UART3RST	UART2RST	SPDIFRXRST	SP13RST	SP12RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							

表 20. RCC レジスタマップとリセット値 (続き)

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x24	RCC_APB2RSTR	Res	Res	Res	Res	Res	LTCRST	Res	Res	Res	SAI2RST	SAI1RST	SPI6RST	SPI5RST	Res	TIM11RST	TIM10RST	TIM9RST	Res	SYSCFGGRST	SP45RST	SPI1RST	SDMMC1RST	Res	Res	ADCRST	Res	Res	USART6RST	USART11RST	Res	Res	TIM8RST	TIM1RST		
0x28	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		
0x2C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		
0x30	RCC_AHB1ENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMA2DEN	DMA2EN	DMA1EN	DTCMRAMEN	Res	BKPSRAMEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x34	RCC_AHB2ENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x38	RCC_AHB3ENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x3C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x40	RCC_APB1ENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	SPDIFRXEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x44	RCC_APB2ENR	Res	Res	Res	Res	Res	Res	Res	Res	Res	SAI2EN	SAI1EN	SPI6EN	SPI5EN	Res	TIM11EN	TIM10EN	TIM9EN	Res	SYSCFGEN	SPI4EN	SPI1EN	SDMMC1EN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x48	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x4C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x50	RCC_AHB1LPER	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMA2DLPEN	DMA2LPEN	DMA1LPEN	DTCMLPEN	Res	BKPSRAMLPEN	SRAM2LPEN	SRAM1LPEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x54	RCC_AHB2LPER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x58	RCC_AHB3LPER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x5C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x60	RCC_APB1LPER	Res	Res	Res	Res	Res	Res	Res	Res	Res	I2C3LPEN	I2C2LPEN	I2C1LPEN	UART5LPEN	UART4LPEN	USART3LPEN	USART2LPEN	SPDIFRXLPEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x64	RCC_APB2LPER	Res	Res	Res	Res	Res	Res	Res	Res	Res	SAI2LPEN	SAI1LPEN	SPI6LPEN	SPI5LPEN	Res	TIM11LPEN	TIM10LPEN	TIM9LPEN	Res	SYSCFGLPEN	SPI4LPEN	SPI1LPEN	SDMMC1LPEN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x68	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x6C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x70	RCC_BDCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res



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

表 20. RCC レジスタマップとリセット値 (続き)

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x74	RCC_CSR	LPWRRSTF	MWDGRSTF	WDGRSTF	SFTRSTF	PORRSTF	PADRSTF	BORRSTF	RMVF	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LSIRDY	LSION							
0x78	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x7C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x80	RCC_SSCGR	SSCGEN	SPREADSEL	Res	Res	INCSTEP										MODPER																		
0x84	RCC_PLLI2SCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x88	RCC_PLLSAICFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x8C	RCC_DCKCFGR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x90	RCC_DCKCFGR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 6 汎用 I/O (GPIO)

### 6.1 概要

各汎用 I/O ポートにはそれぞれ 4 つの 32 ビット設定レジスタ (GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR)、2 つの 32 ビットデータレジスタ (GPIOx\_IDR、GPIOx\_ODR)、および 1 つの 32 ビットセット/クリアレジスタ (GPIOx\_BSRR) があります。さらに、すべての GPIO には、それぞれ 32 ビットロックレジスタ (GPIOx\_LCKR) と 2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx\_AFRH、GPIOx\_AFLR) があります。

### 6.2 GPIO の主な機能

- 出力状態：プッシュプルまたはオープンドレイン + プルアップ/プルダウン
- 出力データレジスタ (GPIOx\_ODR) またはペリフェラル (オルタネート機能出力) からの出力データ
- 各 I/O のスピード選択
- 入力状態：フローティング、プルアップ/プルダウン、アナログ
- 入力データレジスタ (GPIOx\_IDR) またはペリフェラル (オルタネート機能入力) への入力データ
- GPIOx\_ODR へのビット単位の書き込みアクセス用のビットセット/クリアレジスタ (GPIOx\_BSRR)
- I/O ポート設定を固定するロック機構 (GPIOx\_LCKR)
- アナログ機能
- オルタネート機能選択レジスタ
- 2 クロックサイクルごとに変化可能な高速トグル
- 柔軟性の高いピンの多重化により、I/O ピンを GPIO またはいくつかのペリフェラル機能の 1 つとして使用可能

### 6.3 GPIO の機能説明

各 I/O ポートの特定のハードウェア特性については、データシートに記載されています。汎用 I/O (GPIO) ポートの各ポートビットは、ソフトウェアによって以下の動作モードを個別に設定できます。

- 入力フローティング
- 入力プルアップ
- 入力プルダウン
- アナログ
- プルアップまたはプルダウン機能を持つ出力オープンドレイン
- プルアップまたはプルダウン機能を持つ出力プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能オープンドレイン

各 I/O ポートビットは自由にプログラム可能ですが、I/O ポートレジスタには 32 ビットワード、ハーフワード、またはバイト単位でアクセスする必要があります。GPIOx\_BSRR および GPIOx\_BRR レジスタを使用すると、任意の GPIOx\_ODR レジスタに不可分な読み出し/変更アクセスを行うことができます。これにより、読み出しと変更アクセスの間に IRQ が発生するリスクを回避できます。

図 16 および図 17 は、標準 I/O ポートビットと 5V トレラント I/O ポートビットの基本構成をそれぞれ示します。表 21 には、実行可能なポートビットの設定を示します。

図 16. I/O ポートビットの基本構成

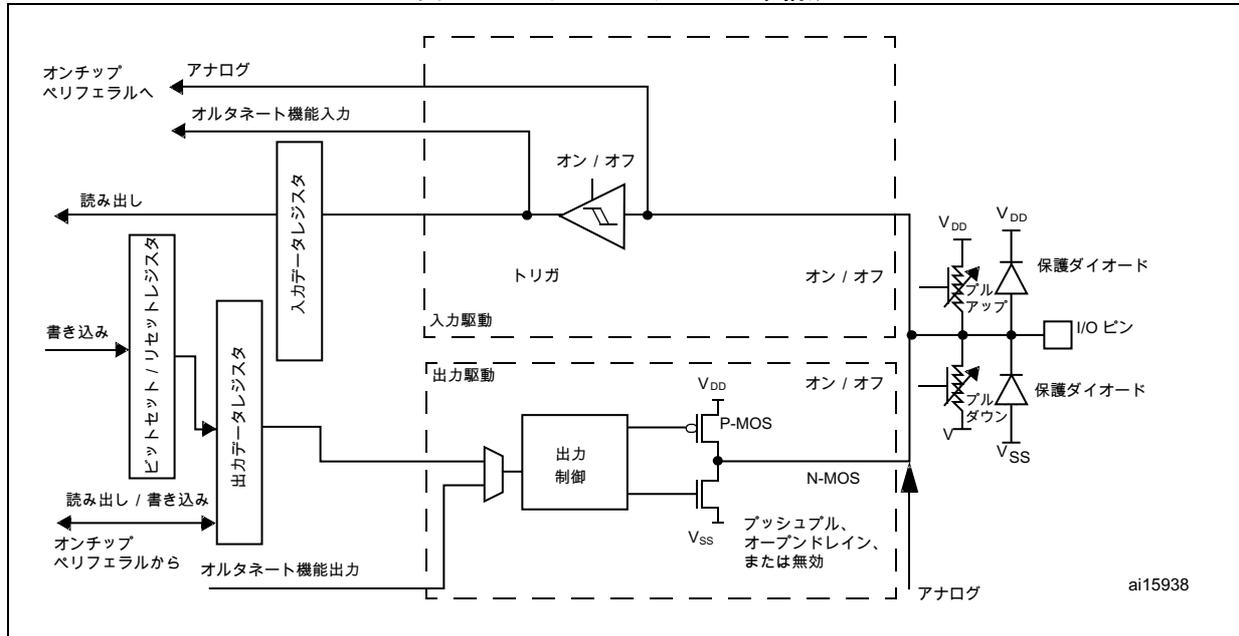
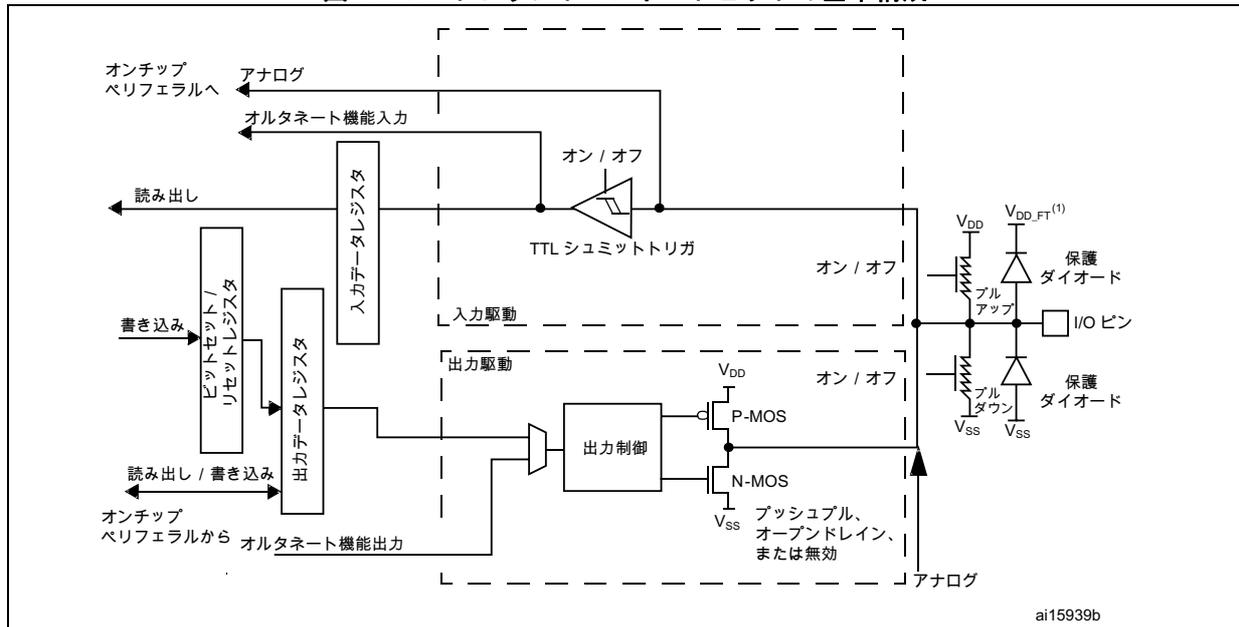


図 17. 5V トレラント I/O ポートビットの基本構成



1.  $V_{DD\_FT}$  は、5V トレラント I/O に固有の電位であり、 $V_{DD}$  とは異なります。

表 21. ポートビット設定表<sup>(1)</sup>

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 設定	
01	0	SPEED [1:0]		0	0	GP 出力	PP
	0			0	1	GP 出力	PP + PU
	0			1	0	GP 出力	PP + PD
	0			1	1	予約済み	
	1			0	0	GP 出力	OD
	1			0	1	GP 出力	OD + PU
	1			0	0	GP 出力	OD + PD
	1			1	予約済み (GP 出力 OD)		
10	0	SPEED [1:0]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	予約済み	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			0	0	AF	OD + PD
	1			1	予約済み		
00	x	x	x	0	0	入力	フローティング
	x	x	x	0	1	入力	PU
	x	x	x	1	0	入力	PD
	x	x	x	1	1	予約済み (入力フローティング)	
11	x	x	x	0	0	入力/出力	アナログ
	x	x	x	0	1	予約済み	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 汎用、PP = プッシュプル、PU = プルアップ、PD = プルダウン、OD = オープンドレイン、AF = オルタネート機能

### 6.3.1 汎用 I/O (GPIO)

リセット中とリセット直後は、オルタネート機能は有効ではなく、ほとんどの I/O ポートは入力フローティングモードに設定されています。

リセット後、デバッグピンはオルタネート機能のプルアップ/プルダウンに設定されています。

- PA15 : JTDI プルアップ
- PA14 : JTCK/SWCLK プルダウン
- PA13 : JTMS/SWDAT プルアップ
- PB4 : NJTRST プルアップ
- PB3 : JTDO フローティング状態

ピンが出力として設定されている場合、出力データレジスタ (GPIOx\_ODR) に書き込まれた値が I/O ピンに出力されます。出力ドライバをプッシュプルモードまたはオープンドレインモード (ローレベルのみが駆動され、ハイレベルはハイインピーダンス) で使用することができます。

入力データレジスタ (GPIOx\_IDR) は、AHB クロックサイクルごとに、I/O ピン上のデータをキャプチャします。

すべての GPIO ピンに、内部ウィークプルアップ抵抗とウィークプルダウン抵抗があり、GPIOx\_PUPDR レジスタの値によってこれらを有効化/無効化できます。

## 6.3.2 I/O ピンオルタネート機能マルチプレクサと配置

デバイスの I/O ピンは、マルチプレクサを介してオンボードのペリフェラル/モジュールに接続され、一度に 1 つのペリフェラルオルタネート機能 (AF) のみが 1 つの I/O ピンに接続可能となっています。この方法により、同じ I/O ピンを共有するペリフェラル間での競合を無くすることができます。

各 I/O ピンは、最大 16 のオルタネート機能入力 (AF0~AF15) を持つマルチプレクサを内蔵しており、GPIOx\_AFRL (ピン 0~7) と GPIOx\_AFRH (ピン 8~15) レジスタを介して設定することができます。

- リセット後、マルチプレクサの選択はオルタネート機能 0 (AF0) です。I/O は、GPIOx\_MODER レジスタを通してオルタネート機能モードで設定されます。
- 各ピンに固有のオルタネート機能割り当てについての詳細は、デバイスデータシートに記載されています。
- Cortex-M7 with FPU EVENTOUT は AF15 に配置されています。

この柔軟性の高い I/O 多重化アーキテクチャに加え、各ペリフェラルではオルタネート機能がそれぞれの I/O ピンに配置されており、さらに小型のパッケージで使用できるペリフェラルの数を最適化します。

I/O を任意の設定で使用するには、次の手順に従います。

- **デバッグ機能**：各デバイスのリセット後、これらのピンはデバッガホストによってすぐに使用可能なオルタネート機能ピンとして割り当てられます。
- **システム機能**：MCOx ピンはオルタネート機能モードに設定する必要があります。
- **GPIO**：必要とする I/O を、GPIOx\_MODER レジスタで出力、入力、またはアナログとして設定します。
- **ペリフェラルオルタネート機能**
  - I/O を GPIOx\_AFRL または GPIOx\_AFRH レジスタのどちらかで必要とする AFx に接続します。
  - タイプ、プルアップ/プルダウン、出力スピードをそれぞれ GPIOx\_OTYPER、GPIOx\_PUPDR、GPIOx\_OSPEEDER レジスタで選択します。
  - 必要とする I/O を、GPIOx\_MODER レジスタでオルタネート機能として設定します。
- **追加機能**：
  - ADC および DAC については、GPIOx\_MODER レジスタのアナログモードで必要とする I/O を設定し、ADC および DAC レジスタで必要な機能を設定します。
  - RTC\_OUT、RTC\_TS、RTC\_TAMPx、WKUPx、オシレータなどの追加機能については、関連する RTC、PWR、および RCC レジスタで必要な機能を設定します。これらの機能は、標準の GPIO レジスタの設定よりも優先されます。RTC による I/O 制御の詳細は、[セクション 29.3 : RTC の機能説明 \(875 ページ\)](#) を参照してください。

- EVENTOUT
  - AF15 に接続することで、I/O ピンを Cortex<sup>®</sup>-M7 with FPU EVENTOUT 信号を出力する用途に設定します。

オルタネート機能 I/O ピンの配置に関する詳細は、デバイスデータシートの「オルタネート機能配置」表を参照してください。

### 6.3.3 I/O ポート制御レジスタ

各 GPIO ポートには 32 ビットメモリマップド制御レジスタが 4 つ (GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR) あり、最大 16 個の I/O を設定します。GPIOx\_MODER レジスタは I/O モード (入力、出力、AF、アナログ) を選択するために使用されます。GPIOx\_OTYPER および GPIOx\_OSPEEDR レジスタは、出力タイプ (プッシュプルまたはオープンドレイン) および速度を選択するために使用されます。I/O の方向がどちらであっても、GPIOx\_PUPDR レジスタは、プルアップ/プルダウンを選択するために使用されます。

### 6.3.4 I/O ポートデータレジスタ

各 GPIO には、16 ビットメモリマップドデータレジスタが 2 つあります。入力データレジスタ (GPIOx\_IDR) と出力データレジスタ (GPIOx\_ODR) です。GPIOx\_ODR は出力されるデータを格納し、読み出し/書き込みアクセスが可能です。I/O から入力されるデータは読み出し専用の入力データレジスタ (GPIOx\_IDR) に格納されます。

レジスタの説明は、[セクション 6.4.5 : GPIO ポート入力データレジスタ \(GPIOx\\_IDR\) \(x = A~K\)](#) および [セクション 6.4.6 : GPIO ポート出力データレジスタ \(GPIOx\\_ODR\) \(x = A~K\)](#) を参照してください。

### 6.3.5 I/O データのビット単位の操作

ビットセット/クリアレジスタ (GPIOx\_BSRR) は 32 ビットレジスタで、アプリケーションによる出力データレジスタ (GPIOx\_ODR) のビット単位のセット/クリアを可能にします。ビットセットリセットレジスタは GPIOx\_ODR の 2 倍のサイズです。

GPIOx\_ODR の各ビットには GPIOx\_BSRR の 2 つの制御ビット BS(i) と BR(i) が対応します。ビット BS(i) に 1 を書き込むと、対応する ODR(i) ビットが**セット**されます。ビット BR(i) に 1 を書き込むと、対応する ODR(i) ビットが**リセット**されます。

GPIOx\_BSRR のいかなるビットに 0 を書き込んでも GPIOx\_ODR の対応するビットには影響しません。仮に、GPIOx\_BSRR のビットに対してセットおよびリセットの両方を実行しようとした場合、セット動作が優先されます。

GPIOx\_BSRR レジスタを使用した GPIOx\_ODR 内の個々のビットの変更には、1 回限りの効果しかなく、GPIOx\_ODR ビットを固定するものではありません。GPIOx\_ODR のビットは常に直接アクセスすることができます。GPIOx\_BSRR レジスタによって、ビット単位の不可分操作を行うことができます。

GPIOx\_ODR をビットレベルでプログラムする場合は、ソフトウェアで割り込みを無効にする必要はありません。1 回の不可分な AHB 書き込みアクセスで 1 ビットまたは複数ビットを変更することができます。

## 6.3.6 GPIO ロック機構

GPIOx\_LCKR レジスタへ特定の書き込みシーケンスを行うことにより、GPIO 制御レジスタをロックすることができます。ロックされるレジスタは、GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR、GPIOx\_AFRL、GPIOx\_AFRH です。

GPIOx\_LCKR レジスタに書き込むには、特定の書き込み／読み出しシーケンスを行う必要があります。このレジスタのビット 16 に適切な LOCK シーケンスを行う場合、LCKR[15:0] の値を使用して I/O の設定を固定します（この書き込みシーケンス中、LCKR[15:0] の値は同じである必要があります）。あるポートビットに LOCK シーケンスが行われると、次の MCU リセットまたはペリフェラルリセットまで、そのポートビットの値を変更できなくなります。GPIOx\_LCKR の各ビットによって、制御レジスタ (GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR、GPIOx\_AFRL、GPIOx\_AFRH) の対応するビットが停止されます。

GPIOx\_LCKR ビット 16 を [15:0] ビットと同時に設定する必要があるため、この LOCK シーケンス（[セクション 6.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx\\_LCKR\) \(x = A ~ K\)](#) を参照）は、GPIOx\_LCKR レジスタへのワード（32 ビット長）アクセスを使用してのみ実行できます。

詳細については、[セクション 6.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx\\_LCKR\) \(x = A ~ K\)](#) の LCKR レジスタの説明を参照してください。

## 6.3.7 I/O オルタネート機能の入力／出力

各 I/O が使用できるオルタネート機能入力／出力の 1 つを選択するため、2 つのレジスタが用意されています。これらのレジスタを使用し、必要に応じて、アプリケーションでオルタネート機能を他のピンに接続することができます。

つまり、GPIOx\_AFRL および GPIOx\_AFRH オルタネート機能レジスタを使用していくつかの使用可能なペリフェラル機能が、各 GPIO に多重化されることとなります。こうして各 I/O に使用可能な機能のどれか 1 つをアプリケーションによって選択できます。AF 選択信号はオルタネート機能入力およびオルタネート機能出力に共通なので、任意の I/O が持つオルタネート機能入力／出力に対し 1 つのチャンネルが選択されます。

どの機能が各 GPIO ピンに多重化されているかについてはデバイスデータシートを参照してください。

## 6.3.8 外部割り込み／ウェイクアップライン

すべてのポートに外部割り込み機能があります。外部割り込みラインを使用するには、ポートを入力モードに設定する必要があります。[セクション 11: 拡張割り込み／イベントコントローラ \(EXTI\)](#) および [セクション 11.3: ウェイクアップイベント管理](#) を参照してください。

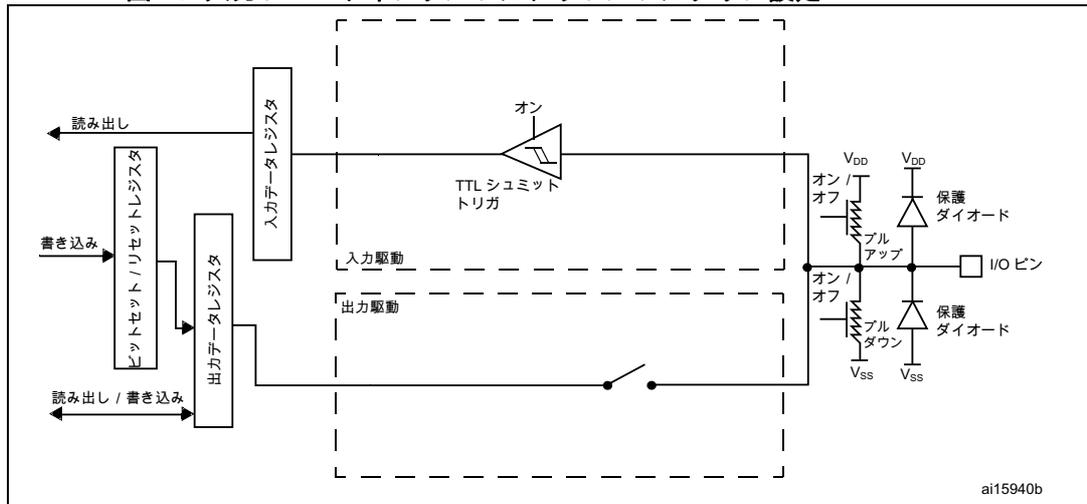
## 6.3.9 入力設定

I/O ポートが入力としてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力が有効になります。
- GPIOx\_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

 [18](#) は、I/O ポートビットの入力設定を示しています。

図 18. 入力フローティング/プルアップ/プルダウン設定



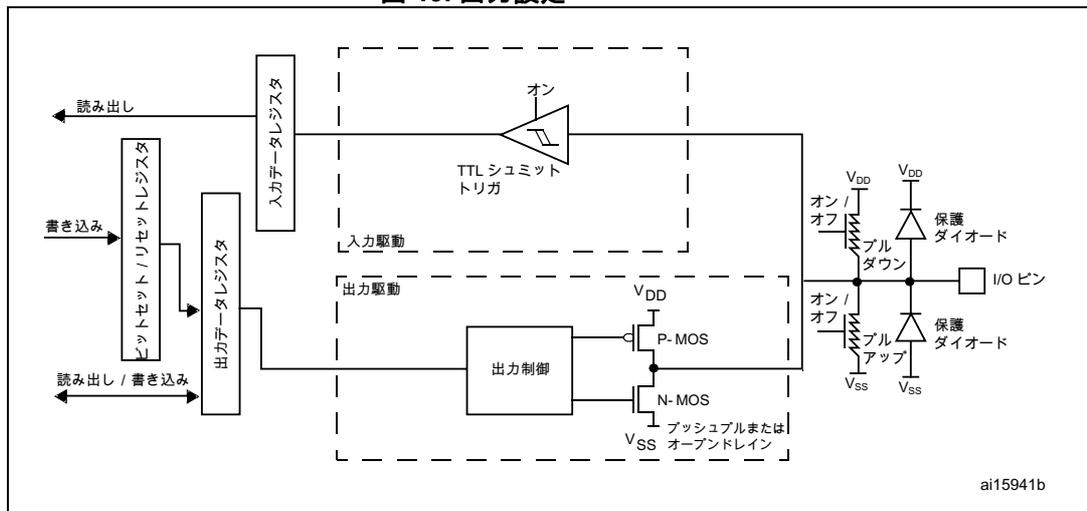
## 6.3.10 出力設定

I/O ポートが出力としてプログラムされた場合、

- 出力バッファが有効になります。
  - オープンドレインモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときはポートはハイインピーダンス状態のままです (P-MOS が有効になることはありません)。
  - プッシュプルモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときは P-MOS が有効になります。
- シュミットトリガ入力があります。
- GPIOx\_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。
- 出力データレジスタの読み出しアクセスによって、最後に書き込まれたデータが得られます。

図 19 は、I/O ポートビットの出力設定を示しています。

図 19. 出力設定



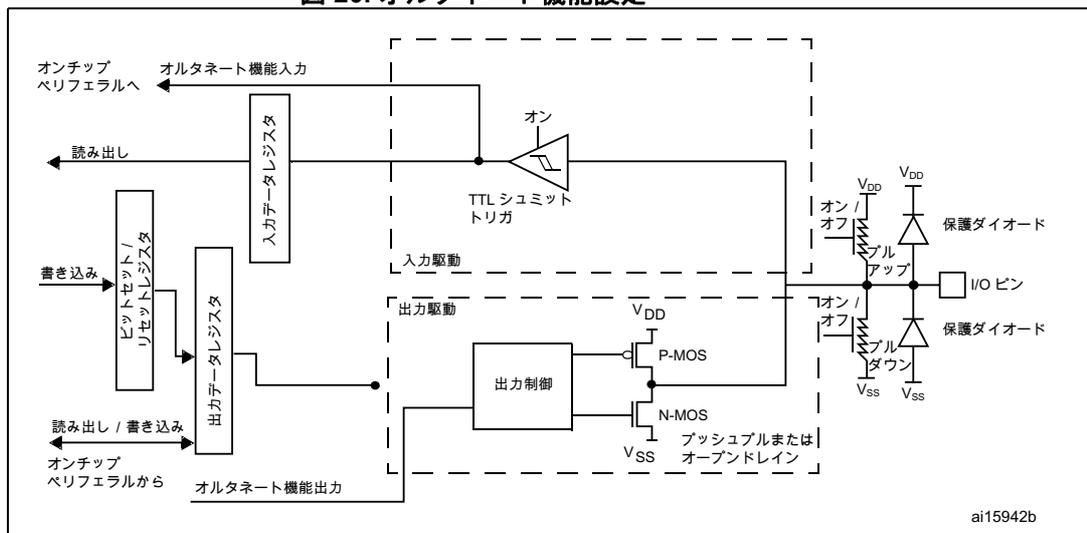
### 6.3.11 オルタネート機能設定

I/O ポートがオルタネート機能としてプログラムされた場合、

- 出力バッファをオープンドレインまたはプッシュプルモードに設定することができます。
- 出力バッファが、ペリフェラル (トランスミッタイネーブルおよびデータ) からの信号で駆動されます。
- シュミットトリガ入力が有効になります。
- ウィークプルアップ抵抗およびプルダウン抵抗が有効になるか否かは、GPIOx\_PUPDR レジスタの値によって決まります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

図 20 は、I/O ポートビットのオルタネート機能設定を示しています。

図 20. オルタネート機能設定



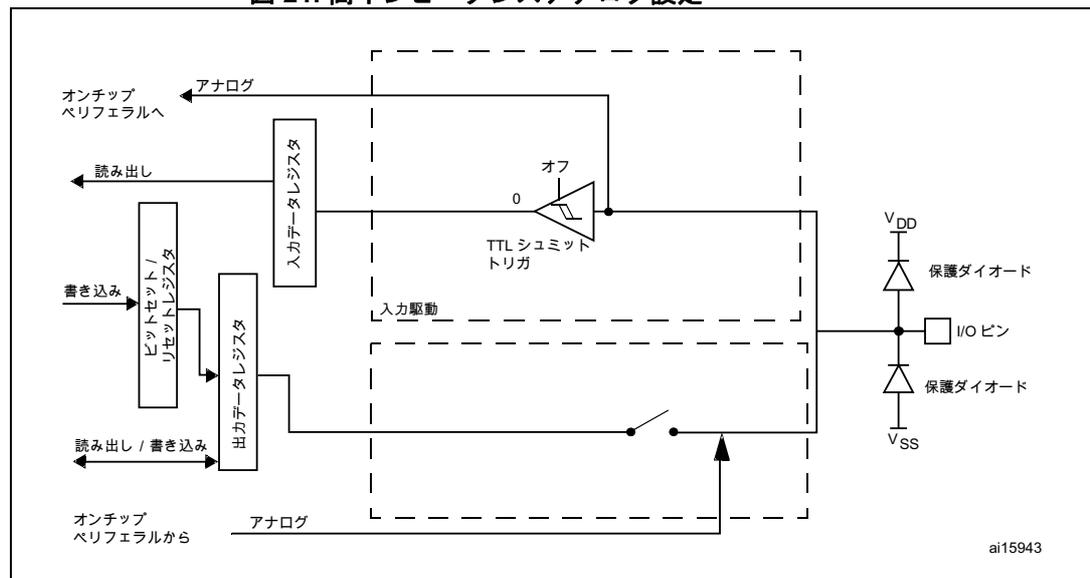
## 6.3.12 アナログ設定

I/O ポートがアナログとしてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力は無効になり、I/O ピンのどのようなアナログ値に対しても消費電力をゼロに抑えます。シュミットトリガ回路の出力は、常に“0”になります。
- ウィークプルアップ抵抗およびプルダウン抵抗はハードウェアによって無効にされます。
- 入力データレジスタの読み出しアクセスを行うと、値“0”が得られます。

図 21 は、I/O ポートビットの高インピーダンスアナログ入力設定を示しています。

図 21. 高インピーダンスアナログ設定



## 6.3.13 HSEまたはLSE オシレータのピンを GPIO として使用

HSE または LSE オシレータがスイッチオフされた場合（リセット後のデフォルト状態）、関連のオシレータピンを通常の GPIO として使用することができます。

HSE または LSE オシレータがスイッチオンされた場合（RCC\_CSR レジスタの HSEON または LSEON ビットを設定することで）、オシレータは関連ピンを制御しますが、これらのピンの GPIO 設定は無効です。

オシレータがユーザ外部クロックモードに設定されている場合、OSC\_IN または OSC32\_IN ピンのみがクロック入力のために確保されますが、OSC\_OUT または OSC32\_OUT ピンは通常の GPIO として使用することができます。

## 6.3.14 GPIO ピンをバックアップ供給ドメインで使用

コア供給ドメインの電源がオフになったとき（デバイスが STANDBY モードに移行したとき）、PC13/PC14/PC15/PI8 の GPIO が機能しなくなります。この場合、それらの GPIO 設定が RTC 設定によってバイパスされなければ、これらのピンはアナログ入力モードに設定されます。

セクション 29.3 : RTC の機能説明 (875 ページ)。

## 6.4 GPIO レジスタ

このセクションには、GPIO レジスタの詳細な説明が記載されています。

レジスタビット、レジスタアドレスオフセット、リセット値の概要は、表 22 を参照してください。

ペリフェラルレジスタはワード、ハーフワード、バイトのいずれかのモードで書き込むことができます。

### 6.4.1 GPIO ポートモードレジスタ (GPIOx\_MODER) (x = A~K)

アドレスオフセット : 0x00

リセット値 :

- ポート A 0xA800 0000
- ポート B 0x0000 0280
- 他のポート 0x0000 0000

31		30		29		28		27		26		25		24		23		22		21		20		19		18		17		16	
MODER15[1:0]				MODER14[1:0]				MODER13[1:0]				MODER12[1:0]				MODER11[1:0]				MODER10[1:0]				MODER9[1:0]				MODER8[1:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW																					
15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
MODER7[1:0]				MODER6[1:0]				MODER5[1:0]				MODER4[1:0]				MODER3[1:0]				MODER2[1:0]				MODER1[1:0]				MODER0[1:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW																					

ビット  $2y+1:2y$  MODERy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O モードを設定するために、ソフトウェアによって書き込まれます。

- 00 : 入力モード (リセット状態)
- 01 : 汎用出力モード
- 10 : オルタネート機能モード
- 11 : アナログモード

## 6.4.2 GPIO ポート出力タイプレジスタ (GPIOx\_OTYPERR) (x = A~K)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw															

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **OTy** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O 出力タイプを設定するために、ソフトウェアによって書き込まれます。

0 : 出力プッシュプル (リセット状態)

1 : 出力オープンドレイン

## 6.4.3 GPIO ポート出力スピードレジスタ (GPIOx\_OSPEEDR) (x = A~K)

アドレスオフセット : 0x08

リセット値 :

- ポート A 0x0C00 0000
- ポート B 0x0000 00C0
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3 [1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw										

ビット 2y+1:2y **OSPEEDRy[1:0]** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O の出力スピードを設定するために、ソフトウェアによって書き込まれます。

00 : ロースピード

01 : ミディアムスピード

10 : 準高速

11 : ハイスピード

**注 :** OSPEEDRy ビット値対 V<sub>DD</sub> 範囲および外部負荷については製品データシートを参照してください。

## 6.4.4 GPIO ポートプルアップ／プルダウンレジスタ (GPIOx\_PUPDR) (x = A~K)

アドレスオフセット : 0x0C

リセット値 :

- ポート A 0x6400 0000
- ポート B 0x0000 0100
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w										

ビット 2y+1:2y PUPDRy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O のプルアップまたはプルダウンを設定するために、ソフトウェアによって書き込まれます。

- 00 : プルアップ／プルダウンなし
- 01 : プルアップ
- 10 : プルダウン
- 11 : 予約済み

## 6.4.5 GPIO ポート入力データレジスタ (GPIOx\_IDR) (x = A~K)

アドレスオフセット : 0x10

リセット値 : 0x0000 XXXX (X は未定義)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 IDRy : ポート入力データビット (y = 0~15)

これらのビットは読み出し専用です。これらのビットには、対応する I/O ポートの入力値が格納されています。

## 6.4.6 GPIO ポート入力データレジスタ (GPIOx\_ODR) (x = A~K)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ODRy** : ポート出入力データビット (y = 0~15)

これらのビットは、ソフトウェアによって読み出し/書き込みができます。

**注 :** 不可分なビット単位のセット/クリアのために、GPIOx\_BSRR または GPIOx\_BRR レジスタ (x = A~F) への書き込みによって、ODR ビットを個々にセット/クリアすることができます。

## 6.4.7 GPIO ポートビットセット/クリアレジスタ (GPIOx\_BSRR) (x = A~K)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 **BRy** : ポート x リセットビット y (y = 0~15)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODRx ビットの値は変化しません。

1 : 対応する ODRx ビットをリセットします。

**注 :** BSx ビットと BRx ビットの両方がセットされた場合、BSx ビットが優先されます。

ビット 15:0 **BSy** : ポート x セットビット y (y = 0~15)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODRx ビットの値は変化しません。

1 : 対応する ODRx ビットをセットします。



## 6.4.8 GPIO ポート設定ロックレジスタ (GPIOx\_LCKR) (x = A~K)

このレジスタは、ビット 16 (LCKK) に正しい書き込みシーケンスが行われたときに、ポートビットの設定をロックするために使用されます。ビット [15:0] の値は、GPIO の設定をロックするために使用されます。書き込みシーケンスの間は、LCKR[15:0] の値を変更することはできません。あるポートビットに LOCK シーケンスが適用されると、次の MCU リセットまたはペリフェラルリセットまで、このポートビットの値を変更できなくなります。

**注：** GPIOx\_LCKR レジスタへの書き込みには特定の書き込みシーケンスが使われます。このロックシーケンスではワードアクセス (32 ビット長) のみ可能です。

各ロックビットによって、特定の設定レジスタ (制御レジスタおよびオルタネート機能レジスタ) が固定されます。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 LCKK : ロックキー

このビットは常時読み出しができます。ロックキー書き込みシーケンスを使用しなければ変更できません。

0 : ポート設定ロックキーは無効です。

1 : ポート設定ロックキーは有効です。GPIOx\_LCKR レジスタは、次の MCU リセットまたはペリフェラルリセットまでロックされます。

ロックキー書き込みシーケンス :

書き込み LCKR[16] = "1" + LCKR[15:0]

書き込み LCKR[16] = "0" + LCKR[15:0]

書き込み LCKR[16] = "1" + LCKR[15:0]

読み出し LCKR

読み出し LCKR[16] = "1" (この読み出し操作は任意だが、ロックが有効であることを確認できる。)

**注：** ロックキー書き込みシーケンスの間は、LCK[15:0] の値を変更することはできません。

ロックシーケンス中にエラーが発生すると、ロックは中止されます。

ポートの任意のビットの最初のロックシーケンスの後、次の MCU リセットまたはペリフェラルリセットまでは、LCKK ビットのいかなる読み出しアクセスに対しても、“1”が返されます。

ビット 15:0 LCKy : ポート x ロックビット y (y = 0~15)

これらのビットは読み出し/書き込みができますが、書き込めるのは LCKK ビットが“0”のときだけです。

0 : ポート設定はロックされません。

1 : ポート設定はロックされます。

## 6.4.9 GPIO オルタネート機能下位レジスタ (GPIOx\_AFRL) (x = A~K)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
r/w	r/w	r/w	r/w												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
r/w	r/w	r/w	r/w												

ビット 31:0 AFRy[3:0] : ポート x ピン y (y = 0~7) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFSEly 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15

## 6.4.10 GPIO オルタネート機能上位レジスタ (GPIOx\_AFRH) (x = A~J)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
r/w	r/w	r/w	r/w												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
r/w	r/w	r/w	r/w												

ビット 31:0 AFRy[3:0] : ポート x ピン y (y = 8~15) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFSEly 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15



## 6.4.11 GPIO レジスタマップ

次の表に、GPIO レジスタマップとリセット値を示します。

表 22. GPIO レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	GPIOB_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	
0x00	GPIOx_MODER (x = C~K)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	GPIOx_OTYPER (x = A~K)	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOA_OSPEEDR	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	リセット値	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	GPIOB_OSPEEDR	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	
0x08	GPIOx_OSPEEDR (x = C~K)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	GPIOA_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	リセット値	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	GPIOB_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
0x10	GPIOx_IDR (x = A~I/J/K)	Res.	Res.	Res.	Res.	Res.	Res.	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0										
	リセット値																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x



表 22. GPIO レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14	GPIOx_ODR (x = A~K)	Res.	Res.	Res.	Res.	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0												
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (x = A~I/J/K)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	GPIOx_LCKR (x = A~K)	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0												
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (x = A~K)	AFR7[3:0]			AFR6[3:0]			AFR5[3:0]			AFR4[3:0]			AFR3[3:0]			AFR2[3:0]			AFR1[3:0]			AFR0[3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (x = A~J)	AFR15[3:0]			AFR14[3:0]			AFR13[3:0]			AFR12[3:0]			AFR11[3:0]			AFR10[3:0]			AFR9[3:0]			AFR8[3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 7 システム設定コントローラ (SYSCFG)

システム設定コントローラは、主に以下の用途で使用されます。

- メモリ領域の再配置
- イーサネット PHY インタフェースの選択
- GPIO への外部割り込みライン接続の管理

### 7.1 I/O 補正セル

デフォルトでは I/O 補正セルは使用されません。しかしながら、I/O 出力バッファ速度が 50 MHz または 100 MHz モードに設定されている場合、I/O の状態変化  $t_{r(10)out}/t_{r(10)out}$  のスルーレート制御のため、補正セルを使用して電源の I/O ノイズを低減することを推奨します。

補正セルが有効になると、補正セルが使用可能であることを示すため READY フラグがセットされ、使用可能となります。I/O 補正セルは、電源電圧が 2.4 から 3.6 V の範囲にある場合にのみ使用できます。

### 7.2 SYSCFG レジスタ

#### 7.2.1 SYSCFG メモリ再配置レジスタ (SYSCFG\_MEMRMP)

このレジスタは、メモリ配置時の特定の設定に使用されます。

- 1 ビットは、BOOT\_ADD0 または BOOT\_ADD1 のどちらかのオプションバイトがブートメモリベースアドレスを定義するかを示すために使用されます。
- その他のビットは、FMC SDRAM バンクを FMC NOR/PSRAM バンクとスワップするために使用されます。

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **SWP\_FMC[1:0]** : FMC メモリマッピングスワップ

ソフトウェアでセット/クリアされます。これらのビットは、デフォルトの MPU 属性を変更することなく SDRAM バンクからのコード実行を可能にすることを目的とし、FMC SDRAM バンクと FMC NOR/PSRAM をスワップするために使用されます。

00 : FMC メモリマッピングのスワッピングはありません。

SDRAM バンク 1 およびバンク 2 は、それぞれ 0xC000 0000 および 0xD000 0000 にマッピングされます (デフォルトのマッピング)。

NOR/RAM は 0x60000000 でアクセス可能です (デフォルトのマッピング)。

01 : NOR/RAM および SDRAM メモリマッピングはスワップされ、

SDRAM バンク 1 およびバンク 2 は、それぞれ 0x6000 0000 および 0x7000 0000 にマッピングされます。

NOR/PSRAM バンクは、0xC000 0000 にマッピングされます。

10 : 予約済み

11 : 予約済み

ビット 9:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **MEM\_BOOT** : メモリブート配置

このビットは、BOOT\_ADD0 または BOOT\_ADD1 のどちらのオプションバイトがブートメモリアドレスを定義するかを示します。

0 : ブートメモリアドレスは BOOT\_ADD0 オプションバイトによって定義されます (工場リセット値 : TCM-FLASH は 0x00200000 にマッピングされます)。

1 : ブートメモリアドレスは BOOT\_ADD1 オプションバイトによって定義されます。

(工場リセット値 : システムメモリは 0x001 0000 にマッピングされます)。

**注 :** ブートメモリアドレス選択の詳細は、セクション 2.3 : メモリマップを参照してください。

## 7.2.2 SYSCFG ペリフェラルモード設定レジスタ (SYSCFG\_PMC)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MII_RMII_SEL	Res.	Res.	Res.	Res.	ADCxDC2									
								rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **MII\_RMII\_SEL** : イーサネット PHY インタフェース選択

ソフトウェアによってセット/クリアされます。これらのビットによって、イーサネット MAC の PHY インタフェースを制御します。

0 : MII インタフェースが選択されます。

1 : RMII PHY インタフェースが選択されます。

**注 :** この設定は、MAC のリセット期間中かつ MAC クロックを有効にする前に行う必要があります。

ビット 22:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **ADCxDC2** :

0 : 影響はありません。

1 : このビットの使い方は、AN4073 を参照してください。

**注 :** これらのビットは、次の条件が満足されたときのみセットできます。

- ADC クロックが 30 MHz 以上

- 複数の ADC 変換が同時に開始せず、サンプリング時間が異なる場合、ADCxDC2 ビットは、ただ 1 つだけ選択する必要があります。

- PWR\_CR レジスタで ADCDC1 ビットがセットされている場合、これらのビットをセットすることはできません。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

## 7.2.3 SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG\_EXTICR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
r/w	r/w	r/w	r/w												

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 0~3)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン

0111 : PH[x] ピン

1000 : PI[x] ピン

1001 : PJ[x] ピン

1010 : PK[x] ピン

## 7.2.4 SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG\_EXTICR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
r/w	r/w	r/w	r/w												

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 4~7)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン

0111 : PH[x] ピン

1000 : PI[x] ピン

1001 : PJ[x] ピン

1010 : PK[x] ピン

## 7.2.5 SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG\_EXTICR3)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 8~11)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : PF[x] ピン

0110 : PG[x] ピン

0111 : PH[x] ピン

1000 : PI[x] ピン

1001 : PJ[x] ピン

1010 : PK[x] ピン

**注 :** **PK[11:8] は使用されません。**

## 7.2.6 SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG\_EXTICR4)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
r/w	r/w	r/w	r/w												

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 12~15)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- 0000 : PA[x] ピン
- 0001 : PB[x] ピン
- 0010 : PC[x] ピン
- 0011 : PD[x] ピン
- 0100 : PE[x] ピン
- 0101 : PF[x] ピン
- 0110 : PG[x] ピン
- 0111 : PH[x] ピン
- 1001 : PJ[x] ピン
- 1010 : PK[x] ピン

**注 :** PK[15:12] は使用されません。

## 7.2.7 補正セル制御レジスタ (SYSCFG\_CMPCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **READY** : 補正セルレディフラグ

- 0 : I/O 補正セルはレディ状態ではありません。
- 1 : I/O 補正セルはレディ状態です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CMP\_PD** : 補正セルパワーダウン

- 0 : I/O 補正セルパワーダウンモード
- 1 : I/O 補正セルは有効です。



## 7.2.8 SYSCFG レジスタマップ

次の表に、SYSCFG レジスタマップとリセット値を示します。

表 23. SYSCFG レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SYSCFG_MEMRMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWP_FMC(1:0)		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM_BOOT								
	リセット値																						0	0									
0x04	SYSCFG_PMC	Res.	MIL_RMII_SEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値									0						0	0	0															
0x08	SYSCFG_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[3:0]			EXTI2[3:0]			EXTI1[3:0]			EXTI0[3:0]														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	SYSCFG_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7[3:0]			EXTI6[3:0]			EXTI5[3:0]			EXTI4[3:0]														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	SYSCFG_EXTICR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11[3:0]			EXTI10[3:0]			EXTI9[3:0]			EXTI8[3:0]														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	SYSCFG_EXTICR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15[3:0]			EXTI14[3:0]			EXTI13[3:0]			EXTI12[3:0]														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	SYSCFG_CMPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	CMP_PD								
	リセット値																									0							

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 8 ダイレクトメモリアクセスコントローラ (DMA)

### 8.1 DMA の概要

DMA (Direct Memory Access) は、ペリフェラルとメモリ間、およびメモリ間で高速データ転送を行うために使用します。データは、CPU の動作を必要とすることなく、DMA によって高速に移動できます。これにより、CPU リソースを他の操作のためにあけておくことができます。

このDMA コントローラは、高機能なバスマトリックスアーキテクチャに基づいてシステムの帯域幅を最大限に利用するため、強力なデュアル AHB マスタバスアーキテクチャと、独立した FIFO を兼ね備えています。

2 つの DMA コントローラは合計 16 ストリーム (各コントローラごとに 8 つ) を搭載し、それぞれが 1 つ以上のペリフェラルからのメモリアセリクエストを管理します。各ストリームは、合計で最大 8 チャンネル (リクエスト) を処理することができます。また、それぞれ DMA リクエスト間の優先順位を操作するためのアービタを内蔵しています。

### 8.2 DMA の主な機能

DMA の主な機能は次のとおりです。

- デュアル AHB マスタバスアーキテクチャ (メモリアクセス専用、およびペリフェラルへのアクセス専用)
- 32 ビットアクセスのみをサポートする AHB スレーブプログラミングインタフェース
- DMA コントローラごとに 8 ストリーム、ストリームごとに最大 8 チャンネル (リクエスト)
- ストリームごとに独立した 4 つの 32 ビットファーストインファーストアウトメモリバッファ (FIFO)。 (FIFO モードまたはダイレクトモードで使用可能)
  - FIFO モード: FIFO サイズの 1/4、1/2 または 3/4 からソフトウェアで選択可能な閾値レベルを持つ
  - ダイレクトモード  
各 DMA リクエストによって、直ちにメモリとの転送を開始します。DMA がダイレクトダイレクトモード (FIFO が無効) に設定されている場合、“メモリからペリフェラル”モードでデータを転送するため、ペリフェラルからの DMA リクエストが発生した際、即座にかつ確実にデータ転送が行われるよう、DMA はデータをメモリから内部 FIFO へ 1 つだけプリロードします。
- 各ストリームはハードウェアによって次のように設定することができます。
  - “ペリフェラルからメモリ”、“メモリからペリフェラル”および“メモリ間”転送をサポートするレギュラチャネル
  - メモリ側でダブルバッファリングもサポートするダブルバッファチャネル
- 8 つのストリームのそれぞれは、専用のハードウェア DMA チャネル (リクエスト) に接続されます。
- DMA ストリームリクエスト間の優先順位は、ソフトウェアによってプログラミングできます (最高、高、中、低の 4 レベル)。レベルが等しい場合はハードウェアによって決まります (リクエスト 0 はリクエスト 1 よりも優先、など) です。

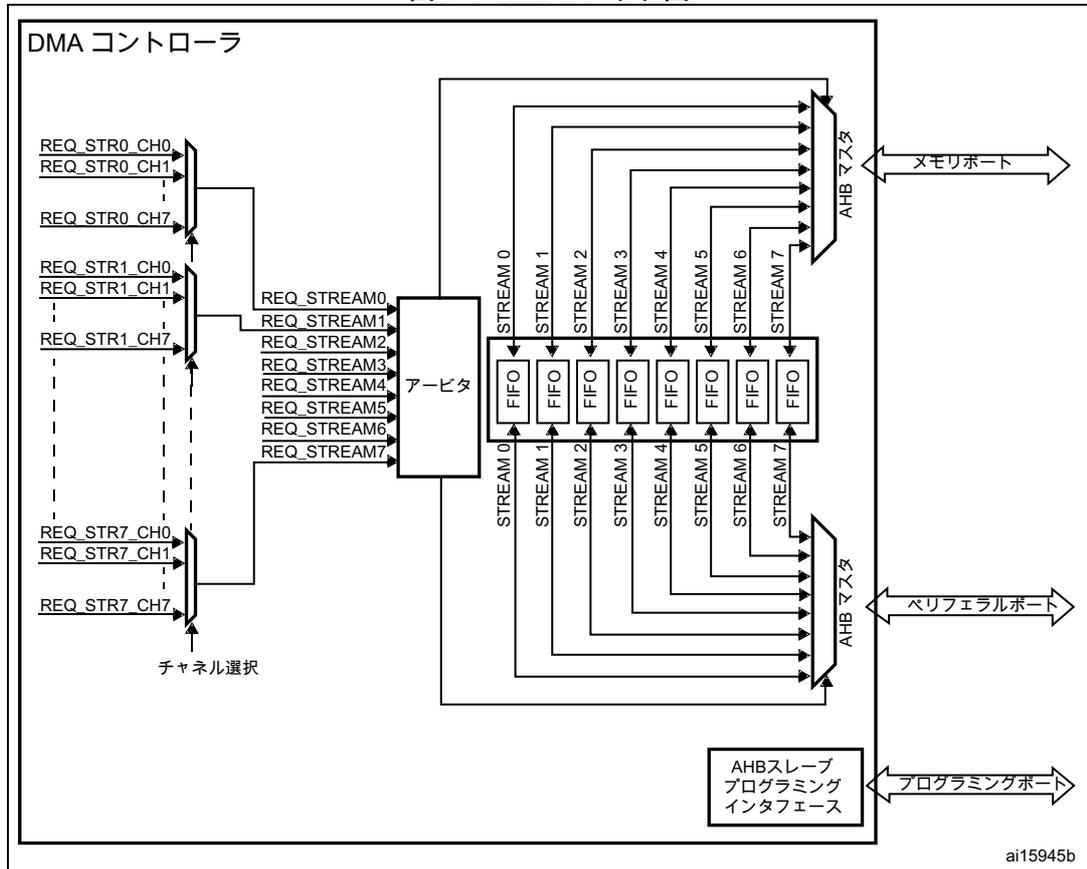
- 各ストリームは、メモリ間転送のためのソフトウェアトリガもサポートしています (DMA2 コントローラでのみ使用可能)。
- 各ストリームリクエストは、発生しうる最大 8 つのチャネルリクエストから選択することができます。この選択は、ソフトウェアによって設定可能であり、これによっていくつかのペリフェラルが DMA リクエストを発生することができるようになります。
- 転送するデータの数、DMA コントローラまたはペリフェラルのいずれかによって管理することができます。
  - DMA フローコントローラ：転送するデータ数は、1 から 65535 の間でソフトウェアによってプログラム可能です。
  - ペリフェラルフローコントローラ：転送されるデータ項目の数は不明であり、ハードウェアによって転送の終了を通知する転送元または転送先ペリフェラルによって制御されます。
- 転送元および転送先の転送幅 (バイト、ハーフワード、ワード) に依存しない：転送元と転送先のデータ幅が異なる場合、DMA によって自動的に必要な転送のパック/アンパックが行われ、バンド幅が最適化されます。この機能は、FIFO モードでのみ使用できます。
- 転送元および転送先に対するインクリメントアドレッシングまたはノンインクリメントアドレッシング
- 4、8、または 16 ビートのインクリメンタルバースト転送をサポート。バーストのサイズはソフトウェアで設定可能であり、通常はペリフェラルの FIFO サイズの半分です。
- サークュラバッファ管理は、各ストリームがサポートします。
- 各ストリームに対する同一の割り込みリクエストにおいて、5 つのイベントフラグ (DMA 1/2 転送、DMA 転送完了、DMA 転送エラー、DMA FIFO エラー、ダイレクトモードエラー) の論理和がとられます。

## 8.3 DMA の機能説明

### 8.3.1 概要

図 22 に、DMA のブロック図を示します。

図 22. DMA ブロック図



DMA コントローラはダイレクトメモリ転送を行います。AHB マスタとして、AHB トランザクションを開始するための AHB バスマトリックスの制御を行うことができます。

以下のトランザクションを実行することができます。

- ペリフェラルからメモリ
- メモリからペリフェラル
- メモリ間

この DMA コントローラは 2 つの AHB マスタポートを搭載しています。メモリに接続するための AHB メモリポート、およびペリフェラルに接続するための AHB ペリフェラルポートです。しかし、メモリ間転送を可能とするため、AHB ペリフェラルポートはメモリにもアクセスできる必要があります。

AHB スレーブポートは DMA コントローラをプログラムするために用いられます (32 ビットアクセスのみサポート)。

**注:** **DMA1 コントローラの AHB ペリフェラルポートは、DMA2 コントローラのようにバスマトリックスに接続されていないので、DMA2 ストリームだけがメモリ間転送を実行することができます。**

2 つの DMA コントローラのシステムへの実装については 図 1 を参照してください。

## 8.3.2 DMA トランザクション

DMA トランザクションは、一定数のデータ転送の列で構成されています。転送されるデータ項目の数とその幅 (8 ビット、16 ビット、または 32 ビット) は、ソフトウェアでプログラム可能です。

各 DMA 転送は次の 3 つの操作で構成されます。

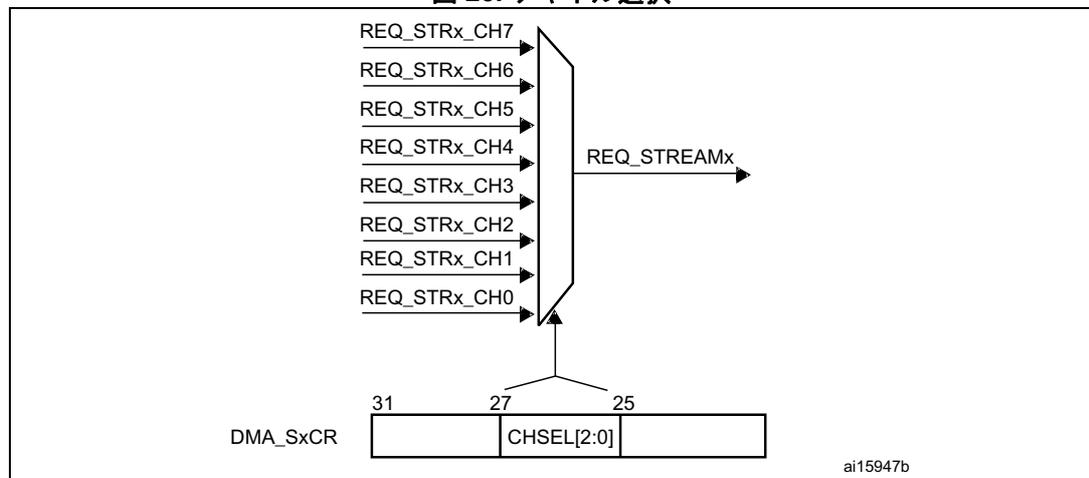
- DMA\_SxPAR または DMA\_SxM0AR レジスタを通じてアドレス指定されたメモリ上の場所またはペリフェラルデータレジスタからのデータのロード。
- DMA\_SxPAR または DMA\_SxM0AR レジスタを通じてアドレス指定されたメモリ内の場所またはペリフェラルデータレジスタにロードされたデータの格納。
- このあと予定される DMA トランザクションの数を保持している DMA\_SxNDTR レジスタをカウントダウンします。

イベントの終了後、ペリフェラルは DMA コントローラにリクエスト信号を送信します。DMA コントローラは、チャンネルの優先順位に応じて、リクエストを処理します。DMA コントローラがペリフェラルにアクセスするとすぐに、DMA コントローラはペリフェラルに確認応答信号を送信します。ペリフェラルは、DMA コントローラからの確認応答信号を受け取るとすぐにそのリクエストを解除します。ペリフェラルによってリクエストがネゲートされると、DMA コントローラは確認応答信号を解除します。さらにリクエストがある場合、ペリフェラルは次のトランザクションを開始できます。

## 8.3.3 チャンネル選択

各ストリームは、8 つのチャンネルリクエストの中から選択することができる DMA リクエストに関連付けられています。チャンネル選択は DMA\_SxCR レジスタの CHSEL[2:0] ビットによって制御されます。

図 23. チャンネル選択



ペリフェラルからの 8 つのリクエスト (TIM、ADC、SPI、I2C など) は各チャンネルに個々に接続されており、それらの接続は製品の実装によって異なります。

表 24 および表 25 に DMA リクエストマッピングの例を示します。

表 24. DMA1 リクエストマッピング

ペリフェラルリクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 0	SPI3_RX	SPDIFRX_DT	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	SPDIFRX_CS	SPI3_TX
チャンネル 1	I2C1_RX	I2C3_RX	TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX

表 24. DMA1 リクエストマッピング (続き)

ペリフェラル リクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 2	TIM4_CH1	-	I2C4_RX	TIM4_CH2	-	I2C4_TX	TIM4_UP	TIM4_CH3
チャンネル 3	-	TIM2_UP TIM2_CH3	I2C3_RX	-	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
チャンネル 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
チャンネル 5	UART8_TX	UART7_TX	TIM3_CH4 TIM3_UP	UART7_RX	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX	TIM3_CH3
チャンネル 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	-	TIM5_UP	-
チャンネル 7	-	TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

表 25. DMA2 リクエストマッピング

ペリフェラル リクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 0	ADC1	SAI1_A	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A	ADC1	SAI1_B	TIM1_CH1 TIM1_CH2 TIM1_CH3	SAI2_B
チャンネル 1	-	DCMI	ADC2	ADC2	SAI1_B	SPI6_TX	SPI6_RX	DCMI
チャンネル 2	ADC3	ADC3	-	SPI5_RX	SPI5_TX	CRYP_OUT	CRYP_IN	HASH_IN
チャンネル 3	SPI1_RX	-	SPI1_RX	SPI1_TX	SAI2_A	SPI1_TX	SAI2_B	QUADSPI
チャンネル 4	SPI4_RX	SPI4_TX	USART1_RX	SDMMC1	-	USART1_RX	SDMMC1	USART1_TX
チャンネル 5	-	USART6_RX	USART6_RX	SPI4_RX	SPI4_TX	-	USART6_TX	USART6_TX
チャンネル 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
チャンネル 7	-	TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX	SPI5_TX	TIM8_CH4 TIM8_TRIG TIM8_COM

### 8.3.4 アービタ

アービタは、2つの AHB マスタポート (メモリポートおよびペリフェラルポート) のそれぞれについて、その優先順位に基づいて 8つの DMA ストリームリクエストを管理し、ペリフェラル/メモリアクセスシーケンスを起動します。

優先順位は 2段階で管理されます。

- ソフトウェア : 各ストリームの優先順位は、DMA\_SxCR レジスタで設定できます。4つのレベルがあります。
  - 最優先
  - 高優先
  - 中優先
  - 低優先
- ハードウェア : 2つのリクエストのソフトウェア優先順位レベルが同じ場合、小さな番号のストリームが大きな番号のストリームよりも優先されます。たとえば、ストリーム 2 はストリーム 4 よりも優先されます。

## 8.3.5 DMA ストリーム

8 つの DMA コントローラストリームのそれぞれに、転送元と転送先間の単方向転送リンクを備えています。

各ストリームは次のことを行うよう設定することができます。

- レギュラータイプトランザクション: メモリからペリフェラル、ペリフェラルからメモリまたはメモリ間の転送
- ダブルバッファタイプトランザクション: メモリ用に 2 つのメモリポインタを使用したダブルバッファ転送 (DMA がバッファの読み書き込みを行っている間、アプリケーションは他のバッファの読み/書きが可能)

転送するデータの量 (65535 以内) はプログラム可能で、ペリフェラル AHB ポートに接続された DMA 転送を要求するペリフェラルのデータ幅に関連しています。転送されるデータ数を格納しているレジスタは、トランザクションの終了ごとにデクリメントされます。

## 8.3.6 転送元、転送先、および転送モード

転送元および転送先は両方とも、アドレス 0x0000 0000 と 0xFFFF FFFF の間の全 4 GB メモリ領域内のペリフェラルおよびメモリを転送アドレスとして使用できます。

方向は DMA\_SxCR レジスタの DIR[1:0] ビットを使用して設定し、メモリからペリフェラル、ペリフェラルからメモリ、およびメモリ間の転送の 3 つの選択肢があります。表 26 に対応する転送元と転送先アドレスを示します。

表 26. 転送元および転送先アドレス

DMA_SxCR レジスタの DIR [1:0] ビット	方向	転送元アドレス	転送先アドレス
00	ペリフェラルからメモリ	DMA_SxPAR	DMA_SxM0AR
01	メモリからペリフェラル	DMA_SxM0AR	DMA_SxPAR
10	メモリ間	DMA_SxPAR	DMA_SxM0AR
11	予約済み	-	-

DMA\_SxCR レジスタの PSIZE または MSIZE ビットでプログラムされたデータ幅が、それぞれハーフワードまたはワードである場合、DMA\_SxPAR または DMA\_SxM0AR/M1AR レジスタに書き込まれたペリフェラルアドレスまたはメモリアドレスは、それぞれワードアドレス境界またはハーフワードアドレス境界で整列する必要があります。

### “ペリフェラルからメモリ”モード

図 24 でこのモードを説明しています。

このモードが有効になっている場合 (DMA\_SxCR レジスタの EN ビットをセットすることにより)、ペリフェラルリクエストが発生するたびに、ストリームは転送元からの転送を開始し、FIFO を満たします。

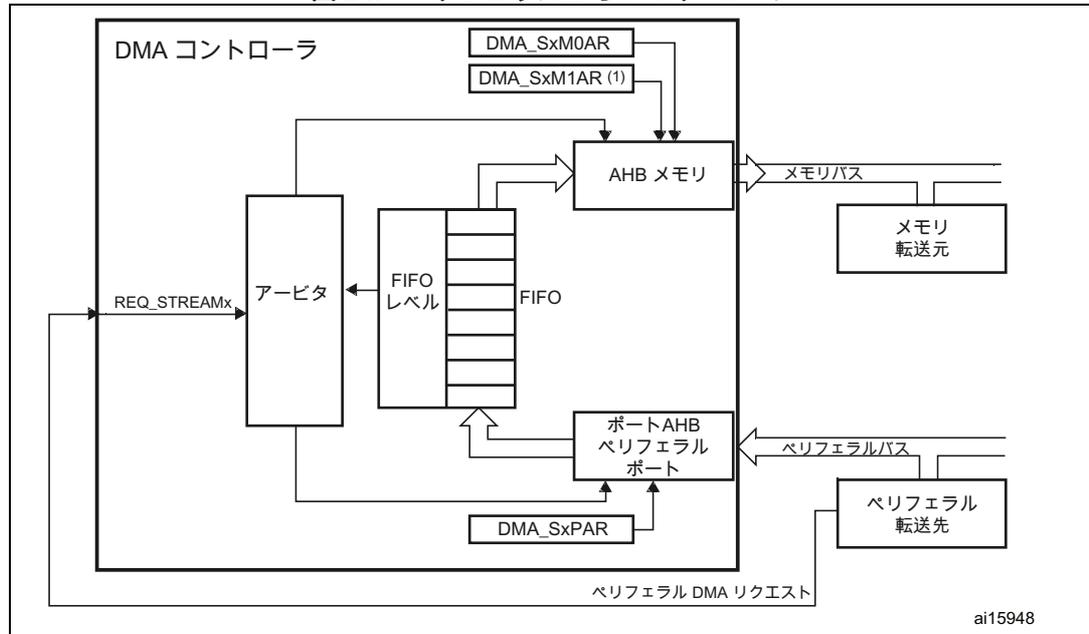
FIFO の閾値レベルに達すると、FIFO の内容が排出され、転送先に格納されます。

DMA\_SxNDTR レジスタがゼロに達したか、ペリフェラルが転送の終了を要求したか (ペリフェラルフローコントローラの場合)、またはソフトウェアによって DMA\_SxCR レジスタの EN ビットがクリアされたとき、転送は停止します。

ダイレクトモード (DMA\_SxFCR レジスタの DMDIS 値が“0”のとき) では、FIFO の閾値レベルは使用されません。ペリフェラルから FIFO への各データの転送後、対応するデータはすぐに排出されて転送先に格納されます。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA\_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

図 24. “ペリフェラルからメモリ”モード



1. ダブルバッファモード用。

## “メモリからペリフェラル”モード

図 25 でこのモードを説明しています。

このモードが有効になっている場合 (DMA\_SxCR レジスタの EN ビットをセットしている場合)、ストリームはすぐに転送元から転送を開始し、FIFO を完全に満たします。

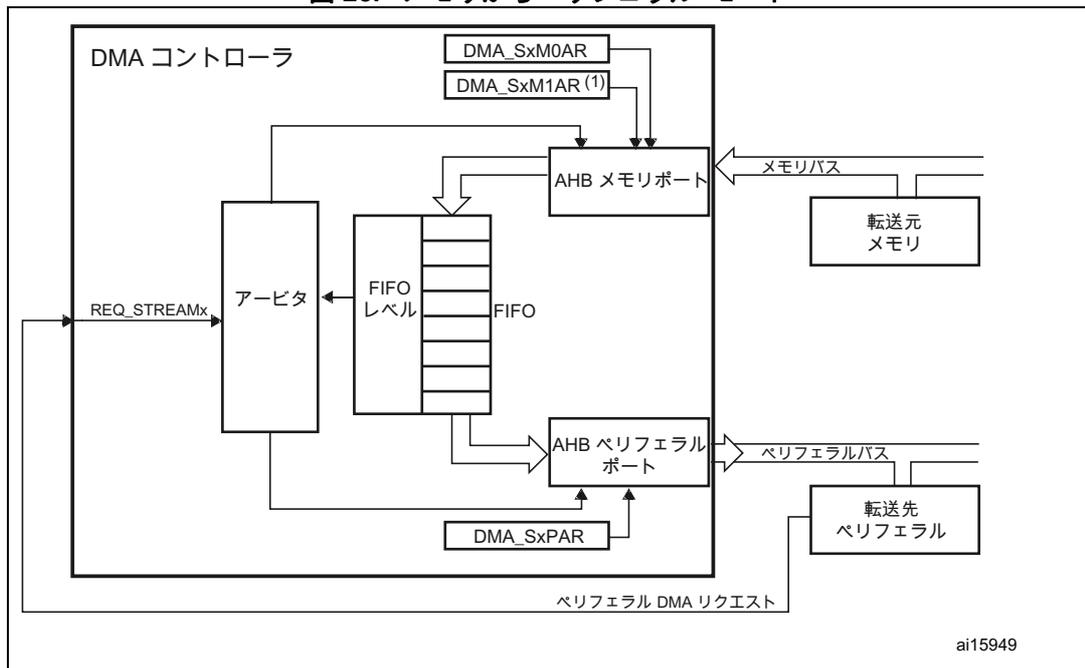
ペリフェラルリクエストが発生するたびに、FIFO のデータが転送されて転送先に格納されます。FIFO のレベルがあらかじめ設定された閾値レベルより低いとき、FIFO はメモリからのデータで完全に再ロードされます。

DMA\_SxNDTR レジスタがゼロに達したか、ペリフェラルが転送の終了を要求したか (ペリフェラルフローコントローラの場合)、またはソフトウェアによって DMA\_SxCR レジスタの EN ビットがクリアされたとき、転送は停止します。

ダイレクトモード (DMA\_SxFCR レジスタの DMDIS 値が“0”のとき) では、FIFO の閾値レベルは使用されません。ストリームが有効になると、DMA は内部 FIFO に転送するための最初のデータをプリロードします。ペリフェラルがデータ転送を要求すると、DMA はプリロードされたデータを設定された転送先へすぐに転送します。そしてまた転送すべき次のデータを空の内部 FIFO にリロードします。プリロードデータサイズは、DMA\_SxCR レジスタの PSIZE ビットフィールドの値に対応します。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA\_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

図 25. “メモリからペリフェラル”モード



1. ダブルバッファモード用。

## メモリ間モード

DMA チャンネルは、ペリフェラルからのリクエストによってトリガされなくても機能します。これは [図 26](#) で説明しているメモリ間モードです。

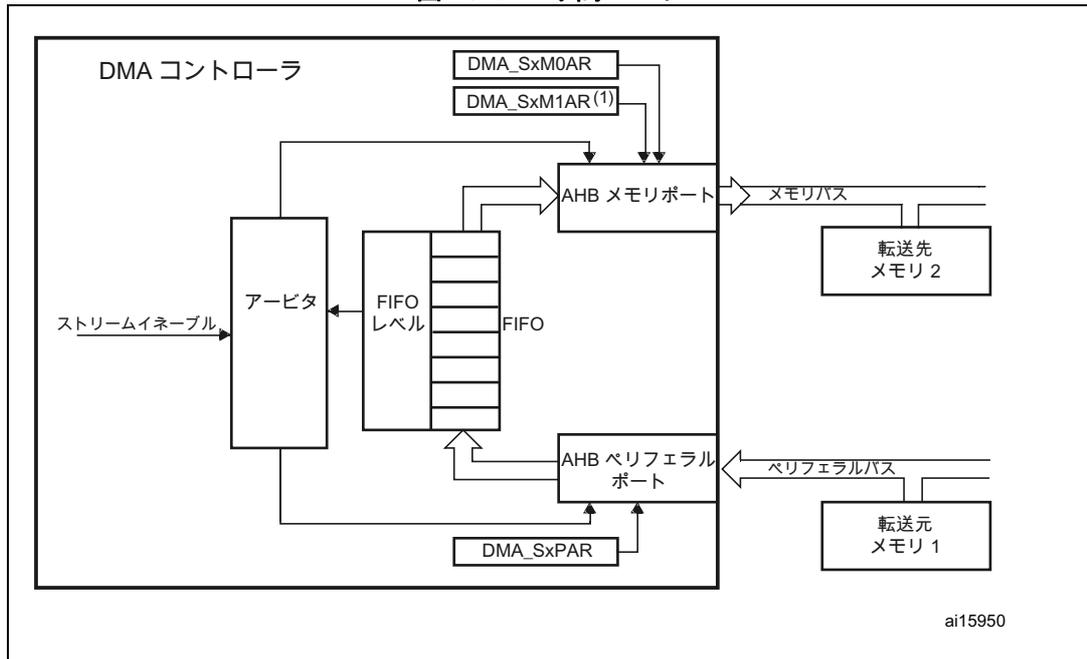
DMA\_SxCR レジスタのイネーブルビット (EN) をセットすることによってストリームが有効になっている場合、ストリームはすぐに閾値レベルまで FIFO への書き込みを開始します。FIFO の閾値レベルに達すると、FIFO のデータが転送され、転送先に格納されます。

DMA\_SxNDTR レジスタがゼロに達するかソフトウェアによって DMA\_SxCR レジスタの EN ビットがクリアされたとき転送を停止します。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA\_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

**注：** **メモリ間モードを使用する場合、サーキュラモードおよびダイレクトモードは使用できません。**  
**DMA2 コントローラのみメモリ間転送を行うことができます。**

図 26. メモリ間モード



1. ダブルバッファモード用。

### 8.3.7 ポインタのインクリメント

ペリフェラルやメモリのポインタは、DMA\_SxCR レジスタの PINC および MINC ビットに応じて、各転送後に自動的にポストインクリメントまたは一定に保つことができます。

インクリメントモードの無効化は、転送元または転送先ペリフェラルデータを 1 つのレジスタを介してアクセスする場合に便利です。

インクリメントモードが有効な場合、次の転送アドレスは、DMA\_SxCR レジスタの PSIZE または MSIZE ビットにプログラムされたデータ幅に応じて、前回のアドレスに 1 (バイトの場合)、2 (ハーフワードの場合)、または 4 (ワードの場合) をインクリメントしたアドレスになります。

パッキング動作を最適化するため、AHB ペリフェラルポート上で転送されるデータのサイズに関わらずペリフェラルアドレスのインクリメントオフセットサイズを固定することができます。DMA\_SxCR レジスタの PINCOS ビットは、インクリメントオフセットサイズをペリフェラル AHB ポート上のデータサイズまたは 32 ビットアドレス (この場合アドレスは 4 つインクリメントされます) に調整するために使用されます。PINCOS ビットは AHB ペリフェラルポートにのみ影響します。

PINCOS ビットがセットされている場合、次の転送のアドレスは、PSIZE 値にかかわらず前のアドレスに 4 をインクリメント (自動的に 32 ビットアドレスに調整) したアドレスです。しかしながら、この動作による AHB メモリポートへの影響はありません。

## 8.3.8 サークュラモード

サーキュラモードを使用すると、サーキュラバッファや連続したデータフロー (ADC スキャンモードなど) を処理できます。この機能は、DMA\_SxCR レジスタの CIRC ビットを使用して有効にできます。

サーキュラモードが有効になると、転送されるデータ項目数にはストリーム設定段階でプログラミングされた初期値が自動的に再ロードされ、DMA リクエストの処理が続行されます。

**注：** **サーキュラモードでは、メモリ用にバーストモードが設定されている場合、次のルールを守る必要があります。**

$DMA\_SxNDTR = ((Mburst \text{ ビート}) \text{ の倍数} \times (Msize)/(Psize))$ 、ここで

- $(Mburst \text{ ビート}) = 4, 8 \text{ または } 16$  (DMA\_SxCR レジスタの MBURST ビットによって異なる)
- $((Msize)/(Psize)) = 1, 2, 4, 1/2 \text{ または } 1/4$  (Msize と Psize は DMA\_SxCR レジスタの MSIZE と PSIZE ビットを表す。これらはバイト依存)
- $DMA\_SxNDTR = AHB \text{ ペリフェラルポート上で転送するデータ項目の数}$

**例：** Mburst ビート=8 (INCR8)、MSIZE=00 (バイト) および PSIZE=01 (ハーフワード)、この場合、DMA\_SxNDTR は  $(8 \times 1/2 = 4)$  の倍数でなければなりません。

**この式を守らないと DMA 動作とデータの整合性は保証されません。**

**NDTR もペリフェラルバーストサイズの倍数にペリフェラルデータサイズを乗じた値でなければならず、さもないと不正な DMA 動作を引き起こす可能性があります。**

## 8.3.9 ダブルバッファモード

このモードは、すべての DMA1 と DMA2 ストリームに対して使用可能です。

ダブルバッファモードは、DMA\_SxCR レジスタの DBM ビットをセットすることで有効になります。

ダブルバッファストリームは通常の (シングルバッファ) ストリームとして動作しますが、2つのメモリポインタを持っているところが異なります。ダブルバッファモードが有効になっていると、サーキュラモードが自動的に有効になり (DMA\_SxCR の CIRC ビットは無視)、トランザクションが終わるごとにメモリポインタをスワップします。

このモードでは、DMA コントローラはトランザクションが終わるごとに、あるメモリターゲットから他へスワップします。これにより、DMA 転送によってある 1つのメモリ領域が満杯または使用中である間に、ソフトウェアは別のメモリ領域を処理することができます。ダブルバッファストリームは、[表 27：ダブルバッファモード \(DBM=1\) における転送元および転送先アドレスレジスタ](#)で説明しているように両方向に動作することができます (メモリは転送元または転送先のいずれにも設定可能)。

**注：** **ダブルバッファモードでは、ストリームが有効なとき、以下の条件を守ることにより、いつでも AHB メモリポートのベースアドレス (DMA\_SxM0AR または DMA\_SxM1AR) を更新することができます。**

- DMA\_SxCR レジスタで CT ビットが“0”のとき DMA\_SxM1AR レジスタに書き込むことができます。CT=1 のときに、このレジスタに書き込みを試みると、エラーフラグ (TEIF) がセットされ、ストリームが自動的に無効になります。
- DMA\_SxCR レジスタで CT ビットが“1”のとき DMA\_SxM0AR レジスタに書き込むことができます。CT=0 のときに、このレジスタに書き込みを試みると、エラーフラグ (TEIF) がセットされ、ストリームが自動的に無効になります。

**エラー条件を回避するためには、TCIF フラグがアサートされたらできるだけ早くベースアドレスを変更することを推奨します。なぜなら、この時点で、上記 2つの内いずれかの条件にしたがって**

DMA\_SxCR レジスタの CT の値に応じてターゲットメモリがメモリ 0 から 1 (または 1 から 0) に変更されているためです。

他のすべてのモード (ダブルバッファモードを除く) に対し、ストリームが有効になるとすぐに、メモリアドレスレジスタは書き込み保護されます。

表 27. ダブルバッファモード (DBM=1) における転送元および転送先アドレスレジスタ

DMA_SxCR レジスタの DIR [1:0] ビット	方向	転送元アドレス	転送先アドレス
00	ペリフェラルからメモリ	DMA_SxPAR	DMA_SxM0AR / DMA_SxM1AR
01	メモリからペリフェラル	DMA_SxM0AR / DMA_SxM1AR	DMA_SxPAR
10	設定禁止 <sup>(1)</sup>		
11	予約済み	-	-

1. ダブルバッファモードが有効になっている場合、サーキュラモードが自動的に有効になります。メモリ間モードはサーキュラモードと互換性がないため、ダブルバッファモードが有効になっているときは、メモリ間モードを設定することはできません。

### 8.3.10 プログラム可能なデータ幅、パッキング/アンパッキング、エンディアン形式

転送されるデータ項目の数は、ストリームを有効にする前に DMA\_SxNDTR (転送データ項目数ビット、NDT) にプログラムする必要があります (フローコントローラがペリフェラルで、DMA\_SxCR の PFCTRL ビットがセットされている場合を除く)。

内部 FIFO を使用する場合、転送元と転送先データのデータ幅は DMA\_SxCR レジスタの PSIZE と MSIZE ビットを介してプログラム可能です (8、16、または 32 ビットが可能)。

PSIZE と MSIZE が等しくない場合 :

- DMA\_SxNDTR レジスタに設定されている転送データ項目数のデータ幅は、ペリフェラルバスの幅 (DMA\_SxCR レジスタの PSIZE ビットで設定されている) と同じです。たとえば、ペリフェラルからメモリ、メモリからペリフェラル、またはメモリ間転送のケースで PSIZE[1:0] ビットがハーフワード用に設定されている場合、転送されるバイト数は 2×NDT です。
- DMA コントローラは、転送元と転送先の両方でリトルエンディアンアドレッシングのみ取り扱うことができます。これについては、[表 28 : パッキング/アンパッキングおよびエンディアンの動作 \(ビット PINC=MINC=1\)](#) で説明されています。

このパッキング/アンパッキング手順には、データが完全にパック/アンパックされる前に操作が中断された場合にデータが破損する危険性があります。したがって、データの一貫性を確保するため、ストリームはバースト転送を生成するよう設定することができます。この場合、バーストに属する各転送グループは分割できません ([セクション 8.3.11 : シングル転送とバースト転送](#)を参照)。

ダイレクトモード (DMA\_SxFCR レジスタで DMDIS=0 のとき) では、データのパッキング/アンパッキングはできません。この場合には、転送元と転送先の転送データ幅が異なってはなりません。両方が等しく、DMA\_SxCR の PSIZE ビットで定義します (MSIZE ビットは無視されます)。

表 28. パッキング/アンパッキングおよびエンディアンの動作 (ビット PINC=MINC=1)

AHB メモリ ポート幅	AHB ペリ フェラル ポート 幅	転送する データ 項目の数 (NDT)	メモリ転 送番号	メモリポート アドレス/ バイトレーン	ペリフェラル 転送番号	ペリフェラルポートアドレス/ バイトレーン	
						PINCOS=1	PINCOS=0
8	8	4	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
8	16	2	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
8	32	1	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1	0x0 / B3[B2:B1]B0[31:0]	0x0 / B3[B2:B1]B0[31:0]
16	8	4	1 2	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
16	16	2	1 2	0x0 / B1[B0:15:0] 0x2 / B1[B0:15:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
16	32	1	1 2	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]	1	0x0 / B3[B2:B1]B0[31:0]	0x0 / B3[B2:B1]B0[31:0]
32	8	4	1	0x0 / B3[B2:B1]B0[31:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
32	16	2	1	0x0 / B3[B2:B1]B0[31:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
32	32	1	1	0x0 / B3[B2:B1]B0 [31:0]	1	0x0 / B3[B2:B1]B0 [31:0]	0x0 / B3[B2:B1]B0[31:0]

注： ペリフェラルポートは、転送元または転送先として設定可能です (メモリ間転送の場合はメモリ転送元としての設定も可能です)。

PSIZE、MSIZE、および NDT[15:0] は、最後の転送が不完全にならないよう必ず設定する必要があります。これはペリフェラルポートのデータ幅 (PSIZE ビット) がメモリポートのデータ幅 (MSIZE ビット) より小さい場合に発生する可能性があります。この制約の一覧を表 29 に示します。

表 29. PSIZE と MSIZE に対する NDT に関する制限事項

PSIZE[1:0] of DMA_SxCR	MSIZE[1:0] of DMA_SxCR	NDT[15:0] of DMA_SxNDTR
00 (8 ビット)	01 (16 ビット)	2 の倍数でなければならない
00 (8 ビット)	10 (32 ビット)	4 の倍数でなければならない
01 (16 ビット)	10 (32 ビット)	2 の倍数でなければならない

## 8.3.11 シングル転送とバースト転送

DMA コントローラは、シングル転送を生成することもできれば、4、8、または 16 ビートのインクリメンタルバースト転送を生成することもできます。

バーストのサイズは、DMA\_SxCR レジスタの MBURST[1:0] と PBURST[1:0] ビットを使用して 2 つの AHB ポートに対して別々にソフトウェアで設定されます。

バーストサイズは、バースト内のビート数を示します。転送するバイト数ではありません。

データの一貫性を確保するため、バーストを形成する転送の各グループは分割できません。AHB 転送はロックされ、AHB バスマトリックスのアービタは、バースト転送シーケンス中 DMA マスタからバスの使用を奪取しません。

シングルまたはバーストの設定によって各 DMA リクエストによって AHB ペリフェラルポート上で開始される転送の数は異なります。

- AHB ペリフェラルポートがシングル転送用に設定されている場合、各 DMA リクエストによって、DMA\_SxCR レジスタの PSIZE[1:0] ビットに応じた、バイト、ハーフワード、またはワードのデータ転送が生成されます。
- AHB ペリフェラルポートがバースト転送用に設定されている場合、各 DMA リクエストによって、DMA\_SxCR レジスタの PBURST[1:0] ビットと PSIZE[1:0] ビットに応じた、4、8、または 16 ビートのバイト、ハーフワード、またはワードの転送が生成されます。

MBURST と MSIZE ビットに関しても、AHB メモリポートに対して上記と同じことを考慮する必要があります。

ダイレクトモードでは、ストリームはシングル転送のみ生成することができ、MBURST[1:0] と PBURST[1:0] ビットはハードウェアで決まっています。

アドレスポインタ (DMA\_SxPAR または DMA\_SxM0AR レジスタ) は、バーストブロック内のすべての転送が転送サイズに等しいアドレス境界で整列するように選択する必要があります。

バースト設定は、AHB プロトコルを守るよう選択する必要があります。ここで、1 個のスレーブに割り当てることができる最小アドレス空間が 1 KB であるため、バーストが 1 KB アドレス境界を越えることはできません。つまり、バーストブロック転送が 1 KB アドレス境界を超えてはならず、超えた場合には AHB エラーが生成される可能性があります。このエラーは DMA レジスタで報告されません。

## 8.3.12 FIFO

### FIFO の構造

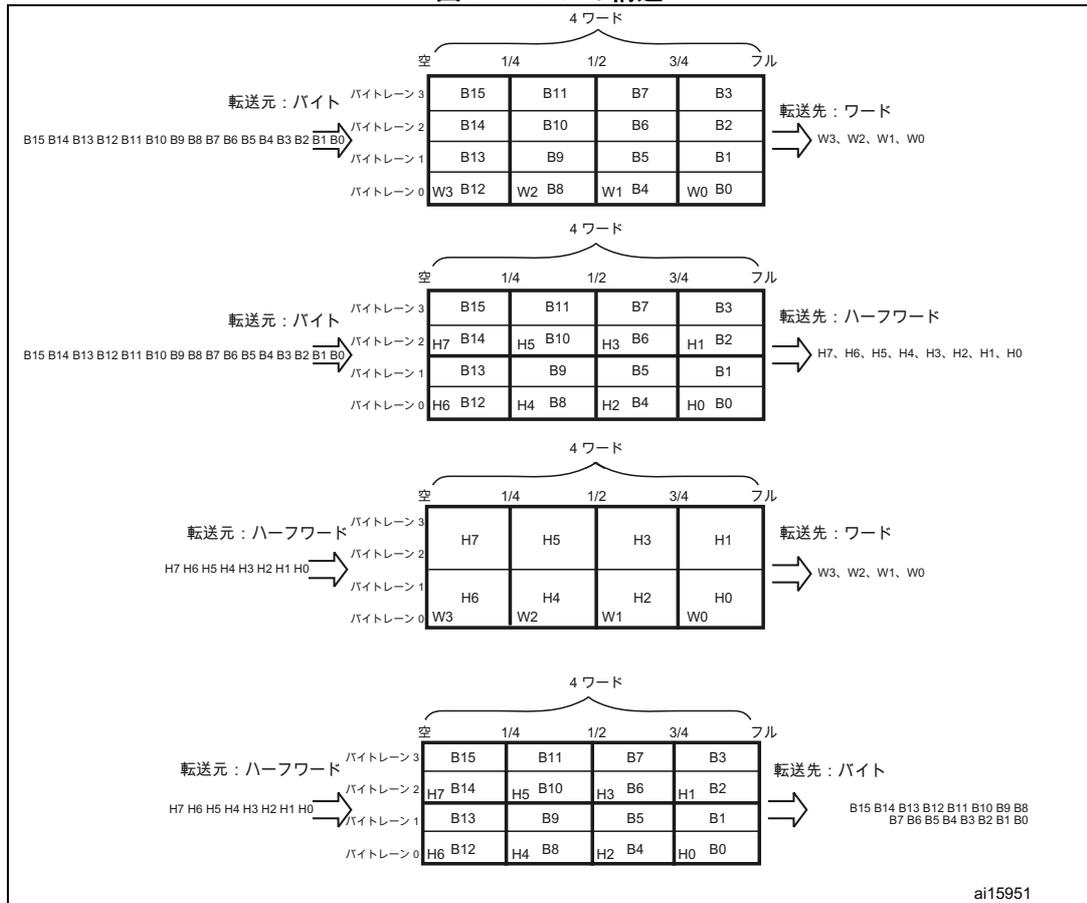
FIFO は転送元からのデータを転送先へ送信する前に一時的に格納するために使用します。

各ストリームには独立した 4 ワードの FIFO があり、閾値レベルはソフトウェアで 1/4、1/2、3/4、またはフルに設定できます。

FIFO 閾値レベルを使用できるようにするには、DMA\_SxFCR レジスタの DMDIS ビットをセットしてダイレクトモードを無効にする必要があります。

FIFO の構造は、転送元および転送先のデータ幅によって異なり、[図 27 : FIFO の構造](#)で説明しています。

図 27. FIFO の構造



## FIFO 閾値とバーストの設定

FIFO の閾値 (DMA\_SxFCR レジスタの FTH[1:0] ビット) とメモリバーストのサイズ (DMA\_SxFCR レジスタの MBURST[1:0]) を選択する場合は注意が必要です。FIFO 閾値が指す内容はメモリバースト転送の整数倍に一致する必要があります。一致しない場合、ストリームが有効なとき FIFO エラー (DMA\_HISR または DMA\_LISR レジスタの FEIFx フラグ) が発生し、ストリームが自動的に無効になります。可能な設定と禁止されている設定の説明は表 30 を参照してください。禁止されている設定は、表内でグレイでハイライトされています。

表 30. FIFO 閾値設定

MSIZE	FIFO レベル	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
バイト	1/4	4 ビートの 1 バースト	設定禁止	設定禁止
	1/2	4 ビートの 2 バースト	8 ビートの 1 バースト	
	3/4	4 ビートの 3 バースト	設定禁止	
	フル	4 ビートの 4 バースト	8 ビートの 2 バースト	16 ビートの 1 バースト

表 30. FIFO 閾値設定 (続き)

MSIZE	FIFO レベル	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
ハーフワード	1/4	設定禁止	設定禁止	設定禁止
	1/2	4 ビートの 1 パースト		
	3/4	設定禁止		
	フル	4 ビートの 2 パースト		
ワード	1/4	設定禁止	設定禁止	設定禁止
	1/2			
	3/4			
	フル			

いかなる場合でも、パーストサイズにデータサイズを乗じた値が FIFO サイズを超えることはできません (可能なデータサイズは 1 (バイト)、2 (ハーフワード) または 4 (ワード))。

次のいずれかの条件が発生した場合、DMA 転送の終了時に、パースト転送が不完全となる可能性があります。

- AHB ペリフェラルポート設定において、データ項目の合計数 (DMA\_SxNDTR レジスタに設定) が、パーストサイズにデータサイズを乗じた値の倍数ではない
- AHB メモリポート設定において、メモリに転送する FIFO 内の残りのデータ項目数が、パーストサイズにデータサイズを乗じた値の倍数ではない

このような場合、DMA ストリーム設定時にパーストトランザクションが要求されていても、転送すべき残りのデータは、DMA によってシングルモードで管理されます。

**注:** ペリフェラル AHB ポートでパースト転送が要求されており、FIFO が使用されている場合 (DMA\_SxCR レジスタの DMDIS=1)、DMA ストリーム方向に応じて、恒久的なアンダーランまたはオーバーラン状態を回避するため、次のルールを守ることが必要です。

$(PBURST \times PSIZE) = FIFO\_SIZE$  (4 ワード) の場合、 $PSIZE = 1, 2$ 、または 4 であり、かつ、 $PBURST = 4, 8$ 、または 16 の時、 $FIFO\_Threshold = 3/4$  は禁止。

このルールによって、ペリフェラルからの要求を一度に処理するのに十分な FIFO スペースを確実に確保できます。

## FIFO フラッシュ

FIFO は DMA\_SxCR レジスタの EN ビットをリセットすることによりストリームが無効化された場合、およびストリームが“ペリフェラルからメモリ”転送またはメモリ間転送を管理するよう設定された場合、フラッシュすることができます。ストリームを無効化したときに FIFO にデータが残っている場合、DMA コントローラは引き続き残りのデータを転送先へ転送します (ストリームが事実上無効化されているとしても)。このフラッシュが完了すると、DMA\_LISR または DMA\_HISR レジスタの転送完了ステータスビット (TCIFx) がセットされます。

この場合、残りのデータカウンタ DMA\_SxNDTR は値を保持し、現在いくつかのデータ項目が転送先メモリに格納できるかを示します。

FIFO フラッシュ動作中に、FIFO 内のメモリに転送する残りのデータ項目数 (バイト単位) がメモリデータ幅より少ない場合 (たとえば、MSIZE がワードに設定されているとき FIFO 内に 2 バイト)、データは DMA\_SxCR レジスタの MSIZE ビットに設定されたデータ幅で送信されます。つまり、メモリには不要な値が書き込まれることとなります。ソフトウェアで DMA\_SxNDTR レジスタを読み出すことにより、正しいデータが格納されたメモリ領域 (開始アドレスと終了アドレス) を判断することができます。



FIFO 内の残りのデータ項目数がバーストサイズよりも小さい場合 (AHB メモリポート上のバーストを管理するストリームを設定するため DMA\_SxCR レジスタの MBURST ビットがセットされている場合)、FIFO フラッシュを完了するためシングルトランザクションが生成されます。

## ダイレクトモード

デフォルトでは、FIFO はダイレクトモードで動作し (DMA\_SxFCR の DMDIS ビットがリセットされています)、FIFO 閾値レベルは使用しません。このモードは、システムが各 DMA リクエスト後にメモリとの即時シングル転送を必要とする場合に便利です。

DMA がダイレクトモード (FIFOが無効) に設定されている場合、“メモリからペリフェラル”モードでデータを転送するため、ペリフェラルからの DMA リクエストが発生した際、即座にかつ確実にデータ転送が行われるように、DMA はデータをメモリから内部 FIFO へ 1 つプリロードします。

FIFO の飽和を避けるためには、対応するストリームを高い優先度で設定することを推奨します。

このモードは、次のような転送に制限されています。

- 転送元および転送先の転送幅が等しく、両方とも DMA\_SxCR の PSIZE[1:0] ビットによって定義されている (MSIZE[1:0] ビットは無視される)
- バースト転送は不可 (DMA\_SxCR の PBURST[1:0] と MBURST[1:0] ビットは無視される)

メモリ間転送を実行する際はダイレクトモードを使用することはできません。

### 8.3.13 DMA 転送の完了

DMA\_LISR または DMA\_HISR ステータスレジスタの TCIFx ビットを設定することにより、さまざまなイベントによって転送を終了させることができます。

- DMA フローコントローラモード：
  - “メモリからペリフェラル”モードで DMA\_SxNDTR カウンタがゼロに達した
  - (DMA\_SxCR レジスタの EN ビットをクリアすることにより) ストリームが転送終了する前に無効化され、(転送がペリフェラルからメモリまたはメモリ間の場合) 残りのすべてのデータが FIFO からメモリにフラッシュされた
- ペリフェラルフローコントローラモード：
  - 最後の外部バーストまたはシングルリクエストがペリフェラルから生成され、(DMA が“ペリフェラルからメモリ”モードで動作しているとき) 残りのデータが FIFO からメモリへ転送された
  - ストリームがソフトウェアによって無効化され、(DMA が“ペリフェラルからメモリ”モードで動作しているとき)、残りのデータが FIFO からメモリへ転送された

**注：** **転送完了は、“ペリフェラルからメモリ”モードの場合のみ、メモリに転送される FIFO 内の残りのデータによって異なります。この条件は、“メモリからペリフェラル”モードには適用されません。**

ストリームがノンサーキュラモードで設定されている場合、転送終了後 (転送すべきデータの数がゼロに達したとき)、DMA は停止し (DMA\_SxCR レジスタの EN ビットがハードウェアでクリアされます)、ソフトウェアによってストリームを再プログラムして再有効化 (DMA\_SxCR レジスタの EN ビットを設定) しない限り、DMA リクエストは処理されません。

## 8.3.14 DMA 転送の中断

DMA 転送は、いつでも中断して再開することもできれば、DMA 転送が終了する前に完全に無効化することもできます。

これには、2つのケースがあります。

- ストリームは、転送を無効にし、停止した場所以降再開しません。ストリームを無効化するために DMA\_SxCR レジスタの EN ビットをクリアする以外、特に行うことはありません。ストリームを無効にするのに時間を要する可能性があります（まず、進行中の転送を完了します）。転送の終了を示すために転送完了割り込みフラグ (DMA\_LISR または DMA\_HISR レジスタの TCIF) がセットされます。その時点で、DMA\_SxCR の EN ビットの値が“0”となって、ストリームが中断されます。ソフトウェアによって、ストリームが中断される前に転送されたデータ項目の数を判断できるように、DMA\_SxNDTR レジスタに、ストリームが停止した時点で残っているデータ項目の数を保持しています。
- ストリームは、DMA\_SxNDTR レジスタ内の転送されるべき残りのデータ項目数が 0 に到達する前に転送を中断します。その目的は、ストリームを再度有効にすることで後から転送を再開することにあります。転送が停止した時点から再開するにあたり、ソフトウェアが既に収集したデータ項目数を知るため DMA\_SxCR レジスタの EN ビットに書き込んで（その後それが“0”であることを確認します）ストリームを無効にした後で DMA\_SxNDTR レジスタを読み出す必要があります。その後、
  - アドレスポインタを調整するために、ペリフェラルやメモリのアドレスを更新する必要があります。
  - SxNDTR レジスタは、転送すべきデータ項目の残数（ストリームを無効にしたとき読み出される値）で更新する必要があります。
  - ストリームは、その後再度有効化して転送が停止したところから再開することができます。

**注：** *ストリームの中断による、転送の終了を示すため、転送完了割り込みフラグ (DMA\_LISR または DMA\_HISR の TCIF) がセットされることに注意してください。*

## 8.3.15 フローコントローラ

転送するデータの数を制御するものはフローコントローラとして知られています。このフローコントローラは、DMA\_SxCR レジスタの PFCTRL ビットを使用して、各ストリームごとに個別に設定されます。

フローコントローラは次のように使用することができます。

- DMA コントローラ：この場合、DMA ストリームを有効にする前に、転送するデータ項目数を DMA\_SxNDTR レジスタにソフトウェアでプログラムします。
- ペリフェラル転送元またはペリフェラル転送先：この場合、転送されるデータ項目数は不明です。最後のデータが転送される時、ペリフェラルがハードウェアによって DMA コントローラに知らせます。この機能は、転送の終了を通知できるペリフェラルでのみサポートされています。すなわち、
  - SDMMC1

特定のストリームでペリフェラルフローコントローラを使用した場合、DMA\_SxNDTR に書き込まれる値は DMA 転送に影響しません。実際に、いかなる値が書き込まれても、以下のスキームを守るためストリームが有効になるとすぐに、ハードウェアによって強制的に 0xFFFF になります。

- 想定されるストリームの中断：最終データを示すハードウェア信号（シングルまたはバースト）がペリフェラルによって送信される前に、DMA\_SxCR レジスタの EN ビットをソフトウェアによって 0 にリセットし、ストリームを停止します。このようなケースでは、“ペリフェラルからメモリ”DMA 転送の場合、ストリームが停止し、FIFO フラッシュが作動します。対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、DMA の完了を示します。DMA

転送中に転送されたデータ項目数を知るには、DMA\_SxNDTR レジスタを読み出し次の式を適用します。

$$\text{転送されたデータ数} = 0xFFFF - \text{DMA\_SxNDTR}$$

- 最終データを示すハードウェア信号の受信による通常ストリームの中断: ペリフェラルが最後の転送 (シングルまたはバースト) を要求しこの転送が完了したときストリームは自動的に中断されます。対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、DMA 転送の完了を示します。転送されたデータ項目数を知るには、DMA\_SxNDTR レジスタを読み出し上記と同じ式を適用します。
- DMA\_SxNDTR レジスタが 0 に到達: 対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、強制的な DMA 転送の完了を示します。最終データを示すハードウェア信号 (シングルまたはバースト) がまだアサートされていない場合でも、ストリームは自動的に停止します。既に転送されたデータは失われません。つまり、ペリフェラルフローコントローラモードの場合でも、1 つの DMA のトランザクションにより最大 65535 のデータ項目が管理できることとなります。

**注:** メモリ間モードに設定されているとき、DMA は常にフローコントローラとなり PFCTRL ビットはハードウェアによって強制的に 0 になります。

ペリフェラルフロー制御モードでは、サーキュラモードは禁止されています。

### 8.3.16 実現可能な DMA 設定の概要

表 31 に、実現可能なさまざまな DMA 設定をまとめています。禁止されている設定は、表内でグレイでハイライトされています。

表 31. 実現可能な DMA 設定

DMA 転送モード	転送元	転送先	フローコントローラ	サーキュラモード	転送タイプ	ダイレクトモード	ダブルバッファモード
ペリフェラルからメモリ	AHB ペリフェラルポート	AHB メモリポート	DMA	設定可能	シングル	設定可能	設定可能
					バースト	設定禁止	
			ペリフェラル	設定禁止	シングル	設定可能	設定禁止
					バースト	設定禁止	
メモリからペリフェラル	AHB メモリポート	AHB ペリフェラルポート	DMA	設定可能	シングル	設定可能	設定可能
					バースト	設定禁止	
			ペリフェラル	設定禁止	シングル	設定可能	設定禁止
					バースト	設定禁止	
メモリ間	AHB ペリフェラルポート	AHB メモリポート	DMA のみ	設定禁止	シングル	設定禁止	設定禁止
					バースト		

## 8.3.17 ストリーム設定手順

DMA ストリーム x の設定は、次の手順で行います (x はストリーム番号です)。

1. ストリームが有効になっている場合、DMA\_SxCR レジスタの EN ビットをリセットすることによって無効にし、その後、進行中のストリーム動作がないことを確認するため、このビットを読み出します。このビットに 0 を書き込んでもすぐに効果はありません。それは現在行われているすべての転送が完了した後、実際に 0 が書き込まれるからです。EN ビットが 0 として読み出される場合、ストリームが設定可能な状態であることを意味します。したがって、ストリームの設定を開始する前に EN ビットがクリアされるのを待つ必要があります。以前のデータブロックの DMA 転送からステータスレジスタ (DMA\_LISR と DMA\_HISR) に設定されたすべてのストリーム専用ビットは、ストリームを再度有効にする前にクリアする必要があります。
2. DMA\_SxPAR レジスタにペリフェラルポートレジスタアドレスを設定します。ペリフェラルイベントの後、データはこのアドレスとペリフェラルポートの間で転送されます。
3. DMA\_SxMA0R レジスタ (およびダブルバッファモードの場合は DMA\_SxMA1R レジスタ) にメモリアドレスを設定します。ペリフェラルイベント後は、このメモリに対してデータを読み書きします。
4. 転送されるデータ項目の合計数を DMA\_SxNDTR レジスタで設定します。各ペリフェラルイベントまたはバーストの各ビットの後、この値はデクリメントされます。
5. DMA\_SxCR レジスタの CHSEL[2:0] を使用して DMA チャンネル (リクエスト) を選択します。
6. ペリフェラルをフローコントローラとして使用することを意図し、かつこの機能をサポートしている場合、DMA\_SxCR レジスタの PFCTRL ビットをセットします。
7. DMA\_SxCR レジスタの PL[1:0] ビットを使用して、ストリームの優先順位を設定します。
8. FIFO の使用方法を設定します (有効または無効にする、送信と受信における閾値)。
9. DMA\_SxCR レジスタで以下の項目を設定します。データ転送方向、ペリフェラルとメモリのインクリメントモード/固定モード、シングルランザクションまたはバーストランザクション、ペリフェラルとメモリのデータ幅、サーキュラモード、ダブルバッファモード、ハーフ転送やフル転送後の割り込み、およびエラー。
10. DMA\_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

ストリームが有効になると、そのストリームに接続されているペリフェラルからの DMA リクエストを処理できます。

AHB 転送先ポート上でデータの半分が転送されると、1/2 転送フラグ (HTIF) がセットされ、1/2 転送割り込みイネーブルビット (HTIE) がセットされている場合には、割り込みが生成されます。転送が終了すると、転送完了フラグ (TCIF) がセットされ、転送完了割り込みイネーブルビット (TCIE) がセットされている場合には、割り込みが生成されます。

---

**警告 :** DMA ストリームリクエストに接続されているペリフェラルを切断するには、まず、ペリフェラルが接続されている DMA ストリームを切断し、次いで EN ビット=0 になるのを待つ必要があります。その後はじめてペリフェラルを安全に無効にすることができます。

---

## 8.3.18 エラー管理

DMA コントローラによって、次のエラーを検出することができます。

- **転送エラー**：転送エラー割り込みフラグ (TEIFx) は次の場合にセットされます。
  - DMA 読み出しましたは書き込みアクセス時にバスエラーが発生した場合
  - ダブルバッファモードでは、メモリアドレスレジスタの書き込みアクセスはソフトウェアによって要求されますが、ストリームが有効になり、メモリアドレスレジスタへの書き込みによって影響を受けるのは現在のターゲットメモリです ([セクション 8.3.9: ダブルバッファモード](#)を参照)。
- **FIFO エラー**：FIFO エラー割り込みフラグ (FEIFx) は次の場合にセットされます。
  - FIFO アンダーラン状態が検出された場合
  - FIFO オーバーラン状態が検出された場合 (リクエストと転送は DMA が内部で管理するため、このエラーはメモリ間モードでは検出されない)
  - FIFO 閾値レベルがメモリバーストのサイズに適合していないときにストリームが有効化された場合 ([表 30: FIFO 閾値設定](#)を参照)
- **ダイレクトモードエラー**：ダイレクトモードで動作して DMA\_SxCR レジスタの MINC ビットがクリアされたときのみ、“ペリフェラルからメモリ”モードにおいてダイレクトモードエラー割り込みフラグ (DMEIFx) がセットされることがあります。このフラグは、メモリバスの使用権が与えられていなかったことにより、前回のデータがまだ完全にメモリに転送されていない状態で DMA リクエストが発生したときにセットされます。このケースでは、このフラグは 2 つのデータ項目が同じ転送先アドレスに連続して転送されたことを示し、転送先がこの状況を管理できない場合に、このことが問題となる可能性があります。

ダイレクトモードでは、FIFO エラーフラグが以下の条件でセットされることもあります。

- “ペリフェラルからメモリ”モードでは、複数のペリフェラルリクエストのためにメモリバスの使用が許可されない場合、FIFO が飽和 (オーバーラン) することがあります。
- “メモリからペリフェラル”モードでは、ペリフェラルリクエストが発生する前に、メモリバスの使用が許可されていない場合、アンダーラン状態が発生する可能性があります。

バーストサイズと FIFO 閾値レベルが適合しないことによって TEIFx または FEIFx フラグがセットされた場合、欠陥のあるストリームは対応するストリーム設定レジスタ (DMA\_SxCR) の EN ビットがハードウェアによってクリアされることにより、自動的に無効化されます。

オーバーランまたはアンダーラン状態によって DMEIFx または FEIFx フラグがセットされる場合、欠陥のあるストリームは自動的に無効化されず、DMA\_SxCR レジスタの EN ビットをリセットすることによってストリームを無効化するかどうかはソフトウェアに任されています。なぜならば、この種のエラーが発生してもデータの損失がないためです。

DMA\_LISR または DMA\_HISR レジスタにあるストリームのエラー割り込みフラグ (TEIF、FEIF、DMEIF) がセットされると、DMA\_SxCR または DMA\_SxFCR レジスタの対応する割り込みイネーブルビット (TEIE、FEIE、DMIE) がセットされていれば、割り込みが発生します。

**注：** *FIFO オーバーランまたはアンダーラン状態が発生した場合、オーバーランまたはアンダーラン状態がクリアされるまではペリフェラルリクエストがストリームによって承認されないため、データが失われることはありません。この承認に時間がかかりすぎた場合、ペリフェラル自体がその内部バッファのオーバーランまたはアンダーラン状態を検出し、データが失われる可能性があります。*

## 8.4 DMA 割り込み

各 DMA ストリームに対し、次のイベントにて割り込みを生成することができます。

- 1/2 転送終了
- 転送完了
- 転送エラー
- FIFO エラー (オーバーラン、アンダーラン、または FIFO レベルのエラー)
- ダイレクトモードエラー

柔軟性を持たせるため、表 32 に示すよう個別の割り込みイネーブル制御ビットが使用できます。

表 32. DMA 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
1/2 転送	HTIF	HTIE
転送完了	TCIF	TCIE
転送エラー	TEIF	TEIE
FIFO オーバーラン/アンダーラン	FEIF	FEIE
ダイレクトモードエラー	DMEIF	DMEIE

**注：** イネーブル制御ビットを '1' に設定する前に、対応するイベントフラグをクリアする必要があります。さもないと直ちに割り込みが生成されてしまいます。

## 8.5 DMA レジスタ

DMA レジスタには、ワード (32 ビット) 単位でアクセスすることができます。

### 8.5.1 DMA ロー割り込みステータスレジスタ (DMA\_LISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2
r	r	r	r	r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Res.	FEIF0
r	r	r	r	r	r	r	r		r	r	r	r	r		r

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **TCIFx** : ストリーム x 転送完了割り込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA\_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送完了イベントは発生していません。

1 : ストリーム x に転送完了イベントが発生しました。



ビット 26、20、10、4 **HTIFx** : ストリーム x 1/2 転送割り込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA\_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に 1/2 転送イベントは発生していません。  
1 : ストリーム x に 1/2 転送イベントが発生しました。

ビット 25、19、9、3 **TEIFx** : ストリーム x 転送エラー割り込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA\_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送エラーは発生していません。  
1 : ストリーム x に転送エラーが発生しました。

ビット 24、18、8、2 **DMEIFx** : ストリーム x ダイレクトモードエラー割り込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA\_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x にダイレクトモードエラーは発生していません。  
1 : ストリーム x にダイレクトモードエラーが発生しました。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **FEIFx** : ストリーム x FIFO エラー割り込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA\_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に FIFO エラーイベントは発生していません。  
1 : ストリーム x に FIFO エラーイベントが発生しました。

## 8.5.2 DMA ハイ割り込みステータスレジスタ (DMA\_HISR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res.	FEIF6
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Res.	FEIF4
				r	r	r	r		r	r	r	r	r		r

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **TCIFx** : ストリーム x 転送完了割り込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA\_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送完了イベントは発生していません。  
1 : ストリーム x に転送完了イベントが発生しました。

ビット 26、20、10、4 **HTIFx** : ストリーム x 1/2 転送割り込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA\_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に 1/2 転送イベントは発生していません。  
1 : ストリーム x に 1/2 転送イベントが発生しました。

ビット 25、19、9、3 **TEIFx** : ストリーム x 転送エラー割り込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA\_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

- 0 : ストリーム x に転送エラーは発生していません。
- 1 : ストリーム x に転送エラーが発生しました。

ビット 24、18、8、2 **DMEIFx** : ストリーム x ダイレクトモードエラー割り込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA\_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

- 0 : ストリーム x にダイレクトモードエラーは発生していません。
- 1 : ストリーム x にダイレクトモードエラーが発生しました。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **FEIFx** : ストリーム x FIFO エラー割り込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA\_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

- 0 : ストリーム x に FIFO エラーイベントは発生していません。
- 1 : ストリーム x に FIFO エラーイベントが発生しました。

### 8.5.3 DMA ロー割り込みフラグクリアレジスタ (DMA\_LIFCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Res.	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res.	CFEIF2
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Res.	CFEIF0
				w	w	w	w		w	w	w	w	w		w

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **CTCIFx** : ストリーム x クリア転送完了割り込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA\_LISR レジスタの対応する TCIFx フラグをクリアします。

ビット 26、20、10、4 **CHTIFx** : ストリーム x クリア 1/2 転送割り込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA\_LISR レジスタの対応する HTIFx フラグをクリアします。

ビット 25、19、9、3 **CTEIFx** : ストリーム x クリア転送エラー割り込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA\_LISR レジスタの対応する TEIFx フラグをクリアします。

ビット 24、18、8、2 **CDMEIFx** : ストリーム x クリアダイレクトモードエラー割り込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA\_LISR レジスタの対応する DMEIFx フラグをクリアします。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **CFEIFx** : ストリーム x クリア FIFO エラー割り込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA\_LISR レジスタの対応する CFEIFx フラグをクリアします。



## 8.5.4 DMA ハイ割り込みフラグクリアレジスタ (DMA\_HIFCR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Res.	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res.	CFEIF6
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res.	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Res.	CFEIF4
				w	w	w	w		w	w	w	w	w		w

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **CTCIFx** : ストリーム x クリア転送完了割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA\_HISR レジスタの対応する TCIFx フラグをクリアします。

ビット 26、20、10、4 **CHTIFx** : ストリーム x クリア 1/2 転送割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA\_HISR レジスタの対応する HTIFx フラグをクリアします。

ビット 25、19、9、3 **CTEIFx** : ストリーム x クリア転送エラー割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA\_HISR レジスタの対応する TEIFx フラグをクリアします。

ビット 24、18、8、2 **CDMEIFx** : ストリーム x クリアダイレクトモードエラー割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA\_HISR レジスタの対応する DMEIFx フラグをクリアします。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **CFEIFx** : ストリーム x クリア FIFO エラー割り込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA\_HISR レジスタの対応する CFEIFx フラグをクリアします。

## 8.5.5 DMA ストリーム x 設定レジスタ (DMA\_SxCR) (x=0~7)

このレジスタは当該ストリームの設定に使用されます。

アドレスオフセット :  $0x10 + 0x18 \times \text{ストリーム番号}$

リセット値 :  $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CHSEL[3:0]			MBURST [1:0]		PBURST[1:0]		Res.	CT	DBM または予約済み	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw または r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:25 **CHSEL[2:0]** : チャネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。

000 : チャネル 0 が選択されます。

001 : チャネル 1 が選択されます。

010 : チャネル 2 が選択されます。

011 : チャネル 3 が選択されます。

100 : チャネル 4 が選択されます。

101 : チャネル 5 が選択されます。

110 : チャネル 6 が選択されます。

111 : チャネル 7 が選択されます。

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ビット 24:23 **MBURST** : メモリバースト転送設定

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : シングル転送

01 : INCR4 (4 ビートのインクリメンタルバースト)

10 : INCR8 (8 ビートのインクリメンタルバースト)

11 : INCR16 (16 ビートのインクリメンタルバースト)

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ダイレクトモードでは、これらのビットは EN ビット=“1”になり次第ハードウェアによって強制的に 0x0 になります。

ビット 22:21 **PBURST[1:0]** : ペリフェラルバースト転送設定

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : シングル転送

01 : INCR4 (4 ビートのインクリメンタルバースト)

10 : INCR8 (8 ビートのインクリメンタルバースト)

11 : INCR16 (16 ビートのインクリメンタルバースト)

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ダイレクトモードでは、これらのビットはハードウェアによって強制的に 0x0 になります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

### ビット 19 **CT** : 現在のターゲット (ダブルバッファモード時のみ)

このビットは、ハードウェアによってセット/クリアされます。また、ソフトウェアによって書き込むこともできます。

0 : 現在のターゲットメモリはメモリ 0 (DMA\_SxM0AR ポインタによってアドレス指定) です。

1 : 現在のターゲットメモリはメモリ 1 (DMA\_SxM1AR ポインタによってアドレス指定) です。

このビットは、EN が '0' のときのみ書き込むことができ、最初の転送の対象メモリ領域を示します。ストリームが有効になると、このビットはステータスフラグとして動作し、どのメモリ領域が現在のターゲットかを示します。

### ビット 18 **DBM** : ダブルバッファモード

このビットは、ソフトウェアによってセット/クリアされます。

0 : 転送終了時にバッファを切り替えません

1 : DMA 転送終了時にメモリターゲットを切り替えます

このビットは保護されており、EN が "0" のときのみ書き込みできます。

### ビット 17:16 **PL[1:0]** : 優先順位レベル

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : 低

01 : 中

10 : 高

11 : 最優先

これらのビットは保護されており EN が "0" のときのみ書き込みできます。

### ビット 15 **PINCOS** : ペリフェラルインクリメントオフセットサイズ

このビットは、ソフトウェアによってセット/クリアされます。

0 : ペリフェラルアドレス計算のためのオフセットサイズは PSIZE と関連付けされています。

1 : ペリフェラルアドレス計算のためのオフセットサイズは 4 に固定されています (32 ビットアライメント)。

ビット PINC="0" の場合、このビットは意味を持ちません。

このビットは保護されており EN="0" のときのみ書き込みできます。

ダイレクトモードが選択されているか、PBURSTが "00" ではない場合、ストリームが有効になったとき (EN ビット="1") このビットはハードウェアによって強制的にローになります。

### ビット 14:13 **MSIZE[1:0]** : メモリデータサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : バイト (8 ビット)

01 : ハーフワード (16 ビット)

10 : ワード (32 ビット)

11 : 予約済み

これらのビットは保護されており EN が "0" のときのみ書き込みできます。

ダイレクトモードでは、EN ビット="1" になり次第、MSIZE はハードウェアによって強制的に PSIZE と同じ値になります。

### ビット 12:11 **PSIZE[1:0]** : ペリフェラルデータサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : バイト (8 ビット)

01 : ハーフワード (16 ビット)

10 : ワード (32 ビット)

11 : 予約済み

これらのビットは保護されており EN が "0" の場合のみ書き込みできます。

### ビット 10 **MINC** : メモリインクリメントモード

このビットは、ソフトウェアによってセット/クリアされます。

0 : メモリアドレスポインタは固定されています。

1 : メモリアドレスポインタは、各データ転送後にインクリメントされます (MSIZE に応じてインクリメント)

このビットは保護されており、EN が "0" のときのみ書き込みできます。

**ビット 9 PINC** : ペリフェラルインクリメントモード

このビットは、ソフトウェアによってセット/クリアされます。

0 : ペリフェラルアドレスポインタは固定されています

1 : ペリフェラルアドレスポインタは、各データ転送後にインクリメントされます (PSIZE に応じてインクリメント)

このビットは保護されており、EN が“0”のときのみ書き込みできます。

**ビット 8 CIRC** : サーキュラモード

このビットは、ソフトウェアによってセット/クリアされ、かつハードウェアによってクリアできません。

0 : サーキュラモードは無効です。

1 : サーキュラモードは有効です。

ペリフェラルがフローコントローラ (PFCTRL ビット =1) となっていて、ストリームが有効なとき (ビット EN=1)、このビットは自動的にハードウェアによって 0 になります。

DBM ビットがセットされている場合、ストリームが有効になり次第 (EN ビット =“1”) ハードウェアにより強制的に 1 になります。

**ビット 7:6 DIR[1:0]** : データ転送方向

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : ペリフェラルからメモリ

01 : メモリからペリフェラル

10 : メモリ間

11 : 予約済み

これらのビットは保護されており EN が“0”のときのみ書き込みできます。

**ビット 5 PFCTRL** : ペリフェラルフローコントローラ

このビットは、ソフトウェアによってセット/クリアされます。

0 : DMA がフローコントローラ

1 : ペリフェラルがフローコントローラ

このビットは保護されており、EN が“0”のときのみ書き込みできます。

メモリ間モードが選択されている場合 (DIR[1:0] ビット =10)、このビットは自動的にハードウェアにより強制的に 0 になります。

**ビット 4 TCIE** : 転送完了割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : TC 割り込みは無効です。

1 : TC 割り込みは有効です。

**ビット 3 HTIE** : 1/2 転送割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : HT 割り込みは無効です。

1 : HT 割り込みは有効です。

**ビット 2 TEIE** : 転送エラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : TE 割り込みは無効です。

1 : TE 割り込みは有効です。

**ビット 1 DMEIE** : ダイレクトモードエラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DME 割り込みは無効です。

1 : DME 割り込みは有効です。

ビット 0 EN : ストリームイネーブル/ロー読み出し時のストリームレディフラグ

このビットは、ソフトウェアによってセット/クリアされます。

0 : ストリームは無効です。

1 : ストリームは有効です。

このビットは、次のときハードウェアによってクリアできます。

- DMA 転送終了時 (ストリームが設定準備完了)
- AHB マスタバス上で転送エラーが発生した場合
- メモリ AHB ポート上の FIFO 閾値がバーストの大きさと適合しない場合

このビットが 0 として読み出された場合、ソフトウェアによって設定レジスタと FIFO ビットレジスタをプログラムすることができます。EN ビットが 1 として読み出された場合、これらのレジスタの書き込みは禁止されています。

**注:** 新しい転送を開始するために EN ビットを“1”に設定する前に、DMA\_LISR または DMA\_HISR レジスタのストリームに対応するイベントフラグをクリアする必要があります。

## 8.5.6 DMA ストリーム x データ数レジスタ (DMA\_HIFCR) (x=0~7)

アドレスオフセット :  $0x14 + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 NDT[15:0] : 転送するデータ項目の数

転送されるデータ項目数 (0~65535)。このレジスタに書き込めるのは、ストリームが無効のときだけです。ストリームが有効になると、このレジスタは読み出し専用になり、送信されるべき残りのデータ項目を示します。このレジスタは、各 DMA 転送後にデクリメントされます。

転送が完了すると、このレジスタは、ゼロのままにすることもできれば (ストリームが通常モードのとき)、次の場合には事前にプログラムされた値を自動的にリロードすることもできます。

- ストリームがサーキュラモードに設定されている場合
- EN ビットを“1”に設定することにより、ストリームが再び有効になった場合

このレジスタの値が 0 の場合、ストリームが有効であっても、トランザクションは処理できません。

## 8.5.7 DMA ストリーム x ペリフェラルアドレスレジスタ (DMA\_SxPAR) (x=0~7)

アドレスオフセット :  $0x18 + 0x18 \times \text{ストリーム番号}$

リセット値 :  $0x0000\ 0000$

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

ビット 31:0 **PAR[31:0]** : ペリフェラルアドレス

データの読み出し/書き込みが行われるペリフェラルデータレジスタのベースアドレス。

これらのビットは書き込み保護されており、DMA\_SxCR レジスタの EN ビット = "0" のときのみ書き込みできます。

## 8.5.8 DMA ストリーム x メモリ 0 アドレスレジスタ (DMA\_SxM0AR) (x=0~7)

アドレスオフセット :  $0x1C + 0x18 \times \text{ストリーム番号}$

リセット値 :  $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **M0A[31:0]** : メモリ 0 アドレス

データの読み出し/書き込みが行われるメモリ領域 0 のベースアドレス。

これらのビットは書き込み保護されています。次の場合にのみ書き込みが可能です。

- ストリームが無効 (DMA\_SxCR レジスタの EN = "0") の場合、または
- ストリームが有効 (DMA\_SxCR レジスタの EN = "1") で DMA\_SxCR レジスタのビット CT="1" (ダブルバッファモード) の場合。

## 8.5.9 DMA ストリーム x メモリ 1 アドレスレジスタ (DMA\_SxM1AR) (x=0~7)

アドレスオフセット :  $0x20 + 0x18 \times \text{ストリーム番号}$

リセット値 :  $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M1A[31:16]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1A[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **M1A[31:0]** : メモリ 1 アドレス (ダブルバッファモードの場合に使用)

データの読み出し/書き込みが行われるメモリ領域 1 のベースアドレス。

このレジスタは、ダブルバッファモードでのみ使用されます。

これらのビットは書き込み保護されています。次の場合にのみ書き込みが可能です。

- ストリームが無効 (DMA\_SxCR レジスタの EN = "0") の場合、または
- ストリームが有効 (DMA\_SxCR レジスタの EN = "1") で DMA\_SxCR レジスタのビット CT="0" の場合。

## 8.5.10 DMA ストリーム x FIFO 制御レジスタ (DMA\_SxFCR) (x=0~7)

アドレスオフセット :  $0x24 + 0x24 \times \text{ストリーム番号}$

リセット値 :  $0x0000\ 0021$

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **FEIE** : FIFO エラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : FE 割り込みは無効です。

1 : FE 割り込みは有効です。

ビット 6 予約済みであり、リセット値に保持する必要があります。

**ビット 5:3 FS[2:0] : FIFO ステータス**

これらのビットは読み出し専用です。

000 : 0 < FIFO レベル < 1/4

001 : 1/4 ≤ FIFO レベル < 1/2

010 : 1/2 ≤ FIFO レベル < 3/4

011 : 3/4 ≤ FIFO レベル < 満杯

100 : FIFO は空です。

101 : FIFO は満杯です。

その他 : 意味を持ちません。

これらのビットは、ダイレクトモードには無関係です (DMDIS ビットがゼロ)。

**ビット 2 DMDIS : ダイレクトモードディセーブル**

このビットは、ソフトウェアによってセット/クリアされます。ハードウェアによってセットすることができます。

0 : ダイレクトモードは有効です。

1 : ダイレクトモードは無効です。

このビットは保護されており、EN が“0”のときのみ書き込みできます。

このビットは、メモリ間モードが選択されており (DMA\_SxCR の DIR ビットが“10”)、DMA\_SxCR レジスタの EN ビットが“1”の場合、ハードウェアによってセットされます。その理由は、ダイレクトモードではメモリ間転送設定が使用できないからです。

**ビット 1:0 FTH[1:0] : FIFO 閾値選択**

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : FIFO 満杯の 1/4

01 : FIFO 満杯の 1/2

10 : FIFO 満杯の 3/4

11 : FIFO 満杯

これらのビットは、DMIS 値がゼロのときダイレクトモードで使用されません。

これらのビットは保護されており EN が“0”のときのみ書き込みできます。

## 8.5.11 DMA レジスタマップ

表 33 に DMA レジスタの要約を示します。

表 33. DMA レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	DMA_LISR	Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2	Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	平均読み込み	FEIF0
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	0
0x0004	DMA_HISR	Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res.	FEIF6	Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	平均読み込み	FEIF4
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	
0x0008	DMA_LIFCR	Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	CTEIF3	CDMEIF3	平均読み込み	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res.	CFEIF2	Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	平均読み込み	CFEIF0
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	
0x000C	DMA_HIFCR	Res.	Res.	Res.	Res.	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	平均読み込み	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res.	CFEIF6	Res.	Res.	Res.	Res.	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res.	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	平均読み込み	CFEIF4
	リセット値					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0	0	
0x0010	DMA_S0CR	Res.	Res.	Res.	Res.	CHSEL[2:0]				MBURST[1:0]	PBURST[1:0]		Res.	CT	DBM	PL[1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]	PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN		
	リセット値					0	0	0		0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0014	DMA_S0NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x0018	DMA_S0PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	DMA_S0M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	DMA_S0M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	DMA_S0FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH		
	リセット値																									0		1	0	0	0	1	
0x0028	DMA_S1CR	Res.	Res.	Res.	Res.	CHSEL [2:0]				MBURST[1:1]	PBURST[1:0]		Res.	CT	DBM	PL[1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]	PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN		
	リセット値					0	0	0		0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x002C	DMA_S1NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x0030	DMA_S1PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 33. DMA レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0034	DMA_S1M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0038	DMA_S1M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	DMA_S1FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH [1:0]		
	リセット値																									0		1	0	0	0	0	1
0x0040	DMA_S2CR	Res.	Res.	Res.	Res.	CHSEL [2:0]		Res.	MBURST[1:0]		PBURST[1:0]		Res.	CT	DBM	PL [1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]	PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN		
	リセット値					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0044	DMA_S2NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x0048	DMA_S2PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	DMA_S2M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	DMA_S2M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	DMA_S2FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH [1:0]		
	リセット値																									0		1	0	0	0	0	1
0x0058	DMA_S3CR	Res.	Res.	Res.	Res.	CHSEL[2:0]		Res.	MBURST[1:0]		PBURST[1:0]		Res.	CT	DBM	PL [1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]	PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN		
	リセット値					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x005C	DMA_S3NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x0060	DMA_S3PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	DMA_S3M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0068	DMA_S3M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x006C	DMA_S3FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS	FTH [1:0]		
	リセット値																									0		1	0	0	0	0	1



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

表 33. DMA レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0070	DMA_S4CR	Res.	Res.	Res.	Res.	Res.	CHSEL[2:0]	Res.	Res.	MBURST[1:0]	Res.	Res.	Res.	CT	Res.	DBM	PL[1:0]	PINCOS	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR [1:0]	PFCtrl	TCIE	HTIE	TEIE	DMEIE	EN			
	リセット値					0	0	0		0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0074	DMA_S4NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x0078	DMA_S4PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x007C	DMA_S4MOAR	MOA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0080	DMA_S4M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	DMA_S4FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]	Res.	DMDIS	FTH [1:0]			
	リセット値																								0		1	0	0	0	0	1	
0x0088	DMA_S5CR	Res.	Res.	Res.	Res.	Res.	CHSEL[2:0]	Res.	Res.	MBURST[1:0]	Res.	Res.	Res.	CT	Res.	DBM	PL[1:0]	PINCOS	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR[1:0]	PFCtrl	TCIE	HTIE	TEIE	DMEIE	EN			
	リセット値					0	0	0		0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x008C	DMA_S5NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x0090	DMA_S5PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094	DMA_S5MOAR	MOA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	DMA_S5M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x009C	DMA_S5FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]	Res.	DMDIS	FTH [1:0]			
	リセット値																								0		1	0	0	0	0	1	
0x00A0	DMA_S6CR	Res.	Res.	Res.	Res.	Res.	CHSEL[2:0]	Res.	Res.	MBURST[1:0]	Res.	Res.	Res.	CT	Res.	DBM	PL[1:0]	PINCOS	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR[1:0]	PFCtrl	TCIE	HTIE	TEIE	DMEIE	EN			
	リセット値					0	0	0		0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00A4	DMA_S6NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x00A8	DMA_S6PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00AC	DMA_S6MOAR	MOA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	DMA_S6M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B4	DMA_S6FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]	Res.	DMDIS	FTH [1:0]			
	リセット値																								0		1	0	0	0	0	1	

表 33. DMA レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00B8	DMA_S7CR	Res.	Res.	Res.	Res.	Res.	CHSEL[2:0]	Res.	Res.	MBURST[1:0]	Res.	PEURST[1:0]	Res.	CT	DBM	PL[1:0]	PINCOS	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR[1:0]	PFCtrl	TCIE	HTIE	TEIE	DMEIE	EN				
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00BC	DMA_S7NDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C0	DMA_S7PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	DMA_S7M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8	DMA_S7M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00CC	DMA_S7FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEIE	Res.	FS[2:0]		DMDIS		FTH		
	リセット値																								0		1	0	0	0	0	1	

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 9 Chrom-Art Accelerator™ コントローラ (DMA2D)

### 9.1 DMA2D の概要

Chrom-Art Accelerator™ (DMA2D) はイメージ操作専用の特殊な DMA です。以下の操作を実行できます。

- 転送先イメージの一部または全部を特定の色で塗りつぶす
- 転送元イメージの一部または全部を、転送先イメージの一部または全部にコピーする
- 転送元イメージの一部または全部を、ピクセルフォーマット変換を行って転送先イメージの一部または全部にコピーする
- 異なるピクセルフォーマットを持つ転送元イメージの全部または一部をブレンドし、その結果を別のピクセルフォーマットで転送先イメージの一部または全体にコピーする

すべての標準的なピクセルフォーマットがサポートされており、4bit/pixel から 32bit/pixel のインデックスカラーモードまたはダイレクトカラーモードに対応しています。DMA2D は CLUT (color look-up tables : カラールックアップテーブル) 用に専用メモリを搭載しています。

## 9.2 DMA2D の主な機能

DMA2D の主な機能は次のとおりです。

- シングル AHB マスタバスアーキテクチャ。
- 8/16/32 ビットアクセスをサポートする AHB スレーブプログラミングインタフェース (32 ビットの CLUT アクセスを除く)。
- ユーザが任意に設定可能なワーク領域サイズ
- ユーザが任意に設定可能な転送元および転送先領域のオフセット
- 全メモリ空間でユーザが任意に設定可能な転送元および転送先のアドレス
- 最大 2 つの転送元のブレンド操作
- アルファ値を調整可能 (ソース値、固定値、または変調値)
- ユーザが任意に設定可能な転送元および転送先のカラーフォーマット
- インダイレクトカラーまたはダイレクトカラーで 4bit/pixel から 32bit/pixel まで、最大 11 のカラーフォーマットをサポート。
- インダイレクトカラーモードにおける CLUT 格納用に 2 つの内部メモリを搭載
- 自動 CLUT ローディングまたは CPU を使用した CLUT プログラミング
- ユーザが任意にプログラム可能な CLUT サイズ
- AHB 帯域幅を制御する内部タイマ
- 4 つの動作モード: レジスタからメモリへの転送、メモリ間転送、ピクセルフォーマット変換を伴うメモリ間転送、およびピクセルフォーマット変換およびブレンディングを伴うメモリ間転送
- 固定カラーでの矩形領域塗りつぶし
- 領域間のコピー
- 転送元イメージと転送先イメージ間のピクセルフォーマット変換を伴うコピー
- ピクセルフォーマットの異なる 2 つの転送元のコピー、およびブレンディング
- DMA2D 動作のアボートおよびサスペンド
- ユーザが任意に設定可能な転送先ラインにおけるウォーターマーク割り込み
- バスエラーまたはアクセス競合時の割り込み生成
- プロセス完了時の割り込み生成

## 9.3 DMA2D の機能詳細

### 9.3.1 概要

DMA2D コントローラはダイレクトメモリ転送を行います。AHB マスタとして、AHB バスマトリックスを制御して AHB トランザクションを開始することができます。

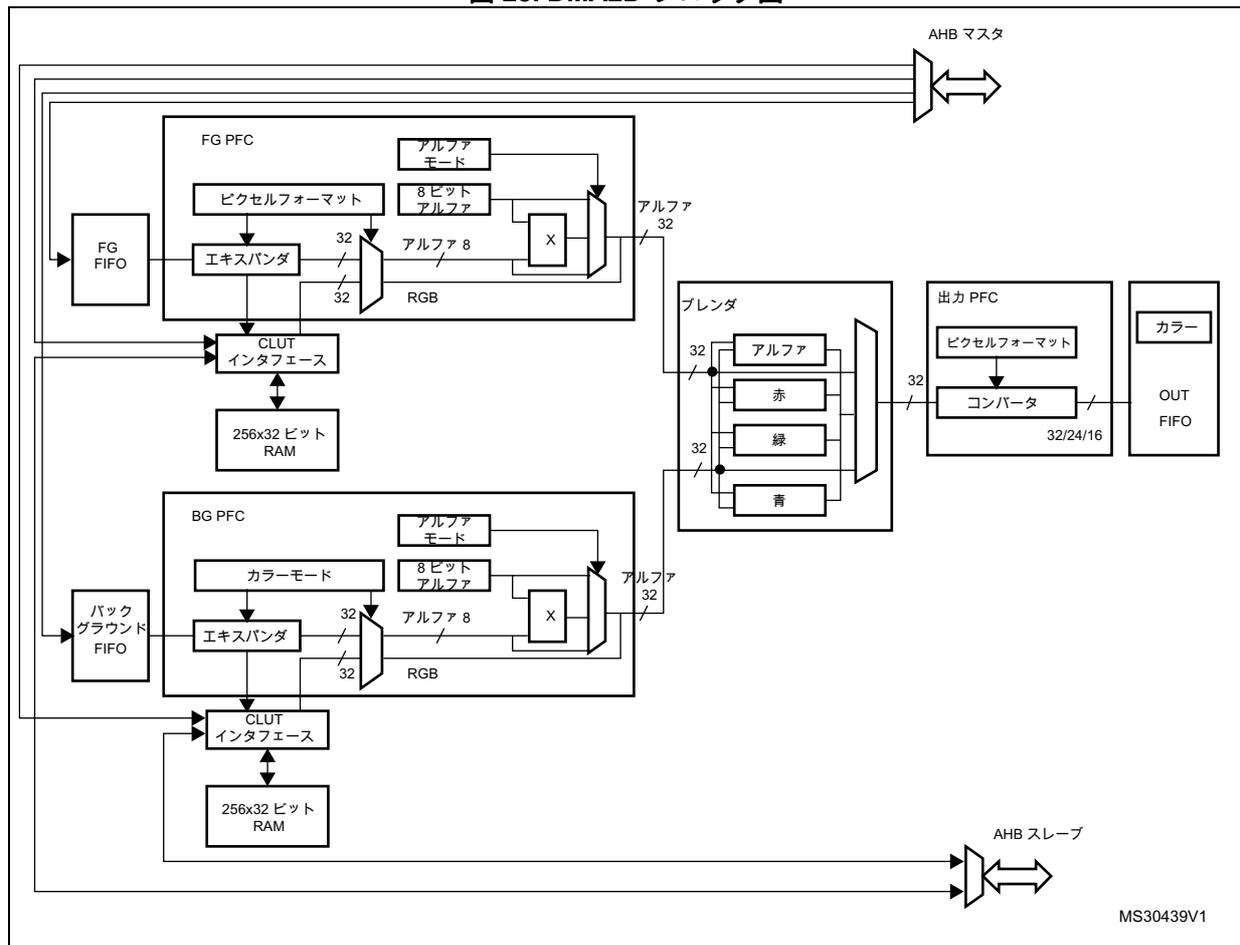
DMA2D は、次のモードで動作することができます。

- レジスタからメモリへの転送
- メモリ間転送
- ピクセルフォーマット変換を伴うメモリ間転送
- ピクセルフォーマット変換およびブレンディングを伴うメモリ間転送

AHB スレーブポートは、DMA2D コントローラをプログラムするために使用されます。

DMA2D インタフェースのブロック図を [図 28 : DMA2D ブロック図](#) に示します。

図 28. DMA2D ブロック図



### 9.3.2 DMA2D 制御

DMA2D コントローラは、DMA2D 制御レジスタ (DMA2D\_CR) を使用して設定します。このレジスタによって次の選択を行うことができます。

ユーザアプリケーションで次の操作を実行できます。

- 動作モードの選択
- DMA2D 割り込みの有効化/無効化
- 進行中のデータ転送の開始/サスペンド/アボート

### 9.3.3 DMA2D フォアグラウンドおよびバックグラウンド FIFO

DMA2D フォアグラウンド (FG) FIFO およびバックグラウンド (BG) FIFO はコピーなどの処理をする入力データをフェッチします。

これらの FIFO は、それぞれのピクセルフォーマットコンバータ (PFC) に定義されているピクセルフォーマットに従ってピクセルデータをフェッチします。

これらは次の制御レジスタにより設定可能です。

- DMA2D フォアグラウンドメモリアドレスレジスタ (DMA2D\_FGMAR)
- DMA2D フォアグラウンドオフセットレジスタ (DMA2D\_FGOR)
- DMA2D バックグラウンドメモリアドレスレジスタ (DMA2D\_BGMAR)
- DMA2D バックグラウンドオフセットレジスタ (DMA2D\_BGOR)
- DMA2D ライン数レジスタ (ライン数およびラインあたりのピクセル数) (DMA2D\_NLR)

DMA2D がレジスタからメモリへの転送モードで動作する場合、どちらの FIFO もアクティブになりません。

DMA2D がメモリ間転送モードで動作する (ピクセルフォーマット変換、ブレンディング処理のいずれも行わない) 場合は、FG FIFO のみアクティブになりバッファとして機能します。

DMA2D がピクセルフォーマット変換を伴うメモリ間転送モードで動作する (ブレンディング処理を行わない) 場合は、BG FIFO はアクティブになりません。

### 9.3.4 DMA2D フォアグラウンドおよびバックグラウンドのピクセルフォーマットコンバータ (PFC)

DMA2D フォアグラウンドピクセルフォーマットコンバータ (PFC) およびバックグラウンドピクセルフォーマットコンバータはピクセルフォーマット変換を行って 32bit/pixel のデータを生成します。PFC はアルファ値に調整を加えることもできます。

コンバータの一段目ではピクセルフォーマットを変換します。フォアグラウンドピクセルとバックグラウンドピクセルの元のピクセルフォーマットは、それぞれ DMA2D\_FGPFCCR および DMA2D\_BGPFCCR の CM[3:0] ビットによって設定します。

サポートされている入力フォーマットを [表 34: 入力でサポートされているカラーモード](#) に示します。

表 34. 入力でサポートされているカラーモード

CM[3:0]	カラーモード
0000	ARGB8888
0001	RGB888
0010	RGB565
0011	ARGB1555
0100	ARGB4444
0101	L8
0110	AL44
0111	AL88
1000	L4
1001	A8
1010	A4

ピクセルフォーマットは次のとおりコード化されています。

- アルファ値フィールド：透明度  
0xFF 値は不透明ピクセルに対応し 0x00 は透明ピクセルに対応します。
- R フィールドは赤
- G フィールドは緑
- B フィールドは青
- L フィールド：輝度  
このフィールドは RGB から 3 つの、ARGB から 4 つの成分を読み出すための CLUT のインデックスです。

元のフォーマットがダイレクトカラーモードの場合、MSB を LSB へコピーしてチャンネルあたり 8 ビットへの拡張が行われます。これにより変換の線形性が完全に保証されます。

元のフォーマットにアルファ値が無い場合、アルファ値は自動的に 0xFF (不透明) にセットされます。

元のフォーマットがインダイレクトカラーモードの場合、CLUT が必要となり、各ピクセルフォーマットコンバータは、256 エントリの 32 ビット CLUT と連動します。

特定のアルファモード A4 および A8 では、カラー情報の格納、インデックスの利用のいずれも行われません。イメージ生成に使用されるカラーは固定となっており、フォアグラウンドピクセル用のカラーは DMA2D\_FGCOLR レジスタで、バックグラウンドピクセル用のカラーは DMA2D\_BGCOLOR レジスタで定義されます。

システムメモリ内のフィールドの順序は表 35：メモリ内のデータ順序で定義しています。

表 35. メモリ内のデータ順序

カラーモード	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A <sub>0</sub> [7:0]	R <sub>0</sub> [7:0]	G <sub>0</sub> [7:0]	B <sub>0</sub> [7:0]
RGB888	B <sub>1</sub> [7:0]	R <sub>0</sub> [7:0]	G <sub>0</sub> [7:0]	B <sub>0</sub> [7:0]
	G <sub>2</sub> [7:0]	B <sub>2</sub> [7:0]	R <sub>1</sub> [7:0]	G <sub>1</sub> [7:0]
	R <sub>3</sub> [7:0]	G <sub>3</sub> [7:0]	B <sub>3</sub> [7:0]	R <sub>2</sub> [7:0]
RGB565	R <sub>1</sub> [4:0]G <sub>1</sub> [5:3]	G <sub>1</sub> [2:0]B <sub>1</sub> [4:0]	R <sub>0</sub> [4:0]G <sub>0</sub> [5:3]	G <sub>0</sub> [2:0]B <sub>0</sub> [4:0]
ARGB1555	A <sub>1</sub> [0]R <sub>1</sub> [4:0]G <sub>1</sub> [4:3]	G <sub>1</sub> [2:0]B <sub>1</sub> [4:0]	A <sub>0</sub> [0]R <sub>0</sub> [4:0]G <sub>0</sub> [4:3]	G <sub>0</sub> [2:0]B <sub>0</sub> [4:0]
ARGB4444	A <sub>1</sub> [3:0]R <sub>1</sub> [3:0]	G <sub>1</sub> [3:0]B <sub>1</sub> [3:0]	A <sub>0</sub> [3:0]R <sub>0</sub> [3:0]	G <sub>0</sub> [3:0]B <sub>0</sub> [3:0]
L8	L <sub>3</sub> [7:0]	L <sub>2</sub> [7:0]	L <sub>1</sub> [7:0]	L <sub>0</sub> [7:0]
AL44	A <sub>3</sub> [3:0]L <sub>3</sub> [3:0]	A <sub>2</sub> [3:0]L <sub>2</sub> [3:0]	A <sub>1</sub> [3:0]L <sub>1</sub> [3:0]	A <sub>0</sub> [3:0]L <sub>0</sub> [3:0]
AL88	A <sub>1</sub> [7:0]	L <sub>1</sub> [7:0]	A <sub>0</sub> [7:0]	L <sub>0</sub> [7:0]
L4	L <sub>7</sub> [3:0]L <sub>6</sub> [3:0]	L <sub>5</sub> [3:0]L <sub>4</sub> [3:0]	L <sub>3</sub> [3:0]L <sub>2</sub> [3:0]	L <sub>1</sub> [3:0]L <sub>0</sub> [3:0]
A8	A <sub>3</sub> [7:0]	A <sub>2</sub> [7:0]	A <sub>1</sub> [7:0]	A <sub>0</sub> [7:0]
A4	A <sub>7</sub> [3:0]A <sub>6</sub> [3:0]	A <sub>5</sub> [3:0]A <sub>4</sub> [3:0]	A <sub>3</sub> [3:0]A <sub>2</sub> [3:0]	A <sub>1</sub> [3:0]A <sub>0</sub> [3:0]

32 ビットに調整された 24 ビット RGB888 は ARGB8888 モードでサポートされます。

32 ビット値が生成されると、表 36：アルファモード設定に示すように DMA2D\_FGPFCCR/DMA2D\_BGPFCCR レジスタの AM[1:0] フィールドに従ってアルファチャンネルを調整することができます。

この時アルファチャネルは以下が可能です。

- そのままを維持 (調整なし)
- DMA2D\_FGPFCCR/DMA2D\_BGPFCCR の ALPHA[7:0] 値で置換
- もしくは、DMA2D\_FGPFCCR/DMA2D\_BGPFCCR の ALPHA[7:0] 値を 255 で除算した後、元のアルファ値に乗算した値で置換

表 36. アルファモード設定

AM[1:0]	アルファモード
00	調整なし
01	DMA2D_xxPFCCR の値で置換
10	DMA2D_xxPFCCR を 255 で除算した値を元の値に乗算して置換
11	予約済み

### 9.3.5 DMA2D フォアグラウンドおよびバックグラウンド CLUT インタフェース

CLUT メモリアクセスおよび CLUT の自動ローディングは CLUT インタフェースによって管理されます。

次の 3 種類のアクセスが可能です。

- ピクセルフォーマット変換操作中の PFC による CLUT の読み出し
- CPU が CLUT に対してデータの読み出しまたは書き込みを行っているときの AHB スレーブポートを経由した CLUT へのアクセス
- CLUT の自動ローディングが行われているときの AHB マスタポートを経由した CLUT への書き込み

CLUT メモリローディングは次の 2 つの方法で行うことができます。

- オートローディング  
CLUT のローディングは、次の順序で行う必要があります。
  - a) DMA2D\_FGCMAR レジスタ (フォアグラウンド CLUT) または DMA2D\_BGCMAR レジスタ (バックグラウンド CLUT) に CLUT アドレスを設定します。
  - b) DMA2D\_FGPFCCR レジスタ (フォアグラウンド CLUT) または DMA2D\_BGPFCCR レジスタ (バックグラウンド CLUT) の CS[7:0] フィールドに CLUT サイズを設定します。
  - c) DMA2D\_FGPFCCR レジスタ (フォアグラウンド CLUT) または DMA2D\_BGPFCCR レジスタ (バックグラウンド CLUT) の START ビットをセットして転送を開始します。このオートローディング処理中、CLUT に対する CPU によるアクセスはできません。競合が発生した場合、CLUT アクセスエラー割り込みが発生します。ただし、DMA2D\_CR 内の CAEIE ビットが“1”に設定されている必要があります。
- マニュアルローディング  
ローカル CLUT メモリがマップされている DMA2D AHB スレーブポートを使用して、アプリケーションがマニュアルで CLUT を設定する必要があります。フォアグラウンド CLUT はアドレスオフセット 0x0400 に配置され、バックグラウンド CLUT はアドレスオフセット 0x0800 に配置されます。

CLUT フォーマットは、24 ビットまたは 32 ビットが可能です。[表 37: サポートされている CLUT カラーモード](#)に示すように DMA2D\_FGPFCCR レジスタ (フォアグラウンド CLUT) または DMA2D\_BGPFCCR レジスタ (バックグラウンド CLUT) の CCM ビットを介して CLUT フォーマットを設定します。



表 37. サポートされている CLUT カラーモード

CCM	CLUT カラーモード
0	32 ビット ARGB8888
1	24 ビット RGB888

システムメモリ内の CLUT データの構成方法は表 38 : メモリ内の CLUT データ順序に規定していません。

表 38. メモリ内の CLUT データ順序

CLUT カラーモード	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A <sub>0</sub> [7:0]	R <sub>0</sub> [7:0]	G <sub>0</sub> [7:0]	B <sub>0</sub> [7:0]
RGB888	B <sub>1</sub> [7:0]	R <sub>0</sub> [7:0]	G <sub>0</sub> [7:0]	B <sub>0</sub> [7:0]
	G <sub>2</sub> [7:0]	B <sub>2</sub> [7:0]	R <sub>1</sub> [7:0]	G <sub>1</sub> [7:0]
	R <sub>3</sub> [7:0]	G <sub>3</sub> [7:0]	B <sub>3</sub> [7:0]	R <sub>2</sub> [7:0]

### 9.3.6 DMA2D ブレンダー

DMA2D ブレンダーは、転送元ピクセルをペアでブレンドし、出来上がりのピクセルを計算します。ブレンドは次の式に従って行われます。

$$\alpha_{Mult} = \frac{\alpha_{FG} \cdot \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

$$C_{OUT} = \frac{C_{FG} \alpha_{FG} + C_{BG} \alpha_{BG} - C_{BG} \alpha_{Mult}}{\alpha_{OUT}} \quad \text{ここで } C = R \text{ または } G \text{ または } B$$

除算では、最も近い整数に切り捨てられます。

ブレンダーは設定レジスタを必要としません。ブレンダーの使用方法は DMA2D\_CR レジスタの MODE[1:0] フィールドで定義される DMA2D 動作モードによって異なります。

## 9.3.7 DMA2D 出力 PFC

出力 PFC は、32 ビットから DMA2D 出力ピクセルフォーマット変換設定レジスタ (DMA2D\_OPFCCR) の CM[2:0] フィールドで定義された出力フォーマットへのピクセルフォーマット変換を行います。

サポートされている出力フォーマットを [表 39: 出力でサポートされているカラーモード](#) に示します。

**表 39. 出力でサポートされているカラーモード**

CM[2:0]	カラーモード
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444

## 9.3.8 DMA2D 出力 FIFO

出力 FIFO は、出力 PFC に定義されているカラーフォーマットに従ってピクセルをプログラムします。

転送先領域は、次の制御レジスタによって定義します。

- DMA2D 出力メモリアドレスレジスタ (DMA2D\_OMAR)
- DMA2D 出力オフセットレジスタ (DMA2D\_OOR)
- DMA2D ライン数レジスタ (ライン数およびラインあたりのピクセル数) (DMA2D\_NLR)

DMA2D がレジスタからメモリモードで動作する場合、設定された出力矩形領域が、32 ビット、24 ビットまたは 16 ビットの固定値が格納された DMA2D 出力カラーレジスタ (DMA2D\_OCCLR) で指定された色で塗りつぶされます。フォーマットは、DMA2D\_OPFCCR レジスタの CM[2:0] フィールドで選択します。

データは [表 40: メモリ内のデータ順序](#) で定義されている順番でメモリに格納されます。

**表 40. メモリ内のデータ順序**

カラーモード	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A <sub>0</sub> [7:0]	R <sub>0</sub> [7:0]	G <sub>0</sub> [7:0]	B <sub>0</sub> [7:0]
RGB888	B <sub>1</sub> [7:0]	R <sub>0</sub> [7:0]	G <sub>0</sub> [7:0]	B <sub>0</sub> [7:0]
	G <sub>2</sub> [7:0]	B <sub>2</sub> [7:0]	R <sub>1</sub> [7:0]	G <sub>1</sub> [7:0]
	R <sub>3</sub> [7:0]	G <sub>3</sub> [7:0]	B <sub>3</sub> [7:0]	R <sub>2</sub> [7:0]
RGB565	R <sub>1</sub> [4:0]G <sub>1</sub> [5:3]	G <sub>1</sub> [2:0]B <sub>1</sub> [4:0]	R <sub>0</sub> [4:0]G <sub>0</sub> [5:3]	G <sub>0</sub> [2:0]B <sub>0</sub> [4:0]
ARGB1555	A <sub>1</sub> [0]R <sub>1</sub> [4:0]G <sub>1</sub> [4:3]	G <sub>1</sub> [2:0]B <sub>1</sub> [4:0]	A <sub>0</sub> [0]R <sub>0</sub> [4:0]G <sub>0</sub> [4:3]	G <sub>0</sub> [2:0]B <sub>0</sub> [4:0]
ARGB4444	A <sub>1</sub> [3:0]R <sub>1</sub> [3:0]	G <sub>1</sub> [3:0]B <sub>1</sub> [3:0]	A <sub>0</sub> [3:0]R <sub>0</sub> [3:0]	G <sub>0</sub> [3:0]B <sub>0</sub> [3:0]

32 ビットに調整された RGB888 は ARGB8888 モードでサポートされます。

## 9.3.9 DMA2D AHB マスタポートタイマ

AHB マスタポートには 8 ビットのタイマが組込まれており、クロスバーの帯域幅を任意に制限することができます。

このタイマは AHB クロックによって駆動され、2 つの連続アクセス間のデッドタイムをカウントします。これによって帯域幅の使用を制限することができます。

タイマの有効化とデッドタイム値は AHB マスタポートタイマ設定レジスタ (DMA2D\_AMPTCR) によって設定します。

## 9.3.10 DMA2D トランザクション

DMA2D トランザクションは、一定数のデータ転送のシーケンスで構成されています。データの数と幅はソフトウェアによりプログラムすることができます。

各 DMA2D データ転送は、最大 4 つのステップで構成されています。

1. DMA2D\_FGMAR レジスタで示されるメモリ位置からのデータローディングおよびDMA2D\_FGCR で定義されるピクセルフォーマット変換。
2. DMA2D\_BGMAR レジスタで示されるメモリ位置からのデータローディングおよびDMA2D\_BGCR で定義されるピクセルフォーマット変換。
3. アルファ値に関する PFC 操作の結果として生じるアルファチャネルに従ったすべての読み出しピクセルのブレンディング。
4. DMA2D\_OCR レジスタに従って出来たピクセルのピクセルフォーマット変換および DMA2D\_OMAR レジスタによってアドレス指定されたメモリ位置へのデータのプログラミング。

## 9.3.11 DMA2D 設定

データ転送元およびデータ転送先は両方ともアドレス 0x0000 0000 と 0xFFFF FFFF の間の全 4 G バイトメモリ領域内のペリフェラルおよびメモリを対象にすることができます。

DMA2D は、DMA2D\_CR レジスタの MODE[1:0] ビットによって次の 4 つのモードのいずれでも動作することができます。

- レジスタからメモリへの転送
- メモリ間転送
- PFC を伴うメモリ間転送
- PFC とブレンディングを伴うメモリ間転送

### レジスタからメモリへの転送

“レジスタからメモリ”モードは、ユーザ定義領域を事前定義された色で塗りつぶすために使用します。

カラーフォーマットは DMA2D\_OPFCCR で設定します。

DMA2D はどのソースからもデータのフェッチを行いません。DMA2D\_OCOLR レジスタで定義された色を、DMA2D\_OMAR で示されるアドレスの、DMA2D\_NLR および DMA2D\_OOR で定義される領域に書き込むだけです。

### メモリ間転送

メモリ間モードでは、DMA2D はグラフィカルなデータの変換は何も行いません。フォアグラウンド入力 FIFO はバッファの役目を果たし、データは、DMA2D\_FGMAR で定義される転送元のメモリ位置から DMA2D\_OMAR が示す転送先のメモリ位置へ転送されます。

DMA2D\_FGPFCCR レジスタの CM[3:0] ビットでプログラムされているカラーモードによって、入力および出力の両方のピクセルあたりのビット数を定義します。

転送する領域のサイズは、転送元に対しては DMA2D\_NLR および DMA2D\_FGOR レジスタで定義され、デスティネーションに対しては DMA2D\_NLR および DMA2D\_OOR レジスタで定義されます。

## PFC を伴うメモリ間転送

このモードでは DMA2D は転送元データのピクセルフォーマット変換を行いそれを転送先メモリ位置に格納します。

転送する領域のサイズは、転送元に対しては DMA2D\_NLR および DMA2D\_FGOR レジスタで定義され、転送先に対しては DMA2D\_NLR および DMA2D\_OOR レジスタで定義されます。

データは DMA2D\_FGMAR レジスタで定義されている位置からフェッチされ、フォアグラウンドPFCによって処理されます。元のピクセルフォーマットは DMA2D\_FGPFCCR レジスタで設定します。

元のピクセルフォーマットがダイレクトカラーモードである場合、カラーチャネルはすべて 8 ビットに拡張されます。

ピクセルフォーマットがインダイレクトカラーモードである場合、関連する CLUT を CLUT メモリにロードする必要があります。

CLUT のローディングは以下の流れに従って自動的に行うことができます。

1. CLUT アドレスを DMA2D\_FGCMAR に設定します。
2. DMA2D\_FGPFCCR レジスタの CS[7:0] ビットに CLUT サイズを設定します。
3. DMA2D\_FGPFCCR レジスタの CCM ビットに CLUT フォーマット (24 ビットまたは 32 ビット) を設定します。
4. DMA2D\_FGPFCCR レジスタの START ビットをセットすることにより CLUT ローディングを開始します。

CLUT のローディングが完了すると、DMA2D\_IFR レジスタの CTCIF フラグが立ち、DMA2D\_CR レジスタの CTCIE ビットがセットされていれば割り込みが生成されます。自動 CLUT ローディング処理は標準的な DMA2D 転送と並行して動作することはできません。

CLUT は CPU から書き込むこともできれば、APB ポート経由で他のマスタによって書き込むこともできます。DMA2D 転送が進行中であつ CLUT を使用しているとき (インダイレクトカラーフォーマット)、CLUT にはアクセスできません。

カラー変換処理と並行して、DMA2D\_FGPFCCR レジスタにプログラムされている値によっては、アルファ値を追加または変更することができます。元のイメージにアルファチャネルが無い場合、デフォルトのアルファ値 0xFF が自動的に追加されて完全に不透明なピクセルになります。アルファ値は、DMA2D\_FGPFCCR レジスタの AM[1:0] ビットに従って修正することができます。

- 変更しない設定ができます。
- DMA2D\_FGPFCCR レジスタの ALPHA[7:0] 値で定義された値で置き換えることができます。
- 元の値に DMA2D\_FGPFCCR レジスタの ALPHA[7:0] 値を掛けて 255 で割った値で置き換えることができます。

結果の 32 ビットデータは OUT PFC によって、DMA2D\_OPFCCR レジスタの CM[2:0] フィールドで指定されたフォーマットにエンコードされます。CLUT 生成処理がサポートされていないため、出力ピクセルフォーマットをインダイレクトモードにすることはできません。

処理されたデータは DMA2D\_OMAR が指すメモリ位置に書き込まれます。

## PFC とブレンディングを伴うメモリ間転送

このモードでは、DMA2D\_FGMAR および DMA2D\_BGMAR で定義されるメモリ位置からフォアグラウンド FIFO およびバックグラウンド FIFO に 2 つの転送元データがフェッチされます。

メモリ間モードで説明しているように 2 つのピクセルフォーマットコンバータを、設定する必要があります。各ピクセルフォーマットコンバータは独立して独自の CLUT メモリを持つため、これらに対して別々に設定することができます。

各ピクセルがそれぞれの PFC によって 32 ビットに変換された後、それらは次の式に従ってブレンドされます。

$$\text{次の場合: } \alpha_{\text{Mult}} = \frac{\alpha_{\text{FG}} \cdot \alpha_{\text{BG}}}{255}$$

$$\alpha_{\text{OUT}} = \alpha_{\text{FG}} + \alpha_{\text{BG}} - \alpha_{\text{Mult}}$$

$$C_{\text{OUT}} = \frac{C_{\text{FG}} \cdot \alpha_{\text{FG}} + C_{\text{BG}} \cdot \alpha_{\text{BG}} - C_{\text{BG}} \cdot \alpha_{\text{Mult}}}{\alpha_{\text{OUT}}} \quad \text{ここで } C = R \text{ または } G \text{ または } B$$

除算では、小数点以下は切り捨てられます。

結果の 32 ビットピクセル値は出力 PFC により指定出力フォーマットに従ってエンコードされ、データは DMA2D\_OMAR が指す転送先のメモリ位置に書き込まれます。

## 設定エラー検出

DMA2D は転送を行う前に設定が正しいことを確認します。新規の転送/自動ローディングが開始される時、誤った設定が検出されると、ハードウェアによって設定エラー割り込みフラグがセットされます。DMA2D\_CR レジスタの CEIE ビットがセットされている場合、割り込みが生成されます。

検出することができる誤った設定は次のとおりです。

- フォアグラウンド CLUT 自動ローディング: DMA2D\_FGCMAR の MA ビットが DMA2D\_FGPFCCR の CCM ビットと整合が取れていない
- バックグラウンド CLUT 自動ローディング: DMA2D\_BGCMAR の MA ビットが DMA2D\_BGPFCCR の CCM ビットと整合が取れていない
- メモリ転送 (“レジスタからメモリ”モードを除く): DMA2D\_FGMAR の MA ビットが DMA2D\_FGPFCCR の CM ビットと整合が取れていない
- メモリ転送 (“レジスタからメモリ”モードを除く): DMA2D\_FGPFCCR の CM ビットが無効
- メモリ転送 (“レジスタからメモリ”モードを除く): DMA2D\_FGPFCCR の CM ビットが A4 または L4 のとき DMA2D\_NLR の PL ビットが奇数
- メモリ転送 (“レジスタからメモリ”モードを除く): DMA2D\_FGPFCCR の CM ビットが A4 または L4 のとき DMA2D\_FGOR の LO ビットが奇数
- メモリ転送(ブレンディングモードのみ): DMA2D\_BGMAR の MA ビットが DMA2D\_BGPFCCR の CM ビットと整合が取れていない
- メモリ転送: DMA2D\_BGPFCCR の CM ビットが無効 (ブレンディングモードのみ)
- メモリ転送 (ブレンディングモードのみ): DMA2D\_BGPFCCR の CM ビットが A4 または L4 のとき DMA2D\_NLR の PL ビットが奇数
- メモリ転送 (ブレンディングモードのみ): DMA2D\_BGPFCCR の CM ビットが A4 または L4 のとき DMA2D\_BGOR の LO ビットが奇数

- メモリ転送 (メモリ間モードを除く) : DMA2D\_OMAR の MA ビットが DMA2D\_OPFCCR の CM ビットと整合が取れていない
- メモリ転送 (メモリ間モードを除く) : DMA2D\_OPFCCR の CM ビットが無効
- メモリ転送 : DMA2D\_NLR の NL ビット = 0
- メモリ転送 : DMA2D\_NLR の PL ビット = 0

### 9.3.12 DMA2D 転送制御 (開始、サスペンド、アボート、完了)

DMA2D が設定されると、DMA2D\_CR レジスタの START ビットをセットすることにより転送を開始できます。転送が完了すると、START ビットが自動的にリセットされ、DMA2D\_ISR レジスタの TCIF フラグが立ちます。DMA2D\_CR レジスタの TCIE ビットがセットされている場合は、割り込みを生成できます。

DMA2D\_CR レジスタの SUSP ビットをセットすることにより、ユーザアプリケーションからいつでも DMA2D をサスペンドすることができます。その後、DMA2D\_CR レジスタの ABORT ビットをセットすることにより、トランザクションをアボートすることができます。また、DMA2D\_CR レジスタの SUSP ビットをリセットすることによりリスタートすることができます。

DMA2D\_CR レジスタの ABORT ビットをセットすることにより、ユーザアプリケーションからいつでも進行中のトランザクションをアボートすることができます。この場合、TCIF フラグは立ちません。

DMA2D\_FGPFCCR や DMA2D\_BGPFCCR レジスタのそれぞれの START ビットを使用することによって、自動 CLUT 転送もアボートしたりサスペンドしたりすることができます。

### 9.3.13 ウォーターマーク

ウォーターマークをプログラムして所定のラインの最後のピクセルが転送先のメモリ領域に書き込まれたときに割り込みを生成することができます。

ライン番号は、DMA2D\_LWR レジスタの LW[15:0] フィールドで定義します。

ラインの最後のピクセルが転送されると、DMA2D\_CR レジスタの TWIE ビットがセットされていれば、DMA2D\_ISR レジスタの TWIF フラグが立ち、割り込みが生成されます。

### 9.3.14 エラー管理

次の 2 種類のエラーをトリガすることができます。

- DMA2D\_ISR レジスタの TEIF フラグによって通知される AHB マスタポートエラー。
- CLUT アクセス (CLUT ローディング中または DMA2D 転送中に CPU が CLUT にアクセスしようとする) によって引き起こされる競合。DMA2D\_ISR レジスタの CAEIF フラグによって通知されます。

両方のフラグはそれぞれ DMA2D\_CR レジスタの割り込みイネーブルフラグに関連付けられており、必要に応じて割り込みを生成します (TEIE と CAEIE)。

### 9.3.15 AHB デッドタイム

AHB 帯域幅の使用を制限するために、2 つの連続した AHB アクセス間のデッドタイムをプログラムすることができます。

この機能は、DMA2D\_AMTCR レジスタの EN ビットをセットすることによって有効にできます。

デッドタイム値は、DMA2D\_AMTCR レジスタの DT[7:0] フィールドに格納します。この値は AHB バス上の 2 つの連続するトランザクション間の最小保証サイクル数を表します。

DMA2D が実行中のデッドタイム値の更新は、次の AHB 転送に反映されます。



## 9.4 DMA2D 割り込み

割り込みは次のイベントによって生成できます。

- 設定エラー
- CLUT 転送完了
- CLUT アクセスエラー
- 転送ウォーターマークに到達
- 転送完了
- 転送エラー

高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 41. DMA2D 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
設定エラー	CEIF	CEIE
CLUT 転送完了	CTCIF	CTCIE
CLUT アクセスエラー	CAEIF	CAEIE
転送ウォーターマーク	TWF	TWIE
転送完了	TCIF	TCIE
転送エラー	TEIF	TEIE

## 9.5 DMA2D レジスタ

### 9.5.1 DMA2D 制御レジスタ (DMA2D\_CR)

アドレスオフセット : 0x0000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CEIE	CTCIE	CAEIE	TWIE	TCIE	TEIE	Res.	Res.	Res.	Res.	Res.	ABORT	SUSP	START
		rW	rW	rW	rW	rW	rW						rs	rW	rs

ビット 31:18 予約済みであり、リセット値のままにしておかなければなりません。

ビット 17:16 **MODE** : DMA2D モード

このビットは、ソフトウェアによってセット/クリアされます。転送中は修正できません。

00 : メモリ間転送 (FG フェッチのみ)

01 : PFCを伴うメモリ間転送 (FG PFC アクティブで FG フェッチのみ)

10 : ブレンディングを伴うメモリ間転送 (PFC およびブレンディングを伴う FG フェッチおよび BG フェッチ)

11 : レジスタからメモリ転送 (FG および BG なし、出カステージのみアクティブ)

ビット 15:14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13 **CEIE** : 設定エラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : CE 割り込みは無効です。

1 : CE 割り込みは有効です。

ビット 12 **CTCIE** : CLUT 転送完了割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : CTC 割り込みは無効です。

1 : CTC 割り込みは有効です。

ビット 11 **CAEIE** : CLUT アクセスエラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : CAE 割り込みは無効です。

1 : CAE 割り込みは有効です。

ビット 10 **TWIE** : 転送ウォーターマーク割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : TW 割り込みは無効です。

1 : TW 割り込みは有効です。

ビット 9 **TCIE** : 転送完了割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : TC 割り込みは無効です。

1 : TC 割り込みは有効です。

ビット 8 **TEIE** : 転送エラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : TE 割り込みは無効です。

1 : TE 割り込みは有効です。

ビット 7:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2 **ABORT** : アボート

このビットを使用して現在の転送を中止できます。このビットはソフトウェアによってセットされ、START ビットがリセットされたときにハードウェアによって自動的にリセットされます。

0 : 転送アボートは要求されていません。

1 : 転送アボートが要求されています。

ビット 1 **SUSP** : サスペンド

このビットを使用して現在の転送をサスペンドできます。このビットは、ソフトウェアによってセット/クリアされます。このビットは START ビットがリセットされたときにハードウェアによって自動的にリセットされます。

0 : 転送は一時停止されません。

1 : 転送は一時停止されます。

ビット 0 **START** : Start

このビットを使用し、各種設定レジスタにロードしたパラメータに従って DMA2D を開始することができます。このビットは以下のイベントにより自動的にリセットされます。

- 転送終了時
- DMA2D\_CR の ABORT ビットをセットすることにより、ユーザアプリケーションがデータ転送をアボートしたとき
- データ転送エラーが発生したとき
- 設定エラーまたはすでに別の転送動作を行っているためデータ転送が開始されないとき (自動 CLUT ローディング)。

## 9.5.2 DMA2D 割り込みステータスレジスタ (DMA2D\_ISR)

アドレスオフセット : 0x0004

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF									
										r	r	r	r	r	r

ビット 31:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5 **CEIF** : 設定エラー割り込みフラグ

このビットは DMA2D\_CR、DMA2DFGPFCCR、または DMA2D\_BGPFCCR の START ビットがセットされ、誤った設定がプログラムされたときセットされます。

ビット 4 **CTCIF** : CLUT 転送完了割り込みフラグ

このビットは、システムメモリ領域から内部 DMA2D メモリへの CLUT コピーが完了したときセットされます。

ビット 3 **CAEIF** : CLUT アクセスエラー割り込みフラグ

このビットは、CLUT がシステムメモリから内部 DMA2D へ自動的にコピーされている間に CPU が CLUT にアクセスした場合にセットされます。

ビット 2 **TWIF** : 転送ウォーターマーク割り込みフラグ

このビットはウォーターマークを付けたラインの最後のピクセルが転送されたときセットされず。

ビット 1 **TCIF** : 転送完了割り込みフラグ

このビットは DMA2D 転送動作が完了したときセットされます (データ転送のみ)。

ビット 0 **TEIF** : 転送エラー割り込みフラグ

このビットは DMA 転送中 (データ転送または自動 CLUT ローディング) にエラーが発生したときセットされます。

## 9.5.3 DMA2D 割り込みフラグクリアレジスタ (DMA2D\_IFCR)

アドレスオフセット : 0x0008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	CTEIF									
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5 **CCEIF** : 設定エラー割り込みフラグクリア

このビットを 1 にプログラムすると DMA2D\_ISR レジスタの CEIF フラグがクリアされます。

ビット 4 **CCTCIF** : CLUT 転送完了割り込みフラグクリア

このビットを 1 にプログラムすると DMA2D\_ISR レジスタの CTCIF フラグがクリアされます。

ビット 3 **CAECIF** : CLUT アクセスエラー割り込みフラグクリア

このビットを 1 にプログラムすると DMA2D\_ISR レジスタの CAEIF フラグがクリアされます。

ビット 2 **CTWIF** : ウォーターマーク転送割り込みフラグクリア

このビットを 1 にプログラムすると DMA2D\_ISR レジスタの TWIF フラグがクリアされます。

ビット 1 **CTCIF** : 転送完了割り込みフラグクリア

このビットを 1 にプログラムすると DMA2D\_ISR レジスタの TCIF フラグがクリアされます。

ビット 0 **CTEIF** : 転送エラー割り込みフラグクリア

このビットを 1 にプログラムすると DMA2D\_ISR レジスタの TEIF フラグがクリアされます。

## 9.5.4 DMA2D フォアグラウンドメモリアドレスレジスタ (DMA2D\_FGMAR)

アドレスオフセット : 0x000C

リセット値 : 0x0000 0000

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

ビット 31:0 **MA[31:0]** : メモリアドレス

フォアグラウンドイメージに使用されるデータのアドレスです。このレジスタには、データ転送が無効のときにのみ書き込みできます。データ転送が開始されるとこのレジスタは読み出し専用になります。

アドレス割り当てでは選択されたイメージフォーマットと整合する必要があります。たとえば、32 ビット/ピクセルフォーマットは 32 ビットに調整する必要があり、16 ビット/ピクセルフォーマットは 16 ビットに調整する必要があり、4 ビット/ピクセルフォーマットは 8 ビットに調整する必要があります。

## 9.5.5 DMA2D フォアグラウンドオフセットレジスタ (DMA2D\_FGOR)

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

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

ビット 31:14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13:0 **LO[13:0]** : ラインオフセット

ピクセルで表されるフォアグラウンドに使用するラインオフセットです。この値はアドレスの生成に使用されます。これは次のラインの開始アドレスを決定するために各ラインの最後に追加されます。

これらのビットは、データ転送が無効のときにのみ書き込みできます。データ転送が開始されるとこれらは読み出し専用になります。

イメージフォーマットが 4 ビット/ピクセル場合、ラインオフセットは偶数である必要があります。



## 9.5.6 DMA2D バックグラウンドメモリアドレスレジスタ (DMA2D\_BGMAR)

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

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

ビット 31:0 **MA[31:0]** : メモリアドレス

バックグラウンドイメージに使用されるデータのアドレスです。このレジスタには、データ転送が無効のときにのみ書き込みできます。データ転送が開始されるとこのレジスタは読み出し専用になります。

アドレス割り当ては選択されたイメージフォーマットと整合する必要があります。たとえば、32 ビット/ピクセルフォーマットは 32 ビットに調整する必要があり、16 ビット/ピクセルフォーマットは 16 ビットに調整する必要があり、4 ビット/ピクセルフォーマットは 8 ビットに調整する必要があります。

## 9.5.7 DMA2D バックグラウンドオフセットレジスタ (DMA2D\_BGOR)

アドレスオフセット : 0x0018

リセット値 : 0x0000 0000

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

ビット 31:14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13:0 **LO[13:0]** : ラインオフセット

ピクセルで表されるバックグラウンドに使用するラインオフセットです。この値はアドレスの生成に使用されます。これは次のラインの開始アドレスを決定するために各ラインの最後に追加されます。

これらのビットは、データ転送が無効のときにのみ書き込みできます。データ転送が開始されるとこれらは読み出し専用になります。

イメージフォーマットが 4 ビット/ピクセル場合、ラインオフセットは偶数である必要があります。

## 9.5.8 DMA2D フォアグラウンド PFC 制御レジスタ (DMA2D\_FGPFCCR)

アドレスオフセット : 0x001C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	AM[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w							r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w			rc_w1	r/w	r/w	r/w	r/w	r/w

### ビット 31:24 ALPHA[7:0] : アルファ値

これらのビットによって固定のアルファチャネル値を定義します。この値は、元のアルファ値を置き換えるか、または AM[1:0] ビットによって選択されたアルファモードに従って元のアルファ値を乗算することができます。

これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 23:18 予約済みであり、リセット値のままにしておかなければなりません。

### ビット 17:16 AM[1:0] : アルファモード

これらのビットによって、フォアグラウンドイメージに使用するアルファチャネル値を選択します。これらは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

00 : フォアグラウンドイメージアルファチャネル値は修正されません。

01 : 元のフォアグラウンドアルファチャネル値を ALPHA[7:0] によって置き換えます。

10 : 元のフォアグラウンドアルファチャネル値を ALPHA[7:0] に元のアルファチャネル値を掛けた値で置き換えます。

他の設定は意味を持ちません。

### ビット 15:8 CS[7:0] : CLUT サイズ

これらのビットによって、フォアグラウンドイメージに使用する CLUT のサイズを定義します。CLUT 転送が開始されるとこのフィールドは読み出し専用になります。

CLUT エントリの数は CS[7:0] + 1 に等しい値となります。

ビット 7:6 予約済みであり、リセット値のままにしておかなければなりません。

## ビット 5 **START** : Start

このビットをセットして CLUTの自動ローディングを開始することができます。これは次の場合自動的にリセットされます。

- 転送終了時
- DMA2D\_CR の ABORT ビットをセットすることにより、ユーザアプリケーションが転送をアボートしたとき
- 転送エラーが発生したとき
- 設定エラーまたはすでに別の転送動作（データ転送または自動バックグラウンド CLUT 転送）を行っているため転送が開始されないとき

## ビット 4 **CCM** : CLUT カラーモード

このビットによって、CLUT のカラーフォーマットを定義します。これは転送が無効のときのみ書き込みできます。CLUT 転送が開始されるとこのビットは読み出し専用になります。

0 : ARGB8888

1 : RGB888

その他 : 意味を持ちません。

## ビット 3:0 **CM[3:0]** : カラーモード

これらのビットによって、フォアグラウンドイメージのカラーフォーマットを定義します。これらは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

0000 : ARGB8888

0001 : RGB888

0010 : RGB565

0011 : ARGB1555

0100 : ARGB4444

0101 : L8

0110 : AL44

0111 : AL88

1000 : L4

1001 : A8

1010 : A4

その他 : 意味を持ちません。

## 9.5.9 DMA2D フォアグラウンドカラーレジスタ (DMA2D\_FGCOLR)

アドレスオフセット : 0x0020

リセット値 : 0x0000 0000

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:16 **RED[7:0]** : 赤の値

これらのビットによって、フォアグラウンドイメージの A4 または A8 モードに対する赤の値を定義します。これらは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 15:8 **GREEN[7:0]** : 緑の値

これらのビットによって、フォアグラウンドイメージの A4 または A8 モードに対する緑の値を定義します。これらは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 7:0 **BLUE[7:0]** : 青の値

これらのビットによって、フォアグラウンドイメージの A4 または A8 モードに対する青の値を定義します。これらは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

## 9.5.10 DMA2D バックグラウンド PFC 制御レジスタ (DMA2D\_BGPFCCR)

アドレスオフセット : 0x0024

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	AM[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW							rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW			rc_w1	rW	rW	rW	rW	rW

ビット 31:24 **ALPHA[7:0]** : アルファ値

これらのビットによって固定のアルファチャネル値を定義します。この値は、元のアルファ値を置き換えるか、または AM[1:0] ビットによって選択されたアルファモードに従って元のアルファ値を乗算することができます。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 23:18 予約済みであり、リセット値のままにしておかなければなりません。

ビット 17:16 **AM[1:0]** : アルファモード

これらのビットによって、バックグラウンドイメージに使用するアルファチャネル値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

00 : フォアグラウンドイメージアルファチャネル値は修正されません。

01 : 元のバックグラウンドアルファチャネル値を ALPHA[7:0] によって置き換えます。

10 : 元のバックグラウンドアルファチャネル値を ALPHA[7:0] に元のアルファチャネル値を掛けた値で置き換えます。

その他 : 意味を持ちません。

ビット 15:8 **CS[7:0]** : CLUT サイズ

これらのビットによって、BG に使用する CLUT のサイズを定義します。CLUT 転送が開始されるとこのフィールドは読み出し専用になります。

CLUT エントリの数は CS[7:0] + 1 に等しい値となります。

ビット 7:6 予約済みであり、リセット値のままにしておかなければなりません。

**ビット 5 START** : Start

このビットをセットして CLUTの自動ローディングを開始します。このビットは次のときに自動的にリセットされます。

- 転送終了時
- DMA2D\_CR の ABORT ビットをセットすることにより、ユーザアプリケーションが転送をアボートしたとき
- 転送エラーが発生したとき
- 設定エラーまたはすでに別の転送動作（データ転送または自動バックグラウンド CLUT 転送）を行っているため転送が開始されないとき

**ビット 4 CCM** : CLUT カラーモード

これらのビットによって、CLUT のカラーフォーマットを定義します。このレジスタは、転送が無効のときにのみ書き込みできます。CLUT 転送が開始されるとこのビットは読み出し専用になります。

0 : ARGB8888

1 : RGB888

その他 : 意味を持ちません。

**ビット 3:0 CM[3:0]** : カラーモード

これらのビットによって、バックグラウンドイメージのカラーフォーマットを定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

0000 : ARGB8888

0001 : RGB888

0010 : RGB565

0011 : ARGB1555

0100 : ARGB4444

0101 : L8

0110 : AL44

0111 : AL88

1000 : L4

1001 : A8

1010 : A4

その他 : 意味を持ちません。

## 9.5.11 DMA2D バックグラウンドカラーレジスタ (DMA2D\_BGCOLR)

アドレスオフセット : 0x0028

リセット値 : 0x0000 0000

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:16 **RED[7:0]** : 赤の値

これらのビットによって、バックグラウンドイメージの A4 または A8 モードに対する赤の値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 15:8 **GREEN[7:0]** : 緑の値

これらのビットによって、バックグラウンドイメージの A4 または A8 モードに対する緑の値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 7:0 **BLUE[7:0]** : 青の値

これらのビットによって、バックグラウンドイメージの A4 または A8 モードに対する青の値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

## 9.5.12 DMA2D フォアグラウンド CLUT メモリアドレスレジスタ (DMA2D\_FGCMAR)

アドレスオフセット : 0x002C

リセット値 : 0x0000 0000

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

ビット 31:0 **MA[31:0]** : メモリアドレス

フォアグラウンドイメージ専用の CLUT アドレスに使用されるデータのアドレスです。このレジスタには、転送が行われていないときにのみ書き込みできます。CLUT 転送が開始されるとこのレジスタは読み出し専用になります。

フォアグラウンド CLUT フォーマットが 32 ビットの場合、アドレスは 32 ビット境界に整列する必要があります。

## 9.5.13 DMA2D バックグラウンド CLUT メモリアドレスレジスタ (DMA2D\_BGCMAR)

アドレスオフセット : 0x0030

リセット値 : 0x0000 0000

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

ビット 31:0 **MA[31:0]** : メモリアドレス

バックグラウンドイメージ専用の CLUT アドレスに使用されるデータのアドレスです。このレジスタには、転送が行われていないときのみ書き込みできます。CLUT 転送が開始されるとこのレジスタは読み出し専用になります。

バックグラウンド CLUT フォーマットが 32 ビットの場合、アドレスは 32 ビットに調整する必要があります。

## 9.5.14 DMA2D 出力 PFC 制御レジスタ (DMA2D\_OPFCCR)

アドレスオフセット : 0x0034

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2:0 **CM[2:0]** : カラーモード

これらのビットによって、出力イメージのカラーフォーマットを定義します。これらのビットは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

000 : ARGB8888

001 : RGB888

010 : RGB565

011 : ARGB1555

100 : ARGB4444

その他 : 意味を持ちません。

## 9.5.15 DMA2D 出力カラーレジスタ (DMA2D\_OCOLR)

アドレスオフセット : 0x0038

リセット値 : 0x0000 0000

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

ビット 31:24 **ALPHA[7:0]** : アルファチャンネル値

これらのビットによって、出力カラーのアルファチャンネルを定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 23:16 **RED[7:0]** : 赤の値

これらのビットによって、出カイメージの赤の値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 15:8 **GREEN[7:0]** : 緑の値

これらのビットによって、出カイメージの緑の値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

ビット 7:0 **BLUE[7:0]** : 青の値

これらのビットによって、出カイメージの青の値を定義します。これらのビットは、データ転送が無効のときにのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

## 9.5.16 DMA2D 出力メモリアドレスレジスタ (DMA2D\_OMAR)

アドレスオフセット : 0x003C

リセット値 : 0x0000 0000

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

ビット 31:0 **MA[31:0]** : メモリアドレス

出力 FIFO に使用されるデータのアドレスです。これらのビットは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

アドレス割り当ては選択されたイメージフォーマットと整合する必要があります。例えば、32 ビット/ピクセルフォーマットは 32 ビットに調整する必要があり、4 ビット/ピクセルフォーマットは 8 ビットに調整する必要があります。

## 9.5.17 DMA2D 出力オフセットレジスタ (DMA2D\_OOR)

アドレスオフセット : 0x0040

リセット値 : 0x0000 0000

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

ビット 31:14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13:0 **LO[13:0]** : ラインオフセット

ピクセルで表現される出力に使用するラインオフセットです。この値はアドレスの生成に使用されます。これは次のラインの開始アドレスを決定するために各ラインの最後に追加されます。これらのビットは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

## 9.5.18 DMA2D ライン数レジスタ (DMA2D\_NLR)

アドレスオフセット : 0x0044

リセット値 : 0x0000 0000

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

ビット 31:30 予約済みであり、リセット値のままにしておかなければなりません。

ビット 29:16 **PL[13:0]** : ラインあたりのピクセル

転送する領域のラインあたりのピクセル数です。これらのビットは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

入カイメージフォーマットのいずれかがピクセルあたり 4 ビットの場合、ラインあたりのピクセル数は偶数である必要があります。

ビット 15:0 **NL[15:0]** : ライン数

転送する領域のライン数です。これらのビットは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

## 9.5.19 DMA2D ラインウォーターマークレジスタ (DMA2D\_LWR)

アドレスオフセット : 0x0048

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **LW[15:0]** : ラインウォーターマーク

これらのビットによって、割り込み生成のラインウォーターマークを設定することができます。ウォーターマークを付けたラインの最後のピクセルが転送されたとき割り込みが発生します。これらのビットは、データ転送が無効のときのみ書き込みできます。転送が開始されるとこれらは読み出し専用になります。

## 9.5.20 DMA2D AHB マスタタイマ設定レジスタ (DMA2D\_AMTCR)

アドレスオフセット : 0x004C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済み

ビット 15:8 **DT[7:0]** : デッドタイム

AHB マスタポートに対する 2 つの連続アクセス間に挿入される AHB クロックサイクルのデッドタイム値です。これらのビットは 2 つの連続 AHB アクセス間の最小保証サイクル数を表します。

ビット 7:1 予約済み

ビット 0 **EN** : イネーブル

デッドタイム機能を有効にします。

## 9.5.21 DMA2D レジスタマップ

次の表に DMA2D レジスタの一覧を示します。DMA2D レジスタベースアドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

表 42. DMA2D レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	DMA2D_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	Res.	Res.	Res.	Res.	CEIE	CTCIE	CAEIE	TWIE	TCIE	TEIE	Res.	Res.	Res.	Res.	ABORT	SUSP	START
	リセット値															0	0					0	0	0	0	0					0	0	0
0x0004	DMA2D_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF
	リセット値																											0	0	0	0	0	0
0x0008	DMA2D_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	CTEIF
	リセット値																											0	0	0	0	0	0
0x000C	DMA2D_FGMAR	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0010	DMA2D_FGOR	Res.																LO[13:0]															
	リセット値																																
0x0014	DMA2D_BGMAR	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	DMA2D_BGOR	Res.																LO[13:0]															
	リセット値																																



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

表 42. DMA2D レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
0x001C	DMA2D_FGPFCCR	ALPHA[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	AM[1:0]	CS[7:0]							Res.	Res.	START	CCM	CM[3:0]																						
	リセット値	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0														
0x0020	DMA2D_FGCOLR	APLHA[7:0]								RED[7:0]							GREEN[7:0]							BLUE[7:0]																										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
0x0024	DMA2D_BGPFCCR	ALPHA[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	AM[1:0]	CS[7:0]							Res.	Res.	START	CCM	CM[3:0]																						
	リセット値	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
0x0028	DMA2D_BGCOLR	APLHA[7:0]								RED[7:0]							GREEN[7:0]							BLUE[7:0]																										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
0x002C	DMA2D_FGCMAR	MA[31:0]																																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x0030	DMA2D_BGCMAR	MA[31:0]																																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
0x0034	DMA2D_OPFCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.															CM[2:0]																		
	リセット値																															0	0	0																
0x0038	DMA2D_OCCLR	APLHA[7:0]								RED[7:0]							GREEN[7:0]							BLUE[7:0]																										
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[4:0]	GREEN[6:0]	BLUE[4:0]											
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	A	RED[4:0]	GREEN[4:0]	BLUE[4:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x003C	DMA2D_OMAR	MA[31:0]																																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
0x0040	DMA2D_OOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.															LO[13:0]																		
	リセット値																															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	DMA2D_NLR	Res.	Res.	PL[13:0]											NL[15:0]																																			
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
0x0048	DMA2D_LWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LW[15:0]																																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x004C	DMA2D_AMTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DT[7:0]														EN																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
0x0050-0x03FF		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.												
0x0400-0x07FF	DMA2D_FGCLUT	APLHA[7:0][255:0]								RED[7:0][255:0]							GREEN[7:0][255:0]							BLUE[7:0][255:0]																										
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											
0x0800-0x0BFF	DMA2D_BGCLUT	APLHA[7:0][255:0]								RED[7:0][255:0]							GREEN[7:0][255:0]							BLUE[7:0][255:0]																										
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X										

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

## 10 ネスト化されたベクタ割り込みコントローラ (NVIC)

### 10.1 NVIC の機能

ネスト化されたベクタ割り込みコントローラ NVIC には、次のような機能があります。

- 最大 98 本の STM32F75xxx および STM32F74xxx 用のマスク可能な割り込みチャネル (FPU 搭載 Cortex<sup>®</sup>-M7 の 16 本の割り込みラインは数に含まれていません。)
- 16 のプログラム可能な優先レベル (4 ビットの割り込み優先順位を使用)
- 遅延時間の少ない例外および割り込み処理
- 電源管理制御
- システム制御レジスタの実装

NVIC とプロセッサコアのインターフェースは密に結合され、割り込み処理の遅延時間を小さくし、遅れて到着した割り込みを効率的に処理できます。

コア例外を含むすべての割り込みは、NVIC によって管理されます。例外と NVIC プログラミングの詳細については、プログラミングマニュアル PM0253 を参照してください。

#### 10.1.1 SysTick 較正值レジスタ

SysTick 較正值は 18750 に固定され、SysTick クロックを 18.75 MHz (HCLK を 150 MHz に設定した状態で HCLK/8) に設定した状態で 1 ms という基準タイムベースを与えます。

#### 10.1.2 割り込みベクタと例外ベクタ

表 43 は、STM32F75xxx および STM32F74xxx デバイスのベクタテーブルです。

表 43. STM32F75xxx および STM32F74xxx ベクタテーブル

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	-	-	-	予約済み	0x0000 0000
-	-3	固定	リセット	リセット	0x0000 0004
-	-2	固定	NMI	ノンマスクابل割り込み。RCC クロックセキュリティシステム (CSS) は NMI ベクタにリンクされます。	0x0000 0008
-	-1	固定	HardFault	あらゆる種類の異常	0x0000 000C
-	0	設定可能	MemManage	メモリ管理	0x0000 0010
-	1	設定可能	BusFault	ブリフエッチ異常、メモリアクセス異常	0x0000 0014
-	2	設定可能	UsageFault	未定義命令または無効状態	0x0000 0018
-	-	-	-	予約済み	0x0000 001C - 0x0000 002B
-	3	設定可能	SVCall	SWI 命令によるシステムサービスコール	0x0000 002C
-	4	設定可能	デバッグモニタ	デバッグモニタ	0x0000 0030
-	-	-	-	予約済み	0x0000 0034
-	5	設定可能	PendSV	ペンディング可能なシステムサービスリクエスト	0x0000 0038



表 43. STM32F75xxx および STM32F74xxx ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	6	設定可能	SysTick	システムティックタイマ	0x0000 003C
0	7	設定可能	WWDG	ウィンドウ型ウォッチドッグ割り込み	0x0000 0040
1	8	設定可能	PVD	PVD 検出割り込み (EXTI ライン経由)	0x0000 0044
2	9	設定可能	TAMP_STAMP	タンパおよびタイムスタンプ割り込み (EXTI ライン経由)	0x0000 0048
3	10	設定可能	RTC_WKUP	RTC ウェイクアップ割り込み (EXTI ライン経由)	0x0000 004C
4	11	設定可能	FLASH	フラッシュグローバル割り込み	0x0000 0050
5	12	設定可能	RCC	RCC グローバル割り込み	0x0000 0054
6	13	設定可能	EXTI0	EXTI ライン 0 割り込み	0x0000 0058
7	14	設定可能	EXTI1	EXTI ライン 1 割り込み	0x0000 005C
8	15	設定可能	EXTI2	EXTI ライン 2 割り込み	0x0000 0060
9	16	設定可能	EXTI3	EXTI ライン 3 割り込み	0x0000 0064
10	17	設定可能	EXTI4	EXTI ライン 4 割り込み	0x0000 0068
11	18	設定可能	DMA1_Stream0	DMA1 ストリーム 0 グローバル割り込み	0x0000 006C
12	19	設定可能	DMA1_Stream1	DMA1 ストリーム 1 グローバル割り込み	0x0000 0070
13	20	設定可能	DMA1_Stream2	DMA1 ストリーム 2 グローバル割り込み	0x0000 0074
14	21	設定可能	DMA1_Stream3	DMA1 ストリーム 3 グローバル割り込み	0x0000 0078
15	22	設定可能	DMA1_Stream4	DMA1 ストリーム 4 グローバル割り込み	0x0000 007C
16	23	設定可能	DMA1_Stream5	DMA1 ストリーム 5 グローバル割り込み	0x0000 0080
17	24	設定可能	DMA1_Stream6	DMA1 ストリーム 6 グローバル割り込み	0x0000 0084
18	25	設定可能	ADC	ADC1、ADC2、ADC3 グローバル割り込み	0x0000 0088
19	26	設定可能	CAN1_TX	CAN1 TX 割り込み	0x0000 008C
20	27	設定可能	CAN1_RX0	CAN1 RX0 割り込み	0x0000 0090
21	28	設定可能	CAN1_RX1	CAN1 RX1 割り込み	0x0000 0094
22	29	設定可能	CAN1_SCE	CAN1 SCE 割り込み	0x0000 0098
23	30	設定可能	EXTI9_5	EXTI ライン [9:5] 割り込み	0x0000 009C
24	31	設定可能	TIM1_BRK_TIM9	TIM1 ブレーク割り込みおよび TIM9 グローバル割り込み	0x0000 00A0
25	32	設定可能	TIM1_UP_TIM10	TIM1 更新割り込みおよび TIM10 グローバル割り込み	0x0000 00A4
26	33	設定可能	TIM1_TRG_COM_TIM11	TIM1 トリガおよびコミュニケーション割り込みおよび TIM11 グローバル割り込み	0x0000 00A8
27	34	設定可能	TIM1_CC	TIM1 キャプチャ/比較割り込み	0x0000 00AC
28	35	設定可能	TIM2	TIM2 グローバル割り込み	0x0000 00B0
29	36	設定可能	TIM3	TIM3 グローバル割り込み	0x0000 00B4
30	37	設定可能	TIM4	TIM4 グローバル割り込み	0x0000 00B8

表 43. STM32F75xxx および STM32F74xxx ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
31	38	設定可能	I2C1_EV	I <sup>2</sup> C1 イベント割り込み	0x0000 00BC
32	39	設定可能	I2C1_ER	I <sup>2</sup> C1 エラー割り込み	0x0000 00C0
33	40	設定可能	I2C2_EV	I <sup>2</sup> C2 イベント割り込み	0x0000 00C4
34	41	設定可能	I2C2_ER	I <sup>2</sup> C2 エラー割り込み	0x0000 00C8
35	42	設定可能	SPI1	SPI1 グローバル割り込み	0x0000 00CC
36	43	設定可能	SPI2	SPI2 グローバル割り込み	0x0000 00D0
37	44	設定可能	USART1	USART1 グローバル割り込み	0x0000 00D4
38	45	設定可能	USART2	USART2 グローバル割り込み	0x0000 00D8
39	46	設定可能	USART3	USART3 グローバル割り込み	0x0000 00DC
40	47	設定可能	EXTI15_10	EXTI ライン [15:10] 割り込み	0x0000 00E0
41	48	設定可能	RTC_ALARM	RTC アラーム(A および B) 割り込み (EXTI ライン経由)	0x0000 00E4
42	49	設定可能	OTG_FS WKUP	USB OTG FS ウェイクアップ割り込み (EXTI ライン経由)	0x0000 00E8
43	50	設定可能	TIM8_BRK_TIM12	TIM8 ブレーク割り込みおよび TIM12 グローバル割り込み	0x0000 00EC
44	51	設定可能	TIM8_UP_TIM13	TIM8 更新割り込みおよび TIM13 グローバル割り込み	0x0000 00F0
45	52	設定可能	TIM8_TRG_COM_TIM14	TIM8 トリガおよびコミュレーション割り込みおよび TIM14 グローバル割り込み	0x0000 00F4
46	53	設定可能	TIM8_CC	TIM8 キャプチャ/比較割り込み	0x0000 00F8
47	54	設定可能	DMA1_Stream7	DMA1 ストリーム 7 グローバル割り込み	0x0000 00FC
48	55	設定可能	FSMC	FSMC グローバル割り込み	0x0000 0100
49	56	設定可能	SDMMC1	SDMMC1 グローバル割り込み	0x0000 0104
50	57	設定可能	TIM5	TIM5 グローバル割り込み	0x0000 0108
51	58	設定可能	SPI3	SPI3 グローバル割り込み	0x0000 010C
52	59	設定可能	UART4	UART4 グローバル割り込み	0x0000 0110
53	60	設定可能	UART5	UART5 グローバル割り込み	0x0000 0114
54	61	設定可能	TIM6_DAC	TIM6 グローバル割り込み、DAC1 および DAC2 アンダーランエラー割り込み	0x0000 0118
55	62	設定可能	TIM7	TIM7 グローバル割り込み	0x0000 011C
56	63	設定可能	DMA2_Stream0	DMA2 ストリーム 0 グローバル割り込み	0x0000 0120
57	64	設定可能	DMA2_Stream1	DMA2 ストリーム 1 グローバル割り込み	0x0000 0124
58	65	設定可能	DMA2_Stream2	DMA2 ストリーム 2 グローバル割り込み	0x0000 0128
59	66	設定可能	DMA2_Stream3	DMA2 ストリーム 3 グローバル割り込み	0x0000 012C
60	67	設定可能	DMA2_Stream4	DMA2 ストリーム 4 グローバル割り込み	0x0000 0130
61	68	設定可能	ETH	イーサネットグローバル割り込み	0x0000 0134

表 43. STM32F75xxx および STM32F74xxx ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
62	69	設定可能	ETH_WKUP	イーサネットウェイクアップ割り込み (EXTI ライン経由)	0x0000 0138
63	70	設定可能	CAN2_TX	CAN2 TX 割り込み	0x0000 013C
64	71	設定可能	CAN2_RX0	CAN2 RX0 割り込み	0x0000 0140
65	72	設定可能	CAN2_RX1	CAN2 RX1 割り込み	0x0000 0144
66	73	設定可能	CAN2_SCE	CAN2 SCE 割り込み	0x0000 0148
67	74	設定可能	OTG_FS	USB OTG FS グローバル割り込み	0x0000 014C
68	75	設定可能	DMA2_Stream5	DMA2 ストリーム 5 グローバル割り込み	0x0000 0150
69	76	設定可能	DMA2_Stream6	DMA2 ストリーム 6 グローバル割り込み	0x0000 0154
70	77	設定可能	DMA2_Stream7	DMA2 ストリーム 7 グローバル割り込み	0x0000 0158
71	78	設定可能	USART6	USART6 グローバル割り込み	0x0000 015C
72	79	設定可能	I2C3_EV	I <sup>2</sup> C3 イベント割り込み	0x0000 0160
73	80	設定可能	I2C3_ER	I <sup>2</sup> C3 エラー割り込み	0x0000 0164
74	81	設定可能	OTG_HS_EP1_OUT	USB OTG HS エンドポイント 1 出力グローバル割り込み	0x0000 0168
75	82	設定可能	OTG_HS_EP1_IN	USB OTG HS エンドポイント 1 入力グローバル割り込み	0x0000 016C
76	83	設定可能	OTG_HS_WKUP	USB OTG HS ウェイクアップ割り込み (EXTI 経由)	0x0000 0170
77	84	設定可能	OTG_HS	USB OTG HS グローバル割り込み	0x0000 0174
78	85	設定可能	DCMI	DCMI グローバル割り込み	0x0000 0178
79	86	設定可能	CRYP	CRYP 暗号グローバル割り込み	0x0000 017C
80	87	設定可能	HASH_RNG	ハッシュおよび RNG グローバル割り込み	0x0000 0180
81	88	設定可能	FPU	FPU グローバル割り込み	0x0000 0184
82	89	設定可能	UART7	UART7 グローバル割り込み	0x0000 0188
83	90	設定可能	UART8	UART8 グローバル割り込み	0x0000 018C
84	91	設定可能	SPI4	SPI4 グローバル割り込み	0x0000 0190
85	92	設定可能	SPI5	SPI5 グローバル割り込み	0x0000 0194
86	93	設定可能	SPI6	SPI6 グローバル割り込み	0x0000 0198
87	94	設定可能	SAI1	SAI1 グローバル割り込み	0x0000 019C
88	95	設定可能	LCD-TFT	LCD-TFT グローバル割り込み	0x0000 01A0
89	96	設定可能	LCD-TFT	LCD-TFT グローバルエラー割り込み	0x0000 01A4
90	97	設定可能	DMA2D	DMA2D グローバル割り込み	0x0000 01A8
91	98	設定可能	SAI2	SAI2 グローバル割り込み	0x0000 01AC
92	99	設定可能	Quad SPI	Quad SPI グローバル割り込み	0x0000 01B0
93	100	設定可能	LP Timer1	LP タイマ 1 グローバル割り込み	0x0000 01B4
94	101	設定可能	HDMI-CEC	HDMI-CEC グローバル割り込み	0x0000 01B8

表 43. STM32F75xxx および STM32F74xxx ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
95	102	設定可能	I2C4_EV	I2C4 イベント割り込み	0x0000 01BC
96	103	設定可能	I2C4_ER	I2C4 エラー割り込み	0x0000 01C0
97	104	設定可能	SPDIFRX	SPDIFRX グローバル割り込み	0x0000 01C4

## 11 拡張割り込み／イベントコントローラ (EXTI)

イベント／割り込みリクエストを生成するための外部割り込み／イベントコントローラは、最大 24 のエッジ検出回路で構成されます。各入力ラインは、個別に設定可能であり、タイプ（割り込みまたはイベント）と対応するトリガイベント（立ち上がり、立ち下がり、または両方）を選択できます。また、ラインごとにマスクをかけることもできます。ペンディングレジスタは、割り込みリクエストのステータスラインを維持します。

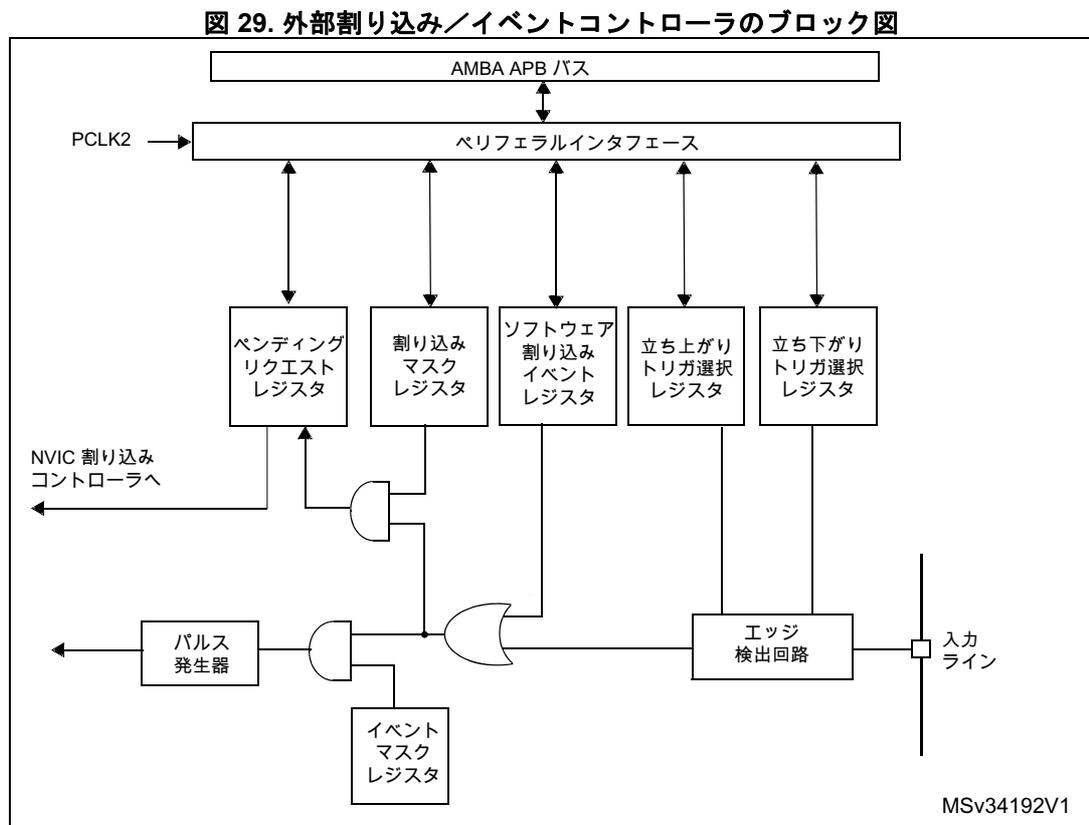
### 11.1 EXTI の主な機能

EXTI コントローラの主な機能は次のとおりです。

- 割り込み／イベントラインごとに独立したトリガとマスク
- 割り込みラインごとに専用のステータスビット
- 最大 24 本のソフトウェアイベント／割り込みリクエストの生成
- APB2 クロック周期よりも短いパルス幅の外部信号の検出。このパラメータの詳細については、STM32F75xxx および STM32F74xxx データシートの電気特性のセクションを参照してください。

### 11.2 EXTI ブロック図

図 29 にブロック図を示します。



## 11.3 ウェイクアップイベント管理

STM32F75xxx および STM32F74xxx は、コアをウェイクアップするために、外部イベントや内部イベントを処理することができます (WFE)。ウェイクアップイベントは、次のいずれかによって生成できます。

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、Cortex<sup>®</sup>-M7 システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャンネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードで設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャンネルペンディングビットをクリアする必要はありません。

外部ラインをウェイクアップイベントとして使用する方法については、[セクション 11.4: 機能詳細](#)を参照してください。

## 11.4 機能詳細

割り込みを生成するには、割り込みラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、割り込みマスクレジスタの対応するビットに "1" を書き込むことによって割り込みリクエストを有効にします。選択されたエッジが外部割り込みラインで発生すると、割り込みリクエストが生成されます。割り込みラインに対応するペンディングビットもセットされます。このリクエストは、ペンディングレジスタに "1" を書き込むことによってリセットされます。

イベントを生成するには、イベントラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、イベントマスクレジスタの対応するビットに "1" を書き込むことによってイベントリクエストを有効にします。選択されたエッジがイベントラインで発生すると、イベントパルスが生成されます。イベントラインに対応するペンディングビットはセットされません。

割り込み/イベントリクエストは、ソフトウェア割り込み/イベントレジスタに "1" を書き込むことによって、ソフトウェアによって生成することもできます。

## 11.5 ハードウェア割り込みの選択

ラインを割り込みソースとして設定するには、次の手順を使用します。

1. 対応するマスクビット (EXTI\_IMR) を設定します。
2. 割り込みラインのトリガ選択ビットを設定します (EXTI\_RTISR および EXTI\_FTISR)。
3. 24 本のラインのいずれかからの割り込みを正しく処理できるように、外部割り込みコントローラ (EXTI) にマップされた NVIC IRQ チャンネルを制御するイネーブルビットとマスクビットを設定します。

## 11.6 ハードウェアイベントの選択

ラインをイベントソースとして設定するには、次の手順を使用します。

1. 対応するマスクビット (EXTI\_EMR) を設定します。
2. イベントラインのトリガ選択ビットを設定します (EXTI\_RTISR および EXTI\_FTISR)。



## 11.7 ソフトウェア割り込み/イベントの選択

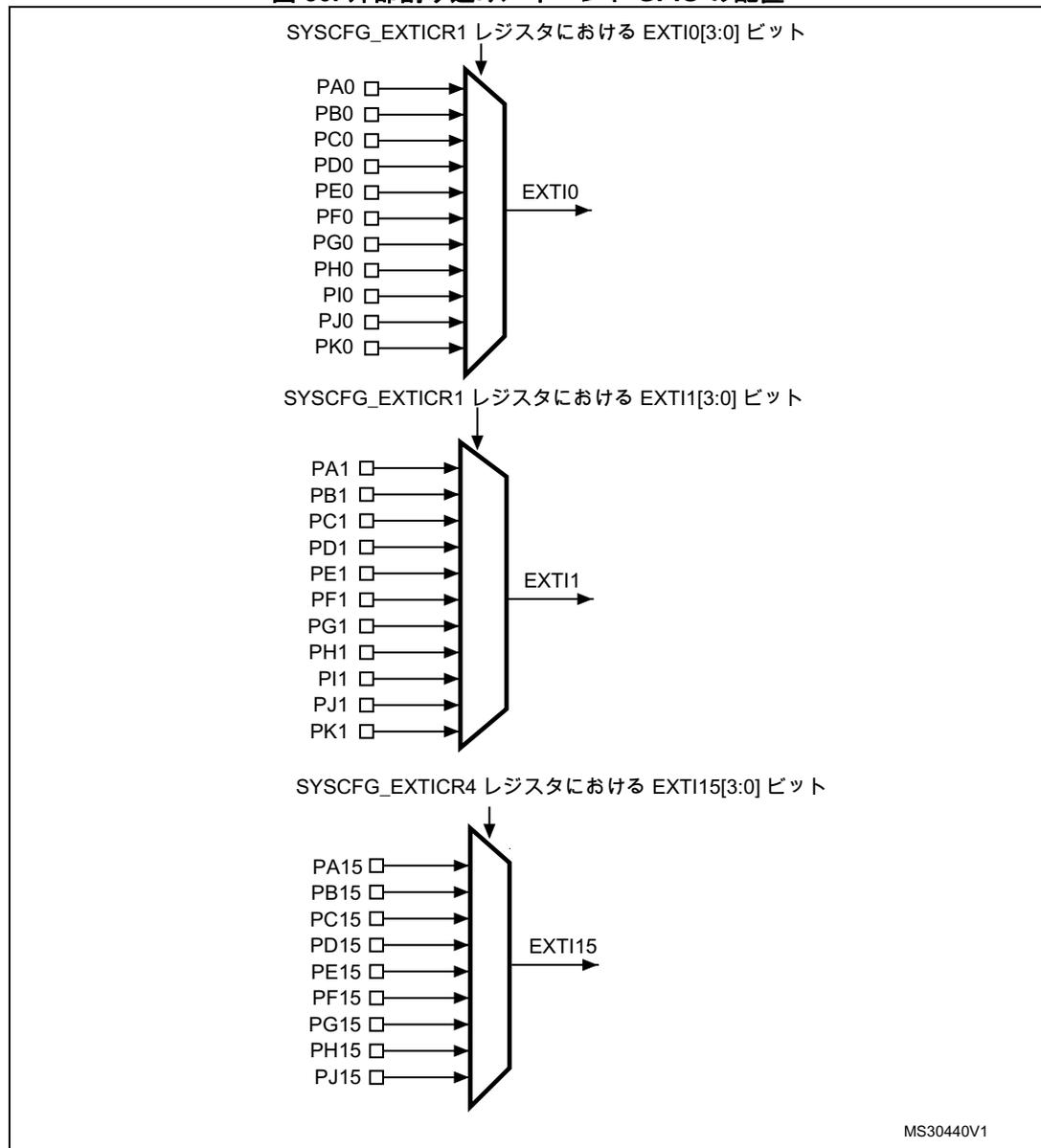
ラインは、ソフトウェア割り込み/イベントラインとして設定できます。ソフトウェア割り込みの生成には、次の手順を使用します。

1. 対応するマスクビット (EXTI\_IMR、EXTI\_EMR) を設定します。
2. ソフトウェア割り込みレジスタの必要なビットをセットします (EXTI\_SWIER)。

## 11.8 外部割り込み/イベントラインの配置

最大 168 個の GPIO は、次のように、16 本の外部割り込み/イベントラインに接続されます。

図 30. 外部割り込み/イベント GPIO の配置



他の 8 本の EXTI ラインは次のように接続されます。

- EXTI ライン 16 は PVD 出力に接続されます。
- EXTI ライン 17 は RTC アラームイベントに接続されます。
- EXTI ライン 18 は USB OTG FS ウェイクアップイベントに接続されます。
- EXTI ライン 19 はイーサネットウェイクアップイベントに接続されます。
- EXTI ライン 20 は USB OTG HS (FS で設定) ウェイクアップイベントに接続されます。
- EXTI ライン 21 は RTC のタンイベントおよびタイムスタンプイベントに接続されます。
- EXTI ライン 22 は RTC ウェイクアップイベントに接続されます。
- EXTI ライン 23 は LPTIM1 非同期イベントに接続されます。

## 11.9 EXTI レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

### 11.9.1 割り込みマスクレジスタ (EXTI\_IMR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16							
								r/w							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
r/w															

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **MRx** : ライン x の割り込みマスク

- 0 : ライン x からの割り込みリクエストはマスクされます。
- 1 : ライン x からの割り込みリクエストはマスクされません。

## 11.9.2 イベントマスクレジスタ (EXTI\_EMR)

アドレスオフセット : 0x04  
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16							
								r/w							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
r/w															

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **MRx** : ライン x のイベントマスク

- 0 : ライン x からのイベントリクエストはマスクされます。
- 1 : ライン x からのイベントリクエストはマスクされません。

## 11.9.3 立ち上がりトリガ選択レジスタ (EXTI\_RTISR)

アドレスオフセット : 0x08  
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TR23	TR22	TR21	TR20	TR19	TR18	TR17	TR16							
								r/w							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w															

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **TRx** : ライン x の立ち上がりトリガイベント設定ビット

- 0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。
- 1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

**注 :** 外部ウェイクアップラインはエッジトリガであるため、グリッチが生成されないようにする必要があります。

**EXTI\_RTISR** レジスタへの書き込み中に外部割り込みラインで立ち上がりエッジが発生した場合、ペンディングビットはセットされません。

同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この設定では、両方のエッジでトリガ条件が生成されます。

## 11.9.4 立ち下がりトリガ選択レジスタ (EXTI\_FTISR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TR23	TR22	TR21	TR20	TR19	TR18	TR17	TR16							
								r/w							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w															

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 TRx : ライン x の立ち下がりトリガイベント設定ビット

0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です

1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です

**注 :** 外部ウェイクアップラインはエッジトリガであるため、グリッチが生成されないようにする必要があります。  
EXTI\_FTISR レジスタへの書き込み中に外部割り込みラインで立ち下がりエッジが発生した場合、ペンディングビットはセットされません。  
同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この設定では、両方のエッジでトリガ条件が生成されます。

## 11.9.5 ソフトウェア割り込みイベントレジスタ (EXTI\_SWIER)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER2 3	SWIER2 2	SWIER2 1	SWIER 20	SWIER 19	SWIER 18	SWIER 17	SWIER 16
								r/w							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER 15	SWIER 14	SWIER 13	SWIER 12	SWIER 11	SWIER 10	SWIER 9	SWIER 8	SWIER7	SWIER6	SWIER5	SWIER4	SWIER3	SWIER2	SWIER 1	SWIER 0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 SWIERx : ライン x のソフトウェア割り込み

割り込みが EXTI\_IMR レジスタのライン x で有効になっている場合、SWIERx ビットが“0”のときに“1”を書き込むと EXTI\_PR レジスタの対応するペンディングビットがセットされるため、割り込みリクエストが生成されます。

EXTI\_PR レジスタの対応するビットをクリアする (対応ビットに“1”を書き込む) と、このビットはクリアされます。

## 11.9.6 ペンディングレジスタ (EXTI\_PR)

アドレスオフセット : 0x14  
リセット値 : 定義されていません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	PR23	PR22	PR21	PR20	PR19	PR18	PR17	PR16							
								rc_w1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1															

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 PRx : ペンディングビット

0 : トリガリクエストは発生していません。

1 : 選択されたトリガリクエストが発生しました。

このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。

このビットは、“1”をプログラムすることによってクリアされます。

## 11.9.7 EXTI レジスタマップ

表 44 に、EXTI レジスタマップとリセット値を示します。

表 44. 外部割り込み/イベントコントローラのレジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_IMR	Res.	MR[23:0]																														
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR	Res.	MR[23:0]																														
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_RTSR	Res.	TR[23:0]																														
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	EXTI_FTSTR	Res.	TR[23:0]																														
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	EXTI_SWIER	Res.	SWIER[23:0]																														
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	EXTI_PR	Res.	PR[23:0]																														
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 12 巡回冗長検査計算ユニット (CRC)

### 12.1 概要

CRC (Cyclic Redundancy Check : 巡回冗長検査) 計算ユニットは、8、16、または 32 ビットデータワードと、ある生成多項式から、CRC コードを得るために使用されます。

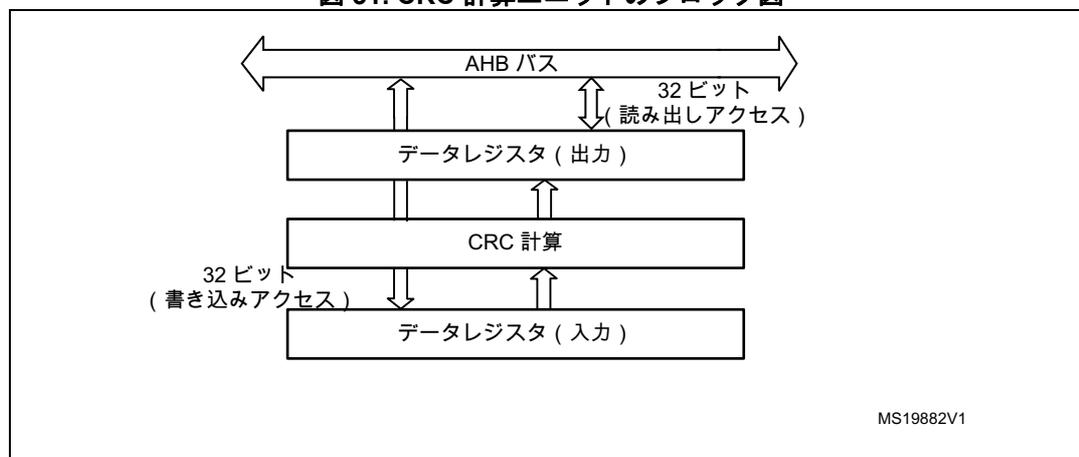
他のアプリケーションの中でも、CRC ベースのテクニックは、データ転送やストレージの整合性を確認するために使用されます。機能安全規格の範囲内では、このテクニックがフラッシュメモリの整合性を確認するひとつの手段となっています。CRC 計算ユニットは、実行時にソフトウェアのシグネチャ計算を支援します。リンク時に生成されて、特定のメモリ領域に保存されたりファレンスシグネチャと計算されたソフトウェアシグネチャが比較されます。

### 12.2 CRC の主な機能

- プログラム可能なサイズ (7、8、16、32 ビット) の完全にプログラム可能な多項式。
- サイズが 8、16、32 ビットのデータを取り扱います。
- プログラム可能な CRC の初期値
- シングル入力/出力 32 ビットデータレジスタ
- 計算時のバスのストールを避けるための入力バッファ
- データサイズが 32 ビットの場合、CRC の計算は AHB クロック 4 サイクル (HCLK) 以内に行われます。
- 汎用 8 ビットレジスタ (一時ストレージとして使用可能)
- I/O データの可逆性のオプション

## 12.3 CRC の機能説明

図 31. CRC 計算ユニットのブロック図



CRC 計算ユニットは、1つの32ビット読み出し/書き込みデータレジスタ (CRC\_DR) を持っています。このレジスタを使用して、新しいデータを入力し (書き込みアクセス)、前回のCRC計算結果を保持します (読み出しアクセス)。

データレジスタへの書き込み操作のたびに、前回のCRC値 (CRC\_DRに格納) と新しい値の組み合わせが作成されます。CRC計算は、書き込まれるデータのフォーマットに応じ、32ビットデータワード全体に対して、またはバイト単位で行われます。

CRC\_DRレジスタは、ワード、右詰めのハーフワード、右詰めのバイトによってアクセスできます。他のレジスタについては、32ビットアクセスのみ可能です。

計算の時間はデータ幅に依存します。

- 32ビットの場合、4 AHB クロックサイクル
- 16ビットの場合、2 AHB クロックサイクル
- 8ビットの場合、1 AHB クロックサイクル

入力バッファを使うと、前回のCRC計算によるウェイトステートを待つことなく、すぐに第2のデータを書き込むことができます。

与えられたバイト数に対する書き込みアクセス数を最小限に抑えるために、データサイズを動的に調節することができます。たとえば、5バイトのCRCは、1ワードの書き込みと、それに続く1バイトの書き込みで計算することができます。

入力データを逆にして、さまざまなエンディアンネス方式を管理することができます。逆転操作は、CRC\_CRレジスタのREV\_IN[1:0]ビットに応じて、8、16、および32ビットで行うことができます。

たとえば、入力データ0x1A2B3C4Dは、CRC計算では以下のように使用されます。

- 0x58D43CB2 (バイト単位でビットが逆転)
- 0xD458B23C (ハーフワード単位でビットが逆転)
- 0xB23CD458 (フルワードでビットが逆転)

また、出力データも、CRC\_CRレジスタのREV\_OUTビットをセットすることによって逆にすることができます。

操作はビットレベルで行われます。たとえば、出力データ0x11223344は0x22CC4488に変換されます。

CRC 計算機は、CRC\_CR レジスタの RESET 制御ビットを使用して、プログラム可能な値に初期化することができます (デフォルト値は 0xFFFFFFFF)。

CRC の初期値は、CRC\_INIT レジスタでプログラムすることができます。CRC\_DR レジスタは、CRC\_INIT レジスタの書き込みアクセス時に自動的に初期化されます。

CRC\_IDR レジスタを使用して、CRC 計算に関する一時的な値を保持することができます。このレジスタは、CRC\_CR レジスタの RESET ビットによる影響を受けません。

## 多項式のプログラミング可能性

多項式係数は CRC\_POL レジスタを介して完全にプログラム可能であり、多項式のサイズは CRC\_CR レジスタの POLYSIZE[1:0] ビットをプログラムすることにより、7、8、16、または 32 ビットに設定できます。偶多項式はサポートされていません。

CRC データが 32 ビット未満であれば、その値は CRC\_DR レジスタの下位ビットから読み出すことができます。

信頼できる CRC 計算結果を得るために、CRC 計算の実行中に多項式の値やサイズを変更することはできません。そのため、CRC 計算が進行中である場合、アプリケーションは計算をリセットするか、または CRC\_DR の読み出しを行ってから、多項式を変更しなければなりません。

多項式のデフォルト値は、CRC-32 (イーサネット) 多項式、0x4C11DB7。

## 12.4 CRC レジスタ

### 12.4.1 データレジスタ (CRC\_DR)

アドレスオフセット : 0x00

リセット値 : 0xFFFF FFFF

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

ビット 31:0 **DR[31:0]** : データレジスタビット

このレジスタを使用して、CRC 計算機に新しいデータを書き込みます。

読み出し時には、前回の CRC 計算結果を保持します。

データサイズが 32 ビット未満であれば、下位ビットを使用して正しい値の書き込み/読み出しを行います。

## 12.4.2 独立型データレジスタ (CRC\_IDR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:8 予約済み、クリア状態を保つ必要があります。

ビット 7:0 **IDR[7:0]** : 汎用 8 ビットデータレジスタビット

これらのビットは、1 バイトの一時的なストレージ領域として使用できます。

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

## 12.4.3 制御レジスタ (CRC\_CR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	REV_OUT	REV_IN[1:0]			POLYSIZE[1:0]		Res.	Res.	リセット							
								rw	rw	rw	rw	rw			rs	

ビット 31:8 予約済み、クリア状態を保つ必要があります。

ビット 7 **REV\_OUT** : 出力データを逆にします。

このビットは、出力データビット順序の反転を制御します。

0 : ビットの順序は変わりません。

1 : 出力フォーマットのビットが反転しています。

ビット 6:5 **REV\_IN[1:0]** : 入力データを逆にします。

これらのビットは、入力データビット順序の反転を制御します。

00 : ビットの順序は変わりません。

01 : バイト単位でビットが反転しています。

10 : ハーフワード単位でビットが反転しています。

11 : ワード単位でビットが反転しています。

ビット 4:3 **POLYSIZE[1:0]** : 多項式のサイズ  
 これらのビットは、多項式のサイズを制御します。  
 00 : 32 ビットの多項式  
 01 : 16 ビットの多項式  
 10 : 8 ビットの多項式  
 11 : 7 ビットの多項式

ビット 2:1 予約済み、クリア状態を保つ必要があります。

ビット 0 **RESET** : RESET ビット  
 このビットは、CRC 計算ユニットをリセットし、CRC\_INIT レジスタに格納された値にデータレジスタをセットするために、ソフトウェアによってセットされます。このビットはセットのみが可能で、ハードウェアによって自動的にクリアされます。

## 12.4.4 CRC の初期値 (CRC\_INIT)

アドレスオフセット : 0x10

リセット値 : 0xFFFF FFFF

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

ビット 31:0 **CRC\_INIT** : プログラム可能な CRC の初期値  
 このレジスタを使用して、CRC の初期値を書き込みます。

## 12.4.5 CRC 多項式 (CRC\_POL)

アドレスオフセット : 0x14

リセット値 : 0x04C11DB7

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

ビット 31:0 **POL[31:0]** : プログラム可能な多項式  
 このレジスタを使用して、CRC 計算に使用される多項式の係数を書き込みます。  
 多項式のサイズが 32 ビット未満であれば、下位ビットを使用して正しい値をプログラムする必要があります。



## 12.4.6 CRC レジスタマップ

表 45. CRC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	CRC_DR	DR[31:0]																																		
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x04	CRC_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]									
	リセット値																										0	0	0	0	0	0	0	0	0	
0x08	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_INIT[1:0]		POLYSIZE[1:0]		Res.	Res.	リセット		
	リセット値																										0	0	0	0	0			0		
0x10	CRC_INIT	CRC_INIT[31:0]																																		
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x14	CRC_POL	多項式係数																																		
	リセット値	0x04C11DB7																																		

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 13 フレキシブル メモリコントローラ (FMC)

フレキシブル メモリコントローラ (FMC) には、次の 3 つの メモリコントローラが含まれています。

- NOR/PSRAM メモリコントローラ
- NAND メモリコントローラ
- 同期 DRAM (SDRAM/モバイル LPDDR SDRAM) コントローラ

### 13.1 FMC の主な機能

FMC 機能ブロックは、同期および非同期のスタティックメモリ、SDRAM メモリ、NAND フラッシュメモリとのインタフェースを構成します。主要目的は、

- AHB トランザクションを該当する外部デバイスプロトコルへ翻訳すること
- 外部メモリデバイスのアクセス時間条件を満たすようにすること

すべての外部メモリが、アドレス信号、データ信号、制御信号をコントローラと共有します。各外部デバイスへ固有のチップセレクトを使ってアクセスします。FMC は外部デバイスへ一度に 1 回だけアクセスします。

FMC コントローラの主要機能は次の通りです。

- 次のスタティックメモリにマップされたデバイスとのインタフェース
  - スタティックランダムアクセスメモリ (SRAM)
  - NOR 型フラッシュメモリ/OneNAND フラッシュメモリ
  - PSRAM (4 メモリバンク)
  - 最大 8 KB のデータをチェックできる ECC ハードウェア付き NAND フラッシュメモリ
- 同期 DRAM (SDRAM/モバイル LPDDR SDRAM) メモリとのインタフェース
- NOR フラッシュメモリ、PSRAM、SDRAM などの同期デバイスに対するアクセスを高速化するバーストモードのサポート
- 非同期アクセスと同期アクセス用のプログラム可能な連続クロック出力
- 8 ビット、16 ビット、または 32 ビット幅のデータバス
- 各メモリバンクに対する独立したチップセレクト制御
- メモリバンクごとに独立した設定
- PSRAM デバイス、SRAM デバイス、SDRAM デバイスで使用する書き込みイネーブルとバイトレーン選択出力
- 外部非同期ウェイト制御
- 16x32 ビットの深さの書き込み FIFO
- SDRAM コントローラ用の 6x32 ビットの深さ (6x14 ビットアドレスタグ) のキャッシュ可能読み出し FIFO

書き込み FIFO はすべてのメモリコントローラに共通であり、以下の構成となっています。

- 書き込みデータ FIFO は、メモリへ書き込む AHB データ (最大 32 ビット) と、AHB 転送 (バーストモードまたは非シーケンシャルモード) 用の 1 ビットを格納
- 書き込みアドレス FIFO は、AHB アドレス (最大 28 ビット) と AHB データサイズ (最大 2 ビット) を格納バーストモードで動作する場合、ページ境界 (PSRAM と SDRAM) を跨ぐとき以外は、開始アドレスのみを格納します。この場合、AHBバーストは 2 つの FIFO エントリに分割されます。

書き込み FIFO は、ソフトウェアで FMC\_BCR1 レジスタの WFDIS ビットをセットすることによって無効にできます。

開始時に、ユーザーアプリケーションから FMC ピンを設定する必要があります。アプリケーションで使用しない FMC I/O ピンは、他の用途に使用することができます。

外部デバイスタイプおよびその特性を定義する FMC レジスタは、通常、起動時に設定され、次のリセットまたは電源投入まで変更されません。ただし、設定値は何時でも変更することができます。

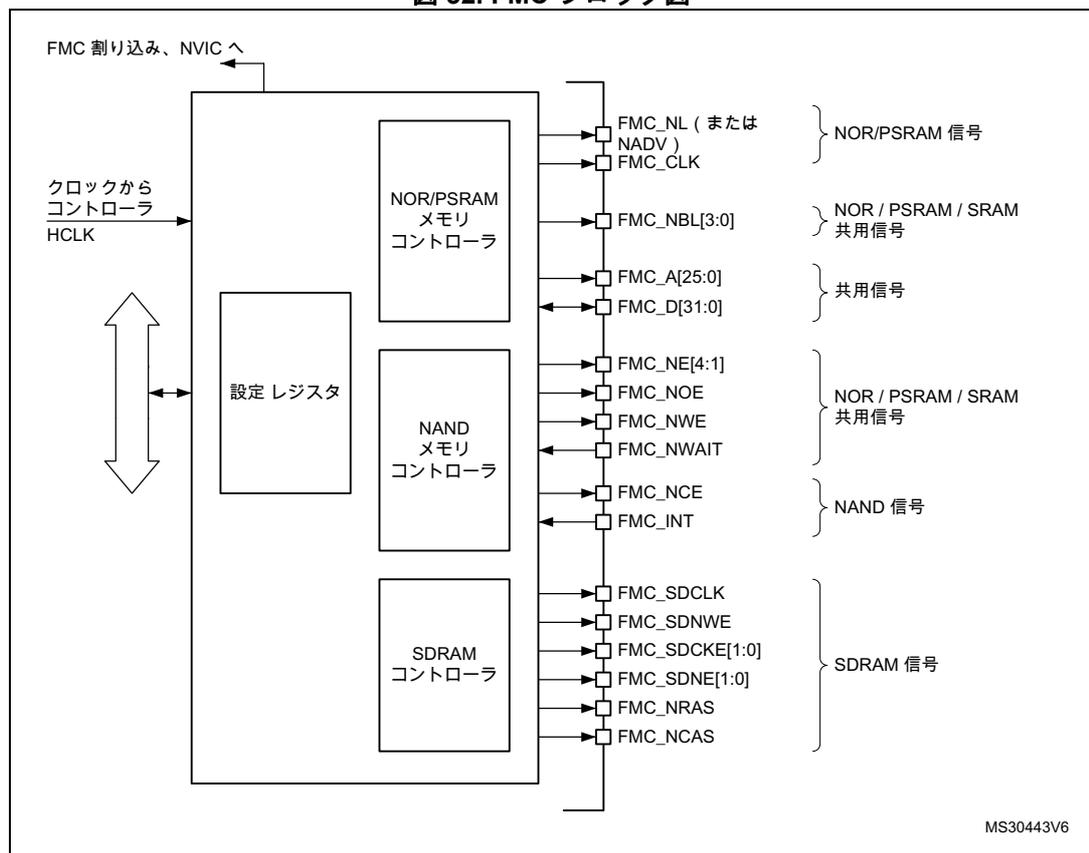
## 13.2 ブロック図

FMC は次のメインブロックで構成されます。

- AHB インタフェース (FMC 設定レジスタを含む)
- NOR フラッシュ /PSRAM/SRAM コントローラ
- SDRAM コントローラ
- NAND コントローラ

以下にブロック図を示します。

図 32. FMC ブロック図



## 13.3 AHB インタフェース

AHB スレーブインタフェースは、内部 CPU および他のバスマスタペリフェラルから外部スタティックメモリへのアクセスを可能にします。

AHB トランザクションは、外部デバイスプロトコルに変換されます。特に、選択された外部メモリが 16 または 8 ビット幅の場合、AHB 上の 32 ビット幅のトランザクションは、16 または 8 ビットの連続した複数のアクセスに分割されます。FMC チップセレクト (FMC\_NEX) は、複数の連続アクセス間にトグルしません。

FMC は、次の条件で AHB エラーを発生します。

- 有効化していない FMC バンク (バンク1~4) の読み出しまたは書き込みを行ったとき。
- FMC\_BCRx レジスタの FACCEN ビットがクリアされている状態で NOR フラッシュバンクの読み出しまたは読み出しを行ったとき。
- 書き込み保護された SDRAM バンク (SDRAM\_SDCRx レジスタの WP ビットがセット中) に対して書き込みを行ったとき。
- SDRAM アドレス範囲に違反したとき (予約済みアドレス範囲へのアクセス)。

この AHB エラーの影響は、読み出し/書き込みアクセスを試みた AHB マスタに依存します。

- アクセスが FPU を搭載した Cortex<sup>®</sup>-M7 の CPU から行われた場合は、ハードフォルト割り込みが発生します。
- アクセスが DMA コントローラから行われた場合は、DMA 転送エラーが発生し、対応する DMA チャネルが自動的に無効化されます。

AHB クロック (HCLK) は、FMC のリファレンスクロックです。

### 13.3.1 サポートされるメモリおよびトランザクション

#### 一般的なトランザクション規則

リクエストされる AHB トランザクションのデータサイズは、アクセスされる外部デバイスのデータ幅が固定であっても、8、16、または 32 ビット幅です。このため、転送に矛盾が生じることがあります。

したがって、次のようないくつかの単純なトランザクション規則を守る必要があります。

- AHB トランザクションサイズとメモリデータサイズが等しい場合：
  - この場合は問題ありません。
- AHB トランザクションサイズがメモリサイズより大きい場合：
  - この場合、FMC は、AHB トランザクションを外部データ幅に合わせて、連続した小さなメモリアクセスに分割します。FMC チップセレクト (FMC\_NEX) は、複数の連続アクセス間にトグルしません。
- AHB トランザクションサイズがメモリサイズより小さい場合：
  - 外部デバイスのタイプによっては、転送に一貫性がなくなる場合があります：
    - バイト選択機能を持つデバイス (SRAM、ROM、PSRAM、SDRAM) に対するアクセス
      - この場合、FMC は、読み出し/書き込みトランザクションを許可して、バイトレーン NBL[3:0] を通じて正しいデータにアクセスします。
      - 書き込み対象バイトは NBL[3:0] によりアドレス指定されます。
      - すべてのメモリバイトが読み出され (読み出しトランザクション中 NBL[3:0] はローに駆動されます)、不要なバイトは無視されます。

- バイト選択機能を持たないデバイス (NOR および NAND フラッシュメモリ) に対するアクセス  
この状況は、16 ビット幅のフラッシュメモリへのバイトアクセスが要求されたときに発生します。バイトモードでデバイスをアクセスできないため (フラッシュメモリに対しては 16 ビットワードの読み出し/書き込みのみが可能)、書き込みトランザクションと読み出しトランザクションが可能です (コントローラは 16 ビットメモリワード全体を読み出して、必要なバイトのみを使用)。

## NOR 型フラッシュ/PSRAM および SDRAM のラップサポート

同期メモリは、すべてのマスタがラップトランザクションを発行できるとは限らないため、不定長のリニアバーストモードに設定する必要があります。

マスタが AHB ラップトランザクションを生成すると、

- 読み出しは、2 つのリニアバーストトランザクションに分割されます。
- 書き込みは、書き込み FIFO が有効な場合は 2 つのリニアバーストトランザクションに、無効な場合はいくつかのリニアバーストトランザクションに分割されます。

## 設定レジスタ

FMC は、レジスタのセットとして設定することができます。NOR フラッシュ/PSRAM コントローラレジスタの詳細については、[セクション 13.5.6](#)を参照してください。また、[セクション 13.6.7](#)は、NAND フラッシュレジスタの詳細説明、[セクション 13.7.5](#)は SDRAM コントローラレジスタの詳細を説明しています。

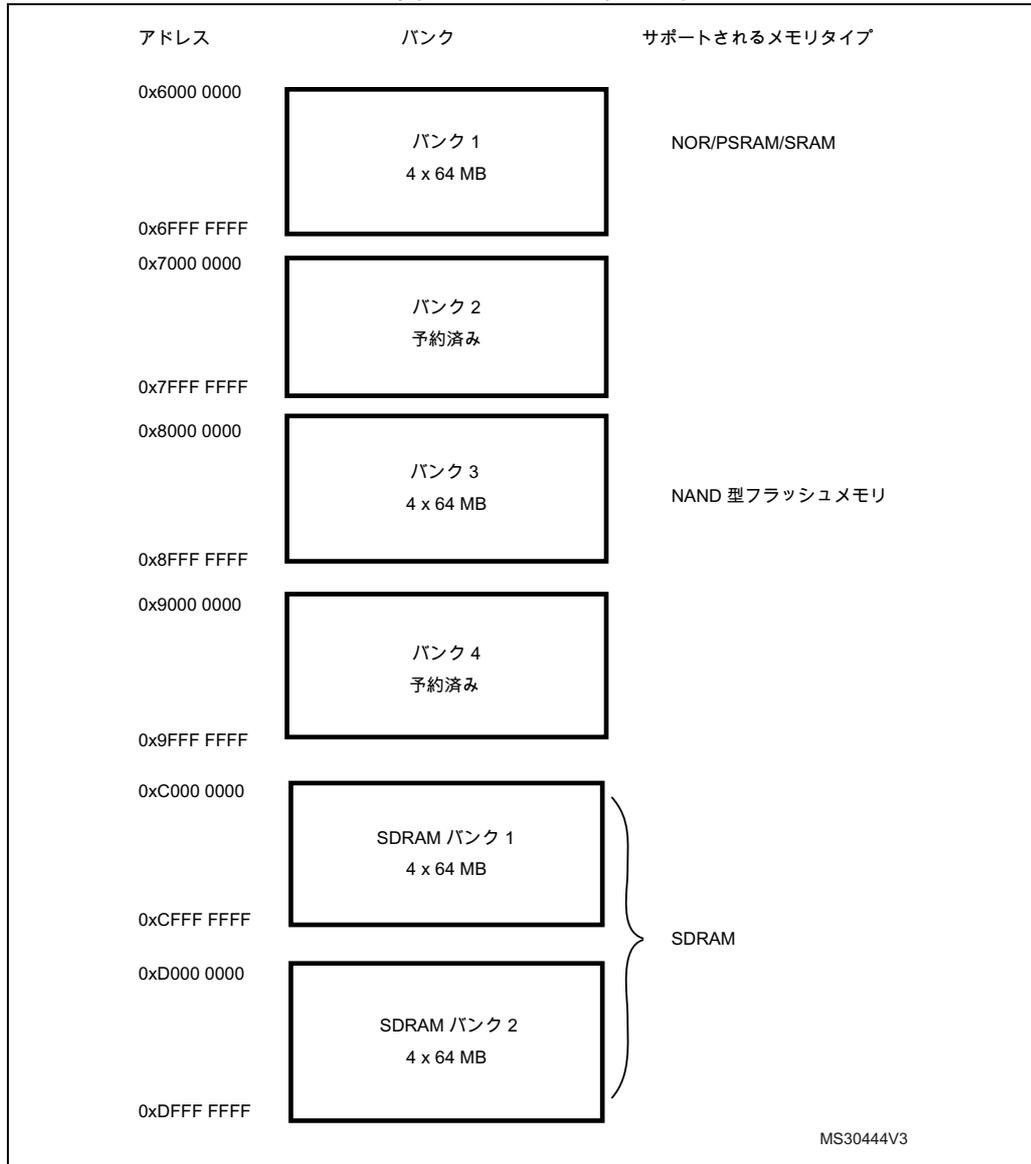
## 13.4 外部デバイスアドレスマッピング

FMC から見ると、外部メモリは、それぞれ 256 MB の固定サイズのバンクに分けられています ([図 33](#)を参照)。

- バンク 1 は、4 つまでの NOR フラッシュメモリまたは PSRAM メモリデバイスのアドレス指定に使用されます。このバンクは、4 つの専用チップセレクトを持つ 4 つの NOR/PSRAM サブバンクに分割されます。
  - バンク 1 - NOR/PSRAM 1
  - バンク 1 - NOR/PSRAM 2
  - バンク 1 - NOR/PSRAM 3
  - バンク 1 - NOR/PSRAM 4
- バンク 3 は、NAND フラッシュメモリデバイスのアドレス指定に使用されます。この空間の MPUメモリ属性をソフトウェアによってデバイスに再設定する必要があります。
- バンク 4 と 5 は、SDRAM デバイスのアドレス指定に使用されます (バンクごとに 1 つのデバイス)。

各バンクで使用されるメモリのタイプは、ユーザアプリケーションから設定レジスタを使って設定することができます。

図 33. FMC メモリバンク



## 13.4.1 NOR/PSRAM アドレスマッピング

HADDR[27:26] ビットは、表 46 に示される 4 つのメモリバンクの 1 つを選択するために使用されます。

表 46. NOR/PSRAM バンク選択

HADDR[27:26] <sup>(1)</sup>	選択されるバンク
00	バンク 1 - NOR/PSRAM 1
01	バンク 1 - NOR/PSRAM 2
10	バンク 1 - NOR/PSRAM 3
11	バンク 1 - NOR/PSRAM 4

1. HADDR は、外部メモリに変換される内部 AHB アドレスラインです。

HADDR[25:0] ビットは、外部メモリアドレスを含みます。HADDR はバイトアドレスですが、メモリはワードレベルでアクセスされるので、メモリに対して実際に発行されるアドレスは、次の表に示されるように、メモリのデータ幅に応じて変わります。

表 47. NOR/PSRAM 外部メモリアドレス

メモリ幅 <sup>(1)</sup>	メモリに発行されるデータアドレス	最大メモリ容量 (ビット)
8 ビット	HADDR[25:0]	64 Mバイト * 8 = 512 M ビット
16 ビット	HADDR[25:1] >> 1	64 Mバイト/2 * 16 = 512 M ビット
32 ビット	HADDR[25:2] >> 2	64 Mバイト/4 * 32 = 512 M ビット

1. 外部メモリが 16 ビット幅の場合、FMC は、内部で HADDR[25:1] を使用して、外部メモリ FMC\_A[24:0] に対するアドレスを生成します。メモリが 32 ビット幅の場合、FMC は、内部で HADDR[25:2] を使用して、外部アドレスを生成します。外部メモリの幅にかかわらず、FMC\_A[0] は外部メモリアドレス A[0] に接続されます。

## 13.4.2 NAND フラッシュメモリアドレスマッピング

NANDバンクは表 48 に示されるメモリ空間に分割されます。

表 48. NAND メモリマッピングおよびタイミングレジスタ

開始アドレス	終了アドレス	FMC バンク	メモリ空間	タイミングレジスタ
0x8800 0000	0x8BFF FFFF	バンク 3 - NAND フラッシュ	属性	FMC_PATT (0x8C)
0x8000 0000	0x83FF FFFF		共通	FMC_PMEM (0x88)

NAND 型フラッシュメモリでは、共通および属性メモリ空間は、下位 256 KB に位置する 3 つのセクションに分割されます (表 49 を参照)。

- データセクション (共通/属性メモリ空間の最初の 64 KB)
- コマンドセクション (共通/属性メモリ空間の 2 番目の 64 KB)
- アドレスセクション (共通/属性メモリ空間の次の 128 KB)

表 49. NAND バンク選択

セクション名	HADDR[17:16]	アドレス範囲
アドレスセクション	1X	0x020000-0x03FFFF
コマンドセクション	01	0x010000-0x01FFFF
データセクション	00	0x000000-0x00FFFF

アプリケーションソフトウェアは、3つのセクションを使用して、NAND型フラッシュメモリにアクセスします。

- **NAND フラッシュメモリにコマンドを送信するときは**、ソフトウェアからコマンドセクションの任意のメモリ位置にコマンドの値を書き込む必要があります。
- **読み出しまたは書き込みを行う必要がある NAND フラッシュのアドレスを指定するには**、ソフトウェアは、アドレスセクションの任意のメモリ位置にアドレス値を書き込む必要があります。アドレスは4または5バイト長なので（実際のメモリサイズに依存）、アドレス全体を指定するには、アドレスセクションへのいくつかの連続した書き込みが必要です。
- **データの読み出しまたは書き込みを行うときは**、ソフトウェアは、データセクションの任意のメモリ位置からデータを読み出すか、書き込む必要があります。

NAND型フラッシュメモリはアドレスを自動的にインクリメントするので、連続したメモリ位置にアクセスするには、データセクションのアドレスをインクリメントする必要はありません。

### 13.4.3 SDRAM アドレスマッピング

HADDR[28] ビット（内部 AHB アドレスライン 28）を使って、2つのメモリバンクから1つを選択します（表 50を参照）。

表 50. SDRAM バンク選択

HADDR[28]	選択されるバンク	制御レジスタ	タイミングレジスタ
0	SDRAM バンク 1	FMC_SDCR1	FMC_SDTR1
1	SDRAM バンク 2	FMC_SDCR2	FMC_SDTR2

次の表に、13ビット行/11ビット列構成に対するSDRAMマッピングを示します。

表 51. SDRAM アドレスマッピング

メモリ幅 <sup>(1)</sup>	内部バンク	行アドレス	列アドレス <sup>(2)</sup>	最大メモリ容量 (Mバイト)
8ビット	HADDR[25:24]	HADDR[23:11]	HADDR[10:0]	64 MB : 4 * 8K * 2K
16ビット	HADDR[26:25]	HADDR[24:12]	HADDR[11:1]	128 MB : 4 * 8K * 2K * 2
32ビット	HADDR[27:26]	HADDR[25:13]	HADDR[12:2]	256 MB : 4 * 8K * 2K * 4

- 16ビットメモリとインタフェースする場合、FMCは内部でHADDR[11:1]内部AHBアドレスラインを使用して外部アドレスを発生します。32ビットメモリとインタフェースする場合、FMCは内部でHADDR[12:2]ラインを使用して外部アドレスを発生します。メモリの幅にかかわらず、FMC\_A[0]は外部メモリアドレスA[0]に接続する必要があります。
- 自動プリチャージはサポートされていません。FMC\_A[10]は外部メモリアドレスA[10]に接続する必要がありますが、常にローレベルに駆動されます。



HADDR[27:0] ビットは、SDRAM コントローラの設定に応じて外部SDRAM アドレスへ変換されます。

- データサイズ : 8、16 または 32 ビット
- 行サイズ : 11、12 または 13 ビット
- 列サイズ : 8、9、10 または 11 ビット
- 内部バンク数 : 2 つまたは 4 つの内部バンク

次の表に、SDRAM アドレスマッピングと SDRAM コントローラ設定の関係を示します。

**表 52.8 ビットデータバス幅での SDRAM アドレスマッピング<sup>(1)(2)</sup>**

行サイズ設定	HADDR (AHB 内部アドレスライン)																											
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11 ビット 行サイズ設定	Res.						BANK [1:0]		ROW [10:0]										COLUMN [7:0]									
	Res.						BANK [1:0]		ROW [10:0]										COLUMN [8:0]									
	Res.						BANK [1:0]		ROW [10:0]										COLUMN [9:0]									
	Res.						BANK [1:0]		ROW [10:0]										COLUMN [10:0]									
12 ビット 行サイズ設定	Res.						BANK [1:0]		ROW [11:0]										COLUMN [7:0]									
	Res.						BANK [1:0]		ROW [11:0]										COLUMN [8:0]									
	Res.						BANK [1:0]		ROW [11:0]										COLUMN [9:0]									
	Res.						BANK [1:0]		ROW [11:0]										COLUMN [10:0]									
13 ビット 行サイズ設定	Res.						BANK [1:0]		ROW [12:0]										COLUMN [7:0]									
	Res.						BANK [1:0]		ROW [12:0]										COLUMN [8:0]									
	Res.						BANK [1:0]		ROW [12:0]										COLUMN [9:0]									
	Res.						BANK [1:0]		ROW [12:0]										COLUMN [10:0]									

1. BANK[1:0] は、バンクアドレス BA[1:0] になります。2 つの内部バンクのみを使用する場合、BA1 は常に"0"に設定する必要があります。
2. 予約済み (Res.) アドレス範囲をアクセスすると、AHB エラーが発生します。

表 53. 16 ビットデータバス幅での SDRAM アドレスマッピング<sup>(1)(2)</sup>

行サイズ設定	HADDR (AHB アドレスライン)																										
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
11 ビット行サイズ 設定	Res.				BANK [1:0]		ROW [10:0]						COLUMN [7:0]				BM0 <sup>(3)</sup>										
	Res.				BANK [1:0]		ROW [10:0]						COLUMN [8:0]				BM0										
	Res.			BANK [1:0]		ROW [10:0]						COLUMN [9:0]				BM0											
	Res.		BANK [1:0]		ROW [10:0]						COLUMN [10:0]				BM0												
12 ビット行サイズ 設定	Res.				BANK [1:0]		ROW [11:0]						COLUMN [7:0]				BM0										
	Res.			BANK [1:0]		ROW [11:0]						COLUMN [8:0]				BM0											
	Res.		BANK [1:0]		ROW [11:0]						COLUMN [9:0]				BM0												
	Res.	BANK [1:0]	ROW [11:0]						COLUMN [10:0]				BM0														
13 ビット行サイズ 設定	Res.				BANK [1:0]		ROW [12:0]						COLUMN [7:0]				BM0										
	Res.			BANK [1:0]		ROW [12:0]						COLUMN [8:0]				BM0											
	Res.		BANK [1:0]		ROW [12:0]						COLUMN [9:0]				BM0												
	Res.	BANK [1:0]	ROW [12:0]						COLUMN [10:0]				BM0														

1. BANK[1:0] は、バンクアドレス BA[1:0] になります。2 つの内部バンクのみを使用する場合、BA1 は常に"0"に設定する必要があります。
2. 予約済み (Res.) アドレス空間をアクセスすると、AHB エラーが発生します。
3. BM0 は、16 ビットアクセスに対するバイトマスクです。

表 54. 32 ビットデータバス幅での SDRAM アドレスマッピング<sup>(1)(2)</sup>

行サイズ設定	HADDR (AHB アドレスライン)																										
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
11 ビット行サイズ 設定	Res.				BANK [1:0]		ROW [10:0]						COLUMN [7:0]				BM[1:0] <sup>(3)</sup>										
	Res.				BANK [1:0]		ROW [10:0]						COLUMN [8:0]				BM[1:0]										
	Res.			BANK [1:0]		ROW [10:0]						COLUMN [9:0]				BM[1:0]											
	Res.		BANK [1:0]		ROW [10:0]						COLUMN [10:0]				BM[1:0]												



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

表 54. 32 ビットデータバス幅での SDRAM アドレスマッピング<sup>(1)(2)</sup> (続き)

行サイズ設定	HADDR (AHB アドレスライン)																															
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
12 ビット行サイズ 設定	Res.		BANK [1:0]		ROW [11:0]														COLUMN [7:0]					BM[1:0]								
	Res.		BANK [1:0]		ROW [11:0]														COLUMN [8:0]					BM[1:0]								
	Res.		BANK [1:0]		ROW [11:0]														COLUMN [9:0]					BM[1:0]								
	Res.		BANK [1:0]		ROW [11:0]														COLUMN [10:0]					BM[1:0]								
13 ビット行サイズ 設定	Res.		BANK [1:0]		ROW [12:0]														COLUMN [7:0]					BM[1:0]								
	Res.		BANK [1:0]		ROW [12:0]														COLUMN [8:0]					BM[1:0]								
	Res.		BANK [1:0]		ROW [12:0]														COLUMN [9:0]					BM[1:0]								
	BANK [1:0]			ROW [12:0]														COLUMN [10:0]					BM[1:0]									

1. BANK[1:0] は、バンクアドレス BA[1:0] になります。2つの内部バンクのみを使用する場合、BA1 は常に"0"に設定する必要があります。
2. 予約済み (Res.) アドレス空間をアクセスすると、AHB エラーが発生します。
3. BM[1:0] は、32 ビットアクセスに対するバイトマスクです。

## 13.5 NOR 型フラッシュ/PSRAM コントローラ

FMC は、以下のタイプのメモリを駆動するのに適した信号タイミングを発生します。

- 非同期 SRAM および ROM
  - 8 ビット
  - 16 ビット
  - 32 ビット
- PSRAM (Cellular RAM)
  - 非同期モード
  - 同期アクセスに対するバーストモード
  - マルチプレクスまたは非マルチプレクス
- NOR フラッシュメモリ
  - 非同期モード
  - 同期アクセスに対するバーストモード
  - マルチプレクスまたは非マルチプレクス

FMC は、バンクごとに 1 つのチップセレクト信号 NE[4:1] を出力します。その他の信号 (アドレス、データ、および制御) はすべて、共有されます。

FMC は、次のようなプログラム可能なタイミングにより、広範囲なデバイスをサポートしています。

- プログラム可能なウェイトステート (15 まで)
- プログラム可能なバスターンアラウンドサイクル (15 まで)
- プログラム可能な出カインーブルおよび書き込みインーブル遅延 (15 まで)
- 独立した読み出しおよび書き込みタイミングとプロトコルにより、広範囲なメモリおよびタイミングをサポート
- プログラム可能な連続クロック (FMC\_CLK) 出力

FMC クロック (FMC\_CLK) は HCLK クロックの約数。このクロックは、同期アクセス中にのみ、または FMC\_BCR1 レジスタの CCKEN ビット設定に応じて非同期アクセスおよび同期アクセス中に、選択した外部デバイスへ供給することができます。

- CCLKEN ビットがリセットされている場合、FMC は同期アクセス (読み出し/書き込みトランザクション) 中にのみクロック (CLK) を生成します。
- CCLKEN ビットがセットされている場合、FMC は非同期アクセス中および同期アクセス中に連続クロックを生成します。FMC\_CLK 連続クロックを生成するためには、バンク 1 を同期モードに設定する必要があります (セクション 13.5.6 : NOR/PSRAM コントローラレジスタを参照)。すべての同期メモリに対して同じクロックが使用されるため、連続出カクロックを生成して同期アクセスを実行する場合、AHB データサイズはメモリデータ幅 (MWID) に一致する必要があります。そうしないと、AHB データトランザクションに応じて、FMC\_CLK 周波数が変化します (FMC\_CLK 分周比の式については、セクション 13.5.5 : 同期トランザクションを参照)。

各バンクのサイズは 64 MB に固定されます。各バンクは、専用レジスタによって設定されます (セクション 13.5.6 : NOR/PSRAM コントローラレジスタを参照)。

プログラム可能なメモリパラメータは、アクセスタイミング (表 55 を参照) を含み、ウェイト管理をサポートします (バーストモードでの PSRAM および NOR フラッシュへのアクセス)。

表 55. プログラム可能な NOR/PSRAM のアクセスパラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
アドレスセットアップ	アドレスセットアップフェーズ時間	非同期	AHB クロック サイクル (HCLK)	0	15
アドレスホールド	アドレスホールドフェーズ時間	非同期、マルチプレクス I/O	AHB クロック サイクル (HCLK)	1	15
データセットアップ	データセットアップフェーズ時間	非同期	AHB クロック サイクル (HCLK)	1	256
バスターン	バスターンアラウンドフェーズ時間	非同期および同期読み出し/書き込み	AHB クロック サイクル (HCLK)	0	15
クロック分周比	1つのメモリクロックサイクル (CLK) に必要な AHB クロック サイクル (HCLK) 数	同期	AHB クロック サイクル (HCLK)	2	16
データ遅延	バーストの最初のデータの前にメモリに発行されるクロックサイクル数	同期	メモリクロック サイクル (CLK)	2	17

## 13.5.1 外部メモリインタフェース信号

表 56、表 57 および 表 58 に、NOR フラッシュメモリ、SRAM、および PSRAM へのインタフェースで一般に使用される信号を示します。

注： 接頭辞「N」は、アクティブローの信号を表します。

### NOR フラッシュメモリ、非マルチプレクス I/O

表 56. 非マルチプレクス I/O NOR フラッシュメモリ

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:0]	O	アドレスバス
D[31:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書き込みインーブル
NL(= NADV)	O	ラッチインーブル (この信号は、一部の NOR 型フラッシュデバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FMC への NOR フラッシュウェイト入力信号

最大容量は、512 M ビット (26 アドレスライン) です。

### NOR フラッシュメモリ、16 ビットマルチプレクス I/O

表 57. 16 ビットマルチプレクス I/O NOR フラッシュメモリ

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書き込みインーブル
NL(= NADV)	O	ラッチインーブル (この信号は、一部の NOR 型フラッシュデバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FMC への NOR フラッシュウェイト入力信号

最大容量は 512 M ビットです。

## PSRAM/SRAM、非マルチプレクス I/O

表 58. 非マルチプレクス I/O PSRAM/SRAM

FMC 信号名	I/O	機能
CLK	O	クロック (PSRAM 同期アクセス専用)
A[25:0]	O	アドレスバス
D[31:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (Cellular RAM すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインーブル
NWE	O	書き込みインーブル
NL(= NADV)	O	PSRAM 入力の場合のみアドレス有効 (メモリ信号名 : NADV)
NWAIT	I	FMC への PSRAM ウェイト入力信号
NBL[3:0]	O	バイトレーン出カバイト 0~3 制御 (上位および下位バイトインーブル)

最大容量は 512 M ビットです。

## PSRAM、16 ビットマルチプレクス I/O

表 59. 16 ビットマルチプレクス I/O PSRAM

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (Cellular RAM すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインーブル
NWE	O	書き込みインーブル
NL(= NADV)	O	アドレスパリティ PSRAM 入力 (メモリ信号名 : NADV)
NWAIT	I	FMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出カバイト 0 およびバイト 1 制御 (上位および下位バイトインーブル)

最大容量は、512 M ビット (26 アドレスライン) です。

## 13.5.2 サポートされるメモリおよびトランザクション

以下の表 60 に、NOR フラッシュメモリ、PSRAM、SRAM に対してメモリデータバスが 16 ビット幅の場合にサポートされているデバイス、アクセスモード、トランザクションの例を示します。FMC では許容されない（すなわち非サポート）トランザクションは、この例で灰色表示しています。

表 60. NOR フラッシュ/PSRAM : サポートされているメモリおよびトランザクションの例

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	許可/禁止	コメント
NOR 型 フラッシュ (マルチプレクス I/O と 非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割
	非同期ページ	R	-	16	N	モードはサポートされていません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-
PSRAM (マルチプレクス I/O および 非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割
	非同期ページ	R	-	16	N	モードはサポートされていません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-
	同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	同期	W	16/32	16	Y	-
SRAM と ROM	非同期	R	8/16	16	Y	-
	非同期	W	8/16	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割、 バイトレーン NBL[1:0] の使用

## 13.5.3 一般的なタイミング規則

### 信号の同期

- すべてのコントローラ出力信号は、内部クロック (HCLK) の立ち上がりエッジで変化します。
- 同期モード (読み出しまたは書き込み) では、すべての出力信号が HCLK の立ち上がりエッジで変化します。CLKDIV の値によらず、すべての信号は次のように変化します。
  - NOEL/NWEL/ NEL/NADVL/ NADVH/ NBL/ アドレスの有効出力は、FMC\_CLK クロックの立ち下がりエッジで変化します。
  - NOEH/ NWEH/ NEH/ NOEH/NBLH/ アドレスの無効出力は、FMC\_CLK クロックの立ち上がりエッジで変化します。

## 13.5.4 NOR フラッシュ /PSRAM コントローラ非同期トランザクション

### 非同期スタティックメモリ (NOR フラッシュ、PSRAM、SRAM)

- 信号は、内部クロック HCLK によって同期されます。このクロックはメモリには発行されません。
- FMC は、チップセレクト信号 NE をネゲートする前に、常にデータをサンプリングします。これにより、メモリのデータホールドタイミングに関する制約が満たされます (チップイネーブルハイからデータ遷移までの最小時間は通常 0 ns です)。
- 拡張モードを有効にした場合 (FMC\_BCRx レジスタの EXTMOD ビットをセット)、最大 4 つの拡張モード (A、B、C、D) が使用できます。書き込み動作および読み出し動作で、A、B、C、D モードをミックスすることができます。例えば、読み出し動作をモード A で実行し、書き込み動作をモード B で実行することができます。
- 拡張モードを無効にした場合 (FMC\_BCRx レジスタで EXTMOD ビットをリセット)、FMC は次のようにモード 1 またはモード 2 で動作することができます。
  - SRAM/PSRAM メモリタイプを選択した場合 (FMC\_BCRx レジスタで MTYP = 0x0 または 0x01)、モード 1 がデフォルトモードです。
  - NOR メモリタイプを選択した場合 (FMC\_BCRx レジスタで MTYP = 0x10)、モード 2 がデフォルトモードです。

## モード 1 - SRAM/PSRAM (CRAM)

次の図に、サポートされているモードに対する読み出しトランザクションと書き込みトランザクションを示します。それに続いては、FMC\_BCRx レジスタ、および FMC\_BTRx/FMC\_BWTRx レジスタの必要とされる設定を示します。

図 34. モード 1 読み出しアクセス波形

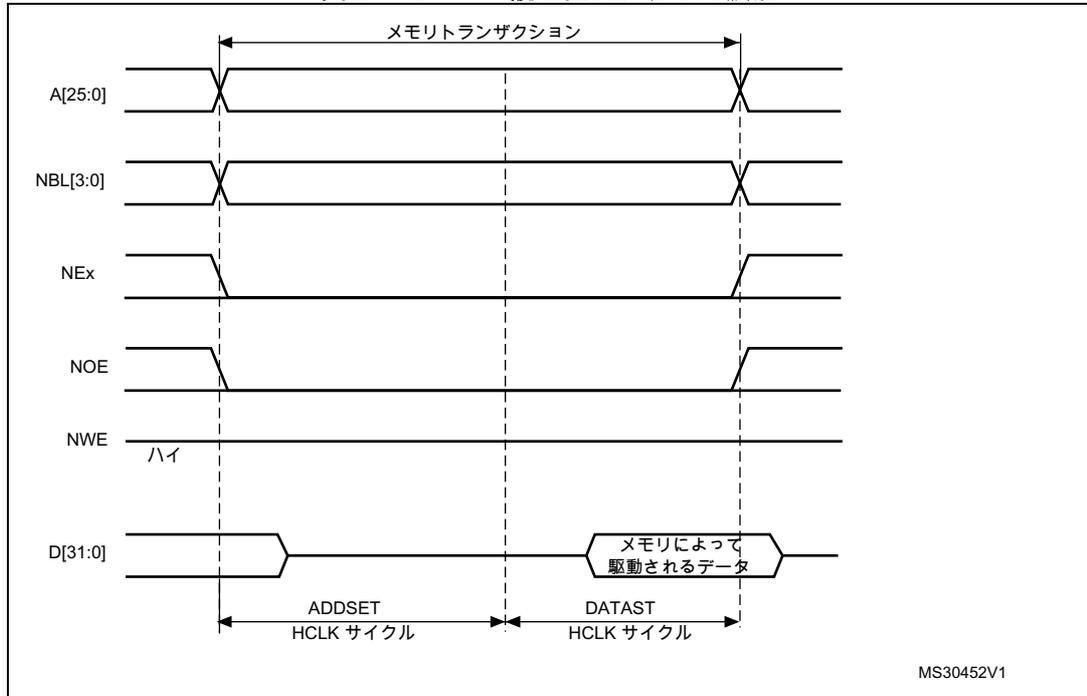
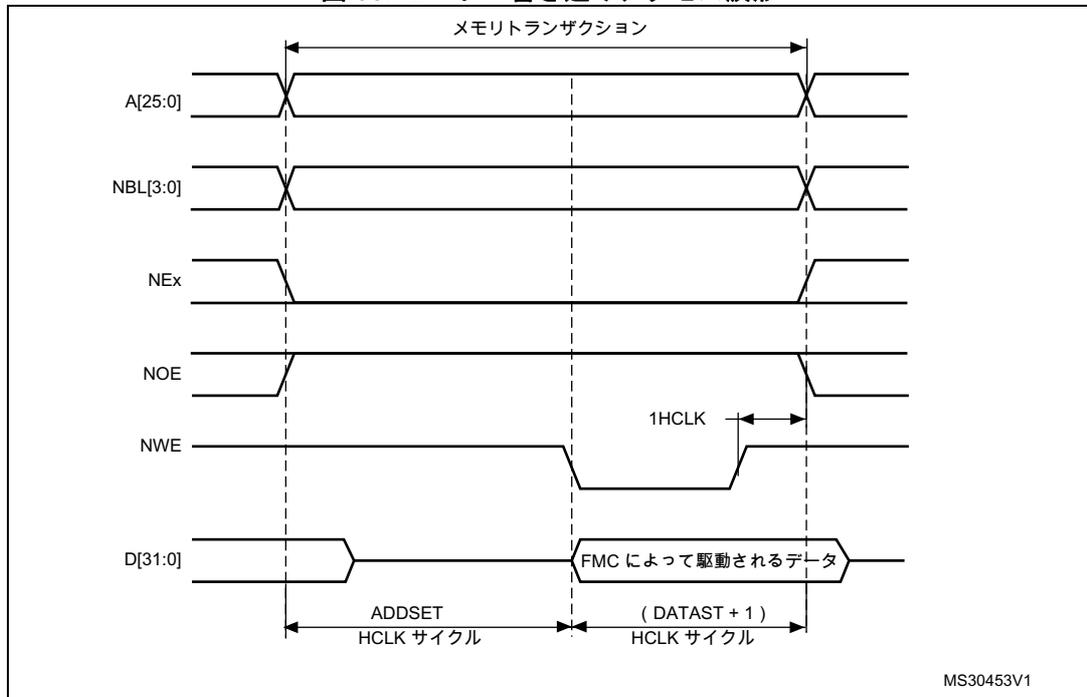


図 35. モード 1 書き込みアクセス波形



書き込みトランザクションの最後の 1 HCLK サイクルは、NWE の立ち上がりエッジ後のアドレスおよびデータホールド時間の保証に役立ちます。この HCLK サイクルがあるため、DATAST の値はゼロより大きくなければなりません (DATAST > 0)。

表 61. FMC\_BCRx ビットフィールド

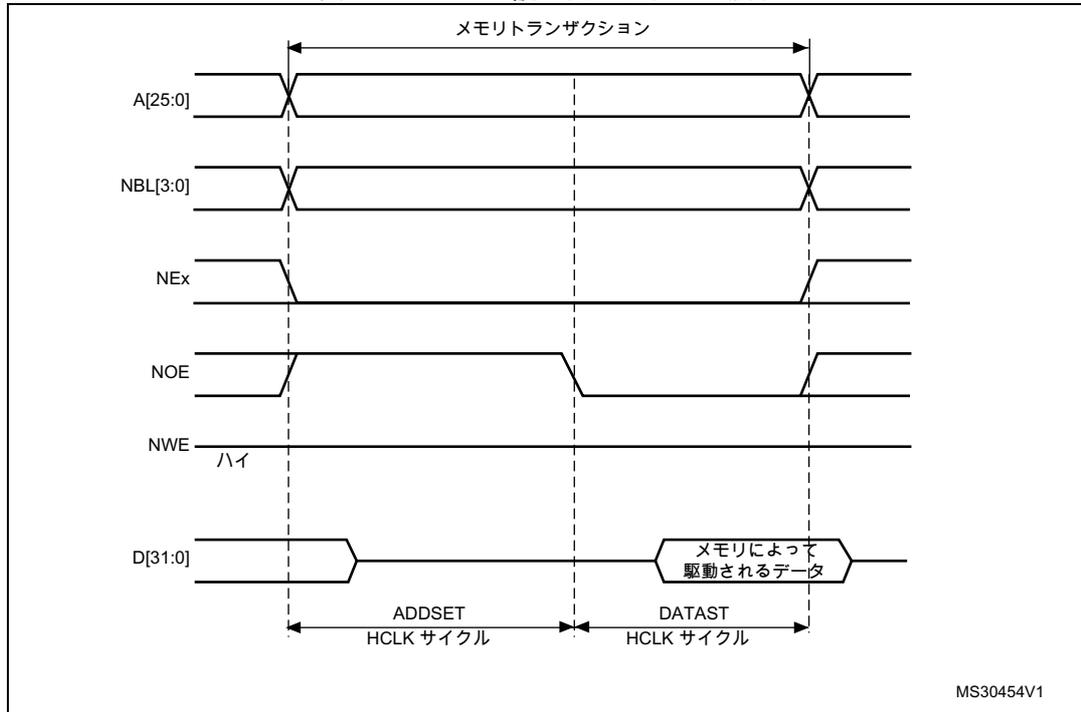
ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	予約済み	0x0
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5-4	MWID	必要に応じて設定します。
3-2	MTYP	必要に応じて、0x2 (NOR フラッシュメモリ) を除きます。
1	MUXE	0x0
0	MBKEN	0x1

表 62. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	無視
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	2 番目のアクセスフェーズの時間 (書き込みアクセスの場合は DATAST + 1 HCLK サイクル、読み出しアクセスの場合は DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

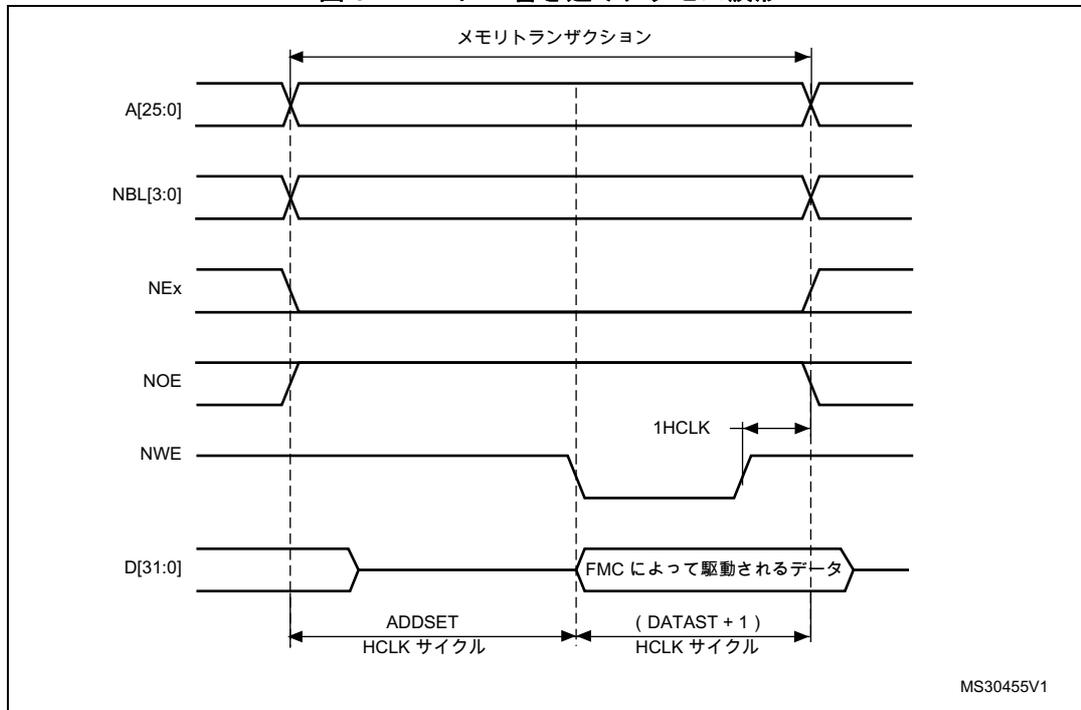
## モード A - SRAM/PSRAM (CRAM) OE トグル

図 36. モード A 読み出しアクセス波形



1. NBL[3:0] は読み出しアクセス中ローレベルに駆動されます。

図 37. モード A 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NOE のトグリングと、独立した読み出しおよび書き込みのタイミングです。

表 63. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
11	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5-4	MWID	必要に応じて設定します。
3-2	MTYP	必要に応じて、0x2 (NOR フラッシュメモリ) を除きます。
1	MUXEN	0x0
0	MBKEN	0x1

表 64. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 65. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

## モード 2/B - NOR 型フラッシュ

図 38. モード 2 およびモード B 読み出しアクセス波形

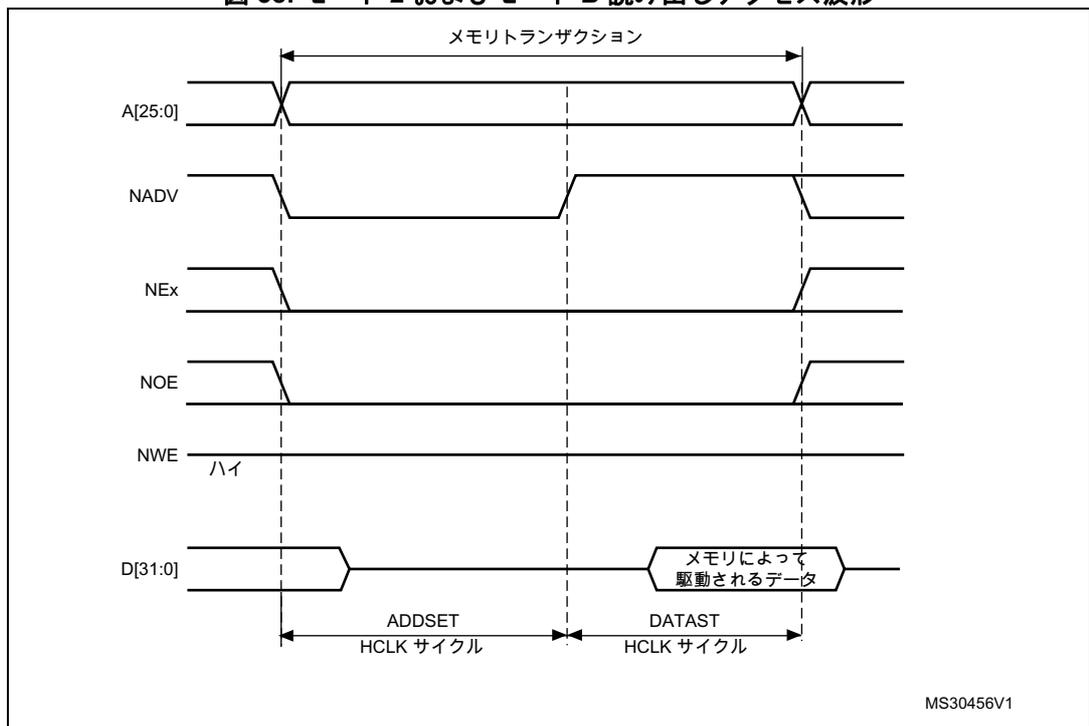


図 39. モード 2 書き込みアクセス波形

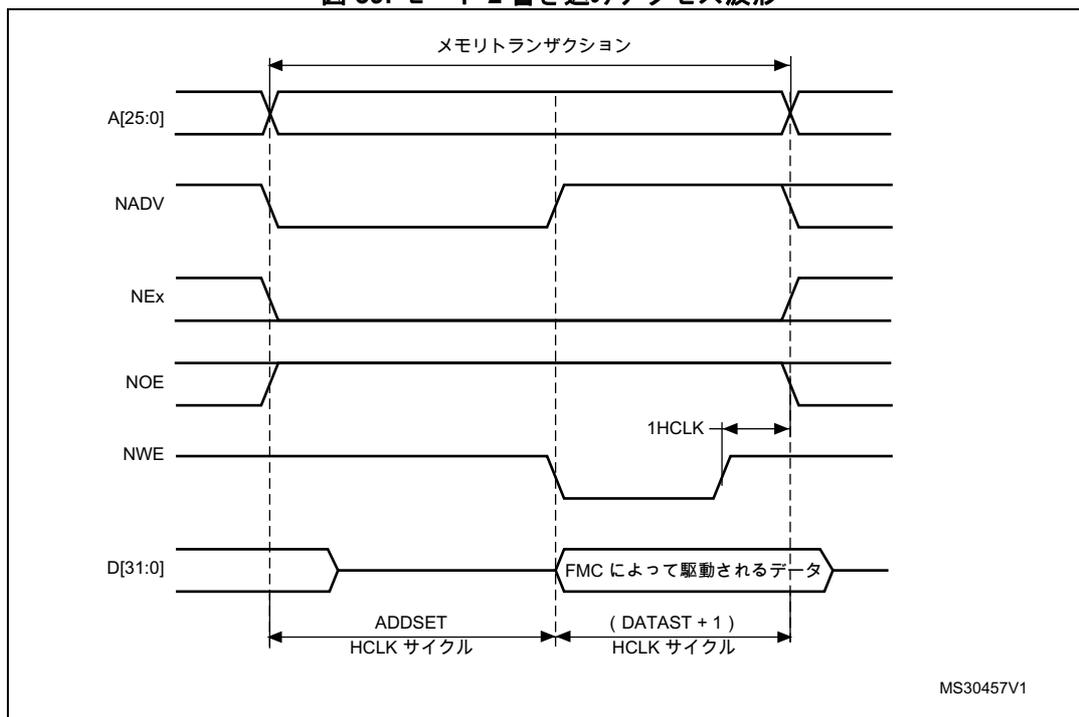
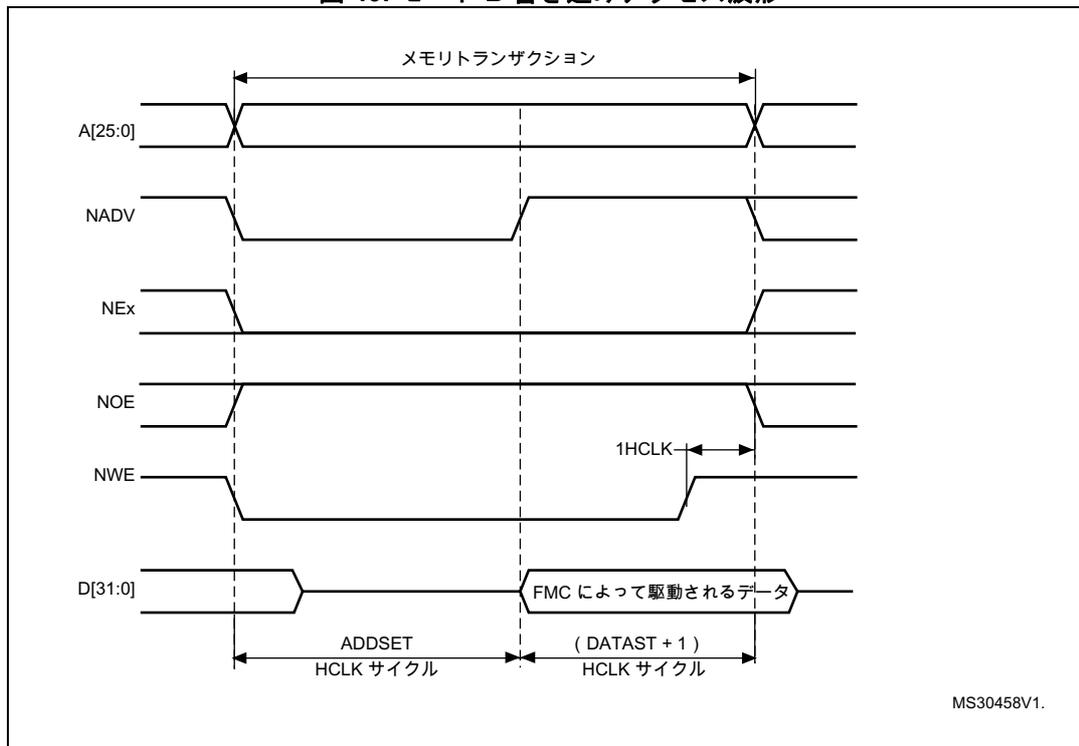


図 40. モード B 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NWE のトグルングと拡張モードがセットされたとき（モード B）の独立した読み出しおよび書き込みのタイミングです。

表 66. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	モード B では 0x1、モード 2 では 0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x2 (NOR 型フラッシュメモリ)
1	MUXEN	0x0
0	MBKEN	0x1

表 67. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	拡張モードがセットされている場合は 0x1
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 68. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	拡張モードがセットされている場合は 0x1
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NE <sub>x</sub> ハイから NE <sub>x</sub> ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

注: FMC\_BWTRx レジスタは、拡張モードがセットされている場合 (モード B) のみ有効です。それ以外の場合には、その内容は意味を持ちません。

### モード C - NOR フラッシュ - OE トグリング

図 41. モード C 読み出しアクセス波形

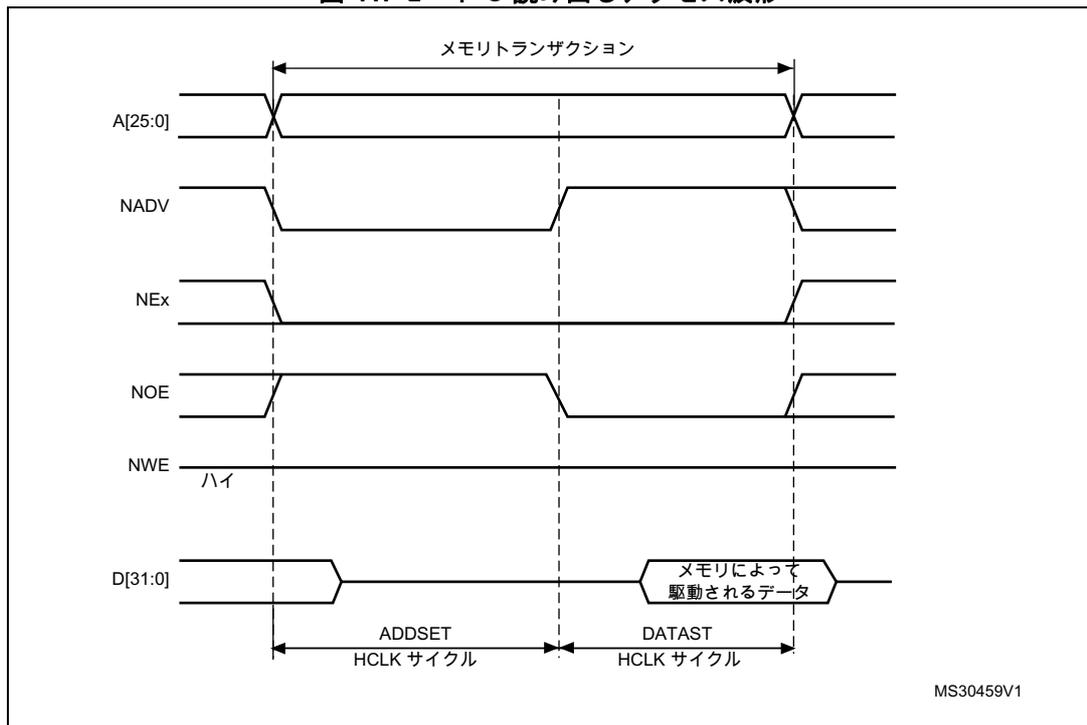
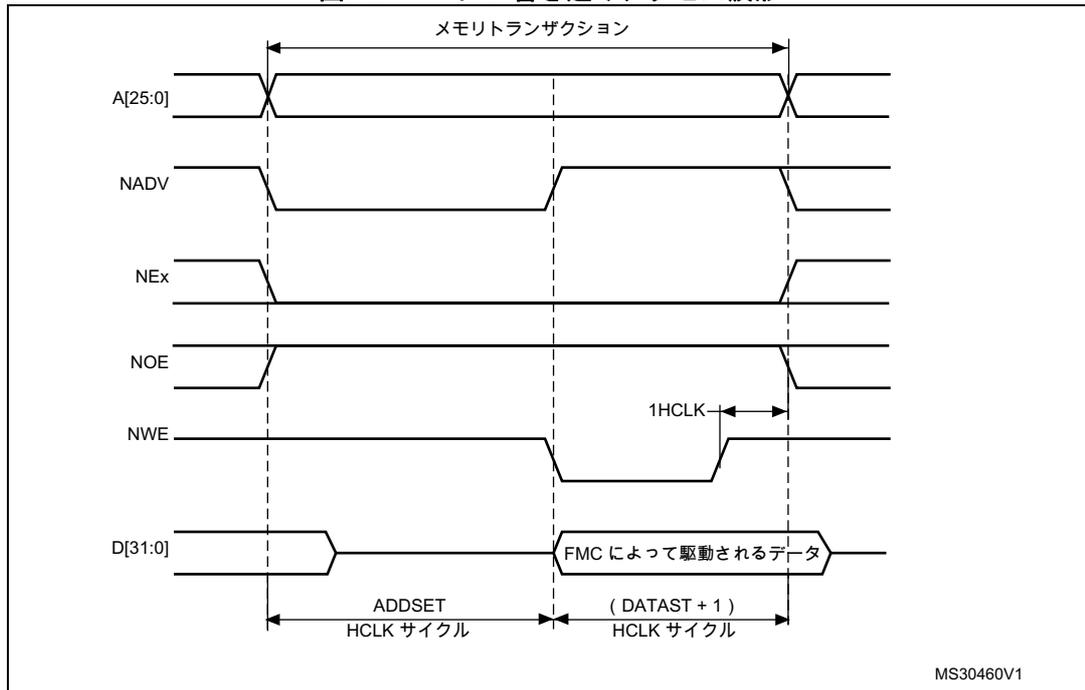


図 42. モード C 書き込みアクセス波形



MS30460V1

モード 1 と比較した場合の相違点は、NOE のトグルリングと、独立した読み出しおよび書き込みのタイミングです。

表 69. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x02 (NOR 型フラッシュメモリ)

表 69. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
1	MUXEN	0x0
0	MBKEN	0x1

表 70. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	0x0
23-20	CLKDIV	0x0
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 71. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

## モード D - 拡張アドレスによる非同期アクセス

図 43. モード D 読み出しアクセス波形

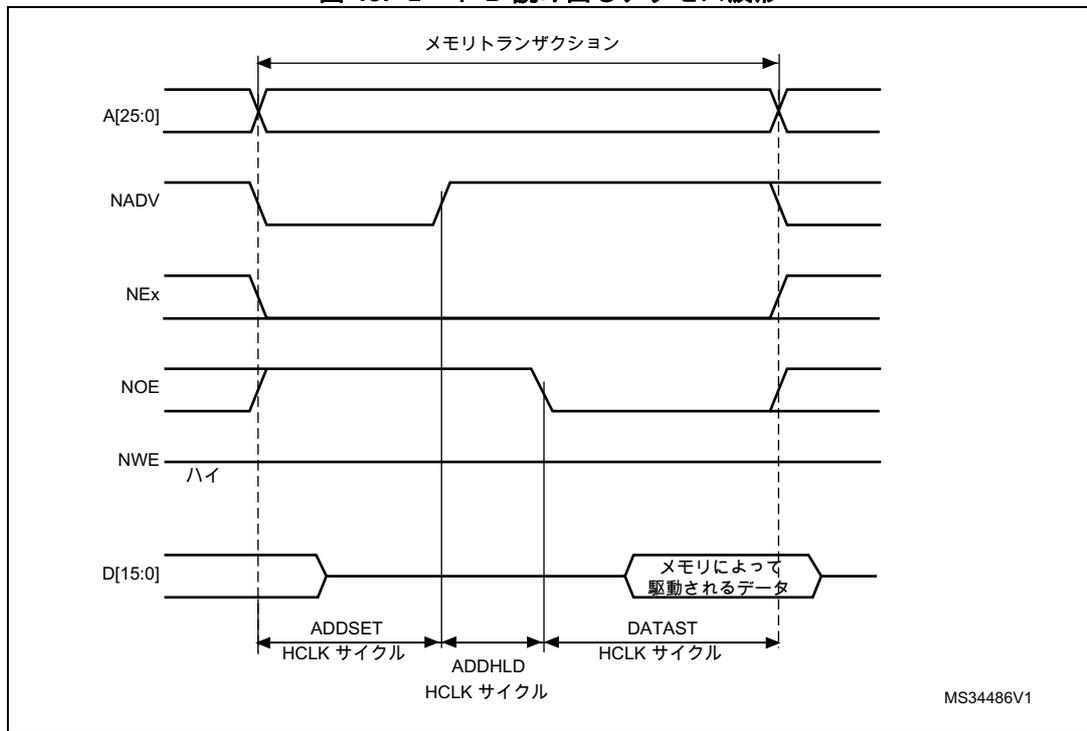
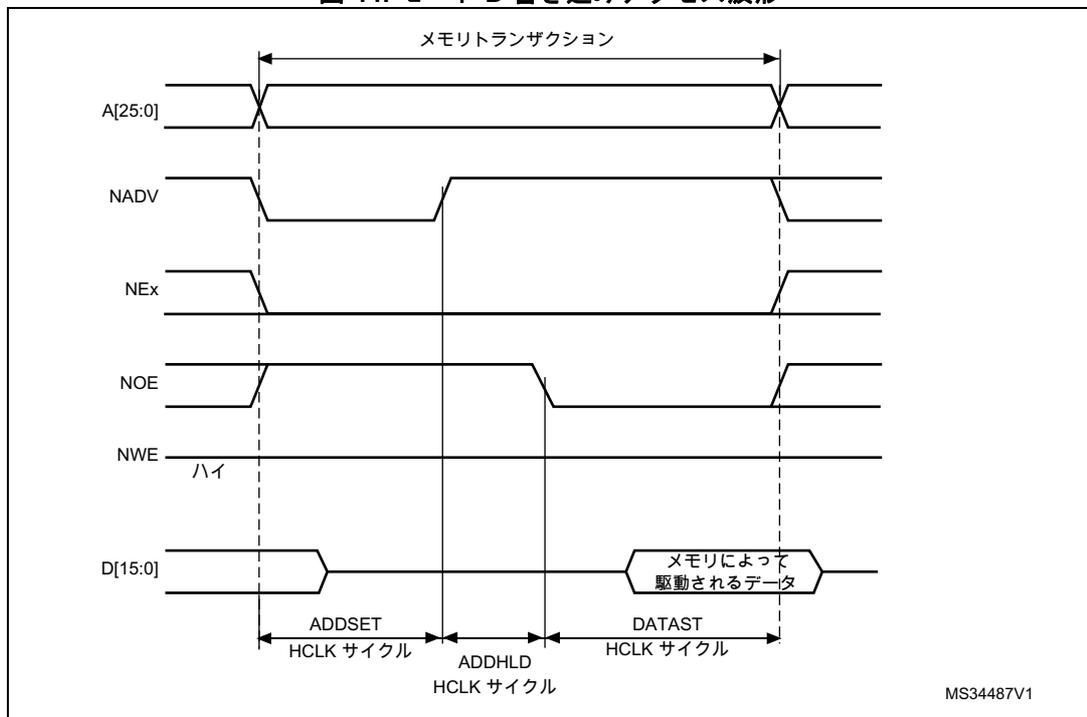


図 44. モード D 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NADV が変化した後にトグルを続ける NOE のトグル と独立した読み出し書き込みタイミングです。

表 72. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5-4	MWID	必要に応じて設定します。
3-2	MTYP	必要に応じて設定します。
1	MUXEN	0x0
0	MBKEN	0x1

表 73. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	読み出しにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

表 74. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29:28	ACCMOD	0x3
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15:8	DATAST	書き込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST1 HCLK サイクル)
7:4	ADDHLD	書き込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3:0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

## マルチプレクスモード - NOR 型フラッシュメモリに対するマルチプレクス非同期アクセス

図 45. マルチプレクス読み出しアクセス波形

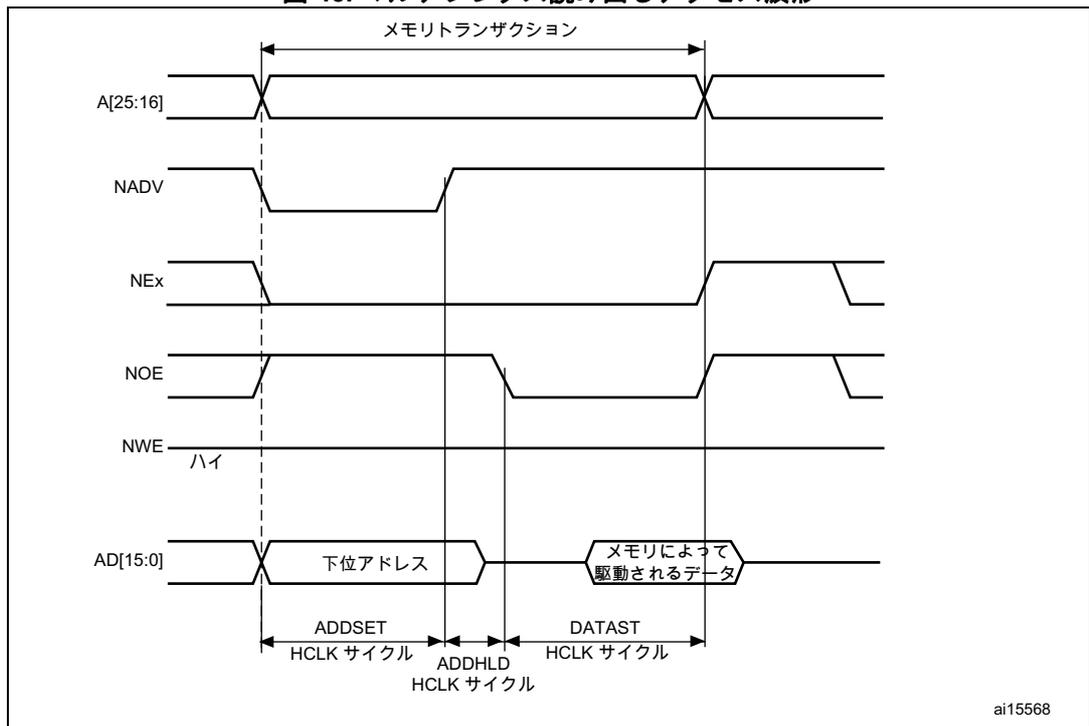
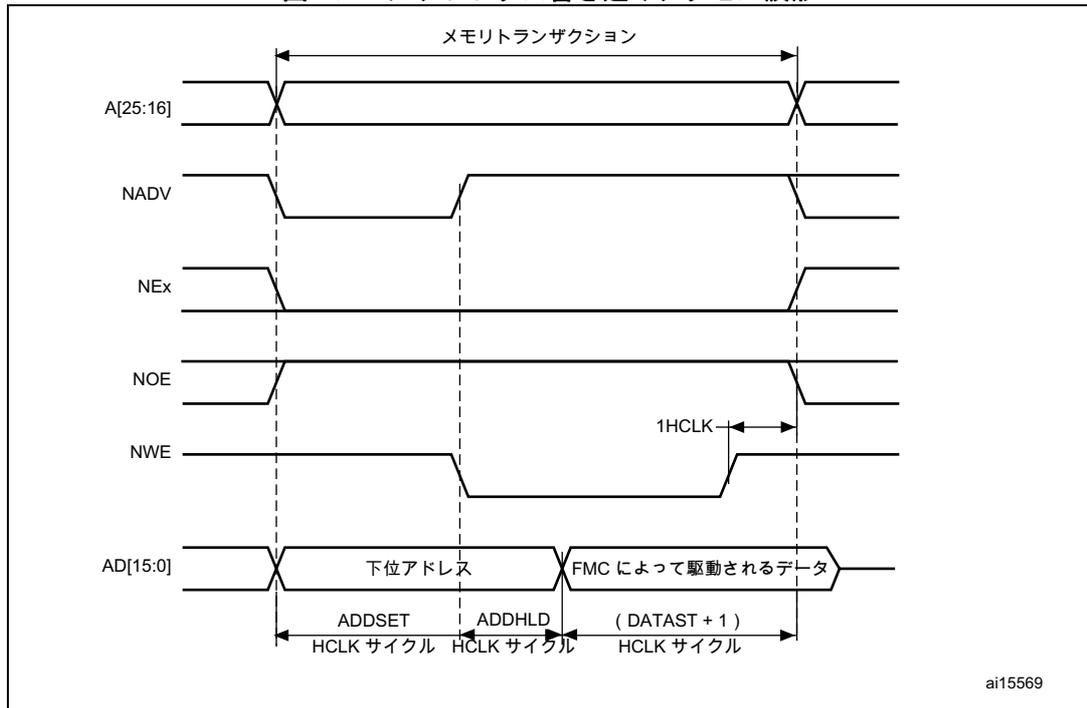


図 46. マルチプレクス書き込みアクセス波形



モード D との相違点は、データバス上の下位アドレスバイトのドライブです。

表 75. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	予約済み	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x2 (NOR 型フラッシュメモリ)

表 75. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
1	MUXEN	0x1
0	MBKEN	0x1

表 76. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	2 番目のアクセスフェーズの時間 (読み出しアクセスの場合は DATAST HCLK サイクル、書き込みアクセスの場合は DATAST+1 HCLK サイクル)。
7-4	ADDHLD	書き込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)。
3-0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

## 非同期アクセスにおけるウェイト管理

非同期メモリが、データの受け取りまたは提供の準備がまだできていないことを知らせるために、WAIT 信号をアサートした場合、FMC\_BCRx レジスタの ASYNCWAIT ビットをセットする必要があります。

WAIT 信号がアクティブな場合 (WAITPOL ビットに依存して、ハイまたはロー)、DATAST ビットによってプログラムされる 2 番目のアクセスフェーズ (データセットアップフェーズ) は、WAIT ビットが非アクティブになるまで延長されます。データセットアップフェーズとは異なり、ADDSET および ADDHLD ビットによってプログラムされる最初のアクセスフェーズ (アドレスセットアップおよびアドレスホールドフェーズ) は、WAIT に対応しないので、延長されません。

データセットアップフェーズは、メモリトランザクションが終了する 4 HCLK サイクル前に WAIT が検出されるようにプログラムする必要があります。次のようなケースを考慮する必要があります。

1. メモリが、次のようにトグルする NOE/NWE に合わせた WAIT 信号をアサートする場合。

$$\text{DATAST} \geq (4 \times \text{HCLK}) + \text{max\_wait\_assertion\_time}$$

2. メモリが、NEx (またはトグルしない NOE/NWE) に合わせた WAIT 信号をアサートする場合。次の条件が成立するならば、

$$\text{max\_wait\_assertion\_time} > \text{address\_phase} + \text{hold\_phase}$$

以下の通りです。

$$\text{DATAST} \geq (4 \times \text{HCLK}) + (\text{max\_wait\_assertion\_time} - \text{address\_phase} - \text{hold\_phase})$$

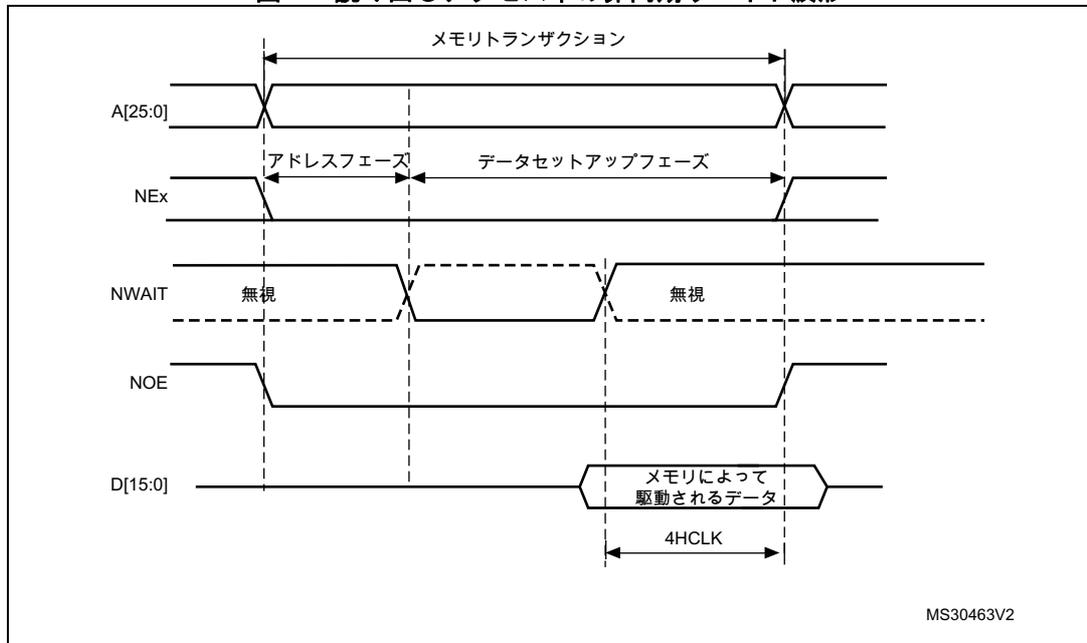
それ以外の場合は、

$$\text{DATAST} \geq 4 \times \text{HCLK}$$

ここで、max\_wait\_assertion\_time は、NEx/NOE/NWE がローになったときにメモリが WAIT 信号をアサートするために必要な最大時間です。

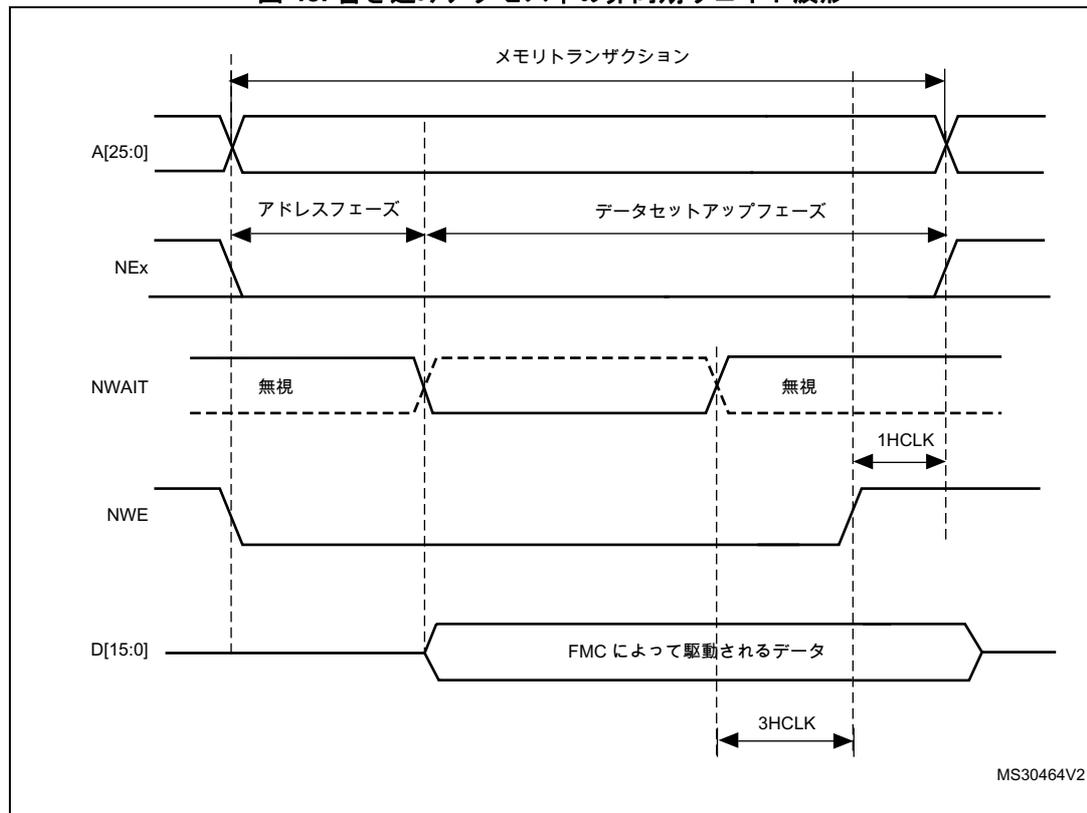
図 47 と 図 48 に、非同期メモリによって WAIT が解除された後、メモリアクセスフェーズに追加される HCLK クロックのサイクル数を示します (上記のケースに無関係)。

図 47. 読み出しアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FMC\_BCRx レジスタの WAITPOL ビット設定値に依存します。

図 48. 書き込みアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FMC\_BCRx レジスタの WAITPOL ビット設定値に依存します。

## 13.5.5 同期トランザクション

メモリクロック FMC\_CLK は、HCLK の約数です。この値は CLKDIV 値と MWID/ AHB データサイズに依存し、次式で与えられます。

$$\text{FMC\_CLK divider ratio} = \max(\text{CLKDIV} + 1, \text{MWID}(\text{AHB data size}))$$

MWID が 16 ビットまたは 8 ビットの場合、FMC\_CLK の分周比は常に CLKDIV のプログラムされた値で決まります。

MWID が 32 ビットの場合、FMC\_CLK の分周比は AHB データサイズにも依存します。

例 :

- CLKDIV = 1、MWID = 32ビット、AHB データサイズ = 8 ビットの場合、FMC\_CLK = HCLK / 4
- CLKDIV = 1、MWID = 16ビット、AHB データサイズ = 8 ビットの場合、FMC\_CLK = HCLK / 2

NOR 型フラッシュメモリは、NADV のアサーションから CLK がハイになるまでの最小時間を指定します。この制約を満たすために、FMC は、同期アクセスの最初の内部クロックサイクルでは (NADV アサーションの前)、クロックをメモリに供給しません。これにより、メモリクロックの立ち上がりエッジは、NADV ローパルスの中で発生します。

## データ遅延と NOR メモリ遅延

データ遅延は、データをサンプリングする前のウェイトサイクル数です。DATLAT の値は、NOR 型フラッシュ設定レジスタで指定された遅延の値と一致しなければなりません。FMC は、NADV がローのときのクロックサイクルをデータ遅延カウントに含めません。

**注意：** 一部の NOR フラッシュメモリは、データ遅延カウントに NADV ローサイクルを含めるので、NOR フラッシュの遅延と FMC DATLAT パラメータの厳密な関係は、次のいずれかです。

- NOR 型フラッシュ遅延 = (DATLAT + 2) CLK クロックサイクル、または
- NOR 型フラッシュ遅延 = (DATLAT + 3) CLK クロックサイクル

最近のメモリの中には、遅延フェーズにおいて NWAIT をアサートするものがあります。そのような場合には、DATLAT を最小値に設定することができます。結果として、FMC はデータをサンプリングして、データが有効であるかどうか評価するのに十分な時間、ウェイトします。このように、FMC は、メモリが遅延から抜けたことを検出し、実際のデータが処理されます。

その他のメモリは、遅延中には NWAIT をアサートしません。この場合、FMC とメモリの両方について、遅延が正しく設定されなければなりません。そうしないと、正しいデータの代わりに無効なデータが取られたり、有効なデータがメモリサイクルの初期フェーズで失われることになります。

## シングルバースト転送

選択されたバンクが同期アクセスのバーストモードに設定されている場合、例えば、16 ビットメモリで AHB シングルバーストトランザクションがリクエストされると、FMC は長さ 1 (AHB 転送が 16 ビットの場合) または 2 (AHB 転送が 32 ビットの場合) のバーストトランザクションを実行して、最後のデータがストローブされると、チップセレクトをデアサートします。

このような転送はサイクル数の点で、非同期読み出し動作に比較して最も効率的ではありません。しかし、時間がかかるメモリアクセスモードの再プログラムには、まず、ランダム非同期アクセスが必要です。

## Cellular RAM 1.5 ページ境界跨ぎ

Cellular RAM 1.5 では、ページ境界を跨ぐバーストアクセスは許可されていません。FMC コントローラは、メモリページサイズに応じて FMC\_BCR1 レジスタの CPSIZE ビットを設定することでそのメモリページサイズに達した時点で、バーストアクセスを自動的に分割することができます。

## ウェイト管理

同期 NOR フラッシュメモリでは、NWAIT がプログラムされた遅延期間 ((DATLAT+2) CLK サイクルに対応) 後に評価されます。

NWAIT がアクティブの場合 (WAITPOL = 0 ではローレベル、WAITPOL = 1 ではハイレベル)、ウェイトステートは NWAIT が非アクティブ (WAITPOL = 0 ではハイレベル、WAITPOL = 1 ではローレベル) になるまで挿入されます。

NWAIT が非アクティブのときには、データはただちに (ビット WAITCFG=1) または次のクロックエッジで (ビット WAITCFG=0)、有効とみなされます。

NWAIT 信号によるウェイトステートの挿入時には、コントローラはメモリへのクロックパルスの送信を続け、チップセレクトと出カインエーブル信号を維持します。データは有効と見なしません。

バーストモードでの NOR フラッシュの NWAIT 信号には、2 つのタイミング設定があります。

- フラッシュメモリは、ウェイトステートの 1 データサイクル前に NWAIT 信号をアサートします (リセット後のデフォルト)。
- フラッシュメモリは、ウェイトステートの間に NWAIT 信号をアサートします。

FMC は FMC\_BCRx レジスタ (x = 0..3) の WAITCFG ビットを使って、各チップセレクトに対する両 NOR フラッシュウェイトステート設定をサポートします。

図 49. 待ち設定波形

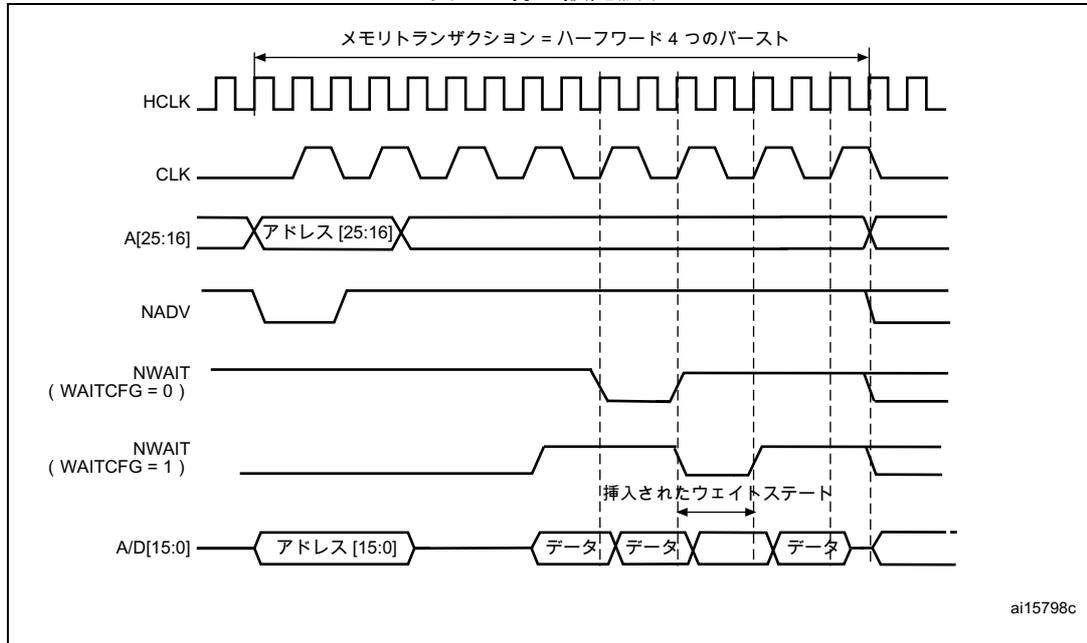
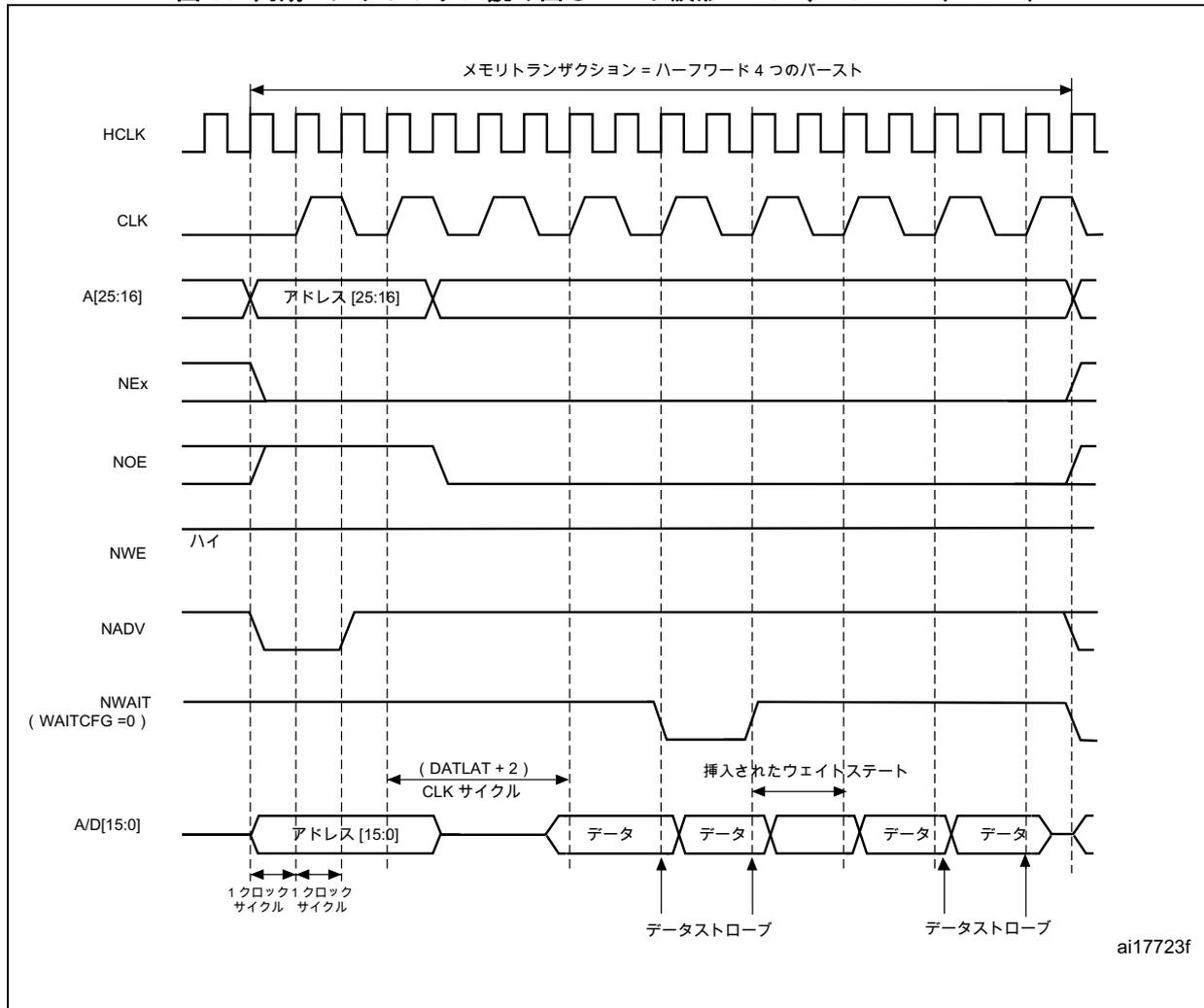


図 50. 同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)



1. バイトレーン出力 (NBL は示されていません。NOR アクセス時にはハイに保たれ、PSRAM (CRAM) アクセス時にはローに保たれます。)

表 77. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	同期読み出し時は影響しません。
18:16	CPSIZE	0x0 (非同期モードで無効)
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。
12	WREN	同期読み出し時は影響しません。

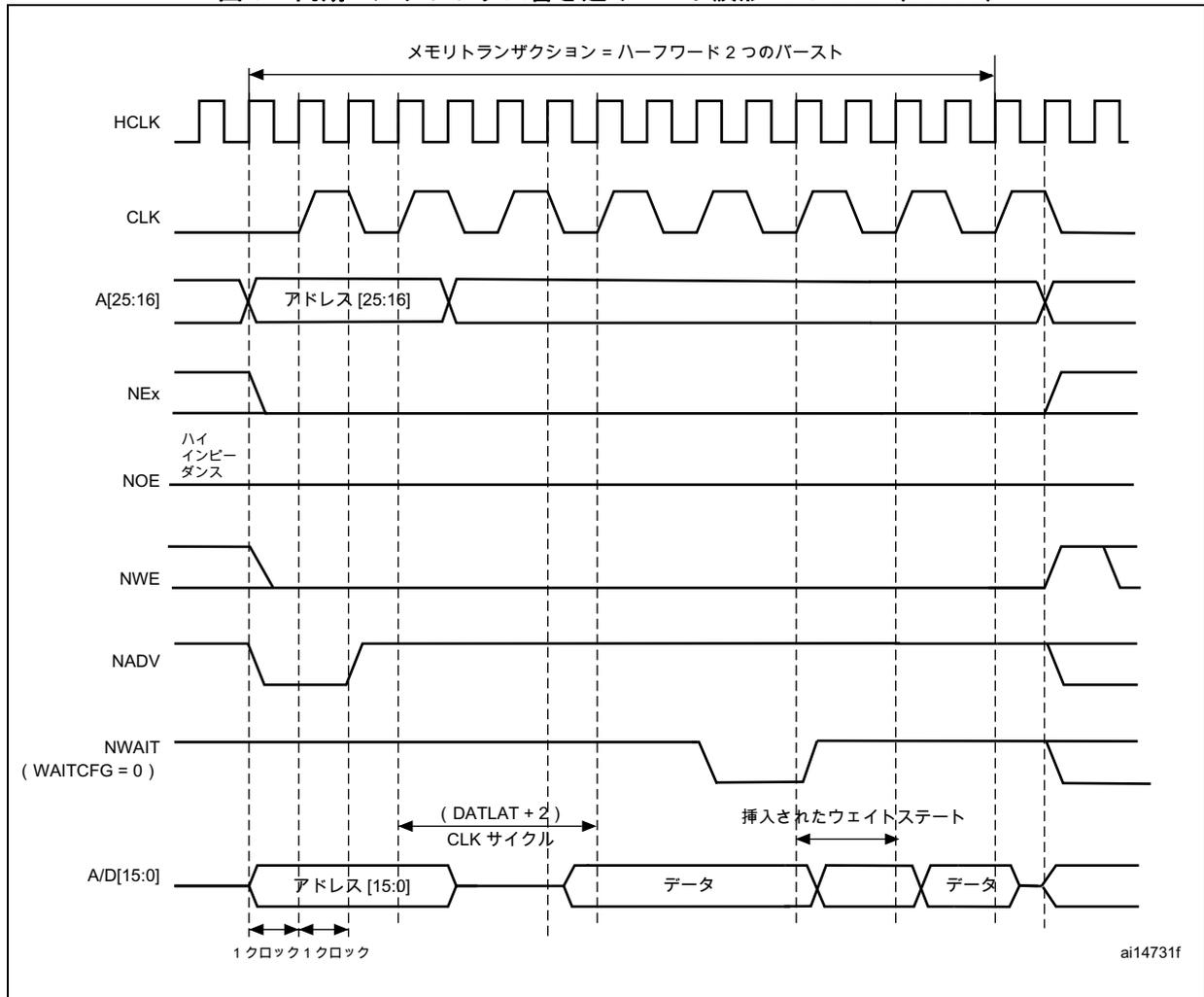
表 77. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
11	WAITCFG	メモリに応じて設定します。
10	予約済み	0x0
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	0x1
7	予約済み	0x1
6	FACCEN	メモリサポートに従って設定します (NOR 型フラッシュメモリ)
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x1 または 0x2
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 78. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	データ遅延
27-24	DATLAT	データ遅延
23-20	CLKDIV	0x0 の場合、CLK = HCLK となります (サポートされません)。 0x1 の場合、CLK = 2×HCLK となります。 ..
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	無視
7-4	ADDHLD	無視
3-0	ADDSET	無視

図 51. 同期マルチプレクス書き込みモード波形 - PSRAM (CRAM)



1. メモリは、NWAIT 信号を 1 サイクル前に発行しなければならないので、WAITCFG を 0 にプログラムする必要があります。
2. バイトレーン (NBL) 出力は、図に示されておらず、NEx がアクティブな間はローに保たれます。

表 79. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-22	予約済み	0x000
21	WFDIS	必要に応じて設定します。
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x1
18:16	CPSIZE	必要に応じて (CRAM 1.5 は 0x1)
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。
12	WREN	0x1

表 79. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
11	WAITCFG	0x0
10	予約済み	0x0
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	同期読み出し時は影響しません。
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x1
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 80. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	データ遅延
23-20	CLKDIV	0x0 の場合、CLK = HCLK となります (サポートされません)。 0x1 の場合、CLK = 2×HCLK となります。
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	無視
7-4	ADDHLD	無視
3-0	ADDSET	無視

## 13.5.6 NOR/PSRAM コントローラレジスタ

### SRAM/NOR 型フラッシュチップセレクト制御レジスタ 1.4 (FMC\_BCR1.4)

アドレスオフセット :  $8 * (x - 1)$ ,  $x = 1..4$

リセット値 : バンク 1 は 0x0000 30DB、バンク 2~4 は 0x0000 30D2

このレジスタは、SRAM、PSRAM、および NOR 型フラッシュメモリで使用される各メモリバンクの制御情報を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WFDIS	CCLK EN	CBURST RW	CPSIZE[2:0]		
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNC WAIT	EXT MOD	WAIT EN	WREN	WAIT CFG	Res.	WAIT POL	BURSTEN	Res.	FACC EN	MWID		MTYP		MUX EN	MBK EN
rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31 : 22 予約済みであり、リセット値のままにしておかなければなりません。

#### ビット 21 WFDIS : 書き込み FIFO 無効化

このビットは、FMC コントローラで使用される書き込み FIFO を無効にします。

- 0 : 書き込み FIFO 有効 (リセット後のデフォルト)
- 1 : 書き込み FIFO 無効

**注 :** FMC\_BCR2.4 レジスタの WFDIS ビットは無視されます。このビットは FMC\_BCR1 レジスタからのみ有効化できます。

#### ビット 20 CCLKEN : 連続クロック有効化

このビットは、外部メモリデバイスに対する FMC\_CLK クロック出力を有効化します。

- 0 : FMC\_CLK は同期メモリアクセス (読み出し/書き込みトランザクション) 時にのみ発生されます。FMC\_CLK クロック比は、FMC\_BCRx レジスタに設定した CLKDIV 値により指定されます (リセット後のデフォルト)。
- 1 : FMC\_CLK は、非同期アクセスと同期アクセス時に連続的に発生されます。CCLKEN がセットされると、FMC\_CLK クロックがアクティブになります。

**注 :** FMC\_BCR2.4 レジスタの CCLKEN ビットは無視されます。このビットは FMC\_BCR1 レジスタからのみ有効化できます。FMC\_CLK 連続クロックが発生するときは、バンク 1 を同期モードに設定する必要があります。

**注 :** CCLKEN ビットがセットされている場合、FMC\_CLK クロック比は FMC\_BTR1 レジスタの CLKDIV 値で指定されます。FMC\_BWTR1 の CLKDIV は、無視されます。

**注 :** 同期モードを使用し、かつ CCLKEN ビットがセットされる場合、バンク 1 以外のバンクに接続される同期メモリはすべて、同じクロックで駆動されます (他のバンクに対する FMC\_BTR2.4 レジスタと FMC\_BWTR2.4 レジスタの CLKDIV 値は無効です)。

#### ビット 19 CBURSTRW : 書き込みバースト有効化

バーストモードでの PSRAM (CRAM) 動作では、このビットは書き込み動作時に同期アクセスを可能にします。同期読み出しアクセスに対するイネーブルビットは、FMC\_BCRx レジスタの BURSTEN ビットです。

- 0 : 書き込み動作は、常に非同期モードで実行されます。
- 1 : 書き込み動作は、同期モードで実行されます。

**ビット 18:16 CPSIZE[2:0] : CRAM ページサイズ**

これらのビットは、ページ間でアドレス境界を跨ぐバーストアクセスができないCellular RAM 1.5 に使用されます。これらのビットを設定すると、FMC コントローラは、メモリページサイズに達した時点で自動的にバーストアクセスを分割します（ページサイズについてはメモリのデータシートを参照）。

000 : ページ境界を跨ぐ場合、バーストは分割されません（リセット後のデフォルト）

001 : 128 バイト

010 : 256 バイト

011 : 512 バイト

100 : 1024 バイト

その他 : 予約済み

**ビット 15 ASYNCWAIT : 非同期転送中のウエイト信号**

このビットは、非同期プロトコル時でも、FMC が NWAIT 信号を有効化/無効化できるようにします。

0 : 非同期プロトコル動作中は、NWAIT 信号は考慮されません（リセット後のデフォルト）。

1 : 非同期プロトコル動作中に NWAIT 信号が考慮されます。

**ビット 14 EXTMOD : 拡張モード有効化**

このビットを使うと、FMC が FMC\_BWTR レジスタで非マルチプレクス非同期アクセスの書き込みタイミングをプログラムできるため、読み出しと書き込みで異なるタイミングが可能になります。

0 : FMC\_BWTR レジスタ内部の値は考慮されません（リセット後のデフォルト）。

1 : FMC\_BWTR レジスタ内部の値が考慮されます。

**注 :** 拡張モードを無効化した場合、FMC は次のようにモード1 またはモード2 で動作することができます。

- モード1は、SRAM/PSRAM メモリタイプを選択した場合 (MTYP = 0x0 または 0x01) のデフォルトモードです。
- モード2は、NOR メモリタイプを選択した場合 (MTYP = 0x10) のデフォルトモードです。

**ビット 13 WAITEN : ウェイトイネーブルビット**

このビットは、同期モードでフラッシュメモリをアクセスするとき NWAIT 信号を使用したウェイトステートの挿入を有効化/無効化します。

0 : NWAIT 信号は無効です（レベルは考慮されず、プログラムされたフラッシュ遅延時間後にウェイトステートは挿入されません）。

1 : NWAIT 信号は有効です（プログラムされた遅延時間後にレベルが考慮され、アサートされた場合に、ウェイトステートが挿入されます）（リセット後のデフォルト）。

**ビット 12 WREN : 書き込みイネーブルビット**

このビットは、FMC によるバンクへの書き込み動作の有効/無効を示します。

0 : FMC によるバンクへの書き込み動作は無効にされています。AHB エラーが報告されます。

1 : FMC によるバンクへの書き込み動作は有効にされています（リセット後のデフォルト）。

**ビット 11 WAITCFG : ウェイトタイミング設定**

NWAIT 信号は、メモリからのデータが有効か、または同期モードでフラッシュメモリをアクセスするときウエイトステートを挿入すべきかを表示します。この設定ビットは、ウェイトステートの1クロック前、またはウエイトステート中に、メモリによって NWAIT がアサートされるかどうかを定義します。

0 : NWAIT 信号は、ウェイトステートの1データサイクル前にアクティブです（リセット後のデフォルト）。

1 : NWAIT 信号は、ウェイトステートの間アクティブです（PSRAM には不使用）。

**ビット 10 予約済みであり、リセット値のままにしておかなければなりません。****ビット 9 WAITPOL : ウェイト信号極性ビット**

同期モードまたは非同期モードで使用されるメモリからのウェイト信号の極性を定義します。

0 : NWAIT はアクティブローです（リセット後のデフォルト）。

1 : NWAIT はアクティブハイです。

**ビット 8 BURSTEN : パーストイネーブルビット**

このビットは、読み出し動作での同期アクセスを有効化/無効化します。これは、パーストモードでの同期メモリ動作に対してのみ有効です。

0 : パーストアクセスモードは無効です (リセット後のデフォルト)。読み出しアクセスは非同期モードで実行されます。

1 : パーストモードを有効化。読み出しアクセスは同期モードで実行されます。

ビット 7 予約済みであり、リセット値のままにしておかなければなりません。

**ビット 6 FACEN : フラッシュアクセス有効化**

NOR 型フラッシュメモリアクセス動作を有効にします。

0 : 対応する NOR 型フラッシュメモリアクセスは無効です。

1 : 対応する NOR フラッシュメモリアクセスは有効です (リセット後のデフォルト)。

**ビット 5:4 MWID[1:0] : メモリデータバス幅。**

外部メモリデバイスの幅を定義します。すべてのタイプのメモリに対して有効です。

00 : 8 ビット

01 : 16 ビット (リセット後のデフォルト)

10 : 32 ビット

11 : 予約済み

**ビット 3:2 MTYP[1:0] : メモリタイプ**

対応するメモリバンクに接続される外部メモリのタイプを定義します。

00 : SRAM (バンク 2..4 のリセット後のデフォルト)

01 : PSRAM (CRAM)

10 : NOR 型フラッシュ /OneNAND フラッシュ (バンク 1 のリセット後のデフォルト)

11 : 予約済み

**ビット 1 MUXEN : アドレス / データマルチプレクシングイネーブルビット**

このビットがセットされているときには、データバス上でアドレスとデータがマルチプレクスされます。NOR および PSRAM メモリでのみ有効です。

0 : アドレス/データはマルチプレクスされません。

1 : アドレス/データはデータバス上でマルチプレクスされます (リセット後のデフォルト)。

**ビット 0 MBKEN : メモリバンクイネーブルビット。**

メモリバンクを有効にします。リセット後は、バンク 1 は有効であり、その他はすべて無効です。無効なバンクにアクセスすると、AHB バス上で ERROR が発生します。

0 : 対応するメモリバンクは無効です

1 : 対応するメモリバンクは有効です。

**SRAM/NOR 型フラッシュチップセレクトタイミングレジスタ 1.4 (FMC\_BTR1..4)**

アドレスオフセット :  $0x04 + 8 * (x - 1)$ ,  $x = 1..4$

リセット値 :  $0x0FFF FFFF$

このレジスタは、SRAM、PSRAM、および NOR 型フラッシュメモリで使用される各メモリバンクの制御情報を含みます。FMC\_BCRx レジスタの EXTMOD ビットがセットされている場合、このレジスタは、書き込みと読み出しアクセスに分割されます。すなわち、読み出しアクセスを設定するためのレジスタ (このレジスタ) と、書き込みアクセスを設定するためのレジスタ (FMC\_BWTRx レジスタ) の 2 つのレジスタが使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ACCMOD		DATLAT				CLKDIV				BUSTURN			
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST								ADDHLD				ADDSET			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値のままにしておかなければなりません。

ビット 29:28 **ACCMOD[1:0]** : アクセスモード

タイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FMC\_BCRx レジスタの EXTMOD ビットが 1 のときのみ考慮されます。

- 00 : アクセスモード A
- 01 : アクセスモード B
- 10 : アクセスモード C
- 11 : アクセスモード D

ビット 27:24 **DATLAT[3:0]** : (下のビット説明参照) 同期メモリのデータ遅延

読み出し/書き込みバーストモードを有効にした同期 アクセスの場合 (BURSTEN / CBURSTRW ビットをセット)、最初のデータの読み書きの前にメモリに発行するメモリクロックサイクル数 (+2) を定義します。

このタイミングパラメータは、HCLK 周期ではなく、FMC\_CLK 周期で表されます。

非同期アクセスでは、この値は無視されます。

- 0000 : 最初のバーストアクセスでは、2 CLK クロックサイクルのデータ遅延
- 1111 : 最初のバーストアクセスでは、17 CLK クロックサイクルのデータ遅延 (リセット後のデフォルト)

ビット 23:20 **CLKDIV[3:0]** : クロック分周比 (FMC\_CLK 信号)

FMC\_CLK クロック出力信号の周期を定義します。HCLK サイクル数で表されます。

- 0000 : 予約済み
- 0001 : FMC\_CLK 周期 = 2 × HCLK 周期
- 0010 : FMC\_CLK 周期 = 3 × HCLK 周期
- 1111 : FMC\_CLK 周期 = 16 × HCLK 周期 (リセット後のデフォルト値)

非同期 NOR 型フラッシュ、SRAM、または PSRAM アクセスでは、この値は無視されます。

**注 :** **FMC\_CLK 分周比の式については、セクション 13.5.5 : 同期トランザクションを参照してください。**

ビット 19:16 **BUSTURN[3:0]** : バスターンアラウンドフェーズ時間

これらのビットは、書き込みから読み出しまで (および読み出しから書き込みまで) のトランザクションの終わりに遅延を追加するためにソフトウェアで書き込まれます。

バスターンアラウンド遅延も、異なるバンク間の 2 つの連続した読み出し/書き込みトランザクションの間に挿入されます。同じ FMC バンクに 2 つの連続した読み出しトランザクションを行う場合は、BUSTURN のタイミングの設定にかかわらず、1 HCLK クロックサイクルだけ挿入されます。また、同じ FMC バンクに 2 つの連続した書き込みトランザクションを行う場合は、バスターンアラウンド遅延は挿入されません。

バスターンアラウンド遅延を使うと、連続トランザクション間の最小時間 (NEx ハイから NEx ローまでの  $t_{EH\ell}$ ) と読み出しアクセス後にメモリがデータバスを開放するために要する最大時間 ( $t_{EHQZ}$ ) を次のように一致させることができます。

- (BUSTURN + 1) HCLK 周期  $\geq t_{EH\ell\min}$  かつ (BUSTURN + 2) HCLK 周期  $\geq t_{EHQZ\max}$  : EXTMOD = 0 の場合
- (BUSTURN + 2) HCLK 周期  $\geq \max(t_{EH\ell\min}, t_{EHQZ\max})$  : EXTMOD = 1 の場合
- 0000 : BUSTURN フェーズ時間 = 0 HCLK クロックサイクル
- ...
- 1111 : BUSTURN フェーズ時間 = 15 × HCLK クロックサイクル (リセット後のデフォルト)

ビット 15:8 **DATAST[7:0]** : データフェーズ時間

これらのビットは、非同期アクセスで使用されるデータフェーズの時間を定義するためにソフトウェアにより書き込まれます (図 34 ~ 図 46 を参照)。

0000 0000 : 予約済み  
 0000 0001 : DATAST フェーズ時間 = 1 \* HCLK クロックサイクル  
 0000 0010 : DATAST フェーズ時間 = 2 \* HCLK クロックサイクル  
 ...

1111 1111 : DATAST フェーズ時間 = 255 \* HCLK クロックサイクル (リセット後のデフォルト値)

各メモリタイプおよびアクセスモードでのデータフェーズ時間については、それぞれの図 (図 34 ~ 図 46) を参照してください。

例 : モード 1、書き込みアクセス、DATAST=1 : データフェーズ時間 = DATAST+1 = 2 HCLK クロックサイクル

**注 :** 同期アクセスでは、この値は無視されます。

ビット 7:4 **ADDHLD[3:0]** : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアから書き込まれ (図 34 ~ 図 46 を参照)、モード D またはマルチプレクスアクセスで使用されます。

0000 : 予約済み  
 0001 : ADDHLD フェーズ時間 = 1 \* HCLK クロックサイクル  
 0010 : ADDHLD フェーズ時間 = 2 \* HCLK クロックサイクル  
 ...

1111 : ADDHLD フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスホールドフェーズ時間については、それぞれの図 (図 34 ~ 図 46) を参照してください。

**注 :** 同期アクセスでは、この値は使用されず、アドレスホールドフェーズは、常に 1 メモリクロック周期の長さです。

ビット 3:0 **ADDSET[3:0]** : アドレスセットアップフェーズ時間

これらのビットは、アドレスセットアップフェーズの時間を定義するために、ソフトウェアによって書き込まれ (図 34 ~ 図 46 を参照)、SRAM、ROM、非同期 NOR フラッシュ、および PSRAM で使用されます。

0000 : ADDSET フェーズ時間 = 0 \* HCLK クロックサイクル  
 ...

1111 : ADDSET フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスセットアップフェーズ時間については、それぞれの図 (図 34 ~ 図 46) を参照してください。

**注 :** 同期アクセスでは、この値は無視されます。

マルチプレクスモードとモード D では、ADDSET の最小値は 1 です。

**注 :** PSRAM (CRAM) の遅延は、内部更新のために可変です。したがって、これらのメモリは、必要に応じて遅延を延長するために、遅延フェーズ全体について NWAIT 信号を発行します。

SPRAM (CRAM) では、フィールド DATLAT を 0 に設定しなければなりません。こうすると、FMC は、すぐに遅延フェーズを終了して、メモリからの NWAIT 信号のサンプリングを開始し、メモリがレディ状態になると、読み出しまたは書き込みを開始します。

この方法は、古い世代のフラッシュメモリと異なり、NWAIT 信号を発行する最新世代の同期フラッシュメモリでも使用できません (使用中の特定のフラッシュメモリのデータシートを参照してください)。

## SRAM/NOR 型フラッシュ書き込みタイミングレジスタ 1.4 (FMC\_BWTR1..4)

アドレスオフセット :  $0x104 + 8 * (x - 1)$ ,  $x = 1...4$

リセット値 : 0x0FFF FFFF

このレジスタには各メモリバンクの制御情報が格納されます。SRAM、PSRAM、NOR フラッシュメモリに使用されます。FMC\_BCRx レジスタの EXTMOD ビットがセットされているときには、このレジスタは書き込みアクセスについてアクティブになります。

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

ビット 31:30 予約済みであり、リセット値のままにしておかなければなりません。

ビット 29:28 **ACCMOD[1:0]** : アクセスモード。

次のタイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FMC\_BCRx レジスタの EXTMOD ビットが 1 にセットされている場合のみ考慮されます。

- 00 : アクセスモード A
- 01 : アクセスモード B
- 10 : アクセスモード C
- 11 : アクセスモード D

ビット 27:20 予約済みであり、リセット値のままにしておかなければなりません。

ビット 19:16 **BUSTURN[3:0]** : バスターンアラウンドフェーズ時間

これらのビットは、書き込みから読み出しまでのトランザクションの終わりに遅延を追加して、次のように連続トランザクション間の最小時間 (ENx ハイから ENx ローまでの  $t_{EHEL}$ ) を一致させるために、ソフトウェアによって書き込まれます。

(BUSTURN + 1) HCLK 周期  $\geq t_{EHELmin}$

0000 : BUSTURN フェーズ時間 = 0 HCLK クロックサイクル

...

1111 : BUSTURN フェーズ時間 = 15 HCLK クロックサイクル追加 (リセット後のデフォルト値)

ビット 15:8 **DATAST[7:0]** : データフェーズ時間

これらのビットは、非同期 SRAM、PSRAM、NOR フラッシュメモリアクセスで使用されるデータフェーズの時間を定義するために、ソフトウェアによって書き込まれます (図 34 ~ 図 46 を参照)。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 \* HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 \* HCLK クロックサイクル

...

1111 1111 : DATAST フェーズ時間 = 255 \* HCLK クロックサイクル (リセット後のデフォルト値)

ビット 7:4 **ADDHLD[3:0]**: アドレスホールドフェーズ時間

これらのビットは、アドレスホールド フェーズの時間を定義するためにソフトウェアによって書き込まれ (図 43 ~ 図 46を参照)、非同期マルチプレクスアクセスで使用されます。

- 0000: 予約済み
- 0001: ADDHLD フェーズ時間 = 1 \* HCLK クロックサイクル
- 0010: ADDHLD フェーズ時間 = 2 \* HCLK クロックサイクル
- ...
- 1111: ADDHLD フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

**注:** 同期 NOR 型フラッシュアクセスでは、この値は使用されず、アドレスホールドフェーズは常に1 フラッシュクロック周期の長さです。

ビット 3:0 **ADDSET[3:0]**: アドレスセットアップフェーズ時間。

これらのビットは、アドレスセットアップ フェーズの時間を HCLK サイクル数で定義するためにソフトウェアによって書き込まれ (図 34 ~ 図 46を参照)、非同期アクセスで使用されます。

- 0000: ADDSET フェーズ時間 = 0 \* HCLK クロックサイクル
- ...
- 1111: ADDSET フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

**注:** 同期アクセスでは、この値は使用されず、アドレスセットアップフェーズは常に1 フラッシュクロック周期の長さです。マルチプレクスモードでは、ADDSET の最小値は1 です。

## 13.6 NAND 型フラッシュコントローラ

FMC は、以下のタイプのデバイスを駆動するのに適した信号タイミングを発生します。

- 8ビットと 16ビットの NAND フラッシュメモリ

NAND バンクは、専用レジスタによって設定されます (セクション 13.6.7を参照)。プログラ可能なメモリパラメータは、アクセスタイミング (表 81を参照) と ECC 構成を含みます。

表 81. プログラ可能なNAND フラッシュのアクセスパラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
メモリセットアップ時間	コマンドアサーション前にアドレスをセットアップするために必要なクロックサイクル (HCLK) 数	読み出し / 書き込み	AHB クロック サイクル (HCLK)	1	255
メモリウェイト	コマンドアサーションの最小時間 (HCLK クロックサイクル数)	読み出し / 書き込み	AHB クロック サイクル (HCLK)	2	255
メモリホールド	コマンドアサーション解除後に、アドレスを (書き込みアクセス時はデータも) 保持する必要があるクロックサイクル数 (HCLK)。	読み出し / 書き込み	AHB クロック サイクル (HCLK)	1	254
メモリデータバスハイインピーダンス	書き込みアクセス開始後にデータバスがハイインピーダンス状態に保たれるクロックサイクル数 (HCLK)	書き込み	AHB クロック サイクル (HCLK)	1	255

## 13.6.1 外部メモリインタフェース信号

次の表に、通常、NAND 型フラッシュメモリとのインタフェースに使用される信号を示します。

注： 接頭辞「N」は、アクティフローの信号を表します。

### 8 ビット NAND フラッシュメモリ

表 82. 8 ビット NAND 型フラッシュ

FMC 信号名	I/O	機能
A[17]	O	NAND 型フラッシュアドレスラッチイネーブル (ALE) 信号
A[16]	O	NAND 型フラッシュコマンドラッチイネーブル (CLE) 信号
D[7:0]	I/O	8 ビットマルチプレクス、双方向アドレス/データバス
NCE	O	チップセレクト
NOE(= NRE)	O	出カイネーブル (メモリ信号名: 読み出しイネーブル、NRE)
NWE	O	書き込みイネーブル
NWAIT/INT	I	FMC への NAND フラッシュレディ/ビジー入力信号

FMC は必要なだけのアドレスサイクルを管理できるので、容量についての理論上の限界はありません。

### 16 ビット NAND フラッシュメモリ

表 83. 16 ビット NAND 型フラッシュ

FMC 信号名	I/O	機能
A[17]	O	NAND 型フラッシュアドレスラッチイネーブル (ALE) 信号
A[16]	O	NAND 型フラッシュコマンドラッチイネーブル (CLE) 信号
D[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス
NCE	O	チップセレクト
NOE(= NRE)	O	出カイネーブル (メモリ信号名: 読み出しイネーブル、NRE)
NWE	O	書き込みイネーブル
NWAIT/INT	I	FMC への NAND フラッシュレディ/ビジー入力信号

FMC は必要なだけのアドレスサイクルを管理できるので、容量についての理論上の限界はありません。

## 13.6.2 NAND 型フラッシュによってサポートされるメモリとトランザクション

表 84 に、サポートされるデバイス、アクセスモード、およびトランザクションを示します。NAND フラッシュコントローラによって許可されない（またはサポートされない）トランザクションは、グレイで示されています。

表 84. サポートされるメモリおよびトランザクション

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	設定許可/ 設定禁止	コメント
NAND 8 ビット	非同期	R	8	8	Y	-
	非同期	W	8	8	Y	-
	非同期	R	16	8	Y	2 つの FMC アクセスに分割
	非同期	W	16	8	Y	2 つの FMC アクセスに分割
	非同期	R	32	8	Y	4 つの FMC アクセスに分割
	非同期	W	32	8	Y	4 つの FMC アクセスに分割
NAND 16 ビット	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割

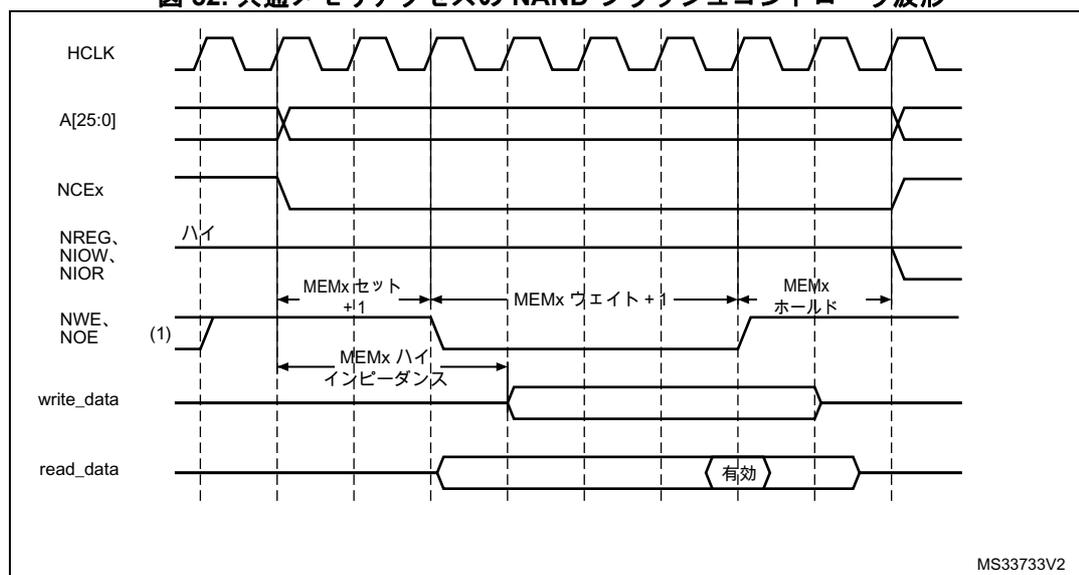
## 13.6.3 NAND フラッシュメモリのタイミング図

NAND 型フラッシュメモリのバンクは、以下のレジスタを通じて管理されます。

- 制御レジスタ : FMC\_PCR
- 割り込みステータスレジスタ : FMC\_SR
- ECC レジスタ : FMC\_ECCR
- 共通メモリ空間用タイミングレジスタ : FMC\_PMEM
- 属性メモリ空間用タイミングレジスタ : FMC\_PATT

各タイミング設定レジスタは、NAND フラッシュアクセスの 3 つのフェーズの HCLK サイクル数を定義するための 3 つのパラメータと、書き込みアクセス時にデータバスの駆動を開始するタイミングを定義する 1 つのパラメータを格納します。図 52 に共通メモリアccessのNANDフラッシュコントローラ波形を示します。属性メモリ空間のアクセスタイミングが同様であることがわかります。

図 52. 共通メモリアクセスの NAND フラッシュコントローラ波形



- 書き込みアクセス時、NOE はハイ (非アクティブ) に保たれます。読み出しアクセス時、NWE はハイ (非アクティブ) に保たれます。

## 13.6.4 NAND フラッシュ動作

NAND フラッシュデバイスのコマンドラッチイネーブル (CLE) およびアドレスラッチイネーブル (ALE) 信号は、FMC コントローラのアドレス信号によって駆動されます。これは、NAND フラッシュメモリにコマンドまたはアドレスを送信するには、CPU は、CPU のメモリ空間内の特定のアドレスに書き込みを行う必要があることを意味します。

NAND フラッシュデバイスからの一般的なページ読み出し動作には、次のステップが必要です。

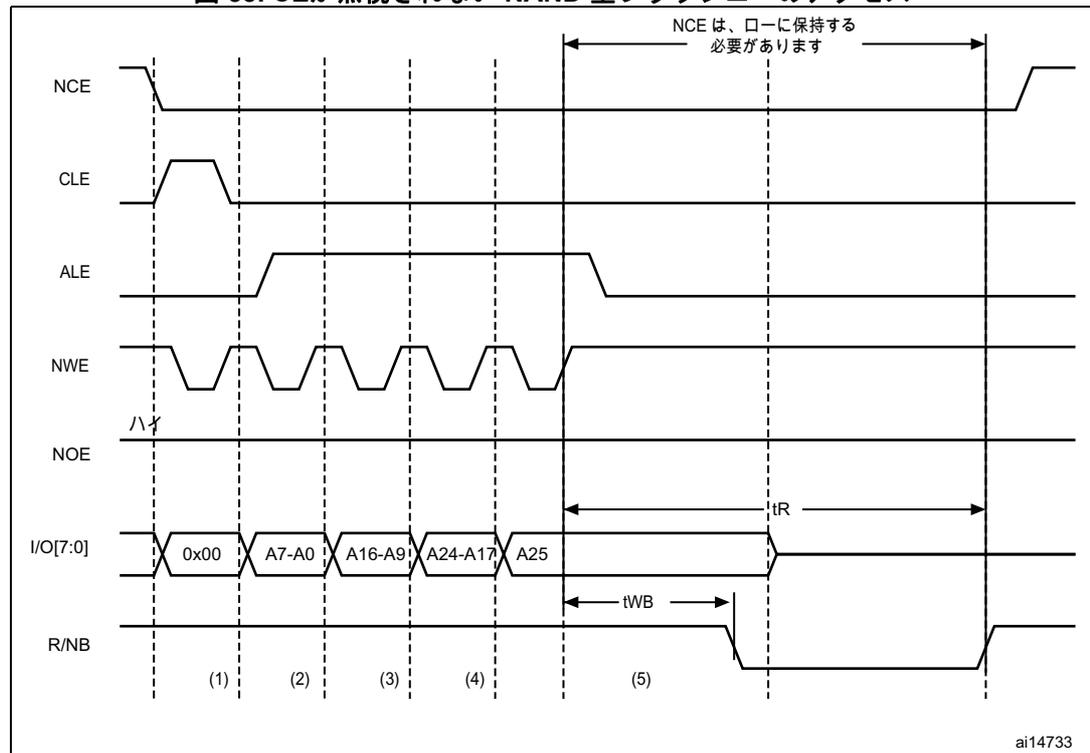
- FMC\_PCR レジスタと FMC\_PMEM レジスタ (デバイスによっては、FMC\_PATT も、[セクション 13.6.5 : NAND フラッシュのプリウエイト機能](#)を参照) を、NAND フラッシュメモリの特徴 (NAND フラッシュのデータバス幅に対する PWID ビット、PTYP = 1、必要に応じて PWAITEN = 0 または 1、タイミング設定については [セクション 13.4.2 : NAND フラッシュメモリアドレスマッピング](#)を参照) に従って設定することで、対応するメモリバンクのプログラムおよび有効化を行います。
- CPU は、1 つのフラッシュコマンドバイトと等しいデータバイト (たとえば、Samsung の NAND フラッシュデバイスの場合は 0x00) で、共通メモリ空間にバイト書き込みを行います。NAND フラッシュの LE 入力は、書き込みストローブ (NWE 上のローパルス) の間はアクティブであり、書き込まれたバイトは、NAND フラッシュによってコマンドとして認識されます。コマンドがメモリデバイスにラッチされたら、次のページ読み出し動作のために書き込む必要はありません。
- CPU は、4 バイト (より小容量のデバイスの場合は 3 バイト) を共通メモリまたは属性空間に書き込むことによって、読み出し操作の開始アドレス (STARTAD) を送信できます (STARTAD[7:0]、STARTAD[16:9]、STARTAD[24:17]、および 64 M ビット x 8 ビット NAND フラッシュの場合は最後に STARTAD[25])。NAND 型フラッシュデバイスの ALE 入力は、書き込みストローブの間アクティブであり (NWE 上のローパルス)、書き込まれたバイトは、読み出し動作の開始アドレスとして解釈されます。属性メモリ空間を使用すると、FMC の異なるタイミング設定を使用でき、一部の NAND フラッシュメモリで必要とされるプリウエイト機能を実装できません (詳細については、[セクション 13.6.5 : NAND フラッシュのプリウエイト機能](#)を参照してください)。

6. コントローラは、NAND フラッシュメモリがレディ (R/NB 信号がハイ) になるのを待ってから、同じメモリバンクまたは別のメモリバンクへの新しいアクセスを開始します。待っている間、コントローラは NCE 信号をアクティブ (ロー) に保ちます。
7. CPU は、NAND フラッシュページ (データフィールド+スペアフィールド) をバイト単位で読み出すために、共通メモリ空間からバイト読み出し動作を行います。
8. 次の NAND フラッシュページは、CPU コマンドやアドレス書き込み動作なしに読み出すことができます。これは次の 3 つの方法で行うことができます。
  - ステップ 5 に記述された動作を実行します。
  - ステップ 3 から動作をリスタートすることによって、新しいランダムアドレスにアクセスできます。
  - ステップ 2 からリスタートすることによって、NAND 型フラッシュデバイスに新しいコマンドを送信できます。

### 13.6.5 NAND フラッシュのプリウエイト機能

一部の NAND フラッシュデバイスでは、アドレスの最後の部分を書き込んだ後、コントローラは R/NB 信号を待つ必要があります。(図 53を参照。)

図 53. CEが無視されない NAND 型フラッシュへのアクセス



1. CPU はアドレス 0x7001 0000 にバイト 0x00 を書き込みました。
2. CPU はアドレス 0x7002 0000 にバイト A7~A0 を書き込みました。
3. CPU はアドレス 0x7002 0000 にバイト A16~A9 を書き込みました。
4. CPU はアドレス 0x7002 0000 にバイト A24~A17 を書き込みました。
5. CPU はアドレス 0x7002 0000 にバイト A25 を書き込みました : FMC は FMC\_PATT2 のタイミング定義を使用して書き込みアクセスを実行します。ここで  $ATTHOLD \geq 7$  (ただし、 $(7+1) * HCLK = 112 \text{ ns} > t_{WB \text{ max}}$ )。これにより、NCE は R/NB がローから再びハイになるまでローを維持します (NCE が無視されない NAND 型フラッシュメモリの場合のみ必要)。

この機能が必要な場合、MEMHOLD 値をプログラムすることにより  $t_{WB}$  タイミングを確実に満たすことができます。ただし、NAND フラッシュメモリに対するすべての CPU 読み出しまたは書き込みアクセスでは、(MEMHOLD+ 1) HCLK サイクルのホールド遅延が NWE 信号の立ち上がりエッジと次のアクセスとの間に挿入されます。

このタイミング制約に対処するためには、 $t_{WB}$  タイミングに応じた ATTHOLD 値でタイミングレジスタをプログラムし、MEMHOLD 値を最小値のままにしておくことで、属性メモリ空間を使用できます。この場合、CPU は NAND フラッシュのすべての読み出しおよび書き込みアクセスには共通メモリ空間を使用する必要があります。ただし、NAND フラッシュデバイスに最後のアドレスバイトを書き込むときには、属性メモリ空間に書き込む必要があります。

## 13.6.6 エラー訂正コード (ECC) の計算 NAND 型フラッシュメモリ

FMC NAND カードコントローラには、メモリバンクごとに 1 つずつ、2 つのエラー訂正コード計算ハードウェアブロックがあります。これらのハードウェアブロックは、ソフトウェアで ECC を処理する場合、ホスト CPU の負荷を削減します。

この 2 つの ECC ブロックは同一であり、それぞれバンク 2 とバンク 3 に接続されます。結果として、バンク 4 に接続されたメモリでは、ハードウェア ECC 計算を使用できません。

FMC に実装されたエラー訂正コード (ECC) のアルゴリズムでは、NAND フラッシュメモリに対する 256、512、1024、2048、4096、または 8192 バイトの読み出しまたは書き込みごとに、1 ビットのエラー訂正と 2 ビットのエラー検出を実行できます。ハミングコーディングアルゴリズムを採用し、行と列のパリティ計算を行います。

ECC モジュールは、NAND フラッシュメモリバンクがアクティブになるたびに、NAND フラッシュデータバスおよび読み出し/書き込み信号 (NCE と NWE) を監視します。

ECC は次のように動作します。

- NAND フラッシュメモリのバンク 2 またはバンク 3 へアクセスするときは、D[15:0] バス上のデータがラッチされて、ECC 計算に使用されます。
- NAND フラッシュメモリの他のアドレスをアクセスするときは、ECC ロジックはアイドル状態となり、動作しません。このため、NAND フラッシュメモリに対するコマンドまたはアドレスを定義する書き込み動作は、ECC 計算で考慮されません。

目的のバイト数がホスト CPU によって NAND フラッシュメモリから読み出されるか、NAND フラッシュメモリに書き込まれたら、FMC\_ECCR レジスタを読み出して、計算値を取得する必要があります。読み出し後は、ECCEN ビットを "0" にリセットすることによってクリアされます。新しいデータブロックを計算するには、FMC\_PCR レジスタの ECCEN ビットを 1 にセットする必要があります。

ECC 計算を行うときは：

1. FMC\_PCR レジスタで ECCEN ビットを有効にします。
2. NAND 型フラッシュメモリページヘータを書き込みます。NAND ページへ書き込む際、ECC ブロックは ECC 値を計算します。
3. FMC\_ECCR レジスタにある ECC 値を読み出して、変数内に格納します。
4. ECCEN ビットをクリアし、FMC\_PCR レジスタで有効にした後、書き込んだデータを NAND ページから読み出します。NAND ページを読み出す際、ECC ブロックは ECC 値を計算します。
5. FMC\_ECCR レジスタにある新しい ECC 値を読み出します。
6. 2 つの ECC 値が同じ場合、訂正は不要です。その他の場合、ECC エラーが存在するためソフトウェア訂正ルーチンがエラーが訂正できるか否かの情報を返します。

## 13.6.7 NAND フラッシュコントローラレジスタ

### NAND フラッシュ制御レジスタ (FMC\_PCR)

アドレスオフセット : 0x80

リセット値 : 0x0000 0018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	ECCPS			TAR									
												r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR			TCLR				Res.	Res.	ECCEN	PWID		PTYP	PBKEN	PWAITEN	Res.
r/w			r/w	r/w	r/w	r/w	r/w	r/w							

ビット 31:20 予約済みであり、リセット値のままにしておかなければなりません。

ビット 19:17 **ECCPS[2:0]** : ECC ページサイズ

外部 ECC のページサイズを定義します :

- 000 : 256 バイト
- 001 : 512 バイト
- 010 : 1024 バイト
- 011 : 2048 バイト
- 100 : 4096 バイト
- 101 : 8192 バイト

ビット 16:13 **TAR[3:0]** : ALE から RE 間の遅延

ALE ローから RE ローまでの時間を AHB クロックサイクル (HCLK) 数で設定します。時間は、 $t_{ar} = (TAR + SET + 2) \times THCLK$  です。THCLK は HCLK クロック周期です。

- 0000 : 1 HCLK サイクル (デフォルト)
- 1111 : 16 HCLK サイクル

**注 :** SET は、アドレス指定される空間に応じて、MEMSET または ATTSET です。

ビット 12:9 **TCLR[3:0]** : CLE から RE 間の遅延

CLE ローから RE ローまでの時間を AHB クロックサイクル (HCLK) 数で設定します。時間は、 $t_{clr} = (TCLR + SET + 2) \times THCLK$  です。THCLK は HCLK クロック周期です。

- 0000 : 1 HCLK サイクル (デフォルト)
- 1111 : 16 HCLK サイクル

**注 :** SET は、アドレス指定される空間に応じて、MEMSET または ATTSET です。

ビット 8:7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6 **ECCEN** : ECC 計算ロジックイネーブルビット

- 0 : ECC ロジックは無効であり、リセットされています (リセット後のデフォルト)。
- 1 : ECC ロジックは有効です。

ビット 5:4 **PWID[1:0]** : データバス幅。

外部メモリデバイスの幅を定義します。

- 00 : 8 ビット
- 01 : 16 ビット (リセット後のデフォルト)
- 10 : 予約済み
- 11 : 予約済み

ビット 3 **PTYP** : メモリタイプ

対応するメモリバンクに接続されたデバイスのタイプを定義します。

- 0 : 予約済みであり、リセット値のままにしておかなければなりません。
- 1 : NAND 型フラッシュ (リセット後のデフォルト)



ビット 2 **PBKEN** : NAND フラッシュメモリバンクイネーブルビット

メモリバンクを有効にします。無効なメモリバンクにアクセスすると、AHB バスの ERROR になります。

0 : 対応するメモリバンクは無効です (リセット後のデフォルト)。

1 : 対応するメモリバンクは有効です。

ビット 1 **PWAITEN** : ウェイト機能イネーブルビット

NAND 型フラッシュメモリバンクのウェイト機能を有効にします。

0 : 無効

1 : 有効

ビット 0 予約済みであり、リセット値のままにしておかなければなりません。

### FIFO ステータスおよび割り込みレジスタ (FMC\_SR)

アドレスオフセット : 0x84

リセット値 : 0x0000 0040

このレジスタは、FIFO ステータスと割り込みに関する情報を格納します。FMC には、AHB の 16 ワードまでのデータをメモリに書き込んで格納するときを使用される FIFO があります。

これは、FMC が FIFO をメモリへ転送すると同時に、FIFO に迅速に書き込んで、FMC 以外のペリフェラルへのトランザクションに AHB を解放するために使用されます。これらのレジスタビットの 1 つは、ECC のために、FIFO のステータスを示します。

ECC は、データのメモリへの書き込み中に計算されます。正しい ECC を読み出すため、ソフトウェアは FIFO がエンプティになるまで待つ必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS								
									r	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6 **FEMPT** : FIFO エンプティ

FIFO のステータスを表示する読み取り専用ビット

0 : FIFO 非エンプティ

1 : FIFO エンプティ

ビット 5 **IFEN** : 割り込み立ち下がりがエッジ検出イネーブルビット

0 : 割り込み立ち下がりがエッジ検出リクエストは無効

1 : 割り込み立ち下がりがエッジ検出リクエストは有効

ビット 4 **ILEN** : 割り込みハイレベル検出イネーブルビット

0 : 割り込みハイレベル検出リクエストは無効

1 : 割り込みハイレベル検出リクエストは有効

ビット 3 **IREN** : 割り込み立ち上がりエッジ検出イネーブルビット

0 : 割り込み立ち上がりエッジ検出リクエストは無効

1 : 割り込み立ち上がりエッジ検出リクエストは有効

ビット 2 **IFS** : 割り込み立ち下がりがエッジステータス

このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。

0 : 割り込み立ち下がりがエッジは発生しませんでした。

1 : 割り込み立ち下がりがエッジが発生しました。

ビット 1 **ILS** : 割り込みハイレベルステータス

このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。

- 0 : 割り込みハイレベルは発生しませんでした。
- 1 : 割り込みハイレベルが発生しました。

ビット 0 **IRS** : 割り込み立ち上がりエッジステータス

このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。

- 0 : 割り込み立ち上がりエッジは発生しませんでした。
- 1 : 割り込み立ち上がりエッジが発生しました。

## 共通メモリ空間タイミングレジスタ 2.4 (FMC\_PMEM)

アドレスオフセット : アドレス : 0x88

リセット値 : 0xFCFC FCFC

FMC\_PMEMx 読み出し/書き込みレジスタは、NAND フラッシュメモリバンクのタイミング情報を格納します。この情報は、NAND フラッシュの共通メモリ空間に、コマンド、アドレスの書き込みアクセス、およびデータ読み出し/書き込みアクセスするために使われます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEMHIZx								MEMHOLDx							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMWAITx								MEMSETx							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **MEMHIZ[7:0]** : 共通メモリ x データバスハイインピーダンス時間

ソケット上の共通メモリ空間への NAND フラッシュによる書き込みアクセスの開始後に、データバスをハイインピーダンスに保つ HCLK クロックサイクル数を定義します。これは、書き込み トランザクションに対してのみ有効です。

- 0000 0000 : 1 HCLK サイクル
- 1111 1110 : 255 HCLK サイクル
- 1111 1111 : 予約済み

ビット 23:16 **MEMHOLD[7:0]** : 共通メモリホールド時間

ソケット x 上の共通メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドのデアサーション (NWE、NOE) 後にアドレスを (書き込みアクセス時はデータも) 保持する HCLK クロックサイクル数を定義します。

- 0000 0000 : 予約済み
- 0000 0001 : 1 HCLK サイクル
- 1111 1110 : 254 HCLK サイクル
- 1111 1111 : 予約済み

ビット 15:8 **MEMWAIT[7:0]** : 共通メモリウェイト時間

ソケット上の共通メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンド (NEW、NOE) をアサートするための HCLK (+1) クロックサイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサクションの時間は延長されません。

- 0000 0000 : 予約済み
- 0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)
- 1111 1110 : 255 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)
- 1111 1111 : 予約済み



ビット 7:0 **MEMSET[7:0]** : 共通メモリ x セットアップ時間

ソケット x 上の共通メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドアサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロックサイクル数を定義します。

0000 0000 : 1 HCLK サイクル  
 1111 1110 : 255 HCLK サイクル  
 1111 1111 : 予約済み

## 属性メモリ空間タイミングレジスタ 2.4 (FMC\_PATT)

アドレスオフセット : 0x8C

リセット値 : 0xFCFC FCFC

FMC\_PATT 読み出し/書き込みレジスタは、NAND フラッシュメモリバンクのタイミング情報を格納します。この情報は、最後のアドレス書き込みアクセスの際に、タイミングが以前のアクセスのタイミングと異なる必要がある場合に、NAND フラッシュの属性メモリ空間へ 8 ビットアクセスするために使用されます (レディ/ビジー管理については、[セクション 13.6.5 : NAND フラッシュのプリウェイト機能](#)を参照)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTHIZx								ATTHOLDx							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATTWAITx								ATTSETx							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **ATTHIZ[7:0]** : 属性メモリデータバスハイインピーダンス時間

ソケット上の属性メモリ空間への NAND フラッシュによる書き込みアクセスの開始後に、データバスをハイインピーダンスに保つ HCLK クロックサイクル数を定義します。これは、書き込みトランザクションに限り有効です。

0000 0000 : 0 HCLK サイクル  
 1111 1110 : 255 HCLK サイクル (リセット後のデフォルト値)  
 1111 1111 : 予約済み

ビット 23:16 **ATTHOLD[7:0]** : 属性メモリホールド時間

ソケット上の属性メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドのデアサーション (NWE、NOE) 後にアドレスを (書き込みアクセス時はデータも) 保持する HCLK クロックサイクル数を定義します。

0000 0000 : 予約済み  
 0000 0001 : 1 HCLK サイクル  
 1111 1110 : 254 HCLK サイクル  
 1111 1111 : 予約済み

ビット 15:8 **ATTWAIT[7:0]** : 属性メモリウェイト時間

ソケット x 上の属性メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンド (NEW、NOE) をアサートするための HCLK (+1) クロックサイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサーションの時間は延長されません。

0000 0000 : 予約済み  
 0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)  
 1111 1110 : 255 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)  
 1111 1111 : 予約済み

ビット 7:0 **ATTSET[7:0]**: 属性メモリセットアップ時間

ソケット上の属性メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドアサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロックサイクル数を定義します。

- 0000 0000 : 1 HCLK サイクル
- 1111 1110 : 255 HCLK サイクル
- 1111 1111 : 予約済み

## ECC 結果レジスタ (FMC\_ECCR)

アドレスオフセット : 0x94

リセット値 : 0x0000 0000

このレジスタは、FMC NAND コントローラの ECC 計算モジュールによって計算された現在のエラー訂正コード値を格納します。CPU が、NAND フラッシュメモリページ上の正しいアドレスからデータを読み出すときには (セクション 13.6.6 : エラー訂正コード (ECC) の計算 NAND 型フラッシュメモリを参照)、NAND フラッシュメモリに対して読み書きされたデータは、ECC 計算モジュールによって自動的に処理されます。X バイトを読み出した後 (FMC\_PCR レジスタの ECCPS フィールドに従って)、CPU は FMC\_ECC レジスタから計算された ECC 値を読み出す必要があります。次に、これらの計算されたパリティデータがスピア領域に記録されているパリティ値と一致するか検証して、ページが有効か否か、訂正が必要か否かを確認します。FMC\_ECCR レジスタは、読み出し後、ECCEN ビットを "0" に設定することによってクリアする必要があります。新しいデータブロックを計算するときは、ECCEN ビットを "1" にセットする必要があります。

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

ビット 31:0 **ECC** : ECC 結果

このフィールドは、ECC 計算ロジックによって計算された値を格納します。表 85 に、これらのビットフィールドの値を示します。

表 85. ECC 結果関連ビット

ECCPS[2:0]	ページサイズ (バイト数)	ECC ビット
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

## 13.7 SDRAM コントローラ

### 13.7.1 SDRAM コントローラの主な機能

SDRAM コントローラの主な機能は次の通りです。

- 独立した設定を持つ 2 つの SDRAM バンク
- 8 ビット、16 ビット、32 ビットのデータバス幅
- 13 ビットのアドレス行、11 ビットのアドレス列、4 つの内部バンク : 4x16Mx32 ビット (256 MB)、4x16Mx16 ビット (128 MB)、4x16Mx8 ビット (64 MB)
- ワード、ハーフワード、バイトアクセス
- HCLK/2 または HCLK/3 の SDRAM クロックが可能
- 行およびバンクの自動境界管理
- マルチバンクピンポンアクセス
- プログラム可能なタイミングパラメータ
- プログラム可能なリフレッシュレートでの自動リフレッシュ動作
- セルフリフレッシュモード
- パワーダウンモード
- ソフトウェアによる SDRAM パワーアップ初期化
- CAS 遅延 = 1、2、3
- 深さ 6 ライン \* 32 ビット (6x14 ビットアドレスタグ) のキャッシュ可能読み出し FIFO

### 13.7.2 SDRAM 外部メモリインタフェース信号

FMC SDRAM コントローラと外部 SDRAM デバイスとのインタフェースに使用される SDRAM I/O ピンは、起動時にユーザーアプリケーションから設定する必要があります。アプリケーションで使用しない SDRAM コントローラ I/O ピンは、他の用途に使用することができます。

表 86. SDRAM 信号

SDRAM 信号	I/O タイプ	説明	オルタネート機能
SDCLK	O	SDRAM クロック	-
SDCKE[1:0]	O	SDCKE0 : SDRAM バンク 1 クロック有効化 SDCKE1 : SDRAM バンク 2 クロック有効化	-
SDNE[1:0]	O	SDNE0 : SDRAM バンク 1 チップ有効化 SDNE1 : SDRAM バンク 2 チップ有効化	-
A[12:0]	O	アドレス	FMC_A[12:0]
D[31:0]	I/O	双方向データバス	FMC_D[31:0]
BA[1:0]	O	バンクアドレス	FMC_A[15:14]
NRAS	O	行アドレスストロープ	-
NCAS	O	列アドレスストロープ	-
SDNWE	O	書き込みイネーブル	-
NBL[3:0]	O	書き込みアクセス用の出力バイトマスク (メモリ信号名 : DQM[3:0])	FMC_NBL[3:0]

## 13.7.3 SDRAM コントローラの機能説明

すべての SDRAM コントローラ出力 (信号、アドレス、データ) は、メモリクロック (FMC\_SDCLK) の立ち下がリエッジで変化します。

### SDRAM 初期化

初期化シーケンスはソフトウェアから管理されます。2 つのバンクを使用する場合、FMC\_SDCMR レジスタの対象バンクビット CTB1 と CTB2 をセットして、初期化シーケンスをバンク 1 とバンク 2 に対して同時に発生させる必要があります。

1. FMC\_SDCRx レジスタにメモリデバイス機能をプログラムします。SDRAM クロック、RBURST、RPIPE は FMC\_SDCR1 レジスタに設定する必要があります。
2. メモリデバイスタイミングは FMC\_SDTRx レジスタに設定します。TRP タイミングと TRC タイミングは、FMC\_SDTR1 レジスタに設定します。
3. MODE ビットに"001"を書き込み、FMC\_SDCMR レジスタの対象バンクビット (CTB1 および /または CTB2) を設定して、メモリに対するクロックの供給を開始させます (SDCKE はハイに駆動されます)。
4. 所定の遅延時間を待ちます。代表的遅延時間は約 100 ms です (パワーアップ後に必要とされる遅延については、SDRAM データシート参照)。
5. MODE ビットに"010"を書き込み、FMC\_SDCMR レジスタの対象バンクビット (CTB1 および /または CTB2) を設定して、「Precharge All」コマンドを発行させます。
6. MODE ビットに"011"を書き込み、FMC\_SDCMR レジスタの対象バンクビット (CTB1 および /または CTB2) 、さらに連続自動リフレッシュコマンド数 (NRFS) を設定します。発行が必要な自動リフレッシュコマンド数については SDRAM データシートを参照してください。代表値は 8 です。
7. SDRAM デバイスに応じて MRD フィールドを設定し、MODE ビットに"100"を書き込み、FMC\_SDCMR レジスタの対象バンクビット (CTB1 および /または CTB2) を設定して、「Load Mode Register」コマンドを発行させ、SDRAM デバイスに書き込みを行います。

特に、

- a) CAS 遅延は、FMC\_SDCR1/2 レジスタの設定値に従って選択する必要があります。
- b) バースト長 (BL) の 1 は、モードレジスタの M[2:0] ビットに"000"を設定することによって選択する必要があります。SDRAM デバイスのデータシートを参照してください。

モードレジスタが両 SDRAM バンクに対して一致しない場合、このステップを各バンクに 1 回ずつ 2 回繰り返し、目標バンクビットをセットする必要があります。

8. リフレッシュレートを FMC\_SDRTR レジスタに設定します。  
リフレッシュレートは、リフレッシュサイクル間の遅延に対応します。この値を SDRAM デバイスに対して使用する必要があります。
9. モバイル SDRAM デバイスの場合、拡張モードレジスタのプログラムは SDRAM デバイスが初期化されてから行ってください。まず、BA1 = 1 かつ BA = 0 (BA[1:0] アドレスマッピングについては、SDRAM アドレスマッピングのセクションを参照) であるときに、ダミーの読み出しアクセスを 1 回行う必要があります。これは、ロードモードレジスタではなく拡張モードレジスタを選択して、必要な値を書き込むためです。

この段階で、SDRAM デバイスはコマンドを受け付ける準備が整います。SDRAM アクセスの実行中にシステムリセットが発生した場合、データバスはそのまま SDRAM デバイスによって駆動される可能性があります。従って、SDRAM デバイスはリセット後、NOR フラッシュ、PSRAM、SRAM、NAND フラッシュコントローラのいずれかで新しいアクセスを発行する前に、再初期化をする必要があります。

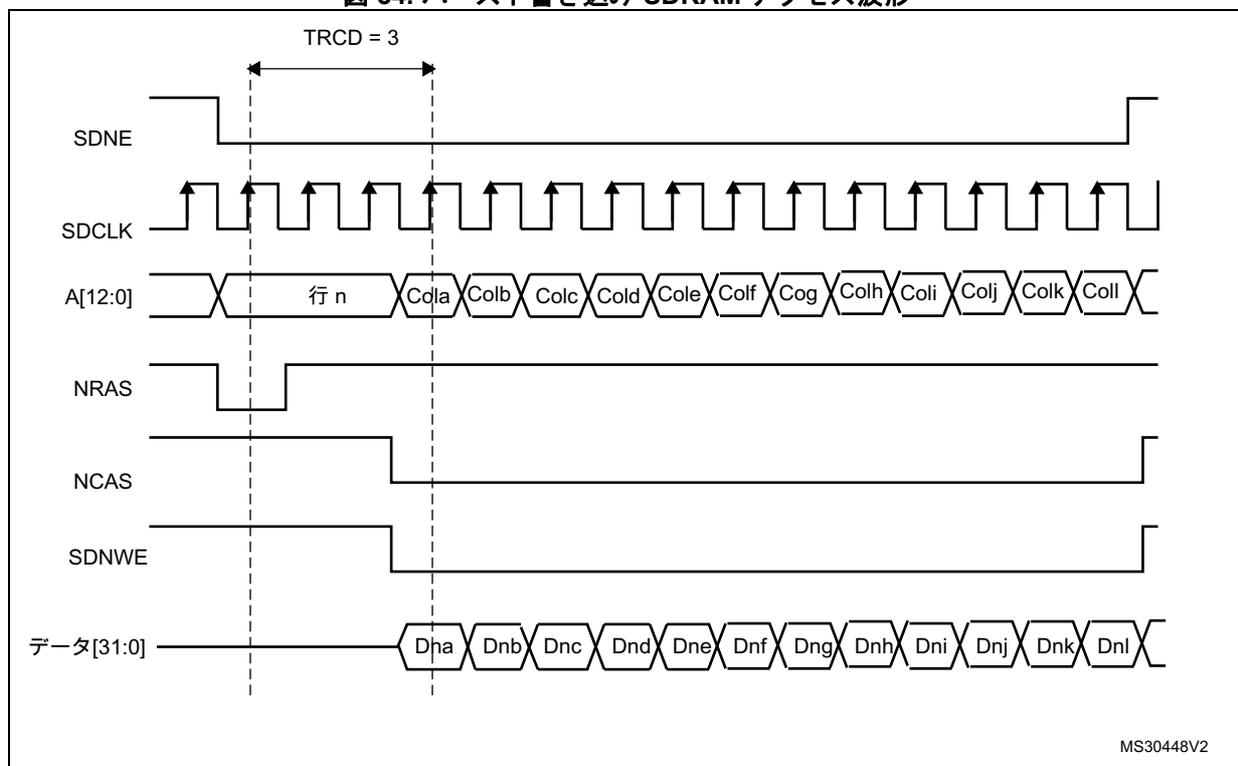
注： 2 つの SDRAM デバイスを FMC に接続する場合は、コマンドモードレジスタにより両デバイスに同時に実行されるすべてのアクセス（ロードモードレジスタコマンド）は、FMC\_SDTR1 レジスタで SDRAM バンク 1 に対して設定されたタイミングパラメータ（TMRD タイミング、TRAS タイミング）を使って発行されます。

## SDRAM コントローラ書き込みサイクル

SDRAM コントローラはシングルおよびバースト書き込みリクエストを受け付けて、これらをシングルメモリアクセスに変換します。両ケースとも、SDRAM コントローラはアクティブ行を記録して、各バンクが異なるバンクへ連続書き込みアクセスを行えるようにします（マルチバンクピンポンアクセス）。

書き込みアクセスを行う前に、FMC\_SDCRx レジスタの WP ビットをクリアして、SDRAM バンク書き込み保護機能を無効化しておく必要があります。

図 54. バースト書き込み SDRAM アクセス波形



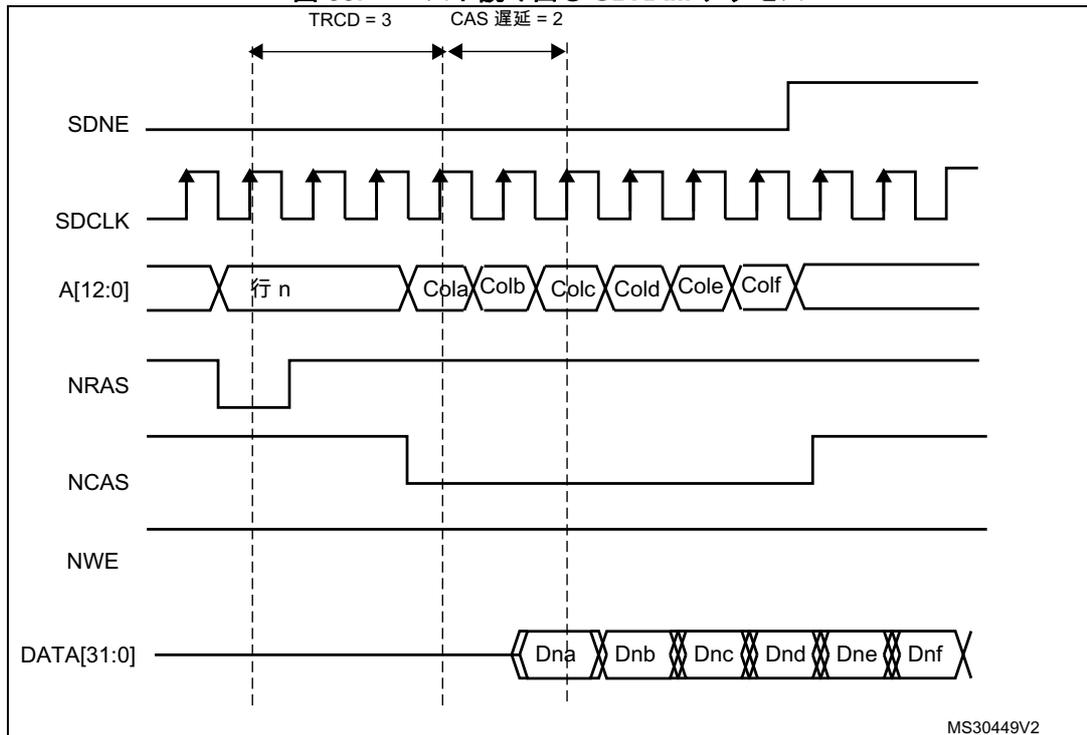
SDRAM コントローラは常に次のアクセスをチェックしています。

- 次のアクセスが同じ行または別のアクティブな行である場合、書き込み動作が実行されます。
- 次のアクセスが別の行（非アクティブ）を対象とする場合、SDRAM コントローラはプリチャージコマンドを発生し、新しい行をアクティブにして、書き込みコマンドを開始します。

## SDRAM コントローラ読み出しサイクル

SDRAM コントローラはシングルおよびバースト読み出しリクエストを受け付けて、これらをシングルメモリアクセスに変換します。両ケースとも、SDRAM コントローラはアクティブ行を記録して、各バンクが異なるバンクへ連続読み出しアクセスを行えるようにします（マルチバンクピンポンアクセス）。

図 55. バースト読み出し SDRAM アクセス



FMC SDRAM コントローラは、キャッシュ可能読み出し FIFO (6 ライン x 32 ビット) を内蔵しています。このFIFOは、CAS 遅延中および RPIPE 遅延中に次式により予め読み出しておいたデータを保存するために使われます。次の読み出しアクセスを予測するときは、FMC\_SDQR1 レジスタのRBURST ビットをセットする必要があります。

予測データの番号 = CAS 遅延 + 1 + (RPIPE 遅延)/2

例 :

- CAS 遅延 = 3、RPIPE 遅延 = 0 の場合 : 4 つのデータ (使途未定) が FIFO に保存されます。
- CAS 遅延 = 3、RPIPE 遅延 = 0 の場合 : 5 つのデータ (使途未定) が FIFO に保存されます。

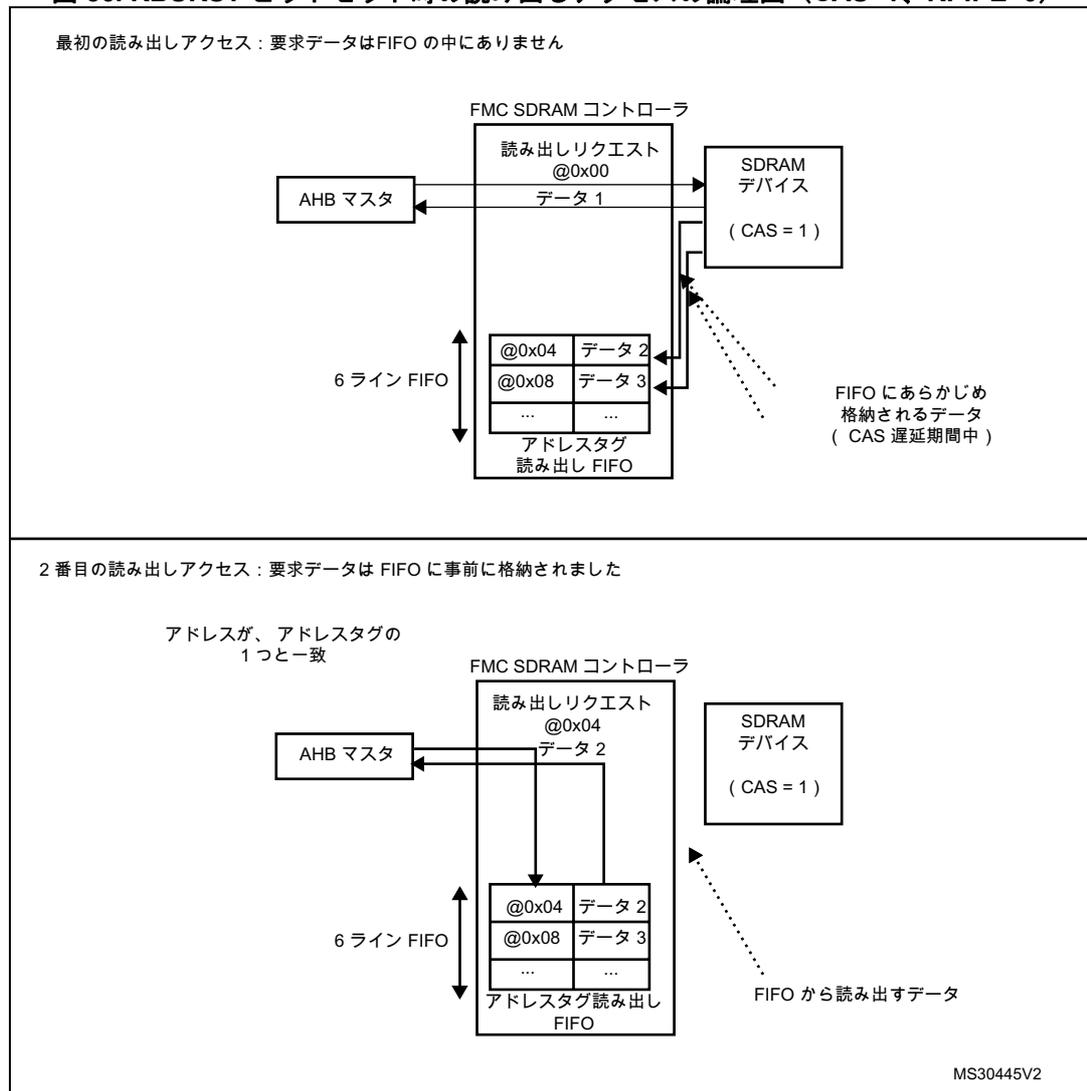
読み出し FIFO は、各ラインに対して値を識別する 14 ビットのアドレスタグを持っています。11ビットは列アドレスに、2 ビットは内部バンクとアクティブ行の選択に、1 ビットは SDRAM デバイスの選択に、それぞれ使用します。

AHB バースト読み出し時に予め行の終わりに到達している場合、予め読み出されたデータ (使途未定) は読み出し FIFO に保存されません。シングル読み出しアクセスの場合、データは正常に FIFO に保存されます。

読み出しリクエストが発生するたびに、SDRAM コントローラは次をチェックします。

- アドレスがアドレスタグの 1 つに一致すると、データが直接 FIFO から読み出され、対応するアドレスタグ/ライン値がクリアされ、FIFO 内の残りのデータはエンブティラインを回避するためコンパクト化されます。
- その他の場合、新しい読み出しコマンドがメモリに対して発行され、FIFO は新しいデータで更新されます。FIFO がフルの場合、古いデータは失われます。

図 56. RBURST ビットセット時の読み出しアクセスの論理図 (CAS=1、RPIPE=0)



書き込みアクセス時またはプリチャージコマンド時、読み出し FIFO が一掃されて、新しいデータ書き込みの準備が整います。

最初の読み出しリクエストの後、現在のアクセスが行境界に対して実行されなかった場合、SDRAM コントローラは、CAS遅延中および RPIPE遅延中（設定した場合）に次の読み出しアクセスがあるものと見なします。これは、メモリアドレスをインクリメントして行います。次の条件を満たす必要があります。

- FMC\_SDCR1 レジスタの RBURST 制御ビットを“1”にセットする必要があります。

アドレスは次の AHB リクエストに応じて管理されます。

- 次の AHB リクエストはシーケンシャル (AHB バースト) である。  
この場合、SDRAM コントローラはアドレスをインクリメントさせます。
- 次の AHB リクエストはシーケンシャルでない。
  - 新しい読み出しリクエストが同じ行または別のアクティブな行を対象とする場合、新しいアドレスがメモリへ渡され、マスタは CAS 遅延の間停止され、メモリからの新しいデータを待ちます。
  - 新しい読み出しリクエストがアクティブな行を対象としない場合、SDRAM コントローラはプリチャージコマンドを発生し、新しい行をアクティブにして、読み出しコマンドを開始します。

RBURST がリセットされると、読み出し FIFO は使用されません。

## 行およびバンク境界管理

読み出しアクセスまたは書き込みアクセスが行境界を跨ぐとき、次の読み出しアクセスまたは書き込みアクセスがシーケンシャルで、かつ現在のアクセスが行境界に対して実行された場合、SDRAM コントローラは次の動作を行います。

1. アクティブ行のプリチャージ
2. 新しい行のアクティブ化
3. 読み出し／書き込みコマンドの開始

行境界で、すべての列とデータバス幅設定に対して次の行の自動アクティブ化をサポート

必要に応じて、SDRAM コントローラは次のようにコマンドの間に追加クロックサイクルを挿入します。

- プリチャージコマンドとアクティブコマンドの間に挿入して TRP パラメータを一致させます (次のアクセスが同じバンクの異なる行の場合のみ)
- アクティブコマンドと読み出しコマンドの間に挿入して TRCD パラメータを一致させます。

これらのパラメータは FMC\_SDTRx レジスタで定義されます。

行境界を跨ぐ読み出しアクセスとバースト書き込みアクセスについては、[図 57](#) と [図 58](#) を参照してください。

図 57. 行境界を跨ぐ読み出しアクセス

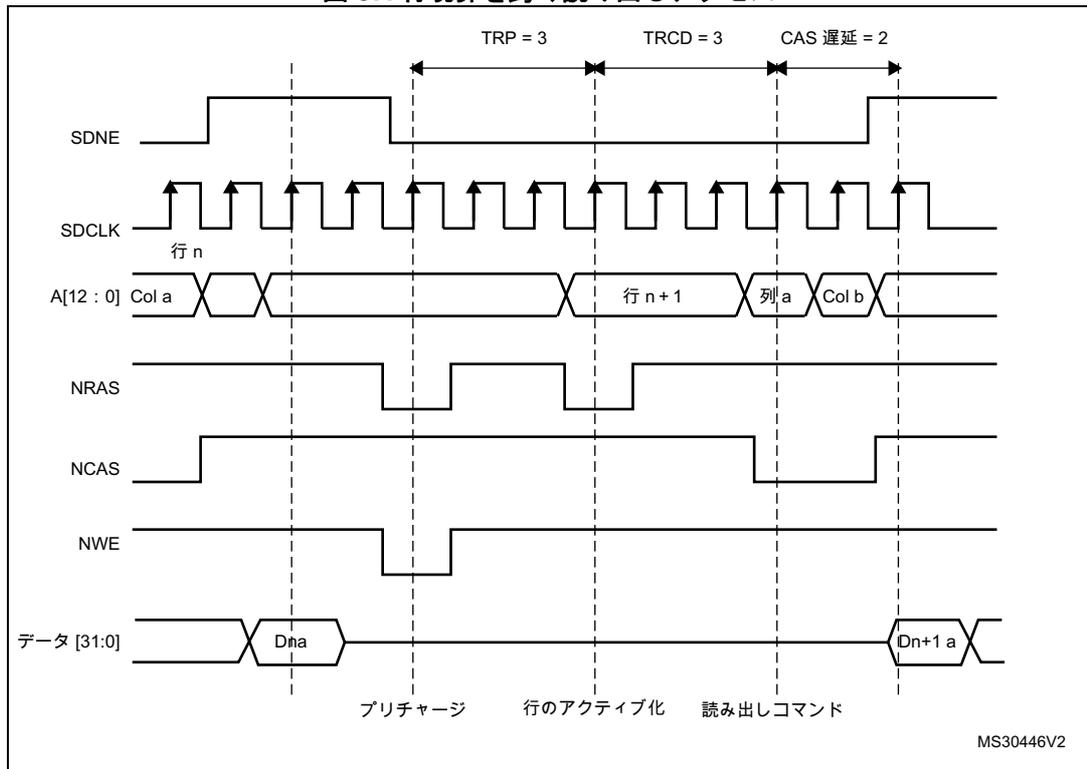
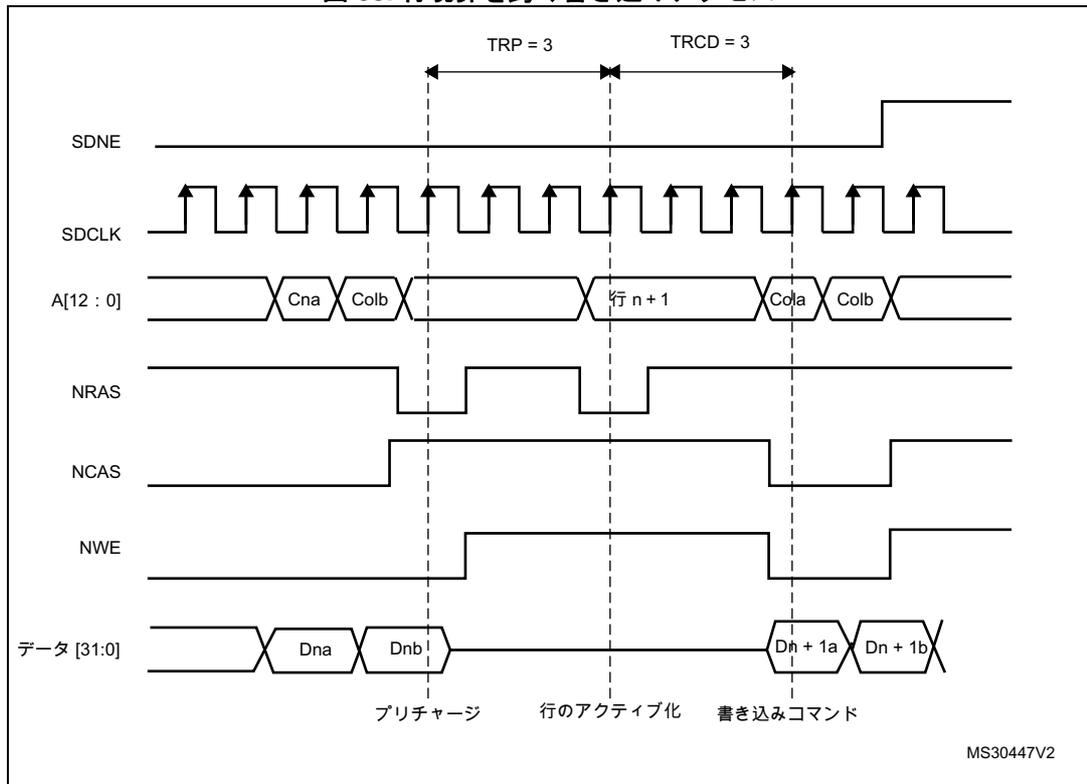


図 58. 行境界を跨ぐ書き込みアクセス



次のアクセスがシーケンシャルで、かつ現在のアクセスがバンク境界を跨ぐ場合、SDRAM コントローラは次のバンクの最初の行をアクティブ化して、新しい読み出し／書き込みコマンドを開始します。次の 2 つのケースが可能です。

- 現在のバンクが直前のバンクでない場合、新しいバンク内のアクティブな行をプリチャージする必要があります。バンク境界で、すべての行／列とデータバス幅設定に対して次の行の自動アクティブ化をサポートしています。
- 現在のバンクが直前のバンクである場合、次の行の自動アクティブ化は、13 ビット行、11 ビット列、4 つの内部バンク、32 ビットデータバスの SDRAM デバイスをアドレス指定する場合にのみサポートしています。その他の場合には、SDRAM アドレス範囲に違反するため、AHB エラーが発生します。
- 13 ビット行アドレス、11 ビット列アドレス、4 つの内部バンク、バス幅 32 ビットの SDRAM メモリでは、バンク境界で、SDRAM コントローラは 2 番目の SDRAM デバイスに対する読み出し／書き込みを続けます（ただし初期化済みとします）。
  - a) 最初の内部バンク内にアクティブな行が既に存在している場合、SDRAM コントローラはアクティブな行をプリチャージした後に最初の行をアクティブ化し、新しい読み出し／書き込みコマンドを開始します。
  - b) 最初の行が既にアクティブ化されている場合、SDRAM コントローラは読み出し／書き込みコマンドを開始するのみです。

## SDRAM コントローラリフレッシュサイクル

自動リフレッシュコマンドは、SDRAM デバイスの値をリフレッシュするときに使います。SDRAM コントローラは、自動リフレッシュコマンドを周期的に発行します。内部カウンタに、FMC\_SDRTR レジスタの COUNT 値がロードされます。この値は、リフレッシュサイクル間のメモリクロックサイクル数を決定します (リフレッシュレート)。このカウンタがゼロになると、内部パルスが発生されます。

メモリアクセスが進行中である場合、自動リフレッシュリクエストは遅延されます。ただし、メモリアクセスと自動リフレッシュリクエストが同時に発生した場合は、自動リフレッシュリクエストが優先されます。

メモリアクセスが自動リフレッシュ動作中に発生すると、リクエストはバッファされて、自動リフレッシュが完了した後に処理されます。

前のリクエストの処理中に、新しい自動リフレッシュリクエストが発生すると、ステータスレジスタの RE (リフレッシュエラー) ビットがセットされます。有効化されている場合 (REIE = "1")、割り込みが発生します。

SDRAM ラインがアイドル状態でない (すべての行がクローズしていない) 場合、SDRAM コントローラは自動リフレッシュの前に PALL (プリチャージ ALL) コマンドを発行します。

FMC\_SDCMR コマンドモードレジスタから自動リフレッシュコマンドが発生されると (MODE ビット = "011")、最初に PALL コマンド (MODE ビット = "010") を発行する必要があります。

### 13.7.4 低電力モード

次の 2 つの低消費電力モードがあります。

- セルフリフレッシュモード  
SDRAM デバイス自体により自動リフレッシュサイクルが実行されて、外部クロックなしでデータを保持します。
- パワーダウンモード  
自動リフレッシュサイクルは SDRAM コントローラにより実行されます。

#### セルフリフレッシュモード

MODE ビットに "101" を設定し、かつ FMC\_SDCMR レジスタの目標バンクビット (CTB1 および/または CTB2) を設定すると、このモードが選択されます。

SDRAM クロックは TRAS 遅延後に停止し、内部リフレッシュタイマは次の条件をみたすときにカウントを停止します。

- セルフリフレッシュコマンドが両デバイスに発行される。
- 一方のデバイスは非アクティブである (SDRAM バンクが初期化されていない)。

セルフリフレッシュモードになる前に、SDRAM コントローラは PALL コマンドを自動的に発行します。

書き込みデータ FIFO が非エンプティの場合、すべてのデータがメモリに送られた後にセルフリフレッシュモードが開始され、BUSY ステータスフラグはセットされたままになります。

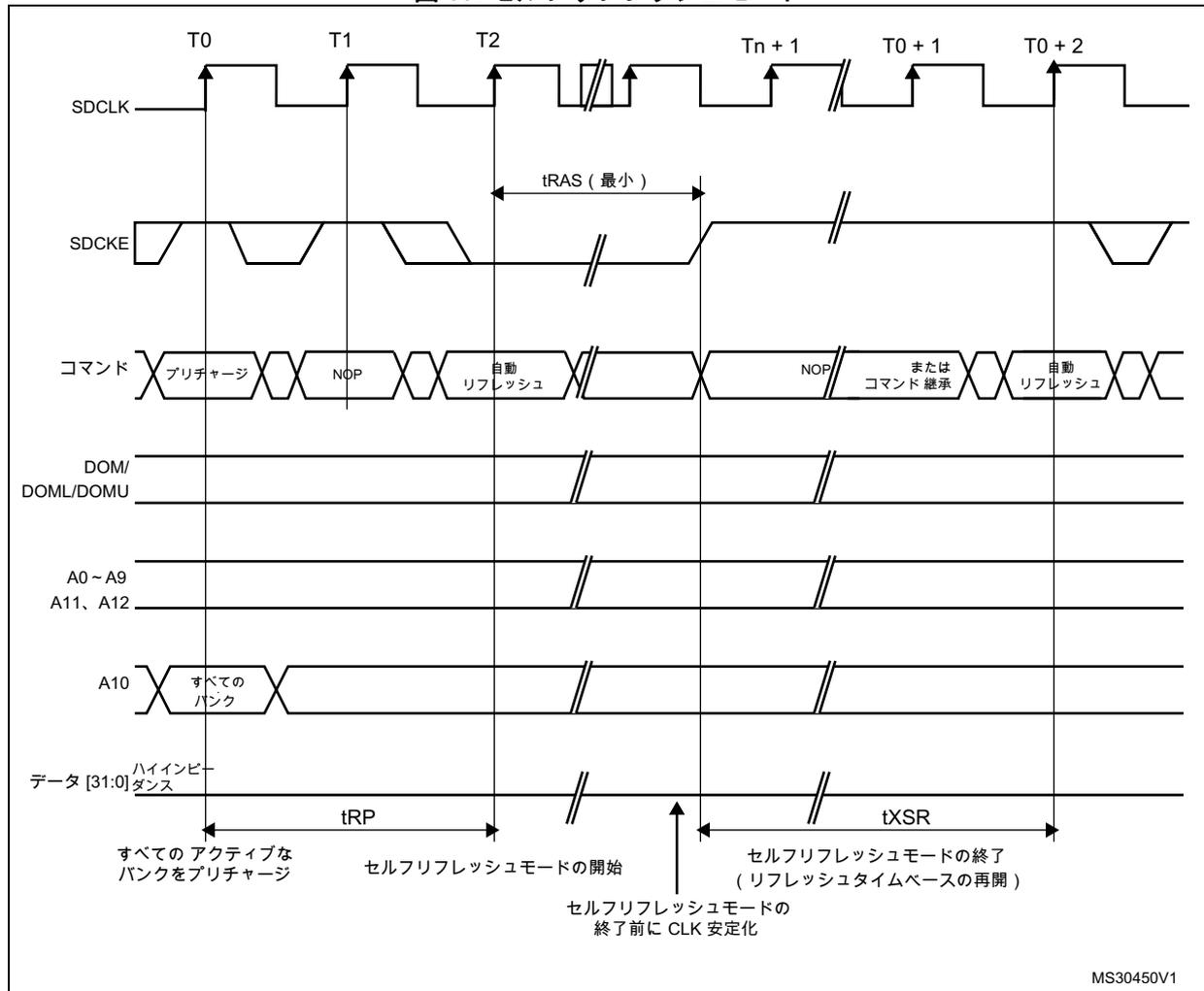
セルフリフレッシュモードでは、ローを維持する SDCKE 以外のすべての SDRAM デバイス入力が無視されます。

SDRAM デバイスは最小時間 TRAS の間セルフリフレッシュモードに留まる必要があり、無限時間セルフリフレッシュモードを維持することもできます。この最小時間を保証するため、TRAS 遅延中にセルフリフレッシュを開始した後、BUSY ステータスフラグはハイを維持します。

SDRAM デバイスを選択すると直ちに、SDRAM コントローラはセルフリフレッシュモードを終わらせる一連のコマンドを発生します。メモリアクセスの後、選択されたデバイスは通常モードに留まります。

セルフリフレッシュを終了するときは、MODE ビットに"000" (通常モード) を設定し、FMC\_SDCMR レジスタの目標バンクビット (CTB1 および/または CTB2) を設定する必要があります。

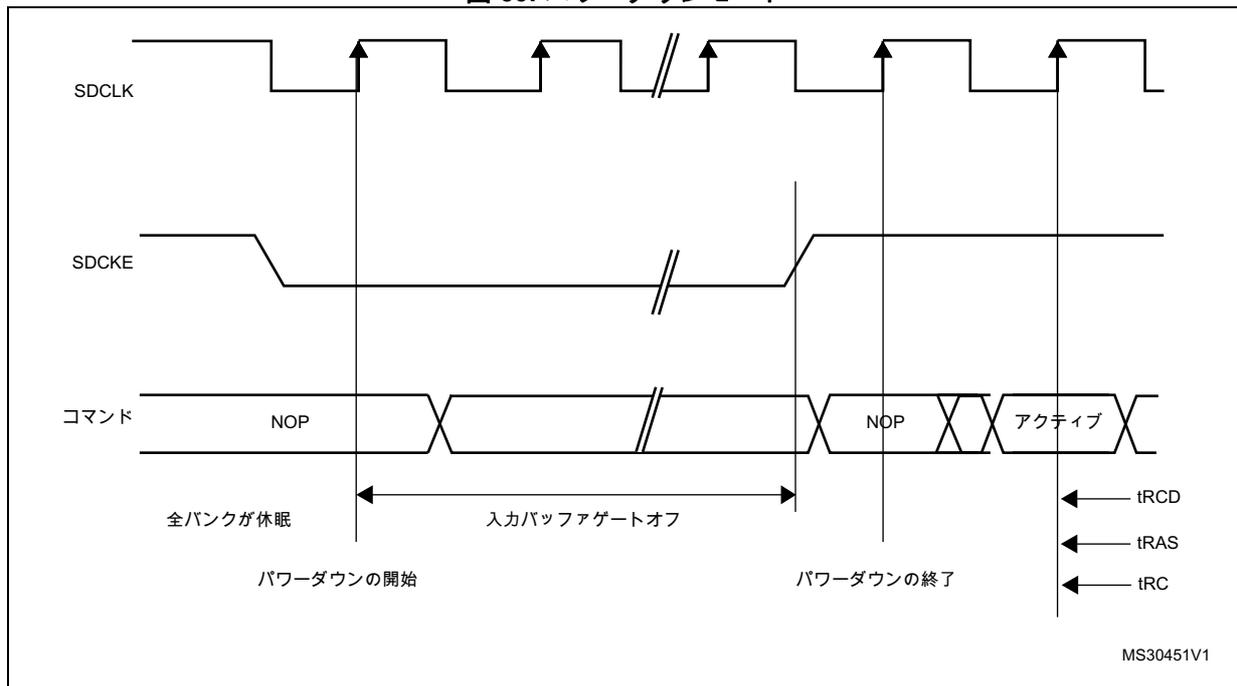
図 59. セルフリフレッシュモード



## パワーダウンモード

MODE ビットに"110"を設定し、かつ FMC\_SDCMR レジスタの目標バンクビット (CTB1 および/または CTB2) を設定すると、このモードが選択されます。

図 60. パワーダウンモード



書き込みデータ FIFO が非エンptyの場合、すべてのデータがメモリに送られた後にパワーダウンモードが開始されます。

SDRAM デバイスが選択されると直ちに、SDRAM コントローラはパワーダウンモードを終了します。メモリアクセスの後、選択されたデバイスは通常モードに留まります。

パワーダウンモード時、ローを維持する SDCKE 以外のすべての SDRAM デバイス入力と出力バッファは非アクティブになります。

SDRAM デバイスはリフレッシュ周期より長い間パワーダウンモードに留まることができないため、自分自身で自動リフレッシュサイクルを実行できません。したがって、SDRAM コントローラは次の動作によりリフレッシュ動作を実行します。

1. パワーダウンモードを終了し、SDCKE をハイに駆動します。
2. パワーダウンモード中に行がアクティブであった場合にのみ PALL コマンドを発生します。
3. 自動リフレッシュコマンドを発生します。
4. 再度 SDCKE をローに駆動し、パワーダウンモードに戻ります。

パワーダウンモードを終了するときは、MODE ビットに"000" (通常モード) を設定し、FMC\_SDCMR レジスタの目標バンクビット (CTB1 および/または CTB2) を設定する必要があります。

## 13.7.5 SDRAM コントローラレジスタ

### SDRAM 制御レジスタ 1、2 (FMC\_SDCR1、2)

アドレスオフセット :  $0x140 + 4 * (x - 1)$ ,  $x = 1, 2$

リセット値 :  $0x0000\ 02D0$

このレジスタは各 SDRAM メモリバンクの制御パラメータを格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	RPIPE		RBURST	SDCLK		WP	CAS		NB	MWID		NR		NC	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:15 予約済みであり、リセット値のままにしておかなければなりません。

#### ビット 14:13 RPIPE[1:0] : 読み出しパイプ

これらのビットは、CAS遅延後のデータ読み出しに対して、HCLK クロックサイクル数で遅延を指定します。

- 00 : HCLK クロックサイクル遅延なし
- 01 : 1 HCLK クロックサイクル遅延
- 10 : 2 HCLK クロックサイクル遅延
- 11 : 予約済み

**注 :** FMC\_SDCR2 レジスタの対応するビットは無視されます。

#### ビット 12 RBURST : パースト読み出し

このビットはパースト読み出しモードを有効化します。SDRAM コントローラは CAS遅延中に次の読み出しコマンドを予測し、データを読み出し FIFO に保存します。

- 0 : シングル読み出しリクエストをパーストとして管理しません。
- 1 : シングル読み出しリクエストを常にパーストとして管理します。

**注 :** FMC\_SDCR2 レジスタの対応するビットは無視されます。

#### ビット 11:10 SDCLK[1:0] : SDRAM クロック設定

これらのビットは両 SDRAM バンクの SDRAM クロック周期を指定し、周波数の変更前にクロックの無効化を可能にします。この場合、SDRAM の再初期化が必要です。

- 00 : SDCLK クロックを無効化
- 01 : 予約済み
- 10 : SDCLK 周期 = 2 x HCLK 周期
- 11 : SDCLK 周期 = 3 x HCLK 周期

**注 :** FMC\_SDCR2 レジスタの対応するビットは無視されます。

#### ビット 9 WP : 書き込み保護機能

このビットは、SDRAM バンクに対する書き込みモードアクセスを有効化します。

- 0 : 書き込みアクセスを許可
- 1 : 書き込みアクセスを無視

#### ビット 8:7 CAS[1:0] : CAS 遅延

このビットは、メモリクロックサイクル数で SDRAM CAS遅延を設定します。

- 00 : 予約済み
- 01 : 1 サイクル
- 10 : 2 サイクル
- 11 : 3 サイクル



**ビット 6 NB : 内部バンク数**

このビットは内部バンク数を設定します。

- 0 : 2 内部バンク
- 1 : 4 内部バンク

**ビット 5:4 MWID[1:0] : メモリデータバス幅**

これらのビットはメモリデバイス幅を指定します。

- 00 : 8 ビット
- 01 : 16 ビット
- 10 : 32 ビット
- 11 : 予約済み

**ビット 3:2 NR[1:0] : 行アドレスビット数**

これらのビットは、行アドレスのビット数を指定します。

- 00 : 11 ビット
- 01 : 12 ビット
- 10 : 13 ビット
- 11 : 予約済み

**ビット 1:0 NC[1:0] : 列アドレスビット数**

これらのビットは、列アドレスのビット数を指定します。

- 00 : 8 ビット
- 01 : 9 ビット
- 10 : 10 ビット
- 11 : 11 ビット。

**注 :** *RBURST* または *RPIPE* の設定値を変更する前、または *SDCLK* クロックを無効化する前に、先に *PALL* コマンドを送信して実行中の動作の完了を確認する必要があります。

## SDRAM タイミングレジスタ 1、2 (FMC\_SDTR1、2)

アドレスオフセット :  $0x148 + 4 * (x - 1)$ ,  $x = 1, 2$

リセット値 : 0x0FFF FFFF

このレジスタは各 SDRAM バンクのタイミングパラメータを格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TRCD				TRP				TWR			
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRC				TRAS				TXSR				TMRD			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

**ビット 27:24 TRCD[3:0] : 行から列までの遅延**

これらのビットは、アクティブコマンドと読み出し／書き込みコマンドとの間の遅延をメモリクロックサイクル数で指定します。

- 0000 : 1 サイクル
- 0001 : 2 サイクル
- ....
- 1111 : 16 サイクル

**ビット 23:20 TRP[3:0]: 行プリチャージ遅延**

これらのビットは、プリチャージコマンドと別のコマンドとの間の遅延をメモリクロックサイクル数で指定します。TRP タイミングは FMC\_SDTR1 レジスタでのみ設定されます。2 つの SDRAM デバイスを使用する場合、TRP には低速側デバイスのタイミングを設定する必要があります。

0000 : 1 サイクル

0001 : 2 サイクル

....

1111 : 16 サイクル

**注:** FMC\_SDTR2 レジスタの対応するビットは無視されます。

**ビット 19:16 TWR[3:0]: 復帰遅延**

これらのビットは、書き込みコマンドとプリチャージコマンドとの間の遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

....

1111 : 16 サイクル

**注:** TWR は SDRAM のデータシートで規定される書き込み復帰時間( $t_{WR}$ )に一致するようにプログラムし、次を保証する必要があります。

$TWR \geq TRAS - TRCD$  かつ  $TWR \geq TRC - TRCD - TRP$

**例:** TRAS = 4 サイクル、TRCD = 2 サイクル。したがって、TWR  $\geq$  2 サイクル。TWR には 0x1 を設定する必要があります。

2 つの SDRAM デバイスを使用する場合、FMC\_SDTR1 および FMC\_SDTR2 には、より低速な SDRAM デバイスと同じ TWR タイミングを設定する必要があります。

**ビット 15:12 TRC[3:0]: 行サイクル遅延**

これらのビットは、リフレッシュコマンドとアクティブコマンドとの間の遅延、および連続する 2 つのリフレッシュコマンドの間の遅延をそれぞれ定義します。メモリクロックサイクル数で表されます。TRC タイミングは FMC\_SDTR1 レジスタでのみ設定されます。2 つの SDRAM デバイスを使用する場合、TRC には低速側デバイスのタイミングを設定する必要があります。

0000 : 1 サイクル

0001 : 2 サイクル

....

1111 : 16 サイクル

**注:** TRC は SDRAM デバイスデータシートで規定される TRC と TRFC (自動リフレッシュ周期) タイミングに一致する必要があります。

**注:** FMC\_SDTR2 レジスタの対応するビットは無視されます。

**ビット 11:8 TRAS[3:0]: セルフリフレッシュ時間**

これらのビットは、メモリクロックサイクルで最小セルフリフレッシュ周期を指定します。

0000 : 1 サイクル

0001 : 2 サイクル

....

1111 : 16 サイクル

**ビット 7:4 TXSR[3:0] : セルフリフレッシュ遅延の終了**

これらのビットは、セルフリフレッシュコマンドの解除からアクティブコマンド発行までの遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

....

1111 : 16 サイクル

**注 :** 2 つの SDRAM デバイスを使用する場合、FMC\_SDTR1 および FMC\_SDTR2 には、より低速な SDRAM デバイスと同じ TXSR タイミングを設定する必要があります。

**ビット 3:0 TMRD[3:0] : ロードモードレジスタのアクティブまでの時間**

これらのビットは、ロードモードレジスタコマンドとアクティブまたはリフレッシュコマンドとの間の遅延をメモリクロックサイクル数で指定します。

0000 : 1 サイクル

0001 : 2 サイクル

....

1111 : 16 サイクル

**注 :** 2 つの SDRAM デバイスを接続する場合は、コマンドモードレジスタにより両デバイスに同時に実行されるすべてのアクセス (ロードモードレジスタコマンド) は、FMC\_SDTR1 レジスタでバンク 1 に対して設定されたタイミングパラメータ (TMRD タイミング、TRAS タイミング) を使って発行されます。TRP タイミングと TRC タイミングは FMC\_SDTR1 レジスタでのみ設定されます。2 つの SDRAM デバイスを使用する場合、TRP と TRC には低速側デバイスのタイミングを設定する必要があります。

## SDRAM コマンドモードレジスタ (FMC\_SDCMR)

アドレスオフセット : 0x150

リセット値 : 0x0000 0000

このレジスタは、SDRAM デバイスをアクセスするときに発行されたコマンドを格納します。このレジスタを使って、SDRAM デバイスの初期化、セルフリフレッシュの開始、パワーダウンモードの開始を行います。MODE フィールドに書き込みを行うと直ちに、CTB1 コマンドビットと CTB2 コマンドビットに応じて一方または両方の SDRAM バンクへコマンドが発行されます。このレジスタは、両 SDRAM バンクに対して同じです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MRD														
										rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRD							NRFS				CTB1	CTB2	MODE		
rW	rW	rW	rW	rW	rW	rW									

**ビット 31:22 予約済みであり、リセット値のままにしておかなければなりません。**

**ビット 21:9 MRD[12:0] : モードレジスタの定義**

この 13 ビットフィールドは、SDRAM モードレジスタ値を指定します。モードレジスタは、ロードモードレジスタコマンドを使って設定します。

**ビット 8:5 NRFS[3:0] : 自動リフレッシュ数**

これらのビットは、MODE = "011"のときに発行される連続自動リフレッシュコマンド数を指定します。

- 0000 : 1自動リフレッシュサイクル
- 0001 : 2自動リフレッシュサイクル
- ....
- 1110 : 15自動リフレッシュサイクル
- 1111 : 16自動リフレッシュサイクル

**ビット 4 CTB1 : コマンドターゲットバンク 1**

このビットは、SDRAM バンク 1 へコマンドを発行するか否かを指定します。

- 0 : SDRAM バンク 1 へコマンドを発行しない
- 1 : SDRAM バンク 1 へコマンドを発行する

**ビット 3 CTB2 : コマンドターゲットバンク 2**

このビットは、SDRAM バンク 2 へコマンドを発行するか否かを指定します。

- 0 : SDRAM バンク 2 へコマンドを発行しない
- 1 : SDRAM バンク 2 へコマンドを発行する

**ビット 2:0 MODE[2:0] : コマンドモード**

これらのビットは、SDRAM デバイスへ発行するコマンドを指定します。

- 000 : 通常モード
- 001 : クロック設定イネーブル
- 010 : PALL (「全バンクプリチャージ」) コマンド
- 011 : 自動リフレッシュコマンド
- 100 : ロードモードレジスタ
- 101 : セルフリフレッシュコマンド
- 110 : パワーダウンコマンド
- 111 : 予約済み

**注 :** コマンドを発行する場合、少なくとも 1 つのコマンド目標バンクビット (CTB1 または CTB2) をセットする必要があります。そうしないと、そのコマンドは無視されます。

**注 :** 2 つの SDRAM バンクを使用する場合、自動リフレッシュコマンドおよび PALL コマンドを CTB1 および CTB2 ビットがセットされた 2 つにデバイスに対して同時に発行する必要があります。そうしないと、そのコマンドは無視されます。

**注 :** 1 つの SDRAM バンクのみを使用し、該当する CTB ビットをセットした状態でコマンドを発行する場合、もう一方の使用されていないバンクの CTB ビットを 0 に保持する必要があります。

## SDRAM リフレッシュタイマレジスタ (FMC\_SDRTR)

アドレスオフセット : 0x154

リセット値 : 0x0000 0000

このレジスタは、リフレッシュタイマカウント値を設定することにより、リフレッシュサイクル間のSDCLK クロックサイクル数でリフレッシュレートを設定します。

$$\text{リフレッシュレート} = (\text{COUNT} + 1) \times \text{SDRAMクロック周波数}$$

$$\text{COUNT} = (\text{SDRAMリフレッシュ周期} / \text{行数}) - 20$$

### 例

$$\text{リフレッシュレート} = 64 \text{ ms} / (8196 \text{ 行}) = 7.81 \mu\text{s}$$

ここで、64 ms は SDRAM リフレッシュ周期です。

$$7.81 \mu\text{s} \times 60 \text{ MHz} = 468.6$$

読み出しリクエストが受理されているときに内部リフレッシュリクエストが発生した場合には、安全マージンを確保するためリフレッシュレートを 20 SDRAM クロックサイクルだけ大きくする必要があります (上の例の場合)。これは、COUNT 値 "0000111000000" (448) に対応します。

この 13 ビットフィールドは、SDRAM クロックを使ってデクリメントされるタイマにロードされません。このタイマはゼロになると、リフレッシュパルスが発生します。COUNT 値は、少なくとも 40 SDRAM クロックサイクルである必要があります。

FMC\_SDRTR レジスタが設定されると直ちに、タイマはカウントを開始します。レジスタに設定された値が "0" の場合、リフレッシュは実行されません。初期化の後にこのレジスタを再設定して、リフレッシュレートが変更されるのを回避する必要があります。

リフレッシュパルスが発生されるたびに、この 13 ビット COUNT フィールドがカウンタへ再ロードされます。

メモリアクセスが進行中の場合、自動リフレッシュリクエストは遅延させられます。ただし、メモリアクセスと自動リフレッシュリクエストが同時に発生した場合は、自動リフレッシュリクエストが優先されます。メモリアクセスがリフレッシュ動作中に発生すると、リクエストはバッファされて、リフレッシュが完了した後に処理されます。

このレジスタは、SDRAM バンク 1 と SDRAM バンク 2 に共通です。

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

ビット 31:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14 **REIE** : RES 割り込み有効化

0 : 割り込みを無効化します

1 : RE = 1 の場合に割り込みが発生します

ビット 13:1 **COUNT[12:0]** : リフレッシュタイマカウンタ

この 13 ビットフィールドは、SDRAM デバイスのリフレッシュレートを指定します。メモリクロックサイクル数で表されます。この値は、少なくとも 40 SDRAM クロックサイクル (0x28) である必要があります。

$$\text{リフレッシュレート} = (\text{COUNT} + 1) * \text{SDRAM クロック周波数}$$

$$\text{COUNT} = (\text{SDRAM リフレッシュ周期} / \text{行数}) - 20$$

ビット 0 **CRE** : リフレッシュエラーフラグのクリア

このビットを使って、ステータスレジスタのリフレッシュエラーフラグ (RE) をクリアします。

0 : 影響なし。

1 : リフレッシュエラーフラグをクリア

**注 :** *プログラムされた COUNT 値は、以下で求めるタイミングの合計とイコールであってはなりません。  
TWR + TRP + TRC + TRCD + 4 メモリクロックサイクル*

## SDRAM ステータスレジスタ (FMC\_SDSR)

アドレスオフセット : 0x158

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	BUSY	MODES2		MODES1		RE									
										r	r	r	r	r	r

ビット 31:5 **予約済みであり、リセット値のままにしておかなければなりません。**

ビット 5 **BUSY** : ビジーステータス

このビットは、コマンドモードリクエスト後の SDRAM コントローラのステータスを指定します。

0 : SDRAM コントローラは新しいリクエスト受付の準備完了

1 : SDRAM コントローラは新しいリクエスト受付の準備未完了

ビット 4:3 **MODES2[1:0]** : **バンク 2 のステータスモード**

このビットは、SDRAM バンク 2 のステータスモードを指定します。

00 : 通常モード

01 : セルフリフレッシュモード

10 : パワーダウンモード

ビット 2:1 **MODES1[1:0]** : **バンク 1 のステータスモード**

このビットは、SDRAM バンク 1 のステータスモードを指定します。

00 : 通常モード

01 : セルフリフレッシュモード

10 : パワーダウンモード

ビット 0 **RE** : リフレッシュエラーフラグ

0 : リフレッシュエラーは検出されない

1 : リフレッシュエラーが検出された

REIE = 1 かつ RE = 1 の場合割り込みが発生します。



## 13.8 FMC レジスタマップ

次の表に FMC レジスタの一覧を示します。

表 87. FMC レジスタマップ

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x00	FMC_BCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]	ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]	MTYP [1:0]	MUXEN	MBKEN									
	リセット値											0	0	0	0	0	0	1	1	0	0	0	0	1	0	1	1	0	1	1	0	1	1					
0x08	FMC_BCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]	ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]	MTYP [1:0]	MUXEN	MBKEN									
	リセット値													0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0					
0x10	FMC_BCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]	ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]	MTYP [1:0]	MUXEN	MBKEN									
	リセット値													0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0					
0x18	FMC_BCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]	ASYNCAWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]	MTYP [1:0]	MUXEN	MBKEN									
	リセット値													0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0					
0x04	FMC_BTR1	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x0C	FMC_BTR2	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
0x14	FMC_BTR3	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x1C	FMC_BTR4	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]		CLKDIV[3:0]		BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x104	FMC_BWTR1	Res.	Res.	ACCMOD[1:0]						BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x10C	FMC_BWTR2	Res.	Res.	ACCMOD[1:0]						BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x114	FMC_BWTR3	Res.	Res.	ACCMOD[1:0]						BUSTURN[3:0]		DATAST[7:0]		ADDHLD[3:0]		ADDSET[3:0]																						
	リセット値			0	0					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

表 87. FMC レジスタマップ (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x11C	FMC_BWTR4	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]			DATAST[7:0]					ADDHLD[3:0]			ADDSET[3:0]								
	リセット値			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x80	FMC_PCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS[2:0]		TAR[3:0]			TCLR[3:0]			Res.	Res.	Res.	ECCEN	PWID[1:0]	PTYP	PBKEN	PWAITEN	Res.			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
0x84	FMC_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS	
	リセット値																									1	0	0	0	0	0	0	
0x88	FMC_PMEM	MEMHIZx[7:0]					MEMHOLDx[7:0]					MEMWAITx[7:0]					MEMSETx[7:0]																
	リセット値	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0
0x8C	FMC_PATT	ATTHIZ[7:0]					ATTHOLD[7:0]					ATTWAIT[7:0]					ATTSET[7:0]																
	リセット値	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x94	FMC_ECCR	ECCx[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	FMC_SDCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIPE[1:0]	SDCLK[1:0]	WP	CAS[1:0]	NB	MWID[1:0]	NR[1:0]	NC							
	リセット値																		0	0	1	1	0	1	0	0	0	0	0	0	0	0	
0x144	FMC_SDCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WP	CAS[1:0]	NB	MWID[1:0]	NR[1:0]	NC						
	リセット値																					0	1	1	0	1	0	0	0	0	0	0	
0x148	FMC_SDTR1	Res.	Res.	Res.	Res.	TRCD[3:0]			TRP[3:0]			TWR[3:0]			TRC[3:0]			TRAS[3:0]			TXSR[3:0]			TMRD[3:0]									
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x14C	FMC_SDTR2	Res.	Res.	Res.	Res.	TRCD[3:0]			TRP[3:0]			TWR[3:0]			TRC[3:0]			TRAS[3:0]			TXSR[3:0]			TMRD[3:0]									
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x150	FMC_SDCMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRD[12:0]								NRFS[3:0]			CTB1	CTB2	MODE[2:0]						
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x154	FMC_SDRTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REIE	COUNT[12:0]												CRE	
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x158	FMC_SDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	MODES2[1:0]		MODES1[1:0]		Res.
	リセット値																										0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。



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

## 14 Quad SPI インタフェース (QUADSPI)

### 14.1 概要

QUADSPI は、シングル、デュアル、または Quad SPI フラッシュメモリを対象とする特殊な通信インタフェースです。このインタフェースは、次の3つのモードのいずれかで動作できます。

- インダイレクトモード：すべての動作は QUADSPI レジスタを使用して実行されます。
- ステータスポーリングモード：外部フラッシュメモリステータスレジスタが定期的に読み出され、フラグのセット時に割り込みを生成することができます。
- メモリマップドモード：外部フラッシュメモリはマイクロコントローラのアドレス空間に配置され、システムによって内部メモリであるかのようにみなされます。

デュアルフラッシュモードを使用すると、スループットと容量の両方を2倍に増やすことができます。このモードでは、2つの Quad SPI フラッシュメモリに同時にアクセスします。

### 14.2 QUADSPI の主な機能

- 3つの機能モード：インダイレクト、ステータスポーリング、メモリマップド
- 2つのフラッシュメモリに並列にアクセスすることで8ビットの同時送受信を可能にするデュアルフラッシュモード
- SDR および DDR サポート
- インダイレクトモードとメモリマップドモードの両方で完全にプログラム可能な OP コード
- インダイレクトモードとメモリマップドモードの両方で完全にプログラム可能なフレームフォーマット
- 受信および送信用の内蔵 FIFO
- 8、16、32ビットのデータアクセスを許可
- インダイレクトモードで動作するための DMA チャンネル
- FIFO 閾値、タイムアウト、動作完了、アクセスエラーによる割り込み生成

### 14.3 QUADSPI の機能説明

#### 14.3.1 QUADSPI ブロック図

図 61. QUADSPI ブロック図 (デュアルフラッシュモードが無効のとき)

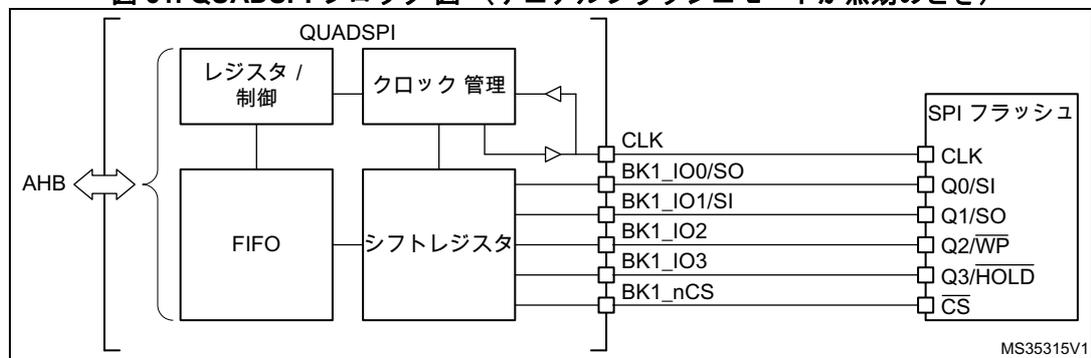
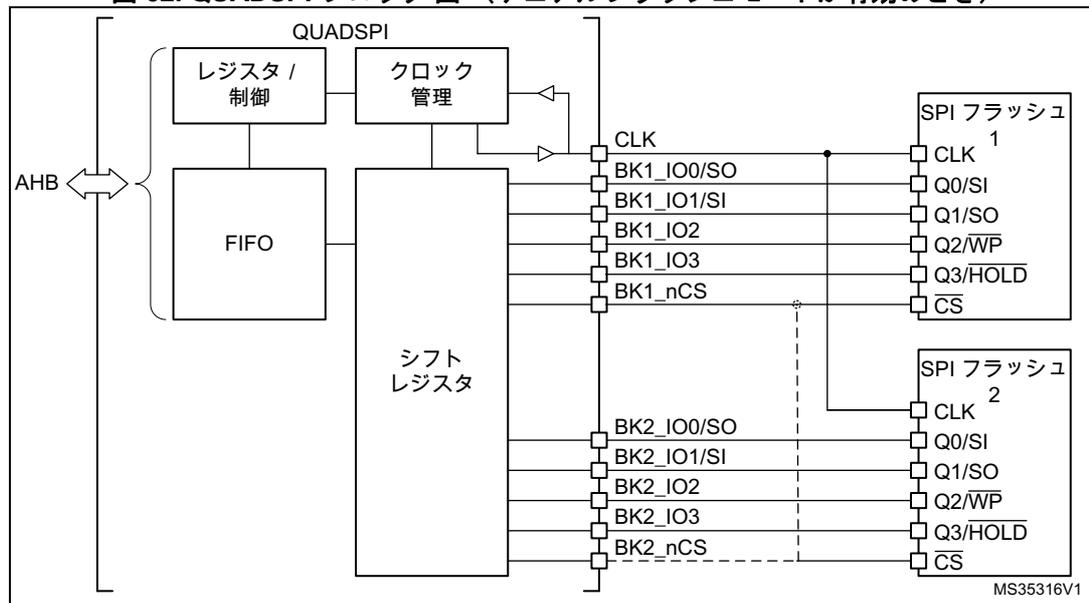


図 62. QUADSPI ブロック 図 (デュアルフラッシュモードが有効のとき)



QUADSPI は、シングルフラッシュメモリとのインタフェースでは 6 つの信号、デュアルフラッシュモードの 2 つのフラッシュメモリ (フラッシュ 1 および フラッシュ 2) とのインタフェースでは 10 ~11 の信号を使用します。

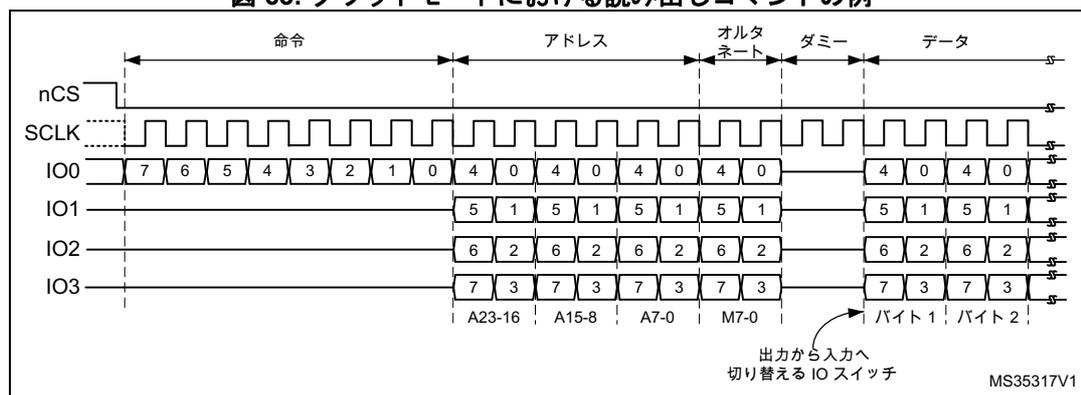
- CLK - クロック出力 (フラッシュ 1 およびフラッシュ 2)
- BK1\_IO0/SO - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル出力 (フラッシュ 1 のみ)
- BK1\_IO1/SI - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル入力 (フラッシュ 1 のみ)
- BK1\_IO2 - クワッドモードでの双方向入出力 (フラッシュ 1 のみ)
- BK1\_IO3 - クワッドモードでの双方向入出力 (フラッシュ 1 のみ)
- BK2\_IO0/SO - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル出力 (フラッシュ 2 のみ)
- BK2\_IO1/SI - デュアル/クワッドモードでの双方向入出力 またはシングルモードでのシリアル入力 (フラッシュ 2 のみ)
- BK2\_IO2 - クワッドモードでの双方向入出力 (フラッシュ 2 のみ)
- BK2\_IO3 - クワッドモードでの双方向入出力 (フラッシュ 2 のみ)
- BK1\_nCS - チップセレクト出力 (アクティブロー) (フラッシュ 1 のみ)。デュアルフラッシュモードで QUADSPI が常時使用される場合は、フラッシュ 2 にも使用可能
- BK2\_nCS - チップセレクト出力 (アクティブロー) (フラッシュ 2 のみ)。デュアルフラッシュモードで QUADSPI が常時使用される場合は、フラッシュ 1 にも使用可能

## 14.3.2 QUADSPI コマンドシーケンス

QUADSPI はコマンドを使用してフラッシュメモリと通信します。各コマンドには、命令、アドレス、オルタネートバイト、ダミー、データの 5 つのフェーズが含まれます。これらのフェーズのいずれかをスキップするように設定することができますが、命令、アドレス、オルタネートバイト、データの少なくとも 1 つのフェーズが必要です。

nCS は各コマンドの開始前に立ち下がり、各コマンドの終了前に再び立ち上がります。

図 63. クワッドモードにおける読み出しコマンドの例



### 命令フェーズ

このフェーズでは、QUADSPI\_CCR[7:0] レジスタの INSTRUCTION フィールドに設定された 8 ビットの命令がフラッシュメモリに送信され、実行すべき動作のタイプが指定されます。

ほとんどのフラッシュメモリは IO0/SO 信号 (シングル SPI モード) から同時に 1 ビットの命令しか受信できませんが、命令フェーズでは、オプションで、同時に 2 ビット (デュアル SPI モードの IO0/IO1 経由) または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) を送信することができます。この設定には、QUADSPI\_CCR[9:8] レジスタの IMODE[1:0] フィールドを使用することができます。

IMODE = 00 の場合、命令フェーズはスキップされますので、コマンドシーケンスはアドレスフェーズが存在する場合はそこから開始されます。

### アドレスフェーズ

アドレスフェーズでは、動作のアドレスを示す 1~4 バイトがフラッシュメモリに送信されます。送信されるアドレスバイト数は、QUADSPI\_CCR[13:12] レジスタの ADSIZE[1:0] フィールドに設定されます。インダイレクトモードと自動ポーリングモードでは、送信されるアドレスバイトは QUADSPI\_AR レジスタの ADDRESS[31:0] フィールドに指定されます。一方、メモリマップドモードの場合は、アドレスは AHB (Cortex® から、または DMA から) 経由で直接与えられます。

アドレスフェーズで同時に送信できるのは、1 ビット (シングル SPI モードの SO 経由)、2 ビット (デュアル SPI モードの IO0/IO1 経由)、または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) です。この設定には、QUADSPI\_CCR[11:10] レジスタの ADMODE[1:0] フィールドを使用することができます。

ADMODE = 00 の場合、アドレスフェーズはスキップされますので、コマンドシーケンスは次のフェーズが存在する場合は直接そこに進みます。

## オルタネートバイトフェーズ

オルタネートバイトフェーズでは、一般に動作モードを制御する 1~4 バイトがフラッシュメモリに送信されます。送信されるオルタネートバイト数は、QUADSPI\_CCR[17:16] レジスタの ABSIZE[1:0] フィールドに設定されます。送信されるバイトは QUADSPI\_ABR レジスタで指定されます。

オルタネートバイトフェーズで同時に送信できるのは、1 ビット (シングル SPI モードの SO 経由)、2 ビット (デュアル SPI モードの IO0/IO1 経由)、または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) です。この設定には、QUADSPI\_CCR[15:14] レジスタの ABMODE[1:0] フィールドを使用することができます。

ABMODE = 00 の場合、オルタネートバイトフェーズはスキップされますので、コマンドシーケンスは次のフェーズが存在する場合は直接そこに進みます。

場合によっては、オルタネートバイトフェーズ中にフルバイトではなく 1 ニブルを送信する必要があるかもしれません。たとえば、デュアルモード使用時で、オルタネートバイトに 2 サイクルだけが使われる場合などです。このような場合、ファームウェアは、クワッドモード (ABMODE = 11) を使用し、ALTERNATE のビット 7 とビット 3 を "1" にセット (IO3 ラインをハイに保持) し、ビット 6 とビット 2 を "0" にセット (IO2 ラインをローに保持) して、1 バイトを送信することができます。送信されるニブルの上位 2 ビットは ALTERNATE のビット 4:3 にセットされ、下位 2 ビットはビット 1 および 0 にセットされます。たとえば、ニブル 2 (0010) が IO0/IO1 を経由して送信される場合、ALTERNATE は 0x8A (1000\_1010) にセットされる必要があります。

## ダミーサイクルフェーズ

ダミーサイクルフェーズでは、データの送受信は一切なく、1~31 サイクルが実行されます。これは、より高いクロック周波数が使われている場合に、データフェーズの準備をする時間をフラッシュメモリに与えるためです。このフェーズ中に実行されるサイクル数は、QUADSPI\_CCR[22:18] レジスタの DCYC[4:0] フィールドで指定されます。SDR モードと DDR モードのどちらでも、処理時間にはフル CLK サイクルの数が指定されます。

DCYC = 00 の場合、ダミーサイクルフェーズはスキップされますので、コマンドシーケンスはデータフェーズが存在する場合は直接そこに進みます。

ダミーサイクルフェーズの動作モードは DMODE によって決まります。

データ信号を出力モードから入力モードへ変更するのに十分な「ターンアラウンド」時間を確保するために、デュアルまたはクワッドモードを使用してフラッシュメモリからデータを受信する場合、少なくとも 1 ダミーサイクルが必要です。

## データフェーズ

データフェーズ中は、フラッシュメモリに対して何バイトでも送受信することができます。

インダイレクトモードと自動ポーリングモードでは、送受信するバイト数は QUADSPI\_DLR レジスタで指定されます。

インダイレクト書き込みモードでは、フラッシュメモリに送信されるデータを QUADSPI\_DR レジスタに書き込む必要があります。一方、インダイレクト読み出しモードでは、フラッシュメモリから受信するデータは QUADSPI\_DR レジスタから読み出して取得します。

メモリマップドモードでは、読み出されるデータは、直接 AHB を経由して Cortex または DMA へ送り戻されます。

データフェーズで同時に送受信できるのは、1 ビット (シングル SPI モードの SO/SI 経由)、2 ビット (デュアル SPI モードの IO0/IO1 経由)、または 4 ビット (Quad SPI モードの IO0/IO1/IO2/IO3 経由) です。この設定には、QUADSPI\_CCR[15:14] レジスタの ABMODE[1:0] フィールドを使用することができます。

DMODE = 00 の場合、データフェーズはスキップされますので、コマンドシーケンスは nCS を立ち上げることですぐに終了します。この設定はインダイレクト書き込みモードのみで使用してください。

## 14.3.3 QUADSPI シングルインタフェースプロトコルモード

### シングル SPI モード

レガシー SPI モードでは、1 ビットの順次送受信のみが可能です。このモードでは、データは SO 信号 (I/O を IO0 と共有) を経由してフラッシュメモリへ送信されます。フラッシュメモリからのデータは、SI 信号 (I/O を IO1 と共有) を経由して受信されます。

QUADSPI\_CCR レジスタのそれぞれのフェーズに対応するフィールド (IMODE/ADMODE/ABMODE/DMODE) を“01”にセットすることで、このシングルビットモードを使用できるよう、フェーズごとに個別に設定することができます。

各フェーズをシングルモードに設定する場合

- IO0 (SO) は出力モードです。
- IO1 (SI) は入力モードです (ハイインピーダンス)。
- IO2 は出力モードで、強制的に“0”に設定されます (書き込み保護機能を無効にするため)。
- IO3 は出力モードで、強制的に“1”に設定されます (ホールド機能を無効にするため)。

これは、DMODE = 01 であれば、ダミーフェーズにも該当します。

### デュアル SPI モード

デュアル SPI モードでは、IO0/IO1 信号を経由して 2 ビットが同時に送受信されます。

QUADSPI\_CCR レジスタのそれぞれのフェーズに対応するフィールド (IMODE/ADMODE/ABMODE/DMODE) を“10”にセットすることで、このデュアル SPI モードを使用できるよう、フェーズごとに個別に設定することができます。

各フェーズをデュアルモードに設定する場合

- IO0/IO1 は、データフェーズの読み出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。
- IO2 は出力モードで、強制的に“0”に設定されます。
- IO3 は出力モードで、強制的に“1”に設定されます。

ダミーフェーズでは、DMODE = 01 の場合、IO0/IO1 は常にハイインピーダンスです。

### Quad SPI モード

Quad SPI モードでは、IO0/IO1/IO2/IO3 信号を経由して 4 ビットが同時に送受信されます。

QUADSPI\_CCR レジスタのそれぞれのフェーズに対応するフィールド (IMODE/ADMODE/ABMODE/DMODE) を“11”にセットすることで、この Quad SPI モードを使用できるよう、フェーズごとに個別に設定することができます。

クワッドモードに設定されている各フェーズでは、IO0/IO1/IO2/IO3 はすべてデータフェーズの読み出し動作時はハイインピーダンス (入力) ですが、他のすべての場合は出力です。

ダミーフェーズでは、DMODE = 11 の場合、IO0/IO1/IO2/IO3 はすべて常にハイインピーダンスです。

IO2 と IO3 は Quad SPI モードでのみ使用されます。Quad SPI モードを使用できるように設定されたフェーズが 1 つもない場合、IO2 および IO3 に対応するピンは QUADSPI がアクティブなときでも、他の機能に使用することができます。

## SDR モード

DDRM ビット (QUADSPI\_CCR[31]) は、デフォルトで 0 に設定されており、QUADSPI はシングルデータレート (SDR) モードで動作します。

SDR モードでは、QUADSPI が IO0/SO、IO1、IO2、IO3 信号を駆動しているとき、これらの信号は CLK の立ち下がりエッジでのみ遷移します。

SDR モードでデータを受信する場合、QUADSPI はフラッシュメモリも CLK の立ち下がりエッジを使用してデータを送信するとみなします。デフォルトでは (SSHIFT = 0 の場合)、信号は次の CLK の (立ち上がり) エッジを使用してサンプリングされます。

## DDR モード

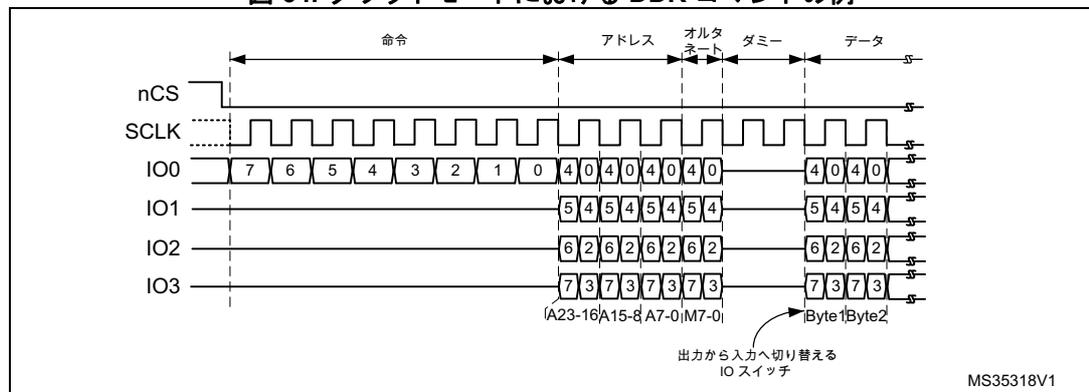
DDRM ビット (QUADSPI\_CCR[31]) が 1 にセットされている場合、QUADSPI はダブルデータレート (DDR) モードで動作します。

DDR モードでは、QUADSPI が アドレス、オルタネートバイト、またはデータフェーズで IO0/SO、IO1、IO2、IO3 信号を駆動しているとき、CLK の立ち下がりまたは立ち上がりエッジで 1 ビットが送信されます。

命令フェーズは DDRM の影響を受けません。命令は常に CLK の立ち下がりエッジを使用して送信されます。

DDR モードでデータを受信する場合、QUADSPI はフラッシュメモリも CLK の立ち上がりエッジと立ち下がりエッジの両方を使用してデータを送信するとみなします。DDRM = 1 の場合、ファームウェアは SSHIFT ビット (QUADSPI\_CR[4]) をクリアする必要があります。したがって、信号は 1/2 CLK サイクル後にサンプリングされます (下図の逆エッジで)。

図 64. クワッドモードにおける DDR コマンドの例



## デュアルフラッシュモード

DFM ビット (QUADSPI\_CR[6]) が 1 の場合、QUADSPI はデュアルフラッシュモードです。このモードでは、2 つの外部 Quad SPI フラッシュメモリ (フラッシュ 1 と フラッシュ 2) が使用されており、サイクルごとに 8 ビットの送受信 (DDR モードでは 16 ビット) を行うことで、スループットおよび容量を効果的に倍にします。

どちらのフラッシュメモリも同じ CLK を使用し、オプションで同じ nCS 信号を使用しますが、それぞれが個別の IO0、IO1、IO2、IO3 信号を有しています。

デュアルフラッシュモードは、SDR モードまたは DDR モードと組み合わせて使用できるだけでなく、シングルビットモード、デュアルビットモード、クワッドビットモードとも組み合わせて使用できます。

FSIZE[4:0] (QUADSPI\_DCR[20:16]) で指定されているフラッシュメモリのサイズは、フラッシュメモリの総容量 (コンポーネント 1 個のサイズの 2 倍) を反映していなければなりません。

アドレス X が偶数の場合、QUADSPI はアドレス X に対してフラッシュ 1 のアドレス X/2 にあるバイトを与え、アドレス X+1 に対してフラッシュ 2 のアドレス X/2 にあるバイトを与えます。言い換えれば、偶数アドレスにあるバイトはすべてフラッシュ 1 に、奇数アドレスにあるバイトはすべてフラッシュ 2 に格納されます。

デュアルフラッシュモードでフラッシュメモリステータスレジスタを読み出す場合、シングルフラッシュモードで読み出すバイト数の 2 倍のバイトを読み出す必要があります。つまり、各フラッシュメモリがステータスレジスタのフェッチ命令後に有効な 8 ビットを提供する場合、QUADSPI はデータ長 2 バイト (16 ビット) で設定する必要があります。それにより QUADSPI は各フラッシュメモリから 1 バイトを受信します。各フラッシュメモリが 16 ビットのステータスを提供する場合、QUADSPI は、4 バイトを読み出して、デュアルフラッシュモードの両方のフラッシュメモリのステータスビットをすべて取得できるように設定する必要があります。リザルト (データレジスタ内) の最下位バイトはフラッシュ 1 ステータスレジスタの最下位バイト、次のバイトはフラッシュ 2 ステータスレジスタの最下位バイトとなります。さらに、データレジスタの 3 番目のバイトはフラッシュ 1 の 2 番目のバイト、4 番目のバイトはフラッシュ 2 の 2 番目のバイトとなります (フラッシュメモリに 16 ビットステータスレジスタがある場合)。

偶数バイトは常にデュアルフラッシュモードでアクセスする必要があります。このため、DRM = 1 のとき、データ長フィールドのビット 0 (QUADSPI\_DLR[0]) は 1 のままとなります。

デュアルフラッシュモードでは、フラッシュ 1 インタフェース信号の動作は基本的に通常モードの動作と同じです。フラッシュ 2 インタフェース信号の波形は、命令、アドレス、オルタネートバイト、およびタミーサイクルフェーズの間は、フラッシュ 1 とまったく同じです。つまり、どちらのフラッシュメモリも常に同じ命令や同じアドレスを受信するということです。データフェーズ中は、2 つのバス BK1\_IOx と BK2\_IOx は並列してデータ転送を行います。フラッシュ 1 に対する送信 (または受信) データはフラッシュ 2 のものとは異なります。

### 14.3.4 QUADSPI インダイレクトモード

インダイレクトモードでは、QUADSPI レジスタに書き込むことでコマンドが開始され、データレジスタへの書き込み/読み出しによってデータが転送されます。その方法はその他の通信ペリフェラルの場合と同様です。

FMODE = 00 (QUADSPI\_CCR[27:26]) の場合、QUADSPI はインダイレクト書き込みモードです。このモードでは、データフェーズ中にフラッシュメモリへのバイトの送信が行われます。データは、データレジスタ (QUADSPI\_DR) への書き込みによって提供されます。

FMODE = 01 の場合、QUADSPI はインダイレクト読み出しモードです。このモードでは、データフェーズ中にフラッシュメモリからのバイトの受信が行われます。データは、QUADSPI\_DR の読み出しによって復旧されます。

送受信するバイト数はデータ長レジスタ (QUADSPI\_DLR) で指定されます。QUADSPI\_DLR = 0xFFFF\_FFFF (すべて“1”) の場合、データ長は不定長と見なされ、QUADSPI はフラッシュメモリの終了 (FSIZE によって定義される) までデータを転送し続けるのみです。転送されるバイトが無い場合は、DMODE (QUADSPI\_CCR[25:24]) を“00”にセットする必要があります。

QUADSPI\_DLR = 0xFFFF\_FFFF かつ FSIZE = 0x1F (4 GB のフラッシュメモリを示す最大値) の場合、このような特殊なケースでは、転送は無限に続けられ、アボートリクエスト後または QUADSPI の無効化後にやっと停止します。最後のメモリアドレスが読み出された後も (アドレス 0xFFFF\_FFFF で)、アドレス = 0x0000\_0000 で読み出しが続けられます。

プログラムされた送信または受信バイト数に達すると、TCF がセットされ、TCIE = 1 であれば割り込みが生成されます。データ数が定義されていない場合、外部 SPI メモリの限界に達した時点で、QUADSPI\_CR に定義されているフラッシュメモリのサイズに応じて TCF がセットされます。

## コマンド開始のトリガ

コマンドは、基本的に、ファームウェアがそのコマンドに必要な最期の情報を提供するとすぐに開始されます。インダイレクトモードでコマンドを開始する方法は、QUADSPI の設定に応じて、3 通りあります。コマンドは以下の動作の直後に開始されます。

1. アドレスを必要としない場合 (ADMODE = 00) およびファームウェアによるデータの提供を必要としない場合 (FMODE = 01 または DMODE = 00) は、INSTRUCTION[7:0] (QUADSPI\_CCR) への書き込み動作直後
2. アドレスを必要とする場合 (ADMODE != 00) およびファームウェアによるデータの提供を必要としない場合 (FMODE = 01 または DMODE = 00) は、ADDRESS[31:0] (QUADSPI\_AR) への書き込み動作直後
3. アドレスを必要とする場合 (ADMODE != 00) およびファームウェアによるデータの提供を必要とする場合 (FMODE = 00 および DMODE != 00) は、DATA[31:0] (QUADSPI\_DR) への書き込み動作直後

オルタネートバイトレジスタ (QUADSPI\_ABR) への書き込みが通信開始をトリガすることはありません。オルタネートバイトが必要な場合は、事前にそれらをプログラムする必要があります。

コマンドが開始されたらすぐに、BUSY ビット (QUADSPI\_SR の ビット5) が自動的にセットされます。

## FIFO およびデータの管理

インダイレクトモードでは、データは QUADSPI に内蔵されている 32 バイト FIFO を通ります。FLEVEL[5:0] (QUADSPI\_SR[13:8]) は、現在 FIFO に保持されているバイト数を示します。

インダイレクト書き込みモード (FMODE = 00) では、ファームウェアは QUADSPI\_DR に書き込む時点で FIFO にデータを追加します。書き込みにより FIFO に追加されるバイト数は、ワード書き込みで 4 バイト、ハーフワード書き込みで 2 バイト、バイト書き込みで 1 バイトです。ファームウェアが FIFO に追加するバイト数が多すぎる (DL[31:0] によって示される数より多い) 場合、余分なバイトは、書き込み動作の終了時 (TCF がセットされる時) に FIFO から一掃されます。

QUADSPI\_DR へのバイト/ハーフワードアクセスは、32 ビットレジスタの最下位バイト/ハーフワードにのみ実行されます。

FTHRES[3:0] を使用して FIFO 閾値が定義されます。閾値に達すると、FTF (FIFO 閾値フラグ) がセットされます。インダイレクト読み出しモードでは、FIFO から読み出される有効バイト数が閾値を超えると FTF がセットされます。また、FTHRES の設定に関わらず、フラッシュメモリから最後のバイトが読み出された後に FIFO 内にデータが残っている場合も FTF がセットされます。インダイレクト書き込みモードでは、FIFO 内の空のバイト数が閾値を超えると FTF がセットされます。

FTIE = 1 の場合、FTF がセットされた時点で割り込みが発生します。DMAEN = 1 の場合、FTF がセットされた時点で DMA 転送が開始されます。閾値条件が真でなくなる (CPU または DMA によって十分なデータが転送された後) とすぐに、HW によって FTF がクリアされます。

インダイレクト読み出しモードでは、FIFO が満杯になると、QUADSPI は一時的にフラッシュメモリからのバイトの読み出しを停止して、オーバーランを回避します。フラッシュメモリの読み出しは、FIFO 内の 4 バイトが空になるまで再開されませんのでご注意ください (FLEVEL ≤ 11 の場合)。したがって、FTHRES ≥ 13 の場合、QUADSPI は確実にフラッシュメモリからのデータ取得を再開できるものとして、アプリケーションによる十分なバイトの読み出しが行われる必要があります。そうしないと、11 < FLEVEL < FTHRES である限り、FTF フラグは 0 のままとなります。

## 14.3.5 QUADSPI ステータスフラグポーリングモード

自動ポーリングモードでは、QUADSPI は定期的にコマンドを開始させて、定義された数のステータスバイト（最大 4）を読み出します。受信バイトをマスクして、ステータスビットの一部を分離することができます。また、選択されたビットに値が定義されている場合は割り込みを生成することができます。

フラッシュメモリへのアクセス方法はインダイレクト読み出しモードの場合と同様です。アドレスを必要としない場合 (ADMODE = 00) は、QUADSPI\_CCR が書き込まれるとすぐにアクセスが開始されます。反対にアドレスを必要とする場合は、QUADSPI\_AR が書き込まれた時点で最初のアクセスが始まります。この時点で BUSY がハイになり、定期的なアクセスが実行される間もハイのままです。

MASK[31:0] (QUADSPI\_PSMAR) の内容を使用して、自動ポーリングモードにおけるフラッシュメモリからのデータがマスクされます。MASK[n] = 0 の場合、リザルトのビット n はマスクされ、考慮されません。MASK[n] = 1 かつ bit[n] の値が MATCH[n] (QUADSPI\_PSMAR) の値と同じである場合、bit n との一致が見られます。

ポーリング一致モードビット (PMM、QUADSPI\_CR[23]) が 0 の場合、「AND」一致モードがアクティブになります。つまり、ステータス一致フラグ (SMF) は、マスクされていないすべてのビットで一致が検出された場合にのみセットされます。

PMM = 1 の場合、「OR」一致モードがアクティブになります。つまり、SMF は、マスクされていない任意のビットで一致が検出された場合にセットされます。

SMIE = 1 の場合に SMF がセットされると、割り込みが呼び出されます。

自動ポーリングモード停止 (APMS) ビットがセットされている場合、一致が検出されるとすぐに動作は停止し、BUSY フラグが 0 になります。あるいは、BUSY は 1 のままで、アボートが検出される、または QUADSPI が無効化される (EN = 0) まで定期的なアクセスが継続します。

データレジスタ (QUADSPI\_DR) には、最後に受信したステータスバイト (FIFO の無効化) が含まれます。データレジスタの内容は、一致ロジックで使用されるマスクの影響を受けません。FTF ステータスビットは新たなステータスの読み出しが完了するとすぐにセットされ、データが読み出された直後にクリアされます。

## 14.3.6 QUADSPI メモリマップドモード

メモリマップドモードに設定された場合、外部 SPI デバイスは内部メモリとみなされます。

QUADSPI ペリフェラルを正しく設定し、有効にする前に、QUADSPI フラッシュバンク領域にアクセスすることは禁じられています。

フラッシュメモリの容量がもっと大きい場合でも、アドレス指定できるのは 256 MB までです。

FSIZE で規定されている範囲外のアドレスにアクセスすると、256 MB の範囲内であっても、AHB エラーが発生します。このエラーの影響は、アクセスを試みた AHB マスタに依存します。

- Cortex<sup>®</sup> CPU の場合は、ハードフォールト割り込みが発生します。
- DMA の場合は、DMA 転送エラーが発生し、対応する DMA チャネルが自動的に無効になります。

バイト、ハーフワード、およびワードアクセスはすべてサポートされます。

XIP (Execute In Place) 動作のサポートが実装されています。この場合、QUADSPI は次のマイクロコントローラのアクセスを予測し、あらかじめ次のアドレスにバイトをロードします。その後のアクセスが実際に連続したアドレスに対して行われる場合、値がすでにプリフェッチされているので、アクセスは高速で完了します。

デフォルトでは QUADSPI がプリフェッチ動作を停止することはなく、フラッシュメモリへのアクセスが長時間なくても、前の読み出し動作はアクティブな状態を保ち、nCs もローに保持されます。nCS が

ローに保持されているとフラッシュメモリの消費電力が増える傾向があるため、アプリケーションではタイムアウトカウンタをアクティブにしようとするかもしれません (TCEN = 1, QUADSPI\_CR[3])。これは、FIFO がプリフェッチしたデータで満杯になって以来アクセスが 1 回もないまま TIMEOUT[15:0] (QUADSPI\_LPTR) サイクルが経過した後に、nCS がリリースされるようにするためです。

最初のメモリマップドアクセスが発生するとすぐに、BUSY はハイになります。プリフェッチ動作の影響で、BUSY の立ち下がり、タイムアウトが起こる、アボートが検出される、またはペリフェラルが無効化されるまで起こりません。

## 14.3.7 QUADSPI フラッシュメモリの設定

デバイス設定レジスタ (QUADSPI\_DCR) を使用して、外部 SPI フラッシュメモリの特性を指定することができます。

FSIZE[4:0] フィールドは、次の式を使用して外部メモリのサイズを定義します。

$$\text{フラッシュメモリ内のバイト数} = 2^{\text{FSIZE}+1}$$

FSIZE+1 はフラッシュメモリのアドレスを指定するために必要とされる有効なアドレスビット数です。フラッシュメモリの容量はインダイレクトモードでは最大 4 GB (32 ビットを使用してアドレス指定) ですが、メモリマップドモードにおけるアドレス指定可能な空間は 256 MB に制限されます。

DFM = 1 の場合、FSIZE は 2 つのフラッシュメモリの総容量を示します。

QUADSPI が 2 つのコマンドを実行する場合、一方のコマンドはもう一方のコマンドの直後に実行され、デフォルトでは 1 CLK サイクル中の 2 つのコマンドの間にハイレベルのチップセレクト信号 (nCS) を立ち上げます。フラッシュメモリがコマンド間により長い時間を必要とする場合、チップセレクトハイ時間 (CSHT) フィールドを使用して、nCS がハイに保たなければならない CLK サイクルの回数 (最大 8) を指定することができます。

クロックモード (CKMODE) ビットは、コマンド間の CLK 信号ロジックレベルを示します (nCS = 1 の場合)。

## 14.3.8 QUADSPI 遅延データサンプリング

デフォルトでは、QUADSPI がフラッシュメモリに駆動されているデータのサンプリングを行うタイミングは、フラッシュメモリが信号を駆動してから 1/2 CLK サイクル後です。

外部信号の遅延が発生した場合、それは後でデータをサンプリングする際に有用となるかもしれません。SSHIFT ビット (QUADSPI\_CR[4]) を使用することで、データのサンプリングを 1/2 CLK サイクルだけシフトすることができます。

クロックのシフトは DDR モードではサポートされていないので、DDRM ビットがセットされた時点で SSHIFT ビットをクリアする必要があります。

## 14.3.9 QUADSPI の設定

QUADSPI の設定は次の 2 つのフェーズで行われます。

- QUADSPI IP の設定
- QUADSPI フラッシュメモリの設定

QUADSPI は、設定および有効化が行われると、インダイレクトモード、ステータスポーリングモード、メモリマップドモードの 3 つの動作モードのいずれかで使用することができます。

QUADSPI IP の設定



QUADSPI IP は QUADSPI\_CR を使用して設定します。ユーザは、受信データに対してクロックプリスケラ分周比とサンプリングシフトを設定します。

DDR モードは DDRM ビットによって設定することができます。このモードを有効にすると、アドレスおよびオルタネートバイトが両方のクロックエッジに送信され、両方のクロックエッジでデータの送受信が行われます。DDRM ビットの設定にかかわらず、命令は常に SDR モードで送信されます。

DMA リクエストは DMAEN ビットを設定することで有効化されます。割り込みを使用する場合、それぞれのイネーブルビットもこのフェーズ中にセットされます。

DMA リクエスト生成または割り込み生成の FIFO レベルは FTHRES ビットで設定されます。

タイムアウトの発生が必要な場合、TCEN ビットをセットして、QUADSPI\_LPTR レジスタにタイムアウト値を設定することができます。

デュアルフラッシュモードは DFM を “1” にセットすることで有効にできます。

## QUADSPI フラッシュメモリの設定

対象の外部フラッシュメモリに関連するパラメータは QUADSPI\_DCR レジスタによって設定されます。ユーザは FSIZE ビットでフラッシュメモリのサイズを、CSHT ビットでチップセレクトハイ時間の最小値を、MODE ビットで機能モード（モード 0 またはモード 3）を設定します。

### 14.3.10 QUADSPI の使用

動作モードは FMODE[1:0] (QUADSPI\_CCR[27:26]) を使用して選択されます。

#### インダイレクトモードの手順

FMODE が 00 に設定されている場合、インダイレクト書き込みモードが選択され、データをフラッシュメモリに送信することができます。FMODE が 01 に設定されている場合、インダイレクト読み出しモードが選択され、データをフラッシュメモリから読み出すことができます。

QUADSPI がインダイレクトモードで使用される場合、フレームは次のように構成されます。

1. QUADSPI\_DLR に、読み出される／書き込まれるデータのバイト数を指定します。
2. QUADSPI\_CCR に、フレームフォーマット、モード、および命令コードを指定します。
3. QUADSPI\_ABR に、アドレスフェーズの直後に送信されるオプションのオルタネートバイトを指定します。
4. QUADSPI\_CR に、動作モードを指定します。FMODE = 00 (インダイレクト書き込みモード) かつ DMAEN = 1 の場合、QUADSPI\_AR は QUADSPI\_CR より前に指定されなければなりません。そうしないと、QUADSPI\_AR が更新される前に QUADSPI\_DR が DMA によって書き込まれる可能性があるからです (DMA コントローラがすでに有効な場合)。
5. QUADSPI\_AR に、ターゲットアドレスを指定します。
6. QUADSPI\_DR により FIFO に対してデータの読み出し／書き込みを行います。

制御レジスタ (QUADSPI\_CR) に書き込む場合は次の設定値を指定してください。

- イネーブルビット (EN) を“1”にセット
- RAM に対するデータ送受信の DMA イネーブルビット (DMAEN)
- タイムアウトカウンタイネーブルビット (TCEN)
- サンプリングシフト設定 (SSHIFT)
- FTF フラグをセットするタイミングを示す FIFO 閾値レベル (FTRHES)
- 割り込みイネーブル
- 自動ポーリングモードのパラメータ: 一致モードおよび STOP モード (FMODE=11 のとき有効)
- クロックプリスケアラ

通信設定レジスタ (QUADSPI\_CCR) に書き込む場合は次のパラメータを指定してください。

- INSTRUCTION ビットによる命令バイト
- IMODE ビット (1、2、または4 ライン) により命令を送信する方法
- ADMODE ビット (なし、または1、2、4 ライン) によりアドレスを送信する方法
- ADSIZE ビットによるアドレスサイズ (8、16、24、または32 ビット)
- ADMODE ビット (なし、または1、2、4 ライン) によりオルタネートバイトを送信する方法
- ABSIZE ビットによるオルタネートバイト数 (1、2、3、または4)
- DBMODE ビットによるダミーバイトの有無
- DCYC ビットによるダミーバイト数
- DMODE ビットによるデータの送受信方法 (なし、または1、2、4 ライン)

アドレスレジスタ (QUADSPI\_AR) とデータレジスタ (QUADSPI\_DR) のどちらも特定のコマンドに関して更新する必要がない場合、QUADSPI\_CCR が書き込まれるとすぐにコマンドシーケンスが開始されます。これは、ADMODE と DMODE がどちらも 00 である場合、またはインダイレクト読み出しモード (FMODE=01) で ADMODE が 00 の場合に該当します。

アドレスを必要とする場合 (ADMODE が 00 以外) で、データレジスタに書き込む必要がない場合 (FMODE = 01 または DMODE = 00)、QUADSPI\_AR が書き込まれてアドレスが更新されるとすぐにコマンドシーケンスが開始されます。

データ転送時 (FMODE = 00 かつ DMODE! = 00)、QUADSPI\_DR による FIFO への書き込みによって通信の開始がトリガされます。

## ステータスフラグポーリングモード

ステータスフラグポーリングモードは、FMODE フィールド (QUADSPI\_CCR[27:26]) を 10 にセットすることで有効になります。このモードでは、プログラムされたフレームが送信され、定期的にデータが取得されます。

各フレームに読み込まれるデータの最大量は 4 バイトです。QUADSPI\_DLR で もっと大量のデータが要求されても、そのリクエストは無視され、4 バイトだけが読み出されます。

QUADSPI\_PISR レジスタには周期性が指定されます。

ステータスデータが取得されると、そのデータを次の目的のために内部で処理することができます。

- ステータス一致フラグをセットし、有効になっていれば割り込みを生成する。
- ステータスバイトの定期的な取得を自動的に停止する。

受信した値は QUADSPI\_PSMKR に格納されている値でマスクし、QUADSPI\_PSMAR に格納されている値で論理和または論理積をとることができます。

一致する場合、ステータス一致フラグがセットされ、有効になっていれば割り込みが生成されます。また、AMPS ビットがセットされている場合、QUADSPI は自動的に停止する可能性があります。

いずれにしろ、一番最後に取得した値は QUADSPI\_DR で使用可能です。

## メモリマップドモード

メモリマップドモードでは、外部フラッシュメモリは内部メモリとみなされますが、アクセス中にいくらかの遅延があります。このモードでは外部フラッシュメモリに対して読み出し動作のみが許可されます。

メモリマップドモードに移行するには、QUADSPI\_CCR レジスタの FMODE を 11 にセットします。

プログラムされた命令およびフレームは、AHB マスタがメモリマップド空間にアクセスしているときに送信されます。

FIFO は、リニア読み出しを予測するプリフェッチバッファとして使用されます。このモードでの QUADSPI\_DR へのアクセスはすべてゼロで返されます。

メモリマップドモードでは、データ長レジスタ (QUADSPI\_DLR) は意味を持ちません。

### 14.3.11 1 回限りの命令の送信

一部のフラッシュメモリ (Winbound など) が提供するモードは、最初のコマンドシーケンスでのみ命令を送信し、その後のコマンドは直接アドレスから開始されるというものです。SIOO ビット (QUADSPI\_CCR[28]) を使用して、そのような機能を活用することができます。

SIOO はすべての機能モードで有効です (インダイレクト、自動ポーリング、メモリマップド)。SIOO ビットがセットされている場合、QUADSPI\_CCR への書き込み後の最初のコマンドに対してのみ命令が送信されます。その後のコマンドシーケンスは、命令フェーズでは、QUADSPI\_CCR への書き込みがあるまでスキップされます。

IMODE = 00 (命令なし) の場合、SIOO は効果がありません。

### 14.3.12 QUADSPI エラー管理

エラーは次の場合に生成できます。

- インダイレクトモードまたはステータスフラグポーリングモードでは、QUADSPI\_AR に間違っただレスがプログラムされている場合 (QUADSPI\_DCR の FSIZE[4:0] で定義されているフラッシュメモリのサイズによる)、TEF がセットされ、有効になっていれば割り込みが生成されます。
- また、インダイレクトモードでは、アドレスおよびデータ長がフラッシュメモリのサイズを超えている場合は、アクセスがトリガされるとすぐに TEF がセットされます。
- メモリマップドモードでは、AHB マスタによる範囲外のアクセスが行われた場合または QUADSPI が無効な場合は、誤った AHB リクエストへの応答として AHB エラーが生成されます。
- メモリマップドモードが無効なときに AHB マスタがメモリマップド空間にアクセスすると、誤った AHB リクエストへの応答として AHB エラーが生成されます。

### 14.3.13 QUADSPI の BUSY ビットおよびアボート機能

QUADSPI がフラッシュメモリでの動作を開始すると、BUSY ビットが QUADSPI\_SR に自動的にセットされます。

インダイレクトモードでは、QUADSPI が要求されたコマンドシーケンスを完了し、FIFO が空になると、BUSY ビットがリセットされます。

自動ポーリングモードでは、一致 (APMS = 1 のとき) またはアボートの検出により、最後の定期的なアクセスが完了して初めて BUSY ビットはローになります。

メモリマップドモードでの最初のアクセス後、タイムアウトイベントまたはアボートの検出により、BUSY ビットはローになります。

QUADSPI\_CR レジスタの ABORT ビットをセットすることによって、動作を中止することができます。アボートが完了すると、BUSY ビットおよび ABORT ビットは自動的にリセットされ、FIFO が一掃されます。

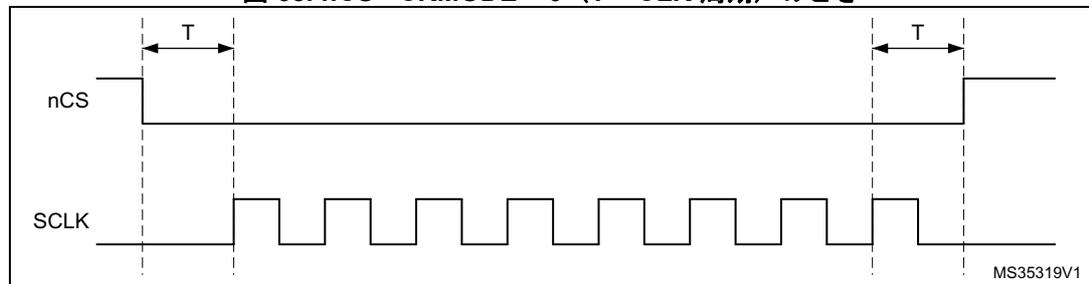
**注：** フラッシュメモリには、ステータスレジスタへの書き込み動作が中止された場合に誤動作するものがあります。

### 14.3.14 nCS の動作

デフォルトでは nCS はハイで、外部フラッシュメモリは選択解除されています。nCS の立ち下がり動作が始まる前に、立ち上がりは動作終了の直後に起こります。

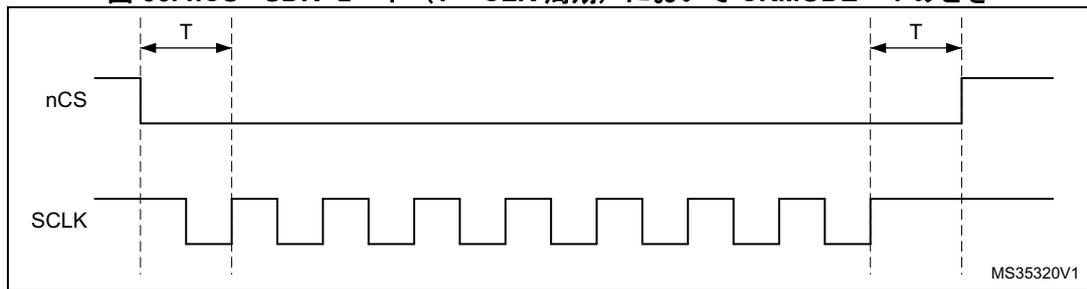
CKMODE = 0 (「モード0」で、この状態では進行中の動作がない場合は CLK がローに保持されている) のとき、nCS の立ち下がり動作の最初の CLK エッジ立ち上がりの 1 CLK サイクル前に、nCS の立ち上がりは動作の最後の CLK エッジ立ち上がりから 1 CLK サイクル後に起こります(図 65を参照)。

図 65. nCS - CKMODE = 0 (T = CLK 周期) のとき



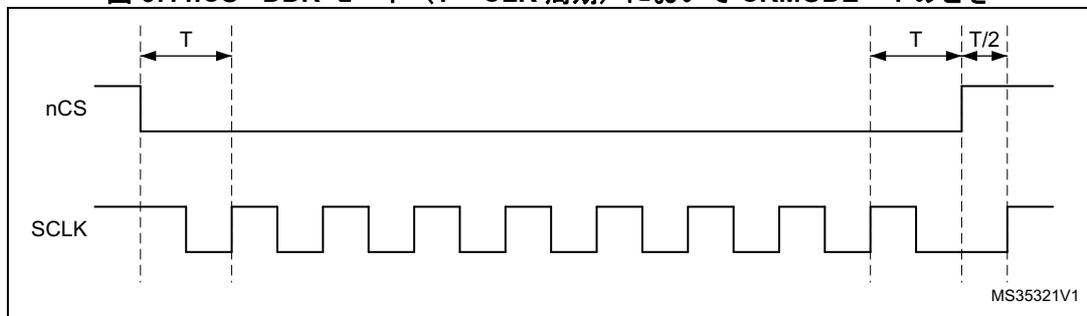
CKMODE = 1 (「モード3」で、この状態では進行中の動作がない場合に CLK がハイになる)、かつ DDRM = 0 (SDR モード) のとき、nCS は動作の最初の CLK エッジ立ち上がりの 1 CLK サイクル前でさらに立ち下がり、動作の最後の CLK エッジ立ち上がりから 1 CLK サイクル後に立ち上がります(図 66を参照)。

図 66. nCS - SDR モード (T = CLK 周期) において CKMODE = 1 のとき



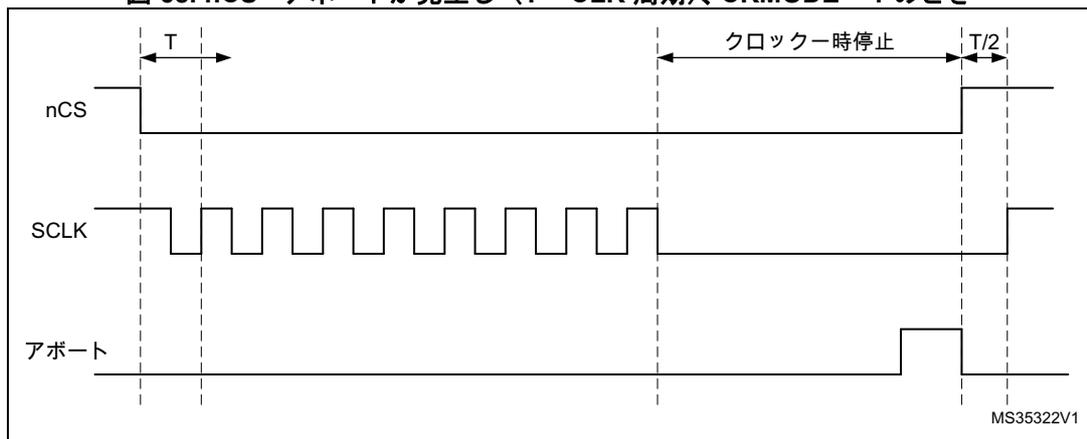
CKMODE = 1 (モード3) かつ DDRM = 1 (DDR モード) の場合、nCS の立ち下がり動作の最初の CLK エッジ立ち上がりの 1 CLK サイクル前に、nCS の立ち上がり動作の最後の アクティブ CLK エッジ立ち上がりから 1 CLK サイクル後に起こります (図 67を参照)。DDR 動作は立ち下がりエッジで終わらなければならないため、nCS が立ち上がる時 CLK はローとなり、その後 CLK は 1/2 CLK サイクルだけ後ろ向きに立ちあがります。

図 67. nCS - DDR モード (T = CLK 周期) において CKMODE = 1 のとき



読み出し動作で FIFO が満杯のまま、または書き込み動作で FIFO が空のままであれば、ファームウェアが FIFO を処理するまでその動作は停止し、CLK はローのままとなります。動作が停止した時点でアボートが発生した場合、アボートが要求された直後に nCS の立ち上がりが起こり、CLK は 1/2 CLK サイクル後に立ち上がります (図 68を参照)。

図 68. nCS - アボートが発生し (T = CLK 周期)、CKMODE = 1 のとき



デュアルフラッシュモード以外のおとき (DFM = 0) はフラッシュ 1 のみがおアクセスされるので、BK2\_nCS はハイのままです。デュアルフラッシュモードでは、BK2\_nCS は BK1\_nCS とまったく同じように動作します。したがって、フラッシュ 2 が存在し、アプリケーションがずっとデュアルフラッシュモードのままであれば、フラッシュ 2 は BK1\_nCS を使用することができ、BK2\_nCS を出力するピンを他の機能に使用することができます。

## 14.4 QUADSPI 割り込み

割り込みは次のイベントによって生成できます。

- タイムアウト
- ステータス一致
- FIFO 閾値
- 転送完了
- 転送エラー

高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 88. QUADSPI 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
タイムアウト	TOF	TOIE
ステータス一致	SMF	SMIE
FIFO 閾値	FTF	FTIE
転送完了	TCF	TCIE
転送エラー	TEF	TEIE

## 14.5 QUADSPI レジスタ

### 14.5.1 QUADSPI 制御レジスタ (QUADSPI\_CR)

アドレスオフセット : 0x0000

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTHRES					FSEL	DFM	Res.	SSHIFT	TCEN	DMAEN	ABORT	EN
			r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	w1s	r/w	w1s

#### ビット 31 : 24 PRESCALER[7:0] : クロックプリスケアラ

このフィールドは、AHB クロック (値 + 1)に基づいて CLK を生成するスケアラ係数を定義します。

0 :  $F_{CLK} = F_{AHB}$ 、QUADSPI CLK (迂回されたプリスケアラ)として直接的に使用される AHB クロック

1 :  $F_{CLK} = F_{AHB}/2$

2 :  $F_{CLK} = F_{AHB}/3$

...

255 :  $F_{CLK} = F_{AHB}/256$

クロック分周比が奇数の場合、CLK のデューティサイクルは 50% ではありません。クロック信号は、ローに保持された時間より 1 サイクル分長く、ハイに保持されます。

このフィールドは、BUSY が 0 のときにのみ修正できます。

#### ビット 23 PMM : ポーリング一致モード

このビットは、自動ポーリングモード時に「一致」を決定するために使うべき方法を示します。  
0 : AND 一致モード SMF は、フラッシュメモリから受信したマスクされていないすべてのビットが一致レジスタの対応するビットと一致している場合に、セットされます。

1 : OR 一致モード SMF は、フラッシュメモリから受信したマスクされていないビットのいずれかが一致レジスタの対応するビットと一致している場合に、セットされます。

このビットは、BUSY が 0 のときにのみ修正できます。

#### ビット 22 APMS : 自動ポーリングモード停止

このビットは、一致後に自動ポーリングを停止するかどうかを決定します。

0 : 自動ポーリングモードは、アボートまたは QUADSPI の無効化によってのみ停止されます。

1 : 自動ポーリングモードは、一致が検出されるとすぐに停止されます。

このビットは、BUSY が 0 のときにのみ修正できます。

#### ビット 21 予約済みであり、リセット値に保持する必要があります。

#### ビット 20 TOIE : タイムアウト割り込み有効化

このビットは、タイムアウト割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

#### ビット 19 SMIE : ステータス一致割り込み有効化

このビットは、ステータス一致割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

**ビット 18 FTIE** : FIFO 閾値割り込み有効化

このビットは、FIFO 閾値割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

**ビット 17 TCIE** : 転送完了割り込み有効化

このビットは、転送完了割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

**ビット 16 TEIE** : 転送エラー割り込み有効化

このビットは、転送エラー割り込みを有効にします。

0 : 割り込みは無効です。

1 : 割り込みは有効です。

ビット 15 : 13 予約済みであり、リセット値に保持する必要があります。

**ビット 12:8 FTHRES[4:0]** FIFO 閾値レベル

インダイレクトモードでは、FIFO 閾値フラグ (FTF、QUADSPI\_SR[2]) がセットされる FIFO 内のバイトの閾値数を定義します。

インダイレクト書き込みモード (FMODE = 00)の場合

0:FTF は、FIFO 内に書き込みのできる空きバイトが 1 バイト以上ある場合にセットされます。

1:FTF は、FIFO 内に書き込みのできる空きバイトが 2 バイト以上ある場合にセットされます。

...

31 : FTF は、FIFO 内に書き込みのできる空きバイトが 32 バイト以上ある場合にセットされます。

インダイレクト読み出しモード (FMODE = 01)の場合

0 : FTF は、FIFO から読み出せる有効なバイトが 1 バイト以上ある場合にセットされます。

1 : FTF は、FIFO から読み出せる有効なバイトが 2 バイト以上ある場合にセットされます。

...

31 : FTF は、FIFO から読み出せる有効なバイトが 32 バイト以上ある場合にセットされます。  
DMAEN = 1 の場合、対応するチャネル用の DMA コントローラは FTHRES 値を変更する前に無効にする必要があります。

**ビット 7 FSEL** : フラッシュメモリの選択

このビットは、シングルフラッシュモード (DFM = 0) でアドレス指定するフラッシュメモリを選択します。

0 : フラッシュ 1 が選択されます。

1 : フラッシュ 2 が選択されます。

このビットは、BUSY が 0 のときにのみ修正できます。

このビットは、DFM = 1 のときは無視されます。

**ビット 6 DFM** : デュアルフラッシュモード

このビットはデュアルフラッシュモードをアクティブにします。このモードでは、2 つの外部フラッシュメモリを同時に使用して、スループットおよび容量を倍にします。

0 : デュアルフラッシュモードは無効です。

1 : デュアルフラッシュモードは有効です。

このビットは、BUSY が 0 のときにのみ修正できます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

### ビット 4 SSHIFT : サンプリングシフト

デフォルトでは、QUADSPI はフラッシュメモリによるデータ駆動の 1/2 CLK サイクル後にデータをサンプリングします。このビットを使用することで、外部信号遅延を考慮して、後でデータをサンプリングすることができます。

0 : シフトなし

1 : 1/2 サイクルのシフト

ファームウェアは、DDR モード (DDRM = 1) では SSHIFT = 0 であることを予測する必要があります。

このフィールドは、BUSY が 0 のときにのみ修正できます。

### ビット 3 TCEN : タイムアウトカウンタ有効化

このビットは、メモリマップドモードが選択されている (FMODE = 11) 場合にのみ有効です。ある一定の時間アクセスがない場合 (この時間は TIMEOUT[15:0] (QUADSPI\_LPTR) によって定義されます)、このビットをアクティブにすることでチップセレクト (nCS) がリリースされず (これにより消費電力が削減されます)。

タイムアウトカウンタを有効にします。

デフォルトでは、QUADSPI がプリフェッチ動作を停止することはなく、フラッシュメモリへのアクセスが長時間なくても、前の読み出し動作はアクティブな状態を保ち、nCs もローに保持されます。nCS がローに保持されるとフラッシュメモリの消費電力が増える傾向があるため、アプリケーションではタイムアウトカウンタをアクティブにしようとするかもしれません (TCEN = 1, QUADSPI\_CR[3])。これは、FIFO がプリフェッチしたデータで満杯になって以来アクセスが 1 回もないまま TIMEOUT[15:0] (QUADSPI\_LPTR) サイクルが経過した後に、nCS がリリースされるようにするためです。

0 : タイムアウトカウンタは無効です。したがって、チップセレクト (nCS) はメモリマップドモードにおける 1 回のアクセス後は無限にアクティブなままです。

1 : タイムアウトカウンタは有効です。したがって、チップセレクトは、インアクティブなフラッシュメモリの TIMEOUT[15:0] サイクル後にメモリマップドモードでリリースされます。このビットは、BUSY が 0 のときにのみ修正できます。

### ビット 2 DMAEN : DMA 有効化

インダイレクトモードでは、DMA を使用して、QUADSPI\_DR レジスタ経由でデータを入出力することができます。DMA 転送は、FTF (FIFO 閾値フラグ) がセットされた時点で開始されます。

0 : インダイレクトモードでは DMA は無効です。

1 : インダイレクトモードでは DMA は有効です。

### ビット 1 ABORT : アボートリクエスト

このビットは進行中のコマンドシーケンスを停止します。すべてのアボートが完了すると自動的にリセットされます。

このビットは現在の転送を中止させます。

ポーリングモードまたはメモリマップドモードでは、このビットは APM ビットまたは DM ビットもリセットします。

0 : アボートは要求されていません。

1 : アボートが要求されています。

### ビット 0 EN : 有効化

QUADSPI. を有効にします。

0 : QUADSPI. は無効です。

1 : QUADSPI. は有効です。

## 14.5.2 QUADSPI デバイス設定レジスタ (QUADSPI\_DCR)

アドレスオフセット : 0x0004

リセット値 : 0x0000 0000

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

ビット 31 : 21 予約済みであり、リセット値に保持する必要があります。

ビット 20 : 16 **FSIZE[4:0]** : フラッシュメモリサイズ

このフィールドは、次の式を使用して外部メモリのサイズを定義します。

$$\text{フラッシュメモリ内のバイト数} = 2^{[FSIZE+1]}$$

FSIZE+1 はフラッシュメモリのアドレスを指定するために必要とされる有効なアドレスビット数です。フラッシュメモリの容量はインダイレクトモードでは最大 4 GB (32 ビットを使用してアドレス指定) ですが、メモリマップドモードにおけるアドレス指定可能な空間は 256 MB に制限されます。

DFM = 1 の場合、FSIZE は 2 つのフラッシュメモリの総容量を示します。

このフィールドは、BUSY が 0 のときのみ修正できます。

ビット 15 : 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **CSHT[2:0]** : チップセレクトハイ時間

CSHT+1 は、フラッシュメモリに対して発行されるコマンド間でチップセレクト (nCS) がハイに保たれなければならない CLK サイクルの最小数を定義します。

0 : nCS は、フラッシュメモリコマンド間の少なくとも 1 サイクルの間はハイに保持されます。

1 : nCS は、フラッシュメモリコマンド間の少なくとも 2 サイクルの間はハイに保持されます。

...

7 : nCS は、フラッシュメモリコマンド間の少なくとも 8 サイクルの間はハイに保持されます。

このフィールドは、BUSY が 0 のときのみ修正できます。

ビット 7 : 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CKMODE** : モード 0 / モード 3

このビットは、コマンド間での CLK のレベルを示します (nCS = 1 のとき)。

0 : CLK は nCS がハイである間はローに保持する必要があります (チップセレクトがリリースされる)。これは、モード 0 と呼ばれます。

1 : CLK は nCS がハイである間はハイに保持する必要があります (チップセレクトがリリースされる)。これは、モード 3 と呼ばれます。

このフィールドは、BUSY が 0 のときのみ修正できます。

## 14.5.3 QUADSPI ステータスレジスタ (QUADSPI\_SR)

アドレスオフセット : 0x0008

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
		r	r	r	r	r	r			r	r	r	r	r	r

ビット 31 : 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 : 8 **FLEVEL[5:0]** : FIFO レベル

このフィールドは FIFO 内に保持されている有効バイト数を示します。FIFO が空のときは FLEVEL = 0、FIFO が満杯のときは FLEVEL = 32 です。メモリマップドモードおよび自動ステータスポーリングモードでは、FLEVEL はゼロです。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **BUSY** : ビジー

このビットは動作が進行中のときにセットされます。このビットは、フラッシュメモリの動作が終わり、FIFO が空になると自動的にクリアされます。

ビット 4 **TOF** : タイムアウトフラグ

このビットは、タイムアウトが発生するとセットされます。CTOF に 1 を書き込むことによってクリアされます。

ビット 3 **SMF** : ステータス一致フラグ

このビットは、マスクされていない受信データが一致レジスタ (QUADSPI\_PSMAR) の対応するビットと一致している場合に、自動ポーリングモードでセットされます。CSMF に 1 を書き込むことによってクリアされます。

ビット 2 **FTF** : FIFO 閾値フラグ

インダイレクトモードでは、このビットは FIFO 閾値に達した時点で、またはフラッシュメモリからの読み出し完了後に FIFO 内にデータが残っている場合にセットされます。閾値条件が真でなくなるとすぐに自動的にクリアされます。

自動ポーリングモードでは、このビットはステータスレジスタが読み出されるたびにセットされ、データレジスタが読み出された時点でクリアされます。

ビット 1 **TCF** : 転送完了フラグ

このビットは、プログラムされたデータ数が転送された時点でインダイレクトモードでセットされるか、または転送が中止された時点でいずれかのモードでセットされます。また、CTCF に 1 を書き込むことによってクリアされます。

ビット 0 **TEF** : 転送エラーフラグ

このビットは、インダイレクトモードで無効なアドレスにアクセスされているときに、インダイレクトモードでセットされます。CTEF に 1 を書き込むことによってクリアされます。

## 14.5.4 QUADSPI フラグクリアレジスタ (QUADSPI\_FCR)

アドレスオフセット : 0x000C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CTOF	CSMF	Res.	CTCF	CTEF										
											w1o	w1o		w1o	w1o

ビット 31 : 4 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CTOF** : タイムアウトフラグのクリア

このビットに 1 を書き込むと、QUADSPI\_SR レジスタの TOF フラグがクリアされます。

ビット 3 **CSMF** : ステータス一致フラグのクリア

このビットに 1 を書き込むと、QUADSPI\_SR レジスタの SMF フラグがクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CTCF** : 転送完了フラグのクリア

このビットに 1 を書き込むと、QUADSPI\_SR レジスタの TCF フラグがクリアされます。

ビット 0 **CTEF** : 転送エラーフラグのクリア

このビットに 1 を書き込むと、QUADSPI\_SR レジスタの TEF フラグがクリアされます。

## 14.5.5 QUADSPI データ長レジスタ (QUADSPI\_DLR)

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

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

ビット 31:0 **DL[31:0]** : データ長

インダイレクトモードおよびステータスポーリングモードで取得されるデータ数 (値 + 1) ステータスポーリングモードの場合、3 より大きい値 (4 バイトを示す) を使用する必要があります。インダイレクトモードのすべて1は、不定長を意味します。この場合、FSIZE によって定義されているメモリの終了まで QUADSPI が継続します。

0x0000\_0000 : 1 バイトが転送されます。

0x0000\_0001 : 2 バイトが転送されます。

0x0000\_0002 : 3 バイトが転送されます。

0x0000\_0003 : 4 バイトが転送されます。

...

0xFFFF\_FFFD : 4,294,967,294 (4G-2) バイトが転送されます。

0xFFFF\_FFFE : 4,294,967,295 (4G-1) バイトが転送されます。

0xFFFF\_FFFF : 不定長 -- すべてのバイトがフラッシュメモリの終了 (FSIZE によって定義) まで転送されます。FSIZE = 0x1F の場合、読み出しが無限に続けられます。

デュアルフラッシュモード (DFM=1) では、DL[0] は、たとえこのビットに 0 が書き込まれても 1 のままとなります。したがって、アクセスのたびに必ず偶数バイトが転送されます。

このフィールドはメモリマップドモード (FMODE = 10) では効果がありません。

このフィールドは、BUSY が 0 のときのみ書き込みができます。

## 14.5.6 QUADSPI 通信設定レジスタ (QUADSPI\_CCR)

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DDRM	DHHC	Res.	SIOO	FMODE[1:0]		DMODE		Res.	DCYC[4:0]				ABSIZE		
r/w	r/w		r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE		ADSIZE		ADMODE		IMODE		INSTRUCTION[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **DDRM** : データレートモードを倍にします。

このビットは、アドレス、オルタネートバイト、およびデータフェーズの DDR モードをセットします。

0 : DDR モードは無効です。

1 : DDR モードは有効です。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 30 **DHHC** : DDR ホールド

DDR モードで、データの出力を 1/4 QUADSPI 出カクロックサイクルだけ遅延させます。

0 : アナログ遅延を使用してデータ出力を遅延させます。

1 : データの出力を 1/4 QUADSPI 出カクロックサイクルだけ遅延させます。

この機能は、DDR モードでのみアクティブです。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **SIOO** : モードにつき 1 回のみ命令を送信します。

[セクション 14.3.11 : 1 回限りの命令の送信 \(392 ページ\)](#) を参照してください。このビットは、IMODE = 00 のときには効果がありません。

0 : トランザクションごとに命令を送信します。

1 : 最初のコマンドに対してのみ命令を送信します。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 27:26 **FMODE[1:0]** : 機能モード

このフィールドは、動作の QUADSPI 機能モードを定義します。

00 : インダイレクト書き込みモード

01 : インダイレクト読み出しモード

10 : 自動ポーリングモード

11 : メモリマップドモード

DMAEN = 1 の場合、対応するチャネル用の DMA コントローラは FMODE 値を変更する前に無効にする必要があります。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 25:24 **DMODE[1:0]** : データモード

このフィールドは、動作のデータフェーズのモードを次のように定義します。

00 : データはありません。

01 : データは 1 本のライン上にあります。

10 : データは 2 本のライン上にあります。

11 : データは 4 本のライン上にあります。

また、このフィールドは動作のダミーフェーズモードも決定します。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **DCYC[4:0]** : ダミーサイクル数

このフィールドはダミーフェーズの処理時間を定義します。SDR モードと DDR モードの両方で、CLK サイクルの数を指定します (0~31)。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 17:16 **ABSIZE[1:0]** : オルタネートバイトサイズ

このビットはオルタネートバイトのサイズを定義します。

00 : 8 ビットオルタネートバイト

01 : 16 ビットオルタネートバイト

10 : 24 ビットオルタネートバイト

11 : 32 ビットオルタネートバイト

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 15:14 **ABMODE[1:0]** : オルタネートバイトモード

このフィールドは、動作のオルタネートバイトフェーズのモードを次のように定義します。

00 : オルタネートバイトはありません。

01 : オルタネートバイトは 1 本のライン上にあります。

10 : オルタネートバイトは 2 本のライン上にあります。

11 : オルタネートバイトは 4 本のライン上にあります。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 13:12 **ADSIZE[1:0]** : アドレスサイズ

このビットはアドレスサイズを定義します。

00 : 8 ビットアドレス

01 : 16 ビットアドレス

10 : 24 ビットアドレス

11 : 32 ビットアドレス

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 11:10 **ADMODE[1:0]** : アドレスモード

このフィールドは、動作のアドレスフェーズのモードを次のように定義します。

00 : アドレスはありません。

01 : アドレスは 1 本のライン上にあります。

10 : アドレスは 2 本のライン上にあります。

11 : アドレスは 4 本のライン上にあります。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 9:8 **IMODE[1:0]** : 命令モード

このフィールドは、動作の命令フェーズのモードを次のように定義します。

00 : 命令はありません。

01 : 命令は 1 本のライン上にあります。

10 : 命令は 2 本のライン上にあります。

11 : 命令は 4 本のライン上にあります。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

ビット 7 : 0 **INSTRUCTION[7:0]** : 命令

外部 SPI デバイスに送信される命令

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

## 14.5.7 QUADSPI アドレスレジスタ (QUADSPI\_AR)

アドレスオフセット : 0x0018

リセット値 : 0x0000 0000

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

ビット 31:0 **ADDRESS[31:0]** : アドレス

外部フラッシュメモリに送信されるアドレス

このフィールドへの書き込みは、BUSY = 0 または FMODE = 11 (メモリマップドモード) の場合は無視されます。

デュアルフラッシュモードでは、アドレスは常に偶数でなければならないので、ADDRESS[0] は自動的に 0 に固定されます。

## 14.5.8 QUADSPI オルタネートバイトレジスタ (QUADSPI\_ABR)

アドレスオフセット : 0x001C

リセット値 : 0x0000 0000

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

ビット 31 : 0 **ALTERNATE [31:0]** : オルタネートバイト

アドレスの直後に外部 SPI デバイスに送信されるオプションのデータ

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

## 14.5.9 QUADSPI データレジスタ (QUADSPI\_DR)

アドレスオフセット : 0x0020

リセット値 : 0x0000 0000

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

### ビット 31 : 0 DATA [ 31:0 ] : データ

外部 SPI デバイスに対して送受信されるデータ

インダイレクト書き込みモードでは、このレジスタに書き込まれたデータはいったん FIFO に格納されてから、データフェーズ時にフラッシュメモリに送信されます。FIFO が満杯の場合、書き込まれるデータ量を受け入れるだけの十分な空間が FIFO にできるまで、書き込み動作は停止されます。

インダイレクト読み出しモードでは、このレジスタを読み出すことで (FIFO 経由)、フラッシュメモリから受信したデータが提供されます。FIFO のバイト数が読み出し動作で要求したバイト数ほど多くない場合で、BUSY = 1 のとき、データ量が十分になるか、転送が完了するかの内いずれか早く起こるほうまで、読み出し動作は停止されます。

自動ポーリングモードでは、このレジスタはフラッシュメモリから読み出された最後のデータを格納します (マスキングなし)。

このレジスタへのワード、ハーフワード、バイトアクセスはサポートされます。インダイレクト書き込みモードでの書き込みにより FIFO に追加されるバイト数は、バイト書き込みで 1 バイト、ハーフワード書き込みで 2 バイト、ワード書き込みで 4 バイトです。同様に、インダイレクト読み出しモードでの読み出しにより FIFO から削除されるバイト数は、バイト読み出しで 1 バイト、ハーフワード読み出しで 2 バイト、ワード読み出しで 4 バイトです。インダイレクトモードでのアクセスは、このレジスタのボトムに整列されなければなりません。つまり、バイト読み出しでは DATA[7:0] が、ハーフワード読み出しでは DATA[15:0] が読み出されます。

## 14.5.10 QUADSPI ポーリングステータスマスクレジスタ (QUADSPI\_PSMKR)

アドレスオフセット : 0x0024

リセット値 : 0x0000 0000

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

ビット 31 : 0 **MASK[31:0]** : ステータスマスク

ポーリングモードで受信したステータスバイトに適用されるマスク

ビット n の場合

0 : 自動ポーリングモードで受信したデータのビット n はマスクされ、その値は一致ロジックでは考慮されません。

1 : 自動ポーリングモードで受信したデータのビット n はマスクされず、その値は一致ロジックで考慮されます。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

## 14.5.11 QUADSPI ポーリングステータス一致レジスタ (QUADSPI\_PSMAR)

アドレスオフセット : 0x0028

リセット値 : 0x0000 0000

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

ビット 31 : 0 **MATCH[ 31:0]** : ステータス一致

一致を得るためにマスクされたステータスレジスタと比較される値

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

## 14.5.12 QUADSPI ポーリングインターバルレジスタ (QUADSPI\_PIR)

アドレスオフセット : 0x002C

リセット値 : 0x0000 0000

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

ビット 31 : 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 : 0 **INTERVAL [15:0]** : ポーリングインターバル

自動ポーリングフェーズ時の読み出し間の CLK サイクル数

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

## 14.5.13 QUADSPI 低電力タイムアウトレジスタ (QUADSPI\_LPTR)

アドレスオフセット : 0x0030

リセット値 : 0x0000 0000

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

ビット 31 : 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 : 0 **TIMEOUT [15:0]** : タイムアウト期間

メモリマップドモードでのアクセス終了のたびに、QUADSPI はその後のバイトをプリフェッチし、それらを FIFO に保持します。このフィールドは、FIFO が満杯になってから nCS を立ち上げるまでに QUADSPI が待つ CLK サイクルの数を示し、フラッシュメモリを消費電力がより少ない状態にします。

このフィールドは、BUSY が 0 のときにのみ書き込みができます。

## 14.5.14 QUADSPI レジスタマップ

表 89. QUADSPI レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x0000	QUADSPI_CR	PRESCALER[7:0]										PMM	APMS	res.	TOIE	SMIE	FTIE	TCIE	TEIE	res.	res.	res.	FTHRES [4:0]				FSEL	DFM	res.	SSHIFT	TCEN	DMAEN	ABORT	EN	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0004	QUADSPI_DCR	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	FSIZE[4:0]				res.	res.	res.	res.	CSHT				res.	res.	res.	res.	res.	CKMODE						
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	QUADSPI_SR	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	
	リセット値																																		
0x000C	QUADSPI_FCR	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	
	リセット値																																		
0x0010	QUADSPI_DLR	DL[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	QUADSPI_CCR	DDRM	DHHC	res.	SIOO	FMODE[1:0]	DMODE[1:0]	res.	DCYC[4:0]				ABSIZ[1:0]	ABMODE[1:0]	ADSIZ[1:0]	ADM[1:0]	IMODE[1:0]	INSTRUCTION[7:0]																	
	リセット値	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	QUADSPI_AR	ADDRESS[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x001C	QUADSPI_ABR	ALTERNATE[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0020	QUADSPI_DR	DATA[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0024	QUADSPI_PSMKR	MASK[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0028	QUADSPI_PSMAR	MATCH[31:0]																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x002C	QUADSPI_PIR	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	
	リセット値																																		
0x0030	QUADSPI_LPTR	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	res.	
	リセット値																																		

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



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

## 15 アナログデジタルコンバータ (ADC)

### 15.1 ADC の概要

この 12 ビット ADC は、逐次比較型アナログデジタルコンバータです。最大 19 の多重化チャネルを持ち、16 の外部ソース、2 つの内部ソース、および  $V_{BAT}$  チャネルの測定を行うことができます。これらのチャネルの A/D 変換は、シングル、連続、スキャン、または不連続モードで行うことができます。ADC の結果は、左詰めまたは右詰めで 16 ビットのデータレジスタに格納されます。

アナログウォッチドッグ機能により、入力電圧が、ユーザ定義の上限値または下限値から逸脱していないかを、アプリケーションで検出することができます。

### 15.2 ADC の主な機能

- 12 ビット、10 ビット、8 ビット、または 6 ビットの設定可能な分解能
- 変換終了時、インジェクト変換終了時、および、アナログウォッチドッグイベントまたはオーバーランイベント時に割り込みを生成
- シングルおよび連続変換モード
- チャネル 0 からチャネル“n”を自動変換するためのスキャンモード
- 組み込みのデータコヒーレンシによるデータ配置
- チャネル単位でプログラム可能なサンプリング時間
- レギュラ変換とインジェクト変換の両方について極性設定可能な外部トリガオプション
- 不連続モード
- デュアル/トリプルモード (2 つ以上の ADC を持つデバイス)
- デュアル/トリプル ADC モードにおいて設定可能な DMA データ保存
- デュアル/トリプルインターリーブモード変換間の設定可能な遅延
- ADC 変換タイプ (データシートを参照)
- ADC 電源仕様: フルスピード時 2.4 V から 3.6 V、低速時 1.8 V から低下可能
- ADC 入力電圧範囲:  $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- レギュラチャネル変換中の DMA リクエストの生成

 69 に、ADC のブロック図を示します。

**注:**  $V_{REF-}$  ピンが用意されているパッケージでは、このピンを  $V_{SSA}$  に固定する必要があります。

## 15.3 ADC の機能詳細

図 69 に 1 つの ADC ブロック図を、表 90 に ADC ピンの説明を示します。

図 69. 1 つの ADC ブロック図

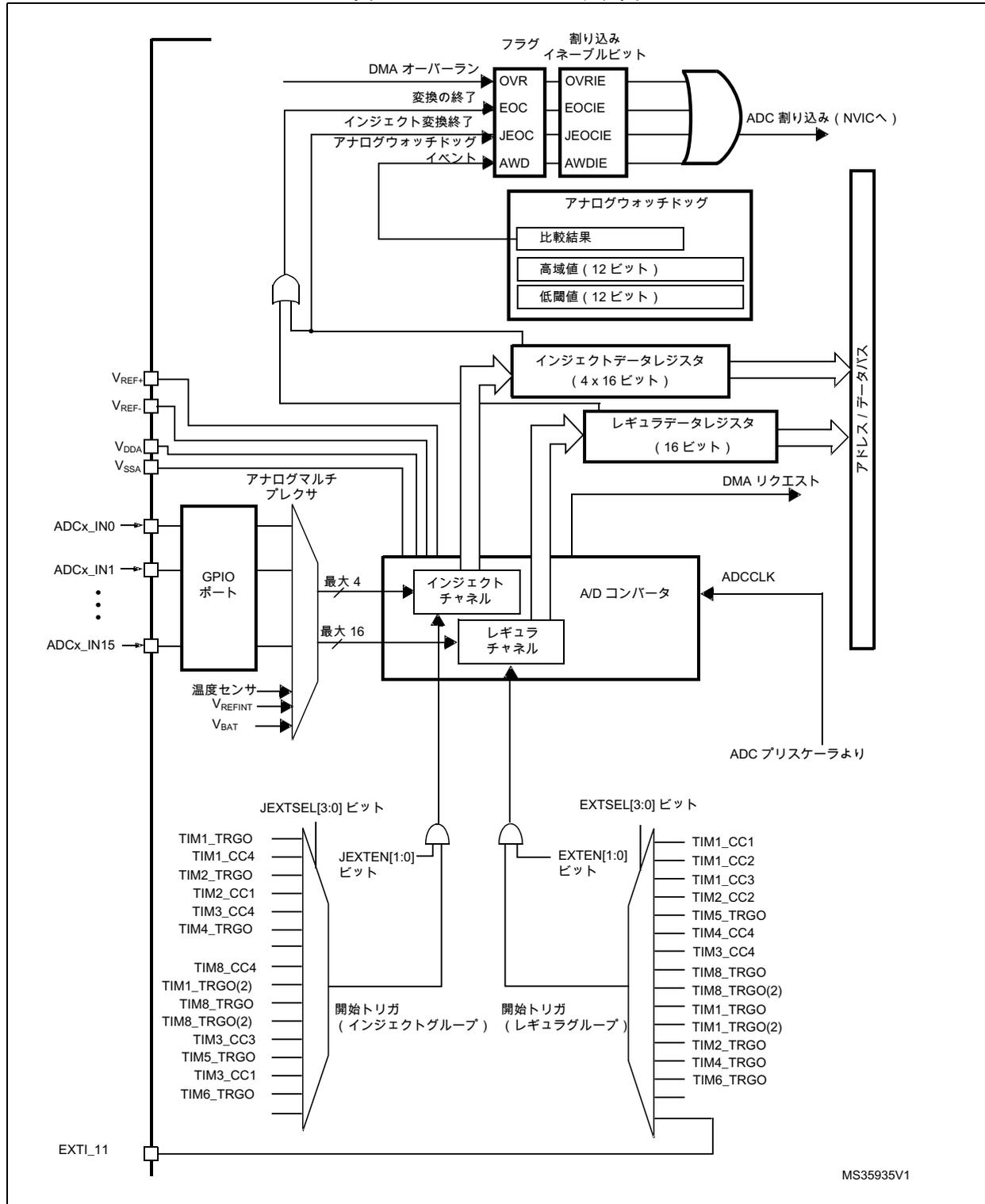


表 90. ADC ピン

名前	信号タイプ	説明
$V_{REF+}$	入力、アナログ基準電圧正	ADC のハイレベル／正基準電圧 $1.8\text{ V} \leq V_{REF+} \leq V_{DDA}$
$V_{DDA}$	入力、アナログ電源供給	アナログ電源供給、 $V_{DD}$ に等しく、 $2.4\text{ V} \leq V_{DDA} \leq V_{DD}$ (3.6 V) フルスピード用 $1.8\text{ V} \leq V_{DDA} \leq V_{DD}$ (3.6 V) 低速用
$V_{REF-}$	入力、アナログ基準電圧負	ADC のローレベル／負基準電圧 $V_{REF-} = V_{SSA}$
$V_{SSA}$	入力、アナログ供給グラウンド	$V_{SS}$ に等しいアナログ電源供給のグラウンド
ADCx_IN[15:0]	アナログ入力信号	16 チャンネルのアナログ入力

### 15.3.1 ADC のオン／オフ制御

ADC は、ADC\_CR2 レジスタの ADON ビットをセットすることで、電源オンの状態になります。ADON ビットが最初にセットされると、ADC はパワーダウンモードからウェイクアップします。

SWSTART ビットまたは JSWSTART ビットのいずれかがセットされると変換が開始します。

ADON ビットをクリアすることによって、変換を停止し、ADC をパワーダウンモードにできます。このモードでは、ADC は電力をほとんど消費しません (数  $\mu\text{A}$  程度)。

## 15.3.2 ADC1/2 および ADC3 接続性

ADC1、ADC2、および ADC3 は密接に結合されており、いくつかの外部チャンネルを共有しています。  
 図 70、図 71、および 図 72 を参照してください。

図 70. ADC1 接続性

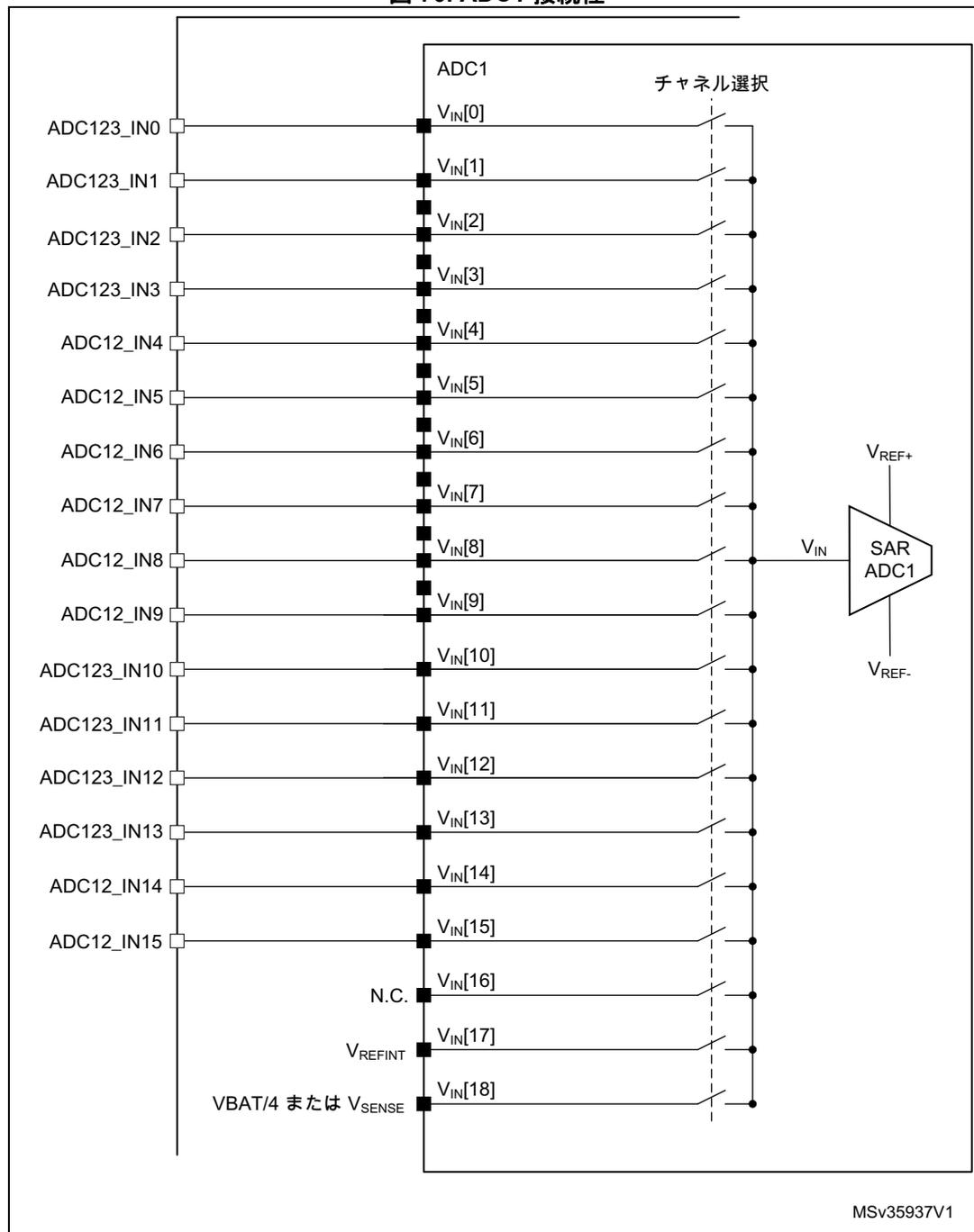


図 71. ADC2 接続性

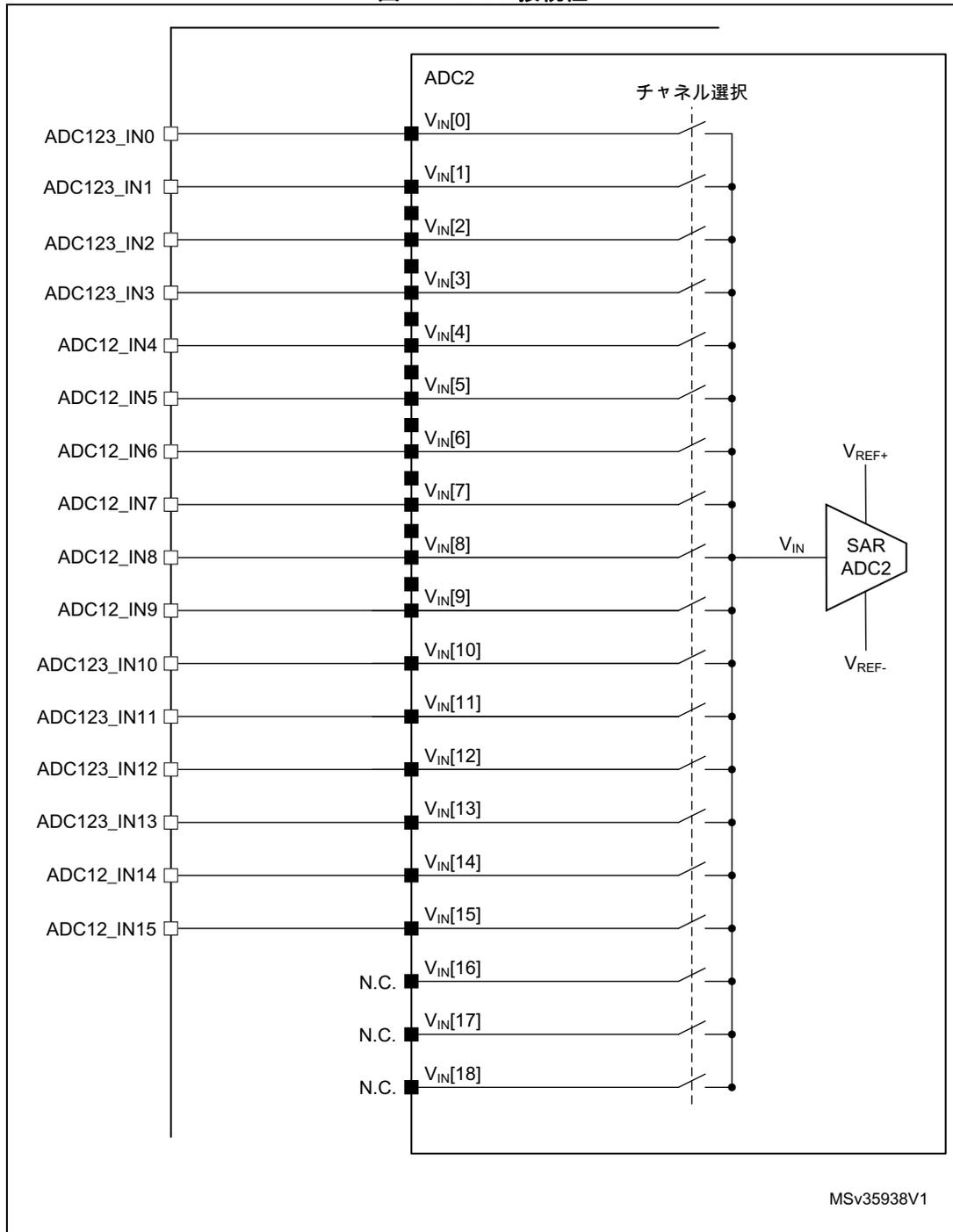
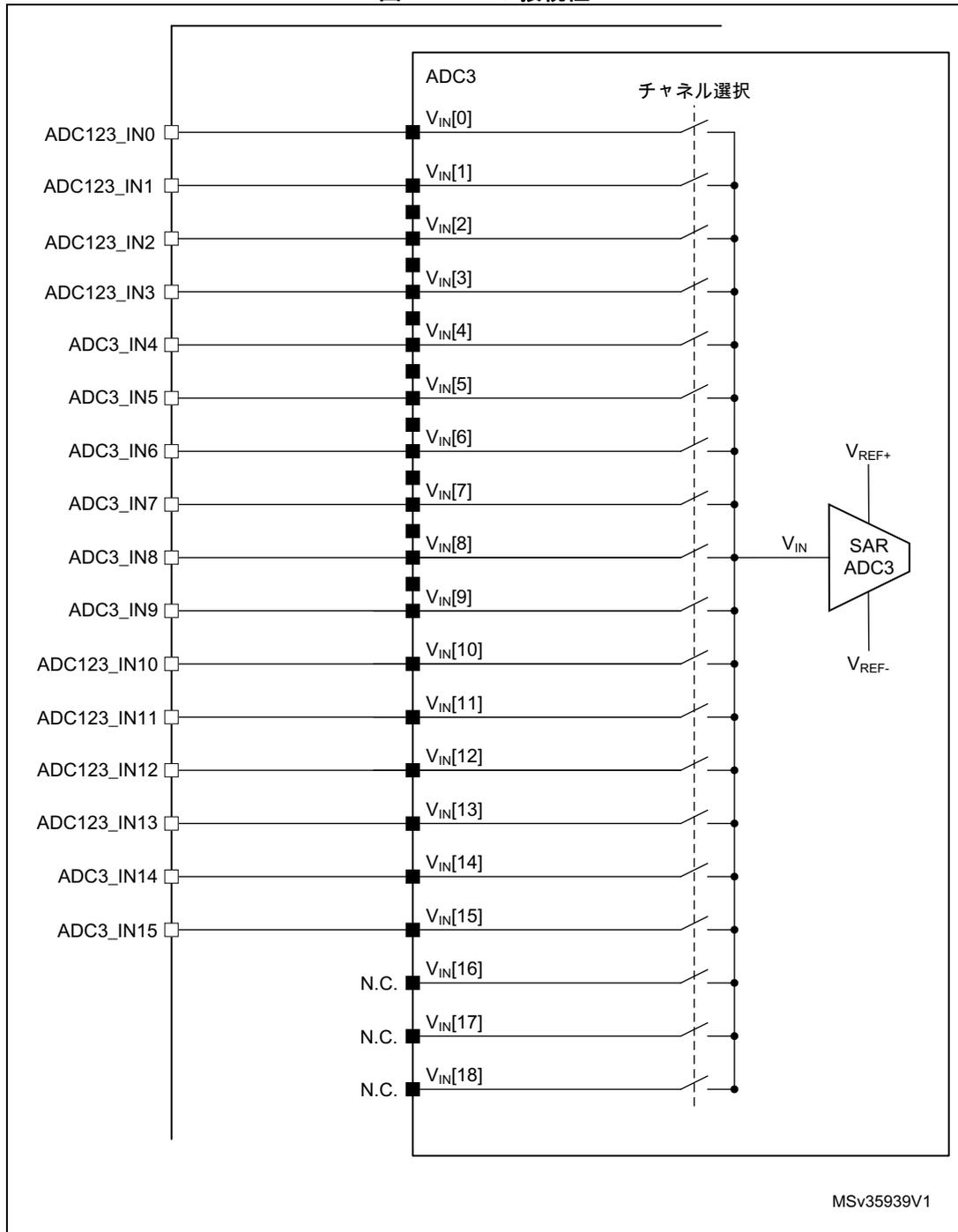


図 72. ADC3 接続性



## 15.3.3 ADC クロック

この ADC には 2 つのクロックスキームがあります。

- アナログ回路のクロック : ADCCLK (すべての ADC に共通)  
このクロックは、プログラム可能なプリスケラによって分周された APB2 クロックによって生成され、ADC は  $f_{PCLK2}/2$ 、 $/4$ 、 $/6$  または  $/8$  で動作することができます。ADCCLK の最大値に関してはデータシートを参照してください。
- デジタルインタフェース用のクロック (レジスタの読み書きアクセスに使用)  
このクロックは APB2 クロックと同一です。デジタルインタフェースクロックは、RCC APB2 ペリフェラルクロックイネーブルレジスタ (RCC\_APB2ENR) を使用して ADC ごとに有効/無効にすることができます。

## 15.3.4 チャネル選択

16 の多重化チャネルがあります。変換は、レギュラとインジェクトの 2 つのグループに構成することができます。各グループは、任意のチャネルに対して任意の順序で行うことができる一連の変換で構成されます。たとえば、次のような順で変換順を設定することができます : ADC\_IN3、ADC\_IN8、ADC\_IN2、ADC\_IN2、ADC\_IN0、ADC\_IN2、ADC\_IN2、ADC\_IN15

- レギュラグループは、最大 16 の変換で構成されます。レギュラチャネルと変換シーケンス内での順序は、ADC\_SQRx レジスタで選択する必要があります。レギュラグループの合計変換数は、ADC\_SQR1 レジスタの L[3:0] ビットに書き込む必要があります。
- インジェクトグループは、最大 4 つの変換で構成されます。インジェクトチャネルと変換シーケンス内での順序は、ADC\_JSQR レジスタで選択する必要があります。インジェクトグループの合計変換数は、ADC\_JSQR レジスタの L[1:0] ビットに書き込む必要があります。

変換中に ADC\_SQRx または ADC\_JSQR レジスタが変更された場合、現在の変換はリセットされ、新しい開始パルスが ADC に送られて、新しく選ばれたグループを変換します。

### 温度センサ、 $V_{REFINT}$ 、および $V_{BAT}$ 内部チャネル

- 温度センサは VBAT と共用で ADC1\_IN18 チャネルに内部的に接続されています。一度に選択できるのは、温度センサまたは VBAT のうち、どちらか 1 つだけです。温度センサと VBAT 変換を同時に設定すると、VBAT 変換のみ行われます。  
内部基準電圧 VREFINT は ADC1\_IN17 に接続されています。

$V_{BAT}$  チャネルは ADC1\_IN18 チャネルに接続されています。これも、インジェクトまたはレギュラチャネルとして変換を行うことができます。

**注 :** 温度センサ、 $V_{REFINT}$  および  $V_{BAT}$  チャネルは、マスタ ADC1 ペリフェラルでのみ使用可能です。

## 15.3.5 シングル変換モード

シングル変換モードでは、ADC は変換を 1 回行います。CONT ビットが 0 の時、次のいずれかによってこのモードが開始します。

- ADC\_CR2 レジスタの SWSTART ビットをセットすること (レギュラチャネルのみ)
- JSWSTART ビットをセットすること (インジェクトチャネル)
- 外部トリガ (レギュラまたはインジェクトチャネル)

選択されたチャネルの変換が完了すると、

- レギュラチャネルが変換された場合：
  - 変換されたデータは 16 ビットの ADC\_DR レジスタに格納されます。
  - EOC (end of conversion) フラグがセットされます。
  - EOCIE ビットがセットされている場合、割り込みが生成されます。
- インジェクトチャネルが変換された場合：
  - 変換されたデータは 16 ビットの ADC\_JDR1 レジスタに格納されます。
  - JEOP (end of conversion injected) フラグがセットされます。
  - JEOCIE ビットがセットされている場合、割り込みが生成されます。

その後、ADC が停止します。

## 15.3.6 連続変換モード

連続変換モードでは、ADC は 1 回の変換が終了すると次の変換に移ります。CONT ビットが 1 の時、外部トリガまたは ADC\_CR2 レジスタの SWSTRT ビットをセットすることによって、このモードが開始されます (レギュラチャネルのみ)。

各変換後、

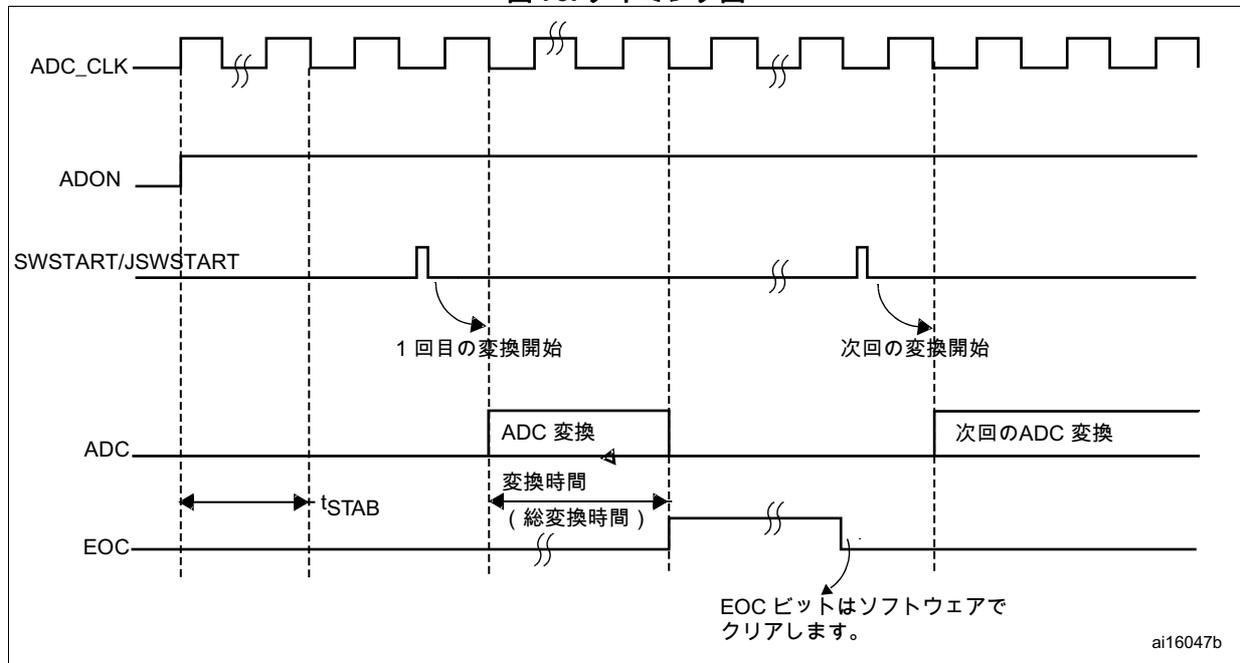
- レギュラチャネルのグループが変換された場合：
  - 最後に変換されたデータは 16 ビットの ADC\_DR レジスタに格納されます。
  - EOC (end of conversion) フラグがセットされます。
  - EOCIE ビットがセットされている場合、割り込みが生成されます。

**注：** *インジェクトチャネルを連続して変換することはできません。唯一の例外は、連続モードに設定されたレギュラチャネルの後にインジェクトチャネルを自動的に変換するように設定されている (JAUTO ビットを使用) 場合です。自動インジェクションセクションを参照してください。*

## 15.3.7 タイミング図

図 73 に示すように、ADC は正確な変換を開始する前に、安定時間  $t_{STAB}$  を必要とします。ADC 変換の開始後 15 クロックサイクルで EOC フラグがセットされ、変換されたデータが 16 ビットの ADC データレジスタに格納されます。

図 73. タイミング図



## 15.3.8 アナログウォッチドッグ

アナログウォッチドッグ (AWD) ステータスビットは、ADC によって変換されたアナログ電圧が低閾値を下回るか、高閾値を上回る場合にセットされます。これらの閾値は、ADC\_HTR および ADC\_LTR 16 ビットレジスタの下位 12 ビットに設定します。ADC\_CR1 レジスタの AWDIE ビットを使用して、割り込みを有効にできます。

閾値は、ADC\_CR2 レジスタの ALIGN ビットによって選択された配置とは無関係です。アナログ電圧は配置される前に低閾値および高閾値と比較されます。

表 91 に、1 つまたは複数のチャンネル上でアナログウォッチドッグを有効にするためにどのようにして ADC\_CR1 レジスタを設定するかを示します。

図 74. アナログウォッチドッグによって保護される領域



表 91. アナログウォッチドッグチャンネル選択

アナログウォッチドッグによって保護されるチャンネル	ADC_CR1 レジスタ制御ビット (x = 無視)		
	AWDSGL ビット	AWDEN ビット	JAWDEN ビット
なし	x	0	0
すべてのインジェクトチャンネル	0	0	1
すべてのレギュラチャンネル	0	1	0
すべてのレギュラおよびインジェクトチャンネル	0	1	1
1つの <sup>(1)</sup> インジェクトチャンネル	1	0	1
1つの <sup>(1)</sup> レギュラチャンネル	1	1	0
1つの <sup>(1)</sup> レギュラまたはインジェクトチャンネル	1	1	1

1. AWDCH[4:0] ビットによって選択

### 15.3.9 スキャンモード

このモードは、アナログチャンネルのグループをスキャンするために使用されます。

スキャンモードを選択するには、ADC\_CR1 レジスタの SCAN ビットをセットします。このビットがセットされると、ADC は ADC\_SQRx レジスタ (レギュラチャンネルの場合) または ADC\_JSQR レジスタ (インジェクトチャンネルの場合) で選択されたすべてのチャンネルをスキャンします。グループの各チャンネルに対して 1 回の変換が行われます。各変換が完了すると、グループ内の次のチャンネルが自動的に変換されます。CONT ビットがセットされている場合、グループの最後の選択チャンネルでレギュラチャンネル変換は停止せず、選択チャンネルの最初から続行します。

DMA ビットがセットされている場合、変換されたデータ (ADC\_DR レジスタに格納されている) は、各レギュラチャンネル変換後、ダイレクトメモリアクセス (DMA) コントローラを使用して、レギュラチャンネルグループから SRAM に転送されます。

次の場合に ADC\_SR レジスタの EOC ビットがセットされます。

- EOCs ビットが 0 にクリアされている場合、各レギュラグループシーケンス終了後
- EOCs ビットが 1 にセットされている場合、各レギュラチャンネル変換終了後

インジェクトチャンネルから変換されたデータは常に ADC\_JDRx レジスタに格納されます。

### 15.3.10 インジェクトチャンネルの管理

#### トリガインジェクション

トリガインジェクションを使用するには、ADC\_CR1 レジスタの JAUTO ビットがクリアされていなければなりません。

1. 外部トリガによって、または、ADC\_CR2 レジスタの SWSTART ビットをセットすることによって、レギュラチャンネルのグループの変換を開始します。
2. レギュラチャンネルグループの変換中に外部インジェクトトリガが発生した場合または JSWSTART ビットがセットされた場合、現在の変換はリセットされ、インジェクトチャンネルシーケンスはスキャンワンスモードに切り替わります。

- その後、レギュラチャネルグループのレギュラ変換が、最後に中断されたレギュラ変換から再開されます。  
インジェクト変換中にレギュライベントが発生した場合、インジェクト変換は中断されず、インジェクトシーケンスの終了時にレギュラシーケンスが実行されます。図 75 に対応するタイミング図を示します。

**注：** トリガインジェクションを使用する場合は、トリガイベントの間隔がインジェクションシーケンスより長くなるようにしなければなりません。たとえば、シーケンスの長さが 30 ADC クロックサイクルの場合 (3 クロック周期のサンプリング時間を持つ 2 つの変換)、トリガの最小間隔は 31 ADC クロックサイクルでなければなりません。

## 自動インジェクション

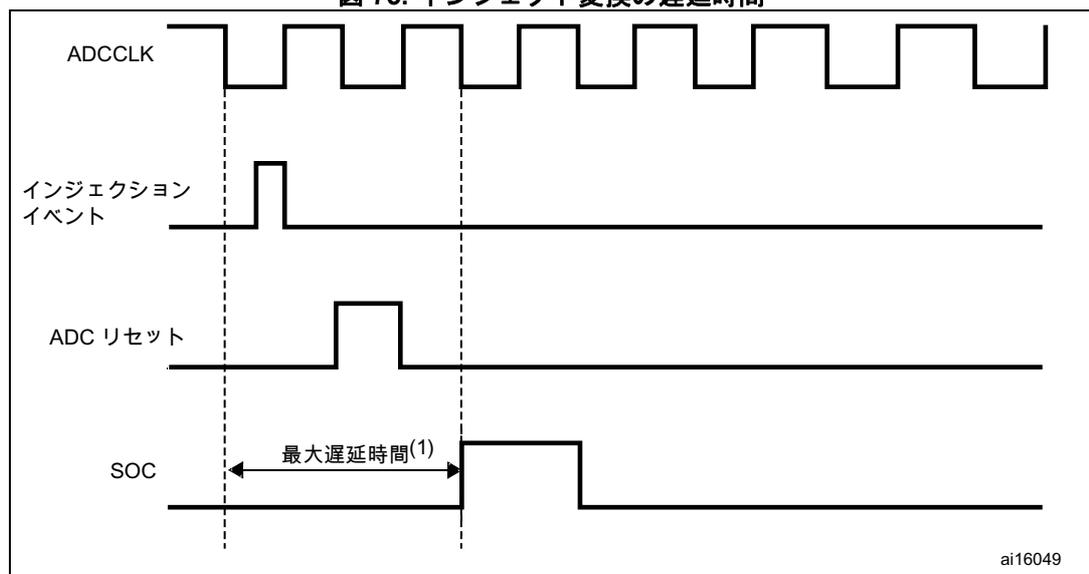
JAUTO ビットがセットされている場合、レギュラグループチャネルの変換後、インジェクトグループチャネルが自動的に変換されます。これを使用して、ADC\_SQRx レジスタおよび ADC\_JSQR レジスタにプログラミングされた最大 20 回の変換シーケンスを実行することができます。

このモードでは、インジェクトチャネルの外部トリガは無効でなければなりません。

JAUTO ビットに加えて CONT ビットもセットされている場合、レギュラチャネルに続いてインジェクトチャネルが連続して変換されます。

**注：** 自動インジェクトモードと不連続モードを同時に使用することはできません。

図 75. インジェクト変換の遅延時間



- 最大遅延時間の値は、STM32F75xxx および STM32F74xxx のデータシートの電気特性に記載されています。

## 15.3.11 不連続モード

### レギュラグループ

このモードは、ADC\_CR1 レジスタの DISCEN ビットをセットすることによって有効になります。これを使用して、ADC\_SQRx レジスタで選択された変換シーケンスの一部である短い変換シーケンス ( $n \leq 8$ ) を変換できます。n の値は、ADC\_CR1 レジスタの DISCNUM[2:0] ビットに書き込むことによって指定します。

外部トリガが発生すると、ADC\_SQRx レジスタで選択された次の n 個の変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC\_SQR1 レジスタの L[3:0] ビットによって定義されます。

例：

- n = 3、変換されるチャンネル = 0、1、2、3、6、7、9、10
- 最初のトリガ：0、1、2 が変換されます。各変換で EOC イベントが生成されます。
- 2 番目のトリガ：3、6、7 が変換されます。各変換で EOC イベントが生成されます。
- 3 番目のトリガ：9、10 が変換されます。各変換で EOC イベントが生成されます。
- 4 番目のトリガ：0、1、2 が変換されます。各変換で EOC イベントが生成されます。

**注：** *レギュラグループが不連続モードで変換される際には、ロールオーバーは発生しません。すべてのサブグループが変換されると、次のトリガでは、最初のサブグループの変換が開始されます。上の例では、4 番目のトリガで最初のサブグループのチャンネル 0、1、および 2 が再変換されます。*

### インジェクトグループ

このモードは、ADC\_CR1 レジスタの JDISCEN ビットをセットすることで有効になります。これを使用して、ADC\_JSQR レジスタで選択されたシーケンスを、外部トリガイイベント後に 1 チャンネルずつ変換することができます。

外部トリガが発生すると、ADC\_JSQR レジスタで選択された次のチャンネル変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC\_JSQR レジスタの JL[1:0] ビットによって定義されます。

例：

- n = 1、変換されるチャンネル = 1、2、3
- 最初のトリガ：チャンネル 1 が変換されます。
- 2 番目のトリガ：チャンネル 2 が変換されます。
- 3 番目のトリガ：チャンネル 3 が変換され、JEOC イベントが生成されます。
- 4 番目のトリガ：チャンネル 1 が変換されます。

**注：** *すべてのインジェクトチャンネルが変換されると、次のトリガでは、最初のインジェクトチャンネルの変換が開始されます。上の例では、4 番目のトリガで最初のインジェクトチャンネル 1 が再変換されます。自動インジェクトモードと不連続モードを同時に使用することはできません。レギュラグループとインジェクトグループの両方で不連続モードを同時に設定することはできません。不連続モードは、1 つのグループ変換に対してのみ有効にする必要があります。*

## 15.4 データの配置

ADC\_CR2 レジスタの ALIGN ビットは、変換後に格納されるデータの配置を選択します。データは、[図 76](#) および [図 77](#) に示すように、右詰めまたは左詰めに配置できます。

インジェクトグループチャンネルの変換後のデータ値は、ADC\_JOFRx レジスタに書き込まれたユーザー定義オフセットだけ減算されるので結果は負の値になることもあります。SEXT ビットは、拡張符号値を表します。

レギュラグループのチャンネルの場合、オフセットは引かれないので、12 ビットのみが有効です。

図 76. 12 ビットデータの右詰め

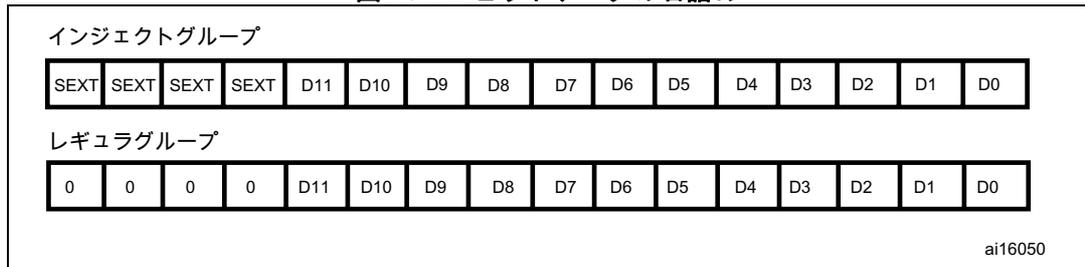
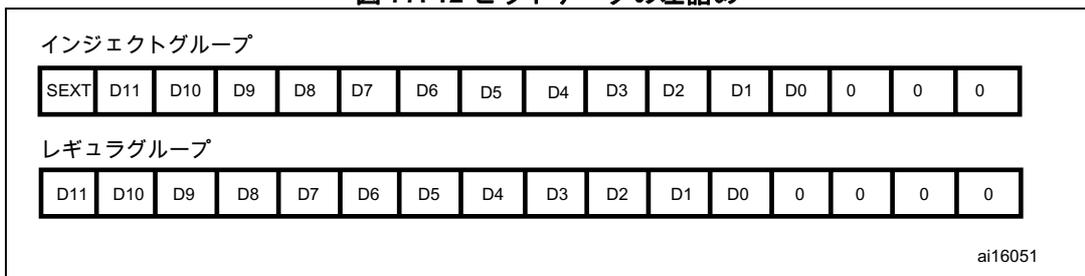
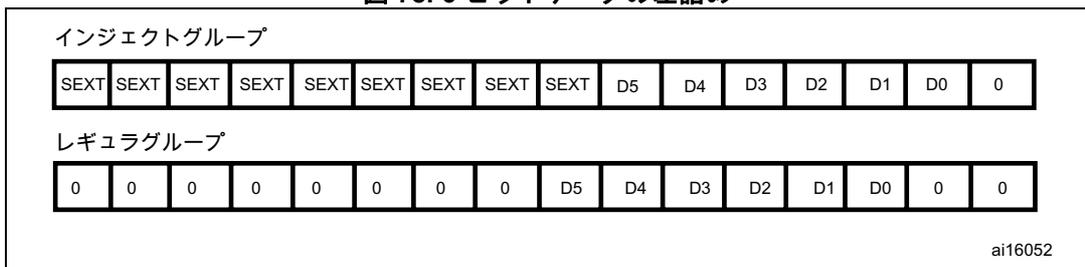


図 77. 12 ビットデータの左詰め



特殊なケース：左詰めの場合、バイト単位に配置される場合で分解能が 6 ビットに設定されている場合 ([図 78](#)) を除きデータはハーフワード単位に配置されます。

図 78. 6 ビットデータの左詰め



## 15.5 チャンネル単位でプログラム可能なサンプリング時間

ADC は、一定の ADCCLK サイクル数だけ入力電圧をサンプリングしますが、この時間は、ADC\_SMPR1 および ADC\_SMPR2 レジスタの SMP[2:0] ビットを使用して変更できます。チャンネルごとに異なるサンプリング時間でサンプリングすることができます。

合計変換時間は、次のように計算されます。

$$T_{\text{conv}} = \text{サンプリング時間} + 12 \text{ サイクル}$$

例：

ADCCLK = 30 MHz、サンプリング時間が 3 サイクルのとき：

$$T_{\text{conv}} = 3 + 12 = 15 \text{ サイクル} = 0.5 \mu\text{s} \text{ (APB2 が 60 MHz の場合)}$$

## 15.6 外部トリガによる変換およびトリガ極性

変換は、外部イベント (タイマキャプチャ、EXTI ラインなど) によってトリガできます。EXTEN[1:0] 制御ビット (レギュラ変換の場合) または JEXTEN[1:0] ビット (インジェクト変換の場合) が "0b00" 以外の場合、選択した極性で外部イベントによって変換をトリガすることができます。表 92 に EXTEN[1:0] と JEXTEN[1:0] の値とトリガ極性の対応を示します。

表 92. トリガ極性の設定

転送元	EXTEN[1:0] / JEXTEN[1:0]
トリガ検出は無効です。	00
立ち上がりエッジで検出します。	01
立ち下がりエッジで検出します。	10
立ち上がりエッジと立ち下がりエッジの両方で検出します。	11

**注：** 外部トリガの極性は動作中に変更することができます。

EXTSEL[3:0] および JEXTSEL[3:0] 制御ビットを使用してレギュラおよびインジェクトグループの変換をトリガできるイベントを 16 のイベントから選択します。

表 93 に、レギュラ変換に使用できる外部トリガを示します。

表 93. レギュラチャネルの外部トリガ

転送元	タイプ	EXTSEL[3:0]
TIM1_CC1 イベント	オンチップタイマからの内部信号	0000
TIM1_CC2 イベント		0001
TIM1_CC3 イベント		0010
TIM2_CC2 イベント		0011
TIM5_TRGO イベント		0100
TIM4_CC4 イベント		0101
TIM3_CC4		0110
TIM8_TRGO イベント		0111
TIM8_TRGO(2) イベント		1000
TIM1_TRGO イベント		1001
TIM1_TRGO(2) イベント		1010
TIM2_TRGO イベント		1011
TIM4_TRGO イベント		1100
TIM6_TRGO イベント		1101
EXTI ライン 11		外部ピン

表 94 に、インジェクト変換に使用できる外部トリガを示します。

表 94. インジェクトチャネルの外部トリガ

転送元	接続タイプ	JEXTSEL[3:0]
TIM1_TRGO イベント	オンチップタイマからの内部信号	0000
TIM1_CC4 イベント		0001
TIM2_TRGO イベント		0010
TIM2_CC1 イベント		0011
TIM3_CC4 イベント		0100
TIM4_TRGO イベント		0101
TIM8_CC4 イベント		0111
TIM1_TRGO(2) イベント	オンチップタイマからの内部信号	1000
TIM8_TRGO イベント		1001
TIM8_TRGO(2) イベント		1010
TIM3_CC3 イベント		1011
TIM5_TRGO イベント		1100
TIM3_CC1 イベント		1101
TIM6_TRGO イベント		1110

ソフトウェアソーストリガイベントは、ADC\_CR2 レジスタの SWSTART (レギュラ変換の場合) ビットまたは JSWSTART (インジェクト変換の場合) ビットをセットすることによって生成できます。

レギュラグループ変換は、インジェクトトリガによって中断することが可能です。

**注：** トリガの選択は動作中に変更することができます。ただし、選択が変更される時 1 APB クロックサイクルの期間トリガの検出が無効になります。これは、変更時の誤検出を避けるためです。

## 15.7 高速変換モード

ADC の分解能を下げることによって、高速変換を行うことができます。RES ビットを使用してデータレジスタで使用できるビット数を選択します。各精度に対する最小変換時間は、次のとおりです。

- 12 ビット :  $3 + 12 = 15$  ADCCLK サイクル
- 10 ビット :  $3 + 10 = 13$  ADCCLK サイクル
- 8 ビット :  $3 + 8 = 11$  ADCCLK サイクル
- 6 ビット :  $3 + 6 = 9$  ADCCLK サイクル

## 15.8 データ管理

### 15.8.1 DMA の使用

変換されたレギュラチャンネルの値は特定のデータレジスタに格納されるので、複数のレギュラチャンネルの変換には DMA の使用が便利です。これによって、ADC\_DR レジスタにすでに格納されているデータの損失を防ぐことができます。

DMA モードが有効 (ADC\_CR2 レジスタの DMA ビットが 1 に設定されている) な場合、レギュラチャンネルが変換されるごとに DMA リクエストが生成されます。これにより、変換されたデータを ADC\_DR レジスタからソフトウェアで選択した場所へ転送することができます。

これにもかかわらず、データが失われる場合 (オーバーラン)、ADC\_SR レジスタの OVR ビットがセットされ割り込みが生成されます (OVRIE イネーブルビットがセットされている場合)。続いて DMA 転送が無効になり、その後の DMA リクエストは受け付けられません。この場合、DMA リクエストが行われると、実行中のレギュラ変換は中断され、その後のレギュラトリガは無視されます。そのときは、必要な変換済みチャンネルデータを正しいメモリ位置へ転送するために OVR フラグおよび使用している DMA ストリームの DMAEN ビットをクリアして DMA と ADC を再初期化する必要があります。その後初めて、変換を再開することができ、データ転送を再び有効にすることができます。インジェクトチャンネル変換はオーバーランエラーの影響を受けません。

DMA モードで OVR = 1 の場合、最後の有効なデータが転送されたあとに DMA リクエストがブロックされます。つまり、RAM へ転送されたデータはすべて有効とみなすことができます。

最後の DMA 転送 (DMA コントローラの DMA\_SxNTR レジスタに設定されている転送数) が終了すると、

- ADC\_CR2 レジスタの DDS ビットが 0 にクリアされている場合、DMA コントローラへの新規の DMA リクエストは発生しません (これによりオーバーランエラーの発生を防ぎます)。ただし、DMA ビットはハードウェアによってクリアされることはありません。新たな転送を開始するにはそこに 0 を書き込み、そのあと 1 を書き込む必要があります。
- DDS ビットが 1 にセットされている場合、リクエストの生成は継続することができます。これにより DMA をダブルバッファサーキュラモードに設定することができます。



DMA 使用時に ADC を OVR 状態から回復するには次の手順に従います。

1. DMA を再初期化します (転送先アドレスと NDTR カウンタを調整します)
2. ADC\_SR レジスタの ADC OVR ビットをクリアします。
3. ADC をトリガして変換を開始します。

## 15.8.2 DMA を使用しない変換シーケンスの管理

変換が十分遅い場合、ソフトウェアで変換シーケンスを処理することができます。この場合、変換シーケンスの終わりだけではなく各変換が終わるたびに EOC ステータスビットをセットするため、ADC\_CR2 レジスタの EOCS ビットをセットする必要があります。EOCS = 1 のとき、オーバーラン検出は自動的に有効になります。したがって、変換が終了するごとに EOC がセットされ ADC\_DR レジスタを読み出すことができます。オーバーラン管理は DMA を使用するときと同じです。

EOCS がセットされているとき ADC を OVR 状態から回復するには次の手順に従います。

1. ADC\_SR レジスタの ADC OVR ビットをクリアします。
2. ADC をトリガして変換を開始します。

## 15.8.3 DMA およびオーバーラン検出を使用しない変換

毎回データの読み出しをせずに ADC に 1 つまたは複数のチャンネルを変換させると便利な場合があります (たとえば、アナログウォッチドッグがある場合)。そのためには、DMA を無効 (DMA = 0) にし、シーケンスの終了時にのみ EOC ビットをセットする必要があります (EOCS = 0)。この設定では、オーバーラン検出が無効になります。

## 15.9 マルチ ADC モード

2 つ以上の ADC を持つデバイスでは、デュアル ADC モード (2 つの ADC) およびトリプル ADC モード (3 つの ADC) を使用することができます (図 79 を参照)。

マルチ ADC モードでは、変換の開始は、ADC\_CCR レジスタの MULTI[4:0] ビットによって選択されたモードに応じて、ADC1 マスタにより ADC2 および ADC3 スレーブに交互または同時にトリガされます。

**注：** マルチ ADC モードでは、変換トリガを外部イベントに設定する場合、誤ったトリガが不要なスレーブの変換を開始しないように、アプリケーションはマスタのトリガのみをセットし、スレーブのトリガを無効にする必要があります。

次の 4 つのモードを備えています。

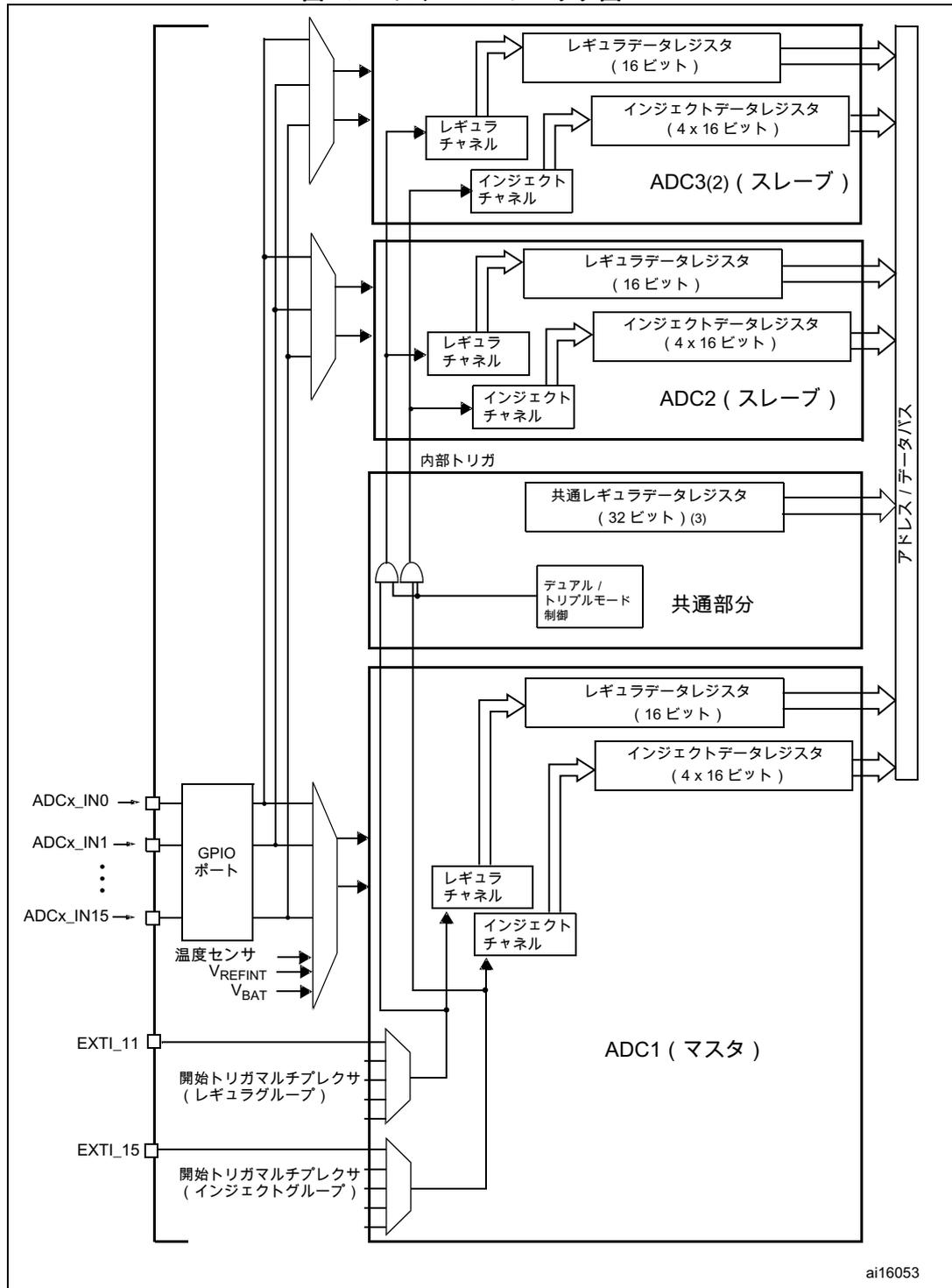
- インジェクト同時モード
- レギュラ同時モード
- インタリーブモード
- オルタネートトリガモード

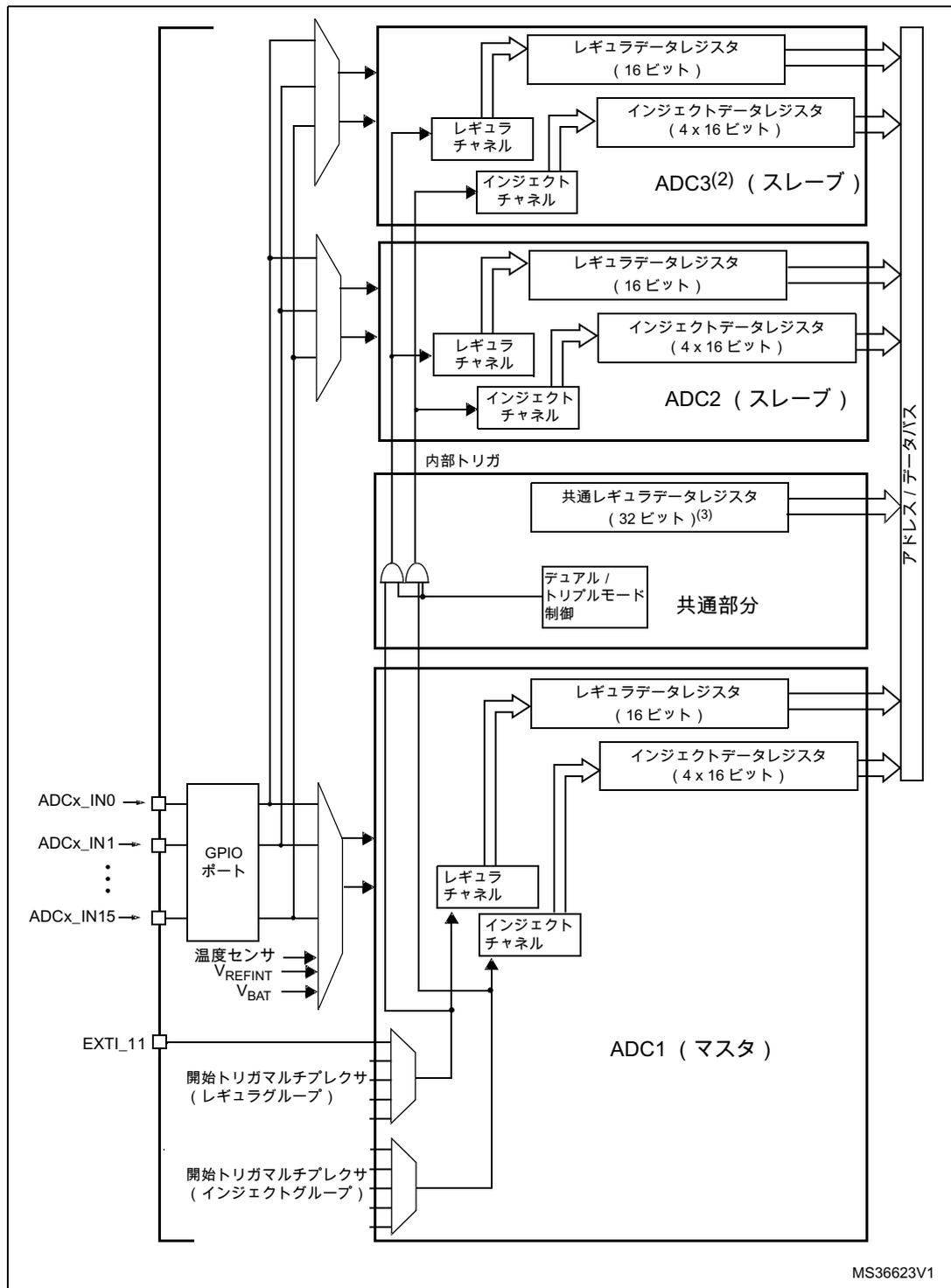
これらのモードを次のように組み合わせて使用することも可能です。

- インジェクト同時モード + レギュラ同時モード
- レギュラ同時モード + オルタネートトリガモード

**注：** マルチ ADC モードでは、変換されたデータをマルチモードデータレジスタ (ADC\_CDR) から読み出すことができます。ステータスビットはマルチモードステータスレジスタ (ADC\_CSR) から読み出すことができます。

図 79. マルチ ADC ブロック図 (1)





MS36623V1

1. 外部トリガは ADC2 および ADC3 にもありますが、この図には示していません。
2. デュアル ADC モードでは、ADC3 スレーブ部分はがありません。
3. トリプル ADC モードでは、ADC 共通データレジスタ (ADC\_CDR) に ADC1、ADC2、および ADC3 のレギュラ変換データが格納されます。全 32 レジスタビットは、選択したストレージ順に従って使用されます。デュアル ADC モードでは、ADC 共通データレジスタ (ADC\_CDR) に ADC1 および ADC2 の両方のレギュラ変換データが格納されます。全 32 レジスタビットを使用します。

- マルチ ADC モードにおける DMA リクエスト :
 

マルチ ADC モードでは、変換されたデータを 3 つの異なるモードで転送するよう DMA を設定することができます。すべての場合において ADC に接続されている DMA ストリームを使用します。

  - **DMA モード 1 :** DMA リクエストごとに (データ項目は 1 つ使用可能)、ADC 変換済データを表すハーフワードが転送されます。
 

デュアル ADC モードでは、最初のリクエストで ADC1 のデータが転送され、2 番目のリクエストで ADC2 のデータが転送され、以下同様です。

トリプル ADC モードでは、最初のリクエストで ADC1 のデータが転送され、2 番目のリクエストで ADC2 のデータが転送され、3 番目のリクエストで ADC3 のデータが転送され、以下このシーケンスを繰り返します。このように、DMA はまず ADC1 のデータを転送し、次に ADC2 のデータ、ADC3 のデータと続きます。

DMA モード 1 はレギュラ同時トリプルモードで使用します。

**例 :**

レギュラ同時トリプルモード : 3 つの連続 DMA リクエストが生成されます (変換データ項目ごとに 1 つ)。

1 番目のリクエスト : ADC\_CDR[31:0] = ADC1\_DR[15:0]  
 2 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0]  
 3 番目のリクエスト : ADC\_CDR[31:0] = ADC3\_DR[15:0]  
 4 番目のリクエスト : ADC\_CDR[31:0] = ADC1\_DR[15:0]
  - **DMA モード 2 :** DMA リクエストごとに (データ項目は 2 つ使用可能)、2 つの ADC 変換データを表す 2 つのハーフワードが 1 ワードとして転送されます。
 

デュアル ADC モードでは、最初のリクエストで ADC2 と ADC1 の両データが転送され (ADC2 データが上位ハーフワードを使用し、ADC1 データが下位ハーフワードを使用します)、以下同様です。

トリプル ADC モードでは、3 つの DMA リクエストが生成されます。最初のリクエストで、ADC2 と ADC1 の両データが転送されます (ADC2 データが上位ハーフワードを使用し、ADC1 データが下位ハーフワードを使用します)。2 番目のリクエストで、ADC1 と ADC3 の両データが転送されます (ADC1 データが上位ハーフワードを使用し、ADC3 データが下位ハーフワードを使用します)。3 番目のリクエストで、ADC3 と ADC2 の両データが転送され (ADC3 データが上位ハーフワードを使用し、ADC2 データが下位ハーフワードを使用します)、以降同様です。

DMA モード 2 はインタリーブモードおよびレギュラ同時モード (デュアル ADC モードのみ) で使用されます。

**例 :**

a) インタリーブデュアルモード : 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。

1 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0] | ADC1\_DR[15:0]  
 2 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0] | ADC1\_DR[15:0]

b) インタリーブトリプルモード : 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。

1 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0] | ADC1\_DR[15:0]  
 2 番目のリクエスト : ADC\_CDR[31:0] = ADC1\_DR[15:0] | ADC3\_DR[15:0]  
 3 番目のリクエスト : ADC\_CDR[31:0] = ADC3\_DR[15:0] | ADC2\_DR[15:0]  
 4 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0] | ADC1\_DR[15:0]

- **DMA モード 3** : このモードは DMA モード 2 に類似しています。唯一の違いは、DMA リクエストごとに (データ項目は 2 つ使用可能)、2 つの ADC 変換データを表す 2 バイトが ハーフワードとして転送されます。データ転送順序は、DMA モード 2 に類似しています。DMA モード 3 は 6 ビットおよび 8 ビット分解能でのインタリーブモードで使用します。

例 :

- a) インタリーブデュアルモード : 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。  
1 番目のリクエスト :  $ADC\_CDR[15:0] = ADC2\_DR[7:0] | ADC1\_DR[7:0]$   
2 番目のリクエスト :  $ADC\_CDR[15:0] = ADC2\_DR[7:0] | ADC1\_DR[7:0]$
- b) インタリーブトリプルモード : 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。  
1 番目のリクエスト :  $ADC\_CDR[15:0] = ADC2\_DR[7:0] | ADC1\_DR[7:0]$   
2 番目のリクエスト :  $ADC\_CDR[15:0] = ADC1\_DR[7:0] | ADC3\_DR[15:0]$   
3 番目のリクエスト :  $ADC\_CDR[15:0] = ADC3\_DR[7:0] | ADC2\_DR[7:0]$   
4 番目のリクエスト :  $ADC\_CDR[15:0] = ADC2\_DR[7:0] | ADC1\_DR[7:0]$

**オーバーラン検出** : 当該の ADC (デュアルオードおよびトリプルモードにおける ADC1 と ADC2、トリプルモードにおける ADC3 のみ) の 1 つでオーバーランが検出された場合、RAM に転送されたすべてのデータが有効であることを保証するため DMA リクエストの発行を停止します。ある ADC のデータレジスタに有効なデータが格納されていることによりその ADC に対応する EOC ビットがセットされたままになることがあります。

## 15.9.1 インジェクト同時モード

このモードは、インジェクトチャネルグループを変換します。外部トリガのソースは、ADC1 のインジェクトグループマルチプレクサ (ADC1\_CR2 レジスタの JEXTSEL[3:0] ビットによって選択) から供給されます。同時トリガは、ADC2 と ADC3 に供給されます。

**注 :** *2 つ/3 つの ADC の同じチャネルを変換しないでください。(同じチャネルを変換する場合、2 つ/3 つの ADC のサンプリング時間が重なってはいけません。)*

*同時モードでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つ (デュアル ADC モード) / 3 つ (トリプル ADC モード) のシーケンスのうち長い方のシーケンスより長くなるようにしなければなりません。さもないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。*

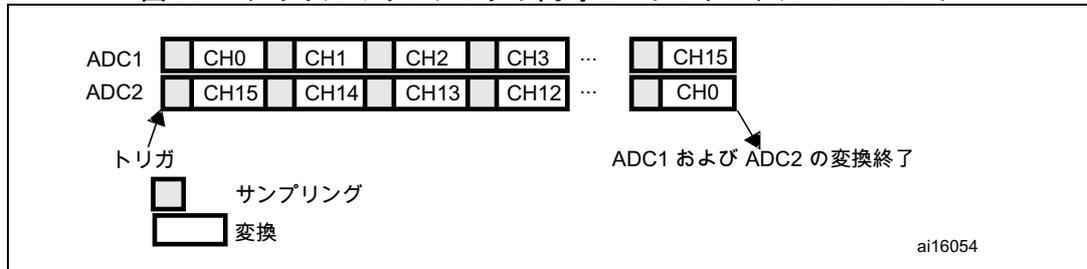
*レギュラ変換であれば、1 つまたはすべての ADC で行うことができます。その場合、それらは互いに独立していてインジェクトイベントが発生すると中断されます。それら (レギュラ変換) はインジェクト変換グループ終了時に再開されます。*

## デュアル ADC モード

ADC1 または ADC2 の変換終了イベント時に、

- 変換されたデータは、各 ADC インタフェースの ADC\_JDRx レジスタに格納されます。
- ADC1/ADC2 のインジェクトチャンネルがすべて変換されると、JEOC 割り込みが生成されます (2つの ADC インタフェースの 1 つで有効な場合)。

図 80.4 チャンネルのインジェクト同時モード：デュアル ADC モード

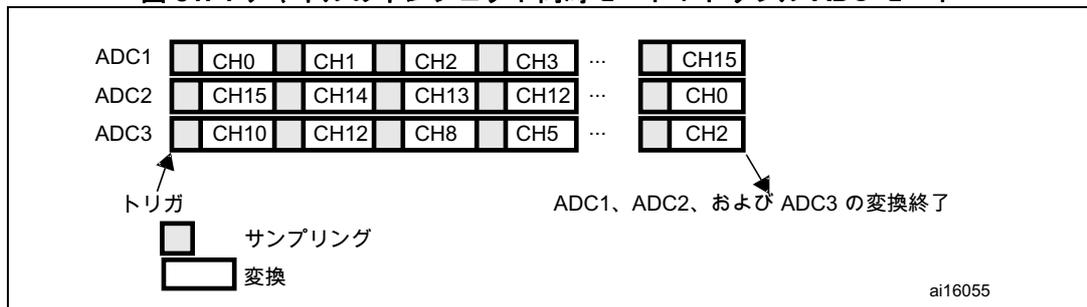


## トリプル ADC モード

ADC1、ADC2 または ADC3 の変換終了イベント時に、

- 変換されたデータは、各 ADC インタフェースの ADC\_JDRx レジスタに格納されます。
- ADC1/ADC2/ADC3 のインジェクトチャンネルがすべて変換されると、JEOC 割り込みが生成されます (3つの ADC インタフェースの 1 つで有効な場合)。

図 81.4 チャンネルのインジェクト同時モード：トリプル ADC モード



## 15.9.2 レギュラ同時モード

このモードは、レギュラチャンネルグループに対して行われます。外部トリガのソースは、ADC1 のレギュラグループマルチプレクサ (ADC1\_CR2 レジスタの EXTSEL[3:0] ビットによって選択) から供給されます。同時トリガは、ADC2 と ADC3 に供給されます。

**注：** 2 つ/3 つの ADC の同じチャンネルを変換しないでください。(同じチャンネルを変換する場合、2 つ/3 つの ADC のサンプリング時間が重なってはいけません。)

レギュラ同時モードでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つ (デュアル ADC モード) / 3 つ (トリプル ADC モード) のシーケンスのうち最も長い変換時間より長くなるようにしなければなりません。さもないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

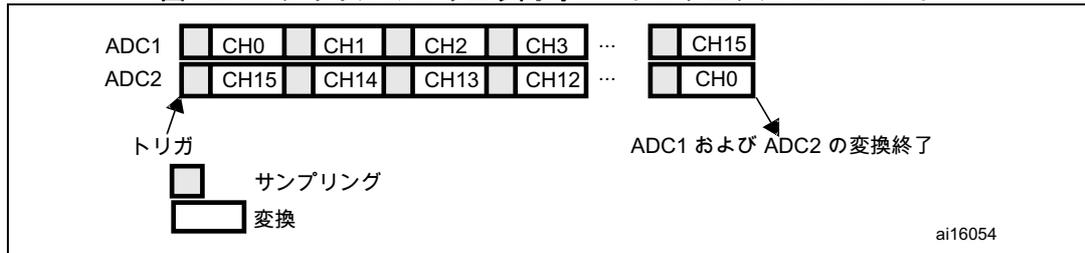
インジェクト変換は無効にする必要があります。

## デュアル ADC モード

ADC1 または ADC2 の変換終了イベント時に、

- 32 ビット DMA 転送リクエストが生成されます (ADC\_CCR レジスタの DMA[1:0] ビットが 0b10 の場合)。このリクエストによって、ADC\_CDR 32 ビットレジスタの上位ハーフワードに格納されている ADC2 変換データは SRAM に転送され、そのとき ADC\_CCR の下位ハーフワードに格納されている ADC1 変換データが SRAM に転送されます。
- ADC1/ADC2 のレギュラチャンネルがすべて変換されると、EOC 割り込みが生成されます (2 つの ADC インタフェースの 1 つで有効な場合)。

図 82. 16 チャンネルのレギュラ同時モード：デュアル ADC モード

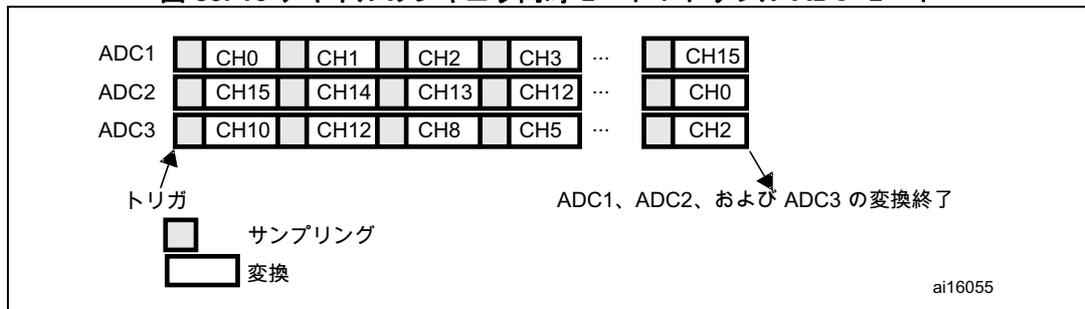


## トリプル ADC モード

ADC1、ADC2 または ADC3 の変換終了イベント時に、

- 3 つの 32 ビット DMA 転送リクエストが生成されます (ADC\_CCR レジスタの DMA[1:0] ビットが 0b01 の場合)。その後 ADC\_CDR 32 ビットレジスタから SRAM へ 3 つの転送が行われます。最初は ADC1 変換データ、次に ADC2 変換データ、最後に ADC3 変換データです。この処理を新たな 3 つの変換ごとに繰り返します。
- ADC1/ADC2/ADC3 レギュラチャンネルがすべて変換されると、EOC 割り込みが生成されます (3 つの ADC インタフェースの 1 で有効な場合)。

図 83. 16 チャンネルのレギュラ同時モード：トリプル ADC モード



## 15.9.3 インタリーブモード

このモードは、レギュラグループ（通常は 1 つのチャンネル）でのみ開始できます。外部トリガソースは、ADC1 のレギュラチャンネルマルチプレクサから供給されます。

### デュアル ADC モード

外部トリガが発生した後、

- ADC1 はただちに開始し、
- ADC2 は数 ADC クロックサイクルの遅延後に開始します。

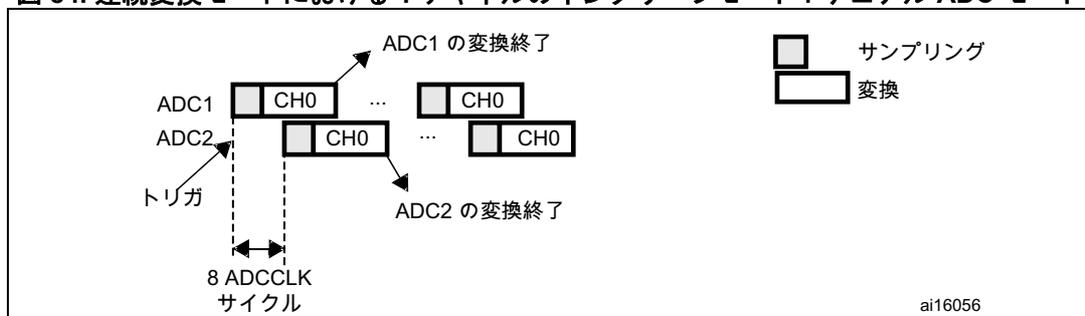
インタリーブモードにおける 2 つの変換間の最小遅延は ADC\_CCR レジスタの DELAY ビットで設定します。ただし、ADC は相補 ADC が依然として入力をサンプリングしている場合、変換を開始できません（一度に 1 つの ADC のみ入力信号をサンプリングすることができます）。この場合、遅延は、サンプリング時間 + 2 ADC クロックサイクルとなります。たとえば、DELAY = 5 クロックサイクルで両 ADC のサンプリングに 15 クロックサイクルかかる場合、ADC1 と ADC2 変換間は 17 クロックサイクルです。

ADC1 と ADC2 の両方で CONT ビットがセットされている場合、両方の ADC の選択されたレギュラチャンネルが連続的に変換されます。

**注：** *変換シーケンスが中断されると（たとえば、DMA 転送終了が発生）、インタリーブモードを再プログラムする前にまずマルチ ADC シーケンスを独立モードに設定することによって（ビット DUAL[4:0]=00000）リセットする必要があります。*

ADC2 により EOC 割り込みが生成された後（EOCIE ビットによって有効になっている場合）、32 ビット DMA 転送リクエストが生成されます（ADC\_CCR の DMA[1:0] ビットが 0b10 の場合）。このリクエストによってまず ADC\_CDR 32 ビットレジスタの上位ハーフワードに格納されている ADC2 変換データは SRAM に転送され、そのときこのレジスタの下位ハーフワードに格納されている ADC1 変換データが SRAM に転送されます。

図 84. 連続変換モードにおける 1 チャンネルのインタリーブモード：デュアル ADC モード



### トリプル ADC モード

外部トリガが発生した後、

- ADC1 はただちに開始し、
- ADC2 は数 ADC クロックサイクルの遅延後に開始します。
- ADC3 は ADC2 変換からさらに数 ADC クロックサイクルの遅延後に開始します。

インタリーブモードにおける 2 つの変換間の最小遅延は ADC\_CCR レジスタの DELAY ビットで設定します。ただし、ADC は相補 ADC が依然として入力をサンプリングしている場合、変換を開始できません（一度に 1 つの ADC のみ入力信号をサンプリングすることができます）。この場合、遅延は、サンプリング時間 + 2 ADC クロックサイクルとなります。たとえば、DELAY = 5 クロックサイ

クルで 3 つの ADC のサンプリングに 15 クロックサイクルかかる場合、ADC1、ADC2、および ADC3 の変換間は 17 クロックサイクルです。

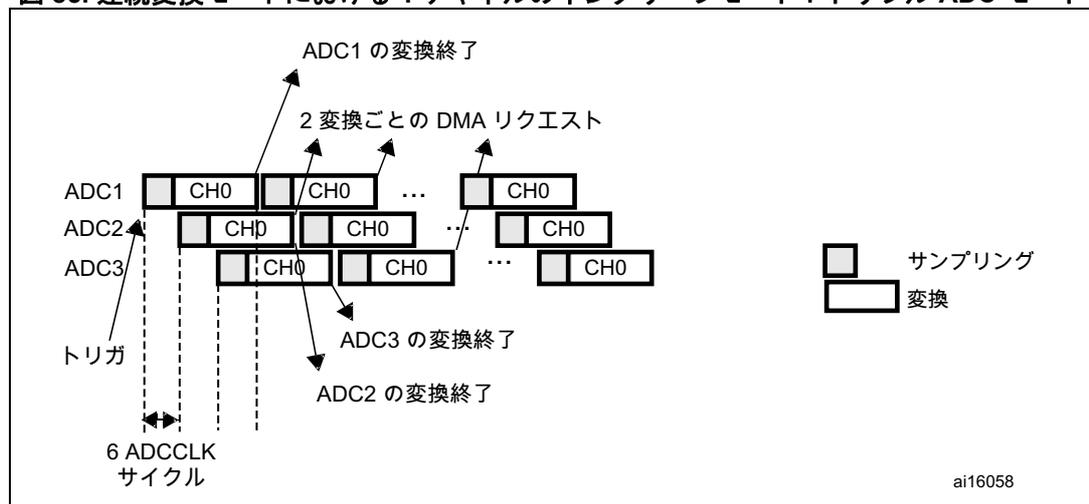
ADC1、ADC2、および ADC3 で CONT ビットがセットされている場合、すべての ADC の選択されたレギュラチャネルが連続的に変換されます。

**注：** *変換シーケンスが中断されると (たとえば、DMA 転送終了が発生)、インタリーブモードを再プログラムする前にまずマルチ ADC シーケンサを独立モードに設定することによって (ビット DUAL[4:0]=00000) リセットする必要があります。*

このモードでは 2 つのデータ項目が使用可能になるごとに DMA 転送リクエストが生成されます (ADC\_CCR レジスタの DMA[1:0] ビットが 0b10 の場合)。このリクエストによってまず ADC\_CDR32 ビットレジスタの下位ハーフワードに格納されている変換データが SRAM に転送され、そのとき ADC\_CDR の上位ハーフワードに格納されている 2 番目の変換データが SRAM に転送されます。シーケンスは、次のとおりです。

- 1 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0] | ADC1\_DR[15:0]
- 2 番目のリクエスト : ADC\_CDR[31:0] = ADC1\_DR[15:0] | ADC3\_DR[15:0]
- 3 番目のリクエスト : ADC\_CDR[31:0] = ADC3\_DR[15:0] | ADC2\_DR[15:0]
- 4 番目のリクエスト : ADC\_CDR[31:0] = ADC2\_DR[15:0] | ADC1\_DR[15:0], ...

図 85. 連続変換モードにおける 1 チャネルのインタリーブモード : トリプル ADC モード



## 15.9.4 オルタネートトリガモード

このモードは、インジェクトグループでのみ開始できます。外部トリガのソースは、ADC1 のインジェクトグループマルチプレクサです。

**注：** *レギュラ変換であれば 1 つまたはすべての ADC で有効にすることができます。この場合、レギュラ変換は互いに独立しています。ADC がインジェクト変換を行う必要があるとき、レギュラ変換は中断されます。インジェクト変換が終了すると (レギュラ変換が) 再開されます。*

*変換シーケンスが中断されると (たとえば、DMA 転送終了が発生)、インタリーブモードを再プログラムする前にまずマルチ ADC シーケンサを独立モードに設定することによって (ビット DUAL[4:0]=00000) リセットする必要があります。*

*2 つのトリガイベント間の間隔は 1 ADC クロック期間と同等か長い必要があります。同じ ADC で変換を開始する 2 つのトリガイベント間の最小間隔はシングル ADC モードと同じです。*

## デュアル ADC モード

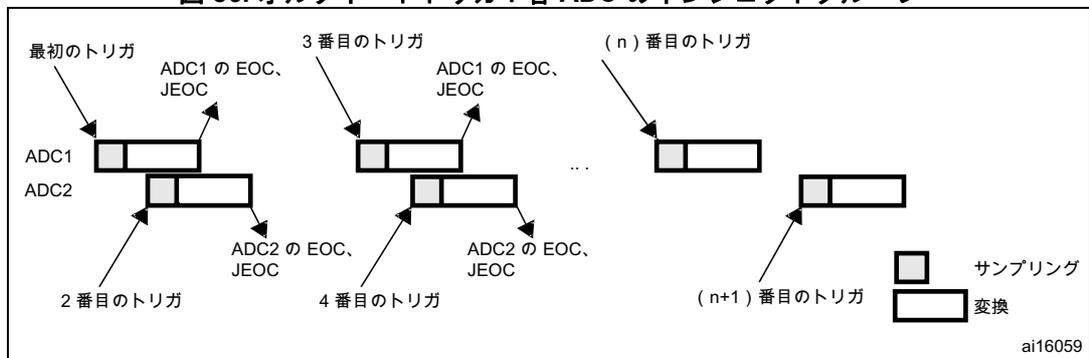
- 最初のトリガが発生すると、グループ内のすべてのインジェクト ADC1 チャンネルが変換されます。
- 2 番目のトリガが発生すると、グループ内のすべてのインジェクト ADC2 チャンネルが変換されます。
- 以下同様です。

グループ内のすべてのインジェクト ADC1 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内のすべてのインジェクト ADC2 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内すべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、グループ内のインジェクト ADC1 チャンネルを変換することによって、オルタネートトリガプロセスがリスタートします。

図 86. オルタネートトリガ : 各 ADC のインジェクトグループ



ADC1 と ADC2 の両方に対してインジェクト不連続モードが有効な場合、

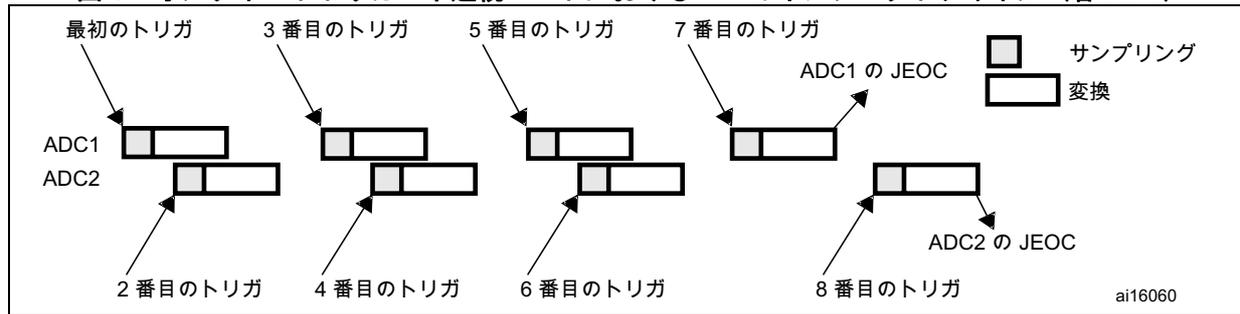
- 最初のトリガが発生すると、最初のインジェクト ADC1 チャンネルが変換されます。
- 2 番目のトリガが発生すると、最初のインジェクト ADC2 チャンネルが変換されます。
- 以下同様です。

グループ内のすべてのインジェクト ADC1 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内のすべてのインジェクト ADC2 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内のすべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、オルタネートトリガプロセスがリスタートします。

図 87. オルタネートトリガ：不連続モードにおける 4 つのインジェクトチャンネル (各 ADC)



### トリプル ADC モード

- 最初のトリガが発生すると、グループ内のすべてのインジェクト ADC1 チャンネルが変換されます。
- 2 番目のトリガが発生すると、グループ内のすべてのインジェクト ADC2 チャンネルが変換されます。
- 3 番目のトリガが発生すると、グループ内のすべてのインジェクト ADC3 チャンネルが変換されます。
- 以下同様です。

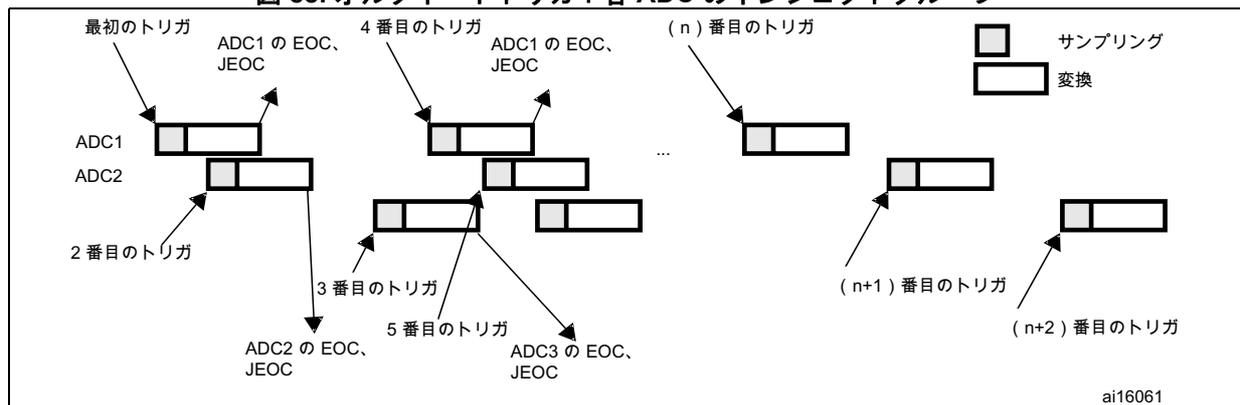
グループ内のすべてのインジェクト ADC1 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内のすべてのインジェクト ADC2 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内のすべてのインジェクト ADC3 チャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内すべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、グループ内のインジェクト ADC1 チャンネルを変換することによって、オルタネートトリガプロセスがリスタートします。

図 88. オルタネートトリガ：各 ADC のインジェクトグループ



## 15.9.5 レギュラ/インジェクト同時モードの組み合わせ

レギュラグループの同時変換を中断して、インジェクトグループの同時変換を開始することができます。

**注：** レギュラ/インジェクト同時モードの組み合わせでは、同じ長さのシーケンスを変換するか、トリガ間隔が2つ (デュアルADCモード) / 3つ (トリプルADCモード) のシーケンスのうち最も長い変換時間より長くなるようにしなければなりません。さもないと、長いシーケンスのADCが前の変換を完了する前に、短いシーケンスのADCがリスタートすることがあります。

## 15.9.6 レギュラ同時 + オルタネートトリガモードの組み合わせ

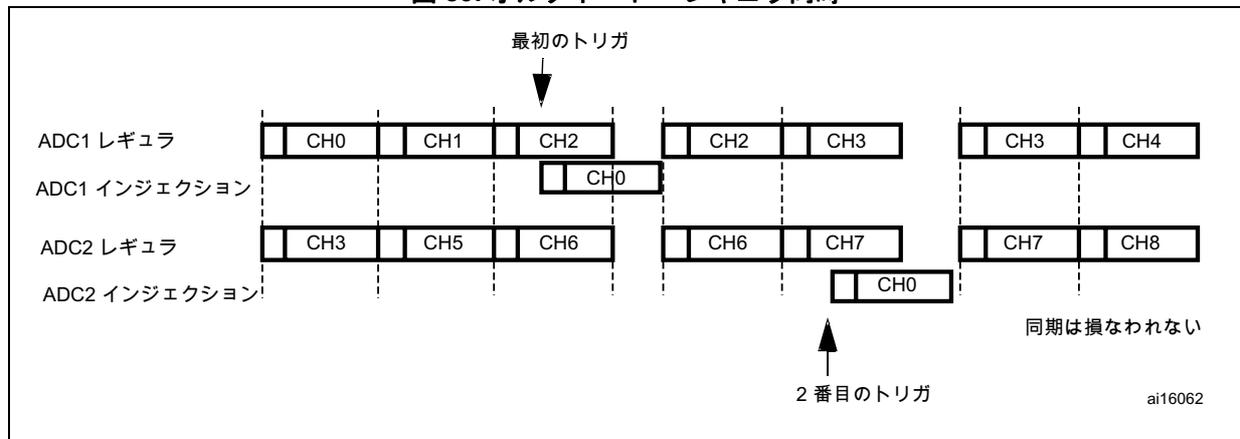
レギュラグループの同時変換を中断して、インジェクトグループのオルタネートトリガ変換を開始することができます。図 89は、同時レギュラ変換に割り込むオルタネートトリガの動作を示します。

インジェクトオルタネート変換は、インジェクトイベント後、ただちに開始されます。すでにレギュラ変換の実行中であった場合、インジェクト変換後の同期を確保するために、すべての (マスタ/スレーブ) ADC のレギュラ変換は停止し、インジェクト変換の終了と同期して再開されます。

**注：** レギュラ同時 + オルタネートトリガモードの組み合わせでは、同じ長さのシーケンスを変換するか、トリガ間隔が2つ (デュアルADCモード) / 3つ (トリプルADCモード) のシーケンスのうち最も長い変換時間より長くなるようにしなければなりません。さもないと、長いシーケンスのADCが前の変換を完了する前に、短いシーケンスのADCがリスタートすることがあります。

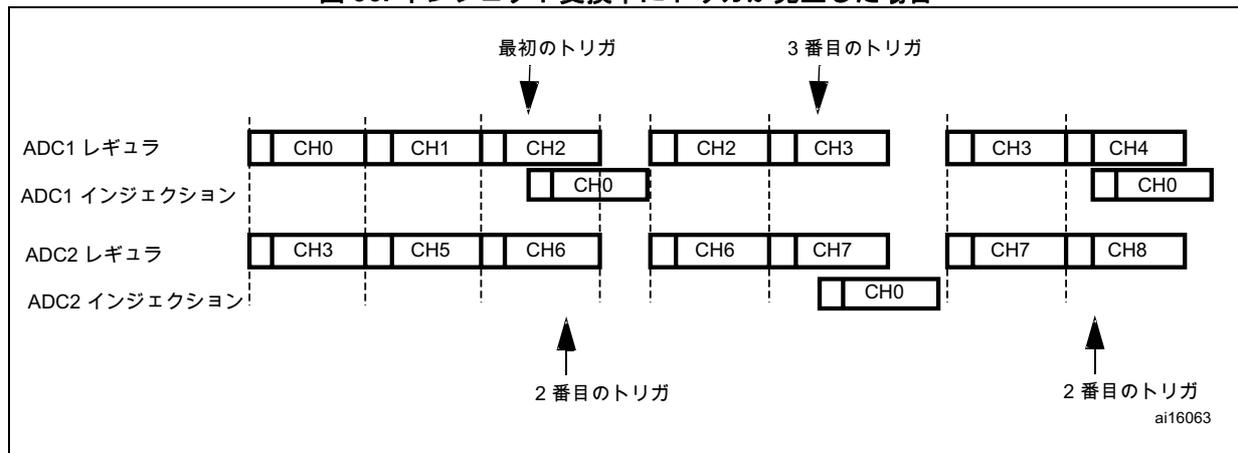
変換シーケンスが中断されると (たとえば、DMA 転送終了が発生)、インタリーブモードを再プログラムする前にまずマルチADCシーケンサを独立モードに設定することによって (ビット DUAL[4:0]=00000) リセットする必要があります。

図 89. オルタネート + レギュラ同時



レギュラ変換に割り込んだインジェクト変換の最中にトリガが発生した場合は無視されます。図 90は、この場合の動作を示します (2番目のトリガは無視されます)。

図 90. インジェクト変換中にトリガが発生した場合



## 15.10 温度センサ

温度センサを使用して、デバイスの温度 ( $T_A$ ) を測定できます。

- STM32F75xxx および STM32F74xxx デバイスでは温度センサは、内部で VBAT と同じ入力チャネル、ADC1\_IN18 に接続されています。ADC1\_IN18 はセンサの出力電圧または VBAT をデジタル値に変換するために使用されます。一度に選択できるのは、温度センサまたは VBAT のうち、どちらか 1 つだけです。温度センサと VBAT 変換を同時に設定すると、VBAT 変換のみ行われます。

図 91 に、温度センサのブロック図を示します。

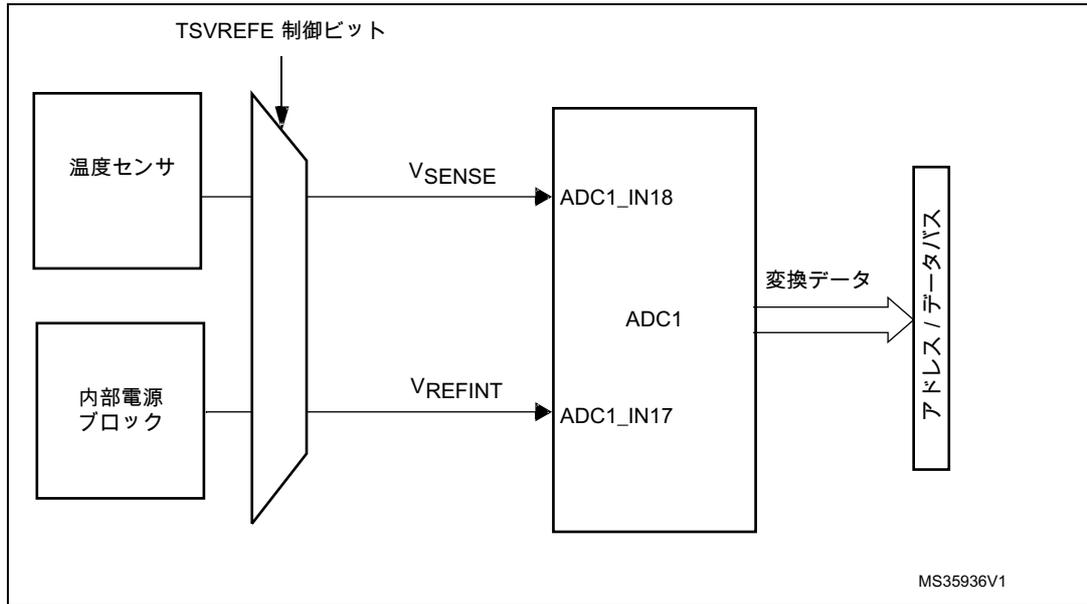
使用しないときには、センサをパワーダウンモードにすることができます。

**注：** 両方の内部チャネル、すなわち、ADC1\_IN18 (温度センサ) および ADC1\_IN17 (VREFINT) の変換を有効にするには、TSVREFE ビットをセットする必要があります。

### 主な特長

- サポートしている温度範囲 :  $-40 \sim 125 \text{ }^\circ\text{C}$
- 精度 :  $\pm 1.5 \text{ }^\circ\text{C}$

図 91. 温度センサおよび V<sub>REFINT</sub> チャンネルのブロック図



1. V<sub>SENSE</sub> が ADC1\_IN18 に入力されます。

## 温度の読み出し

温度センサを使用するには：

3. ADC1\_IN18 入力チャンネルを選択します。
4. データシートに指定されている最小サンプリング時間より長いサンプリング時間を選択します。
5. ADC\_CCR レジスタの TSVREFE ビットをセットして、温度センサをパワーダウンモードからウェイクアップします。
6. SWSTART ビットをセットすることによって（または外部トリガによって）ADC 変換を開始します。
7. ADC データレジスタの V<sub>SENSE</sub> データを読み出します。
8. 次の式を使用して、温度を計算します。

$$\text{温度 (摂氏)} = \{(V_{\text{SENSE}} - V_{25}) / \text{Avg\_Slope}\} + 25$$

ここで、

- V<sub>25</sub> = 25° C における V<sub>SENSE</sub> 値
- Avg\_Slope = 温度対 V<sub>SENSE</sub> 曲線の平均傾斜 (mV/°C または μV/°C)

V<sub>25</sub> および Avg\_Slope の実際の値については、データシートの電気特性のセクションを参照してください。

**注：** センサがパワーダウンモードからウェイクアップして、正しいレベルで V<sub>SENSE</sub> を出力できるようになるまでには時間がかかります (スタートアップ時間)。ADC にも起動後のスタートアップ時間があるので、遅延を最小にするには、ADON ビットと TSVREFE ビットを同時にセットしてください。

温度センサの出力電圧は、温度に比例して変化します。この一次関数のオフセットは、プロセスのばらつきにより、チップごとに異なります (チップ間で最大 45 °C)。

内部温度センサは、絶対温度の代わりに温度変化を検出するアプリケーションに適しています。正確な計測温度が必要な場合は、外部温度センサを使用してください。

## 15.11 バッテリ充電監視

ADC\_CCR レジスタの VBATE ビットはバッテリー電圧に切り替えるために使用します。V<sub>BAT</sub> 電圧は V<sub>DDA</sub> より高くなることがあるので、ADC の正しい動作を確保するために、V<sub>BAT</sub> ピンはブリッジ分圧回路に内部接続されています。

VBATE をセットするとブリッジが自動的に有効になり以下を接続します。

- VBAT/4 を ADC1\_IN18 入力チャンネル

**注：** VBAT および温度センサは同じ ADC 内部チャンネル (ADC1\_IN18) に接続されます。一度に 1 つの変換 (温度センサまたは VBAT のどちらか一方) を選択する必要があります。両方の変換を同時に有効すると、VBAT 変換のみ行われます。

## 15.12 ADC 割り込み

アナログウォッチドッグステータスビットがセットされ、かつオーバーランステータスビットがセットされていれば、レギュラおよびインジェクトグループの変換終了時、割り込みを生成することができます。高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

ADC\_SR レジスタには他に 2 つのフラグがありますが、これらは割り込みには無関係です。

- JSTRT (インジェクトグループチャンネルの変換開始)
- STRT (レギュラグループチャンネルの変換開始)

表 95. ADC 割り込み

割り込みイベント	イベントフラグ	イネーブル制御ビット
レギュラグループの変換終了	EOC	EOCIE
インジェクトグループの変換終了	JEOC	JEOCIE
アナログウォッチドッグステータスビットのセット	AWD	AWDIE
オーバーラン	OVR	OVRIE

## 15.13 ADC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワードレベル (32 ビット) で書き込む必要があります。読み出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位で行うことができます。

### 15.13.1 ADC ステータスレジスタ (ADC\_SR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD									
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

#### ビット 5 OVR : オーバーラン

このビットはデータが失われたとき (シングルモードまたはデュアル/トリプルモード時) ハードウェアによってセットされます。ソフトウェアによってクリアされます。オーバーラン検出は DMA = 1 または EOCS = 1 のときのみ有効です。

- 0 : オーバーランは発生していません。
- 1 : オーバーランが発生しました。

#### ビット 4 STRT : レギュラチャネル開始フラグ

このビットは、レギュラチャネルの変換が開始したときにハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : レギュラチャネルの変換は開始されていません。
- 1 : レギュラチャネルの変換が開始されました。

#### ビット 3 JSTRT : インジェクトチャネル開始フラグ

このビットは、インジェクトグループの変換が開始すると、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : インジェクトグループの変換は開始されていません。
- 1 : インジェクトグループの変換が開始されました。

**ビット 2 JEOC** : インジェクトチャネルの変換終了フラグ

このビットは、グループ内のすべてのインジェクトチャネル変換が終了したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : 変換が完了していません。
- 1 : 変換が完了しました。

**ビット 1 EOC** : レギュラチャネルの変換終了フラグ

このビットは、レギュラチャネルグループの変換が終了したときに、ハードウェアによってセットされます。ソフトウェアによって、または ADC\_DR レジスタを読み出すことによってクリアされます。

- 0 : 変換が終了していない (EOCS=0)、または変換のシーケンスが終了していません (EOCS=1)。
- 1 : 変換が終了した (EOCS=0)、または変換のシーケンスが終了しました (EOCS=1)。

**ビット 0 AWD** : アナログウォッチドッグフラグ

このビットは、変換された電圧が、ADC\_LTR および ADC\_HTR レジスタでプログラミングされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : アナログウォッチドッグイベントは発生していません。
- 1 : アナログウォッチドッグイベントが発生しました。

## 15.13.2 ADC 制御レジスタ 1 (ADC\_CR1)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	OVRIE	RES		AWDEN	JAWDEN	Res	Res	Res	Res	Res	Res
					r/w	r/w	r/w	r/w	r/w						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

**ビット 26 OVRIE** : オーバーラン割り込みイネーブル

このビットは、オーバーラン割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : オーバーラン割り込みは無効です。
- 1 : オーバーラン割り込みは有効です。OVR ビットがセットされると、割り込みが生成されます。

**ビット 25:24 RES[1:0]** : 分解能

これらのビットは、変換の分解能を選択するために、ソフトウェアによって書き込まれます。

- 00 : 12 ビット (15 ADCCLK サイクル)
- 01 : 10 ビット (13 ADCCLK サイクル)
- 10 : 8 ビット (11 ADCCLK サイクル)
- 11 : 6 ビット (9 ADCCLK サイクル)

**ビット 23 AWDEN** : アナログウォッチドッグ有効 (レギュラチャネル)

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : レギュラチャネルに対するアナログウォッチドッグは無効です。
- 1 : レギュラチャネルに対するアナログウォッチドッグは有効です。

**ビット 22 JAWDEN** : アナログウォッチドッグ有効 (インジェクトチャネル)

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : インジェクトチャネルに対するアナログウォッチドッグは無効です。
- 1 : インジェクトチャネルに対するアナログウォッチドッグは有効です。

ビット 21:16 予約済みであり、リセット値に保持する必要があります。

**ビット 15:13 DISCNUM[2:0] : 不連続モードチャンネルカウント**

これらのビットは、外部トリガの受信後に、不連続モードで変換されるレギュラチャンネルの数を定義するために、ソフトウェアによって書き込まれます。

000 : 1 チャンネル

001 : 2 チャンネル

...

111 : 8 チャンネル

**ビット 12 JDISCEN : インジェクトチャンネルの不連続モード**

このビットは、グループのインジェクトチャンネルに対する不連続モードを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : インジェクトチャンネルの不連続モードは無効です。

1 : インジェクトチャンネルの不連続モードは有効です。

**ビット 11 DISCEN : レギュラチャンネルの不連続モード**

このビットは、レギュラチャンネルに対する不連続モードを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : レギュラチャンネルの不連続モードは無効です。

1 : レギュラチャンネルの不連続モードは有効です。

**ビット 10 JAUTO : 自動インジェクトグループ変換**

このビットは、レギュラグループ変換の後、自動インジェクトグループ変換を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : 自動インジェクトグループ変換は無効です。

1 : 自動インジェクトグループ変換は有効です。

**ビット 9 AWDSGL : スキャンモードでの単一チャンネルのウォッチドッグイネーブル**

このビットは、AWDCH[4:0] ビットによって指定されたチャンネルに対するアナログウォッチドッグを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : すべてのチャンネルでアナログウォッチドッグは有効です。

1 : 単一チャンネルでアナログウォッチドッグは有効です。

**ビット 8 SCAN : スキャンモード**

このビットは、スキャンモードを有効/無効にするために、ソフトウェアによってセット/クリアされます。スキャンモードでは、ADC\_SQRx および ADC\_JSQRx レジスタを通じて選択された入力に変換されます。

0 : スキャンモードは無効です。

1 : スキャンモードは有効です。

**注 :** EOCIE ビットがセットされている場合、EOC 割り込みが生成されます。

- EOCs ビットが0にクリアされている場合、各レギュラグループシーケンス終了後

- EOCs ビットが1にセットされている場合、各レギュラチャンネル変換終了後

**注 :** JEOCIE ビットがセットされている場合、最後のチャンネルの変換が終了したときのみ、JEOC 割り込みが生成されます。

**ビット 7 JEOCIE : インジェクトチャンネルの割り込みイネーブル**

このビットは、インジェクトチャンネルの変換終了割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : JEOC 割り込みは無効です。

1 : JEOC 割り込みは有効です。JEOC ビットがセットされると、割り込みが生成されます。

**ビット 6 AWDIE : アナログウォッチドッグ割り込みイネーブル**

このビットは、アナログウォッチドッグ割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : アナログウォッチドッグ割り込みは無効です。

1 : アナログウォッチドッグ割り込みは有効です。

ビット 5 **EOCIE** : EOC 割り込みイネーブル

このビットは、変換終了時の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : EOC 割り込みは無効です。

1 : EOC 割り込みは有効です。EOC ビットがセットされると、割り込みが生成されます。

ビット 4:0 **AWDCH[4:0]** : アナログウォッチドッグチャンネル選択ビット

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグによって保護される入力チャンネルを選択します。

注: **00000** : ADC アナログ入力チャンネル0

**00001** : ADC アナログ入力チャンネル1

...

**01111** : ADC アナログ入力チャンネル15

**10000** : ADC アナログ入力チャンネル16

**10001** : ADC アナログ入力チャンネル17

**10010** : ADC アナログ入力チャンネル18

その他の値は予約済みです。

## 15.13.3 ADC 制御レジスタ 2 (ADC\_CR2)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	SWSTART	EXTEN			EXTSEL[3:0]				Res.	JSWSTART	JEXTEN			JEXTSEL[3:0]		
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	Res.	Res.	ALIGN	EOCS	DDS	DMA	Res.	Res.	Res.	Res.	Res.	Res.	CONT	ADON	
				r/w	r/w	r/w	r/w							r/w	r/w	

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **SWSTART** : レギュラチャンネルの変換開始

このビットは、変換を開始するためにソフトウェアによってセットされ、変換開始直後にハードウェアによってクリアされます。

0 : リセット状態

1 : レギュラチャンネルの変換を開始します。

注: **このビットは ADON = 1 のときのみセットできます。それ以外の場合変換は開始されません。**

ビット 29:28 **EXTEN** : レギュラチャンネルの外部トリガイネーブル

これらのビットは、外部トリガ極性を選択しレギュラグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00 : トリガ検出は無効です。

01 : 立ち上がりエッジでトリガを検出します。

10 : 立ち下がりエッジでトリガを検出します。

11 : 立ち上がりと下がりエッジの両方でトリガを検出します。

**ビット 27:24 EXTSEL[3:0] : レギュラグループの外部イベント選択**

これらのビットは、レギュラグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : タイマ 1 CC1 イベント  
0001 : タイマ 1 CC2 イベント  
0010 : タイマ 1 CC3 イベント  
0011 : タイマ 2 CC2 イベント  
0100 : タイマ 5 TRGO イベント  
0101 : タイマ 4 CC4 イベント  
0110 : タイマ 3 CC4 イベント  
0111 : タイマ 8 TRGO イベント  
1000 : タイマ 8 TRGO(2) イベント  
1001 : タイマ 1 TRGO イベント  
1010 : タイマ 1 TRGO(2) イベント  
1011 : タイマ 2 TRGO イベント  
1100 : タイマ 4 TRGO イベント  
1101 : タイマ 6 TRGO イベント  
1110 : 予約済み  
1111 : EXTI ライン 11

ビット 23 予約済みであり、リセット値に保持する必要があります。

**ビット 22 JSWSTART : インジェクトチャネルの変換開始**

このビットは、ソフトウェアによってセットされ、変換開始直後にハードウェアによってクリアされます。

0 : リセット状態  
1 : インジェクトチャネルの変換を開始します。

**注 :** このビットは  $ADON = 1$  のときのみセットできます。それ以外の場合変換は開始されません。

**ビット 21:20 JEXTEN : インジェクトチャネルの外部トリガイネーブル**

これらのビットは、外部トリガ極性を選択しインジェクトグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00 : トリガ検出は無効です。  
01 : 立ち上がりエッジでトリガを検出します。  
10 : 立ち下がりエッジでトリガを検出します。  
11 : 立ち上がりと下がりエッジの両方でトリガを検出します。

**ビット 19:16 JEXTSEL[3:0] : インジェクトグループの外部イベント選択**

これらのビットは、インジェクトグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : タイマ 1 TRGO イベント  
0001 : タイマ 1 CC4 イベント  
0010 : タイマ 2 TRGO イベント  
0011 : タイマ 2 CC1 イベント  
0100 : タイマ 3 CC4 イベント  
0101 : Timer4 TRGO イベント  
0110 : 予約済み  
0111 : タイマ 8 CC4 イベント  
1000 : タイマ 1 TRGO(2) イベント  
1001 : タイマ 8 TRGO イベント  
1010 : タイマ 8 TRGO(2) イベント  
1011 : タイマ 3 CC3 イベント  
1100 : タイマ 5 TRGO イベント  
1101 : タイマ 3 CC1 イベント  
1110 : タイマ 6 TRGO イベント  
1111 : 予約済み

**ビット 15:12 予約済みであり、リセット値に保持する必要があります。****ビット 11 ALIGN : データの配置**

このビットは、ソフトウェアによってセット/クリアされます。[図 76](#) および [図 77](#) を参照してください。

0 : 右詰め  
1 : 左詰め

**ビット 10 EOCS : 変換選択終了**

このビットは、ソフトウェアによってセット/クリアされます。

0 : EOC ビットはレギュラ変換シーケンスが終了するたびにセットされます。オーバーラン検出は DMA = 1 のときのみ有効です。  
1 : EOC ビットはレギュラ変換が終了するたびにセットされます。オーバーラン検出は有効です。

**ビット 9 DDS : DMA ディセーブル選択 (シングル ADC モード)**

このビットは、ソフトウェアによってセット/クリアされます。

0 : 最後の転送後新たな DMA リクエストは発行されません (DMA コントローラの設定どおり)  
1 : データが変換され DMA=1 である限り DMA リクエストは発行されます。

**ビット 8 DMA : ダイレクトメモリアクセスモード (シングル ADC モード)**

このビットは、ソフトウェアによってセット/クリアされます。詳細については、DMA コントローラの章を参照してください。

0 : DMA モードは無効です。  
1 : DMA モードは有効です。

**ビット 7:2 予約済みであり、リセット値に保持する必要があります。****ビット 1 CONT : 連続変換**

このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされた場合、それがクリアされるまで連続的に変換が行われます。

0 : シングル変換モード  
1 : 連続変換モード

**ビット 0 ADON : A/D コンバータのオン/オフ**

このビットは、ソフトウェアによってセット/クリアされます。

**注 :** 0 : ADC の変換を無効にして、パワーダウンモードに移行します。  
1 : ADC を有効にします。

## 15.13.4 ADC サンプル時間レジスタ 1 (ADC\_SMPR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 : 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:0 **SMPx[2:0]** : チャンネル x サンプリング時間選択

これらのビットは、各チャンネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプリングサイクル中は、チャンネル選択ビットを変更することはできません。

注 : **000 : 3 サイクル**  
**001 : 15 サイクル**  
**010 : 28 サイクル**  
**011 : 56 サイクル**  
**100 : 84 サイクル**  
**101 : 112 サイクル**  
**110 : 144 サイクル**  
**111 : 480 サイクル**

## 15.13.5 ADC サンプル時間レジスタ 2 (ADC\_SMPR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		rw	rw	rw	rw	rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:0 **SMPx[2:0]** : チャンネル x サンプリング時間選択

これらのビットは、各チャンネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプルサイクル中は、チャンネル選択ビットを変更することはできません。

注 : **000 : 3 サイクル**  
**001 : 15 サイクル**  
**010 : 28 サイクル**  
**011 : 56 サイクル**  
**100 : 84 サイクル**  
**101 : 112 サイクル**  
**110 : 144 サイクル**  
**111 : 480 サイクル**



## 15.13.6 ADC インジェクトチャネルデータオフセットレジスタ x (ADC\_JOFRx) (x=1..4)

アドレスオフセット : 0x14-0x20

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **JOFFSETx[11:0]** : インジェクトチャネル x のデータオフセット

これらのビットは、インジェクトチャネルを変換するときに変換データから引かれるオフセットを定義するために、ソフトウェアによって書き込まれます。変換結果は、ADC\_JDRx レジスタで読み出すことができます。

## 15.13.7 ADC ウォッチドッグ高閾値レジスタ (ADC\_HTR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0FFF

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **HT[11:0]** : アナログウォッチドッグ高閾値

これらのビットは、アナログウォッチドッグの高閾値を定義するために、ソフトウェアによって書き込まれます。

## 15.13.8 ADC ウォッチドッグ低閾値レジスタ (ADC\_LTR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **LT[11:0]** : アナログウォッチドッグ低閾値

これらのビットは、アナログウォッチドッグの低閾値を定義するために、ソフトウェアによって書き込まれます。

## 15.13.9 ADC レギュラシーケンスレジスタ 1 (ADC\_SQR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L[3:0]				SQ16[4:1]			
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16_0	SQ15[4:0]				SQ14[4:0]				SQ13[4:0]						
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **L[3:0]** : レギュラチャンネルシーケンス長

これらのビットは、レギュラチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

0000 : 1 変換

0001 : 2 変換

...

1111 : 16 変換

ビット 19:15 **SQ16[4:0]** : レギュラシーケンスの 16 番目の変換

これらのビットには、変換シーケンス内の 16 番目として割り当てられたチャンネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 14:10 **SQ15[4:0]** : レギュラシーケンスの 15 番目の変換

ビット 9:5 **SQ14[4:0]** : レギュラシーケンスの 14 番目の変換

ビット 4:0 **SQ13[4:0]** : レギュラシーケンスの 13 番目の変換

## 15.13.10 ADC レギュラシーケンスレジスタ 2 (ADC\_SQR2)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10_0	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:26 **SQ12[4:0]** : レギュラシーケンスの 12 番目の変換

これらのビットには、シーケンス内の 12 番目の変換として割り当てられたチャンネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 24:20 **SQ11[4:0]** : レギュラシーケンスの 11 番目の変換

ビット 19:15 **SQ10[4:0]** : レギュラシーケンスの 10 番目の変換

ビット 14:10 **SQ9[4:0]** : レギュラシーケンスの 9 番目の変換

ビット 9:5 **SQ8[4:0]** : レギュラシーケンスの 8 番目の変換

ビット 4:0 **SQ7[4:0]** : レギュラシーケンスの 7 番目の変換

## 15.13.11 ADC レギュラシーケンスレジスタ 3 (ADC\_SQR3)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4_0	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:25 **SQ6[4:0]** : レギュラシーケンスの 6 番目の変換

これらのビットには、変換シーケンスで変換される 6 番目として割り当てられたチャンネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 24:20 **SQ5[4:0]** : レギュラシーケンスの 5 番目の変換

ビット 19:15 **SQ4[4:0]** : レギュラシーケンスの 4 番目の変換

ビット 14:10 **SQ3[4:0]** : レギュラシーケンスの 3 番目の変換

ビット 9:5 **SQ2[4:0]** : レギュラシーケンスの 2 番目の変換

ビット 4:0 **SQ1[4:0]** : レギュラシーケンスの 1 番目の変換

## 15.13.12 ADC インジェクトシーケンスレジスタ (ADC\_JSQR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JL[1:0]		JSQ4[4:1]				
										r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
JSQ4[0]	JSQ3[4:0]				JSQ2[4:0]				JSQ1[4:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **JL[1:0]** : インジェクトシーケンス長

これらのビットは、インジェクトチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

- 00 : 1 変換
- 01 : 2 変換
- 10 : 3 変換
- 11 : 4 変換

ビット 19:15 **JSQ4[4:0]** : インジェクトシーケンスの 4 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

これらのビットには、変換シーケンス内の 4 番目として割り当てられたチャンネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 14:10 **JSQ3[4:0]** : インジェクトシーケンスの 3 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

ビット 9:5 **JSQ2[4:0]** : インジェクトシーケンスの 2 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

ビット 4:0 **JSQ1[4:0]** : インジェクトシーケンスの 1 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

- 注 :**
- JL[1:0]=3 のとき (シーケンサにおける 4 つのインジェクト変換) 、ADC はチャンネルを JSQ1[4:0]、JSQ2[4:0]、JSQ3[4:0]、JSQ4[4:0] の順に変換します。**
  - JL=2 のとき (シーケンサにおける 3 つのインジェクト変換) 、ADC はチャンネルを JSQ2[4:0]、JSQ3[4:0]、JSQ4[4:0] の順に変換します。**
  - JL=1 のとき (シーケンサにおける 2 つのインジェクト変換) 、ADC はチャンネルを JSQ3[4:0]、JSQ4[4:0] の順に変換します。**
  - JL=0 のとき (シーケンサにおける 1 つのインジェクト変換) 、ADC は JSQ4[4:0] チャンネルだけ変換します。**

## 15.13.13 ADC インジェクトデータレジスタ x (ADC\_JDRx) (x= 1..4)

アドレスオフセット : 0x3C - 0x48

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **JDATA[15:0]** : インジェクトデータ

これらのビットは読み出し専用です。これらのビットには、インジェクトチャンネル x の変換結果が格納されます。[図 76](#) および [図 77](#) に示すように、データは左詰めまたは右詰めされています。

## 15.13.14 ADC レギュラデータレジスタ (ADC\_DR)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **DATA[15:0]** : レギュラデータ

これらのビットは読み出し専用です。これらのビットには、レギュラチャンネルの変換結果が格納されます。[図 76](#) および [図 77](#) に示すように、データは左詰めまたは右詰めされています。

## 15.13.15 ADC 共通ステータスレジスタ (ADC\_CSR)

アドレスオフセット : 0x00 (このオフセットアドレスは ADC1 ベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

このレジスタにより、別の ADC のステータスビットの写像を得ることができます。ただし、これは読み出し専用で、別の ADC のステータスビットをクリアすることはできません。その代わりに、対応する ADC\_SR レジスタに 0 を書き込むことによって各ステータスビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR3	STRT3	JSTRT3	JEOC3	EOC3	AWD3
										ADC3					
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OVR2	STRT2	JSTRT2	JEOC2	EOC2	AWD2	Res.	Res.	OVR1	STRT1	JSTRT1	JEOC1	EOC1	AWD1
		ADC2								ADC1					
		r	r	r	r	r	r			r	r	r	r	r	r

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **OVR3** : ADC3 オーバーランフラグ

このビットは ADC3\_SR レジスタの OVR ビットのコピーです。

ビット 20 **STRT3** : ADC3 レギュラチャネル開始フラグ

このビットは ADC3\_SR レジスタの STRT ビットのコピーです。

ビット 19 **JSTRT3** : ADC3 インジェクトチャネル開始フラグ

このビットは ADC3\_SR レジスタの JSTRT ビットのコピーです。

ビット 18 **JEOC3** : ADC3 インジェクトチャネル変換終了フラグ

このビットは ADC3\_SR レジスタの JEOC ビットのコピーです。

ビット 17 **EOC3** : ADC3 変換終了フラグ

このビットは ADC3\_SR レジスタの EOC ビットのコピーです。

ビット 16 **AWD3** : ADC3 アナログウォッチドッグフラグ

このビットは ADC3\_SR レジスタの AWD ビットのコピーです。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **OVR2** : ADC2 オーバーランフラグ

このビットは ADC2\_SR レジスタの OVR ビットのコピーです。

ビット 12 **STRT2** : ADC2 レギュラチャネル開始フラグ

このビットは ADC2\_SR レジスタの STRT ビットのコピーです。

ビット 11 **JSTRT2** : ADC2 インジェクトチャネル開始フラグ

このビットは ADC2\_SR レジスタの JSTRT ビットのコピーです。

ビット 10 **JEOC2** : ADC2 インジェクトチャネル変換終了フラグ

このビットは ADC2\_SR レジスタの JEOC ビットのコピーです。

ビット 9 **EOC2** : ADC2 変換終了フラグ

このビットは ADC2\_SR レジスタの EOC ビットのコピーです。

ビット 8 **AWD2** : ADC2 アナログウォッチドッグフラグ

このビットは ADC2\_SR レジスタの AWD ビットのコピーです。



ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **OVR1** : ADC1 オーバーランフラグ  
このビットは ADC1\_SR レジスタの OVR ビットのコピーです。

ビット 4 **STRT1** : ADC1 レギュラチャネル開始フラグ  
このビットは ADC1\_SR レジスタの STRT ビットのコピーです。

ビット 3 **JSTRT1** : ADC1 インジェクトチャネル開始フラグ  
このビットは ADC1\_SR レジスタの JSTRT ビットのコピーです。

ビット 2 **JEOC1** : ADC1 インジェクトチャネル変換終了フラグ  
このビットは ADC1\_SR レジスタの JEOC ビットのコピーです。

ビット 1 **EOC1** : ADC1 変換終了フラグ  
このビットは ADC1\_SR レジスタの EOC ビットのコピーです。

ビット 0 **AWD1** : ADC1 アナログウォッチドッグフラグ  
このビットは ADC1\_SR レジスタの AWD ビットのコピーです。

## 15.13.16 ADC 共通制御レジスタ (ADC\_CCR)

アドレスオフセット : 0x04 (このオフセットアドレスは ADC1 ベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSVREFE	VBATE	Res.	Res.	Res.	Res.	ADCPRE	
								rw	rw					rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA[1:0]		DDS	Res.	DELAY[3:0]				Res.	Res.	Res.	MULTI[4:0]				
rw	rw	rw		rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TSVREFE** : 温度センサと V<sub>REFINT</sub> イネーブル  
このビットは、温度センサおよび V<sub>REFINT</sub> チャネルを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : 温度センサおよび V<sub>REFINT</sub> チャネルは無効です。  
1 : 温度センサおよび V<sub>REFINT</sub> チャネルは有効です。  
**注 :** **TSVREFE** がセットされているときは **VBATE** を無効にする必要があります。両方のビットがセットされている場合、**VBAT** 変換のみ行われます。

ビット 22 **VBATE** : V<sub>BAT</sub> イネーブル  
このビットは、V<sub>BAT</sub> チャネルを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : V<sub>BAT</sub> チャネルは無効です。  
1 : V<sub>BAT</sub> チャネルは有効です。

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

**ビット 17:16 ADCPRE** : ADC プリスケーラ

ADC へのクロックの周波数を選択するために、ソフトウェアによってセット/クリアされます。クロックはすべての ADC に共通です。

- 注:**
- 00** : PCLK2 は 2 分周されます。
  - 01** : PCLK2 は 4 分周されます。
  - 10** : PCLK2 は 6 分周されます。
  - 11** : PCLK2 は 8 分周されます。

**ビット 15:14 DMA** : マルチ ADC モードのためのダイレクトメモリアクセスモード

このビットフィールドは、ソフトウェアによってセット/クリアされます。詳細については、DMA コントローラのセクションを参照してください。

- 00** : DMA モードは無効です。  
**01** : DMA モード 1 は有効です (デュアル ADC/トリプル ADC、ハーフワード毎、ADC1, 2, 3 の順で)  
**10** : DMA モード 2 は有効です (デュアル ADC/トリプル ADC、ハーフワードをペアで、ADC2&1, 1&3, 3&1 の順で)  
**11** : DMA モード 3 は有効です (デュアル ADC/トリプル ADC、バイトをペアで、ADC2&1, 1&3, 3&1 の順で)

**ビット 13 DDS** : DMA ディセーブル選択 (マルチ ADC モード)

このビットは、ソフトウェアによってセット/クリアされます。

- 0** : 最後の転送後新たな DMA リクエストは発行されません (DMA コントローラの設定どおり) DMA ビットはハードウェアによってクリアされませんが、新たな DMA リクエストを生成する前に、ソフトウェアによってクリアし必要なモードに設定する必要があります。  
**1** : データが変換され DMA=01、10、または 11 である限り DMA リクエストは発行されます。

ビット 12 予約済みであり、リセット値に保持する必要があります。

**ビット 11:8 DELAY** : 2 つのサンプリングフェーズ間の遅延

ソフトウェアでセット/クリアされます。これらのビットはデュアルまたはトリプルインターリーブモードで使用します。

- 0000** :  $5 * T_{ADCCLK}$   
**0001** :  $6 * T_{ADCCLK}$   
**0010** :  $7 * T_{ADCCLK}$   
...  
**1111** :  $20 * T_{ADCCLK}$

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **MULTI[4:0]** : マルチ ADC モード選択

これらのビットは、動作モードを選択するために、ソフトウェアによって書き込まれます。

- すべての ADC が単独で動作します。

00000 : 独立モード

- 00001 から 01001 : デュアルモード、ADC1 と ADC2 が協調して動作し、ADC3 は独立して動作します

00001 : レギュラ同時 + インジェクト同時モードの組み合わせ

00010 : レギュラ同時 + オルタネートトリガモードの組み合わせ

00011 : 予約済み

00101 : インジェクト同時モードのみ

00110 : レギュラ同時モードのみ

00111 : 高速インタリーブモードのみ

01001 : オルタネートトリガモードのみ

- 10001 から 11001 : トリプルモード : ADC1、ADC2、ADC3 が協調して動作します

10001 : レギュラ同時 + インジェクト同時モードの組み合わせ

10010 : レギュラ同時 + オルタネートトリガモードの組み合わせ

10011 : 予約済み

10101 : インジェクト同時モードのみ

10110 : レギュラ同時モードのみ

10111 : 高速インタリーブモードのみ

11001 : オルタネートトリガモードのみ

他のすべての組み合わせは予約済みであり、プログラムすることはできません。

**注 :** マルチモードでは、チャンネル構成が変更されると強制終了し、同期が崩れることがあります。構成を変更する前には必ず、マルチ ADC モードを無効にすることが推奨されます。

## 15.13.17 デュアルおよびトリプルモード用 ADC 共通レギュラデータレジスタ (ADC\_CDR)

アドレスオフセット : 0x08 (このオフセットアドレスは ADC1 ベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA2[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **DATA2[15:0]** : レギュラ変換ペアの 2 番目のデータ項目

- デュアルモードでは、これらのビットには ADC2 のレギュラデータが格納されます。 [デュアル ADC モード](#) を参照してください。
- トリプルモードでは、これらのビットには ADC2、ADC1、および ADC3 のレギュラデータが交互に格納されます。 [トリプル ADC モード](#) を参照してください。

ビット 15:0 **DATA1[15:0]** : レギュラ変換ペアの 1 番目のデータ項目

- デュアルモードでは、これらのビットには ADC1 のレギュラデータが格納されます。 [デュアル ADC モード](#) を参照してください。
- トリプルモードでは、これらのビットには ADC1、ADC3、および ADC2 のレギュラデータが交互に格納されます。 [トリプル ADC モード](#) を参照してください。

## 15.13.18 ADC レジスタマップ

次の表は ADC レジスタの一覧です。

表 96. ADC グローバルレジスタマップ

オフセット	レジスタ
0x000 - 0x04C	ADC1
0x050 - 0x0FC	予約済み
0x100 - 0x14C	ADC2
0x118 - 0x1FC	予約済み
0x200 - 0x24C	ADC3
0x250 - 0x2FC	予約済み
0x300 - 0x308	共通レジスタ

表 97. ADC レジスタマップと各 ADC のリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	ADC_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		
	リセット値																													0	0	0	0	0	0
0x04	ADC_CR1	Res	Res	Res	Res	Res	OVRIE	RES[1:0]	AWDEN	JAWDEN											DISC NUM [2:0]	JDISCEN	DISCEN	JAUTO	AWD SGL	SCAN	JEOCIE	AWDIE	EOCIE						
	リセット値						0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	ADC_CR2	Res	SWSTART	EXTEN[1:0]						JSWSTART	JEXTEN[1:0]											ALIGN	EOCS	DDS	DMA										
	リセット値		0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	ADC_SMPR1	サンプル時間ビット SMPx_x																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	ADC_SMPR2	サンプル時間ビット SMPx_x																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	ADC_JOFR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x18	ADC_JOFR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x1C	ADC_JOFR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x20	ADC_JOFR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x24	ADC_HTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x28	ADC_LTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x2C	ADC_SQR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x30	ADC_SQR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x34	ADC_SQR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x38	ADC_JSQR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x3C	ADC_JDR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x40	ADC_JDR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x44	ADC_JDR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x48	ADC_JDR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		
0x4C	ADC_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																		

表 98. ADC レジスタマップとリセット値 (共通 ADC レジスタ)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	ADC_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD	Res.	Res.	Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD	Res.	Res.	OVR	STRT	JSTRT	JEOC	EOC	AWD
	リセット値											0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0
0x04	ADC_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSVREFE	VBATE	Res.	Res.	Res.	Res.	ADCPRE[1:0]	DMA[1:0]	DDS	Res.	DELAY [3:0]			Res.	Res.	Res.	MULTI [4:0]							
	リセット値										0	0					0	0	0	0		0	0	0	0					0	0	0	0	0
0x08	ADC_CDR	レギュラ DATA2[15:0]															レギュラ DATA1[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 16 D/A コンバータ (DAC)

### 16.1 DAC の概要

DAC モジュールは、12 ビットの電圧出力デジタルアナログコンバータです。DAC は、8 または 12 ビットモードで設定でき、DMA コントローラと組み合わせて使用することもできます。12 ビットモードでは、データを左詰め右詰めのものにも配置できます。DAC には 2 つの出力チャンネルがあり、それぞれがコンバータを搭載しています。デュアル DAC チャンネルモードでは、変換は独立して行うか、両方のチャンネルが同期更新操作のためにグループ化されているときには同時に行うことができます。精度を高めるために、入力基準ピン  $V_{REF+}$  (ADC と共用) を使用することができます。

### 16.2 DAC の主な機能

- 2 つの DAC コンバータ：それぞれ 1 つの出力チャンネル
- 12 ビットモードでのデータの左詰めまたは右詰め
- 同期更新機能
- ノイズ波生成
- 三角波生成
- デュアル DAC チャンネルの独立または同時変換
- 各チャンネルでの DMA 利用
- DMA アンダーランエラー検出
- 変換外部トリガ
- 入力基準電圧、 $V_{REF+}$

[図 92](#) は DAC チャンネルのブロック図を、[表 99](#) はピンの概要を示します。

図 92. DAC チャンネルブロック図

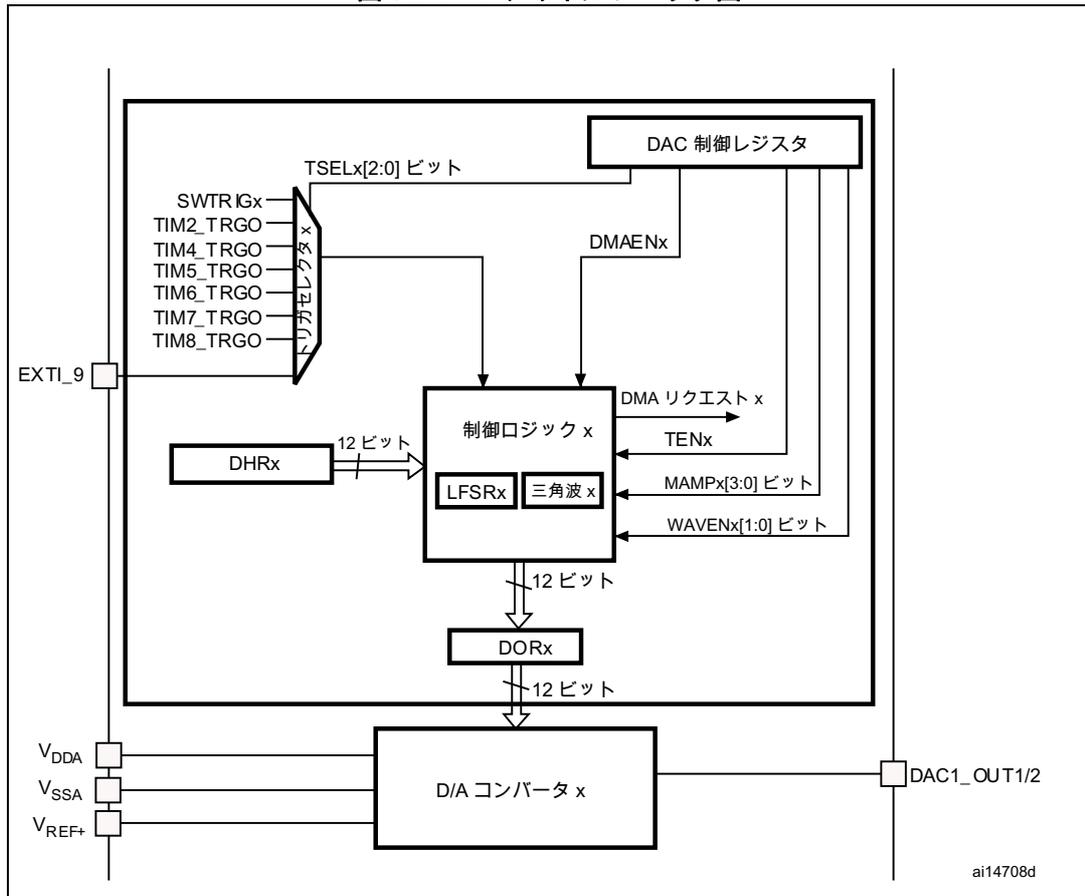


表 99. DAC ピン

名前	信号タイプ	説明
V <sub>REF+</sub>	入力、アナログ基準電圧正	DAC のハイレベル/正基準電圧 1.8 V ≤ V <sub>REF+</sub> ≤ V <sub>DDA</sub>
V <sub>DDA</sub>	入力、アナログ電源供給	アナログ電源供給
V <sub>SSA</sub>	入力、アナログ供給グラウンド	アナログ電源供給のグラウンド
DAC_OUTx	アナログ出力信号	DAC チャンネル x アナログ出力

**注：** DAC チャンネルx が有効になると、対応する GPIO ピン (PA4 または PA5) が自動的にアナログコンバータ出力 (DAC\_OUTx) に接続されます。寄生消費を防ぐために、PA4 または PA5 ピンはまずアナログ (AIN) として設定してください。

## 16.3 DAC の機能詳細

### 16.3.1 DAC チャンネルイネーブル

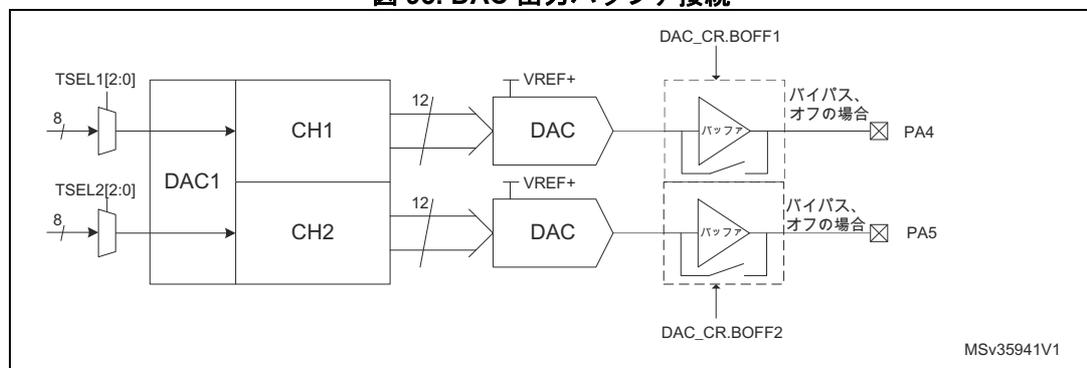
各 DAC チャンネルは、DAC\_CR レジスタの対応する ENx ビットをセットすることによって起動できます。DAC チャンネルは、スタートアップ時間  $t_{WAKEUP}$  後に有効になります。

**注：** ENx ビットは、アナログ DAC チャンネル x のマクロセルのみを有効にします。DAC チャンネル x デジタルインタフェースは、ENx ビットがリセットされた場合でも有効になります。

### 16.3.2 DAC 出力バッファイネーブル

DAC は、出力インピーダンスを減らし、外部動作アンプがなくても外部負荷を駆動できる 2 つの出力バッファを内蔵しています。各 DAC チャンネル出力バッファは、DAC\_CR レジスタの対応する BOFFx ビットを使用して有効/無効にできます。

図 93. DAC 出力バッファ接続



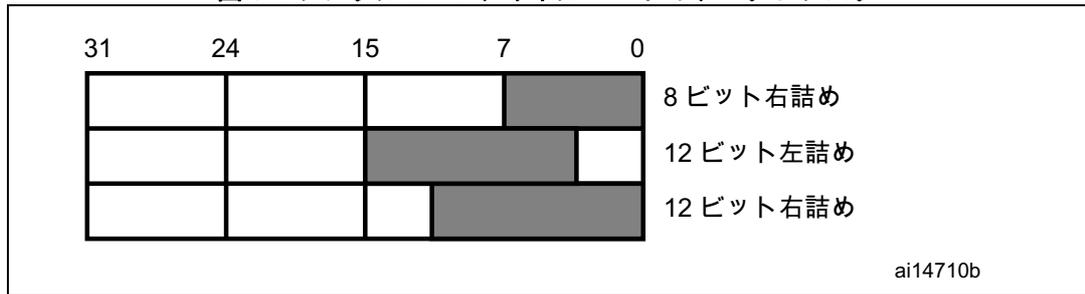
### 16.3.3 DAC データフォーマット

以下に示すように、選択された設定モードに応じて、指定されたレジスタにデータを書き込む必要があります。

- シングル DAC チャンネル x では、3 つの設定が可能です。
  - 8 ビット右詰め：データは、DAC\_DHR8Rx [7:0] ビット (DHRx [11:4] ビットに格納) にソフトウェアによってロードされること。
  - 12 ビット左詰め：データは、DAC\_DHR12Lx [15:4] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。
  - 12 ビット右詰め：データは、DAC\_DHR12Rx [11:0] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。

ユーザによって書き込まれたデータは、ロードされた DAC\_DHRyyyx レジスタに応じて、シフトされてから、DHRx (メモリマップされない内部レジスタであるデータ保持レジスタ x) に格納されます。その後、DHRx レジスタは自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、DORx レジスタにロードされます。

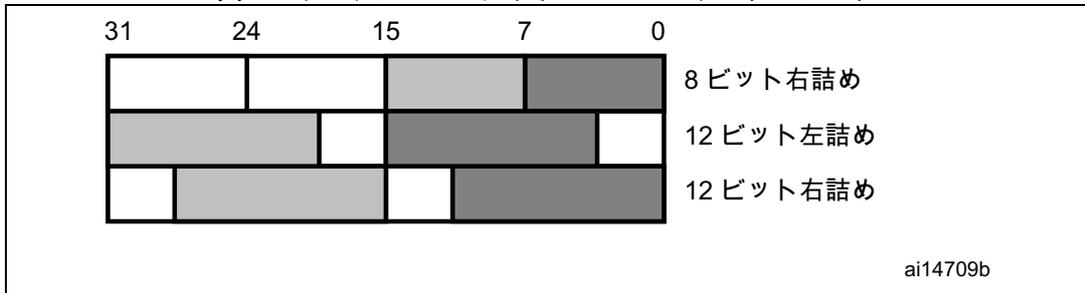
図 94. シングル DAC チャンネルモードのデータレジスタ



- デュアル DAC チャンネルでは、3 つの設定が可能です。
  - 8 ビット右詰め: DAC チャンネル 1 のデータは、DAC\_DHR8RD [7:0] ビット (DHR1[11:4] ビットに格納) に、DAC チャンネル 2 のデータは、DAC\_DHR8RD [15:8] ビット (DHR2[11:4] ビットに格納) にロードされます。
  - 12 ビット左詰め: DAC チャンネル 1 のデータは、DAC\_DHR12LD [15:4] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC\_DHR12LD [31:20] ビット (DHR2[11:0] ビットに格納) にロードされます。
  - 12 ビット右詰め: DAC チャンネル 1 のデータは、DAC\_DHR12RD [11:0] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC\_DHR12LD [27:16] ビット (DHR2[11:0] ビットに格納) にロードされます。

ロードされた DAC\_DHRyyyD レジスタに応じて、ユーザによって書き込まれたデータは、シフトされてから、DHR1 および DHR2 (Data Holding Register、メモリマップされない内部レジスタ) に格納されます。その後、DHR1 および DHR2 レジスタは、自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、それぞれ DOR1 および DOR2 レジスタにロードされます。

図 95. デュアル DAC チャンネルモードのデータレジスタ



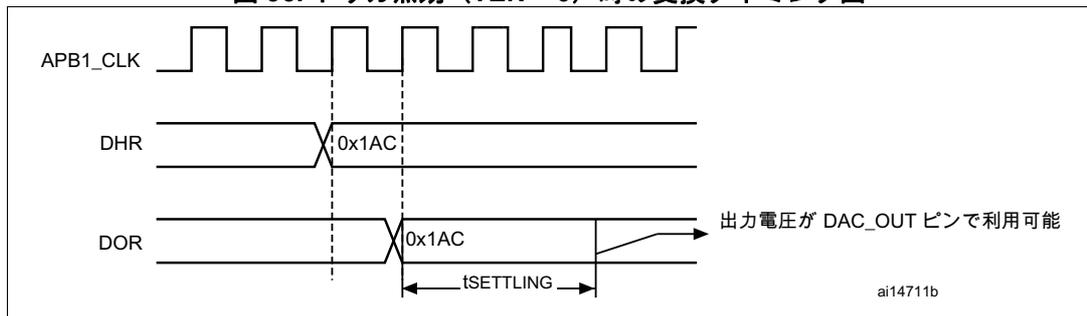
## 16.3.4 DAC 変換

DAC\_DORx に直接書き込むことはできず、DAC\_DHRx レジスタをロードすることによって (DAC\_DHR8Rx、DAC\_DHR12Lx、DAC\_DHR12Rx、DAC\_DHR8RD、DAC\_DHR12LD、または DAC\_DHR12RD への書き込み)、DAC チャンネル x へのデータ転送を行う必要があります。

DAC\_DHRx レジスタに格納されたデータは、ハードウェアトリガが選択されていない (DAC\_CR レジスタの TENx ビットがリセットされている) 場合に、1 APB1 クロックサイクル後に DAC\_DORx レジスタに自動的に転送されます。ただし、ハードウェアトリガが選択されている (DAC\_CR レジスタの TENx ビットがセットされている) ときには、トリガが発生すると、転送は 3 APB1 クロックサイクル後に行われます。

DAC\_DORx に DAC\_DHRx の内容がロードされると、電源電圧とアナログ出力負荷に応じて決定される t<sub>SETTLING</sub> 時間後にアナログ出力電圧が使用可能になります。

図 96. トリガ無効 (TEN = 0) 時の変換タイミング図



### 16.3.5 DAC 出力電圧

デジタル入力は、0 から  $V_{REF+}$  までのリニア変換で出力電圧に変換されます。

各 DAC チャンネルピンのアナログ出力電圧は、次の式によって求められます。

$$DAC_{output} = V_{REF} \times \frac{DOR}{4096}$$

### 16.3.6 DAC トリガ選択

TENx 制御ビットがセットされている場合、外部イベント (タイマカウンタ、外部割り込みラインなど) によって変換をトリガできます。表 100 に示すように、8 つの考えられるイベントのうち、どのイベントが変換をトリガするかは TSELx[2:0] 制御ビットによって決まります。

表 100. 外部トリガ

転送元	タイプ	TSEL[2:0]
タイマ 6 TRGO イベント	オンチップタイマからの内部信号	000
タイマ 8 TRGO イベント		001
タイマ 7 TRGO イベント		010
タイマ 5 TRGO イベント		011
タイマ 2 TRGO イベント		100
タイマ 4 TRGO イベント		101
EXTI ライン 9	外部ピン	110
SWTRIG	ソフトウェア制御ビット	111

DAC インタフェースが選択されたタイマ TRGO 出力または選択された外部割り込みライン 9 の立ち上がりエッジを検出するたびに、DAC\_DHRx レジスタに最後に格納されたデータが DAC\_DORx レジスタに転送されます。DAC\_DORx レジスタは、トリガが発生してから 3 APB1 サイクル後に更新されます。

ソフトウェアトリガが選択されている場合、変換は、SWTRIG ビットがセットされると開始されます。SWTRIG ビットは、DAC\_DHRx レジスタの内容が DAC\_DORx レジスタにロードされると、ハードウェアによってリセットされます。

**注：** ENx ビットがセットされているときには、TSELx[2:0] ビットを変更することはできません。

ソフトウェアトリガが選択されているときには、DAC\_DHRx レジスタから DAC\_DORx レジスタへの転送は、わずか 1 APB1 クロックサイクルで行われます。

## 16.3.7 DMA リクエスト

各 DAC チャンネルは、DMA 機能を備えています。DAC チャンネルの DMA リクエストは、2 つの DMA チャンネルを使用して処理されます。

DAC DMA リクエストは、DMAENx ビットがセットされているときに、外部トリガ（ソフトウェアトリガでなく）が発生したときに生成されます。その場合、DAC\_DHRx レジスタの値が DAC\_DORx レジスタに転送されます。

デュアルモードでは、両方の DMAENx ビットがセットされている場合、2 つの DMA リクエストが生成されます。1 つの DMA リクエストしか必要ない場合には、対応する DMAENx ビットだけをセットしてください。これにより、アプリケーションは 1 つの DMA リクエストと一意な DMA チャンネルを使用して、両方の DAC チャンネルをデュアルモードで管理することができます。

### DMA アンダーラン

DAC DMA リクエストはキューされないため、最初の外部トリガに対する確認応答が受信される（最初のリクエスト）前に 2 つ目の外部トリガが発生すると、新しいリクエストは発行されず、DAC\_SR レジスタの DMA チャンネルx アンダーランフラグ DMAUDRx がセットされてエラー条件を報告します。続いて DMA データ転送が無効になり、その後の DMA リクエストは処理されません。DAC チャンネルx は、古いデータを変換し続けます。

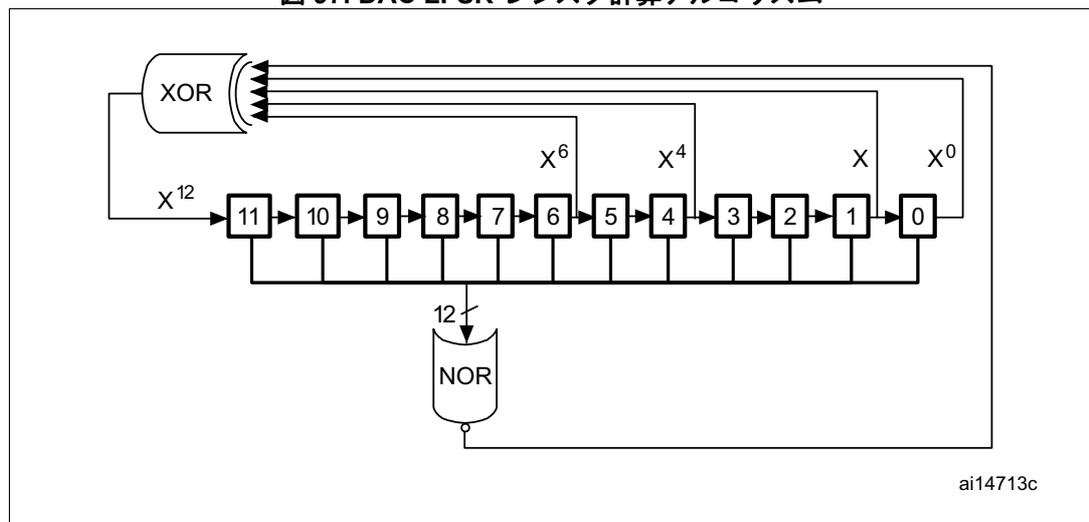
ソフトウェアでは、“1”を書き込むことによって DMAUDRx フラグをクリアし、使用された DMA ストリームの DMAEN ビットをクリアし、DMA と DAC のチャンネル x を再初期化して転送を正しくリスタートさせてください。また、DAC トリガ変換周波数を変更するか DMA の負荷を軽減して、新しい DMA アンダーランを回避してください。最後に、DMA データ転送と変換トリガを有効にすることによって DAC 変換を再開することができます。

各 DAC チャンネルx では、DAC\_CR レジスタの対応する DMAUDRIEx ビットが有効にされた場合、割り込みも生成されます。

## 16.3.8 ノイズ生成

リニアフィードバックシフトレジスタ (LFSR) を使用して、可変振幅の擬似ノイズを生成することができます。DAC ノイズ生成を選択するには、WAVEx[1:0] に“01”をセットします。LFSR にプリロードされる値は 0xAAA です。このレジスタは、各トリガイベントの 3 APB1 クロックサイクル後に、特定の計算アルゴリズムに従って更新されます。

図 97. DAC LFSR レジスタ計算アルゴリズム

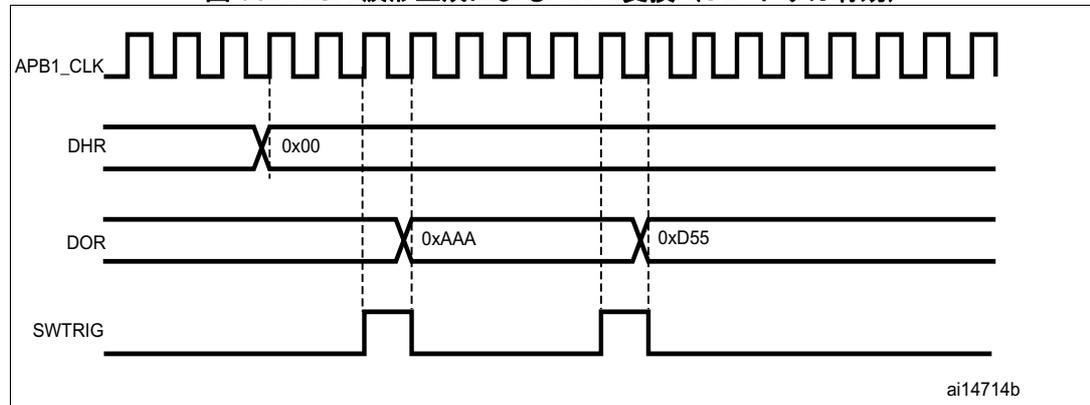


LFSR 値は、DAC\_CR レジスタの MAMPx[3:0] ビットによって部分的または全体的にマスクでき、オーバーフローなしに DAC\_DHRx の内容に加算され、DAC\_DORx レジスタに格納されます。

LFSR が 0x0000 の場合、“1”がインジェクトされます (アンチロックアップメカニズム)。

WAVEx[1:0] ビットをリセットすることによって、LFSR 波形生成をリセットできます。

図 98. LFSR 波形生成による DAC 変換 (SW トリガ有効)



注: ノイズ生成のためには、DAC\_CR レジスタの TENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

### 16.3.9 三角波生成

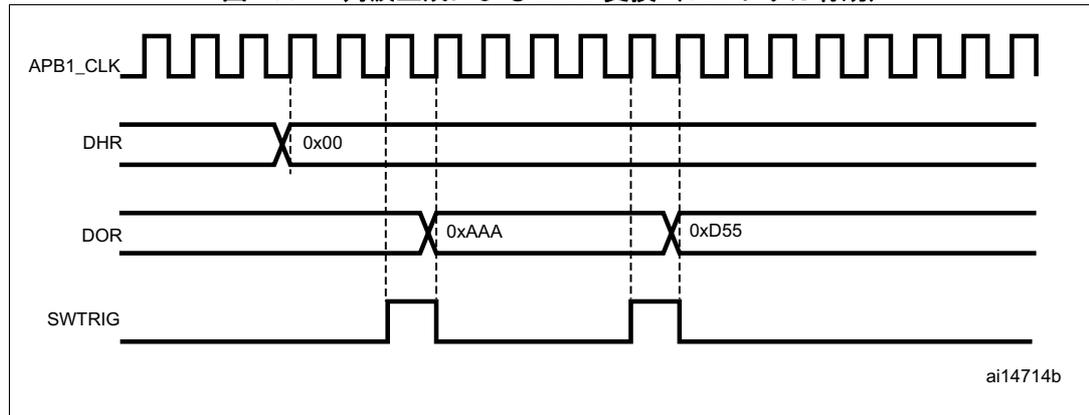
DC または低周波数信号上に、小さな振幅の三角波を追加することが可能です。DAC 三角波生成を選択するには、WAVEx[1:0] を“10”にセットします。振幅は、DAC\_CR レジスタの MAMPx[3:0] ビットを介して設定されます。内部三角波カウンタは、各トリガイベントの 3 APB1 クロックサイクル後にインクリメントされます。このカウンタの値は、オーバーフローなしに DAC\_DHRx レジスタに加えられ、合計は DAC\_DORx レジスタに格納されます。三角波カウンタは、MAMPx[3:0] ビットによって定義された最大振幅以上になるまでインクリメントされます。設定された振幅に達すると、カウンタは 0 にデクリメントされ、再びインクリメントが開始されます。

WAVEx[1:0] ビットをリセットすることによって、三角波生成をリセットすることができます。

図 99. DAC 三角波生成



図 100. 三角波生成による DAC 変換 (SW トリガ有効)



注： ノイズ生成のためには、DAC\_CR レジスタの TENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

DAC を有効にするには、その前に MAMPx[3:0] ビットを設定する必要があります。そうしないと、これらのビットは変更できません。

## 16.4 デュアル DAC チャネル変換

同時に 2 つの DAC チャネルを必要とするアプリケーションで、バスのバンド幅を有効に使用するために、DHR8RD、DHR12RD、および DHR12LD の 3 つのデュアルレジスタが搭載されています。両方の DAC チャネルを同時に駆動するには、一意なレジスタアクセスが必要です。

2 つの DAC チャネルとこれらのデュアルレジスタを使用することで、11 の変換モードが使用可能です。すべての変換モードは、必要な場合には、別の DHRx レジスタを使っても利用できます。

すべてのモードについて、以下の節で説明します。

### 16.4.1 波形生成なしの独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャネルデータをロードします。

DAC チャネル1 トリガが発生すると、DHR1 レジスタが DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。

DAC チャネル2 トリガが発生すると、DHR2 レジスタが DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。

## 16.4.2 1つのLFSR生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
- 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
- 目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、デュアル DAC チャンネルデータをロードします。

DAC チャンネル1 トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャンネル2 トリガが発生すると、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 16.4.3 異なるLFSR生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
- 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMP1[3:0] ビットと MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル1 トリガが発生すると、MAMP1[3:0] によって設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャンネル2 トリガが発生すると、MAMP2[3:0] によって設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 16.4.4 1つの三角波生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
- 2 つの DAC チャンネルの WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル1 トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

DAC チャンネル2 トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

## 16.4.5 異なる三角波生成による独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
- 2 つの DAC チャンネル WAVEx[1:0] ビットを“1x”に、異なる最大振幅値マスクを MAMP1[3:0] と MAMP2[3:0] ビットに設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル1 トリガが発生すると、MAMP1[3:0] によって設定された三角波振幅を持つ DAC チャンネル1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

DAC チャンネル2 トリガが発生すると、MAMP2[3:0] によって設定された三角波振幅を持つ DAC チャンネル2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

## 16.4.6 同時ソフトウェア開始

DAC をこの変換モードに設定するには、次の手順が必要です。

- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) に、デュアル DAC チャンネルデータをロードします。

この設定では、1 APB1 クロックサイクル後に、DHR1 および DHR2 レジスタが DAC\_DOR1 と DAC\_DOR2 にそれぞれ転送されます。

## 16.4.7 波形生成なしの同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) に、デュアル DAC チャンネルデータをロードします。

トリガが発生すると、DHR1 および DHR2 レジスタが DAC\_DOR1 と DAC\_DOR2 にそれぞれ転送されます (3 APB1 クロックサイクル後)。

## 16.4.8 1つのLFSR生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
- 目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、デュアル DAC チャンネルデータをロードします。

トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR1 カウンタが更新されず。同時に、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 16.4.9 異なるLFSR生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 2 つの DAC チャンネルの WAVEx[1:0] ビットを“01”に設定し、MAMP1[3:0] と MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、MAMP1[3:0] で設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

同時に、MAMP2[3:0] で設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 16.4.10 1つの三角波生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 2 つの DAC チャンネルの WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

同時に、同じ三角波振幅を持つ DAC チャンネル2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

## 16.4.11 異なる三角波生成による同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

- TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
- TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
- 2 つの DAC チャンネル WAVEx[1:0] ビットを“1x”に、異なる最大振幅値マスクを MAMP1[3:0] と MAMP2[3:0] ビットに設定します。
- 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、MAMP1[3:0] によって設定された三角波振幅を持つ DAC チャンネル1 の三角波カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル1 の三角波カウンタが更新されます。

同時に、MAMP2[3:0] で設定された三角波振幅を持つ DAC チャンネル2 の三角波カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB1 クロックサイクル後)。その後、DAC チャンネル2 の三角波カウンタが更新されます。

## 16.5 DAC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1 \(59 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

### 16.5.1 DAC 制御レジスタ (DAC\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2	EN2
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[2:0]			TEN1	BOFF1	EN1
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DMAUDRIE2** : DAC チャンネル2 DMA アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル2 DMA アンダーラン割り込みは無効です。

1 : DAC チャンネル2 DMA アンダーラン割り込みは有効です。

ビット 28 **DMAEN2** : DAC チャンネル2 DMA イネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル2 DMA モードは無効です。

1 : DAC チャンネル2 DMA モードは有効です。

ビット 27:24 **MAMP2[3:0]** : DAC チャンネル2 マスク/振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

0000 : LFSR/三角波振幅のアンマスクビット 0 は 1 に等しい。

0001 : LFSR/三角波振幅のアンマスクビット [1:0] は 3 に等しい。

0010 : LFSR/三角波振幅のアンマスクビット [2:0] は 7 に等しい。

0011 : LFSR/三角波振幅のアンマスクビット [3:0] は 15 に等しい。

0100 : LFSR/三角波振幅のアンマスクビット [4:0] は 31 に等しい。

0101 : LFSR/三角波振幅のアンマスクビット [5:0] は 63 に等しい。

0110 : LFSR/三角波振幅のアンマスクビット [6:0] は 127 に等しい。

0111 : LFSR/三角波振幅のアンマスクビット [7:0] は 255 に等しい。

1000 : LFSR/三角波振幅のアンマスクビット [8:0] は 511 に等しい。

1001 : LFSR/三角波振幅のアンマスクビット [9:0] は 1023 に等しい。

1010 : LFSR/三角波振幅のアンマスクビット [10:0] は 2047 に等しい。

≥ 1011 : LFSR/三角波振幅のアンマスクビット [11:0] は 4095 に等しい。

ビット 23:22 **WAVE2[1:0]** : DAC チャンネル2 ノイズ/三角波生成イネーブル

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : 波形生成は無効です。

01 : ノイズ波生成は有効です。

1x : 三角波生成は有効です。

**注 :** ビット **TEN2=1** (DAC チャンネル2 トリガ有効) の場合のみ使用されます。

**ビット 21:19 TSEL2[2:0]** : DAC チャンネル2 トリガ選択

これらのビットは、DAC チャンネル2 をトリガするために使用される外部イベントを選択します。

- 000 : タイマ 6 TRGO イベント
- 001 : タイマ 8 TRGO イベント
- 010 : タイマ 7 TRGO イベント
- 011 : タイマ 5 TRGO イベント
- 100 : タイマ 2 TRGO イベント
- 101 : タイマ 4 TRGO イベント
- 110 : 外部ライン 9
- 111 : ソフトウェアトリガ

**注:** ビット  $TEN2=1$  (DAC チャンネル2 トリガ有効) の場合のみ使用されます。

**ビット 18 TEN2** : DAC チャンネル2 トリガイネーブル

このビットは、DAC チャンネル2 トリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル2 トリガは無効であり、DAC\_DHRx レジスタに書き込まれたデータは、1 APB1 クロックサイクル後に DAC\_DOR2 レジスタに転送されます。
- 1 : DAC チャンネル2 トリガは有効であり、DAC\_DHRx レジスタから転送されたデータは、3 APB1 クロックサイクル後に DAC\_DOR2 レジスタに転送されます。

**注:** ソフトウェアトリガが選択されているときには、DAC\_DHRx から DAC\_DOR2 レジスタへの転送には、1 APB1 クロックサイクルしか必要ありません。

**ビット 17 BOFF2** : DAC チャンネル2 出力バッファディセーブル

このビットは、DAC チャンネル2 出力バッファを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル2 出力バッファは有効です。
- 1 : DAC チャンネル2 出力バッファは無効です。

**ビット 16 EN2** : DAC チャンネル2 イネーブル

このビットは、DAC チャンネル2 を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル2 は無効です。
- 1 : DAC チャンネル2 は有効です。

**ビット 15:14** 予約済みであり、リセット値に保持する必要があります。**ビット 13 DMAUDRIE1** : DAC チャンネル 1 DMA アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 1 DMA アンダーラン割り込みは無効です。
- 1 : DAC チャンネル 1 DMA アンダーラン割り込みは有効です。

**ビット 12 DMAEN1** : DAC チャンネル 1 DMA イネーブル

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 1 DMA モードは無効です。
- 1 : DAC チャンネル 1 DMA モードは有効です。

ビット 11:8 **MAMP1[3:0]** : DAC チャンネル 1 マスク/振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

- 0000 : LFSR/三角波振幅のアンマスクビット 0 は 1 に等しい。
- 0001 : LFSR/三角波振幅のアンマスクビット [1:0] は 3 に等しい。
- 0010 : LFSR/三角波振幅のアンマスクビット [2:0] は 7 に等しい。
- 0011 : LFSR/三角波振幅のアンマスクビット [3:0] は 15 に等しい。
- 0100 : LFSR/三角波振幅のアンマスクビット [4:0] は 31 に等しい。
- 0101 : LFSR/三角波振幅のアンマスクビット [5:0] は 63 に等しい。
- 0110 : LFSR/三角波振幅のアンマスクビット [6:0] は 127 に等しい。
- 0111 : LFSR/三角波振幅のアンマスクビット [7:0] は 255 に等しい。
- 1000 : LFSR/三角波振幅のアンマスクビット [8:0] は 511 に等しい。
- 1001 : LFSR/三角波振幅のアンマスクビット [9:0] は 1023 に等しい。
- 1010 : LFSR/三角波振幅のアンマスクビット [10:0] は 2047 に等しい。
- ≥ 1011 : LFSR/三角波振幅のアンマスクビット [11:0] は 4095 に等しい。

ビット 7:6 **WAVE1[1:0]** : DAC チャンネル 1 ノイズ/三角波生成イネーブル

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 波形生成は無効です。
- 01 : ノイズ波生成は有効です。
- 1x : 三角波生成は有効です。

**注 :** ビット  $TEN1 = 1$  (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

ビット 5:3 **TSEL1[2:0]** : DAC チャンネル 1 トリガ選択

これらのビットは、DAC チャンネル 1 をトリガするために使用される外部イベントを選択します。

- 000 : タイマ 6 TRGO イベント
- 001 : タイマ 8 TRGO イベント
- 010 : タイマ 7 TRGO イベント
- 011 : タイマ 5 TRGO イベント
- 100 : タイマ 2 TRGO イベント
- 101 : タイマ 4 TRGO イベント
- 110 : 外部ライン 9
- 111 : ソフトウェアトリガ

**注 :** ビット  $TEN1 = 1$  (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

**ビット 2 TEN1** : DAC チャンネル 1 トリガイネーブル

このビットは、DAC チャンネル 1 トリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 トリガは無効であり、DAC\_DHRx レジスタに書き込まれたデータは、1 APB1 クロックサイクル後に DAC\_DOR1 レジスタに転送されます。

1 : DAC チャンネル 1 トリガは有効であり、DAC\_DHRx レジスタから転送されたデータは、3 APB1 クロックサイクル後に DAC\_DOR1 レジスタに転送されます。

**注 :** ソフトウェアトリガが選択されているときには、DAC\_DHRx から DAC\_DOR1 レジスタへの転送には、1 APB1 クロックサイクルしか必要ありません。

**ビット 1 BOFF1** : DAC チャンネル 1 出力バッファディセーブル

このビットは、DAC チャンネル 1 出力バッファを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 出力バッファは有効です。

1 : DAC チャンネル 1 出力バッファは無効です。

**ビット 0 EN1** : DAC チャンネル 1 イネーブル

このビットは、DAC チャンネル 1 を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 は無効です。

1 : DAC チャンネル 1 は有効です。

## 16.5.2 DAC ソフトウェアトリガレジスタ (DAC\_SWTRIGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SWTRIG2	SWTRIG1													
														w	w

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 SWTRIG2** : DAC チャンネル 2 ソフトウェアトリガ

このビットは、ソフトウェアトリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : ソフトウェアトリガは無効です。

1 : ソフトウェアトリガは有効です。

**注 :** このビットは、DAC\_DHR2 レジスタの値が DAC\_DOR2 レジスタにロードされると、ハードウェアによってクリアされず (1 APB1 クロックサイクル後)。

**ビット 0 SWTRIG1** : DAC チャンネル 1 ソフトウェアトリガ

このビットは、ソフトウェアトリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : ソフトウェアトリガは無効です。

1 : ソフトウェアトリガは有効です。

**注 :** このビットは、DAC\_DHR1 レジスタの値が DAC\_DOR1 レジスタにロードされると、ハードウェアによってクリアされず (1 APB1 クロックサイクル後)。



## 16.5.3 DAC チャンネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC\_DHR12R1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.5.4 DAC チャンネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC\_DHR12L1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 16.5.5 DAC チャンネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC\_DHR8R1)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DACC1DHR[7:0]														
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 8 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.5.6 DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC\_DHR12R2)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DHR[11:0]** : DAC チャンネル 2 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 2 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.5.7 DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC\_DHR12L2)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 16.5.8 DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC\_DHR8R2)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DACC2DHR[7:0]														
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DACC2DHR[7:0]** : DAC チャンネル2 8ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.5.9 デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC\_DHR12RD)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.5.10 デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC\_DHR12LD)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w				

ビット 31:20 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 16.5.11 デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC\_DHR8RD)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DACC2DHR[7:0]** : DAC チャンネル 2 8ビット右詰めデータ

これらのビットは、DAC チャンネル 2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 8 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.5.12 DAC チャンネル 1 データ出力レジスタ (DAC\_DOR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DOR[11:0]** : DAC チャンネル 1 データ出力

これらのビットは読み出し専用であり、DAC チャンネル 1 のデータ出力を含みます。

## 16.5.13 DAC チャンネル 2 データ出力レジスタ (DAC\_DOR2)

アドレスオフセット : 0x30  
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DOR[11:0]** : DAC チャンネル2 データ出力

これらのビットは読み出し専用であり、DAC チャンネル2 のデータ出力を含みます。

## 16.5.14 DAC ステータスレジスタ (DAC\_SR)

アドレスオフセット : 0x34  
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDR2	Res.												
		rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDR1	Res.												
		rc_w1													

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DMAUDR2** : DAC チャンネル2 DMA アンダーランフラグ

このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。

0 : DAC チャンネル2 に DMA アンダーランエラー条件は発生しませんでした。

1 : DAC チャンネル2 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル2 変換を駆動しています)。

ビット 28:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **DMAUDR1** : DAC チャンネル 1 の DMA アンダーランフラグ

このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。

0 : DAC チャンネル 1 に DMA アンダーランエラー条件は発生しませんでした。

1 : DAC チャンネル1 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル 1 変換を駆動しています)。

ビット 12:0 予約済みであり、リセット値に保持する必要があります。



## 16.5.15 DAC レジスタマップ

表 101 に DAC レジスタの要約を示します。

表 101. DAC レジスタマップ

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	DAC_CR	Res.	Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[2:0]		TSEL2[2:0]		TEN2	BOFF2	EN2	Res.	Res.	Res.	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[2:0]		TSEL1[2:0]		TEN1	BOFF1	EN1			
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	DAC_SWTRIGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																															0	0		
0x08	DAC_DHR12R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]													
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	DAC_DHR12L1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]													
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0		
0x10	DAC_DHR8R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							Res.	Res.	Res.	Res.			
	リセット値																						0	0	0	0	0	0	0	0	0	0	0		
0x14	DAC_DHR12R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]													
	リセット値																						0	0	0	0	0	0	0	0	0	0	0		
0x18	DAC_DHR12L2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]													
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0		
0x1C	DAC_DHR8R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							Res.	Res.	Res.	Res.			
	リセット値																						0	0	0	0	0	0	0	0	0	0	0		
0x20	DAC_DHR12RD	Res.	Res.	Res.	Res.	DACC2DHR[11:0]											Res.	Res.	Res.	Res.	DACC1DHR[11:0]														
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0	
0x24	DAC_DHR12LD	DACC2DHR[11:0]											予約済み					DACC1DHR[11:0]											予約済み						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x28	DAC_DHR8RD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							DACC1DHR[7:0]						
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0		
0x2C	DAC_DOR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DOR[11:0]													
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0		
0x30	DAC_DOR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DOR[11:0]													
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0		
0x34	DAC_SR	Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値			0																		0													

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 17 デジタルカメラインタフェース (DCMI)

### 17.1 DCMI の概要

このデジタルカメラインタフェースは、外部 8、10、12、または 14 ビット CMOS カメラモジュールから高速データフローを受信できる同期型パラレルインタフェースです。YCbCr4:2:2/RGB565 プログレッシブビデオおよび圧縮データ (JPEG) など、さまざまなデータフォーマットをサポートします。

このインタフェースはモノクロカメラ、X24 および X5 カメラで使用し、サイズ変更などすべての前処理はカメラモジュール内で行われることを前提としています。

### 17.2 DCMI の主な機能

- 8、10、12、または 14 ビットのパラレルインタフェース
- 埋め込み/外部ラインおよびフレーム同期
- 連続モードまたはスナップショットモード
- クロップ機能
- 以下のデータフォーマットに対応：
  - 8/10/12/14 ビットのプログレッシブビデオ：モノクロまたは Raw ベイヤー
  - YCbCr 4:2:2 プログレッシブビデオ
  - RGB 565 プログレッシブビデオ
  - 圧縮データ：JPEG

### 17.3 DCMI ピン

表 102 に DCMI ピンを示します。

表 102. DCMI ピン

名前	信号タイプ
D[0:13]	データ入力
HSYNC	水平同期入力
VSYNC	垂直同期入力
PIXCLK	ピクセルクロック入力

### 17.4 DCMI クロック

デジタルカメラインタフェースでは、PIXCLK および HCLK の 2 つのクロックドメインを使用します。PIXCLK と HCLK が安定した後、PIXCLK で生成された信号は、HCLK の立ち上がりエッジでサンプリングされます。カメラからのデータが安定し、サンプリング可能状態であることを示すため、イネーブル信号が HCLK ドメインで生成されます。最大 PIXCLK 周期は、HCLK 2.5 周期分よりも長い必要があります。

## 17.5 DCMI 機能概要

デジタルカメラインタフェースは、高速（最高 54 Mbytes/秒）データフローを受信できる同期型パラレルインタフェースです。最大 14 本のデータライン（D13-D0）と 1 本のピクセルクロックライン（PIXCLK）で構成されています。ピクセルクロックの極性はプログラム可能であるため、データはピクセルクロックの立ち上がりエッジまたは立ち下がりエッジのどちらでもキャプチャすることができます。

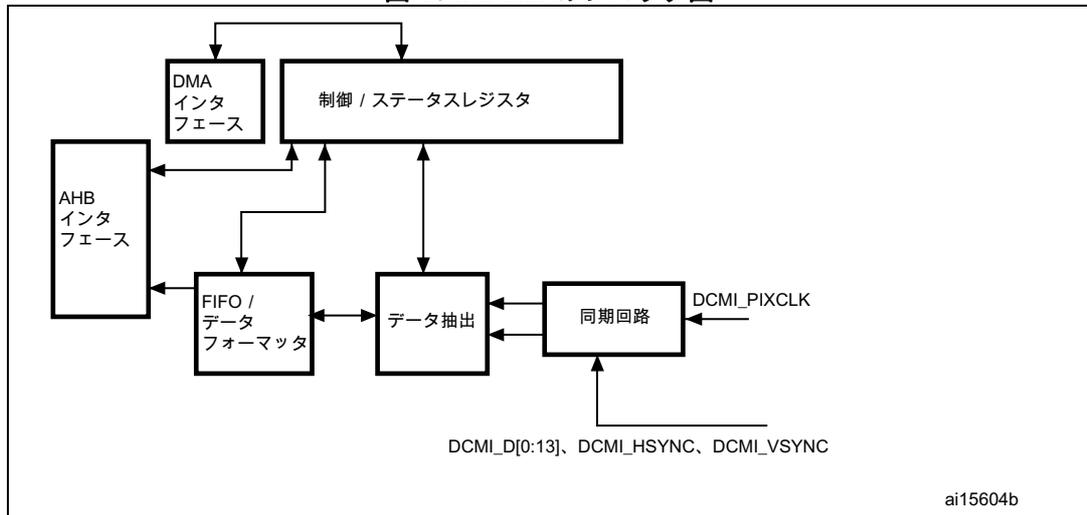
データは 32 ビットのデータレジスタ（DCMI\_DR）に格納され、汎用 DMA チャンネルを介して転送されます。画像バッファはカメラインタフェースではなく DMA が管理します。

カメラから受信したデータは、ライン/フレーム（Raw YUV/RGB/Bayer モード）で構成したり、一連の JPEG 画像にすることができます。JPEG 画像を受信できるようにするには、JPEG ビット（DCMI\_CR レジスタのビット 3）をセットする必要があります。

データフローは、オプションの HSYNC（水平同期）信号および VSYNC（垂直同期）信号を用いてハードウェアで同期をとることもできれば、データフローに埋め込まれた同期コードで同期をとることもできます。

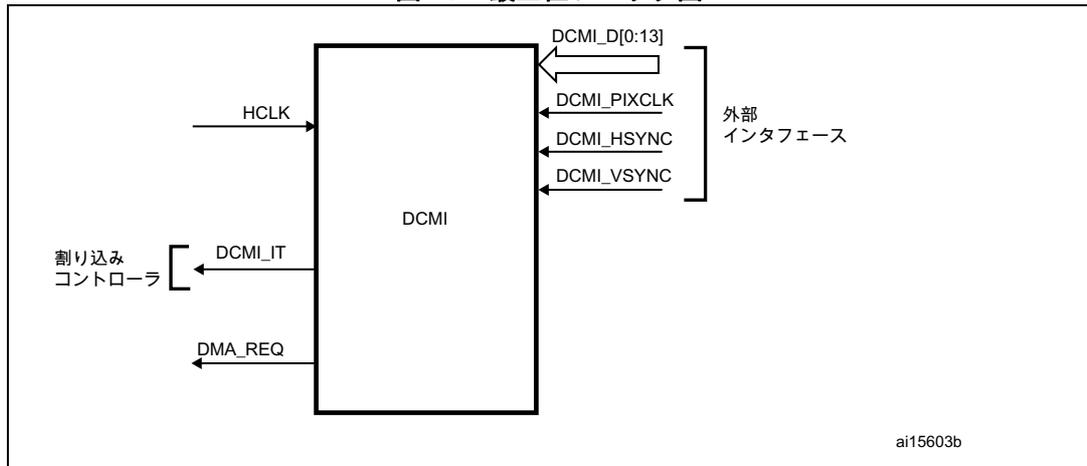
図 101 に DCMI ブロック図を示します。

図 101. DCMI のブロック図



ai15604b

図 102. 最上位ブロック図



## 17.5.1 DMA インタフェース

DCMI\_CR レジスタの CAPTURE ビットがセットされているときには、DMA インタフェースがアクティブです。カメラインタフェースがレジスタにまとまった 32 ビットデータブロックを受信するたびに、DMA リクエストが生成されます。

## 17.5.2 DCMI 物理インタフェース

インタフェースは、11/13/15/17 個の入力で構成されています。スレーブモードにのみ対応していません。

カメラインタフェースは、DCMI\_CR レジスタの EDM[1:0] ビットに従って、8、10、12、または 14 ビットのデータをキャプチャすることができます。使用されるのが 14 ビット未満である場合、使用されない入力ピンはグラウンドに接続する必要があります。

表 103. DCMI 信号

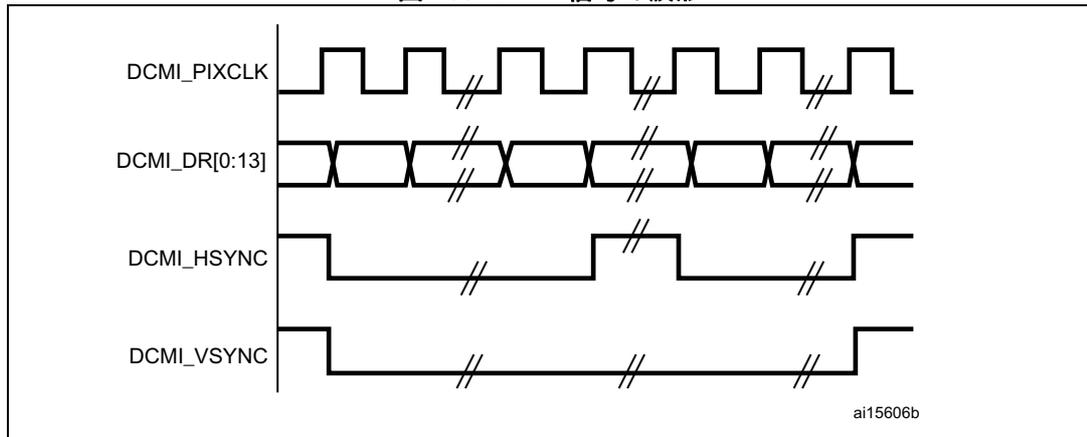
信号名		信号説明
8 ビット	D[0..7]	データ
10 ビット	D[0..9]	
12 ビット	D[0..11]	
14 ビット	D[0..13]	
PIXCLK		ピクセルクロック
HSYNC		水平同期/データ有効
VSYNC		垂直同期

データは PIXCLK と同期していますが、ピクセルクロックの立ち上がり/立ち下がりエッジにおけるデータは極性によって変わります。

HSYNC 信号は、ラインの開始/終了を示します。

VSYNC 信号は、フレームの開始/終了を示します。

図 103. DCMI 信号の波形



1. DCMI\_PIXCLK のキャプチャエッジは立ち下がりエッジであり、DCMI\_HSYNC と DCMI\_VSYNC のアクティブ状態は 1 です。
2. DCMI\_HSYNC と DCMI\_VSYNC の状態は、同時に変化することがあります。

## 8 ビットデータ

DCMI\_CR の EDM[1:0] が“00”にプログラムされていると、このインタフェースは入力 (D[0:7]) で LSB の 8 ビットをキャプチャし、それを 8 ビットデータとして記憶します。D[13:8] 入力は無視されます。この場合、32 ビットワードをキャプチャするには、カメラインタフェースでピクセルクロック 4 サイクルが必要です。

最初にキャプチャされたデータバイトは 32 ビットワードの LSB 位置に置かれ、4 番目にキャプチャされたデータバイトは 32 ビットワードの MSB 位置に置かれます。表 104 に、2 つの 32 ビットワードでキャプチャされたデータバイトの配置例を示します。

表 104. 32 ビットワード (8 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:24	23:16	15:8	7:0
0	D <sub>n+3</sub> [7:0]	D <sub>n+2</sub> [7:0]	D <sub>n+1</sub> [7:0]	D <sub>n</sub> [7:0]
4	D <sub>n+7</sub> [7:0]	D <sub>n+6</sub> [7:0]	D <sub>n+5</sub> [7:0]	D <sub>n+4</sub> [7:0]

## 10 ビットデータ

DCMI\_CR の EDM[1:0] が“01”にプログラムされていると、カメラインタフェースは入力 (D[0..9]) で 10 ビットデータをキャプチャし、それを 16 ビットワードの LSB の 10 ビットとして記憶します。DCMI\_DR レジスタの残りの MSB (ビット 11 から 15) はゼロにクリアされます。したがって、この場合、32 ビットデータワードはピクセルクロック 2 サイクル毎に完成することになります。

表 105 に示すように、最初にキャプチャされたデータは 32 ビットワードの LSB 位置に置かれ、2 番目にキャプチャされたデータは 32 ビットワードの上位 16 ビットの LSB 側に置かれます。

表 105. 32 ビットワード (10 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:26	25:16	15:10	9:0
0	0	D <sub>n+1</sub> [9:0]	0	D <sub>n</sub> [9:0]
4	0	D <sub>n+3</sub> [9:0]	0	D <sub>n+2</sub> [9:0]

## 12 ビットデータ

DCMI\_CR の EDM[1:0] が“10” にプログラムされていると、カメラインタフェースは入力 (D[0..11]) で 12 ビットデータをキャプチャし、それを 16 ビットワードの LSB の 12 ビットに記憶します。残りの MSB はゼロにクリアされます。したがって、この場合、32 ビットデータワードはピクセルクロック 2 サイクル毎に完成することになります。

表 106 に示すように、最初にキャプチャされたデータは 32 ビットワードの LSB 位置に置かれ、2 番目にキャプチャされたデータは 32 ビットワードの上位 16 ビットの LSB 側に置かれます。

表 106. 32 ビットワード (12 ビット幅) でキャプチャされたデータバイトの配置

バイトアドレス	31:28	27:16	15:12	11:0
0	0	D <sub>n+1</sub> [11:0]	0	D <sub>n</sub> [11:0]
4	0	D <sub>n+3</sub> [11:0]	0	D <sub>n+2</sub> [11:0]

## 14 ビットデータ

DCMI\_CR の EDM[1:0] が“11” にプログラムされていると、カメラインタフェースは入力 (D[0..13]) で 14 ビットデータをキャプチャし、それを 16 ビットワードの LSB の 14 ビットに記憶します。残りの MSB はゼロにクリアされます。したがって、この場合、32 ビットデータワードはピクセルクロック 2 サイクル毎に完成することになります。

表 107 に示すように、最初にキャプチャされたデータは 32 ビットワードの LSB 位置に置かれ、2 番目にキャプチャされたデータは 32 ビットワードの上位 16 ビットの LSB 側に置かれます。

表 107. 32 ビットワード (14 ビット幅) でキャプチャされたデータバイトの配置

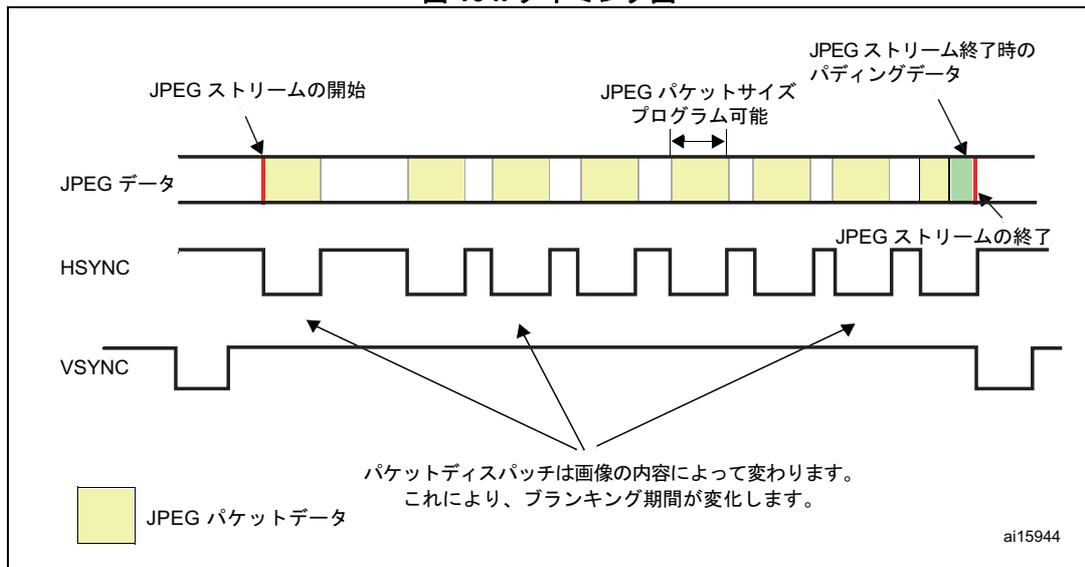
バイトアドレス	31:30	29:16	15:14	13:0
0	0	D <sub>n+1</sub> [13:0]	0	D <sub>n</sub> [13:0]
4	0	D <sub>n+3</sub> [13:0]	0	D <sub>n+2</sub> [13:0]

## 17.5.3 同期

この DCMI は、データ同期またはハードウェア (HSYNC および VSYNC) 同期に対応します。埋め込み同期を使用する場合、0x00 および 0xFF の値が (有効画像データではなく) 同期のためにのみ使用されているのを確認するのはデジタルカメラモジュールの仕様に依存します。埋め込み同期コードは、8 ビットパラレルデータインタフェース幅 (すなわち DCMI\_CR レジスタの EDM[1:0] ビットを“00”にクリア) に対してのみサポートされています。

圧縮データに対しては、DCMI はハードウェア同期モードにのみ対応します。この場合、VSYNC は画像の開始/終了として使用し、HSYNC はデータ有効信号として使用されます。図 104 に対応するタイミング図を示します。

図 104. タイミング図



## ハードウェア同期モード

ハードウェア同期モードでは、2つの同期信号 (HSYNC/VSYN) を使用します。

カメラモジュール/モードによっては、データが水平/垂直同期期間中に転送されることがあります。HSYNC/VSYN のアクティブ期間に受信されるデータはすべて無視されるため、これらの信号はブランキング信号のように振る舞います。

また、画像を正しく DMA/RAM バッファに転送するため、データ転送は VSYNC 信号と同期します。ハードウェア同期モードを選択し、キャプチャを有効 (DCMI\_CR の CAPTURE ビットをセットする) にすると、データ転送は VSYNC 信号の無効化 (フレームの次の開始) と同期します。

その後、連続したバッファまたは同じ/サーキュラバッファへ DMA によって転送された連続フレームにより、連続的な転送ができるようになります。連続したフレームの DMA 管理を可能にするため、各フレームの終了時には VSIF (垂直同期割り込みフラグ) がアクティブになります。

## データ同期モード

この同期モードでは、データフローに埋め込まれている 32 ビットコードを使用してデータフローを同期させます。このコードには、有効画像データでは使用されていない 0x00/0xFF 値を使用します。コードには 4 種類あり、いずれも 0xFF0000XY フォーマットを用います。埋め込み同期コードは、8 ビットパラレルデータ幅キャプチャ (DCMI\_CR レジスタの EDM[1:0] ビットを "00" にプログラム) においてのみサポートされています。他のデータ幅に対しては、このモードは予期しない動作となるため、使用しないでください。

注： カメラモジュールでは、このようなコードが8つ使用可能です (インタリーブモード)。このため、インタリーブモードはDCMIではサポートされません (ハーフフレームごとに捨てられてしまうため)。

- モード2：

4種類の同期コードにより、次のイベントが通知されます。

- フレーム開始 (FS)
- フレーム終了 (FE)
- ライン開始 (LS)
- ライン終了 (LE)

4種類のコードの0xFF0000XYフォーマットにおけるXY値は、プログラム可能です (セクション 17.8.7 : DCMI 埋め込み同期コードレジスタ (DCMI\_ESCR) を参照)。

「フレーム終了」としてプログラムされた値“0xFF”は、使用されていないすべてのコードが有効なフレーム終了コードとして解釈されることを意味します。

このモードでは、カメラインタフェースが有効になると、最初に発生したフレーム終了 (FE) コードの後にフレーム開始 (FS) コードが続くと、フレームのキャプチャを開始します。

- モード1：

もう一方のデータ同期モードとして、カメラモード1が使用できます。このモードはITU656対応です。

以下の同期コードにより次の各イベントが通知されます。

- SAV (アクティブライン) - ライン開始
- EAV (アクティブライン) - ライン終了
- SAV (ブランキング) - インタフレームブランキング期間中のライン終了
- EAV (ブランキング) - インタフレームブランキング期間中のライン終了

このモードは、次のコードをプログラムすることにより対応できます。

- FS ≤ 0xFF
- FE ≤ 0xFF
- LS ≤ SAV (アクティブ)
- LS ≤ EAV (アクティブ)

埋め込み型マスク解除コードもまた、フレーム/ラインの開始コードおよびラインの終了コード用に準備されています。これを使用すれば、プログラムされたコードに対して選択したマスク解除ビットのみを比較することができます。したがって、1つのビットを選択して埋め込みコード内で比較し、フレーム/ラインの開始や終了を検出することができます。つまり、マスク解除ビットの位置が同じでも、フレーム/ラインの開始や終了のコードは異なる可能性があるということです。

## 例

FS = 0xA5

FSのマスク解除コード = 0x10

この場合、フレーム開始コードがフレーム開始コードのビット4に埋め込まれています。

## 17.5.4 キャプチャモード

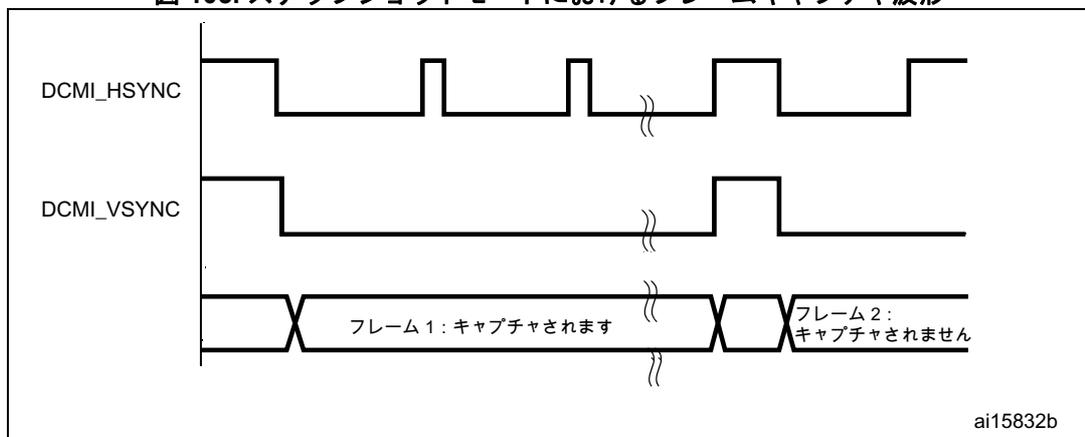
このインタフェースは、スナップショット（単一フレーム）、連続グラブの 2 種類のキャプチャをサポートしています。

### スナップショットモード（単一フレーム）

このモードでは、1つのフレームをキャプチャします (DCMI\_CR レジスタの CM = 1)。DCMI\_CR で CAPTURE ビットがセットされると、インタフェースはフレームの開始が検出されるのを待ってからデータをサンプリングします。最初の完全なフレームを受信した後、カメラインタフェースは自動的に無効になります (DCMI\_CR の CAPTURE ビットがクリアされます)。割り込みが有効になっている場合、割り込み (IT\_FRAME) が生成されます。

オーバーランの場合、フレームが失われ、CAPTURE ビットがクリアされます。

図 105. スナップショットモードにおけるフレームキャプチャ波形

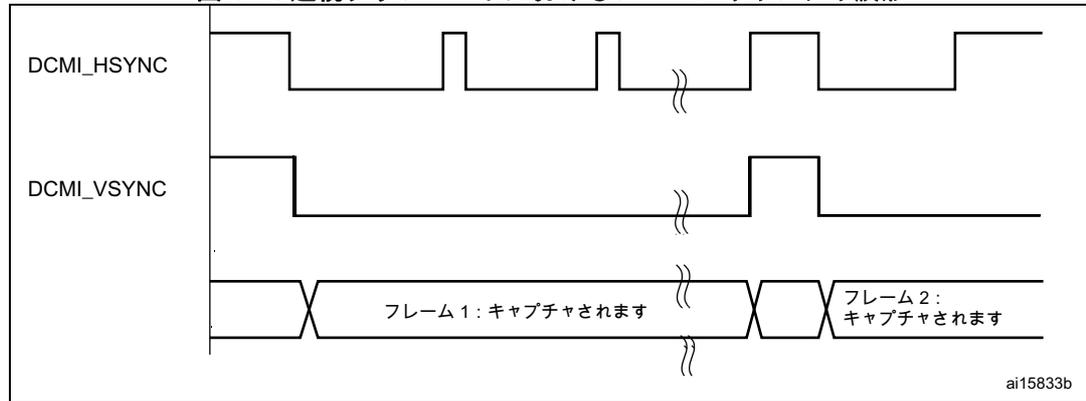


1. ここで、DCMI\_HSYNC と DCMI\_VSYNC のアクティブ状態は 1 です。
2. DCMI\_HSYNC と DCMI\_VSYNC の状態は、同時に変化することがあります。

## 連続グラブモード

このモード (DCMI\_CR の CM ビット = 0) では、DCMI\_CR の CAPTURE ビットがセットされると、モードによっては次の VSYNC または埋め込みフレーム開始でグラブ処理が開始されます。この処理は、DCMI\_CR の CAPTURE ビットがクリアされるまで続きます。いったん CAPTURE ビットがクリアされると、現在のフレームの終了までグラブ処理が続きます。

図 106. 連続グラブモードにおけるフレームキャプチャ波形



- ここで、DCMI\_HSYNC と DCMI\_VSYNC のアクティブ状態は 1 です。
- DCMI\_HSYNC と DCMI\_VSYNC の状態は、同時に変化することがあります。

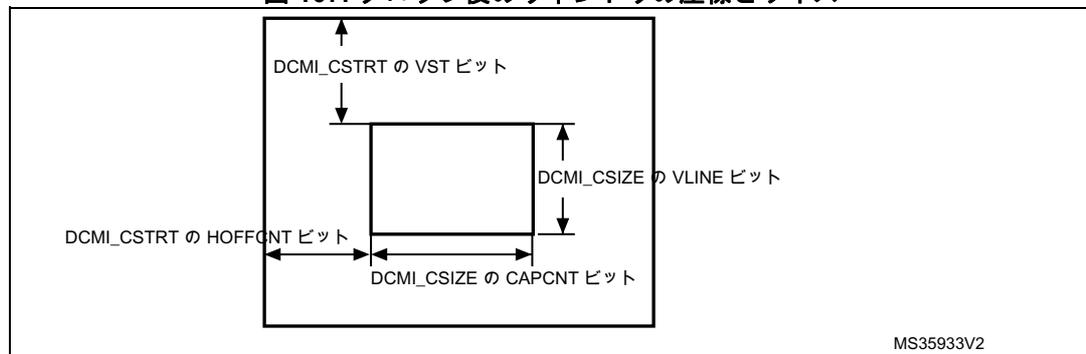
連続グラブモードでは、DCMI\_CR で FCRC ビットを設定し、すべての画像をグラブ、1 枚おきにグラブ、または 4 枚おきにグラブしてフレームキャプチャ速度を下げるすることができます。

**注：** ハードウェア同期モード (DCMI\_CR の ESS = 0) では、IT\_VSYNC 割り込みが有効になっていれば DCMI\_CR で CAPTURE = 0 になっていても IT\_VSYNC 割り込みが発生しますので、スナップショットモードと併せて IT\_VSYNC 割り込みを使用し、2 つのキャプチャ間でのフレーム数を数えることにより、さらにフレームキャプチャレートを下げることができます。これは埋め込みデータ同期モードでは行うことができません。

## 17.5.5 クロップ機能

クロップ機能を用いて、カメラインタフェースが受信した画像から長方形のウィンドウを選択することができます。2 つの 32 ビットレジスタ (DCMI\_CWSTRT および DCMI\_CWSIZE) を使用して開始 (左上の角) 座標とサイズ (水平方向はピクセルクロック数、垂直方向はライン数) を指定します。ウィンドウのサイズは、ピクセルクロック数 (水平方向) とライン数 (垂直方向) で指定します。

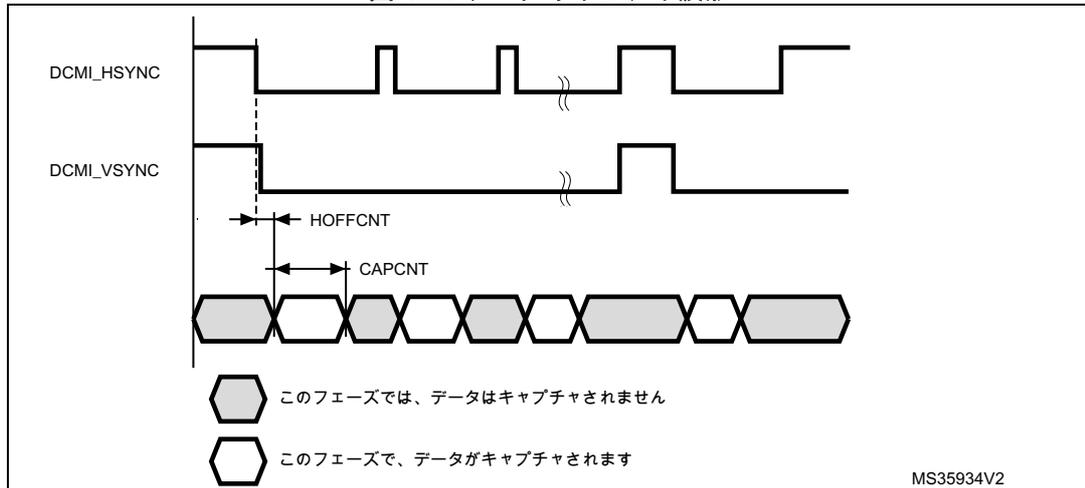
図 107. クロップ後のウィンドウの座標とサイズ



これらのレジスタを用いて、キャプチャウィンドウの開始点の座標をライン番号（フレーム内、0 から開始）とピクセルクロック数（ライン上、0 から開始）で指定し、ウィンドウのサイズをライン番号とピクセルクロック数で指定します。CAPCNT 値は、DMA を介してデータを正しく転送できるよう、必ず 4 の倍数となります（LSB の 2 ビットは強制的に 0）。

DCMI\_CWSIZE レジスタでライン番号が指定される前に VSYNC 信号がアクティブになると、キャプチャが停止し、IT\_FRAME 割り込みが有効になっていれば IT\_FRAME 割り込みが生成されます。

図 108. データキャプチャ波形



- ここで、DCMI\_HSYNC と DCMI\_VSYNC のアクティブ状態は 1 です。
- DCMI\_HSYNC と DCMI\_VSYNC の状態は、同時に変化することがあります。

## 17.5.6 JPEG フォーマット

JPEG 画像を受信できるようにするには、DCMI\_CR レジスタの JPEG ビットをセットする必要があります。JPEG 画像はラインやフレームとして保存されないため、HSYNC がデータイネーブル信号として使用されている間に、VSYNC 信号を使用してキャプチャを開始します。1 ライン内のバイト数は 4 の倍数でないこともあります。キャプチャされたデータから完全な 32 ビットワードが構成されるたびに DMA リクエストが生成されるため、この場合のデータ操作には注意が必要です。フレームの終了が検出され、転送される 32 ビットワードが完全に受信されていない場合、残りのデータは“0”でパディングされ、DMA リクエストが生成されます。

クロップ機能および埋め込み同期コードは、JPEG 形式では使用できません。

## 17.5.7 FIFO

### 入力モード

AHB 上でのデータ転送レートを管理するため、4 ワードの FIFO が搭載されています。この DCMI には、カメラインタフェースが AHB から読み出しを行うたびにインクリメントされる読み出しポインタと、FIFO へ書き込みを行うたびにインクリメントされる書き込みポインタを持った単純な FIFO コントローラを備えています。AHB インタフェースがデータ転送速度を維持できない場合にデータの上書きを防ぐオーバーラン保護はありません。

オーバーランや同期信号でのエラーの場合、FIFO はリセットされ、DCMI インタフェースは新規フレームの開始を待ちます。

## 17.6 データフォーマットの説明

### 17.6.1 データフォーマット

サポートされるデータは次の 3 種類です。

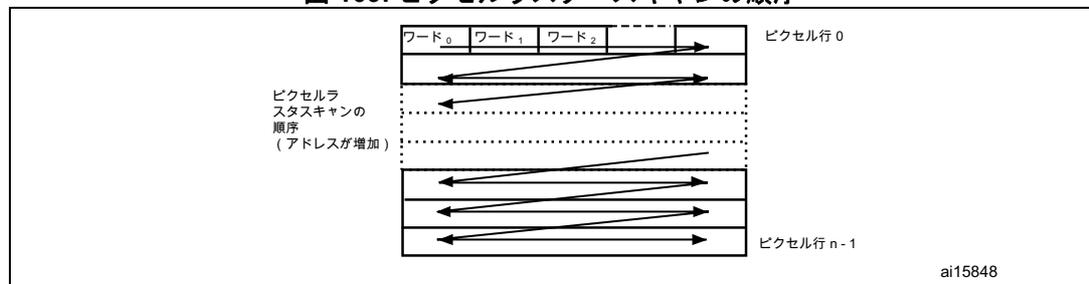
- 8 ビットのプログレッシブビデオ：モノクロまたは Raw ベイヤーフォーマット
- YCbCr 4:2:2 プログレッシブビデオ
- RGB565 プログレッシブビデオ。16 ビット（青が 5 ビット、赤が 5 ビット、緑が 6 ビット）で符号化されたピクセルの転送には、クロック 2 サイクルが必要です。

圧縮データ：JPEG

モノクロ、YCbCr、RGB データの場合、最大入力サイズは 2048 × 2048 ピクセルです。JPEG 圧縮モードでは制限はありません。

モノクロ、RGB、YCbCr の場合、フレームバッファはラスタモードで保存されます。32 ビットワードが使用されます。リトルエンディアン形式のみをサポートします。

図 109. ピクセルラスタースキャンの順序



### 17.6.2 モノクロフォーマット

特徴：

- ラスタフォーマット
- ピクセル当たり 8 ビット

表 108 に、データ保存方法を示します。

表 108. モノクロプログレッシブビデオフォーマットでのデータ保存

バイトアドレス	31:24	23:16	15:8	7:0
0	n + 3	n + 2	n + 1	n
4	n + 7	n + 6	n + 5	n + 4

## 17.6.3 RGB フォーマット

特徴：

- ラスタフォーマット
- RGB
- インタリーブ：バッファ 1 つ：R、G、B をインタリーブ：BRGBRGBRG、など。
- ディスプレイ出力に対して最適

RGB プレーナフォーマットは、標準 OS フレームバッファディスプレイフォーマットと互換性があります。

サポートされるのは 16 BPP (bits per pixel) : RGB565 (32 ビットワードあたり 2 ピクセル) だけです。24 BPP (パレットフォーマット) とグレースケールフォーマットは、サポートされません。ピクセルはラスタースキャン順、つまりピクセル行に対して上から下、ピクセル行内では左から右の順で保存されます。ピクセルコンポーネントは R (赤)、G (緑)、B (青) です。すべてのコンポーネントの空間分解能は同じ (4 : 4 : 4 フォーマット) です。ピクセル単位でインタリーブされたコンポーネントと共に、フレームは単一パートで保存されます。

表 109 に、データ保存方法を示します。

表 109. RGB プログレッシブビデオフォーマットでのデータ保存

バイトアドレス	31:27	26:21	20:16	15:11	10:5	4:0
0	赤 n+1	緑 n+1	青 n+1	赤 n	緑 n	青 n
4	赤 n+4	緑 n+3	青 n+3	赤 n+2	緑 n+2	青 n+2

## 17.6.4 YCbCr フォーマット

特徴：

- ラスタフォーマット
- YCbCr 4:2:2
- インタリーブ：バッファ 1 つ：Y、Cb、Cr をインタリーブ：CbYCrYCbYCr、など。

ピクセルコンポーネントは Y (輝度または「ルマ」)、Cb、Cr (クロミナンスまたは「色度」青および「色度」赤) です。各コンポーネントは 8 ビットで符号化されます。輝度と色度は、表 110 に示すように、(インタリーブして) 一緒に保存されます。

表 110. YCbCr プログレッシブビデオフォーマットでのデータ保存

バイトアドレス	31:24	23:16	15:8	7:0
0	Y n+1	Cr n	Y n	Cb n
4	Y n+3	Cr n+2	Y n+2	Cb n+2

## 17.6.5 YCbCr フォーマット - Y のみ

特徴：

- ラスタフォーマット
- YCbCr 4:2:2
- バッファには Y 情報のみが含まれる - モノクロ画像

ピクセルコンポーネントは Y (輝度または「ルマ」)、Cb、Cr (クロミナンスまたは「色度」青および「色度」赤) です。このモードでは、色度情報がドロップされます。8 ビットでエンコードされた各ピクセルのルマコンポーネントのみが表 111 に示すように保存されます。

結果として、元の YCbCr データと同じ分解能を持つモノクロ画像が得られます。

表 111. YCbCr プログレッシブビデオフォーマットでのデータ保存 - Y 抽出モード

バイトアドレス	31:24	23:16	15:8	7:0
0	Y <sub>n+3</sub>	Y <sub>n+2</sub>	Y <sub>n+1</sub>	Y <sub>n</sub>
4	Y <sub>n+7</sub>	Y <sub>n+6</sub>	Y <sub>n+5</sub>	Y <sub>n+4</sub>

## 17.6.6 ハーフ解像度画像抽出

以前の受信モードを変更したもので、モノクロ、RGB、または Y 抽出モードに適用可能です。

このモードでは、ハーフ解像度の画像のみを保存できます。OELS および LSM 制御ビットを通して選択されます。

## 17.7 DCMI 割り込み

5 種類の割り込みが生成されます。すべての割り込みは、ソフトウェアでマスク可能です。グローバル割り込み (IT\_DCMI) は、個別割り込みすべての論理和 (OR) です。表 112 にすべての割り込みのリストを示します。

表 112. DCMI 割り込み

割り込み名	割り込みイベント
IT_LINE	ライン終了を示します
IT_FRAME	フレームキャプチャ終了を示します
IT_OVR	データ受信のオーバーランを示します
IT_VSYNC	同期フレームを示します
IT_ERR	埋め込み同期フレーム検出におけるエラーの検出を示します
IT_DCMI	上記割り込みの論理和 (OR) です

## 17.8 DCMI レジスタの説明

すべての DCMI レジスタは、32 ビットワードとしてアクセスする必要があり、そうしないとバスエラーが発生します。

### 17.8.1 DCMI 制御レジスタ (DCMI\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OELS	LSM	OEBS	BSM		Res.	ENABLE	Res.	Res.	EDM		FCRC		VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE										
											rW	rW	rW	rW	rW		rW			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **OELS** : 奇数/偶数ライン選択 (ライン選択開始)

このビットは LSM フィールドと連携して動作します (LSM = 1)。

0 : インタフェースはフレーム開始後に最初のラインをキャプチャします (2 番目のラインはドロップされる)。

1 : インタフェースはフレーム開始から 2 番目のラインをキャプチャします (最初のラインはドロップされる)。

ビット 19 **LSM** : ライン選択モード

0 : インタフェースはすべての受信したラインをキャプチャします。

1 : インタフェースは 2 つのラインのうちの 1 つをキャプチャします。

ビット 18 **OEBS** : 奇数/偶数バイト選択 (バイト選択開始)

このビットは BSM フィールドと連携して動作します (BSM <> 00)。

0 : インタフェースはフレーム/ライン開始から最初のデータ (バイトまたはダブルバイト) をキャプチャします (2 番目のデータはドロップされる)。

1 : インタフェースはフレーム/ライン開始から 2 番目のデータ (バイトまたはダブルバイト) をキャプチャします (最初のデータはドロップされる)。

ビット 17:16 **BSM[1:0]** : バイト選択モード

00 : インタフェースはすべての受信したデータをキャプチャします。

01 : インタフェースは受信したデータから 1 つおきにバイトをキャプチャします。

10 : インタフェースは 4 つのバイトのうちの 1 つをキャプチャします。

11 : インタフェースは 4 つのバイトのうちの 2 つをキャプチャします。

**注 :** このモードは EDM[1:0]=00 のみ動作します。その他すべての EDM 値に対し、このビットフィールドをリセット値にプログラムする必要があります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **ENABLE** : DCMI イネーブル

0 : DCMI 無効

1 : DCMI 有効

**注 :** このビットを有効にする前に、DCMI 設定レジスタを正しくプログラムする必要があります。

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

**ビット 11:10 EDM[1:0] : 拡張データモード**

- 00 : インタフェースはピクセルクロックごとに 8 ビットデータをキャプチャします
- 01 : インタフェースはピクセルクロックごとに 10 ビットデータをキャプチャします
- 10 : インタフェースはピクセルクロックごとに 12 ビットデータをキャプチャします
- 11 : インタフェースはピクセルクロックごとに 14 ビットデータをキャプチャします

**ビット 9:8 FCRC[1:0] : フレームキャプチャレート制御**

- このビットによって、フレームキャプチャの頻度を規定します。これらのビットは、連続グラブモードでなければ意味を持ちません。スナップショットモードでは無視されます。
- 00 : 全フレームをキャプチャ
  - 01 : フレームを 1 つおきにキャプチャ (帯域幅 50 % 減少)
  - 10 : 4 フレームごとに 1 フレームキャプチャ (帯域幅 75 % 減少)
  - 11 : 予約済み

**ビット 7 VSPOL : 垂直同期極性**

- このビットは、データがパラレルインタフェース上で有効でないときの VSYNC ピン上でのレベルを示します。
- 0 : VSYNC アクティブロー
  - 1 : VSYNC アクティブハイ

**ビット 6 HSPOL : 水平同期極性**

- このビットは、データがパラレルインタフェース上で有効でないときの HSYNC ピン上でのレベルを示します。
- 0 : HSYNC アクティブロー
  - 1 : HSYNC アクティブハイ

**ビット 5 PCKPOL : ピクセルクロック極性**

- このビットによってピクセルクロックのキャプチャエッジを設定します。
- 0 : 立ち下がりエッジでアクティブ
  - 1 : 立ち上がりエッジでアクティブ

**ビット 4 ESS : 埋め込み同期選択**

- 0 : ハードウェア同期。データキャプチャ (フレーム / ラインの開始 / 終了) は HSYNC/VSYNC 信号と同期します。
- 1 : 埋め込み同期。データキャプチャはデータフローに埋め込まれた同期コードと同期します。

**注 : 8 ビットのパラレルデータに対してのみ有効です。ESS ビットがセットされると、HSPOL/VSPOL は無視されます。**

このビットは、JPEG モードでは無効になります。

**ビット 3 JPEG : JPEG フォーマット**

- 0 : 非圧縮ビデオフォーマット
- 1 : このビットは JPEG データ転送に使用します。HSYNC 信号はデータイネーブルとして使用します。このモードでは、クロップおよび埋め込み同期機能 (ESS ビット) は使用できません。

**ビット 2 CROP : クロップ機能**

- 0 : 画像全体がキャプチャされます。この場合、1 枚の画像フレーム内の総バイト数は 4 の倍数となります。
- 1 : クロップレジスタで指定されたウィンドウ内のデータのみがキャプチャされます。クロップウィンドウのサイズがピクチャサイズを超える場合は、そのピクチャサイズのみがキャプチャされます。

ビット 1 **CM** : キャプチャモード

0 : 連続グラブモード - 受信したデータは DMA を通じて転送先メモリに転送されます。バッファの場所やモード (リニアバッファまたはサーキュラバッファ) は、システム DMA を通じて制御されます。

1 : スナップショットモード (単一フレーム) - アクティブになると、インタフェースはフレーム開始を待ってから DMA を通じて単一フレームを転送します。フレーム終了時、CAPTURE ビットは、自動的にリセットされます。

ビット 0 **CAPTURE** : キャプチャイネーブル

0 : キャプチャは無効です。

1 : キャプチャは有効です。

カメラインタフェースは最初のフレーム開始を待ち、その後 DMA リクエストが生成されて受信したデータが転送先メモリに転送されます。

スナップショットモードでは、受信した最初のフレームの終了時に CAPTURE ビットが自動的にクリアされます。

連続グラブモードにおいて、キャプチャ進行中にソフトウェアがこのビットをクリアすると、そのビットは実質的にフレーム終了時にクリアされます。

**注 :** *DMA コントローラとすべての DCMI 設定レジスタを正しくプログラムしてから、このビットを有効にする必要があります。*

## 17.8.2 DCMI ステータスレジスタ (DCMI\_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0x0000

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **FNEF** : FIFO 非エンプティ

このビットは、FIFO のステータスを示します。

1 : FIFO に有効なデータあり

0 : FIFO エンプティ

ビット 1 **VSYNC**

このビットは、極性が正しくプログラムされた VSYNC ピンの状態を示します。

埋め込み同期コードを使用する場合、このビットは次のような意味を持ちます。

0 : アクティブフレーム

1 : フレーム間同期

埋め込み同期の場合、このビットは DCMI\_CR の CAPTURE ビットがセットされている場合にのみ意味を持ちます。

ビット 0 **HSYNC**

このビットは、極性が正しくプログラムされた HSYNC ピンの状態を示します。

埋め込み同期コードを使用する場合、このビットは次のような意味を持ちます。

0 : アクティブライン

1 : ライン間同期

埋め込み同期の場合、このビットは DCMI\_CR の CAPTURE ビットがセットされている場合にのみ意味を持ちます。

## 17.8.3 DCMI raw 割り込みステータスレジスタ (DCMI\_RIS)

アドレスオフセット : 0x08

リセット値 : 0x0000 0x0000

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

DCMI\_RIS は raw 割り込みステータスを示し、読み出し専用でアクセスできます。読み出し時、このレジスタは DCMI\_IER レジスタ値でマスクする前の関連する割り込みのステータスを返します。

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE\_RIS** : ライン raw 割り込みステータス

HSYNC 信号がインアクティブ状態からアクティブ状態に変わると、このビットがセットされます。ラインが有効でなくてもハイになります。

埋め込み同期の場合、このビットは DCMI\_CR の CAPTURE ビットがセットされている場合にのみセットされます。

DCMI\_ICR の LINE\_ISC ビットに“1”を書き込むことによってクリアされます。

ビット 3 **VSYNC\_RIS** : VSYNC raw 割り込みステータス

VSYNC 信号がインアクティブ状態からアクティブ状態に変わると、このビットがセットされます。

埋め込み同期の場合、このビットは DCMI\_CR の CAPTURE ビットがセットされている場合にのみセットされます。

DCMI\_ICR の VSYNC\_ISC ビットに“1”を書き込むことによってクリアされます。

ビット 2 **ERR\_RIS** : 同期エラー raw 割り込みステータス

0 : 同期エラーは検出されていません。

1 : 埋め込み同期文字が正しい順番で受信されません。

このビットは、埋め込み同期モードでのみ有効です。DCMI\_ICR の ERR\_ISC ビットに“1”を書き込むことによってクリアされます。

**注 :** このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR\_RIS** : オーバーラン raw 割り込みステータス

0 : データバッファオーバーランは発生していません。

1 : データバッファオーバーランが発生し、データ FIFO が不正です。

このビットは、DCMI\_ICR の OVR\_ISC ビットに“1”を書き込むことによってクリアされます。

ビット 0 **FRAME\_RIS** : キャプチャ完了 raw 割り込みステータス

0 : 新規キャプチャはありません。

1 : フレームキャプチャ完了しています。

このビットは、フレームやウィンドウがキャプチャされるとセットされます。

クロップされたウィンドウの場合、このビットは、クロップの最後のラインの終了時にセットされます。キャプチャされたフレームが空でもセットされます (例 : フレーム外でクロップされたウィンドウなど)。

このビットは、DCMI\_ICR の FRAME\_ISC ビットに“1”を書き込むことによってクリアされます。

## 17.8.4 DCMI 割り込み有効レジスタ (DCMI\_IER)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINE_IE	VSYNC_IE	ERR_IE	OVR_IE	FRAME_IE																										
																											r/w	r/w	r/w	r/w	r/w

DCMI\_IER レジスタは、割り込みを有効にするために使用します。DCMI\_IER ビットの 1 つがセットされると、該当する割り込みが有効になります。このレジスタは、読み出しと書き込みの両方でアクセスできます。

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE\_IE** : ライン割り込みイネーブル

- 0 : ライン受信時に割り込みは生成されません。
- 1 : ラインすべてを受信したときに割り込みが生成されます。

ビット 3 **VSYNC\_IE** : VSYNC 割り込みイネーブル

- 0 : 割り込みは生成されません。
  - 1 : VSYNC がインアクティブからアクティブ状態に遷移するたびに割り込みが生成されます。
- VSYNC 信号のアクティブ状態は、VSPOL ビットによって定義されます。

ビット 2 **ERR\_IE** : 同期エラー割り込みイネーブル

- 0 : 割り込みは生成されません。
- 1 : 埋め込み同期コードが正しい順番で受信されない場合に割り込みが生成されます。

**注 :** このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR\_IE** : オーバーラン割り込みイネーブル

- 0 : 割り込みは生成されません。
- 1 : DMA が新しいデータ (32 ビット) を受信する前に最後のデータを転送できなかった場合に割り込みが生成されます。

ビット 0 **FRAME\_IE** : キャプチャ完了割り込みイネーブル

- 0 : 割り込みは生成されません。
- 1 : 受信したフレーム/クロップウィンドウ (クロップモードの場合) の終了時に割り込みが生成されます。

## 17.8.5 DCMI マスク済み割り込みステータスレジスタ (DCMI\_MIS)

DCMI\_MIS レジスタは、読み出し専用レジスタです。読み出し時、該当する割り込みの現在のマスクステータス値 (DCMI\_IER の値によって変わります) を返します。DCMI\_IER 内の該当するイネーブルビットがセットされ、DCMI\_RIS 内の該当ビットがセットされると、このレジスタ内のビットがセットされます。

アドレスオフセット : 0x10

リセット値 : 0x0000

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

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE\_MIS** : マスクされたライン割り込みステータス

このビットは、マスクされたライン割り込みのステータスを示します。

0 : ライン受信時に割り込みは生成されません。

1 : ラインが完全に受信され、DCMI\_IER の LINE\_IE ビットがセットされると割り込みが生成されます。

ビット 3 **VSYNC\_MIS** : マスクされた VSYNC 割り込みステータス

このビットは、マスクされた VSYNC 割り込みのステータスを示します。

0 : VSYNC の遷移で割り込みは生成されません。

1 : VSYNC がインアクティブからアクティブ状態に遷移し、DCMI\_IER の VSYNC\_IE ビットがセットされるたびに割り込みが生成されます。

VSYNC 信号のアクティブ状態は、VSPOL ビットによって定義されます。

ビット 2 **ERR\_MIS** : マスクされた同期エラー割り込みステータス

このビットは、マスクされた同期エラー割り込みのステータスを示します。

0 : 同期エラー時、割り込みは生成されません。

1 : 埋め込み同期コードが正しい順番で受信されず、DCMI\_IER の ERR\_IE ビットがセットされると割り込みが生成されます。

**注 :** このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR\_MIS** : マスクされたオーバーラン割り込みステータス

このビットは、マスクされたオーバーフロー割り込みのステータスを示します。

0 : オーバーラン時、割り込みは生成されません。

1 : DMA が新しいデータ (32 ビット) を受信する前に最後のデータを転送できず、DCMI\_IER の OVR\_IE ビットがセットされると割り込みが生成されます。

ビット 0 **FRAME\_MIS** : マスクされたキャプチャ完了割り込みステータス

このビットは、マスクされたキャプチャ完了割り込みのステータスを示します。

0 : キャプチャ完了で割り込みは生成されません。

1 : 受信したフレーム/クロップウィンドウ (クロップモードの場合) の終了時に DCMI\_IER の FRAME\_IE ビットがセットされると割り込みが生成されます。

## 17.8.6 DCMI 割り込みクリアレジスタ (DCMI\_ICR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0x0000

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

DCMI\_ICR レジスタは、書き込み専用です。このレジスタのビットに“1”を書き込むことによって DCMI\_RIS および DCMI\_MIS レジスタの対応するビットをクリアします。“0”を書き込んで、ビットの値は変化しません。

ビット 15:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **LINE\_ISC** : ライン割り込みステータスクリア

このビットに“1”を書き込むと、DCMI\_RIS レジスタの LINE\_RIS ビットがクリアされます。

ビット 3 **VSYNC\_ISC** : 垂直同期割り込みステータスクリア

このビットに“1”を書き込むと、DCMI\_RIS レジスタの VSYNC\_RIS ビットがクリアされます。

ビット 2 **ERR\_ISC** : 同期エラー割り込みステータスクリア

このビットに“1”を書き込むと、DCMI\_RIS レジスタの ERR\_RIS ビットがクリアされます。

**注:** このビットは、埋め込み同期モードでのみ使用できます。

ビット 1 **OVR\_ISC** : オーバーラン割り込みステータスクリア

このビットに“1”を書き込むと、DCMI\_RIS の OVR\_RIS ビットがクリアされます。

ビット 0 **FRAME\_ISC** : キャプチャ完了割り込みステータスクリア

このビットに“1”を書き込むと、DCMI\_RIS の FRAME\_RIS ビットがクリアされます。

## 17.8.7 DCMI 埋め込み同期コードレジスタ (DCMI\_ESCR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FEC								LEC								LSC								FSC								
r/w																																

ビット 31:24 **FEC** : フレーム終了デリミタコード

このバイトによって、フレーム終了デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、FEC の形式で 4 バイトで構成されています。

FEC が 0xFF にプログラムされている場合、使用されていないすべてのコード (0xFF0000XY) はフレーム終了デリミタとして解釈されます。

ビット 23:16 **LEC** : ライン終了デリミタコード

このバイトによって、ライン終了デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、LEC の形式で 4 バイトで構成されています。

ビット 15:8 **LSC** : ライン開始デリミタコード

このバイトによって、ライン開始デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、LSC の形式で 4 バイトで構成されています。

ビット 7:0 **FSC** : フレーム開始デリミタコード

このバイトによって、フレーム開始デリミタのコードを指定します。このコードは 0xFF、0x00、0x00、FSC の形式で 4 バイトで構成されています。

FSC が 0xFF にプログラムされている場合、フレーム開始デリミタは検出されません。FEC コードの後に現れる最初の LSC は、フレームの開始デリミタとして解釈されます。

## 17.8.8 DCMI 埋め込み同期マスク解除レジスタ (DCMI\_ESUR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEU								LEU								LSU								FSU							
r/w																															

### ビット 31:24 FEU : フレーム終了デリミタマスク解除

このバイトによって、フレーム終了デリミタのコードに適用するマスクを指定します。

0: 受信したデータとフレーム終了デリミタを比較すると同時に、DCMI\_ESCR の FEC バイトの該当ビットがマスクされます。

1: 受信したデータとフレーム終了デリミタを比較すると同時に、DCMI\_ESCR の FEC バイトの該当ビットが比較されます。

### ビット 23:16 LEU : ライン終了デリミタマスク解除

このバイトによって、ライン終了デリミタのコードに適用するマスクを指定します。

0: 受信したデータとライン終了デリミタを比較すると同時に、DCMI\_ESCR の LEC バイトの該当ビットがマスクされます。

1: 受信したデータとライン終了デリミタを比較すると同時に、DCMI\_ESCR の LEC バイトの該当ビットが比較されます。

### ビット 15:8 LSU : ライン開始デリミタマスク解除

このバイトによって、ライン開始デリミタのコードに適用するマスクを指定します。

0: 受信したデータとライン開始デリミタを比較すると同時に、DCMI\_ESCR の LSC バイトの該当ビットがマスクされます。

1: 受信したデータとライン開始デリミタを比較すると同時に、DCMI\_ESCR の LSC バイトの該当ビットが比較されます。

### ビット 7:0 FSU : フレーム開始デリミタマスク解除

このバイトによって、フレーム開始デリミタのコードに適用するマスクを指定します。

0: 受信したデータとフレーム開始デリミタを比較すると同時に、DCMI\_ESCR の FSC バイトの該当ビットがマスクされます。

1: 受信したデータとフレーム開始デリミタを比較すると同時に、DCMI\_ESCR の FSC バイトの該当ビットが比較されます。

## 17.8.9 DCMI クロップウィンドウ開始 (DCMI\_CWSTRT)

アドレスオフセット : 0x20

リセット値 : 0x0000 0x0000

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

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:16 **VST[12:0]** : 垂直方向開始ラインカウント

このライン番号で画像キャプチャを開始します。これより前のラインデータは無視されます。

0x0000 => ライン 1

0x0001 => ライン 2

0x0002 => ライン 3

....

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **HOFFCNT[13:0]** : 水平方向オフセットカウント

キャプチャ開始前にカウントするピクセルクロック数を示します。

## 17.8.10 DCMI クロップウィンドウサイズ (DCMI\_CWSIZE)

アドレスオフセット : 0x24

リセット値 : 0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rRes.	rRes.	VLINEL3:0]													rRes.	rRes.	CAPCNT[13:0]														
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW			rW													

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:16 **VLINEL3:0]** : 垂直方向ラインカウント

開始点からのキャプチャライン数を示します。

0x0000 => 1 ライン

0x0001 => 2 ライン

0x0002 => 3 ライン

....

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **CAPCNT[13:0]** : キャプチャカウント

同一ライン上でキャプチャする開始点からのピクセルクロック数を示します。この値は、パラレルインタフェースのさまざまなデータ幅に対してワードアライメントが施されたデータに対応します。

0x0000 => 1 ピクセル

0x0001 => 2 ピクセル

0x0002 => 3 ピクセル

....

## 17.8.11 DCMI データレジスタ (DCMI\_DR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte3								Byte2								Byte1								Byte0							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 データバイト 3

ビット 23:16 データバイト 2

ビット 15:8 データバイト 1

ビット 7:0 データバイト 0

このデジタルカメラインタフェースは、DMA 転送を要求する前に受信したデータをすべて 32 ビット形式にまとめます。DMA 転送に十分な時間を与え、DMA オーバーランを回避するため、深さ 4 ワードの FIFO が使用できます。

## 17.8.12 DCMI レジスタマップ

表 113 に DCMI レジスタの要約を示します。

表 113. DCMI レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DCMI_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBS	BSM		Res.	ENABLE	Res.	Res.	EDM	FCRC		VSPOL	HSPOL	POKPOL	ESS	JPEG	CROP	CM	CAPTURE	
	リセット値												0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0
0x04	DCMI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FNE	VSYNC	HSYNC	
	リセット値																													0	0	0	0
0x08	DCMI_RIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNC_RIS	ERR_RIS	OVR_RIS	FRAME_RIS
	リセット値																												0	0	0	0	0
0x0C	DCMI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_IE	VSYNC_IE	ERR_IE	OVR_IE	FRAME_IE
	リセット値																												0	0	0	0	0
0x10	DCMI_MIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS
	リセット値																												0	0	0	0	0
0x14	DCMI_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_ISC	VSYNC_ISC	ERR_ISC	OVR_ISC	FRAME_ISC
	リセット値																												0	0	0	0	0
0x18	DCMI_ESCR	FEC								LEC								LSC								FSC							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	DCMI_ESUR	FEU								LEU								LSU								FSU							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DCMI_CWSTRT	Res.	Res.	Res.	VST[12:0]												Res.	Res.	HOFFCNT[13:0]														
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	DCMI_CWSIZE	Res.	Res.	VLIN[13:0]												Res.	Res.	CAPCNT[13:0]															
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	DCMI_DR	Byte3								Byte2								Byte1								Byte0							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 18 LCD-TFT コントローラ (LTDC)

このセクションは、特に指定がない限り、STM32F756xx および STM32F46xx デバイス全体に適用されます。

### 18.1 概要

この LCD-TFT (TFT 液晶) ディスプレイコントローラは、さまざまな LCD および TFT パネルに直接接続するため、デジタルパラレル RGB (赤、緑、青) や、水平/垂直同期、ピクセルクロック、およびデータイネーブルのためのインタフェースを搭載しています。

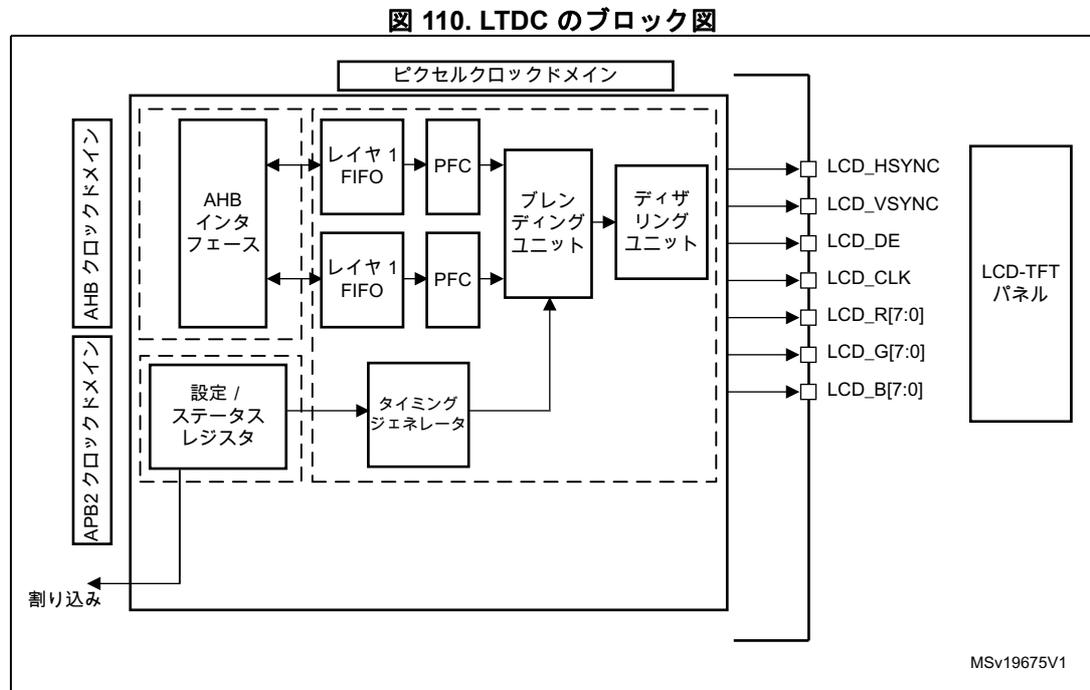
### 18.2 LTDC の主な機能

- 24 ビット RGB パラレル出力、8 ビット/ピクセル (RGB888)
- 専用 FIFO (64x32 ビット) を備えた 2 つの表示レイヤ
- 各レイヤにつき最大 256 色 (256x24 ビット) のカラーlookupアップテーブル (CLUT)
- XGA (1024x768) までの解像度をサポート
- さまざまなディスプレイパネルに対応するプログラム可能なタイミング
- プログラム可能なバックグラウンドカラー
- HSync、VSync、データイネーブルに対するプログラム可能な極性
- 各レイヤにつき最大 8 種類の入力カラーフォーマットが選択可能
  - ARGB8888
  - RGB888
  - RGB565
  - ARGB1555
  - ARGB4444
  - L8 (8 ビット輝度または CLUT)
  - AL44 (4 ビットアルファ+4 ビット輝度)
  - AL88 (8 ビットアルファ+8 ビット輝度)
- 各チャンネルにつき、低ビットに対する疑似ランダムディザリング出力
  - 赤、緑、青に対するディザ幅 2 ビット
- アルファ値 (ピクセルごとまたは一定値) を用いた 2 つのレイヤ間の柔軟なブレンディング
- カラーキーイング (透明色)
- プログラム可能なウィンドウ位置とサイズ
- TFT カラー液晶ディスプレイをサポート
- 16 ワードのバースト付き AHB マスタインタフェース
- 最大 4 つのプログラム可能な割り込みイベント

## 18.3 LTDC の機能詳細

### 18.3.1 LTDC のブロック図

LTDC のブロック図を [図 110 : LTDC のブロック図](#) に示します。



レイヤ FIFO : レイヤごとに FIFO 64x32 ビットを 1 つずつ搭載しています。

PFC : レイヤに対して選択した入力ピクセルフォーマットからワードへのピクセルフォーマット変換を実行するピクセルフォーマットコンバータです。

AHB インタフェース : メモリから FIFO へのデータ転送用です。

ブレンドユニット、ディザリングユニット、タイミングジェネレータ : [セクション 18.4.1](#) および [セクション 18.4.2](#) を参照してください。

### 18.3.2 LTDC のリセットおよびクロック

LTDC では 3 つのクロックドメインを使用します。

- AHB クロックドメイン (HCLK) : メモリからレイヤ FIFO へのデータ転送用
- APB2 クロックドメイン (PCLK2) : レジスタ設定用
- ピクセルクロックドメイン (LCD\_CLK) : LCD-TFT インタフェース信号生成用 LCD\_CLK 出力は、パネル要件に従って設定する必要があります。LCD\_CLK は、PLLSAI を介して設定します (RCC のセクションを参照)。

LTDC は、RCC\_APB2RSTR レジスタの該当するビットをセットすることによってリセットできます。このレジスタで 3 つのクロックドメインをリセットします。

## 18.3.3 LCD-TFT ピンと信号インタフェース

下表に、LTDC 信号インタフェースの一覧を示します。

表 114. LCD-TFT ピンと信号インタフェース

LCD-TFT 信号	I/O	説明
LCD_CLK	O	クロック出力
LCD_HSYNC	O	水平同期
LCD_VSYNC	O	垂直同期
LCD_DE	O	データイネーブル
LCD_R[7:0]	O	データ : 8 ビットの赤のデータ
LCD_G[7:0]	O	データ : 8 ビットの緑のデータ
LCD_B[7:0]	O	データ : 8 ビットの青のデータ

LTDC のピンは、ユーザアプリケーションで設定する必要があります。使用されていないピンは、他の目的に使用できます。

24 ビットまでの LTDC 出力 (RGB888) に対して、16 ビット (RGB565) または 18 ビット (RGB666) カラーのディスプレイに接続するため、8bpp 未満の色深度を使用する場合、ディスプレイ側の RGB データラインは LTDC の RGB データラインの MSB 側に接続する必要があります。例えば、16 ビット (RGB565) カラーのディスプレイを接続する場合、ディスプレイ側の R[4:0]、G[5:0]、B[4:0] のデータラインのピンは LTDC の LCD\_R[7:3]、LCD\_G[7:2]、LCD\_B[7:3] に接続する必要があります。

## 18.4 LTDC のプログラム可能なパラメータ

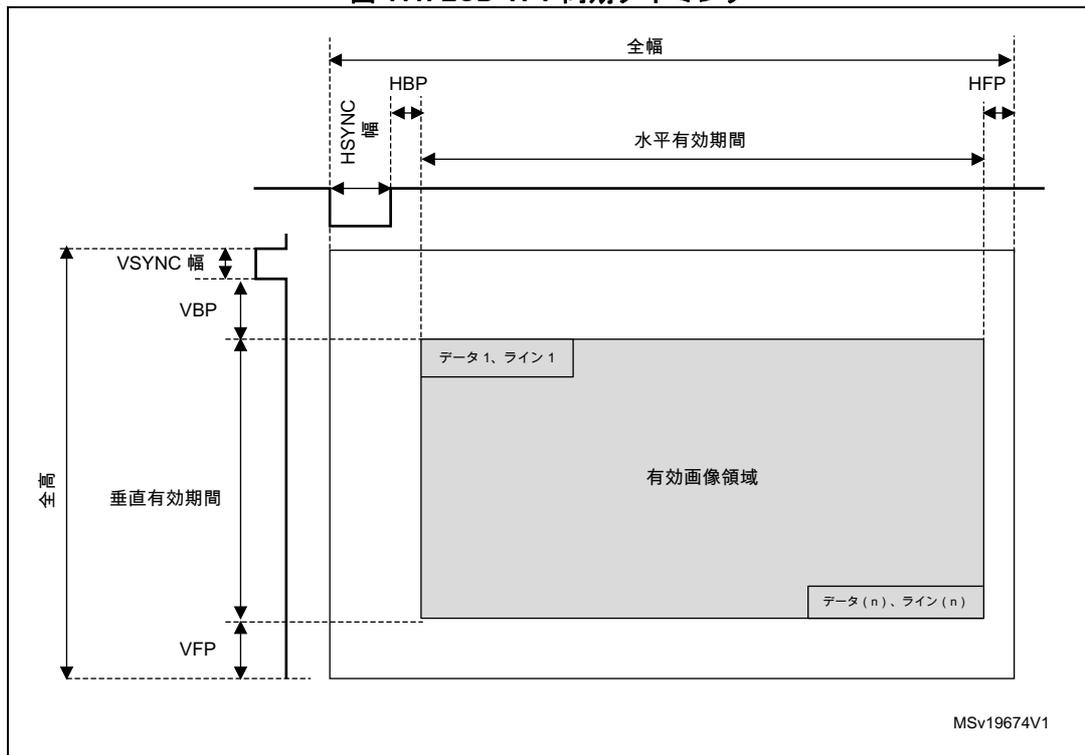
LCD-TFT コントローラのパラメータは、フレキシブルに設定することができます。LTDC\_GCR レジスタを介して有効または無効にすることができます。

### 18.4.1 LTDC グローバル設定パラメータ

#### 同期タイミング :

[図 111](#) は、[図 110](#) のブロック図で示す同期タイミングジェネレータブロックで生成される設定可能なタイミングパラメータを示します。水平/垂直同期信号、ピクセルクロックおよびデータイネーブル信号が生成されます。

図 111. LCD-TFT 同期タイミング



注： HBP および HFP は、それぞれ水平バックポーチ期間およびフロントポーチ期間です。  
VBP および VFP は、それぞれ垂直バックポーチ期間およびフロントポーチ期間です。

LCD-TFT のプログラム可能な同期タイミングは次のとおりです。

- HSYNC および VSYNC 幅: LTDC\_SSCR レジスタにおける **HSYNC 幅 - 1** および **VSYNC 幅 - 1** の値をプログラムして設定される水平および垂直同期信号幅です。
- HBP および VBP: LTDC\_BPCR レジスタにおける **HSYNC 幅 + HBP - 1** の合計値と **VSYNC 幅 + VBP - 1** の合計値をプログラムして設定される水平および垂直バックポーチ幅です。
- アクティブ幅とアクティブ高さ: LTDC\_AWCR レジスタで **HSYNC 幅 + HBP + 水平有効期間 - 1** の合計値と **VSYNC 幅 + VBP + 垂直有効期間 - 1** の合計値をプログラムして設定します (1024x768 までをサポート)。
- 全幅: LTDC\_TWCR レジスタで **HSYNC 幅 + HBP + 水平有効期間 + HFP - 1** の合計値をプログラムして設定します。HFP は、水平フロントポーチ期間です。
- 全高: LTDC\_TWCR レジスタで **VSYNC 高さ + VBP + 垂直有効期間 + VFP - 1** の合計値をプログラムして設定します。VFP は、垂直フロントポーチ期間です。

注： LTDC が有効の場合、垂直同期エリアの最初の水平同期ピクセルとして、X/Y=0/0 位置でタイミング生成が開始し、その後バックポーチ、アクティブデータ表示エリア、フロントポーチと続きます。

LTDC が無効の場合、タイミングジェネレータブロックは X = 全幅 - 1、Y = 全高 - 1 にリセットされ、垂直同期フェーズの前の最後のピクセルを保持し、FIFO は一掃されます。したがって、連続的に出力されるのはブランキングデータのみとなります。

## 同期タイミング設定の例：

TFT-LCD タイミング (パネルデータシートより抽出)：

- 水平および垂直同期幅：0x8 ピクセルおよび 0x4 ライン
- 水平および垂直バックポーチ：0x7 ピクセルおよび 0x2 ライン
- アクティブ幅とアクティブ高さ：0x280 ピクセル、0x1E0 ライン (640x480)
- 水平フロントポーチ：0x6 ピクセル
- 垂直フロントポーチ：0x2 ライン

LTDC タイミングレジスタでプログラムされる値は次のようになります。

- **LTDC\_SSCR** レジスタ：0x00070001 にプログラム (HSW[11:0] は 0x7、VSH[10:0] は 0x3)
- **LTDC\_BPCR** レジスタ：0x000E0005 にプログラム (AHBP[11:0] は 0xE (0x8 + 0x6)、AVBP[10:0] は 0x5 (0x4 + 0x1))
- **LTDC\_AWCR** レジスタ：0x028E01E5 にプログラム (AAW[11:0] は 0x28E (0x8 + 0x7 + 0x27F)、AAH[10:0] は 0x1E5 (0x4 + 0x2 + 0x1DF))
- **LTDC\_TWCR** レジスタ：0x00000294 にプログラム (TOTALW[11:0] は 0x294 (0x8 + 0x7 + 0x280 + 0x5))
- **LTDC\_THCR** レジスタ：0x000001E7 にプログラム (TOTALH[10:0] は 0x1E7 (0x4 + 0x2 + 0x1E0 + 1))

## プログラム可能な極性

水平/垂直同期信号、データイネーブル信号、ピクセルクロックの極性は、**LTDC\_GCR** レジスタでアクティブハイまたはアクティブローにプログラムできます。

## バックグラウンドカラー

一定の背景色 (RGB888) を **LTDC\_BCCR** レジスタでプログラムできます。ボトムレイヤとのブレンドに使用します。

## ディザリング

LFSR を使用した疑似乱数ディザリングを使用して、各ピクセルカラーチャンネル (R、G、または B) 値に小さな乱数値 (閾値) を加えますので、18 ビットディスプレイで 24 ビットデータを表示する際に、MSB が切り上げられる場合があります。このようにして、フレームごとに異なるデータの四捨五入にはディザリングが使用されず。

疑似乱数ディザリングは、LSB を閾値と比較して LSB が閾値以上であれば MSB にのみ 1 を加算することと同じです。ディザリングが適用されると、LSB は通常切り捨てられます。

加算される疑似乱数値の幅は、各カラーチャンネルに対して 2 ビット、つまり赤に対して 2 ビット、緑に対して 2 ビット、青に対して 2 ビットです。

LTDC が有効になると、LFSR が最初のアクティブなピクセルで動作を開始し、ブランキング期間中やディザリングがオフになってもそのまま動作を続けます。LTDC が無効になると、LFSR はリセットされます。

ディザリングは、**LTDC\_GCR** レジスタを使って動作中にオン/オフを切り替えることができます。

## シャドウレジスタの再ロード

シャドウイングされる設定レジスタがいくつかあります。シャドウレジスタの値は、**LTDC\_SRCR** レジスタでの設定に従い、アクティブなレジスタへの書き込み時、または垂直ブランキング期間の始めに、これらのレジスタに直ちに再ロードすることができます。即時再ロード設定が選択されている場

合、すべての新規レジスタが書き込まれてから再ロードがアクティブになるようにする必要があります。

シャドウレジスタは、再ロードが完了するまでは再変更してはいけません。シャドウレジスタを読み出すと、実際にアクティブな値が読み出されます。新しく書き込まれた値は、再ロードが実行されるまでは読み出せません。

LTDC\_IER レジスタで有効になっていれば、レジスタ再ロード割り込みを生成することができます。

LTDC\_LxCLUTWR レジスタを除くすべてのレイヤ 1 とレイヤ 2 のレジスタはシャドウレジスタです。

## 割り込み生成イベント

割り込み設定については、[セクション 18.5 : LTDC 割り込み](#)を参照してください。

## 18.4.2 レイヤのプログラム可能なパラメータ

最大 2 つのレイヤが有効化、無効化でき、また別々に設定することができます。レイヤの表示順序は固定で、下から上に順番に表示されます。2 つのレイヤが有効になっている場合は、レイヤ 2 が上部に表示されるウィンドウとなります。

### ウィンドウイング

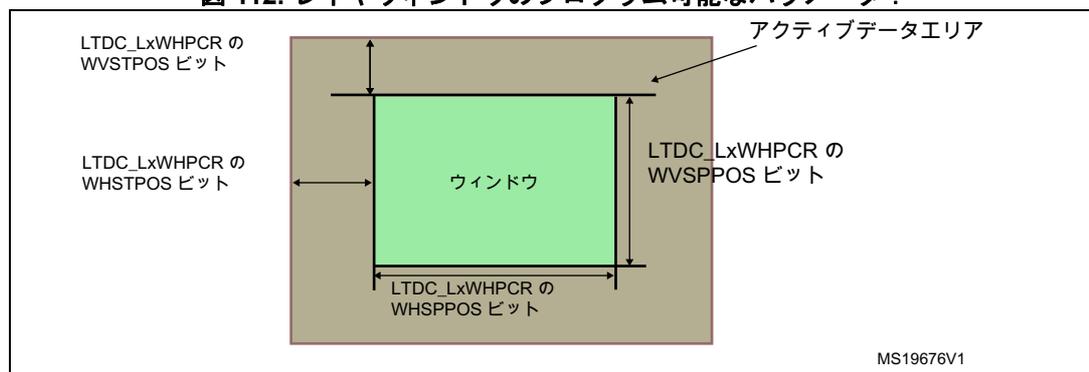
すべてのレイヤは位置決めとサイズ変更ができますが、アクティブ表示エリア内であることが必要です。

ウィンドウ位置およびサイズは、左上と右下の X/Y 位置、および内部タイミングジェネレータによって設定され、この内部タイミングジェネレータは、同期、バックポーチサイズ、アクティブデータエリアから成っています。LTDC\_LxWHPCR および LTDC\_WVPCR レジスタを参照してください。

プログラム可能なレイヤ位置とサイズにより、ラインの最初/最後の見えるピクセルと、ウィンドウの最初/最後の見えるラインが決まります。画像フレームの全体または一部のいずれでも表示することができます。[図 112](#) を参照してください。

- レイヤ内の最初と最後の見えるピクセルは、LTDC\_LxWHPCR レジスタ内の WHSTPOS[11:0] および WHSPPOS[11:0] を設定することによりセットされます。
- レイヤ内の最初と最後の見えるラインは、LTDC\_LxWVPCR レジスタ内の WHSTPOS[10:0] および WHSPPOS[10:0] を設定することによりセットされます。

図 112. レイヤウィンドウのプログラム可能なパラメータ :



## ピクセル入力フォーマット

レイヤのフレームバッファ内に保存されるデータには、プログラム可能なピクセルフォーマットが使用されます。

LTDC\_LxPFCR レジスタで、各レイヤに対して最大 8 種類までの入力ピクセルフォーマットを設定できます。

ピクセルデータはフレームバッファから読み出され、次のように内部 8888 (ARGB) フォーマットに変換されます。

- 幅が 8 ビット未満のコンポーネントは、ビットレプリケーションによって 8 ビットまで拡張されます。選択されたビット範囲は、8 ビットより長くなるまで複数回連結されます。得られたベクタから、8 つの MSB ビットが選択されます。例：RGB565 赤チャネルの 5 ビットは次のようになります (ビット位置)：43210432 (3 つの LSB に 5 ビットの 3 つの MSB が入る)

下図で、選択したフォーマットに応じたピクセルデータのマッピングを説明しています。

表 115. ピクセルデータマッピングとカラーフォーマット

ARGB8888			
@+3 A <sub>x</sub> [7:0]	@+2 R <sub>x</sub> [7:0]	@+1 G <sub>x</sub> [7:0]	@ B <sub>x</sub> [7:0]
@+7 A <sub>x+1</sub> [7:0]	@+6 R <sub>x+1</sub> [7:0]	@+5 G <sub>x+1</sub> [7:0]	@+4 B <sub>x+1</sub> [7:0]
RGB888			
@+3 B <sub>x+1</sub> [7:0]	@+2 R <sub>x</sub> [7:0]	@+1 G <sub>x</sub> [7:0]	@ B <sub>x</sub> [7:0]
@+7 G <sub>x+2</sub> [7:0]	@+6 B <sub>x+2</sub> [7:0]	@+5 R <sub>x+1</sub> [7:0]	@+4 G <sub>x+1</sub> [7:0]
RGB565			
@+3 R <sub>x+1</sub> [4:0] G <sub>x+1</sub> [5:3]	@+2 G <sub>x+1</sub> [2:0] B <sub>x+1</sub> [4:0]	@+1 R <sub>x</sub> [4:0] G <sub>x</sub> [5:3]	@ G <sub>x</sub> [2:0] B <sub>x</sub> [4:0]
@+7 R <sub>x+3</sub> [4:0] G <sub>x+3</sub> [5:3]	@+6 G <sub>x+3</sub> [2:0] B <sub>x+3</sub> [4:0]	@+5 R <sub>x+2</sub> [4:0] G <sub>x+2</sub> [5:3]	@+4 G <sub>x+2</sub> [2:0] B <sub>x+2</sub> [4:0]
ARGB1555			
@+3 A <sub>x+1</sub> [0]R <sub>x+1</sub> [4:0] G <sub>x+1</sub> [4:3]	@+2 G <sub>x+1</sub> [2:0] B <sub>x+1</sub> [4:0]	@+1 A <sub>x</sub> [0] R <sub>x</sub> [4:0] G <sub>x</sub> [4:3]	@ G <sub>x</sub> [2:0] B <sub>x</sub> [4:0]
@+7 A <sub>x+3</sub> [0]R <sub>x+3</sub> [4:0] G <sub>x+3</sub> [4:3]	@+6 G <sub>x+3</sub> [2:0] B <sub>x+3</sub> [4:0]	@+5 A <sub>x+2</sub> [0]R <sub>x+2</sub> [4:0]G <sub>x+2</sub> [4:3]	@+4 G <sub>x+2</sub> [2:0] B <sub>x+2</sub> [4:0]
ARGB4444			
@+3 A <sub>x+1</sub> [3:0]R <sub>x+1</sub> [3:0]	@+2 G <sub>x+1</sub> [3:0] B <sub>x+1</sub> [3:0]	@+1 A <sub>x</sub> [3:0] R <sub>x</sub> [3:0]	@ G <sub>x</sub> [3:0] B <sub>x</sub> [3:0]
@+7 A <sub>x+3</sub> [3:0]R <sub>x+3</sub> [3:0]	@+6 G <sub>x+3</sub> [3:0] B <sub>x+3</sub> [3:0]	@+5 A <sub>x+2</sub> [3:0]R <sub>x+2</sub> [3:0]	@+4 G <sub>x+2</sub> [3:0] B <sub>x+2</sub> [3:0]

表 115. ピクセルデータマッピングとカラーフォーマット (続き)

ARGB8888			
L8			
@+3 L <sub>x+3</sub> [7:0]	@+2 L <sub>x+2</sub> [7:0]	@+1 L <sub>x+1</sub> [7:0]	@ L <sub>x</sub> [7:0]
@+7 L <sub>x+7</sub> [7:0]	@+6 L <sub>x+6</sub> [7:0]	@+5 L <sub>x+5</sub> [7:0]	@+4 L <sub>x+4</sub> [7:0]
AL44			
@+3 A <sub>x+3</sub> [3:0] L <sub>x+3</sub> [3:0]	@+2 A <sub>x+2</sub> [3:0] L <sub>x+2</sub> [3:0]	@+1 A <sub>x+1</sub> [3:0] L <sub>x+1</sub> [3:0]	@ A <sub>x</sub> [3:0] L <sub>x</sub> [3:0]
@+7 A <sub>x+7</sub> [3:0] L <sub>x+7</sub> [3:0]	@+6 A <sub>x+6</sub> [3:0] L <sub>x+6</sub> [3:0]	@+5 A <sub>x+5</sub> [3:0] L <sub>x+5</sub> [3:0]	@+4 A <sub>x+4</sub> [3:0] L <sub>x+4</sub> [3:0]
AL88			
@+3 A <sub>x+1</sub> [7:0]	@+2 L <sub>x+1</sub> [7:0]	@+1 A <sub>x</sub> [7:0]	@ L <sub>x</sub> [7:0]
@+7 A <sub>x+3</sub> [7:0]	@+6 L <sub>x+3</sub> [7:0]	@+5 A <sub>x+2</sub> [7:0]	@+4 L <sub>x+2</sub> [7:0]

## カラーlookupアップテーブル (CLUT)

CLUT は、LTDC\_LxCR レジスタですべてのレイヤに対してランタイム時に有効にできますが、L8、AL44、AL88 入力ピクセルフォーマット使用時のインデックスカラーの場合にのみ役立ちます。

まず、CLUT にそのピクセルの本来の R、G、B 値に代わる R、G、B 値をロードする必要があります (インデックスカラー)。各色 (RGB 値) には、それ自体のアドレスがあり、CLUT 内の位置を表します。

R、G、B 値とそれぞれのアドレスは LTDC\_LxCLUTWR レジスタでプログラムされます。

- L8 および AL88 入力ピクセルフォーマットの場合、CLUT は 256 色でロードする必要があります。各色のアドレスは、LTDC\_LxCLUTWR レジスタ内の CLUTADD ビットで設定されます。
- AL44 入力ピクセルフォーマットの場合、CLUT は 16 色でロードする必要があります。各色のアドレスは、4 ビット L チャンネルを次のように 8 ビットに複製することによって埋める必要があります。
  - L0 (インデックスカラー 0)、アドレス 0x00
  - L1、アドレス 0x11
  - L2、アドレス 0x22
  - .....
  - L15、アドレス 0xFF

## カラーフレームバッファアドレス

各レイヤには、LTDC\_LxCFBAR レジスタで設定されるカラーフレームバッファの開始アドレスがあります。

レイヤが有効になると、カラーフレームバッファからデータがフェッチされます。

## カラーフレームバッファ長

すべてのレイヤには、バイト数単位でのカラーフレームバッファの総ライン長設定とフレームバッファ内のライン数設定があり、それぞれ `LTDC_LxCFBLR` および `LTDC_LxCFBLNR` レジスタで設定可能です。

ライン長とライン数の設定は、フレームバッファの最後でレイヤ FIFO へのデータのプリフェッチを停止するために使用されます。

- 必要なバイト数よりも少なく設定されている場合、FIFO アンダーラン割り込みが事前に有効になっていると、該当する割り込みが生成されます。
- 実際に必要なバイト数よりも多く設定されている場合、FIFO から読み出された不要なデータは破棄されます。不要なデータは表示されません。

## カラーフレームバッファピッチ

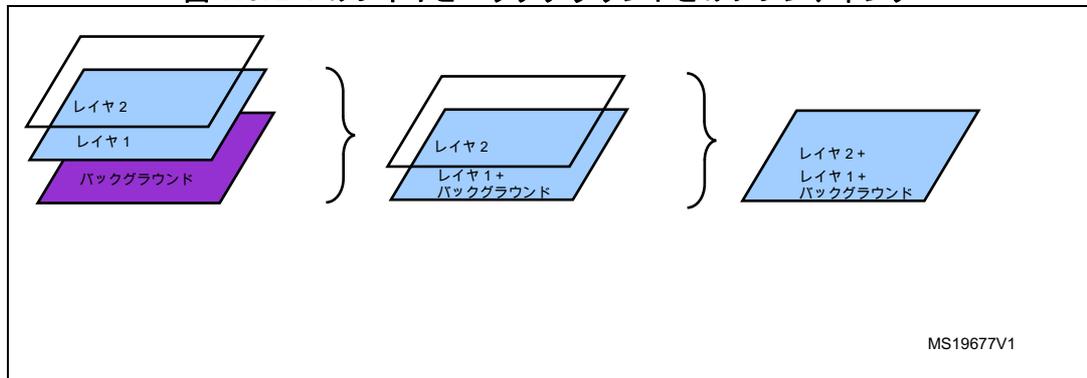
すべてのレイヤには、カラーフレームバッファに対して設定可能なピッチがあります。これはあるラインの開始から次のラインの始まりまでの距離をバイト数で表したものです。ピッチは、`LTDC_LxCFBLR` レジスタで設定します。

## レイヤのブレンディング

ブレンディングは常にアクティブとなっており、2つのレイヤは `LTDC_LxBFCR` レジスタで設定されるブレンディング係数に従ってブレンドすることができます。

混合順序は固定で、下から上に順番に混合されます。2つのレイヤが有効になっている場合、まずレイヤ1がバックグラウンド色と混合され、その後レイヤ2がレイヤ1とバックグラウンドとの混合色と混合されます。図 113を参照してください。

図 113. 2つのレイヤとバックグラウンドとのブレンディング



## デフォルト色

すべてのレイヤには、フォーマット ARGB でデフォルト色を設定することができ、このデフォルト色は定義されているレイヤウィンドウ外またはレイヤが無効になっているときに使用されます。

デフォルト色は、`LTDC_LxDCCR` レジスタで設定します。

ブレンディングは、レイヤの1つが無効になっていても必ず2つのレイヤ間で実施されます。レイヤが無効になっているときにデフォルト色が表示されないようにするには、`LTDC_LxBFCR` レジスタの該当レイヤのブレンディング係数をリセット値に保持します。

## カラーキーイング

カラーキー (RGB) を設定し、透明なピクセルを表すことができます。

カラーキーイングが有効になっていると、現在のピクセル(フォーマット変換後でブレンディング前)がカラーキーと比較されます。プログラムされた RGB 値と一致した場合、そのピクセルのすべてのチャンネル (ARGB) が 0 にセットされます。

カラーキー値を設定することにより、ランタイム時にその値を使用してピクセル RGB 値を置換することができます。

カラーキーイングは、LTDC\_LxCKCR レジスタで有効にします。

## 18.5 LTDC 割り込み

LTDC により、2 つの割り込みベクタの論理和をとった 4 つのマスク可能な割り込みが使用できます。

割り込みソースは、LRDC\_IER レジスタを介して別々に有効または無効にできます。適切なマスクビットを 1 にセットすることで、対応する割り込みが有効になります。

次のイベントによって 2 つの割り込みが生成されます。

- ライン割り込み: プログラムされたライン到達時に生成されます。ライン割り込み位置は、LTDC\_LIPCR レジスタでプログラムします。
- レジスタ再ロード割り込み: 垂直ブランキング期間中にシャドウレジスタの再ロードが実施されると生成されます。
- FIFO アンダーラン割り込み: 空のレイヤ FIFO からピクセルが要求されると生成されます。
- 転送エラー割り込み: データ転送中に AHB バスエラーが発生すると生成されます。

これらの割り込みイベントは、下図に説明するように NVIC コントローラに接続されています。

図 114. 割り込みイベント

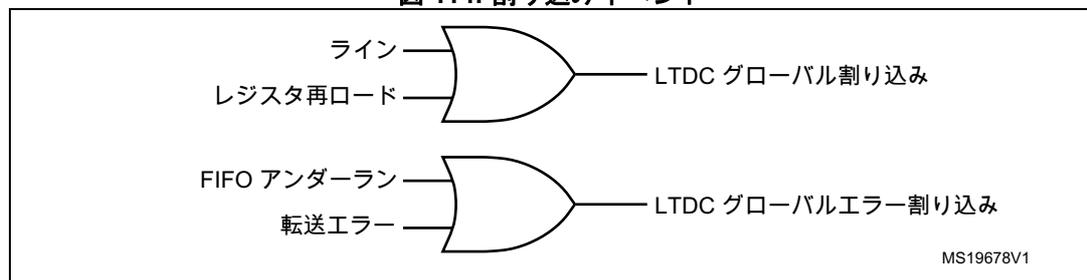


表 116. LTDC 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
ライン	LIF	LIE
レジスタ再ロード	RRIF	RRIEN
FIFO アンダーラン	FUDERRIF	FUDERRIE
転送エラー	TERRIF	TERRIE

## 18.6 LTDC プログラミング手順

- RCC レジスタで LTDC クロックを有効にします。
- パネルデータシートに従って必要なピクセルクロックを設定します。
- 同期タイミングの設定: [セクション 18.4.1: LTDC グローバル設定パラメータ](#)に説明されているように、パネルデータシートに従って VSYNC、HSYNC、垂直/水平バックポーチ、アクティブデータエリア、フロントポーチなどのタイミング。
- LTDC\_GCR レジスタで、同期信号とクロック極性を設定します。
- 必要であれば、LTDC\_BCCR レジスタでバックグラウンド色を設定します。
- LTDC\_IER および LTDC\_LIPCR レジスタで必要な割り込みを設定します。
- 以下をプログラムすることにより、レイヤ 1/2 のパラメータを設定します。
  - LTDC\_LxWHPCR および LTDC\_WVPCR レジスタ内のレイヤウィンドウの水平/垂直位置。レイヤウィンドウは、アクティブデータエリア内にある必要があります。
  - LTDC\_LxPFCR レジスタでピクセル入力フォーマットをプログラム
  - LTDC\_LxCFBAR レジスタでカラーフレームバッファ開始アドレスをプログラム
  - LTDC\_LxCFBLR レジスタでカラーフレームバッファのライン長およびピッチをプログラム
  - LTDC\_LxCFBLNR レジスタでカラーフレームバッファのライン数をプログラム
  - 必要であれば、LTDC\_LxCLUTWR レジスタで RGB 値とそのアドレスとともに CLUT をロードします。
  - 必要であれば、LTDC\_LxDCCR および LTDC\_LxBFCR レジスタでデフォルト色とブレンド係数をそれぞれ設定します。
- レイヤ 1/2 を有効にし、必要であれば LTDC\_LxCR レジスタで CLUT を有効にします。
- 必要であれば、LTDC\_GCR および LTDC\_LxCKCR レジスタでディザリングとカラーキーイングをそれぞれ有効にすることができます。また、動作中に有効にすることもできます。
- LTDC\_SRCR レジスタを介してシャドウレジスタをアクティブレジスタに再ロードします。
- LTDC\_GCR レジスタで LCD-TFT コントローラを有効にします。
- CLUT を除き、すべてのレイヤパラメータは動作中に変更できます。新しい設定は、直ちに再ロードするか、LTDC\_SRCR レジスタを設定することにより、垂直ブランキング期間中に再ロードする必要があります。

**注:** *すべてのレイヤのレジスタはシャドウイングされます。レジスタに書き込まれたら、再ロードが完了するまでは再変更してはなりません。したがって、同じレジスタへの新しい書き込みは、まだ再ロードされていないならばその前の設定に優先します。*

## 18.7 LTDC レジスタ

### 18.7.1 LTDC 同期サイズ設定レジスタ (LTDC\_SSCR)

このレジスタによって、水平同期ピクセル数から 1 を引いた数と、垂直同期ライン数から 1 を引いた数を指定します。設定例については [図 111](#) および [セクション 18.4 : LTDC のプログラム可能なパラメータ](#) を参照してください。

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:16 **HSW[11:0]** : 水平同期幅 (ピクセルクロック周期の単位)

このビットによって、水平同期ピクセル数から 1 を引いた数を指定します。

ビット 15:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **VSH[10:0]** : 垂直同期高さ (水平スキャンラインの単位)

このビットによって、垂直同期の高さから 1 を引いた数を指定します。これは、水平同期ライン数を表します。

## 18.7.2 LTDC バックポーチ設定レジスタ (LTDC\_BPCR)

このレジスタによって、水平同期およびバックポーチの累積ピクセル数から 1 を引いた数 (HSYNC 幅 + HBP- 1) と垂直同期およびバックポーチの累積ライン数から 1 を引いた数 (VSYNC 高さ + VBP- 1) を指定します。設定例については [図 111](#) および [セクション 18.4 : LTDC のプログラム可能なパラメータ](#) を参照してください。

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:16 **AHBP[11:0]** : 累積水平バックポーチ (ピクセルクロック周期の単位)

このビットにより、水平同期および水平バックポーチピクセル数から 1 を引いた数から成る累積水平バックポーチ幅を指定します。

水平バックポーチは、水平同期がインアクティブになってから次のスキャンラインのアクティブ表示エリアの開始までの期間です。

ビット 15:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **AVBP[10:0]** : 累積垂直バックポーチ (水平スキャンラインの単位)

このビットにより、垂直同期および垂直バックポーチライン数から 1 を引いた数から成る累積垂直バックポーチ幅を指定します。

垂直バックポーチは、フレーム開始時から次のフレームの最初のアクティブなスキャンライン開始までの水平スキャンライン数です。

## 18.7.3 LTDC アクティブ幅設定レジスタ (LTDC\_AWCR)

このレジスタによって、水平同期、バックポーチ、アクティブピクセルの累積数から 1 を引いた数 (**HSYNC 幅 + HBP + アクティブ幅 - 1**) と垂直同期、バックポーチライン、アクティブラインの累積数から 1 を引いた数 (**VSYNC 高さ + BVBP + アクティブ高さ - 1**) を指定します。設定例については [図 111](#) および [セクション 18.4 : LTDC のプログラム可能なパラメータ](#) を参照してください。

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:16 **AAW[11:0]** : 累積アクティブ幅 (ピクセルクロック周期の単位)

このビットにより、水平同期、水平バックポーチ、アクティブピクセル数から 1 を引いた数から成る累積アクティブ幅を定義します。

アクティブ幅は、パネルスキャンラインのアクティブ表示エリアにおけるピクセル数です。サポートされている最大アクティブ幅は 0x320 です。

ビット 15:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **AAH[10:0]** : 累積アクティブ高さ (水平スキャンラインの単位)

このビットにより、垂直同期、垂直バックポーチ、アクティブ高さのライン数から 1 を引いた数から成る累積高さを指定します。アクティブ高さは、パネル内のアクティブなラインの数です。サポートされている最大アクティブ高さは 0x258 です。

## 18.7.4 LTDC 全幅設定レジスタ (LTDC\_TWCR)

このレジスタによって、水平同期、バックポーチ、アクティブ、フロントポーチピクセルの累積数から 1 を引いた数 (HSYNC 幅 + HBP + アクティブ幅 + HFP - 1) と垂直同期、バックポーチライン、アクティブ、フロントラインの累積数から 1 を引いた数 (VSYNC 高さ + BVBP + アクティブ高さ + VFP - 1) を指定します。設定例については [図 111](#) および [セクション 18.4: LTDC のプログラム可能なパラメータ](#) を参照してください。

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:16 **TOTALW[11:0]** : 全幅 (ピクセルクロック周期の単位)

このビットにより、水平同期、水平バックポーチ、アクティブ幅、水平フロントポーチピクセル数から 1 を引いた数から成る累積全幅を指定します。

ビット 15:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **TOTALH[10:0]** : 全高 (水平スキャンラインの単位)

このビットにより、垂直同期、垂直バックポーチ、アクティブ高さ、垂直フロントポーチ高さのライン数から 1 を引いた数から成る累積高さを指定します。

## 18.7.5 LTDC グローバル制御レジスタ (LTDC\_GCR)

このレジスタにより、LCD-TFT コントローラのグローバル設定を指定します。

アドレスオフセット : 0x18

リセット値 : 0x0000 2220

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN
rw	rw	rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DRW[2:0]			Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	Res.	LTDCEN
	r	r	r		r	r	r		r	r	r				rw

ビット 31 **HSPOL** : 水平同期極性

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : 水平同期極性アクティブロー
- 1 : 水平同期極性アクティブハイ

ビット 30 **VSPOL** : 垂直同期極性

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : 垂直同期アクティブロー
- 1 : 垂直同期アクティブハイ

ビット 29 **DEPOL** : 非データイネーブル極性

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : 非データイネーブル極性アクティブロー
- 1 : 非データイネーブル極性アクティブハイ

ビット 28 **PCPOL** : ピクセルクロック極性

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : 入力ピクセルクロック
- 1 : 反転入力ピクセルクロック

ビット 27:17 予約済みであり、リセット値のままにしておかなければなりません。

ビット 16 **DEN** : デザリングイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : デザリングディセーブル
- 1 : デザリングイネーブル

ビット 15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:12 **DRW[2:0]** : デザ赤の幅

このビットはデザ赤のビットを返します。

ビット 11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:8 **DGW[2:0]** : デザ緑の幅

このビットはデザ緑のビットを返します。

ビット 7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6:4 **DBW[2:0]** : ディザ青の幅

このビットはディザ青のビットを返します。

ビット 3:1 予約済みであり、リセット値のままにしておかなければなりません。

ビット 0 **LTDCEN** : LCD-TFT コントローライネーブルビット

このビットは、ソフトウェアによってセット/クリアされます。

0 : LTDC ディセーブル

1 : LTDC イネーブル

## 18.7.6 LTDC シャドウ再ロード設定レジスタ (LTDC\_SRCR)

このレジスタにより、シャドウレジスタ値をアクティブレジスタに直ちに再ロードすることもできれば、垂直ブランキング期間中にすることもできます。LTDC\_L1CLUTWR および LTDC\_L2CLUTWR を除き、レイヤ 1 とレイヤ 2 のすべてのレジスタはシャドウレジスタです。

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **VBR** : 垂直ブランキングの再ロード

このビットは、ソフトウェアによってセットされ、再ロード後はハードウェアによってのみクリアされます。(いったんセットされるとレジスタの書き込みによってクリアすることはできません)

0 : 影響なし。

1 : シャドウレジスタは、垂直ブランキング期間中 (アクティブ表示エリア後の最初のライン開始時) に再ロードされます。

ビット 0 **IMR** : 即時再ロード

このビットは、ソフトウェアによってセットされ、再ロード後はハードウェアによってのみクリアされます。

0 : 影響なし。

1 : シャドウレジスタが直ちに再ロードされます。

**注 :** シャドウレジスタは、アクティブな値を読み出します。再ロードが完了するまでは、「古い」値が読み出されます。

## 18.7.7 LTDC バックグラウンド色設定レジスタ (LTDC\_BCCR)

このレジスタにより、バックグラウンド色 (RGB888) を指定します。

アドレスオフセット : 0x2C

**リセット値 : 0x0000 0000**

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:16 **BCRED[7:0]** : バックグラウンド色、赤の値  
このビットにより、バックグラウンドの赤の値を設定します。

ビット 15:8 **BCGREEN[7:0]** : バックグラウンド色、緑の値  
このビットにより、バックグラウンドの緑の値を設定します。

ビット 7:0 **BCBLUE[7:0]** : バックグラウンド色、青の値  
このビットにより、バックグラウンドの青の値を設定します。

## 18.7.8 LTDC 割り込みイネーブルレジスタ (LTDC\_IER)

このレジスタは、対応するビットを 1 にセットすることで、どのステータスフラグが割り込み要求を生成するかを決定します。

アドレスオフセット : 0x34

**リセット値 : 0x0000 0000**

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

ビット 31:4 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **RRIE** : レジスタ再ロード割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : レジスタ再ロード割り込みディセーブル

1 : レジスタ再ロード割り込みイネーブル

ビット 2 **TERRIE** : 転送エラー割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : 転送エラー割り込みディセーブル

1 : 転送エラー割り込みイネーブル

ビット 1 **FUIE** : FIFO アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : FIFO アンダーラン割り込みディセーブル

1 : FIFO アンダーラン割り込みイネーブル

ビット 0 **LIE** : ライン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : ライン割り込みディセーブル

1 : ライン割り込みイネーブル

## 18.7.9 LTDC 割り込みステータスレジスタ (LTDC\_ISR)

このレジスタは、割り込みステータスフラグを返します。

アドレスオフセット : 0x38

**リセット値 : 0x0000 0000**

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **RRIF** : レジスタ再ロード割り込みフラグ

0 : レジスタ再ロード割り込みは生成されません。

1 : 垂直ブランキング再ロード発生時に (およびアクティブエリア到達後の最初のラインで) レジスタ再ロード割り込みが生成されます。

ビット 2 **TERRIF** : 転送エラー割り込みフラグ

0 : 転送エラー割り込みは生成されません。

1 : バスエラー発生時に転送エラー割り込みが生成されます。

ビット 1 **FUIF** : FIFO アンダーラン割り込みフラグ

0 : FIFO アンダーラン割り込みは生成されません。

1 : レイヤ FIFO のいずれかが空で、FIFO からピクセルデータが読み出された場合、FIFO アンダーラン割り込みが生成されます。

ビット 0 **LIF** : ライン割り込みフラグ

0 : ライン割り込みは生成されません。

1 : プログラムされたライン到達時にライン割り込みが生成されます。

## 18.7.10 LTDC 割り込みクリアレジスタ (LTDC\_ICR)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **CRRIF** : レジスタ再ロード割り込みフラグをクリアします。

- 0 : 影響なし。
- 1 : LTDC\_ISR レジスタの RRIF フラグをクリアします。

ビット 2 **CTERRIF** : 転送エラー割り込みフラグをクリアします。

- 0 : 影響なし。
- 1 : LTDC\_ISR レジスタの TERRIF フラグをクリアします。

ビット 1 **CFUIF** : FIFO アンダーラン割り込みフラグをクリアします。

- 0 : 影響なし。
- 1 : LTDC\_ISR レジスタの FUDERRIF フラグをクリアします。

ビット 0 **CLIF** : ライン割り込みフラグをクリアします。

- 0 : 影響なし。
- 1 : LTDC\_ISR レジスタの LIF フラグをクリアします。

## 18.7.11 LTDC ライン割り込み位置設定レジスタ (LTDC\_LIPCR)

このレジスタにより、ライン割り込みの位置を指定します。プログラムされるライン値は、タイミングパラメータに依存します。☞ 111を参照してください。

アドレスオフセット : 0x40

リセット値 : 0x0000 0000

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

ビット 31:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **LIPOS[10:0]** : ライン割り込み位置

このビットにより、ライン割り込み位置を設定します。



## 18.7.12 LTDC 現在位置ステータスレジスタ (LTDC\_CPSR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

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

ビット 31:16 **CXPOS[15:0]** : 現在の X 位置  
このビットは、現在の X 位置を返します。

ビット 15:0 **CYPOS[15:0]** : 現在の Y 位置  
このビットは、現在の Y 位置を返します。

## 18.7.13 LTDC 現在表示ステータスレジスタ (LTDC\_CDSR)

このレジスタは、HSYNC、VSYNC、水平/垂直 DE 信号が制御する現在の表示フェーズのステータスを返します。

例 : 現在の表示フェーズが垂直同期であれば、VSYNCCS ビットがセットされます (アクティブハイ)。現在の表示フェーズが水平同期であれば、HSYNCCS ビットがアクティブハイになります。

アドレスオフセット : 0x48

リセット値 : 0x0000 000F

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **HSYNCS** : 水平同期表示ステータス  
0 : アクティブロー  
1 : アクティブハイ

ビット 2 **VSYNCS** : 垂直同期表示ステータス  
0 : アクティブロー  
1 : アクティブハイ

ビット 1 **HDES** : 水平データ有効表示ステータス  
0 : アクティブロー  
1 : アクティブハイ

ビット 0 **VDES** : 垂直データ有効表示ステータス  
 0 : アクティブロー  
 1 : アクティブハイ

**注 :** 返されるステータスは、**LTDC\_GCR** レジスタで設定されている極性には依存しません。代わりに、現在のアクティブ表示フェーズを返します。

## 18.7.14 LTDC レイヤ x 制御レジスタ (LTDC\_LxCR) (x=1..2)

アドレスオフセット :  $0x84 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ  $x = 1$  または  $2$

リセット値 : **0x0000 0000**

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

ビット 31:5 予約済みであり、リセット値のままにしておかなければなりません。

ビット 4 **CLUTEN** : カラールックアップテーブルイネーブル  
 このビットは、ソフトウェアによってセット/クリアされます。

0 : カラールックアップテーブル無効  
 1 : カラールックアップテーブル有効

CLUT は、L8、AL44、AL88 ピクセルフォーマットの場合にのみ意味を持ちます。[カラールックアップテーブル \(CLUT\) \(516 ページ\)](#) を参照してください。

ビット 3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **COLKEN** : カラーキーイングイネーブル  
 このビットは、ソフトウェアによってセット/クリアされます。

0 : カラーキーイング無効  
 1 : カラーキーイング有効

ビット 0 **LEN** : レイヤイネーブル  
 このビットは、ソフトウェアによってセット/クリアされます。

0 : レイヤ無効  
 1 : レイヤ有効

## 18.7.15 LTDC レイヤ x ウィンドウ水平位置設定レジスタ (LTDC\_LxWHPCR) (x=1..2)

このレジスタにより、レイヤ 1 または 2 ウィンドウの水平位置（最初と最後のピクセル）を指定します。

ラインの最初に見えるピクセルは、LTDC\_BPCR レジスタにおける **AHBP[10:0] ビット + 1** のプログラム値です。

ラインの最後に見えるピクセルは、LTDC\_AWCR レジスタにおける **AAW[10:0] ビット** のプログラム値です。この範囲内にあるすべての値が使用可能です。

アドレスオフセット：0x88 + 0x80 x (レイヤ x - 1)、レイヤ x = 1 または 2

**リセット値：0x0000 0000**

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

ビット 31:28 予約済みであり、リセット値のままにしておかなければなりません。

ビット 27:16 **WHSPPPOS[11:0]**：ウィンドウ水平停止位置

このビットにより、レイヤウィンドウにおけるラインの最後に見えるピクセルを設定します。

ビット 15:12 予約済みであり、リセット値のままにしておかなければなりません。

ビット 11:0 **WHSTPOS[11:0]**：ウィンドウ水平開始位置

このビットにより、レイヤウィンドウにおけるラインの最初に見えるピクセルを設定します。

例：

LTDC\_BPCR レジスタは 0x000E0005 (AHBP[11:0] は 0xE) に設定され、LTDC\_AWCR レジスタは 0x028E01E5 (AAW[11:0] は 0x28E) に設定されます。アクティブデータエリアで水平開始オフセットを 5 ピクセルにして、サイズ 630x460 のウィンドウにおける水平位置を設定するには、

- レイヤウィンドウの最初のピクセル：WHSTPOS[11:0] を 0x14 (0xE+1+0x5) にプログラムします。
- レイヤウィンドウの最後のピクセル：WHSPPPOS[11:0] を 0x28A にプログラムします。

## 18.7.16 LTDC レイヤ x ウィンドウ垂直位置設定レジスタ (LTDC\_LxWVPCR) (x=1..2)

このレジスタにより、レイヤ 1 または 2 ウィンドウの垂直位置 (最初と最後のライン) を指定します。

フレームの最初の見えるラインは、LTDC\_BPCR レジスタにおける AVBP[10:0] ビット + 1 プログラム値です。

フレームの最後の見えるラインは、LTDC\_AWCR レジスタにおける AAH[10:0] ビットのプログラム値です。この範囲内にあるすべての値が使用可能です。

アドレスオフセット :  $0x8C + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ  $x = 1$  または  $2$

**リセット値 : 0x0000 0000**

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

ビット 31:27 予約済みであり、リセット値のままにしておかなければなりません。

ビット 26:16 **WVSPPOS[10:0]** : ウィンドウ垂直停止位置  
このビットにより、レイヤウィンドウの最後の見えるラインを設定します。

ビット 15:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **WVSTPOS[10:0]** : ウィンドウ垂直開始位置  
このビットにより、レイヤウィンドウの最初の見えるラインを設定します。

例 :

LTDC\_BPCR レジスタは 0x000E0005 (AVBP[10:0] は 0x5) に設定され、LTDC\_AWCR レジスタは 0x028E01E5 (AAH[10:0] は 0x1E5) に設定されます。アクティブデータエリアで垂直開始オフセットを 8 ラインにして、サイズ 630x460 のウィンドウにおける垂直位置を設定するには、

- レイヤウィンドウの最初のライン : WVSTPOS[10:0] を 0xE (0x5 + 1 + 0x8) にプログラムします。
- レイヤウィンドウの最後のライン : WVSPPOS[10:0] を 0x1DA にプログラムします。

## 18.7.17 LTDC レイヤ x カラーキーイング設定レジスタ (LTDC\_LxCKCR) (x=1..2)

このレジスタにより、カラーキーイングによって使用されるカラーキー値 (RGB) を指定します。

アドレスオフセット :  $0x90 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤx = 1 または 2

リセット値 : 0x0000 0000

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

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:16 **CKRED[7:0]** : カラーキー赤の値

ビット 15:8 **CKGREEN[7:0]** : カラーキー緑の値

ビット 7:0 **CKBLUE[7:0]** : カラーキー青の値

## 18.7.18 LTDC レイヤ x ピクセルフォーマット設定レジスタ (LTDC\_LxPFCR) (x=1..2)

このレジスタにより、レイヤのフレームバッファ内に保存されるデータに使用するピクセルフォーマットを指定します。ピクセルデータはフレームバッファから読み出され、内部フォーマット 8888(ARGB) に変換されます。

アドレスオフセット :  $0x94 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤx = 1 または 2

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2:0 **PF[2:0]** : ピクセルフォーマット

このビットにより、ピクセルフォーマットを設定します。

000 : ARGB8888

001 : RGB888

010 : RGB565

011 : ARGB1555

100 : ARGB4444

101 : L8 (8 ビット輝度)

110 : AL44 (4 ビットアルファ、4 ビット輝度)

111 : AL88 (8 ビットアルファ、8 ビット輝度)

## 18.7.19 LTDC レイヤ x 定数アルファ設定レジスタ (LTDC\_LxCACR) (x=1..2)

このレジスタにより、アルファブレンディングで使用する定数アルファ値 (ハードウェアによって 255 で除される) を指定します。LTDC\_LxBFCR レジスタを参照してください。

アドレスオフセット :  $0x98 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ  $x = 1$  または  $2$

リセット値 : (レイヤ  $x - 1$ )  $0x0000\ 00FF$

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

ビット 31:8 予約済みであり、リセット値のままにしておかなければなりません。

ビット 7:0 **CONSTA[7:0]** : 定数アルファ

このビットにより、ブレンディングに使用する定数アルファを設定します。定数アルファは、ハードウェアによって 255 で除されます。

例 : プログラムされている定数アルファが  $0xFF$  であれば、定数アルファ値は  $255/255=1$  となります。

## 18.7.20 LTDC レイヤ x デフォルト色設定レジスタ (LTDC\_LxDCCR) (x=1..2)

このレジスタにより、フォーマット ARGB におけるレイヤのデフォルト色を指定します。このデフォルト色は、指定されたレイヤウィンドウ外や、レイヤが無効になっているときに使用されます。リセット値 0x00000000 は、透明な黒を表します。

アドレスオフセット : 0x9C + 0x80 x (レイヤx-1)、レイヤx = 1 または 2

リセット値 : 0x0000 0000

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

- ビット 31:24 **DCALPHA[7:0]** : デフォルト色アルファ  
このビットにより、デフォルトのアルファ値を設定します。
- ビット 23:16 **DCRED[7:0]** : デフォルト色赤  
このビットにより、デフォルト赤の値を設定します。
- ビット 15:8 **DCGREEN[7:0]** : デフォルト色緑  
このビットにより、デフォルト緑の値を設定します。
- ビット 7:0 **DCBLUE[7:0]** : デフォルト色青  
このビットにより、デフォルト青の値を設定します。

## 18.7.21 LTDC レイヤ x ブレンディング係数設定レジスタ (LTDC\_LxBF<sub>CR</sub>) (x=1..2)

このレジスタにより、ブレンディング係数 F1 および F2 を指定します。

基本的なブレンディングの計算式 :  $BC = BF1 \times C + BF2 \times Cs$

- BC = ブレンドされた色
- BF1 = ブレンディング係数 1
- C = 現在のレイヤの色
- BF2 = ブレンディング係数 2
- Cs = ブレンドされた下側レイヤの色

アドレスオフセット :  $0xA0 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ x = 1 または 2

リセット値 : 0x0000 0607

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
					rw	rw	rw						rw	rw	rw

ビット 31:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:8 **BF1[2:0]** : ブレンディング係数 1

このビットにより、ブレンディング係数 F1 を選択します。

- 000 : 予約済み
- 001 : 予約済み
- 010 : 予約済み
- 011 : 予約済み
- 100 : 定数アルファ
- 101 : 予約済み
- 110 : ピクセルアルファ x 定数アルファ
- 111 : 予約済み

ビット 7:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2:0 **BF2[2:0]** : ブレンディング係数 2

このビットにより、ブレンディング係数 F2 を選択します。

- 000 : 予約済み
- 001 : 予約済み
- 010 : 予約済み
- 011 : 予約済み
- 100 : 予約済み
- 101 : 1 - 定数アルファ
- 110 : 予約済み
- 111 : 1 - (ピクセルアルファ x 定数アルファ)



注： 定数アルファ値は、LxCACR レジスタのプログラム値をハードウェアにより 255 で除したものです。  
 例：レイヤ1 のみ有効、BF1 を定数アルファに設定  
 BF2 を“1 - 定数アルファ”に設定  
 定数アルファ：LxCACR レジスタでプログラムされている定数アルファは 240 (0xF0) です。従って、定数アルファ値は  $240/255 = 0.94$  となります。  
 C：現在のレイヤ色は 128  
 Cs：バックグラウンド色は 48  
 レイヤ1 はバックグラウンド色とブレンドされます。  
 $BC = \text{定数アルファ} \times C + (1 - \text{定数アルファ}) \times Cs = 0.94 \times 128 + (1 - 0.94) \times 48 = 123$

## 18.7.22 LTDC レイヤ x カラーフレームバッファアドレスレジスタ (LTDC\_LxCFBAR) (x=1..2)

このレジスタにより、カラーフレームバッファ開始アドレスを指定します。このアドレスは、フレームバッファでレイヤの左上ピクセルのピクセルデータが保存されているアドレスを示している必要があります。

アドレスオフセット：0xAC + 0x80 x (レイヤx - 1)、レイヤx = 1 または 2

リセット値：0x0000 0000

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

ビット 31:0 **CFBADD[31:0]**：カラーフレームバッファ開始アドレス  
 このビットにより、カラーフレームバッファ開始アドレスを指定します。

## 18.7.23 LTDC レイヤ x カラーフレームバッファ長レジスタ (LTDC\_LxCFBLR) (x=1..2)

このレジスタにより、カラーフレームバッファのライン長とピッチを指定します。

アドレスオフセット :  $0xB0 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ  $x = 1$  または  $2$

リセット値 :  $0x0000\ 0000$

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

ビット 31:29 予約済みであり、リセット値のままにしておかなければなりません。

ビット 28:16 **CFBP[17:0]** : カラーフレームバッファピッチ (バイト単位)

このビットにより、あるピクセルのラインの開始から次のラインの開始までの増分であるピッチをバイト単位で指定します。

ビット 15:13 予約済みであり、リセット値のままにしておかなければなりません。

ビット 12:0 **CFBLL[12:0]** : カラーフレームバッファのライン長

このビットにより、バイト単位のピクセルライン長 + 3 を指定します。

ライン長は次のようにして計算します。アクティブハイの幅 x ピクセル当たりのバイト数 + 3

例 :

- フォーマット RGB565 (ピクセル当たり 2 バイト)、幅 256 ピクセル (ライン当たりの総バイト数は  $256 \times 2 = 512$  バイト)、ピッチがライン長であるフレームバッファでは、このレジスタに  $0x02000203$  という値を書き込みます。
- フォーマット RGB888 (ピクセル当たり 3 バイト)、幅 320 ピクセル (ライン当たりの総バイト数は  $320 \times 3 = 960$  バイト)、ピッチがライン長であるフレームバッファでは、このレジスタに  $0x03C003C3$  という値を書き込みます。

## 18.7.24 LTDC レイヤ x カラーフレームバッファライン数レジスタ (LTDC\_LxWVPCR) (x=1..2)

このレジスタにより、カラーフレームバッファ内のライン数を指定します。

アドレスオフセット :  $0xB4 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ  $x = 1$  または  $2$

リセット値 :  $0x0000\ 0000$

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

ビット 31:11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10:0 **CFBLNBR[10:0]** : フレームバッファライン数

このビットにより、アクティブハイ幅に相当するフレームバッファ内ライン数を指定します。

**注 :** このライン数とライン長の設定により、各レイヤに対してフレームごとのデータフェッチ数を指定します。必要なバイト数よりも少なく設定されている場合、FIFO アンダーラン割り込みが有効になっていれば、該当する割り込みが生成されます。

一方、開始アドレスとピッチ設定により、メモリ内の各ラインの正しい開始位置を指定します。

## 18.7.25 LTDC レイヤ x CLUT 書き込みレジスタ (LTDC\_LxCLUTWR) (x=1..2)

このレジスタにより、CLUT アドレスと RGB 値を指定します。

アドレスオフセット :  $0xC4 + 0x80 \times (\text{レイヤ}x - 1)$ 、レイヤ  $x = 1$  または 2

リセット値 :  $0x0000\ 0000$

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

ビット 31:24 **CLUTADD[7:0]** : CLUT アドレス  
このビットにより、各 RGB 値の CLUT アドレス (CLUT 内のカラー位置) を指定します。

ビット 23:16 **RED[7:0]** : 赤の値  
このビットにより、赤の値を設定します。

ビット 15:8 **GREEN[7:0]** : 緑の値  
このビットにより、緑の値を設定します。

ビット 7:0 **BLUE[7:0]** : 青の値  
このビットにより、青の値を設定します。

**注 :** *CLUT 書き込みレジスタは、ブランキング期間中またはレイヤが無効になっている場合にのみ設定してください。CLUT は、LTDC\_LxCR レジスタで有効または無効にできます。*

*CLUT は、L8、AL44、AL88 ピクセルフォーマットの場合にのみ意味を持ちます。*

## 18.7.26 LTDC レジスタマップ

次の表は LTDC レジスタの一覧です。LTDC レジスタベースアドレスについては、レジスタ境界アドレス表を参照してください。

表 117. LTDC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0008	LTDC_SSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C	LTDC_BPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0010	LTDC_AWCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	LTDC_TWCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	LTDC_GCR	HSPOL	VSPOL	DEPOL	PCPOL	Res.	DEN	Res.																									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	LTDC_SRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	LTDC_BCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	LTDC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0038	LTDC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	LTDC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	LTDC_LIPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	LTDC_CPSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	LTDC_CDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	LTDC_L1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 117. LTDC レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0088	LTDC_L1WHPCR	Res.	Res.	Res.	Res.													Res.	Res.	Res.	Res.												
	リセット値																																
0x008C	LTDC_L1WVPCR	Res.	Res.	Res.	Res.	Res.												Res.	Res.	Res.	Res.												
	リセット値																																
0x0090	LTDC_L1CKCR	Res.																															
	リセット値																																
0x0094	LTDC_L1PFCR	Res.																															
	リセット値																																
0x0098	LTDC_L1CACR	Res.																															
	リセット値																																
0x009C	LTDC_L1DCCR																																
	リセット値																																
0x00A0	LTDC_L1BFCCR	Res.																															
	リセット値																																
0x00AC	LTDC_L1CFBAR																																
	リセット値																																
0x00B0	LTDC_L1CFBLR	Res.																															
	リセット値																																
0x00B4	LTDC_L1CFBLNR	Res.																															
	リセット値																																
0x00C4	LTDC_L1CLUTWR																																
	リセット値																																
0x0104	LTDC_L2CR	Res.																															
	リセット値																																
0x0108	LTDC_L2WHPCR	Res.	Res.	Res.	Res.													Res.	Res.	Res.	Res.												
	リセット値																																



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

表 117. LTDC レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x010C	LTDC_L2WVPCR	Res.	Res.	Res.	Res.	Res.																												
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0110	LTDC_L2CKCR	Res.																																
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0114	LTDC_L2PFCR	Res.																																
	リセット値																															0	0	0
0x0118	LTDC_L2CACR	Res.																																
	リセット値																																	
0x011C	LTDC_L2DCCR	Res.																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0120	LTDC_L2BFCCR	Res.																																
	リセット値																																	
0x012C	LTDC_L2CFBAR	Res.																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0130	LTDC_L2CFBLR	Res.																																
	リセット値																																	
0x0134	LTDC_L2CFBLNR	Res.																																
	リセット値																																	
0x0144	LTDC_L2CLUTWR	Res.																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

## 19 乱数発生器 (RNG)

### 19.1 概要

RNG プロセッサは、連続アナログ雑音に基づく乱数発生器であり、読み出し時にホストに対し 32 ビットの乱数値を返します。

本 RNG は、成功率 99% で FIPS PUB (Federal Information Processing Standard Publication : 連邦情報処理規格出版物) 140-2 の試験に合格しています (2001 年 10 月 10 日)。

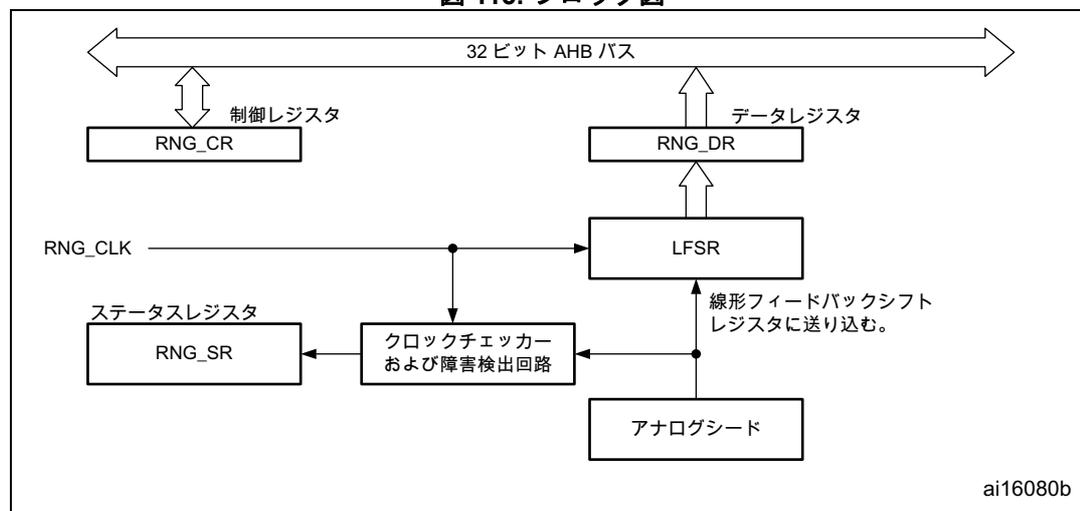
### 19.2 RNG の主な機能

- アナログ発生器によって生成される 32 ビットの乱数を出力します。
- 2 つの連続する乱数の間に必要な時間は、RNG\_CLK クロック信号 40 周期分です。
- RNG エントロピーを監視して異常な動作 (同一値の生成や値の同一シーケンスの生成など) を知らせます。
- 本機能を無効にして消費電力を低減することができます。

### 19.3 RNG の機能説明

図 115 に RNG ブロック図を示します。

図 115. ブロック図



1. RNG クロック (RNG\_CLK) ソースの詳細については、[セクション 5: リセットおよびクロック制御 \(RCC\)](#) を参照してください。

乱数発生器には、アナログ回路を実装しています。この回路は、32 ビットの乱数を生成するため、線形フィードバックシフトレジスタ (RNG\_LFSR) に送り込むシード (「元」) を生成します。

アナログ回路は複数のリングオシレータで構成されており、リングオシレータの出力の排他的論理和をとってこのシードを生成します。RNG\_LFSR は専用クロック (RNG\_CLK) によって一定周波数で駆動されるため、乱数の特性としては HCLK 周波数と無関係になります。十分な数のシードが RNG\_LFSR に取り込まれると、RNG\_LFSR の内容はデータレジスタ (RNG\_DR) に転送されます。

並行して、アナログシードと専用の RNG\_CLK クロックがモニターされます。シードで異常なシーケンスが発生したり、RNG\_CLK クロックの周波数が低すぎる場合、ステータスビット (RNG\_SR レジスタ内) に示されます。エラーが検出された場合、割り込みを生成することができます。

### 19.3.1 動作

RNG を作動させるには、次のステップに従います。

1. 必要に応じて割り込みを有効にします (RNG\_CR レジスタの IE ビットをセットします)。乱数の準備ができたとき、またはエラーが発生したとき、割り込みが生成されます。
2. RNG\_CR レジスタの RNGEN ビットをセットして、乱数発生を有効にします。これによってアナログ部分、RNG\_LFSR およびエラー検出回路が作動します。
3. 割り込みのたびに、エラーが発生していないこと (RNG\_SR レジスタの SEIS ビットおよび CEIS ビットが“0”)、乱数の準備ができていないこと (RNG\_SR レジスタの DRDY ビットが“1”)を確認します。RNG\_DR レジスタの内容が読み出せるようになります。

FIPS PUB 140-2 の要件に従い、RNGEN ビット設定後最初に生成された乱数は使用すべきではありませんが、次に生成される乱数との比較のため保存されます。後に生成される各乱数は、それぞれ前に生成された値と比較する必要があります。比較された 2 つの値が同一の場合、試験は不合格となります (連続乱数発生器試験)。

### 19.3.2 エラー管理

#### CEIS ビットが“1”として読み出された場合 (クロックエラー)

クロックエラーの場合、RNG\_CLK クロックが正しくないため、RNG はそれ以降、乱数を発生することはできません。RNG クロックが供給されるようにクロックコントローラが正しく設定されていることを確認し、CEIS ビットをクリアします。CEIS ビットが“0”になると、RNG が動作できるようになります。クロックエラーはその前に発生した乱数には影響しないため、RNG\_DR レジスタの内容は使用できます。

#### SEIS ビットが“1”として読み出された場合 (シードエラー)

シードエラーの場合、SEIS ビットが“1”の間は乱数発生が中断します。RNG\_DR レジスタにおける値が利用可能な場合であっても、エントロピーが十分でない可能性があるため使用することはできません。

ここでは、SEIS ビットをクリアした後、RNGEN ビットをクリアおよびセットし、RNG を再初期化および再起動します。

## 19.4 RNG レジスタ

RNG は、制御レジスタ、データレジスタ、ステータスレジスタと連動します。これらのレジスタは、ワード (32 ビット) 単位でアクセスする必要があります。

### 19.4.1 RNG 制御レジスタ (RNG\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:4 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **IE** : 割り込みイネーブル

0 : RNG 割り込みは無効です。

1 : RNG 割り込みは有効です。RNG\_SR レジスタで DRDY = 1、SEIS = 1 または CEIS = 1 になると、割り込みは直ちに保留となります。

ビット 2 **RNGEN** : 乱数発生器イネーブル

0 : 乱数発生器は無効です。

1 : 乱数発生器は有効です。

ビット 1:0 予約済みであり、リセット値のままにしておかなければなりません。

### 19.4.2 RNG ステータスレジスタ (RNG\_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY								
									rc_w0	rc_w0			r	r	r

ビット 31:7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6 **SEIS** : シードエラー割り込みステータス

このビットは SECS と同時にセットされ、“0”を書き込むとクリアされます。

0 : 障害シーケンスは検出されませんでした。

1 : 次の障害シーケンスのうち 1 つが検出されました。

- 64 個以上のビットが連続して同じ値 (0 または 1)

- 0 と 1 の繰り返し (0101010101...01) が 32 回以上連続

RNG\_CR レジスタで IE = 1 である場合、割り込みが保留となります。



ビット 5 **CEIS** : クロックエラー割り込みステータス

このビットは CECS と同時にセットされ、“0”を書き込むとクリアされます。

0 : RNG\_CLK クロックが正しく検出されました。

1 : RNG\_CLK が正しく検出されませんでした ( $f_{RNG\_CLK} < f_{HCLK}/16$ )。

RNG\_CR レジスタで IE = 1 である場合、割り込みが保留となります。

ビット 4:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2 **SECS** : シードエラーの現在のステータス

0 : 障害のあるシーケンスは現在のところ検出されていません。SEIS ビットがセットされている場合、これは障害のあるシーケンスが検出されたが、回復していることを意味します。

1 : 次の障害シーケンスのうち 1 つが検出されました。

- 64 個以上のビットが連続して同じ値 (0 または 1)

- 0 と 1 の繰り返し (0101010101...01) が 32 回以上連続

ビット 1 **CECS** : クロックエラーの現在のステータス

0 : RNG\_CLK クロックが正しく検出されました。CEIS ビットがセットされている場合、これはクロックエラーが検出されたが、回復していることを意味します。

1 : RNG\_CLK が正しく検出されませんでした ( $f_{RNG\_CLK} < f_{HCLK}/16$ )。

ビット 0 **DRDY** : データレディ

0 : RNG\_DR レジスタがまだ有効でなく、乱数データは利用できません。

1 : RNG\_DR レジスタに有効な乱数データが入っています。

**注 :** RNG\_CR レジスタで IE = 1 である場合、割り込みが保留となります。

RNG\_DR レジスタがいったん読み出されると、新しい有効な値が計算されるまでこのビットは 0 に戻ります。

## 19.4.3 RNG データレジスタ (RNG\_DR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

RNG\_DR レジスタは読み出し専用レジスタであり、読み出されると 32 ビットの乱数値を返します。いったん読み出されると、最長で RNG\_CLK クロック 40 周期後に、このレジスタは新しい乱数値を出力します。ソフトウェアは、RNDATA 値を読み出す前に DRDY ビットがセットされていることを確認する必要があります。

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

ビット 31:0 **RNDATA** : 乱数データ

32 ビットの乱数データです。

## 19.4.4 RNG レジスタマップ

表 118 に、RNG レジスタマップとリセット値を示します。

表 118. RNG レジスタマップとリセット値

オフセット	レジスタ名 リセット値	レジスタサイズ																																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RNG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																	
0x04	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	IE	SECS	Res.	Res.
	リセット値																									0	0			0	0	0	0	0
0x08	RNG_DR	RNDATA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 20 暗号プロセッサ (CRYP)

このセクションは、特に指定がない限り、STM32F756xx デバイス全体に適用されます。

### 20.1 CRYP の概要

暗号プロセッサは、DES、トリプル DES、または AES (128、192、または256) アルゴリズムを使用したデータの暗号化と復号化の両方に使用できます。次の規格に完全に準拠して実装されています。

- 連邦情報処理規格公報 (FIPS : Federal Information Processing Standards Publication) (FIPS PUB 46-3, 1999 October 25) によって規定されているデータ暗号化標準 (DES : Data Encryption Standard) およびトリプル DES (TDES)。米国規格協会 (ANSI : American National Standards Institute) の X9.52 規格に準拠しています。
- 連邦情報処理規格公報 (FIPS PUB 197, 2001 November 26) によって規定されている高度暗号化標準 (AES : Advanced Encryption Standard)。

CRYP プロセッサは、ECB (電子コードブック) モードまたは CBC (暗号ブロック連鎖) モードで、DES およびトリプル DES アルゴリズムを使用してデータを暗号化/復号化します。

CRYP ペリフェラルは 32 ビットの AHB2 ペリフェラルです。受信データと処理済みデータの DMA 転送をサポートし、入出力 FIFO があります (深さ 各 8 ワード)。

### 20.2 CRYP の主な機能

- AES、DES、トリプル DES 暗号化および復号化処理に適しています。
- AES
  - ECB、CBC、CTR、CCM、および GCM 連結アルゴリズムをサポート
  - 128、192、256 ビットキーをサポート
  - CBC、CTR、CCM、GCM モードで使用する 4 × 32 ビットの初期化ベクタ (IV)

表 119. 各 128 ビットブロックの処理に必要なサイクル数

アルゴリズム/ キー長	ECB	CBC	CTR	GCM				CCM			
				初期化	ヘッダ	ペイロード	タグ	初期化	ヘッダ	ペイロード	タグ
128 ビット	14	14	14	24	10	14	14	12	14	25	14
192b	16	16	16	28	10	16	16	14	16	29	16
256b	18	18	18	32	10	18	18	16	18	33	18

- DES/トリプル DES
  - 単純 DES アルゴリズムの直接実装 (シングルキー K1 を使用)
  - ECB および CBC 連結アルゴリズムをサポート
  - 64、128、192 ビットキー (パリティを含む) をサポート
  - CBC モードで使用する 2 × 32 ビットの初期化ベクタ (IV)
  - DES では、HCLK 16 サイクルで 1 つの 64 ビットブロックを処理
  - トリプル DES では、HCLK 48 サイクルで 1 つの 64 ビットブロックを処理
- DES/トリプル DES および AES に対して共通
  - 入出力 FIFO (深さ各 8 ワード、幅 32 ビットで DES 4 ブロック分または AES 2 ブロック分に相当)
  - 2 つのチャネル (受信データ用と処理済みデータ用) を使用したダイレクトメモリアクセス (DMA) による自動データフロー制御
  - 1、8、16、32 ビットデータをサポートするデータスワッピングロジック

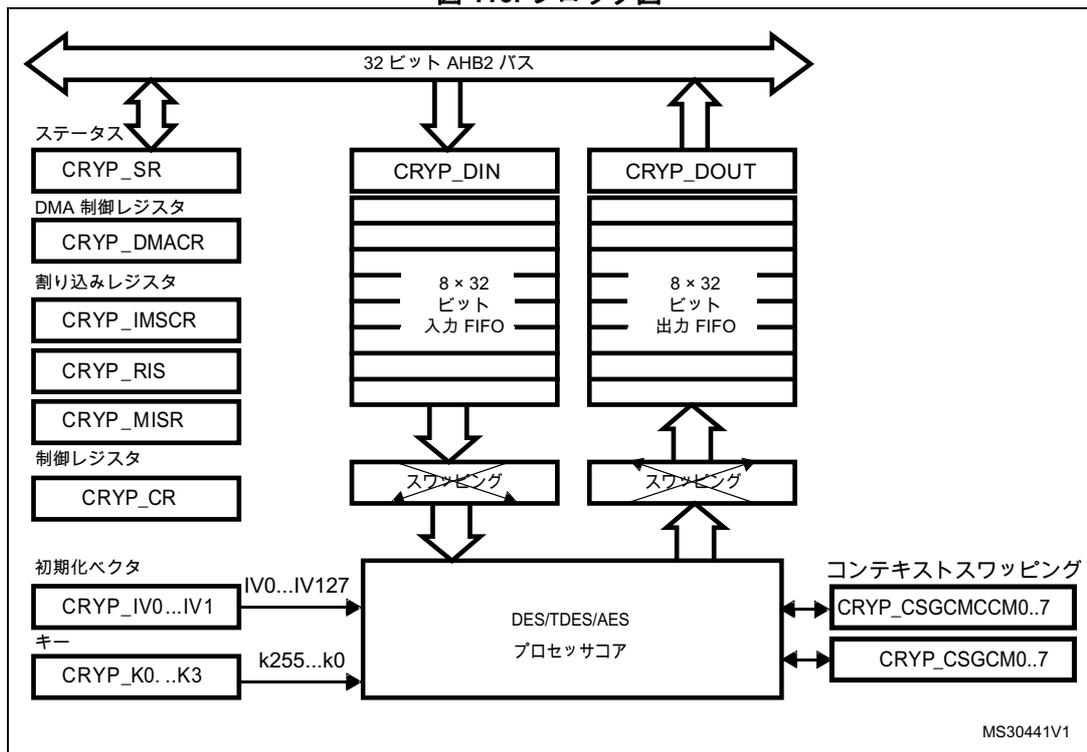
## 20.3 CRYP の機能詳細

この暗号プロセッサは、トリプル DES (DES もサポートするトリプルDES) コアと AES 暗号コアを実装しています。これらのコアの詳細については、[セクション 20.3.1](#) および [セクション 20.3.2](#) で説明します。

トリプルDES および AES アルゴリズムがブロック暗号を使用するため、不完全な入力データブロックの場合、暗号化前にパディングを行う必要があります (データ列の後端に余分なビットを付加する必要があります)。復号後は、パディング部分を破棄する必要があります。ハードウェアではパディング処理を管理しないので、ソフトウェアで処理する必要があります。

図 116 に、暗号プロセッサのブロック図を示します。

図 116. ブロック図



### 20.3.1 DES／トリプル DES 暗号コア

DES/トリプル DES 暗号コアは、3つのコンポーネントで構成されています。

- DES アルゴリズム (DEA)
- 複数のキー (DES アルゴリズムに 1 つ、トリプル DES アルゴリズムに 1 から 3 つ)
- 初期化ベクタ (CBC モードで使用)

トリプル DES での基本的な処理は、次のとおりです。入力ブロックが DEA で読み出され、最初のキーである K1 (トリプル DES モードでは K0 は使用しません) を使って暗号化されます。その後、2つ目のキーである K2 を使って出力が復号化され、3つ目のキーである K3 を使って暗号化されます。キーは使用されるアルゴリズムによって異なります。

- DES モード : キー = [K1]
- トリプル DES モード : キー = [K3 K2 K1]

ここで  $Kx=[KxR KxL]$ 、R = 右、L = 左 です。

実装されるモードに従い、結果として得られる出力ブロックを使用して暗号文を計算します。

中間 DEA ステージの出力が暗号境界の外に明かされることは決してないことに注意してください。

トリプルDES では、3 種類のキーイングオプションが可能です。

- 3 つの独立キー

最初のオプションでは、すべてのキーが独立、すなわち K1、K2、K3 が独立していると規定しています。FIPS PUB 46-3 – 1999 (および ANSI X9.52 – 1998) では、このオプションをキーイングオプション 1、トリプルDES を 3 キートリプルDES としています。

- 2 つの独立キー

2 番目のオプションでは、K1 と K2 が独立しており、K3 が K1 に等しい、すなわち K1 と K2 が独立し、K3 = K1 であると規定しています。FIPS PUB 46-3 – 1999 (および ANSI X9.52 – 1998) では、この 2 つ目のオプションをキーイングオプション 2、トリプルDES を 2 キートリプルDES としています。

- 3 つの同一キー

3 番目のオプションでは、K1、K2、K3 が等しい、すなわち K1 = K2 = K3 であると規定しています。FIPS PUB 46-3 – 1999 (および ANSI X9.52 – 1998) では、この 3 番目のオプションをキーイングオプション 3 としています。この「1 キー」トリプルDES は、シングル DES と同等です。

FIPS PUB 46-3 – 1999 (および ANSI X9.52-1998) では、TDEA (トリプル DES アルゴリズム) が提供するトリプル DES-ECB 暗号化、トリプル DES-ECB 復号化、トリプル DES-CBC 暗号化、トリプル DES-CBC 復号化の 4 つの動作モードに関する処理について詳細に説明しています。

このリファレンスマニュアルでは、各モードについての簡単な説明としています。

## DES およびトリプルDES 電子コードブック (DES/トリプルDES-ECB) モード

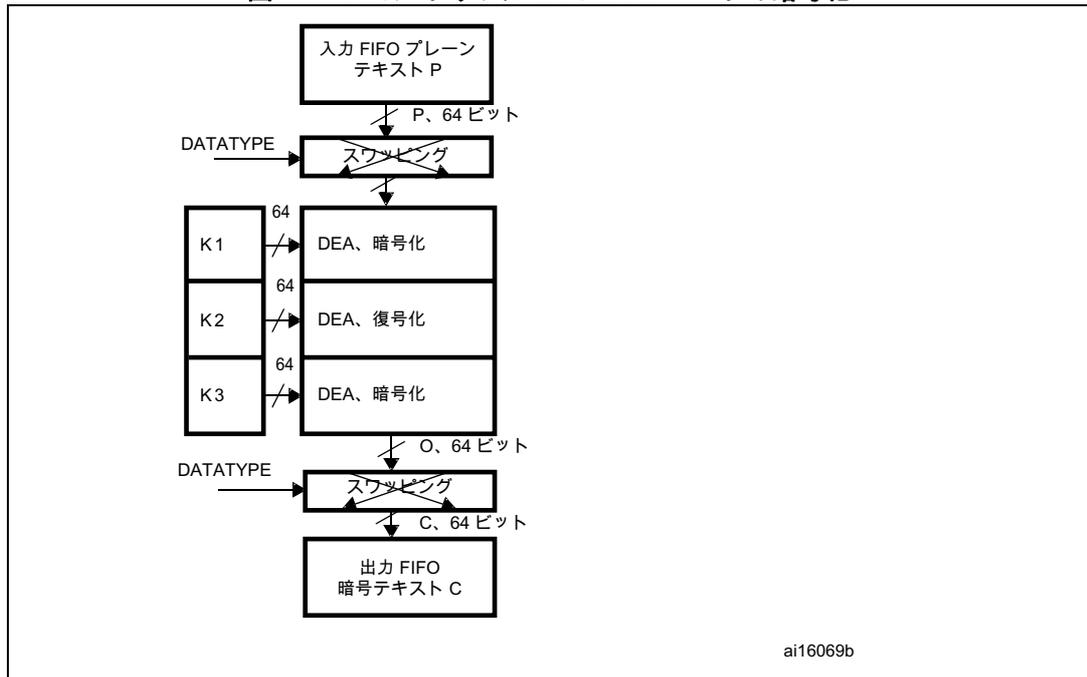
- DES/トリプルDES-ECB モードの暗号化

[図 117](#) に、DES およびトリプル DES 電子コードブック (DES/トリプル DES-ECB) モードにおける暗号化の図を示します。ビット/バイト/ハーフワードのスワッピング後は、64 ビットの平文データブロック (P) を入力ブロック (I) として使用します ([セクション 20.3.3: データ型 \(567 ページ\)](#) を参照)。入力ブロックは、K1 を使った暗号状態で DEA を通じて処理されます。この処理の出力は、DEA の入力に直接フィードバックされます。ここでは、K2 を使った復号状態で DES が行われます。この処理の出力は、DEA の入力に直接フィードバックされます。ここでは、K3 を使った暗号状態で DES が行われます。ビット/バイト/ハーフワードのスワッピング後は、その結果の 64 ビットの出力ブロック (O) が暗号文 (C) として使用され、出力 FIFO に入ります。

- DES/トリプルDES-ECB モードの復号化

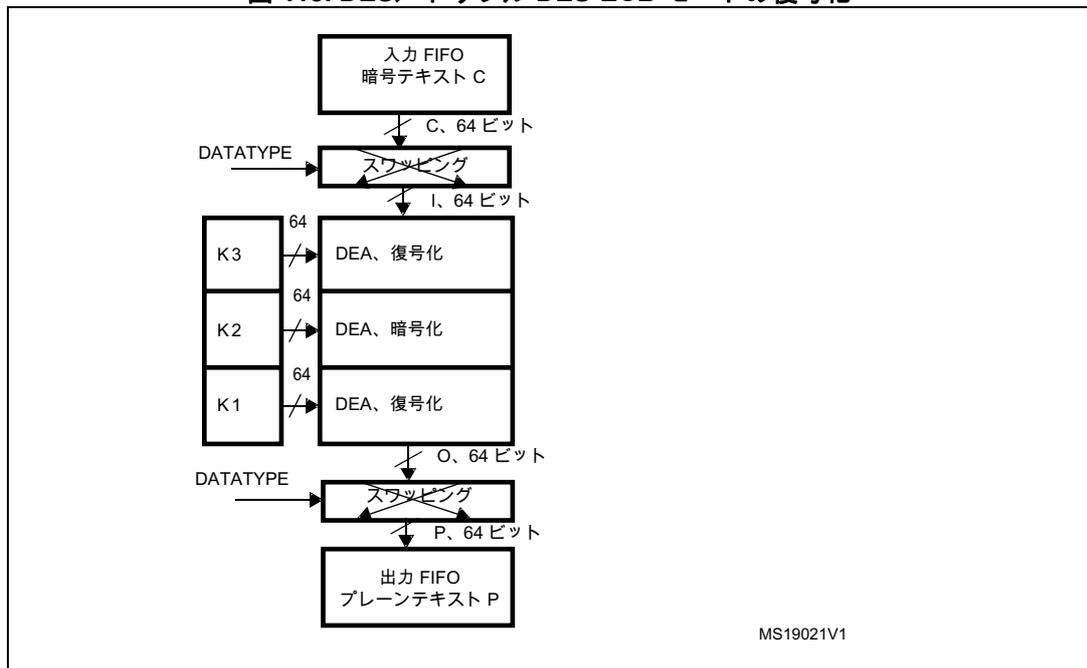
[図 118](#) に、DES/トリプル DES-ECB の復号化を示します。ビット/バイト/ハーフワードのスワッピング後は、64 ビットの暗号文ブロック (C) を入力ブロック (I) として使用します。このキーイングシーケンスは、暗号処理で使用したものに対し、逆になっています。入力ブロックは、K3 を使った復号状態で DEA を通じて処理されます。この処理の出力は、DEA の入力に直接フィードバックされます。ここでは、K2 を使った暗号状態で DES が行われます。新しい結果は、DEA の入力に直接されます。ここでは、K1 を使った復号状態で DES が実施されます。ビット/バイト/ハーフワードのスワッピング後は、その結果の 64 ビットの出力ブロック (O) が平文 (P) を生成します。

図 117. DES/トリプル DES-ECB モードの暗号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、P: 明文

図 118. DES/トリプル DES-ECB モードの復号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、P: 明文

## DES およびトリプル DES 暗号ブロック連鎖 (DES/トリプル DES-CBC) モード

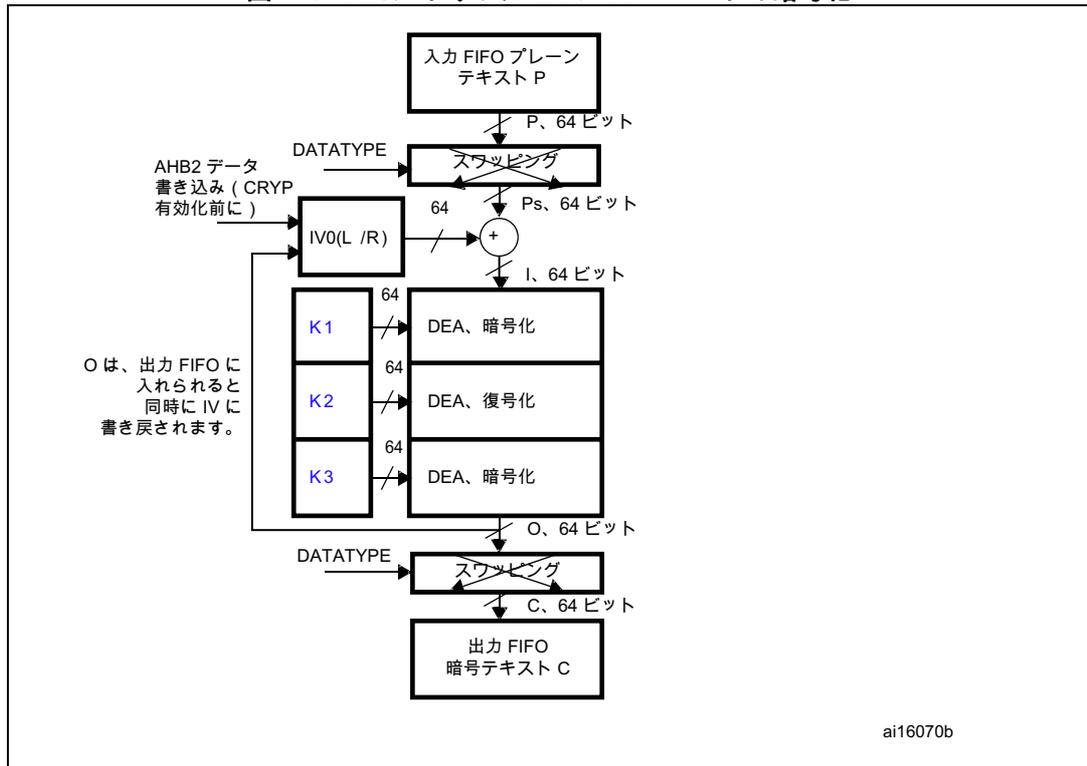
- DES/トリプルDES-CBC モードの暗号化

[図 119](#) に、DES およびトリプル DES 暗号ブロック連鎖 (DES/トリプル DES-CBC) モードの暗号化について図を示します。このモードは、まず平文のメッセージを 64 ビットのデータブロックに分割することから開始します。TCBC 暗号化では、ビット/バイト/ハーフワードのスワッピング後 ([セクション 20.3.3 : データ型 \(567 ページ\)](#) を参照) に取得した最初の入力ブロック ( $I_1$ ) は、最初の平文データブロック ( $P_1$ ) と 64 ビットの初期化ベクタ  $IV$  ( $I_1 = IV \oplus P_1$ ) の排他的論理和をとって形成します。入力ブロックは、 $K_1$  を使った暗号状態で DEA を通じて処理されます。この処理の出力は、DEA の入力に直接フィードバックされます。このとき  $K_2$  を使った復号状態で DES が行われます。この処理の出力は、DEA の入力に直接フィードバックされます。このとき  $K_3$  を使った暗号状態で DES が行われます。その結果の 64 ビットの入力ブロック ( $O_1$ ) は暗号文 ( $C_1$ ) として直接使用されます。つまり、 $C_1 = O_1$  となります。この最初の暗号文ブロックは、その後 2 番目の平文データブロックと排他的論理和をとって 2 番目の入力ブロック ( $I_2 = (C_1 \oplus P_2)$ ) を生成します。 $I_2$  と  $P_2$  が今度は 2 番目のブロックを参照していることに注意してください。2 番目の入力ブロックは TDEA で処理され、2 番目の暗号文ブロックを生成します。この暗号化処理は、メッセージ内の最後の平文ブロックが暗号化されるまで次の暗号ブロックと平文ブロックを連結し続けます。メッセージが整数のデータブロックで構成されていない場合は、最後の不完全なデータブロックがアプリケーションで指定されている方法で暗号化されます。

- DES/トリプルDES-CBC モードの復号化

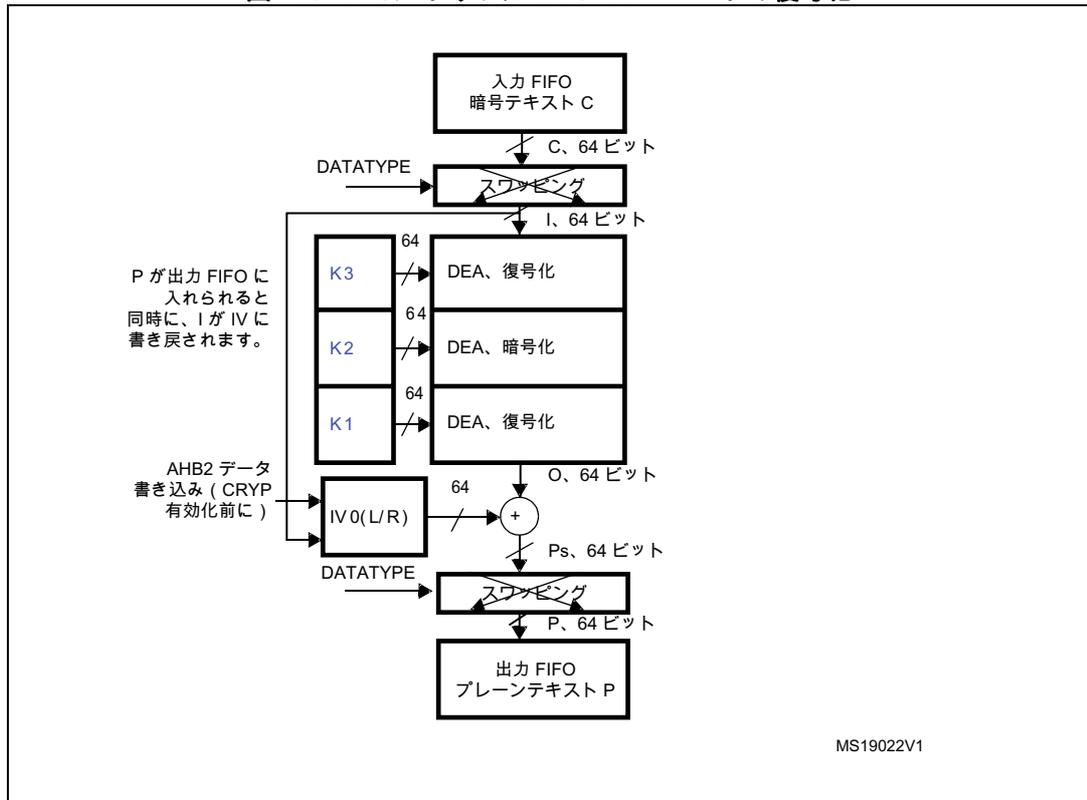
DES/トリプル DES-CBC 復号化 ([図 120](#) を参照) では、最初の暗号文ブロック ( $C_1$ ) をそのまま入力ブロック ( $I_1$ ) として使用します。キーイングシーケンスは、暗号処理で使用したものに対し、逆になっています。入力ブロックは、 $K_3$  を使った復号状態で DEA を通じて処理されます。この処理の出力は、DEA の入力に直接供給されます。このとき、 $K_2$  を使った暗号状態で DES が処理されます。これによって生じる値は、DEA の入力に直接供給されます。このとき、 $K_1$  を使った復号状態で DES が処理されます。その結果の出力ブロックと  $IV$  (暗号化で使用したのと同じである必要があります) の排他的論理和をとって最初の平文ブロック ( $P_1 = O_1 \oplus IV$ ) を生成します。その後 2 番目の暗号文ブロックが次の入力ブロックとして使用され、TDEA で処理されます。その結果の出力ブロックと最初の暗号文ブロックの排他的論理和をとって 2 番目の平文データブロック ( $P_2 = O_2 \oplus C_1$ ) を生成します。(  $P_2$  と  $O_2$  が 2 番目のデータブロックを参照していることに注意してください。) TCBC 復号化処理は、最後の完全な暗号文ブロックが復号化されるまで同様に続きます。不完全なデータブロックを表す暗号文は、アプリケーションで指定された方法で復号化する必要があります。

図 119. DES/トリプル DES-CBC モードの暗号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、Ps: スワッピング前 (復号化時) またはスワッピング後 (暗号化時) の平文、P: 平文、IV: 初期化ベクタ

図 120. DES/トリプル DES-CBC モードの復号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、Ps: スワッピング前 (復号化時) またはスワッピング後 (暗号化時) の平文、P: 平文、IV: 初期化ベクタ

## 20.3.2 AES 暗号コア

AES 暗号コアは、3 つのコンポーネントで構成されています。

- AES 暗号 (AEA : Advanced Encryption Algorithm)
- 複数キー
- 初期化ベクタまたはノンス

AES は、128、192、256 ビットの 3 種類の長さのキーの他、使用する操作モードによってゼロまたは 128 ビットの初期化ベクタ (IV) を使用します。

AES に含まれる基本的な処理は、次のとおりです。まず、128 ビットの入力ブロックが入力 FIFO から読み出され、AEA に送信されて、キー (K0...3) を使って暗号化されます。キー形式は、キーサイズによって変わります。

- キーサイズ = 128 の場合 : キー = [K3 K2]
- キーサイズ = 192 の場合 : キー = [K3 K2 K1]
- キーサイズ = 256 の場合 : キー = [K3 K2 K1 K0]

ここで  $K_x = [K_xR \ K_xL]$ 、R = 右、L = 左

実装されるモードに従い、結果として得られる出力ブロックを使用して暗号文を計算します。

FIPS PUB 197 (November 26, 2001) では、AES コアが供給する AES-ECB 暗号化、AES-ECB 復号化、AES-CBC 暗号化、および AES-CBC 復号化の 4 つの操作モードに関する処理について詳述しています。このリファレンスマニュアルでは、各モードについて簡単な説明のみ提供しています。

## AES 電子コードブック (AES-ECB) モード

- AES-ECB モードの暗号化

図 121 に、AES 電子コードブック (AES-ECB) モードの暗号化の図を示します。

AES-ECB 暗号化では、ビット/バイト/ハーフワードのスイッチング後は、128 ビットの明文データブロック (P) を入力ブロック (I) として使用します (セクション 20.3.3 : データ型 (567 ページ) を参照)。入力ブロックは、128、192、256 ビットキーを使った暗号ステートで AEA を通じて処理されます。ビット/バイト/ハーフワードのスイッチング後は、その結果の 128 ビットの入力ブロック (O) が暗号文 (C) として使用されます。その後出力 FIFO に入れます。

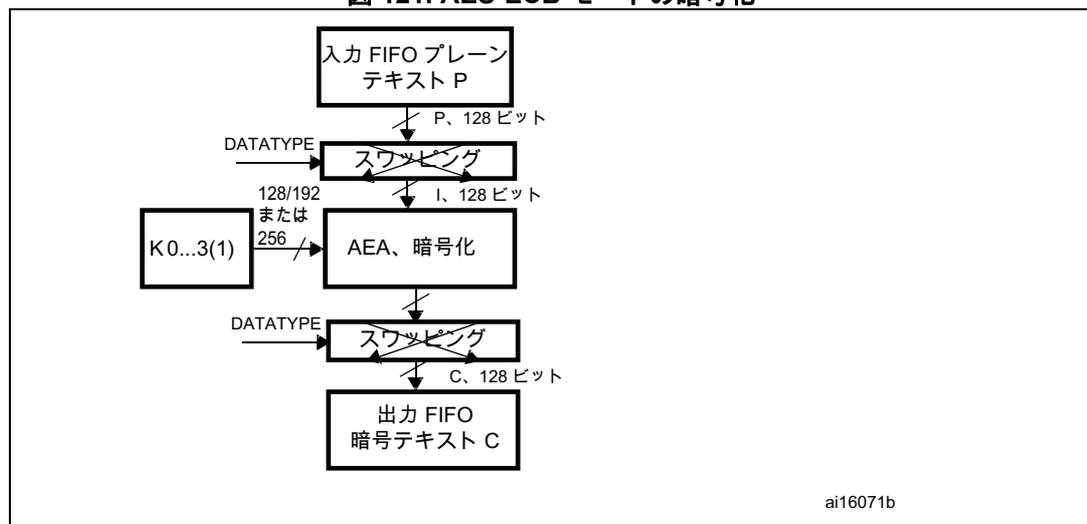
- AES-ECB モードの復号化

図 122 に、AES 電子コードブック (AES-ECB) モードの暗号化の図を示します。

ECB モードで AES 復号化を実施するには、最終ラウンドのキーを回収し、それを暗号文の復号化の最初のラウンドのキーとして使用して、秘密キーを準備する必要があります (暗号化のための完全なキースケジュールを実行する必要があります)。この準備機能は、AES コアで計算されます。キーを準備する方法の詳細については、セクション 20.3.6 : 暗号化または復号化を実施する手順を参照してください。

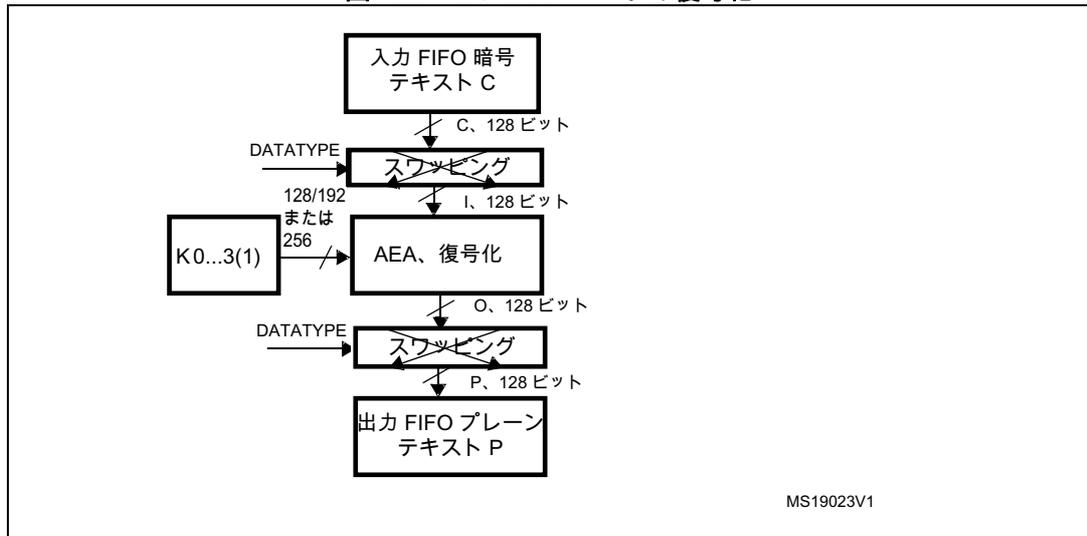
AES-ECB 復号化では、ビット/バイト/ハーフワードのスイッチング後、128 ビットの暗号文ブロック (C) が入力ブロック (I) として使用されます。このキーイングシーケンスは、暗号処理のものに対し、逆になっています。ビット/バイトまたはハーフワードのスイッチング後は、その結果の 128 ビットの入力ブロック (O) が明文 (P) を生成します。

図 121. AES-ECB モードの暗号化



1. K : キー、C : 暗号文、I : 入力ブロック、O : 出力ブロック、P : 明文
2. キーサイズ = 128 の場合 : キー = [K3 K2]  
 キーサイズ = 192 の場合 : キー = [K3 K2 K1]  
 キーサイズ = 256 の場合 : キー = [K3 K2 K1 K0]

図 122. AES-ECB モードの復号化



MS19023V1

1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、P: 明文
2. キーサイズ = 128 => キー = [K3 K2]  
 キーサイズ = 192 => キー = [K3 K2 K1]  
 キーサイズ = 256 => キー = [K3 K2 K1 K0]

## AES 暗号ブロック連鎖 (AES-CBC) モード

### ● AES-CBC モードの暗号化

AES 暗号ブロック連鎖 (AES-CBC) モードの復号化を図 123 に示します。

AES-CBC 暗号化では、ビット/バイト/ハーフワードのスワッピング後 (セクション 20.3.3: データ型 (567 ページ) を参照) に取得した最初の入力ブロック (I<sub>1</sub>) は、最初の明文データブロック (P<sub>1</sub>) と 128 ビットの初期化ベクタ IV (I<sub>1</sub> = IV Å P<sub>1</sub>) の排他的論理和をとって形成します。入力ブロックは、128、192、256 ビットキー (K<sub>0</sub>...K<sub>3</sub>) を使った暗号状態で AEA を通じて処理されます。その結果の 128 ビットの出力ブロック (O<sub>1</sub>) は暗号文 (C<sub>1</sub>) として直接使用されます。つまり、C<sub>1</sub> = O<sub>1</sub> となります。この最初の暗号文ブロックは、その後 2 番目の明文データブロックと排他的論理和をとって 2 番目の入力ブロック (I<sub>2</sub>) = (C<sub>1</sub> Å P<sub>2</sub>) を生成します。I<sub>2</sub> と P<sub>2</sub> が今度は 2 番目のブロックを参照していることに注意してください。2 番目の入力ブロックは AEA で処理され、2 番目の暗号文ブロックを生成します。この暗号化処理は、メッセージ内の最後の明文ブロックが暗号化されるまで次の暗号ブロックと明文ブロックを連結し続けます。メッセージが整数のデータブロックで構成されていない場合は、最後の不完全なデータブロックがアプリケーションで指定されている方法で暗号化されます。

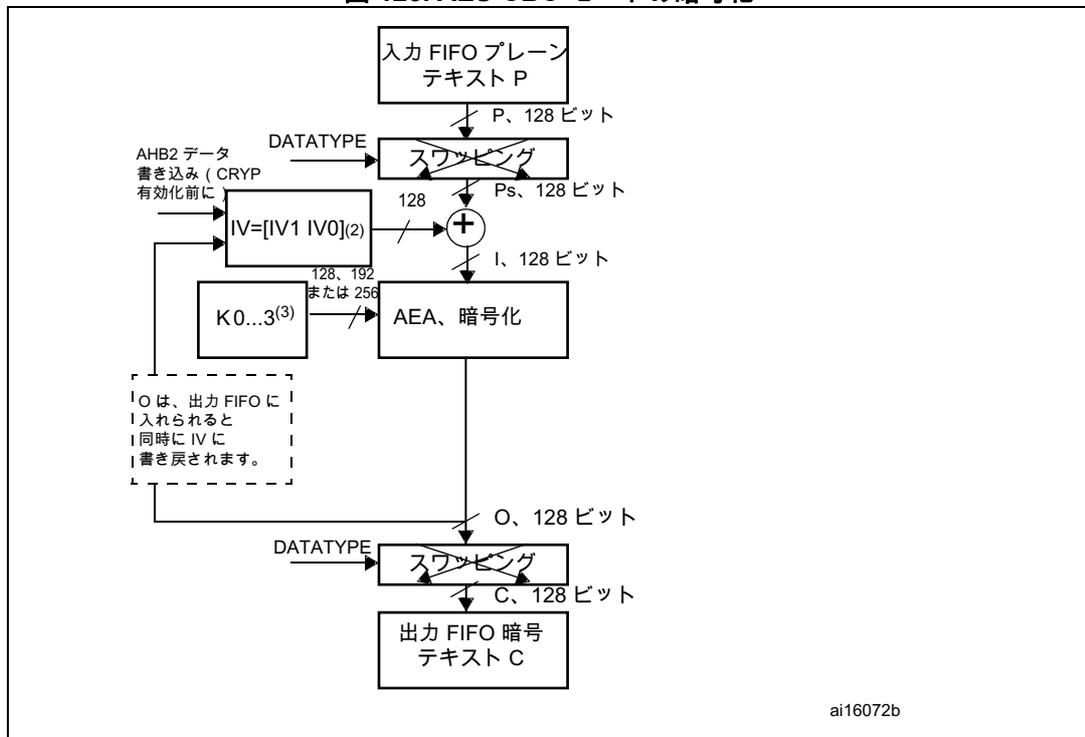
ECB モードの場合と同様に、CBC モードで AES 復号化を実施するには、秘密キーを準備する必要があります。キーを準備する方法の詳細については、セクション 20.3.6: 暗号化または復号化を実施する手順 (572 ページ) を参照してください。

### ● AES-CBC モードの復号化

AES-CBC 復号化 (図 124 を参照) では、最初の 128 ビット暗号文ブロック (C<sub>1</sub>) をそのまま入力ブロック (I<sub>1</sub>) として使用します。入力ブロックは、128、192、256 ビットキーを使った復号状態で AEA を通じて処理されます。その結果の出力ブロックと 128 ビットの初期化ベクタ IV (暗号化で使用したのと同じである必要があります) の排他的論理和をとって最初の明文ブロック (P<sub>1</sub> = O<sub>1</sub> Å IV) を生成します。その後 2 番目の暗号文ブロックが次の入力ブロックとして使用され、AEA で処理されます。その結果の出力ブロックと最初の暗号文ブロックの排他的論理和をとって 2 番目の明文データブロック (P<sub>2</sub> = O<sub>2</sub> Å C<sub>1</sub>) を生成します。(P<sub>2</sub> と O<sub>2</sub> が 2 番目のデータブロックを参照していることに注意してください。) AES-CBC 復号化処理は、最

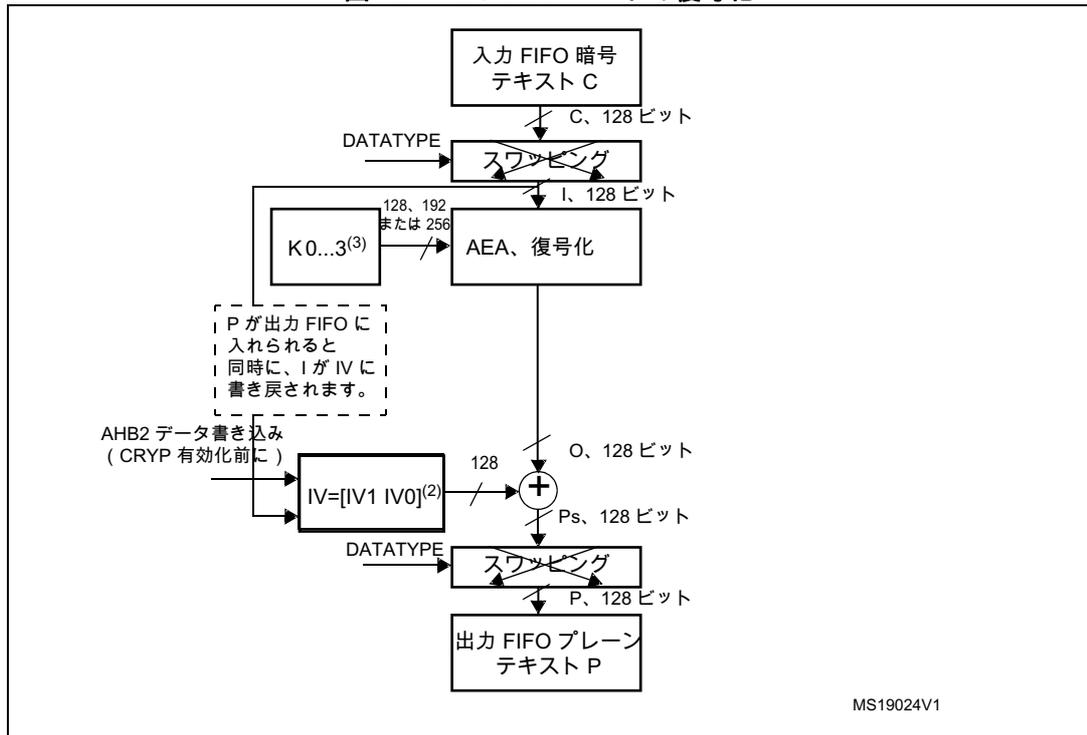
後の完全な暗号文ブロックが復号化されるまでこのように続きます。不完全なデータブロックを表す暗号文は、アプリケーションで指定された方法で復号化する必要があります。

図 123. AES-CBC モードの暗号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、Ps: スワッピング前 (復号化時) またはスワッピング後 (暗号化時) の明文、P: 明文、IV: 初期化ベクタ
2.  $IVx=[IVxR\ IVxL]$ , R = 右、L = 左
3. キーサイズ = 128 => キー = [K3 K2]  
 キーサイズ = 192 => キー = [K3 K2 K1]  
 キーサイズ = 256 => キー = [K3 K2 K1 K0]

図 124. AES-CBC モードの復号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、Ps: スワッピング前 (復号化時) またはスワッピング後 (暗号化時) の平文、P: 平文、IV: 初期化ベクタ
2.  $IVx=[IVxR\ IVxL]$ , R = 右、L = 左
3. キーサイズ = 128 => キー = [K3 K2]  
 キーサイズ = 192 => キー = [K3 K2 K1]  
 キーサイズ = 256 => キー = [K3 K2 K1 K0]

## AES カウンタ (AES-CTR) モード

AES カウンタモードでは、AES ブロックをキーストリームジェネレータとして使用します。生成されたキーは、その後平文との排他的論理和をとって暗号化されます。2つの操作は全く同じであるため、別のCTR暗号化/復号化を論じても意味がありません。

実際に、以下のような場合を考えてみます。

- 平文:  $P[0], P[1], \dots, P[n]$  (各 128 ビット)
- キー K を使用 (サイズは問わない)
- 初期カウンタブロック (ICB とするが、CBC の IV と機能は全く同じ)

暗号は次のように計算されます。

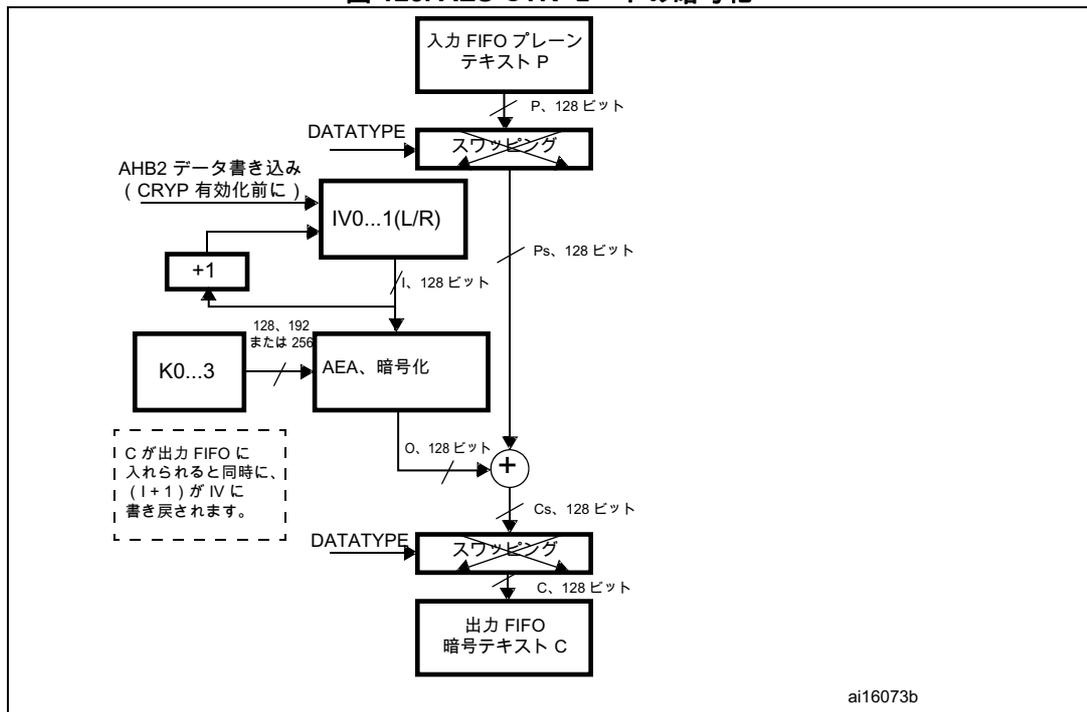
$C[i] = \text{enck}(iv[i]) \text{ xor } P[i]$ 、ここで:

$iv[0] = \text{ICB}$  であり  $iv[i+1] = \text{func}(iv[i])$ 、ここで func は前の iv ブロックに適用される更新関数です。func は基本的には iv ブロックを構成するフィールドの 1 つの増加分です。

復号化の ICB は暗号化の ICB と同じであるため、復号化で生成されるキーストリームは暗号化で生成されるキーストリームと同じです。その後、暗号文とキーストリームの排他的論理和をとって元の平文を取り出します。したがって復号化操作は、暗号化操作と全く同じように動作します。

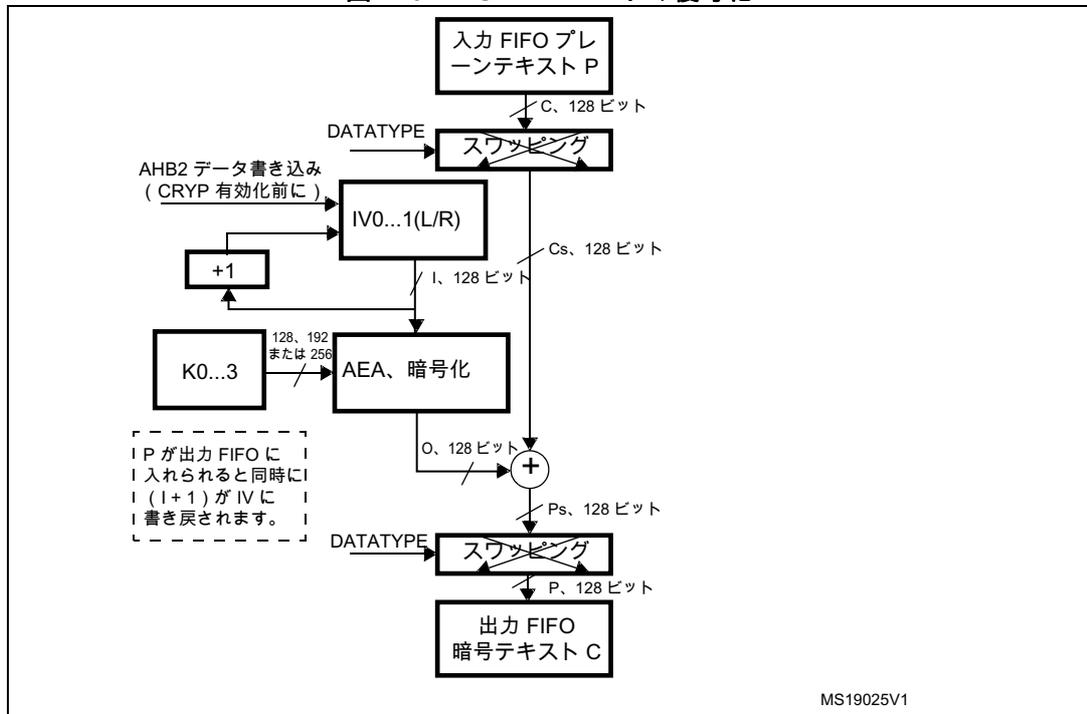
図 125 および 図 126 に、AES-CTR 暗号化と復号化を、それぞれ図示します。

図 125. AES-CTR モードの暗号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、Ps: スワッピング前 (復号化時) またはスワッピング後 (暗号化時) の明文、Cs: スワッピング後 (復号化時) またはスワッピング前 (暗号化時) の暗号化テキスト、P: 明文、IV: 初期化ベクタ

図 126. AES-CTR モードの復号化



1. K: キー、C: 暗号文、I: 入力ブロック、O: 出力ブロック、Ps: スワッピング前 (復号化時) またはスワッピング後 (暗号化時) の明文、Cs: スワッピング後 (復号化時) またはスワッピング前 (暗号化時) の暗号化テキスト、P: 明文、IV: 初期化ベクタ

図 127 に、規格 [2] で定義されている IV ブロックの構造を示します。3 つの異なるフィールドで構成されています。

図 127. カウンタモードでの初期カウンタブロック構造



- ノンスは 1 回だけ使用できる 32 ビットの値です。異なる通信には、それぞれ新しいノンスを割り当てる必要があります。
- 初期化ベクタ (IV) は 64 ビットの値であり、規格では特定のキーに対して特定の値が 1 度だけ使用されるように暗号器が IV を選択する必要があると規定しています。
- カウンタは 32 ビットのビッグエンディアンの整数であり、ブロックが暗号化されるたびに増加します。カウンタの初期値は“1”に設定します。

ブロックは最下位の 32 ビットを増加させますが、他の (最上位の) 96 ビットは変化しません。

## AES ガロア/カウンタモード (GCM)

AES ガロア/カウンタモード (GCM) では、平文を暗号化して認証し、対応する暗号文およびタグ (メッセージ認証コードまたはメッセージの整合性チェックとしても知られる) を生成することができます。このアルゴリズムは、AES カウンタモードに基づいており、機密性を確保しています。固定の有限フィールドを超えて乗算器を使用しタグを生成します。このアルゴリズムの開始時には初期化ベクタが必要です。

処理するメッセージは、2 つの部分に分割されます。

- ヘッダ (追加認証データとしても知られる) : 認証はされますが保護はされないデータです (パケットルーティングのための情報など)。
- ペイロード (平文または暗号文としても知られる) : それ自体が認証され、暗号化されるメッセージです。

**注 :** **ヘッダはペイロードに先行する必要があるため、両者を一緒に混在させることはできません。**

GCM 規格は、メッセージの最後でヘッダのサイズ (64 ビット) およびペイロードのサイズ (64 ビット) で構成される特定の 128 ビットのブロックを渡すことを要求しています。計算中は、ヘッダブロックをペイロードブロックと区別する必要があります。

GCM モードでは、暗号化/復号化の実施に 4 つのステップが必要です。

## 1. GCM 初期フェーズ

この最初のステップでは、HASH キーが計算され、すべてのブロックの処理に使用されるように内部に保存されます。以下の手順に従うことを推奨します。

- a) CRYP\_CR レジスタの CRYPEN ビットをクリアして暗号プロセッサが無効になっていることを確認します。
- b) CRYP\_CR で ALGOMODE ビットを“01000”にプログラムして GCM 連結モードを選択します。
- c) CRYP\_CR で GCM\_CCMPH ビットを“00”に設定して GCM 初期フェーズを開始します。
- d) CRYP\_KEYRx でキーレジスタ (128、192、256 ビット) とともに初期化ベクタ (IV) を初期化します。
- e) CRYPEN ビットを“1”にセットして HASH キーの計算を開始します。
- f) CRYPEN ビットが“0”にクリアされるのを待ってから次のフェーズに移ります。
- g) CRYPEN ビットを“1”にセットします。

## 2. GCM ヘッダフェーズ

このステップは、GCM 初期フェーズの後に実施する必要があります。

- h) CRYP\_CR で GCM\_CCMPH ビットを“01”に設定してヘッダフェーズの開始を示します。
- i) ヘッダデータを書き込みます。利用可能な方法は 3 つあります。
  - 32 ビットのブロックごとにデータを CRYP\_DIN レジスタにプログラムし、IFNF フラグを使用して入力 FIFO がデータを受信できるかどうかを判断します。ヘッダのサイズは 128 ビット (4 ワード) の倍数である必要があります。
  - 8 ワードのブロックごとに CRYP\_DIN レジスタにデータをプログラムし、IFEM フラグを使用して入力 FIFO がデータを受信できるかどうかを判断します (IFEM=“1”)。ヘッダのサイズは 128 ビット (4 ワード) の倍数である必要があります。
  - DMA を使用します。
- j) すべてのヘッダデータが供給されたら、CRYP\_SR レジスタで BUSY ビットがクリアされるまで待ちます。

## 3. GCM ペイロードフェーズ (暗号化/復号化)

このステップは、GCM ヘッダフェーズの後に実施する必要があります。

- k) CRYP\_CR レジスタで GCM\_CCMPH を“10”に設定します。
- l) CRYP\_CR で ALGODIR ビットを使用してアルゴリズムの方向 (暗号化または復号化) を選択します。
- m) CRYP\_DIN レジスタにペイロードメッセージをプログラムし、IFNF フラグを使用して入力 FIFO がデータを受信できるかどうかを判断します。または、8 ワードのブロック毎に CRYP\_DIN レジスタにデータをプログラムし、IFEM フラグを使用して入力 FIFO がデータを受信できるかどうかを判断することもできます (IFEM=“1”)。同時に、CRYP\_DOUT

レジスタの OFNE/OFFU フラグを監視して出力 FIFO が空でないかどうかをチェックできません。

- n) すべてのペイロードブロックが暗号化または復号化されるまで前のステップを繰り返します。または、DMA を使用することもできます。

#### 4. GCM 最終フェーズ

このステップでは、認証タグを生成します。

- o) CRYP\_CR で GCM\_CCMPH[1:0] を“11”に設定します。
- p) CRYP\_DIN レジスタに入力を 4 回書き込みます。入力には、ペイロードのビット数 (64 ビット) に連結されたヘッダのビット数 (64 ビット) が含まれている必要があります。
- q) CRYP\_SR レジスタで OFNE フラグ (FIFO 出力が空でない) が“1”にセットされるまで待ちます。
- r) CRYP\_DOUT レジスタを 4 回読み出します。この出力が認証タグに相当します。
- s) 暗号プロセッサを無効にします (CRYP\_CR の CRYPEN ビット=“0”)

**注:** 復号化を実施するときは、最初にキーを計算する必要はありません。復号化の最後に、生成されたタグをメッセージとともに渡された予測タグと比較します。さらに、ALGODIR ビット (アルゴリズムの方向) を“1”にセットする必要があります。

ヘッダフェーズからタグフェーズに移行するときに、CRYP プロセッサを無効/有効にする必要はありません。

### AES ガロアメッセージ認証コード (GMAC)

暗号プロセッサも、平文を認証するように GMAC をサポートしています。GCM アルゴリズムと固定の有界フィールドを超えて乗算器を使用し、対応するタグを生成します。

このアルゴリズムの開始時には初期化ベクタが必要です。

実際は、GMAC アルゴリズムはヘッダのみで構成されているメッセージに適用される GCM アルゴリズムに相当します。結果として、ペイロードフェーズは必要ありません。

### AES を組み合わせた暗号マシン (CCM)

CCM アルゴリズムでは、平文を暗号化して認証することもできれば、対応する暗号文およびタグを生成することもできます (メッセージ認証コードまたはメッセージの整合性チェックとしても知られる)。このアルゴリズムは、AES カウンタモードに基づいており、機密性を確保しています。AES CBC モードを使用して 128 ビットのタグを生成します。

CCM 規格 (RFC 3610 Counter with CBC-MAC (CCM), September 2003) では、最初の認証ブロック (規格では B0 と呼ぶ) に対して特定の暗号化規則を定義しています。具体的に言うと、最初のブロックにはフラグ、ノンス、ペイロード長 (単位: バイト) が含まれています。CCM 規格は、暗号化/復号化に対して A またはカウンタと呼ばれる別の形式を指定しています。カウンタはペイロードフェーズで増加し、32 個の最下位ビットがタグ生成中に“1”に初期化されます (CCM 規格では A0 パケットと呼ぶ)。

**注:** B0 パケットのフォーマット処理はハードウェアが実行するわけではありません。これはソフトウェアで処理します。

GCM アルゴリズムについていうと、処理するメッセージは 2 つの部分に分割されます。

- ヘッダ (追加認証データとしても知られる): 認証はされますが保護はされないデータです (パケットルーティングのための情報など)。
- ペイロード (平文または暗号文としても知られる): それ自体が認証され、暗号化されるメッセージです。

**注:** ヘッダ部分はペイロードに先行する必要があり、両者を一緒に混在させることはできません。



CCM モードでは、暗号化／復号化の実施に 4 つのステップが必要です。

## 1. CCM 初期フェーズ

この最初のステップでは、CCM メッセージの B0 パケット (最初のパケット) が CRYP\_DIN レジスタにプログラムされます。このフェーズでは、CRYP\_DOUT レジスタには出力データは一切格納されていません。

次のシーケンスに従う必要があります。

- a) CRYP\_CR レジスタの CRYPEN ビットをクリアして暗号プロセッサが無効になっていることを確認します。
- b) CRYP\_CR レジスタで ALGOMODE ビットを“01001”にプログラムして、CCM 連結モードを選択します。
- c) CRYP\_CR で GCM\_CCMPH ビットを“00”に設定して CCM 初期フェーズを開始します。
- d) CRYP\_KEYRx でキーレジスタ (128、192、256 ビット) とともに初期化ベクタ (IV) を初期化します。
- e) CRYP\_CR で CRYPEN ビットを“1”にセットします。
- f) B0 パケットを入力データレジスタにプログラムします。
- g) CRYPEN ビットがクリアされるのを待ってから次のフェーズに移ります。
- h) CRYPEN を“1”にセットします。

## 2. CCM ヘッダフェーズ

このステップは、CCM 初期フェーズの後に実施する必要があります。シーケンスは、暗号化と復号化のシーケンスと同一です。

このフェーズでは、CRYP\_DOUT レジスタには出力データは一切格納されていません。

このフェーズは、追加認証データがなければスキップできます。

次のシーケンスに従う必要があります。

- i) CRYP\_CR で GCM\_CCMPH ビットを“01”に設定してヘッダフェーズの開始を示します。
- j) 利用可能な方法は 3 つあります。
  - 32 ビットのブロックごとにヘッダデータを CRYP\_DIN レジスタにプログラムし、IFNF フラグを使用して入力 FIFO がデータを受信できるかどうかを判断します。ヘッダのサイズは 128 ビット (4 ワード) の倍数である必要があります。
  - 8 ワードのブロックごとに CRYP\_DIN レジスタにヘッダデータをプログラムし、IFEM フラグを使用して入力 FIFO がデータを受信できるかどうかを判断します (IFEM=“1”)。ヘッダのサイズは 128 ビット (4 ワード) の倍数である必要があります。
  - DMA を使用します。

**注：** **最初のブロック B1 は、ヘッダ長でフォーマットする必要があります。このタスクはソフトウェアで処理します。**

- k) すべてのヘッダデータが供給されたら、BUSY フラグがクリアされるまで待ちます。

## 3. CCM ペイロードフェーズ (暗号化／復号化)

このステップは、CCM ヘッダフェーズの後に実施する必要があります。このフェーズでは、暗号化／復号化したペイロードが CRYP\_DOUT レジスタに保存されます。

次のシーケンスに従う必要があります。

- l) CRYP\_CR で GCM\_CCMPH ビットを“10”に設定します。
- m) CRYP\_CR で ALGODIR ビットを使用してアルゴリズムの方向 (暗号化または復号化) を選択します。
- n) CRYP\_DIN レジスタにペイロードメッセージをプログラムし、IFNF フラグを使用して入力 FIFO がデータを受信できるかどうかを判断します。または、8 ワードのブロック毎に CRYP\_DIN レジスタにデータをプログラムし、IFEM フラグを使用して入力 FIFO がデー

タを受信できるかどうかを判断することもできます (IFEM="1")。同時に、CRYP\_DOUT レジスタの OFNE/OFFU フラグを監視して出力 FIFO が空でないかどうかをチェックできます。

- o) すべてのペイロードブロックが暗号化または復号化されるまで前のステップを繰り返します。または、DMA を使用することもできます。

#### 4. CCM 最終フェーズ

このステップでは、認証タグを生成します。このフェーズで、メッセージの認証タグが生成され、CRYP\_DOUT レジスタに保存されます。

- p) CRYP\_CR で GCM\_CCMPH[1:0] ビットを"11"に設定します。
- q) 初期化された A0 カウンタを読み込み、CRYP\_DIN レジスタに 32 ビットを 4 回書き込んで 128 ビットの A0 値をプログラムします。
- r) CRYP\_SR レジスタで OFNE フラグ (FIFO 出力が空でない) が"1"にセットされるまで待ちます。
- s) CRYP\_DOUT レジスタを 4 回読み出します。出力は暗号化された認証タグに相当します。
- t) 暗号プロセッサを無効にします (CRYP\_CR の CRYPEN ビット="0")

**注：** ハードウェアでは、元の B0 および B1 パケットのフォーマットを行うこともなければ、暗号化と復号化の間でのタグを比較を行うこともありません。これはソフトウェアで処理する必要があります。ヘッダフェーズからタグフェーズに移行するときに、暗号プロセッサを無効/有効にする必要はありません。

#### AES 暗号メッセージ認証コード (CMAC)

CMAC アルゴリズムでは、平文を認証し、対応するタグを生成することができます。CMAC シーケンスは、ペイロードフェーズが省かれることを除き CCM のシーケンスと同一です。

## 20.3.3 データ型

データは、CRYP\_DIN レジスタに書き込まれると一度に 32 ビット (ワード) ずつ CRYP プロセッサに入ります。DES の原理は、データストリームが 64 ビットごとに処理され、各 64 ビットのブロックでは M1 をブロックの一番左のビット、M64 を一番右のビットとして M1 から M64 までビットに番号を振るといふものです。同じ原理が AES でも使用されていますが、ブロックサイズは 128 ビットです。

システムメモリ構成はリトルエンディアンです。使用されるデータタイプにかかわらず (ビット、バイト、16 ビットハーフワード、32 ビットワード)、最下位データが最下位アドレス位置を使用します。したがって、ビット、バイト、ハーフワードのスワッピング処理 (暗号化されるデータの種類による) は、IN FIFO から読み出されるデータが CRYP プロセッサに入る前に行う必要があります。同じスワッピング処理を、CRYP データが OUT FIFO に書き込まれる前に実施します。たとえば、ASCII テキストストリームの場合、この処理はバイトスワッピングになります。

処理されるデータの種類の、CRYP 制御レジスタ (CRYP\_CR) 内の DATATYPE ビットフィールドで設定します。

表 120 の色の凡例では、緑、青、赤、および白は、システムメモリに表される TDES ブロック値をハイライトします。

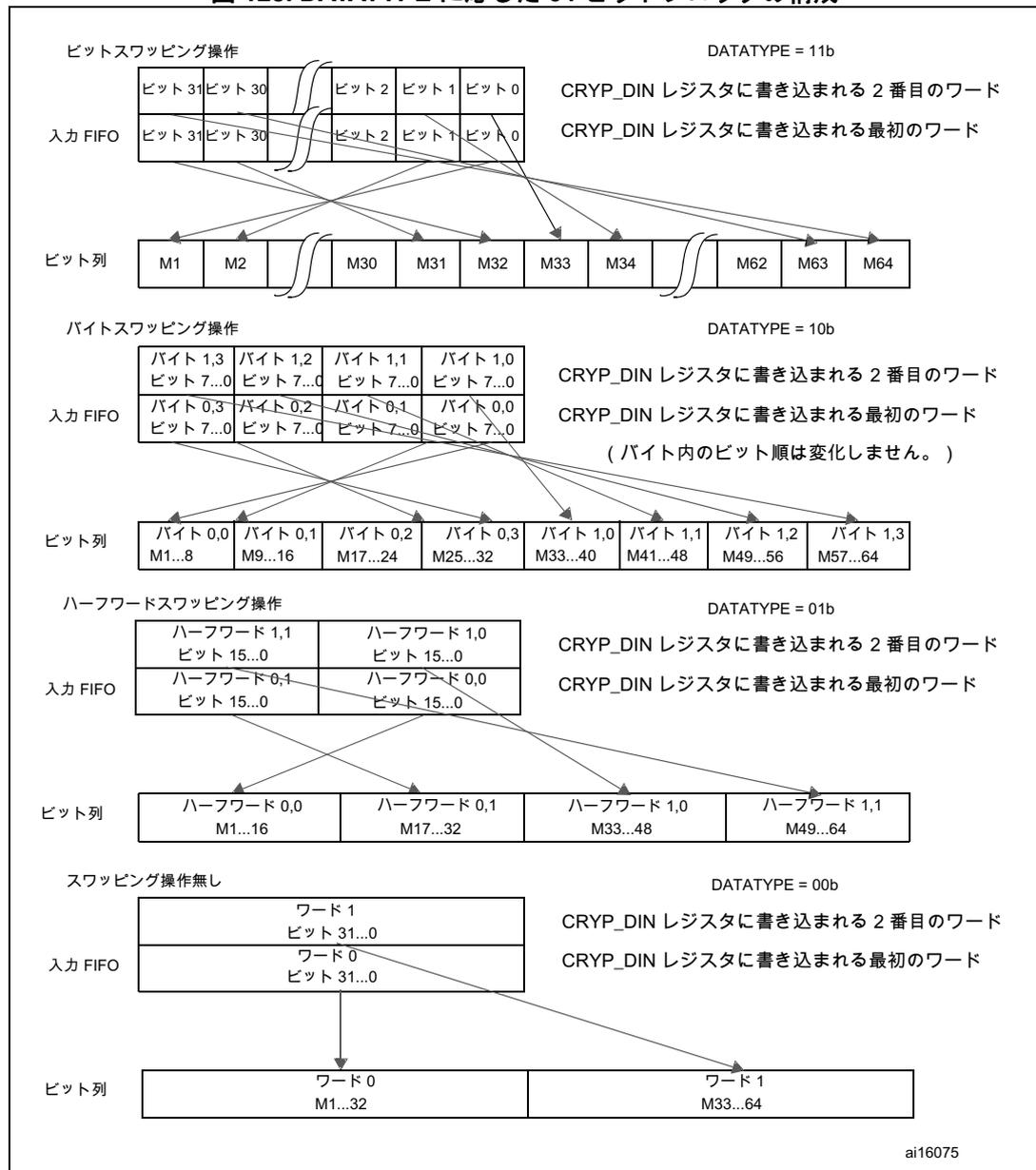
表 120. データ型

CRYP_CR 内の DATATYPE	実施される スワッピング処理	システムメモリデータ (平文または暗号)
00b	スワッピングなし	<p>例：トリプル DES ブロック値 <b>0xABCD77206973FE01</b> はシステムメモリでは次のように表現されます。</p> <p>TDES ブロックサイズ = 64 ビット = 2x 32 ビット</p>
01b	ハーフワード (16 ビット) スワッピング	<p>例：トリプル DES ブロック値 <b>0xABCD77206973FE01</b> はシステムメモリでは次のように表現されます。</p> <p>TDES ブロックサイズ = 64 ビット = 2x 32 ビット</p>
10b	バイト (8 ビット) スワッピング	<p>例：トリプル DES ブロック値 <b>0xABCD77206973FE01</b> はシステムメモリでは次のように表現されます。</p> <p>TDES ブロックサイズ = 64 ビット = 2x 32 ビット</p>
11b	ビットスワッピング	<p>トリプル DES ブロック値 <b>0x4E6F772069732074</b> はシステムメモリでは次のように表現されます。</p> <p>TDES ブロックサイズ = 64 ビット = 2x 32 ビット</p>

図 128 に、DATATYPE 値に従い、CRYP プロセッサによって 64 ビットのデータブロック M1...64 が IN FIFO から出力される 2 つの連続した 32 ビットワードでどのように構成されるかを示します。AES 暗号アルゴリズムの場合も、同じ図を拡張して 128 ビットブロックが簡単に形成できます (AES の場合、ブロック長は 32 ビットワード 4 つですが、スワッピングはワードレベルで行われるため、ここでトリプルDES について説明したものと全く同じになります)。

**注：** IN FIFO と CRYP データブロック間および CRYP データブロックと OUT FIFO 間でも同じスワッピングが行われます。

図 128. DATATYPE に応じた 64 ビットブロックの構成



## 20.3.4 初期化ベクタ CRYP\_IV0...1(L/R)

初期化ベクタは、2つの64ビットデータ項目とみなされます。したがって、初期化ベクタはシステムメモリ内で平文や暗号データと同じデータ形式や表現とはならず、DATATYPE 値にも影響されません。

初期化ベクタは2つの連続した32ビットワードである CRYP\_IVL (左部分、ビット IV1...32 と記述される) と CRYP\_IVR (右部分、ビット IV33...64 と記述される) によって定義されます。

DES またはトリプル DES CBC の暗号化では、DATATYPE 値に応じたスワッピング後に、CRYP\_IV0(L/R) ビットは、IN FIFO から出力される64ビットのデータブロック、つまりデータブロックの M1...64 ビットとの排他的論理和がとられます。DEA3 ブロックの出力が利用可能な場合は、この出力が CRYP\_IV0(L/R) ベクタにコピーされ、この新しい内容と次に IN FIFO から出力される64ビットのデータブロックとの排他的論理和がとられ、同様に続きます。

DES またはトリプル DES CBC 復号化では、DATATYPE 値に応じたスワッピング前に、CRYP\_IV0(L/R) ビットは、TDEA1 ブロックから引き渡される64ビットのデータブロック (つまり M1...64 ビット) との排他的論理和がとられ、OUT FIFO に押し出されます。排他的論理和の結果がスワッピングされ、OUT FIFO に押し出されると、CRYP\_IV0(L/R) 値が IN FIFO の出力で置換され、IN FIFO が空になって新しい64ビットのデータブロックが処理できるようになります。

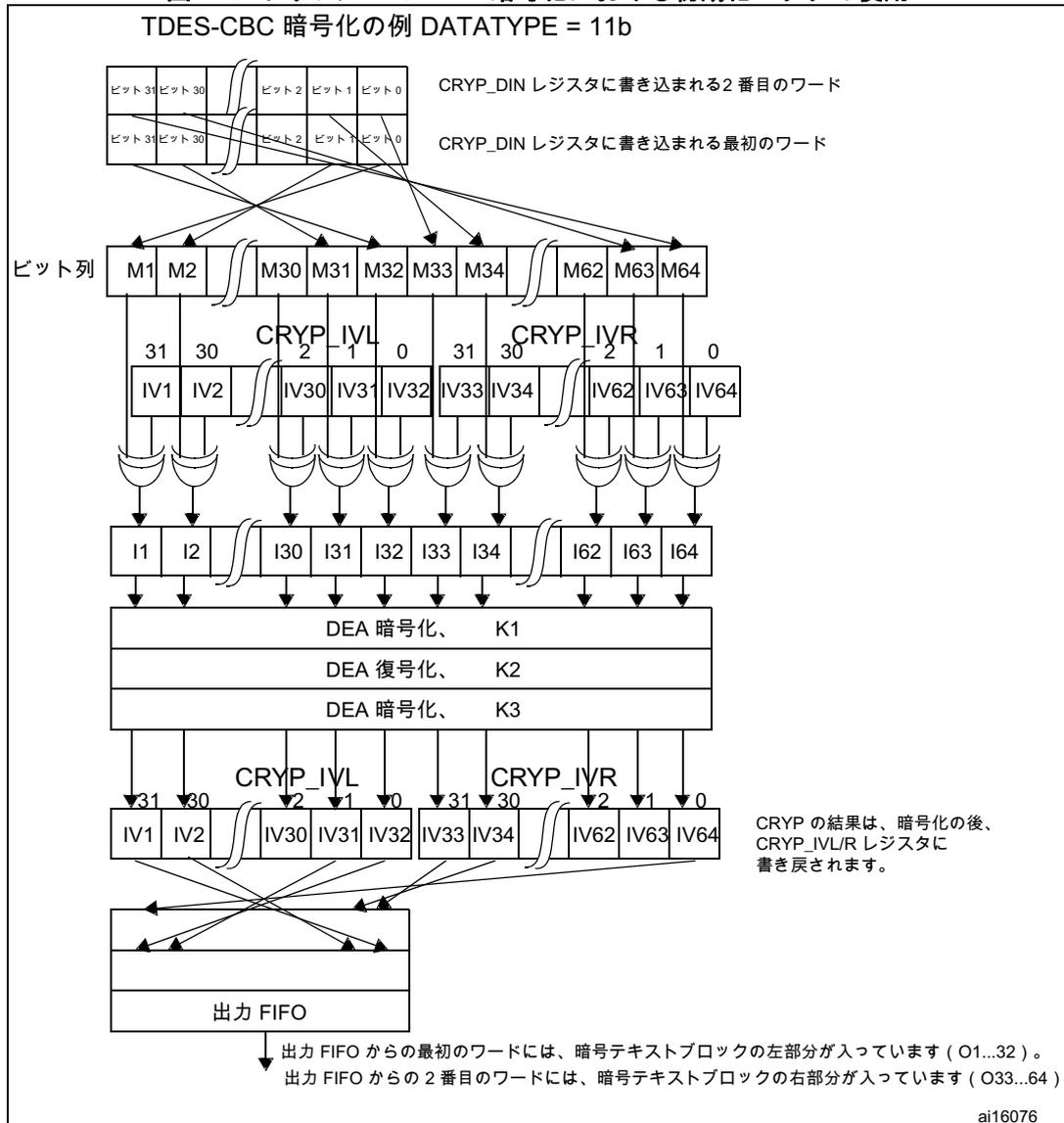
AES CBC 暗号化では、DATATYPE 値に応じたスワッピング後に、CRYP\_IV0...1(L/R) ビットと、IN FIFO から出力される128ビットのデータブロックとの排他的論理和がとられます。AES コアの出力が利用可能な場合は、この出力が CRYP\_IV0...1(L/R) ベクタにコピーされ、この新しい内容と次に IN FIFO から出力される128ビットのデータブロックとの排他的論理和がとられ、同様に続きます。

AES CBC 復号化では、DATATYPE 値に準じたスワッピング前に、CRYP\_IV0...1(L/R) ビットと、AES コアから引き渡される128ビットのデータブロックとの排他的論理和がとられ、OUT FIFO に押し出されます。排他的論理和の結果がスワッピングされ、OUT FIFO に押し出されると、CRYP\_IV0...1(L/R) 値が IN FIFO の出力で置換され、IN FIFO が空になって新しい128ビットのデータブロックが処理できるようになります。

AES CTR 暗号化または復号化では、CRYP\_IV0...1(L/R) ビットが AES コアによって暗号化されます。その後、暗号化の結果は、DATATYPE 値に応じたスワッピング後に、IN FIFO から出力される128ビットのデータブロックとの排他的論理和がとられます。排他的論理和の結果がスワッピングされ、OUT FIFO に押し出されると、CRYP\_IV0...1(L/R) 値のカウンタ部分 (32 LSB) がインクリメントされます。

CRYP\_SR レジスタの BUSY ビット = 1b になっているときは、CRYP\_IV0...1(L/R) レジスタに対するあらゆる書き込み操作が無視されます (CRYP\_IV0...1(L/R) レジスタの内容は変更できません)。そのため、初期化ベクタを変更する前に、BUSY ビット = 0b であることを確認する必要があります。

図 129. トリプルDES-CBC 暗号化における初期化ベクタの使用



## 20.3.5 CRYP ビジーステート

入力 FIFO に十分なデータがあり (DES またはトリプルDES アルゴリズムモードに対して 2 ワード以上、AES アルゴリズムモードに対して 4 ワード以上)、出力 FIFO に十分な空き領域があり (DES/トリプルDES に対して 2 ワード以上、AES に対して 4 ワード以上の領域)、CRYP\_CR レジスタで CRYPEN ビット = 1 である場合は、暗号プロセッサが自動的に暗号化または復号化処理を開始します (CRYP\_CR レジスタの ALGODIR ビットの値に応じて)。

この処理には、トリプル TDS アルゴリズムに対して AHB2 クロック 48 サイクル分、単純な DES アルゴリズムで AHB2 クロック 16 サイクル分、キー長 128、192、256 ビットの AES に対してそれぞれ AHB2 クロック 14、16、18 サイクル分の時間がかかります。この処理の間中、CRYP\_SR レジスタの BUSY ビットが“1”にセットされます。処理の最後に、CRYP コアによって 2 つ (DES/トリプルDES の場合) または 4 つ (AES の場合) のワードが出力 FIFO に書き込まれ、BUSY ビットがクリアされます。CBC、CTR モードでは、初期化ベクタである CRYP\_IVx(L/R)R (x=0..3) も更新されます。

暗号プロセッサがビジーの場合 (CRYP\_SR レジスタの BUSY ビット = 1b)、キーレジスタ (CRYP\_Kx(L/R)R, x = 0..3)、初期化レジスタ (CRYP\_IVx(L/R)R, x = 0..3) や CRYP\_CR レジスタのビット [9:2] への書き込み操作は無視され、レジスタは変更されません。そのため、暗号プロセッサがデータブロックを処理している間は、設定を変更することはできません。ただし、BUSY = 1 のときに CRYPEN ビットをクリアすることはできます。現在進行中の DES、トリプルDES、AES 処理が完了し、2 ワードまたは 4 ワードの結果が出力 FIFO に書き込まれた場合に限り、BUSY ビットがクリアされます。

**注：** *ブロックが DES またはトリプルDES モードで処理される場合、出力 FIFO が一杯になり、入力 FIFO に 1 つ以上の新しいブロックが含まれている場合は、新しいブロックが入力 FIFO から出力されますが、この新しいブロックを出力 FIFO に保存できる十分なスペースができるまで、BUSY ビットはハイのままになります。*

## 20.3.6 暗号化または復号化を実施する手順

### 初期化

1. ペリフェラルを初期化します (AES-ECB または AES-CBC 復号化のキーの準備を除き、操作の順序は重要ではありません。キーサイズとキー値はキーを準備する前に入力する必要があり、アルゴリズムはキーが準備できてから設定する必要があります)。
  - a) CRYP\_CR レジスタの KEYSIZE ビットでキーサイズ (AES の場合にのみ 128、192、256 ビット) を設定します。
  - b) 対称キーを CRYP\_KxL/R レジスタに書き込みます (アルゴリズムによって書き込まれるレジスタは 2 個から 8 個と異なります)。
  - c) CRYP\_CR レジスタの DATATYPE ビットでデータ型 (1、8、16、32 ビット) を設定します。
  - d) AES-ECB または AES-CBC での復号化の場合、キーを準備する必要があります。CRYP\_CR レジスタで ALGOMODE ビットを“111”にセットしてキー準備モードに設定します。その後 CRYPEN ビットに“1”を書き込みます。これで BUSY ビットがセットされます。BUSY が 0 に戻るまで待ちます (CRYPEN も自動的にクリアされます)。これで復号化用にキーが準備できました。
  - e) アルゴリズムと連結 (ECB/CBC では DES/トリプル DES、ECB/CBC/CTR/GCM/CCM では AES) を CRYP\_CR レジスタの ALGOMODE ビットで設定します。
  - f) CRYP\_CR レジスタの ALGODIR ビットで方向 (暗号化/復号化) を設定します。
  - g) CRYP\_IVxL/R レジスタに初期化ベクタを書き込みます (CBC または CTR モードのみ)。
2. CRYP\_CR レジスタで FFLUSH ビットに 1 を書き込み、IN および OUT FIFO を一掃します。

### DMA を使用してデータをメモリからまたはメモリに転送する場合の処理

1. 入力データをメモリから転送するよう DMA コントローラを設定します。メッセージ長が転送長となります。メッセージパッドはペリフェラルで管理されないため、メッセージ長は全ブロック数である必要があります。データはバーストモードで転送されます。バースト長は AES で 4 ワード、DES/トリプルDES で 2 または 4 ワードです。処理の完了を示すため、出力データの転送完了時に割り込みをセットするように DMA を設定します。
2. CRYPEN ビットに 1 を書き込んで、暗号プロセッサを有効にします。CRYP\_DMOCR レジスタの DIEN ビットと DOEN ビットをセットして、DMA リクエストを有効にします。
3. すべての転送と処理は、DMA および暗号プロセッサによって管理されます。DMA 割り込みが、処理の完了を示します。どちらの FIFO も通常は空で BUSY = 0 となっています。

## 割り込み時に CPU によってデータが転送される場合の処理

1. CRYP\_IMSCR レジスタの INIM ビットと OUTIM ビットをセットして、割り込みを有効にします。
2. CRYP\_CR レジスタの CRYPEN ビットをセットして、暗号プロセッサを有効にします。
3. 入力データを管理する割り込み：入力メッセージを IN FIFO にロードします。一度に 2 ワードまたは 4 ワードずつロードするか、または FIFO が一杯になるまでデータをロードすることができます。メッセージの最後のワードが FIFO に入ったら、INIM ビットをクリアして割り込みを無効にします。
4. 出力データを管理する割り込み：出力メッセージを OUT FIFO から読み出します。一度に 1 ブロック (2 ワードまたは 4 ワード) ずつ読み出すか、または FIFO が空になるまでデータを読み出すことができます。最後のワードが読み出されると、INIM=0、BUSY=0 になり、どちらの FIFO も空になります (IFEM=1 および OFNE=0)。OUTIM ビットをクリアすることで割り込みを無効にでき、CRYPEN ビットをクリアすることでペリフェラルを無効にできます。

## DMA も割り込みも使わない処理

1. CRYP\_CR レジスタの CRYPEN ビットをセットして、暗号プロセッサを有効にします。
2. 入力 FIFO に最初のブロックを書き込みます (2 から 8 ワード)。
3. メッセージ全体が処理されるまで次のシーケンスを繰り返します。
  - a) OFNE=1 となるのを待ってから、OUT-FIFO を読み出します (1 ブロックまたは FIFO が空になるまで)。
  - b) IFNF=1 となるのを待ってから、IN-FIFO に書き込みます (1 ブロックまたは FIFO が一杯になるまで)。
4. 処理の最後には、BUSY=0 になり、どちらの FIFO も空になります (IFEM=1 および OFNE=0)。ペリフェラルは、CRYPEN ビットをクリアすることで無効にできます。

## 20.3.7 コンテキストスワッピング

OS によって開始された新しいタスクがこのリソースを必要とするためにコンテキストスワッピングが必要な場合、コンテキスト全体を復元するために次のタスクを実施する必要があります (DMA を使用する場合の例)。

### AES および DES の場合

1. コンテキストの保存
  - a) CRYP\_DMACR レジスタの DIEN ビットをクリアして、IN FIFO での DMA 転送を停止します。
  - b) IN および OUT FIFO がどちらも空になり (CRYP\_SR レジスタで IFEM=1 および OFNE=0)、BUSY ビットがクリアされるまで待ちます。
  - c) CRYP\_DMACR レジスタの DOEN ビットに 0 を書き込んで OUT FIFO での DMA 転送を停止し、CRYPEN ビットをクリアします。
  - d) 現在の設定 (CRYP\_CR レジスタのビット [9:2] およびビット 19) を保存し、ECB モードでない場合には初期化ベクタも保存します。キー値はすでにメモリ内で利用できるようになっている必要があります。必要に応じて、DMA ステータス (IN および OUT メッセージのポインタ、残りバイト数など) を保存します。  
GCM/GMAC または CCM/CMAC アルゴリズムが使用される場合、追加ビットを保存します。
    - CRYP\_CR レジスタのビット [17:16]
    - コンテキストスワッピングレジスタ：  
GCM/GMAC または CCM/CMAC アルゴリズムの CRYP\_CSGCMCCM0..7  
GCM/GMAC アルゴリズムの CRYP\_CSGCM0..7

2. 他の処理を設定し、実行します。
3. コンテキストの復元
  - a) 保存した設定で、[セクション 20.3.6 : 暗号化または復号化を実施する手順 \(572 ページ\)](#)の初期化と同様にプロセッサを設定します。AES-ECB または AES-CBC 復号化の場合、キーをもう一度準備する必要があります。
  - b) 必要に応じて、DMA コントローラを再設定してメッセージの残りを転送します。
  - c) CRYPEN ビットをセットしてプロセッサを有効にし、DIEN ビットおよび DOEN ビットをセットして DMA リクエストを有効にします。

## トリプルDES の場合

トリプルDES では、AES の場合と同様の方法でコンテキストスワッピングを実施できます。しかし、入力 FIFO には最大 4 つの未処理ブロックを含むことがあり、またブロックあたりの処理時間が長くなるため、場合によっては IN FIFO が空になるのを待たずに処理を中断する方が速いこともあります。

1. コンテキストの保存
  - a) CRYP\_DMACR レジスタの DIEN ビットをクリアして、IN FIFO での DMA 転送を停止します。
  - b) CRYPEN ビットをクリアしてプロセッサを無効にします (現在のブロックの終了時に処理が停止します)。
  - c) OUT FIFO が空になり (CRYP\_SR レジスタで OFNE=0)、BUSY ビットがクリアされるまで待ちます。
  - d) CRYP\_DMACR レジスタの DOEN ビットに 0 を書き込み、OUT FIFO での DMA 転送を停止します。
  - e) 現在の設定 (CRYP\_CR レジスタのビット [9:2] およびビット 19) を保存し、ECB モードでない場合には初期化ベクタも保存します。キー値はすでにメモリ内で利用できるようになっている必要があります。必要に応じて、DMA ステータス (IN および OUT メッセージのポインタ、残りバイト数など) を保存します。未処理の IN FIFO にロードされたデータを読み戻し、それらを FIFO が空になるまでメモリに保存します。

**注 :** **GCM/GMAC または CCM/CMAC モードでは、CRYP\_CR レジスタのビット [17:16] も保存します。**

2. 他の処理を設定し、実行します。
3. コンテキストの復元
  - a) 保存した設定で、[セクション 20.3.6 : 暗号化または復号化を実施する手順 \(572 ページ\)](#)の初期化と同様にプロセッサを設定します。AES-ECB または AES-CBC 復号化の場合、キーをもう一度準備する必要があります。
  - b) コンテキストの保存中に保存されたデータを IN FIFO に書き込みます。
  - c) 必要に応じて、DMA コントローラを再設定してメッセージの残りを転送します。
  - d) CRYPEN ビットをセットしてプロセッサを有効にし、DIEN ビットおよび DOEN ビットをセットして DMA リクエストを有効にします。

## 20.4 CRYP 割り込み

CRYP が生成するマスク可能で独立した割り込みソースが 2 つあります。この 2 つのソースを組み合わせて 1 つの割り込み信号にします。これが NVIC (ネスト型ベクタ割り込みコントローラ) を駆動する CRYP からの唯一の割り込み信号です。この複合割り込みは、マスクされた各ソースの論理和関数であり、次にリストする各割り込みのいずれかがアサートされて有効になるとアサートされます。

CRYP\_IMSCR レジスタのマスクビットを変更することにより、これらの割り込みソースを個別に有効にしたり無効にしたりすることができます。適切なマスクビットを“1”にセットすることで、その割り込みが有効になります。

各割り込みソースのステータスは、raw 割り込みステータスに対しては CRYP\_RISR レジスタから、マスクされた割り込みステータスに対しては CRYP\_MISR レジスタから読み出すことができます。

### 出力 FIFO サービス割り込み - OUTMIS

出力 FIFO に 1 つ以上のデータ項目 (32 ビットワード) がある場合、出力 FIFO サービス割り込みがアサートされます。この割り込みは、有効な (32 ビット) ワードがなくなるまで出力 FIFO からデータを読み出すことによってクリアされます (つまり、この割り込みは OFNE (出力 FIFO が空でない) フラグの状態に従います)。

出力 FIFO サービス割り込み OUTMIS は、CRYP イネーブルビットでは有効になりません。結果として、出力 FIFO が空でない場合、CRYP を無効にしても OUTMIS 信号を強制的にローにすることはできません。

### 入力 FIFO サービス割り込み - INMIS

入力 FIFO が 4 ワード未満となった場合、入力 FIFO サービス割り込みがアサートされます。この割り込みは、入力 FIFO に 4 ワード以上を保持するようになるまで書き込み操作を行うことによってクリアされます。

入力 FIFO サービス割り込み INMIS は、CRYP イネーブルビットで有効になります。結果として、CRYP を無効にすると、入力 FIFO が空であっても INMIS 信号がローになります。

## 20.5 CRYP DMA インタフェース

暗号プロセッサには、DMA コントローラに接続するインタフェースが搭載されています。DMA 操作は CRYP DMA 制御レジスタである CRYP\_DMCCR を通して制御します。

バースト転送リクエスト信号とシングル転送リクエスト信号は、互いに排他的ではありません。それらは同時にアサートすることができます。たとえば、OUT FIFO で 6 ワードが利用可能な場合、バースト転送リクエストとシングル転送リクエストがアサートされます。4 ワードのバースト転送の後、残りの利用可能な 2 ワードを転送するにはシングル転送リクエストでなければアサートされません。これはストリーム内で受信する残りのワード数がバーストよりも少ないときに効果的です。

各リクエスト信号は、関連する DMA クリア信号がアサートされるまでアサートされたままとなります。リクエストクリア信号がネゲートされると、上述した条件に応じて、リクエスト信号を再びアクティブにすることができます。CRYP ペリフェラルが無効になるか、DMA イネーブルビットがクリアされると (CRYP\_DMCCR レジスタにおいて IN FIFO に対しては DIEN ビット、OUTFIFO に対しては DOEN ビット)、すべてのリクエスト信号がネゲートされます。

**注:** *DMA コントローラを 4 ワード以下のバーストを実行するように設定する必要があります。さもないと、一部のデータが失われる場合があります。*

*IN FIFO を一杯にする前に OUT FIFO を空にするよう DMA コントローラが動作するようにするには、OUTDMA チャンネルの優先度が INDMA チャンネルより高い必要があります。*



## 20.6 CRYP レジスタ

暗号コアは、複数の制御およびステータスレジスタ、8 個のキーレジスタ、4 個の初期化ベクタレジスタと連動しています。

### 20.6.1 CRYP 制御レジスタ (CRYP\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGOMODE [3]	Res.	GCM_CCMPH	
												r/w		r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRYPEN	FFLUSH	Res.	Res.	Res.	Res.	KEYSIZE		DATATYPE		ALGOMODE[2:0]			ALGODIR	Res.	Res.
r/w	w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		

ビット 31:20 予約済み、ハードウェアによって 0 に固定されています。

ビット 18 予約済み、ハードウェアによって 0 に固定されています。

ビット 17:16 GCM\_CCMPH[1:0] : 「GCM または CCM アルゴリズム」がセットされていない場合、影響はありません。

- 00 : GCM\_CCM 初期化フェーズ
- 01 : GCM\_CCM ヘッダフェーズ
- 10 : GCM\_CCM ペイロードフェーズ
- 11 : GCM\_CCM 最終フェーズ

ビット 15 **CRYPEN** : 暗号プロセッサイネーブル

- 0 : CRYP プロセッサは無効です。
- 1 : CRYP プロセッサは有効です。

**注 :** **CRYPEN** ビットは、**キーの準備プロセスが終了するか (ALGOMODE=111b)、GCM\_CCM 初期化フェーズになると自動的にハードウェアによってクリアされます。**

ビット 14 **FFLUSH** : FIFO フラッシュ

CRYPEN = 0 の場合、このビットに 1 を書き込むと IN および OUT FIFO が一掃されます (つまり、FIFO の読み出しおよび書き込みポインタがリセットされます)。このビットに 0 を書き込んで影響はありません。

CRYPEN = 1 の場合、このビットに 0 や 1 を書き込んで影響はありません。このビットを読み出すと常に 0 が返されます。

ビット 13:10 予約済み、ハードウェアによって 0 に固定されています。

ビット 9:8 **KEYSIZE**[1:0] : キーサイズ選択 (AES モードのみ)

このビットフィールドにより、AES 暗号コアに使用するキーのビット長を規定します。このビットフィールドは、DES またはトリプルDES モードでは意味を持ちません。

- 00 : 128 ビットのキー長
- 01 : 192 ビットのキー長
- 10 : 256 ビットのキー長
- 11 : 予約済み。この値は使用しないでください。

ビット 7:6 **DATATYPE[1:0]** : データ型選択

このビットフィールドによって、CRYP\_DIN レジスタに入力されるデータのフォーマットを規定します ([セクション 20.3.3 : データ型](#)を参照)。

00 : 32 ビットデータ。各ワードのスワッピング無し。IN FIFO に入れられる (または OUT FIFO から出される) 最初のワードがデータブロックのビット 1 から 32 を形成し、2 番目のワードがビット 33 から 64 を形成します。

01 : 16 ビットデータ、またはハーフワード。IN FIFO に入れられる (または OUT FIFO から出される) 各ワードは、相互スワッピングされた 2 つのハーフワードとみなされます。

10 : 8 ビットデータ、またはバイト。IN FIFO に入れられる (または OUT FIFO から出される) 各ワードは、相互スワッピングされた 4 バイトとみなされます。

11 : ビットデータ、またはビット列。IN FIFO に入れられる (または OUT FIFO から出される) 各ワードは、相互スワッピングされた 32 ビット (位置 0 にあるビット列の最初のビット) とみなされます。

ビット 19 および **ALGOMODE[3:0]** : アルゴリズムモード

5:3

0000 : トリプル DES-ECB (トリプル DES 電子コードブック) : データブロック間にはフィードバックはありません。初期化ベクタ (CRYP\_IV0(L/R)) は使用されず、3 つのキーベクタ (K1、K2、K3) が使用されます (K0 は使用されません)。

0001 : トリプル DES-CBC (トリプル DES 暗号ブロック連鎖) : 出力ブロックは、それ以降に続く入力ブロックと排他的論理和がとられてから、アルゴリズムへ入力されます。初期化ベクタ (CRYP\_IV0(L/R)) は初期化する必要があります、3 つのキーベクタ (K1、K2、K3) が使用されます (K0 は使用されません)。

0010 : DES-ECB (シンプル DES 電子コードブック) : データブロック間にはフィードバックはありません。初期化ベクタ (CRYP\_IV0(L/R)) は使用されず、1 つのキーベクタ (K1) のみが使用されます (K0、K2、K3 は使用されません)。

0011 : DES-CBC (シンプル DES 暗号ブロック連鎖) : 出力ブロックは、それ以降に続く入力ブロックと排他的論理和がとられてから、アルゴリズムへ入力されます。初期化ベクタ (CRYP\_IV0(L/R)) は初期化する必要があります。1 つのキーベクタ (K1) のみが使用されます (K0、K2、K3 は使用されません)。

0100 : AES-ECB (AES 電子コードブック) : データブロック間にはフィードバックはありません。初期化ベクタ (CRYP\_IV0(L/R)...1(L/R)) は使用されません。4 つのキーベクタ (K0...K3) はすべて使用されます。

0101 : AES-CBC (AES 暗号ブロック連鎖) : 出力ブロックは、それ以降に続く入力ブロックと排他的論理和がとられてから、アルゴリズムへ入力されます。初期化ベクタ (CRYP\_IV0(L/R)...1(L/R)) は初期化する必要があります。4 つのキーベクタ (K0...K3) はすべて使用されます。

0110 : AES-CTR (AES カウンタモード) : 出力ブロックは、それ以降に続く入力ブロックと排他的論理和がとられてから、アルゴリズムへ入力されます。初期化ベクタ (CRYP\_IV0(L/R)...1(L/R)) は初期化する必要があります。4 つのキーベクタ (K0...K3) はすべて使用されます。コアが常に現在のカウンタブロックを暗号化して、入力における平文や暗号と排他的論理和がとられるキーストリームを生成するので、CTR 復号化は CTR 暗号化と変わりません。したがって、ALGOMODE = 110b の場合は ALGODIR は意味を持たず、復号化用にキーを展開 (準備) してはなりません。

0111 : 復号化モード用の AES キーの準備です。CRYPEN = 1 の場合、この値を書き込むと、ただちにキー準備用の AES ラウンドが開始となります。秘密キーは、事前に K0...K3 レジスタにロードされている必要があります。CRYP\_SR レジスタの BUSY ビットは、キー準備中にセットされます。キー処理の後、その結果生成されるキーが再び K0...K3 レジスタにコピーされ、BUSY ビットがクリアされます。

1000 : ガロアカウンタモード (GCM) です。このアルゴリズムモードは、GMAC アルゴリズムに対しても使用されます。

1001 : CBC-MAC (CCM) のカウンタです。このアルゴリズムモードは、CMAC アルゴリズムに対しても使用されます。

ビット 2 **ALGODIR** : アルゴリズムの方向

- 0 : 暗号化
- 1 : 復号化

ビット 1:0 予約済みであり、0 のままにしておかなければなりません。

**注 :** **BUSY=1** である間に **KEYSIZE**、**DATATYPE**、**ALGOMODE**、**ALGODIR** ビットに書き込んでも何の影響もありません。これらのビットは、**BUSY=0** のときにのみ設定できます。**FFLUSH** ビットは、**BUSY=0** のときにのみセットする必要があります。さもないと、**FIFO** は一掃されますが処理中のブロックは一掃操作の直後に出力 **FIFO** に押し込まれ、**FIFO** が空でなくなることがあります。

## 20.6.2 CRYP ステータスレジスタ (CRYP\_SR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0003

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

ビット 31:5 予約済みであり、リセット値のままにしておかなければなりません。

ビット 4 **BUSY** : ビジービット

0 : CRYP コアは、いかなるデータも処理していません。理由は次のいずれかです。

- CRYP コアが無効になっており (CRYP\_CR レジスタで CRYPEN=0)、最後の処理が完了している、または
- CRYP コアが入力 FIFO に十分なデータが入るまで待っている、または出力 FIFO に十分な空きができるまで待っている (つまり、それぞれ DES の場合で 2 ワード以上、AES の場合で 4 ワード以上)。

1 : CRYP コアが現在、データブロックまたは (AES 復号化の) キー準備の処理をしています。

ビット 3 **OFFU** : 出力 FIFO フル

- 0 : 出力 FIFO はフルではありません。
- 1 : 出力 FIFO はフルです。

ビット 2 **OFNE** : 出力 FIFO ノットエンプティ

- 0 : 出力 FIFO は空です。
- 1 : 出力 FIFO は空ではありません。

ビット 1 **IFNF** : 入力 FIFO ノットフル

- 0 : 入力 FIFO はフルです。
- 1 : 入力 FIFO はフルではありません。

ビット 0 **IFEM** : 入力 FIFO エンプティ

- 0 : 入力 FIFO は空ではありません。
- 1 : 入力 FIFO は空です。

## 20.6.3 CRYP データ入力レジスタ (CRYP\_DIN)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

CRYP\_DIN レジスタは、データ入力レジスタです。このレジスタは 32 ビット幅です。このレジスタを使用して、暗号化時、最大 4 つの 64 ビット (トリプルDES) または 2 つの 128 ビット (AES) 平文を、復号化時は、暗号文ブロックを一度に 32 ビットワードずつ入力 FIFO に入力します。

最初に FIFO に書き込まれるワードは、入力ブロックの MSB です。入力ブロックの LSB は、最後に書き込まれます。データスワッピングを無視すると、次のようになります。

- DES/トリプル DES モード : ひとつのブロックは、ビット 1 (左端のビット) からビット 64 (右端のビット) まで番号が振られた一連のビットです。ビット 1 は FIFO に入力された最初のワードの MSB (ビット 31) に対応し、ビット 64 は FIFO に入力された 2 番目のワードの LSB (ビット 0) に対応します。
- AES モード : ひとつのブロックは、ビット 0 (左端のビット) からビット 127 (右端のビット) まで番号が振られた一連のビットです。ビット 0 は FIFO に書き込まれた最初のワードの MSB (ビット 31) に対応し、ビット 127 は FIFO に書き込まれた 4 番目のワードの LSB (ビット 0) に対応します。

さまざまなデータサイズに対応するため、CRYP\_CR レジスタの DATATYPE ビットを設定することにより、CRYP\_DIN レジスタに書き込まれたデータは、処理が行われる前にスワッピングすることができます。詳細については、[セクション 20.3.3 : データ型 \(567 ページ\)](#) を参照してください。

CRYP\_DIN レジスタに書き込まれると、データは入力 FIFO に押し込まれます。DES/トリプルDES モードにおいて、少なくとも 2 つ以上の 32 ビットワード (AES モードでは 4 つ以上の 32 ビットワード) が入力 FIFO に押し込まれており、かつ出力 FIFO に少なくとも 2 ワード以上の空きがある場合、CRYP エンジンは暗号化または復号化プロセスを開始します。このプロセスは、各プロセスラウンドにつき DES/トリプルDES モードでは入力 FIFO から 2 つの 32 ビットワード (AES モードでは 4 つの 32 ビットワード) を取り込み、出力 FIFO に 2 つの 32 ビットワード (AES モードでは 4 つの 32 ビットワード) を引き渡します。

CRYP\_DIN レジスタが読み出されると、

- CRYPEN = 0 の場合、FIFO が作動し、最も古いもの (最初の読み出し) から最新のもの (最後の読み出し) まで、入力 FIFO にあるデータが返されます。各読み出し操作の前に、IFEM フラグをチェックして FIFO が空でないことを確認する必要があります。
- CRYPEN = 1 の場合、未定義の値が返されます。

CRYP\_DIN レジスタが 1 回以上読み出された後は、新しいデータを処理する前に FFLUSH ビットをセットして FIFO を一掃する必要があります。

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

### ビット 31:0 DATAIN : データ入力

読み出し : CRYPEN = 0 の場合、入力 FIFO の内容を返し、それ以外の場合、未定義の値を返します。

書き込み : 入力 FIFO に書き込まれます。



## 20.6.4 CRYP データ出力レジスタ (CRYP\_DOUT)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

CRYP\_DOUT レジスタは、データ出力レジスタです。読み出し専用で 32 ビット幅です。このレジスタを使用して、暗号化時、最大 4 つの 64 ビット (トリプルDES モード) または 2 つの 128 ビット (AES モード) の暗号文を、復号化時は、平文ブロックを一度に 32 ビットワードずつ出力 FIFO から取り出します。

入力データの場合と同様、出力 FIFO から最初に読み出されるワードが出力ブロックの MSB です。出力ブロックの LSB は、最後に読み出されます。データスワッピングを無視すると、次のようになります。

- DES/トリプル DES モード : ビット 1 (左端のビット) は FIFO から読み出される最初のワードの MSB (ビット 31) に対応し、ビット 64 (右端のビット) は FIFO から読み出される 2 番目のワードの LSB (ビット 0) に対応します。
- AES モード : ビット 0 (左端のビット) は FIFO から読み出される最初のワードの MSB (ビット 31) に対応し、ビット 127 (右端のビット) は FIFO から読み出される 4 番目のワードの LSB (ビット 0) に対応します。

さまざまなデータサイズに対応するため、CRYP\_CR レジスタの DATATYPE ビットを設定することにより、データは、処理後にスワッピングすることができます。詳細については、[セクション 20.3.3: データ型 \(567 ページ\)](#) を参照してください。

CRYP\_DOUT レジスタを読み出すと、出力 FIFO に最後に入ったデータ (読み出しポインタが指すデータ) が返されます。

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

ビット 31:0 DATAOUT : データ出力

読み出し : 出力 FIFO の内容を返します。

書き込み : 何の影響もありません。

## 20.6.5 CRYP DMA 制御レジスタ (CRYP\_DMACR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **DOEN** : DMA 出カインーブル

- 0 : 送信データ転送用 DMA は無効です。
- 1 : 送信データ転送用 DMA は有効です。

ビット 0 **DIEN** : DMA 入カインーブル

- 0 : 受信データ転送用 DMA は無効です。
- 1 : 受信データ転送用 DMA は有効です。

## 20.6.6 CRYP 割り込みマスクセット/クリアレジスタ (CRYP\_IMSCR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

CRYP\_IMSCR レジスタは、割り込みマスクセットまたはクリアのためのレジスタです。読み出し/書き込みレジスタです。読み出し操作時、このレジスタは、関連する割り込みの現在のマスク値を示します。特定のビットに 1 を書き込むとマスクがセットされ、割り込みが読み出せるようになります。このビットに 0 を書き込むと、該当するマスクがクリアされます。ペリフェラルがリセットされると、すべてのビットが 0 にクリアされます。

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

ビット 31:2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **OUTIM** : 出力 FIFO サービス割り込みマスク

- 0 : 出力 FIFO サービス割り込みはマスクされています。
- 1 : 出力 FIFO サービス割り込みはマスクされていません。

ビット 0 **INIM** : 入力 FIFO サービス割り込みマスク

- 0 : 入力 FIFO サービス割り込みはマスクされています。
- 1 : 入力 FIFO サービス割り込みはマスクされていません。



## 20.6.7 CRYP raw 割り込みステータスレジスタ (CRYP\_RISR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0001

CRYP\_RISR レジスタは、raw 割り込みステータスレジスタです。読み出し専用レジスタです。読み出し時、このレジスタは、マスク前の該当する割り込みの現在の raw ステータスを示します。書き込み操作は何の影響もありません。

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

ビット 31:2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **OUTRIS** : 出力 FIFO サービス raw 割り込みステータス

出力 FIFO サービス割り込みをマスクする前の raw 割り込みの状態を示します。

0 : Raw 割り込みは保留中ではありません。

1 : Raw 割り込みは保留中です。

ビット 0 **INRIS** : 入力 FIFO サービス raw 割り込みステータス

入力 FIFO サービス割り込みをマスクする前の raw 割り込みの状態を示します。

0 : Raw 割り込みは保留中ではありません。

1 : Raw 割り込みは保留中です。

## 20.6.8 CRYP マスク済み割り込みステータスレジスタ (CRYP\_MISR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

CRYP\_MISR レジスタは、マスク済み割り込みステータスレジスタです。読み出し専用レジスタです。読み出し時、このレジスタは、マスク前の該当する割り込みの現在のマスクのステータスを示します。書き込み操作は何の影響もありません。

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

ビット 31:2 予約済みであり、リセット値のままにしておかなければなりません。

ビット 1 **OUTMIS** : 出力 FIFO サービスマスク済み割り込みステータス

出力 FIFO サービス割り込みマスク後の割り込みの状態を示します。

0 : 割り込みは保留中ではありません。

1 : 割り込みが保留中です。

ビット 0 INMIS : 入力 FIFO サービスマスク済み割り込みステータス  
 入力 FIFO サービス割り込みマスク後の割り込みの状態を示します。  
 0 : 割り込みは保留中ではありません。  
 1 : CRYPEN = 1 の場合、割り込みは保留中です。

## 20.6.9 CRYP キーレジスタ (CRYP\_K0...3(L/R)R)

アドレスオフセット : 0x20 から 0x3C

リセット値 : 0x0000 0000

これらのレジスタには、暗号キーが入っています。

トリプルDES モードでは、キーは 64 ビットのバイナリ値 (左から右に番号が振られています。つまり左端のビットがビット 1 です。) であり、K1、K2、K3 (K0 は使用しません) と名づけられています。また、各キーは 56 の情報ビットと 8 つのパリティビットで構成されています。パリティビットはエラー検出目的で予約されており、現在のブロックでは使用しません。このため、各 64 ビットのキー値  $Kx[1:64]$  のビット 8、16、24、32、40、48、56、64 は使用されません。

AES モードでは、キーは 1 つの 128、192、256 ビット長のビット列  $k_0k_1k_2\dots k_{127/191/255}$  ( $k_0$  は左端のビット) とみなされます。AES キーは次のようにレジスタに入力されます。

- AES-128 :  $k_0..k_{127}$  は  $b_{127}..b_0$  に対応 ( $b_{255}..b_{128}$  は使用されない)
- AES-192 :  $k_0..k_{191}$  は  $b_{191}..b_0$  に対応 ( $b_{255}..b_{192}$  は使用されない)
- AES-256 :  $k_0..k_{255}$  は  $b_{255}..b_0$  に対応

どの場合でも、 $b_0$  が右端のビットです。

### CRYP\_K0LR (アドレスオフセット : 0x20)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
b255	b254	b253	b252	b251	b250	b249	b248	b247	b246	b245	b244	b243	b242	b241	b240
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
b239	b238	b237	b236	b235	b234	b233	b232	b231	b230	b229	b228	b227	b226	b225	b224
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

### CRYP\_K0RR (アドレスオフセット : 0x24)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
b223	b222	b221	b220	b219	b218	b217	b216	b215	b214	b213	b212	b211	b210	b209	b208
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
b207	b206	b205	b204	b203	b202	b201	b200	b199	b198	b197	b196	b195	b194	b193	b192
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

### CRYP\_K1LR (アドレスオフセット : 0x28)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
k1.1 b191	k1.2 b190	k1.3 b189	k1.4 b188	k1.5 b187	k1.6 b186	k1.7 b185	k1.8 b184	k1.9 b183	k1.10 b182	k1.11 b181	k1.12 b180	k1.13 b179	k1.14 b178	k1.15 b177	k1.16 b176
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
k1.17 b175	k1.18 b174	k1.19 b173	k1.20 b172	k1.21 b171	k1.22 b170	k1.23 b169	k1.24 b168	k1.25 b167	k1.26 b166	k1.27 b165	k1.28 b164	k1.29 b163	k1.30 b162	k1.31 b161	k1.32 b160
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w



# 参考資料

## CRYP\_K1RR (アドレスオフセット : 0x2C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
k1.33 b159	k1.34 b158	k1.35 b157	k1.36 b156	k1.37 b155	k1.38 b154	k1.39 b153	k1.40 b152	k1.41 b151	k1.42 b150	k1.43 b149	k1.44 b148	k1.45 b147	k1.46 b146	k1.47 b145	k1.48 b144
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
k1.49 b143	k1.50 b142	k1.51 b141	k1.52 b140	k1.53 b139	k1.54 b138	k1.55 b137	k1.56 b136	k1.57 b135	k1.58 b134	k1.59 b133	k1.60 b132	k1.61 b131	k1.62 b130	k1.63 b129	k1.64 b128
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

## CRYP\_K2LR (アドレスオフセット : 0x30)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
k2.1 b127	k2.2 b126	k2.3 b125	k2.4 b124	k2.5 b123	k2.6 b122	k2.7 b121	k2.8 b120	k2.9 b119	k2.10 b118	k2.11 b117	k2.12 b116	k2.13 b115	k2.14 b114	k2.15 b113	k2.16 b112
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
k2.17 b111	k2.18 b110	k2.19 b109	k2.20 b108	k2.21 b107	k2.22 b106	k2.23 b105	k2.24 b104	k2.25 b103	k2.26 b102	k2.27 b101	k2.28 b100	k2.29 b99	k2.30 b98	k2.31 b97	k2.32 b96
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

## CRYP\_K2RR (アドレスオフセット : 0x34)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
k2.33 b95	k2.34 b94	k2.35 b93	k2.36 b92	k2.37 b91	k2.38 b90	k2.39 b89	k2.40 b88	k2.41 b87	k2.42 b86	k2.43 b85	k2.44 b84	k2.45 b83	k2.46 b82	k2.47 b81	k2.48 b80
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
k2.49 b79	k2.50 b78	k2.51 b77	k2.52 b76	k2.53 b75	k2.54 b74	k2.55 b73	k2.56 b72	k2.57 b71	k2.58 b70	k2.59 b69	k2.60 b68	k2.61 b67	k2.62 b66	k2.63 b65	k2.64 b64
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

## CRYP\_K3LR (アドレスオフセット : 0x38)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
k3.1 b63	k3.2 b62	k3.3 b61	k3.4 b60	k3.5 b59	k3.6 b58	k3.7 b57	k3.8 b56	k3.9 b55	k3.10 b54	k3.11 b53	k3.12 b52	k3.13 b51	k3.14 b50	k3.15 b49	k3.16 b48
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
k3.17 b47	k3.18 b46	k3.19 b45	k3.20 b44	k3.21 b43	k3.22 b42	k3.23 b41	k3.24 b40	k3.25 b39	k3.26 b38	k3.27 b37	k3.28 b36	k3.29 b35	k3.30 b34	k3.31 b33	k3.32 b32
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

## CRYP\_K3RR (アドレスオフセット : 0x3C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
k3.33 b31	k3.34 b30	k3.35 b29	k3.36 b28	k3.37 b27	k3.38 b26	k3.39 b25	k3.40 b24	k3.41 b23	k3.42 b22	k3.43 b21	k3.44 b20	k3.45 b19	k3.46 b18	k3.47 b17	k3.48 b16
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
k3.49 b15	k3.50 b14	k3.51 b13	k3.52 b12	k3.53 b11	k3.54 b10	k3.55 b9	k3.56 b8	k3.57 b7	k3.58 b6	k3.59 b5	k3.60 b4	k3.61 b3	k3.62 b2	k3.63 b1	k3.64 b0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

**注 :** 暗号プロセッサがビジー (CRYP\_SR レジスタでBUSY ビット=1) のとき、これらのレジスタへの書き込みアクセスは無視されます。



## 20.6.10 CRYP 初期化ベクタレジスタ (CRYP\_IV0...1(L/R)R)

アドレスオフセット : 0x40 から 0x4C

リセット値 : 0x0000 0000

CRYP\_IV0...1(L/R)R は、初期化ベクタの左ワードレジスタおよび右ワードレジスタ (DES/トリプル DES の場合は 64 ビット、AES の場合は 128 ビット) であり、CBC (暗号ブロック連鎖) および カウンタ (CTR) モードで使用します。トリプル DES または AES コアの各計算ラウンドの後、[セクション : DES およびトリプル DES 暗号ブロック連鎖 \(DES/トリプル DES-CBC\) モード \(555 ページ\)](#)、[セクション : AES 暗号ブロック連鎖 \(AES-CBC\) モード \(559 ページ\)](#)、および [セクション : AES カウンタ \(AES-CTR\) モード \(561 ページ\)](#) で説明されているように、CRYP\_IV0...1(L/R)R レジスタが更新されます。

IV0 は、初期化ベクタの左端のビットであり、IV63 (DES、トリプル DES) または IV127 (AES) は右端のビットです。IV1(L/R)R は、AES でのみ使用します。

### CRYP\_IV0LR (アドレスオフセット : 0x40)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0	IV1	IV2	IV3	IV4	IV5	IV6	IV7	IV8	IV9	IV10	IV11	IV12	IV13	IV14	IV15
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV16	IV17	IV18	IV19	IV20	IV21	IV22	IV23	IV24	IV25	IV26	IV27	IV28	IV29	IV30	IV31
rw															

### CRYP\_IV0RR (アドレスオフセット : 0x44)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV32	IV33	IV34	IV35	IV36	IV37	IV38	IV39	IV40	IV41	IV42	IV43	IV44	IV45	IV46	IV47
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV48	IV49	IV50	IV51	IV52	IV53	IV54	IV55	IV56	IV57	IV58	IV59	IV60	IV61	IV62	IV63
rw															

### CRYP\_IV1LR (アドレスオフセット : 0x48)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV64	IV65	IV66	IV67	IV68	IV69	IV70	IV71	IV72	IV73	IV74	IV75	IV76	IV77	IV78	IV79
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV80	IV81	IV82	IV83	IV84	IV85	IV86	IV87	IV88	IV89	IV90	IV91	IV92	IV93	IV94	IV95
rw															

### CRYP\_IV1RR (アドレスオフセット : 0x4C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV96	IV97	IV98	IV99	IV100	IV101	IV102	IV103	IV104	IV105	IV106	IV107	IV108	IV109	IV110	IV111
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV112	IV113	IV114	IV115	IV116	IV117	IV118	IV119	IV120	IV121	IV122	IV123	IV124	IV125	IV126	IV127
rw															



**注：** DES/3DES モードでは、CRYP\_IV0(L/R) のみを使用します。  
暗号プロセッサがビジー (CRYP\_SR レジスタでBUSY ビット=1) のとき、これらのレジスタへの書き込みアクセスは無視されます。

## 20.6.11 CRYP コンテキストスワップレジスタ (CRYP\_CSGCMCCM0..7R および CRYP\_CSGCM0..7R)

アドレスオフセット：

- CRYP\_CSGCMCCM0..7 : 0x050 から 0x06C : GCM/GMAC または CCM/CMAC アルゴリズムにのみ使用
- CRYP\_CSGCM0..7 : 0x070 から 0x08C : GCM/GMAC アルゴリズムにのみ使用

リセット値 : 0x0000 0000

GCM/GMAC または CCM/CMAC アルゴリズムが選択された場合、CRYP プロセッサの内部レジスタステータスはすべてこれらのレジスタ内にあります。暗号プロセッサがすでに別のタスクによって使用されているとき、優先度の高いタスクが暗号プロセッサを必要とするためにコンテキストスワッピングを実行する必要がある場合に役立ちます。

そのような場合、CRYP\_CSGCMCCM0..7R および CRYP\_CSGCM0..7R (GCM/GMAC モード時) または CRYP\_CSGCMCCM0..7R (CCM/CMAC モード時) レジスタを読み出し、取得した値をシステムメモリ空間に保存する必要があります。その後、優先権のあるタスクが暗号プロセッサを使用できるようになり、暗号計算が完了すると保存されたコンテキストをメモリから読み出し、該当するコンテキストスワップレジスタに書き戻すことができます。

**注：** これらのレジスタは、GCM/GMAC または CCM/CMAC アルゴリズムモードが選択されているときにのみ使用します。

### CRYP\_CSGCMCCMxR : ここで x=[7:0]

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

### CRYP\_CSGCMxR : ここで x=[7:0]

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

## 20.6.12 CRYP レジスタマップ

表 121. CRYP レジスタマップとリセット値

オフ セット	レジスタ名 リセット値	レジスタサイズ																																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00 0x00	CRYP_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値													ALGOMODE[3]	0	0	0	0																
0x04	CRYP_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x08	CRYP_DIN	DATAIN																																
	リセット値																																	
0x0C	CRYP_DOUT	DATAOUT																																
	リセット値																																	
0x10	CRYP_DMCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x14	CRYP_IMSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x18	CRYP_RISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x1C	CRYP_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x20	CRYP_K0LR	CRYP_K0LR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x24	CRYP_K0RR	CRYP_K0RR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
...																																		
0x38	CRYP_K3LR	CRYP_K3LR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x3C	CRYP_K3RR	CRYP_K3RR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x40	CRYP_IV0LR	CRYP_IV0LR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x44	CRYP_IV0RR	CRYP_IV0RR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x48	CRYP_IV1LR	CRYP_IV1LR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x4C	CRYP_IV1RR	CRYP_IV1RR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50	CRYP_CSGCMCCMR	CRYP_CSGCMCCMR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	



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

表 121. CRYP レジスタマップとリセット値 (続き)

オフ セット	レジスタ名 リセット値	レジスタサイズ																																
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x54	CRYP_CSGCMCCM1R	CRYP_CSGCMCCM1R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	CRYP_CSGCMCCM2R	CRYP_CSGCMCCM2R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	CRYP_CSGCMCCM3R	CRYP_CSGCMCCM3R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	CRYP_CSGCMCCM4R	CRYP_CSGCMCCM4R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64	CRYP_CSGCMCCM5R	CRYP_CSGCMCCM5R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	CRYP_CSGCMCCM6R	CRYP_CSGCMCCM6R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6C	CRYP_CSGCMCCM7R	CRYP_CSGCMCCM7R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	CRYP_CSGCM0R	CRYP_CSGCM0R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	CRYP_CSGCM1R	CRYP_CSGCM1R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x78	CRYP_CSGCM2R	CRYP_CSGCM2R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	CRYP_CSGCM3R	CRYP_CSGCM3R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	CRYP_CSGCM4R	CRYP_CSGCM4R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	CRYP_CSGCM5R	CRYP_CSGCM5R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	CRYP_CSGCM6R	CRYP_CSGCM6R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	CRYP_CSGCM7R	CRYP_CSGCM7R																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 21 ハッシュプロセッサ (HASH)

このセクションは、特に指定がない限り、STM32F756xx デバイス全体に適用されます。

### 21.1 HASH の概要

ハッシュプロセッサは、各種のアプリケーションに適したセキュアハッシュアルゴリズム (SHA-1、SHA-224、SHA-256)、MD5 (メッセージダイジェストアルゴリズム 5) ハッシュアルゴリズムおよび HMAC (鍵付きハッシュメッセージ認証コード) アルゴリズムに完全に準拠した処理系となっています。最長 ( $2^{64} - 1$ ) ビットのメッセージに対して、メッセージダイジェスト (SHA-1 アルゴリズムは 160 ビット、SHA-256 アルゴリズムは 256 ビット、SHA-224 アルゴリズムは 224 ビット、MD5 アルゴリズムは 128 ビット) を生成し、さらに HMAC アルゴリズムによって、ハッシュ関数を用いたメッセージの認証を可能としています。HMAC アルゴリズムは、SHA-1、SHA-224、SHA-256、MD5 のいずれかのハッシュ関数を 2 回呼び出すことを基本としています。

### 21.2 HASH の主な機能

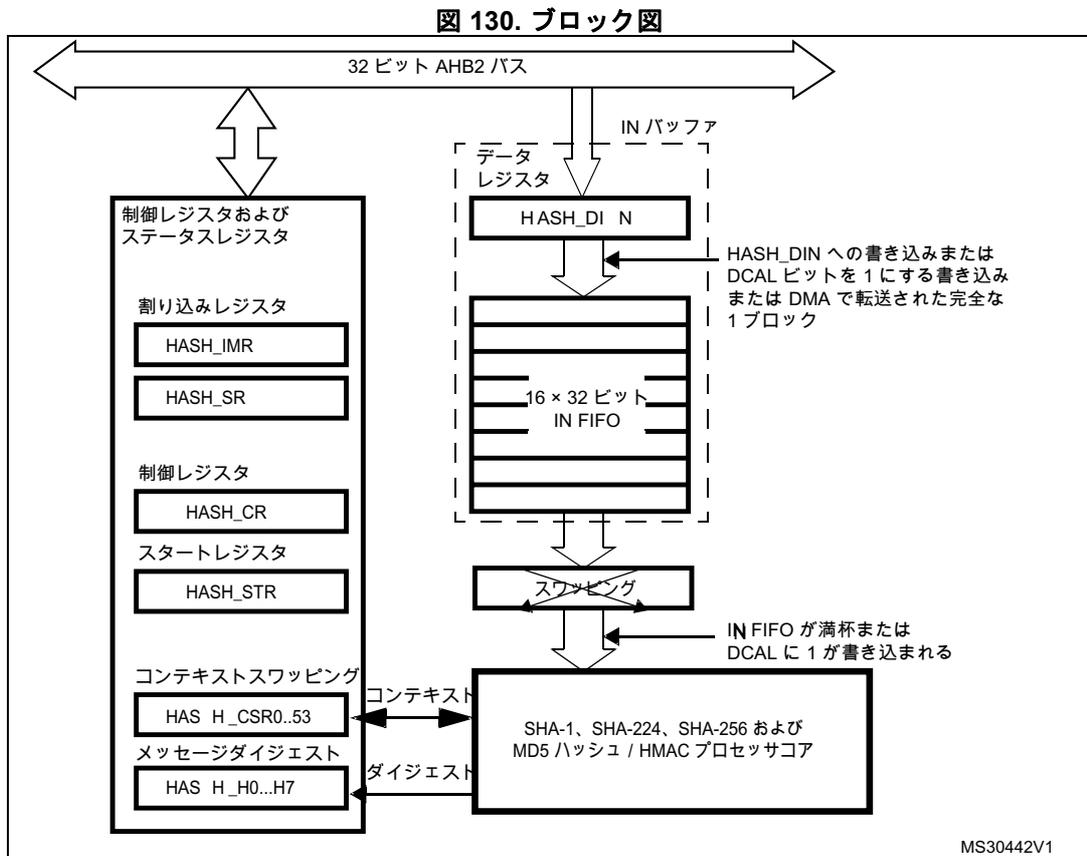
- 以下に準拠したデータ認証アプリケーションに好適
  - FIPS PUB 180-2 (連邦情報処理規格公報 180-2)
  - セキュアハッシュ標準規格 (SHA-1、SHA-224、および SHA-256)
  - IETF RFC 1321 (インターネット技術タスクフォース RFC1321) 仕様 (MD5)
- SHA-1、SHA-224、SHA-256、および MD5 の高速計算
- AHB スレーブペリフェラル
- 入力データ用 32 ビットデータワード。ワード、ハーフワード、バイト、ビットのビット列表現をサポート。リトルエンディアンデータ表現のみ対応
- リトルエンディアンの入力ビット列表現に対応し、ビッグエンディアンの SHA1、SHA-224、SHA-256 計算標準に適合させるための自動スワッピング
- モジュロ 512 ( $16 \times 32$  ビット) のメッセージダイジェスト計算に適合するよう入力ビット列を補完する自動パディング
- 出力メッセージダイジェスト用に  $8 \times 32$  ビットワード (H0 から H7) を再ロードでき、中断されたメッセージダイジェスト計算の続行が可能
- 連続するメッセージブロックから、対応する 32 ビットワードのダイジェストが相互に追加され、メッセージ全体のダイジェストを形成
- ダイレクトメモリアクセス (DMA) をサポートする自動データフロー制御

**注：** **SHA-1、SHA-224、SHA-256 アルゴリズムで定義されているとおり、 $bx1$  で 1 ビット付加し、続く  $bx0$  で  $N$  ビット付加することで、512 で割ったときの余りが 448 になるようにメッセージの長さをパディング (拡張) します。このあと、元のメッセージ長のバイナリ表記である 64 ビット整数によって、メッセージが完成します。**

**このハッシュプロセッサでは、メッセージ入力の最低必要量は 32 ビットワードですので、入力した 32 ビットワード内の有効ビット数を表す情報を、メッセージ入力の最後に追加する必要があります。**

## 21.3 HASH の機能詳細

図 130 に、ハッシュプロセッサのブロック図を示します。



FIPS PUB 180-2 標準および IETF RFC 1321 公報では、メッセージまたはデータファイルの圧縮表記を生成するための SHA-1、SHA-224、SHA-256、および MD5 セキュアハッシュアルゴリズムが、それぞれ規定されています。入力メッセージの長さが  $2^{64}$  ビット未満の場合、SHA-1、SHA-224、SHA-256、および MD5 は、メッセージダイジェストと呼ばれる 160 ビット、224 ビット、256 ビット、および 128 ビットの出力文字列をそれぞれ生成します。その後、メッセージの署名を作成したり検証したりするため、メッセージダイジェストはデジタル署名アルゴリズムを使用して処理することができますようになります。メッセージ自体ではなくメッセージダイジェストに署名を行うことで、多くの場合処理効率が向上します。これは通常、メッセージダイジェストの方がメッセージよりもはるかにサイズが小さいからです。デジタル署名の検証者は、デジタル署名の作成者が使用したのと同じハッシュアルゴリズムを使用する必要があります。

SHA-1、SHA-224、SHA-256 および MD5 は「安全」とされています。それはあるメッセージダイジェストに対応するメッセージの発見、もしくは同じメッセージダイジェストを生成する 2 つの異なるメッセージの発見が計算上不可能だからです。メッセージの転送中に何らかの変更が加えられた場合、極めて高い確率で異なるメッセージダイジェストになり、その署名は検証を通過できません。SHA-1 または SHA-224 および SHA-256 アルゴリズムの詳細は FIPS PUB 180-2 (連邦情報処理規格公報 180-2) 2002 年 8 月 1 日版を参照してください。

この規格の現時点での実装はリトルエンディアン入力データ規則に従って動作します。例えば、C 言語文字列「abc」はメモリ内で 24 ビットの 16 進数値 0x434241 として表す必要があります。

ハッシュプロセッサで処理するメッセージやデータファイルはビット列とみなされます。メッセージ長は、メッセージのビット数（空メッセージは長さ 0）です。この 32 ビットのビット列が 32 ビットワードを形成するとみなすことができます。FIPS PUB 180-1 標準はビット列が左から右へ増える規則を採用しており、ビットの集まりをバイト（8 ビット）またはワード（32 ビット）にグループ化できることに注意してください（ただし、ハーフワード（16 ビット）を使用する場合もあれば、暗黙的にビッグエンディアンのバイト順序（ハーフワード）を使用する実装もいくつかあります）。この規則は、主にパディングに重要です（[12 ページのセクション 1.3.4「メッセージのパディング」](#)を参照）。

## 21.3.1 処理時間

メッセージの中間ブロックの計算には次の時間が必要です。

- SHA-1 では 66 HCLK クロックサイクル
- SHA-224 では 50 HCLK クロックサイクル
- SHA-256 では 50 HCLK クロックサイクル
- MD5 では 50 HCLK クロックサイクル

これに中間ブロック（16 ワード）をプロセッサにロードするために必要な時間（512 ビットブロックに対して少なくとも 16 クロックサイクル）を追加する必要があります。

メッセージ（または HMAC のキー）の最終ブロックを処理するのに必要な時間はこれより長い場合があります。この時間は最終ブロックの長さでキー（HMAC モードの）の長さによって異なります。中間ブロックの処理と比べて以下の倍数増える場合があります。

- ハッシュメッセージは 1 から 2.5 倍
- HMAC 入力キーは約 2.5 倍
- HMAC メッセージは 1 から 2.5 倍
- ショートキーの場合の HMAC 出力キーは約 2.5 倍
- ロングキーの場合の HMAC 出力キーは約 3.5 から 5 倍

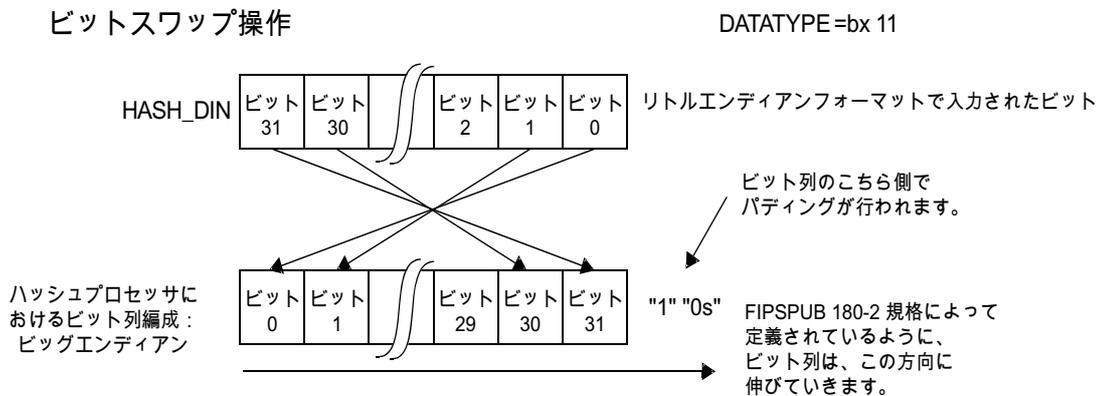
## 21.3.2 データ型

データは HASH\_DIN レジスタに書き込むことにより 32 ビット（ワード）ごとにハッシュプロセッサに入力されます。しかし、元のビット列はバイト、ハーフワード、またはワード単位で構成することができ、ビットで表記することもできます。システムのメモリ構成はリトルエンディアンであるのに対して SHA1、SHA-224 と SHA-256 計算はビッグエンディアンなので、ハッシュプロセッサは、元のビット列のグループ化方法により、ビット、バイト、またはハーフワードスワップ操作を自動的にを行います。

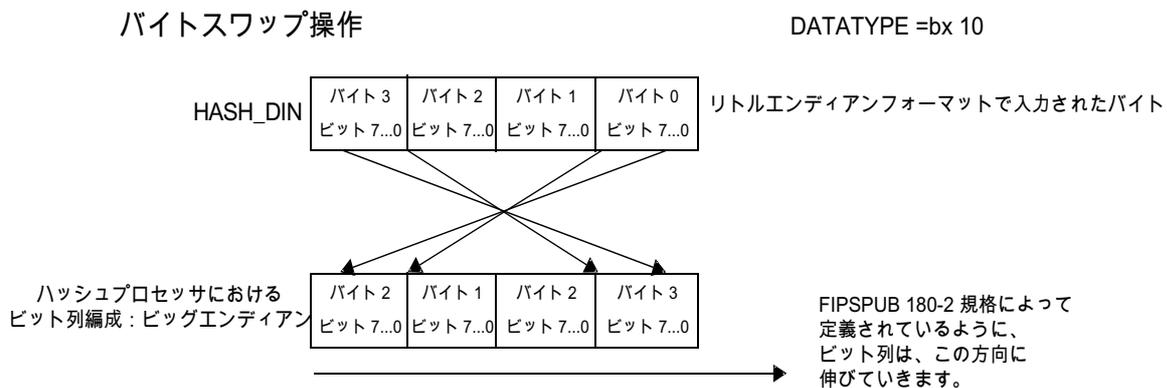
処理するデータの種類の種類は、HASH 制御レジスタ（HASH\_CR）の DATATYPE ビットフィールドで設定します。

図 131. ビット、バイト、およびハーフワードスワッピング

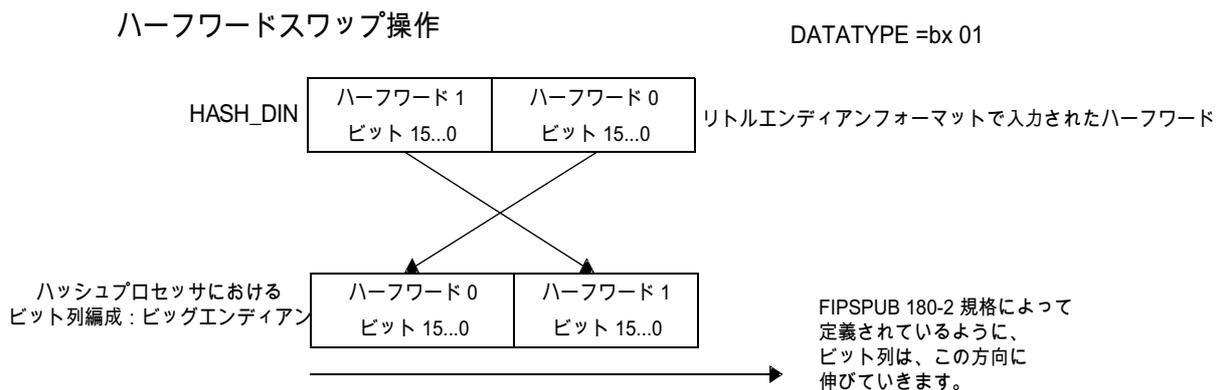
A バイナリデータハッシュの場合、すべてのビットが次のようにスワップされます。



B バイトデータハッシュの場合、すべてのバイトが次のようにスワップされます。



C ハーフワードハッシュの場合、すべてのハーフワードが次のようにスワップされます。



ai16082

メッセージの最下位ビットはハッシュプロセッサに入力される最初のワードの位置 0 (右) でなければならず、ビット列の 32 番目のビットはハッシュプロセッサに入力される 2 番目のワードの位置 0 でなければなりません。それ以降は同様です。

## 21.3.3 メッセージダイジェストの計算

HASH は、メッセージダイジェストを計算する際に、512 ビットのブロックごとに順次処理します。したがって、DMA または CPU により  $16 \times 32$  ビットワード (= 512 ビット) がハッシュプロセッサに書き込まれるごとに HASH は自動的にメッセージダイジェストの計算を開始します。この操作は、部分ダイジェスト計算として知られています。

処理するメッセージは、HASH\_DIN レジスタに書き込まれる 32 ビットワード単位でペリフェラルに入力されます。HASH\_DIN レジスタの現在の内容は、このレジスタに新しいデータが書き込まれるごとに、入力 FIFO (IN FIFO) に転送されます。HASH\_DIN と入力 FIFO で、長さ 17 ワードの FIFO を形成します (IN バッファと呼びます)。

ブロックの処理はブロックの最終値が IN\_FIFO に入力された後で初めて開始できます。ペリフェラルは、HASH\_DIN レジスタにメッセージの最終ビットが含まれるか否かについての情報を得る必要があります。次の 2 つの場合があります。

- DMA を使用しない場合：
  - 部分ダイジェスト計算は、HASH\_DIN レジスタに追加のワードを書き込むことによって行います (実際には次のブロックの最初のワードです)。その後ソフトウェアは、プロセッサが再びレディになるのを待ってから (DINIS=1 になるまで)、新しいデータを HASH\_DIN に書き込む必要があります。
  - 最終ダイジェスト計算 (入力された最終ブロック) は、DCAL ビットに 1 を書き込むことによって行います。

- DMA を使用する場合：
 

HASH\_DIN レジスタの内容は DMA コントローラから送られる情報により自動的に解釈されます。

- 単一 DMA 転送の場合：多重 DMA 転送 (MDMAT) ビットをクリアする必要があります。最終ブロックが DMA チャンネル経由で HASH\_DIN レジスタに転送されると、最終ダイジェスト計算を開始するため DCAL ビットが HASH\_STR レジスタ内で自動的に 1 にセットされます。
- 多重 DMA 転送の場合：ハードウェアによって DCAL ビットが自動的にセットされないよう、多重 DMA 転送 (MDMAT) ビットをソフトウェアによって 1 にセットする必要があります。この場合、DMA 転送要求の最後でハッシュおよび HMAC の各フェーズに対する最終ダイジェスト計算は開始せず、プロセッサは新たな DMA 転送を受け取ることができます (HMAC フェーズの詳細は HMAC 操作のセクションを参照)。最終ブロックの最後で DCAL ビットを自動的にセットして最終ダイジェスト計算を開始するため、最終 DMA 転送の間に、ソフトウェアによって多重 DMA 転送 (MDMAT) ビットをクリアする必要があります。
- HASH\_DIN レジスタの内容は DMA コントローラから送られる情報により自動的に解釈されます。

この処理、つまりデータ入力と部分ダイジェスト計算は、オリジナルメッセージの最後のビットが HASH\_DIN レジスタに書き込まれるまで続きます。メッセージの長さ (ビット数) は任意の整数値をとることができるため、HASH プロセッサに書き込まれる最後のワードの有効ビット数は 1 から 32 まで可能です。最終メッセージダイジェスト計算の前にメッセージのパディングを正しく行うため、最終ワードの有効ビット数 NBLW を HASH\_STR レジスタに書き込む必要があります。

一旦これを行うと、HASH\_STR に DCAL=1 を書き込むことによって、ハッシュプロセッサは最後に入力されたメッセージブロックの処理を開始します。この処理は次の項目で構成されています。

- メッセージのパディング操作の自動実行: この操作の目的は、パディングを行った後のメッセージ長が 512 の倍数になるようにすることです。HASH は、メッセージダイジェストを計算する際に、512 ビットのブロックごとに順次処理します。
- 最終メッセージダイジェストの計算

DMA が有効な場合、最終データワードが転送される時、ハッシュプロセッサに情報が提供されず。すると、DCAL に 1 が書き込まれたかのように、パディングとダイジェスト計算が自動的に行われます。

### 21.3.4 メッセージのパディング

メッセージのパディングは、もとのメッセージの最後に「1」を付加し、その後 m 個の「0」を付加して最後に 64 ビットの整数を付加することで成り立っており、長さ 512 のパディングされたメッセージブロックを作ります。HASH\_DIN レジスタに対して最後に書き込まれたワードの中の NBLW ビットフィールドで定義された位置に、「1」が追加され、残りの上位ビットは「0」にクリアされます。

例: オリジナルメッセージが ASCII バイナリコード形式の「abc」で、長さ L=24 であると仮定します。

```
byte 0    byte 1    byte 2    byte 3
01100001 01100010 01100011 UUUUUUUU
<-- HASH_DINに書かれた最初のワード-->
```

NBLW に値 24 をロードする必要があります。ビット列のビット位置 24 (上記ビット列で左から右へカウントします) に「1」が付加されます。これは HASH\_DIN レジスタのビット 31 に対応します (リトルエンディアン規約):

```
01100001 01100010 01100011 1UUUUUUU
```

L = 24 であるため、上記のビット列のビット数は 25 ビットであり、423 個の「0」が付加されて 448 ビットになります。その結果、ビッグエンディアン形式の 16 進数で以下ようになります。

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

2 ワード表記の L 値 (00000000 00000018) が付加されます。したがってパディング処理により最終的に得られるメッセージは、16 進数で次のようになります。

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000028
```

HASH がリトルエンディアンバイト入力形式を使用するようプログラムされている場合、次の手順に従って上記メッセージを入力する必要があります。

1. HASH\_DIN レジスタに 0xUU636261 を書き込みます (ここで、「U」は影響なしを意味します)。
2. HASH\_STR レジスタに 0x18 を書き込みます (オリジナルメッセージの長さは 24 ビットなので HASH\_DIN レジスタに最後に書き込まれるワードの有効ビット数は 24 です)。
3. HASH\_STR レジスタに 0x10 を書き込んでメッセージのパディング処理とダイジェスト計算を開始します。NBLW ≠ 0x00 の場合、メッセージのパディング処理によって HASH\_DIN レジスタの NBLW 値で指定される位置に「1」が書き込まれ、ビット位置 [31:(NBLW+1)] に「0」が挿入されます。NBLW == 0x00 の場合、メッセージのパディング処理によって、値 0x0000

0001 の新しいワードが 1 つ挿入されます。その後、すべてゼロのワード (0x0000 0000) および 2 ワード表記のメッセージ長が付加され、16 x 32 ビットワードのブロックが得られます。

4. HASH 計算が行われ、HASH\_Hx レジスタ (x = 0..4) に SHA-1 アルゴリズムに対するメッセージダイジェストが得られます。例：

```
H0 = 0xA9993E36
H1 = 0x4706816A
H2 = 0xBA3E2571
H3 = 0x7850C26C
H4 = 0x9CD0D89D
```

## 21.3.5 ハッシュ操作

HASH\_CR の MODE ビットが“0”のとき HASH\_CR レジスタの INIT ビットに“1”を書き込むとハッシュ機能 (SHA-1、SHA-224、SHA-256 および MD5) が選択されます。同時に (INIT ビットがセットされている場合) ALGO ビットを使用してアルゴリズム (SHA-1、SHA-224、SHA-256 または MD5) が選択されます。

その後 HASH\_DIN レジスタにワード単位で書き込むことによりメッセージを送ることができます。512 ビット (つまり 16 ワード) のブロックが書き込まれていれば、次のブロックの最初のデータを書き込むことによって、部分ダイジェスト計算が開始されます。ハッシュプロセッサは、SHA-1 アルゴリズムでは 66 サイクル、MD5 アルゴリズム、SHA-224 アルゴリズム、および SHA-256 アルゴリズムでは 50 サイクルの間、ビジーになります。

この後、この処理はメッセージの最後のワードまで繰り返すことができます。DMA 転送を使用する場合は、[DMA によってデータをロードする手順](#)セクションを参照してください。さもなければ、メッセージ長が 512 ビットの整数倍ではない場合、最終ダイジェストの計算を開始するため、HASH\_STR レジスタに書き込みを行う必要があります。

いったん計算が行われれば、ダイジェストは HASH\_H0..HASH\_H7 レジスタから (MD5 アルゴリズムでは HASH\_H4 は無関係) から読み出すことができます。

ここで、MD5 アルゴリズムを選択した場合、HASH\_H4..HASH\_H7 は無関係です。

HASH\_H5..SHA-1 アルゴリズムを選択した場合、HASH\_H7 は無関係です。

SHA-224 アルゴリズムを選択した場合、HASH\_H7 は無関係です。

## 21.3.6 HMAC 操作

HMAC アルゴリズムは、処理するメッセージをユーザが選択したキーに対して不可逆的に紐づけすることによるメッセージ認証に使用します。HMAC の仕様については、「HMAC: keyed-hashing for message authentication」、1997 年 2 月、H. Krawczyk、M. Bellare、R. Canetti 著を参照してください。

基本的に、このアルゴリズムは 2 つのネストされたハッシュ操作で構成されています。

$$\text{HMAC}(\text{message}) = \text{Hash}(((\text{key} \mid \text{pad}) \text{ XOR } 0x5C) \mid \text{Hash}(((\text{key} \mid \text{pad}) \text{ XOR } 0x36) \mid \text{message}))$$

ここで、

- pad はゼロの列で、基礎となるハッシュ関数データブロックの長さまでキーを拡張するために必要です (SHA-1、SHA224、SHA-256、および MD5 ハッシュアルゴリズムのいずれに対しても 512 ビットです)。
- | は連結演算子を表します。

HMAC を計算するには 4 つの異なるフェーズが必要です。

1. MODE ビットが「1」の状態では ALGO ビットを望みのアルゴリズムに対応する値にセットして INIT ビットに「1」を書き込むことによりブロックを初期化します。使用するキーが 64 バイトを超える場合、このフェーズでは LKEY ビットもセットする必要があります (この場合、HMAC 仕様は、実際のキーの代わりにそのキーのハッシュを使用する必要があることを想定しています)。
2. そして、内部ハッシュ関数に使用するキーをコアに与えます。この操作では、ハッシュ操作においてメッセージを送るのと同じ方法を取ります (つまり、HASH\_DIN に書き込み、最後に HASH\_STR に書き込みます)。
3. 最後のワードが入力され計算が開始されると、ハッシュプロセッサがキーを算出します。その後、ハッシュプロセッサはハッシュ操作でメッセージを送るために使用したのと同じ方法を使用してメッセージテキストを受け取ることができるようになります。
4. 最初のハッシュ処理終了後、ハッシュプロセッサは「レディ」を返して外部ハッシュ関数に使用するキーを受け取る用意ができたことを示します (通常、このキーは内部ハッシュ関数に使用するものと同じです)。キーの最後のワードが入力され計算が開始されると HASH\_H0...HASH\_H7 レジスタ内で HMAC の計算結果が使用可能になります。

**注:** 1 **HMAC 初期計算遅延はキーとメッセージの長さによって変わります。HMAC を同じキー長 (ロングまたはショート) を持つ 2 つのネストされた基礎ハッシュ関数として使用できます。**

### 21.3.7 コンテキストスワッピング

ハッシュ/HMAC 処理を中断してより優先度の高い別の処理を行い、その処理が完了した後に中断した元の処理を完了することができます。これを行うには、中断した作業のコンテキストをハッシュレジスタからメモリに保存し、その後ハッシュレジスタからメモリに復元する必要があります。

データフローをソフトウェアまたは DMA によって制御する手順を以下に示します。

#### ソフトウェアによってデータをロードする手順

コンテキストは、現在処理しているブロックがないときのみ保存することができます。つまり、DINIS=1 (最終ブロックの処理が完了し FIFO が空の状態) または NBW ≠ 0 (FIFO は満杯ではなく進行中の処理がない状態) を待つ必要があります。

- コンテキストの保存:  
次のレジスタの内容をメモリに保存します。
  - HASH\_IMR
  - HASH\_STR
  - HASH\_CR
  - HASH\_CSR0 から HASH\_CSR53
- コンテキストの復元:  
優先度の高いタスクが完了したときには、コンテキストを復元できます。以下のシーケンスの順序に従ってください。
  - a) メモリに保存した値を HASH\_IMR、HASH\_STR、および HASH\_CR レジスタに書き込みます。
  - b) HASH\_CR レジスタの INIT ビットをセットすることによって、ハッシュプロセッサを初期化します。
  - c) メモリに保存した値を、HASH\_CSR0 から HASH\_CSR53 レジスタに書き込みます。これで、処理が中断したところから再開することができます。

## DMA によってデータをロードする手順

この場合、DMA 転送が進行中なのか処理が継続中なのかを予測することができません。したがって、メッセージの処理を中断するためには、DMA 転送を停止し、HASH がレディになるまで待つ必要があります。

- 処理の中断：
  - DMAE ビットをクリアして DMA インタフェースを無効にします。
  - 現在の DMA 転送が完了するまで待ちます (HASH\_SR レジスタ内で DMAES=0 になるのを待ちます)。ブロックが完全に HASH レジスタに転送されている場合と、そうでない場合があることに注意してください。
  - DMA コントローラの対応するチャンネルを無効にします。
  - ハッシュプロセッサがレディになるまで待ちます (処理中のブロックがない状態)。つまり、DINIS=1 になるのを待ちます。
- コンテキストの保存と復元フェーズは上記と同じです (ソフトウェアによってデータをロードする手順を参照)。

メッセージの端を転送するために DMA コントローラを再設定します。これで、DMAE ビットをセットすることにより処理が中断したところから再開することができます。

**注：** コンテキストスワッピングが HMAC 操作を伴わない場合、HASH\_CSR38 から HASH\_CSR53 のレジスタを保存し復元する必要はありません。

コンテキストスワッピングが2つのブロック間で発生する場合 (最終ブロックが完全に処理されつつも、次のブロックはまだIN FIFO にプッシュされていない状態。HASH\_CR レジスタのNBW=000)、HASH\_CSR22 から HASH\_CSR37 のレジスタを保存し復元する必要はありません。

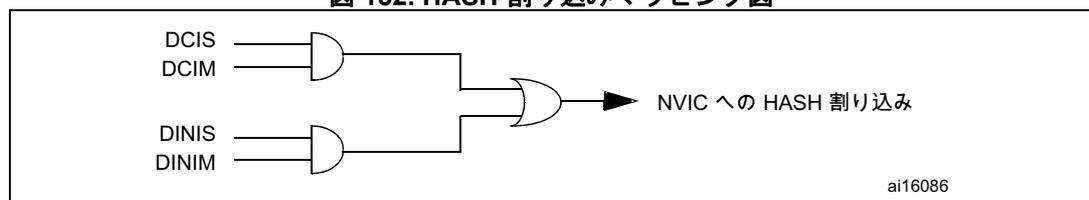
### 21.3.8 HASH 割り込み

HASH プロセッサによって生成され個別にマスク可能な2つの割り込みソースがあります。これらは、同じ割り込みベクタに接続されています。

HASH\_IMR レジスタのマスクビットを変更することにより、これらの割り込みソースを個別に有効にしたり無効にしたりすることができます。適切なマスクビットを1にセットすることで、その割り込みが有効になります。

個別の割り込みソースのステータスは HASH\_SR レジスタから読み出すことができます。

図 132. HASH 割り込みマッピング図



## 21.4 HASH レジスタ

HASH コアは、いくつかの制御レジスタ、ステータスレジスタ、および 5 つのメッセージダイジェストレジスタと連携しています。

これらのレジスタはすべてワード単位でのみアクセス可能です。さもなければ AHB エラーが発生します。

### 21.4.1 HASH 制御レジスタ (HASH\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[1]	Res.	LKEY
													rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MDMAT	DINNE	NBW				ALGO[0]	MODE	DATATYPE		DMAE	INIT	Res.	Res.
		rw	r	r	r	r	r	rw	rw	rw	rw	rw	w		

ビット 31:19 予約済み、ハードウェアによって 0 に固定されています。

ビット 17 予約済み、ハードウェアによって 0 に固定されています。

ビット 16 **LKEY** : ロングキー選択

このビットによって、HMAC モードにおけるショートキー (≤ 64 バイト) またはロングキー (> 64 バイト) を選択します。

0 : ショートキー (≤ 64 バイト)

1 : ロングキー (> 64 バイト)

**注 :** この選択は、INIT ビットがセットされ MODE=1 の場合にのみ有効です。計算中にこのビットを変更しても影響はありません。

ビット 15:14 予約済み、ハードウェアによって 0 に固定されています。

ビット 13 **MDMAT** : 多重 DMA 転送

多重 DMA 転送が必要な大容量のファイルをハッシングするときに、このビットをセットします。

0 : DCAL は DMA 転送の最後に自動的にセットされます。

1 : DCAL は DMA 転送の最後に自動的にセットされません。

ビット 12 **DINNE** : DIN ノットエンブティ

このビットは HASH\_DIN レジスタに有効なデータがある場合 (少なくとも 1 度は書き込まれたあと) にセットされます。このビットは INIT ビット (初期化) または DCAL ビット (前回のメッセージ処理の完了) のどちらかに 1 が書き込まれるとクリアされます。

0 : データ入力バッファにデータがありません

1 : 入力バッファには少なくとも 1 ワードのデータがあります

**ビット 11:8 NBW** : 既にプッシュされたワード数

このビットフィールドは、既に IN FIFO にプッシュされているメッセージ内のワード数を示します。

NBW は、DINNE=1 のときに HASH\_DIN レジスタに書き込みアクセスが行われると値が増加 (+1) します。

NBWは、INIT ビットに 1 が書き込まれたとき、またはダイジェスト計算が開始されたとき (DCAL に 1 が書き込まれたか DMA 転送終了時) に 0000 になります。

## ● DMA を使用しない場合 :

0000 かつ DINNE=0 の場合 : DIN バッファにプッシュされているワードはありません (バッファは空で、HASH\_DIN レジスタと IN\_FIFO の両方も空)

0000 かつ DINNE=1 の場合 : DIN バッファに 1 ワードがプッシュされています (HASH\_DIN レジスタには 1 ワード有り、IN\_FIFO は空)

0001: DIN バッファに 2 ワードがプッシュされています (HASH\_DIN レジスタと IN\_FIFO にそれぞれ 1 ワード有り)

...

1111 : DIN バッファに 16 ワードがプッシュされています

## ● DMA を使用している場合、NBW には IN FIFO にプッシュされている正確なワード数が格納されています。

**ビット 18 および ALGO[1:0]** : アルゴリズム選択

**ビット 7** これらのビットによって SHA-1、SHA-224、SHA256 または MD5 のアルゴリズムを選択します。

00 : SHA-1 アルゴリズムを選択

01 : MD5 アルゴリズムを選択

10 : SHA224 アルゴリズムを選択

11 : SHA256 アルゴリズムを選択

**注 :** この選択は、INIT ビットがセットされている場合にのみ有効です。計算中にこのビットを変更しても影響はありません。

**ビット 6 MODE** : モード選択

このビットによって、選択したアルゴリズムに対して HASH または HMAC モードを選択します。

0 : ハッシュモードを選択します

1 : HMAC モードを選択します。使用するキーが 64 バイトを超える場合、LKEY をセットする必要があります。

**注 :** この選択は、INIT ビットがセットされている場合にのみ有効です。計算中にこのビットを変更しても影響はありません。

**ビット 5:4 DATATYPE** : データ型選択

HASH\_DIN レジスタに入力したデータのフォーマットを定義します。

00 : 32 ビットデータ。HASH\_DIN に書き込まれたデータは順序を変えずに直接 HASH 処理に使用されます。

01 : 16 ビットデータ、またはハーフワード。HASH\_DIN に書き込まれたデータは 2 つのハーフワードとみなされ、HASH 処理に使用する前にスワップされます。

10 : 8 ビットデータ、またはバイト。HASH\_DIN に書き込まれたデータは 4 バイトとみなされ、HASH 処理に使用する前にスワップされます。

11 : ビットデータ、またはビット列。HASH\_DIN に書き込まれたデータは 32 ビット (ビット位置 0 におけるデータ列の最初のビット) とみなされ、HASH 処理 (ビット位置 31 におけるデータ列の最初のビット) に使用する前にスワップされます。

ビット 3 DMAE : DMA イネーブル

0 : DMA 転送は無効です

1 : DMA 転送は有効です。DMA 要求は、HASH コアがデータを受け取る準備ができるとすぐに送られます。

**注:** 1 : このビットは、メッセージの最後のデータを転送中に、DMA が DMA ターミナルカウント信号をアサートすると、ハードウェアによってクリアされます。このビットは、INIT ビットに 1 が書き込まれた場合、クリアされません。

2 : すでに DMA に対して DMA 転送が要求されてからこのビットに 0 が書き込まれた場合は、DMAE はクリアされますが現在の転送は中断されません。その代わりに、転送が完了するか INIT に 1 が書き込まれるまで、DMA インタフェースは内部的に有効になっています。

ビット 2 INIT : メッセージダイジェスト計算の初期化

このビットに 1 を書き込むとハッシュプロセッサコアがリセットされるので、HASH は新しいメッセージのメッセージダイジェストを計算可能となります。

このビットに 0 を書き込んで影響はありません。

このビットを読み出すと常に 0 が返ります。

ビット 1:0 予約済み、クリア状態を保つ必要があります。

## 21.4.2 HASH データ入力レジスタ (HASH\_DIN)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

HASH\_DIN はデータ入力レジスタです。このレジスタは 32 ビット幅です。このレジスタを使用して、512 ビットのブロック毎にメッセージを入力します。HASH\_DIN レジスタに書き込みを行うと、AHB データバス上の値が HASH コアに「プッシュ」され、このレジスタには AHB データバス上の新しい値が入ります。正しいメッセージ表記を得るためには HASH\_CR レジスタ内の DATATYPE ビットを事前に設定しておく必要があります。

HASH\_DIN レジスタに 16 ワードのブロックが書き込まれると、以下のように中間ダイジェストの計算が開始されます。

- DMA を使用していない場合、HASH\_DIN レジスタに新しいデータを書き込む（次のブロックの最初のワード）ことにより開始されます（中間ダイジェストの計算）。
- DMA を使用している場合は自動的に開始されます。

HASH\_DIN レジスタに最後のブロックが書き込まれると、以下のように最終ダイジェストの計算（パディングを含む）が開始されます。

- HASH\_STR レジスタの DCAL ビットに 1 を書き込むことにより開始されます（最終ダイジェストの計算）。
- DMA を使用し、かつ MDMAT ビットが“0”に設定されている場合は自動的に開始されます。

ダイジェストの計算（中間または最終）が進行中の場合、HASH\_DIN レジスタへの新たなアクセスはすべて HASH 計算が完了するまで（AHB バスにウェイトステートを挿入することにより）延長されます。

HASH\_DIN レジスタが読み出されると、このアドレスに書き込まれた最後のワードがアクセスされず（リセット後はゼロ）。

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

### ビット 31:0 DATAIN : データ入力

読み出し = レジスタの現在の内容を返します。

書き込み = レジスタの現在の内容が IN FIFO にプッシュされ、レジスタには AHB データバス上の新しい値が入ります。

## 21.4.3 HASH スタートレジスタ (HASH\_STR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

HASH\_STR レジスタには 2 つの機能があります。

- ハッシュプロセッサに入力されたメッセージの最後のワードにおける有効ビット数を定義します (HASH\_DIN レジスタに書き込まれた最後のデータの有効最下位ビット数)。
- DCAL ビットに 1 を書き込むことによりメッセージの最後のブロックの処理を開始します。

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

ビット 31:9 予約済み、ハードウェアによって 0 に固定されています。

ビット 8 **DCAL** : ダイジェスト計算

このビットに 1 を書き込むと以前書き込まれた NBLW の値を使用してメッセージのパディングが開始され、INIT ビットに最後に 1 が書き込まれてから IN FIFO に書き込まれたすべてのデータワードを使用して最終メッセージダイジェストの計算が開始されます。このビットを読み出すと 0 が返されます。

ビット 7:5 予約済み、ハードウェアによって 0 に固定されています。

ビット 4:0 **NBLW** : ハッシュプロセッサのビット列編成におけるメッセージの最終ワードの有効ビット数

これらのビットに書き込みが行われていて DCAL が“0”の場合、それらは AHB データバス上の値になります。

0x00 : ハッシュプロセッサのビット列編成に書き込まれた最後のデータの全 32 ビット (データスワッピング後) は有効です。

0x01 : ハッシュプロセッサのビット列編成に書き込まれた最後のデータのビット [31] (データスワッピング後) のみ有効です。

0x02 : ハッシュプロセッサのビット列編成に書き込まれた最後のデータのビット [31:30] (データスワッピング後) のみ有効です。

0x03 : ハッシュプロセッサのビット列編成に書き込まれた最後のデータのビット [31:29] (データスワッピング後) のみ有効です。

...

0x1F : ハッシュプロセッサのビット列編成に書き込まれた最後のデータのビット [0] (データスワッピング後) のみ有効です。

これらのビットに書き込みが行われ、DCAL が“1”の場合、ビットフィールドは変更されません。

それらを読み出すと最後に NBLW に書き込まれた値が返されます。

**注 :** これらのビットは DCAL ビットをセットする前に設定する必要があります。さもないと有効になりません。特に、NBLW と DCAL を同時に設定することはできません。

## 21.4.4 HASH ダイジェストレジスタ (HASH\_HR0..4/5/6/7)

アドレスオフセット : 0x0C から 0x32C

リセット値 : 0x0000 0000

これらのレジスタには、次のように指定されたメッセージダイジェストの計算結果が入っています。

1. SHA1 アルゴリズム仕様上、それぞれ H0、H1、H2、H3、および H4  
この場合、HASH\_H5 から HASH\_H7 のレジスタは使われず、ゼロが読み出されることに注意してください。
2. MD5 アルゴリズム仕様上、それぞれ A、B、C、および D  
この場合、HASH\_H4 から HASH\_H7 のレジスタは使われず、ゼロが読み出されることに注意してください。
3. SHA224 アルゴリズム仕様上、それぞれ H0 から H6  
この場合、HASH\_H7 のレジスタは使われず、ゼロが読み出されることに注意してください。
4. SHA256 アルゴリズム仕様上、それぞれ H0 から H7

HASH コアが中間ダイジェストまたは最終メッセージダイジェストを計算している最中にこれらのいずれかのレジスタへの読み出しアクセスが発生すると (DCAL ビットに 1 が書き込まれたとき)、この読み出しは HASH 計算が完了するまで一時停止します。

**注 :** H0、H1、H2、H3 および H4 のマッピングは 2 つの領域に複製されます。

### HASH\_HR0

アドレスオフセット : 0x310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

### HASH\_HR1

アドレスオフセット : 0x314

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r



## HASH\_HR2

アドレスオフセット : 0x318

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H2															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H2															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

## HASH\_HR3

アドレスオフセット : 0x31C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H3															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H3															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

## HASH\_HR4

アドレスオフセット : 0x320

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H4															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H4															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

## HASH\_HR5

アドレスオフセット : 0x324

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H5															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H5															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

## HASH\_HR6

アドレスオフセット : 0x328

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H6															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H6															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

## HASH\_HR7

アドレスオフセット : 0x32C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H7															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H7															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

注 : **INIT ビットに1を書き込むことによって、新しいビットストリームのダイジェスト計算を開始するとき、これらのレジスタはリセット値をとります。**

## 21.4.5 HASH 割り込み有効レジスタ (HASH\_IMR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

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

ビット 31:2 予約済み、ハードウェアによって 0 に固定されています。

ビット 1 **DCIE** : ダイジェスト計算完了割り込みイネーブル

- 0 : ダイジェスト計算完了割り込みは無効です
- 1 : ダイジェスト計算完了割り込みは有効です

ビット 0 **DINIE** : データ入力割り込みイネーブル

- 0 : データ入力割り込みは無効です
- 1 : データ入力割り込みは有効です

## 21.4.6 HASH ステータスレジスタ (HASH\_SR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	BUSY	DMAS	DCIS	DINIS											
											r	r	rc_w0	rc_w0	

ビット 31:4 予約済み、ハードウェアによって 0 に固定されています。

### ビット 3 **BUSY** : ビジービット

- 0 : 現在処理中のブロックはありません。
- 1 : ハッシュコアが処理中のデータブロックがあります。

### ビット 2 **DMAS** : DMA ステータス

- このビットから、DMA インタフェース動作に関する情報を得ることができます。このビットは DMAE と共にセットされ、DMAE=0、かつ DMA 転送中ではない場合にクリアされます。このビットに関連する割り込みはありません。
- 0 : DMA インタフェースは無効 (DMAE=0)、かつ転送中ではありません。
  - 1 : DMA インタフェースは有効 (DMAE=1)、または転送中です。

### ビット 1 **DCIS** : ダイジェスト計算完了割り込みステータス

- このビットは、ダイジェストがレディーになったとき (メッセージ全体が処理済)、ハードウェアによってセットされます。このビットは、そこに 0 を書き込むか HASH\_CR レジスタの INIT ビットに 1 を書き込むことによってクリアされます。
- 0 : HASH\_Hx レジスタに利用可能なダイジェストはありません。
  - 1 : ダイジェスト計算完了。HASH\_Hx レジスタに利用可能なダイジェストがあります。HASH\_IMR レジスタの DCIE ビットがセットされている場合、割り込みが生成されます。

### ビット 0 **DINIS** : データ入力割り込みステータス

- このビットは、入力バッファが新しいブロックを取得する準備ができたときにハードウェアによってセットされます (空き領域が 16 か所)。このビットは、そこに 0 を書き込むか HASH\_DIN レジスタに書き込みを行うことによってクリアされます。
- 0 : 入力バッファの空き領域は 16 か所未満です。
  - 1 : 入力バッファに新しいブロックを取り込むことができます。HASH\_IMR レジスタの DINIE ビットがセットされている場合、割り込みが生成されます。

## 21.4.7 HASH コンテキストスワップレジスタ (HASH\_CSRx)

アドレスオフセット : 0x0F8 から 0x1CC

- HASH\_CSR0 レジスタ : リセット値は 0x0000 0002
- 他のレジスタ : リセット値は 0x0000 0000、ただし、HASH\_CSR2 レジスタのリセット値が 0x2000 0000 であるデバイスを除く

これらのレジスタで、ハッシュプロセッサの内部レジスタの状態がすべて分かるため、別のタスクがすでにハッシュプロセッサを使用している間に、優先度の高いタスクがハッシュプロセッサを使用する必要があってコンテキストスワップを行う必要があるときに便利です。

そのような事象が起こった場合、HASH\_CSRx レジスタを読み出し、その値をシステムメモリ空間のどこかに保存する必要があります。このようにして、優先権のあるタスクがハッシュプロセッサを使用することができ、ハッシュ計算が終了した時点で、保存したコンテキストをメモリから読み出し、HASH\_CSRx レジスタに書き戻すことができます。

### HASH\_CSRx

アドレスオフセット : 0x0F8 から 0x1CC

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



## 21.4.8 HASH レジスタマップ

表 122 に HASH レジスタマップの概要およびリセット値を示します。

表 122. HASH レジスタマップとリセット値

オフ セット	レジスタ名 リセット値	レジスタサイズ																																	
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	HASH_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[1]	予約済み	LKEY	Res.	Res.	Res.	MDMAT	DINNE	NBW				ALGO[0]	MODE	DATATYPE	DMAE	INIT	Res.	Res.	
	リセット値															0		0					0	0	0	0	0	0	0	0	0	0			
0x04	HASH_DIN	DATAIN																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	HASH_STR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																		
0x0C	HASH_HR0	H0																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	HASH_HR1	H1																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	HASH_HR2	H2																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	HASH_HR3	H3																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1C	HASH_HR4	H4																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x20	HASH_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x24	HASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	DMAE	DCIS	DINIS
	リセット値																															0	0	0	1
0xF8	HASH_CSR0	CSR0																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
		...																																	
0x1CC	HASH_CSR53	CSR53																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		予約済み																																	
0x310	HASH_HR0	H0																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x314	HASH_HR1	H1																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x318	HASH_HR2	H2																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x31C	HASH_HR3	H3																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x320	HASH_HR4	H4																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x324	HASH_HR5	H5																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x328	HASH_HR6	H6																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x32C	HASH_HR7	H7																																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

## 22 高機能制御タイマ (TIM1/TIM8)

### 22.1 TIM1/TIM8 の概要

高機能制御タイマ (TIM1/TIM8) は、プログラム可能なプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM、デッドタイムを挿入した相補 PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

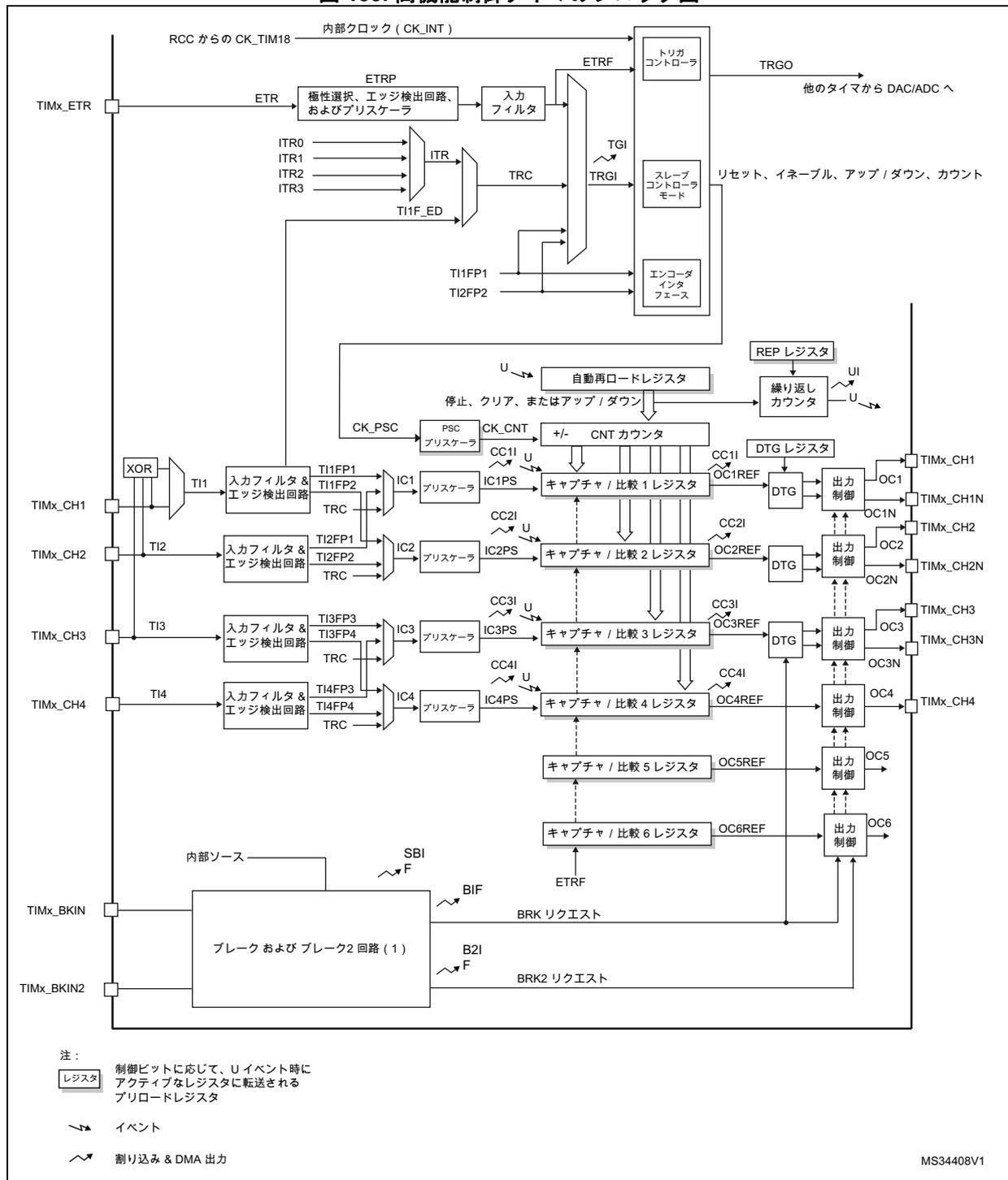
高機能制御タイマ (TIM1/TIM8) と汎用タイマ (TIMx) は、互いに独立しており、リソースを共有しません。これらのタイマは、[セクション 22.3.26](#)に示すように、相互に同期させることができます。

### 22.2 TIM1/TIM8 の主な機能

TIM1/TIM8 タイマの主な機能は、次のとおりです。

- 16 ビットのアップカウンタ、ダウンカウンタ、アップ/ダウン自動再ロードカウンタ。
- 16 ビットのプログラム可能なプリスケラ。カウンタクロック周波数を 1 から 65536 の間で分周でき、分周比の動作中の変更も可能。
- 次の機能を持つ、最大 6 つの独立チャネル。
  - 入力キャプチャ (ただしチャネル 5 および 6)
  - 出力比較
  - PWM 生成 (エッジアラインモードとセンターアラインモード)
  - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をユーザが選択可能な安全な設定にする 2 つのブレイク入力。
- 以下のイベント時の割り込み/DMA 生成。
  - 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
  - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
  - 入力キャプチャ
  - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 133. 高機能制御タイマのブロック図



1. 詳細については、[図 175: ブレーク および ブレーク2 回路の概要](#)を参照してください。

## 22.3 TIM1/TIM8 の機能説明

### 22.3.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC)
- 自動再ロードレジスタ (TIMx\_ARR)
- 繰り返しカウンタレジスタ (TIMx\_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー (またはダウンカウント時はアンダーフロー) に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK\_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx\_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 134](#) と [図 135](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 134. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

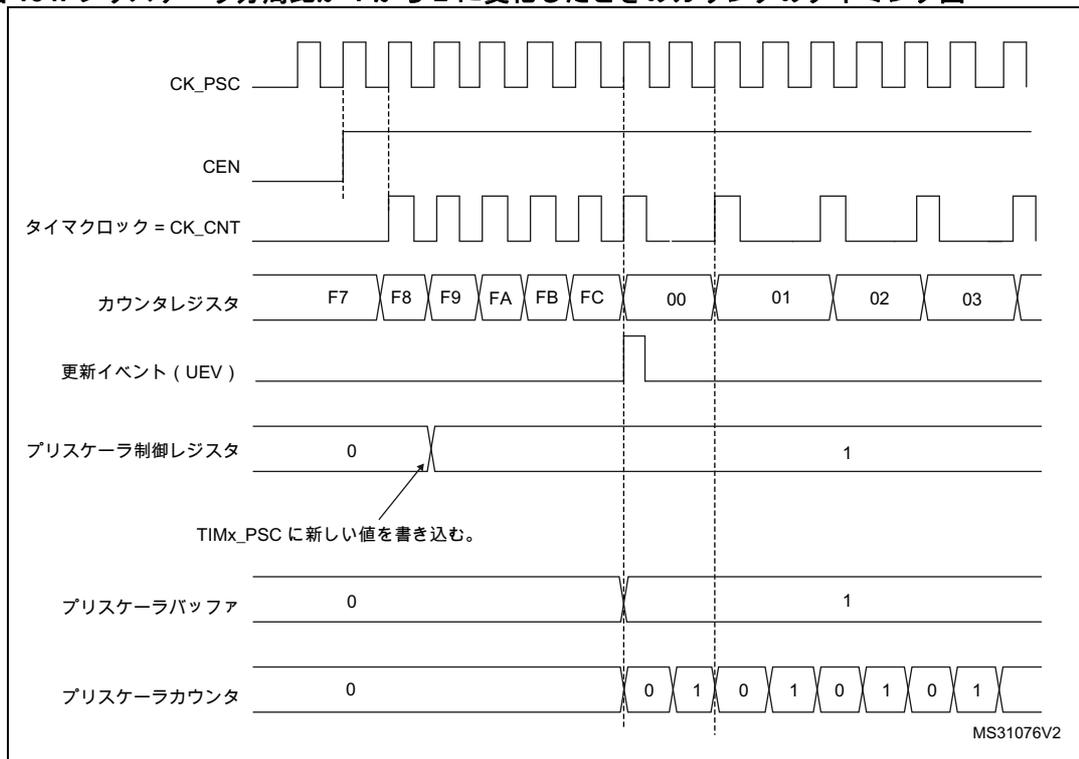
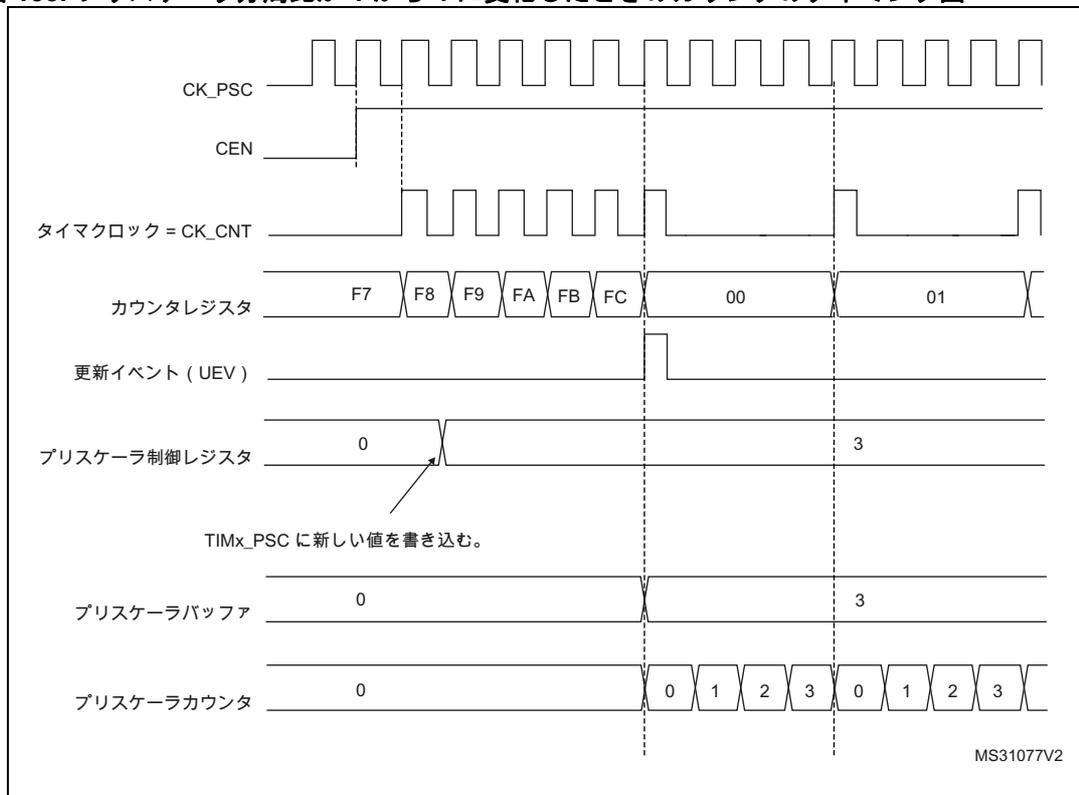


図 135. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 22.3.2 カウンタモード

### アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx\_RCR) + 1 までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 136. 内部クロック分周比が 1 の場合のカウンタのタイミング図

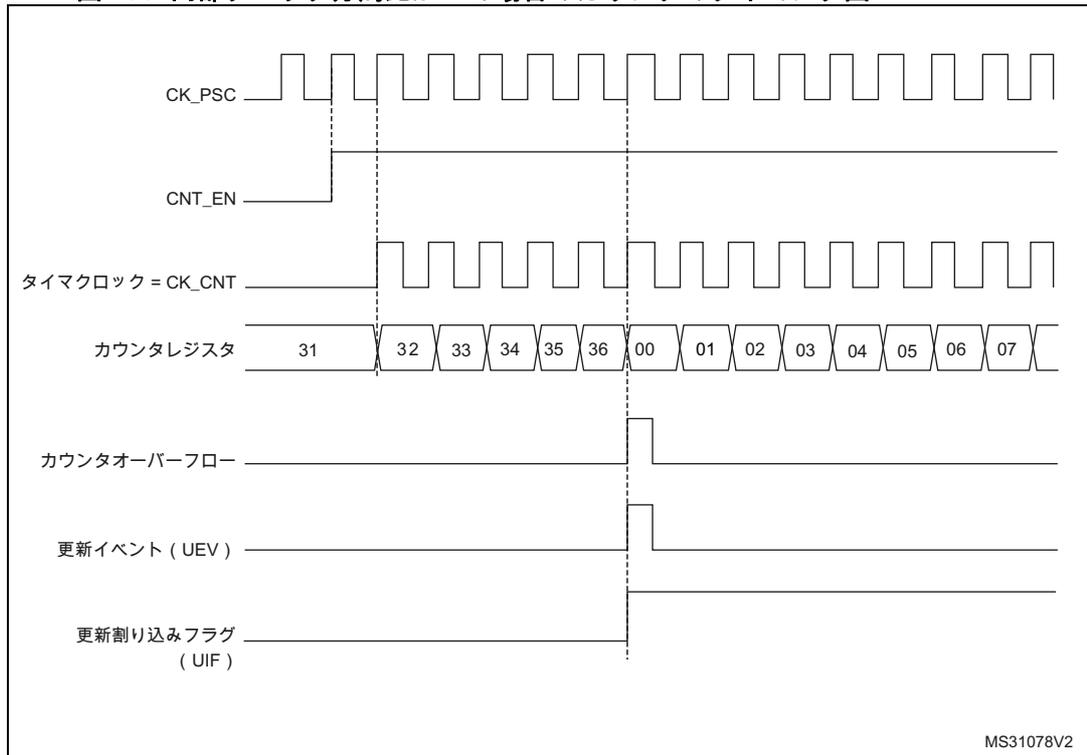


図 137. 内部クロック分周比が 2 の場合のカウンタのタイミング図

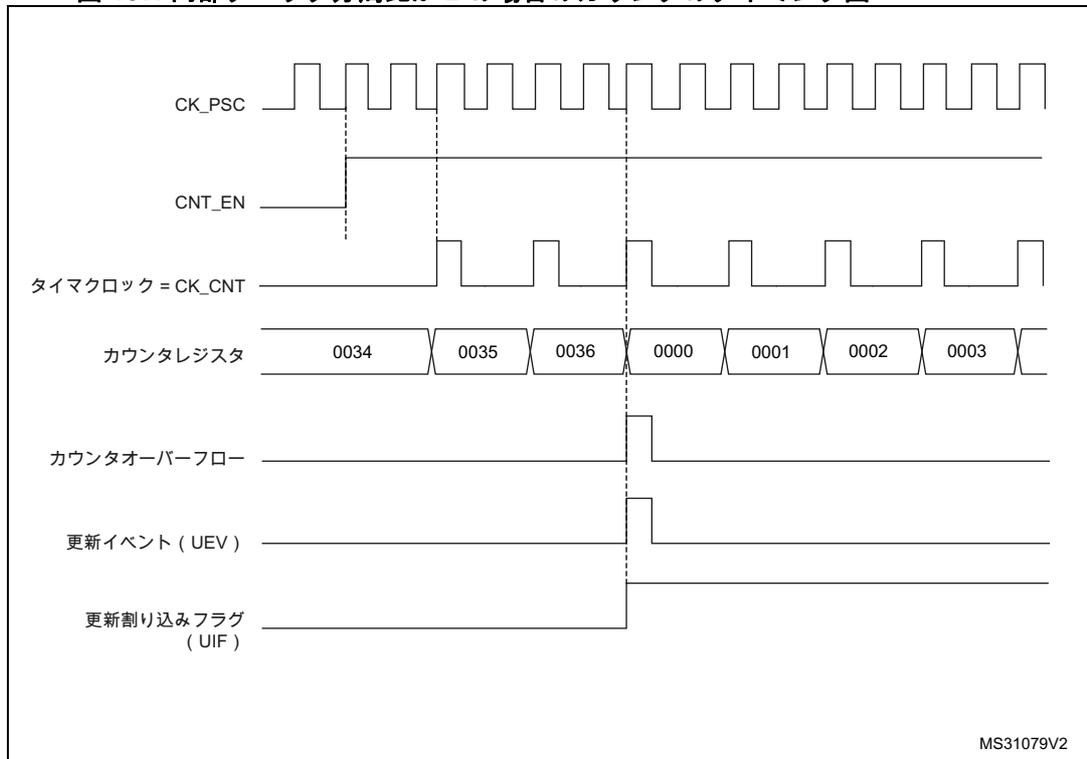


図 138. 内部クロック分周比が 4 の場合のカウンタのタイミング図

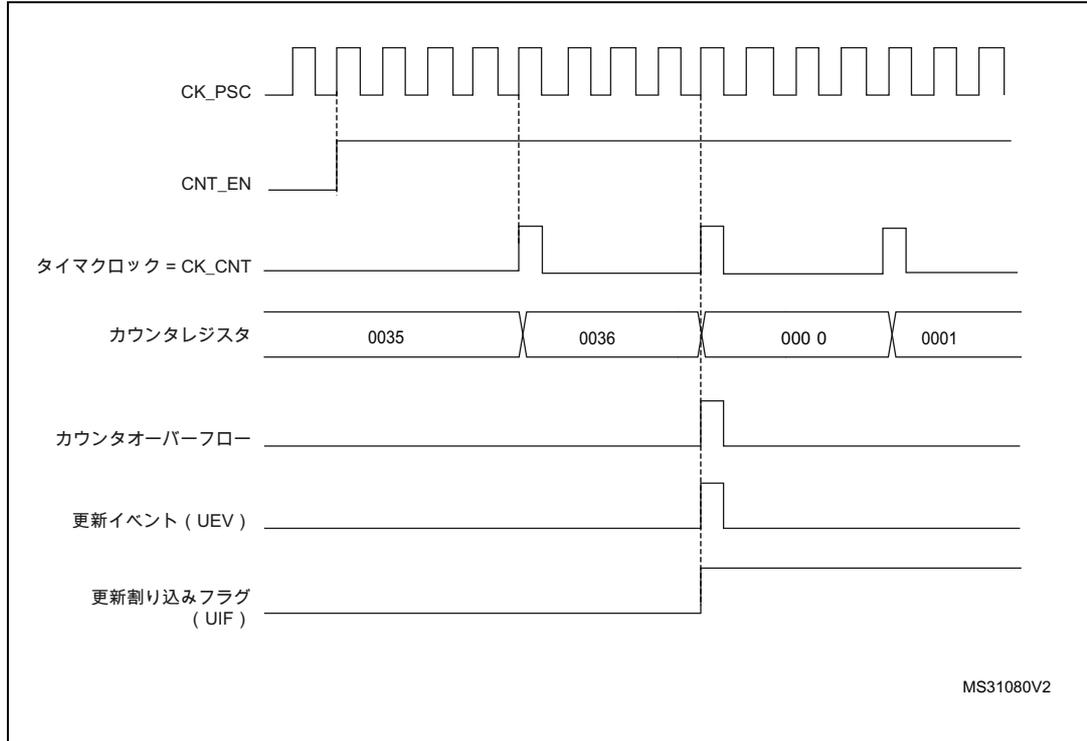


図 139. 内部クロック分周比が N の場合のカウンタのタイミング図

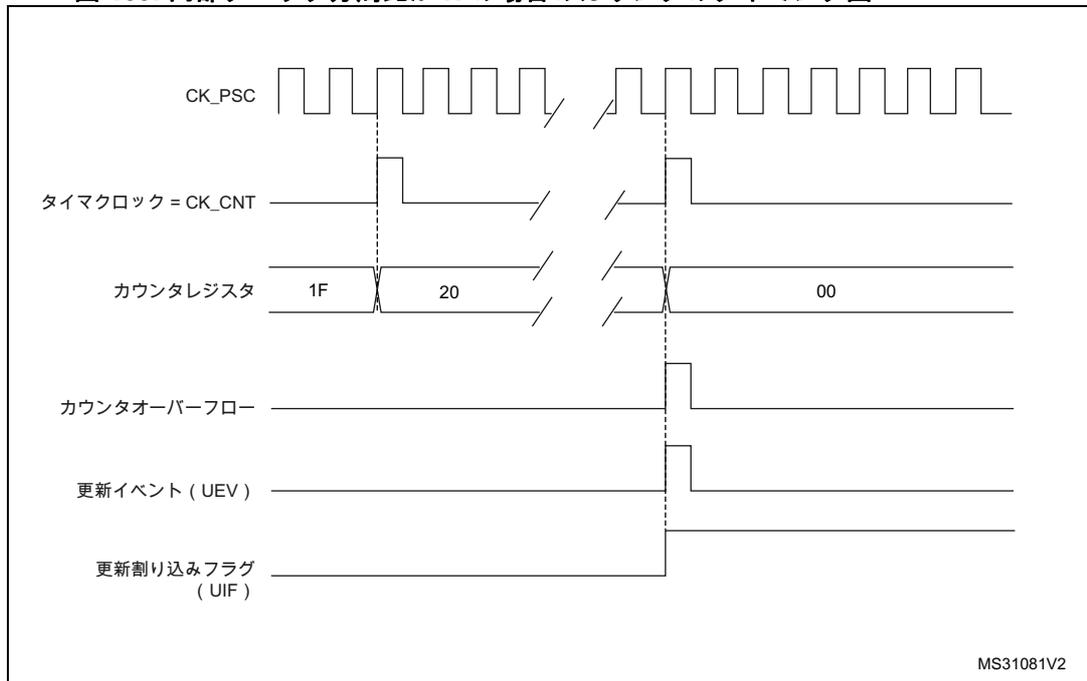


図 140. ARPE = 0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

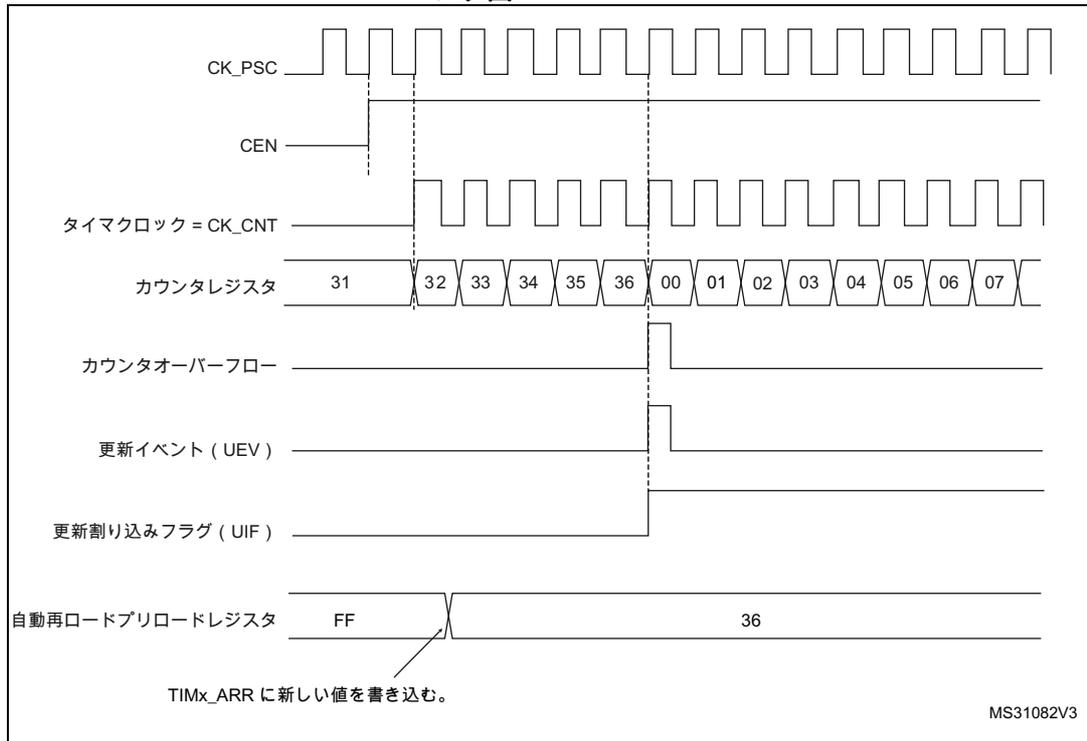
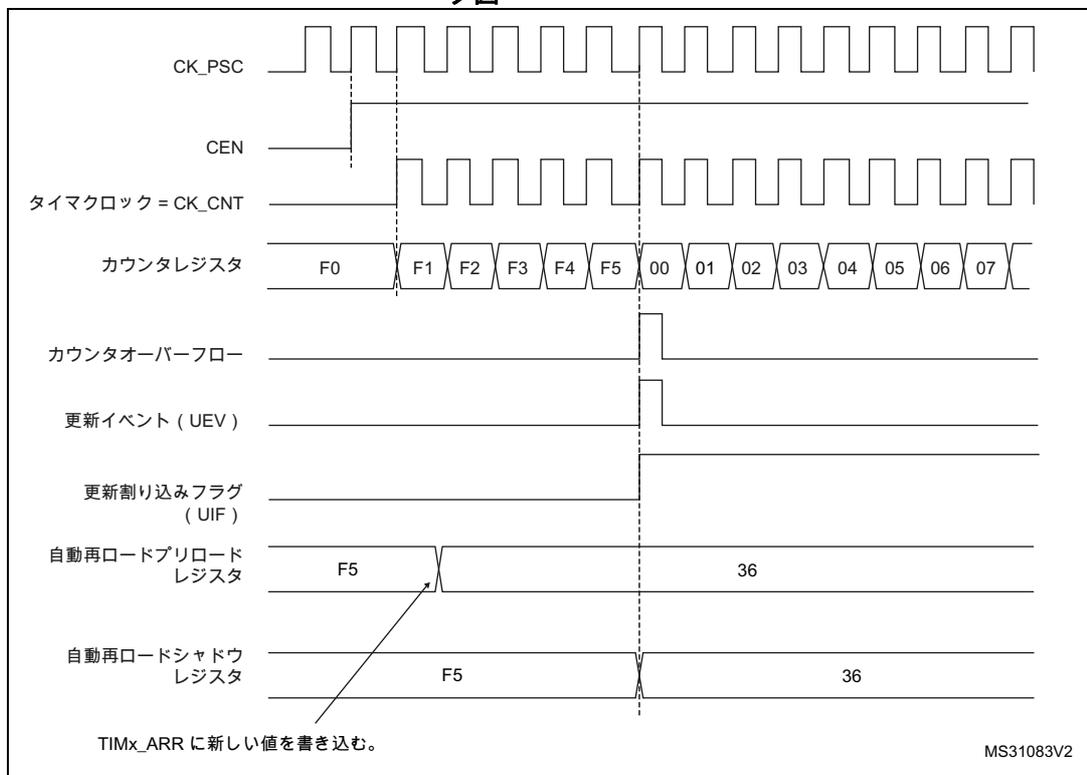


図 141. ARPE = 1 (TIMx\_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



## ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx\_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx\_RCR) + 1 までダウンカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのアンダーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 142. 内部クロック分周比が 1 の場合のカウンタのタイミング図

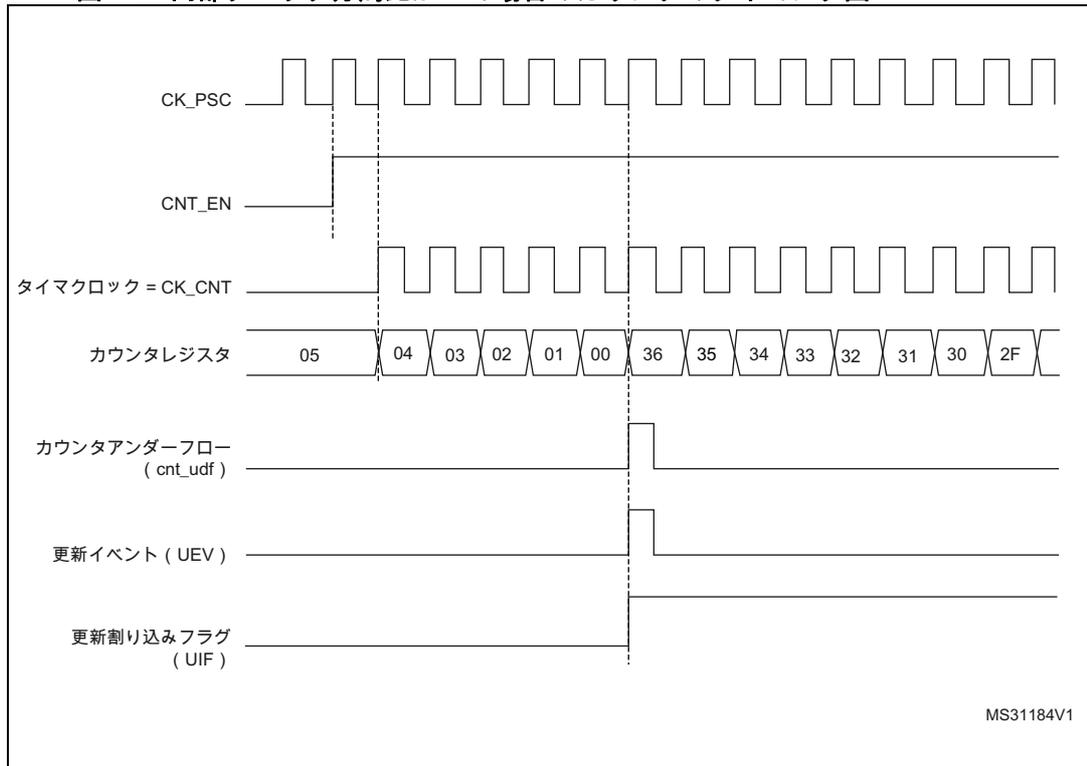


図 143. 内部クロック分周比が 2 の場合のカウンタのタイミング図

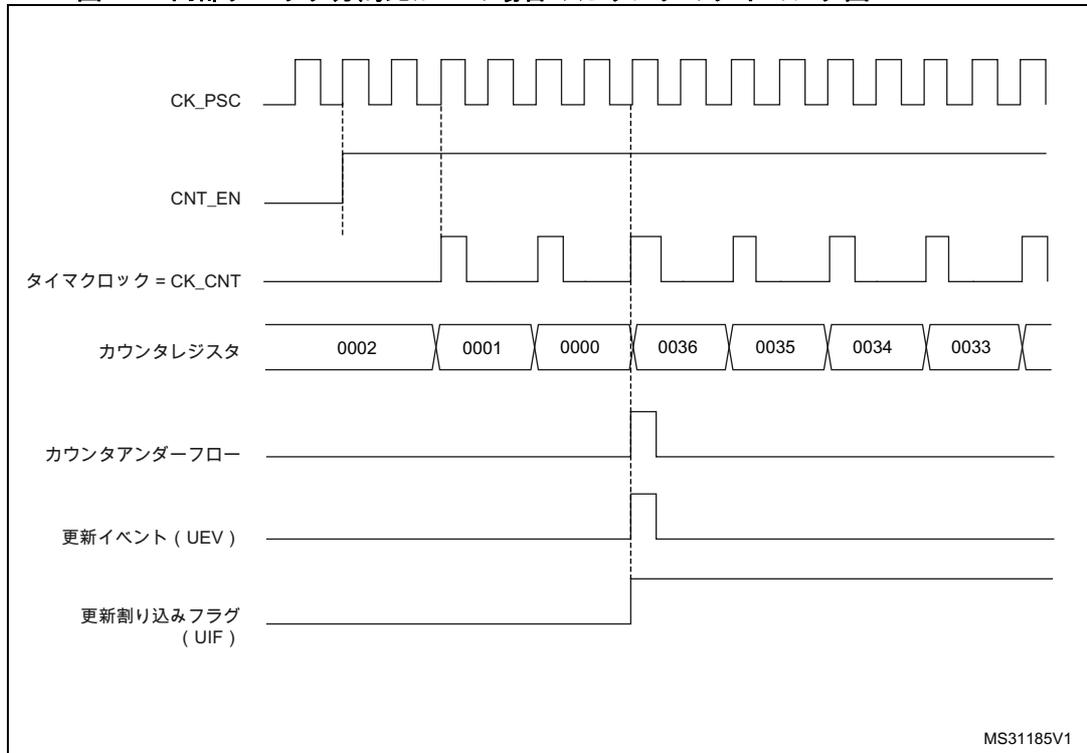


図 144. 内部クロック分周比が 4 の場合のカウンタのタイミング図

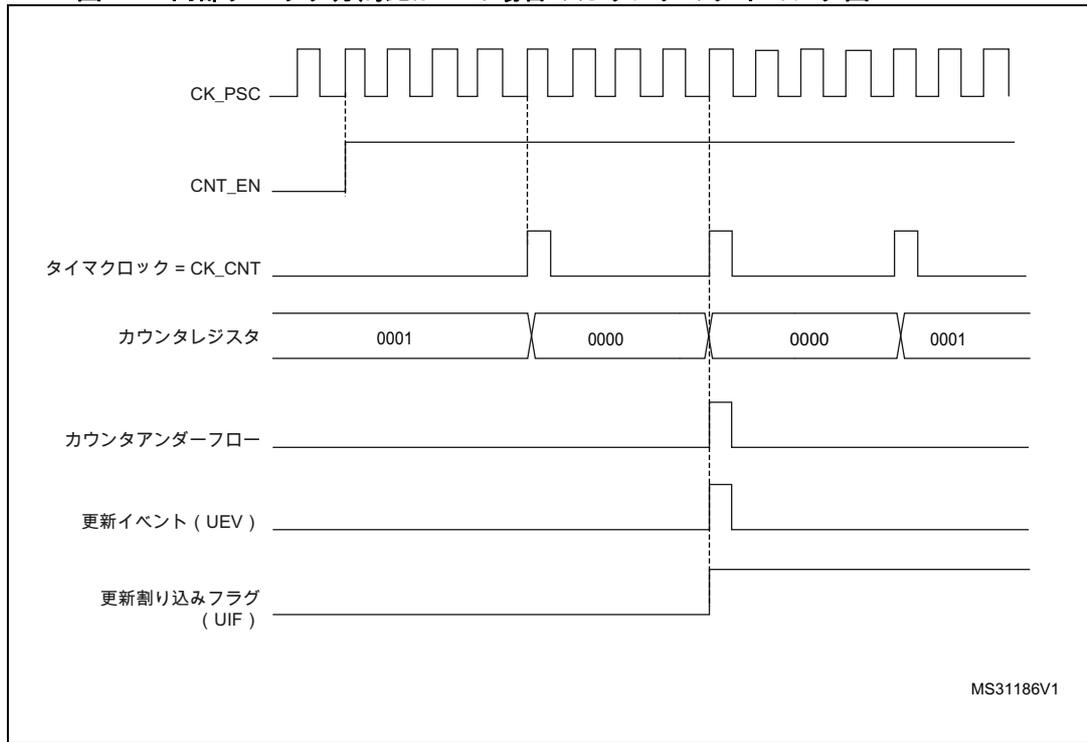


図 145. 内部クロック分周比が N の場合のカウンタのタイミング図

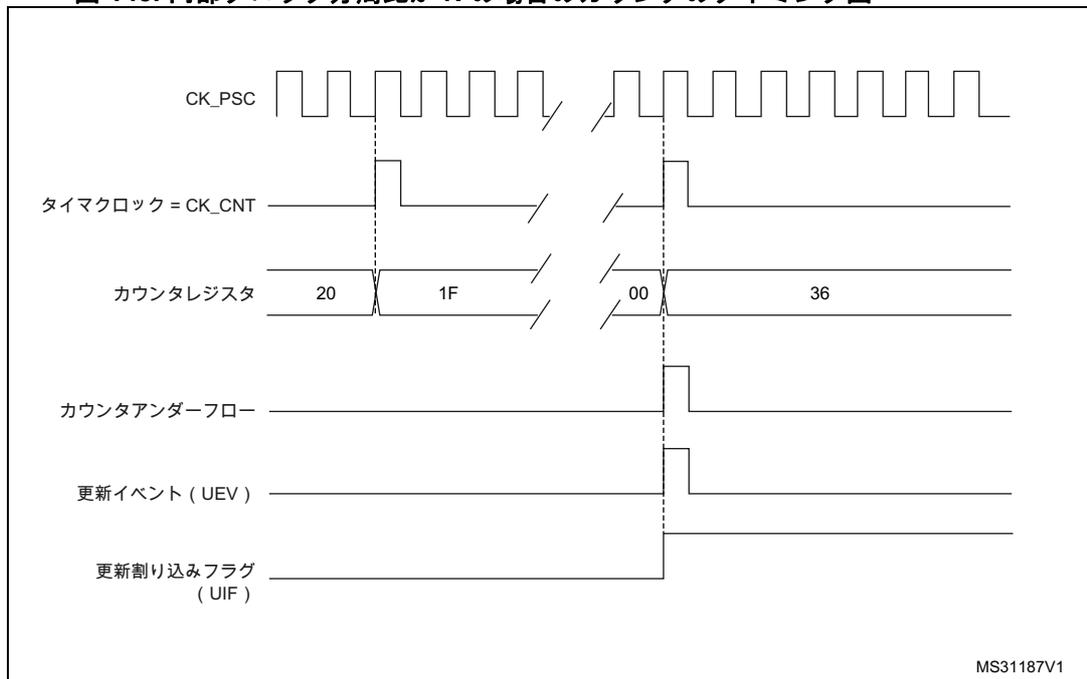
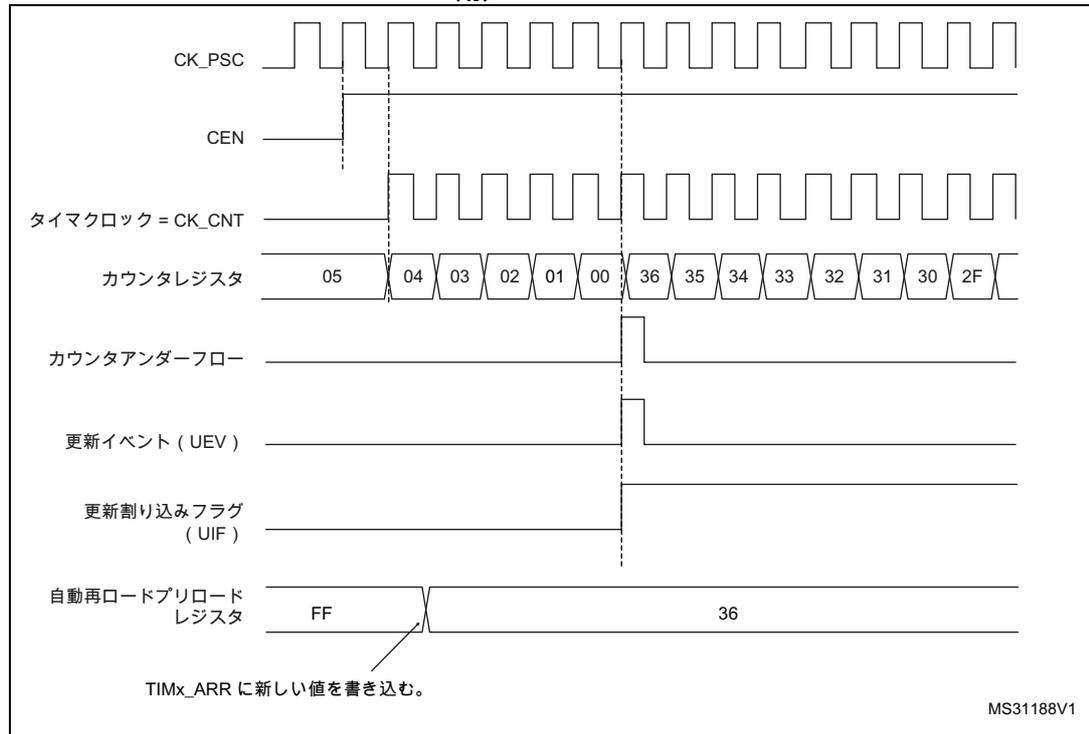


図 146. 繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図 (不使用)



## センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センターアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS=11) にセットされます。

このモードでは、TIMx\_CR1 レジスタの方向ビット (DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合、UG ビットをセットすると UEV 更新イベントが生成されますが、UIF フラグはセットされません (した

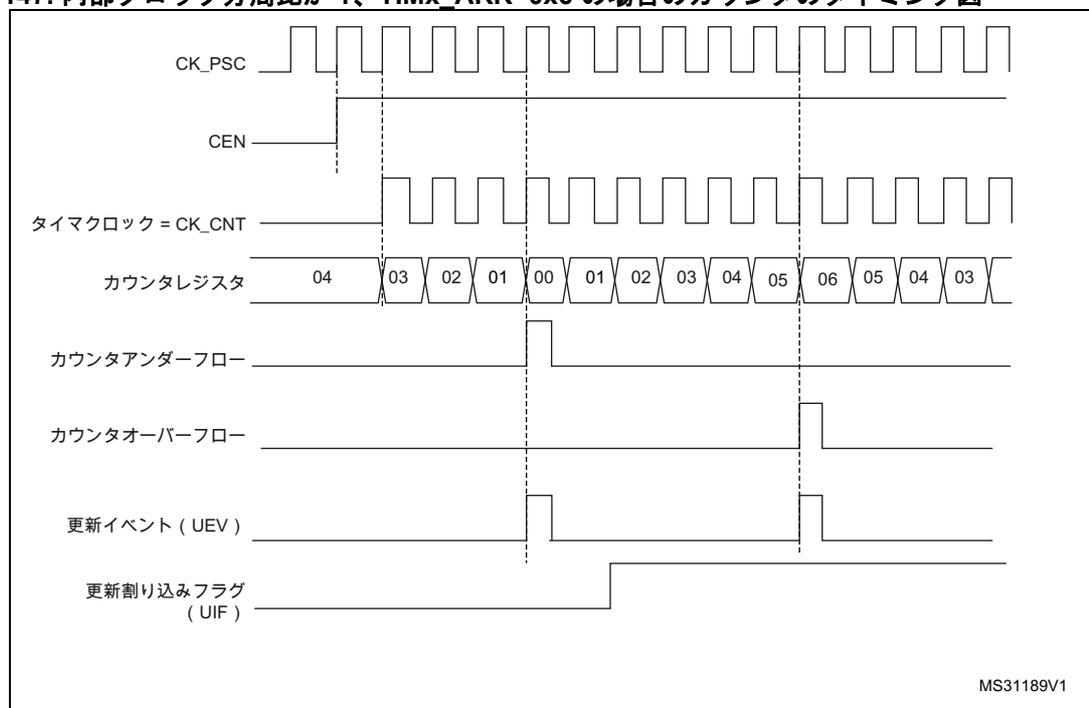
がって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 147. 内部クロック分周比が 1、TIMx\_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 22.4 : TIM1/TIM8 レジスタ](#)を参照)。

図 148. 内部クロック分周比が 2 の場合のカウンタのタイミング図

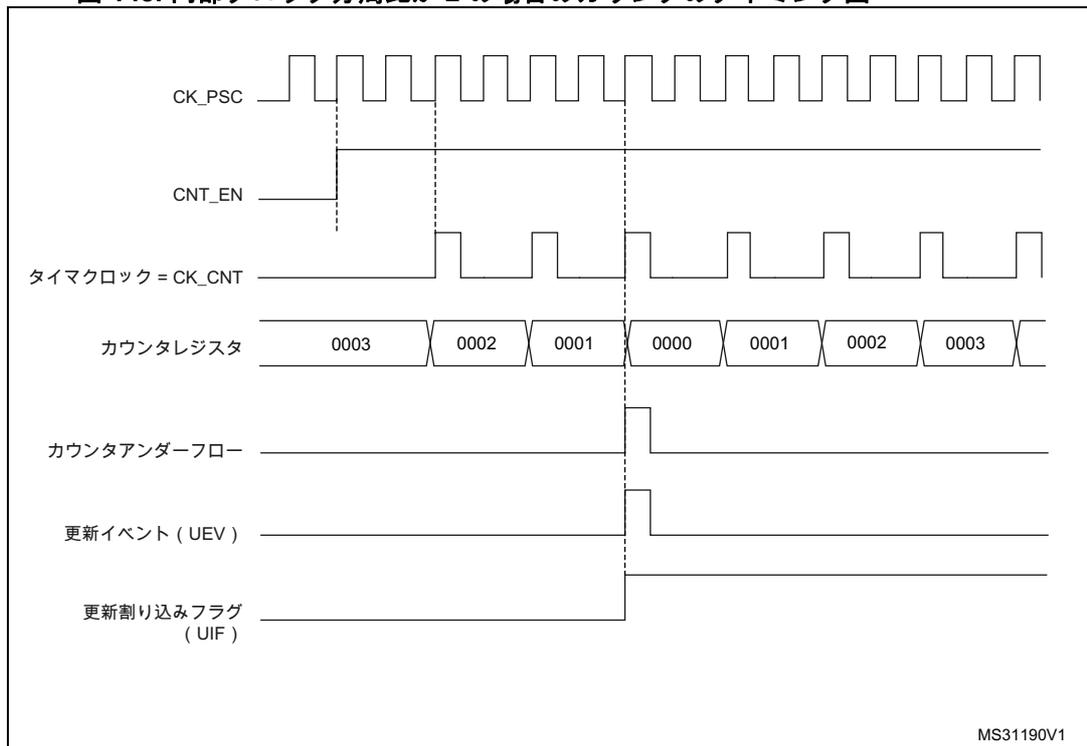


図 149. 内部クロック分周比が 4、TIMx\_ARR=0x36 の場合のカウンタのタイミング図

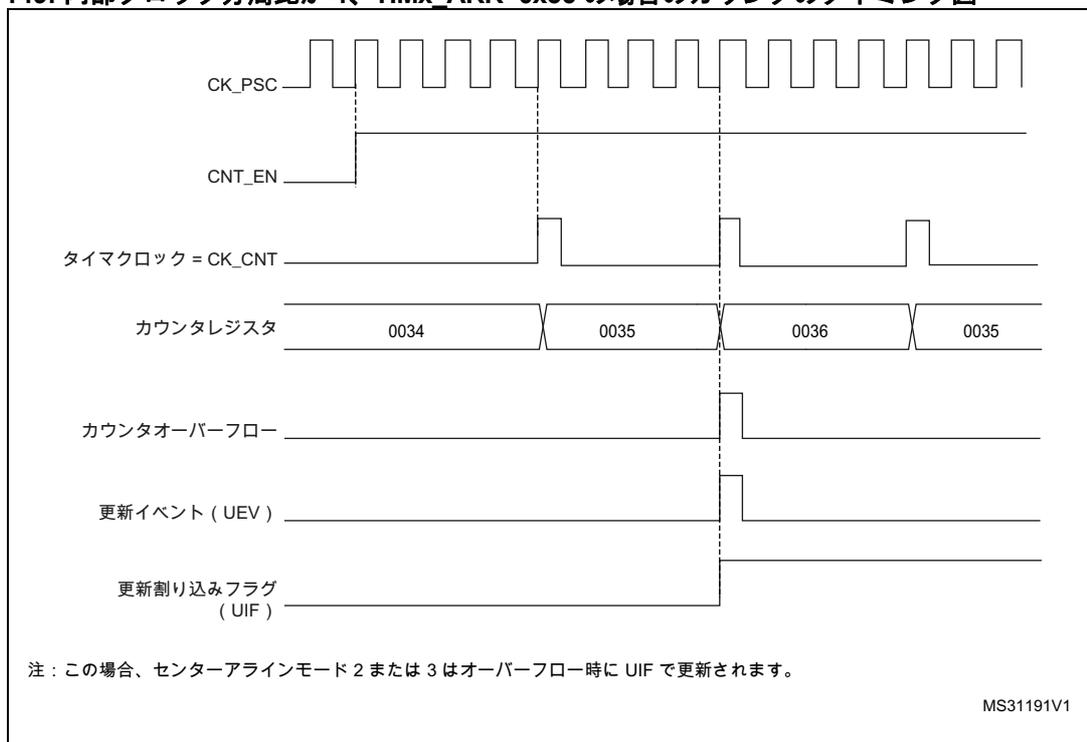


図 150. 内部クロック分周比が N の場合のカウンタのタイミング図

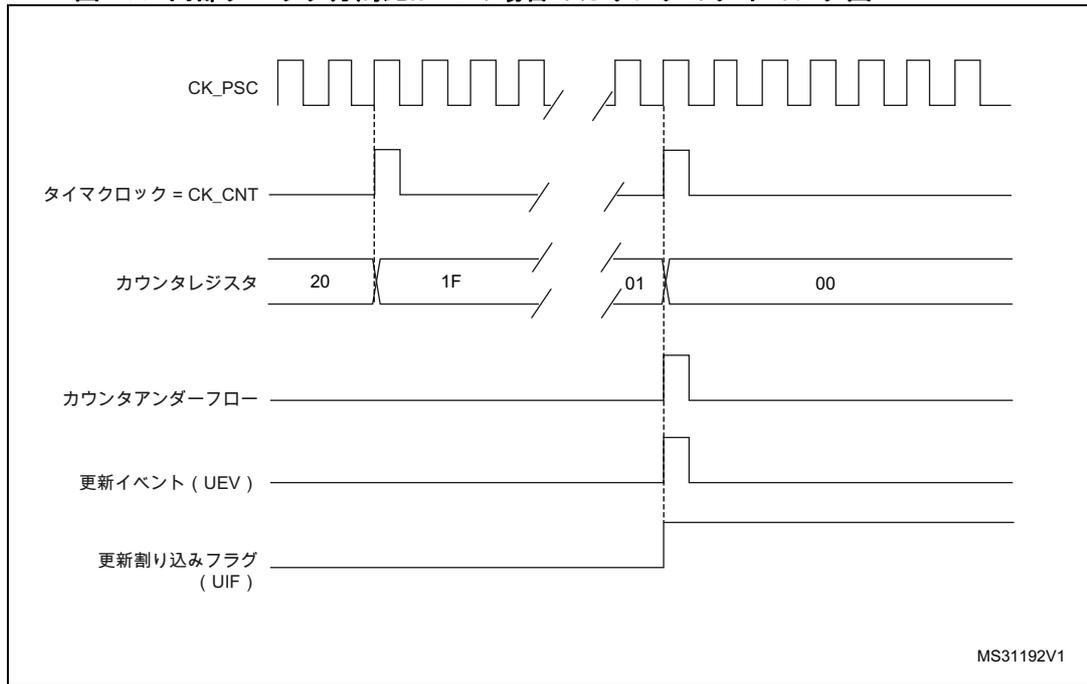


図 151. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図

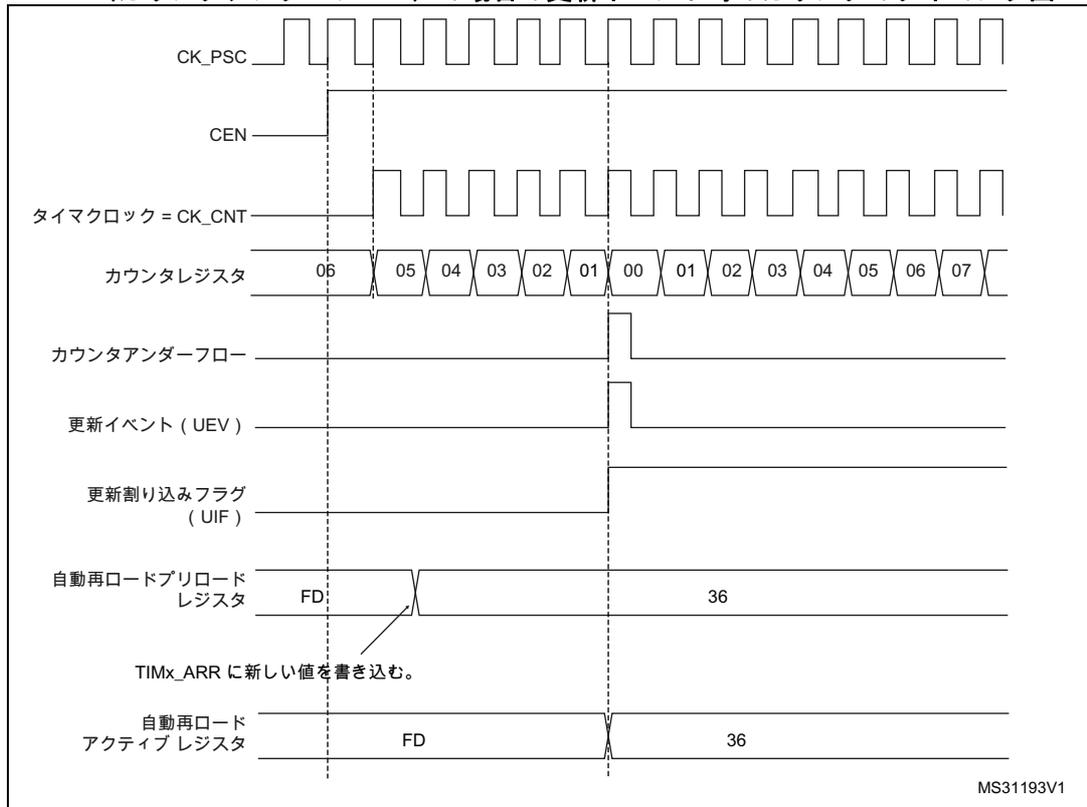
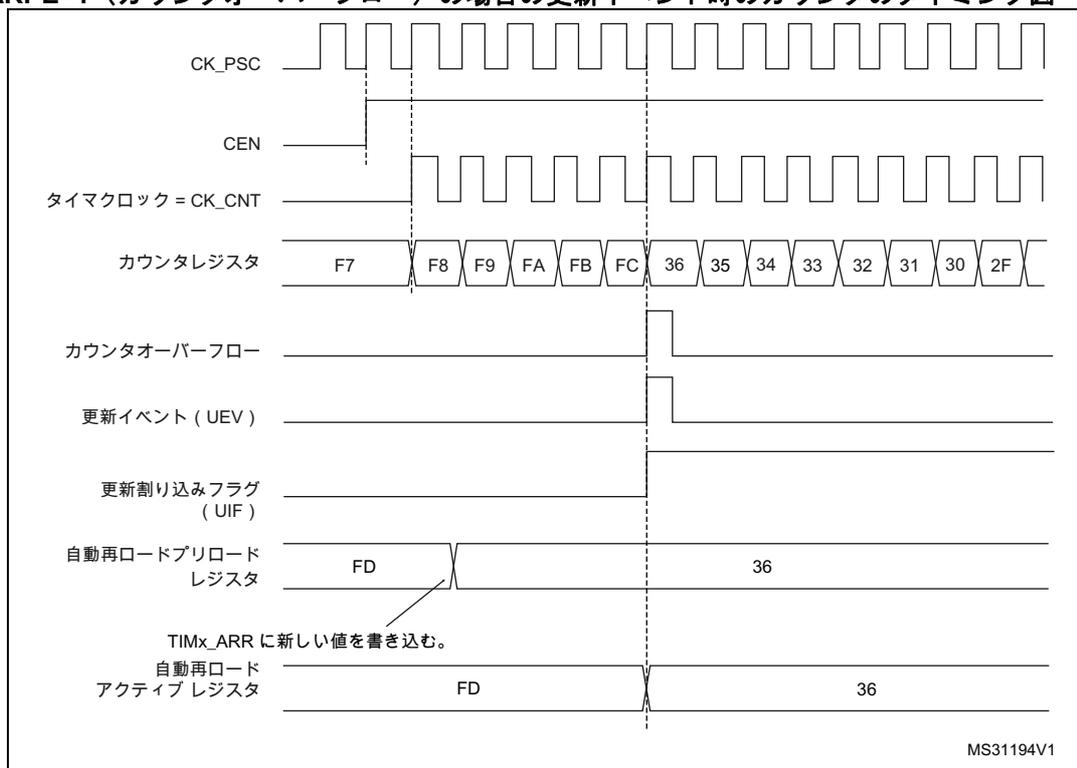


図 152. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



## 22.3.3 繰り返しカウンタ

セクション 22.3.1: タイムベースユニットに、カウンタオーバーフロー/アンダーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰り返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx\_RCR 繰り返しカウンタレジスタの値を N とすると、N+1 回目のカウンタオーバーフローまたはアンダーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx\_ARR 自動再ロードレジスタ、TIMx\_PSC プリスケアラレジスタ、比較モードの TIMx\_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

繰り返しカウンタは、次の場合にデクリメントします。

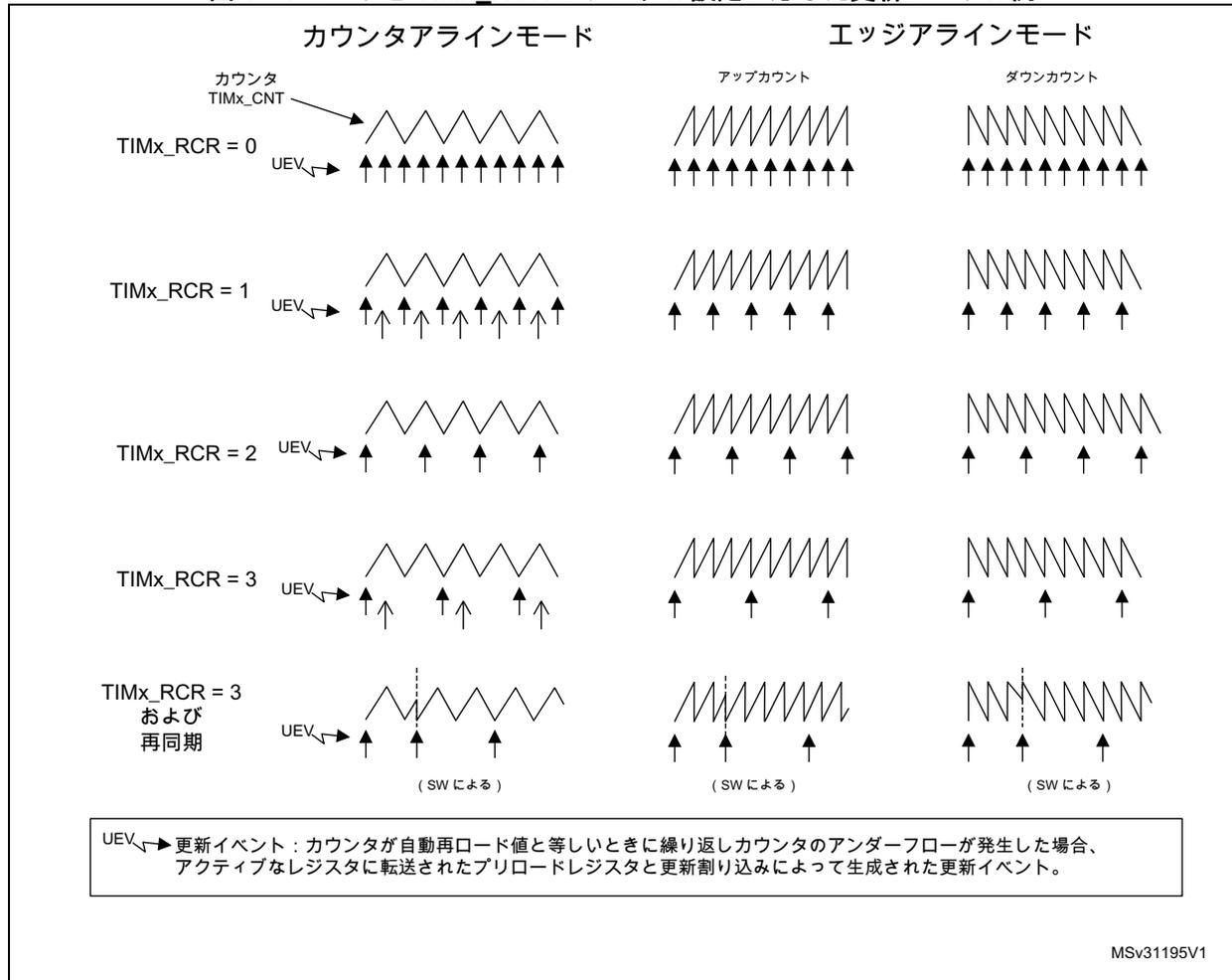
- アップカウントモードで、カウンタオーバーフローごと
- ダウンカウントモードで、カウンタアンダーフローごと
- センターアラインモードで、カウンタオーバーフローとカウンタアンダーフローごと最大繰り返し回数は 32768 PWM サイクルに限られますが、PWM 周期ごとにデューティサイクルを 2 回更新することが可能になります。センターアラインモードで比較レジスタの値を PWM 周期あたり 1 回のみ更新するときには、パターンが対称なので、最大精度は  $2xT_{ck}$  です。

繰り返しダウンカウンタは自動再ロードタイプです。繰り返しの回数は、TIMx\_RCR レジスタの値によって定義されたとおりに維持されます (図 153 を参照してください)。ソフトウェアによって (TIMx\_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰り返しカウンタの値にかかわらず常にイベントが発生し、繰り返しカウンタに TIMx\_RCR レジスタの内容が再ロードされます。

センターラインモードでは、RCR が奇数の場合、RCR レジスタが書き込まれたタイミングおよびカウンタが開始されたタイミングに応じてオーバーフローまたはアンダーフロー時に更新イベントが発生します。カウントの開始前に RCR が書き込まれた場合は、オーバーフローで、UEV が発生します。カウントの開始後に RCR が書き込まれた場合は、アンダーフローで UEV が発生します。

たとえば、RCR = 3 の場合、RCR の書き込みタイミングに応じて 4 回目のオーバーフローイベントまたはアンダーフローイベントごとに UEV が発生します。

図 153. モードと TIMx\_RCR レジスタの設定に応じた更新レートの例



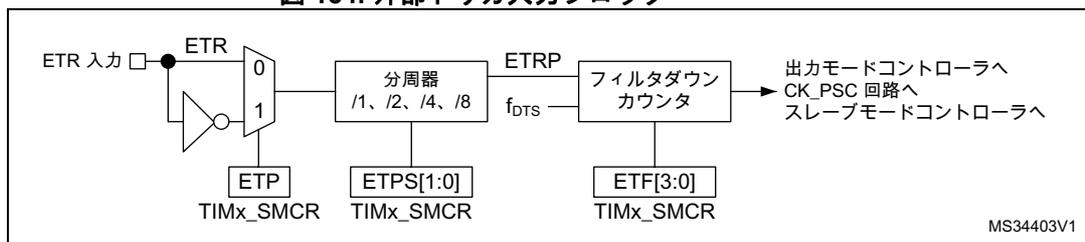
## 22.3.4 外部トリガ入力ブロック

タイマには外部トリガ入力 ETR 機能があります。以下の目的で使用できます。

- 外部クロック (外部クロックモード 2、[セクション 22.3.5](#) を参照)
- スレープモードのトリガ ([セクション 22.3.25](#) を参照)
- サイクルごとの電流調整の PWM リセット入力 ([セクション 22.3.7](#) を参照)

以下の [図 154](#) は、ETR の入力条件付けについて説明しています。入力の極性は、TIMxSMCR レジスタの ETP ビットで定義されています。トリガは ETPS[1:0] ビットフィールドでプログラムされた分周器でプリスケールし、ETF[3:0] ビットフィールドでデジタル的にフィルタリングすることができます。

図 154. 外部トリガ入力ブロック



## 22.3.5 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

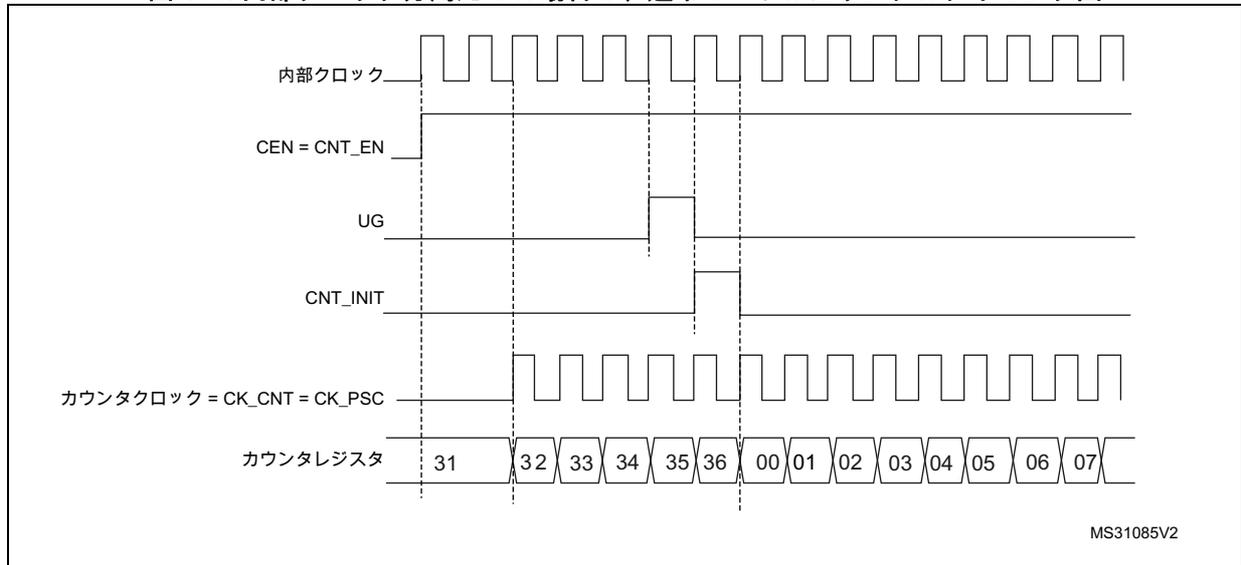
- 内部クロック (CK\_INT)
- 外部クロックモード 1 : 外部入力ピン
- 外部クロックモード 2 : 外部トリガ入力 ETR
- エンコーダモード

### 内部クロックソース (CK\_INT)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx\_CR1 レジスタ)、および UG ビット (TIMx\_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK\_INT が供給されます。

図 155 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

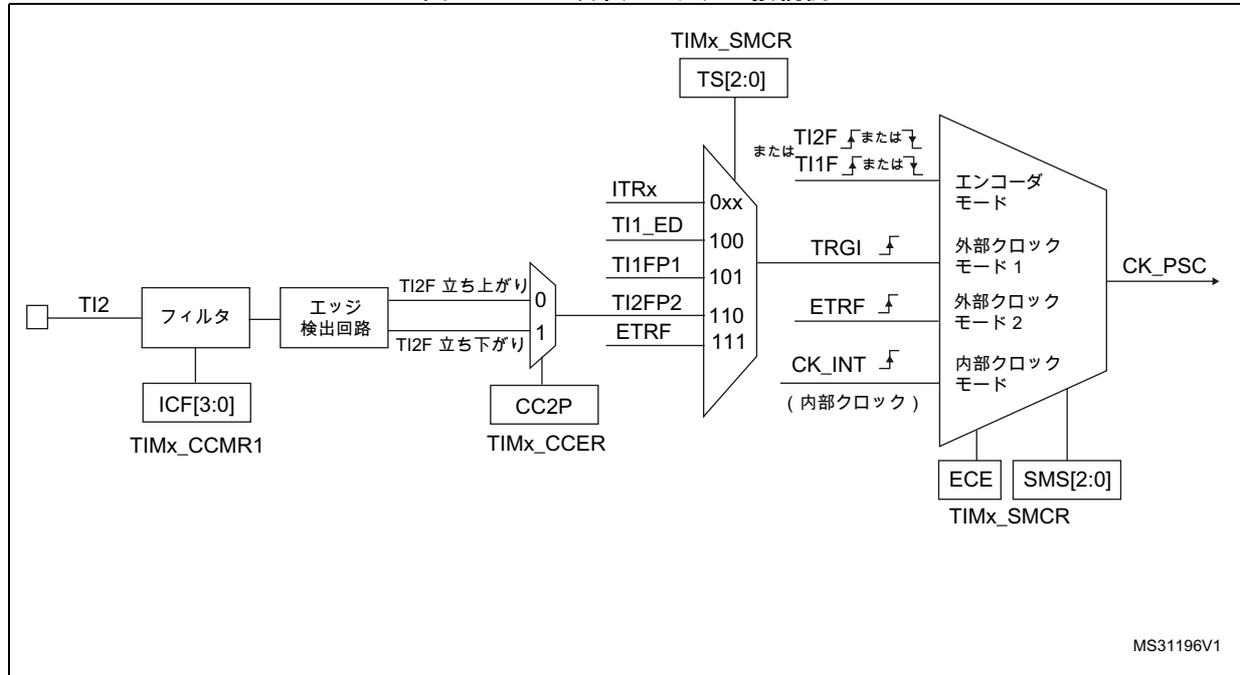
図 155. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



## 外部クロックソースモード 1

このモードは TIMx\_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 156. TI2 外部クロックの接続例



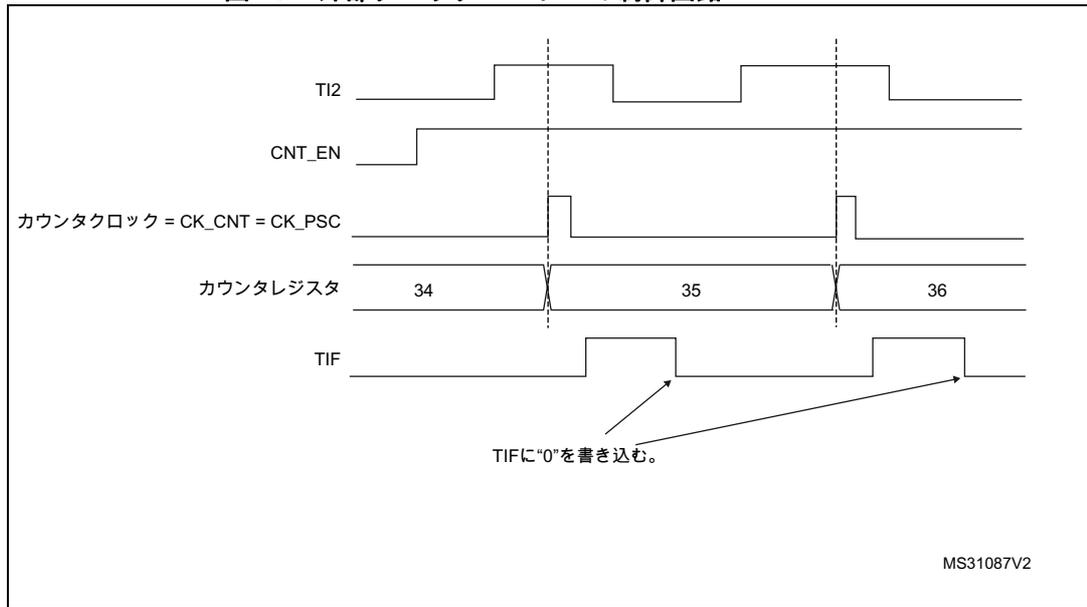
たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx\_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。
3. CC2P=0 と CC2NP=0 を TIMx\_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx\_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx\_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

**注：** キャプチャプリスケアラはトリガには使用されないため、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 157. 外部クロックモード 1 の制御回路



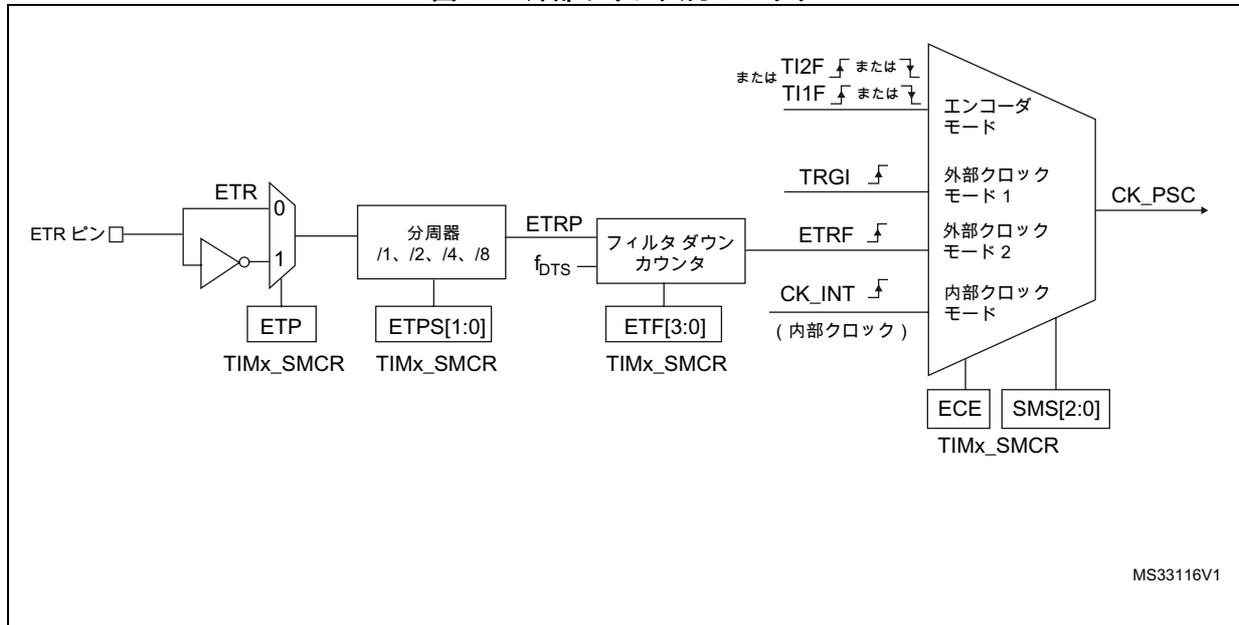
## 外部クロックソースモード 2

このモードは、TIMx\_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 158 に、外部トリガ入力ブロックの概要を示します。

図 158. 外部トリガ入力ブロック



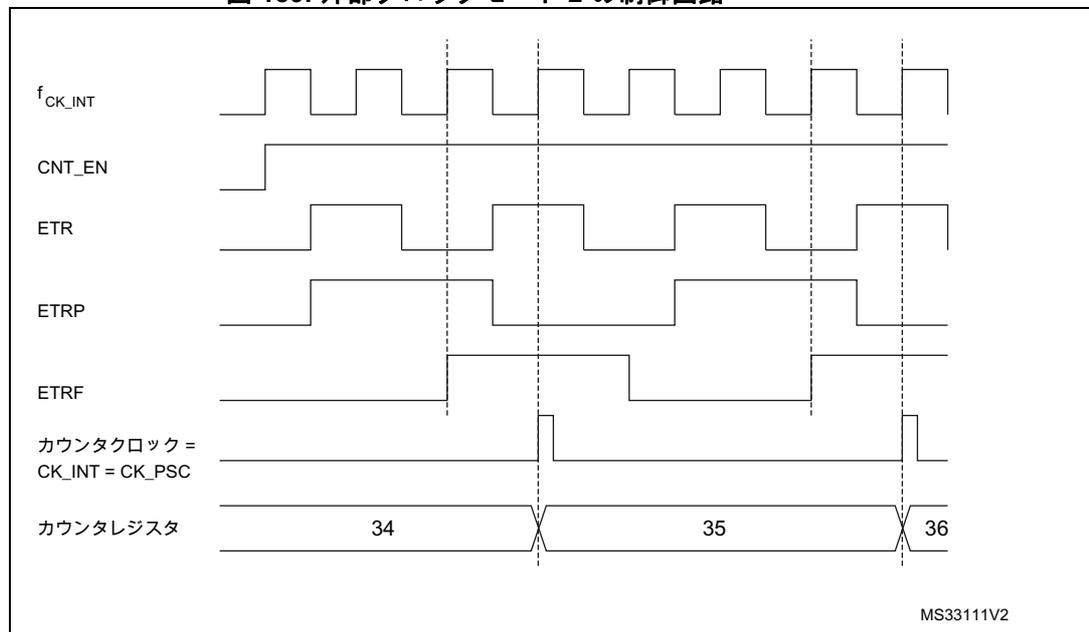
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx\_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx\_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx\_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx\_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 159. 外部クロックモード 2 の制御回路



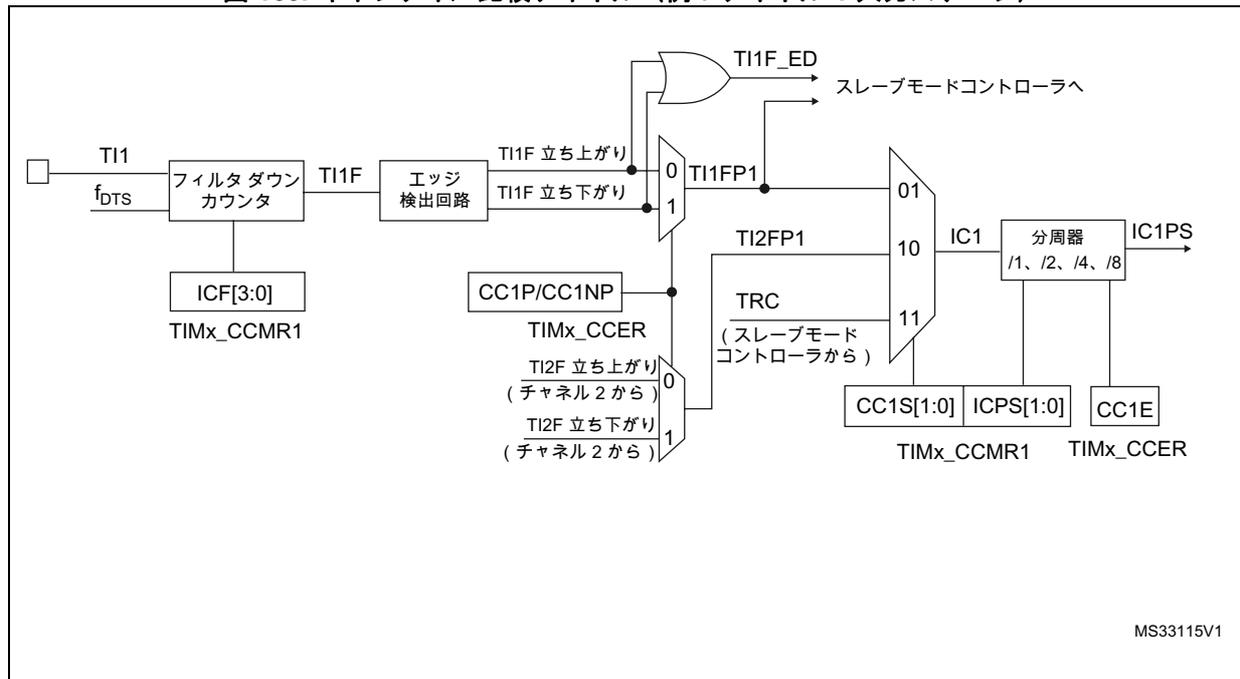
## 22.3.6 キャプチャ／比較チャンネル

各キャプチャ／比較チャンネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（チャンネル 5 および 6 を除くデジタルフィルタ、マルチプレクサ、プリスケアラ）、および出カステージ（比較回路と出力制御）から構成されています。

図 160 から 図 163 に、1 つのキャプチャ／比較チャンネルの概要を示します。

入カステージは、対応する Tix 入力をサンプリングして、フィルタリングを行った TixF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号（TixFPx）を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ（ICxPS）に渡されます。

図 160. キャプチャ／比較チャンネル（例：チャンネル 1 入カステージ）



出カステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 161. キャプチャ/比較チャンネル 1 メイン回路

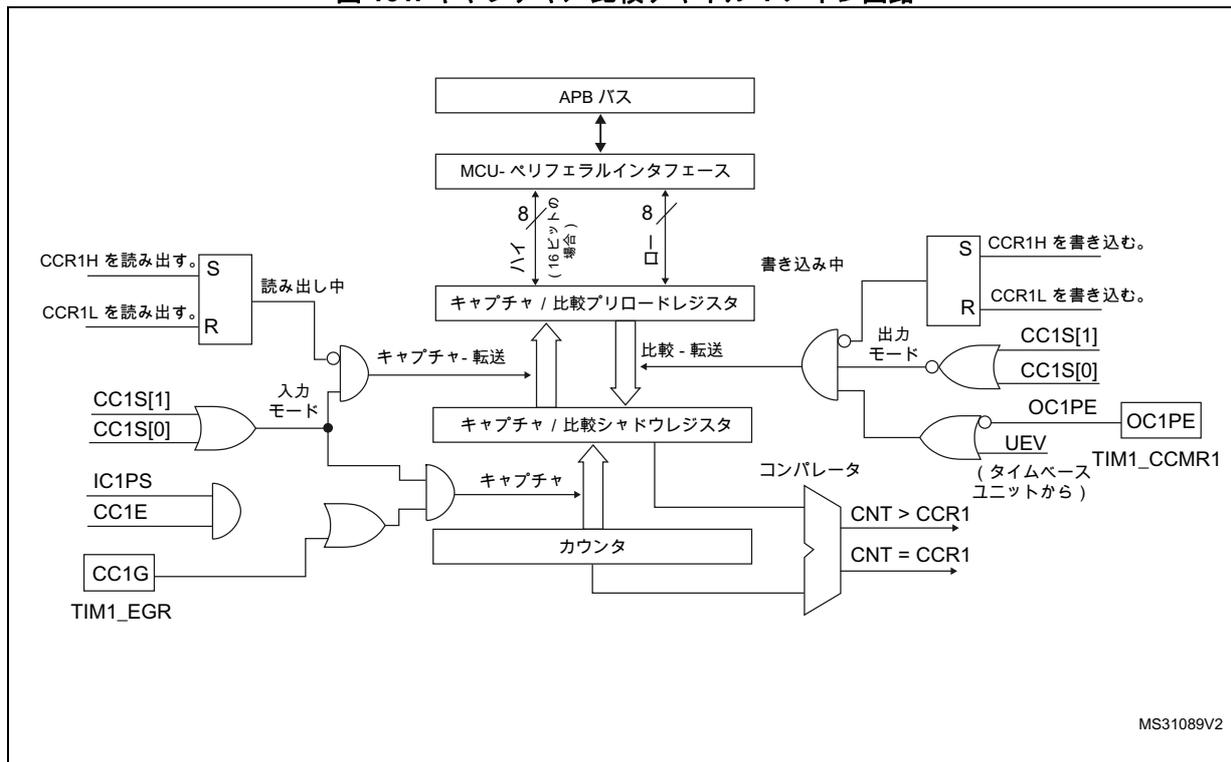
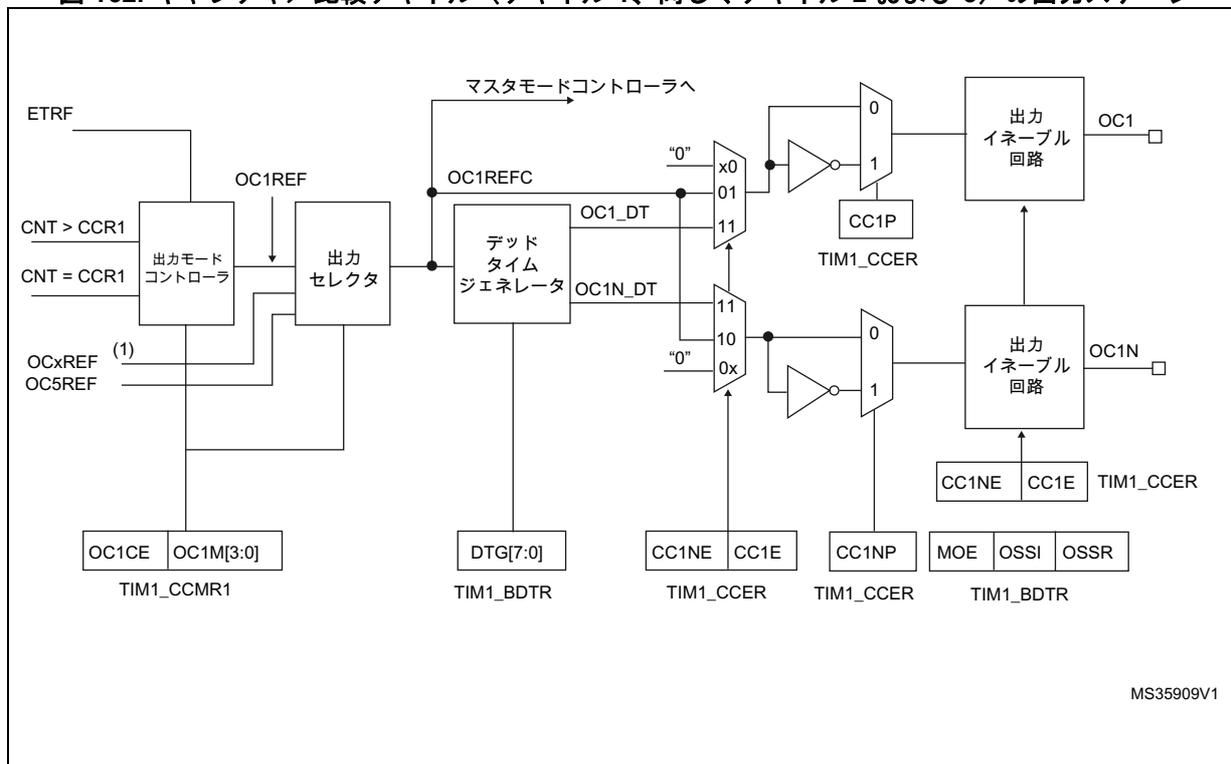


図 162. キャプチャ/比較チャンネル (チャンネル 1、同じくチャンネル 2 および 3) の出力ステージ



1. OCxREF、ここで x は相補チャンネルのランク

図 163. キャプチャ/比較チャンネル (チャンネル 4) の出力ステージ

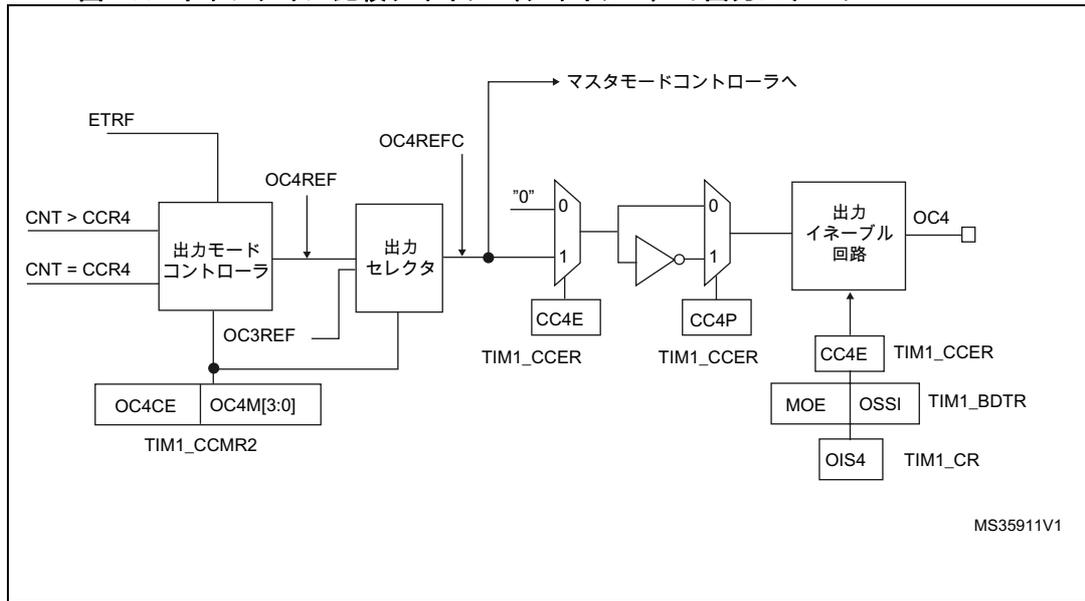
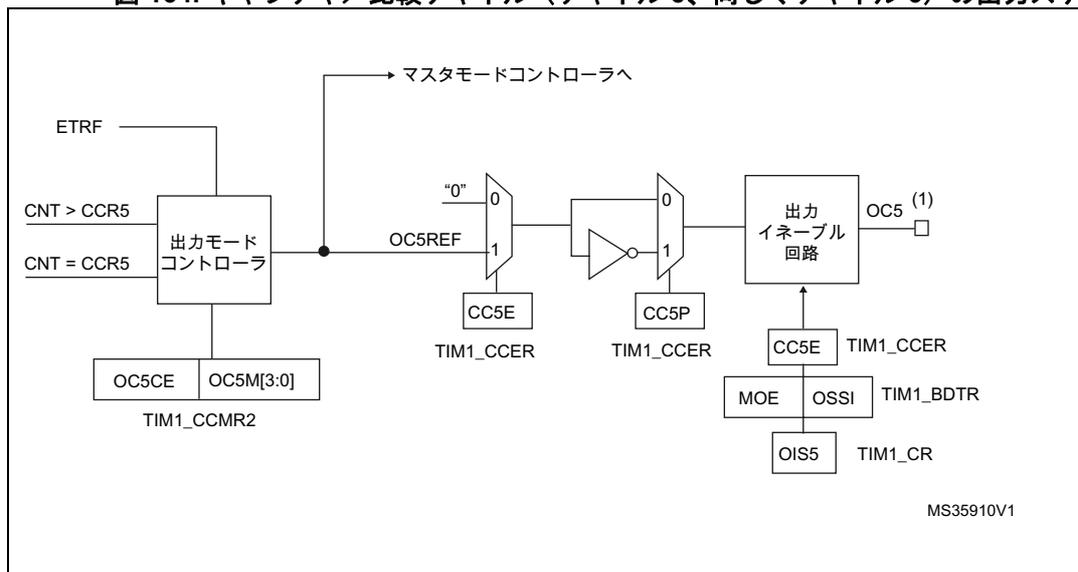


図 164. キャプチャ/比較チャンネル (チャンネル 5、同じくチャンネル 6) の出力ステージ



1. 外部的には使用できません。

キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

## 22.3.7 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx\_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx\_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx\_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx\_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx\_CCR1 は TI1 入力とリンクされていなければならないため、このためには TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx\_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数  $f_{DTS}$  でサンプリング)。この場合、TIMx\_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ変化のエッジを選択します。このためには、TIMx\_CCER レジスタの CC1P ビットと CC1NP ビットに“0”を書き込みます (この場合、立ち上がりエッジの選択)。
- 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいため、プリスケアラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
- TIMx\_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要な場合は、TIMx\_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx\_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

**注：** IC 割り込みと DMA リクエストは、TIMx\_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

## 22.3.8 PWM 入力モード

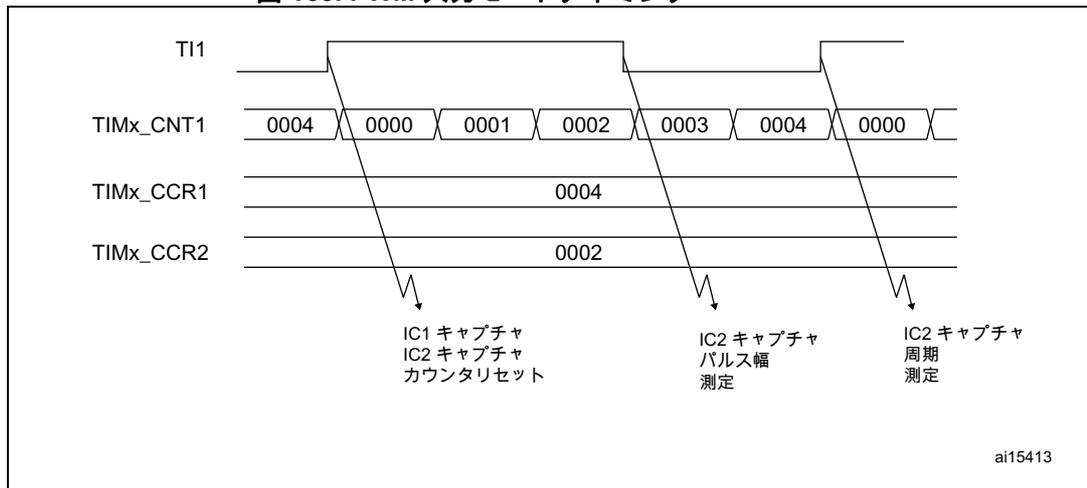
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます (手順は、CK\_INT 周波数とプリスケール値によって、若干異なることがあります)。

- TIMx\_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx\_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx\_CCR1 のキャプチャとカウンタクリアの両方に使用)。
- TIMx\_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
- CC2P ビットと CC2NP ビットに CC2P/CC2NP=“10”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx\_CCR2 のキャプチャに使用されます)。
- TIMx\_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx\_SMCR レジスタの SMS ビットに 0100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx\_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 165. PWM 入力モードタイミング



## 22.3.9 強制出力モード

出力モード (TIMx\_CCMRx レジスタの CCxS ビット =00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルとするには、対応する TIMx\_OCMRx レジスタの OCxM ビットに“0101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例 : CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx\_CCMRx レジスタの OCxM ビットに“0100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx\_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

## 22.3.10 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。マイクロコントロール内でチャンネル 5 および 6 のみが使用可能である場合に、チャンネル 1 から 4 を出力できます (たとえば、合成波形生成または ADC トリガのため)。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx\_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=0000)、アクティブにセットされるか (OCxM=0001)、非アクティブにセットされるか (OCxM=0010)、または反転されます (OCxM=0011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx\_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx\_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx\_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx\_CR2 レジスタの CCDS ビットが使用されます)。

TIMx\_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx\_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

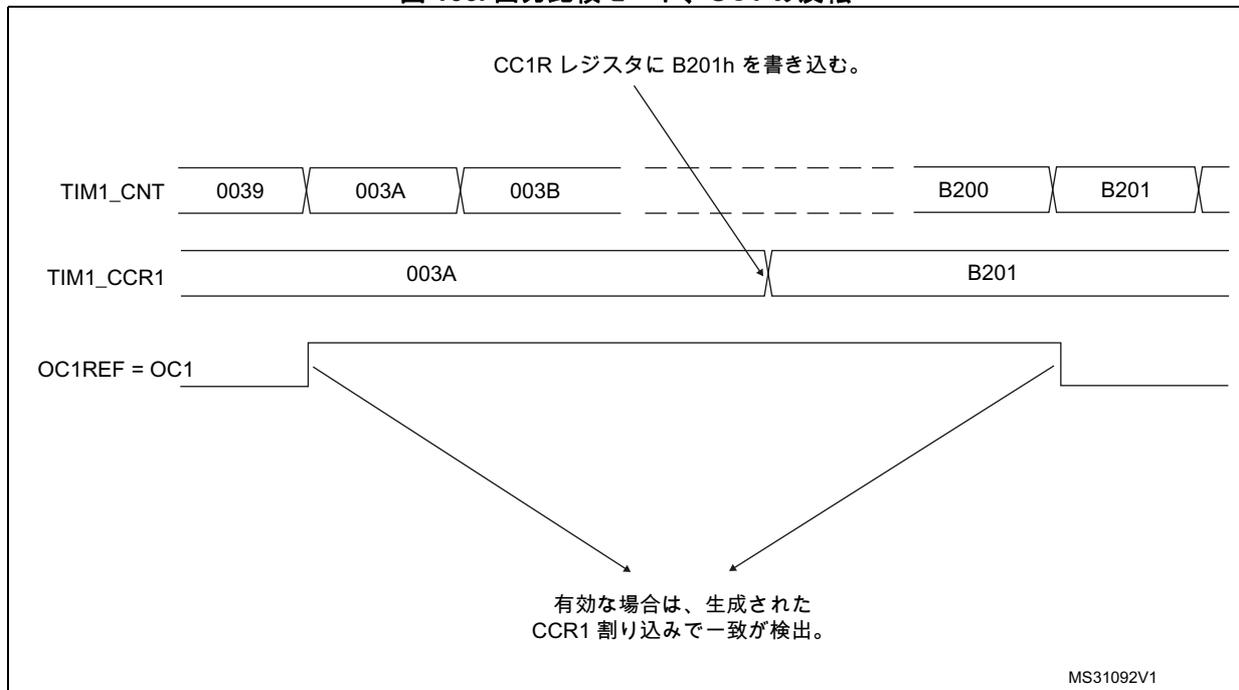
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは、単一パルスを出力するためにも使用できます (ワンパルスモード)。

## 手順

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx\_ARR レジスタと TIMx\_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
  - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに 0011 を書き込みます。
  - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
  - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
  - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
5. TIMx\_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx\_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx\_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 166](#) に示します。

図 166. 出力比較モード、OC1 の反転



## 22.3.11 PWM モード

パルス幅変調(PWM)モードでは、TIMx\_ARR レジスタの値によって決められた周波数と TIMx\_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“0110” (PWM モード 1) または“0111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx\_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx\_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx\_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx\_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx\_CCER および TIMx\_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx\_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx\_CNT と TIMx\_CCRx が常に比較されて、TIMx\_CCRx ≤ TIMx\_CNT または TIMx\_CNT ≤ TIMx\_CCRx かどうか判断されます (カウントの方向によります)。

タイマは、TIMx\_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

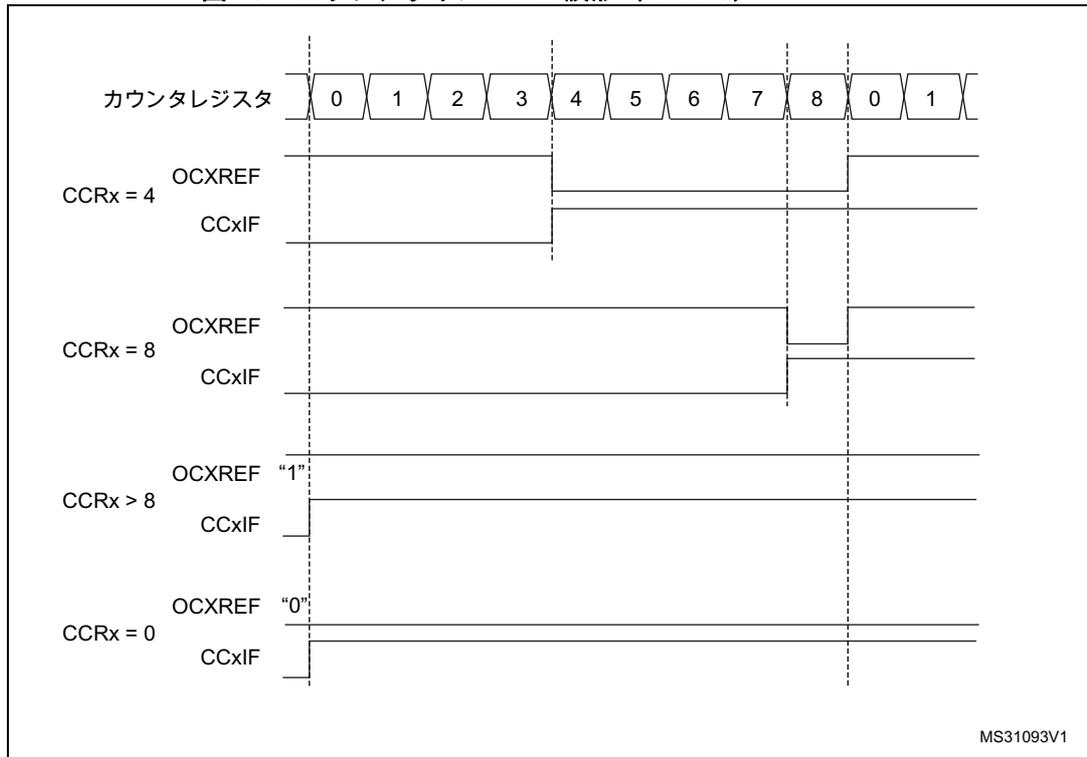
### PWM エッジアラインモード

- アップカウント構成

TIMx\_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード \(613 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx\_CNT < TIMx\_CCRx の間はハイに、そうでない場合はローになります。TIMx\_CCRx の比較値が自動再ロード値 (TIMx\_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 167](#) に TIMx\_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 167. エッジライン PWM 波形 (ARR=8)



- **ダウンカウント構成**  
TIMx\_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード \(617 ページ\)](#) を参照してください。  
PWM モード 1 では、基準信号 OCxRef は、TIMx\_CNT > TIMx\_CCRx の間はローであり、そうでない場合はハイになります。TIMx\_CCRx の比較値が TIMx\_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0% の PWM 信号を生成することはできません。

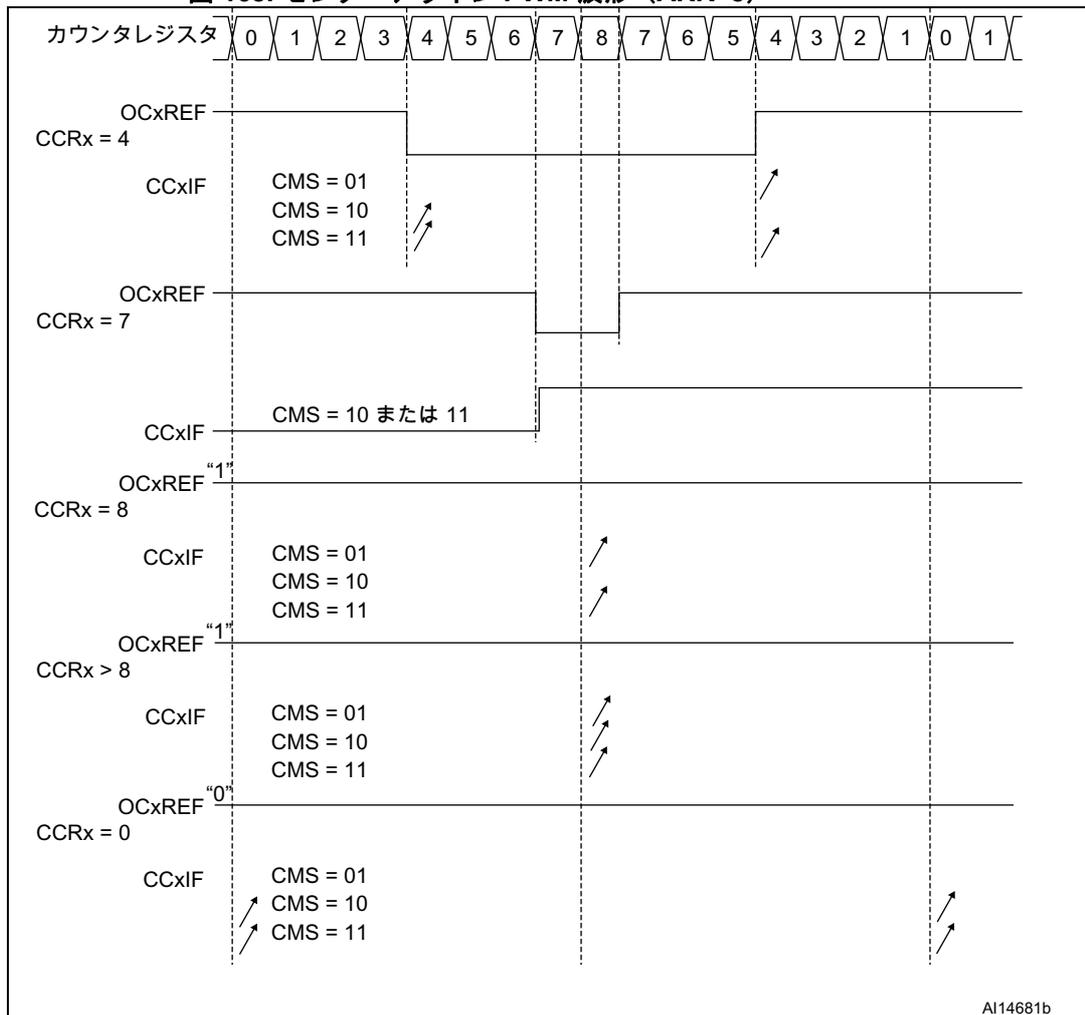
## PWM センターアラインモード

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”でないときにアクティブです (その他すべての構成は、OCxRef/OCx 信号に対して同じ効果を持ちます)。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx\_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターアラインモード \(アップ/ダウンカウント\) \(620 ページ\)](#) を参照してください。

図 168 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx\_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx\_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 168. センターアライン PWM 波形 (ARR=8)



### センターアラインモードの使用に関するヒント

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx\_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
  - 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx\_CNT > TIMx\_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
  - カウンタに 0 または TIMx\_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx\_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

## 22.3.12 非対称 PWM モード

非対称モードでは、プログラム可能な位相シフトによって 2 つのセンターアライン PWM 信号の生成を可能にします。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや位相シフトは TIMx\_CCRx レジスタペアで決定されます。1 つ目のレジスタがアップカウント時の PWM を制御し、2 つ目のレジスタがダウンカウント時の PWM を制御することで、PWM は PWM ハーフサイクルごとに調整されます。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されません。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されません。

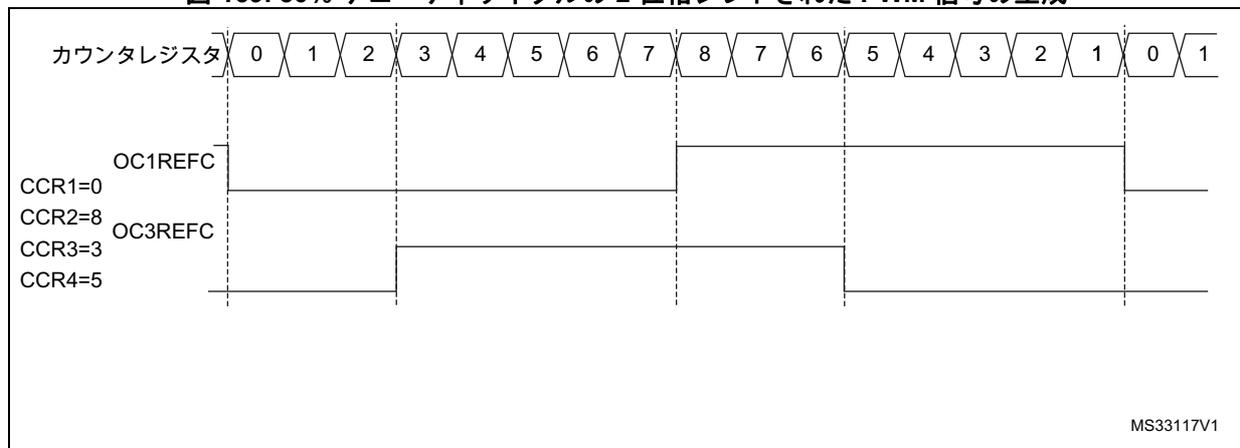
非対称 PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1110” (非対称 PWM モード 1) または“1111” (非対称 PWM モード 2) を書き込むことによって、2 チャネルごとに選択できます (CCR レジスタペアごとに 1 つの OCx 出力)。

**注：** OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

特定のチャンネルが非対称の PWM チャンネルとして使用されると、その相補チャンネルも使用できます。たとえば、OC1REFC 信号がチャンネル 1 (非対称 PWM モード 1) に生成されると、チャンネル 2 の OC2REF 信号、または非対称 PWM モード 1 の結果として得られる OC2REFC 信号を出力できます。

図 169 は、非対称 PWM モードを使用して生成される信号の例を表します (チャンネル 1 から 4 は非対称 PWM モード 1 として設定されます)。これにより、デッドタイムジェネレータとともにフルブリッジ位相シフト DC-DC コンバータを制御できます。

図 169. 50% デューティサイクルの 2 位相シフトされた PWM 信号の生成



MS33117V1

## 22.3.13 組み合わせ PWM モード

組み合わせ PWM モードでは、2つのエッジラインまたはセンターライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は2つの TIMx\_CCRx レジスタで決定されます。結果として得られる信号 OCxREFC は、2つの PWM 基準信号の OR または AND による論理結合から成ります。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

組み合わせ PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1100” (組み合わせ PWM モード 1) または“1101” (組み合わせ PWM モード 2) を書き込むことによって、2チャンネルごとに選択できます (CCR レジスタペアごとに1つの OCx 出力)。

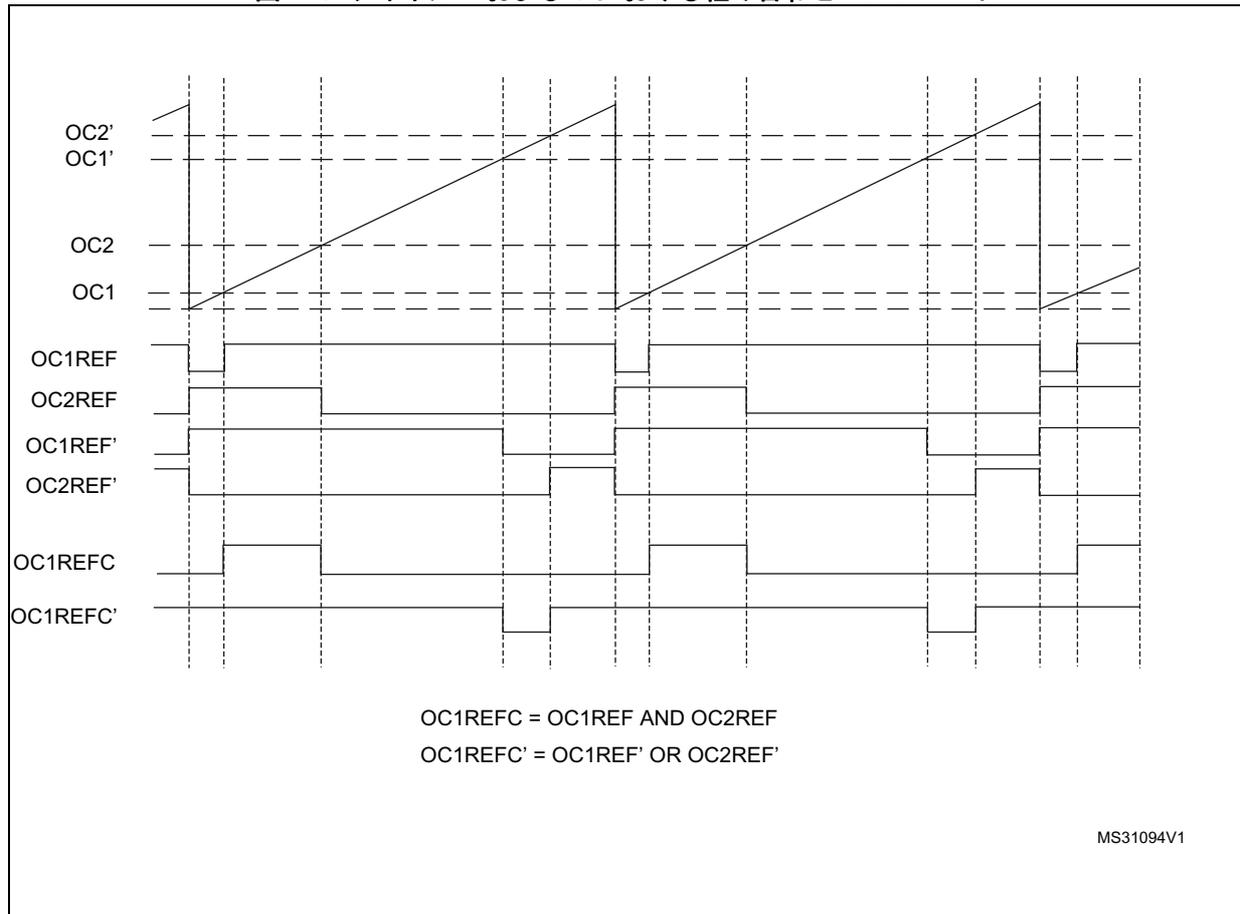
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、相補チャンネルを反対の PWM モードに設定する必要があります (たとえば、1つを組み合わせ PWM モード 1、もう1つを組み合わせ PWM モード 2 にします)。

**注:** OCxM[3:0] ビットフィールドは互換性を確保するために2つのパーツに分割され、最上位ビットと3つの最下位ビットとは隣接していません。

 170 は、次の設定で取得可能な非対称 PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合
- チャンネル 3 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 4 が PWM モード 1 で設定されている場合

図 170. チャンネル 1 および 3 における組み合わせ PWM モード



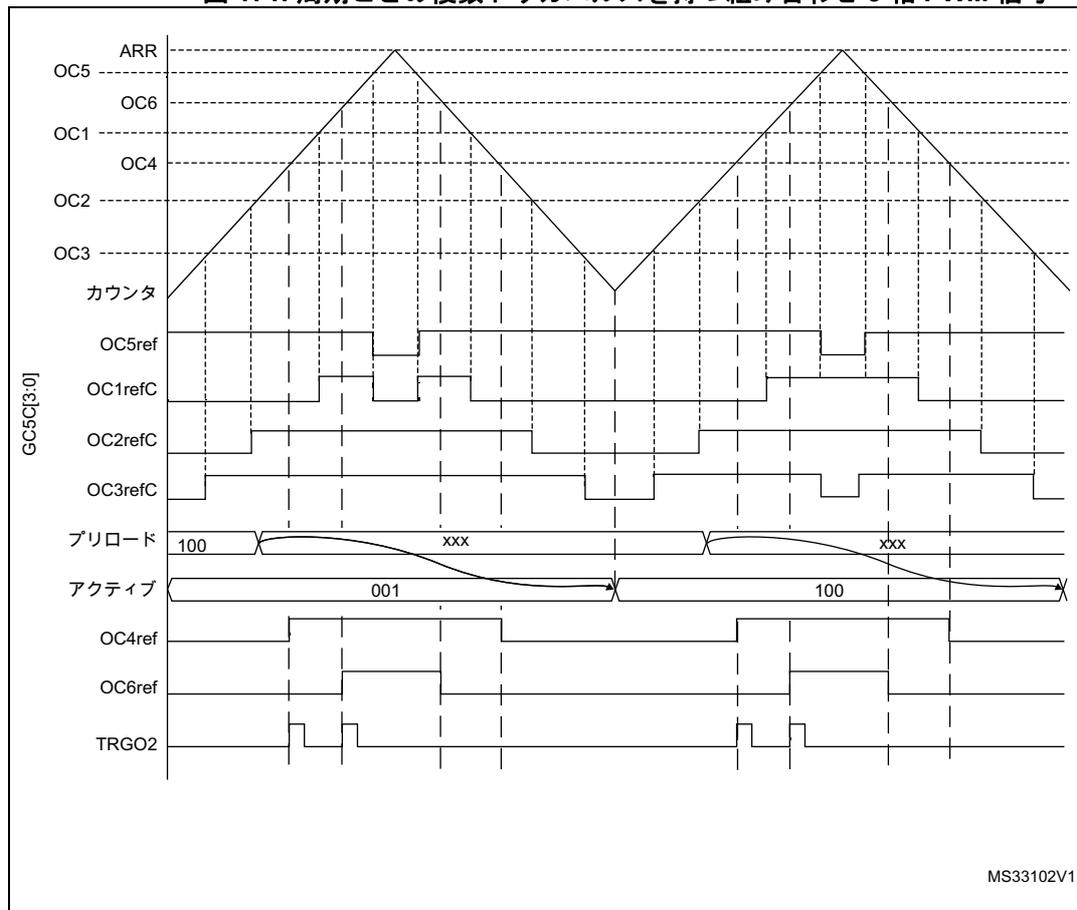
## 22.3.14 組み合わせ 3 相 PWM モード

組み合わせ 3 相 PWM モードでは、パルスの中で論理積を取った単一のプログラム可能な信号とともに 1 つから 3 つのセンターアライン PWM 信号を生成できます。結果として得られる組み合わせ信号の定義には、OC5REF 信号が使用されます。TIMx\_CCR5 の 3 ビット GC5C[3:1] では、OC5REF を組み合わせる基準信号を選択できます。結果として得られる信号 OCxREFC は、2 つの PWM 基準信号の AND による論理結合から生成されます。

- GC5C1 がセットされると、OC1REFC は TIMx\_CCR1 および TIMx\_CCR5 によって制御されます。
- GC5C2 がセットされると、OC2REFC は TIMx\_CCR2 および TIMx\_CCR5 によって制御されます。
- GC5C3 がセットされると、OC3REFC は TIMx\_CCR3 および TIMx\_CCR5 によって制御されます。

組み合わせ 3 相 PWM モードは、少なくとも 3 ビット GC5C[3:1] の 1 つをセットすることでチャンネル 1 から 3 で個別に選択できます。

図 171. 周期ごとの複数トリガパルスを持つ組み合わせ 3 相 PWM 信号



TRGO2 波形は、特定の 3 相 PWM 信号での ADC の同期方法を示します。詳細については、[セクション 22.3.26 : ADC の同期](#)を参照してください。

## 22.3.15 相補出力とデッドタイム挿入

高機能制御タイマ (TIM1) は、2 つの相補信号を出力して、出力時のスイッチオンおよびスイッチオフを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性 (レベルシフタの内在的な遅延、電源スイッチによる遅延など) に応じて調整する必要があります。

出力の極性 (主出力 OCx または補 OCxN) は、出力ごとに独自に選択できます。これは TIMx\_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 OCx および OCxN は、TIMx\_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx\_BDTR レジスタと TIMxCR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、[表 126: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能 \(689 ページ\)](#)を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるときに)、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッドタイムジェネレータがあります。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点が異なります。
- OCxN 出力信号は、立ち上がりエッジが基準波形の立ち下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (OCx または OCxN) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 OCxREF との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 172. デッドタイム挿入のある相補出力

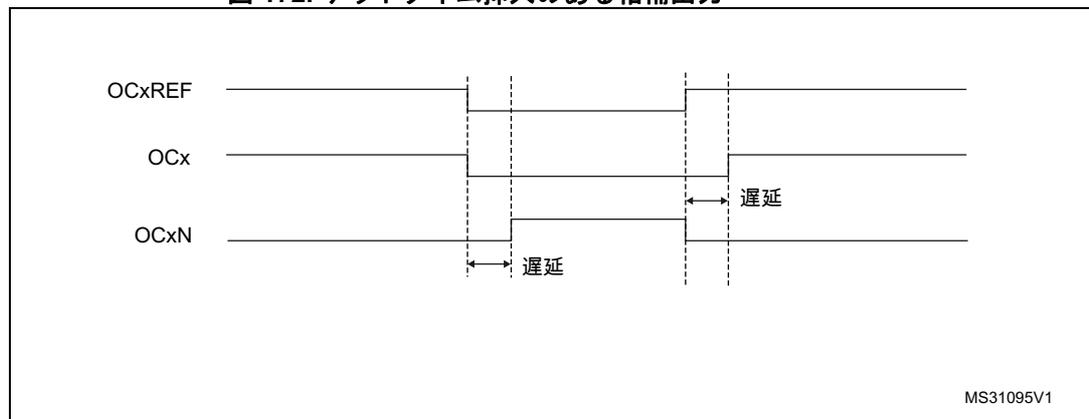


図 173. 負のパルスより長い遅延があるときのデッドタイムの波形

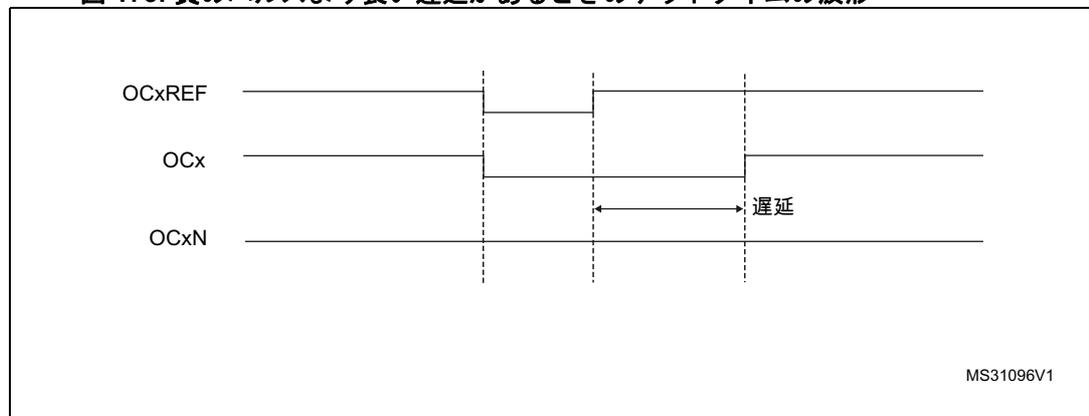
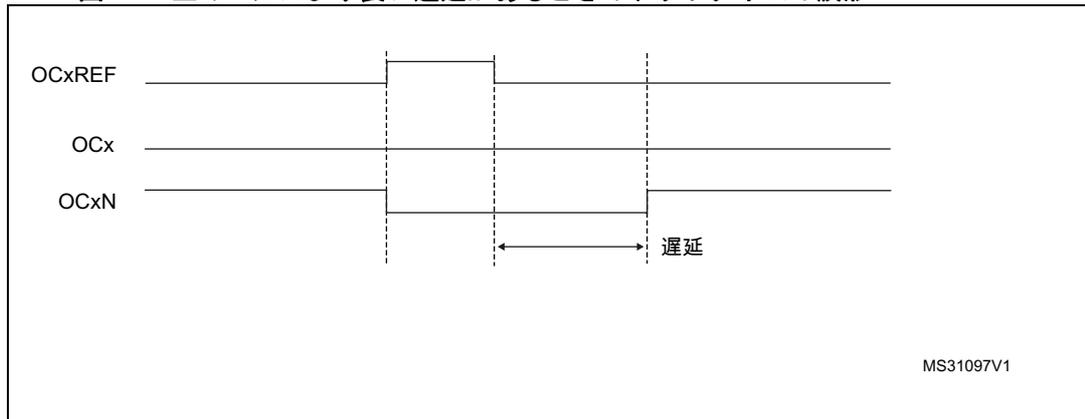


図 174. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャンネルで同じであり、TIMx\_BDTR レジスタの DTG ビットでプログラミングできます。遅延計算については、[セクション 22.4.18 : TIM1/TIM8 ブレークおよびデッドタイムレジスタ \(TIMx\\_BDTR\) \(693 ページ\)](#) を参照してください。

### OCxREF 信号の OCx または OCxN へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx\_CCER レジスタの CCxE ビットおよび CCxNE ビットを構成することによって、OCxREF 信号を OCx 出力または OCxN 出力にリダイレクトできます。

これにより、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定することができます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができま

**注：** OCxN のみが有効なときには (CCxE=0, CCxNE=1)、相補にならず、OCxREF がハイレベルとなるとアクティブになります。たとえば、CCxNP=0 の場合は、OCxN=OCxRef です。他方、OCx と OCxN の両方が有効なときには (CCxE=CCxNE=1)、OCxREF がハイになると OCx はアクティブになり、OCxREF がローのときには、OCxN は補信号であり、アクティブになります。

### 22.3.16 ブレーク機能の使用

ブレーク機能の目的は、TIM1 および TIM8 タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。2つのブレーク入力通常、パワーステージおよび3相インバータの異常出力に接続されています。アクティブ化すると、ブレーク回路はPWM出力を遮断し、強制的に事前定義された安全な状態に移行させます。出力の遮断をトリガするために、いくつかの内部MCUイベントを選択することも可能です。

このブレークには、2つのチャンネルがあります。システムレベル障害（クロック障害、パリティエラーなど）とアプリケーション障害（入力ピンおよび内蔵比較回路）の両方を集め、出力をデッドタイムの持続時間経過後に事前定義されたレベル（アクティブまたはインアクティブ）に強制できるブレークチャンネル。アプリケーション障害のみを含み、出力をインアクティブ状態に強制できるブレーク2チャンネル。

ブレーク時の出力有効信号および出力レベルは、いくつかの制御ビットに依存しています。

- TIMx\_BDTR レジスタの MOE ビット。ソフトウェアによって出力を有効/無効にすることができます。また、ブレーク または ブレーク2 イベント時にリセットされます。
- TIMx\_BDTR レジスタの OSSI ビット。出力をインアクティブ状態で制御するか、GPIO コントローラへの制御を解除するかについて、タイマを定義します (通常、ハイインピーダンスモードにするため)。
- TIMx\_CR2 レジスタの OISx および OISxN ビット。アクティブまたはインアクティブな出力遮断レベルをセットします。OISx および OISxN の値にかかわらず、一度に OCx 出力と OCxN 出力を両方ともアクティブレベルにセットすることはできません。詳細については、[表 126: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能 \(689 ページ\)](#) を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。TIMx\_BDTR レジスタの BKE および BKE2 ビットをセットすることによって、ブレーク機能を有効にできます。ブレーク入力の極性は、同じレジスタの BKP および BKP2 ビットを設定することによって選択できます。BKEx と BKPx は、同時に変更できます。BKEx および BKPx ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がリエッジは非同期のことがあるので、実際の信号 (出力に作用する信号) と同期制御ビット (TIMx\_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延 (ダミー命令) を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

BIRDIE コンディショナルタグが有効な場合、このセクションを以下のように置き換える必要があります。

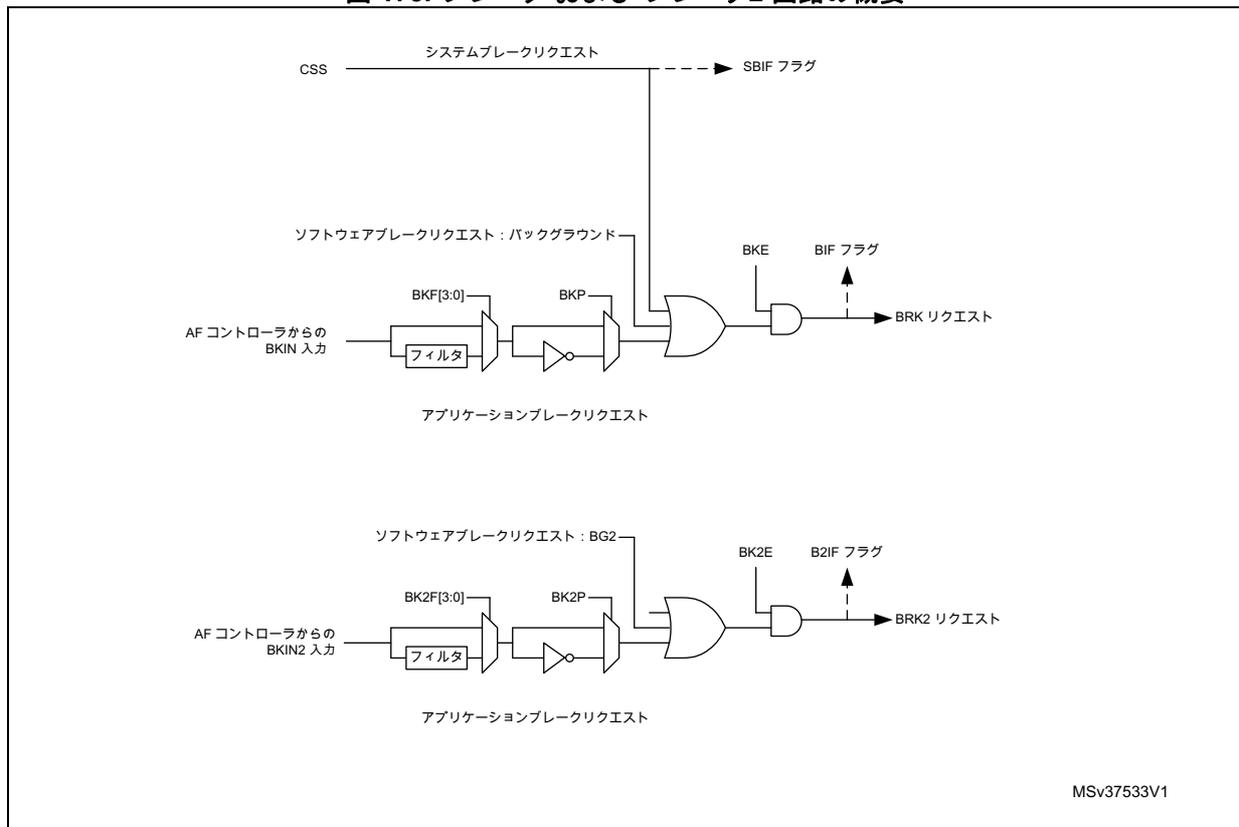
ブレーク (BRK) イベントは、論理和が取られた 2 つのソースから生成できます。

- AFIO コントローラでの選択に従って BKIN ピンの 1 つに接続された外部ソース
- 内部ソース: CSS 検出回路によって生成されたクロック障害イベント

ブレーク2 (BRK2) は、AFIO コントローラでの選択に従って BKIN2 ピンの 1 つに接続された外部ソースから生成できます。

ブレークイベントは、TIMx\_EGR レジスタで BG および B2G ビットを使用して、ソフトウェアによって生成することもできます。

図 175. ブレーク および ブレーク2 回路の概要



**注：** 非同期（クロックなし）動作は、プログラム可能なフィルタが無効な場合にのみ保証されます。有効になっている場合は、必ずブレークイベントが処理されるように、フェイルセーフクロックモード（たとえば、内部 PLL や CSS を使用）を使用する必要があります。

ブレークが 1 つ発生すると（ブレーク入力の 1 つで選択されたレベル）、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態またはアイドル状態になるか、GPIO コントローラへの制御が解除されます（OSSI ビットで選択）。これは、MCU オシレータがオフの場合に有効です。
- 各出力チャネルは、MOE=0 になったとき、TIMx\_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマは出力の制御（GPIO コントローラによって引き継がれた）を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには：
  - － 出力は、まずインアクティブ状態に置かれます（極性に依存します）。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
  - － タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください（約 2 CK\_TIM クロックサイクル）。
  - － OSSI=0 の場合、タイマは出力の制御（ハイインピーダンス状態を強制する GPIO コントローラによって引き継がれた）を解除し、そうでない場合、イネーブル出力はハイのままか、CCxE または CCxNE ビットのどちらかがハイになったときにハイになります。

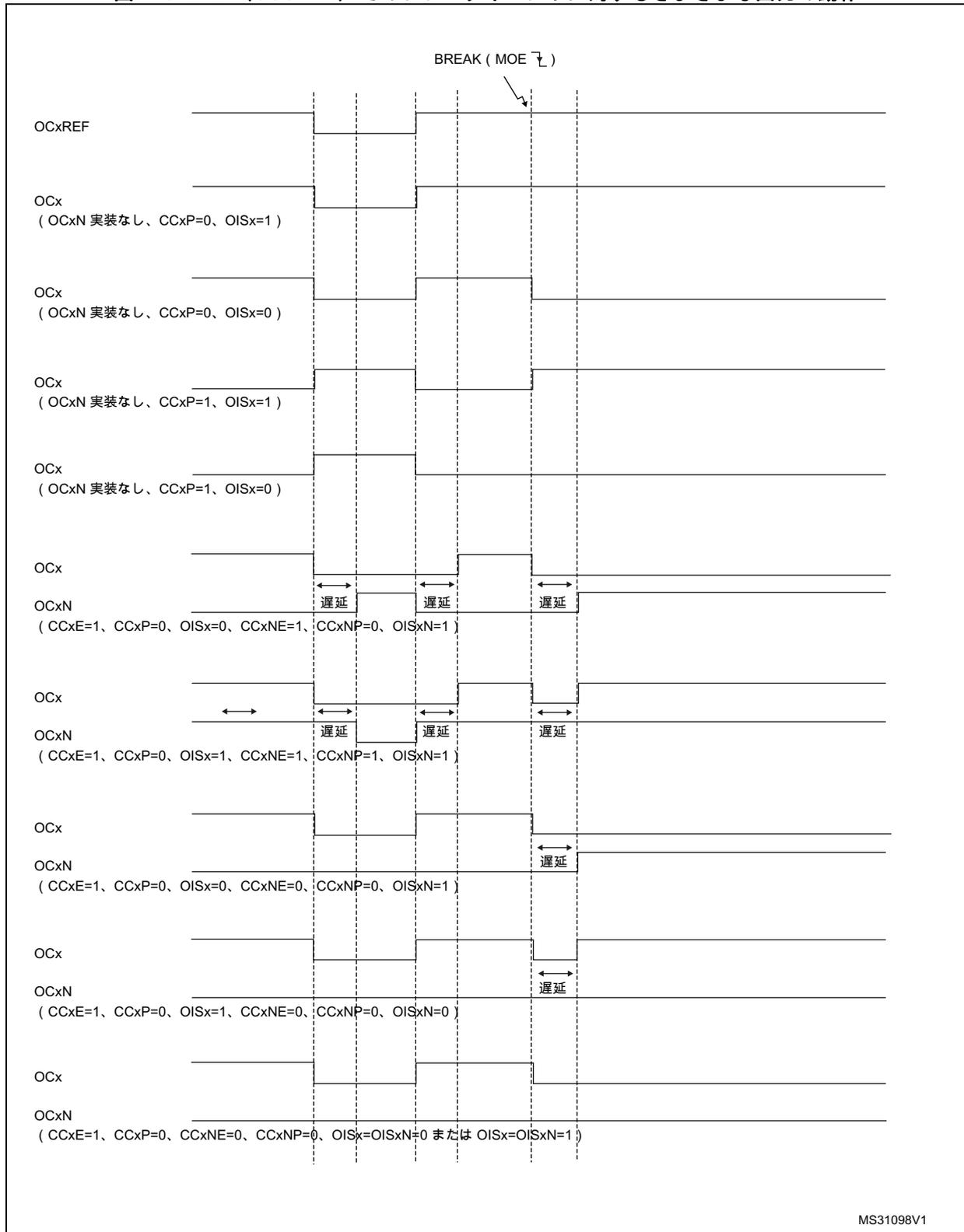
- ブレーク状態フラグ (TIMx\_SR レジスタの SBIF、BIF、および B2IF ビット) がセットされず、TIMx\_DIER レジスタの BIE ビットがセットされている場合、割り込みが生成されます。TIMx\_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx\_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント (UEV) で再び自動的にセットされます。たとえば、これを使用してレギュレーションを行うことができます。そうでない場合、MOE ビットはアプリケーションが再び“1”をセットするまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

**注：** **ブレーク入力は、信号レベルに対してアクティブです。このため、ブレーク入力がアクティブな間は、MOE をセットできません (自動的にも、ソフトウェアによっても)。この間、ステータスフラグ BIF および B2IF をクリアできません。**

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ (デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性) を固定することができます。TIMx\_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 22.4.18 : TIM1/TIM8 ブレークおよびデッドタイムレジスタ \(TIMx\\_BDTR\) \(693 ページ\)](#) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。

[図 176](#) に、ブレークに対する出力の動作例を示します。

図 176. BRK (OSSI = 1) でのブレーキイベントに対するさまざまな出力の動作



2つのブレーク入力は、タイマ出力で異なる動作を示します。

- BRK 入力は、無効化（インアクティブ状態）するか、PWM 出力を強制的に事前定義した安全な状態に移行できます。
- BRK2 は、PWM 出力の無効化（インアクティブ状態）のみ可能です。

表 123 に示すように、BRK の優先順位は BRK2 入力よりも高いです。

注： **BRK2 は OSSR = OSSI = 1 の場合にのみ使用してください。**

表 123. タイマ出力と BRK/BRK2 入力の動作

BRK	BRK2	タイマ出力状態	通常の使用例	
			OCxN 出力 (ローサイドスイッチ)	OCx 出力 (ハイサイドスイッチ)
アクティブ	X	- インアクティブから強制される出力状態 (デッドタイム後) - OSSI = 0 の場合、出力は無効 (GPIO ロジックが制御を引き継ぐ)	デッドタイム挿入後 ON	オフ
インアクティブ	アクティブ	インアクティブ	オフ	オフ

図 177 では、BRK および BRK2 入力で信号がアクティブな場合の、OCx および OCxN 出力の動作の例を示します。この場合、両方の出力がアクティブハイ極性になります (TIMx\_CCER レジスタの CCxP = CCxNP = 0)。

図 177. BRK および BRK2 ピンのアサート後の PWM 出力状態 (OSSI=1)

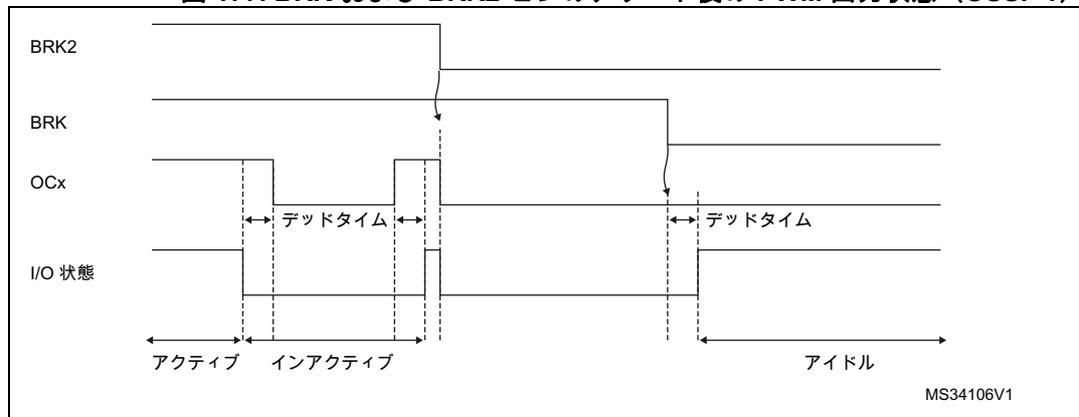
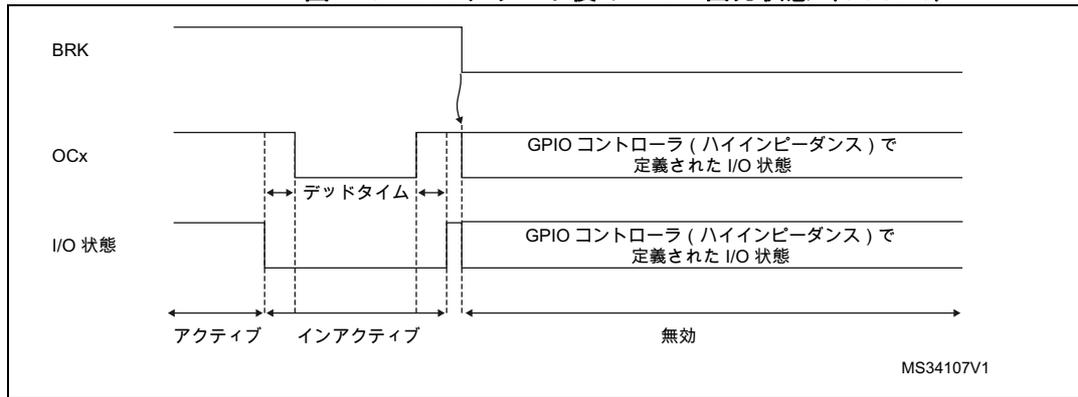


図 178. BRK アサート後の PWM 出力状態 (OSS1=0)



## 22.3.17 外部イベントによる OCxREF 信号のクリア

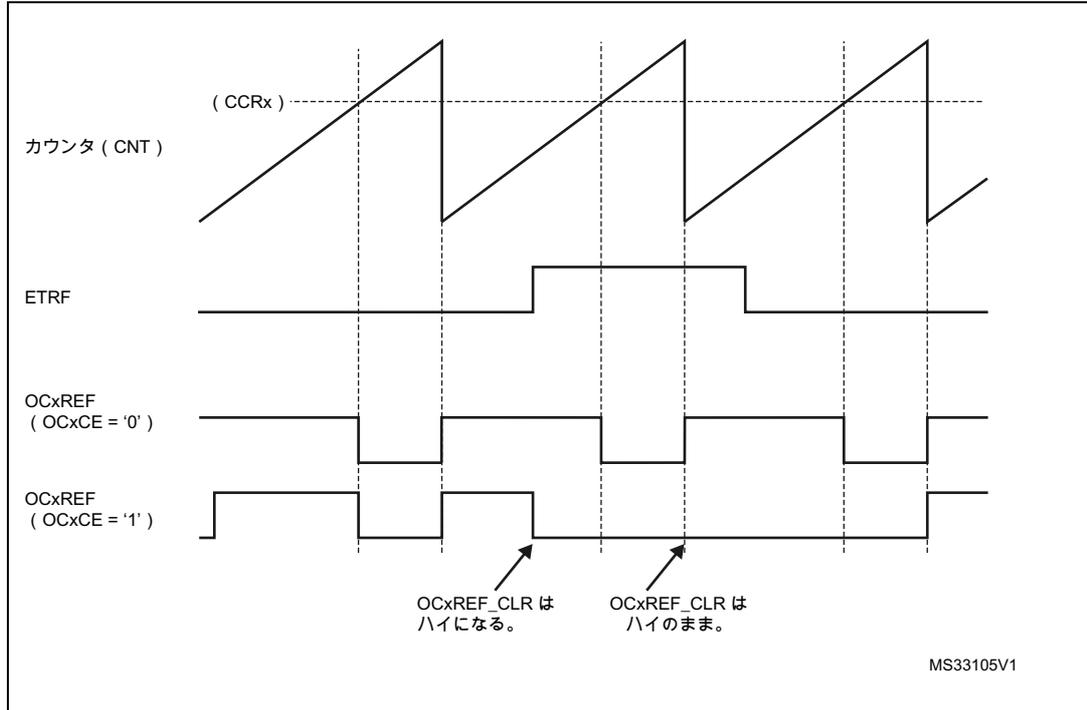
特定のチャネルの OCxREF 信号は、ETRF 入力にハイレベルを適用する (対応する TIMx\_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットすることによってローに駆動できません。OCxREF 信号は、次の更新イベント UEV が発生するまで、ローレベルを保ちます。

ETRF が選択された場合、ETR は次のように設定する必要があります。

1. 外部トリガプリスケアラはオフ状態に維持します (TIMx\_SMCR レジスタの ETPS[1:0] ビットを“00”にセット)。
2. 外部クロックモード 2 を無効にします (TIMx\_SMCR レジスタの ECE ビットを“0”にセット)。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、ユーザのニーズに応じて設定できます。

図 179 に、イネーブルビット OCxCE の両方の値について、ETRF 入力がハイレベルになったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 179. TIMx OCxREF のクリア



注： 100% デューティサイクルのPWMの場合 (CCR<sub>x</sub> > ARRの場合)、次のカウンタオーバーフローでOCxREFが再度有効になります。

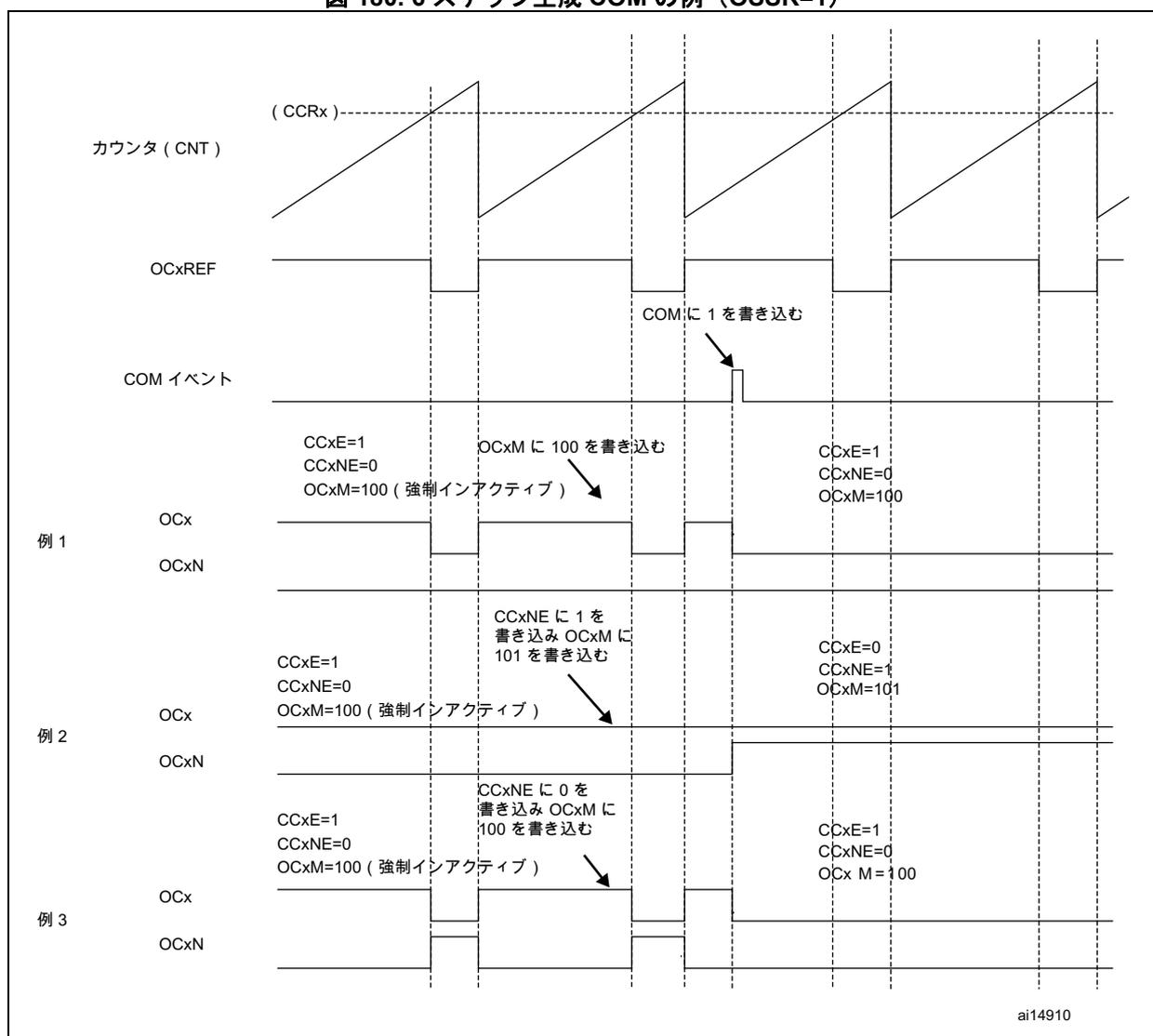
## 22.3.18 6 ステップ PWM 生成

チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転換イベントでシャドウビットにコピーされます。これにより、次のステップの構成をあらかじめプログラミングして、すべてのチャンネルの構成を同時に変更することができます。COM は、TIMx\_EGR レジスタの COM ビットをセットすることによってソフトウェアによって、またはハードウェアによって (TRGI 立ち上がりエッジで) 生成することができます。

フラグは、COM イベントが発生したときにセットされ (TIMx\_SR レジスタの COMIF ビット)、これによって割り込み (TIMx\_DIER レジスタの COMIE ビットがセットされている場合) または DMA リクエスト (TIMx\_DIER レジスタの COMDE ビットがセットされている場合) を生成できます。

図 180 に、COM イベントが発生したときの OCx と OCxN 出力の動作を、3 種類のプログラミング構成の例で示します。

図 180. 6 ステップ生成 COM の例 (OSSR=1)



## 22.3.19 ワンパルスモード

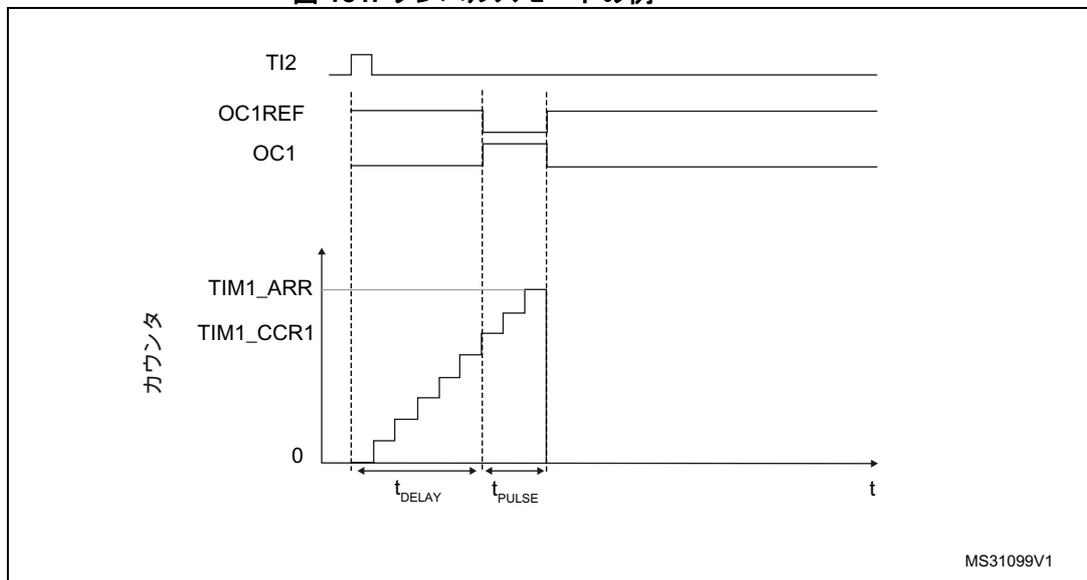
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 :  $CNT < CCRx \leq ARR$  (特に  $0 < CCRx$  の場合)
- ダウンカウント時 :  $CNT > CCRx$

図 181. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が  $t_{PULSE}$  の正のパルスを遅延時間  $t_{DELAY}$  後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx\_CCER レジスタで CC2P=“0”と CC2NP =“0”を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx\_SMCR レジスタの TS ビットに“110”を書き込みます。
- TI2FP2 を使用してカウンタを開始します。このためには、TIMx\_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- $t_{\text{DELAY}}$  は、TIMx\_CCR1 レジスタに書き込まれた値によって定義されます。
- $t_{\text{PULSE}}$  は、自動再ロード値と比較値の差 (TIMx\_ARR - TIMx\_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx\_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx\_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx\_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx\_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので (シングルモード)、TIMx\_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx\_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース : OCx 高速イネーブル :

ワンパルスモードでは、TIx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ( $t_{\text{DELAY min}}$ ) が制限されます。

最小遅延で波形を出力したい場合は、TIMx\_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

### 22.3.20 再トリガ可能なワンパルスモード (OPM)

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 22.3.19](#) で説明する再トリガ不可能なワンパルスモードについて、次のような違いがあります。

- パルスはトリガが発生し次第開始します (プログラム可能な遅延はありません)。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

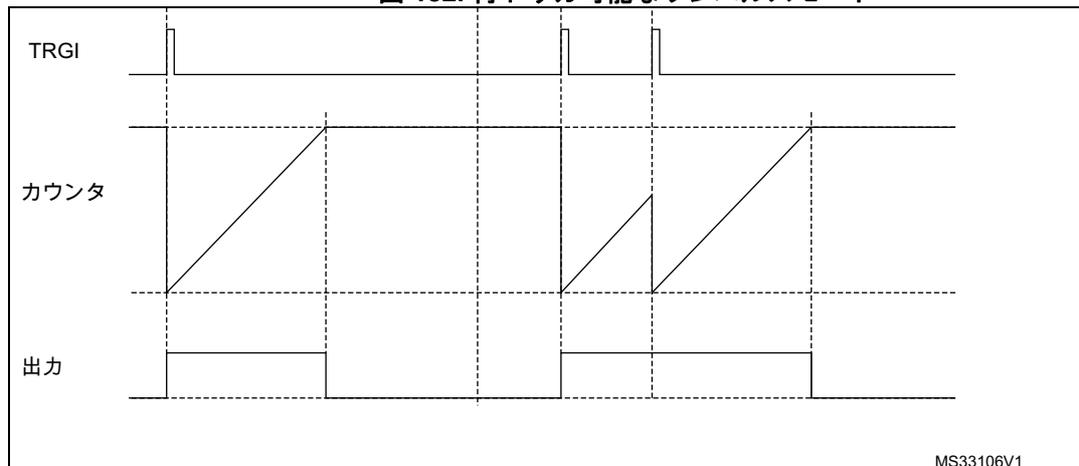
タイマはスレーブモードである必要があり、このときビットは TIMx\_SMCR レジスタで SMS[3:0] = “1000” (リセットモードとトリガモードの組み合わせ)、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が “1000” または “1001” にセットされています。

タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります (ARR レジスタによってパルス長がセットされます)。タイマをダウンカウントモードで設定した場合、ARR を 0 にセットする必要があります (CCRx レジスタによってパルス長がセットされます)。

**注 :** OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

再トリガ可能なワンパルスモードでは、CCxIF は意味を持ちません。

図 182. 再トリガ可能なワンパルスモード



## 22.3.21 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx\_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS="001"を、TI1 エッジのみをカウントしている場合は SMS="010"を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS="011"を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx\_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2つの入力 TI1 と TI2 は、直交エンコーダとのインタフェースに使用されます。表 124を参照してください。カウンタのクロック供給は、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択の後は TI1 と TI2、フィルタされず反転されない場合は TI1FP1=TI1、フィルタされず反転されない場合は TI2FP2=TI2) での有効な遷移ごとに行われます。ただし、カウンタは有効である (TIMx\_CR1 レジスタの CEN ビットに"1"が書き込まれている) ことが前提です。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx\_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIRビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx\_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx\_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、繰り返しカウンタ、トリガ出力の機能は通常どおりに機能を続けます。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタは直交エンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (TI1 と TI2 は同時に切り替わらないと想定しています)。

表 124. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	ハイ	ダウン	アップ	カウントなし	カウントなし
	ロー	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	ハイ	カウントなし	カウントなし	アップ	ダウン
	ロー	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の両方を カウント	ハイ	ダウン	アップ	アップ	ダウン
	ロー	アップ	ダウン	ダウン	アップ

直交エンコーダは、外部インターフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 183 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx\_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S="01" (TIMx\_CCMR2 レジスタ、TI1FP2 は TI2 に配置)
- CC1P="0"、CC1NP="0" (TIMx\_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1)
- CC2P="0"、CC2NP="0" (TIMx\_CCER レジスタ、TI1FP2 非反転、TI1FP2=TI2)
- SMS="011" (TIMx\_SMCR レジスタ、両方の入力が立ち上がり立ち下がりの両エッジでアクティブ)
- CEN="1" (TIMx\_CR1 レジスタ、カウンタ有効)

図 183. エンコーダインターフェースモードにおけるカウンタの動作例

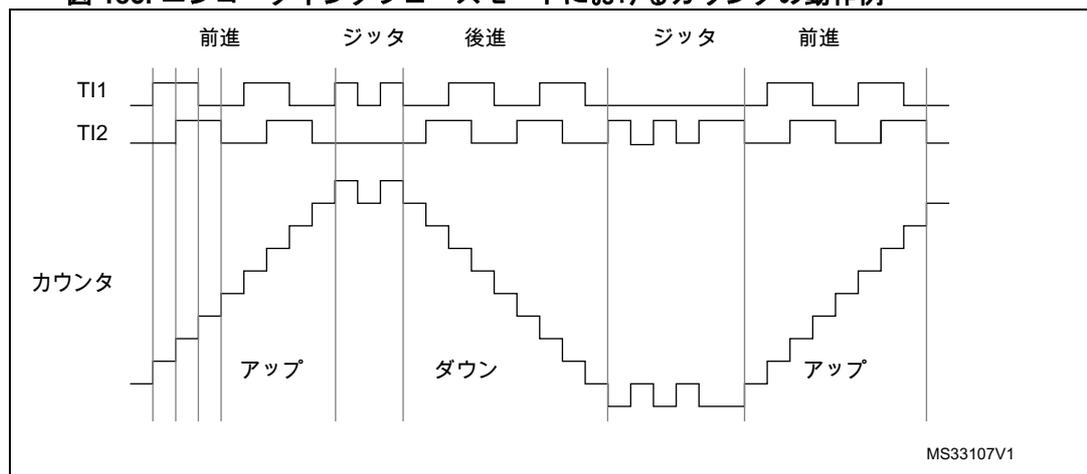
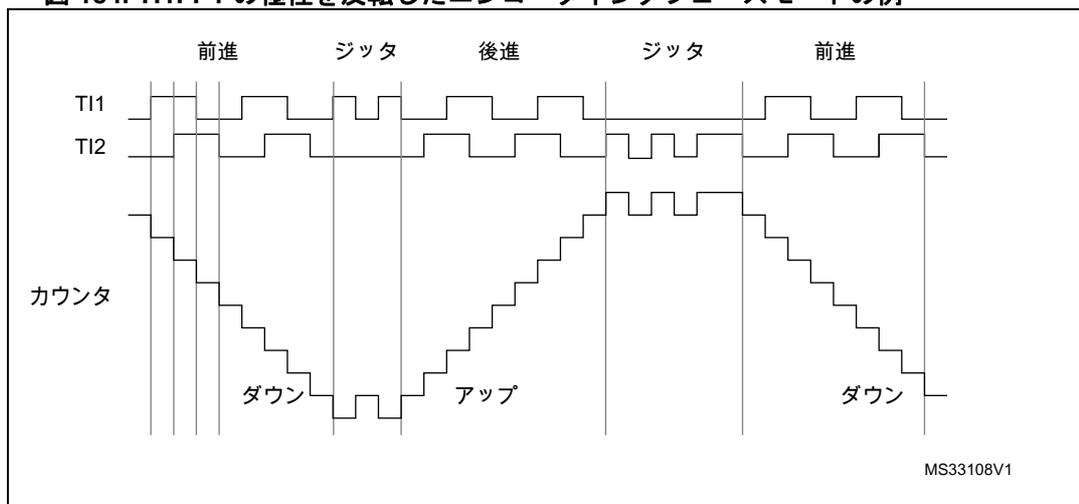


図 184 に、TI1FP1 の極性を反転したときのカウンタの動作を示します (上記と同じ設定ですが、CC1P="1")。

図 184. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします (キャプチャ信号は周期的でなければならず、別のタイマによって生成できます)。使用可能なときには、DMA リクエストを通じて値を読み出すことも可能です。

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー状態を分割できないものとして読み取ることができます。バックグラウンドタスク (カウンタの読み出し) と中断 (更新の中断) との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読み出しアクセス時に UIFCPY フラグによって上書きされます (カウンタの最上位ビットには書き込みモード時のみアクセス可能)。

## 22.3.22 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。特殊なケースでは、バックグラウンドタスク (カウンタの読み出し) と中断 (更新の中断) との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

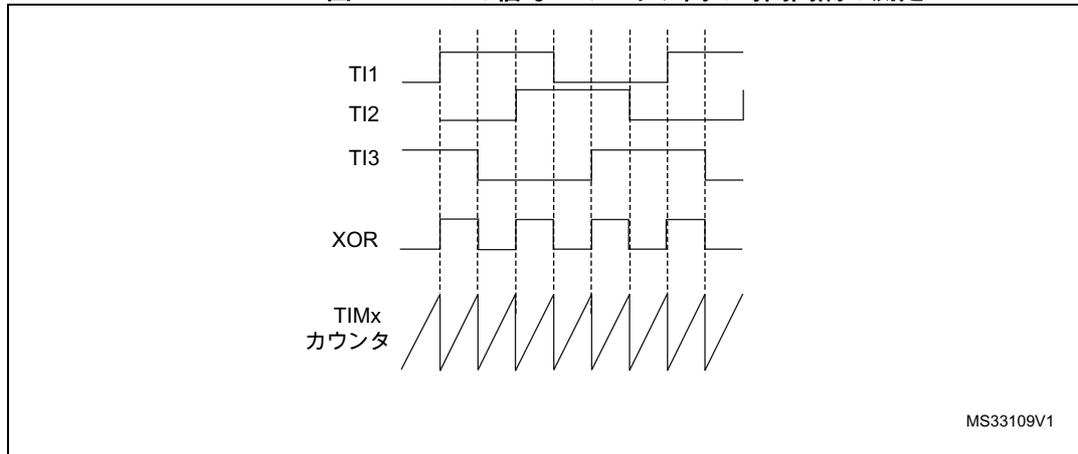
UIF と UIFCPY フラグのアサートの間には、遅延はありません。

## 22.3.23 タイマ入力 XOR 機能

TIMx\_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 2 つの入力ピン TIMx\_CH1 および TIMx\_CH2 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。次の [図 185](#) に示すように、2 つの入力信号上のエッジ間の間隔を測定するのに便利です。

図 185. 3 つの信号上のエッジ間の時間間隔の測定



MS33109V1

## 22.3.24 ホールセンサとのインタフェース

これは、高機能制御タイマ (TIM1 または TIM8) を使用して PWM 信号を生成し、モータと [図 186](#) で「インタフェースタイマ」と記されている別のタイマ TIMx (TIM2、TIM3、または TIM4) を駆動することによって実現します。「インタフェースタイマ」は、XOR を通じて TI1 入力チャンネル (TIMx\_CR2 レジスタの TI1S ビットをセットすることで選択できます) に接続された 3 つのタイマ入力ピン (CC1、CC2、CC3) をキャプチャします。

スレープモードコントローラはリセットモードに設定され、スレープ入力は TI1F\_ED です。したがって、3 つの入力のいずれかが反転するごとに、カウンタは 0 からカウントをリスタートします。これが、ホール入力の変化によってトリガされるタイムベースとなります。

「インタフェースタイマ」上で、キャプチャ/比較チャンネル 1 がキャプチャモードで設定され、キャプチャ信号は TRC です ([図 160: キャプチャ/比較チャンネル \(例: チャンネル 1 入カステージ\) \(631 ページ\)](#) を参照)。キャプチャされた値は、入力の 2 回の変化の間の経過時間に対応し、モータの速度情報を与えます。

「インタフェースタイマ」を出力モードで使用して、(COM イベントをトリガすることで) 高機能制御タイマ (TIM1 または TIM8) のチャンネルの設定を変更するパルスを生成することができます。TIM1 タイマは、モータを駆動する PWM 信号を生成するために使用されます。このためには、プログラミングした遅延の後に正パルスが生成されるように (出力比較モードまたは PWM モードで) インタフェースタイマチャンネルをプログラミングする必要があります。このパルスは、TRGO 出力を通じて高機能制御タイマ (TIM1 または TIM8) に送られます。

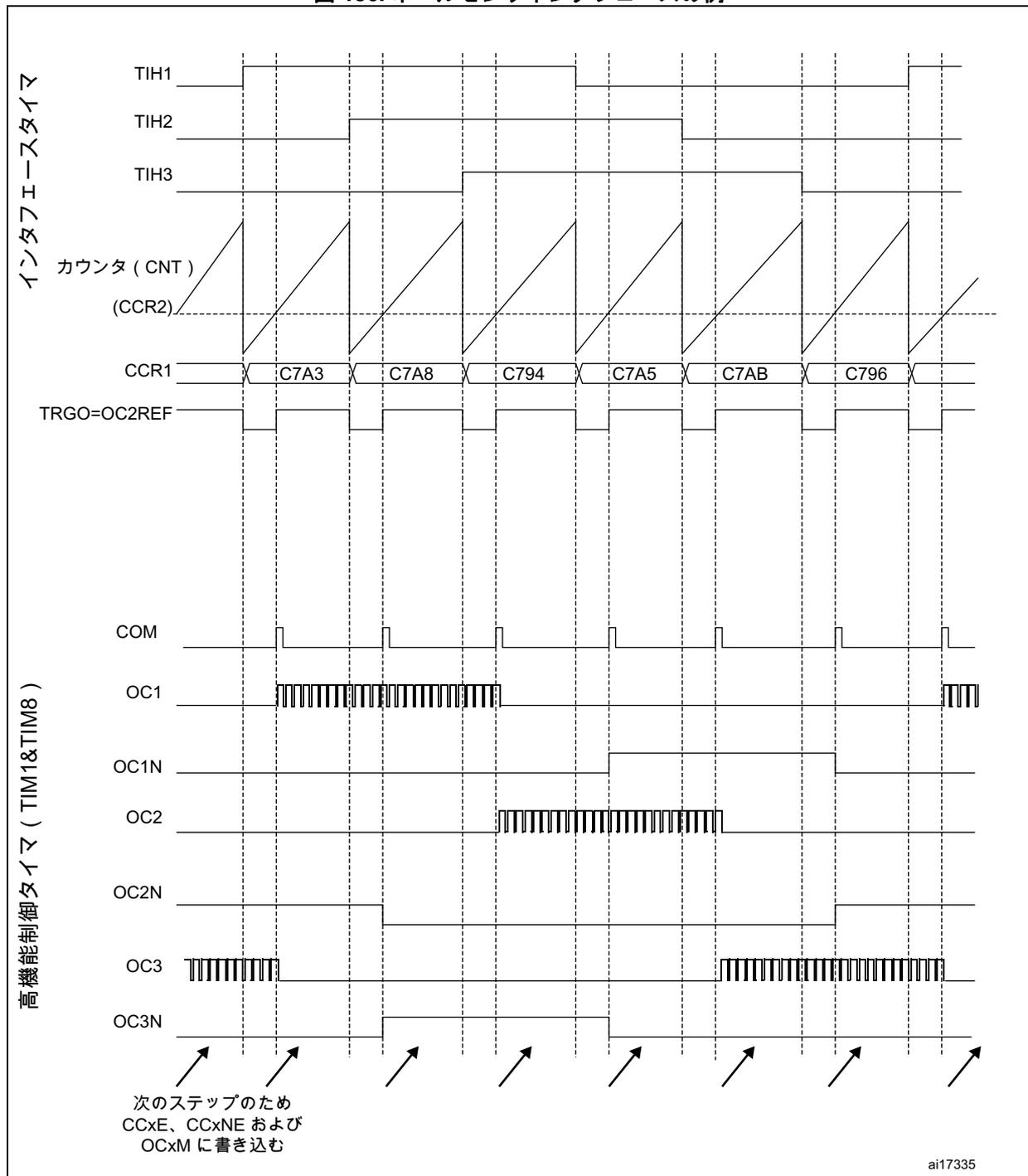
例：TIMx タイマの 1 つに接続されているホール入力に変化するたびに、プログラミングした遅延の後に高機能制御タイマ TIM1 の PWM 設定を変更するとします。

- 3 つのタイマ入力を TI1 入力チャンネルに OR 接続します。このためには、TIMx\_CR2 レジスタの TI1S ビットに“1”を書き込みます。
- タイムベースをプログラムします。このためには、TIMx\_ARR に最大値を書き込み、TI1 の変化でカウンタがクリアされるようにします。最大カウンタ時間がセンサの 2 回の変化の間の時間より長くなるように、プリスケアラを設定します。
- チャンネル 1 をキャプチャモード (TRC 選択) にプログラムします。すなわち、TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。必要な場合は、デジタルフィルタをプログラムすることもできます。
- チャンネル 2 を PWM 2 モードにプログラミングし、希望の遅延を指定します。このためには、TIMx\_CCMR1 レジスタの OC2M ビットに“111”を、CC2S ビットに“00”を書き込みます。
- TRGO 上のトリガ出力として OC2REF を選択します。このためには、TIMx\_CR2 レジスタの MMS ビットに“101”を書き込みます。

高機能制御タイマ TIM1 で、トリガ入力として適切な ITR 入力を選択する必要があり、タイマが PWM 信号を生成するようにプログラミングし、キャプチャ/比較制御信号がプリロードされ (TIMx\_CR2 レジスタの CCPC=1)、COM イベントがトリガ入力によって制御されなければなりません (TIMx\_CR2 レジスタの CCUS=1)。PWM 制御ビット (CCxE、OCxM) は、COM イベント後に次のステップのために書き込まれます (これは、OC2REF の立ち上がりエッジによって生成される割り込みサブルーチンで行うことができます)。

 186 に、この例を示します。

図 186. ホールセンサインタフェースの例



## 22.3.25 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。これらは、リセットモード、ゲートモード、およびトリガモードで同期できます。

### スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx\_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx\_ARR、TIMx\_CCRx) が更新されます。

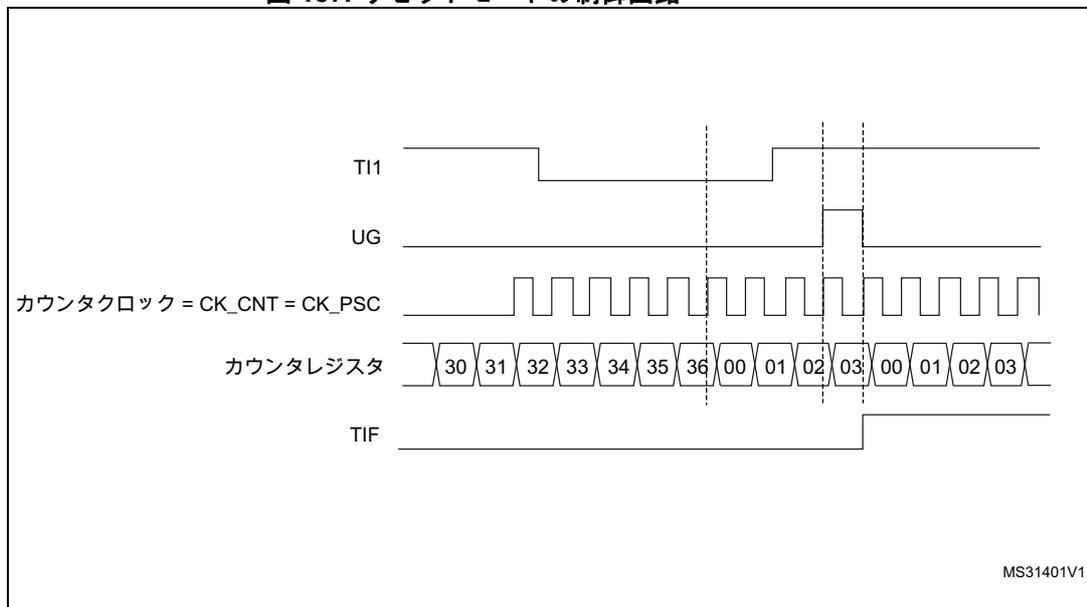
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01)。TIMx\_CCER レジスタに CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx\_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx\_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx\_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx\_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 187. リセットモードの制御回路



## スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

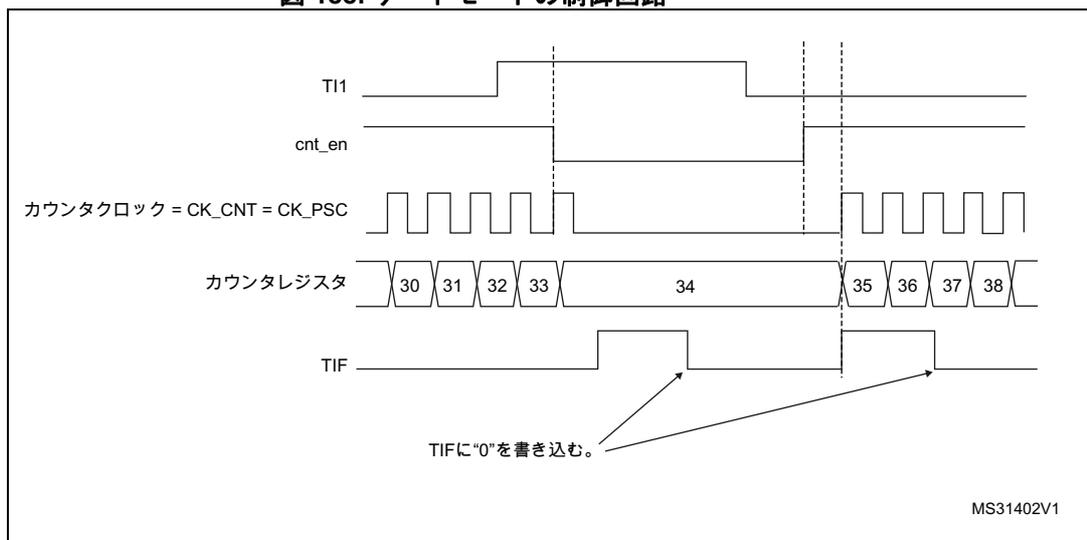
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01 ビット)。TIMx\_CCER レジスタで CC1P=1 と CC1NP=0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx\_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx\_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx\_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 188. ゲートモードの制御回路



## スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

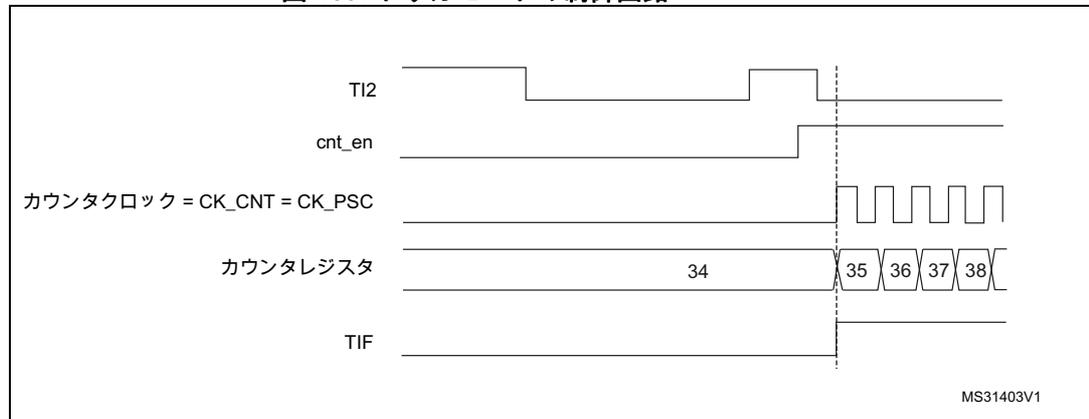
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx\_CCMR1 レジスタの CC2S=01)。TIMx\_CCER レジスタで CC2P=1 と CC2NP=0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 189. トリガモードの制御回路



### スレーブモード：リセットモードとトリガモードの組み合わせ

この場合、選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

### スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます (ただし、外部クロックモード 1 とエンコーダモードは除きます)。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます (リセットモード、ゲートモード、またはトリガモード)。TIMx\_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

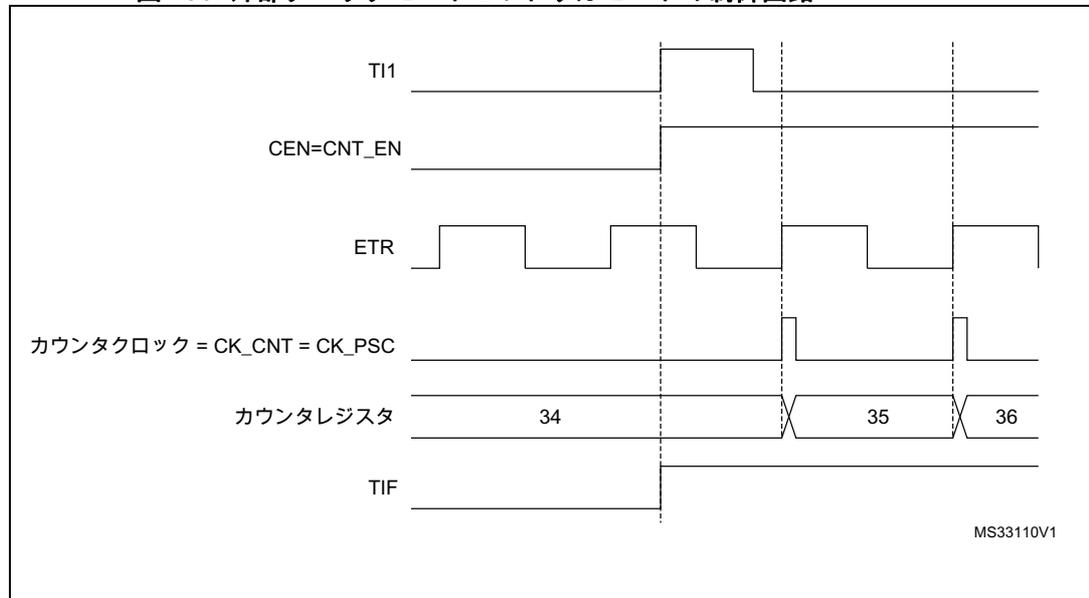
次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

- TIMx\_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
  - ETF = 0000 : フィルタなし
  - ETPS=00 : プリスケール無効
  - ETP=0 : ETR の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
  - IC1F=0000 : フィルタなし。
  - キャプチャプリスケールはトリガには使用されないため、設定する必要はありません。
  - TIMx\_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
  - TIMx\_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

T11 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 190. 外部クロックモード 2+トリガモードの制御回路



**注:** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間は動作中に変更しないでください。

## 22.3.26 ADC の同期

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイベントを生成できます。以下のような内部エッジ検出から発行されたパルスを生成することもできます。

- OC4ref の立ち上がりおよび立ち下がりエッジ
- OC5ref の立ち上がりエッジまたは OC6ref の立ち下がりエッジ

トリガは、ADC にリダイレクトされる TRGO2 内部ラインで発行されます。使用できるイベントは全部で 16 個あり、TIMx\_CR2 レジスタの MMS2[3:0] ビットを使用して選択できます。

3 相モータ駆動のアプリケーションの例については、[図 171 \(644 ページ\)](#) を参照してください。

**注:** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間は動作中に変更しないでください。

**注:** ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間は動作中に変更しないでください。

## 22.3.27 DMA バーストモード

TIMx タイマには、1つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、タイマの一部をソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx\_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト (バースト) を開始します。TIMx\_DMAR レジスタへの各書き込みは、実際にタイマレジスタの1つにリダイレクトされます。

TIMx\_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数 (ハーフワードまたはバイト) です。

TIMx\_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx\_CR1

00001 : TIMx\_CR2

00010 : TIMx\_SMCR

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
  - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
  - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
  - 転送データ数 = 3 とします (下の注を参照)。
  - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。  
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタの UDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注 :

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

## 22.3.28 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex<sup>®</sup>-M7 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 40.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 22.4 TIM1/TIM8 レジスタ

レジスタの説明で使用されている略語のリストを参照してください。

### 22.4.1 TIM1/TIM8 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFRE MAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				rW		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (ETR、TIx) によって使用されるデッドタイムおよびサンプリングクロック ( $t_{DTS}$ ) との間の分周比を示します。

00 :  $t_{DTS}=t_{CK\_INT}$

01 :  $t_{DTS}=2*t_{CK\_INT}$

10 :  $t_{DTS}=4*t_{CK\_INT}$

11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:5 **CMS[1:0]** : センターアラインモード選択

00 : エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01 : センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10 : センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。

11 : センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

**注 :** カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

0 : カウンタはアップカウンタとして使用されます。

1 : カウンタはダウンカウンタとして使用されます。

**注 :** このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。
  - カウンタオーバーフロー/アンダーフロー
  - UG ビットのセット
  - スレーブモードコントローラからの更新生成
- 1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。
  - カウンタオーバーフロー/アンダーフロー
  - UG ビットのセット
  - スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

- 1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

**注 :** 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

## 22.4.2 TIM1/TIM8 制御レジスタ 2 (TIMx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **MMS2[3:0]** : マスタモード選択 2

これらのビットにより、選択される同期 (TRGO2) について、ADC に情報を送信できるようになります。組み合わせは、次のとおりです。

0000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO2) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO2 信号は実際のリセットより遅延します。

0001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO2) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO2 には遅延が存在しません (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

0010 : **更新** - 更新イベントがトリガ出力 (TRGO2) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。

0011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1F フラグがセットされる時 (すでにハイであった場合も)、トリガ出力は正のパルスを送信します (TRGO2)。

0100 : **比較** - OC1REF 信号がトリガ出力 (TRGO2) として使用されます。

0101 : **比較** - OC2REF 信号がトリガ出力 (TRGO2) として使用されます。

0110 : **比較** - OC3REF 信号がトリガ出力 (TRGO2) として使用されます。

0111 : **比較** - OC4REF 信号がトリガ出力 (TRGO2) として使用されます。

1000 : **比較** - OC5REF 信号がトリガ出力 (TRGO2) として使用されます。

1001 : **比較** - OC6REF 信号がトリガ出力 (TRGO2) として使用されます。

1010 : **パルス比較** - OC4REF の立ち上がりエッジまたは立ち下がりエッジによって、TRGO2 にパルスが生成されます。

1011 : **パルス比較** - OC6REF の立ち上がりエッジまたは立ち下がりエッジによって、TRGO2 にパルスが生成されます。

1100 : **パルス比較** - OC4REF または OC6REF の立ち上がりエッジによって、TRGO2 にパルスが生成されます。

1101 : **パルス比較** - OC4REF の立ち上がりエッジまたは OC6REF の立ち下がりエッジによって、TRGO2 にパルスが生成されます。

1110 : **パルス比較** - OC5REF または OC6REF の立ち上がりエッジによって、TRGO2 にパルスが生成されます。

1111 : **パルス比較** - OC5REF の立ち上がりエッジまたは OC6REF の立ち下がりエッジによって、TRGO2 にパルスが生成されます。

**注 :** **スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間は動作中に変更しないでください。**

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **OIS6** : 出力アイドル状態 6 (OC6 出力)

OIS1 ビットの説明を参照してください。

ビット 17 予約済みであり、リセット値に保持する必要があります。



- ビット 16 **OIS5** : 出力アイドル状態 5 (OC5 出力)  
OIS1 ビットの説明を参照してください。
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **OIS4** : 出力アイドル状態 4 (OC4 出力)  
OIS1 ビットの説明を参照してください。
- ビット 13 **OIS3N** : 出力アイドル状態 3 (OC3N 出力)  
OIS1N ビットの説明を参照してください。
- ビット 12 **OIS3** : 出力アイドル状態 3 (OC3 出力)  
OIS1 ビットの説明を参照してください。
- ビット 11 **OIS2N** : 出力アイドル状態 2 (OC2N 出力)  
OIS1N ビットの説明を参照してください。
- ビット 10 **OIS2** : 出力アイドル状態 2 (OC2 出力)  
OIS1 ビットの説明を参照してください。
- ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)  
0 : MOE=0 のとき、デッドタイム後に OC1N=0  
1 : MOE=1 のとき、デッドタイム後に OC1N=0  
**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。
- ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)  
0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)  
1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)  
**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。
- ビット 7 **TI1S** : TI1 選択  
0 : TIMx\_CH1 ピンが TI1 入力に接続されます。  
1 : TIMx\_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。
- ビット 6:4 **MMS[1:0]** : マスタモード選択  
これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。  
000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。  
001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役に立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、GEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在しません (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。  
010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。  
011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる時 (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。 (TRGO)  
100 : **比較** - OC1REF 信号がトリガ出力 (TRGO) として使用されます。  
101 : **比較** - OC2REF 信号がトリガ出力 (TRGO) として使用されます。  
110 : **比較** - OC3REF 信号がトリガ出力 (TRGO) として使用されます。  
111 : **比較** - OC4REF 信号がトリガ出力 (TRGO) として使用されます。  
**注 :** スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

- 0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。
- 1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ/比較制御更新選択

- 0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。
- 1 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ/比較プリロード制御

- 0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。
- 1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込みの後、これらのビットは、転換イベント (COM) が発生した時にのみ更新されます (CCUS ビットに応じて、COMG ビットがセットまたは TRGI で立ち上がりエッジが検出されたとき)。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

## 22.4.3 TIM1/TIM8 のスレーブモード制御レジスタ (TIMx\_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]			MSM	TS[2:0]		Res.	SMS[2:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SMS[3]** : スレーブモード選択 - ビット 3

SMS 説明を参照 - ビット 2:0

ビット 15 **ETP** : 外部トリガ極性

このビットは、ETR と ETR のいずれがトリガ動作に使用されるかを選択します。

- 0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。
- 1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブです。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

- 0 : 外部クロックモード 2 は無効です。
- 1 : 外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

**注 :** 1: ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=111)。

2: 外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません (TS ビットが 111 でないことが必要)。

3: 外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力は ETRF です。



### ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、TIMxCLK 周波数の 1/4 までに制限されます。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。

- 00 : プリスケアラオフ
- 01 : ETRP 周波数は 2 分周されます。
- 10 : ETRP 周波数は 4 分周されます。
- 11 : ETRP 周波数は 8 分周されます。

### ビット 11:8 **ETF[3:0]** : 外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

- 0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。
- 0001 :  $f_{SAMPLING} = f_{CK\_INT}$ 、 $N = 2$
- 0010 :  $f_{SAMPLING} = f_{CK\_INT}$ 、 $N = 4$
- 0011 :  $f_{SAMPLING} = f_{CK\_INT}$ 、 $N = 8$
- 0100 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 6$
- 0101 :  $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$
- 0110 :  $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$
- 0111 :  $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$
- 1000 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 8$
- 1001 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 8$
- 1010 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 16$
- 1011 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 16$
- 1100 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 16$
- 1101 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 32$
- 1110 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 32$
- 1111 :  $f_{SAMPLING} = f_{DTS}$ 、 $N = 32$

### ビット 7 **MSM** : マスタ/スレーブモード

- 0 : 影響なし。
- 1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

### ビット 6:4 **TS[2:0]** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)
- 100 : TI1 エッジ検出回路 (TI1F\_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)
- 111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、表 125 : TIMx 内部トリガ接続 (675 ページ) を参照してください。

**注 :** 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : エンコーダモード 1 - カウンタは、TI1FP2 のレベルに応じて、TI2FP1 のエッジでカウントアップ/ダウンします。

0010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。

0011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。

0100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

1000 以上のコード : 予約済み

**注 :** トリガ入力として **TI1F\_ED** が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F\_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

**注 :** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 125. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

## 22.4.4 TIM1/TIM8 DMA／割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル

0 : COM DMA リクエストは無効です。

1 : COM DMA リクエストは有効です。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割り込みイネーブル

0 : ブレーク割り込みは無効です。

1 : ブレーク割り込みは有効です。

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5 **COMIE** : COM 割り込みイネーブル

0 : COM 割り込みは無効です。

1 : COM 割り込みは有効です。

ビット 4 **CC4IE** : キャプチャ／比較 4 割り込みイネーブル

0 : CC4 割り込みは無効です。

1 : CC4 割り込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割り込みイネーブル

0 : CC3 割り込みは無効です。

1 : CC3 割り込みは有効です。

ビット 2 **CC2IE** : キャプチャ/比較 2 割り込みイネーブル

- 0 : CC2 割り込みは無効です。
- 1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル

- 0 : CC1 割り込みは無効です。
- 1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

- 0 : 更新割り込みは無効です。
- 1 : 更新割り込みは有効です。

## 22.4.5 TIM1/TIM8 のステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C6IF	C5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.		CC4OF	CC3OF	CC2OF	CC1OF	B2IF	B1F	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0												

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CC6IF** : 比較 6 割り込みフラグ

CC1IF の説明を参照してください。(注 : チャンネル 6 は出力としてのみ設定できます。)

ビット 16 **CC5IF** : 比較 5 割り込みフラグ

CC1IF の説明を参照してください。(注 : チャンネル 5 は出力としてのみ設定できます。)

ビット 15 : 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8 **B2IF** : ブレーク 2 割り込みフラグ

このフラグは、ブレーク 2 入力アクティブになると、ハードウェアによってセットされます。ブレーク 2 入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク 2 入力アクティブレベルが検出されました。TIMx\_DIER レジスタの BIE=1 の場合、割り込みが生成されます。



ビット 7 **BIF** : ブレーク割り込みフラグ

このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク入力アクティブレベルが検出されました。TIMx\_DIER レジスタの BIE=1 の場合、割り込みが生成されます。

ビット 6 **TIF** : トリガ割り込みフラグ

このフラグは、トリガイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力アクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

0 : トリガイベントは発生していません。

1 : トリガ割り込みが保留中です。

ビット 5 **COMIF** : COM 割り込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき)。ソフトウェアによってクリアされます。

0 : COM イベントは発生していません。

1 : COM 割り込みがペンディング中です。

ビット 4 **CC4IF** : キャプチャ/比較 4 割り込みフラグ

CC1IF の説明を参照してください。

ビット 3 **CC3IF** : キャプチャ/比較 3 割り込みフラグ

CC1IF の説明を参照してください。

ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合** : このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます (センターアラインモードでは、例外もあります。TIMx\_CR1 レジスタの CMS ビットの説明を参照してください)。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。TIMx\_CCR1 の内容が TIMx\_ARR の内容より大きいときには、カウンタオーバーフロー時 (アップカウントおよびアップ/ダウンカウントモードの場合)、またはアンダーフロー時 (ダウンカウントモードの場合) に CC1IF ビットはハイになります。

**CC1 チャンネルが入力として設定されている場合** : このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフローまたはアンダーフロー (繰り返しカウンタ=0 の場合の更新)、および TIMx\_CR1 レジスタの UDIS=0 の場合。
- TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき (セクション 22.4.3 : TIM1/TIM8 のスレーブモード制御レジスタ (TIMx\_SMCR) を参照)。

## 22.4.6 TIM1/TIM8 のイベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	B2G	バックラウンド	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG						
							w	w	w	w	w	w	w	w	w

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **B2G** : ブレーク 2 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレーク 2 イベントが生成されます。MOE ビットがクリアされ、B2IF フラグがセットされます。有効な場合は、関連する割り込みが発生します。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx\_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ/比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4 **CC4G** : キャプチャ/比較 4 イベント生成

CC1G の説明を参照してください。



- ビット 3 **CC3G** : キャプチャ/比較 3 イベント生成  
CC1G の説明を参照してください。
- ビット 2 **CC2G** : キャプチャ/比較 2 イベント生成  
CC1G の説明を参照してください。
- ビット 1 **CC1G** : キャプチャ/比較 1 イベント生成  
このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。  
0 : 影響なし。  
1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。  
**CC1 チャンネルが出力として設定されている場合 :**  
CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。  
**CC1 チャンネルが入力として設定されている場合 :**  
カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。
- ビット 0 **UG** : 更新生成  
このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。  
0 : 影響なし。  
1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx\_ARR) をとります。

### 22.4.7 TIM1/TIM8 のキャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]
							Res								Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]			IC1F[3:0]					IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

## 出力比較モード :

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OC2M[3]** : 出力比較 2 モード - ビット 3

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

Bits16 **OC1M[3]** : 出力比較 1 モード - ビット 3

ビット 6:4 の OC1M 説明を参照

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC2S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

OC1CE : 出力比較 1 クリアイネーブル

0 : OC1Ref は ETRF 入力の影響を受けません。

1 : OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

### ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

0001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

0010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

0011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

0100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

0101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 OR です。

1101 : 組み合わせ PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 AND です。

1110 : 非対称 PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

1111 : 非対称 PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

**注 :** これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

**注 :** PWM モードでは、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされません。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとりず。

**ビット 3 OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注** : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを有効にしなくても使用できます。そうでない場合、動作は保証されません。

**ビット 2 OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

**ビット 1:0 CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注** : CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケーラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC2S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

0001 :  $f_{SAMPLING} = f_{CK\_INT}$ , N = 2

0010 :  $f_{SAMPLING} = f_{CK\_INT}$ , N = 4

0011 :  $f_{SAMPLING} = f_{CK\_INT}$ , N = 8

0100 :  $f_{SAMPLING} = f_{DTS}$ , N = 6

0101 :  $f_{SAMPLING} = f_{DTS}/2$ , N = 8

0110 :  $f_{SAMPLING} = f_{DTS}/4$ , N = 6

0111 :  $f_{SAMPLING} = f_{DTS}/4$ , N = 8

1000 :  $f_{SAMPLING} = f_{DTS}$ , N = 8

1001 :  $f_{SAMPLING} = f_{DTS}$ , N = 8

1010 :  $f_{SAMPLING} = f_{DTS}$ , N = 16

1011 :  $f_{SAMPLING} = f_{DTS}$ , N = 16

1100 :  $f_{SAMPLING} = f_{DTS}$ , N = 16

1101 :  $f_{SAMPLING} = f_{DTS}$ , N = 32

1110 :  $f_{SAMPLING} = f_{DTS}$ , N = 32

1111 :  $f_{SAMPLING} = f_{DTS}$ , N = 32

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケーラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケーラの分周比を定義します。プリスケーラは、CC1E=0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケーラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC1S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 22.4.8 TIM1/TIM8 のキャプチャ/比較モードレジスタ 2 (TIMx\_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

上記の CCMR1 レジスタの説明を参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							Res.								Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

### 出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OC4M[3]** : 出力比較 4 モード - ビット 3

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **OC3M[3]** : 出力比較 3 モード - ビット 3

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC3S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。



## 入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケール

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときのみ書き込み可能です。

ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケール

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC3S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときのみ書き込み可能です。

## 22.4.9 TIM1/TIM8 のキャプチャ/比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rW	rW			rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rW		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **CC6P** : キャプチャ/比較 6 出力極性

CC1P の説明を参照してください。

ビット 20 **CC6E** : キャプチャ/比較 6 出力イネーブル

CC1E の説明を参照してください。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CC5P** : キャプチャ/比較 5 出力極性

CC1P の説明を参照してください。

ビット 16 **CC5E** : キャプチャ/比較 5 出力イネーブル

CC1E の説明を参照してください。

- ビット 15 **CC4NP** : キャプチャ/比較 4 相補出力極性  
CC1NP の説明を参照してください。
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **CC4P** : キャプチャ/比較 4 出力極性  
CC1P の説明を参照してください。
- ビット 12 **CC4E** : キャプチャ/比較 4 出カイナーブル  
CC1E の説明を参照してください。
- ビット 11 **CC3NP** : キャプチャ/比較 3 相補出力極性  
CC1NP の説明を参照してください。
- ビット 10 **CC3NE** : キャプチャ/比較 3 相補出カイナーブル  
CC1NE の説明を参照してください。
- ビット 9 **CC3P** : キャプチャ/比較 3 出力極性  
CC1P の説明を参照してください。
- ビット 8 **CC3E** : キャプチャ/比較 3 出カイナーブル  
CC1E の説明を参照してください。
- ビット 7 **CC2NP** : キャプチャ/比較 2 相補出力極性  
CC1NP の説明を参照してください。
- ビット 6 **CC2NE** : キャプチャ/比較 2 相補出カイナーブル  
CC1NE の説明を参照してください。
- ビット 5 **CC2P** : キャプチャ/比較 2 出力極性  
CC1P の説明を参照してください。
- ビット 4 **CC2E** : キャプチャ/比較 2 出カイナーブル  
CC1E の説明を参照してください。
- ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性  
**CC1 チャンネルが出力として設定されている場合 :**  
0 : OC1N はアクティブハイです。  
1 : OC1N はアクティブローです。  
**CC1 チャンネルが入力として設定されている場合 :**  
このビットは、TI1FP1とTI2FP1の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。  
**注 :** このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。  
**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。
- ビット 2 **CC1NE** : キャプチャ/比較 1 相補出カイナーブル  
0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N、および CC1E ビットによって決まります。  
1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。  
**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1NE アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

**CC1 チャンネルが出力として設定されている場合 :**

- 0 : OC1 はアクティブハイです。
- 1 : OC1 はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :** CC1NP/CC1Pビットは、トリガまたはキャプチャ動作に対するTI1FP1とTI2FP1のアクティブ極性を選択します。

00 : 非反転/立ち上がりエッジ。この回路は TIXFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIXFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

01 : 反転/立ち下がりエッジ。この回路は TIXFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIXFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジこの回路は TIXFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIXFP1 は反転されません (ゲートモードでのトリガ動作)。この設定をエンコーダモードで使用することはできません。

**注 :** このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx\_BDTR レジスタの LOCK ビット)。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル

**CC1 チャンネルが出力として設定されている場合 :**

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :** このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1E アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

表 126. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能

制御ビット					出力状態 <sup>(1)</sup>	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	X	0	0	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=0、OCxN=0	
		0	0	1	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP
		0	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCxN=0
		X	1	1	OCREF + 極性 + デッドタイム	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム
		1	0	1	オフ状態 (インアクティブ状態で出力有効) OCx=CCxP	OCxREF + 極性 OCxN = OCxREF xor CCxNP
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP	オフ状態 (インアクティブ状態で出力有効) OCxN=CCxNP
0	1	X	X	X	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=CCxP、OCxN=CCxNP	
			0	0		
			0	1	オフ状態 (インアクティブ状態で出力有効)	
			1	0	非同期: OCx=CCxP、OCxN=CCxNP (BRK または BRK2 がトリガされた場合)	
			1	1	クロックがある場合 (これは BRK がトリガされている場合にのみ有効): アクティブな状態で OISx と OISxN が OCx と OCxN にそれぞれ対応しないとみなされる場合 (ハーフブリッジ設定でスイッチを駆動した場合に短絡の原因となる)、デッドタイム後に OCx=OISx および OCxN=OISxN となります。 注: BRK2 は OSSI = OSSR = 1 の場合にのみ使用できます。	

1. チャンネルの両方の出力が使用されないとき (GPIO が制御を引き継いだ場合)、OISx、OISxN、CCxP、および CCxNP はクリアされたままではなければなりません。

注: 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OCx および OCxN チャンネルの状態と、GPIO レジスタに依存します。



## 22.4.10 TIM1/TIM8 のカウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー。TIMxCR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

## 22.4.11 TIM1/TIM8 プリスケアラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 (CK\_CNT) は  $f_{CK\_PSC}/(PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx\_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

## 22.4.12 TIM1/TIM8 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **ARR[15:0]**:プリスケアラ値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 22.3.1 : タイムベースユニット \(611 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 22.4.13 TIM1/TIM8 繰り返しカウンタレジスタ (TIMx\_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **REP[15:0]** : 繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート (プリロードレジスタからアクティブレジスタへの定期的な転送) と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP\_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U\_RC でのみ、REP\_CNT に REP 値がロードされるので、TIMx\_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) は次のことを意味します。

エッジアラインモードでは、PWM 周期の数

センターアラインモードでは、PWM の 1/2 周期の数

## 22.4.14 TIM1/TIM8 キャプチャ/比較レジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値

**CC1 チャンネルが出力として設定されている場合** : CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**CC1 チャンネルが入力として設定されている場合** : CR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 22.4.15 TIM1/TIM8 キャプチャ/比較レジスタ 2 (TIMx\_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR2[15:0]** : キャプチャ/比較 2 値

**CC2 チャンネルが出力として設定されている場合** : CCR2 は、実際のキャプチャ/比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 2 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC2 出力に送信される値を含みます。

**CC2 チャンネルが入力として設定されている場合** : CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

## 22.4.16 TIM1/TIM8 キャプチャ/比較レジスタ 3 (TIMx\_CCR3)

アドレスオフセット : 0x3C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR3[15:0]** : キャプチャ/比較値

**CC3 チャンネルが出力として設定されている場合** : CCR3 は、実際のキャプチャ/比較 3 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 3 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC3 出力に送信される値を含みます。

**CC3 チャンネルが入力として設定されている場合** : CCR3 は、最後の入力キャプチャ 3 イベント (IC3) によって転送されたカウンタ値です。

## 22.4.17 TIM1/TIM8 キャプチャ/比較レジスタ 4 (TIMx\_CCR4)

アドレスオフセット : 0x40

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **CCR4[15:0]** : キャプチャ / 比較値

**CC4 チャンネルが出力として設定されている場合** : CCR4 は、実際のキャプチャ/比較 4 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 4 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC4 出力に送信される値を含みます。

**CC4 チャンネルが入力として設定されている場合** : CCR4 は、最後の入力キャプチャ 4 イベント (IC4) によって転送されたカウンタ値です。

## 22.4.18 TIM1/TIM8 ブレークおよびデッドタイムレジスタ (TIMx\_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]			BK3F[3:0]				
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w						

**注 :** ビット **BK2P**、**BK2E**、**BK2F[3:0]**、**BK3F[3:0]**、**AOE**、**BKP**、**BKE**、**OSSI**、**OSSR**、および **DTG[7:0]** は、**LOCK** 設定に応じて書き込みがロックされるので、**TIMx\_BDTR** レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 31:26 予約済み

ビット 25 **BK2P** : ブレーク 2 極性

0 : ブレーク入力 BRK2 はアクティフローです。

1 : ブレーク入力 BRK2 はアクティブハイです。

**注 :** このビットは、**LOCK** レベル 1 がプログラムされている場合 (**TIMx\_BDTR** レジスタの **LOCK** ビット)、変更できません。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

### ビット 24 BK2E : ブレーク 2 イネーブル

このビットは、完全なブレーク 2 保護を有効にします (bk\_acth および BKIN ソースにそれぞれ接続されたすべてのソースを含む)。

- 0 : ブレーク 2 機能は無効です。
- 1 : ブレーク 2 機能は有効です。

**注 :** BRKIN2 は OSSR = OSSI = 1 の場合にのみ使用してください。

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

### ビット 23:20 BK2F[3:0] : ブレーク 2 フィルタ

このビットフィールドは、BRK2 入力をサンプリングする周波数と、BRK2 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、BRK2 は非同期として動作します。

0001 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 2

0010 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 4

0011 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 8

0100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 6

0101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N = 8

0110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 6

0111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 8

1000 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1001 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1010 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1011 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

ビット 19:16 **BKF[3:0]** : ブレークフィルタ

このビットフィールドは、BRK 入力をサンプリングする周波数と、BRK に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、BRK は非同期として動作します。

0001 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 2

0010 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 4

0011 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 8

0100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 6

0101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ 、N = 8

0110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 6

0111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N = 8

1000 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1001 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1010 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1011 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

ビット 15 **MOE** : メイン出カイナーブル

このビットは、ブレーク入力の 1 つがアクティブになると、ハードウェアによって非同期にクリアされず (BRK または BRK2)。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : ブレーク 2 イベントへの対応。OC および OCN 出力は無効です。

ブレークイベントへの対応、または MOE が 0 に書き込まれた場合 : OC および OCN 出力が無効か、OSSI ビットによって強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのイナーブルビット (TIMx\_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN イナーブルの説明を参照してください (セクション 22.4.9 : TIM1/TIM8 のキャプチャ/比較有効レジスタ (TIMx\_CCER) (686 ページ))。

ビット 14 **AOE** : 自動出カイナーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力 BRK および BRK2 のいずれもがアクティブでない場合)。

**注 :** このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 **BKP** : ブレーク極性

0 : ブレーク入力 BRK はアクティブローです。

1 : ブレーク入力 BRK はアクティブハイです。

**注 :** このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

### ビット 12 BKE : ブレークイネーブル

このビットは、完全なブレーク保護を有効にします (bk\_acth および BKIN ソースにそれぞれ接続されたすべてのソースを含む)。

- 0 : ブレーク機能は無効です。
- 1 : ブレーク機能は有効です。

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

### ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 22.4.9 : TIM1/TIM8 のキャプチャ/比較有効レジスタ (TIMx\_CCER) (686 ページ))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する GPIO ロジックによって引き継がれます)。
- 1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります (出力は引き続きタイマで制御される)。

**注 :** このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

### ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE = 0 のとき、ブレークイベントまたはソフトウェアの書き込みにより、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 22.4.9 : TIM1/TIM8 のキャプチャ/比較有効レジスタ (TIMx\_CCER) (686 ページ))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する GPIO ロジックによって引き継がれます)。
- 1 : インアクティブのとき、OC/OCN 出力はまず強制的にインアクティブレベルにされ、次にデッドタイム後に強制的にアイドルレベルにされます。タイマは出力の制御を保持します。

**注 :** このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

### ビット 9:8 LOCK[1:0] : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

- 00 : LOCK オフ - どのビットも書き込み保護されません。
- 01 : LOCK レベル 1 = TIMx\_BDTR レジスタの DTG ビット、TIMx\_CR2 レジスタの OISx および OISxN ビット、および TIMx\_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。
- 10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込みできなくなります。
- 11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCMR レジスタの OCxM および OCxPE ビット) が書き込みできなくなります。

**注 :** LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx\_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

**ビット 7:0 DTG[7:0] : デッドタイムジェネレータのセットアップ**

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

DTG[7:5]=0xx =>  $DT=DTG[7:0] \times t_{dtg}$ 、ここで  $t_{dtg}=t_{DTS}$ 。

DTG[7:5]=10x =>  $DT=(64+DTG[5:0]) \times t_{dtg}$ 、ここで  $T_{dtg}=2 \times t_{DTS}$ 。

DTG[7:5]=110 =>  $DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで  $T_{dtg}=8 \times t_{DTS}$ 。

DTG[7:5]=111 =>  $DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで  $T_{dtg}=16 \times t_{DTS}$ 。

例 :  $T_{DTS}=125\text{ns}$  (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16  $\mu\text{s}$  から 31750 ns (250 ns 単位)

32  $\mu\text{s}$  から 63  $\mu\text{s}$  (1  $\mu\text{s}$  単位)

64  $\mu\text{s}$  から 126  $\mu\text{s}$  (2  $\mu\text{s}$  単位)

**注 :** このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

## 22.4.19 TIM1/TIM8 DMA 制御レジスタ (TIMx\_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送長 (タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します)、つまり転送回数を指定します。転送は、ハーフワードまたはバイトです (以下の例を参照)。

00000 : 1 回転送  
 00001 : 2 回転送  
 00010 : 3 回転送  
 ...  
 10001 : 18 回転送

例 : 次の転送を考えます : DBL = 7 バイトかつ DBA = TIM2\_CR1。

- DBL = 7 バイトおよび DBA = TIM2\_CR1 が転送するバイトのアドレスを表す場合、転送のアドレスは次の式で与えられます。

(TIMx\_CR1 アドレス) + DBA + (DMA インデックス)、ここで DMA インデックス = DBL

この例では、(TIMx\_CR1 アドレス) + DBA に 7 バイトが追加され、データのコピー元/コピー先アドレスが与えられます。この場合、転送は、(TIMx\_CR1 アドレス) + DBA から始めて、7 つのレジスタに対して行われます。

DMA データサイズの設定に応じて、いくつかのケースが想定されます。

- DMA データサイズをハーフワードで設定した場合、7 つのレジスタにそれぞれ 16 ビットのデータが転送されます。
- DMA データサイズをバイトで設定した場合も、データは 7 つのレジスタに転送されます (最初のレジスタには最初の MSB バイトが含まれ、2 番目のレジスタには最初の LSB バイトが含まれるなど、以降同様)。タイマへの転送で、DMA によって転送されるデータサイズを指定する必要もあります。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx\_CR1  
 00001 : TIMx\_CR2  
 00010 : TIMx\_SMCR  
 ...

## 22.4.20 フル転送のための TIM1/TIM8 DMA アドレス (TIMx\_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります : (TIMx\_CR1 アドレス) + (DBA + DMA インデックス) x 4。

ここで、TIMx\_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx\_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx\_DCR 内で設定)。

## 22.4.21 TIM1/TIM8 のキャプチャ/比較モードレジスタ 3 (TIMx\_CCMR3)

アドレスオフセット : 0x54

リセット値 : 0x0000 0000

上記の CCMR1 レジスタの説明を参照してください。チャンネル 5 および 6 は出力としてのみ設定できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

### 出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OC6M[3]** : 出力比較 6 モード - ビット 3

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **OC5M[3]** : 出力比較 5 モード - ビット 3

ビット 15 **OC6CE** : 出力比較 6 クリアイネーブル

ビット 14:12 **OC6M** : 出力比較 6 モード

ビット 11 **OC6PE** : 出力比較 6 プリロードイネーブル

ビット 10 **OC6FE** : 出力比較 6 高速イネーブル

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OC5CE** : 出力比較 5 クリアイネーブル

ビット 6:4 **OC5M** : 出力比較 5 モード



- ビット 3 **OC5PE** : 出力比較 5 プリロードイネーブル
- ビット 2 **OC5FE** : 出力比較 5 高速イネーブル
- ビット 1:0 予約済みであり、リセット値に保持する必要があります。

## 22.4.22 TIM1/TIM8 キャプチャ/比較レジスタ 5 (TIMx\_CCR5)

アドレスオフセット : 0x58

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.												
r/w	r/w	r/w													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 31 **GC5C3** : チャンネル 5 およびチャンネル 3 のグループ化  
 チャンネル 3 出力におけるひずみ :  
 0 : OC5REF の OC3REFC への影響はありません。  
 1 : OC3REFC は、OC3REFC と OC5REF の論理 AND です。  
 このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR2 でプリロード機能を選択している場合)。  
**注 :** このひずみを組み合わせ PWM 信号に適用することもできます。
- ビット 30 **GC5C2** : チャンネル 5 およびチャンネル 2 のグループ化  
 チャンネル 2 出力におけるひずみ :  
 0 : OC5REF の OC2REFC への影響はありません。  
 1 : OC2REFC は、OC2REFC と OC5REF の論理 AND です。  
 このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR1 でプリロード機能を選択している場合)。  
**注 :** このひずみを組み合わせ PWM 信号に適用することもできます。
- ビット 29 **GC5C1** : チャンネル 5 およびチャンネル 1 のグループ化  
 チャンネル 1 出力におけるひずみ :  
 0 : OC5REF の OC1REFC5 への影響はありません。  
 1 : OC1REFC は、OC1REFC と OC5REF の論理 AND です。  
 このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR1 でプリロード機能を選択している場合)。  
**注 :** このひずみを組み合わせ PWM 信号に適用することもできます。
- ビット 28:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15:0 **CCR5[15:0]** : キャプチャ/比較 5 値  
 CCR5 は、実際のキャプチャ/比較 5 レジスタにロードされる値 (プリロード値) です。  
 TIMx\_CCMR3 レジスタの OC5PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 5 レジスタにコピーされます。  
 アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC5 出力に送信される値を含みます。

## 22.4.23 TIM1/TIM8 キャプチャ/比較レジスタ 6 (TIMx\_CCR6)

アドレスオフセット : 0x5C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **CCR6[15:0]** : キャプチャ/比較 6 値

CCR6 は、実際のキャプチャ/比較 6 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR3 レジスタの OC6PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 6 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC6 出力に送信される値を含みます。

## 22.4.24 TIM1 レジスタマップ

TIM1 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 127. TIM1 レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIM1_CR1	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]	Res.	ARPE	Res.	CMS [1:0]	DIR	OPM	URS	UDIS	GEN															
	リセット値																						0		0	0	0	0	0	0	0	0	0
0x04	TIM1_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.															
	リセット値																																
0x08	TIM1_SMCR	Res.	SMS[3]	ETP	ECE	ETPS [1:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値																																
0x0C	TIM1_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.															
	リセット値																																
0x10	TIM1_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.															
	リセット値																																
0x14	TIM1_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.															
	リセット値																																



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

表 127. TIM1 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18	TIM1_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2OE	OC2M [2:0]		OC2PE	OC2FE	OC2S [1:0]	OC1OE	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIM1_CCMR1 入力キャプチャモード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F [3:0]		IC2PSC [1:0]	IC2S [1:0]	IC1F [3:0]		IC1PSC [1:0]	CC1S [1:0]								
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	TIM1_CCMR2 出力比較モード	Res	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	OC4OE	OC4M [2:0]		OC4PE	OC4FE	CC4S [1:0]	OC3OE	OC3M [2:0]		OC3PE	OC3FE	CC3S [1:0]				
	リセット値								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIM1_CCMR2 入力キャプチャモード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F [3:0]		IC4PSC [1:0]	IC4S [1:0]	IC3F [3:0]		IC3PSC [1:0]	CC3S [1:0]								
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	TIM1_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値												0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	TIM1_CNT	UIFCPY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	リセット値	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIM1_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIM1_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	TIM1_RCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REP[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	TIM1_CCR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR1[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIM1_CCR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR2[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIM1_CCR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR3[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIM1_CCR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR4[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIM1_BDTR	Res	Res	Res	Res	Res	Res	BK2P	BK2E	BK2F [3:0]			BKF [3:0]			MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT [7:0]										
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	TIM1_DCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBL [4:0]				Res	Res	Res	DBA [4:0]					
	リセット値																					0	0	0	0	0			0	0	0	0	0

表 127. TIM1 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x4C	TIM1_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	TIM1_CCMR3 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	OC5M[3]	OC6CE	OC6M [2:0]		OC8PE	OC8FE	Res.	Res.	OC5CE	OC5M [2:0]		OC5PE	OC5FE	Res.	Res.								
	リセット値								0									0	0	0	0	0	0	0			0	0	0	0	0	0	0
0x58	TIM1_CCR5	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	リセット値	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5C	TIM1_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 22.4.25 TIM8 レジスタマップ

TIM8 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 128. TIM8 レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	TIM8_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UJFREMAP	Res.	Res.	CKD [1:0]	Res.	ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN									
	リセット値																						0			0	0	0	0	0	0	0	0	0	
0x04	TIM8_CR2	Res.	MMS2[3:0]			Res.	Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	THS	MMS [2:0]		CCDS	CCUS	Res.	CCPC										
	リセット値									0	0	0	0			0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x08	TIM8_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]			MSM	TS[2:0]		Res.	SMS[2:0]														
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0C	TIM8_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE										
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	TIM8_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C6IF	C5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF									
	リセット値																0	0			0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	TIM8_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	バックグラウンド			TG	COM	CC4G	CC3G	CC2G	CC1G	UG								
	リセット値																									0	0	0	0	0	0	0	0	0	



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

表 128. TIM8 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18	TIM8_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2OE	OC2M [2:0]		OC2PE	OC2FE	OC2S [1:0]	OC1OE	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIM8_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]		IC2PSC [1:0]	IC2S [1:0]	IC1F[3:0]		IC1PSC [1:0]	CC1S [1:0]								
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	TIM8_CCMR2 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4OE	OC4M [2:0]		OC4PE	OC4FE	OC4S [1:0]	OC3OE	OC3M [2:0]		OC3PE	OC3FE	CC3S [1:0]				
	リセット値								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIM8_CCMR2 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]		IC4PSC [1:0]	IC4S [1:0]	IC3F[3:0]		IC3PSC [1:0]	CC3S [1:0]								
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	TIM8_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x24	TIM8_CNT	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																					
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIM8_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIM8_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	TIM8_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIM8_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIM8_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIM8_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIM8_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]			BK2F[3:0]			MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]										
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	TIM8_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]				
	リセット値																						0	0	0	0	0			0	0	0	0



表 128. TIM8 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x4C	TIM8_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	TIM8_CCMR3 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	OC5M[3]	OC6CE	OC6M [2:0]		OC8PE	OC8FE	Res.	Res.	OC5CE	OC5M [2:0]		OC5PE	OC5FE	Res.	Res.								
	リセット値							0									0	0	0	0	0	0			0	0	0	0	0	0	0	0	
0x58	TIM8_CCR5	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	リセット値	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	TIM8_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。



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

## 23 汎用タイマ (TIM2/TIM3/TIM4/TIM5)

### 23.1 TIM2/TIM3/TIM4/TIM5 の概要

この汎用タイマは、プログラマブルなプリスケラによって駆動される 16 ビットまたは 32 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較と PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

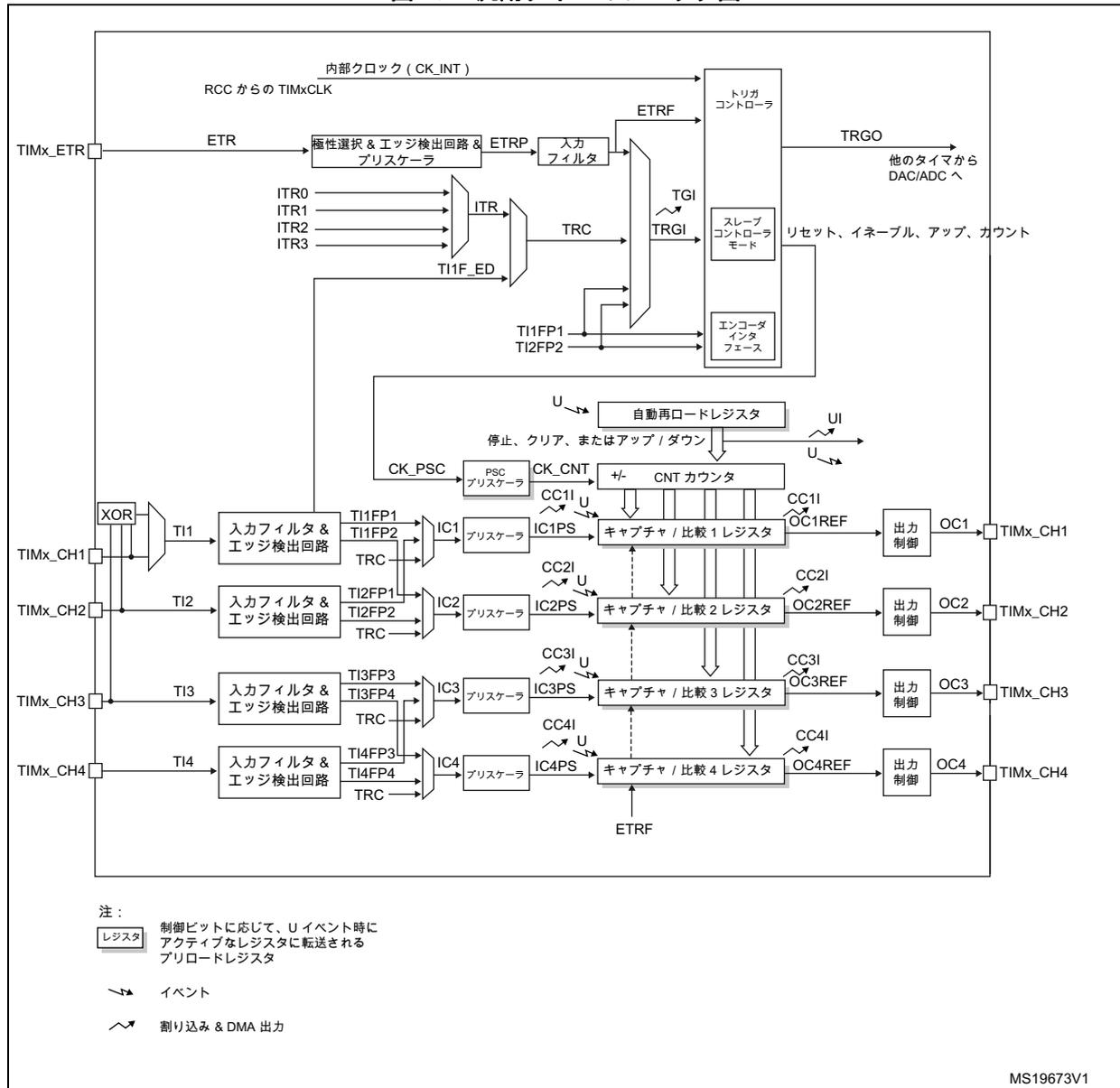
タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 23.3.18](#)に示すように、相互に同期させることができます。

### 23.2 TIM2/TIM3/TIM4/TIM5 の主な機能

汎用 TIMx タイマの主な機能は、次のとおりです。

- 16 ビット (TIM3、TIM4) または 32 ビット (TIM2 および TIM5) のアップ、ダウン、アップ/ダウン自動再ロードカウンタ。
- カウンタクロック周波数を、1 から 65535 の間で分周する 16 ビットプログラム可能プリスケラ。
- 次の機能を持つ、最大 4 つの独立チャンネル。
  - 入力キャプチャ
  - 出力比較
  - PWM 生成 (エッジアラインモードとセンターアラインモード)
  - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み/DMA 生成。
  - 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
  - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
  - 入力キャプチャ
  - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 191. 汎用タイマのブロック図



## 23.3 TIM2/TIM3/TIM4/TIM5 の機能説明

### 23.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビット/32 ビットカウンタです。カウンタは、カウントアップ、カウントダウン、またはカウントアップとカウントダウンの両方を行います。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC) :
- 自動再ロードレジスタ (TIMx\_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー (またはダウンカウント時はアンダーフロー) に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK\_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

実際のカウンタイネーブル信号 CNT\_EN は、CEN の 1 クロックサイクル後にセットされます。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビット/32 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 192](#) と [図 193](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 192. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

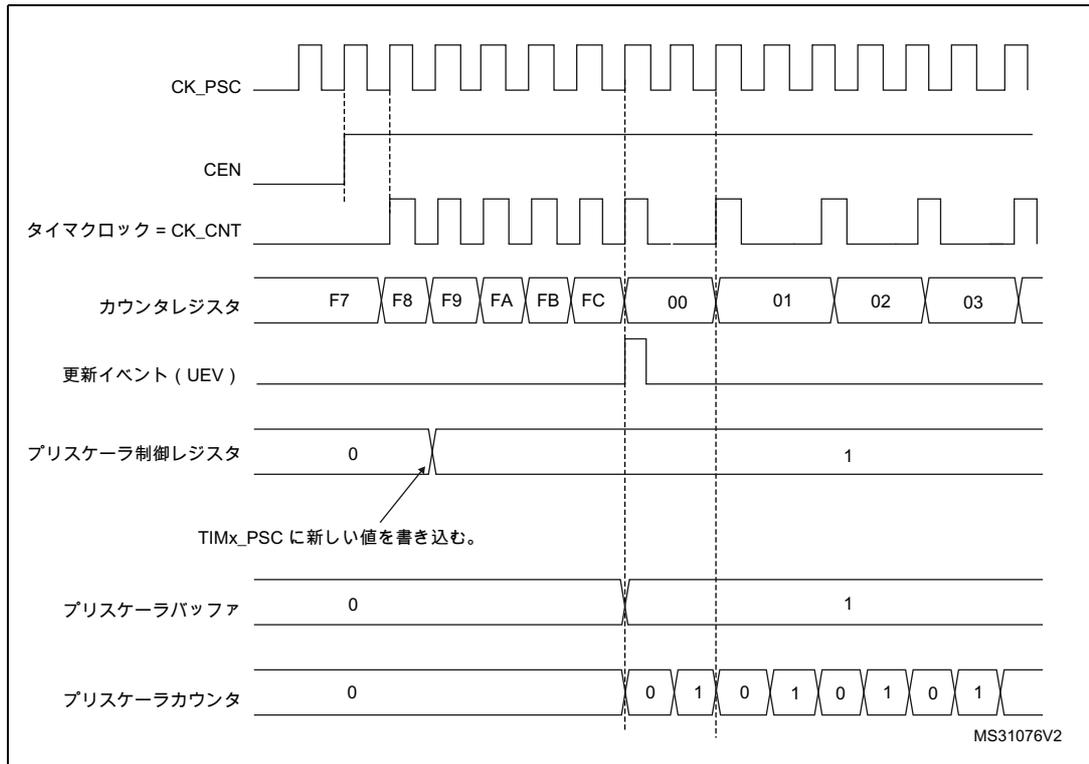
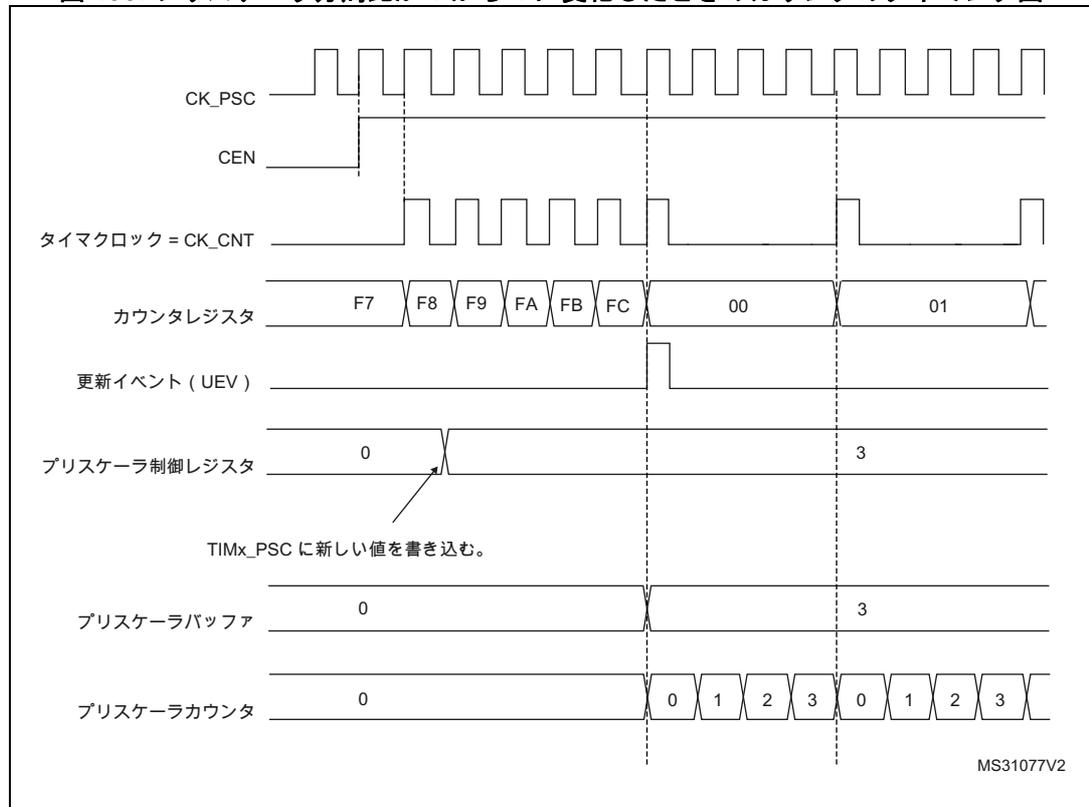


図 193. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 23.3.2 カウンタモード

### アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、(ソフトウェアで、または、スレープモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることで生成できます。

UEV イベントは、TIMx\_CR1 レジスタの UDIS ビットをセットすることにより、ソフトウェアで無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 194. 内部クロック分周比が 1 の場合のカウンタのタイミング図

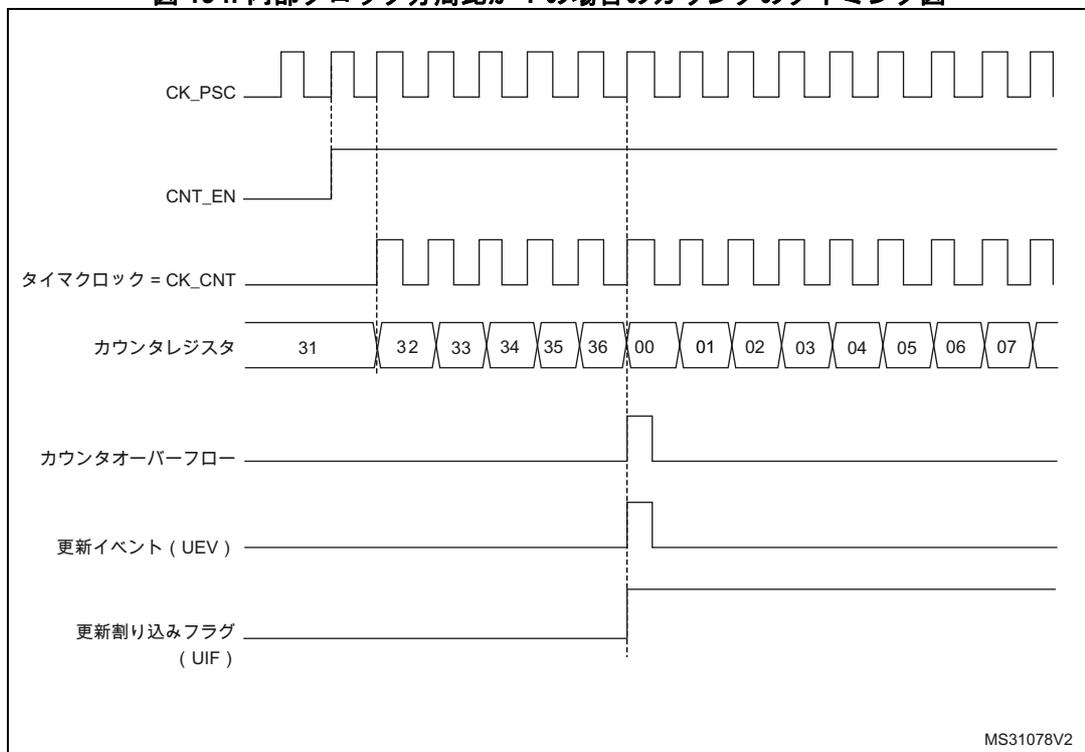


図 195. 内部クロック分周比が 2 の場合のカウンタのタイミング図

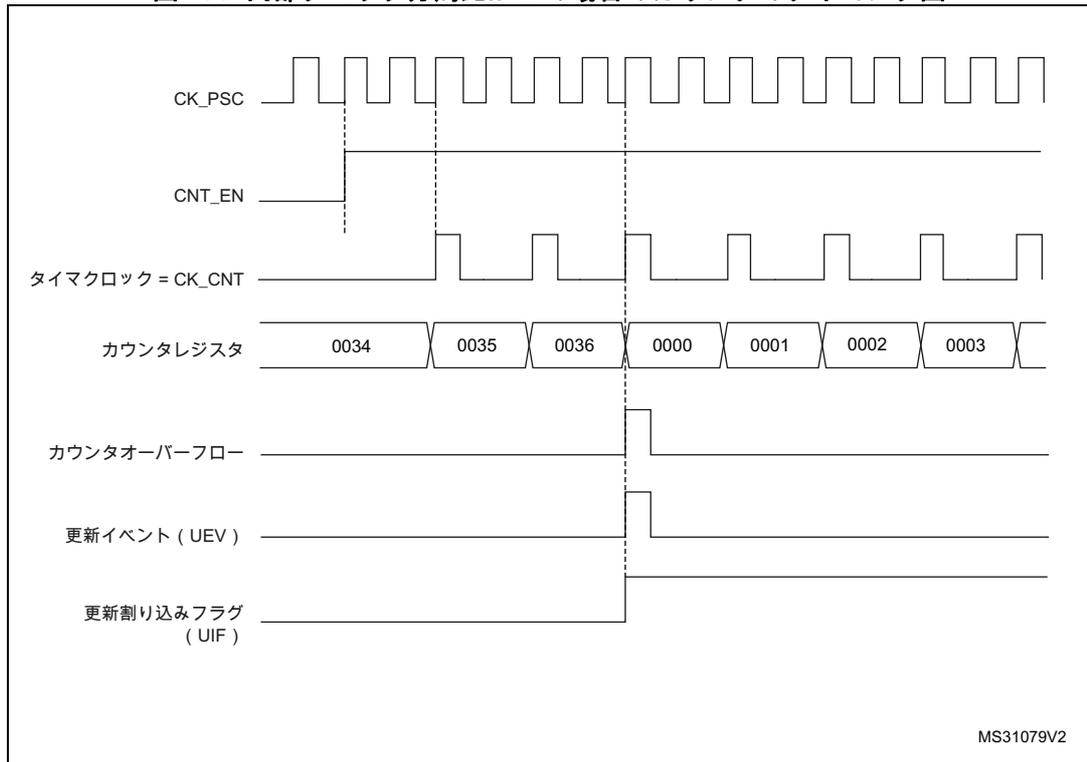


図 196. 内部クロック分周比が 4 の場合のカウンタのタイミング図

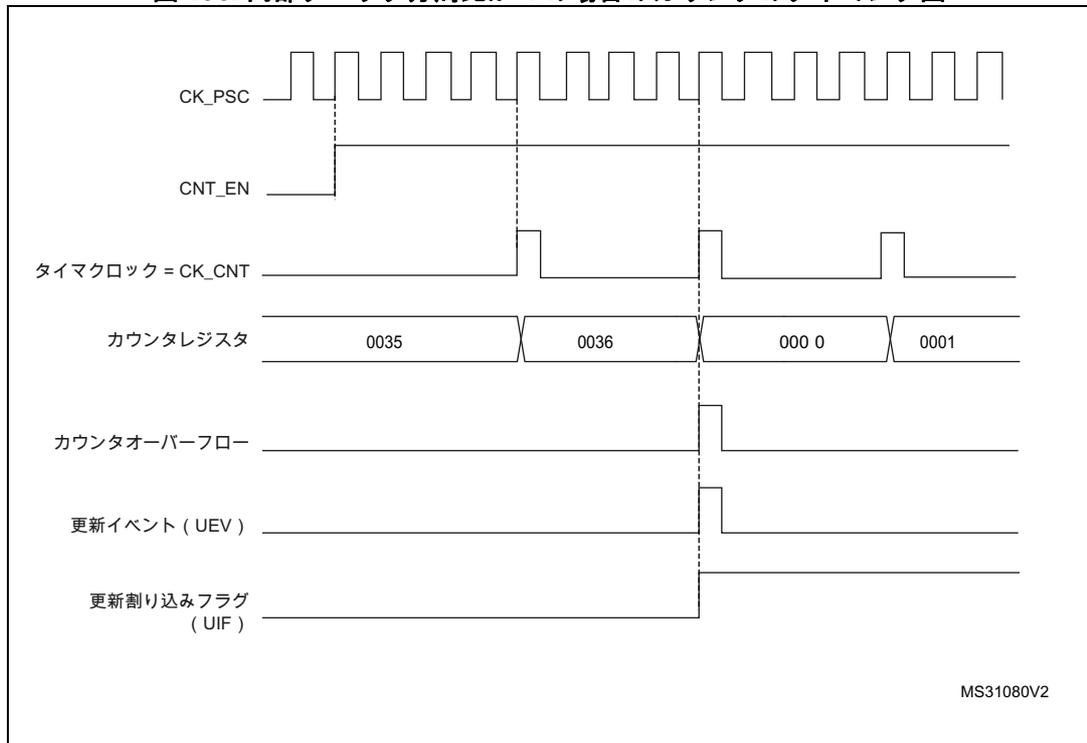


図 197. 内部クロック分周比が N の場合のカウンタのタイミング図

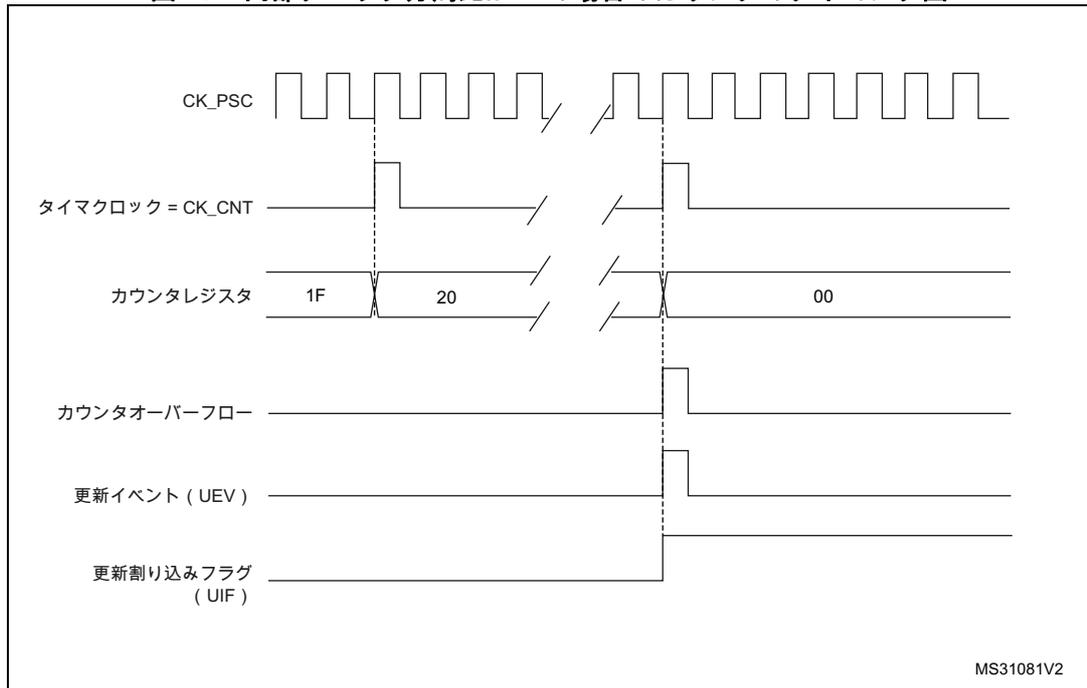


図 198. ARPE=0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

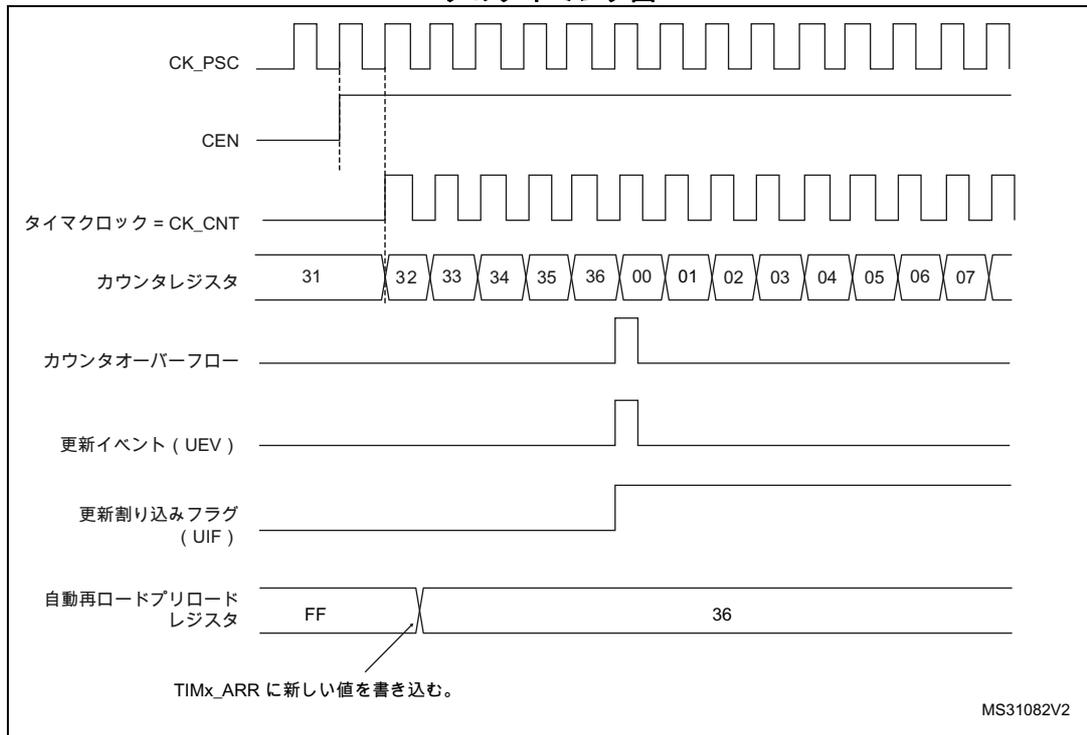
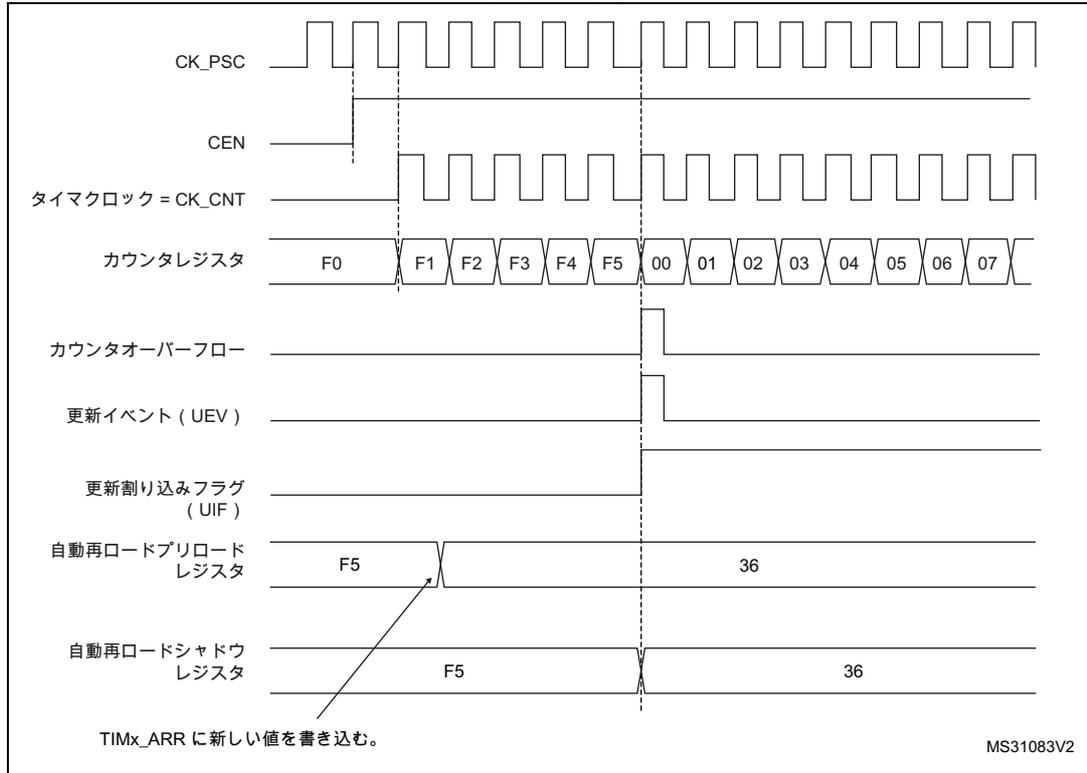


図 199. ARPE=1 (TIMx\_ARR はプリロードされる) のときの更新イベント時のカウンタのタイミング図



## ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx\_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

更新イベントは、カウンタアンダーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることにより生成できます。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 200. 内部クロック分周比が 1 の場合のカウンタのタイミング図

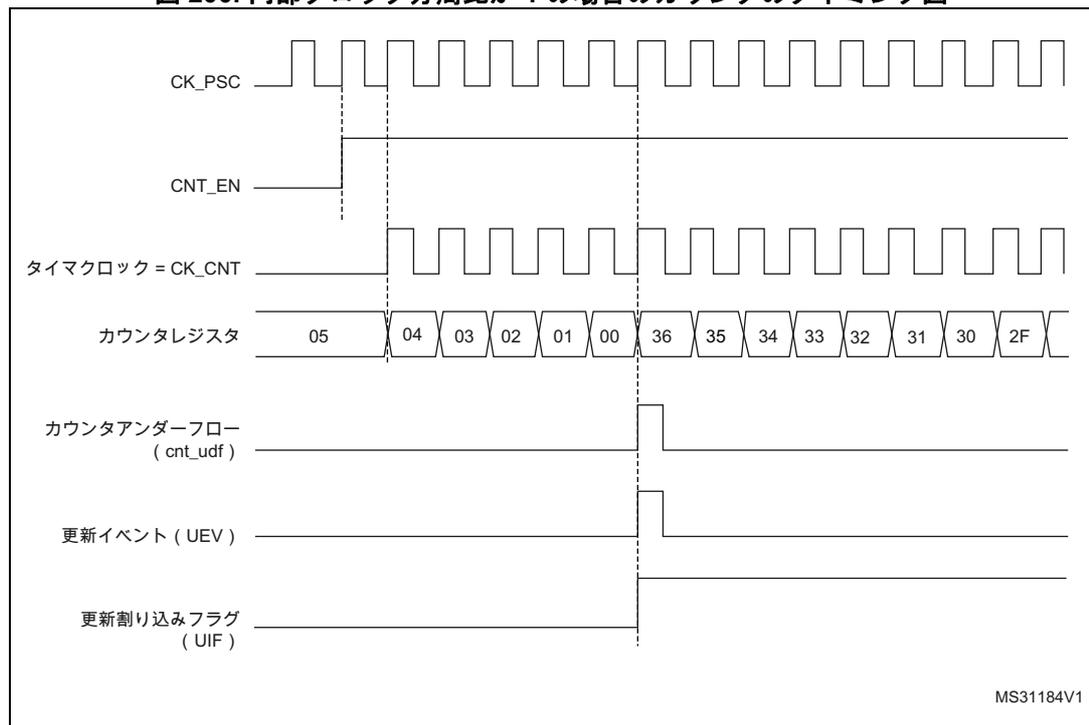


図 201. 内部クロック分周比が 2 の場合のカウンタのタイミング図

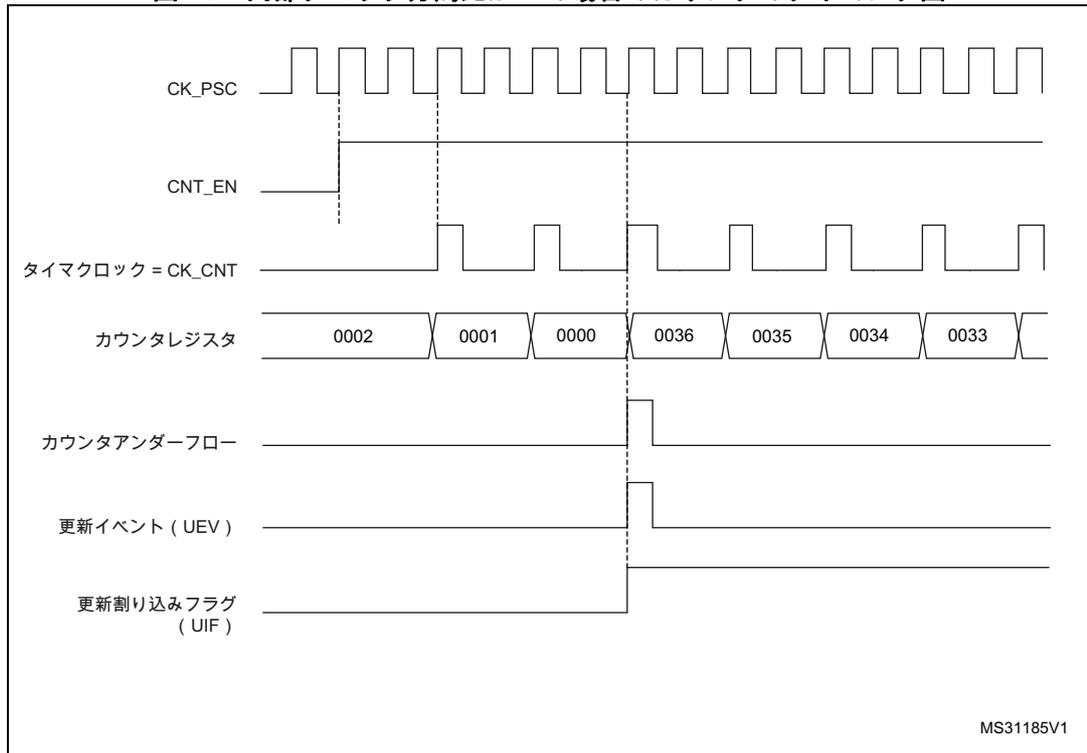


図 202. 内部クロック分周比が 4 の場合のカウンタのタイミング図

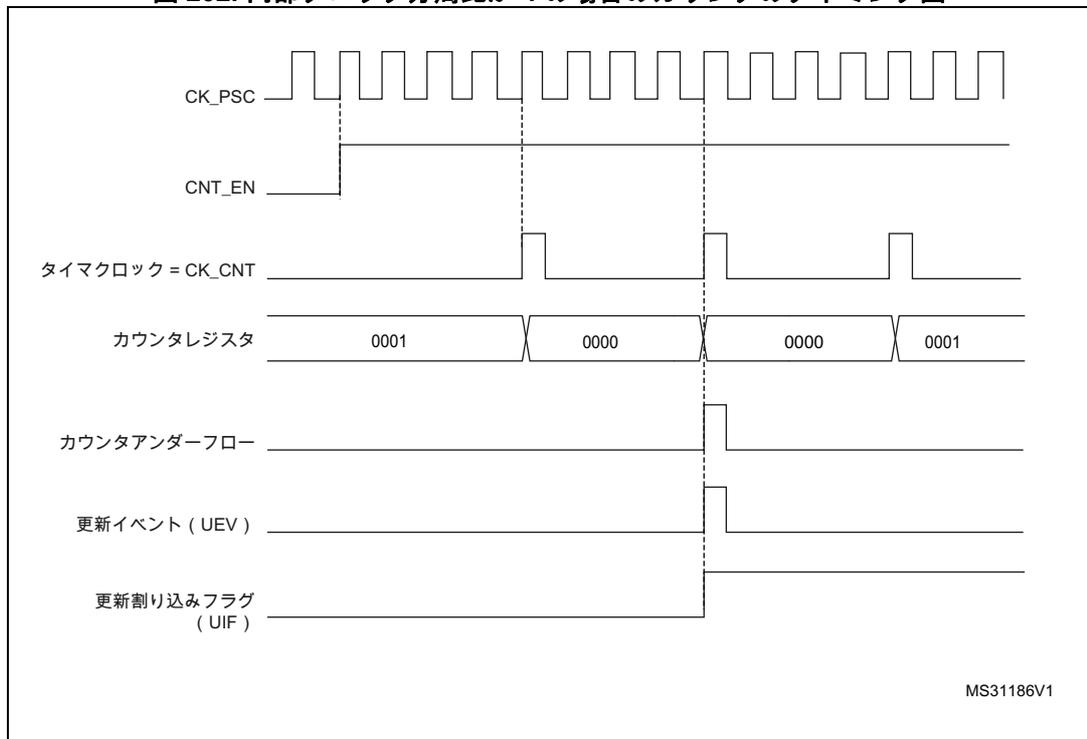


図 203. 内部クロック分周比が N の場合のカウンタのタイミング図

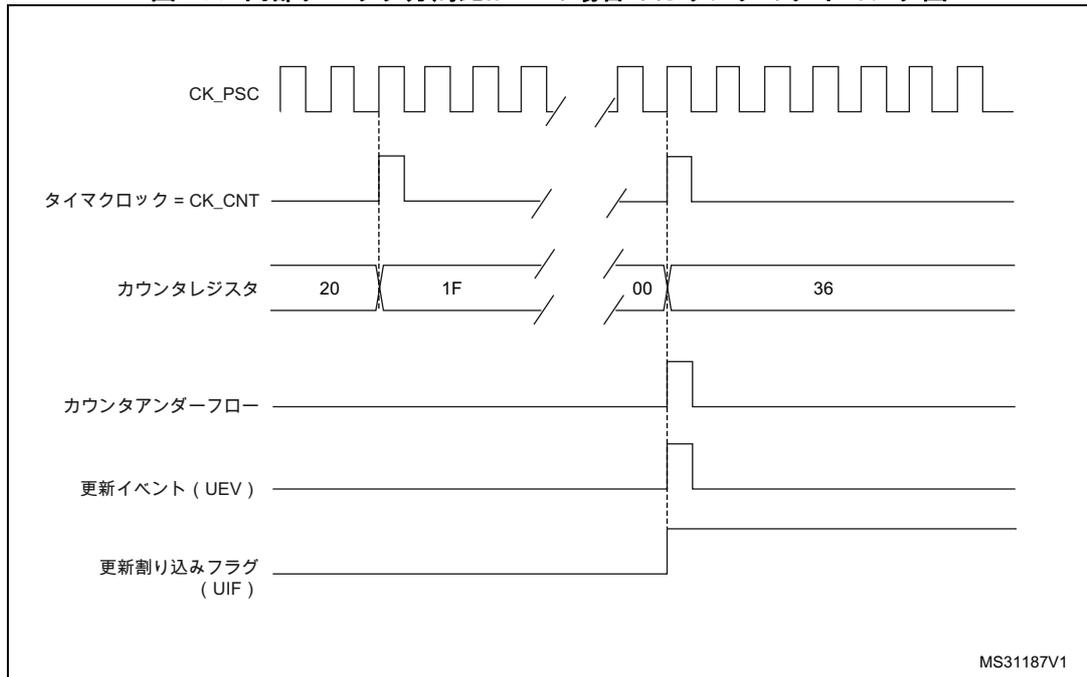
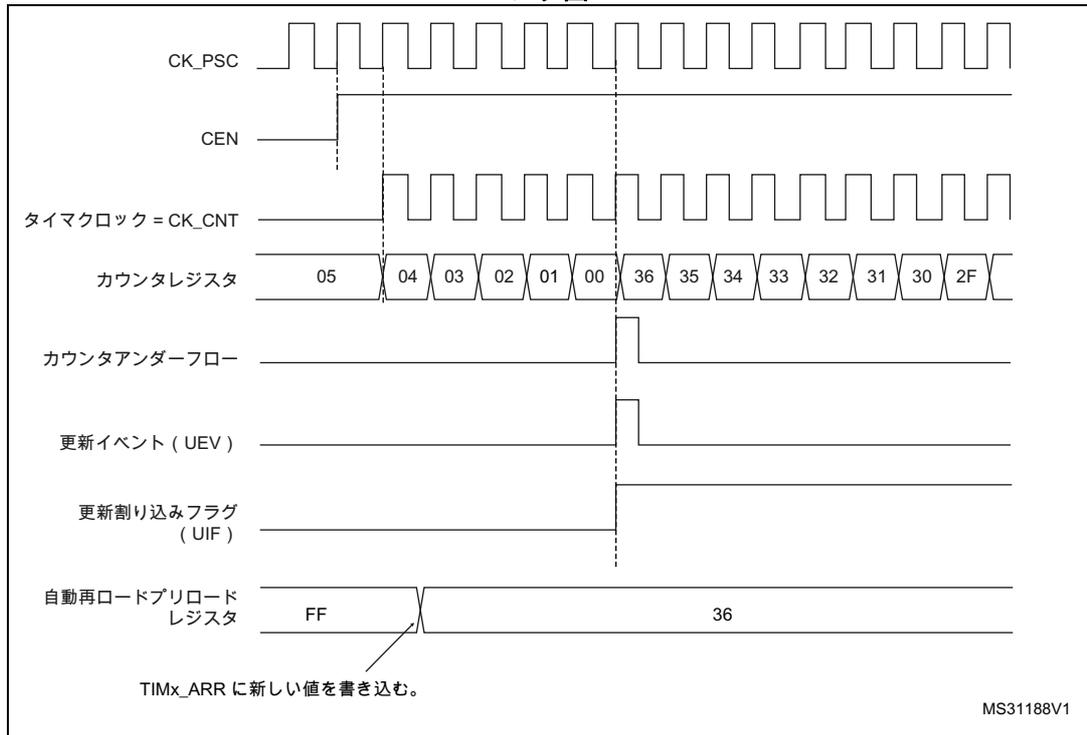


図 204. 繰り返しカウンタが使用されていないときの更新イベント時のカウンタのタイミング図



## センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センターアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS=11) にセットされます。

このモードでは、方向ビット (TIMx\_CR1 レジスタの DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレープモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

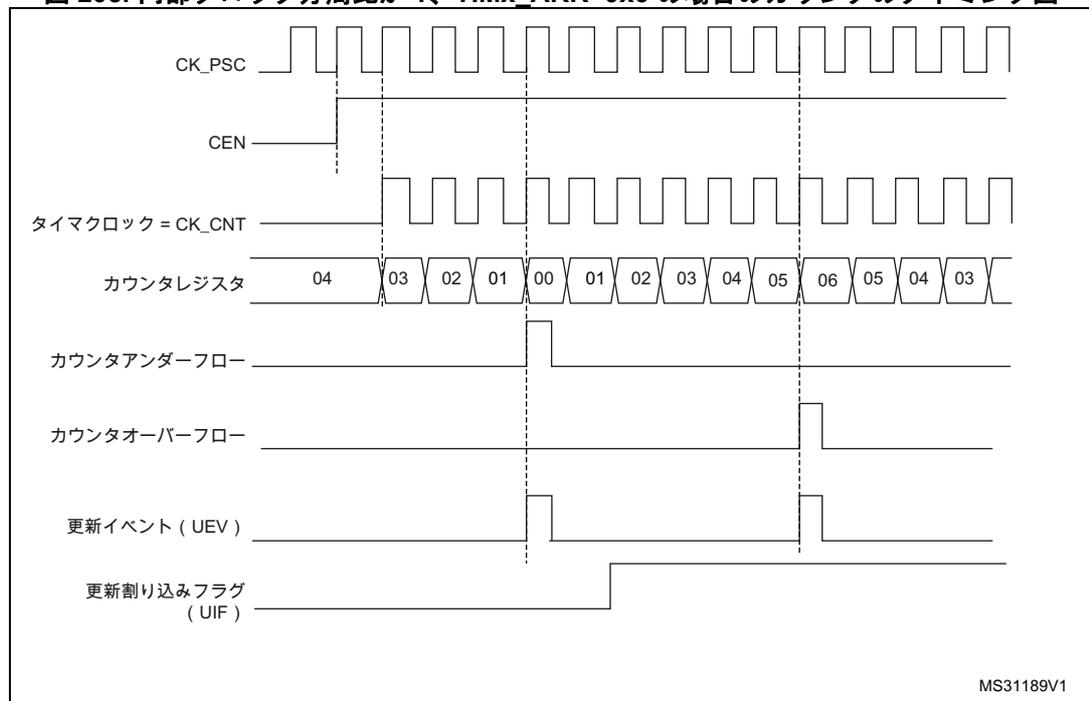
さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 205. 内部クロック分周比が 1、TIMx\_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 23.4.1: TIMx 制御レジスタ 1 \(TIMx\\_CR1\)](#) (751 ページ) を参照)。

図 206. 内部クロック分周比が 2 の場合のカウンタのタイミング図

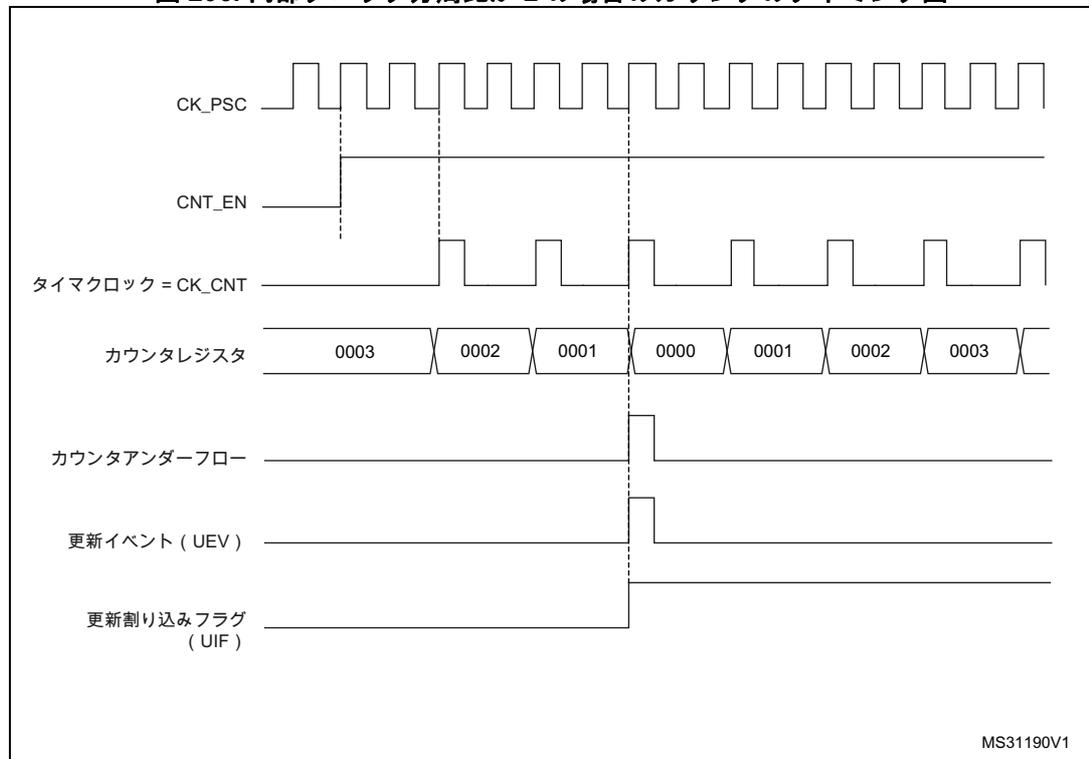
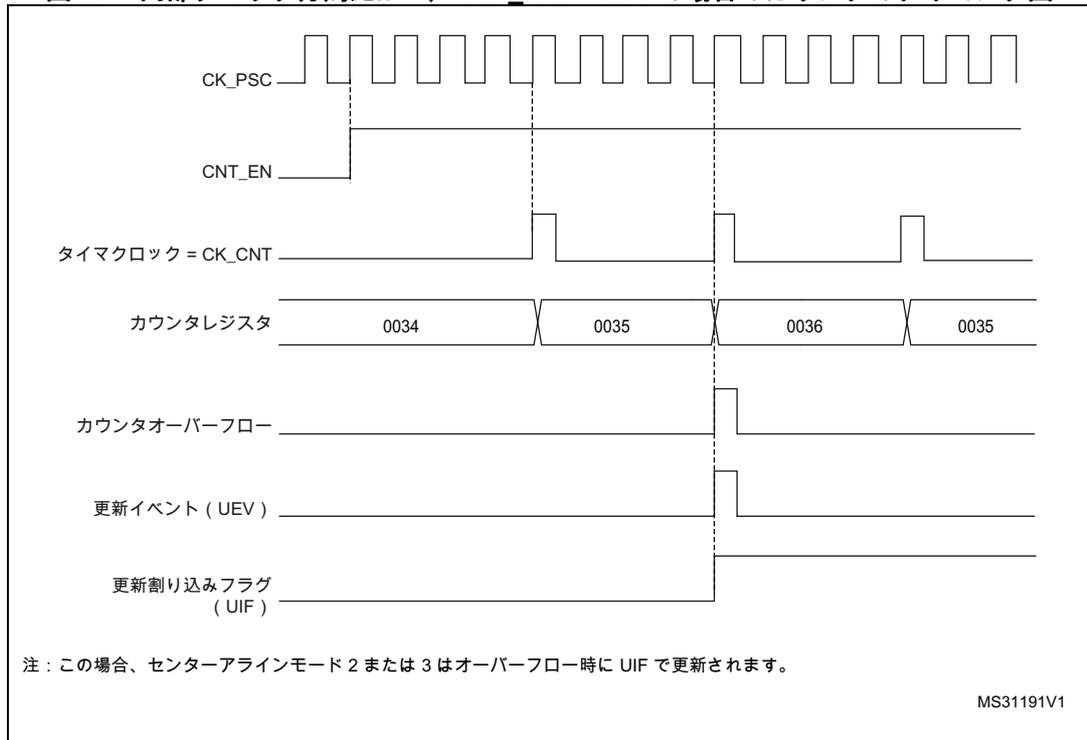


図 207. 内部クロック分周比が 4、TIMx\_ARR=0x36 の場合のカウンタのタイミング図



1. センターアラインモード 2 または 3 が使用され、オーバーフロー時に UIF がセットされます。

図 208. 内部クロック分周比が N の場合のカウンタのタイミング図

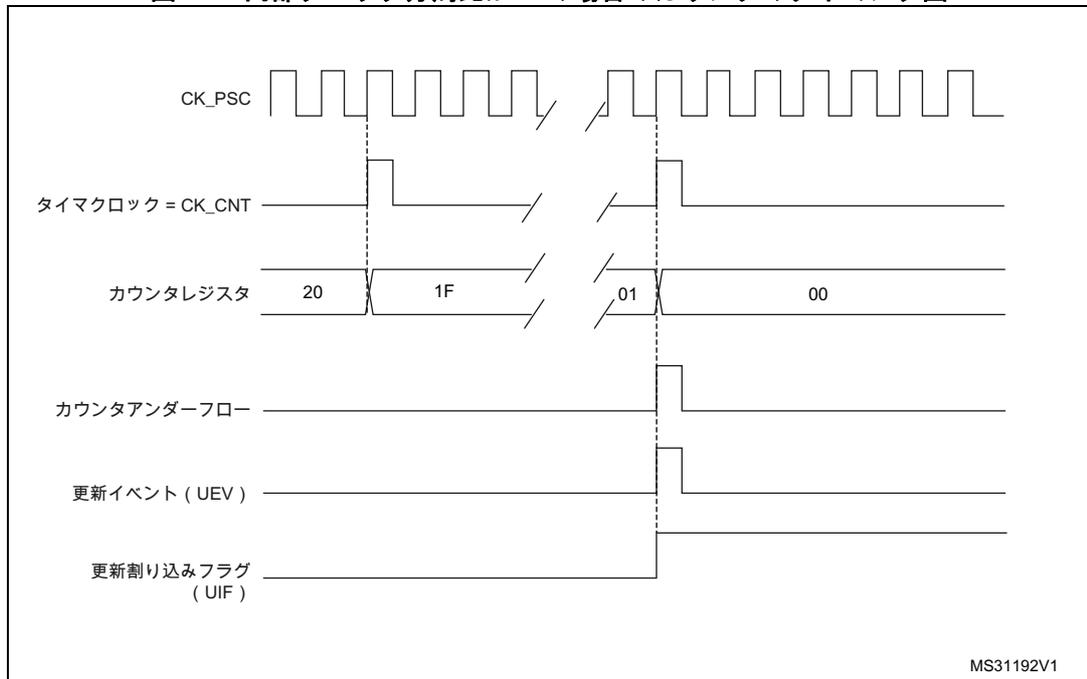


図 209. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図

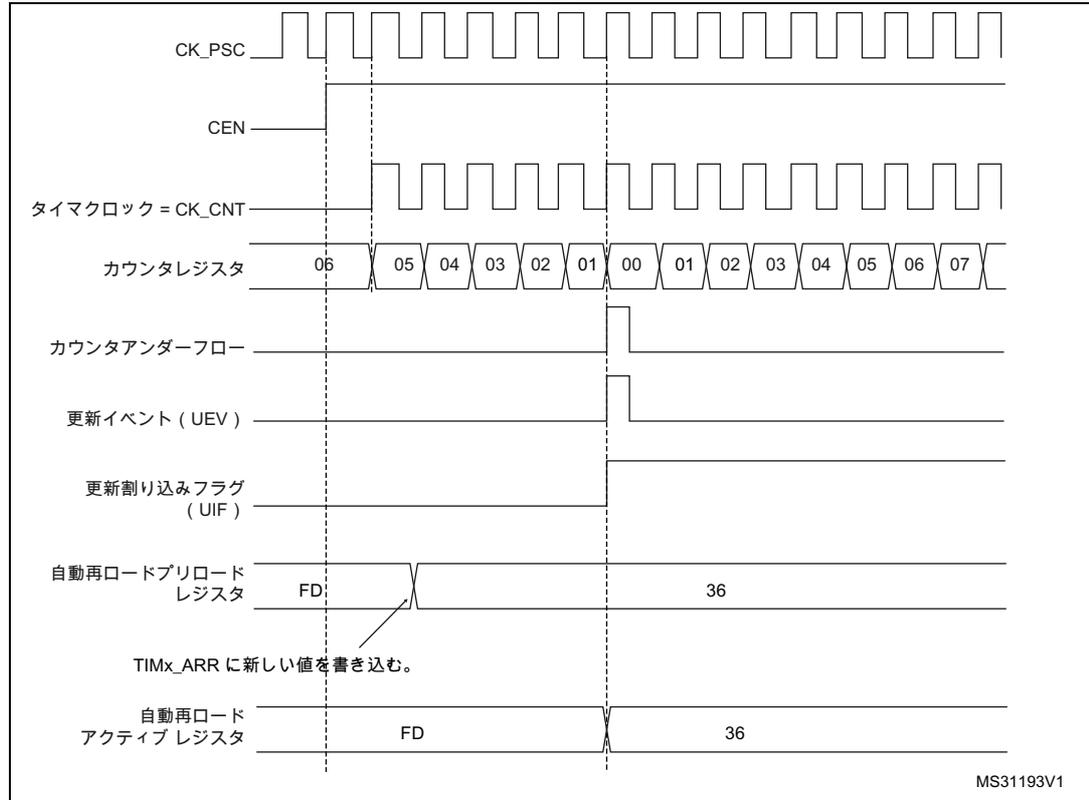
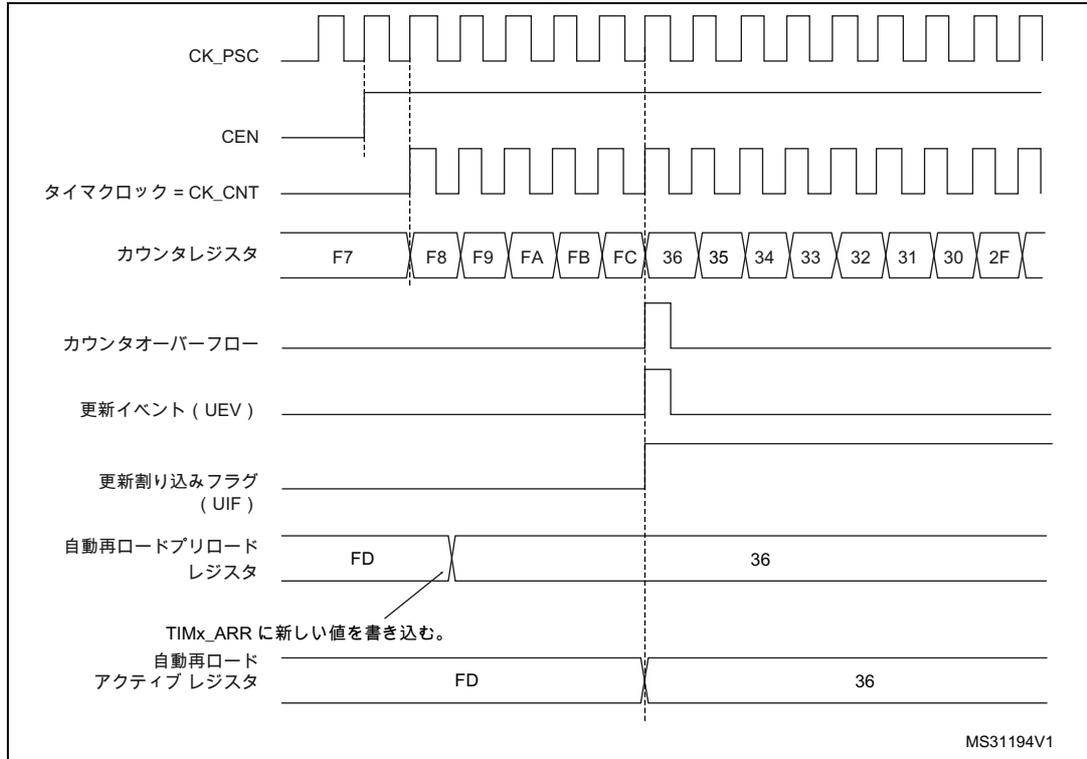


図 210. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



### 23.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

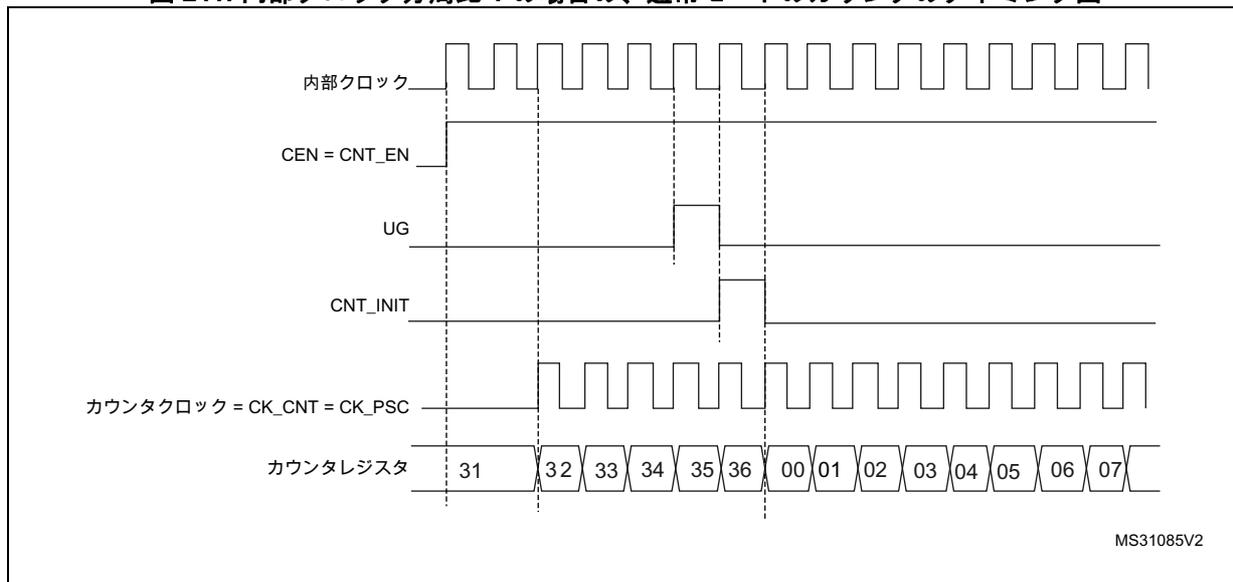
- 内部クロック (CK\_INT)
- 外部クロックモード 1 : 外部入力ピン (Tix)
- 外部クロックモード 2 : 外部トリガ入力 (ETR)
- 内部トリガ入力 (ITRx) : あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 13 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する \(745 ページ\)](#) を参照してください。

#### 内部クロックソース (CK\_INT)

スレーブモードコントローラが無効の場合 (TIMx\_SMCR レジスタの SMS=000)、CEN、DIR ビット (TIMx\_CR1 レジスタ) と UG ビット (TIMx\_EGR レジスタ) が実際の制御ビットであり、ソフトウェアでのみ変更できます (自動的にクリアされたままの UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK\_INT が供給されます。

図 211 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

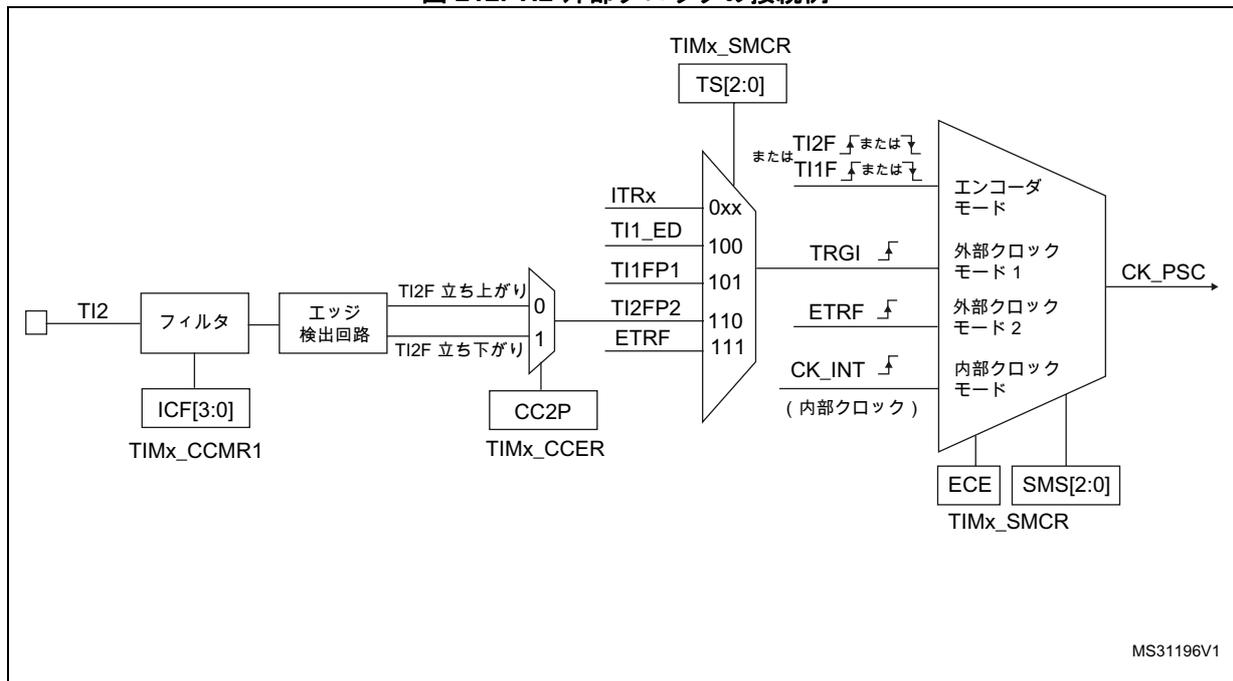
図 211. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



## 外部クロックソースモード 1

このモードは TIMx\_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 212. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx\_CCMR1レジスタに CC2S=01 を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx\_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。

注：

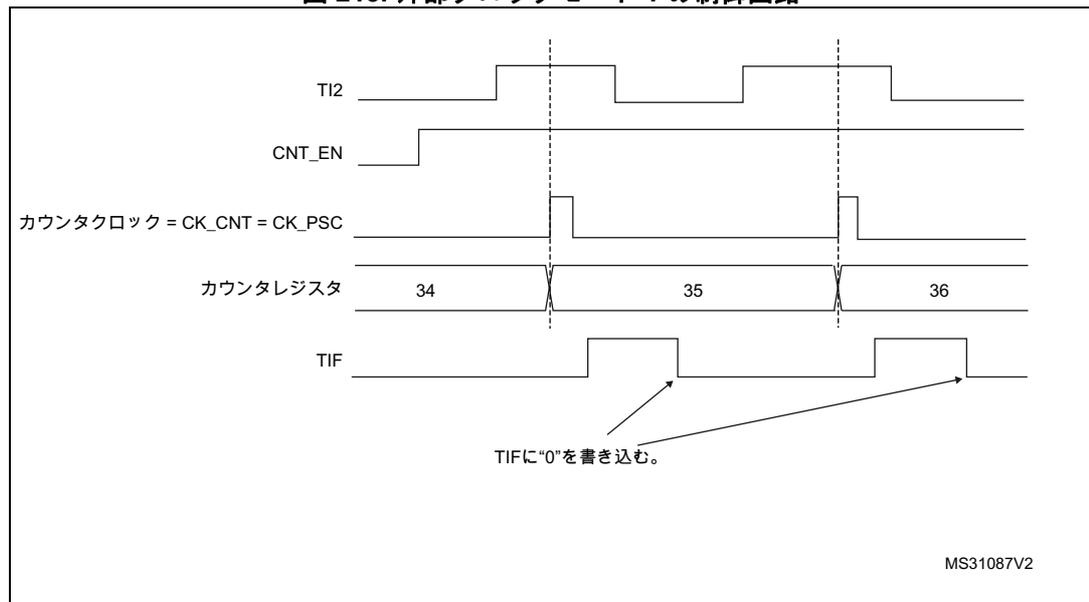
**キャプチャプリスケアラはトリガには使用されないなので、設定は不要です。**

3. CC2P=0、CC2NP=0、および CC2P=0 を TIMx\_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx\_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。
6. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウントクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 213. 外部クロックモード 1 の制御回路



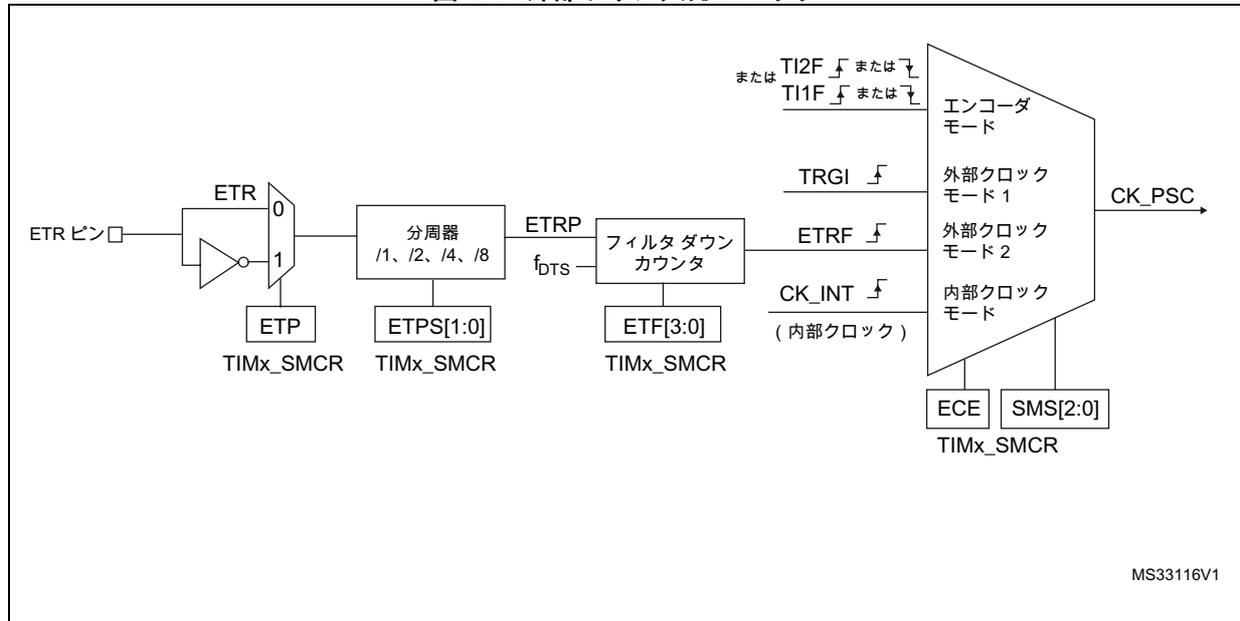
## 外部クロックソースモード 2

このモードは、TIMx\_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 214 に、外部トリガ入力ブロックの概要を示します。

図 214. 外部トリガ入力ブロック



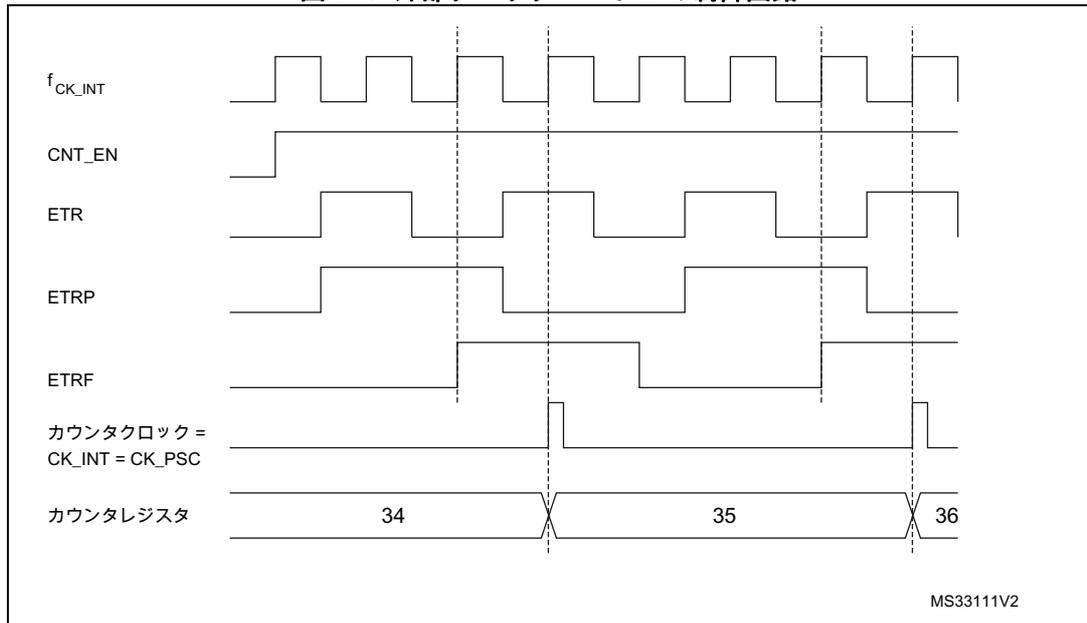
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx\_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx\_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx\_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx\_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 215. 外部クロックモード 2 の制御回路



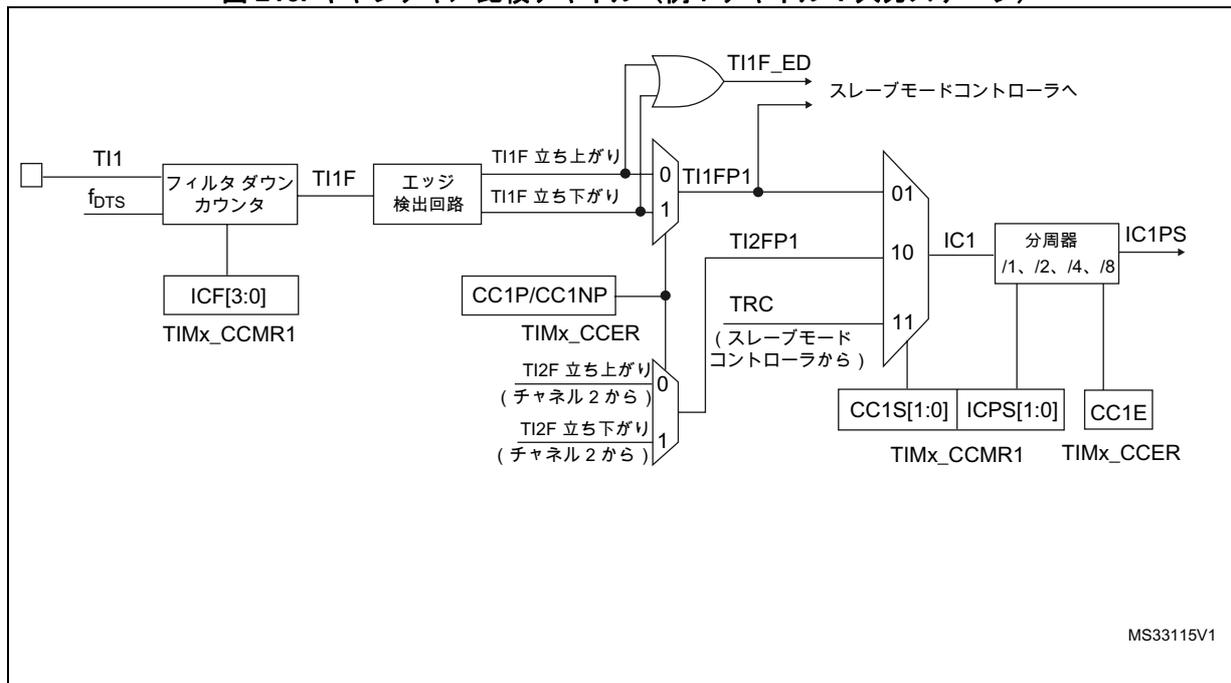
### 23.3.4 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケラ）、および出力カステージ（比較回路と出力制御）から構成されています。

次の図に、キャプチャ／比較チャネルの概要を示します。

入カステージは、対応する Tl<sub>x</sub> 入力をサンプリングして、フィルタリングを行った Tl<sub>x</sub>F を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (Tl<sub>x</sub>FP<sub>x</sub>) を生成します。この信号はプリスケラを通じて、キャプチャレジスタ (IC<sub>x</sub>PS) に渡されます。

図 216. キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)



出力ステージは、OCxRef (アクティブハイ) として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 217. キャプチャ/比較チャンネル 1 メイン回路

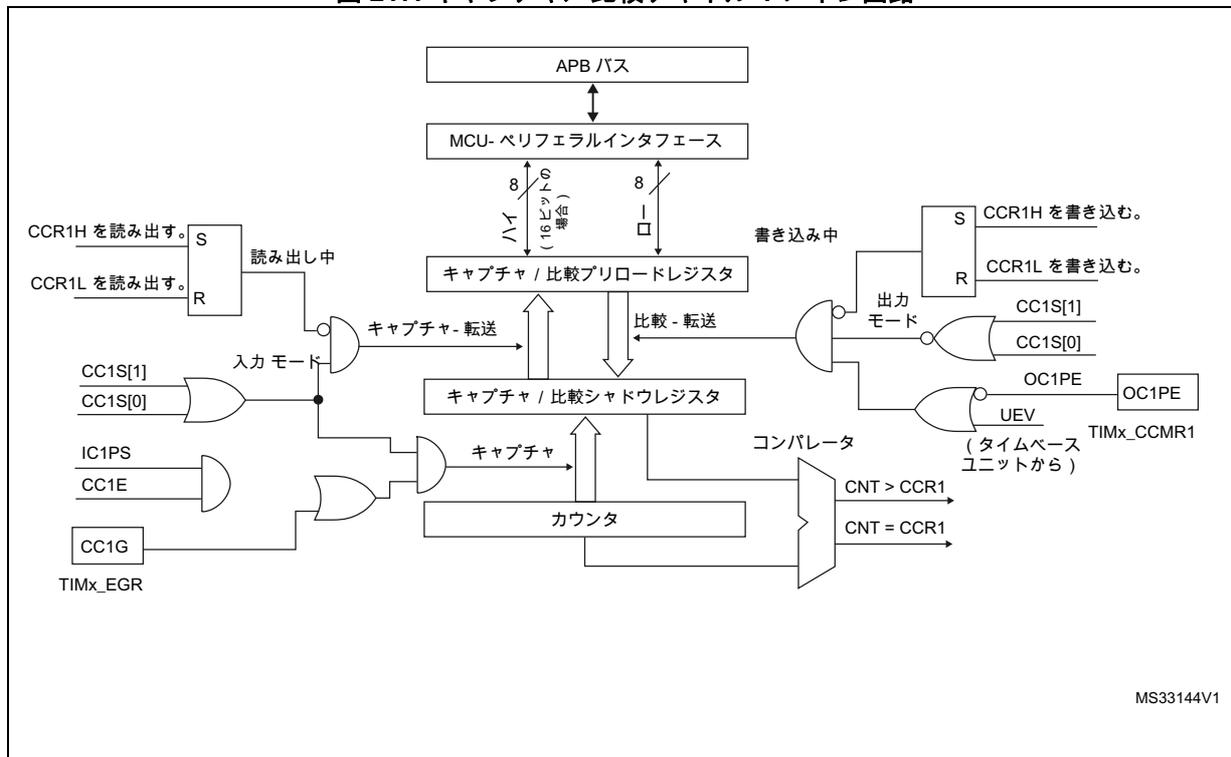
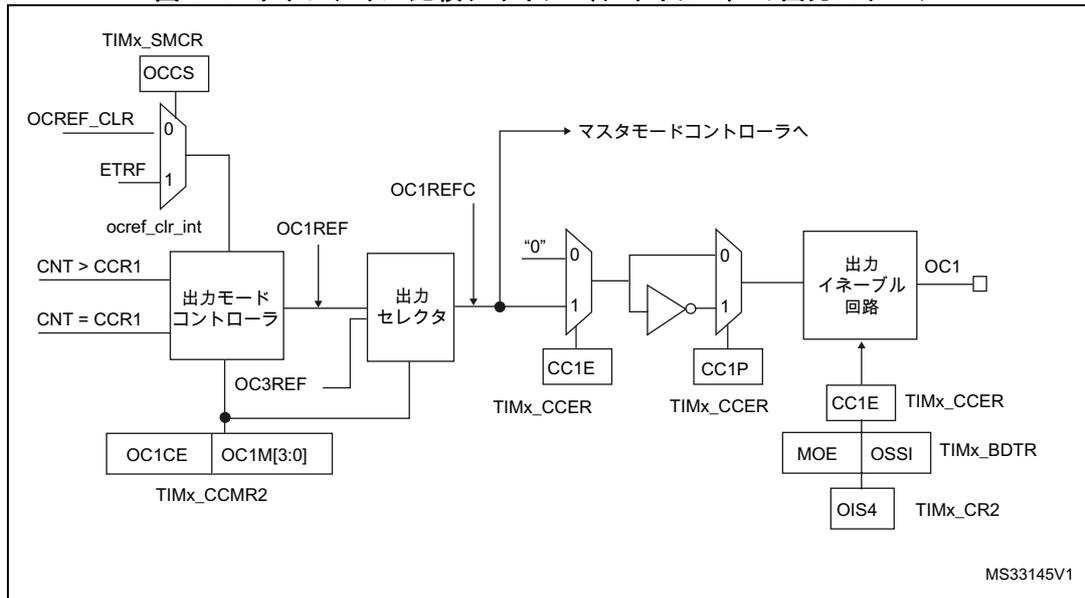


図 218. キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ



キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

### 23.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx\_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx\_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx\_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx\_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx\_CCR1 は TI1 入力とリンクされていなければならないので、このためには TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx\_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数  $f_{DTS}$  でサンプリング)。この場合、TIMx\_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx\_CCER レジスタの CC1P、CC1NP、および CC1NP ビットに“000”を書き込みます (この場合、立ち上がりエッジの選択)。



4. 入力プリスケアラをプログラムします。今回の例では、有効な信号変化ごとにキャプチャを行いたいので、プリスケアラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに 00 を書き込みます)。
5. TIMx\_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
6. 必要な場合は、TIMx\_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx\_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

**注：** IC 割り込みと DMA リクエストは、TIMx\_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

## 23.3.6 PWM 入力モード

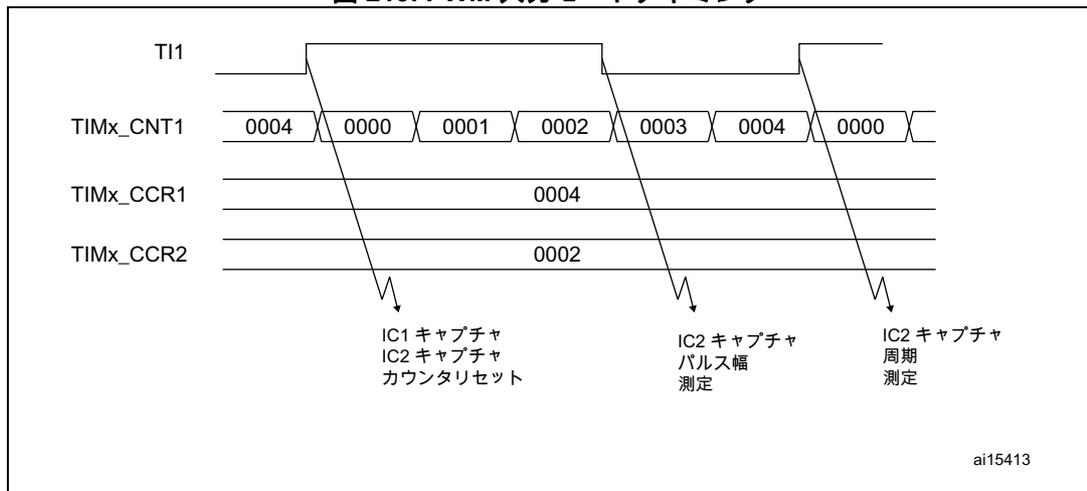
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2 つの ICx 信号が同じ Tlx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます (手順は、CK\_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

1. TIMx\_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx\_CCR1 のアクティブ入力を選択します。
2. CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブ極性を選択します (TIMx\_CCR1 のキャプチャとカウンタクリアの両方に使用します)。
3. TIMx\_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
4. CC2P ビットに“1”を、CC2NP ビットに“0”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx\_CCR2 のキャプチャに使用されます)。
5. TIMx\_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
6. TIMx\_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
7. TIMx\_CCER レジスタの CC1E と CC2E ビットに 1 を書き込むことによって、キャプチャを有効にします。

図 219. PWM 入力モードタイミング



1. TI1FP1 と TI2FP2 のみがスレーブモードコントローラに接続されているので、PWM 入力モードは TIMx\_CH1/TIMx\_CH2 信号でのみ使用できます。

## 23.3.7 強制出力モード

このモード (TIMx\_CCMRx レジスタの CCxS=00) では、各出力比較信号 (OCxREF、そして OCx) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx\_OCMRx レジスタの OCxM ビットに 101 を書き込みます。これにより、OCxREF は強制的にハイレベルになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例 : CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

TIMx\_CCMRx レジスタの OCxM ビットに 100 を書き込むことによって、OCxREF 信号を強制的にローにできます。

いずれにしても、TIMx\_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、出力比較モードのセクションで説明します。

## 23.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx\_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx\_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx\_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx\_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信しません (DMA リクエスト選択には、TIMx\_CR2 レジスタの CCDS ビットが使用されます)。

TIMx\_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx\_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

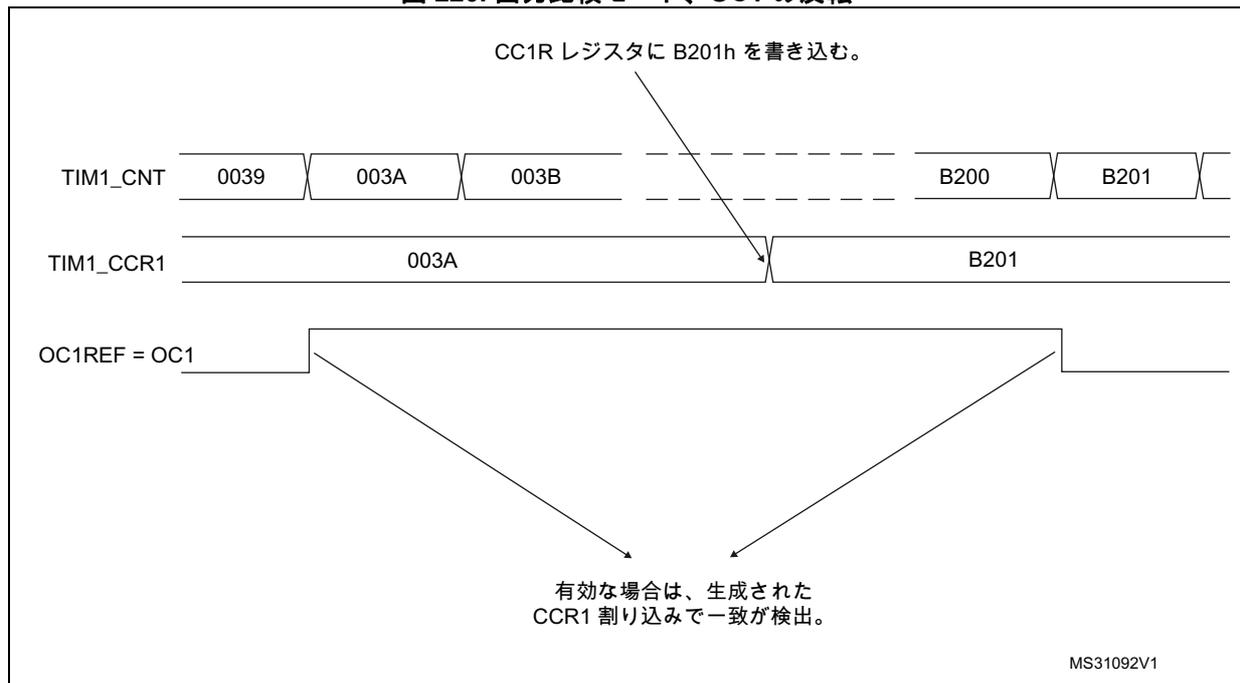
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

### 手順

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx\_ARR レジスタと TIMx\_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエスト/DMA リクエストを生成する場合は、CCxIE ビット/CCxDE ビットをセットします。
4. 出力モードを選択します。たとえば、CNT が CCRx と一致したときに OCx 出力をトグルし、CCRx プリロードを使用せず、OCx が有効でアクティブハイのときには、OCxM=011、OCxPE=0、CCxP=0、CCxE=1 を書き込みます。
5. TIMx\_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアで TIMx\_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限ります (OCxPE=0)。そうでない場合、TIMx\_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 220](#) に示します。

図 220. 出力比較モード、OC1 の反転



## 23.3.9 PWM モード

パルス幅変調(PWM)モードでは、TIMx\_ARR レジスタの値によって決められた周波数と TIMx\_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、個々のチャンネル (OCx 出力ごとに PWM 1 波形) で、TIMx\_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) や“111” (PWM モード 2) を書き込むことで、独自に選択できません。TIMx\_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx\_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx\_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx\_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx\_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx\_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx\_CNT と TIMx\_CCRx が常に比較されて、TIMx\_CCRx ≤ TIMx\_CNT または TIMx\_CNT ≤ TIMx\_CCRx かどうか判断されます (カウントの方向によります)。ただし、OCREF\_CLR 機能 (OCREF は、次の PWM 周期までは ETR 信号を通じて外部イベントによってクリアできる) に従って、OCREF 信号は次の場合にのみアサートされます。

- 比較結果が変化したとき、または
- 出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) が停止構成 (比較なし、OCxM=000) から PWM モードの 1 つ (OCxM=110 または 111) へ切り替えられたとき。

タイマの動作中は、ソフトウェアで強制的に PWM になります。

タイマは、TIMx\_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

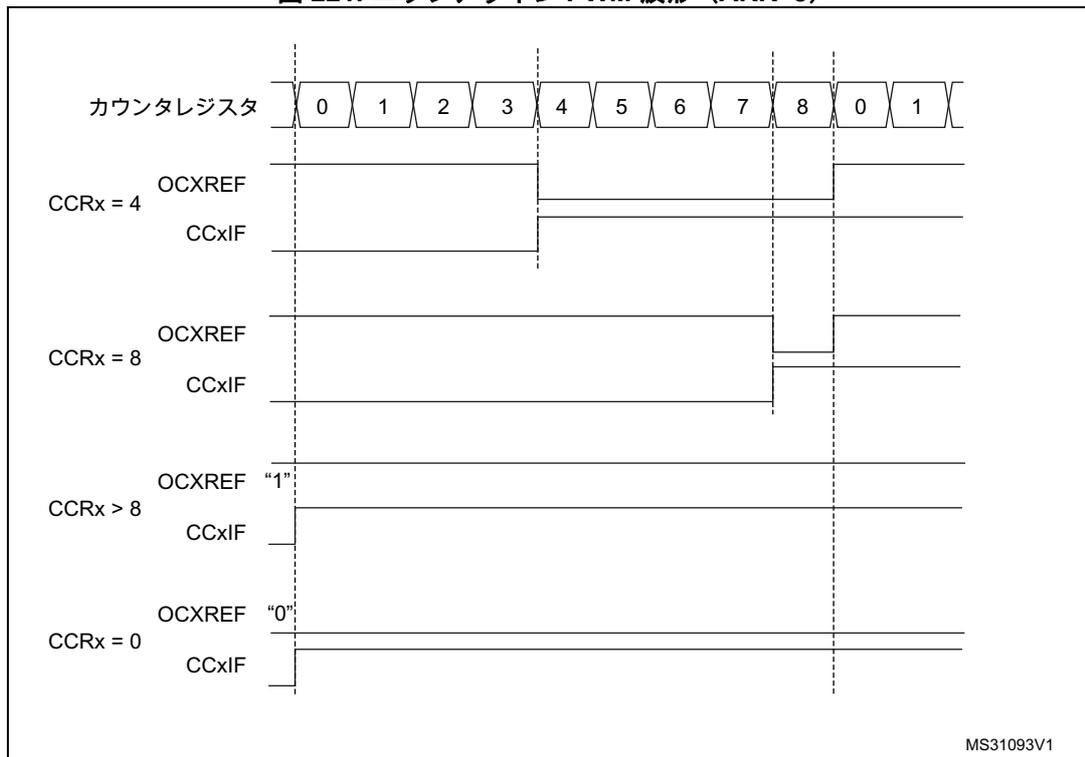
### PWM エッジアラインモード

アップカウント構成

TIMx\_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード \(710 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx\_CNT < TIMx\_CCRx の間はハイに、そうでない場合はローになります。TIMx\_CCRx の比較値が自動再ロード値 (TIMx\_ARR レジスタの) より大きい場合、OCxREF は“1”です。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 221](#) に TIMx\_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 221. エッジライン PWM 波形 (ARR=8)



## ダウンカウント構成

TIMx\_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード \(713 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx\_CNT > TIMx\_CCRx の間はローであり、そうでない場合はハイになります。TIMx\_CCRx の比較値が TIMx\_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0% の PWM 信号を生成することはできません。

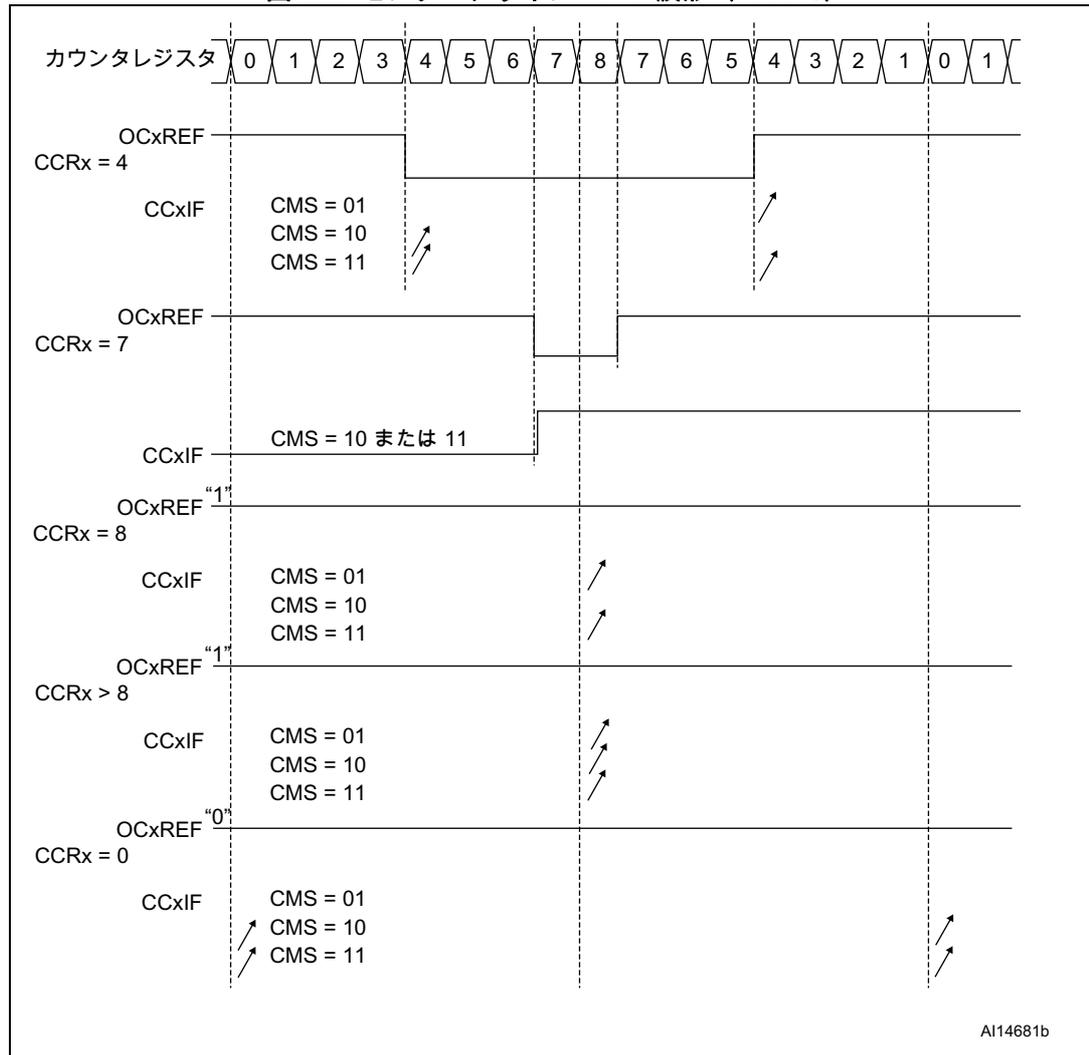
## PWM センターアラインモード

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”でないときにアクティブです (その他すべての設定は、OCxRef/OCx 信号に対して同じ効果を持ちます)。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx\_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターアラインモード \(アップ/ダウンカウント\) \(717 ページ\)](#) を参照してください。

図 222 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx\_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx\_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 222. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント :

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx\_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
  - 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx\_CNT > TIMx\_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
  - カウンタに 0 または TIMx\_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx\_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

## 23.3.10 非対称 PWM モード

非対称モードでは、プログラム可能な位相シフトによって 2 つのセンターアライン PWM 信号の生成を可能にします。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや位相シフトは TIMx\_CCRx レジスタペアで決定されます。1 つ目のレジスタがアップカウント時の PWM を制御し、2 つ目のレジスタがダウンカウント時の PWM を制御することで、PWM は PWM ハーフサイクルごとに調整されます。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

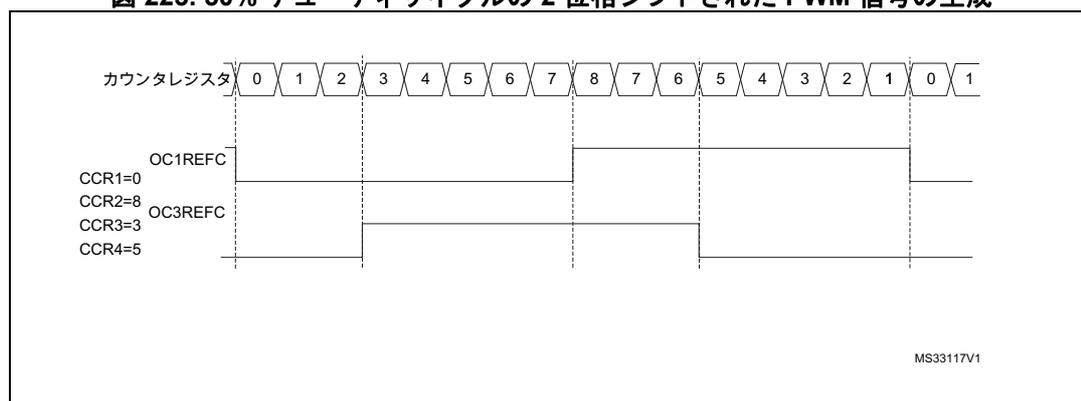
非対称 PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1110” (非対称 PWM モード 1) または“1111” (非対称 PWM モード 2) を書き込むことによって、2 チャネルごとに選択できます (CCR レジスタペアごとに 1 つの OCx 出力)。

**注：** OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

特定のチャンネルが非対称の PWM チャンネルとして使用されると、その 2 次チャンネルも使用できます。たとえば、OC1REFC 信号がチャンネル 1 (非対称 PWM モード 1) に生成されると、チャンネル 2 の OC2REF 信号、または非対称 PWM モード 2 の結果として得られる OC2REFC 信号を出力できます。

図 223 は、非対称 PWM モードを使用して生成される信号の例を表します (チャンネル 1 から 4 は非対称 PWM モード 1 として設定されます)。

図 223. 50% デューティサイクルの 2 位相シフトされた PWM 信号の生成



## 23.3.11 組み合わせ PWM モード

組み合わせ PWM モードでは、2 つのエッジアラインまたはセンターアライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2 つの TIMx\_CCRx レジスタで決定されます。結果として得られる信号 OCxREFC は、2 つの PWM 基準信号の OR または AND による論理結合から成ります。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

組み合わせ PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1100” (組み合わせ PWM モード 1) または“1101” (組み合わせ PWM モード 2) を書き込むことによって、2 チャネルごとに選択できます (CCR レジスタペアごとに 1 つの OCx 出力)。

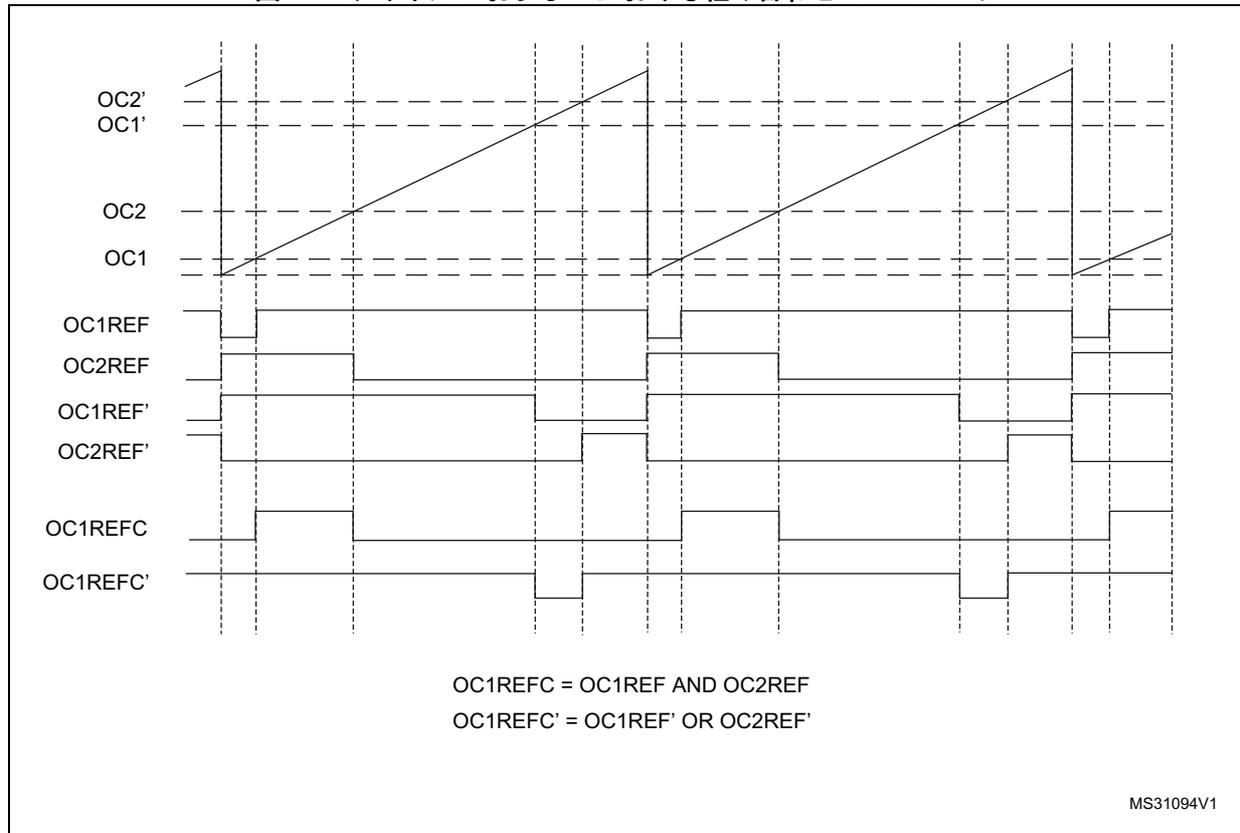
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、2 次チャンネルを反対の PWM モードに設定する必要があります (たとえば、1 つを組み合わせ PWM モード 1、もう 1 つを組み合わせ PWM モード 2 にします)。

**注：** OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

図 224 は、次の設定で取得可能な非対称 PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合
- チャンネル 3 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 4 が PWM モード 1 で設定されている場合

図 224. チャンネル 1 および 3 における組み合わせ PWM モード



## 23.3.12 外部イベントによる OCxREF 信号のクリア

特定のチャネルの OCxREF 信号は OCREF\_CLR\_INPUT にハイレベルを適用するとクリアされます (対応する TIMx\_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする)。OCxREF は、次の更新イベント (UEV) が発生するまで、ローレベルを保ちます。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

OCREF\_CLR\_INPUT は、TIMx\_SMCR レジスタで OCCS ビットを設定することで、OCREF\_CLR 入力と ETRF (フィルタ後の ETR) の間で選択できます。

ETRF が選択された場合、ETRF は次のように設定する必要があります。

特定のチャネルの OCxREF 信号は ETRF 入力をハイレベルとする (対応する TIMx\_CCMRx レジスタの OCxCE イネーブルビットを 1 にセットする) ことでリセットすることができます。OCxREF は、次の更新イベント (UEV) が発生するまで、ローレベルを保ちます。

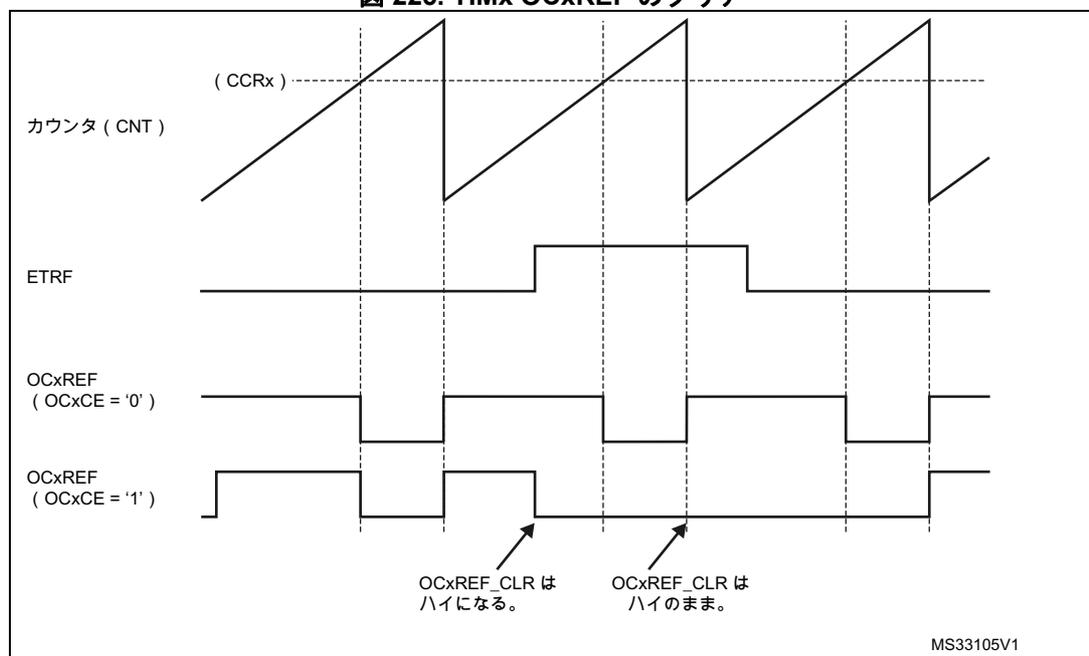
この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

たとえば、OCxREF 信号を比較回路の出力に接続して、電流処理に使用することができます。この場合、ETRF は次のように設定する必要があります。

1. 外部トリガプリスケアラをオフに保つ必要があります。すなわち、TIMx\_SMCR レジスタのビット ETPS[1:0] が 00 にクリアされます。
2. 外部クロックモード 2 を無効にする必要があります。すなわち、TIM1\_SMCR レジスタのビット ECE が 0 にクリアされます。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて設定できます。

図 225 に、OCxCE イネーブルビットの両方の値について、ETRF 入力がハイになったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 225. TIMx OCxREF のクリア



注: 100% デューティサイクルの PWM の場合 (CCR<sub>x</sub>>ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

## 23.3.13 ワンパルスモード

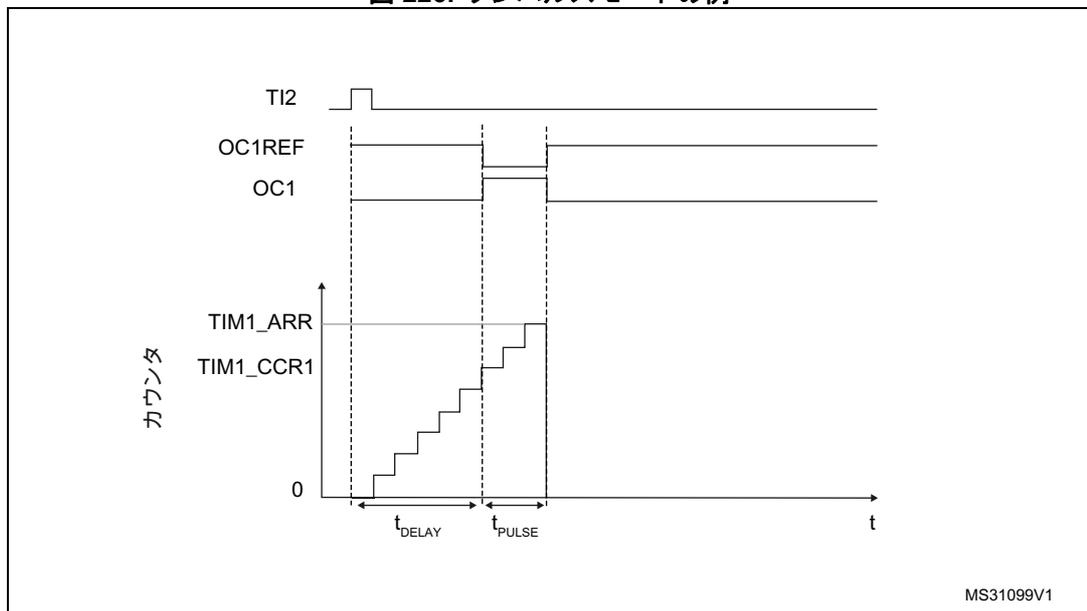
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- $CNT < CCRx \leq ARR$  (特に、 $0 < CCRx$ )

図 226. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が  $t_{PULSE}$  の正のパルスを遅延時間  $t_{DELAY}$  後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx\_CCMR1 レジスタに IC2S=01 を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx\_CCER レジスタで CC2P=0 と CC2NP=0 を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として構成するために、TIMx\_SMCR レジスタの TS="110" を書き込みます。
- TI2FP2 を使用してカウンタを開始するために、TIMx\_SMCR レジスタの SMS ビットに "110" (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- $t_{\text{DELAY}}$  は、TIMx\_CCR1 レジスタに書き込まれた値によって定義されます。
- $t_{\text{PULSE}}$  は、自動再ロード値と比較値の差 (TIMx\_ARR - TIMx\_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx\_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。オプションで、TIMx\_CCMR1 レジスタの OC1PE=1 と TIMx\_CR1 レジスタの ARPE=1 を書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx\_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので (シングルモード)、TIMx\_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx\_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

### 特殊なケース : OCx 高速イネーブル :

ワンパルスモードでは、TIx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ( $t_{\text{DELAY min}}$ ) が制限されます。

最小遅延で波形を出力したい場合は、TIMx\_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較を考慮せずにトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

## 23.3.14 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx\_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS=“001”を、TI1 エッジのみをカウントしている場合は SMS=“010”を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS=“011”を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx\_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。CC1NP と CC2NP はクリア状態に維持する必要があります。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。[表 129](#) を参照してください。カウンタのクロックは、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択後は TI1 および TI2。フィルタされず、反転されない場合は TI1FP1=TI1、フィルタされず、反転されない場合は TI2FP2=TI2) の有効な変化によって駆動されます。ただし、カウンタ有効なことが前提となります (TIMx\_CR1 レジスタの CEN ビットが“1”)。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx\_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx\_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx\_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、およびトリガ出力機能は、通常動作を続けます。

このモードでは、カウンタは直交エンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (TI1 と TI2 は同時に切り替わらないと想定しています)。

表 129. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	ハイ	ダウン	アップ	カウントなし	カウントなし
	ロー	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	ハイ	カウントなし	カウントなし	アップ	ダウン
	ロー	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の両方を カウント	ハイ	ダウン	アップ	アップ	ダウン
	ロー	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 227 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入カジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S= 01 (TIMx\_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S= 01 (TIMx\_CCMR2 レジスタ、TI2FP2 は TI2 に配置)
- CC1P = 0、CC1NP = 0、IC1F = 0000 (TIMx\_CCER レジスタ、TI1FP1 非反転、TI1FP1 = TI1)
- CC2P = 0、CC2NP = 0、IC2F = 0000 (TIMx\_CCER レジスタ、TI2FP2 非反転、TI2FP2 = TI2)
- SMS=011 (TIMx\_SMCR レジスタ、両方の入力が立ち上がり立ち下がり両エッジでアクティブ)
- CEN=1 (TIMx\_CR1 レジスタ、カウンタ有効)

図 227. エンコーダインタフェースモードにおけるカウンタの動作例

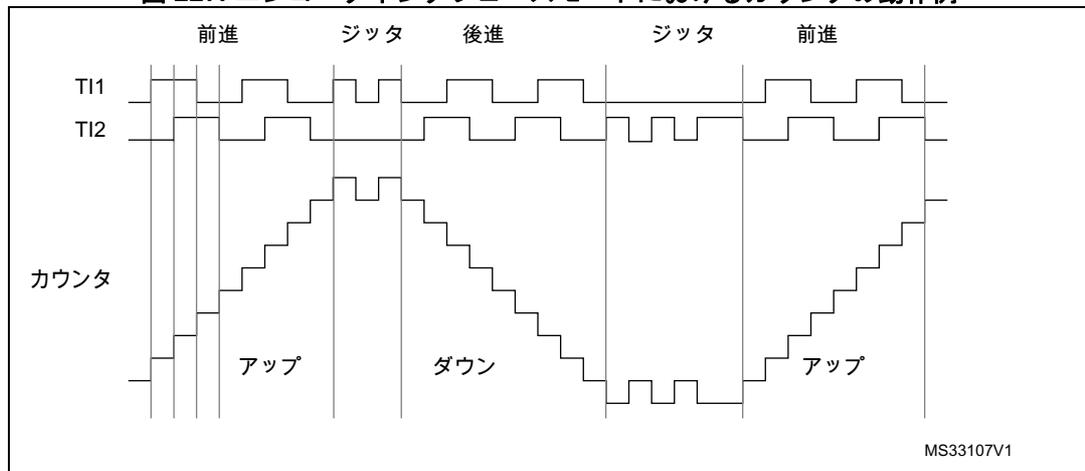
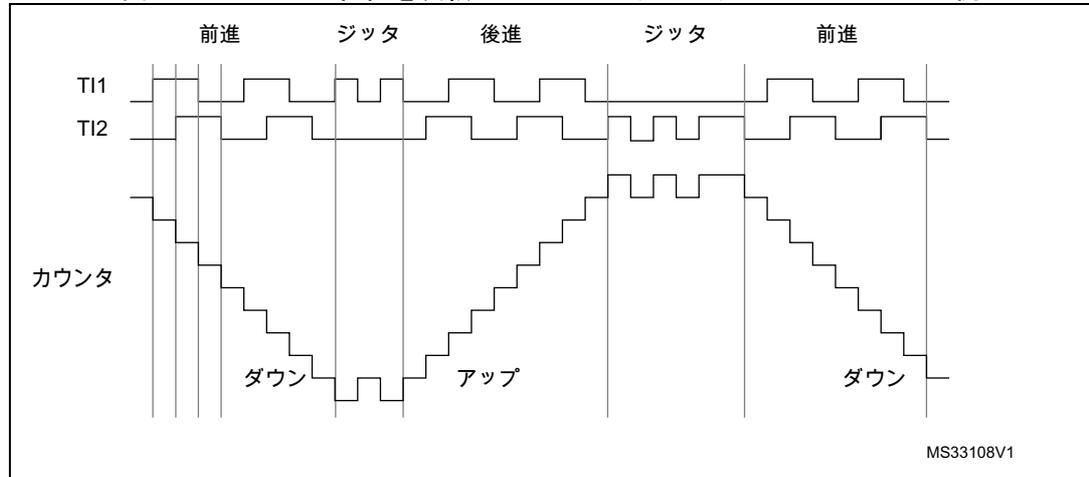


図 228 に、TI1FP1 の極性を反転したときのカウンタの動作を示します (上記と同じ設定ですが、CC1P=1)。

図 228. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします (キャプチャ信号は周期的でなければならず、別のタイマによって生成できます)。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

### 23.3.15 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー状態を分割できないものとして読み取ることができます。バックグラウンドタスク (カウンタの読み出し) と中断 (更新の中断) との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読み出しアクセス時に UIFCPY フラグによって上書きされます (カウンタの最上位ビットには書き込みモード時のみアクセス可能)。

### 23.3.16 タイマ入力 XOR 機能

TIMx\_CR2 レジスタの TI1S ビットを使用して、チャンネル 1 の入力フィルタを、TIMx\_CH1 から TIMx\_CH3 までの 3 つの入力ピンを組み合わせさせた XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。

この機能をホールセンサのインタフェースに使用した例を [セクション 22.3.24 : ホールセンサとのインタフェース \(660 ページ\)](#) に示します。

## 23.3.17 タイマと外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

### スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx\_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx\_ARR、TIMx\_CCRx) が更新されます。

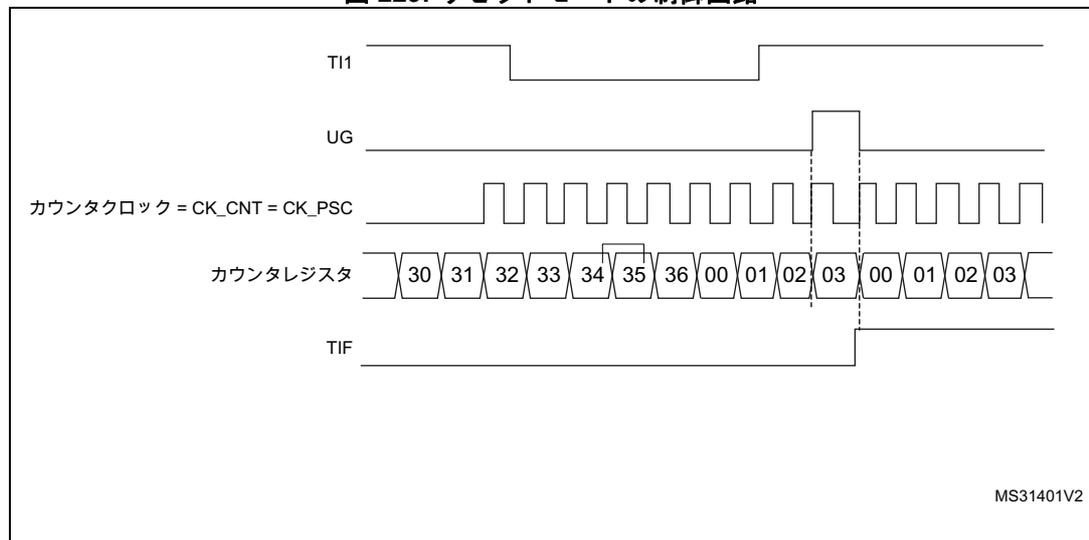
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01)。TIMx\_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx\_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx\_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx\_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 229. リセットモードの制御回路



MS31401V2

## スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

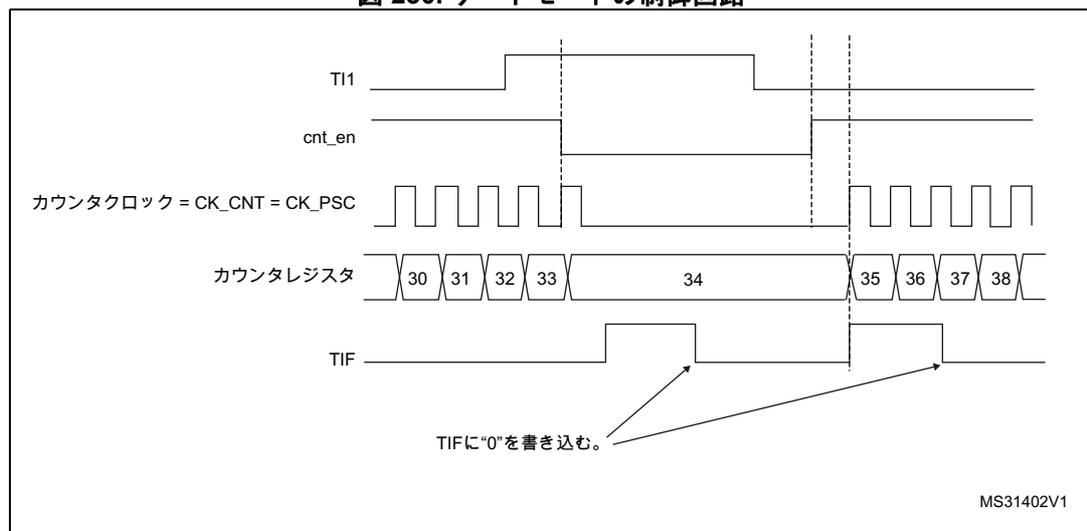
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

1. TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01 ビット)。TIMx\_CCER レジスタで CC1P = 1 と CC1NP = 0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx\_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 230. ゲートモードの制御回路



1. ゲートモードはエッジではなくレベルに対して動作するため、CCxP=CCxNP=1 の設定 (立ち上がり立ち下がり両エッジの検出) はゲートモードでは意味がありません。

**注：** ゲートモードはエッジではなくレベルに対して動作するため、CCxP=CCxNP=1 の設定 (立ち上がり立ち下がり両エッジの検出) はゲートモードでは意味がありません。

## スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

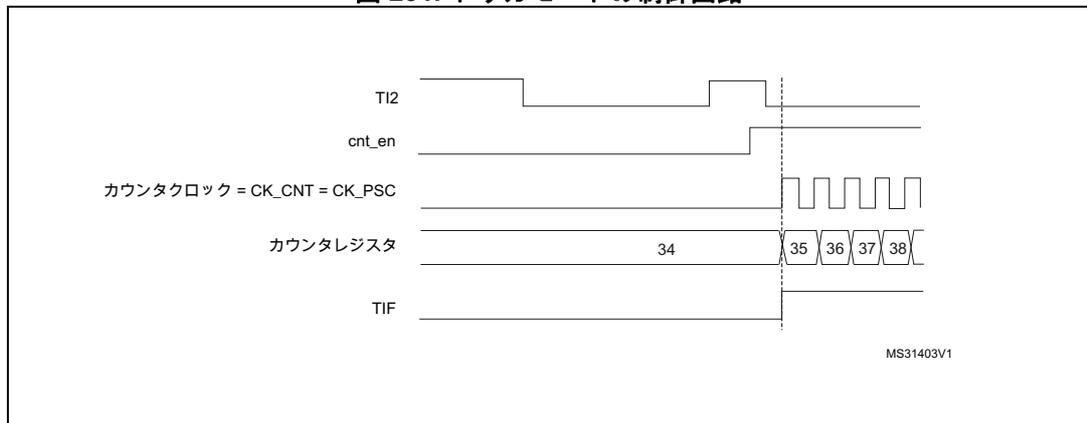
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

1. TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC2S=01)。TIMx\_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 231. トリガモードの制御回路



## スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます (ただし、外部クロックモード 1 とエンコーダモードは除きます)。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます (リセットモード、ゲートモード、およびトリガモードで動作している場合)。TIMx\_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

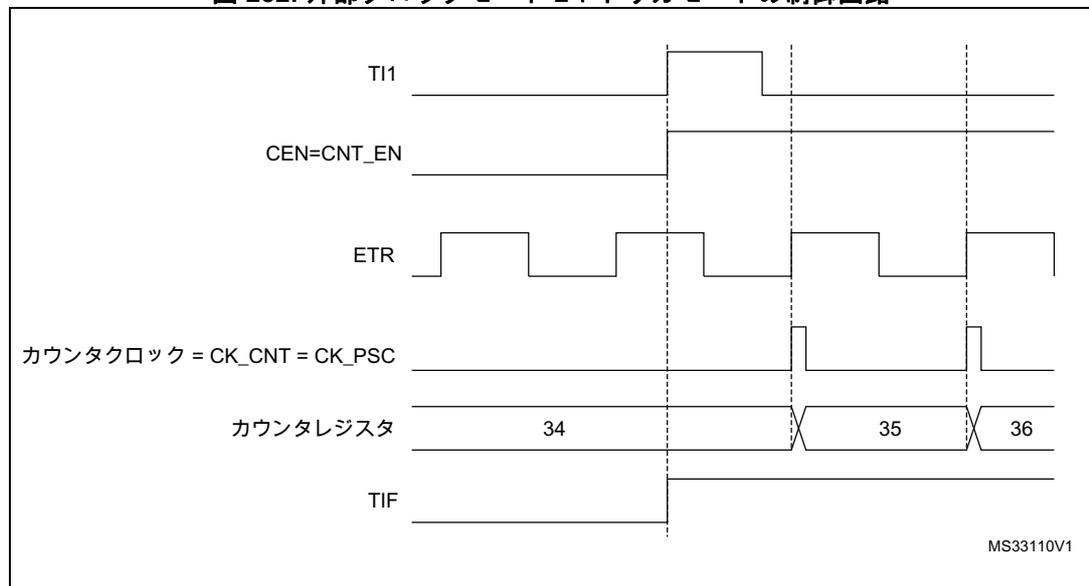
1. TIMx\_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
  - ETF = 0000 : フィルタなし
  - ETPS=00 : プリスケアラ無効
  - ETP=0 : ETR の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。

2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
  - IC1F=0000 : フィルタなし。
  - キャプチャプリスケラはトリガには使用されないなので、設定する必要はありません。
  - TIMx\_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
  - TIMx\_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
3. TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 232. 外部クロックモード 2+トリガモードの制御回路



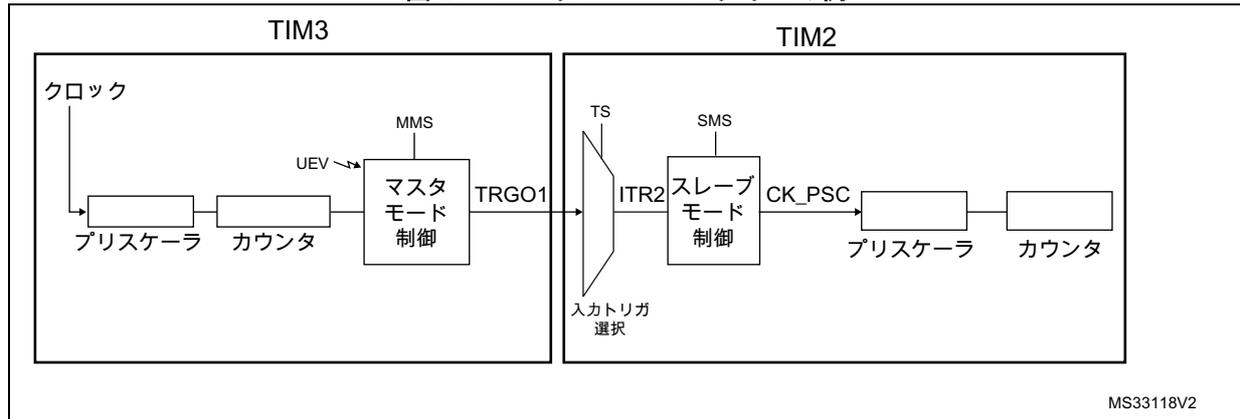
## 23.3.18 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。マスターモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

図 233: マスタ/スレーブタイマの例に、トリガ選択およびマスタモード選択ブロックの概要を示します。

### タイマを別のタイマのプリスケアラとして使用する

図 233. マスタ/スレーブタイマの例



たとえば、TIM 3 が TIM のプリスケアラとして動作するように設定できます。図 233を参照してください。このためには、次の操作を行います。

1. 更新イベント UEV ごとに定期的なトリガ信号を出力するように、TIM 3 をマスタモードに設定します。TIM3\_CR2 レジスタの MMS=010 を書き込んだ場合、更新イベントが生成されるたびに、TRGO1 で立ち上がりエッジが出力されます。
2. TIM 3 の TRGO1 出力を TIM に接続するには、ITR2 を内部トリガとして使用して、TIM をスレーブモードで設定する必要があります。このためには、TIM\_SMCR レジスタの TS ビットを使用します (010 を書き込みます)。
3. 次に、スレーブモードコントローラを外部クロックモード 1 にします (TIM\_SMCR レジスタの SMS=111 を書き込みます)。これにより TIM は、TIM3 の定期的なトリガ信号の立ち上がりエッジ (TIM3 カウンタのオーバーフローに対応しています) をクロックとして動作します。
4. 最後に、それぞれの CEN ビット (TIMx\_CR1 レジスタ) をセットすることによって、両方のタイマを有効にする必要があります。

**注:** TIM3 のトリガ出力として OCx が選択された場合 (MMS=1xx)、その立ち上がりエッジが TIM カウンタのクロックとして使用されます。

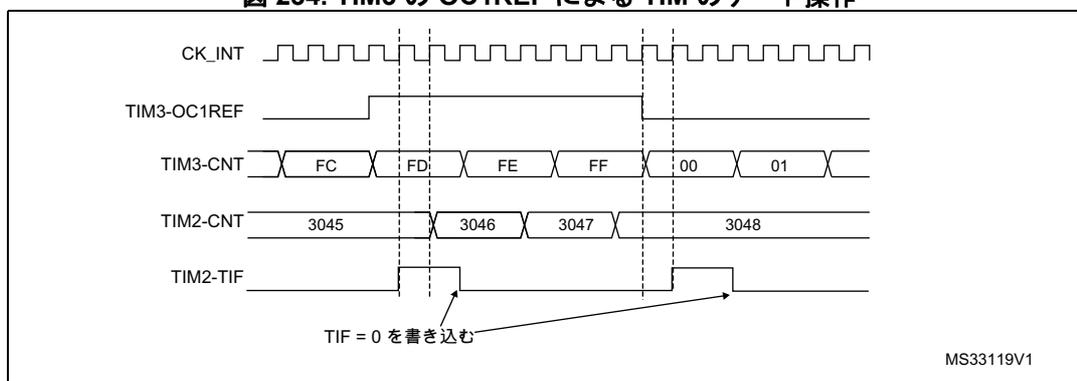
## タイマを使用して別のタイマを有効にする

この例では、TIM の有効化を、タイマ 3 の出力比較 1 で制御します。接続については、[図 233](#) を参照してください。TIM は、分周された内部クロックをカウントします (TIM3 の OC1REF がハイのときのみ) 両方のカウンタクロック周波数は、CK\_INT をプリスケラで 3 分周したものです ( $f_{CK\_CNT} = f_{CK\_INT}/3$ )。

1. TIM 3 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=100)。
2. TIM3 の OC1REF 波形を設定します (TIM3\_CCMR1 レジスタ)。
3. TIM3 から入力トリガを受け取るように TIM を設定します (TIM\_SMCR レジスタの TS=010)。
4. TIM をゲートモードに設定します (TIM\_SMCR レジスタの SMS=101)。
5. CEN ビット (TIM\_CR1 レジスタ) に“1”を書き込んで、TIM を有効にします。
6. CEN ビット (TIM3\_CR1 レジスタ) に“1”を書き込んで、TIM3 を開始します。

**注：** カウンタのクロックはカウンタ 1 と同期しないので、このモードは TIM カウンタのイネーブル信号にのみ影響します。

図 234. TIM3 の OC1REF による TIM のゲート操作

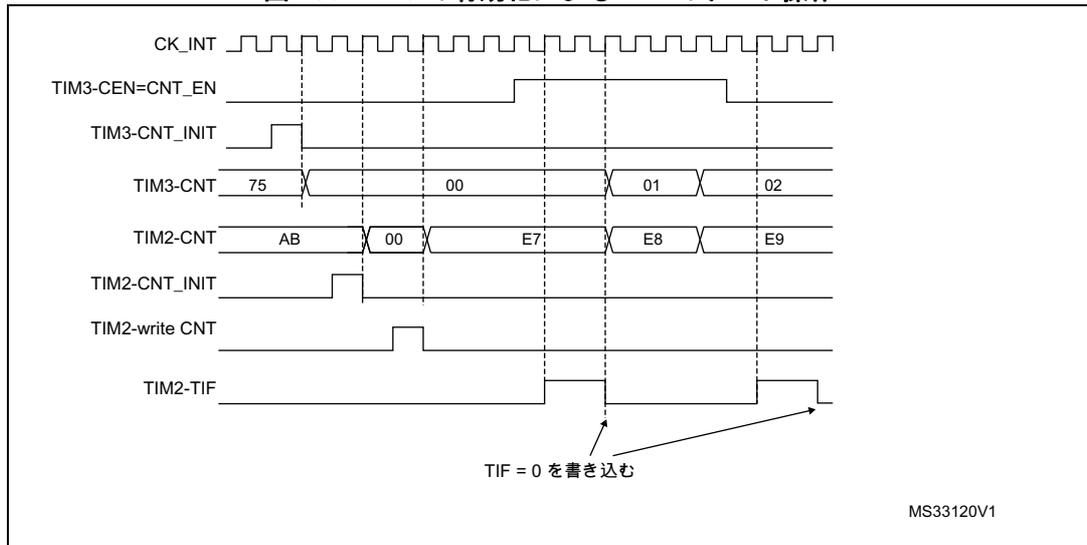


[図 234](#) の例では、TIM のカウンタとプリスケラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。TIM 3 を開始する前に両方のタイマをリセットすることによって、特定の値から開始することが可能です。この場合、タイマカウンタに任意の値を書き込むことができます。TIMx\_EGR レジスタの UG ビットを使用して、ソフトウェアで容易にタイマをリセットできます。

次の例では、TIM3 と TIM を同期させます。TIM 3 はマスタであり、0 からカウントを開始します。TIM はスレーブであり、0xE7 から開始します。プリスケラの分周比は両方のタイマで同じです。TIM3\_CR1 レジスタの CEN ビットに“0”を書き込むことによって TIM3 を無効にすると、TIM は停止します。

1. TIM 3 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=100)。
2. TIM3 の OC1REF 波形を設定します (TIM3\_CCMR1 レジスタ)。
3. TIM3 から入力トリガを受け取るように TIM を設定します (TIM\_SMCR レジスタの TS=010)。
4. TIM をゲートモードに設定します (TIM\_SMCR レジスタの SMS=101)。
5. UG ビット (TIM3\_EGR レジスタ) に“1”を書き込んで、TIM3 をリセットします。
6. UG ビット (TIM\_EGR レジスタ) に“1”を書き込んで、TIM をリセットします。
7. TIM カウンタ (TIM\_CNTL) に“0xE7”を書き込んで、TIM を 0xE7 に初期化します。
8. CEN ビット (TIM\_CR1 レジスタ) に“1”を書き込んで、TIM を有効にします。
9. CEN ビット (TIM3\_CR1 レジスタ) に“1”を書き込んで、TIM3 を開始します。
10. CEN ビット (TIM3\_CR1 レジスタ) に“0”を書き込んで、TIM3 を停止します。

図 235. TIM3 の有効化による TIM のゲート操作

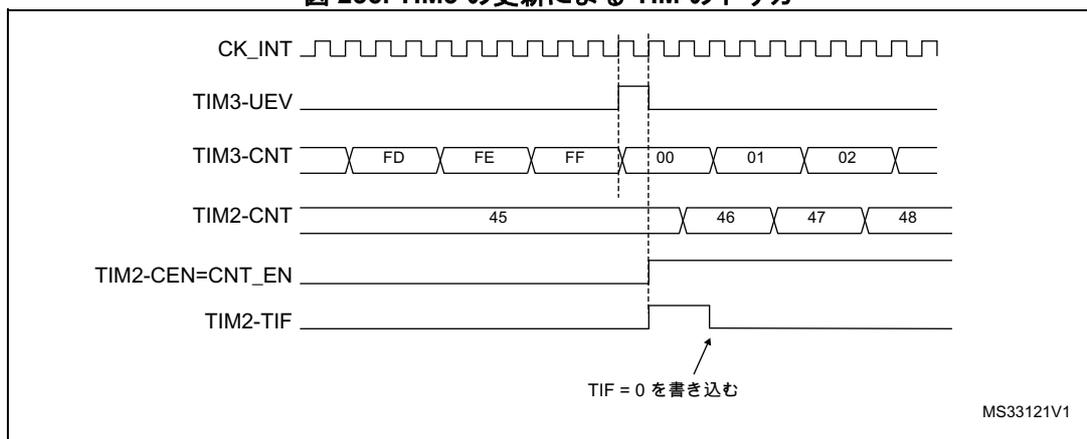


## タイマを使用して別のタイマを開始する

この例では、タイマ 3 の更新イベントによってタイマ の有効化を設定します。接続については、[図 233](#) を参照してください。タイマ 1 によって更新イベントが生成されると、タイマは、分周された内部クロックで現在値（ゼロである必要はありません）からカウントを開始します。タイマがトリガ信号を受信すると、その CEN ビットが自動的にセットされ、カウンタは TIM\_CR1 レジスタの CEN ビットに“0”が書き込まれるまでカウントします。両方のカウンタクロック周波数は、CK\_INT をプリスケラで 3 分周したものです ( $f_{CK\_CNT} = f_{CK\_INT}/3$ )。

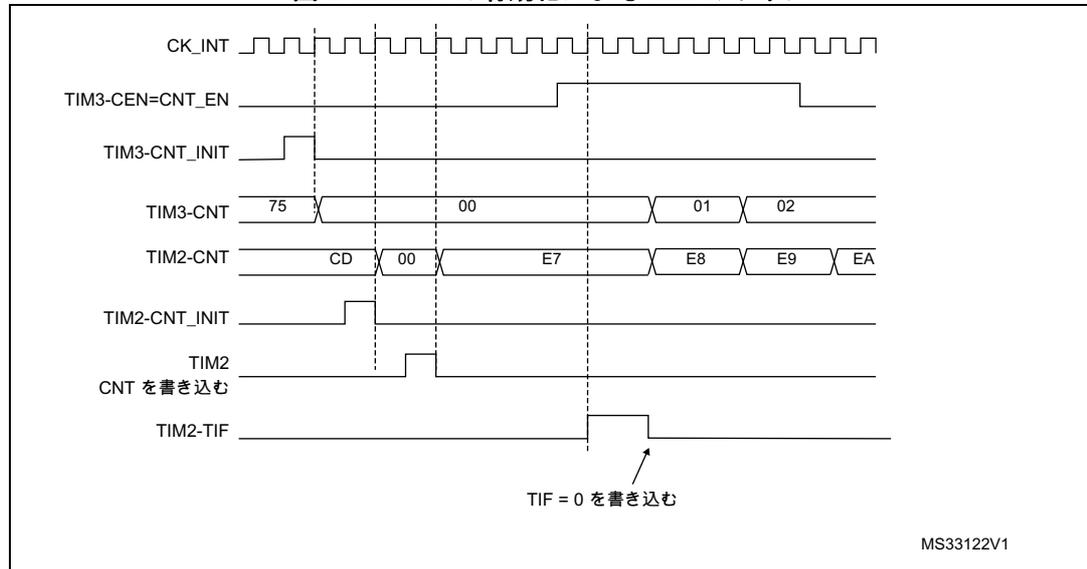
1. TIM3 をマスターモードに設定して、その更新イベント (UEV) をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=010)。
2. TIM3 の周期を設定します (TIM3\_ARR レジスタ)。
3. TIM3 から入力トリガを受け取るように TIM を設定します (TIM\_SMCR レジスタの TS=010)。
4. TIM をトリガモードに設定します (TIM\_SMCR レジスタの SMS=110)。
5. CEN ビット (TIM3\_CR1 レジスタ) に“1”を書き込んで、TIM3 を開始します。

図 236. TIM3 の更新による TIM のトリガ



前の例と同じように、カウントを開始する前に両方のカウンタを初期化することができます。図 237 は 図 236 と同じ設定ですが、ゲートモードではなくトリガモードでの動作を示します (TIM\_SMCR レジスタの SMS=110)。

図 237. TIM3 の有効化による TIM のトリガ



外部トリガに対応して 2 つのタイマを同期して開始します。

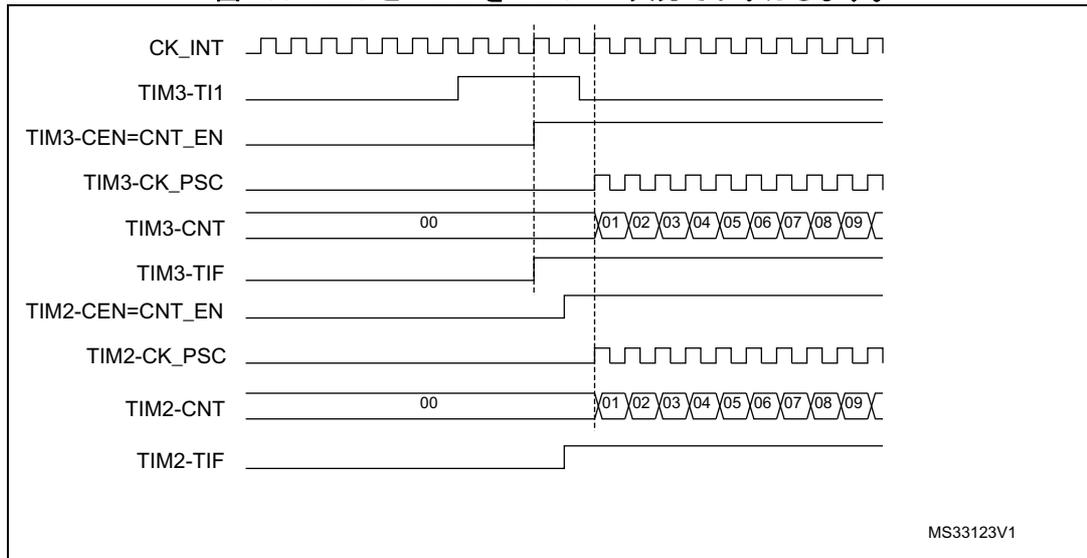
この例では、T11 入力の立ち上がりエッジで TIM3 を有効にし、TIM3 の有効で、TIM2 を有効にします。接続については、図 233 を参照してください。カウンタの同時性を確保するため、TIM3 はマスタ/スレーブモードに設定する必要があります (T11 に対してはスレーブ、TIM2 に対してはマスタ)。

1. TIM3 をマスタモードに設定して、その有効化をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=001)。
2. TIM3 をスレーブモードに設定して、T11 から入力トリガを受け取るようにします (TIM3\_SMCR レジスタの TS=100)。
3. TIM3 をトリガモードに設定します (TIM3\_SMCR レジスタの SMS=110)。
4. MSM=1 を書き込むことによって TIM3 をマスタ/スレーブモードに設定します (TIM3\_SMCR レジスタ)。
5. TIM2 を設定して TIM3 から入力トリガを受け取るように します。(TIM2\_SMCR レジスタの TS=000)。
6. TIM2 をトリガモードに設定します (TIM2\_SMCR レジスタの SMS=110)。

T11 (TIM3) で立ち上がりエッジが発生すると、両方のカウンタが同時に内部クロックによるカウントを開始し、両方の TIF フラグがセットされます。

**注:** この例では、両方のタイマが開始前に初期化されます (それぞれの UG ビットをセットすることによって)。両方のカウンタは 0 から開始しますが、カウンタレジスタ (TIMx\_CNT) に書き込むことによって、容易にオフセットを挿入できます。マスタ/スレーブモードでは、TIM3 の CNT\_EN と CK\_PSC の間に遅延が挿入されます。

図 238. TIM3 と TIM2 を TIM3 TI1 入力でトリガします。



注：スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

### 23.3.19 DMA バーストモード

TIMx タイマには、1つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、タイマの一部をソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx\_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト (バースト) を開始します。TIMx\_DMAR レジスタへの各書き込みは、実際にタイマレジスタの1つにリダイレクトされます。

TIMx\_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数 (ハーフワードまたはバイト) です。

TIMx\_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx\_CR1

00001 : TIMx\_CR2

00010 : TIMx\_SMCR

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
  - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
  - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
  - 転送データ数 = 3 とします (下の注を参照)。
  - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。  
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタのUDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注 :

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

## 23.3.20 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex<sup>®</sup>-M7 コアは停止状態)、TIMx カウンタは、DBGMCU モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、停止します。詳細については、[セクション 40.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 23.4 TIM2/TIM3/TIM4/TIM5 レジスタ

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

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

### 23.4.1 TIMx 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF RE-MAP	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
				r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP**：UIF ステータスビットの再配置

0：再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1：再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD**：クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デジタルフィルタ (ETR、Tix) によって使用されるサンプリングクロックとの間の分周比を示します。

00:  $t_{DTS} = t_{CK\_INT}$

01:  $t_{DTS} = 2 \times t_{CK\_INT}$

10:  $t_{DTS} = 4 \times t_{CK\_INT}$

11: 予約済み

ビット 7 **ARPE**：自動再ロードプリロードイネーブル

0：TIMx\_ARR レジスタはバッファされません。

1：TIMx\_ARR レジスタはバッファされます。

ビット 6:5 **CMS**：センターアラインモード選択

00：エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01：センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10：センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。

11：センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

**注：** カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR**：方向

0：カウンタはアップカウンタとして使用されます。

1：カウンタはダウンカウンタとして使用されます。

**注：** このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。



**ビット 3 OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

**ビット 2 URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。
  - カウンタオーバーフロー/アンダーフロー
  - UG ビットのセット
  - スレーブモードコントローラからの更新生成
- 1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。

**ビット 1 UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。
  - カウンタオーバーフロー/アンダーフロー
  - UG ビットのセット
  - スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

- 1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

**ビット 0 CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

**注 :** 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

## 23.4.2 TIMx 制御レジスタ 2 (TIMx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.							
								rw	rw	rw	rw	rw			

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TI1S** : TI1 選択

0 : TIMx\_CH1 ピンが TI1 入力に接続されます。

1 : TIMx\_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。参照先 : [セクション 22.3.24 : ホールセンサとのインタフェース \(660 ページ\)](#)

ビット 6:4 **MMS** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、GEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在します (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。 (TRGO)

100 : **比較** - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : **比較** - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

110 : **比較** - OC3REF 信号がトリガ出力 (TRGO) として使用されます。

111 : **比較** - OC4REF 信号がトリガ出力 (TRGO) として使用されます。

**注 :** **スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。**

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

## 23.4.3 TIMx スレーブモード制御レジスタ (TIMx\_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SMS[3]** : スレーブモード選択 - ビット 3

SMS 説明を参照 - ビット 2:0

ビット 15 **ETP** : 外部トリガ極性

このビットは、ETR と  $\overline{\text{ETR}}$  のいずれがトリガ動作に使用されるかを選択します。

0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。

1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブになります。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

0 : 外部クロックモード 2 は無効です。

1 : 外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

1 : ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=111)。

2 : 外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません (TS ビットが 111 でないことが必要)。

3 : 外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力は ETRF です。

ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケラ

外部トリガ信号 ETRP の周波数は、最大でも CK\_INT 周波数の 1/4 でなければなりません。プリスケラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。

00 : プリスケラオフ

01 : ETRP 周波数は 2 分周されます。

10 : ETRP 周波数は 4 分周されます。

11 : ETRP 周波数は 8 分周されます。

ビット 11:8 **ETF[3:0]** : 外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

- 0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。
- 0001 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=2
- 0010 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=4
- 0011 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=8
- 0100 :  $f_{SAMPLING}=f_{DTS}/2$ , N=6
- 0101 :  $f_{SAMPLING}=f_{DTS}/2$ , N=8
- 0110 :  $f_{SAMPLING}=f_{DTS}/4$ , N=6
- 0111 :  $f_{SAMPLING}=f_{DTS}/4$ , N=8
- 1000 :  $f_{SAMPLING}=f_{DTS}/8$ , N=6
- 1001 :  $f_{SAMPLING}=f_{DTS}/8$ , N=8
- 1010 :  $f_{SAMPLING}=f_{DTS}/16$ , N=5
- 1011 :  $f_{SAMPLING}=f_{DTS}/16$ , N=6
- 1100 :  $f_{SAMPLING}=f_{DTS}/16$ , N=8
- 1101 :  $f_{SAMPLING}=f_{DTS}/32$ , N=5
- 1110 :  $f_{SAMPLING}=f_{DTS}/32$ , N=6
- 1111 :  $f_{SAMPLING}=f_{DTS}/32$ , N=8

ビット 7 **MSM** : マスタ/スレーブモード

- 0 : 影響なし。
- 1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)。予約済み
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)。予約済み
- 100 : TI1 エッジ検出回路 (TI1F\_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)
- 111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、[表 130:TIMx 内部トリガ接続\(756 ページ\)](#)を参照してください。

**注 :** 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 **OCcs** : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

- 0 : OCREF\_CLR\_INT は、OCREF\_CLR 入力に接続されています。
- 1 : OCREF\_CLR\_INT は、ETRF に接続されています。

ビット 2:0 SMS : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : エンコーダモード 1 - カウンタは、TI1FP2 のレベルに応じて、TI2FP1 のエッジでカウントアップ/ダウンします。

0010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。

0011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。

0100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力の立ち上がりエッジ (TRGI)

カウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

**注 :** トリガ入力として *TI1F\_ED* が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F\_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

**注 :** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 130. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM8/OTG_FS SOF <sup>(1)</sup>	TIM3	TIM4
TIM3	TIM1	TIM2	TIM15	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8

1. TIM2\_OR1 レジスタのビット ITR1\_RMP に依存

## 23.4.4 TIMx DMA／割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IE** : キャプチャ／比較 4 割り込みイネーブル

0 : CC4 割り込みは無効です。

1 : CC4 割り込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割り込みイネーブル

0 : CC3 割り込みは無効です。

1 : CC3 割り込みは有効です。

ビット 2 **CC2IE** : キャプチャ／比較 2 割り込みイネーブル

0 : CC2 割り込みは無効です。

1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル  
 0 : 更新割り込みは無効です。  
 1 : 更新割り込みは有効です。

## 23.4.5 TIMx ステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ  
 CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ  
 CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ  
 CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ  
 このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。  
 0 : オーバーキャプチャは検出されていません。  
 1: CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIF** : トリガ割り込みフラグ  
 このフラグは、トリガイベント時 (スレープモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。  
 ソフトウェアによってクリアされます。  
 0 : トリガイベントは発生していません。  
 1 : トリガ割り込みが保留中です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IF** : キャプチャ/比較 4 割り込みフラグ  
 CC1IF の説明を参照してください。

ビット 3 **CC3IF** : キャプチャ/比較 3 割り込みフラグ  
 CC1IF の説明を参照してください。

ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ  
 CC1IF の説明を参照してください。

**ビット 1 CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合** : このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます。センターアラインモード (TIMx\_CR1 レジスタの CMS ビットの説明を参照) および再トリガ可能なワンパルスモードでは、例外もあります。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。

**CC1 チャンネルが入力として設定されている場合** : このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (選択された極性に一致するエッジが IC1 で検出されました)。

**ビット 0 UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

オーバーフローまたはアンダーフロー時 (TIM2 から TIM4)、および TIMx\_CR1 レジスタの UDIS=0 である場合。

TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイイベントによって CNT が再初期化されたとき (同期制御レジスタの説明を参照)。

## 23.4.6 TIMx イベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG								
									w		w	w	w	w	w

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx\_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4G** : キャプチャ/比較 4 生成  
CC1G の説明を参照してください。

ビット 3 **CC3G** : キャプチャ/比較 3 生成  
CC1G の説明を参照してください。

ビット 2 **CC2G** : キャプチャ/比較 2 生成  
CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

**CC1 チャンネルが出力として設定されている場合 :**

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

**CC1 チャンネルが入力として設定されている場合 :**

カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケラカウンタもクリアされます (プリスケラ比は変化しません)。センターラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx\_ARR) をとります。

## 23.4.7 TIMx キャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							Res.								Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

### 出力比較モード

ビット 31:25 予約済み、常に 0 として読み出されます。

ビット 24 **OC2M[3]** : 出力比較 2 モード - ビット 3

ビット 23:17 予約済み、常に 0 として読み出されます。

ビット 16 **OC1M[3]** : 出力比較 1 モード - ビット 3

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 6:4 の OC1M 説明を参照

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC2S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

OC1CE : 出力比較 1 クリアイネーブル

0 : OC1Ref は ETRF 入力の影響を受けません。

1 : OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。



ビット 6:4 **OC1M** : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

0001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

0010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

0011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

0100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

0101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合は非アクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 OR です。

1101 : 組み合わせ PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 AND です。

1110 : 非対称 PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

1111 : 非対称 PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

**注 :** 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2: PWM モードでは、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときのみ、OCREF のレベルが変化します。

**ビット 3 OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注 :** 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2: PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

**ビット 2 OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

**ビット 1:0 CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済み、常に 0 として読み出されます。

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケーラ

**ビット 9:8 CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** CC2S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。

**ビット 7:4 IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

0001 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2

0010 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4

0011 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8

0100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 6

0101 :  $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 :  $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 :  $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8

1001 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8

1010 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1011 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

1110 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

1111 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

**ビット 3:2 IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E=0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

**ビット 1:0 CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 23.4.8 TIMx キャプチャ/比較モードレジスタ 2 (TIMx\_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000

上記の CCMR1 レジスタの説明を参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
							Res.								Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]					IC3F[3:0]			IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

### 出力比較モード

ビット 31:25 予約済み、常に 0 として読み出されます。

ビット 24 **OC4M[3]** : 出力比較 2 モード - ビット 3

ビット 23:17 予約済み、常に 0 として読み出されます。

ビット 16 **OC3M[3]** : 出力比較 1 モード - ビット 3

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

OC1M の説明 (TIMx\_CCMR1 レジスタのビット 6:4) を参照

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード

OC1M の説明 (TIMx\_CCMR1 レジスタのビット 6:4) を参照

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC3S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときのみ書き込み可能です。**

## 入力キャプチャモード

ビット 31:16 予約済み、常に 0 として読み出されます。

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケール

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC4S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときのみ書き込み可能です。**

ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケール

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC3S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときのみ書き込み可能です。**

## 23.4.9 TIMx キャプチャ／比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw												

ビット 15 **CC4NP** : キャプチャ／比較 4 出力極性  
CC1NP の説明を参照してください。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC4P** : キャプチャ／比較 4 出力極性  
CC1P の説明を参照してください。

ビット 12 **CC4E** : キャプチャ／比較 4 出カインェブル。  
CC1E の説明を参照してください。

ビット 11 **CC3NP** : キャプチャ／比較 3 出力極性  
CC1NP の説明を参照してください。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC3P** : キャプチャ／比較 3 出力極性  
CC1P の説明を参照してください。

ビット 8 **CC3E** : キャプチャ／比較 3 出カインェブル。  
CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性  
CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ／比較 2 出力極性  
CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ／比較 2 出カインェブル。  
CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ／比較 1 出力極性

**CC1** チャンネルが出力として設定されている場合 : この場合、CC1NP はクリアされたままでなければなりません。

**CC1** チャンネルが入力として設定されている場合 : このビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

**CC1 チャンネルが出力として設定されている場合 :**

- 0 : OC1 はアクティブハイです。
- 1 : OC1 はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :** CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。

00 : 非反転/立ち上がりエッジ

回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。

01 : 反転/立ち下がりエッジ

回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ

回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル。

**CC1 チャンネルが出力として設定されている場合 :**

- 0 : オフ - OC1 はアクティブではありません。
- 1 : オン - OC1 信号は、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :** このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

- 0 : キャプチャは無効です。
- 1 : キャプチャは有効です。

**表 131. 標準 OCx チャンネルの出力制御ビット**

CCxE ビット	OCx 出力状態
0	出力無効 (OCx=0、OCx_EN=0)
1	OCx=OCxREF + 極性、OCx_EN=1

**注 :** 標準 OCx チャンネルに接続されている外部 IO ピンの状態は、OCx チャンネルの状態と、GPIO および AFIO レジスタに依存します。

## 23.4.10 TIMx カウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31] または UIFCPY	CNT[30:16] (タイマに依存)															
rw または r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT[15:0]															
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 値は TIMx\_CR1 の UIFREMAP に依存します。

UIFREMAP = 0 の場合 :

**CNT[31]** : カウンタ値の最上位ビット (TIM2 および TIM5)

他のタイマで予約済み

UIFREMAP = 1 の場合 :

**UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー

ビット 30:16 **CNT[30:16]** : カウンタ値の最上位部分 (TIM2 および TIM5)

ビット 15:0 **CNT[15:0]** : カウンタ値の最下位部分

## 23.4.11 TIMx プリスケーラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSC[15:0]															
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **PSC[15:0]** : プリスケーラ値

カウンタクロック周波数 CK\_CNT は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケーラレジスタにロードされる値を含みます。

## 23.4.12 TIMx 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **ARR[31:16]** : 自動再ロード値上位ビット (TIM2 および TIM5)

ビット 15:0 **ARR[15:0]** : 自動再ロードプリスケアラ値下位ビット

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 23.3.1 : タイムベースユニット \(708 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 23.4.13 TIMx キャプチャ/比較モードレジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR1[31:16]** : キャプチャ/比較 1 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値下位ビット

**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**チャンネル CC1 が入力として設定されている場合 :**

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 23.4.14 TIMx キャプチャ/比較モードレジスタ 2 (TIMx\_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR2[31:16]** : キャプチャ/比較 2 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR2[15:0]** : キャプチャ/比較 2 値下位ビット

**CC2 チャンネルが出力として設定されている場合 :**

CCR2 は、実際のキャプチャ/比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 2 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC2 出力に送信される値を含みます。

**CC2 チャンネルが入力として設定されている場合 :**

CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

## 23.4.15 TIMx キャプチャ/比較モードレジスタ 3 (TIMx\_CCR3)

アドレスオフセット : 0x3C

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR3[31:16]** : キャプチャ/比較 3 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR3[15:0]** : キャプチャ/比較値下位ビット

**CC3 チャンネルが出力として設定されている場合 :**

CCR3 は、実際のキャプチャ/比較 3 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 3 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC3 出力に送信される値を含みます。

**チャンネル CC3 が入力として設定されている場合 :**

CCR3 は、最後の入力キャプチャ 3 イベント (IC3) によって転送されたカウンタ値です。

## 23.4.16 TIMx キャプチャ/比較モードレジスタ 4 (TIMx\_CCR4)

アドレスオフセット : 0x40

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR4[31:16]** : キャプチャ/比較 4 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR4[15:0]** : キャプチャ/比較値下位ビット

1. CC4 チャンネルが出力として設定されている場合 (CC4Sビット) :  
CCR4 は、実際のキャプチャ/比較 4 レジスタにロードされる値 (プリロード値) です。  
TIMx\_CCMR2 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 4 レジスタにコピーされます。  
アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC4 出力に送信される値を含みます。
2. CC4 チャンネルが入力として設定されている場合 (TIMx\_CCMR4 レジスタの CC4S ビット) :  
CCR4 は、最後の入力キャプチャ 4 イベント (IC4) によって転送されたカウンタ値です。

## 23.4.17 TIMx DMA 制御レジスタ (TIMx\_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送回数 (タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します) を指定します。

- 00000 : 1 回転送
- 00001 : 2 回転送、
- 00010 : 3 回転送、
- ...
- 10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

- 例 :
- 00000 : TIMx\_CR1
  - 00001 : TIMx\_CR2
  - 00010 : TIMx\_SMCR
  - ...

例 : 次の転送を考えます : DBL = 7 回転送 かつ DBA = TIMx\_CR1。この場合、転送は、TIMx\_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

## 23.4.18 完全転送の TIMx DMA アドレス (TIMx\_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(TIMx\_CR1 \text{ アドレス}) + (DBA + \text{DMA インデックス}) \times 4$$

ここで、TIMx\_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx\_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx\_DCR 内で設定)。



## 23.4.19 TIM2 オプションレジスタ 1 (TIM2\_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ITR1_RMP[1:0]		Res.									
				rw	rw										

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **ITR1\_RMP[1:0]** : 内部トリガ 1 再配置

ソフトウェアでセット/クリアされます。

00 : TIM8\_TRGOUT

01 : PTP トリガ出力は TIM2\_ITR1 に接続されます。

10 : OTG FS SOF は TIM2\_ITR1 入力に接続されます。

11 : OTG HS SOF は TIM2\_ITR1 入力に接続されます。

ビット 9:0 予約済みであり、リセット値に保持する必要があります。

## 23.4.20 TIM2 オプションレジスタ 1 (TIM5\_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TI4_RMP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.							
								rw	rw						

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **TI4\_RMP[1:0]** : タイマ 入力 4 再配置

ソフトウェアでセット/クリアされます。

00 : TIM5 チャンネル 4 を GPIO に接続 : データシートのオルタナート機能配置表を参照してください。

01 : LSI の内部クロックは較正のために TIM5\_CH4 入力へ接続

10 : LSE の外部クロックは較正のために TIM5\_CH4 入力へ接続

11 : RTC ウェイクアップ割り込みは較正のために TIM5\_CH4 入力へ接続

ウェイクアップ割り込みを有効化する必要があります。

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

## 23.4.21 TIM3 オプションレジスタ 1 (TIM3\_OR1)

アドレスオフセット : 0x50

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	T11_RMP[1:0]														
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 T11\_RMP[1:0] : 入力キャプチャ 1 再配置

00 : TIM3 入力キャプチャ 1 は I/O へ接続されます。

01 : TIM3 入力キャプチャ 1 は COMP1\_OUT へ接続されます。

10 : TIM3 入力キャプチャ 1 は COMP2\_OUT へ接続されます。

11 : TIM3 入力キャプチャ 1 は、COMP1\_OUT と COMP2\_OUT との論理 OR へ接続されます。

## 23.4.22 TIMx レジスタマップ

TIMx レジスタは、次の表のようにマップされます。

表 132. TIM2/TIM3/TIM4/TIM5 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	UIFREMAP	Res.	Res.	CKD [1:0]	ARPE	CMS [1:0]	DIR	OPM	URS	UDIS	CEN																		
	リセット値																									0	0	0	0	0	0	0	0	0
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T11S	MMS[2:0]		CCDS	Res.	Res.	Res.																		
	リセット値																										0	0	0	0	0			
0x08	TIMx_SMCR	Res.	SMS[3]	ETP	ECE	ETPS [1:0]	ETF[3:0]			MSM	TS[2:0]		Res.	SMS[2:0]																				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	TIMx_DIER	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE																	
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF																
	リセット値																											0	0	0	0	0	0	0
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG																	
	リセット値																											0	0	0	0	0	0	



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

表 132. TIM2/TIM3/TIM4/TIM5 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18	TIMx_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	OC1M[3]	OC2CE	OC2M [2:0]	Res.	Res.	OC2PE	OC2FE	CC2S [1:0]	Res.	OC1CE	OC1M [2:0]	Res.	Res.	OC1PE	OC1FE	CC1S [1:0]							
	リセット値								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]	IC2 PSC [1:0]	Res.	Res.	IC4PSC	IC4FE	CC4S [1:0]	Res.	IC3F[3:0]	Res.	Res.	IC3PSC	IC3FE	CC3S [1:0]		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	TIMx_CCMR2 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	OC3M[3]	O24CE	OC4M [2:0]	Res.	Res.	OC4PE	OC4FE	CC4S [1:0]	Res.	OC3CE	OC3M [2:0]	Res.	Res.	OC3PE	OC3FE	CC3S [1:0]							
	リセット値								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]	IC4 PSC [1:0]	Res.	Res.	IC4PSC	IC4FE	CC4S [1:0]	Res.	IC3F[3:0]	Res.	Res.	IC3PSC	IC3FE	CC3S [1:0]		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	TIMx_CNT	CNT[31]または UIFCPY CNT[30:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CNT[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIMx_PSC	PSC[15:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIMx_ARR	ARR[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																ARR[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	????																																
0x34	TIMx_CCR1	CCR1[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR1[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIMx_CCR2	CCR2[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR2[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIMx_CCR3	CCR3[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR3[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIMx_CCR4	CCR4[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR4[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	予約済み																																



表 132. TIM2/TIM3/TIM4/TIM5 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	TIMx_DCR	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]																					
	リセット値																					0	0	0	0	0				0	0	0	0
0x4C	TIMx_DMAR	Res.	DMAB[15:0]																														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	TIM2_OR	Res.	Res.	Res.	Res.	ITR1_RMP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																
	リセット値																						0	0									
0x50	TIM5_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI4_RMP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.																
	リセット値																									0	0						
0x50	TIM3_OR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																
	リセット値																																

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 24 汎用タイマ (TIM9 から TIM14)

### 24.1 TIM9 から TIM14 の概要

TIM9 から TIM14 汎用タイマは、プログラム可能なプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

これらのカウンタは、入力信号パルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM) など様々な目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

タイマ TIM9 から TIM14 は完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 24.3.12](#)に示すように、相互に同期させることができます。

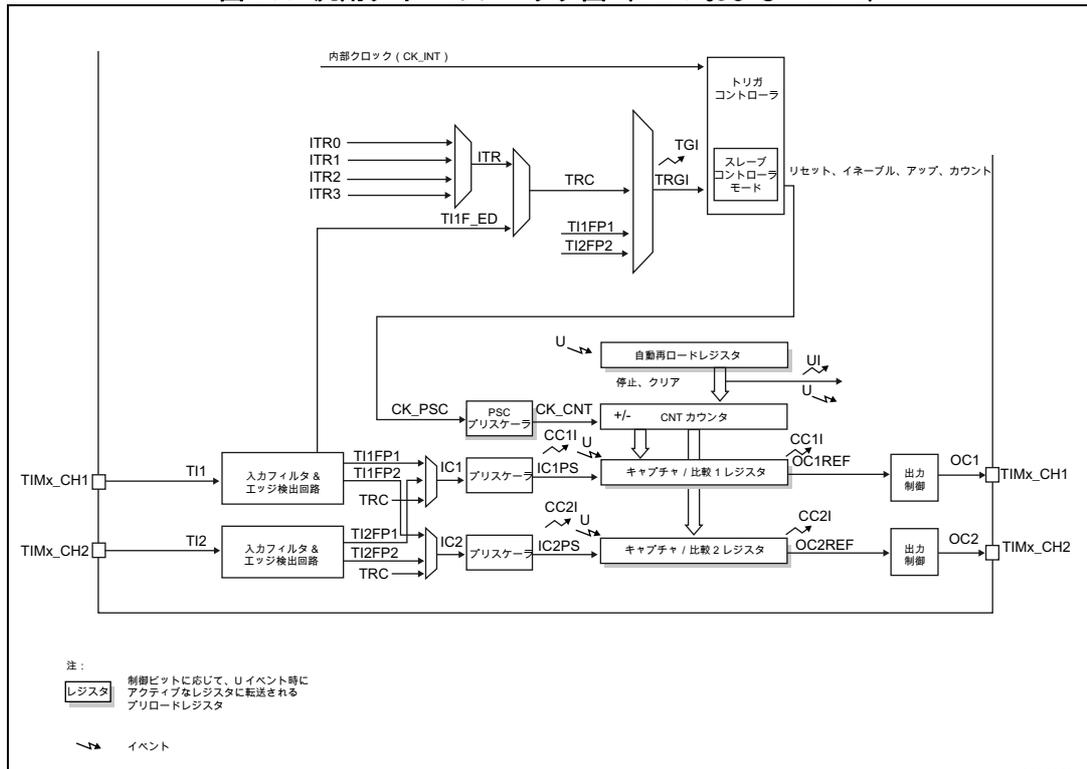
### 24.2 TIM9 から TIM14 の主な機能

#### 24.2.1 TIM9/TIM12 の主な機能

TIM9 から TIM14 の汎用タイマの主な機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケラ (動作中に変更可能)。
- 次の機能を持つ、最大 2 つの独立チャンネル。
  - 入力キャプチャ
  - 出力比較
  - PWM 生成 (エッジアラインモード)
  - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み生成。
  - 更新: カウンタオーバーフロー、カウンタ初期化 (ソフトウェアまたは内部トリガによる)
  - トリガイベント (内部トリガによるカウンタの開始、停止、初期化、またはカウント)
  - 入力キャプチャ
  - 出力比較

図 239. 汎用タイマのブロック図 (TIM9 および TIM12)

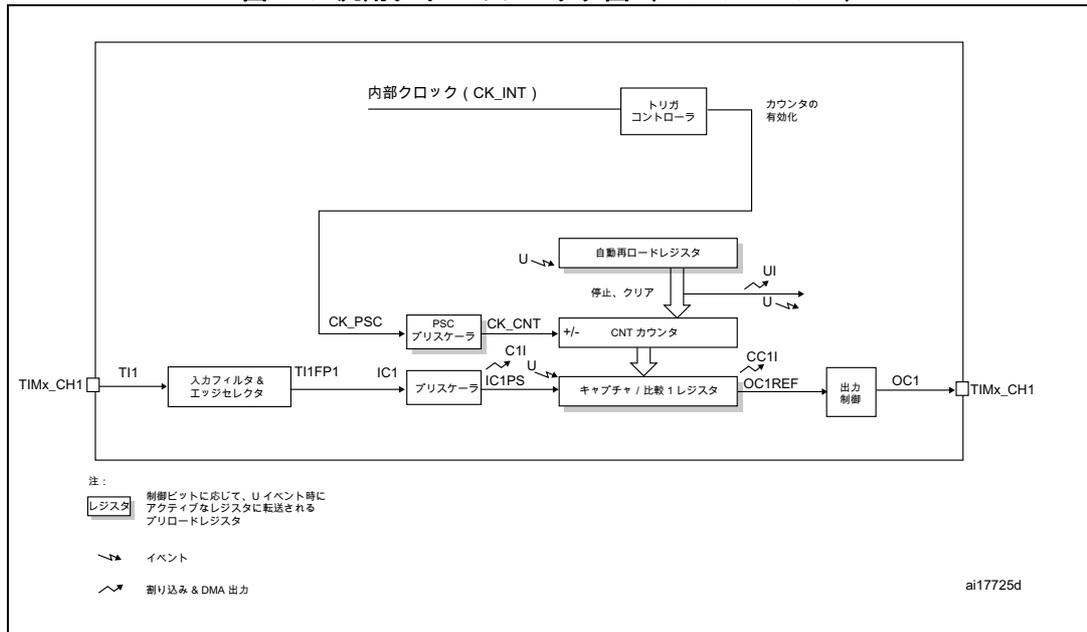


## 24.2.2 TIM10/TIM11 および TIM13/TIM14 の主な機能

汎用タイマ TIM10/TIM11 および TIM13/TIM14 の主な機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケアラ (動作中に変更可能)。
- 次の機能を持つ独立チャネル：
  - 入力キャプチャ
  - 出力比較
  - PWM 生成 (エッジアラインモード)
  - ワンパルスモード出力
- 以下のイベント時の割り込み生成。
  - 更新：カウンタオーバーフロー、カウンタの初期化 (ソフトウェアによる)
  - 入力キャプチャ
  - 出力比較

図 240. 汎用タイマのブロック図 (TIM10/11/13/14)



## 24.3 TIM9 から TIM14 の機能説明

### 24.3.1 タイムベースユニット

タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。このカウンタはカウントアップします。

カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC)
- 自動再ロードレジスタ (TIMx\_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定ごとに詳しく説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK\_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx\_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 241](#) と [図 242](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 241. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

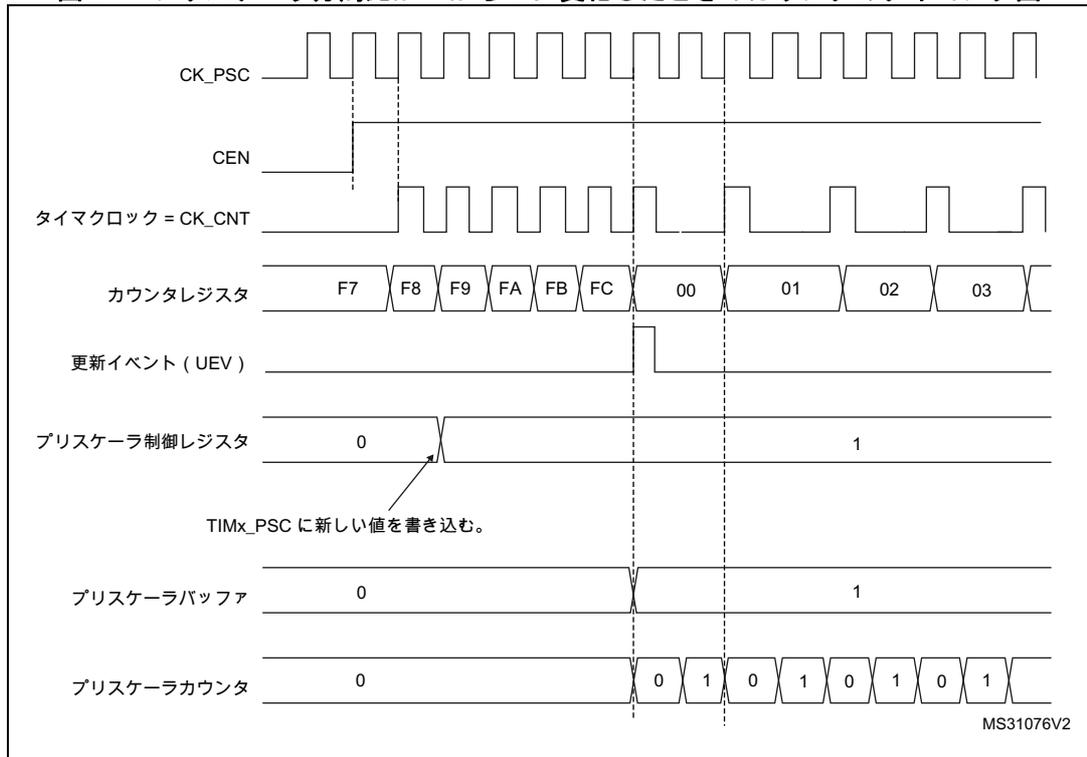
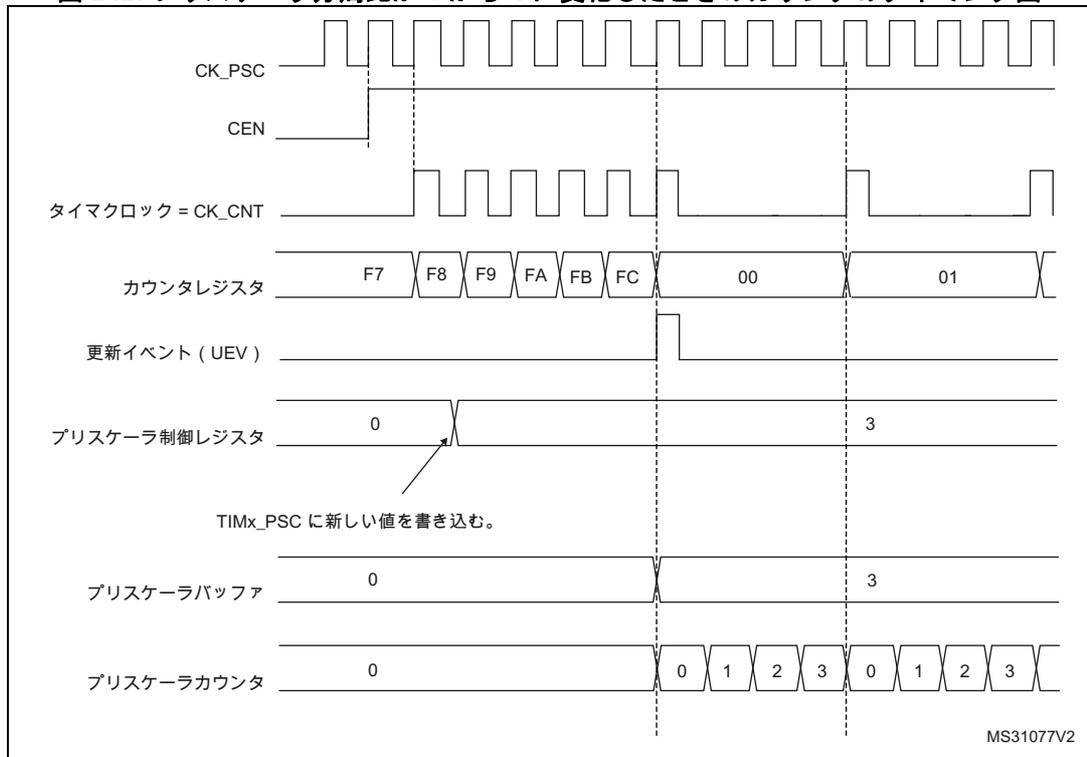


図 242. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 24.3.2 カウンタモード

### アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

(ソフトウェアによって、または TIM9 および TIM12 のスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケーラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケーラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。
- プリスケーラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 243. 内部クロック分周比が 1 の場合のカウンタのタイミング図

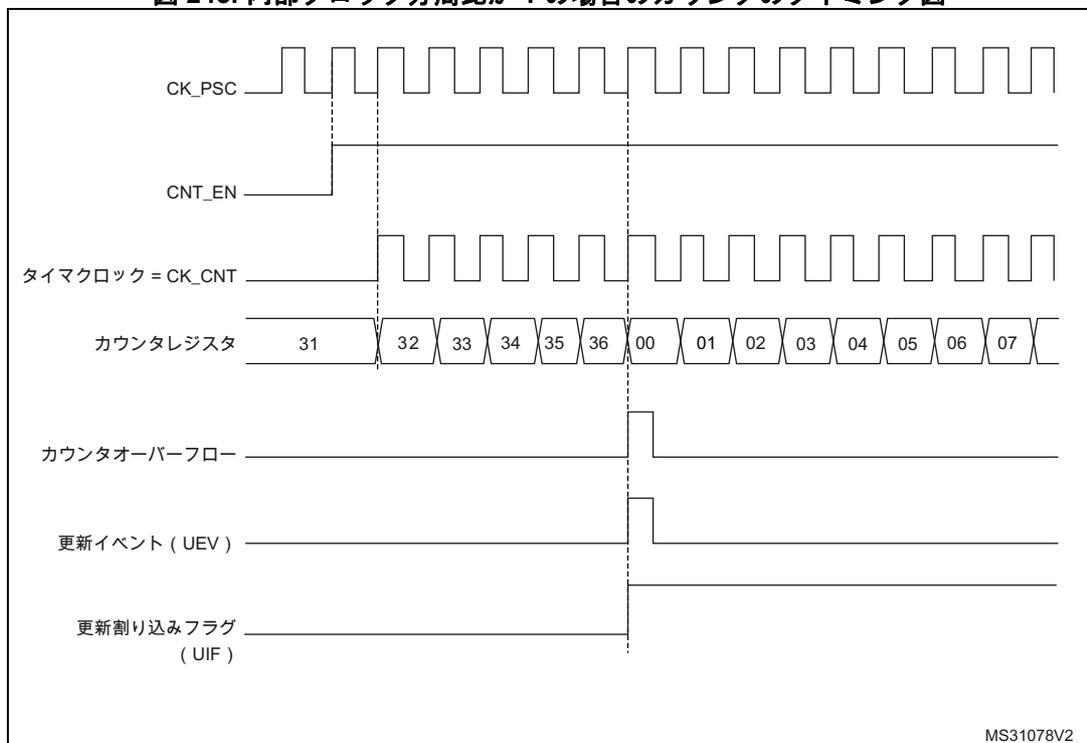


図 244. 内部クロック分周比が 2 の場合のカウンタのタイミング図

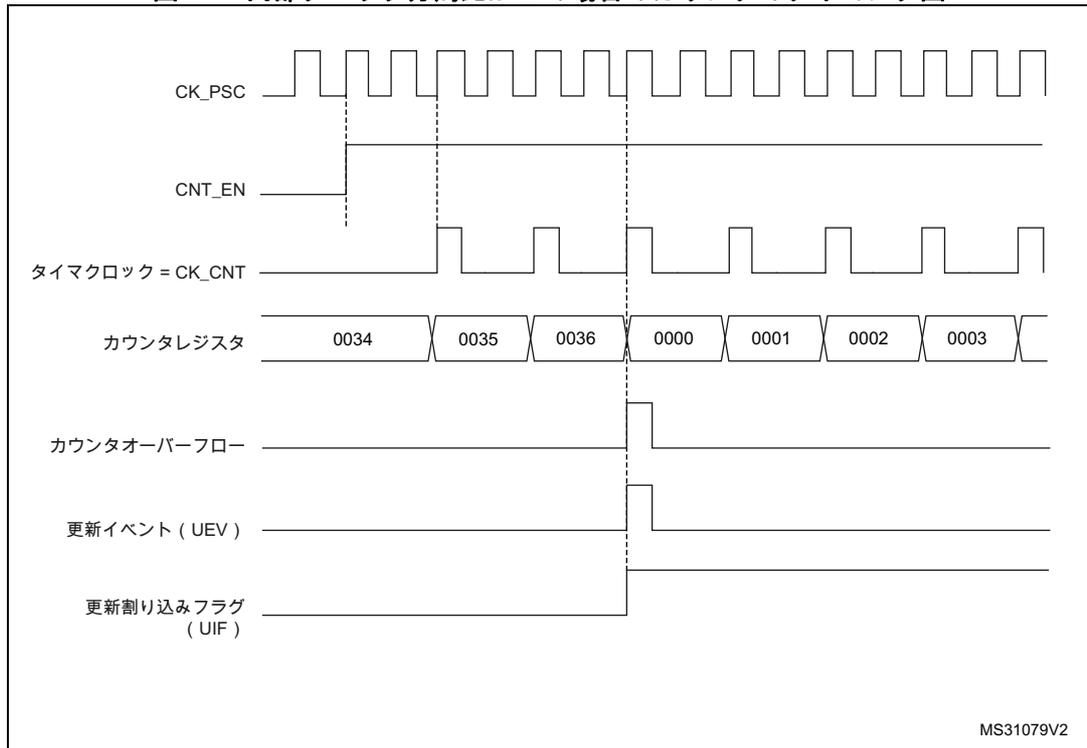


図 245. 内部クロック分周比が 4 の場合のカウンタのタイミング図

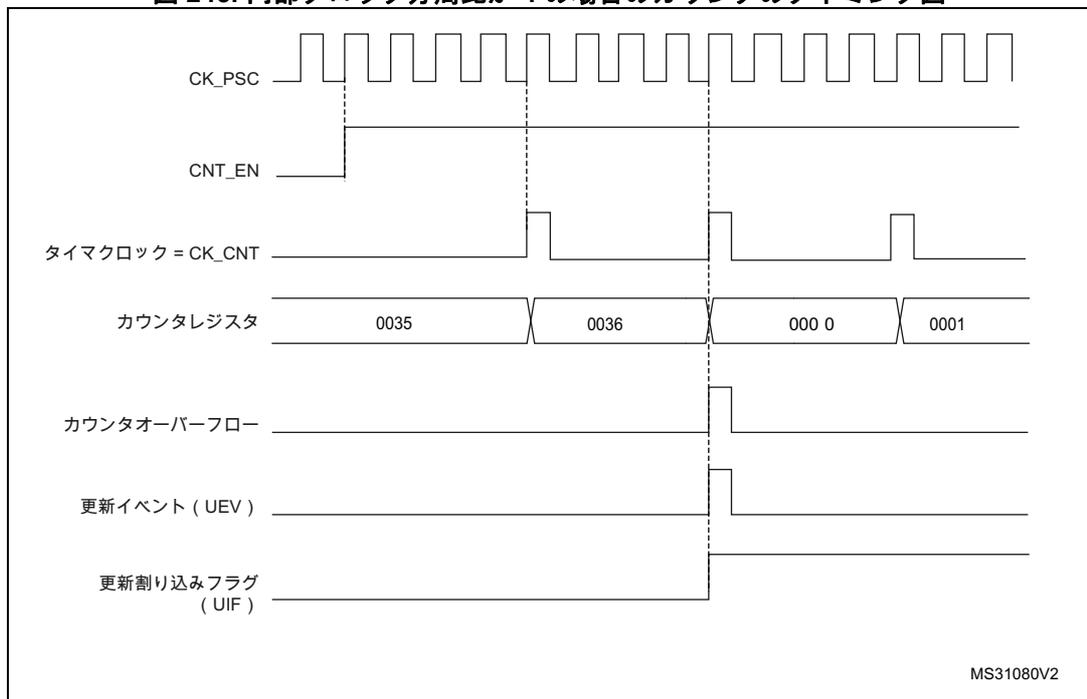


図 246. 内部クロック分周比が N の場合のカウンタのタイミング図

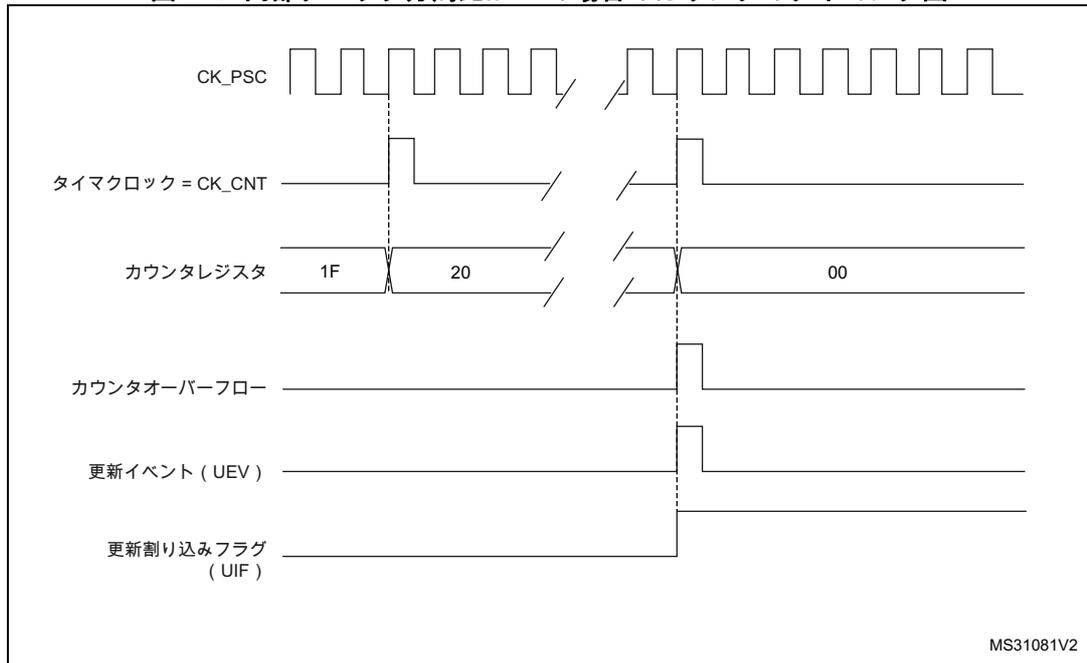


図 247. ARPE=0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

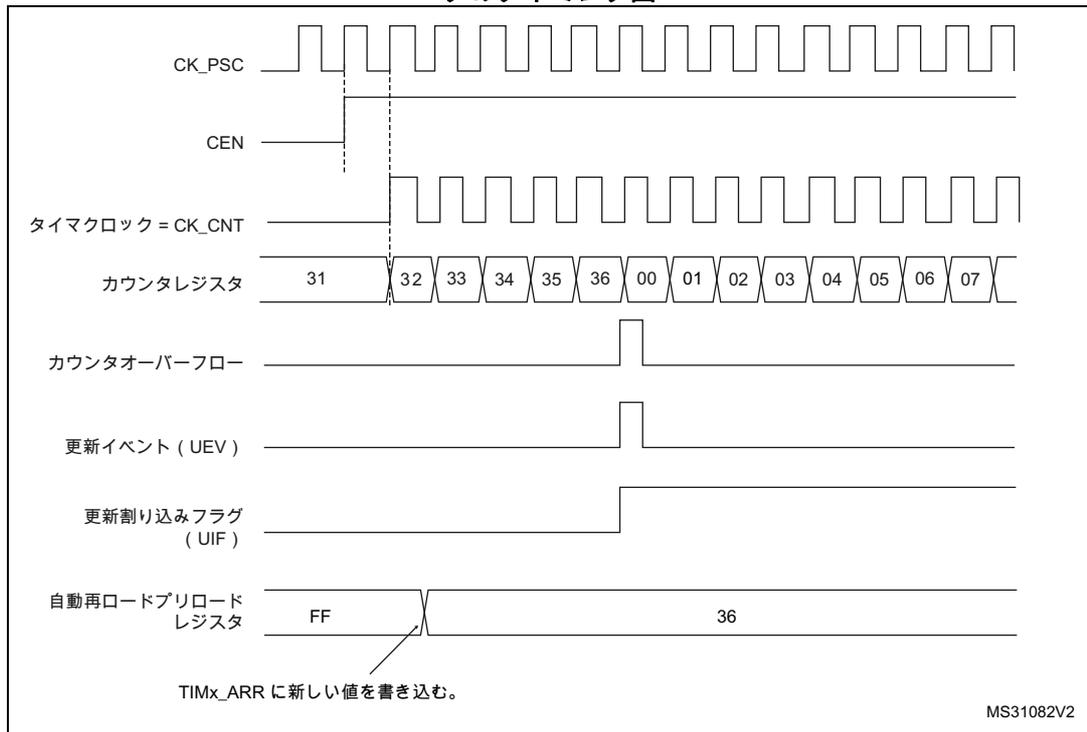
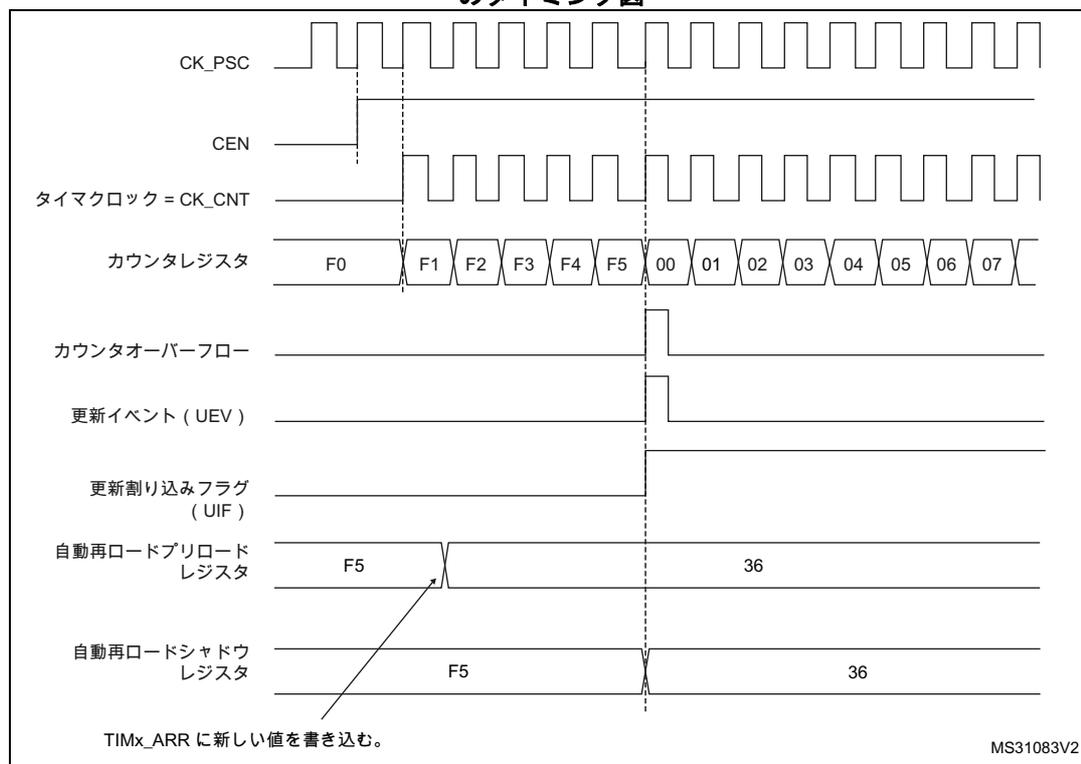


図 248. ARPE = 1 (TIMx\_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



## 24.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

- 内部クロック (CK\_INT)
- 外部クロックモード 1 (TIM9 および TIM12) : 外部入力ピン (Tix)
- 内部トリガ入力 (ITRx) (TIM9 および TIM12) : 他のタイマからトリガ出力を接続。詳細については、[セクション : タイマを別のタイマのプリスケラとして使用する](#)を参照してください。

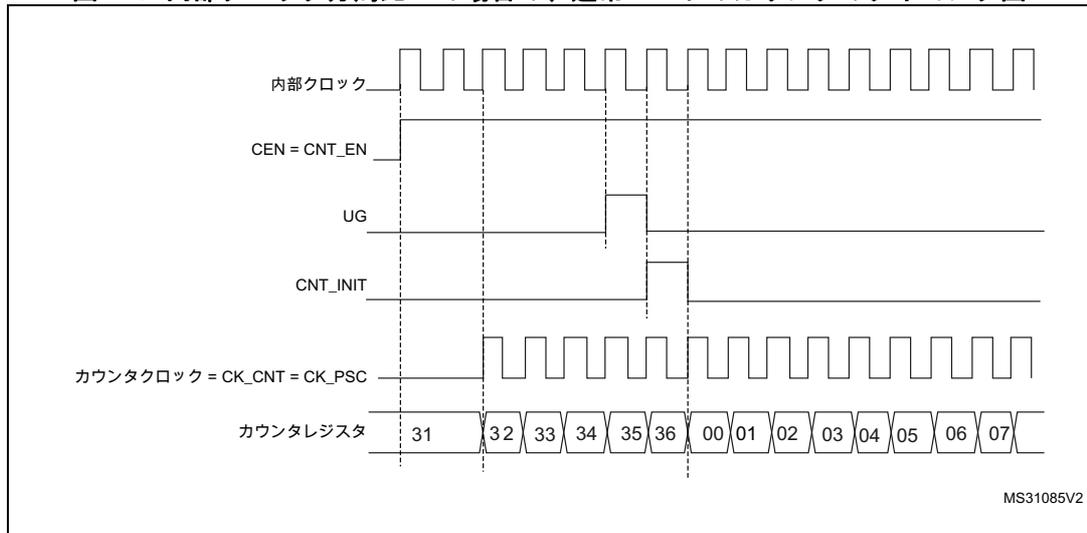
### 内部クロックソース (CK\_INT)

内部クロックソースは、TIM10/TIM11 および TIM13/TIM14 のデフォルトクロックソースです。

TIM9 および TIM12 に対しては、スレーブモードコントローラが無効の場合 (SMS="000")、内部クロックソースが選択されます。その際、TIMx\_CR1 レジスタの CEN ビットと TIMx\_EGR レジスタの UG ビットが制御ビットとして使用され、ソフトウェアからのみ変更できます (クリアされたままの UG ビットは除く)。CEN ビットに 1 が設定されると直ちに、プリスケラにはクロックとして内部クロック CK\_INT が供給されます。

図 249 に、プリスケラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

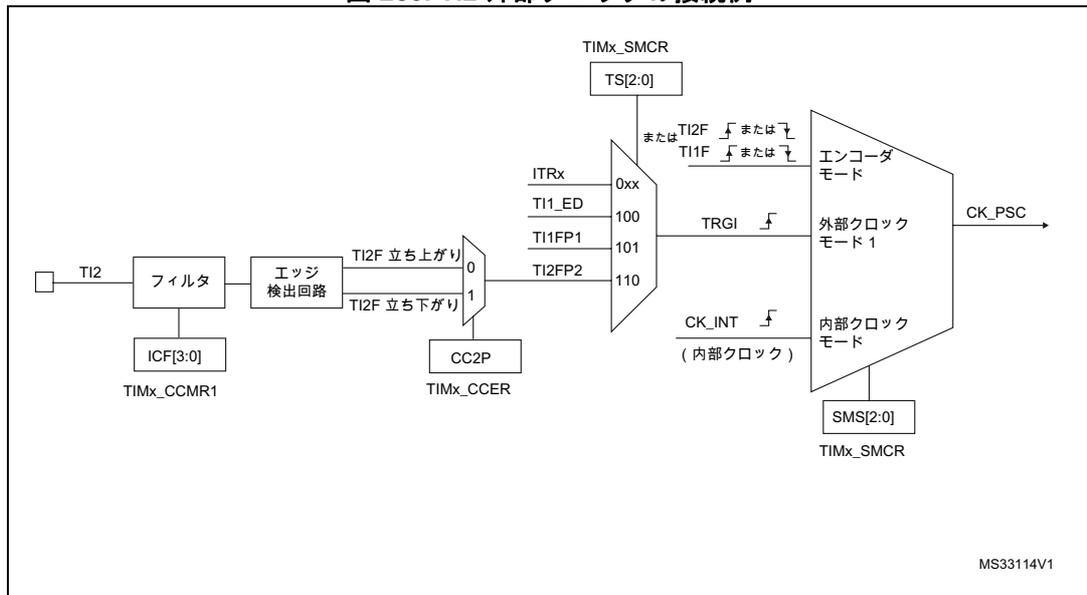
図 249. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



## 外部クロックソースモード 1 (TIM9 および TIM12)

このモードは TIMx\_SMCR レジスタの SMS = 111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 250. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

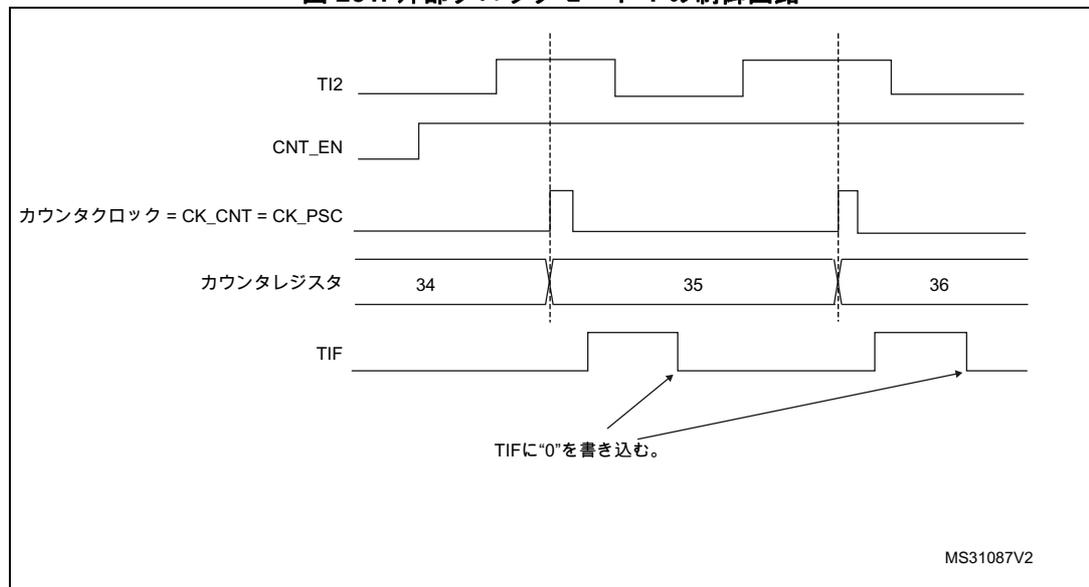
1. TIMx\_CCMR1レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx\_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F = 0000 にしておきます)。
3. TIMx\_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込むことによって、立ち上がりエッジ極性を選択します。
4. TIMx\_SMCR レジスタの SMS ビットに“111”を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx\_SMCR レジスタの TS ビットに“110”を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx\_CR1 レジスタの CEN ビットに“1”を書き込むことによって、カウンタを有効にします。

**注:** キャプチャプリスケアラはトリガには使用されないため、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 251. 外部クロックモード 1 の制御回路



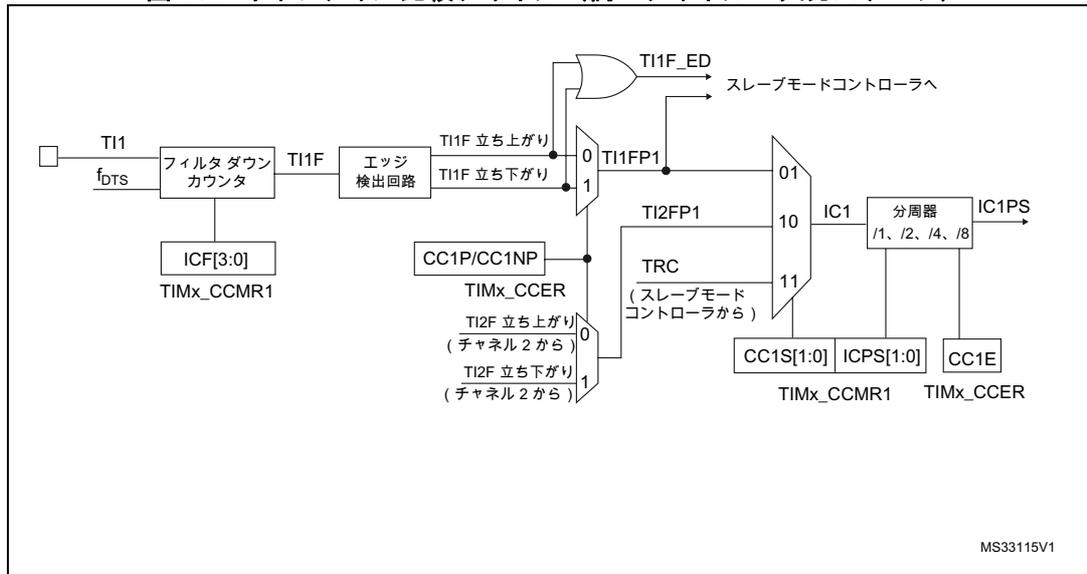
## 24.3.4 キャプチャ/比較チャネル

各キャプチャ/比較チャネルは、キャプチャ/比較レジスタ (シャドウレジスタを含む)、キャプチャの入カステージ (デジタルフィルタ、マルチプレクサ、プリスケアラ)、および出力ステージ (比較回路と出力制御) から構成されています。

図 252 から 図 254 に、1 つのキャプチャ/比較チャネルの概要を示します。

入カステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った TIxF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIxFPx) を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 252. キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)



出力ステージは、OCxRef (アクティブハイ) として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 253. キャプチャ/比較チャンネル 1 メイン回路

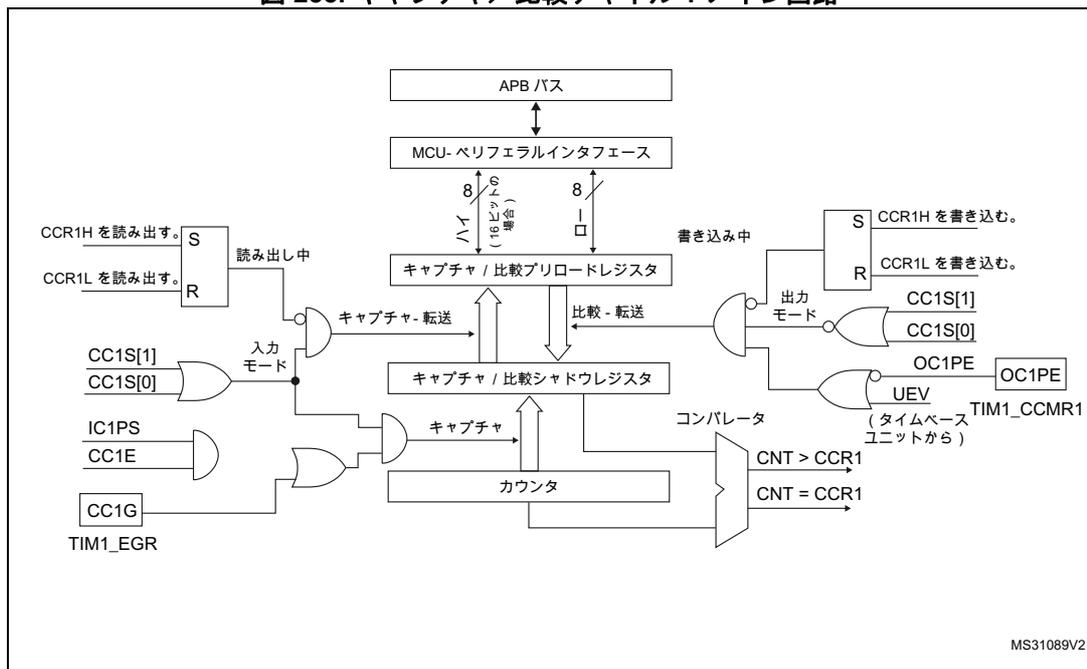
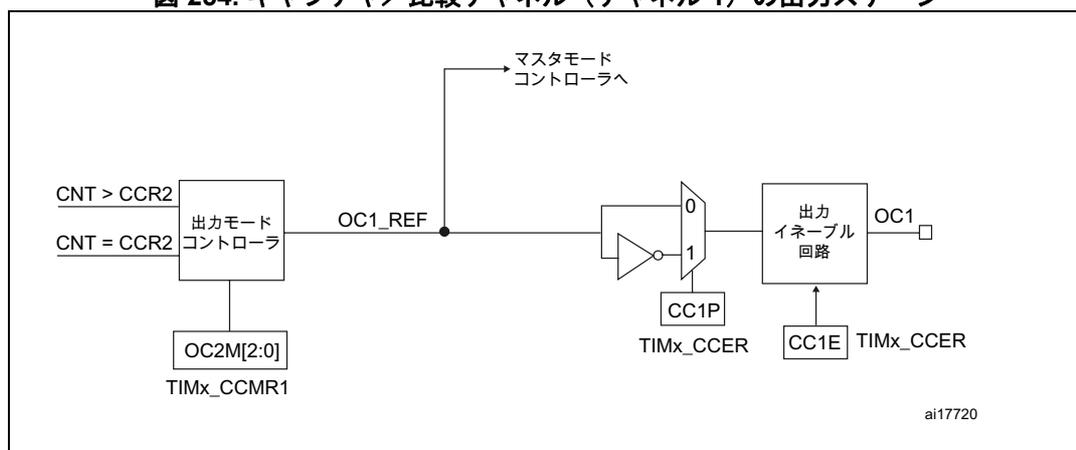


図 254. キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ



キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

## 24.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx\_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx\_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx\_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx\_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx\_CCR1 は TI1 入力にリンクされていなければならないので、TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化するとすぐに、チャンネルは入力モードに設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号に対して必要とする入力フィルタ時間を設定します (入力が TIx 入力の内の一つである場合、TIMx\_CCMRx レジスタの ICxF ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数  $f_{DTS}$  でサンプリング)。次に、TIMx\_CCMR1 レジスタの IC1F ビットに“0011”を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx\_CCER レジスタの CC1P ビットと CC1NP ビットに“00”を設定します (この場合、立ち上がりエッジの選択)。
4. 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
5. TIMx\_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。

- 必要に応じて、TIMx\_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

**注：** TIMx\_EGR レジスタの対応する CCxG ビットをセットすることで、IC 割り込みリクエストをソフトウェアで発生させることができます。

## 24.3.6 PWM 入力モード (TIM9/12 の場合のみ)

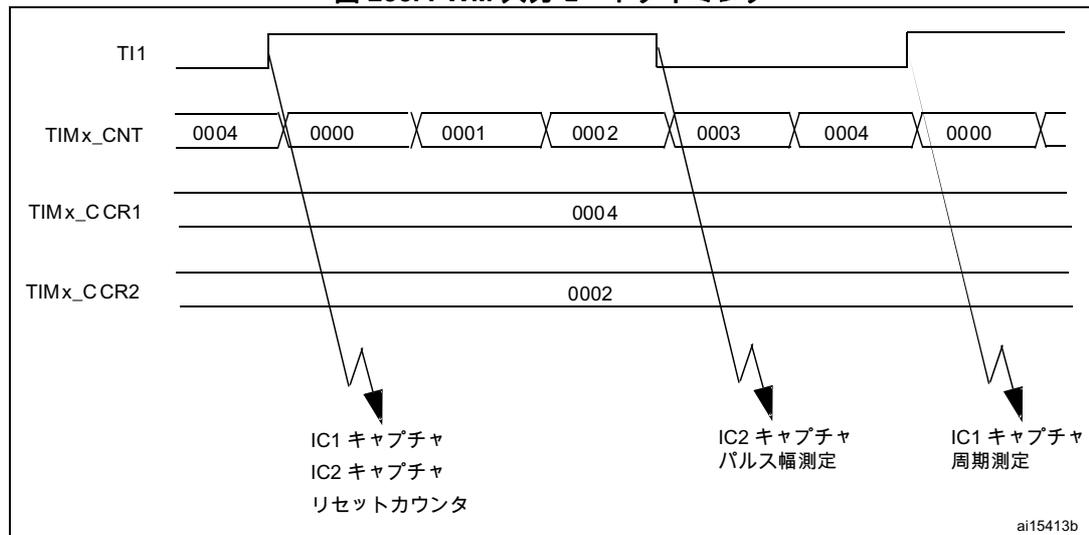
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2 つの ICx 信号が同じ Tlx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます (手順は、CK\_INT 周波数とプリスケール値によって、若干異なることがあります)。

- TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込むことによって (TI1 を選択)、TIMx\_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“00”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx\_CCR1 のキャプチャとカウンタクリアの両方に使用)。
- TIMx\_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
- CC2P ビットと CC2NP ビットに“11”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx\_CCR2 のキャプチャに使用されます)。
- TIMx\_SMCR レジスタの TS ビットに“101”を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx\_SMCR レジスタの SMS ビットに“100”を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx\_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 255. PWM 入力モードタイミング



1. TI1FP1 と TI2FP2 のみがスレーブモードコントローラに接続されているので、PWM 入力モードは TIMx\_CH1/TIMx\_CH2 信号でのみ使用できます。

## 24.3.7 強制出力モード

出力モード (TIMx\_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF、次に OCx) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx\_OCMRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例 : CCxP = 0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx\_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx\_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みリクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

## 24.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

1. 対応する出力ピンに、出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx\_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM = 000)、アクティブにセットされるか (OCxM = 001)、インアクティブにセットされるか (OCxM = 010)、または反転されます (OCxM = 011)。
2. 割り込みステータスレジスタのフラグをセットします (TIMx\_SR レジスタの CCxIF ビット)。
3. 対応する割り込みマスク (TIMx\_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。

TIMx\_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx\_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

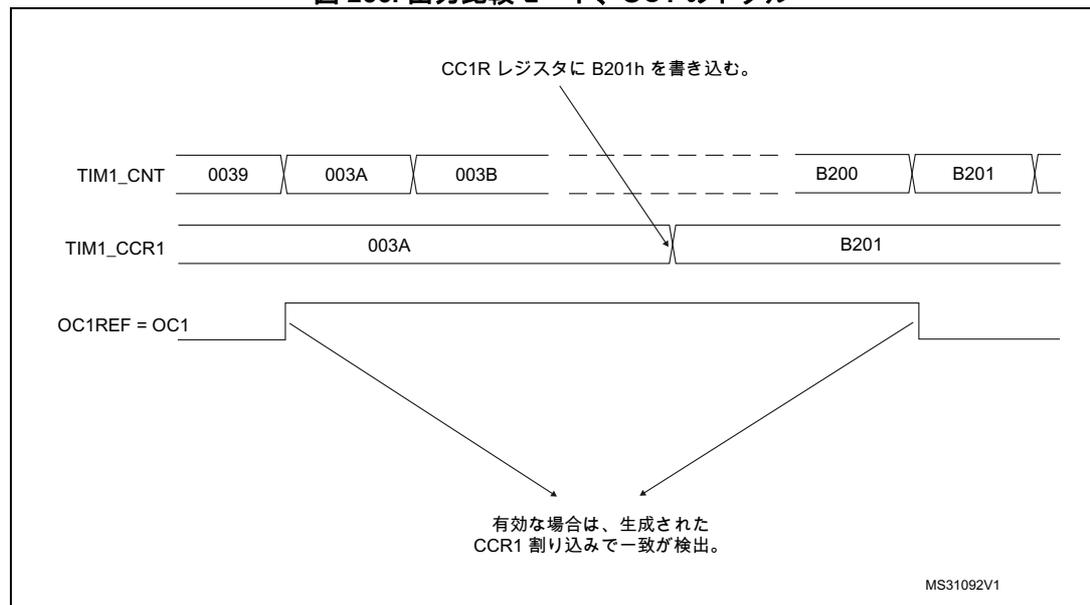
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順：

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx\_ARR レジスタと TIMx\_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
  - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに“011”を書き込みます。
  - プリロードレジスタを無効にするには、OCxPE ビットに“0”を書き込みます。
  - アクティブハイ極性を選択するには、CCxP ビットに“0”を書き込みます。
  - 出力を有効にするには、CCxE ビットに“1”を書き込みます。
5. TIMx\_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx\_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx\_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 256](#) に示します。

図 256. 出力比較モード、OC1 のトグル



## 24.3.9 PWM モード

パルス幅変調(PWM)モードでは、TIMx\_ARR レジスタの値によって決められた周波数と TIMx\_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx\_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx\_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx\_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx の極性は、TIMx\_CCER レジスタの CCxP ビットを使用して、ソフトウェアでプログラムできます。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx\_CCER レジスタの CCxE ビットによって有効化されます。詳細については、TIMx\_CCERx レジスタの説明を参照してください。

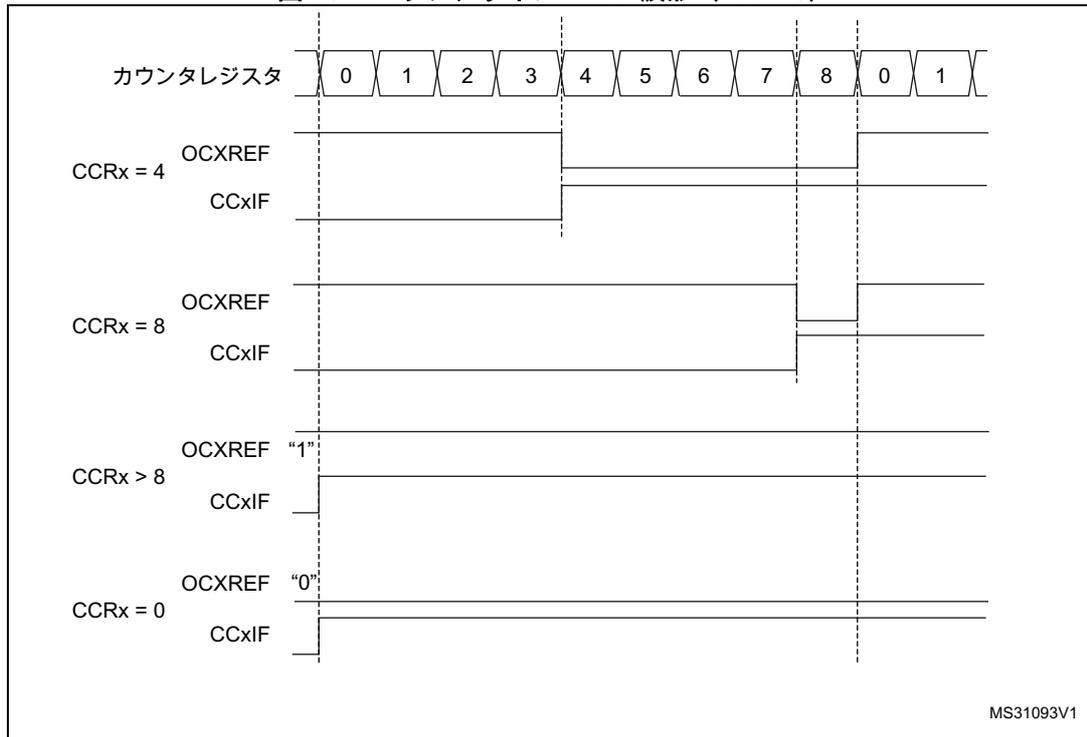
PWM モード(1 または 2)では、TIMx\_CNT と TIMx\_CCRx を常に比較して、 $TIMx\_CNT \leq TIMx\_CCRx$  かどうか判断されます。

カウンタはカウントアップしているため、タイマはエッジアラインモードでのみ PWM を生成できません。

### PWM エッジアラインモード

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、 $TIMx\_CNT < TIMx\_CCRx$  の間はハイに、そうでない場合はローになります。TIMx\_CCRx の比較値が自動再ロード値 (TIMx\_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。図 257 に TIMx\_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 257. エッジアライン PWM 波形 (ARR=8)



## 24.3.10 ワンパルスモード

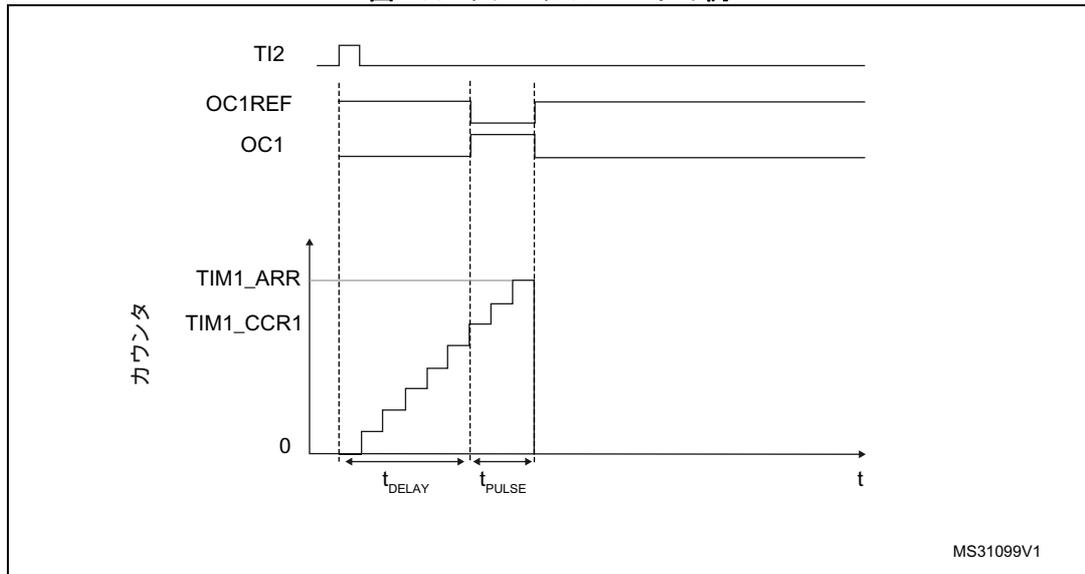
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

$$CNT < CCRx \leq ARR \text{ (特に } 0 < CCRx \text{ の場合)}$$

図 258. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が  $t_{PULSE}$  の正のパルスを遅延時間  $t_{DELAY}$  後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

1. TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
2. TI2FP2 は、立ち上がりエッジを検出して、TIMx\_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込みます。
3. TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx\_SMCR レジスタの TS ビットに“110”を書き込みます。
4. TI2FP2 を使用してカウンタを開始します。このためには、TIMx\_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケールを考慮に入れて)。

- $t_{DELAY}$  は、TIMx\_CCR1 レジスタに書き込まれた値によって定義されます。
- $t_{PULSE}$  は、自動再ロード値と比較値の差 (TIMx\_ARR - TIMx\_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx\_CCMR1 レジスタの OC1M ビットに“111”を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx\_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx\_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイベントを待ちます。この例では、CC1P に“0”を書き込みます。

必要なパルスは 1 つだけなので (シングルモード)、TIMx\_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx\_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

## 特殊なケース : OCx 高速イネーブル

ワンパルスモードでは、Tlx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ( $t_{\text{DELAY min}}$ ) が制限されます。

最小遅延で波形を出力したい場合は、TIMx\_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

## 24.3.11 TIM9/12 外部トリガ同期

TIM9/12 タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

### スレーブモード : リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx\_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx\_ARR、TIMx\_CCRx) が更新されます。

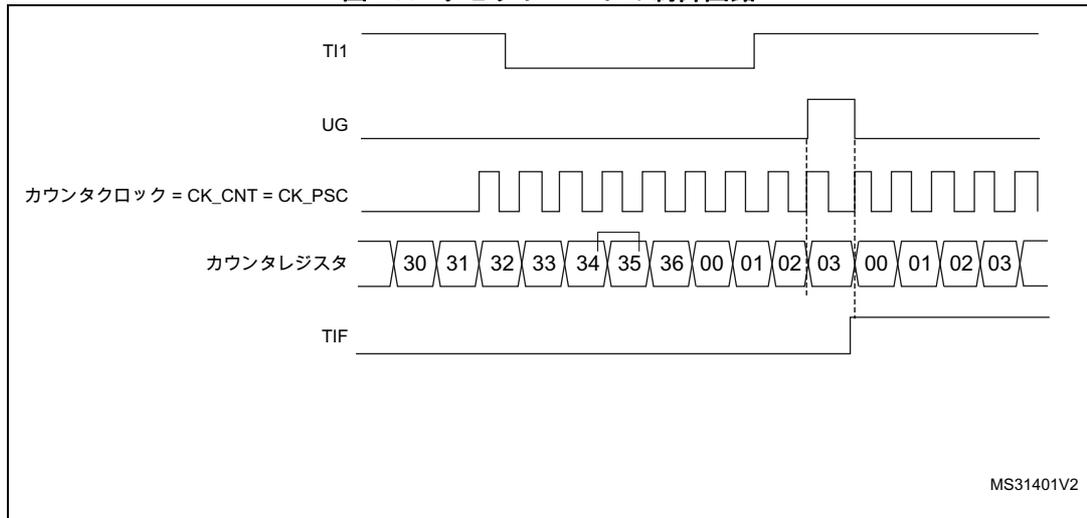
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S = 01)。TIMx\_CCER レジスタの CC1P と CC1NP に "00" を書き込んで、極性を有効にします (その後、立ち上がりエッジのみを検出)。
2. TIMx\_SMCR レジスタの SMS ビットに "100" を書き込むことによって、タイマをリセットモードに設定します。TIMx\_SMCR レジスタの TS ビットに "101" を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタの CEN ビットに "1" を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx\_SR レジスタの TIF ビット)、(TIMx\_DIER レジスタの TIE ビットが) 有効な場合は割り込みリクエストを送信できます。

次の図は、自動再ロードレジスタ TIMx\_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 259. リセットモードの制御回路



## スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

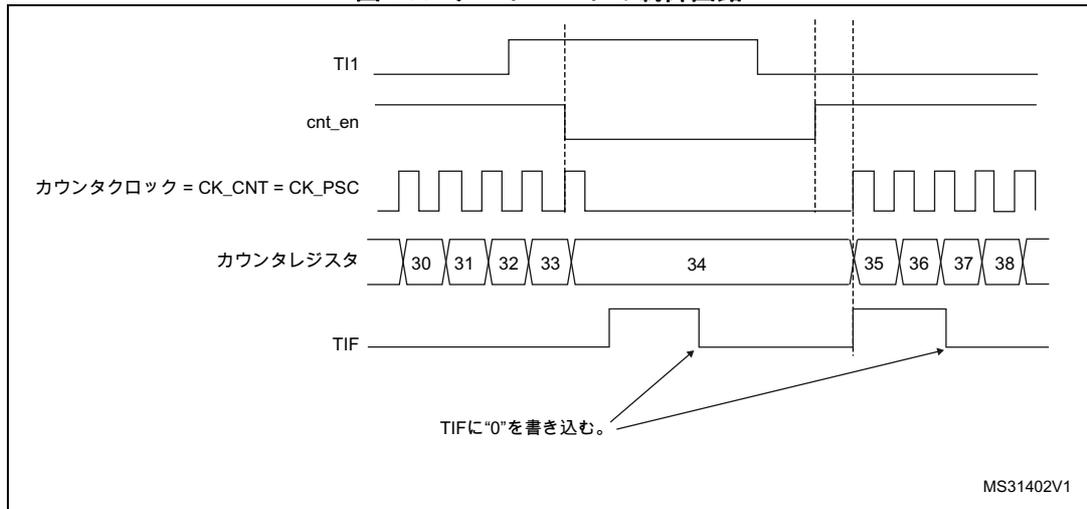
次の例では、アップカウンタは TI1 入力が高いときだけカウントします。

1. TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S = 01)。TIMx\_CCER レジスタの CC1P ビットに“1”、CC1NP ビットに“0”を書き込んで、極性を有効にします (その後、立ち上がりエッジのみを検出)。
2. TIMx\_SMCR レジスタの SMS ビットに“101”を書き込むことによって、タイマをゲートモードに設定します。TIMx\_SMCR レジスタの TS ビットに“101”を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタの CEN ビットに“1”を書き込んで、カウンタを有効にします (ゲートモードでは、CEN = 0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx\_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 260. ゲートモードの制御回路



## スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

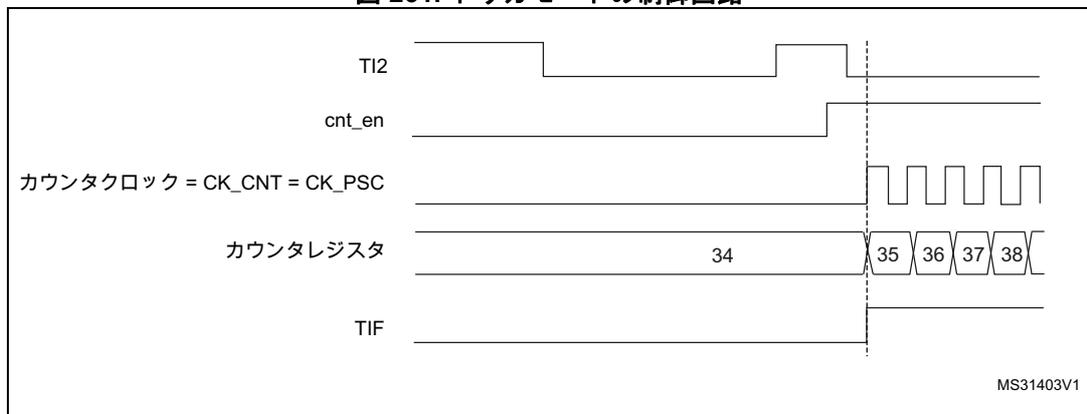
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

1. TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx\_CCMR1 レジスタの CC2S = 01)。TIMx\_CCER レジスタの CC2P ビットに“1”、CC2NP ビットに“0”を書き込んで、極性を有効にします (その後、ローレベルのみを検出)。
2. TIMx\_SMCR レジスタの SMS ビットに“110”を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタの TS ビットに“110”を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 261. トリガモードの制御回路



## 24.3.12 タイマ同期 (TIM9/12)

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 22.3.25 : タイマの同期 \(663 ページ\)](#) を参照してください。

**注：** スレープタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があり、マスタタイマからトリガを受信している間は動作中に変更しないでください。

## 24.3.13 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex<sup>®</sup>-M7 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 40.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#) を参照してください。

## 24.4 TIM9 および TIM12 レジスタ

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

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位で書き込む必要があります。読み出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位で行うことができます。

### 24.4.1 TIM9/12 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	CKD[1:0]		ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
						r/w	r/w	r/w				r/w	r/w	r/w	r/w

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デジタルフィルタ (Tix) によって使用されるサンプリングクロックとの間の分周比を示します。

- 00:  $t_{DTS} = t_{CK\_INT}$
- 01:  $t_{DTS} = 2 \times t_{CK\_INT}$
- 10:  $t_{DTS} = 4 \times t_{CK\_INT}$
- 11: 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0: TIMx\_ARR レジスタはバッファされません。
- 1: TIMx\_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

- 0: カウンタは更新イベントで停止しません。
- 1: カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

**ビット 2 URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされません。

0 : 次のイベントのいずれかが更新割り込みを生成します (有効な場合)。

- カウンタオーバーフロー
- UG ビットのセット

1 : カウンタオーバーフローのみが更新割り込みを生成します (有効な場合)。

**ビット 1 UDIS** : 更新ディセーブル

このビットは、更新イベント (UEV) 生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー
- UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

**ビット 0 CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

## 24.4.2 TIM9/12 のスレーブモード制御レジスタ (TIMx\_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MSM	TS[2:0]			Res.	SMS[2:0]									
								rw	rw	rw	rw		rw	rw	rw

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期する場合に役立ちます。

ビット 6:4 **TS** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)
- 100 : TI1 エッジ検出回路 (TI1F\_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)
- 111 : 予約済み

各タイマにとっての ITRx の意味の詳細については、表 133 : TIMx 内部トリガ接続 (803 ページ) を参照してください。

**注 :** 遷移時の誤ったエッジ検出を避けるために、これらのビットは必ず使用されていないとき (SMS = 000 のときなど) に変更してください。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力として選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

- 000 : スレーブモードは無効です。GEN = 1 の場合、プリスケアラは内部クロックによって直接クロック供給されます。
- 001 : 予約済み
- 010 : 予約済み
- 011 : 予約済み
- 100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
- 101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。
- 110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。
- 111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

**注 :** トリガ入力として TI1F\_ED が選択されている場合 (TS = 100)、ゲートモードを使用することはできません。TI1F\_ED は TI1F の遷移ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

**注 :** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 133. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = '000')	ITR1 (TS = '001')	ITR2 (TS = '010')	ITR3 (TS = '011')
TIM9	TIM2	TIM3	TIM10_OC	TIM11_OC
TIM12	TIM4	TIM5	TIM13_OC	TIM14_OC

## 24.4.3 TIM9/12 割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	TIE	Res	Res	Res	CC2IE	CC1IE	UIE								
									rw				rw	rw	rw

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IE** : キャプチャ/比較 2 割り込みイネーブル

0 : CC2 割り込みは無効です。

1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

## 24.4.4 TIM9/12 ステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ  
CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIF** : トリガ割り込みフラグ

このフラグは、トリガイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

0 : トリガイベントは発生していません。

1 : トリガ割り込みが保留中です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合 :**

このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。TIMx\_CCR1 の内容が TIMx\_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

**CC1 チャンネルが入力として設定されている場合 :**

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (選択された極性に一致するエッジが IC1 で検出されました)。

**ビット 0 UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx\_CR1 レジスタで UDIS = 0 の場合。
- TIMx\_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx\_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、トリガイベントによって CNT が再初期化されたとき (同期制御レジスタの説明を参照)。

## 24.4.5 TIM9/12 のイベント発生レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG								
									w				w	w	w

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

**ビット 6 TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx\_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みが発生します。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

**ビット 2 CC2G** : キャプチャ/比較 2 生成

CC1G の説明を参照してください。

**ビット 1 CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

**CC1 チャンネルが出力として設定されている場合 :**

CC11F フラグがセットされ、対応する割り込みが送信されます (有効な場合)。

**CC1 チャンネルが入力として設定されている場合 :**

現在のカウンタ値が TIMx\_CCR1 レジスタにキャプチャされます。CC11F フラグがセットされ、対応する割り込みが送信されます (有効な場合)。CC11F フラグがすでにハイの場合、CC10F フラグがセットされます。

**ビット 0 UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラのカウンタもクリアされます (分周比は変化しません)。カウンタはクリアされます。



## 24.4.6 TIM9/12 のキャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のすべてのビットは、入力モードと出力モードで機能が異なります。特定のビットについて、OCxx は、チャンネルが出力モードに設定されているときの機能を記述し、ICxx は、チャンネルが入力モードに設定されているときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

### 出力比較モード

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC2S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E = 0) のときにのみ書き込み可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **OC1M** : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは、それぞれ CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、TIMx\_CNT カウンタがキャプチャ/比較レジスタ 1 (TIMx\_CCR1) に一致したとき、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、TIMx\_CNT カウンタがキャプチャ/比較レジスタ 1 (TIMx\_CCR1) に一致したとき、強制的にローになります。

011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

100 : 強制インアクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はインアクティブ (OC1REF = 0) に、そうでない場合はアクティブ (OC1REF = 1) になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

**注 :** PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注 :** PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 は、トリガがオンのときでも、カウンタと CCR1 の値に依存して、通常どおりに動作します。トリガ入力にエッジが発生してから CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E = 0) のときにのみ書き込み可能です。

## 入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケーラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC2S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E = 0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 :  $f_{DTS}1000$ :  $f_{SAMPLING}=f_{DTS}/8$ , N=6 ではフィルタサンプリングは行われません。

0001 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=21001 :  $f_{SAMPLING}=f_{DTS}/8$ , N=8

0010 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=41010 :  $f_{SAMPLING}=f_{DTS}/16$ , N=5

0011 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=81011 :  $f_{SAMPLING}=f_{DTS}/16$ , N=6

0100 :  $f_{SAMPLING}=f_{DTS}/2$ , N=61100 :  $f_{SAMPLING}=f_{DTS}/16$ , N=8

0101 :  $f_{SAMPLING}=f_{DTS}/2$ , N=81101 :  $f_{SAMPLING}=f_{DTS}/32$ , N=5

0110 :  $f_{SAMPLING}=f_{DTS}/4$ , N=61110 :  $f_{SAMPLING}=f_{DTS}/32$ , N=6

0111 :  $f_{SAMPLING}=f_{DTS}/4$ , N=81111 :  $f_{SAMPLING}=f_{DTS}/32$ , N=8

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケーラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケーラの比を定義します。

プリスケーラは、CC1E = 0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケーラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC1S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E = 0) のときにのみ書き込み可能です。

## 24.4.7 TIM9/12 のキャプチャ／比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E							
								rW		rW	rW	rW		rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性  
CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ／比較 2 出力極性  
CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ／比較 2 出力イネーブル  
CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ／比較 1 相補出力極性  
CC1 チャンネルが出力として設定されている場合 : CC1NP はクリア状態に維持する必要があります。  
CC1 チャンネルが入力として設定されている場合 : CC1NP ビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性  
**CC1 チャンネルが出力として設定されている場合 :**  
0 : OC1 はアクティブハイです。  
1 : OC1 はアクティブローです。  
**CC1 チャンネルが入力として設定されている場合 :**  
CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。  
00 : 非反転／立ち上がりエッジ  
回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。  
01 : 反転／立ち下がりエッジ  
回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。  
10 : 予約済み。この設定は使用しないでください。

**注 : 11 : 非反転／両エッジ**  
*回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。*

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル  
**CC1 チャンネルが出力として設定されている場合 :**  
0 : オフ - OC1 はアクティブではありません。  
1 : オン - OC1 信号は、対応する出力ピンに出力されます。  
**CC1 チャンネルが入力として設定されている場合 :**  
このビットによって、カウンタ値のキャプチャ／比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。  
0 : キャプチャは無効です。  
1 : キャプチャは有効です。



表 134. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx = 0、OCx_EN = 0)
1	OCx = OCxREF + 極性、OCx_EN = 1

注：標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

## 24.4.8 TIM9/12 のカウンタ (TIMx\_CNT)

アドレスオフセット：0x24

リセット値：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 CNT[15:0]：カウンタ値

## 24.4.9 TIM9/12 プリスケーラ (TIMx\_PSC)

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 PSC[15:0]：プリスケーラ値

カウンタクロック周波数 CK\_CNT は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブなプリスケーラレジスタにロードされる値を含みます。

## 24.4.10 TIM9/12 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット：0x2C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 ARR[15:0]:自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 24.3.1：タイムベースユニット \(781 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 24.4.11 TIM9/12 キャプチャ/比較モードレジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値

**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、プリロード値がロードされたままになります。そうでない場合、プリロード値は、更新イベントが発生すると、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブなキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**チャンネル CC1 が入力として設定されている場合 :**

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 24.4.12 TIM9/12 キャプチャ/比較モードレジスタ 2 (TIMx\_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR2[15:0]** : キャプチャ/比較 2 値

**CC2 チャンネルが出力として設定されている場合 :**

CCR2 は、実際のキャプチャ/比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、プリロード値がロードされたままになります。そうでない場合、プリロード値は、更新イベントが発生すると、アクティブキャプチャ/比較 2 レジスタにコピーされます。

アクティブなキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC2 出力に送信される値を含みます。

**CC2 チャンネルが入力として設定されている場合 :**

CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

## 24.4.13 TIM9/12 レジスタマップ

TIM9/12 のレジスタは、次のように 16 ビットのアドレス可能レジスタとしてマップされます。

**表 135. TIM9/12 レジスタマップとリセット値**

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	TIMx_CR1 リセット値	Res	CKD [1:0]	ARPE	Res	Res	Res	OPM	URS	UDIS	CEN																										
0x08	TIMx_SMCR リセット値	Res	MSM	TS[2:0]			SMS[2:0]																														
0x0C	TIMx_DIER リセット値	Res	Res	TIE	Res	Res	Res	Res	Res	CC2IE	CC1IE	UIE																									
0x10	TIMx_SR リセット値	Res	Res	Res	TIF	Res	Res	Res	Res	Res	CC2IF	CC1IF	UIF																								
0x14	TIMx_EGR リセット値	Res	Res	Res	TG	Res	Res	Res	Res	Res	CC2G	CC1G	UG																								
0x18	TIMx_CCMR1 出力比較モード リセット値	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																							
	TIMx_CCMR1 入力キャプチャ モード リセット値	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								
0x1C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								
0x20	TIMx_CCER リセット値	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								
0x24	TIMx_CNT リセット値	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								
0x28	TIMx_PSC リセット値	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								
0x2C	TIMx_ARR リセット値	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								
0x30	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																								

表 135. TIM9/12 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	TIMx_CCR1	Res.	CCR1[15:0]																														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIMx_CCR2	Res.	CCR2[15:0]																														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C~ 0x4C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																

レジスタ境界アドレスについては、[セクション 2.2.2: メモリマップとレジスタ境界アドレス \(66 ページ\)](#) を参照してください。

## 24.5 TIM10/11/13/14 レジスタ

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位で書き込む必要があります。読み出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位で行うことができます。

### 24.5.1 TIM10/11/13/14 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	Res.	URS	UDIS	CEN
						r/w	r/w	r/w					r/w	r/w	r/w

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デジタルフィルタ (Tix) によって使用されるサンプリングクロックとの間の分周比を示します。

- 00:  $t_{DTS} = t_{CK\_INT}$
- 01:  $t_{DTS} = 2 \times t_{CK\_INT}$
- 10:  $t_{DTS} = 4 \times t_{CK\_INT}$
- 11: 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0: TIMx\_ARR レジスタはバッファされません。
- 1: TIMx\_ARR レジスタはバッファされます。

ビット 6:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **URS** : 更新リクエストソース

このビットは、更新割り込み (UEV) ソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0: 次のイベントのいずれかが UEV を発生します (有効な場合)。
  - カウンタオーバーフロー
  - UG ビットのセット
- 1: カウンタオーバーフローのみが UEV を発生します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、更新割り込み (UEV) イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0: UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。
  - カウンタオーバーフロー
  - UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

- 1: UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR、PSC、CCRx) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0: カウンタは無効です。
- 1: カウンタは有効です。

## 24.5.2 TIM10/11/13/14 割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC1IE	UIE													
														rw	rw

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

## 24.5.3 TIM10/11/13/14 ステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	CC1IF	UIF						
						rc_w0								rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合 :**

このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。TIMx\_CCR1 の内容が TIMx\_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

**CC1 チャンネルが入力として設定されている場合 :**

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (選択された極性に一致するエッジが IC1 で検出されました)。

**ビット 0 UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx\_CR1 レジスタで UDIS = 0 の場合。
- TIMx\_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

## 24.5.4 TIM10/11/13/14 のイベント発生レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC1G	UG													
														w	w

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

**CC1 チャンネルが出力として設定されている場合 :**

CC1IF フラグがセットされ、対応する割り込みが送信されます (有効な場合)。

**CC1 チャンネルが入力として設定されている場合 :**

カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

**ビット 0 UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。カウンタはクリアされます。

## 24.5.5 TIM10/11/13/14 のキャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]									
Res.	IC1F[3:0]			IC1PSC[1:0]											
								rW	rW	rW	rW	rW	rW	rW	rW

## 出力比較モード

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **OC1M** : 出力比較 1 モード

これらのビットは、OC1 の元となる出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 のアクティブレベルは CC1P ビットに依存します。

000 : 停止。出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

111 : PWM モード 2 - チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

**注 :** PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときに、OCREF のレベルが変化します。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注 :** PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 :

11 :

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 :  $f_{DTS}1000$ :  $f_{SAMPLING}=f_{DTS}/8$ , N=6 ではフィルタサンプリングは行われません。

0001 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=21001 :  $f_{SAMPLING}=f_{DTS}/8$ , N=8

0010 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=41010 :  $f_{SAMPLING}=f_{DTS}/16$ , N=5

0011 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=81011 :  $f_{SAMPLING}=f_{DTS}/16$ , N=6

0100 :  $f_{SAMPLING}=f_{DTS}/2$ , N=61100 :  $f_{SAMPLING}=f_{DTS}/16$ , N=8

0101 :  $f_{SAMPLING}=f_{DTS}/2$ , N=81101 :  $f_{SAMPLING}=f_{DTS}/32$ , N=5

0110 :  $f_{SAMPLING}=f_{DTS}/4$ , N=61110 :  $f_{SAMPLING}=f_{DTS}/32$ , N=6

0111 :  $f_{SAMPLING}=f_{DTS}/4$ , N=81111 :  $f_{SAMPLING}=f_{DTS}/32$ , N=8

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : 予約済み

11 : 予約済み

**注:** **CC1S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

## 24.5.6 TIM10/11/13/14 のキャプチャ/比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CC1NP	Res.	CC1P	CC1E											
												rw		rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 : CC1NP はクリアされたままにする必要があります。  
 CC1 チャンネルが入力として設定されている場合 : CC1NP ビットは、TI1FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

**CC1 チャンネルが出力として設定されている場合 :**

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :**

CC1P ビットは、トリガ動作またはキャプチャ動作での TI1FP1 および TI2FP1 の極性を選択します。

00 : 非反転/立ち上がりエッジ

回路は TI1FP1 の立ち上がりエッジに反応し (キャプチャモード)、TI1FP1 は反転されません。

01 : 反転/立ち下がりエッジ

回路は TI1FP1 の立ち下がりエッジに反応し (キャプチャモード)、TI1FP1 は反転されます。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ

回路は TI1FP1 の立ち上がりエッジに反応し (キャプチャモード)、TI1FP1 は反転されません。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル。

**CC1 チャンネルが出力として設定されている場合 :**

0 : オフ - OC1 はアクティブではありません。

1 : オン - OC1 信号は、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :**

このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

表 136. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx=0, OCx_EN=0)
1	OCx = OCxREF + 極性、OCx_EN = 1

注 : 標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

## 24.5.7 TIM10/11/13/14 のカウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 CNT[15:0] : カウンタ値

## 24.5.8 TIM10/11/13/14 プリスケーラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 PSC[15:0] : プリスケーラ値

カウンタクロック周波数 CK\_CNT は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケーラレジスタにロードされる値を含みます。

## 24.5.9 TIM10/11/13/14 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 ARR[15:0] : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

APR の更新と動作の詳細については、[セクション 24.3.1 : タイムベースユニット \(781 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 24.5.10 TIM10/11/13/14 キャプチャ/比較レジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値

**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**チャンネル CC1 が入力として設定されている場合 :**

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 24.5.11 TIM11 オプションレジスタ 1 (TIM11\_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM11_RMP[1:0]														
															rw

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TI1\_RMP[1:0]** : TIM11 入力 1 再配置機能

ソフトウェアでセット/クリアされます。

00 : TIM11 のチャンネル 1 は、GPIO に接続されます (代替機能配置を参照)。

01 : SPDIFRX フレーム同期

10 : HSE 内部クロック (RTC では 1 MHz) は、測定のため TIM11\_CH1 入力に接続されます。

11 : MCO1 が TIM11\_CH1 入力に接続されます。

## 24.5.12 TIM10/11/13/14 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットのアドレス可能レジスタとしてマップされます。

表 137. TIM10/11/13/14 レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	TIMx_CR1	Res	CKD [1:0]	ARPE	Res	Res	Res	Res	Res	URS	UDIS	CEN																								
	リセット値																							0	0	0					0	0	0			
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res																										
	リセット値																																			
0x0C	TIMx_DIER	Res	Res	Res	Res	Res	Res	Res	Res	CC1IE	UIE																									
	リセット値																															0	0			
0x10	TIMx_SR	Res	CC1OF	Res	Res	Res	Res	Res	Res	CC1IF	UIF																									
	リセット値																								0							0	0			
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	CC1G	UG																									
	リセット値																															0	0			
0x18	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	OC1M [2:0]	OC1PE	OC1FE	CC1S [1:0]																							
	リセット値																														0	0	0	0	0	0
	TIMx_CCMR1 入力キャプチャモード	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC1PSC [1:0]	CC1S [1:0]																								
	リセット値																													0	0	0	0	0	0	
0x1C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																									
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC1NP	CC1P	CC1E																						
	リセット値																															0	0	0		
0x24	TIMx_CNT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																							
	リセット値																																			
0x28	TIMx_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																							
	リセット値																																			
0x2C	TIMx_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																							
	リセット値																																			
0x30	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res																							

表 137. TIM10/11/13/14 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	TIMx_CCR1	Res.	CCR1[15:0]																														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38~ 0x4C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																	
0x50	TIMx_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																	
	リセット値																																0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 25 基本タイマ (TIM6/TIM7)

### 25.1 TIM6/TIM7 の概要

基本タイマ TIM6 および TIM7 は、プログラマブルなプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

このタイマは、タイムベース生成を目的とした汎用タイマとして使用できますが、特にデジタルアナログコンバータ (DAC) の駆動にも使用されます。実際、これらのタイマは内部で DAC に接続されており、トリガ出力を通じて駆動できます。

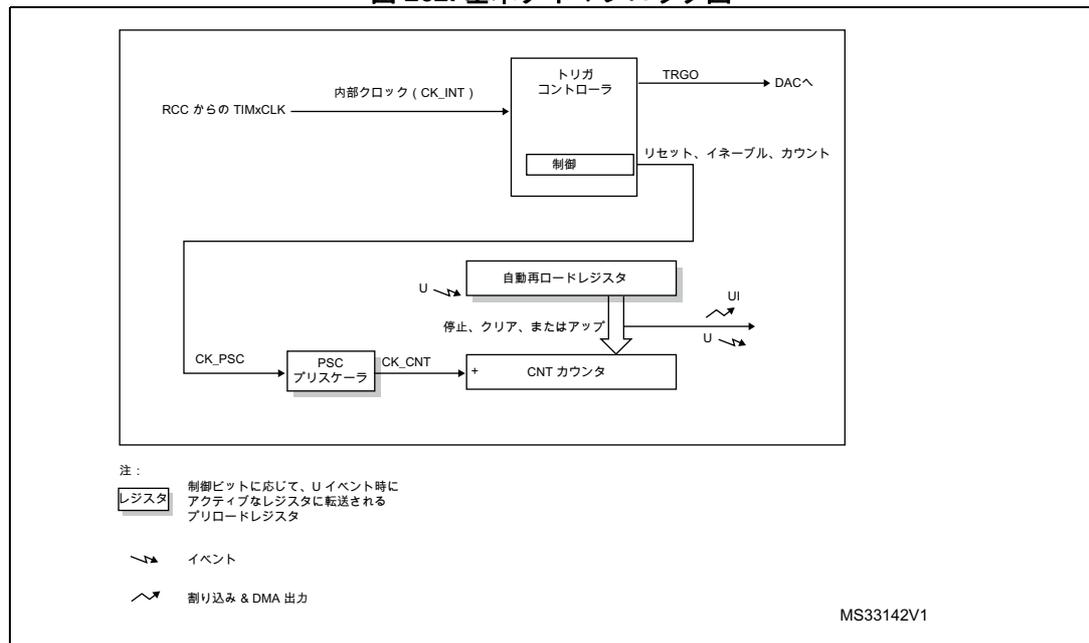
タイマは完全に独立していて、いかなるリソースも共有しません。

### 25.2 TIM6/TIM7 の主な機能

基本タイマ (TIM6/TIM7) の機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケラ (動作中も変更可能) で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- DAC をトリガする同期回路
- 更新イベント時の割り込み/DMA 生成 : カウンタオーバーフロー

図 262. 基本タイマブロック図



## 25.3 TIM6/TIM7 の機能詳細

### 25.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC)
- 自動再ロードレジスタ (TIMx\_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きはプリロードレジスタへのアクセスとなります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント UEV ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 である場合、カウンタがオーバーフロー値に達すると、更新イベントが送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときのみ、プリスケアラ出力 CK\_CNT から供給されます。

実際のカウンタイネーブル信号 CNT\_EN は、CEN の 1 クロックサイクル後にセットされます。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。TIMx\_PSC 制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 263](#) と [図 264](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 263. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

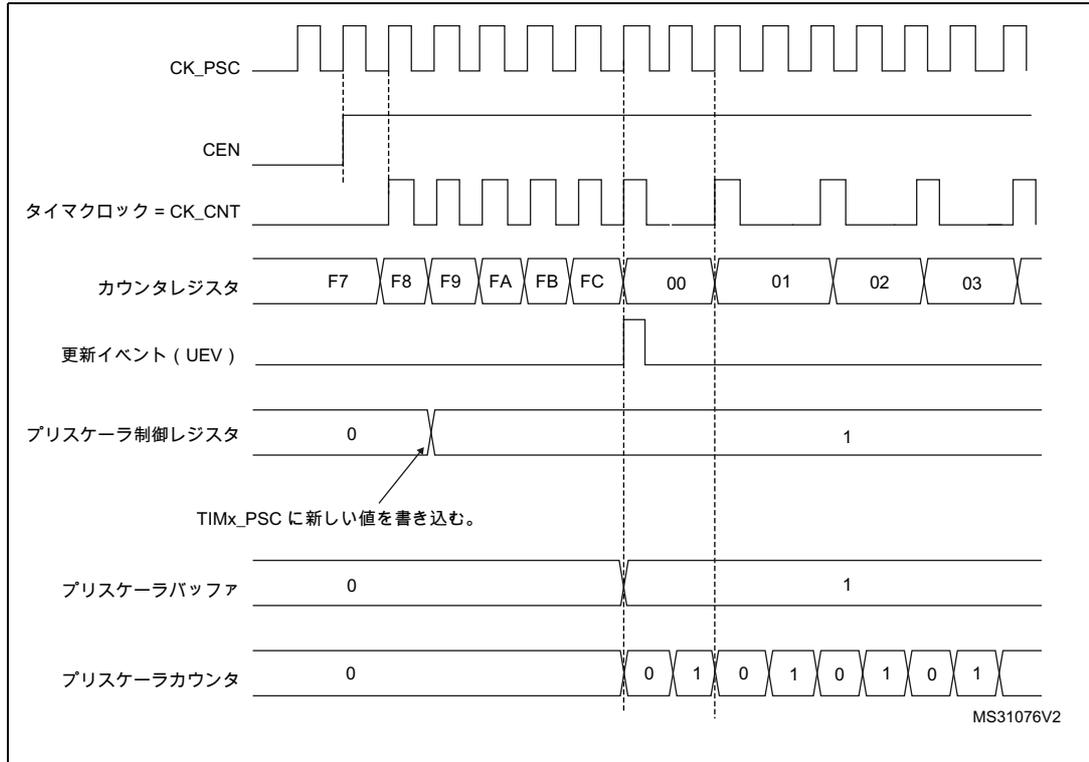
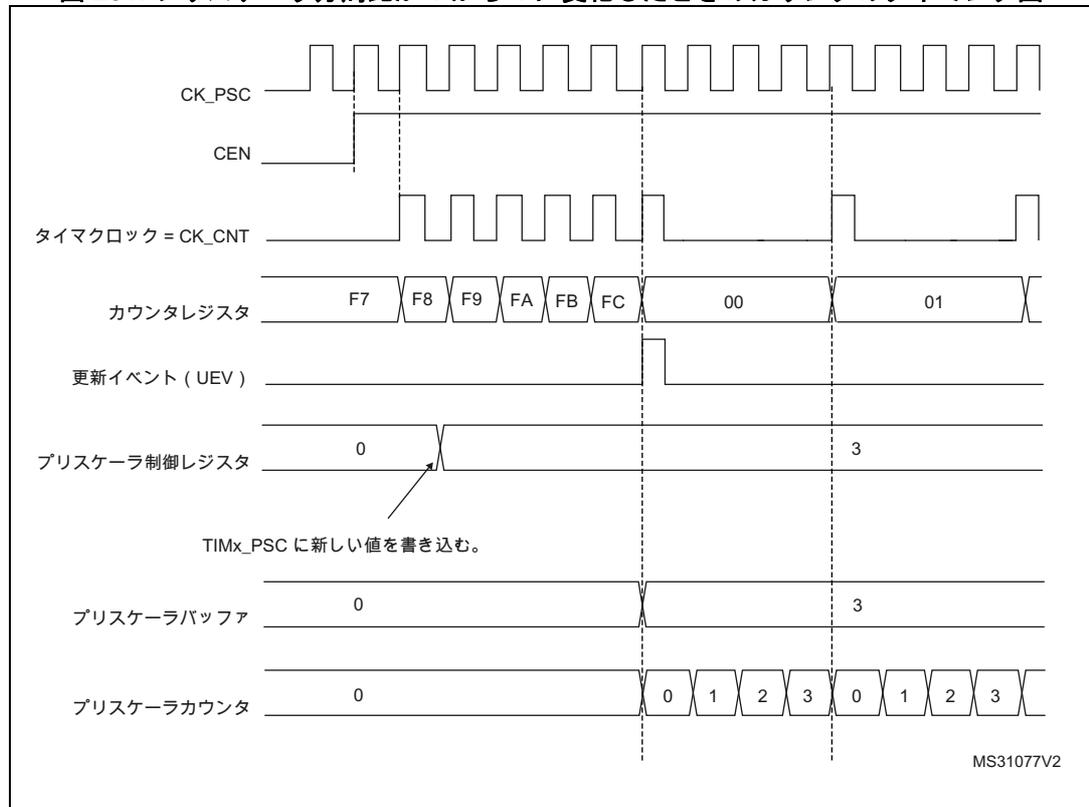


図 264. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 25.3.2 カウントモード

カウンタは、0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントした後、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、TIMx\_EGR レジスタの UG ビットをセットすることによって (ソフトウェアで、または、スレーブモードコントローラを使用して) 生成できます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。このようにすると、UDIS ビットに 0 が書き込まれるまで更新イベントは発生しませんが、カウンタとプリスケアラカウンタは両方とも 0 からリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS (更新リクエスト選択) ビットがセットされている場合、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (割り込みや DMA リクエストは送信されません)。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットの設定に応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 265. 内部クロック分周比が 1 の場合のカウンタのタイミング図

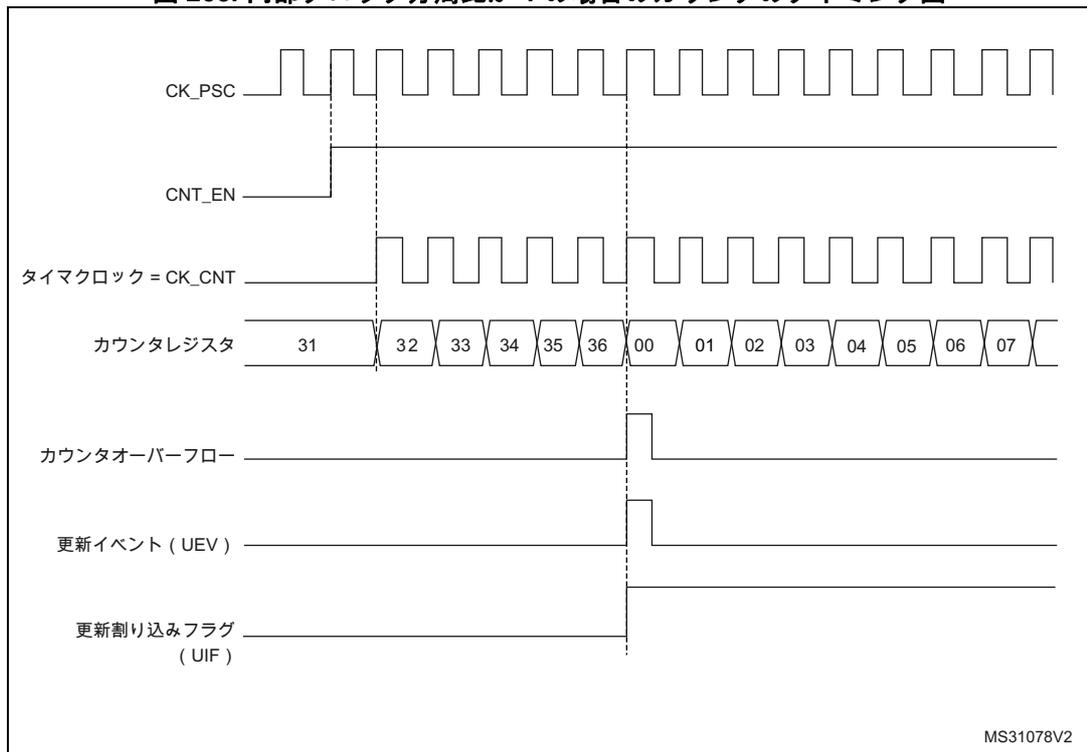


図 266. 内部クロック分周比が 2 の場合のカウンタのタイミング図

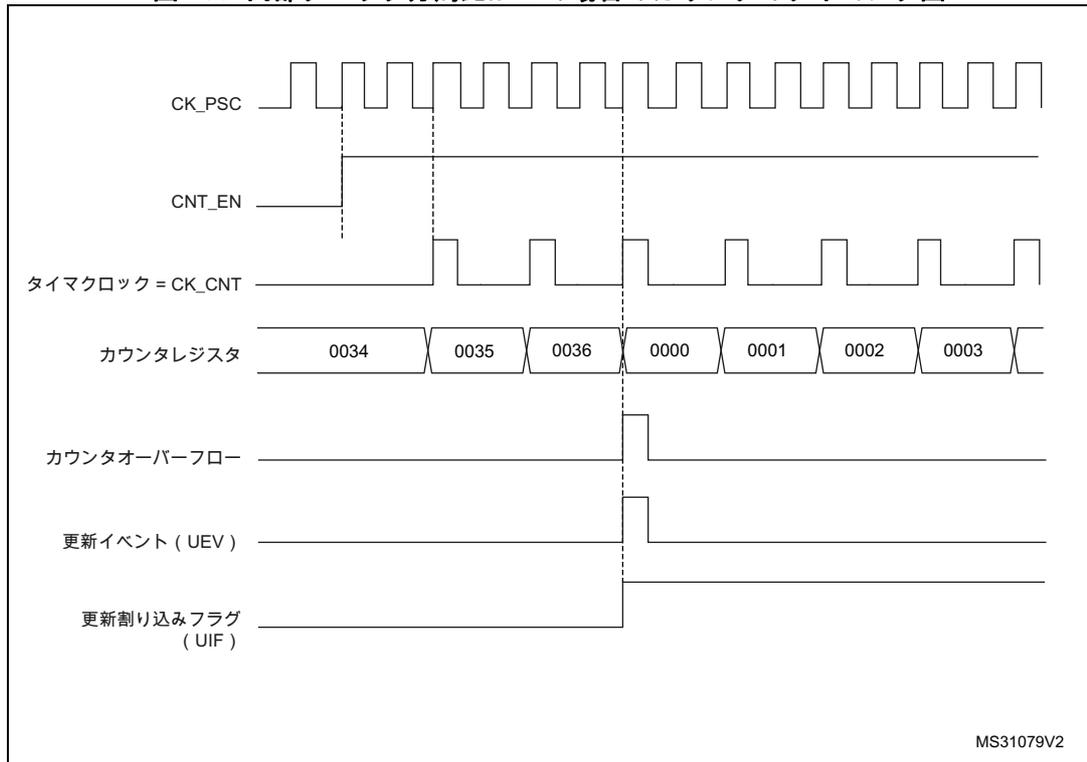


図 267. 内部クロック分周比が 4 の場合のカウンタのタイミング図

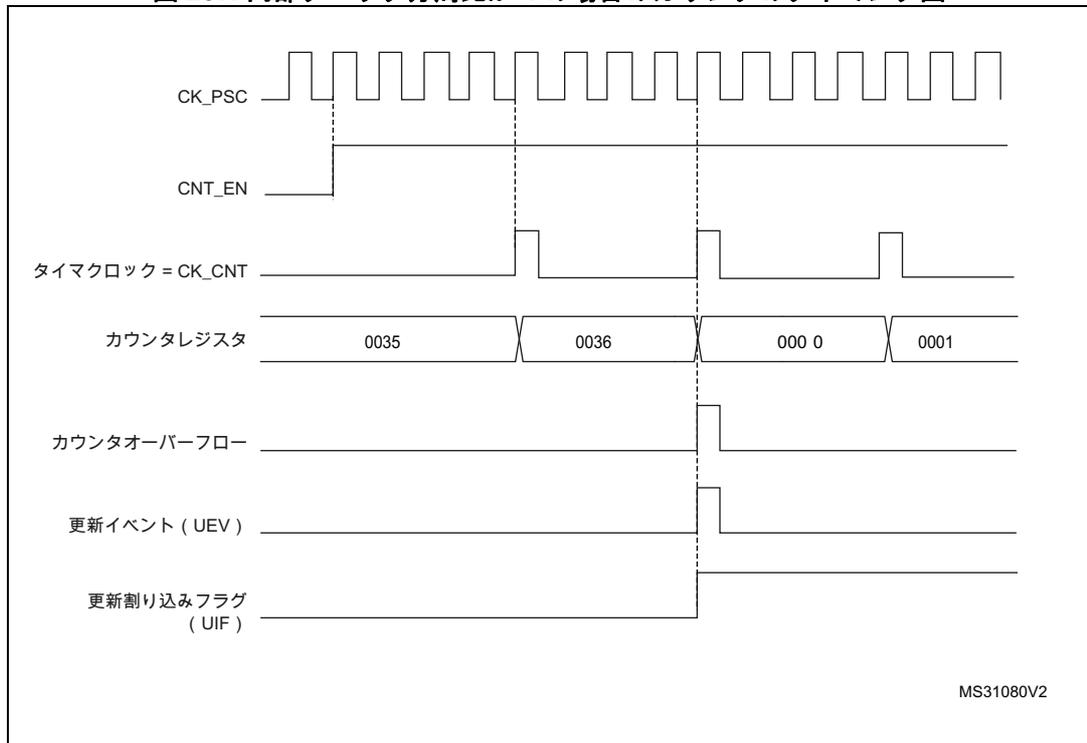


図 268. 内部クロック分周比が N の場合のカウンタのタイミング図

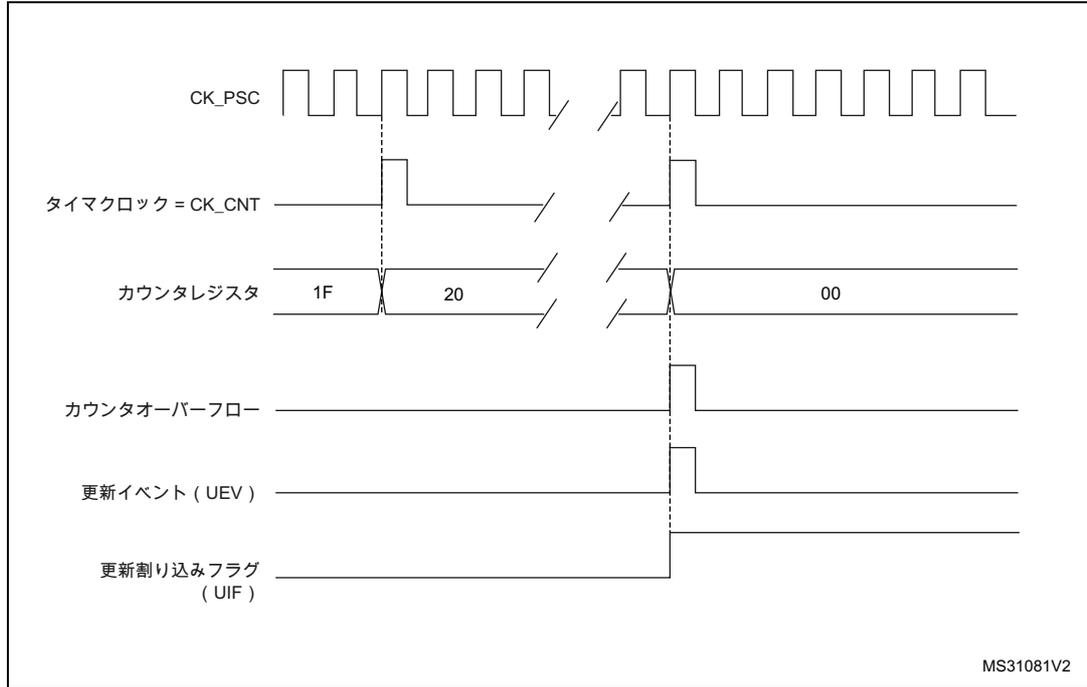


図 269. ARPE=0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

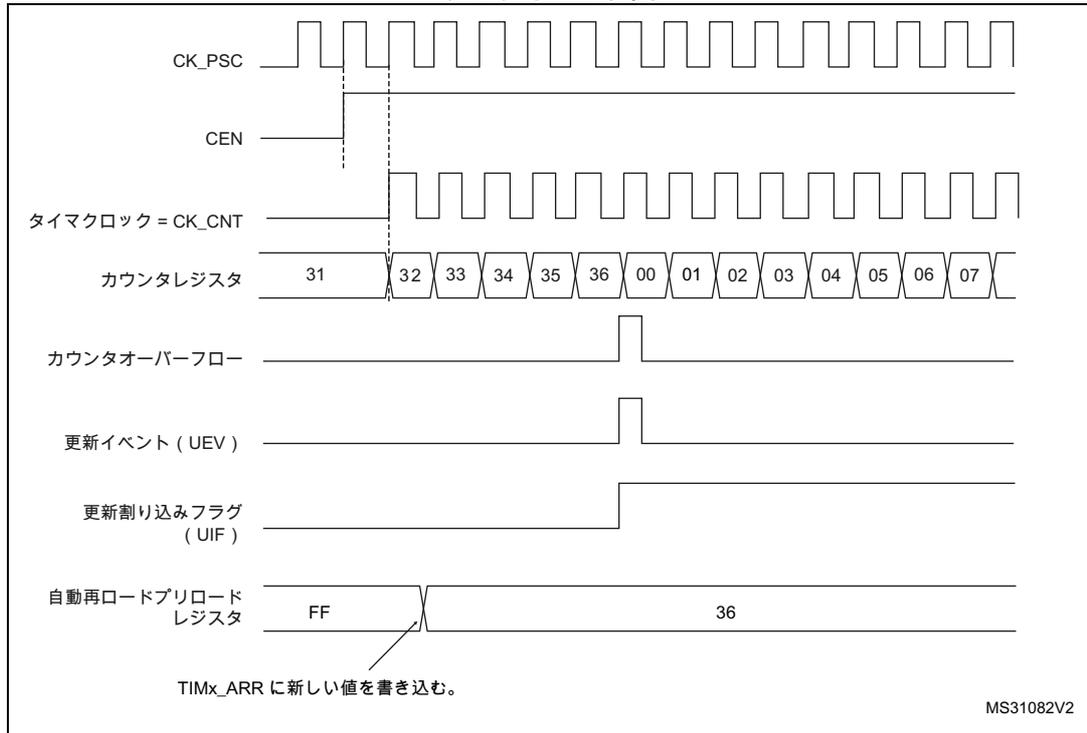
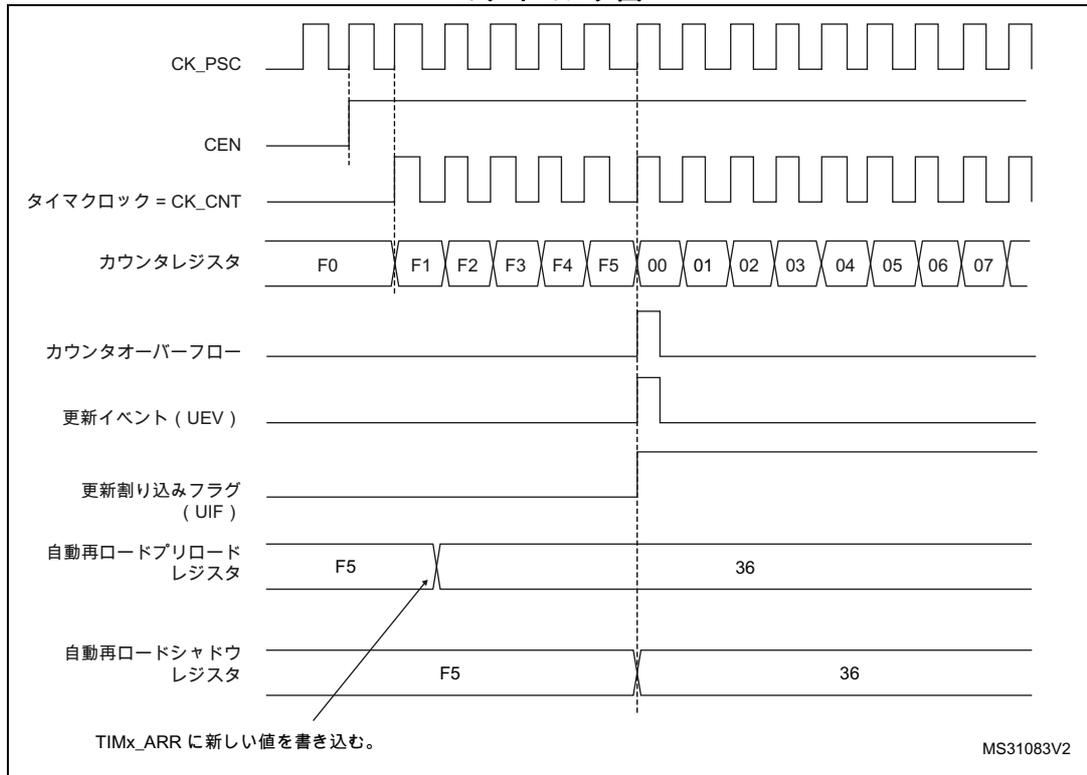


図 270. ARPE = 1 (TIMx\_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



### 25.3.3 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。特定のケースでは、バックグラウンドタスク (カウンタの読み出し) と割り込み (更新の割り込み) との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

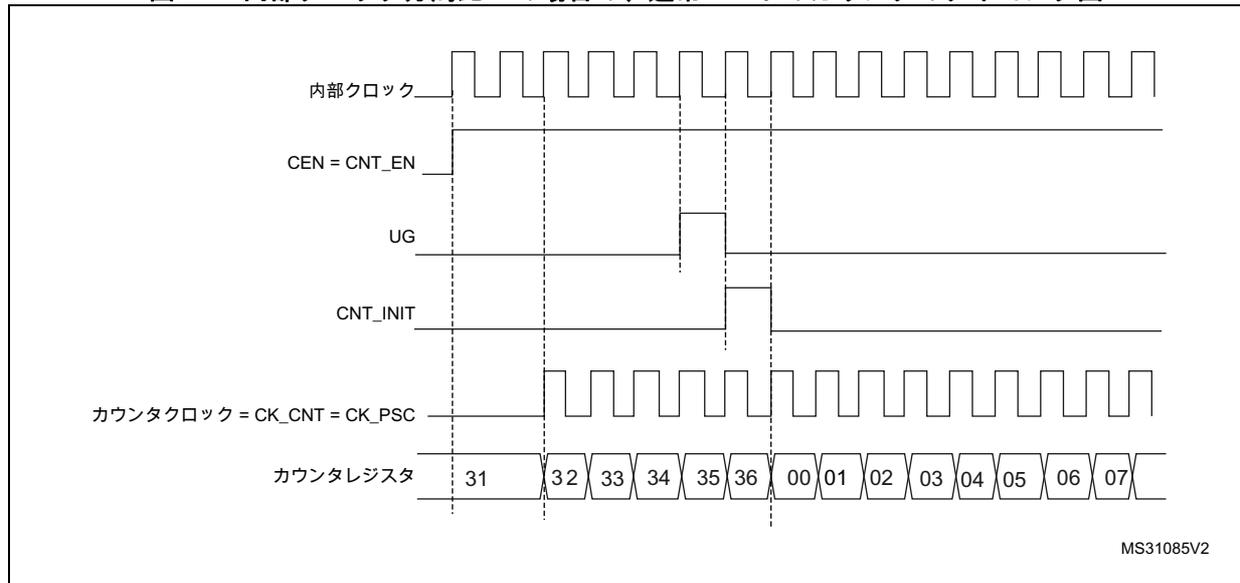
### 25.3.4 クロックソース

カウンタクロックは、内部クロック (CK\_INT) ソースから供給されます。

TIMx\_CR1 レジスタの CEN ビットと TIMx\_EGR レジスタの UG ビットは実際の制御ビットであり、ソフトウェアによってのみ変更できます (ただし、自動的にクリア状態が保持される UG ビットを除く)。CEN ビットに 1 が書き込まれると、プリスケラにはクロックとして内部クロック CK\_INT が供給されます。

図 271 に、プリスケラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

図 271. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



## 25.3.5 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex<sup>®</sup>-M7 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 40.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 25.4 TIM6/TIM7 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#)を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 25.4.1 TIM6/TIM7 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF RE-MAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw				rw				rw	rw	rw	rw

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成しません (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC) はそれぞれの値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

**注 :** ゲートモードは、CEN ビットが事前にソフトウェアでセットされている場合にのみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

## 25.4.2 TIM6/TIM7 制御レジスタ 2 (TIMx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MMS[2:0]			Res.	Res.	Res.	Res.								
									rw	rw	rw				

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **MMS** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (TRGO) を選択するために使用します。組み合わせは、次のとおりです。

000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが発生したとき (スレーブモードコントローラがリセットモードに設定されているとき) TRGO 信号は実際のリセットから遅れて発生します。

001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除いて、TRGO に遅延が存在します (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

**注 :** **スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間は動作中に変更しないでください。**

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 25.4.3 TIM6/TIM7 DMA/割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UDE	Res.	UIE												
							rw								rw

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

## 25.4.4 TIM6/TIM7 のステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UIF														
															rc_w0

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- TIMx\_CR1 レジスタの UDIS = 0 であり、繰り返しカウンタ値でオーバーフローまたはアンダーフローが発生したとき。
- TIMx\_CR1 レジスタの URS = 0 かつ UDIS = 0 の場合に、TIMx\_EGR レジスタの UG ビットを使用して、ソフトウェアで CNT が再初期化されたとき。

## 25.4.5 TIM6/TIM7 のイベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UG														
															w

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : タイマカウンタを再初期化して、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

## 25.4.6 TIM6/TIM7 のカウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

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

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー。TIMx\_CR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

## 25.4.7 TIM6/TIM7 プリスケアラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 CK\_CNT は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブなプリスケアラレジスタにロードされる値を含みます。

## 25.4.8 TIM6/TIM7 の自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **ARR[15:0]** : プリスケアラ値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 25.3.1: タイムベースユニット \(825 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 25.4.9 TIM6/TIM7 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 138. TIM6/TIM7 レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIMx_CR1	Res.	UIFREMAP	Res.	OPM	URS	UDIS	CEN																										
	リセット値																						0				0			0	0	0	0	
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値																										0	0	0					
0x08	予約済み																																	
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値																																	
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値																																	
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値																																	
0x18-0x20	予約済み																																	
0x24	TIMx_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値	0																																
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値																																	
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値																																	

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 26 低電力タイマ (LPTIM)

### 26.1 概要

LPTIM は、消費電力削減の究極的な進展を利用した 16 ビットタイマです。クロックソースの多様性により、LPTIM は STANDBY モードを除くすべての電力モードで実行し続けることができます。内部クロックソースがなくても実行できるため、LPTIM は、一部のアプリケーションで役立つ「パルスカウンタ」として使用することができます。また、LPTIM はシステムを低電力モードからウェイクアップできるため、消費電力が極端に低い「タイムアウト機能」の実現に適しています。

LPTIM の柔軟性の高いクロック方式は、必要な機能性とパフォーマンスを提供しながら、消費電力を最小化します。

### 26.2 LPTIM の主な機能

- 16 ビットアップカウンタ
- 8 つの分周比 (1、2、4、8、16、32、64、128) を持つ 3 ビットプリスケアラ
- 選択可能なクロック
  - 内部クロックソース : LSE、LSI、HSI、または APB クロック
  - ULPTIM 入力経由の外部クロック (LP オシレータが実行していないときに動作、パルスカウンタアプリケーションによって使用)
- 16 ビット ARR 自動再ロードレジスタ
- 16 ビット比較レジスタ
- 連続/ワンショットモード
- 選択可能なソフトウェア/ハードウェア入力トリガ
- プログラム可能なデジタルグリッチフィルタ
- 設定可能な出力 : パルス、PWM
- 設定可能な I/O 極性
- エンコーダモード

### 26.3 LPTIM の実装

表 139 には、STM32F75xxx および STM32F74xxx デバイスへの LPTIM の実装について記載されています。LPTIM1 にはその機能のすべてが実装されています。LPTIM2 でサポートされている機能は LPTIM1 よりも少ないですが、他の点では全く同等です。

表 139. STM32F75xxx および STM32F74xxx LPTIM の機能

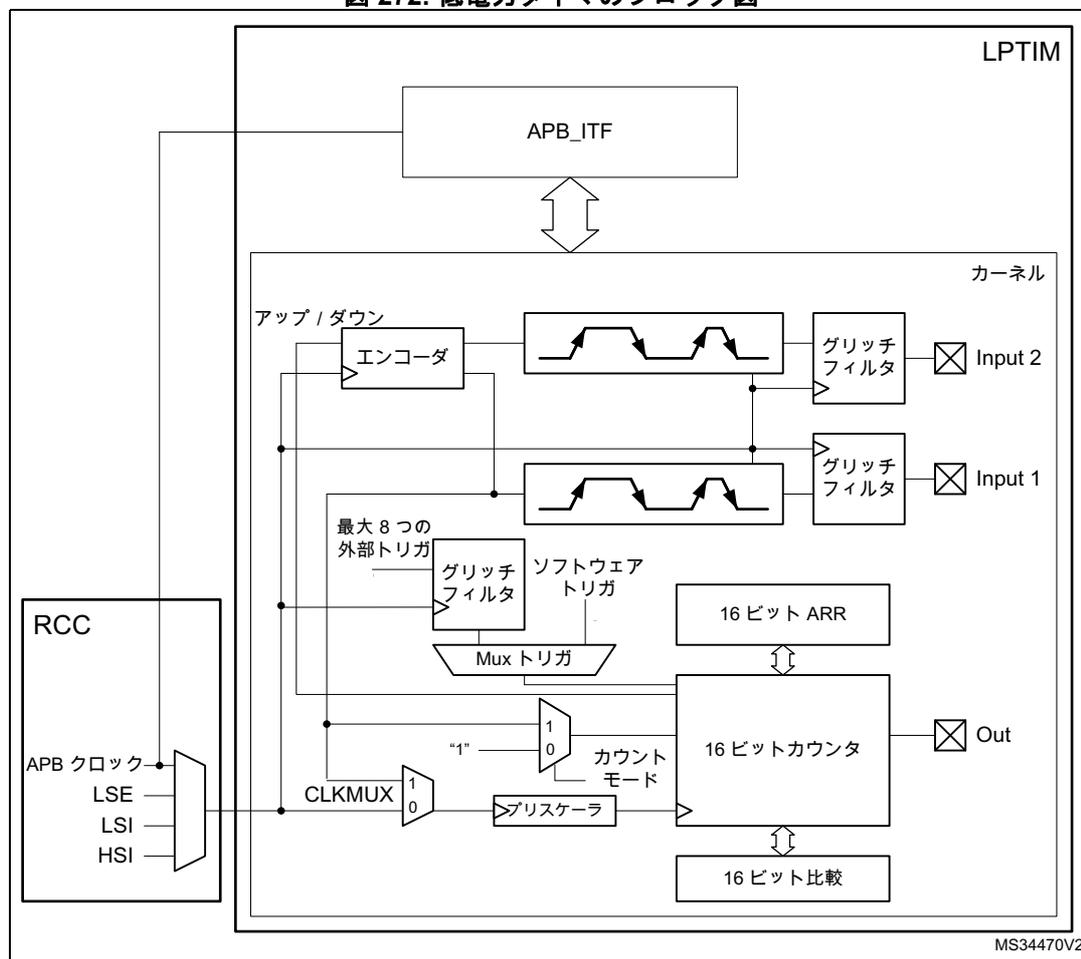
LPTIM モード/機能 <sup>(1)</sup>	LPTIM1	LPTIM2
エンコーダモード	X	-

1. X : サポートされています。

## 26.4 LPTIM の機能詳細

### 26.4.1 LPTIM ブロック図

図 272. 低電力タイマのブロック図



### 26.4.2 LPTIM のリセットとクロック

LPTIM のクロックには、いくつかのクロックソースを使用できます。クロックツリーコントローラ (RCC) によって APB、LSI、LSE、または HSI ソースから選択できる内部クロック信号を使用できます。また、LPTIM のクロックには、外部 Input1 に入力された外部クロック信号を使用することもできます。外部クロックソースでクロック供給される際には、LPTIM は、次の 2 つの構成のいずれかで動作します。

- 最初の構成では、LPTIM のクロックは外部信号によって供給されますが、同時に、APB またはその他の埋め込みオシレータ (LSE、LSI、HSI など) から内部クロック信号も LPTIM に供給されます。
- 2 番目の構成では、LPTIM のクロックは外部 Input1 を通じて外部クロックソースによってのみ供給されます。この構成は、低電力モードになった後、すべての埋め込みオシレータがオフになるときに、タイムアウト機能またはパルスカウンタ機能を実現するために使用されます。

CKSEL ビットおよび COUNTMODE ビットに書き込むことによって、LPTIM が外部クロックソースと内部クロックソースのいずれを使用するかを決めることができます。

外部クロックソースを使用するように設定されたときには、CKPOL ビットを使用して外部クロック信号のアクティブエッジを選択します。両方のエッジがアクティブとして設定された場合は、内部クロック信号も供給されます（最初の構成）。この場合、内部クロック信号の周波数は、外部クロック信号の周波数の 4 倍以上である必要があります。

### 26.4.3 グリッチフィルタ

LPTIM 入力（外部または内部）はデジタルフィルタによって保護され、グリッチとノイズの心配が LPTIM 内部に伝播されるのを防ぎます。これは、誤ったカウントまたはトリガを避けるためです。

デジタルフィルタをアクティブにする前に、内部クロックソースを LPTIM に供給する必要があります。これは、フィルタの正しい動作を保証するために必要です。

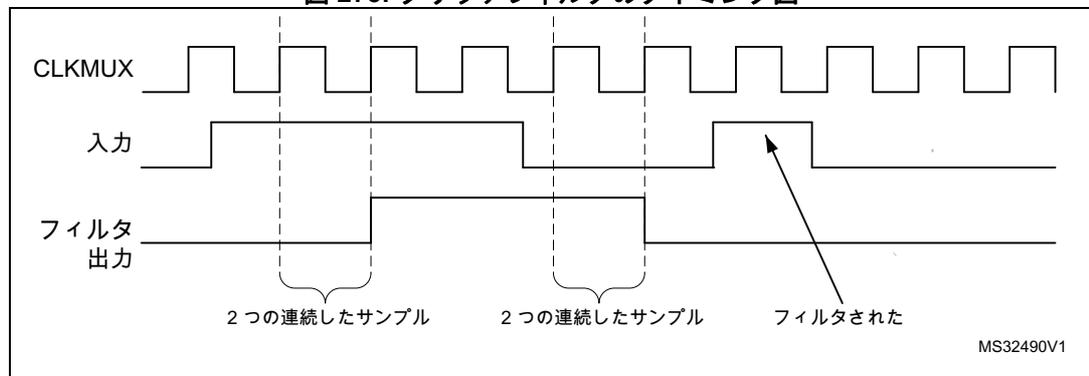
デジタルフィルタは、2 つのグループに分けられます。

- 最初のグループのデジタルフィルタは、LPTIM の外部入力を保護します。デジタルフィルタの感度は CKFLT ビットによって制御されます。
- 2 番目のグループのデジタルフィルタは、LPTIM の内部トリガ入力を保護します。デジタルフィルタの感度は TRGFLT ビットによって制御されます。

**注：** デジタルフィルタの感度は、グループごとに制御されます。同じグループ内の各デジタルフィルタの感度を個別に設定することはできません。

フィルタの感度は、信号のレベル変化を有効な遷移とみなすために、LPTIM 入力の 1 つで検出される連続した等しいサンプルの数に基づいて作用します。図 273 に、2 つの連続サンプルがプログラムされた場合のグリッチフィルタの動作例を示します。

図 273. グリッチフィルタのタイミング図



**注：** 内部クロック信号が供給されない場合は、CKFLT および TRGFLT ビットを 0 にセットすることによって、デジタルフィルタを無効にする必要があります。その場合、外部アナログフィルタを使用して、LPTIM の外部入力をグリッチから保護できます。

## 26.4.4 プリスケーラ

LPTIM 16 ビットカウンタの前には、設定可能な 2 のべき乗プリスケーラがあります。プリスケーラの分周比は PRESC[2:0] 3 ビットフィールドによって制御されます。下の表に、可能な分周比を示します。

表 140. プリスケーラの分周比

プログラミング	分周比
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

## 26.4.5 トリガマルチプレクサ

LPTIM カウンタは、ソフトウェアによって、または 8 つのトリガ入力の 1 つのアクティブエッジの検出後に開始できます。

LPTIM のトリガソースを決めるには、TRIGEN[1:0] が使用されます。

- TRIGEN[1:0] が 00 の場合、LPTIM カウンタは、CNTSTRT または SNGSTRT ビットがソフトウェアによってセットされるとすぐに開始します。
- TRIGEN[1:0] の残りの 3 つの可能な値は、トリガ入力によって使用されるアクティブエッジを設定するために使用されます。LPTIM カウンタは、アクティブエッジが検出されるとすぐに開始します。

TRIGEN[1:0] が 00 以外のときには、カウンタの開始に使用される 8 つのトリガ入力の 1 つを選択するために、TRIGSEL[2:0] が使用されます。

外部トリガは、LPTIM の場合、非同期信号とみなされます。したがって、同期のために、トリガ検出後、タイマが実行を開始するまでに 2 カウンタクロック周期の遅延が必要です。

タイマがすでに開始しているときに新しいトリガイベントが発生した場合、無視されます (タイムアウト機能が有効な場合を除きます)。

**注:** SNGSTRT/CNTSTRT ビットをセットする前に、タイマが有効になっている必要があります。タイマが無効なときにこれらのビットに書き込むと、ハードウェアによって破棄されます。

## 26.4.6 動作モード

LPTIM には 2 つの動作モードがあります。

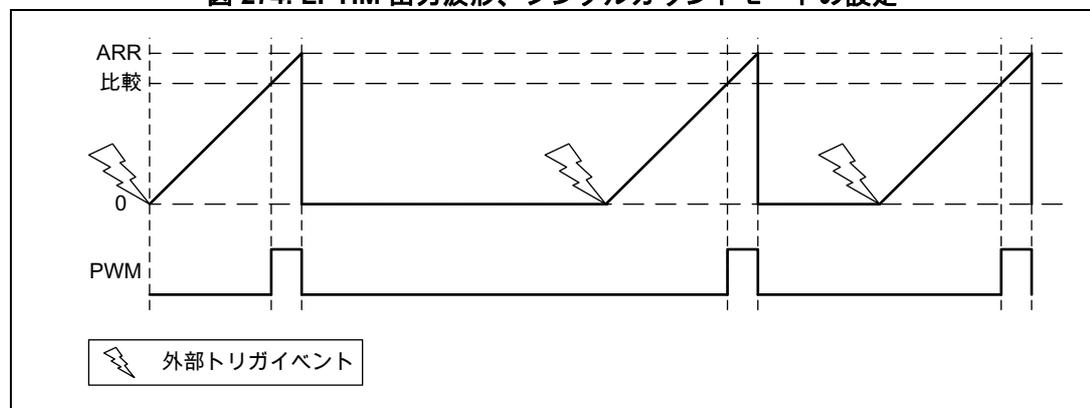
- 連続モード: タイマはフリーランニングし、トリガイベントによって開始し、タイマが無効にされるまで停止しません。
- ワンショットモード: タイマはトリガイベントによって開始し、ARR 値に達すると停止します。

新しいトリガイベントが発生すると、タイマは再開始します。カウンタの開始後、カウンタが ARR に達する前に発生したトリガイベントは破棄されます。

ワンショットカウントを有効にするには、SNGSTRT ビットをセットする必要があります。

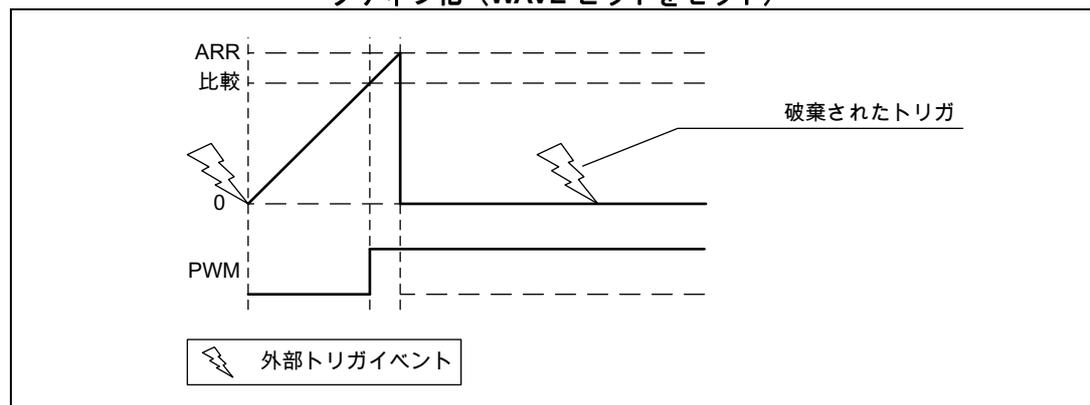
外部トリガが選択された場合、SNGSTRT ビットがセットされ、かつカウンタレジスタが停止された (ゼロ値を含む) 後に各外部トリガイベントが着信すると、カウンタは新たなワンショットカウントサイクルを開始します (図 274 を参照)。

図 274. LPTIM 出力波形、シングルカウントモードの設定



LPTIMx\_CFGR レジスタの WAVE ビットフィールドがセットされると、セットワンモードがアクティブになることに注意してください。この場合、最初のトリガに続いて一度だけ開始され、その後発生したトリガイベントはすべて破棄されます (図 274 を参照)。

図 275. LPTIM 出力波形、シングルカウントモードの設定およびセットワンモードのアクティブ化 (WAVE ビットをセット)



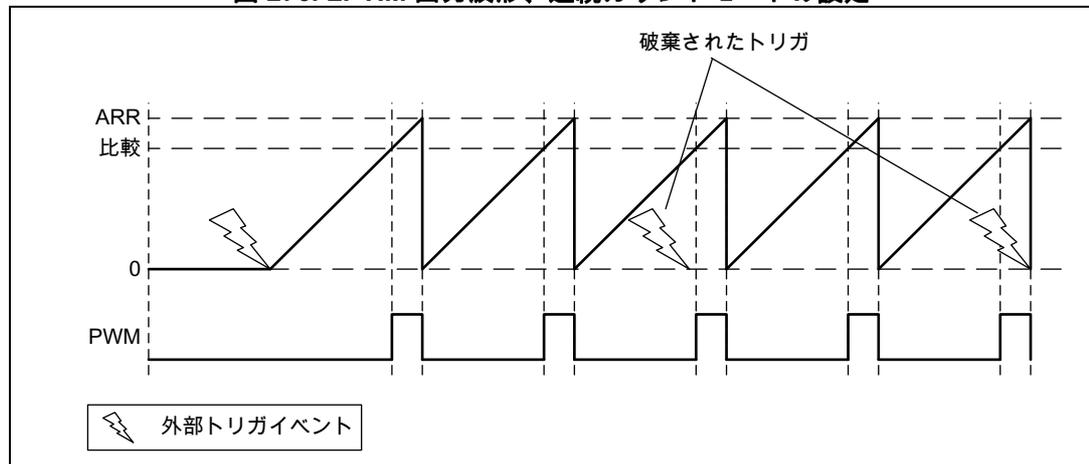
ソフトウェアによる開始の場合 (TRIGEN[1:0] = 00)、SNGSTRT をセットすると、カウンタはワンショットカウントを開始します。

連続カウントを有効にするには、CNTSTRT ビットをセットする必要があります。

外部トリガが選択された場合、CNTSTRT がセットされた後に外部トリガイイベントが着信すると、カウンタは連続カウントを開始します。その後発生した外部トリガイイベントはすべて、[図 276](#)に示すように破棄されます。

ソフトウェアによる開始の場合 (TRIGEN[1:0] = 00)、CNTSTRT をセットすると、カウンタは連続カウントを開始します。

図 276. LPTIM 出力波形、連続カウントモードの設定



SNGSTRT および CNTSTRT ビットは、タイマが有効なときのみ (イネーブルビットが 1 にセットされている)、セットできます。動作中にワンショットモードから連続モードに変更することが可能です。

以前に連続モードが選択されていた場合、SNGSTRT をセットすると、LPTIM はワンショットモードに切り替わります。カウンタ (アクティブな場合) は、ARR に達するとすぐに停止します。

以前にワンショットモードが選択されていた場合、CNTSTRT をセットすると、LPTIM は連続モードに切り替わります。カウンタ (アクティブな場合) は、ARR に達するとすぐに再開始します。

## 26.4.7 タイムアウト機能

選択されたトリガ入力のアクティブエッジの検出を使用して、LPTIM カウンタをリセットできます。この機能は、TIMOUT ビットで制御されます。

最初のトリガイイベントでタイマが開始し、その後のトリガイイベントでカウントがリセットされ、タイマが再開始します。

低電力タイムアウト機能を実現できます。タイムアウト値は、比較値に対応します。予期された時間内にトリガが発生しなかった場合、比較一致イベントによって MCU がウェイクアップします。

## 26.4.8 波形生成

2つの16ビットレジスタ LPTIMx\_ARR (自動再ロードレジスタ) と LPTIMx\_CMP (比較レジスタ) は、LPTIM 出力のいくつかの異なる波形を生成するために使用されます。

タイマは次の波形を生成できます。

- PWM モード: LPTIM 出力は、LPTIMx\_CMP レジスタと LPTIMx\_CNT レジスタの間で一致が発生するとすぐにセットされます。LPTIM 出力は、LPTIMx\_ARR レジスタと LPTIMx\_CNT レジスタの間で一致が発生するとすぐにリセットされます。
- ワンパルスモード: 出力波形は、最初のパルスについては PWM モードの波形と同様であり、その後、出力は永続的にリセットされます。
- セットワンスモード: 出力波形はワンパルスモードと同様ですが、出力は最後の信号レベルに保たれます (設定された出力極性に応じて)。

上記のモードでは、LPTIMx\_ARR レジスタの値は LPTIMx\_CMP レジスタの値より大きい必要があります。

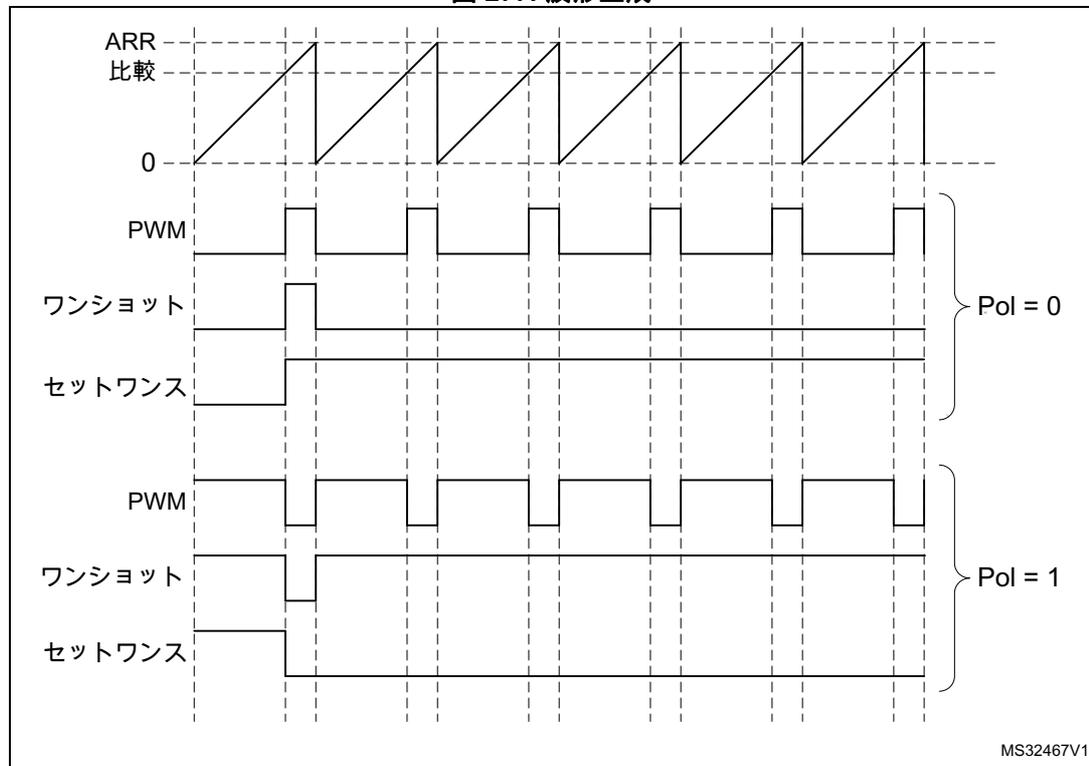
LPTIM 出力波形は、WAVE ビットによって次のように設定できます。

- WAVE ビットを0にリセットすると、LPTIM は CNTSTRT または SNGSTRT のいずれのビットがセットされているかに応じて、PWM 波形またはワンパルス波形のいずれかを生成します。
- WAVE ビットを1にセットすると、LPTIM はセットワンスモード波形を生成します。

WAVPOL ビットは、LPTIM 出力の極性を制御します。変更はただちに有効になるので、極性が再設定されると、タイマが有効になる前でも、出力はただちに変更されます。

LPTIM クロック周波数を2分周した周波数までの信号を生成できます。図 277 に、LPTIM 出力で生成できる3つの波形を示します。また、WAVPOL ビットを使用して極性を変更したときの効果も示します。

図 277. 波形生成



## 26.4.9 レジスタの更新

LPTIMx\_ARR レジスタと LPTIMx\_CMP レジスタは、APB バス書き込み操作の直後に、またはタイマがすでに開始していた場合は現在の周期の終了時に更新されます。

PRELOAD ビットは、LPTIMx\_ARR および LPTIMx\_CMP レジスタの更新方法を制御します。

- PRELOAD ビットが 0 にリセットされたときには、LPTIMx\_ARR および LPTIMx\_CMP レジスタは書き込みアクセスの直後に更新されます。
- PRELOAD ビットが 1 にセットされたときには、LPTIMx\_ARR および LPTIMx\_CMP レジスタは、タイマがすでに開始していた場合、現在の周期の終了時に更新されます。

APB バスと LPTIM は異なるクロックを使用するので、APB 書き込みから、これらの値がカウンタコンパレータで使用可能になるまで、遅延があります。この遅延中は、これらのレジスタへの追加の書き込みを避ける必要があります。

LPTIMx\_ISR レジスタの ARROK フラグと CMPOK フラグは、それぞれ、LPTIMx\_ARR レジスタと LPTIMx\_CMP レジスタへの書き込み操作が完了したことを示します。

LPTIMx\_ARR レジスタまたは LPTIMx\_CMP レジスタへの書き込みの後、同じレジスタへの新しい書き込み操作は、前の書き込み操作が完了してからでなければ実行できません。ARROK フラグまたは CMPOK フラグがセットされる前に連続した書き込みが行われると、予測不能な結果になります。

## 26.4.10 カウンタモード

LPTIM カウンタを使用して、LPTIM Input1 の外部イベントをカウントするか、内部クロックサイクルをカウントすることができます。CKSEL ビットおよび COUNTMODE ビットは、カウンタの更新にどのソースを使用するかを決定します。

LPTIM が Input1 の外部イベントをカウントするように設定された場合、カウンタは、CKPOL[1:0] ビットに書き込まれた値に応じて、立ち上がりエッジ、立ち下がりエッジ、または両方のエッジで更新できます。

CKSEL および COUNTMODE の値に応じて、以下に示すカウントモードを選択できます。

- CKSEL = 0 : LPTIM のクロックは、内部クロックソースによって供給されます。
  - COUNTMODE = 0  
LPTIM が内部クロックソースによってクロック供給されるように設定され、LPTIM カウンタが LPTIM 外部 Input1 で検出されたアクティブエッジによって更新されるように設定されたときには、LPTIM に供給される内部クロックをプリスケールしてはなりません (PRESC[2:0] = 000)。
  - COUNTMODE = 1  
LPTIM 外部 Input1 は、LPTIM に供給される内部クロックでサンプリングされます。結果として、イベントをミスしないためには、外部 Input1 信号の変化の周波数が、LPTIM に供給される内部クロックの周波数を超えない必要があります。
- CKSEL = 1 : LPTIM のクロックは、外部クロックソースによって供給されます。  
COUNTMODE の値は無視されます。  
この構成では、LPTIM は内部クロックソースを必要としません (グリッチフィルタ有効である場合を除く)。LPTIM 外部 Input1 に入力された信号が LPTIM のシステムクロックとして使用されます。この構成は、埋め込みオシレータを有効にしない動作モードに適しています。  
この構成の場合、LPTIM カウンタは、Input1 クロック信号の立ち上がりエッジまたは立ち下がりエッジで更新できますが、立ち上がりと立ち下がりの両方のエッジで更新することはできません。  
LPTIM 外部 Input1 に入力された信号は LPTIM のクロック動作にも使用されるので、カウンタがインクリメントされる前 (LPTIM が有効にされた後)、初期遅延があります。より正確には、LPTIM 外部 Input1 の (LPTIM が有効になった後の) 最初の 5 つのアクティブエッジは失われます。

## 26.4.11 タイマ有効

LPTIMx\_CR レジスタの イネーブルビットは、LPTIM を有効化／無効化するために使用されます。イネーブルビットをセットした後、LPTIM が実際に有効になるまで、2 カウンタクロックの遅延が必要です。

LPTIMx\_CFGR および LPTIMx\_IER レジスタの変更は、LPTIM が無効なときにのみ行う必要があります。

## 26.4.12 エンコーダモード

このモードでは、ロータリー素子の角度位置の検出に使用される直交エンコーダからの信号を処理できます。エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。これは、カウンタが 0 と LPTIMx\_ARR レジスタでプログラムされた自動再ロード値の間で（方向に応じて、0 から ARR まで、または ARR から 0 まで）連続的にカウントすることを意味します。したがって、開始前に LPTIMx\_ARR を設定する必要があります。Input1 と Input2 の 2 つの外部入力信号から、LPTIM カウンタのクロックのためのクロック信号が生成されます。この 2 つの信号の間の位相によって、カウント方向が決まります。

エンコーダモードは、LPTIM が内部クロックソースからクロック供給されるときにのみ使用できます。Input1 と Input2 の両方の信号周波数は、LPTIM 内部クロック周波数を 4 分周したものを超えてはなりません。これは、LPTIM の正しい動作を保証するために必要です。

方向の変更は、LPTIMx\_ISR レジスタの Down と Up の 2 つのフラグによって通知されます。また、LPTIMx\_IER レジスタを通じて有効化された場合、両方の方向変更イベントで割り込みを生成できます。

エンコーダモードを有効にするには、ENC ビットを 1 にセットする必要があります。LPTIM を、まず、連続モードに設定する必要があります。

エンコーダモードがアクティブなとき、LPTIM カウンタはインクリメンタルエンコーダの速度と方向に従って自動的に変更されます。したがって、その内容は常にエンコーダの位置を表します。カウント方向は、Up および Down フラグによって通知され、接続されているセンサの回転方向に対応します。

CKPOL[1:0] ビットを使用して設定されたエッジ感度に応じて、さまざまなカウントシナリオが可能です。次の表に、可能な組み合わせを示します（Input1 と Input2 は同時に切り替わらないと想定しています）。

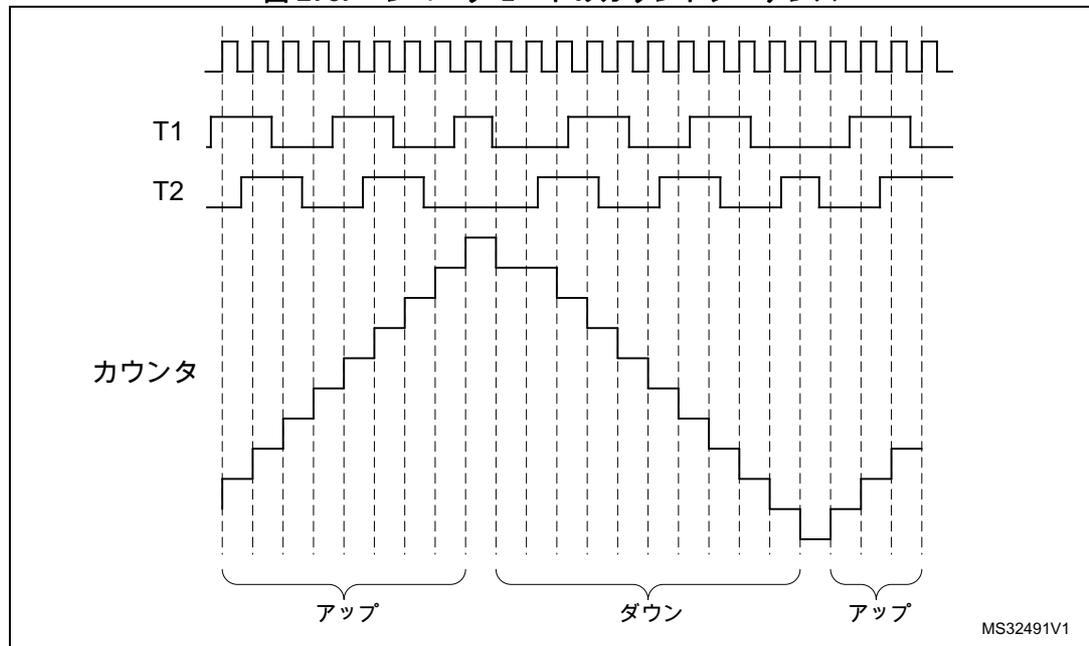
表 141. エンコーダのカウントシナリオ

アクティブエッジ	他方の信号のレベル (Input2 に対する Input1、Input1 に対する Input2)	Input1 信号		Input2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
立ち上がりエッジ	ハイ	ダウン	カウントなし	アップ	カウントなし
	ロー	アップ	カウントなし	ダウン	カウントなし
立ち下がりエッジ	ハイ	カウントなし	アップ	カウントなし	ダウン
	ロー	カウントなし	ダウン	カウントなし	アップ
両エッジ	ハイ	ダウン	アップ	アップ	ダウン
	ロー	アップ	ダウン	ダウン	アップ

次の図に、両方のエッジ感度が設定された場合のエンコーダモードのカウンタシーケンスを示します。

**注意：** このモードでは、LPTIM のクロックは内部クロックソースによって供給される必要があるため、CKSEL ビットをリセット値 (0) に維持する必要があります。また、プリスケアラの分周比はリセット値である 1 に等しくなければなりません (PRESC[2:0] ビットが 000 である必要があります)。

図 278. エンコーダモードのカウンタシーケンス



## 26.5 LPTIM 割り込み

LPTIMx\_IER レジスタで有効化されていた場合、次のイベントが発生すると、割り込み/ウェイクアップイベントが生成されます。

- 比較一致
- 自動再ロード一致 (エンコーダモードの場合は方向にかかわらず)
- 外部トリガイベント
- 自動再ロードレジスタへの書き込み完了
- 比較レジスタへの書き込み完了
- 方向変更 (エンコーダモード)、プログラム可能 (アップ/ダウン/両方)

**注：** LPTIMx\_ISR レジスタ (ステータスレジスタ) の対応するフラグがセットされた後で LPTIMx\_IER レジスタ (割り込み有効レジスタ) のビットがセットされた場合、割り込みはアサートされません。

## 26.6 LPTIM レジスタ

### 26.6.1 LPTIM 割り込みおよびステータスレジスタ (LPTIMx\_ISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM								
									r	r	r	r	r	r	r

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **DOWN** : カウンタの方向をアップからダウンへ変更

エンコーダモードでは、DOWN ビットは、カウンタの方向がアップからダウンに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。

ビット 5 **UP** : カウンタの方向をダウンからアップへ変更

エンコーダモードでは、UP ビットは、カウンタの方向がダウンからアップに変更されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。

ビット 4 **ARROK** : 自動再ロードレジスタ更新 OK

ARROK は、LPTIMx\_ARR レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。その場合、新しい書き込みを開始できます。

ビット 3 **CMPOK** : 比較レジスタ更新 OK

CMPOK は、LPTIMx\_CMP レジスタへの APB バス書き込み操作が正常に完了したことをアプリケーションに知らせるために、ハードウェアによってセットされます。その場合、新しい書き込みを開始できます。

ビット 2 **EXTTRIG** : 外部トリガエッジイベント

EXTTRIG は、選択された外部トリガ入力で有効なエッジが発生したことをアプリケーションに知らせるために、ハードウェアによってセットされます。タイマがすでに開始していたためにトリガが無視された場合、このフラグはセットされません。

ビット 1 **ARRM** : 自動再ロード一致

ARRM は、LPTIMx\_CNT レジスタの値が LPTIMx\_ARR レジスタの値に達したことをアプリケーションに知らせるために、ハードウェアによってセットされます。

ビット 0 **CMPM** : 比較一致

CMPM は、LPTIMx\_CNT レジスタの値が LPTIMx\_CMP レジスタの値に達したことをアプリケーションに知らせるために、ハードウェアによってセットされます。

## 26.6.2 LPTIM 割り込みクリアレジスタ (LPTIMx\_ICR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF								
									w	w	w	w	w	w	w

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **DOWNCF** : ダウンへの方向変更フラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの DOWN フラグがクリアされます。

ビット 5 **UPCF** : アップへの方向変更フラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの UP フラグがクリアされます。

ビット 4 **ARROKCF** : 自動再ロードレジスタ更新 OK フラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの ARROK フラグがクリアされます。

ビット 3 **CMPOKCF** : 比較レジスタ更新 OK フラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの CMPOK フラグがクリアされます。

ビット 2 **EXTTRIGCF** : 外部トリガ有効エッジフラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの EXTTRIG フラグがクリアされます。

ビット 1 **ARRMCF** : 自動再ロード一致フラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの ARRM フラグがクリアされます。

ビット 0 **CMPMCF** : 比較一致フラグクリア

このビットに 1 を書き込むと LPT\_ISR レジスタの CMP フラグがクリアされます。

## 26.6.3 LPTIM 割り込み有効レジスタ (LPTIMx\_IER)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE								
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **DOWNIE** : ダウンへの方向変更割り込み有効化

- 0 : DOWN 割り込み無効
- 1 : DOWN 割り込み有効

ビット 5 **UPIE** : アップへの方向変更割り込み有効化

- 0 : UP 割り込み無効
- 1 : UP 割り込み有効

ビット 4 **ARROKIE** : 自動再ロードレジスタ更新 OK 割り込み有効化

- 0 : ARROK 割り込み無効
- 1 : ARROK 割り込み有効

ビット 3 **CMPOKIE** : 比較レジスタ更新 OK 割り込み有効化

- 0 : CMPOK 割り込み無効
- 1 : CMPOK 割り込み有効

ビット 2 **EXTTRIGIE** : 外部トリガ有効エッジ割り込み有効化

- 0 : EXTTRIG 割り込み無効
- 1 : EXTTRIG 割り込み有効

ビット 1 **ARRMIE** : 自動再ロード一致割り込み有効化

- 0 : ARRM 割り込み無効
- 1 : ARRM 割り込み有効

ビット 0 **CMPMIE** : 比較一致割り込み有効化

- 0 : CMPM 割り込み無効
- 1 : CMPM 割り込み有効

**注意 :** LPTIMx\_IER レジスタの変更は、LPTIM が無効 (イネーブルビットが0 にリセットされている) のときにのみ行う必要があります。

## 26.6.4 LPTIM 設定レジスタ (LPTIMx\_CFGR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN		Res.
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGSEL			Res.	PRESC			Res.	TRGFLT		Res.	CKFLT		CKPOL		CKSEL
r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w		r/w	r/w	r/w	r/w	r/w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ENC** : エンコーダモード有効化

ENC ビットは、エンコーダモードを制御します。

- 0 : エンコーダモード無効
- 1 : エンコーダモード有効

ビット 23 **COUNTMODE** : カウンタモード有効化

COUNTMODE ビットは、LPTIM がカウンタのクロックに使用するクロックソースを選択します。

- 0 : カウンタは各内部クロックのパルスに従ってインクリメントされます。
- 1 : カウンタは LPTIM 外部 Input1 の各有効なクロックパルスに従ってインクリメントされます。

ビット 22 **PRELOAD** : レジスタ更新モード

PRELOAD ビットは、LPTIMx\_ARR および LPTIMx\_CMP レジスタの更新方法を制御します。

- 0 : レジスタは、各 APB バス書き込みアクセス後に更新されます。
- 1 : レジスタは、現在の LPTIM 周期の終了時に更新されます。

ビット 21 **WAVPOL** : 波形極性

WAVPOL ビットは、出力の極性を制御します。

- 0 : LPTIM 出力は LPTIMx\_ARR レジスタと LPTIMx\_CMP レジスタの比較結果を反映します。
- 1 : LPTIM 出力は LPTIMx\_ARR レジスタと LPTIMx\_CMP レジスタの比較結果の逆を反映します。

ビット 20 **WAVE** : 波形

WAVE ビットは、出力波形を制御します。

- 0 : セットワンスモード、PWM/ワンパルス波形 (OPMODE ビットに応じて) を非アクティブにします。
- 1 : セットワンスモードをアクティブにします。

ビット 19 **TIMOUT** : タイムアウト有効化

TIMOUT ビットは、タイムアウト機能を制御します。

- 0 : タイマがすでに開始しているときに着信したトリガイベントは無視されます。
- 1 : タイマがすでに開始しているときにトリガイベントが着信すると、カウンタがリセットされ、再開始します。

ビット 18:17 **TRIGEN** : トリガ有効化および極性

TRIGEN ビットは、LPTIM カウンタが外部トリガによって開始されるかどうかを制御します。外部トリガオプションが選択された場合、トリガのアクティブエッジについて 3 つの構成が可能です。

- 00 : ソフトウェアトリガ (カウンタの開始はソフトウェアによって行われます)。
- 01 : 立ち上がりエッジがアクティブエッジです。
- 10 : 立ち下がりエッジがアクティブエッジです。
- 11 : 両方のエッジがアクティブエッジです。

ビット 16 予約済みであり、リセット値に保持する必要があります。

**ビット 15:13 TRIGSEL** : トリガセレクタ

TRIGSEL ビットは、次の 8 つのソースの中の LPTIM のトリガイベントの役目を果たすトリガソースを選択します。

000 : ext\_trig0  
001 : ext\_trig1  
010 : ext\_trig2  
011 : ext\_trig3  
100 : ext\_trig4  
101 : ext\_trig5  
110 : ext\_trig6  
111 : ext\_trig7

各タイマにとっての ITRx の意味の詳細については、[表 142 : LPTIM 外部トリガ接続](#) を参照してください。

ビット 12 予約済みであり、リセット値に保持する必要があります。

**ビット 11:9 PRESC** : クロックプリスケアラ

PRESC ビットは、プリスケアラ分周比を設定します。次の分周比から選択できます。

000 : /1  
001 : /2  
010 : /4  
011 : /8  
100 : /16  
101 : /32  
110 : /64  
111 : /128

ビット 8 予約済みであり、リセット値に保持する必要があります。

**ビット 7:6 TRGFLT** : 設定可能なトリガ用デジタルフィルタ

TRGFLT 値は、有効なレベル遷移とみなされる前に、内部トリガでレベル変更が発生したときに検出されなければならない連続した等しいサンプルの数を設定します。この機能を使用するには、内部クロックソースが必要です。

00 : トリガのアクティブレベル変更は、有効なトリガとみなされます。

01 : トリガのアクティブレベル変更が有効なトリガとみなされるためには、少なくとも 2 クロック周期にわたって安定している必要があります。

10 : トリガのアクティブレベル変更が有効なトリガとみなされるためには、少なくとも 4 クロック周期にわたって安定している必要があります。

11 : トリガのアクティブレベル変更が有効なトリガとみなされるためには、少なくとも 8 クロック周期にわたって安定している必要があります。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:3 **CKFLT** : 設定可能な外部クロック用デジタルフィルタ

CKFLT 値は、有効なレベル遷移とみなされる前に、外部クロック信号でレベル変更が発生したときに検出されなければならない連続した等しいサンプルの数を設定します。この機能を使用するには、内部クロックソースが必要です。

- 00 : 外部クロック信号のレベル変更は、有効な遷移とみなされます。
- 01 : 外部クロック信号のレベル変更が有効な遷移とみなされるためには、少なくとも 2 クロック周期にわたって安定している必要があります。
- 10 : 外部クロック信号のレベル変更が有効な遷移とみなされるためには、少なくとも 4 クロック周期にわたって安定している必要があります。
- 11 : 外部クロック信号のレベル変更が有効な遷移とみなされるためには、少なくとも 8 クロック周期にわたって安定している必要があります。

ビット 2:1 **CKPOL** : クロック極性

LPTIM のクロックが、外部クロックソースによって供給される場合 :

LPTIM のクロックが外部クロックソースによって供給される時、CKPOL ビットは、カウンタによって使用されるアクティブエッジを設定するために使用されます。

- 00 : 立ち上がりエッジが、カウンタに使用されるアクティブエッジです。
- 01 : 立ち下がりエッジが、カウンタに使用されるアクティブエッジです。
- 10 : 両方のエッジがアクティブエッジです。外部クロック信号の両方のエッジがアクティブエッジとみなされるときには、LPTIM のクロックは内部クロックソースからも供給される必要があります、その周波数は外部クロック周波数の 4 倍以上である必要があります。
- 11 : 使用できません。

LPTIM がエンコーダモードで設定されている (ENC ビットがセットされている) 場合 :

- 00 : エンコーダサブモード 1 がアクティブです。
- 01 : エンコーダサブモード 2 がアクティブです。
- 10 : エンコーダサブモード 3 がアクティブです。

エンコーダモードのサブモードの詳細については、[セクション 26.4.12 : エンコーダモード](#)を参照してください。

ビット 0 **CKSEL** : クロックセレクタ

CKSEL ビットは、LPTIM が使用するクロックソースを選択します :

- 0 : LPTIM のクロックは内部クロックソースによって供給されます (APB クロックまたは埋め込みオシレータ)。
- 1 : LPTIM のクロックは、LPTIM 外部 Input1 を通じて外部クロックソースによって供給されます。

**注意 :** *LPTIMx\_CFGR レジスタの変更は、LPTIM が無効 (イネーブルビットが 0 にリセットされている) のときにのみ行う必要があります。*

表 142. LPTIM 外部トリガ接続

TRIGSEL	外部トリガ
ext_trig0	GPIO
ext_trig1	RTC_ALARM
ext_trig2	RTC_ALARMB
ext_trig3	RTC_TAMP1_OUT
ext_trig4	RTC_TAMP2_OUT
ext_trig5	RTC_TAMP3_OUT
ext_trig6	予約済み
ext_trig7	予約済み

## 26.6.5 LPTIM 制御レジスタ (LPTIMx\_CR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

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

ビット 31 : 3 予約済みであり、リセット値に保持する必要があります。

### ビット 2 CNTSTRT : 連続モードでタイマ開始

このビットは、ソフトウェアによってセットされ、ハードウェアによってクリアされます。

ソフトウェア開始 (TRIGEN[1:0] = 00) の場合、このビットをセットすると、LPTIM は連続モードで開始します。ソフトウェア開始が無効 (TRIGEN[1:0] が 00 以外) の場合、このビットをセットすると、外部トリガが検出されるとすぐに、タイマは連続モードで開始します。

シングルパルスモードでのカウント中にこのビットがセットされた場合、LPTIMx\_ARR レジスタと LPTIMx\_CNT レジスタが次に一致したときにタイマは停止せず、LPTIM カウンタは連続モードでのカウントを続行します。このビットをセットできるのは、LPTIM が有効なときだけです。ハードウェアによって自動的にリセットされます。

### ビット 1 SNGSTRT : シングルモードで LPTIM 開始

このビットは、ソフトウェアによってセットされ、ハードウェアによってクリアされます。

ソフトウェア開始 (TRIGEN[1:0] = 00) の場合、このビットをセットすると、LPTIM はシングルパルスモードで開始します。

ソフトウェア開始が無効 (TRIGEN[1:0] が 00 以外) の場合、このビットをセットすると、外部トリガが検出されるとすぐに、LPTIM はシングルパルスモードで開始します。

LPTIM が連続カウントモードのときにこのビットがセットされた場合、LPTIM は LPTIMx\_ARR レジスタと LPTIMx\_CNT レジスタが次に一致したときに停止します。

このビットをセットできるのは、LPTIM が有効なときだけです。ハードウェアによって自動的にリセットされます。

### ビット 0 ENABLE : LPTIM 有効化

イネーブルビットは、ソフトウェアによってセット/クリアされます。

0 : LPTIM は無効です。

1 : LPTIM は有効です。

## 26.6.6 LPTIM 比較レジスタ (LPTIMx\_CMP)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CMP** : 比較値

CMP は、LPTIM によって使用される比較値です。

LPTIMx\_CMP レジスタの内容の変更は、LPTIM が有効 (イネーブルビットが 1 にセットされている) のときのみ行う必要があります。

## 26.6.7 LPTIM 自動再ロードレジスタ (LPTIMx\_ARR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0001

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ARR** : 自動再ロード値

ARR は、LPTIM の自動再ロード値です。

この値は、CMP[15:0] 値より大きくなければなりません。

LPTIMx\_ARR レジスタの内容の変更は、LPTIM が有効 (イネーブルビットが 1 にセットされている) のときのみ行う必要があります。

## 26.6.8 LPTIM カウンタレジスタ (LPTIMx\_CNT)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 CNT : カウンタ値

LPTIM が非同期クロックで実行しているとき、LPTIMx\_CNT レジスタを読み出すと、信頼できない値が返されることがあります。したがって、この場合、2つの連続した読み出しアクセスを実行して、返された2つの値が同じかどうかを確認する必要があります。

信頼できる LPTIM\_CNT レジスタの読み出しアクセスでは、2回の連続した読み出しアクセスを実行して比較する必要がありますことに注意してください。2回の連続読み出しアクセスで得られた値が同じであるとき、各読み出しアクセスは信頼できると考えられます。

## 26.6.9 LPTIM1 オプションレジスタ (LPTIM1\_OR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OR_1	OR_0													
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 OR\_1 : オプションレジスタビット 1

- 0 : LPTIM1 入力 2 は I/O に接続されます。
- 1 : LPTIM1 入力 2 は COMP2\_OUT に接続されます。

ビット 0 OR\_0 : オプションレジスタビット 0

- 0 : LPTIM1 入力 1 は I/O に接続されます。
- 1 : LPTIM1 入力 1 は COMP1\_OUT に接続されます。



## 26.6.10 LPTIM2 オプションレジスタ (LPTIM2\_OR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OR_1	OR_0													
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **OR\_1** : オプションレジスタビット 1

- 0 : LPTIM2 入力 1 は I/O に接続されます。
- 1 : LPTIM2 入力 1 は COMP2\_OUT に接続されます。

ビット 0 **OR\_0** : オプションレジスタビット 0

- 0 : LPTIM2 入力 1 は I/O に接続されます。
- 1 : LPTIM2 入力 1 は COMP1\_OUT に接続されます。

**注 :** **OR\_1 と OR\_0 の両方がセットされている場合、LPTIM2 入力 1 は (COMP1\_OUT OR COMP2\_OUT) に接続されます。**

## 26.6.11 LPTIM レジスタマップ

次の表に LPTIM レジスタの一覧を示します。

表 143. LPTIM レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	LPTIMx_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM									
	リセット値																										0	0	0	0	0	0	0	
0x04	LPTIMx_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF									
	リセット値																										0	0	0	0	0	0	0	
0x08	LPTIMx_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE									
	リセット値																										0	0	0	0	0	0	0	
0x0C	LPTIMx_CFGR	Res.	ENC	COUNTMODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN	Res.	TRIGSEL	Res.	PRESC	Res.	TRGFLT	Res.	CKFLT	CKPOL	CKSEL															
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
0x10	LPTIMx_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTSTRT	SNGSTRT	ENABLE										
	リセット値																													0	0	0		
0x14	LPTIMx_CMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
	リセット値																																	
0x18	LPTIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
	リセット値																																	
0x1C	LPTIMx_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
	リセット値																																	
0x20	LPTIMx_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.									
	リセット値																																	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。



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

## 27 独立型ウォッチドッグ (IWDG)

### 27.1 概要

デバイスは、内蔵ウォッチドッグペリフェラルを搭載しており、使用上、高い安全レベル、タイミングの正確さ、および柔軟性を兼ね備えています。独立型ウォッチドッグペリフェラルは、ソフトウェア障害による誤動作を検出および解決し、カウンタが与えられたタイムアウト値に達すると、システムリセットをトリガします。

独立型ウォッチドッグ (IWDG) は、独自の低速クロック (LSI) によってクロック供給されるので、メインクロックに障害があってもアクティブなままです。

IWDG は、メインアプリケーションの外部で、完全に独立したプロセスとして実行するウォッチドッグが必要な場合に最適ですが、タイミング精度が低いという制約があります。ウィンドウ型ウォッチドッグの詳細については、[セクション 28 \(867 ページ\)](#) を参照してください。

### 27.2 IWDG の主な機能

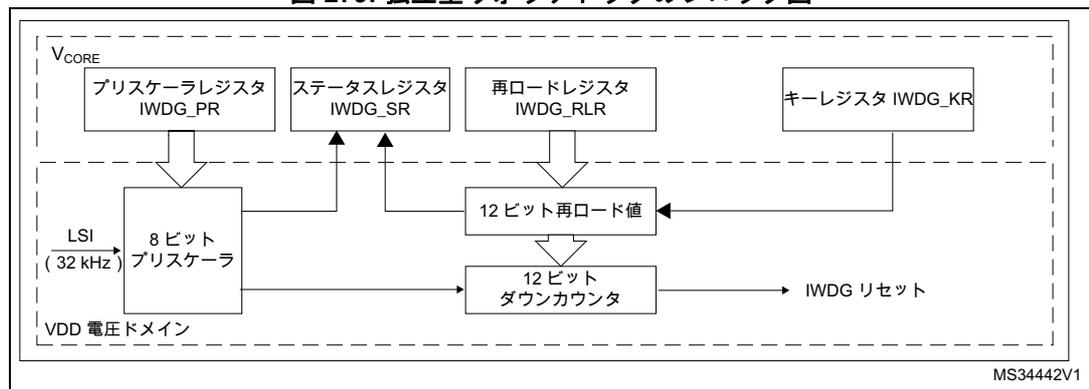
- フリーランニングダウンカウンタ
- 独立した RC オシレータからのクロック供給 (STANDBY および STOP モードで動作可能)
- 条件付きリセット
  - ダウンカウンタの値が 0x000 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
  - ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合)。

### 27.3 IWDG の機能説明

#### 27.3.1 IWDG ブロック図

図 279 に、独立型ウォッチドッグモジュールの機能ブロックを示します。

図 279. 独立型ウォッチドッグのブロック図



**注：** ウォッチドッグ機能は、STOP モードおよびSTANDBY モードでも機能する  $V_{CORE}$  電圧ドメインに実装されています。

キーレジスタ (IWDG\_KR) に値 0x0000 CCCC が書き込まれることによって独立型ウォッチドッグが開始すると、カウンタはリセット値 0xFFFF からカウントダウンを開始します。カウント値の終わり (0x000) に達すると、リセット信号が生成されます (IWDG\_reset)。

IWDG\_KR レジスタにキー値 0x0000 AAAA が書き込まれると、IWDG\_RLR の値がカウンタに再ロードされ、ウォッチドッグのリセットが防止されます。

## 27.3.2 ウィンドウオプション

IWDG は、IWDG\_WINR レジスタに適切なウインドウをセットすることによって、ウィンドウ型ウォッチドッグとしても機能します。

カウンタがウィンドウレジスタ (IWDG\_WINR) に格納された値より大きい間に再ロード操作が行われると、リセットが生成されます。

IWDG\_WINR のデフォルト値は 0x0000 0FFF です。この値が更新されない場合は、ウィンドウオプションは無効にされます。

ウィンドウ値が変わるとすぐに再ロード操作が行われ、ダウンカウンタを IWDG\_RLR 値にリセットし、次の再ロードを生成するためのサイクル数計算を容易にします。

### ウィンドウオプションが有効な場合の IWDG の設定

1. IWDG\_KR レジスタに 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. IWDG\_KR レジスタに 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. IWDG\_PR を 0 から 7 までプログラムすることによって、IWDG のプリスケアラに書き込みを行います。
4. 再ロードレジスタ (IWDG\_RLR) に書き込みます。
5. レジスタが更新されるのを待ちます (IWDG\_SR = 0x0000 0000)。
6. ウィンドウレジスタ IWDG\_WINR に書き込みます。これにより、カウンタ値 IWDG\_RLR が自動的にリフレッシュされます。

**注：** ウィンドウ値を書き込むことで、IWDG\_SR が“0x0000 0000”にセットされた時点でカウンタ値を RLR でリフレッシュすることができます。

### ウィンドウオプションが無効な場合の IWDG の設定

ウィンドウオプションが使用されていない場合、IWDG は以下のように設定することができます。

1. IWDG\_KR レジスタに 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. IWDG\_KR レジスタに 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. IWDG\_PR を 0 から 7 までプログラムすることによって、IWDG のプリスケアラに書き込みを行います。
4. 再ロードレジスタ (IWDG\_RLR) に書き込みます。
5. レジスタが更新されるのを待ちます (IWDG\_SR = 0x0000 0000)。
6. カウンタ値を IWDG\_RLR (IWDG\_KR = 0x0000 AAAA) でリフレッシュします。

## 27.3.3 ハードウェアウォッチドッグ

デバイスのオプションビットを使って「ハードウェアウォッチドッグ」機能が有効化されると、ウォッチドッグは電源投入時に自動的に有効になり、カウンタがカウントの終わりに達する前にソフトウェアによってキーレジスタへ書き込まれない限り、またはダウンカウンタがウィンドウ内に再ロードされた場合は、リセットを生成します。

## 27.3.4 低電力凍結

IWDG\_STOP および IWDG\_STBY のオプション設定に応じて、IWDG では STOP モード中および STANDBY モード中にそれぞれカウントを継続または停止できます。STOP または STANDBY モード中に IWDG を実行したままにすると、このモードからデバイスをウェイクアップできます。詳細については、[Section : User and read protection option bytes](#) を参照してください。

## 27.3.5 レジスタのアクセス保護

IWDG\_PR、IWDG\_RLR、および IWDG\_WINR レジスタへの書き込みアクセスは保護されます。これらを変更するには、まず、IWDG\_KR レジスタにコード 0x0000 5555 を書き込む必要があります。このレジスタに別の値を書き込むと、シーケンスがブレイクされ、レジスタへのアクセスが再び保護されます。これは、再ロード操作 (0x0000 AAAA の書き込み) であることを意味します。ステータスレジスタは、プリスケアラの更新、あるいはダウンカウンタ再ロード値やウィンドウ値の更新が行われていることを示すために使用されます。

## 27.3.6 デバッグモード

マイクロコントローラがデバッグモードになると (コアは停止状態)、IWDG カウンタは、DBG モジュールの DBG\_IWDG\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

## 27.4 IWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 27.4.1 キーレジスタ (IWDG\_KR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされる)

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **KEY[15:0]** : キー値 (書き込み専用、読み出しでは 0x0000)

これらのビットには、ソフトウェアによって一定間隔でキー値 0xAAAA が書き込まれなければなりません。そうしないと、カウンタが 0 に達した時点でウォッチドッグがリセットを生成します。

キー値 0x5555 を書き込むことによって、IWDG\_PR、IWDG\_RLR、および IWDG\_WINR レジスタへのアクセスが可能になります ([セクション 27.3.5 : レジスタのアクセス保護](#)を参照)。

キー値 CCCCh を書き込むと、ウォッチドッグが開始します (ハードウェアウォッチドッグオプションが選択されている場合を除く)。

## 27.4.2 プリスケーラレジスタ (IWDG\_PR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **PR[2:0]** : プリスケーラ分周回路

これらのビットは、書き込みアクセス保護されています ([セクション 27.3.5: レジスタのアクセス保護](#)を参照)。カウンタクロックを供給するプリスケーラ分周回路を選択するようにソフトウェアで書き込まれます。プリスケーラ分周比を変更できるようにするには、IWDG\_SR レジスタの PVU ビットをリセットする必要があります。

- 000 : 4 分周
- 001 : 8 分周
- 010 : 16 分周
- 011 : 32 分周
- 100 : 64 分周
- 101 : 128 分周
- 110 : 256 分周
- 111 : 256 分周

**注 :** このレジスタを読み出すと、VDD 電圧ドメインからプリスケーラ値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG\_SR レジスタの PVU ビットがリセットされているときのみとなります。

## 27.4.3 再ロードレジスタ (IWDG\_RLR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット11:0 **RL[11:0]** : ウォッチドッグカウンタ再ロード値

これらのビットは、書き込みアクセス保護されています ([セクション 27.3.5](#)を参照)。IWDG\_KR レジスタに値 0xAAAA が書き込まれるたびにウォッチドッグカウンタにロードされる値を定義するために、ソフトウェアで書き込まれます。ウォッチドッグカウンタは、この値からカウントダウンします。タイムアウトまでの時間は、この値とクロックプリスケールによって決まります。タイムアウトに関する詳細はデータシートを参照してください。

再ロード値を変更できるようにするには、IWDG\_SR レジスタの RVU ビットをリセットする必要があります。

**注:** このレジスタを読み出すと、VDD 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG\_SR レジスタの RVU ビットがリセットされているときのみとなります。

## 27.4.4 ステータスレジスタ (IWDG\_SR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされません)

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

### ビット 2 WVU : ウォッチドッグカウンタウィンドウ値の更新

このビットは、ウィンドウ値の更新が進行中であることを示すために、ハードウェアによってセットされます。V<sub>DD</sub> 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

ウィンドウ値は、WVU ビットがリセットされているときのみ更新できます。

このビットは、一般のウィンドウが 1 の場合のみ生成されます。

### ビット 1 RVU : ウォッチドッグカウンタ再ロード値の更新

このビットは、再ロード値の更新が進行中であることを示すために、ハードウェアによってセットされます。V<sub>DD</sub> 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

再ロード値は、RVU ビットがリセットされているときのみ更新できます。

### ビット 0 PVU : ウォッチドッグプリスケアラ値の更新

このビットは、プリスケアラ値の更新が進行中であることを示すために、ハードウェアによってセットされます。V<sub>DD</sub> 電圧ドメインでプリスケアラ更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

プリスケアラ値は、PVU ビットがリセットされているときのみ更新できます。

注 :

**複数の再ロード値、プリスケアラ値、またはウィンドウ値がアプリケーションで使用される場合は、それぞれ、再ロード値を変更する前に RVU ビットがリセットされるまで待つか、プリスケアラ値を変更する前に PVU ビットがリセットされるまで待つか、またはウィンドウ値を変更する前に WVU ビットがリセットされるまで待つ必要があります。ただし、プリスケアラ値、再ロード値、またはウィンドウ値を更新した後は、RVU、PVU、または WVU がリセットされるのを待たずに、コード実行を続けることができます (低電力モードに入った場合を除く)。**

## 27.4.5 ウィンドウレジスタ (IWDG\_WINR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット11:0 **WIN[11:0]** : ウォッチドッグカウンタウィンドウ値

これらのビットは、書き込みアクセス保護されています ([セクション 27.3.5](#)を参照)。これらのビットは、ダウンカウンタと比較されるウィンドウ値の上限を含みます。

リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、0x0 よりも大きい間にダウンカウンタを再ロードする必要があります。

再ロード値を変更できるようにするには、IWDG\_SR レジスタの WVU ビットをリセットする必要があります。

**注 :** このレジスタを読み出すと、V<sub>DD</sub> 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG\_SR レジスタの WVU ビットがリセットされているときのみとなります。

## 27.4.6 IWDG レジスタマップ

次の表に、IWDG レジスタマップとリセット値を示します。

表 144. IWDG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	IWDG_KR	Res	KEY[15:0]																																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	IWDG_PR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR[2:0]																		
	リセット値																																0	0	0
0x08	IWDG_RLR	Res	Res	Res	Res	Res	RL[11:0]																												
	リセット値																						1	1	1	1	1	1	1	1	1	1	1	1	
0x0C	IWDG_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WVU																	
	リセット値																																	0	0
0x10	IWDG_WINR	Res	Res	Res	Res	Res	WIN[11:0]																												
	リセット値																						1	1	1	1	1	1	1	1	1	1	1	1	

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 28 システムウィンドウ型ウォッチドッグ (WWDG)

### 28.1 概要

システムウィンドウ型ウォッチドッグ (WWDG) は、通常、外部の影響や予期しない論理条件などによって発生し、アプリケーションプログラムを正常なシーケンスから逸脱させるソフトウェア障害の発生を検出するために使用されます。ウォッチドッグ回路は、T6 ビットがクリアされる前にプログラムがダウンカウンタの内容をリフレッシュしない限り、プログラムされた時間の経過後に MCU リセットを生成します。MCU リセットは、ダウンカウンタがウィンドウレジスタ値に達する前に 7 ビットのダウンカウンタの値 (制御レジスタ内) がリフレッシュされた場合にも生成されます。このことは、限られた時間枠 (time-window) の間にカウンタがリフレッシュされなければならないことを意味します。

WWDG クロックは、APB クロックから分周され、また設定可能な時間枠 (time-window) があるので、これをプログラムしてアプリケーション動作の異常な進み・遅れを検出できます。

WWDG は、正確な時間枠内で反応するウォッチドッグが必要なアプリケーションに適しています。

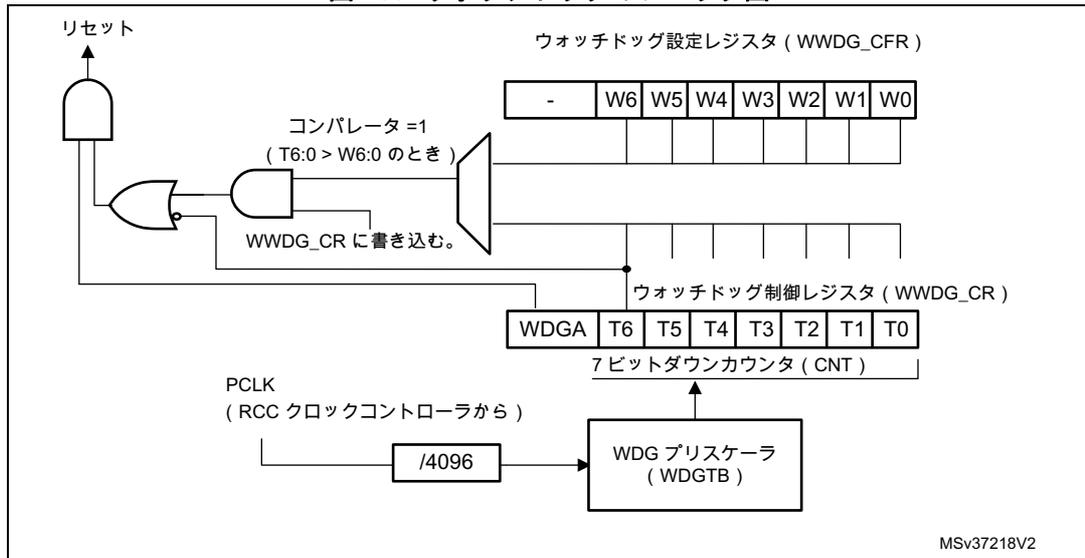
### 28.2 WWDG の主な機能

- プログラム可能なフリーランニングダウンカウンタ
- 条件付きリセット
  - ダウンカウンタの値が 0x40 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
  - ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合) ([図 281](#)を参照)。
- 早期ウェイクアップ割り込み (EWI) : ダウンカウンタが 0x40 になったときにトリガ (有効であり、ウォッチドッグがアクティブな場合)

### 28.3 WWDG の機能説明

ウォッチドッグが有効 (WWDG\_CR レジスタの WDGA ビットがセットされている) な場合、7 ビットのダウンカウンタ (T[6:0] ビット) が 0x40 に達して 0x3F に戻った (T6 がクリアされた) 時点で、リセットを開始します。カウンタがウィンドウレジスタに格納された値より大きい間にソフトウェアがカウンタを再ロードした場合にも、リセットが生成されます。

図 280. ウォッチドッグのブロック図



アプリケーションプログラムは、通常動作時には定期的に WWDG\_CR レジスタへの書き込みを行って、MCU リセットを防ぐ必要があります。この操作は、カウンタの値がウィンドウレジスタの値より小さいときに限られます。WWDG\_CR レジスタに格納される値は、0xFF から 0xC0 の間でなければなりません。

### 28.3.1 ウォッチドッグの有効化

ウォッチドッグはリセット後は常に無効です。これを有効にするには、WWDG\_CR レジスタの WDGA ビットをセットします。この後は、リセット以外の方法でウォッチドッグを無効にすることはできません。

### 28.3.2 ダウンカウンタの制御

このダウンカウンタはフリーランニングであり、ウォッチドッグが無効状態であってもカウントダウンを続けます。ウォッチドッグを有効にするときには、T6 ビットをセットして、ただちにリセットが生成されるのを防ぐ必要があります。

T[5:0] ビットは、ウォッチドッグがリセットを生成するまでの時間遅延を表すインクリメント数を含みます。このタイミングは、WWDG\_CR レジスタへの書き込み時のプリスケアラの状態が不明なので最小値から最大値の間で変化します (図 281 参照)。設定レジスタ (WWDG\_CFR) は、ウィンドウの上限值を含みます。リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、0x3F よりも大きい間にダウンカウンタを再ロードする必要があります。図 281 に、ウィンドウ型ウォッチドッグのプロセスを示します。

**注:** T6 ビットを使用して、ソフトウェアリセットを生成することができます (WDGA ビットはセット、T6 ビットはクリアされる)。

## 28.3.3 高度なウォッチドッグ割り込み機能

実際にリセットが生成される前に特定の安全処理やデータロギングを実施する必要がある場合は、早期ウェイクアップ割り込み (EWI) が使用できます。EWI 割り込みは、WWDG\_CFR レジスタの EWI ビットをセットすることによって有効になります。ダウンカウンタ値が 0x40 に到達すると、EWI 割り込みが生成され、対応する割り込みサービスルーチン (ISR) を使用してデバイスをリセットする前に特定の処理 (通信やデータロギングなど) をトリガすることができます。

アプリケーションによっては、EWI 割り込みを使用して、WWDG リセットを生成せずにソフトウェアのシステムチェックやシステム復旧/グレースフルデグラデーションを管理することができます。この場合、対応する割り込みサービスルーチン (ISR) で WWDG カウンタを再ロードし、WWDG リセットを回避してから必要な操作をトリガしてください。

EWI 割り込みは、WWDG\_SR レジスタの EWIF ビットに“0”を書き込むことによってクリアされます。

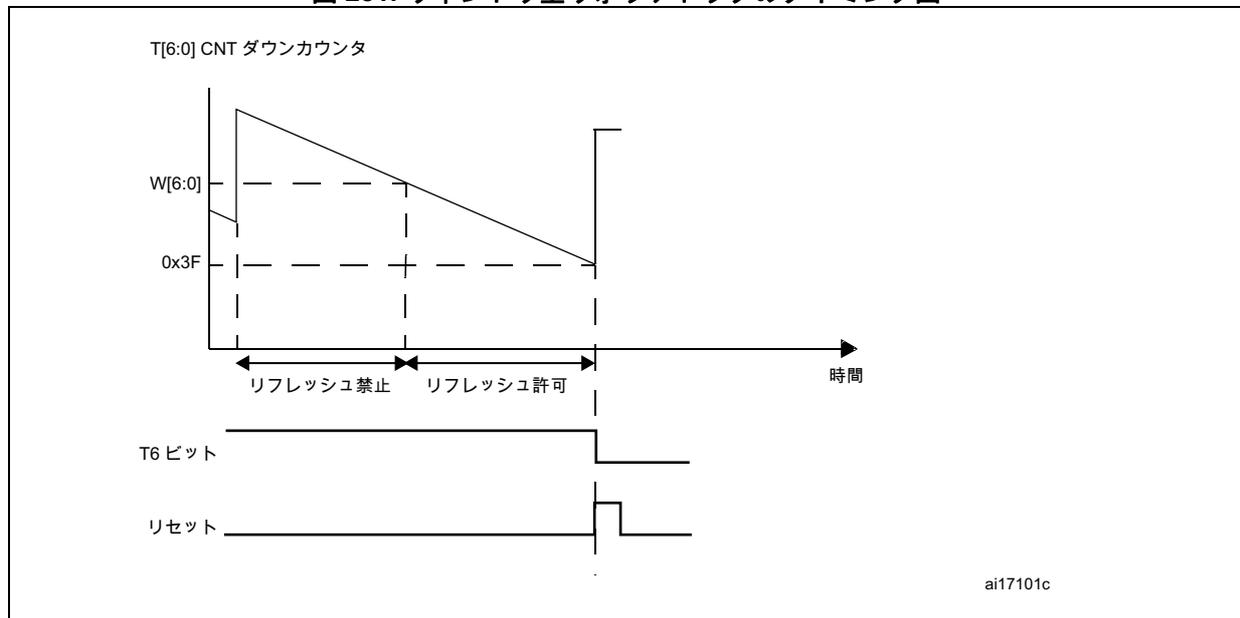
**注：** *優先順位の高いタスクにおけるシステムロックなどによって EWI 割り込みが使用できない場合、最終的には WWDG リセットが生成されます。*

## 28.3.4 ウォッチドッグタイムアウトをプログラムする方法

図 281 の式を使用して、WWDG のタイムアウトを計算することができます。

**警告：** WWDG\_CR レジスタに書き込むときには、ただちにリセットされるのを防ぐために、常に T6 ビットに 1 を書き込んでください。

図 281. ウィンドウ型ウォッチドッグのタイミング図



タイムアウト値は次の式で算出されます。

$$t_{\text{WWDG}} = t_{\text{PCLK}} \times 4096 \times 2^{\text{WDGTB}[1:0]} \times (\text{T}[5:0] + 1) \quad (\text{ms})$$

ここで、

$t_{\text{WWDG}}$  : WWDG タイムアウト

$t_{\text{PCLK}}$  : APB クロック周期の測定値 (ms)

4096 : 内部分周器に対応する値

たとえば、APB 周波数が 48 MHz と等しい場合、WDGTB[1:0] は 3 にセットされ、T[5:0] は 63 にセットされます。

$$t_{\text{WWDG}} = 1/48000 \times 4096 \times 2^3 \times (63 + 1) = 43.69 \text{ ms}$$

$t_{\text{WWDG}}$  の最小値と最大値については、データシートを参照してください。

## 28.3.5 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex<sup>®</sup>-M7 コアは停止状態)、WWDG カウンタは、DBG モジュールの DBG\_WWDG\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 40.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 28.4 WWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

パリティレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 28.4.1 制御レジスタ (WWDG\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 007F

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDGA** : 有効化ビット

このビットは、ソフトウェアでセットされ、リセット後はハードウェアによってのみクリアされます。WDGA = 1 のとき、ウォッチドッグはリセットを生成できます。

- 0 : ウォッチドッグは無効です。
- 1 : ウォッチドッグは有効です。

ビット 6:0 **T[6:0]** : 7 ビットカウンタ (MSB から LSB まで)

これらのビットは、ウォッチドッグカウンタの値を含みます。(4096 x 2<sup>WDGTB[1:0]</sup> PCLK サイクルごとにデクリメントされます。0x40 に達して 0x3F に戻ると (T6 がクリアされると)、リセットが生成されます。

## 28.4.2 設定レジスタ (WWDG\_CFR)

アドレスオフセット : 0x04

リセット値 : 0x0000 007F

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

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **EWI** : 早期ウェイクアップ割り込み

このビットがセットされているときには、カウンタの値が 0x40 に達したときに割り込みが発生します。この割り込みは、リセット後にハードウェアによってのみクリアされます。

ビット 8:7 **WDGTB[1:0]** : タイマーベース

プリスケアラのタイムベースは、次のように変更できます。

00 : CK カウンタクロック (PCLK/4096) 1 分周

01 : CK カウンタクロック (PCLK/4096) 2 分周

10 : CK カウンタクロック (PCLK/4096) 4 分周

11 : CK カウンタクロック (PCLK/4096) 8 分周

ビット 6:0 **W[6:0]** : 7 ビットウィンドウ値

これらのビットは、ダウンカウンタと比較されるウィンドウ値を含みます。

## 28.4.3 ステータスレジスタ (WWDG\_SR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EWIF														
															rc_w0

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EWIF** : 早期ウェイクアップ割り込みフラグ

このビットは、カウンタの値が 0x40 に達したときにハードウェアによってセットされます。“0”を書き込んでソフトウェアでクリアする必要があります。“1”を書き込んでも、ビットの値は変化しません。このビットは、割り込みが有効でない場合にもセットされます。

## 28.4.4 WWDG レジスタマップ

次の表に、WWDG のレジスタマップとリセット値を示します。

表 145. WWDG レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	WWDG_CR	Res.	WDGA	T[6:0]																														
	リセット値																									0	1	1	1	1	1	1	1	
0x04	WWDG_CFR	Res.	EWI	WDGTB1	W[6:0]																													
	Reset value																							0	0	0	0	0	1	1	1	1	1	1
0x08	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF																								
	リセット値																																	0

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 29 リアルタイムクロック (RTC)

### 29.1 概要

RTC は、あらゆる低電力モードを管理する自動ウェイクアップ機能を提供します。

本リアルタイムクロック (RTC) は、独立した BCD タイマ/カウンタです。RTC は、プログラム可能なアラーム割り込み機能を備えた時刻クロック/カレンダーを搭載しています。

また、割り込み機能を備えたプログラム可能な周期的ウェイクアップフラグも搭載しています。

2 つの 32 ビットレジスタには、2 進化 10 進数形式 (BCD) で表現した秒、分、時 (12 時間または 24 時間形式)、曜日、日、月、年が含まれています。サブセカンドの値もバイナリ形式で利用できます。

28 日、29 日 (うるう年)、30 日、31 日の補正は、自動的に行われます。サマータイム補正も行われます。

サブセカンド、秒、分、時、曜日、日付のプログラム可能なアラームを備えた 32 ビットレジスタが追加されています。

クリスタルオシレータ精度の偏差を補正するために、デジタル較正機能が利用可能です。

Backup ドメインリセット後、すべての RTC レジスタは、起こりうる不要な書き込みアクセスから保護されます。

供給電圧が動作範囲内にある間は、デバイスのステータス (実行モード、低電力モード、またはリセット中) に関係なく、RTC が停止することはありません。

### 29.2 RTC の主な機能

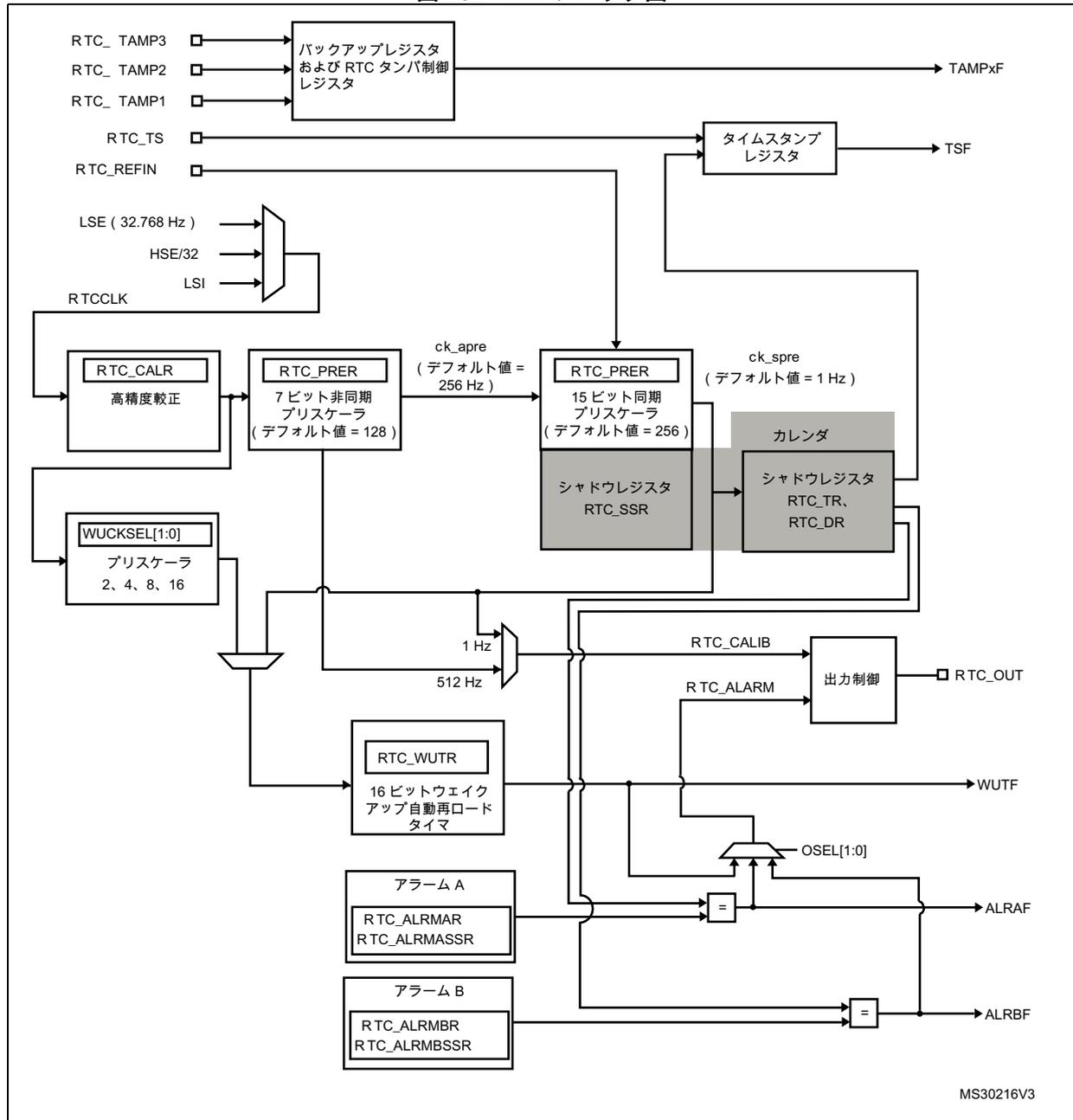
RTC ユニットの主な機能を以下に示します (図 282 : RTC ブロック図を参照)。

- サブセカンド、秒、分、時 (12 または 24 時間形式)、曜日、日、月、年に対応するカレンダー。
- ソフトウェアでプログラム可能なサマータイム補正。
- 割り込み機能を備えたプログラム可能なアラーム。アラームは、カレンダー項目のどの組み合わせでもトリガ可能。
- 自動ウェイクアップ割り込みをトリガする周期的なフラグを生成する自動ウェイクアップユニット。
- リファレンスクロック検出: より正確な秒のクロックソース (50 または 60 Hz) の使用で、カレンダーの精度を向上。
- サブセカンドシフト機能を使用する外部クロックとの正確な同期。
- デジタル較正回路 (周期的なカウンタ修正): 数秒の較正時間範囲で得られる 0.95 ppm の精度。
- イベントを保存するタイムスタンプ機能。
- 設定可能なフィルタおよび内部ブルアップのあるタンパ検出イベント。
- マスク可能な割り込み/イベント:
  - アラーム A
  - アラーム B
  - ウェイクアップ割り込み
  - タイムスタンプ
  - タンパ検出
- 32 バックアップレジスタ。

## 29.3 RTC の機能説明

### 29.3.1 RTC ブロック図

図 282. RTC ブロック図



RTC には次の要素が含まれます。

- 2本のアラーム
- 3つのタンパイベント
- 32 x 32 ビットバックアップレジスタ
  - バックアップレジスタ (RTC\_BKPxR) は、VDD 電源が遮断された場合に VBAT によって電源が供給される、RTC ドメインに搭載されています。
- オルタネート機能出力：次の2つの出力のうち1つを選択する RTC-OUTです。
  - RTC\_CALIB：512 Hz または 1Hz のクロック出力 (LSE 周波数 32.768 kHz の場合)。この出力は、RTC\_CR レジスタの COE ビットをセットして有効にします。
  - RTC\_ALARM：この出力は、アラーム A、アラーム B、またはウェイクアップ出力を選択する RTC\_CR レジスタの OSEL[1:0] ビットを設定することで有効にします。
- オルタネート機能入力：
  - RTC\_TS：タイムスタンプイベント
  - RTC\_TAMP1：タンパ1 イベント検出
  - RTC\_TAMP2：タンパ2 イベント検出
  - RTC\_TAMP3：タンパ3 イベント検出
  - RTC\_REFIN：50 または 60 Hz のリファレンスクロック入力

### 29.3.2 RTC によって制御される GPIO

RTC\_OUT、RTC\_TS、および RTC\_TAMP1 は、同一ピン (PC13) 上に配置されます。PC13 ピンの設定は、PC13 GPIO の設定にかかわらず、RTC によって制御されます。PC13 に配置された RTC 機能はすべての低電力モードおよび VBAT モードで使用できます。

出力形式は [表 146](#) に示す優先順位に従います。

表 146. RTC ピン PC13 の設定<sup>(1)</sup>

PC13 ピンの設定と機能	OSEL[1:0] ビット (RTC_ALARM 出力イネーブル)	COE ビット (RTC_CALIB 出力イネーブル)	RTC_ALARM _TYPE ビット	TAMP1E ビット (RTC_TAMP1 入力イネーブル)	TSE ビット (RTC_TS 入力イネーブル)	TSINSEL ビット
RTC_ALARM 出力 OD	01、10、または 11	無視	0	無視	無視	無視
RTC_ALARM 出力 PP	01、10、または 11	無視	1	無視	無視	無視
RTC_CALIB 出力 PP	00	1	無視	無視	無視	無視
RTC_TAMP1 入力フローティング	00	0	無視	1	0	無視
	00	1				
	01、10、または 11	0				

表 146. RTC ピン PC13 の設定<sup>(1)</sup> (続き)

PC13 ピンの設定と機能	OSEL[1:0] ビット (RTC_ALARM 出力イネーブル)	COE ビット (RTC_CALIB 出力イネーブル)	RTC_ALARM _TYPE ビット	TAMP1E ビット (RTC_TAMP1 入力イネーブル)	TSE ビット (RTC_TS 入力イネーブル)	TSINSEL ビット
RTC_TS および RTC_TAMP1 入力フローティング	00	0	無視	1	1	00
	00	1				
	01、10、または 11	0				
RTC_TS 入力フローティング	00	0	無視	0	1	00
	00	1				
	01、10、または 11	0				
ウェイクアップピンまたは標準 GPIO	00	0	無視	0	0	無視

1. OD : オープンドレイン、PP : プッシュプル

RTC\_TAMP2 および RTC\_TS は、同一ピン (PI8) 上に配置されます。PI8 の設定は、PI8 GPIO の設定にかかわらず、RTC によって制御されます。PI8 に配置された RTC 機能はすべての低電力モードおよび VBAT モードで使用できます。

出力形式は 表 147 に示す優先順位に従います。

表 147. RTC ピン PI8 の設定

PI8 ピンの設定と機能	TAMP2E ビット (RTC_TAMP2 入力イネーブル)	TSE ビット (RTC_TS 入力イネーブル)	TSINSEL ビット (タイムスタンプピン選択)
RTC_TAMP2 入力フローティング	1	0	無視
RTC_TS および RTC_TAMP2 入力フローティング	1	1	01
RTC_TS 入力フローティング	0	1	01
ウェイクアップピンまたは標準 GPIO	0	0	無視

RTC\_TAMP3 および RTC\_TS は、同一ピン (PC1) 上に配置されます。PC1 の設定は、PC1 GPIO の設定にかかわらず、RTC によって制御されます。PC1 に配置された RTC 機能はすべての低電力モードで使用できますが、VBAT モードでは使用できません。

表 148. RTC ピン PC2 の設定

PC2 ピンの設定と機能	TAMP2E ビット (RTC_TAMP3 入力フローティング)	TSE ビット (RTC_TS 入力フローティング)	TSINSEL ビット (タイムスタンプ ピン選択)
RTC_TAMP3 入力フローティング	1	0	無視
RTC_TS および RTC_TAMP3 入力フローティング	1	1	10 または 11
RTC_TS 入力フローティング	0	1	01
ウェイクアップピン または標準 GPIO	0	0	無視

RTC\_REFIN は PB15 に配置されます。RTC\_REFIN 機能を使用するには、PB15 をオルタネート機能モードで設定する必要があります。RTC\_REFIN は、VBAT モードおよび STANDBY モードでは使用できません。

次の表は、すべてのモードでの RTC ピンと機能の一覧です。

表 149. RTC 機能 (モード共通)

ピン	RTC 機能	すべての低電力モードで 使用可能な機能 (STANDBY モードを除く)	STANDBY モードで 使用可能な機能	VBAT モードで 使用可能な機能
PC13	RTC_TAMP1 RTC_TS RTC_OUT	○	○	○
PI8	RTC_TAMP2 RTC_TS	○	○	○
PC1	RTC_TAMP3 RTC_TS	○	○	X
PB15	RTC_REFIN	○	X	X

### 29.3.3 クロックとプリスケアラ

RTC クロックソース (RTCCLK) は、LSE クロック、LSI オシレータクロック、HSE クロックのうちから、クロックコントローラを介して選択されます。RTC クロックソースの設定に関する詳細は、[セクション 5 : リセットおよびクロック制御 \(RCC\)](#) を参照してください。

プログラム可能なプリスケアラステージで、カレンダーの更新に使用する 1 Hz のクロックを生成します。消費電力を最少に抑えるため、プリスケアラは以下に示す 2 つのプログラム可能なプリスケアラに分割されます ([図 282 : RTC ブロック図](#)を参照)。

- RTC\_PRER レジスタの PREDIV\_A ビットで設定される 7 ビットの非同期プリスケアラ
- RTC\_PRER レジスタの PREDIV\_S ビットで設定される 15 ビットの同期プリスケアラ

**注 :** **両方のプリスケアラを使用する場合は、非同期プリスケアラを高い値に設定して消費を最低限に抑えることをお勧めします。**

LSE 周波数 32.768 kHz で 1 Hz (ck\_spre) の内部クロック周波数を得るため、非同期プリスケアラ分周比は 128、同期プリスケアラの分周比は 256 に設定されます。

最低分周比は 1、最大分周比は  $2^{22}$  です。

これは、約 4 MHz の最大入力周波数に相当します。

$f_{ck\_apre}$  は、次の式で与えられます。

$$f_{CK\_APRE} = \frac{f_{RTCCLK}}{PREDIV\_A + 1}$$

ck\_apre クロックは、サブセカンドダウンカウンタであるバイナリ RTC\_SSR にクロックを供給するために使用されます。値がゼロになると、RTC\_SSR は、PREDIV\_S の内容で再ロードされます。

$f_{ck\_apre}$  は、次の式で与えられます。

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

ck\_spre クロックは、カレンダーの更新に、または 16 ビットウェイクアップ自動再ロードタイマのタイムベースとして使用できます。短いタイムアウト期間を得るため、16 ビットウェイクアップ自動再ロードタイマを、プログラム可能な 4 ビット非同期プリスケアラで分周した RTCCLK で動作させることもできます (詳細は [セクション 29.3.6 : 周期的自動ウェイクアップ](#) を参照)。

## 29.3.4 リアルタイムクロックとカレンダー

RTC カレンダーの時刻および日付レジスタには、PCLK (APB クロック) と同期するシャドウレジスタからアクセスします。同期持続の待ち時間を避けるため、これらのレジスタに直接アクセスすることもできます。

- サブセカンド用 RTC\_SSR
- 時刻用 RTC\_TR
- 日付用 RTC\_DR

RTCCLK 2 サイクルごとに現在のカレンダー値がシャドウレジスタにコピーされ、RTC\_ISR レジスタの RSF ビットがセットされます ([セクション 29.6.4 : RTC 初期化とステータスレジスタ \(RTC\\_ISR\)](#) を参照)。STOP モードおよび STANDBY モードでは、コピーは行われません。これらのモードが終了すると、RTCCLK 2 サイクル以内にシャドウレジスタが更新されます。

アプリケーションが、カレンダーレジスタを読み出す際、実際にはシャドウレジスタの内容にアクセスします。RTC\_CR レジスタの BYPSHAD 制御ビットをセットすることにより、カレンダーレジスタに直接アクセスできます。デフォルトでは、このビットはクリアされており、ユーザはシャドウレジスタにアクセスします。

RTC\_SSR、RTC\_TR または RTC\_DR レジスタを BYPSHAD = 0 の状態で読み出す際は、APB クロックの周波数 ( $f_{APB}$ ) は、RTC クロック ( $f_{RTCCLK}$ ) の周波数の 7 倍以上でなければなりません。

シャドウレジスタは、システムリセットによってリセットされます。

## 29.3.5 プログラム可能なアラーム

RTC ユニットは、以下に示すプログラム可能なアラーム、アラーム A およびアラーム B を搭載しています。以下に示すのはアラーム A に関する説明ですが、アラーム B についても同様です。

プログラム可能なアラーム機能は、RTC\_CR レジスタの ALRAE ビットを通じて有効にします。ALRAF は、カレンダーのサブセカンド、秒、分、時、日または曜日がそれぞれアラームレジスタ RTC\_ALRMASR および RTC\_ALRMAR にプログラムされている値と一致する場合は 1 にセットされます。各カレンダー項目は、RTC\_ALRMAR レジスタの MSKx ビットおよび RTC\_ALRMASR レジスタの MASKSSx ビットで個別に選択できます。アラームの割り込みは、RTC\_CR レジスタの ALRAIE ビットを通じて有効にします。

**注意：** *秒の項目が選択されている (RTC\_ALRMAR で MSK1 ビットがリセットされている) 場合、正しい動作を保証するため、RTC\_PRER レジスタでセットされる同期プリスケアラの分周比は 3 以上でなければなりません。*

アラーム A および アラーム B (RTC\_CR レジスタの OSEL[0:1] ビットで有効になっている場合) は、RTC\_ALARM 出力に送ることができます。RTC\_ALARM 出力の極性は、RTC\_CR レジスタの POL ビットを通じて設定できます。

## 29.3.6 周期的自動ウェイクアップ

周期的ウェイクアップフラグは、16 ビットのプログラム可能な自動再ロードダウンカウンタによって生成されます。ウェイクアップタイマの範囲は 17 ビットまで拡張できます。

ウェイクアップ機能は、RTC\_CR レジスタの WUTE ビットを通じて有効にします。

ウェイクアップタイマクロック入力には、次のものが使用できます。

- 2、4、8、または 16 分周した RTC クロック (RTCCLK)  
RTCCLK が LSE (32.768kHz) である場合、最小分解能 61  $\mu$ s で、ウェイクアップ割り込み周期を 122  $\mu$ s から 32 s の範囲で設定できます。
- ck\_spre (通常は 1 Hz の内部クロック)  
ck\_spre 周波数が 1 Hz の場合、1 秒の分解能でウェイクアップ時間を 1 秒からおよそ 36 時間までの範囲で設定できます。このプログラム可能な広い時間範囲は、2 つの部分に分かれます。
  - WUCKSEL [2:1] = 10 の場合は 1 秒から 18 時間、
  - WUCKSEL [2:1] = 11 の場合は約 18 時間から 36 時間です。後者の場合、16 ビットカウンタの現在値に 216 が加算されます。初期化シーケンスが完了すると ([ウェイクアップタイマのプログラミング \(882 ページ\)](#) を参照)、タイマがカウントダウンを開始します。ウェイクアップ機能が有効な場合、低電力モードでもカウントダウンはアクティブのままとなります。さらに、カウンタがゼロに到達すると、RTC\_ISR レジスタの WUTF フラグがセットされ、ウェイクアップカウンタが再ロード値 (RTC\_WUTR レジスタ値) で自動的に再ロードされます。

その後、WUTF フラグはソフトウェアでクリアする必要があります。

RTC\_CR2 レジスタの WUTIE ビットをセットして周期的ウェイクアップ割り込みを有効にすると、デバイスは低電力モードを終了できます。

周期的なウェイクアップフラグは、RTC\_CR レジスタの OSEL[0:1] ビットを通じて有効になっている場合に限り、RTC\_ALARM 出力に送ることができます。RTC\_ALARM 出力の極性は、RTC\_CR レジスタの POL ビットを通じて設定できます。

低電力モード (SLEEP、STOP、STANDBY) と同様に、システムリセットもウェイクアップタイマには影響しません。

## 29.3.7 RTC の初期化と設定

### RTC レジスタアクセス

RTC レジスタは、32 ビットのレジスタです。APB インターフェイスは、RTC レジスタアクセスに 2 ウェイトステートを挿入します。ただし、BYPHAD = 0 のときのカレンダーシャドウレジスタへの読み出しアクセスは除きます。

### RTC レジスタ書き込み保護

システムリセット後、RTC レジスタは、PWR\_CR1 レジスタの DBP ビットをクリアすることによって、不要な書き込みアクセスから保護されます（電源制御のセクションを参照）。RTC レジスタ書き込みアクセスを可能にするには、DBP ビットをセットする必要があります。

Backup ドメインリセット後、すべての RTC レジスタは書き込み保護されます。RTC レジスタへの書き込みは、書き込み保護レジスタ RTC\_WPR にキーを書き込むことにより有効になります。

RTC\_TAMPCR、RTC\_BKPxR、RTC\_OR、および RTC\_ISR[13:8] を除くすべての RTC レジスタの書き込み保護を解除するには、次のステップが必要です。

1. RTC\_WPR レジスタに“0xCA”を書き込みます。
2. RTC\_WPR レジスタに“0x53”を書き込みます。

誤ったキーを書き込むと、書き込み保護が再度アクティブになります。

保護メカニズムは、システムリセットの影響を受けません。

### カレンダーの初期化と設定

時間形式やプリスケアラ設定を含むカレンダー時刻と日付の初期値をプログラムするには、次のシーケンスが必要です。

1. RTC\_ISR レジスタで INIT ビットを 1 にセットして、初期化モードに入ります。このモードでは、カレンダーカウンタが停止し、その値を更新することができます。
2. RTC\_ISR レジスタの INITF ビットをポーリングします。INITF が 1 にセットされると、初期化フェーズモードに入ります。これには RTCCLK クロック約 2 サイクルを必要とします（クロック同期のため）。
3. カレンダーカウンタのための 1 Hz クロックを生成するには、RTC\_PRER レジスタで両方のプリスケアラ分周比をプログラムします。
4. シャドウレジスタ (RTC\_TR および RTC\_DR) に時刻と日付の初期値をロードし、RTC\_CR レジスタの FMT ビットを介して時間形式 (12 時間または 24 時間) を設定します。
5. INIT ビットをクリアして初期化モードを終了します。その後、カレンダーカウンタの実際の値が自動的にロードされ、4 RTCCLK クロックサイクル後にカウントが再開します。

初期化シーケンスが完了すると、カレンダーがカウントを開始します。

**注：** システムリセット後、アプリケーションは RTC\_ISR レジスタの INITF フラグを読み出し、カレンダーが初期化されたか否かを確認できるようになります。このフラグが 0 であれば、カレンダーの年の項目が Backup ドメインリセットデフォルト値 (0x00) にセットされているため、初期化されていません。初期化後にカレンダーを読み出すには、まずソフトウェアで RTC\_ISR レジスタの RSF フラグがセットされていることを確認する必要があります。

## サマータイム

サマータイム管理は、RTC\_CR レジスタの SUB1H ビット、ADD1H ビット、BKP ビットを介して行われます。

SUB1H または ADD1H を使用すると、ソフトウェアは初期化手順を踏まずに 1 度の操作で、カレンダーから 1 時間引いたり足したりすることができます。

さらに、ソフトウェアは BKP ビットを使用してこの操作を記憶することができます。

## アラームのプログラミング

プログラム可能なアラームをプログラムまたは更新するには、同様な手順を踏む必要があります。以下に示すのはアラーム A の手順ですが、アラーム B についても同様です。

1. RTC\_CR の ALRAE をクリアしてアラーム A を無効にします。
2. アラーム A レジスタ (RTC\_ALRMSSR/RTC\_ALRMAR) をプログラムします。
3. RTC\_CR レジスタで ALRAE をセットしてアラーム A を再び有効にします。

**注：** RTC\_CR レジスタの各変更は、クロック同期のため RTCCLK クロック約 2 サイクル後に有効になりません。

## ウェイクアップタイマのプログラミング

ウェイクアップタイマ自動再ロード値 (RTC\_WUTR の WUT[15:0]) の設定または変更には、次の手順が必要です。

1. RTC\_CR の WUTE をクリアしてウェイクアップタイマを無効にします。
2. RTC\_ISR の WUTWF がセットされ、ウェイクアップ自動再ロードカウンタおよび WUCKSEL[2:0] ビットへのアクセスが許可されていることが確認されるまで WUTWF をポーリングします。これには RTCCLK クロック約 2 サイクルを必要とします (クロック同期のため)。
3. ウェイクアップ自動再ロード値 WUT[15:0] およびウェイクアップクロック選択 (RTC\_CR の WUCKSEL[2:0] ビット) をプログラムします。RTC\_CR で WUTE をセットしてタイマを再び有効にします。ウェイクアップタイマがカウントダウンを再開します。WUTWF ビットは、クロックの同期化により、WUTE クリア後、2 RTCCLK クロックサイクルまでクリアされます。

## 29.3.8 カレンダーの読み出し

### RTC\_CR レジスタの BYPSHAD 制御ビットがクリアされている場合

RTC カレンダーレジスタ (RTC\_SSR、RTC\_TR、および RTC\_DR) を正しく読み出すには、APB クロック周波数 ( $f_{PCLK}$ ) が RTC クロック周波数 ( $f_{RTCCLK}$ ) の 7 倍以上でなければなりません。これにより、同期メカニズムの安全な動作が保証されます。

APB クロック周波数が RTC クロック周波数の 7 倍未満である場合、ソフトウェアによってカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC\_TR の 2 回目の読み出しが 1 回目の読み出しと同じ結果であれば、データが正しいことが保証されます。同じでない場合は、3 回目の読み出しアクセスを行う必要があります。どの場合も、APB クロック周波数は必ず RTC クロック周波数以上でなければなりません。

RTC\_ISR レジスタの RSF ビットは、カレンダーレジスタが RTC\_SSR、RTC\_TR、および RTC\_DR シャドウレジスタにコピーされるたびにセットされます。コピーは、2 RTCCLK サイクルごとに行われます。3 つの値における一貫性を保証するため、RTC\_SSR または RTC\_TR のどちらかを読み出すと、高次カレンダーシャドウレジスタの値は RTC\_DR が読み出されるまでロックされます。ソフトウェアが 2 RTCCLK サイクル未満の間隔でカレンダーの読み出しアクセスを行う場合、最初のカレンダー読み出し後に RSF をソフトウェアでクリアする必要があり、その後ソフトウェアは、RSF ビットがセットされるまで待つてから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを再読み出す必要があります。

低電力モード (STOP または STANDBY) からのウェイクアップ後は、RSF をソフトウェアでクリアする必要があります。その後、ソフトウェアは、いま一度 RSF がセットされるまで待つてから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを再読み出す必要があります。

RSF ビットは、ウェイクアップ後にクリアする必要がありますが、低電力モードに入る前には、その必要はありません。

システムリセット後、ソフトウェアは RSF がセットされるまで待つてから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを読み出す必要があります。実際、システムリセットがかかると、シャドウレジスタはデフォルト値にリセットされます。

初期化 ([カレンダーの初期化と設定 \(881 ページ\)](#)) を参照) 後、ソフトウェアは RSF がセットされるまで待つてから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを読み出す必要があります。

同期 ([セクション 29.3.10 : RTC の同期](#)) を参照) 後、ソフトウェアは RSF がセットされるまで待つてから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを読み出す必要があります。

## RTC\_CR レジスタ (バイパスシャドウレジスタ) の BYPSHAD 制御ビットがセットされている場合

カレンダーレジスタを読み出すと、カレンダーカウンタの値が直接与えられるため、RSF ビットがセットされるのを待つ必要はありません。シャドウレジスタは低電力モード (STOP または STANDBY) では更新されないため、低電力モード終了後に特にこのような読み出しが有用です。

BYPSHAD ビットが 1 にセットされている場合、レジスタへの 2 回の読み出しアクセス間で RTCCLK エッジが発生した場合は、さまざまなレジスタ間で互いに不整合が起きる場合があります。さらに、読み出し操作中に RTCCLK エッジが発生した場合、レジスタの 1 つが不正な値となる場合があります。ソフトウェアはすべてのレジスタを 2 回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要があります。その代わりに、ソフトウェアはカレンダーレジスタの最下位の数値を 2 回比較するだけで構いません。

**注 :** *BYPSHAD = 1 の間、カレンダーレジスタの読み出し命令が完了するには 1 APB サイクルだけ余計に必要となります。*

## 29.3.9 RTC のリセット

カレンダーシャドウレジスタ (RTC\_SSR、RTC\_TR、RTC\_DR) および RTC ステータスレジスタ (RTC\_ISR) の一部のビットは、利用可能なすべてのシステムリセットリソースによってデフォルト値にリセットされます。

逆に、次のレジスタは Backup ドメインリセットによってそれぞれのデフォルト値にリセットされ、システムリセットの影響を受けません。RTC の現在のカレンダーレジスタ、RTC 制御レジスタ (RTC\_CR)、プリスケールレジスタ (RTC\_PRER)、RTC 較正レジスタ (RTC\_CALR)、RTC シフトレジスタ (RTC\_SHIFTR)、RTC タイムスタンプレジスタ (RTC\_TSSSR、RTC\_TSTR および RTC\_TSDR)、RTC タンパおよびオルタネート機能設定レジスタ (RTC\_TAMPCR)、RTC バックアップレジスタ (RTC\_BKPxR)、ウェイクアップタイマレジスタ (RTC\_WUTR)、アラーム A とアラーム B レジスタ (RTC\_ALRMSSR/RTC\_ALRMAR と RTC\_ALRMBSSR/RTC\_ALRMBR)、およびオプションレジスタ (RTC\_OR)。

さらに、RTC はリセットソースが Backup ドメインリセットと異なる場合は、システムリセット中も動作し続けます。Backup ドメインリセットが発生すると、RTC は停止し、すべての RTC レジスタがリセット値にセットされます。



## 29.3.10 RTC の同期

RTC は、高精度でリモートクロックと同期できます。サブセカンド項目 (RTC\_SSR または RTC\_TSSSR) を読み出すと、リモートクロックによって維持されている時刻と RTC 間の正確なオフセットが計算できます。その後、RTC\_SHIFTR を使用してほんの一瞬クロックを「シフト」することによって RTC を調整し、このオフセットを取り除くことができます。

RTC\_SSR には、同期プリスケアラのカウンタの値が入っています。これにより、RTC によって維持されている正確な時刻を  $1/(\text{PREDIV\_S} + 1)$  秒の分解能まで計算することができます。その結果、同期プリスケアラ値 (PREDIV\_S[14:0]) を増加させることにより分解能を改善できます。許可されている最大分解能 (32768 Hz クロックで 30.52  $\mu\text{s}$ ) は、PREDIV\_S を 0x7FFF にセットすることにより得られます。

ただし、PREDIV\_S を増加させるということは、同期プリスケアラの出力を 1 Hz に維持するため PREDIV\_A を減らす必要があることを意味します。このように、非同期プリスケアラの出力周波数が増加すると、RTC の動的消費電力が増加する場合があります。

RTC は、RTC シフト制御レジスタ (RTC\_SHIFTR) を使って微調整できます。RTC\_SHIFTR に書き込むことにより、 $1/(\text{PREDIV\_S} + 1)$  秒の分解能で、クロックを最大 1 秒だけシフト (遅れ/進み) させることができます。このシフト操作の本質は、同期プリスケアラのカウンタ SS[15:0] に SUBFS[14:0] 値を加算することであり、この操作はクロックを遅らせることとなります。同時に ADD1S ビットがセットされた場合、1 秒追加すると同時に秒の小数部を差し引くことになるため、クロックを進めることとなります。

**注意：** シフト操作を始める前に、ユーザは SS[15] = 0 であることを確認し、オーバーフローが発生しないようにする必要があります。

RTC\_SHIFTR レジスタへの書き込みによってシフト操作が始まるとすぐに、シフト操作が保留中であることを示す SHPF フラグがハードウェアによってセットされます。このビットは、シフト操作が完了するとすぐに、ハードウェアによってクリアされます。

**注意：** この同期機能はリファレンスクロック検出機能とは両立できません。具体的には、REFCKON = 1 のときにファームウェアから RTC\_SHIFTR への書き込みはできません。

## 29.3.11 RTC リファレンスクロック検出

RTC カレンダの更新は、リファレンスクロックである RTC\_REFIN に同期させることができます。通常は商用電源 (50 または 60 Hz) です。RTC\_REFIN リファレンスクロックには、32.768 kHz LSE クロックより高い精度が必要です。RTC\_REFIN 検出が有効になると (RTC\_CR の REFCKON ビットが 1 にセット)、カレンダは引き続き LSE クロックによって駆動されますが、RTC\_REFIN を使用してカレンダ更新周波数 (1 Hz) の不正確さを補正します。

各 1 Hz クロックエッジは、一番近い RTC\_REFIN クロックエッジ (所与の時間枠内に見つかった場合) と比較されます。ほとんどの場合、2 つのクロックエッジは正しく整列しています。LSE クロックが不正確なために 1 Hz のクロックがずれた場合、RTC は 1 Hz のクロックを少しシフトさせ、その後の 1 Hz のクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダはリファレンスクロックと同様に正確になります。

RTC は、32.768 kHz クォーツから生成される 256 Hz クロック (ck\_apre) を使用して、リファレンスクロックソースがあるかどうかを検出します。検出は各カレンダ更新 (1 秒ごと) 程度の時間枠で行われます。最初のリファレンスクロックエッジを検出する際、この時間枠は ck\_apre 7 周期に等しくなります。その後のカレンダ更新では、ck\_apre 3 周期より短い時間枠が使用されます。

リファレンスクロックがこの時間枠内で検出されるたびに、ck\_apre クロックを出力する非同期プリスケアラは強制的に再ロードされます。プリスケアラは同時に再ロードされるので、リファレンスクロックおよび 1 Hz のクロックが整列するタイミングには影響しません。クロックが整列していない

場合、後の 1 Hz クロックエッジは、リファレンスクロックと整列するように再ロードによって少しシフトされます。

リファレンスクロックが停止した (ck\_apre 3 周期の枠内でリファレンスクロックエッジが発生しない) 場合、カレンダーは LSE クロックのみを基準にして更新が継続されます。その後 RTC は ck\_spre エッジを中心として ck\_apre 7 周期という広い検出時間枠でリファレンスクロックを待ちます。

基準クロック検出を有効にした場合、PREDIV\_A および PREDIV\_S を以下に示すそれぞれのデフォルト値にセットする必要があります。

- PREDIV\_A = 0x007F
- PREDIV\_S = 0x00FF

**注:** *RTC\_REFIN クロック検出は、STANDBY モードでは利用できません。*

## 29.3.12 RTC の高精度デジタル較正

RTC 周波数の精度は、-487.1~+488.5 ppm の範囲で、分解能約 0.954 ppm でデジタル的に較正できます。周波数の修正は、一連の微調整 (個々の RTCCLK パルスの追加や削除) によって行われます。このような調整は、短い期間で観測された場合でも RTC が十分に較正されるように、かなり広範に分散して行われます。

この高精度デジタル較正は、入力周波数が 32768 Hz の場合、RTCCLK 約 2<sup>20</sup> パルスのサイクルまたは 32 秒の間で行われます。このサイクルは、RTCCLK によって駆動される 20 ビットカウンタ、cal\_cnt[19:0] によって維持されます。

高精度較正レジスタ (RTC\_CALR) によって、32 秒サイクル中にマスクされる RTCCLK クロックサイクル数を指定します。

- CALM[0] ビットを 1 にセットすると、32 秒サイクルの中でちょうど 1 パルスがマスクされます。
- CALM[1] ビットを 1 にセットすると、さらに 2 サイクルがマスクされます。
- CALM[2] ビットを 1 にセットすると、さらに 4 サイクルがマスクされます。
- CALM[8] ビットを 1 にセットするまで続けると、256 クロックがマスクされます。

**注:** *CALM[8:0] (RTC\_CALR) によって、32 秒サイクル中にマスクされる RTCCLK パルス数を指定します。CALM[0] ビットを 1 にセットすると、cal\_cnt[19:0] = 0x80000 になった時点で、32 秒サイクル中でちょうど 1 パルスがマスクされます。CALM[1] = 1 では、さらに 2 サイクルがマスクされ (cal\_cnt = 0x40000 および 0xC0000)、CALM[2] = 1 では、さらに 4 サイクルがマスクされ (cal\_cnt = 0x20000/0x60000/0xA0000/0xE0000)、CALM[8] = 1 まで続けると、256 クロックがマスクされ (cal\_cnt = 0xXX800)。*

CALM では、細かい分解能で RTC 周波数を最大 487.1 ppm 負の方向に調整することができ、CALP ビットでは周波数を 488.5 ppm 正の方向に調整することができます。CALP を 1 にセットすることにより、実質上は、RTCCLK 2<sup>11</sup> サイクルごとに、RTCCLK パルスが 1 パルス追加で挿入されます。すなわち、32 秒サイクルごとに 512 クロックが追加されることとなります。

CALM を CALP と合わせて使用すると、32 秒サイクルの間に RTCCLK -511 から +512 サイクルまでのオフセットが追加でき、これは約 0.954 ppm の分解能で較正範囲 -487.1~+488.5 ppm に換算されます。

有効較正周波数 (FCAL) を入力周波数 (FRTCCLK) に対して求める計算式は次のとおりです。

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$



## PREDIV\_A < 3 の場合の較正

非同期プリスケアラ値 (RTC\_PRER レジスタの PREDIV\_A ビット) が 3 未満の場合、CALP ビットを 1 にセットすることはできません。CALP がすでに 1 にセットされていて、PREDIV\_A ビットが 3 未満の値にセットされた場合、CALP の設定値は無視され、CALP が 0 に設定された場合と同じように較正されます。

PREDIV\_A が 3 未満の状態では較正を実施するには、各秒のカウントが 8 RTCCLK クロックサイクル分早められるように同期プリスケアラ値 (PREDIV\_S) を小さくする必要があります。これは 32 秒毎に 256 クロックサイクル追加することに相当します。結果として、CALM ビットのみを使用して 32 秒サイクルの間に 255~256 クロックパルス (243.3~244.1 ppm の較正範囲に相当) を追加することができます。

公称 RTCCLK 周波数が 32768 Hz で、PREDIV\_A が 1 (分周比 2) の場合、PREDIV\_S を 16383 ではなく 16379 (4 少ない) にセットする必要があります。また、PREDIV\_A が 0 の場合、PREDIV\_S を 32767 ではなく 32759 (8 少ない) にセットする必要がありますので注意して下さい。

PREDIV\_S をこのように減少させた場合、較正された入力クロックの有効周波数の式は次のようになります。

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

この場合、RTCCLK が正確に 32768.00 Hz であれば、CALM[7:0] が 0x100 (CALM 設定範囲の中間値) と等しくなるのが正しい設定です。

## RTC 較正值の確認

RTC の精度は、RTCCLK の正確な周波数を測定し、正しい CALM 値および CALP 値を計算することにより保証されます。オプションの 1 Hz 出力が搭載されており、アプリケーションによって RTC 精度の測定と確認を行うことができます。

ある時間間隔で RTC の周波数を精密に測定すると、デジタル較正サイクルを測定周期とどのように合わせているかにより、測定期間中に最大 2 RTCCLK クロックサイクルの測定誤差が生じます。

ただし、この測定誤差は、測定周期が較正サイクル周期と同じ長さであれば排除できます。この場合、観測される唯一の誤差はデジタル較正の分解能による誤差となります。

- デフォルトでは、較正サイクル周期は 32 秒です。

このモードを使用して正確に 32 秒で 1 Hz 出力の精度を測定すると、その精度は 0.477 ppm (較正分解能の制限により 32 秒で 0.5 RTCCLK サイクル) 以内となることが保証されます。

- RTC\_CALR レジスタの CALW16 ビットを 1 にセットして、較正サイクル周期を強制的に 16 秒にすることができます。

この場合、RTC 精度は最大誤差 0.954 ppm (16 秒で 0.5 RTCCLK サイクル) で 16 秒間で測定できます。ただし、較正分解能が下がるため、長期的な RTC 精度もまた 0.954 ppm に下がります。CALW16 が 1 にセットされると、CALM[0] ビットは 0 のままとなります。

- RTC\_CALR レジスタの CALW8 ビットを 1 にセットして、較正サイクル周期を強制的に 8 秒にすることができます。

この場合、RTC 精度は最大誤差 1.907 ppm (8 秒で 0.5 RTCCLK サイクル) で 8 秒で測定できます。長期的な RTC 精度もまた 1.907 ppm に下がります。CALW8 が 1 にセットされると、CALM[1:0] ビットは 00 のままとなります。

## 動作中の再較正

次の処理を実施することにより、RTC\_ISR/INITF = 0の間でも、較正レジスタ (RTC\_CALR) を動作中に更新することができます。

1. RTC\_ISR/RECALPF (再較正保留フラグ) をポーリングします。
2. このフラグが 0 にセットされている場合は、必要に応じて新しい値を RTC\_CALR に書き込みます。すると、RECALPF が自動的に 1 にセットされます。
3. RTC\_CALR への書き込み動作後 ck\_apre 3 サイクル以内に、新しい較正設定が有効になります。

## 29.3.13 タイムスタンプ機能

タイムスタンプは、RTC\_CR レジスタの TSE または ITSE ビットを 1 にセットすることにより有効になります。

TSE がセットされている場合：

RTC\_TS ピンでタイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC\_TSSSR、RTC\_TSTR、RTC\_TSDR) にカレンダーが保存されます。

ITSE がセットされている場合：

内部タイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC\_TSSSR、RTC\_TSTR、RTC\_TSDR) にカレンダーが保存されます。内部タイムスタンプイベントは、VBAT 電源への切り替えによって生成されます。

また、タイムスタンプイベントが発生すると、内部または外部イベントによって、RTC\_ISR レジスタのタイムスタンプフラグビット (TSF) がセットされます。イベントが内部である場合は、ITSF フラグが RTC\_ISR レジスタにもセットされます。

RTC\_CR レジスタの TSIE ビットをセットすることにより、タイムスタンプイベントが発生したときに割り込みが生成されます。

タイムスタンプフラグ (TSF) がすでにセットされている間に新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) がセットされ、タイムスタンプレジスタ (RTC\_TSTR および RTC\_TSDR) は、その前のイベントの結果を維持します。

**注：** *同期処理のため、TSF はタイムスタンプイベント発生から ck\_apre 2 サイクル後にセットされます。一方、TSOVF のセットに遅延はありません。これは、2 つのタイムスタンプイベントの発生したタイミングが近い場合、TSF がまだ“0”であっても TSOVF が“1”と検出される可能性があることを意味します。よって、TSOVF のポーリングは TSF がセットされた後に実施することをお勧めします。*

**注意：** *TSF ビットがクリアされたと思われた直後にタイムスタンプイベントが発生した場合、TSF および TSOVF ビットの両方がセットされます。同時に発生するタイムスタンプイベントのマスキングを回避するため、TSF がすでに“1”と読み出されていない限りは、アプリケーションによる“0”クリアの処理を行ってはなりません。*

オプション機能として、タンパイイベントによってタイムスタンプイベントを記録することもできます。TAMPTS 制御ビットの詳細については、[セクション 29.6.14 : RTC タイムスタンプサブセカンドレジスタ \(RTC\\_TSSSR\)](#) を参照してください。

## 29.3.14 タンパ検出

RTC\_TAMPx 入力イベントは、フィルタを使用してエッジ検出またはレベル検出のどちらにも設定できます。

タンパ検出は次の目的で設定することができます。

- RTC バックアップレジスタ (デフォルト設定) を消去する。
- STOP モードおよび STANDBY モードからのウェイクアップが可能な割り込みを生成する。
- 低電力タイマのためのハードウェアトリガを生成する。

### RTC バックアップレジスタ

バックアップレジスタ (RTC\_BKPxR) は、システムリセットや STANDBY モードからのウェイクアップではリセットされません。

バックアップレジスタ (RTC\_BKPxR) は、タンパ検出イベント発生時にリセットされます ([セクション 29.6.20 : RTC バックアップレジスタ \(RTC\\_BKPxR\)](#) および [タンパ検出の初期化 \(888 ページ\)](#)) を参照。またはフラッシュの読み出し保護がレベル 1 からレベル 0 に変わったとき) が、TAMPxNOERASE ビットがセットされている場合や RTC\_TAMPCR レジスタの TAMPxMF がセットされている場合は例外です。

### タンパ検出の初期化

各入力は、RTC\_TAMPCR レジスタにおいて該当する TAMPxE ビットを 1 にセットすることにより有効にできます。

各 RTC\_TAMPx タンパ検出入力は、RTC\_ISR レジスタの TAMPxF フラグに関連付けられています。

TAMPxMF がクリアされている場合

TAMPxF フラグは、ピン上でタンパイベントが発生した後にアサートされます。その際の遅延時間を以下に示します。

- TAMPFLT = 0x0 以外の場合、ck\_apre 3 サイクル (フィルタを使ったレベル検出)
- TAMPTS = 1 の場合、ck\_apre 3 サイクル (タンパイベント時のタイムスタンプ)
- TAMPFLT = 0x0 (エッジ検出) および TAMPTS = 0 の場合、遅延なし

TAMPxF がセットされている場合に、この周期中に同一ピンで発生した新たなタンパイベントを検出することはできません。

TAMPxMF がセットされている場合

同一ピンで発生した新たなタンパイベントは、上述した遅延の間とさらに ck\_rtc 2.5 サイクルの間は検出できません。

RTC\_TAMPCR レジスタの TAMPIE ビットをセットすることにより、タンパ検出イベント発生時に割り込みが生成されます (TAMPxF がセットされている場合)。1 つ以上の S TAMPxMF がセットされている場合、TAMPIE をセットすることはできません。

TAMPIE がクリアされている場合は、各タンパピンイベント/割り込みは、RTC\_TAMPCR レジスタの対応する TAMPxIE ビットをセットすることによって個別に有効にすることができます。対応する TAMPxMF がセットされている場合、TAMPxIE をセットすることはできません。

## タンパイベント時のトリガ出力の生成

タンパイベント検出は、低電力タイマによるトリガ入力として使用できます。

RTC\_TAMPCR レジスタの TAMPxMF ビットがクリアされている場合、TAMPxF フラグをソフトウェアでクリアして、同一ピンで新たなタンパイベントを検出できるようにする必要があります。

TAMPxMF ビットがセットされている場合、TAMPxF フラグはマスクされ、RTC\_ISR レジスタでクリアされたままとなります。この設定により、TAMPxF をクリアするためにシステムをウェイクアップする必要もなく、STOP モードのまま低電力タイマを自動的にトリガすることができます。この場合、バックアップレジスタはクリアされません。

## タンパイベント時のタイムスタンプ

TAMPTS を“1”にセットすると、すべてのタンパイベントがタイムスタンプイベントを発生させるようになります。この場合、通常のタイムスタンプイベント発生時と同様に TSF ビットまたは TSOVF ビットが RTC\_ISR でセットされます。TSF または TSOVF がセットされるのと同時に、影響を受けるタンパフラグレジスタ、TAMPxF がセットされます。

## タンパ入力でのエッジ検出

TAMPFLT ビットが“00”の場合、該当する TAMPxTRG ビットに応じて立ち上がりエッジまたは立ち下がりエッジが観測されると、RTC\_TAMPx ピンがタンパ検出イベントを生成します。エッジ検出を選択すると、RTC\_TAMPx 入力の内部プルアップ抵抗が無効になります。

**注意：** エッジ検出を使用する場合は、タンパ検出の有効化直後のタンパピンのレベルをソフトウェアでチェックすることをお勧めします (GPIO レジスタを読み出す)。また、タンパイベントの検出を有効化する前にアクティブエッジが発生していないかどうかを確認するため、影響を受ける値をバックアップレジスタに書き込む前にソフトウェアでチェックすることをお勧めします。  
TAMPFLT="00" で TAMPxTRG = 0 (立ち上がりエッジ検出) のとき、タンパ検出を有効にする前にタンパ入力が入力にハイレベルにある場合は、タンパイベントをハードウェアで検出できます。

タンパイベントが検出されクリアされた後に、バックアップレジスタ (RTC\_BKPxR) を再プログラムする場合には、事前に、RTC\_TAMPx オルタネート機能を無効にしてから再度有効にする (TAMPxE を 1 にセット) 必要があります。これによって、RTC\_TAMPx 入力の値がタンパ検出を示している間に、アプリケーションがバックアップレジスタにデータを書き込むのを防ぎます。これは、RTC\_TAMPx オルタネート機能の入力でのレベル検出に相当します。

**注：** タンパ検出は、V<sub>DD</sub> 電源がオフのときでも有効です。バックアップレジスタの不必要なリセットを避けるには、RTC\_TAMPx オルタネート機能が設定されているピンを外部で適切な信号レベルに接続しておく必要があります。

## RTC\_TAMPx 入力でのフィルタを使ったレベル検出

フィルタを使ったレベル検出は、TAMPFLT を 0 以外の値にセットすることにより行われます。タンパ検出イベントは、(TAMPFLT に応じて) 2、4 または 8 回のいずれかの連続したサンプルが TAMPxTRG ビットで指定するレベルで観測されたときに生成されます。

RTC\_TAMPx 入力は、TAMPPUDIS が 1 にセットされて無効な状態になっていない限り、その状態がサンプリングされる前に I/O の内部プルアップ抵抗でプリチャージされています。プリチャージの継続時間は TAMPPRCH ビットによって決定され、RTC\_TAMPx 入力ピンにおけるより大きな容量を持たせることができます。

タンパ検出の遅延時間と、プルアップによる電力消費との間のトレードオフは、TAMPFREQ を使用してレベル検出のサンプリング周波数を決定することにより、最適化できます。

**注：** プルアップ抵抗の電気的特性については、データシートを参照してください。



## 29.3.15 較正クロック出力

RTC\_CR レジスタで COE ビットが 1 にセットされると、RTC\_CALIB デバイス出力にリファレンスクロックが供給されます。

RTC\_CR レジスタの COSEL ビットがリセットされ、かつ PREDIV\_A = 0x7F である場合、RTC\_CALIB 周波数は  $f_{RTCCLK}/64$  です。これは 32.768 kHz の RTCCLK 周波数に対する 512 Hz の較正出力に相当します。立ち下がリエッジには軽いジッタがあるため、RTC\_CALIB のデューティサイクルは不規則になります。したがって、立ち上がりエッジの使用が推奨されます。

COSEL がセットされ、かつ “PREDIV\_S+1” がゼロ以外の 256 の倍数である場合 (すなわち、PREDIV\_S[7:0] = 0xFF)、RTC\_CALIB 周波数は  $f_{RTCCLK}/(256 * (PREDIV_A+1))$  となります。これは、RTCCLK 周波数が 32.768 kHz で、プリスケラデフォルト値 (PREDIV\_A = 0x7F、PREDIV\_S = 0xFF) に対する 1 Hz の較正出力に相当します。

**注:** *RTC\_CALIB または RTC\_ALARM 出力が選択されると、RTC\_OUT ピンは自動的に出力オルタネート機能に設定されます。*

## 29.3.16 アラーム出力

RTC\_CR レジスタの OSEL[1:0] 制御ビットを使用してアラームオルタネート機能出力、RTC\_ALARM を有効にし、出力となる機能を選択します。これらの機能は、RTC\_ISR レジスタの該当するフラグの内容を反映します。

RTC\_CR の POL 制御ビットは、POL が 1 にセットされているときに選択されたフラグビットの逆が出力されるよう、出力の極性を決定します。

### アラームオルタネート機能出力

RTC\_ALARM ピンは、制御ビット RTC\_ALARM\_TYPE (RTC\_OR レジスタ) を使用して、出力オープンドレインまたは出力プッシュプルに設定できます。

**注:** *RTC\_ALARM 出力が有効になると、この設定は、RTC\_CALIB (COE ビットは無視され、クリアされたまま) の設定よりも優先されます。*

*RTC\_CALIB または RTC\_ALARM 出力が選択されると、RTC\_OUT ピンは自動的に出力オルタネート機能に設定されます。*

## 29.4 RTC 低電力モード

表 150. 低電力モードが RTC に与える影響

モード	説明
SLEEP	影響なし。 RTC 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスは STOP モードから復帰します。
STANDBY	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスは STANDBY モードから復帰します。

## 29.5 RTC 割り込み

すべての RTC 割り込みは、EXTI コントローラに接続されています。 [セクション 11 : 拡張割り込み / イベントコントローラ \(EXTI\)](#) を参照してください。

RTC アラーム割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで RTC アラームイベントに対応する EXTI ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で RTC\_ALARM IRQ チャンネルを設定し、有効にします。
3. RTC が RTC アラームを生成するように設定します。

RTC タンパ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで RTC タンパイベントに対応する EXTI ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で RTC\_TAMP\_STAMP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC タンパイベントを検出するように設定します。

RTC タイムスタンプ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで RTC タイムスタンプイベントに対応する EXTI ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で RTC\_TAMP\_STAMP IRQ チャンネルを設定し、有効にします。
3. RTC タイムスタンプイベントを検出するよう RTC を設定します。

ウェイクアップタイマ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードでウェイクアップタイマ割り込みイベントに対応する EXTI ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で RTC\_WKUP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC ウェイクアップタイマイベントを検出するように設定します。

表 151. 割り込み制御ビット

割り込みイベント	イベントフラグ	イネーブル制御ビット	SLEEP モードの終了	STOP モードの終了	STANDBY モードの終了
アラーム A	ALRAF	ALRAIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
アラーム B	ALRBF	ALRBIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TS 入力 (タイムスタンプ)	TSF	TSIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TAMP1 入力検出	TAMP1F	TAMPIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TAMP2 入力検出	TAMP2F	TAMPIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TAMP3 入力検出	TAMP3F	TAMPIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
ウェイクアップタイマ割り込み	WUTF	WUTIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>

1. STOP モードおよび STANDBY モードからのウェイクアップは、RTC クロックソースが LSE または LSI のときのみ可能です。

## 29.6 RTC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 1.1 \(59 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

### 29.6.1 RTC 時刻レジスタ (RTC\_TR)

RTC\_TR は、カレンダー時刻シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。 [カレンダーの初期化と設定 \(881 ページ\)](#) および [カレンダーの読み出し \(882 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、 [RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x00

Backup ドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 29.6.2 RTC 日付レジスタ (RTC\_DR)

RTC\_DR は、カレンダー日付シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。[カレンダーの初期化と設定 \(881 ページ\)](#) および [カレンダーの読み出し \(882 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x04

Backup ドメインリセット値 : 0x0000 2101

システムリセット : BYPSHAD = 0 の場合、0x0000 2101 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:20 **YT[3:0]** : BCD 形式での年の十の位

ビット 19:16 **YU[3:0]** : BCD 形式での年の一の位

ビット 15:13 **WDU[2:0]** : 曜日

- 000 : 禁止
- 001 : 月曜日
- ...
- 111 : 日曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

## 29.6.3 RTC 制御レジスタ (RTC\_CR)

アドレスオフセット : 0x08

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
							r/w	r/w	r/w	r/w	r/w	r/w	r/w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEEDGE	WUCKSEL[2:0]		
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ITSE** : 内部イベントでのタイムスタンプイネーブル

- 0 : 内部イベントでのタイムスタンプは無効です。
- 1 : 内部イベントでのタイムスタンプは有効です。

ビット 23 **COE** : 較正出カインーブル

- このビットは、RTC\_CALIB 出力を有効にします。
- 0 : 較正出力は無効です。
- 1 : 較正出力は有効です。

ビット 22:21 **OSEL[1:0]** : 出力選択

- これらのビットは、RTC\_ALARM 出力に送られるフラグの選択に使用します。
- 00 : 出力は無効です。
- 01 : アラーム A 出力は有効です。
- 10 : アラーム B 出力は有効です。
- 11 : ウェイクアップ出力は有効です。

ビット 20 **POL** : 出力極性

- このビットは、RTC\_ALARM 出力の極性の設定に使用します。
- 0 : ALRAF/ALRBF/WUTF がアサートされると、このピンがハイになります (OSEL[1:0] に応じて)。
- 1 : ALRAF/ALRBF/WUTF がアサートされると、このピンがローになります (OSEL[1:0] に応じて)。

ビット 19 **COSEL** : 較正出力選択

- COE = 1 のとき、このビットによって RTC\_CALIB に出力される信号を選択します。
- 0 : 較正出力は 512 Hz です。
- 1 : 較正出力は 1 Hz です。
- これらの周波数は、RTCCLK が 32.768 kHz で、プリスケラがデフォルト値 (PREDIV\_A = 127 および PREDIV\_S = 255) の場合に有効です。 [セクション 29.3.15 : 較正クロック出力](#) を参照してください。

ビット 18 **BKP** : バックアップ

- このビットは、サマータイムの変更を実施したか否かを記憶しておくため、ユーザが書き込むことができます。

ビット 17 **SUB1H** : 1 時間差し引き (冬時間変更)

- このビットを初期化モード以外のときにセットすると、現在時刻が 0 でない場合にカレンダー時刻から 1 時間を差し引きます。このビットは常に 0 として読み出されます。
- 現在時間が 0 のときにこのビットをセットしても、影響はありません。
- 0 : 影響なし。
- 1 : 現在時刻から 1 時間差し引きます。これは、冬時間変更で使用できます。

ビット 16 **ADD1H** : 1 時間加算 (サマータイム変更)

このビットを初期化モード以外のときにセットすると、カレンダー時刻に 1 時間加算します。このビットは常に 0 として読み出されます。

0 : 影響なし。

1 : 現在時刻に 1 時間加算します。これは、サマータイム変更に使用できます。

ビット 15 **TSIE** : タイムスタンプ割り込みイネーブル

0 : タイムスタンプ割り込みは無効です。

1 : タイムスタンプ割り込みは有効です。

ビット 14 **WUTIE** : ウェイクアップタイマ割り込みイネーブル

0 : ウェイクアップタイマ割り込みは無効です。

1 : ウェイクアップタイマ割り込みは有効です。

ビット 13 **ALRBIE** : アラーム B 割り込みイネーブル

0 : アラーム B 割り込みは無効です。

1 : アラーム B 割り込みは有効です。

ビット 12 **ALRAIE** : アラーム A 割り込みイネーブル

0 : アラーム A 割り込みは無効です。

1 : アラーム A 割り込みは有効です。

ビット 11 **TSE** : タイムスタンプイネーブル

0 : タイムスタンプは無効です。

1 : タイムスタンプは有効です。

ビット 10 **WUTE** : ウェイクアップタイマイネーブル

0 : ウェイクアップタイマは無効です。

1 : ウェイクアップタイマは有効です。

ビット 9 **ALRBE** : アラーム B イネーブル

0 : アラーム B は無効です。

1 : アラーム B は有効です。

ビット 8 **ALRAE** : アラーム A イネーブル

0 : アラーム A は無効です。

1 : アラーム A は有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FMT** : 時間形式

0 : 24 時間 / 日形式

1 : AM / PM 時間形式

ビット 5 **BYPHAD** : シャドウレジスタをバイパスします。

0 : カレンダー値は (RTC\_SSR、RTC\_TR、RTC\_DR から読み出す場合)、シャドウレジスタから取得され、これらは 2 RTCCLK サイクルごとに 1 回更新されます。

1 : カレンダー値は (RTC\_SSR、RTC\_TR、RTC\_DR から読み出す場合)、カレンダーカウンタから直接取得されます。

**注 :** **APB クロックの周波数が RTCCLK の 7 倍未満である場合、BYPHAD は“1”にセットする必要があります。**

ビット 4 **REFCKON** : RTC\_REFIN リファレンスクロック検出イネーブル (50 または 60 Hz)

0 : RTC\_REFIN 検出は無効です。

1 : RTC\_REFIN 検出は有効です。

**注 :** *PREDIV\_S は 0x00FF である必要があります。*

ビット 3 **TSEDGE** : タイムスタンプイベントアクティブエッジ

0 : RTC\_TS 入力の立ち上がりエッジによってタイムスタンプイベントを生成します。

1 : RTC\_TS 入力の立ち下がりエッジによってタイムスタンプイベントを生成します。

不要な TSF 設定を回避するため、TSEDGE が変化した場合には TSE をリセットする必要があります。

ビット 2:0 **WUCKSEL[2:0]** : ウェイクアップクロック選択

000 : RTC/16 クロックが選択されます。

001 : RTC/8 クロックが選択されます。

010 : RTC/4 クロックが選択されます。

011 : RTC/2 クロックが選択されます。

10x : ck\_spre (通常は 1 Hz) クロックが選択されます。

11x : ck\_spre (通常は 1 Hz) クロックが選択され、 $2^{16}$  が WUT カウンタ値に加算されます (下記注を参照)。

**注 :** *初期化モード (RTC\_ISR/INITF = 1) の場合のみ、このレジスタのビット 7、6、4 が書き込めます。  
WUT = ウェイクアップユニットカウンタ値  $WUT = (0x0000 \sim 0xFFFF) + 0x10000$  (WUCKSEL[2:1] = 11 の場合追加されます。)*

*このレジスタのビット 2~0 は、RTC\_CR WUTE ビット = 0 かつ RTC\_ISR WUTWF ビット = 1 の場合にのみ書き込めます。*

*カレンダーの時間項目のインクリメント中は時間を変更しないことが推奨されます。カレンダーの時間項目のインクリメントがマスクされる可能性があるためです。*

*ADD1H および SUB1H の変更は、次の秒から有効になります。*

*このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。*

## 29.6.4 RTC 初期化とステータスレジスタ (RTC\_ISR)

このレジスタは、書き込み保護されています (RTC\_ISR[13:8] ビットを除く)。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x0C

Backup ドメインリセット値 : 0x0000 0007

システムリセット : 0 にクリアされる INIT、INITF、RSF ビット以外は影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF
														rc_w0	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRB WF	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

ビット 31:18 予約済みであり、リセット値のままにしておかなければなりません。

### ビット 17 **ITSF** : 内部タイムスタンプフラグ

このフラグは、内部イベントでタイムスタンプが発生したときに、ハードウェアによってセットされます。

このフラグは、0 を書き込むことでソフトウェアによってクリアされ、両方のビットに 0 を書き込むことで TSF ビットとともにクリアする必要があります。

### ビット 16 **RECALPF** : 再較正保留フラグ

ソフトウェアによって RTC\_CALR レジスタに書き込みが行われると、RECALPF ステータスフラグが自動的に“1”にセットされ、RTC\_CALR レジスタがブロックされたことを示します。新たな較正設定が認識されると、このビットは“0”に戻ります。[動作中の再較正](#)を参照してください。

### ビット 15 **TAMP3F** : RTC\_TAMP3 検出フラグ

このフラグは、RTC\_TAMP3 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ソフトウェアで 0 を書き込むことによってクリアされます。

### ビット 14 **TAMP2F** : RTC\_TAMP2 検出フラグ

このフラグは、RTC\_TAMP2 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ソフトウェアで 0 を書き込むことによってクリアされます。

### ビット 13 **TAMP1F** : RTC\_TAMP1 検出フラグ

このフラグは、RTC\_TAMP1 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ソフトウェアで 0 を書き込むことによってクリアされます。

### ビット 12 **TSOVF** : タイムスタンプオーバーフローフラグ

このフラグは、TSF が既にセットされている間にタイムスタンプイベントが発生したときに、ハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。そうしないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

### ビット 11 **TSF** : タイムスタンプフラグ

このフラグは、タイムスタンプイベントが発生したときに、ハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。ITSF フラグがセットされている場合は、TSF は両方のビットに 0 を書き込むことで、ITSF とともにクリアする必要があります。

**ビット 10 WUTF** : ウェイクアップタイマフラグ

このフラグは、ウェイクアップ自動再ロードカウンタが 0 に到達したときに、ハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

**ビット 9 ALRBF** : アラーム B フラグ

このフラグは、時刻/日付レジスタ (RTC\_TR および RTC\_DR) がアラーム B レジスタ (RTC\_ALRMBR) と一致したときにハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

**ビット 8 ALRAF** : アラーム A フラグ

このフラグは、時刻/日付レジスタ (RTC\_TR および RTC\_DR) がアラーム A レジスタ (RTC\_ALRMAR) と一致したときにハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

**ビット 7 INIT** : 初期化モード

0 : フリーランニングモード

1 : 時刻と日付レジスタ (RTC\_TR と RTC\_DR)、およびプリスケアラレジスタ (RTC\_PRER) のプログラムに使用する初期化モードです。INIT がリセットされると、カウンタは停止し、新しい値からカウントし始めます。

**ビット 6 INITF** : 初期化フラグ

このビットが 1 にセットされると、RTC は初期化状態となり、時刻、日付およびプリスケアラレジスタが更新できます。

0 : カレンダーレジスタを更新できません。

1 : カレンダーレジスタを更新できます。

**ビット 5 RSF** : レジスタ同期フラグ

このビットは、カレンダーレジスタがシャドウレジスタ (RTC\_SSRx、RTC\_TRx および RTC\_DRx) にコピーされるたびにハードウェアによってセットされます。このビットは、シフト操作が保留中 (SHPF = 1) に初期化モードで、またはバイパスシャドウレジスタモード (BYPSHAD = 1) で、ハードウェアによってクリアされます。このビットは、ソフトウェアでクリアすることもできます。

初期化モードでソフトウェアまたはハードウェアによってクリアされます。

0 : カレンダーシャドウレジスタはまだ同期していません。

1 : カレンダーシャドウレジスタは同期しています。

**ビット 4 INITS** : 初期化ステータスフラグ

このビットは、カレンダーの年の項目が 0 ではないとき (Backup ドメインリセット状態) にハードウェアによってセットされます。

0 : カレンダーは初期化されていません。

1 : カレンダーは初期化されています。

**ビット 3 SHPF** : シフト操作保留

0 : 保留中のシフト操作はありません。

1 : 保留中のシフト操作があります。

このフラグは、RTC\_SHIFTR への書き込みによってシフト操作が開始された直後に、ハードウェアによってセットされます。該当するシフト操作が実行されると、ハードウェアによってクリアされます。SHPF ビットに書き込んでも影響はありません。

**ビット 2 WUTWF** : ウェイクアップタイマ書き込みフラグ

このビットは、ハードウェアによって WUTE ビットが RTC\_CR で 0 にセットされた後で 2 RTCCLK サイクルまでセットされ、WUTE ビットが 1 にセットされた後で 2 RTCCLK サイクルまでクリアされます。ウェイクアップタイマ値は、WUTE ビットがクリアされ、WUTWF がセットされたときに変更されます。

0 : ウェイクアップタイマ設定は更新できません。

1 : ウェイクアップタイマ設定は更新できます。

**ビット 1 ALRBWF** : アラーム B 書き込みフラグ

このビットは、RTC\_CR で ALRBIE ビットが 0 にセットされた後、アラーム B 値が変更可能な時にハードウェアによってセットされます。

初期化モードでハードウェアによってクリアされます。

0 : アラーム B は更新できません。

1 : アラーム B は更新できます。

**ビット 0 ALRAWF** : アラーム A 書き込みフラグ

このビットは、RTC\_CR で ALRAE ビットが 0 にセットされた後、アラーム A 値が変更可能な時にハードウェアによってセットされます。

初期化モードでハードウェアによってクリアされます。

0 : アラーム A は更新できません。

1 : アラーム A は更新できます。

**注 :** **ALRAF、ALRBF、WUTF、および TSF ビットは、0 にプログラムされてから 2 APB クロック サイクル後にクリアされます。**

## 29.6.5 RTC プリスケーラレジスタ (RTC\_PRER)

このレジスタは、必ず初期化モードで書き込む必要があります。初期化は、2 回の書き込みアクセスに分けて行う必要があります。[カレンダーの初期化と設定 \(881 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x10

Backup ドメインリセット値 : 0x007F 00FF

システムリセット : 影響なし

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

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22:16 **PREDIV\_A[6:0]** : 非同期プリスケーラ分周比

非同期分周比です。

$ck\_apre \text{ 周波数} = RTCCLK \text{ 周波数} / (PREDIV\_A + 1)$

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **PREDIV\_S[14:0]** : 同期プリスケーラ分周比

同期分周比です。

$ck\_spre \text{ 周波数} = ck\_apre \text{ 周波数} / (PREDIV\_S + 1)$

## 29.6.6 RTC ウェイクアップタイマレジスタ (RTC\_WUTR)

このレジスタは、RTC\_ISR の WUTWF が 1 にセットされているときのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x14

Backup ドメインリセット値 : 0x0000 FFFF

システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **WUT[15:0]** : ウェイクアップ自動再ロード値ビット

ウェイクアップタイマが有効 (WUTE が 1 にセット) なとき、ck\_wut の (WUT[15:0] + 1) サイクル毎に WUTF フラグがセットされます。ck\_wut の周期は、RTC\_CR レジスタの WUCKSEL[2:0] ビットで選択します。

WUCKSEL[2] = 1 のとき、ウェイクアップタイマは 17 ビットとなり、WUCKSEL[1] が事実上タイマに再ロードされる最上位ビットである WUT[16] となります。

WUTF の最初のアサートは、WUTE がセットされてから ck\_wut の (WUT+1) サイクル後に発生します。WUCKSEL[2:0] = 011 (RTCCLK/2) のときに WUT[15:0] を 0x0000 にセットすることはできません。

## 29.6.7 RTC アラーム A レジスタ (RTC\_ALRMAR)

このレジスタは、RTC\_ISR の ALRAWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x1C

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]		SU[3:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **MSK4** : アラーム A 日付マスク

- 0 : 日付/曜日が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では日付/曜日を無視します。

ビット 30 **WDSEL** : 曜日選択

- 0 : DU[3:0] は日付の一の位を表します。
- 1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位

ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

ビット 23 **MSK3** : アラーム A 時マスク

- 0 : 時が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では時を無視します。

ビット 22 **PM** : AM/PM 表記

- 0 : AM または 24 時間形式
- 1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 **MSK2** : アラーム A 分マスク

- 0 : 分が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では分を無視します。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 **MSK1** : アラーム A 秒マスク

- 0 : 秒が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では秒を無視します。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 29.6.8 RTC アラーム B レジスタ (RTC\_ALRMBR)

このレジスタは、RTC\_ISR の ALRBWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x20

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]		SU[3:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **MSK4** : アラーム B 日付マスク

- 0 : 日付/曜日が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では日付/曜日を無視します。

ビット 30 **WDSEL** : 曜日選択

- 0 : DU[3:0] は日付の一の位を表します。
- 1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位

ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

ビット 23 **MSK3** : アラーム B 時マスク

- 0 : 時が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では時を無視します。

ビット 22 **PM** : AM/PM 表記

- 0 : AM または 24 時間形式
- 1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 **MSK2** : アラーム B 分マスク

- 0 : 分が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では分を無視します。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 **MSK1** : アラーム 秒マスク

- 0 : 秒が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では秒を無視します。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 29.6.9 RTC 書き込み保護レジスタ (RTC\_WPR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **KEY** : 書き込み保護キー

このバイトはソフトウェアで書き込まれます。

このバイトを読み出すと常に 0x00 が返されます。

RTC レジスタの書き込み保護解除方法については、[RTC レジスタ書き込み保護](#) を参照してください。

## 29.6.10 RTC サブセカンドレジスタ (RTC\_SSR)

アドレスオフセット : 0x28

Backup ドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS** : サブセカンド値

SS[15:0] は、同期プリスケアラのカウンタ内の値です。秒の小数部は、下の式によって与えられます。

$$\text{秒の小数部} = (\text{PREDIV}_S - \text{SS}) / (\text{PREDIV}_S + 1)$$

**注 :** **SS** は、シフト操作後に限り、**PREDIV\_S** より大きな値となる場合があります。この場合、正確な時刻/日付は、**RTC\_TR/RTC\_DR** で示される値よりも 1 秒少ない値となります。

## 29.6.11 RTC シフト制御レジスタ (RTC\_SHIFTR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x2C

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **ADD1S** : 1 秒加算

0 : 影響なし。

1 : 時計/カレンダーに 1 秒加算します。

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC\_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

この機能は、SUBFS (下記説明を参照) と共に使用されることを想定しており、不可分操作で、効果的に時計に秒の小数部を加算することを目的としています。

ビット 30:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:0 **SUBFS** : 秒の小数部差し引き

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC\_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

SUBFS に書き込まれた値は、同期プリスケアラのカウンタに加算されます。このカウンタはカウントダウンして行くので、この操作によって、次の式で求める値が効果的にクロックから差し引き (遅延) されます。

$$\text{遅れ (秒)} = \text{SUBFS} / (\text{PREDIV}_S + 1)$$

ADD1S 機能が SUBFS と共に用いられた場合、秒の小数部を効果的にクロックに加算する (クロックを進める) ことができ、実際のクロックの進みは次の式のとおりとなります。

$$\text{進み (秒)} = (1 - (\text{SUBFS} / (\text{PREDIV}_S + 1)))$$

**注 :** **SUBFS に書き込むことにより RSF はクリアされます。その後、ソフトウェアが RSF = 1 まで待つことにより、シャドウレジスタがシフトされた時刻で更新されていることが確実にになります。**

## 29.6.12 RTC タイムスタンプ時刻レジスタ (RTC\_TSTR)

このレジスタの内容は、RTC\_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x30

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式  
1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 29.6.13 RTC タイムスタンプ日付レジスタ (RTC\_TSDR)

このレジスタの内容は、RTC\_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x34

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:13 **WDU[1:0]** : 曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

## 29.6.14 RTC タイムスタンプサブセカンドレジスタ (RTC\_TSSSR)

このレジスタの内容は、RTC\_ISR/TSF がセットされている場合にのみ有効です。また、RTC\_ISR/TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x38

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS** : サブセカンド値

SS[15:0] は、タイムスタンプイベントが発生したときの同期プリスケアラのカウンタの値です。

## 29.6.15 RTC 較正レジスタ (RTC\_CALR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(881 ページ\)](#) を参照してください。

アドレスオフセット : 0x3C

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15 **CALP** : RTC 周波数を 488.5 ppm 増加

0 : RTCCLK パルスは加えられません。

1 : RTCCLK の  $2^{11}$  パルスごとに 1 パルス効果的に挿入されます (周波数が 488.5 ppm 増加)。

この機能は、CALM と共に使用されることを想定しており、カレンダーの周波数を高分解能で下げることができます。入力周波数が 32768 Hz の場合、32 秒枠の間に追加される RTCCLK パルスの数は次のように算出されます :  $(512 * CALP) - CALM$

[セクション 29.3.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 14 **CALW8** : 8 秒較正サイクル周期の使用

CALW8 が“1”にセットされると、8 秒較正サイクル周期が選択されます。

**注 :** **CALW8 = 1 の場合、CALM[1:0] は“00”に固定されます。** [セクション 29.3.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 13 **CALW16** : 16 秒較正サイクル周期の使用

CALW16 が“1”にセットされると、16 秒較正サイクル周期が選択されます。CALW8 = 1 の場合、このビットを“1”にセットすることはできません。

**注 :** **CALW16 = 1 の場合、CALM[0] は“0”に固定されます。** [セクション 29.3.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 12:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8:0 **CALM[8:0]** : 較正マイナス

RTCCLK  $2^{20}$  パルス (入力周波数が 32768 Hz の場合 32 秒) 内の CALM をマスクすることによって、カレンダーの周波数が下げられます。この方法により、カレンダーの周波数を 0.9537 ppm の分解能で下げることができます。

カレンダーの周波数を上げるには、この機能を CALP と共に使用する必要があります。[セクション 29.3.12 : RTC の高精度デジタル較正 \(885 ページ\)](#) を参照してください。

## 29.6.16 RTC タンパ設定レジスタ (RTC\_TAMPCR)

アドレスオフセット : 0x40

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3 MF	TAMP3 NO ERASE	TAMP3 IE	TAMP2 MF	TAMP2 NO ERASE	TAMP2 IE	TAMP1 MF	TAMP1 NO ERASE	TAMP1 IE
							rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP PUDIS	TAMP PRCH [1:0]		TAMP FLT[1:0]		TAMP FREQ[2:0]			TAMP TS	TAMP3 TRG	TAMP3 E	TAMP2 TRG	TAMP2 E	TAMP1 E	TAMP1 TRG	TAMP1 E
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TAMP3MF** : タンパ 3 マスクフラグ

0 : タンパ 3 イベントによってトリガイベントが生成され、TAMP3F はソフトウェアでクリアして、次のタンパイベントを検出できるようにする必要があります。

1 : タンパ 3 イベントによってトリガイベントが生成されます。TAMP3F はマスクされ、ハードウェアによって内部でクリアされます。バックアップレジスタは消去されません。

**注 :** **TAMP3MF がセットされている場合、タンパ 3 割り込みを有効にすることはできません。**

ビット 23 **TAMP3NOERASE** : タンパ 3 消去なし

0 : タンパ 3 イベントでバックアップレジスタは消去されます。

1 : タンパ 3 イベントでバックアップレジスタは消去されません。

ビット 22 **TAMP3IE** : タンパ 3 割り込みイネーブル

0 : TAMP3IE = 0 の場合、タンパ 3 割り込みは無効です。

1 : タンパ 3 割り込みは有効です。

ビット 21 **TAMP2MF** : タンパ 2 マスクフラグ

0 : タンパ 2 イベントによってトリガイベントが生成され、TAMP2F はソフトウェアでクリアして、次のタンパイベントを検出できるようにする必要があります。

1 : タンパ 2 イベントによってトリガイベントが生成されます。TAMP2F はマスクされ、ハードウェアによって内部でクリアされます。バックアップレジスタは消去されません。

**注 :** **TAMP2MF がセットされている場合、タンパ 2 割り込みを有効にすることはできません。**

ビット 20 **TAMP2NOERASE** : タンパ 2 消去なし

0 : タンパ 2 イベントでバックアップレジスタは消去されます。

1 : タンパ 2 イベントでバックアップレジスタは消去されません。

ビット 19 **TAMP2IE** : タンパ 2 割り込みイネーブル

0 : TAMP2IE = 0 の場合、タンパ 2 割り込みは無効です。

1 : タンパ 2 割り込みは有効です。

ビット 18 **TAMP1MF** : タンパ 1 マスクフラグ

0 : タンパ 1 イベントによってトリガイベントが生成され、TAMP1F はソフトウェアでクリアして、次のタンパイベントを検出できるようにする必要があります。

1 : タンパ 1 イベントによってトリガイベントが生成されます。TAMP1F はマスクされ、ハードウェアによって内部でクリアされます。バックアップレジスタは消去されません。

**注 :** **TAMP1MF がセットされている場合、タンパ 1 割り込みを有効にすることはできません。**

ビット 17 **TAMP1NOERASE** : タンパ 1 消去なし

- 0 : タンパ 1 イベントでバックアップレジスタは消去されます。
- 1 : タンパ 1 イベントでバックアップレジスタは消去されません。

ビット 16 **TAMP1IE** : タンパ 1 割り込みイネーブル

- 0 : TAMPIE = 0 の場合、タンパ 1 割り込みは無効です。
- 1 : タンパ 1 割り込みは有効です。

ビット 15 **TAMPPUDIS** : RTC\_TAMPx ブルアップディセーブル

このビットにより、毎回のサンプリング前に各 RTC\_TAMPx ピンをプリチャージするかどうか決定します。

- 0 : サンプリング前に RTC\_TAMPx ピンをプリチャージします (内部ブルアップを有効化)。
- 1 : RTC\_TAMPx ピンのプリチャージを無効化します。

ビット 14:13 **TAMPPRCH[1:0]** : RTC\_TAMPx プリチャージ時間

これらのビットにより、各サンプリングの前にブルアップを有効化している時間を決定します。TAMPPRCH は、各 RTC\_TAMPx 入力に対して有効です。

- 0x0 : 1 RTCCLK サイクル
- 0x1 : 2 RTCCLK サイクル
- 0x2 : 4 RTCCLK サイクル
- 0x3 : 8 RTCCLK サイクル

ビット 12:11 **TAMPFLT[1:0]** : RTC\_TAMPx フィルタカウント

これらのビットにより、タンパイベントをアクティブにするのに必要な指定のレベル (TAMP\*TRG) での連続サンプリングの数を決定します。TAMPFLT は、各 RTC\_TAMPx 入力に対して有効です。

- 0x0 : RTC\_TAMPx 入力 that アクティブレベル (RTC\_TAMPx 入力における内部ブルアップ無し) に遷移するときのエッジで、タンパイベントがアクティブになります。
- 0x1 : アクティブレベルでの連続した 2 回のサンプリングの後、タンパイベントがアクティブになります。
- 0x2 : アクティブレベルでの連続した 4 回のサンプリングの後、タンパイベントがアクティブになります。
- 0x3 : アクティブレベルでの連続した 8 回のサンプリングの後、タンパイベントがアクティブになります。

ビット 10:8 **TAMPFREQ[2:0]** : タンパサンプリング周波数

各 RTC\_TAMPx 入力 that サンプリングされる周波数を決定します。

- 0x0 : RTCCLK / 32768 (RTCCLK = 32768 Hz の場合 1 Hz)
- 0x1 : RTCCLK / 16384 (RTCCLK = 32768 Hz の場合 2 Hz)
- 0x2 : RTCCLK / 8192 (RTCCLK = 32768 Hz の場合 4 Hz)
- 0x3 : RTCCLK / 4096 (RTCCLK = 32768 Hz の場合 8 Hz)
- 0x4 : RTCCLK / 2048 (RTCCLK = 32768 Hz の場合 16 Hz)
- 0x5 : RTCCLK / 1024 (RTCCLK = 32768 Hz の場合 32 Hz)
- 0x6 : RTCCLK / 512 (RTCCLK = 32768 Hz の場合 64 Hz)
- 0x7 : RTCCLK / 256 (RTCCLK = 32768 Hz の場合 128 Hz)

ビット 7 **TAMPTS** : タンパ検出イベント時のタイムスタンプの有効化

- 0 : タンパ検出イベントがあっても、タイムスタンプは保存されません。
- 1 : タンパ検出イベント時、タイムスタンプが保存されます。

RTC\_CR レジスタで TSE = 0 であっても TAMPTS は有効です。

ビット 6 **TAMP3TRG** : RTC\_TAMP3 入力のアクティブレベル

TAMPFLT ≠ 00 の場合 :

- 0 : RTC\_TAMP3 入力 that ローのままのとき、タンパ検出イベント that トリガされます。
- 1 : RTC\_TAMP3 入力 that ハイのままのとき、タンパ検出イベント that トリガされます。

TAMPFLT = 00 の場合

- 0 : RTC\_TAMP3 入力の立ち上がりエッジ with タンパ検出イベント that トリガされます。
- 1 : RTC\_TAMP3 入力の立ち下がりエッジ with タンパ検出イベント that トリガされます。

**注：** VBAT への切り替えが使用されている場合、タンパ3 の立ち下がリエッジ検出は使用できません。そうしないと、検出は常に VBAT モードの入力時に発生します。

ビット 5 **TAMP3E** : RTC\_TAMP3 検出イネーブル

- 0 : RTC\_TAMP3 入力検出は無効です。
- 1 : RTC\_TAMP3 入力検出は有効です。

ビット 4 **TAMP2TRG** : RTC\_TAMP2 入力のアクティブレベル

TAMPFLT != 00 の場合

- 0 : RTC\_TAMP2 入力ローのままのとき、タンパ検出イベントがトリガされます。
- 1 : RTC\_TAMP2 入力ハイのままのとき、タンパ検出イベントがトリガされます。

TAMPFLT = 00 の場合

- 0 : RTC\_TAMP2 入力の立ち上がりエッジでタンパ検出イベントがトリガされます。
- 1 : RTC\_TAMP2 入力の立ち下がりエッジでタンパ検出イベントがトリガされます。

ビット 3 **TAMP2E** : RTC\_TAMP2 入力検出イネーブル

- 0 : RTC\_TAMP2 検出は無効です。
- 1 : RTC\_TAMP2 検出は有効です。

ビット 2 **TAMPIE** : タンパ割り込みイネーブル

- 0 : タンパ割り込みは無効です。
- 1 : タンパ割り込みは有効です。

**注：** このビットは、TAMPxIE レベルにかかわらず、すべてのタンパピンイベントに対する割り込みを有効にします。このビットがクリアされた場合、TAMPxIE をセットすることによって、各タンパイベントの割り込みを個別に有効にすることができます。

ビット 1 **TAMP1TRG** : RTC\_TAMP1 入力のアクティブレベル

TAMPFLT != 00 の場合

- 0 : RTC\_TAMP1 入力ローのままのとき、タンパ検出イベントがトリガされます。
- 1 : RTC\_TAMP1 入力ハイのままのとき、タンパ検出イベントがトリガされます。

TAMPFLT = 00 の場合

- 0 : RTC\_TAMP1 入力の立ち上がりエッジでタンパ検出イベントがトリガされます。
- 1 : RTC\_TAMP1 入力の立ち下がりエッジでタンパ検出イベントがトリガされます。

ビット 0 **TAMP1E** : RTC\_TAMP1 入力検出イネーブル

- 0 : RTC\_TAMP1 検出は無効です。
- 1 : RTC\_TAMP1 検出は有効です。

**注意：** TAMPFLT = 0 の場合、TAMP1E ビットは、TAMP1TRG が変更されたらリセットし、TAMP1F の不要なセットを回避する必要があります。

## 29.6.17 RTC アラーム A サブセカンドレジスタ (RTC\_ALRMSSR)

このレジスタは、RTC\_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込めます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護を参照してください。[RTC レジスタ書き込み保護 \(881 ページ\)](#)

アドレスオフセット : 0x44

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 MASKSS[3:0] : このビットから始まる最上位ビットのマスク

0 : アラーム A に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

1 : アラーム A の比較では SS[14:1] を無視します。SS[0] のみ比較されます。

2 : アラーム A の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。

3 : アラーム A の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。

...

12 : アラーム A の比較では SS[14:12] を無視します。SS[11:0] が比較されます。

13 : アラーム A の比較では SS[14:13] を無視します。SS[12:0] が比較されます。

14 : アラーム A の比較では SS[14] を無視します。SS[13:0] が比較されます。

15 : アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。

同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 SS[14:0] : サブセカンド値

この値が同期プリスケラのカウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

## 29.6.18 RTC アラーム B サブセカンドレジスタ (RTC\_ALRMBSSR)

このレジスタは、RTC\_CR レジスタの ALRBE がリセットされた場合、または初期化モードの場合にのみ書き込みめます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[セクション : RTC レジスタ書き込み保護](#)を参照してください。

アドレスオフセット : 0x48

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **MASKSS[3:0]** : このビットから始まる最上位ビットのマスク

0x0 : アラーム B に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

0x1 : アラーム B の比較では SS[14:1] を無視します。SS[0] のみ比較されます。

0x2 : アラーム B の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。

0x3 : アラーム B の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。

...

0xC : アラーム B の比較では SS[14:12] を無視します。SS[11:0] が比較されます。

0xD : アラーム B の比較では SS[14:13] を無視します。SS[12:0] が比較されます。

0xE : アラーム B の比較では SS[14] を無視します。SS[13:0] が比較されます。

0xF : アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。

同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **SS[14:0]** : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム B をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

## 29.6.19 RTC オプションレジスタ (RTC\_OR)

アドレスオフセット : 0x4C

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **RTC\_ALARM\_TYPE** : PC13 への RTC\_ALARM の出力タイプ

0 : RTC\_ALARM は、PC13 に配置される場合は、オープンドレイン出力です。

1 : RTC\_ALARM は、PC13 に配置される場合は、プッシュプル出力です。

ビット 2:1 **TSINSEL[1:0]** : TIMESTAMP マッピング

00 : TIMESTAMP は PC13 に配置されます。

01 : TIMESTAMP は PI8 に配置されます。

10 : TIMESTAMP は PC1 に配置されます。

11 : TIMESTAMP は PC1 に配置されます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

## 29.6.20 RTC バックアップレジスタ (RTC\_BKPxR)

アドレスオフセット : 0x50 から 0xCC

Backup ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

### ビット 31:0 BKP[31:0]

アプリケーションはこれらのレジスタに対してデータの読み書きをすることができます。

これらのレジスタは、V<sub>DD</sub> がオフになった場合、V<sub>BAT</sub> によって電源が供給されるため、システムリセットによりリセットされず、デバイスが低電力モードで動作する場合、レジスタの内容は有効なまま保持されます。

このレジスタは、TAMPx<sub>F</sub> = 1 である限り、あるいはフラッシュ読み出し保護が無効なときは、タンパ検出イベント時にリセットされます。

## 29.6.21 RTC レジスタマップ

表 152. RTC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RTC_TR	Res	Res	Res	Res	Res	Res	Res	Res	Res	PM	HT [1:0]	HU[3:0]			Res	MNT[2:0]			MNU[3:0]			Res	ST[2:0]		SU[3:0]								
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	RTC_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	YT[3:0]			YU[3:0]			WDU[2:0]		MT	MU[3:0]			Res	Res	DT [1:0]	DU[3:0]								
	リセット値										0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1		0	0	0	0	1	
0x08	RTC_CR	Res	Res	Res	Res	Res	Res	Res	Res	ITSE	COE	OSEL [1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTIE	ALRBE	ALRAE	Res	FMT	BYPSHAD	REFCKON	TSEDGE	WUICKSEL [2:0]		
	リセット値									0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	RTC_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																	
0x10	RTC_PRER	Res	Res	Res	Res	Res	Res	Res	Res	Res	PREDIV_A[6:0]						PREDIV_S[14:0]																	
	リセット値										1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	RTC_WUTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUT[15:0]																
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	RTC_ALRMAR	MSK4	WDSSEL	DT [1:0]	DU[3:0]			MSK3	PM	HT [1:0]	HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK1	ST[2:0]		SU[3:0]											
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	RTC_ALRMBR	MSK4	WDSSEL	DT [1:0]	DU[3:0]			MSK3	PM	HT [1:0]	HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK2	ST[2:0]		SU[3:0]											
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	RTC_WPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	
0x28	RTC_SSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	
0x2C	RTC_SHIFTR	ADD1S	SUBFS[14:0]																															
	リセット値	0	0																															
0x30	RTC_TSTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	PM	HT [1:0]	HU[3:0]			Res	MNT[2:0]		MNU[3:0]			Res	ST[2:0]		SU[3:0]									
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x34	RTC_TSDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	
0x38	RTC_TSSSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	
0x3C	RTC_CALR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	



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

表 152. RTC レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	RTC_TAMPCR	Res	Res	Res	Res	Res	Res	Res	TAMP3MF	TAMP3NOERASE	TAMP3IE	TAMP2MF	TAMP2NOERASE	TAMP2IE	TAMP1MF	TAMP1NOERASE	TAMP1IE	TAMPPUDIS	TAMPPRCH[1:0]	TAMPFLT[1:0]	TAMPFREQ[2:0]	TAMPTS	TAMP3-TRG	TAMP3E	TAMP2-TRG	TAMP2E	TAMP1-TRG	TAMP1E	0	0	0	0	
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x44	RTC_ALRMASRR	Res	Res	Res	Res	MASKSS [3:0]			Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SS[14:0]														
	リセット値					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x48	RTC_ALRMBSSR	Res	Res	Res	Res	MASKSS [3:0]			Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SS[14:0]														
	リセット値					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x4C	RTC_OR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																												0	0	0	0	
0x50 to 0xCC	RTC_BKP0R	BKP[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50 to 0xCC	RTC_BKP31R へ	BKP[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 30 I2C (Inter-integrated circuit) インタフェース

### 30.1 概要

I<sup>2</sup>C (Inter-integrated circuit) バスインタフェースは、マイクロコントローラとシリアル I<sup>2</sup>C バス間の通信を処理します。マルチマスタ機能を備え、すべての I<sup>2</sup>C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準モード (Sm)、高速モード (Fm)、および高速モードプラス (Fm+) をサポートします。

また、SMBus (System Management Bus) および PMBus (Power Management Bus) と互換性があります。

DMA を使用して、CPU の負荷を軽減できます。

### 30.2 I2C の主な機能

- I<sup>2</sup>C バス仕様 rev03 との互換性：
  - スレーブおよびマスタモード
  - マルチマスタ機能
  - 標準モード (最大 100 kHz)
  - 高速モード (最大 400 kHz)
  - 高速モードプラス (最大 1 MHz)
  - 7ビットおよび 10 ビットアドレッシングモード
  - 複数の 7 ビットスレーブアドレス (2 つのアドレス、1 つは設定可能なマスク付き)
  - すべての 7 ビットアドレス確認応答モード
  - 同報 (General call) コール
  - プログラム可能なセットアップおよびホールド時間
  - 使いやすいイベント管理
  - クロックストレッチオプション
  - ソフトウェアリセット

- DMA 機能付きの 1 バイトバッファ
- プログラム可能なアナログおよびデジタルノイズフィルタ

製品の実装によっては、次の追加機能も使用できます ([セクション 30.3 : I2C の実装](#) を参照) :

- SMBus 仕様 rev 2.0 との互換性：
  - ハードウェア PEC (Packet Error Checking) の生成と ACK 制御による確認
  - コマンドおよびデータ確認応答制御
  - アドレス解決プロトコル (ARP) サポート
  - ホストおよびデバイスのサポート
  - SMBus アラート
  - タイムアウトおよびアイドル条件の検出
- PMBus rev 1.1 標準との互換性
- 独立したクロック : 独立したクロックソースの選択により、I2C の通信速度は PCLK の再プログラミングから独立

## 30.3 I2C の実装

このマニュアルでは、I2C1、I2C2、I2C3、および I2C4 に実装されているすべての機能について説明しています。I2C1、I2C2、I2C3、および I2C4 は同一で、次の表に示すように、すべての機能を実装しています。

表 153. STM32F75xxx および STM32F74xxxI2C の実装

I2C の機能 <sup>(1)</sup>	I2C1	I2C2	I2C3	I2C4
7 ビットアドレスモード	X	X	X	X
10 ビットアドレスモード	X	X	X	X
標準モード (最大 100 kbit/s)	X	X	X	X
高速モード (最大 400 kbit/s)	X	X	X	X
高速モードプラス (最大 1 Mbit/s)	X	X	X	X
独立クロック	X	X	X	X
SMBus	X	X	X	X

1. X: サポートされています。

## 30.4 I2C の機能詳細

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割り込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン（SDA）とクロックピン（SCL）によって I2C バスに接続されます。標準（最大 100 kHz）、高速モード（最大 400 kHz）、または高速モードプラス（最大 1 MHz）の I<sup>2</sup>C バスで接続できます。

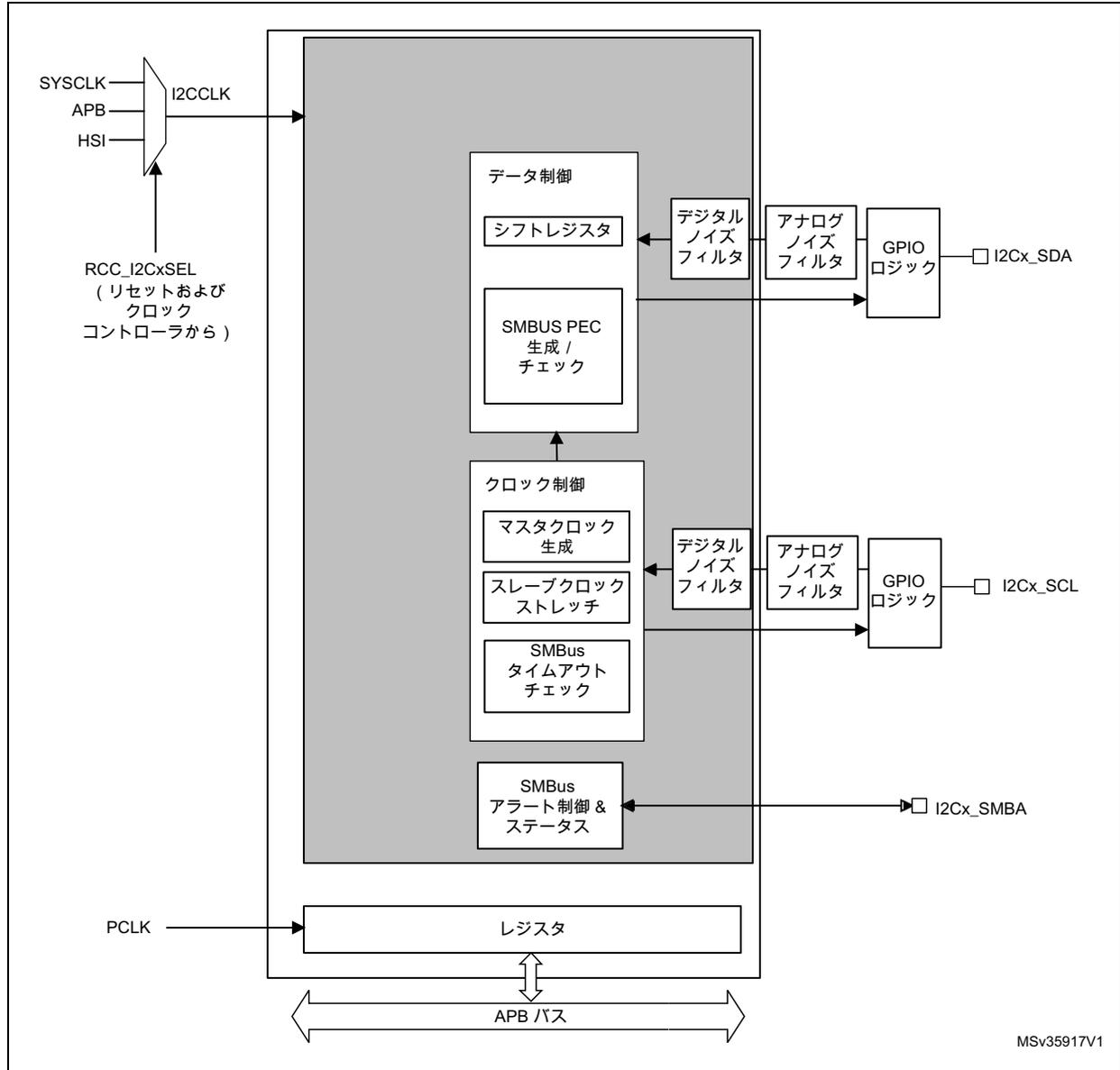
このインタフェースは、データピン（SDA）とクロックピン（SCL）によって SMBus に接続することもできます。

SMBus 機能がサポートされる場合、追加の SMBus アラートピン（SMBA）オプションも使用できます。

## 30.4.1 I2C ブロック図

I2C インタフェースのブロック図を [図 283](#) に示します。

図 283. I2C ブロック図



I2C は、独立したクロックソースによってクロック供給されるため、I2C は PCLK 周波数から独立して動作できます。

この独立したクロックソースは、次の 3 つのクロックソースから選択できます：

- PCLK1 : APB1 クロック (デフォルト値)
- HSI : 高速内部オシレータ
- SYSCLK : システムクロック

詳細については、[セクション 5 : リセットおよびクロック制御 \(RCC\)](#) を参照してください。

## 30.4.2 I2C クロックの要件

I2C カーネルは I2CCLK によってクロック供給されます。

I2CCLK の周期  $t_{I2CCLK}$  は、次の条件を満たす必要があります。

$$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4 \text{ and } t_{I2CCLK} < t_{HIGH}$$

ここで：

$t_{LOW}$  : SCL ロー時間、および  $t_{HIGH}$  : SCL ハイ時間

$t_{filters}$  : 有効なときには、アナログフィルタとデジタルフィルタによる遅延の合計。

アナログフィルタの遅延は、最大 260 ns です。デジタルフィルタの遅延は、 $DNF \times t_{I2CCLK}$  です。

PCLK の周期  $t_{PCLK}$  は、次の条件を満たす必要があります。

$$t_{PCLK} < 4/3 t_{SCL}$$

$t_{SCL}$  : SCL 周期

**注意：** I2C カーネルが PCLK によってクロック供給されると、PCLK は  $t_{I2CCLK}$  の条件を満たす必要があります。

## 30.4.3 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます：

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成したときにはスレーブからマスタへ、アービトレーションの喪失または STOP 生成が発生したときにはマスタからスレーブへ自動的に切り替わるため、マルチマスタ機能を使用できます。

### 通信の流れ

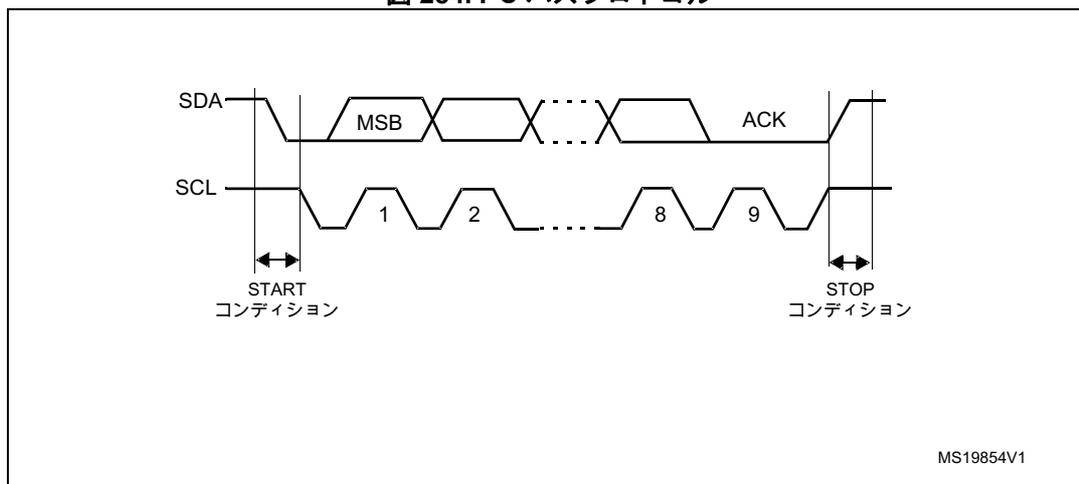
マスタモードでは、I2C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス (7 または 10 ビット) と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効または無効にできます。予約済みの SMBus アドレスもソフトウェアによって有効にできます。

データとアドレスは、MSB ファーストの 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト (7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト) にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります。次の図を参照してください。

図 284. I<sup>2</sup>C バスプロトコル



確認応答 (Acknowledge) は、ソフトウェアによって有効または無効にできます。I2C インタフェースのアドレスは、ソフトウェアによって選択できます。

## 30.4.4 I2C の初期化

### ペリフェラルの有効化と無効化

I2C ペリフェラルクロックは、クロックコントローラで設定し、有効にする必要があります ([セクション 5: リセットおよびクロック制御 \(RCC\)](#) を参照)。

そして、I2C\_CR1 レジスタの PE ビットをセットすることによって、I2C を有効にできます。

I2C が無効なときには (PE=0)、I<sup>2</sup>C はソフトウェアリセットを実行します。詳細については、[セクション 30.4.5: ソフトウェアリセット](#) を参照してください。

### ノイズフィルタ

I2C\_CR1 レジスタの PE ビットをセットすることによって I2C ペリフェラルを有効にする前に、必要な場合は、ノイズフィルタを設定する必要があります。デフォルトでは、SDA および SCL 入力にアナログノイズフィルタがあります。このアナログフィルタは I<sup>2</sup>C 仕様に準拠しており、高速モードおよび高速モードプラスで最大 50 ns のパルス幅を持つスパイクを抑制します。ANFOFF ビットをセットすることによって、このアナログフィルタを無効にし、I2C\_CR1 レジスタの DNF[3:0] ビットを設定することによってデジタルフィルタを選択することができます。

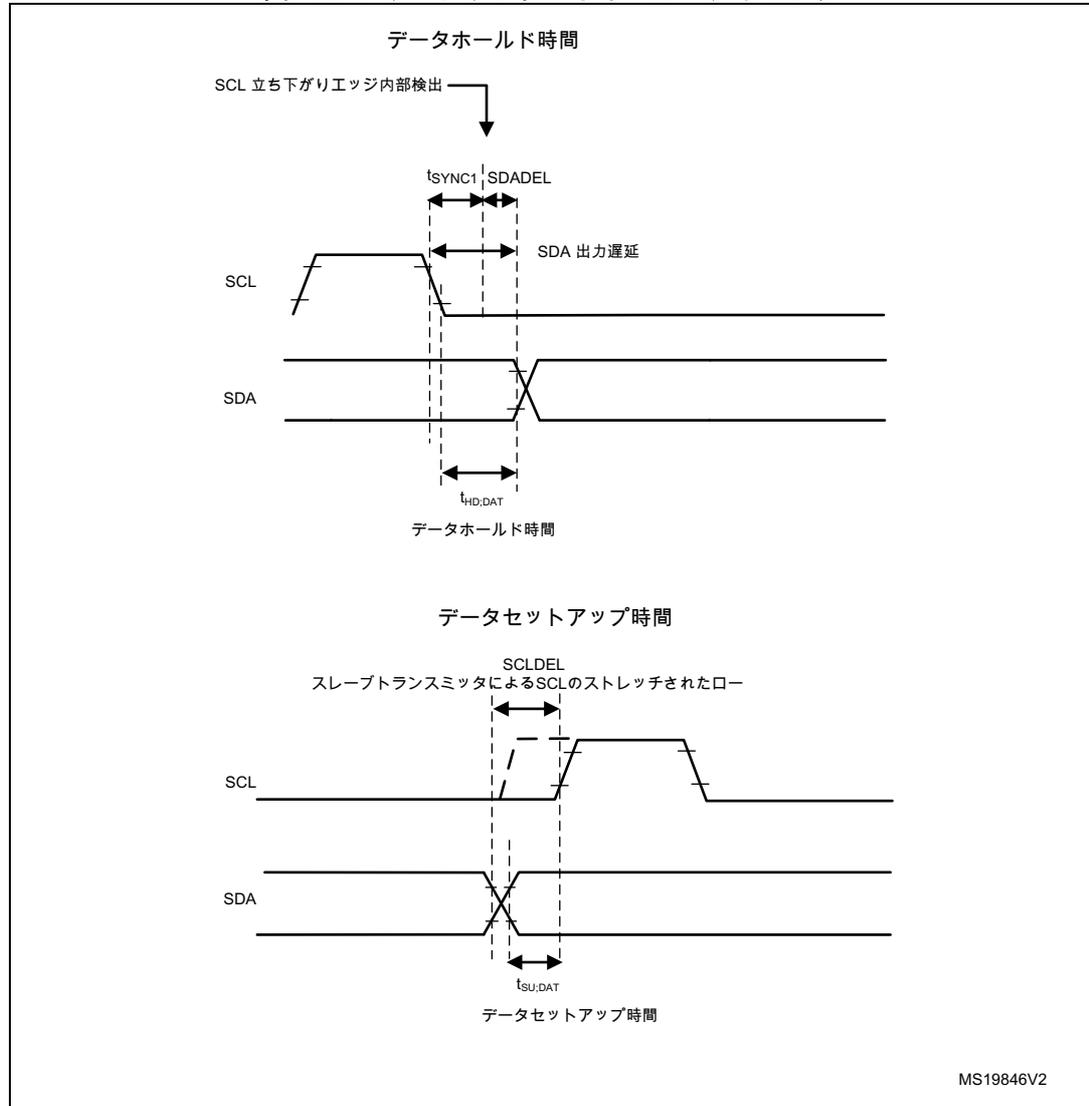
デジタルフィルタが有効なときには、SCL または SDA ラインのレベルは、DNF x I2CCLK 周期より長く安定していた場合のみ、内部で変更されます。これにより、プログラム可能な 1 ~ 15 I2CCLK 周期の長さを持つスパイクを抑制できます。

**注意:** I2C が有効なときには、フィルタ構成の変更はできません。

## I2C のタイミング

マスタおよびスレーブモードで正しいデータホールドおよびセットアップ時間が使用されるのを保証するためには、タイミングを設定する必要があります。これを行うには、I2C\_TIMINGR レジスタの PRESC[3:0]、SCLDEL[3:0]、および SDADEL[3:0] ビットをプログラムします。

図 285. セットアップおよびホールドタイミング



- SCL 立ち下がりエッジが内部で検出されると、SDA 出力を送信する前に遅延が挿入されます。この遅延は、 $t_{SDADEL} = SDADEL \times t_{PRESC} + t_{2CCLK}$  であり  $t_{PRESC} = (PRESC+1) \times t_{2CCLK}$  です。 $t_{SDADEL}$  はホールド時間  $t_{HD,DAT}$  に影響を与えます。

SDA 出力遅延の合計は、次のとおりです：

$$t_{SYNC1} + \{[SDADEL \times (PRESC+1) + 1] \times t_{2CCLK}\}$$

$t_{\text{SYNC1}}$  の長さは、次のパラメータに依存します。

- SCL 立ち下がり傾斜
- アナログフィルタが有効なときの入力遅延:  $t_{\text{AF}(\text{min})} < t_{\text{AF}} < t_{\text{AF}(\text{max})}$  ns
- デジタルフィルタが有効なときの入力遅延:  $t_{\text{DNF}} = \text{DNF} \times t_{\text{I2CCLK}}$
- SCL と I2CCLK クロックの同期による遅延 (2 ~ 3 I2CCLK 周期)

SCL 立ち下がりエッジの未定義の領域をブリッジするためには、SDADEL を次のようにプログラムする必要があります:

$$\{t_{\text{r}(\text{max})} + t_{\text{HD;DAT}(\text{min})} - t_{\text{AF}(\text{min})} - [(DNF + 3) \times t_{\text{I2CCLK}}]\} / \{(PRESC + 1) \times t_{\text{I2CCLK}}\} \leq \text{SDADEL}$$

$$\text{SDADEL} \leq \{t_{\text{HD;DAT}(\text{max})} - t_{\text{AF}(\text{max})} - [(DNF + 4) \times t_{\text{I2CCLK}}]\} / \{(PRESC + 1) \times t_{\text{I2CCLK}}\}$$

**注:**  $t_{\text{AF}(\text{min})} / t_{\text{AF}(\text{max})}$  は、アナログフィルタが有効なときのみ、等式に含まれます。 $t_{\text{AF}}$  の値については、デバイスのデータシートを参照してください。

最大  $t_{\text{HD;DAT}}$  は、標準モード、高速モード、および高速モードプラスで 3.45  $\mu\text{s}$ 、0.9  $\mu\text{s}$ 、および 0.45  $\mu\text{s}$  ですが、遷移時間による  $t_{\text{VD;DAT}}$  の最大値より短い必要があります。この最大値を満たす必要があるのは、デバイスが SCL 信号の LOW 周期 ( $t_{\text{LOW}}$ ) をストレッチしない場合だけです。クロックが SCL をストレッチする場合、クロックをリリースする前に、データがセットアップ時間まで有効である必要があります。

SDA 立ち上がりエッジは、通常、最悪ケースであり、この場合、前の等式は次のようになります:

$$\text{SDADEL} \leq \{t_{\text{VD;DAT}(\text{max})} - t_{\text{r}(\text{max})} - 260 \text{ ns} - [(DNF + 4) \times t_{\text{I2CCLK}}]\} / \{(PRESC + 1) \times t_{\text{I2CCLK}}\}$$

**注:** **NOSTRETCH=0** のときには、SCLDEL の値に従って、デバイスはセットアップ時間を保証するために SCL ローをストレッチするので、この条件に違反することがあります。

$t_{\text{r}}$ 、 $t_{\text{HD;DAT}}$ 、および  $t_{\text{VD;DAT}}$  の標準値については、表 154: I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を参照してください。

- SDA 出力の送信後、SCL ラインはセットアップ時間中、ローレベルに保たれます。このセットアップ時間は、 $t_{\text{SCLDEL}} = (\text{SCLDEL} + 1) \times t_{\text{PRESC}}$  であり  $t_{\text{PRESC}} = (\text{PRESC} + 1) \times t_{\text{I2CCLK}}$  です。

$t_{\text{SCLDEL}}$  は、セットアップ時間  $t_{\text{SU;DAT}}$  に影響を与えます。

SDA 遷移 (立ち上がりエッジは通常、最悪のケース) の未定義の領域をブリッジするためには、SCLDEL を次のようにプログラムする必要があります:

$$\{[t_{\text{r}(\text{max})} + t_{\text{SU;DAT}(\text{min})}] / [(\text{PRESC} + 1)] \times t_{\text{I2CCLK}}\} - 1 \leq \text{SCLDEL}$$

$t_{\text{r}}$  および  $t_{\text{SU;DAT}}$  の標準値については、表 154: I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を参照してください。

使用される SDA および SCL 遷移時間の値は、アプリケーションの値です。標準から最大値を使用すると、SDADEL と SCLDEL の計算の制約が増えますが、アプリケーションにかかわらず、この機能を使用できます。

表 154. I<sup>2</sup>C-SMBUS 仕様のデータのセットアップおよびホールド時間

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
t <sub>HD;DAT</sub>	データホールド時間	0	-	0	-	0	-	0.3	-	μs
t <sub>VD;DAT</sub>	データ有効時間	-	3.45	-	0.9	-	0.45	-	-	
t <sub>SU;DAT</sub>	データセットアップ時間	250	-	100	-	50	-	250	-	ns
t <sub>r</sub>	SDA および SCL 信号の立ち上がり時間	-	1000	-	300	-	120	-	1000	
t <sub>f</sub>	SDA および SCL 信号の立ち下がり時間	-	300	-	300	-	120	-	300	

また、マスタモードでは、I2C\_TIMINGR レジスタの PRESC[3:0]、SCLH[7:0]、および SCLL[7:0] ビットをプログラムすることによって、SCL クロックのハイおよびローレベルを設定する必要があります。

- SCL 立ち下がりエッジが内部で検出されると、SCL 出力をリリースする前に遅延が挿入されます。この遅延は、 $t_{SCLL} = (SCLL+1) \times t_{PRESC}$  であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCCLK}$  です。  
t<sub>SCLL</sub> は、SCL ロー時間 t<sub>LOW</sub> に影響を与えます。
- SCL 立ち上がりエッジが内部で検出されると、SCL 出力を強制的にローレベルにする前に遅延が挿入されます。この遅延は、 $t_{SCLH} = (SCLH+1) \times t_{PRESC}$  であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCCLK}$  です。t<sub>SCLH</sub> は、SCLハイ時間 t<sub>HIGH</sub> に影響を与えます。

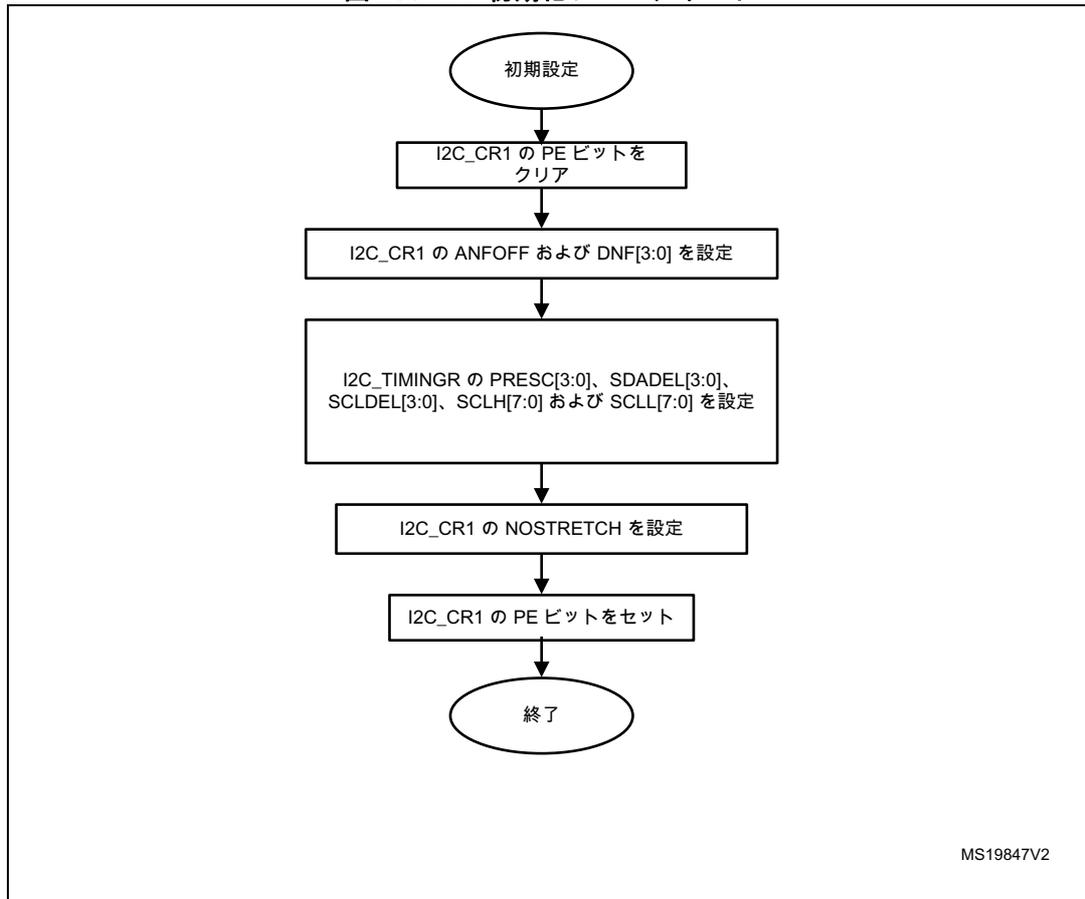
詳細については、セクション [: I2C マスタ初期化](#) を参照してください。

**注意 :** I2C が有効なときには、タイミング構成の設定はできません。

ペリフェラルを有効にする前に、I2C スレーブ NOSTRETCH モードも設定する必要があります。詳細については、[: I2C スレーブ初期化](#) を参照してください。

**注意 :** I2C が有効なときには、NOSTRETCH 構成の変更はできません。

図 286. I2C 初期化フローチャート



## 30.4.5 ソフトウェアリセット

ソフトウェアリセットを行うには、I2C\_CR1 レジスタの PE ビットをクリアします。その場合、I2C のライン SCL および SDA がリリースされます。内部状態マシンがリセットされ、通信制御ビットとステータスビットがリセット値に戻ります。構成レジスタは影響を受けません。

影響を受けるレジスタのビットは、以下のとおりです：

1. I2C\_CR2 レジスタ：START、STOP、NACK
2. I2C\_ISR レジスタ：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO、OVR

SMBus 機能がサポートされるときには、以下も影響を受けます：

1. I2C\_CR2 レジスタ：PECBYTE
2. I2C\_ISR レジスタ：PECERR、TIMEOUT、ALERT

ソフトウェアリセットを実行するためには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たなければならない。このためには、次のソフトウェアシーケンスを書き込みます：- PE=0 を書き込む - PE=0 を確認する - PE=1 を書き込む

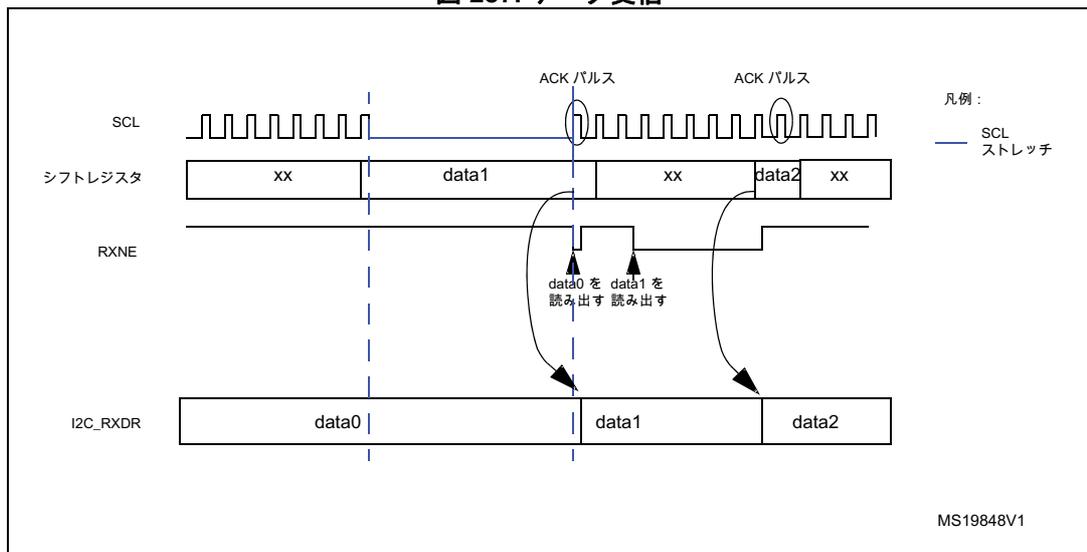
## 30.4.6 データ転送

データ転送は、送受信データレジスタとシフトレジスタを通じて管理されます。

### 受信

SDA 入力はシフトレジスタに送られます。8 番目の SCL パルスの後 (完全なデータバイトの受信後)、シフトレジスタは、I2C\_RXDR レジスタが空の場合 (RXNE=0)、このレジスタにコピーされます。RXNE=1 の場合、すなわち、前に受信されたデータバイトがまだ読み出されていなかった場合、SCL ラインは I2C\_RXDR が読み出されるまでストレッチされます。ストレッチは、8 番目と 9 番目の SCL パルスの間 (確認応答パルスの前) に挿入されます。

図 287. データ受信

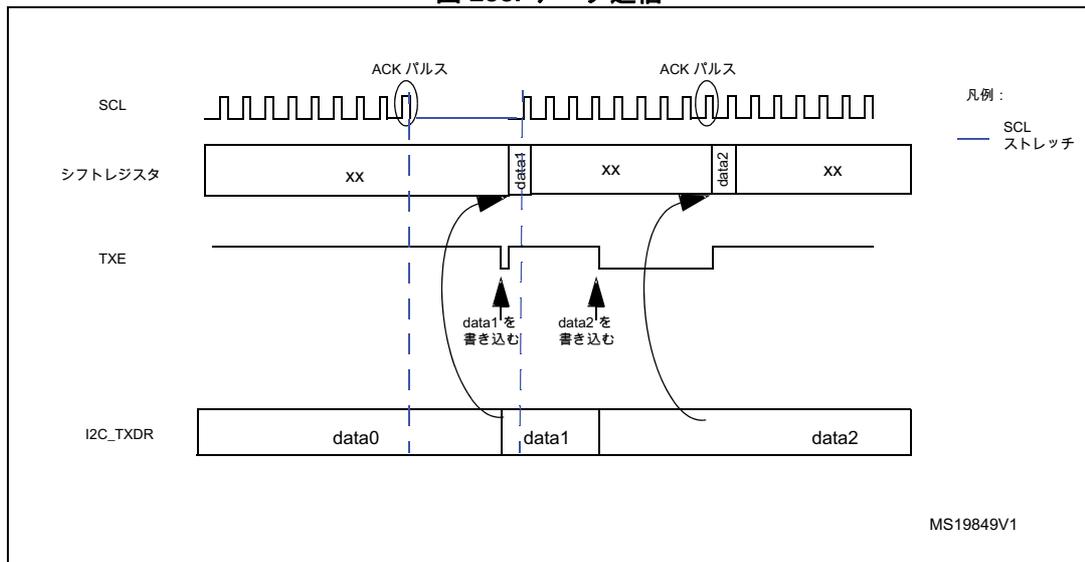


MS19848V1

### 送信

I2C\_TXDR レジスタが空 (TXE=0) でない場合、その内容が 9 番目の SCL パルス (確認応答パルス) の後、シフトレジスタにコピーされます。次に、シフトレジスタの内容が SDA ラインにシフトアウトされます。TXE=1 の場合、すなわち、I2C\_TXDR にデータがまだ書き込まれていない場合、SCL ラインは I2C\_TXDR に書き込まれるまでストレッチされます。ストレッチは、9 番目の SCL パルスの後で行われます。

図 288. データ送信



## ハードウェア転送管理

次のようにさまざまなモードでバイト転送を管理し、通信をクローズするために、I2C にはハードウェアにバイトカウンタが組み込まれています。

- マスタモードでの NACK、STOP、および ReSTART 生成
- スレーブレシーバモードでの ACK 制御
- SMBus 機能がサポートされているときの PEC 生成/確認

バイトカウンタは、マスタモードでは常に使用されます。デフォルトでは、スレーブモードでは無効ですが、I2C\_CR2 レジスタの SBC (スレーブバイト制御) ビットをセットすることによって、ソフトウェアにより有効にできます。

転送されるバイト数は、I2C\_CR2 レジスタの NBYTES[7:0] ビットフィールドでプログラムされます。転送バイト数 (NBYTES) が 255 より大きい場合、またはレシーバが受信データバイトの確認応答値を制御したい場合には、I2C\_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。このモードでは、NBYTES でプログラムされたバイト数が転送されると、TCR フラグがセットされ、TCIE がセットされている場合は割り込みが生成されます。SCL は、TCR フラグがセットされている間、ストレッチされます。TCR は、NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

NBYTES カウンタに最後のバイト数が再ロードされたときには、RELOAD ビットがクリアされる必要があります。

マスタモードで RELOAD=0 のときには、カウンタは 2 つのモードで使用できます：

- **自動終了モード** (I2C\_CR2 レジスタの AUTOEND = "1")。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、マスタは STOP コンディションを自動的に送信します。
- **ソフトウェア終了モード** (I2C\_CR2 レジスタの AUTOEND = 0)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、ソフトウェアアクションが求められます。TC フラグがセットされ、TCIE ビットがセットされている場合は割り込みが生成されます。SCL 信号は、TC フラグがセットされている間、ストレッチされます。TC フラグは、I2C\_CR2 レジスタの START または STOP ビットがセットされたときに、ソフトウェアによってクリアされます。マスタが RESTART コンディションを送信したいときには、このモードを使用する必要があります。

**注意：** AUTOEND ビットは、RELOAD ビットがセットされているときには効果がありません。

表 155. I2C 設定表

機能	SBC ビット	RELOAD ビット	AUTOEND ビット
マスタ Tx/Rx NBYTES + STOP	x	0	1
マスタ Tx/Rx + NBYTES + RESTART	x	0	0
スレーブ Tx/Rx すべての受信バイトに ACK	0	x	x
スレーブ Rx および ACK 制御	1	1	x

## 30.4.7 I2C スレーブモード

### I2C スレーブ初期化

スレーブモードで動作するには、少なくとも 1 つのスレーブアドレスを有効にする必要があります。2 つのレジスタ I2C\_OAR1 と I2C\_OAR2 を使用して、スレーブ専用アドレス OA1 および OA2 をプログラムできます。

- OA1 は、I2C\_OAR1 レジスタの OA1MODE ビットをセットすることによって、7 ビットモード (デフォルト) または 10 ビットアドレッシングモードに設定できます。

OA1 を有効にするには、I2C\_OAR1 レジスタの OA1EN ビットをセットします。

- 追加のスレーブアドレスが必要な場合は、2 番目のスレーブアドレス OA2 を設定できます。I2C\_OAR2 レジスタの OA2MSK[2:0] ビットを設定することによって、最大 7 つの OA2 LSB をマスクできます。したがって、OA2MSK が 1 から 6 まで設定された場合、OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]、または OA2[7] のみが受信アドレスと比較されます。OA2MSK が 0 に等しくなくなるとすぐに、OA2 のアドレスコンパレータは、確認応答されない I2C 予約済みアドレス (0000 XXX および 1111 XXX) を除外します。OA2MSK=7 の場合、受信されたすべてのアドレスが確認応答されます (予約済みアドレスを除く)。OA2 は常に 7 ビットアドレスです。

これらの予約済みアドレスは、特定のイネーブルビットによって有効化された場合、I2C\_OAR1 または I2C\_OAR2 レジスタが OA2MSK=0 でプログラムされた場合、確認応答できます。

OA2 を有効にするには、I2C\_OAR2 レジスタの OA2EN ビットをセットします。

- 同報アドレスは、I2C\_CR1 レジスタの GCEN ビットをセットすることで有効になります。

I2C が有効アドレスの 1 つによって選択されると、ADDR 割り込みステータスフラグがセットされ、ADDRIE ビットがセットされている場合は割り込みが生成されます。

デフォルトでは、スレーブはクロックストレッチ機能を使用し、必要なときには、ソフトウェアアクションを実行するために、SCL 信号をローレベルでストレッチすることを意味します。マスタがクロックストレッチをサポートしない場合、I2C\_CR1 レジスタの NOSTRETCH=1 で I2C を設定する必要があります。

ADDR 割り込みの受信後、いくつかのアドレスが有効な場合は、I2C\_ISR レジスタの ADDCODE[6:0] ビットを読み出して、一致するアドレスを確認する必要があります。転送方向を知るために、DIR フラグも確認する必要があります。

## スレーブクロックストレッチ (NOSTRETCH = 0)

デフォルトモードでは、I2C スレーブは次の状況で SCL クロックをストレッチします：

- ADDR フラグがセットされると：受信アドレスは有効なスレーブアドレスの 1 つと一致します。このストレッチは、ADDRCF ビットをセットすることによりソフトウェアによって ADDR フラグがクリアされたときにリリースされます。
- 送信時、前のデータ送信が完了し、新しいデータが I2C\_TXDR レジスタに書き込まれなかった場合、または ADDR フラグがクリアされたときに (TXE=1)、最初のデータバイトが書き込まれていなかった場合。このストレッチは、データが I2C\_TXDR レジスタに書き込まれたときにリリースされます。
- 受信時、I2C\_RXDR レジスタがまだ読み出されておらず、新しいデータ受信が完了したとき。このストレッチは、I2C\_RXDR が読み出されたときにリリースされます。
- スレーブバイト制御モードおよび再ロードモード (SBC=1 および RELOAD=1) で TCR = 1 のとき、すなわち、最後データバイトが転送されたとき。このストレッチは、NBYTES[7:0] フィールドにゼロ以外の値を書き込むことによって TCR がクリアされたときにリリースされます。
- SCL 立ち下がリエッジの検出後、I2C は、 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCLK}$  の間、SCL ローをストレッチします。

## クロックストレッチなしのスレーブ (NOSTRETCH = 1)

I2C\_CR1 レジスタの NOSTRETCH = 1 のとき、I2C スレーブは SCL 信号をストレッチしません。

- ADDR フラグがセットされている間、SCL クロックはストレッチされません。
- 送信時、転送に対応する最初の SCL パルスが発生する前に、I2C\_TXDR レジスタにデータが書き込まれる必要があります。そうでない場合、アンダーランが発生し、I2C\_ISR レジスタで OVR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。OVR フラグは、最初のデータ送信が開始し、STOPF ビットがまだセットされている (クリアされていない) ときにもセットされます。したがって、次の転送で送信される最初のデータを書き込んだ後でのみ、前に転送の STOPF フラグをクリアすることによって、送信される最初のデータについても、OVR ステータスが提供されることを確実にできます。
- 受信時、次のデータバイトの 9 番目の SCL パルス (ACK パルス) が発生する前に、I2C\_RXDR レジスタからデータが読み出される必要があります。そうでない場合、オーバーランが発生し、I2C\_ISR レジスタの OVR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## スレーブバイト制御モード

スレーブ受信モードでバイト ACK 制御を可能にするためには、I2C\_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。これは、SMBus 標準に準拠する必要があります。

スレーブ受信モードでバイト ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD=1)。各バイトの制御を得るには、ADDR 割り込みサブルーチンで NBYTES を 0x1 に初期化し、各受信バイト後に 0x1 に再ロードする必要があります。バイトが受信されると、TCR ビットがセットされ、8 番目と 9 番目の SCL パルスの間で、SCL 信号ローをストレッチします。I2C\_RXDR レジスタからデータを読み出すことができ、その後、I2C\_CR2 レジスタの ACK ビットを設定することによって、確認応答するかどうかを決定できます。SCL ストレッチは、NBYTES をゼロ以外の値にプログラムすることによってリリースされ、確認応答または非確認応答が送信され、次のバイトを受信できます。

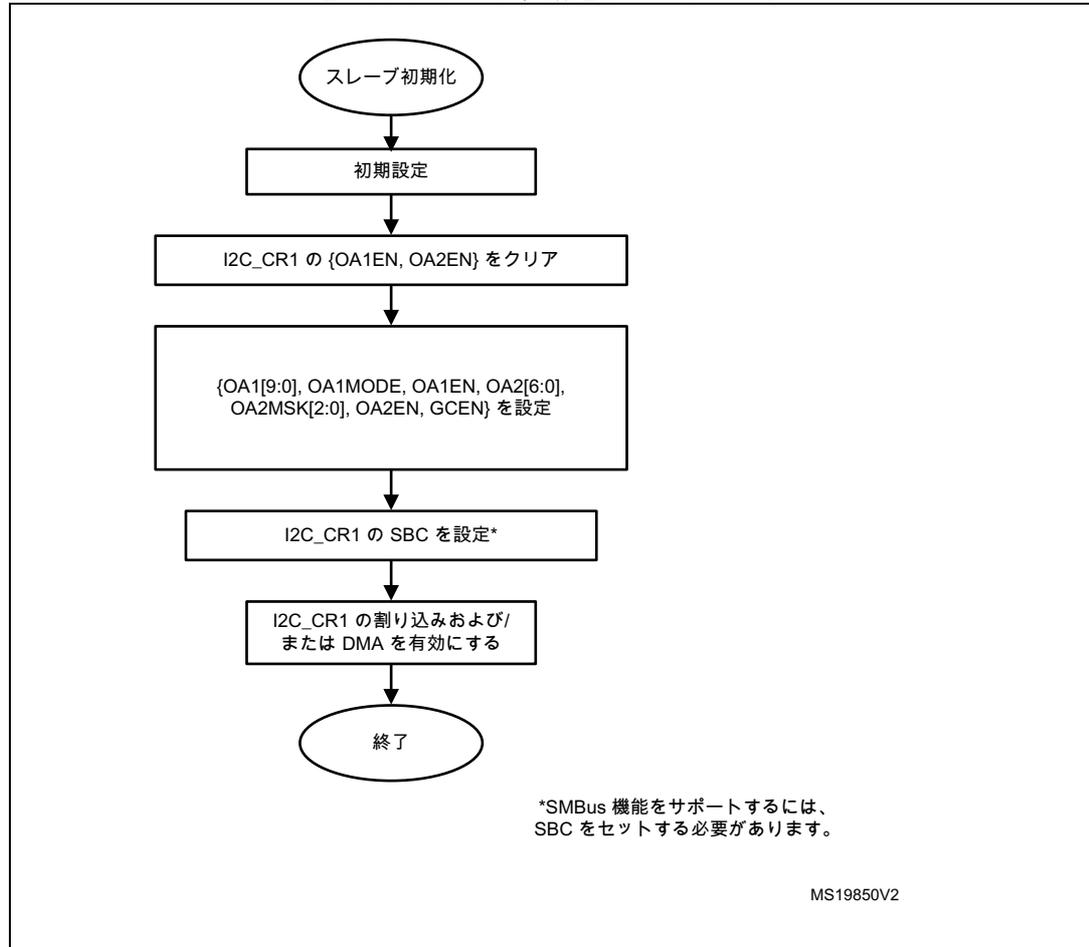
NBYTES に 0x1 より大きい値をロードでき、この場合、受信フローは NBYTES データ受信、継続します。

**注：** SBC ビットは、I2C が無効なとき、またはスレーブがアドレス指定されていないとき、または ADDR=1 のときに設定する必要があります。

RELOAD ビットの値は、ADDR=1 のとき、または TCR=1 のときに変更できます。

**注意：** スレーブバイト制御モードは、NOSTRETCH モードと互換性がありません。NOSTRETCH=1 のときに SBC をセットすることはできません。

図 289. スレーブ初期化フローチャート



## スレーブトランスミッタ

I2C\_TXDR レジスタが空になると、送信割り込みステータス (TXIS) が生成されます。I2C\_CR1 レジスタの TXIE ビットがセットされている場合は、割り込みが生成されます。

TXIS ビットは、I2C\_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

NACK が受信されると、I2C\_ISR レジスタの NACKF ビットがセットされ、I2C\_CR1 レジスタの NACKIE ビットがセットされていた場合は割り込みが生成されます。マスタが STOP または RESTART コンディションを実行できるように、スレーブは SCL および SDA ラインを自動的にリリースします。TXIS ビットは、NACK 受信時にはセットされません。

STOP が受信され、I2C\_CR1 レジスタの STOPIE ビットがセットされると、I2C\_ISR レジスタの STOPF フラグがセットされ、割り込みが生成されます。ほとんどのアプリケーションでは、SBC は通常、0 にプログラムされます。この場合、スレーブアドレスが受信されたときに (ADDR=1)、TXE

= 0 であった場合、I2C\_TXDR レジスタの内容を最初のデータバイトとして送信するか、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C\_TXDR レジスタをフラッシュするかを選択できます。

スレーブバイト制御モード (SBC=1) では、送信バイト数をアドレス一致割り込みサブルーチンの NBYTES でプログラムする必要があります (ADDR=1)。この場合、転送中の TXIS イベントの数は、NBYTES でプログラムされた値に対応します。

**注意 :** *NOSTRETCH=1 のとき、SCL クロックは ADDR フラグがセットされている間はストレッチされないため、最初のデータバイトをプログラムするために ADDR サブルーチンで I2C\_TXDR レジスタの内容をフラッシュすることはできません。最初に送信されるデータバイトは、I2C\_TXDR レジスタで前もってプログラムされている必要があります。*

- このデータは、前の送信メッセージの最後の TXIS イベントで書き込まれたデータでもかまいません。
- このデータバイトが送信データバイトでない場合、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C\_TXDR レジスタをフラッシュできます。アドレスの確認応答に続いて、最初のデータ送信が開始する前にこれらが実行されることを保証するためには、STOPF ビットのクリアは、これらのアクションの後でのみ行う必要があります。

最初のデータ送信が開始したときに STOPF がまだセットされていた場合、アンダーランエラーが生成されます (OVR フラグがセットされます)。

TXIS イベントが必要な場合 (送信割り込みまたは送信 DMA リクエスト)、TXIS イベントを生成するためには、TXE ビットに加えて TXIS ビットもセットする必要があります。

図 290. I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0)

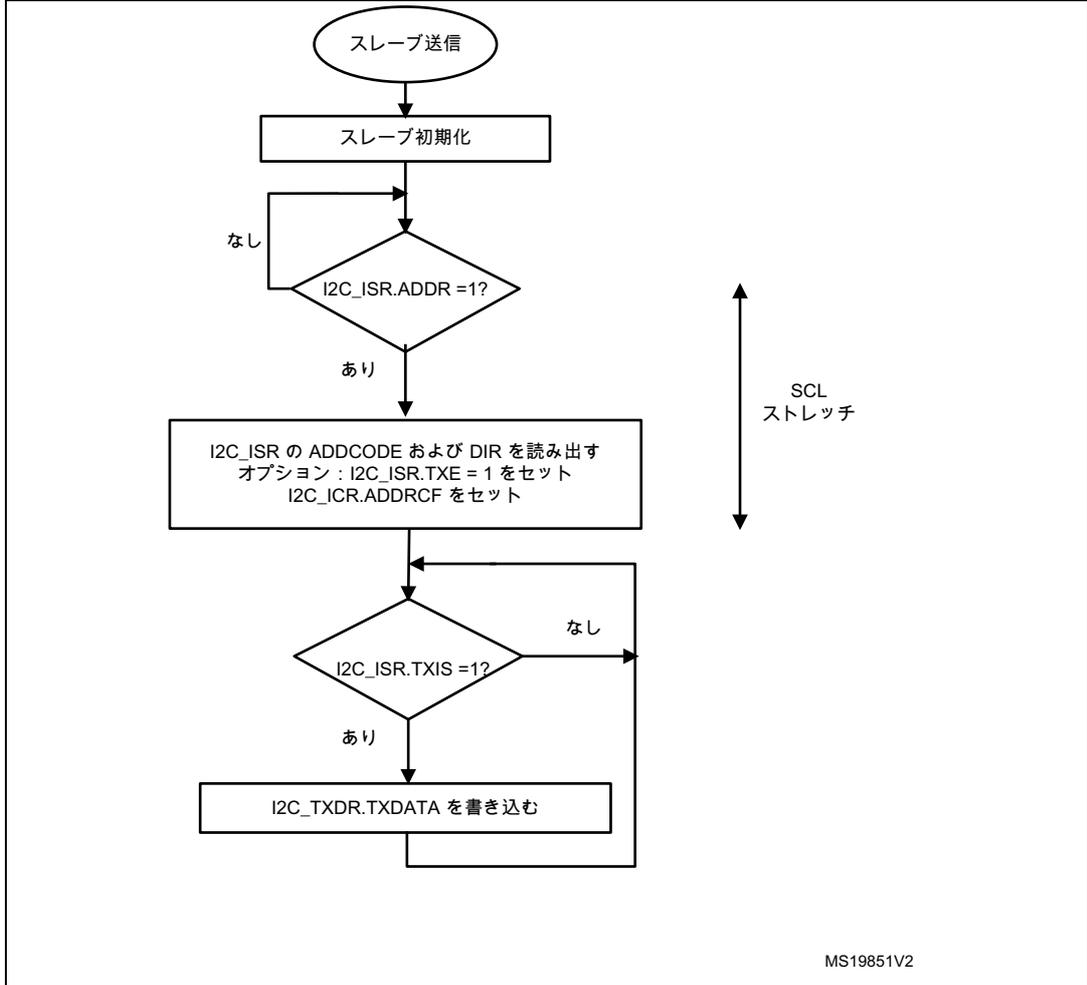


図 291. I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1)

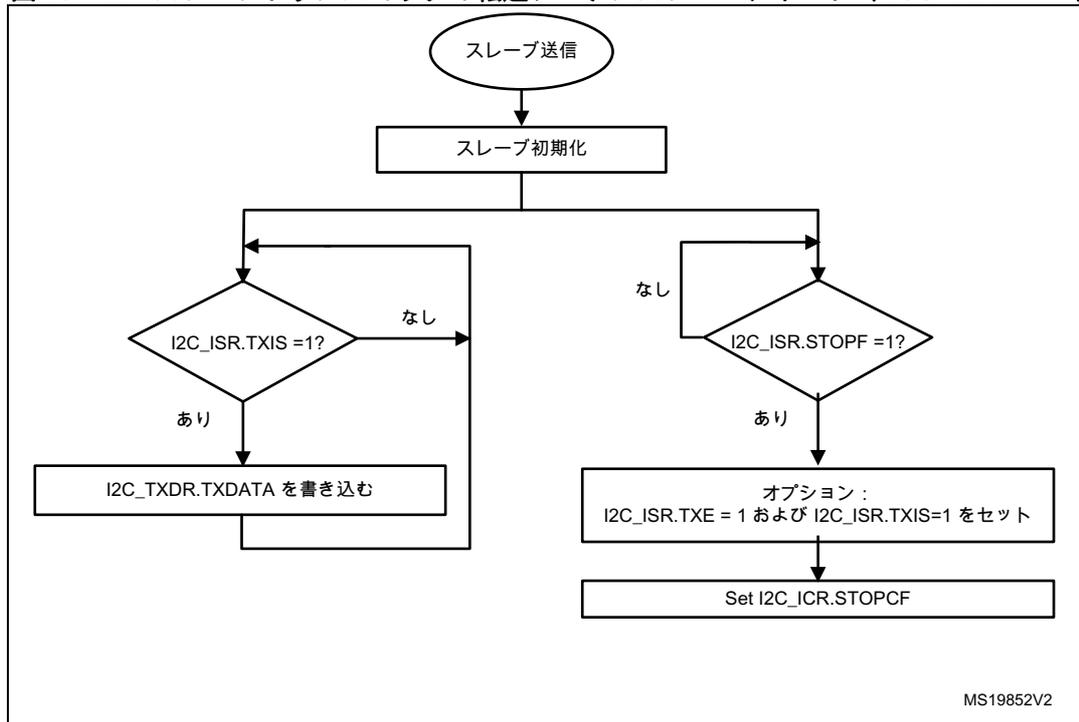
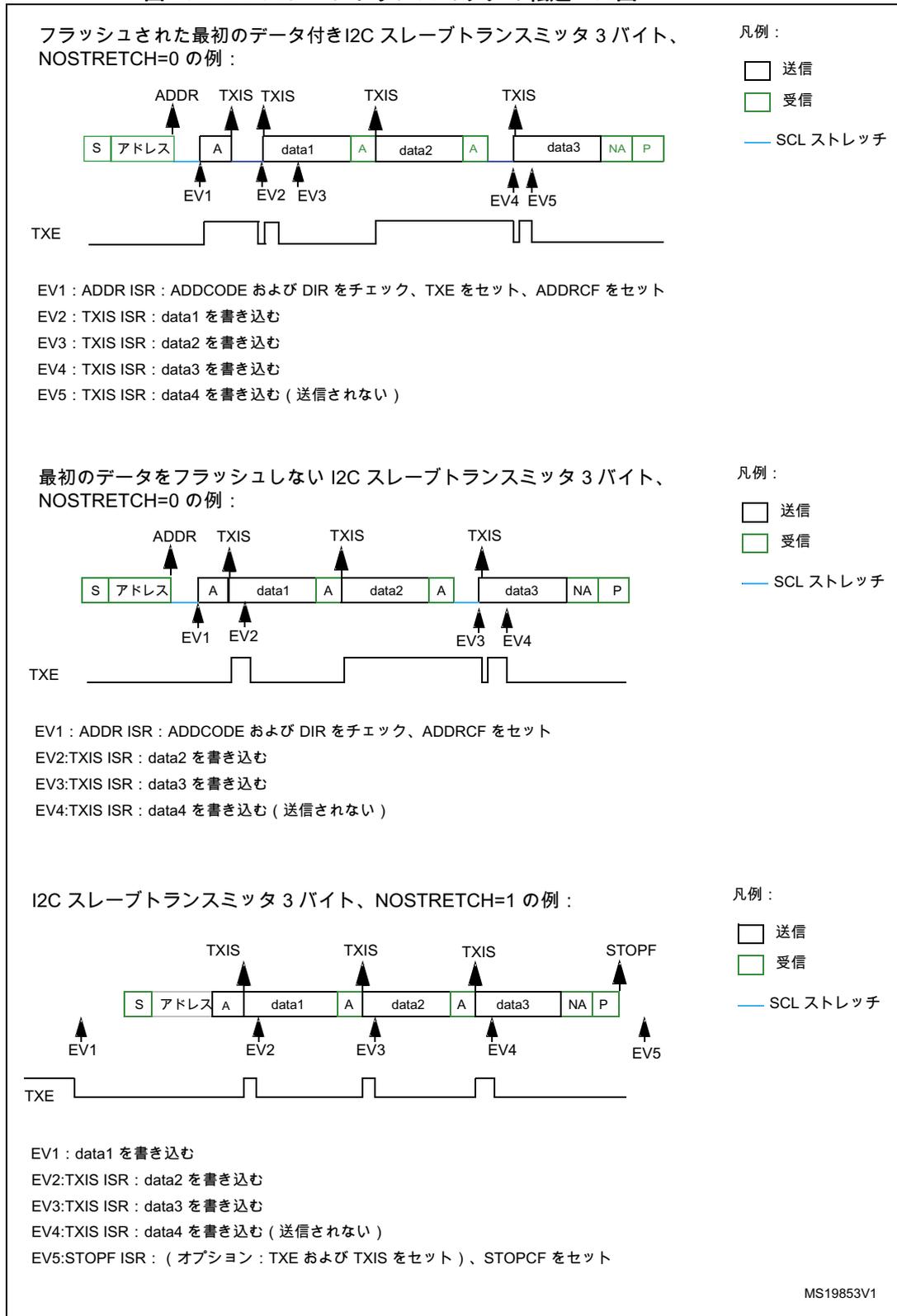


図 292. I2C スレーブトランスミッタの転送バス図



## スレーブレシーバ

I2C\_RXDR がフルのときには、I2C\_ISR の RXNE がセットされ、I2C\_CR1 の RXIE がセットされている場合は割り込みが生成されます。RXNE は、I2C\_RXDR が読み出されたときにクリアされます。

STOP が受信され、I2C\_CR1 レジスタの STOPIE ビットがセットされると、I2C\_ISR の STOPF がセットされ、割り込みが生成されます。

図 293. スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0)

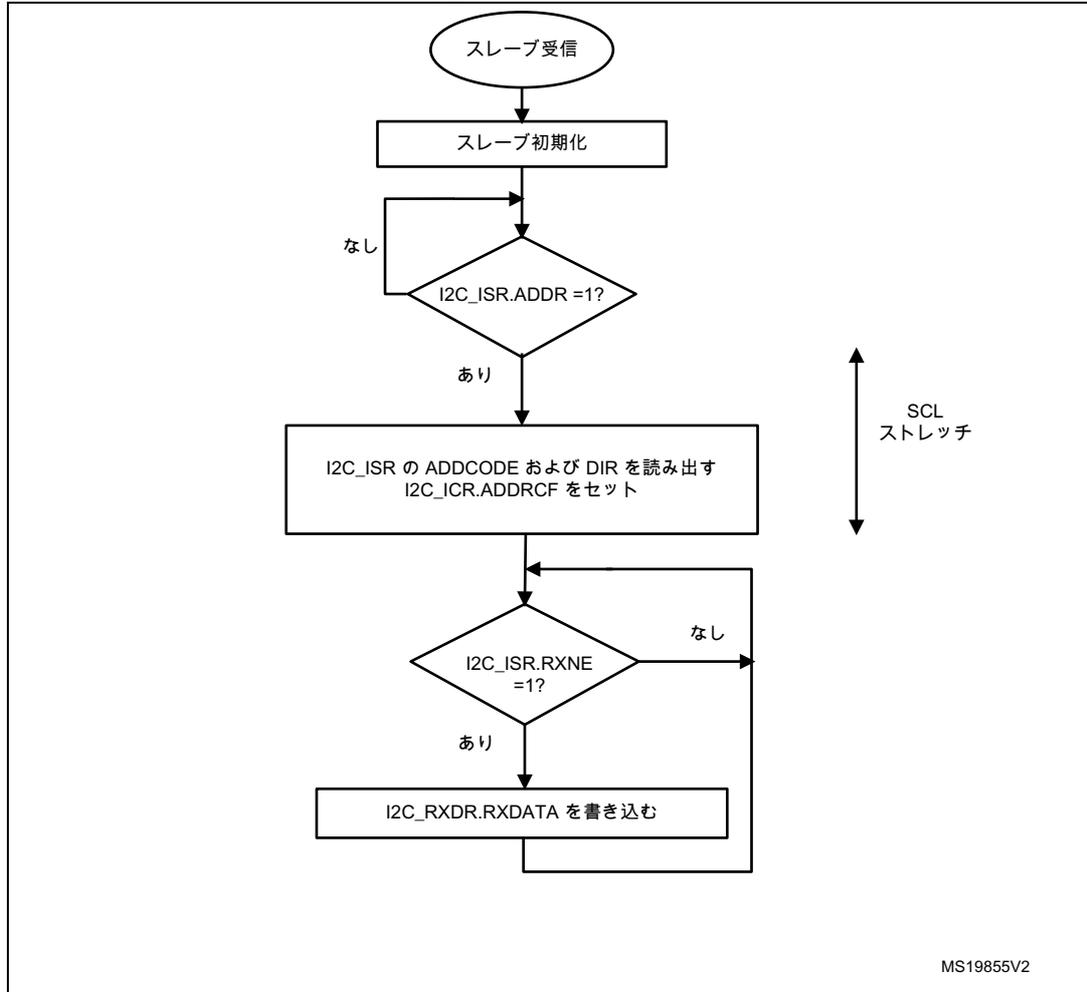


図 294. スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1)

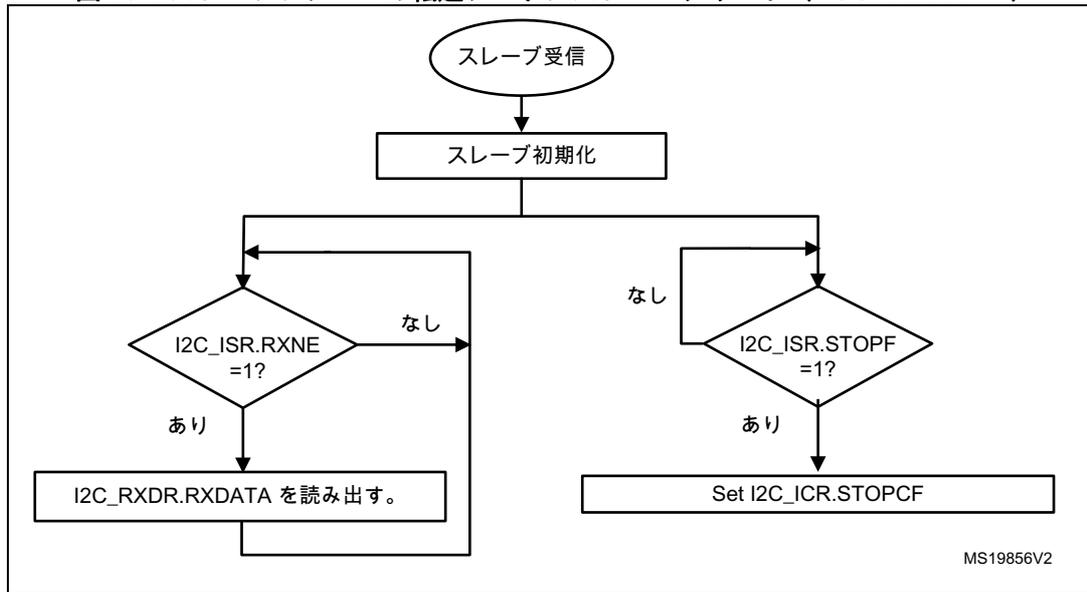
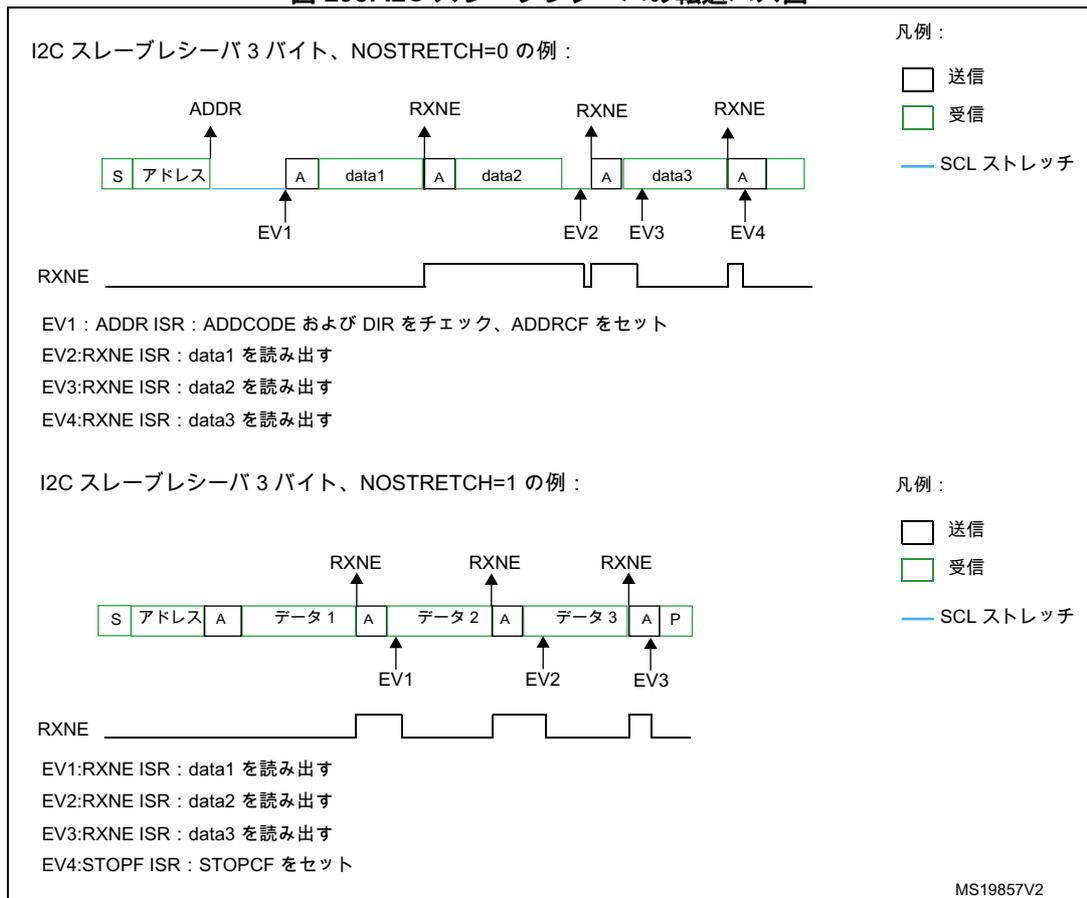


図 295. I2C スレーブレシーバの転送バス図



## 30.4.8 I2C マスタモード

### I2C マスタ初期化

ペリフェラルを有効にする前に、I2C\_TIMINGR レジスタの SCLH および SCLL ビットをセットすることによって、I2C マスタクロックを設定する必要があります。

マルチマスタ環境とスレーブクロックストレッチをサポートするために、クロック同期メカニズムが実装されています。

クロック同期を可能にするために：

- クロックのローレベルは SCLL カウンタを使用してカウントされ、SCL ローレベル内部検出から開始されます。
- クロックのハイレベルは SCLH カウンタを使用してカウントされ、SCL ハイレベル内部検出から開始されます。

I2C は、SCL 立ち下がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延  $t_{\text{SYNC1}}$  の後に SCL ローレベルを検出します。SCLL カウンタが I2C\_TIMINGR レジスタの SCLL[7:0] ビットでプログラムされた値に達すると、I2C は SCL をハイレベルにリリースします。

I2C は、SCL 立ち上がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延  $t_{\text{SYNC2}}$  の後に SCL ハイレベルを検出します。SCLH カウンタが I2C\_TIMINGR レジスタの SCLH[7:0] ビットでプログラムされた値に達すると、I2C は SCL をローレベルにします。

結果として、マスタクロック周期は次のとおりです：

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}+1) + (\text{SCLL}+1)] \times (\text{PRESC}+1) \times t_{\text{I2CCLK}}\}$$

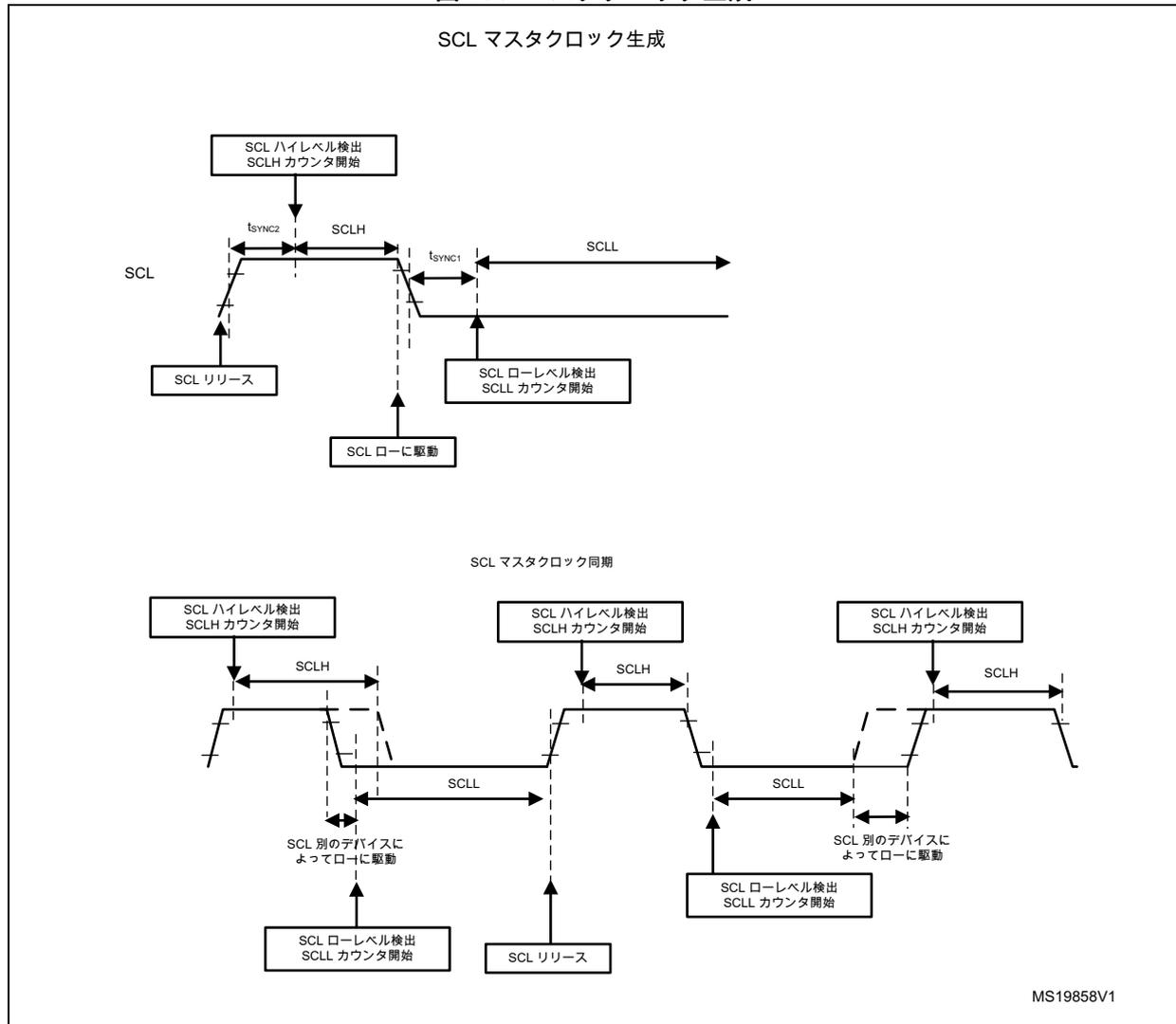
$t_{\text{SYNC1}}$  の長さは、次のパラメータに依存します：

- SCL 立ち下がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延： $\text{DNF} \times t_{\text{I2CCLK}}$
- I2CCLK クロックとの SCL 同期による遅延（2 ~ 3 I2CCLK 周期）

$t_{\text{SYNC2}}$  の長さは、次のパラメータに依存します：

- SCL 立ち上がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延： $\text{DNF} \times t_{\text{I2CCLK}}$
- I2CCLK クロックとの SCL 同期による遅延（2 ~ 3 I2CCLK 周期）

図 296. マスタクロック生成



MS19858V1

注意: I<sup>2</sup>C または SMBus 準拠のためには、マスタクロックは次のタイミングを満たす必要があります:

表 156. I<sup>2</sup>C-SMBUS 仕様のクロックタイミング

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モード プラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
f <sub>SCL</sub>	SCL クロック周波数		100		400		1000		100	kHz
t <sub>HD:STA</sub>	(反復) START コンディションのホールド時間	4.0	-	0.6		0.26	-	4.0	-	μs
t <sub>SU:STA</sub>	反復 START コンディションのセットアップ時間	4.7	-	0.6		0.26	-	4.7	-	μs
t <sub>SU:STO</sub>	STOP コンディションのセットアップ時間	4.0	-	0.6		0.26	-	4.0	-	μs



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

表 156. I<sup>2</sup>C-SMBUS 仕様のクロックタイミング (続き)

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モード プラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
t <sub>BUF</sub>	STOP コンディションと START コンディションの間のバスフリー時間	4.7	-	1.3		0.5	-	4.7	-	μs
t <sub>LOW</sub>	SCL クロックのロー周期	4.7	-	1.3		0.5	-	4.7	-	μs
t <sub>HIGH</sub>	SCL クロックの周期	4.0	-	0.6		0.26	-	4.0	50	μs
t <sub>r</sub>	SDA および SCL 信号の立ち上がり時間	-	1000	-	300		120	-	1000	ns
t <sub>f</sub>	SDA および SCL 信号の立ち下がり時間	-	300	-	300		120	-	300	ns

注: SCLL は、t<sub>BUF</sub> および t<sub>SU:STA</sub> タイミングの生成にも使用されます。

SCLH は、t<sub>HD:STA</sub> および t<sub>SU:STO</sub> タイミングの生成にも使用されます。

I2C\_TIMINGR 設定と I2CCLK 周波数の例については、[セクション 30.4.9: I2C\\_TIMINGR レジスタの設定例](#)を参照してください。

### マスタ通信の初期化 (アドレスフェーズ)

通信を初期化するためには、I2C\_CR2 レジスタでアドレス指定されたスレーブについて次のパラメータをプログラムする必要があります。

- アドレッシングモード (7 ビットまたは 10 ビット) : ADD10
- 送信されるスレーブアドレス : SADD[9:0]
- 転送方向 : RD\_WRN
- 10 ビットアドレスが読み出される場合 : HEAD10R ビット。HEAD10R を設定して、完全なアドレスシーケンスが送信されなければならないか、ヘッダのみ (方向の変更の場合) かを示す必要があります。
- 転送されるバイト数 : NBYTES[7:0]。バイト数が 255 バイト以上の場合、NBYTES[7:0] に 0xFF を書き込む必要があります。

次に、I2C\_CR2 レジスタの START ビットをセットする必要があります。START ビットがセットされているとき、上記のすべてのビットを変更することはできません。

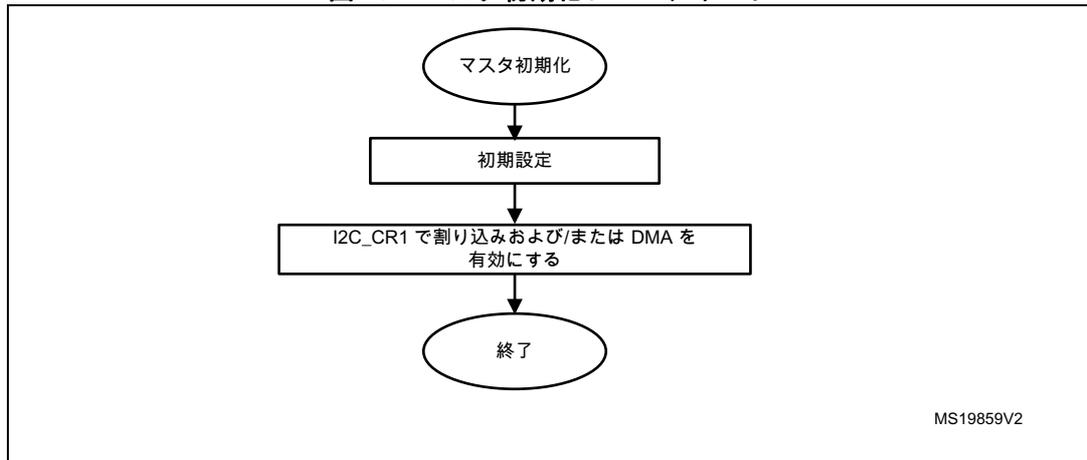
その場合、マスタは、バスがフリーである (BUSY = 0) ことを検出すると、t<sub>BUF</sub> の遅延後に、自動的に START コンディションとスレーブアドレスを送信します。

アービトレーション喪失の場合、マスタはスレーブモードに自動的に切り替えて、スレーブとしてアドレス指定された場合は専用アドレスを確認応答できます。

注: START ビットは、スレーブアドレスがバスに送信されたとき、受信した確認応答値にかかわらず、ハードウェアによってリセットされます。START ビットは、アービトレーション喪失が発生した場合にも、ハードウェアによってリセットされます。START ビットがセットされているときに、I2C がスレーブとしてアドレス指定された場合 (ADDR=1)、I2C はスレーブモードに切り替わり、ADDRCF ビットがセットされたときに START ビットがクリアされます。

注: 反復スタートコンディションにも同じ手順が適用されます。この場合、BUSY=1 です。

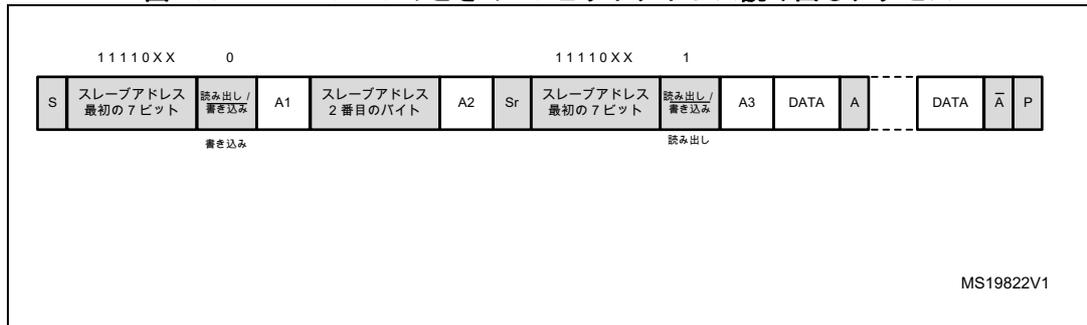
図 297. マスタ初期化フローチャート



## 10 ビットアドレススレーブをアドレス指定するマスタレシーバの初期化

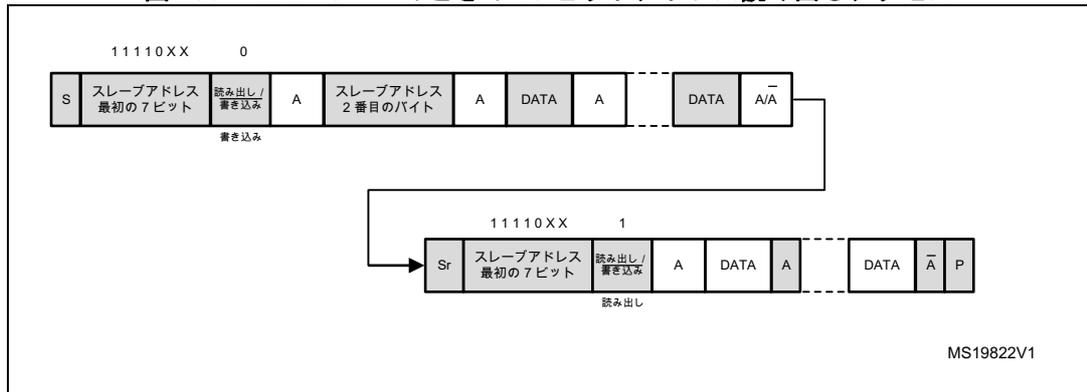
- スレーブアドレスが 10 ビット形式の場合、I2C\_CR2 レジスタの HEAD10R ビットをクリアすることによって、完全な読み出しシーケンスを送信することができます。この場合、マスタは、START ビットがセットされた後、次のような完全なシーケンスを自動的に送信します：(Re) START + スレーブアドレス 10 ビットヘッダ書き込み + スレーブアドレスの 2 番目のバイト + RStart + スレーブアドレス 10 ビットヘッダ読み出し。

図 298. HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス



- マスタが 10 ビットアドレススレーブをアドレス指定して、このスレーブアドレスにデータを送信した後、同じスレーブからデータを読み出す場合には、まず、マスタ送信フローが行われる必要があります。その場合、反復開始が、HEAD10R=1 で設定された 10 ビットスレーブアドレスでセットされます。この場合、マスタは次のシーケンスを送信します：ReStart + スレーブアドレス 10 ビットヘッダ読み出し

図 299. HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス



## マスタトランスミッタ

書き込み転送の場合、ACK が受信されたときの9 番目の SCL パルス後、各バイトの送信後にTXIS フラグがセットされます。

I2C\_CR1 レジスタの TXIE ビットがセットされている場合、TXIS イベント時に割り込みが生成されます。このフラグは、I2C\_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

転送中の TXIS イベントの数は、NBYTES[7:0] でプログラムされた値に対応します。送信されるデータバイト数の合計が 255 より大きい場合、I2C\_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

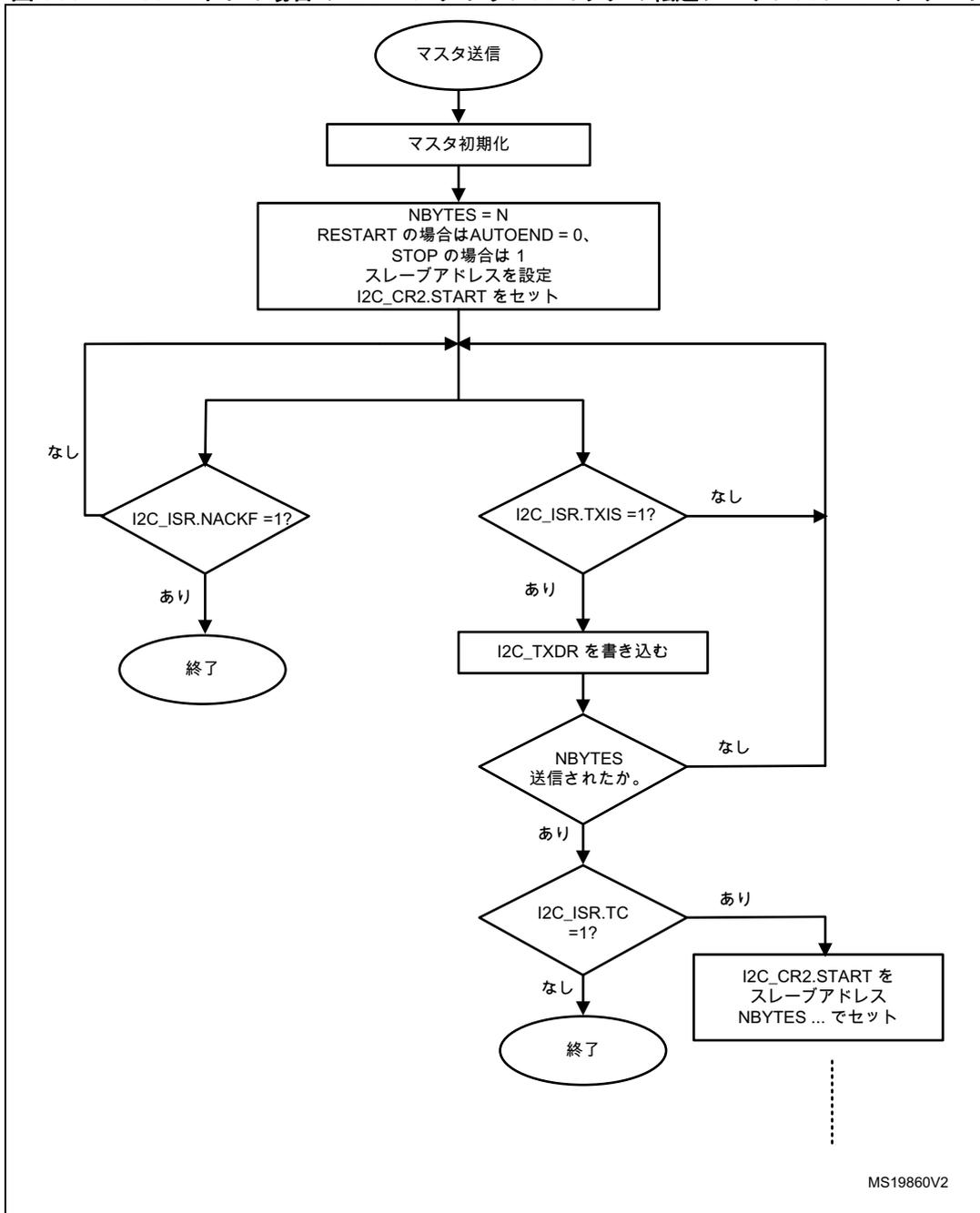
TXIS フラグは、NACK 受信時にはセットされません。

- RELOAD=0 でNBYTES データが転送されたとき：
  - － 自動終了モード (AUTOEND=1) では、STOP が自動的に送信されます。
  - － ソフトウェア終了モード (AUTOEND=0) では、TC フラグがセットされ、ソフトウェアアクションを実行するために SCL ラインがローでストレッチされます：
 

正しいスレーブアドレス設定と転送バイト数で I2C\_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションがバスに送信されます。

I2C\_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。
- NACK が受信された場合：TXIS フラグはセットされず、NACK 受信後、自動的に STOP コンディションが送信されます。I2C\_ISR レジスタの NACKF フラグがセットされ、NACKIE ビットがセットされていた場合は割り込みが生成されます。

図 300. N≤255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート

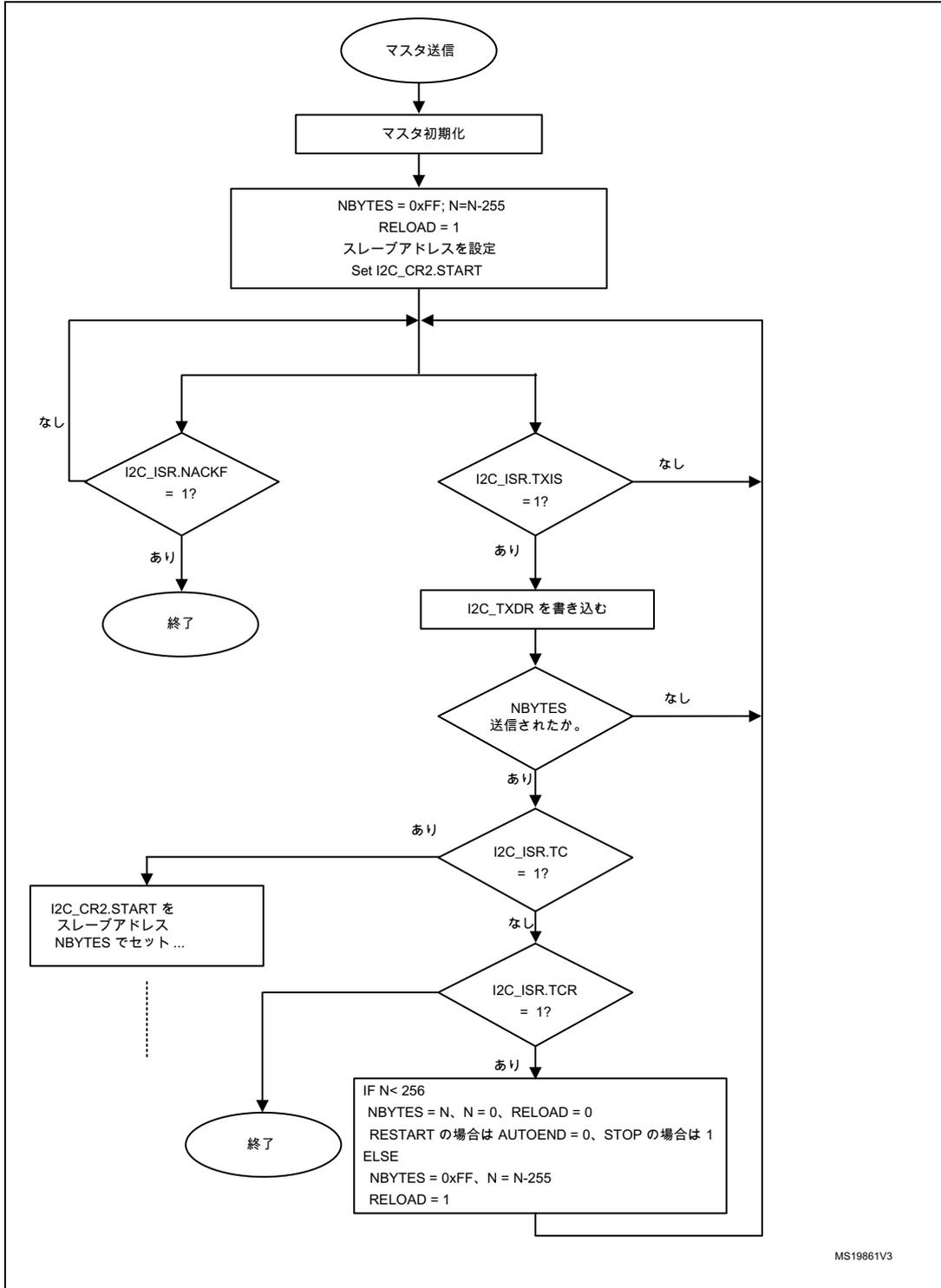


MS19860V2



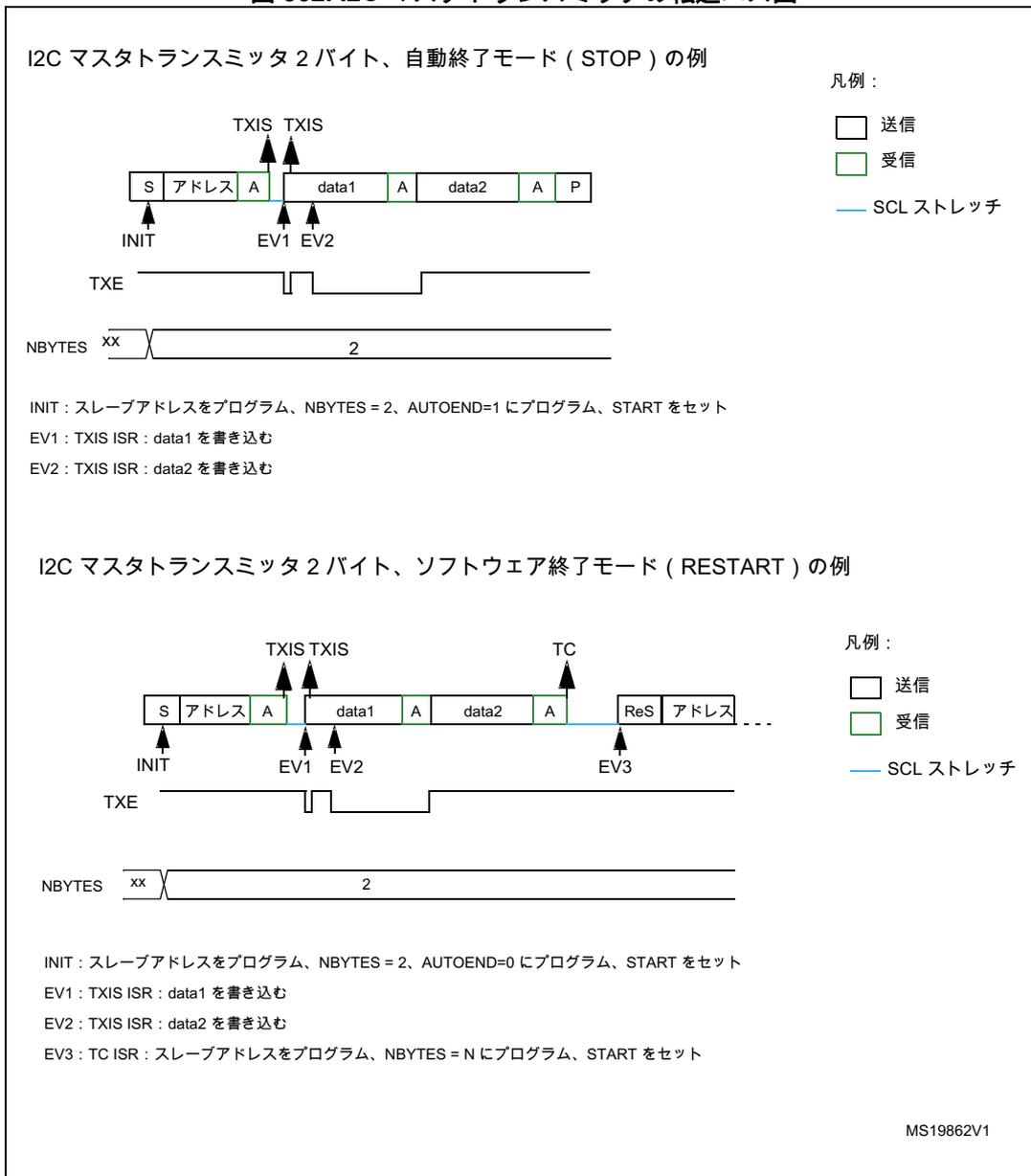
この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス社が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス社は、現行の正規英語版資料の更新に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

図 301. N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート



MS19861V3

図 302. I2C マスタトランスミッタの転送バス図



### マスタレシーバ

読み出し転送の場合、各バイトの受信後や8番目の SCL パルス後に RXNE フラグがセットされます。I2C\_CR1 レジスタの RXIE ビットがセットされている場合、RXNE イベント時に割り込みが生成されます。このフラグは、I2C\_RXDR が読み出されたときにクリアされます。

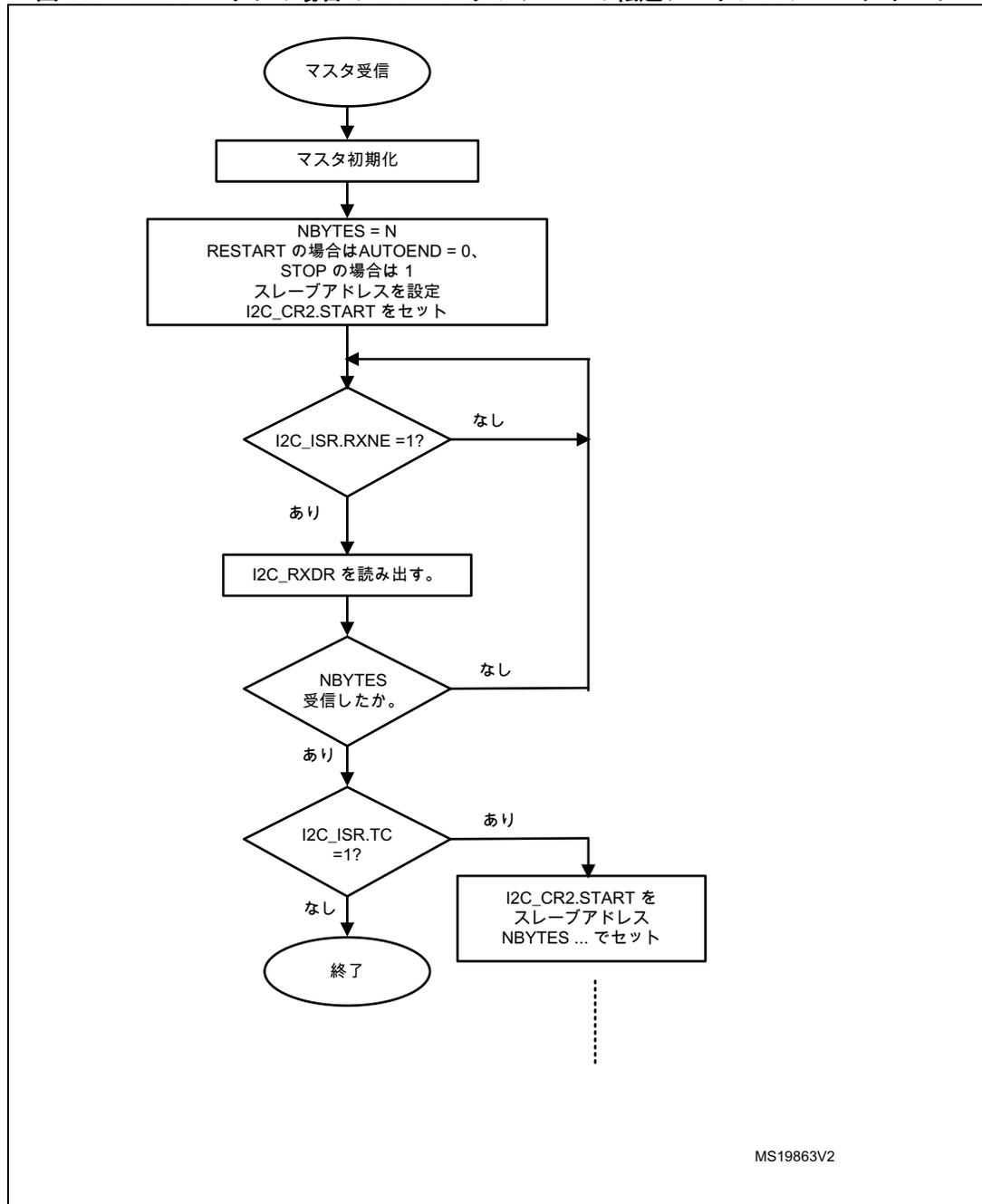
受信されるデータバイト数の合計が 255 より大きい場合、I2C\_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES[7:0] データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

- RELOAD=0 で NBYTES[7:0] データが転送されたとき：
  - 自動終了モード (AUTOEND=1) では、最後の受信バイト後に NACK および STOP が自動的に送信されます。
  - ソフトウェア終了モード (AUTOEND=0) では、最後の受信バイト後に NACK が自動的に送信され、TC フラグがセットされ、ソフトウェアアクションを実行できるように、SCL ラインがローでストレッチされます。

正しいスレーブアドレス設定と転送バイト数で I2C\_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションとスレーブアドレスがバスに送信されます。

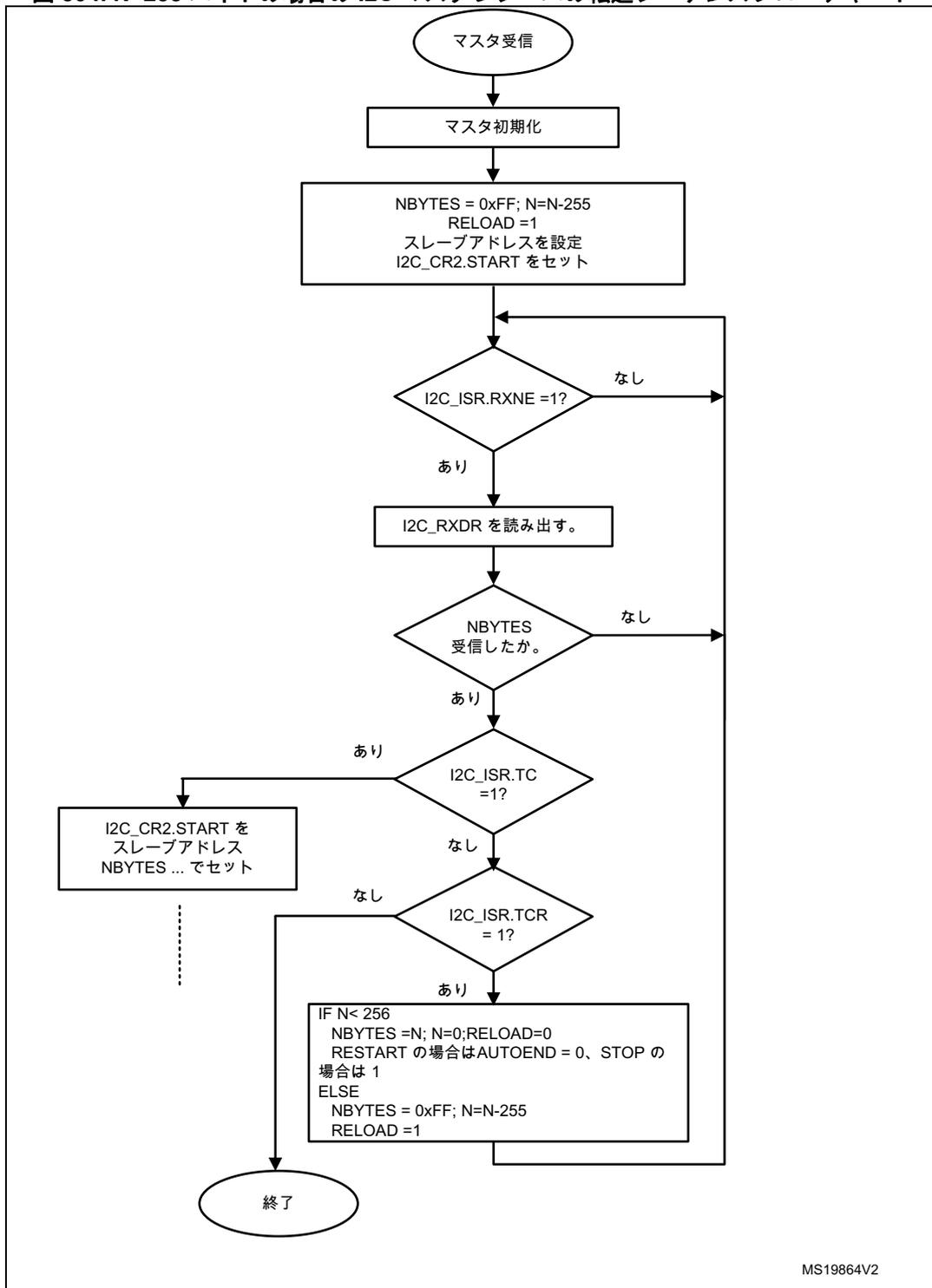
I2C\_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。

図 303. N ≤ 255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート



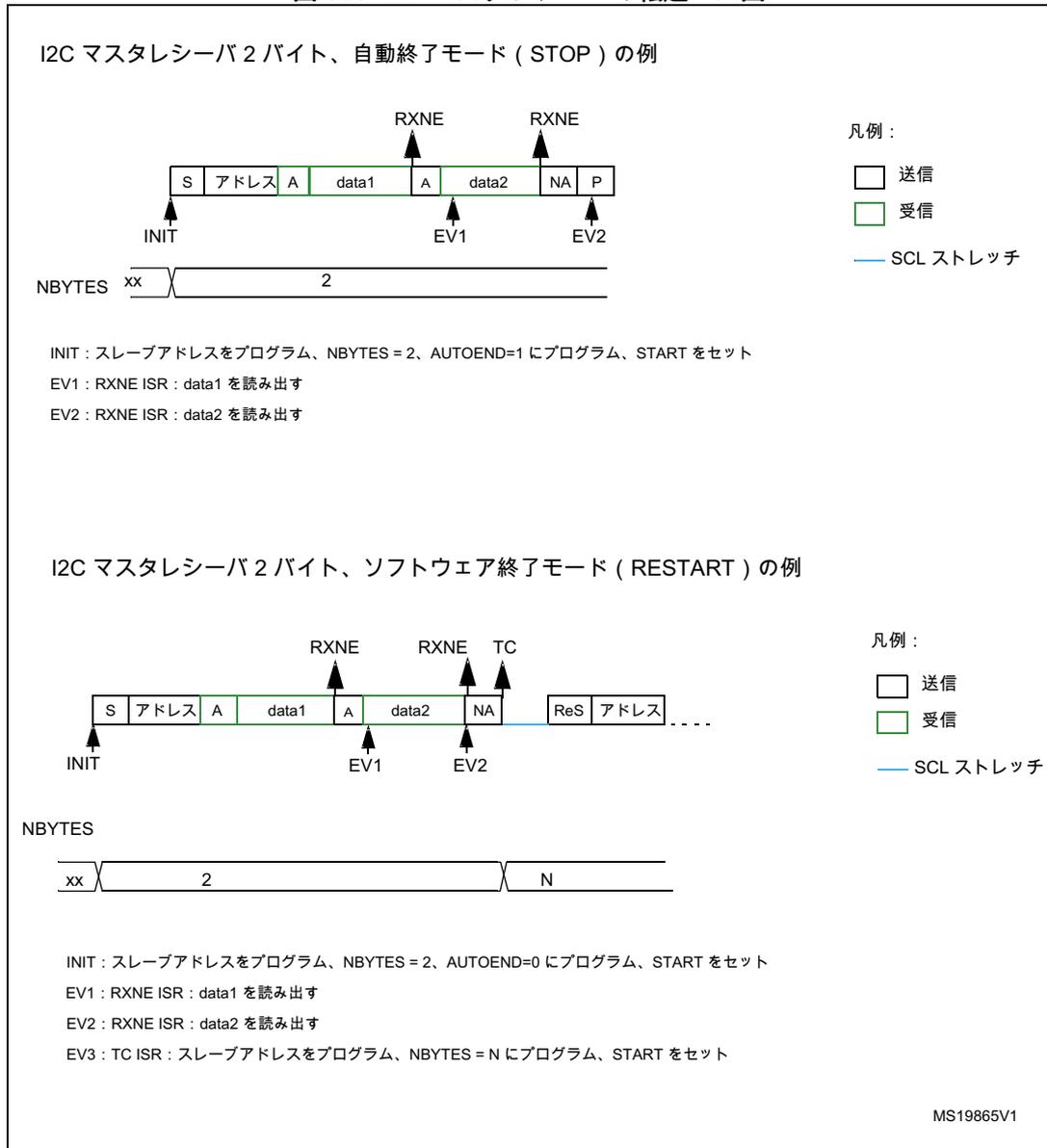
MS19863V2

図 304. N>255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート



MS19864V2

図 305. I2C マスタレシーバの転送バス図



## 30.4.9 I2C\_TIMINGR レジスタの設定例

下の表に、I<sup>2</sup>C 仕様準拠したタイミングを得るための I2C\_TIMINGR をプログラムする方法の例を示します。より正確な設定値を得るには、アプリケーションノート I<sup>2</sup>C タイミング設定ツール (AN4235) および該当するソフトウェア STSW-STM32126 を参照してください。

## 30.4.10 SMBus 固有の機能

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 30.3 : I2C の実装を参照してください。

### 概要

システム管理バス (SMBus) は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I<sup>2</sup>C の動作原理に基づきます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。

このペリフェラルは、SMBUS 仕様 rev 2.0 (<http://smbus.org>) と互換性があります。

システム管理バス仕様では、3 種類のデバイスを規定しています。

- スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。
- マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。
- ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ/スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

このペリフェラルは、マスタまたはスレーブデバイスとして、また、ホストとしても設定できます。

### SMBUS は、I<sup>2</sup>C 仕様 rev 2.1 に基づきます。

### バスプロトコル

特定のデバイスについて、11 の可能なコマンドプロトコルがあります。デバイスは、11 のプロトコルの一部または全部を使用して通信できます。プロトコルは、Quick Command、Send Byte、Receive Byte、Write Byte、Write Word、Read Byte、Read Word、Process Call、Block Read、Block Write、および Block Write-Block Read Process Call です。これらのプロトコルは、ユーザのソフトウェアによって実装してください。

これらのプロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

### アドレス解決プロトコル (ARP)

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス割り当てを目的とする各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子 (UDID) を実装する必要があります。128 ビットの番号がソフトウェアによって実装されます。

このペリフェラルは、アドレス解決プロトコル (ARP) をサポートします。SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C\_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。ARP コマンドは、ユーザのソフトウェアによって実装してください。

ARP サポートのために、スレーブモードでアービトレーションも行われます。

SMBus アドレス解決プロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

## 受信コマンドおよびデータ確認応答制御

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C\_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、セクション [スレーブバイト制御モード \(932 ページ\)](#) を参照してください。

## Host Notify プロトコル

このペリフェラルは、I2C\_CR1 レジスタの SMBHEN ビットをセットすることによって、Host Notify (ホスト通知) プロトコルをサポートします。この場合、ホストは SMBus ホストアドレス (0b0001 000) を確認応答します。

このプロトコルが使用されると、デバイスはマスタとして動作し、ホストはスレーブとして動作します。

## SMBus アラート

SMBus ALERT オプション信号がサポートされます。スレーブ専用デバイスは、通信したいホストの SMBALERT# ピンを通じてホストに信号を送信します。ホストは、割り込みを処理し、アラート応答アドレス (0b0001 100) を通じて全 SMBALERT# デバイスに同時にアクセスします。SMBALERT# をローに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。

スレーブデバイスとして設定されたとき (SMBHEN=0)、I2C\_CR1 レジスタの ALERTEN ビットをセットすることによって、SMBA ピンはローに引き下げられます。同時に、アラート応答アドレスが有効になります。

ホストとして設定されたとき (SMBHEN=1)、SMBA ピンで立ち下がりエッジが検出され、ALERTEN=1 のとき、I2C\_ISR レジスタの ALERT フラグがセットされます。I2C\_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。ALERTEN=0 のときには、外部 SMBA ピンがローの場合でも、ALERT ラインはハイとみなされます。

SMBus ALERT ピンが不要な場合には、ALERTEN=0 の場合、SMBA ピンを標準 GPIO として使用できません。

## パケットエラーチェック

信頼性と通信の堅牢性を向上させるために、SMBus 仕様にパケットエラーチェックメカニズムが導入されました。パケットエラーチェックは、各メッセージ転送の終わりにパケットエラーコード (PEC) を付加することによって実装されます。PEC は、すべてのメッセージバイト (アドレスと読み出し/書き込みビットを含む) に対して  $C(x) = x_8 + x^2 + x + 1$  CRC-8 多項式を使用して計算されます。

ペリフェラルはハードウェア PEC 計算機が組み込まれ、受信バイトがハードウェアによって計算された PEC に一致しないときには自動的に非確認応答を送信できます。

## タイムアウト

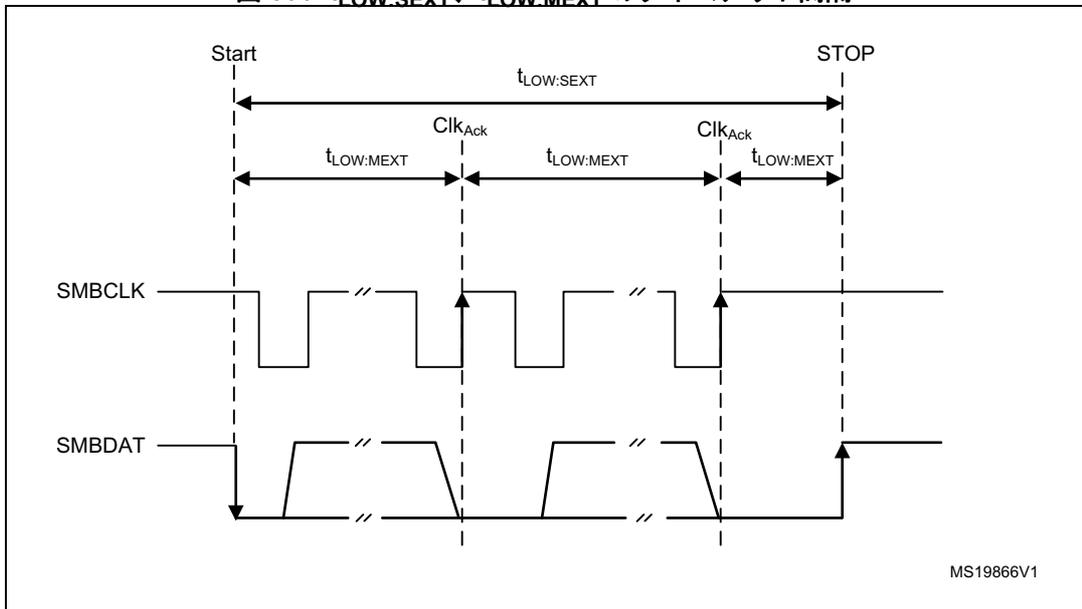
このペリフェラルは、SMBus 仕様バージョン 2.0 で定義された 3 つのタイムアウトに準拠するために、ハードウェアタイマが組み込まれています。

表 157. SMBus タイムアウト仕様

記号	パラメータ	リミット		単位
		最小値	最大値	
$t_{\text{TIMEOUT}}$	クロックロータイムアウト検出	25	35	ms
$t_{\text{LOW:SEXT}}^{(1)}$	累積クロックロー延長時間 (スレーブデバイス)	-	25	ms
$t_{\text{LOW:MEXT}}^{(2)}$	累積クロックロー延長時間 (マスタデバイス)	-	10	ms

- $t_{\text{LOW:SEXT}}$  は、特定のスレーブデバイスが初めての START から STOP までの 1 つのメッセージのクロックサイクルを延長できる累積時間です。別のスレーブデバイスまたはマスタもクロックを延長して、合計のクロックロー延長時間が  $t_{\text{LOW:SEXT}}$  より大きくなる場合があります。したがって、このパラメータは、スレーブデバイスをフルスピードのマスタの単独のターゲットとして測定されます。
- $t_{\text{LOW:MEXT}}$  は、マスタがメッセージの各バイト内のクロックサイクルを START-to-ACK、ACK-to-ACK、または ACK-to-STOP から定義に従って延長できる累積時間です。スレーブデバイスまたは別のマスタもクロックを延長して、合計のクロックロー時間が  $t_{\text{LOW:MEXT}}$  より大きくなる場合があります。したがって、このパラメータは、フルスピードスレーブデバイスをマスタの単独のターゲットとして測定されます。

図 306.  $t_{\text{LOW:SEXT}}$ 、 $t_{\text{LOW:MEXT}}$  のタイムアウト間隔



MS19866V1

## バスアイドル検出

マスタは、クロックおよびデータ信号が  $t_{\text{HIGH,MAX}}$  より大きい  $t_{\text{IDLE}}$  の間ハイであった場合、バスはフリーであるとみなすことができます (表 156: 12C-SMBUS 仕様のクロックタイミングを参照してください)

このタイミングパラメータは、マスタがバスに動的に追加し、SMBCLK または SMBDAT ラインで状態遷移を検出できなかった可能性のあるコンディションをカバーします。この場合、マスタは十分に長い時間待って、転送が進行中でないことを確認する必要があります。このペリフェラルは、ハードウェアバスアイドル検出をサポートします。

## 30.4.11 SMBus 初期化

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

SMBus 通信を行うためには、I2C 初期化に加えて、他にも特定の初期化を行う必要があります。

### 受信コマンドおよびデータ確認応答制御 (スレーブモード)

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C\_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、 [スレーブバイト制御モード \(932 ページ\)](#) を参照してください。

### 特定アドレス (スレーブモード)

必要な場合は、特定の SMBus アドレスを有効にしてください。詳細については、 [バスアイドル検出 \(954 ページ\)](#) を参照してください。

- SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C\_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。
- SMBus ホストアドレス (0b0001 000) は、I2C\_CR1 レジスタの SMBHEN ビットをセットすることによって有効になります。
- アラート応答アドレス (0b0001100) は、I2C\_CR1 レジスタの ALERTEN ビットをセットすることで有効になります。

### パケットエラーチェック

PEC 計算を有効にするには、I2C\_CR1 レジスタの PECEN ビットをセットします。その場合、PEC 転送はハードウェアバイトカウンタ (I2C\_CR2 レジスタの NBYTES[7:0]) を使用して管理されます。PECEN ビットは、I2C を有効にする前に設定する必要があります。

PEC 転送はハードウェアバイトカウンタによって管理されるので、スレーブモードで SMBus とインタフェースするときには SBC ビットをセットする必要があります。PEC は、PECBYTE ビットがセットされ、RELOAD ビットがクリアされたとき、NBYTES-1 データの転送後に転送されます。RELOAD がセットされた場合、PECBYTE は効果がありません。

**注意 :** I2C が有効なときには、PECEN 設定の変更はできません。

表 158. SMBUS の PEC 設定

モード	SBC ビット	RELOAD ビット	AUTOEND ビット	PECBYTE ビット
マスタ Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
マスタ Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
スレーブ Tx/Rx と PEC	1	0	x	1

## タイムアウト検出

タイムアウト検出は、I2C\_TIMEOUTR レジスタの TIMOUTEN および TEXTEN ビットをセットすることによって有効になります。SMBus 仕様バージョン 2.0 で指定された最大時間の前にタイムアウトを検出するようにタイマをプログラムする必要があります。

- $t_{\text{TIMEOUT}}$  チェック

$t_{\text{TIMEOUT}}$  チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] ビットを  $t_{\text{TIMEOUT}}$  パラメータをチェックするためにタイマ再ロード値でプログラムする必要があります。SCL ローレベルタイムアウトを検出するためには、TIDLE ビットを 0 に設定する必要があります。

その場合、タイマは、I2C\_TIMEOUTR レジスタの TIMOUTEN をセットすることによって有効になります。

SCL が  $(\text{TIMEOUTA}+1) \times 2048 \times t_{\text{I2CCLK}}$  より長い時間、ローに設定された場合、I2C\_ISR レジスタの TIMEOUT フラグがセットされます。

[表 159: さまざまな I2CCLK 周波数での TIMEOUTA の設定例 \(最大値  \$t\_{\text{TIMEOUT}} = 25 \text{ ms}\$ \)](#) を参照してください。

**注意:** **TIMEOUTEN ビットがセットされているときには、TIMEOUTA[11:0] ビットおよび TIDLE ビットの設定変更はできません。**

- $t_{\text{LOW:SEXT}}$  および  $t_{\text{LOW:MEXT}}$  チェック

ペリフェラルがマスタとして設定されているか、スレーブとして設定されているかに応じて、12 ビットの TIMEOUTB タイマは、スレーブの場合は  $t_{\text{LOW:SEXT}}$  をチェックするために、マスタの場合は  $t_{\text{LOW:MEXT}}$  をチェックするために、設定する必要があります。標準では最大値のみが規定されているので、両方について同じ値を選ぶことができます。

その場合、タイマは、I2C\_TIMEOUTR レジスタの TEXTEN ビットをセットすることによって有効になります。

SMBus ペリフェラルが、 $(\text{TIMEOUTB}+1) \times 2048 \times t_{\text{I2CCLK}}$  より長い時間および [バスアイドル検出 \(954 ページ\)](#) セクションで述べられているタイムアウト間隔で、累積 SCL ストレッチを実行した場合、I2C\_ISR レジスタの TIMEOUT フラグがセットされます。

[表 160: さまざまな I2CCLK 周波数での TIMEOUTB の設定例](#) を参照してください。

**注意:** **TEXTEN ビットがセットされているときには、TIMEOUTB 設定変更はできません。**

## バスアイドル検出

$t_{\text{IDLE}}$  チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] フィールドを  $t_{\text{IDLE}}$  パラメータを得るためにタイマ再ロード値でプログラムする必要があります。SCL および SDA ハイレベルタイムアウトを検出するためには、TIDLE ビットを 1 に設定する必要があります。

その場合、タイマは、I2C\_TIMEOUTR レジスタの TIMEOUTEN ビットをセットすることによって有効になります。

SCL および SDA の両方のラインが  $(\text{TIMEOUTA}+1) \times 4 \times t_{\text{I2CCLK}}$  より長い間ハイのままであった場合、I2C\_ISR レジスタの TIMEOUT フラグがセットされます。

[表 161: さまざまな I2CCLK 周波数での TIMEOUTA の設定例 \(最大値  \$t\_{\text{IDLE}} = 50 \text{ ms}\$ \)](#) を参照してください。

**注意:** **TIMEOUTEN がセットされているときに、TIMEOUTA および TIDLE 設定を変更することはできません。**

## 30.4.12 SMBus : I2C\_TIMEOUTR レジスタの設定例

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

- $t_{\text{TIMEOUT}}$  の最大時間を 25 ms に設定 :

**表 159. さまざまな I2CCLK 周波数での TIMEOUTA の設定例**  
(最大値  $t_{\text{TIMEOUT}} = 25$  ms)

$f_{\text{I2CCLK}}$	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	$t_{\text{TIMEOUT}}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms

- $t_{\text{LOW:SEXT}}$  および  $t_{\text{LOW:MEXT}}$  の最大時間を 8 ms に設定 :

**表 160. さまざまな I2CCLK 周波数での TIMEOUTB の設定例**

$f_{\text{I2CCLK}}$	TIMEOUTB[11:0] ビット	TEXTEN ビット	$t_{\text{LOW:EXT}}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms

- $t_{\text{IDLE}}$  の最大時間を 50  $\mu$ s に設定

**表 161. さまざまな I2CCLK 周波数での TIMEOUTA の設定例**  
(最大値  $t_{\text{IDLE}} = 50$  ms)

$f_{\text{I2CCLK}}$	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	$t_{\text{IDLE}}$
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 $\mu$ s
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 $\mu$ s

## 30.4.13 SMBus スレープモード

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

2C スレープ転送管理 ([セクション 30.4.7 : I2C スレープモード](#) を参照) に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

### SMBus スレープトランスミッタ

IP が SMBus で使用されるときには、SBCは、プログラムされたデータバイト数の終わりの PEC 送信を可能にするため、1 にプログラムする必要があります。PECBYTE ビットがセットされているときには、NBYTES[7:0] でプログラムされたバイト数には PEC 送信が含まれます。その場合、TXIS 割り込みの合計数は NBYTES-1 であり、NBYTES-1 データ転送後にマスタが追加のバイトをリクエストした場合、I2C\_PECR レジスタの内容が自動的に送信されます。

**注意 :** **PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。**



図 307. N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート

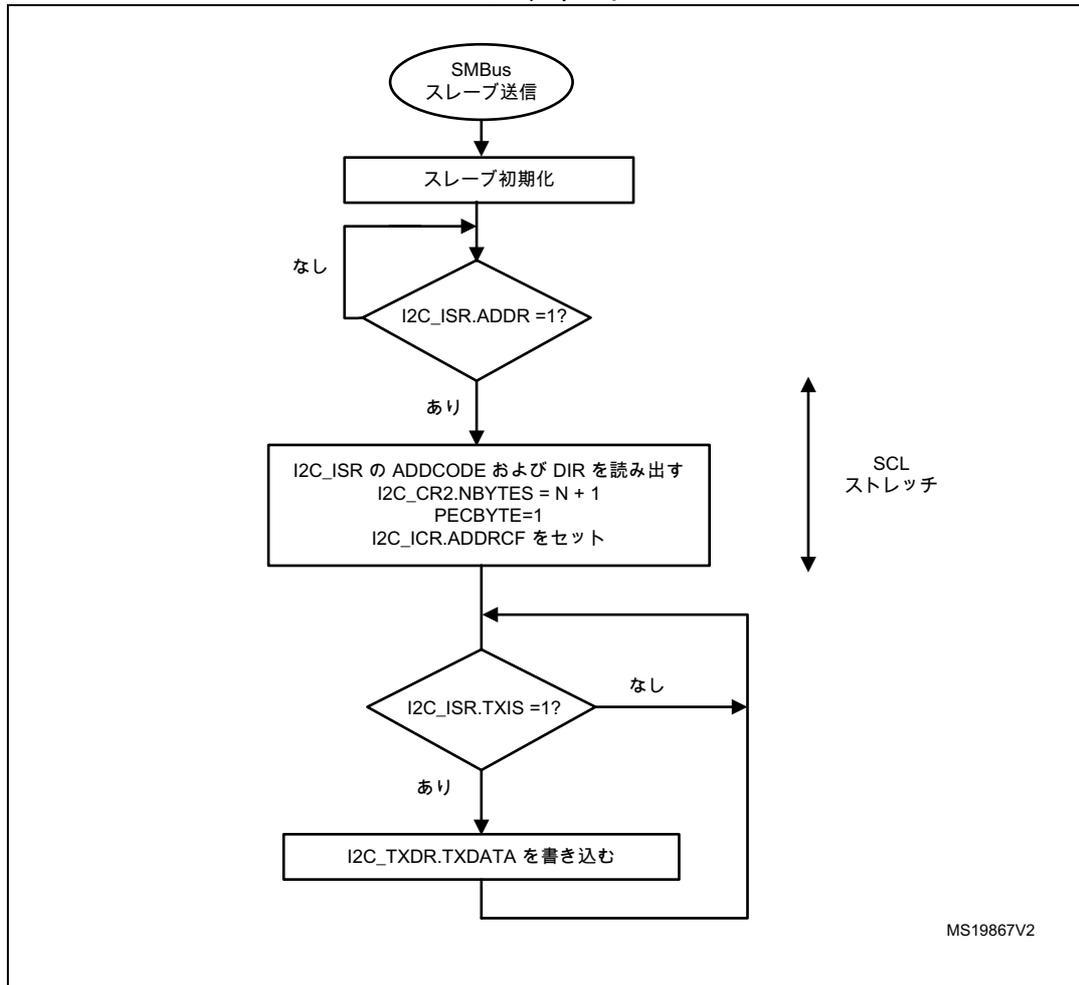
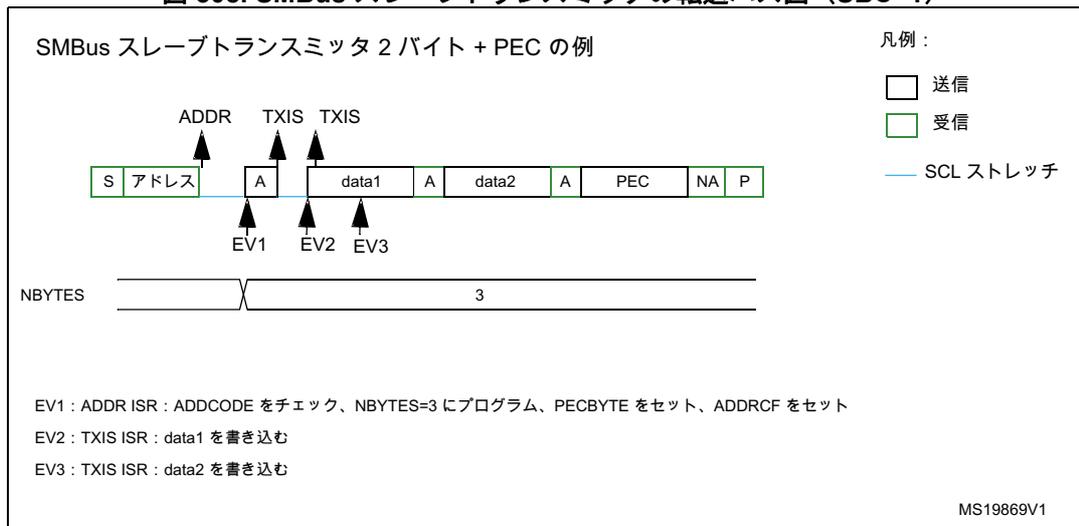


図 308. SMBus スレーブトランスミッタの転送バス図 (SBC=1)



## SMBus スレーブレシーバ

I2C が SMBus モードで使用される際には、SBC は、プログラムされたデータバイト数の終わりの PEC チェックを可能にするため、1 にプログラムする必要があります。各バイトの ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD=1)。詳細については、[スレーブバイト制御モード \(932 ページ\)](#) を参照してください。

PEC バイトをチェックするためには、RELOAD ビットをクリアして、PECBYTE ビットをセットする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが内部 I2C\_PECR レジスタの内容と比較されます。ACK ビットの値にかかわらず、比較が一致しなかった場合は NACK が自動的に生成され、比較が一致した場合は ACK が自動的に生成されます。PEC バイトが受信されると、他のデータと同様に I2C\_RXDR レジスタにコピーされ、RXNE フラグがセットされます。

PEC 不一致の場合、PECERR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

ACK ソフトウェア制御が不要な場合は、PECBYTE=1 をプログラムし、同じ書き込み操作で NBYTES を連続フローで受信するバイト数にプログラムします。NBYTES-1 が受信された後、次の受信バイトが PEC であるかどうかチェックされます。

**注意：** *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 309. N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート

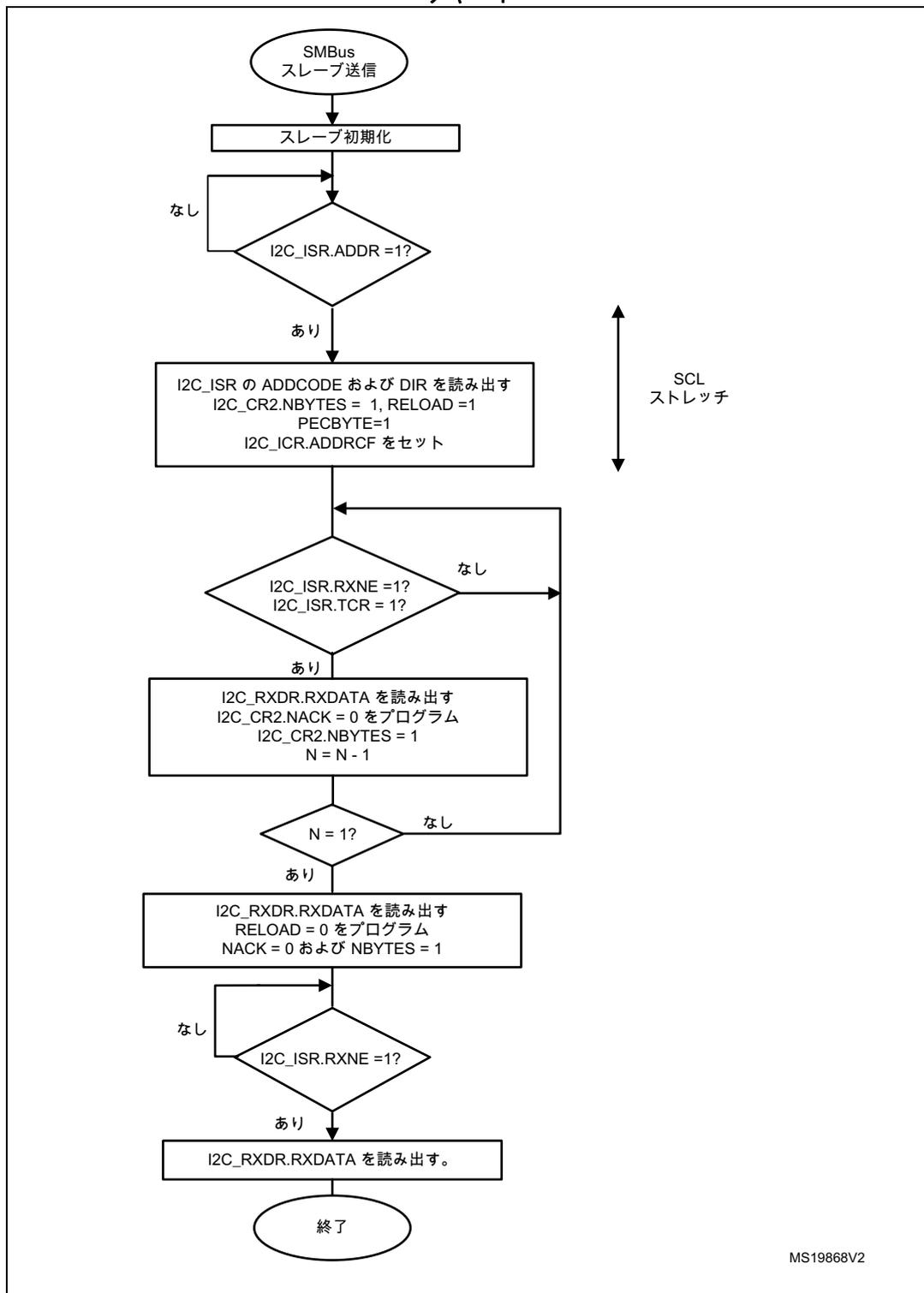
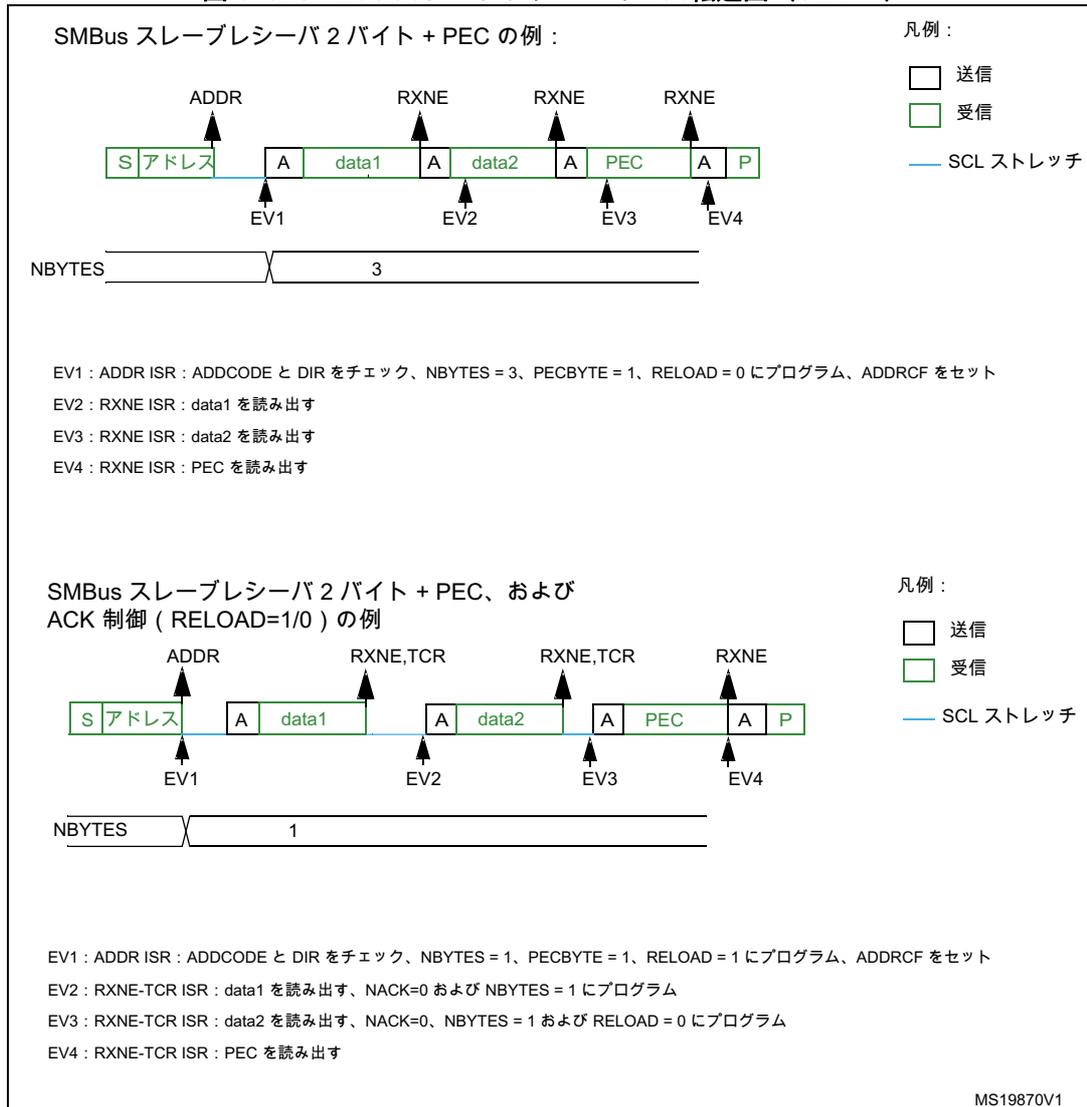


図 310. SMBus スレーブレシーバのバス転送図 (SBC=1)



このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

I2C マスタ転送管理 ([セクション 30.4.8 : I2C マスタモード](#) を参照) に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

## SMBus マスタトランスミッタ

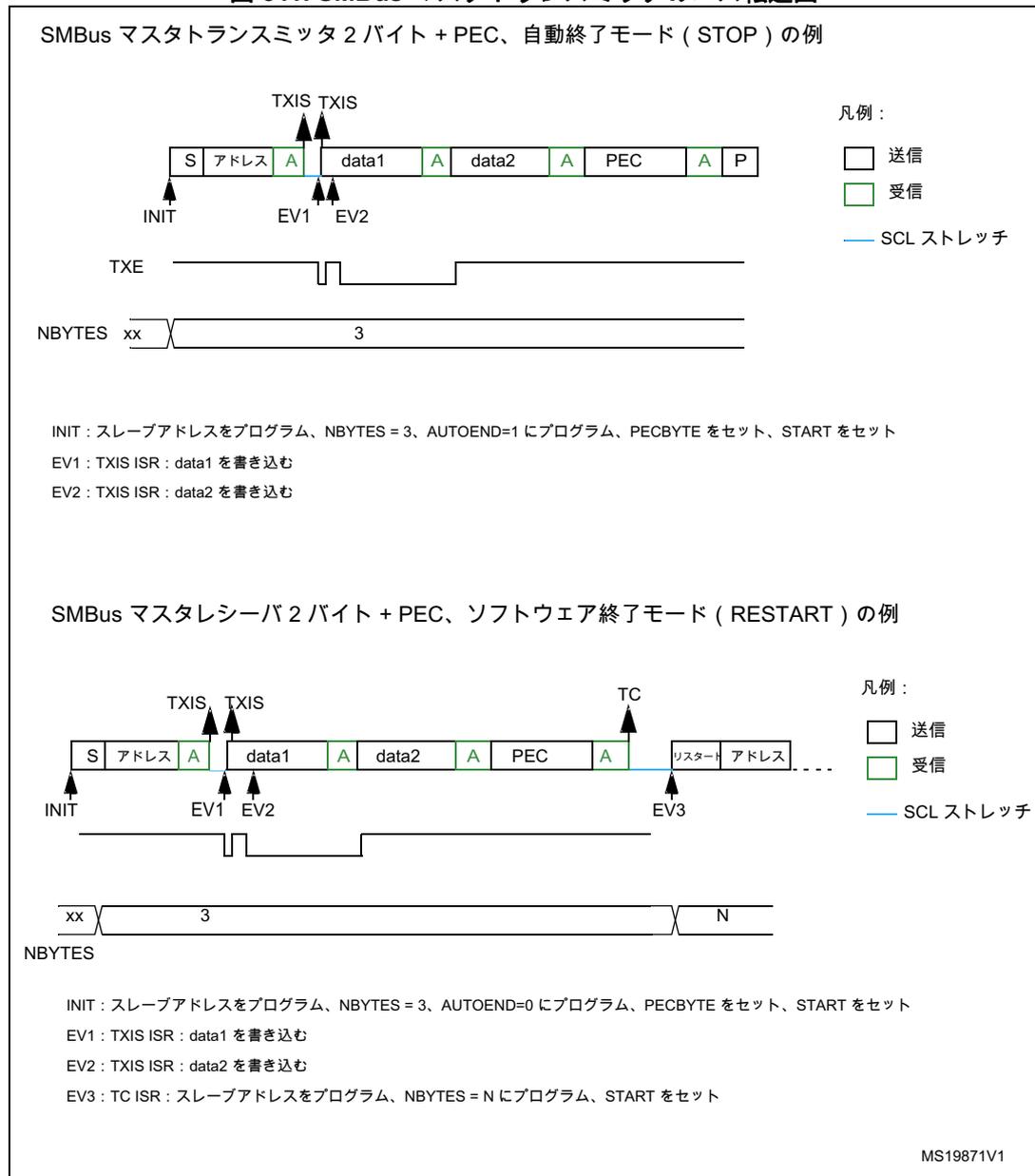
SMBus マスタが PEC を送信したいときには、START ビットをセットする前に、PECBYTE ビットをセットする必要があり、バイト数を NBYTES[7:0] フィールドでプログラムする必要があります。この場合、TXIS 割り込みの合計数は NBYTES-1 になります。したがって、NBYTES=0x1 のときに PECBYTE ビットがセットされた場合、I2C\_PECR レジスタの内容が自動的に送信されます。

SMBus マスタが PEC 後に STOP コンディションを送信したい場合は、自動終了モードを選択してください (AUTOEND=1)。この場合、PEC 送信に続いて、STOP コンディションが自動的に送信されます。

SMBus マスタが PEC 後に RESTART コンディションを送信したい場合は、ソフトウェアモードを選択してください (AUTOEND=0)。この場合、NBYTES-1 が送信されると、PEC 送信後に I2C\_PECR レジスタの内容が送信され、TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションを TC 割り込みサブルーチンでプログラムする必要があります。

**注意：** PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 311. SMBus マスタトランスミッタのバス転送図



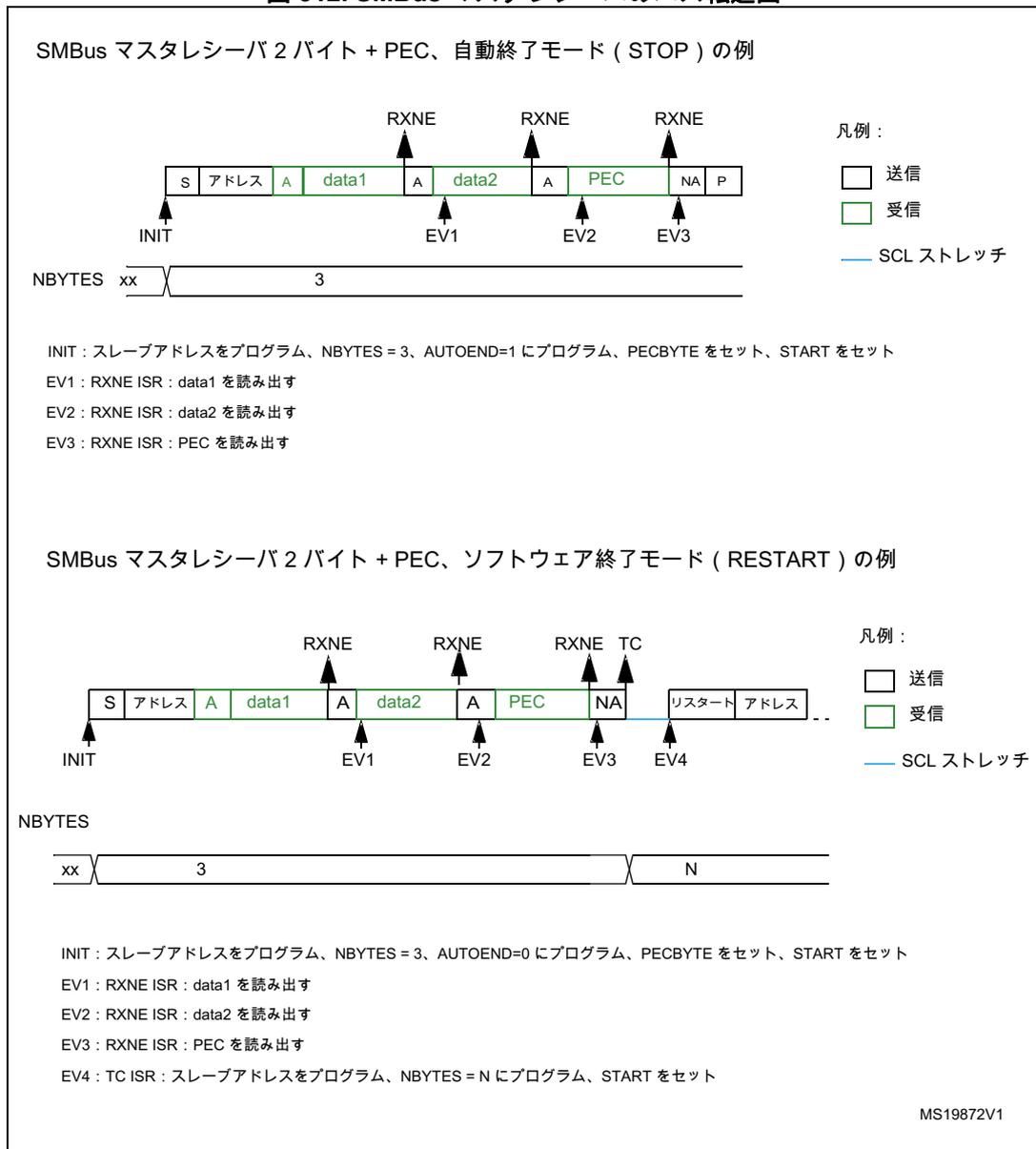
## SMBus マスタレシーバ

SMBus マスタが転送終了時に PEC を受信してから STOP を受信したいときには、自動終了モードを選択できます (AUTOEND=1)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが I2C\_PECR レジスタの内容と自動的に照合されます。PEC バイトに対して NACK 応答が与えられた後、STOP コンディションが送信されます。

SMBus マスタが転送終了時に PEC バイトを受信してから RESTART を受信したいときには、ソフトウェアモードを選択する必要があります (AUTOEND=0)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが I2C\_PECR レジスタの内容と自動的に照合されます。PEC バイト受信後に TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションは、TC 割り込みサブルーチンでプログラムできます。

**注意：** *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 312. SMBus マスタレシーバのバス転送図



## 30.4.14 エラー条件

以下は、通信エラーを引き起こす可能性のあるエラー条件です。

### バスエラー (BERR)

バスエラーは、START または STOP コンディションが検出され、複数の 9 SCL クロックパルス後になかったときに検出されます。START または STOP コンディションは、SCL がハイとのときに SDA エッジが発生した場合に検出されます。

バスエラーフラグは、I2C がマスタまたはアドレス指定されたスレーブとして転送に関与する場合にのみ（すなわち、スレーブモードのアドレスフェーズでないとき）、セットされます。

スレーブモードで START または RESTART の誤配置が検出された場合、I2C は、正しい START コンディションの場合と同様に、アドレス認識状態に入ります。

バスエラーが検出されると、I2C\_ISR レジスタの BERR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

### アービトレーション喪失 (ARLO)

アービトレーション喪失は、SDA ラインでハイレベルが送信されたが、SCL 立ち上がりエッジでローレベルがサンプリングされたときに検出されます。

- マスタモードでは、アービトレーション喪失は、アドレスフェーズ、データフェーズ、およびデータ確認応答フェーズで検出されます。この場合、SDA および SCL ラインはリリースされ、START 制御ビットがハードウェアによってクリアされ、マスタは自動的にスレーブモードに切り替わります。
- スレーブモードでは、アービトレーション喪失は、データフェーズとデータ確認応答フェーズで検出されます。この場合、転送は中止され、SCL および SDA ラインがリリースされます。

アービトレーション喪失が検出されると、I2C\_ISR レジスタの ARLO フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

### オーバーラン/アンダーランエラー (OVR)

オーバーランまたはアンダーランエラーは、スレーブモードで NOSTRETCH=1 のとき、および次のときに検出されます：

- 受信時、新しいバイトが受信され、RXDR レジスタがまだ読み出されていないとき。新しい受信バイトは失われ、新しいバイトへの応答として NACK が自動的に送信されます。
- 送信時：
  - STOPF=1 のときには、最初のデータバイトが送信されなければなりません。TXE=0、0xFF の場合、I2C\_TXDR レジスタの内容が送信され、そうでない場合は送信されません。
  - 新しいバイトが送信されるべきときに、I2C\_TXDR レジスタがまだ書き込まれていなかった場合、0xFF が送信されます。

オーバーランまたはアンダーランエラーが検出されると、I2C\_ISR レジスタの OVR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

### パケットエラーチェックエラー (PECERR)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

PEC エラーは、受信した PEC バイトが I2C\_PECR レジスタの内容と一致しなかったときに検出されます。正しくない PEC の受信後、NACK が自動的に送信されます。

PEC エラーが検出されると、I2C\_ISR レジスタの PECERR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## タイムアウトエラー (TIMEOUT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

タイムアウトエラーは、次のような条件で発生します：

- TIDLE=0 であり、SCL が TIMEOUTA[11:0] ビットで定義された時間だけローのままであった場合：これは SMBus タイムアウトの検出に使用されます。
- TIDLE=1 であり、SDA および SCL が TIMEOUTA[11:0] ビットで定義された時間だけハイのままであった場合：これはバスアイドル状態の検出に使用されます。
- マスタ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合 (SMBus  $t_{LOW:MEXT}$  パラメータ)。
- スレーブ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合 (SMBus  $t_{LOW:SEXT}$  パラメータ)。

マスタモードでタイムアウト違反が検出されると、STOP コンディションが自動的に送信されます。

スレーブモードでタイムアウト違反が検出されると、SDA および SCL ラインが自動的にリリースされます。

タイムアウトエラーが検出されると、I2C\_ISR レジスタの TIMEOUT フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## アラート (ALERT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 30.3 : I2C の実装](#) を参照してください。

ALERT フラグは、I2C インタフェースがホストとして設定され (SMBHEN=1)、アラートピン検出が有効であり (ALERTEN=1)、SMBA ピンで立ち下がりエッジが検出されたときにセットされます。I2C\_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。

## 30.4.15 DMA リクエスト

### DMA を使用した送信

送信について DMA (Direct Memory Access) を有効にするには、I2C\_CR1 レジスタの TXDMAEN ビットをセットします。TXIS ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 8 : ダイレクトメモリアクセスコントローラ \(DMA\) \(221 ページ\)](#)) を参照) を使用して設定された SRAM 領域から I2C\_TXDR レジスタにロードされます。

データのみが DMA で転送されます。

- マスタモード：初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます (送信されたスレーブアドレスを DMA で転送することはできません)。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。 [マスタトランスミッタ \(944 ページ\)](#) を参照してください。
- スレーブモードでは：
  - NOSTRETCH=0 では、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
  - NOSTRETCH=1 では、アドレス一致イベントの前に DMA を初期化する必要があります。

- SMBus をサポートする場合 : PEC 転送は NBYTES カウンタによって管理されます。 [SMBus スレーブトランスミッタ \(957 ページ\)](#) および [SMBus マスタトランスミッタ \(961 ページ\)](#) を参照してください。

**注 :** DMA が送信に使用される場合、TXIE ビットが有効である必要はありません。

## DMA を使用した受信

受信について DMA (Direct Memory Access) を有効にするには、I2C\_CR1 レジスタの RXDMAEN ビットをセットします。RXNE ビットがセットされているときには、データは、I2C\_RXDR レジスタから DMA ペリフェラル ([セクション 8: ダイレクトメモリアクセスコントローラ \(DMA\)](#) を参照) を使用して設定された SRAM 領域にロードされます。データのみ (PEC を含む) が DMA で転送されます。

- マスタモード、初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。
- NOSTRETCH=0 のスレーブモードでは、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
- SMBus がサポートされる場合 ([セクション 30.3 : I2C の実装](#) を参照) : PEC 転送は NBYTES カウンタによって管理されます。 [SMBus スレーブレシーバ \(959 ページ\)](#) および [SMBus マスタレシーバ \(963 ページ\)](#) を参照してください。

**注 :** DMA が受信に使用される場合、RXIE ビットが有効である必要はありません。

## 30.4.16 デバッグモード

マイクロコントローラがデバッグモードに入ると (コア停止)、DBG モジュールの DBG\_I2Cx\_STOP 設定ビットに応じて、SMBus タイムアウトは、通常の動作を続行するか、あるいは停止します。

## 30.5 I2C 低電力モード

表 162. 低電力モード

モード	説明
SLEEP	影響なし。 I2C 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	I2C レジスタの内容は保たれます。
STANDBY	I2C ペリフェラルはパワーダウンされ、STANDBY 終了後に再初期化する必要があります。

## 30.6 I2C 割り込み

次の表に、I2C 割り込みリクエストの一覧を示します。

表 163. I2C 割り込みリクエスト

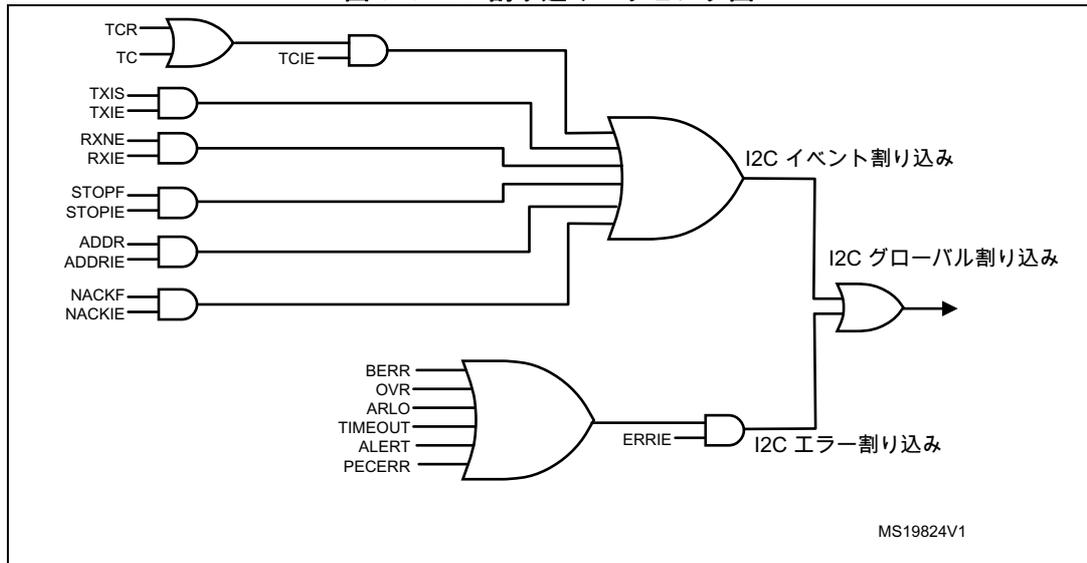
割り込みイベント	イベントフラグ	イベントフラグ/ 割り込みのクリア方法	割り込み有効制御 ビット
受信バッファノットエンプティ	RXNE	I2C_RXDR レジスタを 読み出す	RXIE
送信バッファ割り込みステータス	TXIS	I2C_TXDR レジスタに 書き込む	TXIE
STOP 検出割り込みフラグ	STOPF	STOPCF=1 を書き込む	STOPIE
転送完了再ロード	TCR	I2C_CR2 の NBYTES[7:0] <sup>1</sup> に 0 を 書き込む	TCIE
転送完了	TC	START=1 または STOP=1 を 書き込む	
アドレス一致	ADDR	ADDRCF=1 を書き込む	ADDRIE
NACK 受信	NACKF	NACKCF=1 を書き込む	NACKIE
バスエラー	BERR	BERRCF=1 を書き込む	ERRIE
アービトレーション喪失	ARLO	ARLOCF=1 を書き込む	
オーバーラン/アンダーラン	OVR	OVRCF=1 を書き込む	
PEC エラー	PECERR	PECERRCF=1 を書き込む	
タイムアウト/ <sub>t</sub> LOW エラー	TIMEOUT	TIMEOUTCF=1 を書き込む	
SMBus アラート	ALERT	ALERTCF=1 を書き込む	

製品実装に応じて、これらの割り込みイベントのすべてで同じ割り込みベクトルを共有するか (I2C グローバル割り込み)、2 つの割り込みベクトル (I2C イベント割り込みと I2C エラー割り込み) にグループ化することができます。詳細については、を参照してください。

I2C 割り込みを有効にするには、次のシーケンスが必要です。

1. NVIC で I2C IRQ チャンネルを設定し、有効にします。
2. 割り込みを生成するように I2C を設定します。

図 313. I2C 割り込みマッピング図



## 30.7 I2C レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

ペリフェラルレジスタは、ワード (32 ビット) 単位でアクセスされます。

### 30.7.1 制御レジスタ 1 (I2C\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	Res.	NOSTRETCH	SBC	
								rw	rw	rw	rw	rw		rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RXDMAEN	TXDMAEN	Res.	ANFOF	DNF				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE	
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **PECEN** : PEC イネーブル

0 : PEC 計算は無効です。

1 : PEC 計算は有効です。

**注 :** **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に0に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 22 **ALERTEN**:SMBus アラート有効

**デバイスモード (SMBHEN=0) :**

0 : SMBA ピンをハイにリリースし、アラート応答アドレスヘッダを無効にします。0001100x の後に NACK が続きます。

1 : SMBA ピンをローに駆動し、アラート応答アドレスヘッダを有効にします : 0001100x の後に ACK が続きます。

**ホストモード (SMBHEN=1) :**

0 : SMBus アラートピン (SMBA) はサポートされません。

1 : SMBus アラートピン (SMBA) はサポートされます。

**注 :** **ALERTEN=0 のときには、SMBA ピンを標準 GPIO として使用できます。**

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に0に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 21 **SMBDEN** : SMBus デバイスデフォルトアドレス有効

0 : デバイスデフォルトアドレス無効。アドレス 0b1100001x は NACK されます。

1 : デバイスデフォルトアドレス有効。アドレス 0b1100001x は ACK されます。

**注 :** **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に0に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 20 **SMBHEN** : SMBus ホストアドレス有効

0 : ホストアドレス無効。アドレス 0b0001000x は NACK されます。

1 : ホストアドレス有効。アドレス 0b0001000x は ACK されます。

**注 :** **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に0に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 19 **GCEN** : 同報イネーブル

0 : 同報は無効です。アドレス 0b00000000 は NACK されます。

1 : 同報は有効です。アドレス 0b00000000 は ACK されます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **NOSTRETCH** : クロックストレッチ無効

このビットは、スレーブモードでのクロックストレッチを無効にするために使用されます。マスタモードではクリアされたままでなければなりません。

0 : クロックストレッチ有効

1 : クロックストレッチ無効

**注 :** **このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。**

ビット 16 **SBC** : スレーブバイト制御

このビットは、スレーブモードでのハードウェアバイト制御を有効にするために使用されます。

0 : スレーブバイト制御無効

1 : スレーブバイト制御有効

ビット 15 **RXDMAEN** : DMA 受信リクエスト有効

0 : DMA モードは受信に無効

1 : DMA モードは受信に有効

ビット 14 **TXDMAEN** : DMA 送信リクエスト有効

0 : DMA モードは送信に無効

1 : DMA モードは送信に有効

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **ANFOFF** : アナログノイズフィルタ OFF

0 : アナログノイズフィルタ有効

1 : アナログノイズフィルタ無効

**注 :** このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。

ビット 11:8 **DNF[3:0]** : デジタルノイズフィルタ

これらのビットは、SDA および SCL 入力のデジタルノイズフィルタを設定するために使用されます。デジタルフィルタは、最大  $DNF[3:0] * t_{I2CCCLK}$  の長さのスパイクを除去します。

0000 : デジタルフィルタ無効

0001 : デジタルフィルタは有効であり、最大  $1 t_{I2CCCLK}$  の除去能力を持ちます。

...

1111 : デジタルフィルタは有効であり、最大  $15 t_{I2CCCLK}$  の除去能力を持ちます。

**注 :** アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。

このフィルタは、I2C が無効 (PE = 0) のときのみプログラムできます。

ビット 7 **ERRIE** : エラー割り込み有効

0 : エラー検出割り込み無効

1 : エラー検出割り込み有効

**注 :** 次のようなエラーが発生すると、割り込みが生成されます :

アービトレーション喪失 (ARLO)

バスエラー検出 (BERR)

オーバーラン/アンダーラン (OVR)

タイムアウト検出 (TIMEOUT)

PEC エラー検出 (PECERR)

アラートピンイベント検出 (ALERT)

ビット 6 **TCIE** : 転送完了割り込み有効

0 : 転送完了割り込み無効

1 : 転送完了割り込み有効

**注 :** 次のようなイベントが発生すると、割り込みが生成されます :

転送完了 (TC)

転送完了再ロード (TCR)

ビット 5 **STOPIE** : STOP 検出割り込み有効

0 : STOP 検出 (STOPF) 割り込み無効

1 : STOP 検出 (STOPF) 割り込み有効

ビット 4 **NACKIE** : 非確認応答受信割り込み有効

0 : 非確認応答 (NACKF) 受信割り込み無効

1 : 非確認応答 (NACKF) 受信割り込み有効

ビット 3 **ADDRIE** : アドレス一致割り込み有効 (スレーブのみ)

0 : アドレス一致 (ADDR) 割り込み無効

1 : アドレス一致 (ADDR) 割り込み有効

ビット 2 **RXIE** : RX 割り込み有効

0 : 受信 (RXNE) 割り込み無効

1 : 受信 (RXNE) 割り込み有効

ビット 1 **TXIE** : TX 割り込み有効

0 : 送信 (TXIS) 割り込み無効

1 : 送信 (TXIS) 割り込み有効

ビット 0 PE : ペリフェラルは有効です。

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

**注 :** PE=0 のとき、I2C SCL および SDA ラインはリリースされます。内部ステートマシンおよびスタートビットはリセット値に戻されます。クリアされたときには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たれる必要があります。

## 30.7.2 制御レジスタ 2 (I2C\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTOE ND	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD1 OR	ADD10	RD_WR N	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw									

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **PECBYTE** : パケットエラーチェックバイト

このビットはソフトウェアによってセットされ、PEC が転送されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、また、PE=0 のとき、ハードウェアによってクリアされます。

0 : PEC 転送なし。

1 : PEC 送信/受信がリクエストされます。

**注 :** このビットに 0 を書き込んでも、効果はありません。

このビットは、RELOAD がセットされているときには効果がありません。

このビットは、SBC=0 のとき、スレーブモードでは効果がありません。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 25 **AUTOEND** : 自動終了モード (マスタモード)

このビットは、ソフトウェアによってセット/クリアされます。

0 : ソフトウェア終了モード : NBYTES データが転送されると TC フラグがセットされ、SCL ローをストレッチします。

1 : 自動終了モード : NBYTES データが転送されると、STOP コンディションが自動的に送信されます。

**注 :** このビットは、スレーブモードまたは RELOAD ビットがセットされているときには効果がありません。

ビット 24 **RELOAD** : NBYTES 再ロードモード

このビットは、ソフトウェアによってセット/クリアされます。

0 : 転送は、NBYTES データの転送後、完了します (STOP または RESTART が続きます)。

1 : 転送は、NBYTES データの転送後に完了しません (NBYTES が再ロードされます)。NBYTES データが転送されると TCR フラグがセットされ、SCL ローをストレッチします。

ビット 23:16 **NBYTES[7:0]** : バイト数

送受信されるバイト数は、ここでプログラムされます。このフィールドは、SBC=0 のスレーブモードでは効果がありません。

**注 :** START ビットがセットされているときに、これらのビットを変更することはできません。

### ビット 15 NACK : NACK 生成 (スレーブモード)

このビットはソフトウェアによってセットされ、NACK が送信されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、または PE=0 のとき、ハードウェアによってクリアされます。

- 0 : 現在の受信バイト後に ACK が送信されます。
- 1 : 現在の受信バイト後に NACK が送信されます。

**注 :** このビットに 0 を書き込んでも、効果はありません。

このビットは、スレーブモードでのみ使用されます : マスタレシーバモードでは、NACK ビットの値にかかわらず、STOP または RESTART コンディション前の最後のバイト後に NACK が自動的に生成されます。

スレーブレシーバ NOSTRETCH モードでオーバーランが発生すると、NACK ビットの値にかかわらず、NACK が自動的に生成されます。

ハードウェア PEC チェックが有効なとき (PECBYTE=1)、PEC 確認応答値は NACK 値に依存しません。

### ビット 14 STOP : STOP 生成 (マスタモード)

このビットはソフトウェアによってセットされ、STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

**マスタモード :**

- 0 : STOP 生成なし。
- 1 : 現在のバイト転送後の STOP 生成。

**注 :** このビットに 0 を書き込んでも、効果はありません。

### ビット 13 START : START 生成

このビットはソフトウェアによってセットされ、START とアドレスシーケンスが送信された後、アービトレーション喪失によって、タイムアウトエラー検出によって、または PE=0 のときに、ハードウェアによってクリアされます。I2C\_ICR レジスタの ADDRCF ビットに“1”を書き込むことによって、ソフトウェアによってクリアすることもできます。

- 0 : START 生成なし。
- 1 : RESTART/START 生成 :

- I2C がすでにマスタモードであり、AUTOEND=0 の場合、このビットをセットすると、RELOAD=0 のとき、NBYTES 転送の終了後に REPEATED START コンディションが生成されます。

- そうでない場合、このビットをセットすると、バスがフリーになると、START コンディションが生成されます。

**注 :** このビットに 0 を書き込んでも、効果はありません。

バスが BUSY の場合、または I2C がスレーブモードの場合でも、START ビットをセットできません。

このビットは、RELOAD がセットされているときには効果がありません。

### ビット 12 HEAD10R : 読み出し方向のみの 10 ビットアドレスヘッダ (マスタレシーバモード)

- 0 : マスタは完全な 10 ビットスレーブアドレス読み出しシーケンスを送信します : START + 2 バイトの書き込み方向の 10 ビットアドレス + RESTART + 読み出し方向の 10 ビットアドレスの最初の 7 ビット。
- 1 : マスタは 10 ビットアドレスの最初の 7 ビットのみを送信し、その後読み出し方向を送信します。

**注 :** START ビットがセットされているときに、このビットを変更することはできません。

### ビット 11 ADD10 : 10 ビットアドレッシングモード (マスタモード)

- 0 : マスタは 7 ビットアドレッシングモードで動作します。
- 1 : マスタは 10 ビットアドレッシングモードで動作します。

**注 :** START ビットがセットされているときに、このビットを変更することはできません。

### ビット 10 RD\_WRN : 転送方向 (マスタモード)

- 0 : マスタは書き込み転送をリクエストします。
- 1 : マスタは読み出し転送をリクエストします。

**注 :** START ビットがセットされているときに、このビットを変更することはできません。

ビット 9:8 **SADD[9:8]** : スレーブアドレスビット 9:8 (マスタモード)

**7 ビットアドレッシングモード (ADD10=0) :**

これらのビットは無視されます。

**10 ビットアドレッシングモード (ADD10=1) :**

これらのビットには、送信されるスレーブアドレスのビット 9:8 を書き込みます。

**注:** *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 7:1 **SADD[7:1]** : スレーブアドレスビット 7:1 (マスタモード)

**7 ビットアドレッシングモード (ADD10=0) :**

これらのビットには、送信される 7 ビットのスレーブアドレスを書き込みます。

**10 ビットアドレッシングモード (ADD10=1) :**

これらのビットには、送信されるスレーブアドレスのビット 7:1 を書き込みます。

**注:** *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 0 **SADD0** : スレーブアドレスビット 0 (マスタモード)

**7 ビットアドレッシングモード (ADD10=0) :**

このビットは無視されます。

**10 ビットアドレッシングモード (ADD10=1) :**

このビットには、送信されるスレーブアドレスのビット 0 を書き込みます。

**注:** *START ビットがセットされているときに、これらのビットを変更することはできません。*

## 30.7.3 Own Address 1 レジスタ (I2C\_OAR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]						OA1[0]	
rw					rw	rw		rw						rw	

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA1EN** : Own Address 1 有効

0 : Own Address 1無効受信されたスレーブアドレス OA1 は NACK されます。

1 : Own Address 1 有効受信されたスレーブアドレス OA1 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **OA1MODE** Own Address 1 10 ビットモード

0 : Own Address 1は 7 ビットアドレスです。

1 : Own Address 1は 10 ビットアドレスです。

**注 :** このビットは、**OA1EN=0** のときのみ書き込むことができます。

ビット 9:8 **OA1[9:8]** : インタフェースアドレス

7 ビットアドレッシングモード : 無視されます。

10 ビットアドレッシングモード : アドレスのビット 9:8。

**注 :** これらのビットは、**OA1EN=0** のときのみ書き込むことができます。

ビット 7:1 **OA1[7:1]** : インタフェースアドレス

アドレスのビット 7:1

**注 :** これらのビットは、**OA1EN=0** のときのみ書き込むことができます。

ビット 0 **OA1[0]** : インタフェースアドレス

7 ビットアドレッシングモード : 無視されます。

10 ビットアドレッシングモード : アドレスのビット 0。

**注 :** このビットは、**OA1EN=0** のときのみ書き込むことができます。

## 30.7.4 Own Address 2 レジスタ (I2C\_OAR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]						Res.	
rw					rw			rw							

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA2EN** : Own Address 2 有効

- 0 : Own Address 2無効受信されたスレーブアドレス OA2 は NACK されます。
- 1 : Own Address 2 有効受信されたスレーブアドレス OA2 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **OA2MSK[2:0]** : Own Address 2マスク

- 000 : マスクなし。
- 001 : OA2[1] はマスクされ、無視されます。OA2[7:2] のみ比較されます。
- 010 : OA2[2:1] はマスクされ、無視されます。OA2[7:3] のみ比較されます。
- 011 : OA2[3:1] はマスクされ、無視されます。OA2[7:4] のみ比較されます。
- 100 : OA2[4:1] はマスクされ、無視されます。OA2[7:5] のみ比較されます。
- 101 : OA2[5:1] はマスクされ、無視されます。OA2[7:6] のみ比較されます。
- 110 : OA2[6:1] はマスクされ、無視されます。OA2[7] のみ比較されます。
- 111 : OA2[7:1] はマスクされ、無視されます。比較は行われず、すべての (予約済みを除く) 7 ビット受信アドレスが確認応答されます。

**注 :** これらのビットは、OA2EN=0 のときのみ書き込むことができます。

**OA2MSK が 0 でなくなると、予約済み I2C アドレス (0b0000xxx および 0b1111xxx) は、比較が一致した場合でも確認応答されません。**

ビット 7:1 **OA2[7:1]** : インタフェースアドレス

アドレスのビット 7:1

**注 :** これらのビットは、OA2EN=0 のときのみ書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。



## 30.7.5 タイミングレジスタ (I2C\_TIMINGR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

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

ビット 31:28 **PRESC[3:0]** : タイミングプリスケアラ

このフィールドは、データのセットアップおよびホールドカウンタ ([I2C のタイミング \(925 ページ\)](#)) を参照) と SCL ハイおよびローレベルカウンタ ([I2C マスタ初期化 \(940 ページ\)](#)) を参照) に使用されるクロック周期  $t_{PRESC}$  を生成するため、I2CCLK をプリスケールするために使用されます。

$$t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$$

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **SCLDEL[3:0]** : データセットアップ時間

このフィールドは、送信モードの SDA エッジと SCL 立ち上がりエッジの間に遅延  $t_{SCLDEL}$  を生成するために使用されます。

$$t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$$

**注 :**  $t_{SCLDEL}$  は、 $t_{SU:DAT}$  タイミングを生成するために使用されます。

ビット 19:16 **SDADEL[3:0]** : データホールド時間

このフィールドは、送信モードの SCL 立ち下がりエッジと SDA エッジの間に遅延  $t_{SDADEL}$  を生成するために使用されます。

$$t_{SDADEL} = SDADEL \times t_{PRESC}$$

**注 :**  $SDADEL$  は、 $t_{HD:DAT}$  タイミングを生成するために使用されます。

ビット 15:8 **SCLH[7:0]** : SCL ハイ周期 (マスタモード)

このフィールドは、マスタモードで SCL ハイ周期を生成するために使用されます。

$$t_{SCLH} = (SCLH+1) \times t_{PRESC}$$

**注 :**  $SCLH$  は、 $t_{SU:STO}$  および  $t_{HD:STA}$  タイミングを生成するためにも使用されます。

ビット 7:0 **SCLL[7:0]** : SCL ロー周期 (マスタモード)

このフィールドは、マスタモードで SCL ロー周期を生成するために使用されます。

$$t_{SCLL} = (SCLL+1) \times t_{PRESC}$$

**注 :**  $SCLL$  は、 $t_{BUF}$  および  $t_{SU:STA}$  タイミングを生成するためにも使用されます。

**注 :** このレジスタは、I2C が無効 (PE=0) のときに設定する必要があります。

## 30.7.6 タイムアウトレジスタ (I2C\_TIMEOUTR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

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

ビット 31 **TEXTEN** : 拡張クロックタイムアウト有効

0 : 拡張クロックタイムアウト検出は無効です。

1 : 拡張クロックタイムアウト検出は有効です。  $t_{LOW:EXT}$  を超える累積 SCL ストレッチが I2C インタフェースによって行われると、タイムアウトエラーが検出されます (TIMEOUT=1)。

ビット 30:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **TIMEOUTB[11:0]** : バスタイムアウト B

このフィールドは、累積クロック拡張タイムアウトを設定するために使用されます :

マスタモードでは、マスタ累積クロックロー拡張時間 ( $t_{LOW:MEXT}$ ) が検出されます。

スレーブモードでは、スレーブ累積クロックロー拡張時間 ( $t_{LOW:SEXT}$ ) が検出されます。

$$t_{LOW:EXT} = (TIMEOUTB + 1) \times 2048 \times t_{I2CCLK}$$

**注 :** これらのビットは、**TEXTEN=0** のときのみ書き込むことができます。

ビット 15 **TIMOUTEN** : クロックタイムアウト有効

0 : SCL タイムアウト検出は無効です。

1 : SCL タイムアウト検出は有効です : SCL が  $t_{TIMEOUT}$  (TIDLE=0) を超えてローであるか、 $t_{IDLE}$  (TIDLE=1) を超えてハイであった場合、タイムアウトエラーが検出されます (TIMEOUT=1)。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TIDLE** : アイドルクロックタイムアウト検出

0 : TIMEOUTA は、SCL ロータイムアウトの検出に使用されます。

1 : TIMEOUTA は、SCL と SDA の両方のハイタイムアウト (バスアイドル条件) の検出に使用されます。

**注 :** このビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

ビット 11:0 **TIMEOUTA[11:0]** : バスタイムアウト A

このフィールドは、以下を設定するために使用されます :

- TIDLE=0 のときの SCL ロータイムアウト条件  $t_{TIMEOUT}$

$$t_{TIMEOUT} = (TIMEOUTA + 1) \times 2048 \times t_{I2CCLK}$$

- TIDLE=1 のときのバスアイドル条件 (SCL と SDA の両方のハイ)

$$t_{IDLE} = (TIMEOUTA + 1) \times 4 \times t_{I2CCLK}$$

**注 :** これらのビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

**注 :** **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 30.3 : I2C の実装を参照してください。**

## 30.7.7 割り込みおよびステータスレジスタ (I2C\_ISR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0001

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r							r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 **ADDCODE[6:0]** : アドレス一致コード (スレーブモード)

これらのビットは、アドレス一致イベントが発生したときに (ADDR = 1)、受信したアドレスで更新されます。

10 ビットアドレスの場合、ADDCODE は 10 ビットのヘッダとその後のアドレスの 2 つの MSB を示します。

ビット 16 **DIR** : 転送方向 (スレーブモード)

このフラグは、アドレス一致イベントが発生したときに (ADDR=1)、更新されます。

0 : 書き込み転送、スレーブはレシーバモードになります。

1 : 読み出し転送、スレーブはトランスミッタモードになります。

ビット 15 **BUSY** : バスビジー

このフラグは、バスで通信が進行中であることを示します。START コンディションが検出されたときに、ハードウェアによってセットされます。STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERT** : SMBus アラート

このフラグは、SMBHEN=1 (SMBus ホスト設定)、ALERTEN=1、および SMBALERT イベント (立ち下がりがエッジ) が SMBA ピンで検出されたときに、ハードウェアによってセットされます。ALERTCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 12 **TIMEOUT** : タイムアウトまたは  $t_{LOW}$  検出フラグ

このフラグは、タイムアウトまたは拡張クロックタイムアウトが発生したときに、ハードウェアによってセットされます。TIMEOUTCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 11 **PECERR** : 受信時の PEC エラー

このフラグは、受信した PEC が PEC レジスタの内容に一致しないときに、ハードウェアによってセットされます。正しくない PEC の受信後、NACK が自動的に送信されます。PECCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。**

ビット 10 **OVR** : オーバーラン/アンダーラン (スレーブモード)

このフラグは、NOSTRETCH=1 のスレーブモードで、オーバーラン/アンダーランエラーが発生したときに、ハードウェアによってセットされます。OVRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

ビット 9 **ARLO** : アービトレーション喪失

このフラグは、アービトレーション喪失の場合に、ハードウェアによってセットされます。ARLOCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

ビット 8 **BERR** : バスエラー

このフラグは、ペリフェラルが転送に関与しているため、START または STOP コンディションの誤配置が検出されたときに、ハードウェアによってセットされます。このフラグは、スレーブモードのアドレスフェーズではセットされません。BERRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

ビット 7 **TCR** : 転送完了再ロード

このフラグは、RELOAD=1 および NBYTES データが転送されたときに、ハードウェアによってセットされます。NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

**このフラグは、マスタモード、または SBC ビットがセットされているときのスレーブモードでのみ使用されます。**

ビット 6 **TC** : 転送完了 (マスタモード)

このフラグは、RELOAD=0、AUTOEND=0、および NBYTES データが転送されたときに、ハードウェアによってセットされます。START ビットまたは STOP ビットがセットされたときに、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

ビット 5 **STOPF** : STOP 検出フラグ

このフラグは、バス上で STOP コンディションが検出され、ペリフェラルがこの転送に関与しているときに、ハードウェアによってセットされます:

- マスタとして。ただし、STOP コンディションがペリフェラルによって生成される場合。
- または、スレーブとして。ただし、ペリフェラルがこの転送中にアドレス指定されていた場合。

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

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

ビット 4 **NACKF** : 非確認応答受信フラグ

このフラグは、バイト送信後に NACK を受信したときに、ハードウェアによってセットされます。NACKCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。

ビット 3 **ADDR** : アドレス一致 (スレーブモード)

このビットは、受信したスレーブアドレスが有効なスレーブアドレスの 1 つに一致したときに、ハードウェアによってセットされます。ADDRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、 $PE=0$  のとき、ハードウェアによってクリアされます。



ビット 2 **RXNE** : 受信データレジスタノットエンプティ (レシーバ)

このビットは、受信データが I2C\_RXDR レジスタにコピーされ、読み出す準備ができたときに、ハードウェアによってセットされます。I2C\_RXDR が読み出されたときにクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 1 **TXIS** : 送信割り込みステータス (トランスミッタ)

このビットは、I2C\_TXDR レジスタが空であり、送信データを I2C\_TXDR レジスタに書き込む必要があるときに、ハードウェアによってセットされます。次の送信データが I2C\_TXDR レジスタに書き込まれたときにクリアされます。

このビットは、NOSTRETCH=1 のときのみ、ソフトウェアによって 1 を書き込んで、TXIS イベントを生成することができます (TXIE=1 の場合に割り込み、または TXDMAEN=1 の場合に DMA リクエスト)。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 0 **TXE** : 送信データレジスタエンプティ (トランスミッタ)

このビットは、I2C\_TXDR レジスタが空のときに、ハードウェアによってセットされます。次の送信データが I2C\_TXDR レジスタに書き込まれたときにクリアされます。

このビットは、ソフトウェアによって 1 を書き込んで、送信データレジスタ I2C\_TXDR をフラッシュできます。

**注:** このビットは、PE=0 のとき、ハードウェアによってセットされます。

## 30.7.8 割り込みクリアレジスタ (I2C\_ICR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERTC F	TIM OUTCF	PECCF	OVRCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDRC F	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERTCF** : アラートフラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの ALERT フラグがクリアされます。

**注:** SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。

ビット 12 **TIMOUTCF** : タイムアウト検出フラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの TIMEOUT フラグがクリアされます。

**注:** SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。

ビット 11 **PECCF** : PEC エラーフラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの PECERR フラグがクリアされます。

**注:** SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 30.3 : I2C の実装を参照してください。

ビット 10 **OVRCF** : オーバーラン/アンダーランフラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの OVR フラグがクリアされます。

- ビット 9 **ARLOCF** : アービトレーション喪失フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの ARLO フラグがクリアされます。
- ビット 8 **BERRCF** : バスエラーフラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの BERRF フラグがクリアされます。
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **STOPCF** : STOP 検出フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの STOPF フラグがクリアされます。
- ビット 4 **NACKCF** : 非確認応答フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの ACKF フラグがクリアされます。
- ビット 3 **ADDRCF** : アドレス一致フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの ADDR フラグがクリアされます。このビットに 1 を書き込むと、I2C\_CR2 レジスタの START ビットもクリアされます。
- ビット 2:0 予約済みであり、リセット値に保持する必要があります。

### 30.7.9 PEC レジスタ (I2C\_PECR)

アドレスオフセット : 0x20  
リセット値 : 0x0000 0000  
アクセス : ウェイト状態なし

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

- ビット 31:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7:0 **PEC[7:0]** パケットエラーチェックレジスタ  
PECEN=1 のとき、このフィールドは内部 PEC を含みます。  
PEC は、PE=0 のとき、ハードウェアによってクリアされます。

**注 :** **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 30.3 : I2C の実装 を参照してください。**



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

## 30.7.10 受信データレジスタ (I2C\_RXDR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RXDATA[7:0]** 8 ビット受信データ  
I<sup>2</sup>C バスから受信したデータバイト。

## 30.7.11 送信データレジスタ (I2C\_TXDR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TXDATA[7:0]** 8 ビット送信データ  
I<sup>2</sup>C バスに送信されるデータバイト。  
**注 :** これらのビットは、TXE=1 のときのみ書き込むことができます。

## 30.7.12 I2C レジスタマップ

次の表に、I2C のレジスタマップとリセット値を示します。

表 164. I2C レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0	I2C_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	PECCN	ALERTEN	SMBDEN	SMBHEN	GCEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res	ANFOFF	DNF[3:0]			ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE		
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x4	I2C_CR2	Res	Res	Res	Res	Res	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]							NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]											
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x8	I2C_OAR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OA1EN	Res	Res	Res	Res	Res	OA1MODE	OA1[9:0]									
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0xC	I2C_OAR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OA2EN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	I2C_TIMINGR	PRESC[3:0]			Res	Res	Res	Res	Res	SCLDEL[3:0]			SDADEL[3:0]			SCLH[7:0]			SCLL[7:0]															
	リセット値	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	I2C_TIMEOUTR	TEXTEN	Res	Res	Res	TIMEOUTB[11:0]										TIMOUTEN	Res	TIDLE	TIMEOUTA[11:0]															
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	I2C_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DIR	BUSY	Res	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDRF	RXNE	TXIS	TXE
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	I2C_ICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ALERTCF	TIMEOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res	Res	STOPCF	NACKCF	ADDRCF	Res	Res	Res	
	リセット値																			0	0	0	0	0	0			0	0	0				
0x20	I2C_PECR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	
0x24	I2C_RXDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	
0x28	I2C_TXDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																																	



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

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 31 USART (Universal synchronous asynchronous receiver transmitter)

### 31.1 概要

USART (Universal Synchronous Asynchronous Receiver Transmitter) を使用すると、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟に全二重データ交換を行うことができます。USART は、プログラム可能なボーレートジェネレータを使用して、非常に広範囲のボーレートを提供します。

USART は、マルチプロセッサ通信に加え、同期式単方向通信と半二重単線通信をサポートします。また、LIN (Local Interconnect Network)、スマートカードプロトコル、および IrDA (Infrared Data Association) SIR ENDEC 仕様、およびモデム操作 (CTS/RTS) もサポートします。

マルチバッファ設定で DMA (直接メモリアクセス) を使用することによって、高速データ通信が可能です。

### 31.2 USART の主な機能

- 全二重非同期通信
- NRZ 標準フォーマット (マーク/スペース)
- 16 倍または 8 倍に設定可能なオーバーサンプリング方式によって、速度とクロック誤差の間の柔軟性を実現
- 最大 27 Mbit/s の共通にプログラム可能な送信および受信ボーレート (USART クロックソースがシステムクロック周波数 (最大 216 MHz) で、8 倍のオーバーサンプリング使用時)
- 以下を許可するデュアルクロックドメイン：
  - PCLK 再プログラミングから独立した便利なボーレートプログラミング
- 自動ボーレート検出
- プログラム可能なデータワード長 (7 または 8 または 9 ビット)
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 設定可能なストップビット (1 または 2 個のストップビット)
- 同期通信のための同期モードとクロック出力
- 単線半二重通信
- DMA を使用した連続通信
- 送受信バイトは集中型 DMA を使用して専用 SRAM にバッファリング
- トランスミッタとレシーバ用に個別の有効ビット
- 送信と受信の信号極性を個別に制御
- スワップ可能な Tx/Rx ピン設定
- モデムと RS-485 トランシーバのハードウェアフロー制御

- 通信制御／エラー検出フラグ
- パリティ制御：
  - － パリティビットの送信
  - － 受信したデータバイトのパリティ検査
- 14 の割り込みソース（フラグ付き）
- マルチプロセッサ通信  
アドレスが一致しない場合、USART はミュートモードに入ります。
- ミュートモードからのウェイクアップ（アイドルライン検出またはアドレスマーク検出による）

## 31.3 USART の拡張機能

- LIN マスタの同期ブ레이크送信機能と LIN スレーブのブ레이크検出機能
  - － USART が LIN 用にハードウェア設定されている場合、13 ビットのブ레이크生成と 10/11 ビットのブ레이크検出
- 通常モードで 3/16 ビット期間をサポートする IrDA SIR エンコーダデコーダ
- スマートカードモード
  - － ISO/IEC 7816-3 標準で定義されているスマートカードの T=0 および T=1 非同期プロトコルをサポート
  - － スマートカード動作用に 0.5 個および 1.5 個のストップビット
- Modbus 通信のサポート
  - － タイムアウト機能
  - － CR/LF キャラクタ認識

## 31.4 USART の実装

表 165. STM32F75xxx および STM32F74xxx USART の機能

USART のモード/機能 <sup>(1)</sup>	USART1/USART2/ USART3/USART6	UART4/UART5/ UART7/UART8
モデムのハードウェアフロー制御	X	X
DMA を使用した連続通信	X	X
マルチプロセッサ通信	X	X
同期モード	X	-
スマートカードモード	X	-
単線半二重通信	X	X
Ir SIR ENDEC ブロック	X	X
LIN モード	X	X
デュアルクロックドメイン	X	X
レシーバタイムアウト割り込み	X	X
Modbus 通信	X	X
自動ボーレート検出	X	X
ドライバインープル	X	X
USART データ長	7、8、および9ビット	

1. X: サポートされています。

## 31.5 USART の機能詳細

USART の双方向通信には、少なくとも 2 本のピンが必要です。すなわち、受信データ入力 (RX) と送信データ出力 (TX) です。

- **RX**: 受信データ入力。  
これはシリアルデータ入力です。データ復旧にはオーバーサンプリング技術が使用され、有効な入力データとノイズを区別しています。
- **TX**: 送信データ出力。  
トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、送信すべきデータがないとき、TX ピンはハイレベルになります。単線 およびスマートカードモードでは、この I/O はデータの送受信に使用されます。

シリアルデータは、通常の USART モードでは、これらのピンを通じて送受信されます。フレームは、以下で構成されます。

- 送受信前のアイドルライン
- スタートビット
- データワード (7、8、または 9 ビット) LSBファースト
- フレームの完了を示す 0.5 個、1 個、1.5 個、2 個のストップビット
- USART インタフェースはポーレートジェネレータを使用
- ステータスレジスタ (USARTx\_ISR)
- 受信および送信データレジスタ (USARTx\_RDR、USARTx\_TDR)
- ポーレートレジスタ (USARTx\_BRR)
- ガードタイムレジスタ (USARTx\_GTPR) - スマートカードモードの場合

各ビットの定義については、[セクション 31.8:USART レジスタ \(1030 ページ\)](#) を参照してください。

同期モードおよびスマートカードモードでインタフェースするには、以下のピンが必要です。

- **SCLK** : クロック出力。このピンは、SPI マスタモードに対応する同期送信用のトランスミッタデータクロックを出力します (スタートビットとストップビットのクロックパルスはなく、ソフトウェアオプションで最後のデータビットのクロックパルスを送信します)。これと並行して、RX でデータを同期受信できます。これを使用して、シフトレジスタを持つペリフェラルを制御できます。クロックの位相と極性は、ソフトウェアでプログラム可能です。スマートカードモードでは、SCLK 出力はスマートカードにクロックを供給できます。

RS232 ハードウェアフロー制御モードでは、以下のピンが必要です。

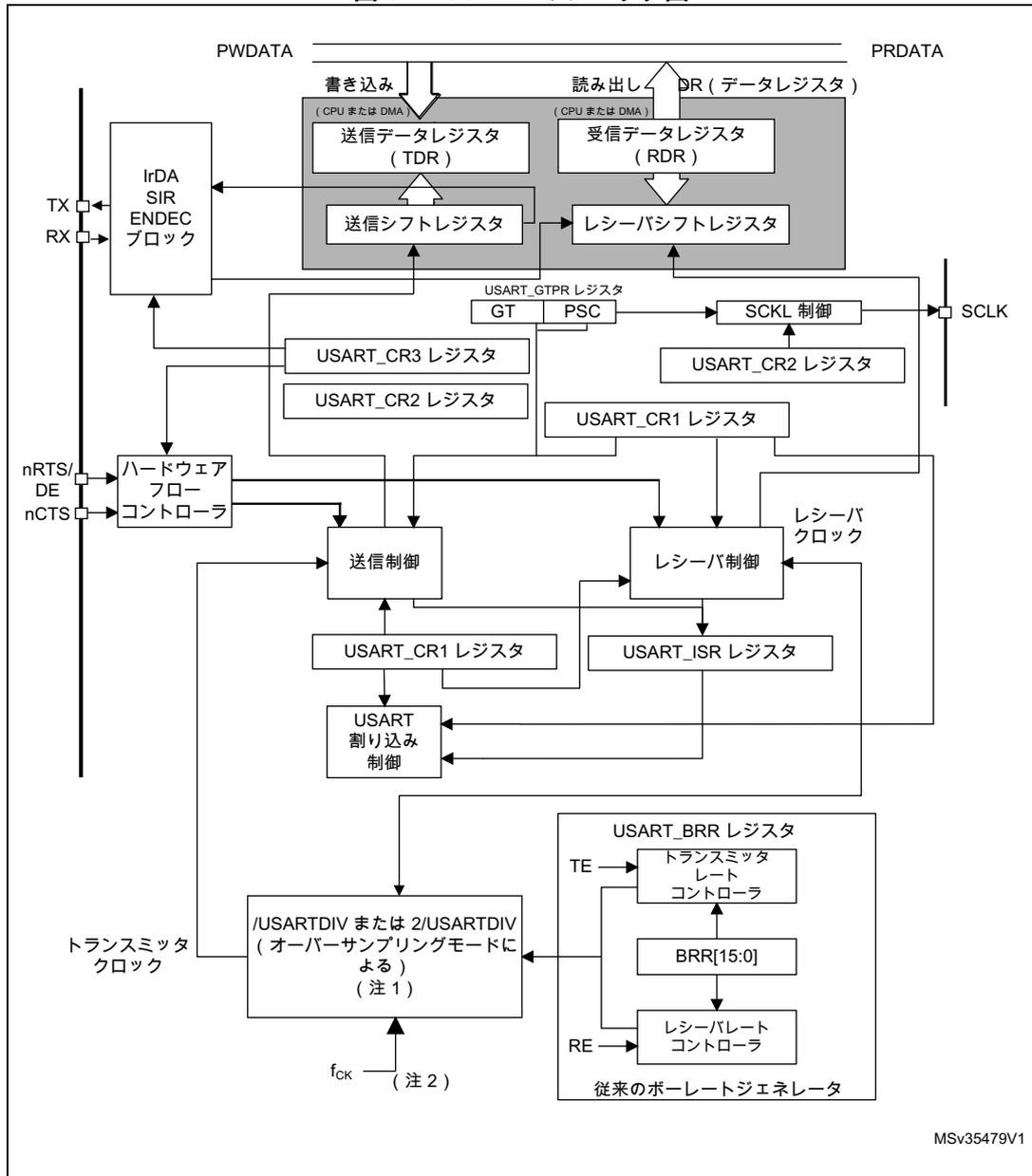
- **nCTS** : Clear To Send は、ハイレベルのとき、現在の転送の終わりにデータ送信をブロックします。
- **nRTS** : Request To Send は、ローレベルのとき、USART がデータを受信する準備ができたことを示します。

RS485 ハードウェアフロー制御モードでは、以下のピンが必要です。

- **DE** : Driver Enable は、外部トランシーバの送信モードを有効にします。

**注 :** **DE と nRTS は同じピンを共有します。**

図 314. USART のブロック図



MSV35479V1

1. USARTx\_BRR レジスタの USARTDIV のコーディングの詳細については、[セクション 31.5.4: ボーレート生成](#)を参照してください。
2. f<sub>CK</sub> は、f<sub>LSE</sub>、f<sub>HSI</sub>、f<sub>PCLK</sub>、f<sub>SYS</sub> のいずれかです。

## 31.5.1 USART キャラクタの説明

ワード長は、USARTx\_CR1 レジスタの M[1:0] ビットをプログラムすることによって、7、8、または 9 ビットとして選択できます (図 315 を参照)。

- 7 ビットのキャラクタ長 : M[1:0] = 10
- 8 ビットのキャラクタ長 : M[1:0] = 00
- 9 ビットのキャラクタ長 : M[1:0] = 01

**注 :** 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ボーレート (0x7F および 0x55 フレーム検出) はサポートされません。7 ビットモードは、一部の USART でのみサポートされます。

デフォルト設定では、信号 (TX または RX) はスタートビットの処理中ではロー状態です。また、ストップビットの処理中にはハイ状態です。

これらの値は、極性設定制御により、各信号について個別に反転できます。

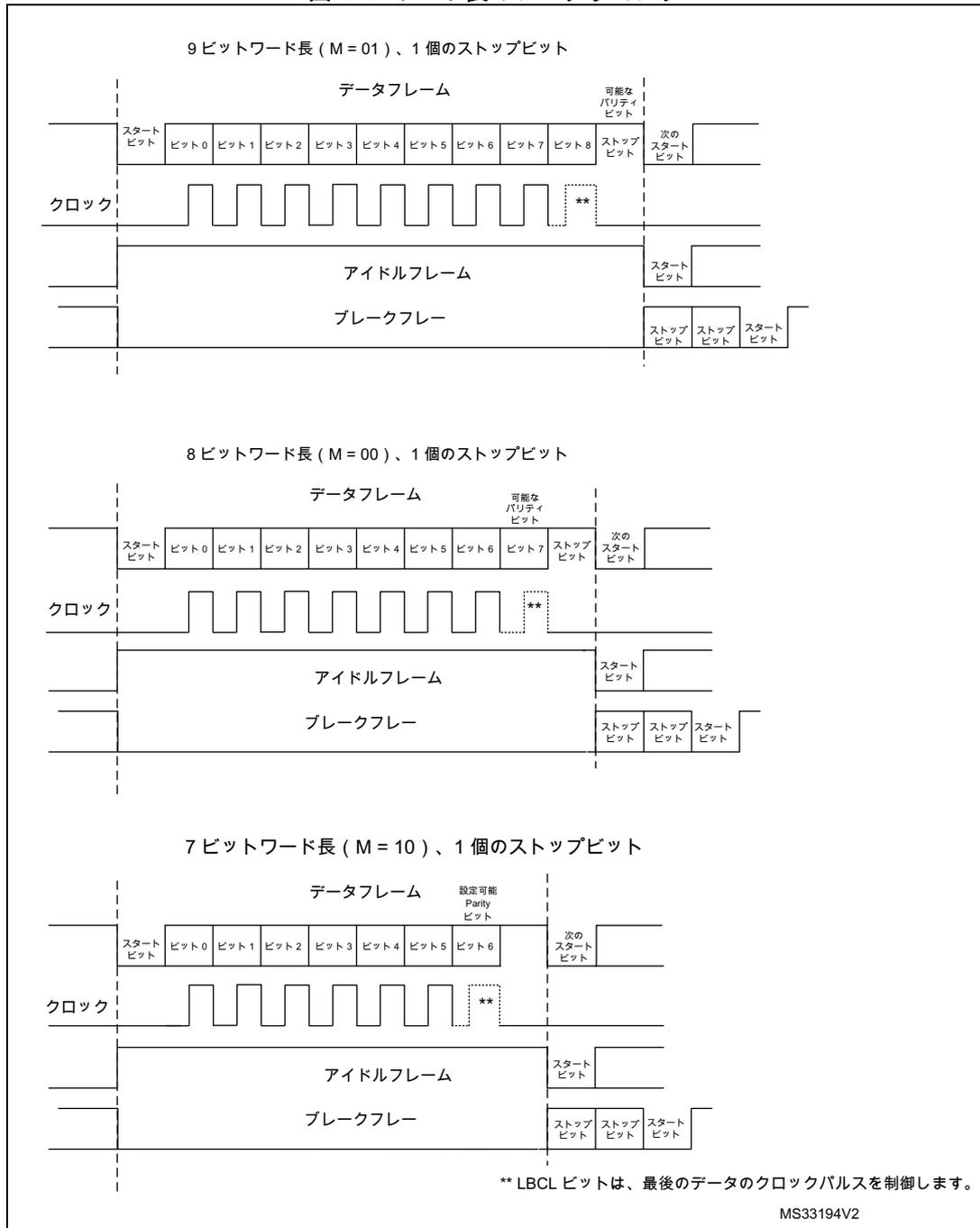
**アイドルキャラクタ**は、すべてが「1」のフレームとして解釈されます (「1」の数にはストップビットの数が含まれます)。

**ブレイクキャラクタ**は、フレーム周期中に「0」を受信することと解釈されます。ブレイクフレームの終了時、トランスミッタは 2 個のストップビットを挿入します。

送信と受信は、共通のボーレートジェネレータによって駆動され、それぞれのクロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。

図 315. ワード長のプログラミング



## 31.5.2 トランスミッタ

トランスミッタは、M ビットのステータスに応じて、7 または 8 または 9 ビットのデータワードを送信できます。トランスミッタ機能を有効にするには、送信イネーブルビット (TE) をセットする必要があります。送信シフトレジスタ内のデータは TX ピンで出力され、対応するクロックパルスは SCLK ピンで出力されます。



## キャラクタ送信

USART 送信時、データはLSBファースト（デフォルト設定）で TX ピンにシフトアウトされます。このモードでは、USARTx\_TDR レジスタは、内部バスと送信シフトレジスタの間のバッファ（TDR）で構成されます（図 314 を参照）。

各キャラクタの前には、スタートビット（1 ビット周期、論理レベルロー）があります。キャラクタは、設定可能な数のストップビットで終端されます。

USART では、0.5、1、1.5、および 2 個のストップビットがサポートされます。

**注：** 送信データを USARTx\_TDR に書き込む前に、TE ビットをセットする必要があります。

データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、ポーレートカウンタが停止されるため、TX ピンのデータが破壊されます。送信中のデータは失われます。

TE ビットが有効になると、アイドルフレームが送信されます。

### 設定可能なストップビット

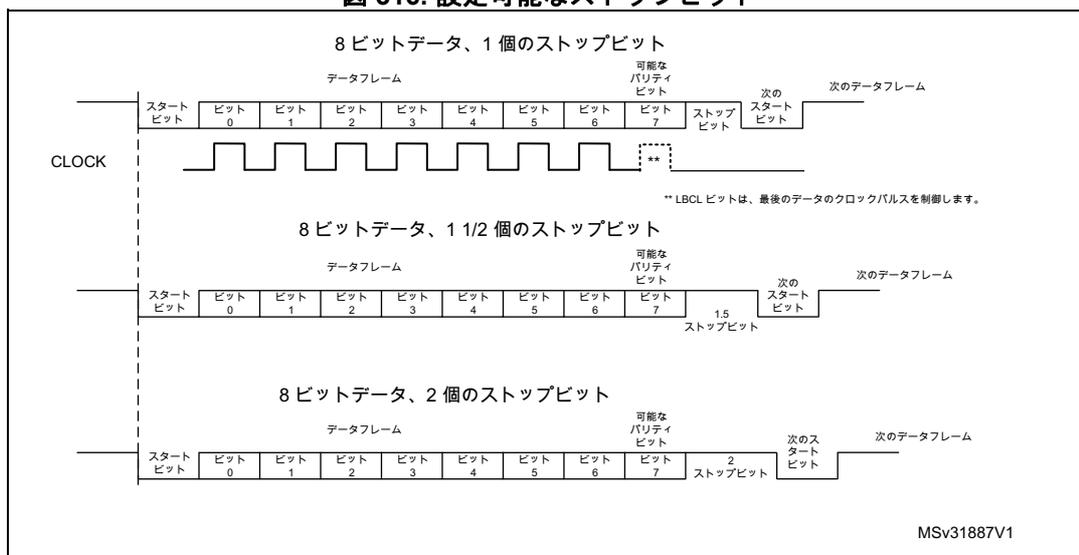
各キャラクタとともに送信されるストップビットの数は、制御レジスタ 2 のビット 13、12 でプログラミングできます。

- **1 個のストップビット：** ストップビット数のデフォルト値です。
- **2 個のストップビット：** 通常の USART モード、単線モード、およびモデムモードでサポートされます。
- **ストップビット 1、5 個：** スマートカードモードで使用されます。
- **0.5 個のストップビット：** スマートカードモードでのデータの受信に使用されます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信は、10 個のロービット（M[1:0] = 00 のとき）または 11 個のロービット（M[1:0] = 01 のとき）または 9 個のロービット（M[1:0] = 10 のとき）の後に 2 個のストップビットが続きます（図 316 を参照）。長いブ레이크（9/10/11 個のロービットを超える長さのブ레이크）を送信することはできません。

図 316. 設定可能なストップビット



## キャラクタ送信手順

1. ワード長を定義するには、USARTx\_CR1 の M ビットをプログラムします。
2. USARTx\_BRR レジスタを使用して、目的のボーレートを選択します。
3. USARTx\_CR2 レジスタでストップビットの数をプログラミングします。
4. USARTx\_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USARTx\_CR3 レジスタの DMA 有効 (DMAT) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
6. USARTx\_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。
7. 送信するデータを USARTx\_TDR レジスタに書き込みます (これによって TXE ビットがクリアされます)。シングルバッファの場合、送信される各データにこれを繰り返します。
8. USARTx\_TDR レジスタに最後のデータを書き込んだ後、TC=1 になるまで待ちます。これは、最後のフレームの送信が完了したことを示します。これは、たとえば最後の送信内容の破壊を避けるために、USART が無効になったり HALT モードに入ったりする場合に必要です。

## 1 バイト通信

TXE ビットは、常に送信データレジスタへの書き込みによってクリアされます。

TXE ビットは、ハードウェアによってセットされ、次を示します。

- データは USARTx\_TDR レジスタからシフトレジスタへ移動され、データ送信が開始していません。
- USARTx\_TDR レジスタは空です。
- 次のデータを、前のデータに上書きせずに、USARTx\_TDR レジスタに書き込みます。

TXEIE ビットがセットされている場合、このフラグは割り込みを生成します。

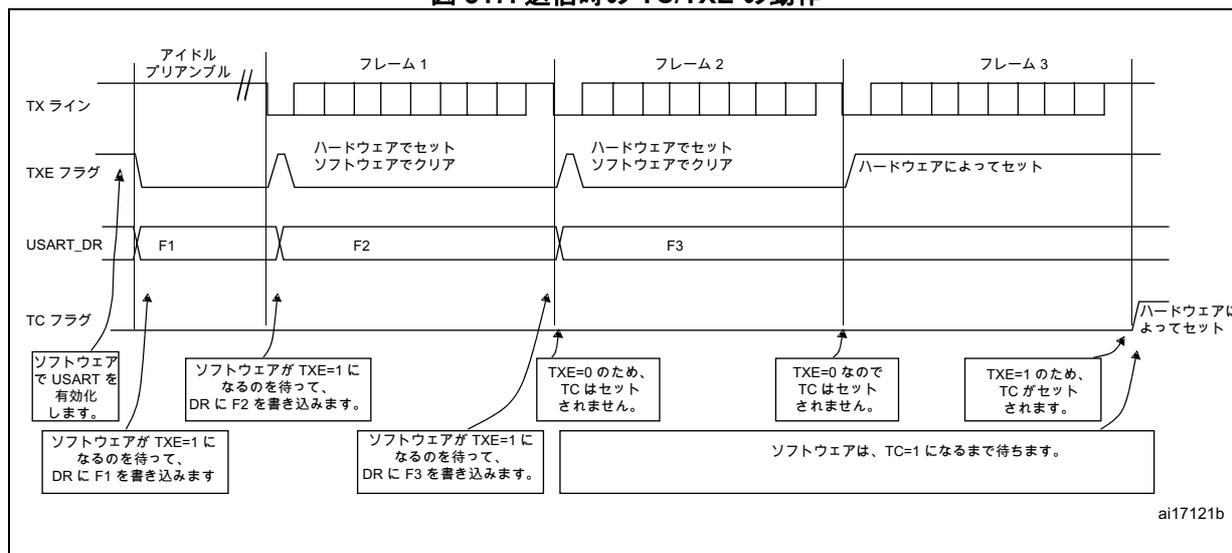
送信が行われている場合、USARTx\_TDR レジスタへの書き込み命令によってデータが TDR レジスタに格納され、次に、データが現在進行中の送信の最後にシフトレジスタにコピーされます。

送信が行われていないときには、USARTx\_TDR レジスタへの書き込み命令によってデータがシフトレジスタに格納され、データ送信が開始され、TXE ビットがセットされます。

フレームが送信され (ストップビットの後)、TXE ビットがセットされると、TC ビットはハイレベルになります。USARTx\_CR1 レジスタの TCIE ビットがセットされると、割り込みが生成されます。

USARTx\_TDR レジスタに最後のデータを書き込んだ後は、USART を無効にしたり、マイクロコントローラを低電力モードにする前に TC=1 になるまで待つ必要があります (図 317 : 送信時の TC/TXE の動作を参照)。

図 317. 送信時の TC/TXE の動作



## ブレイクキャラクタ

SBKRQ ビットをセットすると、ブレイクキャラクタが送信されます。ブレイクフレーム長は、M ビットに依存します (図 315 を参照)。

SBKRQ ビットに 1 が書き込まれた場合、現在のキャラクタ送信の完了後、TX ラインにブレイクキャラクタが送信されます。SBKF ビットは書き込み操作によってセットされ、ブレイクキャラクタが完了すると (ブレイクキャラクタの後のストップビット時に) ハードウェアによってリセットされます。USART は、次のフレームのスタートビットが正しく認識されるように、ブレイクフレームの終わりに 2 ビットの長さの間、ロジック 1 信号 (STOP) を挿入します。

アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブレイクキャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを待つ必要があります。

## アイドルキャラクタ

TE ビットをセットすると、USART は、最初のデータフレームの前にアイドルフレームを送信します。

## 31.5.3 レシーバ

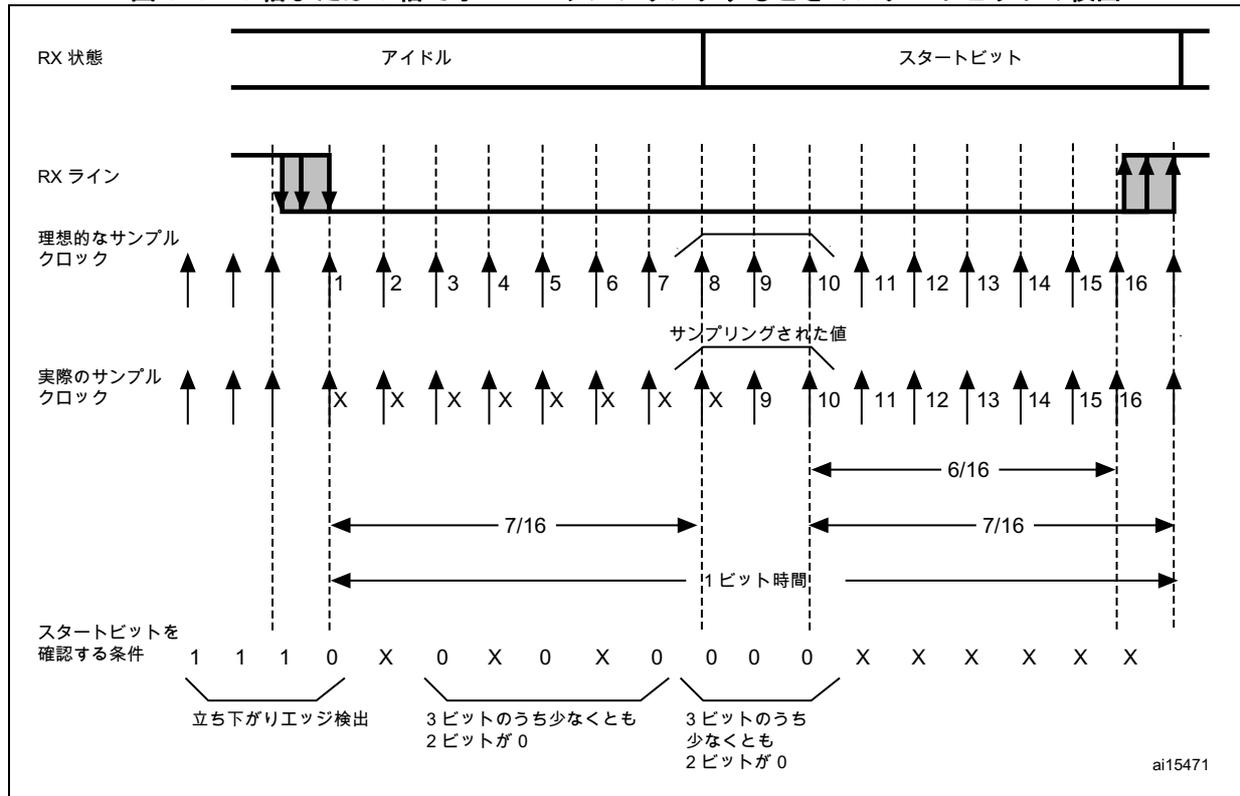
USART は、USARTx\_CR1 レジスタの M ビットに応じて、7、8 または 9 ビットのデータワードを受信できます。

### スタートビット検出

スタートビット検出シーケンスは、16 倍または 8 倍でオーバーサンプリングするときと同じです。

USART では、サンプルの特定シーケンスが認識されると、スタートビットが検出されます。このシーケンスは、1110X0X0X0X0X0X0 です。

図 318. 16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出



**注：** シーケンスが完了していない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻って（フラグはセットされません）、立ち下がりエッジを待ちます。

サンプリングされた 3 つのビットが 0 の場合（第 3、第 5、および第 7 ビットでの最初のサンプリングで 3 つのビットが 0 であり、第 8、第 9、および第 10 ビットでの 2 回目のサンプリングでも 3 つのビットが 0）、スタートビットが確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）。

次の場合、スタートビットは確認されますが（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）、NF ノイズフラグがセットされます。

a. 両方のサンプリングについて、サンプリングされた 3 ビットのうちの 2 ビットが 0 の場合（第 3、第 5、および第 7 ビットのサンプリングと、第 8、第 9、および第 10 ビットのサンプリング）。

または

b. いずれか一方のサンプリングで（第 3、第 5、および第 7 ビットでのサンプリングまたは第 8、第 9、および第 10 ビットでのサンプリング）、3 つのビットのうち 2 つが 0 の場合。

a. または b. のいずれの条件も満たされない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻ります（フラグはセットされません）。

## キャラクタの受信

USART の受信時には、データは RX ピンを通じて LSB ファーストでシフトインされます。このモードでは、USARTx\_RDR レジスタは、内部バスと受信シフトレジスタの間のバッファ (RDR) で構成されます。

### キャラクタ受信手順

1. ワード長を定義するには、USARTx\_CR1 の M ビットをプログラムします。
2. ボーレートレジスタ USARTx\_BRR を使用して、目的のボーレートを選択します。
3. USARTx\_CR2 レジスタでストップビットの数をプログラミングします。
4. USARTx\_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USARTx\_CR3 レジスタの DMA 有効 (DMAR) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
6. USARTx\_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

キャラクタが受信されると、

- RXNE ビットは、シフトレジスタの内容が RDR レジスタに転送されたことを示すためにセットされます。言い換えると、データは受信され、読み出し可能です (関連するエラーフラグも同様です)。
- RXNEIE ビットがセットされていた場合、割り込みが生成されます。
- 受信中にフレームエラー、ノイズまたはオーバーランエラーが検出された場合、エラーフラグをセットできます。RXNE によって PE フラグもセットできます。
- マルチバッファでは、RXNE はバイト受信ごとにセットされ、受信データレジスタの DMA 読み出しによってクリアされます。
- シングルバッファモードでは、RXNE ビットのクリアは、ソフトウェアによる USARTx\_RDR レジスタの読み出しによって行われます。RXNE フラグは、USARTx\_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE ビットをクリアする必要があります。

## ブレイクキャラクタ

ブレイクキャラクタを受信すると、USART はブレイクキャラクタをフレーミングエラーとして処理します。

## アイドルキャラクタ

アイドルフレームが検出されると、受信データキャラクタの場合と同じ手順が行われ、IDLEIE ビットがセットされていた場合は割り込みが生成されます。

## オーバーランエラー

RXNE ビットがリセットされていないときにキャラクタを受信すると、オーバーランエラーが発生します。RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。

RXNE フラグは、バイトを受信するたびにセットされます。次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

- ORE ビットがセットされます。
- RDR の内容は失われません。USARTx\_RDR の読み出しが行われると、前のデータが使用できません。
- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXNEIE ビットがセットされているか、EIE ビットがセットされている場合、割り込みが生成されます。
- ORE ビットは、ICR レジスタの ORECF ビットをセットすることによってリセットされます。

**注：** ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。この場合、次の 2 つの可能性あります。

- RXNE=1 の場合、最後の有効なデータが受信レジスタ RDR に格納され、読み出すことができます。

- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR には読み出すべきものがないことを意味します。このケースは、有効な最後のデータが RDR で読み出されると同時に新しい（そして失われた）データが受信されると発生します。

## クロックソースと適切なオーバーサンプリング方式の選択

クロックソースの選択は、クロック制御システムを通じて行われます（リセットおよびクロック制御 (RCC) のセクションを参照）。クロックソースは、USART を有効にする（UE ビットをセットする）前に選ぶ必要があります。

クロックソース周波数は、 $f_{CK}$  です。

デュアルクロックドメインがサポートされるときには、クロックソースは次のいずれかのソースです。PCLK（デフォルト）、LSE、HSI、または SYSCLK。

通信速度の範囲（特に最大通信速度）もクロックソースによって決まります。

レシーバは有効な受信データとノイズを区別して、データを復旧するユーザ設定可能なさまざまなオーバーサンプリング技術を実装しています。これにより、最大通信速度とノイズ/クロック精度の耐性の間でトレードオフができます。

オーバーサンプリング方式は、USARTx\_CR1 レジスタの OVER8 ビットをプログラムすることによって選択でき、ボーレートクロックの 16 倍または 8 倍にすることができます（[図 319](#) および [図 320](#)）。

アプリケーションに応じて、

- 高速（最大  $f_{CK}/8$ ）を達成するには、8 倍（OVER8=1）のオーバーサンプリングを選択します。この場合、クロック偏差に対するレシーバの最大許容誤差は軽減されます（[セクション 31.5.5：クロック偏差に対する USART レシーバの許容誤差 \(1005 ページ\)](#) を参照）。
- クロック偏差に対するレシーバの許容誤差を増やすには、16 倍（OVER8=0）のオーバーサンプリングを選択します。この場合、最大速度は  $f_{CK}/16$  に制限されます。 $f_{CK}$  はクロックソース周波数です。

論理レベルの評価方法を選択するには、USARTx\_CR3 レジスタの ONEBIT ビットをプログラミングします。これには 2 つのオプションがあります。

- 受信されたビットの中央にある 3 つのサンプルの多数決。この場合、多数決に使用された 3 つのサンプルが等しくないとき、NF ビットがセットされます。
- 受信されたビットの中央にある 1 つのサンプル。

アプリケーションに応じて、

- ノイズの多い環境で操作するときは 3 つのサンプルの多数決方式 (ONEBIT=0) を選択します。そしてノイズが検出された場合 (図 166 を参照) は、サンプリング中にグリッチが発生していることとなりますので、そのデータを除去します。
- ラインがノイズフリーであるときは、1 つのサンプルによる方式 (ONEBIT=1) を選択し、クロック偏差に対するレシーバの許容誤差を増やします (セクション 31.5.5: クロック偏差に対する USART レシーバの許容誤差 (1005 ページ) を参照)。この場合、NF ビットはセットされません。

フレーム内でノイズが検出された場合 :

- RXNE ビットの立ち上がりエッジで、NF ビットがセットされます。
- 無効なデータがシフトレジスタから USARTx\_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USARTx\_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

NF ビットは、ICR レジスタの NFCF ビットをセットすることによってリセットされます。

**注 :** 8 倍のオーバーサンプリングは、LIN、スマートカード、および IrDA の各モードでは使用できません。これらのモードでは、OVER8 ビットはハードウェアによって 0 に固定されています。

図 319. データサンプリング (16 倍のオーバーサンプリング)

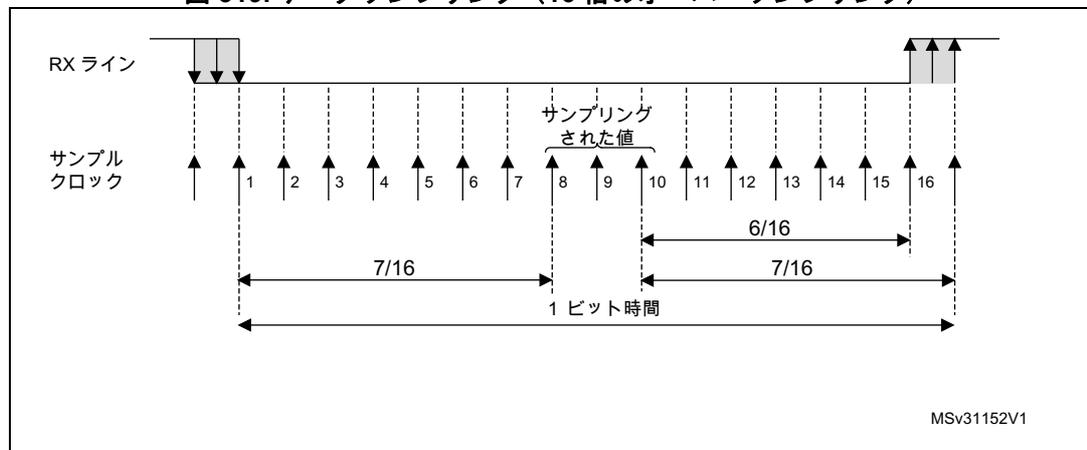


図 320. データサンプリング (8 倍のオーバーサンプリング)

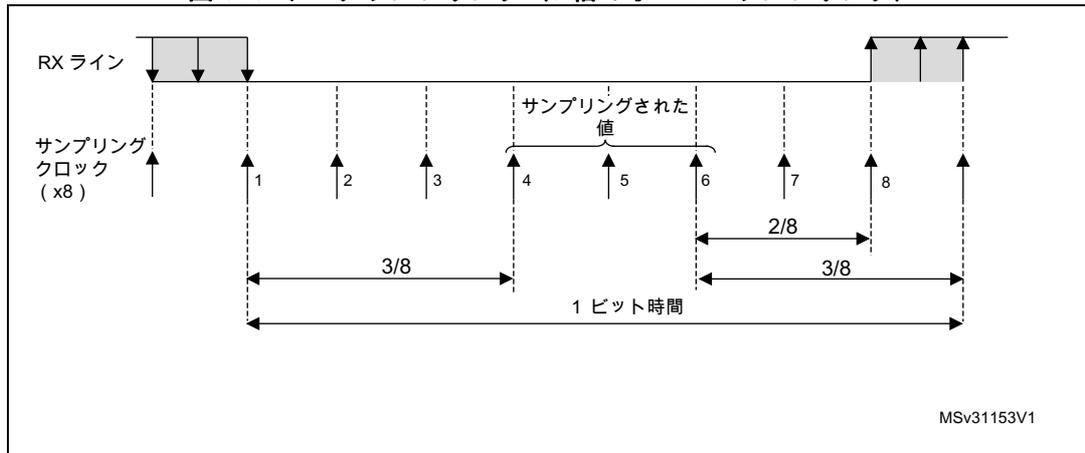


表 166. サンプリングされたデータからのノイズ検出

サンプリングされた値	NE ステータス	受信ビットの値
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

## フレーミングエラー

フレーミングエラーは、次の場合に検出されます。

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから USARTx\_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USARTx\_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

USARTx\_ICR レジスタの FECF に 1 を書き込むことによって、FE ビットがリセットされます。

## 受信時の設定可能なストップビット

受信するストップビット数は、制御レジスタ 2 の制御ビットを通じて設定でき、通常モードでは 1 または 2、スマートカードモードでは 0.5 または 1.5 にできます。

- **0.5 個のストップビット (スマートカードモードでの受信)**: 0.5 個のストップビットでは、サンプリングは行われません。したがって、0.5 個のストップビットが選択されている場合、フレーミングエラーやブレイクフレームは検出されません。
- **1 個のストップビット**: ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
- **1.5 個のストップビット (スマートカードモード)**: スマートカードモードでの送信時は、デバイスは、データが正しく送信されたことをチェックする必要があります。したがって、レシーバブロックを有効にし (USARTx\_CR1 レジスタの RE=1)、ストップビットをチェックして、スマートカードがパリティエラーを検出したかどうかをテストする必要があります。パリティエラーが発生した場合、スマートカードはサンプリング時のデータ信号を強制的にローレベルにします (これは、フレーミングエラーとしてフラグされる NACK 信号です)。その後、1.5 個のストップビットの最後に、RXNE ビットとともに FE フラグがセットされます。ストップビット 1.5 個のサンプリングは、16 番目、17 番目、および 18 番目のサンプルで (ストップビットの開始から 1 ボークロック周期後に) 行われます。1.5 個のストップビットは、2 つの部分に分解できます。すなわち、何も起こらない 0.5 ボークロック周期と、途中でサンプリングが行われる通常の 1 ストップビット周期です。詳細については、[セクション 31.5.13: スマートカードモード \(1016 ページ\)](#) を参照してください。
- **2 個のストップビット**: ストップビット 2 個のサンプリングは、最初のストップビットの 8 番目、9 番目、および 10 番目のサンプルで行われます。最初のストップビットでフレーミングエラーが検出された場合、フレーミングエラーフラグがセットされます。2 番目のストップビットでは、フレーミングエラーの検査は行われません。RXNE フラグは、最初のストップビットの最後でセットされます。

## 31.5.4 ボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートは、USARTx\_BRR レジスタでプログラムされたものと同じ値に設定されます。

### 式 1: 標準 USART のボーレート (SPI モードを含む) (OVER8 = 0 または 1)

16 倍のオーバーサンプリングの場合、等式は次のとおりです。

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{\text{USARTDIV}}$$

8 倍のオーバーサンプリングの場合、等式は次のとおりです。

$$\text{Tx/Rx ボー} = \frac{2 \times f_{\text{CK}}}{\text{USARTDIV}}$$

### 式 2: スマートカード、LIN、および IrDA モードのボーレート (OVER8 = 0)

スマートカード、LIN、および IrDA モード。16 倍のオーバーサンプリングのみをサポート:

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{\text{USARTDIV}}$$

USARTDIV は、符号なしの固定小数点数であり、USARTx\_BRR レジスタでコード化されます。

- OVER8 = 0 のとき、BRR = USARTDIV です。
- OVER8 = 1 のとき、
  - BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。
  - BRR[3] は、クリアされたままにする必要があります。
  - BRR[15:4] = USARTDIV[15:4]

**注：** USARTx\_BRR レジスタへの書き込み操作の後、ポーカウンタはポーレジスタの新しい値に更新されます。したがって、通信中はポーレートレジスタの値を変更しないでください。

16 倍または 8 倍のオーバーサンプリングの場合、USARTDIV は 16d 以上である必要があります。

## USARTx\_BRR レジスタの値から USARTDIV を得る方法

### Example 1

$f_{CK} = 8 \text{ MHz}$  で 9600 ボーを得るには、

- 16 倍のオーバーサンプリングの場合：  
USARTDIV =  $8\,000\,000/9600$   
BRR = USARTDIV = 833d = 0341h
- 8 倍のオーバーサンプリングの場合：  
USARTDIV =  $2 * 8\,000\,000/9600$   
USARTDIV = 1666,66 (1667d = 683h)  
BRR[3:0] = 3h >> 1 = 1h  
BRR = 0x681

### Example 2

$f_{CK} = 48 \text{ MHz}$  で 921.6 Kbaud を得るには、

- 16 倍のオーバーサンプリングの場合：  
USARTDIV =  $48\,000\,000/921\,600$   
BRR = USARTDIV = 52d = 34h
- 8 倍のオーバーサンプリングの場合：  
USARTDIV =  $2 * 48\,000\,000/921\,600$   
USARTDIV = 104 (104d = 68h)  
BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h  
BRR = 0x64

表 167.  $f_{CK} = 216 \text{ MHz}$  でプログラムされたボーレートのエラー計算  
(8 倍のオーバーサンプリング (OVER8 = 1) の場合)<sup>(1)</sup>

目的のボーレート (Bps)	実際のボーレート (Bps)	BRR	誤差
9600	9600.853	AFC4	0.00888968
19200	19201.707	57E2	0.00888968
38400	38403.414	2BF1	0.00888968
57600	57646.117	1D46	0.08006405
115200	115292.234	EA3	0.08006405
230400	230646.022	751	0.10678057
460800	463519.313	3A4	0.59012876
921600	927038.627	1D2	0.59012876
13500000	13500000.000	20	0.00000000
27000000	27000000.000	10	0.00000000

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 168.  $f_{CK} = 216 \text{ MHz}$  でプログラムされたボーレートのエラー計算  
(16 倍のオーバーサンプリング (OVER8 = 0) の場合)<sup>(1)</sup>

目的のボーレート (Bps)	実際のボーレート (Bps)	BRR	誤差
9600	9600.000	57E4	0.00000000
19200	19200.000	2BF2	0.00000000
38400	38400.000	15F9	0.00000000
57600	57600.000	EA6	0.00000000
115200	115200.000	753	0.00000000
230400	230522.946	3A9	0.05336179
460800	461538.462	1D4	0.16025641
921600	923076.923	EA	0.16025641
4000000	4000000.000	36	0.00000000
6000000	6000000.000	24	0.00000000
10000000	10285714.286	15	2.85714286
13500000	13500000.000	10	0.00000000

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

## 31.5.5 クロック偏差に対する USART レシーバの許容誤差

USART の非同期レシーバは、クロックシステムの合計偏差が USART レシーバの許容誤差未満の場合のみ、正しく動作します。合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタのローカルオシレータの偏差も含みます)
- DQUANT : レシーバのボーレート量子化に起因する誤差
- DREC : レシーバのローカルオシレータの偏差
- DTCL : 送信ラインに起因する偏差 (一般には、ローからハイへの遷移タイミングとハイからローへの遷移タイミングの間に非対称性をもたらす可能性のあるトランシーバに起因)

$$DTRA + DQUANT + DREC + DTCL < \text{USART receiver's 許容誤差}$$

USART レシーバは、次の選択に応じて、表 169 および 表 170 で指定された最大許容偏差まで、データを正しく受信できます。

- USARTx\_CR1 レジスタの M ビットで定義された 9、10、または 11 ビットキャラクタ長
- USARTx\_CR1 レジスタの OVER8 ビットによって定義された 8 倍または 16 倍のオーバーサンプリング
- USARTx\_BRR レジスタのビット BRR[3:0] が 0000 に等しいかどうか。
- データのサンプリングに 1 ビットを使用するか 3 ビットを使用するか (USARTx\_CR3 レジスタの ONEBIT ビットの値に依存)。

表 169. BRR [3:0] = 0000 のときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 170. BRR[3:0] が 0000 でないときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注 : 表 169 および 表 170 で指定されたデータは、M ビット = 00 のとき、受信フレームに正確に 10 ビット持続時間のアイドルフレームが含まれる特殊なケースで、若干異なることがあります (M ビット = 01 のときには 11 ビット持続時間、または M ビット = 10 のときには 9 ビット持続時間)。



## 31.5.6 自動ボーレート検出

USART は、1 キャラクタの受信に基づいて、USARTx\_BRR レジスタ値を検出し、自動的にセットすることができます。自動ボーレート検出は、2 つの状況で便利です。

- システムの通信速度が事前に分かっていないとき。
- システムが比較的低い精度のクロックソースを使用している場合、このメカニズムによって、クロック偏差を測定しなくても、正しいボーレートを求めることができます。

クロックソース周波数は、予期される通信速度と互換性がなければなりません (16 倍のオーバーサンプリングのとき、ボーレートは  $f_{CK}/65535$  から  $f_{CK}/16$  までの範囲内です。8 倍のオーバーサンプリングのとき、ボーレートは  $f_{CK}/65535$  から  $f_{CK}/8$  までの範囲内です)。

自動ボーレート検出を有効にする前に、自動ボーレート検出モードを選択する必要があります。キャラクタパターンに基づいて、いくつかのモードがあります。

これらのモードは、USARTx\_CR2 レジスタの ABRMOD[1:0] フィールドで選択できます。これらの自動ボーレートモードでは、同期データ受信中にボーレートが数回測定され、各測定値が前回の測定値と比較されます。

以下のモードがあります。

- **モード 0** : 1 のビットで始まるキャラクタ。この場合、USART はスタートビットの時間を測定します (立ち下がリエッジから立ち上がりエッジまで)。
- **モード 1** : 10xx ビットパターンで始まるキャラクタ。この場合、USART はスタートと最初のデータビットの時間を測定します。低速な信号傾斜の場合の精度を高めるために、測定は立ち下がリエッジから立ち上がりエッジまでで行われます。
- **モード 2** : 0x7F キャラクタフレーム (LSB ファーストモードでは 0x7F キャラクタ、または MSB ファーストモードでは 0xFE)。この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 6 の終了時に更新されます (立ち下がリエッジから立ち上がりエッジまで行われた測定に基づいて、BR6)。ビット 0 からビット 6 までが BRs でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。
- **モード 3** : 0x55 キャラクタフレーム。この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 0 の終了時に更新され (立ち下がリエッジから立ち上がりエッジまで行われた測定に基づいて BR0)、最後にビット 6 (BR6) の終了時に更新されます。ビット 0 は BRs でサンプリングされ、ビット 1 からビット 6 までは BR0 でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。

並行して、RX ラインの中間遷移ごとに別のチェックが行われます。RX の遷移がレシーバと十分に同期していない場合はエラーが生成されます (レシーバは、ビット 0 で計算されたボーレートに基づきます)。

自動ボーレート検出を有効にする前に、ゼロ以外のボーレート値を書き込むことによって、USARTx\_BRR レジスタを初期化する必要があります。

自動ボーレート検出を有効にするには、USARTx\_CR2 レジスタの ABREN ビットをセットします。USART は、RX ラインの最初のキャラクタを待ちます。自動ボーレート動作の完了は、USARTx\_ISR レジスタの ABRF フラグのセットによって示されます。ラインにノイズが多い場合、正しいボーレート検出を保証できません。この場合、BRR 値が破損して、ABRE エラーフラグがセットされることがあります。また、通信速度が自動ボーレート検出の範囲と互換性がない場合にも発生します (ビット時間が 16 から 65536 までのクロック周期でなく (16 倍のオーバーサンプリング)、8 から 65536 までのクロック周期でない (8 倍のオーバーサンプリング))。

RXNE 割り込みは、操作の終了を知らせます。

その後いつでも、ABRF フラグをリセットすることによって (0 を書き込むことによって)、自動ボーレート検出を再開できます。

注： 自動ポーレート操作中に USART が無効化された場合 (UE=0)、BRR 値が破損することがあります。

## 31.5.7 マルチプロセッサ通信

マルチプロセッサ通信では、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの LINEN ビット
- USART\_CR3 レジスタの HDSEL、IREN、および SCEN ビット

USART とのマルチプロセッサ通信が可能です (ネットワーク内で複数の USART を接続して)。たとえば、1 つの USART をマスタとして、その TX 出力を別の USART の RX 入力に接続することができます。別の USART はスレーブであり、それぞれの TX 出力の論理積をとった上でマスタの RX 入力に接続します。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する USART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードにできます。ミュートモード機能を使用するためには、USARTx\_CR1 レジスタの MME ビットをセットする必要があります。

ミュートモードでは、

- 受信ステータスビットはセットできません。
- 受信割り込みはすべて禁止されます。
- USARTx\_ISR レジスタの RWU ビットは 1 にセットされます。特定の条件下では、USARTx\_RQR レジスタの MMRQ ビットを通じて、RWU をハードウェアまたはソフトウェアによって自動的に制御できます。

USART は、USARTx\_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

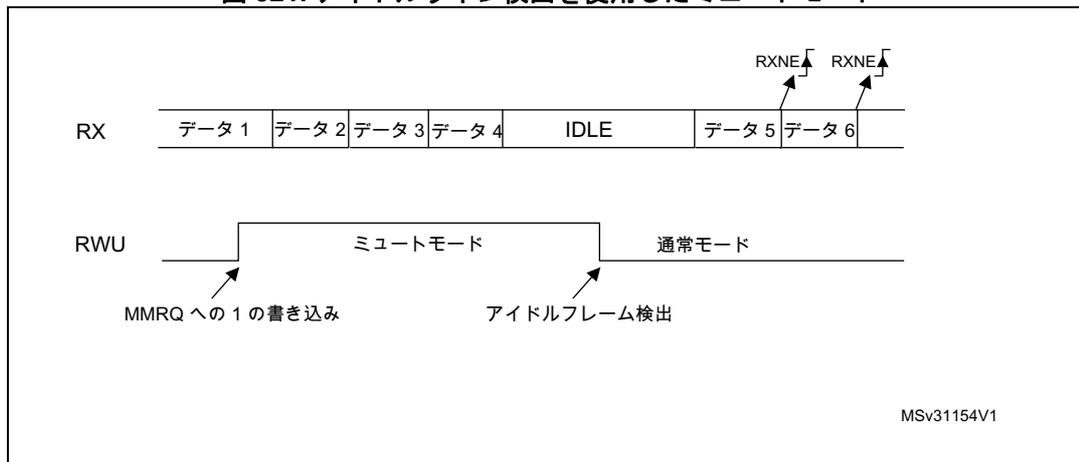
- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

### アイドルライン検出 (WAKE=0)

MMRQ ビットに 1 が書き込まれ、RWU が自動的にセットされたときには、USART はミュートモードに入ります。

USART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、USARTx\_ISR レジスタの IDLE ビットはセットされません。アイドルライン検出を使用したミュートモードの動作例を [図 321](#) に示します。

図 321. アイドルライン検出を使用したミュートモード



**注：** IDLE キャラクタがすでに経過しているときに MMRQ がセットされた場合は、ミュートモードに入りません (RWU はセットされません)。

ラインが IDLE のときに USART が有効にされた場合、1 IDLE フレーム後にアイドル状態が検出されず (1 キャラクタフレームの受信後だけでなく)。

#### 4 ビット/7 ビットアドレスマーク検出 (WAKE=1)

このモードでは、MSB が 1 の場合、バイトはアドレスとして認識され、そうでない場合はデータとみなされます。アドレスバイトのうち、ターゲットレシーバのアドレスは 4 または 7 LSB です。7 または 4 ビットアドレス検出の選択は、ADDM7 ビットを使用して行われます。この 4 ビット/7 ビットワードは、レシーバによって、USARTx\_CR2 レジスタの ADD ビットでプログラムされたレシーバの自己アドレスと比較されます。

**注：** 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) で行われます。

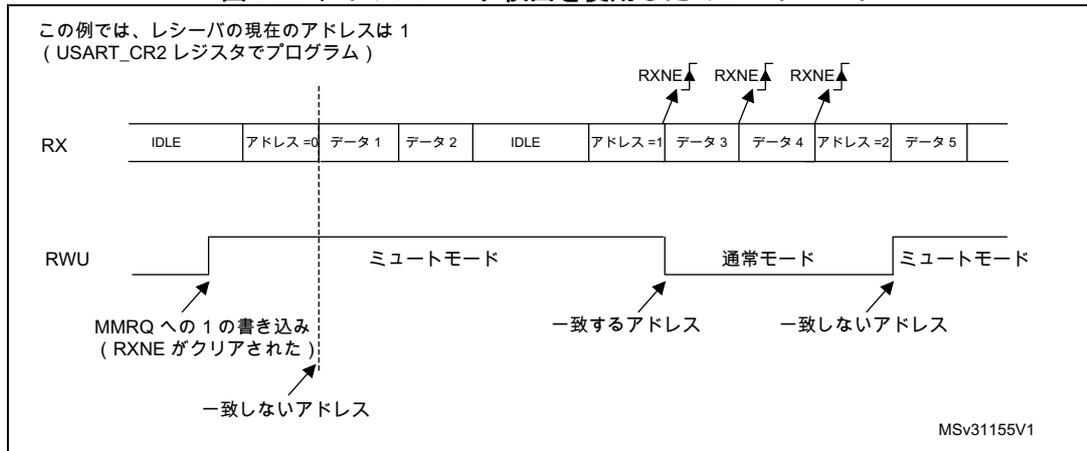
プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、USART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。USART がミュートモードに入ったときには、このアドレスバイトに対して RXNE フラグはセットされず、割り込みも DMA リクエストも発行されません。

MMRQ ビットに 1 が書き込まれたときにも、USART はミュートモードに入ります。この場合、RWU ビットも自動的にセットされます。

プログラムされたアドレスに一致するアドレスキャラクタを受信すると、USART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE ビットがセットされます。

アドレスマーク検出を使用したミュートモードの動作例を [図 322](#) に示します。

図 322. アドレスマーク検出を使用したミュートモード



## 31.5.8 Modbus 通信

USART は、Modbus/RTU および Modbus/ASCII プロトコルの実装に対する基本的サポートを備えています。Modbus/RTU は、半二重のブロック転送プロトコルです。プロトコルの制御部分（アドレス認識、ブロック整合性制御、およびコマンド解釈）は、ソフトウェアで実装する必要があります。

USART は、ソフトウェアに負荷をかけず、他のリソースを使用せずに、ブロック検出の終了に対する基本的サポートを提供します。

### Modbus/RTU

このモードでは、1つのブロックの終了は2キャラクタ時間を超える「サイレンス」（アイドルライン）によって認識されます。この機能は、プログラム可能なタイムアウト機能を通じて実装されます。

タイムアウト機能と割り込みは、USARTx\_CR2 レジスタの RTOEN ビットと、USARTx\_CR1 レジスタの RTOIE を通じて有効にする必要があります。2キャラクタ時間のタイムアウトに対応する値（たとえば、22 x ビット時間）を RTO レジスタでプログラムする必要があります。最後のストップビットの受信後、この時間にわたって受信ラインがアイドルのときには、割り込みが生成されて、現在のブロック受信が完了したことをソフトウェアに知らせます。

### Modbus/ASCII

このモードでは、ブロックの終了は特定の（CR/LF）キャラクタシーケンスによって認識されます。USART は、キャラクター一致機能を使用して、このメカニズムを管理します。

ADD[7:0] フィールドで LF ASCII コードをプログラムし、キャラクター一致割り込みを有効にすることによって（CMIE=1）、LF が受信されたときにソフトウェアに通知し、ソフトウェアは DMA バッファの CR/LF をチェックできます。

## 31.5.9 パリティ制御

パリティ制御（送信中のパリティビット生成と受信中のパリティ検査）を有効にするには、USARTx\_CR1 レジスタの PCE ビットをセットします。M ビットによって定義されたフレーム長に応じて、可能な USART フレームフォーマットを表 171 に示します。

表 171. フレームフォーマット

M ビット	PCE ビット	USART フレーム <sup>(1)</sup>
00	0	SB 8 ビットデータ STB
00	1	SB 7 ビットデータ PB STB
01	0	SB 9 ビットデータ STB
01	1	SB 8 ビットデータ PB STB
10	0	SB 7 ビットデータ STB
10	1	SB 6 ビットデータ PB STB

1. 凡例：SB：スタートビット、STB：ストップビット、PB：パリティビット。データレジスタでは、PB は常に MSB 位置を取ります（M ビットの値に応じて、9、8、または 7 番目）。

### 偶数パリティ

パリティビットは、6、7、または 8 LSB ビット（M ビットの値に応じて）とパリティビットから構成されるフレーム内で「1」の数が偶数になるように計算されます。

たとえば、データ = 00110101 であり、4 ビットがセットされた場合、偶数パリティが選択された場合（USARTx\_CR1 の PS ビット = 0）、パリティビットは 0 になります。

### 奇数パリティ

パリティビットは、6、7、または 8 LSB ビット（M ビットの値に応じて）とパリティビットから構成されるフレーム内で「1」の数が奇数になるように計算されます。

たとえば、データ = 00110101 であり、4 ビットがセットされた場合、奇数パリティが選択された場合（USARTx\_CR1 の PS ビット = 1）、パリティビットは 1 になります。

### 受信中のパリティチェック

パリティチェックに失敗した場合、USARTx\_ISR レジスタの PE フラグがセットされ、USARTx\_CR1 レジスタの PEIE ビットがセットされている場合は割り込みが生成されます。PE フラグは、USARTx\_ICR レジスタの PECF に 1 を書き込むことによってクリアされます。

### 送信中のパリティ生成

USARTx\_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます（偶数パリティが選択された場合（PS=0）は偶数個の「1」、奇数パリティが選択された場合（PS=1）は奇数個の「1」）。

## 31.5.10 LIN (Local Interconnection Network) モード

このセクションは、LIN モードがサポートされるときにのみ適用されます。[セクション 31.4: USART の実装 \(989 ページ\)](#) を参照してください。

LIN モードを選択するには、USARTx\_CR2 レジスタの LINEN ビットをセットします。LIN モードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx\_CR2 レジスタの CLKEN
- USARTx\_CR3 レジスタの STOP[1:0]、SCEN、HDSEL、および IREN

### LIN 送信

LIN マスタ送信の場合、[セクション 31.5.2: トランスミッタ](#) に説明されている手順を適用する必要があります。通常の USART 送信と同じですが、次のような違いがあります。

- 8 ビットのワード長を設定するには M ビットをクリアします。
- LIN モードに入るには、LINEN ビットをセットします。この場合、SBKRQ ビットをセットすると、13 個の "0" ビットがブレークキャラクタとして送信されます。その後、値 "1" の 2 ビットが送信され、次の START 検出が可能になります。

### LIN 受信

LIN モードが有効になると、ブレーク検出回路が有効になります。この検出は、通常の USART レシーバとは完全に独立しています。アイドル状態時やフレームの処理中には、発生のたびにブレークが検出できます。

レシーバが有効になると (USARTx\_CR1 レジスタの RE=1)、RX 入力の START 信号を探します。スタートビットの検出方法は、ブレークキャラクタやデータの検索方法と同じです。スタートビットが検出された後、データの場合とまったく同様に次のビットがサンプリングされます (8、9、および 10 番目のサンプル)。10 個 (USARTx\_CR2 レジスタの LBDL=0) または 11 個 (USARTx\_CR2 レジスタの LBDL=1) の連続したビットが "0" として検出され、その後にデリミタキャラクタが続く場合、USARTx\_ISR レジスタの LBDF フラグがセットされます。LBDIE ビットが 1 の場合、割り込みが生成されます。ブレークを確認する前に、RX ラインがハイレベルに戻ったことを知らせるデリミタが検査されます。

この 10 または 11 が発生する前に "1" がサンプリングされた場合、ブレーク検出回路は現在の検出をキャンセルし、再びスタートビットを検索します。

LIN モードが無効にされた場合 (LINEN=0)、レシーバは、ブレーク検出を考慮することなく、通常の USART として機能し続けます。

LIN モードが有効にされた場合 (LINEN=1)、フレーミングエラーが発生 (つまり、ブレークフレームの場合と同様に、"0" の位置でストップビットを検出) すると、レシーバは停止し、ブレーク検出回路が "1" (ブレークワードが完全でなかった場合) またはデリミタキャラクタ (ブレークが検出された場合) を受信するまで停止状態を維持します。

ブレーク検出回路ステートマシンの動作とブレークフラグを [図 323: LIN モードでのブレーク検出 \(11 ビットブレーク長、LBDL=1\) \(1012 ページ\)](#) に示します。

ブレークフレームの例を [図 324: LIN モードでのブレーク検出とフレーミングエラー検出 \(1013 ページ\)](#) に示します。

図 323. LIN モードでのブレイク検出 (11 ビットブレイク長、LBDL=1)

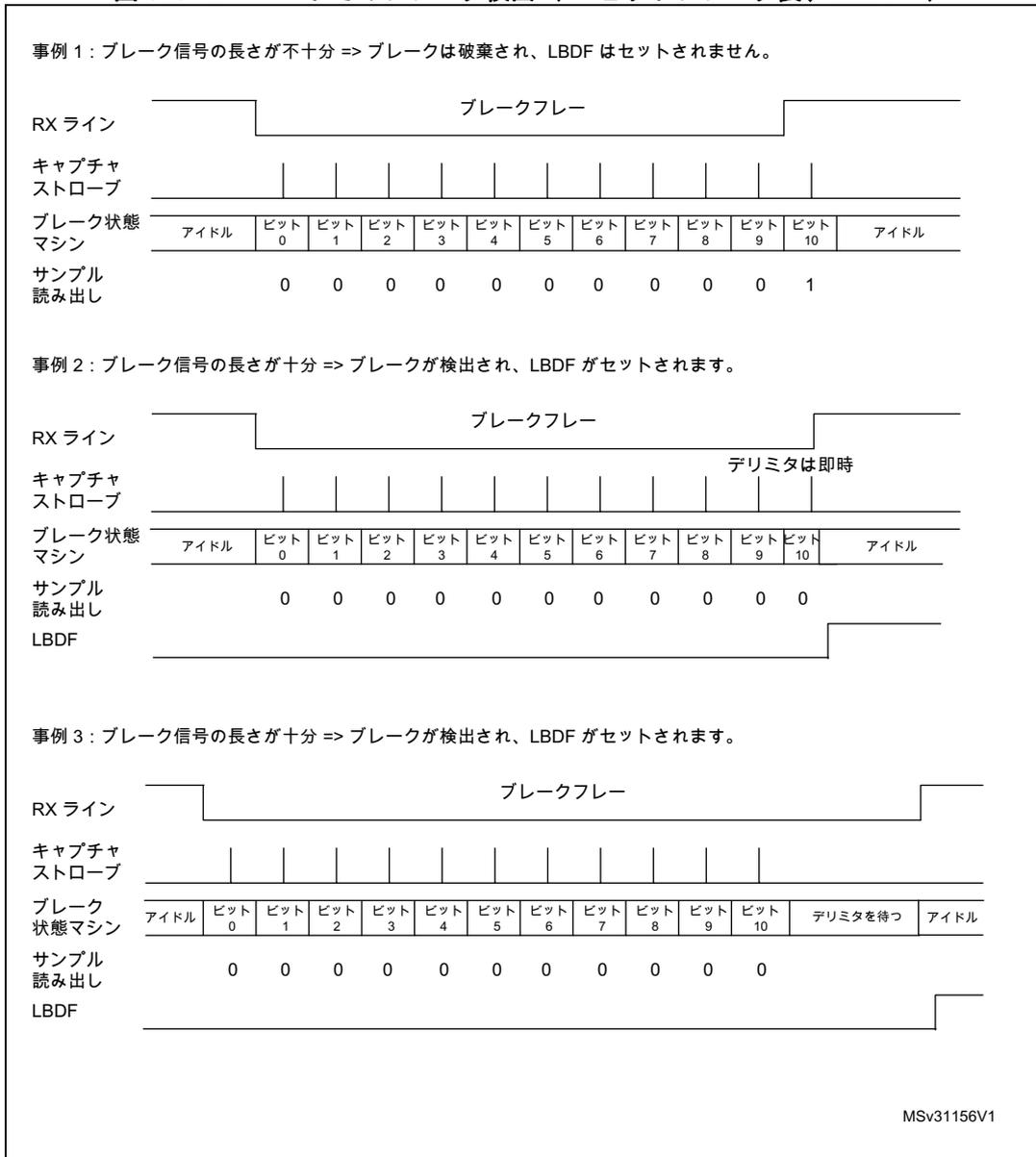
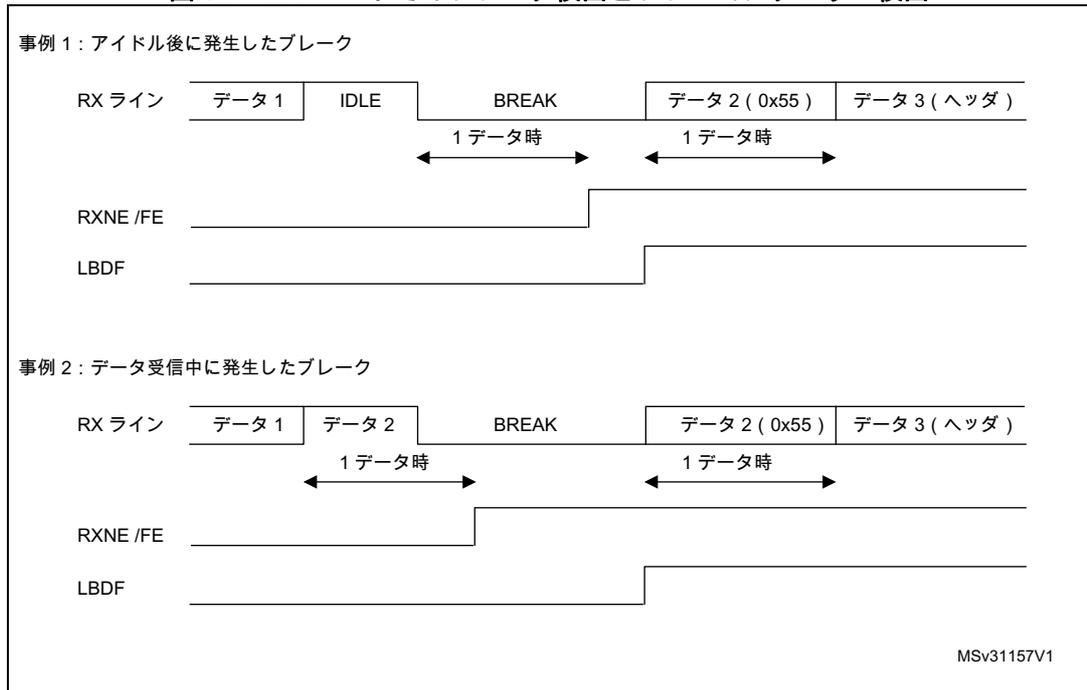


図 324. LIN モードでのブレーク検出とフレーミングエラー検出



## 31.5.11 USART 同期モード

同期モードを選択するには、USARTx\_CR2 レジスタの CLKEN ビットに 1 を書き込みます。同期モードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx\_CR2 レジスタの LINEN ビット
- USARTx\_CR3 レジスタの SCEN、HDSEL、および IREN ビット

このモードでは、USART を使用して、双方向同期シリアル通信をマスタモードで制御できます。SCLK ピンは USART トランスミッタクロックの出力です。スタートビットとストップビットの処理中には、SCLK ピンにクロックパルスは送信されません。USARTx\_CR2 レジスタの LBCL ビットの状態によっては、有効な最後のデータビット（アドレスマーク）の処理中にクロックパルスが生成されることもあります。USARTx\_CR2 レジスタの CPOL ビットは、クロック極性を選択するために使用され、USARTx\_CR2 レジスタの CPHA ビットは、外部クロックの位相を選択するために使用されます（図 325、図 326、および図 327 を参照）。

アイドル時、プリアンブル処理時、およびブレーク送信時には、外部 SCLK クロックは起動されません。

同期モードでは、USART トランスミッタは非同期モードの場合とまったく同じように機能します。しかし、CPOL と CPHA に基づいて SCLK が TX と同期すると、TX 上のデータが同期します。

このモードでは、USART レシーバは非同期モードの場合とは異なる動作をします。RE=1 の場合、データはオーバーサンプリングなしで、SCLK (CPOL と CPHA に応じて立上りまたは立下りエッジ) でサンプリングされます。ポーレート (1/16 ビット持続時間) に応じたセットアップ時間とホールド時間を守る必要があります。

注： SCLK ピンはTX ピンと連携して動作します。したがって、クロックが供給されるのは、トランスミッタが有効であり (TE=1)、データが送信中 (データレジスタ USARTx\_TDR への書き込み) の場合に限られます。つまり、データ送信を行わずに同期データを受信することはできません。  
 USART が無効にされたときには (UE=0)、クロックパルスが正常に機能するように、LBCL、CPOL、およびCPHA ビットを選択する必要があります。

図 325. USART の同期送信の例

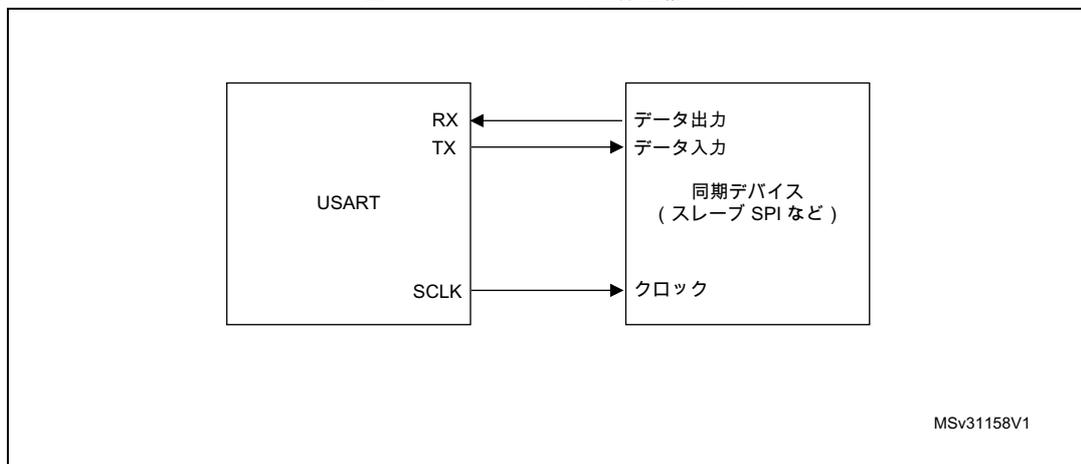


図 326. USART データクロックタイミング図 (M ビット = 00)

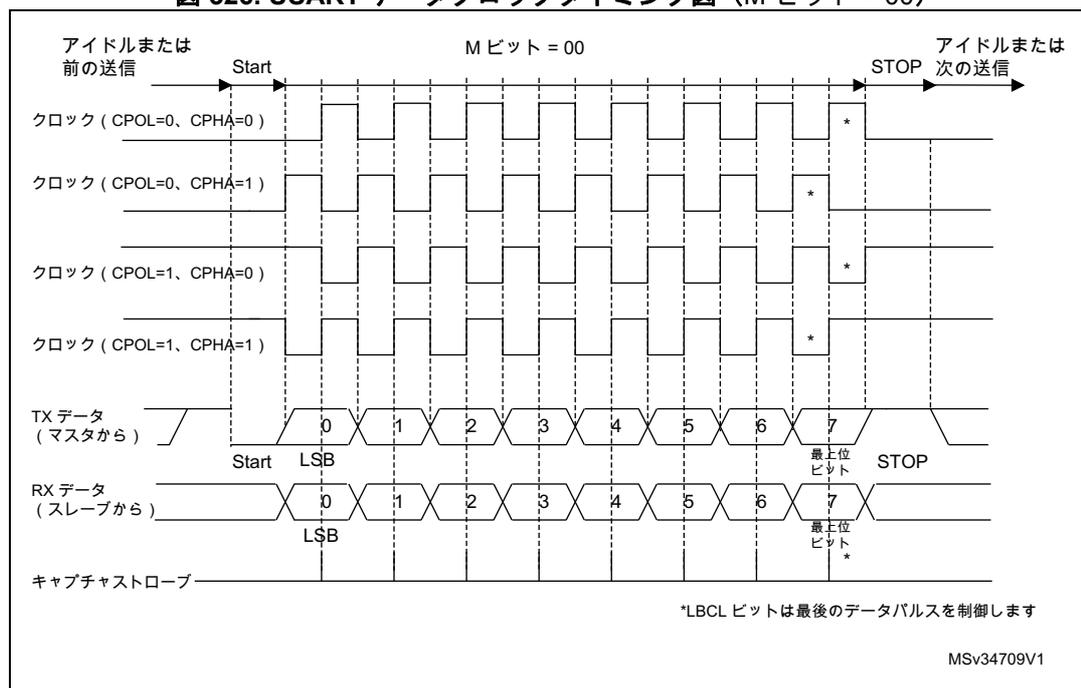


図 327. USART データクロックタイミング図 (M ビット = 01)

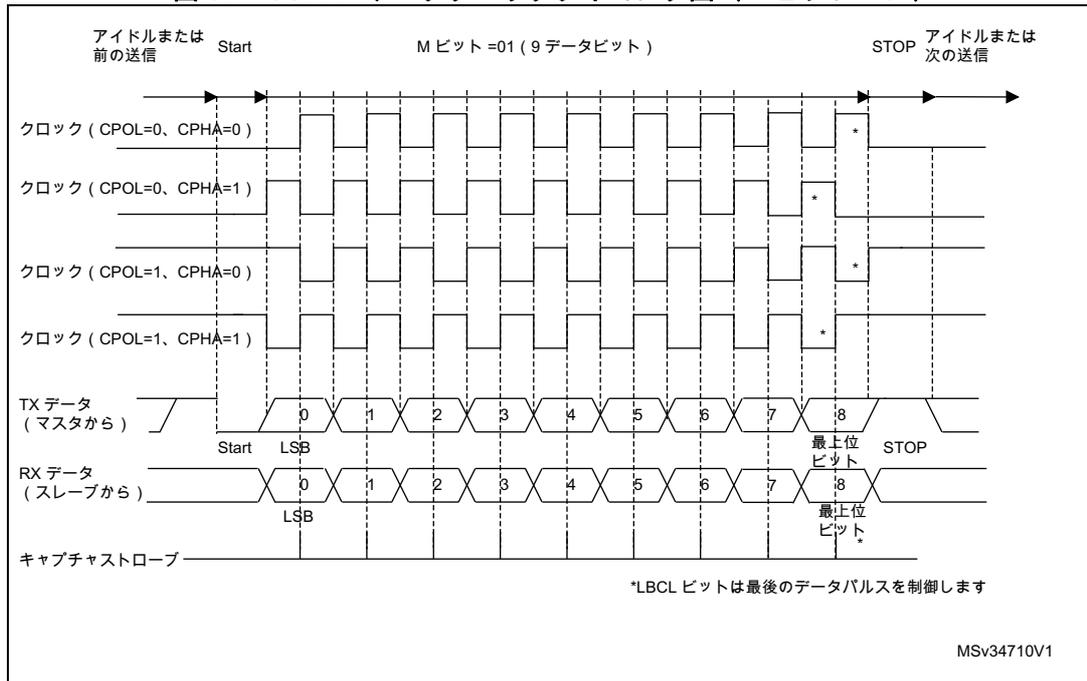
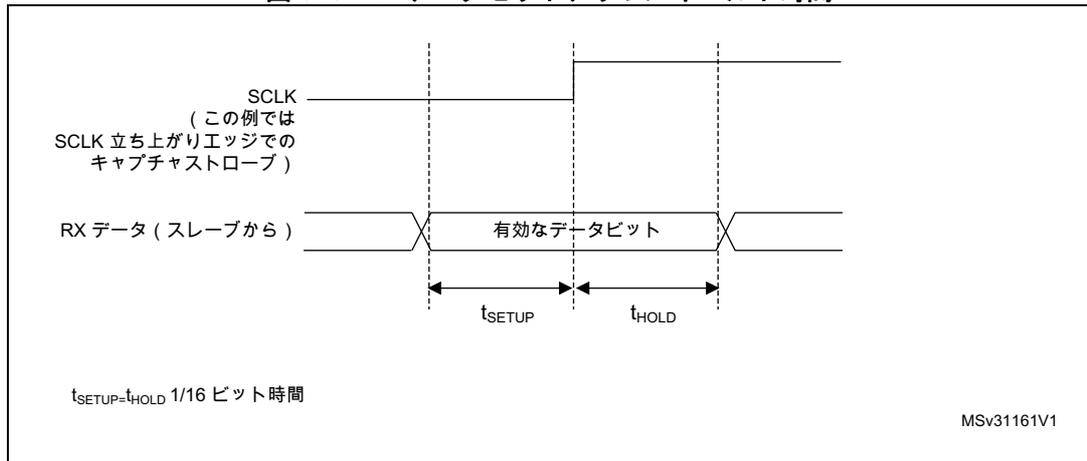


図 328. RX データセットアップ/ホールド時間



注: スマートカードモードでは SCLK の機能が異なります。詳細については、[セクション 31.5.13 : スマートカードモード](#)を参照してください。

## 31.5.12 単線半二重通信

単線半二重モードを選択するには、USARTx\_CR3 レジスタの HDSEL ビットをセットします。このモードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx\_CR2 レジスタの LINEN および CLKEN ビット
- USARTx\_CR3 レジスタの SCEN および IREN ビット

USART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、USART\_CR3 レジスタの制御ビット HDSEL で行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が外部プルアップ付きの代替機能オープンドレインとして設定されるように、I/O を設定する必要があります。

この点を除くと、通信プロトコルは通常の USART モードと同じです。ラインの競合はソフトウェアによって管理する必要があります (たとえば、集中型アービタを使用して)。特に、送信がハードウェアによってブロックされることはなく、TE ビットがセットされている間は、データレジスタにデータが書き込まれるとすぐに、送信が続行されます。

## 31.5.13 スマートカードモード

このセクションは、スマートカードモードがサポートされるときにのみ適用されます。 [セクション 31.4 : USART の実装 \(989 ページ\)](#) を参照してください。

スマートカードモードを選択するには、USARTx\_CR3 レジスタの SCEN ビットをセットします。スマートカードモードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx\_CR2 レジスタの LINEN ビット
- USARTx\_CR3 レジスタの HDSEL および IREN ビット

さらに、スマートカードにクロックを供給するために CLKEN ビットをセットすることもできます。

スマートカードインタフェースは、ISO 7816-3 標準で定義されたスマートカードの非同期プロトコルをサポートするように設計されています。T=0 (キャラクタモード) と T=1 (ブロックモード) の両方がサポートされます。

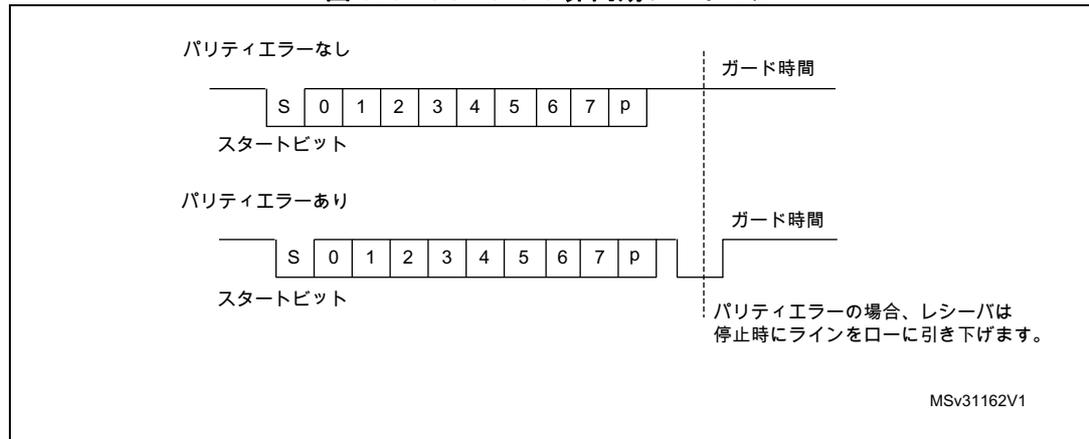
USART は次のように設定してください。

- 8 ビット+パリティ : ワード長が 8 ビットにセットされ、および USARTx\_CR1 レジスタの PCE=1。
- 1.5 個のストップビット : USARTx\_CR2 レジスタの STOP=11。受信にはストップビット 0.5 個を選択することも可能です。

T=0 (キャラクタ) モードでは、パリティエラーはガードタイム中の各キャラクタの終わりに示されます。

[図 329](#) に、パリティエラーの有無によるデータラインの状況の変化の例を示します。

図 329. ISO 7816-3 非同期プロトコル



スマートカードと接続されると、USART の TX 出力は、スマートカードによっても駆動される双方向ラインを駆動します。TX ピンは、オープンドレインとして設定される必要があります。

スマートカードモードは、単線半二重通信プロトコルを実装します。

- 送信シフトレジスタからのデータの送信は、少なくとも 1/2 ボークロックの遅れが保証されません。通常動作では、満杯の送信シフトレジスタは、次のボークロックエッジでシフト動作を開始します。スマートカードモードでは、この送信は、保証された 1/2 ボークロック分だけさらに遅れます。
- 送信時、スマートカードがパリティエラーを検出した場合には、ラインをローに駆動することによって (NACK)、この状態を USART に知らせます。この NACK 信号 (1 ボークロックの間、送信ラインをローに引き下げ) は、1.5 個のストップビットが組み込まれたトランスミッタ側にフレーミングエラーを引き起こします。USART は、プロトコルに従って、データの自動再送信を処理できます。再試行回数は、SCARCNT ビットフィールドでプログラムされます。プログラムされた再試行回数後も USART が NACK を受信し続けた場合は、送信を停止して、エラーをフレーミングエラーとして通知します。TXE ビットは、USARTx\_RQR レジスタの TXFRQ ビットを使用してセットできます。
- 送信時のスマートカード自動再試行: USART による NACK の検出と反復キャラクタのスタートビットの間に 2.5 ボー周期の遅延が挿入されます。最後の反復キャラクタの受信終了時、ただちに TC ビットがセットされず (ガードタイムはありません)。ソフトウェアで再び繰り返したい場合は、規格によって指定されている 2 ボー周期以上を確保する必要があります。
- 1.5 個のストップビット周期でプログラムされたフレームの受信時にパリティエラーが検出された場合、受信フレームの完了後 1 ボークロック周期間、送信ラインがローに引き下げられます。これは、USART に送信されたデータが正しく受信されなかったことをスマートカードに知らせるためです。NACK 制御ビットがセットされている場合、パリティエラーはレシーバによって "NACK" され、そうでない場合、NACK は送信されません (T=1 モードで使用されます)。受信したキャラクタにエラーがあった場合、RXNE/受信 DMA リクエストは有効になりません。プロトコルの仕様に従って、スマートカードは同じキャラクタを再送信する必要があります。SCARCNT ビットフィールドで指定された最大試行回数後も、受信したキャラクタにエラーがあった場合、USART は NACK の送信を停止して、エラーをパリティエラーとして通知します。
- 受信時のスマートカード自動再試行: USART がカードを NACK したが、カードがキャラクタを繰り返さなかった場合、BUSY フラグはセットされたままです。
- 送信時、USART は 2 つの連続するキャラクタの間にガードタイム (ガードタイムレジスタでプログラム) を挿入します。ガードタイムは前のキャラクタのストップビット後に測定されるので、GT[7:0] レジスタを目的の CGT (7816-3 仕様で定義されている Character Guard Time) から 12 (1 キャラクタの時間) を引いた値にプログラムする必要があります。

- TC フラグのアサーションは、ガードタイムレジスタをプログラムすることによって遅らせることができます。通常動作では、TC がアサートされるのは、送信シフトレジスタが空であり、他に未処理の送信リクエストがない場合です。スマートカードモードでは、空の送信シフトレジスタは、ガードタイムカウンタをトリガして、ガードタイムレジスタにプログラミングされた値までカウントアップします。この間、TC は強制的にローレベルに保たれます。ガードタイムカウンタがプログラミングされた値に達すると、TC がハイにアサートされます。
- TC フラグのネゲートは、スマートカードモードの影響を受けません。
- レシーバからの NACK によってトランスミッタ端でフレーミングエラーが検出された場合、トランスミッタの受信ブロックは、この NACK をスタートビットとして検出しません。ISO プロトコルによれば、受信される NACK の期間は 1 または 2 ボークロック周期です。
- レシーバ側では、パリティエラーが検出されて NACK が送信された場合、レシーバはこの NACK をスタートビットとして検出しません。

**注：** スマートカードモードでは、ブレイクキャラクタは意味を持ちません。フレーミングエラー発生時のデータ 0x00 は、ブレイクではなくデータとして処理されます。

TE ビットをトグルするとき、アイドルフレームは送信されません。アイドルフレームは、他の設定では定義されますが、ISO プロトコルでは定義されていません。

図 330 に、USART による NACK 信号のサンプリング方法を示します。この例では、USART はデータを送信中であり、ストップビットが 1.5 個組み込まれています。データと NACK 信号の整合性を検査するために、USART のレシーバ部が有効にされます。

図 330. ストップビット 1.5 個を使用したパリティエラー検出



USART は、SCLK 出力を通じてスマートカードにクロックを供給できます。スマートカードモードでは、SCLK は通信に関係せず、5 ビットのプリスケアラを通じて単に内部のペリフェラル入力クロックから取得されます。この分周比は、プリスケアラレジスタ USARTx\_ で設定されます。SCLK 周波数は、 $f_{CK}/2$  から  $f_{CK}/62$  までの範囲でプログラムでき、 $f_{CK}$  はペリフェラル入力クロックです。

## ブロックモード (T=1)

T=1 (ブロック) モードでは、パリティエラー送信は、USART\_CR3 レジスタの NACK ビットをクリアすることによって無効化されます。

ブロックモードでスマートカードからの読み出しをリクエストするときには、ソフトウェアは USART\_CR2 レジスタの RTOEN ビットをセットすることによってレシーバタイムアウト機能を有効にし、RTOR レジスタの RTO ビットフィールドを BWT (ブロックウェイトタイム) - 11 の値にプログラムする必要があります。この期間が終了する前にカードからの応答が受信されなかった場合、RTOF フラグがセットされ、タイムアウト割り込みが生成されます (USART\_CR1 レジスタの RTOIE ビットがセットされていた場合)。この期間が終了する前に最初のキャラクタが受信された場合は、RXNE 割り込みによって示されます。

**注：** *ブロックモードのスマートカードからの読み出しに USART を DMA モードで使用するときでも、RXNE 割り込みを有効にする必要があります。並行して、DMA は最初の受信バイトの後でのみ有効にする必要があります。*

2 つの連続するキャラクタの間で最大ウェイトタイムの自動チェックを行うには、最初のキャラクタの受信後 (RXNE 割り込み)、RTOR レジスタの RTO ビットフィールドを CWT (キャラクタウェイトタイム) - 11 の値にプログラムする必要があります。この時間は、ボータイム単位で表されます。スマートカードが前のキャラクタの終了後、CWT 未満の時間内に新しいキャラクタを送信しなかった場合、USART は RTOF フラグと割り込み (RTOIE ビットがセットされているとき) によって、これをソフトウェアに通知します。

**注：** *RTO カウンタはカウントを開始します。*

- STOP = 00 の場合、ストップビットの最後から

- STOP = 10 の場合、2 番目のストップビットの最後から

- STOP = 11 の場合、ストップビットの最初から 1 ビット時間後

*スマートカードプロトコルの定義にあるように、BWT/CWT 値は最後のキャラクタの開始 (スタートビット) から定義されます。RTO レジスタは、最後のキャラクタ自体の長さを考慮して、それぞれ BWT - 11 または CWT - 11 にプログラムする必要があります。*

ブロック長カウンタは、USART が受信するすべてのキャラクタをカウントするために使用されます。このカウンタは、USART の送信時 (TXE=0) にリセットされます。ブロックの長さは、スマートカードによってブロックの 3 番目のバイト (プロローグフィールド) で伝えられます。この値を USARTx\_RTOR レジスタの BLEN フィールドでプログラムする必要があります。DMA モードを使用するときには、ブロックの開始前に、このレジスタフィールドを最小値 (0x0) にプログラムする必要があります。この値では、4 番目の受信キャラクタの後に割り込みが生成されます。ソフトウェアは LEN フィールド (3 番目のバイト) を読み出す必要があり、その値は受信バッファから読み出される必要があります。

割り込み駆動受信モードでは、ブロックの長さはソフトウェアによって、または BLEN 値をプログラムすることによってチェックできます。ただし、ブロックの開始前に、BLEN の最大値 (0xFF) をプログラムすることができます。実際の値は、3 番目のキャラクタの受信後にプログラムされます。

ブロックが LRC 水平冗長検査 (1 エピローグバイト) を使用している場合は、BLEN=LEN です。ブロックが CRC メカニズム (2 エピローグバイト) を使用している場合は、BLEN=LEN+1 をプログラムする必要があります。合計ブロック長 (プロローグ、エピローグ、および情報フィールドを含む) は、BLEN+4 に等しくなります。ブロックの終わりは EOPF フラグと割り込み (EOPIE ビットがセットされているとき) によってソフトウェアに通知されます。

ブロック長エラーの場合、ブロックの終わりは RTO 割り込みによって通知されます (キャラクタウェイトタイムオーバーフロー)。

**注：** *エラーチェックコード (LRC/CRC) は、ソフトウェアによって計算/確認されなければなりません。*



## ダイレクトおよびインバースコンベンション

スマートカードプロトコルは、ダイレクトとインバースの2つコンベンションを定義しています。

ダイレクトコンベンションは、LSB ファースト、論理ビットの値 1 がラインの H 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=0、DATAINV=0 (デフォルト値) をプログラムする必要があります。

インバースコンベンションは、MSB ファースト、論理ビットの値 1 が単線の L 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=1、DATAINV=1 をプログラムする必要があります。

**注：** *論理データ値が反転されると (0=H、1=L)、パリティビットも同じように反転されます。*

カードのコンベンションを認識するために、カードは最初のキャラクタ TS を ATR (Answer To Reset) フレームの最初のキャラクタとして送信します。TS には、LHHL LLL LLH と LHHL HHH LLH の2つのパターンがあります。

- (H) LHHL LLL LLH は、インバースコンベンションをセットアップします。状態 L が値 1 にエンコードされ、モーメント 2 は最上位ビットを含みます (MSB ファースト)。インバースコンベンションによってデコードされると、送受信されたバイトは 3F に等しくなります。
- (H) LHHL HHH LLH は、ダイレクトコンベンションをセットアップします。状態 H が値 1 にエンコードされ、モーメント 2 は最下位ビットを含みます (LSB ファースト)。ダイレクトコンベンションによってデコードされると、送受信されたバイトは 3B に等しくなります。

キャラクタパリティは、2 から 10 までの 9 個のモーメントに 1 にセットされた偶数個のビットがあったときに正しいとみなされます。

USART はカードが使用するコンベンションを知らないので、いずれのパターンであるかを認識して、それに応じて動作できる必要があります。パターン認識はハードウェアでは行われず、ソフトウェアシーケンスによって行われます。さらに、USART がダイレクトコンベンション (デフォルト) で設定され、カードがインバースコンベンションで応答した場合、TS = LHHL LLL LLH になり、USART が受信したキャラクタは 03 になり、パリティは奇数です。

したがって、TS パターン認識には 2 つの方式を使用できます。

### 方式 1

USART は、標準スマートカードモード/ダイレクトコンベンションでプログラムされます。この場合、TS パターンの受信によってパリティエラー割り込みと、カードに対するエラー信号が生成されます。

- パリティエラー割り込みは、カードがダイレクトコンベンションで正しく応答しなかったことをソフトウェアに知らせます。ソフトウェアは、USART をインバースコンベンションで再プログラムします。
- エラー信号に対して、カードは同じ TS キャラクタを再試行し、再プログラムされた USART によって今度は正しく受信されます。

または、パリティエラー割り込みに応答して、ソフトウェアは USART を再プログラムし、カードに対して新しいリセットコマンドを生成してから、TS を再び待つこともできます。

### 方式 2

USART は 9 ビット/パリティなしモード、ビット反転なしでプログラムされます。このモードでは、次のように 2 つの TS パターンのいずれかを受信します。

(H) LHHL LLL LLH = 0x103 -> インバースコンベンションを選択

(H) LHHL LLL LLH = 0x103 -> ダイレクトコンベンションを選択

ソフトウェアは受信されたキャラクタをこの 2 つのパターンと照合して、いずれかに一致した場合は、それに応じて、次のキャラクタ受信に備えて USART をプログラムします。

2つのうちのどちらとも認識されなかった場合、ネゴシエーションを再開始するためにカードリセットが生成されます。

## 31.5.14 Ir SIR ENDEC ブロック

このセクションは、IrDA モードがサポートされるときにのみ適用されます。[セクション 31.4:USART の実装 \(989 ページ\)](#) を参照してください。

IrDA モードを選択するには、USARTx\_CR3 レジスタの IREN ビットをセットします。IrDA モードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx\_CR2 レジスタの LINEN、STOP、および CLKEN ビット
- USARTx\_CR3 レジスタの SCEN および HDSEL ビット

IrDA SIR 物理層は、ロジック 0 を赤外光パルスとして表現する RZI (Return to Zero, Inverted) 変調方式の使用を指定します ([図 331](#) を参照)。

SIR 送信エンコーダは、USART からの NRZ (Non Return to Zero) 送信ビットストリーム出力を変調します。出力パルスストリームは、外部の出力ドライバと赤外線 LED に送信されます。SIR ENDEC の場合、USART は最大 115.2 Kbps のビットレートしかサポートしません。通常モードでは、送信されるパルス幅は、ビット周期の 3/16 と指定されます。

SIR 受信デコーダは、赤外線検出回路からの RZ (Return to Zero) ビットストリームを復調し、受信した NRZ シリアルビットストリームを USART に出力します。デコーダの入力は、アイドル状態のノーマルハイレベル (マーク状態) です。送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。デコーダ入力がローレベルのとき、スタートビットが検出されます。

- IrDA は半二重通信プロトコルです。トランスミッタがビジーである場合 (USART が IrDA エンコーダにデータを送信しているとき)、IrDA デコーダは IrDA 受信ライン上にあるすべてのデータを無視します。レシーバがビジーである (USART が IrDA デコーダからデコードされたデータを受信している) 場合、IrDA は USART から IrDA への TX 上のデータをエンコードしません。データの受信中は、送信データの破壊を防ぐために、送信を避けてください。
- 0 はハイパルスとして送信され、1 は 0 として送信されます。通常モードでは、パルスの幅は、選択されたビット周期の 3/16 と規定されます ([図 332](#) を参照)。
- SIR デコーダは、IrDA 準拠の受信信号を USART 用のビットストリームに変換します。
- SIR 受信ロジックは、ハイ状態を論理値 1 とみなし、ローパルスを論理値 0 とみなします。
- 送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。SIR 出力は、アイドル時にロー状態になります。
- IrDA 仕様では、1.41 us より大きなパルスを受け入れる必要があります。受け入れられるパルス幅は、プログラム可能です。レシーバ端のグリッチ検出回路は、PSC 2 周期 (PSC は USARTx\_GTPR でプログラムされたプリスケール値) より小さな幅のパルスをフィルタします。PSC 1 周期より小さな幅のパルスは常に拒否されますが、1 周期以上 2 周期未満の幅のパルスは受け入れられることも、拒否されることもあります。2 周期より大きな幅のパルスは、パルスとして受け入れられます。PSC=0 のとき、IrDA エンコーダ/デコーダは機能しません。
- レシーバは、低電力トランスミッタと通信できます。
- IrDA モードでは、USARTx\_CR2 レジスタのストップビットを「1 ストップビット」に設定する必要があります。

## IrDA 低電力モード

### トランスミッタ

低電力モードでは、パルス幅はビット周期の 3/16 に維持されません。代わりに、パルス幅は低電力ボーレート（最小で 1.42 MHz）の 3 倍となります。

一般に、この値は 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz) です。低電力モードのプログラム可能な分周器は、この値を得るためにシステムクロックを分周します。

### レシーバ

低電力モードでの受信は、通常モードでの受信と同様です。グリッチ検出の場合、USART は 1 PSC よりも短いパルスを破棄する必要があります。有効なローレベルは、その期間が IrDA 低電力ボーレート (USARTx\_GTPR の PSC 値) の 2 周期分を超える場合にのみ受け入れられます。

**注：** PSC 2 周期未満 1 周期以上の幅のパルスは、拒否されることも、拒否されないこともあります。  
レシーバのセットアップ時間は、ソフトウェアで管理してください。IrDA 物理層仕様では、送信と受信の間に最小 10 ms の遅延を指定しています (IrDA は半二重プロトコルです)。

図 331. IrDA SIR ENDEC- ブロック図

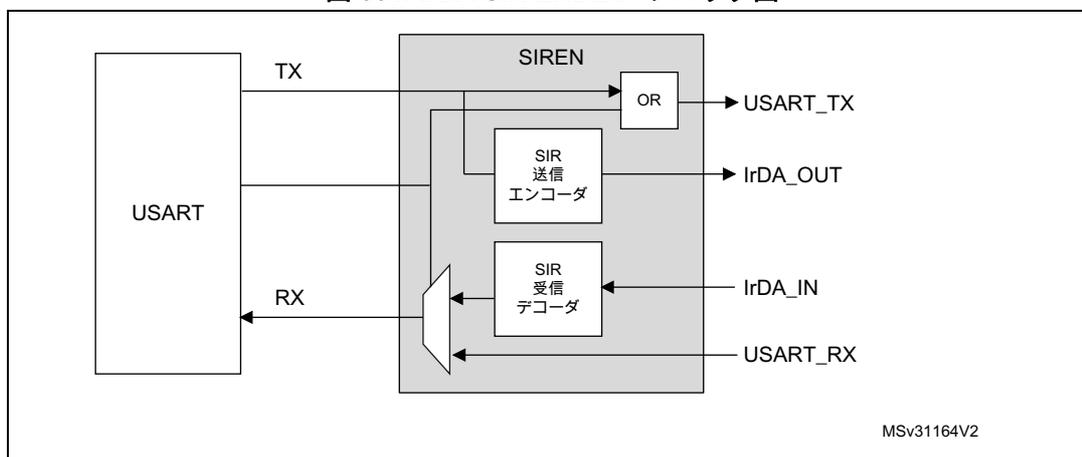
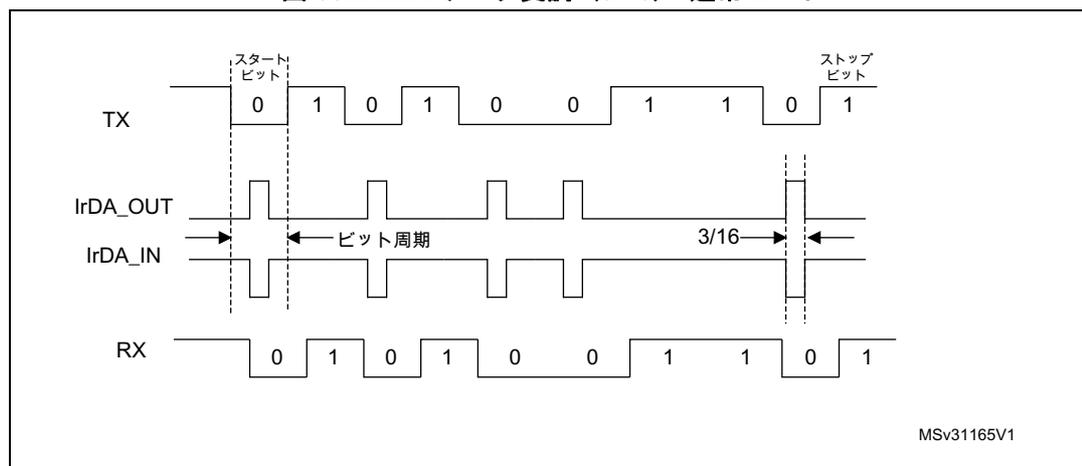


図 332. IrDA データ変調 (3/16) - 通常モード



## 31.5.15 DMA を使用した連続通信

USART は、DMA を使用して連続通信を行うことができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

**注：** *DMA モードがサポートされるかどうかについては、[セクション 31.4 : USART の実装 \(989 ページ\)](#) を参照してください。DMA がサポートされない場合は、[セクション 31.5.2 : トランスミッタ](#) または [セクション 31.5.3 : レシーバ](#) での説明に従って USART を使用してください。連続通信を行うには、USARTx\_ISR レジスタの TXE/RXNE フラグをクリアします。*

### DMA を使用した送信

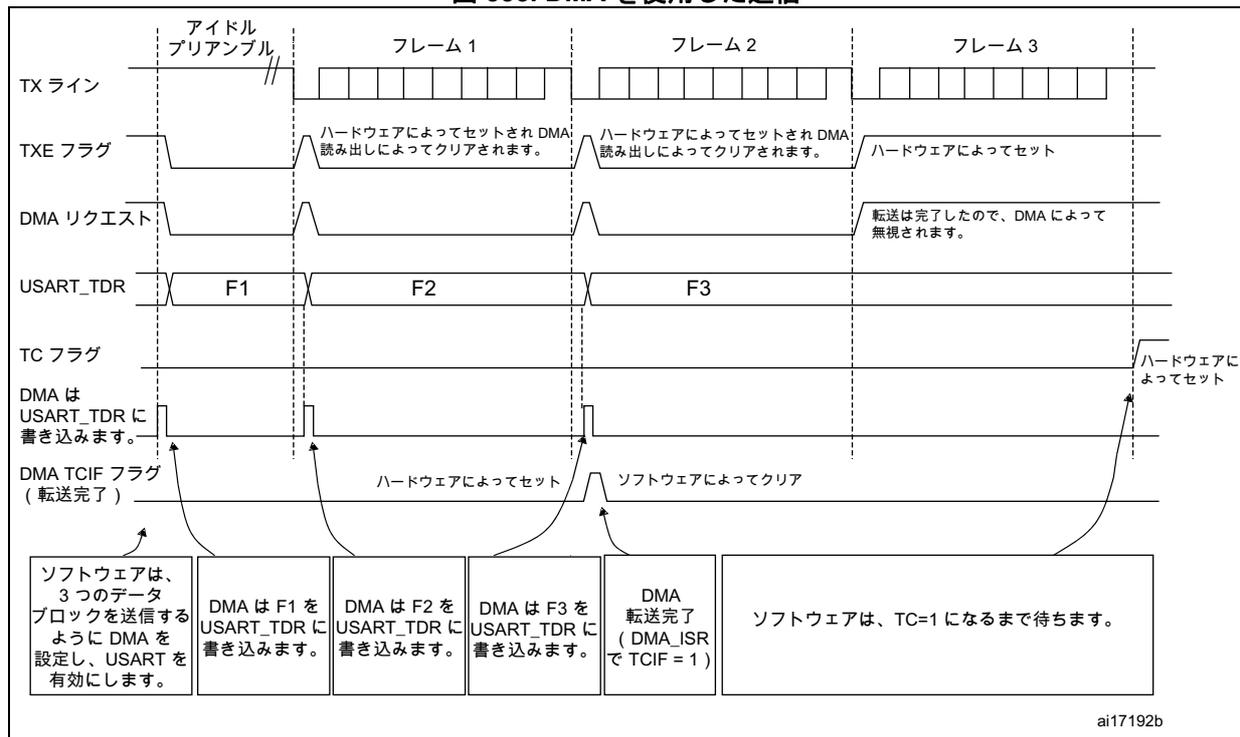
DMA モードでの送信を有効にするには、USARTx\_CR3 レジスタの DMAT ビットをセットします。TXE ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 8 : ダイレクトメモリアクセスコントローラ \(DMA\) \(221 ページ\)](#)) を参照) を使用して設定された SRAM 領域から USARTx\_TDR レジスタにロードされます。DMA チャンネルを USART 送信用に割り付けるには、次の手順を実行します (x はチャンネル番号を示します)

1. DMA 制御レジスタに USARTx\_TDR レジスタのアドレスを書き込み、これを転送先として設定します。データは、各 TXE イベント後にメモリからこのアドレスに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送元として設定します。データは、各 TXE イベント後に、このメモリ領域から USARTx\_TDR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. USARTx\_ICR レジスタの TCCF ビットをセットすることによって、USARTx\_ISR レジスタの TC フラグをクリアします。
7. DMA レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと (DMA\_ISR レジスタの TCIF フラグがセットされます)、TC フラグを観察して USART 通信の完了を確認することができます。これは、USART を無効にしたり STOP モードに入ったりする前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、すべてのデータ転送中、クリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

図 333. DMA を使用した送信



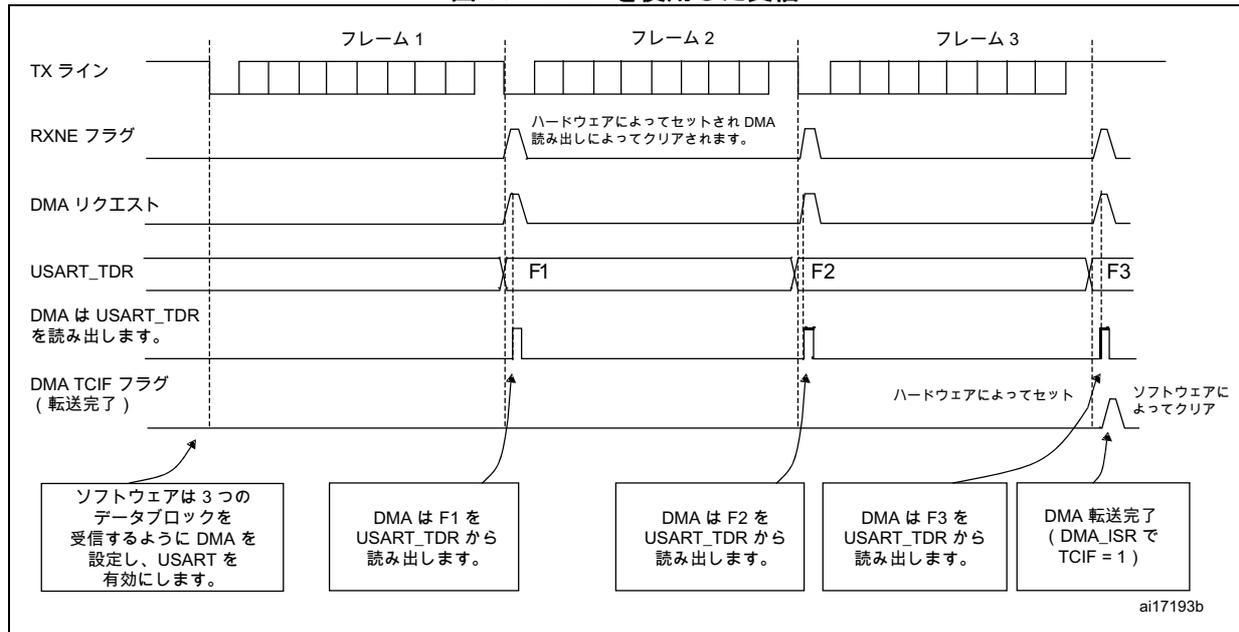
## DMA を使用した受信

DMA モードでの受信を有効にするには、USARTx\_CR3 レジスタの DMAR ビットをセットします。データは、データバイトが受信されると、USARTx\_RDR レジスタから、DMA ペリフェラル ([セクション 8: ダイレクトメモリアクセスコントローラ \(DMA\)](#) を参照) を使用して設定された SRAM 領域にロードされます。DMA チャンネルを USART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに USARTx\_RDR レジスタのアドレスを書き込み、これを転送元として設定します。データは、各 RXNE イベント後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送先として設定します。データは、各 RXNE イベント後に、USARTx\_RDR からこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割り込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

図 334. DMA を使用した受信



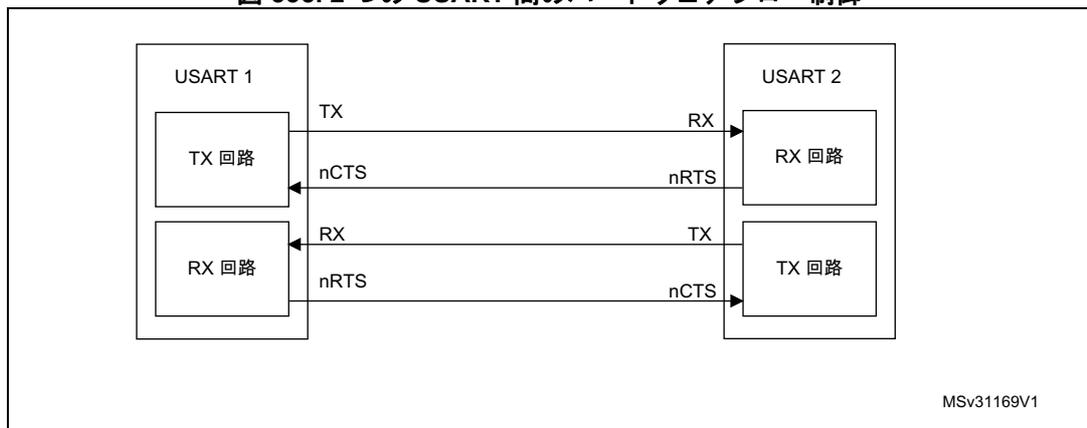
## マルチバッファ通信における割り込み生成とエラーフラグ

マルチバッファ通信でトランザクション中にエラーが発生した場合、現在のバイトの後でエラーフラグがアサートされます。割り込み有効フラグがセットされている場合、割り込みが生成されます。1 バイト受信において RXNE とともにアサートされるフレーミングエラー、オーバーランエラー、およびノイズフラグに関しては、別のエラーフラグ割り込み有効ビット (USARTx\_CR3 レジスタの EIE ビット) があり、これがセットされている場合、いずれかのエラーが発生すると、現在のバイトの後で割り込みが有効になります。

## 31.5.16 RS232 ハードウェアフロー制御および RS485 ドライバ有効

nCTS 入力と nRTS 出力を使用すると、2 つのデバイス間でシリアルデータフローを制御できます。[図 335](#) に、このモードで 2 つのデバイスを接続する方法を示します。

図 335. 2 つの USART 間のハードウェアフロー制御

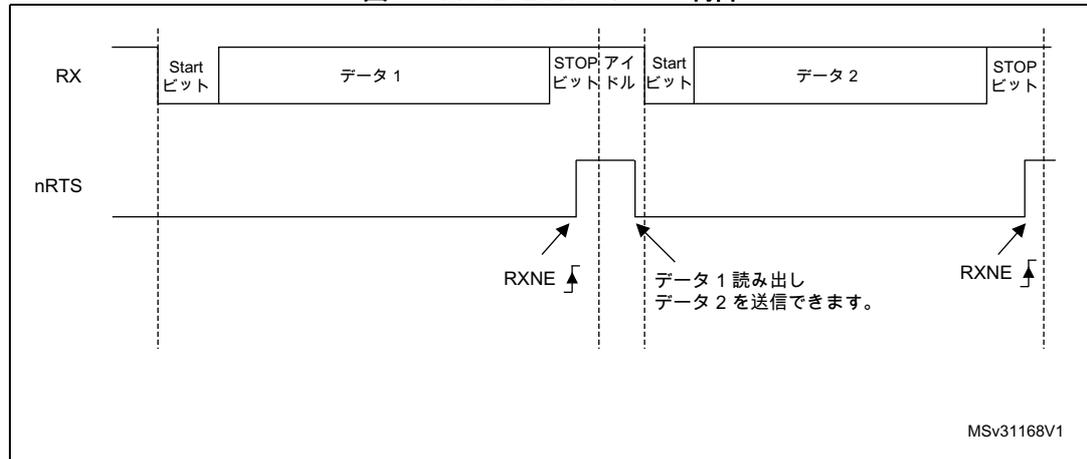


RS232 RTS と CTS のフロー制御は、USARTx\_CR3 レジスタの RTSE ビットと CTSE ビットにそれぞれ 1 を書き込むことによって、個別に有効にできます。

## RS232 RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、USART レシーバが新しいデータを受信可能である限り、nRTS がアサートされます (ローレベル接続)。受信レジスタが満杯になると nRTS がネゲートされ、現在のフレームの終わりに送信が停止する予定であることを示します。図 336 に、RTS フロー制御が有効な場合の通信例を示します。

図 336. RS232 RTS フロー制御

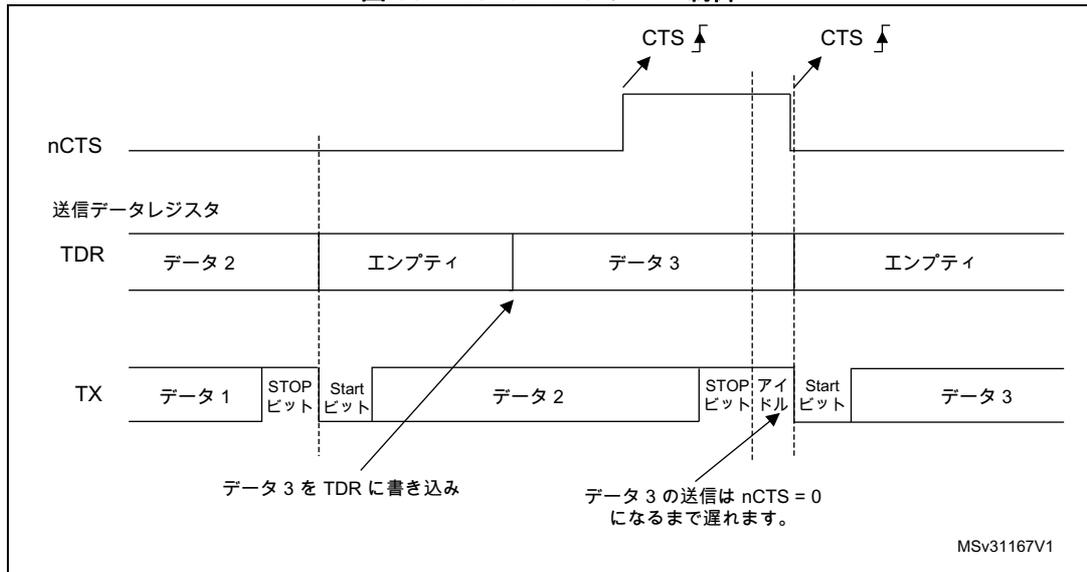


## RS232 CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、nCTS 入力を検査してから、次のフレームを送信します。nCTS がアサートされた場合 (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE=0 の場合)。そうでない場合、送信は行われません。送信中に nCTS がネゲートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、nCTS 入力がトグルすると、CTSIF ステータスビットはハードウェアによって自動的にセットされます。このビットは、レシーバの通信準備ができていないかどうかを示します。USARTx\_CR3 レジスタの CTSIE ビットがセットされている場合、割り込みが生成されます。図 337 に、CTS フロー制御が有効な場合の通信例を示します。

図 337. RS232 CTS フロー制御



注：正しい動作のために、nCTS は、現在のキャラクタの終了の少なくとも 3 USART クロックソース周期前にアサートする必要があります。さらに、2 x PCLK 周期より短いパルスでは CTSCF フラグがセットされない場合があることに注意してください。

### RS485 ドライバ有効

ドライバ有効機能を有効にするには、USARTx\_CR3 制御レジスタのビット DEM をセットします。これにより、DE (Driver Enable) 信号によって外部トランシーバ制御を有効にできます。アサーション時間は、DE 信号の有効化から START ビットの開始までの時間です。USARTx\_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。ネゲート時間は、送信メッセージの最後のストップビットの終了から DE 信号の無効化までの時間です。USARTx\_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。DE 信号の極性は、USARTx\_CR3 制御レジスタの DEP ビットを使用して設定できます。

USART では、DEAT および DEDT はサンプル時間単位（オーバーサンプリングレートに応じて 1/8 または 1/16 ビット持続時間）で表されます。

## 31.6 USART 低電力モード

表 172. 低電力モードが USART に与える影響

モード	説明
SLEEP	影響はありません。USART 割り込みによって、デバイスは SLEEP モードを終了します。
STOP	影響はありません。
STANDBY	USART はパワーダウンされ、デバイスが STANDBY モードを終了したときに再初期化する必要があります。

## 31.7 USART 割り込み

表 173. USART 割り込みリクエスト

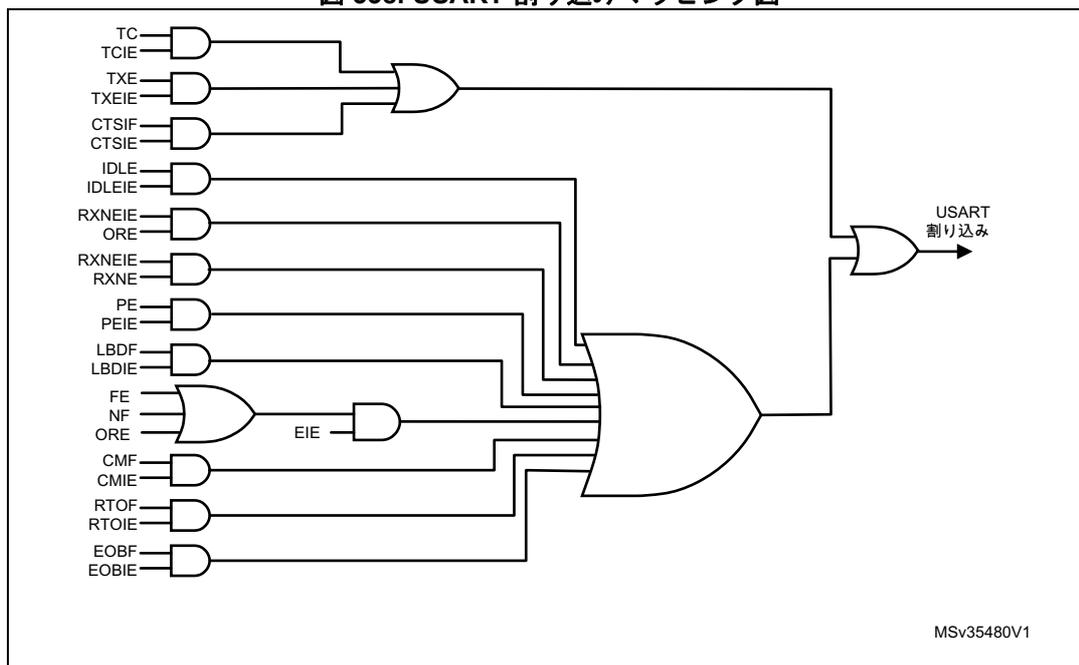
割り込みイベント	イベントフラグ	イネーブル制御ビット
送信データレジスタエンプティ	TXE	TXEIE
CTS 割り込み	CTSIF	CTSIE
送信完了	TC	TCIE
受信データレジスタノットエンプティ (データの読み出し可能)	RXNE	RXNEIE
オーバーランエラー検出	ORE	
アイドルライン検出	IDLE	IDLEIE
パリティエラー	PE	PEIE
LIN ブレーク	LBDF	LBDIE
マルチバッファ通信におけるノイズフラグ、オーバーランエラー、およびフレーミングエラー。	NF または ORE または FE	EIE
キャラクター致	CMF	CMIE
レシーバタイムアウト	RTOF	RTOIE
ブロックの終了	EOBF	EOBIE

USART 割り込みイベントは、同じ割り込みベクトルに接続されます (図 338 を参照)。

- 送信時: 送信完了、Clear to Send、送信データレジスタエンプティ、またはフレーミングエラー (スマートカードモード) 割り込み。
- 受信時: アイドルライン検出、オーバーランエラー、受信データレジスタノットエンプティ、パリティエラー、LIN ブレーク検出、ノイズフラグ、フレーミングエラー、キャラクター致など。

これらのイベントは、対応する有効制御ビットがセットされている場合に割り込みを生成します。

図 338. USART 割り込みマッピング図



## 31.8 USART レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

### 31.8.1 制御レジスタ 1 (USARTx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	M1	EOBIE	RTOIE	DEAT[4:0]				DEDT[4:0]					
			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	Res.	UE
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w

ビット 31:29 予約済みであり、リセット値のままにしておかなければなりません。

ビット 28 **M1** : ワード長

このビットとビット 12 (M0) によって、ワード長が決まります。ソフトウェアによってセット/クリアされます。

M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個

M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個

M[1:0] = 10: スタートビット 1 個、データビット 7 個、ストップビット n 個

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ポーレート (0x7F および 0x55 フレーム検出) はサポートされません。

ビット 27 **EOBIE** : ブロック終了割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの EOBIF フラグがセットされると、USART 割り込みが生成されます。

**注:** USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 26 **RTOIE** : レシーバタイムアウト割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの RTOIF フラグがセットされると、USART 割り込みが生成されます。

**注:** USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ)。

ビット 25:21 **DEAT[4:0]** : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット持続時間) で表されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

**ビット 20:16 DEDT[4:0]** : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット持続時間) で表されます。

DEDT 時間中に USARTx\_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 31.4 : USART の実装 (989 ページ) を参照してください。**

**ビット 15 OVER8** : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **LIN、IrDA、およびモードでは、このビットは常にクリア状態に保つ必要があります。**

**ビット 14 CMIE** : キャラクター致割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの CMF ビットがセットされると、USART 割り込みが生成されます。

**ビット 13 MME** : ミュートモード有効

このビットは、USART のミュートモード機能を有効にします。セットされると、USART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット/クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

**ビット 12 M0** : ワード長

このビットとビット 28 (M1) によって、ワード長が決まります。ソフトウェアによってセット/クリアされます。ビット 28 (M1) の説明を参照してください。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 11 WAKE** : レシーバウェイクアップ方式

このビットによって、ミュートモードからの USART のウェイクアップ方式が決まります。ソフトウェアによってセット/クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 10 PCE** : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合はビット 9、M=0 の場合はビット 8) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 9 PS** : パリティ選択

このビットは、パリティの生成/検出が有効である (PCE ビットがセットされている) とき、奇数パリティ/偶数パリティを選択します。ソフトウェアによってセット/クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

### ビット 8 PEIE : PE 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの PE=1 のときには、USART 割り込みが生成されます。

### ビット 7 TXEIE : 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの TXE=1 のときには、USART 割り込みが生成されます。

### ビット 6 TCIE : 転送完了割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの TC=1 のときには、USART 割り込みが生成されます。

### ビット 5 RXNEIE : RXNE 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの ORE=1 または RXNE=1 のときには、USART 割り込みが生成されます。

### ビット 4 IDLEIE : IDLE 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの IDLE=1 のときには、USART 割り込みが生成されます。

### ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット/クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

**注 :** 送信時、TE ビットに 0 パルス (0 の後に 1) を与えると、現在のワードの後にプリアンブル (アイドルライン) を送信します (スマートカードモードの場合を除きます)。アイドルキャラクタを生成するためには、すぐには TE に 1 を書き込まないでください。必要な時間を確保するために、ソフトウェアは USARTx\_ISR レジスタの TEACK ビットをポーリングできます。

スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

### ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット/クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

### ビット 1 予約済みであり、リセット値に保持する必要があります。

### ビット 0 UE : USART 有効

このビットがクリアされると、USART プリスケアラと出力はただちに停止され、現在の操作は破棄されます。USART の設定は保たれますが、USARTx\_ISR のステータスフラグはすべてそれぞれのデフォルト値にセットされます。このビットは、ソフトウェアによってセット/クリアされます。

0 : USART プリスケアラと出力は無効であり、低電力モードです。

1 : USART は有効です。

**注 :** ラインにエラーを生成せずに低電力モードに入るためには、TE ビットをリセットする必要があります。ソフトウェアは USARTx\_ISR の TC ビットがセットされるのを待ってから、UE ビットをリセットする必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャンネルを無効にする必要があります。

## 31.8.2 制御レジスタ 2 (USARTx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw				

### ビット 31:28 ADD[7:4] : USART ノードのアドレス

このビットフィールドは、認識される USART ノードのアドレスまたはキャラクタコードを指定します。これは、マルチプロセッサ通信において、7 ビットアドレスマーク検出によるウェイクアップのために、ミュートモード時に使用されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。通常の受信時、ミュートモードが無効なときに (たとえば、Modbus プロトコルのブロック終了検出)、キャラクタ検出のために使用することができます。この場合、受信されたキャラクタ全体 (8 ビット) が ADD[7:0] 値と比較され、一致した場合は CMF フラグがセットされます。このビットフィールドは、受信が無効のとき (RE=0) または USART が無効のとき (UE=0) のみ、書き込むことができます。

### ビット 27:24 ADD[3:0] : USART ノードのアドレス

このビットフィールドは、認識される USART ノードのアドレスまたはキャラクタコードを指定します。これは、マルチプロセッサ通信において、アドレスマーク検出によるウェイクアップのために、ミュートモード時に使用されます。このビットフィールドは、受信が無効のとき (RE=0) または USART が無効のとき (UE=0) のみ、書き込むことができます。

### ビット 23 RTOEN : レシーバタイムアウト有効

このビットは、ソフトウェアによってセット/クリアされます。  
 0 : レシーバタイムアウト機能は無効です。  
 1 : レシーバタイムアウト機能は有効です。  
 この機能が有効なとき、RTOR (レシーバタイムアウトレジスタ) でプログラムされた時間にわたって RX ラインがアイドル (受信なし) であった場合、USARTx\_ISR レジスタの RTOF フラグがセットされます。

**注 :** USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

### ビット 22:21 ABRMOD[1:0] : 自動ポーレートモード

これらのビットは、ソフトウェアによってセット/クリアされます。  
 00 : スタートビットの測定がポーレートの検出に使用されます。  
 01 : 立ち上がりエッジから立ち上がりエッジまでの測定。(受信されたフレームはシングルビット = 1 で始まらなければならない、その場合、フレーム = Start10xxxxxx)  
 10 : 0x7F フレーム検出  
 11 : 0x55 フレーム検出  
 このビットフィールドは、ABREN=0 または USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** DATAINV=1 および/または MSBFIRST=1 の場合、パターンはライン上で同じである必要があります (たとえば、MSBFIRST の場合は 0xAA)。USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。



**ビット 20 ABREN** : 自動ポーレート有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 自動ポーレート検出は無効です。

1 : 自動ポーレート検出は有効です。

**注:** *USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

**ビット 19 MSBFIRST** : 最上位ビットファースト

このビットは、ソフトウェアによってセット/クリアされます。

0 : スタートビットに続いて、データはデータビット 0 から順に送受信されます。

1 : スタートビットに続いて、データは MSB (ビット 7/8/9) から順に送受信されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 18 DATAINV** : バイナリデータ反転

このビットは、ソフトウェアによってセット/クリアされます。

0 : データレジスタからの論理データは正/ダイレクトロジックで送受信されます。(1=H、0=L)

1 : データレジスタからの論理データは、負/インバースロジックで送受信されます。(1=L、0=H) パリティビットも反転されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 17 TXINV** : TX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット/クリアされます。

0 : TX ピン信号は標準ロジックレベルを使用して機能します (V<sub>DD</sub>=1/アイドル、Gnd=0/マーク)。

1 : TX ピン信号値は反転されます。(V<sub>DD</sub>=0/マーク、Gnd=1/アイドル)。

これにより、TX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 16 RXINV** : RX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット/クリアされます。

0 : RX ピン信号は標準ロジックレベルを使用して機能します (V<sub>DD</sub>=1/アイドル、Gnd=0/マーク)。

1 : RX ピン信号値は反転されます。(V<sub>DD</sub>=0/マーク、Gnd=1/アイドル)。

これにより、RX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 15 SWAP** : TX/RX ピンのスワップ

このビットは、ソフトウェアによってセット/クリアされます。

0 : TX/RX ピンは標準ピンアウトでの定義に従って使用されます。

1 : TX および RX ピンの機能はスワップされます。これにより、別の USART へのクロスワイヤ接続の場合に動作できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 14 LINEN** : LIN モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : LIN モードは無効です。

1 : LIN モードは有効です。

LIN モードでは、USARTx\_RQR レジスタの SBKRQ ビットを使用して LIN 同期ブ레이크 (下位 13 ビット) を送信し、LIN 同期ブ레이크を検出することができます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** *USART が LIN モードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

**ビット 13:12 STOP[1:0] : STOP ビット**

このビットは、ストップビットのプログラミングに使用します。

00 : 1 個のストップビット

01 : 0.5 個のストップビット

10 : 2 個のストップビット

11 : 1.5 個のストップビット

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 11 CLKEN : クロック有効**

このビットによって、SCLK ピンを有効にできます。

0 : SCLK ピンは無効です。

1 : SCLK ピンは有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 同期モードまたはスマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

**注 :** スマートカードに SCLK クロックを正しく供給するには、次のステップを順守する必要があります。  
- UE = 0  
- SCEN = 1  
- GTPR 設定 (PSC を設定する必要がある場合は、USARTx\_GTPR レジスタへのシングルアクセスで PSC と GT を設定することを推奨)  
- CLKEN = 1  
- UE = 1

**ビット 10 CPOL : クロック極性**

このビットによって、同期モードにおける SCLK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して動作し、希望するクロック/データ関係になるようにします。

0 : 送信ウィンドウの外で、SCLK ピンはローレベルを維持します。

1 : 送信ウィンドウの外で、SCLK ピンはハイレベルを維持します。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

**ビット 9 CPHA : クロック位相**

このビットは、同期モードでの SCLK ピンのクロック出力の位相を選択するために使用されます。CPOL ビットと連携して動作し、希望するクロック/データ関係になるようにします (図 326 および 図 327 を参照)。

0 : 最初のクロック遷移が最初のデータキャプチャエッジです。

1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

**ビット 8 LBCL : 最終ビットのクロックパルス**

このビットは、同期モードで送信される最終データビット (MSB) に関連するクロックパルスを、SCLK ピンに出力する必要があるかどうかを選択するために使用されます。

0 : 最終データビットのクロックパルスは、SCLK ピンに出力されません。

1 : 最終データビットのクロックパルスは、SCLK ピンに出力されます。

**注意 :** 最終ビットは、USARTx\_CR1 レジスタの M ビットによって選択された 7 または 8 または 9 ビットフォーマットに応じて送信された 7 番目または 8 番目または 9 番目のデータビットです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 LBDIE : LIN ブレーク検出割り込み有効

ブレーク割り込みマスクです (ブレークデリミタを使用したブレーク検出)。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの LBDIE=1 のときには、割り込みが生成されます。

**注 :** LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 5 LBDL : LIN ブレーク検出長

このビットでは、10 ビットと 11 ビットのブレーク検出を選択します。

0 : 10 ビットブレーク検出

1 : 11 ビットブレーク検出

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 4 ADDM7 : 7 ビットアドレス検出 / 4 ビットアドレス検出

このビットは、4 ビットアドレス検出と 7 ビットアドレス検出の選択に使用されます。

0 : 4 ビットアドレス検出

1 : 7 ビットアドレス検出 (8 ビットデータモード)

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) に対して行われます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

**注 :** トランスミッタが有効なときには、3 つのビット (CPOL、CPHA、LBCL) に書き込まないでください。

## 31.8.3 制御レジスタ 3 (USARTx\_CR3)

アドレスオフセット : 0x08

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCARCNT[2:0]			Res.	
												rW	rW	rW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	V	V	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:17 **SCARCNT[2:0]** : スマートカード自動再試行カウンタ

このビットフィールドは、スマートカードにおける送受信の再試行回数を指定します。

送信モードでは、送信エラーが生成されるまでの送信の自動再試行回数を指定します (FE ビットをセット)。

受信モードでは、受信エラーが生成されるまでの受信の試行エラー回数を指定します (RXNE および PE ビットをセット)。

このビットフィールドは、USART が無効 (UE=0) のときのみプログラムする必要があります。

USART が有効になると (UE=1)、このビットフィールドは再送信を停止するために 0x0 にのみ書き込み可能です。

0x0 : 再送信無効 - 送信モードでの自動再送信禁止

0x1 から 0x7 : 自動再送信試行回数 (信号エラーの生成前)

**注 :** スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **DEP** : ドライバ有効極性選択

0 : DE 信号はアクティブハイです。

1 : DE 信号はアクティブローです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 14 **DEM** : ドライバ有効モード

このビットにより、DE 信号によって外部トランシーバ制御を有効にできます。

0 : DE 機能は無効です。

1 : DE 機能は有効です。DE 信号は RTS ピンで出力されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 31.4 : USART の実装 (989 ページ)。

**ビット 13 DDRE** : 受信エラー時 DMA 無効

0 : 受信エラーの場合、DMA は無効になります。対応するエラーフラグはセットされますが、RXNE は 0 に保たれ、オーバーランを防ぎます。結果として、DMA リクエストはアサートされないため、エラーのあるデータは転送されず (DMA リクエストなし)、次の正しい受信データが転送されます (スマートカードモードで使用される)。

1 : 受信エラーの後、DMA は無効化されます。対応するエラーフラグと RXNE がセットされます。エラーフラグがクリアされるまで、DMA リクエストはマスクされます。つまり、ソフトウェアは、まず、DMA リクエストを無効にするか (DMAR=0)、RXNE をクリアしてから、エラーフラグをクリアする必要があります。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 受信エラーは、パリティエラー、フレーミングエラー、またはノイズエラーです。

**ビット 12 OVRDIS** : オーバーラン無効

このビットは、受信オーバーラン検出を無効にするために使用されます。

0 : オーバーランエラーフラグ、ORE は、受信データが読み出される前に新しいデータを受信したときにセットされます。

1 : オーバーラン機能は無効です。RXNE フラグがまだセットされている間に新しいデータを受信した場合、

ORE フラグはセットされず、新しく受信されたデータが USARTx\_RDR レジスタの前の内容に上書きされます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** この制御ビットにより、データを読み出さずに通信フローをチェックできます。

**ビット 11 ONEBIT** : 1 サンプルビット方式有効

このビットによって、サンプル方式を選択できます。1 サンプルビット方式が選択されると、ノイズ検出フラグ (NF) が無効になります。

0 : 3 サンプルビット方式

1 : 1 サンプルビット方式

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 10 CTSIE** : CTS 割り込み有効

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタの CTSIF=1 のときには、割り込みが生成されます。

**注 :** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

**ビット 9 CTSE** : CTS 有効

0 : CTS ハードウェアフロー制御が無効です。

1 : CTS モードが有効です。データは nCTS 入力のアサート (0 に関係) されている場合のみ転送されます。データの送信中に nCTS 入力にネゲートされた場合、送信は停止前に完了します。nCTS がネゲートされている間にデータがデータレジスタに書き込まれた場合、nCTS がアサートされるまで送信は延期されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

**ビット 8 RTSE** : RTS 有効

0 : RTS ハードウェアフロー制御が無効です。

1 : RTS 出力は有効であり、レシーババッファにスペースがあるときにのみ、データがリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。データを受信できるとき、nRTS 出力がアサートされます (0 にプルされます)。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** **ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。**

**ビット 7 DMAT** : DMA 有効トランスミッタ

このビットは、ソフトウェアでセット/クリアされます。

1 : DMA モードは送信に有効

0 : DMA モードは送信に無効です。

**ビット 6 DMAR** : DMA 有効レシーバ

このビットは、ソフトウェアでセット/クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。

**ビット 5 SCEN** : スマートカードモード有効

このビットはスマートカードモードを有効にするために使用します。

0 : スマートカードモードが無効です。

1 : スマートカードモードが有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** **USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。**

**ビット 4 NACK** : スマートカード NACK 有効

0 : パリティエラーの際の NACK 転送が無効です。

1 : パリティエラー時の NACK 転送が有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** **USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。**

**ビット 3 HDSEL** : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 2 IRLP** : IrDA 低電力

このビットは、通常と低電力の IrDA モードの選択に使用されます。

0 : 通常モード

1 : 低電力モード

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** **IrDA モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。**

ビット 1 **IREN** : IrDA モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : IrDA は無効です。

1 : IrDA は有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** IrDA モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 0 **EIE** : エラー割り込みイネーブル

エラー割り込み有効ビットは、フレーミングエラー、オーバーランエラー、またはノイズフラグ (USARTx\_ISR レジスタの FE=1 または ORE=1 または NF=1) の場合に割り込み生成を有効にするために必要です。

0 : 割り込みは禁止されています。

1 : USARTx\_ISR レジスタで FE=1、ORE=1、または NF=1 になると、割り込みが生成されます。

## 31.8.4 ボーレートレジスタ (USARTx\_BRR)

このレジスタは、USART が無効 (UE=0) のときのみ書き込むことができます。自動ボーレート検出モードでハードウェアによって自動的に更新されます。

アドレスオフセット : 0x0C

リセット値 : 0x0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **BRR[15:4]**

BRR[15:4] = USARTDIV[15:4]

ビット 3:0 **BRR[3:0]**

OVER8 = 0 のとき、BRR[3:0] = USARTDIV[3:0]。

OVER8 = 1 のとき、

BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。

BRR[3] は、クリアされたままにする必要があります。

## 31.8.5 ガード時間およびプリスケアラレジスタ (USARTx\_GTPR)

アドレスオフセット : 0x10

リセット値 : 0x0000

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:8 **GT[7:0]** : ガード時間値

このビットフィールドは、ガード時間値をポーロック周期数でプログラムするために使用します。これはスマートカードモードで使用します。このガード時間値の後は転送完了フラグがセットされます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

ビット 7:0 **PSC[7:0]** : プリスケアラ値

**IrDA 低電力および IrDA 通常モード :**

PSC[7:0] = IrDA 通常および低電力ポーレート

USART クロックソースを分周して低電力周波数を得るためのプリスケアラのプログラミングに使用します。

ソースクロックは、レジスタに与えられた値 (上位 8 ビット) で分周されます。

00000000 : 予約済み - この値はプログラミングしないでください。

00000001 : クロックソースは 1 で分周されます。

00000010 : クロックソースは 2 で分周されます。

...

**スマートカードモード :**

PSC[4:0] : プリスケアラ値

USART クロックソースを分周してスマートカードのクロックを提供するプリスケアラのプログラミングに使用します。

レジスタで指定された値 (上位 5 ビット) を 2 倍して、ソースクロック周波数の分周比を求めます。

00000 : 予約済み - この値はプログラミングしないでください。

00001 : クロックソースは 2 で分周されます。

00010 : クロックソースは 4 で分周されます。

00011 : クロックソースは 6 で分周されます。

...

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** スマートカードモードが使用される場合、ビット [7:5] はクリアされたままにする必要があります。

スマートカードモードや IrDA モードがサポートされない場合、このビットフィールドは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

## 31.8.6 レシーバタイムアウトレジスタ (USARTx\_RTOR)

アドレスオフセット : 0x14

リセット値 : 0x0000

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

### ビット 31:24 BLEN[7:0] : ブロック長

このビットフィールドは、受信時のスマートカード T=1 のブロック長を指定します。この値は、情報文字の数 + エピログフィールドの長さ (1-LEC/2-CRC) - 1 と等しくなります。

例 :

BLEN = 0 -> 0 情報文字 + LEC

BLEN = 1 -> 0 情報文字 + CRC

BLEN = 255 -> 254 情報文字 + CRC (合計 256 文字)

スマートカードモードでは、TXE=0 のときにブロック長カウンタがリセットされます。

このビットフィールドは、他のモードでも使用できます。この場合、RE=0 (受信無効) のとき、および/または EOBCF ビットが 1 に書き込まれたときに、ブロック長カウンタがリセットされます。

**注 :** この値は、ブロック受信の開始後にプログラムできます (プロログフィールドの LEN 文字のデータを使用)。必ず受信したブロックにつき一度だけプログラムするようにしてください。

### ビット 23:0 RTO[23:0] : レシーバタイムアウト値

このビットフィールドは、レシーバタイムアウト値をビット持続時間の数で指定します。

標準モードでは、最後の受信キャラクタの後、RTO 値を超える間、新しいスタートビットが検出されなかった場合、RTOF フラグがセットされます。

スマートカードモードでは、この値は CWT および BWT を実装するために使用されます。詳細については、スマートカードの章を参照してください。

この場合、タイムアウト測定は最後の受信キャラクタのスタートビットから始めて行われます。

**注 :** この値は、受信キャラクタごとにプログラムされる必要があります。

**注 :** RTOR は、動作中に書き込むことができます。新しい値がカウンタ以下の場合、RTOF フラグがセットされます。

レシーバタイムアウト機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって 0x00000000 に強制的に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。

## 31.8.7 リクエストレジスタ (USARTx\_RQR)

アドレスオフセット : 0x18

リセット値 : 0x0000

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

ビット 31:5 予約済みであり、リセット値のままにしておかなければなりません。

### ビット 4 TXFRQ : 送信データー掃リクエスト

このビットに 1 を書き込むと、TXE フラグがセットされます。

これにより、送信データを破棄できます。このビットは、エラー (NACK) によりデータが送信されなかった場合、および USARTx\_ISR レジスタで FE フラグがアクティブである場合に、スマートカードモードでのみ使用する必要があります。

USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 31.4 : USART の実装 \(989 ページ\)](#) を参照してください。

### ビット 3 RXFRQ : 受信データー掃リクエスト

このビットに 1 を書き込むと、RXNE フラグがクリアされます。

これにより、受信したデータを読み出さずに破棄して、オーバーラン条件を避けることができます。

### ビット 2 MMRQ : ミュートモードリクエスト

このビットに 1 を書き込むと、USART はミュートモードになり、RWU フラグがセットされます。

### ビット 1 SBKRQ : ブレーク送信リクエスト

このビットに 1 を書き込むと、SBKF フラグがセットされ、送信マシンが使用可能になるとすぐに、ラインで BREAK を送信するリクエストが発行されます。

**注 :** アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブレークキャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを待つ必要があります。

### ビット 0 ABRRQ : 自動ポーレートリクエスト

このビットに 1 を書き込むと、USARTx\_ISR の ABRF フラグがリセットされ、次の受信データフレームでの自動ポーレート測定をリクエストします。

**注 :** USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 31.4 : USART の実装 \(989 ページ\)](#) を参照してください。

## 31.8.8 割り込みおよびステータスレジスタ (USARTx\_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0200 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TEACK	Res.	Res.	SBKF	CMF	BUSY						
										r			r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	Res.	EOBF	RTOF	CTS	CTSIF	LBDIF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **TEACK** : 送信有効確認応答フラグ

このビットは、送信有効値が USART によって考慮される時に、ハードウェアによってセット/クリアされます

USARTx\_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **SBKF** : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。USARTx\_RQR レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタは送信されません。

1 : ブレークキャラクタは送信されます。

ビット 17 **CMF** : キャラクター一致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。USARTx\_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx\_CR1 レジスタの CMIE=1 の場合、割り込みが生成されます。

0 : キャラクター一致は検出されていません。

1 : キャラクター一致が検出されました。

ビット 16 **BUSY** : ビジーフラグ

このビットは、ハードウェアによってセット/クリアされます。RX ラインで通信中 (スタートビットの検出時) はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : USART はアイドルです (受信なし)。

1 : 受信中です。

ビット 15 **ABRF** : 自動ポーレートフラグ

このビットは、自動ポーレートがセットされたときにハードウェアによってセットされ (RXNE もセットされ、RXNEIE=1 の場合は割り込みが生成されます)、または、自動ポーレート操作が成功せずに完了したときにセットされます (ABRE=1) (この場合、ABRE、RXNE、および FE もセットされます)。新しい自動ポーレート検出をリクエストするために、USARTx\_RQR レジスタの ABRRQ に 1 を書き込むことによって、ソフトウェアによってクリアされます。

**注 :** USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

**ビット 14 ABRE** : 自動ボーレートエラー

このビットは、ボーレート測定が失敗した場合に、ハードウェアによってセットされます (範囲外のボーレートまたはキャラクタ比較の失敗)。

USARTx\_CR3 レジスタの ABRRQ ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

**注:** *USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

**ビット 13** 予約済みであり、リセット値に保持する必要があります。**ビット 12 EOBF** : ブロック終了フラグ

このビットは、完全なブロックが受信されたときに、ハードウェアによってセットされます (たとえば、T=1 スマートカードモード)。検出は、受信バイト数が BLEN + 4 以上である場合に行われます (ブロックの開始時から、プロローグを含む)。

USARTx\_CR2 レジスタの EOBI=1 である場合、割り込みが生成されます。

USARTx\_ICR レジスタの EOBCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ブロック終了に達していません。

1 : ブロック終了 (文字数) に達しました。

**注:** *スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

**ビット 11 RTOF** : レシーバタイムアウト

このビットは、RTOR レジスタでプログラムされたタイムアウト値が通信なしで経過したときに、ハードウェアによってセットされます。USARTx\_ICR レジスタの RTOCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx\_CR2 レジスタの RTOIE=1 の場合、割り込みが生成されます。

スマートカードモードでは、タイムアウトは CWT または BWT タイミングに対応します。

0 : タイムアウト値に達していません。

1 : データを受信せずにタイムアウト値に達しました。

**注:** *時間が RTOR レジスタでプログラムされた値に等しい場合、2 つのキャラクタが分離され、RTOF はセットされません。この時間がこの値に 2 サンプル時間 (オーバーサンプリング方式によって 2/16 または 2/8) を加えた値を超える場合、RTOF フラグがセットされます。*

*カウンタは RE=0 の場合でもカウントしますが、RTOF は RE=1 のときのみセットされます。RE がセットされたときにタイムアウトがすでに経過していた場合、RTOF はセットされます。*

*USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

**ビット 10 CTS** : CTS フラグ

このビットは、ハードウェアによってセット/クリアされます。nCTS 入力ピンのステータスの反転コピーです。

0 : nCTS ラインはセットされました。

1 : nCTS ラインはリセットされました。

**注:** *ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

**ビット 9 CTSIF** : CTS 割り込みフラグ

このビットは、CTSE ビットがセットされていた場合、nCTS 入力が入力されたときにハードウェアによってセットされます。USARTx\_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

また、USARTx\_CR3 レジスタで CTSIE=1 であれば、割り込みが生成されます。

0 : nCTS ステータスラインでの変更はありません。

1 : nCTS ステータスラインで変更がありました。

**注:** *ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*



**ビット 8 LBDF** : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。USARTx\_ICR レジスタの LBDIE に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx\_CR2 レジスタの LBDIE=1 である場合、割り込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

**注 :** *USART が LIN モードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

**ビット 7 TXE** : 送信データレジスタエンプティ

このビットは、USARTx\_TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。USARTx\_TDR レジスタへの書き込みによってクリアされます。

TXE フラグは、USARTx\_RQR レジスタの TXFRQ に 1 を書き込んでクリアし、データを破棄することもできます (スマートカード T=0 モードでの送信失敗の場合のみ)。

USARTx\_CR1 レジスタの TXEIE ビット =1 の場合、割り込みが生成されます。

0 : データはシフトレジスタに転送されません。

1 : データはシフトレジスタに転送されます。

**注 :** *このビットは、シングルバッファ送信時に使用されます。*

**ビット 6 TC** : 送信完了

データを含むフレームの送信が完了し、TXE がセットされている場合、このビットはハードウェアによってセットされます。USARTx\_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。USARTx\_ICR レジスタの TCCF に 1 を書き込むことによって、または USARTx\_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

USARTx\_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。

0 : 送信は完了していません。

1 : 送信は完了しています。

**注 :** *TE ビットがリセットされ、送信中でなかった場合、TC ビットはただちにセットされます。*

**ビット 5 RXNE** : 読み出しデータレジスタノットエンプティ

このビットは、RDR シフトレジスタの内容が USARTx\_RDR レジスタに転送されると、ハードウェアによってセットされます。USARTx\_RDR レジスタの読み出しによってクリアされます。RXNE フラグは、USARTx\_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。

USARTx\_CR1 レジスタの RXNEIE=1 の場合、割り込みが生成されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

**ビット 4 IDLE** : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。

USARTx\_CR1 レジスタの IDLEIE=1 である場合、割り込みが生成されます。USARTx\_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

**注 :** *RXNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。*

*ミュートモードが有効な場合 (MME=1)、USART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。*

## ビット 3 ORE : オーバーランエラー

このビットは、RXNE=1 のときに、シフトレジスタで現在受信中のデータを RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USARTx\_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx\_CR1 レジスタの RXNEIE=1 または EIE=1 の場合、割り込みが生成されます。

0 : オーバーランエラーはありません。

1 : オーバーランエラーが検出されました。

**注 :** このビットがセットされると、RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割り込みが生成されます。

USARTx\_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。

## ビット 2 NF : START ビットノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。USARTx\_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

**注 :** このビットは、割り込みを生成する RXNE ビットと同時に出現するため、割り込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NF フラグがセットされた場合、割り込みが生成されます。

**注 :** ラインがノイズフリーであるとき、NF フラグを無効にして、ONEBIT ビットに 1 をプログラミングして偏差に対する USART の許容誤差を増加させることができます (セクション 31.5.5 : クロック偏差に対する USART レシーバの許容誤差 (1005 ページ) を参照)。

## ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。USARTx\_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでは、送信時、成功せずに (カードがデータフレームを NACK) 最大送信試行回数に達すると、このビットがセットされます。

USARTx\_CR1 レジスタの EIE=1 の場合、割り込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

## ビット 0 PE : パリティエラー

このビットは、レシーバモードでパリティエラーが発生したときに、ハードウェアによってセットされます。USARTx\_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx\_CR1 レジスタの PEIE=1 の場合、割り込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

## 31.8.9 割り込みフラグクリアレジスタ (USARTx\_ICR)

アドレスオフセット : 0x20

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMCF	Res.
														w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EOBCF	RTOCF	Res.	CTSCF	LBDCF	Res.	TCCF	Res.	IDLECF	ORECF	NCF	FECF	PECF
			w	w		w	w		w		w	w	w	w	w

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CMCF** : キャラクター一致フラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの CMF フラグがクリアされます。

ビット 16:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **EOBCF** : ブロック終了クリアフラグ

このビットに 1 を書き込むと、USARTx\_ISR レジスタの EOBF フラグがクリアされます。

**注:** *USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

ビット 11 **RTOCF** : レシーバタイムアウトフラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの RTOF フラグがクリアされます。

**注:** *USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CTSCF** : CTS フラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの CTSIF フラグがクリアされます。

**注:** *ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

ビット 8 **LBDCF** : LIN ブレーク検出クリアフラグ

このビットに 1 を書き込むと、USARTx\_ISR レジスタの LBDF フラグがクリアされます。

**注:** *LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 31.4 : USART の実装 (989 ページ) を参照してください。*

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TCCF** : 送信完了フラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの TC フラグがクリアされます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **IDLECF** : アイドルライン検出フラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの IDLE フラグがクリアされます。

ビット 3 **ORECF** : オーバーランエラーフラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの ORE フラグがクリアされます。

ビット 2 **NCF** : ノイズ検出フラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの NF フラグがクリアされます。

ビット 1 **FECF** : フレーミングエラーフラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの FE フラグがクリアされます。

ビット 0 **PECF** : パリティエラーフラグクリア

このビットに 1 を書き込むと、USARTx\_ISR レジスタの PE フラグがクリアされます。

## 31.8.10 受信データレジスタ (USARTx\_RDR)

アドレスオフセット : 0x24

リセット値 : 未定義

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **RDR[8:0]** : 受信データ値

受信データキャラクタを含みます。

RDR レジスタは、入カシフトレジスタと内部バスとの間にパラレルインタフェースを提供します (図 314 を参照)。

パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

## 31.8.11 送信データレジスタ (USARTx\_TDR)

アドレスオフセット : 0x28

リセット値 : 未定義

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **TDR[8:0]** : 送信データ値

送信されるデータキャラクタを含みます。

TDR レジスタは、内部バスと出カシフトレジスタとの間にパラレルインタフェースを提供します (図 314 を参照)。

パリティを有効にして (USARTx\_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。

**注:** このレジスタは、TXE=1 のときのみ書き込むことができます。



## 31.8.12 USART レジスタマップ

次の表に、USART のレジスタマップとリセット値を示します。

表 174. USART レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	USARTx_CR1	Res	Res	Res	M1	EOBIE	RTOIE	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDI4	DEDI3	DEDI2	DEDI1	DEDI0	OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	Res	UE	
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USARTx_CR2	ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD1	ABRMOD0	ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res	LBDIE	LBDL	ADDM7	Res	Res	Res	Res	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	USARTx_CR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SCARCNT[2:0]		Res	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSSEL	IRLP	IREN	EIE		
	リセット値													0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	USARTx_BRR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BRR[15:0]																	
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	USARTx_GTPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GT[7:0]							PSC[7:0]									
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	USARTx_RTOR	BLEN[7:0]							RTO[23:0]																									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	USARTx_RQR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ	
	リセット値																												0	0	0	0	0	0
0x1C	USARTx_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TEACK	Res	Res	SBKF	CMF	BUSY	ABRF	ABRE	Res	EOBF	RTOF	CTS	CTSIF	LBDIF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	リセット値												0			0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x20	USARTx_ICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																					EOBCF	RTOCF	Res	CTSCF	LBDICF	Res	TCCF	Res	IDLECF	ORECF	NCF	FECF	PECF
0x24	USARTx_RDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RDR[8:0]									
	リセット値																								X	X	X	X	X	X	X	X	X	X
0x28	USARTx_TDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDR[8:0]									
	リセット値																								X	X	X	X	X	X	X	X	X	X

レジスタ境界アドレスについては、[セクション 2.2 \(66 ページ\)](#) を参照してください。

## 32 シリアルペリフェラルインタフェース/I2S (SPI/I2S)

### 32.1 概要

SPI/I<sup>2</sup>S インタフェースを使用して、SPI プロトコルまたは I<sup>2</sup>S オーディオプロトコルに基づき外部デバイスと通信することができます。SPI または I<sup>2</sup>S モードはソフトウェアによって選択可能です。デバイスのリセット後は、デフォルトで SPI モトローラモードが選択されます。

SPI (シリアルペリフェラルインタフェース) プロトコルは、外部デバイスとの半二重、全二重、および単方向の同期シリアル通信をサポートしています。このインタフェースはマスタとして設定することも可能で、その場合、外部スレーブデバイスに通信クロック (SCK) を供給します。このインタフェースは、マルチマスタ設定で動作することもできます。

I<sup>2</sup>S プロトコルも、同期シリアル通信インタフェースです。スレーブまたはマスタモードで、半二重通信として動作することができます。フィリップス I<sup>2</sup>S 規格、MSB/LSB 詰め規格、PCM規格など、4 つのオーディオ規格に対応できます。

### 32.2 SPI の主な機能

- マスタまたはスレーブ動作
- 3本のラインでの全二重同期転送
- 2本のラインでの半二重同期転送 (双方向データライン有り)
- 2本のラインでの単方向同期転送 (単方向データライン有り)
- 4 ~ 16 ビットのデータサイズ選択
- マルチマスタモード機能
- 8個のマスタモードポーレートプリスケアラ (最大周波数  $f_{PCLK}/2$ )
- スレーブモード周波数 (最大周波数  $f_{PCLK}/2$ )
- マスタとスレーブの両方に対するハードウェア/ソフトウェアによる NSS 管理 : マスタ/スレーブ動作の動的切り替え
- クロックの極性と位相をプログラム可能
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 専用の送受信フラグ (割り込み機能付き)
- SPI バスビジステータスフラグ
- SPI モトローラモードをサポート
- ハードウェア CRC 機能による信頼性の高い通信 :
  - Tx モードでは CRC 値を最終バイトとして送信可能
  - 最終受信バイトに対する CRC エラーの自動チェック
- マスタモードの障害、オーバーランの各フラグ (割り込み機能付き)
- CRC エラーフラグ
- DMA 機能付きの 2 つの 32 ビット内蔵 Rx および Tx FIFO
- SPI TI モードをサポート

## 32.3 I2S の主な機能

- 半二重通信（トランスミッタまたはレシーバのみ）
- マスタまたはスレーブ動作
- 正確なオーディオサンプリング周波数（8～192 kHz）を実現するプログラム可能な 8 ビットのリニアプリスケアラ
- 16、24、または 32 ビットのデータフォーマット
- パケットフレームはオーディオチャネルによって 16 ビット（16 ビットデータフレーム）または 32 ビット（16、24、32 ビットデータフレーム）に固定。
- プログラム可能なクロック極性（定常状態）
- スレーブ送信モードのアンダーランフラグ、受信モード（マスタおよびスレーブ）のオーバーランフラグ、受信モードと送信モード（スレーブの場合のみ）のフレームエラーフラグ
- 送受信用の 16 ビットレジスタ（両チャンネルサイドに対して 1 個のデータレジスタ）
- 以下の I<sup>2</sup>S プロトコルをサポート。
  - フィリップス I<sup>2</sup>S 規格
  - MSB 詰め規格（左詰め）
  - LSB 詰め規格（右詰め）
  - PCM 規格（16 ビットチャンネルフレーム、または 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームでの、ショートおよびロングフレーム同期付き）
- データ方向は常に MSB ファースト。
- 送受信（16 ビット幅）用の DMA 機能
- 外部オーディオコンポーネントを駆動するためのマスタクロックを出力可能。周波数比は、 $256 \times F_S$ （ $F_S$  はオーディオサンプリング周波数）に固定。

## 32.4 SPI/I2S の実装

このマニュアルでは、STM32F75xxx and STM32F74xxx デバイスでの SPI/I2S 実装を示します。

表 175. STM32F75xxx および STM32F74xxx SPI の実装

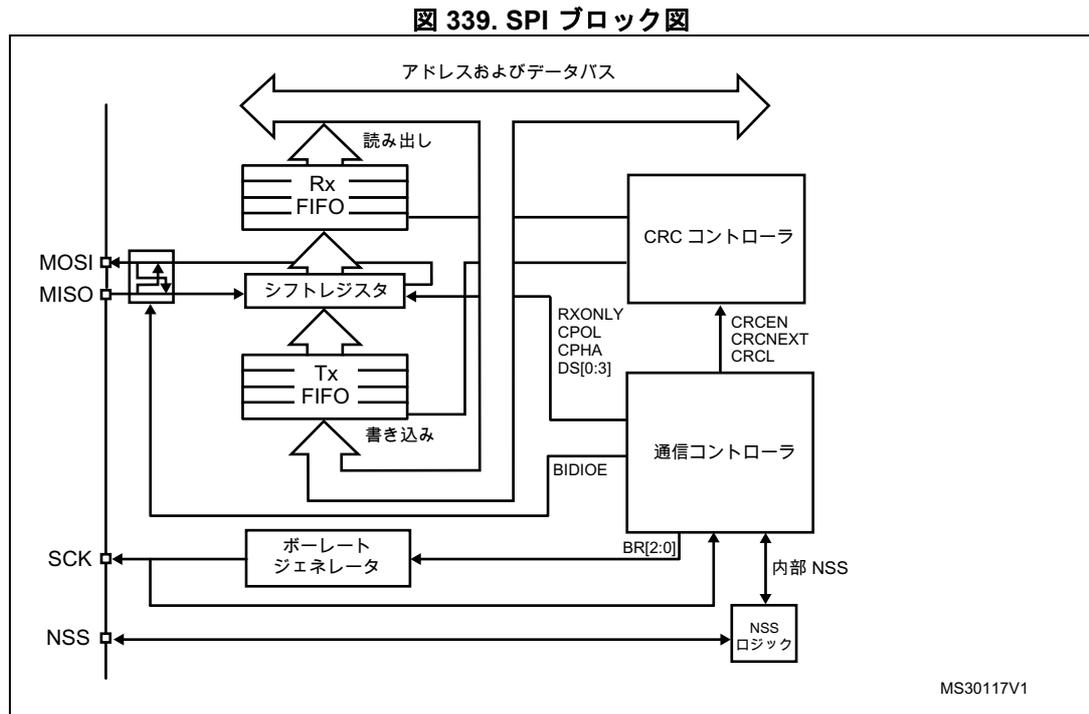
SPI の機能 <sup>(1)</sup>	SPI1	SPI2	SPI3	SPI4	SPI5	SPI6
ハードウェア CRC 計算	X	X	X	X	X	X
Rx/Tx FIFO	X	X	X	X	X	X
NSS パルスモード	X	X	X	X	X	X
I2S モード	X	X	X	-	-	-
TI モード	X	X	X	X	X	X

1. X: サポートされています。

## 32.5 SPI の機能説明

### 32.5.1 概要

SPI では、MCU と外部デバイス間の同期シリアル通信が可能です。アプリケーションソフトウェアは、ステータスフラグをポーリングするか、または専用の SPI 割り込みを使用することで、通信を管理することができます。SPI の主要要素およびそれらの相互作用を以下のブロック図 ( [図 339](#) に示します。



4 本の I/O ピンが外部デバイスとの SPI 通信専用に使われます。

- **MISO** : マスターイン/スレーブアウトデータ。一般に、このピンは、スレーブモードではデータの送信に、マスタモードではデータの受信に使われます。
- **MOSI** : マスターアウト/スレーブインデータ。一般に、このピンは、マスタモードではデータの送信に、スレーブモードではデータの受信に使われます。
- **SCK** : SPI マスタではシリアルクロックの出力に、SPI スレーブでは入力に使われます。
- **NSS** : スレーブ選択用のピンです。このピンは、SPI および NSS の設定に応じて、以下のいずれかに使われます。
  - 個々の通信用スレーブデバイスを選択する
  - データフレームを同期させる
  - 複数のマスタ間での競合を検出する

詳細は、[セクション 32.5.4 : スレーブ選択 \(NSS\) ピンの管理](#) を参照してください。

SPI バスを使用することで、1 つのマスタデバイスと 1 つ以上のスレーブデバイスとの間で通信することができます。バスは 2 本以上の線から成り、1 本はクロック信号用、その他はデータの同期転送用です。SPI ノード間でのデータ交換とそれらのスレーブ選択信号管理に応じて、その他の信号を追加することができます。

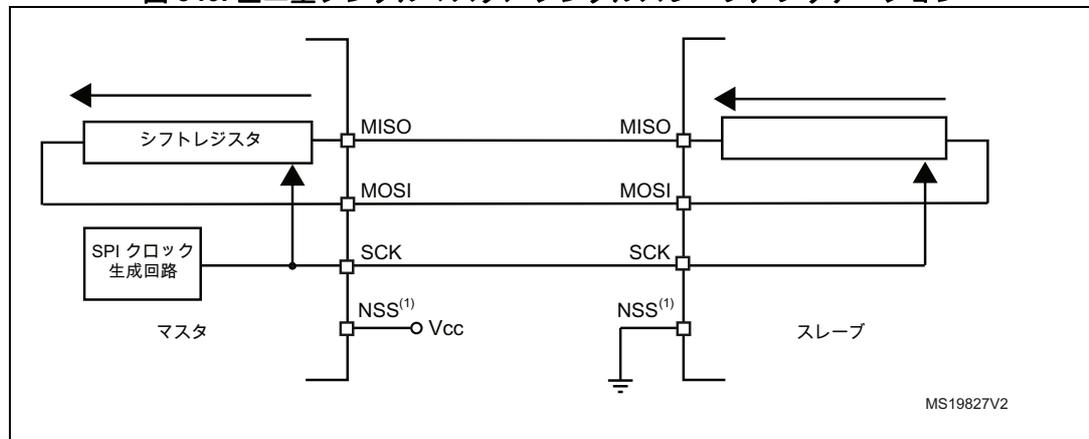
## 32.5.2 マスタとスレーブの 1 対 1 の通信

SPI を使用することで、MCU は対象となるデバイスやアプリケーション要件に応じたさまざまな設定で通信ができます。これらの設定には、2 または 3 本の線（ソフトウェア NSS 管理あり）、あるいは 3 または 4 本の線（ハードウェア NSS 管理あり）が使われます。通信は常にマスタによって開始されます。

### 全二重通信

SPI は、デフォルトで全二重通信に設定されます。この設定では、マスタおよびスレーブのシフトレジスタは、MOSI ピンと MISO ピンの間に 2 本の単方向ラインを介してリンクされます。SPI 通信の間、データはマスタから供給される SCK クロックのエッジに同期してシフトされます。マスタは、送信すべきデータを MOSI ライン経由でスレーブに送信し、MISO ライン経由でスレーブからデータを受信します。データフレーム転送が完了した（すべてのビットがシフトされた）時点で、マスタとスレーブの間で情報が交換されます。

図 340. 全二重シングルマスタ/シングルスレーブアプリケーション

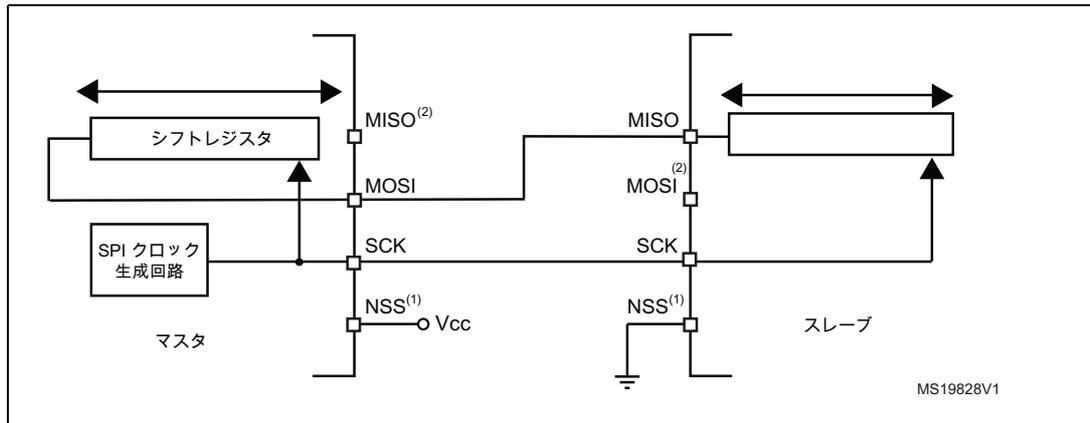


1. この場合、NSS ピンは入力として設定されます。

### 半二重通信

SPIx\_CR1 レジスタの BIDIMODE ビットをセットすることで、SPI は半二重モードで通信できます。この設定では、1 本の交差接続ラインを使用して、マスタとスレーブのシフトレジスタを互いにリンクさせます。この通信中に、データは SCK クロックのエッジに同期して、シフトレジスタ間でシフトされます。シフトの方向は、マスタとスレーブの両方が SPIx\_CR1 レジスタの BDIOE ビットを使用して相互に選択された転送方向となります。この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、他のアプリケーションで使用でき、GPIO として機能します。

図 341. 半二重シングルマスタ/シングルスレーブアプリケーション



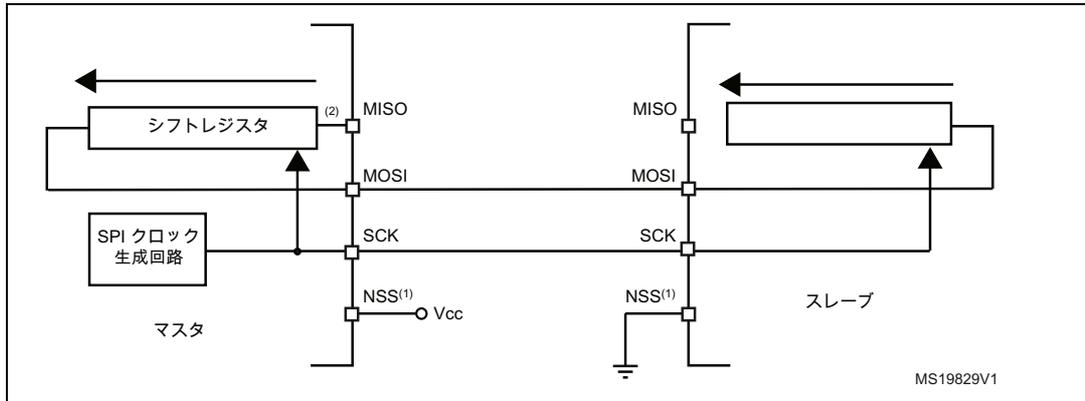
1. この場合、NSS ピンは入力として設定されます。
2. この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、GPIO として使用できます。

## 単方向通信

SPI は、SPIx\_CR2 レジスタの RXONLY ビットを使用して送信専用または受信専用を設定することにより、単方向モードで通信できます。この設定では、マスタとスレーブのシフトレジスタ間の転送に使用するのは 1 ラインのみです。残りの MISO ピンと MOSI ピンのペアは通信には使用されず、標準の GPIO として使用できます。

- **送信専用モード (RXONLY = 0) の場合**：設定は全二重の場合と同じです。アプリケーションは、未使用の入力ピンでキャプチャされた情報を無視する必要があります。このピンは標準の GPIO として使用できます。
- **受信専用モード (RXONLY = 1) の場合**：アプリケーションにて、RXONLY ビットをセットすることによって、SPI 出力機能を無効にできます。スレーブ設定では、MISO 出力が無効化され、ピンを GPIO として使用することができます。スレーブ選択信号がアクティブな間は、スレーブは MOSI ピンからデータを受信し続けます (32.5.4 : [スレーブ選択 \(NSS\) ピンの管理](#)を参照)。データバッファの設定に応じて、受信データイベントが出現します。マスタ設定では、MOSI 出力が無効化され、ピンを GPIO として使用することができます。SPI が有効である間はクロック信号が生成され続けます。クロックを停止させる唯一の方法は、クロックの設定に応じて、RXONLY ビットまたは SPE ビットをクリアし、MISO ピンからの受信パターンが終了し、データバッファ構造への書き込みが行われるまで待つことです。

図 342. 単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ)



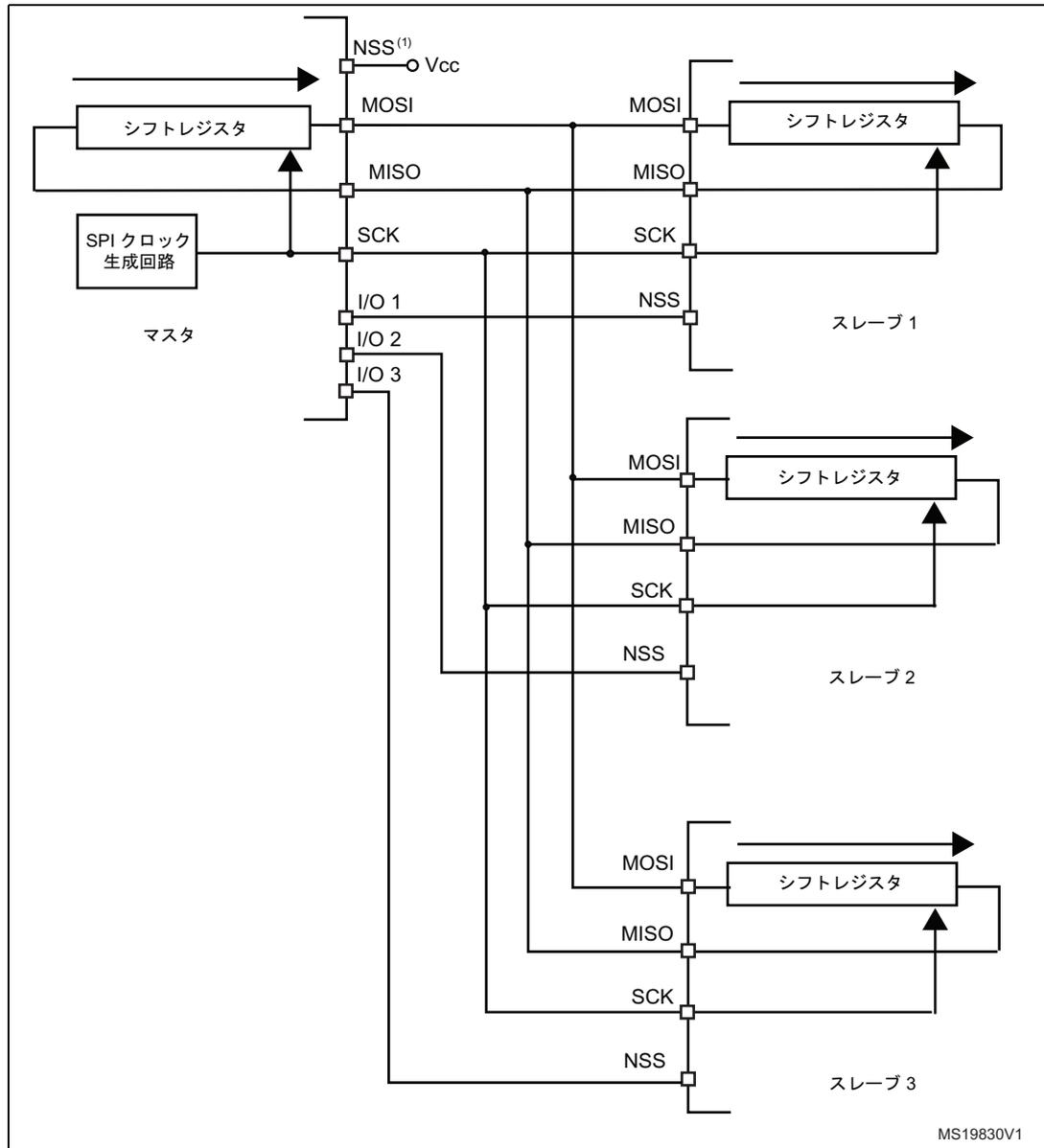
1. この場合、NSS ピンは入力として設定されます。
2. 入力情報はシフトレジスタでキャプチャされますが、標準の送信専用モード (たとえば、OVF フラグ) では無視する必要があります。
3. この設定では、両方のMISO ピンを GPIO として使用できます。

**注：** **すべての単方向通信は、トランザクション方向の設定を固定して (双方向モードはBDIO ビットが変化しない限り有効)、別の半二重通信に置き換えることができます。**

### 32.5.3 標準マルチスレーブ通信

2 つ以上の独立したスレーブがある設定の場合、マスタは GPIO ピンを使用して、各スレーブのチップセレクトラインを管理します (図 343を参照)。マスタは、スレーブの NSS 入力に接続されている GPIO をローレベルにプルダウンすることによって、スレーブの 1 つを選択する必要があります。これを行うことにより、標準マスタと専用スレーブの通信が確立します。

図 343. マスタと 3 つの独立したスレーブ



1. スレーブの MISO ピンは相互接続されているので、すべてのスレーブにおいて、その MISO ピンの GPIO 設定をオルタネート機能オープンドレインとしてセットする必要があります (セクション 6.3.7 : I/O オルタネート機能の入力/出力 (202 ページ) を参照)。

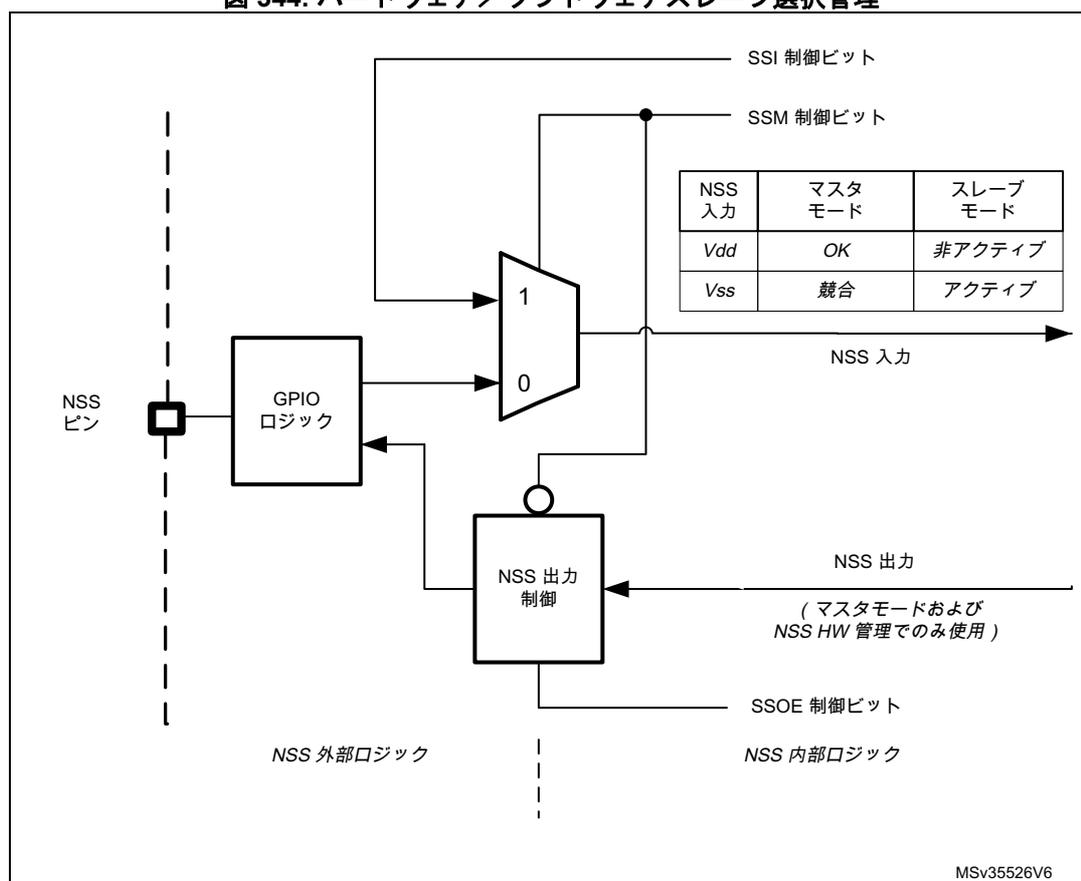
## 32.5.4 スレーブ選択 (NSS) ピンの管理

スレーブモードでは、NSS は標準の「チップセレクト」入力として機能し、スレーブをマスタと通信させます。マスタモードでは、NSS は出力としても入力としても使用できます。入力として使用する場合は、NSS はマルチマスタのバスの衝突を未然に防ぎ、出力として使用する場合は 1 つのスレーブのスレーブ選択信号を駆動させることができます。

ハードウェアまたはソフトウェアのスレーブ選択管理は、SPIx\_CR1 レジスタの SSM ビットを使用して、以下のようにセットすることができます。

- **ソフトウェア NSS 管理 (SSM = 1)** : この設定では、スレーブ選択情報は SPIx\_CR1 レジスタの SSI ビットの値によって内部で駆動されます。外部 NSS ピンは他のアプリケーションで使用できます。
- **ハードウェア NSS 管理 (SSM = 0)** : この場合、2 通りの設定が可能です。次のどちらの設定を使用するかは、NSS 出力設定 (SPIx\_CR1 レジスタの SSOE ビット) によって決まります。
  - **NSS 出力が有効な場合 (SSM = 0, SSOE = 1)** : この設定は、MCU がマスタとしてセットされている場合にのみ使用します。NSS ピンはハードウェアによって管理されます。NSS 信号は、SPI がマスタモードで有効になる (SPE = 1) とすぐにローレベルに駆動され、SPI が無効化される (SPE = 0) までローレベルに保たれます。パルスは、NSS パルスモードが有効になると連続通信間で生成できます (NSSP = 1)。この NSS 設定では、SPI はマルチマスタ設定で機能させることはできません。
  - **NSS 出力が無効な場合 (SSM = 0, SSOE = 0)** : マイクロコントローラがバスでマスタとして機能している場合、この設定によりマルチマスタ機能が可能になります。このモードで、NSS ピンがローレベルにプルダウンされた場合、SPI はマスタモードのフォールト状態に入り、デバイスは自動的にスレーブモードに再設定されます。スレーブモードでは、NSS ピンは標準の「チップセレクト」入力として機能し、NSS ラインがローレベルの間はスレーブが選択されます。

図 344. ハードウェア/ソフトウェアスレーブ選択管理



## 32.5.5 通信フォーマット

SPI 通信中は受信と送信の操作が同時に行われます。シリアルクロック (SCK) は、データライン上で行われる情報のシフトとサンプリングを同期させます。通信フォーマットは、クロック位相、クロック極性、およびデータフレームフォーマットに応じて決定されます。マスタデバイスとスレーブデバイスの通信を可能にするには、双方が同じ通信フォーマットに従う必要があります。

### クロックの位相および極性の制御

SPIx\_CR1 レジスタの CPOL ビットと CPHA ビットを使用することによって、考えられる4つのタイミングの関係をソフトウェアで選択できます。CPOL (クロック極性) ビットは、データが転送されていないときのクロックのアイドル状態の値を制御します。このビットは、マスタモードとスレーブモードの両方に影響を与えます。CPOL がリセットされると、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされると、SCK ピンはハイレベルのアイドル状態になります。

CPHA ビットがセットされると、SCK ピンの2番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがリセットされていれば立ち下がりエッジ、CPOL ビットがセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。CPHA ビットがリセットされている場合、SCK ピンの1番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがセットされていれば立ち下がりエッジ、CPOL ビットがリセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。

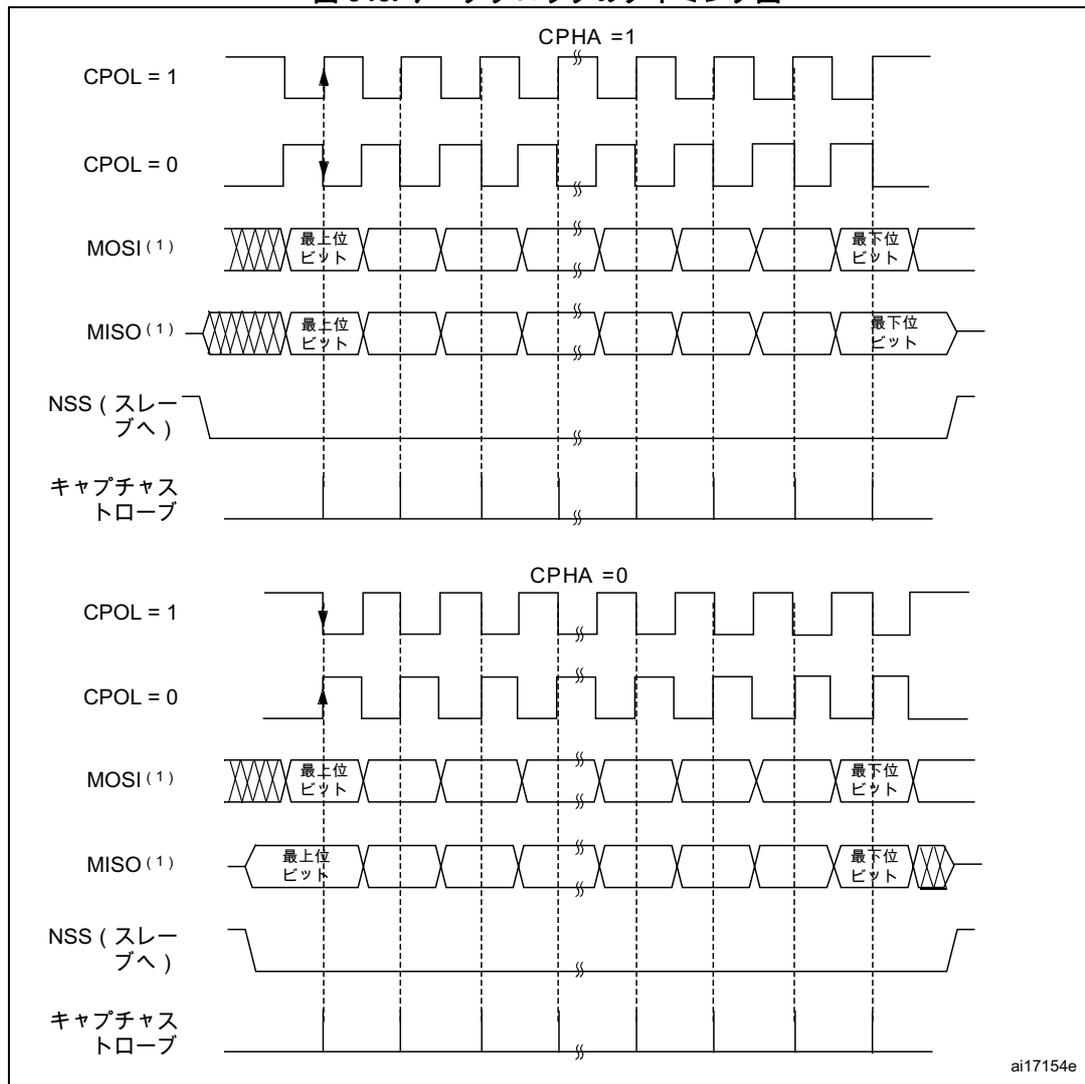
CPOL (クロック極性) ビットと CPHA (クロック位相) ビットの組み合わせによって、データキャプチャのクロックエッジを選択できます。

 345は、CPHA ビットと CPOL ビットの4つの組み合わせによる SPI 全二重転送を示しています。

**注：** CPOL または CPHA ビットを変更する前に、SPE ビットをリセットすることによって、SPI を無効にする必要があります。

SCK のアイドル状態は、SPIx\_CR1 レジスタで (CPOL = 1 なら SCK のプルアップ、CPOL = 0 なら SCK のプルダウンによって) 選択された極性に一致する必要があります。

図 345. データクロックのタイミング図

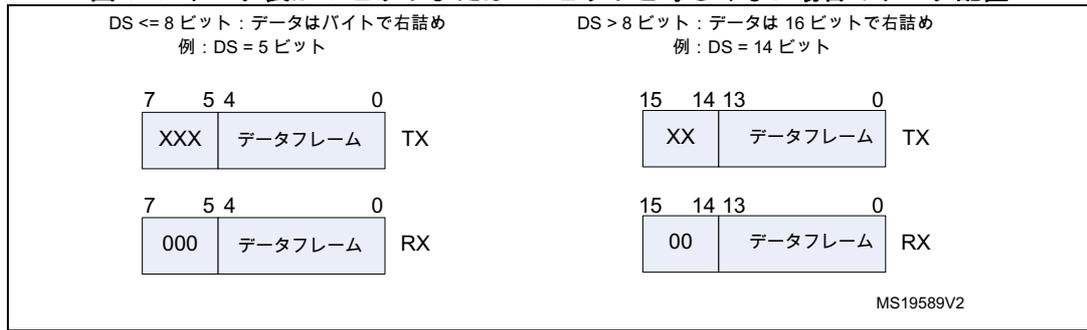


1. データビットの順序は LSBFIRST ビットの設定値に依存します。

## データフレームフォーマット

LSBFIRST ビットの値に応じて、SPI シフトレジスタを設定することで、シフトをMSB ファーストまたは LSB ファーストに設定することができます。データフレームサイズは、DS ビットを使用して選択できます。4 ビットから 16 ビット長までの間にセットでき、この設定は送信と受信の両方に適用できます。選択されたデータフレームサイズにかかわらず、FIFO への読み出しアクセスは FRXTH レベルに整列されている必要があります。SPIx\_DR レジスタにアクセスすると、データフレームはバイト（データがバイトに適合する場合）またはハーフワード（図 346 を参照）のいずれかに常に右詰めされます。通信中、データフレーム内のビットのみがクロック供給され、送信されます。

図 346. データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置



注： 最小データ長は 4 ビットです。4 ビット未満のデータ長が選択された場合は、8 ビットのデータフレームサイズに固定されます。

## 32.5.6 SPI の設定

設定手順は、マスタとスレーブではほぼ同じです。特定のモードの設定については、それぞれのモードに関する章を参照してください。標準通信を初期化する必要があるときは、以下の手順を実行します。

- 適切な GPIO レジスタに書き込みを行います。MOSIピン、MISOピン、SCKピンの GPIO 設定を行います。
- SPI\_CR1 レジスタに書き込みを行います。
  - BR[2:0] ビットを使用して、シリアルクロックポーレートを設定します(注注: 4 を参照)。
  - CPOL ビットと CPHA ビットの組み合わせを設定して、データ転送とシリアルクロックの 4 つの関係のうち 1 つを定義します (CPHA は NSSP モードでクリアする必要があります)。(注注: 2 を参照)。
  - RXONLY または BIDIMODE、および BIDIOE を設定することによって (RXONLY と BIDIMODE は同時にセットできません)、単方向または半二重モードを選択します。
  - LSBFIRST ビットを設定して、フレームフォーマットを定義します (注注: 2 を参照)。
  - CRC が必要な場合は (SCK クロック信号がアイドル状態のとき)、CRCL と CRCEN ビットを設定します。
  - SSM と SSI を設定します (注注: 2 および 3 を参照)。
  - MSTR ビットを設定します (マルチマスタ NSS 設定では、MODF エラーを防ぐためにマスタが設定されている場合、NSS での競合を避けること)。
- 以下のように、SPI\_CR2 レジスタに書き込みを行います。
  - 転送のデータ長を選択するために DS[3:0] ビットを選択します。
  - SSOE を設定します (注注: 1 および 2 および 3 を参照)。
  - TI プロトコルが必要な場合は FRF ビットをセットします (TI モードで NSSP ビットをクリアされたままにする)。
  - 2 つのデータユニット間で NSS パルスモードが必要な場合は NSSP ビットをセットします (NSSP モードで CHPA と TI ビットをクリアされたままにする)。
  - FRXTH ビットを設定します。RXFIFO 閾値は、SPIx\_DR レジスタの読み出しアクセスサイズに揃える必要があります。
  - DMA がバックモードで使用される場合は、LDMA\_TX および LDMA\_RX ビットを初期化します。
- SPI\_CRCPR レジスタに書き込みを行います。必要に応じて CRC 多項式を設定します。
- 適切な DMA レジスタに書き込みを行います。DMA ストリームが使用されている場合は、DMA レジスタに SPI Tx および Rx 専用の DMA ストリームを設定します。

- 注：
- (1) このステップはスレーブモードでは必要ありません。
  - (2) このステップはTI モードでは必要ありません。
  - (3) このステップはNSSP モードでは必要ありません。
  - (4) このステップは、スレーブモードにてTI モードで動作している場合を除き、スレーブモードでは必要ありません。

## 32.5.7 SPI を有効にする手順

マスタがクロックを送信する前に、SPI スレーブを有効にすることを推奨します。さもなければ、望ましくないデータ送信が発生することがあります。スレーブのデータレジスタは、マスタとの通信を開始する前に、送信データをすでに格納していなければなりません（通信クロックの1番目のエッジに、またはクロック信号が連続的なきは現在の通信の最後の前に）。SPI スレーブが有効になる前に、SCK 信号を選択された極性に対応するアイドル状態のレベルに安定させる必要があります。

全二重（または送信専用モード）のマスタは、SPI が有効で TXFIFO がエンプティでない場合、または次の TXFIFO への書き込み時に通信を開始します。

あらゆるマスタ受信専用モードにおいて（RXONLY = 1 または BIDIMODE = 1、および BIDIOE = 0）、SPI が有効になるとすぐに、マスタは通信を開始し、クロックは動作を開始します。

DMA を処理するには、該当する章を参照してください。

## 32.5.8 データの送受信手順

### RXFIFO および TXFIFO

すべての SPI データトランザクションは 32 ビット埋め込み FIFO を通過します。これにより、SPI は連続フローで動作できるようになり、またデータフレームサイズが短い場合にオーバーランを防ぐことができます。各方向にはそれぞれ TXFIFO と RXFIFO と呼ばれる固有の FIFO があります。これらの FIFO は、CRC 計算を有効にした状態で、受信専用モード（スレーブまたはマスタ）を除くすべての SPI モードで使用されます（[セクション 32.5.13 : CRC 計算](#)を参照）。

FIFO の扱いはデータ変換モード（二重、単方向）、データフレームフォーマット（フレーム内のビット数）、FIFO データレジスタで実行されるアクセスサイズ（8 ビットまたは 16 ビット）、および FIFO アクセス時にデータパッキングが使用されるかどうかに従います（[セクション 32.5.12 : TI モード](#)を参照）。

SPIx\_DR レジスタへの読み出しアクセスからは、RXFIFO に保管された、まだ読み出されていない一番古い値が返されます。SPIx\_DR への書き込みアクセスでは、送信キューの最後に TXFIFO に書き込まれるデータを保管します。読み出しアクセスは、必ず SPIx\_CR2 レジスタの FRXTH ビットによって設定された RXFIFO 閾値に揃える必要があります。FTLV[1:0] および FRLVL[1:0] ビットは、両方の FIFO について現在の占有レベルを示します。

SPIx\_DR レジスタへの読み出しアクセスは RXNE イベントで管理する必要があります。このイベントは、データが RXFIFO に保管され、閾値（FRXTH ビットで定義される）に達した場合にトリガされます。RXNE がクリアされると、RXFIFO はエンプティであるとみなされます。同じように、送信するデータフレームの書き込みアクセスは TXE イベントで管理されます。このイベントは、TXFIFO レベルが容量の半分以下である場合にトリガされます。そうでない場合、TXE はクリアされ、TXFIFO がフルであるとみなされます。このように、データフレームフォーマットが 8 ビットを超えない場合、TXFIFO では最大 3 つのデータフレームのみを保管できるのに対し、RXFIFO では最大 4 つを保管することができます。この差異は、ソフトウェアが 16 ビットモードで TXFIFO により多くのデータの書き込みを試みた場合に、すでに TXFIFO に保管されている 3x 8 ビットデータフレームが破損する

可能性を防ぎます。TXE イベントと RXNE イベントの両方を割り込みによってポーリングまたは処理できます。図 348 から 図 351 までを参照してください。

データ交換を管理するもう一つの方法は、DMA を使用することです (セクション 8.2 : DMA の主な機能を参照)。

RXFIFO がフルのときに次のデータを受信した場合、オーバーランイベントが発生します (セクション 32.5.9 : SPI ステータスフラグの OVR フラグの説明を参照)。オーバーランイベントは割り込みによってポーリングまたは処理できます。

セットされる BSY ビットは、現在のデータフレームの進行中のトランザクションを示します。クロック信号が流れ続けているときは、BSY フラグはマスタ側のデータフレーム間でセットされたままになりますが、スレーブの各データフレーム転送間の 1 つの SPI クロックの最小時間においてローレベルになります。

## シーケンス処理

いくつかのデータフレームを単一シーケンスに渡してメッセージを完成させることができます。送信が有効な場合、シーケンスはマスタ側の TXFIFO に何らかのデータが存在する場合に開始し、その間続行します。TXFIFO がエンプティになるまで、マスタによってクロック信号が供給され続け、その後追加のデータを待つことを停止します。

半二重 (BIDIMODE=1、BIDIOE=0) または単方向 (BIDIMODE=0、RXONLY=1) の受信専用モードでは、SPI が有効化され、受信専用モードがアクティブ化されると、直ちにマスタによってシーケンスが開始されます。マスタによってクロック信号が供給されますが、この信号はマスタが SPI または受信専用モードを無効にするまで停止しません。マスタは、クロック信号が停止するまでデータフレームを受信し続けます。

マスタはあらゆるトランザクションを連続モードで供給できる (SCK 信号は連続的) 一方で、データフローおよびその内容をいつでも処理できるスレーブ機能を優先する必要があります。必要に応じて、マスタは通信速度を下げ、より低速のクロックか、または十分な遅延を含む個別のフレーム/データセッションを供給する必要があります。次の 2 点に注意してください。SPI モードのマスタまたはスレーブに対するアンダーフローエラー信号はありません。また、スレーブからのデータは、たとえスレーブがそれらのデータを時間内に適切に準備できない場合でも、常にマスタによってトランザクション処理されます。スレーブが DMA を使用することが特に望ましいのは、データフレームが短く、バスが高速の場合です。

マルチスレーブシステムにおいて通信用のスレーブを 1 つだけ選択するには、各シーケンスを NSS パルス内に収める必要があります。単一のスレーブシステムでは NSS によってスレーブを制御する必要はありませんが、スレーブを各データ転送シーケンスの開始と同期させるために、ここにもパルスを供給することをお勧めします。NSS は、ソフトウェアとハードウェアの両方で管理できます (セクション 32.5.4 : スレーブ選択 (NSS) ピンの管理を参照)。

セットされた BSY ビットは、進行中のデータフレームトランザクションを示します。専用のフレームトランザクションが終了すると、RXNE フラグが立てられます。最後のビットは単にサンプリングされ、すべてのデータフレームが RXFIFO に保管されます。

## SPI を無効にする手順

SPI を無効にする場合は、本項に記載されている無効化手順に従ってください。この手順は、ペリフェラルクロックが停止し、システムが低電力モードに入る前に行うことが重要です。この場合、進行中のトランザクションが破壊されることがあります。モードによっては、この無効化手順が連続通信を停止させる唯一の方法です。

全二重または送信専用モードでは、マスタは、転送するデータの供給を停止した時点でいかなるトランザクションも終了することができます。この場合、クロックは最後のデータトランザクション後に停止します。奇数のデータフレームがダミーバイトの変換を防ぐためにトランザクション処理された

場合、パッキングモードでは特に注意する必要があります (データパッキングセクションを参照)。これらのモードで SPI を無効化する前に、ユーザは標準的な無効化手順に従う必要があります。フレームトランザクションの進行中または次のデータフレームを TXFIFO に保管しているときにマスタトランスミッタで SPI を無効化した場合、SPI の動作は保証されません。

マスタがいずれかの受信専用モードに設定されている場合、連続クロックを停止する唯一の方法は SPE=0 にしてペリフェラルを無効にすることです。これは、最初のビットのサンプリング時間の間、および最後のビット転送が開始される前における最後のデータフレームトランザクション内の特定の時間枠内で発生する必要があります (予測されるデータフレームの総数を受信し、最後の有効データフレーム後に追加の「ダミー」データの読み出しを防ぐため)。このモードで SPI を無効にするには、特定の手順に従う必要があります。

SPI が無効の場合、受信済みで読み出されていないデータは RXFIFO に保管されたままになり、次に SPI を有効にする際に、新しいシーケンスを開始する前に処理する必要があります。未読のデータを防ぐには、SPI を無効にする際に RXFIFO がエンptyであることを確認してください。これは、正しい無効化手順を使用して、またはペリフェラルリセット専用の固有レジスタを制御してソフトウェアリセットを行い、すべての SPI レジスタを初期化することで確認できます (RCC\_APB1RSTR レジスタの SPIIRST ビットを参照)。

標準的な無効化手順は、送信セッションが完全に終わったかどうかをチェックするために、FTLV[1:0] および BSY フラグの状態をプルして行われます。このチェックは、たとえば以下に示すように、進行中のトランザクションの終わりを識別する必要があるような特別な場合にも行うことができます。

- NSS 信号がソフトウェアによって管理されており、マスタはスレーブに適切な NSS パルスの終わりを提供する必要がある場合
- 最後のデータフレームまたは CRC フレームのトランザクションがまだペリフェラルバスで行っている間に、DMA または FIFO からのトランザクションのストリームが完了した場合

正しい無効化手順を以下に示します (受信専用モードが使用されている場合を除く)。

1. FTLVL[1:0] = 00 (送信するデータがなくなる) まで待ちます。
2. BSY = 0 (最後のデータフレームが処理される) まで待ちます。
3. SPI を無効にします (SPE = 0)。
4. FRLVL[1:0] = 00 (受信したデータをすべて読み出す) までデータを読み出します。

受信専用モードの場合の正しい無効化手順を以下に示します。

1. 最後のデータフレームの進行中に特定の時間枠内で SPI を無効 (SPE = 0) にすることにより、受信フローへの割り込みを行います。
2. BSY = 0 (最後のデータフレームが処理される) まで待ちます。
3. FRLVL[1:0] = 00 (受信したデータをすべて読み出す) までデータを読み出します。

**注:** *パッキングモードが使用され、8 ビット以下 (1 バイトに適合) のフォーマットの奇数のデータフレームを受信する必要がある場合は、FRLVL[1:0] = 01 のときに FRXTH をセットしなければなりません。これにより、最後の奇数データフレームを読み出し、良好な FIFO ポインタの整列を維持するための RXNE イベントが生成されます。*

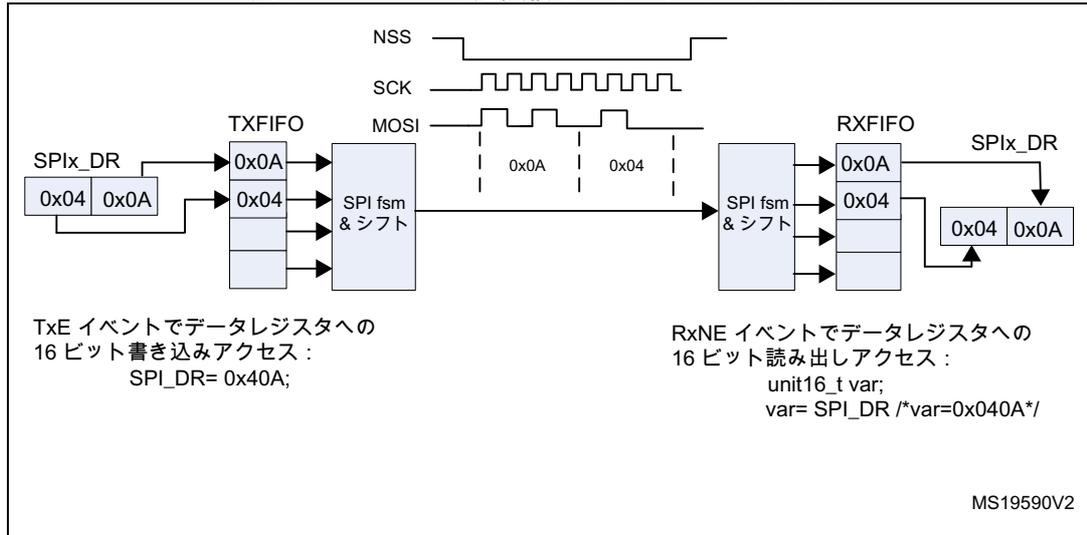
## データパッキング

データフレームサイズが 1 バイト (8 ビット以下) に適合する場合、SPIx\_DR レジスタで 16 ビットの読み出しまたは書き込みアクセスが実行されたときに、自動的にデータパッキングが使用されます。この場合、二重データフレームパターンが並行して処理されます。最初に、SPI はアクセスしたワードの LSB に保管されたパターンを使用して動作します。次に、MSB に保管された残り半分を使用して動作します。図 347 に、データパッキングモードのシーケンス処理の例を示します。2 つのデータフレームは、トランスミッタの SPIx\_DR レジスタに単一の 16 ビットがアクセスした後で送信されます。このシーケンスでは、RXFIFO 閾値が 16 ビット (FRXTH=0) にセットされている場合に、レシーバで 1 つの RXNE イベントのみを生成できます。この場合、この単一の RXNE イベントへの

レスポンスとして、レシーバは SPIx\_DR の単一の 16 ビット読み出しにより、両方のデータフレームにアクセスする必要があります。RxFIFO 閾値設定および以下の読み出しアクセスは、常にレシーバ側に整理させる必要があります。整理されていない場合は、データを失う可能性があります。

これらの奇数の「1 バイトに適合」するデータフレームを処理する必要がある場合、特定の問題が発生します。トランスミッタ側では、SPIx\_DR への 8 ビットアクセスを持つ任意の奇数シーケンスの最後のデータフレームを書き込むだけで十分です。レシーバでは、RXNE イベントを生成するために、フレームの奇数シーケンスで受信した最後のデータフレームの Rx\_FIFO 閾値レベルを変更する必要があります。

図 347. FIFO での送受信のデータのパッキング



- 例：データサイズ DS[3:0] は、CPOL=0、CPHA=1、および LSBFIRST =0 を設定した状態で 4 ビット。有効ビットがバスでのみ実行される場合、LSB バイトの内容が最初に処理される場合、トランスミッタ側で未使用のビットが考慮される場合、およびレシーバ側でゼロでパディングされる場合、データ保存は常に右詰めです。

## DMA (ダイレクトメモリアクセス) を使用する通信

最高速度で動作し、オーバーランを回避するために必要なデータレジスタの読み出し/書き込み処理を容易にするために、SPI は簡単なリクエスト/確認応答プロトコルを実現する DMA 機能を備えています。

SPIx\_CR2 レジスタの TXE または RXNE イネーブルビットをセットすると、DMA アクセスがリクエストされます。TxバッファとRxバッファには、別々のリクエストを発行する必要があります。

- 送信では、TXE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx\_DR レジスタに書き込みます。
- 受信では、RXNE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx\_DR レジスタを読み出します。

図 348 から 図 351 までを参照してください。

SPI がデータの送信にのみ使用される場合、SPI Tx DMA チャンネルのみを有効にすることができます。この場合、受信したデータは読み出されないため、OVR フラグがセットされます。SPI がデータの受信にのみ使用される場合、SPI Rx DMA チャンネルのみを有効にすることができます。

送信モードで、DMA がすべての送信データを書き込んだとき (DMA\_ISR レジスタの TCIF フラグがセットされます)、BSY フラグを監視することで SPI 通信の完了を確認できます。最後の送信内容の破壊を避けるために、SPI を無効にする前、または STOP モードに入る前にこの操作を行う必要があります。ソフトウェアは、まず FTLVL[1:0] = 00 になるまで待つから、BSY=0 になるまで待つ必要があります。

DMA を使用して通信を開始する場合、DMA チャンネルの管理によるエラーイベントを防ぐために、以下の手順を順に実行する必要があります。

1. DMA Rx が使用される場合、SPI\_CR2 レジスタの RXDMAEN ビットの DMA Rx バッファを有効にします。
2. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを有効にします。
3. DMA Tx が使用される場合、SPI\_CR2 レジスタの TXDMAEN ビットにて DMA Tx バッファを有効にします。
4. SPE ビットをセットして、SPI を有効にします。

通信を終了するには、以下の手順を順に実行する必要があります。

1. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを無効にします。
2. SPI 無効化手順に従って SPI を無効にします。
3. DMA Tx および (または) DMA Rx が使用されている場合、SPI\_CR2 レジスタの TXDMAEN ビットおよび RXDMAEN ビットをクリアすることにより、DMA Tx バッファおよび Rx バッファを無効にします。

## DMA によるパッキング

DMA (SPIx\_CR2 レジスタでセットされた TXDMAEN および RXDMAEN) で転送が管理される場合、パッキングモードは SPI TX および SPI RX DMA チャンネルで設定される PSIZE 値によって、自動的に有効/無効になります。DMA チャンネルの PSIZE 値が 16 ビットと等しい場合、SPI データサイズは 8 ビット以下で、パッキングモードは有効になります。その後、DMA は SPIx\_DR レジスタへの書き込み動作を管理します。

データパッキングモードを使用する場合で転送データ数が 2 の倍数でない場合は、LDMA\_TX/LDMA\_RX ビットをセットする必要があります。SPI では、最後の DMA 転送の送受信に 1 つのデータのみを考慮します (詳細については、[データパッキング \(1064 ページ\)](#) を参照)。

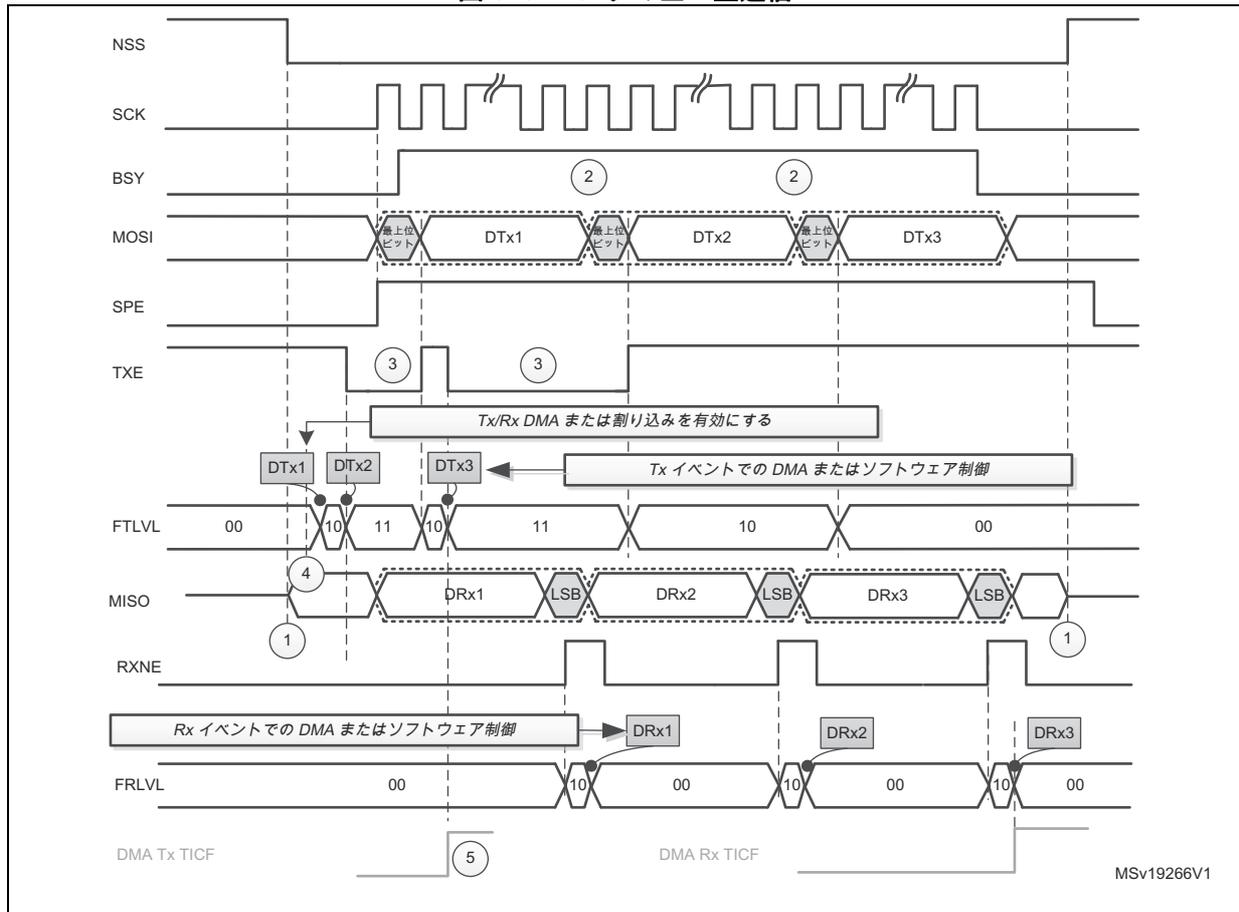
## 通信図

このセクションでは、一部の標準的なタイミング構成について説明します。これらの構成は、SPI イベントの処理方法（ポーリング、割り込み、または DMA）にかかわらず有効です。単純にするには、ここでは LSBFIRST=0、CPOL=0、および CPHA=1 設定を共通の前提として使用します。DMA ストリームの完全な設定は提供されません。

以下の番号付された注は、[図 348 \(1068 ページ\)](#) から [図 351 \(1071 ページ\)](#) に共通です。

- スレーブは、NSS がアクティブで SPI が有効になると MISO ラインを制御し始め、そのいずれかがリリースされるとラインから切断されます。事前にマスタ専用のデータを準備するには、トランザクションを開始する前にスレーブに十分な時間を与える必要があります。  
マスタでは、SPI が有効な場合のみ、SPI ペリフェラルが MOSI および SCK 信号で（また、ときに NSS 信号でも）制御できるようになります。SPI が無効の場合、SPI ペリフェラルは GPIO ロジックから切断され、これによりこれらのラインのレベルは GPIO 設定にのみ依存します。
- マスタでは、通信（クロック信号）が連続的な場合に、BSY はフレーム間でアクティブなままとなります。スレーブでは、BSY 信号は必ずフレーム間で最低でも 1 クロックサイクルは無効になります。
- TXE 信号は TXFIFO がフルの場合にのみクリアされます。
- DMA アービトレーションプロセスは TXDMAEN ビットのセット後すぐに開始します。TXE 割り込みは TXEIE のセット後すぐに生成されます。TXE 信号がアクティブレベルにあると、TxFIFO へのデータ転送が開始され、TxFIFO がフルまたは DMA 転送が完了するまで続きます。
- 送信するデータをすべて TxFIFO に適合することができる場合、SPI バスでの通信が開始される前に DMA Tx TCIF フラグを立てることができます。このフラグは、常に SPI トランザクションが完了する前に立てられます。
- パッケージの CRC 値は、SPIx\_TxCRCR および SPIx\_RxCRCR レジスタで、フレームごとに連続的に計算されます。CRC 情報はすべてのデータパッケージが完了した後で処理されます。これは、自動的に DMA で（必ず Tx チャンネルを処理するフレーム数にセットすること）、またはソフトウェアで（必ず最後のデータフレーム処理中に CRCNEXT を処理すること）行われます。SPIx\_TxCRCR で計算された CRC 値がトランスミッタによって簡単に送信されるのに対し、受信した CRC 情報は RxFIFO に読み込まれ、SPIx\_RxCRCR レジスタの内容と比較されます（差異がある場合は、ここで CRC エラーフラグが立てられます）。このため、ユーザは注意してこの情報を FIFO から一掃する必要があります。これには、RxFIFO のすべての保管された内容を読み出すソフトウェアか、Rx チャンネルの適切なデータフレーム数がプリセットされている場合は DMA を使用します（データフレーム数 + CRC フレーム数）（前提条件の例の設定を参照）。
- データがパックされたモードで、TxE および RxNE イベントはペアにされ、データフレーム数が偶数になるまで FIFO への各読み出し／書き込みアクセスは 16 ビット幅となります。TxFIFO が  $\frac{3}{4}$  フルの場合、FTLVL ステータスは FIFO フルレベルのままです。このため、最後の奇数のデータフレームを TxFIFO が  $\frac{1}{2}$  フルになる前に格納することはできません。このフレームは 8 ビットアクセスを持つ TxFIFO に格納されます。これは、ソフトウェアで、または LDMA\_TX 制御がセットされている場合は自動的に DMA で行われます。
- パックされたモードで最後の奇数のデータフレームを受信するには、最後のデータフレームが処理された時に Rx 閾値を 8 ビットに変更する必要があります。これは、ソフトウェアで FRXTH=1 に設定するか、LDMA\_RX がセットされている場合は自動的に DMA 内部信号で行われます。

図 348. マスタの全二重通信



マスタの全二重通信の前提条件の例：

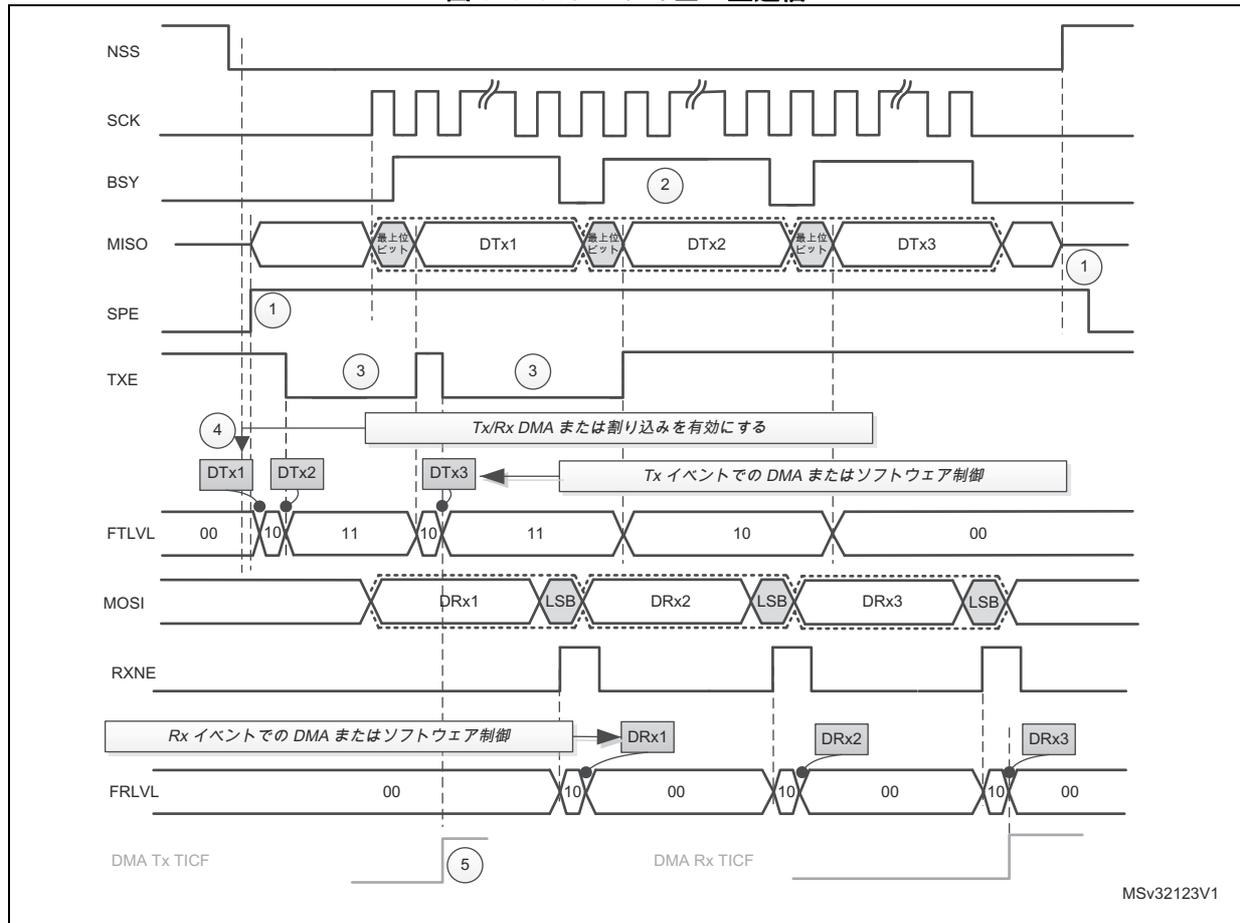
- データサイズ > 8 ビット

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(1067 ページ\)](#) を参照してください。

図 349. スレーブの全二重通信



スレーブの全二重通信の前提条件の例：

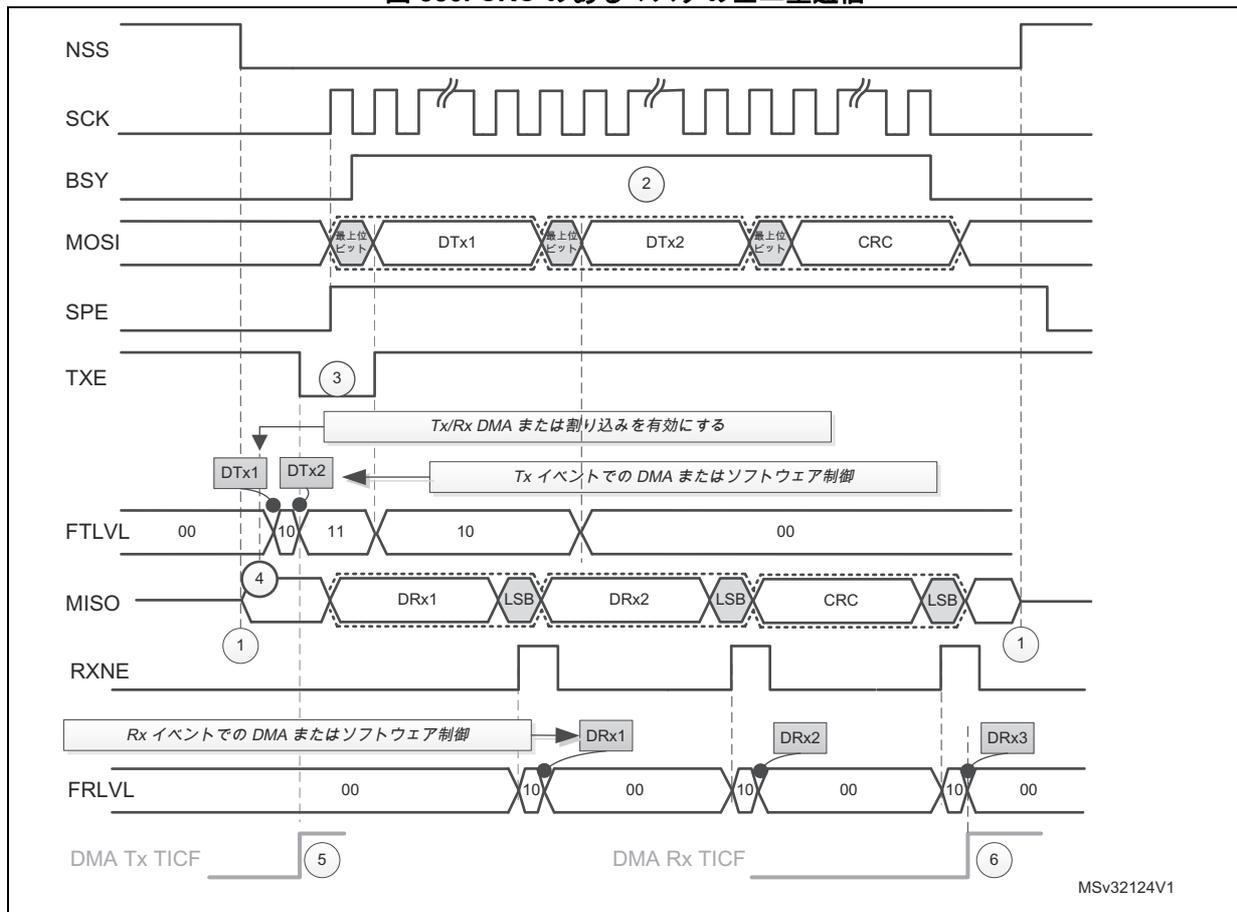
- データサイズ > 8 ビット

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(1067 ページ\)](#) を参照してください。

図 350. CRC のあるマスタの全二重通信



CRC のあるマスタの全二重通信の前提条件の例 :

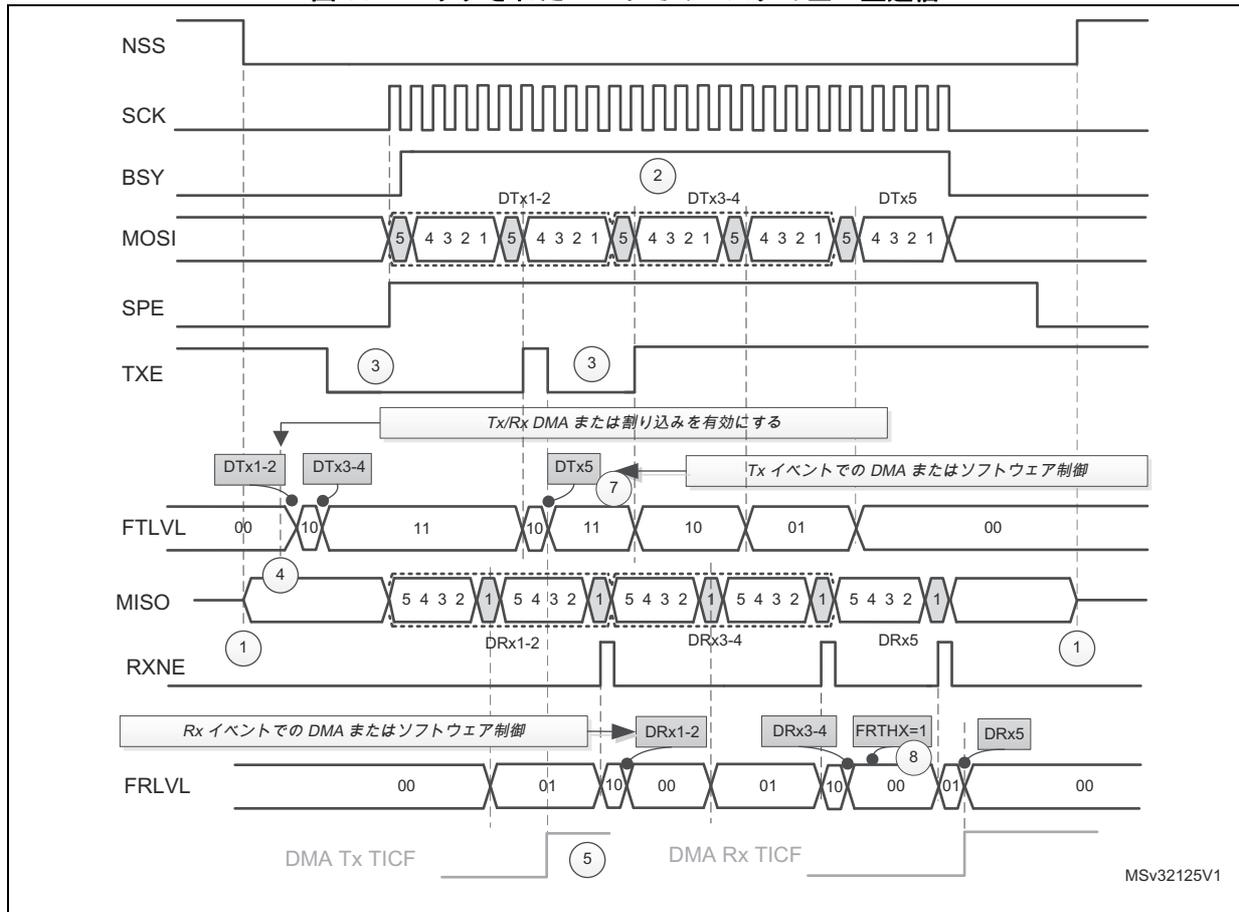
- データサイズ = 16 ビット
- CRC は有効

DMA を使用する場合 :

- DMA でトランザクションされる Tx フレーム数を 2 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(1067 ページ\)](#) を参照してください。

図 351. パックされたモードでのマスタの全二重通信



パックされたモードでのマスタの全二重通信の前提条件の例：

- データサイズ = 5 ビット
- 読み出し／書き込み FIFO は、ほとんど 16 ビットアクセスで実行されます。
- FRXTH=0

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット
- Tx と Rx DMA チャンルの両方の PSIZE を 16 ビットにセット
- LDMA\_TX=1 および LDMA\_RX=1

共通の前提条件および注の詳細については、[通信図 \(1067 ページ\)](#) を参照してください。

## 32.5.9 SPI ステータスフラグ

アプリケーションが SPI バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

### Tx バッファエンプティフラグ (TXE)

TXE フラグは、送信 TXFIFO に送信データを格納するための十分なスペースがある場合にセットされます。TXE フラグは TXFIFO レベルにリンクされます。フラグはハイになり、TXFIFO レベルが FIFO の深さの 1/2 以下になってもハイを維持します。SPIx\_CR2 レジスタの TXEIE ビットがセットされている場合は、割り込みを生成できます。このビットは、TXFIFO レベルが 1/2 を超えると自動的にクリアされます。

### Rx バッファノットエンプティ (RXNE)

SPIx\_CR2 レジスタの FRXTH ビット値によって、RXNE フラグがセットされます。

- FRXTH をセットすると RXNE はハイになり、RXFIFO レベルが 1/4 (8 ビット) 以上になってもハイを維持します。
- FRXTH をクリアすると RXNE はハイになり、RXFIFO レベルが 1/2 (16 ビット) 以上になってもハイを維持します。

SPIx\_CR2 レジスタの RXNEIE ビットがセットされている場合は、割り込みを生成できます。

RXNE は、上記の条件が真ではなくなった場合にハードウェアによって自動的にクリアされます。

### ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます (このフラグへの書き込みは無効)。

BSY フラグがセットされると、SPI 上でデータ転送が進行中であることを示します (SPI バスはビジー)。

BSY は、転送終了を検出するために特定のモードで使用できます。これにより、低電力モード (ペリフェラルにクロックを供給しない) に入る前に、ソフトウェアで SPI やそのペリフェラルクロックを無効にすることができます。これによって、最後の転送データの破壊を回避します。

BSY フラグは、マルチマスタシステムでの書き込み衝突の回避にも役立ちます。

BSY フラグは次のいずれかの条件下でクリアされます。

- SPI が正常に無効にされたとき
- マスタモードで、障害が検出 (MODF ビットが 1 にセットされます) されたとき
- マスタモードで、データ送信が終了し、送信準備ができていない新しいデータがないとき
- スレーブモードで、各データ転送間で少なくとも SPI の 1 クロックサイクルの間、BSY フラグが 0 にセットされているとき

**注：** 次の送信がすぐにマスタで処理される場合 (マスタが受信専用モードに設定されているか、その送信 FIFO がエンプティでない場合)、通信は連続的で、BSY フラグはマスタ側での送信の間、“1”にセットされたままになります。スレーブとは異なりますが、必ず (BSY フラグを使用する代わりに) TXE フラグと RXNE フラグを使用して、データの送受信の処理を行うことを推奨します。

## 32.5.10 SPI エラーフラグ

次のいずれかのエラーフラグがセットされていて、ERRIE ビットをセットすることにより割り込みが有効になっている場合、SPI 割り込みが生成されます。

### オーバーランフラグ (OVR)

マスタまたはスレーブでデータを受信し、RXFIFO に受信データを格納するための十分なスペースがない場合に、オーバーラン状態が発生します。これは、ソフトウェアまたは DMA が RXFIFO に格納された前の受信データを読み出すために必要な時間が十分になかったか、データストレージのための空間が制限されている場合に発生する可能性があります。たとえば、RXFIFO は CRC が受信専用モードでのみ有効な場合は使用できないため、この場合、受信バッファは単一データフレームバッファに制限されます ([セクション 32.5.13 : CRC 計算](#)を参照)。

オーバーラン条件が発生すると、RXFIFO にある前の値を新しく受信した値で上書きすることはできません。新しく受信した値は破棄され、それ以降に送信されたすべてのデータは失われます。OVR ビットをクリアするには、SPI\_DR レジスタを読み出し、続けて SPI\_SR レジスタを読み出しアクセスを行います。

### モードフォールト (MODF)

モードフォールトは、マスタデバイスが内部 NSS 信号 (NSS ハードウェアモードでは NSS ピン、NSS ソフトウェアモードでは SSI ビット) をローレベルにプルダウンしたときに発生します。これにより、MODF ビットが自動的にセットされます。マスタモードフォールトは、SPI インタフェースに次のような影響を与えます。

- MODF ビットがセットされ、ERRIE ビットがセットされている場合は SPI 割り込みが生成されます。
- SPE ビットがクリアされます。これによって、デバイスからのすべての出力がブロックされ、SPI インタフェースが無効になります。
- MSTR ビットがクリアされ、デバイスは強制的にスレーブモードになります。

MODF ビットをクリアするには、次のソフトウェアシーケンスを実行します。

1. MODF ビットがセットされている間、SPIx\_SR レジスタを読み出し/書き込みアクセスを行います。
2. 次に、SPIx\_CR1 レジスタに書き込みを行います。

複数の MCU で構成されるシステムでスレーブ間の競合を避けるには、MODF ビットをクリアするシーケンス中、NSS ピンをハイレベルにプルアップする必要があります。このクリアシーケンスの後、SPE ビットと MSTR ビットは、元の状態に戻すことができます。安全のため、MODF ビットがセットされている間、ハードウェアは SPE ビットと MSTR ビットのセットを許可しません。スレーブデバイスでは、MODF ビットはセットできません。ただし、前回のマルチマスタ競合の結果としてセットする場合は例外です。

### CRC エラー (CRCERR)

このフラグを使用して、SPIx\_CR1 レジスタの CRCEN ビットがセットされているときに受信された値の有効性を検証します。シフトレジスタに受信された値が、レシーバである SPIx\_RXCRCR の値と一致しなかった場合、SPIx\_SR レジスタの CRCERR フラグがセットされます。フラグはソフトウェアによってクリアされます。

## TI モードフレームフォーマットエラー (FRE)

SPI がスレーブモードで動作し、かつ TI モードプロトコルに準拠した設定となっている場合、通信の進行中に NSS パルスが発生すると、TI モードフレームフォーマットエラーが検出されます。このエラーが発生すると、SPIx\_SR レジスタの FRE フラグがセットされます。エラー発生時には SPI は無効にされず、この NSS パルスは無視されます。SPI は次の NSS パルスを待ってから新規の転送を開始します。このエラーの検出により 2 バイトのデータが失われるため、データは破壊される可能性があります。

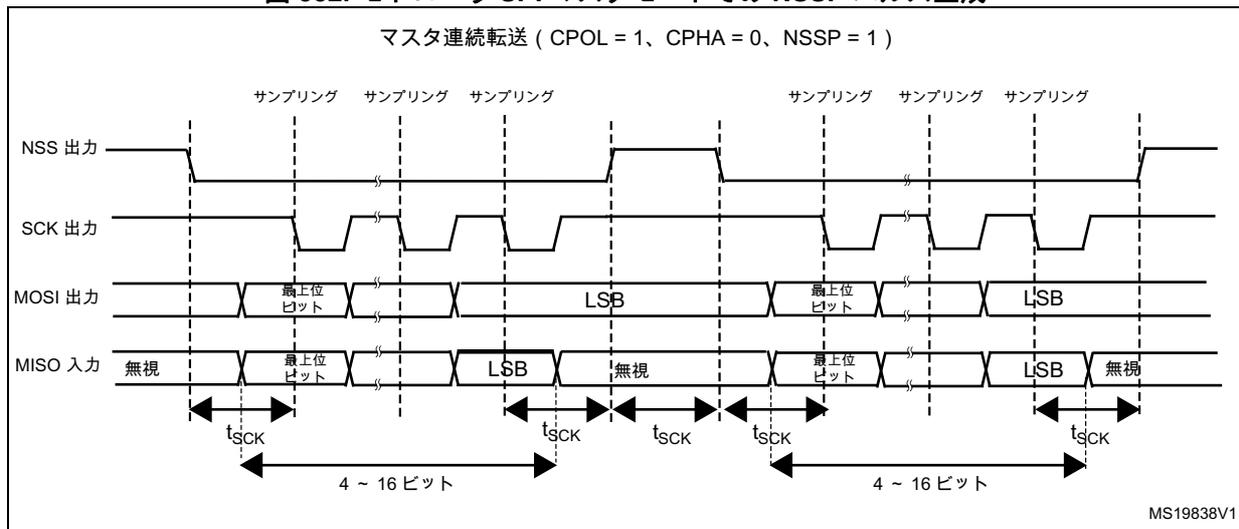
SPIx\_SR レジスタを読み出すと、FRE フラグがクリアされます。ERRIE ビットがセットされている場合、NSS エラー検出時に割り込みが生成されます。この場合、データの一貫性が保証されなくなるため、SPI を無効にする必要があり、またスレーブ SPI が再び有効化された場合は、マスタによって通信を再起動する必要があります。

## 32.5.11 NSS パルスモード

このモードは SPIx\_CR2 レジスタの NSSP ビットで有効化され、SPI インタフェースが 1 番目のエッジにキャプチャのある モトローラ SPI マスタ (FRF=0) として設定されている場合にのみ有効になります (SPIx\_CR1 CPHA = 0、CPOL 設定は無視されます)。有効にすると、最低 1 つのクロック周期の間、NSS がハイレベルを維持する場合に、2 つの連続したデータフレーム転送の間で NSS パルスが生成されます。このモードでは、スレーブでデータをラッチできます。NSSP パルスモードは、シングルマスタ - スレーブペアのアプリケーション用に設計されています。

図 352 に、NSSP パルスモードが有効な場合の NSS ピンの管理の図を示します。

図 352. モトローラ SPI マスタモードでの NSSP パルス生成



注: CPOL = 0 の場合に同様の動作が発生します。この場合、サンプリングエッジは SCK の立ち上がりエッジで、NSS のアサートおよびネゲートはこのサンプリングエッジを参照します。

## 32.5.12 TI モード

### マスターモードでの TI プロトコル

SPI インタフェースは TI プロトコルと互換性があります。SPIx\_CR2 レジスタの FRF ビットを使って、SPI をこのプロトコルに準拠させるように設定することができます。

SPIx\_CR1 レジスタにセットされる値によらず、クロックの極性と位相は TI プロトコル要件に必ず適合します。NSS 管理も TI プロトコルに固有なものになります。これにより、この場合の SPIx\_CR1 レジスタと SPIx\_CR2 レジスタによる NSS 管理の設定 (SSM、SSI、SSOE) ができなくなります。

スレーブモードでは、SPI ボーレートプリスケラを使用して、現在のトランザクションが終了した時点で、MISO ピンの状態がハイインピーダンスに変化するタイミングを制御します (図 353 を参照)。任意のボーレートが使用できるため、このタイミングを非常に柔軟に決定することができます。ただし、ボーレートは外部マスタクロックボーレートに設定されるのが一般的です。MISO 信号がハイインピーダンス ( $t_{\text{release}}$ ) になるまでの遅延は、内部再同期と SPIx\_CR1 レジスタの BR[2:0] ビットで設定されたボーレート値によって変わります。この値は次式で求められます：

$$\frac{t_{\text{baud\_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud\_rate}}}{2} + 6 \times t_{\text{pclk}}$$

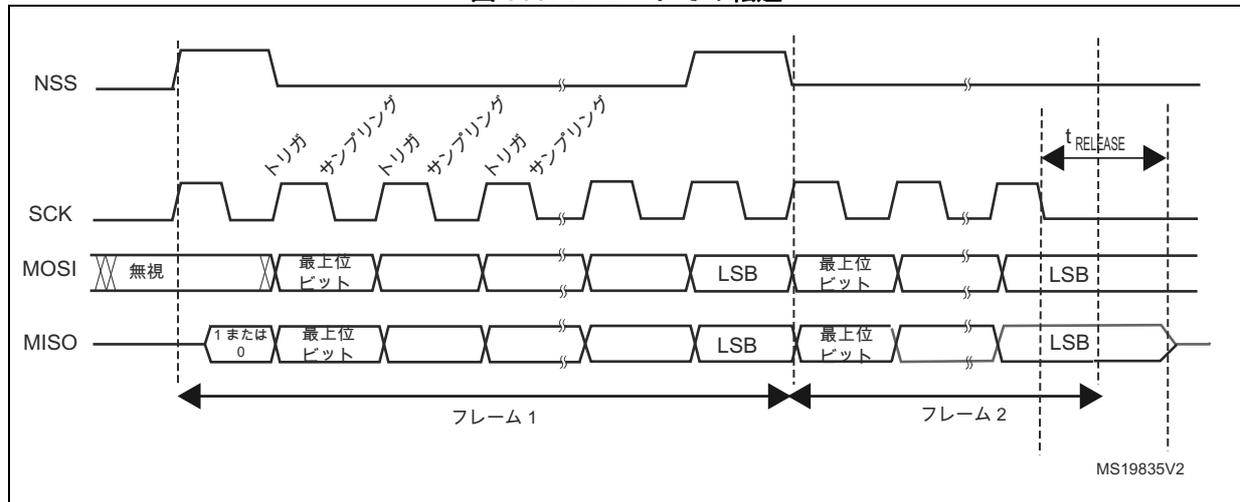
スレーブがデータフレームトランザクション中に NSS パルスの誤配置を検出すると、TIFRE フラグがセットされます。

データサイズが 4 ビットまたは 5 ビットである場合、全二重モードまたは送信専用モードのマスタは、最下位ビット後に追加された 1 つまたは複数のダミーデータビットを持つプロトコルを使用しません。TI NSS パルスは、各周期の最下位ビットではなく、このダミービットクロックサイクル上に生成されます。

この機能はモトローラの SPI 通信には使用できません (FRF ビットを 0 に設定)。

図 353 : TI モードでの転送 TI モードが選択されているときの SPI の通信波形を示します。

図 353. TI モードでの転送



## 32.5.13 CRC 計算

2つのCRC計算機が、送信データおよび受信データの信頼性をチェックするために実装されています。SPIは、フレームデータ長とは別に、8ビットまたは16ビットに固定されるCRC8またはCRC16を計算します。その他のデータフレーム長では、CRCは使用できません。

### CRCの原理

CRC計算は、SPIが有効(SPE = 1)になる前に、SPIx\_CR1レジスタのCRCENビットをセットすることによって有効になります。CRC値は、各ビットに対して奇数のプログラム可能な多項式の値を使用して計算されます。計算は、SPIx\_CR1レジスタのCPHAビットとCPOLビットによって定義されるサンプリングクロックエッジで行われます。計算されたCRC値は、データブロックの最後のみならず、CPUまたはDMAによって管理される転送に関しても、自動的にチェックされます。受信データをもとに内部で計算されたCRC値とトランスミッタが送信したCRC値の間に不一致が検出された場合、データ破壊エラーを示すためにCRCERRフラグがセットされます。CRC計算を処理する正しい手順は、SPIの設定および選択された転送管理によって変わります。

**注：** *多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。*

### CPUによって管理されるCRC転送

通信が開始され、最後のデータフレームがSPIx\_DRレジスタで送信または受信されるまで正常に続きます。次に、CRCフレームトランザクションが現在処理中のデータフレームトランザクションの後に行われることを示すために、SPIx\_CR1レジスタのCRCNEXTビットをセットする必要があります。CRCNEXTビットは、最後のデータフレームトランザクションの終了前にセットしてください。CRCのトランザクション中はCRC計算は行われません。

受信したCRCは、データバイトまたはワードのようにRXFIFOに格納されます。このため、CRCモードでは、受信バッファを一度に1データフレームずつ受信するために使用する単一の16ビットバッファとみなす必要があります。

CRCフォーマットトランザクションは、通常データシーケンスの最後に通信を行うためにデータフレームをもう1つ受け取ります。ただし、16ビットCRCによってチェックされた8ビットデータフレームを設定する場合、CRC全体を送信するにはフレームがあと2つ必要です。

最後のCRCデータを受信すると、受信した値とSPIx\_RXCRCレジスタの値を比較する自動チェックが行われます。ソフトウェアは、SPIx\_SRレジスタのCRCERRフラグをチェックして、データ転送の内容が破壊されているか否かを判断する必要があります。ソフトウェアは、CRCERRフラグに“0”を書き込んでクリアします。

CRC受信後、CRC値はRXFIFOに格納され、RXNEフラグをクリアするためにSPIx\_DRレジスタを読み出す必要があります。

### DMAによって管理されるCRC転送

SPI通信がCRC通信とDMAモードで有効化される場合、CRCの送受信は通信の最後に自動で行われます(ただし、受信専用モードでCRCデータを読みだす場合を除く)。CRCNEXTビットはソフトウェアで処理する必要はありません。SPI送信用DMAチャネルのカウンタは、転送するデータフレーム数からCRCフレームを除いてセットする必要があります。レシーバ側では、受信したCRC値はトランザクションの終了時にDMAによって自動的に処理されますが、RXFIFOから受信したCRC情報は常にレシーバにロードされますので、ユーザは注意してそれを一掃してください。全二重モードでは、受信DMAチャネルのカウンタを、受信するデータフレームの数(CRCを含む)にセットすることができます。つまり、たとえば16ビットCRCでチェックされた8ビットデータフレームの特定の例を、次の式で表すことができます。

$$\text{DMA\_RX} = \text{Numb\_of\_data} + 2$$

受信専用モードでは、DMA 受信チャンネルカウンタには転送されたデータ量のみが含まれます (CRC 計算を除く)。次に、DMA からの全体の転送に基づき、すべての CRC 値はこのモードでは単一のバッファとして動作するため、ソフトウェアによって FIFO から読み戻す必要があります。

転送中に内容の破壊が生じた場合、データと CRC の転送の最後に SPIx\_SR レジスタの CRCERR フラグがセットされます。

パッキングモードが使用され、データ数が奇数の場合は、LDMA\_RX ビットを管理する必要があります。

## SPIx\_TXCRC および SPIx\_RXCRC の値のリセット

CRC フェーズのあとで新しいデータがサンプリングされた場合、SPIx\_TXCRC 値および SPIx\_RXCRC 値は自動的にクリアされます。これにより、データを中断することなく転送するために、DMA サーキュラモードが使用可能になります (受信専用モードでは使用不可)。いくつかのデータブロックは中間の CRC チェックフェーズによってカバーされます。

通信中に SPI が無効化された場合は、次のシーケンスに従う必要があります。

1. SPI を無効にします。
2. CRCEN ビットをクリアします。
3. CRCEN ビットを有効にします。
4. SPI を有効にします。

**注：** SPI がスレープモードのとき、SPE ビットの値に関係なく、CRCEN がセットされるとすぐに、CRC 計算機は SCK スレープ入カクロックの影響を受けます。CRC 計算の間違いを回避するために、ソフトウェアは、クロックが安定している (定常状態にある) ときのみ、CRC 計算を有効にする必要があります。SPI インタフェースがスレープとして設定されている場合、データフェーズと CRC フェーズの間で NSS 内部信号をローレベルに維持する必要があります。

## 32.6 SPI 割り込み

SPI 通信中、割り込みは次のイベントによって生成できます。

- ロードする準備ができていない TXFIFO の送信
- 受信 RXFIFO で受信したデータ
- マスタモードフォールト
- オーバーランエラー
- TI フレームフォーマットエラー
- CRC プロトコルエラー

割り込みは個別に有効または無効にできます。

表 176. SPI 割り込みリクエスト

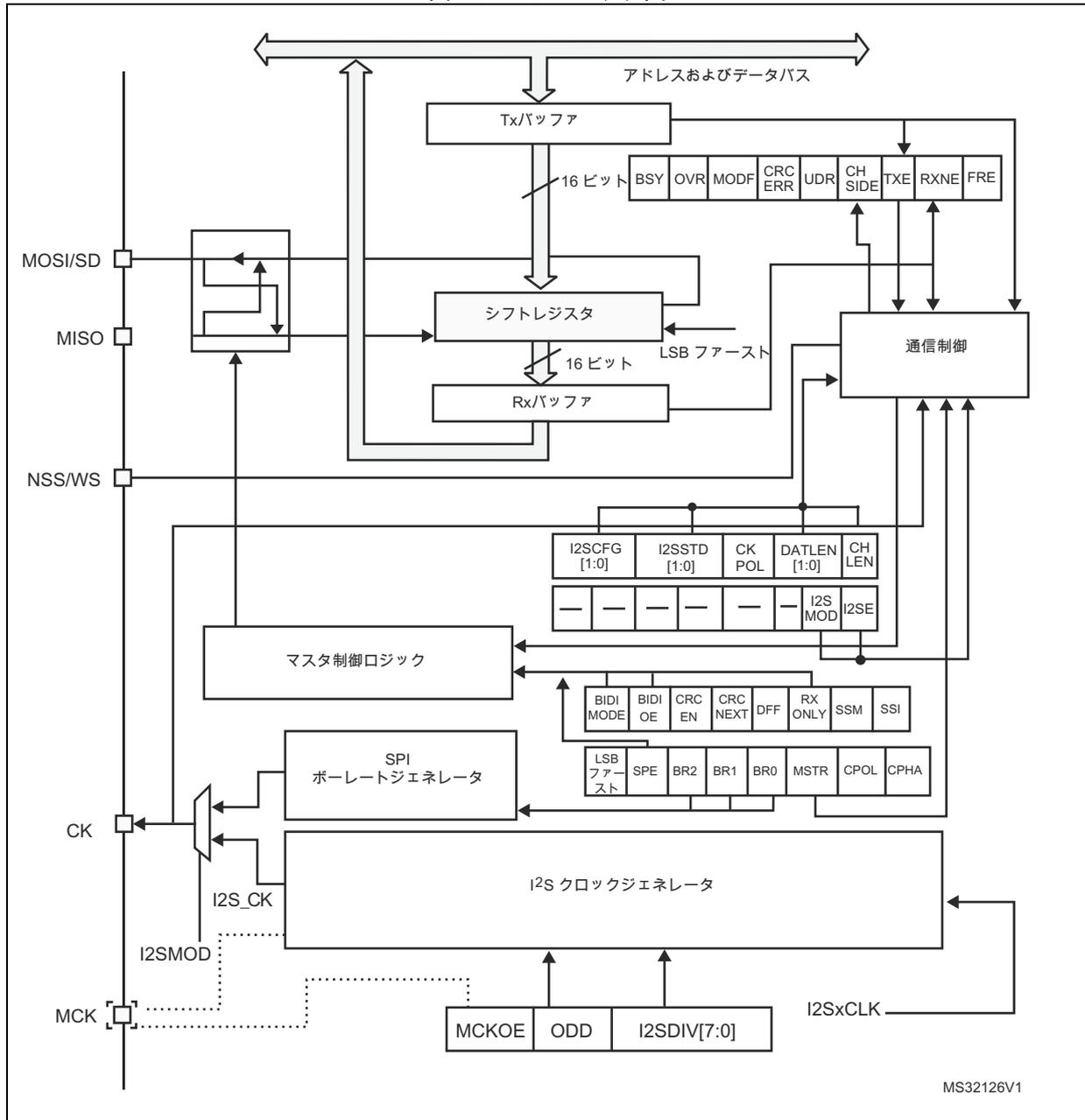
割り込みイベント	イベントフラグ	イネーブル制御ビット
ロードする準備ができていない TXFIFO の送信	TXE	TXEIE
RXFIFO で受信したデータ	RXNE	RXNEIE
マスタモードフォールトイベント	MODF	ERRIE
オーバーランエラー	OVR	
TI フレームフォーマットエラー	FRE	
CRC プロトコルエラー	CRCERR	

## 32.7 I<sup>2</sup>S の機能説明

### 32.7.1 I<sup>2</sup>S の概要

I<sup>2</sup>S のブロック図を [図 354](#) に示します。

図 354. I<sup>2</sup>S ブロック図



1. MCK は MISO ピンに配置されます。

SPI はオーディオ I<sup>2</sup>S インタフェースとして機能することができます。それには、SPIx\_I2SCFGR レジスタの I2SMOD ビットをセットして I<sup>2</sup>S 機能を有効にしてください。このインタフェースは、主に SPI と同じピン、フラグ、および割り込みを使用します。

I<sup>2</sup>S と SPI は、以下の 3 つのピンを共用します。

- SD (シリアルデータ) : MOSI ピンに配置され、2 つの時間多重化データチャネルを送受信します (半二重モードのみ)。
- WS (ワードセレクト) : NSS ピンに配置され、マスタモードではデータ制御信号の出力、スレーブモードでは入力です。
- CK (シリアルクロック) : SCK ピンに配置され、マスタモードではシリアルクロック出力、スレーブモードではシリアルクロック入力です。

外部オーディオデバイスにマスタクロック出力が必要な場合、追加のピンを使用できます。

- MCK (マスタクロック) : 別々に配置され、I<sup>2</sup>S がマスタモードに設定されている (かつ、SPIx\_I2SPR レジスタの MCKOE ビットがセットされている) とき、 $256 \times f_S$  と等しい ( $f_S$  はオーディオサンプリング周波数) 設定済みの周波数で生成されたこの追加クロックを出力するために使用されます。

I<sup>2</sup>S は、マスタモードにセットされているとき、専用のクロックジェネレータを使用して通信クロックを生成します。このクロックジェネレータは、マスタクロック出力のソースでもあります。I<sup>2</sup>S モードでは、2 つの追加レジスタを使用できます。1 つはクロックジェネレータ設定 SPIx\_I2SPR にリンクされ、もう 1 つは汎用 I<sup>2</sup>S 設定レジスタ SPIx\_I2SCFGR (オーディオ規格、スレーブ/マスタモード、データフォーマット、パケットフレーム、クロック極性など) です。

SPIx\_CR1 レジスタとすべての CRC レジスタは、I<sup>2</sup>S モードでは使用されません。同様に、SPIx\_CR2 レジスタの SSOE ビットと SPIx\_SR レジスタの MODF および CRCERR ビットも使用されません。

I<sup>2</sup>S は、16 ビット幅モードでのデータ転送に同じ SPI レジスタ (SPIx\_DR) を使用します。

## 32.7.2 サポートされるオーディオプロトコル

3 線バスでは、一般に 2 つのチャネル (右チャネルと左チャネル) で時間多重化されたオーディオデータのみを処理する必要があります。しかしながら、送受信には 1 つの 16 ビットレジスタしかありません。したがって、各チャネルサイドに対応する適切な値をデータレジスタに書き込んだり、データレジスタからデータを読み出して SPIx\_SR レジスタの CHSIDE ビットをチェックして対応するチャネルを識別したりすることは、ソフトウェアの責任です。左チャネルは常に最初に送信され、その後で右チャネルが送信されます (CHSIDE は PCM プロトコルには無関係です)。

4 つのデータおよびパケットフレームを使用できます。データは次のフォーマットで送信されます。

- 16 ビットフレームにパックされた 16 ビットデータ
- 16 ビットフレームにパックされた 32 ビットデータ
- 24 ビットフレームにパックされた 32 ビットデータ
- 32 ビットフレームにパックされた 32 ビットデータ

32 ビットパケットに拡張された 16 ビットデータを使用するとき、最初の 16 ビット (MSB) は上位ビットであり、LSB の 16 ビットは、ソフトウェア動作や DMA リクエストを必要とせずに (1 回の読み込み/書き込み動作のみで) 強制的に 0 にされます。

24 ビットと 32 ビットのデータフレームは、SPIx\_DR レジスタとの間で 2 回の CPU 読み出しまたは書き込み動作 (あるいはアプリケーションにとって DMA が望ましい場合は 2 回の DMA 動作) を必要とします。特に 24 ビットのデータフレームの場合、8 つの下位ビットはハードウェアによって 0 ビットで 32 ビットに拡張されます。

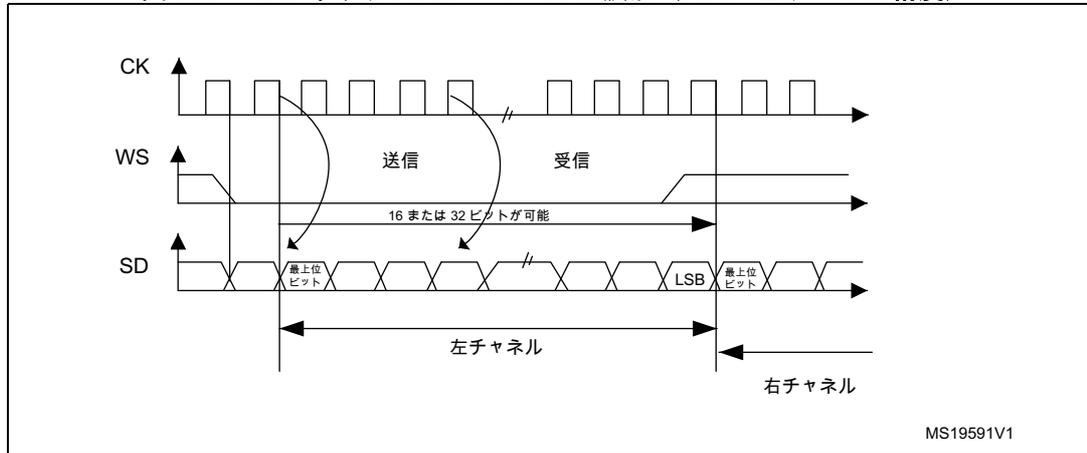
すべてのデータフォーマットと通信規格に対して、最上位ビットは常に最初に送信されます (MSB ファースト)。

I<sup>2</sup>S インタフェースは、SPIx\_I2SCFGR レジスタの I2SSTD[1:0] と PCMSYNC ビットを使用して設定可能な 4 つのオーディオ規格をサポートします。

## フィリップス I<sup>2</sup>S 規格

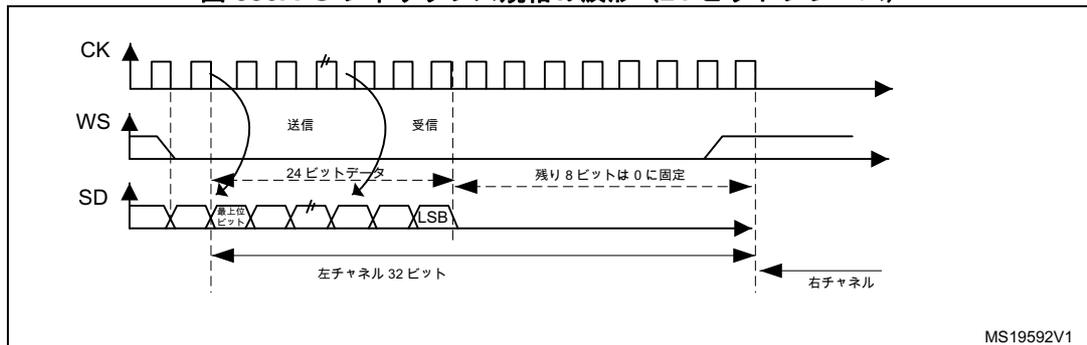
この規格では、どのチャンネルが送信されているかを示すために WS 信号を使用します。この信号が有効になってから 1 CK クロックサイクル後に最初のビット (MSB) が使用可能になります。

図 355. I<sup>2</sup>S フィリップスプロトコルの波形 (16/32 ビットフル精度)



データは、CK の立ち下がりエッジでラッチされ (トランスミッタの場合)、立ち上がりエッジで読み出されます (レシーバの場合)。WS 信号も CK の立ち下がりエッジでラッチされます。

図 356. I<sup>2</sup>S フィリップス規格の波形 (24 ビットフレーム)

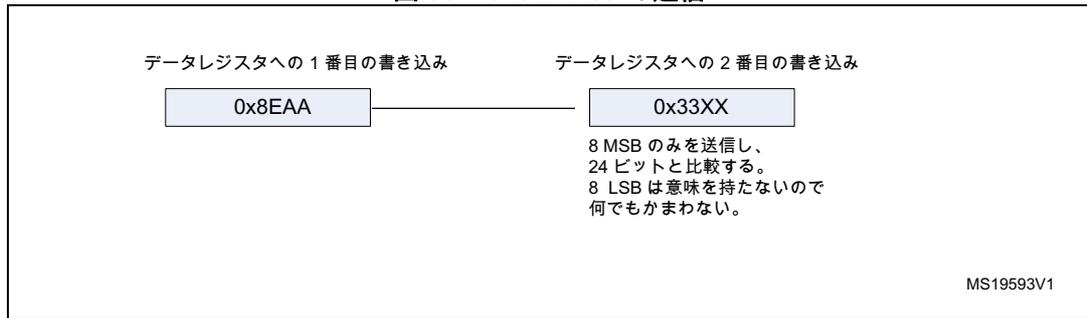


このモードでは、SPIx\_DR レジスタに対して 2 回の書き込みまたは読み出し動作が必要です。

- 送信モードの場合

0x8EAA33 を送信する必要がある場合 (24 ビット) :

図 357. 0x8EAA33 の送信



- 受信モードの場合  
データ 0x8EAA33 が受信される場合 :

図 358. 0x8EAA33 の受信

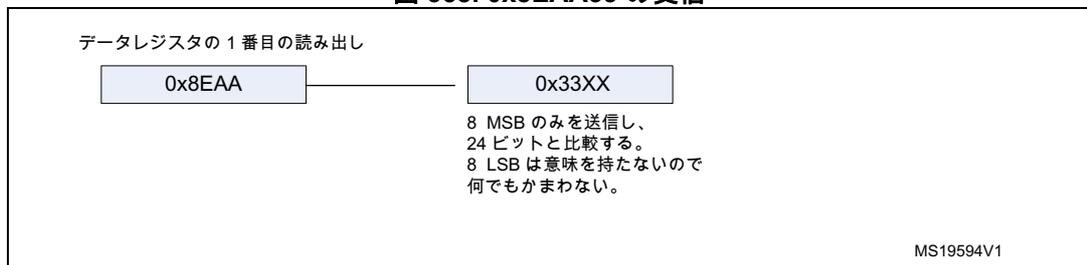
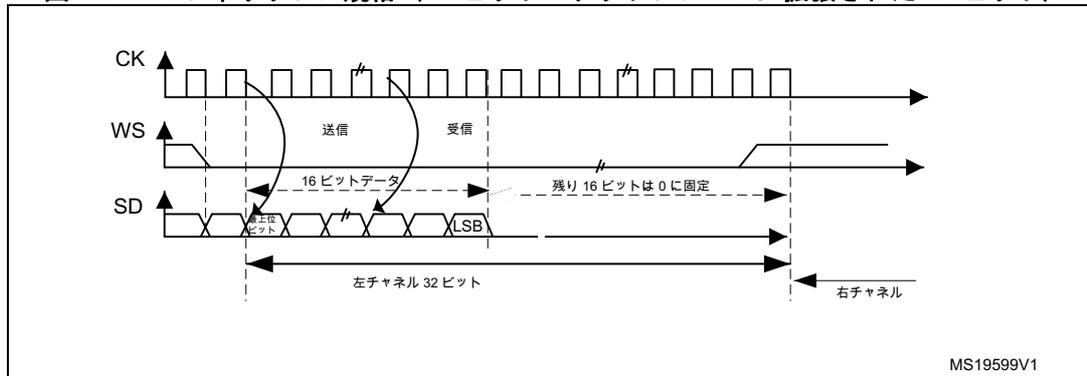


図 359. I<sup>2</sup>S フィリップス規格 (32 ビットパケットフレームに拡張された 16 ビット)



I<sup>2</sup>S 設定フェーズで、32 ビットチャンネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx\_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x76A30000) であるとき、[図 360](#) に示す動作が要求されます。

図 360. 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームの例



送信では、SPIx\_DR に MSB が書き込まれるたびに、TXE フラグがセットされ、可能ならば、SPIx\_DR レジスタに新しい送信値をロードするために割り込みが生成されます。これはハードウェアによって行われるため、0x0000 がまだ送信されていなくても行われます。

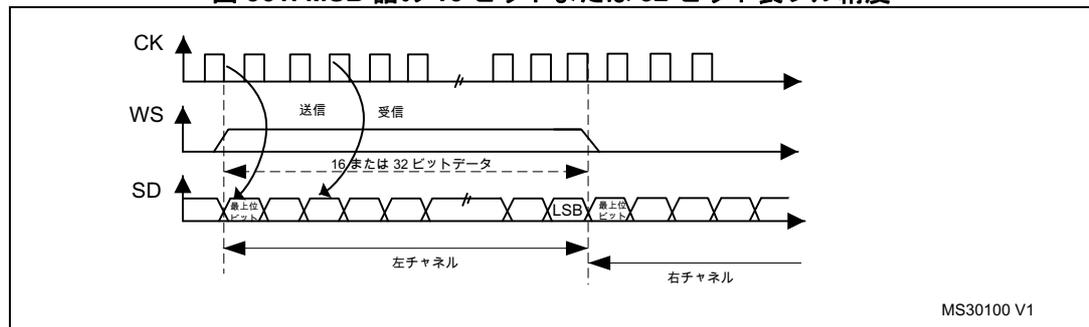
受信では、RXNE フラグがセットされ、可能ならば、最初の 16 MSB ハーフワードの受信時にその割り込みが生成されます。

このように、2 回の書き込みまたは読み出し動作の間にはより多くの時間が設けられるため、アンダーランまたはオーバーラン状態（データ転送の方向に依存）を避けられます。

## MSB 詰め規格

この規格では、WS 信号は最初のデータビット（最上位ビット）と同時に生成されます。

図 361. MSB 詰め 16 ビットまたは 32 ビット長フル精度



データは、CK の立ち下がりエッジでラッチされ（トランスミッタの場合）、立ち上がりエッジで読み出されます（レシーバの場合）。

図 362. MSB 詰め 24 ビットフレーム長

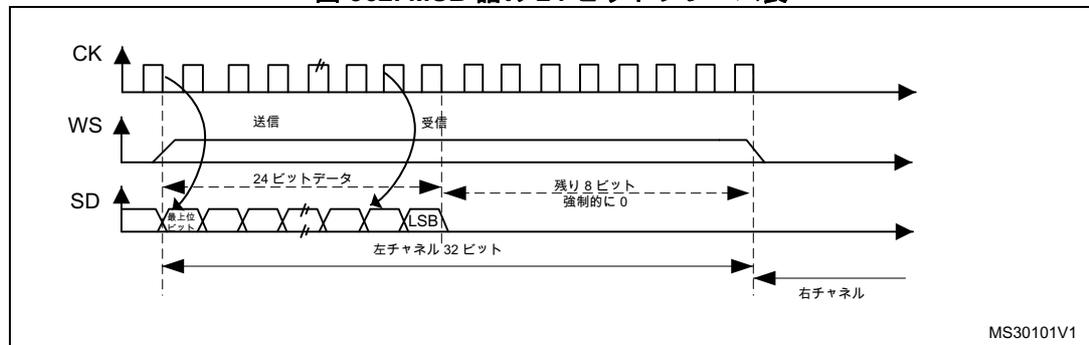
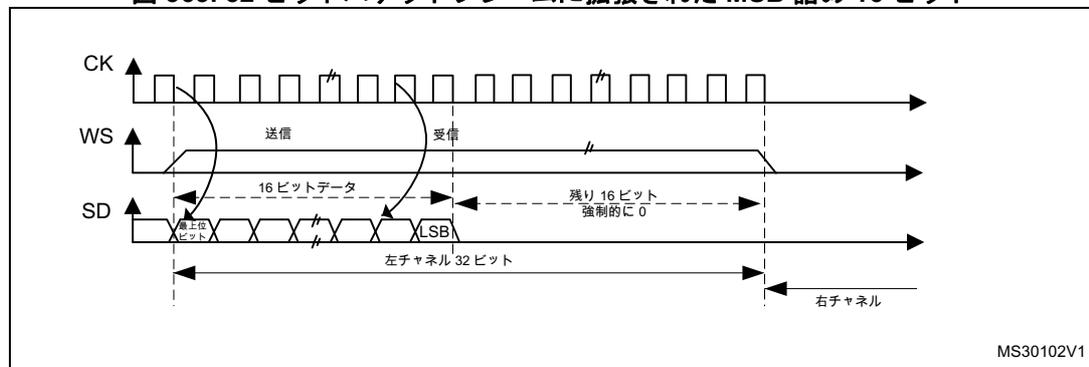


図 363. 32 ビットパケットフレームに拡張された MSB 詰め 16 ビット

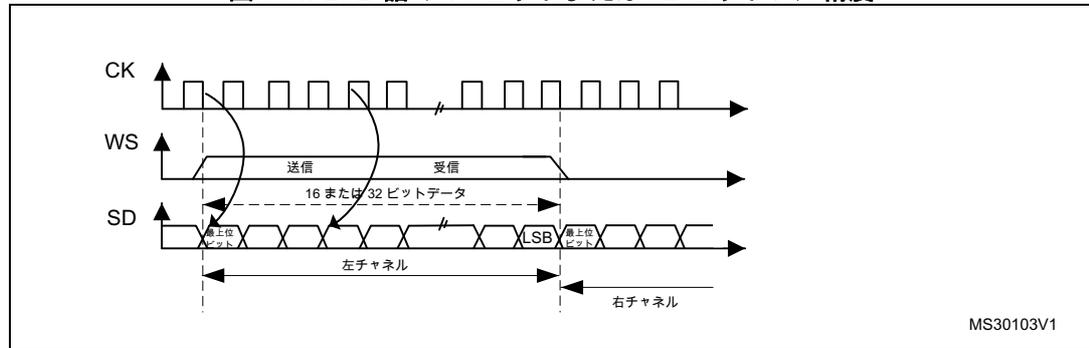


## LSB 詰め規格

この規格は、MSB 詰め規格と似ています (16 ビットと 32 ビットのフル精度フレームフォーマットに違いはありません)。

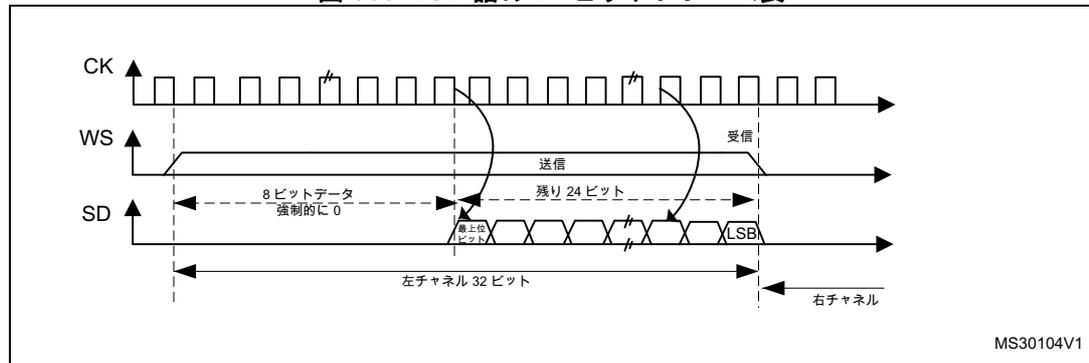
入出力信号のサンプリングは I2S フィリップス規格と同じです。

図 364. LSB 詰め 16 ビットまたは 32 ビットフル精度



MS30103V1

図 365. LSB 詰め 24 ビットフレーム長

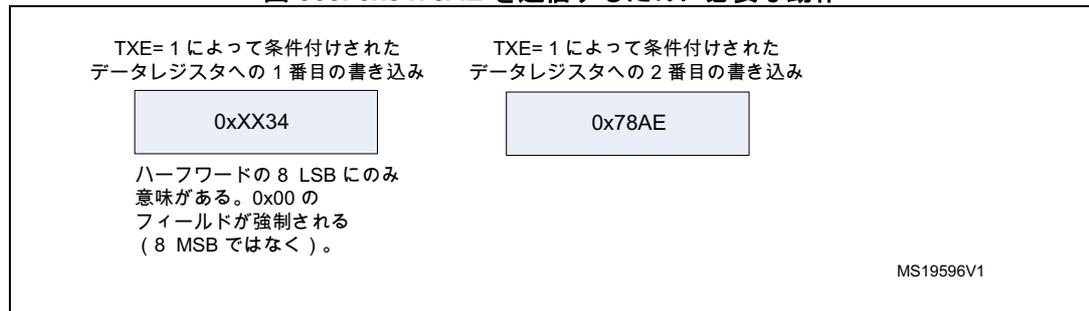


MS30104V1

- 送信モードの場合

データ 0x3478AE を送信する必要がある場合、ソフトウェアまたは DMA によって SPIx\_DR レジスタへの 2 回の書き込み動作が必要です。この動作を次に示します。

図 366. 0x3478AE を送信するために必要な動作



MS19596V1

- 受信モードの場合

データ 0x3478AE が受信される場合、RXNE イベントごとに、SPIx\_DR レジスタから連続する 2 回の読み出し動作が必要です。

図 367. 0x3478AE の受信に必要な動作

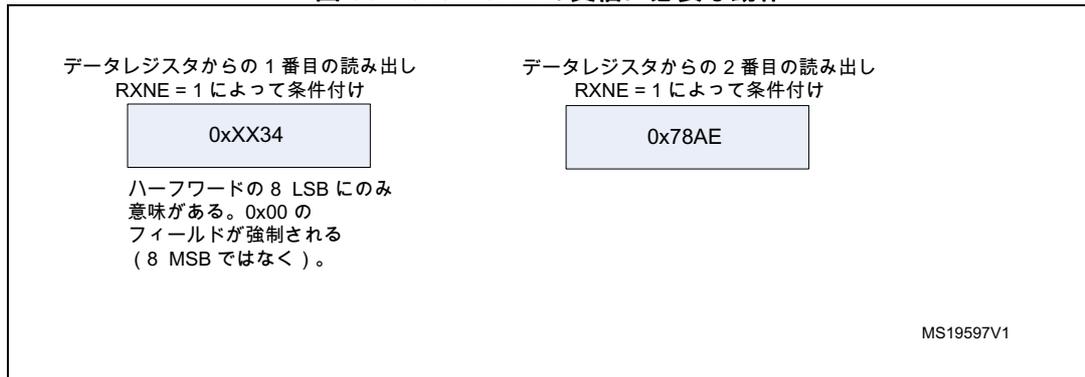
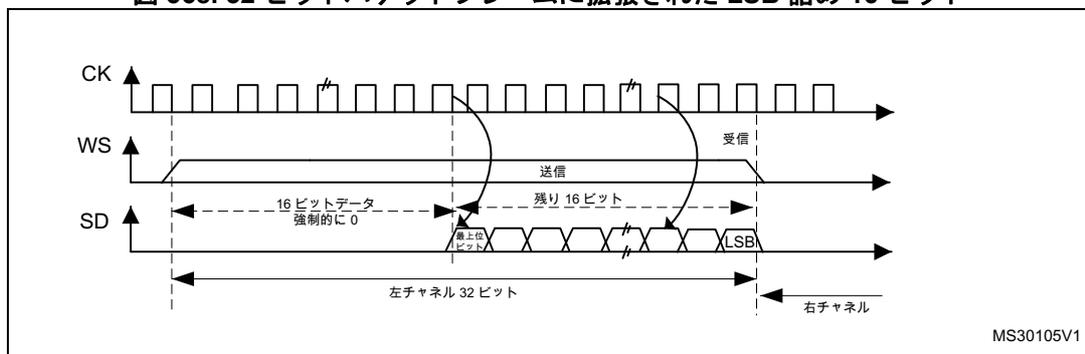


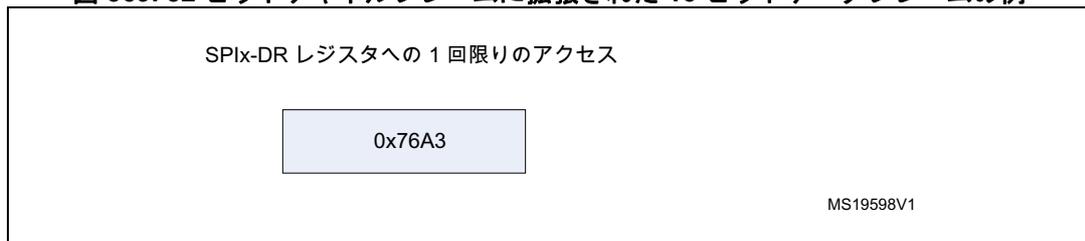
図 368. 32 ビットパケットフレームに拡張された LSB 詰め 16 ビット



I<sup>2</sup>S 設定フェーズで、32 ビットチャンネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx\_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。この場合、それはハーフワード MSB に相当します。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x0000 76A3) であるとき、[図 369](#) に示す動作が要求されます。

図 369. 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームの例



送信モードでは、TXE イベントが発生すると、アプリケーションは送信するデータ (この場合は 0x76A3) を書き込む必要があります。0x000 フィールドが最初に送信されます (32 ビット拡張部)。有効なデータ (0x76A3) が SD に送信されると、すぐに TXE フラグが再びセットされます。

受信モードでは、0x0000 フィールドではない有効なハーフワードが受信されると、すぐに RXNE がアサートされます。

このように、アンダーランやオーバーランの状態を防ぐために、2 回の書き込みまたは読み出し動作の間には、より多くの時間が設けられています。

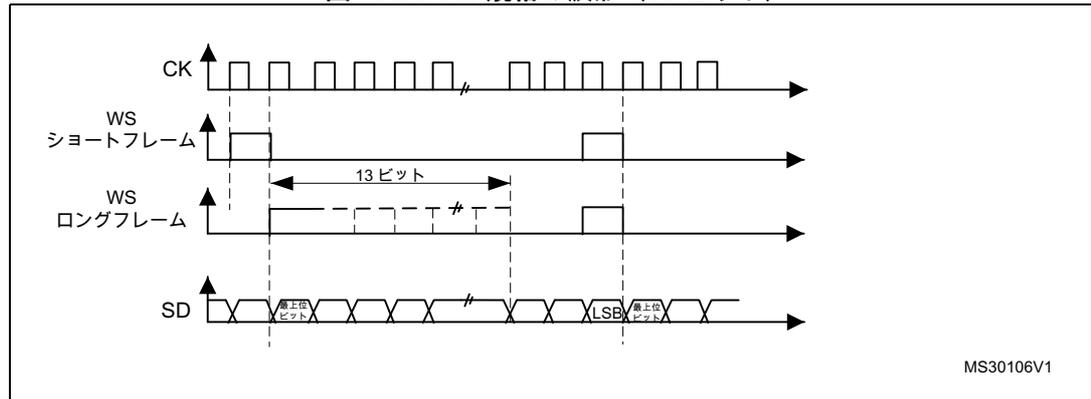
## PCM 規格

PCM 規格では、チャンネルサイド情報を使用する必要はありません。SPIx\_I2SCFGR レジスタの PCMSYNC ビットを使用して、2 つの PCM モード（ショートおよびロングフレーム）の使用および設定が可能です。

PCM モードで、出力信号（WS、SD）は CK 信号の立ち上がりエッジでサンプリングされます。入力信号（WS、SD）は立ち下がりエッジでキャプチャされます。

CK および WS は、マスタモードで出力として設定されることに注意してください。

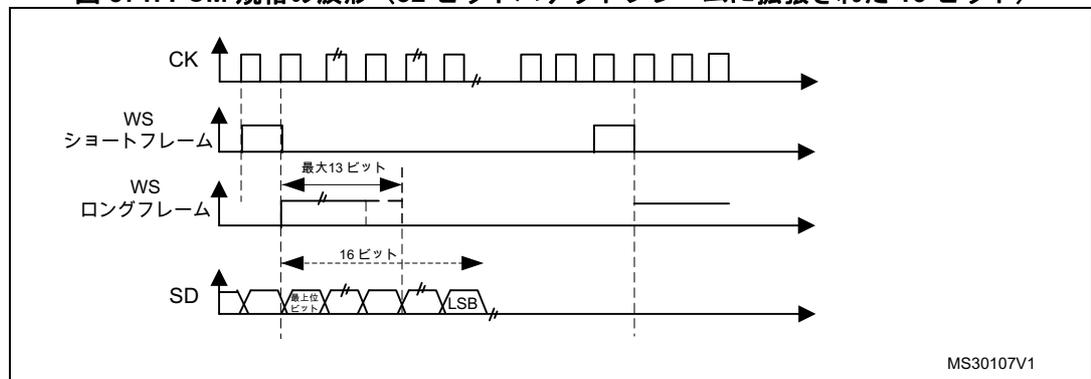
図 370. PCM 規格の波形（16 ビット）



ロングフレーム同期では、WS 信号のアサーション時間はマスタモードで 13 ビットに固定されています。

ショートフレーム同期では、WS 同期信号の長さは、わずか 1 サイクルです。

図 371. PCM 規格の波形（32 ビットパケットフレームに拡張された 16 ビット）

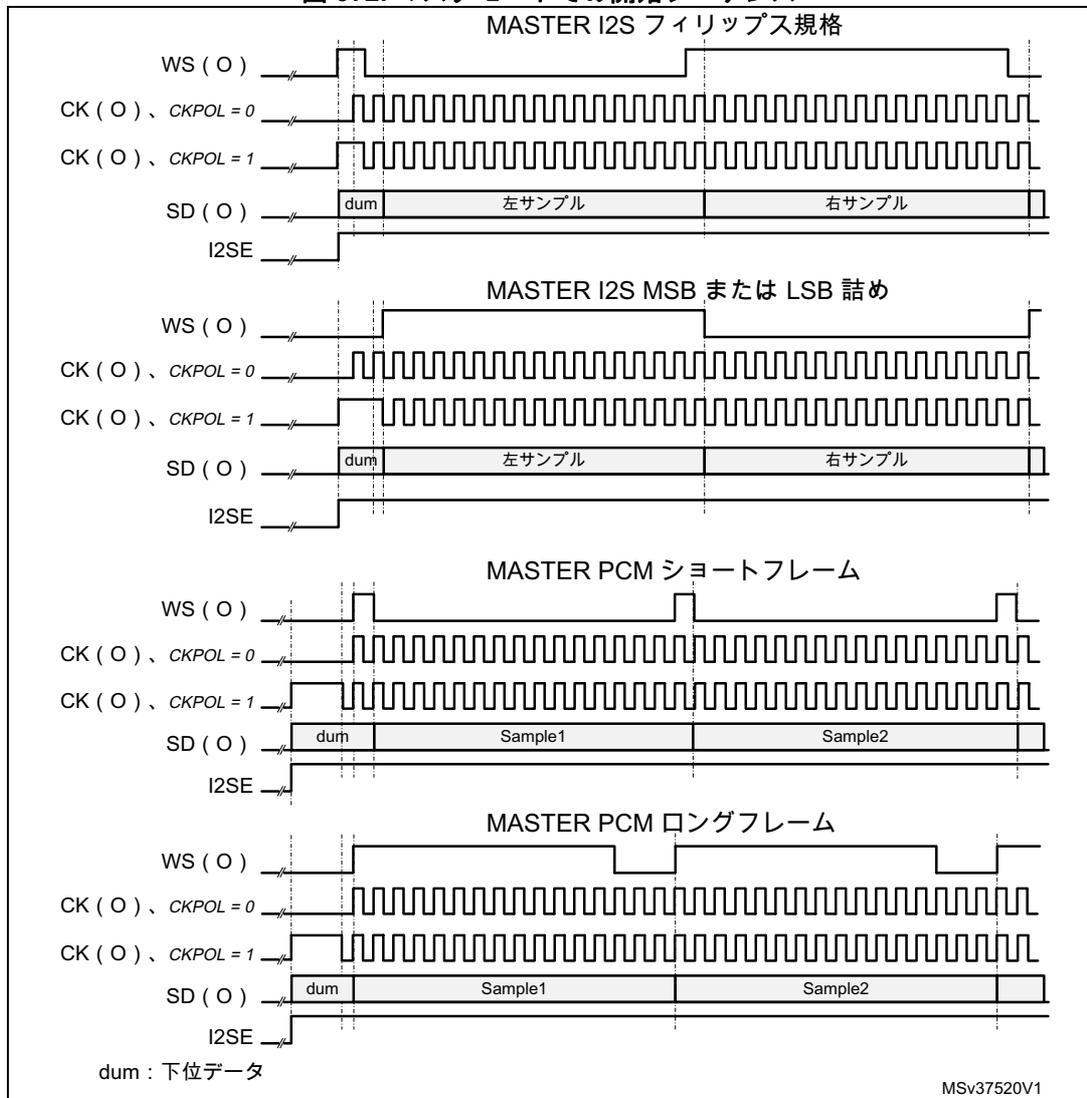


**注：** 2 つのモード（マスタとスレーブ）と 2 つの同期（ショートとロング）に関しては、スレーブモードでも、連続した 2 つのデータ（したがって 2 つの同期信号）間のビット数を（SPIx\_I2SCFGR レジスタの DATLEN および CHLEN ビットで）指定する必要があります。

## 32.7.3 起動に関する説明

図 372 は、SPI/I2S が有効化された場合 (I2SE ビット) の MASTER モードでのシリアルインタフェースの処理について示しています。生成された信号での CKPOL の影響についても示します。

図 372. マスタモードでの開始シーケンス



スレーブモードで、フレームの同期の検出方法は ASTRTEN ビットの値に依存します。

ASTRTEN = 0 の場合、オーディオインタフェースを有効 (I2SE = 1) にすると、ハードウェアは CK 信号を使用して受信 WS 信号で適切な遷移が行われるのを待ちます。

WS 信号での適切な遷移は、I2S フィリップス規格が使用されている場合は立ち下がりエッジ、それ以外の規格の場合は立ち上がりエッジです。立ち下がりエッジは、最初の WS を 1 にサンプリングし、続いて 0 にサンプリングすることで検出されます。立ち上がりエッジの検出方法はこの逆です。

ASTRTEN = 1 の場合、ユーザは WS がアクティブになる前にオーディオインタフェースを有効にする必要があります。これは、I2S フィリップス規格が WS = 1、またはその他の規格が WS = 0 の場合に、I2SE ビットを 1 にセットする必要があることを意味しています。

## 32.7.4 クロックジェネレータ

I<sup>2</sup>S ビットレートにより、I<sup>2</sup>S データライン上のデータフローと I<sup>2</sup>S クロック信号周波数が決まります。

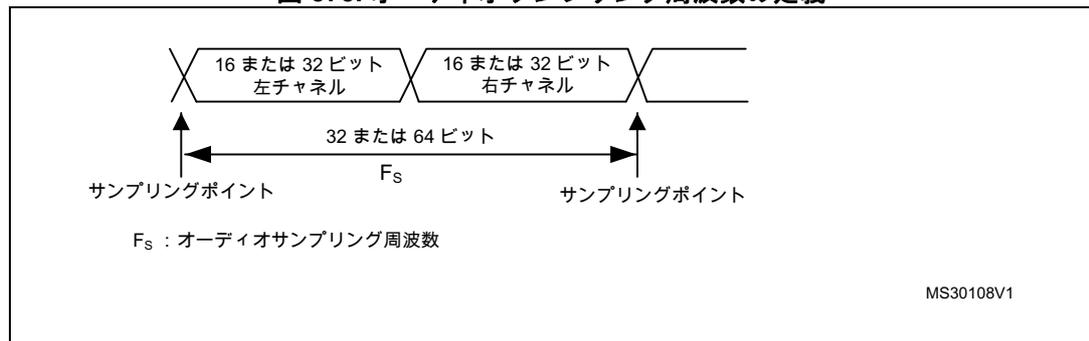
I<sup>2</sup>S ビットレート = チャンネルあたりのビット数 × チャンネル数 × オーディオサンプリング周波数

16 ビットオーディオ、左チャンネルおよび右チャンネルの場合、I<sup>2</sup>S ビットレートは次のように算出されます。

$$I^2S \text{ ビットレート} = 16 \times 2 \times f_s$$

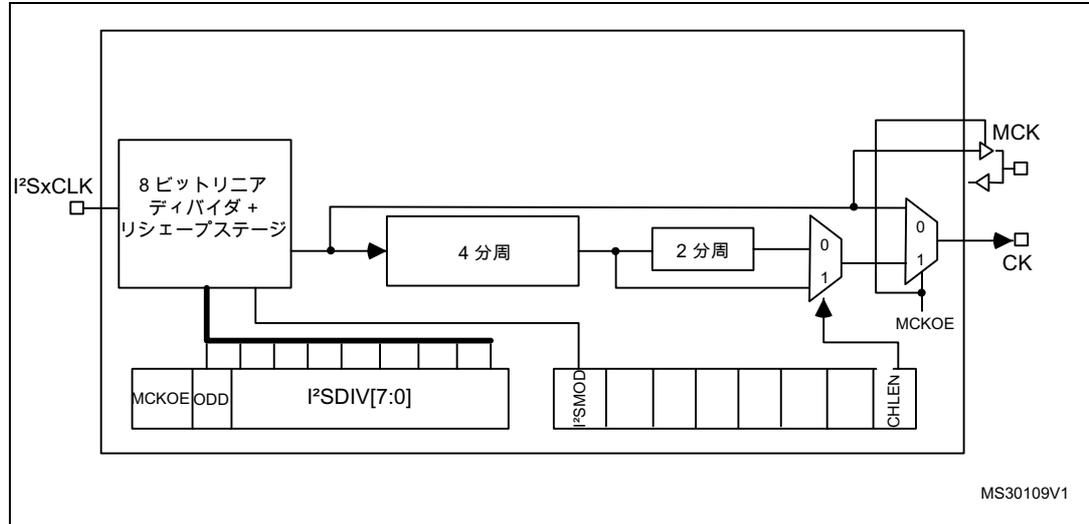
パケット長が 32 ビットの場合、I<sup>2</sup>S ビットレート = 32 × 2 × f<sub>s</sub> となります。

図 373. オーディオサンプリング周波数の定義



マスタモードが設定された場合、希望するオーディオ周波数で通信するために、特定の措置を講じてリニアディバイダを適切にプログラムする必要があります。

図 374. I<sup>2</sup>S クロックジェネレータのアーキテクチャ



1. x は 2 または 3 です。

図 374 に、通信クロックのアーキテクチャを示します。I2Sx クロックは常にシステムクロックです。

オーディオサンプリング周波数の取り得る値は、192 kHz、96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz、8 kHz（またはその他のこの範囲にある値）。希望する周波数を達成するには、次の式に基づいてリニアディバイダをプログラムする必要があります。

マスタクロックが生成される (SPIx\_I2SPR レジスタの MCKOE ビットをセットされる) 場合

$$f_S = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8], \text{ チャネルフレームが 16 ビット幅の場合}$$

$$f_S = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4], \text{ チャネルフレームが 32 ビット幅の場合}$$

マスタクロックが無効にされた場合 (MCKOE ビットをクリア)

$$f_S = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)], \text{ チャネルフレームが 16 ビット幅の場合}$$

$$f_S = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)], \text{ チャネルフレームが 32 ビット幅の場合}$$

表 177 に、さまざまなクロック設定に対する精度値の例を示します。

注: **最適クロック精度を可能にするその他の設定も可能です。**

表 177. 標準 8 MHz HSE を使用した場合のオーディオ周波数精度<sup>(1)</sup>

SYSCLK (MHz)	データ長	I2SDIV	I2SODD	MCLK	ターゲット fs (Hz)	実際の fs (Hz)	誤差
48	16	8	0	なし	96000	93750	2.3438%
48	32	4	0	なし	96000	93750	2.3438%
48	16	15	1	なし	48000	48387.0968	0.8065%
48	32	8	0	なし	48000	46875	2.3438%
48	16	17	0	なし	44100	44117.647	0.0400%
48	32	8	1	なし	44100	44117.647	0.0400%
48	16	23	1	なし	32000	31914.8936	0.2660%
48	32	11	1	なし	32000	32608.696	1.9022%
48	16	34	0	なし	22050	22058.8235	0.0400%
48	32	17	0	なし	22050	22058.8235	0.0400%
48	16	47	0	なし	16000	15957.4468	0.2660%
48	32	23	1	なし	16000	15957.447	0.2660%
48	16	68	0	なし	11025	11029.4118	0.0400%
48	32	34	0	なし	11025	11029.412	0.0400%
48	16	94	0	なし	8000	7978.7234	0.2660%
48	32	47	0	なし	8000	7978.7234	0.2660%
48	16	2	0	あり	48000	46875	2.3430%
48	32	2	0	あり	48000	46875	2.3430%
48	16	2	0	あり	44100	46875	6.2925%
48	32	2	0	あり	44100	46875	6.2925%
48	16	3	0	あり	32000	31250	2.3438%
48	32	3	0	あり	32000	31250	2.3438%
48	16	4	1	あり	22050	20833.333	5.5178%
48	32	4	1	あり	22050	20833.333	5.5178%
48	16	6	0	あり	16000	15625	2.3438%
48	32	6	0	あり	16000	15625	2.3438%

表 177. 標準 8 MHz HSE を使用した場合のオーディオ周波数精度<sup>(1)</sup> (続き)

SYSCCLK (MHz)	データ長	I2SDIV	I2SODD	MCLK	ターゲット fs (Hz)	実際の fs (Hz)	誤差
48	16	8	1	あり	11025	11029.4118	0.0400%
48	32	8	1	あり	11025	11029.4118	0.0400%
48	16	11	1	あり	8000	8152.17391	1.9022%
48	32	11	1	あり	8000	8152.17391	1.9022%

1. この表の値は、さまざまなクロック設定に対するほんの一例です。最適クロック精度を可能にするその他の設定も可能です。

## 32.7.5 I<sup>2</sup>S マスタモード

I<sup>2</sup>S はマスタモードで設定できます。つまり、シリアルクロックは、ワードセレクト信号 WS だけでなく、CK ピン上でも生成されます。マスタクロック (MCK) は、SPIx\_I2SPR レジスタの MCKOE ビットで、出力するかしないかを制御できます。

### 手順

- 適切なオーディオサンプリング周波数に到達するシリアルクロックポーレートを定義するため、SPIx\_I2SPR レジスタの I2SDIV[7:0] ビットを選択します。SPIx\_I2SPR レジスタの ODD ビットも定義する必要があります。
- 通信クロックの一定したレベルを定義するために、CKPOL ビットを選択します。外部の DAC/ADC オーディオコンポーネントにマスタクロック MCK を供給する必要がある場合、SPIx\_I2SPR レジスタの MCKOE ビットをセットします (I2SDIV と ODD の値は、MCK 出力の状態に応じて計算する必要があります。詳細については、[セクション 32.7.4 : クロックジェネレータ](#)を参照)。
- SPIx\_I2SCFGR レジスタの I2SMOD ビットをセットして I<sup>2</sup>S 機能を有効にし、I2SSTD[1:0] と PCMSYNC ビットにより I<sup>2</sup>S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してチャンネルあたりのビット数を、それぞれ選択します。SPIx\_I2SCFGR レジスタの I2SCFG[1:0] ビットにより I<sup>2</sup>S マスタモードと方向 (トランスミッタまたはレシーバ) も選択します。
- 必要な場合は、SPIx\_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
- SPIx\_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

WS と CK は出力モードに設定されます。SPIx\_I2SPR の MCKOE ビットがセットされている場合、MCK も出力です。

### 送信シーケンス

送信シーケンスは、Txバッファにハーフワードが書き込まれたときに開始されます。

Txバッファに書き込まれる最初のデータは、左チャンネルのデータに対応すると想定します。データが Txバッファからシフトレジスタに転送されると、TXE がセットされ、右チャンネルに対応するデータを Txバッファに書き込む必要があります。CHSIDE フラグは、どのチャンネルが送信されるかを示します。CHSIDE フラグは TXE がハイレベルになったときに更新されるため、このフラグは TXE フラグがセットされた場合に意味を持ちます。

完全なフレームとは、左チャンネルのデータ送信と、それに続く右チャンネルのデータ送信であるとみなす必要があります。左チャンネルのみが送信される部分的フレームは実現できません。



データハーフワードは、最初のビット送信時に 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx\_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

選択された I<sup>2</sup>S 規格モードに応じた書き込み動作の詳細については、[セクション 32.7.2: サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx\_DR レジスタに書きこむ必要があります。

I2SE をクリアすることによって I<sup>2</sup>S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

## 受信シーケンス

動作モードは、ポイント 3 を除いて、送信モードの場合と同じです ([セクション 32.7.5: I2S マスタモード](#)に示す手順を参照)。ポイント 3 では、I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットのパケットによって受信されます。つまり、Rxバッファが満杯になるたびに RXNE フラグがセットされ、SPIx\_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャンネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャンネルまたは左チャンネルに受信されることがあります。

RXNE ビットは、SPIx\_DR レジスタの読み出しによってクリアされます。

CHSIDE は毎回の受信後に更新されます。CHSIDE は、I<sup>2</sup>S セルによって生成される WS 信号に反応します。

選択された I<sup>2</sup>S 規格モードに応じた読み出し動作の詳細については、[セクション 32.7.2: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx\_CR2 レジスタの ERRIE ビットがセットされている場合、割り込みが生成されてエラーを知らせます。

I<sup>2</sup>S をスイッチオフするには、I<sup>2</sup>S が新しいデータ転送を開始することなく転送サイクルを適切に完了できるように、特定の動作が要求されます。そのシーケンスは、データ長とチャンネル長の設定、および選択したオーディオプロトコルモードに依存します。ケース別の説明

- 32 ビットチャンネル長に拡張された 16 ビットデータ長(DATLEN = 00、CHLEN = 1)、LSB 詰めモードを使用 (I2SSTD = 10)
  - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
  - b) 次に I<sup>2</sup>S の 17 クロックサイクルを待ちます (ソフトウェアループを使用)。
  - c) I<sup>2</sup>S を無効にします (I2SE = 0)。
- 32 ビットチャンネル長に拡張された 16 ビットデータ長(DATLEN = 00 および CHLEN = 1)、MSB 詰め、I<sup>2</sup>S または PCM モード (それぞれ、I2SSTD = 00、I2SSTD = 01、または I2SSTD = 11)
  - a) 最後の RXNE を待ちます。
  - b) 次に I<sup>2</sup>S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
  - c) I<sup>2</sup>S を無効にします (I2SE = 0)。
- DATLEN と CHLEN のその他すべての組み合わせについては、I2SSTD ビットを通じて選択したオーディオモードが何であれ、次のシーケンスを実行して I<sup>2</sup>S をスイッチオフします。
  - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
  - b) 次に I<sup>2</sup>S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
  - c) I<sup>2</sup>S を無効にします (I2SE = 0)。

**注：** 転送時、BSY フラグはローレベルに保持されます。

## 32.7.6 I<sup>2</sup>S スレーブモード

スレーブ設定の場合、I<sup>2</sup>S は送信または受信モードに設定できます。動作モードは、主として I<sup>2</sup>S マスタ設定に関して述べたものと同じ規則に従います。スレーブモードでは、I<sup>2</sup>S インタフェースによって生成されるクロックはありません。クロックと WS 信号は、I<sup>2</sup>S インタフェースに接続された外部マスタから入力されます。したがって、ユーザがクロックを設定する必要はありません。

設定ステップを次に示します。

1. SPIx\_I2SCFGR レジスタの I2SMOD ビットをセットして I<sup>2</sup>S モードを選択し、I2SSTD[1:0] ビットにより I<sup>2</sup>S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してフレームのチャンネルあたりのビット数を、それぞれ選択します。SPIx\_I2SCFGR レジスタの I2SCFG[1:0] ビットにより、スレーブのモード (送信または受信) も選択します。
2. 必要な場合は、SPIx\_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
3. SPIx\_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

### 送信シーケンス

送信シーケンスは、外部マスタデバイスがクロックを送信したときと、NSS\_WS 信号がデータの転送を要求したときに開始されます。スレーブを有効にしなければ、外部マスタは通信を開始しません。I<sup>2</sup>S データレジスタは、マスタが通信を開始する前に、ロードする必要があります。

I<sup>2</sup>S、MSB 詰め、および LSB 詰めモードの場合、データレジスタに書き込まれる最初のデータは、左チャンネルのデータに対応します。通信が開始されると、データは Tx バッファからシフトレジスタに転送されます。次に、右チャンネルのデータを I<sup>2</sup>S データレジスタに書き込むように要求するために、TXE フラグがセットされます。

CHSIDE フラグは、どのチャンネルが送信されるかを示します。スレーブモードでは、マスタ送信モードに比べて、CHSIDE は外部マスタからの WS 信号に反応します。つまり、スレーブが最初のデータの送信準備をしなければ、マスタはクロックを生成できません。WS アサーションは、最初に送信される左チャンネルに対応します。

**注：** *I<sup>2</sup>SE は、マスタの最初のクロックが CK ラインに到達するよりも、少なくとも 2 PCLK サイクル前に書き込まれる必要があります。*

データハーフワードは、最初のビット送信時に内部バスから 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx\_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

なお、Txバッファへの書き込みの前に、TXE フラグが 1 であることを確認する必要があります。

選択された I<sup>2</sup>S 規格モードに応じた書き込み動作の詳細については、[セクション 32.7.2: サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx\_DR レジスタに書きこむ必要があります。次のデータ通信の最初のクロックエッジよりも前にデータが SPIx\_DR レジスタに書き込まれない場合、アンダーランフラグがセットされ、割り込みが生成されることがあります。これによって、転送データに誤りがあることがソフトウェアに知らされます。SPIx\_CR2 レジスタの ERRIE ビットがセットされた場合、SPIx\_SR レジスタの UDR フラグがハイレベルになると割り込みが生成されます。この場合、I<sup>2</sup>S をスイッチオフし、左チャンネルからデータ転送をリスタートする必要があります。

I<sup>2</sup>SE をクリアすることによって I<sup>2</sup>S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

## 受信シーケンス

動作モードは、ポイント 1 を除いて、送信モードの場合と同じです ([セクション 32.7.6: I<sup>2</sup>S スレーブモード](#)に示す手順を参照)。ポイント 1 では、SPIx\_I2SCFGR レジスタの I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットのパケットによって受信されます。つまり、Rxバッファがフルになるたびに SPIx\_SR レジスタの RXNE フラグがセットされ、SPIx\_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャンネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャンネルまたは左チャンネルに受信されることがあります。

CHSIDE フラグは、SPIx\_DR レジスタから読み出されるデータが受信されるたびに更新されます。このフラグは、外部マスタコンポーネントによって管理される外部 WS ラインに反応します。

RXNE ビットは、SPIx\_DR レジスタの読み出しによってクリアされます。

選択された I<sup>2</sup>S 規格モードに応じた読み出し動作の詳細については、[セクション 32.7.2: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx\_CR2 レジスタの ERRIE ビットがセットされた場合、エラーを示すために割り込みが生成されます。

受信モードで I<sup>2</sup>S をスイッチオフするには、最後の RXNE = 1 を受信した直後に I<sup>2</sup>SE をクリアする必要があります。

**注：** *外部マスタコンポーネントには、オーディオチャンネルを介して 16 ビットまたは 32 ビットのパケットでデータを送受信する機能が必要です。*

## 32.7.7 I<sup>2</sup>S ステータスフラグ

アプリケーションが I<sup>2</sup>S バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

### ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます (このフラグへの書き込みは無効)。このフラグは I<sup>2</sup>S の通信レイヤーの状態を示します。

BSY がセットされると、I<sup>2</sup>S が通信中でビジーであることを示します。マスタ受信モード (I2SCFG=11) には 1 つの例外があり、BSY フラグは受信時にローレベルに保持されます。

ソフトウェアが I<sup>2</sup>S を無効にする必要がある場合、BSY フラグは転送終了を検出するために役立ちます。これによって、最後の転送データの破壊を回避します。このため、下記の手順を厳守する必要があります。

BSY フラグは転送が開始されるとセットされます。ただし、I<sup>2</sup>S がマスタ受信モードにある場合を除きます。

BSY フラグをクリアするタイミング

- 転送が完了したとき (ただし、通信が連続的だと思われるマスタ送信モードの場合を除く)
- I<sup>2</sup>S が無効化されたとき

通信が連続的な場合

- マスタ送信モードでは、BSY フラグはすべての転送期間を通じてハイレベルに保持されます。
- スレーブモードでは、BSY フラグは、各転送間で I<sup>2</sup>S の 1 クロックサイクルの間ローレベルになります。

**注：** 各データの送受信の処理には BSY フラグを使用しないでください。代わりに、TXE フラグと RXNE フラグを使用することをお勧めします。

### Txバッファエンプティフラグ (TXE)

このフラグがセットされると、Txバッファはエンプティ (空) であり、次に送信するデータをバッファにロードできることを示します。送信されるデータがすでにTxバッファに格納されているとき、TXE フラグはリセットされます。TXE フラグは、I<sup>2</sup>S が無効にされているときにもリセットされます (I2SE ビットがリセットされる)。

### Rxバッファノットエンプティ (RXNE)

このフラグがセットされると、Rxバッファに有効な受信データがあることを示します。このフラグは、SPIx\_DR レジスタが読み出されるとリセットされます。

### チャンネルサイドフラグ (CHSIDE)

送信モードでは、このフラグは TXE がハイレベルになるとリフレッシュされます。このフラグは、SD 上の転送データが属するチャンネルサイドを示します。スレーブ送信モードでアンダーランエラーイベントが発生した場合、このフラグは信頼できないため、通信を再開する前に、I<sup>2</sup>S をスイッチオフし、さらにスイッチオンする必要があります。

受信モードでは、このフラグは SPIx\_DR にデータが受信されるとリフレッシュされます。このフラグは、どちらのチャンネルサイドからデータが受信されたかを示します。なお、エラー (OVR など) が発生した場合、このフラグは無意味になるため、I<sup>2</sup>S を無効にし、さらに有効にすることによってリセットする必要があります (変更が必要な場合は設定する)。

このフラグは、PCM 規格では意味を持ちません (ショートとロングフレームの両モード)。

SPIx\_SR の OVR または UDR フラグがセットされ、SPIx\_CR2 の ERRIE ビットもセットされると、割り込みが生成されます。この割り込みをクリアするには、割り込みソースをいったんクリアした後で、SPIx\_SR ステータスレジスタを読み出します。

## 32.7.8 I<sup>2</sup>S エラーフラグ

I<sup>2</sup>S セルには 3 つのエラーフラグがあります。

### アンダーランフラグ (UDR)

スレーブ送信モードでは、ソフトウェアが SPIx\_DR にまだ値をロードしていない間に、データ送信用の最初のクロックが現れると、このフラグがセットされます。このフラグは、SPIx\_I2SCFGR レジスタの I2SMOD ビットがセットされると使用できます。SPIx\_CR2 レジスタの ERRIE ビットがセットされている場合は、割り込みを生成できます。

UDR ビットは、SPIx\_SR レジスタの読み出し動作によってクリアされます。

### オーバーランフラグ (OVR)

このフラグがセットされるのは、データが受信され、前のデータが SPIx\_DR レジスタからまだ読み出されていないときです。結果として、受信データは失われます。SPIx\_CR2 の ERRIE ビットがセットされている場合、割り込みが生成されることがあります。

この場合、受信バッファの内容は、送信側のデバイスからの新しい受信データによって更新されません。SPIx\_DR レジスタへの読み出し動作によって、前に正しく受信されたデータが返されます。それ以降に送信されたすべてのハーフワードは失われます。

OVR ビットをクリアするには、SPIx\_DR レジスタを読み出し、続けて SPIx\_SR レジスタに読み出しアクセスを行います。

### フレームエラーフラグ (FRE)

このフラグは、I<sup>2</sup>S がスレーブモードに設定された場合にのみハードウェアによってセットすることができます。このフラグは、スレーブが WS ラインの変化を想定していないときに外部マスタが WS ラインを変化させようとしている場合にセットされます。同期が失われた場合、この状態から回復し、外部マスタデバイスと I<sup>2</sup>S スレーブデバイスを再同期させるためには、次のステップに従います。

1. I<sup>2</sup>S を無効にします。
2. I2S インタフェースを再度有効にします (ASTRTEN = 0 のままとする)。

マスタデバイスとスレーブデバイス間の同期外れは、CK 通信クロック上または WS フレーム同期ライン上のノイズの多い環境に起因する可能性があります。ERRIE ビットをセットすれば、エラー割り込みを生成させることができます。同期外れフラグ (FRE) は、ステータスレジスタを読み出すときに、ソフトウェアでクリアされます。

## 32.7.9 DMA の機能

I<sup>2</sup>S モードでは、DMA は SPI モードと全く同じように機能します。異なる点は、I<sup>2</sup>S モードではデータ転送保護システムがないため、CRC 機能を使用できない点のみです。

## 32.8 I<sup>2</sup>S 割り込み

表 178 に I<sup>2</sup>S 割り込みのリストを示します。

表 178. I<sup>2</sup>S 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
送信バッファエンプティフラグ	TXE	TXEIE
受信バッファノットエンプティフラグ	RXNE	RXNEIE
オーバーランエラー	OVR	ERRIE
アンダーランエラー	UDR	
フレームエラーフラグ	FRE	

## 32.9 SPI および I<sup>2</sup>S レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。さらに、SPI\_DR へは8 ビット単位でアクセスできます。

### 32.9.1 SPI 制御レジスタ 1 (SPIx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15 **BIDIMODE** : 双方向データモードイネーブル。このビットは、共通の双方向データラインを 1 本使用して、半二重通信を有効にします。双方向モードがアクティブのときは、RXONLY ビットをクリアされたままにします。

0 : 2 線単方向データモードを選択します。

1 : 1 線双方向データモードを選択します。

**注 :** このビットは I<sup>2</sup>S モードでは使用しません。

ビット 14 **BIDIOE** : 双方向モードでの出力イネーブル

双方向モードでの転送方向は、このビットと BIDIMODE ビットを組み合わせで選択します。

0 : 出力は無効です (受信専用モード)。

1 : 出力は有効です (送信専用モード)。

**注 :** マスタモードでは MOSI ピンが使用され、スレーブモードでは MISO ピンが使用されます。  
このビットは I<sup>2</sup>S モードでは使用しません。

ビット 13 **CRCEN** : ハードウェア CRC 計算イネーブル

0 : CRC 計算は無効です。

1 : CRC 計算は有効です。

**注 :** 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。  
このビットは I<sup>2</sup>S モードでは使用しません。

ビット 12 **CRCNEXT** : 送信 CRC Next

0 : 次の送信値は Tx バッファから送信されます。

1 : 次の送信値は Tx CRC レジスタから送信されます。

**注 :** このビットは、最後のデータが SPIx\_DR レジスタに書き込まれた直後に書き込む必要があります。

このビットは I<sup>2</sup>S モードでは使用しません。

ビット 11 **CRCL** : CRC 長

このビットは、CRC 長をセットするために、ソフトウェアによってセット/クリアされます。

0 : 8 ビットの CRC 長

1 : 16 ビットの CRC 長

**注 :** 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。  
このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 10 RXONLY** : 受信専用モードイネーブル

このビットは、データ受信専用の単方向ラインを 1 本使用して、単方向通信を有効にします。受信専用モードがアクティブのときは、BIDIMODE ビットをクリアされたままにします。このビットはマルチスレーブシステムでも役立ちます。そのシステムでは、この特定のスレーブはアクセスされず、アクセスされたスレーブからの出力は破壊されません。

0 : 全二重 (送受信)

1 : 出力は無効です (受信専用モード)。

**注 :** このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 9 SSM** : ソフトウェアスレーブ管理

SSM ビットがセットされているとき、NSS ピンの入力は SSI ビットからの値に置き換えられます。

0 : ソフトウェアスレーブ管理は無効です。

1 : ソフトウェアスレーブ管理は有効です。

**注 :** このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 8 SSI** : 内部スレーブ選択

このビットは、SSM ビットがセットされているときにのみ有効です。このビットの値は強制的に NSS ピンに設定され、NSS ピンの I/O 値は無視されます。

**注 :** このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 7 LSBFIRST** : フレームフォーマット

0 : データは MSB ファーストとともに送信/受信されます。

1 : データは LSB ファーストとともに送信/受信されます。

**注 :** 1. このビットは、通信中には変更しないでください。

2. このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 6 SPE** : SPI イネーブル

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

**注 :** SPI を無効にするときは、SPI を無効にする手順 (1063 ページ) に記載されている手順に従ってください。

このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 5:3 BR[2:0]** : ボーレート制御

000 :  $f_{HCLK} / 2$

001 :  $f_{HCLK} / 4$

010 :  $f_{HCLK} / 8$

011 :  $f_{HCLK} / 16$

100 :  $f_{HCLK} / 32$

101 :  $f_{HCLK} / 64$

110 :  $f_{HCLK} / 128$

111 :  $f_{HCLK} / 256$

**注 :** これらのビットは、通信中には変更しないでください。

このビットは I<sup>2</sup>S モードでは使用しません。

ビット 2 **MSTR** : マスタ選択

- 0 : スレーブ設定
- 1 : マスタ設定

**注 :** このビットは、通信中には変更しないでください。  
このビットは I<sup>2</sup>S モードでは使用しません。

ビット 1 **CPOL** : クロック極性

- 0 : アイドル時に CK は 0 となります。
- 1 : アイドル時に CK は 1 となります。

**注 :** このビットは、通信中には変更しないでください。  
このビットは SPI TI モードでは使用しません。

ビット 0 **CPHA** : クロック位相

- 0 : 最初のクロック遷移が最初のデータキャプチャエッジです。
- 1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

**注 :** このビットは、通信中には変更しないでください。  
このビットは SPI TI モードでは使用しません。

## 32.9.2 SPI 制御レジスタ 2 (SPIx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXTH	DS [3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **LDMA\_TX** : 送信の最後の DMA 転送

このビットは、DMA で送信するデータの合計数が奇数であるか偶数であるかを定義するために、データバッキングモードで使用されます。これは、SPIx\_CR2 レジスタの TXDMAEN ビットがセットされている場合で、バッキングモードが使用されている場合にのみ意味を持ちます (データ長 ≤ 8 ビット、SPIx\_DR への書き込みアクセスは 16 ビット幅)。SPI が無効化された場合に書き込む必要があります (SPIx\_CR1 レジスタで SPE = 0)。

- 0 : 転送データ項目の数は偶数です。
- 1 : 転送データ項目の数は奇数です。

**注 :** **CRCEN** ビットがセットされている場合、**SPI** を無効にする手順 (1063 ページ) を参照してください。  
このビットは I<sup>2</sup>S モードでは使用しません。

ビット 13 **LDMA\_RX** : 受信の最後の DMA 転送

このビットは、DMA で受信するデータの合計数が奇数であるか偶数であるかを定義するために、データバッキングモードで使用されます。これは、SPIx\_CR2 レジスタの RXDMAEN ビットがセットされている場合で、バッキングモードが使用されている場合にのみ意味を持ちます (データ長 ≤ 8 ビット、SPIx\_DR への書き込みアクセスは 16 ビット幅)。SPI が無効化された場合に書き込む必要があります (SPIx\_CR1 レジスタで SPE = 0)。

- 0 : 転送データ項目の数は偶数です。
- 1 : 転送データ項目の数は奇数です。

**注 :** **CRCEN** ビットがセットされている場合、**SPI** を無効にする手順 (1063 ページ) を参照してください。  
このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 12 FRXTH** : FIFO 受信閾値

このビットは、RXNE イベントをトリガする RXFIFO の閾値をセットするために使用されます。

0 : FIFO レベルが 1/2 (16 ビット) 以上である場合に RXNE イベントが生成されます。

1 : FIFO レベルが 1/4 (8 ビット) 以上である場合に RXNE イベントが生成されます。

**注 :** *This bit is not used in I<sup>2</sup>S mode.*

**ビット 11:8 DS [3:0]** : データサイズ

以下のビットは SPI 転送のデータ長を設定します。

0000 : 未使用

0001 : 未使用

0010 : 未使用

0011 : 4 ビット

0100 : 5 ビット

0101 : 6 ビット

0110 : 7 ビット

0111 : 8 ビット

1000 : 9 ビット

1001 : 10 ビット

1010 : 11 ビット

1011 : 12 ビット

1100 : 13 ビット

1101 : 14 ビット

1110 : 15 ビット

1111 : 16 ビット

ソフトウェアが「未使用」値のいずれかの書き込みを試みた場合、値は強制的に「0111」(8 ビット)になります。

**注 :** *このビットは I<sup>2</sup>S モードでは使用しません。*

**ビット 7 TXEIE** : Txバッファエンプティ割り込みイネーブル

0 : TXE 割り込みはマスクされます。

1 : TXE 割り込みはマスクされません。TXE フラグがセットされたとき、割り込みリクエストの生成に使用されます。

**ビット 6 RXNEIE** : Rxバッファエンプティ割り込みイネーブル

0 : RXNE 割り込みはマスクされます。

1 : RXNE 割り込みはマスクされません。RXNE フラグがセットされたとき、割り込みリクエストの生成に使用されます。

**ビット 5 ERRIE** : エラー割り込みイネーブル

このビットは、エラー状態が発生したとき (SPI モードでは CRCERR、OVR、MODF ; TI モードでは FRE ; I<sup>2</sup>S モードでは UDR、OVR、FRE)、割り込みの生成を制御します。

0 : エラー割り込みはマスクされます。

1 : エラー割り込みは有効です。

**ビット 4 FRF** : フレームフォーマット

0 : SPI モトローラモード

1 : SPI TI モード

**注 :** *このビットは、SPI が無効 (SPE=0) のときにのみ書き込む必要があります。*

*このビットは I<sup>2</sup>S モードでは使用しません。*

**ビット 3 NSSP** : NSS パルス管理

このビットは、マスタモードでのみ使用されます。これにより、SPI は連続転送中に 2 つの連続したデータ間で NSS パルスを生成できます。単一のデータ転送の場合、転送後、NSS ピンは強制的にハイレベルになります。

CPHA = '1' の場合も FRF = '1' の場合も意味を持ちません。

- 0 : NSS パルスなし
- 1 : NSS パルス発生回路

**注:** 1.このビットは、SPI が無効 (SPE=0) のときのみ書き込む必要があります。  
2.このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 2 SSOE** : SS 出カインーブル

- 0 : マスタモードで SS 出力は無効にされ、SPI インタフェースはマルチマスタ設定で機能できます。
- 1 : SPI インタフェースが有効であるとき、マスタモードで SS 出力は有効です。SPI インタフェースはマルチマスタ環境では機能できません。

**注:** このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 1 TXDMAEN** : Txバッファ DMA イネーブル

このビットがセットされると、TXE フラグがセットされるたびに DMA リクエストが生成されます。

- 0 : Txバッファ DMA は無効です。
- 1 : Txバッファ DMA は有効です。

**ビット 0 RXDMAEN** : Rxバッファ DMA イネーブル

このビットがセットされると、RXNE フラグがセットされるたびに DMA リクエストが生成されます。

- 0 : Rxバッファ DMA は無効です。
- 1 : Rxバッファ DMA は有効です。

## 32.9.3 SPI ステータスレジスタ (SPIx\_SR)

アドレスオフセット : 0x08

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTLVL[1:0]		FRLVL[2:0]		FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0			r	r

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:11 **FTLVL[1:0]** : FIFO 送信レベル

これらのビットは、ハードウェアによってセット/クリアされます。

- 00 : FIFO エンプティ
- 01 : 1/4 FIFO
- 10 : 1/2 FIFO
- 11 : FIFO フル (FIFO 閾値が 1/2 より大きい場合、FULL とみなす)

**注 :** これらのビットは I<sup>2</sup>S モードでは使用されません。

ビット 10:9 **FRLVL[1:0]** : FIFO 受信レベル

これらのビットは、ハードウェアによってセット/クリアされます。

- 00 : FIFO エンプティ
- 01 : 1/4 FIFO
- 10 : 1/2 FIFO
- 11 : FIFO フル

**注 :** これらのビットは、CRC 計算が有効なときは I<sup>2</sup>S モードや SPI 受信専用モードでは使用されません。

ビット 8 **FRE** : フレームフォーマットエラー

このフラグは、TI スレープモードと I<sup>2</sup>S スレープモードの SPI で使用されます。[セクション 32.5.10 : SPI エラーフラグ](#)および[セクション 32.7.8 : I2S エラーフラグ](#)を参照してください。

このフラグは、ハードウェアによってセットされ、SPIx\_SR がソフトウェアによって読み出されるとリセットされます。

- 0 : フレームフォーマットエラーはありません。
- 1 : フレームフォーマットエラーが発生しました。

ビット 7 **BSY** : ビジーフラグ

- 0 : SPI (または I2S) はビジー状態ではありません。
  - 1 : SPI (または I2S) が通信ビジー状態であるか、または Tx バッファが空ではありません。
- このフラグはハードウェアによってセット/クリアされます。

**注 :** BSY フラグを使用する際は注意が必要です。[セクション 32.5.9 : SPI ステータスフラグ](#)および[SPI を無効にする手順 \(1063 ページ\)](#)を参照してください。

ビット 6 **OVR** : オーバーランフラグ

- 0 : オーバーランは発生していません。
- 1 : オーバーランが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[I2S エラーフラグ \(1094 ページ\)](#)を参照してください。

ビット 5 **MODF** : モードフォールト

- 0 : モードフォールトは発生していません。
- 1 : モードフォールトが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[セクション : モードフォールト \(MODF\) \(1073 ページ\)](#)を参照してください。

**注：** このビットはI<sup>2</sup>S モードでは使用しません。

ビット4 **CRCERR** : CRC エラーフラグ

- 0 : 受信した CRC 値が SPIx\_RXCRCR 値と一致します。
- 1 : 受信した CRC 値が SPIx\_RXCRCR 値と一致しません。

このフラグは、ハードウェアによってセットされ、ソフトウェアによって 0 を書き込むことでクリアされます。

**注：** このビットはI<sup>2</sup>S モードでは使用しません。

ビット3 **UDR** : アンダーランフラグ

- 0 : アンダーランは発生していません。
- 1 : アンダーランが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[I2S エラーフラグ \(1094 ページ\)](#) を参照してください。

**注：** このビットはSPI モードでは使用しません。

ビット2 **CHSIDE** : チャネルサイド

- 0 : 左チャネルを送信する必要があるか、または受信が行われました。
- 1 : 右チャネルを送信する必要があるか、または受信が行われました。

**注：** このビットはSPI モードでは使用しません。PCM モードでは意味を持ちません。

ビット1 **TXE** : 送信バッファエンプティ

- 0 : Txバッファは空ではありません。
- 1 : Txバッファは空です。

ビット0 **RXNE** : 受信バッファノットエンプティ

- 0 : Rxバッファは空です。
- 1 : Rxバッファは空ではありません。

## 32.9.4 SPI データレジスタ (SPIx\_DR)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **DR[15:0]** : データレジスタ

受信したデータまたは送信されるデータ

このデータレジスタは、Rx および Tx FIFO 間のインタフェースとして使用できます。データレジスタが読み出されると、データレジスタへの書き込みが TxFIFO にアクセスしている間に、RxFIFO にアクセスされます ([セクション 32.5.8 : データの送受信手順](#)を参照)。

**注：** データは常に右詰めです。未使用のビットは、レジスタへの書き込み時に無視され、レジスタの読み出し時にゼロとして読み出されます。Rx 閾値設定は、常に現在使用中の読み出しアクセスに対応している必要があります。

## 32.9.5 SPI CRC 多項式レジスタ (SPIx\_CRCPR)

アドレスオフセット : 0x10

リセット値 : 0x0007

CRCPOLY[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CRCPOLY[15:0]** : CRC 多項式レジスタ

このレジスタは、CRC 計算用の多項式を格納します。

CRC 多項式 (0007h) は、このレジスタのリセット値です。必要に応じて、別の多項式を設定することができます。

**注 :** 多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。

## 32.9.6 SPI Rx CRC レジスタ (SPIx\_RXCRCR)

アドレスオフセット : 0x14

リセット値 : 0x0000

RxCRC[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **RxCRC[15:0]** : Rx CRC レジスタ

CRC 計算が有効なとき、RxCRC[15:0] ビットには、その後受信したバイトから算出された CRC 値が格納されています。このレジスタは、SPIx\_CR1 レジスタの CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPIx\_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPIx\_CR1 の CRCL ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPIx\_CR1 レジスタの CRCL ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

**注 :** BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。

これらのビットは I<sup>2</sup>S モードでは使用されません。

## 32.9.7 SPI Tx CRC レジスタ (SPIx\_TXCRCR)

アドレスオフセット : 0x18

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **TxCRC[15:0]** : Tx CRC レジスタ

CRC 計算が有効なとき、TxCRC[7:0] ビットには、その後に送信されたバイトから算出された CRC 値が格納されます。このレジスタは、SPIx\_CR1 の CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPIx\_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPIx\_CR1 の CRCL ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPIx\_CR1 レジスタの CRCL ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

**注:** *BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。*

*これらのビットは I<sup>2</sup>S モードでは使用されません。*

## 32.9.8 SPIx\_I<sup>2</sup>S 設定レジスタ (SPIx\_I2SCFGR)

アドレスオフセット : 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	ASTRTEN	I2SMOD	I2SE	I2SCFG		PCMSYNC	Res.	I2SSTD		CKPOL	DATLEN		CHLEN
			r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:13 予約済み。ハードウェアによって 0 に固定されています。

ビット 12 **ASTRTEN** : 非同期開始イネーブル

0 : 非同期開始は無効です。

I2S がスレーブモードで有効な場合、ハードウェアは I2S クロックを受信し、WS 信号で適切な遷移を検出すると、転送を開始します。

1 : 非同期開始は有効です。

I2S がスレーブモードで有効な場合、ハードウェアは I2S クロックを受信し、WS 信号で適切なレベルを検出すると、転送を開始します。

**注:** *WS 信号での適切な遷移は、I2S フィリップス規格が使用されている場合は立ち下がリエッジ、それ以外の規格の場合は立ち上がりエッジです。*

*WS 信号での適切なレベルは、I2S フィリップス規格が使用されている場合はローレベル、それ以外の規格の場合はハイレベルです。*

*詳細については、セクション 32.7.3 : 起動に関する説明を参照してください。*

ビット 11 **I2SMOD** : I2S モード選択

0 : SPI モードが選択されます。

1 : I2S モードが選択されます。

**注:** *このビットは SPI が無効なときに設定してください。*

ビット 10 **I2SE** : I2S イネーブル

0 : I2S ペリフェラルは無効です。

1 : I2S ペリフェラルは有効です。

**注 :** このビットは SPI モードでは使用しません。

ビット 9:8 **I2SCFG** : I2S 設定モード

00 : スレーブ - 送信

01 : スレーブ - 受信

10 : マスタ - 送信

11 : マスタ - 受信

**注 :** これらのビットは、I2S が無効なときに設定してください。

これらは SPI モードでは使用しません。

ビット 7 **PCMSYNC** : PCM フレーム同期

0 : ショートフレーム同期

1 : ロングフレーム同期

**注 :** このビットは、I2SSTD = 11 (PCM 規格使用) の場合にのみ意味を持ちます。

SPI モードでは使用しません。

ビット 6 予約済みであり、ハードウェアによって 0 に固定されています。

ビット 5:4 **I2SSTD** : I2S 規格選択

00 : フィリップス I2S 規格

01 : MSB 詰め規格 (左詰め)

10 : LSB 詰め規格 (右詰め)

11 : PCM 規格

I2S 規格に関する詳細は、[セクション 32.7.2 \(1079 ページ\)](#) を参照してください。

**注 :** 正しい動作のためには、これらのビットは、I2S が無効のときに設定してください。

これらは SPI モードでは使用しません。

ビット 3 **CKPOL** : インアクティブ状態のクロック極性

0 : I2S クロックのインアクティブ状態はローレベルです。

1 : I2S クロックのインアクティブ状態はハイレベルです。

**注 :** 正しい動作のためには、このビットは、I2S が無効のときに設定してください。

SPI モードでは使用しません。

CKPOL ビットは、SD 信号および WS 信号を受信または送信するために使用される CK エッジ感度には影響を与えません。

ビット 2:1 **DATLEN** : 転送されるデータ長

00 : 16 ビットデータ長

01 : 24 ビットデータ長

10 : 32 ビットデータ長

11 : 設定禁止

**注 :** 正しい動作のためには、これらのビットは、I2S が無効のときに設定してください。

これらは SPI モードでは使用しません。

ビット 0 **CHLEN** : チャネル長 (オーディオチャネルごとのビット数)

0 : 16 ビット幅

1 : 32 ビット幅

ビット書き込み動作は、DATLEN=00 のときにのみ意味を持ちます。そうでない場合、書き込まれた値とは無関係に、チャネル長はハードウェアによって 32 ビットに固定されます。

**注 :** 正しい動作のためには、このビットは、I2S が無効のときに設定してください。

SPI モードでは使用しません。

## 32.9.9 SPIx\_I2S プリスケアラレジスタ (SPIx\_I2SPR)

アドレスオフセット : 0x20

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV							
						rw	rw	rw							

ビット 15:10 予約済み。ハードウェアによって 0 に固定されています。

ビット 9 **MCKOE** : マスタクロック出カインエーブル

0 : マスタクロック出力は無効です。

1 : マスタクロック出力は有効です。

**注 :** このビットは、I<sup>2</sup>S が無効なときに設定してください。これらのビットは、I<sup>2</sup>S がマスタモードのときのみ使用します。

**SPI モードでは使用しません。**

ビット 8 **ODD** : プリスケアラの奇数分周比

0 : 真のディバイダ値 = I2SDIV \* 2

1 : 真のディバイダ値 = (I2SDIV \* 2) + 1

[セクション 32.7.3 \(1086 ページ\)](#) を参照してください。

**注 :** このビットは、I<sup>2</sup>S が無効なときに設定してください。これらのビットは、I<sup>2</sup>S がマスタモードのときのみ使用します。

**SPI モードでは使用しません。**

ビット 7:0 **I2SDIV** : I<sup>2</sup>S リニアプリスケアラ

I2SDIV [7:0] = 0 または I2SDIV [7:0] = 1 は禁止されている値です。

[セクション 32.7.3 \(1086 ページ\)](#) を参照してください。

**注 :** これらのビットは、I<sup>2</sup>S が無効なときに設定してください。これらのビットは、I<sup>2</sup>S がマスタモードのときのみ使用します。

**これらは SPI モードでは使用しません。**

## 32.9.10 SPI/I2S レジスタマップ

表 179 に、SPI/I2S レジスタマップとリセット値を示します。

表 179. SPI レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SPIx_CR1	Res.	BIDIMODE	BIDIOE	CRCE	CRCEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]		MSTR	CPOL	CPHA																
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	SPIx_CR2	Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]			TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	CPOL	CPHA																
	リセット値																			0	0	0	0	1	1	1	0	0	0	0	0	0	0
0x08	SPIx_SR	Res.	FTLV[1:0]	FRLV[1:0]			FRE	BSY	OVR	MODF	CRCE	UDR	CHSIDE	TXE	RXNE																		
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x0C	SPIx_DR	Res.	DR[15:0]																														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SPIx_CRCPR	Res.	CRCPOLY[15:0]																														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0x14	SPIx_RXCRCR	Res.	RxCRC[15:0]																														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SPIx_TXCRCR	Res.	TxCRC[15:0]																														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	SPIx_I2SCFGR	Res.	ASTRTEN	I2SMOD	I2SE	I2SCFG	PCMSYNC	Res.	I2SSTD	CKPOL	DATLEN	CHLEN																					
	リセット値																			0	0	0	0	0		0	0	0	0	0			
0x20	SPIx_I2SPR	Res.	Res.	Res.	MCKOE	ODD	I2SDIV																										
	リセット値																						0	0	0	0	0	0	0	0	0	1	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 33 シリアルオーディオインタフェース (SAI)

### 33.1 概要

この SAI インタフェース (シリアルオーディオインタフェース) は、柔軟性があり幅広い構成が可能のため、多様なオーディオプロトコルに対応できます。多くのステレオまたはモノラルオーディオアプリケーションが対象になります。例えば、I2S 規格、LSB/MSB 詰め、PCM/DSP、TDM、AC'97 の各プロトコルに対応することができます。オーディオブロックがトランスミッタとして設定された場合、SPDIF 出力が提供されます。

このレベルの柔軟性と再設定機能を実現するため、SAI は独立した 2 つのオーディオサブブロックを内蔵しています。各ブロックには専用のクロックジェネレータと I/O ラインコントローラが備わっています。

この SAI は、マスタまたはスレーブの設定で動作することができます。このオーディオサブブロックはレシーバにもトランスミッタにもなることができ、他の一方に対して同期または非同期で動作することができます。

SAI を他の SAI と接続することで、互いに同期して動作することができます。

## 33.2 SAI の主な機能

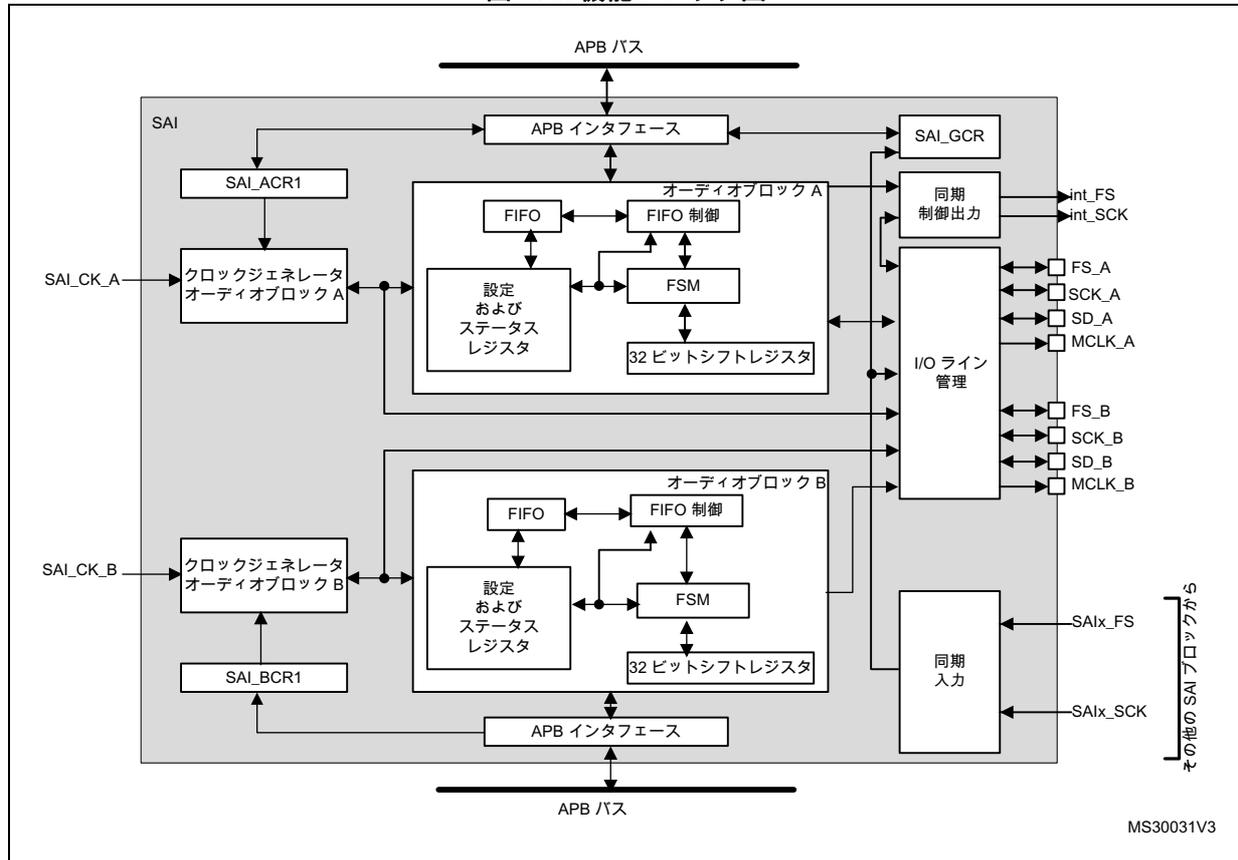
- 個別に FIFO を内蔵し、トランスミッタまたはレシーバとなることができる 2 つの独立したオーディオサブブロック。
- 各オーディオサブブロックに 8 ワード FIFO を内蔵。
- オーディオサブブロック間での同期モードまたは非同期モードが可能。
- 複数の SAI 間の同期が可能。
- 両オーディオサブブロックに対して独立してマスタ設定またはスレーブ設定が可能。
- 両オーディオサブブロックをマスタモードに設定した場合に、個別のオーディオサンプリング周波数に対応する各オーディオブロック用クロックジェネレータ。
- 設定可能なデータサイズ : 8、10、16、20、24、32 ビット
- オーディオプロトコル : I2S、LSB/MSB 詰め、PCM/DSP、TDM、AC'97
- 必要に応じて、SPDIF 出力を利用可能。
- サイズ設定が可能な最大 16 個のロットが利用可能。
- フレームごとのビット数が設定可能。
- フレーム同期アクティブレベルが設定可能 (オフセット、ビット長、レベル)。
- スロット内の先頭アクティブビット位置が設定可能。
- データ転送における LSB ファーストまたは MSB ファーストが可能。
- ミュートモード。
- ステレオ/モノラルオーディオフレーム機能。
- 通信クロックストローブエッジが設定可能 (SCK)。
- 対応する割り込み付きのエラーフラグ (それぞれイネーブル付き)
  - オーバーラン検出およびアンダーラン検出、
  - スレーブモードにおける予測フレーム同期信号検出、
  - スレーブモードにおけるレイトフレーム同期信号検出、
  - 受信時 AC'97 モードに対するコーデックノットレディ
- 割り込みソース (有効化時) :
  - エラー
  - FIFO リクエスト
- 2 チャンネル DMA インタフェース

## 33.3 SAI の機能詳細

### 33.3.1 SAI ブロック図

図 375に SAI ブロック図を示します。

図 375. 機能ブロック図



この SAI は大部分が 2 つのオーディオサブブロックから構成されており、それぞれに固有のクロックジェネレータを装備しています。各オーディオブロックは、それぞれの機能ステートマシンによって制御される 32 ビットシフトレジスタを内蔵しています。データは、専用 FIFO に保存されたり、そこから読み出されたりします。FIFO は CPU からアクセスすることもできれば、通信時に CPU を開放するため DMA からアクセスすることもできます。各オーディオブロックは独立しています。互いに同期することもできます。

I/O ラインコントローラは、SAI 内の指定オーディオブロックに対する 4 本の専用ピン (SD、SCK、FS、MCLK) を管理します。この 2 つのサブブロックを同期するように指定すると、いくつかのピンを共用していくつかのピンを開放し、汎用 I/O として使用することができます。アプリケーション、デコーダ要件、オーディオブロックがマスタとして設定されているか否かによって、MCLK ピンを出力にしたり、それ以外にしたりすることができます。

1 つの SAI が別の SAI と同期して動作するように設定されている場合、より多くの I/O を開放することができます (SD\_x ピンを除く)。

この機能ステートマシンは、広範囲なオーディオプロトコルを処理するように設定することができます。所望のプロトコルを設定するため、いくつかのレジスタがあります (オーディオフレーム波形ジェネレータ)。

オーディオサブブロックは、マスタモードまたはスレーブモードでトランスミッタまたはレシーバとなり得ます。マスタモードでは、SCK\_x ビットクロックとフレーム同期信号が SAI で生成されますが、スレーブモードでは、これらの信号を別の外部マスタまたは内部マスタから受け取ります。FS 信号の方向がマスタモードまたはスレーブモードの定義に直接結びつかない特別なケースがあります。AC'97 プロトコルでは、SAI (リンクコントローラ) が SCK クロックを入力するように設定されている場合でも (つまりスレーブモードでも)、FS 信号は SAI の出力になります。

**注 :** *読みやすくするために、このセクションでは SAI\_A または SAI\_B を SAI\_x と表記しています。「x」は SAI A サブブロックまたは SAI B サブブロックのことで、*

## 33.3.2 主要な SAI モード

SAI の各オーディオサブブロックは、選択したオーディオブロックの SAI\_xCR1 レジスタの MODE ビットを使ってマスタまたはスレーブに設定することができます。

### マスタモード

マスタモードでは、SAI は接続された外部デバイスへタイミング信号を送り出します。

- ビットクロックとフレーム同期はそれぞれ SCK\_x ピンと FS\_x ピンに出力されます。
- 必要であれば、SAI は MCLK\_x ピンにマスタクロックを生成することもできます。

SCK\_x、FS\_x、および MCLK\_x のいずれも出力として設定されます。

### スレーブモード

SAI は、外部デバイスからのタイミング信号の受信を待ちます。

- SAI サブブロックが非同期モードに設定されている場合、SCK\_x ピンおよび FS\_x ピンは入力として設定されます。
- SAI サブブロックが他の SAI インタフェースまたは 2 番目のオーディオサブブロックと同期して動作するように設定されている場合、対応する SCK\_x および FS\_x ピンは汎用 I/O として使用できるよう空けておきます。

MCLK\_x ピンはスレーブモードでは使用されず、別の機能に割り当てることができます。

スレーブデバイスを有効にした後でマスタを有効にすることをお勧めします。

### SAI モードの設定と有効化

各オーディオサブブロックは、該当するオーディオブロックにある SAI\_xCR1 レジスタの MODE ビットを使って個別にトランスミッタまたはレシーバに指定することができます。そのため、SAIx\_SD ピンは出力または入力としてそれぞれ設定されます。

同じ SAI にある 2 個のマスタオーディオブロックは、2 つの異なる MCLK および SCK クロック周波数で設定することができます。この場合、オーディオブロックは非同期モードに設定する必要があります。

SAI 内の各オーディオブロックは、SAI\_xCR1 レジスタの SAIXEN ビットにより有効化されます。このビットがアクティブになると直ちに、トランスミッタまたはレシーバはスレーブモードでクロックライン、データライン、同期ラインの動作に反応するようになります。

マスタ TX モードでは、オーディオブロックを有効化すると直ちに、FIFO にデータがなくとも、外部スレーブに対するビットクロックを生成しますが、FS 信号の生成は、FIFO 内のデータの存在が条件となります。FIFO が先頭の送信データを受け取った後、このデータが外部スレーブに出力されます。FIFO に送信データがない場合、オーディオフレームにおいては 0 値が送信されて、アンダーランフラグが生成されます。

スレーブモードでは、オーディオブロックが有効化されて、フレームの開始が検出されたときに、オーディオフレームが開始されます。

スレーブ TX モードでは、オーディオブロックを有効化した後の最初のフレームでアンダーランイベントが生成されることはありません。理由は、次の動作シーケンスが必要となるためです。

1. ソフトウェアまたは DMA により SAI\_xDR へ書き込みを行います。
2. FIFO 閾値 (FLH) フラグが 000b (FIFO エンプティ) でなくなるまで待ちます。
3. スレーブ送信ミッタモードでオーディオブロックを有効化します。

### 33.3.3 SAI 同期モード

オーディオサブブロックレベルと SAI レベルの 2 種類の同期レベルがあります。

#### 内部同期

オーディオサブブロックは、同じ SAI 内の 2 番目のオーディオサブブロックと同期して動作できるよう設定できます。この場合、ピットクロック信号とフレーム同期信号が共用され、通信に使用する外部ピン数を少なくすることができます。同期モードに設定されたオーディオブロックでは、その SCK\_x、FS\_x、および MCLK\_x ピンは GPIO として開放状態に戻されています。非同期モードに設定されたオーディオブロックのほうが、FS\_x、SCK\_x、MCLK\_x の I/O ピンに適したブロックとなります (オーディオブロックをマスタと考えた場合)。

一般に、同期モードのオーディオブロックを使って、SAI を全二重モードに設定することができます。2 個のオーディオブロックの片方をマスタとして設定し、他方をスレーブとして設定することができます。あるいは、両方をスレーブとして設定し、1 つは非同期ブロック (SAI\_xCR1 の対応する SYNCEN[1:0] ビットを 00 にセット) に、もう一方を同期ブロック (SAI\_xCR1 の対応する SYNCEN[1:0] ビットを 01 にセット) にすることもできます。

**注：** *内部再同期ステージのため、PCLK APB 周波数はピットレートクロック周波数の 2 倍以上である必要があります。*

#### 外部同期

オーディオサブブロックは他の SAI と同期して動作するよう設定することもできます。この方法は以下の通りです。

1. SAI が、他の SAI の同期元として設定されている場合は、その相手の SAI に FS および SCK 信号を提供するオーディオサブブロックを定義する必要があります。これは SYNCOUT[1:0] ビットを設定することによって行います。
2. 同期信号を受信する側の SAI は、SYNCIN[1:0] ビットに適正な値をセットすることによって同期を提供してくれる SAI を選択する必要があります。2 個の SAI オーディオサブブロックのそれぞれに対して、ユーザはそのサブブロックが他方の SAI と同期して動作するかどうかを SYNCEN ビットで指定します。

**注：** *SYNCIN[1:0] および SYNCOUT[1:0] ビットは SAI\_GCR レジスタ内にあり、SYNCEN ビットは SAI\_xCR1 レジスタ内にあります。*

SAI の 2 個のオーディオサブブロックがどちらも他の SAI と同期して動作する必要がある場合は、次の設定のいずれかを選択することができます。

- 他の SAI ブロックと同期させる各オーディオブロックを SYNCEN[1:0] ビットにより設定します。
- 他の SAI と同期させる 1 個のオーディオブロックを SYNCEN[1:0] ビットにより設定します。このとき、もう一方のオーディオブロックは 2 番目の SAI オーディオブロックと同期させるように、SYNCEN[1:0] ビットによって設定されます。

下表は、使用される SAI ブロックに応じた適切な同期信号の選び方を示します。たとえば、SAI2 は SAI2 SYNCIN を 0 に設定することで、SAI1 からの同期を選択できます。SAI1 に SAI2 からの同期を選択させたい場合は、SAI1 SYNCIN を 1 にセットする必要があります。「res」と記載されている位置は使用されていません。

表 180. 外部同期の選択

ブロックの例	SYNCIN = 3	SYNCIN = 2	SYNCIN = 1	SYNCIN = 0
SAI1	Res.	Res.	SAI2 同期	Res.
SAI2	Res.	Res.	Res.	SAI1 同期

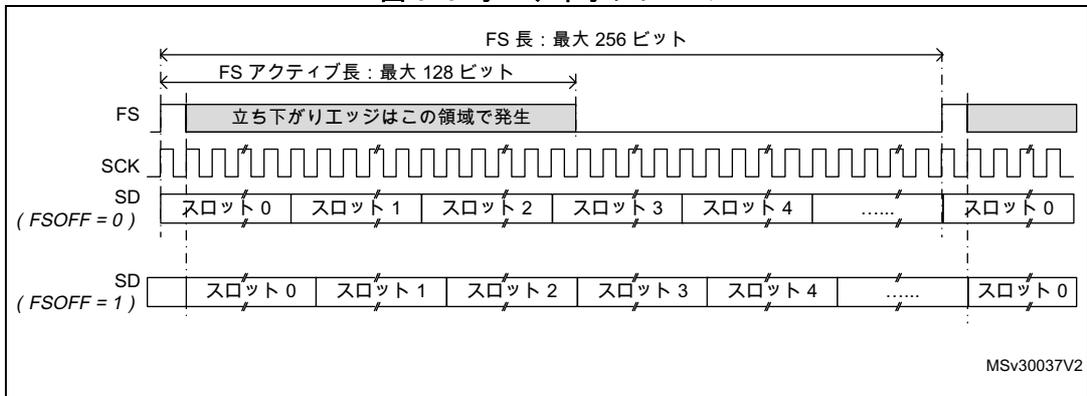
### 33.3.4 オーディオデータサイズ

オーディオフレームは、SAI\_xCR1 レジスタの DS[2:0] ビットを設定することにより、様々なデータサイズを対象とすることができます。データサイズとしては、8、10、16、20、24、32 ビットが可能です。転送時、SAI\_xCR1 レジスタの LSBFIRST ビットの設定によってデータの MSB を先頭にして送信することもできれば LSB を先頭にすることもできます。

### 33.3.5 フレーム同期

FS 信号は、オーディオフレームにおいてフレーム同期信号として機能します（フレーム開始）。この信号の波形は、このフレーム同期動作に関する独自の仕様を持つ様々なオーディオプロトコルに対応するため、全面的に設定可能となっています。この再設定は SAI\_xFRCR レジスタを使って実行されます。図 376 にこの柔軟性について図示します。

図 376. オーディオフレーム



AC'97 モード または SPDIF モード (SAI\_xCR1 レジスタの PRTCFCF[1:0] ビット = 10 または PRTCFCF[1:0] ビット = 01) では、フレーム同期波形を強制的に AC'97 プロトコルに適合させます。SAI\_xFRCR レジスタ値は無視されます。

各オーディオブロックは独立であるため、それぞれに固有の設定が必要となります。

## フレーム長

- マスタモード

オーディオフレーム長は、SAI\_xFRCR レジスタの FRL[7:0] ビットを設定して、最大 256 ビットクロックに設定することができます。

フレーム長がフレームに対して指定されたスロット数より大きい場合、SAI\_xCR2 レジスタの TRIS ビットの状態に応じて、残りの送信ビットが 0 に拡張されるか、または SD ラインがハイインピーダンス (HI-Z) 状態になります ([セクション : FS 信号の機能](#)を参照)。受信モードでは、残りのビットは無視されます。

NODIV ビットがクリアされている場合、(FRL + 1) は 8 から 256 のうちの 2 の累乗の数に調整する必要があります。これは、オーディオフレームがビットクロックサイクル当たり整数個の MCLK パルスを含むようにするためです。

NODIV ビットがセットされている場合、(FRL + 1) フィールドは 8 から 256 のうちの任意の値をとることができます。[セクション 33.3.7 : SAI クロックジェネレータ](#)を参照してください。

- スレーブモード

オーディオフレーム長は主に、外部マスタから送信されるオーディオフレーム当たりのビットクロック数をスレーブに対して指定するために使用されます。また、処理中のオーディオフレームでマスタからの想定されるフレーム同期信号または遅延したフレーム同期信号の発生を検出するためにも使用されます。この場合、エラーが生成されます。詳細については、[セクション 33.3.12 : エラーフラグ](#)を参照してください。

スレーブモードでは、SAI\_xFRCR レジスタの FRL[7:0] ビット設定に制約はありません。

フレーム内のビット数は FRL[7:0] + 1 に等しくなります。

オーディオフレーム内で転送する最小ビット数は 8 です。

## フレーム同期極性

SAI\_xFRCR レジスタの FSPOL ビットにより、フレームが開始される FS ピンのアクティブ極性が設定されます。フレーム開始は、エッジに反応します。

スレーブモードにおいて、オーディオブロックは有効なフレームを待って、送信または受信を開始します。フレーム開始はこの信号に同期します。これは、通信中のフレーム開始が検出されず、予測のフレーム開始と一致しない場合にのみ有効です ([セクション 33.3.12 : エラーフラグ](#)を参照)。

マスタモードでは、SAI\_xCR1 レジスタの SAIXEN ビットがクリアされるまで、オーディオフレームが完了するごとにフレーム同期が毎回送信されます。前のオーディオフレームの終了時に FIFO にデータがない場合、アンダーラン状態は[セクション 33.3.12 : エラーフラグ](#)の説明のように扱われますが、オーディオ通信フローが中断されることはありません。

## フレーム同期アクティブレベル長

SAI\_xFRCR レジスタの FSALL[6:0] ビットにより、フレーム同期信号のアクティブレベル長が設定されます。この長さは、1 から 128 のビットクロックに設定することができます。

例えば、I2S、LSB 詰めまたは MSB 詰めモードでは、アクティブ長をフレーム長の 1/2 に、PCM/DSP、TDM モードでは 1 ビット幅に、それぞれ設定することができます。

## フレーム同期オフセット

アプリケーションで対象とするオーディオプロトコルに応じて、オーディオフレームの最終ビットまたは先頭ビットを送信したときフレーム同期信号をアサートすることができます (例えば、それぞれ I2S 規格プロトコルおよび MSB 詰めプロトコルの場合)。SAI\_xFRCR レジスタの FSOFF ビットにより、2 つの設定のいずれかを選択することができます。

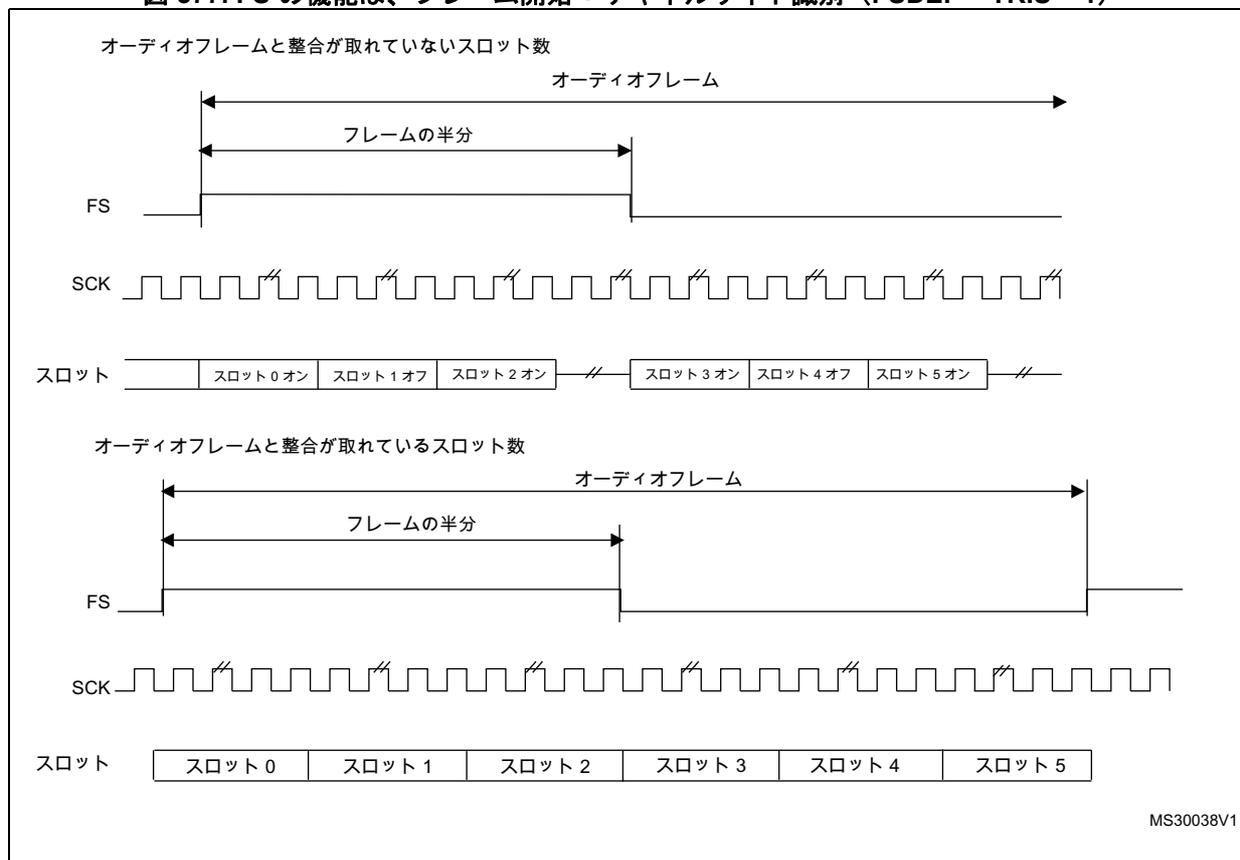
## FS 信号の機能

FS 信号は、FS の機能によって意味が異なります。SAI\_xFRCR レジスタの FSDEF ビットによりどちらの意味を持たせるかを選択します。

- 0: 例えば PCM/DSP、TDM、AC'97 オーディオプロトコルのようなフレーム開始
- 1: 例えば I2S、MSB 詰めまたは LSB 詰めプロトコルのような、オーディオフレーム内のチャンネルサイド識別およびフレーム開始

FS 信号をフレーム開始およびフレーム内チャンネルサイド識別と見なす場合、指定スロット数は半分を左チャンネル用、もう半分を右チャンネル用と見なす必要があります。ハーフオーディオフレームのビットクロック数がチャンネルサイド専用スロット数より大きい場合、TRIS = 0 であれば、SAI\_xCR2 レジスタの残りのビットクロックに対して 0 が送信されます。TRIS = 1 であれば、SD ラインはハイインピーダンス (HI-Z) 状態になります。受信モードでは、残りのビットクロックサイクルはチャンネルサイドが変化するまで無視されます。

図 377. FS の機能は、フレーム開始 + チャンネルサイド識別 (FSDEF = TRIS = 1)

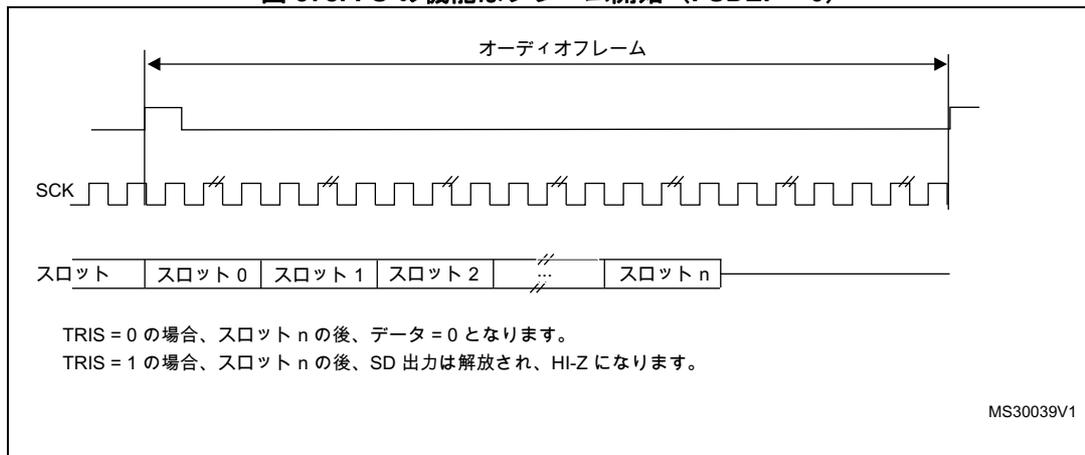


1. フレーム長は偶数である必要があります。

SAI\_xFRCR の FSDEF ビットがクリアされたままの場合、FS 信号はフレーム開始に相当し、さらに SAI\_xSLOTR の NBSLOT[3:0] ビットで指定されたスロット数に SAI\_xSLOTR の SLOTSZ[1:0] ビットで設定されたスロット当たりのビット数を乗算した値が、フレームサイズ (SAI\_xFRCR レジスタの FRL[7:0] ビット) より小さい場合、

- SAI\_xCR2 レジスタで TRIS = 0 のとき、送信の場合はフレームの終わりまで、最終スロット後の残りのビットは強制的に 0 に設定されます。
- TRIS = 1 のとき、これら残りのビットの転送中、ラインはハイインピーダンス (HI-Z) になります。受信モードでは、これらのビットは無視されます。

図 378. FS の機能はフレーム開始 (FSDEF = 0)



送信モードでオーディオブロックが SD ラインの SPDIF 出力を取得するように設定されている場合は、FS 信号は使用されません。対応する FS の I/O は開放され、他の目的のために空けられます。

### 33.3.6 スロットの設定

スロットは、オーディオフレームの基本要素です。オーディオフレーム内のスロット数は  $NBSLOT[3:0] + 1$  に等しくなります。

オーディオフレーム当たりの最大スロット数は 16 に固定されています。

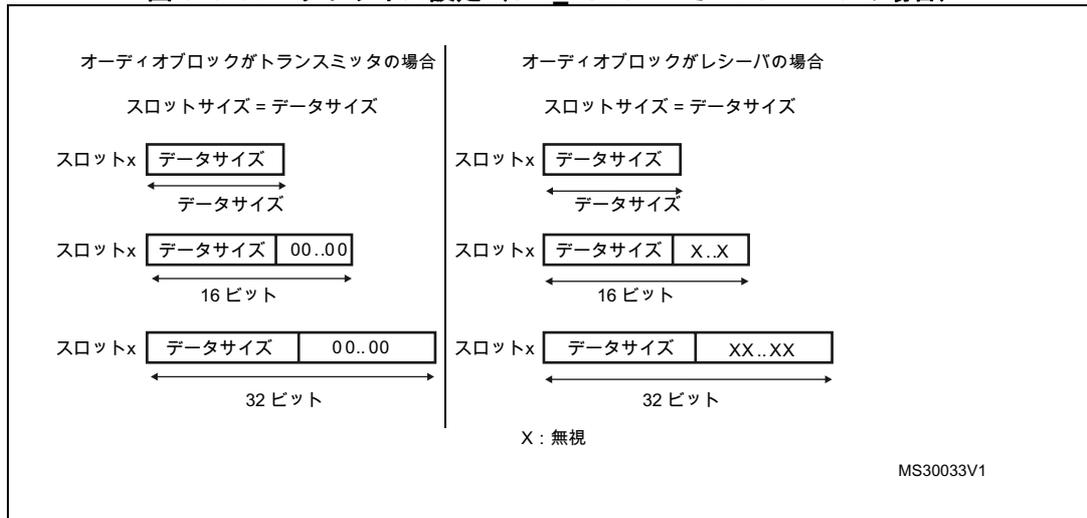
AC'97 プロトコル または SPDIF (PRTCFG[1:0] ビット = 10 または PRTCFG[1:0] ビット = 01 の場合) に対しては、スロット数は、そのプロトコル仕様を対象とするよう自動的に設定され、NBSLOT[3:0] の値は無視されます。

各スロットは、SAI\_xSLOTR レジスタの SLOTEN[15:0] ビットにより有効/無効を指定することができます。

送信モードでは、無効なスロットが転送されると、SD データラインは TRIS ビットの設定に応じて強制的に 0 に設定されるか、または開放されてハイインピーダンス (HI-Z) 状態になり開放されます (セクション: [非アクティブスロットでの出力データライン管理](#) を参照)。受信モードでは、このスロットの終わりからの受信した値は無視されます。結果として、FIFO アクセスがないため、この非アクティブなスロット状態にリンクした FIFO からの読み出しリクエストや FIFO への書き込みリクエストは起こりません。

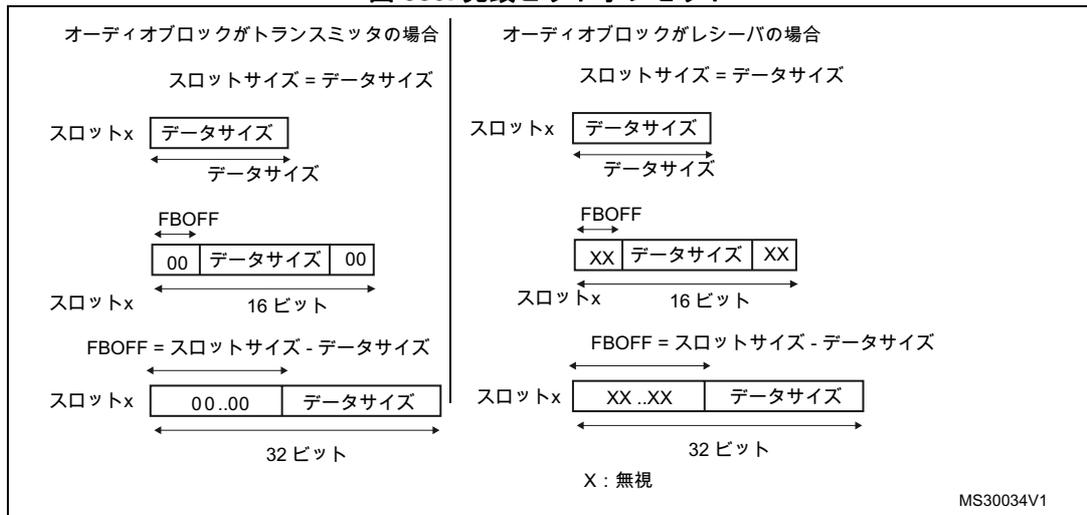
また、スロットのサイズも [図 379](#) に示すように設定可能です。スロットのサイズは、SAI\_xSLOTR レジスタの SLOTSZ[1:0] ビットを設定することで選択できます。サイズは、オーディオフレームの各スロットに同じように適用されます。

図 379. スロットサイズ設定 (SAI\_xSLOTR で FBOFF = 0 の場合)



スロット内で転送する先頭データビットの位置を選択することができます。このオフセットは、SAI\_xSLOTR レジスタの FBOFF[4:0] ビットによって設定されます。送信モードでは、スロットの開始からこのオフセット位置に到達するまでの間 0 値が挿入されます。受信時、オフセットフェーズ内のビットは無視されます。この機能は LSB 詰めプロトコルを対象としています (オフセットが、スロットサイズ - データサイズに等しい場合)。

図 380. 先頭ビットオフセット



SAI の誤動作を避けるためには、次の条件を満足する必要があります。

- FBOFF ≤ (SLOTSZ - DS)、
- DS ≤ SLOTSZ、
- NBSLOT x SLOTSZ ≤ FRL (フレーム長)、

SAI\_xFRCR レジスタの FSDEF ビットがセットされている場合でも、スロット数は偶数でなければなりません。

AC'97 および SPDIF プロトコル (PRTCFCFG[1:0] ビット = 10 または PRTCFCFG[1:0] ビット = 01) では、スロットサイズは、[セクション 33.3.9: AC'97 リンクコントローラ](#)で示されるように自動的に設定されます。

AC'97 モードでのクロックジェネレータのプログラミングに関する詳細は [セクション 33.3.9 : AC'97 リンクコントローラ](#) を、SPDIF モードでのクロックジェネレータのプログラミングに関する詳細は [セクション 33.3.10 : SPDIF 出力](#) を参照してください。

## 33.3.7 SAI クロックジェネレータ

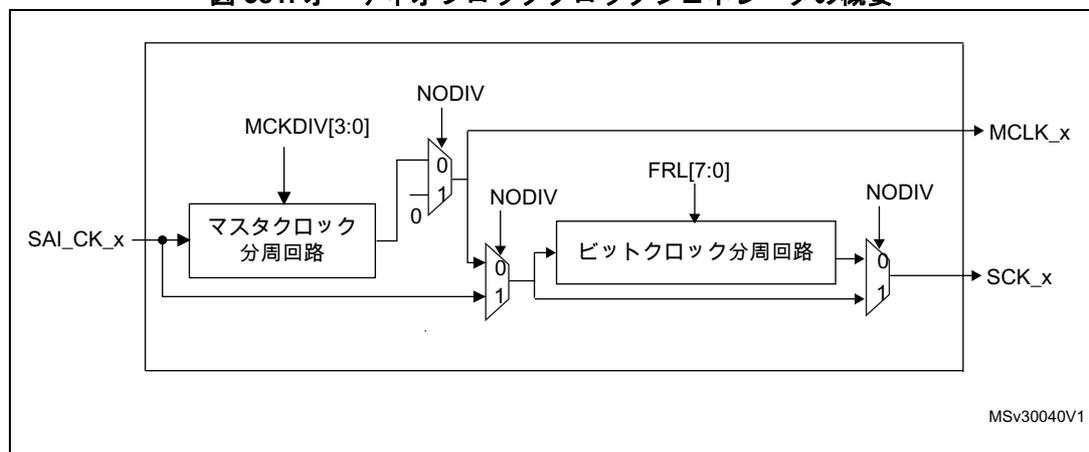
各オーディオブロックは、固有のクロックジェネレータを搭載しており、これら 2 つのブロックを完全に独立させています。これら 2 つのクロックジェネレータの間には機能的な違いはありません。

オーディオブロックをマスタに指定すると、クロックジェネレータは通信クロック (ビットクロック) と外部デコーダ用のマスタクロックを供給します。

オーディオブロックをスレーブに指定すると、クロックジェネレータは OFF になります。

[図 381](#) に、オーディオブロッククロックジェネレータのアーキテクチャを示します。

**図 381. オーディオブロッククロックジェネレータの概要**



**注 :** **NODIV が 1 に設定された場合で、このピンが GPIO ペリフェラル内で SAI ピンに設定されている場合、MCLK\_x 信号は 0 レベルに設定されます。**

クロックジェネレータのクロックソースは、製品のクロックコントローラから来ます。SAI\_CK\_x クロックはマスタクロックに相当し、このクロックは MCKDIV[3:0] ビットを使って次のように外部デコーダ用に分周されます。

$$MCLK_x = SAI\_CK\_x / (MCKDIV[3:0] * 2) \quad (MCKDIV[3:0] \text{ が } 0000 \text{ に等しくない場合})$$

$$MCLK_x = SAI\_CK_x \quad (MCKDIV[3:0] = 0000 \text{ の場合})$$

MCLK\_x 信号は TDMでのみ使用されます。

MCLK 出力と SCK\_x クロックのデューティサイクルを 50% に維持するため、分周比は偶数である必要があります。MCKDIV[3:0] ビット = 0000 の場合、MCLK\_x = SAI\_CK\_x を実現するため分周比 1 が使用されます。

SAI では、単純比 MCLK/FS = 256 とみなされます。多くの場合、[表 181](#)に示すように 3 つの周波数範囲を取り扱うことになります。

表 181. オーディオサンプリング周波数範囲の例

入力 SAI_CK_x クロック周波数	最も一般的に使用されるオーディオサンプリング周波数	MCKDIV[3:0]
192 kHz * 256	192 kHz	MCKDIV[3:0] = 0000
	96 kHz	MCKDIV[3:0] = 0001
	48 kHz	MCKDIV[3:0] = 0010
	16 kHz	MCKDIV[3:0] = 0110
	8 kHz	MCKDIV[3:0] = 1100
44.1 kHz * 256	44.1 kHz	MCKDIV[3:0] = 0000
	22.05 kHz	MCKDIV[3:0] = 0001
	11.025 kHz	MCKDIV[3:0] = 0010
SAI_CK_x = MCLK <sup>(1)</sup>	MCLK	MCKDIV[3:0] = 0000

1. これは、製品クロックコントローラが PLL クロックの代わりに外部 クロックソースを選択した場合に発生することがあります。

SAI\_xCR1 レジスタの NODIV ビット を 0 に設定して対応するオーディオブロックをマスタとして指定した場合、マスタクロックを外部デコーダへ接続した端子に生成することができます。スレーブでは、クロックジェネレータが OFF で、かつ MCLK\_x I/O ピンが汎用 I/O として開放されているため、この最終ビットに設定された値は無視されます。

ビットクロックはマスタクロックから生成されます。ビットクロック分周器は、ビットクロック (SCK\_x) とマスタクロック (MCLK\_x) との間の分周比を次式により設定します。

$$SCK_x = MCLK \times (FRL[7:0] + 1) / 256$$

ここで、

256 は、MCLK とオーディオサンプリング周波数間の固定比率です。

FRL[7:0] はオーディオフィーム内のビットクロックサイクル数 - 1 であり、SAI\_xFRCR レジスタで設定されます。

マスタモードでは、ビットクロックサイクル当たりの MCLK\_x パルス (偶数個の整数) が得られるよう、(FRL[7:0] + 1) が 2 の累乗値 ( [セクション 33.3.5: フレーム同期](#) を参照) と等しくなる必要があります。50 % のデューティサイクルがビットクロック (SCK\_x) で保証されます。

SAI\_CK\_x クロックを、ビットクロック周波数に等しくすることもできます。この場合、SAI\_xCR1 レジスタの NODIV ビットがセットされることになり、MCKDIV 分周器内の値とビットクロック分周器は無視されます。この場合、フレーム当たりのビット数は 2 の累乗である必要はなく自由に設定可能です。

SCK のビットクロックストロブエッジは、SAI\_xCR1 レジスタの CKSTR ビットで設定することができます。

SPDIF モードでのクロックジェネレータのプログラミングの詳細は、 [セクション 33.3.10: SPDIF 出力](#) を参照してください。

## 33.3.8 内部 FIFO

SAI 内の各オーディオブロックには固有の FIFO を搭載しています。そのブロックがトランスミッタに指定されるかレシーバに指定されるかによって、FIFO にそれぞれ書き込んだり読み出したりすることができます。したがって、SAI\_xSR レジスタの FREQ ビットに対応した FIFO リクエストは 1 つしかありません。

SAI\_xIM レジスタの FREQIE ビットが有効化されている場合、割り込みが生成されます。これは次の条件によって変わります：

- FIFO 閾値設定 (SAI\_xCR2 の FLTH ビット)
- 通信方向 (トランスミッタまたはレシーバ) [セクション：送信モードでの割り込み生成](#) および [セクション：受信モードでの割り込み生成](#) を参照してください。

### 送信モードでの割り込み生成

送信モードでの割り込みの生成は、次のように FIFO の設定によって変わります：

- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO エンプティ (FTH[2:0] = 000b) に設定されている場合、SAI\_xDR レジスタに有効なデータがない (SAI\_xSR の FLTH[2:0] ビット < 001b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO がエンプティ (空) でなくなった場合 (SAI\_xSR の FLTH[2:0] ビット ≠ 000b)、すなわち FIFO に 1 個以上のデータが格納されている状態になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/4 (FTH[2:0] = 001b) に設定されている場合、データが FIFO の 1/4 より少ない (SAI\_xSR の FLTH[2:0] ビット < 010b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO の 1/4 以上にデータがある状態 (SAI\_xSR の FLTH[2:0] ビット ≥ 010b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/2 (FTH[2:0] = 010b) に設定されている場合、データが FIFO の 1/2 より少ない (SAI\_xSR の FLTH[2:0] ビット < 011b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO の 1/2 以上にデータがある状態 (SAI\_xSR の FLTH[2:0] ビット ≥ 011b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 3/4 (FTH[2:0] = 011b) に設定されている場合、データが FIFO の 3/4 より少ない (SAI\_xSR の FLTH[2:0] ビット < 100b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO の 3/4 以上にデータがある状態 (SAI\_xSR の FLTH[2:0] ビット ≥ 100b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフル (FTH[2:0] = 100b) に設定されている場合、FIFO がフルでない (SAI\_xSR の FLTH[2:0] ビット < 101b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO データがフル (SAI\_xSR の FLTH[2:0] ビット = 101b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。

## 受信モードでの割り込み生成

受信モードでの割り込みの発生は、次のように FIFO の設定によって変わります：

- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO エンプティ (FTH[2:0] = 000b) に設定されている場合、SAI\_xDR レジスタに 1 個以上のデータがある (SAI\_xSR の FLTH[2:0] ビット  $\geq$  001b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO がエンプティになる (SAI\_xSR の FLTH[2:0] ビット = 000b)、すなわち FIFO にデータがなくなると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/4 (FTH[2:0] = 001b) に設定されている場合、FIFO データロケーションの 1/4 以上が使用可能 (SAI\_xSR の FLTH[2:0] ビット  $\geq$  010b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。使用可能な FIFO データロケーションが 1/4 未満 (SAI\_xSR の FLTH[2:0] ビット  $<$  010b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 1/2 (FTH[2:0] = 010b) に設定されている場合、FIFO データロケーションの 1/2 以上が使用可能 (SAI\_xSR の FLTH[2:0] ビット  $\geq$  011b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。使用可能な FIFO データロケーションが 1/2 未満 (SAI\_xSR の FLTH[2:0] ビット  $<$  011b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフルの 3/4 (FTH[2:0] = 011b) に設定されている場合、FIFO データロケーションの 3/4 以上が使用可能 (SAI\_xSR の FLTH[2:0] ビット  $\geq$  100b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。使用可能な FIFO データロケーションが 3/4 未満 (SAI\_xSR の FLTH[2:0] ビット  $<$  100b) になると、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。
- SAI\_xCR2 レジスタの FIFO 閾値ビットが FIFO がフル (FTH[2:0] = 100b) に設定されている場合、FIFO がフル (SAI\_xSR の FLTH[2:0] ビット = 101b) という条件で、割り込みが生成されます (SAI\_xSR レジスタの FREQ ビットがハードウェアによって 1 にセットされます)。FIFO データがフルでなくなると (SAI\_xSR の FLTH[2:0] ビット  $<$  101b)、この割り込み (SAI\_xSR レジスタの FREQ ビット) はハードウェアによってクリアされます。

SAI\_xCR1 レジスタの DMAEN ビットがセットされている場合、SAI は、割り込み生成と同様に DMA を使用することができます。FREQ ビットアサーションメカニズムは、上記 FREQIE について説明した割り込み生成メカニズムと同じです。

各 FIFO は 8 ワードの FIFO です。FIFO に対する書き込み操作や読み出し操作は、そのアクセスサイズによらず、1 回につき FIFO の 1 ワード分を対象とします。FIFO の 1 ワードには 1 オーディオスロットが格納されます。FIFO ポインタは、SAI\_xDR レジスタに対する各アクセスの後に 1 ワードずつインクリメントされます。

SAI\_xDR に書き込む際、データは右詰めである必要があります。

受信データは SAI\_xDR で右詰めとなっています。

SAI\_xCR2 レジスタの FFLUSH ビットをセットして SAI を無効化すると、FIFO ポインタを再初期化することができます。SAI が有効なときに FFLUSH をセットすると、FIFO 内にあるデータは自動的に失われます。

## 33.3.9 AC'97 リンクコントローラ

SAI は AC'97 リンクコントローラとしても機能することができます。このプロトコルの仕様は次のとおりです：

- スロット番号とスロットサイズは固定です。
- フレーム同期信号は完全に定義されており波形は固定です。

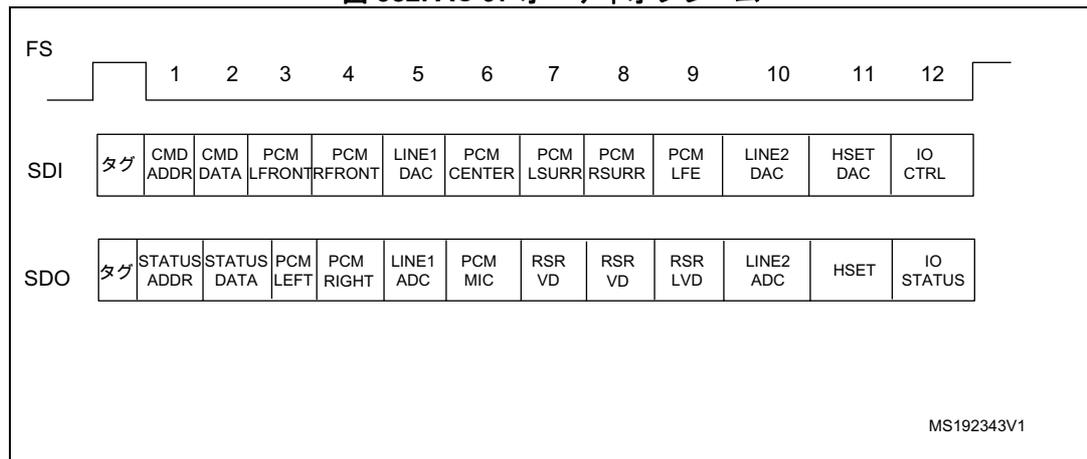
このプロトコルを選択するには、SAI\_xCR1 レジスタの PRTCFG[1:0] ビットを 10 に設定します。AC'97 モードを選択した場合、使用可能なデータサイズは 16 ビットまたは 20 ビットのみであり、その他の場合、SAI の動作は保証されません。

- したがって、NBSLOT[3:0] ビットと SLOTSZ[1:0] ビットは無視されます。
- スロット数は 13 スロットに固定されています。最初のスロットは 16 ビット幅で、他はすべて 20 ビット幅（データスロット）です。
- SAI\_xSLOTR レジスタの FBOFF[4:0] ビットは無視されます。
- SAI\_xFRCCR レジスタは無視されます。
- MCLK は使用されません。

非同期指定されたブロックからの FS 信号は、自動的に出力として設定されます。これは、マスタ/スレーブの設定にかかわらず AC'97 コントローラリンクが FS 信号を駆動するためです。

図 382 に、AC'97 オーディオフレーム構造を示します。

図 382. AC'97 オーディオフレーム



注： AC'97 プロトコルでは、タグのビット 2 は予約済みです（常に 0）。したがって、SAI FIFO に書き込まれる値によらず、タグのビット 2 は 0 レベルに固定されます。

タグ表示についての詳細は、AC'97 プロトコル 規格を参照してください。

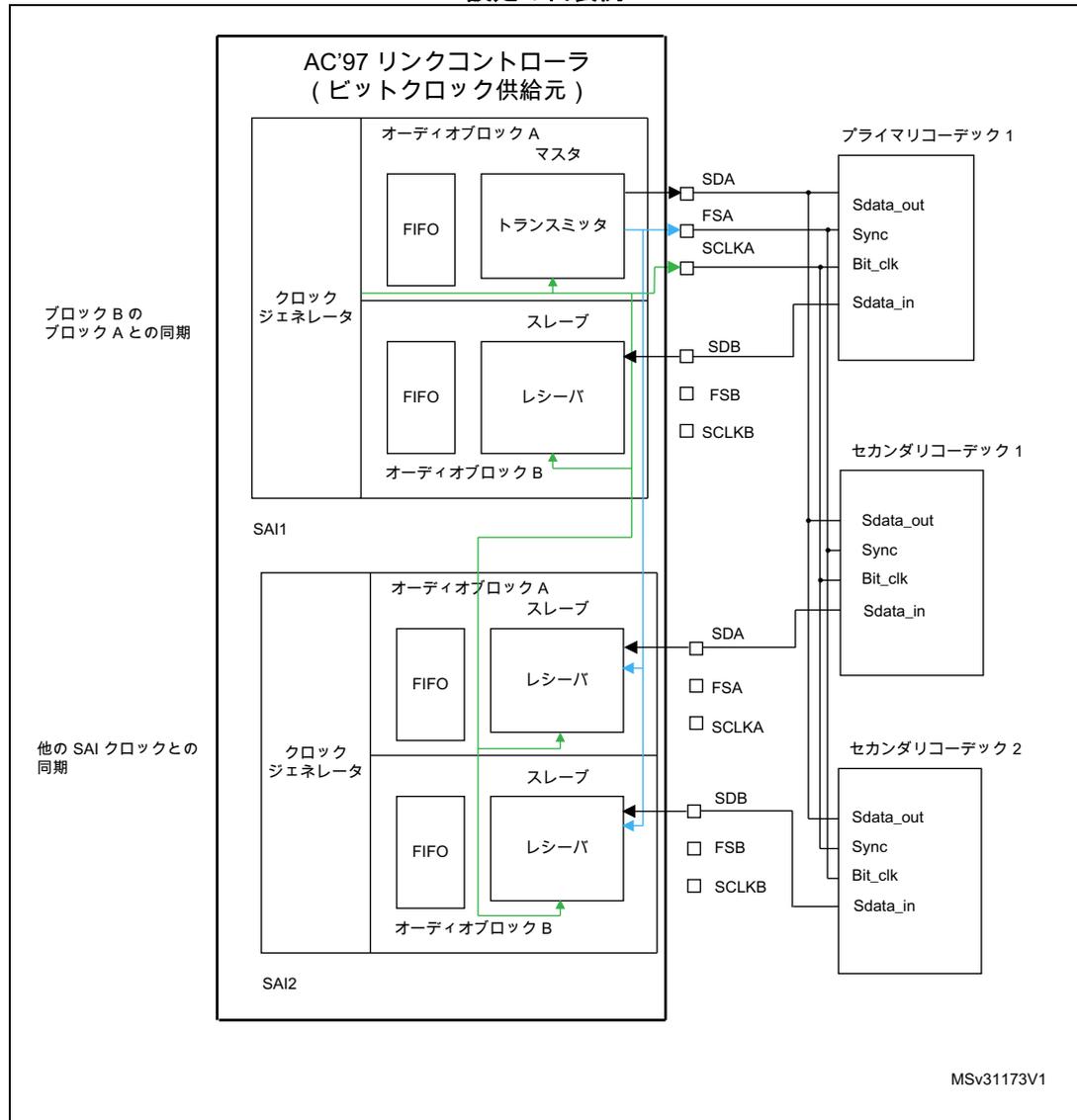
1 つの SAI を AC'97 の 2 点間通信に的を絞って使用することができます。

2 つの SAI を使用することで（デバイスに 2 つの内蔵 SAI を搭載）、図 383 に示す 3 個の外部 AC'97 デコーダを制御することができます。

SAI1 では、オーディオブロック A を非同期マスタトランスミッタに指定する必要がありますが、オーディオブロック B はスレーブレシーバに指定され、オーディオブロック A と内部的に同期します。

SAI2 では、オーディオブロック A と B がどちらもスレーブ受信モードで外部 SAI1 と同期するように設定されます。

図 383. 2 つ以上の内蔵SAI を搭載したデバイス (3 個の外部 AC'97 デコーダ) の AC'97 設定の代表例



受信モードでは、スロット 0 のコーデックレディビットがローにデコードされた場合、AC'97 リンクコントローラとして機能する SAI は、FIFO リクエストを必要としないため FIFO 内のデータストレージも不要です。SAI\_xIM レジスタの CNRDYIE ビットが有効化された場合、SAI\_xSR レジスタの CNRDY フラグがセットされ、割り込みが生成されます。このフラグは、AC'97 プロトコル専用です。

### AC'97 モードでのクロックジェネレータのプログラミング

AC'97 モードでは、フレーム長は 256 ビットに固定され、その周波数は 48 kHz に設定されます。[セクション 33.3.7 : SAI クロックジェネレータ](#) に示す式には、適切なフレームレート ( $F_{FS\_x}$ ) を求めるために、 $FRL = 255$  という条件が適用されます。

## 33.3.10 SPDIF 出力

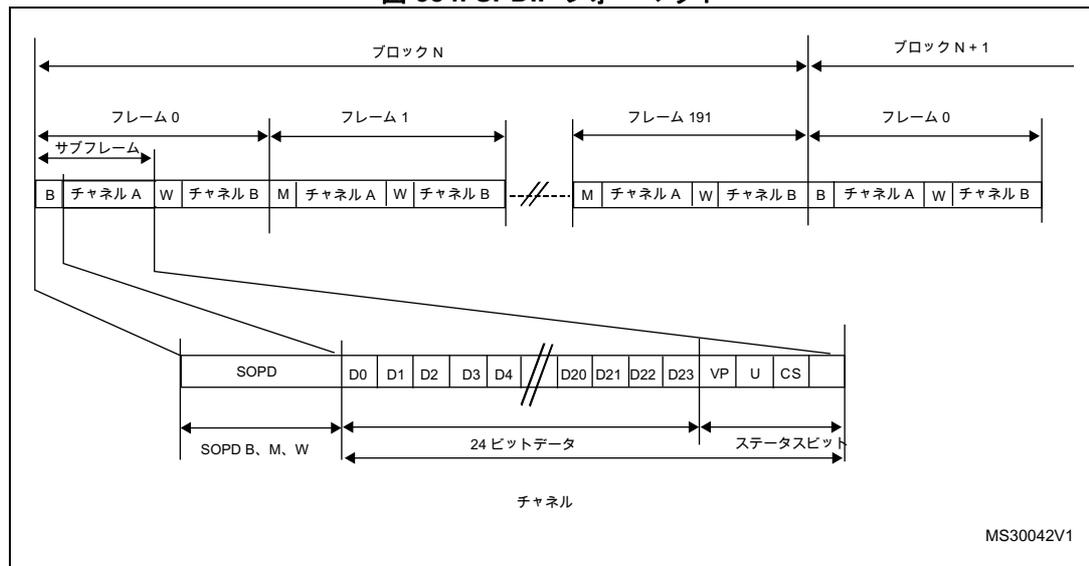
SPDIF インタフェースは送信モードでのみ使用できます。このインタフェースはオーディオ IEC60958 をサポートします。

SPDIF モードを選択するには、SAI\_xCR1 レジスタの PRTCFG[1:0] ビットに 01 をセットします。

SPDIF プロトコルを使用する場合

- SD データラインのみが有効です。
- I/O ピンの FS、SCK、MCLK は空けてあります。
- MODE[1]ビットはマスタモードで、SAI のクロックジェネレータを有効にしたりSD ラインでのデータレートを管理するためには、0を選択してください。
- データサイズは強制的に 24 ビットに設定されます。SAI\_xCR1 レジスタの DS[2:0] ビットにセットされた値は無視されます。
- クロックジェネレータのシンボルレートを、ビットクロックがシンボルレートの 2 倍でなければならないことを考慮して、定義する必要があります。データはマンチェスタープロトコルでコード化されます。
- SAI\_xFRCCR レジスタおよび SAI\_xSLOTR レジスタは無視されます。SAI は、SPDIF プロトコル要件 ( [図 384](#)を参照) に適合するように内部的に設定されます。

図 384. SPDIF フォーマット



SPDIF ブロックには 192 個のフレームが含まれています。各フレームは、2 個の 32 ビットサブフレーム（一般に左チャンネル用と右チャンネル用に 1 つずつ）で構成されています。各サブフレームは SOPD パターン（4 ビット）で構成されています。このパターンを使用して、サブフレームがブロックの先頭である（つまり、チャンネル A を識別している）、ブロックのどこかでチャンネル A を識別している、またはチャンネル B を参照しているのいずれかを指定します（表 182を参照）。チャンネル情報の次の 28 ビットは 24 ビットのデータと 4 ビットの状態情報から成ります。

表 182. SOPD パターン

SOPD	プリアンプルのコード化		説明
	最終ビットが 0	最終ビットが 1	
B	11101000	00010111	ブロックの始めにあるチャンネル A データ
W	11100100	00011011	ブロックのどこかにあるチャンネル B データ
M	11100010	00011101	チャンネル A データ

SAI\_xDR に格納されているデータは 次のような内容でなければなりません。

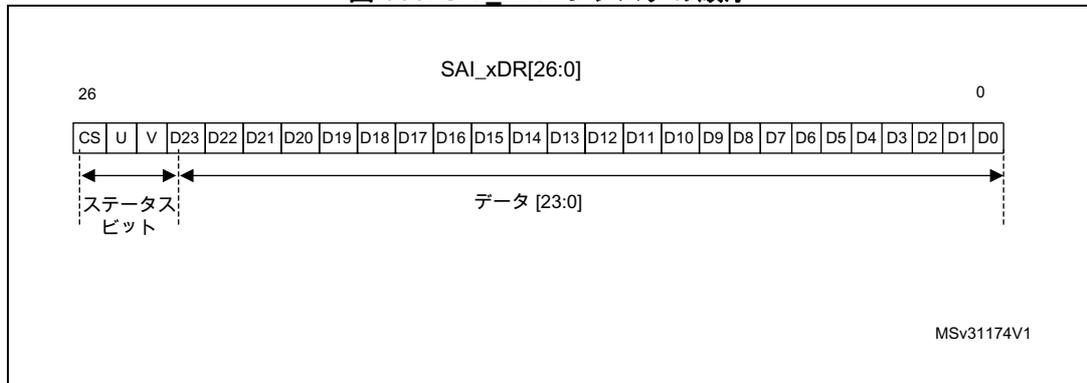
- SAI\_xDR[26:24] は、チャンネルのステータスビット、ユーザービット、有効性ビットを含みます。
- SAI\_xDR[23:0] は、検討されたチャンネルに関するデータ 24 ビットを含みます。

データサイズが 20 ビットの場合、データは SAI\_xDR[23:4] に配置されます。

データサイズが 16 ビットの場合、データは SAI\_xDR[23:8] に配置されます。

SAI\_xDR[23] は常に MSB を表します。

図 385. SAI\_xDR レジスタの順序



**注：** 必ず LSB (最下位ビット) から転送されます。

SAI はまず最初にブロック内の各サブフレームの適切なプリアンプル部を送信します。次に、SAI\_xDR が SD ラインに送信されます (マンチェスターコード化)。SAI は、表 183 の記載通りに計算されたパリティビットを転送することで、サブフレームを終了させます。

表 183. パリティビットの計算

SAI_xDR[26:0]	転送されるパリティビット P の値
奇数個の "0"	0
奇数個の "1"	1

SAI は送信モードでしか動作できないため、アンダーランは SPDIF モードの SAI\_xSR レジスタで有効な唯一のエラーフラグです。そのため、アンダーラン割り込みまたはアンダーランステータスビットによって検出されたアンダーランエラーから復帰するためには、次のシーケンスを実行する必要があります。

1. DMA が使用されている場合は、DMA ストリームを無効にします (DMA ペリフェラルを使用)。
2. SAI を無効にし、SAI\_xCR1 レジスタの SAIXEN ビットをポーリングすることでペリフェラルが物理的に無効化されていることを確認します。
3. SAI\_xCLRFR レジスタの COVRUNDR フラグをクリアします。
4. FIFO を一掃するために、SAI\_xCR2 の FFLUSH ビットをセットします。  
ソフトウェアで、新しいブロックの始めに今後格納するデータ (プリアンプル部B用のデータ) のアドレスを指し示す必要があります。DMA が使用されている場合、DMA ソーススペースアドレスのポインタは適宜更新してください。
5. DMA を使用して、新しいソーススペースアドレスに従ってデータ転送を管理する場合は、DMA ストリーム (DMA ペリフェラル) を再び有効にします。
6. SAI\_xCR1 レジスタの SAIXEN ビットをセットして、再び SAI を有効にします。

## SPDIF ジェネレータモードでのクロックジェネレータのプログラミング

SPDIF ジェネレータモードの場合、SAI はシンボルレートに等しいビットクロックを供給します。下表では、オーディオサンプリングレートに対するシンボルレートの一般例を示します。

表 184. オーディオサンプリング周波とシンボルレート (SHARK)

オーディオサンプリング周波 (F <sub>S</sub> )	シンボルレート
44.1 kHz	2.8224 MHz
48 kHz	3.072 MHz
96 kHz	6.144 MHz
192 kHz	12.288 MHz

より一般的には、オーディオサンプリングレート (F<sub>S</sub>) とビットクロックレート (F<sub>SCK\_X</sub>) の関係は、次の式で求められます。

$$F_S = \frac{F_{SAI\_CK\_X}}{64}$$

### 33.3.11 特有の機能

SAI インタフェースには、選択したオーディオプロトコルに応じて役立つ特定の機能が組み込まれています。これらの機能へは、SAI\_xCR2 レジスタの特定ビットを使ってアクセスすることができます。

#### ミュートモード

ミュートモードは、オーディオサブブロックがトランスミッタまたはレシーバであるとき使用することができます。

#### 送信モード時のオーディオサブブロック

送信モード時、ミュートモードはいつでも選択可能です。ミュートモードは、オーディオフレーム全体に対して有効です。処理中のフレーム内で SAI\_xCR2 レジスタの MUTE ビットがセットされると、ミュートモードが有効になります。

ミュートモードビットは、フレームの終わりでのみ認識されます。この時点でミュートモードビットがセットされた場合は、ミュートモードは新しいオーディオフレームの先頭から次に来るフレームの終了までフレーム全体に対して有効になります。その後、このビットを認識して、次のフレームもミュートフレームか否かを決定します。

SAI\_xSLOTR レジスタの NBSLOT[3:0] ビットに設定されたスロット数が 2 以下の場合、ミュートモード中に送信される値が 0 なのか、または各スロットの最後の値なのかを指定することができます。この選択は、SAI\_xCR2 レジスタの MUTEVAL ビットを使って行います。

SAI\_xSLOTR レジスタの NBSLOT[3:0] ビットに設定されたスロット数が 2 より大きい場合、各スロットの各ビットで 0 値が送信されるため、SAI\_xCR2 の MUTEVAL ビットは無意味となります。

ミュートモード時は、FIFO ポインタのインクリメントは続きます。つまり、FIFO にデータは供給され続けますが、ミュートモードの間は破棄されます。

### 受信モード時のオーディオサブブロック

受信モードでは、指定されかつ有効なオーディオフレームのすべてのスロットが、指定された連続した数のオーディオフレームに対して 0 を受信したとき、外部 トランスミッタから送信されたミュートモードを検出することが可能です (SAI\_xCR2 レジスタの MUTE CNT[5:0] ビット)。

ミュートフレーム数が検出されると、SAI\_xSR レジスタの MUTEDET フラグがセットされて、SAI\_xCR2 の MUTEDETIE ビットがセットされている場合は割り込みを生成することができます。

オーディオサブブロックが無効化されたとき、または有効なスロットがオーディオフレーム内で 1 つ以上のデータを受信したとき、ミュートフレームカウンタがクリアされます。カウンタが MUTE CNT[5:0] ビットに指定された値に到達すると、割り込みが 1 回だけ生成されます。次に、カウンタがクリアされると割り込みイベントが再初期化されます。

**注：** *ミュートモードは SPDIF オーディオブロックには使用できません。*

### モノラル/ステレオモード

送信モードでは、スロット数が 2 と想定し (SAI\_xSLOTR で NBSLOT[3:0] = 0001)、メモリ内でデータを前処理することなくモノラルモードを指定できます。このような場合、スロット 0 のデータがデータスロット 1 へ複製されるため、FIFO へのアクセスタイムは半減されます。

モノラルモードを有効にするには、

1. SAI\_xCR1 レジスタの MONO ビットに 1 をセットします。
2. SAI\_xSLOTR レジスタの NBSLOT ビットに 1、SLOTEN ビットに 3 をセットします。

受信モードでは、MONO ビットをセットすることができますが、送信モードの場合と同様にスロット数が 2 のときだけ意味があります。このビットをセットした場合、スロット 0 のデータだけが FIFO に格納されます。この場合、前のスロットと同じと見なされるため、スロット 1 に属するデータは無視されます。受信時のデータの流が左右でデータが異なる真のステレオオーディオフローである場合、MONO ビットは意味を成しません。出力ステレオファイルから等価なモノラルファイルへの変換は、ソフトウェアで行われます。

### コンパニングモード

通信アプリケーションでは、データコンパニングアルゴリズムにより送信データまたは受信データの処理が必要となることがあります。

[図 386](#) に示すように、SAI\_xCR2 レジスタの COMP[1:0] ビットに応じて (TDM モードを選択した場合のみ使用)、SD シリアル出力ラインに送信する前に処理 (圧縮) するか否か、あるいは SD シリアル入力ラインから受信した後にデータを処理 (伸長) するか否かをソフトウェアで選択することができます。サポートされている 2 つのコンパニングモードは  $\mu$ -Law と A-Law (対数) で、これらは CCITT G.711 勧告の一部となっています。



米国と日本で採用されているコンパANDING規格は  $\mu$ -Law であり、14 ビットのダイナミックレンジが確保できます (SAI\_xCR2 レジスタで COMP[1:0] = 10)。

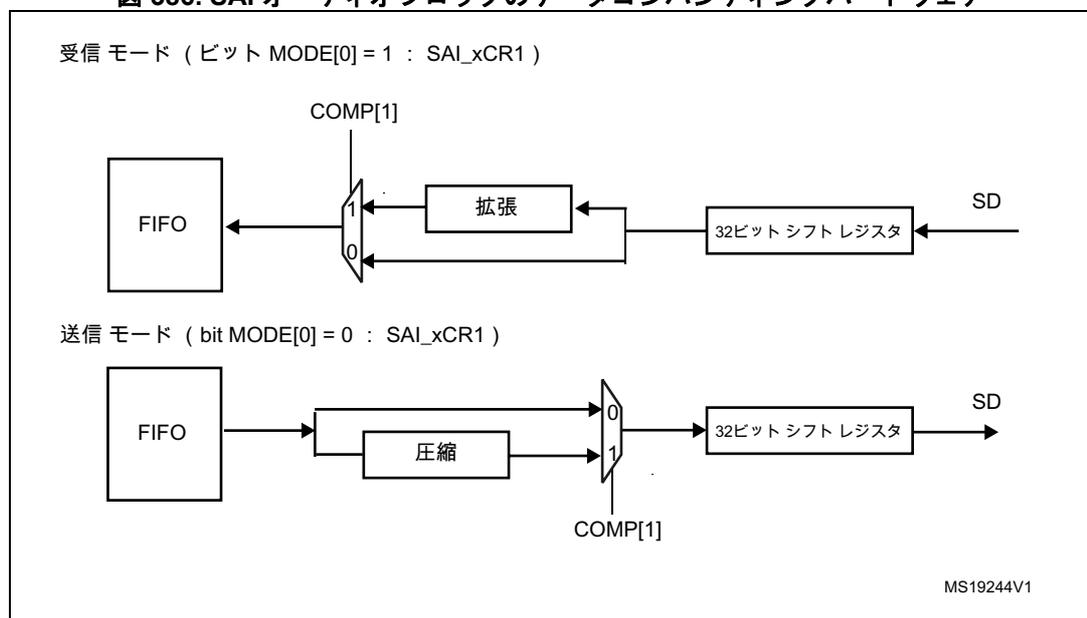
ヨーロッパのコンパANDING規格は A-Law であり、13 ビットのダイナミックレンジが確保できます (SAI\_xCR2 レジスタで COMP[1:0] = 11)。

コンパANDING規格 ( $\mu$ -Law または A-Law) は、SAI\_xCR2 レジスタの CPL ビットの設定に応じて、1 の補数表現または 2 の補数表現に基づいて計算することができます。

$\mu$ -Law および A-Law 規格では、データを MSB 詰め の 8 ビットコードにコード化します。コンパANDされたデータは常に 8 ビット幅です。このため、SAI オーディオブロックが有効化されていて (SAI\_xCR1 レジスタで SAIXEN ビット = 1)、さらに、COMP[1:0] ビットによりこれら 2 つのコンパANDINGモードのいずれかを選択している場合、SAI\_xCR1 レジスタの DS[2:0] ビットは 010 に固定されます。

コンパANDINGが不要な場合は、COMP[1:0] ビットをクリアしたままにします。

図 386. SAI オーディオブロックのデータコンパANDINGハードウェア



1. AC'97 を選択した場合は適用されません。

伸張モードまたは圧縮モードは SAI\_xCR2 により自動的に選択されます。

- SAI オーディオブロックがトランスミッタに設定され、かつ SAI\_xCR2 レジスタの COMP[1] ビットがセットされている場合、圧縮モードが適用されます。
- SAI オーディオブロックがレシーバに指定された場合、伸長アルゴリズムが適用されます。

### 非アクティブスロットでの出力データライン管理

送信モードでは、非アクティブスロットがデータラインに送信されたときに出力となっている SD ラインの動作を選択することができます (TRIS ビットを使用)。

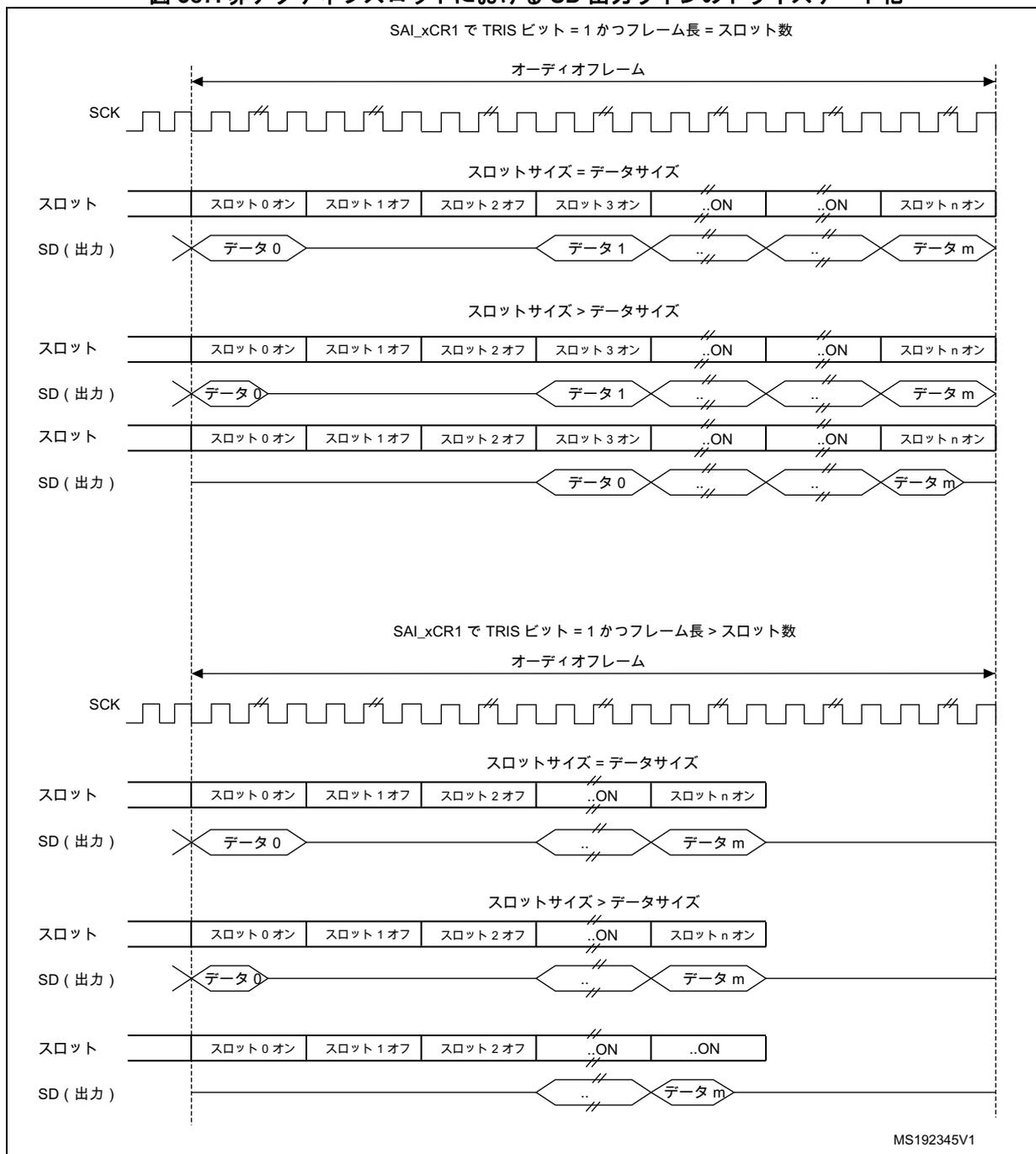
- 非アクティブスロットが送信される時、SAI が SD 出力ラインを 0 に固定することもできれば、
- 送信データの最終ビットの終わりでラインを HI-Z 状態にして、このノードに接続された他のトランスミッタへラインを開放することもできます。

2つのトランスミッタが同じSD出力ピンを同時に駆動しないように注意することが重要です。同時駆動した場合、短絡の原因となります。送信間の間隔を確保するため、データが32ビットより短い場合は、SAI\_xSLOTRレジスタのSLOTSZ[1:0]ビット=10に設定することにより、データを32ビットに拡張することができます。すると、次のスロットが非アクティブに指定された場合、アクティブスロットのLSBの終わりで（データを32ビットに拡張するため0を詰めるフェーズで）SD出力ピンがトライステートになります。

さらに、スロット数とスロットサイズの積がフレーム長より小さい場合、0を詰めてオーディオフレームを完成させるときにSD出力ラインがトライステートになります。

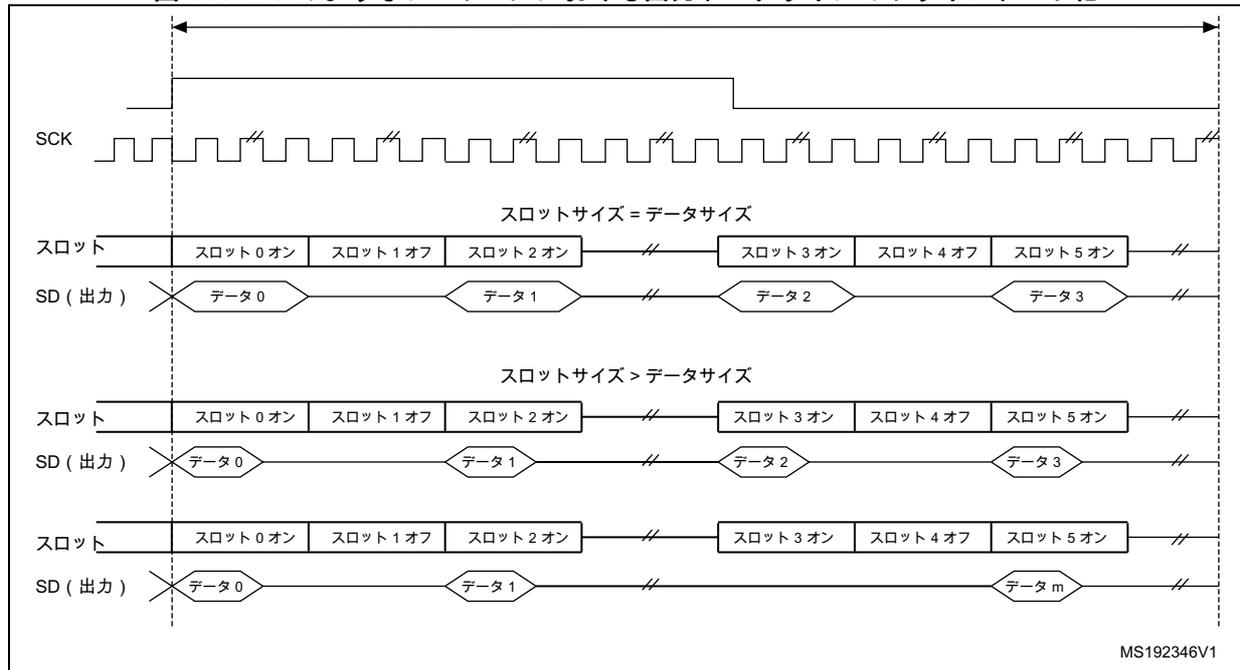
このような動作を  387 に示します。

図 387. 非アクティブスロットにおける SD 出力ラインのトライステート化



選択したオーディオプロトコルが FS 信号をフレーム開始とチャネルサイド識別として使用する場合 (SAI\_xFRCR レジスタでビット FSDEF = 1)、[図 388](#) に従ってトライステートモードが管理されます (この図では、SAI\_xCR1 レジスタでビット TRIS = 1、FSDEF = 1、ハーフフレーム長 > スロット数 / 2、NBSLOT = 6)。

図 388. I2S のようなプロトコルにおける出力データラインのトライステート化



SAI\_xCR2 レジスタの TRIS ビットがクリアされている場合、図 387 と 図 388 に示す SD 出力ライン上のすべてのハイインピーダンス状態は 0 値での駆動として置き換えます。

## 33.3.12 エラーフラグ

SAI は次のエラーフラグを実装しています。

- FIFO オーバーラン/アンダーラン
- 予測フレーム同期検出
- レイトフレーム同期検出
- コーデックノットレディ (AC'97 専用)
- マスタモードにおける不正クロック設定

### FIFO オーバーラン/アンダーラン (OVRUDR)

FIFO オーバーラン/アンダーランビットは、SAI\_xSR レジスタで OVRUDR と呼ばれます。

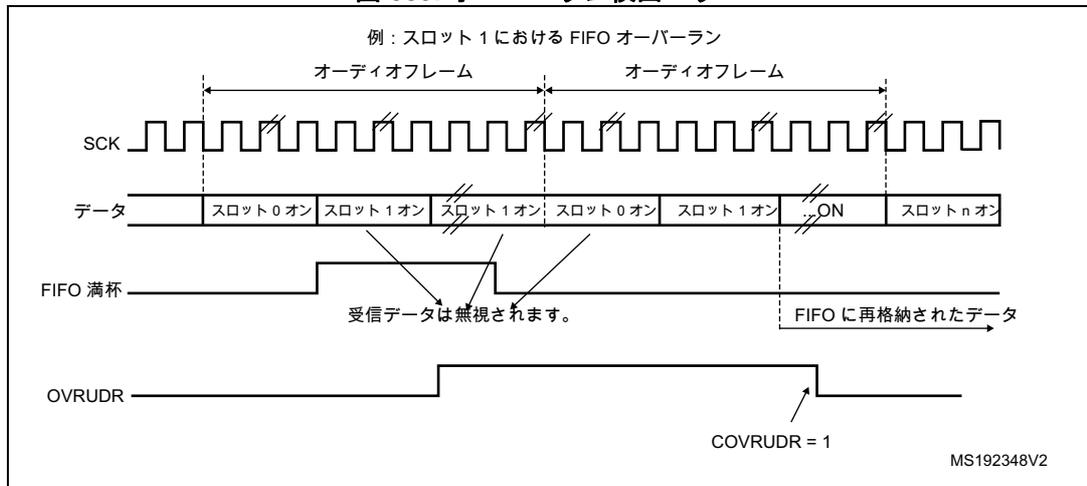
オーディオブロックはレシーバまたはトランスミッタのいずれかであり、SAI 内の各オーディオブロックに固有の SAI\_xSR レジスタを搭載しているため、オーバーランエラーとアンダーランエラーは、同じビットを使用しています。

#### オーバーラン

オーディオブロックがレシーバに設定されていて、FIFO がフルで受信データを保存できないときにオーディオフレームでデータが受信された場合に、オーバーラン状態が発生します。この場合、受信データが失われ、SAI\_xSR レジスタの OVRUDR フラグがセットされ、SAI\_xIM レジスタの OVRUDRIE ビットがセットされている場合は割り込みが生成されます。オーバーランが発生したスロット番号が、内部に保存されます。FIFO が新しいデータを格納できるようになるまで、データは FIFO に保存されません。FIFO に 1 データ分以上の空きがある場合、SAI オーディオブロックレシーバは、オーバーラン状態が検出されたとき内部に保存されたスロット番号からの新しいデータ (新しいオーディオフレームの) を保存します。これにより、転送先メモリでのデータスロットの配置の狂いを回避します (図 389 を参照)。

SAI\_xCLRFR レジスタの COVRUDR ビットがセットされると、OVRUDR フラグがクリアされます。

図 389. オーバーラン検出エラー



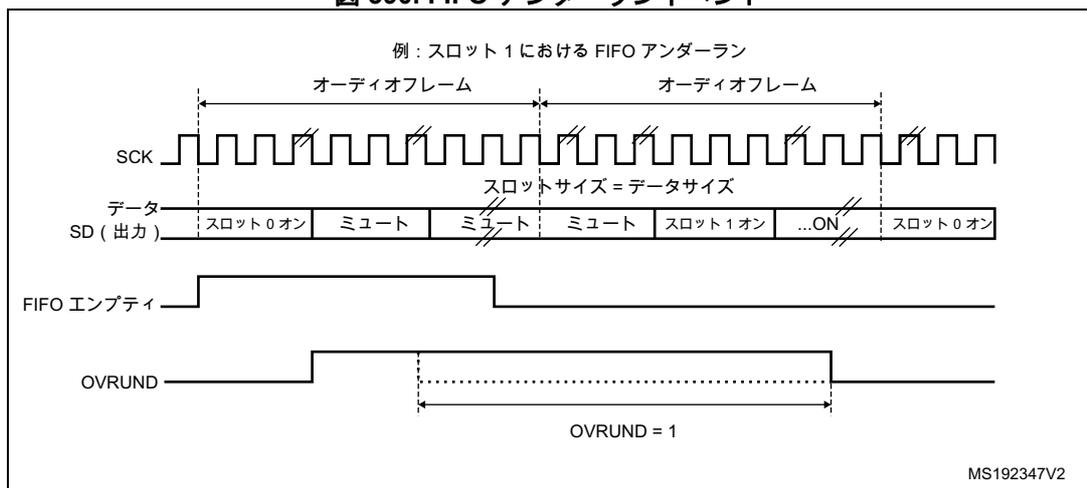
## アンダーラン

SAI 内のオーディオブロックがトランスミッタで、かつデータの送信が必要なとき FIFO がエンプティ（空）の場合に、アンダーランが発生します。アンダーランが検出されると、イベントが発生したスロット番号が保存され、FIFO がアンダーランが検出されたスロットに対応するデータを送信する準備ができるまで MUTE 値 (00) が送信されます (図 390 を参照)。これにより、メモリポインタとオーディオフレーム内のスロットの同期はずれを回避します。

アンダーランイベントにより SAI\_xSR レジスタの OVRUDR フラグがセットされ、SAI\_xIM レジスタの OVRUDRIE ビットがセットされている場合、割り込みが生成されます。このフラグをクリアするには、SAI\_xCLRFR レジスタの COVRUDR ビットをセットします。

アンダーランイベントは、オーディオサブブロックがマスタまたはスレーブとして設定されている場合に発生します。

図 390. FIFO アンダーランイベント



## 予測フレーム同期検出 (AFSDET)

AFSDET フラグはスレーブモードでのみ使用されます。マスタモードでは決してアサートされません。フレーム長、フレーム極性、フレームオフセットが設定されていて既知であるため、フレーム同期 (FS) が予測より早く検出されることを示します。

予測フレームの検出により、SAI\_xSR レジスタの AFSDET フラグがセットされます。

この検出は、現在のオーディオフレームに影響を与えません。現在のオーディオフレームは予測 FS に影響されません。つまり、現在のオーディオフレームを乱すことなく、FS 信号に関する「寄生」イベントフラグが立つということです。

SAI\_xIM レジスタの AFSDETIE ビットがセットされている場合は、割り込みが生成されます。AFSDET フラグをクリアするには、SAI\_xCLRFR レジスタの CAFSDET ビットをセットする必要があります。

予測フレーム検出エラーの後にマスタと再同期させるには、次の 4 ステップが必要です。

1. SAI\_xCR1 レジスタの SAIXEN ビットをリセットして、SAI ブロックを無効化します。SAI が無効であることを確認するには、SAIXEN ビットを読み返し、それが 0 にセットされていることを確認します。
2. SAI\_xCR2 レジスタの FFLUS ビットを使って FIFO を一掃します。
3. SAI ペリフェラルを再度有効化します (SAIXEN ビットを 1 に設定)。
4. SAI ブロックは FS の開始を待ってマスタとの同期を再開します。

**注：** SAIXEN フラグは AC'97 モードではアサートされません。これは SAI オーディオブロックがリンクコントローラとして機能し、スレーブに指定された場合でも FS 信号を生成するためです。FS 信号は使用されないため、このフラグは SPDIF モードでは意味を持ちません。

## レイトフレーム同期検出

SAI オーディオブロックがスレーブとして動作する場合のみ、SAI\_xSR レジスタの LFSDET フラグをセットすることができます。フレーム長、フレーム極性、フレームオフセット設定は、SAI\_xFRCR レジスタから知ることができます。

外部マスタが FS 信号を予想タイミングで送信しない (信号の発生が遅すぎる) 場合、LFSDET フラグがセットされ、SAI\_xIM レジスタの LFSDETIE ビットがセットされている場合は割り込みが生成されます。

SAI\_xCLRFR レジスタの CLFSDET ビットがセットされると、LFSDET フラグがクリアされます。

レイトフレーム同期検出フラグは、対応するエラーが検出されるとセットされます。SAI はマスタと再同期させる必要があります (シーケンスは、[セクション：予測フレーム同期検出 \(AFSDET\)](#) を参照)。

ノイズの多い環境では、オーディオブロックステートマシンが SCK クロック上のグリッチを間違っ  
て検出し、SAI データが間違っ  
たフレーム位置にシフトされることがあります。このようなイベントは SAI によって検出され、レイトフレーム同期検出エラーとして報告することができます。

外部マスタが連続モードでオーディオデータフレーム転送を管理していない場合は破壊は起こりませんが、ほとんどの場合、アプリケーションでこのような使われ方はされません。この場合、LFSDET フラグがセットされます。

**注：** LFSDET フラグは AC'97 モードではアサートされません。これは SAI オーディオブロックがリンクコントローラとして機能し、スレーブと指定された場合でも FS 信号を生成するためです。FS 信号はプロトコルで使用されないため、このフラグは SPDIF モードでは意味を持ちません。



## コーデックノットレディ (CNRDY AC'97)

SAI オーディオブロックが AC'97 モードで動作するように設定されている場合にのみ SAI\_xSR レジスタの CNRDY フラグが関係します (SAI\_xCR1 レジスタの PRTCFG[1:0] ビット = 10)。SAI\_xIM レジスタの CNRDYIE ビットがセットされている場合、CNRDY フラグがセットされたときに割り込みが生成されます。

AC'97 オーディオフレームのタグ 0 (スロット 0) の受信中にコーデックが通信の準備ができていないとき、CNRDY フラグがアサートされます。この場合、コーデックが動作可能であることをタグ 0 が示すまではコーデックは動作可能状態にはないため、FIFO に自動的に保存されるデータはありません。コーデックが動作可能状態になると、SAI\_xSLOTR レジスタで指定されたすべてのアクティブスロットがキャプチャされます。

CNRDY フラグをクリアするには、SAI\_xCLRFR レジスタの CCNRDY ビットをセットする必要があります。

## マスタモードでの不正なクロック設定 (NODIV = 0 の場合)

オーディオブロックがマスタとして動作し (MODE[1] = 0)、NODIV ビットが 0 に等しい場合、次の条件が満たされて SAI が有効になるとすぐに、WCKCFG フラグがセットされます。

- (FRL + 1) は 2 の累乗値ではない、かつ
- (FRL + 1) は 8~256 の範囲外である。

MODE、NODIV、SAIXEN ビットは SAI\_xCR1 レジスタに、FRL ビットは SAI\_xFRCR レジスタに属します。

WCKCFGIE ビットがセットされている場合、SAI\_xSR レジスタで WCKCFG フラグがセットされると割り込みが生成されます。このフラグをクリアするには、SAI\_xCLRFR レジスタの CWCKCFG ビットをセットします。

WCKCFG ビットがセットされると、オーディオブロックが自動的に無効化され、ハードウェアによって SAIXEN ビットがクリアされます。

### 33.3.13 SAI の無効化

SAI\_xCR1 レジスタの SAIXEN ビットをクリアすることによって、SAI 内のオーディオブロックをいつでも無効化することができます。すでに開始しているすべてのフレームは、SAI が動作を停止する前に自動的に完了します。現在のオーディオフレーム転送の最後に SAI が完全にオフするまで、SAIXEN ビットはハイを保持します。

他の SAI と同期している SAI 内にオーディオブロックがある場合、マスタになっている SAI を先に無効化する必要があります。

## 33.3.14 SAI DMA インタフェース

CPU を開放し、バスの帯域幅を最適化するため、各 SAI オーディオブロックには SAI\_xDR レジスタ (内部 FIFO へのアクセス用) に対して読み書きを行うための独立した DMA インタフェースを搭載しています。オーディオサブブロック毎に 1 つの DMA チャンネルがあり、基本的な DMA リクエスト/確認応答プロトコルに対応しています。

DMA 転送を行うようにオーディオブロックを設定するには、SAI\_xCR1 レジスタの DMAEN ビットをセットします。DMA リクエストは、FIFO 閾値レベルに応じて FIFO コントローラによって直接管理されます (詳細は、[セクション 33.3.8 : 内部 FIFO](#) を参照してください)。DMA 転送方向は、SAI オーディオサブブロック設定と次の様に関連付けられています。

- オーディオブロックがトランスミッタとして動作する場合、オーディオブロックの FIFO コントローラが DMA リクエストを出力して SAI\_xDR レジスタに書き込まれたデータを FIFO にロードします。
- オーディオブロックがレシーバとして動作する場合、DMA リクエストは SAI\_xDR レジスタからの読み出し動作に関わります。

SAI インタフェースを DMA モードに設定するには、次のシーケンスに従ってください。

1. SAI と FIFO 閾値レベルを設定して、DMA リクエストを開始するタイミングを指定します。
2. SAI DMA チャンネルを設定します。
3. DMA を有効にします。
4. SAI インタフェースを有効にします。

**注：** SAI ブロックを設定する前に、SAI DMA チャンネルを無効化する必要があります。

## 33.4 SAI 割り込み

SAI は [表 185](#) に示す 7 つの割り込みソースに対応しています。

表 185. SAI 割り込みソース

割り込みソース	割り込みグループ	オーディオブロックモード	割り込みイネーブル	割り込みクリア
FREQ	周波数	マスタ/スレーブ 受信または送信モード	SAI_xIM レジスタの FREIE	下記に応じてクリアします。 – FIFO 閾値設定 (SAI_xCR2 の FLTH ビット) – 通信方向 (トランスミッタ/レシーバ) 詳細については、 <a href="#">セクション 33.3.8: 内部 FIFO</a> を参照してください。
OVRUDR	エラー	マスタ/スレーブ 受信または送信モード	SAI_xIM レジスタの OVRUDRIE	SAI_xCLRFR レジスタで COVRUDR = 1
AFSDDET	エラー	スレーブ (AC'97 モード および SPDIF モードでは使用しません。)	SAI_xIM レジスタの AFSDDETIE	SAI_xCLRFR レジスタで CAFSDDET = 1
LFSDET	エラー	スレーブ (AC'97 モード および SPDIF モードでは使用しません。)	SAI_xIM レジスタの LFSDETIE	SAI_xCLRFR レジスタで CLFSDET = 1
CNRDY	エラー	スレーブ (AC'97 モードの場合のみ)	SAI_xIM レジスタの CNRDYIE	SAI_xCLRFR レジスタで CCNRDY = 1
MUTEDET	ミュート	マスタ/スレーブ 受信モードのみ	SAI_xIM レジスタの MUTEDETIE	SAI_xCLRFR レジスタで CMUTEDET = 1
WCKCFG	エラー	マスタ (SAI_xCR1 レジスタ で NODIV = 0 の場合)	SAI_xIM レジスタの WCKCFGIE	SAI_xCLRFR レジスタで CWCKCFG = 1

割り込みを有効にするには、次のシーケンスに従ってください。

1. SAI 割り込みを無効化します。
2. SAI を設定します。
3. SAI 割り込みソースを設定します。
4. SAI を有効化します。

## 33.5 SAI レジスタ

### 33.5.1 グローバル設定レジスタ (SAI\_GCR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:6 予約済み、常に 0 として読み出されます。

ビット 5:4 **SYNCOUT[1:0]** : 同期出力

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : 同期信号は出力されません。オーディオブロックが SPDIF として設定されている場合、SYNCOUT[1:0] は「同期信号出力なし」に設定する必要があります。

01 : ブロック A は他の SAI 用に追加された同期に使用します。

10 : ブロック B は他の SAI 用に追加された同期に使用します。

11 : 予約済み。両方のオーディオブロック (A および B) を無効化して、これらのビットを設定する必要があります。

ビット 3:2 予約済み、常に 0 として読み出されます。

ビット 1:0 **SYNCIN[1:0]** : 同期入力

これらのビットは、ソフトウェアによってセット/クリアされます。

このフィールドの設定方法については、表 180 : 外部同期の選択を参照してください。

両方のオーディオブロック (A および B) を無効化して、これらのビットを設定する必要があります。

2 つのオーディオブロックのいずれかが外部 SAI と同期して動作するように設定されている場合 (SAI\_ACR1 または SAI\_BCR1 レジスタで SYNCEN[1:0] = 01)、これらのビットは意味を持ちます。

### 33.5.2 設定レジスタ 1 (SAI\_ACR1 / SAI\_BCR1)

アドレスオフセット : ブロック A : 0x004

アドレスオフセット : ブロック B : 0x024

リセット値 : 0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCKDIV[3:0]				NODIV	Res.	DMAEN	SAIXEN
								rW	rW	rW	rW	rW		rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW



ビット 31:24 予約済み、常に 0 として読み出されます。

ビット 23:20 MCKDIV[3:0] : マスタクロック分周器。

これらのビットは、ソフトウェアによってセット/クリアされます。オーディオブロックがスレープモードで動作するとき、これらのビットは意味を持ちません。オーディオブロックを無効化して、これらのビットを設定する必要があります。

0000 : 分周比 1 のマスタクロック入力。

その他 : マスタクロック周波数は次式で計算されます。

$$F_{SCK\_x} = \frac{F_{SAI\_CK\_x}}{MCKDIV \times 2}$$

ビット 19 NODIV : 分周器なし。

このビットは、ソフトウェアによってセット/クリアされます。

0 : マスタクロックジェネレータは有効です。

1 : クロックジェネレータで分周器を不使用 (この場合、マスタクロック分周器ビットは無効)

ビット 18 予約済み、常に 0 として読み出されます。

ビット 17 DMAEN : DMA 有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : DMA は無効です。

1 : DMA は有効です。

**注 :** **オーディオブロックはリセット後にトランスミッタとして動作するようにデフォルト設定されていますので、受信モードでの DMA リクエストを回避するために、MODE[1:0] ビットは DMAEN をセットする前に設定する必要があります。**

ビット 16 SAIXEN : オーディオブロック有効化。ここで x は A または B です。

このビットはソフトウェアでセットされます。

オーディオブロックをスイッチオフするには、アプリケーションソフトウェアでこのビットを 0 に設定し、このビットが 0 に読み返されるまでポーリングする必要があります。これは、ブロックを完全に無効にするということです。このビットを 1 にセットする前に、それが 0 にセットされていることを確認します。そうしないと、有効なコマンドが考慮されないことになります。

このビットを使って、SAIx オーディオブロックの状態を制御することができます。オーディオフレーム転送の途中で無効化した場合、処理中の転送は完了し、このオーディオフレーム転送の転送後にセル全体が完全に無効化されます。

0 : SAIx オーディオブロックは無効です。

1 : SAIx オーディオブロックは有効です。

**注 :** **SAIx ブロックをマスタモードに設定した場合、SAIXEN ビットをセットする前に SAIx の入力にクロックが存在する必要があります。**

ビット 15:14 予約済み、常に 0 として読み出されます。

ビット 13 OUTDRIV : 出力駆動。

このビットは、ソフトウェアによってセット/クリアされます。

0 : SAIXEN がセットされたときオーディオブロック出力が駆動されます。

1 : このビットをセットした直後にオーディオブロック出力が駆動されます。

**注 :** **このビットは、オーディオブロックを有効化する前で、かつオーディオブロックを設定した後にセットする必要があります。**

ビット 12 MONO : モノラルモード

このビットは、ソフトウェアによってセット/クリアされます。このビットはスロット数が 2 のときにのみ意味を持ちます。モノラルモードを選択した場合、オーディオブロックがトランスミッタとして動作する場合にスロット 0 のデータがスロット 1 に複製されます。受信モードでは、スロット 1 は破棄され、スロット 0 から受け取ったデータのみが保存されます。詳細については、[セクション : モノラル/ステレオモード](#)を参照してください。

0 : ステレオモード

1 : モノラルモード

ビット 11:10 **SYNCEN[1:0]** : 同期有効化。

これらのビットは、ソフトウェアによってセット/クリアされます。オーディオサブブロックを無効化して、これらのビットを設定する必要があります。

00 : オーディオサブブロックは非同期モードです。

01 : オーディオサブブロックは他の内部オーディオサブブロックと同期します。この場合、オーディオサブブロックはスレーブモードに設定する必要があります。

10 : オーディオサブブロックは他の SAI 内蔵ペリフェラルと同期します。この場合、オーディオサブブロックはスレーブモードに設定する必要があります。

11 : 予約済み

**注 :** **オーディオサブブロックは、SPDIF モードが有効な場合、非同期として設定する必要があります。**

ビット 9 **CKSTR** : クロックストローブエッジ。

このビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、このビットを設定する必要があります。このビットは、SPDIF オーディオプロトコルでは意味を持ちません。

0 : SCK 立ち上がりエッジでの SAI の変化によって信号が生成され、SCK 立ち下がりエッジでの SAI のサンプリングによって信号が受信されます。

1 : SCK 立ち下がりエッジでの SAI の変化によって信号が生成され、SCK 立ち上がりエッジでの SAI のサンプリングによって信号が受信されます。

ビット 8 **LSBFIRST** : LSB ファースト。

このビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、このビットを設定する必要があります。AC'97 ではデータは常に MSB 先頭で転送されるため、このビットは AC'97 オーディオプロトコルでは意味を持ちません。SPDIF ではデータは常に LSB 先頭で転送されるため、このビットは SPDIF オーディオプロトコルでは意味を持ちません。

0 : データは MSB から先に転送されます。

1 : データは LSB から先に転送されます。

ビット 7:5 **DS[2:0]** : データサイズ。

これらのビットは、ソフトウェアによってセット/クリアされます。SPDIF プロトコルを選択した場合 (PRTCFCFG[1:0] ビット)、フレームおよびデータサイズは固定されているため、これらのビットは無視されます。コンパANDINGモードを選択した場合 (COMP[1:0] ビット)、データサイズはアルゴリズムによって 8 ビットに固定されているため、DS[1:0] は無視されます。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

000 : 予約済み

001 : 予約済み

010 : 8 ビット

011 : 10 ビット

100 : 16 ビット

101 : 20 ビット

110 : 24 ビット

111 : 32 ビット

ビット 4 予約済み、常に 0 として読み出されます。

ビット 3:2 **PRTCFG[1:0]** : プロトコル設定。

これらのビットは、ソフトウェアによってセット/クリアされます。オーディオブロックを無効化して、これらのビットを設定する必要があります。

00 : フリープロトコル。フリープロトコルを選択すると、オーディオブロックの強力な設定機能を使用し、大部分の設定レジスタビットとフレーム設定レジスタを設定して、特定のオーディオプロトコル（例えば、I2S、LSB/MSB 詰め、TDM、PCM/DSP）を処理することができます。

01 : SPDIF プロトコル

10 : AC'97 プロトコル

11 : 予約済み

ビット 1:0 **MODE[1:0]** : SAIx オーディオブロックモード

これらのビットは、ソフトウェアによってセット/クリアされます。SAIx オーディオブロックを無効化した場合、これらのビットを設定する必要があります。

00 : マスタトランスミッタ

01 : マスタレシーバ

10 : スレーブトランスミッタ

11 : スレーブレシーバ

**注 :** オーディオブロックを SPDIF モードに設定した場合、マスタ送信モードが強制されず (MODE[1:0] = 00)。マスタ送信モードでは、オーディオブロックはただちに FS およびクロックの生成を開始します。

## 33.5.3 設定レジスタ 2 (SAI\_ACR2 / SAI\_BCR2)

アドレスオフセット : ブロック A : 0x008

アドレスオフセット : ブロック B : 0x028

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
COMP[1:0]		CPL	MUTECONT[5:0]						MUTEV AL	MUTE	TRIS	FFLUSH	FTH			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	w	rW	rW	rW

ビット 31:16 予約済み。常に 0 として読み出されます。

ビット 15:14 **COMP[1:0]** : コンパANDINGモード。

これらのビットは、ソフトウェアによってセット/クリアされます。μ-Law と A-Law(対数) は CCITT G.711 勧告の一部であり、使用される補数のタイプは CPL ビットによって決まります。

データ伸張かデータ圧縮かは、MODE[0] ビットの状態で決まります。

オーディオブロックがトランスミッタとして設定された場合、データ圧縮が適用されます。

オーディオブロックがレシーバとして設定された場合、自動的にデータ伸張が適用されます。

詳細については、[セクション : コンパANDINGモード](#)を参照してください。

00 : コンパANDINGアルゴリズムなし

01 : 予約済み

10 : μ-Law アルゴリズム

11 : A-Law アルゴリズム

**注 :** コンパANDINGモードは、TDM が選択された場合にのみ適用されます。

ビット 13 **CPL** : 補数ビット。

このビットは、ソフトウェアによってセット/クリアされます。

このビットにより、コンパANDINGモードに使用される補数のタイプを指定します。

0 : 1 の補数表現。

1 : 2 の補数表現。

**注 :** コンパANDINGモードがμ-Law アルゴリズムまたは A-Law アルゴリズムの場合にのみ、このビットは有効です。

ビット 12:7 **MUTECONT[5:0]** : ミュートカウンタ

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは受信モードでのみ使用されます。

これらのビットに設定された値は、受信で検出された連続ミュートフレーム数と比較されます。ミュートフレーム数がこの値と一致すると、MUTEDET フラグがセットされ、MUTEDETIE ビットがセットされている場合、割り込みが生成されます。

詳細については、[セクション : ミュートモード](#)を参照してください。

**ビット 6 MUTEVAL** : ミュート値。

このビットは、ソフトウェアによってセット/クリアされます。また、オーディオブロックを有効にする前に書き込む必要があります。(SAIXEN ビット) オーディオブロックがトランスミッタとして動作し、スロット数が 2 以下で、かつ MUTE ビットがセットされている場合にのみ、このビットは意味を持ちます。

これより多くのスロットが指定された場合、ミュートモードでの送信時に送信されるビット値は、この MUTEVAL ビットの値とは無関係に 0 になります。

スロット数が 2 以下で MUTEVAL = 1 の場合、送信される各スロットのミュート値は、前のフレームで送信したものと同じになります。

詳細については、[セクション : ミュートモード](#)を参照してください。

0 : ミュートモード時にビット値 0 を送信します。

1 : ミュートモード時に最後の値を送信します。

**注 :** このビットは意味を持たないので、SPDIF オーディオブロックには使用すべきではありません。

**ビット 5 MUTE** : ミュート。

このビットは、ソフトウェアによってセット/クリアされます。このビットは、オーディオブロックがトランスミッタとして動作する場合にのみ意味を持ちます。スロット数が 2 以下の場合、ミュート値は MUTEVAL 値に関連付けられますが、スロット数が 2 より大きい場合、ミュート値は 0 になります。

詳細については、[セクション : ミュートモード](#)を参照してください。

0 : ミュートモードではありません。

1 : ミュートモードは有効です。

**注 :** このビットは意味を持たないので、SPDIF オーディオブロックには使用すべきではありません。

**ビット 4 TRIS** : データラインのトライステート管理。

このビットは、ソフトウェアによってセット/クリアされます。このビットは、オーディオブロックがトランスミッタとして設定された場合にのみ意味を持ちます。オーディオブロックが SPDIF モードに設定されている場合、このビットは使用されません。このビットは SAI を無効にして設定してください。

詳細については、[セクション : 非アクティブスロットでの出力データライン管理](#)を参照してください。

0 : スロットが非アクティブのときでも、SD 出力ラインは SAI によって駆動されます。

1 : 次のスロットが非アクティブの場合、最終アクティブスロットの最終データビットの終わりで SD 出力ラインを開放 (HI-Z) します。

**ビット 3 FFLUSH** : FIFO を一掃します。

このビットはソフトウェアでセットされます。常に 0 が読み出されます。このビットは SAI を無効にして設定してください。

0 : FIFO を一掃しません。

1 : FIFO を一掃します。このビットを 1 に設定すると、FIFO の一掃がトリガされます。すべての内部 FIFO ポインタ (読み出しと書き込み) がクリアされます。この場合、FIFO に残っているデータが失われます (送受信データはそれ以上失われません)。FIFO を一掃する前に、SAI DMA ストリーム/割り込みを無効化する必要があります。

**ビット 2:0 FTH** : FIFO 閾値。

このビットは、ソフトウェアによってセット/クリアされます。

000 : FIFO エンプティ

001 : FIFO の ¼

010 : FIFO の ½

011 : FIFO の ¾

100 : FIFO がフルのとき

101 : 予約済み

110 : 予約済み

111 : 予約済み

## 33.5.4 フレーム設定レジスタ (SAI\_AFRCR / SAI\_BFRCR)

アドレスオフセット : ブロック A : 0x00C

アドレスオフセット : ブロック B : 0x02C

リセット値 : 0x0000 0007

**注 :** このレジスタは、AC'97 および SPDIF オーディオプロトコルでは意味を持ちません。

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

ビット 31:19 予約済み、常に 0 として読み出されます。

ビット 18 **FSOFF** : フレーム同期オフセット。

このビットは、ソフトウェアによってセット/クリアされます。AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。オーディオブロックを無効化して、このビットを設定する必要があります。

0 : FS はスロット 0 の先頭ビットで開始されます。

1 : FS はスロット 0 の先頭ビットの 1 ビット前でアサートされます。

ビット 17 **FSPOL** : フレーム同期極性。

このビットは、ソフトウェアによってセット/クリアされます。このビットを使用して、FS 信号上のフレーム開始のレベルを設定します。AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。

オーディオブロックを無効化して、このビットを設定する必要があります。

0 : FS はアクティブラロー (立ち下がリエッジ) になります

1 : FS はアクティブハイ (立ち上がりエッジ) になります

ビット 16 **FSDEF** : フレーム同期の定義。

このビットは、ソフトウェアによってセット/クリアされます。

0 : FS 信号がフレーム開始信号になります

1 : FS 信号がフレーム開始信号 + チャネルサイド識別子になります

このビットがセットされている場合、SAI\_xSLOTR レジスタで指定されるスロット数は偶数である必要があります。これは、このスロット数の半分を左チャンネルに、残りのスロットを右チャンネルに、それぞれ割り当てることを意味します (例えば、このビットは、I2S プロトコルまたは MSB/LSB 詰めプロトコルなどに対して設定する必要があります)。

AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。オーディオブロックを無効化して、このビットを設定する必要があります。

ビット 15 予約済み、常に 0 として読み出されます。

ビット 14:8 **FSALL[6:0]** : フレーム同期アクティブレベル長。

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、オーディオフレームにおける FS 信号のアクティブレベルの長さを、

ビットクロック数、(SCK) + 1 (FSALL[6:0] + 1) で指定します。

AC'97 または SPDIF オーディオブロックの設定には、意味を持たないので、使用されません。

オーディオブロックを無効化して、これらのビットを設定する必要があります。

ビット 7:0 **FRL[7:0]** : フレーム長。

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、オーディオフレーム長を SCK クロックサイクル数で定義します。フレーム内のビット数は FRL[7:0] + 1 に等しくなります。

オーディオフレームで転送する最小ビット数は 8 でなければならず、その他の場合、オーディオブロックの動作は予測できません。これは、データサイズが 8 ビットで、SAI\_xSLOTR レジスタの NBSLOT[4:0] でスロット 0 が 1 つだけ指定されている場合の例です (NBSLOT[3:0] = 0000)。

マスタモードで、マスタクロック (MCLK\_x ピンで使用可能) が使用されている場合、フレーム長は、8 から 256 のうちの 2 の累乗の数に調整する必要があります。マスタクロックが使用されていない場合 (NODIV = 1)、フレーム長を 8 から 256 のうちの任意の値に設定することをお勧めします。

これらのビットは意味を持たないので、AC'97 または SPDIF オーディオブロックの設定には使用されません。

## 33.5.5 スロットレジスタ (SAI\_ASLOTR / SAI\_BSLOTR)

アドレスオフセット : ブロック A : 0x010

アドレスオフセット : ブロック B : 0x030

リセット値 : 0x0000 0000

**注 :** このレジスタは、AC'97 および SPDIF オーディオプロトコルでは意味を持ちません。

SLOTEN[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
				r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w

ビット 31:16 **SLOTEN[15:0]** : スロット有効化。

これらのビットは、ソフトウェアによってセット/クリアされます。  
各 SLOTEN ビットは 0 から 15 までのスロット位置に対応します (最大 16 スロット)。  
0 : 非アクティブスロット。  
1 : アクティブスロット。  
オーディオブロックを無効化して、スロットを有効にする必要があります。  
これらのビットは、AC'97 または SPDIF モードでは無視されます。

ビット 15:12 予約済み、常に 0 として読み出されます。

ビット 11:8 **NBSLOT[3:0]** : オーディオフレーム内スロット数。

これらのビットは、ソフトウェアによってセット/クリアされます。  
このビットフィールドに設定された値は、オーディオフレーム内のスロット数 + 1 を表します (非アクティブスロット数を含む)。最大スロット数は 16 です。  
SAI\_xFRCCR レジスタの FSDEF ビットがセットされている場合、スロット数は偶数である必要があります。  
オーディオブロックを無効化して、スロット数を設定する必要があります。  
これらのビットは、AC'97 または SPDIF モードでは無視されます。

ビット 7:6 **SLOTSZ[1:0]** : スロットサイズ

このビットは、ソフトウェアによってセット/クリアされます。  
スロットサイズは、データサイズ以上である必要があります。この条件が満たされない場合、SAI の動作は不確定となります。  
SD ラインの駆動方法については、[セクション : 非アクティブスロットでの出力データライン管理](#) を参照してください。  
オーディオブロックを無効化して、これらのビットを設定する必要があります。  
これらのビットは、AC'97 または SPDIF モードでは無視されます。  
00 : スロットサイズをデータサイズ (SAI\_xCR1 レジスタの DS[3:0] で指定) に一致させます。  
01 : 16 ビット  
10 : 32 ビット  
11 : 予約済み

ビット 1 予約済み、常に 0 として読み出されます。

ビット 4:0 **FBOFF[4:0]** : 先頭ビットオフセット

これらのビットは、ソフトウェアによってセット/クリアされます。  
このビットフィールドに設定された値は、スロット内の転送データの先頭ビット位置を定義します。これはオフセット値となっています。送信モードでは、データフィールド範囲外のビットは強制的に 0 に設定されます。受信モードでは、余分な受信ビットは無視されます。  
オーディオブロックを無効化して、これらのビットを設定する必要があります。  
これらのビットは、AC'97 または SPDIF モードでは無視されます。



## 33.5.6 割り込みマスクレジスタ 2 (SAI\_AIM / SAI\_BIM)

アドレスオフセット - ブロック A : 0x014

アドレスオフセット - ブロック B : 0x034

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LFSDETI E	AFSDETI E	CNRDYI E	FREQI E	WCKCFG E	MUTEDETI E	OVRUDRI E								
									rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済み、常に 0 として読み出されます。

ビット 6 **LFSDETI** : レイトフレーム同期検出割り込み有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

このビットをセットすると、SAI\_xSR レジスタの LFSDET ビットがセットされている場合は割り込みが生成されます。

このビットは、AC'97、SPDIF モードのとき、またはオーディオブロックがマスタとして動作するときは意味を持ちません。

ビット 5 **AFSDETI** : 予測フレーム同期検出割り込み有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

このビットをセットすると、SAI\_xSR レジスタの AFSDET ビットがセットされている場合は割り込みが生成されます。

このビットは、AC'97、SPDIF モードのとき、またはオーディオブロックがマスタとして動作するときは意味を持ちません。

ビット 4 **CNRDYI** : コーデックノットレディ割り込み有効化 (AC'97)

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

割り込みを有効化すると、オーディオブロックは、AC'97 フレームのスロット 0 (tag0) でこのラインに接続されているコーデックが動作可能状態 (レディ) であるか否かを検出します。動作可能状態にない場合は、SAI\_xSR レジスタの CNRDY フラグがセットされ、割り込みが生成されます。

このビットは、AC'97 モードが選択され (PRTCFCFG[1:0] ビット)、かつオーディオブロックがレシーバとして動作する場合にのみ意味を持ちます。

ビット 3 **FREQIE** : FIFO リクエスト割り込み有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

このビットをセットすると、SAI\_xSR レジスタの FREQ ビットがセットされている場合は割り込みが生成されます。

オーディオブロックはリセット後にトランスミッタとして動作するようにデフォルト設定されていますので、受信モードでの寄生割り込みを回避するために、MODE ビットは FREQIE をセットする前に設定する必要があります。

ビット 2 **WCKCFGIE** : 不正クロック設定割り込み有効化。

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

このビットは、オーディオブロックがマスタとして設定されている場合 (MODE[1] = 0 かつ NODIV = 0) のみ考慮されます。

SAI\_xSR レジスタの WCKCFG フラグがセットされている場合は割り込みが生成されます。

**注 :** このビットは TDM モードでのみ使用され、他のモードでは意味を持ちません。

ビット 1 **MUTEDETIE** : ミュート検出割り込み有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

このビットをセットすると、SAI\_xSR レジスタの MUTEDET ビットがセットされている場合は割り込みが生成されます。

オーディオブロックが受信モードに設定されている場合のみ、このビットは意味を持ちます。

ビット 0 **OVRRUDRIE** : オーバーラン/アンダーラン割り込み有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは無効です

1 : 割り込みは有効です

このビットをセットすると、SAI\_xSR レジスタの OVRUDRIE ビットがセットされている場合は割り込みが生成されます。

## 33.5.7 ステータスレジスタ (SAI\_ASR / SAI\_BSR)

アドレスオフセット - ブロック A : 0x018

アドレスオフセット - ブロック B : 0x038

リセット値 : 0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FLTH											
													r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR								
									r	r	r	r	r	r	r

ビット 31:19 予約済み、常に 0 として読み出されます。

ビット 18:16 **FLTH** : FIFO レベル閾値。

このビットは読み出し専用です。FIFO レベル閾値フラグはハードウェアにより管理され、その設定は SAI ブロックの設定 (送信モードまたは受信モード) によって決まります。

SAI ブロックをトランスミッタとして設定した場合

000 : FIFO エンプティ

001 : FIFO  $\leq \frac{1}{4}$  ただしエンプティではない

010 :  $\frac{1}{4} < \text{FIFO} \leq \frac{1}{2}$

011 :  $\frac{1}{2} < \text{FIFO} \leq \frac{3}{4}$

100 :  $\frac{3}{4} < \text{FIFO}$  ただしフルではない

101 : FIFO はフルです

SAI ブロックをレシーバとして設定した場合 :

000 : FIFO エンプティ

001 : FIFO  $< \frac{1}{4}$  ただしエンプティではない

010 :  $\frac{1}{4} \leq \text{FIFO} < \frac{1}{2}$

011 :  $\frac{1}{2} \leq \text{FIFO} < \frac{3}{4}$

100 :  $\frac{3}{4} \leq \text{FIFO}$  ただしフルではない

101 : FIFO 満杯

ビット 15:7 予約済み、常に 0 として読み出されます。

ビット 6 **LFSDET** : レイトフレーム同期検出。

このビットは読み出し専用です。

0 : エラーなし。

1 : フレーム同期信号が正しいタイミングで現れない。

オーディオブロックがスレーブモードに設定された場合のみ、このフラグをセットすることができます。

このフラグは、AC'97 または SPDIF モードでは使用されません。

SAI\_xIM レジスタの LFSDETIE ビットがセットされている場合は割り込みが生成されます。

ソフトウェアによって SAI\_xCLRFR レジスタの CLFSDET ビットをセットすると、このフラグはクリアされません。

ビット 5 **AFSDET** : 予測フレーム同期検出。

このビットは読み出し専用です。

0 : エラーなし。

1 : 予想より前にフレーム同期信号が検出されました。

オーディオブロックがスレーブモードに設定された場合のみ、このフラグをセットすることができます。

このフラグは、AC'97または SPDIF モードでは使用されません。

SAI\_xIM レジスタの AFSDETIE ビットがセットされている場合は割り込みが生成されます。

ソフトウェアによって SAI\_xCLRFR レジスタの CAFSDET ビットをセットすると、このフラグはクリアされま

す。

ビット 4 **CNRDY** : コーデックノットレディ。

このビットは読み出し専用です。

0 : 外部 AC'97 コーデックが動作可能状態です。

1 : 外部 AC'97 コーデックが動作可能状態ではありません。

SAI\_xCR1 レジスタで AC'97 オーディオプロトコルが選択され、かつ受信モードに設定された場合にのみ、このビットが使用されます。

SAI\_xIM レジスタの CNRDYIE ビットがセットされている場合は割り込みが生成されます。

ソフトウェアによって SAI\_xCLRFR レジスタの CCNRDY ビットをセットすると、このフラグはクリアされま

す。

ビット 3 **FREQ** : FIFO リクエスト。

このビットは読み出し専用です。

0 : FIFO リクエストはありません。

1 : SAI\_xDR に対する読み出しまたは書き込みの FIFO リクエストがあります。

このリクエストはオーディオブロック設定によって決まります。

– ブロックが送信モードに設定されている場合、FIFO リクエストは SAI\_xDR 内の書き込みリクエスト動作に関

わります。

– ブロックが受信モードに設定されている場合、FIFO リクエストは SAI\_xDR 内の読み出しリクエスト動作に関

わります。

SAI\_xIM レジスタの FREQIE ビットがセットされている場合、このフラグによって割り込みを生成することが

できます。

**ビット 2 WCKCFG** : 不正クロック設定フラグ。

このビットは読み出し専用です。

0 : クロック設定は正常です。

1 : クロック設定が [セクション 33.3.5 : フレーム同期](#) で規定するフレーム長仕様に関する規則に準拠していません (SAI\_xFRCR レジスタの FRL[7:0] ビットの設定)。

このビットは、オーディオブロックがマスタモードで動作する場合 (MODE[1] = 0 かつ NODIV = 0) にのみ使用されます。

SAI\_xIM レジスタの WCKCFGIE ビットがセットされている場合は割り込みが生成されます。

ソフトウェアによって SAI\_xCLRFR レジスタの CWCKCFG ビットをセットすると、このフラグはクリアされます。

**ビット 1 MUTEDET** : ミュート検出。

このビットは読み出し専用です。

0 : SD 入カラインでミュートは検出されません

1 : 指定数の連続オーディオフレームに対して SD 入カラインでミュート値が検出されました (0 値)

連続するオーディオフレームに対してオーディオフレームの各スロットで連続する 0 値が受信された場合、このフラグがセットされます (SAI\_xCR2 レジスタの MUTECONT ビットをセット)。

SAI\_xIM レジスタの MUTEDETIE ビットがセットされている場合は割り込みが生成されます。

ソフトウェアによって SAI\_xCLRFR レジスタの CMUTEDET ビットをセットすると、このフラグはクリアされます。

**ビット 0 OVRUDR** : オーバーラン/アンダーラン。

このビットは読み出し専用です。

0 : オーバーラン/アンダーランエラーはありません。

1 : オーバーラン/アンダーランエラーが検出されました。

オーバーランおよびアンダーラン状態は、オーディオブロックがそれぞれレシーバとトランスミッタとして設定されている場合にのみ発生します。

SAI\_xIM レジスタの OVRUDRIE ビットがセットされている場合は割り込みが生成されます。

ソフトウェアによって SAI\_xCLRFR レジスタの COVRUDR ビットをセットすると、このフラグはクリアされます。

## 33.5.8 クリアフラグレジスタ (SAI\_ACLRFR / SAI\_BCLRFR)

アドレスオフセット - ブロック A : 0x01C

アドレスオフセット - ブロック B : 0x03C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTED ET	COVRUDR								
									w	w	w		w	w	w

ビット 31:7 予約済み、常に 0 として読み出されます。

ビット 6 **CLFSDET** : レイトフレーム同期検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI\_xSR レジスタの LFSDET フラグがクリアされます。

このビットは、AC'97 または SPDIF モードでは使用されません。

このビットを読み出すと、常に 0 値が返されます。

ビット 5 **CAFSDET** : 予測フレーム同期検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI\_xSR レジスタの AFSDET フラグがクリアされます。

このビットは AC'97 または SPDIF モードでは使用されません。

このビットを読み出すと、常に 0 値が返されます。

ビット 4 **CCNRDY** : コーデックノットレディフラグのクリア

このビットは書き込み専用です。

このビットを 1 に設定すると SAI\_xSR レジスタの CNRDY フラグがクリアされます。

SAI\_xCR1 レジスタで AC'97 オーディオプロトコルが選択された場合にのみ、このビットが使用されます。

このビットを読み出すと、常に 0 値が返されます。

ビット 3 予約済み、常に 0 として読み出されます。

ビット 2 **CWCKCFG** : 不正クロック設定フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI\_xSR レジスタの WCKCFG フラグがクリアされます。

オーディオブロックがマスタとして設定され (MODE[1]=0)、かつ SAI\_xCR1 レジスタで NODIV = 0 の場合にのみ、このビットが使用されます。

このビットを読み出すと、常に 0 値が返されます。

ビット 1 **CMUTEDET** : ミュート検出フラグクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI\_xSR レジスタの MUTEDET フラグがクリアされます。

このビットを読み出すと、常に 0 値が返されます。

ビット 0 **COVRUDR** : オーバーラン/アンダーランクリア。

このビットは書き込み専用です。

このビットを 1 に設定すると SAI\_xSR レジスタの OVRUDR フラグがクリアされます。

このビットを読み出すと、常に 0 値が返されます。

## 33.5.9 データレジスタ (SAI\_ADR / SAI\_BDR)

アドレスオフセット - ブロック A : 0x020

アドレスオフセット - ブロック B : 0x040

リセット値 : 0x0000 0000

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

ビット 31:0 DATA[31:0] : データ

FIFO がフルでない場合、このレジスタへ書き込むと、FIFO への書き込みが行われます。

FIFO がエンプティでない場合、このレジスタを読み出すと、FIFO からの読み出しが行われます。

## 33.5.10 SAI レジスタマップ

次の表に SAI レジスタの一覧を示します。

表 186. SAI レジスタマップとリセット値

オフセット	レジスタとリセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	SAI_GCR	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	SYNCOUT[1:0]	予約済み	予約済み	予約済み	予約済み	SYNCIN[1:0]
	リセット値																											0	0			0	0
0x0004 または 0x0024	SAI_xCR1	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	MC.IDIV[3:0]	予約済み	予約済み	予約済み	NODIV	予約済み	DMAEN	SAIXEN	予約済み	予約済み	OUTDRV	MONO	SYNCEN[1:0]	予約済み	CKSTR	LSBFIRST	予約済み	予約済み	DS[2:0]	予約済み	PRTCFG[1:0]	予約済み	MODE[1:0]	
	リセット値									0	0	0	0	0		0	0			0	0	0	0	0	0	0	0	1	0	0	0	0	
0x0008 または 0x0028	SAI_xCR2	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	COMPT[1:0]	CPL	MUTECHN[5:0]					MUTE VAL	MUTE	TRIS	FLLUS	FTH				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x000C または 0x002C	SAI_xFRCR	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	FSOFF	FSPOL	FSDEF	FSALL[6:0]						FRL[7:0]									
	リセット値														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0x0010 または 0x0030	SAI_xSLOTR	SLOTEN[15:0]															予約済み	予約済み	予約済み	予約済み	NBSLOT[3:0]	予約済み	SLOTSZ[1:0]	予約済み	FBOFF[4:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0

表 186. SAI レジスタマップとリセット値 (続き)

オフセット	レジスタとリセット値	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0014 または 0x0034	SAI_xIM	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	LFSDET	AFSDETIE	CNRDYIE	FREQIE	WCKCFG	MUTEDET	OVRUDRIE
	リセット値																										0	0	0	0	0	0	0
0x0018 または 0x0038	SAI_xSR	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	FLVL[2:0]												LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
	リセット値														0	0	0										0	0	0	0	1	0	0
0x001C または 0x003C	SAI_xCLRFR	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	LFSDET	CAFSDET	CNRDY		WCKCFG	MUTEDET	OVRUDR
	リセット値																										0	0	0		0	0	0
0x0020 または 0x0040	SAI_xDR	DATA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 34 SPDIF 受信インタフェース (SPDIFRX)

### 34.1 SPDIFRX インタフェースの概要

SPDIFRX インタフェースを使用して、S/PDIF オーディオプロトコルを管理することができます。

### 34.2 SPDIFRX の主な機能

- 最大 4 個の入力が利用可能。
- 自動シンボルレート検出。
- 最大シンボルレート : 12.288 MHz
- 8~192 kHz のステレオストリームをサポート。
- デジタルオーディオインタフェース規格 IEC-60958 および IEC-61937、民生用アプリケーションをサポート。
- パリティビットの管理。
- オーディオサンプリング用の DMA を使用した通信。
- 制御およびユーザチャンネル情報用の DMA を使用した通信。
- 割り込み機能を搭載。

### 34.3 SPDIFRX の機能詳細

SPDIFRX ペリフェラルは、IEC-60958 および IEC-61937 に適合した S/PDIF フローを受信するように設計されています。これらの規格は、高いサンプリングレートに適した単純なステレオストリーム、およびドルビーサウンドや DTS サウンドと言われるような圧縮されたマルチチャンネルサラウンドサウンドをサポートしています。

レシーバは、シンボルレートを検出するために必要なすべての機能を提供し、受信データをデコードします。ユーザ情報およびチャンネル情報の専用パスを使用して、インタフェース処理を容易にすることができます。図 391 に簡略化されたブロック図を示します。

SPDIFRX\_DC ブロックは、SPDIFRX\_IN[4:1] 入力から受け取った S/PDIF ストリームをデコードするブロックです。このブロックは、受信信号の再サンプリングを行い、マンチェスタストリームをデコードし、フレーム、サブフレーム、ブロックの要素を認識します。また、このブロックは REG\_IF 部分、デコードされたデータ、および関連のステータスフラグを出力します。

ペリフェラルは APB1 バスに經由で完全に制御され、次の 2 つの DMA チャンネルを処理することができます。

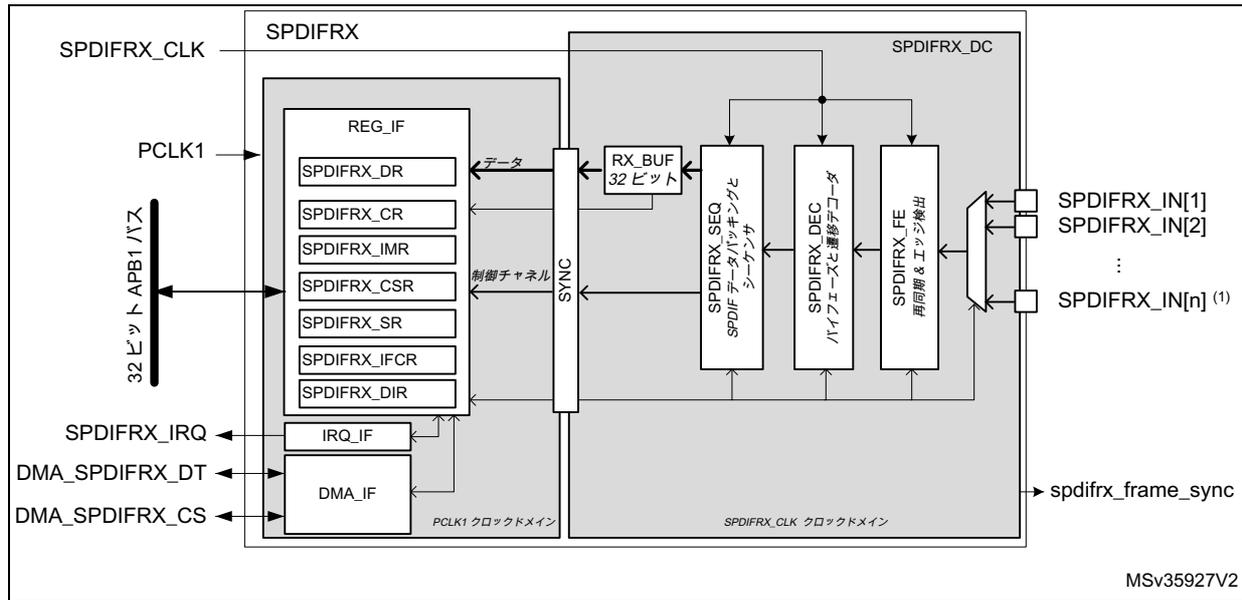
- オーディオサンプル転送専用の DMA チャンネル
- IEC60958 チャンネルステータスおよびユーザ情報の転送専用の DMA チャンネル

割り込みサービスは、DMA に対するオルタネート機能として、またはペリフェラルのシグナリングエラーやキーステータス用として使用することもできます。

また、SPDIFRX は、サブフレームのプリアンブルが検出されるたびにトグルする `spdifrx_frame_sync` と呼ばれる信号も提供します。したがって、デューティサイクルは 50 %、周波数はフレームレートに等しくなります。

この信号をタイマイイベントに接続して、周波数ドリフトを計算することができます。

図 391. SPDIFRX ブロック図



1. "n"は4に固定されています。

### 34.3.1 S/PDIF プロトコル (IEC-60958)

#### S/PDIF ブロック

S/PDIF フレームは 2 つのサブフレームで構成されています (図 393 を参照)。各サブフレームは 32 ビット (またはタイムスロットと呼ばれる) で構成されます。

- ビット 0~3 には同期プリアンプルのうちの 1 つが入っています。
- ビット 4~27 にはオーディオサンプルワードがリニア 2 の補数表現で入っています。最上位ビット (MSB) はビット 27 に書き込まれています。20 ビットコーディング範囲が使用されている場合、ビット 8~27 にはオーディオサンプルワードが入り、ビット 8 は LSB が入ります。
- ビット 28 (有効性ビット“V”) はデータが有効であるか否かを示します (たとえば、データをアナログ変換する場合)。
- ビット 29 (ユーザデータビット“U”) にはユーザデータ情報 (コンパクトディスクのトラック数など) が入っています。
- ビット 30 (チャンネルステータスビット“C”) には、チャンネルステータス情報 (サンプリングレート、複製保護など)。
- ビット 31 (パリティビット“P”) には、パリティビットが入ります。たとえば、偶数パリティの場合、ビット 4~31 のビット列に偶数個の“1”と偶数個の“0”が入るように送られてきます。

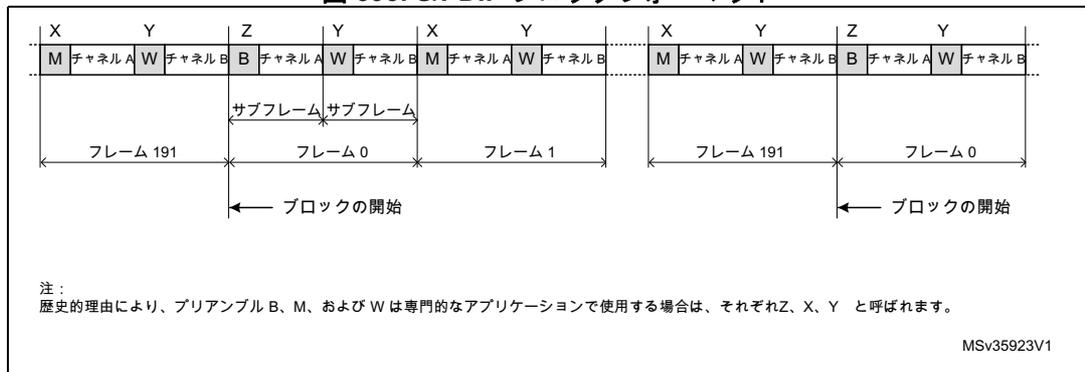
図 392. S/PDIF サブフレームフォーマット



線形コード化されたオーディオアプリケーションでは、最初のサブフレーム（左側、またはステレオ動作ではチャンネル A/モノラル動作では一次チャンネル）は通常プリアンブル M から始まります。ただし、プリアンブルは 192 フレームに 1 回プリアンブル B に変わること、チャンネルステータスおよびユーザ情報の構成に使用されるブロック構造の開始を識別します。2 番目のサブフレーム（右側、またはステレオ動作ではチャンネル B/モノラル動作では二次チャンネル）は常にプリアンブル W から始まります。

S/PDIF ブロックには 192 ペアの 32 ビットサブフレームが格納されています。

図 393. S/PDIF ブロックフォーマット

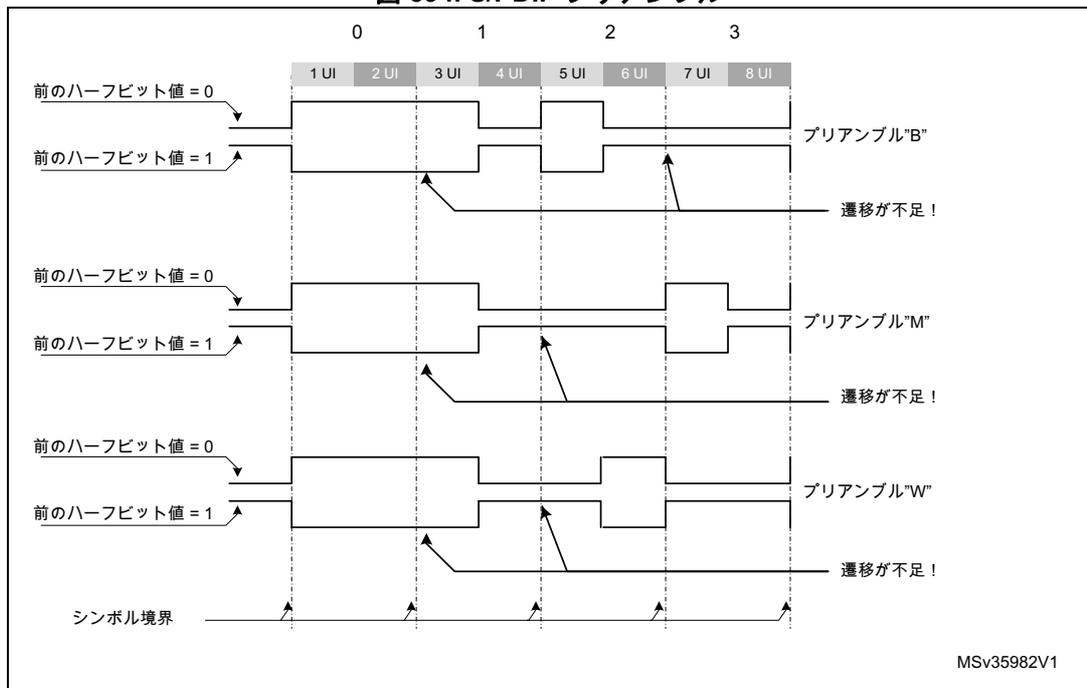


注：  
歴史的により、プリアンブル B、M、および W は専門的なアプリケーションで使用する場合は、それぞれ Z、X、Y と呼ばれます。

## 同期プリアンブル

プリアンブルパターンは、前のハーフビット値に応じて反転されたり、されなかったりします。ここで言う前のハーフビット値とは、最初のフレームの最初のプリアンブル B への転送を有効にする前のラインのレベルのことです。他のプリアンブルの場合の前のハーフビット値は、前のサブフレームのパリティビットの 2 番目のハーフビットのことです。プリアンブルパターン B、M、W を [図 394](#) に示します。

図 394. S/SPDIF プリアンブル



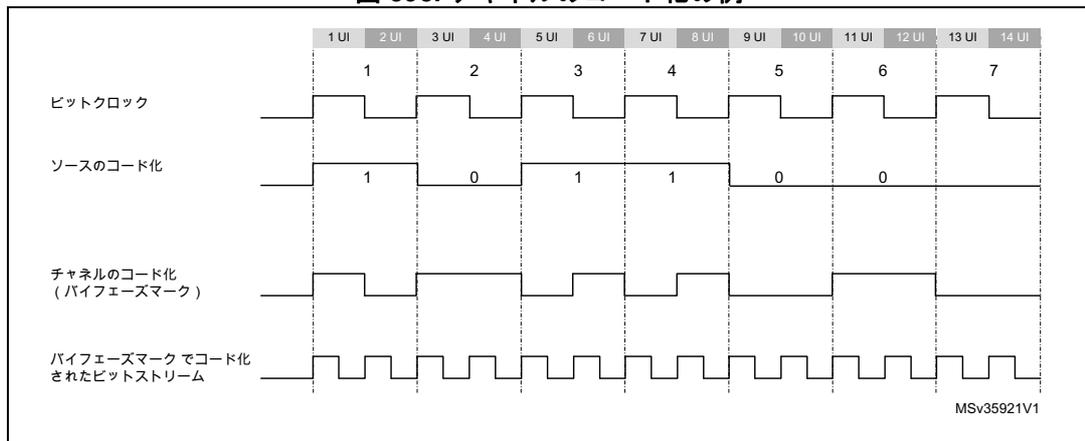
## 情報ビットのコード化

送信ライン上で DC 成分値を最小にし、データストリームからのクロックリカバリを容易にするために、ビット 4~31 をバイフェーズマークでエンコードします。

送信される各ビットは 2 つの連続したバイナリ状態から成るシンボルで表されます。シンボルの第 1 の状態とその前のシンボルの第 2 の状態は常に異なります。送信されるビットが論理“0”のとき、シンボルの第 2 の状態と第 1 の状態は同じですが、論理“1”の場合はそれらは異なります。このような状態は、IEC-60958 規格では“UI” (ユニットインターバル) 呼ばれます。

24 個のデータビットが LSB ファーストで転送されます。

図 395. チャンネルのコード化の例



### 34.3.2 SPDIFRX デコーダ (SPDIFRX\_DC)

#### 主要原理

S/PDIF ストリームをデコードするために SPDIFRX で使われる技法は、2 つの連続したエッジ間の時間間隔の測定に基づくものです。S/PDIF ストリームでは次の 3 種類の時間間隔が見られます。

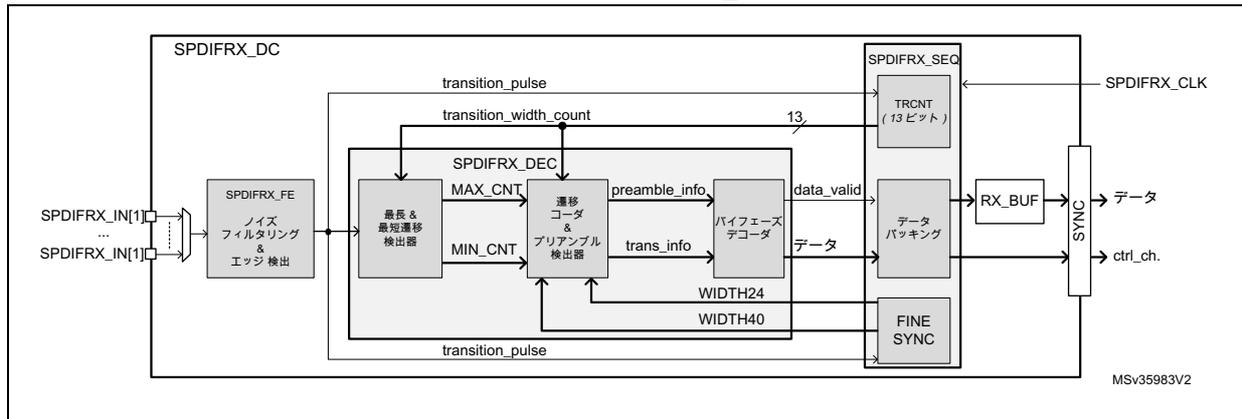
- 長い時間間隔 (TL)。継続時間は 3 x UI。プリアンブル時のみ出現。
- ある程度長い時間間隔 (TM)。継続時間は 2 x UI。一部のプリアンブルおよび情報フィールドの両方に出現。
- 短い時間間隔 (TS)。継続時間は 1 x UI。一部のプリアンブルおよび情報フィールドの両方に出現。

SPDIFRX\_DC ブロックは受信した S/PDIF ストリームをデコードするブロックです。このブロックは次の機能を果たします。

- 受信信号の再サンプリングとフィルタリング
- 時間間隔の概算
- シンボルレートおよび同期の概算
- シリアルデータのデコード、整合性チェック
- ブロックおよびサブフレームプリアンブルの検出
- シンボルレートの連続追跡

図 396 に SPDIFRX デコーダの詳細図を示します。

図 396. SPDIFRX\_DC デコーダ

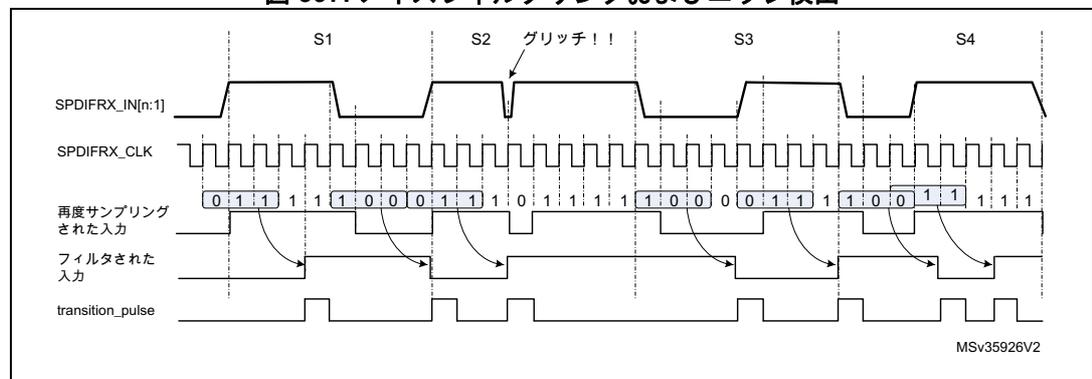


## ノイズフィルタリングおよび立ち上がり／立ち下がりエッジ検出

選択された SPDIFRX\_IN で受信した S/PDIF 信号は、SPDIFRX\_CLK クロック（取得クロック）を使用して再サンプリングされます。スプールを取り消すには、簡単なフィルタリングを適用します。このフィルタリングは、エッジ遷移を検出するステージで行われます。エッジ遷移は次のように検出されます。

- 立ち上がりエッジは、シーケンス 0 とそれに続く 2 個の“1”がサンプリングされたときに検出されます。
- 立ち下がりエッジは、シーケンス 1 とそれに続く 2 個の“0”がサンプリングされたときに検出されます。
- 立ち上がりエッジの後に立ち下がりエッジシーケンスが予期されます。
- 立ち下がりエッジの後に立ち上がりエッジシーケンスが予期されます。

図 397. ノイズフィルタリングおよびエッジ検出



## 最長および最短の遷移検出器

最長および最短の遷移検出器のブロックは、2 つの遷移間の最大 (MAX\_CNT) および最小 (MIN\_CNT) の継続時間を検出します。TRCNT カウンタを使用して、間隔の継続時間を測定します。カウンタのクロックは SPDIFRX\_CLK 信号によって供給されます。1 遷移パルスごとに、カウンタの値が保存され、カウンタはリセットされて再びカウントを開始します。

最大継続時間は通常プリアンブル期間中に確認されます。この最大継続時間は MAX\_CNT として送り出されます。最小継続時間は MIN\_CNT として送り出されます。

最長および最短遷移の検索は、遷移タイムがタイムアウトになった時点で停止されます。遷移タイムは、70 回の受信信号の遷移後にトリガを生成するウォッチドッグタイムのようなものです。70 回の遷移をカウントすることで、確実にサブフレームより少し長い遅延が発生しますのでご注意ください。

また、2 個のパルス間の間隔が長すぎて TRCNT がオーバーフローした場合、SPDIFRX が停止し、SPDIFRX\_SR レジスタの TERR フラグが 1 にセットされます。

## 遷移コードとプリアンブル検出器

「遷移コードとプリアンブル検出器」のブロックは MAX\_CNT および MIN\_CNT を受信します。また、TRCNT カウンタから現在の遷移幅も受信します (図 396 を参照)。このブロックは、現在の遷移幅を TH<sub>HI</sub> および TH<sub>LO</sub> と呼ばれる 2 つの異なる閾値と比較することでエンコードします。

- 現在の遷移幅が (TH<sub>LO</sub> - 1) より小さい場合、受信データはデータビット“1”の半分の部分であり、TS としてコード化されます。
- 現在の遷移幅が (TH<sub>LO</sub> - 1) より大きく、TH<sub>HI</sub> より小さい場合、受信データはデータビット“0”であり、TM としてコード化されます。
- 現在の遷移幅が (TH<sub>LO</sub> - 1) より大きい場合、受信データはプリアンブルの長パルスであり、TL としてコード化されます。
- そうでない場合、エラーコードが生成されます (FERR フラグがセットされます)。

閾値 TH<sub>HI</sub> と TH<sub>LO</sub> は 2 つの異なる方法で求められます。

ペリフェラルがその最初の同期 (「粗い同期」と呼ばれる) を実行している場合、閾値は次式で求められます。

- $TH_{LO} = MAX\_CNT / 2$
- $TH_{HI} = MIN\_CNT + MAX\_CNT / 2$

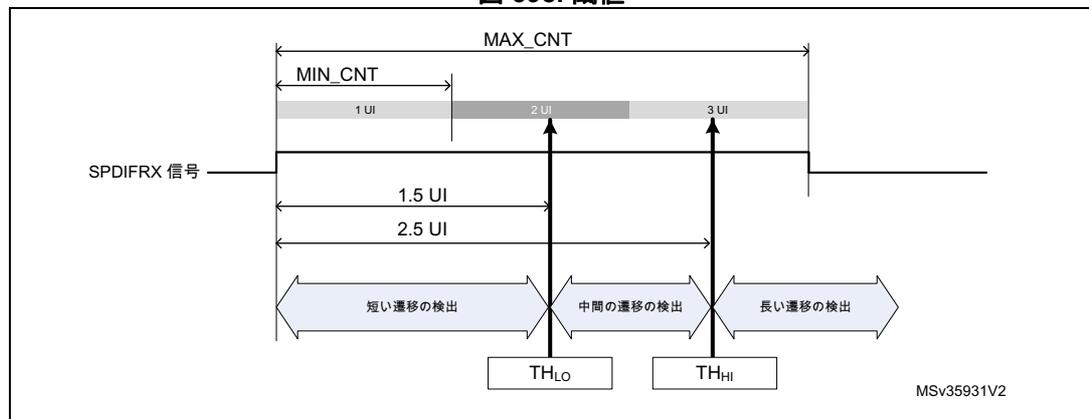
「粗い同期」モードが完了すると、SPDIFRX はより正確な基準を使用して閾値を算出します。SPDIFRX は、24 個のシンボル (WIDTH24) の長さを測定して TH<sub>LO</sub> を設定し、40 個のシンボル (WIDTH40) の長さを測定して TH<sub>HI</sub> を設定します。TH<sub>HI</sub> と TH<sub>LO</sub> は次のように計算します。

- $TH_{LO} = (WIDTH24) / 32$
- $TH_{HI} = (WIDTH40) / 32$

この 2 番目の同期フェーズは「細かい同期」と呼ばれます。詳細については、図 400 を参照してください。

下図に示すように、TH<sub>LO</sub> は 1.5 UI に等しく、TH<sub>HI</sub> は 2.5 UI に等しいのが理想的です。

図 398. 閾値



プリアンブル検出器は、特定のシーケンスの 4 個の連続した遷移をチェックして、それらがプリアンブルの一部を形成しているかどうかを判断します。TRANS0、TRANS1、TRANS2、TRANS3 が、前述のようにエンコードされた 4 個の連続した遷移を表しているとしましょう。表 187 に、1 つのプリアンブルを形成するこれら 4 個の遷移の値を示します。このパターンが検出されないと、これらの遷移はサブフレーム内のデータの一部を示し、バイフェーズデコーダがそれらをデコードするでしょう。

表 187. プリアンブルの遷移シーケンス

プリアンブルタイプ	バイフェーズデータパターン	TRANS3	TRANS2	TRANS1	TRANS0
プリアンブル B	11101000	TL	TS	TS	TL
プリアンブル M	11100010	TL	TL	TS	TS
プリアンブル W	11100100	TL	TM	TS	TM

### バイフェーズデコーダ

バイフェーズデコーダは、「遷移コードとプリアンブル検出器」のブロックが提供する遷移情報を使用して、入力バイフェーズマークが付いたデータストリームをデコードします。まず、プリアンブル検出情報を待ちます。プリアンブルが検出されたら、バイフェーズデコーダは次の遷移情報をデコードします。

- 受信した遷移情報が TM の場合、“0”としてデコードされます。
- 2 つの連続した TS は“1”としてデコードされます。
- その他の遷移シーケンスはエラー信号を生成します (FERR を 1 にセット)。

このように 28 個のデータビットをデコードし終わったら、バイフェーズデコーダは次のプリアンブルデータを探します。新たなプリアンブルが予想したものと違ってした場合、このブロックはエラー信号を生成します (FERR に 1 をセット)。エラーフラグの詳細については、[セクション 34.3.8 : 受信エラー](#)を参照してください。

### データのパッキング

このブロックは IEC-60958 フレームおよびブロックをデコードする役目があります。また、RX\_BUF または SPDIFRX\_CSR レジスタへの書き込みも処理します。

## 34.3.3 クロック偏差に対する SPDIFRX の許容誤差

クロック偏差に対する SPDIFRX の許容誤差は 1 ビットスロット内のサンプリングクロックサイクル数で決まります。SPDIFRX\_CLK が高速であればあるほど、その受信はより安定したものになります。SPDIFRX\_CLK 周波数とシンボルレートの比率は最低でも 11 とならなければなりません。

次の (少なくとも) 2 種類の現象により、受信品質が落ちる可能性があります。

- 2 個の連続した遷移間の遷移長さの差を反映したサイクル間ジッタ。
- サイクル間ジッタの累積的影響を反映したロングタームジッタ。これは低周波数シンボルの変調とみなされます。

## 34.3.4 SPDIFRX の同期

同期フェーズは、SPDIFRXEN を 0b01 または 0b11 にセットした時点で開始されます。[図 399](#) に同期プロセスを示します。

SPDIFRX\_CR レジスタの WFA ビットが 1 にセットされている場合、ペリフェラルは、同期プロセスを開始する前にまず選択された SPDIFRX\_IN ラインでアクティビティを検出する必要があります。選択された SPDIFRX\_IN ライン上の 4 個の遷移を検出することで、アクティビティを検出します。ペリフェラルはこの遷移状態がすべて検出されるまでこの状態にとどまります。この機能は粗い同期のモードに入ってしまった時に、同期エラーの誤検出を防ぐために特に有効です。詳細については、[セクション 34.4 : プログラミング手順](#)を参照してください。

ユーザは引き続き SPDIFRXEN を 0 にセットすることで、SPDIFRX を STATE\_IDLE に設定することができます。WFA が 0 にセットされている場合、ペリフェラルはアクティビティをチェックせずに粗い同期モードを開始します。

次のステップでは、細かい同期 (FINE SYNC) モードを実行するために、閾値 (COARSE SYNC) の最初の概算を行います。SPDIFRX ラインの障害により、初回のプロセスが実行されないという事態が発生する可能性があります。このような事態を回避するため、ユーザは SERR エラーフラグをセットする前に許容リトライ回数 (NBTR) をプログラムすることができます。

SPDIFRX が 24 個および 40 個の連続したシンボルの継続時間を正確に測定できたら、細かい同期 (FINE SYNC) モードは完了し、閾値が更新され、SYNCD フラグが 1 にセットされます。詳細については、[セクション : 遷移コードとプリアンブル検出器](#)を参照してください。

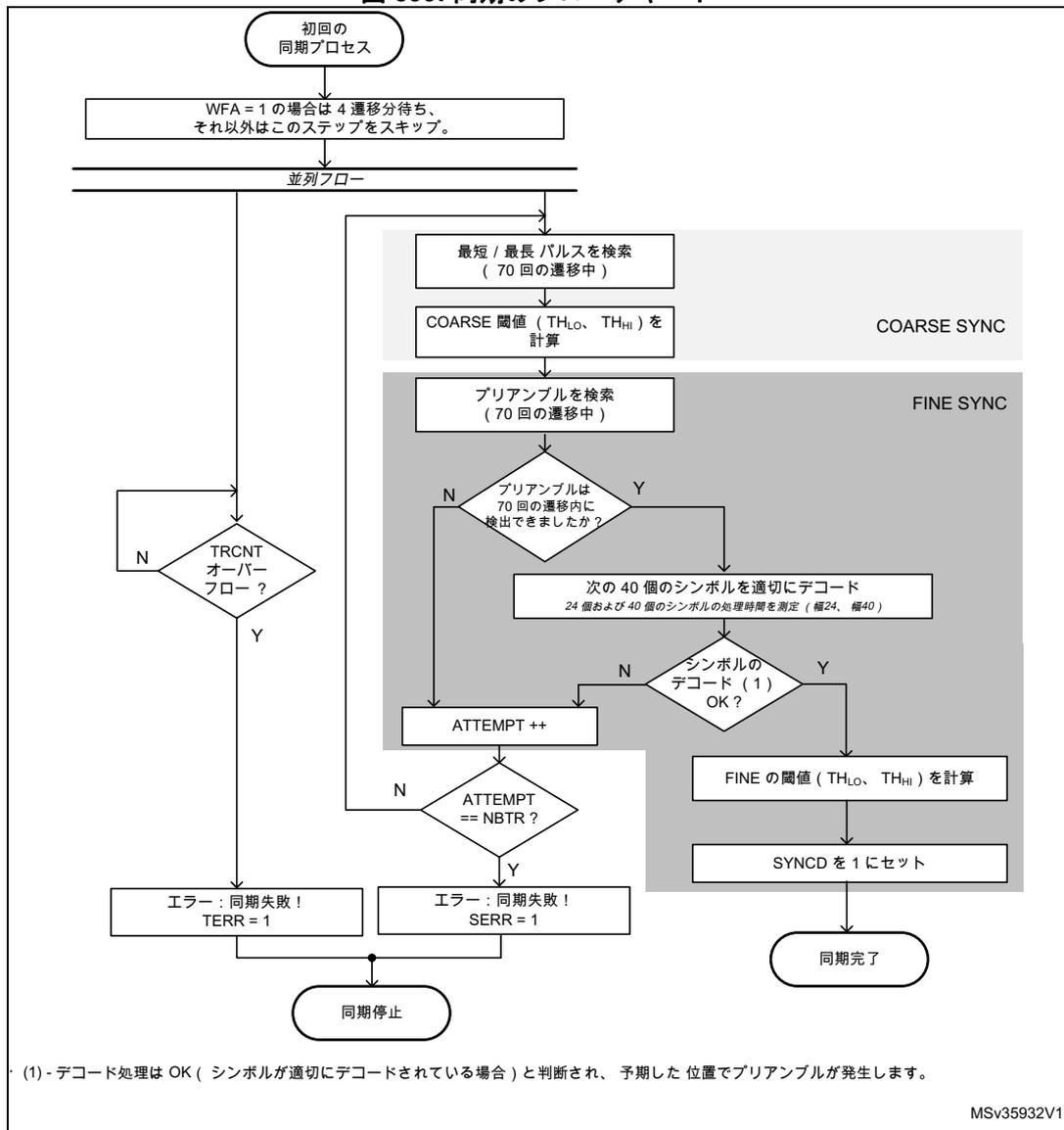
次の 2 種類のエラーが検出されます。

- TRCNT がオーバーフローしました。このエラーは通常、有効な S/PDIF ストリームが入力ラインにないことを意味します。このオーバーフローは TERR フラグで示されます。
- 再試行回数がプログラムされた値に達しました。このエラーは、S/PDIF 信号に大きなジッタが存在することを意味します。このエラーは SERR フラグで示されます。

最初の FINE SYNC が完了すると、次のプリアンブル B が検出された時点でチャンネルステータス (C) およびユーザデータ (U) の受信が始まります ([図 403](#)を参照)。次に、ユーザは SPDIFRX\_CSR レジスタにより IEC-60958 C および U ビットを読み出すことができます。この情報を基に、ユーザは DRFMT および RXSTEO の適切な設定値を選択できます。たとえば、現在のオーディオストリームによるエンコードされたデータの転送を検出した場合、データ受信を開始する前に、RXSTEO に 0、DRFMT に 0b10 をセットすることができます。DRFMT および RXSTEO は SPDIFRXEN = 0b11 の場合は修正できませんので注意してください。SPDIFRXEN がすでに 0b11 である場合、これらのフィールドへの書き込みは無視されます。ただし、SPDIFRXEN を 0b11 にせよという同じ書き込み命令でこれらのフィールドを変更することはできません。

その後、SPDIFRX は SPDIFRXEN = 0b11 となり、プリアンブル B が検出されるのを待ってから、オーディオサンプルの保存を開始します。

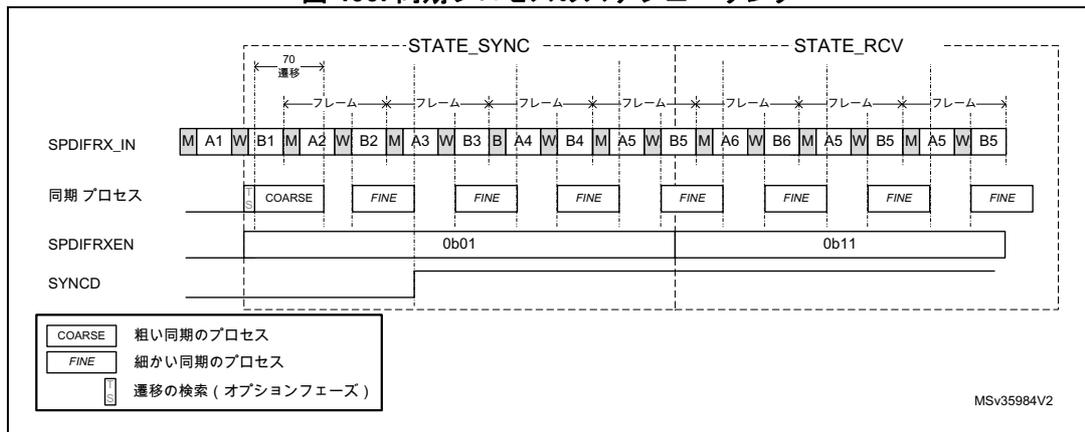
図 399. 同期のフローチャート



TRCNT オーバーフローに関する詳細情報は、[フレーム構造と同期エラー](#)を参照してください。

FINE SYNC プロセスはフレームごとに再トリガされます。この目的は、[図 400](#) に示す閾値を更新して、S/PDIF 同期を連続的に追跡することです。

図 400. 同期プロセスのスケジューリング



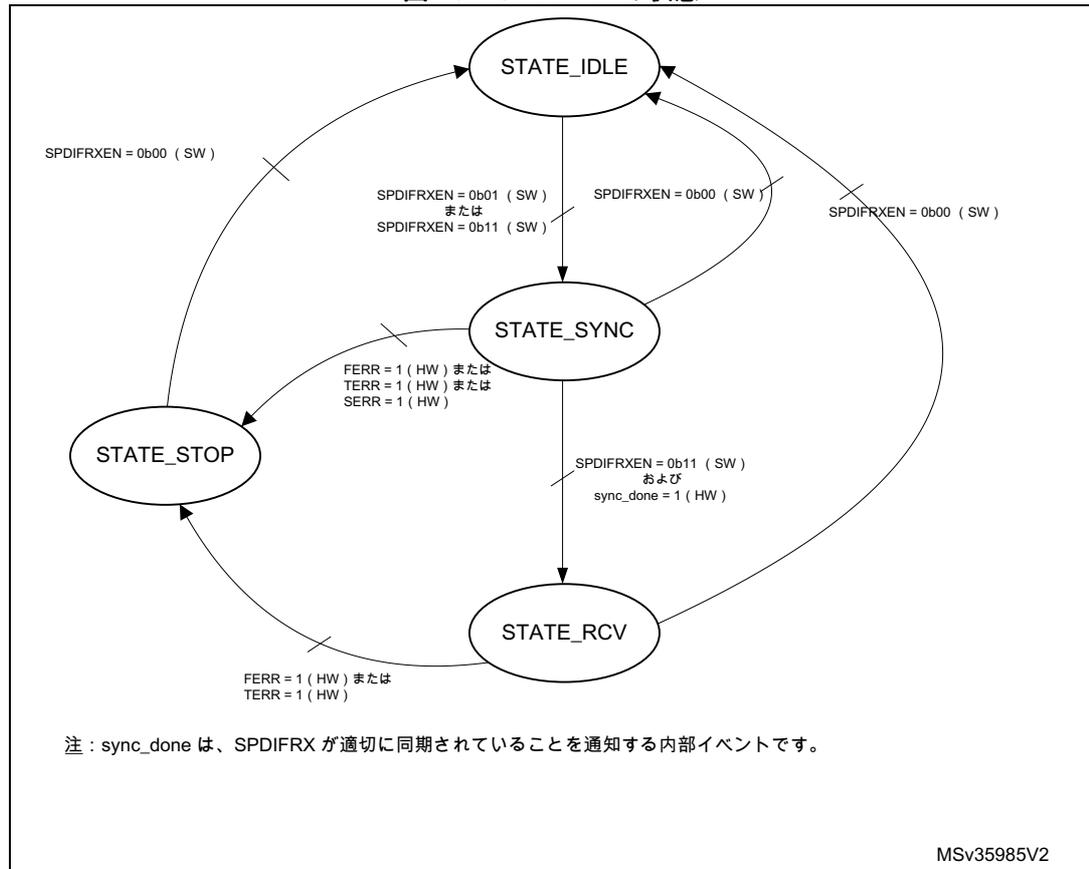
### 34.3.5 SPDIFRX 処理

ソフトウェアで SPDIFRXEN フィールドを使用して SPDIFRX の状態を制御することができます。SPDIFRX は次の状態のいずれかになり得ます。

- STATE\_IDLE :  
ペリフェラルが無効化され、SPDIFRX\_CLK ドメインがリセットされた状態です。PCLK1 ドメインは機能しています。
- STATE\_SYNC :  
ペリフェラルはストリームに同期され、閾値が定期的に更新され、DMA の割り込みによりユーザおよびチャネルステータスを読み出すことができる状態です。オーディオサンプルは受信バッファに提供されません。
- STATE\_RCV :  
ペリフェラルはストリームに同期され、閾値が定期的に更新され、割り込みまたはDMA チャンネルにより、ユーザ、チャネルステータスおよびオーディオサンプルを読み出すことができる状態です。SPDIFRXEN が 0b11 の場合、SPDIFRX はプリアンブル B を待ってから、オーディオサンプルの保存を開始します。
- STOP\_STATE :  
ペリフェラルはもはや同期されず、ユーザ、チャネルステータス、オーディオサンプルの受信が停止した状態です。ソフトウェアによる SPDIFRX の再開が待たれます。

図 401 に、起こり得る SPDIFRX の状態、およびある状態から別の状態への遷移方法を示します。ソフトウェア制御されているビットの後ろに“(SW)”, IP 制御されているビットの後ろに“(HW)”を付けます。

図 401. SPDIFRX の状態



SPDIFRX が STATE\_IDLE である場合

- ソフトウェアで SPDIFRXEN を 0b01 または 0b11 にセットすることで、STATE\_SYNC へ遷移できます。

SPDIFRX が STATE\_SYNC である場合

- 同期に失敗した場合、または受信データが適切にデコードされておらず、再同期なしで復帰する可能性はない場合 (FERR、SERR、または TERR = 1)、SPDIFRX は STATE\_STOP になり、ソフトウェアによる確認応答を待ちます。
- 同期フェーズが完了したとき、SPDIFRXEN = 0b01 であればペリフェラルはこの状態にとどまります。
- ソフトウェアにより SPDIFRXEN を 0 にセットできるときはいつでも、SPDIFRX はすぐに STATE\_IDLE に戻ります。DMA 転送中は正常に完了します。
- SPDIFRXEN = 0b11 かつ SYNCDC = 1 のとき、SPDIFRX は STATE\_RCV になります。

SPDIFRX が STATE\_RCV である場合

- 受信データが適切にデコードされておらず、再同期なしで復帰する可能性はない場合 (FERR、SERR、または TERR = 1)、SPDIFRX は STATE\_STOP になり、ソフトウェアによる確認応答を待ちます。
- ソフトウェアにより SPDIFRXEN を 0 にセットできるときはいつでも、SPDIFRX はすぐに STATE\_IDLE に戻ります。DMA 転送中は正常に完了します。

SPDIFRX が STATE\_STOP である場合

- SPDIFRX は受信および同期を停止し、エラーフラグをクリアするために、ソフトウェアによって SPDIFRXEN ビットが 0 にセットされるのを待ちます。

SPDIFRXEN が 0 にセットされている場合、IP は無効化されます。これは、ステートマシンがすべてリセットされ、RX\_BUF が一掃されることを意味します。FERR、SERR、および TERR フラグもリセットされますのでご注意ください。

## 34.3.6 データ受信管理

SPDIFRX はオーディオサンプル受信のためにダブルバッファを提供します。32 ビットのバッファが、SPDIFRX\_CLK クロックドメイン (RX\_BUF) 内および SPDIFRX\_DR レジスタに配置されます。RX\_BUF に格納されている有効データは、SPDIFRX\_DR がエンプティであればすぐに SPDIFRX\_DR へ転送されます。

RX\_BUF に格納されている有効データは、次の 2 つの条件が満たされると SPDIFRX\_DR へ転送されます。

- パリティビット (P) と次のプリアンプルの間の遷移が検出されます (ワードが完全に受信されたことを意味します)。
- SPDIFRX\_DR はエンプティです。

2 ワードバッファを備えることで、遅延の制約に対する柔軟性が向上します。

最大許容遅延を求める計算式:  $T_{SAMPLE} - 2T_{PCLK} - 2T_{SPDIFRX\_CLK}$

ここで  $T_{SAMPLE}$  は受信したステレオオーディオサンプルのオーディオサンプリングレートであり、 $T_{PCLK}$  は PCLK1 クロックの周期、 $T_{SPDIFRX\_CLK}$  は SPDIFRX\_CLK クロックの周期です。

SPDIFRX により、DMA (spdifrx\_dma\_req/clr\_d) または割り込みによるオーディオサンプルのメモリ転送のいずれかを使用することができます。DMA を選択することをお勧めします。詳細については、[セクション 34.3.10 : DMA インタフェース](#) を参照してください。

SPDIFRX では、受信データを処理する方法がいくつかあります。ユーザは、制御情報およびオーディオサンプルに関して個別のフローを得るか、またはそれらすべてを 1 つにまとめることができます。

サブフレームごとに、データ受信レジスタの SPDIFRX\_DR には 24 個のデータビットと、オプションの V、U、C、PE ステータスビット、および PT が格納されています ([データの混合と制御フロー](#) を参照)。

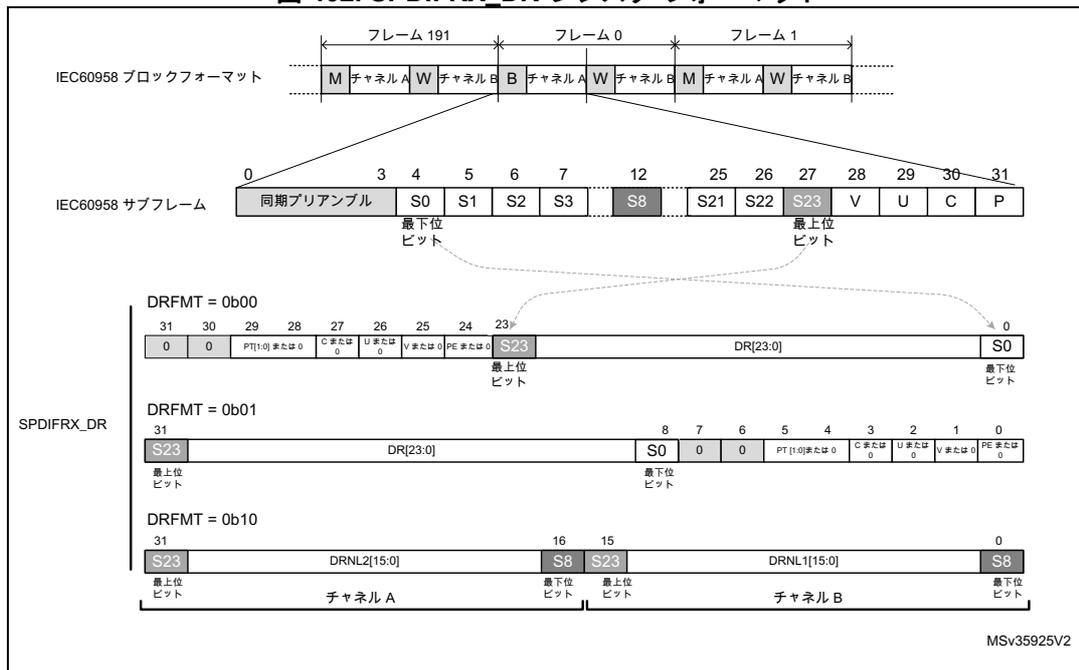
PE ビットはパリティエラービットを表し、デコードされたサブフレームでパリティエラーが検出されると、このビットには 1 がセットされます。

PT フィールドにはプリアンプルタイプ (B、M、または W) が表示されます。

V、U、および C は SPDIF インタフェースから受信した値を直接コピーしたものです。

DRFMT ビットにより、[図 402](#) に示す 3 種類のオーディオフォーマットから選択することができます。

図 402. SPDIFRX\_DR レジスタ フォーマット



DRFMT を 0b00 または 0b01 にセットすることで、データを SPDIFRX\_DR レジスタ内に右詰めまたは左詰めで格納することができます。ステータス情報は、ソフトウェアで希望する情報の処理方法に応じて、有効化するか、強制的に 0 に設定するかを選択できます。

DRFMT = 0b10 のときに与えられるフォーマットは、サブフレーム当たり 16 ビットしか使われないことからノンリニアモードを対象としています。このフォーマットを使うことにより、2 個の連続したサブフレームのデータを SPDIFRX\_DR に格納し、メモリ半分にします。なお、RXSTEO = 1 のとき、ミスマライメントの危険性はありません（つまり、ChA からのデータは常に SPDIFRX\_DR[31:16] に格納されます）。RXSTEO = 0 のときに起こり得るミスマライメントはオーバーランの状況です。その場合、SPDIFRX\_DR[31:16] には常に一番古い値が含まれ、SPDIFRX\_DR[15:0] にはより新しい値が含まれます（図 404 を参照）。

このフォーマットでは、ステータス情報をデータと混在させることはできませんが、ユーザはその情報を SPDIFRX\_CSR レジスタで取得し、専用の DMA チャンネルまたは割り込みを利用した転送を使用して情報をメモリに転送することができます（セクション 34.3.7: 専用制御フローを参照）。

## データの混合と制御フロー

ユーザはこのモードの使用を選択して、制御フローを十分な柔軟性を持って処理することができます。また、データレジスタ (SPDIFRX\_DR) 内のどのフィールドを確保すべきかも選択できます。

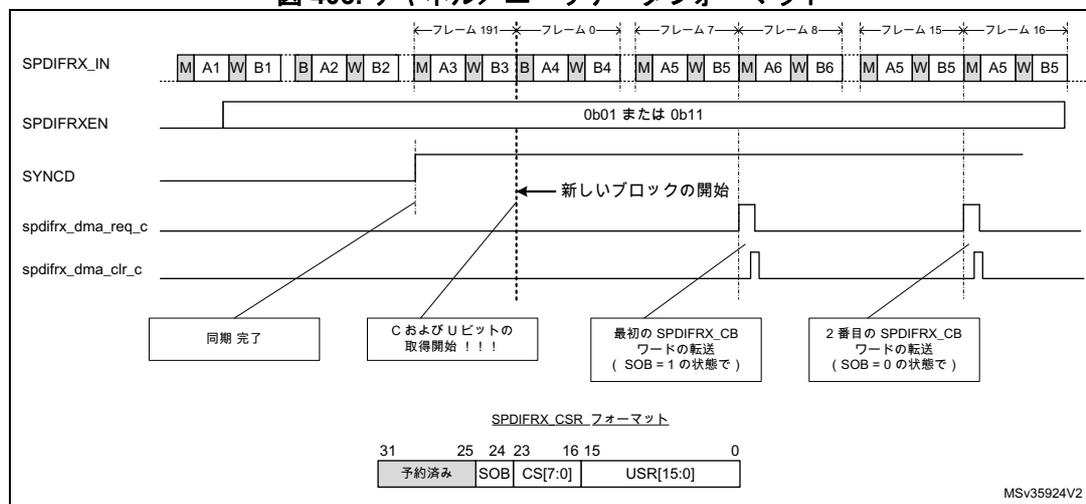
- PMSK ビット = 1 のとき、パリティエラー情報はマスクされます (0 にセットされる)。そうしないと、SPDIFRX\_DR にコピーされます。
- VMSK ビット = 1 のとき、有効性情報はマスクされます (0 にセットされる)。そうしないと、SPDIFRX\_DR にコピーされます。
- CUMSK ビット = 1 のとき、チャンネルステータスおよびユーザデータ情報はマスクされず (0 にセットされる)。そうしないと、SPDIFRX\_DR にコピーされます。
- PTMSK ビット = 1 のとき、プリアンプルタイプはマスクされます (0 にセットされる)。そうしないと、SPDIFRX\_DR にコピーされます。

## 34.3.7 専用制御フロー

SPDIFRX により、専用 DMA チャンネルを使用してユーザデータとチャンネルステータス情報の両方をキャッチすることができます。この機能を使用することで、SPDIFRX はチャンネルステータス情報とユーザ情報を継続的に取得することができます。取得は IEC 60958 ブロックの開始時に開始されます。このパスを制御するため使用可能な 2 つのフィールドは、CBDMAEN と SPDIFRXEN です。SPDIFRXEN が 0b01 または 0x11 にセットされている場合、取得は同期フェーズの完了後に開始されます。8 個のチャンネルステータスビットおよび 16 個のユーザデータビットが受信されると、それらはパックされて SPDIFRX\_CSR レジスタに格納されます。CBDMAEN ビットが 1 にセットされると、DMA リクエストがトリガされます (図 403 を参照)。

CS[0] が新しいブロックの先頭ビットに対応する場合、SOB ビットは 1 にセットされます。セクション 34.5.8 : チャンネルステータスレジスタ (SPDIFRX\_CSR) を参照してください。CHSEL ビットを使用して、チャンネル A または B のどちらからのチャンネルステータス情報 (C) をユーザが選択できます。

図 403. チャンネル/ユーザデータフォーマット



- 注: ブロックの最初の開始が検出されると (プリアンブル B)、SPDIFRX は 8 フレームおきにプリアンブルタイプをチェックします。
- 注: SPDIFRX\_DR レジスタのオーバーランエラーはこのパスには影響を与えません。

## 34.3.8 受信エラー

### フレーム構造と同期エラー

SPDIFRX は、次のいずれかの条件が発生した場合にエラーを検出します。

- 以下の条件では、FERR ビットは 1 にセットされます。
  - 28 個の情報ビットのそれぞれについて、1 つのシンボル遷移シーケンスが正しくない場合。たとえば、ショートパルスがペアでグループ化されていない場合。
  - プリアンブルが予期せぬ場所で発生した場合、または予期したプリアンブルが受信されない場合。
- 同期に失敗すると SERR ビットがセットされます。これは、リトライ回数がプログラムされた値を超えたためです。
- 2 個の遷移間の幅を概算するために使われるカウンタがオーバーフローすると TERR ビットがセットされます (TRCNT)。
 

SPDIFRX\_CLK クロックの 8129 周期中に一度も遷移が検出されない場合にオーバーフローが発生します。オーバーフローは最大でも 11.6 フレームの時間間隔を表します。

上記のフラグのうちのいずれかが 1 になると、選択された SPDIFRX\_IN のトラフィックは無視され、ISPDIFRX\_CR レジスタの IFEIE ビットがセットされていれば割り込みが生成されます。

上記のエラーのいずれかが発生した場合の通常の手順は次の通りです。

- SPDIFRXEN を 0 にセットして、エラーフラグをクリアします。
- SPDIFRXEN を 0b01 または 0b11 にセットして、IP を再開します。

詳細については、[図 401](#)を参照してください。

### パリティエラー

各サブフレームでは、28 個の情報ビット内に偶数個の“0”および“1”が入ることが予想されます。そうでない場合、SPDIFRX\_SR レジスタのパリティエラービット (PERR) がセットされ、SPDIFRX\_CR レジスタのパリティ割り込みイネーブルビット PERRIE がセットされている場合は割り込みが生成されます。割り込みがペンディング状態のままでも、データの受信は一時中止することなく、SPDIFRX は SPDIFRX\_DR ヘデータを送り出し続けます。

割り込みの確認応答をするには、PERRCF ビットによって PERR フラグをクリアします。

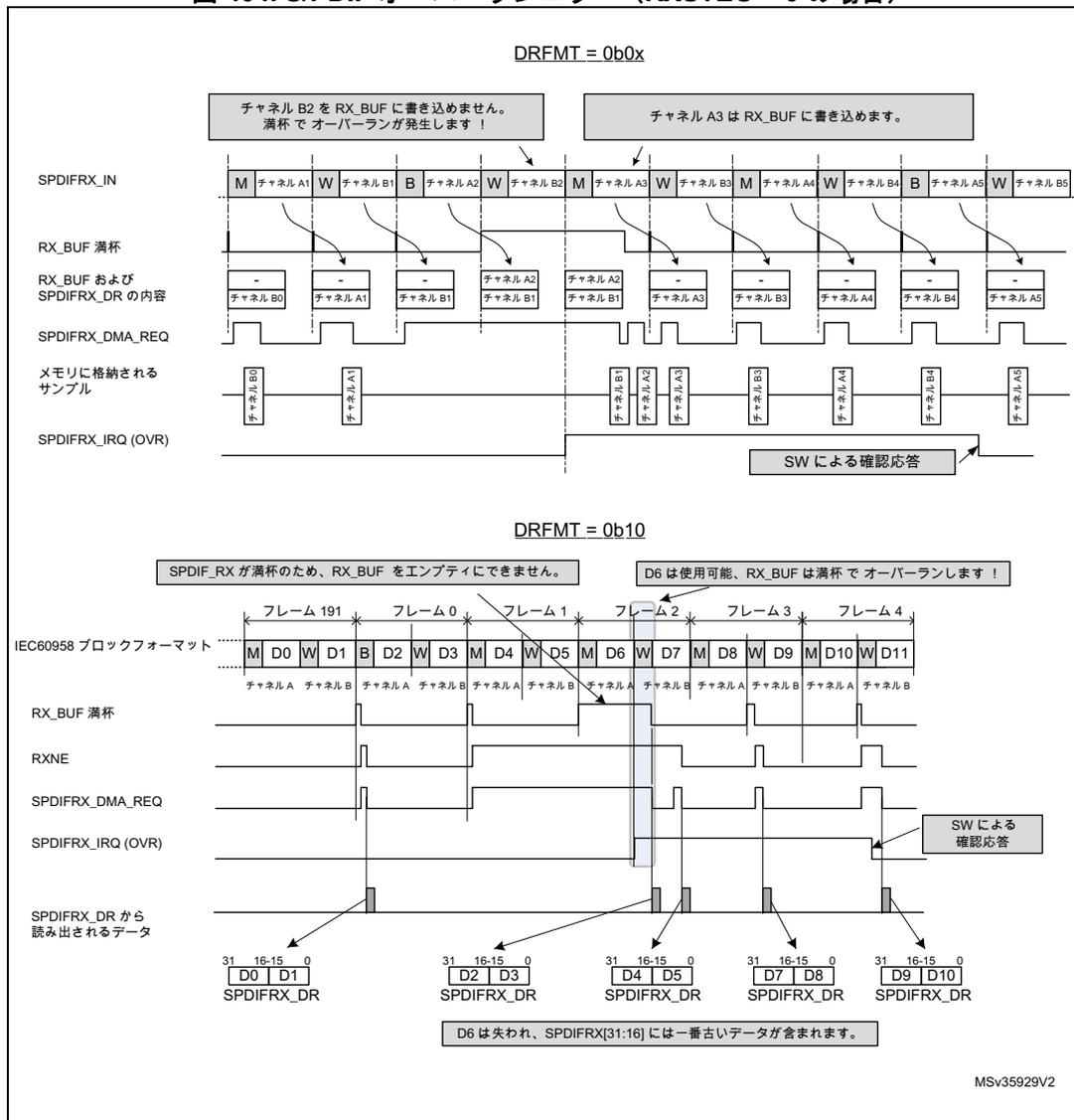
SPDIFRX\_DR レジスタで読み出されたデータと PERR ビットの値の間の一貫性をソフトウェアで保証したい場合は、PMSK ビットを 0 にセットする必要があります。

### オーバーランエラー

SPDIFRX\_DR と RX\_BUF の両方がフルの場合、SPDIFRX\_DC が RX\_BUF に新しいサンプルを書き込む必要がある場合、この新しいサンプルは捨てられ、オーバーラン状態が開始されます。SPDIFRX\_SR レジスタの OVR オーバーランエラーフラグがセットされ、SPDIFRX\_CR レジスタの OVRIE ビットがセットされている場合は割り込みが生成されます。

RXSTEO ビットが 0 にセットされている場合、RX\_BUF がエンプティになるとすぐに、IP は OVR フラグがペンディング中であっても次の受信データを格納します。この主な目的は、サンプルの損失をできるだけ少なくすることです。なお、動作は DRFMT 設定値に関係なく同様です。[図 404](#)を参照してください。

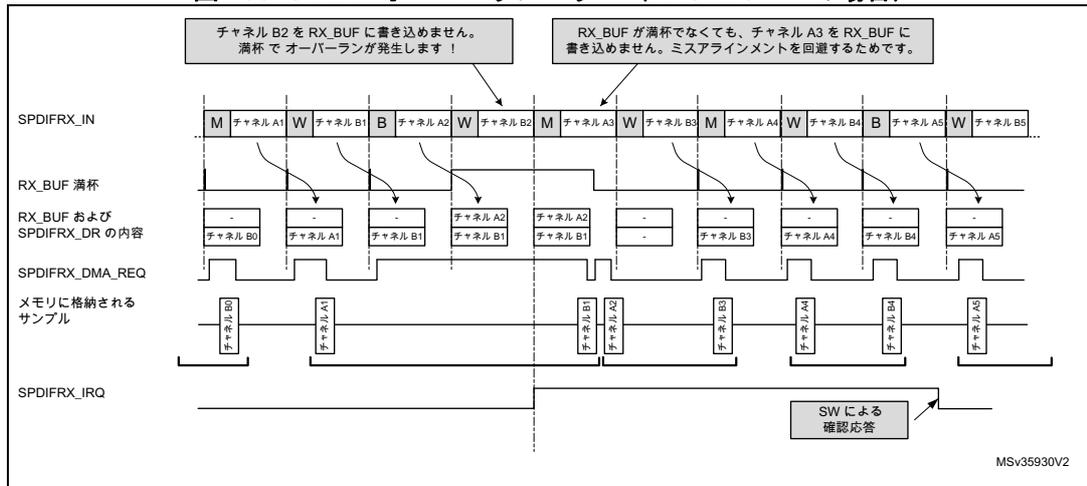
図 404. S/PDIF オーバーランエラー (RXSTEO = 0 の場合)



RXSTEO ビットが 1 にセットされている状態はステレオデータが転送されることを意味しますので、SPDIFRX は左チャンネルと右チャンネルの間のミスアラインメントを避ける必要があります。したがって、ペリフェラルは、たとえ RX\_BUF 内に余裕があったとしても、ミスアラインメントを避けるために 2 番目のサンプルを捨てなければなりません。さらに、OVR フラグがまだペンディング状態であっても、通常は受信サンプルを RX\_BUF に書き込むことができます。図 405 を参照してください。

OVRFCF ビットを 1 にセットすることによって、OVR フラグはソフトウェアによってクリアされます。

図 405. S/PDIF オーバーランエラー (RXSTEO = 1 の場合)



## 34.3.9 クロック供給の方法

SPDIFRX ブロックには次の 2 種類のクロックが必要です。

- APB1 クロック (PCLK1) : レジスタインタフェースに使われます。
- SPDIFRX\_CLK : 主に SPDIFRX\_DC 部分によって使われます。これらのクロックは非同期クロックになりますので、両方のクロックドメインを跨ぐすべての信号は再同期されず (図 391 の SYNC ブロック)。

受信ストリーム S/PDIF を適切にデコードするために、SPDIFRX\_DC は最大シンボルレートより最低でも 11 倍高速、またはオーディオサンプリングレートより 704 倍高速のクロックで受信データを再サンプリングします。たとえば、ユーザが最大 12.288 MHz のシンボルレートを受信したい場合、サンプリングレートは最低でも 135.2 MHz となります。SPDIFRX\_DC が使用するクロックは、SPDIFRX\_CLK です。

PCLK1 の周波数はシンボルレート以上でなければなりません。

表 188. SPDIFRX\_CLK 最小周波数とオーディオサンプリングレート

シンボルレート	SPDIFRX_CLK 最小周波数	コメント
3.072 MHz	33.8 MHz	48 kHz ストリーム用
6.144 MHz	67.6 MHz	96 kHz ストリーム用
12.288 MHz	135.2 MHz	192 kHz ストリーム用

## 34.3.10 DMA インタフェース

SPDIFRX インタフェースは、DMA を使用して通信できます。

**注 :** *DMA コントローラの有効性については製品仕様を参照してください。*

SPDIFRX は 2 つの独立した DMA チャンネルを提供します。

- データ転送専用の DMA チャンネル
- チャンネルステータスおよびユーザデータの転送専用の DMA チャンネル

データ受信用の DMA モードを有効にするには、SPDIFRX\_CR レジスタの RXDMAEN ビットをセットします。この場合、SPDIFRX\_DR がエンプティでなくなるとすぐに、SPDIFRX インタフェースは DMA に対して転送リクエストを送信します。DMA は、CPU の介入なしに SPDIFRX\_DR レジスタによって受信したデータを読み出します。

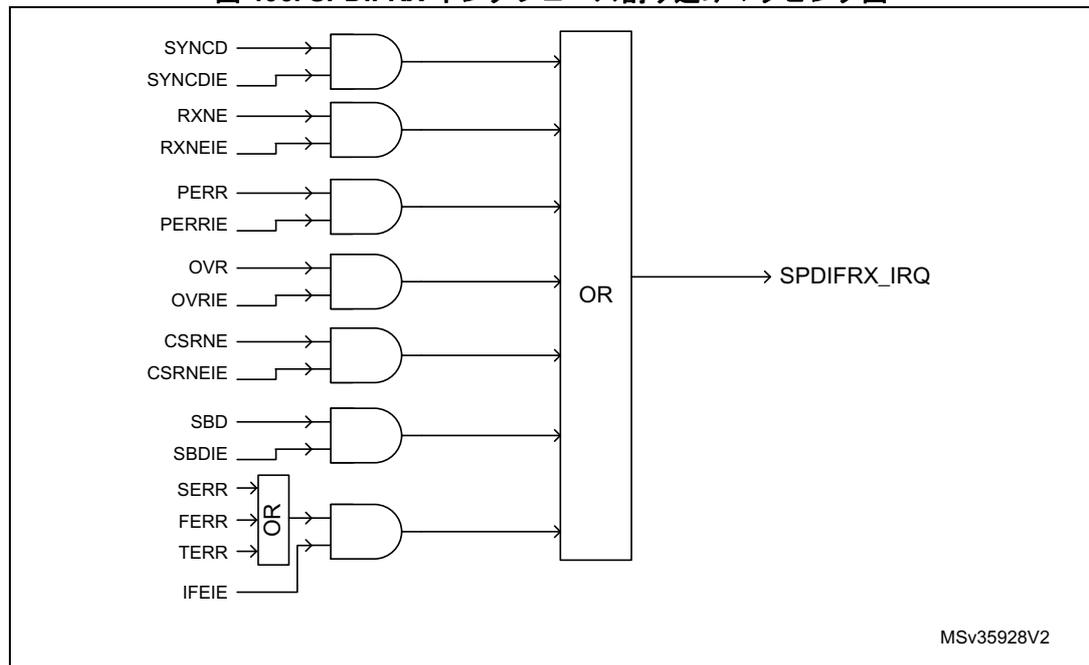
制御データの DMA の使用については、[セクション 34.3.7 : 専用制御フロー](#)を参照してください。

## 34.3.11 割り込みの生成

割り込みラインは次の間で共有されます。

- データフローの受信イベント (RXNE)
- 制御フローの受信イベント (CSRNE)
- データ破損検出 (PERR)
- 転送フロー割り込み (OVR)
- フレーム構造と同期エラー (SERR、TERR、FERR)
- 新しいブロック割り込みの開始 (SBD)
- 同期完了 (SYNCD)

図 406. SPDIFRX インタフェース割り込みマッピング図



### 割り込みソースのクリア

- RXNE は、SPDIFRX\_DR レジスタが読み出されたときにクリアされます。
- CSRNE は、SPDIFRX\_CSR レジスタが読み出されたときにクリアされます。
- FERR は、SPDIFRXEN が 0 にセットされたときにクリアされます。
- SERR は、SPDIFRXEN が 0 にセットされたときにクリアされます。
- TERR は、SPDIFRXEN が 0 にセットされたときにクリアされます。
- それ以外は SPDIFRX\_IFCR レジスタによってクリアされます。

注： SBD イベントは、SPDIFRX が入カストリームに同期される場合のみ起こり得ます。(SYNCD = 1)

プリアンプル B を含むサブフレームがオーバーランにより失われた場合、SBD フラグの動作は保証されません。

## 34.3.12 レジスタの保護

SPDIFRX ブロックは、制御レジスタの誤使用を回避するために、何らかのハードウェア保護を内蔵しています。下表に、SPDIFRX の状態に応じたビットフィールドの特性を示します。

表 189. ビットフィールドの特性と SPDIFRX の状態

レジスタ	フィールド	SPDIFRXEN		
		0b00 (STATE_IDLE)	0b01 (STATE_SYNC)	0b11 (STATE_RCV)
SPDIFRX_CR	INSEL	rw	r	r
	WFA	rw	r	r
	NBTR	rw	r	r
	CHSEL	rw	r	r
	CBDMAEN	rw	rw	rw
	PTMSK	rw	rw	rw
	CUMSK	rw	rw	rw
	VMSK	rw	rw	rw
	PMSK	rw	rw	rw
	DRFMT	rw	rw	r
	RXSTEO	rw	rw	r
	RXDMAEN	rw	rw	rw
SPDIFRX_IMR	すべてのフィールド	rw	rw	rw

この表を見ると、IP が STATE\_IDLE にある時に、INSEL などのフィールドを設定する必要があることが明らかです。他の IP 状態では、ハードウェアはこのフィールドへの書き込みを阻止します。

**注：** ハードウェアが CBDMAEN および RXDMAEN への「動作中の」書き込みを許可する場合でも、IP がすでにデータを受信しているときに DMA を有効化することはお勧めしません。

**注：** 各マスクビット (PMSK、VMSK、...) は IP がどんな状態でも動作中に変更することができますが、変更が SPDIFRX\_DR に保持されているデータに影響を及ぼすことはありません。

## 34.4 プログラミング手順

次の例は、SPDIFRX ブロックのアクティブ化の完全な手順を示します。データパスおよびチャンネルステータスおよびユーザ情報はどちらも専用の DMA チャンネルを使用します。アクティブ化のシーケンスは次のステップに分割されます。

- 選択された SPDIFRX\_IN 入力で有効データを待ちます。
- S/PDIF ストリームに同期させます。
- チャンネルステータスとユーザ情報を読み出して、完全なオーディオパスセットアップします。
- データ取得を開始します。

SPDIFRX\_IN ラインに使用できる有効なデータがあるかどうかを確認する簡単な方法は、WFA ビットを 1 にセットしたまま、SPDIFRX を STATE\_SYNC に切り替えることです。これ以降は、検出方法についての説明になります。この機能を次のように実装することも可能です。

- SPDIFRX が同期を検出できるかどうかをソフトウェアで時々チェックする必要があります (たとえば、100 ミリ秒に 1 回)。チェックとは、TERR ビットがセットされているかどうかのチェックです。セットされていれば、アクティビティは 1 つも検出されていないことが分かります。
- SPDIFRX\_IN ラインの遷移を検出するために、SPDIFRX\_IN 入力を外部割込みイベントブロックに接続します。アクティビティが検出された場合、SPDIFRXEN を 0b01 または 0b11 にセットすることができます。

このような 2 種類の実装では、WFA ビットを 0 にセットします。

### 34.4.1 初期化フェーズ

初期化機能について以下に説明します。

- オーディオサンプル、および IEC60958 チャンネルステータスとユーザ情報 (DMA チャンネルの選択とアクティブ化、優先順位、データ転送回数、サーキュラモード/ノンサーキュラモード、DMA 割り込み) の両方について、DMA 転送を設定します。
- 転送先アドレスを設定します。
  - SPDIFRX\_CSR レジスタのアドレスを、IEC60958 チャンネルステータスおよびユーザ情報のソースアドレスとして設定します。
  - SPDIFRX\_DR レジスタのアドレスを、オーディオサンプルのソースアドレスとして設定します。
- SPDIFRX\_CLK の生成を有効にします。最小クロック周波数とサポートされているオーディオサンプリングレートを定義するには、表 188 を参照してください。受信したストリームのオーディオサンプリングレートを前もって知ることはできません。つまり、ユーザは SPDIFRX\_CLK 周波数として、アプリケーションで処理するはずの最大オーディオサンプリングレートよりも最低でも 704 倍高速の周波数を選択する必要があります。たとえば、アプリケーションで最高 96 kHz のストリームを処理するのであれば、 $F_{SPDIFRX\_CLK}$  は最低でも  $704 \times 96 \text{ kHz} = 67.6 \text{ MHz}$  でなければなりません。
- エラーおよびイベントシグナリングの割り込みを有効にします (IFEIE、SYNCDIE、OVRIE、PERRIE を 1 に、その他を 0 にセットします)。なお、SYNCDIE は 0 にセットすることができます。

- SPDIFRX\_CR レジスタを次のように設定します。
  - INSEL には希望の入力を選択する。
  - NBTR = 2、WFA = 1 (16 回のリトライが可能。アクティビティを待ってから、同期フェーズに進むこと。)
  - PTMSK = CUMSK = 1 (プリアンブル、C ビット、U ビットにはデータを混合させないこと。)
  - VMSK = PMSK = 0 (パリティエラーおよび有効性ビットにはデータを有効にすること。)
  - CHSEL = 0 (チャンネルステータスはサブフレーム A から読み出される。)
  - DRFMT = 0b01 (データ左詰め)
  - RXSTEO = 1 (ステレオモードリニアが预期される。)
  - CBDMAEN = RXDMAEN = 1 (DMA チャンネルを有効にする。)
  - SPDIFRXEN = 0b01 (SPDIFRX を STATE\_SYNC に切り替える。)
- CPU は WFI モードに入ることができます。

その後、CPU は DMA または SPDIFRX からの割り込みを受信します。

## 34.4.2 SPDIFRX からの割り込みの処理

SPDIFRX からの割り込みが受信された場合、ソフトウェアによって、SPDIFRX\_SR レジスタを読み出すことで、割り込みの転送元を確認する必要があります。

- SYNCNCD が 1 にセットされていれば、同期が問題なく完了したことを意味します。この場合、DMA はすでにプログラムされているケースでは、いかなるソフトウェアアクションも必要ありません。ソフトウェアに必要なのは、チャンネルステータス情報を読み出すために、DMA 割り込みを待つことだけです。SPDIFRX\_IFCR レジスタの SYNCNDCF ビットを 1 にセットすることで、SYNCNCD フラグをクリアする必要があります。
- TERR、SERR、FERR のいずれかが 1 にセットされている場合は、ソフトウェアで SPDIFRXEN を 0 にセットし、初期化フェーズから再開する必要があります。
  - TERR は、初期化フェーズ中またはフェーズ後のタイムアウトの発生を示します。
  - SERR は、最大許容リトライ回数に達したことによる同期の失敗を示します。
  - FERR は、同期失敗後の情報の読み出しを示します (预期せぬプリアンブル、データのデコード不良など)。
- PERR が 1 にセットされている場合は、パリティエラーが検出され、それにより、受信したオーディオサンプル、チャンネルステータス、ユーザデータのいずれかのビットが破損したことを意味します。ここで必要な操作はアプリケーションによって異なります。たとえば、現在のチャンネルステータスブロックが信頼できないとして、ドロップすることもあります。同期は失われていないので、初期化フェーズから再開する必要はありません。SPDIFRX\_IFCR レジスタの PERRCF ビットを 1 にセットすることで、PERR フラグをクリアする必要があります。

## 34.4.3 DMA からの割り込みの処理

チャンネルステータス (SPDIFRX\_CSR) に使用される DMA チャンネルから割り込みを受信する場合

エラーが発生していない場合 (PERR)、CPU はチャンネル情報のデコードを開始できます。たとえば、チャンネルステータスのビット 1 は現在のストリームがリニアか否かをユーザに通知します。この情報は、適切な処理チェーンをセットアップする上でとても重要です。同様に、チャンネルステータスのビット 24~27 は受信ストリームのサンプリング周波数を提供します。

このような情報のおかげで、ユーザはデータ受信を開始する前に RXSTEO ビットおよび DRFMT フィールドを設定することができます。たとえば、現在のストリームがリニア PCM でない場合、RXSTEO は 0 にセットされ、DRFMT は 0b10 にセットされます。その場合、ユーザは SPDIFRXEN を 0b11 にセットすることで、データ受信を有効にできます。

SOB ビットが 1 にセットされている場合は新しいブロックの始めを意味します。この情報により、ソフトウェアによるチャンネルステータスのビット 0 の識別が容易になります。24 個の値がメモリに転送されるたびに DMA が割り込みを生成する場合、その先頭ワードは常に新しいブロックの始めに相当することに注意してください。

オーディオサンプル (SPDIFRX\_DR) に使用される DMA チャンネルから割り込みを受信する場合

実行するプロセスはデータタイプ (リニアまたはノンリニア)、および選択されたデータフォーマットによって異なります。

たとえば、リニアモードでは、PE または V ビットがセットされている場合、ローカルに特殊な処理を実施して、出力でのスプールを回避することができます。ノンリニアモードでは、データフレームに独自のチェックサムがあるので、これらのビットは重要ではありません。

## 34.5 SPDIFRX インタフェースレジスタ

### 34.5.1 制御レジスタ (SPDIFRX\_CR)

アドレスオフセット : 0x00

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSEL <sup>(1)</sup>		
													rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WFA <sup>(1)</sup>	NBTR[1:0] <sup>(1)</sup>		CHSEL <sup>(1)</sup>	CBDMAEN <sup>(1)</sup>	PTMSK <sup>(1)</sup>	CUMSK <sup>(1)</sup>	VMSK <sup>(1)</sup>	PMSK <sup>(1)</sup>	DRFMT <sup>(1)</sup>		RXSTEO <sup>(1)</sup>	RXDMAEN <sup>(1)</sup>	SPDIFRXEN[1:0] <sup>(1)</sup>	
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

1. フィールドの特性の詳細については、[セクション 34.3.12 : レジスタの保護](#)を参照してください。

ビット 31:19 予約済み、ハードウェアによって 0 に固定されています。

ビット 18:16 **INSEL** : SPDIFRX 割り込みの選択

- 0b000 : SPDIFRX\_IN1 が選択されます。
- 0b001 : SPDIFRX\_IN2 が選択されます。
- 0b010 : SPDIFRX\_IN3 が選択されます。
- 0b011 : SPDIFRX\_IN4 が選択されます。
- その他 : 予約済み

ビット 15 予約済み、ハードウェアによって 0 に固定されています。

ビット 14 **WFA** : アクティビティ待ち

- このビットは、ソフトウェアでセット/クリアされます。
- 1 : SPDIFRX は、SPDIFRX\_IN ラインでのアクティビティ (4 個の遷移) を待ってから同期を実行します。
- 0 : SPDIFRX は、SPDIFRX\_IN ラインでのアクティビティを待たずに同期を実行します。

ビット 13:12 **NBTR** : 同期フェーズ中の最大許容リトライ回数

- 0b00 : リトライは許可されません (1 回の試行のみ)。
- 0b01 : 3 回のリトライが許可されます。
- 0b10 : 15 回のリトライが許可されます。
- 0b11 : 63 回のリトライが許可されます。

ビット 11 **CHSEL** : チャネルの選択

- このビットは、ソフトウェアでセット/クリアされます。
- 1 : 制御フローはチャネル B からチャネルステータスを取得します。
- 0 : 制御フローはチャネル A からチャネルステータスを取得します。

ビット 10 **CBDMAEN** : 制御フロー用の制御バッファ DMA 有効化

- このビットは、ソフトウェアでセット/クリアされます。
- 1 : チャネルステータスおよびユーザデータ情報の受信に対して、DMA モードが有効です。
- 0 : チャネルステータスおよびユーザデータ情報の受信に対して、DMA モードが無効です。

このビットがセットされると、CSRNE フラグがセットされるたびに DMA リクエストが行われます。

- ビット 9      **PTMSK** : プリマブルタイプビットのマスク  
このビットは、ソフトウェアでセット/クリアされます。  
1 : プリマブルタイプビットは SPDIFRX\_DR にコピーされず、代わりにゼロが書き込まれます。  
0 : プリマブルタイプビットが SPDIFRX\_DR にコピーされます。
- ビット 8      **CUMSK** : チャネルステータスビットおよびユーザビットのマスク  
このビットは、ソフトウェアでセット/クリアされます。  
1 : チャネルステータスビットおよびユーザビットは SPDIFRX\_DR にコピーされず、代わりにゼロが書き込まれます。  
0 : チャネルステータスビットおよびユーザビットが SPDIFRX\_DR にコピーされます。
- ビット 7      **VMSK** : 有効性ビットのマスク  
このビットは、ソフトウェアでセット/クリアされます。  
1 : 有効性ビットは SPDIFRX\_DR にコピーされず、代わりにゼロが書き込まれます。  
0 : 有効性ビットが SPDIFRX\_DR にコピーされます。
- ビット 6      **PMSK** : パリティエラービットのマスク  
このビットは、ソフトウェアでセット/クリアされます。  
1 : パリティエラービットは SPDIFRX\_DR にコピーされず、代わりにゼロが書き込まれます。  
0 : パリティエラービットが SPDIFRX\_DR にコピーされます。
- ビット 5:4    **DRFMT** : RX データフォーマット  
このビットは、ソフトウェアでセット/クリアされます。  
0b11 : 予約済み  
0b10 : データサンプルは、2 個の 16 ビットサンプルを 32 ビットワードにセットすることでパックされます。  
0b01 : データサンプルは左詰めに配置されます (MSB)。  
0b00 : データサンプルは右詰めに配置されます (LSB)。
- ビット 3      **RXSTEO** : ステレオモード  
このビットは、ソフトウェアでセット/クリアされます。  
1 : ペリフェラルはステレオモードにあります。  
0 : ペリフェラルはモノラルモードにあります。  
このビットは、オーバーラン状況が発生した場合にミスアラインメントを処理するためにも使われます。
- ビット 2      **RXDMAEN** : データフロー用レシーバ DMA 有効化  
このビットは、ソフトウェアでセット/クリアされます。  
1 : DMA モードが受信に有効です。  
0 : DMA モードが受信に無効です。  
このビットがセットされると、RXNE フラグがセットされるたびに DMA リクエストが行われます。
- ビット 1:0    **SPDIFRXEN** : ペリフェラルブロック有効化  
このフィールドはソフトウェアで変更されます。  
このフィールドは、次の 3 つの予想される状態のうち、ペリフェラルフェーズを変更するために使用されます。STATE\_IDLE、STATE\_SYNC、および STATE\_RCV。  
  
0b00 : SPDIFRX (STATE\_IDLE) を無効にします。  
0b01 : SPDIFRX 同期のみを有効にします。  
0b10 : 予約済み  
0b11 : SPDIF レシーバを有効にします。  
  
**注:**      1    **STATE\_RCV から STATE\_SYNC への遷移はできません、まず STATE\_IDLE へ進んでください。**  
            2    **STATE\_IDLE から STATE\_RCV への遷移は可能です、ペリフェラルは STATE\_IDLE から STATE\_SYNC へ遷移し、同期が行われるとすぐに STATE\_RCV へ進みます。**

## 34.5.2 割り込みマスクレジスタ (SPDIFRX\_IMR)

アドレスオフセット : 0x04

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	IFEIE	SYNCDIE	SBLKIE	OVRIE	PERRIE	CSRNEIE	RXNEIE									
										rw	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済み、ハードウェアによって 0 に固定されています。

ビット 6 **IFEIE** : シリアルインタフェースエラー割り込み有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの SERR = 1、TERR = 1 または FERR = 1 のときには、SPDIFRX 割り込みが生成されます。

ビット 5 **SYNCDIE** : 同期完了  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの SYNCD = 1 のときには、SPDIFRX インタフェース割り込みが生成されます。

ビット 4 **SBLKIE** : 同期ブロック検出割り込み有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの SBD = 1 のときには、SPDIFRX インタフェース割り込みが生成されます。

ビット 3 **OVRIE** : オーバーランエラー割り込み有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの OVR = 1 のときには、SPDIFRX インタフェース割り込みが生成されます。

ビット 2 **PERRIE** : パリティエラー割り込み有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの PERR = 1 のときには、SPDIFRX インタフェース割り込みが生成されます。

ビット 1 **CSRNEIE** : 制御バッファレディ割り込み有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの CSRNE = 1 のときには、SPDIFRX インタフェース割り込みが生成されます。

ビット 0 **RXNEIE** : RXNE 割り込み有効化  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : 割り込みは禁止されています。  
 1 : SPDIFRX\_SR レジスタの RXNE = 1 のときには、SPDIFRX インタフェース割り込みが生成されます。

## 34.5.3 ステータスレジスタ (SPDIFRX\_SR)

アドレスオフセット : 0x08

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	WIDTH5[14:0]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR	SERR	FERR	SYNCD	SBD	OVR	PERR	CSRNE	RXNE
							r	r	r	r	r	r	r	r	r

ビット 31 予約済み、ハードウェアによって 0 に固定されています。

ビット 30:16 **WIDTH5** : SPDIFRX\_CLK でカウントされた 5 個のシンボルの継続時間

この値は、5 個の連続したシンボルの全長に含まれる SPDIFRX\_CLK クロック周期の数を表します。この値を使用して、S/PDIF シンボルレートを概算できます。その精度は SPDIFRX\_CLK の周波数によって制限されます。

たとえば、SPDIFRX\_CLK が 84 MHz に固定されており、かつ WIDTH5 = 147d の場合、S/PDIF ストリームの概算サンプリングレートは、

$F_s = 5 * F_{SPDIFRX\_CLK} / (WIDTH5 * 64) \sim 44.6 \text{ kHz}$  となり、したがって最も近い基準サンプリングレートは 44.1 kHz となります。

WIDTH5 は SYNCD がハイになったときおよびフレームごとに、ハードウェアによって更新されます。

ビット 15:9 予約済み、ハードウェアによって 0 に固定されています。

ビット 8 **TERR** : タイムアウトエラー

このビットは、カウンタ TRCNT がその最大値に達したときにハードウェアによってセットされます。このビットは 2 個の遷移の間隔が長すぎることを示します。また、ほとんどの場合 SPDIFRX\_IN 入力に有効な信号がないことを意味します。

このフラグは、SPDIFRXEN に 0 を書き込むことによってクリアされます。

SPDIFRX\_IMR レジスタの IFEIE = 1 である場合、割り込みが生成されます。

0 : シーケンスエラーは検出されていません。

1 : シーケンスエラーが検出されました。

ビット 7 **SERR** : 同期エラー

このビットは、同期の失敗が NBTR でセットしたりトライ数に達した時、ハードウェアによってセットされます。

このフラグは、SPDIFRXEN に 0 を書き込むことによってクリアされます。

SPDIFRX\_IMR レジスタの IFEIE = 1 である場合、割り込みが生成されます。

0 : 同期エラーは検出されていません。

1 : 同期エラーが検出されました。

ビット 6 **FERR** : フレーミングエラー

このビットは、データ受信中にエラーが発生した場合にハードウェアによってセットされます (予想した場所にプリアンプルがない、短い遷移はペアでグループ化されない、など)。

このビットは、同期が完了している場合にのみハードウェアによってセットされます。(SYNCD = 1)

このフラグは、SPDIFRXEN に 0 を書き込むことによってクリアされます。

SPDIFRX\_IMR レジスタの IFEIE = 1 である場合、割り込みが生成されます。

0 : マンチェスタプロトコル違反は検出されていません。

1 : マンチェスタプロトコル違反が検出されました。



- ビット 5      **SYNCD** : 同期完了
- このビットは、最初の同期フェーズが問題なく完了したときに、ハードウェアによってセットされます。  
このフラグは、SPDIFRX\_CLR\_SR レジスタの対応するビットに 1 を書き込むことによってクリアされます。  
SPDIFRX\_IMR レジスタの SYNCDIE = 1 である場合、割り込みが生成されます。  
0 : 同期は完了していません。  
1 : 同期が完了しました。
- ビット 4      **SBD** : 同期ブロック検出
- このビットは、プリアンブル B が検出されたときに、ハードウェアによってセットされます。  
このフラグは、SPDIFRX\_CLR\_SR レジスタの対応するビットに 1 を書き込むことによってクリアされます。  
SPDIFRX\_IMR レジスタの SBLKIE = 1 である場合、割り込みが生成されます。  
0 : プリアンブル B は検出されていません。  
1 : プリアンブル B が検出されました。
- ビット 3      **OVR** : オーバーランエラー
- このビットはRXNE=1で、SPDIFRX\_DRレジスタへ受信データの転送準備ができたとき、かつSPDIFRX\_DR とRX\_BUFがフルの場合、ハードウェアによってセットされます。  
このフラグは、SPDIFRX\_CLR\_SR レジスタの対応するビットに 1 を書き込むことによってクリアされます。  
SPDIFRX\_IMR レジスタの OVRIE = 1 である場合、割り込みが生成されます。  
0 : オーバーランエラーはありません。  
1 : オーバーランエラーが検出されました。
- 注 :**      *このビットがセットされている場合、SPDIFRX\_DR レジスタの内容は失われませんが、最後に受信したデータは失われます。*
- ビット 2      **PERR** : パリティエラー
- このビットは、受信したサブフレームのデータおよびステータスビットに奇数個の“0”および“1”が含まれるときに、ハードウェアによってセットされます。  
このフラグは、SPDIFRX\_CLR\_SR レジスタの対応するビットに 1 を書き込むことによってクリアされます。  
SPDIFRX\_IMR レジスタの PIE = 1 である場合、割り込みが生成されます。  
0 : パリティエラーはありません。  
1 : パリティエラー
- ビット 1      **CSRNE** : 制御バッファレジスタはエンプティではありません。
- このビットは、有効な制御情報が準備されているときに、ハードウェアによってセットされます。  
このフラグは、SPDIFRX\_CSR レジスタを読み出すときにクリアされます。  
SPDIFRX\_IMR レジスタの CBRDYIE = 1 である場合、割り込みが生成されます。  
0 : SPDIFRX\_CSR レジスタには使用可能な制御ワードはありません。  
1 : SPDIFRX\_CSR レジスタには使用可能な制御ワードがあります。
- ビット 0      **RXNE** : 読み出しデータレジスタノットエンプティ
- このビットは、SPDIFRX\_DR レジスタに使用可能な有効データがあるときに、ハードウェアによってセットされます。  
このフラグは、SPDIFRX\_DR レジスタの読み出しによってクリアされます。  
SPDIFRX\_IMR レジスタの RXNEIE = 1 である場合、割り込みが生成されます。  
0 : データは受信されていません。  
1 : 受信データを読み出すことができます。

## 34.5.4 割り込みフラグクリアレジスタ (SPDIFRX\_IFCR)

アドレスオフセット : 0x0C

リセット値 : 0x00000000

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

ビット 31:6 予約済み、ハードウェアによって 0 に固定されています。

ビット 5 **SYNDCDF** : 同期完了フラグのクリア  
 このビットに 1 を書き込むと、SPDIFRX\_SR レジスタの SYNCD フラグがクリアされます。  
 このビットを読み出すと、常に 0 値が返されます。

ビット 4 **SBDCF** : 同期ブロック検出フラグのクリア  
 このビットに 1 を書き込むと、SPDIFRX\_SR レジスタの SBD フラグがクリアされます。  
 このビットを読み出すと、常に 0 値が返されます。

ビット 3 **OVRCF** : オーバーランエラーフラグのクリア  
 このビットに 1 を書き込むと、SPDIFRX\_SR レジスタの OVR フラグがクリアされます。  
 このビットを読み出すと、常に 0 値が返されます。

ビット 2 **PERRCF** : パリティエラーフラグのクリア  
 このビットに 1 を書き込むと、SPDIFRX\_SR レジスタの PERR フラグがクリアされます。  
 このビットを読み出すと、常に 0 値が返されます。

ビット 1:0 予約済み



## 34.5.5 データ入力レジスタ (SPDIFRX\_DR)

アドレスオフセット : 0x10

リセット値 : 0x00000000

このレジスタは DRFMT に応じて 3 通りのフォーマットを取ることができます。DRFMT = 0b00 の場

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

合のフォーマットを以下に示します。

ビット 31:30 予約済み : ハードウェアによって 0 に固定されています。

ビット 29:28 **PT** : プリアンブルタイプ

これらのビットは受信したプリアンブルを示します。

00 : 未使用

01 : プリアンブル B を受信しました。

10 : プリアンブル M を受信しました。

11 : プリアンブル W を受信しました。

PTMSK = 1 の場合、このフィールドはゼロに固定されますので、ご注意ください。

ビット 27 **C** : チャネルステータスビット

CUMSK = 0 の場合、受信したチャネルステータスビットを表示します。その他の場合は 0 に固定されます。

ビット 26 **U** : ユーザビット

CUMSK = 0 の場合、受信したユーザビットを表示します。その他の場合は 0 に固定されます。

ビット 25 **V** : 有効性ビット

VMSK = 0 の場合、受信した有効性ビットを表示します。その他の場合は 0 に固定されます。

ビット 24 **PE** : パリティエラービット

PMSK = 0 の場合、PERR ビットのコピーを表示します。その他の場合は 0 に固定されます。

ビット 23:0 **DR** : データ値

24 個の受信データビット (D[23] に整列) を表示します。

## 34.5.6 データ入力レジスタ (SPDIFRX\_DR)

アドレスオフセット : 0x10

リセット値 : 0x00000000

このレジスタは DRFMT に応じて 3 通りのフォーマットを取ることができます。DRFMT = 0b01 の場合のフォーマットを以下に示します。

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

- ビット 31:8 **DR** : データ値  
24 個の受信データビット (D[23] に整列) を表示します。
- ビット 7:6 予約済み : ハードウェアによって 0 に固定されています。
- ビット 5:4 **PT** : プリアンブルタイプ  
これらのビットは受信したプリアンブルを示します。  
00 : 未使用  
01 : プリアンブル B を受信しました。  
10 : プリアンブル M を受信しました。  
11 : プリアンブル W を受信しました。  
PTMSK = 1 の場合、このフィールドはゼロに固定されますので、ご注意ください。
- ビット 3 **C** : チャネルステータスビット  
CUMSK = 0 の場合、受信したチャネルステータスビットを表示します。その他の場合は 0 に固定されます。
- ビット 2 **U** : ユーザビット  
CUMSK = 0 の場合、受信したユーザビットを表示します。その他の場合は 0 に固定されます。
- ビット 1 **V** : 有効性ビット  
VMSK = 0 の場合、受信した有効性ビットを表示します。その他の場合は 0 に固定されます。
- ビット 0 **PE** : パリティエラービット  
PMSK = 0 の場合、PERR ビットのコピーを表示します。その他の場合は 0 に固定されます。

## 34.5.7 データ入力レジスタ (SPDIFRX\_DR)

アドレスオフセット : 0x10

リセット値 : 0x00000000

このレジスタは DRFMT に応じて 3 通りのフォーマットを取ることができます。

DRFMT = 0b10 の場合に提示されたデータフォーマットは 16 ビットしか使用されないため (S/SPDIF サブフレームのビット 23~8)、ノンリニアモード専用です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRNL2[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRNL1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **DRNL2** : データ値  
このフィールドはチャンネル A を含みます。

ビット 15:0 **DRNL1** : データ値  
このフィールドはチャンネル B を含みます。

## 34.5.8 チャンルステータスレジスタ (SPDIFRX\_CSR)

アドレスオフセット : 0x14

リセット値 : 0x00000000

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

ビット 31:25 予約済み

ビット 24 **SOB** : ブロックの開始

このビットは、CS[0] ビットが新しいブロックの先頭ビットに対応するかどうかを示します。

0 : CS[0] は新しいブロックの先頭ビットではありません。

1 : CS[0] は新しいブロックの先頭ビットです。

ビット 23:16 **CS[7:0]** : チャンネル A ステータス情報

CS[0] ビットは一番古い値です。

ビット 15:0 **USR[15:0]** : ユーザデータ情報

USR[0] ビットは一番古い値で、チャンネル A から受信します。USR[1] はチャンネル B から受信します。

したがって、USR[n] ビットの"n"は偶数であり、そうでない場合はチャンネル B から受信したことになります。

## 34.5.9 デバッグ情報レジスタ (SPDIFRX\_DIR)

アドレスオフセット : 0x18

リセット値 : 0x00000000

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

ビット 31:29 予約済み、ハードウェアによって 0 に固定されています。

ビット 16:28 **TLO** : 閾値ロー (TLO = 1.5 \* UI / T<sub>SPDIFRX\_CLK</sub>)  
 このフィールドには現在の閾値ローの概算値が含まれています。この値を使用して、受信ストリームのサンプリングレートを概算することができます。TLO の精度は SPDIFRX\_CLK の 1 周期に制限されます。サンプリングレートは次のように算出することができます。  

$$\text{サンプリングレート} = [2 * \text{TLO} * T_{\text{SPDIFRX\_CLK}} \pm T_{\text{SPDIFRX\_CLK}}] * 2/3$$
  
 TLO は SYNCDC がハイになったときおよびフレームごとに、ハードウェアによって更新されます。

ビット 15:13 予約済み、ハードウェアによって 0 に固定されています。

ビット 12:0 **THI** : 閾値ハイ (THI = 2.5 \* UI / T<sub>SPDIFRX\_CLK</sub>)  
 このフィールドには現在の閾値ハイの概算値が含まれています。この値を使用して、受信ストリームのサンプリングレートを概算することができます。THI の精度は SPDIFRX\_CLK の 1 周期に制限されます。サンプリングレートは次のように算出することができます。  

$$\text{サンプリングレート} = [2 * \text{THI} * T_{\text{SPDIFRX\_CLK}} \pm T_{\text{SPDIFRX\_CLK}}] * 2/5$$
  
 THI は SYNCDC がハイになったときおよびフレームごとに、ハードウェアによって更新されます。

## 34.5.10 SPDIFRX インタフェースレジスタマップ

次の表に、SPDIFRX インタフェースレジスタマップとリセット値を示します。

表 190. SPDIFRX インタフェースレジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	SPDIFRX_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSEL[2:0]	Res.	Res.	WFA	NBTR	CHSEL	CBDMAEN	PTMSK	CUMSK	VMSK	PMSK	DRFMT[1:0]	DRFMT[1:0]	RXSTEO	RXDMAEN	SPDIFRXEN						
	リセット値														0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	SPDIFRX_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEIE	SYNCDIE	SBLKIE	OVRIE	PERRIE	CSRNEIE	RXNEIE				
	リセット値																									0	0	0	0	0	0	0	0			
0x08	SPDIFRX_SR	Res.	WIDTH5[14:0]														Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR	SERR	FERR	SYNCD	SBD	OVR	PERR	CSRNE	RXNE
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0			
0x0C	SPDIFRX_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
	リセット値																																			
0x10	SPDIFRX_DR	Res.	Res.	PT[1:0]	C	U	V	PE	DR[23:0]																											
		DR[23:0]																			Res.	PT[1:0]	C	U	V	PE										
		DRNL2[15:0]															DRNL1[15:0]																			
リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x14	SPDIFRX_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOB	CS[7:0]							USR[15:0]																			
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x18	SPDIFRX_DIR	Res.	Res.	Res.	TLO[12:0]											予約済み						THI[12:0]														
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			



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

## 35 SD/SDIO/MMC カードホストインタフェース (SDMMC)

### 35.1 SDMMC の主な特長

SD/SDIO MMC カードホストインタフェース (SDMMC) は、APB2 ペリフェラルバスと MultiMediaCard (MMC)、SD メモリカード、および SDIO カード間のインタフェースを提供します。

MMCA 技術委員会から発行された MultiMediaCard システム仕様書は、JEDECのウェブサイトから入手できます。

SD メモリカードと SD I/O カードシステム仕様書は、SD アソシエーションのウェブサイトから入手できます。

SDMMC には以下の機能があります。

- MultiMediaCard System Specification Version 4.2 に完全準拠。3 種類の異なるデータバスモードのカードに対応：1 ビット (デフォルト)、4 ビットおよび 8 ビット
- 以前のバージョンの MultiMediaCard との完全互換 (上位互換性)
- SD Memory Card Specifications Version 2.0 に完全準拠
- SD I/O Card Specification Version 2.0 完全準拠：2 種類の異なるデータバスモードのカードに対応：1 ビット (デフォルト) および 4 ビット
- 8 ビットモードで最大 48 MHz のデータ転送
- 外部双方向ドライバを制御するデータおよびコマンド出力有効信号

注： 1 SDMMC は SPI 互換の通信モードを備えていません。

2 SD メモリカードプロトコルは、MultiMediaCard system specification V2.11 で定義されている MultiMediaCard プロトコルのスーパーセットです。SD メモリデバイスに必要なコマンドの中には、SD I/O 専用カードやコンボカードの I/O 部分ではサポートされないものもあります。消去コマンドなど、SD I/O デバイスで使用されないコマンドは、SDIO プロトコルでサポートされません。また、コマンドの中には SD メモリカードと SD I/O カードで異なるものがあり、このようなコマンドは SDIO プロトコルでサポートされません。詳細については、SD I/O card Specification Version 1.0 を参照してください。

MultiMediaCard/SD バスは、カードをコントローラに接続します。

現行バージョンの SDMMC は、1 度に 1 枚の SD/SDIO/MMC 4.2 カードと、MMC4.1 以前のスタックだけをサポートします。

### 35.2 SDMMC バストポロジ

バス上の通信は、コマンドとデータ転送に基づいて行われます。

MultiMediaCard/SD/SD I/O バス上の基本的なトランザクションは、コマンド/レスポンストランザクションです。これらの種類のバストランザクションでは、コマンドやレスポンスの決められた通信フォーマットでカードの情報を直接転送します。また、データトークンを持つ操作もあります。

SD/SDIO メモリカードに対するデータ転送は、データブロック単位で行われます。MMC に対するデータ転送は、データブロックまたはストリーム単位で行われます。

図 407. 「no response」と「no data」操作

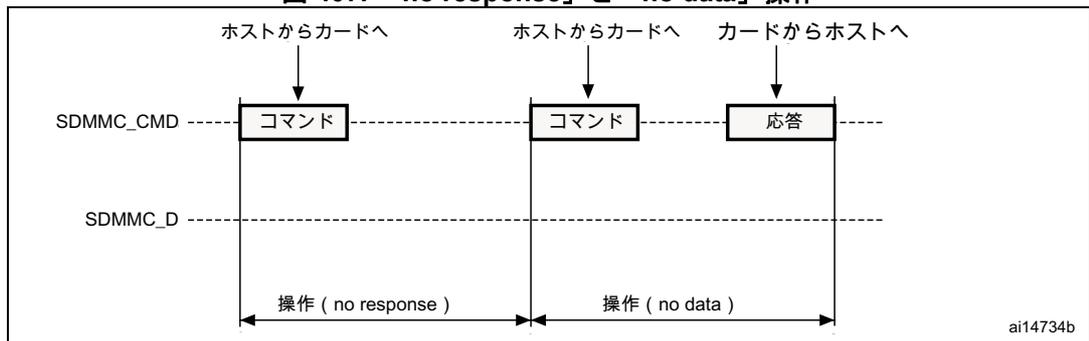


図 408. (複数) ブロック読み出し操作

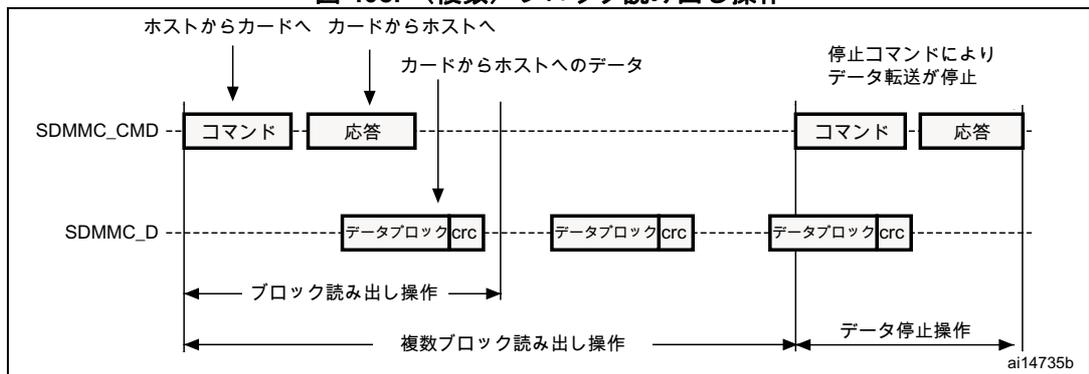
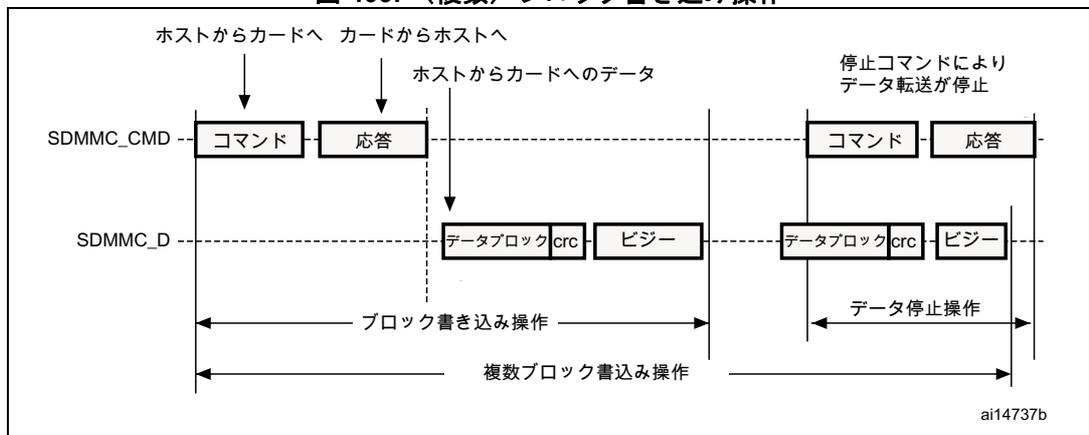


図 409. (複数) ブロック書き込み操作



注: SDMMC は、ビジー信号がアサートされている限り (SDMMC\_D0 をローへ固定)、データを一切送信しません。

図 410. 連続読み出し操作

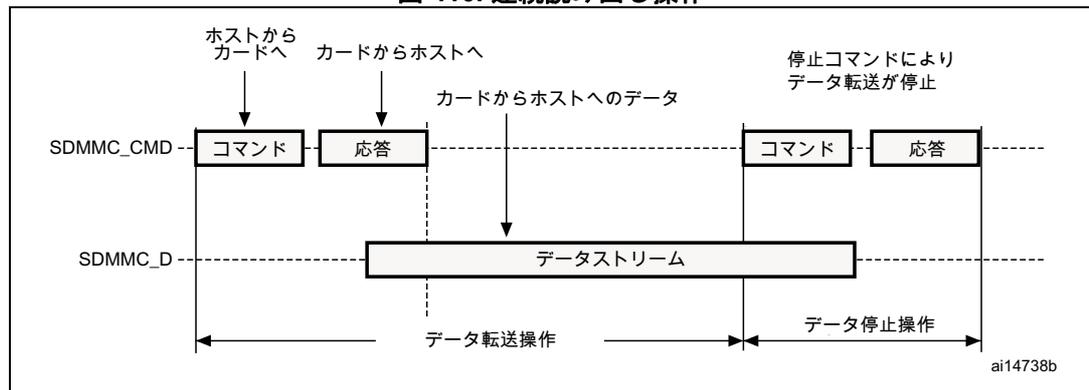
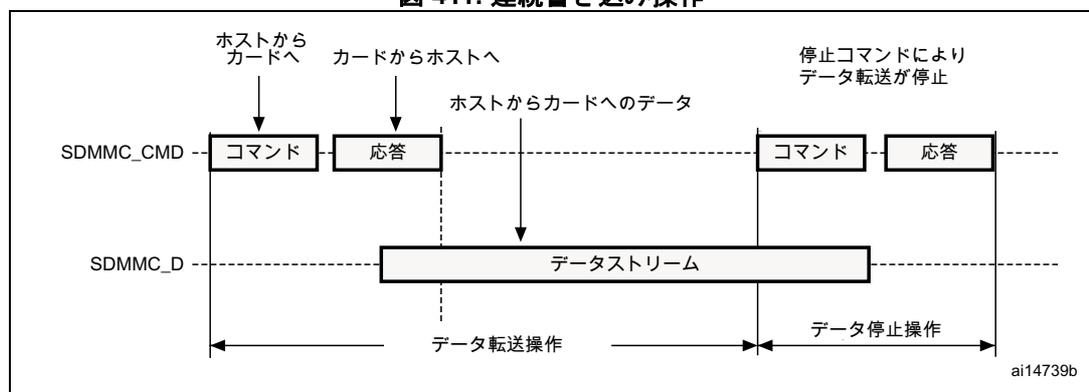


図 411. 連続書き込み操作

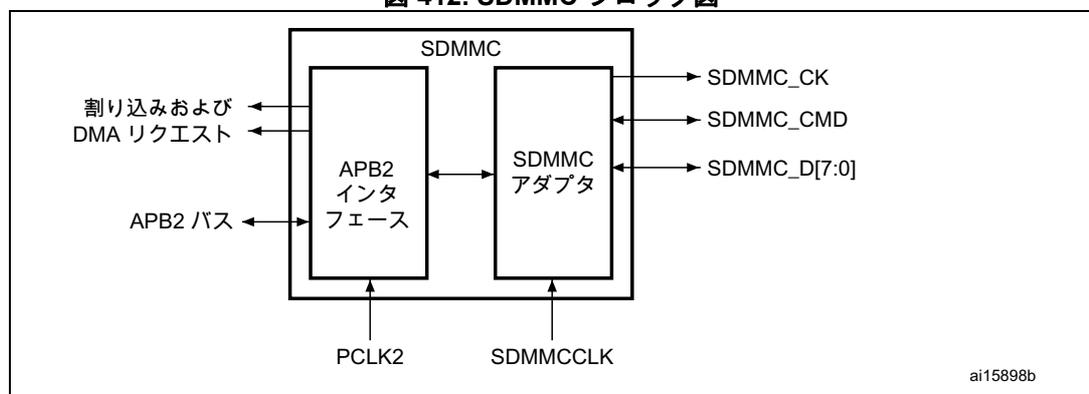


## 35.3 SDMMC機能詳細

SDMMC は、2 つの部分で構成されています。

- SDMMC アダプタブロックは、クロック生成ユニット、コマンド、およびデータ転送など、MMC/SD/SD I/O カードに固有のすべての機能を提供します。
- APB2 インタフェースは、SDMMC アダプタレジスタにアクセスして、割り込みおよび DMA リクエスト信号を生成します。

図 412. SDMMC ブロック図



デフォルトでは、SDMMC\_D0 はデータ転送に使用されます。初期化後、ホストはデータバス幅を変更することができます。

MultiMediaCard がバスに接続されている場合、SDMMC\_D0、SDMMC\_D[3:0]、または SDMMC\_D[7:0] はデータ転送に使用できます。MMC V3.31 以前では、1 ビットのデータのみがサポートされるため、使用できるのは SDMMC\_D0 だけです。

SD または SD I/O カードがバスに接続されている場合、ホストはデータ転送に SDMMC\_D0 または SDMMC\_D[3:0] を使用するよう設定できます。すべてのデータラインはプッシュプルモードで動作します。

**SDMMC\_CMD** には、2 つの動作モードがあります。

- 初期化の場合は、オープンドレイン (MMCV3.31 以前)
- コマンド転送の場合は、プッシュプル (SD/SD I/O カード MMC4.2 は、初期化にもプッシュプルドライバを使用)

**SDMMC\_CK** は、カードに供給されるクロックです。各クロックサイクルで、コマンドラインとデータラインの両方に 1 ビットが転送されます。クロック周波数は、MultiMediaCard V3.31 の場合は 0 MHz から 20 MHz、MultiMediaCard V4.0/4.2 の場合は 0 から 48 MHz、SD/SD I/O カードの場合は 0 から 25 MHz の間で変化します。

SDMMC は 2 つのクロック信号を使用します。

- SDMMC アダプタクロック (SDMMCCLK = 48 MHz)
- APB2 バスクロック (PCLK2)

PCLK2 および SDMMC\_CK クロック周波数は、次の条件を満足している必要があります。

$$\text{Frequenc(PCLK2)} > ((3 \times \text{Width}) / 32) \times \text{Frequency(SDMMC\_CK)}$$

表 191 に示す信号は、MultiMediaCard/SD/SD I/O カードバス上で使用されます。

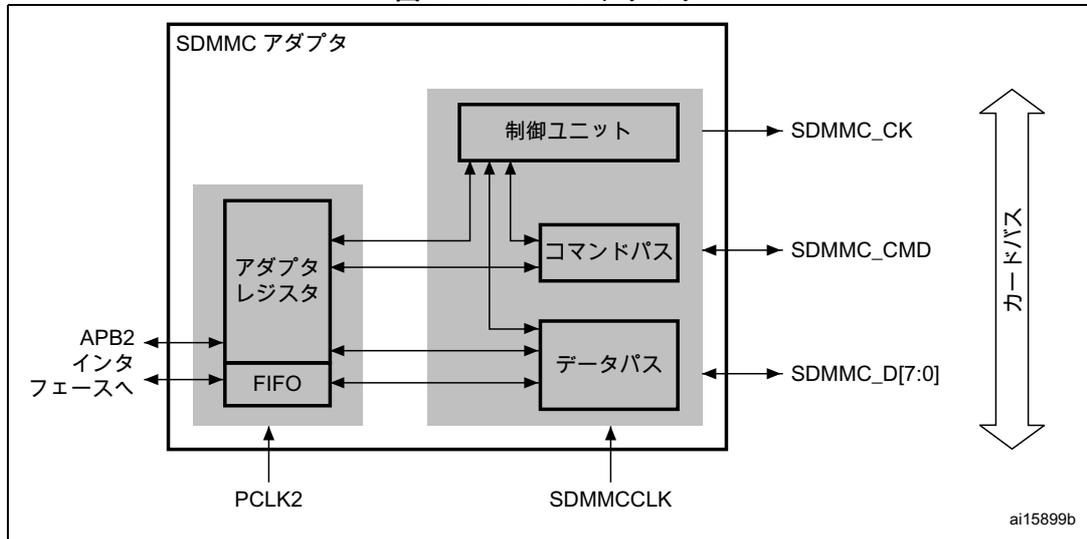
表 191. SDMMC I/O 定義

ピン	方向	説明
SDMMC_CK	出力	MultiMediaCard/SD/SDIO カードクロック。このピンは、ホストからカードへのクロックです。
SDMMC_CMD	双方向	MultiMediaCard/SD/SDIO カードコマンド。このピンは、双方向のコマンド/レスポンス信号です。
SDMMC_D[7:0]	双方向	MultiMediaCard/SD/SDIO カードデータ。これらのピンは、双方向データバスです。

## 35.3.1 SDMMC アダプタ

図 413 は、SDMMC アダプタの簡略化されたブロック図です。

図 413. SDMMC アダプタ



SDMMC アダプタは、マルチメディアカードスタックやセキュアデジタルメモ리카ードとのインタフェースを提供するマルチメディア/セキュアデジタルメモ리카ードバスマスタです。5つのサブユニットで構成されています。

- アダプタレジスタブロック
- 制御ユニット
- コマンドバス
- データバス
- データ FIFO

**注：** アダプタレジスタと FIFO は、APB2 バックロクロドメイン (PCLK2) を使用します。制御ユニット、コマンドバス、およびデータバスは、SDMMC アダプタクロクロドメイン (SDMMCCLK) を使用します。

### アダプタレジスタブロック

アダプタレジスタブロックは、すべてのシステムレジスタを含んでいます。このブロックは、マルチメディアカードのスタティックフラグをクリアする信号も生成します。クリア信号は、SDMMC クリアレジスタの対応するビット位置に 1 が書き込まれると生成されます。

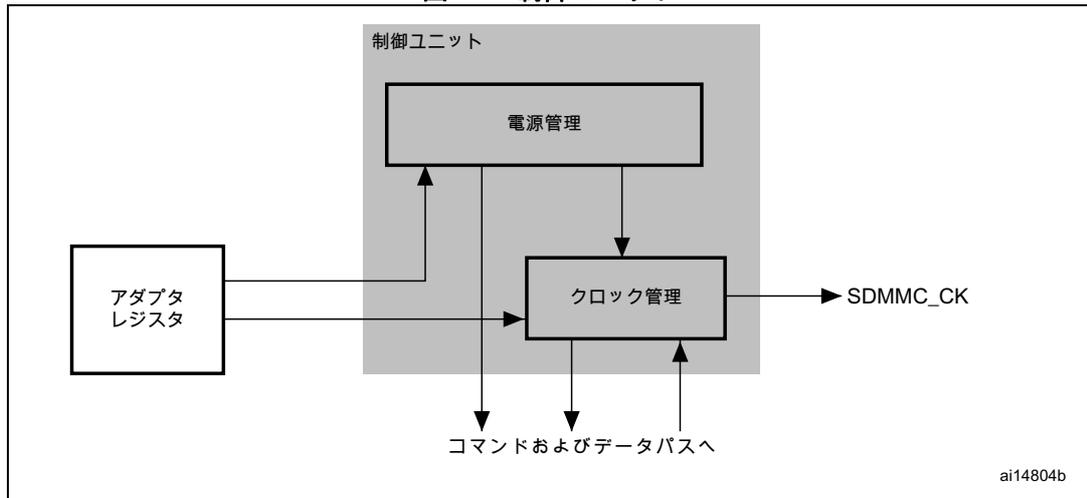
### 制御ユニット

制御ユニットには、電源管理機能とメモ리카ードクロックのクロック分周器が含まれています。

3つの電源フェーズがあります。

- パワーオフ
- パワーアップ
- パワーオン

図 414. 制御ユニット



制御ユニットは、[図 414](#) に示されます。電源管理サブユニットとクロック管理サブユニットで構成されています。

電源管理サブユニットは、パワーオフおよびパワーアップフェーズの間、カードバス出力信号を無効にします。

クロック管理サブユニットは、SDMMC\_CK 信号を生成し、制御します。SDMMC\_CK 出力は、クロック分周またはクロックバイパスモードを使用することができます。クロック出力は、次の場合にインアクティブです。

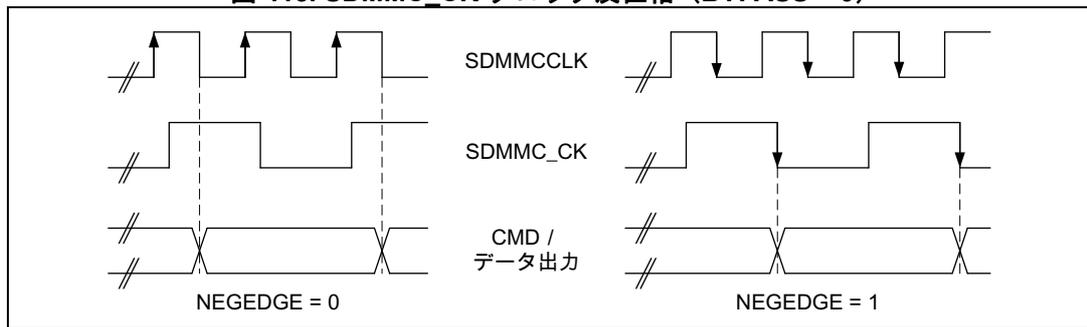
- リセット後
- パワーオフまたはパワーアップフェーズ中
- 節電モードが有効であり、カードバスがアイドル状態の場合（コマンドおよびデータバスサブユニットがアイドルフェーズに入ってから 8 クロック周期後）

クロック管理サブユニットは、SDMMC\_CK位相をずらす制御をします。バイパスモードでない場合、SDMMC コマンドとデータ出力は SDMMC\_CK の立ち上がりエッジの次の SDMMC CLK 立ち下がりエッジで生成されます。SDMMC\_CK 立ち上がりエッジは、SDMMC\_CLKCR[13] ビットがリセット (NEGEDGE = 0) された場合に SDMMCCLK 立ち上がりエッジで発生します。SDMMC\_CLKCR[13] ビットがセットされると (NEGEDGE = 1)、SDMMC\_CK 立ち下がりエッジで SDMMC コマンドとデータが変更されます。

SDMMC\_CLKCR[10] がセットされると (BYPASS = 1)、SDMMCCLK 立ち上がりエッジで SDMMC\_CK 立ち上がりエッジが発生します。NEGEDGE 値にかかわらず、SDMMCCLK 立ち下がりエッジでデータとコマンドが変更されます。

データおよびコマンドレスポンスは、SDMMC\_CK 立ち上がりエッジを使用してラッチされます。

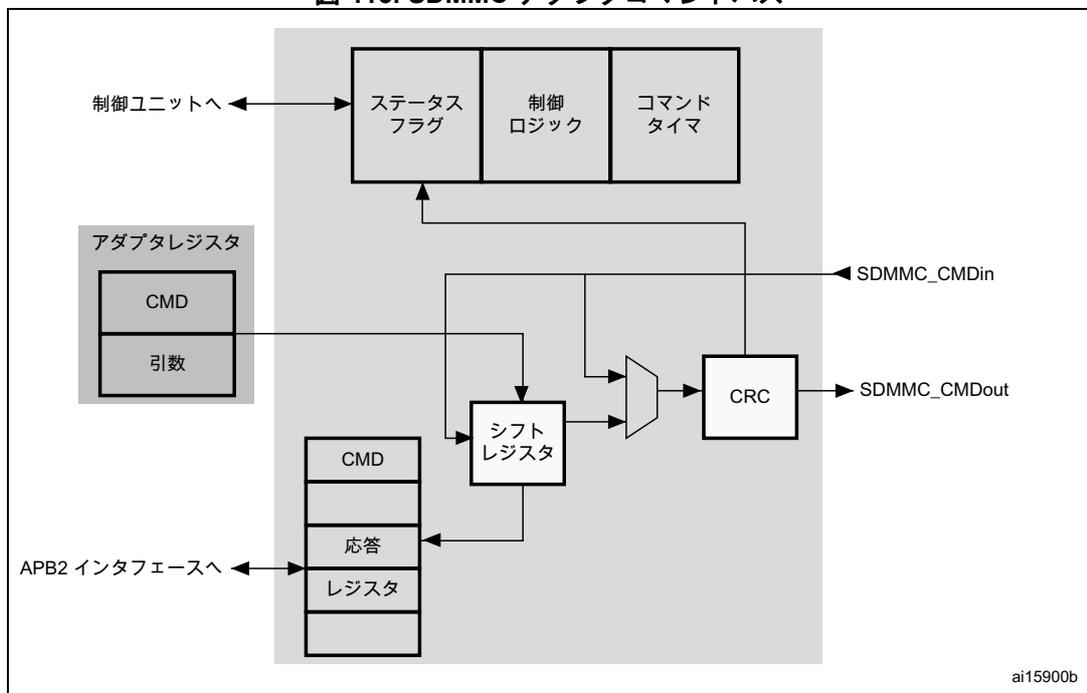
図 415. SDMMC\_CK クロック反位相 (BYPASS = 0)



## コマンドパス

コマンドパスユニットは、カードにコマンドを送信し、カードからレスポンスを受信します。

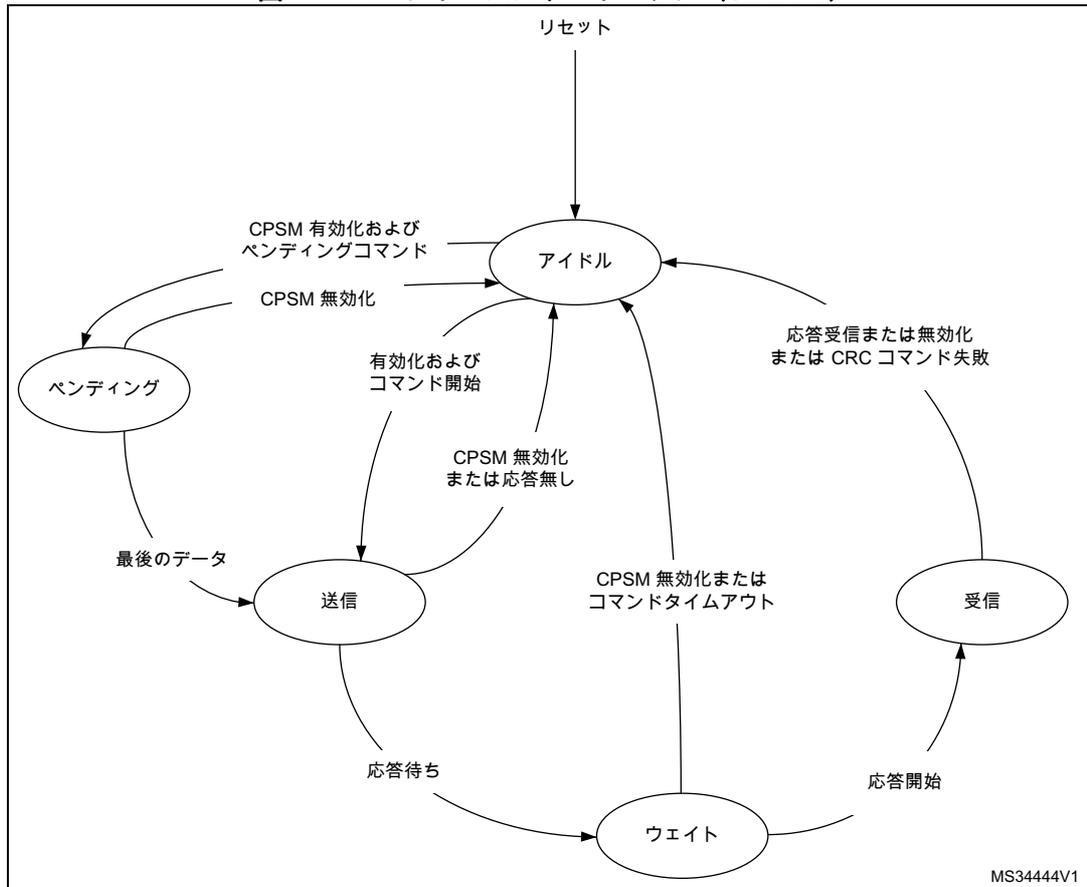
図 416. SDMMC アダプタコマンドパス



- コマンドパスステートマシン (CPSM)

- コマンドレジスタが書き込まれて、有効ビットがセットされると、コマンド転送が開始します。コマンドが送信されると、コマンドパスステートマシン (CPSM) は、ステータスフラグをセットし、レスポンスが不要な場合はアイドル状態になります。レスポンスが必要な場合は、レスポンスを待ちます (図 417 (1197 ページ) を参照)。レスポンスを受信されると、受信した CRC コードと内部生成されたコードを比較して適切なステータスフラグをセットします。

図 417. コマンドパスステートマシン (SDMMC)



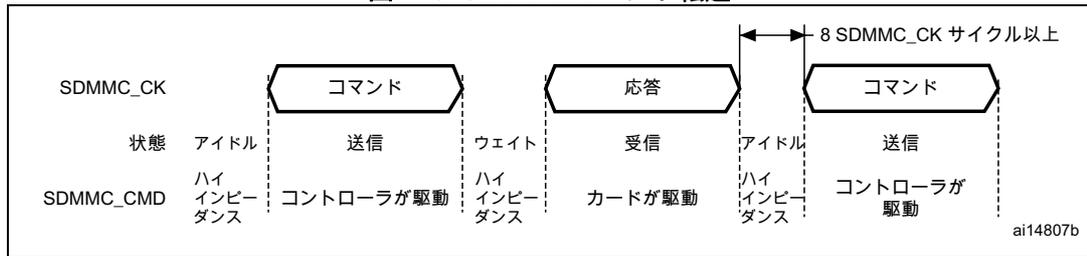
ウェイト状態に入ると、コマンドタイムが作動します。CPSM が受信状態になる前にタイムアウトに達した場合には、タイムアウトフラグがセットされ、アイドル状態に入ります。

**注：** コマンドタイムアウトは、64 SDMMC\_CK クロック周期の固定値です。

コマンドレジスタの割込みビットがセットされると、タイムは無効になり、CPSM はカードの 1 枚からの割込みリクエストを待ちます。コマンドレジスタのペンディングビットがセットされると、CPSM はペンディング状態に入り、データパスサブユニットからの CmdPend 信号を待ちます。CmdPend が検出されると、CPSM は送信状態に移行します。これによって、停止コマンドの送信をトリガするデータカウンタが有効になります。

**注：** CPSM は、 $N_{CC}$  および  $N_{RC}$  タイミング制約を満たすために、少なくとも 8 SDMMC\_CK 周期の間、アイドル状態を保ちます。 $N_{CC}$  は、2 つのホストコマンド間の最小遅延であり、 $N_{RC}$  は、ホストコマンドとカードレスポンスの間の最小遅延です。

図 418. SDMMC コマンド転送



● コマンドフォーマット

- コマンド: コマンドは、動作を開始するトークンです。コマンドは、ホストから 1 枚のカード (アドレス付きコマンド) または接続されたすべてのカードに送信されます (ブロードキャストコマンドは MMCV3.31 以前で使用可能)。コマンドは、CMD ライン上で連続的に転送されます。すべてのコマンドは、固定長 48 ビットです。MultiMediaCard、SD メモリカード、および SDIO カードのコマンドトークンの一般的なフォーマットを表 192 に示します。

コマンドパスは、コマンドとレスポンスを送受信できるように半二重モードで動作します。CPSM が送信状態でない場合、SDMMC\_CMD 出力は図 418 (1198 ページ) に示すようにハイインピーダンス状態になります。SDMMC\_CMD のデータは、SDMMC\_CK の立ち上がりエッジと同期します。表 192 に、コマンドフォーマットを示します。

表 192. コマンドフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	1	送信ビット
[45:40]	6	-	コマンドインデックス
[39:8]	32	-	引数
[7:1]	7	-	CRC7
0	1	1	終了ビット

- レスポンス: レスポンスは、以前に受信したコマンドへの応答として、アドレス指定されたカードから (MMC V3.31 以前の場合はすべてのカードから同期して) ホストに送信されるトークンです。レスポンスは、CMD ライン上で連続的に転送されます。

SDMMC は、2 種類のレスポンスをサポートします。どちらも CRC エラーチェックを使用します。

- 48 ビットショートレスポンス
- 136 ビットロングレスポンス

注: レスポンスが CRC を含まない場合 (CMD1 レスポンス)、デバイスドライバは CRC 障害ステータスを無視する必要があります。

表 193. ショートレスポンスフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	-	コマンドインデックス
[39:8]	32	-	引数
[7:1]	7	-	CRC7 (または 1111111)
0	1	1	終了ビット

表 194. ロングレスポンスフォーマット

ビット位置	幅	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	111111	予約済み
[127:1]	127	-	CID または CSD (内部 CRC7 を含む)
0	1	1	終了ビット

コマンドレジスタは、コマンドインデックス（カードに送信される 6 ビット）とコマンドタイプを含みます。これらによって、コマンドがレスポンスを必要とするかどうかと、レスポンスが 48 ビット長か 136 ビット長かが決まります（[セクション 35.8.4 \(1235 ページ\)](#) を参照）。コマンドパスは、[表 195](#) に示すようにステータスフラグを実装します。

表 195. コマンドパスステータスフラグ

フラグ	説明
CMDREND	レスポンス CRC が OK の場合にセットされます。
CCRCFAIL	レスポンス CRC が失敗した場合にセットされます。
CMDSENT	コマンド（レスポンスを必要としない）が送信されたときにセットされます。
CTIMEOUT	レスポンスタイムアウトです。
CMDACT	コマンド転送中です。

CRC 生成回路は、CRC コードの前のすべてのビットについて CRC チェックサムを計算します。これには、スタートビット、送信ビット、コマンドインデックス、およびコマンド引数（またはカードステータス）が含まれます。CRC チェックサムは、ロングレスポンスフォーマットの場合、CID または CSD の最初の 120 ビットについて計算されます。スタートビット、送信ビット、および 6 つの予約ビットは、CRC 計算では使用されませんので注意してください。

CRC チェックサムは 7 ビットの値です。

$$\text{CRC}[6:0] = \text{剰余} [(M(x) * x^7) / G(x)]$$

$$G(x) = x^7 + x^3 + 1$$

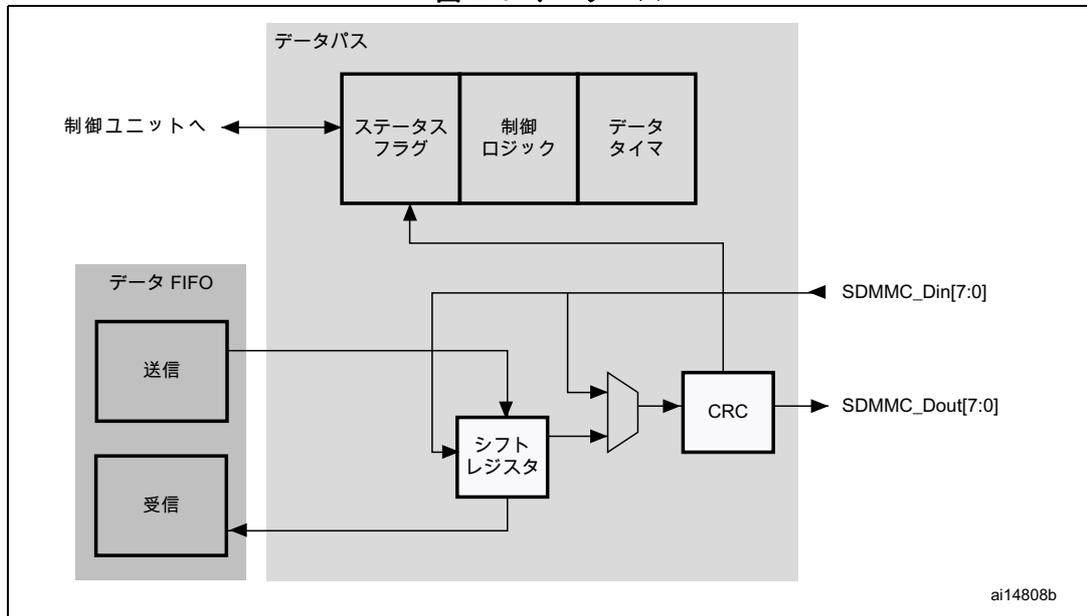
$$M(x) = (\text{スタートビット}) * x^{39} + \dots + (\text{CRC 前の最後のビット}) * x^0, \text{または}$$

$$M(x) = (\text{スタートビット}) * x^{119} + \dots + (\text{CRC 前の最後のビット}) * x^0$$

## データバス

データバスサブユニットは、カードとの間でデータ転送を行います。図 419 に、データバスのブロック図を示します。

図 419. データバス



カードデータバス幅は、クロック制御レジスタを使用してプログラムできます。4 ビット幅バスモードが有効な場合、データは 4 つのデータ信号すべて (SDMMC\_D[3:0]) で、クロックサイクルごとに 4 ビットで転送されます。8 ビット幅バスモードが有効な場合、データは 8 つのデータ信号すべて (SDMMC\_D[7:0]) で、クロックサイクルごとに 8 ビットで転送されます。ワイドバスモードが有効でない場合は、SDMMC\_D0 上でクロックサイクルごとに 1 ビットだけが転送されます。

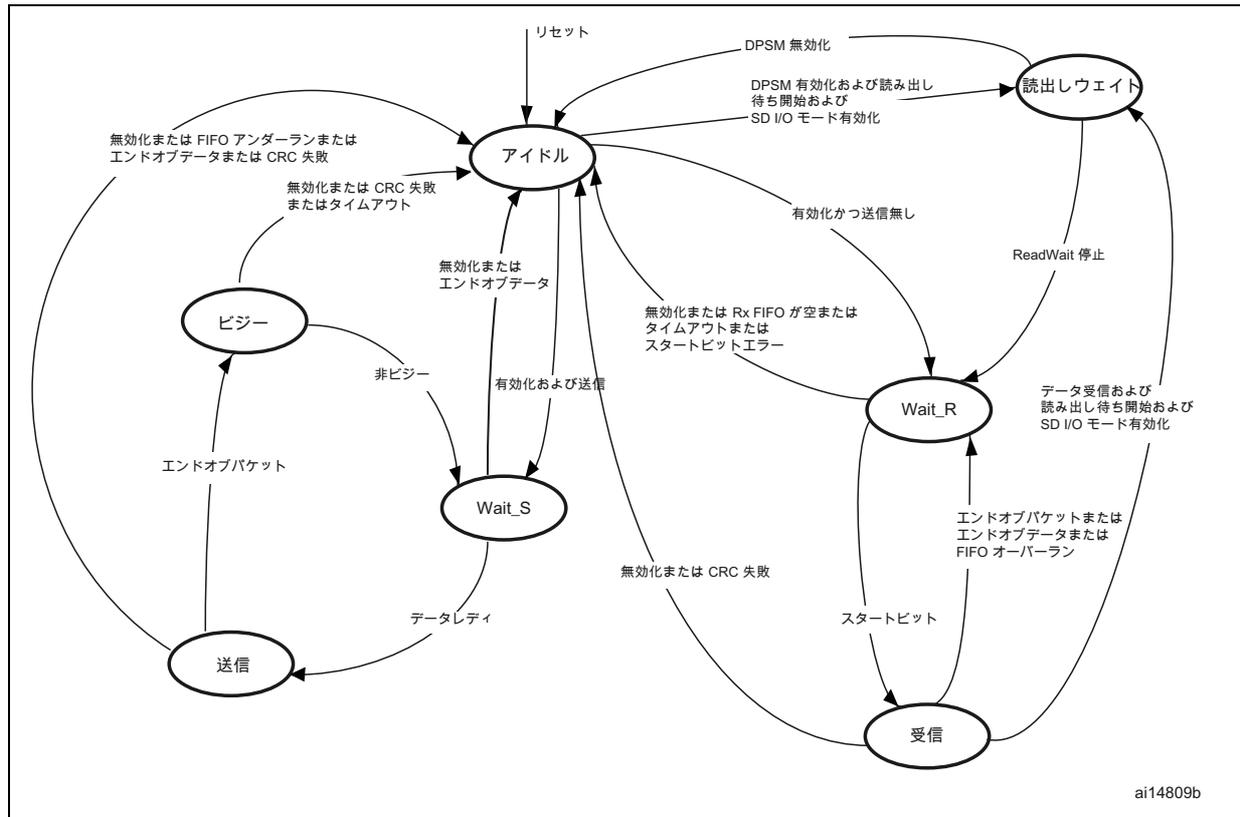
転送の方向 (送信または受信) によって、データバスステートマシン (DPSM) は、有効になったときに Wait\_S または Wait\_R 状態へ移行します。

- 送信 : DPSM は Wait\_S 状態へ移行します。送信 FIFO にデータがある場合、DPSM は送信状態へ移行し、データバスサブユニットはカードへのデータ転送を開始します。
- 受信 : DPSM は Wait\_R 状態へ移行して、スタートビットを待ちます。スタートビットを受信すると、DPSM は受信状態へ移行し、データバスサブユニットはカードからのデータ受信を開始します。

### データバスステートマシン (DPSM)

DPSM は、SDMMC\_CK 周波数で動作します。カードバス信号上のデータは、SDMMC\_CK の立ち上がりエッジと同期しています。図 420 : データバスステートマシン (DPSM) に示すように、DPSM には、6 つの異なる状態があります。

図 420. データパスステートマシン (DPSM)



- **アイドル**：データパスはインアクティブであり、SDMMC\_D[7:0] 出力はハイインピーダンスです。データ制御レジスタが書き込まれ、有効ビットがセットされると、DPSM はデータカウンタに新しい値をロードし、データ方向ビットに応じて Wait\_S または Wait\_R 状態へ移行します。
  - **Wait\_R**：データカウンタがゼロに等しい場合、DPSM は受信 FIFO が空になると、アイドル状態へ移行します。データカウンタがゼロでない場合、DPSM は SDMMC\_D 上のスタートビットを待ちます。タイムアウト前にスタートビットを受信した場合、DPSM は受信状態に移行し、データブロックカウンタをロードします。スタートビットを検出する前にタイムアウトに達した場合、アイドル状態へ移行して、タイムアウトステータスフラグをセットします。
  - **受信**：カードから受信したシリアルデータは数バイトにまとめられ、データ FIFO に書き込まれます。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
    - ブロックモードでは、データブロックカウンタがゼロに達すると、DPSM は CRC コードを受信するまで待ちます。受信したコードが内部生成された CRC コードと一致する場合、DPSM は Wait\_R 状態へ移行します。そうでない場合、CRC 失敗ステータスフラグをセットして、DPSM はアイドル状態へ移行します。
    - ストリームモードでは、DPSM はデータカウンタがゼロでない間、データを受信します。カウンタがゼロになると、シフトレジスタ内の残りのデータがデータ FIFO に書き込まれ、DPSM は Wait\_R 状態へ移行します。
- FIFO オーバーランエラーが発生した場合、DPSM は FIFO エラーフラグをセットして、アイドル状態へ移行します。
- **Wait\_S**：データカウンタがゼロの場合、DPSM はアイドル状態へ移行します。そうでない場合は、データ FIFO エンプティフラグがネゲートされるまで待ってから、送信状態へ移行します。

**注：** DPSM は、 $N_{WR}$  タイミングの要件を満たすため、少なくとも 2 クロック周期の間、Wait\_S 状態を保ちます。ここで、 $N_{WR}$  は、カードレスポンスの受信とホストからのデータ転送の開始との間のクロックサイクル数です。

- 送信：DPSM はカードへのデータの送信を開始します。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
  - ブロックモードでは、データブロックカウンタがゼロに達すると、DPSM は内部生成された CRC コードと終了ビットを送信して、ビジー状態へ移行します。
  - ストリームモードでは、DPSM は、有効ビットがハイであり、データカウンタがゼロでない間、カードへデータを送信します。その後、アイドル状態へ移行します。

FIFO アンダーランエラーが発生した場合、DPSM は FIFO エラーフラグをセットして、アイドル状態へ移行します。

- ビジー：DPSM は、CRC ステータスフラグを待ちます。
  - 良好な CRC ステータスを受信しなかった場合、DPSM はアイドル状態へ移行し、CRC 失敗ステータスフラグをセットします。
  - 良好な CRC ステータスを受信した場合、SDMMC\_D0 がローでない（カードがビジーでない）場合は Wait\_S 状態へ移行します。

DPSM がビジー状態の間にタイムアウトエラーが発生した場合には、DPSM はデータタイムアウトフラグをセットして、アイドル状態へ移行します。

データタイマは、DPSM が Wait\_R またはビジー状態のときに有効になり、以下の場合にデータタイムアウトを生成します。

- データ送信中、DPSM がプログラムされたタイムアウト時間より長く、ビジー状態にとどまっていた場合。
- データ受信時、データの終わりが真でなく、DPSM がプログラムされたタイムアウト時間より長く Wait\_R 状態にとどまっていた場合、タイムアウトが発生します。
- データ：データは、カードからホストへ、またはホストからカードへ転送できます。データは、データラインを介して転送されます。32 ワードの FIFO に格納され、各ワードは 32 ビット幅です。

表 196. データトークンフォーマット

説明	スタートビット	データ	CRC 16	終了ビット
ブロックデータ	0	-	あり	1
ストリームデータ	0	-	なし	1

## DPSM フラグ

データパスサブユニットの転送ステータスはいくつかのステータスフラグによって報告されます。

表 197. DPSM フラグ

フラグ	説明
DBCKEND	データブロックの送受信 CRC チェックに成功すると、ハイにセットされます。SDIO マルチバイト転送モードでは、このフラグは転送終了時にセットされます (ホストでは、マルチバイト転送は単一のブロック転送とみなされます)。
DATAEND	SDMMC_DCOUNT レジスタがデクリメントされ、0 に達するとハイにセットされます。DATAEND は、SDMMC データラインでの転送終了を示します。
DTIMEOUT	データタイムアウト期間に達すると、ハイにセットされます。DPSM が Wait_R またはビジー状態のときにデータタイムは 0 に達すると、タイムアウトがセットされます。DPSM がプログラムされた期間より長くビジー状態にとどまっている場合、DATAEND の後で DTIMEOUT をセットできます。
DCRCFAIL	データブロックの送受信 CRC チェックに失敗すると、ハイにセットされます。

## データ FIFO

データ FIFO (ファーストインファーストアウト) サブユニットは、送受信ユニットを持つデータバッファです。

FIFO には、32 ビット幅、32 ワードの深さのデータバッファと、送受信ロジックがあります。データ FIFO は APB2 クロックドメイン (PCLK2) で動作するため、SDMMC クロックドメイン (SDMMCCLK) のサブユニットからの信号はすべて再同期されます。

TXACT および RXACT フラグに応じて、FIFO は無効、送信有効、または受信有効にできます。TXACT と RXACT は、データパスサブユニットによって駆動され、相互排他的となっています。

- 送信 FIFO は、TXACT がアサートされると、送信ロジックとデータバッファを参照します。
- 受信 FIFO は、RXACT がアサートされると、受信ロジックとデータバッファを参照します。

- 送信 FIFO :

データは、SDMMC が送信有効なときに、APB2 インタフェースを通じて送信 FIFO に書かれます。

送信 FIFO は、32 個の連続アドレスを介してアクセス可能です。送信 FIFO には、読み出しポインタが示すデータワードを保持するデータ出力レジスタが含まれています。データパスサブユニットがシフトレジスタへロードすると、読み出しポインタがインクリメントされ、新しいデータが出力されます。

送信 FIFO が無効な場合、すべてのステータスフラグがネゲートされます。データパスサブユニットは、データを送信するときに TXACT をアサートします。

表 198. 送信 FIFO ステータスフラグ

フラグ	説明
TXFIFO	32 個すべての送信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
TXFIFOE	送信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
TXFIFOHE	8 個以上の送信 FIFO ワードが空のとき、ハイにセットされます。このフラグは DMA リクエストとして使用できます。
TXDAVL	送信 FIFO が有効なデータを含んでいるとき、ハイにセットされます。このフラグは、TXFIFIOE フラグの反転です。
TXUNDERR	アンダーランエラーが発生すると、ハイにセットされます。このフラグは、SDMMC クリアレジスタへの書き込みによってクリアされます。 <b>注:</b> TXUNDERR の場合で SDMMC FIFO に書き込むために DMA が使用される場合、ユーザソフトウェアでは DMA ストリームを無効にし、SDMMC_DCTRL の DMAEN ビットに '0' を書き込む必要があります (DMA リクエストの生成を無効にするため)。

- 受信 FIFO

データパスサブユニットがワードデータを受信すると、書き込みデータバス上にデータが出力されます。書き込み操作が完了すると、書き込みポインタがインクリメントされます。読み出し側では、読み出しポインタの現在値が示す FIFO ワードの内容が、読み出しバス上に出力されます。受信 FIFO が無効な場合、すべてのステータスフラグがネゲートされ、読み出しおよび書き込みポインタがリセットされます。データパスサブユニットは、データを受信すると、RXACT をアサートします。表 199 に、受信 FIFO ステータスフラグをリストします。受信 FIFO は、32 個の連続アドレスを介してアクセス可能です。

表 199. 受信 FIFO ステータスフラグ

フラグ	説明
RXFIFO	32 個すべての受信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
RXFIFOE	受信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
RXFIFOHF	8 個以上の受信 FIFO ワードが空のとき、ハイにセットされます。このフラグは DMA リクエストとして使用できます。
RXDAVL	受信 FIFO が空でないとき、ハイにセットされます。このフラグは RXFIFIOE フラグの反転です。
RXOVERR	オーバーランエラーが発生すると、ハイにセットされます。このフラグは、SDMMC クリアレジスタへの書き込みによってクリアされます。 <b>注:</b> RXOVERR の場合で SDMMC FIFO を読み出すために DMA が使用される場合、ユーザソフトウェアでは DMA ストリームを無効にし、SDMMC_DCTRL の DMAEN ビットに '0' を書き込む必要があります (DMA リクエストの生成を無効にするため)。

## 35.3.2 SDMMC APB2 インタフェース

APB2 インタフェースは、割り込みおよび DMA リクエストを生成し、SDMMC アダプタレジスタとデータ FIFO にアクセスします。このインタフェースは、データパス、レジスタデコーダ、および割り込み/DMA ロジックで構成されています。

### SDMMC 割り込み

割り込みロジックは、選択されたステータスフラグの少なくとも 1 つがハイのときにアサートされる割り込みリクエスト信号を生成します。割り込み生成条件が選択できるように、マスクレジスタが提供されています。ステータスフラグは、対応するマスクフラグがセットされている場合に割り込みリクエストを生成します。

### SDMMC/DMA インタフェース

SDMMC APB インタフェースは、ホストとカードの間の転送を実行するために、すべてのサブユニットを制御します。

### DMA を使用した読み出し手順の例

次のように、CMD17 (READ\_BLOCK) を送信します。

- a) SDMMC データ長レジスタをプログラムします (SDMMC データタイムレジスタは、カード識別プロセスの前にすでにプログラムされている必要があります)。
- b) DMA チャンネルをプログラムします ([SDMMC コントローラの DMA の設定](#)を参照)。
- c) SDMMC データ制御レジスタをプログラムします。DTEN は“1” (SDMMC カードホストはデータの送信が可能)、DTDIR は“1” (カードからコントローラへ)、DTMODE は“0” (ブロックデータ転送)、DMAEN は“1” (DMA は有効)、DBLOCKSIZ は 0x9 (512 バイト) にプログラムします。他のフィールドは無視します。
- d) SDMMC 引数レジスタに、データが転送されるカードのアドレス位置をプログラムします。
- e) SDMMC コマンドレジスタをプログラムします。CmdIndex は 17 (READ\_BLOCK) に、WaitResp は“1” (SDMMC カードホストはレスポンスを待つ) に、CPSMEN は“1” (SDMMC カードホストはコマンドの送信が可能) にプログラムします。他のフィールドはリセット値です。
- f) SDMMC\_STA[6] = CMDREND 割り込みを待ちます (コマンドパス上にエラーがない場合は CMDREND がセットされる)。
- g) SDMMC\_STA[10] = DBCKEND を待ちます (CRC 確認が成功するまでエラーがない場合は DBCKEND がセットされる)。
- h) FIFO がエンプティになるまで待ち、FIFO がエンプティになったら SDMMC\_STA[5] = RXOVERR の値で読み出しが成功したことを確認してください。

**注：** 最後の 1~4 バイトで FIFO オーバーランエラーが発生した場合は、DATAEND フラグをセットした 2 APB クロックサイクル後に RXOVERR フラグがセットされている可能性があります。読み出し操作を成功させるには、FIFO がエンプティになってから RXOVERR を必ず確認する必要があります。

## DMA を使用した書き込み手順の例

次のように、CMD24 (WRITE\_BLOCK) を送信します。

- a) SDMMC データ長レジスタをプログラムします (SDMMC データタイムレジスタは、カード識別プロセスの前にすでにプログラムされている必要があります)。
- b) DMA チャンネルをプログラムします ([SDMMC コントローラの DMA の設定](#)を参照)。
- c) SDMMC 引数レジスタに、データが転送されるカードのアドレス位置をプログラムします。
- d) SDMMC コマンドレジスタをプログラムします。CmdIndex は 24 (WRITE\_BLOCK) に、WaitResp は“1” (SDMMC カードホストはレスポンスを待つ) に、CPSMEN は“1” (SDMMC カードホストはコマンドの送信が可能) にプログラムします。他のフィールドはリセット値です。
- e) SDMMC\_STA[6] = CMDREND 割り込みを待ち、SDMMC データ制御レジスタを次のようにプログラムします。DTEN は“1” (SDMMC カードホストはデータの送信が可能)、DTDIR は“0” (コントローラからカードへ)、DTMODE は“0” (ブロックデータ転送)、DMAEN は“1” (DMA は有効)、DBLOCKSIZE は 0x9 (512 バイト) にプログラムします。他のフィールドは無視します。
- f) SDMMC\_STA[10] = DBCKEND を待ちます (エラーがない場合は DBCKEND がセットされる)。

## SDMMC コントローラの DMA の設定

- a) DMA2 コントローラを有効にして、ペンディング中の割り込みをすべてクリアします。
- b) DMA2\_Channel4 (or DMA2\_Channel5) 転送元アドレスレジスタをメモリ位置のベースアドレスでプログラムし、DMA2\_Channel4 (or DMA2\_Channel5) 転送先アドレスレジスタを SDMMC\_FIFO レジスタアドレスでプログラムします。
- c) DMA2\_Channel4 (or DMA2\_Channel5) 制御レジスタをプログラムします (ペリフェラルインクリメントではなくメモリインクリメント、またペリフェラル幅とソース幅はワードサイズ)。
- d) DMA2\_Channel4 (or DMA2\_Channel5) をプログラムしてペリフェラルをフローコントローラとして選択します (DMA\_S3CR または DMA\_S6CR 設定レジスタの PFCTRL ビットをセットします)。
- e) DMA2\_Channel4 (or DMA2\_Channel5) で、インクリメンタルバースト転送を 4 ビート (最低でもペリフェラル側から) に設定します。
- f) DMA2\_Channel4 (or DMA2\_Channel5) を有効にします。

**注：** SDMMC ホストでは、ペリフェラルフローコントローラモードでのみ DMA を使用できます。SDMMC で使用する DMA ストリームは、ペリフェラルフローコントローラモードで設定する必要があります。SDMMC では、DMA コントローラに DMA バーストリクエストのみを生成します。DMA は、ペリフェラルサイドでは、インクリメントバーストモードで設定する必要があります。

## 35.4 カード機能詳細

### 35.4.1 カード識別モード

カード識別モードでは、ホストはすべてのカードをリセットし、動作電圧範囲を検証し、カードを識別し、バス上の各カードの相対カードアドレス (RCA) をセットします。カード識別モードでは、すべてのデータ通信にコマンドライン (CMD) のみを使用します。

### 35.4.2 カードリセット

GO\_IDLE\_STATE コマンド (CMD0) は、ソフトウェアリセットコマンドであり、MultiMediaCard と SD メモリをアイドル状態にします。IO\_RW\_DIRECT コマンド (CMD52) は、SD I/O カードをリセットします。パワーアップまたは CMD0 の後、すべてのカード出力バスドライバはハイインピーダンス状態となり、カードはデフォルトの相対カードアドレス (RCA=0x0001) とデフォルトのドライバステージレジスタ設定 (最低速度、最高駆動電流) で初期化されます。

### 35.4.3 動作電圧範囲の検証

すべてのカードは、仕様範囲内の任意の動作電圧で SDMMC カードホストと通信できます。サポートされる最小および最大  $V_{DD}$  値は、カード上の動作条件レジスタ (OCR) で定義されています。

ペイロードメモリにカード識別番号 (CID) とカード固有データ (CSD) を格納するカードは、この情報をデータ転送  $V_{DD}$  条件下でのみ通信することができます。SDMMC カードホストモジュールとカードの  $V_{DD}$  範囲に互換性がないときには、カードは識別サイクルを完了できず、CSD データを送信できません。このため、SDMMC カードホストに必要な  $V_{DD}$  範囲に一致しないカードを識別して拒否するメカニズムとして、特殊なコマンド SEND\_OP\_COND (CMD1)、SD\_APP\_OP\_COND (SD メモリ用 ACMD41)、および IO\_SEND\_OP\_COND (SD I/O 用 CMD5) が用意されています。SDMMC カードホストは、これらのコマンドのオペランドとして、必要な  $V_{DD}$  電圧範囲を送信します。指定された範囲でデータ転送を行えないカードは、バスから切断され、インアクティブ状態になります。

電圧範囲をオペランドとして含めずにこれらのコマンドを使用すると、SDMMC カードホストは各カードへの問い合わせを行い、共通の電圧範囲を判断した後に、範囲外のカードをインアクティブ状態にできます。この問い合わせは、SDMMC カードホストが共通の電圧範囲を選択できるときや、カードが使用不可能であることをユーザに通知する必要があるときに使用されます。

### 35.4.4 カード識別プロセス

カード識別プロセスは、MultiMediaCard と SD カードで異なります。MultiMediaCard では、識別プロセスは  $F_{od}$  クロックレートで開始します。SDMMC\_CMD ライン出力ドライバはオープンドレインであり、このプロセス中は並行カード操作が可能です。登録プロセスは、次のように行われます。

1. バスがアクティベートされます。
2. SDMMC カードホストは、SEND\_OP\_COND (CMD1) をブロードキャストして、動作条件を受信します。
3. レスポンスは、すべてのカードからの動作条件レジスタのワイヤード AND 操作です。
4. 互換性のないカードはインアクティブ状態となります。
5. SDMMC カードホストは、すべてのアクティブカードに ALL\_SEND\_CID (CMD2) を伝達します。
6. アクティブカードは、それぞれの CID 番号をシリアルに同時送信します。発信 CID ビットがコマンドラインのビットと一致しないカードは、転送を停止して、次の識別サイクルを待たなければなりません。接続されたカードがすべての CID を正常に SDMMC カードホストに送信すると、識別状態へ移行します。

- SDMMC カードホストは、そのカードに SET\_RELATIVE\_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードを認識します。これを割り当てられたカードはスタンバイ状態に遷移し、その後の識別サイクルには反応せず、出力はオーブンドレインからプッシュプルへ切り替わります。

- SDMMC カードホストは、タイムアウト条件を受信するまで、ステップ 5~7 を繰り返します。

SD カードの場合、識別プロセスは  $F_{od}$  クロックレートで開始し、SDMMC\_CMD ライン出力ドライバはオーブンドレインでなくプッシュプルドライバです。登録プロセスは、次のように行われます。

- バスがアクティベートされます。
- SDMMC カードホストは SD\_APP\_OP\_COND (ACMD41) をブロードキャストします。
- カードは、それぞれの動作条件レジスタの内容で応答します。
- 互換性のないカードは、インアクティブ状態へ移行されます。
- SDMMC カードホストは、すべてのアクティブカードに ALL\_SEND\_CID (CMD2) をブロードキャストします。
- カードは、一意のカード識別番号 (CID) を送り返し、識別状態に入ります。
- SDMMC カードホストは、アドレスのあるアクティブなカードに SET\_RELATIVE\_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードを認識します。これを割り当てられたカードは、スタンバイ状態に遷移します。SDMMC カードホストは、このコマンドを再発行して RCA を変更することができます。カードの RCA は、最後に割り当てられた値です。
- SDMMC カードホストは、すべてのアクティブなカードについて、ステップ 5~7 を繰り返します。

SD I/O カードの場合、登録プロセスは次のように行われます。

- バスがアクティベートされます。
- SDMMC カードホストは IO\_SEND\_OP\_COND (CMD5) を送信します。
- カードは、それぞれの動作条件レジスタの内容で応答します。
- 互換性のないカードは、インアクティブ状態に設定されます。
- SDMMC カードホストは、アドレスのあるアクティブなカードに SET\_RELATIVE\_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードは、スタンバイ状態に遷移します。SDMMC カードホストは、このコマンドを再発行して RCA を変更することができます。カードの RCA は、最新の値が割り当てられます。

## 35.4.5 ブロック書き込み

ブロック書き込み (CMD24-27) では、1 つ以上のブロックデータがホストからカードに転送され、ホストによって各ブロックの最後に CRC が付加されます。ブロック書き込みをサポートするカードは、WRITE\_BL\_LEN によって定義されたデータブロックを常に受け入れることができます。CRC が失敗した場合、カードは SDMMC\_D ライン上で失敗を示し、転送されたデータは破棄されて書き込まれず、それ以降に転送されたブロック (複数ブロック書き込みモード時) はすべて無視されます。

ホストが、累積長がブロック長に合わない部分的なブロックを使用し、ブロックのミスアラインメントが許可されない場合 (CSD パラメータ WRITE\_BLK\_MISALIGN がセットされていない)、カードはミスアラインされた最初のブロック開始前にブロックミスアラインメントエラーを検出します。(ステータスレジスタの ADDRESS\_ERROR ビットがセットされます)。ホストが書き込み保護されたエリアに書き込みもうとした場合も、書き込み操作がアボートされます。ただし、この場合、カードは WP\_VIOLATION ビットをセットします。

CID および CSD レジスタをプログラムする場合、以前のブロック長設定は不要です。転送されるデータも、CRC 保護されます。CSD または CID レジスタの一部が ROM に格納される場合、この変更不

能部分が受信バッファの対応する部分と一致しなければなりません。これが一致しない場合、カードはエラーを報告し、レジスタの内容を変更しません。一部のカードでは、ブロックデータの書き込みに時間がかかり、その時間が予測不能ことがあります。ブロックデータの受信と CRC チェックの完了後、カードは書き込みを開始し、書き込みバッファがフルで新しい WRITE\_BLOCK コマンドから新しいデータを受け入れられない場合には、SDMMC\_D ラインをローに保ちます。ホストは、SEND\_STATUS コマンド (CMD13) でいつでもカードのステータスをポーリングでき、カードはそれぞれのステータスで応答します。READY\_FOR\_DATA ステータスビットは、カードが新しいデータを受け入れられるかどうか、または書き込み動作がまだ進行中かどうかを示します。ホストは、カードを切断状態にする CMD7 を発行してカードの選択を解除し (別のカードを選択するため)、書き込み操作を中断せずに SDMMC\_D ラインを解放することができます。カードを再び選択したときにプログラミングがまだ続行中であり、書き込みバッファが使用できない場合には、SDMMC\_D をローにプルすることによって、ビジー表示を再びアクティブにします。

## 35.4.6 ブロック読み出し

ブロック読み出しモードでは、データ転送の基本単位はブロックであり、その最大サイズは CSD (READ\_BL\_LEN) で定義されます。READ\_BL\_PARTIAL がセットされると、開始アドレスと終了アドレスが 1 つの物理ブロック (READ\_BL\_LEN で定義) に完全に含まれる小さなブロックを送信することもできます。各ブロックの最後には CRC を付加して、データの整合性を確保します。CMD17 (READ\_SINGLE\_BLOCK) は、ブロック読み出しを開始し、転送完了後、カードは転送状態に戻ります。

CMD18 (READ\_MULTIPLE\_BLOCK) は、複数の連続したブロックの転送を開始します。

ホストは、複数ブロック操作内においてはタイプに関係なく、いつでも読み出しをアボートできます。トランザクションは、送信停止コマンドを送信してアボートします。

カードが複数ブロック読み出し操作中にエラー (範囲外、アドレスのミスアラインメント、内部エラーなど) を検出した場合は、データ転送を停止して、データ状態に留まります。その場合、ホストは、送信停止コマンドを送信して操作をアボートする必要があります。読み出しエラーは、送信停止コマンドに対するレスポンスで報告されます。

カードが事前定義されたブロック数による複数ブロック操作で、最後のブロックを送信した後でホストが送信停止コマンドを送信した場合には、カードがすでにデータ状態ではないため、不正なコマンドとして応答されます。ホストが、累積長がブロック長に合わない部分的なブロックを使用し、ブロックのミスアラインメントが許可されない場合、カードはミスアラインされた最初のブロック開始前にブロックミスアラインメントエラーを検出します (ステータスレジスタの ADDRESS\_ERROR ビットがセットされます)。

## 35.4.7 ストリームアクセス、ストリーム書き込み、およびストリーム読み出し (MultiMediaCard のみ)

ストリームモードでは、データはバイト単位で転送され、各ブロックの最後に CRC は付加されません。

### ストリーム書き込み (MultiMediaCard のみ)

WRITE\_DAT\_UNTIL\_STOP (CMD20) は、SDMMC カードホストからカードへのデータ転送を指定されたアドレスで開始し、SDMMC カードホストが停止コマンドを発行するまで続けます。部分的なブロックが許可される場合 (CSD パラメータ WRITE\_BL\_PARTIAL がセットされている)、データストリームはカードアドレス空間内の任意のアドレスで開始および停止できます。そうでない場合は、ブロック境界でのみ開始と停止が可能です。転送データ量を事前に決めることができないため、CRC は使用できません。データ送信中にメモリ範囲の終わりに達し、SDMMC カードホストから停止コマンドが送信されなかった場合、その後転送されるデータは破棄されます。



ストリーム書き込み操作の最大クロック周波数は、カード固有データレジスタの次の式で与えられます。

$$\text{Maximumspeed} = \text{MIN}(\text{TRANSPEED}, \frac{(8 \times 2^{\text{writeblen}})(-\text{NSAC})}{\text{TAAC} \times \text{R2WFACTOR}})$$

- Maximumspeed = 書き込み最大周波数
- TRANSPEED = データ転送最大速度
- writeblen = 書き込みデータブロック最大長
- NSAC = データ読み出しアクセス時間 2 (CLK サイクル数)
- TAAC = データ読み出しアクセス時間 1
- R2WFACTOR = 書き込み速度係数

ホストがより高い周波数を使用しようとする、カードはデータを処理できずにプログラミングを停止し、ステータスレジスタの OVERRUN エラービットをセットすることがあります。この場合、それ以降のデータ転送は無視され、停止コマンドを待ちます (データ受信状態で)。ホストが書き込み保護領域に書き込もうとした場合も、書き込み操作がアボートされます。ただし、この場合、カードは WP\_VIOLATION ビットをセットします。

## ストリーム読み出し (MultiMediaCard のみ)

READ\_DAT\_UNTIL\_STOP (CMD11) は、ストリーム指向のデータ転送を制御します。

このコマンドは、SDMMC カードホストが STOP\_TRANSMISSION (CMD12) を送信するまで、指定されたアドレスからデータを送信するようにカードに指示します。停止コマンドには、シリアルコマンド送信による実行の遅延があり、データ転送は停止コマンドの終了ビット後に停止します。データ送信中にメモリ範囲の終わりに達し、SDMMC カードホストから停止コマンドが送信されなかった場合、その後転送されるデータは未定義と考慮されます。

ストリーム読み出し操作の最大クロック周波数は次の式によって与えられ、カード固有データレジスタのフィールドを使用します。

$$\text{Maximumspeed} = \text{MIN}(\text{TRANSPEED}, \frac{(8 \times 2^{\text{readblen}})(-\text{NSAC})}{\text{TAAC} \times \text{R2WFACTOR}})$$

- Maximumspeed = 読み出し最大周波数
- TRANSPEED = データ転送最大速度
- readblen = 読み出しデータブロック最大長
- writeblen = 書き込みデータブロック最大長
- NSAC = データ読み出しアクセス時間 2 (CLK サイクル数)
- TAAC = データ読み出しアクセス時間 1
- R2WFACTOR = 書き込み速度係数

ホストがより高い周波数を使用しようとする、カードはデータ転送を維持できません。これが発生した場合、カードはステータスレジスタの UNDERRUN エラービットをセットし、送信をアボートし、データ状態で停止コマンドを待ちます。

## 35.4.8 消去：グループ消去とセクタ消去

MultiMediaCard の消去可能単位は、消去グループです。消去グループは、カードの基本的な書き込み可能単位である書き込みブロックで分けます。消去グループのサイズはカード固有のパラメータであり、CSD で定義されます。

ホストは連続した範囲の消去グループを消去することができます。消去プロセスは、3 ステップのシーケンスで開始します。

はじめにホストは、ERASE\_GROUP\_START (CMD35) コマンドを使用して範囲の開始アドレスを定義し、次に、ERASE\_GROUP\_END (CMD36) コマンドを使用して範囲の最後のアドレスを定義し、最後に、ERASE (CMD38) コマンドを発行して消去プロセスを開始します。消去コマンドのアドレスフィールドは、バイト単位の消去グループアドレスです。カードは、消去グループサイズ未満のすべての LSB を無視して、アドレスを効果的に消去グループの境界まで切り捨てます。

消去コマンドが誤った順序で受信されると、カードはステータスレジスタの ERASE\_SEQ\_ERROR ビットをセットし、シーケンス全体をリセットします。

シーケンス外 (SEND\_STATUS を除き、どちらの消去コマンドでもない) のコマンドを受信した場合、カードはステータスレジスタの ERASE\_RESET ステータスビットをセットして消去シーケンスをリセットし、最後のコマンドを実行します。

消去範囲に書き込み保護ブロックが含まれていると、それらのブロックは消去されず、保護されていないブロックだけが消去されます。ステータスレジスタの WP\_ERASE\_SKIP ステータスビットがセットされます。

カードは、SDMMC\_D をローに保持することによって、消去が進行中であることを示します。実際の消去時間はかなり長くなることもあり、ホストは CMD7 を発行して、カードを選択解除できます。

## 35.4.9 ワイドバス選択または選択解除

ワイドバス (4 ビットバス幅) 操作モードは、SET\_BUS\_WIDTH (ACMD6) を使用して選択または選択解除されます。パワーアップ後または GO\_IDLE\_STATE (CMD0) 後のデフォルトのバス幅は、1 ビットです。SET\_BUS\_WIDTH (ACMD6) は、転送状態でのみ有効です。すなわち、SELECT/DESELECT\_CARD (CMD7) によって選択された後でなければバス幅を変更することはできません。

## 35.4.10 保護管理

SDMMC カードホストモジュールでは、カードの書き込み保護方法が 3 種類サポートされています。

1. 内部カード書き込み保護 (カード側の責任)
2. 機械的書き込み保護スイッチ (SDMMC カードホストモジュール側のみの責任)
3. パスワードで保護されたカードロック操作

### 内部カード書き込み保護

カードデータは、書き込みや消去から保護できます。CSD の永久的または一時的書き込み保護ビットをセットすると、製造業者やコンテンツプロバイダによってカード全体を永久的に保護することができます。セクタのグループの書き込み保護をサポートするカードの場合、CSD の WP\_GRP\_ENABLE ビットをセットするとデータの一部を保護することができ、書き込み保護はアプリケーションによって変更できます。書き込み保護は、CSD で指定された WP\_GRP\_SIZE セクタ単位です。SET\_WRITE\_PROT および CLR\_WRITE\_PROT コマンドは、アドレス指定されたグループの保護を制御します。SEND\_WRITE\_PROT コマンドは、1 ブロックの読み出しコマンドと似ていますが、カードは、32 個の書き込み保護ビット (指定されたアドレスから始まる 32 個の書き込み保護

ループを表す) に続いて、16個の CRC ビットを含むデータブロックを送信します。書き込み保護コマンドのアドレスフィールドは、バイト単位のグループアドレスです。

カードは、グループサイズ未満のすべての LSB を無視します。

## 機械的書き込み保護スイッチ

カードの横にある機械的なスライドタブによって、カードの書き込み保護をセット/クリアできます。スライドタブが開位置にあるときにはカードは書き込み保護され、閉位置にあるときには、カードの内容を変更できます。ソケット側の対応するスイッチは、カードが書き込み保護されていることを SDMMC カードホストモジュールに示します。カードを保護するのは、SDMMC カードホストモジュールです。書き込み保護スイッチの位置は、カードの内部回路にはわかりません。

## パスワード保護

パスワード保護機能によって、SDMMC カードホストモジュールはカードをパスワードでロック/アンロックできます。パスワードは 128 ビットの PWD レジスタに格納され、そのサイズは 8 ビットの PWD\_LEN レジスタで設定されます。これらのレジスタは不揮発性のため、パワーサイクルでは消去されません。ロックされたカードは、特定のコマンドに応答し、それを実行します。すなわち、SDMMC カードホストモジュールは、リセット、初期化、選択、およびステータスの問い合わせを行うことはできますが、カード上のデータにアクセスすることはできません。パスワードが設定されると (PWD\_LEN の値がゼロ以外)、カードはパワーアップ後に自動的にロックされます。CSD および CID レジスタの書き込みコマンド同様、ロック/アンロックコマンドは転送状態でのみ使用できます。この状態では、コマンドはアドレス引数を含まず、カードは使用前に選択しなければなりません。カードロック/アンロックコマンドは、通常の 1 ブロック書き込みコマンドの構造とバストランザクシオンタイプとなっています。転送されるデータブロックは、コマンドに必要なすべての情報 (パスワード設定モード、PWD 自体、およびカードのロック/アンロック) を含みます。コマンドデータブロックサイズは、カードロック/アンロックコマンド送信前に、SDMMC カードホストモジュールによって定義され、表 213 に示す構造となっています。

ビット設定は、次のとおりです。

- ERASE : セットすると、強制的に消去操作が行われます。他のすべてのビットはゼロでなければならない、コマンドバイトだけが送信されます。
- LOCK\_UNLOCK : セットすると、カードをロックします。LOCK\_UNLOCK は SET\_PWD と同時にセットできますが、CLR\_PWD と同時にセットすることはできません。
- CLR\_PWD : セットすると、パスワードデータをクリアします。
- SET\_PWD : セットすると、パスワードデータをメモリに保存します。
- PWD\_LEN : パスワードの長さ (バイト数) を定義します。
- PWD : パスワード (コマンドに応じて、新しいパスワードまたは現在使用中のパスワード)。

以下のセクションでは、パスワードのセット/クリア、カードのロック/アンロック、および強制消去のコマンドシーケンスを説明します。

## パスワードの設定

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT\_CARD、CMD7)。
2. 送信するブロック長 (SET\_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック/アンロックモード、8 ビットの PWD\_LEN、および新しいパスワードのバイト数によって定義されます。パスワードの書き換えが完了すると、ブロックサイズでは古いパスワードと新しいパスワードの両方がコマンドとともに送信されることを考慮する必要があります。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (SET\_PWD= 1)、長さ (PWD\_LEN)、

およびパスワード (PWD) 自体を示します。パスワードの書き換えが完了すると、長さ値 (PWD\_LEN) には古いパスワードと新しいパスワードの両方の長さが含まれ、PWD フィールドには、古いパスワード (現在使用中) に続いて新しいパスワードが含まれます。

4. パスワードが一致すると、新しいパスワードとそのサイズがそれぞれ PWD と PWD\_LEN フィールドに保存されます。送信された古いパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされ、パスワードは変更されません。

パスワード長フィールド (PWD\_LEN) は、パスワードが現在設定されているかどうかを示します。このフィールドがゼロ以外のときには、パスワードが設定されていて、カードはパワーアップ後に自身をロックします。LOCK\_UNLOCK ビットをセットするか (パスワードの設定時に)、カードをロックする追加のコマンドを送信すると、現在のパワーセッションでただちにカードをロックすることが可能です。

## パスワードのリセット

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT\_CARD、CMD7)。
2. 送信するブロック長 (SET\_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック/アンロックモード、8 ビットの PWD\_LEN、および現在使用中のパスワードのバイト数によって定義されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (CLR\_PWD=1)、長さ (PWD\_LEN)、およびパスワード (PWD) 自体を示します。LOCK\_UNLOCK ビットは無視されます。
4. パスワードが一致すると、PWD フィールドはクリアされ、PWD\_LEN は 0 にセットされます。送信されたパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされ、パスワードは変更されません。

## カードのロック

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT\_CARD、CMD7)。
2. 送信するブロック長 (SET\_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック/アンロックモード (表 213 のバイト 0)、8 ビットの PWD\_LEN、および現在のパスワードのバイト数で定義されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (LOCK\_UNLOCK = 1)、長さ (PWD\_LEN)、およびパスワード (PWD) 自体を示します。
4. パスワードが一致すると、カードはロックされ、カードステータスレジスタの CARD\_IS\_LOCKED ステータスビットがセットされます。送信されたパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされ、ロックは失敗します。

パスワードの設定とカードのロックは、同じシーケンスで行うことが可能です。この場合、SDMMC カードホストモジュールは、パスワードの設定に必要なすべてのステップを実行します ([パスワードの設定 \(1212 ページ\)](#) を参照)。ただし、新しいパスワードコマンドを送信するときには、ステップ 3 で LOCK\_UNLOCK ビットをセットする必要があります。

パスワードが以前に設定されているときには (PWD\_LEN が 0 でない)、カードは電源投入リセット後に自動的にロックされます。ロックされたカードをロックしようとしたり、パスワードが設定されていないカードをロックしようすると、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされます。



## カードのアンロック

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT\_CARD、CMD7)。
2. 送信するブロック長 (SET\_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード (表 213 のバイト 0)、8 ビットの PWD\_LEN、および現在のパスワードのバイト数で定義されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (LOCK\_UNLOCK = 0)、長さ (PWD\_LEN)、およびパスワード (PWD) 自体を示します。
4. パスワードが一致すると、カードはアンロックされ、カードステータスレジスタの CARD\_IS\_LOCKED ステータスビットがクリアされます。送信されたパスワードのサイズまたは内容が予期されたパスワードと一致しないときには、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされ、カードはロックされたままとなります。

アンロック機能は、現在のパワーセッションでのみ有効です。PWD フィールドがクリアされていないときには、カードは次のパワーアップ時に自動的にロックされます。

アンロックされたカードをアンロックしようとする、アンロックは失敗し、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされます。

## 強制消去

ユーザがパスワード (PWD の内容) を忘れた場合は、カード上のすべてのデータをクリアした後で、カードにアクセスすることができます。この強制消去操作では、すべてのカードデータとすべてのパスワードデータが消去されます。

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT\_CARD、CMD7)。
2. ブロック長 (SET\_BLOCKLEN、CMD16) を 1 バイトに設定します。8 ビットのカードロック／アンロックバイト (表 213 のバイト 0) だけが送信されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータバイトとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (ERASE = 1) を示します。他のすべてのビットはゼロでなければなりません。
4. ERASE ビットがデータフィールドでセットされた唯一のビットであるときには、PWD および PWD\_LEN フィールドを含め、すべてのカード内容が消去され、カードはロックされなくなります。他にもセットされているビットがあるときには、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされ、カードはすべてのデータを保持し、ロックされたままとなります。

アンロックされたカードを強制消去しようすると、カードステータスレジスタの LOCK\_UNLOCK\_FAILED エラービットがセットされます。

## 35.4.11 カードステータスレジスタ

レスポンスフォーマット R1 には、カードステータスと呼ばれる 32 ビットのフィールドがあります。このフィールドは、カードステータス情報（ローカルステータスレジスタに格納可能）をホストに送信するためのものです。特に記載がない限り、ステータスエントリは、常に、以前に発行されたコマンドに関連するものとしします。

表 200 に、さまざまなステータスエントリを示します。表中のタイプおよびクリア条件フィールドは、次のように略記されます。

タイプ :

- E : エラービット。
- S : ステータスビット。
- R : 実際のコマンドレスポンスに対して検出され、セットされます。
- X : コマンド実行中に検出され、セットされます。SDMMC カードホストは、ステータスコマンドを発行してカードをポーリングし、これらのビットを読み出します。

クリア条件 :

- A : カードの現在の状態によります。
- B : 常に前のコマンドに関連しています。有効なコマンドを受信すると、クリアされます（1 コマンドの遅延）。
- C : 読み出すとクリアされます。

表 200. カードステータス

ビット	識別子	タイプ	値	説明	クリア条件
31	ADDRESS_OUT_OF_RANGE	E R X	0= エラーなし 1= エラー	コマンドアドレス引数が、このカードの許容範囲を超えています。 複数ブロックまたはストリーム読み出し/書き込み操作は、有効なアドレスから開始していますが、カード容量を超える読み込みまたは書き込みをしようとしています。	C
30	ADDRESS_MISALIGN		0= エラーなし 1= エラー	現在設定されているブロック長にしたがってコマンドアドレス引数が指している最初のデータブロックは、カードの物理ブロックに合っていません。 複数ブロックの読み出し/書き込み操作は、（有効なアドレス/ブロック長の組み合わせから開始している場合でも）カードの物理ブロックに合っていないデータブロックの読み込みまたは書き込みをしようとしています。	C
29	BLOCK_LEN_ERROR		0= エラーなし 1= エラー	SET_BLOCKLEN コマンドの引数がカードの許容する最大値を超えているか、以前に定義されたブロック長が現在のコマンドに対して無効です（ホストが書き込みコマンドを発行し、現在のブロック長がカードの許容する最大値より小さく、部分的なブロックの書き込みが許可されない場合など）。	C
28	ERASE_SEQ_ERROR		0= エラーなし 1= エラー	消去コマンドのシーケンス中にエラーが発生しました。	C
27	ERASE_PARAM	E X	0= エラーなし 1= エラー	消去に対して無効な消去グループが選択されました。	C



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

表 200. カードステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
26	WP_VIOLATION	E X	0= エラーなし 1= エラー	書き込み保護されたブロックをプログラムしようとした。	C
25	CARD_IS_LOCKED	S R	0 = カードがアンロックされています。 1 = カードがロックされています。	セットされると、カードがホストによってロックされていることを示します。	A
24	LOCK_UNLOCK_FAILED	E X	0= エラーなし 1= エラー	ロック/アンロックコマンドで、シーケンスエラーまたはパスワードエラーが発生したときにセットされます。	C
23	COM_CRC_ERROR	E R	0= エラーなし 1= エラー	以前のコマンドの CRC チェックが失敗しました。	B
22	ILLEGAL_COMMAND	E R	0= エラーなし 1= エラー	カードの状態に対して不正なコマンドです。	B
21	CARD_ECC_FAILED	E X	'0' = 成功 '1' = 失敗	カードの内部 ECC が適用されましたが、データの訂正に失敗しました。	C
20	CC_ERROR	E R	0= エラーなし 1= エラー	(標準では定義されていません) ホストコマンドに関係ないカードエラーが発生しました。	C
19	ERROR	E X	0= エラーなし 1= エラー	(標準では定義されていません) 最後のホストコマンドの実行に関係した一般カードエラーが検出されました (読み出しまたは書き込みの失敗など)。	C
18	予約済み				
17	予約済み				
16	CID/CSD_OVERWRITE	E X	'0'= エラーなし '1'= エラー	次のいずれかのエラーです。 - CID レジスタはすでに書き込まれていて、上書きできません。 - CSD の読み出し専用セクションが、カードの内容と一致しません。 - 不正なコピー (オリジナルとしてセット) または永久的な WP ビット (保護されない) を元に戻そうとしました。	C
15	WP_ERASE_SKIP	E X	0= 保護されていません 1= 保護されています	既存の書き込みが理由で、部分的なアドレス空間だけが消去されたときにセットされます。	C
14	CARD_ECC_DISABLED	S X	0= 有効 1= 無効	内部 ECC を使用せずに、コマンドが実行されました。	A
13	ERASE_RESET		0= クリア 1= セット	シーケンス外の消去コマンド (CMD35、CMD36、CMD38、または CMD13 以外) が受信されたため、実行前に消去シーケンスがクリアされました。	C

表 200. カードステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
12:9	CURRENT_STATE	S R	0 = アイドル 1 = レディ状態 2 = 識別 3 = スタンバイ 4 = 転送 5 = データ 6 = 受信 7 = プログラム 8 = Dis 9 = Btst 10-15 = 予約済み	コマンドを受信したときのカードの状態。コマンドの実行によってカードの状態が変化する場合、次のコマンドのレスポンスで、ホストに状態が知らされます。4 ビットは、0 から 15 までの2 進数として解釈されます。	B
8	READY_FOR_DATA	S R	0 = 非レディ状態 1 = レディ	バス上のバッファ空信号に対応します。	-
7	SWITCH_ERROR	E X	0 = エラーなし 1 = スイッチエラー	セットされた場合に、カードは SWICTH コマンドで要求された予期されたモードに切り替わりませんでした。	B
6	予約済み				
5	APP_CMD	S R	0 = 無効 1 = 有効	カードは ACMD を期待しているか、コマンドが ACMD として解釈されたという指示を期待します。	C
4	SD I/O カード用に予約済み				
3	AKE_SEQ_ERROR	E R	0 = エラーなし 1 = エラー	認証プロセスのシーケンスにエラーがあります。	C
2	アプリケーション固有のコマンド用に予約済み。				
1	製造業者のテストモード用に予約済み。				
0					

## 35.4.12 SD ステータスレジスタ

SD ステータスは、SD メモリカード独自の機能に関連するステータスビットを含み、将来のアプリケーション固有の用途で使用できます。SD ステータスのサイズは、512 ビットのデータブロックです。このレジスタの内容は、ACMD13 が送信された場合に（CMD55 に続いて CMD13）、SDMMC カードホストに送信されます。ACMD13 は、転送状態のカード（カードが選択されている）にのみ送信できます。

表 201 に、SD ステータスレジスタのさまざまなエントリを示します。表中のタイプおよびクリア条件フィールドは、次のように略記されます。

タイプ :

- E: エラービット。
- S: ステータスビット。
- R: 実際のコマンドレスポンスに対して検出され、セットされます。
- X: コマンド実行中に検出され、セットされます。SDMMC カードホストは、ステータスコマンドを発行してカードをポーリングし、これらのビットを読み出します。

クリア条件 :

- A: カードの現在の状態によります。
- B: 常に前のコマンドに関連しています。有効なコマンドを受信すると、クリアされます（1 コマンドの遅延）。
- C: 読み出すとクリアされます。

表 201. SD ステータス

ビット	識別子	タイプ	値	説明	クリア条件
511 : 510	DAT_BUS_WIDTH	S R	00= 1 (デフォルト) 01= 予約済み 10= 4 ビット幅 11= 予約済み	SET_BUS_WIDTH コマンドによって定義された、現在定義されているバス幅を示します。	A
509	SECURED_MODE	S R	0= セキュアモードではありません。 1= セキュアモードです。	カードはセキュア操作モードになっています（「SD セキュリティ仕様」を参照）。	A
508 : 496	予約済み				
495 : 480	SD_CARD_TYPE	S R	00xxh= 物理仕様 Ver1.01-2.00 で定義された SD メモリカード（“x”= 無視） 現在定義されているカードは、次のとおりです。 0000= 通常の SD RD/WR カード 0001= SD ROM カード	将来は、8 つの LSB を使用して、さまざまな種類の SD メモリカードを定義する予定です（各ビットは異なる SD タイプを定義します）。8 つの MSB は、現在の SD 物理レイヤ仕様に準拠しない SD カードを定義するために使用します。	A
479 : 448	SIZE_OF_PROTECTED_AREA	S R	保護領域のサイズ（下記を参照）	（下記を参照）	A
447 : 440	SPEED_CLASS	S R	カードの速度クラス（下記を参照）	（下記を参照）	A
439 : 432	PERFORMANCE_MOVE	S R	1 [MB/s] ステップが表す移動性能（下記を参照）	（下記を参照）	A
431:428	AU_SIZE	S R	AU のサイズ（下記を参照）	（下記を参照）	A

表 201. SD ステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
427:424	予約済み				
423:408	ERASE_SIZE	S R	一度に消去される AU の数	(下記を参照)	A
407:402	ERASE_TIMEOUT	S R	UNIT_OF_ERASE_AU で指定された領域を消去する際のタイムアウト値	(下記を参照)	A
401:400	ERASE_OFFSET	S R	消去時間に追加される固定オフセット値	(下記を参照)	A
399:312	予約済み				
311:0	製造業者のために予約済み。				

## SIZE\_OF\_PROTECTED\_AREA

このフィールドの設定は、標準カードと大容量カードで異なります。標準容量のカードでは、保護領域の容量は次のように計算されます。

$$\text{保護領域} = \text{SIZE\_OF\_PROTECTED\_AREA} \times \text{MULT} \times \text{BLOCK\_LEN}$$

SIZE\_OF\_PROTECTED\_AREA は、MULT\*BLOCK\_LEN 単位で指定されます。

大容量のカードでは、保護領域の容量は、このフィールドで指定されます。

$$\text{保護領域} = \text{SIZE\_OF\_PROTECTED\_AREA}$$

SIZE\_OF\_PROTECTED\_AREA は、バイト単位で指定されます。

## SPEED\_CLASS

この 8 ビットフィールドは速度クラスを示し、値は  $P_{W/2}$  で求めることができます ( $P_W$  は書き込み性能)。

表 202. 速度クラスコードフィールド

SPEED_CLASS	値の定義
00h	クラス 0
01h	クラス 2
02h	クラス 4
03h	クラス 6
04h – FFh	予約済み

## PERFORMANCE\_MOVE

この 8 ビットフィールドは、Pm（移動パフォーマンス）を示し、値は、1 [MB/sec] ステップで設定できます。カードが使用済みの RU（記録ユニット）を移動しない場合は、Pm を無限大であるとみなします。このフィールドを FFh にセットすると、無限を意味します。

表 203. 移動パフォーマンスフィールド

PERFORMANCE_MOVE	値の定義
00h	定義されません。
01h	1 [MB/sec]
02h	02h 2 [MB/sec]
-----	-----
FEh	254 [MB/sec]
FFh	無限

## AU\_SIZE

この 4 ビットフィールドは、AU サイズを示し、値は、16 KB から 2 の累乗で選択できます。

表 204. AU\_SIZE フィールド

AU_SIZE	値の定義
00h	定義されません。
01h	16 KB
02h	32 KB
03h	64 KB
04h	128 KB
05h	256 KB
06h	512 KB
07h	1 MB
08h	2 MB
09h	4 MB
Ah – Fh	予約済み

最大 AU サイズはカード容量に依存し、表 205 に示されています。カードは、RU サイズと最大 AU サイズの間の任意の AU サイズに設定できます。

表 205. 最大 AU サイズ

容量	16 MB-64 MB	128 MB-256 MB	512 MB	1 GB-32 GB
最大 AU サイズ	512 KB	1 MB	2 MB	4 MB

## ERASE\_SIZE

この 16 ビットフィールドは、NERASE を示します。AU の NERASE 数が消去される時、タイムアウト値は ERASE\_TIMEOUT によって指定されます ([ERASE\\_TIMEOUT](#) を参照)。ホストは、1 回の操作で消去される正しい AU の数を決定する必要があります。それにより、ホストは消去操作の進捗状況を表示することができます。このフィールドが 0 にセットされた場合、消去タイムアウト計算はサポートされません。

表 206. 消去サイズフィールド

ERASE_SIZE	値の定義
0000h	消去タイムアウトの計算はサポートされません。
0001h	1 AU
0002h	2 AU
0003h	3 AU
-----	-----
FFFFh	65535 AU

## ERASE\_TIMEOUT

この 6 ビットフィールドは、TERASE を示し、値は、ERASE\_SIZE によって指定された複数の AU が消去される時のオフセットからの消去タイムアウトを示します。ERASE\_TIMEOUT の範囲は、63 秒までの間で定義でき、カード製造業者は実装に応じて、ERASE\_SIZE と ERASE\_TIMEOUT の任意の組み合わせを選ぶことができます。ERASE\_TIMEOUT を決めると、ERASE\_SIZE が決まります。

表 207. 消去タイムアウトフィールド

ERASE_TIMEOUT	値の定義
00	消去タイムアウトの計算はサポートされません。
01	1 [sec]
02	2 [sec]
03	3 [sec]
-----	-----
63	63 [sec]

## ERASE\_OFFSET

この 2 ビットフィールドは、TOFFSET を示し、4 つの値から 1 つを選択できます。このフィールドは、ERASE\_SIZE および ERASE\_TIMEOUT フィールドが 0 にセットされている場合は意味を持ちません。

表 208. 消去オフセットフィールド

ERASE_OFFSET	値の定義
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

## 35.4.13 SD I/O モード

### SD I/O 割込み

SD インタフェースのピンには、SD I/O カードから MultiMediaCard/SD モジュールへの割込みを可能にする割込み機能が用意されています。ピン 8 は、4 ビット SD モードで動作するときに SDMMC\_D1 として使用され、MultiMediaCard/SD モジュールへのカード割込みを示します。割込みの使用は、各カードまたはカード内の各機能でオプションとなっています。SD I/O 割込みはレベル対応であるため、割込みラインは MultiMediaCard/SD モジュールによって認識され、割込みが処理されるか、割込み期間の終了によってネゲートされるまで、アクティブ（ロー）に保たれる必要があります。MultiMediaCard/SD モジュールが割込み処理を終えると、SD I/O カードの内部レジスタの該当するビットへの I/O 書き込みによって割込みステータスビットがクリアされます。すべての SD I/O カードの割込み出力はアクティブローであり、アプリケーションはすべてのデータライン (SDMMC\_D[3:0]) 上に、外部プルアップ抵抗を提供する必要があります。MultiMediaCard/SD モジュールは、割込み期間に限って、割込み検出器にピン 8 (SDMMC\_D/IRQ) のレベルをサンプリングします。その他の状況では、MultiMediaCard/SD モジュールはこの値を無視します。

割込み期間は、メモリと I/O の両方の操作に適用されます。単一ブロックの操作の割込み期間の定義は、複数ブロックのデータ転送の定義とは異なります。

### SD I/O のサスペンドとレジューム

マルチファンクション SD I/O または I/O とメモリの両方の機能を持つカードでは、MMC/SD バスのアクセスを共有する複数のデバイス (I/O とメモリ) があります。複数デバイス間で MMC/SD モジュールへのアクセスを共有するため、SD I/O およびコンボカードはオプションでサスペンド/レジュームの概念を実装します。カードがサスペンド/レジュームをサポートするときには、MMC/SD モジュールは 1 つの機能またはメモリへのデータ転送を一時的に中止 (サスペンド) して、別の機能またはメモリへの、より優先順位の高い転送用にバスを解放できます。この優先順位の高い転送が完了すると、本来の転送がアポートした所から再開 (リスタート) されます。サスペンド/レジュームのサポートは、カードごとのオプションです。MMC/SD バス上でサスペンド/レジューム操作を行うには、MMC/SD モジュールは以下の手順を実行します。

1. SDMMC\_D [3:0] ラインを現在使用している機能を特定します。
2. 優先順位が低いトランザクションや時間のかかるトランザクションをサスペンドするように要求します。
3. トランザクションのサスペンドの完了を待ちます。
4. 優先順位の高いトランザクションを開始します。
5. 優先順位の高いトランザクションの完了を待ちます。
6. サスペンドされたトランザクションをリストアします。

### SD I/O ReadWait

オプションの ReadWait (RW) 操作は、SD の 1 ビットおよび 4 ビットモードについてのみ定義されます。ReadWait 操作によって、MMC/SD モジュールは複数のレジスタを読み出していることをカードに知らせ (IO\_RW\_EXTENDED、CMD53)、データ転送を一時的にストールし、その間に MMC/SD モジュールは SD I/O デバイス内の機能にコマンドを送信できます。カードが ReadWait プロトコルをサポートしているかどうかを判断するには、MMC/SD モジュールが内部カードレジスタの機能ビットをテストする必要があります。ReadWait のタイミングは、割込み期間に基づいています。

## 35.4.14 コマンドおよびレスポンス

### アプリケーション固有コマンドと汎用コマンド

SDMMC カードホストモジュールシステムは、さまざまな種類のアプリケーションに標準インタフェースを提供するように設計されています。この環境では、特定の顧客やアプリケーション機能へのニーズがあります。これらの機能を実装するために、この標準ではアプリケーション固有コマンド (ACMD) と汎用コマンド (GEN\_CMD) の 2 種類のコマンドが定義されています。

APP\_CMD (CMD55) コマンドを受信すると、カードは次のコマンドがアプリケーション固有コマンドであると予想します。ACMD は、通常の MultiMediaCard コマンドと同じ構造を持っており、同じ CMD 番号を持つことができます。カードは、APP\_CMD (CMD55) の後に現れるコマンドを ACMD として認識します。APP\_CMD (CMD55) のすぐ後に続くコマンドが定義済みのアプリケーション固有コマンドでないときには、標準コマンドが使用されます。たとえば、カードで SD\_STATUS (ACMD13) が定義されているときに APP\_CMD (CMD55) の直後に CMD13 を受信すると、これは SD\_STATUS (ACMD13) であると解釈されます。ただし、カードが APP\_CMD (CMD55) の直後に CMD7 を受信し、カードで ACMD7 が定義されていないときには、これは標準 (SELECT/DESELECT\_CARD) CMD7 であると解釈されます。

製造業者固有の ACMD の 1 つを使用するには、SD カードホストが以下の手順を実行する必要があります。

1. APP\_CMD (CMD55) を送信します。  
カードは MultiMediaCard/SD モジュールにตอบสนองして、APP\_CMD ビットがセットされていることと、ACMD が予期されていることを示します。
2. 必要な ACMD を送信します。  
カードは MultiMediaCard/SD モジュールにตอบสนองして、APP\_CMD ビットがセットされていることと、受け取ったコマンドが ACMD として解釈されたことを示します。ACMD でないコマンドが送信されたときには、カードはそのコマンドを通常の MultiMediaCard コマンドとして扱い、カードステータスレジスタの APP\_CMD ビットはクリアされたままとなります。

無効なコマンド (ACMD でも CMD でもない) が送信されたときには、標準の MultiMediaCard 不正コマンドエラーとして処理されます。

GEN\_CMD のバストランザクションは、単一ブロックの読み出しまたは書き込みコマンド (WRITE\_BLOCK、CMD24、または READ\_SINGLE\_BLOCK、CMD17) と同じです。この場合、引数はアドレスではなくデータ転送の方向を示し、データブロックはベンダ固有のフォーマットと意味を持ちます。

カードは、GEN\_CMD (CMD56) を送信する前に選択されなければなりません (転送状態でなければなりません)。データブロックサイズは、SET\_BLOCKLEN (CMD16) によって定義されます。GEN\_CMD (CMD56) へのレスポンスは、R1b フォーマットです。

### コマンドタイプ

アプリケーション固有コマンドと汎用コマンドは、どちらも、次の 4 つのタイプに分類されます。

- **ブロードキャストコマンド (BC)** : すべてのカードに送信されます。レスポンスは返されません。
- **レスポンス付きブロードキャストコマンド (BCR)** : すべてのカードに送信されます。レスポンスは、すべてのカードから同時に受信します。
- **アドレス指定 (ポイントツーポイント) コマンド (AC)** : 選択されたカードに送信されます。SDMMC\_D ライン上のデータ転送は含みません。
- **アドレス指定 (ポイントツーポイント) データ転送コマンド (ADTC)** : 選択されたカードに送信されます。SDMMC\_D ライン上のデータ転送は含みます。



## コマンドフォーマット

コマンドフォーマットについては、表 192 (1198 ページ) を参照してください。

## MultiMediaCard/SD モジュール用コマンド

表 209. ブロック指向書き込みコマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD23	ac	[31:16] 0 にセット [15:0] ブロック数	R1	SET_BLOCK_COUNT	複数ブロック読み出しまたは書き込みコマンドで転送されるブロック数を定義します。
CMD24	adtc	[31:0] データアドレス	R1	WRITE_BLOCK	SET_BLOCKLEN コマンドによって選択されたサイズのブロックを書き込みます。
CMD25	adtc	[31:0] データアドレス	R1	WRITE_MULTIPLE_BLOCK	STOP_TRANSMISSION まで、または、要求されたブロック数を受信するまで、データのブロックを書き込み続けます。
CMD26	adtc	[31:0] スタッピット	R1	PROGRAM_CID	カード識別レジスタのプログラミングです。このコマンドは、1 枚のカードにつき 1 度だけ発行する必要があります。カードには、最初のプログラミングの後にこの操作を防ぐためのハードウェアが含まれています。通常、このコマンドは製造業者用に予約されています。
CMD27	adtc	[31:0] スタッピット	R1	PROGRAM_CSD	CSD のプログラマブルビットのプログラミング用です。

表 210. ブロック指向書き込み保護コマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD28	ac	[31:0] データアドレス	R1b	SET_WRITE_PROT	カードに書き込み保護機能がある場合、このコマンドはアドレス指定されたグループの書き込み保護ビットをセットします。書き込み保護のプロパティは、カード固有データ (WP_GRP_SZIE) でコード化されます。
CMD29	ac	[31:0] データアドレス	R1b	CLR_WRITE_PROT	カードが書き込み保護機能を備えている場合、このコマンドはアドレス指定されたグループの書き込み保護ビットをクリアします。
CMD30	adtc	[31:0] 書き込み保護 データアクセス	R1	SEND_WRITE_PROT	カードが書き込み保護機能を備えている場合、このコマンドはカードに書き込み保護ビットのステータスを送るように要求します。
CMD31	予約済み				

表 211. 消去コマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD32 ... CMD34					予約済み古いバージョンの MultiMediaCard との下位互換性を保つため、これらのコマンドインデックスは使用できません。
CMD35	ac	[31:0] データアドレス	R1	ERASE_GROUP_START	消去対象として選択される範囲の最初の消去グループのアドレスを設定します。
CMD36	ac	[31:0] データアドレス	R1	ERASE_GROUP_END	消去対象として選択される連続した範囲内の最後の消去グループのアドレスを設定します。
CMD37					予約済み古いバージョンの MultiMediaCard との下位互換性を保つため、これらのコマンドインデックスは使用できません。
CMD38	ac	[31:0] スタッフビット	R1	ERASE	以前に選択されたすべての書き込みブロックを消去します。

表 212. I/O モードコマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD39	ac	[31:16] RCA [15:15] レジスタ書き込みフラグ [14:8] レジスタアドレス [7:0] レジスタデータ	R4	FAST_IO	8 ビット (レジスタ) データフィールドの読み出しおよび書き込みに使用されます。このコマンドは、カードとレジスタをアドレス指定し、書き込みフラグがセットされている場合は書き込み用データを提供します。R4 レスポンスは、アドレス指定されたレジスタから読み出されたデータを含みます。このコマンドは、MultiMediaCard 標準で定義されていないアプリケーション依存レジスタにアクセスします。
CMD40	bcr	[31:0] スタッフビット	R5	GO_IRQ_STATE	システムを割り込みモードにします。
CMD41		予約済み			

表 213. カードのロック

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD42	adtc	[31:0] スタッフビット	R1b	LOCK_UNLOCK	パスワードのセット/クリア、またはカードのロック/アンロックを行います。データブロックのサイズは、SET_BLOCK_LEN コマンドによって設定されます。
CMD43 ... CMD54		予約済み			



表 214. アプリケーション固有コマンド

CMD インデックス	タイプ	引数	レスポンスフォーマット	略語	説明
CMD55	ac	[31:16] RCA [15:0] スタッピット	R1	APP_CMD	次のコマンドビットが標準コマンドでなく、アプリケーション固有コマンドであることをカードに示します。
CMD56	adtc	[31:1] スタッピット [0] : RD/WR	-	-	汎用/アプリケーション固有コマンドに対し、データブロックをカードに転送するため、またはカードからデータを取得するために使用されます。データブロックのサイズは、SET_BLOCK_LEN コマンドによって設定されます。
CMD57 ... CMD59	予約済み				
CMD60 ... CMD63	製造業者用に予約されています。				

## 35.5 レスポンスフォーマット

すべてのレスポンスは SDMMC コマンドライン SDMMC\_CMD 経由で送信されます。レスポンスの送信は、常に、レスポンスコードワードに対応するビット文字列の左のビットから始まります。コード長は、レスポンスのタイプに依存します。

レスポンスは、常にスタートビット (常に 0) から始まり、送信方向を示すビットが続きます (card=0)。下の表で x で示されている値は、可変エントリを示します。R3 レスポンスを除くすべてのレスポンスは、CRC によって保護されます。すべてのコマンドコードワードは、終了ビット (常に 1) で終わります。

レスポンスには 5 つのタイプがあります。フォーマットは、次のとおりです。

### 35.5.1 R1 (ノーマルレスポンスコマンド)

コード長 = 48 ビット。45:40 ビットはレスポンスされるコマンドのインデックスを示します。この値は 2 進コード番号 (0~63) として解釈されます。カードのステータスは 32 ビットにコード化されます。

表 215. R1 レスポンス

ビット位置	幅 (ビット)	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	X	コマンドインデックス
[39:8]	32	X	カードステータス
[7:1]	7	X	CRC7
0	1	1	終了ビット

## 35.5.2 R1b

R1 と同じですが、オプションとしてデータライン上にビジー信号を出力します。カードは、コマンドを受信する前の状態に基づいて、これらのコマンドを受信した後、ビジーになることがあります。

## 35.5.3 R2 (CID、CSD レジスタ)

コード長 = 136 ビット。CID レジスタの内容は、CMD2 および CMD10 コマンドのレスポンスとして送信されます。CSD レジスタの内容は、CMD9 のレスポンスとして送信されます。CID および CSD のビット [127... 1] だけが送信され、これらのレジスタの予約ビット [0] は、レスポンスの終了ビットで置き換えられます。カードは、SDMMC\_D0 をローに保持することによって、消去が進行中であることを示します。実際の消去時間はかなり長くなることもあり、ホストは CMD7 を発行して、カードを選択解除できます。

表 216. R2 レスポンス

ビット位置	幅 (ビット)	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	"111111"	コマンドインデックス
[127:1]	127	X	カードステータス
0	1	1	終了ビット

## 35.5.4 R3 (OCR レジスタ)

コード長：48 ビット。OCR レジスタの内容は、CMD1 へのレスポンスとして送信されます。レベルのコーディングは、次のようになります。制限された電圧ウィンドウ=ロー、カードビジ=ロー。

表 217. R3 レスポンス

ビット位置	幅 (ビット)	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	"111111"	予約済み
[39:8]	32	X	OCR レジスタ
[7:1]	7	"1111111"	予約済み
0	1	1	終了ビット

## 35.5.5 R4 (高速 I/O)

コード長：48 ビット。引数フィールドは、アドレス指定されたカードの RCA、読み出しまたは書き込みレジスタアドレス、およびその内容を含みます。

表 218. R4 レスポンス

ビット位置	幅 (ビット)	値	説明	
47	1	0	スタートビット	
46	1	0	送信ビット	
[45:40]	6	"100111"	CMD39	
[39:8] 引数フィールド	[31:16]	16	X	RCA
	[15:8]	8	X	レジスタアドレス
	[7:0]	8	X	読み出しレジスタの内容
[7:1]	7	X	CRC7	
0	1	1	終了ビット	

## 35.5.6 R4b

SD I/O のみ : CMD5 を受信した SDIO カードは、一意の SDIO レスポンス R4 で応答します。フォーマットは、次のとおりです。

表 219. R4b レスポンス

ビット位置	幅 (ビット)	値	説明	
47	1	0	スタートビット	
46	1	0	送信ビット	
[45:40]	6	X	予約済み	
[39:8] 引数フィールド	39	16	X	カードはレディ状態
	[38:36]	3	X	I/O 機能数
	35	1	X	現在のメモリ
	[34:32]	3	X	スタッフビット
	[31:8]	24	X	I/O ORC
[7:1]	7	X	予約済み	
0	1	1	終了ビット	

SD I/O カードが CMD5 を受信すると、カードの I/O 部分は、以降のすべてのコマンドに対して通常どおりに応答できるようになります。I/O カード内の機能のこの I/O 有効は、カードがリセット、パワーサイクル、または I/O リセットに書き込む CMD52 を受信するまで、セットされたままです。SD メモリ専用カードは、CMD5 に応答することがありますので注意してください。メモリ専用カードの正しいレスポンスは、*現在のメモリ* = 1 と *I/O 機能数* = 0 です。SD メモリカード仕様 version 1.0 に準拠するように作られたメモリ専用カードは、CMD5 を不正なコマンドとして検出して、応答しません。I/O を認識できるホストは、CMD5 を送信します。カードがレスポンス R4 で応答した場合、ホストはレスポンス R4 に含まれるデータから、カードの構成を判断します。

## 35.5.7 R5 (割り込みリクエスト)

MultiMediaCard 専用です。コード長 : 48 ビット。ホストによってレスポンスが生成された場合、引数の RCA フィールドは 0x0 になります。

表 220. R5 レスポンス

ビット位置	幅 (ビット)	値	説明	
47	1	0	スタートビット	
46	1	0	送信ビット	
[45:40]	6	"101000"	CMD40	
[39:8] 引数フィールド	[31:16]	16	X	獲得したカードまたはホストの RCA [31:16]
	[15:0]	16	X	定義されていません。IRQ データに使われる場合があります。
[7:1]	7	X	CRC7	
0	1	1	終了ビット	

## 35.5.8 R6

SD I/O 専用です。メモリデバイスによる CMD3 への通常のレスポンスです。表 221 に示されています。

表 221. R6 レスポンス

ビット位置	幅 (ビット)	値	説明	
47	1	0	スタートビット	
46	1	0	送信ビット	
[45:40]	6	"101000"	CMD40	
[39:8] 引数フィールド	[31:16]	16	X	獲得したカードまたはホストの RCA [31:16]
	[15:0]	16	X	定義されていません。IRQ データに使われる場合があります。
[7:1]	7	X	CRC7	
0	1	1	終了ビット	

カード [23:8] ステータスビットは、CMD3 が I/O 専用カードに送信されると変更されます。この場合、レスポンスの 16 ビットは、SD I/O 専用の値です。

- ビット [15] COM\_CRC\_ERROR
- ビット [14] ILLEGAL\_COMMAND
- ビット [13] ERROR
- ビット [12:0] 予約済み

## 35.6 SDIO I/O カード固有の操作

以下の機能は、SD I/O 固有の操作です。

- SDMMC\_D2 信号による SDIO 読み出しウェイト操作
- クロックの停止による SDIO 読み出しウェイト操作
- SDIO サスペンド/レジューム操作 (書き込みおよび読み出しサスペンド)
- SDIO 割込み

SDMMC は、SDMMC\_DCTRL[11] ビットがセットされている場合のみ、これらの機能をサポートします。ただし、特定のハードウェア実装を必要としない読み出しサスペンドを除きます。

### 35.6.1 SDMMC\_D2 信号による SDIO I/O 読み出しウェイト操作

最初のブロックが受信される前に読み出しウェイトインターバルを開始することができます。データパスが有効 (SDMMC\_DCTRL[0] ビットがセット)、SDIO 固有操作が有効 (SDMMC\_DCTRL[11] ビットがセット)、読み出しウェイト開始 (SDMMC\_DCTRL[10]=0 および SDMMC\_DCTRL[8]=1)、データ方向がカードから SDMMC (SDMMC\_DCTRL[1]=1) のとき、DPSM はアイドルから ReadWait へ直接移行します。ReadWait では、DPSM は 2 SDMMC\_CK クロックサイクル後に、SDMMC\_D2 を 0 に駆動します。この状態では、RWSTOP ビット (SDMMC\_DCTRL[9]) をセットすると、DPSM は、さらに 2 SDMMC\_CK クロックサイクルの間ウェイト状態にとどまり、1 クロックサイクルの間 SDMMC\_D2 を 1 に駆動します (SDIO 仕様に従う)。その後、DPSM はカードからデータを受信するまで再びウェイトを開始します。DPSM は、読み出しウェイト開始がセットされている場合でも、ブロックを受信中は読み出しウェイトインターバルを開始しません。読み出しウェイトインターバルは、CRC の受信後に開始されます。新しい読み出しウェイト操作を開始するには、RWSTOP ビットをクリアする必要があります。

まず、読み出しウェイトインターバル中は、SDMMC は SDMMC\_D1 上で SDIO 割込みを検出することができます。

## 35.6.2 SDMMC\_CK の停止による SDIO 読み出しウェイト操作

SDIO カードが先の読み出しウェイト方法をサポートしない場合、SDMMC は SDMMC\_CK を停止することで読み出しウェイトを実行することができます (SDMMC\_DCTRL は [セクション 35.6.1](#)に記載される方法と同じようにセットされますが、SDMMC\_DCTRL[10] = 1 です)。DPSM は、現在受信しているブロックの終了ビットの 2 SDMMC\_CK サイクル後にクロックを停止し、読み出しウェイトスタートビットがセットされた後に再びクロックを開始します。

SDMMC\_CK が停止されると、任意のコマンドをカードに発行できます。読み出しウェイトインターバル中、SDMMC は SDMMC\_D1 上の SDIO 割込みを検出できます。

## 35.6.3 SDIO サスペンド/レジューム操作

カードにデータを送信している間、SDMMC は書き込み操作をサスペンドできます。SDMMC\_CMD[11] ビットがセットされ、現在のコマンドがサスペンドコマンドであることを CPSM に示します。CPSM はレスポンスを解析し、カードから ACK を受信すると (サスペンドの受入れ)、現在のブロックの CRC トークンの受信後にアイドル状態となることを DPSM に知らせます。

ハードウェアは、サスペンド操作 (レジューム) を完了させるために送信しなければならない残りの送信ブロック数を保存しません。

書き込み操作は、カードからサスペンドコマンドの ACK を受信したときに、DPSM を無効にするだけで (SDMMC\_DCTRL[0] = 0)、ソフトウェアによってサスペンドできます。その場合、DPSM はアイドル状態に入ります。

読み出しをサスペンドするには、DPSM は Wait\_r 状態でサスペンドされる機能が完全なパケットを送信するのを待ってから、データ処理を停止します。アプリケーションは、FIFO が空になって、DPSM が自動的にアイドル状態になるまで、RxFIFO の読み出しを続けます。

## 35.6.4 SDIO 割込み

SDMMC\_DCTRL[11] ビットがセットされると、SDMMC\_D1 ライン上で SDIO 割込みが検出されます。

SDIO 割込みが検出されると、SDMMC\_STA[22] (SDIOIT) ビットがセットされます。この静的ビットは、クリアビット SDMMC\_ICR[22] (SDIOITC) によってクリアできます。SDIOIT ステータスビットをセットすると、割り込みを生成できます。個別の割り込みイネーブル SDMMC\_MASK[22] ビット (SDIOITE) は、割り込みリクエストを有効化および無効化するために使用できます。

SD カード割り込みの発生 (SDMMC\_STA[22] ビットがセットされている) に対応するために、ホストソフトウェアは次のステップに従います。

1. SDIOIT ビットをクリアして、SDIOITE 割り込み信号を無効にします (SDMMC\_MASK[22] = '0')。
2. カードの割り込みリクエストを処理し、SD カードの割り込みのソースをクリアします。
3. SDIOITC ビットに '1' を書き込んで、SDIOIT ビットをクリアします (SDMMC\_ICR[22] = '1')。
4. SDIOITE ビットに '1' を書き込んで、SDIOIT 割り込み信号を有効にします (SDMMC\_MASK[22] = '1')。

ステップ 2 から 4 は、SDIO の割り込みサービスルーチンから実行できます。

## 35.7 HW フロー制御

HW フロー制御機能は、FIFO アンダーラン (TX モード) およびオーバーラン (RX モード) エラーを回避するために使用します。

この機能は、SDMMC\_CK を停止し、SDMMC ステートマシンをフリーズします。FIFO を使用してデータを送受信できない間、データ転送がストールされます。SDMMCCLK によってクロック供給されるステートマシンだけが停止し、APB2 インタフェースは動作を続けます。したがって、フロー制御がアクティブであっても、FIFO がフルになったり、エンプティになることがあります。

HW フロー制御を有効にするには、SDMMC\_CLKCR[14] レジスタビットを 1 にセットする必要があります。リセット後、フロー制御は無効になります。

## 35.8 SDMMC レジスタ

デバイスは、APB2 経由でアクセス可能な 32 ビット幅の制御レジスタによってシステムと通信します。

### 35.8.1 SDMMC 電源制御レジスタ (SDMMC\_POWER)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

[1:0] PWRCTRL : 電源制御ビット

これらのビットは、カードクロックの現在の機能状態を定義するために使用されます。

00 : パワーオフ : カードへのクロック供給は停止されます。

01 : 予約済み

10 : 予約済み、パワーアップ

11 : パワーオン : カードにクロックが供給されます。

**注 :** このレジスタへの連続書き込みアクセスをする場合には、7 PCLK2 クロック周期以上が必要です。

**注 :** データ書き込み後、3 SDMMCCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込みません。

## 35.8.2 SDMMC クロック制御レジスタ (SDMMC\_CLKCR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

SDMMC\_CLKCR レジスタは、SDMMC\_CK 出力クロックを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HWFC_EN	NEGEDGE	WIDBUS		BYPASS	PWRSAV	CLKEN	CLKDIV							
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **HWFC\_EN** : HW フロー制御有効

0b : HW フロー制御は無効です。

1b : HW フロー制御は有効です。

HW フロー制御が有効なときの TXFIFOE および RXFIFOE 割込み信号の意味については、[セクション 35.8.11](#) の SDMMC ステータスレジスタの定義を参照してください。

ビット 13 **NEGEDGE** : SDMMC\_CK 反位相選択ビット

0b : コマンドとデータはSDMMCCLK立下りエッジに続くSDMMC\_CK立ち上がりエッジで変更されます。(SDMMC\_CK 立ち上がりエッジはSDMMCCLK 立ち上がりエッジで発生します)。

1b : コマンドとデータはSDMMC\_CK立下りエッジで変更されます。

BYPASS がアクティブな場合、NEGEDGE 値にかかわらず、SDMMCCLK 立ち下がりエッジでデータとコマンドが変更されます。

ビット 12:11 **WIDBUS** : ワイドバスモード有効ビット

00 : デフォルトバスモード : SDMMC\_D0 を使用します。

01 : 4 ワイドバスモード : SDMMC\_D[3:0] を使用します。

10 : 8 ワイドバスモード : SDMMC\_D[7:0] を使用します。

ビット 10 **BYPASS** : クロック分周器バイパス有効ビット

0 : バイパス無効 : SDMMCCLK は、SDMMC\_CK 出力信号を駆動する前に、CLKDIV の値に従って分周されます。

1 : バイパス有効 : SDMMCCLK は、SDMMC\_CK 出力信号を直接駆動します。

ビット 9 **PWRSAV** : 節電構成ビット

節電のため、バスがアイドル状態のときには、PWRSAV をセットして SDMMC\_CK クロック出力を無効にできます。

0 : SDMMC\_CK クロックは常に有効です。

1 : SDMMC\_CK はバスがアクティブなときに限り有効です。

ビット 8 **CLKEN** : クロック有効ビット

0 : SDMMC\_CK は無効です。

1 : SDMMC\_CK は有効です。

ビット 7:0 **CLKDIV** : クロック分周ファクタ

このフィールドは、入力クロック (SDMMCCLK) と出力クロック (SDMMC\_CK) の間の分周ファクタを定義します (SDMMC\_CK 周波数 = SDMMCCLK / [CLKDIV + 2])。

- 注：
- 1 SD/SDIO カードまたは MultiMediaCard が識別モードのときには、SDMMC\_CK 周波数は 400 kHz 未満である必要があります。
  - 2 クロック周波数は、相対カードアドレスがすべてのカードに割り当てられているときには、最大カードバス周波数に変更できます。
  - 3 データ書き込み後、3 SDMMCCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込めません。SDMMC\_CK は、SD I/O カードの読み出しウェイトインターバルの間、停止することができます。この場合、SDMMC\_CLKCR レジスタは SDMMC\_CK を制御しません。

### 35.8.3 SDMMC 引数レジスタ (SDMMC\_ARG)

アドレスオフセット：0x08

リセット値：0x0000 0000

SDMMC\_ARG レジスタは、コマンドメッセージの一部としてカードに送信される 32 ビットのコマンド引数を含みます。

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

ビット 31:0 **CMDARG** : コマンド引数

コマンドメッセージの一部としてカードに送られるコマンド引数です。コマンドが引数を含む場合、コマンドをコマンドレジスタに書き込む前に、このレジスタにロードする必要があります。

## 35.8.4 SDMMC コマンドレジスタ (SDMMC\_CMD)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

SDMMC\_CMD レジスタは、コマンドインデックスおよびコマンドタイプビットを含みます。コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。コマンドタイプビットは、コマンドパスステートマシン (CPSM) を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SDIO SUSPEND	CPSM EN	WAIT PEND	WAIT INT	WAITRESP		CMDINDEX					
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **SDIOSUSPEND** : SD I/O サスペンドコマンド

このビットがセットされると、送信されるコマンドはサスペンドコマンドです (SDIO カードでのみ使用されます)。

ビット 10 **CPSMEN** : コマンドパスステートマシン (CPSM) 有効ビット

このビットがセットされると、CPSM が有効になります。

ビット 9 **WAITPEND** : CPSM のデータ転送終了待ち (CmdPend 内部信号)。

このビットがセットされると、CPSM は、データ転送の終了を待ってから、コマンドの送信を開始します。この機能は、ストリームデータの転送モードが SDMMC\_DCTRL[2] = 1 である場合にのみ使用できます。

ビット 8 **WAITINT** : CPSM の割り込みリクエスト待ち

このビットがセットされると、CPSM は、コマンドタイムアウトを無効にして、割り込みリクエストを待ちます。

ビット 7:6 **WAITRESP** : レスポンスビット待ち

これらは、CPSM がレスポンスを待つかどうか、および、待つ場合にはレスポンスの種類を設定するために使用されます。

00 : レスポンスなし。CMDSENT フラグを待ちます。

01 : ショートレスポンス。CMDREND または CCRCFAIL フラグを待ちます。

10 : レスポンスなし。CMDSENT フラグを待ちます。

11 : ロングレスポンス。CMDREND または CCRCFAIL フラグを待ちます。

ビット 5:0 **CMDINDEX** : コマンドインデックス

コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。

- 注 :
- 1 データ書き込み後、3 SDMMCCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込みません。
  - 2 MultiMediaCards はショートレスポンス (48 ビット長) またはロングレスポンス (136 ビット長) の 2 種類のレスポンスを送信できます。SD カードと SD I/O カードはショートレスポンスのみ送信でき、引数はレスポンスのタイプによって異なることがあります。ソフトウェアは、送信されたコマンドに応じてレスポンスのタイプを区別します。

## 35.8.5 SDMMC コマンドレスポンスレジスタ (SDMMC\_RESPCMD)

アドレスオフセット : 0x10



リセット値 : 0x0000 0000

SDMMC\_RESPCMD レジスタは、最後に受信したコマンドレスポンスのコマンドインデックスフィールドを含みます。コマンドレスポンス送信にコマンドインデックスフィールドが含まれていない場合（ロングまたは OCR レスポンス）、RESPCMD フィールドは未知ですが、11111b（レスポンスの予約フィールドの値）が含まれている必要があります。

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

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **RESPCMD** : レスポンスコマンドインデックス

読み出し専用ビットフィールドです。最後に受信したコマンドレスポンスのコマンドインデックスを含みます。

### 35.8.6 SDMMC レスポンス 1 ~ 4 レジスタ (SDMMC\_RESPx)

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

リセット値 : 0x0000 0000

SDMMC\_RESP1/2/3/4 レジスタは、受信したレスポンスの一部であるカードのステータスを含みます。

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

ビット 31:0 **CARDSTATUSx** : [表 222](#) を参照してください。

カードステータスサイズは、レスポンスのタイプによって 32 または 127 ビットです。

**表 222. レスポンスタイプと SDMMC\_RESPx レジスタ**

レジスタ	ショートレスポンス	ロングレスポンス
SDMMC_RESP1	カードステータス [31:0]	カードステータス [127:96]
SDMMC_RESP2	使用されない	カードステータス [95:64]
SDMMC_RESP3	使用されない	カードステータス [63:32]
SDMMC_RESP4	使用されない	カードステータス [31:1]0b

カードステータスの最上位ビットから先に受信します。SDMMC\_RESP4 レジスタ LSB は常に 0b です。

## 35.8.7 SDMMC データタイムレジスタ (SDMMC\_DTIMER)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

SDMMC\_DTIMER レジスタは、データタイムアウト期間(カードバスクロック周期単位)を含みます。

カウンタは SDMMC\_DTIMER レジスタから値をロードして、データパスステートマシン (DPSM) が Wait\_R またはビジー状態に入ったときに、デクリメントを開始します。DPSM がこのどちらかの状態のときにタイムが 0 に達した場合、タイムアウトステータスフラグがセットされます。

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

ビット 31:0 **DATATIME** : データタイムアウト期間

カードバスクロック周期で表されたデータタイムアウト期間です。

**注 :** データ転送する時は、データ制御レジスタに書き込まれる前に、データタイムレジスタとデータ長レジスタに書き込んでください。

## 35.8.8 SDMMC データ長レジスタ (SDMMC\_DLEN)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

SDMMC\_DLEN レジスタは、転送されるデータバイト数を含みます。値は、データ転送が開始されたときに、データカウンタにロードされます。

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATALENGTH** : データ長の値  
転送されるデータバイト数です。

**注 :** ブロックデータ転送では、データ長レジスタの値はブロックサイズの倍数である必要があります (SDMMC\_DCTRL を参照してください)。データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。

SDMMC マルチバイト転送の場合、データ長レジスタ内の値は 1 から 512 まででなければなりません。

## 35.8.9 SDMMC データ制御レジスタ (SDMMC\_DCTRL)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

SDMMC\_DCTRL レジスタは、データパスステートマシン (DPSM) を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SDIO EN	RW MOD	RW STOP	RW START	DBLOCKSIZE				DMA EN	DT MODE	DTDIR	DTEN
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **SDIOEN** : SD I/O 有効機能  
このビットがセットされると、DPSM は SD I/O カード固有の操作を実行します。

ビット 10 **RWMOD** : 読み出しウェイトモード  
0 : SDMMC\_D2 を停止する読み出しウェイト制御です。  
1 : SDMMC\_CK を使用した読み出しウェイト制御です。

ビット 9 **RWSTOP** : 読み出しウェイト停止  
0 : RWSTART ビットがセットされると、読み出しウェイトが進行中となります。  
1 : RWSTART ビットがセットされると、読み出しウェイト停止が有効になります。

ビット 8 **RWSTART** : 読み出しウェイト開始

このビットがセットされると、読み出しウェイト操作が開始されます。

ビット 7:4 **DBLOCKSIZE** : データブロックサイズ

ブロックデータ転送モードが選択されているときのデータブロック長を定義します。

0000 : (0 10 進) ロック長 =  $2^0$  = 1 バイト  
0001 : (1 10 進) ロック長 =  $2^1$  = 2 バイト  
0010 : (2 10 進) ロック長 =  $2^2$  = 4 バイト  
0011 : (3 10 進) ロック長 =  $2^3$  = 8 バイト  
0100 : (4 10 進) ロック長 =  $2^4$  = 16 バイト  
0101 : (5 10 進) ロック長 =  $2^5$  = 32 バイト  
0110 : (6 10 進) ロック長 =  $2^6$  = 64 バイト  
0111 : (7 10 進) ロック長 =  $2^7$  = 128 バイト  
1000 : (8 10 進) ロック長 =  $2^8$  = 256 バイト  
1001 : (9 10 進) ロック長 =  $2^9$  = 512 バイト  
1010 : (10 10 進) ロック長 =  $2^{10}$  = 1024 バイト  
1011 : (11 10 進) ロック長 =  $2^{11}$  = 2048 バイト  
1100 : (12 10 進) ロック長 =  $2^{12}$  = 4096 バイト  
1101 : (13 10 進) ロック長 =  $2^{13}$  = 8192 バイト  
1110 : (14 10 進) ロック長 =  $2^{14}$  = 16384 バイト  
1111 : (15 10 進) 予約済み

ビット 3 **DMAEN** : DMA 有効

0 : DMA は無効です。  
1 : DMA は有効です。

ビット 2 **DTMODE** : データ転送モード選択 1 : ストリームまたは SDIO マルチバイトデータ転送

0 : ブロックデータ転送です。  
1 : ストリームまたは SDIO マルチバイトデータ転送です。

ビット 1 **DTDIR** : データ転送方向選択

0 : コントローラからカードへ  
1 : カードからコントローラへ

[0] **DTEN** : データ転送有効ビット

データ転送は、1b が DTEN ビットに書き込まれると開始します。方向ビット DTDIR に応じて、DPSM は Wait\_S、Wait\_R、または転送の始めにすぐに RW Start がセットされた場合には ReadWait へ移行します。データ転送の終わりに有効ビットをクリアする必要はありませんが、新しいデータ転送を有効にするには、SDMMC\_DCTRL を更新する必要があります。

**注 :** データ書き込み後、3 SDMMCCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込みません。

**DTMODE** ビットの意味は、**SDIOEN** ビットの値によって変化します。**SDIOEN=0** および **DTMODE=1** の場合、MultiMediaCard ストリームモードが有効になり、**SDIOEN=1** および **DTMODE=1** の場合、ペリフェラルが SDIO マルチバイト転送を有効にします。

## 35.8.10 SDMMC データカウンタレジスタ (SDMMC\_DCOUNT)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

SDMMC\_DCOUNT レジスタは、DPSM がアイドル状態から Wait\_R または Wait\_S 状態へ移行すると、データ長レジスタから値をロードします (SDMMC\_DLEN を参照してください)。データが転送されると、カウンタは 0 に達するまで値をデクリメントします。その後、DPSM はアイドル状態へ移行し、データステータス終了フラグ DATAEND がセットされます。

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATACOUNT** : データカウント値

このビットを読み出すと、残りの転送バイト数が返されます。書き込み操作は何の影響も与えません。

**注 :** このレジスタは、データ転送が完了したときだけ読み出してください。

## 35.8.11 SDMMC ステータスレジスタ (SDMMC\_STA)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

SDMMC\_STA レジスタは、読み出し専用レジスタです。2 種類のフラグを含みます。

- **スタティックフラグ** (ビット [23:22,10:0]) : これらのビットは、SDMMC 割り込みクリアレジスタ (SDMMC\_ICR を参照) に書き込んでクリアされるまで、アサートされたままです。
- **ダイナミックフラグ** (ビット [21:11]) : これらのビットは、基礎回路の状態に応じて状態が変わります (たとえば、FIFO フルおよびエンプティフラグは、データが FIFO に書き込まれるとアサートおよびネゲートされます)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOIT	RXD AVL	TXD AVL	RX FIFOE	TX FIFOE	RX FIFOF	TX FIFOF
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HF	TX FIFO HE	RXACT	TXACT	CMD ACT	DBCK END	Res.	DATA END	CMDSE NT	CMDRE ND	RX OVERR	TXUNDE RR	DTIMEO UT	CTIMEO UT	DCRC FAIL	CCRC FAIL
r	r	r	r	r	r		r	r	r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SDIOIT** : SDIO 割り込みが受信されました。

ビット 21 **RXDAVL** : 受信 FIFO にデータがあります。

ビット 20 **TXDAVL** : 送信 FIFO にデータがあります。

ビット 19 **RXFIFOE** : 受信 FIFO エンプティ

- ビット 18 **TXFIFOE** : 送信 FIFO エンプティ  
HW フロー制御が有効なとき、FIFO に 2 ワードが含まれると、TXFIFOE 信号がアクティブになります。
- ビット 17 **RXFIFOF** : 受信 FIFO がフルです。  
HW フロー制御が有効なとき、FIFO がフルになる 2 ワード前に RXFIFOF 信号がアクティブになります。
- ビット 16 **TXFIFOF** : 送信 FIFO がフルです。
- ビット 15 **RXFIFOHF** : 受信 FIFO がハーフフル状態です。FIFO には少なくとも 8 ワード入っています。
- ビット 14 **TXFIFOHE** : 送信 FIFO がハーフエンプティ状態です。FIFO には少なくとも 8 ワード書き込み可能です。
- ビット 13 **RXACT** : データ受信中です。
- ビット 12 **TXACT** : データ送信中です。
- ビット 11 **CMDACT** : コマンド転送中です。
- ビット 10 **DBCKEND** : データブロック送受信 (CRC 確認成功) しました。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **DATAEND** : データの終わり (データカウンタ SDIDCOUNT がゼロ) です。
- ビット 7 **CMDSENT** : コマンドが送信されました (レスポンス不要)。
- ビット 6 **CMDREND** : コマンドレスポンスが受信されました (CRC 確認成功)。
- ビット 5 **RXOVERR** : FIFO オーバーランエラーを受信しました。  
**注:** *SDMMC FIFO を読み出すために DMA が使用される場合 (SDMMC\_DCTRL レジスタの DMAEN ビットがセットされている場合)、ユーザソフトウェアでは DMA ストリームを無効にし、'0' を書き込む必要があります (DMA リクエストの生成を無効にするため)。*
- ビット 4 **TXUNDERR** : 送信 FIFO アンダーランエラーです。  
**注:** *SDMMC FIFO に書き込むために DMA が使用される場合 (SDMMC\_DCTRL レジスタの DMAEN ビットがセットされている場合)、ユーザソフトウェアでは DMA ストリームを無効にし、DMAEN に '0' を書き込む必要があります (DMA リクエストの生成を無効にするため)。*
- ビット 3 **DTIMEOUT** : データタイムアウトです。
- ビット 2 **CTIMEOUT** : コマンドレスポンスタイムアウトです。  
コマンドタイムアウト期間は、64 SDMMC\_CLK クロック周期固定です。
- ビット 1 **DCRCFAIL** : データブロック送受信 (CRC 確認失敗) しました。
- ビット 0 **CCRCFAIL** : コマンドレスポンス受信 (CRC 確認失敗) しました。

## 35.8.12 SDMMC 割り込みクリアレジスタ (SDMMC\_ICR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

SDMMC\_ICR レジスタは、書き込み専用レジスタです。ビットに 1b を書き込むと、SDMMC\_STA ステータスレジスタの対応するビットがクリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIO ITC	Res.	Res.	Res.	Res.	Res.	Res.
									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DBCK ENDC	Res.	DATA ENDC	CMD SENTIC	CMD RENDC	RX OVERR C	TX UNDERR C	DTIMEO UTC	CTIMEO UTC	DCRC FAILC	CCRC FAILC
					rw		rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SDIOITC** : SDIOIT フラグクリアビット

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

0 : SDIOIT はクリアされません。

1 : SDIOIT はクリアされます。

ビット 21:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DBCKENDC** : DBCKEND フラグクリアビット

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

0 : DBCKEND はクリアされません。

1 : DBCKEND はクリアされます。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **DATAENDC** : DATAEND フラグクリアビット

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

0 : DATAEND はクリアされません。

1 : DATAEND はクリアされます。

ビット 7 **CMDSENTC** : CMDSENT フラグクリアビット

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

0 : CMDSENT はクリアされません。

1 : CMDSENT はクリアされます。

ビット 6 **CMDRENDC** : CMDREND フラグクリアビット

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

0 : CMDREND はクリアされません。

1 : CMDREND はクリアされます。

ビット 5 **RXOVERRC** : RXOVERR フラグクリアビット

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

0 : RXOVERR はクリアされません。

1 : RXOVERR はクリアされます。

ビット 4 **TXUNDERRC** : TXUNDERR フラグクリアビット

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

0 : TXUNDERR はクリアされません。

1 : TXUNDERR はクリアされます。

- ビット 3 **DTIMEOUTC** : DTIMEOUT フラグクリアビット  
DTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。  
0 : DTIMEOUT はクリアされません。  
1 : DTIMEOUT はクリアされます。
- ビット 2 **CTIMEOUTC** : CTIMEOUT フラグクリアビット  
CTIMEOUT フラグをクリアするために、ソフトウェアによってセットされます。  
0 : CTIMEOUT はクリアされません。  
1 : CTIMEOUT はクリアされます。
- ビット 1 **DCRCFAILC** : DCRCFAIL フラグクリアビット  
DCRCFAIL フラグをクリアするために、ソフトウェアによってセットされます。  
0 : DCRCFAIL はクリアされません。  
1 : DCRCFAIL はクリアされます。
- ビット 0 **CCRCFAILC** : CCRCFAIL フラグクリアビット  
CCRCFAIL フラグをクリアするために、ソフトウェアによってセットされます。  
0 : CCRCFAIL はクリアされません。  
1 : CCRCFAIL はクリアされます。

### 35.8.13 SDMMC マスクレジスタ (SDMMC\_MASK)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

割込みマスクレジスタは、対応するビットを 1b にセットすることによって、割込みリクエストを生成するステータスフラグを決めます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIO ITIE	RXD AVLIE	TXD AVLIE	RX FIFO EIE	TX FIFO EIE	RX FIFO FIE	TX FIFO FIE
									r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HFIE	TX FIFO HEIE	RX ACTIE	TX ACTIE	CMD ACTIE	DBCK ENDIE	Res.	DATA ENDIE	CMD SENT IE	CMD REND IE	RX OVERR IE	TX UNDERR IE	DTIMEO UTIE	CTIMEO UTIE	DCRC FAILIE	CCRC FAILIE
r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **SDIOITIE** : SDIO モード割込み受信割込み有効

SDIO モード割込みを受信したときに生成される割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : SDIO モード割込み受信割込みは無効です。

1 : SDIO モード割込み受信割込みは有効です。

ビット 21 **RXDAVLIE** : Rx FIFO データ有無割込み有効

Rx FIFO のデータの有無によって生成される割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : Rx FIFO のデータの有無による割込みは無効です。

1 : Rx FIFO のデータの有無による割込みは有効です。



- ビット 20 **TXDAVLIE** : Tx FIFO データ有無割込み有効  
Tx FIFO のデータの有無によって生成される割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Tx FIFO のデータの有無による割込みは無効です。  
1 : Tx FIFO のデータの有無による割込みは有効です。
- ビット 19 **RXFIFOEIE** : Rx FIFO エンプティ割込み有効  
Rx FIFO エンプティによる割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Rx FIFO エンプティ割込みは無効です。  
1 : Rx FIFO エンプティ割込みは有効です。
- ビット 18 **TXFIFOEIE** : Tx FIFO エンプティ割込み有効  
Tx FIFO エンプティによる割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Tx FIFO エンプティ割込みは無効です。  
1 : Tx FIFO エンプティ割込みは有効です。
- ビット 17 **RXFIFOFIE** : Rx FIFO フル割込み有効  
Rx FIFO がフルになったときに生成される割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Rx FIFO フル割込みは無効です。  
1 : Rx FIFO フル割込みは有効です。
- ビット 16 **TXFIFOFIE** : Tx FIFO フル割込み有効  
Tx FIFO がフルになったときに生成される割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Tx FIFO フル割込みは無効です。  
1 : Tx FIFO フル割込みは有効です。
- ビット 15 **RXFIFOHFIE** : Rx FIFO ハーフフル割込み有効  
Rx FIFO がハーフフル状態になったときに生成される割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Rx FIFO ハーフフル割込みは無効です。  
1 : Rx FIFO ハーフフル割込みは有効です。
- ビット 14 **TXFIFOHEIE** : Tx FIFO ハーフエンプティ割込み有効  
Tx FIFO がハーフエンプティ状態になったときに生成される割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Tx FIFO ハーフエンプティ割込みは無効です。  
1 : Tx FIFO ハーフエンプティ割込みは有効です。
- ビット 13 **RXACTIE** : データ受信割込み有効  
データ受信中の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : データ受信割込みは無効です。  
1 : データ受信割込みは有効です。
- ビット 12 **TXACTIE** : データ送信割込み有効  
データ送信中の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : データ送信割込みは無効です。  
1 : データ送信割込みは有効です。
- ビット 11 **CMDACTIE** : コマンド動作中割込み有効  
コマンド転送中の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : コマンド動作中割込みは無効です。  
1 : コマンド動作中割込みは有効です。

- ビット 10 **DBCKENDIE** : データブロックエンド割込み有効  
データブロックエンドによる割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : データブロックエンド割込みは無効です。  
1 : データブロックエンド割込みは有効です。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **DATAENDIE** : データエンド割込み有効  
データエンドによる割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : データエンド割込みは無効です。  
1 : データエンド割込みは有効です。
- ビット 7 **CMDSENTIE** : コマンド送信割込み有効  
コマンド送信による割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : コマンド送信割込みは無効です。  
1 : コマンド送信割込みは有効です。
- ビット 6 **CMDRENDIE** : コマンドレスポンス受信割込み有効  
コマンドレスポンスの受信による割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : コマンドレスポンス受信割込みは無効です。  
1 : コマンドレスポンス受信割込みは有効です。
- ビット 5 **RXOVERRIE** : Rx FIFO オーバーランエラー割込み有効  
Rx FIFO オーバーランエラーによる割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Rx FIFO オーバーランエラー割込みは無効です。  
1 : Rx FIFO オーバーランエラー割込みは有効です。
- ビット 4 **TXUNDERRIE** : Tx FIFO アンダーランエラー割込み有効  
Tx FIFO アンダーランエラーによる割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : Tx FIFO アンダーランエラー割込みは無効です。  
1 : Tx FIFO アンダーランエラー割込みは有効です。
- ビット 3 **DTIMEOUTIE** : データタイムアウト割込み有効  
データタイムアウトによる割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : データタイムアウト割込みは無効です。  
1 : データタイムアウト割込みは有効です。
- ビット 2 **CTIMEOUTIE** : コマンドタイムアウト割込み有効  
コマンドタイムアウトによる割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : コマンドタイムアウト割込みは無効です。  
1 : コマンドタイムアウト割込みは有効です。
- ビット 1 **DCRCFAILIE** : データ CRC 失敗割込み有効  
データ CRC 失敗による割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : データ CRC 失敗割込みは無効です。  
1 : データ CRC 失敗割込みは有効です。

ビット 0 **CCRCFAILIE** : コマンド CRC 失敗割込み有効

コマンド CRC 失敗による割込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : コマンド CRC 失敗割込みは無効です。

1 : コマンド CRC 失敗割込みは有効です。

## 35.8.14 SDMMC FIFO カウンタレジスタ (SDMMC\_FIFOCNT)

アドレスオフセット : 0x48

リセット値 : 0x0000 0000

SDMMC\_FIFOCNT レジスタは、FIFO に書き込まれるか、FIFO から読み出される残りのワード数を含みます。FIFO カウンタは、データ制御レジスタ (SDMMC\_DCTRL レジスタ) のデータ転送有効ビット DTEN がセットされていて、DPSM がアイドル状態のときに、データ長レジスタ (SDMMC\_DLEN を参照) から値をロードします。データ長がワード境界 (4 の倍数) に合っていない場合、残りの 1~3 バイトは 1 ワードとみなされます。

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

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **FIFOCOUNT** : FIFO に書き込まれるか、FIFO から読み出される残りのワード数

## 35.8.15 SDMMC データ FIFO レジスタ (SDMMC\_FIFO)

アドレスオフセット : 0x80

リセット値 : 0x0000 0000

受信および送信 FIFO は、32 ビット幅のレジスタとして読み書きできます。FIFO は、32 の連続したアドレスに 32 個のエントリを含みます。このため、CPUはソフトウェアによるFIFOへの読み書きを行うことができます。

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

ビット 31:0 **FIFOData** : 受信および送信 FIFO データ

FIFO データは、次のアドレスから 32 ビットワードの 32 個のエントリを占有します。

SDMMC ベース+ 0x080~SDMMC ベース+ 0xFC

## 35.8.16 SDMMC レジスタマップ

次の表に SDMMC レジスタの一覧を示します。

表 223. SDMMC レジスタマップ

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SDMMC_POWER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWRCTRL
	リセット値																																0
0x04	SDMMC_CLKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	SDMMC_ARG	CMDARG																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	SDMMC_CMD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																							0	0	0	0	0	0	0	0	0	0
0x10	SDMMC_RESPCMD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x14	SDMMC_RESP1	CARDSTATUS1																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SDMMC_RESP2	CARDSTATUS2																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	SDMMC_RESP3	CARDSTATUS3																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	SDMMC_RESP4	CARDSTATUS4																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	SDMMC_DTIMER	DATATIME																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	SDMMC_DLEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x2C	SDMMC_DCTRL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																							0	0	0	0	0	0	0	0	0	0



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

表 223. SDMMC レジスタマップ (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	SDMMC_DCOUNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT																								
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	SDMMC_STA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOIT	RXDAVL	TXDAVL	RXFIOE	TXFIOE	RXFIOF	TXFIOF	RXFIOHF	TXFIOHE	RXACT	TXACT	CMDACT	DBCKEND	Res.	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	SDMMC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOITC	Res.	Res.	Res.	Res.	Res.	DBCKENDC	Res.	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC						
	リセット値										0												0		0	0	0	0	0	0	0	0	0
0x3C	SDMMC_MASK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIOEIE	TXFIOEIE	RXFIOFIE	TXFIOFIE	RXFIOHFIE	TXFIOHEIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	Res.	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	SDMMC_FIFOCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFOCOUNT																							
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	SDMMC_FIFO	FIFOData																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 36 コントローラエリアネットワーク (bxCAN)

### 36.1 概要

**Basic Extended CAN** ペリフェラル (略称 **bxCAN**) は CAN ネットワークのインターフェースです。CAN プロトコルバージョン 2.0A および B をサポートしています。小さな CPU 負荷で多くの受信メッセージを効率よく取り扱えるように設計されています。また、送信メッセージの優先順位に関する要件も満たしています。

安全性が重視されるアプリケーションの場合、CAN コントローラは、CAN タイムトリガ通信オプションをサポートするすべてのハードウェア機能を提供します。

### 36.2 bxCAN の主な機能

- CAN プロトコルバージョン 2.0 A、B Active のサポート
- ビットレート：最大 1 Mbit/s
- タイムトリガ通信オプションのサポート

#### 送信

- 3 つの送信メールボックス
- 設定可能な送信優先順位
- SOF 送信時のタイムスタンプ

#### 受信

- 2 つの 3 ステージ受信 FIFO
- スケーラブルフィルタバンク：
  - 28 個のフィルタバンクを CAN1 と CAN2 で共有
- 識別子リスト機能
- 設定可能な FIFO オーバーラン
- SOF 受信時のタイムスタンプ

#### タイムトリガ通信オプション

- 自動再送信モードの無効化
- 16 ビットのフリーランニングタイマ
- タイムスタンプを最後の 2 データバイトで送信

#### 管理

- マスク可能な割り込み
- 固有のアドレス空間でのソフトウェア効率の高いメールボックス配置

#### デュアル CAN

- CAN1：スレーブ bxCAN と 512 バイトの SRAM メモリ間の通信を管理するマスタ bxCAN
- CAN2：スレーブ bxCAN。SRAM メモリへの直接アクセスなし。
- この 2 つの bxCAN セルは、512 バイトの SRAM メモリを共有します ([図 422：デュアル CAN ブロック図](#)を参照)。

## 36.3 bxCAN の概要

今日の CAN アプリケーションでは、ネットワークのノード数が増加し、複数のネットワークがゲートウェイを介してリンクされることもあります。一般に、システム内の（そして各ノードが扱う）メッセージ数は大幅に増加しています。アプリケーションメッセージに加えて、ネットワーク管理や診断のメッセージも導入されています。

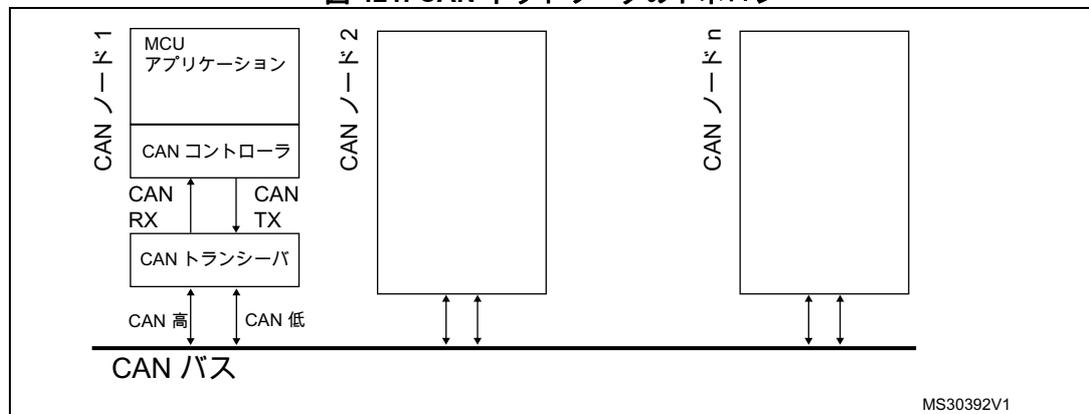
- 各種のメッセージを扱うには、高度なフィルタ機構が要求されます。

さらに、アプリケーションタスクはより多くの CPU 時間を必要とするため、メッセージ受信に起因するリアルタイム制約を減らす必要があります。

- 受信 FIFO 方式では、CPU は、メッセージを失うことなく長時間にわたってアプリケーションタスクに専念できます。

標準の CAN ドライバに基づく標準 HLP (Higher Layer Protocol : 上位層プロトコル) は、CAN コントローラとの効率的なインタフェースを必要とします。

図 421. CAN ネットワークのトポロジ



### 36.3.1 CAN 2.0B アクティブコア

bxCAN モジュールは、CAN メッセージの送受信を完全に自律的に処理します。標準識別子 (11 ビット) と拡張識別子 (29 ビット) は、ハードウェアによって完全にサポートされています。

### 36.3.2 制御レジスタ、ステータスレジスタ、設定レジスタ

アプリケーションは、これらのレジスタを以下の目的で使用します。

- CAN パラメータ (ポーレートなど) の設定
- 送信リクエスト
- 受信処理
- 割り込み管理
- 診断情報の入手

### 36.3.3 送信メールボックス

ソフトウェアがメッセージをセットアップするために、3 つの送信メールボックスが用意されています。最初に送信されるメールボックスは、送信スケジューラによって決定されます。

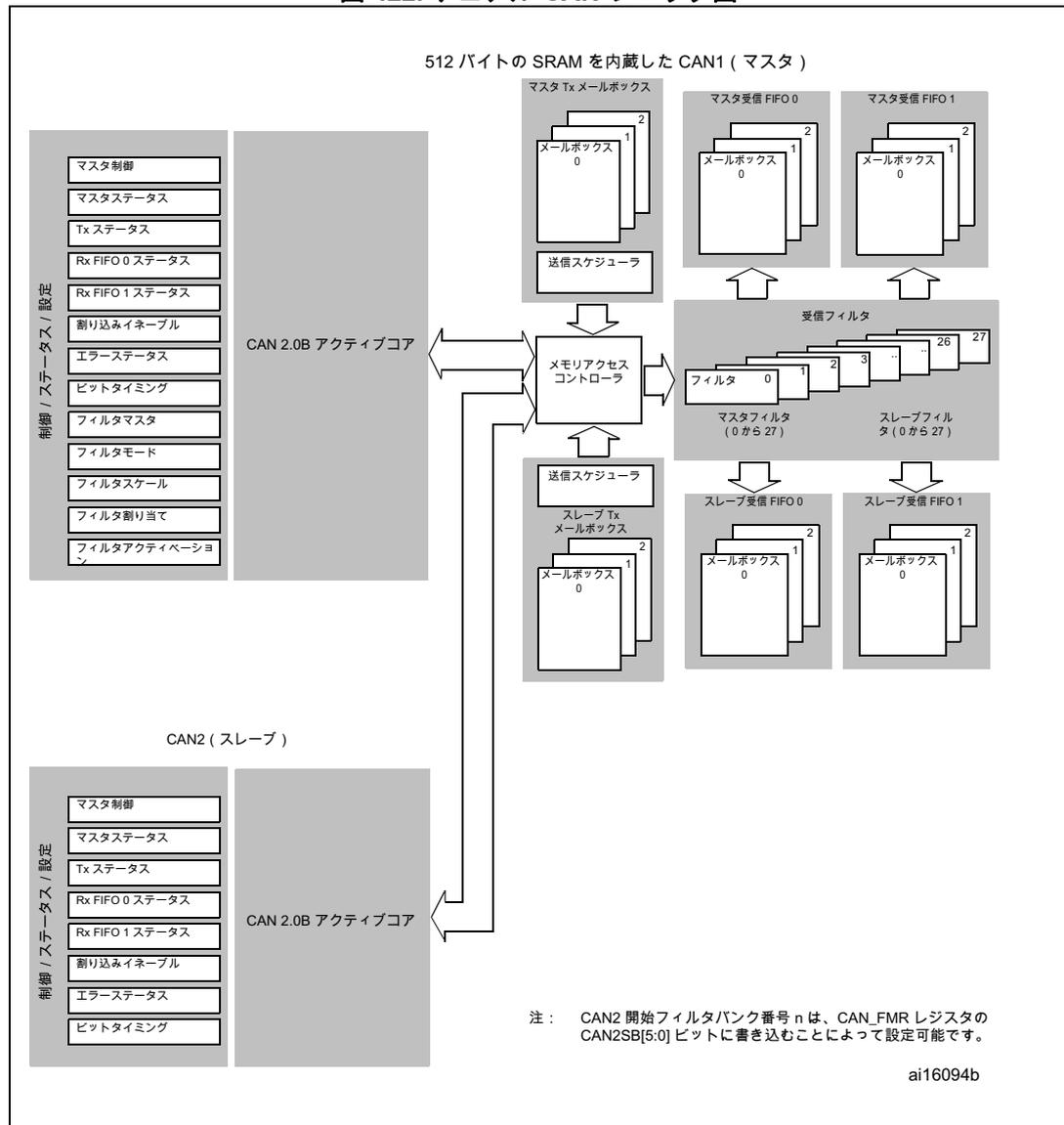
## 36.3.4 受信フィルタ

bxCAN は、受信メッセージからソフトウェアが必要なものを選び出し、他を破棄するために、28 個のスケラブルで設定可能な識別子フィルタバンクを備えています。

### 受信 FIFO

ハードウェアは、受信メッセージを格納するために 2 つの受信 FIFO を使用します。各 FIFO には、3 つの完全なメッセージを格納できます。FIFO の管理はすべてハードウェアによって行われます。

図 422. デュアル CAN ブロック図



## 36.4 bxCAN 動作モード

bxCAN には、**初期化**、**通常**、および**スリープ**という 3 つの主な動作モードがあります。ハードウェアリセットの後、bxCAN は消費電力を低減するためにスリープモードに入り、CANTX では内部プルアップがアクティブとなります。ソフトウェアは、CAN\_MCR レジスタの INRQ または SLEEP ビットをセットすることによって、bxCAN に**初期化**または**スリープ**モードに入るように要求します。いずれかのモードに入ると、bxCAN は、CAN\_MSR レジスタの INAK または SLAK ビットをセットすることによってそれを確定し、内部プルアップは無効になります。INAK も SLAK もセットされていない場合、bxCAN は**通常**モードです。**通常**モードに入る前に、bxCAN は必ず CAN バスで**同期**する必要があります。同期のために、bxCAN は、CAN バスがアイドル状態になる（つまり、CANRX 上で 11 個の連続したリセッシブビットが観測される）のを待ちます。

### 36.4.1 初期化モード

ハードウェアが初期化モードにある間に、ソフトウェアの初期化も行うことができます。このモードに入るには、ソフトウェアは CAN\_MCR レジスタの INRQ ビットをセットし、ハードウェアが CAN\_MSR レジスタの INAK ビットをセットしてこのリクエストを確認するのを待ちます。

初期化モードから出るには、ソフトウェアで INRQ ビットをクリアします。INAK ビットがハードウェアによってクリアされると、bxCAN は初期化モードを終了します。

初期化モードでは、CAN バスとの間のすべてのメッセージ転送は停止し、CAN バス出力 CANTX のステータスはリセッシブ（ハイ）になります。

初期化モードに入っても、設定レジスタの内容は一切変化しません。

CAN コントローラを初期化するには、ソフトウェアでビットタイミング (CAN\_BTR) および CAN オプション (CAN\_MCR) レジスタの設定を行う必要があります。

CAN フィルタバンクに関連付けられたレジスタ (モード、スケール、FIFO 割当て、アクティベーション、およびフィルタ値) を初期化するには、ソフトウェアが FINIT ビット (CAN\_FMR) をセットする必要があります。フィルタの初期化は、初期化モード以外でも行うことができます。

**注:** **FINIT=1 のとき、CAN 受信は無効化されます。**

**フィルタの値は、CAN\_FA1R レジスタの関連するフィルタアクティベーションビットを無効にすることによっても変更できます。**

**フィルタバンクを使用しない場合は、これを非アクティブのままにしておく (対応する FACT ビットをクリアしておく) ことを推奨します。**

### 36.4.2 通常モード

初期化が終了すると、ソフトウェアはハードウェアに対し、CAN バスで同期して送受信を開始できるように通常モードに入るように要求する必要があります。

この通常モードに入るリクエストは、CAN\_MCR レジスタの INRQ ビットをクリアすることによって行われます。bxCAN は通常モードに入り、11 個の連続したリセッシブビットの発生 (バスアイドル状態) を待って CAN バス上のデータ転送と同期すると、バス動作への参加準備完了となります。通常モードへの切り替えは、CAN\_MSR レジスタの INAK ビットをクリアすることによって、ハードウェアによって確認されます。

フィルタ値の初期化は初期化モードとは別に行いますが、フィルタがアクティブでない (対応する FACTx ビットがクリアされている) ときに行う必要があります。フィルタのスケールとモードの設定は、通常モードに入る前に行う必要があります。

## 36.4.3 スリープモード (低電力)

消費電力を低減するため、bxCAN にはスリープモードと呼ばれる低電力モードがあります。ソフトウェアのリクエストで CAN\_MCR レジスタの SLEEP ビットをセットすると、このモードに入ります。このモードでは bxCAN のクロックは停止しますが、ソフトウェアから bxCAN のメールボックスにアクセスすることは可能です。

bxCAN がスリープモードのときにソフトウェアが INRQ ビットをセットして初期化モードに入るように要求する場合には、SLEEP ビットもクリアする必要があります。

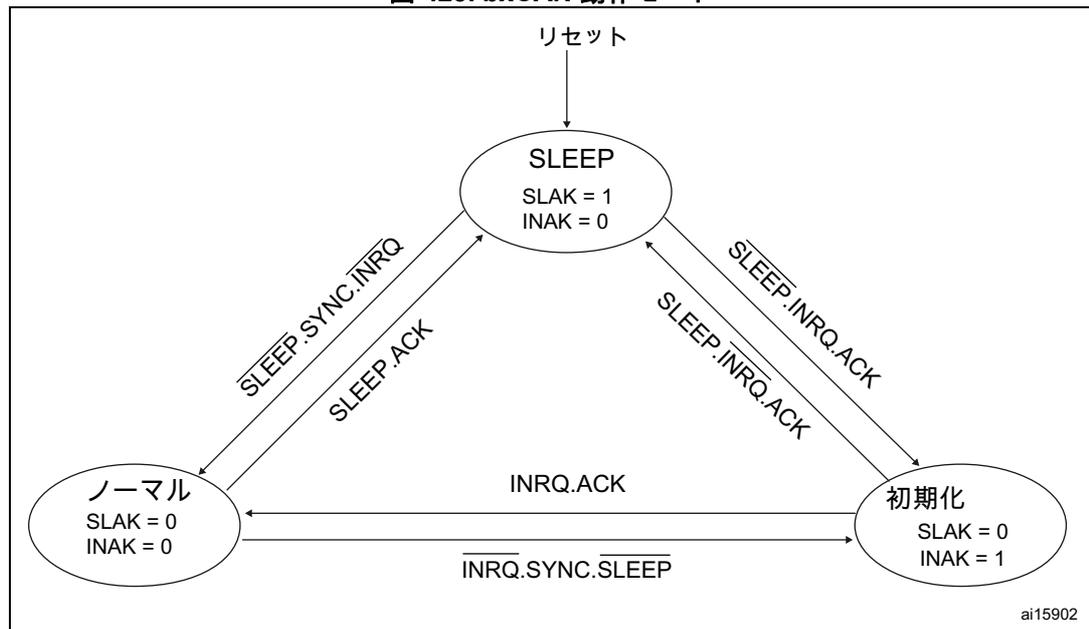
bxCAN がスリープモードからウェイクアップするには、ソフトウェアで SLEEP ビットをクリアするか、CAN バス動作を検出する必要があります。

CAN バス動作を検出すると、ハードウェアは、SLEEP ビットをクリアしてウェイクアップシーケンスを自動的に実行します (CAN\_MCR レジスタの AWUM ビットがセットされている場合)。AWUM ビットがクリアされている場合、スリープモードから出るにはウェイクアップ割込み発生時にソフトウェアが SLEEP ビットをクリアする必要があります。

**注:** ウェイクアップ割込みが有効である (CAN\_IER レジスタの WKUIE ビットがセットされている) 場合には、bxCAN がウェイクアップシーケンスを自動的に実行する場合でも、CAN バス動作の検出時にウェイクアップ割込みが生成されます。

SLEEP ビットがクリアされた後、bxCAN が CAN バスに同期するとスリープモードを終了します (図 423: bxCAN 動作モードを参照)。SLAK ビットがハードウェアによってクリアされると、スリープモードを終了します。

図 423. bxCAN 動作モード



1. ACK = ハードウェアが CAN\_MSR レジスタの INAK または SLAK ビットをセットすることによってリクエストを確認する待ち状態。
2. SYNC = CAN バスがアイドル状態になる (つまり、11 個の連続したリセッピットが CANRX で観測される) までの bxCAN の待ち状態。

## 36.5 テストモード

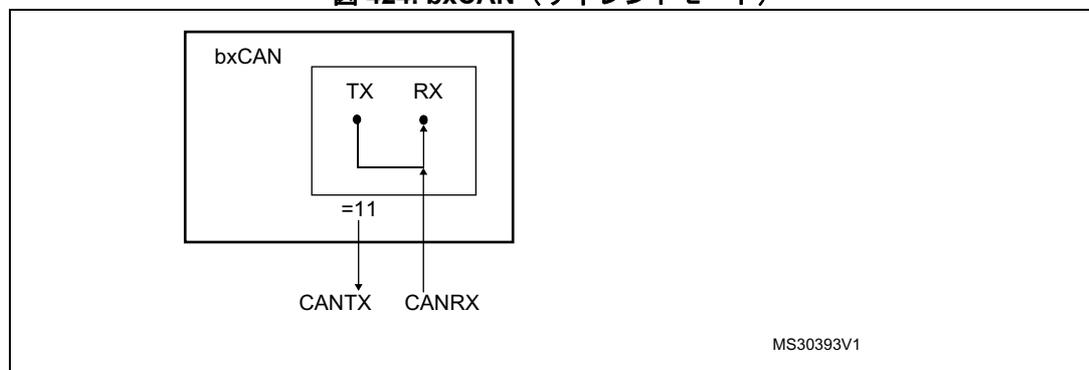
テストモードは、CAN\_BTR レジスタの SILM ビットと LBKM ビットによって選択できます。これらのビットは、bxCAN が初期化モードにある間に設定する必要があります。テストモードが選択された後で通常モードに戻るには、CAN\_MCR レジスタの INRQ ビットをリセットする必要があります。

### 36.5.1 サイレントモード

bxCAN をサイレントモードにするには、CAN\_BTR レジスタの SILM ビットをセットします。

サイレントモードでは、bxCAN は有効なデータフレームやリモートフレームを受信できますが、CAN バスではリセッシブビットのみを送信するため、送信を開始することはできません。bxCAN がドミナントビット (ACK ビット、オーバーロードフラグ、アクティブエラーフラグ) を送信する必要がある場合、CAN バスがリセッシブ状態にとどまることがあっても CAN コアがこのドミナントビットを監視できるように、このビットは内部でルート変更されます。サイレントモードを使用すると、ドミナントビット (ACK ビット、エラーフレーム) の送信による影響を受けずに、CAN バス上のトラフィックを解析できます。

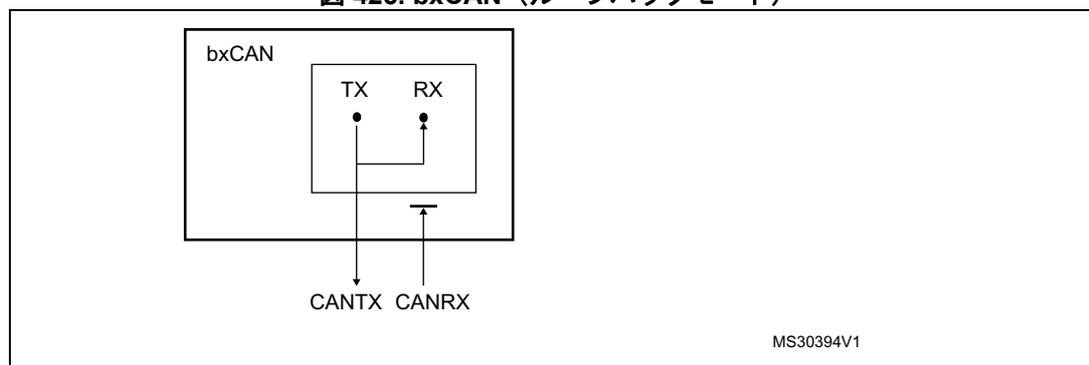
図 424. bxCAN (サイレントモード)



### 36.5.2 ループバックモード

bxCAN をループバックモードに設定するには、CAN\_BTR レジスタの LBKM ビットをセットします。ループバックモードでは、bxCAN は自分が送信したメッセージを受信メッセージとして扱い、それらを受信メールボックスに格納します (受信フィルタを通過した場合)。

図 425. bxCAN (ループバックモード)



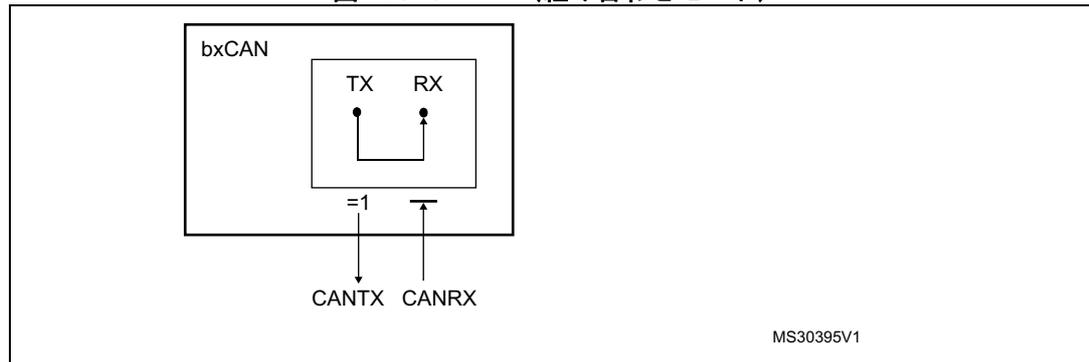
このモードは、セルフテスト機能のために用意されています。外部イベントから独立させるために、ループバックモードでは、CAN コアは ACK エラーを無視します (データ/リモートフレームの ACK スロットではドミナントビットはサンプリングされません)。このモードでは、bxCAN は Tx 出力が

ら Rx 入力へ内部フィードバックを行います。bxCAN は、CANRX 入力ピンの実際の値を無視します。送信メッセージは CANTX ピンで監視できます。

### 36.5.3 サイレントモードとループバックモードの組み合わせ

CAN\_BTR レジスタの LBKM ビットと SILM ビットをセットすることによって、ループバックモードとサイレントモードを組み合わせることもできます。このモードは「ホットセルフテスト」に使用できます。つまり、CANTX および CANRX ピンに接続されている実行中の CAN システムに影響を与えずに、ループバックモードと同様に bxCAN をテストできます。このモードでは、CANRX ピンは bxCAN から切断され、CANTX ピンはリセッショに保持されます。

図 426. bxCAN (組み合わせモード)



## 36.6 デバッグモードでの挙動

マイクロコントローラがデバッグモードになると (Cortex<sup>®</sup>-M7 コア停止)、次の状態に応じて、bxCAN は通常どおりに動作を続けるか、または停止します。

- DBG モジュールで CAN1 の場合は DBG\_CAN1\_STOP ビット、CAN2 の場合は DBG\_CAN2\_STOP ビット。
- CAN\_MCR の DBF ビット。詳細については、[セクション 36.9.2 : CAN 制御/ステータスレジスタ](#)を参照してください。

## 36.7 bxCAN 機能の説明

### 36.7.1 送信処理

メッセージを送信するには、アプリケーションは、**EMPTY** 状態の送信メールボックスを 1 つ選択し、識別子、データ長コード (DLC)、およびデータを設定してから、CAN\_TlRxR レジスタの対応する TXRQ ビットをセットすることによって、送信を要求する必要があります。メールボックスが **EMPTY** 状態でなくなると、ソフトウェアはこのメールボックスレジスタへの書き込みができなくなります。TXRQ ビットがセットされると、メールボックスはただちに **PENDING** 状態になり、最高優先順位のメールボックスになるまで待機します。[送信優先順位](#)を参照してください。メールボックスの優先順位が最高になると、メールボックスは送信の **SCHEDULED** 状態になります。スケジュール済みメールボックスのメッセージの送信は、CAN バスがアイドル状態になると開始されます (**TRANSMIT** 状態になります)。メールボックスの内容が正常に送信されると、そのメールボックスは再び **EMPTY** 状態になります。ハードウェアは、CAN\_TSR レジスタの RQCP ビットと TXOK ビットをセットして、送信が正常に終了したことを示します。

送信が失敗した場合、その原因がアービトレーションロストである場合は CAN\_TSR レジスタの ALST ビットによって、または、送信エラー検出である場合は TERR ビットによって示されます。

## 送信優先順位

識別子による優先順位

複数の送信メールボックスが保留中のときには、送信順序は、メールボックスに格納されているメッセージの識別子によって決まります。CAN プロトコルのアービトレーションに従って、識別子の値が最小のメッセージの優先順位が最高になります。識別子の値が同じ場合には、メールボックス番号の小さい方が優先的にスケジュールされます。

送信リクエストの順序による優先順位

送信メールボックスは、CAN\_MCR レジスタの TXFP ビットをセットすることによって、送信 FIFO として設定できます。このモードでは、優先順位は送信リクエストの順序によって決まります。

このモードは分割送信の場合に非常に役立ちます。

## アポート

ユーザは、CAN\_TSR レジスタの ABRQ ビットをセットすることによって、送信リクエストをアポートできます。**PENDING** または **SCHEDULED** 状態では、メールボックスはただちにアポートされません。メールボックスが **TRANSMIT** 状態のときにアポートリクエストが出されると、次の 2 つの結果につながります。メールボックスの送信が正常に行われた場合、CAN\_TSR レジスタの TXOK ビットがセットされ、メールボックスは **EMPTY** 状態になります。送信に失敗した場合、メールボックスは **SCHEDULED** 状態になり、送信がアポートされ、TXOK ビットがクリアされて、メールボックスは **EMPTY** 状態になります。いずれの場合にも、現在の送信が終了するとメールボックスは最低でも再び **EMPTY** 状態になります。

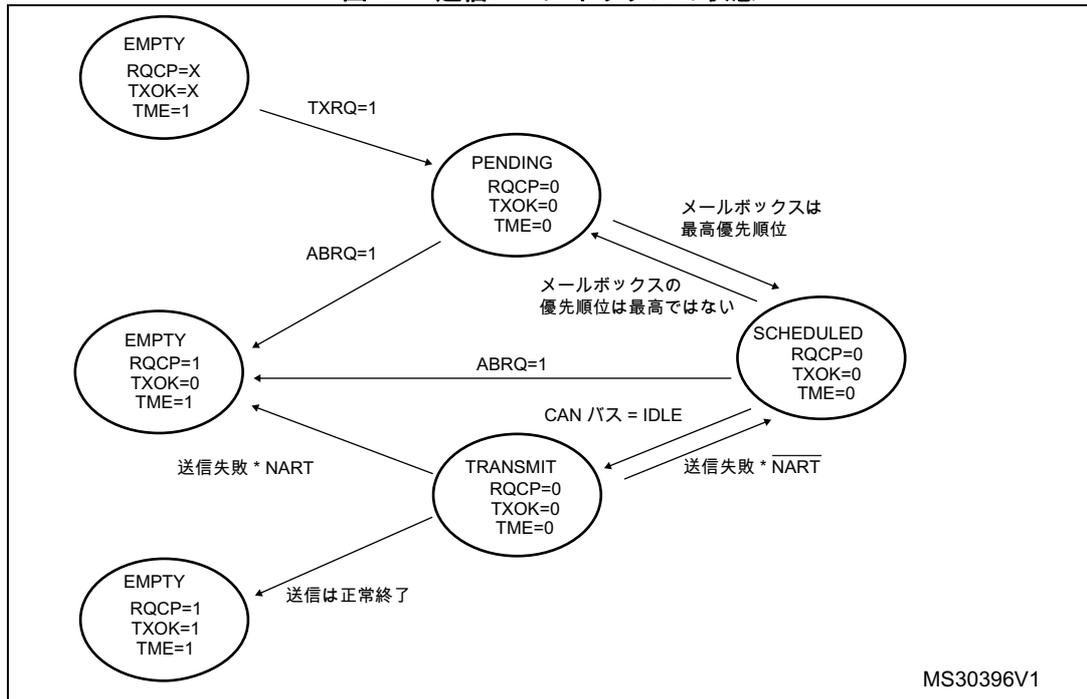
## 自動再送信禁止モード

このモードは、CAN 規格のタイムトリガ通信オプションの要件を満たすために実装されています。ハードウェアをこのモードに設定するには、CAN\_MCR レジスタの NART ビットをセットする必要があります。

このモードでは、個々の送信は一度だけ開始されます。最初の試みがアービトレーションロスやエラーのために失敗した場合、ハードウェアはメッセージ送信を自動的にリスタートしません。

最初の送信試行の終わりに、ハードウェアはそのリクエストが完了したものとみなし、CAN\_TSR レジスタの RQCP ビットをセットします。送信の結果は、CAN\_TSR レジスタの TXOK、ALST、および TERR ビットによって示されます。

図 427. 送信メールボックスの状態



MS30396V1

### 36.7.2 タイムトリガ通信モード

このモードでは、CAN ハードウェアの内部カウンタがアクティブになり、このカウンタを使用して受信/送信メールボックスの CAN\_RDTxR/CAN\_TDTxR レジスタに格納されるタイムスタンプ値を生成します。内部カウンタは、CAN ビット時間ごとにインクリメントされます ([セクション 36.7.7: ビットタイミング](#)を参照)。内部カウンタは、送受信ともに、フレーム開始 (SOF) ビットのサンプルポイントでキャプチャされます。

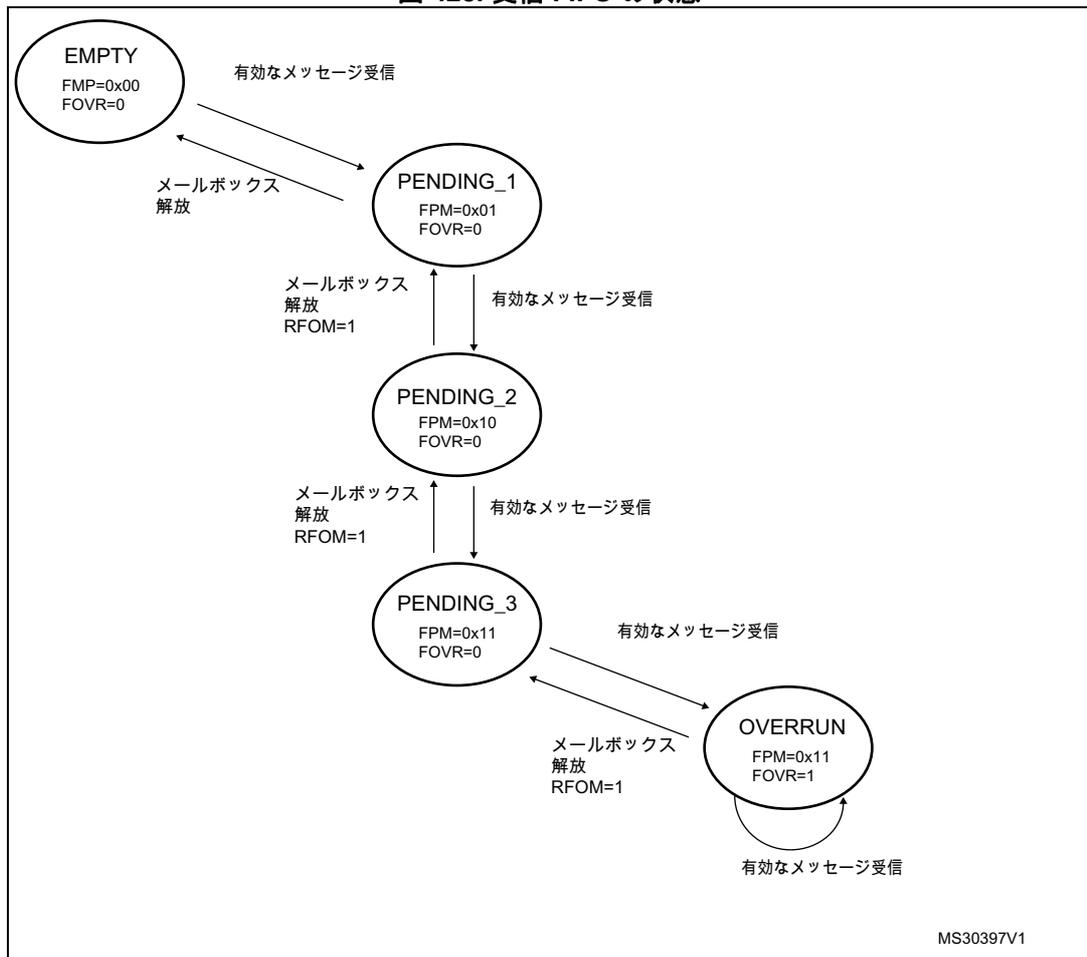
### 36.7.3 受信処理

CAN メッセージの受信には、3つのメールボックスが FIFO として用意されています。CPU 負荷を軽減し、ソフトウェアを単純化し、データの一貫性を保証するために、FIFO の管理はすべてハードウェアによって行われます。アプリケーションは、FIFO 出力メールボックスを通じて、FIFO に格納されたメッセージにアクセスします。

#### 有効なメッセージ

受信メッセージは、CAN プロトコルに従って正しく受信され (EOF フィールドの最後から 2 番目のビットまでエラーなし)、かつ、識別子フィルタリングを正常に通過した場合に、有効であるとみなされます ([セクション 36.7.4: 識別子フィルタリング](#)を参照)。

図 428. 受信 FIFO の状態



## FIFO 管理

**EMPTY** 状態から始まり、受信した最初の有効なメッセージが FIFO に格納されると、FIFO は **PENDING\_1** 状態になります。ハードウェアは、CAN\_RFR レジスタの FMP[1:0] ビットを値 01b に設定して、このイベントを通知します。このメッセージは FIFO 出力メールボックスからアクセスできます。ソフトウェアは、このメールボックスの内容を読み出し、CAN\_RFR レジスタの RFOM ビットをセットすることによってメールボックスを解放します。FIFO は再び **EMPTY** 状態になります。その間に新しい有効なメッセージを受信すると、FIFO は **PENDING\_1** 状態にとどまり、出力メールボックスから新しいメッセージにアクセスできます。

アプリケーションがメールボックスを解放しない場合には、次の有効なメッセージが FIFO に格納され、FIFO は **PENDING\_2** 状態 (FMP[1:0] = 10b) になります。この格納プロセスは、次の有効なメッセージに対しても繰り返され、FIFO は **PENDING\_3** 状態 (FMP[1:0] = 11b) になります。この時点で、メールボックスが次の有効なメッセージを格納できるように、ソフトウェアは、RFOM ビットをセットして、出力メールボックスを解放する必要があります。解放しないと、受信した次の有効なメッセージが失われます。

[セクション 36.7.5 : メッセージストレージ](#)も参照してください。

## オーバーラン

FIFO が **PENDING\_3** 状態 (すなわち、3 個のメールボックスが満杯) になると、次の有効なメッセージの受信で **OVERRUN** 状態となり、メッセージが失われます。ハードウェアは、CAN\_RFR レジスタの FOVR ビットをセットすることによって、オーバーラン状態を通知します。どのメッセージが失われるかは、FIFO の設定に依存します。

- FIFO ロック機能が無効 (CAN\_MCR レジスタの RFLM ビットがクリア) である場合、最後に FIFO に格納されたメッセージが新しい受信メッセージによって上書きされます。この場合、アプリケーションは常に最新のメッセージにアクセスできます。
- FIFO ロック機能が有効 (CAN\_MCR レジスタの RFLM ビットがセット) である場合、最新のメッセージが破棄され、ソフトウェアは FIFO 内にある 3 個の最も古いメッセージにアクセスできます。

## 受信関連の割込み

メッセージが FIFO に格納されると、FMP[1:0] ビットが更新され、割込みリクエストが生成されます (CAN\_IER レジスタの FMPIE ビットがセットされている場合)。

FIFO が満杯になると (3 番目のメッセージが格納されると)、CAN\_RFR レジスタの FULL ビットがセットされ、割込みが生成されます (CAN\_IER レジスタの FFIE ビットがセットされている場合)。

オーバーラン状態になると、FOVR ビットがセットされ、割込みが生成されます (CAN\_IER レジスタの FOVIE ビットがセットされている場合)。

## 36.7.4 識別子フィルタリング

CAN プロトコルでは、メッセージの識別子は、ノードのアドレスには関連付けられておらず、メッセージの内容に関連したものとなっています。したがって、トランスミッタは、そのメッセージをすべてのレシーバにブロードキャスト (一斉送信) します。メッセージを受信すると、受信ノードは、識別子の値に基づいてこのメッセージがソフトウェアに必要なものかどうかを判断します。このメッセージが必要な場合は、SRAM にコピーされます。そうでない場合、このメッセージは、ソフトウェアによる介入なしに破棄される必要があります。

この要件を満たすため、bxCAN コントローラは 28 個の設定可能でスケラブルなフィルタバンク (27-0) をアプリケーションに提供します。それ以外のデバイスでは、ソフトウェアに必要なメッセージのみを受信できるように、bxCAN コントローラは 14 個の設定可能でスケラブルなフィルタバンク (13-0) をアプリケーションに提供します。このハードウェアフィルタリングによって、ソフトウェアでフィルタリングを行う場合に必要となる CPU リソースを節約できます。各フィルタバンク x は、2 個の 32 ビットレジスタ (CAN\_FxR0 と CAN\_FxR1) で構成されています。

### スケラブルな幅

フィルタをアプリケーションニーズに合わせて最適化/適応するため、各フィルタバンクは個別に拡大/縮小できます。フィルタのスケールに応じて、フィルタバンクは以下のフィルタを提供します。

- 1 個の 32 ビットフィルタ : STDID[10:0]、EXTID[17:0]、IDE、および RTR ビット用。
- 2 個の 16 ビットフィルタ : STDID[10:0]、RTR、IDE、および EXTID[17:15] ビット用。

 429を参照してください。

さらに、これらのフィルタはマスクモードまたは識別子リストモードに設定できます。

### マスクモード

マスクモードでは、識別子レジスタは、識別子のどのビットを「一致が必要」として扱い、どのビットを「無視」として扱うかを指定するマスクレジスタに関連付けられます。

## 識別子リストモード

識別子リストモードでは、マスクレジスタが識別子レジスタとして使用されます。このように、識別子とマスクを定義する代わりに、2つの識別子を指定することで、単一識別子の数を倍にします。受信識別子のすべてのビットが、フィルタレジスタで指定されたビットと一致する必要があります。

## フィルタバンクのスケールとモードの設定

フィルタバンクは、対応する CAN\_FMR レジスタによって設定されます。フィルタバンクを設定するには、CAN\_FAR レジスタの FACT ビットをクリアしてフィルタバンクを無効にする必要があります。フィルタのスケールは、CAN\_FS1R レジスタの対応する FSCx ビットによって設定されます (図 429 を参照)。対応するマスク/識別子レジスタの識別子リストまたは識別子マスクモードは、CAN\_FMR レジスタの FBMx ビットによって設定されます。

識別子のグループをフィルタするには、マスクモードでマスク/識別子レジスタを設定します。

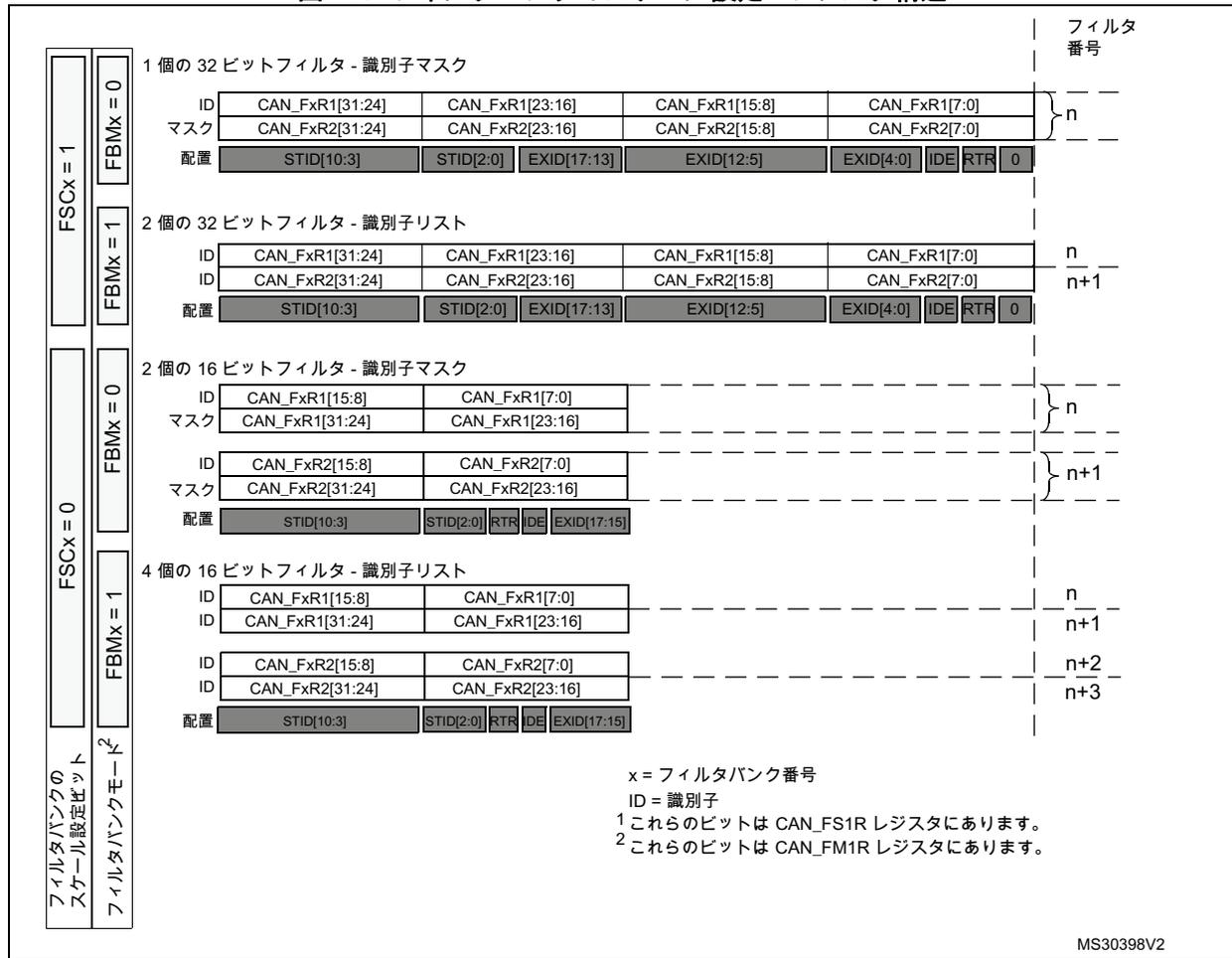
単一識別子を選択するには、識別子リストモードでマスク/識別子レジスタを設定します。

アプリケーションで使用されないフィルタは、非アクティブのままにしておきます。

フィルタバンク内の各フィルタには、各フィルタバンクのモードとスケールに応じて 0 から最大値までの番号 (フィルタ番号) が付けられます。

フィルタの設定については 図 429 を参照してください。

図 429. フィルタバンクのスケール設定 - レジスタ構造



## 該当フィルタインデックス

メッセージは、FIFO で受信するとアプリケーションからアクセス可能になります。通常、アプリケーションのデータは SRAM のいずれかの位置にコピーされます。データを正しい位置にコピーするには、アプリケーションが識別子によってデータを識別する必要があります。この操作を省き、SRAM へのアクセスを容易にするために、CAN コントローラは該当フィルタインデックス (Filter MatchIndex) を搭載しています。

このインデックスは、フィルタの優先順位規則に従って、メッセージとともにメールボックスに格納されます。したがって、受信した各メッセージには関連付けられた該当フィルタインデックスがあります。

該当フィルタインデックスは、次の 2 つの方法で使用できます。

- 該当フィルタインデックスと予期される値のリストを比較します。
- 該当フィルタインデックスを配列のインデックスとして使用し、データの格納先にアクセスします。

マスクされていないフィルタについては、ソフトウェアが識別子を比較する必要はありません。

フィルタがマスクされている場合、ソフトウェアは、マスクされたビットのみを比較します。

フィルタ番号のインデックス値では、フィルタバンクのアクティブ状態を考慮に入れません。さらに、FIFO ごとに 1 つずつ、2 つの独立した番号付け方式を使用します。図 430 の例を参照してください。

図 430. フィルタ番号の例

フィルタバンク	FIFO0	フィルタ番号	フィルタバンク	FIFO1	フィルタ番号
0	ID リスト (32 ビット)	0 1	2	ID マスク (16 ビット)	0 1
1	ID マスク (32 ビット)	2	4	ID リスト (32 ビット)	2 3
3	ID リスト (16 ビット)	3 4 5 6	7	非アクティブ ID リスト (16 ビット)	4 5
5	非アクティブ ID リスト (32 ビット)	7 8	8	ID マスク (16 ビット)	6 7
6	ID マスク (16 ビット)	9 10	10	非アクティブ ID リスト (16 ビット)	8 9 10 11
9	ID リスト (32 ビット)	11 12	11	ID リスト (32 ビット)	12 13
13	ID マスク (32 ビット)	13	12	ID マスク (32 ビット)	14

ID = 識別子

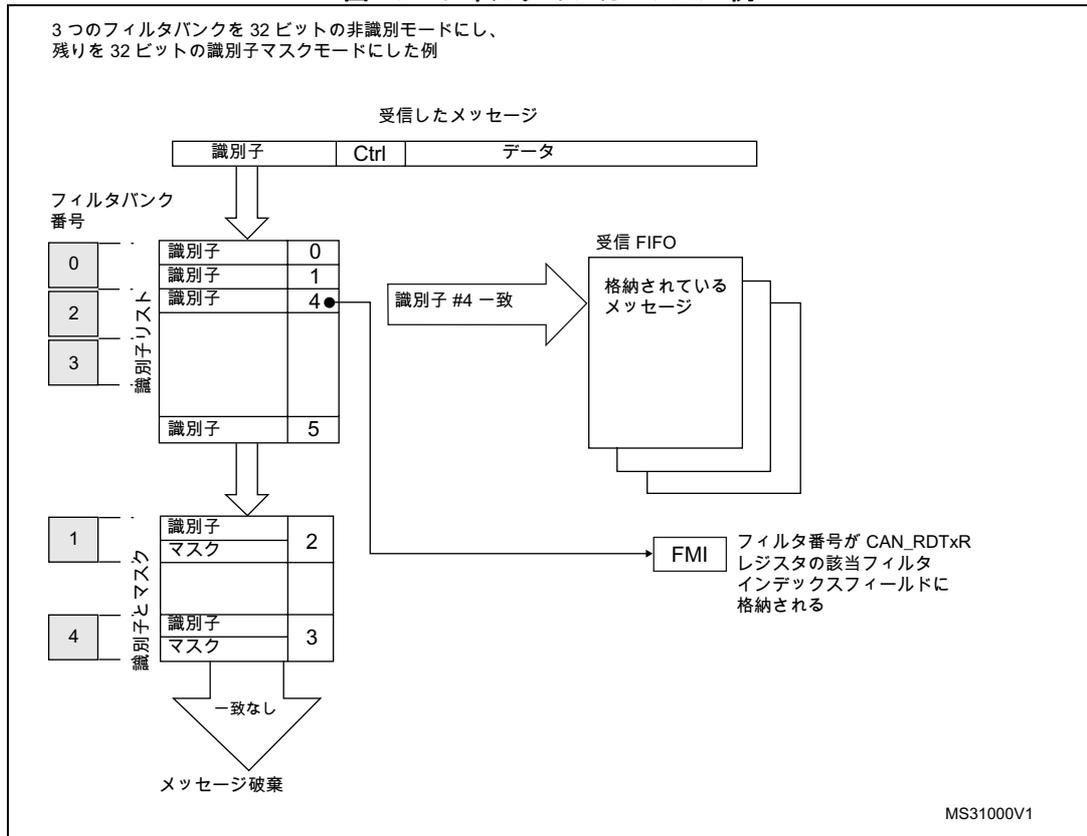
MS30399V1

## フィルタの優先順位規則

フィルタの組み合わせによっては、1 つの識別子が複数のフィルタを正常に通過することがあります。この場合、受信メールボックスに格納される該当フィルタ値は、次の優先順位規則に基づいて選択されます。

- 32 ビットフィルタは 16 ビットフィルタよりも優先されます。
- 同じスケールのフィルタでは、識別子マスクモードよりも識別子リストモードが優先されます。
- スケールとモードが同じフィルタでは、フィルタ番号によって優先順位が与えられます (番号が小さいほど優先順位が高くなります)。

図 431. フィルタのメカニズム - 例



上の例は bxCAN のフィルタリングの原理を示しています。メッセージを受信すると、その識別子は、まず識別子リストモードで設定されたフィルタと比較されます。一致する項目があれば、メッセージは関連する FIFO に格納され、一致したフィルタのインデックスが該当フィルタインデックスに格納されます。この例では、識別子は識別子 #4 と一致するため、メッセージの内容と FMI 2 が FIFO に格納されます。

一致する項目がなければ、受信メッセージの識別子は、次にマスクモードで設定されたフィルタと比較されます。

この識別子がフィルタ内のどの識別子とも一致しない場合、メッセージはハードウェアによって破棄され、ソフトウェアが介入することはありません。

## 36.7.5 メッセージストレージ

CAN メッセージに関するソフトウェアとハードウェアのインターフェースは、メールボックスによって実装されます。メールボックスには、識別子、データ、制御、ステータス、タイムスタンプ情報など、メッセージに関するすべての情報が格納されます。

### 送信メールボックス

ソフトウェアは、送信しようとするメッセージを空の送信メールボックスにセットアップします。送信のステータスは、ハードウェアによって CAN\_TSR レジスタに示されます。

表 224. 送信メールボックスの配置

送信メールボックスのベースアドレスのオフセット	レジスタ名
0	CAN_TlXR
4	CAN_TDTxR
8	CAN_TDLxR
12	CAN_TDHxR

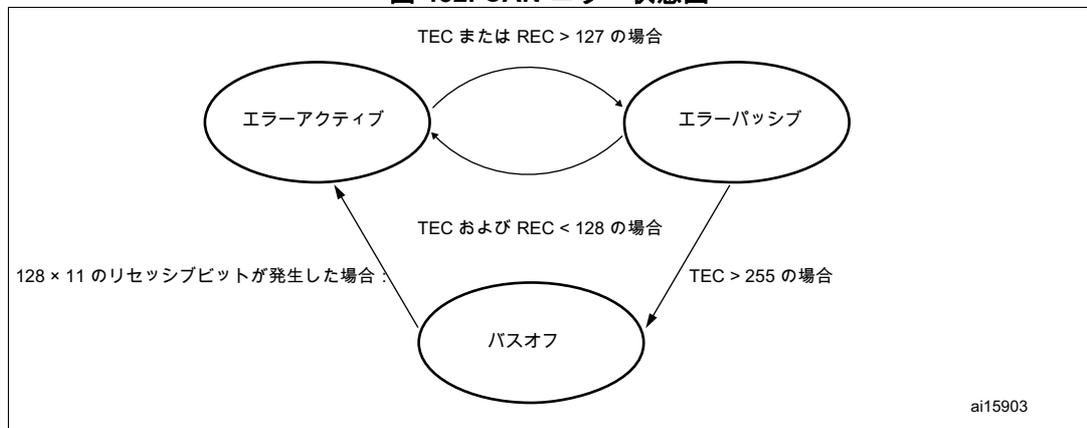
## 受信メールボックス

メッセージを受信すると、ソフトウェアは FIFO 出力メールボックスを通してそのメッセージにアクセスできます。ソフトウェアが読み出しなどのメッセージの処理を終えると、ソフトウェアは、CAN\_RFR レジスタの RFOM ビットによって FIFO 出力メールボックスを解放し、次の受信メッセージにアクセスできるようにする必要があります。該当フィルタインデックスは、CAN\_RDTxR レジスタの MFMI フィールドに格納されます。16 ビットタイムスタンプの値は、CAN\_RDTxR レジスタの TIME[15:0] フィールドに格納されます。

表 225. 受信メールボックスの配置

受信メールボックスのベースアドレスのオフセット (バイト)	レジスタ名
0	CAN_RlXR
4	CAN_RDTxR
8	CAN_RDLxR
12	CAN_RDHxR

図 432. CAN エラー状態図



## 36.7.6 エラー管理

CAN プロトコルに記述されているエラー管理は、エラー条件によってインクリメント/デクリメントされる送信エラーカウンタ (CAN\_ESR レジスタの TEC 値) と受信エラーカウンタ (CAN\_ESR レジスタの REC 値) を使用して、すべてハードウェアによって処理されます。TEC および REC の管理の詳細については、CAN 規格を参照してください。

ソフトウェアは、この 2 つのカウンタを読み出してネットワークの安定性を判断できます。さらに、CAN ハードウェアは、現在のエラーステータスに関する詳細な情報を CAN\_ESR レジスタで提供します。CAN\_IER レジスタ (ERRIE ビットなど) によって、ソフトウェアは、エラー検出時の割込み生成をきわめて柔軟に設定できます。

### バスオフからの復帰

TEC が 255 を超えるとバスオフ状態になります。この状態は CAN\_ESR レジスタの BOFF ビットによって示されます。バスオフ状態では、bxCAN はメッセージの送受信を行うことができません。

CAN\_MCR レジスタの ABOM ビットに応じて、bxCAN は、自動的に、またはソフトウェアのリクエストに基づいて、バスオフ状態から復帰します (再びエラーアクティブになります)。しかし、いずれの場合も、bxCAN は、少なくとも CAN 規格で定められた復帰シーケンス (CANRX で観測される 11 個の連続したリセッショビットの発生 128 回) の間は待機する必要があります。

ABOM がセットされると、bxCAN は、バスオフ状態に入った後で自動的に復帰シーケンスを開始します。

ABOM がクリアされている場合、ソフトウェアは、bxCAN に対して初期化モードに入って出るようにリクエストして、復帰シーケンスを開始する必要があります。

**注:** *初期化モードでは、bxCAN は CANRX 信号を監視しないため、復帰シーケンスを完了できません。復帰するには、bxCAN は通常モードでなければなりません。*

## 36.7.7 ビットタイミング

ビットタイミング回路は、シリアルバスラインを監視し、スタートビットエッジでの同期と後続エッジでの再同期によって、サンプリングとサンプリングポイントの調整を行います。

この動作は、公称ビット時間を次のように 3 つのセグメントに分割するとわかりやすく説明できます。

- **同期セグメント (SYNC\_SEG)**: この時間セグメント内にビット変化が生じることが想定されます。この時間の長さは 1 タイムクオンタム ( $1 \times t_q$ ) に固定されています。
- **ビットセグメント 1 (BS1)**: サンプルポイントの位置を定義します。これには、CAN 規格の PROP\_SEG と PHASE\_SEG1 が含まれます。セグメントの時間は 1~16 タイムクオンタムの間でプログラミング可能ですが、ネットワークのさまざまなノード間の周波数の差に起因する正の位相ドリフトを補正するために、自動的に延長されることがあります。
- **ビットセグメント 2 (BS2)**: 送信ポイントの位置を定義します。これは、CAN 規格の PHASE\_SEG2 に相当します。セグメントの時間は 1~8 タイムクオンタムの間でプログラミング可能ですが、負の位相ドリフトを補正するために、自動的に短縮されることもあります。

再同期ジャンプ幅 (SJW) は、ビットセグメントの延長/短縮量の上限を定め、1~4 タイムクオンタムの間でプログラミング可能です。

有効なエッジとは、コントローラ自体はリセッショビットを送信しないという条件で、あるビット時間において、ドミナントからリセッショへのバスレベルの最初の遷移と定義されます。

有効なエッジが SYNC\_SEG ではなく BS1 で検出された場合、サンプルポイントが遅くなるように、BS1 は最大で SJW 延長されます。



逆に、有効なエッジが SYNC\_SEG ではなく BS2 で検出された場合、送信ポイントが早くなるように、BS2 は最大で SJW 短縮されます。

プログラミングエラーに対する安全対策として、ビットタイミングレジスタ (CAN\_BTR) は、デバイスが STANDBY モードの場合にのみ設定可能です。

**注：** CAN ビットタイミングと再同期メカニズムの詳細については、ISO11898 規格を参照してください。

図 433. ビットタイミング

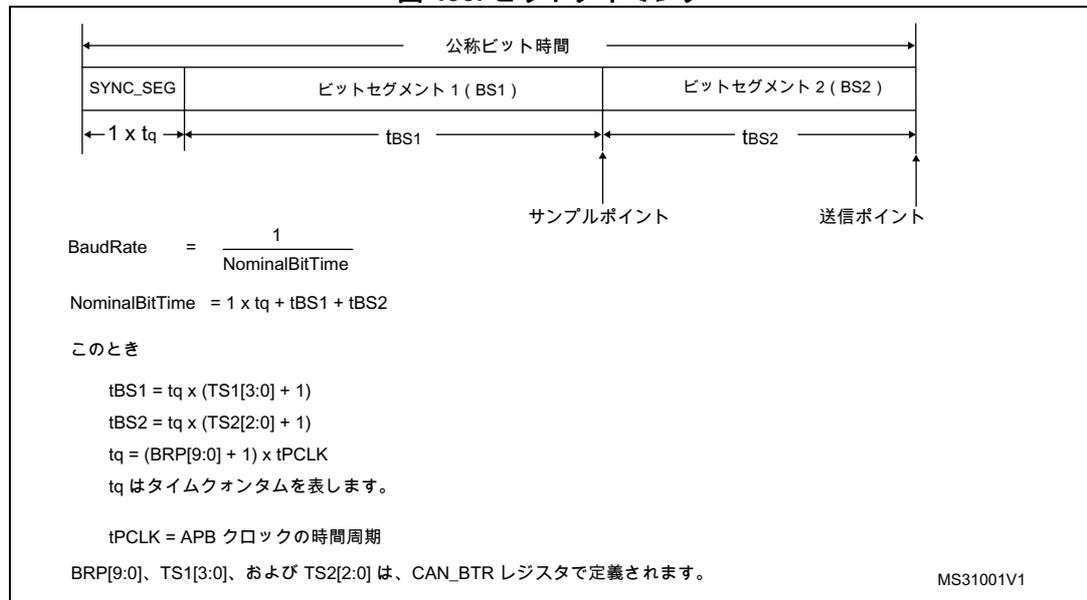
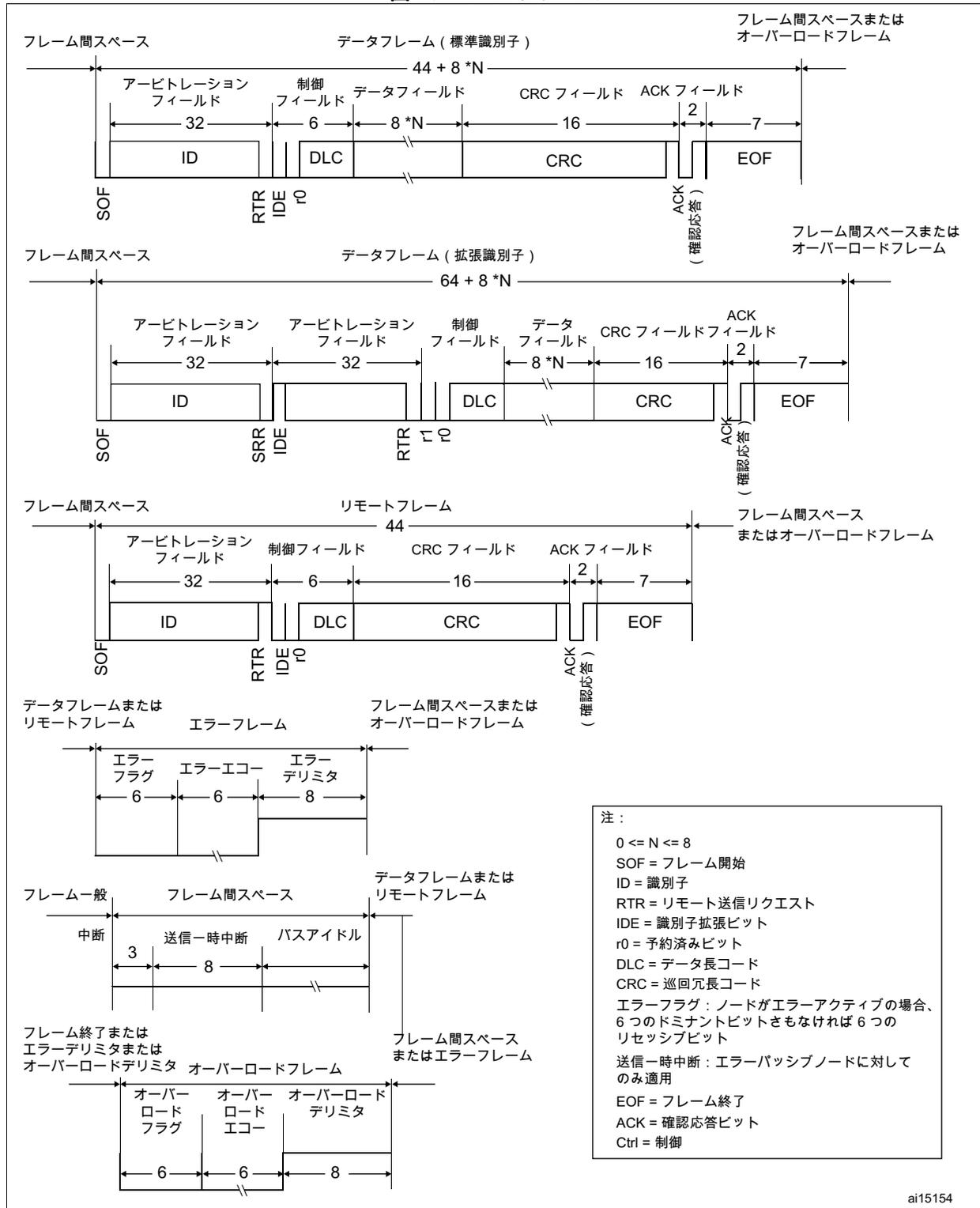


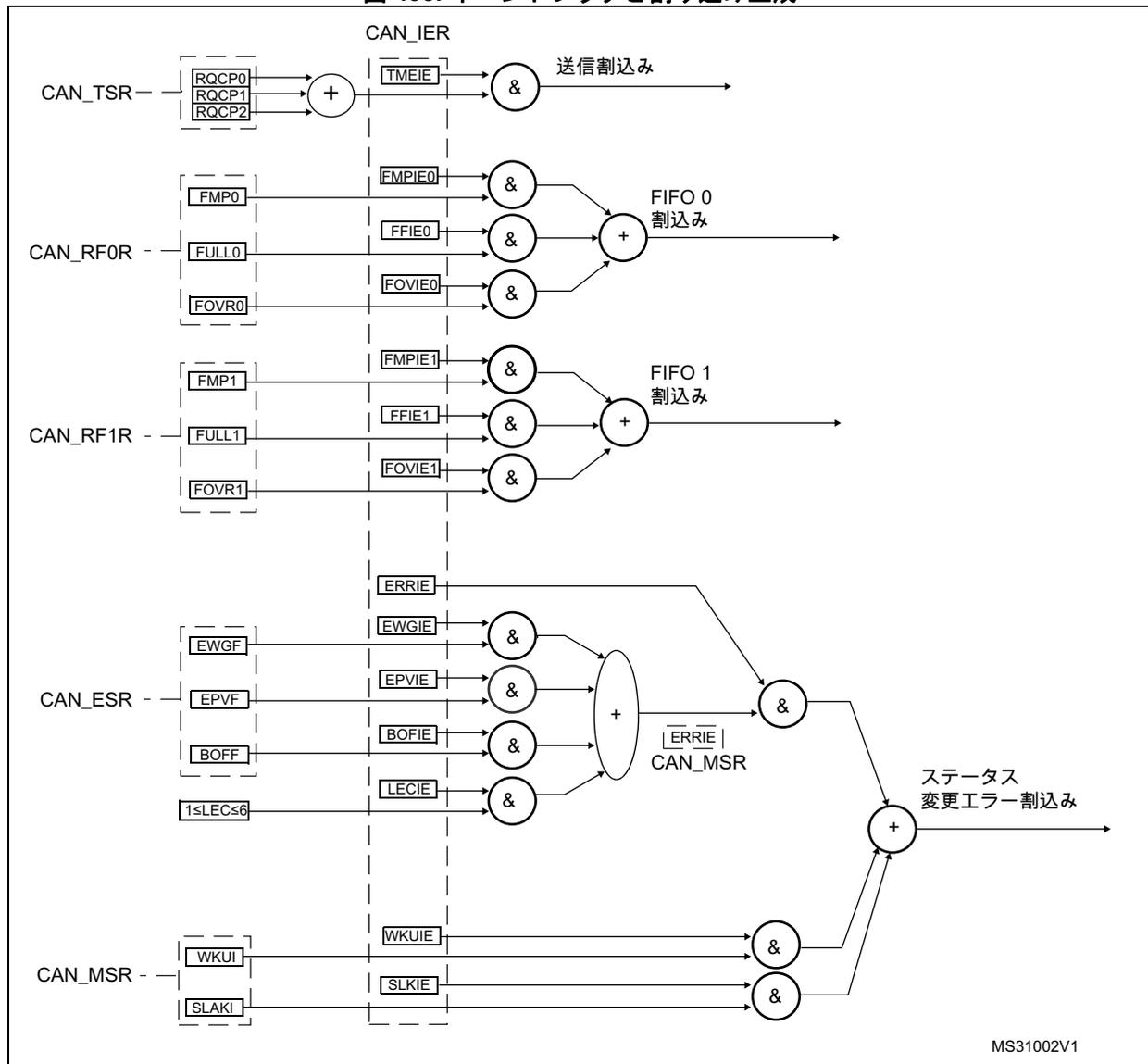
図 434. CAN フレーム



## 36.8 bxCAN の割込み

bxCAN には 4 つの割込みベクタが確保されています。各割込みソースは、CAN 割込み有効レジスタ (CAN\_IER) によって個別に有効/無効にできます。

図 435. イベントフラグと割り込み生成



- **送信割り込み**は次のイベントによって生成できます。
  - 送信メールボックス 0 が空になり、CAN\_TSR レジスタの RQCP0 ビットがセットされます。
  - 送信メールボックス 1 が空になり、CAN\_TSR レジスタの RQCP1 ビットがセットされます。
  - 送信メールボックス 2 が空になり、CAN\_TSR レジスタの RQCP2 ビットがセットされます。
- **FIFO 0 割り込み**は次のイベントによって生成できます。
  - 新しいメッセージの受信、CAN\_RF0R レジスタの FMP0 ビットが“00”ではない場合。
  - FIFO0 の満杯状態、CAN\_RF0R レジスタの FULL0 ビットがセットされる場合。
  - FIFO0 のオーバーラン状態、CAN\_RF0R レジスタの FOVR0 ビットがセットされる場合。

- **FIFO 1 割り込み**は次のイベントによって生成できます。
  - 新しいメッセージの受信、CAN\_RF1R レジスタの FMP1 ビットが“00”ではない場合。
  - FIFO1 の満杯状態、CAN\_RF1R レジスタの FULL1 ビットがセットされる場合。
  - FIFO1 のオーバーラン状態、CAN\_RF1R レジスタの FOVR1 ビットがセットされる場合。
- **エラー/ステータス変更割り込み**は次のイベントによって生成できます。
  - エラー条件：エラー条件の詳細については、CAN エラーステータスレジスタ (CAN\_ESR) を参照してください。
  - ウェイクアップ条件：CAN Rx 信号で SOF が観測されたとき。
  - スリープモードへの遷移。

## 36.9 CAN レジスタ

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

### 36.9.1 レジスタのアクセス保護

特定の設定レジスタに誤ったアクセスを行うと、ハードウェアが CAN ネットワーク全体の動作を一時的に妨害することがあります。このため、ソフトウェアが CAN\_BTR レジスタを変更できるのは、CAN ハードウェアが初期化モードのときに限られます。

誤ったデータを送信しても CAN ネットワークレベルでは問題は生じませんが、アプリケーションには重大な影響を与える可能性があります。ソフトウェアが送信メールボックスを変更できるのは、メールボックスが EMPTY 状態のときに限られます。図 427: [送信メールボックスの状態](#)を参照してください。

フィルタ値を変更するには、関連するフィルタバンクを非アクティブにするか、FINIT ビットをセットします。さらに、CAN\_FMxR、CAN\_FSxR、および CAN\_FFAR レジスタにあるフィルタ設定 (スケール、モード、および FIFO 割当て) の変更は、CAN\_FMR レジスタでフィルタ初期化モードがセットされている (FINIT = 1) 場合に限られます。

### 36.9.2 CAN 制御/ステータスレジスタ

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

#### CAN マスタ制御レジスタ (CAN\_MCR)

アドレスオフセット：0x00

リセット値：0x0001 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DBF														
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
リセット	Res.	TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ						
rs								rw	rw						



ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **DBF** : デバッグ凍結

0 : デバッグ時に CAN が機能します。

1 : デバッグ時に CAN の送受信機能が凍結されます。受信 FIFO は、まだ正常にアクセス/制御できます。

ビット 15 **RESET** : bxCAN ソフトウェアマスタリセット

0 : 通常動作。

1 : bxCAN のマスタリセットを強制します。リセット後は、スリープモードがアクティブになります (FMP ビットと CAN\_MCR レジスタはリセット値に初期化されます)。このビットは自動的に 0 にリセットされます。

ビット 14:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TTCM** : タイムトリガ通信モード

0 : タイムトリガ通信モードは無効です。

1 : タイムトリガ通信モードは有効です。

**注 :** [タイムトリガ通信モードの詳細については、セクション 36.7.2 : タイムトリガ通信モードを参照してください。](#)

ビット 6 **ABOM** : 自動バスオフ管理

このビットは、バスオフ状態から出るときの CAN ハードウェアの動作を制御します。

0 : 11 個のリセシブビットが 128 回観測され、ソフトウェアが CAN\_MCR レジスタの INRQ ビットをまずセットし、クリアすると、ソフトウェアリクエストでバスオフ状態から出ます。

1 : 11 個のリセシブビットが 128 回観測されると、ハードウェアによって自動的にバスオフ状態から出ます。

バスオフ状態の詳細については、[セクション 36.7.6 : エラー管理](#)を参照してください。

ビット 5 **AWUM** : 自動ウェイクアップモード

このビットは、スリープモードでのメッセージ受信時の CAN ハードウェアの動作を制御します。

0 : ソフトウェアリクエストで CAN\_MCR レジスタの SLEEP ビットをクリアしてスリープモードから出ます。

1 : CAN メッセージの検出時にハードウェアによって自動的にスリープモードから出ます。

CAN\_MCR レジスタの SLEEP ビットと CAN\_MSR レジスタの SLAK ビットはハードウェアによってクリアされます。

ビット 4 **NART** : 自動再送信禁止

0 : CAN ハードウェアは、CAN 規格に基づいてメッセージを正常に送信するまで、メッセージを自動的に再送信します。

1 : 送信結果 (正常、エラー、アービトレーションロスト) にかかわらず、メッセージは 1 回だけ送信されます。

ビット 3 **RFLM** : 受信 FIFO ロックモード

0 : 受信 FIFO はオーバーラン時にロックされません。受信 FIFO が満杯になると、次の受信メッセージは前のメッセージに上書きされます。

1 : 受信 FIFO はオーバーランに対してロックされます。受信 FIFO が満杯になると、次の受信メッセージは破棄されます。

**ビット 2 TXFP** : 送信 FIFO 優先順位

このビットは、複数のメールボックスが同時にペンディング状態のときの送信順序を制御します。

0 : 優先順位はメッセージの識別子によって決まります。

1 : 優先順位はリクエスト順に (時系列に) 決まります。

**ビット 1 SLEEP** : スリープモードリクエスト

このビットは、CAN ハードウェアがスリープモードに入るように要求するために、ソフトウェアによってセットされます。現在の CAN 動作 (CAN フレームの送受信) が完了すると、すぐにスリープモードに入ります。

ソフトウェアによってこのビットがクリアされると、スリープモードを終了します。

AWUM ビットがセットされていて、CAN Rx 信号で SOF ビットが検出されると、このビットはハードウェアによってクリアされます。

このビットはリセット後にセットされます。CAN はスリープモードで開始します。

**ビット 0 INRQ** : 初期化リクエスト

ソフトウェアによってこのビットがクリアされると、ハードウェアが通常モードに切り替わります。Rx 信号で 11 個の連続したリセツピットが観測されると、CAN ハードウェアが同期し、送受信の準備ができます。ハードウェアは、CAN\_MSR レジスタの INAK ビットをクリアすることによって、このイベントを通知します。

このビットは、CAN ハードウェアに初期化モードに入るようにリクエストするために、ソフトウェアによってセットされます。ソフトウェアが INRQ ビットをセットすると、CAN ハードウェアは現在の CAN 動作 (送受信) が完了するのを待ってから、初期化モードに入ります。ハードウェアは、CAN\_MSR レジスタの INAK ビットをセットすることによって、このイベントを通知します。

## CAN マスタステータスレジスタ (CAN\_MSR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0C02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	SLAKI	WKUI	ERRI	SLAK	INAK
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

**ビット 11 RX** : CAN Rx 信号

CAN\_RX ピンの実際の値を監視します。

**ビット 10 SAMP** : 最終サンプルポイント

最後のサンプルポイントにおける RX の値 (受信した現在のビット値) です。

**ビット 9 RXM** : 受信モード

CAN ハードウェアは、現在はレシーバです。

**ビット 8 TXM** : 送信モード

CAN ハードウェアは、現在はトランスミッタです。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。



## ビット 4 SLAKI : スリープ確認割込み

SLKIE=1 の場合、このビットは、bxCAN がスリープモードに入ったことを通知するために、ハードウェアによってセットされます。CAN\_IER レジスタの SLKIE ビットがセットされている場合にこのビットがセットされると、ステータス変更割込みが生成されます。

SLAK がクリアされると、このビットはソフトウェアまたはハードウェアによってクリアされます。

**注:** *SLKIE=0 の場合、SLAKI ビットのポーリングはできません。この場合、SLAK ビットはポーリングできません。*

## ビット 3 WKUI : ウェイクアップ割り込み

このビットは、CAN ハードウェアがスリープモードのときに SOF ビットが検出されたことを通知するために、ハードウェアによってセットされます。CAN\_IER レジスタの WKUIE ビットがセットされている場合、このビットをセットするとステータス変更割込みが生成されます。

このビットはソフトウェアによってクリアされます。

## ビット 2 ERRI : エラー割込み

エラー検出時に CAN\_ESR のビットがセットされており、CAN\_IER で対応する割込みが有効になっている場合、このビットはハードウェアによってセットされます。CAN\_IER レジスタの ERRIE ビットがセットされている場合、このビットをセットするとステータス変更割込みが生成されます。

このビットはソフトウェアによってクリアされます。

## ビット 1 SLAK : スリープ確認応答

このビットは、ハードウェアによってセットされ、CAN ハードウェアがスリープモードにあることをソフトウェアに知らせます。このビットは、ソフトウェアからのスリープモードリクエスト (CAN\_MCR レジスタの SLEEP ビットをセット) に対する確認応答になります。

このビットは、CAN ハードウェアがスリープモードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセツピットを観測する必要があります。

**注:** *CAN\_MCR レジスタの SLEEP ビットがクリアされると、スリープモードから出るプロセスがトリガされます。SLEEP ビットのクリアについては、CAN\_MCR レジスタの AWUM ビットの説明を参照してください。*

## ビット 0 INAK : 初期化確認応答

このビットは、CAN ハードウェアが初期化モードにあることをソフトウェアに知らせるために、ハードウェアによってセットされます。このビットは、ソフトウェアからの初期化リクエスト (CAN\_MCR レジスタの INRQ ビットをセット) に対する確認応答になります。

このビットは、CAN ハードウェアが初期化モードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセツピットを観測する必要があります。

## CAN 送信ステータスレジスタ (CAN\_TSR)

アドレスオフセット : 0x08

リセット値 : 0x1C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2
r	r	r	r	r	r	r	r	rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	TERR0	ALST0	TXOK0	RQCP0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

ビット 31 **LOW2** : メールボックス 2 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 2 が最低優先順位るとき、このビットがハードウェアによってセットされます。

ビット 30 **LOW1** : メールボックス 1 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 1 が最低優先順位るとき、このビットがハードウェアによってセットされます。

ビット 29 **LOW0** : メールボックス 0 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 0 が最低優先順位るとき、このビットがハードウェアによってセットされます。

**注 :** *ペンディング状態のメールボックスが1 つしかないとき、LOW[2:0] ビットは 0 にセットされません。*

ビット 28 **TME2** : 送信メールボックス 2 エンプティ

このビットは、メールボックス 2 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 27 **TME1** : 送信メールボックス 1 エンプティ

このビットは、メールボックス 1 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 26 **TME0** : 送信メールボックス 0 エンプティ

このビットは、メールボックス 0 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 25:24 **CODE[1:0]** : メールボックスコード

少なくとも 1 個の送信メールボックスが空き状態の場合、このコード値は、次の空き状態の送信メールボックスの番号になります。

すべての送信メールボックスが待ち状態の場合、このコード値は、最低優先順位の送信メールボックスの番号になります。

ビット 23 **ABRQ2** : メールボックス 2 アポートリクエスト

対応するメールボックスへの送信リクエストをアポートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 22:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **TERR2** : メールボックス 2 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

ビット 18 **ALST2** : メールボックス 2 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。



ビット 17 **TXOK2** : メールボックス 2 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0 : 前回の送信は失敗しました。

1 : 前回の送信は成功しました。

このビットは、メールボックス 2 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 427 を参照してください。

ビット 16 **RQCP2** : メールボックス 2 リクエスト完了

最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN\_T MID2R レジスタの TXRQ2 ビットをセット)。

このビットをクリアすると、メールボックス 2 のすべてのステータスビット (TXOK2、ALST2、および TERR2) がクリアされます。

ビット 15 **ABRQ1** : メールボックス 1 アボートリクエスト

対応するメールボックスへの送信リクエストをアボートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **TERR1** : メールボックス 1 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

ビット 10 **ALST1** : メールボックス 1 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。

ビット 9 **TXOK1** : メールボックス 1 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0 : 前回の送信は失敗しました。

1 : 前回の送信は成功しました。

このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 427 を参照してください。

ビット 8 **RQCP1** : メールボックス 1 リクエスト完了

最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN\_T I1R レジスタの TXRQ1 ビットをセット)。

このビットをクリアするとメールボックス 1 のすべてのステータスビット (TXOK1、ALST1、および TERR1) がクリアされます。

ビット 7 **ABRQ0** : メールボックス 0 アボートリクエスト

対応するメールボックスへの送信リクエストをアボートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TERR0** : メールボックス 0 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

- ビット 2 ALST0** : メールボックス 0 アービトレーションロスト  
このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。
- ビット 1 TXOK0** : メールボックス 0 送信OK  
ハードウェアは、送信を試みた後で毎回このビットを更新します。  
0 : 前回の送信は失敗しました。  
1 : 前回の送信は成功しました。  
このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 427 を参照してください。
- ビット 0 RQCP0** : メールボックス 0 リクエスト完了  
最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN\_TIOER レジスタの TXRQ0 ビットをセット)。  
このビットをクリアするとメールボックス 0 のすべてのステータスビット (TXOK0、ALST0、および TERR0) がクリアされます。

## CAN 受信 FIFO 0 レジスタ (CAN\_RF0R)

アドレスオフセット : 0x0C  
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]										
										rs	rc_w1	rc_w1		r	r

- ビット 31:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 RFOM0** : FIFO 0 出力メールボックスの解放  
FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。  
このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。
- ビット 4 FOVR0** : FIFO 0 オーバーラン  
このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。  
このビットはソフトウェアによってクリアされます。
- ビット 3 FULL0** : FIFO 0 フル  
FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。  
このビットはソフトウェアによってクリアされます。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1:0 FMP0[1:0]** : FIFO 0 メッセージペンディング  
これらのビットは、受信 FIFO 内でペンディング状態になっているメッセージの数を示します。FMP は、ハードウェアが FIFO に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM0 ビットをセットして出力メールボックスを解放するたびに減少します。



## CAN 受信 FIFO 1 レジスタ (CAN\_RF1R)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]										
										rs	rc_w1	rc_w1		r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **RFOM1** : FIFO 1 出力メールボックスの解放

FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。

このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。

ビット 4 **FOVR1** : FIFO 1 オーバーラン

このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。

このビットはソフトウェアによってクリアされます。

ビット 3 **FULL1** : FIFO 1 フル

FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。

このビットはソフトウェアによってクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **FMP1[1:0]** : FIFO 1 メッセージペンディング

これらのビットは、受信 FIFO1 内でペンディング状態になっているメッセージの数を示します。

FMP1 は、ハードウェアが FIFO1 に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM1 ビットをセットして出力メールボックスを解放するたびに減少します。

## CAN 割り込み有効レジスタ (CAN\_IER)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Res.	Res.	Res.	LEC IE	BOF IE	EPV IE	EWG IE	Res.	FOV IE1	FF IE1	FMP IE1	FOV IE0	FF IE0	FMP IE0	TME IE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **SLKIE** : スリープ割り込み有効

- 0 : SLAKI ビットがセットされると、割り込みは生成されません。
- 1 : SLAKI ビットがセットされると、割り込みが生成されます。

ビット 16 **WKUIE** : ウェイクアップ割り込み有効

- 0 : WKUI ビットがセットされると、割り込みは生成されません。
- 1 : WKUI ビットがセットされると、割り込みが生成されます。

ビット 15 **ERRIE** : エラー割り込みイネーブル

- 0 : CAN\_ESR レジスタでエラー条件がペンディング状態のとき、割り込みは生成されません。
- 1 : CAN\_ESR レジスタでエラー条件がペンディング状態のとき、割り込みが生成されます。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **LECIE** : 最終エラーコード割り込み有効

- 0 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされるとき、ERRI ビットはセットされません。
- 1 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされるとき、ERRI ビットがセットされます。

ビット 10 **BOFIE** : バスオフ割り込み有効

- 0 : BOFF がセットされるとき、ERRI ビットはセットされません。
- 1 : BOFF がセットされると、ERRI ビットがセットされます。

ビット 9 **EPVIE** : エラーパッシブ割り込み有効

- 0 : EPVF がセットされるとき、ERRI ビットはセットされません。
- 1 : EPVF がセットされると、ERRI ビットがセットされます。

ビット 8 **EWGIE** : エラー警告割り込み有効

- 0 : EWGF がセットされるとき、ERRI ビットはセットされません。
- 1 : EWGF がセットされると、ERRI ビットがセットされます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FOVIE1** : FIFO オーバーラン割り込み有効

- 0 : FOVR がセットされるとき、割り込みは生成されません。
- 1 : FOVR がセットされると、割り込みが生成されます。

ビット 5 **FFIE1** : FIFO フル割り込み有効

- 0 : FULL ビットがセットされるとき、割り込みは生成されません。
- 1 : FULL ビットがセットされると、割り込みが生成されます。



- ビット 4 **FMPIE1** : FIFO メッセージペンドイング割込み有効  
 0 : FMP[1:0] ビットの状態が 00b でないとき、割込みは生成されません。  
 1 : FMP[1:0] ビットの状態が 00b でないとき、割込みが生成されます。
- ビット 3 **FOVIE0** : FIFO オーバーラン割込み有効  
 0 : FOVR ビットがセットされる時、割込みは生成されません。  
 1 : FOVR ビットがセットされると、割込みが生成されます。
- ビット 2 **FFIE0** : FIFO フル割込み有効  
 0 : FULL ビットがセットされる時、割込みは生成されません。  
 1 : FULL ビットがセットされると、割込みが生成されます。
- ビット 1 **FMPIE0** : FIFO メッセージペンドイング割込み有効  
 0 : FMP[1:0] ビットの状態が 00b でないとき、割込みは生成されません。  
 1 : FMP[1:0] ビットの状態が 00b でないとき、割込みが生成されます。
- ビット 0 **TMEIE** : 送信メールボックスエンプティ割込み有効  
 0 : RQCPx ビットがセットされる時、割込みは生成されません。  
 1 : RQCPx ビットがセットされると、割込みが生成されます。
- 注 :** [セクション 36.8 : bxCAN の割込みを参照してください。](#)

## CAN エラーステータスレジスタ (CAN\_ESR)

アドレスオフセット : 0x18  
 リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REC[7:0]								TEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEC[2:0]			Res.	BOFF	EPVF	EWGF
									r/w	r/w	r/w		r	r	r

- ビット 31:24 **REC[7:0]** : 受信エラーカウンタ  
 CAN プロトコルの障害隔離メカニズムの実装部です。受信時にエラーがあった場合、このカウンタは、CAN 規格に定められたエラー条件に応じて、1 または 8 だけインクリメントされます。受信が正常に行われるたびに、このカウンタは 1 だけデクリメントされます (カウンタの値が 128 を超えた場合は、120 にリセットされます)。カウンタ値が 127 を超えると、CAN コントローラはエラーパッシブ状態に入ります。
- ビット 23:16 **TEC[7:0]** : 9 ビット送信エラーカウンタの最下位バイト  
 CAN プロトコルの障害隔離メカニズムの実装部です。
- ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **LEC[2:0]** : 最終エラーコード

このフィールドはハードウェアによってセットされ、最後に CAN バスで検出されたエラーのエラー条件を示すコードを保持しています。メッセージがエラーなしで転送 (受信または送信) されている場合には、このフィールドは“0”にクリアされます。

LEC[2:0] ビットは、ソフトウェアによって値 0b111 にセットできます。ビットはハードウェアによって更新され、現在の通信ステータスを示します。

- 000 : エラーなし
- 001 : スタッフエラー
- 010 : フォームエラー
- 011 : ACK エラー
- 100 : ビットリセッショエラー
- 101 : ビットドミナントエラー
- 110 : CRC エラー
- 111 : ソフトウェアによってセットされます

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **BOFF** : バスオフフラグ

このビットは、バスオフ状態に入るとハードウェアによってセットされます。バスオフ状態に入るのは、TEC オーバーフローのとき (255 を超えたとき) です。 [セクション 36.7.6 \(1265 ページ\)](#) を参照してください。

ビット 1 **EPVF** : エラーパッシブフラグ

このビットは、エラーパッシブ限界に達したときに (受信エラーカウンタまたは送信エラーカウンタ > 127)、ハードウェアによってセットされます。

ビット 0 **EWGF** : エラー警告フラグ

このビットは、警告限界に達したときに、ハードウェアによってセットされます (受信エラーカウンタまたは送信エラーカウンタ ≥ 96)。

## CAN ビットタイミングレジスタ (CAN\_BTR)

アドレスオフセット : 0x1C

リセット値 : 0x0123 0000

このレジスタは、CAN ハードウェアが初期化モードのときに、ソフトウェアによってのみアクセスできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
SILM	LBKM	Res.	Res.	Res.	Res.	SJW[1:0]		Res.	TS2[2:0]			TS1[3:0]				
rw	rw					rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	Res.	Res.	Res.	Res.	BRP[9:0]										
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

ビット 31 **SILM** : サイレントモード (デバッグ用)

- 0 : 通常動作
- 1 : サイレントモード

ビット 30 **LBKM** : ループバックモード (デバッグ用)

- 0 : ループバックモードは無効です。
- 1 : ループバックモードは有効です。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。



ビット 25:24 **SJW[1:0]** : 再同期ジャンプ幅

これらのビットは、再同期を行うために CAN ハードウェアがビットを伸縮できる最大タイムクオンタム数を定義します。

$$t_{RJW} = t_q \times (SJW[1:0] + 1)$$

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **TS2[2:0]** : 時間セグメント 2

これらのビットは、時間セグメント 2 に含まれるタイムクオンタム数を定義します。

$$t_{BS2} = t_q \times (TS2[2:0] + 1)$$

ビット 19:16 **TS1[3:0]** : 時間セグメント 1

これらのビットは、時間セグメント 1 に含まれるタイムクオンタム数を定義します。

$$t_{BS1} = t_q \times (TS1[3:0] + 1)$$

ビットタイミングの詳細については、[セクション 36.7.7 : ビットタイミング \(1265 ページ\)](#) を参照してください。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **BRP[9:0]** : ボーレートプリスケアラ

これらのビットは、タイムクオンタムの長さを定義します。

$$t_q = (BRP[9:0] + 1) \times t_{PCLK}$$

## 36.9.3 CAN メールボックスレジスタ

ここでは、送受信メールボックスのレジスタについて説明します。詳細なレジスタ配置については、[セクション 36.7.5 : メッセージストレージ \(1263 ページ\)](#) を参照してください。

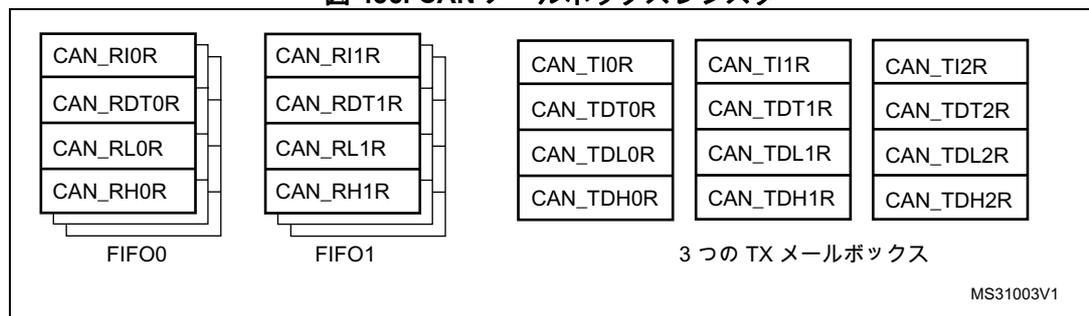
送信メールボックスと受信メールボックスのレジスタは同じですが、次の点が異なります。

- CAN\_RDTxR レジスタの FMI フィールド。
- 受信メールボックスは常に書き込み保護されます。
- 送信メールボックスは空のとき (CAN\_TSR レジスタの対応する TME ビットがセットされているとき) のみ、書き込み可能です。

送信メールボックスは 3 つ、受信メールボックスは 2 つあります。各受信メールボックスでは、3 種類の深さのレベルの FIFO にアクセスでき、FIFO 内の最も古い受信メッセージにのみアクセスできます。

各メールボックスは 4 つのレジスタから構成されています。

図 436. CAN メールボックスレジスタ



MS31003V1

## CAN 送信メールボックス識別子レジスタ (CAN\_TlRxR) (x=0..2)

アドレスオフセット : 0x180、0x190、0x1A0

リセット値 : 0xFFFF XXXX (ただし、ビット 0、TXRQ = 0)

メールボックスが送信待ち (TMEx リセット) のとき、すべての送信レジスタは書き込み保護されます。

このレジスタは、送信リクエスト制御 (ビット 0、リセット値 0) も実装します。

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

ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子

IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。

ビット 20:3 **EXID[17:0]** : 拡張識別子

拡張識別子の最下位ビットです。

ビット 2 **IDE** : 識別子拡張

このビットは、メールボックス内のメッセージの識別子のタイプを定義します。

0 : 標準識別子。

1 : 拡張識別子。

ビット 1 **RTR** : リモート送信リクエスト

0 : データフレーム

1 : リモートフレーム

ビット 0 **TXRQ** : 送信メールボックスリクエスト

対応するメールボックスに送信を要求するために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

## CAN メールボックスデータ長制御/タイムスタンプレジスタ (CAN\_TDTxR) (x = 0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x184、0x194、0x1A4

リセット値 : 0xXXXXX XXXX

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

ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ

このフィールドは、SOF 送信時にキャプチャされた 16 ビットタイマの値を含みます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TGT** : 送信グローバルタイム

このビットは、ハードウェアがタイムトリガ通信モードにある (CAN\_MCR レジスタの TTCM ビットがセットされている) ときのみアクティブとなります。

0 : タイムスタンプ TIME[15:0] は送信されません。

1 : タイムスタンプ TIME[15:0] の値は、8 バイトメッセージの最後の 2 データバイトで送信されます。このとき、データバイト 7 の TIME[7:0] とデータバイト 6 の TIME[15:8] は、CAN\_TDHxR[31:16] レジスタ (DATA6[7:0]、DATA7[7:0]) に書き込まれたデータを置換します。これらの 2 バイトを CAN バス経由で送信するには、DLC を 8 としてプログラミングする必要があります。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **DLC[3:0]** : データ長コード

このフィールドは、データフレーム内のデータバイト数、またはリモートフレームリクエストのデータバイト数を定義します。

メッセージは、DLC フィールドの値に応じて、0~8 個のデータバイトを含むことができます。

## CAN メールボックスデータ下位レジスタ (CAN\_TDLxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x188、0x198、0x1A8

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **DATA3[7:0]** : データバイト 3

メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]** : データバイト 2

メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]** : データバイト 1

メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]** : データバイト 0

メッセージのデータバイト 0 です。

メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

## CAN メールボックスデータ上位レジスタ (CAN\_TDHxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x18C、0x19C、0x1AC

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **DATA7[7:0]** : データバイト 7

メッセージのデータバイト 7 です。

**注 :** このメッセージの TGT と TTCM がアクティブな場合、DATA7 と DATA6 はタイムスタンプの値で置換されます。

ビット 23:16 **DATA6[7:0]** : データバイト 6

メッセージのデータバイト 6 です。

ビット 15:8 **DATA5[7:0]** : データバイト 5

メッセージのデータバイト 5 です。

ビット 7:0 **DATA4[7:0]** : データバイト 4

メッセージのデータバイト 4 です。



## CAN 受信 FIFO メールボックス識別子レジスタ (CAN\_RlRxR) (x=0..1)

アドレスオフセット : 0x1B0、0x1C0

リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

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

ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子

IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。

ビット 20:3 **EXID[17:0]** : 拡張識別子

拡張識別子の最下位ビットです。

ビット 2 **IDE** : 識別子拡張

このビットは、メールボックス内のメッセージの識別子のタイプを定義します。

0 : 標準識別子。

1 : 拡張識別子。

ビット 1 **RTR** : リモート送信リクエスト

0 : データフレーム

1 : リモートフレーム

ビット 0 予約済みであり、リセット値に保持する必要があります。

## CAN 受信 FIFO メールボックスデータ長制御/タイムスタンプレジスタ (CAN\_RDTxR) (x = 0..1)

アドレスオフセット : 0x1B4、0x1C4  
リセット値 : 0xFFFF XXXX

すべての受信レジスタは書き込み保護されます。

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

ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ

このフィールドは、SOF 検出時にキャプチャされた 16 ビットタイマの値を含みます。

ビット 15:8 **FMI[7:0]** : 該当フィルタインデックス

このレジスタは、メールボックスに格納されたメッセージが通過したフィルタの索引を含みます。識別子フィルタリングの詳細については、[セクション 36.7.4 : 識別子フィルタリング \(1259 ページ\)](#) の該当フィルタインデックス項を参照してください。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **DLC[3:0]** : データ長コード

このフィールドは、データフレームに格納されるデータバイト数 (0~8) を定義します。リモートフレームリクエストの場合、このフィールドは 0 です。

## CAN 受信 FIFO メールボックスデータ下位レジスタ (CAN\_RDLxR) (x=0..1)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x1B8、0x1C8

リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA3[7:0]** : データバイト 3  
メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]** : データバイト 2  
メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]** : データバイト 1  
メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]** : データバイト 0  
メッセージのデータバイト 0 です。  
メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

## CAN 受信 FIFO メールボックスデータ上位レジスタ (CAN\_RDHxR) (x=0..1)

アドレスオフセット : 0x1BC、0x1CC

リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA7[7:0]** : データバイト 7  
メッセージのデータバイト 3 です。

ビット 23:16 **DATA6[7:0]** : データバイト 6  
メッセージのデータバイト 2 です。

ビット 15:8 **DATA5[7:0]** : データバイト 5  
メッセージのデータバイト 1 です。

ビット 7:0 **DATA4[7:0]** : データバイト 4  
メッセージのデータバイト 0 です。

## 36.9.4 CAN フィルタレジスタ

### CAN フィルタマスタレジスタ (CAN\_FMR)

アドレスオフセット : 0x200

リセット値 : 0x2A1C 0E01

このレジスタのすべてのビットは、ソフトウェアによってセット/クリアされます。

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

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **CANSB[5:0]** : CAN 開始バンク

これらのビットは、ソフトウェアによってセット/クリアされます。CAN インタフェース (スレーブ) の開始バンクを 1 から 27 の範囲で定義します。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **FINIT** : フィルタ初期化モード

フィルタバンクの初期化モード

0 : アクティブフィルタモード。

1 : フィルタの初期化モード。

### CAN フィルタモードレジスタ (CAN\_FM1R)

アドレスオフセット : 0x204

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN\_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16
				r/w											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0
r/w															

**注 :** [図 429 : フィルタバンクのスケール設定 - レジスタ構造 \(1261 ページ\)](#) を参照してください。

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FBMx** : フィルタモード

フィルタ x のレジスタのモードです。

0 : フィルタバンク x の 2 つの 32 ビットレジスタは識別子マスクモードになっています。

1 : フィルタバンク x の 2 つの 32 ビットレジスタは識別子リストモードになっています。

**注 :** **ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。**



## CAN フィルタスケールレジスタ (CAN\_FS1R)

アドレスオフセット : 0x20C

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN\_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16
				r/w											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0
r/w															

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 FSCx : フィルタスケール設定

これらのビットは、フィルタ 13 - 0 のスケール設定を定義します。

0 : デュアル 16 ビットスケール設定

1 : シングル 32 ビットスケール設定

**注 :** ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

**注 :** [図 429 : フィルタバンクのスケール設定 - レジスタ構造 \(1261 ページ\)](#) を参照してください。

## CAN フィルタ FIFO 割当てレジスタ (CAN\_FFA1R)

アドレスオフセット : 0x214

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN\_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FFA27	FFA26	FFA25	FFA24	FFA23	FFA22	FFA21	FFA20	FFA19	FFA18	FFA17	FFA16
				r/w											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0
r/w															

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 FFAx : フィルタ x のフィルタ FIFO 割り当て

このフィルタを通過するメッセージは、特定の FIFO に保存されます。

0 : フィルタが FIFO 0 に割り当てられます。

1 : フィルタが FIFO 1 に割り当てられます。

**注 :** ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

## CAN フィルタアクティベーションレジスタ (CAN\_FA1R)

アドレスオフセット : 0x21C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	FACT27	FACT26	FACT25	FACT24	FACT23	FACT22	FACT21	FACT20	FACT19	FACT18	FACT17	FACT16
				r/w											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FACT15	FACT14	FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0
r/w															

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FACTx** : アクティブなフィルタ

ソフトウェアがこのビットをセットしてフィルタ x をアクティブにします。フィルタ x レジスタ (CAN\_FxR[0:7]) を変更するには、FACTx ビットをクリアするか、CAN\_FMR レジスタの FINIT ビットをセットする必要があります。

0 : フィルタ x はアクティブではありません。

1 : フィルタ x はアクティブです。

**注 :** ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

## CAN フィルタバンク i レジスタ x (CAN\_FiRx) (i = 0..27, x = 1, 2)

アドレスオフセット : 0x240 から 0x31C

リセット値 : 0xFFFF XXXX

28 個のフィルタバンク (i= 0~27) が存在します。各フィルタバンク i は、2 個の 32 ビットレジスタ (CAN\_FiR[2:1]) で構成されています。

このレジスタを変更できるのは、CAN\_FaRx レジスタの FACTx ビットがクリアされている場合か、CAN\_FMR レジスタの FINIT ビットがセットされている場合に限られます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
r/w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
r/w															

すべての設定において :

ビット 31:0 **FB[31:0]** : フィルタビット

### 識別子

レジスタの各ビットは、予期される識別子の対応するビットのレベルを指定します。

0 : ドミナントビットが予期されます。

1 : リセシブビットが予期されます。

### マスク

レジスタの各ビットは、関連する識別子レジスタのビットが、予期される識別子の対応するビットに一致する必要があるかどうかを指定します。

0 : 無視 (このビットは比較に使用されません)

1 : 一致が必要 (受信識別子のビットのレベルは、フィルタの対応する識別子レジスタで指定されたレベルと同じでなければなりません)

**注 :** [フィルタのスケールとモードの設定に応じて、各レジスタの機能は異なることがあります。フィルタ配置、機能の説明、およびマスクレジスタの関連については、\[セクション 36.7.4 : 識別子フィルタリング \\(1259 ページ\\)\]\(#\) を参照してください。](#)

マスクモードにおけるマスク/識別子レジスタのビット配置は、[識別子リストモードの場合と同じ](#)です。

[フィルタバンクのレジスタ配置/アドレスについては、\[表 226 \\(1291 ページ\\)\]\(#\) を参照してください。](#)

## 36.9.5 bxCAN レジスタマップ

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。オフセット 0x200 から 31C までのレジスタは、CAN1 にのみ存在します。

表 226. bxCAN レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
0x000	CAN_MCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBF	リセット	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値																1	0										0	0	0	0	0	0	1	0													
0x004	CAN_MSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	SAMP	Res.	Res.	Res.																					
	リセット値																						1	1	0	0					0	0	0	1	0													
0x008	CAN_TSR	LOW[2:0]				TIME[2:0]				CODE[1:0]				ABRQ2				Res.	Res.	Res.	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	Res.	Res.															
	リセット値	0	0	0	1	1	1	0	0	0					0	0	0	0	0					0	0	0	0	0																				
0x00C	CAN_RF0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]															
	リセット値																												0	0	0			0	0													
0x010	CAN_RF1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]															
	リセット値																												0	0	0			0	0													
0x014	CAN_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値																0	0	0																													
0x018	CAN_ESR	REC[7:0]								TEC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																															
0x01C	CAN_BTR	SILM	LBKM	Res.	Res.	Res.	Res.	Res.	Res.	SJW[1:0]	Res.	TS2[2:0]		TS1[3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.																					
	リセット値	0	0							0	0		0	1	0	0	0	1	1								0	0	0	0	0	0	0	0	0													
0x020-0x17F		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
0x180	CAN_TI0R	STID[10:0]/EXID[28:18]										EXID[17:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.										
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0													
0x184	CAN_TDT0R	TIME[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.						
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																															
0x188	CAN_TDL0R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]																									
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X													



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

表 226. bxCAN レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x18C	CAN_TDH0R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x190	CAN_T11R	STID[10:0]/EXID[28:18]														EXID[17:0]												IDE	RTR	TXRQ				
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0
0x194	CAN_TDT1R	TIME[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]				
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X									X						X	X	X
0x198	CAN_TDL1R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x19C	CAN_TDH1R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1A0	CAN_TI2R	STID[10:0]/EXID[28:18]														EXID[17:0]												IDE	RTR	TXRQ				
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0
0x1A4	CAN_TDT2R	TIME[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]				
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X									X						X	X	X
0x1A8	CAN_TDL2R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1AC	CAN_TDH2R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1B0	CAN_RI0R	STID[10:0]/EXID[28:18]														EXID[17:0]												IDE	RTR	Res.				
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x1B4	CAN_RDT0R	TIME[15:0]															FM[7:0]							Res.	Res.	Res.	Res.	DLC[3:0]						
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X						X	X	X	X
0x1B8	CAN_RDL0R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1BC	CAN_RDH0R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]											
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1C0	CAN_RI1R	STID[10:0]/EXID[28:18]														EXID[17:0]												IDE	RTR	Res.				
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	



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

表 226. bxCAN レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C4	CAN_RDT1R	TIME[15:0]															FM[7:0]							DLC[3:0]									
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X
0x1C8	CAN_RDL1R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]										
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1CC	CAN_RDH1R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]										
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1D0-0x1FF		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x200	CAN_FMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																					0	0	1	1	1	0						1
0x204	CAN_FM1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x208		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x20C	CAN_FS1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x210		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x214	CAN_FFA1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x218		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x21C	CAN_FA1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x220		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x224-0x23F		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x240	CAN_F0R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x244	CAN_F0R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	



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

表 226. bxCAN レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x248	CAN_F1R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x24C	CAN_F1R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
.	.	.																															
.	.	.																															
.	.	.																															
0x318	CAN_F27R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x31C	CAN_F27R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

## 37 USB On-The-Go フルスピード/ハイスピード (OTG\_FS/OTG\_HS)

### 37.1 概要

Portions Copyright (C) 2004, 2005 Synopsys, Inc. All rights reserved.掲載許可取得済み。

このセクションでは OTG\_FS/OTG\_HS コントローラのアーキテクチャとプログラミングモデルについて説明します。

このセクションでは、次の略記が使用されます。

FS	フルスピード
LS	ロースピード
HS	ハイスピード
MAC	メディアアクセスコントローラ
OTG	On-The-Go
PFC	パケット FIFO コントローラ
PHY	物理層
USB	ユニバーサルシリアル
UTMI	USB 2.0 トランシーバマクロセルインタフェース (UTMI)
UTMI	USB トランシーバマクロセルインタフェース
ULPI	UTMI + ローピンインタフェース
ADP	接続検出プロトコル
LPM	リンク電源管理
BCD	バッテリー充電検出回路
HNP	ホストネゴシエーションプロトコル
SRP	セッションリクエストプロトコル

このセクションでは、参照文献として以下の文書が使用されています。

- USB On-The-Go Supplement, Revision 1.3
- USB On-The-Go Supplement, Revision 2.0
- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- USB 2.0 ECN 用の正誤表 : Link Power Management (LPM) - 7/2007

USB OTG は、デバイスとホストの両方の機能をサポートするデュアルロールデバイス (DRD) コントローラであり、*On-The-Go Supplement to the USB 2.0 Specification* に完全に準拠しています。USB OTG は、*USB 2.0 Specification* に完全に準拠したホスト専用またはデバイス専用のコントローラとして設定することも可能です。ホストモードの場合、OTG\_HS はハイスピード転送 (HS、480 M ビット/秒)、OTG\_FS はフルスピード (FS、12 M ビット/秒) とロースピード (LS、1.5 M ビット/秒) の転送をサポートします。デバイスモードの場合は、OTG\_HS はハイスピード転送 (HS、480 M ビット/秒)、OTG\_FS はフルスピード (FS、12 M ビット/秒) をサポートします。USB OTG は HNP と SRP の両方をサポートします。必要な外部デバイスは、OTG モードで使用される  $V_{BUS}$  用のチャージポンプのみです。



## 37.2 USB\_OTG の主な機能

主な機能は、標準機能、ホストモード機能、およびデバイスモード機能の3つのカテゴリに分けて考えることができます。

### 37.2.1 標準機能

OTG\_FS/OTG\_HS インタフェースの機能概要を次に示します。

- Universal Serial Bus Specification Rev 2.0 に準拠した USB-IF です。
- OTG HS は 次の3つの PHY インタフェースをサポートします。
  - フルスピード PHY を内蔵
  - 外部のフルスピード I2C PHY を対象とする I2C インタフェース
  - 外部のハイスピード PHY を対象とする ULPI インタフェース
- On-The-Go Supplement Rev 1.3 Specification に詳細が規定されているオプションの On-The-Go (OTG) プロトコルの完全サポート (PHY) を含みます。
  - A/B デバイス識別 (ID ライン) を統合的にサポートしています。
  - ホストネゴシェーションプロトコル (HNP) およびセッションリクエストプロトコル (SRP) を統合的にサポートしています。
  - OTG アプリケーションでバッテリーを節約するために、ホストで  $V_{BUS}$  をオフすることができます。
  - 内部コンパレータによって OTG に  $V_{BUS}$  のレベルを監視させることができます。
  - ホスト-ペリフェラルのダイナミックな役割切り替えをサポートします。
- 次の動作を行うようにソフトウェアでの設定が可能です。
  - SRP 対応の USB FS/HS ペリフェラル (B デバイス)
  - SRP 対応の USB FS/HS/LS ホスト (A デバイス)
  - USB On-The-Go フルスピードデュアルロールデバイス
- 以下の要素により FS/HS SOF および LS キープアライブをサポートします。
  - SOF パルス PAD 接続性
  - タイマ (TIMx) に対する SOF パルス内部接続
  - 設定可能なフレーム周期
  - 設定可能なフレームの終わりの割り込み
- OTG HS は、シェアホールディングサポートを持つ DMA を内蔵し、DMA モードで AHB バーストタイプをソフトウェアから選択できます。
- USB サスペンド中のシステム停止、デジタルコア、PHY、および DFIFO 電源管理回路内部のクロックドメインの電源オフなど、節電機能を内蔵しています。
- 以下のような高度な FIFO 制御を備えた 1.25[FS] / 4[HS] KB の専用 RAM を備えています。
  - RAM 空間を複数の FIFO に分割して、RAM を柔軟かつ効率的に使用できるように設定できます。
  - 各 FIFO は複数のパケットを保持できます。
  - 動的なメモリ割り当てが可能です。
  - FIFO を2のべき乗でないサイズに設定して、連続的なメモリ位置の使用が可能です。
- システムの介入なしで、最大1フレーム (1ms) の間、最大 USB バンド幅を保証します。

## 37.2.2 ホストモード機能

ホストモードでの OTG\_FS/OTG\_HS インタフェースの主な機能と要件は次の通りです。

- V<sub>BUS</sub> 電圧発生用の外部チャージポンプを持っています。
- 最大 12[FS] / 16[HS] 個のホストチャンネル（パイプ）を持っています。各チャンネルはどんなタイプの USB 転送も割り当てられるように動的に設定可能です。
- 次を保持するハードウェアスケジューラを内蔵しています。
  - 最大 12[FS] / 16[HS] 個の割り込みに加えて、周期的ハードウェアキュー内でのアイソクロナス転送リクエスト
  - 最大 12[FS] / 16[HS] 個の制御に加えて、非周期的ハードウェアキュー内でのバルク転送リクエスト
- USB データ RAM の効率的な使用のために、共有 Rx FIFO、周期的 Tx FIFO、および非周期的 Tx FIFO を管理します。

## 37.2.3 ペリフェラルモード機能

ペリフェラルモードでの OTG\_FS/OTG\_HS インタフェースの主要機能を次に示します。

- 1つの双方向制御エンドポイント 0 を持っています。
- バルク、割り込み、またはアイソクロナス転送をサポートするために、5[FS] / 7[HS] つの設定可能な IN エンドポイント（EP）を持っています。
- バルク、割り込み、またはアイソクロナス転送をサポートするために、5[FS] / 7[HS] つの設定可能な OUT エンドポイントを持っています。
- USB データ RAM の効率的な使用のために、共有 RX FIFO と Tx-OUT FIFO を管理します。
- アプリケーションの負荷を軽減するために、最大 6[FS] / 8[HS] つの専用 Tx-IN FIFO（アクティブな IN EP に対して 1 つずつ）を管理します。
- ソフト切断機能をサポートしています。

## 37.3 USB\_OTG の実装

表 227. STM32F75xxxの USB\_OTG 実装<sup>(1)</sup>

USB の機能	OTG_FS	OTG_HS
デバイスの双方向エンドポイント (EP0 を含む)	6	8
ホストモードチャンネル	12	16
専用 SRAM のサイズ	1.2 KB	4 KB
USB 2.0 リンク電源管理 (LPM) のサポート	X	
サポートされている OTG レビジョン	1.3 & 2.0	
接続検出プロトコル (ADP) のサポート	-	
バッテリー充電検出 (BCD) のサポート	-	

1. "X" はサポートされています。"-" はサポートされていません。

## 37.4 USB OTG の機能詳細

### 37.4.1 USB OTG ブロック図

図 437. OTG フルスピードブロック図

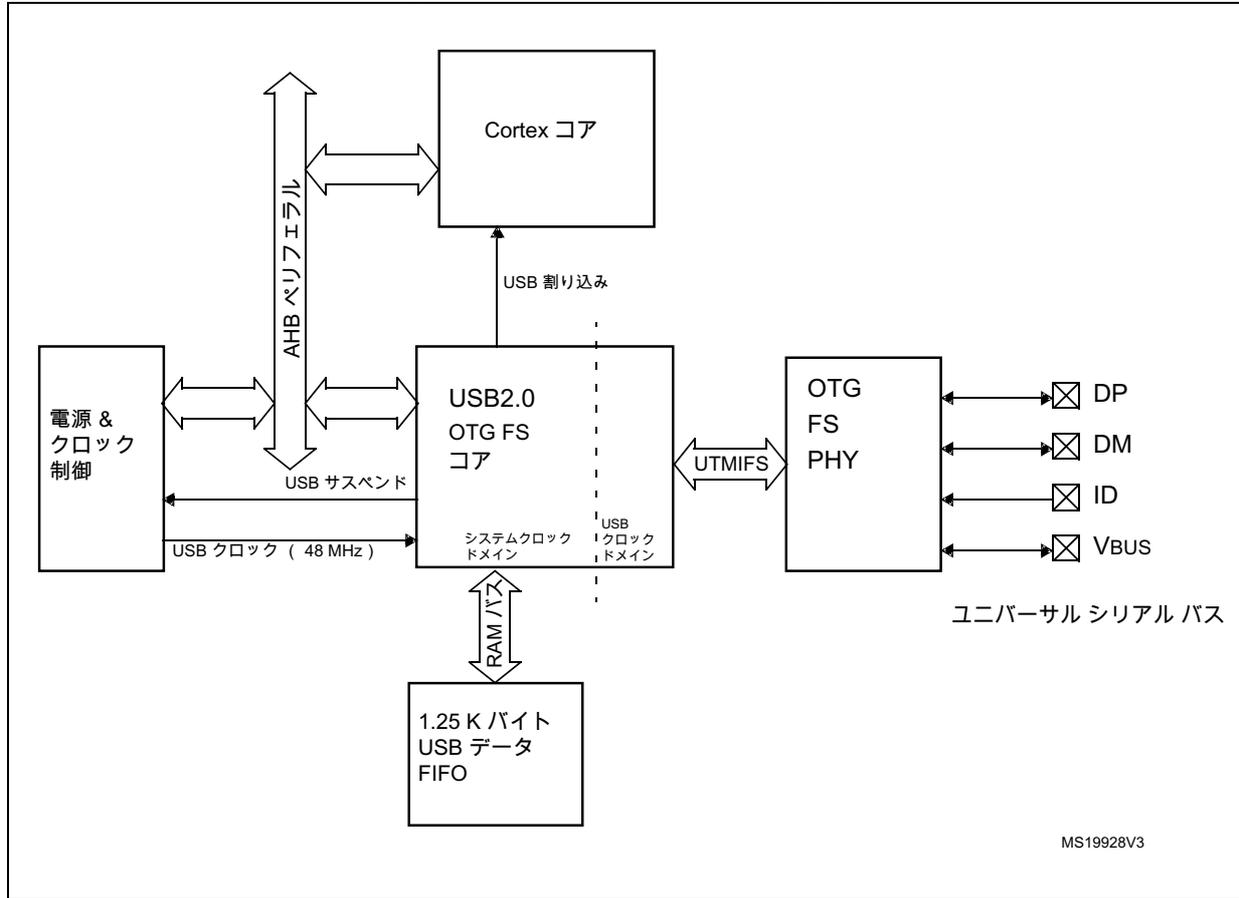
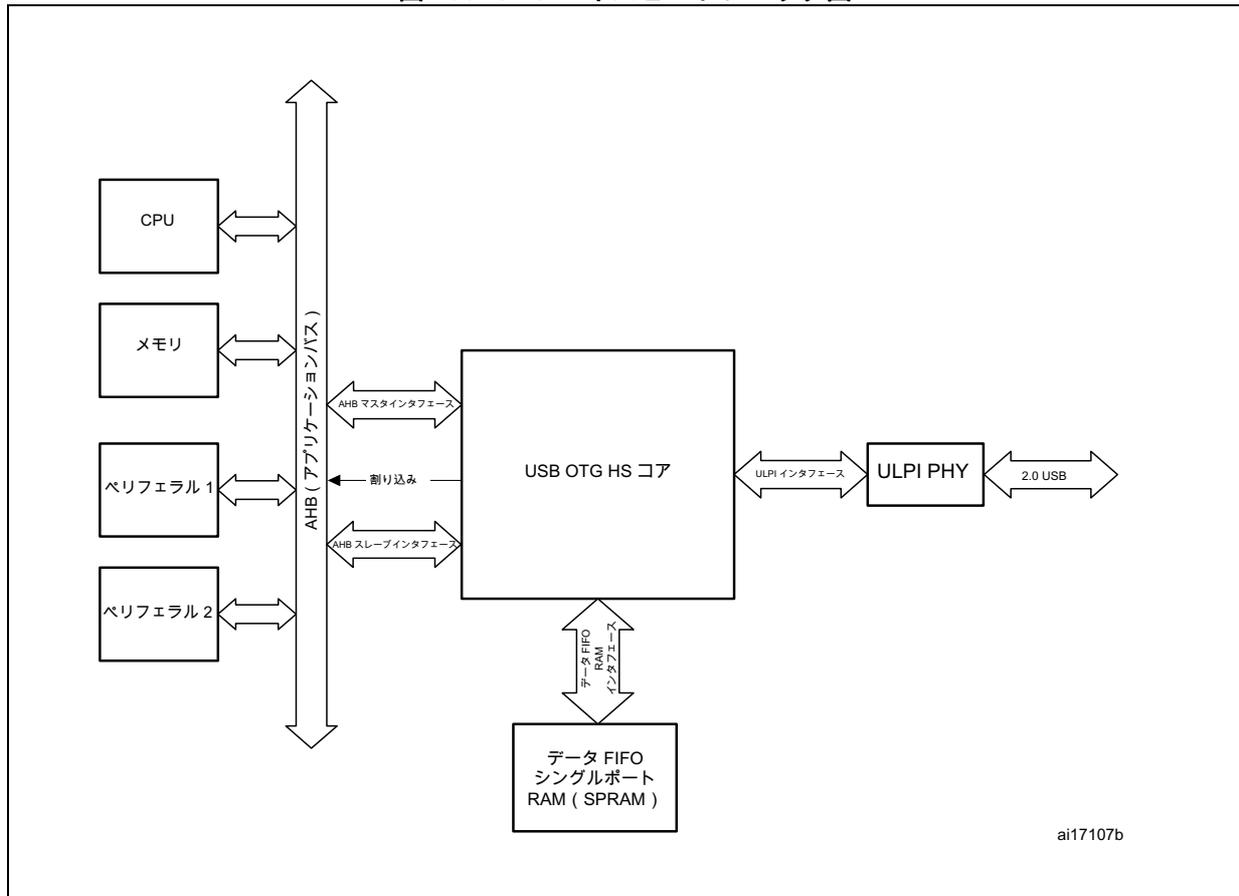


図 438. OTG ハイスピードブロック図



ai17107b

## 37.4.2 OTG コア

USB OTG は、外部水晶発振器を使用して、リセットおよびクロックコントローラ (RCC) から 48 MHz  $\pm$  0.25 % のクロックを受け取ります。USB クロックは、フルスピード (12 M ビット/秒) で 48 MHz ドメインを駆動するために使用され、OTG コアを設定する前に有効にされなければなりません。

CPU は、AHB ペリフェラルバスを使用して OTG コアレジスタに対する読み書きを実行します。USB イベントは 1 本の USB OTG 割り込みラインによって通知されます ([セクション 37.13 : OTG\\_FS/OTG\\_HS 割り込み](#)を参照)。

CPU は、専用の OTG 位置 (プッシュレジスタ) に 32 ビットのワードを書き込むことによって、USB を介してデータを提供します。データは USB データ RAM 内に設定された Tx データ FIFO に自動的に格納されます。各 IN エンドポイント (ペリフェラルモードの場合) または OUT チャネル (ホストモードの場合) に対して 1 つの Tx FIFO プッシュレジスタがあります。

CPU は、専用の OTG アドレス (ポップレジスタ) から 32 ビットのワードを読み出すことによって、USB からデータを受け取ります。データは、1.25[FS] / 4[HS] KB の USB データ RAM 内に設定された共有 Rx FIFO から自動的に取得されます。各 OUT エンドポイントまたは IN チャネルに対して 1 つの Rx FIFO ポップレジスタがあります。

USB プロトコル層は、シリアルインタフェースエンジン (SIE) によって駆動され、オンチップ物理層 (PHY) 内のトランシーバモジュールによって USB 上で直列化 (シリアライズ) されます。この PHY とは、I2C インタフェースを使用する外部 OTG\_HS PHY または外部 OTG\_FS PHY です。

## 37.4.3 フルスピード OTG PHY

組み込みのフルスピード OTG PHY は、OTG FS コアによって制御され、UTMI+ バス (UTMIFS) のフルスピードサブセットを通じて USB 制御信号およびデータ信号を運びます。USB 接続の物理サポートを提供します。

フルスピード OTG PHY には、以下のコンポーネントが含まれます。

- ホストとデバイスの両方で使用される FS/LS トランシーバモジュール。シングルエンド USB ライン上で送受信を直接駆動します。
- A/B デバイス識別のための ID ラインをサンプリングするために使用される内蔵の ID プルアップ抵抗。
- デバイスの現在の役割に依存して OTG\_FS コアによって制御される 組み込みの DP/DM プルアップおよびプルダウン抵抗。ペリフェラルとして、DP プルアップ抵抗を有効にし、フルスピードペリフェラル接続を知らせることができます。信号を送るタイミングは、 $V_{BUS}$  が有効なレベルであると判断されてすぐです (B セッションが有効な場合)。ホストモードでは DP と DM の両方でプルダウン抵抗が有効にされます。プルアップ抵抗とプルダウン抵抗は、デバイスの役割がホストネゴシエーションプロトコル (HNP) を通じて変更されたとき、動的に切り替えられます。
- プルアップ/プルダウン抵抗 ECN 回路。DP プルアップ回路は、USB Rev2.0 に適用される抵抗の技術変更通知 (Engineering Change Notice) に従って、OTG\_FS から個別に制御される 2 つの抵抗で構成されます。DP プルアップの動的なトリミングは、ノイズの低減と Tx/Rx 信号品質の向上を可能にします。
- $V_{BUS}$  レベル検出用コンパレータ。 $V_{BUS}$  の有効性、A-B セッションの有効性、およびセッション終了電圧閾値を検出するために使用され、ヒステリシスを有します。これらは、セッションリクエストプロトコル (SRP) の駆動、有効な起動およびセッション終了条件の検出、および USB 操作中の  $V_{BUS}$  電源の定期的な監視のために使用されます。
- $V_{BUS}$  パルス駆動回路。SRP 中に抵抗により  $V_{BUS}$  を充電/放電するために使用します (弱駆動)。

**注意：** USB OTG FS ペリフェラルの正常動作を保証するためには、AHB 周波数を 14.2 MHz より高くする必要があります。

**注：** このセクションの内容は、USB OTG FS にのみ適用されます。

## 37.4.4 内蔵フルスピード OTG PHY

フルスピード OTG PHY には、以下のコンポーネントが含まれます。

- ホストとデバイスの両方で使用される FS/LS トランシーバモジュール。シングルエンド USB ライン上で送受信を直接駆動します。
- A/B デバイス識別のための ID ラインをサンプリングするために使用される組み込みの ID プルアップ抵抗。
- デバイスの現在の役割に依存して OTG\_HS コアによって制御される内蔵の DP/DM プルアップおよびプルダウン抵抗。ペリフェラルとして、DP プルアップ抵抗を有効にし、フルスピードペリフェラル接続を知らせます。信号を送るタイミングは、 $V_{BUS}$  が有効なレベルであると判断されてすぐです (B セッションが有効な場合)。ホストモードでは DP と DM の両方でプルダウン抵抗が有効にされます。プルアップ抵抗とプルダウン抵抗は、ペリフェラルの役割がホストネゴシエーションプロトコル (HNP) を通じて変更されたとき、動的に切り替えられます。
- プルアップ/プルダウン抵抗 ECN 回路。DP プルアップは、USB Rev2.0 に適用される抵抗の技術変更通知 (Engineering Change Notice) に従って、OTG\_HS から個別に制御される 2 つの抵抗で構成されます。DP プルアップの動的なトリミングは、ノイズの低減と Tx/Rx 信号品質の向上を可能にします。

- $V_{BUS}$  レベル検出用コンパレータ。  $V_{BUS}$  の有効性、A-B セッションの有効性、およびセッション終了電圧閾値を検出するために使用されるヒステリシスを有します。これらは、セッションリクエストプロトコル (SRP) の駆動、有効な起動およびセッション終了条件の検出、および USB 操作中の  $V_{BUS}$  電源の定常的な監視のために使用されます。
- $V_{BUS}$  パルス駆動回路。 SRP 中に抵抗により  $V_{BUS}$  を充電/放電するために使用します (弱駆動)。

USB OTG HS ペリフェラルの正常動作を保証するためには AHB 周波数を 30 MHz より高くする必要があります。

**注:** このセクションの内容は、USB OTG HS にのみ適用されます。

### 37.4.5 ハイスピード OTG PHY

USB OTG HS コアは、外部 HS PHY を接続する ULPI インタフェースを内蔵しています。

**注:** このセクションの内容は、USB OTG HS にのみ適用されます。

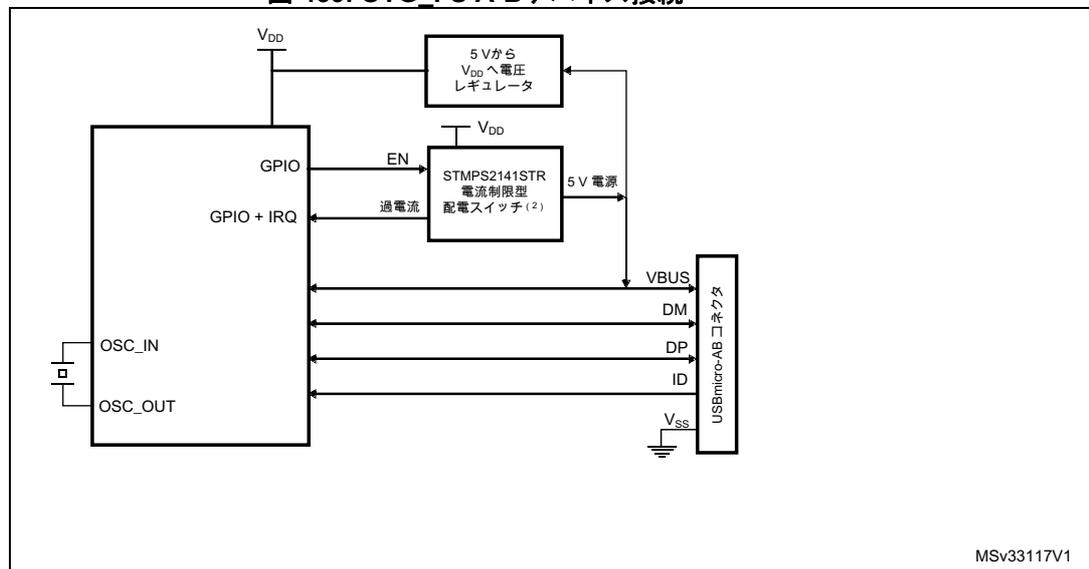
### 37.4.6 I2C インタフェースを使用する外部フルスピード OTG PHY

USB OTG HS コアは、外部 FS PHY を接続する I2C インタフェースを内蔵しています。

**注:** このセクションの内容は、USB OTG HS にのみ適用されます。

## 37.5 OTG デュアルロールデバイス (DRD)

図 439. OTG\_FS A-B デバイス接続



1. VBUS で電源が供給されるデバイスを設定するときだけ必要な外部電圧レギュレータ
2. アプリケーションが VBUS で電源が供給されるデバイスをサポートしなければならない場合のみ必要な STMP2141STR。アプリケーションボード上で 5V が使用可能な場合、基本的なパワースイッチを使用することができます。

## 37.5.1 ID ラインの検出

ホストかペリフェラル（デフォルト）かの役割は、ID 入力ピンに応じて割り当てられます。ID ラインのステータスは、USB 端子への挿入時に、USB ケーブルのどちら側がマイクロ AB レセプタクルに接続されたかによって決まります。

- USB ケーブルの B 側がフローティング ID ワイヤに接続された場合、内蔵のプルアップ抵抗が ID ラインのハイレベルを検出して、デフォルトのペリフェラルの役割が確立されます。この設定では、OTG\_FS/OTG\_HS はセクション 6.8.2 で説明する標準 FSM に準拠しています（On-The-Go Specification Rev1.3 supplement to the USB2.0 の On-The-Go B デバイス）。
- USB ケーブルの A 側がアース付き ID ラインに接続された場合、OTG\_FS/OTG\_HS は、ホストのソフトウェア初期化のために ID ラインステータス変化割り込み（OTG\_GINTSTS の CIDSCHG ビット）を発行して、自動的にホストの役割に切り替わります。この設定では、OTG\_FS/OTG\_HS はセクション 6.8.1 で説明する標準 FSM に準拠しています（On-The-Go Specification Rev1.3 supplement to the USB2.0 の On-The-Go A デバイス）。

## 37.5.2 HNP デュアルロールデバイス

グローバル USB 設定レジスタの HNP 対応ビット（OTG\_GUSBCFG の HNPCAP ビット）は、OTG\_FS/OTG\_HS コアがホストネゴシエーションプロトコル（HNP）に従って、役割を A ホストから A ペリフェラルに（またその逆）、または B ペリフェラルから B ホスト（またその逆）に動的に変更できるようにします。現在のデバイスのステータスは、グローバル OTG 制御およびステータスレジスタのコネクタ ID ステータスビット（OTG\_GOTGCTL の CIDSTS ビット）とグローバル割り込みおよびステータスレジスタの現在の動作モードビット（OTG\_GINTSTS の CMOD ビット）の値の組み合わせによって読み出すことができます。

HNP プログラミングモデルの詳細は、[セクション 37.16 : OTG\\_FS/OTG\\_HS プログラミングモデル](#)に記載されています。

## 37.5.3 SRP デュアルロールデバイス

グローバル USB 設定レジスタの SRP 対応ビット（OTG\_GUSBCFG の SRPCAP ビット）は、OTG\_FS/OTG\_HS コアが A デバイスの節電のために  $V_{BUS}$  の生成を停止できるようにします。A デバイスは常に  $V_{BUS}$  を駆動する役割を持っている事に注意してください。OTG\_FS/OTG\_HS の役割がホストかペリフェラルかは関係ありません。

SRP A/B デバイスのプログラミングモデルの詳細は、[セクション 37.16 : OTG\\_FS/OTG\\_HS プログラミングモデル](#)に記載されています。

## 37.6 USB ペリフェラル

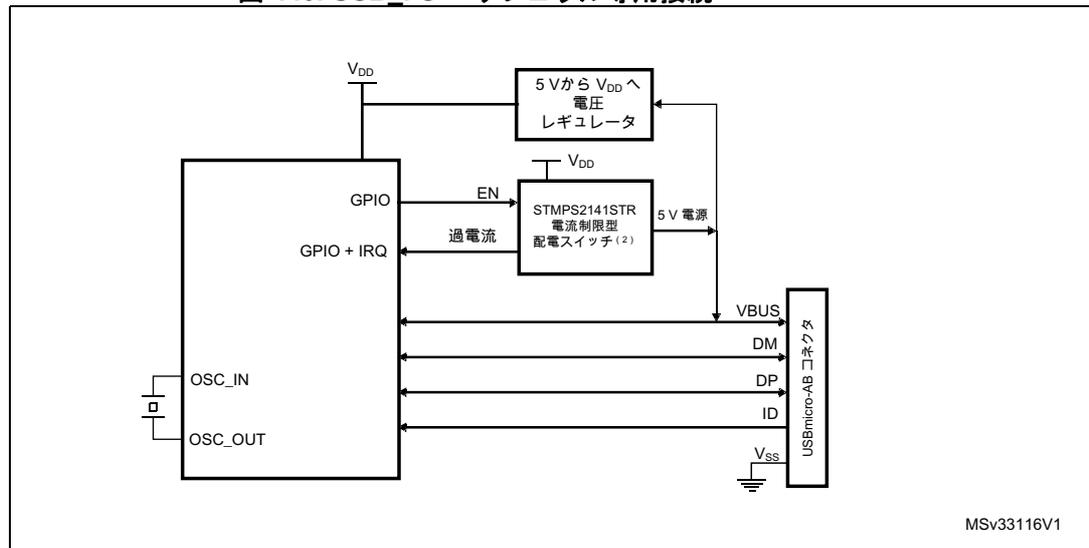
このセクションでは、USB ペリフェラルモードでの OTG\_FS/OTG\_HS の機能について説明します。OTG\_FS/OTG\_HS は、以下の環境で USB ペリフェラルとして機能します。

- OTG B ペリフェラル
  - USB ケーブルの B 側が挿入された場合の OTG B デバイスのデフォルトの状態
- OTG A ペリフェラル
  - HNP が OTG\_FS/OTG\_HS の役割をペリフェラルに切り替えた後の OTG A デバイスの状態
- Bデバイス
  - ID ラインが存在し、機能していて、USB ケーブルの B 側に接続され、かつ、グローバル USB 設定レジスタの HNP 対応ビット (OTG\_GUSBCFG の HNPCAP ビット) がクリアされている場合 On-The-Go Rev1.3 の 6.8.3 項を参照。
- ペリフェラルのみ (図 440 : USB\_FS ペリフェラル専用接続を参照)

セクション 37.15.4 : OTG USB 設定レジスタ (OTG\_GUSBCFG) の強制デバイスモードビット (FDMOD) が 1 にセットされると、OTG\_FS/OTG\_HS コアは USB ペリフェラル専用として機能するよう強制されます (On-The-Go Rev1.3 の 6.8.3 項を参照)。この場合、ID ラインは、USB コネクタ上に存在していても無視されます。

**注 :** B デバイスまたはペリフェラル専用設定の場合に電源がバスから供給されるデバイスを実装するには、 $V_{DD}$  テップ供給 ( $V_{BUS}$ からの) を生成する外部レギュレータを追加する必要があります。

図 440. USB\_FS ペリフェラル専用接続



1. バスから電源が供給されるデバイスを構築するには、レギュレータを使用します。

## 37.6.1 SRP 対応ペリフェラル

グローバル USB 設定レジスタの SRP 対応ビット (OTG\_GUSBCFG の SRPCAP ビット) は、OTG\_FS/OTG\_HS がセッションリクエストプロトコル (SRP) をサポートできるようにします。この方法により、USB セッションがサスペンド状態にある間、 $V_{BUS}$  の供給を停止して、リモート A デバイスの消費電力を節約できます。

SRP ペリフェラルモードのプログラミングモデルの詳細は、[B デバイスセッションリクエストプロトコル](#)のセクションに記載されています。

## 37.6.2 ペリフェラルの状態

### 電源が供給された状態

$V_{BUS}$  入力は、USB ペリフェラルが電源供給状態に入ることが許される B セッション有効電圧を検出します (USB 2.0 の 9.1 項を参照)。次に、

OTG\_FS/OTG\_HS は、DP プルアップ抵抗を自動的に接続して、フルスピードデバイス接続をホストに知らせ、セッションリクエスト割り込みを生成して (OTG\_GINTSTS の SRQINT ビット)、電源供給状態を通知します。

$V_{BUS}$  入力は、また、USB 動作中にホストによって適切な  $V_{BUS}$  レベルが供給されることを保証します。 $V_{BUS}$  電源が B セッションの有効値以下に低下したことが検出された場合 (たとえば、電源の変動があった場合や、ホストポートの電源がオフとなった場合など)、OTG\_FS/OTG\_HS は自動的に切断され、セッション終了検出割り込み (OTG\_GOTGINT の SEDET ビット) が生成されて、OTG\_FS/OTG\_HS が電源供給状態でなくなったことを通知します。

電源供給状態では、OTG\_FS/OTG\_HS は、ホストからのリセットシグナリングの受信を待っています。他の USB 操作を実行することはできません。リセットシグナリングを受け取ると、リセット検出割り込み (OTG\_GINTSTS の USBRST ビット) が生成されます。リセットシグナリングが完了すると、エニュメレーション終了割り込み (OTG\_GINTSTS の ENUMDNE ビット) が生成され、OTG\_FS/OTG\_HS はデフォルト状態に入ります。

### ソフト切断

電源供給状態は、ソフト切断機能によってソフトウェアで終了させることができます。DP プルアップ抵抗は、デバイス制御レジスタのソフト切断ビット (OTG\_DCTL の SDIS ビット) をセットすることによって取り除かれ、USB ケーブルが実際にはホストポートから取り除かれていなくても、ホスト側にデバイス切断検出割り込みを生成します。

### デフォルト状態

デフォルト状態では、OTG\_FS/OTG\_HS はホストからの SET\_ADDRESS コマンドの受信を待っています。他の USB 操作を実行することはできません。有効な SET\_ADDRESS コマンドが USB 上でデコードされると、アプリケーションは対応する番号をデバイス設定レジスタのデバイスアドレスフィールドに書き込みます (OTG\_DCFG の DAD ビット)。その場合、OTG\_FS/OTG\_HS はアドレス状態に入り、設定された USB アドレスでホストトランザクションに応答する準備が整います。

### サスペンド状態

OTG\_FS/OTG\_HS ペリフェラルは、USB の動作状態を定期的に監視します。USB のアイドル状態が 3 ms 続くと、アーリーサスペンド割り込み (OTG\_GINTSTS の ESUSP ビット) が発行され、適切な場合は 3 ms 後にサスペンド割り込み (OTG\_GINTSTS の USBUSP ビット) によって確認されます。その場合、デバイスステータスレジスタのデバイスサスペンドビット (OTG\_DSTS の SUSPSTS ビット) が自動的にセットされ、OTG\_FS/OTG\_HS はサスペンド状態に入ります。

サスペンド状態は、デバイス自身によって終了することもできます。この場合、アプリケーションはデバイス制御レジスタのリモートウェイクアップシグナリングビット (OTG\_DCTL の RWUSIG ビット) をセットし、1~15 ms 後にクリアします。

ホストからレジャームシグナリングが検出されると、レジャーム割り込み (OTG\_GINTSTS の WKUPINT ビット) が生成され、デバイスサスペンドビットは自動的にクリアされます。

## 37.6.3 ペリフェラルエンドポイント

OTG\_FS/OTG\_HS コアは、次の USB エンドポイントをインスタンス化します。

- 制御エンドポイント 0:
  - 双方向であり、制御メッセージのみを取り扱います。
  - 入力および出力トランザクションを処理する個別のレジスタセットを備えています。
  - 固有の制御レジスタ (OTG\_DIEPCTL0/OTG\_DOEPCTL0)、転送設定レジスタ (OTG\_DIEPTSIZ0/OTG\_DOEPTSIZ0)、およびステータス割り込みレジスタ (OTG\_DIEPINT0/OTG\_DOEPINT0) を備えています。制御および転送サイズレジスタ内で使用可能なビットのセットは、他のエンドポイントとは少し異なります。
- 5[FS]/7[HS] IN エンドポイント
  - それぞれ、アイソクロナス転送、バルク転送、または割り込み転送タイプをサポートするように設定できます。
  - また、それぞれに固有の制御レジスタ (OTG\_DIEPCTLx)、転送設定レジスタ (OTG\_DIEPTSIZx)、およびステータス割り込みレジスタ (OTG\_DIEPINTx) を備えています。
  - デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG\_DIEPMSK) を使用して、すべての IN エンドポイント (EP0 も含む) 上で 1 種類のエンドポイント割り込みソースを有効/無効にできます。
  - 現在のフレーム内で転送が完了していないアイソクロナス IN エンドポイントが少なくとも 1 つあるときにアサートされる、不完全アイソクロナス IN 転送割り込み (OTG\_GINTSTS の IISOIXFR ビット) をサポートします。この割り込みは、周期的フレーム割り込み (OTG\_GINTSTS の EOPF) の終わりとともにアサートされます。
- 5[FS]/7[HS] OUT エンドポイント
  - それぞれ、アイソクロナス転送、バルク転送、または割り込み転送タイプをサポートするように設定できます。
  - また、それぞれに固有の制御レジスタ (OTG\_DOEPCTLx)、転送設定レジスタ (OTG\_DOEPTSIZx)、およびステータス割り込みレジスタ (OTG\_DOEPINTx) を備えています。
  - デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG\_DOEPMSK) を使用して、すべての OUT エンドポイント (EP0 も含む) 上で 1 種類のエンドポイント割り込みソースを有効/無効にすることができます。
  - 現在のフレーム内で転送が完了していないアイソクロナス OUT エンドポイントが少なくとも 1 つあるときにアサートされる、不完全アイソクロナス OUT 転送割り込み (OTG\_GINTSTS の INCOMPISOOUT ビット) をサポートします。この割り込みは、周期的フレーム割り込み (OTG\_GINTSTS の EOPF) の終わりとともにアサートされます。

## エンドポイント制御

- 以下のエンドポイント制御は、デバイスエンドポイント-x IN/OUT 制御レジスタ (OTG\_DIEPCTLx/OTG\_DOEPCTLx) を通じて、アプリケーションから使用できます。
  - エンドポイント有効/無効
  - 現在の設定でのエンドポイントのアクティブ化
  - USB 転送タイプ (アイソクロナス、バルク、割り込み) のプログラミング
  - サポートされるパケットサイズのプログラミング
  - IN エンドポイントに関連する Tx FIFO 番号のプログラミング
  - 期待される、または送信されるデータ 0/データ 1 PID (バルク/割り込みのみ) のプログラミング
  - トランザクションが送受信される偶数/奇数フレームのプログラミング (アイソクロナス転送のみ)
  - オプションで、FIFO のステータスに関係なく、常にホストに否定応答する NAK ビットのプログラミング
  - オプションで、エンドポイントに対するホストトークンを常に停止する STALL ビットのプログラミング
  - オプションで、OUT エンドポイントが受信データの CRC フィールドをチェックしないようにする SNOOP モードのプログラミング

## エンドポイント転送

アプリケーションは、デバイスエンドポイント-x 転送サイズレジスタ (OTG\_DIEPTSIZx/OTG\_DOEPTSIZx) を使用して、転送サイズパラメータをプログラムし、転送ステータスを読み出すことができます。エンドポイント制御レジスタのエンドポイント有効化ビットをセットする前にプログラミングを行う必要があります。エンドポイントが有効になると、これらのフィールドは、OTG\_FS/OTG\_HS コアが現在の転送ステータスで更新すると、読み出し専用になります。

以下の転送パラメータがプログラム可能です。

- 転送サイズ (バイト単位)
- 全体の転送サイズを構成するパケットの数

## エンドポイントのステータス/割り込み

デバイスエンドポイント-x 割り込みレジスタ (OTG\_DIEPINTx/OTG\_DOEPINTx) は、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。アプリケーションは、コア割り込みレジスタの OUT エンドポイント割り込みビットまたは IN エンドポイント割り込みビット (OTG\_GINTSTS の OEPINT ビット、または OTG\_GINTSTS の IEPINT ビット) がセットされたとき、これらのレジスタを読み出さなければなりません。アプリケーションがこれらのレジスタを読み出すためには、まず、デバイスオールエンドポイント割り込み (OTG\_DAIN) レジスタを読み出して、デバイスエンドポイント-x 割り込みレジスタの正確なエンドポイント番号を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG\_DAIN および OTG\_GINTSTS レジスタの対応するビットをクリアする必要があります。

ペリフェラルコアは、以下のステータスチェックおよび割り込み生成を行います。

- データ転送がアプリケーション側 (AHB) と USB 側の両方で完了したことを示す転送完了割り込み
- セットアップステージが終了しているかどうか (制御 OUT のみ)
- 関連する送信 FIFO が半分エンプティか完全エンプティか (IN エンドポイント)
- NAK 応答がホストに送信されたかどうか (アイソクロナス IN のみ)
- Tx FIFO がエンプティのとき IN トークンが受け取られたかどうか (パルク IN/割り込み INのみ)
- エンドポイントがまだ有効になっていないときに OUT トークンを受け取ったかどうか
- バブルエラー条件が検出されたかどうか
- アプリケーションによるエンドポイントの無効化が可能かどうか
- アプリケーションによるエンドポイント NAK が可能かどうか (アイソクロナス IN のみ)
- 3つを超える連続したセットアップパケットが受信されたかどうか (制御 OUT のみ)
- タイムアウト条件が検出されたかどうか (制御 IN のみ)
- アイソクロナス OUT パケットが割り込みを生成せずにドロップされたかどうか

## 37.7 USB ホスト

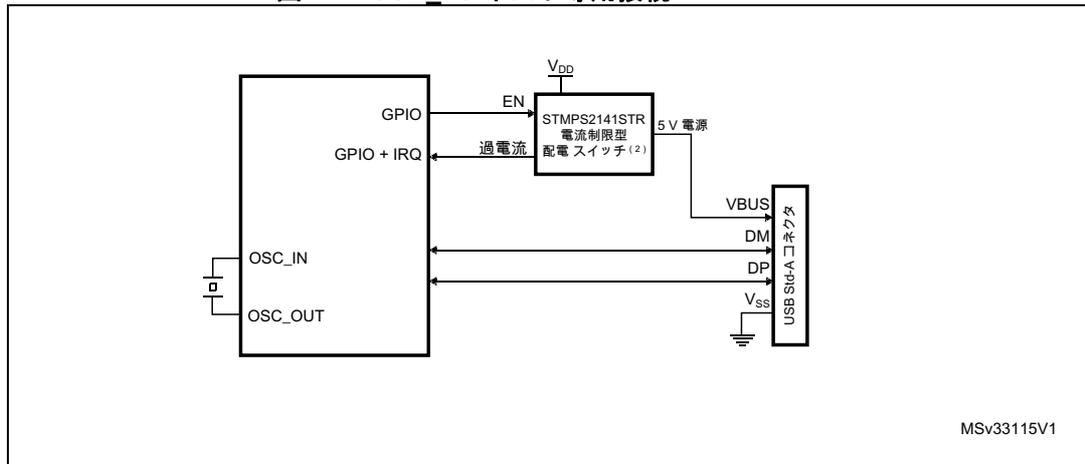
このセクションでは、USB ホストモードでの OTG\_FS/OTG\_HS の機能について説明します。OTG\_FS/OTG\_HS は、以下の環境で USB ホストとして機能します。

- OTG A ホスト
  - USB ケーブルの A 側が挿入されたときの OTG A デバイスのデフォルト状態
- OTG B ホスト
  - HNP がホストの役割に切り替わった後の OTG B デバイス
- A デバイス
  - ID ラインが存在し、機能していて、USB ケーブルの A 側に接続され、かつ、グローバル USB 設定レジスタの HNP 対応ビット (OTG\_GUSBCFG の HNPCAP ビット) がクリアされている場合。組み込みのプルダウン抵抗が自動的に DP/DM ライン上にセットされます。
- ホストのみ
  - [37.15.4](#) グローバル USB 設定レジスタの強制ホストモードビット (OTG\_GUSBCFG の FHMOD ビット) は、OTG\_FS/OTG\_HS コアを強制的に USB ホスト専用として機能させます。この場合、ID ラインは、USB コネクタ上に存在していても無視されます。組み込みのプルダウン抵抗が自動的に DP/DM ライン上にセットされます。

**注：** *オンチップの 5 V  $V_{BUS}$  の生成はサポートされていません。このため、チャージポンプ、または 5 V がアプリケーションボード上で使用可能な場合は基本的な電源スイッチを外部で追加して、5 V の  $V_{BUS}$  ラインを駆動しなければなりません。外付けチャージポンプは、任意の GPIO 出力からでも駆動できます。これは、OTG A ホスト、A デバイス、およびホスト専用設定の場合に必要です。*



図 441. USB\_FS ホスト専用接続



1. アプリケーションが  $V_{BUS}$  で電源が供給されるデバイスをサポートしなければならない場合のみ必要な STMP2141STR。アプリケーションボード上で 5 V が使用可能な場合、基本的なパワースイッチを使用することができます。
2.  $V_{DD}$  範囲は 2.0~3.6 V です。

## 37.7.1 SRP 対応ホスト

SRP のサポートはグローバル USB 設定レジスタの SRP 対応ビット (OTG\_GUSBCFG の SRPCAP ビット) を通じて使用可能です。SRP 機能が有効なときには、ホストは、USB セッションがサスペンド状態にある間、 $V_{BUS}$  の電力供給を停止することによって、消費電力を節約できます。

SRP ホストモードのプログラミングモデルの詳細は、[A デバイスセッションリクエストプロトコル](#) のセクションに記載されています。

## 37.7.2 USB ホストの状態

### ホストポートパワー

オンチップの 5 V  $V_{BUS}$  の発生はサポートされていません。このため、チャージポンプ、または 5 V がアプリケーションボード上で使用可能な場合は基本的な電源スイッチを外部で追加して、5 V の  $V_{BUS}$  ラインを駆動しなければなりません。外付けチャージポンプは、任意の GPIO 出力からでも駆動できます。アプリケーションが選んだ GPIO を使用して  $V_{BUS}$  に電源を供給することにした場合は、ホストポート制御およびステータスレジスタのポートパワービット (OTG\_HPRT の PPWR ビット) もセットする必要があります。

### 有効な $V_{BUS}$

HNP または SRP が有効化された場合、VBUS センシングピンは  $V_{BUS}$  に接続する必要があります。  $V_{BUS}$  入力は、USB 動作中にチャージポンプによって適切な  $V_{BUS}$  レベルが供給されることを保証します。予期せぬ  $V_{BUS}$  電圧の低下 ( $V_{BUS}$  の有効閾値 4.25 V 以下へ) が発生すると、セッション終了検出ビット (OTG\_GOTGINT の SEDET ビット) によって OTG 割り込みがトリガされます。その場合、アプリケーションは  $V_{BUS}$  電源供給を停止し、ポートパワービットをクリアする必要があります。

HNP と SRP が共に無効化された場合、VBUS センシングピンを  $V_{BUS}$  に接続する必要はなく、GPIO として使用できます。

チャージポンプの過電流フラグは、電氣的ダメージを防止するためにも使用できます。チャージポンプからの過電流フラグ出力を任意の GPIO 入力に接続して、アクティブレベルになったときにポート割り込みを生成するように設定します。過電流 ISR は迅速に  $V_{BUS}$  の発生を無効にし、ポートパワービットをクリアする必要があります。

## ペリフェラル接続のホスト検出

SRP または HNP を有効化し、かつ USB ペリフェラルまたは B デバイスをいつでも接続できる場合でも、

OTG\_FS/OTG\_HS はバス接続を検出しません ( $V_{BUS}$  が有効レベル (5 V) で検出できなくなるまで)。  $V_{BUS}$  が有効なレベルにあり、リモート B デバイスが接続されると、OTG\_FS/OTG\_HS コアは、ホストポート制御およびステータスレジスタのデバイス接続ビット (OTG\_HPRT の PCDET ビット) によってトリガされるホストポート割り込みを発行します。

HNP と SRP が共に切り離されたとき、USB ペリフェラルまたは B デバイスが接続されると直ちに検出されます。OTG\_FS/OTG\_HS コアは、ホストポート制御およびステータスレジスタのデバイス接続ビット (OTG\_HPRT の PCDET ビット) によってトリガされるホストポート割り込みを発行します。

## ペリフェラル切断のホスト検出

ペリフェラル切断イベントは、切断検出割り込み (OTG\_GINTSTS の DISCINT ビット) をトリガします。

## ホストのエニユメレーション

ペリフェラル接続を検出した後、ホストは、USB リセットおよび設定コマンドを新しいペリフェラルに送信することによって、エニユメレーションプロセスを開始しなければなりません。

USB リセットの駆動を開始する前に、アプリケーションは、デバウンス終了ビット (OTG\_GOTGINT の DBCDNE ビット) によってトリガされる OTG 割り込みを待ちます。これは、DP (FS) または DM (LS) 上のプルアップ抵抗の接続によって生じた電氣的デバウンスの後、バスが再び安定状態に戻ったことを示します。

アプリケーションは、ホストポート制御およびステータスレジスタのポートリセットビット (OTG\_HPRT の PRST ビット) を最短 10 ms、最長 20 ms の間セット状態に保つことによって、USB を介した USB リセットシグナリング (シングルエンド、ゼロ) を駆動します。アプリケーションは、タイミングカウントをチェックした後、ポートリセットビットがクリアされているかどうかを確認します。

USB リセットシーケンスが完了すると、ポートイネーブル/ディセーブル変更ビット (OTG\_HPRT の PENCHNG ビット) によってホストポート割り込みがトリガされます。これにより、エニユメレーションされたペリフェラルの速度をホストポート制御およびステータスレジスタのポート速度フィールド (OTG\_HPRT の PSPD ビット) から読み出せることと、ホストが SOF (FS) またはキープアライブ (LS) の駆動を開始しようとしていることがアプリケーションに通知されます。これで、ホストは、ペリフェラル設定コマンドを送信することによってペリフェラルエニユメレーションを完了する準備ができたこととなります。

## ホストのサスペンド

アプリケーションは、ホストポート制御およびステータスレジスタのポートサスペンドビット (OTG\_HPRT の PSUSP ビット) をセットすることによって、USB の活動をサスペンド (一時停止) します。OTG\_FS/OTG\_HS コアは、SOF の送信を停止し、サスペンド状態に入ります。

サスペンド状態は、リモートデバイスが主導して (リモートウェイクアップ)、終了することもできます。この場合、ホストポート制御およびステータスレジスタのポートレジュームビット (OTG\_HPRT の PRES ビット) セルフセットが検出されると、リモートウェイクアップ割り込み (OTG\_GINTSTS の WKUPINT ビット) が生成され、USB によってレジュームシグナリングが自動的に駆動されます。アプリケーションは、レジュームウィンドウの時間を決めてから、ポートレジュームビットをクリアしてサスペンド状態を終了し、SOF をリスタートする必要があります。

サスペンド状態をホスト主導で終了した場合、アプリケーションは、ポートレジュームビットをセットして、ホストポートでのシグナリングのレジュームを開始し、レジュームウィンドウの時間を決め、最後に、ポートレジュームビットをクリアする必要があります。

## 37.7.3 ホストチャネル

OTG\_FS/OTG\_HS コアは 12[FS]/ 16[HS] ホストチャネルをインスタンス化します。各ホストチャネルは USB ホスト転送をサポートします (USB パイプ)。ホストは、同時に 12[FS]/ 16[HS] を超える転送リクエストをサポートすることはできません。アプリケーションからの 12[FS]/ 16[HS] を超える転送リクエストがペンディング中の場合、ホストコントローラドライバ (HCD) は、チャネルが以前のデューティから使用可能になったとき、すなわち、転送完了およびチャネル停止割り込みの受信後、チャネルの再割り当てを行う必要があります。

各ホストチャネルは、入力/出力および任意のタイプの周期的/非周期的トランザクションをサポートするように設定できます。各ホストチャネルは、適切な制御レジスタ (OTG\_HCCHARx)、転送設定レジスタ (OTG\_HCTSIZx)、およびステータス/割り込みレジスタ (OTG\_HCINTx) と関連するマスクレジスタ (OTG\_HCINTMSKx) を使用します。

### ホストチャネル制御

- アプリケーションは、ホストチャネル x 特性レジスタ (OTG\_HCCHARx) を通じて、以下のホストチャネル制御を使用できます。
  - チャネルの有効化/無効化
  - ターゲット USB ペリフェラルの HS/FS/LS 速度のプログラミング
  - ターゲット USB ペリフェラルのアドレスのプログラミング
  - ターゲット USB ペリフェラルのエンドポイント数のプログラミング
  - IN/OUT 転送方向のプログラミング
  - USB 転送タイプ (制御、バルク、割り込み、アイソクロナス) のプログラミング
  - 最大パケットサイズ (MPS) のプログラミング
  - 奇数/偶数フレームで実行される周期的転送のプログラミング

### ホストチャネル転送

アプリケーションは、ホストチャネル転送サイズレジスタ (OTG\_HCTSIZx) を使用して、転送サイズパラメータをプログラムし、転送ステータスを読み出すことができます。プログラミングは、ホストチャネル特性レジスタのチャネル有効化ビットをセットする前に行う必要があります。エンドポイントが有効になると OTG\_FS/OTG\_HS コアが現在の転送ステータスに従って更新するので、パケットカウントフィールドは読み出し専用になります。

- 以下の転送パラメータがプログラム可能です。
  - 転送サイズ (バイト単位)
  - 全体の転送サイズを構成するパケットの数
  - 初期データ PID

## ホストチャネルのステータス/割り込み

ホストチャネル x 割り込みレジスタ (OTG\_HCINTx) は、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。アプリケーションは、コア割り込みレジスタのホストチャネル割り込みビット (OTG\_GINTSTS の HCINT ビット) がセットされたとき、これらのレジスタを読み出さなければなりません。アプリケーションがこれらのレジスタを読み出すためには、まず、ホストオールチャネル割り込み (OTG\_HAINT) レジスタを読み出して、ホストチャネル x 割り込みレジスタの正確なチャネル数を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG\_HAINT および OTG\_GINTSTS レジスタの対応するビットをクリアする必要があります。各チャネルの各割り込みソースのマスクビットも OTG\_HCINTMSKx レジスタで使用可能です。

- ホストコアは、以下のステータスチェックおよび割り込み生成を行います。
  - データ転送がアプリケーション側 (AHB) と USB 側の両方で完了したことを示す転送完了割り込み
  - 転送完了、USB トランザクションエラー、またはアプリケーションからの無効化コマンドのためにチャネルが停止したかどうか
  - 関連する送信 FIFO が半分エンプティか完全エンプティか (IN エンドポイント)
  - ACK 応答を受け取った時
  - NAK 応答を受け取った時
  - STALL 応答を受け取った時
  - CRC エラー、タイムアウト、ビットスタッフエラー、偽 EOP による USB トランザクションエラー
  - バブルエラー
  - フレームオーバーラン
  - データトグルエラー

## 37.7.4 ホストスケジューラ

ホストコアは、アプリケーションによってポストされた USB トランザクションリクエストを自律的に並べ替えて、管理する組み込みハードウェアスケジューラを備えています。各フレームの開始時に、ホストは、周期的 (アイソクロナスおよび割り込み) トランザクションを最初に実行し、次に、非周期的 (制御およびバルク) トランザクションを行うことによって USB の仕様に従って、アイソクロナスおよび割り込み転送タイプを優先します。

ホストは、リクエストキュー (1 つは周期的転送のキュー、もう 1 つは非周期的転送のキュー) を通じて USB トランザクションを処理します。各リクエストキューは、最大 8 個のエントリを保持できます。各エントリはアプリケーションからのペンディング中のトランザクションリクエストを表し、IN または OUT チャネル番号と USB 上でトランザクションを実行するためのその他の情報を保持しています。リクエストがキューに書き込まれた順序によって USB インタフェース上のトランザクションの順序が決まります。

各フレームの先頭で、ホストは、まず、周期的リクエストキューを処理し、次に、非周期的リクエストキューを処理します。現在のフレームに対してスケジュールされたアイソクロナスまたは割り込みトランザクションが現在のフレームの終了時点でまだペンディング中である場合、ホストは、不完全周期転送割り込み (OTG\_GINTSTS の IPXFR ビット) を発行します。OTG\_FS/OTG\_HS コアは周期的と非周期的リクエストキューの管理を十分に行います。周期的送信 FIFO およびキューステータスレジスタ (OTG\_HPTXSTS) と非周期的送信 FIFO およびキューステータスレジスタ (OTG\_HNPTXSTS) は読み出し専用レジスタであり、アプリケーションから各リクエストキューのステータスを読み出すときに使うことができます。これらのレジスタには次が含まれます。

- 周期 (非周期) リクエストキューで現在使用可能なフリーエントリの数 (最大 8 エントリ)。



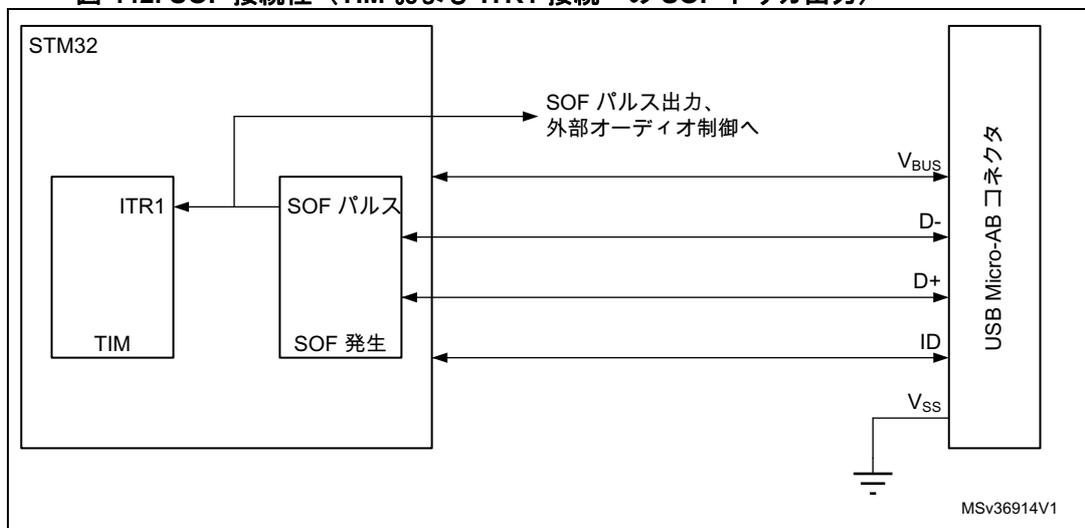
- 周期 (非周期) Tx FIFO (OUT トランザクション) で現在使用可能な空きスペース
- IN/OUT トークン、ホストチャンネル番号、およびその他のステータス情報

リクエストキューはそれぞれ最大 8 つのエントリを保持できるので、アプリケーションは、最大 8 つのペンディング中の周期トランザクションに加えて、8 つのペンディング中の非周期トランザクションについて、SB に物理的に到達する前に、ホストトランザクションを事前にスケジュールできます。

ホストスケジューラ (キュー) ヘトランザクションリクエストをポストするときは、アプリケーションは OTG\_HNPTXSTS レジスタの PTXQSAV ビット、または OTG\_HNPTXSTS レジスタの NPTQSAV ビットを読み出して、周期的 (非周期的) リクエストキュー内に少なくとも 1 つのエントリが存在することを確認する必要があります。

## 37.8 SOF トリガ

図 442. SOF 接続性 (TIM および ITR1 接続への SOF トリガ出力)



OTG\_FS/OTG\_HS コアは、ホスト、ペリフェラル、SOF パルス出力接続機能内で SOF フレーミングを監視、追跡、設定する手段を提供します。

これらのユーティリティは、オーディオペリフェラルが PC によって提供されるアイソクロナスストリームに同期する必要がある場合や、ホストがオーディオペリフェラルの要件に従ってフレーミング速度をトリム (微調整) する必要がある場合など、適応型オーディオクロック生成技術にとって特に役立ちます。

### 37.8.1 ホストの SOF

ホストモードでは、2 つの連続した SOF (HS/FS) またはキープアライブ (LS) トークンの生成の間に発生する PHY クロックの数は、ホストフレームインターバルレジスタ (HFIR) でプログラム可能であり、それにより、SOF フレーミング周期中にアプリケーションによる制御が可能です。割り込みは、どのフレームの開始時点でも生成されます (OTG\_GINTSTS の SOF ビット)。現在のフレーム番号と次の SOF までの残り時間が、ホストフレーム番号レジスタ (HFNUM) で追跡されます。

SOF パルス信号は、任意の SOF 開始トークンで生成され、幅は 12 システムクロックサイクルであり、グローバル制御および設定レジスタの SOFOUTEN ビットを使用して、SOF ピン上で外部的に使用可能になります。SOF パルスも、タイマの入力トリガに内部接続されます。

## 37.8.2 ペリフェラルの SOF

デバイスモードでは、フレーム開始の割り込みは、SOF トークンが USB 上で受信されるたびに生成されます (OTG\_GINTSTS の SOF ビット)。対応するフレーム番号は、デバイスステータスレジスタ (OTG\_DSTS の FNSOF ビット) から読み出すことができます。12 システムクロックサイクルの幅を持つ SOF パルス信号も生成され、グローバル制御および設定レジスタの SOF 出力イネーブルビット (OTG\_GCCFG の SOFOUTEN ビット) を使用して SOF ピン上で外部的に使用可能になります。SOF パルス信号は、タイマ 2 (TIM2) 入力トリガにも内部接続されるので、入力キャプチャ機能、出力比較機能、およびタイマを SOF パルスによってトリガできます。TIM2 接続は、TIM2 オプションレジスタ (TIM2\_OR) の ITR1\_RMP ビットを使って有効化します。

周期的フレーム終わりの割り込み (OTG\_GINTSTS の EOPF) は、デバイス設定レジスタの周期的フレームインターバルフィールド (OTG\_DCFG の PFIVL ビット) に応じて、タイムフレームインターバルの 80 %、85 %、90 % または 95 % が経過した時点を実アプリケーションに通知するために使用されます。この機能を使用して、そのフレームのアイソクロナストラフィックのすべてが完了したかどうかを判断できます。

## 37.9 電力オプション

OTG PHY の消費電力は、一般コア設定レジスタの 2 つのビットによって制御されます。

- PHY パワーダウン (OTG\_GCCFG の PWRDWN)  
PHY のフルスピードトランシーバモジュールのオン/オフを切り替えます。事前にセットして、すべての USB 動作を可能にする必要があります。
- V<sub>BUS</sub> 検出イネーブル (OTG\_GCCFG の VB DEN)  
OTG の動作に関連する V<sub>BUS</sub> レベル検出用コンパレータのオン/オフを切り替えます。

USB セッションがまだ有効でないか、デバイスが切断されているときには、USB サスペンド状態で節電テクニックを使用できます。

- PHY クロックを停止します (OTG\_PCGCCTL の STPPCLK ビット)。  
クロックゲーティング制御レジスタのストップ PHY クロックビットをセットすると、OTG 全速コア内部の 48 MHz クロックドメインのほとんどがクロックゲーティングによってスイッチオフされます。USB クロック切り替え機能による動的な電力消費は、48 MHz クロック入力アプリケーションによって動作状態を維持された場合でも節減されます。  
ほとんどのトランシーバも無効になり、非同期レジュームやリモートウェイクアップイベントの検出を担当する部分だけが動作状態を維持します。
- ゲート HCLK (OTG\_PCGCCTL の GATEHCLK ビット)  
クロックゲーティング制御レジスタのゲート HCLK ビットをセットすると、OTG\_FS/OTG\_HS コア内部のシステムクロックドメインのほとんどがクロックゲーティングによってスイッチオフされます。レジスタ読み出しおよび書き込みインタフェースのみが動作状態を維持します。USB クロック切り替え機能による動的な電力消費は、システムクロックが他の目的のためにアプリケーションによって動作状態を維持されている場合でも節減されます。
- USB システムの停止  
OTG\_FS/OTG\_HS が USB サスペンド状態にある場合、アプリケーションはシステム内のすべてのクロックソースを完全にシャットダウンすることによって、電力消費全体を大幅に削減する方法を取ることができます。USB システムの停止は、まず、ストップ PHY クロックビットをセットし、次に、パワー制御システムモジュール (PWR) でシステムディープスリープモードを設定することによって有効になります。  
OTG\_FS/OTG\_HS コアは、USB 上でリモートウェイクアップ (ホストとして) またはリモートレジューム (デバイスとして) のシグナリングの非同期検出によって、システムと USB クロックの両方を自動的に再起動します。

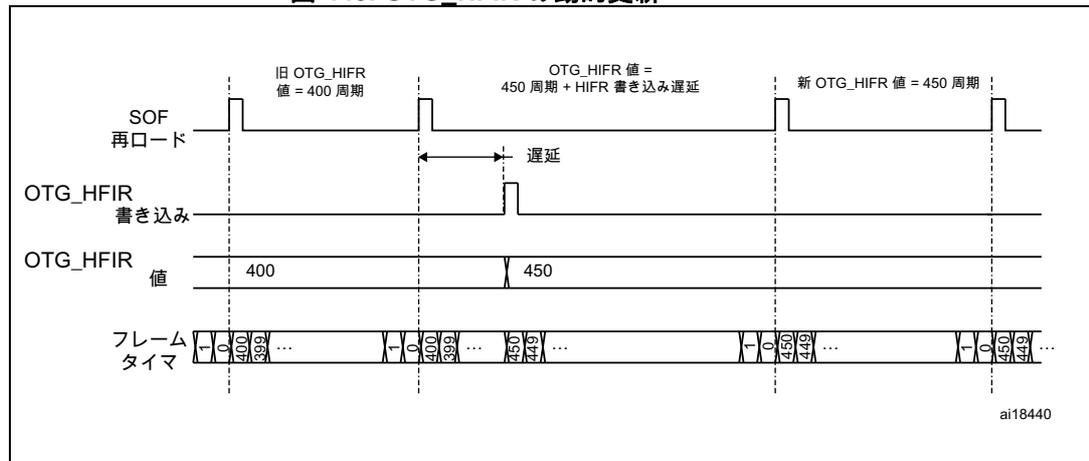
動的消費電力を節減するために、USB データ FIFO は、OTG\_FS/OTG\_HS コアによってアクセスされたときだけクロック駆動されます。

## 37.10 OTG\_HFIR レジスタの動的更新

USB コアは、ホストモードで micro-SOF[HS] / SOF[FS] フレーミング周期を動的にトリミングする機能を内蔵しています。この機能を使うと、外部デバイスを マイクロ SOF[HS] / SOF[FS] フレームと同期させることができます。

現在の マイクロ SOF[HS] / SOF[FS] フレーム内で OTG\_HFIR レジスタを変更すると、[図 443](#)で説明するように次のフレーム内で SOF 周期補正が適用されます。

図 443. OTG\_HFIR の動的更新

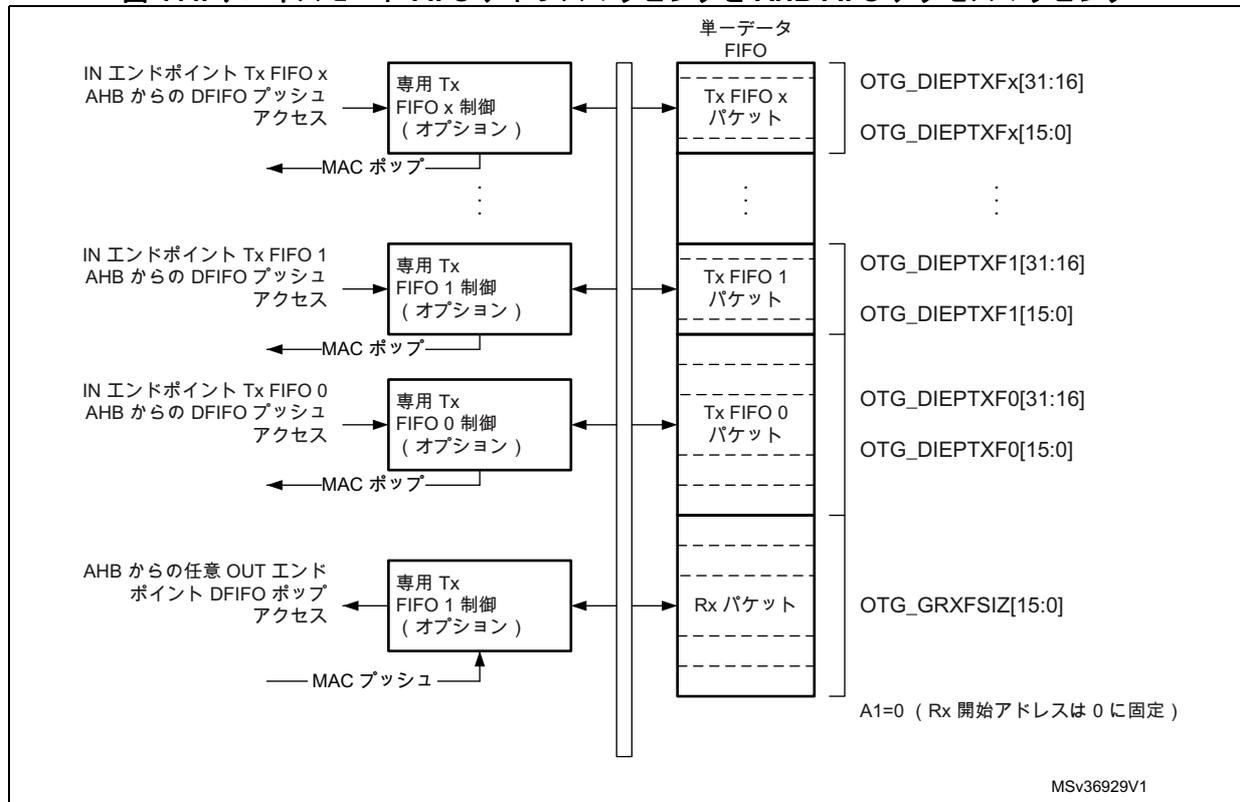


## 37.11 USB データ FIFO

USB システムは、高度な FIFO 制御メカニズムを持つ 1.25[FS] / 4[HS] KB の専用 RAM を備えています。OTG\_FS/OTG\_HS コアの パケット FIFO コントローラモジュールは、RAM 空間を、アプリケーションがデータを USB 転送の前に一時的に格納する（プッシュする）複数の Tx FIFO と、USB から受信したデータをアプリケーションが取得する（ポップする）前に一時的に格納する 1 つの Rx FIFO に編成します。割り当てられる FIFO の数と、これらが RAM の内部でどのように構成されるかは、デバイスの役割に依存します。ペリフェラルモードでは、追加の Tx FIFO がアクティブな IN エンドポイントのそれぞれに対して割り当てられます。FIFO のサイズは、アプリケーションの要件に合うように、ソフトウェアで設定できます。

## 37.11.1 ペリフェラル FIFO アーキテクチャ

図 444. デバイスモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング



### ペリフェラル Rx FIFO

OTG ペリフェラルは、すべての OUT エンドポイントに向けたデータを受信する単一の受信 FIFO を使用します。受信バケットは、Rx FIFO の空きスペースが使用可能になるまで、連続的にスタックされます。受信バケットのステータス (OUT エンドポイントの宛先番号、バイト数、データ PID、および受信データの有効性を含みます) も、コアによってデータペイロードの最上部に格納されます。使用可能なスペースがなくなると、ホストトランザクションは NACK され、アドレス指定されたエンドポイントに割り込みが受信されます。受信 FIFO のサイズは、受信 FIFO サイズレジスタ (OTG\_GRXFSIZ) で設定されます。

単一受信 FIFO アーキテクチャは、USB ペリフェラルによる受信 RAM バッファへの書き込みをより効率的にします。

- すべての OUT エンドポイントは同じ RAM バッファを共有します (共有 FIFO)。
- OTG\_FS/OTG\_HS コアは、OUT トークンのいかなるホストシーケンスについても、限度まで受信 FIFO に書き込むことができます。

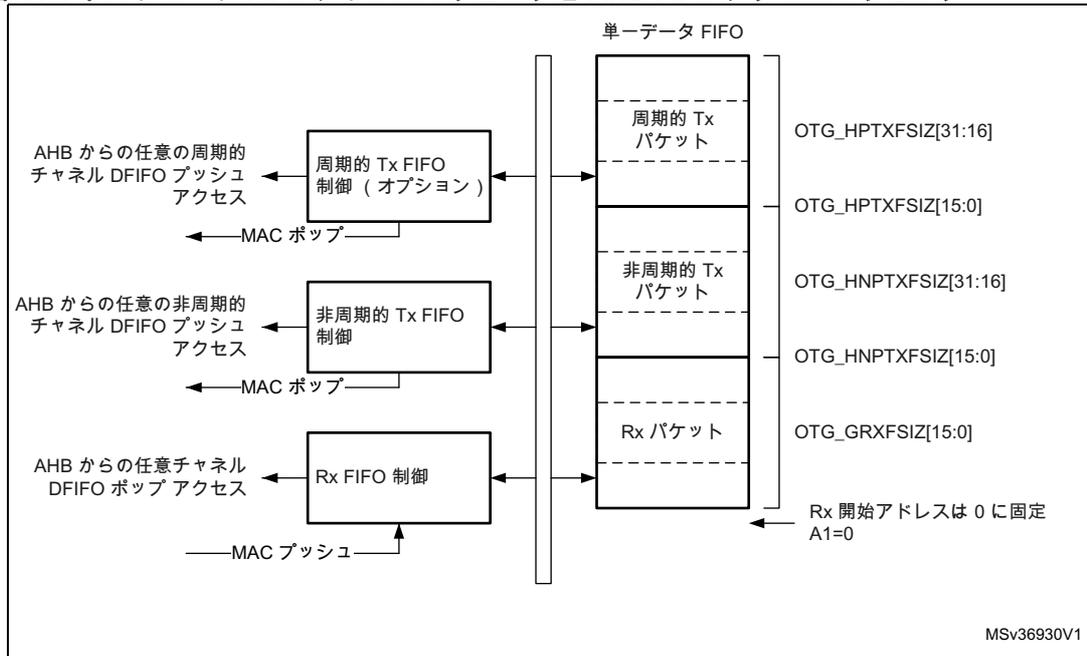
アプリケーションは Rx FIFO 非エンpty割り込み (OTG\_GINTSTS の RXFLVL ビット) の受信を、少なくとも 1 バケットがダウンロード可能である限り続けます。アプリケーションは、受信ステータス読み出しおよびポップレジスタ (OTG\_GRXSTSP) からバケット情報を読み出し、最後に、エンドポイントに関するポップアドレスから読み取ることによって受信 FIFO からデータを取得します。

## ペリフェラル Tx FIFO

コアは、各 IN エンドポイント用の専用 FIFO を備えています。アプリケーションは、IN エンドポイント 0 についてはエンドポイント 0 送信 FIFO サイズレジスタ (OTG\_DIEPTXF0) に書き込み、IN エンドポイント x についてはデバイス IN エンドポイント送信 FIFOx レジスタ (OTG\_DIEPTXFx) に書き込むことによって、FIFO のサイズを設定します。

## 37.11.2 ホスト FIFO アーキテクチャ

図 445. ホストモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング



## ホスト Rx FIFO

ホストは、すべての周期的および非周期的のトランザクションについて 1 つの受信 FIFO を使用します。この FIFO は、USB から受信したデータ (受信パケットのペイロード) をシステムメモリに転送されるまで保持する受信バッファとして使用されます。任意のリモート IN エンドポイントから受信されたパケットは、空きスペースがある限り連続的にスタックされます。各受信パケット (ホストチャンネルデスティネーション、バイト数、データ PID、および受信データの有効性を含みます) のステータスも、FIFO に格納されます。受信 FIFO のサイズは、受信 FIFO サイズレジスタ (OTG\_GRXFSIZ) で設定されます。

単一受信 FIFO アーキテクチャは、USB ホストによる受信データバッファへの書き込みをより効率的にします。

- IN に設定されたすべてのホストチャンネルは、同じ RAM バッファを共有します (共有 FIFO)。
- OTG\_FS/OTG\_HS コアは、ホストのソフトウェアによって駆動された IN トークンのいかなるシーケンスについても、限度まで受信 FIFO に書き込むことができます。

アプリケーションは、少なくとも 1 つのパケットがダウンロード可能である限り、Rx FIFO 非エンピティ割り込みを受け取ります。アプリケーションは、受信ステータス読み出しおよびポップレジスタからパケット情報を読み出し、最後に、受信 FIFO からデータを吸い上げます。

## ホスト Tx FIFO

ホストは、すべての非周期的（制御およびバルク）OUT トランザクションについて 1 つの送信 FIFO を、すべての周期的（アイソクロナスおよび割り込み）OUT トランザクションについて 1 つの送信 FIFO を使用します。FIFO は、USB 経由で送信されるデータ（送信パケットのペイロード）を保持する送信バッファとして使用されます。周期的（非周期的）Tx FIFO のサイズは、ホストの周期的（非周期的）送信 FIFO サイズレジスタ（OTG\_HPTXFSIZ/OTG\_HNPTXFSIZ）で設定されます。

Tx FIFO が 2 つ実装されているのは、USB フレームよりも周期的トラフィックタイプに対して高い優先度が与えられていることによります。各フレームの先頭で、組み込みのホストスケジューラは、まず、周期的リクエストキューを処理し、次に、非周期的リクエストキューを処理します。

2 つの送信 FIFO アーキテクチャにより、USB ホストは周期的送信データバッファと非周期的送信データバッファを個別に最適化できます。

- OUT 方向の周期的（非周期的）トランザクションをサポートするように設定されたすべてのホストチャネルは、同じ RAM バッファを共有します（共有 FIFO）。
- OTG\_FS/OTG\_HS コアは、ホストのソフトウェアによって駆動された OUT トークンのいかなるシーケンスについても、限度まで周期的（非周期的）送信 FIFO に書き込むことができます。

OTG\_FS/OTG\_HS コアは、AHB 設定レジスタの周期的 Tx FIFO エンプティレベルビット（OTG\_GAHBCFG の PTXFEVLV ビット）の値に応じて、周期的 Tx FIFO が半分エンプティか完全エンプティである限り、周期的 Tx FIFO エンプティ割り込み（OTG\_GINTSTS の PTXFE ビット）を発行します。アプリケーションは、周期的 Tx FIFO と周期的リクエストキューの両方に空きスペースが存在する限り、前もって送信データをプッシュすることができます。両方のレジスタにどれだけのスペースがあるかは、ホストの周期的送信 FIFO およびキューステータスレジスタ（OTG\_HPTXSTS）を読み出すことで知ることができます。

OTG\_FS/OTG\_HS コアは、AHB 設定レジスタの非周期的 Tx FIFO エンプティレベルビット（OTG\_GAHBCFG の TXFEVLV ビット）の値に応じて、非周期的 Tx FIFO が半分エンプティか完全エンプティである限り、非周期的 Tx FIFO エンプティ割り込み（OTG\_GINTSTS の NPTXFE ビット）を発行します。アプリケーションは、非周期的 Tx FIFO と非周期的リクエストキューの両方に空きスペースが存在する限り、送信データをプッシュすることができます。両方のレジスタにどれだけのスペースがあるかは、ホストの非周期的送信 FIFO およびキューステータスレジスタ（OTG\_HNPTXSTS）を読み出すことで知ることができます。

## 37.11.3 FIFO RAM の割り当て

### デバイスモード

**受信 FIFO RAM 割り当て:** アプリケーションは SETUP パケット用に RAM を割り当てる必要があります。制御エンドポイントで SETUP パケットを受信するため受信 FIFO に 10 ロケーションを確保する必要があります。コアは、SETUP パケット用に予約されている位置に他のデータを書き込むことはありません。1 つの位置は、グローバル OUT NAK のために割り当てられます。各受信パケットとともに、ステータス情報が FIFO に書き込まれます。したがって、パケットを受信するには、少なくとも、(最大パケットサイズ / 4) + 1 のスペースを割り当てる必要があります。複数のアイソクロナスエンドポイントが有効な場合、連続パケットを受信するには、少なくとも、(最大パケットサイズ / 4) + 1 の 2 倍のスペースが割り当てられなければなりません。一般に、(最大パケットサイズ / 4) + 1 の 2 倍のスペースが推奨されます。前のパケットが CPU に転送されている間に、USB は次のパケットを受信できるからです。

各エンドポイントの最後のパケットとともに、送信完了ステータス情報が FIFO にプッシュされます。一般に、各 OUT エンドポイントに対して 1 つのロケーションを予約することが推奨されます。

**送信 FIFO RAM の割り当て:** 各 IN エンドポイント送信 FIFO に必要な最小 RAM スペースは、その特定の IN エンドポイントに対する最大パケットサイズを収容できるスペースとなります。

**注:** **送信 IN エンドポイント FIFO に割り当てるスペースが大きいほど、USB のパフォーマンスは高くなります。**



## ホストモード

受信 FIFO RAM 割り当て：

各受信パケットとともに、ステータス情報が FIFO に書き込まれます。したがって、パケットを受信するには、少なくとも、(最大パケットサイズ/4) + 1 のスペースを割り当てる必要があります。複数のアイソクロナスチャネルが有効な場合、連続パケットを受信するには、少なくとも、(最大パケットサイズ/4) + 1 の 2 倍のスペースが割り当てられなければなりません。一般に、(最大パケットサイズ/4) + 1 の 2 倍のスペースが推奨されます。前のパケットが CPU に転送されている間に、USB は次のパケットを受信できるからです。

ホストチャネルの最後のパケットとともに、転送完了ステータス情報が FIFO にプッシュされます。したがって、このために 1 つの位置が割り当てられなければなりません。

送信 FIFO RAM の割り当て：

ホストの非周期的送信 FIFO に必要な最小 RAM スペースは、サポートされるすべての非周期的 OUT チャネルの中で最大のパケットサイズを収容できるスペースとなります。

一般に、最大パケットサイズの 2 倍のスペースが推奨されます。現在のパケットが USB に転送されているときに、CPU が次のパケットを取得できるからです。

ホストの周期的送信 FIFO に必要な最小 RAM スペースは、サポートされるすべての周期的 OUT チャネルの中で最大のパケットサイズを収容できるスペースとなります。少なくとも 1 つのアイソクロナス OUT エンドポイントがある場合には、スペースはそのチャネルの最大パケットサイズの少なくとも 2 倍でなければなりません。

**注：** *非周期的送信 FIFO に割り当てるスペースが大きいほど、USB のパフォーマンスは高くなります。*

## 37.12 OTG\_FS システムのパフォーマンス

USB とシステムの最高のパフォーマンスを達成するには、大きな RAM バッファ、柔軟に設定可能な FIFO サイズ、AHB プッシュ/ポップレジスタを通した迅速な 32 ビット FIFO アクセス、そして、特に、高度な FIFO 制御メカニズムが必要です。このメカニズムにより、OTG\_FS は、現在の USB シーケンスに関係なく、使用可能な RAM スペースに効率的に書き込むことができます。これらの機能によって、以下のことが可能になります。

- アプリケーションは、CPU のバンド幅使用率を最適化するための介入を調整する十分な余裕ができます。
  - データを効率的に USB に送信できるときまで、大量の送信データを事前に蓄積することができます。
  - 単一の受信 FIFO からデータをダウンロードするための時間の余裕が増えるという利点があります。
- USB コアは、最高動作速度を維持できるので、アプリケーションの介入がある場合に比べて、自律的に動作できる幅が広がり、最大のフルスピードバンド幅を実現できます。
  - 大量の送信データを自由に蓄積して、USB 経由のデータ送信を自律的に管理できます。
  - 受信バッファの使用可能な空きスペースが大量にあるので、USB から入ってくるデータを自律的に書き込むことができます。

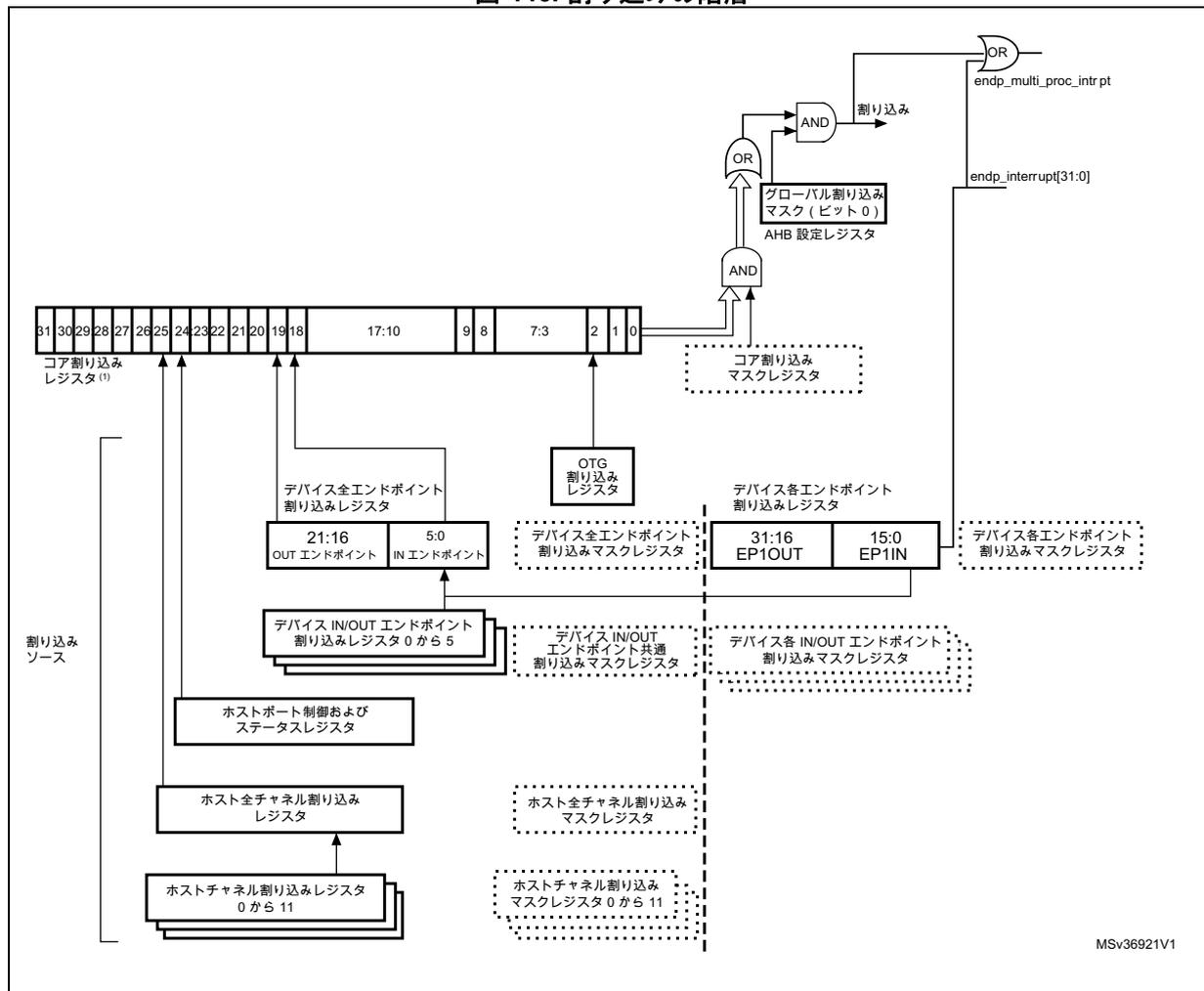
OTG\_FS コアは 1.25 KB の RAM バッファを非常に効率的に使用でき、また、1.25 KB の送信/受信データはフルスピードフレームをカバーするには十分以上なので、USB システムは CPU の介入がなくても、最大 1 USB フレーム (1 ms) に対応する最高のフルスピードデータ速度に耐えることができます。

## 37.13 OTG\_FS/OTG\_HS 割り込み

OTG\_FS/OTG\_HS コントローラがデバイスモードまたはホストモードのいずれかで動作しているときには、アプリケーションは、別のモードからレジスタにアクセスしてはなりません。不正なアクセスが発生した場合、モード不一致割り込みが生成され、コア割り込みレジスタ (OTG\_GINTSTS レジスタの MMIS ビット) に反映されます。コアがあるモードから別のモードに切り替わったとき、新しい動作モードにおけるレジスタは、パワーオンリセット後と同様に再プログラムされなければなりません。

図 446 に、割り込みの階層を示します。

図 446. 割り込みの階層



1. コア割り込みレジスタビットは、OTG コア割り込みレジスタ (OTG\_GINTSTS) (1337 ページ) に示されています。

## 37.14 OTG\_FS/OTG\_HS 制御およびステータスレジスタ

アプリケーションは AHB スレーブインタフェースを通じて制御およびステータスレジスタ (CSR) を読み書きすることによって、OTG\_FS/OTG\_HS コントローラを制御します。これらのレジスタは 32 ビット幅であり、アドレスは 32 ビットブロックに配置されています。OTG\_FS/OTG\_HS レジスタは、ワード (32 ビット) 単位でアクセスする必要があります。

CSR は、次のように分類されます。

- コアグローバルレジスタ
- ホストモードレジスタ
- ホストグローバルレジスタ
- ホストポート CSR
- ホストチャネル固有レジスタ
- デバイスモードレジスタ
- デバイスグローバルレジスタ
- デバイスエンドポイント固有レジスタ
- パワーおよびクロックゲーティングレジスタ
- データ FIFO (DFIFO) アクセスレジスタ

コアグローバル、パワーおよびクロックゲーティング、データ FIFO (DFIFO) アクセス、およびホストポート CSR レジスタのみが、ホストモードおよびデバイスモードの両方でのアクセスが可能です。OTG\_FS/OTG\_HS コントローラがデバイスモードまたはホストモードのいずれかで動作しているときには、アプリケーションは、別のモードからレジスタにアクセスしてはなりません。不正なアクセスが発生した場合、モード不一致割り込みが生成され、コア割り込みレジスタ (OTG\_GINTSTS レジスタの MMIS ビット) に反映されます。コアがあるモードから別のモードに切り替わったとき、新しい動作モードにおけるレジスタは、パワーオンリセット後と同様に再プログラムされなければなりません。

### 37.14.1 CSR メモリマップ

ホストモードおよびデバイスモードレジスタは、それぞれ異なるアドレスを占めます。すべてのレジスタは AHB クロックドメインで実装されます。

#### グローバル CSR マップ

これらのレジスタは、ホストモードとデバイスモードの両方で使用可能です。

表 228. コアグローバル制御およびステータスレジスタ (CSR)

項目 (略称)	アドレス オフセット	レジスタ名
OTG_GOTGCTL	0x000	OTG 制御およびステータスレジスタ (OTG_GOTGCTL) (1326 ページ)
OTG_GOTGINT	0x004	OTG 割り込みレジスタ (OTG_GOTGINT) (1329 ページ)
OTG_GAHBCFG	0x008	OTG AHB 設定レジスタ (OTG_GAHBCFG) (1330 ページ)
OTG_GUSBCFG	0x00C	OTG USB 設定レジスタ (OTG_GUSBCFG) (1332 ページ)
OTG_GRSTCTL	0x010	OTG リセットレジスタ (OTG_GRSTCTL) (1335 ページ)
OTG_GINTSTS	0x014	OTG コア割り込みレジスタ (OTG_GINTSTS) (1337 ページ)
OTG_GINTMSK	0x018	OTG 割り込みマスクレジスタ (OTG_GINTMSK) (1342 ページ)

表 228. コアグローバル制御およびステータスレジスタ (CSR) (続き)

項目 (略称)	アドレス オフセット	レジスタ名
OTG_GRXSTSR	0x01C	OTG_FS 受信ステータスデバッグ読み出し/OTGステータス読み出しおよびポップ レジスタ (OTG_GRXSTSR/OTG_GRXSTSP) (1345 ページ)
OTG_GRXSTSP	0x020	
OTG_GRXFSIZ	0x024	OTG 受信 FIFO サイズレジスタ (OTG_GRXFSIZ) (1348 ページ)
OTG_HNPTXFSIZ/OTG_DIEPTXF0 <sup>(1)</sup>	0x028	OTG ホスト非周期的送信 FIFO サイズレジスタ (OTG_HNPTXFSIZ) / エンドポ イント 0 送信 FIFO サイズ (OTG_DIEPTXF0)
OTG_HNPTXSTS	0x02C	OTG 非周期的送信 FIFO / キューステータスレジスタ (OTG_HNPTXSTS) (1349 ページ)
OTG_GI2CCTL	0x030	OTG I2C アクセスレジスタ (OTG_GI2CCTL) (1350 ページ)
OTG_GCCFG	0x038	OTG 一般コア設定レジスタ (OTG_GCCFG) (1351 ページ)
OTG_CID	0x03C	OTG コア ID レジスタ (OTG_CID) (1352 ページ)
OTG_HPTXFSIZ	0x100	OTG ホスト周期的送信 FIFO サイズレジスタ (OTG_HPTXFSIZ) (1356 ページ)
OTG_DIEPTXFx	0x104	OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_DIEPTXFx)(x = 1..5[FS] / 7[HS], ここで x は FIFO_number) (1357 ページ) (USB_OTG FS の場合)
	0x124	
	...	
	0x184	
OTG_DIEPTXFx	0x104	OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_DIEPTXFx)(x = 1..5[FS] / 7[HS], ここで x は FIFO_number) (1357 ページ) (USB_OTG HS の場合)
	0x124	
	...	
	0x1B4	

1. 一般的ルールとして、OTG\_HNPTXFSIZ はホストモードで、OTG\_DIEPTXF0 はデバイスモードで使用されます。

## ホストモード CSR マップ

これらのレジスタは、コアがホストモードに変わるたびにプログラムされなければなりません。

表 229. ホストモード制御およびステータスレジスタ (CSR)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_HCFG	0x400	OTG ホスト設定レジスタ (OTG_HCFG) (1358 ページ)
OTG_HFIR	0x404	OTG ホストフレームインターバルレジスタ (OTG_HFIR) (1359 ページ)
OTG_HFNUM	0x408	OTG ホストフレーム番号/残りフレーム時間 レジスタ (OTG_HFNUM) (1360 ページ)
OTG_HPTXSTS	0x410	OTG ホスト周期的送信 FIFO / キューステータスレジスタ (OTG_HPTXSTS) (1361 ページ)
OTG_HAINT	0x414	OTG ホスト全チャンネル割り込みレジスタ (OTG_HAINT) (1362 ページ)
OTG_HAINTMSK	0x418	OTG ホスト全チャンネル割り込みマスクレジスタ (OTG_HAINTMSK) (1362 ペ ージ)
OTG_HPRT	0x440	OTG ホストポート制御およびステータスレジスタ (OTG_HPRT) (1363 ページ)



表 229. ホストモード制御およびステータスレジスタ (CSR) (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_HCCHARx	0x500 0x520 ... 0x660	OTG ホストチャネル x 特性レジスタ (OTG_HCCHARx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1365 ページ) (USB_OTG FS の場合)
OTG_HCCHARx	0x500 0x520 ... 0x6E0	OTG ホストチャネル x 特性レジスタ (OTG_HCCHARx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1365 ページ) (USB_OTG HS の場合)
OTG_HCSPLTx	0x504 0x524 ... 0x6E4	OTG ホストチャネル x スプリット制御レジスタ (OTG_HCSPLTx) (x = 0..15, こ こで x = Channel_number) (1367 ページ)
OTG_HCDMAX	0x514 0x534 ... 0x6F4	OTG ホストチャネル x DMA アドレスレジスタ (OTG_HCDMAX) (x = 0..15, こ こで x = Channel_number) (1372 ページ)
OTG_HCINTx	0x508 0x528 ... 0x668	OTG ホストチャネル x 割り込みレジスタ (OTG_HCINTx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1368 ページ) (USB_OTG FS の場合)
OTG_HCINTx	0x508 0x528 ... 0x6E8	OTG ホストチャネル x 割り込みレジスタ (OTG_HCINTx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1368 ページ) (USB_OTG HS の場合)
OTG_HCINTMSKx	0x50C 0x52C ... 0x66C	OTG ホストチャネル x 割り込みマスクレジスタ (OTG_HCINTMSKx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1369 ページ) (USB_OTG FS の場合)
OTG_HCINTMSKx	0x50C 0x52C ... 0x6EC	OTG ホストチャネル x 割り込みマスクレジスタ (OTG_HCINTMSKx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1369 ページ) (USB_OTG HS の場合)
OTG_HCTSIZx	0x510 0x530 ... 0x670	OTG ホストチャネル x 転送サイズレジスタ (OTG_HCTSIZx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1371 ページ) (USB_OTG FS の場合)
OTG_HCTSIZx	0x510 0x530 ... 0x6F0	OTG ホストチャネル x 転送サイズレジスタ (OTG_HCTSIZx) (x = 0..15[HS] / 11[FS], ここで x = Channel_number) (1371 ページ) (USB_OTG HS の場合)

## デバイスモード CSR マップ

これらのレジスタは、コアがデバイスモードに変わるたびにプログラムする必要があります。

表 230. デバイスモード制御およびステータスレジスタ

項目 (略称)	オフセット アドレス	レジスタ名
OTG_DCFG	0x800	OTG デバイス設定レジスタ (OTG_DCFG) (1372 ページ)
OTG_DCTL	0x804	OTG デバイス制御レジスタ (OTG_DCTL) (1374 ページ)
OTG_DSTS	0x808	OTG デバイスステータスレジスタ (OTG_DSTS) (1376 ページ)
OTG_DIEPMSK	0x810	OTG デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG_DIEPMSK) (1377 ページ)
OTG_DOEPMSK	0x814	OTG デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG_DOEPMSK) (1379 ページ)
OTG_DAIN	0x818	OTG デバイス全エンドポイント割り込みレジスタ (OTG_DAIN) (1380 ページ)
OTG_DAINMSK	0x81C	OTG 全エンドポイント割り込みマスクレジスタ (OTG_DAINMSK) (1381 ページ)
OTG_DVBUSDIS	0x828	OTG デバイス VBUS 放電時間レジスタ (OTG_DVBUSDIS) (1381 ページ)
OTG_DVBUSPULSE	0x82C	OTG デバイス VBUS パルシング時間レジスタ (OTG_DVBUSPULSE) (1382 ページ)
OTG_DTHRCTL	0x0830	OTG デバイス閾値制御レジスタ (OTG_DTHRCTL) (1382 ページ)
OTG_DIEPEMPMSK	0x834	OTG デバイス IN エンドポイント FIFO エンプティ割り込みマスクレジスタ (OTG_DIEPEMPMSK) (1384 ページ)
OTG_DEACHINT	0x838	OTG デバイス各エンドポイント割り込みレジスタ (OTG_DEACHINT) (1383 ページ)
OTG_DEACHINTMSK	0x83C	OTG デバイス各エンドポイント割り込みレジスタマスク (OTG_DEACHINTMSK) (1384 ページ)
OTG_DIEPCTL0	0x900	OTG デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_DIEPCTL0) (1385 ページ) (USB_OTG FS の場合)
OTG_DIEPCTLx	0x920 0x940 ... 0x9A0	OTG デバイスエンドポイント x 制御レジスタ (OTG_DIEPCTLx) (x = 1..5[FS]/0..7[HS], ここで x = Endpoint_number) (1386 ページ) (USB_OTG FS の場合)
OTG_DIEPCTLx	0x900 0x920 ... 0x9E0	OTG デバイスエンドポイント x 制御レジスタ (OTG_DIEPCTLx) (x = 1..5[FS]/0..7[HS], ここで x = Endpoint_number) (1386 ページ) (USB_OTG HS の場合)
OTG_DIEPINTx	0x908 0x928 ... 0x9A8	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DIEPINTx) (x = 0..5[FS]/7[HS], ここで x = Endpoint_number) (1392 ページ) (USB_OTG FS の場合)



表 230. デバイスモード制御およびステータスレジスタ (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_DIEPINTx	0x908 0x928 ... 0x9E8	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DIEPINTx) ( $x = 0..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1392 ページ) (USB_OTG HS の場合)
OTG_DIEPTSIZE0	0x910	OTG デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_DIEPTSIZE0) (1395 ページ)
OTG_DTXFSTSx	0x918 0x938 ... 0x9B8	OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_DTXFSTSx) ( $x = 0..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1398 ページ) (USB_OTG FS の場合)
OTG_DTXFSTSx	0x918 0x938 ... 0x9F8	OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_DTXFSTSx) ( $x = 0..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1398 ページ) (USB_OTG HS の場合)
OTG_DIEPTSIZEx	0x930 0x950 ... 0x9B0	OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG_DIEPTSIZEx) ( $x = 1..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1397 ページ) (USB_OTG FS の場合)
OTG_DIEPTSIZEx	0x930 0x950 ... 0x9F0	OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG_DIEPTSIZEx) ( $x = 1..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1397 ページ) (USB_OTG HS の場合)
OTG_DOEPCTL0	0xB00	OTG デバイス制御 OUT エンドポイント 0 制御レジスタ (OTG_DOEPCTL0) (1389 ページ)
OTG_DOEPCTLx	0xB20 0xB40 ... 0xBA0	OTG デバイスエンドポイント x 制御レジスタ (OTG_DOEPCTLx) ( $x = 1..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1390 ページ) (USB_OTG FS の場合)
OTG_DOEPCTLx	0xB20 0xB40 ... 0xBE0	OTG デバイスエンドポイント x 制御レジスタ (OTG_DOEPCTLx) ( $x = 1..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1390 ページ) (USB_OTG HS の場合)
OTG_DOEPINTx	0xB08 0xB28 ... 0xBA8	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DOEPINTx) ( $x = 0..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1394 ページ) (USB_OTG FS の場合)
OTG_DOEPINTx	0xB08 0xB28 ... 0xBE8	OTG デバイスエンドポイント x 割り込みレジスタ (OTG_DOEPINTx) ( $x = 0..5[FS]/7[HS]$ 、ここで $x = \text{Endpoint\_number}$ ) (1394 ページ) (USB_OTG HS の場合)
OTG_DOEPTSIZE0	0xB10	OTG デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG_DOEPTSIZE0) (1396 ページ)

表 230. デバイスモード制御およびステータスレジスタ (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_DOEPTSIZE <sub>x</sub>	0xB30 0xB50 ... 0xBB0	OTG デバイス OUT エンドポイント $x$ 転送サイズレジスタ (OTG_DOEPTSIZE <sub>x</sub> ) ( $x = 1..5[FS] / 7[HS]$ , ここで $x = \text{Endpoint\_number}$ ) (1398 ページ) (USB_OTG FS の場合)
OTG_DOEPTSIZE <sub>x</sub>	0xB30 0xB50 ... 0xBF0	OTG デバイス OUT エンドポイント $x$ 転送サイズレジスタ (OTG_DOEPTSIZE <sub>x</sub> ) ( $x = 1..5[FS] / 7[HS]$ , ここで $x = \text{Endpoint\_number}$ ) (1398 ページ) (USB_OTG HS の場合)

## データ FIFO (DFIFO) アクセスレジスタマップ

これらのレジスタは、ホストモードとデバイスモードの両方で使用可能であり、特定のエンドポイントまたはチャンネルの FIFO スペースを指定された方向で読み書きするために使用されます。ホストチャンネルが IN タイプの場合 FIFO はそのチャンネル上では読み出しのみが可能です。同様に、ホストチャンネルが OUT タイプの場合 FIFO はそのチャンネル上では書き込みのみが可能です。

表 231. データ FIFO (DFIFO) アクセスレジスタマップ

FIFO アクセスレジスタセクション	アドレス範囲	アクセス
デバイス IN エンドポイント 0/ホスト OUT チャンネル 0: DFIFO 書き込みアクセス デバイス OUT エンドポイント 0/ホスト IN チャンネル 0: DFIFO読み出しアクセス	0x1000 – 0x1FFC	W r
デバイス IN エンドポイント 1/ホスト OUT チャンネル 1: DFIFO 書き込みアクセス デバイス OUT エンドポイント 1/ホスト IN チャンネル 1: DFIFO読み出しアクセス	0x2000 – 0x2FFC	W r
...	...	...
デバイス IN エンドポイント $x^{(1)}$ /ホスト OUT チャンネル $x^{(1)}$ : DFIFO 書き込みアクセス デバイス OUT エンドポイント $x^{(1)}$ /ホスト IN チャンネル $x^{(1)}$ : DFIFO読み出しアクセス	0xX000 – 0xXFFC	W r

1. ここで  $x$  は、 $5[FS] / 7[HS]$  (デバイスモードの場合)、 $11[FS] / 15[HS]$  (ホストモードの場合) です。

## パワーおよびクロックゲーティング CSR マップ

パワーおよびクロックゲーティングの場合、シングルレジスタです。このレジスタは、ホストモードとデバイスモードの両方で使用できます。

表 232. パワーおよびクロックゲーティング制御およびステータスレジスタ

レジスタ名	項目 (略称)	オフセットアドレス: 0xE00 – 0xFFFF
パワーおよびクロックゲーティング制御レジスタ	PCGCR	0xE00 - 0xE04
予約済み	-	0xE05 – 0xFFFF

## 37.15 OTG\_FS/OTG\_HS レジスタ

これらのレジスタは、ホストモードとデバイスモードの両方で使用でき、これらのモード間で切り替わっても再プログラムの必要はありません。

レジスタの説明で示されているビット値は、特に記載がない限りバイナリで表されています。

### 37.15.1 OTG 制御およびステータスレジスタ (OTG\_GOTGCTL)

アドレスオフセット : 0x000

リセット値 : 0x0001 0000

OTG\_GOTGCTL レジスタは、コアの OTG 機能の動作を制御し、そのステータスを反映します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTG VER	BSVLD	ASVLD	DBCT	CID STS
											rW	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EHEN	DHNP EN	HSHNP EN	HNP RQ	HNG SCS	Res.	Res.	Res.	Res.	Res.	Res.	SRQ	SRQ SCS
			rW	rW	rW	rW	r							rW	r

**注 :** **USB OTG FS の設定レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSVLD	ASVLD	DBCT	CID STS
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EHEN	DHNP EN	HSHNP EN	HNP RQ	HNG SCS	BVALO VAL	BVALO EN	AVALO VAL	AVALO EN	VBVAL OVAL	VBVAL OEN	SRQ	SRQ SCS
			rW	rW	rW	rW	r	rW	rW	rW	rW	rW	rW	rW	r

**注 :** **USB OTG HS の設定レジスタ**

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 20 **OTGVER** : OTG バージョン

OTG リビジョンを選択します。

0 : OTG バージョン 1.3 です。このバージョンでは、コアは SRP のデータラインパルシングおよび VBUS パルシングをサポートしています。

1 : OTG バージョン 2.0 です。このバージョンでは、コアは SRP のデータラインパルシングのみをサポートしています。

ビット 19 **BSVLD** : B セッション有効

デバイスモードでのトランシーバのステータスを示します。

0 : Bセッションは無効。

1 : Bセッションは有効。

OTG モードでは、このビットを使用して、デバイスが接続されているか、切断されているかを判断できます。

**注 :** **デバイスモードでのみアクセス可能です。**

**ビット 18 ASVLD** : A セッション有効

ホストモードでのトランシーバのステータスを示します。

0 : Aセッションは無効。

1 : Aセッションは有効。

**注:** *ホストモードでのみアクセス可能です。*

**ビット 17 DBCT** : ロング/ショートデバウンス時間

検出された接続のデバウンス時間を示します。

0 : ロングデバウンス時間 (100 ms + 2.5  $\mu$ s)。物理的接続に使用されます。

1 : ショートデバウンス時間 (2.5  $\mu$ s)。ソフト接続に使用されます。

**注:** *ホストモードでのみアクセス可能です。*

**ビット 16 CIDSTS** : コネクタ ID ステータス

接続イベント時のコネクタ ID のステータスを示します。

0 : OTG\_FS/OTG\_HS コントローラは A デバイスモードです。

1 : OTG\_FS/OTG\_HS コントローラは B デバイスモードです。

**注:** *デバイスモードとホストモードの両方でアクセス可能です。*

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

**ビット 12 EHEN** : 内蔵ホスト有効化

OTG A デバイスのステートマシンと内蔵ホストのステートマシンから選択するために使用します。

0 : OTG A デバイスのステートマシンを選択します。

1 : 内蔵ホストのステートマシンを選択します。

**ビット 11 DHNPEN** : デバイス HNP 有効化

アプリケーションは、接続された USB ホストから SetFeature.SetHNPEnable コマンドを正常に受信したときに、このビットをセットします。

0 : HNP は、アプリケーションでは無効です。

1 : HNP は、アプリケーションで有効です。

**注:** *デバイスモードでのみアクセス可能です。*

**ビット 10 HSHNPEN** : ホストセット HNP 有効化

アプリケーションは、接続されたデバイスで HNP を正常に有効できたとき (SetFeature.SetHNPEnable コマンドを使用して)、このビットをセットします。

0 : ホストセット HNP は無効です。

1 : ホストセット HNP は有効です。

**注:** *ホストモードでのみアクセス可能です。*

**ビット 9 HNPRQ** : HNP リクエスト

アプリケーションは、接続された USB ホストへの HNP リクエストを開始するために、このビットをセットします。アプリケーションは、OTG\_GOTGINT レジスタのホストネゴシエーション成功ステータス変更ビット (OTG\_GOTGINT の HNSSCHG ビット) がセットされたとき、0 を書き込むことによって、このビットをクリアできます。コアは HNSSCHG ビットがクリアされたとき、このビットをクリアします。

0 : HNP リクエストなし

1 : HNP リクエストあり

**注:** *デバイスモードでのみアクセス可能です。*

**ビット 8 HNGSCS** : ホストネゴシエーション成功

コアは、ホストネゴシエーションに成功したとき、このビットをセットします。コアは、このレジスタの HNP リクエストビット (HNPRQ) がセットされたとき、このビットをクリアします。

0 : ホストネゴシエーション失敗

1 : ホストネゴシエーション成功

**注:** *デバイスモードでのみアクセス可能です。*

**ビット 7 BVALOVAL** : B ペリフェラルセッション有効上書き値

このビットは、BVALOEN ビットがセットされている場合に、Bvalid 信号に上書き値をセットするために使用されます。

0 : BVALOEN = 1 のとき、Bvalid 値は“0”です。

1 : BVALOEN = 1 のとき、Bvalid 値は“1”です。

**注:** デバイスモードでのみアクセス可能です。

**ビット 6 BVALOEN** : B ペリフェラルセッション有効値の上書き有効化

このビットは、BVALOVAL ビットを使用して Bvalid 信号を上書きするソフトウェアを有効化/無効化するために使用されます。

0 : 上書きは無効です。選択された各 PHY からの Bvalid 信号は、コアが内部的に使用します。

1 : PHY から内部的に受け取った Bvalid 信号は BVALOVAL ビット値で上書きされます。

**注:** デバイスモードでのみアクセス可能です。

**ビット 5 AVALOVAL** : A ペリフェラルセッション有効上書き値

このビットは、AVALOEN ビットがセットされている場合に、Avalid 信号に上書き値をセットするために使用されます。

0 : AVALOEN = 1 のとき、Avalid 値は“0”です。

1 : AVALOEN = 1 のとき、Avalid 値は“1”です。

**注:** ホストモードでのみアクセス可能です。

**ビット 4 AVALOEN** : A ペリフェラルセッション有効値の上書き有効化

このビットは、AVALOVAL ビットを使用して Avalid 信号を上書きするソフトウェアを有効化/無効化するために使用されます。

0 : 上書きは無効です。選択された各 PHY からの Avalid 信号は、コアが内部的に使用します。

1 : PHY から内部的に受け取った Avalid 信号は AVALOVAL ビット値で上書きされます。

**注:** ホストモードでのみアクセス可能です。

**ビット 3 VBVALOVAL** : V<sub>BUS</sub> 有効上書き値

このビットは、VBVALOEN ビットがセットされている場合に、vbusvalid 信号に上書き値をセットするために使用されます。

0 : VBVALOEN = 1 のとき、vbusvalid 値は“0”です。

1 : VBVALOEN = 1 のとき、vbusvalid 値は“1”です。

**注:** ホストモードでのみアクセス可能です。

**ビット 2 VBVALOEN** : V<sub>BUS</sub> 有効値上書き有効化

このビットは、VBVALOVAL ビットを使用して vbusvalid 信号を上書きするソフトウェアを有効化/無効化するために使用されます。

0 : 上書きは無効です。選択された各 PHY からの vbusvalid 信号は、コアが内部的に使用します。

1 : PHY から内部的に受け取った vbusvalid 信号は VBVALOVAL ビット値で上書きされます。

**注:** ホストモードでのみアクセス可能です。

ビット 2:7 予約済みであり、リセット値に保持する必要があります。

**ビット 1 SRQ** : セッションリクエスト

アプリケーションは USB 上でセッションリクエストを開始するために、このビットをセットします。アプリケーションは、OTG\_GOTGINT レジスタのホストネゴシエーション成功ステータス変更ビット (OTG\_GOTGINT の HNSSCHG ビット) がセットされたとき、0 を書き込むことによって、このビットをクリアできます。コアは HNSSCHG ビットがクリアされたとき、このビットをクリアします。

USB 1.1 フルスピードシリアルトランシーバインタフェースを使用してセッションリクエストを開始する場合、アプリケーションは、このレジスタの B セッション有効ビット (OTG\_GOTGCTL の BSVLD ビット) がクリアされた後 V<sub>BUS</sub> が 0.2 V へ放電するまで待つ必要があります。この放電時間は PHY ごとに異なり PHY ベンダーから得ることができます。

0 : セッションリクエストなし  
1 : セッションリクエスト

**注** : デバイスモードでのみアクセス可能です。

**ビット 0 SRQSCS** : セッションリクエスト成功

コアは、セッションリクエストの開始に成功したとき、このビットをセットします。

0 : セッションリクエスト失敗  
1 : セッションリクエスト成功

**注** : デバイスモードでのみアクセス可能です。

## 37.15.2 OTG 割り込みレジスタ (OTG\_GOTGINT)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アプリケーションは OTG 割り込みがあると、このレジスタを読み出し、このレジスタのビットをクリアすることによって OTG 割り込みをクリアします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	ID CHNG	DBC DNE	ADTO CHG	HNG DET	Res.						
											rc_w1	rc_w1	rc_w1	rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HNSS CHG	SRSS CHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
						rc_w1	rc_w1						rc_w1		

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

**ビット 20 IDCHNG** :

このビットがセットされている場合、ID 入力ピンの値に変更があることを示します。

**ビット 19 DBCDNE** : デバウンス終了

コアは、デバイスが接続された後、デバウンスが完了したときに、このビットをセットします。アプリケーションは、この割り込みを確認した後に USB リセットの駆動を開始できます。このビットは、OTG\_GUSBCFG レジスタの HNP 対応ビットまたは SRP 対応ビット (OTG\_GUSBCFG の HNPCAP ビットと SRPCAP ビット) がセットされたときのみに有効です。

**注** : ホストモードでのみアクセス可能です。

**ビット 18 ADTOCHG** : A デバイスタイムアウト変更

コアは B デバイスの接続を待っている間に A デバイスがタイムアウトしたことを示すために、このビットをセットします。

**注** : デバイスモードとホストモードの両方でアクセス可能です。



ビット 17 HNGDET : ホストネゴシエーションの検出

コアは USB 上でホストネゴシエーションリクエストを検出したときに、このビットをセットします。

**注:** デバイスマードとホストモードの両方でアクセス可能です。

ビット 16:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 HNSSCHG : ホストネゴシエーション成功ステータス変化

コアは USB ホストネゴシエーションリクエストの成功または失敗時に、このビットをセットします。アプリケーションは、OTG\_GOTGCTL レジスタのホストネゴシエーション成功ビット (OTG\_GOTGCTL の HNGSCS ビット) を読み出して、成功か失敗かをチェックする必要があります。

**注:** デバイスマードとホストモードの両方でアクセス可能です。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 8 SRSSCHG : セッションリクエスト成功ステータス変化

コアは、セッションリクエストの成功または失敗のときに、このビットをセットします。アプリケーションは、OTG\_GOTGCTL レジスタのセッションリクエスト成功ビット (OTG\_GOTGCTL の SRQSCS ビット) を読み出して、成功か失敗かをチェックする必要があります。

**注:** デバイスマードとホストモードの両方でアクセス可能です。

ビット 2 SEDET : セッション終了検出

コアは、このビットをセットして、V<sub>BUS</sub> の電圧レベルが B ペリフェラルセッションにとって有効でなくなったことを示します (V<sub>BUS</sub> < 0.8 V のとき)。

**注:** デバイスマードとホストモードの両方でアクセス可能です。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

### 37.15.3 OTG AHB 設定レジスタ (OTG\_GAHBCFG)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

このレジスタを使用して、電源投入後、またはモード変化後のコアを設定できます。このレジスタは、主に AHB システム関連の設定パラメータを含んでいます。初期プログラミングの後には、このレジスタを変更しないでください。アプリケーションは AHB または USB のいずれかでトランザクションを開始する前に、このレジスタをプログラムする必要があります。

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

**注:** USB OTG FS の設定レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PTXFE LVL	TXFE LVL	Res.	DMAEN	HBSTLEN				GINT MSK						
							rw	rw		rw	rw	rw	rw	rw	rw

**注: USB OTG HS の設定レジスタ**

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PTXFELVL**: 周期的 Tx FIFO エンプティレベル

OTG\_GINTSTS レジスタの周期的 Tx FIFO エンプティ割り込みビット (OTG\_GINTSTS の PTXFE ビット) がトリガされるタイミングを示します。

0: OTG\_GINTSTS の PTXFE 割り込みビットは、周期的 Tx FIFO が半分エンプティであることを示します。

1: OTG\_GINTSTS の PTXFE 割り込みビットは、周期的 Tx FIFO が完全エンプティであることを示します。

**注: ホストモードでのみアクセス可能です。**

ビット 7 **TXFELVL**: Tx FIFO エンプティレベル

デバイスモードでは、このビットは、IN エンドポイント送信 FIFO エンプティ割り込み (OTG\_DIEPINTx の TXFE) がトリガされるタイミングを示します。

0: OTG\_DIEPINTx の TXFE 割り込みは、IN エンドポイント Tx FIFO が半分エンプティであることを示します。

1: OTG\_DIEPINTx の TXFE 割り込みは、IN エンドポイント Tx FIFO が完全エンプティであることを示します。

ホストモードでは、このビットは、非周期的 Tx FIFO エンプティ割り込み (OTG\_GINTSTS の NPTXFE ビット) がトリガされるタイミングを示します。

0: OTG\_GINTSTS の NPTXFE 割り込みは、非周期的 Tx FIFO が半分エンプティであることを示します。

1: OTG\_GINTSTS の NPTXFE 割り込みは、非周期的 Tx FIFO が完全エンプティであることを示します。

ビット 6:1 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 6 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

Bits5 **DMAEN**: USB OTG HS の DMA 有効

0: コアはスレーブモードで動作します。

1: コアは DMA モードで動作します。

ビット 4:1 **HBSTLEN**: USB OTG HS のバースト長/タイプ

0000 シングル

0001 INCR

0011 INCR4

0101 INCR8

0111 INCR16

その他: 予約済み

ビット 0 **GINTMSK**: グローバル割り込みマスク

アプリケーションは、このビットを使用して、アプリケーション自身に対する割り込みラインのアサーションをマスクまたはマスク解除します。このビットの設定に関係なく、割り込みステータスレジスタはコアによって更新されます。

0: アプリケーションへの割り込みアサーションをマスクします。

1: アプリケーションへの割り込みアサーションをマスク解除します。

**注: デバイスモードとホストモードの両方でアクセス可能です。**

## 37.15.4 OTG USB 設定レジスタ (OTG\_GUSBCFG)

アドレスオフセット : 0x00C

リセット値 : 0x0000 1440

このレジスタを使用して、電源投入後、またはホストモードあるいはデバイスモードへの変化後に、コアを設定することができます。このレジスタは USB および USB-PHY 関連の設定パラメータを含んでいます。アプリケーションは AHB または USB のいずれかでトランザクションを開始する前に、このレジスタをプログラムする必要があります。初期プログラミングの後には、このレジスタを変更しないでください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FD MOD	FH MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRDT				HNP CAP	SRP CAP	Res.	PHY SEL	Res.	Res.	Res.	TOCAL		
		rw				rw	rw		r				rw		

### 注: USB OTG FS の設定レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FD MOD	FH MOD	Res.	Res.	Res.	ULPIPD	PTCI	PCCI	TSDPS	ULPIE VBUSI	ULPIE VBUSD	ULPIC SM	ULPIAR.	ULPIFS L.	Res.
	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYL PC.	Res.	TRDT				HNP CAP	SRP CAP	Res.	PHY SEL	Res.	Res.	Res.	TOCAL		
rw		rw				rw	rw		r				rw		

### 注: USB OTG HS の設定レジスタ

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **FDMOD**: 強制デバイスモード

このビットに 1 を書き込むと、コアは OTG\_ID 入力ピンに関係なく、強制的にデバイスモードになります。

0: 通常モード

1: 強制デバイスモード

強制ビットをセットした後、アプリケーションは、変更が有効になるまで少なくとも 25 ms 待つ必要があります。

**注: デバイスモードとホストモードの両方でアクセス可能です。**

ビット 29 **FHMOD**: 強制ホストモード

このビットに 1 を書き込むと、コアは OTG\_ID 入力ピンに関係なく、強制的にホストモードになります。

0: 通常モード

1: 強制ホストモード

強制ビットをセットした後、アプリケーションは、変更が有効になるまで少なくとも 25 ms 待つ必要があります。

**注: デバイスモードとホストモードの両方でアクセス可能です。**

ビット 28:24 予約済みであり、リセット値に保持する必要があります。

ビット 25:15 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

- ビット 25 **ULPIIPD** : USB OTG HS の ULPI インタフェース保護の無効化  
このビットは PHY に組み込まれた回路を制御して、リンクが stp とデータをトライステートにすると ULPI インタフェースを保護します。この機能のために使用するブルアップ抵抗またはブルダウン抵抗は無効化できません。詳細については ULPI 仕様を参照してください。  
0 : インタフェース保護回路を有効化  
1 : インタフェース保護回路を無効化
- ビット 24 **PTCI** : USB OTG HS のインジケータパススルー  
このビットは、内部  $V_{BUS}$  有効コンパレータを使用して、RX CMD の  $V_{BUS}$  状態で使用する前に、相補出力を評価するか否かを制御します。詳細については ULPI 仕様を参照してください。  
0 : 相補出力信号を内部  $V_{BUS}$  有効コンパレータで評価します。  
1 : 相補出力信号を内部  $V_{BUS}$  有効コンパレータで評価しません。
- ビット 23 **PCCI** : USB OTG HS の インジケータ相補  
このビットは PHY を制御して外部  $V_{bus}$  インジケータ入力信号を反転させて、相補出力を発生します。詳細については ULPI 仕様を参照してください。  
0 : PHY は外部  $V_{bus}$  インジケータ信号を反転させません。  
1 : PHY は外部  $V_{bus}$  インジケータ信号を反転させます。
- ビット 22 **TS DPS** : USB OTG HS の TermSel DLine パルシング選択  
このビットは SRP (セッションリクエストプロトコル) 時にデータラインパルスを駆動する `utmi_termselect` を選択します。  
0 : `utmi_txvalid` を使用するデータラインパルシング (デフォルト)  
1 : `utmi_termsel` を使用するデータラインパルシング
- ビット 21 **ULPIEBUSI** : USB OTG HS の ULPI 外部  $V_{BUS}$  インジケータ  
このビットは ULPI の PHY が外部  $V_{BUS}$  過電流のインジケータを使用するよう指定します。  
0 : PHY が内部  $V_{BUS}$  有効コンパレータを使用  
1 : PHY が外部  $V_{BUS}$  有効コンパレータを使用
- ビット 20 **ULPIEBUSD** : USB OTG HS の ULPI 外部  $V_{BUS}$  駆動  
このビットは、ULPI の PHY で  $V_{BUS}$  で 5 V を駆動するため内部電源または外部電源のいずれを使用するかを選択します。  
0 : PHY は 内部チャージポンプを使って  $V_{BUS}$  を駆動 (デフォルト)  
1 : PHY は外部電源を使って  $V_{BUS}$  を駆動
- ビット 19 **ULPICSM** : USB OTG HS の ULPI の クロックサスペンドモードビット  
このビットは ULPI PHY のインタフェース制御レジスタのクロックサスペンドモード ビットを設定します。このビットは、シリアルモードと `carKit` モードでのみ使用されます。  
0 : PHY はサスペンド中に内部クロックをパワーダウンさせます。  
1 : PHY は内部クロックをパワーダウンさせません。
- ビット 18 **ULPIAR** : USB OTG HS の ULPI 自動レジューム  
このビット PHY のインタフェース制御レジスタの自動レジュームビットを設定します。  
0 : PHY は自動レジューム機能を使用しません。  
1 : PHY は自動レジューム機能を使用します。
- ビット 17 **ULPIFSL** : USB OTG HS の ULPI FS/LS 選択  
アプリケーションではこのビットを使用し PHY シリアルインタフェースを選択します。このビットは PHY シリアルトランシーバが選択されたときのみ有効です。  
0 : ULPI インタフェース  
1 : ULPI FS/LS シリアルインタフェース
- ビット 16 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

### ビット 15 **PHYLPCS** : USB OTG HS の PHY 低電カクロック選択

このビットは 480 MHz または 48 MHz (低消費電力) モードを選択します。FS モードと LS モードでは PHY は 48 MHz クロックで動作して消費電力を削減できます。

0 : 480 MHz 内部 PLL クロック

1 : 48 MHz 外部クロック

480 MHz モードでは UTMI インタフェースは選択した 8 ビットまたは 16 ビットのデータ幅に応じ 60 または 30 MHz で動作します。48 MHz モードでは UTMI インタフェースは FS モードと LS モードで 48 MHz 動作します。

### ビット 14 予約済みであり、リセット値に保持する必要があります。

### ビット 13:10 **TRDT** : USB ターンアラウンド時間

ターンアラウンド時間を PHY クロックで設定します。

次の計算式を使用して TRDT の値を求めます。

$$\text{TRDT} = 4 * \text{AHB クロック} + 1 \text{ PHY クロック}$$

例 :

AHB クロック = 72 MHz (PHY クロック = 48) のとき、TRDT は 9 にセットされます。

AHB クロック = 48 MHz (PHY クロック = 48) のとき、TRDT は 5 にセットされます。

**注 :** デバイスモードでのみアクセス可能です。

### ビット 9 **HNPCAP** : HNP 機能

アプリケーションは、このビットを使用して OTG\_FS/OTG\_HS コントローラの HNP 機能を制御します。

0 : HNP 機能は無効です。

1 : HNP 機能は有効です。

**注 :** デバイスモードとホストモードの両方でアクセス可能です。

### ビット 8 **SRPCAP** : SRP 機能

アプリケーションは、このビットを使用して OTG\_FS/OTG\_HS コントローラの SRP 機能を制御します。コアが非 SRP 対応 B デバイスとして動作する場合、

デバイスは、接続された A デバイス (ホスト) に  $V_{BUS}$  を起動し、セッションを開始させるように要求することはできません。

0 : SRP 機能は無効です。

1 : SRP 機能は有効です。

**注 :** デバイスモードとホストモードの両方でアクセス可能です。

### ビット 7 予約済みであり、リセット値に保持する必要があります。

### ビット 6 **PHYSEL** : フルスピードシリアルトランシーバの選択

このビットは常に 1 であり、アクセスは読み出し専用です。

### ビット 5:3 予約済みであり、リセット値に保持する必要があります。

### ビット 2:0 **TCAL** : FS タイムアウトの較正

アプリケーションがこのフィールドでプログラムする PHY クロックの数は、PHY によって導入される追加遅延を考慮して、コアのフルスピードパケット間タイムアウト時間に追加されます。ライン状態条件の生成時に PHY によって導入される遅延は PHY ごとに異なる場合があるので、較正が必要なことがあります。

フルスピード動作での USB の標準タイムアウト値は 16~18 ビット時間です。アプリケーションは、エニュメレーションのスピードに基づいて、このフィールドをプログラムする必要があります。PHY クロックごとに追加されるビット時間の数は 0.25 ビット時間です。

## 37.15.5 OTG リセットレジスタ (OTG\_GRSTCTL)

アドレスオフセット : 0x10

リセット値 : 0x8000 0000

アプリケーションは、このレジスタを使用して、コア内のさまざまなハードウェア機能をリセットします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB IDL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXFNUM				TXF FLSH	RXF FLSH	Res.	FCRST	PSRST	CSRST	
					rw				rs	rs		rs	rs	rs	

**注 :** **USB OTG FS の設定レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB IDL	DMAR EQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXFNUM				TXF FLSH	RXF FLSH	Res.	Res.	PSRST	CSRST	
					rw				rs	rs		rs	rs	rs	

**注 :** **USB OTG HS の設定レジスタ**

ビット 31 **AHBIDL** : AHB マスタアイドル

AHB マスタステートマシンがアイドル状態にあることを示します。

**注 :** **デバイスモードとホストモードの両方でアクセス可能です。**

ビット 30:11 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 30 **DMAREQ** : USB OTG HS の DMA リクエスト信号有効

このビットは DMA リクエストが進行中であることを表示します。デバッグには使用しません。

ビット 29:11 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 10:6 **TXFNUM** : Tx FIFO 番号

これは Tx FIFO フラッシュビットを使用して一掃しなければならない FIFO 番号です。コアが Tx FIFO フラッシュビットをクリアするまでは、このフィールドを変更しないでください。

00000 :

- ホストモードでの非周期的 Tx FIFO の一掃
- デバイスモードでの Tx FIFO 0 の一掃

00001 :

- ホストモードでの周期的 Tx FIFO の一掃
- デバイスモードでの Tx FIFO 1 の一掃

00010 : デバイスモードでの Tx FIFO 2 の一掃

...

01111 : デバイスモードでの Tx FIFO 15 の一掃

10000 : デバイスモードまたはホストモードのすべての送信 FIFO を一掃します。

**注 :** **デバイスモードとホストモードの両方でアクセス可能です。**



## ビット 5 TXFFLSH : Tx FIFO の一掃

このビットは、単一またはすべての送信 FIFO を選択的にフラッシュしますが、コアがトランザクション中の場合は一掃できません。

アプリケーションは、コアが Tx FIFO への書き込み中か Tx FIFO からの読み出し中のどちらでもないことを確認した後でなければ、このビットに書き込むことはできません。以下のレジスタを使用して確認してください。

読み出し — NAK 有効割り込みを使用すると、コアが FIFO から読み出しを行っていないことを確認できます。

書き込み — OTG\_GRSTCTL の AHBIDL ビットで、コアが FIFO に書き込みを行っていないことを保証します。

FIFO が再設定される場合は通常一掃を実施することをお勧めします。デバイスエンドポイントが無効の間も、FIFO 一掃の実施をお勧めします。アプリケーションは、コアがこのビットをクリアするまで待つから、他の操作を実行しなければなりません。phy\_clk または hclk のより低速なクロックを使用してこのビットをクリアするには、8 クロックかかります。

**注：** デバイスモードとホストモードの両方でアクセス可能です。

## ビット 4 RXFFLSH : Rx FIFO の一掃

アプリケーションは、このビットを使用して Rx FIFO 全体を一掃できますが、まず、コアがトランザクション中でないことを確認する必要があります。

アプリケーションは、コアが Rx FIFO からの読み出し中、または Rx FIFO への書き込み中のどちらでもないことを確認した後でなければ、このビットに書き込んでなりません。

アプリケーションは、他の操作を実行する前に、このビットがクリアされるまで待つ必要があります。このビットがクリアされるまで、8 クロック (PHY または AHB クロックの中で最も低速なクロック) かかります。

**注：** デバイスモードとホストモードの両方でアクセス可能です。

ビット 3 予約済みであり、リセット値に保持する必要があります。

## ビット 2 FCRST : ホストフレームカウンタリセット

アプリケーションは、このビットに書き込むことによって、コア内のフレーム数カウンタをリセットします。フレームカウンタがリセットされると、コアによって次に送出される SOF のフレーム番号は 0 になります。

アプリケーションでこのビットに 1 が書き込まれた場合、このビットは数クロックサイクル内にコアによってクリアされますので、その値を読み戻すことはできません。

**注：** ホストモードでのみアクセス可能です。

## ビット 1 PSRST : 部分的なソフトリセット

内部ステートマシンをリセットしますが、エニユメレーション情報は保持されます。特定の PHY エラーからの復帰に使用できます。

**注：** デバイスモードとホストモードの両方でアクセス可能です。

ビット 0 CSRST : コアソフトリセット

以下のように HCLK および PHY クロックドメインをリセットします。  
 割り込みと、以下のビットを除くすべての CSR レジスタビットをクリアします。

- OTG\_PCGCCTL レジスタ の GATEHCLK ビット
- OTG\_PCGCCTL レジスタ の STPPCLK ビット
- OTG\_HCFG レジスタ の FSLSPCS ビット
- OTG\_DCFG レジスタ の DSPD ビット
- OTG\_DCTL レジスタ の SDIS ビット
- OTG\_GCCFG レジスタ
- OTG\_GPWRDN レジスタ

すべてのモジュールのステートマシン (AHB スレーブユニット用を除く) がアイドル状態にリセットされ、すべての送信 FIFO および受信 FIFO が一掃されます。

AHB 転送の最後のデータフェーズの完了後 AHB マスタ上のすべてのトランザクションは速やかに終了されます。USB 上のすべてのトランザクションをただちに終了します。

アプリケーションは、いつでもこのビットに書き込むことによって、コアをリセットできます。これはセルフクリアビットであり、コア内のすべての必要なロジックがリセットされた後、コアがこのビットをクリアします。コアの現在の状態によって、クリアには数クロックかかります。このビットがクリアされると、ソフトウェアは PHY ドメインにアクセスできるようになるまで少なくとも 3 PHY クロック待つ必要があります (同期遅延)。また、ソフトウェアは、操作を開始する前に、このレジスタのビット 31 が 1 にセットされていること (AHB マスタがアイドル状態) を確認する必要があります。

一般に、ソフトウェアリセットは、ソフトウェア開発時に使用され、上記の USB 設定レジスタの PHY 選択ビットを動的に変更するときにも使用されます。PHY を変更するときには PHY の対応クロックが選択され PHY ドメインで使用されます。新しいクロックが選択されたら、適切な動作のためには PHY ドメインをリセットする必要があります。

**注:** デバイスモードとホストモードの両方でアクセス可能です。

## 37.15.6 OTG コア割り込みレジスタ (OTG\_GINTSTS)

アドレスオフセット : 0x014

リセット値 : 0x1400 0020

このレジスタは、現在のモード (デバイスモードまたはホストモード) において、システムレベルのイベントのためにアプリケーションに割り込みをかけます。

このレジスタには、ホストモードでのみ有効なビットと、デバイスモードでのみ有効なビットがあります。このレジスタは、現在のモードも示します。rc\_w1 タイプの割り込みステータスビットをクリアするには、アプリケーションは、そのビットに 1 を書き込む必要があります。

FIFO ステータス割り込みは読み出し専用です。これらの割り込みの処理中に、ソフトウェアが FIFO への書き込みまたは FIFO からの読み出しを行うと、FIFO 割り込み条件は自動的にクリアされます。

初期化の前に割り込みが発生するのを避けるためには、アプリケーションは、割り込みビットのマスクを解除する前に OTG\_GINTSTS レジスタをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUP INT	SRQ INT	DISC INT	CIDS CHG	LPM INT	PTXFE	HCINT	HPRT INT	RST DET	Res.	IPXFR/ IN COMP ISO OUT	IISOI XFR	OEP INT	IEPINT	Res.	Res.
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	rc_w1		rc_w1	rc_w1	r	r		



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPF	ISOO DRP	ENUMD NE	USB RST	USB SUSP	ESUSP	Res.	Res.	GO NAK EFF	GI NAK EFF	NPTXFE	RXF LVL	SOF	OTG INT	MMIS	CMOD
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

**注：** **USB OTG FS の設定レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUP INT	SRQ INT	DISC INT	CIDS CHG	Res.	PTXFE	HCINT	HPRT INT	Res.	DATAF SUSP	IPXFR/ IN COMP ISO OUT	IISOI XFR	OEP INT	IEPINT	Res.	Res.
rc_w1	rc_w1	rc_w1	rc_w1		r	r	r		rc_w1	rc_w1	rc_w1	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPF	ISOO DRP	ENUMD NE	USB RST	USB SUSP	ESUSP	Res.	Res.	GO NAK EFF	GI NAK EFF	NPTXFE	RXF LVL	SOF	OTG INT	MMIS	CMOD
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

**注：** **USB OTG HS の設定レジスタ**

ビット 31 **WKUPINT** : レジューム/リモートウェイクアップ検出割り込み

– サスペンド (L2) または LPM (L1) 状態中のウェイクアップ割り込み

– サスペンド (L2) 中

デバイスモードでは、この割り込みは、USB 上でレジュームが検出されたときにアサートされます。ホストモードでは、この割り込みは USB 上でリモートウェイクアップが検出されたときにアサートされます。

– LPM (L1) 中

この割り込みは USB 上でホストから開始されたレジュームまたはデバイスから開始されたリモートウェイクアップのいずれかに対してアサートされます。

**注：** デバイスモードとホストモードの両方でアクセス可能です。

ビット 30 **SRQINT** : セッションリクエスト/新規セッション検出割り込み

ホストモードでは、この割り込みは、デバイスからのセッションリクエストが検出されたときにアサートされます。デバイスモードでは、この割り込みは、V<sub>BUS</sub> が B ペリフェラルデバイスに対して有効な範囲にあるときにアサートされます。デバイスモードとホストモードの両方でアクセス可能です。

ビット 29 **DISCINT** : 切断検出割り込み

デバイスの切断が検出されたときにアサートされます。

**注：** ホストモードでのみアクセス可能です。

ビット 28 **CIDSCHG** : コネクタ ID ステータス変化

コアはコネクタ ID ステータスに変化があったとき、このビットをセットします。

**注：** デバイスモードとホストモードの両方でアクセス可能です。

ビット 27 **LPMINT** : LPM 割り込み

デバイスモードでは、この割り込みは、デバイスが LPM トランザクションを受信し、非エラー応答で応答したときにアサートされます。

ホストモードでは、この割り込みは、デバイスが LPM トランザクションに非エラー応答で応答したとき、またはホストのコアがプログラムした回数 (OTG\_GLPMCFG の RETRYCNT) の LPM トランザクションを完了したときにアサートされます。

このフィールドは、OTG\_GLPMCFG の LPMCAP ビットが 1 にセットされたときにのみ有効です。

ビット 27 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

**ビット 26 PTXFE** : 周期的 TxFIFO エンプティ

周期的送信 FIFO が半エンプティか完全エンプティであり、周期的リクエストキューに少なくとも 1 つのエントリを書き込むためのスペースがあるときにアサートされます。半エンプティか完全エンプティステータスであるかは、OTG\_GAHBCFG レジスタの周期的 TxFIFO エンプティレベルビット (OTG\_GAHBCFG の PTXFELVL ビット) によって判断されます。

**注:** ホストモードでのみアクセス可能です。

**ビット 25 HCINT** : ホストチャネル割り込み

コアは、コアのチャネルの 1 つで割り込みがペンディング中であることを示すために、このビットをセットします (ホストモード)。アプリケーションは、OTG\_HAINT レジスタを読み出して、割り込みが発生しているチャネルの正確な番号を確認した後、対応する OTG\_HCINTx レジスタを読み出して、割り込みの正確な原因を確認する必要があります。アプリケーションは OTG\_HCINTx レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

**注:** ホストモードでのみアクセス可能です。

**ビット 24 HPRINT** : ホストポート割り込み

コアは、ホストモードの OTG\_FS/OTG\_HS コントローラポートの 1 つのポートステータスに変化があったことを示すために、このビットをセットします。アプリケーションは、OTG\_HPRT レジスタを読み出して、この割り込みの原因となった正確なイベントを特定する必要があります。アプリケーションは OTG\_HPRT レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

**注:** ホストモードでのみアクセス可能です。

**ビット 23 RSTDET** : リセット検出割り込み

デバイスモードでは、この割り込みは、デバイスがサスペンド状態にあるときに部分パワーダウンモードで USB 上にリセットが検出されたときにアサートされます。

**注:** デバイスモードでのみアクセス可能です。

ビット 23 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 22 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

**ビット 22 DATAFSUSP** : USB OTG HS のデータフェッチサスペンド

この割り込みは DMA モードでのみ有効です。この割り込みは TxFIFO スペースまたはリクエストキュースペースがないためコアが IN エンドポイントに対するデータフェッチを停止したことを表示します。アプリケーションでは、エンドポイント不一致アルゴリズムに対してこの割り込みを使います。例えば、エンドポイント不一致を検出後、アプリケーションは以下を実行します。

- グローバルな非周期的 IN NAK ハンドシェイクを設定
- IN エンドポイントを無効化
- FIFO を一掃
- IN トークンシーケンス学習キューからトークンシーケンスを決定
- エンドポイントを再度有効化

グローバル非周期的 IN NAK がクリアされ、コアは IN エンドポイントのデータをフェッチしておらず、かつ IN トークンを受信した場合、すなわちコアが「FIFO エンプティ時 IN トークン受信」割り込みを生成した場合、グローバル非周期的 IN NAK ハンドシェイクをクリアします。次に OTG は NAK 応答をホストへ送信します。このシナリオを回避するため、アプリケーションは OTG\_GINTSTS 内で FetSusp 割り込みをチェックすることができます。これにより、グローバル NAK ハンドシェイクをクリアする前に FIFO が満杯であることが保証されます。あるいは、グローバル NAK ハンドシェイクをクリアするとき、アプリケーションが「FIFO エンプティ中に IN トークンを受信」割り込みはマスクされることができません。

- ビット 21 **IPXFR** : 不完全な周期的転送  
ホストモードでは、コアは、現在のフレームでスケジュールされていて、まだペンディング中の不完全な周期的トランザクションが存在するとき、この割り込みビットをセットします。  
**INCOMPISOOUT** : 不完全アイソクロナス OUT 転送  
デバイスモードでは、コアは、現在のフレームで転送が完了していないアイソクロナス OUT エンドポイントが少なくとも 1 つあることを示すために、この割り込みをセットします。この割り込みは、このレジスタの周期的フレーム割り込み終了 (EOPF) ビットとともにアサートされます。
- ビット 20 **IISOIXFR** : 不完全アイソクロナス IN 転送  
コアは、現在のフレームで転送が完了していないアイソクロナス IN エンドポイントが少なくとも 1 つあることを示すために、この割り込みをセットします。この割り込みは、このレジスタの周期的フレーム割り込み終了 (EOPF) ビットとともにアサートされます。  
**注** : デバイスモードでのみアクセス可能です。
- ビット 19 **OEPINT** : OUT エンドポイント割り込み  
コアは、コアの OUT エンドポイントの 1 つで割り込みがペンディングされていることを示すために、このビットをセットします (デバイスモード)。アプリケーションは、OTG\_DAINTE レジスタを読み出して、割り込みが発生した OUT エンドポイントの正確な番号を確認した後、対応する OTG\_DOEPINTx レジスタを読み出して、割り込みの正確な原因を特定する必要があります。アプリケーションは、対応する OTG\_DOEPINTx レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。  
**注** : デバイスモードでのみアクセス可能です。
- ビット 18 **IEPINT** : IN エンドポイント割り込み  
コアは、コアの IN エンドポイントの 1 つで割り込みがペンディングされていることを示すために、このビットをセットします (デバイスモード)。アプリケーションは、OTG\_DAINTE レジスタを読み出して、割り込みが発生した IN エンドポイントの正確な番号を確認した後、対応する OTG\_DIEPINTx レジスタを読み出して、割り込みの正確な原因を特定する必要があります。アプリケーションは、対応する OTG\_DIEPINTx レジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。  
**注** : デバイスモードでのみアクセス可能です。
- ビット 17:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **EOPF** : 周期的フレーム終了割り込み  
現在のフレームで、OTG\_DCFG レジスタの周期的フレームインターバルフィールド (OTG\_DCFG の PFIVL ビット) で指定された時間に達したことを示します。  
**注** : デバイスモードでのみアクセス可能です。
- ビット 14 **ISOODRP** : アイソクロナス OUT パケット欠落割り込み  
コアは、アイソクロナス OUT エンドポイントの最大サイズのパケットに対応できるだけの十分なスペースが Rx FIFO になかったために Rx FIFO にアイソクロナス OUT パケットを書き込めなかったとき、このビットをセットします。  
**注** : デバイスモードでのみアクセス可能です。
- ビット 13 **ENUMDNE** : エンユメレーション終了  
コアは、スピードのエンユメレーションが完了したことを示すために、このビットをセットします。アプリケーションは、OTG\_DSTS レジスタを読み出して、エンユメレーションされた速度を取得する必要があります。  
**注** : デバイスモードでのみアクセス可能です。
- ビット 12 **USBRST** : USB リセット  
コアは USB 上でリセットが検出されたことを示すために、このビットをセットします。  
**注** : デバイスモードでのみアクセス可能です。

**ビット 11 USBSUSP** : USB サスペンド

コアは USB 上でサスペンドが検出されたことを示すために、このビットをセットします。コアは、データライン上で長時間アクティビティが 1 つもなかった場合、サスペンド状態に入ります。

**注:** デバイスモードでのみアクセス可能です。

**ビット 10 ESUSP** : アーリーサスペンド

コアは USB 上で 3 ms 間、アイドル状態が検出されたことを示すために、このビットをセットします。

**注:** デバイスモードでのみアクセス可能です。

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

**ビット 7 GONAKEFF** : グローバル OUT NAK 有効

アプリケーションによってセットされた OTG\_DCTL レジスタのセットグローバル OUT NAK ビット (OTG\_DCTL の SGONAK ビット) がコアで有効になったことを示します。このビットは、OTG\_DCTL レジスタのクリアグローバル OUT NAK ビット (OTG\_DCTL の CGONAK ビット) に書き込むことによってクリアできます。

**注:** デバイスモードでのみアクセス可能です。

**ビット 6 GINAKEFF** : グローバル IN 非周期的 NAK 有効

アプリケーションによってセットされた OTG\_DCTL レジスタのセットグローバル 非周期的 IN NAK ビット (OTG\_DCTL の SGINAK ビット) がコアで有効になったことを示します。すなわち、コアは、アプリケーションによってセットされたグローバル IN NAK ビットをサンプリングしました。このビットは、OTG\_DCTL レジスタのクリアグローバル非周期的 IN NAK ビット (OTG\_DCTL の CGI NAK ビット) をクリアすることによってクリアできます。

この割り込みは、必ずしも NAK ハンドシェイクが USB に送出されることを意味しません。STALL ビットが NAK ビットに優先します。

**注:** デバイスモードでのみアクセス可能です。

**ビット 5 NPTXFE** : 非周期的 Tx FIFO エンプティ

この割り込みは、非周期的 Tx FIFO が半エンプティか完全エンプティであり、非周期的送信リクエストキューに少なくとも 1 つのエントリを書き込むスペースがあるときにアサートされます。半分エンプティか完全エンプティであるかは、OTG\_GAHBCFG レジスタの非周期的 Tx FIFO エンプティレベルビット (OTG\_GAHBCFG の TXFELVL ビット) によって判断されます。

**注:** ホストモードでのみアクセス可能です。

**ビット 4 RXFLVL** : Rx FIFO 非エンプティ

Rx FIFO から読み出されるべきペンディング中のパケットが少なくとも 1 つあることを示します。

**注:** ホストモードとデバイスモードの両方でアクセス可能です。

**ビット 3 SOF** : フレーム開始

ホストモードでは、コアは SOF (FS) またはキーブアラライブ (LS) が USB で送信されることを示すために、このビットをセットします。アプリケーションは、このビットに 1 を書き込むことによって、割り込みをクリアする必要があります。

デバイスモードでは、コアは、USB 上で SOF トークンが受信されたことを示すために、このビットをセットします。アプリケーションは、OTG\_DSTS レジスタを読み出すことによって、現在のフレーム番号を知ることができます。この割り込みは、コアが FS で動作しているときのみ確認できます。

**注:** **注:** このレジスタはパワーオンリセット後すぐに読み出されると“1”を返すことがあります。パワーオンリセット後すぐに“1”が読み出された場合、そのレジスタのビットが、SOF が送信された (ホストモードの場合) または SOF が受信された (デバイスモードの場合) ことを示すわけではありません。この割り込みの読み出し値は、ホストとデバイス間の有効な接続が確立されて初めて有効になります。パワーオンリセット後にビットがセットされた場合、アプリケーションはそのビットをクリアすることができます。

**注:** ホストモードとデバイスモードの両方でアクセス可能です。

**ビット 2 OTGINT** : OTG 割り込み

コアは、OTG プロトコルイベントが完了したことを示すために、このビットをセットします。アプリケーションは OTG 割り込みステータスレジスタ (OTG\_GOTGINT) を読み出して、この割り込みの原因となった正確なイベントを確認する必要があります。アプリケーションは OTG\_GOTGINレジスタの該当するステータスビットをクリアすることによって、このビットをクリアする必要があります。

**注:** ホストモードとデバイスモードの両方でアクセス可能です。

**ビット 1 MMIS** : モード不一致割り込み

コアは、アプリケーションが以下へのアクセスを試みているときに、このビットをセットします。

- ホストモードレジスタ。コアがデバイスモードで動作しているとき。
- デバイスモードレジスタ。コアがホストモードで動作しているとき。

レジスタへのアクセスは、OKAY 応答によって AHB 上で完了しますが、内部的にはコアによって無視され、コアの動作には影響を与えません。

**注:** ホストモードとデバイスモードの両方でアクセス可能です。

**ビット 0 CMOD** : 現在の動作モード

現在の動作モードを示します。

- 0 : デバイスモード
- 1 : ホストモード

**注:** ホストモードとデバイスモードの両方でアクセス可能です。

## 37.15.7 OTG 割り込みマスクレジスタ (OTG\_GINTMSK)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

このレジスタは、コア割り込みレジスタとともに働き、アプリケーションに割り込みをかけます。割り込みビットがマスクされると、そのビットに関連する割り込みは生成されません。ただし、その割り込みに対応するコア割り込みレジスタ (OTG\_GINTSTS) のビットは、セットされたままです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	Res.	IPXFRM/IISOIXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	r	rw		rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

**注:** USB OTG FS の設定レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	FSUSPM	IPXFRM/IISOIXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

**注:** USB OTG HS の設定レジスタ

- ビット 31 **WUIM** : レジューム/リモートウェイクアップ検出割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 30 **SRQIM** : セッションリクエスト/新規セッション検出割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 29 **DISCINT** : 切断検出割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 28 **CIDSCHGM** : コネクタ ID ステータス変化マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 27 **LPMINTM** : LPM 割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 26 **PTXFEM** : 周期的 Tx FIFO エンプティマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードでのみアクセス可能です。**
- ビット 25 **HCIM** : ホストチャネル割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードでのみアクセス可能です。**
- ビット 24 **PRTIM** : ホストポート割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードでのみアクセス可能です。**
- ビット 23 **RSTDETM** : リセット検出割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 22 予約済みであり、USB OTG FS のリセット値に保持する必要があります。
- ビット 22 **FSUSPM** : USB OTG HS のデータフェッチサスペンドマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
ペリフェラルモードでのみアクセス可能です。

- ビット 21 **IPXFRM** : 不完全周期的転送マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **ホストモードでのみアクセス可能です。**  
**IISOXFRM** : 不完全アイソクロナス OUT 転送マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 20 **IISOXFRM** : 不完全アイソクロナス IN 転送マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 19 **OEPINT** : OUT エンドポイント割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 18 **IEPINT** : IN エンドポイント割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 17:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **EOPFM** : 周期的フレーム終了割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 14 **ISOODRPM** : アイソクロナス OUT パケット欠落割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 13 **ENUMDNEM** : エニユメレーション終了マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 12 **USBRST** : USB リセットマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 11 **USBSUSPM** : USB サスペンドマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 10 **ESUSPM** : アーリーサスペンドマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注:** **デバイスモードでのみアクセス可能です。**
- ビット 9:8 予約済みであり、リセット値に保持する必要があります。

- ビット 7 **GONAKEFFM** : グローバル OUT NAK 有効マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** デバイスモードでのみアクセス可能です。
- ビット 6 **GINAKEFFM** : グローバル非周期的IN NAK 有効マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** デバイスモードでのみアクセス可能です。
- ビット 5 **NPTXFEM** : 非周期的 Tx FIFO エンプティマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** ホストモードでのみアクセス可能です。
- ビット 4 **RXFLVLM** : 受信 FIFO非エンブティマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** デバイスモードとホストモードの両方でアクセス可能です。
- ビット 3 **SOFM** : フレーム開始マスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** デバイスモードとホストモードの両方でアクセス可能です。
- ビット 2 **OTGINT** : OTG 割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** デバイスモードとホストモードの両方でアクセス可能です。
- ビット 1 **MMISM** : モード不一致割り込みマスク  
0 : 割り込みはマスクされます。  
1 : 割り込みはマスクされません。  
**注 :** デバイスモードとホストモードの両方でアクセス可能です。
- ビット 0 予約済みであり、リセット値に保持する必要があります。

## 37.15.8 OTG\_FS 受信ステータスデバッグ読み出し/OTGステータス読み出しおよびポップレジスタ (OTG\_GRXSTSR/OTG\_GRXSTSP)

読み出しアドレスオフセット : 0x01C

ポップアドレスオフセット : 0x020

リセット値 : 0x0000 0000

受信ステータスデバッグ読み出しレジスタを読み出すと、受信 FIFO の最上部の内容が返されます。受信ステータス読み出しおよびポップレジスタを読み出すと Rx FIFO の最上部のデータエントリも取り出され (ポップされ) ます。

受信ステータスの内容は、ホストモードとデバイスモードで異なった解釈をする必要があります。コアは、受信 FIFO がエンブティのときには、受信ステータスのポップ/読み出しを無視して、0x0000 0000 の値を返します。アプリケーションは、コア割り込みレジスタの受信 FIFO 非エンブティビット (OTG\_GINTSTS の RXFLVL ビット) がアサートされたときのみ、受信ステータス FIFO をポップしなければなりません。

ホストモード :

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

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:17 **PKTSTS** : パケットステータス

受信パケットのステータスを示します。

0010 : IN データパケットが受信されました。

0011 : IN 転送が完了しました (割り込みをトリガします)。

0101 : データトグルエラー (割り込みをトリガします)。

0111 : チャネル停止 (割り込みをトリガします)。

その他 : 予約済み

ビット 16:15 **DPID** : データ PID

受信パケットのデータ PID を示します。

00 : DATA0

10 : DATA1

01 : DATA2

11 : MDATA

ビット 14:4 **BCNT** : バイトカウント

受信した IN データパケットのバイト数を示します。

ビット 3:0 **CHNUM** : チャネル番号

現在の受信パケットが属するチャネル番号を示します。

# 参考資料

RM0385

USB On-The-Go フルスPEED/ハイスPEED (OTG\_FS/OTG\_HS)

## デバイスモード :

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:21 **FRMNUM** : フレーム番号

これは USB 上でパケットが受信されるフレーム番号の下位 4 ビットです。このフィールドは、アイソクロナス OUT エンドポイントがサポートされる場合にのみサポートされます。

ビット 20:17 **PKTSTS** : パケットステータス

受信パケットのステータスを示します。

0001 : グローバル OUT NAK (割り込みをトリガします)。

0010 : OUT データパケットが受信されました。

0011 : OUT 転送が完了しました (割り込みをトリガします)。

0100 : SETUP トランザクションが完了しました (割り込みをトリガします)。

0110 : SETUP データパケットが受信されました。

その他 : 予約済み

ビット 16:15 **DPID** : データ PID

受信した OUT データパケットのデータ PID を示します。

00 : DATA0

10 : DATA1

01 : DATA2

11 : MDATA

ビット 14:4 **BCNT** : バイトカウント

受信したデータパケットのバイト数を示します。

ビット 3:0 **EPNUM** : エンドポイント番号

現在の受信パケットが属するエンドポイント番号を示します。



DocID026670 Rev 2

1347/1653

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

## 37.15.9 OTG 受信 FIFO サイズレジスタ (OTG\_GRXFSIZ)

アドレスオフセット : 0x024

リセット値 : 0x0000 0200

アプリケーションは RxFIFO に割り当てられなければならない RAM サイズをプログラムできます。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RXFD** : Rx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

最大値は 1024 です。

プログラムされた値は、使用可能な FIFO メモリの割り当てに従う必要があり、パワーオン値を超えてはなりません。

## 37.15.10 OTG ホスト非周期的送信 FIFO サイズレジスタ (OTG\_HNPTXFSIZ) / エンドポイント 0 送信 FIFO サイズ (OTG\_DIEPTXF0)

アドレスオフセット : 0x028

リセット値 : 0x0200 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPTXFD/TX0FD															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSA/TX0FSA															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ホストモード

ビット 31:16 **NPTXFD** : 非周期的 Tx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

プログラムされた値は、使用可能な FIFO メモリの割り当てに従う必要があり、パワーオン値を超えてはなりません。

ビット 15:0 **NPTXFSA** : 非周期的送信 RAM 開始アドレス

このフィールドは、非周期的送信 FIFO RAM のメモリ開始アドレスを設定します。

デバイスモード

ビット 31:16 **TX0FD** : エンドポイント 0 TxFIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

プログラムされた値は、使用可能な FIFO メモリの割り当てに従う必要があり、パワーオン値を超えてはなりません。

ビット 15:0 **TX0FSA** : エンドポイント 0 送信 RAM 開始アドレス

このフィールドは、エンドポイント 0 送信 FIFO RAM のメモリ開始アドレスを設定します。

## 37.15.11 OTG 非周期的送信 FIFO/キューステータスレジスタ (OTG\_HNPTXSTS)

アドレスオフセット : 0x02C

リセット値 : 0x0008 0200

**注 :** デバイスモードでは、このレジスタは無効です。

この読み出し専用レジスタは、非周期的 Tx FIFO および非周期的送信リクエストキューの空きスペース情報を含みます。

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:24 **NPTXQTOP** : 非周期的送信リクエストキューの先頭

MAC によって処理中の非周期的 Tx リクエストキューのエントリ。

ビット 30:27チャンネル/エンドポイント番号

ビット 26:25

00 : IN/OUT トークン

01 : ゼロ長送信パケット (デバイス IN/ホスト OUT)

11 : チャンネル停止コマンド

ビット 24 : 終了 (選択されたチャンネル/エンドポイントの最後のエントリ)

ビット 23:16 **NPTQXSAV** : 非周期的送信リクエストキューの使用可能スペース

非周期的送信リクエストキューの使用可能な空きスペースの量を示します。このキューは IN および OUT リクエストの両方を保持します。

0 : 非周期的送信リクエストキューは満杯です。

1 : 1 位置が使用可能です。

2 : 2 位置が使用可能です。

n : n 位置が使用可能です (0 ≤ n ≤ 8)。

その他 : 予約済み

ビット 15:0 **NPTXFSAV** : 使用可能な非周期的 Tx FIFO スペース  
 非周期的 Tx FIFO 内で使用できる空きスペースの量を示します。  
 値は 32 ビットワード単位です。  
 0 : 非周期的 Tx FIFO は満杯です。  
 1 : 1ワードが使用可能です。  
 2 : 2 ワードが使用可能です。  
 n : n ワードが使用可能です (0 ≤ n ≤ 512)。  
 その他 : 予約済み

## 37.15.12 OTG I<sup>2</sup>C アクセスレジスタ (OTG\_GI2CCTL)

アドレスオフセット : 0x030

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
BSYDNE	RW	Res.	I2CD ATSE	I2CDEVADR			Res.	ACK (確 認応答)	I2CEN	ADDR						
rW	rW		rW	rW	rW		rW	rW	rW	rW	rW	rW	rW	rW	rW	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
REGADDR								RWDATA								
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	

ビット 31 **BSYDNE** : I2C ビジー/完了

アプリケーションはこのビットに 1 を設定して I<sup>2</sup>C インタフェースのリクエストを開始します。転送が完了すると、コアはこのビットに 0 を設定します。このビットがセットされ I<sup>2</sup>C インタフェースがビジーであることを表示しているかぎり、アプリケーションはインタフェースの次のリクエストを開始できません。

ビット 30 **RW** : 読み出し/書き込みインジケータ

このビットは、インタフェースでレジスタ読み出しまたは書き込み転送を実行する必要があることを表示します。

0 : 書き込み  
 1 : 読み出し

**注 :** パースト読み出し/書き込みは、レジスタに対してサポートされていません。

ビット 29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **I2CDATSE0** : I<sup>2</sup>C DatSe0 USB モード

このビットを使ってフルスピードインタフェース USB モードを選択します。

0 : VP\_VM USB モード  
 1 : DAT\_SE0 USB モード

ビット 27:26 **I2CDEVADR** : I<sup>2</sup>C デバイスアドレス

このビットは、コアが OTG シグナリングに使用する USB 1.1 フルスピードシリアルトランシーバ上の I<sup>2</sup>C スレーブのアドレスを選択します。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **ACK** : I<sup>2</sup>C ACK

このビットは ACK 応答が I<sup>2</sup>C スレーブから受信されたか否かを示します。これは、アプリケーションが I<sup>2</sup>C アクセスを開始した後にコアが BSYDNE をクリアしたときに有効になります。

0 : NAK  
 1 : ACK (確認応答)

ビット 23 **I2CEN** : I<sup>2</sup>C 有効化

このビットは I<sup>2</sup>C マスタを有効化し、I<sup>2</sup>C インタフェース上でトランザクションを開始させます。

ビット 22:16 **ADDR** : I<sup>2</sup>C アドレス

この 7 ビットの I<sup>2</sup>C デバイスアドレスは、アプリケーションが外部 I<sup>2</sup>C スレーブにアクセスするとき使用するもので、USB 1.1 OTG フルスピードシリアルトランシーバの I<sup>2</sup>C スレーブを含みます。

ビット 15:8 **REGADDR** : I<sup>2</sup>C レジスタアドレス

これらのビットを使うと、読み出しまたは書き込み対象のレジスタアドレスをプログラムすることができます。

ビット 7:0 **RWDATA** : I<sup>2</sup>C 読み出し/書き込みデータ

レジスタ読み出し動作後、これらのビットはアプリケーションの読み出しデータを保持します。書き込み動作では、アプリケーションはこのレジスタを使用してレジスタへ書き込むデータをプログラムすることができます。

**注 :** **USB OTG HS にのみ適用される設定レジスタ**

### 37.15.13 OTG 一般コア設定レジスタ (OTG\_GCCFG)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

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

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **VBDEN** : USB V<sub>BUS</sub> 検出有効化

V<sub>BUS</sub> レベル検出用コンパレータを有効にして、V<sub>BUS</sub> の有効レベルを検出します (USB ホストおよびデバイス動作用の V<sub>BUS</sub> パッド上で)。HNP または SRP サポートが有効であれば、V<sub>BUS</sub> コンパレータは VB DEN 値とは関係なく、自動的に有効化されます。

0 : V<sub>BUS</sub> 検出無効

1 : V<sub>BUS</sub> 検出有効

ビット 20:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **PWRDWN** : パワーダウン制御

送信/受信のためにトランシーバを起動するために使用されます。リセットされている場合、トランシーバはパワーダウン状態に保たれます。0 : USB FS トランシーバ無効

1 : USB FS トランシーバ有効

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 37.15.14 OTG コア ID レジスタ (OTG\_CID)

アドレスオフセット : 0x03C

リセット値 : 0x0000 2000

これは、製品 ID を含む読み出し専用レジスタです。

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

ビット 31:0 **PRODUCT\_ID** : 製品 ID フィールド

アプリケーションによってプログラム可能な ID フィールド。

## 37.15.15 OTG コア LPM 設定レジスタ (OTG\_GLPMCFG)

アドレスオフセット : 0x54

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EN BESL	LPMRCNTSTS			SND LPM	LPMRCNT			LPMCHIDX				L1RSMOK
			rW	r	r	r	rS	rW	rW	rW	rW	rW	rW	rW	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLP STS	LPMRSP		L1DS EN	BESLTHRS				L1SS EN	REM WAKE	BESL				LPM ACK (確認応答)	LPM EN
r	r	r	rW	rW	rW	rW	rW	rW	rW/r	rW/r	rW/r	rW/r	rW/r	rW	rW

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **ENBESL** : ベストエフォート型サービス遅延の有効化

このビットは LPM 正誤表で定義されているように BESL 機能を有効にします。

0 : コアは以下の文書に記載されているように機能します。

*USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007*

1 : コアは LPM 正誤表に記載されているように機能します。

*USB 2.0 ECN 用の正誤表 : Link Power Management (LPM) - 7/2007*

**注 :** この文書では更新された動作 (LPM 正誤表に記載) のみが考慮されているため、アプリケーションソフトウェアで ENBESL ビットを 1 にセットする必要があります。

ビット 27:25 **LPMRCNTSTS** : LPM 再試行カウントステータス

現在の LPM シーケンスでは、送信すべき LPM ホストの再試行回数がまだ残っています。

**注 :** ホストモードでのみアクセス可能です。

ビット 24 **SNLPM** : 送信 LPM トランザクション

アプリケーションソフトウェアがこのビットをセットすると、2つのトークン EXT および LPM を含む LPM トランザクションが送信されます。デバイスから有効な応答 (STALL、NYET、または ACK) を受信した時点、またはコアがプログラムされた回数の LPM 再試行を送信し終えた時点で、このビットはハードウェアによってクリアされます。

**注:** このビットは、ホストがローカルポートに接続されている場合のみセットする必要があります。

**注:** ホストモードでのみアクセス可能です。

ビット 23:21 **LPMRCNT** : LPM 再試行カウント

デバイスが ERROR 応答を返すと、このビットは追加の LPM 再試行回数となります。追加の再試行は、有効なデバイスからの応答 (STALL、NYET、または ACK) が受信されるまで、ホストによって実行されます。

**注:** ホストモードでのみアクセス可能です。

ビット 20:17 **LPMCHIDX** : LPM チャネルインデックス

ローカルデバイスへの LPM トランザクションの送信中に LPM トランザクションを適用すべきチャネルの番号を示します。LPM チャネルインデックスに基づき、コアは自動的に対応するチャネルでプログラムされたデバイスアドレスとエンドポイント番号を LPM トランザクションに挿入します。

**注:** ホストモードでのみアクセス可能です。

ビット 16 **L1RSMOK** : スリープ状態レジューム OK

デバイスまたはホストがスリープ状態からレジュームを開始できることを示します。このビットは LPM スリープ状態 (L1) のときに有効です。このビットは 50  $\mu$ s ( $T_{L1Residency}$ ) の遅延後に SLEEP モードでセットされます。

このビットは、SLPSTS = 0 のときにリセットされます。

1: アプリケーションまたはホストはスリープ状態からレジュームを開始できます。

0: アプリケーションまたはホストはスリープ状態からレジュームを開始できません。

ビット 15 **SLPSTS** : ポートスリープステータス

**デバイスモードの場合**

このビットは、USB バスにスリープ状態が存在する限りセットされます。コアは、ACK 応答が LPM トランザクションに送信され、 $T_{L1TokenRetry}$  タイマが時間切れになった時点で、スリープ状態に入ります。PHY クロックを停止させるには、アプリケーションで OTG\_PCGCCTL の STPPCLK ビットをセットすることにより、PHY サスペンド入力信号をアサートする必要があります。

アプリケーションはスリープへの遷移を確認する際に、LPMRSP の ACK ではなく、SLPSTS を信頼する必要があります。

コアは、以下の場合にスリープ状態から抜け出します。

- USB ライン上で何らかのアクティビティがあったとき。
- アプリケーションが OTG\_DCTL の RWUSIG ビットへ書き込むときやデバイスをリセットまたはソフト切断するとき。

**ホストモードの場合**

デバイスから ACK 応答を受け取ったローカルポートへのコアによる LPM トランザクションの成功の副作用として、ホストはスリープ状態 (L1) へ遷移します。このビットの読み出し値は、ポートの現在のスリープ状態を反映しています。

コアは、次の動作後にこのビットをクリアします。

- コアがリモート L1 ウェイクアップ信号を検出します、
- アプリケーションで、OTG\_HPRT レジスタの PRST ビットまたは PRES ビットをセットします、または
- アプリケーションで、コア割り込みレジスタの L1 レジューム/リモートウェイクアップ検出割り込みビットまたは切断検出割り込みビット (OTG\_GINTSTS の WKUPINT または DISCINT ビット) をセットします。

0: コアは L1 状態ではありません。

1: コアは L1 状態です。

ビット 14:13 **LPMRST** : LPM 応答

**デバイスモードの場合**

受信した LPM トランザクションへのコアの応答は、これら 2 つのビットに反映されます。

**ホストモードの場合**

ローカルデバイスから受け取った LPM トランザクションへのハンドシェイク応答

11 : ACK (確認応答)

10 : NYET

01 : STALL

00 : ERROR (ハンドシェイク応答なし)

ビット 12 **L1DSEN** : L1 ディープスリープ有効化

L1 SLEEP モードにおける PHY のサスペンドを有効にします。L1 SLEEP モード時に最大限の節電を行うために、このビットはいかなる場合もアプリケーションソフトウェアによって 1 にセットする必要があります。

ビット 11:8 **BESLTHRS** : BESL 閾値

**デバイスモードの場合**

BESL 値がこの BESL\_Thres[3:0] フィールドの規定値以上である場合、コアは PHY を L1 状態でディープ低電力モードに移行させます。

**ホストモードの場合**

コアは PHY を L1 状態でディープ低電力モードに移行させます。BESLTHRS[3:0] は、デバイスから開始されたレジュームを検出した場合に、レジュームシグナリングに USB バス上のホスト (T<sub>L1HubDrvResume2</sub>) が反映されるまでの時間を指定します。

ホストモードでは、BESLTHRS には 1100b より大きい値を設定してはなりません。この値が T<sub>L1HubDrvResume2</sub> の最大値を超えてしまうからです。

Thres[3:0]ホストモードレジュームシグナリング時間 (µs)

0000 : 75

0001 : 100

0010 : 150

0011 : 250

0100 : 350

0101 : 450

0110 : 950

その他の値は予約済みです。

ビット 7 **L1SSEN** : L1 シャロースリープ有効化

L1 SLEEP モードにおける PHY のサスペンドを有効にします。L1 SLEEP モード時に最大限の節電を行うために、このビットはいかなる場合もアプリケーション切り替えによって 1 にセットする必要があります。

ビット 6 **REMWAKE** : bRemoteWake 値

**ホストモードの場合**

LPM トランザクションの wIndex フィールドの送信されるリモートウェイクアップの値

**デバイスモード (読み出し専用) の場合**

このフィールドは、ACK、NYET、または STALL 応答が LPM トランザクションに送信される場合に、受信した LPM トークン bRemoteWake bmAttribute で更新されます。

**ビット 5:2 BESL** : ベストエフォート型サービス遅延**ホストモードの場合**

LPM トランザクションの送信される BESL の値この値は、ホストから開始されたレジュームの処理時間 ( $T_{L1HubDrvResume1}$ ) の間にレジュームを開始するためにも使用されます。

**デバイスモード (読み出し専用) の場合**

このフィールドは、ACK、NYET、または STALL 応答が LPM トランザクションに送信される場合に、受信した LPM トークン BESL bmAttribute で更新されます。

BESL[3:0] $T_{BESL}$  ( $\mu$ s)

0000 : 125

0001 : 150

0010 : 200

0011 : 300

0100 : 400

0101 : 500

0110 : 1000

0111 : 2000

1000 : 3000

1001 : 4000

1010 : 5000

1011 : 6000

1100 : 7000

1101 : 8000

1110 : 9000

1111 : 10000

**ビット 1 LPMACK** : LPM トークン確認応答有効化

デバイスのアプリケーションソフトウェアによって事前にプログラムされた LPM トークンへのハンドシェイク応答

1 : ACK (確認応答)

ACK は事前にプログラムされているものの、コア (デバイス) は成功した LPM トランザクションに対してのみ ACK 応答を返します。LPM トランザクションは次の場合に成功します。

– EXT トークンまたは LPM トークンのどちらにも PID/CRC5 エラーがない (それ以外の場合は ERROR)。

– LPM トランザクションで受信した有効な bLinkState = 0001B (L1) (それ以外の場合は STALL)。

– 送信シーケンスにペンディング中のデータがない (それ以外の場合は NYET)。

0 : NYET

事前にプログラムされたソフトウェアビットは、次の場合に、LPM トークンへの応答に上書きされません。

– 受信した bLinkState が L1 でない場合 (STALL 応答)、または

– いずれかの LPM トークンパケットで、破損によるエラーが検出された場合 (エラー応答)

**注:** デバイスモードでのみアクセス可能です。

**ビット 0 LPMEN** : LPM サポート有効化

アプリケーションは、このビットを使用して OTG\_FS/OTG\_HS コアの LPM 機能を制御します。

コアが非 LPM 対応ホストとして動作する場合、ホストは接続されたデバイスまたはハブに LPM モードを有効にするように要求することはできません。

コアが非 LPM 対応デバイスとして動作する場合、デバイスはいかなる LPM トランザクションにも応答できません。

0 : LPM 機能は無効です。

1 : LPM 機能は有効です。

## 37.15.16 OTG ホスト周期的送信 FIFO サイズレジスタ (OTG\_HPTXFSIZ)

アドレスオフセット : 0x100

リセット値 : 0x0200 0400

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

ビット 31:16 **PTXFD** : ホスト周期的 Tx FIFO 深さ  
 この値は 32 ビットワード単位です。  
 最小値は 16 です。

ビット 15:0 **PTXSA** : ホスト周期的 Tx FIFO の開始アドレス  
 このフィールドは、周期的送信 FIFO RAM のメモリ開始アドレスを設定します。

## 37.15.17 OTG デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG\_DIEPTXFx)(x = 1..5[FS] / 7[HS]、ここで x は FIFO\_number)

アドレスオフセット :  $0x104 + (\text{FIFO\_number} - 1) * 0x04$

リセット値 :

$\text{FIFO\_number} = 7[\text{HS}] / 5[\text{FS}] : 0x0200\ 0200 + (7[\text{HS}] / 5[\text{FS}] * 0x200)$

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

ビット 31:16 **INEPTXFD** : IN エンドポイント Tx FIFO 深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

ビット 15:0 **INEPTXSA** : IN エンドポイント FIFOx 送信 RAM 開始アドレス

このフィールドは IN エンドポイント送信 FIFOx のメモリ開始アドレスを含みます。アドレスは 32 ビットメモリロケーションに整列されている必要があります。

## 37.15.18 ホストモードレジスタ

レジスタの説明で示されているビット値は、特に記載がない限りバイナリで表されています。

ホストモードレジスタは、ホストモードでのコアの動作に影響を与えます。ホストモードレジスタは、結果が定義されていないため、デバイスモードではアクセスしないでください。ホストモードレジスタは、次のように分類することができます。

## 37.15.19 OTG ホスト設定レジスタ (OTG\_HCFG)

アドレスオフセット : 0x400

リセット値 : 0x0000 0000

このレジスタは、電源投入後のコアを設定します。ホストを初期化した後は、このレジスタを変更しないでください。

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **FSLSS** : FS および LS のみのサポート

アプリケーションは、このビットを使用して、コアのエnumレーション速度を制御します。このビットを使用して、アプリケーションは、接続されたデバイスが HS トラフィックをサポートする場合でも、FS ホストとしてコアにエnumレーションさせることができます。初期プログラミングの後は、このフィールドを変更しないでください。

1: 接続されたデバイスが HS トラフィックをサポートしている場合でも、FS/LS のみ (読み出し専用)。

ビット 1:0 **FSLSPCS** : FS/LS PHY クロック選択

コアが FS ホストモードのとき、

01 : PHY クロックは 48 MHz で動作しています。

その他 : 予約済み

コアが LS ホストモードのとき、

00 : 予約済み

01 : PHY クロック周波数として 48 MHz を選択

10 : PHY クロック周波数として 6 MHz を選択

11 : 予約済み

**注 :** **FSLSPCS** は、接続されたデバイスの速度に応じて、接続イベント時にセットする必要があります (このビットを変更した後、ソフトウェアリセットを実行する必要があります)。

## 37.15.20 OTG ホストフレームインターバルレジスタ (OTG\_HFIR)

アドレスオフセット : 0x404

リセット値 : 0x0000 EA60

このレジスタは OTG\_FS/OTG\_HS コントローラがエニューメレーションした現在のスピードでのフレームインターバル情報を格納します。

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

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **RLDCTRL** : 再ロード制御

このビットを使用して、ランタイム時に HFIR レジスタの動的再ロードが可能です。

0 : HFIR は動的に再ロードすることができません。

1 : HFIR はランタイム時に動的に再ロードすることができます。

このビットは、初期設定時にプログラムする必要があり、ランタイム時に変更してはなりません。

ビット 15:0 **FRIVL** : USB OTG FS のフレームインターバル

アプリケーションがこのフィールドにプログラムする値は、2つの連続した SOF (FS) またはキープアライブトークン (LS) の間のインターバルを指定します。このフィールドは、必要なフレームインターバルを構成する PHY クロック数を含みます。アプリケーションは、ホストポート制御およびステータスレジスタのポートイネーブルビット (OTG\_HPRT の PENA ビット) がセットされた後でのみ、このレジスタに値を書き込むことができます。値がプログラムされていない場合、コアは、ホスト設定レジスタの FS/LS PHY クロック選択フィールド (OTG\_HCFG の FSLSPCS) で指定された PHY クロックに基づいて値を計算します。初期設定の後には、RLDCTRL ビットがセットされている場合を除いて、このフィールドの値を変更しないでください。そのような場合、FRIVL には各 SOF イベントが再ロードされます。

ビット 15:0 **FRIVL** : USB OTG HS のフレームインターバル

アプリケーションがこのフィールドにプログラムする値は、2つの連続したマイクロ SOF (HS) またはキープアライブトークン (LS) の間のインターバルを指定します。このフィールドは、必要なフレームインターバルを構成する PHY クロック数を含みます。アプリケーションは、ホストポート制御およびステータスレジスタのポートイネーブルビット (OTG\_HPRT の PENA ビット) がセットされた後でのみ、このレジスタに値を書き込むことができます。値がプログラムされていない場合、コアは、ホスト設定レジスタの FS/LS PHY クロック選択フィールド (OTG\_HCFG の FSLSPCS) で指定された PHY クロックに基づいて値を計算します。初期設定の後には、RLDCTRL ビットがセットされている場合を除いて、このフィールドの値を変更しないでください。そのような場合、FRIVL には各 SOF イベントが再ロードされます。

## 37.15.21 OTG ホストフレーム番号/残りフレーム時間 レジスタ (OTG\_HFNUM)

アドレスオフセット : 0x408

リセット値 : 0x0000 3FFF

このレジスタは、現在のフレーム番号を示します。また、現在のフレームの残り時間 (PHY クロック数) も示します。

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

ビット 31:16 **FTREM** : 残りフレーム時間

現在のフレームの残り時間を PHY クロック数で示します。このフィールドは PHY クロックごとにデクリメントされます。値がゼロになると、このフィールドにはフレームインターバルレジスタの値が再ロードされ、新しい SOF が USB に送信されます。

ビット 15:0 **FRNUM** : フレーム番号

このフィールドは USB 上で新しい SOF が送信されるたびにインクリメントされ、0x3FFF に達すると 0 にクリアされます。

## 37.15.22 OTG ホスト周期的送信 FIFO / キューステータスレジスタ (OTG\_HPTXSTS)

アドレスオフセット : 0x410

リセット値 : 0x0008 0100

この読み出し専用レジスタは、周期的 Tx FIFO および周期的送信リクエストキューの空きスペース情報を含みます。

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

ビット 31:24 **PTXQTOP** : 周期的送信リクエストキューの先頭

これは MAC によって処理中の周期的 Tx リクエストキューのエントリを示します。

このレジスタは、デバッグに使用されます。

ビット 31 : 奇数/偶数フレーム

0 : 偶数フレームで送信

1 : 奇数フレームで送信

ビット 30:27チャンネル/エンドポイント番号

ビット 26:25タイプ

00 : IN/OUT

01 : 長さゼロの packets

11 : チャンネルコマンドディセーブル

ビット 24 : 終了 (選択されたチャンネル/エンドポイントの最後のエントリ)

ビット 23:16 **PTXQSAV** : 周期的送信リクエストキューの使用可能スペース

周期的送信リクエストキューで書き込みに使用できる空きロケーションの数を示します。このキューは IN および OUT リクエストの両方を保持します。

00 : 周期的送信リクエストキューは満杯です。

01 : 1 位置が使用可能です。

10 : 2 位置が使用可能です。

bxn : n 位置が使用可能です (0 ≤ n ≤ 8)

その他 : 予約済み

ビット 15:0 **PTXFSAVL** : 周期的送信データ FIFO の使用可能スペース

周期的 Tx FIFO で書き込みに使用できる空きロケーションの数を示します。

値は 32 ビットワード単位です。

0000 : 周期的 Tx FIFO は満杯です。

0001 : 1ワードが使用可能です。

0010 : 2ワードが使用可能です。

bxn : n ワードが使用可能です (0 ≤ n ≤ PTXFD)

その他 : 予約済み

## 37.15.23 OTG ホスト全チャンネル割り込みレジスタ (OTG\_HAINT)

アドレスオフセット : 0x414

リセット値 : 0x0000 000

チャンネル上で重大なイベントが発生すると、ホスト全チャンネル割り込みレジスタは、コア割り込みレジスタのホストチャンネル割り込みビット (OTG\_GINTSTS の HCINT ビット) を使用して、アプリケーションに割り込みをかけます。これを [図 446](#) に示します。チャンネルごとに 1 つの割り込みビットがあります (最大 16 ビット)。このレジスタのビットは、アプリケーションが対応するホストチャンネル x 割り込みレジスタのビットをセット/クリアしたときにセット/クリアされます。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **HAINT** : チャンネル割り込み

チャンネルあたり 1 ビット:チャンネル 0 に対してビット 0、チャンネル 15 に対してビット 15

## 37.15.24 OTG ホスト全チャンネル割り込みマスクレジスタ (OTG\_HAINTMSK)

アドレスオフセット : 0x418

リセット値 : 0x0000 0000

ホスト全チャンネル割り込みマスクレジスタは、ホスト全チャンネル割り込みレジスタとともに働き、チャンネル上でイベントが発生したときに、アプリケーションに割り込みをかけます。チャンネルごとに 1 つの割り込みマスクビットがあります (最大 16 ビット)。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **HAINTM** : チャンネル割り込みマスク

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

チャンネルあたり 1 ビット:チャンネル 0 に対してビット 0、チャンネル 15 に対してビット 15

## 37.15.25 OTG ホストポート制御およびステータスレジスタ (OTG\_HPRT)

アドレスオフセット : 0x440

リセット値 : 0x0000 0000

このレジスタは、ホストモードでのみ使用可能です。現時点では OTG ホストは 1 つのポートのみをサポートしています。

1 つのレジスタは、ポートごとの USB リセット、イネーブル、サスペンド、レジューム、接続ステータス、テストモードなどの USB ポート関連情報を保持します。これを [図 446](#) に示します。このレジスタの rc\_w1 ビットは、コア割り込みレジスタのホストポート割り込みビット (OTG\_GINTSTS の HPRTINT ビット) を使用して、アプリケーションへの割り込みをトリガできます。ポート割り込みの場合、アプリケーションは、このレジスタを読み出して、割り込みの原因となったビットをクリアする必要があります。rc\_w1 ビットの場合、アプリケーションは、このビットに 1 を書き込んで、割り込みをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD		PTCTL
													r	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCTL			PPWR	PLSTS		Res.	PRST	PSUSP	PRES	POC CHNG	POCA	PEN CHNG	PENA	PCDET	PCSTS
rw	rw	rw	rw	r	r		rw	rs	rw	rc_w1	r	rc_w1	rc_w0	rc_w1	r

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:17 **PSPD** : ポートスピード

このポートに接続されたデバイスのスピードを示します。

- 01 : フルスピード
- 10 : ロースピード
- 11 : 予約済み
- 00 : ハイスピード

ビット 16:13 **PTCTL** : ポートテスト制御

アプリケーションがこのフィールドにゼロ以外の値を書き込んで、ポートをテストモードにすると、対応するパターンがポートにシグナリングされます。

- 0000 : テストモードは無効です
- 0001 : Test\_J モード
- 0010 : Test\_K モード
- 0011 : Test\_SE0\_NAK モード
- 0100 : Test\_Packet モード
- 0101 : Test\_Force\_Enable
- その他 : 予約済み

ビット 12 **PPWR** : ポートパワー

アプリケーションは、このフィールドを使用して、このポートへの電力供給を制御し、コアは、過電流条件が発生すると、このビットをクリアします。

- 0 : パワーオフ
- 1 : パワーオン

ビット 11:10 **PLSTS** : ポートラインステータス

USB データラインの現在のロジックレベルを示します。

- ビット 10 : OTG\_FS\_DP のロジックレベル
- ビット 11 : OTG\_FS\_DM のロジックレベル



ビット 9 予約済みであり、リセット値に保持する必要があります。

#### ビット 8 PRST : ポートリセット

アプリケーションがこのビットをセットすると、このポートでリセットシーケンスが開始されます。アプリケーションは、リセットシーケンスの完了後、リセット時間を決めて、このビットをクリアする必要があります。

0 : ポートは非リセット中

1 : ポートはリセット中

アプリケーションは、ポートのリセットが開始されるまで、少なくとも 10 ms の間、このビットをセット状態に保つ必要があります。USB 規格では最大限度が規定されていませんが、アプリケーションは、このビットをクリアする前に、必要最小時間に加えて、さらに 10 ms の間、このビットをセット状態に保つことができます。

ハイスピード : 50 ms

フルスピード/ロースピード : 10 ms

#### ビット 7 PSUSP : ポートサスペンド

アプリケーションは、このビットをセットして、このポートをサスペンドモードにします。コアは、このビットがセットされると SOF の送信のみを停止します。PHY クロックを停止するには、アプリケーションは PHY のサスペンド入力ピンをアサートするポートクロック停止ビットをセットする必要があります。

このビットの読み出し値は、ポートの現在のサスペンドステータスを反映しています。このビットは、リモートウェイクアップ信号の検出後、またはアプリケーションがこのレジスタのポートリセットビットまたはポートレジャービット、またはコア割り込みレジスタのレジャー/リモートウェイクアップ検出割り込みビットまたは切断検出割り込みビット（それぞれ OTG\_GINTSTS の WKUINT または DISCINT ビット）をセットしたときに、コアによってクリアされます。

0 : ポートは非サスペンドモード

1 : ポートはサスペンドモード

#### ビット 6 PRES : ポートレジャー

アプリケーションは、このビットをセットして、ポート上でレジャーシグナリングを駆動します。コアは、アプリケーションがこのビットをクリアするまで、レジャー信号の駆動を続行します。

コアが、コア割り込みレジスタのポートレジャー/リモートウェイクアップ検出割り込みビット (OTG\_GINTSTS の WKUINT ビット) によって示される USB リモートウェイクアップシーケンスを検出した場合、コアは、アプリケーションの介入なしでレジャーシグナリングの駆動を開始し、切断条件を検出すると、このビットをクリアします。このビットの読み出し値は、コアが現在レジャーシグナリングを駆動中であるかどうかを示します。

0 : レジャー信号を駆動していません。

1 : レジャー信号を駆動しています。

LPM が有効で、コアが L1 状態にあるとき、このビットは次のように動作します。

1. アプリケーションは、このビットをセットして、ポート上でレジャーシグナリングを駆動します。

2. コアは OTG\_GLPMPCFG レジスタの BESLTHRS[3:0] フィールドに指定された所定時間まで、レジャー信号の駆動を続行します。

3. コアが、コア割り込みレジスタのポート L1 レジャー/リモート L1 ウェイクアップ検出割り込みビット (OTG\_GINTSTS の WKUINT) によって示される USB リモートウェイクアップシーケンスを検出した場合、コアは、アプリケーションの介入なしでレジャーシグナリングの駆動を開始し、レジャーの終了時にこのビットをクリアします。このビットはコアとアプリケーションの両方でセット/クリアできます。このビットは、ホストにデバイスが 1 つも接続されていない場合でもコアによってクリアされます。

#### ビット 5 POCCHNG : ポート過電流の変化

コアは、このレジスタのポート過電流アクティブビット (ビット 4) に変化があったとき、このビットをセットします。

#### ビット 4 POCA : ポート過電流アクティブ

ポートの過電流条件を示します。

0 : 非過電流状態

1 : 過電流状態

**ビット 3 PENCHNG** : ポート有効化/ディセーブルの変化

コアは、このレジスタのポート有効化ビット 2 のステータスに変化があったとき、このビットをセットします。

**ビット 2 PENA** : ポート有効化

ポートは、リセットシーケンス後、コアによってのみ有効にされ、過電流条件、切断条件、またはアプリケーションがこのビットをクリアすることによって無効にされます。アプリケーションは、レジスタへの書き込みによってこのビットをクリアすることはできません。クリアして、ポートを無効にすることだけができます。このビットは、アプリケーションへの割り込みをトリガしません。

- 0 : ポートを無効化
- 1 : ポートを有効化

**ビット 1 PCDET** : ポート接続の検出

コアは、デバイス接続が検出されると、このビットをセットして、コア割り込みレジスタのホストポート割り込みビット (OTG\_GINTSTS の HPRTINT ビット) を使用して、アプリケーションへの割り込みをトリガします。アプリケーションは、このビットに 1 を書き込むことによって、割り込みをクリアする必要があります。

**ビット 0 PCSTS** : ポート接続ステータス

- 0 : ポートにはデバイスが接続されていません。
- 1 : ポートにデバイスが接続されています。

## 37.15.26 OTG ホストチャネル x 特性レジスタ (OTG\_HCCHARx) (x = 0..15[HS] / 11[FS]、ここで x = Channel\_number)

アドレスオフセット :  $0x500 + (\text{Channel\_number} * 0x20)$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP		LSDEV	Res.
rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPDIR		EPNUM					MPSIZ								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

**ビット 31 CHENA** : チャネル有効化

このフィールドは、アプリケーションによってセットされ OTG ホストによってクリアされます。  
 0 : チャネルは無効です。  
 1 : チャネルは有効です。

**ビット 30 CHDIS** : チャネル無効化

アプリケーションは、チャネルへの転送が完了する前でも、このビットをセットして、そのチャネル上での送信/受信を停止することができます。アプリケーションは、チャネルを無効として扱う前に、チャネル無効割り込みを待つ必要があります。

**ビット 29 ODDFRM** : 奇数フレーム

このフィールドは OTG ホストが奇数フレームで転送を実行しなければならないことを示すために、アプリケーションによってセット (リセット) されます。このフィールドは、周期的 (アイソクロナスおよび割り込み) トランザクションに対してのみ適用されます。  
 0 : 偶数フレーム  
 1 : 奇数フレーム



ビット 28:22 **DAD** : デバイスアドレス

このフィールドは、データソースまたはデータシンクとして機能する特定のデバイスを選択します。

ビット 21:20 **MCNT** : マルチカウント

このフィールドは、この周期的エンドポイントについてフレームあたりで実行されなければならないトランザクションの数をホストに示します。非周期的転送の場合、このフィールドは使用されません。

00 : 予約済みこのフィールドは定義されていない結果をもたらします。

01 : 1 トランザクション

10 : このエンドポイントに対して発行されるべきトランザクションは、フレームあたり 2 つです。

11 : このエンドポイントに対して発行されるべきトランザクションは、フレームあたり 3 つです。

**注 :** このフィールドは、01 以上にセットする必要があります。

ビット 19:18 **EPTYP** : エンドポイントタイプ

選択された転送タイプを示します。

00 : 制御

01 : アイソクロナス

10 : バルク

11 : 割り込み

ビット 17 **LSDEV** : ロースピードデバイス

このフィールドは、このチャンネルがロースピードデバイスと通信中であることを示すために、アプリケーションによってセットされます。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **EPDIR** : エンドポイントの方向

トランザクションが IN か OUT かを示します。

0 : OUT

1 : IN

ビット 14:11 **EPNUM** : エンドポイント番号

データソースまたはデータシンクとして機能しているデバイスのエンドポイント番号を示します。

ビット 10:0 **MPSIZ** : 最大パケットサイズ

関連するエンドポイントの最大パケットサイズを示します。

## 37.15.27 OTG ホストチャネル x スプリット制御レジスタ (OTG\_HCSPLTx) (x = 0..15、ここで x = Channel\_number)

アドレスオフセット : 0x504 + (Channel\_number \* 0x20)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPLIT EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP LSPLT
r/w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XACTPOS		HUBADDR								PRTADDR					
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **SPLITEN** : スプリット有効化

アプリケーションはこのビットをセットして、このチャネルがスプリットトランザクション処理用に有効化されていることを表示します。

ビット 30:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **COMPLSPLT** : 完全なスプリットを実行

アプリケーションはこのビットをセットして OTG ホストによる完全なスプリットトランザクション実行を要求します。

ビット 15:14 **XACTPOS** : トランザクションポジション

このフィールドを使って、各OUTトランザクションと一緒に全て、先頭、中央、または最終のいずれかのペイロードを送信するか否かを決めます。

- 11 : 全このトランザクションの全データペイロード (188 バイト以下)
- 10 : 先頭このトランザクションの先頭データペイロード (188 バイト超)
- 00 : 中央このトランザクションの中央のペイロード (188 バイト超)
- 01 : 最終このトランザクションの最終ペイロード (188 バイト超)

ビット 13:7 **HUBADDR** : ハブアドレス

このフィールドは、トランザクショントランスレータのハブのデバイスアドレスを保持します。

ビット 6:0 **PRTADDR** : ポートアドレス

このフィールドは、受信トランザクショントランスレータのポート番号です。

**注 :** **設定レジスタは USB OTG HS にのみ適用されます。**

## 37.15.28 OTG ホストチャネル x 割り込みレジスタ (OTG\_HCINTx) (x = 0..15[HS] / 11[FS]、ここで x = Channel\_number)

アドレスオフセット : 0x508 + (Channel\_number \* 0x20)

リセット値 : 0x0000 0000

このレジスタは USB および AHB 関連イベントに関するチャネルのステータスを示します。これを [図 446](#) に示します。アプリケーションは、コア割り込みレジスタのホストチャネル割り込みビット (OTG\_GINTSTS の HCINT ビット) がセットされたとき、このレジスタを読み出す必要があります。アプリケーションがこれらのレジスタを読み出すためには、まず、ホスト全チャネル割り込みレジスタ (OTG\_HAINT) を読み出して、ホストチャネル x 割り込みレジスタの正確なチャネル番号を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG\_HAINT および OTG\_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR	FRM OR	BBERR	TXERR	Res.	ACK (確認応答)	NAK	STALL	Res.	CHH	XFRC
					rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

注: **USB OTG FS の設定レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR	FRM OR	BBERR	TXERR	NYET	ACK (確認応答)	NAK	STALL	AHBE RR	CHH	XFRC
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

注: **USB OTG HS の設定レジスタ**

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DTERR** : データトグルエラー

ビット 9 **FRMOR** : フレームオーバーラン

ビット 8 **BBERR** : バブルエラー

ビット 7 **TXERR** : トランザクションエラー

USB 上で以下のエラーの 1 つが発生したことを示します。

CRC チェック失敗

タイムアウト

ビットスタッフエラー

偽の EOP

ビット 6 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 6 **NYET** : USB OTG HS の応答受信準備完了割り込み

ビット 5 **ACK** : ACK 応答受信/送信割り込み

ビット 4 **NAK** : NAK 応答受信割り込み

ビット 3 **STALL** : STALL 応答受信割り込み

ビット 2 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 2 **AHBERR** : USB OTG HS の AHB エラー

このエラーは、AHB 読み出し/書き込み動作中に AHB エラーが発生したときに内部 DMA モードでのみ生成されます。

アプリケーションは対応する DMA チャンネルアドレスレジスタを読み出して、エラーアドレスを取得することができます。

ビット 1 **CHH** : チャンネル停止

USB トランザクションエラーまたはアプリケーションによる無効化リクエストへの応答のいずれかの理由により、転送が異常終了したことを示します。

ビット 0 **XFRC** : 転送完了

転送がエラーなく正常に完了しました。

### 37.15.29 OTG ホストチャンネル x 割り込みマスクレジスタ (OTG\_HCINTMSKx) (x = 0..15[HS] / 11[FS]、ここで x = Channel\_number)

アドレスオフセット : 0x50C + (Channel\_number \* 0x20)

リセット値 : 0x0000 0000

このレジスタは、前のセクションで説明した各チャンネルステータスのマスクを反映します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR M	FRM ORM	BBERR M	TXERR M	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
					rw	rw	rw	rw		rw	rw	rw		rw	rw

注 : **USB OTG FS の設定レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR M	FRM ORM	BBERR M	TXERR M	NYET	ACKM	NAKM	STALLM	AHBE RRM	CHHM	XFRCM
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注 : **USB OTG HS の設定レジスタ**



ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **DTERRM** : データトグルエラーマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 9 **FRMORM** : フレームオーバーランマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 8 **BBERRM** : バブルエラーマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 7 **TXERRM** : トランザクションエラーマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 6 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 6 **NYET** : USB OTG HS の応答受信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 5 **ACKM** : ACK 応答受信/送信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 4 **NAKM** : NAK 応答受信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 3 **STALLM** : STALL 応答受信割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 2 **AHBERR** : USB OTG HS の AHB エラー

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 2 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 1 **CHHM** : チャネル停止マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 0 **XFRM** : 転送完了マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

## 37.15.30 OTG ホストチャネル x 転送サイズレジスタ (OTG\_HCTSIZx) (x = 0..15[HS] / 11[FS]、ここで x = Channel\_number)

アドレスオフセット : 0x510 + (Channel\_number \* 0x20)

リセット値 : 0x0000 0000

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **DPID** : データ PID

アプリケーションは、このフィールドを最初のトランザクションで使用する PID のタイプでプログラムします。ホストは、残りの転送の間、このフィールドを維持します。

00 : DATA0

01 : DATA2

10 : DATA1

11 : セットアップ (制御) / 予約済み[FS]MDATA[HS] (非制御)

ビット 28:19 **PKTCNT** : パケットカウント

このフィールドは、アプリケーションによって、送信 (OUT) または受信 (IN) されるパケットの予想数でプログラムされます。

ホストは OUT/IN パケットの送信または受信が成功するたびに、このカウントをデクリメントします。このカウントがゼロに達すると、アプリケーションに割り込んで、正常完了を示します。

ビット 18:0 **XFRSIZ** : 転送サイズ

OUT の場合、このフィールドは、ホストが転送時に送信するデータバイト数です。

IN の場合、このフィールドは、アプリケーションが転送のために確保したバッファサイズです。アプリケーションは、IN トランザクション (周期的および非周期的) の場合、このフィールドを最大パケットサイズの整数倍でプログラムすることが期待されます。

### 37.15.31 OTG ホストチャネル x DMA アドレスレジスタ (OTG\_HCDMAx) (x = 0..15、ここで x = Channel\_number)

アドレスオフセット : 0x514 + (Channel\_number \* 0x20)

リセット値 : 0x0000 0000

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

ビット 31:0 DMAADDR : DMA アドレス

このフィールドは、外部メモリ内のエンドポイントのデータがフェッチされる、または格納される開始アドレスを保持します。このレジスタは、各 AHB トランザクションごとにインクリメントされます。

**注 :** 設定レジスタは USB OTG HS にのみ適用されます。

### 37.15.32 デバイスモードレジスタ

これらのレジスタは、コアがデバイスモードに変わるたびにプログラムする必要があります。

### 37.15.33 OTG デバイス設定レジスタ (OTG\_DCFG)

アドレスオフセット : 0x800

リセット値 : 0x0220 0000

このレジスタは、電源投入後、または特定の制御コマンドあるいはエニュメレーションの後、コアをデバイスモードに設定します。初期プログラミングの後には、このレジスタを変更しないでください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRAT IM	Res.	Res.	PFIVL		DAD						Res.	NZLSO HSK	DSPD		
rW			rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW

**注 :** USB OTG FS の設定レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PERSCHIVL		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRAT IM	Res.	Res.	PFIVL		DAD						Res.	NZLSO HSK	DSPD		
rW			rW	rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW

**注：** **USB OTG HS の設定レジスタ**

ビット 31:16 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 31:26 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 25:24 **PERSCHIVL** : USB OTG HS の周期的スケジュールインターバル

このフィールドは、周期的 IN エンドポイントデータのフェッチ用に内部 DMA エンジンが割り当てる必要のある時間を指定します。

周期的エンドポイントの数に基づいて、この値を（マイクロ）フレームの 25、50、75 % のいずれかに指定する必要があります。

- アクティブな周期的エンドポイントが存在するとき、内部 DMA エンジンは、フェッチ中の周期的 IN エンドポイントデータに所定の時間を割り当てます。
- アクティブな周期的エンドポイントが存在しない場合、内部 DMA エンジンは非周期的エンドポイントを処理し、このフィールドを無視します。
- （マイクロ）フレーム内で所定時間経過後 DMA は非周期的エンドポイントのフェッチへ切り替わります。

00 : (マイクロ) フレームの25%

01 : (マイクロ) フレームの50%

10 : (マイクロ) フレームの75%

11 : 予約済み

ビット 23:16 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 15 **ERRATIM** : 不規則エラー割り込みマスク

1 : 不規則エラー時のアーリーサスペンド割り込みをマスクします。

0 : 不規則エラー時にアーリーサスペンド割り込みが生成されます。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:11 **PFIVL** : 周期的フレームインターバル

周期的フレーム終了割り込みを使用してアプリケーションに通知しなければならないフレーム内の時間を示します。これを使用して、そのフレームのアイソクロナストラフィックのすべてが完了したかどうかを判断できます。

00 : フレームインターバルの 80%

01 : フレームインターバルの 85%

10 : フレームインターバルの 90%

11 : フレームインターバルの 95%

ビット 10:4 **DAD** : デバイスアドレス

アプリケーションは、各 SetAddress 制御コマンドの後、このフィールドをプログラムする必要があります。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **NZLSOHSK** : 非ゼロ長ステータスOUTハンドシェイク

アプリケーションは、このフィールドを使用して、制御転送のステータスステージの OUT トランザクション中に、コアが非ゼロ長のデータパケットを受信したときに送信するハンドシェイクを選択できます。

1 : 非ゼロ長ステータスの OUT トランザクション時には STALL ハンドシェイクを送信し、受信した OUT パケットをアプリケーションに送信しません。

0 : 受信した OUT パケット（ゼロ長または非ゼロ長）をアプリケーションに送信し、デバイスエンドポイント制御レジスタの NAK および STALL ビットに基づいてハンドシェイクを送信します。

ビット 1:0 **DSPD** : デバイススピード

アプリケーションがコアにエニュメレーションを要求するスピード、またはアプリケーションがサポートできる最高スピードを示します。ただし、実際のバススピードは、チャープシーケンスの完了後に決まり、コアが接続される USB ホストのスピードに基づきます。

- 00 : 予約済み
- 01 : 予約済み
- 10 : 予約済み
- 11 : フルスピード (USB 1.1 トランシーバのクロックは 48 MHz です)

ビット 1:0 **DSPD** : デバイススピード

アプリケーションがコアにエニュメレーションを要求するスピード、またはアプリケーションがサポートできる最高スピードを示します。ただし、実際のバススピードは、チャープシーケンスの完了後に決まり、コアが接続される USB ホストのスピードに基づきます。

- 00 : ハイスピード
- 01 : 予約済み
- 10 : 予約済み
- 11 : フルスピード (USB 1.1 トランシーバのクロックは 48 MHz です)

### 37.15.34 OTG デバイス制御レジスタ (OTG\_DCTL)

アドレスオフセット : 0x804

リセット値 : 0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DS BESL RJCT	Res.	Res.
													rW		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PO PRG DNE	CGO NAK	SGO NAK	CGI NAK	SGI NAK	TCTL			GON STS	GIN STS	SDIS	RWU SIG
				rW	w	w	w	w	rW	rW	rW	r	r	rW	rW

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **DSBESLRJCT** : ディープスリープ BESL 拒否

コアは、プログラムされた BESL 閾値より大きい BESL 値を持つ LPM リクエストを拒否します。NYET 応答は、BESL 閾値より大きい BESL 値を持つ LPM トークンに送信されます。デフォルトでは、ディープスリープ BESL 拒否機能は無効です。

ビット 17:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **POPRGDNE** : パワーオンプログラミング終了

アプリケーションは、このビットを使用して、パワーダウンモードからのウェイクアップ後に、レジスタのプログラミングが終了したことを示します。

ビット 10 **CGONAK** : グローバル OUT NAK のクリア

このフィールドに書き込むと、グローバル OUT NAK がクリアされます。

**ビット 9 SGONAK** : グローバル OUT NAK のセット

このフィールドに書き込むと、グローバル OUT NAK がセットされます。

アプリケーションは、このビットを使用して、すべての OUT エンドポイントに NAK ハンドシェイクを送信します。

アプリケーションがこのビットをセットするのは、コア割り込みレジスタのグローバル OUT NAK 有効ビット (OTG\_GINTSTS の GONAKEFF ビット) がクリアされていることを確認した後でなければなりません。

**ビット 8 CGINAK** : グローバル IN NAK のクリア

このフィールドに書き込むと、グローバル IN NAK がクリアされます。

**ビット 7 SGINAK** : グローバル IN NAK のセット

このフィールドに書き込むと、グローバル非周期的 IN NAK がセットされます。アプリケーションは、このビットを使用して、すべての非周期的 IN エンドポイントに NAK ハンドシェイクを送信します。

アプリケーションがこのビットをセットするのは、コア割り込みレジスタのグローバル IN NAK 有効ビット (OTG\_GINTSTS の GINAKEFF ビット) がクリアされていることを確認した後でなければなりません。

**ビット 6:4 TCTL** : テスト制御

000 : テストモードは無効です

001 : Test\_J モード

010 : Test\_K モード

011 : Test\_SE0\_NAK モード

100 : Test\_Packet モード

101 : Test\_Force\_Enable

その他 : 予約済み

**ビット 3 GONSTS** : グローバル OUT NAK のステータス

0 : ハンドシェイクは、FIFO ステータスと NAK および STALL ビットの設定に基づいて送信されます。

1 : 使用可能なスペースに関係なく RxFIFO にはデータが書き込まれません。SETUP トランザクションを除き、すべてのパケットに NAK ハンドシェイクを送信します。すべてのアイソクロナス OUT パケットがドロップされます。

**ビット 2 GINSTS** : グローバル IN NAK ステータス

0 : ハンドシェイクは、送信 FIFO 内の使用可能なデータに基づいて送信されます。

1 : NAK ハンドシェイクは、送信 FIFO 内の使用可能なデータに関係なく、非周期的 IN エンドポイントに送出されます。

**ビット 1 SDIS** : ソフト切断

アプリケーションは、このビットを使用して USB OTG コアにソフト切断を実行することを伝えます。このビットがセットされている限り、ホストはデバイスが接続されているとは解釈せず、デバイスは USB 上で信号を受信しません。コアは、アプリケーションがこのビットをクリアするまで、切断状態にとどまります。

0 : 通常動作。ソフト切断後にこのビットがクリアされると、コアは USB ホストに対してデバイス接続イベントを生成します。デバイスが再接続されると、USB ホストはデバイスのエnumレーションをリスタートします。

1 : コアは、USB ホストに対して、デバイス切断イベントを生成します。

ビット 0 **RWUSIG** : リモートウェイクアップシグナリング

アプリケーションがこのビットをセットすると、コアはリモートシグナリングを実行して USB ホストをウェイクアップします。コアをサスペンド状態から抜け出させるには、アプリケーションは、このビットをセットする必要があります。USB 2.0 仕様で規定されているように、アプリケーションは、このビットをセットしてから 1 ms から 15 ms 後にクリアする必要があります。

LPM が有効で、コアが L1 (スリープ) 状態にあるとき、アプリケーションがこのビットをセットすると、コアは L1 リモートシグナリングを開始して、USB ホストをウェイクアップします。コアをスリープ状態から抜け出させるには、アプリケーションは、このビットをセットする必要があります。LPM 仕様で規定されているように、このビットはアプリケーションによってセットされてから 50 μs (T<sub>L1DevDrvResume</sub>) 後にハードウェアによって自動的にクリアされます。アプリケーションは、前の LPM トランザクションからの bRemoteWake がゼロのときこのビットをセットしてはいけません (GLPMCFCG レジスタの REMWAKE ビットを参照)。

表 233 は USB ホストがデバイスの切断を検出するためにソフト切断 (SDIS) ビットがセットされなければならない最短時間 (デバイスの状態による) を示します。クロックジッタに対応するには、アプリケーションは指定された最短時間まで遅延を追加することが推奨されます。

表 233. ソフト切断のための最短時間

動作速度	デバイスの状態	最短時間
フルスピード	サスペンド状態	1 ms + 2.5 μs
フルスピード	アイドル	2.5 μs
フルスピード	アイドル状態でもサスペンド状態でもない (トランザクション実行中)	2.5 μs
ハイスピード	アイドル状態でもサスペンド状態でもない (トランザクション実行中)	125 μs

### 37.15.35 OTG デバイスステータスレジスタ (OTG\_DSTS)

アドレスオフセット : 0x808

リセット値 : 0x0000 0010

このレジスタは USB 関連イベントに関するコアのステータスを示します。デバイス全割り込みレジスタ (OTG\_DAIN1) からの割り込み時に読み出されなければなりません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVLNSTS		FNSOF					
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FNSOF								Res.	Res.	Res.	Res.	EERR	ENUMSPD	SUSP STS	
r	r	r	r	r	r	r	r					r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:22 **DEVLNSTS** : デバイスラインステータス

USB データラインの現在のロジックレベルを示します。

ビット [23] : D+ のロジックレベル

ビット [22] : D- のロジックレベル

ビット 21:8 **FNSOF** : 受信 SOF のフレーム番号

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

**ビット 3 EERR** : 不規則エラー

コアは、このビットをセットして、不規則エラーを報告します。  
 不規則エラーが発生すると、OTG\_FS/OTG\_HS コントローラはサスペンド状態に移行し、OTG\_GINTSTS レジスタのアーリーサスペンドビット (OTG\_GINTSTS の ESUSP ビット) でアプリケーションへの割り込みが生成されます。  
 不規則エラーによるアーリーサスペンドがアサートされた場合、アプリケーションは、ソフト切断回復を実行するしかありません。

**ビット 2:1 ENUMSPD** : エニユメレーションされた速度

チャプシーケンスによる速度検出後の OTG\_FS/OTG\_HS コントローラの速度を示します。  
 01 : 予約済み  
 10 : 予約済み  
 11 : フルスピード (PHY クロックは 48 MHz で動作しています)  
 その他 : 予約済み

**ビット 0 SUSPSTS** : サスペンドステータス

デバイスモードでは、このビットは、USB 上でサスペンド状態が検出されている限り、セットされます。USB データライン上で 3 ms の間アクティビティが 1 つもなかった場合、コアはサスペンド状態に入ります。コアは、以下の場合にサスペンド状態から抜け出します。  
 - USB データライン上で何らかのアクティビティがあったとき。  
 - アプリケーションが OTG\_DCTL レジスタのリモートウェイクアップシグナリングビット (OTG\_DCTL の RWUSIG ビット) に書き込みを行ったとき。

### 37.15.36 OTG デバイス IN エンドポイント共通割り込みマスクレジスタ (OTG\_DIEPMSK)

アドレスオフセット : 0x810

リセット値 : 0x0000 0000

このレジスタは、すべてのエンドポイントの各 OTG\_DIEPINTx レジスタとともに機能して、IN エンドポイントごとに 1 つの割り込みを生成します。このレジスタの対応するビットに書き込むことによって OTG\_DIEPINTx レジスタの特定のステータスに対する IN エンドポイント割り込みはマスクできます。ステータスビットはデフォルトでマスクされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAKM	Res.	Res.	Res.	Res.	Res.	Res.	INEPNE M	INEPNM M	ITTXFE MSK	TOM	Res.	EPDM	XFRM
		rw							rw	rw	rw	rw		rw	rw

**注 :** **USB OTG FS の設定レジスタ**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															



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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAKM	Res.	Res.	Res.	BMA	TXFU RM	Res.	INEPNE M	INEPNM M	ITTXFE MSK	TOM	Res.	EPDM	XFRM
		rw				rw	rw		rw	rw	rw	rw		rw	rw

**注:** **USB OTG HS の設定レジスタ**

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **NAKM** : NAK 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 12:7 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 12:10 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 9 **BIM** : USB OTG HS の BNA 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 8 **TXFURM** : USB OTG HS の FIFO アンダーランマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 7 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 6 **INPNEM** : IN エンドポイント NAK 有効マスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 5 **INPNMM** : EP 不一致付きで受信された IN トークンマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 4 **ITTXFEMSK** : Tx FIFO エンプティ時に受信された IN トークンマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 3 **TOM** : タイムアウト状態マスク (非アイソクロナスエンドポイント)

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDM** : エンドポイント無効化割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 0 **XFRM** : 転送完了割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

## 37.15.37 OTG デバイス OUT エンドポイント共通割り込みマスクレジスタ (OTG\_DOEPMSK)

アドレスオフセット : 0x814

リセット値 : 0x0000 0000

このレジスタは、すべてのエンドポイントの各 OTG\_DOEPINTx レジスタとともに機能して、OUT エンドポイントごとに 1 つの割り込みを生成します。このレジスタの対応するビットに書き込むことによって OTG\_DOEPINTx レジスタの特定のステータスに対する OUT エンドポイント割り込みはマスクできます。ステータスビットはデフォルトでマスクされます。

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

**注 :** USB OTG FS の設定レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NYETMSK	Res.	Res.	Res.	Res.	BOIM	TXFURM	Res.	B2BSTUP	Res.	OTEPDM	STUPM	Res.	EPDM	XFRM
	rw					rw	rw		rw		rw	rw		rw	rw

**注 :** USB OTG HS の設定レジスタ

ビット 31:5 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 31:15 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 14 **NYET** : USB OTG HS の NYET 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 9 **BOIM** : USB OTG HS の BNA 割り込みマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 8 **TXFURM** : USB OTG HS の FIFO アンダーランマスク

- 0 : 割り込みはマスクされます。
- 1 : 割り込みはマスクされません。

ビット 6 予約済みであり、USB OTG FS のリセット値に保持する必要があります。

ビット 7 予約済みであり、USB OTG HS のリセット値に保持する必要があります。



- ビット 6 **B2BSTUP** : 連続 SETUP パケット受信マスク制御 OUT エンドポイントにのみ適用されます。このビットは USB OTG HS 専用です。
  - 0 : 割り込みはマスクされます。
  - 1 : 割り込みはマスクされません。
- ビット 4 **OTEPDM** : エンドポイント無効時 OUT トークン受信マスク制御 OUT エンドポイントにのみ適用されません。
  - 0 : 割り込みはマスクされます。
  - 1 : 割り込みはマスクされません。
- ビット 3 **STUPM** : STUPM : SETUP フェーズ終了マスク制御エンドポイントにのみ適用されます。
  - 0 : 割り込みはマスクされます。
  - 1 : 割り込みはマスクされません。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **EPDM** : エンドポイント無効化割り込みマスク
  - 0 : 割り込みはマスクされます。
  - 1 : 割り込みはマスクされません。
- ビット 0 **XFRM** : 転送完了割り込みマスク
  - 0 : 割り込みはマスクされます。
  - 1 : 割り込みはマスクされません。

### 37.15.38 OTG デバイス全エンドポイント割り込みレジスタ (OTG\_DAIN)

アドレスオフセット : 0x818

リセット値 : 0x0000 0000

エンドポイント上で重要なイベントが発生すると、OTG\_DAIN レジスタは、OTG\_GINTSTS レジスタのデバイス OUT エンドポイント割り込みビットまたはデバイス IN エンドポイント割り込みビット (それぞれ、OTG\_GINTSTS の OEPINT または IEPINT ビット) を使用して、アプリケーションに割り込みをかけます。エンドポイントごとに 1 つの割り込みビットがあります (OUT エンドポイントおよび IN エンドポイントに対してそれぞれ最大 16 ビット)。双方向エンドポイントの場合、対応する IN および OUT 割り込みビットが使用されます。このレジスタのビットは、アプリケーションが対応するデバイスエンドポイント x 割り込みレジスタのビット (OTG\_DIEPINTx/OTG\_DOEPINTx) をセット/クリアしたときに、セット/クリアされます。

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

- ビット 31:16 **OEPINT** : OUT エンドポイント割り込みビット
  - OUT エンドポイントチャンネルあたり 1 ビット :
  - OUT エンドポイント 0 に対してビット 16、OUT エンドポイント 3 に対してビット 18
- ビット 15:0 **IEPINT** : IN エンドポイント割り込みビット
  - IN エンドポイントあたり 1 ビット :
  - IN エンドポイント 0 に対してビット 0、IN エンドポイント 3 に対してビット 3

## 37.15.39 OTG 全エンドポイント割り込みマスクレジスタ (OTG\_DAINMSK)

アドレスオフセット : 0x81C

リセット値 : 0x0000 0000

OTG\_DAINMSK レジスタは、デバイスエンドポイント割り込みレジスタとともに機能して、デバイスエンドポイント上にイベントが発生すると、アプリケーションに割り込みます。ただし、その割り込みに対応する OTG\_DAIN レジスタのビットはセットされたままです。

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

ビット 31:16 **OEPM** : OUT EP 割り込みマスクビット

OUT エンドポイント (EP) あたり 1 ビット :  
 OUT EP 0 に対してビット 16、OUT EP 3 に対してビット 18  
 0 : 割り込みはマスクされます。  
 1 : 割り込みはマスクされません。

ビット 15:0 **IEPM** : IN EP 割り込みマスクビット

IN エンドポイントあたり 1 ビット :  
 IN EP 0 に対してビット 0、IN EP 3 に対してビット 3  
 0 : 割り込みはマスクされます。  
 1 : 割り込みはマスクされません。

## 37.15.40 OTG デバイス V<sub>BUS</sub> 放電時間レジスタ (OTG\_DVBUSDIS)

アドレスオフセット : 0x0828

リセット値 : 0x0000 17D7

このレジスタは SRP 中の V<sub>BUS</sub> 放電時間 (V<sub>BUS</sub> パルシング後) を指定します。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **VBUSDT** : デバイス V<sub>BUS</sub> 放電時間

SRP 中の V<sub>BUS</sub> 放電時間 (V<sub>BUS</sub> パルシング後) を指定します。この値は以下のようになります。  
 V<sub>BUS</sub> 放電時間 (PHY クロック / 1024)  
 この値は V<sub>BUS</sub> の負荷によって調整の必要があるかもしれません。



## 37.15.41 OTG デバイス V<sub>BUS</sub> パルシング時間レジスタ (OTG\_DVBUSPULSE)

アドレスオフセット : 0x082C

リセット値 : 0x0000 05B8

このレジスタは SRP 期間中の V<sub>BUS</sub> パルシング時間を指定します。

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DVBUSP** : デバイス V<sub>BUS</sub> パルシング時間

SRP 中の V<sub>BUS</sub> パルシング時間を指定します。この値は以下ようになります。

V<sub>BUS</sub> パルシング時間 (PHY クロック / 1024)

## 37.15.42 OTG デバイス閾値制御レジスタ (OTG\_DTHRCTL)

アドレスオフセット : 0x0830

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ARPEN	Res.	RXTHRLEN									RXTHREN
				rW		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXTHRLEN									ISOTHREN	NONISOHREN
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **ARPEN** : アービタパーキング有効化

このビットは、IN エンドポイントに対する内部DMA アービタパーキングを制御します。閾値が有効化され、かつこのビットに 1 が設定されると、アービタは USB でトークンが受信された IN エンドポイントにパークします。これは、アンダーラン状態を回避するために実行されます。デフォルトではパーキングは有効化されています。

ビット 26 予約済みであり、リセット値に保持する必要があります。

ビット 25 : 17 **RXTHRLEN** : 受信閾値長

このフィールドは、受信閾値サイズをダブルワード単位で指定します。また、このフィールドはコアが AHB で送信を開始できる前に USB で受信するデータ量も指定します。閾値長は、少なくとも 8 ダブルワードある必要があります。RXTHRLEN の推奨値は、設定された AHB バースト長(OTG\_GAHBCFG の HBSTLEN ビット)と同じです。

ビット 16 **RXTHREN** : 受信閾値有効化

このビットをセットすると、コアは受信方向で閾値を有効化します。

ビット 15 : 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:2 **TXTHRLEN** : 送信閾値長

このフィールドは、送信閾値サイズをダブルワード単位で指定します。このフィールドは、コアが USB で送信を開始できる前に、対応するエンドポイント送信 FIFO に入力するデータバイト数を指定します。閾値長は、少なくとも 8 ダブルワードある必要があります。このフィールドは、アイソクロナスと非アイソクロナスの IN エンドポイント閾値を制御します。TXTHRLEN の推奨値は、設定された AHB バースト長 (OTG\_GAHBCFG の HBSTLEN ビット) と同じです。

ビット 1 **ISOTHREN** : ISO IN エンドポイント閾値有効化

このビットをセットすると、コアはアイソクロナス IN エンドポイントで閾値を有効化します。

ビット 0 **NONISOTHREN** : 非アイソクロナス IN エンドポイント閾値有効化

このビットをセットすると、コアは非アイソクロナス IN エンドポイントで閾値を有効化します。

**注:** **設定レジスタは USB OTG HS にのみ適用されます。**

### 37.15.43 OTG デバイス各エンドポイント割り込みレジスタ (OTG\_DEACHINT)

アドレスオフセット : 0x0838

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OEP1INT	Res.													
														r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	IEP1INT	Res.													
														r	

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **OEP1INT** : OUT エンドポイント 1 割り込みビット

ビット 16:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **IEP1INT** : IN エンドポイント 1 割り込みビット

ビット 0 予約済みであり、リセット値に保持する必要があります。

**注:** **設定レジスタは USB OTG HS にのみ適用されます。**

## 37.15.44 OTG デバイス IN エンドポイント FIFO エンプティ割り込みマスクレジスタ (OTG\_DIEPEMPMSK)

アドレスオフセット : 0x834

リセット値 : 0x0000 0000

このレジスタは、IN エンドポイント FIFO エンプティ割り込みの生成 (TXFE\_OTG\_DIEPINTx) を制御するために使用されます。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **INEPTXFEM** : IN EP Tx FIFO エンプティ割り込みマスクビット

これらのビットは OTG\_DIEPINTx に対するマスクビットとして機能します。

IN EP あたり TXFE 割り込み 1 ビット :

IN EP 0 に対してビット 0、IN EP 3 に対してビット 3

0 : 割り込みはマスクされます。

1 : 割り込みはマスクされません。

## 37.15.45 OTG デバイス各エンドポイント割り込みレジスタマスク (OTG\_DEACHINTMSK)

アドレスオフセット : 0x083C

リセット値 : 0x0000 0000

エンドポイント 1 の IN に対して 1 個の割り込みビットが、エンドポイント 1 の OUT に対して 1 個の割り込みビットが、それぞれ存在します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OEP1 INTM	Res.													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	IEP1 NTM	Res.													
														r/w	
														r/w	

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **OEP1INTM** : OUT エンドポイント 1 割り込みマスクビット

ビット 16:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **IEP1INTM** : IN エンドポイント 1 割り込みマスクビット

ビット 0 予約済みであり、リセット値に保持する必要があります。

**注 :** 設定レジスタは USB OTG HS にのみ適用されます。

## 37.15.46 OTG デバイス制御 IN エンドポイント 0 制御レジスタ (OTG\_DIEPCTL0)

アドレスオフセット : 0x900

リセット値 : 0x0000 0000

このセクションでは、USB\_OTG FS の OTG\_DIEPCTL0 レジスタについて説明します。非ゼロ制御エンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAK STS	Res.
rs	rs			w	w	rw	rw	rw	rw	rs		r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
r														rw	rw

### ビット 31 EPENA : エンドポイント有効化

アプリケーションは、このビットをセットして、エンドポイント 0 でデータ送信を開始します。コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- エンドポイント無効化
- 転送完了

### ビット 30 EPDIS : エンドポイント無効化

アプリケーションは、このビットをセットして、エンドポイントへの転送が完了する前でも、そのエンドポイントでのデータ送信を停止することができます。アプリケーションは、エンドポイントを無効として扱う前に、エンドポイント無効割り込みを待つ必要があります。コアは、エンドポイント無効割り込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイント有効化がすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

### ビット 27 SNAK : NAK セット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、エンドポイント上で SETUP パケットが受信された後、そのエンドポイントに対してこのビットをセットすることもできます。

### ビット 26 CNAK : NAK クリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

### ビット 25:22 TXFNUM : Tx FIFO 番号

この値は、IN エンドポイント 0 に割り当てられる FIFO 番号に設定されます。

### ビット 21 STALL : STALL ハンドシェイク

アプリケーションは、このビットのセットのみが可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル INNAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。

ビット 20 予約済みであり、リセット値に保持する必要があります。

### ビット 19:18 EPTYP : エンドポイントタイプ

制御のために "00" にハードコード化されています。



ビット 17 **NAKSTS** : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

このビットがアプリケーションまたはコアによってセットされると、コアは、Tx FIFO に使用可能なデータがある場合でも、データの送信を停止します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **USBAEP** : USB アクティブエンドポイント

このビットは常に 1 にセットされ、制御エンドポイント 0 がすべての設定およびインタフェースにおいて常にアクティブであることを示します。

ビット 14:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。

00 : 64 バイト

01 : 32 バイト

10 : 16 バイト

11 : 8 バイト

**注 :** **設定レジスタは USB OTG FS にのみ適用されます。**

## 37.15.47 OTG デバイスエンドポイント x 制御レジスタ (OTG\_DIEPCTLx) (x = 1..5[FS] / 0..7[HS]、ここで x = Endpoint\_number)

アドレスオフセット :  $0x900 + (\text{Endpoint\_number} * 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、このレジスタを使用して、エンドポイント 0 以外の各論理エンドポイントの動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNACK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EO NUM/DPID
rs	rs	w	w	w	w	rw	rw	rw	rw	rw/rs		rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **EPENA** : エンドポイント有効化

アプリケーションは、このビットをセットして、エンドポイントでのデータ送信を開始します。

コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効
- 転送完了

**ビット 30 EPDIS** : エンドポイント無効化

アプリケーションは、エンドポイントへの転送が完了する前でも、このビットをセットして、そのエンドポイントでのデータ送信/受信を停止することができます。アプリケーションは、エンドポイントを無効として扱う前に、エンドポイント無効割り込みを待つ必要があります。コアは、エンドポイント無効割り込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイント有効化がすでにセットされている場合のみ、このビットをセットしなければなりません。

**ビット 29 SODDFRM** : 奇数フレームの設定

アイソクロナス IN および OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが奇数フレームに設定されます。

**ビット 28 SD0PID** : DATA0 PID の設定

割り込み/バルク IN エンドポイントにのみ適用されます。

このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA0 に設定されます。

**SEVNFIRM** : 偶数フレーム設定

アイソクロナス IN エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが偶数フレームに設定されます。

**ビット 27 SNAK** : NAK セット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、送信完了割り込み時、またはエンドポイントで SETUP パケットが受信された後、OUT エンドポイントに対してこのビットをセットすることもできます。

**ビット 26 CNAK** : NAK クリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

**ビット 25:22 TXFNUM** : Tx FIFO 番号

これらのビットは、このエンドポイントに関連する FIFO 番号を指定します。各アクティブ IN エンドポイントは、別々の FIFO 番号にプログラムされなければなりません。

このフィールドは、IN エンドポイントについてのみ有効です。

**ビット 21 STALL** : STALL ハンドシェイク

非制御の非アイソクロナス IN エンドポイントにのみ適用されます (アクセスタイプは rw です)。

アプリケーションは、このビットをセットして、USB ホストからこのエンドポイントへのすべてのトークンを停止します。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。アプリケーションでのみこのビットをクリアできます。コアはクリアできません。

制御エンドポイントにのみ適用されます (アクセスタイプは rs です)。

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

**ビット 20** 予約済みであり、リセット値に保持する必要があります。**ビット 19:18 EPTYP** : エンドポイントタイプ

これは、この論理エンドポイントがサポートする転送タイプです。

00 : 制御

01 : アイソクロナス

10 : バルク

11 : 割り込み



## ビット 17 **NAKSTS** : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、

非アイソクロナス IN エンドポイントの場合 : コアは、Tx FIFO に使用可能なデータがある場合でも、IN エンドポイントでのデータの送信を停止します。

アイソクロナス IN エンドポイントの場合 : コアは、Tx FIFO に使用可能なデータがある場合でも、ゼロ長のデータパケットを送出します。

このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

## ビット 16 **EONUM** : 偶数/奇数フレーム

アイソクロナス IN エンドポイントにのみ適用されます。

コアがこのエンドポイントにアイソクロナスデータを送受信するフレーム番号を示します。アプリケーションは、このレジスタの SEVNFRM および SODDFRM フィールドを使用して、このエンドポイントに対してアイソクロナスデータを送信/受信する偶数/奇数番号をプログラムする必要があります。

0 : 偶数フレーム

1 : 奇数フレーム

### **DPID** : エンドポイントデータPID

割り込み/バルク IN エンドポイントにのみ適用されます。

このエンドポイント上で受信または送信されるパケットの PID を含みます。アプリケーションは、エンドポイントがアクティブ化された後、このエンドポイントで受信または送信される最初のパケットの PID をプログラムする必要があります。アプリケーションは SC0PID レジスタを使用して、DATA0 または DATA1 PID のいずれかをプログラムします。

0 : DATA0

1 : DATA1

## ビット 15 **USBAEP** : USB アクティブエンドポイント

このエンドポイントが現在の設定およびインタフェースでアクティブかどうかを示します。コアは、USB リセットを検出した後、すべてのエンドポイント (EP 0 を除き) について、このビットをクリアします。SetConfiguration および SetInterface コマンドを受信した後、アプリケーションはそれに従ってエンドポイントをプログラムし、このビットをセットする必要があります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

## ビット 10:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。この値はバイト単位です。

## 37.15.48 OTG デバイス制御 OUT エンドポイント 0 制御レジスタ (OTG\_DOEPCTL0)

アドレスオフセット : 0xB00

リセット値 : 0x0000 8000

このセクションでは、OTG\_DOEPCTL0 レジスタについて説明します。非ゼロ制御エンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAK STS	Res.
w	r			w	w					rs	rw	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
r														r	r

### ビット 31 EPENA : エンドポイント有効化

アプリケーションは、このビットをセットして、エンドポイント 0 上でデータ送信を開始します。コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効化
- 転送完了

### ビット 30 EPDIS : エンドポイント無効化

アプリケーションは、制御 OUT エンドポイント 0 を無効にできません。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

### ビット 27 SNAK : NAK セット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、送信完了割り込み時、または SETUP パケットを受信した後に、このビットをセットすることもできます。

### ビット 26 CNAK : NAK クリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

### ビット 21 STALL : STALL ハンドシェイク

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。NAK ビット、すなわち、グローバル OUT NAK がこのビットとともにセットされた場合は、STALL ビットが優先します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

### ビット 20 SNPM : スヌープモード

このビットは、エンドポイントをスヌープモードに設定します。スヌープモードでは、コアは OUT パケットをアプリケーションメモリに転送する前に、それらの正確さをチェックしません。

### ビット 19:18 EPTYP : エンドポイントタイプ

制御のために 2'b00 にハードコード化されています。



ビット 17 **NAKSTS** : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、Rx FIFO に受信パケットを収容できるスペースがある場合でも、コアはデータの受信を停止します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **USBAEP** : USB アクティブエンドポイント

このビットは常に 1 にセットされ、制御エンドポイント 0 がすべての設定およびインタフェースにおいて常にアクティブであることを示します。

ビット 14:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **MPSIZ** : 最大パケットサイズ

制御 OUT エンドポイント 0 の最大パケットサイズは、制御 IN エンドポイント 0 でプログラムされるサイズと同じです。

00 : 64 バイト

01 : 32 バイト

10 : 16 バイト

11 : 8 バイト

## 37.15.49 OTG デバイスエンドポイント x 制御レジスタ (OTG\_DOEPTLx) (x = 1..5[FS] / 7[HS]、ここで x = Endpoint\_number)

OUT エンドポイントのアドレスオフセット : 0xB00 + (Endpoint\_number \* 0x20)

リセット値 : 0x0000 0000

アプリケーションは、このレジスタを使用して、エンドポイント 0 以外の各論理エンドポイントの動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SD1 PID/ SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w					rw/rs	rw	rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **EPENA** : エンドポイント有効化

IN および OUT エンドポイントに適用されます。

アプリケーションは、このビットをセットして、エンドポイントでのデータ送信を開始します。

コアは、このエンドポイント上で以下の割り込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効化
- 転送完了

**ビット 30 EPDIS** : エンドポイント無効化

アプリケーションは、エンドポイントへの転送が完了する前でも、このビットをセットして、そのエンドポイントでのデータ送信/受信を停止することができます。アプリケーションは、エンドポイントを無効として扱う前に、エンドポイント無効割り込みを待つ必要があります。コアは、エンドポイント無効割り込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイント有効化がすでにセットされている場合のみ、このビットをセットしなければなりません。

**ビット 29 SD1PID** : DATA1 PID の設定

割り込み/バルク IN および OUT エンドポイントにのみ適用されます。このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA1 に設定されます。

**SODDFRM** : 奇数フレームの設定

アイソクロナス IN および OUT エンドポイントにのみ適用されます。このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが奇数フレームに設定されます。

**ビット 28 SD0PID** : DATA0 PID の設定

割り込み/バルク OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA0 に設定されます。

**SEVNFRM** : 偶数フレームの設定

アイソクロナス OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが偶数フレームに設定されます。

**ビット 27 SNAK** : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、転送完了割り込み時、または SETUP パケットを受信した後に、OUT エンドポイントに対してこのビットをセットすることもできます。

**ビット 26 CNAK** : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

**ビット 21 STALL** : STALL のハンドシェイク

非制御の非アイソクロナス OUT エンドポイントにのみ適用されます (アクセスタイプは rw です)。

アプリケーションは、このビットをセットして、USB ホストからこのエンドポイントへのすべてのトークンを停止します。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。アプリケーションでのみこのビットをクリアできます。コアはクリアできません。

制御エンドポイントにのみ適用されます (アクセスタイプは rs です)。

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

**ビット 20 SNPM** : スヌープモード

このビットは、エンドポイントをスヌープモードに設定します。スヌープモードでは、コアは OUT パケットをアプリケーションメモリに転送する前に、それらの正確さをチェックしません。

**ビット 19:18 EPTYP** : エンドポイントタイプ

これは、この論理エンドポイントがサポートする転送タイプです。

00 : 制御

01 : アイソクロナス

10 : バルク

11 : 割り込み



ビット 17 **NAKSTS** : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、

コアは、Rx FIFO に受信パケットを収容するためのスペースがある場合でも、OUT エンドポイントでのデータの受信を停止します。

このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 **EONUM** : 偶数/奇数フレーム

アイソクロナス IN および OUT エンドポイントにのみ適用されます。

コアがこのエンドポイントにアイソクロナスデータを送受信するフレーム番号を示します。アプリケーションは、このレジスタの SEVNFRM および SODDFRM フィールドを使用して、このエンドポイントに対してアイソクロナスデータを送信/受信する偶数/奇数番号をプログラムする必要があります。

0 : 偶数フレーム

1 : 奇数フレーム

**DPID** : エンドポイントデータPID

割り込み/パルク OUT エンドポイントにのみ適用されます。

このエンドポイント上で受信または送信されるパケットの PID を含みます。アプリケーションは、エンドポイントがアクティブ化された後、このエンドポイントで受信または送信される最初のパケットの PID をプログラムする必要があります。アプリケーションは SC0PID レジスタを使用して、DATA0 または DATA1 PID のいずれかをプログラムします。

0 : DATA0

1 : DATA1

ビット 15 **USBAEP** : USB アクティブエンドポイント

このエンドポイントが現在の設定およびインタフェースでアクティブかどうかを示します。コアは、USB リセットを検出した後、すべてのエンドポイント (EP 0 を除き) について、このビットをクリアします。SetConfiguration および SetInterface コマンドを受信した後、アプリケーションはそれに従ってエンドポイントをプログラムし、このビットをセットする必要があります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。この値はバイト単位です。

## 37.15.50 OTG デバイスエンドポイント x 割り込みレジスタ (OTG\_DIEPINTx) (x = 0..5[FS] / 7[HS]、ここで x = Endpoint\_number)

アドレスオフセット :  $0x908 + (\text{Endpoint\_number} * 0x20)$

リセット値 : 0x0000 0080

このレジスタは、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。これを [図 446](#) に示します。アプリケーションは、コア割り込みレジスタの IN エンドポイント割り込みビット (OTG\_GINTSTS の IEPINT ビット) がセットされたとき、このレジスタを読み出さなければなりません。アプリケーションがこのレジスタを読み出すためには、まず、デバイス全エンドポイント割り込み (OTG\_DAINTE) レジスタを読み出して、デバイスのエンドポイント x 割り込みレジスタの正確なエンドポイント番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG\_DAINTE および OTG\_GINTSTS レジスタの対応するビットをクリアする必要があります。

# 参考資料

RM0385

USB On-The-Go フルスピード/ハイスピード (OTG\_FS/OTG\_HS)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TXFE	INEP NE	Res.	ITTXFE	TOC	Res.	EP DISD	XFRC							
								r	rc_w1/ rw		rc_w1	rc_w1		rc_w1	rc_w1

注: USB OTG FS の設定レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAK	BERR	PKTD RPSTS	Res.	BNA	TXFIF OUD RN	TXFE	INEP NE	Res.	ITTXFE	TOC	Res.	EP DISD	XFRC
		rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	r	rc_w1/ rw		rc_w1	rc_w1		rc_w1	rc_w1

注: USB OTG HS の設定レジスタ

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 31:14 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 13 **NAK**: USB OTG HS の NAK 入力

コアは、デバイスが NAK を送受信したとき、この割り込みを生成します。

アイソクロナス IN エンドポイントの場合、Tx FIFO にデータがないためにゼロ長パケットが送信されたとき、この割り込みが生成されます。

ビット 12 **BERR**: バブルエラー割り込み (USB OTG HS)

ビット 11 **PKTDRPSTS**: USB OTG HS のパケットドロップステータス

このビットは、ISOC OUT パケットがドロップされたことをアプリケーションに示します。このビットには

対応するマスクビットがなく、割り込みは生成されません。

ビット 10 予約済みであり、USB OTG HS のリセット値に保持する必要があります。

ビット 9 **BNA**: USB OTG HS のバッファ使用不能割り込み

コアは、ホストビジイまたは DMA 終了のような処理を行うために、コアに対してアクセスしたディスクリプタがレディでないときにこの割り込みを生成します。

ビット 8 **TXFIFOUDRN**: USB OTG HS の送信 FIFO アンダーラン (TxfifoUndrn)

コアはこのエンドポイントの送信 FIFO アンダーラン状態を検出するとこの割り込みを生成します。依存性: この割り込みは、閾値が有効化されている場合にのみ有効です。

ビット 7 **TXFE**: 送信 FIFO エンプティ

この割り込みは、このエンドポイントの Tx FIFO が半エンプティまたは完全エンプティのいずれかであるとき、アサートされます。半エンプティか完全エンプティであるかは、OTG\_GAHBCFG レジスタの Tx FIFO エンプティレベルビット (OTG\_GAHBCFG の TXFELVL ビット) によって判断されます。



DocID026670 Rev 2

1393/1653

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

**ビット 6 INEPNE : IN エンドポイント NAK 有効**

このビットは、アプリケーションが OTG\_DIEPCTLx レジスタの CNAK ビットに書き込むことによって IN エンドポイント NAK をクリアするとクリアできます。

この割り込みは、コアが (アプリケーションまたはコアによって) 設定された NAK ビットをサンプリングしたことを示します。この割り込みは、アプリケーションによってセットされた IN エンドポイント NAK ビットがコアで有効になったことを示します。

この割り込みは、必ずしも NAK ハンドシェイクが USB に送出されることを保証しません。STALL ビットが NAK ビットに優先します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

**ビット 4 ITTXFE : Tx FIFO がエンプティ時に受信された IN トークン**

非アイソクロナス IN エンドポイントにのみ適用されます。

関連する Tx FIFO (周期的/非周期的) がエンプティのときに IN トークンが受信されたことを示します。この割り込みは、IN トークンが受信されたエンドポイント上でアサートされます。

**ビット 3 TOC : タイムアウト条件**

制御 IN エンドポイントにのみ適用されます。

このエンドポイント上の最後の IN トークンに対して、コアが USB 上でタイムアウト条件を検出したことを示します。

ビット 2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 EPDISD : エンドポイント無効化割り込み**

このビットは、エンドポイントがアプリケーションのリクエストによって無効にされたことを示します。

**ビット 0 XFRC : 転送完了割り込み**

このフィールドは、このエンドポイントに対してプログラムされた転送が AHB に加えて USB 上でも完了したことを示します。

## 37.15.51 OTG デバイスエンドポイント x 割り込みレジスタ (OTG\_DOEPINTx) (x = 0..5[FS] / 7[HS]、ここで x = Endpoint\_number)

アドレスオフセット : 0xB08 + (Endpoint\_number \* 0x20)

リセット値 : 0x0000 0080

このレジスタは、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。これを [図 446](#) に示します。アプリケーションは、OTG\_GINTSTS レジスタの OUT エンドポイント割り込みビット (OTG\_GINTSTS の OEPINT ビット) がセットされたとき、このレジスタを読み出さなければなりません。アプリケーションがこのレジスタを読み出すためには、まず、OTG\_DAINTE レジスタを読み出して、OTG\_DOEPINTx レジスタの正確なエンドポイント番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG\_DAINTE および OTG\_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	B2B STUP	Res.	OEP DIS	STUP	Res.	EP DISD	XFRC								
									rc_w1/rw		rc_w1	rc_w1		rc_w1	rc_w1

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

- ビット 6 **B2BSTUP** : 連続 SETUP パケットの受信  
 制御 OUT エンドポイントにのみ適用されます。  
 このビットは、コアがこの特定のエンドポイントで 3 つを超える連続 SETUP パケットを受信したことを示します。
- ビット 5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **OTEPDIS** : エンドポイント無効時 OUT トークン受信  
 制御 OUT エンドポイントにのみ適用されます。  
 エンドポイントがまだ有効になっていないときに OUT トークンを受け取ったことを示します。この割り込みは、OUT トークンが受信されたエンドポイントでアサートされます。
- ビット 3 **STUP** : SETUP フェーズ終了  
 制御 OUT エンドポイントにのみ適用されます。  
 制御エンドポイントの SETUP フェーズが完了し、現在の制御転送のために、これ以上、連続 SETUP パケットは受信されなかったことを示します。この割り込みにより、アプリケーションは、受信した SETUP データパケットをデコードできます。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **EPDISD** : エンドポイント無効化割り込み  
 このビットは、エンドポイントがアプリケーションのリクエストによって無効にされたことを示します。
- ビット 0 **XFRC** : 転送完了割り込み  
 このフィールドは、このエンドポイントに対してプログラムされた転送が AHB に加えて USB 上でも完了したことを示します。

## 37.15.52 OTG デバイス IN エンドポイント 0 転送サイズレジスタ (OTG\_DIEPTSIZ0)

アドレスオフセット : 0x910

リセット値 : 0x0000 0000

アプリケーションは、エンドポイント 0 を有効にする前に、このレジスタを変更する必要があります。デバイス制御エンドポイント 0 制御レジスタのエンドポイント有効化ビット (OTG\_DIEPCTL0 の EPENA) を使用して、エンドポイント 0 が有効にされると、コアは、このレジスタを変更します。コアがエンドポイント有効化ビットをクリアすると、アプリケーションはこのレジスタを読み出すことができず。

非ゼロエンドポイントは、エンドポイント 1~3 のレジスタを使用します。

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

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:19 **PKTCNT** : パケットカウント

エンドポイント 0 のデータ転送サイズを構成する USB パケットの総数を示します。

このフィールドは、パケット (最大サイズまたはショートパケット) が Tx FIFO から読み出されるたびにデクリメントされます。



ビット 18:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **XFRSIZ** : 転送サイズ

エンドポイント 0 の転送サイズをバイト単位で示します。コアは、データの転送サイズを使いきった後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、外部メモリからのパケットが Tx FIFO に書き込まれるたびに、このフィールドをデクリメントします。

### 37.15.53 OTG デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG\_DOEPTSIZ0)

アドレスオフセット : 0xB10

リセット値 : 0x0000 0000

アプリケーションは、エンドポイント 0 を有効にする前に、このレジスタを変更する必要があります。OTG\_DOEPTCTL0 レジスタのエンドポイント有効化ビット (OTG\_DOEPTCTL0 の EPENA ビット) を使用してエンドポイント 0 が有効にされると、コアは、このレジスタを変更します。コアがエンドポイント有効化ビットをクリアすると、アプリケーションはこのレジスタを読み出すことができなくなります。

非ゼロエンドポイントは、エンドポイント 1–5[FS] / 7[HS]のレジスタを使用します。

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **STUPCNT** : SETUP パケットカウント

このフィールドは、エンドポイントが受信できる連続 SETUP データパケット数を指定します。

01 : 1 パケット

10 : 2 パケット

11 : 3 パケット

ビット 28:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **PKTCNT** : パケットカウント

このフィールドは、パケットが Rx FIFO に書き込まれた後、ゼロまでデクリメントされます。

ビット 18:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **XFRSIZ** : 転送サイズ

エンドポイント 0 の転送サイズをバイト単位で示します。コアは、データの転送サイズを使いきった後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、Rx FIFO からパケットが読み出されて、外部メモリに書き込まれるたびに、このフィールドをデクリメントします。

## 37.15.54 OTG デバイス IN エンドポイント x 転送サイズレジスタ (OTG\_DIEPTSIZx) (x = 1..5[FS] / 7[HS]、ここで x = Endpoint\_number)

アドレスオフセット : 0x910 + (Endpoint\_number \* 0x20)

リセット値 : 0x0000 0000

アプリケーションは、エンドポイントを有効にする前に、このレジスタを変更する必要があります。OTG\_DIEPCTLx レジスタのエンドポイントイネーブルビット (OTG\_DIEPCTLx の EPENA ビット) を使用してエンドポイントが有効にされると、コアはこのレジスタを変更します。コアがエンドポイント有効化ビットをクリアすると、アプリケーションはこのレジスタを読み出すことができせん。

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

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **MCNT** : マルチカウント

周期的 IN エンドポイントの場合、このフィールドは、USB でフレームごとに送信されなければならないパケット数を示します。コアは、このフィールドを使用して、アイソクロナス IN エンドポイントのデータ PID を計算します。

01 : 1 パケット

10 : 2 パケット

11 : 3 パケット

ビット 28:19 **PKTCNT** : パケットカウント

このエンドポイントのデータの転送サイズを構成する USB パケットの総数を示します。このフィールドは、パケット (最大サイズまたはショートパケット) が TxFIFO から読み出されるたびにデクリメントされます。

ビット 18:0 **XFRSIZ** : 転送サイズ

このフィールドは、現在のエンドポイントの転送サイズをバイト単位で示します。コアは、データの転送サイズを使い果たした後のみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、外部メモリからのパケットが TxFIFO に書き込まれるたびに、このフィールドをデクリメントします。

## 37.15.55 OTG デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG\_DTXFSTSx) (x = 0..5[FS] / 7[HS]、ここで x = Endpoint\_number)

IN エンドポイントのアドレスオフセット :  $0x918 + (\text{Endpoint\_number} * 0x20)$  この読み出し専用レジスタは、デバイス IN エンドポイント Tx FIFO の空きスペース情報を含みます。

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

31:16 予約済みであり、リセット値に保持する必要があります。

15:0 INEPTFSAV : IN エンドポイント Tx FIFO の使用可能スペース

エンドポイントの Tx FIFO で使用できる空きスペースの量を示します。

値は 32 ビットワード単位です。

0x0 : エンドポイント Tx FIFO は満杯です。

0x1 : 1 ワードが使用可能です。

0x2 : 2 ワードが使用可能です。

0xn : n ワードが使用可能です。

その他 : 予約済み

## 37.15.56 OTG デバイス OUT エンドポイント x 転送サイズレジスタ (OTG\_DOEPTSIZx) (x = 1..5[FS] / 7[HS]、ここで x = Endpoint\_number)

アドレスオフセット :  $0xB10 + (\text{Endpoint\_number} * 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、エンドポイントを有効にする前に、このレジスタを変更する必要があります。OTG\_DOEPTCTLx レジスタのエンドポイントイネーブルビット (OTG\_DOEPTCTLx の EPENA ビット) を使用してエンドポイントが有効にされると、コアはこのレジスタを変更します。コアがエンドポイント有効化ビットをクリアすると、アプリケーションはこのレジスタを読み出すことができなくなります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ		
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **RXDPID** : 受信データ PID

アイソクロナス OUT エンドポイントにのみ適用されます。

これは、このエンドポイントの最後のバケットで受信されたデータの PID です。

00 : DATA0

01 : DATA2

10 : DATA1

11 : MDATA

**STUPCNT** : SETUP パケットカウント

制御 OUT エンドポイントにのみ適用されます。

このフィールドは、エンドポイントが受信できる連続 SETUP データパケット数を指定します。

01 : 1 パケット

10 : 2 パケット

11 : 3 パケット

ビット 28:19 **PKTCNT** : パケットカウント

このエンドポイントのデータの転送サイズを構成する USB パケットの総数を示します。

このフィールドは、パケット（最大サイズまたはショートパケット）が Rx FIFO に書き込まれるたびにデクリメントされます。

ビット 18:0 **XFRSIZ** : 転送サイズ

このフィールドは、現在のエンドポイントの転送サイズをバイト単位で示します。コアは、データの転送サイズを使い果たした後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、RxFIFO からパケットが読み出されて、外部メモリに書き込まれるたびに、このフィールドをデクリメントします。

## 37.15.57 OTG パワーおよびクロックゲーティング制御レジスタ (OTG\_PCGCCTL)

アドレスオフセット : 0xE00

リセット値 : 0x0000 0000

このレジスタは、ホストモードとデバイスモードで使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUSP	PHY SLEEP	ENL1 GTG	PHY SUSP	Res.	Res.	GATE HCLK	STPP CLK							
								r	r	rW	r			rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **SUSP** : ディープスリープ

このビットは、L1 状態時に PHY がディープスリープであることを示します。

ビット 6 **PHYSLEEP** : スリープ状態の PHY

このビットは、PHY がスリープ状態であることを示します。

ビット 5 **ENL1GTG** : スリープクロックゲーティング有効化

このビットがセットされている場合、コアが utmi\_l1\_suspend\_n をアサートできなければ、スリープ状態でコアの内部クロックゲーティングが有効化されます。このビットがセットされていない場合、PHY クロックはスリープ状態ではゲートされません。

ビット 4 **PHYSUSP** : PHY をサスペンド

PHY がサスペンドされたことを示します。このビットは、アプリケーションが STPPCLK ビットをセットした後、PHY がサスペンドされると更新されます。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **GATEHCLK** : HCLK をゲート

USB がサスペンドされるか、セッションが有効でないとき、アプリケーションは、このビットをセットして、AHB スレープとマスタおよびウェイクアップロジック以外のモジュールへの HCLK をゲーティングします。アプリケーションは、USB がレジュームされるか、新しいセッションが開始されると、このビットをクリアします。

ビット 0 **STPPCLK** : PHY クロックを停止

USB がサスペンドされるか、セッションが有効でないか、デバイスが切断されると、アプリケーションは、このビットをセットして、PHY クロックを停止します。アプリケーションは、USB がレジュームされるか、新しいセッションが開始されると、このビットをクリアします。

## 37.15.58 OTG\_FS/OTG\_HS レジスタマップ

次の表に、USB OTG のレジスタマップとリセット値を示します。

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x000	OTG_GOTGCTL	Res.	OTGVER	BSVLD	ASVLD	DBCT	CIDSTS	Res.	Res.	Res.	Res.	EHEN	DHNPEN	HSHPEN	HNPQR	HNGSCS	BVALOVAL	BVALOEN	AVALOVAL	AVALOEN	VBVALOVAL	VBVALOEN	SRQ	SRQSCS											
	リセット値												0	0	0	0	1					0	0	0	0	0	0	0	0	0	0	0	0	0	
0x004	OTG_GOTGINT	Res.	IDCHNG	DBCNE	ADTOCHG	HNGDET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HNSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.												
	リセット値												0	0	0	0								0	0						0				
0x008	OTG_GAHBCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFELVL	TXFELVL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GINTMSK													
	リセット値																							0	0									0	
0x00C	OTG_GUSBCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HNPICAP	SRPCAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.												
	リセット値		0	0																				0	0	0	1							0	0
0x010	OTG_GRSTCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.												
	リセット値	1																																	
0x014	OTG_GINTSTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.												
	リセット値	0	0	0	1	0	1	0	0	0	0		0	0	0	0																			
0x018	OTG_GINTMSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.											
	リセット値	0	0	0	0	0	0	0	0	0		0	0	0	0	0																			
0x01C	OTG_GRXSTSR (ホストモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.												
	リセット値																																		
	OTG_GRXSTSR (デバイスモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.											
	リセット値																																		



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

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x020	OTG_GRXSTSR (ホストモード)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS	DPID	BCNT										CHNUM								
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	OTG_GRXFSIZ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXFD															
	リセット値																	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0x028	OTG_HNPTXFSIZ/OTG_DIEPTXF0	NPTXFD/TX0FD										NPTXFSA/TX0FSA																					
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
0x02C	OTG_HNPTXSTS	Res.	NPTXQTOP					NPTQXSAV					NPTXFSAV																				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0x030	OTG_GI2CCCTL	BSYDNE	RW	Res.	I2CDATSE	I2CDEVADR	Res.	ACK (確認応答)	I2CEN	ADDR					REGADDR					RWDATA													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x038	OTG_GCCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x03C	OTG_CID	PRODUCT_ID																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0x054	OTG_GLPMCFG	Res.	Res.	Res.	ENBESL	LPMR CNTSTS	SNDLPM	LPM RCNT	LPMCHIDX	L1RSMOK	SLPSTS	LPM RSP	L1DSEN	BESLTHRS	L1SSEN	REMWAKE	BESL	LPMACK	LPMEN														
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0x100	OTG_HPTXFSIZ	PTXFSIZ										PTXSA																					
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
0x104	OTG_DIEPTXF1	INEPTXFD										INEPTXSA																					
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
0x108	OTG_DIEPTXF2	INEPTXFD										INEPTXSA																					
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
0x204	OTG_DIEPTXF5	INEPTXFD										INEPTXSA																					
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
...	...	...																															
...	...	...																															
...	...	...																															



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

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x244	OTG_DIEPTXF7	INEPTXFD																INEPTXSA																
	リセット値	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0x400	OTG_HCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLSPCS
	リセット値																																0	0
0x404	OTG_HFIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLDCTRL	FRIVL																
	リセット値																0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0
0x408	OTG_HFNUM	FTREM																FRNUM																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x410	OTG_HPTXSTS	PTXQTOP								PTXQSAV								PTXFSAVL																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
0x414	OTG_HAINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINT																	
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x418	OTG_HAINTMSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINTM																	
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x440	OTG_HPRT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x504	OTG_HCSPLT0	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0																																
0x500	OTG_HCCHAR0	CHENA	CHDIS	ODDFRM	DAD								MCNT	EPTYP	LSDEV	Res.	EPDIR	EPNUM				MPSIZ												
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x508	OTG_HCINT0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x510	OTG_HCTSIZ0	Res.	DPID	PKTCNT												XFRSIZ																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50C	OTG_HCINTMSK0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x514	OTG_HCDMA0	DMAADDR																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



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

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x520	OTG_HCCHAR1	CHENA	CHDIS	ODDFRM	DAD						MCNT			EPTYP			LSDEV			EPDIR			EPNUM						MPSIZ						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x528	OTG_HCINT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK (確認応)	NAK	STALL	Res.	CHH	XFRM		
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0	
0x52C	OTG_HCINTMSK1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMORM	BBERR	TXERR	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRM		
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0	
0x530	OTG_HCTSIZ1	Res.	DPID	PKTCNT											XFRSIZ																				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0x660	OTG_HCCHAR11	CHENA	CHDIS	ODDFRM	DAD						MCNT			EPTYP			LSDEV			EPDIR			EPNUM						MPSIZ						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0x66C	OTG_HCINTMSK11	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMORM	BBERR	TXERR	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRM		
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0x670	OTG_HCTSIZ11	Res.	DPID	PKTCNT											XFRSIZ																				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0x6E0	OTG_HCCHAR15	CHENA	CHDIS	ODDFRM	DAD						MCNT			EPTYP			LSDEV			EPDIR			EPNUM						MPSIZ						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.



表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x6E4	OTG_HCSPLT15	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACT POS	HUBADDR						PRTADDR								
	リセット値	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...																																
0x6EC	OTG_HCINTMSK15	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	ResNYET	ACKM	NAKM	STALLM	Res.	CHHM	XFCRM
	リセット値																							0	0	0	0	0	0	0	0	0	0
...	...																																
0x6F0	OTG_HCTSIZ15	Res.	DPID	PKTCNT												XFRSIZ																	
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...																																
0x6F4	OTG_HCDMA15	DMAADDR																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...																																
0x728	OTG_HCINT11	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK (確認応)	NAK	STALL	Res.	CHH	XFRC
	リセット値																							0	0	0	0	0	0	0	0	0	0
...	...																																
0x7A8	OTG_HCINT15	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK (確認応)	NAK	STALL	Res.	CHH	XFRC
	リセット値																							0	0	0	0	0	0	0	0	0	0
...	...																																
0x800	OTG_DCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRATIM	Res.	Res.	Res.	PFVIL	DAD						Res.	NZLSOHSK	DSPD		
	リセット値																	0				0	0	0	0	0	0	0	0	0	0	0	0



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

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
0x804	OTG_DCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSBESLURJCT	Res.	POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL				GONSTS	GINSTS	SDIS	RWUSIG													
	リセット値														0								0	0	0	0	0	0	0	0	0	0	0	1	0							
0x808	OTG_DSTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV LN STS	FNSOF																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EERR	ENUMSPD	SUSPSTS
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x810	OTG_DIEPMSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							
	リセット値																						0																			
0x814	OTG_DOEPMSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x818	OTG_DAIN	OEPINT										IEPINT																														
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x81C	OTG_DAINMSK	OEPM										IEPM																														
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x828	OTG_DVBUSDIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x82C	OTG_DVB_USPULSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x830	OTG_DTHRCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x834	OTG_DIE_PEMPSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x838	OTG_DEACHINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x83C	OTG_DEACHINT_MSK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
	リセット値																																									
0x900	OTG_DIEPCTL0	EPENA	EPDIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.														
	リセット値	0	0																																							





表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x9A8	OTG_DIEPINT5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDIS	XFRC	
	リセット値																									1	0	0	0	0	0	0	0	0
...	...																																	
0x9B8	OTG_DTXFSTS5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
...	...																																	
0x9E0	OTG_DIEPCTL7	EPENA	EPDIS	SODDFRM	SDOPIID/SEVNFIRM	SNAK	CNAK	TXFNUM	STALL	Res.	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...																																	
0x9B0	OTG_DIEPTSIZ5	Res.	MCNT	PKTCNT										XFRSIZ																				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...																																	
0x9E8	OTG_DIEPINT7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDIS	XFRC	
	リセット値																									1	0	0	0	0	0	0	0	0
...	...																																	
0x9F0	OTG_DIEPTSIZ7	Res.	MCNT	PKTCNT										XFRSIZ																				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...																																	
0x9F8	OTG_FS_DTXFSTS7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																		0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
...	...																																	
0xB00	OTG_DOEPCCTL0	EPENA	EPDIS	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	Res.	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ										
	リセット値	0	0		0	0						0	0	0	0	0	1																0	0
...	...																																	
0xB08	OTG_DOEPINT0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	予約済み	B2BSTUP	Res.	OTEPDIS	STUP	Res.	EPDIS	XFRC	
	リセット値																									0	0	0	0	0	0	0	0	0
...	...																																	



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

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0xB10	OTG_DOEPTSIZ0	Res.	STUPCNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ									
	リセット値		0	0										0													0	0	0	0	0	0	0			
0xB20	OTG_DOEPCCTL1	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	MPSIZ																
	リセット値	0	0	0	0	0	0						0	0	0	0	0	0	0								0	0	0	0	0	0	0	0		
0xB28	OTG_DOEPINT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	予約済み	B2BSTUP	Res.	OTEPDIS	STUP	Res.	EPDISD	XFRC				
	リセット値																								0	0	0	0	0	0	0	0	0			
0xB30	OTG_DOEPTSIZ1	Res.	RXDPID/ STUPCNT	PKTCNT										XFRSIZ																						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0xB50	OTG_DOEPTSIZ2	Res.	RXDPID/ STUPCNT	PKTCNT										XFRSIZ																						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...			
0xBA0	OTG_DOEPCCTL5	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	MPSIZ																
	リセット値	0	0	0	0	0	0						0	0	0	0	0	0	0								0	0	0	0	0	0	0	0		
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...			
0xBE0	OTG_DOEPCCTL7	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	MPSIZ																
	リセット値	0	0	0	0	0	0						0	0	0	0	0	0	0								0	0	0	0	0	0	0	0		
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...			
0xBA8	OTG_DOEPINT5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	予約済み	B2BSTUP	Res.	OTEPDIS	STUP	Res.	EPDISD	XFRC				
	リセット値																								0	0	0	0	0	0	0	0	0			



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

表 234. OTG\_FS/OTG\_HS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	...	...																															
0xBB0	OTG_ DOEPTSIZ5	Res.	Res.	RXDPID/ STUPCNT	PKTCNT												XFRSIZ																
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...																															
0xBE8	OTG_ DOEPT7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	予約済み	B2BSTUP	Res.	OTEPDIS	STUP	Res.	EPDISD	XFRC
	リセット値																									0	0	0	0	0	0	0	0
...	...	...																															
0xBF0	OTG_ DOEPTSIZ7	Res.	Res.	RXDPID/ STUPCNT	PKTCNT												XFRSIZ																
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xE00	OTG_ PCGCCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHYSLEEP	ENL1GTG	PHYSUSP	Res.	Res.	GATEHCLK	STPPCLK	
	リセット値																								0	0	0	0			0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

## 37.16 OTG\_FS/OTG\_HS プログラミングモデル

### 37.16.1 コアの初期化

アプリケーションはコア初期化シーケンスを実行する必要があります。パワーアップ中にケーブルが接続された場合、OTG\_GINTSTS レジスタの現在の動作モードビット (OTG\_GINTSTS の CMOD ビット) はモードを反映します。OTG\_FS/OTG\_HS コントローラは、A プラグが接続されたときにはホストモードに、B プラグが接続されたときにはデバイスモードに入ります。

このセクションでは、電源投入後の OTG\_FS/OTG\_HS コントローラの初期化について説明します。アプリケーションは、ホストモードかデバイスモードかに関係なく、初期化シーケンスに従う必要があります。コアのすべてのグローバルレジスタは、コアの設定に従って以下の手順で初期化されます。

1. OTG\_GAHBCFG レジスタの以下のフィールドをプログラムします。
  - グローバル割り込みマスクビット GINTMSK = 1
  - Rx FIFO 非エンプティ (OTG\_GINTSTS の RXFLVL ビット)
  - 周期的 Tx FIFO エンプティレベル
2. OTG\_GUSBCFG レジスタの以下のフィールドをプログラムします。
  - HNP 対応ビット
  - SRP 対応ビット
  - OTG\_FS/OTG\_HS タイムアウト較正フィールド
  - USB ターンアラウンド時間フィールド
3. ソフトウェアは、OTG\_GINTMSK レジスタの以下のビットのマスクを解除する必要があります。  
OTG 割り込みマスク  
モード不一致割り込みマスク
4. ソフトウェアは、OTG\_GINTSTS レジスタの CMOD ビットを読み出すことによって、OTG\_FS/OTG\_HS コントローラがホストモードで動作中か、デバイスモードで動作中かを判別することができます。

## 37.16.2 ホストの初期化

コアをホストとして初期化するには、アプリケーションは、以下の手順を実行しなければなりません。

1. OTG\_GINTMSK レジスタの HPRTINT ビットをプログラムして、マスクを解除します。
2. OTG\_HCFG レジスタをプログラムして、フルスピードホストを選択します。
3. OTG\_HPRT レジスタの PPWR ビットを 1 にプログラムします。これによって USB 上で  $V_{BUS}$  が駆動されます。
4. OTG\_HPRT0 の PCDET 割り込みを待ちます。これは、デバイスがポートに接続されていることを示します。
5. OTG\_HPRT レジスタの PRST ビットを 1 にプログラムします。これによってリセットプロセスが開始されます。
6. リセットプロセスが完了するまで、少なくとも 10 ms 待ちます。
7. OTG\_HPRT レジスタの PRST ビットを 0 にプログラムします。
8. OTG\_HPRT レジスタの PENCHNG 割り込みを待ちます。
9. OTG\_HPRT レジスタの PSPD ビットを読み出して、エニューメレーションされたスピードを取得します。
10. HFIR レジスタを、選択された PHY クロック 1 に対応する値でプログラムします。
11. OTG\_HCFG レジスタの FLSPPCS フィールドを、手順 9 で検出したデバイスの速度に従ってプログラムします。FLSPPCS が変更されている場合は、ポートのリセットを行う必要があります。
12. OTG\_GRXFSIZ レジスタをプログラムして、受信 FIFO のサイズを選択します。
13. OTG\_HNPTXFSIZ レジスタをプログラムして、非周期的トランザクションの非周期的送信 FIFO のサイズと開始アドレスを選択します。
14. OTG\_HPTXFSIZ レジスタをプログラムして、周期的トランザクションの周期的送信 FIFO のサイズと開始アドレスを選択します。

デバイスと通信するには、システムソフトウェアは、少なくとも 1 つのチャネルを初期化し、有効にする必要があります。

## 37.16.3 デバイスの初期化

アプリケーションは、パワーアップ時、またはホストモードからデバイスモードへの変更後、以下の手順を実行して、コアをデバイスとして初期化する必要があります。

1. OTG\_DCFG レジスタの以下のフィールドをプログラムします。
  - デバイススピード
  - 非ゼロ長ステータスOUTハンドシェイク
2. OTG\_GINTMSK レジスタをプログラムして、次の割り込みのマスクを解除します。
  - USB リセット
  - エニューメレーション終了
  - アーリーサスペンド
  - USB サスペンド
  - SOF
3. OTG\_GCCFG レジスタの VBUSSEN ビットをプログラムして、B デバイスモードでの  $V_{BUS}$  センシングを有効にし、DP ライン上でプルアップ抵抗の両端に 5 V を供給します。
4. OTG\_GINTSTS の USBRST 割り込みを待ちます。これは、この割り込みの受信時、約 10ms 続くリセットが USB で検出されたことを示します。

OTG\_GINTSTS の ENUMDNE 割り込みを待ちます。この割り込みは、USB 上でリセットが終了したことを示します。この割り込みを受信したとき、アプリケーションは、OTG\_DSTS レジスタを読み出して、エニユメレーション速度を判別し、[エニユメレーション完了時のエンドポイント初期化 \(1445 ページ\)](#) にリストされている手順を実行する必要があります。

この時点で、デバイスは、SOF パケットを受け入れて、制御エンドポイント 0 で制御転送を実行する準備ができたことになります。

## 37.16.4 DMA モード

OTG ホストは AHB マスタインタフェースを使って、送信パケットデータ (AHB から USB へ) をフェッチし、データ更新 (USB から AHB へ) を受信します。AHB マスタはプログラムされた DMA アドレス (ホストモードでの HCDMAx レジスタとペリフェラルモードでの DIEPDMAx/DOEPDMAx レジスタ) を使って、データバッファをアクセスします。

## 37.16.5 ホストプログラミングモデル

### チャネルの初期化

アプリケーションが接続されたデバイスと通信するためには、1 つまたは複数のチャネルを初期化する必要があります。チャネルを初期化して、有効にするには、アプリケーションは、以下の手順を実行する必要があります。

1. OTG\_GINTMSK レジスタをプログラムして、以下の割り込みのマスクを解除します。
2. チャネル割り込み
  - OUT トランザクションの非周期的送信 FIFO エンプティ割り込み (パケットカウントフィールドが複数でプログラムされたパイプライントランザクションレベルで動作する場合に適用可能)
  - OUT トランザクションの非周期的送信 FIFO 半エンプティ割り込み (パケットカウントフィールドが複数でプログラムされたパイプライントランザクションレベルで動作する場合に適用可能)
3. OTG\_HAINTMSK レジスタをプログラムして、選択されたチャネルの割り込みのマスクを解除します。
4. OTG\_HCINTMSK レジスタをプログラムして、ホストチャネル割り込みレジスタで指定された、所定のトランザクション関連割り込みのマスクを解除します。
5. 選択されたチャネルの OTG\_HCTSIZx レジスタを、合計転送サイズ (バイト数) およびショートパケットを含む予想されるパケット数でプログラムします。アプリケーションは、PID フィールドを初期データ PID (最初の OUT トランザクションで使用される PID、または最初の IN トランザクションから期待される PID) でプログラムする必要があります。
6. 選択されたチャネルの OTG\_HCCHARx レジスタを、タイプ、スピード、方向など、デバイスのエンドポイントの特性でプログラムします。(チャネルは、アプリケーションがパケットを送信または受信する準備ができていない場合のみ、チャネル有効化ビットを 1 にセットすることで有効にできます)。
7. OTG\_HS\_HCSPLTx レジスタ内の選択したチャネルにハブアドレスとポートアドレスをプログラムします (スプリットトランザクションの場合)。
8. HCDMAx レジスタ内の選択したチャネルにバッファ開始アドレスをプログラムします (DMA トランザクションのみ)。

## チャンネルの停止

アプリケーションは、OTG\_HCCHARx レジスタの CHDIS および CHENA ビットを 1 にセットすることによって、任意のチャンネルを無効にできます。これにより、OTG\_FS/OTG\_HS ホストは、ポストされたリクエスト（ある場合）を一掃して、チャンネル停止割り込みを生成できます。アプリケーションは、チャンネルを他のトランザクションに再割り当てする前に、OTG\_HCINTx 内の CHH 割り込みを待つ必要があります。OTG\_FS/OTG\_HS ホストは、USB 上ですでに転送が開始しているトランザクションに対しては割り込みを行いません。

DMA モード動作でチャンネルを無効化するとき、アプリケーションはリクエストキュー内のスペースをチェックする必要はありません。OTG\_HS ホストは、調停時の無効化されたチャンネルの回転に関するディセーブルリクエストを書き込むスペースがあるかをチェックします。

その間、HCCHARx の CHDIS ビットに 1 が設定されたとき、要求された全リクエストはリクエストキューからドロップされます。

チャンネルを無効にする前に、アプリケーションは、非周期的リクエストキュー（非周期的チャンネルを無効にするとき）、または周期的リクエストキュー（周期的チャンネルを無効にするとき）に使用可能な空きスペースが少なくとも 1 つ以上存在することを確認する必要があります。アプリケーションは、リクエストキューがフル状態のときには（チャンネルを無効化する前）、OTG\_HCCHARx レジスタの CHDIS ビットを 1 にセットし CHENA ビットを 0 にクリアするだけで、ポストされたリクエストを一掃できます。

アプリケーションは、以下のいずれかの条件のとき、チャンネルを無効にすることが期待されます。

1. IN または OUT チャンネルで、OTG\_HCINTx レジスタの STALL、TXERR、BBERR、または DTERR 割り込みが受信されたとき。アプリケーションは、停止割り込みを受信する前に、同じチャンネルについての他の割り込み（DTERR、Nak、Data、TXERR）を受信できなければなりません。
2. 非周期的 IN 転送または広帯域割り込み IN 転送中に OTG\_HS\_HCINTx の XFRC 割り込みが受信されたとき。
3. OTG\_GINTSTS レジスタの DISCINT（デバイス切断）割り込みを受信したとき。（アプリケーションは、すべての有効なチャンネルを無効にすることが期待されます）。
4. アプリケーションが正常完了の前に転送を中断したとき。

## PING プロトコル

OTG\_HS ホストがハイスピードで動作する場合、ハイスピードバルクエンドポイントまたは制御（データおよびステータスステージ）OUT エンドポイントと交信中に、アプリケーションは PING プロトコルを開始する必要があります。アプリケーションは NAK/NYET/TXERR 割り込みを受信したとき、PING プロトコルを開始する必要があります。HS\_OTG ホストが上記応答の 1 つを受信した場合、特定のエンドポイントに対するトランザクションを継続しないで、（リクエストキューから）ポストまたはフェッチした全 OUT リクエストをドロップさせ、（送信 FIFO から）対応するデータを一掃させます。これはスレーブモードでのみ有効です。スレーブモードでは、アプリケーションは、チャンネルを有効化する前に OTG\_HS\_HCTSIZx 内の DOPING ビットをセットするか、またはチャンネルが有効化済みの場合 OTG\_HS\_HCTSIZx レジスタの DOPING ビットにセットを書き込むことにより、PING トークンを送信することができます。これにより、OTG\_HS ホストが PING リクエストエントリをリクエストキューに書き込めるようになります。アプリケーションは、PING トークンに対する応答（NAK、ACK、または TXERR 割り込み）を待った後に、トランザクションを継続するか、または別の PING トークンを送信する必要があります。アプリケーションは、リクエストされた PING の OUT エンドポイントからの ACK を受信した後のみデータトランザクションを継続することができます。DMA モード動作では、アプリケーションは、バルク/制御 OUT の場合 NAK/NYET 応答に対して OTG\_HS\_HCTSIZx 内の DOPING ビットをセットする必要はありません。OTG\_HS ホストは OTG\_HS\_HCTSIZx 内の DOPING ビットを自動的にセットし、バルク/制御 OUT に対して PING トークンを発行します。OTG\_HS ホストは、ACK を受信するまで PING トークンの送信を続け、自動的にデータトランザクションに切り替わります。

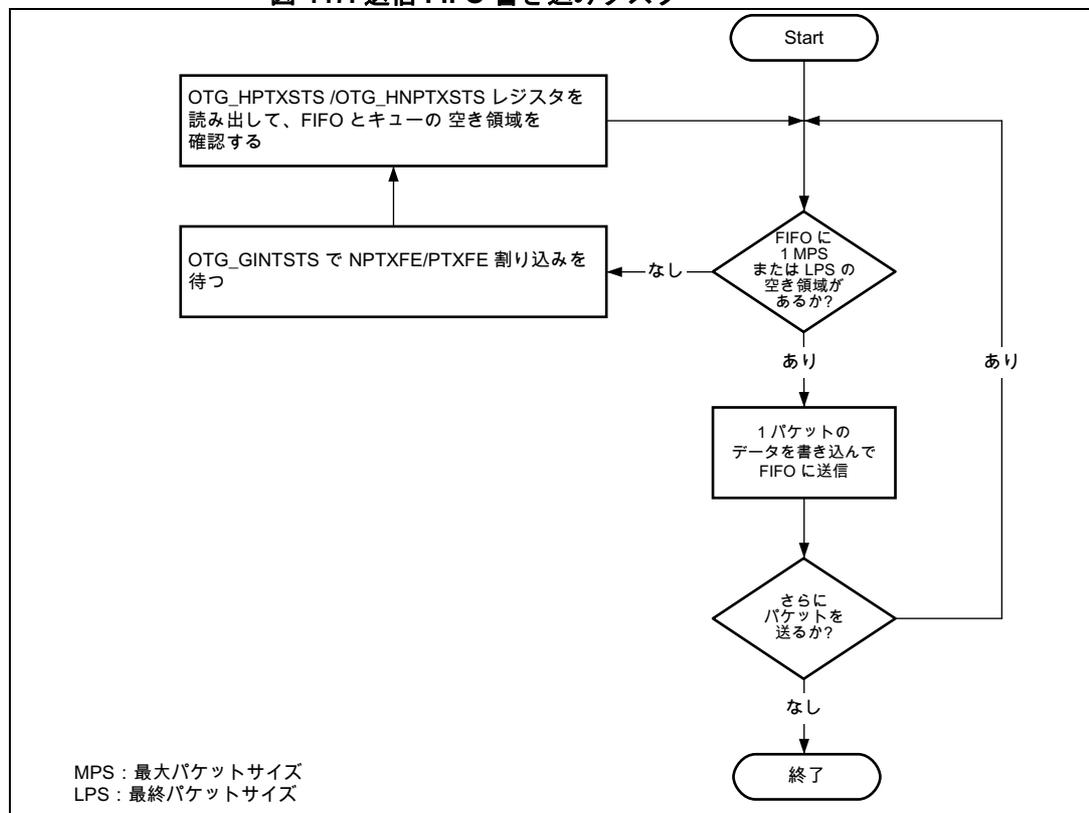
## 動作モデル

アプリケーションが接続されたデバイスと通信するためには、チャンネルを初期化する必要があります。このセクションでは、さまざまなタイプの USB トランザクションについて、実行しなければならない動作のシーケンスについて説明します。

### ● 送信 FIFO への書き込み

OTG\_FS/OTG\_HS ホストは、パケットの最後のダブルワードの書き込みとともに、周期的/非周期的リクエストキューにエントリ (OUT リクエスト) を自動的に書き込みます。アプリケーションは、送信FIFO への書き込みを開始する前に、周期的/非周期的リクエストキューに少なくとも 1 つ以上の空きスペースがあることを確認しなければなりません。アプリケーションは送信 FIFO に必ずダブルワード単位で書き込みを行う必要があります。パケットサイズがダブルワード 単位で揃えられていない場合、アプリケーションはパディングを使用する必要があります。OTG\_FS/OTG\_HS ホストは、プログラムされた最大パケットサイズと転送サイズに基づいて、実際のパケットサイズを決定します。

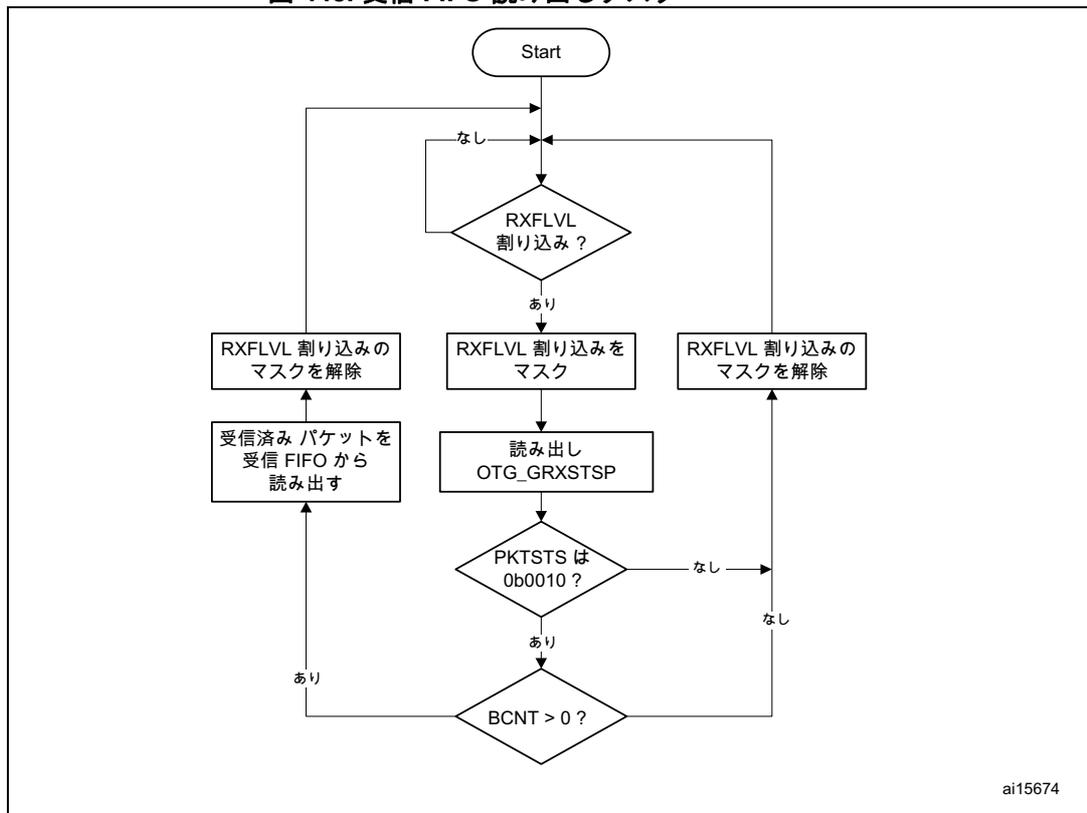
図 447. 送信 FIFO 書き込みタスク



### ● 受信 FIFO の読み出し

アプリケーションは、IN データパケット (bx0010) 以外のすべてのパケットステータスを無視しなければなりません。

図 448. 受信 FIFO 読み出しタスク



● バルクおよび制御 OUT/SETUP トランザクション

典型的なバルクまたは制御 OUT/SETUP パイプライントランザクションレベルの動作を 図 449 に示します。チャンネル 1 (ch\_1) に注目してください。2 つのバルク OUT パケットが転送されます。制御 SETUP トランザクションは同じように動作しますが、1 パケットのみです。ここでは、以下のように仮定されています。

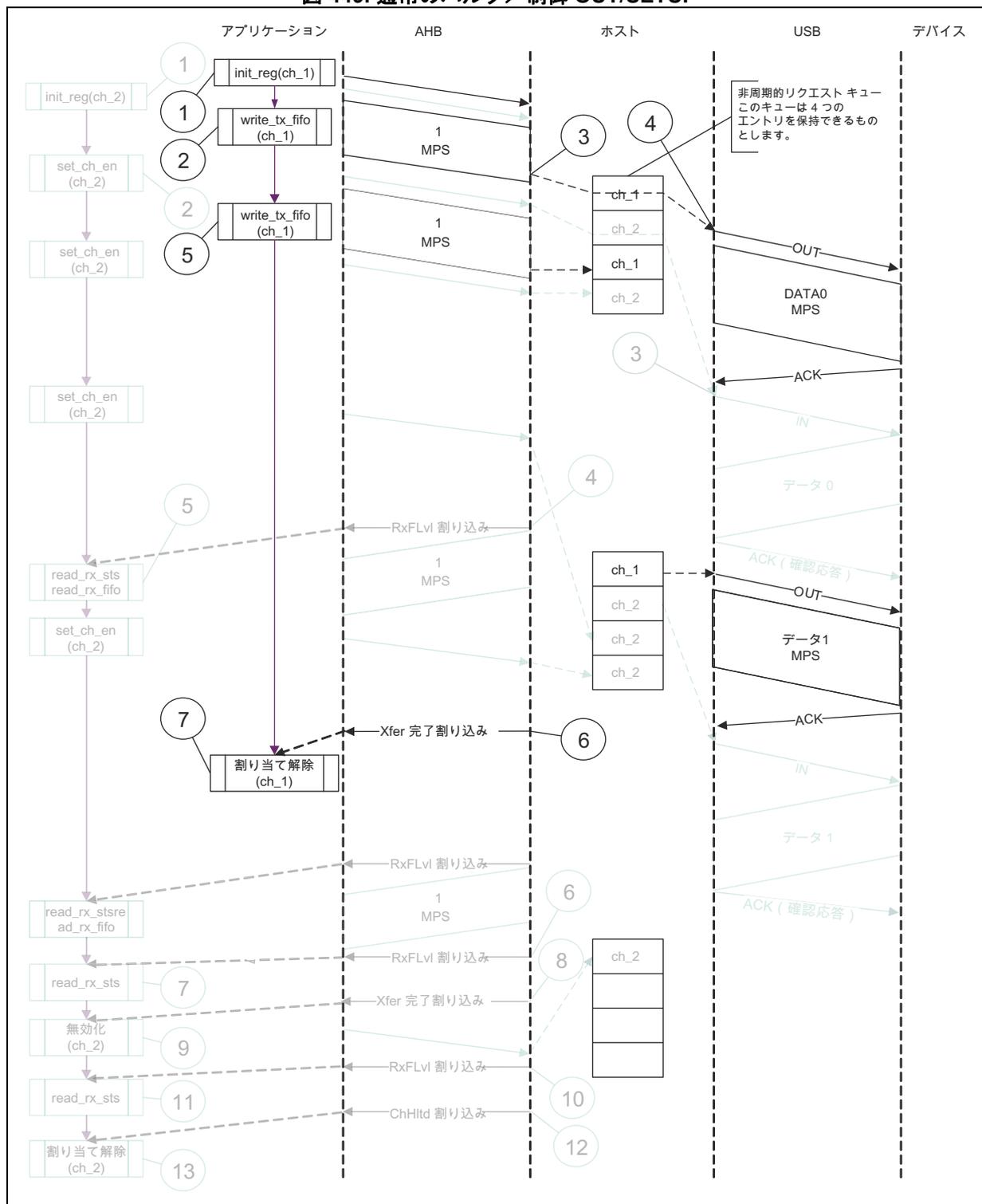
- アプリケーションは、最大パケットサイズの 2 つのパケットを送信しようとしています (転送サイズ = 1,024 バイト)。
- 非周期的送信 FIFO は 2 つのパケットを保持することができます (1 KB - HS の場合、128 バイト - FS の場合)。
- 非周期的リクエストキューの深さ = 4。

● 通常のバルクおよび制御 OUT/SETUP 動作

動作シーケンス (チャンネル 1) は、次のとおりです。

1. チャンネル 1 を初期化します。
2. チャンネル 1 の最初のパケットを書き込みます。
3. 最後のワードの書き込みと同時に、コアは、非周期的リクエストキューにエントリを書き込みます。
4. 非周期的キューがエンプティでなくなるとすぐに、コアは、現在のフレームで OUT トークンの送信を試みます。
5. チャンネル 1 の 2 番目 (最後) のパケットを書き込みます。
6. コアは、最後のトランザクションが正常に完了すると、XFRC 割り込みを生成します。
7. XFRC 割り込みに応答して、他の転送のためにチャンネル割り当てを解除します。
8. 非 ACK 応答の取り扱い

図 449. 通常のパルク/制御 OUT/SETUP



バルクおよび制御 OUT/SETUP トランザクションのためのチャンネル固有の割り込みサービスルーチンを、以下のコードサンプルで示します。

- **バルク/制御 OUT/SETUP とバルクまたは制御 IN トランザクションの割り込みサービスルーチン**

- a) **バルク/制御 OUT/SETUP**

```
Unmask (NAK/TXERR/STALL/XFRC)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    De-allocate Channel
}
else if (STALL)
{
    Transfer Done = 1
    Unmask CHH
    Disable Channel
}
else if (NAK or TXERR )
{
    Rewind Buffer Pointers
    Unmask CHH
    Disable Channel
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
    }
    else
    {
        Reset Error Count
    }
}
else if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel
    }
}
else if (ACK)
{
    Reset Error Count
    Mask ACK
```

}

アプリケーションは、送信 FIFO およびリクエストキューの中に使用可能なスペースがある場合、データパケットを送信 FIFO に書き込むことが期待されます。アプリケーションは、OTG\_GINTSTS レジスタの NPTXFE 割り込みを使用して、送信 FIFO のスペースを見つけることができます。

b) バルク/制御 IN

```

Unmask (TXERR/XFRC/BBERR/STALL/DTERR)
if (XFRC)
{
    Reset Error Count
    Unmask CHH
    Disable Channel
    Reset Error Count
    Mask ACK
}
else if (TXERR or BBERR or STALL)
{
    Unmask CHH
    Disable Channel
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
    }
}
else if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel
    }
}
else if (ACK)
{
    Reset Error Count
    Mask ACK
}
else if (DTERR)
{
    Reset Error Count
}

```

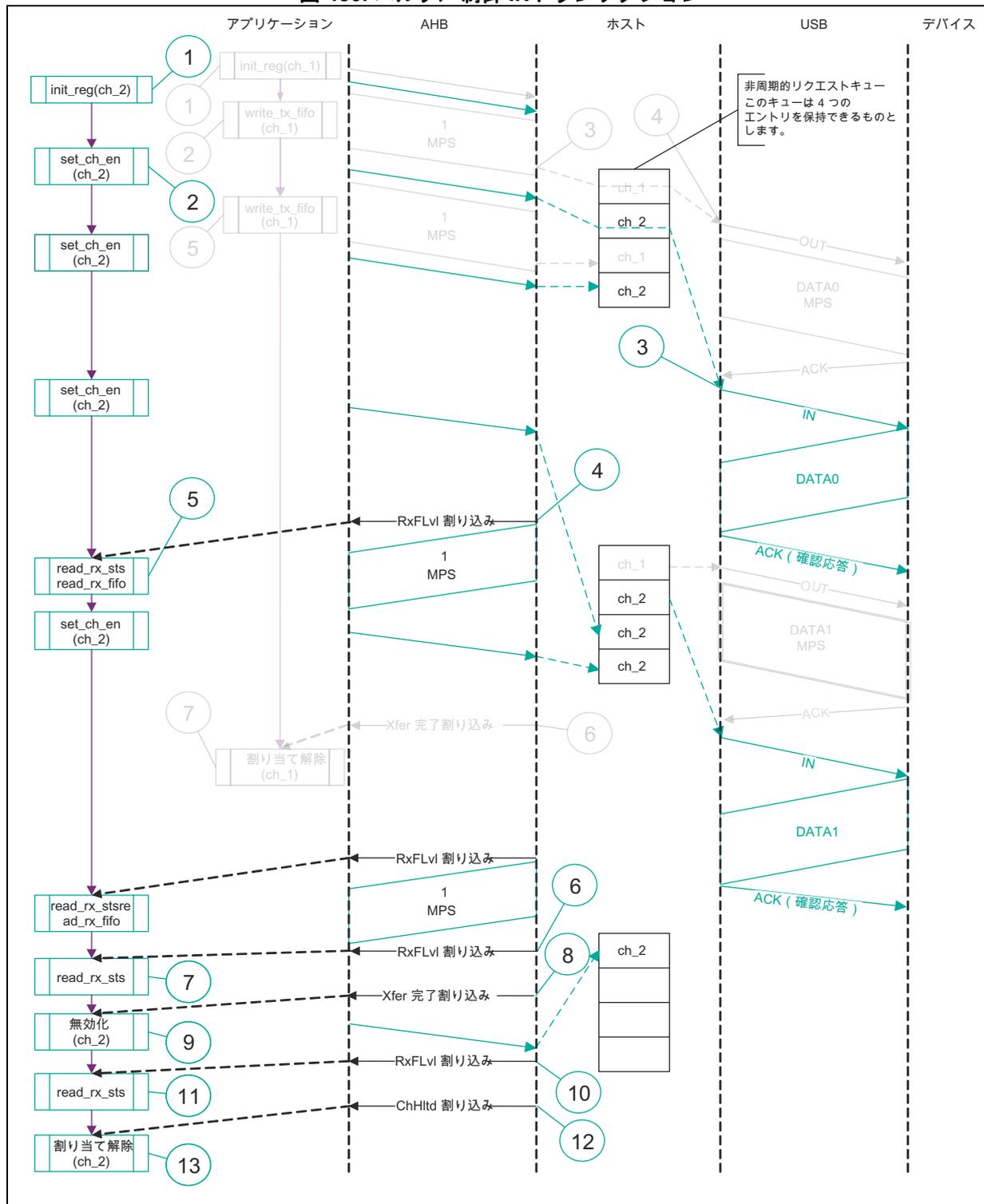
アプリケーションは、リクエストキューのスペースが使用可能なときには、XFRC 割り込みが受信されるまで、リクエストを書き込むことが期待されません。

## ● バルクおよび制御 IN トランザクション

典型的なバルクまたは制御 IN パイプライントランザクションレベルの動作を [図 450](#)に示します。チャンネル 2 (ch\_2) に注目してください。ここでは、以下のように仮定されています。

- アプリケーションは、最大パケットサイズ (転送サイズ = 1,024 バイト) の 2 つのパケットを受信しようとしています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットと、パケットあたり 2 つのステータスワード (72 バイト - FS の場合、520 バイト - HS の場合) を収容することができます。
- 非周期的リクエストキューの深さ = 4。

図 450. バルク/制御 IN トランザクション



動作シーケンスは、次のとおりです。

1. チャンネル 2 を初期化します。
2. OTG\_HCCHAR2 の CHENA ビットをセットして、IN リクエストを非周期的リクエストキューに書き込みます。
3. コアは、現在の OUT トランザクションの完了後、IN トークンの送信を試みます。
4. コアは、受信したパケットが受信 FIFO に書き込まれると、RXFLVL 割り込みを生成します。
5. RXFLVL 割り込みに応答して、RXFLVL 割り込みをマスクし、受信パケットステータスを読み出して受信バイト数を調べた後、それに従って受信 FIFO を読み出します。続いて、RXFLVL 割り込みのマスクを解除します。
6. コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割り込みを生成します。
7. アプリケーションは、受信パケットステータスを読み出して、それが IN データパケットでなかった場合 (OTG\_GRXSTSR の PKTSTS ≠ 0b0010)、無視する必要があります。
8. コアは、受信パケットステータスが読み出されると、XFRC 割り込みを生成します。
9. XFRC 割り込みに応答して、チャンネルを無効にし、それ以降のリクエストについて、OTG\_HCCHAR2 レジスタへの書き込みを停止します。コアは、OTG\_HCCHAR2 に書き込まれると、非周期的リクエストキューにチャンネルディセーブルリクエストを書き込みます。
10. コアは、停止ステータスが受信 FIFO に書き込まれると、RXFLVL 割り込みを生成します。
11. 受信パケットステータスを読み出して、それを無視します。
12. コアは、停止ステータスが受信 FIFO からポップされるとすぐに、CHH 割り込みを生成します。
13. CHH 割り込みに応答して、他の転送のためにチャンネル割り当てを解除します。
14. 非 ACK 応答の取り扱い

## ● 制御トランザクション

制御転送のセットアップ、データ、およびステータスステージは 3 つの個別の転送として実行されなければなりません。セットアップ、データ、およびステータスステージの OUT トランザクションは、すでに説明したバルク OUT トランザクションと同様に実行されます。データまたはステータスステージの IN トランザクションは、すでに説明したバルク IN トランザクションと同様に実行されます。3 つのステージすべてについて、アプリケーションは OTG\_HCCHAR1 レジスタの EPTYP フィールドを Control に設定することが期待されます。セットアップステージでは、アプリケーションは OTG\_HCTSIZ1 レジスタの PID フィールドを SETUP に設定することが期待されます。

## ● 割り込み OUT トランザクション

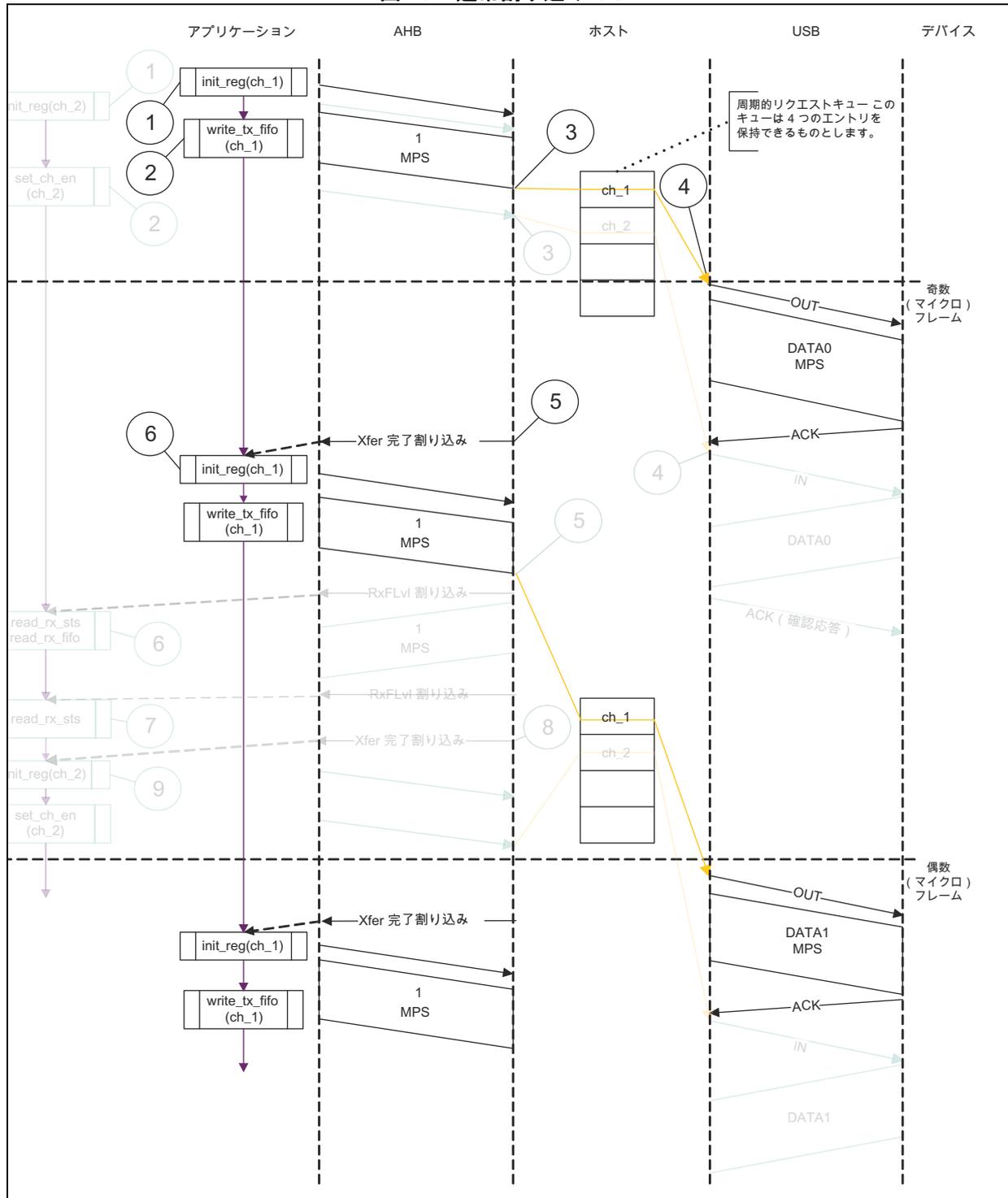
典型的な割り込み OUT 動作を [図 451](#) に示します。ここでは、以下のように仮定されています。

- アプリケーションは、奇数フレーム (転送サイズ = 1 024 バイト) から始めて、フレームごとに 1 パケット (最大 1 パケットサイズ) の送信を試みています。
- 周期的送信 FIFO は 1 パケット (1 KB) を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

1. チャネル 1 を初期化し、有効にします。アプリケーションは OTG\_HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
2. チャネル 1 の最初のパケットを書き込みます。
3. 各パケットの最後のワードの書き込みとともに、OTG\_FS/OTG\_HS ホストは、周期的リクエストキューにエントリを書き込みます。
4. OTG\_FS/OTG\_HS ホストは、次の（奇数）フレームで OUT トークンの送信を試みます。
5. OTG\_FS/OTG\_HS ホストは、最後のパケットが正常に送信されると XFRC 割り込みを生成します。
6. XFRC 割り込みに応答して、次の転送のためにチャネルを再初期化します。

図 451. 通常割り込み OUT



- 割り込み OUT/IN トランザクションのための割り込みサービスルーチン

- a) 割り込み OUT

```

Unmask (NAK/TXERR/STALL/XFRC/FRMOR)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    De-allocate Channel
}
else
    if (STALL or FRMOR)
    {
        Mask ACK
        Unmask CHH
        Disable Channel
        if (STALL)
        {
            Transfer Done = 1
        }
    }
    else
        if (NAK or TXERR)
        {
            Rewind Buffer Pointers
            Reset Error Count
            Mask ACK
            Unmask CHH
            Disable Channel
        }
        else
            if (CHH)
            {
                Mask CHH
                if (Transfer Done or (Error_count == 3))
                {
                    De-allocate Channel
                }
                else
                {
                    Re-initialize Channel (in next b_interval - 1 Frame)
                }
            }
        else
            if (ACK)
            {
                Reset Error Count
                Mask ACK
            }

```

アプリケーションは OTG\_GINTSTS レジスタの NPTXFE 割り込みを使用して、送信 FIFO のスペースを見つけます。

- b) 割り込み IN

```

Unmask (NAK/TXERR/XFRC/BBERR/STALL/FRMOR/DTERR)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    if (OTG_HCTSIZx.PKTCNT == 0)
    {
        De-allocate Channel
    }
    else
    {
        Transfer Done = 1
        Unmask CHH
        Disable Channel
    }
}
else
    if (STALL or FRMOR or NAK or DTERR or BBERR)
    {
        Mask ACK
        Unmask CHH
        Disable Channel
        if (STALL or BBERR)
        {
            Reset Error Count
            Transfer Done = 1
        }
        else
            if (!FRMOR)
            {
                Reset Error Count
            }
    }
else
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
        Unmask CHH
        Disable Channel
    }
else
    if (CHH)
    {
        Mask CHH
        if (Transfer Done or (Error_count == 3))
        {
            De-allocate Channel
        }
        else
            Re-initialize Channel (in next b_interval - 1 /Frame)
    }
}

```

```

    }
else
    if (ACK)
    {
        Reset Error Count
        Mask ACK
    }

```

## ● 割り込み IN トランザクション

ここでは、以下のように仮定されています。

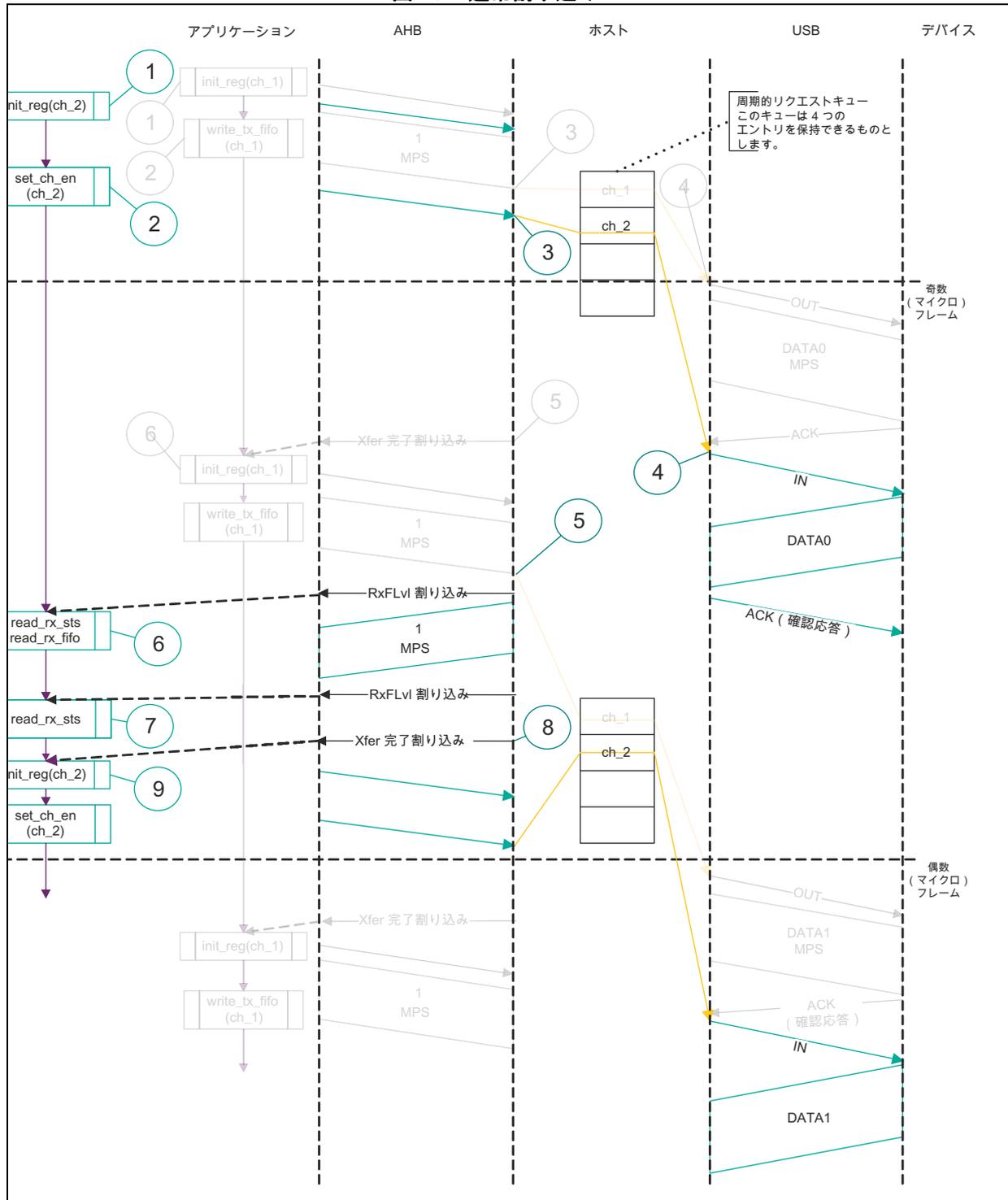
- アプリケーションは、奇数フレーム（転送サイズ = 1024 バイト）から始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の受信を試みています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットとパケットあたり 2 つのステータスワード（1,031 バイト）を保持することができます。
- 周期的リクエストキューの深さ = 4。

## ● 通常の割り込み IN 動作

動作シーケンスは、次のとおりです。

1. チャンネル 2 を初期化します。アプリケーションは OTG\_HCCHAR2 レジスタの ODDFRM ビットをセットする必要があります。
2. OTG\_HCCHAR2 の CHENA ビットをセットして IN リクエストを周期的リクエストキューに書き込みます。
3. OTG\_FS/OTG\_HS ホストは CHENA ビットがセットされた OTG\_HCCHAR2 レジスタの書き込みごとに、周期的リクエストキューに IN リクエストを書き込みます。
4. OTG\_FS/OTG\_HS ホストは、次の（奇数）フレームで IN トークンの送信を試みます。
5. IN パケットが受信され、受信 FIFO に書き込まれると OTG\_FS/OTG\_HS ホストは RXFLVL 割り込みを生成します。
6. RXFLVL 割り込みに応答して、受信パケットステータスを読み出して、受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。アプリケーションは、受信 FIFO を読み出す前に RXFLVL 割り込みをマスクし、すべてのパケットを読み出した後にマスクを解除しなければなりません。
7. コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割り込みを生成します。アプリケーションは、受信パケットステータスを読み出して、それが IN データパケットでなかった場合（GRXSTSR の PKTSTS ≠ 0b0010）、無視する必要があります。
8. コアは、受信パケットステータスが読み出されると XFRC 割り込みを生成します。
9. XFRC 割り込みに応答して OTG\_HCTSIZ2 の PKTCNT フィールドを読み出します。OTG\_HCTSIZ2 の PKTCNT ビットが 0 でない場合は、次の転送（ある場合）のためにチャンネルを再初期化する前に、チャンネルを無効にします。OTG\_HCTSIZ2 の PKTCNT ビットが 0 の場合、次の転送のためにチャンネルを再初期化します。今度は、アプリケーションは OTG\_HCCHAR2 レジスタの ODDFRM ビットをリセットする必要があります。

図 452. 通常割り込み IN



- アイソクロナス OUT トランザクション

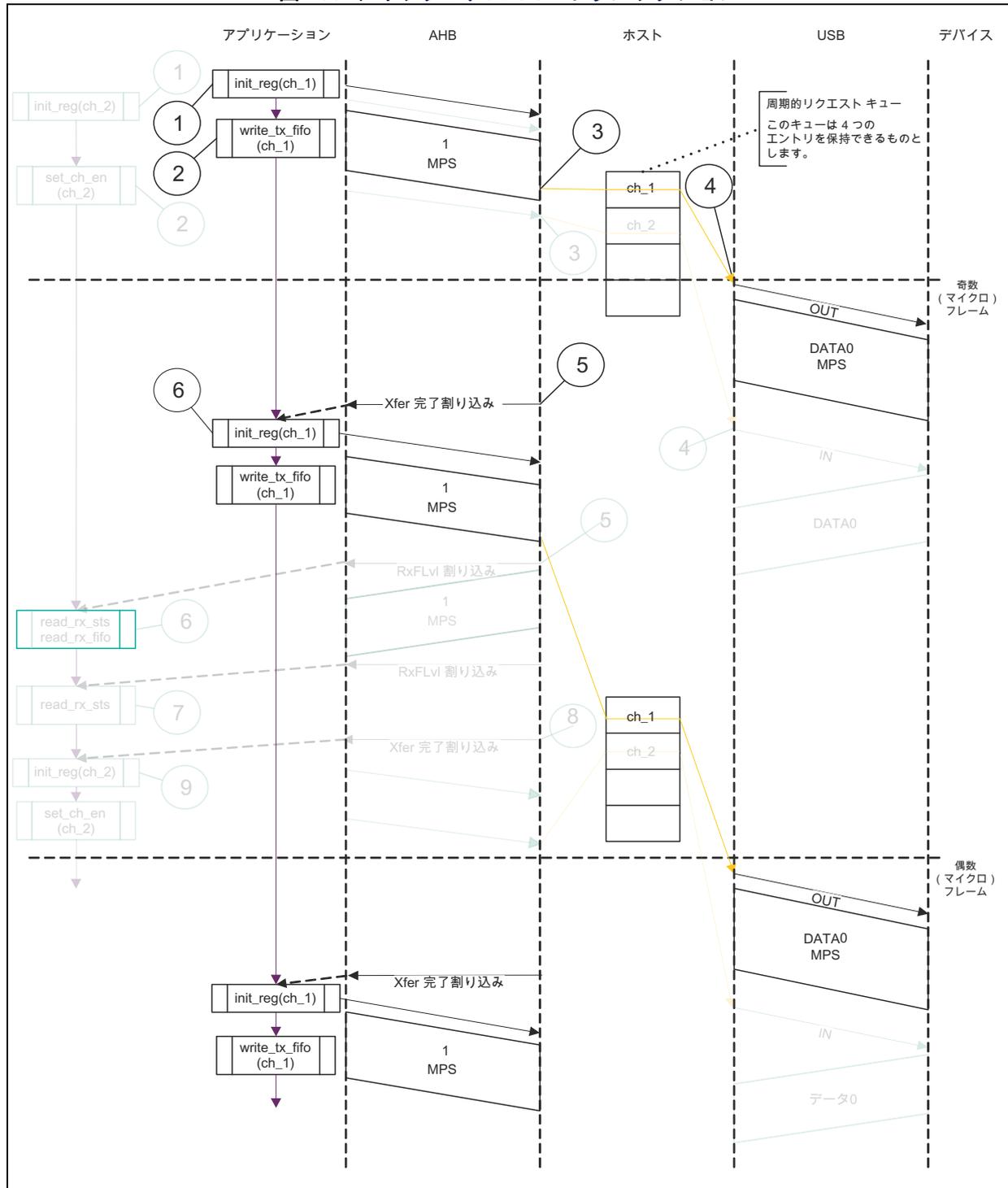
典型的なアイソクロナス OUT 動作を [図 452](#) に示します。ここでは、以下のように仮定されています。

- アプリケーションは、奇数フレーム（転送サイズ = 1,024 バイト）から始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の送信を試みています。
- 周期的送信 FIFO は 1 パケット（1 KB）を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

1. チャンネル 1 を初期化し、有効にします。アプリケーションは OTG\_HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
2. チャンネル 1 の最初のパケットを書き込みます。
3. 各パケットの最後のワードの書き込みとともに、OTG\_FS/OTG\_HS ホストは、周期的リクエストキューにエントリを書き込みます。
4. OTG\_FS/OTG\_HS ホストは、次の（奇数）フレームで OUT トークンの送信を試みます。
5. OTG\_FS/OTG\_HS ホストは、最後のパケットが正常に送信されると XFRC 割り込みを生成します。
6. XFRC 割り込みに応答して、次の転送のためにチャンネルを再初期化します。
7. 非 ACK 応答の取り扱い

図 453. アイソクロナス OUT トランザクション



- アイソクロナス OUT/IN トランザクションのための割り込みサービスルーチン

コードサンプル: アイソクロナス OUT

```
Unmask (FRMOR/XFRC)
if (XFRC)
{
    De-allocate Channel
}
else
if (FRMOR)
{
    Unmask CHH
    Disable Channel
}
else
if (CHH)
{
    Mask CHH
    De-allocate Channel
}
```

コードサンプル: アイソクロナス IN

```
Unmask (TXERR/XFRC/FRMOR/BBERR)
if (XFRC or FRMOR)
{
    if (XFRC and (OTG_HCTSIZx.PKTCNT == 0))
    {
        Reset Error Count
        De-allocate Channel
    }
    else
    {
        Unmask CHH
        Disable Channel
    }
}
else
if (TXERR or BBERR)
{
    Increment Error Count
    Unmask CHH
    Disable Channel
}
else
if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
```

```
        Re-initialize Channel  
    }  
}
```

## ● アイソクロナス IN トランザクション

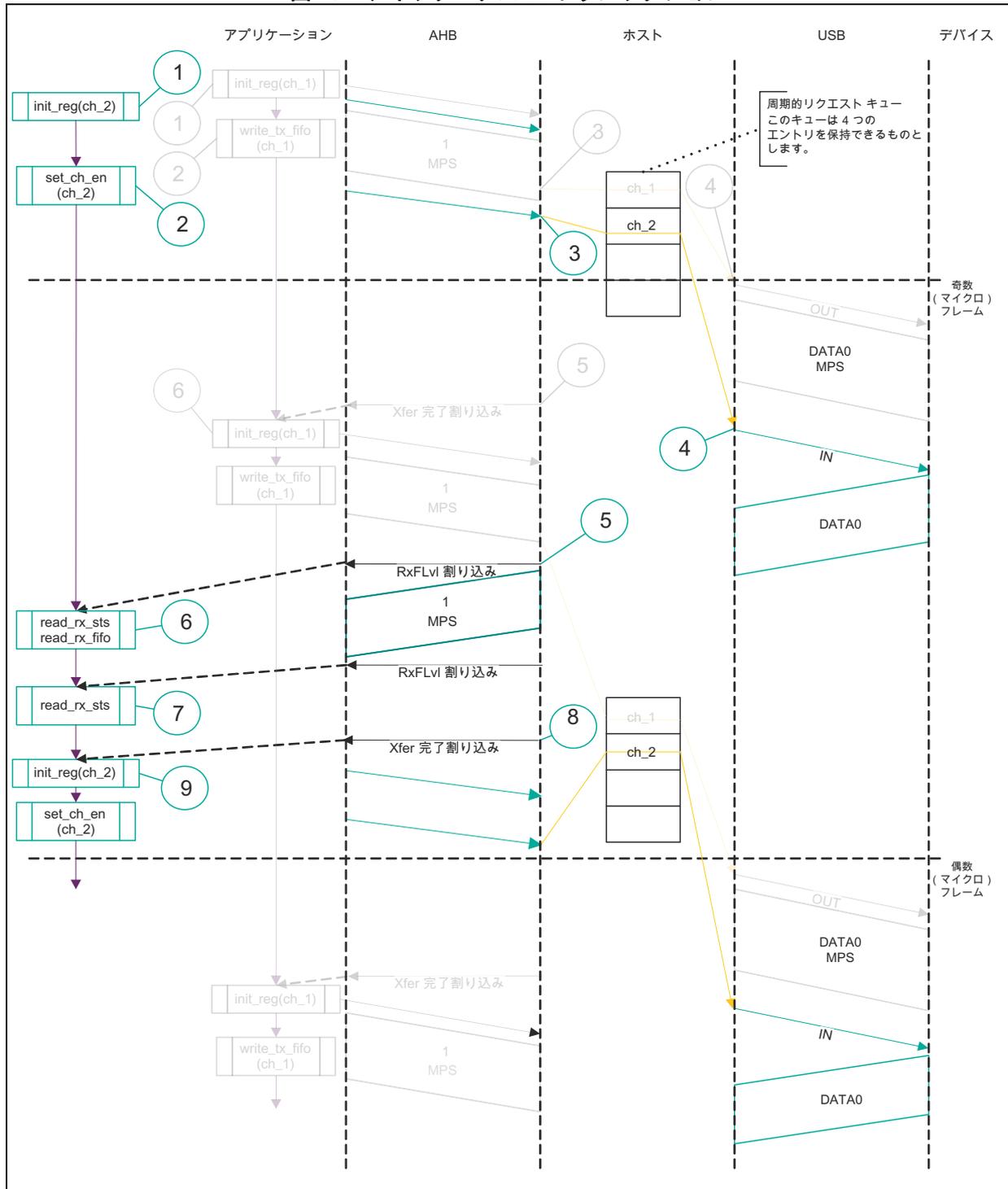
ここでは、以下のように仮定されています。

- アプリケーションは、次の奇数フレーム（転送サイズ = 1,024 バイト）から始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の受信を試みています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットとパケットあたり 2 つのステータスワード（1,031 バイト）を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

1. チャンネル 2 を初期化します。アプリケーションは OTG\_HCCHAR2 レジスタの ODDFRM ビットをセットする必要があります。
2. OTG\_HCCHAR2 の CHENA ビットをセットして IN リクエストを周期的リクエストキューを書き込みます。
3. The OTG\_FS/OTG\_HS ホストは CHENA ビットがセットされた OTG\_HCCHAR2 レジスタの書き込みごとに、周期的リクエストキューに IN リクエストを書き込みます。
4. OTG\_FS/OTG\_HS ホストは、次の奇数フレームで IN トークンの送信を試みます。
5. IN パケットが受信され、受信 FIFO に書き込まれると OTG\_FS/OTG\_HS ホストは RXFLVL 割り込みを生成します。
6. RXFLVL 割り込みに応答して、受信パケットステータスを読み出して、受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。アプリケーションは、受信 FIFO を読み出す前に RXFLVL 割り込みをマスクし、すべてのパケットを読み出した後でマスクを解除しなければなりません。
7. コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割り込みを生成します。今度は、アプリケーションは受信パケットステータスを読み出して、それが IN データパケットでなかった場合 (OTG\_GRXSTSR の PKTSTS ビット  $\neq$  0b0010)、無視する必要があります。
8. コアは、受信パケットステータスが読み出されると XFRC 割り込みを生成します。
9. XFRC 割り込みに応答して OTG\_HCTSIZ2 の PKTCNT フィールドを読み出します。OTG\_HCTSIZ2 の PKTCNT  $\neq$  0 の場合、次の転送（ある場合）のためにチャンネルを再初期化する前に、チャンネルを無効にします。OTG\_HCTSIZ2 の PKTCNT = 0 の場合、次の転送のためにチャンネルを再初期化します。今度は、アプリケーションは OTG\_HCCHAR2 レジスタの ODDFRM ビットをリセットする必要があります。

図 454. アイソクロナス IN トランザクション



- **キューの深さの選択**

周期的および非周期的リクエストキューの深さは、アクセスされる周期的/非周期的エンドポイントの数に合うように注意して選択してください。

非周期的リクエストキューの深さは、非周期的転送の性能に影響を与えます。キューが深いほど (FIFO のサイズが十分であれば)、コアは非周期的転送をより多くパイプライン化できます。キューのサイズが小さいと、コアはキューのスペースが空いたときしか新しいリクエストを入れることができません。

コアの周期的リクエストキューの深さは、周期的転送をスケジュールどおりに実行するために不可欠です。周期的キューの深さは、マイクロフレーム内でスケジュールされた周期的転送の数に基づいて選択してください。周期的リクエストキューの深さがマイクロフレーム内でスケジュールされた周期的転送の数より小さい場合、フレームオーバーラン条件が発生します。

- **バブル条件の取り扱い**

OTG\_FS/OTG\_HS コントローラは、パケットバブルとポートバブルの 2 つのバブルを処理します。パケットバブルは、デバイスがチャネルの最大パケットサイズよりも多くのデータを送信した場合に発生します。ポートバブルは、コアが EOF2 (フレーム 2 の終わりの SOF にきわめて近い) でデバイスからデータを受信し続けると発生します。

OTG\_FS/OTG\_HS コントローラがパケットバブルを検出すると、Rx バッファへのデータの書き込みを停止して、パケットの終わり (EOP) を待ちます。EOP を検出すると コントローラは Rx バッファにすでに書き込まれたデータを一扫して、アプリケーションに対するバブル割り込みを生成します。

OTG\_FS/OTG\_HS コントローラがポートバブルを検出すると、Rx FIFO を一扫して、ポートを無効にします。コアは、ポートディセーブル割り込み (OTG\_GINTSTS の HPRTINT、OTG\_HPRT の PENCHNG) を生成します。この割り込みを受信すると、アプリケーションは OTG\_HPRT の POCA ビットをチェックして、これが過電流条件 (ポートディセーブル割り込みのもう 1 つの原因) によるものではないことを確認してから、ソフトリセットを行う必要があります。コアは、ポートバブル条件を検出した後は、それ以上トークンを送信しません。

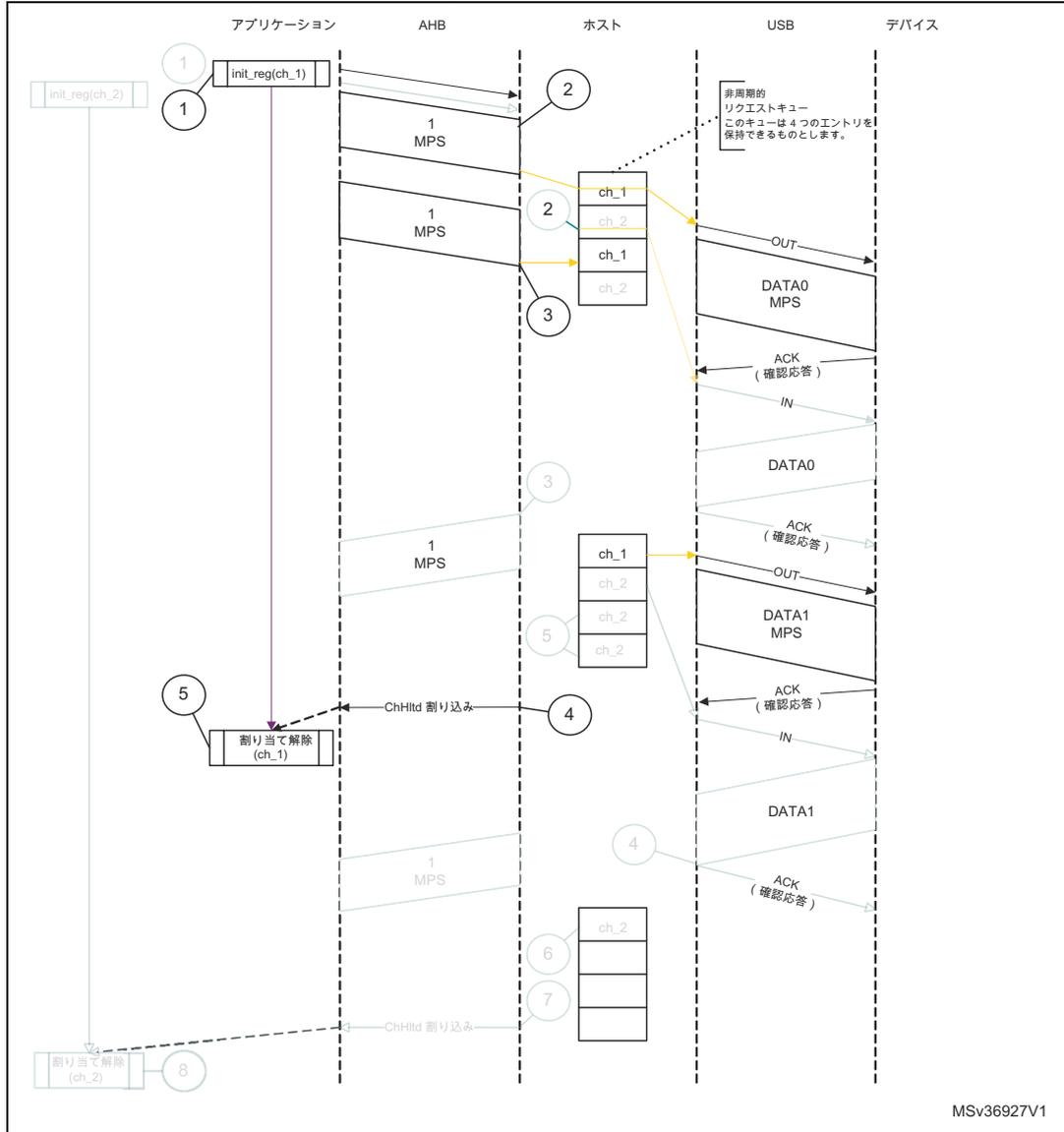
**注：** 以下のセクションで扱う DMA の設定は USB OTG HS にのみ適用されます。

- **バルクおよび制御 OUT/SETUP トランザクション - DMA モード**

動作シーケンスは、次のとおりです。

1. **セクション：チャネルの初期化**で説明しているように、チャネル 1 を初期化し、有効にします。
2. HS\_OTG ホストはチャネルが有効化されると直ちに最初のパケットのフェッチを開始します。内部DMA モードの場合、OTG\_HS ホストはパケットのフェッチにプログラムされ DMA アドレスを使います。
3. 2 つ目 (最後) のパケットの最後のダブルワードをフェッチした後、OTG\_HS ホストは以降の調停に対し内部的にチャネル 1 をマスクします。
4. OTG\_HS ホストは、最後のパケットが送信されると直ちに CHH 割り込みを生成します。
5. CHH 割り込みに応答して、他の転送のためにチャネル割り当てを解除します。

図 455. 通常のバルク/制御 OUT/SETUP トランザクション - DMA



- **内部 DMA による NAK と NYET の処理**

1. OTG\_HS ホストはパルク OUT トランザクションを送信します。
2. デバイスは NAK または NYET で応答します。
3. アプリケーションで NAK または NYET のマスクを解除した場合、コアはアプリケーションへ対応する割り込みを生成します。コアがバッファポインタの戻し処理を行っている場合およびアプリケーション介入なしでチャンネルを再初期化している場合、アプリケーションはこれらの割り込みを生成する必要はありません。
4. コアは自動的にPINGトークンを発行します。
5. デバイスが ACK を返すと、コアは転送を続けます。オプションとして、アプリケーションはこれらの割り込みを使用することができます。この場合 NAK または NYET 割り込みはアプリケーションからマスクされます。

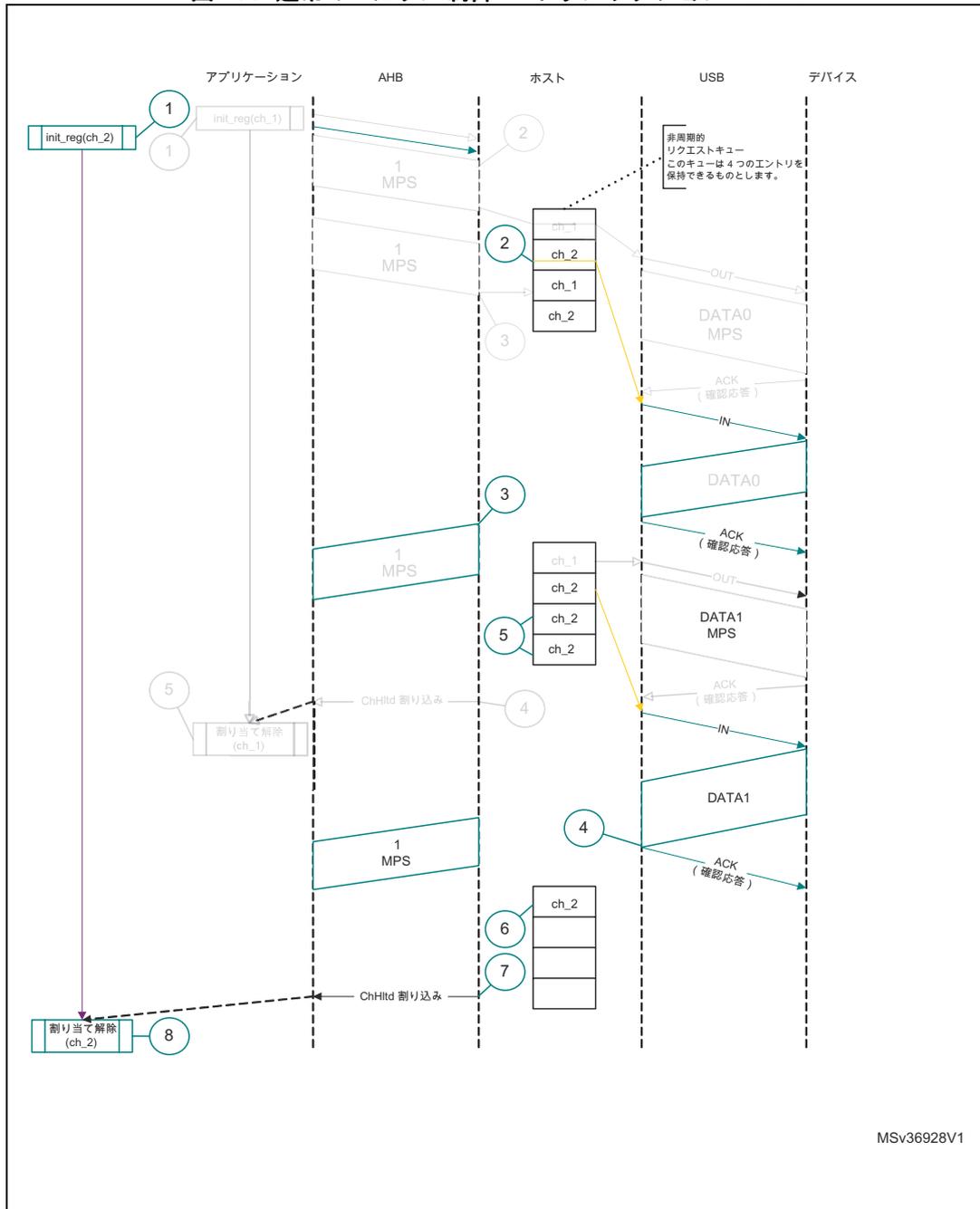
ホスト機能が NAK または NYET を受信した場合、コアは個別の割り込みを生成しません。

- **パルクおよび制御 IN トランザクション - DMA モード**

動作シーケンスは、次のとおりです。

1. **セクション : チャンネルの初期化**で説明しているように、使用するチャンネル (チャンネル x) を初期化および有効化します。
2. チャンネルがアービタから許可を受信すると直ちに、OTG\_HS ホストは IN リクエストをリクエストキューへ書き込みます (調停はラウンドロビン方式で行われます)。
3. OTG\_HS ホストはエラーなしで最終バイトを受信すると、直ちにシステムメモリへの受信データの書き込みを開始します。
4. OTG\_HS ホストは最終パケットを受信すると、内部フラグをセットしてリクエストキューから余分な IN リクエストを削除します。
5. OTG\_HS ホストは余分なリクエストを一掃します。
6. チャンネル x を無効化する最終リクエストがリクエストキューに書き込まれます。この時点で、チャンネル 2 は以降の調停に対して内部でマスクされます。
7. OTG\_HS ホストは、ディセーブルリクエストがキューのトップに来ると直ちに CHH 割り込みを生成します。
8. CHH 割り込みに応答して、他の転送のためにチャンネル割り当てを解除します。

図 456. 通常のバルク/制御 IN トランザクション - DMA

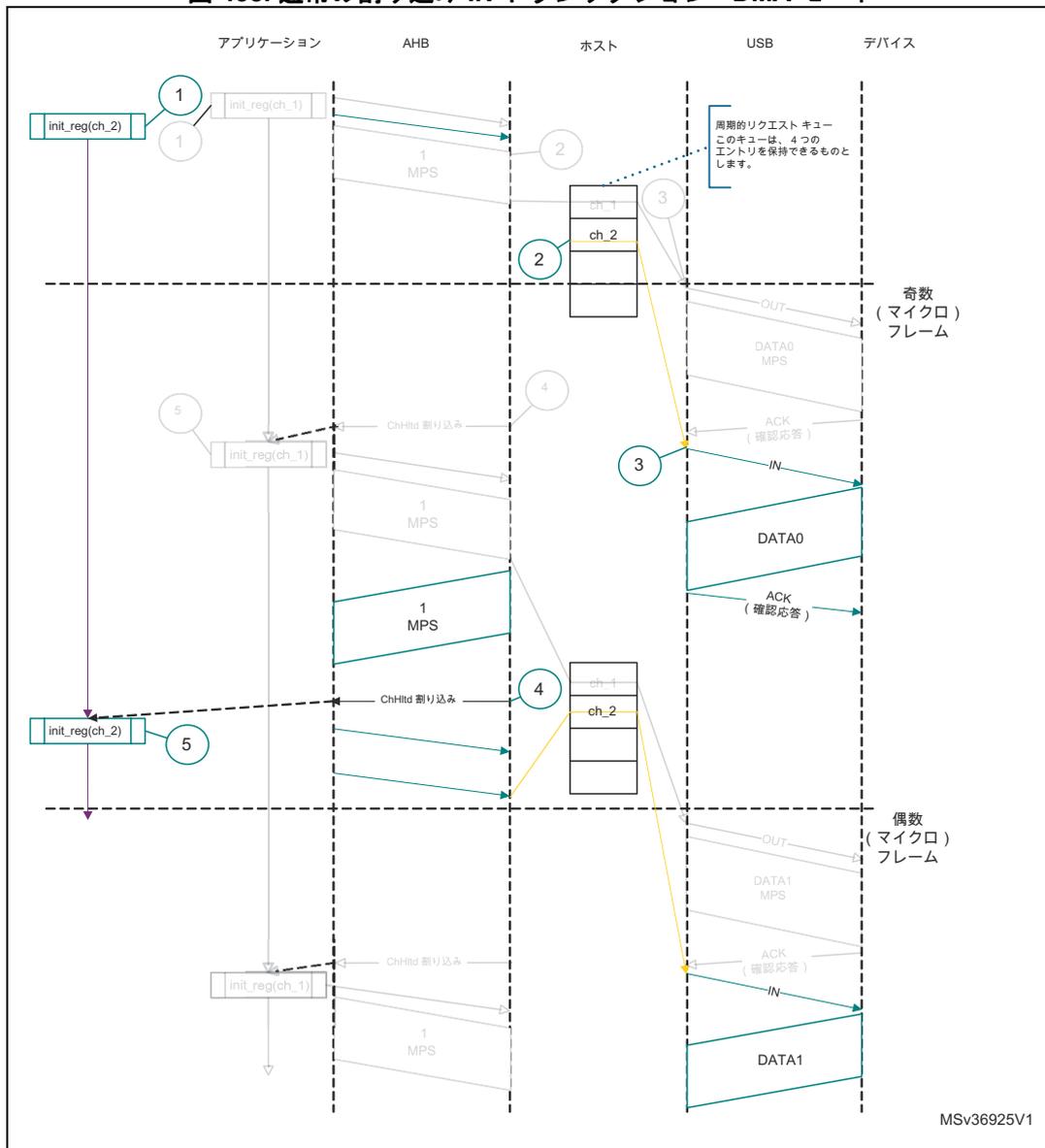


- 割り込み OUT トランザクション - DMA モード

1. **セクション：チャンネルの初期化**で説明しているように、チャンネル x を初期化し、有効にします。
2. OTG\_HS ホストはチャンネルが有効化されると直ちに、最初のパケットのフェッチを開始し OUT リクエストを最後のダブルワードフェッチと一緒に書き込みます。広帯域 転送では HS\_OTG ホストは、次のパケットのフェッチ MC フィールドで指定された値まで) 続けた後に、次のチャンネルへ切り替わります。
3. OTG\_HS ホストは、次の奇数フレーム/マイクロフレームの開始で OUT トークンの送信を試みます。
4. パケットの正常送信の後、OTG\_HS ホストは CHH 割り込みを生成します。
5. CHH 割り込みに応答して、次の転送のためにチャンネルを再初期化します。



図 458. 通常の割り込み IN トランザクション - DMA モード

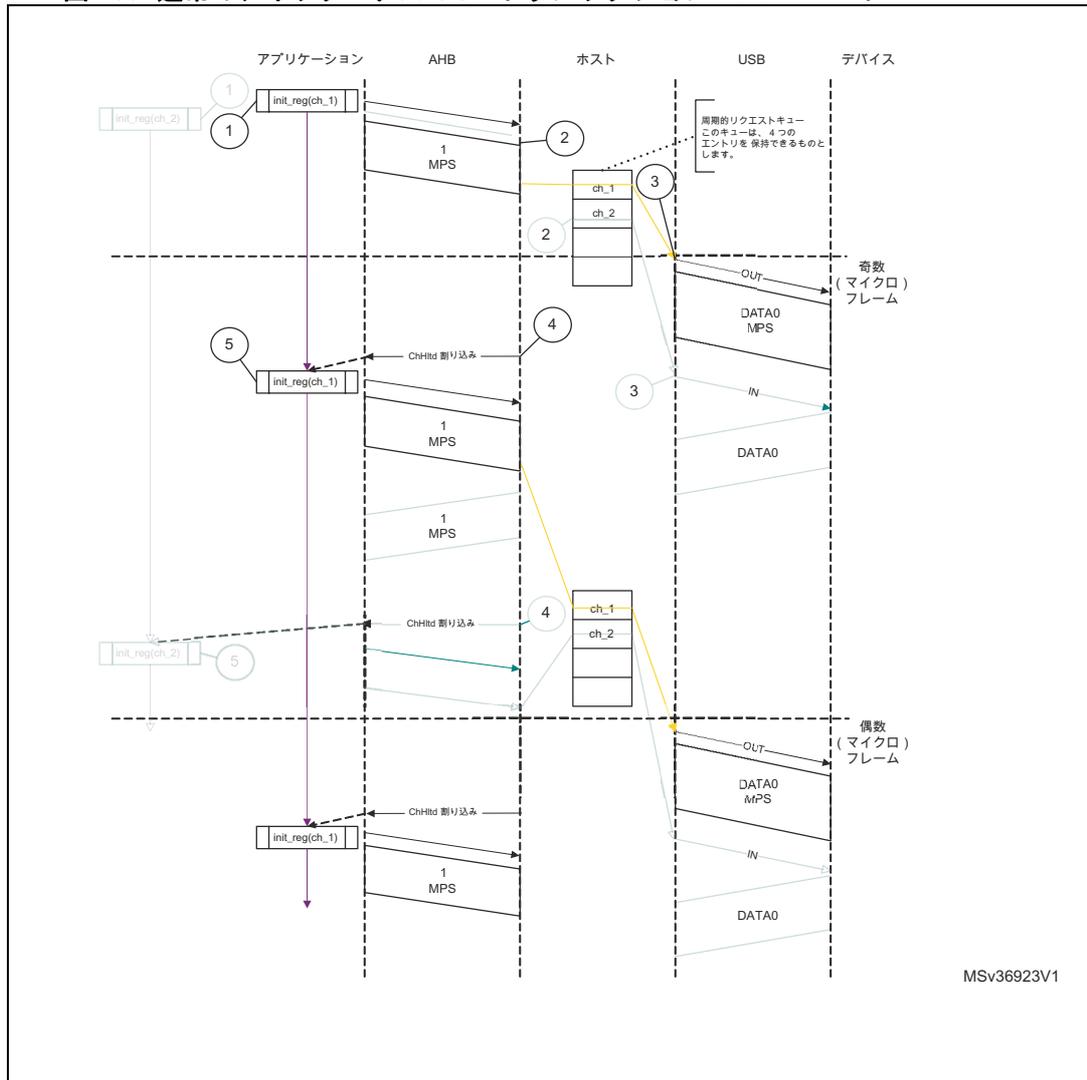


MSv36925V1

● アイソクロナス OUT トランザクション - DMA モード

1. **セクション : チャンルの初期化**で説明しているように、チャンネル x を初期化し、有効にします。
2. OTG\_HS ホストはチャンネルが有効化されると直ちに、最初のパケットのフェッチを開始し、OUT リクエストを最後のダブルワードフェッチと一緒に書き込みます。広帯域 転送では OTG\_HS ホストは、次のパケットのフェッチ (MC フィールドで指定された値まで) 続けた後に、次のチャンネルへ切り替わります。
3. OTG\_HS ホストは、次の奇数フレーム/マイクロフレームの開始で OUT トークンの送信を試みます。
4. パケットの正常送信の後、HS\_OTG ホストは CHH 割り込みを生成します。
5. CHH 割り込みに応答して、次の転送のためにチャンネルを再初期化します。

図 459. 通常のアイソクロナス OUT トランザクション - DMA モード



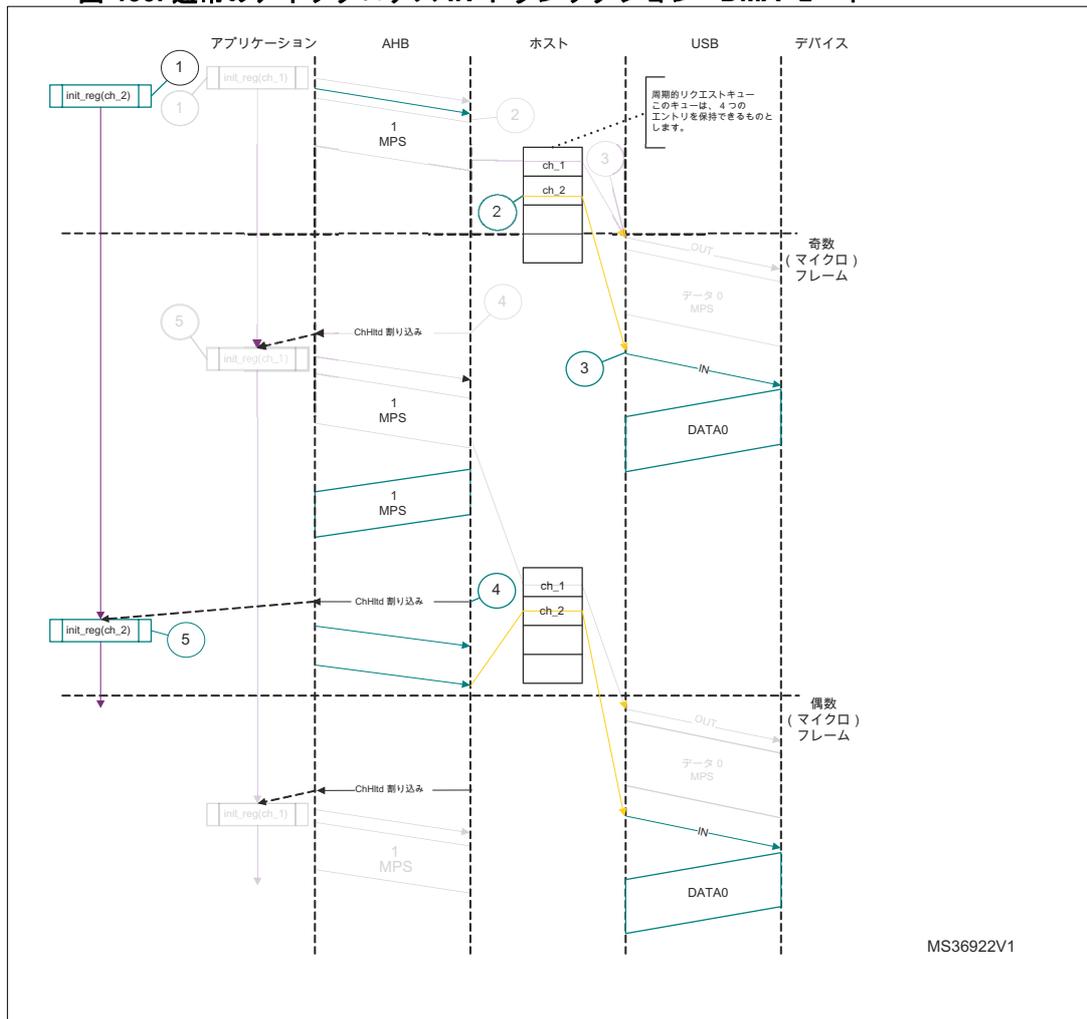
MSV36923V1

● **アイソクロナス IN トランザクション - DMA モード**

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**で説明しているように、チャンネル x を初期化し、有効にします。
2. チャンネル x がアービタから許可を受け取ると直ちに、OTG\_HS ホストは IN リクエストをリクエストキューへ書き込みます (ラウンドロビン方式、フェアネスを使用)。広帯域転送では OTG\_HS ホストは MC 回までの連続書き込みを行います。
3. OTG\_HS ホストは、次の奇数フレーム/マイクロフレームの開始で IN トークンの送信を試みます。
4. パケットが受信され、受信 FIFO に書き込まれると、OTG\_HS ホストは CHH 割り込みを生成します。
5. CHH 割り込みに応答して、次の転送のためにチャンネルを再初期化します。

図 460. 通常のアイソクロナス IN トランザクション - DMA モード



● バルクおよび制御 OUT/SETUP スプリットトランザクション - DMA モード

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**の説明に従い、スタートスプリット用にチャンネル x を初期化および有効化します。
2. OTG\_HS ホストはチャンネルが有効化されると直ちに、最初のパケットのフェッチを開始し OUT リクエストを最後のダブルワードフェッチと一緒に書き込みます。
3. スタートスプリットの正常送信の後 OTG\_HS ホストは CHH 割り込みを生成します。
4. CHH 割り込みに応答してHCSPLT1 の COMPLSPLT ビットをセットし、スプリット全体を送信します。
5. スプリット全体の正常送信の後 OTG\_HS ホストは CHH 割り込みを生成します。
6. CHH 割り込みに応答して、チャンネルの割り当てを解除します。

- **バルク/制御 IN スプリットトランザクション - DMA モード**

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**で説明しているように、チャンネル x を初期化し、有効にします。
2. アービタから許可を得た後、OTG\_HS ホストはスタートスプリットリクエストを非周期的リクエストへ書き込みます。OTG\_HS ホストは、リクエストを書き込んだ後に、調停のために内部でチャンネル x をマスクします。
3. IN トークンが送信されると直ちに OTG\_HS ホストは CHH 割り込みを生成します。
4. CHH 割り込みに応答して HCSPILT2 の COMPLSPLT ビットをセットし、スプリットトークン全体を送信するためにチャンネルを再度有効化します。これにより、調停用にチャンネル x のマスクが解除されます。
5. アービタから許可を得た後、OTG\_HS ホストはスプリット全体リクエストを非周期的リクエストへ書き込みます。
6. OTG\_HS ホストはパケットを正常に受信した後、パケットのシステムメモリへの書き込みを開始します。
7. 受信パケットをシステムメモリへ書き込むと直ちに、OTG\_HS ホストは CHH 割り込みを生成します。
8. CHH 割り込みに応答して、チャンネルの割り当てを解除します。

- **割り込み OUT スプリットトランザクション - DMA モード**

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**の説明に従い、スタートスプリット用にチャンネル 1 を初期化および有効化します。アプリケーションは HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
2. OTG\_HS ホストはパケットの読み出しを開始します。
3. OTG\_HS ホストは、スタートスプリットトランザクションの送信を試みます。
4. スタートスプリットの正常送信の後 OTG\_HS ホストは CHH 割り込みを生成します。
5. CHH 割り込みに応答して HCSPILT1 の COMPLSPLT ビットをセットし、スプリット全体を送信します。
6. スプリット全体トランザクションの正常送信の後、OTG\_HSホストは CHH 割り込みを生成します。
7. CHH 割り込みに応答して、チャンネルの割り当てを解除します。

## ● 割り込み IN スプリットトランザクション - DMA モード

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**の説明に従い、スタートスプリット用にチャンネル x を初期化および有効化します。
2. チャンネル x がアービタから許可を受け取ると直ちに OTG\_HS ホストは IN リクエストをリクエストキューへ書き込みます。
3. OTG\_HS ホストは、次の奇数マイクロフレームの開始にスタートスプリット IN トークンの送信を試みます。
4. OTG\_HS ホストはスタートスプリット IN トークンを正常に送信した後に CHH 割り込みを生成します。
5. CHH 割り込みに応答して HCSPLT2 の COMPLSPLT ビットをセットし、スプリット全体を送信します。
6. OTG\_HS ホストはパケットを正常に受信すると直ちに、データのシステムメモリへの書き込みを開始します。
7. OTG\_HS ホストは受信データをシステムメモリへ転送した後、CHH 割り込みを生成します。
8. CHH 割り込みに応答して、次のスタートスプリットのためにチャンネルの割り当て解除または再初期化を行います。

## ● アイソクロナス OUT スプリットトランザクション - DMA モード

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**の説明に従い、スタートスプリット (開始) 用にチャンネル x を初期化および有効化します。アプリケーションは HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。MPS フィールドをプログラムします。
2. OTG\_HS ホストはパケットの読み出しを開始します。
3. スタートスプリット (開始) の正常送信の後、OTG\_HS ホストは CHH 割り込みを生成します。
4. CHH 割り込みに応答して、スタートスプリット (終わり) を送信するためレジスタを再初期化します。
5. スタートスプリット (終わり) の正常送信の後、OTG\_HS ホストは CHH 割り込みを生成します。
6. CHH 割り込みに応答して、チャンネルの割り当てを解除します。

## ● アイソクロナス IN スプリットトランザクション - DMA モード

動作シーケンス (チャンネル x) は、次のとおりです。

1. **セクション : チャンネルの初期化**の説明に従い、スタートスプリット用にチャンネル x を初期化および有効化します。
2. チャンネル x がアービタから許可を受け取ると直ちに OTG\_HS ホストは IN リクエストをリクエストキューへ書き込みます。
3. OTG\_HS ホストは、次の奇数マイクロフレームの開始にスタートスプリット IN トークンの送信を試みます。
4. OTG\_HS ホストはスタートスプリット IN トークンを正常に送信した後に CHH 割り込みを生成します。
5. CHH 割り込みに応答して HCSPLT2 の COMPLSPLT ビットをセットし、スプリット全体を送信します。
6. OTG\_HS ホストはパケットを正常に受信すると直ちに、データのシステムメモリへの書き込みを開始します。

OTG\_HS ホストは受信データをシステムメモリへ転送した後、CHH 割り込みを生成します。CHH 割り込みに応答して、次のスタートスプリットのためにチャンネルの割り当て解除または再初期化を行います。

注： このセクションの内容は、USB OTG HS にのみ適用されます。

## 37.16.6 デバイスプログラミングモデル

### USB リセット時のエンドポイントの初期化

- すべての OUT エンドポイントの NAK ビットをセットします。
  - OTG\_DOEPCTLx の SNAK = 1 (すべての OUT エンドポイントについて)
- 以下の割り込みビットのマスクを解除します。
  - OTG\_DAINMSK の INEP0 = 1 (制御 0 IN エンドポイント)
  - OTG\_DAINMSK の OUTEP0 = 1 (制御 0 OUT エンドポイント)
  - OTG\_DOEPMSK の STUPM = 1
  - OTG\_DOEPMSK の XFRCM = 1
  - OTG\_DIEPMSK の XFRCM = 1
  - OTG\_DIEPMSK の TOM = 1
- 各 FIFO の DATA FIFO RAM をセットアップします。
  - OTG\_GRXFSIZ レジスタをプログラムして、制御 OUT データとセットアップデータを受信できるようにします。閾値の設定が無効の場合、これは、少なくとも、制御エンドポイント 0 の最大パケットサイズ 1 個分 + 2 ワード (制御 OUT データパケットのステータス用) + 10 ワード (セットアップパケット用) でなければなりません。
  - OTG\_DIEPTXF0 レジスタを、制御 IN データを送信できるようにプログラムします (選択された FIFO 番号に応じて)。これは、少なくとも、制御エンドポイント 0 の最大パケットサイズ 1 個分以上でなければなりません。
- SETUP パケットを受信するために、制御 OUT エンドポイント 0 のエンドポイント固有レジスタの以下のフィールドをプログラムします。
  - OTG\_DOEPTSIZ0 の STUPCNT = 3 (最大 3 つの連続 SETUP パケットを受信するため)
- DMA モードの USB\_OTG\_HS では、DOEPDMA0 レジスタは受信した全 SETUP パケットを保存する有効なメモリアドレスを持つ必要があります。

この時点で SETUP パケットを受信するために必要なすべての初期化が終了したことになります。

### エニユメレーション完了時のエンドポイント初期化

- エニユメレーション終了割り込み (OTG\_GINTSTS の ENUMDNE ビット) 時には OTG\_DSTS レジスタを読み出して、エニユメレーションスピードを決めます。
- OTG\_DIEPCTL0 の MPSIZ フィールドをプログラムして、最大パケットサイズを設定します。このステップでは、制御エンドポイント 0 を設定します。制御エンドポイントの最大パケットサイズは、エニユメレーションスピードに依存します。
- DMA モードの USB OTG HS では、制御 OUT エンドポイント 0 を有効化する DOEPCTL0 レジスタをプログラムして、SETUP パケットを受信するようにします。

この時点で、デバイスは SOF パケットを受信する準備ができ、制御エンドポイント 0 で制御転送を行うように設定されたことになります。

## SetAddress コマンド受信時のエンドポイントの初期化

このセクションでは SETUP パケットで SetAddress コマンドを受信したときにアプリケーションが行わなければならないことについて説明します。

1. OTG\_DCFG レジスタに SetAddress コマンドで受信したデバイスアドレスをプログラムします。
2. ステータス IN パケットを送信するように、コアをプログラムします。

## SetConfiguration/SetInterface コマンド受信時のエンドポイントの初期化

このセクションでは SETUP パケットで SetConfiguration または SetInterface コマンドを受信したときにアプリケーションが行わなければならないことについて説明します。

1. SetConfiguration コマンドを受信したとき、アプリケーションは、新しい設定で有効なエンドポイントの特性で、エンドポイントのレジスタをプログラムする必要があります。
2. SetInterface コマンドを受信したとき、アプリケーションは、このコマンドの影響を受けるエンドポイントのレジスタをプログラムする必要があります。
3. 前の設定または代替設定ではアクティブであったエンドポイントが、新しい設定または代替設定では無効なことがあります。これらの無効なエンドポイントは、機能を停止させる必要があります。
4. アクティブな各エンドポイントの割り込みのマスクを解除し、OTG\_DAINMSK レジスタですべての非アクティブなエンドポイントに対する割り込みをマスクします。
5. 各 FIFO のデータ FIFO RAM をセットアップします。
6. 必要なすべてのエンドポイントを設定した後、アプリケーションは、ステータス IN パケットを送信するようにコアをプログラムする必要があります。

この時点で、デバイスのコアは、任意のタイプのデータパケットを送信/受信できるように設定されたこととなります。

## エンドポイントのアクティブ化

このセクションでは、デバイスエンドポイントをアクティブ化するか、既存のデバイスエンドポイントを新しいタイプに設定するために必要な手順について説明します。

1. 必要なエンドポイントの特性を OTG\_DIEPCTLx レジスタ (IN または双方向エンドポイントの場合)、または OTG\_DOEPCTLx レジスタ (OUT または双方向エンドポイントの場合) の以下のフィールドにプログラムします。
  - 最大パケットサイズ
  - USB アクティブエンドポイント = 1
  - エンドポイント開始データトグル (割り込みおよびバルクエンドポイントの場合)
  - エンドポイントタイプ
  - Tx FIFO 番号
2. エンドポイントがアクティブ化されると、コアは、そのエンドポイントに宛てたトークンのデコードを開始し、そのエンドポイントで受信された有効な各トークンについて有効なハンドシェイクを送出します。

## エンドポイントの機能停止

このセクションでは、既存のエンドポイントの機能を停止させるために必要な手順について説明します。

1. 非アクティブにすべきエンドポイントでは、OTG\_DIEPCTLx レジスタ (IN または双方向エンドポイントの場合)、または OTG\_DOEPCTLx レジスタ (OUT または双方向エンドポイントの場合) の USB のアクティブエンドポイントビットをクリアします。
2. エンドポイントが非アクティブな状態になると、コアは、そのエンドポイントにアドレス指定されたトークンを無視するので USB 上でタイムアウトが発生します。

**注：** アプリケーションは、以下の条件を満たすように、デバイスコアがトラフィックを処理できるようにセットアップする必要があります。

**OTG\_GINTMSK レジスタの NPTXFEM および RXFLVLM をクリアする必要があります。**

## 動作モデル

### SETUP および OUT データ転送

このセクションでは、データ OUT 転送および SETUP トランザクション時の内部データフローとアプリケーションレベルの動作について説明します。

#### ● パケットの読み出し

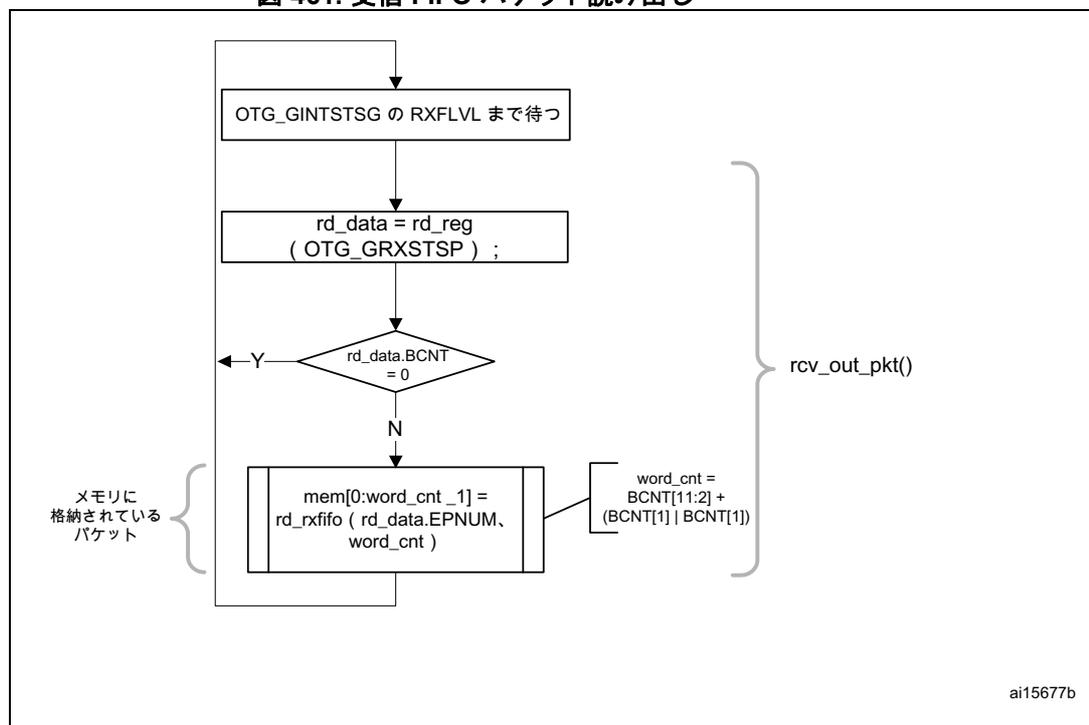
このセクションでは、受信 FIFO からパケット (OUT データおよび SETUP パケット) を読み出す方法について説明します。

1. RXFLVL 割り込み (OTG\_GINTSTS レジスタ) を受信すると、アプリケーションは、受信ステータスポップレジスタ (OTG\_GRXSTSP) を読み出す必要があります。
2. アプリケーションは、RXFLVL = 0 (OTG\_GINTMSK レジスタ) を書き込むことによって、受信 FIFO からそのパケットが読み出されるまで、RXFLVL 割り込み (OTG\_GINTSTS レジスタ) をマスクすることができます。
3. 受信パケットのバイトカウントが 0 でない場合、データのバイトカウントは受信データ FIFO からポップされ、メモリに格納されます。受信パケットのバイトカウントが 0 の場合、受信データ FIFO からデータはポップされません。
4. 受信 FIFO のパケットステータスの読み出し結果は、以下のいずれかを示します。
  - a) グローバル OUT NAK のパターン  
PKTSTS = グローバル OUT NAK、BCNT = 0x000、EPNUM = (0x0)、DPID = (0b00)。これらのデータは、グローバル OUT NAK ビットが有効になっていることを示します。
  - b) SETUP パケットのパターン  
PKTSTS = SETUP、BCNT = 0x008、EPNUM = 制御 EP 番号、DPID = DATA0。これらのデータは、指定されたエンドポイントの SETUP パケットを受信 FIFO から読み出せることを示します。
  - c) セットアップステージ終了パターン  
PKTSTS = セットアップステージ終了、BCNT = 0x0、EPNUM = 制御 EP 番号、DPID = (0b00)。これらのデータは、指定されたエンドポイントのセットアップステージが完了し、データステージが開始されたことを示します。このエントリが受信 FIFO からポップされた後、コアは、指定された制御 OUT エンドポイントでセットアップ割り込みをアサートします。
  - d) データ OUT パケットのパターン  
PKTSTS = DataOUT、BCNT = 受信データ OUT パケットのサイズ ( $0 \leq BCNT \leq 1024$ )、EPNUM = パケットを受信した EPNUM、DPID = 実際のデータ PID。

- e) データ転送完了パターン  
 PKTSTS = データ OUT 転送終了、BCNT = 0x0、EPNUM = データ転送が完了した OUT EP 番号、DPID = (0b00)。  
 これらのデータは、指定された OUT エンドポイントの OUT データ転送が完了したことを示します。このエントリが受信 FIFO からポップされた後、コアは、指定された OUT エンドポイントで転送完了割り込みをアサートします。
- 5. データペイロードが受信 FIFO からポップされた後、RXFLVL 割り込み (OTG\_GINTSTS) がマスク解除されなければなりません。
- 6. 手順 1~5 は、アプリケーションが OTG\_GINTSTS レジスタの RXFLVL による割り込みラインのアサーションを検出するたびに繰り返されます。空の受信 FIFO を読み出すと、定義されていないコア動作を引き起こすことがあります。

図 461 上記の手順のフローチャートです。

図 461. 受信 FIFO パケット読み出し



## SETUP トランザクション

このセクションでは、コアが SETUP パケットを処理する方法と、SETUP トランザクションを処理するアプリケーションのシーケンスについて説明します。

### ● アプリケーションの要件

1. SETUP パケットを受信するには、制御 OUT エンドポイントの STUPCNT フィールド (OTG\_DOEPTSIZx レジスタ) が、ゼロでない値にプログラムされなければなりません。アプリケーションが STUPCNT フィールドをゼロでない値にプログラムすると、NAK のステータスと OTG\_DOEPTLx レジスタの EPENA ビットの設定に関係なく、コアは、SETUP パケットを受信して、受信 FIFO に書き込みます。STUPCNT フィールドは、制御エンドポイントが SETUP パケットを受信するたびにデクリメントされます。SETUP パケットを受信する前に、STUPCNT フィールドが適切な値にプログラムされていなかった場合、コアは SETUP パケットを受信し、

STUPCNT フィールドをデクリメントしますが、アプリケーションは制御転送のセットアップステージで受信した SETUP パケットの正しい数を判定できないことがあります。

- OTG\_DOEPTSIZE レジスタの STUPCNT = 3
2. 制御エンドポイントで 3 つまでの SETUP パケットを受信するためには、アプリケーションは、常に受信データ FIFO に余分のスペースを割り当てておく必要があります。
    - 確保すべきスペースは 10 ワード分です。最初の SETUP パケット用に 3 ワードが必要であり、セットアップステージ終了ワードのために 1 ワード、すべての制御エンドポイントにわたって 2 つの余分な SETUP パケットを格納するために 6 ワードが必要です。
    - 8 バイトの SETUP データと 4 バイトの SETUP ステータス(セットアップパケットパターン) を格納するには、SETUP パケットあたり 3 ワードが必要です。コアは、このスペースを受信データの中に確保します。
    - FIFO は SETUP データ書き込み専用であり、このスペースをデータパケットのために使うことはありません。
  3. アプリケーションは、受信 FIFO から SETUP パケットの 2 ワードを読み出す必要があります。
  4. アプリケーションは受信 FIFO からセットアップステージ終了ワードを読み出して、破棄する必要があります。

#### ● 内部データフロー

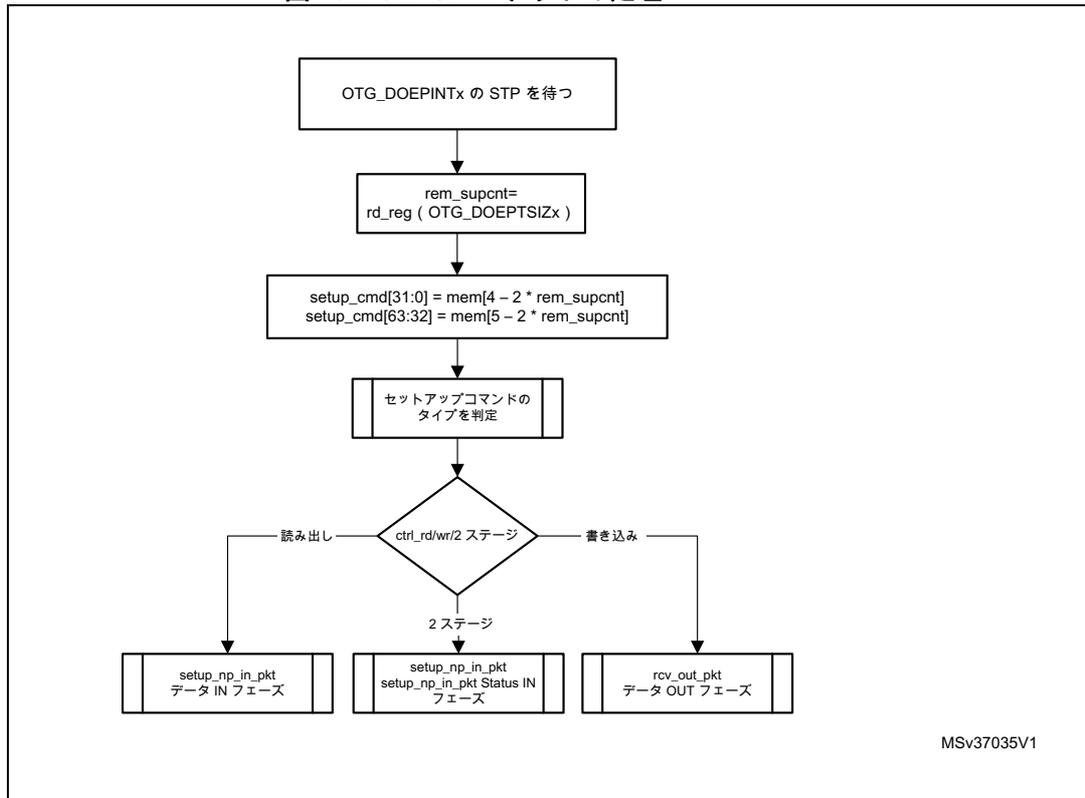
1. SETUP パケットが受信されると、コアは受信 FIFO 内の空きスペースをチェックせずに、また、エンドポイントの NAK および STALL ビットの設定に関係なく、受信したデータを受信 FIFO に書き込みます。
  - コアは、SETUP パケットが受信された制御 IN/OUT エンドポイントの IN NAK および OUT NAK ビットを内部でセットします。
2. USB 上で受信された各 SETUP パケットについて、3 ワードのデータが受信 FIFO に書き込まれ、STUPCNT フィールドが 1 ずつデクリメントされます。
  - 最初のワードは、コアが内部的に使用する制御情報を含みます。
  - 2 番目のワードは、SETUP コマンドの最初の 4 バイトを含みます。
  - 3 番目のワードは、SETUP コマンドの最後の 4 バイトを含みます。
3. セットアップステージがデータ IN/OUT ステージに変わると、コアは、エントリ(セットアップステージ終了ワード)を受信 FIFO に書き込んで、セットアップステージの完了を示します。
4. AHB 側では、SETUP パケットはアプリケーションによってエンプティにされます。
5. アプリケーションが受信 FIFO からセットアップステージ終了ワードをポップすると、コアは、STUP 割り込み (OTG\_DOEPINTx) でアプリケーションに割り込んで、受信した SETUP パケットを処理できることを示します。
6. コアは、制御 OUT エンドポイントのエンドポイントイネーブルビットをクリアします。

#### ● アプリケーションのプログラミングシーケンス :

1. OTG\_DOEPTSIZE レジスタをプログラムします。
  - STUPCNT = 3
2. RXFLVL 割り込み (OTG\_GINTSTS) を待ち、受信 FIFO からデータパケットをエンプティにします。
3. STUP 割り込み (OTG\_DOEPINTx) のアサートは、SETUP データ転送が正常に完了したことを示します。
  - この割り込み時、アプリケーションは、OTG\_DOEPTSIZE レジスタを読み出して、受信した SETUP パケットの数を確認し、最後に受信した SETUP パケットを処理する必要があります。



図 462. SETUP パケットの処理



### ● 3つを超える連続 SETUP パケットの処理

USB 2.0 仕様に従い、通常 SETUP パケットエラー時にはホストは同じエンドポイントに対して3つを超える連続 SETUP パケットを送信しません。ただし USB 2.0 仕様では、ホストが同じエンドポイントに送信できる連続 SETUP パケットの数を制限してはいません。この条件が発生すると OTG\_FS/OTG\_HS コントローラは、割り込み (OTG\_DOEPINTx の B2BSTUP) を生成します。

### ● グローバル OUT NAK の設定

内部データフロー :

1. アプリケーションがグローバル OUT NAK (OTG\_DCTL の SGONAK ビット) をセットすると、コアは SETUP パケットを除き、受信 FIFO へのデータの書き込みを停止します。受信 FIFO の使用可能なスペースの有無に関係なく、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信し、コアはアイソクロナス OUT データパケットを無視します。
2. コアは、グローバル OUT NAK パターンを受信 FIFO に書き込みます。アプリケーションがこのデータパターンを書き込むには、受信 FIFO に十分なスペースを確保する必要があります。
3. アプリケーションがグローバル OUT NAK パターンワードを受信 FIFO からポップすると、コアは、GONAKEFF 割り込み (OTG\_GINTSTS) をセットします。
4. この割り込みを検出したアプリケーションは、コアがグローバル OUT NAK モードにあるとみなすことができます。アプリケーションは OTG\_DCTL の SGONAK ビットをクリアすることによって、この割り込みをクリアできます。

アプリケーションのプログラミングシーケンス：

1. 受信 FIFO でのデータの受信も停止するには、アプリケーションは、以下のフィールドをプログラムすることによって、グローバル OUT NAK ビットをセットする必要があります。
  - OTG\_DCTL レジスタの SGONAK = 1
2. OTG\_GINTSTS の GONAKEFF 割り込みのアサートを待ちます。アサートされた場合、この割り込みは、コアが SETUP パケット以外のいかなるタイプのデータの受信も停止したことを示します。
3. アプリケーションは OTG\_DCTL の SGONAK ビットをセットした後、コアが GONAKEFF 割り込み (OTG\_GINTSTS) をアサートする前に、有効な OUT パケットを受信することができます。
4. アプリケーションは、OTG\_GINTMSK レジスタの GONAKEFFM ビットに書き込むことによって、この割り込みを一時的にマスクできます。
  - OTG\_GINTMSK レジスタの GONAKEFFM = 0
5. アプリケーションがグローバル OUT NAK モードを終了する準備ができたときには OTG\_DCTL レジスタの SGONAK ビットをクリアする必要があります。これによって GONAKEFF 割り込み (OTG\_GINTSTS) もクリアされます。
  - OTG\_DCTL レジスタの CGONAK = 1
6. アプリケーションがこの割り込みをマスクするのが早すぎた場合は、以下のようにしてマスク解除を行う必要があります。
  - OTG\_GINTMSK レジスタの GONAKEFFM = 1

## ● OUT エンドポイントの無効化

アプリケーションが有効にした OUT エンドポイントを無効にするには、このシーケンスを使用する必要があります。

アプリケーションのプログラミングシーケンス：

1. OUT エンドポイントを無効にする前に、アプリケーションは、コアのグローバル OUT NAK モードを有効にする必要があります。
  - OTG\_DCTL レジスタの SGONAK = 1
2. OTG\_GINTSTS の GONAKEFF 割り込みを待ちます。
3. 以下のフィールドをプログラムすることによって、必要な OUT エンドポイントを無効にします。
  - OTG\_DOEPCTLx レジスタの EPDIS = 1
  - OTG\_DOEPCTLx レジスタの SNAK = 1
4. EPDISD 割り込み (OTG\_DOEPINTx) を待ちます。これは OUT エンドポイントが完全に無効にされたことを示します。EPDISD 割り込みがアサートされると、コアは、以下のビットもクリアします。
  - OTG\_DOEPCTLx レジスタの EPDIS = 0
  - OTG\_DOEPCTLx レジスタの EPENA = 0
5. アプリケーションは、無効にされていない他のエンドポイントからのデータ受信を開始するために、グローバル OUT NAK ビットをクリアする必要があります。
  - OTG\_DCTL レジスタの SGONAK = 0

## ● 一般の非アイソクロナス OUT データ転送

このセクションでは、通常非アイソクロナス OUT データ転送（制御、バルク、または割り込み）について説明します。

アプリケーションの要件：

- OUT 転送をセットアップする前に、アプリケーションは OUT 転送の一部として受信されるすべてのデータを収容できるバッファをメモリ内で割り当てる必要があります。
- OUT 転送の場合、エンドポイントの転送サイズレジスタの転送サイズフィールドは、エンドポイントの最大パケットサイズの倍数でなければならず、ワードの境界に揃えられていなければなりません。
  - 転送サイズ [EPNUM] =  $n * (\text{MPSIZ}[\text{EPNUM}] + 4 - (\text{MPSIZ}[\text{EPNUM}] \bmod 4))$
  - パケットカウント [EPNUM] =  $n$
  - $n > 0$
- OUT エンドポイント割り込み時には、アプリケーションは、エンドポイントの転送サイズレジスタを読み出して、メモリ内のペイロードのサイズを計算しなければなりません。受信したペイロードのサイズが、プログラムされた転送サイズより小さいこともあります。
  - メモリ内のペイロードのサイズ = アプリケーションがプログラムした初期転送サイズ - コアが更新した最終転送サイズ
  - このペイロードが受信された USB パケットの数 = アプリケーションがプログラムした初期パケット数 - コアが更新した最終パケット数

内部データフロー：

- アプリケーションがデータを受信するためには、エンドポイント固有レジスタの転送サイズおよびパケットカウントのフィールドを設定し、NAK ビットをクリアし、エンドポイントを有効にする必要があります。
- NAK ビットがクリアされると、コアは、データの受信を開始し、受信 FIFO にスペースがある限り、データを受信 FIFO に書き込みます。USB で受信された各データパケットについて、データパケットとそのステータスが受信 FIFO に書き込まれます。受信 FIFO にパケット（最大パケットサイズまたはショートパケット）が書き込まれるたびに、そのエンドポイントのパケットカウントフィールドが 1 ずつデクリメントされます。
  - 受信された OUT データパケットのデータ CRC が不良な場合、受信 FIFO から自動的に一掃されます。
  - USB 上のパケットに対して ACK を送信した後、コアは ACK を検出できないホストが再送信する非アイソクロナス OUT データパケットを破棄します。アプリケーションは、同じエンドポイント上では、同じデータ PID を持つ複数の連続データ OUT パケットを検出しません。この場合、パケットカウントはデクリメントされません。
  - 受信 FIFO にスペースがない場合、アイソクロナスまたは非アイソクロナスデータパケットは無視され、受信 FIFO には書き込まれません。さらに、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信します。
  - 上記の 3 つのケースのすべてにおいて、データは受信 FIFO に書き込まれないので、パケットカウントはデクリメントされません。
- パケットカウントが 0 になるか、エンドポイント上でショートパケットが受信されると、そのエンドポイントの NAK ビットがセットされます。NAK ビットがセットされると、アイソクロナスまたは非アイソクロナスデータパケットは無視され、受信 FIFO には書き込まれず、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信します。
- データが受信 FIFO に書き込まれた後、アプリケーションは受信 FIFO からデータを読み出して、エンドポイントあたり一度に 1 パケットずつ外部メモリに書き込みます。

- AHB 上で外部メモリへのパケットの書き込みが終わるたびに、書き込まれたパケットのサイズだけエンドポイントの転送サイズがデクリメントされます。
- 以下の条件の 1 つで OUT エンドポイントの OUT データ転送完了パターンが受信 FIFO に書き込まれます。
  - 転送サイズが 0、およびパケットカウントが 0。
  - 受信 FIFO に書き込まれた最後の OUT データパケットがショートパケット。  
( $0 \leq \text{パケットサイズ} < \text{最大パケットサイズ}$ )
- アプリケーションがこのエントリ (OUT データ転送完了) をポップすると、エンドポイントの転送完了割り込みが生成され、エンドポイントイネーブルビットがクリアされます。

アプリケーションのプログラミングシーケンス：

- OTG\_DOEPTSIZEx レジスタで転送サイズおよび対応するパケットカウントをプログラムします。
- OTG\_DOEPCTLx レジスタをエンドポイントの特性でプログラムし、EPENA ビットおよび CNAK ビットをセットします。
  - OTG\_DOEPCTLx レジスタの EPENA = 1
  - OTG\_DOEPCTLx レジスタの CNAK = 1
- RXFLVL 割り込み (OTG\_GINTSTS) を待ち、受信 FIFO からデータパケットをエンプティにします。
  - この手順は、転送サイズに応じて何度でも繰り返すことができます。
- XFRC 割り込み (OTG\_DOEPINTx) のアサートは、非アイソクロナス OUT データ転送が正常に完了したことを示します。
- OTG\_DOEPTSIZEx レジスタを読み出して、受信したデータペイロードのサイズを確認します。

## ● 一般のアイソクロナス OUT データ転送

このセクションでは、通常のアイソクロナス OUT データ転送について説明します。

アプリケーションの要件：

- 非アイソクロナス OUT データ転送のアプリケーションの要件はすべて、アイソクロナス OUT データ転送にも適用されます。
- アイソクロナス OUT データ転送の場合、転送サイズおよびパケットカウントフィールドは、常に、単一フレームで受信できる最大パケットサイズのパケット数を設定しなければなりません。アイソクロナス OUT データ転送は、複数のフレームにまたがってははいけません。
- アプリケーションは、周期的フレームの終わり (OTG\_GINTSTS の EOPF 割り込み) の前に、受信 FIFO からすべてのアイソクロナス OUT データパケット (データおよびステータス) を読み出す必要があります。
- 次のフレームでデータを受信するには EOPF (OTG\_GINTSTS) の後と SOF (OTG\_GINTSTS) の前に、アイソクロナス OUT エンドポイントを有効にしなければなりません。

内部データフロー：

- アイソクロナス OUT エンドポイントの内部データフローは、非アイソクロナス OUT エンドポイントの内部データフローと基本的に同じですが、少し異なっているところがあります。
- エンドポイントイネーブルビットをセットし、NAK ビットをクリアすることによって、アイソクロナス OUT エンドポイントが有効にされたときには、偶数/奇数フレームビットも適切にセットされなければなりません。コアは、以下の条件が満たされた場合に限り、アイソクロナス OUT エンドポイント上で特定のフレームのデータを受信します。
  - EONUM (OTG\_DOEPCTLx レジスタ) = FNSOF[0] (OTG\_DSTS レジスタ)



- アプリケーションが受信 FIFO からアイソクロナス OUT データパケット(データとステータス)を完全に読み出すと、コアは OTG\_DOEPTSIZE レジスタの RXDPID フィールドを、受信 FIFO から読み出された最後のアイソクロナス OUT データパケットのデータ PID で更新します。

アプリケーションのプログラミングシーケンス：

- OTG\_DOEPTSIZE レジスタで転送サイズおよび対応するパケットカウントをプログラムします。
- OTG\_DOEPTCTL レジスタをエンドポイントの特性でプログラムし、エンドポイントイネーブル、ClearNAK、および偶数/奇数フレームの各ビットをセットします。
  - EPENA = 1
  - CNAK = 1
  - EONUM = (0 : 偶数 / 1 : 奇数)
- RXFLVL 割り込み (OTG\_GINTSTS) を待ち、受信 FIFO からのデータパケットをエンプティにします。
  - この手順は、転送サイズに応じて何度でも繰り返すことができます。
- XFRC 割り込み (OTG\_DOEPINTx) のアサートは、アイソクロナス OUT データ転送が正常に完了したことを示します。この割り込みは、必ずしもメモリ内のデータが良好であることを意味しません。
- この割り込みは、アイソクロナス OUT 転送で必ず検出されるとは限りません。その代わりに、アプリケーションは OTG\_GINTSTS で INCOMPISOOUT 割り込みを検出することができます。
- OTG\_DOEPTSIZE レジスタを読み出して、受信した転送データのサイズを確認し、フレームで受信したデータの有効性を確認します。アプリケーションは、以下の条件の 1 つが満たされた場合のみ、メモリに受信されたデータを有効として扱う必要があります。
  - RXDPID = DATA0 (OTG\_DOEPTSIZE レジスタ)、およびこのペイロードが受信された USB パケットの数 = 1
  - RXDPID = DATA1 (OTG\_DOEPTSIZE レジスタ)、およびこのペイロードが受信された USB パケットの数 = 2
  - RXDPID = D2 (OTG\_DOEPTSIZE レジスタ)、およびこのペイロードが受信された USB パケットの数 = 3[HS]  
このペイロードが受信された USB パケットの数 =  
アプリケーションがプログラムした初期パケット数 - コアが更新した最終パケット数  
アプリケーションは、無効なデータパケットを破棄できます。

## ● 不完全アイソクロナス OUT データ転送

このセクションでは、アイソクロナス OUT データパケットがコアの内部でドロップされたときのアプリケーションのプログラミングシーケンスについて説明します。

内部データフロー：

- アイソクロナス OUT エンドポイントの場合、XFRC 割り込み (OTG\_DOEPINTx) は必ずアサートされるわけではありません。コアがアイソクロナス OUT データパケットをドロップした場合、アプリケーションは以下の状況で XFRC 割り込み (OTG\_DOEPINTx) の検出に失敗することがあります。
  - 受信 FIFO が完全な ISO OUT データパケットを収容できない場合、コアは、受信したアイソクロナス OUT データをドロップします。
  - アイソクロナス OUT データパケットが CRC エラー付きで受信されたとき。
  - コアが受信したアイソクロナス OUT トークンが破損しているとき。
  - アプリケーションが受信 FIFO からデータを読み出すのに長時間かかっているとき。

2. コアが、すべてのアイソクロナス OUT エンドポイントへの転送を完了する前に周期的フレームの終わりを検出すると、不完全アイソクロナス OUT データ割り込み (OTG\_GINTSTS レジスタの INCOMPISOOUT) をアサートして、アイソクロナス OUT エンドポイントのうち少なくとも1つで XFRC 割り込み (OTG\_DOEPINTx) がアサートされていないことを示します。この時点で、不完全転送のエンドポイントは有効なままですが USB 上のこのエンドポイントでは、アクティブな転送は進行していません。

アプリケーションのプログラミングシーケンス：

1. INCOMPISOOUT 割り込み (OTG\_GINTSTS) のアサートは、現在のフレームで、少なくとも1つのアイソクロナス OUT エンドポイントが転送を完了していないことを示します。
2. アイソクロナス OUT データがエンドポイントから完全に出されていないためにこの割り込みが発生した場合、アプリケーションは処理を進める前に、受信 FIFO からすべてのアイソクロナス OUT データ (データとステータス) を出力する必要があります。
  - すべてのデータが受信 FIFO から出されると、アプリケーションは XFRC 割り込み (OTG\_DOEPINTx) を検出することができます。この場合、アプリケーションは、次のフレームでアイソクロナス OUT データを受信するには、エンドポイントを再び有効にする必要があります。
3. INCOMPISOOUT 割り込み (OTG\_GINTSTS) を受信すると、アプリケーションは、すべてのアイソクロナス OUT エンドポイントの制御レジスタ (OTG\_DOEPCTLx) を読み出して、現在のマイクロフレームにおいて転送を完了しなかったエンドポイントを確認する必要があります。次の両方の条件が満たされた場合、エンドポイント転送は完了しません。
  - EONUM ビット (OTG\_DOEPCTLx レジスタ) = FNSOF[0] (OTG\_DSTS レジスタ)
  - OTG\_DOEPCTLx レジスタの EPENA = 1
4. 現在のフレーム番号が変更されないように、上記の手順は SOF 割り込み (OTG\_GINTSTS) が検出される前に実行されなければなりません。
5. 不完全転送のアイソクロナス OUT エンドポイントの場合、アプリケーションは、メモリ内のデータを破棄し OTG\_DOEPCTLx レジスタの EPDIS ビットをセットすることによってエンドポイントを無効にする必要があります。
6. EPDISD 割り込み (OTG\_DOEPINTx) を待ち、次のフレームで新しいデータを受信するために、エンドポイントを有効にします。
  - コアがエンドポイントを無効にするには若干の時間がかかるので、アプリケーションは不良なアイソクロナスデータを受信した後、次のフレームのデータを受信できないことがあります。

## ● 非アイソクロナス OUT エンドポイントの停止

このセクションでは、アプリケーションが非アイソクロナスエンドポイントを停止する方法について説明します。

1. コアをグローバル OUT NAK モードにします。
2. 必要なエンドポイントを無効にします。
  - エンドポイントを無効にするときには、OTG\_DOEPCTL の SNAK ビットをセットする代わりに STALL = 1 にセットしてください (OTG\_DOEPCTL)。
  - STALL ビットは常に NAK ビットより優先されます。
3. アプリケーションがエンドポイントの STALL ハンドシェイクを終了する準備ができたときは、STALL ビット (OTG\_DOEPCTLx) をクリアしなければなりません。
4. アプリケーションが SetFeature.Endpoint Halt または ClearFeature.Endpoint Halt コマンドのためにエンドポイントの STALL ビットをセットまたはクリアする場合、STALL ビットは、アプリケーションが制御エンドポイントでステータスステージ転送をセットアップする前にセットまたはクリアされなければなりません。



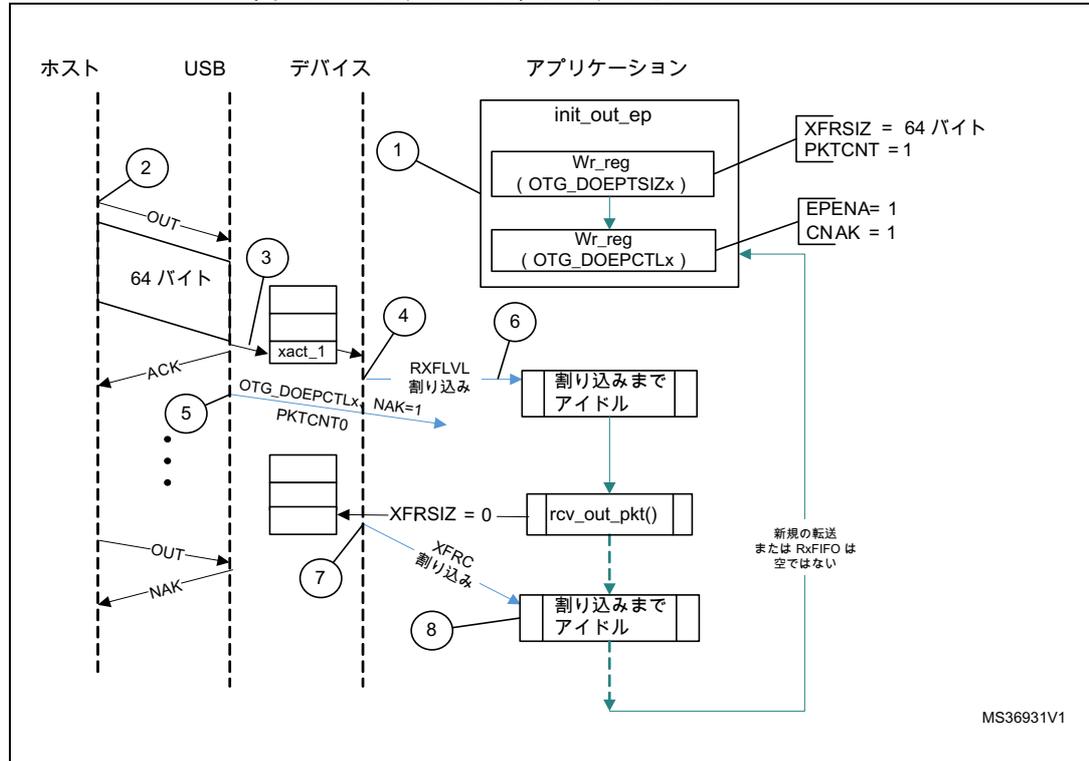
## 例

このセクションでは、いくつかの基本的な転送タイプとシナリオについて説明します。

- バルクOUTトランザクション

図 463 に USB から AHB への単一バルク OUT データパケットの受信と、このプロセスに伴うイベントを示します。

図 463. バルクOUTトランザクション



SetConfiguration/SetInterface コマンドの後、アプリケーションは CNAK = 1 および EPENA = 1 (OTG\_DOEPCTLx レジスタ) をセットし、OTG\_DOEPTSIZx レジスタの適切な XFRSIZ および PKTCNT ビットをセットすることによって、すべての OUT エンドポイントを初期化します。

1. ホストは、データ (OUT トークン) のエンドポイントへの送信を試みます。
2. コアは、USB 上で OUT トークンを受信すると、スペースが使用可能なので、Rx FIFO にパケットを格納します。
3. Rx FIFO に完全なパケットを書き込んだ後、コアは RXFLVL 割り込み (OTG\_GINTSTS) をアサートします。
4. USB パケットの PKTCNT 番号を受信すると、コアは、それ以上パケットが受信されないように、このエンドポイントの NAK ビットを内部でセットします。
5. アプリケーションは割り込みを処理して、Rx FIFO からデータを読み出します。
6. アプリケーションがすべてのデータ (XFRSIZ に相当) を読み出すと、コアは、XFRSIZ 割り込み (OTG\_DOEPINTx) を生成します。
7. アプリケーションは割り込みを処理して、XFRSIZ 割り込みビット (OTG\_DOEPINTx) の設定を使用して、意図した転送が完了したかどうかを確認します。

## IN データ転送

### ● パケットの書き込み

このセクションでは、専用の送信 FIFO が有効なときに、アプリケーションがデータパケットをエンドポイント FIFO に書き込む方法について説明します。

1. アプリケーションは、ポーリングまたは割り込みモードのいずれかを選択できます。
  - ポーリングモードでは、アプリケーションは、OTG\_DTXFSTSx レジスタを読み出すことによってエンドポイント送信データ FIFO のステータスを監視して、データ FIFO に十分なスペースがあるかどうかを確認します。
  - 割り込みモードでは、アプリケーションは、TXFE 割り込み (OTG\_DIEPINTx) を待ってから、OTG\_DTXFSTSx レジスタを読み出して、データ FIFO 内に十分なスペースがあるかどうかを確認します。
  - 単一の非ゼロ長データパケットを書き込むには、パケット全体を書き込むためのスペースがデータ FIFO になければなりません。
  - ゼロ長パケットを書き込むには、アプリケーションは FIFO のスペースを考慮してはなりません。
2. 上記の方法の 1 つを使用して、アプリケーションが送信パケットを書き込むのに十分なスペースがあることを確認するときには、アプリケーションは、データをデータ FIFO に書き込む前に、まず、エンドポイント制御レジスタに書き込む必要があります。通常、アプリケーションは、エンドポイントイネーブルビットをセットする場合を除き、レジスタの内容を変更しないように、OTG\_DIEPCTLx レジスタにリードモディファイライトを行う必要があります。

アプリケーションは、使用可能なスペースがあれば、同じエンドポイントに対する複数のパケットを送信 FIFO に書き込むことができます。周期的 IN エンドポイントの場合、アプリケーションは 1 つのマイクロフレームのパケットのみを書き込む必要があります。アプリケーションは、前のトランザクションの転送完了割り込みを受信した後でのみ、次の周期的トランザクションのパケットを書き込むことができます。

### ● IN エンドポイント NAK の設定

内部データフロー :

1. アプリケーションが特定のエンドポイントの IN NAK をセットすると、コアは、エンドポイントの送信 FIFO にデータがあるかどうかに関係なく、そのエンドポイントでのデータ送信を停止します。
2. 非アイソクロナス IN トークンは NAK ハンドシェイク応答を受信します。
  - アイソクロナス IN トークンはゼロデータ長のパケット応答を受信します。
3. コアは、OTG\_DIEPCTLx の SNAK ビットに応答して、OTG\_DIEPINTx の INEPNE 割り込み (IN エンドポイント NAK 有効) をアサートします。
4. この割り込みがアプリケーションによって検出されると、アプリケーションは、エンドポイントが IN NAK モードにあるとみなすことができます。この割り込みは、アプリケーションが OTG\_DIEPCTLx の CNAK ビットをセットすることによってクリアできます。

アプリケーションのプログラミングシーケンス：

1. 特定の IN エンドポイントでのデータ送信を停止するには、アプリケーションは、IN NAK ビットをセットする必要があります。このビットをセットするには、以下のフィールドをプログラムする必要があります。
  - OTG\_DIEPCTLx レジスタの SNAK = 1
2. OTG\_DIEPINTx の INEPNE 割り込みのアサートを待ちます。この割り込みは、コアがエンドポイントでのデータ送信を停止したことを示します。
3. コアは、アプリケーションが NAK ビットをセットした後の NAK 有効割り込みがアサートされる前に、エンドポイントで有効な IN データを送信できます。
4. アプリケーションは OTG\_DIEPMSK の INEPNEM ビットに書き込むことによって、この割り込みを一時的にマスクできます。
  - OTG\_DIEPMSK レジスタの INEPNEM = 0
5. エンドポイント NAK モードを終了するには、アプリケーションは OTG\_DIEPCTLx の NAK ステータスビット (NAKSTS) をクリアする必要があります。これによって INEPNE 割り込み (OTG\_DIEPINTx) もクリアされます。
  - OTG\_DIEPCTLx レジスタの CNAK = 1
6. アプリケーションがこの割り込みをマスクするのが早すぎた場合、以下のようにしてマスク解除を行う必要があります。
  - OTG\_DIEPMSK レジスタの INEPNEM = 1

## ● IN エンドポイントディセーブル

以前に有効にされた特定の IN エンドポイントを無効にするには、以下のシーケンスを使用します。

アプリケーションのプログラミングシーケンス：

1. アプリケーションは、AHB でのデータ書き込みを停止して、IN エンドポイントを無効にする必要があります。
2. アプリケーションは、エンドポイントを NAK モードに設定する必要があります。
  - OTG\_DIEPCTLx レジスタの SNAK = 1
3. OTG\_DIEPINTx の INEPNE 割り込みを待ちます。
4. 無効にしなければならないエンドポイントについて OTG\_DIEPCTLx レジスタの以下のビットをセットします。
  - OTG\_DIEPCTLx レジスタの EPDIS = 1
  - OTG\_DIEPCTLx レジスタの SNAK = 1
5. OTG\_DIEPINTx レジスタの EPDISD 割り込みのアサートは、コアが指定されたエンドポイントを完全に無効にしたことを示します。割り込みのアサートとともに、コアは、以下のビットもクリアします。
  - OTG\_DIEPCTLx レジスタの EPENA = 0
  - OTG\_DIEPCTLx レジスタの EPDIS = 0
6. アプリケーションは、周期的 IN エンドポイントの OTG\_DIEPTSIZx レジスタを読み出して、エンドポイント上のどれだけのデータが USB で送信されたかを計算する必要があります。
7. アプリケーションは、OTG\_GRSTCTL レジスタの以下のフィールドを設定することによって、エンドポイント送信 FIFO 内のデータを一掃する必要があります。
  - TXFNUM (OTG\_GRSTCTL) = エンドポイント送信 FIFO 番号
  - TXFFLSH (OTG\_GRSTCTL) = 1

アプリケーションは、TXFFLSH ビットがコアによってクリアされるまで（すなわち、一掃動作の終了まで）、OTG\_GRSTCTL レジスタにポーリングする必要があります。このエンドポイントで新しいデータを送信するために、アプリケーションは後で、エンドポイントを再び有効にできます。

## ● 一般の非周期的 IN データ転送

アプリケーションの要件：

1. IN 転送をセットアップする前に、アプリケーションは、IN 転送の一部として送信されるすべてのデータが単一バッファの一部であることを確認する必要があります。
2. IN 転送の場合、エンドポイント転送サイズレジスタの転送サイズフィールドは、最大パケットサイズの複数のパケットと単一のショートパケットから成るペイロードを表します。このショートパケットは、転送の最後に送信されます。
  - 転送の終わりに少数の最大パケットサイズのパケットとショートパケットを送信するには：
 
$$\text{転送サイズ [EPNUM]} = x * \text{MPSIZ[EPNUM]} + sp$$
 (sp > 0) の場合、パケットカウント [EPNUM] = x + 1。  
 そうでない場合、パケットカウント [EPNUM] = x
  - 単一のゼロ長データパケットを送信するには：
 
$$\text{転送サイズ [EPNUM]} = 0$$

$$\text{パケットカウント [EPNUM]} = 1$$
  - 転送の終わりに少数の最大パケットサイズのパケットとゼロ長データパケットを送信するには、アプリケーションは転送を 2 つの部分に分ける必要があります。最初の部分では最大パケットサイズのデータパケットを送信し、2 番目の部分ではゼロ長データパケットのみを送信します。
 
$$\text{最初の転送：転送サイズ [EPNUM]} = x * \text{MPSIZ[epnum]} ; \text{パケットカウント} = n ;$$

$$\text{2 番目の転送：転送サイズ [EPNUM]} = 0 ; \text{パケットカウント} = 1 ;$$
3. エンドポイントがデータ転送のために有効にされると、コアは、転送サイズレジスタを更新します。IN 転送の終了時に、アプリケーションは、転送サイズレジスタを読み出して、送信 FIFO にポストされたデータのうち、どれだけが USB で送信されたかを確認する必要があります。
4. 送信 FIFO にフェッチされたデータ = アプリケーションがプログラムした初期転送サイズ - コアが更新した最終転送サイズ
  - USB で送信されたデータ = (アプリケーションがプログラムした初期パケットカウント - コアが更新した最終パケットカウント) \* MPSIZ[EPNUM]
  - USB 上でこれから送信されるデータ = (アプリケーションがプログラムした初期転送サイズ - USB で送信されたデータ)

内部データフロー：

1. アプリケーションは、エンドポイント固有レジスタの転送サイズおよびパケット数フィールドを設定して、データを送信するためにエンドポイントを有効にする必要があります。
2. アプリケーションは、要求されたデータをエンドポイントの送信 FIFO に書き込む必要もありません。
3. アプリケーションによってパケットが送信 FIFO に書き込まれるたびに、そのエンドポイントの転送サイズがパケットのサイズだけデクリメントされます。エンドポイントの転送サイズが 0 になるまで、データはアプリケーションによってメモリからフェッチされます。FIFO にデータを書き込んだ後、「FIFO 内のパケット数」はインクリメントされます（これは 3 ビットのカウンタであり、各 IN エンドポイントの送信 FIFO について、コアによって内部で維持されます。IN エンドポイント FIFO 内にコアによって一度に維持されるパケットの最大数は 8 です）。ゼロ長のパケットの場合、各 FIFO について、FIFO 内にデータがないことを示す個別のフラグがセットされます。



4. データが送信 FIFO に書き込まれると、コアは、IN トークンの受信時に読み出します。非アイソクロナス IN データパケットが ACK ハンドシェイクとともに送信されるたびに、エンドポイントのパケットカウンタは、0 になるまで 1 ずつデクリメントされます。パケットカウンタは、タイムアウト時にはデクリメントされません。
5. ゼロ長パケットの場合（内部ゼロ長フラグで示されます）、コアは IN トークンのゼロ長パケットを送出し、パケットカウンタフィールドをデクリメントします。
6. 受信された IN トークンのデータが FIFO になく、そのエンドポイントのパケットカウンタフィールドがゼロの場合、コアは、そのエンドポイントについて、「Tx FIFO がエンpty時に受信された IN トークン」割り込み (ITTXFE 割り込み) を生成します。ただし、エンドポイントの NAK ビットがセットされていない場合に限りです。コアは、USB 上の非アイソクロナスエンドポイントに対して NAK ハンドシェイクで応答します。
7. コアは、内部で FIFO ポインタを巻き戻し、タイムアウト割り込みは生成されません。
8. 転送サイズが 0 であり、パケットカウンタが 0 のとき、エンドポイントの転送完了割り込み (XFRC) が生成され、エンドポイント有効化ビットがクリアされます。

アプリケーションのプログラミングシーケンス：

1. OTG\_DIEPTSIZx レジスタで転送サイズと対応するパケットカウンタをプログラムします。
2. OTG\_DIEPCTLx レジスタをエンドポイントの特性でプログラムし、CNAK ビットおよび EPENA (エンドポイントイネーブル) ビットをセットします。
3. 非ゼロ長データパケットを送信するときには、アプリケーションは、OTG\_DTXFSTSx レジスタ (x はそのエンドポイントに関連する FIFO 番号) をポーリングして、データ FIFO に十分なスペースがあるかどうかを確認する必要があります。アプリケーションは、データを書き込む前に、オプションで TXFE (OTG\_DIEPINTx) を使用することができます。

## ● 一般の周期的 IN データ転送

このセクションでは、典型的な周期的 IN データ転送について説明します。

アプリケーションの要件：

1. **一般の非周期的 IN データ転送 (1459 ページ)** のアプリケーションの要件 1、2、3、および 4 は、要件 2 が少し変更されていることを除いて、周期的 IN データ転送にも適用されます。
  - アプリケーションは複数の最大パケットサイズのデータパケットの送信、または複数の最大パケットサイズのパケットに最後にショートパケットを加えた送信のみを行うことができます。転送の終わりに少数の最大パケットサイズのパケットとショートパケットを送信するには、以下の条件が満たされなければなりません。
 
$$\text{転送サイズ [EPNUM]} = x * \text{MPSIZ[EPNUM]} + \text{sp}$$
 (ここで x は、整数  $\geq 0$  および  $0 \leq \text{sp} < \text{MPSIZ[EPNUM]}$ )
   
(sp > 0) の場合、パケットカウンタ [EPNUM] = x + 1
   
そうでない場合、パケットカウンタ [EPNUM] = x ;
   
 $\text{MCNT[EPNUM]} = \text{パケットカウンタ [EPNUM]}$
  - アプリケーションは転送の最後にゼロ長データパケットを送信することはできません。自分で単一のゼロ長パケットを送信することは可能です。単一のゼロ長データパケットを送信するには：
    - 転送サイズ [EPNUM] = 0
   
パケットカウンタ [EPNUM] = 1
   
 $\text{MCNT[EPNUM]} = \text{パケットカウンタ [EPNUM]}$

2. アプリケーションは一度に 1 フレームのデータ転送のみをスケジュールすることができます。
  - $(MCNT - 1) * MPSIZ \leq XFERSIZ \leq MCNT * MPSIZ$
  - $PKTCNT = MCNT$  (OTG\_DIEPTSIZE レジスタ)
  - $XFERSIZ < MCNT * MPSIZ$  の場合、転送の最後のデータパケットはショートパケットです。
  - 以下のことに注意してください。MCNT は OTG\_DIEPTSIZE レジスタに、MPSIZ は OTG\_DIEPTCTL レジスタに、PKTCNT は OTG\_DIEPTSIZE レジスタに、また XFERSIZ は OTG\_DIEPTSIZE レジスタにあります。
3. フレームで送信される完全なデータは、IN トークンが受信される前に、アプリケーションによって送信 FIFO に書き込まれなければなりません。IN トークンが受信されたときに、フレームあたりで送信されるデータの 1 ワードが送信 FIFO 内で欠落していても、コアは FIFO がエンプティである場合のように動作します。送信 FIFO がエンプティのとき：
  - アイソクロナス IN エンドポイントに対して USB 上でゼロデータ長のパケットが送信されません。
  - IN エンドポイントに割り込みをかけるため、USB 上で NAK ハンドシェイクが送信されません。

#### 内部データフロー：

1. アプリケーションは、エンドポイント固有レジスタの転送サイズおよびパケットカウントフィールドを設定して、データを送信するためにエンドポイントを有効にする必要があります。
2. また、アプリケーションは、要求されたデータをエンドポイントの関連する送信 FIFO に書き込む必要があります。
3. アプリケーションがパケットを送信 FIFO に書き込むたびに、そのエンドポイントの転送サイズがパケットのサイズだけデクリメントされます。エンドポイントの転送サイズが 0 になるまで、アプリケーションメモリからデータがフェッチされます。
4. 周期的エンドポイントの IN トークンが受信されると、コアは、FIFO のデータ（ある場合）を送信します。フレームの完全なデータペイロード（専用 FIFO モードでは完全なパケット）が FIFO に存在しない場合、コアは、エンドポイントに対する Tx FIFO エンプティ割り込みが受信されたときに IN トークンを生成します。
  - アイソクロナス IN エンドポイントに対して、USB 上でゼロ長のデータパケットが送信されます。
  - IN エンドポイントに割り込みをかけるため、USB 上で NAK ハンドシェイクが送信されません。
5. エンドポイントのパケットカウントは、以下の条件のもとで 1 ずつデクリメントされます。
  - アイソクロナスエンドポイントに対して、ゼロ長または非ゼロ長データパケットが送信されたとき。
  - 割り込みエンドポイントに対して、ACK ハンドシェイクが送信されたとき。
  - 転送サイズとパケットカウントの両方が 0 のとき、エンドポイントに対して転送完了割り込みが生成され、エンドポイントイネーブルビットがクリアされます。
6. 「周期的フレームインターバル」(OTG\_DCFG の PFIVL によって制御されます) で、コアが現在のフレームについてスケジュールされたアイソクロナス IN エンドポイント FIFO のいずれかが非エンプティであることを検出すると、コアは OTG\_GINTSTS の IISOIXFR 割り込みを生成します。

アプリケーションのプログラミングシーケンス：

1. OTG\_DIEPCTLx レジスタをエンドポイントの特性でプログラムして、CNAK および EPENA ビットをセットします。
2. 次のフレームで送信するデータを送信 FIFO に書き込みます。
3. OTG\_DIEPINTx レジスタの ITTXFE 割り込みのアサートは、アプリケーションが送信するすべてのデータを送信 FIFO にまだ書き込んでいないことを示します。
4. この割り込みが検出されたときに、割り込みエンドポイントがすでに有効だった場合、割り込みは無視されます。まだ有効になっていなかった場合は、次の IN トークン試行の際にデータを送信できるように、エンドポイントを有効にします。
5. OTG\_DIEPINTx レジスタの ITTXFE 割り込みがない状態での XFRC 割り込み (OTG\_DIEPINTx) のアサートは、アイソクロナス IN 転送が正常に完了したことを示します。OTG\_DIEPTSIZx レジスタの読み出し結果は、転送サイズ = 0、かつパケットカウント = 0、すなわち、すべてのデータが USB で送信されたことを示さなければなりません。
6. ITTXFE 割り込み (OTG\_DIEPINTx) の有無にかかわらず、XFRC 割り込み (OTG\_DIEPINTx) のアサートは、割り込み IN 転送が正常に完了したことを示します。OTG\_DIEPTSIZx レジスタの読み出し結果は、転送サイズ = 0、かつパケットカウント = 0、すなわち、すべてのデータが USB で送信されたことを示さなければなりません。
7. 前述のどの割り込みもない状態での OTG\_GINTSTS の不完全アイソクロナス IN 転送 (IISOIXFR) 割り込みのアサートは、コアが現在のフレームで少なくとも 1 つの周期的 IN トークンも受信しなかったことを示します。

## ● 不完全アイソクロナス IN データ転送

このセクションでは、アプリケーションが不完全なアイソクロナス IN データ転送で行う必要があることについて説明します。

内部データフロー：

1. アイソクロナス IN 転送は、以下の条件のいずれかがあてはまる場合に不完全として扱われます。
  - a) コアが、少なくとも 1 つのアイソクロナス IN エンドポイントで破損したアイソクロナス IN トークンを受信した場合。この場合、アプリケーションは、不完全アイソクロナス IN 転送割り込み (OTG\_GINTSTS の IISOIXFR ビット) を検出します。
  - b) アプリケーションが完全なデータペイロードを送信 FIFO に書き込むのに時間がかかり、完全なデータペイロードが FIFO に書き込まれる前に IN トークンが受信された場合。この場合、アプリケーションは OTG\_DIEPINTx の Tx FIFO エンプティ割り込み時に IN トークンの受信を検出します。最終的には、周期的フレームの終わりでの不完全アイソクロナス IN 転送割り込み (OTG\_GINTSTS の IISOIXFR) になるので、アプリケーションはこの割り込みを無視することができます。  
コアは、受信した IN トークンにตอบสนองして、USB 上にゼロ長データパケットを送信します。
2. アプリケーションは、できるだけ速やかに、送信 FIFO へのデータペイロードの書き込みを停止する必要があります。
3. アプリケーションは、エンドポイントの NAK ビットと無効化ビットをセットする必要があります。
4. コアは、エンドポイントを無効にし、無効化ビットをクリアし、エンドポイントのエンドポイントディセーブル割り込みをアサートします。

アプリケーションのプログラミングシーケンス：

1. 最終的には不完全アイソクロナス IN 転送割り込み (OTG\_GINTSTS) になるので、アプリケーションは、アイソクロナス IN エンドポイントで OTG\_DIEPINTx の Tx FIFO エンプティ割り込み時に受信された IN トークンを無視することができます。

2. 不完全アイソクロナス IN 転送割り込み (OTG\_GINTSTS) のアサートは、少なくとも 1 つのアイソクロナス IN エンドポイントで不完全アイソクロナス IN 転送があったことを示します。
3. アプリケーションは、すべてのアイソクロナス IN エンドポイントのエンドポイント制御レジスタを読み出して、不完全 IN データ転送があるエンドポイントを検出する必要があります。
4. アプリケーションは、AHB 上でこれらのエンドポイントに関連する周期的送信 FIFO へのデータの書き込みを停止する必要があります。
5. OTG\_DIEPCTLx レジスタの以下のフィールドをプログラムして、エンドポイントを無効にします。
  - OTG\_DIEPCTLx レジスタの SNAK = 1
  - OTG\_DIEPCTLx レジスタの EPDIS = 1
6. OTG\_DIEPINTx のエンドポイントディセーブル割り込みのアサートは、コアがエンドポイントを無効にしたことを示します。
  - この時点で、アプリケーションは、次のマイクロフレームで新しい転送を行うために、関連する送信 FIFO 内のデータを一掃するか、エンドポイントを有効にすることによって、FIFO 内の既存のデータに上書きする必要があります。データを一掃するには、アプリケーションは OTG\_GRSTCTL レジスタを使用する必要があります。

## ● 非アイソクロナス IN エンドポイントの停止

このセクションでは、アプリケーションが非アイソクロナスエンドポイントを停止する方法について説明します。

アプリケーションのプログラミングシーケンス：

1. 停止する IN エンドポイントを無効にします。STALL ビットもセットします。
2. エンドポイントがすでに有効になっているときには、OTG\_DIEPCTLx の EPDIS = 1 にします。
  - OTG\_DIEPCTLx レジスタの STALL = 1 にします。
  - STALL ビットは常に NAK ビットより優先されます。
3. エンドポイントディセーブル割り込み (OTG\_DIEPINTx) のアサートは、コアが指定されたエンドポイントを無効にしたことをアプリケーションに示します。
4. アプリケーションは、エンドポイントのタイプに応じて、非周期的または周期的 FIFO を一掃する必要があります。非周期的エンドポイントの場合、アプリケーションは、データを送信するために、停止する必要がない他の非周期的エンドポイントを再び有効にする必要があります。
5. アプリケーションがエンドポイントの STALL ハンドシェイクを終了する準備ができたときには、OTG\_DIEPCTLx の STALL ビットがクリアされなければなりません。
6. アプリケーションが SetFeature.Endpoint Halt コマンドまたは ClearFeature.Endpoint Halt コマンドのためにエンドポイントの STALL ビットをセットまたはクリアする場合、STALL ビットは、アプリケーションが制御エンドポイントでステータスステージ転送をセットアップする前にセットまたはクリアされなければなりません。

特殊なケース：制御 OUT エンドポイントの停止

制御転送のデータステージで、ホストが SETUP パケットで指定されたより多くの IN/OUT トークンを送信した場合、コアは、IN/OUT トークンを停止する必要があります。この場合、アプリケーションは、コアが SETUP パケットで指定されたデータ量を転送した後、制御転送のデータステージで、OTG\_DIEPINTx の ITTXFE 割り込みと OTG\_DOEPINTx の OTEPDIS 割り込みを有効にしなければなりません。その後、アプリケーションがこの割り込みを受信したとき、アプリケーションは、対応するエンドポイント制御レジスタの STALL ビットをセットし、この割り込みをクリアする必要があります。

## 37.16.7 最悪ケースの応答時間

OTG\_FS/OTG\_HS コントローラがデバイスとして機能するとき、アイソクロナス OUT に続く任意のトークンで、最悪ケースの応答時間があります。この最悪ケース応答時間は、AHB クロック周波数に依存します。

コアのレジスタは AHB ドメインにあり、コアはこれらのレジスタを更新するまでは、別のトークンを受け入れません。アイソクロナストラザクションの場合、ハンドシェイクはなく、次のトークンがすぐに入ってくる可能性があるため、アイソクロナス OUT に続くトークンで最悪ケースが生じます。この最悪ケース値は、AHB クロックが PHY クロックと同じときには 7 PHY クロックです。AHB クロックが高速なほど、この値は小さくなります。

この最悪ケース条件が発生した場合、コアは、バルク/割り込みトークンに NAK で応答し、アイソクロナスおよび SETUP トークンをドロップします。ホストは、これを SETUP のタイムアウト条件と解釈して、SETUP パケットを再実行します。アイソクロナス転送の場合、不完全アイソクロナス IN 転送割り込み (IISOIXFR) と不完全アイソクロナス OUT 転送割り込み (IISOXFR) は、アイソクロナス IN/OUT パケットがドロップされたことをアプリケーションに知らせます。

### OTG\_GUSBCFG の TRDT の値の選択

TRDT (OTG\_GUSBCFG) の値は、MAC が IN トークンを受信した後、FIFO ステータスと PFC ブロックから最初のデータを取得するまでの時間 (PHY クロック数) です。この時間には、PHY クロックと AHB クロック間の同期遅延も含まれます。この場合の最悪ケース遅延は、AHB クロックが PHY クロックと同じときに発生します。この場合、遅延は 5 クロックです。

MAC が IN トークンを受信すると、この情報 (トークンの受信) は PFC (PFC は AHB クロックで動作します) によって AHB に同期されます。次に、PFC は、SPRAM からデータを読み出し、それらをデュアルクロックソースバッファに書き込みます。MAC は、データをソースバッファ (深さ 4) から読み出します。

AHB が PHY より高い周波数で動作している場合、アプリケーションは TRDT (OTG\_GUSBCFG) の値として、より小さな値を使用することができます。

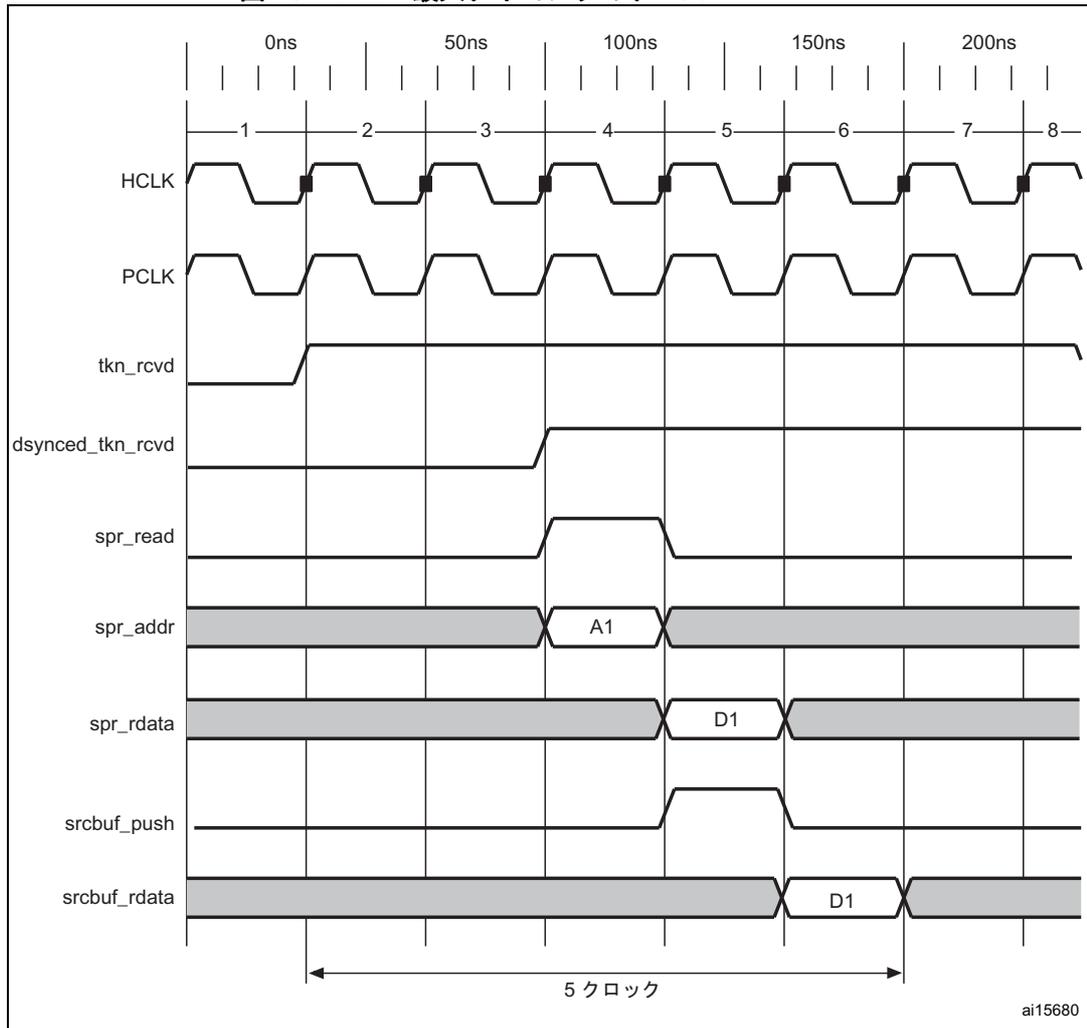
 464 に以下の信号を示します。

- tkn\_rcvd : MAC から PFC への、トークンが受信されたという情報
- dynccd\_tkn\_rcvd : PCLK から HCLK ドメインへの二重同期 tkn\_rcvd
- spr\_read : SPRAM への読み出し
- spr\_addr : SPRAM へのアドレス
- spr\_rdata : SPRAM からのデータの読み出し
- srcbuf\_push : ソースバッファへのプッシュ
- srcbuf\_rdata : ソースバッファからのデータの読み出し MAC によるデータの検出

アプリケーションは次の式を使用して TRDT の値を計算します。

$$4 * \text{AHB クロック} + 1 \text{ PHY クロック} = (2 \text{ クロック同期} + 1 \text{ クロックメモリアドレス} + \text{同期 RAM からの} 1 \text{ クロックメモリデータ}) + 1 \text{ PHY クロック (次の PHY クロック MAC は 2 クロック FIFO 出力をサンプリングすることができます。)}$$

図 464. TRDT 最大タイミングのケース



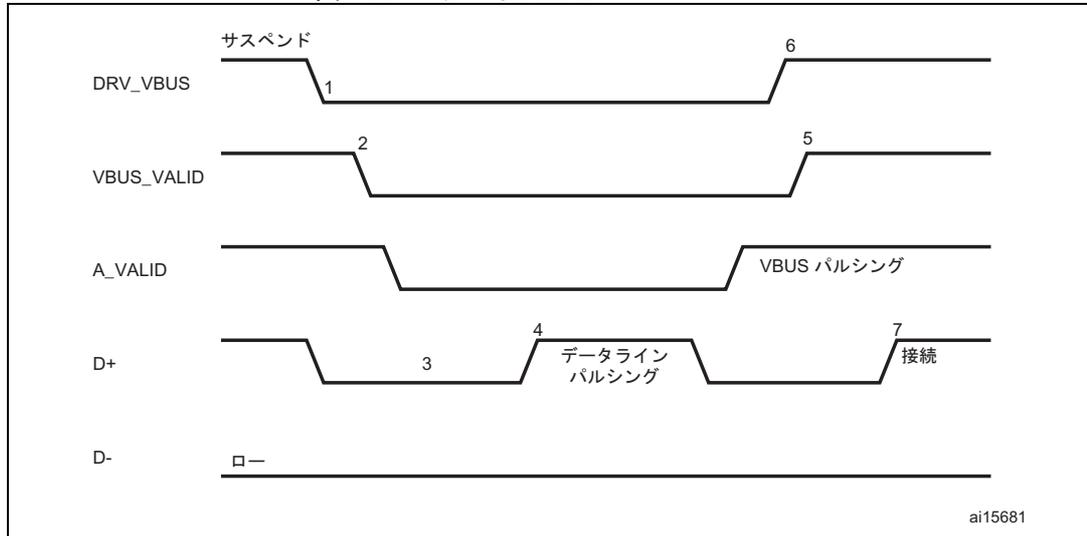
## 37.16.8 OTG プログラミングモデル

OTG\_FS/OTG\_HS コントローラは、HNP と SRP をサポートする OTG デバイスです。コアが A プラグに接続されると、A デバイスと呼ばれます。コアが B プラグに接続されると、B デバイスと呼ばれます。ホストモードでは、OTG\_FS/OTG\_HS コントローラは、電力を節約するために  $V_{BUS}$  をオフにします。SRP は、B デバイスが A デバイスに  $V_{BUS}$  電源をオンにするように信号を送る手段です。デバイスは、データラインパルシングと  $V_{BUS}$  パルシングの両方を実行する必要がありますが、ホストは SRP のデータラインパルシングか  $V_{BUS}$  パルシングのどちらかを検出することができます。HNP は、B デバイスがネゴシエイトし、役割をホストに切り替える手段です。HNP 後のネゴシエイトモードでは、B デバイスはバスをサスペンドし、その役割をデバイスに戻します。

## A デバイスセッションリクエストプロトコル

アプリケーションは、コア USB 設定レジスタの SRP 対応ビットをセットする必要があります。これによって、OTG\_FS/OTG\_HS コントローラは SRP を A デバイスとして検出できるようになります。

図 465. A デバイスの SRP

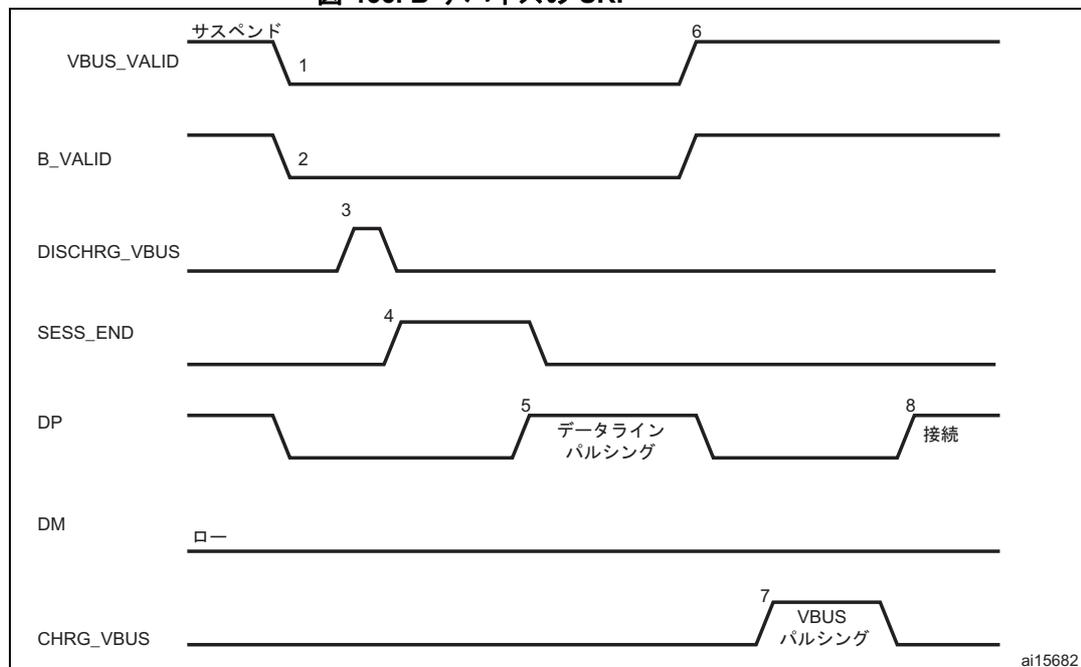


1. DRV\_VBUS = PHY への  $V_{BUS}$  駆動信号  
VBUS\_VALID = PHY からの  $V_{BUS}$  有効信号  
A\_VALID = PHY への A ペリフェラル  $V_{BUS}$  レベル信号  
D+ = データ正ライン  
D- = データ負ライン
1. 節電のためには、アプリケーションは、バスがアイドル状態のときに、ホストポート制御およびステータスレジスタのポートサスペンドビットおよびポートパワービットに書き込むことによって、ポート電源をサスペンドおよびオフにします。
2. PHY は、VBUS\_VALID 信号をネゲートすることによって、ポート電源のオフを示します。
3. デバイスは、 $V_{BUS}$  電源がオフになると、SRP を開始するために少なくとも 2 ms 以上 SE0 を検出しなければなりません。
4. SRP を開始するには、デバイスはデータラインプルアップ抵抗を 5~10 ms の間オンにします。OTG\_FS/OTG\_HS コントローラはデータラインのパルシングを検出します。
5. デバイスは、 $V_{BUS}$  を A デバイスセッション有効レベル以上（最低 2.0 V）に駆動します（ $V_{BUS}$  パルシングに対して）。  
OTG\_FS/OTG\_HS コントローラは、SRP を検出すると、アプリケーションに割り込みをかけます。グローバル割り込みステータスレジスタのセッションリクエスト検出ビットがセットされます（OTG\_GINTSTS の SRQINT）。
6. アプリケーションは、セッションリクエスト検出割り込みを処理して、ホストポート制御およびステータスレジスタのポートパワービットに書き込むことによって、ポート電源をオンにする必要があります。PHY は、VBUS\_VALID 信号をアサートすることによって、ポート電源がオンになったことを示します。
7. USB に電源が投入されると、デバイスが接続され、SRP プロセスが完了します。

## B デバイスセッションリクエストプロトコル

アプリケーションは、コア USB 設定レジスタの SRP 対応ビットをセットする必要があります。これによって、OTG\_FS/OTG\_HS コントローラは SRP を B デバイスとして開始することができます。SRP は、OTG\_FS/OTG\_HS コントローラがホストからの新しいセッションをリクエストする手段です。

図 466. B デバイスの SRP



- 1. VBUS\_VALID = PHY からの V<sub>BUS</sub> 有効信号  
B\_VALID = PHY への B ペリフェラル有効セッション信号  
DISCHRG\_VBUS = PHY への放電信号  
SESS\_END = PHY へのセッション終了信号  
CHRGR\_VBUS = PHY への V<sub>BUS</sub> 充電信号  
DP = データ正ライン  
DM = データ負ライン

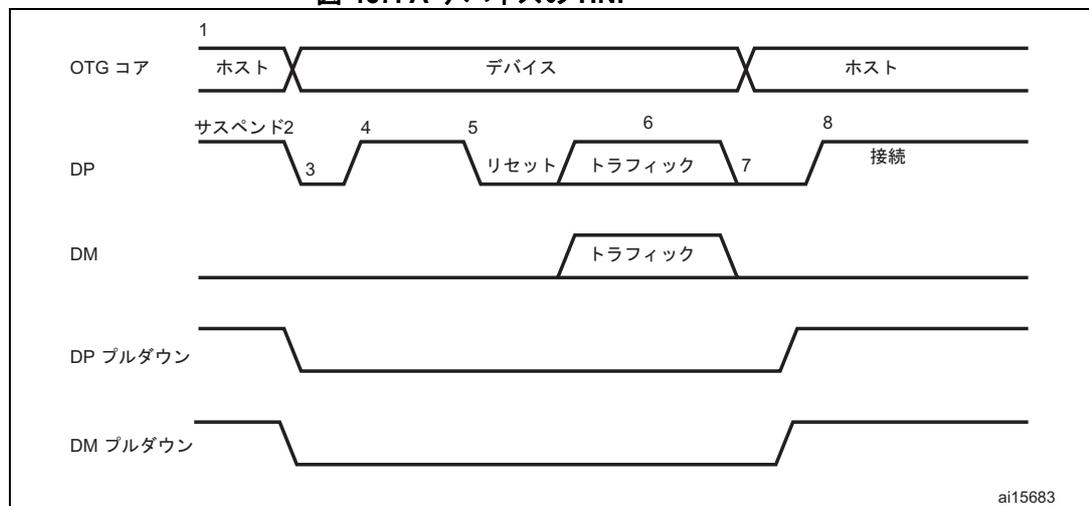
1. 節電のために、ホストはバスがアイドル状態のときはポート電源をサスペンドおよびオフにします。  
OTG\_FS/OTG\_HS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割り込みレジスタのアーリーサスペンドビットをセットします。これに続いて OTG\_FS/OTG\_HS コントローラは、コア割り込みレジスタの USB サスペンドビットをセットします。  
OTG\_FS/OTG\_HS コントローラは、PHY に V<sub>BUS</sub> を放電するように通知します。
2. PHY はデバイスにセッションの終了を知らせます。これが SRP の初期条件です。OTG\_FS/OTG\_HS コントローラは、SRP を初期化する前に 2 ms の SE0 を必要とします。  
USB 1.1 フルスピードシリアルトランシーバの場合、アプリケーションは、BSVLD (OTG\_GOTGCTL) がネゲートされた後、V<sub>BUS</sub> が 0.2 V まで放電するまで待つ必要があります。この放電時間の値は、トランシーバのベンダから入手でき、ベンダごとに異なります。
3. OTG\_FS/OTG\_HS コアは PHY に V<sub>BUS</sub> の放電を加速させることを通知します。
4. アプリケーションは、OTG 制御およびステータスレジスタのセッションリクエストビットに書き込むことによって、SRP を開始します。OTG\_FS/OTG\_HS コントローラは、データラインパルシングに続いて V<sub>BUS</sub> パルシングを実行します。
5. ホストは、データラインパルシングまたは V<sub>BUS</sub> パルシングから SRP を検出して、V<sub>BUS</sub> をオンにします。PHY は、デバイスに V<sub>BUS</sub> 電源オンを知らせます。

- OTG\_FS/OTG\_HS コントローラは、V<sub>BUS</sub> パルシングを実行します。  
ホストは、V<sub>BUS</sub>をオンにして、SRP が成功したことを示すことによって、新しいセッションを開始します。OTG\_FS/OTG\_HS コントローラは OTG 割り込みステータスレジスタのセッションリクエスト成功ステータス変化ビットをセットすることによって、アプリケーションに割り込みをかけます。アプリケーションは、OTG 制御およびステータスレジスタのセッションリクエスト成功ビットを読み出します。
- USB に電源が投入されると、OTG\_FS/OTG\_HS コントローラが接続され、SRP プロセスが完了します。

## A デバイスホストネゴシエーションプロトコル

HNP は、USB ホストの役割を A デバイスから B デバイスに切り替えます。アプリケーションは、OTG\_FS/OTG\_HS コントローラが HNP を A デバイスとして実行できるように、コア USB 設定レジスタの HNP 対応ビットをセットする必要があります。

図 467. A デバイスの HNP



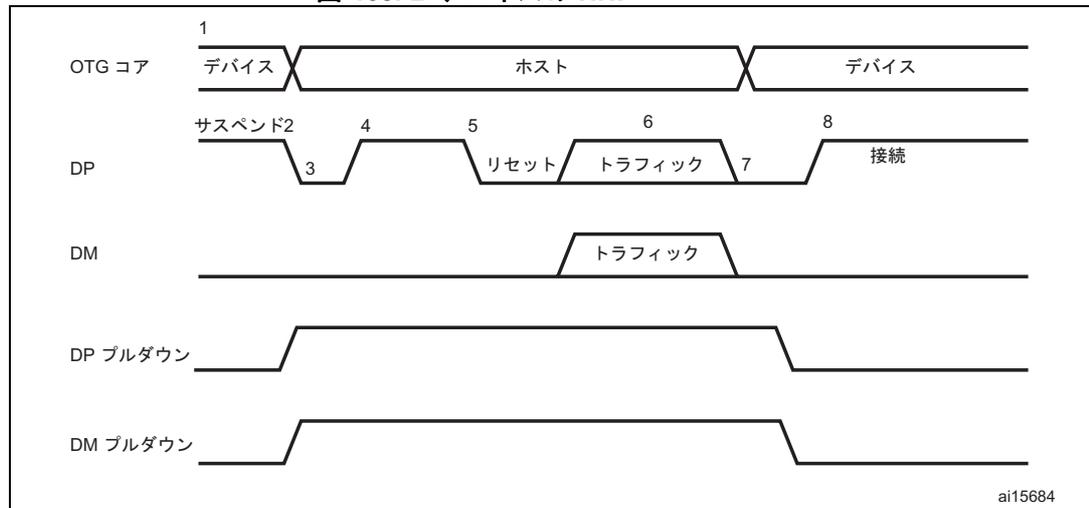
- DPPULLDOWN = PHY 内部の DP ライン上でプルダウンを有効/無効にするためのコアから PHY への信号  
DMPULLDOWN = PHY 内部の DM ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
- OTG\_FS/OTG\_HS コントローラは、SetFeature b\_hnp\_enable 記述子を B デバイスに送信して、HNP サポートを有効にします。B デバイスの ACK 応答は、B デバイスが HNP をサポートしていることを示します。アプリケーションは OTG 制御およびステータスレジスタのセット HNP イネーブルビットをセットして、B デバイスが HNP をサポートしていることを OTG\_FS/OTG\_HS コントローラに示す必要があります。
- アプリケーションがバスの使用を終了したときには、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによって、サスペンドします。
- B デバイスが USB サスペンドを検出すると、切断して、HNP の初期条件を示します。B デバイスは、ホストの役割に切り替える必要があるときにのみ、HNP を開始します。そうでないときには、バスはサスペンド状態を継続します。  
OTG\_FS/OTG\_HS コントローラは OTG 割り込みステータスレジスタのホストネゴシエーション検出割り込みをセットして HNP の開始を示します。  
OTG\_FS/OTG\_HS コントローラは、PHY の DP プルダウンと DM プルダウンをネゲートして、デバイスの役割を示します。PHY は、OTG\_DP プルアップ抵抗を有効にして、B デバイスの接続を示します。  
アプリケーションは、OTG 制御およびステータスレジスタの現在モードビットを読み出して、デバイスモードの動作を確認する必要があります。

4. B デバイスは接続を検出し、USB リセットを発行し、データトラフィックのために OTG\_FS/OTG\_HS コントローラをエニュメレーションします。
5. B デバイスはホストの役割を継続し、トラフィックを開始し、終了時にはバスをサスペンドします。  
OTG\_FS/OTG\_HS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割り込みレジスタのアーリーサスペンドビットをセットします。これに続いて OTG\_FS/OTG\_HS コントローラは、コア割り込みレジスタの USB サスペンドビットをセットします。
6. ネゴシエートモードでは、OTG\_FS/OTG\_HS コントローラは、サスペンドを検出して切断し、再度ホストの役割に切り替えます。OTG\_FS/OTG\_HS コントローラは、ホストの役割の前提を示すために、PHY の DP プルダウンと DM プルダウンをアサートします。
7. OTG\_FS/OTG\_HS コントローラは OTG 割り込みステータスレジスタのコネクタ ID ステータス変更割り込みをセットします。アプリケーションは、OTG 制御およびステータスレジスタのコネクタ ID ステータスを読み出して、OTG\_FS/OTG\_HS コントローラが A デバイスとして動作することを確認する必要があります。これは、HNP の完了をアプリケーションに示します。アプリケーションは、OTG 制御およびステータスレジスタの現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
8. B デバイスが接続され、HNP プロセスが完了します。

## B デバイスホストネゴシエーションプロトコル

HNP は、USB ホストの役割を B デバイスから A デバイスに切り替えます。アプリケーションは、OTG\_FS/OTG\_HS コントローラが HNP を B デバイスとして実行できるように、コア USB 設定レジスタの HNP 対応ビットをセットする必要があります。

図 468. B デバイスの HNP



1. DPPULLDOWN = PHY 内部の DP ライン上でプルダウンを有効/無効にするためのコアから PHY への信号  
DMPULLDOWN = PHY 内部の DM ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
1. A デバイスは、SetFeature b\_hnp\_enable 記述子を送信して、HNP サポートを有効にします。OTG\_FS/OTG\_HS コントローラの ACK 応答は、それが HNP をサポートしていることを示します。アプリケーションは、OTG 制御およびステータスレジスタのデバイス HNP有効化ビットをセットして、HNP をサポートしていることを示す必要があります。  
アプリケーションは、OTG 制御およびステータスレジスタの HNP リクエストビットをセットして、HNP を開始することを OTG\_FS/OTG\_HS コントローラに示す必要があります。

2. A デバイスがバスの使用を終了したときには、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによってサスペンドします。  
OTG\_FS/OTG\_HS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割り込みレジスタのアーリーサスペンドビットをセットします。これに続いて OTG\_FS/OTG\_HS コントローラは、コア割り込みレジスタの USB サスペンドビットをセットします。  
OTG\_FS/OTG\_HS コントローラは切断され、A デバイスはバス上で SE0 を検出して HNP を示します。OTG\_FS/OTG\_HS コントローラは、ホストの役割の前提を示すために、PHY の DP プルダウンと DM プルダウンをアサートします。  
A デバイスは、SE0 を検出してから 3 ms 以内に OTG\_DP プルアップ抵抗を有効にすることによって応答します。OTG\_FS/OTG\_HS コントローラはこれを接続として検出します。  
OTG\_FS/OTG\_HS コントローラは、OTG 割り込みステータスレジスタのホストネゴシエーション成功ステータス変更割り込みをセットして、HNP ステータスを示します。アプリケーションは、OTG 制御およびステータスレジスタのホストネゴシエーション成功ビットを読み出して、ホストネゴシエーションの成功を確認する必要があります。アプリケーションは、コア割り込みレジスタ (OTG\_GINTSTS) の現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
3. アプリケーションはリセットビット (OTG\_HPRT の PRST) をセットし、OTG\_FS/OTG\_HS コントローラは USB リセットを発行し、データトラフィックのために A デバイスをエニュメレーションします。
4. OTG\_FS/OTG\_HS コントローラは、トラフィックを開始するというホストの役割を継続し、これが終了すると、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによってバスをサスペンドします。
5. ネゴシエートモードでは、A デバイスは、サスペンドを検出すると、接続を切断し、ホストの役割に戻ります。OTG\_FS/OTG\_HS コントローラは PHY の DP プルダウンと DM プルダウンをネゲートして、デバイスの役割の前提を示します。
6. アプリケーションは、コア割り込みレジスタ (OTG\_GINTSTS) の現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
7. OTG\_FS/OTG\_HS コントローラが接続され HNP プロセスが完了します。

## 38 イーサネット (ETH) : メディアアクセス制御 (MAC)、DMA コントローラ付き

### 38.1 イーサネットの概要

Portions Copyright (c) 2004, 2005 Synopsys, Inc. All rights reserved. 掲載許可取得済み。

イーサネットペリフェラルでは、STM32F75xxx および STM32F74xxx は、IEEE 802.3-2002 標準に準拠したイーサネット経由でデータを送受信できます。

イーサネットは、さまざまなアプリケーションやユーザのニーズを満たす、設定可能で柔軟なペリフェラルを提供します。イーサネットは、外部物理層 (PHY) に対して 2 つの業界標準インタフェース、すなわち IEEE 802.3 で定義されているデフォルトのメディア独立インタフェース (MII) と、減少メディア独立インタフェース (RMII) をサポートしています。スイッチ、ネットワークインタフェースカードなど、多くのアプリケーションで使用できます。

イーサネットは、以下の標準に準拠しています。

- IEEE 802.3-2002 (イーサネット MAC)
- 高精度ネットワーク化クロック同期の IEEE 1588-2008 標準
- AMBA 2.0 (AHB マスタ/スレーブポート)
- RMII 仕様 (RMII コンソーシアム)

### 38.2 イーサネットの主な機能

イーサネット (ETH) ペリフェラルは、次のような機能を備えています (カテゴリ別)。

#### 38.2.1 MAC コア機能

- 外部 PHY インタフェースで 10/100 Mbit/s のデータ転送速度をサポートします。
- IEEE 802.3 準拠の MII インタフェースで、外部ファストイーサネット PHY と通信します。
- 全二重と半二重の両方の動作をサポートします。
  - 半二重動作では CSMA/CD プロトコルをサポートします。
  - 全二重動作では、IEEE 802.3x フロー制御をサポートします。
  - 全二重動作では、受信したポーズ制御フレームをユーザアプリケーションに転送するオプションを備えています。
  - 半二重動作でのバックプレッシャサポート。
  - 全二重動作では、フロー制御入力のネゲート時に、ゼロクオンタムポーズフレームを自動送信します。
- プリアンブルとフレーム開始データ (SFD) を送信パスに挿入し、受信パスで削除します。
- フレームごとに制御可能な CRC とパッドの自動生成。
- 受信フレームでの自動パッド/CRC ストリッピングのためのオプション。
- 最大 16 KB までのサイズの標準フレームをサポートするプログラム可能なフレーム長。
- プログラム可能なフレーム間ギャップ (40~96 ビット時間、8 ビット時間単位で)。
- 以下のような柔軟なアドレスフィルタリングをサポートします。
  - 各バイトのマスク付きで、最大 4 つの 48 ビット完全アドレス (DA) フィルタ。
  - 各バイトのマスク付きで、最大 3 つの 48 ビット SA アドレス比較チェック。

- マルチキャストおよびユニキャスト (DA) アドレス用 64 ビットハッシュフィルタ (オプション)。
- すべてのマルチキャストアドレスフレームを通過させるオプション。
- ネットワーク監視にフィルタリングを使用せず、すべてのフレームを通過させるプロミスキュアスモードをサポートします。
- ステータスレポート付きで、すべての受信パケットを (フィルタに従って) 通過させます。
- 送信および受信パケットについて、個別の 32 ビットステータス情報を返します。
- 受信フレームの IEEE 802.1Q VLAN タグ検出をサポートします。
- アプリケーションへの、個別の送信、受信、および制御インタフェース。
- RMON/MIB カウンタ (RFC2819/RFC2665) による強制ネットワーク統計をサポートします。
- PHY デバイスの設定と管理のための MDIO インタフェース。
- LAN ウェイクアップフレームおよび AMD Magic Packet™ フレームの検出。
- イーサネットフレームによってカプセル化された受信 IPv4 および TCP パケットのチェックサムオフロード受信機能。
- IPv4 ヘッダチェックサムと、IPv4 または IPv6 データグラムにカプセル化された TCP、UDP、または ICMP チェックサムをチェックする拡張受信機能。
- IEEE 1588-2008 準拠のイーサネットフレームタイムスタンプをサポートします。各フレームの送信または受信ステータスに 64 ビットのタイムスタンプが付加されます。
- 2 セットの FIFO : 閾値を設定できる 2 KB の送信 FIFO と、閾値を設定できる (デフォルトは 64 バイト) 2 KB の受信 FIFO。
- EOF 転送後に受信ステータスペクタが受信 FIFO に挿入されるので、別の FIFO を必要とせずに、受信 FIFO 内の複数フレームストレージにそれらのフレームの受信ステータスを格納できます。
- ストアアンドフォワードモードで、受信時にすべてのエラーフレームをフィルタして、アプリケーションに転送しないオプション。
- アンダーサイズ的良好フレームを転送するオプション。
- 受信 FIFO で (オーバーフローにより) ドロップまたは破損したフレームについて、パルスを生成することにより統計をサポートします。
- MAC コアへの送信について、ストアアンドフォワードメカニズムをサポートします。
- 受信 FIFO フィルレベル (閾値を設定可能) に基づいて、MAC コアへのポーズフレーム制御またはバックプレッシャ信号を自動生成します。
- 送信について、コリジョンフレームの自動再送信を処理します。
- レイトコリジョン、過剰コリジョン、過剰遅延、およびアンダーラン条件時にフレームを破棄します。
- Tx FIFO をフラッシュするためのソフトウェア制御。
- ストアアンドフォワードモードで送信されたフレームに、IPv4 ヘッダチェックサムと TCP、UDP、または ICMP チェックサムを計算して挿入します。
- デバッグのための MII での内部ループバックをサポートします。

## 38.2.2 DMA の機能

- AHB スレーブインタフェースですべての AHB バーストタイプをサポートします。
- ソフトウェアは、AHB マスタインタフェースの AHB バーストのタイプ (固定または無限バースト) を選択できます。
- AHB マスタポートからアドレス整列バーストを選択するオプション。
- フレームデリミタ付きパケット指向 DMA 転送の最適化。
- データバッファサポートのためのバイト整列アドレッシング。
- デュアルバッファ (リング) またはリンクリスト (連鎖) ディスクリプタ連鎖。
- 最小の CPU 介入で大きなデータブロックの転送を可能にするディスクリプタアーキテクチャ。
- 各ディスクリプタは最大 8 KB のデータを転送できます。
- 正常動作とエラーの転送についての包括的なステータスレポート。
- 最適なホストバス利用のために、送信および受信 DMA エンジンのバーストサイズを個別にプログラム可能。
- さまざまな動作条件に対応するプログラム可能な割り込みオプション。
- フレームごとの送信/受信完了割り込み制御。
- 受信エンジンと送信エンジン間のラウンドロビン方式または固定優先順位のアービトレーション。
- 開始/停止モード。
- ステータスレジスタとしての現在の Tx/Rx バッファポインタ。
- ステータスレジスタとしての現在の Tx/Rx ディスクリプタポインタ。

## 38.2.3 PTP の機能

- 受信および送信フレームのタイムスタンプング。
- 粗密補正法。
- システム時間が目標時間を超えたときに割り込みをトリガします。
- 1 秒あたりのパルス出力 (製品代替機能出力)。

## 38.3 イーサネットピン

表 235 に、MAC 信号と MII/RMII 信号の対応を示します。すべての MAC 信号は AF11 にマップされ、いくつかの信号は異なる I/O ピンにマップされて、別の機能モードに設定されます (詳細については、[セクション 6.3.2 : I/O ピンオルタネート機能マルチプレクサと配置](#)を参照してください)。

表 235. オルタネート機能配置

ポート	AF11
	ETH
PA0-WKUP	ETH_MII_CRS
PA1	ETH_MII_RX_CLK / ETH_RMII_REF_CLK
PA2	ETH_MDIO
PA3	ETH_MII_COL
PA7	ETH_MII_RX_DV / ETH_RMII_CRS_DV
PB0	ETH_MII_RXD2
PB1	ETH_MII_RXD3
PB5	ETH_PPS_OUT
PB8	ETH_MII_TXD3
PB10	ETH_MII_RX_ER
PB11	ETH_MII_TX_EN / ETH_RMII_TX_EN
PB12	ETH_MII_TXD0 / ETH_RMII_TXD0
PB13	ETH_MII_TXD1 / ETH_RMII_TXD1
PC1	ETH_MDC
PC2	ETH_MII_TXD2
PC3	ETH_MII_TX_CLK
PC4	ETH_MII_RXD0 / ETH_RMII_RXD0
PC5	ETH_MII_RXD1 / ETH_RMII_RXD1
PE2	ETH_MII_TXD3
PG8	ETH_PPS_OUT
PG11	ETH_MII_TX_EN / ETH_RMII_TX_EN
PG13	ETH_MII_TXD0 / ETH_RMII_TXD0
PG14	ETH_MII_TXD1 / ETH_RMII_TXD1
PH2	ETH_MII_CRS
PH3	ETH_MII_COL
PH6	ETH_MII_RXD2
PH7	ETH_MII_RXD3
PI10	ETH_MII_RX_ER

## 38.4 イーサネット機能の説明 : SMI、MII、RMII

イーサネットペリフェラルは、専用 DMA コントローラ付きの MAC 802.3 (メディアアクセス制御) で構成されます。デフォルトのメディア独立インタフェース (MII) と減少メディア独立インタフェース (RMII) をサポートし、選択ビットを使っていずれかを選択することができます (SYSCFG\_PMC レジスタを参照)。

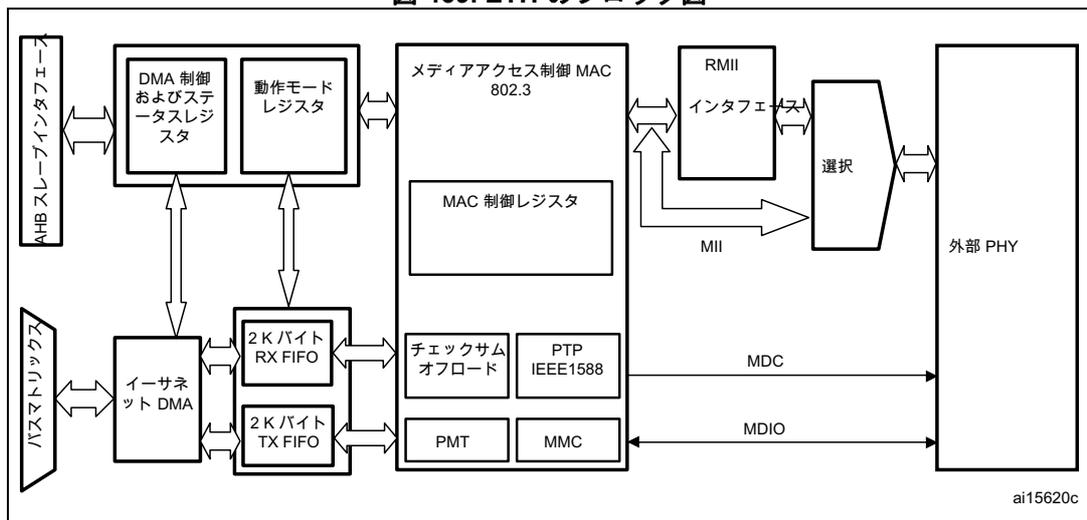
DMA コントローラは、AHB マスタおよびスレーブインタフェースを通じて、コアおよびメモリとインタフェースします。AHB マスタインタフェースはデータ転送を制御し、AHB スレーブインタフェースは制御およびステータスレジスタ (CSR) スペースにアクセスします。

送信 FIFO (Tx FIFO) は、MAC コアによる送信の前に、DMA によってシステムメモリから読み出されたデータをバッファします。同様に、受信 FIFO (Rx FIFO) は、ラインから受信したイーサネットフレームを、DMA によってシステムメモリに転送されるまで格納します。

イーサネットペリフェラルは、外部 PHY と通信するための SMI も含んでいます。一連の設定レジスタによって、ユーザは、MAC および DMA コントローラの動作モードと機能を選択することができます。

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

図 469. ETH のブロック図



1. AHB 接続については、図 1 : STM32F75xxx および STM32F74xxx デバイスのシステムアーキテクチャを参照してください。

### 38.4.1 ステーション管理インタフェース : SMI

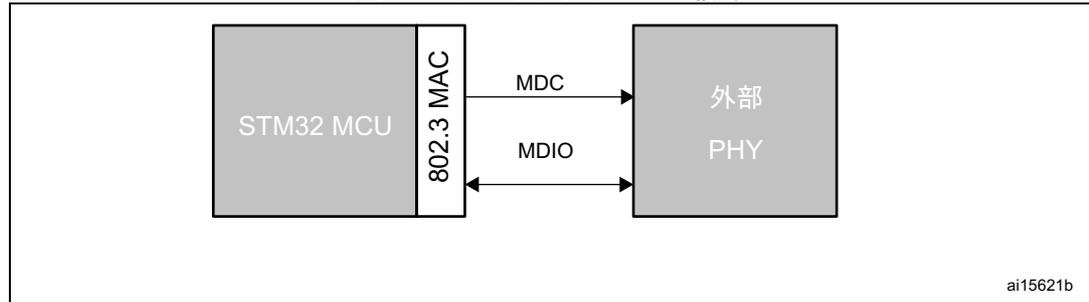
ステーション管理インタフェース (Station Management Interface : SMI) によって、アプリケーションは、2 線のクロックおよびデータラインを通じて任意の PHY にアクセスできます。インタフェースは、最大 32 の PHY へのアクセスをサポートします。

アプリケーションは、32 の PHY のうちの 1 つと、PHY 内の 32 のレジスタのうちの 1 つを選択して、制御データを送信したり、ステータス情報を受信することができます。一度にアドレス指定できるのは、1 つの PHY 内の 1 つのレジスタだけです。

MDC クロックラインと MDIO データラインの両方が、マイクロコントローラ内の代替機能として実装されています。

- MDC : 最大周波数 2.5 MHz でデータを転送するために必要な基準タイミングを供給する周期的クロック。MDC のハイ/ロー時間の最小値は、それぞれ 160 ns でなければならず、MDC の最小周期は 400 ns でなければなりません。アイドル状態では、SMI 管理インタフェースは MDC クロック信号をローに駆動します。
- MDIO : PHY デバイスとの間のステータス情報の転送を MDC クロック信号と同期して行うデータ入力/出力ビットストリーム。

図 470. SMI インタフェース信号



## SMI フレームのフォーマット

読み出しおよび書き込み動作に関連するフレーム構造を表 236 に示します。ビット送信の順序は、左から右でなければなりません。

表 236. 管理フレームのフォーマット

	管理フレームのフィールド							
	プリアンブル (32 ビット)	開始	動作	PADDR	RADDR	TA	データ (16 ビット)	アイドル
読み出し	1...1	01	10	ppppp	rrrrr	Z0	ddddddddddddddd	Z
書き込み	1...1	01	01	ppppp	rrrrr	10	ddddddddddddddd	Z

管理フレームは、8 つのフィールドで構成されます。

- **プリアンブル** : 各トランザクション (読み出しまたは書き込み) は、MDC 上の 32 の対応するサイクルで、MDIO ライン上の 32 の連続する論理 1 ビットに対応するプリアンブルフィールドで開始できます。このフィールドは、PHY デバイスとの同期を確立するために使用されます。
- **開始** : フレームの開始は、ライン上のデフォルトの論理 1 状態から 0 へ、そして 1 へ戻るといふ遷移を確認する <01> パターンによって定義されます。
- **動作** : 進行中のトランザクションのタイプ (読み出しまたは書き込み) を定義します。
- **PADDR** : 5 ビットの PHY アドレスであり、32 の一意な PHY アドレスを指定できます。アドレスの MSB ビットが最初に送受信されます。
- **RADDR** : 5 ビットのレジスタアドレスであり、選択された PHY デバイス内の 32 の個別のレジスタをアドレス指定できます。アドレスの MSB ビットが最初に送受信されます。

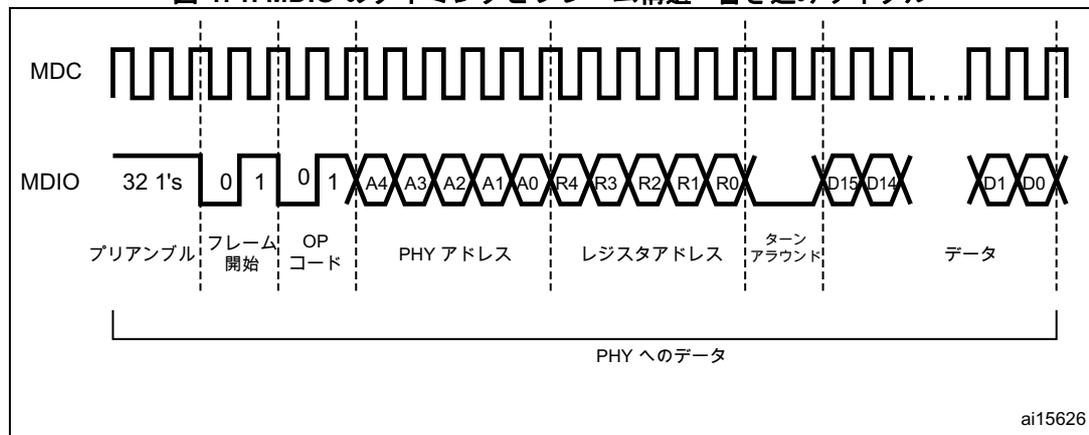
- **TA** : ターンアラウンドフィールドは、読み出しトランザクション時の競合を避けるために、RADDR フィールドと DATA フィールドの間の 2 ビットのパターンを定義します。読み出しトランザクションの場合、MAC コントローラは、TA の 2 ビットに対して MDIO ライン上でハイインピーダンスを駆動します。PHY デバイスは、TA の最初のビットでハイインピーダンス状態を、2 番目のビットでゼロを駆動しなければなりません。  
書き込みトランザクションの場合、MAC コントローラは、TA フィールド時に <10> パターンを駆動します。PHY デバイスは、TA の 2 ビットに対してハイインピーダンス状態を駆動しなければなりません。
- **データ** : データフィールドは 16 ビットです。送受信される最初のビットは、ETH\_MIID レジスタのビット 15 でなければなりません。
- **アイドル** : MDIO ラインはハイインピーダンス状態に駆動されます。すべてのトライステートドライバが無効でなければならず、PHY のプルアップ抵抗はラインを論理 1 に保ちます。

## SMI 書き込み動作

アプリケーションが MII 書き込みビットおよびビジービット(イーサネット MAC MII アドレスレジスタ (ETH\_MACMIIAR)) をセットすると、SMI は、PHY アドレス、PHY 内のレジスタアドレス、および書き込みデータ (イーサネット MAC MII データレジスタ (ETH\_MACMIIDR)) を送信することによって、PHY レジスタへの書き込み動作を開始します。アプリケーションは、トランザクションの進行中は、MII アドレスレジスタの内容や MII データレジスタを変更しないでください。この期間中の MII アドレスレジスタまたは MII データレジスタへの書き込み動作は無視され (ビジービットがハイ)、トランザクションはエラーなく完了します。書き込み動作が完了した後、SMI は、ビジービットをリセットすることによって、これを示します。

図 471 に、書き込み動作の場合のフレームフォーマットを示します。

図 471. MDIO のタイミングとフレーム構造 - 書き込みサイクル

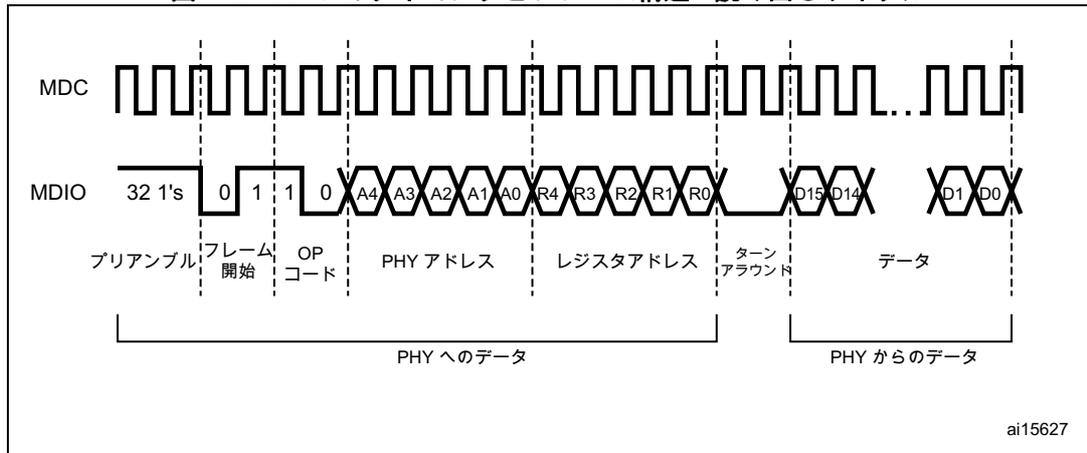


## SMI 読み出し動作

ユーザが MII 書き込みビットが 0 にセットされたイーサネット MAC MII アドレスレジスタ (ETH\_MACMIIAR) の MII ビジービットをセットすると、SMI は、PHY アドレスと PHY 内のレジスタアドレスを転送することによって、PHY レジスタの読み出し動作を開始します。アプリケーションは、トランザクションの進行中は、MII アドレスレジスタの内容や MII データレジスタを変更しないでください。この期間中の MII アドレスレジスタまたは MII データレジスタへの書き込み動作は無視され (ビジービットがハイ)、トランザクションはエラーなく完了します。読み出し動作が完了した後、SMI は、ビジービットをリセットし、PHY から読み出したデータで MII データレジスタを更新します。

図 472 に、読み出し動作の場合のフレームフォーマットを示します。

図 472. MDIO のタイミングとフレーム構造 - 読み出しサイクル



## SMI クロックの選択

MAC は、書き込み／読み出し動作の管理を開始します。SMI クロックは、アプリケーションクロック (AHB クロック) を分周したクロックです。分周比は、MII アドレスレジスタのクロック範囲設定に依存します。

表 237 に、クロック範囲の設定方法を示します。

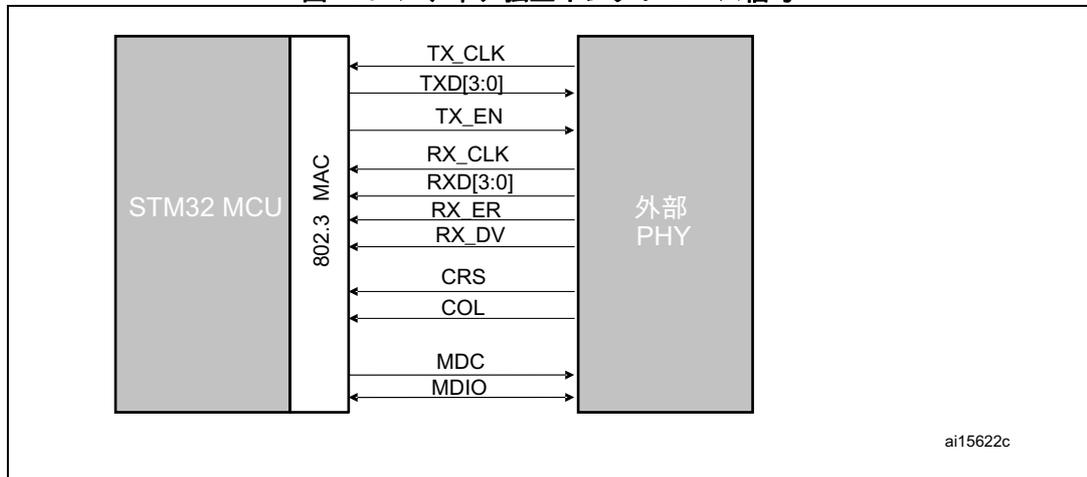
表 237. クロック範囲

選択	CLK クロック	MDC クロック
000	60~100 MHz	AHB クロック/42
001	100~150 MHz	AHB クロック/62
010	20~35 MHz	AHB クロック/16
011	35~60 MHz	AHB クロック/26
100	150~216 MHz	AHB クロック/102
101、110、111	予約済み	-

## 38.4.2 メディア独立インタフェース : MII

メディア独立インタフェース (MII) は、10 Mbit/s および 100 Mbit/s でのデータ転送における MAC サブレイヤと PHY 間の相互接続を定義します。

図 473. メディア独立インタフェース信号



- MII\_TX\_CLK: TX データ転送の基準タイミングを供給する連続クロックです。公称周波数: 10Mbit/s 速度で 2.5 MHz; 100 Mbit/s 速度で 25 MHz。
- MII\_RX\_CLK: RX データ転送の基準タイミングを供給する連続クロックです。公称周波数: 10Mbit/s 速度で 2.5 MHz; 100 Mbit/s 速度で 25 MHz。
- MII\_TX\_EN: 送信イネーブルは、MAC が送信のために MII 上にニブルを提示していることを示します。プリアンプルの最初のニブル (MII\_TX\_CLK) に同期してアサートされなければならず、送信されるすべてのニブルが MII に提示されるまで、アサート状態が維持されなければなりません。
- MII\_TXD[3:0]: 送信データは、MAC サブレイヤによって同期して駆動される 4 つのデータ信号の束であり、MII\_TX\_EN 信号のアサーションによって有効なデータと認められます。MII\_TXD[0] は最下位ビットであり、MII\_TXD[3] が最上位ビットです。MII\_TX\_EN がネゲートされている場合、送信データは PHY に何の影響も与えません。
- MII\_CRD: キャリアセンスは、送信または受信メディアがアイドルでないとき、PHY によってアサートされます。送信メディアと受信メディアの両方ともアイドルのときには、PHY によってネゲートされなければなりません。PHY は、コリジョン条件の全期間を通じて、MII\_CS 信号がアサートされたままであることを保証する必要があります。この信号は、TX および RX クロックに同期して遷移する必要はありません。全二重モードでは、この信号の状態は MAC サブレイヤに影響を与えません。
- MII\_COL: メディア上でコリジョンが検出された場合、PHY によってコリジョン検出がアサートされなければならず、コリジョン条件が続いている間、アサート状態が保たれなければなりません。この信号は、TX および RX クロックに同期して遷移する必要はありません。全二重モードでは、この信号の状態は MAC サブレイヤに影響を与えません。
- MII\_RXD[3:0]: 受信データは、PHY によって同期して駆動される 4 つのデータ信号の束であり、MII\_RX\_DV 信号のアサーションによって有効なデータと認められます。MII\_RXD[0] は最下位ビットであり、MII\_RXD[3] が最上位ビットです。MII\_RX\_EN がネゲートされ、MII\_RX\_ER がアサートされている間、PHY から特定の情報を転送するために特定の MII\_RXD[3:0] 値が使用されます (表 239 を参照)。

- MII\_RX\_DV : 受信データ有効ビットは、PHY が MII 上に受信のために回収され、デコードされたニブルを提示していることを示します。フレームの最初の回収ニブルと同期してアサートされなければならない (MII\_RX\_CLK)、最後の回収ニブルまでアサート状態を維持しなければなりません。最後のニブル後の最初のクロックの前にネゲートされなければなりません。フレームを正しく受信するためには、MII\_RX\_DV 信号は、SFD フィールドよりも遅れることなく開始して、フレームを取り込む必要があります。
- MII\_RX\_ER : 受信エラービットは、フレーム内のどこかでエラーが検出されたことを MAC サブレイヤに示すために、1 つまたは複数のクロック周期 (MII\_RX\_CLK) の間、アサートされなければなりません。このエラー条件は、表 239 での説明のように、MII\_RX\_DV ビットのアサートにより有効とされなければなりません。

表 238. TX インタフェース信号のエンコーディング

MII_TX_EN	MII_TXD[3:0]	説明
0	0000 から 1111 まで	通常のフレーム間
1	0000 から 1111 まで	通常データ送信

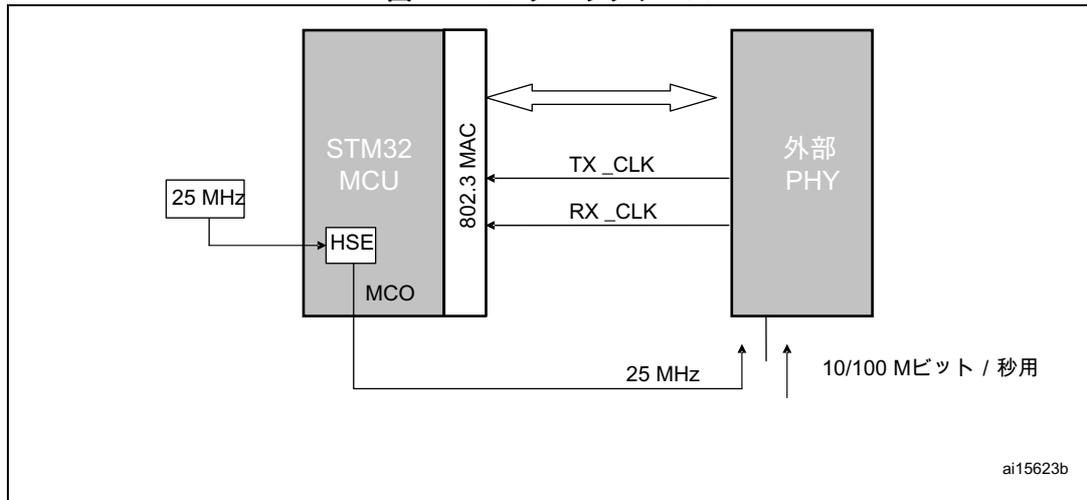
表 239. RX インタフェース信号のエンコーディング

MII_RX_DV	MII_RX_ERR	MII_RXD[3:0]	説明
0	0	0000 から 1111 まで	通常のフレーム間
0	1	0000	通常のフレーム間
0	1	0001 から 1101 まで	予約済み
0	1	1110	不正キャリアの表示
0	1	1111	予約済み
1	0	0000 から 1111 まで	通常データ受信
1	1	0000 から 1111 まで	データ受信エラー

## MII クロックソース

TX\_CLK と RX\_CLK の両方のクロック信号を生成するには、図 474 に示されているように、外部 PHY を外部の 25 MHz クロックソースで駆動する必要があります。このクロックを供給するために外部の 25 MHz クォーツを使用する代わりに、STM32F75xxx and STM32F74xxx マイクロコントローラは、MCO ピンに出力することができます。この場合、PLL マルチプライヤを 25 MHz の外部クォーツから MCO ピン上に必要な周波数を出力するように設定する必要があります。

図 474. MII クロックソース



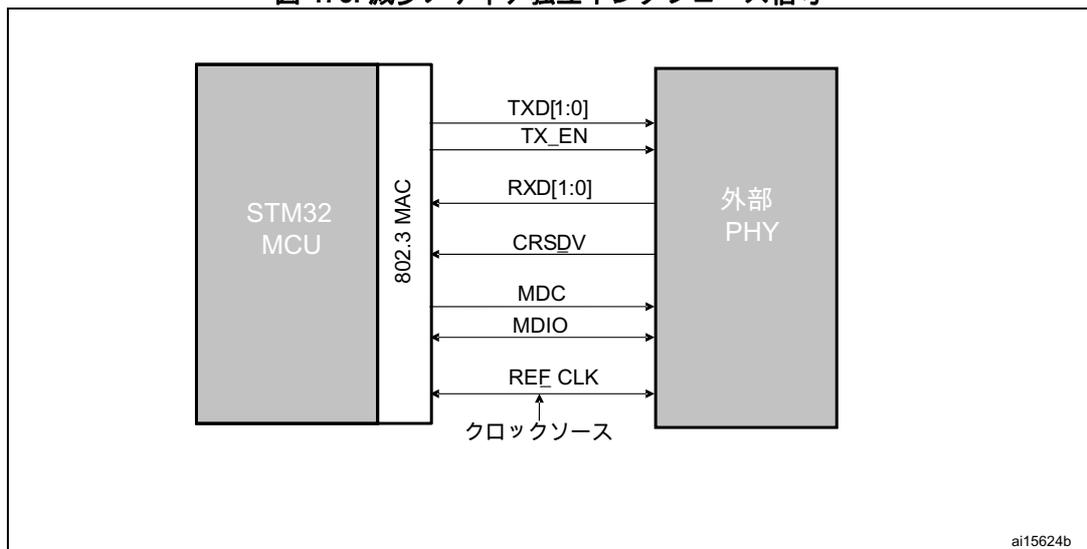
### 38.4.3 減少メディア独立インタフェース : RMII

減少メディア独立インタフェース (RMII) 仕様は、マイクロコントローラーイーサネットペリフェラルと 10/100 Mbit/s の外部イーサネット間のピン数を削減します。IEEE 802.3u 標準に従って、MII は、データおよび制御のために 16 本のピンを備えています。RMII 仕様は、ピン数を 7 本に減らす (ピン数を 62.5% 減らす) ことを目的としています。

RMII は、MAC と PHY の間でインスタンス化されます。これは、MAC の MII を RMII に変換するのに役立ちます。RMII ブロックは、次のような特性を備えています。

- 10 Mbit/s と 100 Mbit/s の動作速度をサポートします。
- 基準クロックは 50 MHz (2 倍) に高められなければなりません。
- 同じ基準クロックが MAC と外部イーサネット PHY の両方に供給されなければなりません。
- 独立した 2 ビット幅 (双ビット) の送信および受信データパスを提供します。

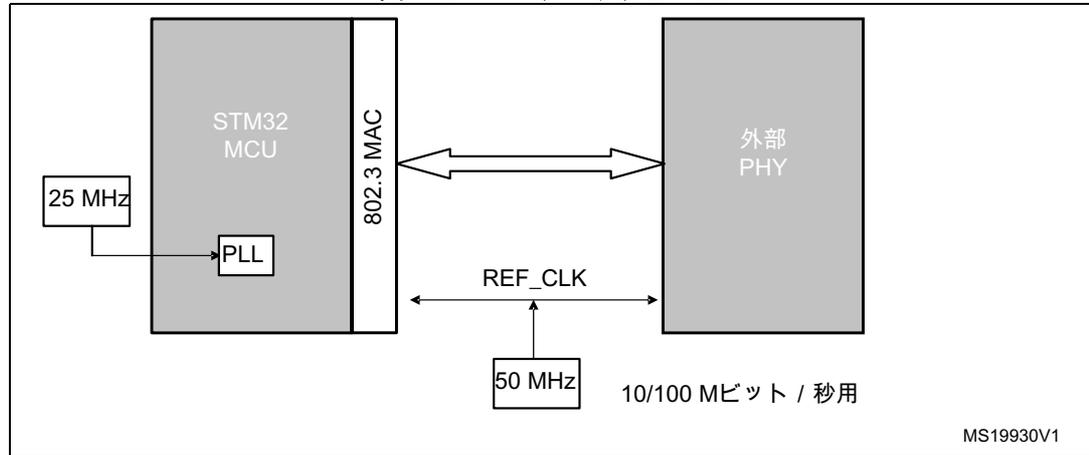
図 475. 減少メディア独立インタフェース信号



## RMII クロックソース

外部50 MHz クロックで PHY を駆動するか、または PHY と内蔵PLL の組み合わせを使って 50 MHz 周波数を発生します。

図 476. RMII クロックソース



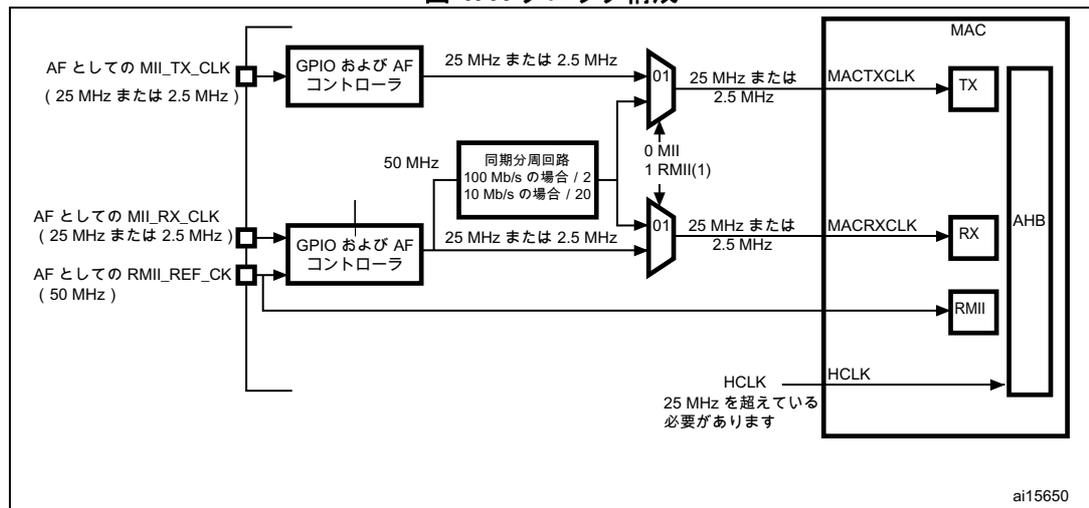
## 38.4.4 MII/RMII の選択

MII モードまたは RMII モードは、SYSCFG\_PMC レジスタの設定ビット 23 (MII\_RMII\_SEL) を使用して選択されます。アプリケーションは、イーサネットコントローラがリセット状態にあるとき、またはクロックを有効にする前に、MII/RMII モードを設定する必要があります。

### MII/RMII 内部クロック構成

MII と RMII の両方、ならびに 10 および 100 Mbit/s 動作をサポートするために必要なクロック構成を [図 477](#) に示します。

図 477. クロック構成



1. MII/RMII 選択は、SYSCFG\_PMCレジスタのビット 23 (MII\_RMII\_SEL) を通じて制御されます。

ピン数を節約するために、2つの入力クロック信号、RMII\_REF\_CLK と MII\_RX\_CLK が同じ GPIO ピンで多重化されます。

## 38.5 イーサネット機能の説明 : MAC 802.3

ローカルエリアネットワーク (LAN) に関する IEEE 802.3 国際標準では、アクセス方式として CSMA/CD (Carrier Sense Multiple Access with Collision Detection) を採用しています。

イーサネットペリフェラルは、メディア独立インタフェース (MII) と専用 DMA コントローラ付きの MAC 802.3 (メディアアクセス制御) で構成されます。

MAC ブロックは、ベースバンドシステムとブロードバンドシステムに対して 10 Mbit/s と 100 Mbit/s のデータレートを持つシステムファミリに対する LAN CSMA/CD サブレイヤを実装しています。半二重および全二重動作がサポートされます。コリジョン検出アクセス方式は、半二重動作モードにのみ適用されます。MAC 制御フレームサブレイヤがサポートされます。

MAC サブレイヤは、データリンク制御プロシージャに関連する以下の機能を実行します。

- データのカプセル化 (送受信)
  - フレーミング (フレーム境界区切り、フレーム同期)
  - アドレッシング (ソースおよびデスティネーションアドレスの処理)
  - エラー検出
- メディアアクセス管理
  - メディア割り当て (コリジョン回避)
  - 衝突解決 (コリジョン処理)

基本的に、MAC サブレイヤには 2 つの動作モードがあります。

- 半二重モード : ステーションは CSMA/CD アルゴリズムを使用して、物理的メディアの使用を競います。
- 全二重モード : 以下の条件がすべて満たされているときには、競合解決を行うことなく同時送受信が可能です (CSMA/CD アルゴリズムは不要です)。
  - 物理メディアが同時送受信機能をサポート可能です。
  - LAN に 2 つのステーションだけが接続されています。
  - どちらのステーションも全二重動作に設定されています。

### 38.5.1 MAC 802.3 フレームフォーマット

MAC ブロックは、IEEE 802.3-2002 標準で規定されている MAC サブレイヤとオプションの MAC 制御サブレイヤ (10/100 Mbit/s) を実装します。

CSMA/CD MAC を使用したデータ通信システムについては、2 つのフレームフォーマットが規定されています。

- 基本 MAC フレームフォーマット
- タグ付き MAC フレームフォーマット (基本 MAC フレームフォーマットの拡張)

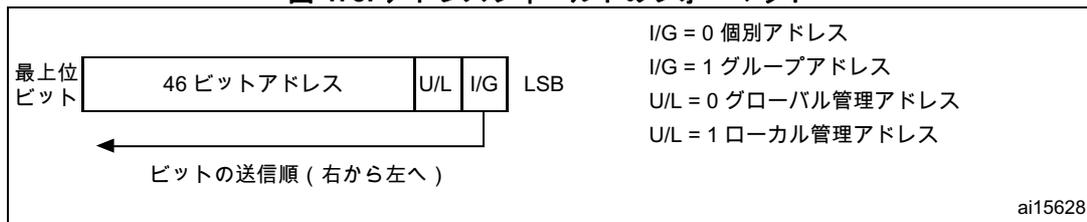
図 479 と 図 480 は、以下のフィールドを含むフレーム構造 (タグなしとタグ付き) を示します。

- プリアンブル : 同期目的で使用される 7 バイトのフィールド (PLS 回路)
  - 16進値 : 55-55-55-55-55-55-55
  - ビットパターン : 01010101 01010101 01010101 01010101 01010101 01010101 01010101 (右から左へビット転送)
- 開始フレームデリミタ (SFD) : フレームの開始を示すために使用される 1 バイトのフィールド。
  - 16進値 : D5
  - ビットパターン : 11010101 (右から左へビット転送)
- デスティネーションおよびソースアドレスフィールド : 以下のようなデスティネーションおよびソースステーションアドレスを示す 6 バイトのフィールド (図 478 を参照)。
  - 各アドレスは 48 ビット長です。
  - デスティネーションアドレスフィールドの最初の LSB ビット (I/G) は、個別のアドレス (I/G = 0) またはグループアドレス (I/G = 1) を示すために使用されます。グループアドレスは、LAN に接続されている 0 個、1 個以上、またはすべてのステーションを識別できます。ソースアドレスの最初のビットは予約済みであり、0 にリセットされます。
  - 2 番目のビット (U/L) は、ローカル管理アドレス (U/L = 1) とグローバル管理アドレス (U/L = 0) を区別します。ブロードキャストアドレスの場合は、このビットも 1 にセットされます。
  - 各アドレスフィールドの各バイトは、最下位ビットから送信されなければなりません。

アドレスの割り当ては、以下のタイプに基づきます。

- 個別アドレス : これは、ネットワーク上の特定のステーションに関連する物理アドレスです。
- グループアドレス : 特定のネットワーク上の 1 つまたは複数のステーションに関連付けられたマルチデスティネーションアドレス。マルチキャストアドレスには 2 種類あります。
  - マルチキャストグループアドレス : 論理的に関連するステーションのグループに関連付けられたアドレス。
  - ブロードキャストアドレス : 常に特定の LAN 上のすべてのステーションを表し、他と区別され、事前定義されたマルチキャストアドレス (デスティネーションアドレスフィールドがすべて 1)。

図 478. アドレスフィールドのフォーマット



- QTag プレフィックス : ソースアドレスフィールドと MAC クライアント長/タイプフィールドの間に挿入される 4 バイトのフィールド。このフィールドは、タグ付き MAC フレームを得るための基本フレーム (タグなし) の拡張です。タグなし MAC フレームには、このフィールドはありません。タグ付けのための拡張は、次のとおりです。
  - 2 バイト定数である長さ/タイプフィールドの値は、802.1Q タグプロトコルタイプ (16 進数の 0x8100) の値に等しいタイプ解釈 (0x0600 より大きい) に一致します。この定数フィールドは、MAC フレームがタグ付きか、タグなしかを区別するために使用されます。
  - タグ制御情報を含む 2 バイトのフィールドは、3 ビットのユーザ優先順位、正規フォーマットインジケータ (CFI)、および 12 ビットの VLAN 識別子に分けられます。タグ付き MAC フレームの長さは、QTag プレフィックスによって 4 バイトだけ拡張されます。

- MAC クライアント長/タイプ : 値によって異なる意味 (相互排他的) を持つ 2 バイトのフィールド。
  - 値が  $\text{maxValidFrame}$  (0d1500) 以下の場合、このフィールドは、後続の 802.3 フレームのデータフィールドに含まれる MAC クライアントデータバイトの数を示します (長さ解釈)。
  - 値が  $\text{MinTypeValue}$  (10 進数の 0d1536、0x0600) 以上の場合、このフィールドは、イーサネットフレームに関する MAC クライアントプロトコルの性質を示します (タイプ解釈)。

長さ/タイプフィールドの解釈に関係なく、データフィールドの長さがプロトコルの正常動作に必要な最小値より小さい場合は、データフィールドと FCS (フレームチェックシーケンス) フィールドの間に、PAD フィールドが追加されます。長さ/タイプフィールドは、上位バイトから送受信されます。

最大ValidLength から 最小TypeValue までの範囲 (境界は除く) の長さ/タイプフィールドの値の場合、MAC サブレイヤの動作は規定されていません。すなわち、MAC サブレイヤを通過することもあれば、通過しないこともあります。

- データおよび PAD フィールド : n バイトのデータフィールド。完全なデータ透過性が提供されます。これは、データフィールドには、バイト値が任意の順序で現われてもよいことを意味しています。PAD (ある場合) のサイズは、データフィールドのサイズによって決まります。データおよび PAD フィールドの最大長と最小長は、次のとおりです。
  - 最大長 = 1500 バイト
  - タグなし MAC フレームの最小長 = 46 バイト
  - タグ付き MAC フレームの最小長 = 42 バイト

データフィールド長が必要な最小値より短い場合、その最小値 (タグ付きフレームの場合は 42 バイト、タグなしフレームの場合は 46 バイト) になるように、PAD フィールドが追加されます。

- フレームチェックシーケンス : 巡回冗長検査 (CRC) の値を含む 4 バイトのフィールド。CRC の計算は、ソースアドレス、デスティネーションアドレス、QTag プレフィックス、長さ/タイプ、LLC データおよび PAD フィールドに基づきます (すなわち、プリアンプル、SFD 以外のすべてのフィールド)。計算多項式は、以下のように与えられます。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

フレームの CRC 値は、次のように計算されます。

- フレームの最初の 2 ビットの補数が取られます。
- フレームの n ビットは、(n - 1) 次の多項式 M(x) の係数です。デスティネーションアドレスの最初のビットは  $x^{n-1}$  項に対応し、データフィールドの最後のビットは  $x^0$  項に対応します。
- M(x) に  $x^{32}$  が乗算され、G(x) で除算されて、 $\leq 31$  次の剰余 R(x) が求められます。
- R(x) の係数は、32 ビットのシーケンスとみなされます。
- このビットシーケンスの補数が CRC になります。
- 32 ビットの CRC 値は、フレームチェックシーケンスに置かれます。 $x^{32}$  項が最初に送信され、 $x^0$  項が最後に送信されます。

図 479. MAC フレームフォーマット

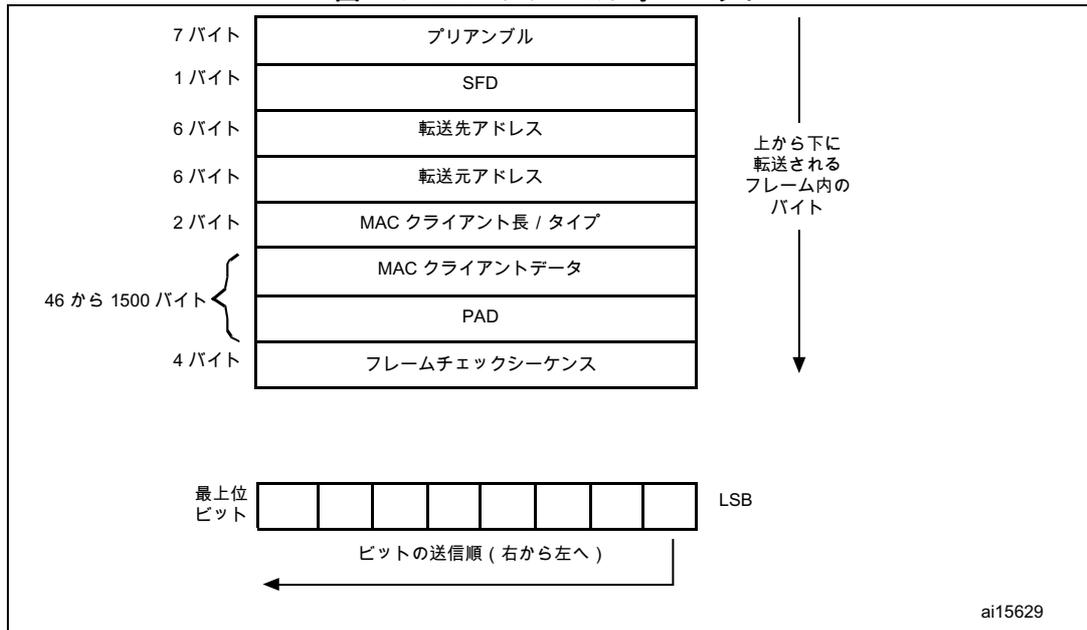
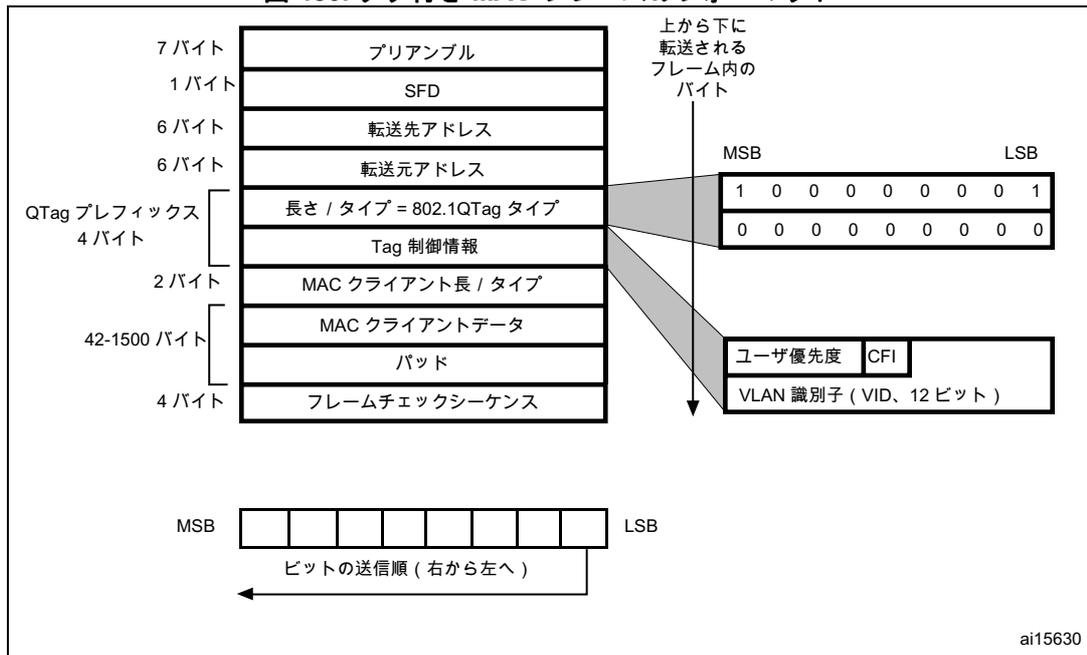


図 480. タグ付き MAC フレームのフォーマット



MAC フレームの各バイトは、FCS フィールドを除き、下位ビットから送信されます。

無効な MAC フレームは、次の条件の 1 つによって定義されます。

- フレーム長が、長さ/タイプフィールドで指定された期待値に一致しません。長さ/タイプフィールドがタイプの値を含んでいる場合、フレーム長はこのフィールドに一致している（無効なフレームではない）とみなされます。
- フレーム長が、整数のバイト数ではありません（エキストラビット）。
- 受信したフレームについて計算された CRC 値が、フレームに含まれている FCS に一致しません。

## 38.5.2 MAC フレームの送信

DMA は、送信パスのすべてのトランザクションを制御します。システムメモリから読み出されたイーサネットフレームは、DMA によって FIFO にプッシュされます。その後、フレームは、ポップアウトされて、MAC コアに転送されます。フレームの終わり (End-of-frame) が転送されると、送信ステータスが MAC コアから取り出されて、DMA に返送されます。送信 FIFO の深さは 2 KB です。FIFO のフィルレベルは DMA に示されるので、DMA は、AHB インタフェースを使用して、要求されたバーストでシステムメモリからのデータフェッチを開始できます。AHB マスタインタフェースからのデータが、FIFO にプッシュされます。

SOF が検出されると、MAC はデータを受け入れて、MII への送信を開始します。アプリケーションが送信を開始した後、MII にフレームデータを送信するのに必要な時間は、IFG 遅延、プリアンプル / SFD の送信時間、半二重モードでのバックオフ遅延などの遅延要素に依存して変わります。EOF が MAC コアに転送されると、コアは通常送信を完了して、DMA に送信ステータスを返送します。送信中に通常のコリジョン (半二重モード) が発生した場合、MAC コアは、送信ステータスを有効にして、次の SOF を受信するまで、その後のすべてのデータを受け入れてドロップします。MAC から再試行リクエストがあったときには (ステータス内で)、同じフレームが SOF から再送信されなければなりません。送信中にデータが連続的に供給されなかった場合、MAC は、アンダーフローステータスを発行します。フレームの通常転送中に、MAC が前のフレームの EOF を受信していないのに SOF を受信した場合、SOF は無視され、新しいフレームが前のフレームの続きとみなされます。

MAC コアへのデータのポッピングには、2 つの動作モードがあります。

- 閾値モードでは、FIFO 内のバイトの数が設定された閾値レベルを超えると (または、閾値を超える前にフレームの終わりが書き込まれたとき)、データをポップアウトして、MAC コアに転送できます。閾値レベルは、ETH\_DMABMR の TTC ビットを使用して設定されます。
- ストアアンドフォワードモードでは、FIFO に完全なフレームが格納された後でのみ、フレームは MAC コアに向けてポップ可能になります。Tx FIFO のサイズが、送信されるイーサネットフレームより小さい場合、Tx FIFO がほぼ満杯になったとき、フレームは MAC コアに向けてポップされます。

アプリケーションは、FTF ビット (ETH\_DMAOMR レジスタ [20]) をセットすることによって、送信 FIFO のすべての内容をフラッシュできます。このビットは自動的にクリアされ、FIFO ポインタをデフォルト状態に初期化します。MAC コアへのフレーム転送中に FTF ビットがセットされた場合、FIFO はエンptyであるとみなされるので、転送は停止されます。したがって、MAC トランスミッタでアンダーフローイベントが発生し、対応するステータスワードが DMA に転送されます。

### CRC およびパッドの自動生成

アプリケーションから受信したバイト数が 60 (DA+SA+LT+ データ) 未満のときには、データ長を正確に 46 バイトにして、IEEE 802.3 の最小データフィールド要件を満たすために、送信フレームにゼロが付加されます。MAC は、パディングを付加しないようにプログラムできます。フレームチェックシーケンス (FCS) フィールドの巡回冗長検査 (CRC) が計算され、送信データに付加されます。MAC がイーサネットフレームの最後に CRC 値を付加しないようにプログラムされているときには、計算された CRC は送信されません。このルールの例外として、MAC が 60 バイト未満のフレーム (DA+SA+LT+ データ) にパッドを付加するようにプログラムされているときには、パッドが付加されたフレームの最後に CRC が付加されます。

CRC ジェネレータは、イーサネットフレームの FCS フィールドの 32 ビットの CRC を計算します。エンコードは、次の多項式によって定義されます。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

## 送信プロトコル

MAC は、イーサネットフレームの送信動作を制御します。IEEE 802.3/802.3z 仕様を満たすために、以下の機能を実行します。MAC は、

- プリアンブルと SFD を生成します。
- 半二重モードでジャムパターンを生成します。
- ジャバタイムアウトを制御します。
- 半二重モードのフローを制御します (バックプレッシャ)。
- 送信フレームのステータスを生成します。
- IEEE 1588 に従って、タイムスタンプスナップショットロジックを含んでいます。

新しいフレーム送信が要求されると、MAC は、プリアンブルと SFD を送出してから、データを送出します。プリアンブルは、7 バイトの 0b10101010 パターンとして定義され、SFD は、1 バイトの 0b10101011 パターンとして定義されます。コリジョンウィンドウは、1 スロット時間として定義されます (10/100 Mbit/s イーサネットの場合、512 ビット時間)。ジャムパターンの生成は、半二重モードにのみ適用され、全二重モードには適用されません。

MII モードでは、フレームの開始から CRC フィールドの終わりまでの任意の時点でコリジョンが発生した場合、MAC は、MII で 32 ビットの 0x5555 5555 ジャムパターンを送信して、コリジョンが発生したステーション以外のすべてのステーションに通知します。プリアンブル送信フェーズ中にコリジョンが検出された場合、MAC は、プリアンブルと SFD の送信を完了した後、ジャムパターンを送信します。

2048 バイト (デフォルト) を超えるフレームを転送する必要がある場合、イーサネットフレームの送信を中断するために、ジャバタイムが保持されます。MAC は、半二重モードでは、フロー制御のための遅延メカニズム (バックプレッシャ) を使用します。アプリケーションがフレームの受信を停止するように要求すると、MAC は、フレームの受信を検出したときに 32 バイトのジャムパターンを送信します。ただし、送信フロー制御が有効な場合に限り、これによりコリジョンが発生し、リモートステーションがバックオフします。アプリケーションは、ETH\_MACFCR レジスタの BPA ビット (ビット 0) をセットすることによって、フロー制御を要求します。アプリケーションがフレームの送信を要求した場合、バックプレッシャが有効なときでも、送信がスケジュールされ、実行されます。バックプレッシャが長時間有効なままの場合 (および 16 を超えるコリジョンイベントが連続して発生した場合)、リモートステーションは過剰コリジョンのために送信を中止します。送信フレームに対して IEEE 1588 タイムスタンプが有効な場合、このブロックは SFD が送信 MII バスに送出されたときにシステム時間のスナップショットをとります。

## 送信スケジューラ

MAC は、MII でのフレーム送信のスケジュールを処理します。2 つの送信フレームの間にフレーム間ギャップを維持し、半二重モードでは、TBEB (Truncated Binary Exponential Backoff) アルゴリズムに従います。MAC は、IFG およびバックオフ遅延を満たした後、送信を有効にします。任意の 2 つの送信フレーム間に設定されたフレーム間ギャップ (ETH\_MACCCR レジスタの IFG ビット) のアイドル期間を維持します。送信フレームが、設定された IFG 時間より早く到着した場合、MII は、MAC からのイネーブル信号を待ってから、送信を開始します。MAC は、MII のキャリア信号が無効に移行すると、IFG カウンタを開始します。プログラムされた IFG 値の最後で、MAC は、全二重モードでの送信を有効にします。半二重モードで IFG が 96 ビット時間に設定されている場合、MAC は、IEEE802.3 仕様のセクション 4.2.3.2.1 で規定されている準拠規則に従います。MAC は、IFG インターバルの最初の 3 分の 2 の期間 (すべての IFG 値の 64 ビット時間) にキャリアが検出された場合、IFG カウンタをリセットします。IFG インターバルの最後の 3 分の 1 の期間にキャリアが検出された場合、MAC は、IFG カウンタを続行して、IFG インターバル後にトランスミッタを有効にします。MAC は、半二重モードで動作するときには、TBEB アルゴリズムを実行します。

## 送信フロー制御

送信フロー制御イネーブルビット (ETH\_MACFCR レジスタの TFE ビット) がセットされると、MAC は、ポーズフレームを生成して、必要に応じて全二重モードで送信します。ポーズフレームは、計算された CRC が付加されて送信されます。ポーズフレームの生成は 2 つの方法で開始できます。

ポーズフレームは、アプリケーションが ETH\_MACFCR レジスタの FCB ビットをセットしたときか、受信 FIFO が満杯 (パケットバッファ) になったときに送信されます。

- アプリケーションが ETH\_MACFCR レジスタの FCB ビットをセットすることによってフロー制御を要求した場合、MAC は 1 つのポーズフレームを生成して送信します。生成されたフレーム内の一時停止時間の値は、ETH\_MACFCR でプログラムされた一時停止時間の値を含みます。一時停止時間を延長したり、前に送信されたポーズフレームで指定された時間の前に一時停止を終了するには、アプリケーションは、一時停止時間の値 (ETH\_MACFCR レジスタの PT) を適切な値でプログラムしてから、別のポーズフレームの送信を要求する必要があります。
- 受信 FIFO が満杯状態のときにアプリケーションがフロー制御を要求した場合、MAC はポーズフレームを生成して送信します。生成されたフレーム内の一時停止時間の値は、ETH\_MACFCR でプログラムされた一時停止時間の値です。受信 FIFO が、この一時停止時間が経過する前に、設定可能なスロット時間数 (ETH\_MACFCR の PLT ビット) の間、満杯状態のままだった場合、2 番目のポーズフレームが送信されます。このプロセスは、受信 FIFO が満杯である限り繰り返されます。サンプリング時間の前に、この条件が満たされなくなった場合、MAC は、一時停止時間がゼロのポーズフレームを送信して、受信バッファが新しいデータフレームを受け取る準備ができたことをリモートエンドに知らせます。

## シングルパケット送信動作

送信動作の一般的なイベントシーケンスは、以下のとおりです。

1. システムが送信すべきデータを持っている場合、DMA コントローラは、それらを AHB インタフェースを通じてメモリからフェッチして、FIFO への転送を開始します。フレームの終わりが転送されるまで、データの受信を続けます。
2. 閾値レベルを超えたとき、または完全なデータパケットが FIFO に受信されたとき、フレームデータはポップされ、MAC コアに送られます。DMA は、完全なパケットが MAC に転送されるまで、FIFO からのデータの転送を続けます。フレームの完了時、DMA コントローラには、MAC からのステータス情報で通知されます。

## 送信動作 — バッファに 2 つのパケットがある場合

1. DMA はホストに制御を渡す前にディスクリプタのステータスを更新しなければならないので、送信 FIFO に 2 つのフレームが存在することがあります。2 番目のフレームは、OSF (Operateon Second Frame) ビットがセットされている場合にのみ、DMA によってフェッチされて、FIFO に格納されます。このビットがセットされていない場合、次のフレームは、MAC がフレームを完全に処理して、DMA がディスクリプタをリリースした後でのみ、メモリからフェッチされます。
2. OSF ビットがセットされている場合、DMA は、最初のフレームの FIFO への転送を完了した直後に、2 番目のフレームのフェッチを開始します。この場合、ステータスの更新を待ちません。すなわち、最初のフレームの送信中に、2 番目のフレームが FIFO に受信されます。最初のフレームが転送され、MAC からステータスが受信されると、DMA にプッシュされます。DMA が FIFO への 2 番目のパケットの送信をすでに完了している場合、2 番目の送信は、最初のパケットのステータスを待ってから、次のフレームに進む必要があります。

## 衝突時の再送信

半二重モードでは、フレームが MAC に転送されているときに、MAC ラインインタフェースでコリジョンイベントが発生することがあります。その場合、MAC は、フレームの終わりが受信される前でも、ステータスを与えることによって、再試行を示します。再送信が有効になり、FIFO からフレームが再びポップアウトされます。MAC コアに対して 96 バイトを超えるデータがポップアウトされた後、FIFO コントローラは、そのスペースを解放して、DMA がさらにデータをプッシュできるようにします。これは、この閾値を越えるか、MAC コアがレイトコリジョンイベントを示したときには、再送信ができないことを意味します。

## 送信 FIFO のフラッシュ動作

MAC は、動作モードレジスタのビット 20 を使用して、送信 FIFO をフラッシュするための制御をソフトウェアに与えます。フラッシュ動作はただちに実行され、Tx FIFO と対応するポインタは、Tx FIFO が MAC コアへのフレームの転送中でも、初期状態にクリアされます。これにより、MAC トランスミッタにアンダーフローイベントが発生し、フレームの送信は中止されます。このようなフレームのステータスは、アンダーフローとフレームフラッシュの両方のイベント (TDES0 ビットの 13 および 1) でマークされます。フラッシュ動作中は、アプリケーション (DMA) から FIFO へのデータ移動は行われません。フラッシュされたフレーム数 (部分的フレームも含む) を示す転送送信ステータスワードがアプリケーションに転送されます。完全にフラッシュされたフレームは、フレームフラッシュステータスビット (TDES0 ビットの 13) がセットされます。フラッシュ動作は、アプリケーション (DMA) が、フラッシュされたフレームのすべてのステータスワードを受け取ったときに完了します。次に、送信 FIFO フラッシュ制御レジスタのビットがクリアされます。この時点で、アプリケーション (DMA) から新しいフレームが受け取られます。フラッシュ動作後に残っている送信データは、SOF マーカーで始まっていない限り、すべて破棄されます。

## 送信ステータスワード

MAC コアへのイーサネットフレーム転送の最後と、コアがフレームの送信を完了した後に、送信ステータスがアプリケーションに送られます。送信ステータスの詳細な説明は、TDES0 のビット [23:0] と同じです。IEEE 1588 タイムスタンプが有効な場合、特定のフレームの 64 ビットのタイムスタンプが送信ステータスとともに返されます。

## 送信チェックサムオフロード

TCP や UDP などの通信プロトコルは、ネットワーク経由で送信されたデータの完全性を確認できるチェックサムフィールドを実装しています。最も広く利用されているイーサネットの用途は、IP データグラムによる TCP や UDP のカプセル化なので、イーサネットコントローラは、チェックサム計算と送信パスへの挿入、および受信パスでのエラー検出をサポートする送信チェックサムオフロード機能を備えています。このセクションでは、送信フレームのチェックサムオフロード機能の動作について説明します。

**注 :** TCP、UDP、または ICMP のチェックサムは、完全なフレームに対して計算された後、対応するヘッダフィールドに挿入されます。この要件のため、この機能は、送信 FIFO がストアアンドフォワードモードに設定されているとき (すなわち、ETH\_ETH\_DMAOMR レジスタの TSF ビットがセットされているとき) のみ有効です。コアが閾値 (カットスルー) モードに設定されている場合、送信チェックサムオフロードはバイパスされます。

ユーザは、フレームが MAC コアトランスミッタに転送される前に、送信 FIFO が完全なフレームを格納できる十分な深さであることを確認する必要があります。FIFO の深さが入力されたイーサネットフレームのサイズより小さい場合、ストアアンドフォワードモードでも、ペイロード (TCP/UDP/ICMP) チェックサム挿入機能はバイパスされ、フレームの IPv4 ヘッダのチェックサムのみが変更されます。

送信チェックサムオフロードは、2 種類のチェックサム計算および挿入をサポートします。チェックサムは、CIC ビット ([TDES1 : 送信ディスクリプタワード1 \(1525 ページ\)](#)) で説明されている TDES1 のビット 28:27) をセットすることによって、フレームごとに制御できます。

IPv4、TCP、UDP、ICMP、IPv6、および ICMPv6 パケットヘッダの仕様については、パケットヘッダの仕様については、IETF 仕様の RFC 791、RFC 793、RFC 768、RFC 792、RFC 2460、および RFC 4443 をそれぞれ参照してください。

## ● IP ヘッダのチェックサム

IPv4 データグラムでは、ヘッダフィールドの完全性は、16 ビットのヘッダチェックサムフィールド (IPv4 データグラムの 11 番目と 12 番目のバイト) によって示されます。チェックサムオフロードは、イーサネットフレームのタイプフィールドの値が 0x0800 であり、IP データグラムのバージョンフィールドの値が 0x4 のときに、IPv4 データグラムを検出します。入力フレームのチェックサムフィールドは、計算時には無視され、計算された値に置き換えられます。IPv6 ヘッダにはチェックサムフィールドがないので、チェックサムオフロードは IPv6 ヘッダフィールドを変更しません。この結果、IP ヘッダチェックサムの計算結果は、送信ステータス (ビット 16) の IP ヘッダエラーステータスビットによって示されます。このステータスビットは、イーサネットタイプフィールドと IP ヘッダのバージョンフィールドの値が一致しないとき、またはイーサネットフレームに十分なデータがない (IP ヘッダの長さフィールドによって示されます) ときにセットされます。言い換えると、このビットは、IP ヘッダエラーが以下のような状況でアサートされたときにセットされます。

### a) IPv4 データグラムの場合 :

- 受信したイーサネットのタイプは 0x0800 であるが、IP ヘッダのバージョンフィールドが 0x4 に等しくないとき。
- IPv4 ヘッダ長フィールドが 0x5 (20 バイト) 未満の値を示しているとき。
- 合計フレーム長が IPv4 ヘッダ長フィールドで指定された値より小さいとき。

### b) IPv6 データグラムの場合 :

- イーサネットのタイプは 0x86DD だが、IP ヘッダのバージョンフィールドが 0x6 に等しくないとき。
- IPv6 ヘッダ (40 バイト) または拡張ヘッダ (拡張ヘッダ内の対応するヘッダ長フィールドで指定) が完全に受信される前にフレームが終了したとき。チェックサムオフロードがこのような IP ヘッダエラーを検出したときでも、イーサネットタイプフィールドが IPv4 ペイロードを示している場合は、IPv4 ヘッダチェックサムを挿入します。

- TCP/UDP/ICMP チェックサム

TCP/UDP/ICMP チェックサムは、IPv4 または IPv6 ヘッダ (拡張ヘッダを含む) を処理して、カプセル化されたペイロードが TCP、UDP、ICMP のいずれかを判断します。

以下のことに注意してください。

- a) 非 TCP、非 UDP、または非 ICMP/ICMPv6 ペイロードの場合、このチェックサムはバイパスされ、フレームには何の変更も加えられません。
- b) 断片化された IP フレーム (IPv4 または IPv6)、セキュリティ機能を持つ IP フレーム (認証ヘッダやカプセル化されたセキュリティペイロードなど)、およびルーティングヘッダを持つ IPv6 フレームは、バイパスされ、チェックサムによる処理は行われません。

TCP、UDP、または ICMP ペイロードのチェックサムが計算され、ヘッダ内の対応するフィールドに挿入されます。チェックサムは次の 2 つのモードで動作します。

- 最初のモードでは、TCP、UDP、または ICMPv6 擬似ヘッダは、チェックサムの計算に含まれず、入力フレームのチェックサムフィールドに存在するとみなされます。チェックサムフィールドはチェックサムの計算に含まれ、最終的に計算されたチェックサムで置き換えられます。
- 第 2 のモードでは、チェックサムフィールドは無視され、TCP、UDP、または ICMPv6 擬似ヘッダデータはチェックサムの計算に含まれ、チェックサムフィールドは最終的な計算値で上書きされます。

ICMP-over-IPv4 パケットの場合、擬似ヘッダは定義されていないので、ICMP パケット内のチェックサムフィールドは、どちらのモードでも常に 0x0000 でなければならないことに注意してください。0x0000 に等しくない場合、正しくないチェックサムがパケットに挿入される可能性があります。

この操作の結果は、送信ステータスペクタのペイロードチェックサムエラーステータスビット (ビット 12) によって示されます。ペイロードチェックサムエラーステータスビットは、次のどちらかが検出されたときにセットされます。

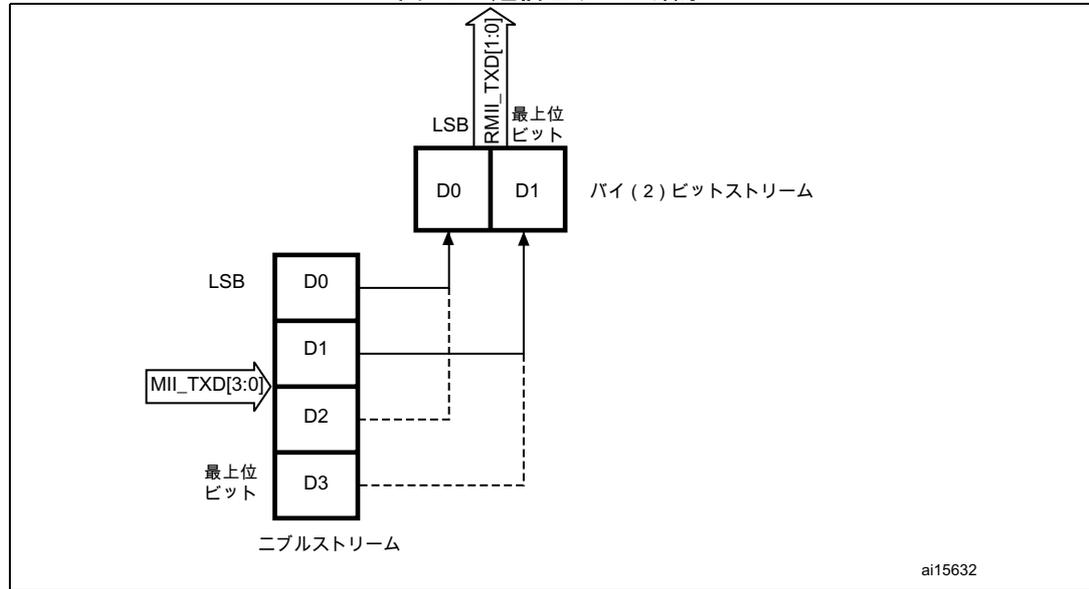
- フレームがストアアンドフォワードモードで MAC トランスミッタに転送されたが、フレームの終わりが FIFO に書き込まれなかったとき。
- IP ヘッダのペイロード長フィールドによって示されているバイト数が受信される前に、パケットが終了したとき。

パケットが、示されているペイロード長よりも長いときには、バイトはスタッフバイトとして無視され、エラーは報告されません。最初のタイプのエラーが検出されたとき、TCP、UDP、または ICMP ヘッダは変更されません。2 番目のタイプのエラーの場合、計算されたチェックサムが、対応するヘッダフィールドに挿入されます。

## MII/RMII 送信ビットの順序

MII からの各ニブルは、RMII で一度に 2 ビット (双ビット) ずつ、[図 481](#) に示されている双ビット送信の順序で送信されます。下位ビット (D1 と D0) が最初に送信され、次に上位ビット (D2 と D3) が送信されます。

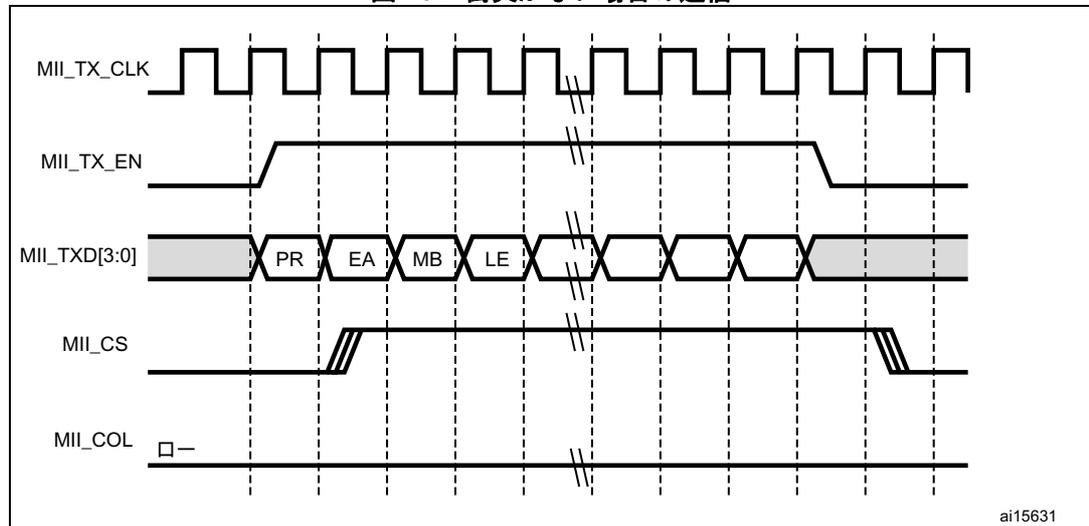
図 481. 送信ビットの順序



ai15632

## MII/RMII 送信タイミング図

図 482. 衝突がない場合の送信



ai15631

図 483. 衝突がある場合の送信

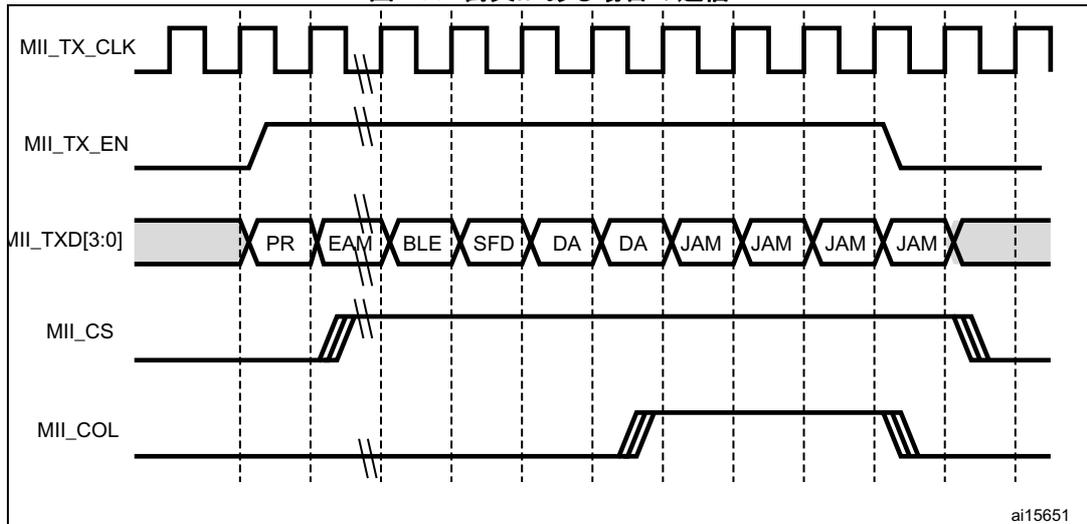
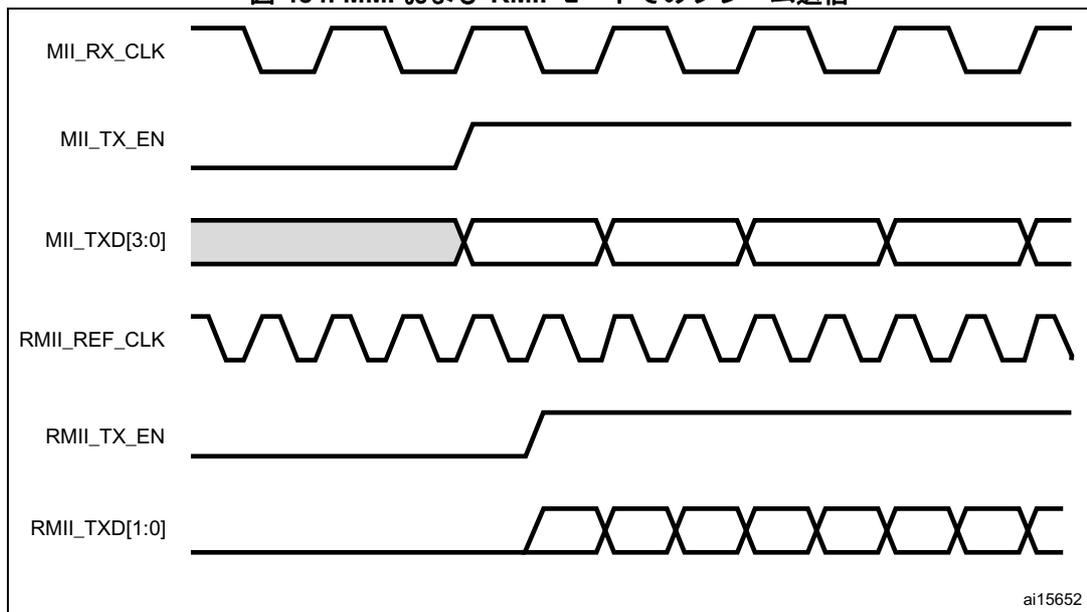


図 484 に、MII および RMII でのフレーム送信を示します。

図 484. MII および RMII モードでのフレーム送信



### 38.5.3 MAC フレームの受信

MAC が受信したフレームは、Rx FIFO にプッシュされます。この FIFO のステータス（フィルレベル）が設定された受信閾値（ETH\_DMAOMR レジスタの RTC）を超えると、そのことが DMA に示されるので、DMA は、AHB インタフェースに対して、事前設定されたバースト転送を開始できます。

デフォルトのカットスルーモードでは、64 バイト（ETH\_DMAOMR レジスタの RTC フィールドで設定）またはデータパケット全体が FIFO に受信されたときに、データがポップアウトされ、データが使用可能になったことが DMA に通知されます。DMA が AHB インタフェースへの転送を開始すると、FIFO からのデータ転送は、完全なパケットが転送されるまで続けられます。EOF フレームの転送が完了すると、ステータスワードがポップアウトされて、DMA コントローラに送信されます。

Rx FIFO のストアアンドフォワードモード (ETH\_DMAOMR レジスタの RSF ビットで設定) では、フレームは、受信 FIFO に完全に書き込まれた後にのみ読み出されます。このモードでは、すべてのエラーフレームがドロップされるので (そのようにコアが設定されている場合)、有効なフレームのみが読み出されて、アプリケーションに転送されます。カットスルーモードでは、エラーステータスはフレームの最後に受信され、その時点では、フレームの最初の部分がすでに FIFO から読み出されてしまっているため、一部のエラーフレームはドロップされません。

受信動作は、MAC が MII で SFD を検出したときに開始されます。コアは、フレームの処理を進める前に、プリアンプルと SFD を削除します。フレームの CRC を確認するために、使用されたフィルタリングと FCS フィールドがヘッダフィールドでチェックされます。アドレスフィルタでエラーが発生した場合、そのフレームはコアでドロップされます。

## 受信プロトコル

受信フレームのプリアンプルと SFD は削除されます。SFD が検出されると、MAC は、イーサネットフレームデータの受信 FIFO への送信を、最初のバイトから始めて、次に SFD (デスティネーションアドレス) という順序で開始します。IEEE タイムスタンプが有効な場合、任意のフレームの SFD が MII で検出されたとき、システム時間のスナッチショットがとられます。MAC がフレームをフィルタリングしてドロップしない限り、このタイムスタンプはアプリケーションに渡されます。

受信フレームの長さ/タイプフィールドが 0x600 未満の場合と、MAC の自動 CRC/パッドストリッピングオプションがプログラムされていた場合には、MAC は、フレームのデータを長さ/タイプフィールドで指定されたカウントまで Rx FIFO に送信してから、バイトのドロッピング (FCS フィールドを含む) を開始します。長さ/タイプフィールドが 0x600 以上の場合、MAC は、プログラムされた自動 CRC ストリッピングオプションの値に関係なく、受信したすべてのイーサネットフレームデータを Rx FIFO に送信します。MAC ウォッチドッグタイマは、デフォルトで有効なので、2048 バイト以上のフレーム (DA + SA + LT + データ + パッド + FCS) は切り捨てられます。この機能は、MAC 設定レジスタのウォッチドッグディセーブル (WD) ビットをプログラムすることによって無効にできます。ただし、ウォッチドッグタイマが無効な場合でも、サイズが 16 KB を超えるフレームは切り捨てられ、ウォッチドッグタイムアウトステータスが与えられます。

## 受信 CRC : 自動 CRC およびパッドストリッピング

MAC は、受信フレームの CRC エラーをチェックします。デスティネーションアドレスフィールドから FCS フィールドまでを含めて、受信フレームの 32 ビットの CRC を計算します。エンコードは、次の多項式によって定義されます。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

自動パッド/CRC ストリッピングに関係なく、MAC は、フレーム全体を受信して、受信フレームの CRC チェックを計算します。

## 受信チェックサムオフロード

データの完全性チェックのために、受信イーサネットフレームの IPv4 または IPv6 が検出され、処理されます。受信チェックサムオフロード機能は、ETH\_MACCCR レジスタの IPCO ビットをセットすることによって有効にできます。MAC レシーバは、受信したイーサネットフレームのタイプフィールドの値が 0x0800 か 0x86DD かをチェックすることによって、そのフレームが IPv4 か IPv6 かを識別します。この識別は、VLAN タグ付きフレームにも適用されます。受信チェックサムオフロードは、IPv4 ヘッダのチェックサムを計算して、受信した IPv4 ヘッダのチェックサムと一致するかどうかを確認します。示されているペイロードタイプ (イーサネットタイプフィールド) と IP ヘッダバージョン間に不一致があった場合、または、受信フレームのバイト数が、IPv4 ヘッダの長さフィールドで示されているバイト数より少ないとき (または、IPv4 または IPv6 ヘッダで使用可能なスペースが 20 バイト未満のとき) には、IP ヘッダエラービットがセットされます。受信チェックサムオフロードは、

受信した IP データグラム (IPv4 または IPv6) のペイロードが TCP、UDP、または ICMP のいずれであるかも識別して、TCP、UDP、または ICMP 仕様での定義に従って、それぞれのペイロードのチェックサムを計算します。チェックサムの計算には、TCP/UDP/ICMPv6 擬似ヘッダバイトも含まれ、受信したチェックサムフィールドの値が計算値に一致するかどうかをチェックします。この操作の結果は、受信ステータスワードのペイロードチェックサムエラービットとして与えられます。このステータスビットは、TCP、UDP、または ICMP ペイロードが IP ヘッダで示された期待されたペイロード長に一致しない場合にもセットされます。[TCP/UDP/ICMP チェックサム \(1492 ページ\)](#) で述べたように、受信チェックサムオフロードは、断片化した IP データグラム、セキュリティ機能付き IP データグラム、IPv6 ルーティングヘッダ、および TCP、UDP、または ICMP 以外のペイロードをバイパスします。この情報 (チェックサムがバイパスされるかどうか) は、[RDES0 : 受信ディスクリプタワード 0](#) セクションで説明されているように、受信ステータスで与えられます。この設定では、コアは、受信したイーサネットフレームにペイロードチェックサムバイトを付加しません。

[RDES0 : 受信ディスクリプタワード 0 \(1531 ページ\)](#) で説明されているように、特定のレジスタビットの意味は、[表 240](#) のように変わります。

表 240. フレームのステータス

ビット 18 : イーサネット フレーム	ビット 27 : ヘッダチェック サムエラー	ビット 28 : ペイロード チェックサムエラー	フレームステータス
0	0	0	フレームは IEEE 802.3 フレームです (長さフィールドの値が 0x0600 未満)。
1	0	0	チェックサムエラーが検出されない IPv4/IPv6 タイプのフレーム。
1	0	1	ペイロードチェックサムエラー (PCE の説明と同様) が検出される IPv4/IPv6 タイプのフレーム。
1	1	0	IP ヘッダチェックサムエラー (IPCO HCE の説明と同様) が検出される IPv4/IPv6 タイプのフレーム。
1	1	1	PCE と IPCO HCE の両方が検出される IPv4/IPv6 タイプのフレーム。
0	0	1	IP HCE がなく、サポートされていないペイロードであるためにペイロードチェックがバイパスされる IPv4/IPv6 タイプのフレーム。
0	1	1	IPv4 でも IPv6 でもないタイプのフレーム (チェックサムオフロードはチェックサムのチェックを完全にバイパスします)。
0	1	0	予約済み

## 受信フレームコントローラ

MAC CSR フレームフィルタレジスタの RA ビットがリセットされた場合、MAC は、デスティネーション/ソースアドレスに基づいて、フレームフィルタリングを行います (その場合でも、アプリケーションは、ラントフレームや CRC エラーフレームなどの不良フレームを受信しないと決めた場合、別のレベルのフィルタリングを実行する必要があります)。フィルタ失敗が検出されると、フレームはドロップされ、アプリケーションに転送されません。フィルタリングパラメータが動的に変更されたとき、(DA-SA) フィルタ失敗の場合、フレームの残りはドロップされ、Rx ステータスワードがただちに更新されて (ゼロフレーム長、CRC エラー、およびラントエラービットがセットされます)、フィルタ失敗を示します。イーサネットパワーダウンモードでは、受信されたすべてのフレームがドロップされ、アプリケーションに転送されません。

## 受信フロー制御

MAC は、ポーズフレームの受信を検出して、受信したポーズフレームで指定された遅延時間の間、フレーム送信を一時停止します(全二重モードの場合のみ)。ポーズフレーム検出機能は、ETH\_MACFCR レジスタの RFCE ビットで有効または無効にできます。受信フロー制御が有効にされると、受信フレームのデスティネーションアドレスの監視が開始され、制御フレームのマルチキャストアドレス (0x0180 C200 0001) と一致しているかどうかを確認されます。一致が検出された場合 (受信フレームのデスティネーションアドレスが受信制御フレームのデスティネーションアドレスと一致)、MAC は、ETH\_MACFFR レジスタの PCF ビットのレベルに基づいて、受信した制御フレームをアプリケーションに転送するかどうかを決定します。

MAC は、受信した制御フレームのタイプ、OP コード、およびポーズタイムフィールドのデコードも行います。ステータスのバイトカウントが 64 バイトを示している場合と、CRC エラーがない場合、MAC トランスミッタは、デコードしたポーズ時間の値にスロット時間 (10/100 Mbit/s の両方のモードについて 64 バイト時間) を掛けた時間の間、フレーム送信を一時停止します。一方、ポーズ時間値がゼロの別のポーズフレームが検出された場合には、MAC は、ポーズ時間をリセットして、この新しいポーズリクエストを管理します。

受信した制御フレームがタイプフィールド (0x8808)、OP コード (0x00001)、またはバイト長 (64 バイト) のいずれにも一致しなかった場合、MAC はポーズを生成しません。

マルチキャストデスティネーションアドレスを持つポーズフレームの場合、MAC は、アドレスの一致に基づいてフレームをフィルタします。

ユニキャストデスティネーションアドレスを持つポーズフレームの場合、MAC のフィルタリングは、DA が MAC アドレス 0 レジスタの内容に一致しているかどうか、および ETH\_MACFCR レジスタの UPDF ビットがセットされている (ユニキャストデスティネーションアドレスを持つポーズフレームでも検出する) かどうかによって依存します。PCF レジスタビット (ETH\_MACFFR レジスタのビット[7:6]) は、アドレスフィルタリングに加えて、制御フレームのフィルタリングを制御します。

## 受信動作におけるマルチフレームの処理

ステータスはデータの直後に使用可能なため、FIFO は、満杯でない限り、任意のフレーム数を格納できます。

## エラー処理

MAC から EOF データを受信する前に Rx FIFO が満杯になった場合は、オーバーフローが宣言され、フレーム全体がドロップされて、ETH\_DMAMFBOCR レジスタのオーバーフローカウンタがインクリメントされます。ステータスは、オーバーフローによる部分的フレームであることを示します。Rx FIFO は、(ETH\_DMAOMR レジスタの FEF および FUGF ビットを使用して) 有効にされた場合、エラーおよびアンダサイズフレームをフィルタできます。

受信 FIFO がストアアンドフォワードモードで動作するように設定されている場合、すべてのエラーフレームがフィルタされ、ドロップされます。

カットスルーモードでは、フレームの SOF が Rx FIFO から読み出されたときにフレームのステータスおよび長さが使用可能な場合、エラーフレームを完全にドロップすることができます。DMA は、受信フレームフラッシュビットを有効にすることによって、FIFO から読み出したエラーフレームをフラッシュできます。アプリケーション (DMA) へのデータ転送は停止し、フレームの残りが内部で読み出されて、ドロップされます。使用可能な場合、次のフレーム転送を開始できます。

## 受信ステータスワード

イーサネットフレーム受信の最後に、MAC は、アプリケーション (DMA) に対して受信ステータスを出力します。受信ステータスの詳細な説明は、RDES0 のビット [31:0] の説明と同じです。 [RDES0 : 受信ディスクリプタワード 0 \(1531 ページ\)](#) を参照してください。

## フレーム長インタフェース

スイッチアプリケーションの場合、アプリケーションと MAC の間でのデータの送受信は、完全なフレーム転送として発生します。アプリケーションレイヤは、フレームを出力ポートに転送するために、入力ポートから受信したフレームの長さを認識する必要があります。MAC コアは、各フレームの受信の最後のステータス内で、受信した各フレームのフレーム長を示します。

**注 :** オーバーフローにより、Rx FIFO に書き込まれた部分フレームについては、フレーム長の値として 0 が与えられます。

## MII/RMII 受信ビットの順序

各ニブルは、RMII から受信した双ビットから MII に、[図 485](#) に示されているニブル送信順序で送信されます。下位ビット (D1 と D0) が最初に受信され、次に上位ビット (D2 と D3) が受信されます。

図 485. 受信ビットの順序

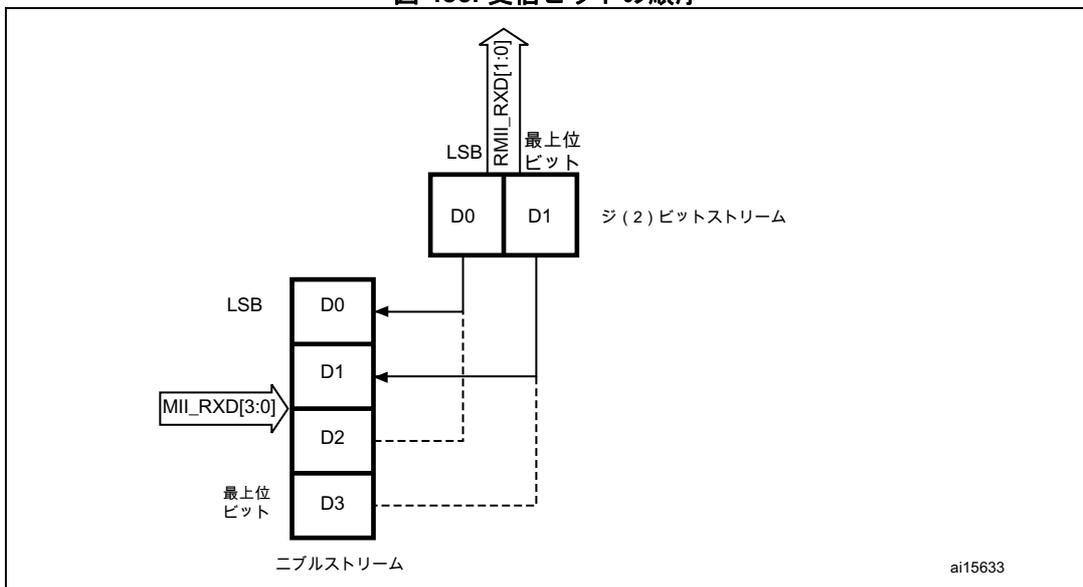


図 486. エラーなしでの受信

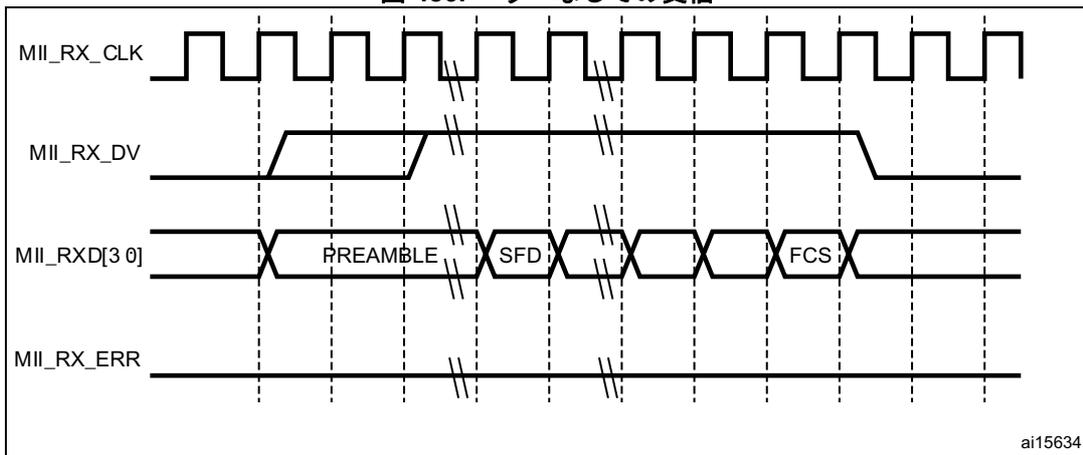


図 487. エラーありでの受信

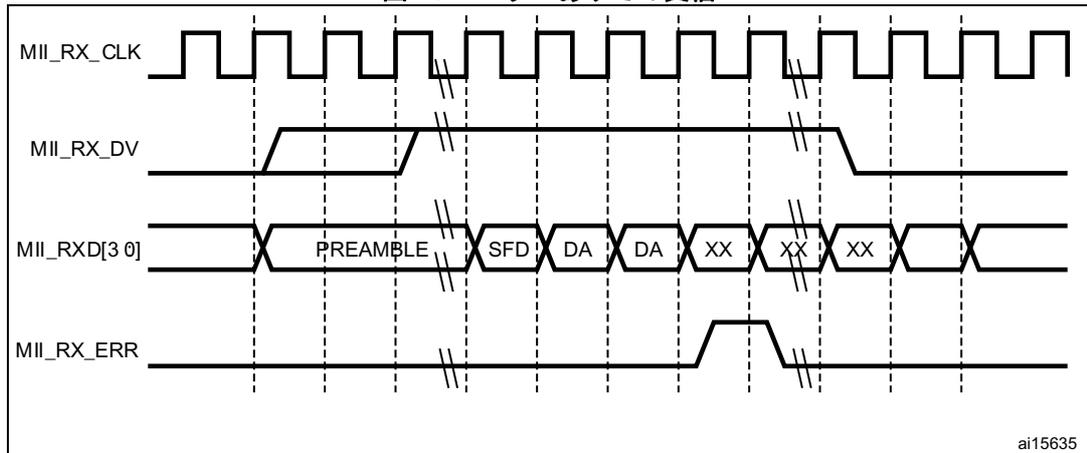
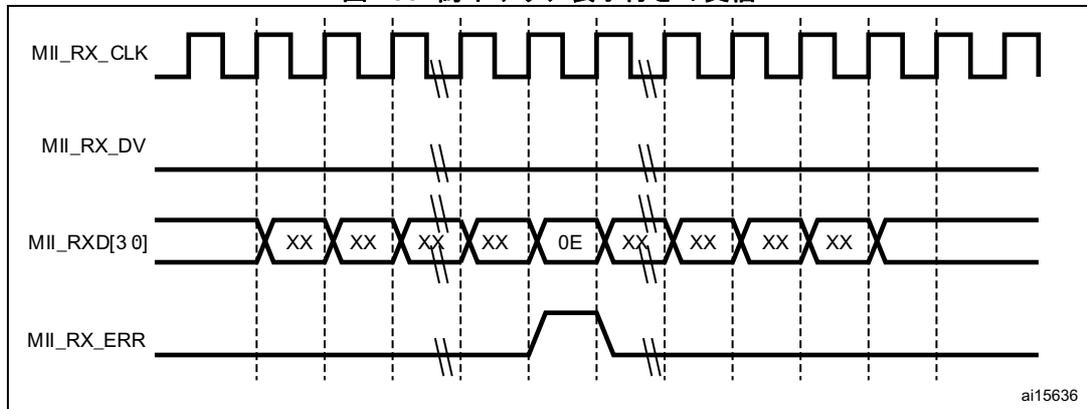


図 488. 偽キャリア表示付きの受信



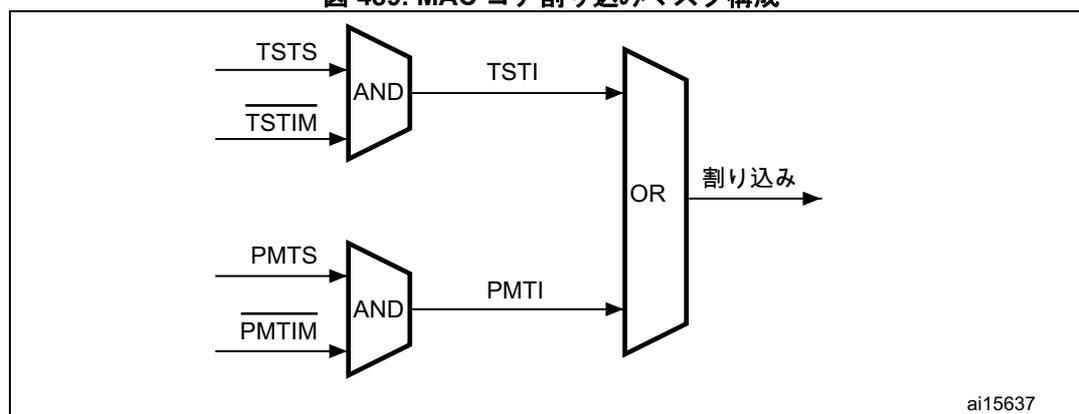
## 38.5.4 MAC 割り込み

さまざまなイベントの結果として、MAC コアから割り込みが生成されることがあります。

ETH\_MACCSR レジスタは、MAC コアからの割り込みの原因となるイベントをディスクリプタします。割り込みマスクレジスタの対応するマスクビットをセットすることによって、各イベントが割り込みをアサートしないようにすることができます。

割り込みレジスタビットは、イベントが報告されるブロックを示すだけです。割り込みをクリアするには、対応するステータスレジスタと他のレジスタを読み出す必要があります。たとえば、割り込みレジスタのビット 3 がハイにセットされているときには、パワーダウンモードでマジックパケットまたは Wake-on-LAN フレームが受信されたことを示します。この割り込みをクリアするには、ユーザは ETH\_MACPMTCSR レジスタを読み出す必要があります。

図 489. MAC コア割り込みマスク構成



ai15637

## 38.5.5 MAC フィルタリング

### アドレスフィルタリング

アドレスフィルタリングは、受信したすべてのフレームのデスティネーションアドレスとソースアドレスをチェックして、それに応じてアドレスフィルタリングステータスが報告されます。アドレスチェックは、アプリケーションによって選択されたさまざまなパラメータ（フレームフィルタレジスタ）に基づきます。フィルタされるフレームも、マルチキャストかブロードキャストかが識別されます。

アドレスフィルタリングでは、ステーションの物理 (MAC) アドレスとマルチキャストハッシュテーブルがアドレスチェックに使用されます。

### ユニキャストデスティネーションアドレスフィルタ

MAC は、ユニキャストの完全なフィルタリングのために、最大 4 つの MAC アドレスをサポートします。完全フィルタリングが選択された場合（フレームフィルタレジスタの HU ビットがリセット）、MAC は、受信したユニキャストアドレスの 48 ビットすべてを、プログラムされた MAC アドレスと比較して、一致しているかどうかを確認します。デフォルトの MacAddr0 は常に有効であり、その他のアドレス (MacAddr1~MacAddr3) は、個別のイネーブルビットで選択されます。その他のアドレス (MacAddr1~MacAddr3) の各バイトは、レジスタの対応するマスクバイト制御ビットをセットすることによって、対応する受信 DA バイトとの比較時にマスクできます。これは、DA のグループアドレスフィルタリングに役立ちます。ハッシュフィルタリングモード (HU ビットがセットされているとき) では、MAC は、64 ビットのハッシュテーブルを使用して、ユニキャストアドレスの不完全フィルタリングを実行します。ハッシュフィルタリングの場合、MAC は、受信したデスティネーションアドレスの CRC の上位 6 ビットを使用して（下記の注 1 を参照）、ハッシュテーブルの内容をインデックスします。値 000000 は、選択されたレジスタのビット 0 を選択し、値 111111 は、ハッシュテーブルレジスタのビット 63 を選択します。対応するビット (6 ビット CRC によって示される) が 1 にセットされた場合、ユニキャストフレームはハッシュフィルタを通過したとみなされ、そうでない場合、フレームはハッシュフィルタ通過できなかったとみなされます。

**注：** この CRC は、次の多項式によって符号化された 32 ビット値です（詳細については、[セクション 38.5.3 : MAC フレームの受信](#)を参照してください）。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

## マルチキャストデスティネーションアドレスフィルタ

MAC は、フレームフィルタレジスタの PAM ビットをセットすることによって、すべてのマルチキャストフレームを通過させるようにプログラムできます。PAM ビットがリセットされた場合、MAC は、フレームフィルタレジスタの HM ビットに基づいて、マルチキャストアドレスのフィルタリングを実行します。完全フィルタリングモードでは、マルチキャストアドレスが、プログラムされた MAC デスティネーションアドレスレジスタ (1~3) と比較されます。グループアドレスフィルタリングもサポートされています。ハッシュフィルタリングモードでは、MAC は、64 ビットのハッシュテーブルを使用して、不完全フィルタリングを実行します。ハッシュフィルタリングの場合、MAC は、受信したマルチキャストアドレスの CRC の上位 6 ビット (下記の注 1 を参照) を使用して、ハッシュテーブルの内容をインデックスします。値 000000 は選択されたレジスタのビット 0 を、値 111111 はハッシュテーブルレジスタのビット 63 を選択します。対応するビットが 1 にセットされた場合は、マルチキャストフレームがハッシュフィルタを通過したことを示し、そうでない場合は、フレームがハッシュフィルタを通過できなかったことを示します。

**注 :** この CRC は、次の多項式によって符号化された 32 ビット値です (詳細については、[セクション 38.5.3 : MAC フレームの受信を参照してください](#))。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

## ハッシュまたは完全アドレスフィルタ

DA フィルタは、フレームフィルタレジスタの HPF ビットをセットし、対応する HU または HM ビットをセットすることによって、DA がハッシュフィルタまたは完全フィルタに一致するときにはフレームを通過させるように設定できます。この設定は、ユニキャストフレームとマルチキャストフレームの両方に適用されます。HPF ビットがリセットされた場合は、1 つのフィルタ (ハッシュまたは完全) だけが受信フレームに適用されます。

## ブロードキャストアドレスフィルタ

デフォルトモードでは、MAC は、ブロードキャストフレームをフィルタしません。ただし、フレームフィルタレジスタの BFD ビットをセットすることによって、すべてのブロードキャストフレームを拒絶するように MAC がプログラムされた場合は、ブロードキャストフレームはドロップされます。

## ユニキャストソースアドレスフィルタ

MAC は、受信フレームのソースアドレスフィールドに基づいて、完全フィルタリングを実行することもできます。デフォルトでは、MAC は、SA フィールドを、SA レジスタでプログラムされた値と比較します。MAC アドレスレジスタ [1:3] は、対応するレジスタのビット 30 をセットすることによって、比較のために DA の代わりに SA を含めるように設定することができます。SA によるグループフィルタリングもサポートされています。フレームフィルタレジスタの SAF ビットがセットされている場合、SA フィルタを通過できなかったフレームは、MAC によってドロップされます。そうでない場合、SA フィルタの結果は、受信ステータスワード内のステータスビットとして与えられます ([RDES0 : 受信ディスクリプタワード 0](#) を参照)。

SAF ビットがセットされた場合、SA フィルタと DA フィルタの結果が AND (論理積) されて、フレームを転送する必要があるかどうかが決まります。これは、どちらかのフィルタの通過に失敗したフレームはドロップされることを意味します。フレームがアプリケーションに転送されるためには、両方のフィルタを通過する必要があります。

## 逆フィルタリング動作

デスティネーションおよびソースアドレスフィルタリングの両方について、最終出力でフィルタ一致結果を反転できるオプションがあります。これらは、それぞれフレームフィルタレジスタの DAIF および SAIF ビットによって制御されます。DAIF ビットは、ユニキャストとマルチキャストの両方の DA フレームに適用可能です。このモードでは、ユニキャスト/マルチキャストデスティネーションアドレスフィルタの結果が反転されます。同様に、SAIF ビットがセットされると、ユニキャスト SA フィルタの結果が反転されます。表 241 と表 242 に、受信フレームのタイプに基づくデスティネーションアドレスとソースアドレスのフィルタリングを示します。

表 241. デスティネーションアドレスフィルタリング

フレームタイプ	PM	HPF	HU	DAIF	HM	PAM	DB	DA フィルタ動作
ブロードキャスト	1	X	X	X	X	X	X	成功
	0	X	X	X	X	X	0	成功
	0	X	X	X	X	X	1	失敗
ユニキャスト	1	X	X	X	X	X	X	すべてのフレームをパス
	0	X	0	0	X	X	X	完全/グループフィルタ一致で成功
	0	X	0	1	X	X	X	完全/グループフィルタ一致で失敗
	0	0	1	0	X	X	X	ハッシュフィルタ一致で成功
	0	0	1	1	X	X	X	ハッシュフィルタ一致で失敗
	0	1	1	0	X	X	X	ハッシュまたは完全/グループフィルタ一致で成功
	0	1	1	1	X	X	X	ハッシュまたは完全/グループフィルタ一致で失敗
マルチキャスト	1	X	X	X	X	X	X	すべてのフレームをパス
	X	X	X	X	X	1	X	すべてのフレームをパス
	0	X	X	0	0	0	X	完全/グループフィルタ一致で成功、PCF = 0x の場合はポーズ制御フレームをドロップ。
	0	0	X	0	1	0	X	ハッシュフィルタ一致で成功、PCF = 0x の場合はポーズ制御フレームをドロップ。
	0	1	X	0	1	0	X	ハッシュまたは完全/グループフィルタ一致で成功、PCF = 0x の場合はポーズ制御フレームをドロップ。
	0	X	X	1	0	0	X	完全/グループフィルタ一致で失敗、PCF = 0x の場合はポーズ制御フレームをドロップ。
	0	0	X	1	1	0	X	ハッシュフィルタ一致で失敗、PCF = 0x の場合はポーズ制御フレームをドロップ。
	0	1	X	1	1	0	X	ハッシュまたは完全/グループフィルタ一致で失敗、PCF = 0x の場合はポーズ制御フレームをドロップ。

表 242. ソースアドレスフィルタリング

フレームタイプ	PM	SAIF	SAF	SA フィルタ動作
ユニキャスト	1	X	X	すべてのフレームをパス
	0	0	0	完全ノグループフィルタ一致で成功ステータスだが、失敗したフレームをドロップしない。
	0	1	0	完全ノグループフィルタ一致で失敗ステータスだが、フレームをドロップしない。
	0	0	1	完全ノグループフィルタ一致で成功、失敗したフレームをドロップする。
	0	1	1	完全ノグループフィルタ一致で失敗、失敗したフレームをドロップする。

### 38.5.6 MAC ループバックモード

MAC は、送信したフレームのレシーバへのループバックをサポートします。デフォルトでは、MAC のループバック機能は無効ですが、MAC ETH\_MACCR レジスタのループバックビットをプログラムすることによって有効にできます。

### 38.5.7 MAC 管理カウンタ : MMC

MAC 管理カウンタ (MMC) は、受信および送信フレームの統計を集めるために、一連のレジスタを維持します。これらのレジスタとしては、レジスタの動作を制御する制御レジスタ、生成された割り込み (受信および送信) を含む 2 つの 32 ビットレジスタ、および割り込みレジスタ (受信および送信) のマスクを含む 2 つの 32 ビットレジスタがあります。これらのレジスタには、アプリケーションからアクセスできます。各レジスタは 32 ビット幅です。

[セクション 38.8 : イーサネットレジスタの説明](#)では、さまざまなカウンタについて説明し、統計カウンタのそれぞれのアドレスを示します。このアドレスは、目的の送信ノ受信カウンタへの読み出しノ書き込みアクセスのために使用されます。

受信 MMC カウンタは、アドレスフィルタリングをパスしたフレームについて更新されます。ドロップされたフレームの統計は、ドロップされたフレームが 6 バイト未満のラントフレーム (DA バイトが完全には受信されていない) でない限り、更新されません。

#### 良好な送信および受信フレーム

送信されたフレームは、正常に送信された場合、「良好」とみなされます。言い換えると、送信されたフレームは、フレーム送信が以下のいずれかのエラーによって中止されなければ、良好なフレームです。

- + ジャバタイムアウト
- + キャリヤがないノキャリヤの喪失
- + レイトコリジョン
- + フレームアンダーフロー
- + 過剰遅延
- + 過剰コリジョン

受信フレームは、以下のエラーが1つもなかった場合に「良好」とみなされます。

- + CRC エラー
- + ラントフレーム (64 バイト未満)
- + アライメントエラー (10/100 Mbit/s のみ)
- + 長さエラー (非型フレームのみ)
- + 範囲外 (非型フレームのみ、最大サイズより長い)
- + MII\_RXER 入力エラー

最大フレームサイズは、次のように、フレームのタイプに依存します。

- + タグなしフレームの最大サイズ = 1518
- + VLAN フレームの最大サイズ = 1522

## 38.5.8 電源管理 : PMT

このセクションでは、MAC によってサポートされる電源管理 (PMT) のメカニズムについて説明します。PMT は、ネットワーク (リモート) ウェイクアップフレームとマジックパケットフレームの受信をサポートします。PMT は、MAC によって受信されたウェイクアップフレームおよびマジックパケットに対して割り込みを生成します。PMT ブロックは、リモートウェイクアップフレームイネーブルとマジックパケットイネーブルで有効にされます。これらのイネーブルビット (WFE および MPE) は、ETH\_MACPMTCSR レジスタにあり、アプリケーションによってプログラムされます。PMT でパワーダウンモードが有効になると、すべての受信フレームが MAC によってドロップされ、アプリケーションに転送されません。MAC は、マジックパケットまたはリモートウェイクアップフレームが受信され、対応する検出機能が有効なときのみ、パワーダウンモードから抜けます。

### リモートウェイクアップフレームフィルタレジスタ

8 つのウェイクアップフレームフィルタレジスタがあります。各レジスタに書き込むには、ウェイクアップフレームフィルタレジスタに値を1つずつロードします。ウェイクアップフレームフィルタレジスタに8回、順にロードすることによって、ウェイクアップフレームフィルタの望ましい値がロードされます。読み出し操作は、書き込み動作と同じです。8 つの値を読み出すには、ウェイクアップフレームフィルタレジスタを8回、最後のレジスタまで読み出す必要があります。各読み出し/書き込みは、ウェイクアップフレームフィルタレジスタを次のフィルタレジスタに向けさせます。

図 490. ウェイクアップフレームフィルタレジスタ

ウェイクアップフレームフィルタレジスタ 0	フィルタ 0 バイトマスク							
ウェイクアップフレームフィルタレジスタ 1	フィルタ 1 バイトマスク							
ウェイクアップフレームフィルタレジスタ 2	フィルタ 2 バイトマスク							
ウェイクアップフレームフィルタレジスタ 3	フィルタ 3 バイトマスク							
ウェイクアップフレームフィルタレジスタ 4	RSVD	フィルタ 3 コマンド	RSVD	フィルタ 2 コマンド	RSVD	フィルタ 1 コマンド	RSVD	フィルタ 0 コマンド
ウェイクアップフレームフィルタレジスタ 5	フィルタ 3 オフセット		フィルタ 2 オフセット		フィルタ 1 オフセット		フィルタ 0 オフセット	
ウェイクアップフレームフィルタレジスタ 6	フィルタ 1 CRC - 16				フィルタ 0 CRC - 16			
ウェイクアップフレームフィルタレジスタ 7	フィルタ 3 CRC - 16				フィルタ 2 CRC - 16			

ai15647

- **フィルタ i バイトマスク**  
このレジスタは、フレームがウェイクアップフレームであるかどうかを判定するために、フィルタ i (0、1、2、および 3) によって調べられるフレームのバイトを定義します。MSB (31 番目のビット) はゼロでなければなりません。ビット j [30:0] はバイトマスクです。バイトマスクのビット j (バイト数) がセットされた場合、受信フレームのフィルタ i オフセット + j が CRC ブロックによって処理されます。そうでない場合、フィルタ i オフセット + j は無視されます。
- **フィルタ i コマンド**  
この 4 ビットのコマンドは、フィルタ i 動作を制御します。ビット 3 は、アドレスタイプを指定し、パターンのデスティネーションアドレスタイプを定義します。このビットがセットされると、パターンはマルチキャストフレームにのみ適用されます。このビットがリセットされると、パターンはユニキャストフレームにのみ適用されます。ビット 2 とビット 1 は予約済みです。ビット 0 はフィルタ i のイネーブルビットです。ビット 0 がセットされていない場合、フィルタは無効です。
- **フィルタ i オフセット**  
このレジスタは、フレームがフィルタ i によって調べられる (フレーム内の) オフセットを定義します。この 8 ビットパターンのオフセットは、フィルタ i によって調べられる最初のバイトのオフセットです。許される最小値は 12 であり、フレームの 13 番目のバイトを指します (オフセット値 0 は、フレームの最初のバイトを指します)。
- **フィルタ i CRC-16**  
このレジスタは、パターンから計算された CRC-16 値に加えて、ウェイクアップフィルタレジスタブロックに対してプログラムされたバイトマスクを含みます。

## リモートウェイクアップフレームの検出

MAC がスリープモードであり、ETH\_MACPMTCSR レジスタのリモートウェイクアップビットが有効なときには、リモートウェイクアップフレームの受信後、通常の動作が再開されます。アプリケーションは、ウェイクアップフレームフィルタレジスタアドレスへの逐次書き込みを行うことによって、8 つのウェイクアップフィルタレジスタのすべてに書き込みます。アプリケーションは、ETH\_MACPMTCSR レジスタのビット 2 に 1 を書き込むことによって、リモートウェイクアップを有効にします。PMT は、さまざまな受信フレームパターンを提供する 4 つのプログラマブルフィルタをサポートします。受信フレームがフィルタコマンドのアドレスフィルタリングをパスし、フィルタ CRC-16 が、チェック対象の受信パターンに一致した場合、ウェイクアップフレームは受信されず。Filter\_offset (最小値は、フレームの 13 番目のバイトを指す 12) は、フレームをチェックするオフセット値を指定します。フィルタバイトマスクは、チェックするフレームのバイトを指定します。バイトマスクの 31 番目のビットは、ゼロにセットされなければなりません。ウェイクアップフレー

ムは、ラントフレームではないことを確認するために、長さエラー、FCS エラー、ドリブルビットエラー、MII エラー、コリジョンのみがチェックされます。ウェイクアップフレームの長さが 512 バイトを超える場合でも、フレームの CRC 値が有効な場合は、有効とみなされます。ウェイクアップフレームの検出は、リモートウェイクアップフレームが受信されるたびに、ETH\_MACPMTCSR レジスタで更新されます。有効な場合は、PMT 割り込みが生成され、リモートウェイクアップフレームの受信を示します。

## マジックパケットの検出

マジックパケットフレームは、Advanced Micro Device 社の Magic Packet テクノロジーを使用する方法に基づき、ネットワーク上のスリープ状態のデバイスを起動します。MAC は、ネットワーク上の特定のノードに宛てたマジックパケットと呼ばれる特定のパケット情報を受信します。デバイスまたはブロードキャストアドレスに宛てられたマジックパケットのみが、ウェイクアップ要件を満たしているかどうかチェックされます。アドレスフィルタリング (ユニキャストまたはブロードキャスト) をパスしたマジックパケットは、すべて 1 の 6 バイトの後に MAC アドレスが 16 回現われるというリモート Wake-on-LAN データフォーマットを満たしているかどうかチェックされます。アプリケーションは、ETH\_MACPMTCSR レジスタのビット 1 に 1 を書き込むことによって、マジックパケットウェイクアップを有効にします。PMT ブロックは、ノードに宛てられた各フレームが特定のマジックパケットパターンを持っているかどうかを常時監視します。受信された各フレームは、デスティネーションおよびソースアドレスフィールドの後に 0xFFFF FFFF FFFF というパターンが続くかどうかチェックされます。次に、PMT ブロックは、フレーム内の MAC アドレスが、中断や途切れがなく、16 回繰り返されているかどうかをチェックします。アドレスの 16 回の繰り返しの途中で中断がある場合、受信フレームに 0xFFFF FFFF FFFF パターンが含まれているかどうか再度チェックされます。16 回の繰り返しはフレーム内のどこにあっても構いませんが、同期ストリーム (0xFFFF FFFF FFFF) の後でなければなりません。デバイスは、MAC アドレスの 16 回の繰り返しが検出される限り、マルチキャストフレームも受け入れます。ノードの MAC アドレスが 0x0011 22334455 である場合は、MAC は、データシーケンスをスキャンします。

```
デスティネーションアドレスとソースアドレス .....FFFF FFFF FFFF
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
0011 2233 4455 0011 2233 4455 0011 2233 4455 0011 2233 4455
...CRC
```

マジックパケットの検出は、受信されたマジックパケットの ETH\_MACPMTCSR レジスタで更新されます。有効な場合、PMT 割り込みが生成され、マジックパケットの受信を示します。

## パワーダウン時のシステムに関する考慮事項

イーサネット PMT ブロックは、システムがストップモードのときでもフレームを検出できます。ただし、EXIT ライン 19 が有効な場合に限りです。

MAC レシーバステートマシンは、パワーダウンモード中も有効なままでなければなりません。これは、マジックパケット/Wake-on-LAN フレームの検出に関係するので、ETH\_MACCCR レジスタの RE ビットがセットされたままでなければならぬことを意味します。ただし、送信ステートマシンは、パワーダウンモード時には、ETH\_MACCCR レジスタの TE ビットをクリアすることによってオフされなければなりません。さらに、マジックパケット/Wake-on-LAN フレームを SRAM にコピーする必要はないので、イーサネット DMA は、パワーダウンモード中は無効でなければなりません。イーサネット DMA を無効にするには、ETH\_DMAOMR レジスタの ST ビットと SR ビット (それぞれ、送信 DMA と受信 DMA に対応) をクリアします。

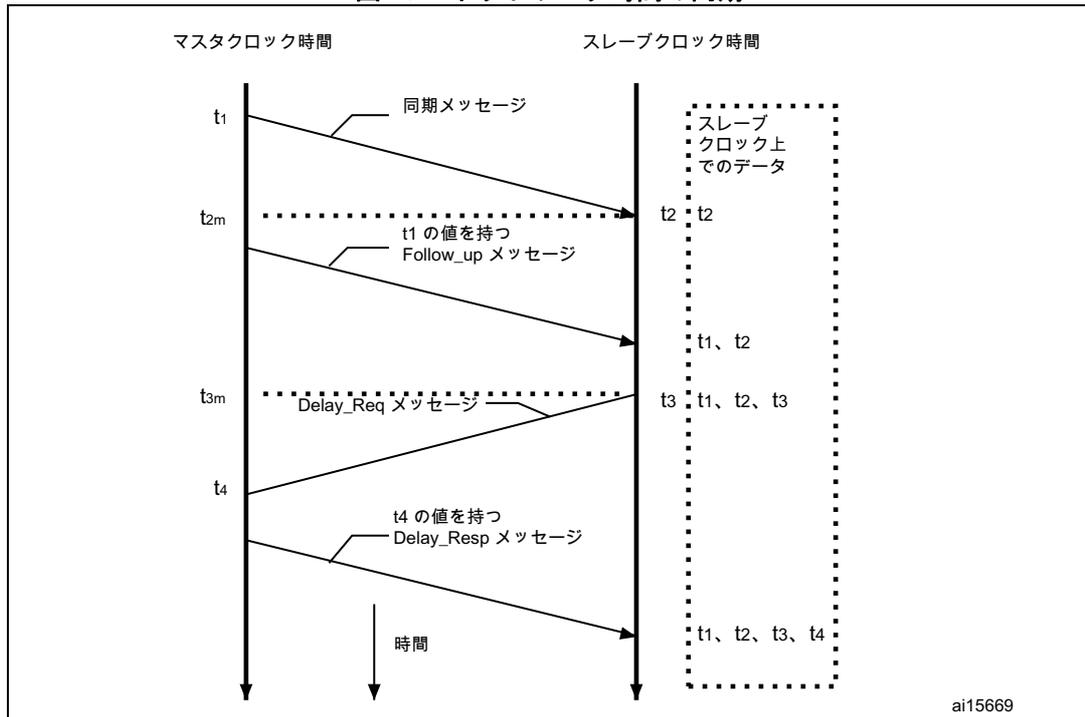
推奨されるパワーダウンおよびウェイクアップシーケンスは、次のとおりです。

1. 送信 DMA を無効にして、前のフレーム送信の完了を待ちます。送信完了は、ETH\_DMASR レジスタ [0] の送信割り込みが受信されると検出できます。
2. ETH\_MACCCR 設定レジスタの RE および TE ビットをクリアすることによって、MAC トランスミッタと MAC レシーバを無効にします。
3. 受信 DMA が Rx FIFO 内のすべてのフレームを空にするのを待ちます。
4. 受信 DMA を無効にします。
5. EXIT ライン 19 を、イベントまたは割り込みを生成するように設定して、有効にします。
6. 割り込みを生成するように EXIT ライン 19 を設定した場合は、ETH\_WKUP\_IRQ ハンドラ機能も正しく設定して、EXIT ライン 19 のペンディングビットをクリアする必要があります。
7. ETH\_MACPMTCSR レジスタの MFE/WFE ビットをセットすることによって、マジックパケット/Wake-on-LAN フレーム検出を有効にします。
8. ETH\_MACPMTCSR レジスタの PD ビットをセットすることによって、MAC パワーダウンモードを有効にします。
9. ETH\_MACCCR レジスタの RE ビットをセットすることによって、MAC レシーバを有効にします。
10. システムをストップモードにします (詳細については、[セクション 4.3.5: STOP モード](#)を参照)。
11. 有効なウェイクアップフレームを受信すると、イーサネットペリフェラルはパワーダウンモードから抜けます。
12. ETH\_MACPMTCSR レジスタを読み出して、電力管理イベントフラグをクリアし、MAC トランスミッタステートマシンと受信および送信 DMA を有効にします。
13. システムクロックを設定します。すなわち、HSE を有効にして、クロックをセットします。

## 38.5.9 高精度時間プロトコル (IEEE1588 PTP)

IEEE 1588 標準は、ネットワーク通信、ローカルコンピューティング、分散オブジェクトなどのテクノロジーで実装される計測および制御システムでの高精度なクロック同期を可能にするプロトコルを定義しています。このプロトコルは、イーサネットをはじめとして (ただし、これに限られず)、マルチキャストメッセージングをサポートするローカルエリアネットワークによって通信するシステムに適用されます。このプロトコルは、固有精度、解像度、および安定性が異なるクロックを含む異種混合システムの同期に使用されます。このプロトコルは、最小限のネットワークおよびローカルクロック計算リソースを使用して、サブマイクロ秒範囲のシステム全体の正確な同期をサポートします。高精度時間プロトコル (PTP) と呼ばれるメッセージベースのプロトコルは、UDP/IP 経由で転送されます。システムまたはネットワークは、タイミング/クロック情報を分配するために、マスターノードとスレーブノードに分類されます。PTP メッセージを交換することによってスレーブノードをマスターノードに同期するこのプロトコルの技法を [図 491](#) に示します。

図 491. ネットワーク時間の同期



1. マスタは、PTP 同期メッセージをすべてのノードにブロードキャストします。同期メッセージは、マスタの基準時間情報を含んでいます。このメッセージがマスタのシステムから送信された時間が、 $t_1$  です。イーサネットポートの場合、この時間は、MII でキャプチャされなければなりません。
2. スレーブは、同期メッセージを受信し、その基準タイミングを使用して、正確な時間  $t_2$  をキャプチャします。
3. マスタは、次に、後で使用する  $t_1$  情報を含んだ Follow\_up メッセージをスレーブに送信します。
4. スレーブは、マスタに Delay\_Req メッセージを送信して、このフレームが MII を離れた正確な時間  $t_3$  を知らせます。
5. マスタは、このメッセージを受信して、メッセージがシステムに着信した正確な時間  $t_4$  をキャプチャします。
6. マスタは、Delay\_Resp メッセージで  $t_4$  情報をスレーブに送信します。
7. スレーブは、 $t_1$ 、 $t_2$ 、 $t_3$ 、および  $t_4$  の 4 つの値を使用して、ローカル基準時間をマスタの基準タイミングに同期します。

プロトコル実装のほとんどは、UDP レイヤの上のソフトウェアで行われます。ただし、すでに述べたように、特定の PTP パケットが MII のイーサネットポートで送受信された正確な時間をキャプチャするには、ハードウェアサポートが必要です。PTP の正しく高精度な実装のためには、このタイミング情報がキャプチャされて、ソフトウェアに返されなければなりません。

## 基準タイミングソース

時間のスナップショットを得るために、コアは、IEEE 1588 仕様での定義に従って、64 ビットフォーマット (2 つの 32 ビットチャネルに分け、上位 32 ビットは時間を秒単位で、下位 32 ビットは時間をマイクロ秒で示す) の基準時間を必要とします。

PTP 基準クロック入力、基準時間 (システム時間とも呼ばれます) を内部で生成し、タイムスタンプをキャプチャするために使用されます。この基準クロックの周波数は、タイムスタンプカウンタの解像度以上でなければなりません。マスタノードとスレーブ間の同期精度の目標は、約 100 ns です。

システム時間の生成、更新、および変更については、[セクション : システム時間の補正方法](#)で説明されています。

精度は、PTP 基準クロック入力周期、オシレータの特性 (ドリフト)、および同期プロシーダの頻度などに依存します。

Tx および Rx クロック入力ドメインから PTP 基準クロックドメインへの同期により、ラッチされたタイムスタンプの値の不確かさは、1 基準クロック周期分となります。解像度による不確かさを追加した場合、タイムスタンプングに対して半周期を追加することになります。

## PTP 機能を使用したフレームの送信

フレームの SFD が MII に出力されると、タイムスタンプがキャプチャされます。タイムスタンプのキャプチャが必要なフレームは、フレームごとに制御可能です。言い換えると、そのフレームのタイムスタンプをキャプチャする必要があるかどうかを、送信フレームごとにマークできます。送信フレームは、PTP フレームを識別するために処理されません。フレーム制御は、送信ディスクリプタの制御ビットを通じて行われます。キャプチャされたタイムスタンプは、フレームのステータスが返されると同じ方法で、アプリケーションに返されます。タイムスタンプは、対応する送信ディスクリプタ内のフレームの送信ステータスとともに送り返されるので、タイムスタンプは特定の PTP フレームに自動的に接続されます。64 ビットのタイムスタンプ情報は、TDES2 および TDES3 フィールドに書き戻されます。TDES2 は、タイムスタンプの下位 32 ビットを保持します。

## PTP 機能を使用したフレームの受信

IEEE 1588 タイムスタンプ機能が有効なときには、イーサネット MAC は、MII で受信したすべてのフレームのタイムスタンプをキャプチャします。MAC は、フレームの受信が完了すると、タイムスタンプを与えます。キャプチャされたタイムスタンプは、フレームのステータスが返されると同じ方法で、アプリケーションに返されます。タイムスタンプは、対応する受信ディスクリプタ内のフレームの受信ステータスとともに送り返されます。64 ビットのタイムスタンプ情報は、RDES2 および RDES3 フィールドに書き戻されます。RDES2 は、タイムスタンプの下位 32 ビットを保持します。

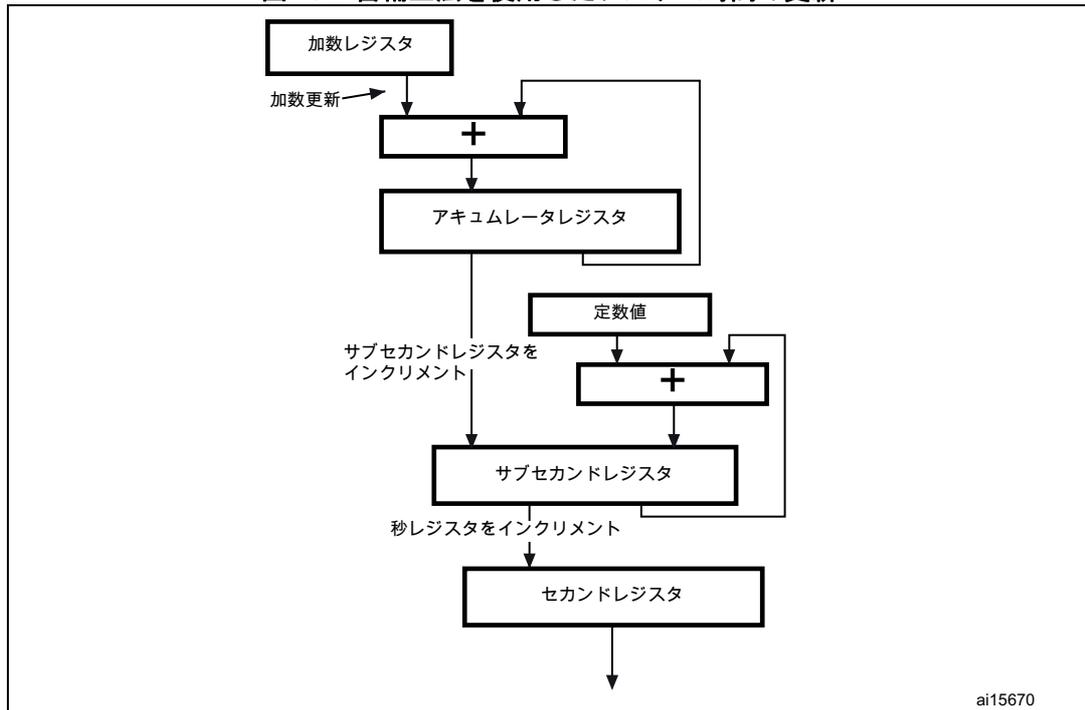
## システム時間の補正方法

64 ビットの PTP 時間は、PTP 入力基準クロック HCLK を使用して更新されます。この PTP 時間は、MII で送受信されるイーサネットフレームのスナッチショット (タイムスタンプ) をとるためのソースとして使用されます。システム時間カウンタは、粗補正法を使用して初期化または補正できます。

粗補正法では、初期値またはオフセット値がタイムスタンプ更新レジスタ ([セクション 38.8.3 : IEEE 1588 タイムスタンプレジスタ \(1566 ページ\)](#)) に書き込まれます。初期化の場合は、システム時間カウンタにタイムスタンプ更新レジスタの値が書き込まれますが、システム時間の補正の場合は、オフセット値 (タイムスタンプ更新レジスタ) がシステム時間に加算または減算されます。

密補正法では、マスタクロックに対するスレーブクロック (基準クロック) の周波数ドリフト (IEEE1588 で定義) が、一定時間にわたって補正されますが、粗補正法では、補正は 1 クロックサイクルで行われます。補正時間が長いほど、リニア時間を維持しやすく、PTP 同期メッセージインターバル間の基準時間に大きな変化 (または大きなジッタ) が生じません。この方法では、[図 492](#) に示されるように、加算器が加数レジスタの内容を累積します。加算器が生成する桁上げは、システム時間カウンタをインクリメントするためのパルスとして使用されます。加算器と加数レジスタは、32 ビットのレジスタです。したがって、加算器は、高精度な周波数逓倍器または分周器として動作します。[図 492](#) に、このアルゴリズムを示します。

図 492. 密補正法を使用したシステム時間の更新



システム時間更新ロジックは、20 ns の精度を達成するために 50 MHz のクロック周波数を必要とします。周波数分周比は、必要なクロック周波数に対する基準周波数の比です。したがって、基準クロック (HCLK) が、たとえば 66 MHz の場合、分周比は  $66 \text{ MHz} / 50 \text{ MHz} = 1.32$  になります。したがって、レジスタで設定するデフォルトの加数値は、 $2^{32} / 1.32$ 、すなわち `0xC1F0 7C1F` です。

基準クロックが低周波数側にドリフトして、たとえば 65 MHz になった場合、分周比は  $65 / 50$ 、すなわち 1.3 となり、加数レジスタで設定する値は  $2^{32} / 1.30$ 、すなわち `0xC4EC 4EC4` です。基準クロックが高周波数側にドリフトして、たとえば 67 MHz になった場合は、加数レジスタを `0xBF0 B7672` に設定する必要があります。クロックのドリフトがゼロのときには、デフォルトの加数値として `0xC1F0 7C1F (2^{32} / 1.32)` をプログラムする必要があります。

図 492 では、サブセカンドレジスタをインクリメントするために使用される定数値は、`0d43` です。これにより、システム時間は 20 ns の精度になります (言い換えると、20 ns 単位でインクリメントされます)。

ソフトウェアは、同期メッセージに基づいて周波数のドリフトを計算し、それに応じて加数レジスタを更新する必要があります。最初に、スレーブクロックは加数レジスタで `FreqCompensationValue0` で設定されます。この値は、次のとおりです。

$$\text{FreqCompensationValue0} = 2^{32} / \text{FreqDivisionRatio}$$

`MasterToSlaveDelay` が連続した同期メッセージで同じであると仮定した場合、以下のアルゴリズムが適用されなければなりません。数サイクルの同期サイクル後、周波数ロックが発生します。スレーブクロックは、正確な `MasterToSlaveDelay` 値を決定して、新しい値を使用してマスタと再同期します。

アルゴリズムは、次のとおりです。

- MasterSyncTime (n) の時点で、マスタはスレーブクロックに同期メッセージを送信します。スレーブは、ローカルクロックが SlaveClockTime (n) のときに、このメッセージを受信して、MasterClockTime (n) を次のように計算します。  
$$\text{MasterClockTime (n)} = \text{MasterSyncTime (n)} + \text{MasterToSlaveDelay (n)}$$
- 現在の同期サイクルのマスタクロックカウント、MasterClockCount (n) は、次の式で求められます。  
$$\text{MasterClockCount (n)} = \text{MasterClockTime (n)} - \text{MasterClockTime (n - 1)}$$
 (MasterToSlaveDelay が同期サイクル n と n-1 で同じであると仮定)
- 現在の同期サイクルのスレーブクロックカウント、SlaveClockCount (n) は、次の式で求められます。  
$$\text{SlaveClockCount (n)} = \text{SlaveClockTime (n)} - \text{SlaveClockTime (n - 1)}$$
- 現在の同期サイクルのマスタとクロックのクロックカウントの差、ClockDiffCount (n) は、次の式で求められます。  
$$\text{ClockDiffCount (n)} = \text{MasterClockCount (n)} - \text{SlaveClockCount (n)}$$
- スレーブクロックの周波数スケールファクタ、FreqScaleFactor (n) は、次の式で求められます。  
$$\text{FreqScaleFactor (n)} = (\text{MasterClockCount (n)} + \text{ClockDiffCount (n)}) / \text{SlaveClockCount (n)}$$
- 加数レジスタの周波数補正值、FreqCompensationValue (n) は、次の式で求められます。  
$$\text{FreqCompensationValue (n)} = \text{FreqScaleFactor (n)} \times \text{FreqCompensationValue (n - 1)}$$

理論的には、このアルゴリズムは 1 同期サイクルでロックを達成しますが、ネットワークの伝搬遅延と動作条件の変動のために、数サイクルかかる場合もあります。

このアルゴリズムは自己補正型です。すなわち、何らかの理由で、最初にスレーブクロックが正しくないマスタからの値に設定された場合でも、アルゴリズムは同期サイクルを数サイクル費やして、それを補正します。

## システム時間生成初期化のプログラミング手順

タイムスタンプ機能は、タイムスタンプ制御レジスタ (ETH\_PTPTSCR) のビット 0 をセットすることによって有効にできます。ただし、タイムスタンプ動作を開始するには、このビットをセットした後、タイムスタンプカウンタを初期化する必要があります。正しいシーケンスは、次のとおりです。

1. MACIMR レジスタのビット 9 をセットすることによって、タイムスタンプトリガ割り込みをマスクします。
2. タイムスタンプレジスタのビット 0 をプログラムして、タイムスタンプを有効にします。
3. PTP クロック周波数に基づいて、サブセカンドインクリメントレジスタをプログラムします。
4. 密補正法を使用する場合は、タイムスタンプ加数レジスタをプログラムし、タイムスタンプ制御レジスタのビット 5 (加数レジスタの更新) をセットします。
5. ビット 5 がクリアされるまで、タイムスタンプ制御レジスタにポーリングします。
6. 密補正法を選択するには (必要な場合)、タイムスタンプ制御レジスタのビット 1 をプログラムします。
7. タイムスタンプハイ更新およびタイムスタンプロー更新レジスタを適切な時間値でプログラムします。
8. タイムスタンプ制御レジスタのビット 2 (タイムスタンプ初期化) をセットします。
9. タイムスタンプカウンタは、タイムスタンプ更新レジスタに書き込まれた値で初期化されると、動作を開始します。
10. 適切なタイムスタンプングのためには、MAC レシーバおよびトランスミッタを有効にします。

注 : ETH\_PTPSCR レジスタのビット 0 をクリアすることによってタイムスタンプ動作が無効にされた場合、タイムスタンプ動作をリスタートするには、上記の手順を繰り返す必要があります。

## 粗補正法でのシステム時間更新のプログラミング手順

システム時間をワンプロセスで同期または更新するには (粗補正法)、以下の手順を実行します。

1. タイムスタンプ更新ハイおよびローレジスタにオフセットを書き込みます (正または負)。
2. タイムスタンプ制御レジスタのビット 3 (TSSTU) をセットします。
3. TSSTU ビットがクリアされると、タイムスタンプ更新レジスタの値がシステム時間に加えられるか、差し引かれます。

## 密補正法でのシステム時間更新のプログラミング手順

システム時間を同期または更新して、システム時間のジッタを減らすには (密補正法)、以下の手順を実行します。

1. **セクション : システム時間の補正方法**で説明されているアルゴリズムを使用して、システム時間のインクリメントをスピードアップまたはスローダウンさせる割合を計算します。
2. タイムスタンプを更新します。
3. 加数レジスタの新しい値として有効にしたい時間まで待ちます。このためには、システム時間が目標値に達した後、タイムスタンプトリガ割り込みを有効にします。
4. 目標時間ハイおよびローレジスタで、必要な目標時間をプログラムします。ETH\_MACIMR レジスタのビット 9 をクリアすることによって、タイムスタンプ割り込みマスクを解除します。
5. タイムスタンプ制御レジスタのビット 4 (TSARU) をセットします。
6. このトリガによって割り込みが発生したときには、ETH\_MACSR レジスタを読み出します。
7. タイムスタンプ加数レジスタを元の値で再プログラムし、ETH\_TPTSCR レジスタのビット 5 を再びセットします。

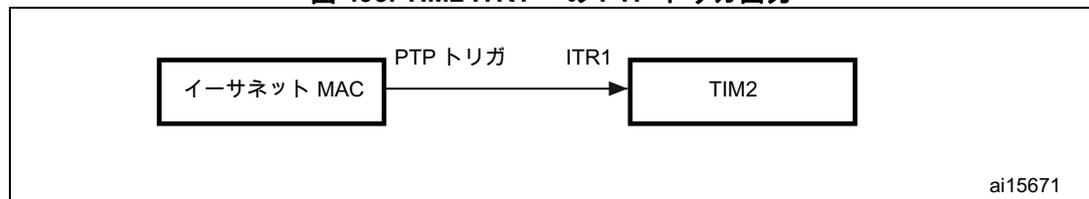
## TIM2 と PTP トリガの内部接続

MAC は、システム時間が目標時間を超えたときにトリガ割り込みを行います。割り込みを使用すると、既知の遅延に加えて、コマンド実行時間の不確実性が導入されます。

この不確実性を避けるために、システム時間が目標時間より大きいときには、PTP トリガ出力信号はハイにセットされます。割り込みは、内部的に TIM2 入力トリガに接続されます。この信号により、入力キャプチャ機能、出力比較機能、およびタイマ波形が、同期された PTP システム時間のトリガによって使用可能になります。タイマ (PCLK1 : TIM2 APB1 クロック) のクロックと PTP 基準クロック (HCLK) は同期しているため、不確実性は生じません。

この PTP トリガ信号は、ソフトウェアによって選択可能な TIM2 ITR1 入力に接続されます。接続は、TIM2 オプションレジスタ (TIM2\_OR) のビット 11 とビット 10 を使って有効化されます。図 493 に、この接続を示します。

図 493. TIM2 ITR1 への PTP トリガ出力



## PTP の PPS (Pulse-per-second) 出力信号

この PTP パルス出力は、ネットワーク内のすべてのノード間の同期をチェックするために使用されます。ローカルスレーブクロックとマスタ基準クロック間の違いをテストできるように、両方のクロックに、必要に応じてオシレータに接続可能な PPS (Pulse-per-second) 出力信号が与えられています。したがって、2 つの信号間の偏差を測定することができます。PPS 出力のパルス幅は 125 ms です。

PPS 出力は、TIM2 オプションレジスタ (TIM2\_OR) のビット 11 とビット 10 を使って有効化されます。

PPS 出力のデフォルト周波数は 1 Hz です。PPSFREQ[3:0] (ETH\_PTPTSCR) を使って PPS 出力の周波数を  $2^{\text{PPSFREQ}}$  Hz に設定することができます。

1 Hz に設定すると、PPS パルス幅は、バイナリロールオーバー (TSSSR=0、ETH\_PTPTSCR のビット 9) で 125 ms に、デジタルロールオーバー (TSSSR=1) で 100 ms に、それぞれなります。2 Hz 以上に設定すると、PPS 出力のデューティサイクルはバイナリロールオーバーで 50% です。

デジタルロールオーバー (TSSSR=1) では、1 Hz 以外の周波数を持つ PPS 出力を使用しないことが推奨されます。これは、異常な波形を持つためです (平均周波数は任意の 1 秒ウィンドウ内で常に正しいのですが)。

図 494. PPS 出力



## 38.6 イーサネット機能の説明 : DMA コントローラ動作

DMA には、独立した送信および受信エンジンと CSR スペースがあります。送信エンジンは、データをシステムメモリから Tx FIFO に転送し、受信エンジンは、データを Rx FIFO からシステムメモリに転送します。コントローラは、ディスクリプタを利用して、データをソースからデスティネーションに最小限の CPU 介入によって効率的に移動します。DMA は、イーサネットのフレームなど、パケット指向のデータ転送を行うように設計されています。コントローラは、フレーム送受信完了やその他の正常/エラー条件発生の場合に CPU に割り込むようにプログラムすることができます。DMA と STM32F75xxx および STM32F74xxx は、次の 2 つのデータ構造を使用して通信します。

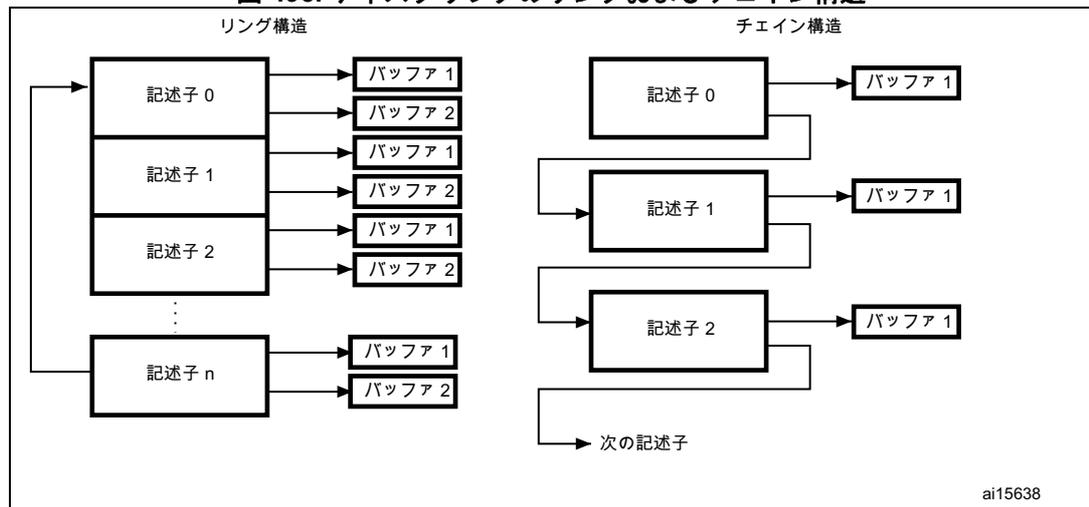
- 制御およびステータスレジスタ (CSR)
- ディスクリプタのリストおよびデータバッファ

制御およびステータスレジスタについては、[セクション 38.8 \(1541 ページ\)](#) で詳しく説明されています。ディスクリプタについては、[セクション \(1521 ページ\)](#) で詳しく説明されています。

DMA は、受信したデータフレームを STM32F75xxx および STM32F74xxx メモリ内の受信バッファに転送し、データフレームを STM32F75xxx および STM32F74xxx メモリ内の送信バッファから転送します。STM32F75xxx および STM32F74xxx メモリ内に存在するディスクリプタは、これらのバッファのポインタとして機能します。ディスクリプタリストには、受信用と送信用の 2 つのリストがあります。各リストのベースアドレスは、それぞれ DMA レジスタ 3 および 4 に書き込まれます。ディスクリプタリストは、転送リンクです (暗黙的または明示的)。最後のディスクリプタから最初のディスクリプタにポイントバックして、リング構造を作ることができます。ディスクリプタの明示的な連鎖は、受信および送信ディスクリプタ (RDES1[14] および TDES0[20]) の両方で連鎖した第 2 のアドレスを設定することによって達成されます。ディスクリプタリストは、ホストの物理メモリ空間に置かれます。各ディスクリプタは、最大 2 つのバッファを指し示すことができます。これにより、メモリ内の隣接した 2 つのバッファではなく、物理的にアドレス指定された 2 つのバッファを使用できま

す。データバッファは、ホストの物理メモリ空間に置かれ、フレーム全体または一部で構成されますが、1 フレームの長さを超えることはできません。バッファは、データのみを含みます。バッファのステータスは、ディスクリプタ内に保持されます。データ連結とは、複数のデータバッファにまたがるフレームを指します。ただし、1つのディスクリプタが複数のフレームにまたがることはできません。DMA は、フレームの終わりが検出されると、次のフレームバッファにスキップします。データチェイニングは、有効または無効にすることができます。ディスクリプタのリングおよびチェーン構造を [図 495](#) に示します。

図 495. ディスクリプタのリングおよびチェーン構造



## 38.6.1 DMA を使用した転送の初期化

MAC の初期化は、次のとおりです。

1. ETH\_DMABMR レジスタに書き込んで、STM32F75xxx および STM32F74xxx のバスアクセスパラメータを設定します。
2. ETH\_DMAIER レジスタに書き込んで、不要な割り込みの原因をマスクします。
3. ソフトウェアドライバは、送信および受信ディスクリプタリストを作成します。次に、ETH\_DMARDLAR および ETH\_DMATDLAR レジスタの両方に書き込んで、各リストの開始アドレスを指定します。
4. MAC レジスタ 1、2、および 3 に書き込んで、望ましいフィルタリングオプションを選択します。
5. MAC ETH\_MACCCR レジスタに書き込んで、送信および受信動作モードを設定し、有効にします。PS および DM ビットは、自動ネゴシエーションの結果 (PHY からの読み出し) に基づいてセットされます。
6. ETH\_DMAOMR レジスタに書き込んで、ビット 13 および 1 をセットし、送信および受信を開始します。
7. 送信および受信エンジンは動作状態に入り、それぞれのディスクリプタリストからディスクリプタの取得を試みます。続いて、受信および送信エンジンは、受信および送信動作の処理を開始します。送信および受信のプロセスは互いに独立していて、個別に開始または停止できます。

## 38.6.2 ホストバスバーストアクセス

DMA は、そのように設定された場合 (ETH\_DMABMR レジスタの FB ビット)、AHB マスタインタフェース上で固定長のバースト転送の実行を試みます。最大バースト長は、PBL フィールド (ETH\_DMABMR [13:8]) によって示され、制限されます。受信および送信ディスクリプタは、16 バイトが読み出されるように、常に可能な最大サイズでアクセスされます (PBL によって制限されず)。

送信 DMA は、送信 FIFO に、フレームの終わりまで、設定されたバーストまたはバイト数を収容できる十分なスペースがあるとき (設定されたバースト長以下であるとき) のみ、データ転送を開始します。DMA は、開始アドレスと要求された転送数を AHB マスタインタフェースに対して示します。AHB インタフェースが固定長バーストに設定されているときには、INCR4、INCR8、INCR16、および SINGLE トランザクションの最良の組み合わせを使用してデータを転送します。固定長バーストでない場合は、INCR (未定義の長さ) および SINGLE トランザクションを使用してデータを転送します。

受信 DMA は、設定されたバースト長に対して十分なスペースが受信 FIFO にあるとき、またはフレームの終わり (設定されたバースト長未満であるとき) が受信 FIFO 内で検出されたときのみ、データ転送を開始します。DMA は、開始アドレスと要求された転送数を AHB マスタインタフェースに対して示します。AHB インタフェースが固定長バーストに設定されているときには、INCR4、INCR8、INCR16、および SINGLE トランザクションの最良の組み合わせを使用してデータを転送します。AHB インタフェース上で固定バーストが終了する前にフレームの終わりに達した場合、固定長バーストを完了するためにダミー転送が実行されます。そうでない場合 (ETH\_DMABMR の FB ビットがリセットされているとき)、INCR (未定義の長さ) および SINGLE トランザクションを使用してデータを転送します。

AHB インタフェースがアドレスアラインドビート用に設定されているときには、両方の DMA エンジンは、AHB が開始する最初のバースト転送が、設定された PBL のサイズ以下であることを確認します。したがって、後続のビートはすべて、設定された PBL に整列されたアドレスで開始します。AHB インタフェースは INCR16 以上をサポートしていないので、DMA は、最大サイズ 16 までのビートについてのみ (PBL > 16)、アドレスを整列できます。

## 38.6.3 ホストデータバッファ整列

送信および受信データバッファには、開始アドレスの整列について制約はありません。32 ビットメモリを持つシステムでは、バッファの開始アドレスは 4 つのバイトのいずれにも整列できます。ただし、DMA は常に、要求されなかったバイトレーンについては、ダミーデータでバス幅に揃えたアドレスで転送を開始します。これは、通常、イーサネットフレームの開始または終わりの転送時に発生します。

- バッファ読み出しの例 :

送信バッファアドレスが 0x0000 0FF2 であり、15 バイトの転送が必要な場合、DMA は、アドレス 0x0000 0FF0 から 5 つの完全なワードを読み出しますが、送信 FIFO へのデータ転送時に、余分なバイト (最初の 2 バイト) がドロップまたは無視されます。同様に、最後の転送の最後の 3 バイトも無視されます。DMA は、フレームの終わりでない限り、常に、完全な 32 ビットデータを送信 FIFO に転送することを保証します。

- バッファ書き込みの例 :

受信バッファアドレスが 0x0000 0FF2 であり、受信したフレームの 16 バイトを転送する必要がある場合、DMA は、5 つの完全な 32 ビットデータアイテムをアドレス 0x0000 0FF0 から書き込みます。ただし、最初の転送の最初の 2 バイトと 3 番目の転送の最後の 2 バイトはダミーデータとなります。

## 38.6.4 バッファサイズの計算

DMA は、送信および受信ディスクリプタのサイズフィールドを更新しません。DMA は、ディスクリプタのステータスフィールド (xDES0) のみを更新します。ドライバはサイズを計算する必要があります。送信 DMA は、MAC コアに向けて正確なバイト数 (TDES1 のバッファサイズフィールドによって示される) を転送します。ディスクリプタが最初としてマークされている場合 (TDES0 の FS ビットがセットされている場合)、DMA は、バッファからの最初の転送をフレームの開始としてマークします。ディスクリプタが最後としてマークされている場合 (TDES0 の LS ビットがセットされている場合)、DMA は、データバッファからの最後の転送をフレームの終わりとしてマークします。受信 DMA は、バッファが満杯になるか、フレームの終わりを受信するまで、バッファにデータを転送します。ディスクリプタが最後としてマークされていない場合 (RDES0 の LS ビット)、そのディスクリプタに対応するバッファは満杯であり、バッファ内の有効データの量は、ディスクリプタの FS ビットがセットされているとき、バッファサイズフィールドからデータバッファポインタオフセットを差し引いた値で正確に示されます。データバッファポインタがデータバス幅に整列されている場合、オフセットはゼロです。ディスクリプタが最後としてマークされている場合、バッファは満杯ではない可能性があります (RDES1 のバッファサイズによって示される)。この最終バッファ内の有効データの量を計算するには、ドライバは、フレーム長 (RDES0[29:16] の FL ビット) を読み出して、その値からこのフレーム内の先行バッファのサイズの合計を差し引く必要があります。受信 DMA は、常に、次のフレームの開始を新しいディスクリプタで転送します。

**注 :** *受信バッファの開始アドレスがシステムデータバス幅に整列されていない場合でも、システムは、システムのバス幅に整列されたサイズの受信バッファを割り当てる必要があります。たとえば、システムがアドレス 0x1000 から始まる 1024 バイト (1 KB) の受信バッファを割り当てた場合、ソフトウェアは、受信ディスクリプタ内のバッファ開始アドレスを 0x1002 のオフセットを持つようにプログラムできます。受信 DMA は、最初の 2 つの位置 (0x1000 と 0x1001) にダミーデータを持つフレームをこのバッファに書き込みます。実際のフレームは 0x1002 の位置から書き込まれます。したがって、このバッファ内の実際に使用可能なスペースは、バッファサイズが 1024 バイトとしてプログラムされた場合でも、開始アドレスのオフセットにより、1022 バイトです。*

## 38.6.5 DMA アービタ

DMA 内のアービタは、AHB マスタインタフェースの送信チャネルアクセスと受信チャネルアクセスの間の調停を行います。ラウンドロビンと固定優先度の 2 種類の調停が可能です。ラウンドロビン調停が選択された場合 (ETH\_DMABMR レジスタの DA ビットがリセット)、アービタは、送信 DMA と受信 DMA の両方が同時にアクセスを要求したとき、ETH\_DMABMR レジスタの PM ビットによってセットされた割合でデータバスを割り当てます。DA ビットがセットされたときには、受信 DMA は、データアクセスの際、常に送信 DMA より優先されます。

## 38.6.6 DMA へのエラー応答

DMA チャネルによって開始されたデータ転送で、スレーブがエラー応答を返した場合、DMA は、すべての動作を停止して、ステータスレジスタ (ETH\_DMASR レジスタ) のエラービットと致命的バスエラービットを更新します。その DMA コントローラは、ペリフェラルのソフトまたはハードリセットと DMA の再初期化の後でのみ、動作を再開できます。

## 38.6.7 Tx DMA の設定

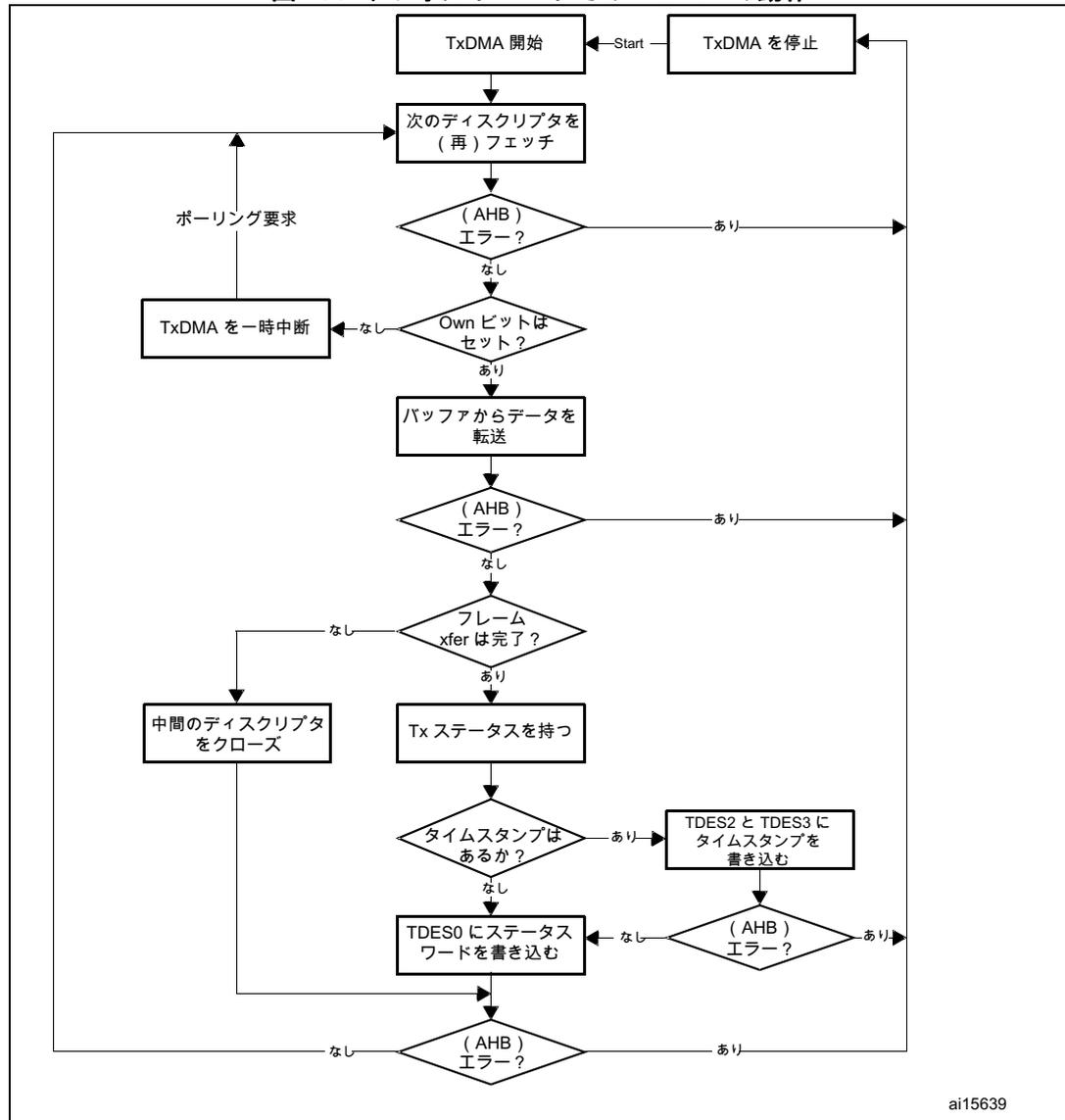
### TxDMA の動作 : デフォルト (非 OSF) モード

デフォルトモードの送信 DMA エンジン、次のように動作します。

1. ユーザは、送信ディスクリプタ (TDES0~TDES3) をセットアップし、対応するデータバッファをイーサネットフレームデータでセットアップした後、OWN ビット (TDES0[3:1]) をセットします。
2. ST ビット (ETH\_DMAOMR レジスタの [13]) がセットされると、DMA は実行状態に入ります。
3. 実行状態の間、DMA は、送信が必要なフレームの送信ディスクリプタリストをポーリングします。ポーリングの開始後は、シーケンシャルディスクリプタリング順またはチェーン順でポーリングを続行します。DMA が CPU に所有されているというフラグが立てられたディスクリプタを検出するか、エラー条件が発生した場合には、送信は中断され、送信バッファ使用不可ビット (ETH\_DMASR レジスタの [2]) と正常割り込み要約ビット (ETH\_DMASR レジスタの [16]) がセットされます。送信エンジンは手順 9 へ進みます。
4. 取得したディスクリプタに DMA の所有であるというフラグが立てられていた場合 (TDES0[31] がセットされている)、DMA は、取得したディスクリプタから送信データバッファアドレスをデコードします。
5. DMA は、STM32F75xxx および STM32F74xxx メモリから送信データをフェッチして、データを転送します。
6. イーサネットフレームが複数のディスクリプタのデータバッファ経由で格納された場合、DMA は、中間のディスクリプタをクローズして、次のディスクリプタをフェッチします。イーサネットフレームデータの終わりが転送されるまで、手順 3、4、および 5 が繰り返されます。
7. フレーム送信が完了すると、IEEE 1588 タイムスタンプがそのフレームに対して有効な場合 (送信ステータスに示される)、タイムスタンプの値がフレーム終わりバッファを含む送信ディスクリプタ (TDES2 および TDES3) に書き込まれます。次に、ステータス情報がこの送信ディスクリプタ (TDES0) に書き込まれます。この手順の間に OWN ビットがクリアされるので、CPU がこのディスクリプタを所有することになります。タイムスタンプがこのフレームに対して有効でない場合、DMA は TDES2 と TDES3 の内容を変更しません。
8. 最後のディスクリプタで完了時割り込み (TDES1[31]) がセットされているフレームの送信を完了した後、送信割り込み (ETH\_DMASR レジスタの [0]) がセットされます。DMA エンジンは手順 3 に戻ります。
9. サスペンド状態では、DMA は送信ポーリング要求を受け取ると、ディスクリプタの再取得を試み (その結果、手順 3 に戻り)、アンダーフロー割り込みステータスビットがクリアされます。

 496 に、デフォルトモードでの TxDMA 送信フローを示します。

図 496. デフォルトモードでの TxDMA の動作



## TxDMA の動作 : OSF モード

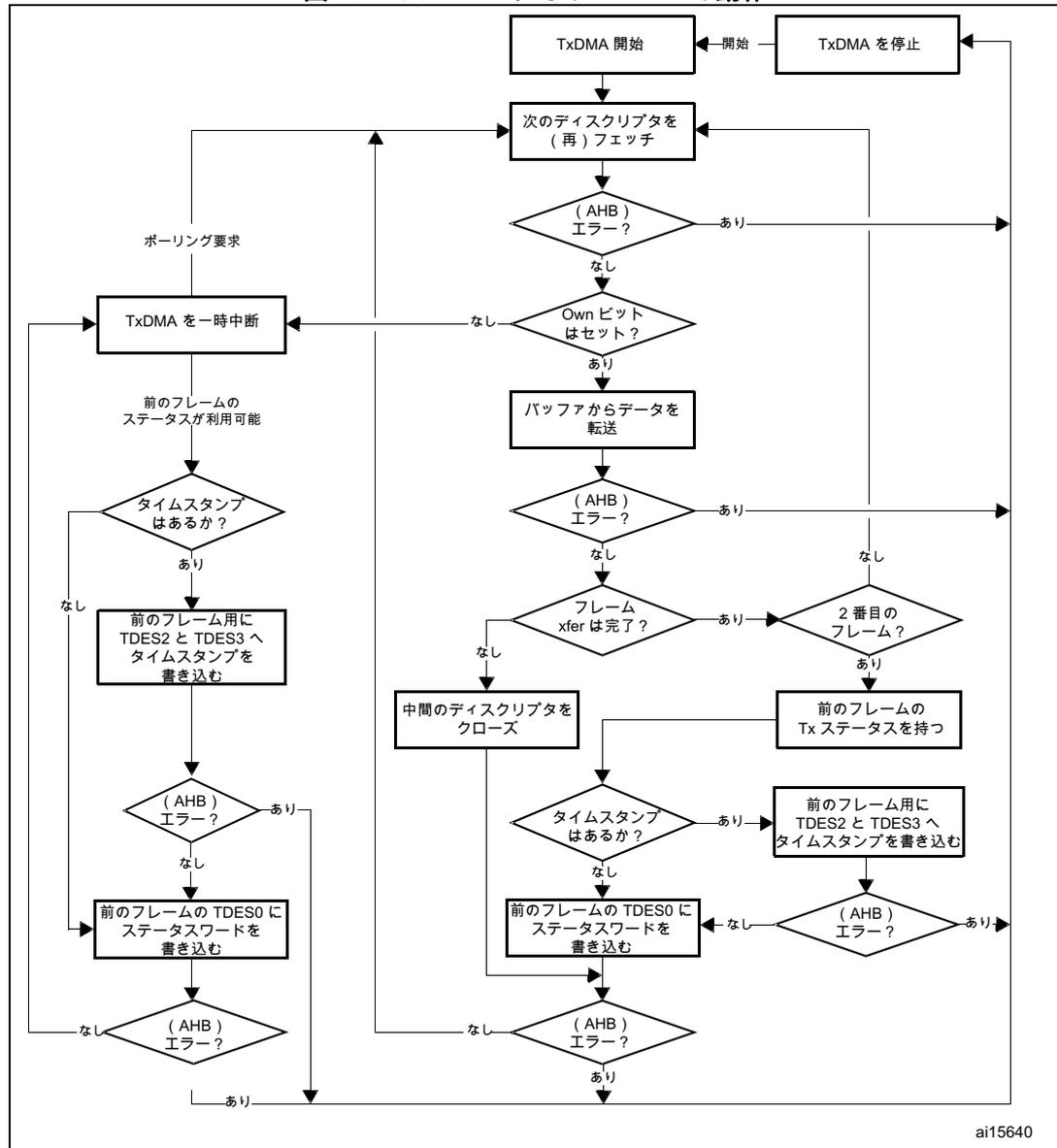
実行状態では、送信プロセスは、最初のフレームのステータスディスクリプタをクローズしなくても、同時に 2 つのフレームを取得することができます (ETH\_DMAOMR レジスタ [2] の OSF ビットがセットされている場合)。送信プロセスが最初のフレームの転送を終了すると、ただちに 2 番目のフレームの送信ディスクリプタリストをポーリングします。2 番目のフレームが有効な場合、送信プロセスは、最初のフレームのステータス情報を書き込む前に、このフレームを転送します。OSF モードでは、実行状態の送信 DMA は、以下のシーケンスに従って動作します。

1. DMA は、TxDMA の手順 1~6 で説明されているように動作します (デフォルトモード)。
2. 前のフレームの最後のディスクリプタをクローズせずに、DMA は、次のディスクリプタをフェッチします。
3. DMA が取得したディスクリプタを所有している場合、DMA は、このディスクリプタ内の送信バッファアドレスをデコードします。DMA がディスクリプタを所有していない場合、DMA は、サスペンドモードに入り、手順 7 をスキップします。

4. DMA は、STM32F75xxx および STM32F74xxx メモリから送信フレームをフェッチして、フレームデータの終わりが転送されるまで、フレームを転送し、このフレームが複数のディスクリプタに分割されている場合は、中間のディスクリプタをクローズします。
5. DMA は、前のフレームの送信ステータスとタイムスタンプを待ちます。ステータスが入手可能なときには、DMA は、タイムスタンプがキャプチャされていた場合は (ステータスビットで示されます)、タイムスタンプを TDES2 と TDES3 に書き込みます。DMA は、次に、OWN ビットがクリアされたステータスを、対応する TDES0 に書き込み、ディスクリプタをクローズします。タイムスタンプが前のフレームに対して有効でなかった場合、DMA は、TDES2 と TDES3 の内容を変更しません。
6. 有効な場合は、送信割り込みがセットされ、DMA は、次のディスクリプタをフェッチしてから、手順 3 へ進みます (ステータスが正常な場合)。前の送信ステータスがアンダーフローエラーを示している場合、DMA はサスペンドモードに移行します (手順 7)。
7. サスペンドモードでは、DMA がペンディングステータスとタイムスタンプを受け取った場合、DMA はタイムスタンプを TDES2 と TDES3 に書き込み (現在のフレームに対して有効な場合)、次に、対応する TDES0 にステータスを書き込みます。次に、該当する割り込みをセットして、サスペンドモードに戻ります。
8. DMA は、送信ポーリング要求 (ETH\_DMATPDR レジスタ) を受け取った後にのみ、サスペンドモードから抜けて、実行状態に入ることができます (ペンディングステータスに応じて、手順 1 または手順 2 へ移行します)。

 497 に、OSF モードでの基本的なフローチャートを示します。

図 497. OSF モードでの TxDMA の動作



## 送信フレームの処理

送信 DMA は、データバッファが、プリアンブル、パッドバイト、および FCS フィールドを除く完全なイーサネットフレームを含んでいることを期待しています。DA、SA、およびタイプ/長さフィールドは、有効なデータを含んでいます。送信ディスクリプタが、MAC コアが CRC またはパッド挿入を無効にする必要があることを示している場合、バッファは、CRC バイトも含めて、完全なイーサネットフレーム（プリアンブルは除外）を持っていなければなりません。フレームは、データチェーンであったり、複数のバッファにまたがっていることがあります。フレームは、最初のディスクリプタ (TDES0[28]) と最後のディスクリプタ (TDES0[29]) によって区切られなければなりません。送信が始まると、最初のディスクリプタの TDES0[28] がセットされなければなりません。これが発生すると、フレームデータがメモリバッファから送信 FIFO に転送されます。同時に、現在のフレームの最後のディスクリプタ (TDES0[29]) がクリアされた場合、送信プロセスは、次のディスクリプタの取得を試みます。送信プロセスは、このディスクリプタ内の TDES0[28] がクリアされることを期待します。TDES0[29] がクリアされた場合、中間バッファであることを示します。TDES0[29] がセッ

トされた場合、フレームの最後のバッファであることを示します。フレームの最後のバッファが送信された後、DMA は、送信ディスクリプタ 0 (TDES0[29]) で最後のセグメントがセットされたディスクリプタの送信ディスクリプタ 0 (TDES0) ワードに最終ステータス情報を書き戻します。この時点で、完了時割り込み (TDES0[30]) がセットされている場合は、送信割り込み (ETH\_DMASR レジスタ [0]) がセットされ、次のディスクリプタがフェッチされ、プロセスが繰り返されます。実際のフレーム送信は、送信 FIFO がプログラマブル送信閾値 (ETH\_DMAOMR レジスタ [16:14]) に達するか、FIFO に完全なフレームが格納された後、開始されます。ストアアンドフォワードモード (ETH\_DMAOMR レジスタ [21]) のオプションもあります。DMA がフレームの転送を終了すると、ディスクリプタはリリースされます (TDES0[31] の OWN ビットがクリアされます)。

## 送信ポーリングのサスペンド

送信ポーリングは、以下の条件のいずれかによって中断できます。

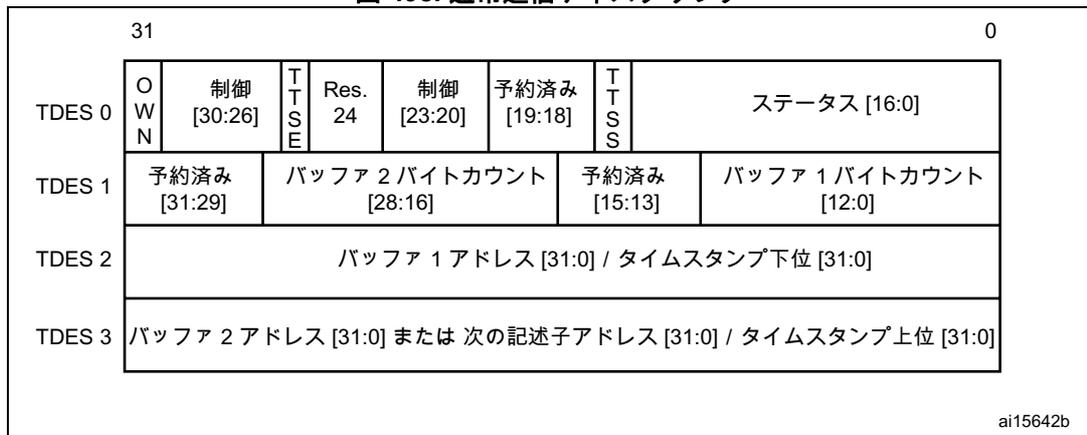
- DMA が CPU によって所有されるディスクリプタ (TDES0[31]=0) を検出したときと、送信バッファ使用不可フラグがセットされたとき (ETH\_DMASR レジスタ [2])。再開するには、ドライバは、ディスクリプタの所有権を DMA に与えた後、ポーリング要求コマンドを発行する必要があります。
- アンダーフローによる送信エラーが検出されると、フレーム送信は中止されます。適切な送信ディスクリプタ 0 (TDES0) ビットがセットされます。2 番目の条件が発生した場合、異常割り込み要約ビット (ETH\_DMASR レジスタ [15]) と送信アンダーフロービット (ETH\_DMASR レジスタ [5]) の両方がセットされ、情報が送信ディスクリプタ 0 に書き込まれて、送信は中断されます。DMA が最初の条件のためにサスペンド状態に移行した場合、正常割り込み要約ビット (ETH\_DMASR レジスタ [16]) と送信バッファ使用不可ビット (ETH\_DMASR レジスタ [2]) の両方がセットされます。どちらの場合も、送信リスト内の位置は維持されます。維持される位置は、DMA によってクローズされた最後のディスクリプタに続くディスクリプタの位置です。ドライバは、中断の原因を修正した後、送信ポーリング要求コマンドを明示的に発行しなければなりません。

## 通常 Tx DMA ディスクリプタ

通常送信ディスクリプタの構造は、[図 498](#) に示されているように、4 つの 32 ビットワードで構成されます。TDES0、TDES1、TDES2、および TDES3 のビットの説明を以下に示します。

タイムスタンプングがアクティブの場合 (ETH\_PTPTSCR ビット 0、TSE=1)、または IPv4 チェックサムオフロードがアクティブの場合 (ETH\_MACCCR ビット 10、IPCO=1)、拡張ディスクリプタを使う必要があることに注意してください。

図 498. 通常送信ディスクリプタ



● **TDES0 : 送信ディスクリプタワード 0**

アプリケーションソフトウェアは、ディスクリプタの初期化の際に、制御ビット [30:26]+[23:20]に加えて、OWN ビット [31] もプログラムする必要があります。DMA がディスクリプタを更新する (または書き戻す) ときには、すべての制御ビットに加えて OWN ビットもリセットして、ステータスビットのみを報告します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	IC	LS	FS	DC	DP	TTSE	Res.	CIC		TER	TCH	Res.		TTSS	IHE
rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	IPE	LCA	NC	LCO	EC	VF	CC				ED	UF	DB
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **OWN** : オウンビット

セットされた場合、このビットは、ディスクリプタが DMA によって所有されていることを示します。このビットがリセットされた場合、ディスクリプタが CPU によって所有されていることを示します。DMA は、フレーム送信を完了したとき、またはディスクリプタの中で割り当てられたバッファが完全に読み出されたときに、このビットをクリアします。フレームの最初のディスクリプタの所有権ビットは、同じフレームに属するすべての後続ディスクリプタがセットされた後にセットされなければなりません。

ビット 30 **IC** : 完了割り込み

セットされると、このビットは、現在のフレームが送信された後、送信割り込みビット (レジスタ 5[0]) をセットします。

ビット 29 **LS** : 最終セグメント

セットされた場合、このビットは、バッファがフレームの最後のセグメントを含んでいることを示します。

ビット 28 **FS** : 先頭セグメント

セットされた場合、このビットは、バッファがフレームの最初のセグメントを含んでいることを示します。

ビット 27 **DC** : CRC を無効化

このビットがセットされると、MAC は、送信フレームの最後に巡回冗長検査コード (CRC) を付加しません。これは、最初のセグメント (TDES0[28]) がセットされたときのみ有効です。

ビット 26 **DP** : パッドを無効化

このビットがセットされると、MAC は、64 バイトより短いフレームにはパッドを自動的に追加しません。このビットがリセットされると、DMA は、64 バイトより短いフレームに自動的にパッドと CRC を追加し、DC (TDES0[27]) ビットの状態に関係なく、CRC フィールドが追加されます。これは、最初のセグメント (TDES0[28]) がセットされたときのみ有効です。

ビット 25 **TTSE** : 送信タイムスタンプ有効化

TTSE がセットされ、TSE がセットされると (ETH\_PTPTSCR ビット 0)、ディスクリプタによって記述された送信フレームに対して、IEEE 1588 ハードウェアタイムスタンプングが有効になります。このフィールドは、最初のセグメント制御ビット (TDES0[28]) がセットされたときのみ有効です。

ビット 24 予約済みであり、リセット値に保持する必要があります。

**ビット 23:22 CIC** : チェックサム挿入制御

これらのビットは、チェックサムの計算と挿入を制御します。ビットエンコーディングは、以下のとおりです。

00 : チェックサム挿入を無効化

01 : IP ヘッダチェックサムのみ計算と挿入を有効化

10 : IP ヘッダチェックサムとペイロードチェックサムの計算と挿入が有効ですが、擬似ヘッダチェックサムはハードウェアでは計算されません。

11 : IP ヘッダチェックサムとペイロードチェックサムの計算と挿入が有効であり、擬似ヘッダチェックサムはハードウェアで計算されます。

**ビット 21 TER** : 送信リングの終わり

セットされた場合、このビットは、ディスクリプタリストが最後のディスクリプタに達したことを示します。DMA は、リストのベースアドレスに戻り、ディスクリプタリングを形成します。

**ビット 20 TCH** : セカンドアドレス連鎖

セットされると、このビットは、ディスクリプタ内の 2 番目のアドレスは 2 番目のバッファアドレスではなく、次のディスクリプタのアドレスであることを示します。TDES0[20] がセットされると、TBS2 (TDES1[28:16]) の値は「don't care」値です。TDES0[21] は、TDES0[20] に優先します。

**ビット 19:18** 予約済みであり、リセット値に保持する必要があります。**ビット 17 TTSS** : 送信タイムスタンプステータス

このフィールドは、記述された送信フレームのタイムスタンプがキャプチャされたことを示すためのステータスビットとして使用されます。このビットがセットされると、TDES2 と TDES3 は、送信フレームに対してキャプチャされたタイムスタンプの値を持ちます。このフィールドは、ディスクリプタの最後のセグメント制御ビット (TDES0[29]) がセットされたときのみ有効です。

拡張ディスクリプタを有効にした場合 (ETH\_DMABMR で EDFE=1)、TTSS=1 は TDES6 と TDES7 がタイムスタンプ値を持つことを表示することに注意してください。

**ビット 16 IHE** : IP ヘッダエラー

セットされた場合、このビットは、MAC トランスミッタが IP データグラムヘッダにエラーを検出したことを示します。トランスミッタは、IPv4 パケットのヘッダ長とアプリケーションから受信したヘッダバイト数を比較照合して、不一致があった場合はエラーステータスを示します。IPv6 フレームの場合、ヘッダエラーは、メインヘッダ長が 40 バイトでない場合に報告されます。さらに、IPv4 または IPv6 フレームのイーサネットの長さ/タイプフィールドの値は、パケットとともに受信された IP ヘッダのバージョンに一致しなければなりません。IPv4 フレームの場合、ヘッダ長フィールドの値が 0x5 未満の場合にも、エラーステータスが示されます。

**ビット 15 ES** : エラー一覧

以下のビットの論理 OR を示します。

TDES0[14] : ジャバタイムアウト

TDES0[13] : フレームフラッシュ

TDES0[11] : キャリア喪失

TDES0[10] : キャリアなし

TDES0[9] : レイトコリジョン

TDES0[8] : 過剰コリジョン

TDES0[2] : 過剰遅延

TDES0[1] : アンダーフローエラー

TDES0[16] : IP ヘッダエラー

TDES0[12] : IP ペイロードエラー

**ビット 14 JT** : ジャバタイムアウト

セットされた場合、このビットは、MAC トランスミッタがジャバタイムアウトを検出しことを示します。このビットは、MAC 設定レジスタの JD ビットがセットされていないときのみ有効です。

**ビット 13 FF** : フレームフラッシュ

セットされた場合、このビットは、DMA/MTL が、CPU によって与えられたソフトウェアフラッシュコマンドにより、フレームをフラッシュしたことを示します。



**ビット 12 IPE** : IP ペイロードエラー

セットされた場合、このビットは、MAC トランスミッタが TCP、UDP、または ICMP IP データグラムペイロード内でエラーを検出したことを示します。トランスミッタは、IPv4 または IPv6 ヘッダで受信したペイロード長を、アプリケーションから受信した TCP、UDP、または ICMP パケットバイトの実際の数と比較照合して、不一致があった場合はエラーステータスを発行します。

**ビット 11 LCA** : キャリア喪失

セットされた場合、このビットは、フレーム送信中にキャリアの喪失が発生した (すなわち、フレーム送信中に 1 または 2 送信クロック周期の間、MII CRS 信号が無効であった) ことを示します。このビットは、MAC が半二重モードで動作しているとき、コリジョンなしで送信されたフレームについてのみ有効です。

**ビット 10 NC** : キャリアなし

セットされた場合、このビットは、送信中に PHY からのキャリアセンス信号がアサートされなかったことを示します。

**ビット 9 LCO** : レイトコリジョン

セットされた場合、このビットは、コリジョンウィンドウ (MII モードで、プリアンブルを含めて 64 バイト時間) の後でコリジョンが発生したためにフレーム送信が中止されたことを示します。このビットは、アンダーフローエラーがセットされた場合は無効です。

**ビット 8 EC** : 過剰コリジョン

セットされた場合、このビットは、現在のフレームの送信を試みている間に 16 の連続したコリジョンが発生した後、送信が中止されたことを示します。MAC 設定レジスタの RD (再試行ディセーブル) ビットがセットされている場合、このビットが最初のコリジョン後にセットされて、フレームの送信が中止されます。

**ビット 7 VF** : VLAN フレーム

セットされた場合、このビットは、送信されたフレームは VLAN タイプであったことを示します。

**ビット 6:3 CC** : コリジョンカウント

この 4 ビットのカウンタの値は、フレームが送信される前に発生したコリジョンの回数を示します。このカウンタは、過剰コリジョンビット (TDES0[8]) がセットされているときには無効です。

**ビット 2 ED** : 過剰遅延

セットされた場合、このビットは、MAC 制御レジスタの遅延チェック (DC) ビットがハイにセットされている場合、24,288 ビット時間を超える過剰遅延があったために送信が終了したことを示します。

**ビット 1 UF** : アンダーフローエラー

セットされた場合、このビットは、RAM メモリからのデータが遅れて到着したために、MAC がフレーム送信を中止したことを示します。アンダーフローエラーは、DMA がフレーム送信中に空の送信バッファを検出したことを示します。送信プロセスはサスペンド状態に移行し、送信アンダーフロービット (レジスタ 5[5]) と送信割り込みビット (レジスタ 5[0]) の両方をセットします。

**ビット 0 DB** : 遅延ビット

セットされた場合、このビットは、キャリアの存在のために MAC がフレーム送信を延期したことを示します。このビットは、半二重モードでのみ有効です。

● **TDES1 : 送信ディスクリプタワード 1**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み		TBS2														予約済み		TBS1													
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

31:29 予約済みであり、リセット値に保持する必要があります。

28:16 **TBS2** : 送信バッファ 2 サイズ

これらのビットは、第 2 のデータバッファのサイズをバイト数で示します。このフィールドは、TDES0 [20] がセットされた場合は無効です。

15:13 予約済みであり、リセット値に保持する必要があります。

12:0 **TBS1** : 送信バッファ 1 サイズ

これらのビットは、第 1 のデータバッファのサイズをバイト数で示します。このフィールドが 0 の場合、DMA は、このバッファを無視して、TCH (TDES0[20]) の値に応じて、バッファ 2 または次のディスクリプタを使用します。

● **TDES2 : 送信ディスクリプタワード 2**

TDES2 は、ディスクリプタの最初のバッファのアドレスポインタ、またはタイムスタンプデータを含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBAP1/TBAP/TTSL																															
r/w																															

ビット 31:0 **TBAP1** : 送信バッファ 1 アドレスポインタ / 送信フレームタイムスタンプロー

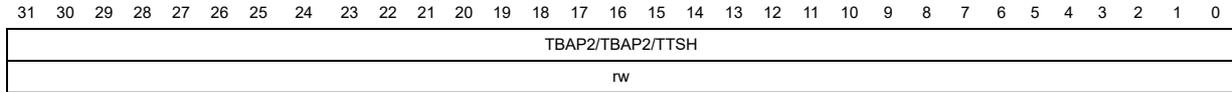
これらのビットは、メモリ内のデータのロケーションを DMA に知らせること、および全データ転送後に、DMA はこれらのビットを使ってタイムスタンプデータを返すという 2 つの機能を持ちます。

**TBAP** : ソフトウェアがこのディスクリプタを DMA から使用可能にすると (TDES0 の OWN ビットが 1 にセットされたとき)、これらのビットは、バッファ 1 の物理アドレスを示します。バッファアドレスの整列に関する制約はありません。バッファアドレスの整列の詳細については、[ホストデータバッファ整列 \(1515 ページ\)](#) を参照してください。

**TTSL** : TDES0 の OWN ビットをクリアする前に、DMA は、対応する送信フレームについてキャプチャされたタイムスタンプの下位 32 ビットでこのフィールドを更新します (TBAP1 の値を上書きします)。このフィールドは、このフレームについてタイムスタンプが有効であり (TTSE、TDES0 ビット 25 を参照)、ディスクリプタの最後のセグメント制御ビット (LS) がセットされている場合のみ、タイムスタンプの値を含みます。

- **TDES3 : 送信ディスクリプタワード 3**

TDES3 は、ディスクリプタまたは次のディスクリプタの第 2 のバッファのアドレスポインタ、またはタイムスタンプデータを含みます。



ビット 31:0 **TBAP2** : 送信バッファ 2 アドレスポインタ (次のディスクリプタアドレス) /送信フレームタイムスタンプハイ

これらのビットは、メモリ内のデータのロケーションを DMA に知らせること、および全データ転送後に、DMA はこれらのビットを使ってタイムスタンプデータを返すという 2 つの機能を持ちます。

**TBAP2** : ソフトウェアがこのディスクリプタを DMA から使用可能にすると (TDES0 の OWN ビットが 1 にセットされたとき)、これらのビットは、ディスクリプタリング構造が使用される時、バッファ 1 の物理アドレスを示します。セカンドアドレス連鎖 (TDES1 [24]) ビットがセットされた場合、このアドレスは、次のディスクリプタが存在する物理メモリのポインタを含みます。バッファアドレスポインタは、TDES1 [24] がセットされているときのみ、バス幅に整列されなければなりません。(LSB は内部で無視されます。)

**TTSH** : TDES0 の OWN ビットをクリアする前に、DMA は、対応する送信フレームのタイムスタンプの上位 32 ビットでこのフィールドを更新します (TBAP2 の値を上書きします)。このフィールドは、このフレームについてタイムスタンプが有効であり (TTSE、TDES0 ビット 25 を参照)、ディスクリプタの最後のセグメント制御ビット (LS) がセットされている場合のみ、タイムスタンプの値を含みます。

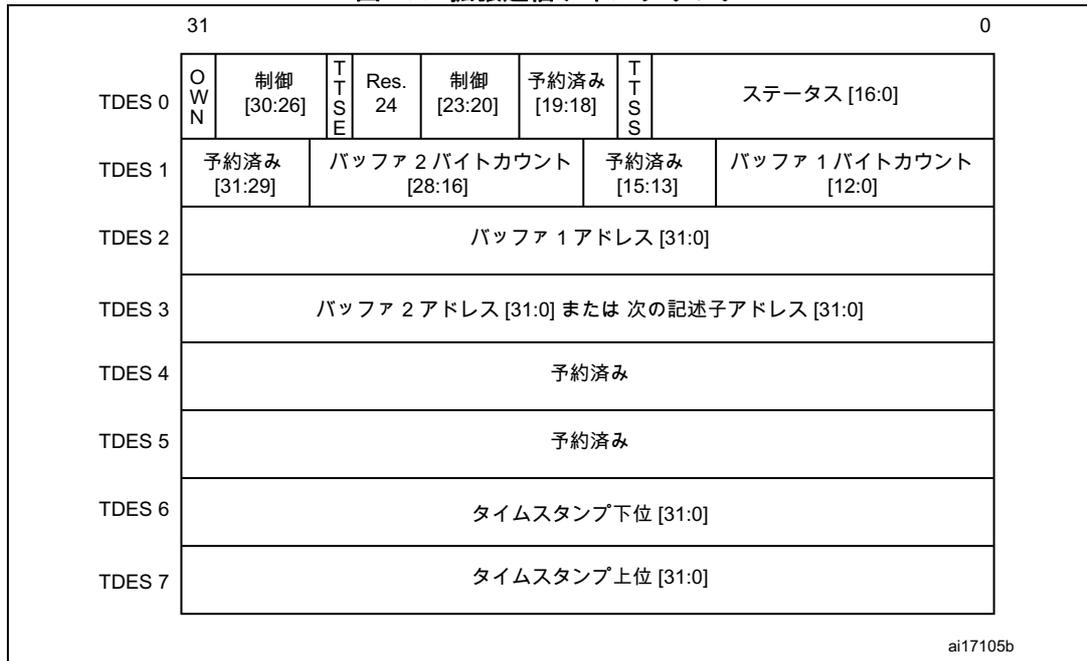
## 拡張 Tx DMA ディスクリプタ

タイムスタンプがアクティブの場合 (TSE=1、ETH\_PTPTSCR ビット 0)、または IPv4 チェックサムオフロードがアクティブの場合 (IPCO=1、ETH\_MACCR ビット 10)、拡張ディスクリプタ (EDFE=1、ETHDMABMR ビット 7 で有効化) を使う必要があります。

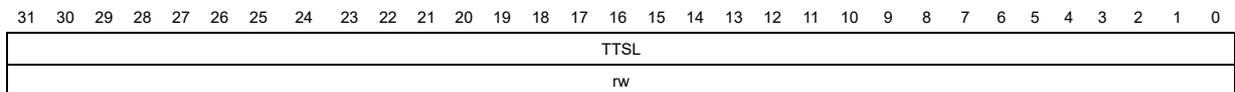
拡張ディスクリプタは 8 個の 32 ビットワードから構成され、通常ディスクリプタサイズの 2 倍です。TDES0、TDES1、TDES2、TDES3 の定義は、通常送信ディスクリプタと同じです ([通常 Tx DMA ディスクリプタ](#) を参照)。TDES6 と TDES7 はタイムスタンプを保持します。TDES4、TDES5、TDES6、TDES7 は次のように定義されます。

拡張ディスクリプタモードを選択する場合、ソフトウェアは 32 バイトのメモリ (8 DWORDS) を各ディスクリプタに割り当てる必要があります。タイムスタンプまたは IPv4 チェックサムオフロードを使用しない場合は、拡張ディスクリプタフォーマットを無効化でき、ソフトウェアは 16 バイトのデフォルトサイズを持つ通常ディスクリプタを使うことができます。

図 499. 拡張送信ディスクリプタ



- TDES4 : 送信ディスクリプタワード 4  
予約済み
- TDES5 : 送信ディスクリプタワード 5  
予約済み
- TDES6 : 送信ディスクリプタワード 6



ビット 31:0 TTSL : 送信フレームタイムスタンプロー

このフィールドは、DMA によって、対応する送信フレームについてキャプチャされたタイムスタンプの下位 32 ビットで更新されます。このフィールドは、ディスクリプタの最後のセグメント制御ビット (LS) がセットされている場合にのみ、タイムスタンプの値を持ちます。

● TDES7 : 送信ディスクリプタワード 7

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

ビット 31:0 TTSH : 送信フレームタイムスタンプハイ

このフィールドは、DMA によって、対応する送信フレームについてキャプチャされたタイムスタンプの上位 32 ビットで更新されます。このフィールドは、ディスクリプタの最後のセグメント制御ビット (LS) がセットされている場合にのみ、タイムスタンプの値を持ちます。

## 38.6.8 Rx DMA の設定

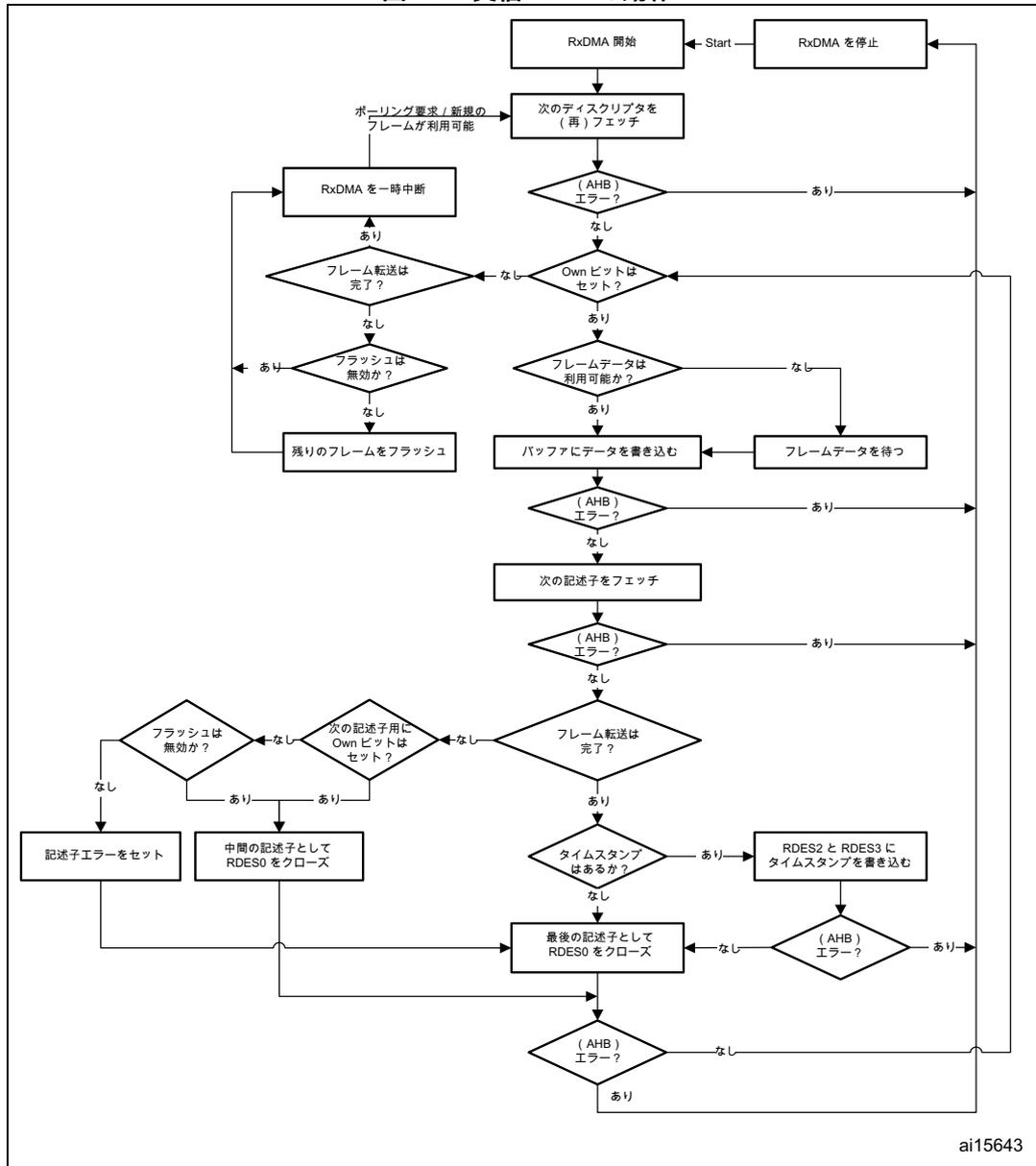
受信 DMA エンジンの受信シーケンスを [図 500](#) に示し、以下で説明します。

1. CPU は、受信ディスクリプタ (RDES0-RDES3) をセットアップし、OWN ビット (RDES0[31]) をセットします。
2. SR ビット (ETH\_DMAOMR レジスタの [13]) がセットされると、DMA は実行状態に入ります。実行状態の間、DMA は、受信ディスクリプタリストをポーリングして、未使用のディスクリプタの取得を試みます。フェッチされたディスクリプタが未使用でない (CPU に所有されている) 場合、DMA はサスペンド状態に入り、手順 9 へジャンプします。
3. DMA は、取得したディスクリプタから受信データバッファアドレスをデコードします。
4. 受信フレームは処理され、取得されたディスクリプタのデータバッファに格納されます。
5. バッファが満杯になるか、フレーム転送が完了すると、受信エンジンは次のディスクリプタをフェッチします。
6. 現在のフレーム転送が完了した場合、DMA は手順 7 に進みます。DMA がフェッチされた次のディスクリプタを所有せず、フレーム転送が完了していない (EOF がまだ転送されていない) 場合、DMA は、RDES0 のディスクリプタエラービットをセットします (フラッシュが無効にされていない場合)。DMA は現在のディスクリプタをクローズし (OWN ビットをクリア)、RDES1 値の最後のセグメント (LS) ビットをクリアすることによって、中間ディスクリプタとしてマークした後 (フラッシュが無効にされていない場合は最後のディスクリプタとしてマーク)、手順 8 に進みます。DMA が次のディスクリプタを所有しているが、現在のフレーム転送をまだ完了していない場合、DMA は現在のディスクリプタを中間ディスクリプタとしてクローズして、手順 4 に戻ります。
7. IEEE 1588 タイムスタンプが有効な場合、DMA は、現在のディスクリプタの RDES2 と RDES3 にタイムスタンプ (入手可能な場合) を書き込みます。DMA は、次に、受信したフレームのステータスを取り出し、そのステータスワードを、OWN ビットがクリアされ、最後のセグメントビット (LS) がセットされている現在のディスクリプタの RDES0 に書き込みます。
8. 受信エンジンは、最新のディスクリプタの OWN ビットをチェックします。CPU がディスクリプタを所有している場合 (OWN ビットが 0)、受信バッファ使用不可ビット (ETH\_DMASR レジスタ [7]) がセットされ、DMA 受信エンジンはサスペンド状態に入ります (手順 9)。DMA がディスクリプタを所有している場合、エンジンは手順 4 に戻り、次のフレームを待ちます。
9. 受信エンジンがサスペンド状態に入る前に、部分フレームは受信 FIFO からフラッシュされます (フラッシュは、TH\_DMAOMR レジスタのビット 24 を使用して制御できます)。
10. 受信 DMA は、ポーリング受信要求が与えられたとき、または受信 FIFO から次のフレームの開始が入手可能となったときに、サスペンド状態から抜けます。エンジンは手順 2 に進み、次のディスクリプタを再びフェッチします。

DMA は、タイムスタンプの書き戻しを完了して、ディスクリプタのステータスの書き戻しを実行する用意ができるまで、ステータスの受け取りを確認応答しません。ソフトウェアが CSR を通じてタイムスタンプを有効にした場合、フレームの有効なタイムスタンプの値が入手できないときは

(たとえば、タイムスタンプの書き込みが完了する前に受信 FIFO が満杯になったときなど)、DMA は RDES2 と RDES3 にすべて 1 を書き込みます。そうでない場合 (すなわち、タイムスタンプが有効でない場合)、RDES2 と RDES3 は変更なく維持されます。

図 500. 受信 DMA の動作



ai15643

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

## 受信ディスクリプタの取得

受信エンジンは、常に、受信フレームを期待して、ディスクリプタをさらに取得しようと試みます。以下の条件のいずれかが満たされた場合、ディスクリプタの取得が試みられます。

- DMA が実行状態に入った直後に、受信スタート/ストップビット (ETH\_DMAOMR レジスタ [1]) がセットされた。
- 現在のディスクリプタのデータバッファが、現在転送中のフレームの終わりの前に満杯である。
- コントローラがフレームの受信を完了したが、現在の受信ディスクリプタがまだクローズされていない。
- CPU 所有のバッファ (RDES0[31] = 0) のために受信プロセスが中断され、新しいフレームが受信された。
- 受信ポーリング要求が発行された。

## 受信フレームの処理

MAC は、フレームがアドレスフィルタを通過し、フレームサイズが受信 FIFO に対して設定された設定可能な閾値バイト以上であるか、完全なフレームがストアアンドフォワードモードで FIFO に書き込まれたときのみ、受信したフレームを STM32F75xxx および STM32F74xxx メモリに転送します。フレームがアドレスフィルタを通過できなかった場合は、MAC ブロック自身の中でドロップされず (すべて受信ビット (ETH\_MACFFR [31]) がセットされている場合を除きます)。コリジョンや早すぎる終了のために 64 バイトより短くなったフレームは、受信 FIFO から消去されます。64 バイト (設定可能な閾値) が受信された後、DMA ブロックは、現在のディスクリプタによって示される受信バッファへのフレームデータの転送を開始します。DMA は、DMA AHB インタフェースがデータ転送を受け取る準備ができた後 (DMA がメモリから送信データをフェッチしていない場合)、最初のディスクリプタ (RDES0[9]) をセットして、フレームを区切ります。ディスクリプタは、データバッファが満杯になるか、フレームの最後のセグメントが受信バッファに転送されたことによって、OWN (RDES0[31]) ビットが 0 にリセットされると、リリースされます。フレームが単一のディスクリプタに含まれている場合、最後のディスクリプタ (RDES0[8]) ビットと最初のディスクリプタ (RDES0[9]) ビットの両方がセットされます。DMA は次のディスクリプタをフェッチし、最後のディスクリプタ (RDES0[8]) ビットをセットし、前のフレームディスクリプタの RDES0 ステータスビットをリリースします。次に、DMA は、受信割り込みビット (ETH\_DMASR レジスタ [6]) をセットします。DMA が CPU によって所有されているというフラグが立てられているディスクリプタを検出しない限り、同じプロセスが繰り返されます。これが発生した場合、受信プロセスは受信バッファ使用不可ビット (ETH\_DMASR レジスタ [7]) をセットした後、サスペンド状態に入ります。受信リスト内の位置は維持されます。

## 中断された受信プロセス

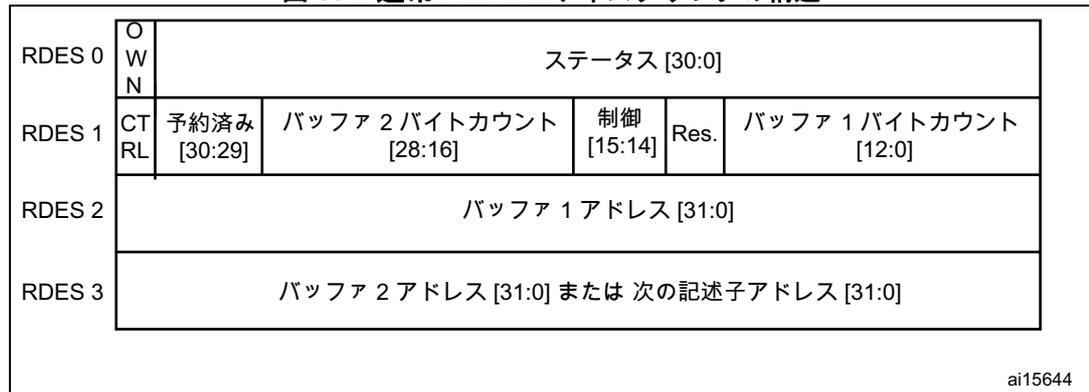
受信プロセスがサスペンド状態にある間に新しい受信フレームが到着した場合、DMA は、STM32F75xxx および STM32F74xxx メモリ内の現在のディスクリプタを再フェッチします。現在、ディスクリプタが DMA によって所有されている場合、受信プロセスは再び実行状態に入り、フレーム受信を開始します。ディスクリプタが依然としてホストによって所有されている場合、デフォルトでは、DMA は、Rx FIFO の最上部にある現在のフレームを破棄して、欠落フレームカウンタをインクリメントします。Rx FIFO 内に複数のフレームが格納されている場合、このプロセスが繰り返されます。Rx FIFO の最上部にあるフレームの破棄またはフラッシュは、DMA 動作モードレジスタのビット 24 (DFRF) をセットすることによって回避できます。このような条件では、受信プロセスは、受信バッファ使用不可ステータスビットをセットして、サスペンド状態に戻ります。

## 通常 Rx DMA ディスクリプタ

通常受信ディスクリプタの構造は、4 つの 32 ビットワード (16 バイト) で構成されます。これらを [図 501](#) に示します。RDES0、RDES1、RDES2、および RDES3 のビットの説明は、以下のとおりです。

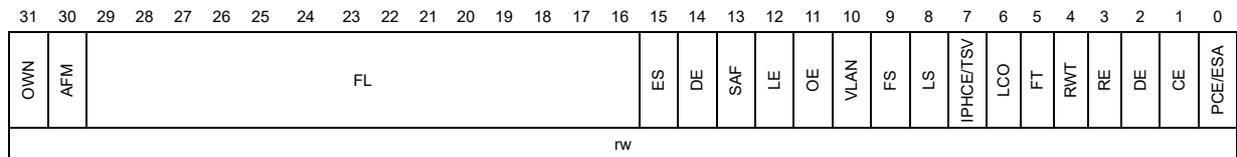
タイムスタンプがアクティブの場合 (TSE=1、ETH\_PTPTSCR ビット 0)、または IPv4 チェックサムオフロードがアクティブの場合 (IPCO=1、ETH\_MACCR ビット 10)、拡張ディスクリプタを使う必要があることに注意してください。

図 501. 通常 Rx DMA ディスクリプタの構造



- **RDES0 : 受信ディスクリプタワード 0**

RDES0 は、受信フレームステータス、フレーム長、およびディスクリプタの所有権情報を含んでいます。



**ビット 31 OWN : オウンビット**

セットされた場合、このビットは、ディスクリプタが MAC サブシステムの DMA によって所有されていることを示します。このビットがリセットされた場合は、ディスクリプタがホストによって所有されていることを示します。DMA は、フレーム受信を完了したとき、または、このディスクリプタに関連するバッファが満杯になったときに、このビットをクリアします。

**ビット 30 AFM : デスティネーションアドレスフィルタ失敗**

セットされた場合、このビットは、フレームが MAC コアで DA フィルタの通過に失敗したことを示します。

**ビット 29:16 FL : フレーム長**

これらのビットは、ホストメモリに転送された受信フレームのバイト長 (CRC を含む) を示します。このフィールドは、最後のディスクリプタ (RDES0[8]) がセットされ、ディスクリプタエラー (RDES0[14]) がリセットされたときのみ有効です。

このフィールドは、最後のディスクリプタ (RDES0[8]) がセットされたときに有効です。最後のディスクリプタビットとエラー要約ビットがセットされていない場合、このフィールドは、現在のフレームで転送された累積バイト数を示します。

**ビット 15 ES : エラー一覧**

以下のビットの論理 OR を示します。

RDES0[1] : CRC エラー

RDES0[3] : 受信エラー

RDES0[4] : ウォッチドッグタイムアウト

RDES0[6] : レイトコリジョン

RDES0[7] : ジャイアントフレーム (これは RDES0[7] が IPv4 ヘッダチェックサムエラーを示しているときは適用されません)。

RDES0[11] : オーバーフローエラー

RDES0[14] : ディスクリプタエラー。

このフィールドは、最後のディスクリプタ (RDES0[8]) がセットされたときにのみ有効です。

**ビット 14 DE : ディスクリプタエラー**

セットされた場合、このビットは、フレームが現在のディスクリプタバッファに入りきらず、DMA が次のディスクリプタを所有していないために、フレームが切り詰められることを示します。フレームは切り詰められます。このフィールドは、最後のディスクリプタ (RDES0[8]) がセットされたときにのみ有効です。

**ビット 13 SAF : ソースアドレスフィルタ失敗**

セットされた場合、このビットは、フレームの SA フィールドが MAC コアの SA フィルタを通過できなかったことを示します。

**ビット 12 LE : 長さエラー**

セットされた場合、このビットは、受信フレームの実際の長さが、長さ/タイプフィールドの値に一致しないことを示します。このビットは、フレームタイプ (RDES0[5]) ビットがリセットされたときにのみ有効です。

**ビット 11 OE : オーバーフローエラー**

セットされた場合、このビットは、受信フレームがバッファオーバーフローのために破損したことを示します。

**ビット 10 VLAN : VLAN タグ**

セットされた場合、このビットは、このディスクリプタによって示されるフレームが MAC コアによってタグ付けされた VLAN フレームであることを示します。

**ビット 9 FS : 先頭ディスクリプタ**

セットされた場合、このビットは、このディスクリプタがフレームの最初のバッファを含んでいることを示します。最初のバッファのサイズが 0 の場合、2 番目のバッファがフレームの開始を含みます。2 番目のバッファのサイズも 0 の場合、次のディスクリプタがフレームの開始を含みます。

**ビット 8 LS : 最終ディスクリプタ**

セットされた場合、このビットは、このディスクリプタによって示されるバッファがフレームの最後のバッファであることを示します。

**ビット 7 IPHCE/TSV : IPv ヘッダチェックサムエラー / タイムスタンプ有効**

IPHCE がセットされた場合は、IPv4 または IPv6 ヘッダにエラーがあることを示します。このエラーの原因は、イーサネットタイプフィールドと IP ヘッダバージョンフィールドの不一致、IPv4 でのヘッダチェックサムの不一致、またはイーサネットフレームが期待される数の IP ヘッダバイトを欠いていることなどが考えられます。このビットは、表 243 で規定するように特別な意味を持つことができます。拡張ディスクリプタフォーマットが有効な場合 (EDFE=1、ETH\_DMABMR のビット 7)、このビットは TSV 機能を持ちます (その他の場合は IPHCE)。TSV がセットされた場合、タイムスタンプのスナップショットがディスクリプタワード 6 (RDES6) と 7 (RDES7) に書き込まれたことを表示します。TSV は、最終ディスクリプタビット (RDES0[8]) がセットされた場合にのみ有効です。

**ビット 6 LCO : レイトコリジョン**

セットされた場合、このビットは、半二重モードでフレームを受信中に、レイトコリジョンが発生したことを示します。

**ビット 5 FT : フレームタイプ**

セットされた場合、このビットは、受信フレームがイーサネットタイプフレーム (LT フィールドが 0x0600 以上) であることを示します。このビットがリセットされているときは、受信フレームが IEEE802.3 フレームであることを示します。このビットは、14 バイト未満のラントフレームについては無効です。通常ディスクリプタフォーマットを使用する場合 (ETH\_DMABMR EDFE=0)、FT は表 243 に規定される特別な意味を持つことができます。

**ビット 4 RWT : 受信ウォッチドッグタイムアウト**

セットされた場合、このビットは、現在のフレームの受信中に受信ウォッチドッグタイムが時間切れになり、ウォッチドッグタイムアウト後に現在のフレームが切り詰められることを示します。

**ビット 3 RE : 受信エラー**

セットされた場合、このビットは、フレーム受信時に RX\_DV がアサートされているときに、RX\_ERR 信号がアサートされたことを示します。

**ビット 2 DE : ドリブルビットエラー**

セットされた場合、このビットは、受信フレームのバイト数が非整数倍 (奇数ニブル) であることを示します。このビットは、MII モードでのみ有効です。

**ビット 1 CE : CRC エラー**

セットされると、このビットは、受信フレームで巡回冗長検査 (CRC) エラーが発生したことを示します。このフィールドは、最後のディスクリプタ (RDES0[8]) がセットされたときにのみ有効です。

**ビット 0 PCE/ESA : ペイロードチェックサムエラー/拡張ステータス使用可能**

セットされた場合、このビットは、コアが計算した TCP、UDP、または ICMP チェックサムが、受信したカプセル化 TCP、UDP、または ICMP セグメントのチェックサムフィールドに一致しないことを示します。このビットは、受信したペイロードのバイト数が、受信したイーサネットフレームのカプセル化された IPv4 または IPv6 データグラムの長さフィールドの値と一致しないときにもセットされます。このビットは、表 243 で規定するように特別な意味を持つことができます。

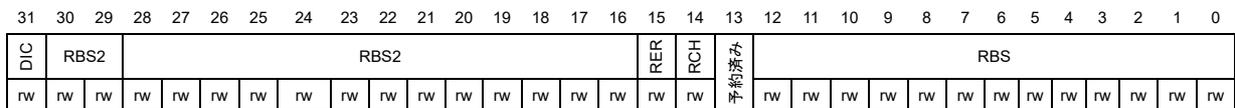
拡張ディスクリプタフォーマットが有効な場合 (EDFE=1、ETH\_DMABMR のビット 7)、このビットは TSV 機能を持ちます (その他の場合は PCE)。ESA をセットした場合、拡張ステータスがディスクリプタワード 4 (RDES4) で使用可能なことを表示します。ESA は、最終ディスクリプタビット (RDES0[8]) がセットされた場合にのみ有効です。

ビット 5、7、および 0 は、表 243 に示されている条件を反映します。

表 243. 受信ディスクリプタ 0 - ビット 7、5、0 のエンコーディング (通常ディスクリプタフォーマットのみ、EDFE=0)

ビット 5 : フレーム タイプ	ビット 7 : IPC チェック サムエラー	ビット 0 : ペイロード チェックサム エラー	フレームステータス
0	0	0	IEEE 802.3 タイプフレーム (長さフィールドの値が 0x0600 未満)。
1	0	0	チェックサムエラーが検出されない IPv4/IPv6 タイプのフレーム。
1	0	1	ペイロードチェックサムエラーが検出される IPv4/IPv6 タイプのフレーム (PCE の説明と同様)。
1	1	0	IP ヘッダチェックサムエラーが検出される IPv4/IPv6 タイプのフレーム (IPC CE の説明と同様)。
1	1	1	IP ヘッダとペイロードの両方のチェックサムエラーが検出される IPv4/IPv6 タイプのフレーム。
0	0	1	IP ヘッダチェックサムエラーはないが、サポートされないペイロードであるため、ペイロードチェックがバイパスされる IPv4/IPv6 タイプのフレーム。
0	1	1	IPv4 でも IPv6 でもないタイプのフレーム (チェックサムオフロードエンジンはチェックサムを完全にバイパスします)。
0	1	0	予約済み

● RDES1 : 受信ディスクリプタワード 1



ビット 31 DIC : 完了時割り込みディセーブル

セットされた場合、このビットは、このディスクリプタによって示されたバッファで終わる受信フレームについて、ステータスレジスタの RS ビット (CSR5[6]) がセットされるのを防ぎます。これにより、そのフレームの RS によるホストへの割り込みのアサートが無効になります。

ビット 30:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:16 RBS2 : 受信バッファ 2 サイズ

これらのビットは、第 2 のデータバッファのサイズをバイト数で示します。バッファサイズは、RDES3 (バッファ 2 アドレスポインタ) がバス幅に整列されていない場合でも、バス幅 (32、64、または 128 ビット) に応じて、それぞれ 4、8、または 16 の倍数でなければなりません。バッファサイズが 4、8、または 16 の適切な倍数でない場合、結果としての動作は未定義です。このフィールドは、RDES1 [14] がセットされた場合は無効です。

ビット 15 RER : 送信リングの終わり

セットされた場合、このビットは、ディスクリプタリストが最後のディスクリプタに達したことを示します。DMA は、リストのベースアドレスに戻り、ディスクリプタリングを形成します。

ビット 14 **RCH** : セカンドアドレス連鎖

セットされると、このビットは、ディスクリプタ内の 2 番目のアドレスは 2 番目のバッファアドレスではなく、次のディスクリプタのアドレスであることを示します。このビットがセットされると、RBS2 (RDES1[28:16]) の値は「don't care」になります。RDES1[15] は、RDES1[14] に優先します。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12:0 **RBS1** : 受信バッファ 1 サイズ

第 1 のデータバッファのサイズをバイト数で示します。バッファサイズは、RDES2 (バッファ 1 アドレスポインタ) がバス幅に整列されていない場合でも、バス幅 (32、64、または 128 ビット) に応じて、それぞれ 4、8、または 16 の倍数でなければなりません。バッファサイズが 4、8、または 16 の倍数でない場合、結果としての動作は未定義です。このフィールドが 0 の場合、DMA は、このバッファを無視して、RCH (ビット[14]) の値に応じて、バッファ 2 または次のディスクリプタを使用します。

● **RDES2 : 受信ディスクリプタワード 2**

RDES2 は、ディスクリプタの最初のデータバッファのアドレスポインタ、またはタイムスタンプデータを含まれます。



ビット 31:0 **RBAP1 / RTSL** : 受信バッファ 1 アドレスポインタ / 受信フレームタイムスタンプロー

これらのビットは、2 つの機能で使用されます。アプリケーションは、これらのビットを使用して、データを書き込むべきメモリ内の位置を DMA に知らせ、すべてのデータを転送した後、DMA は、これらのビットを使用して、タイムスタンプデータを返送できます。

**RBAP1** : ソフトウェアがこのディスクリプタを DMA から使用可能にする (RDES0 の OWN ビットが 1 にセットされたとき)、これらのビットは、バッファ 1 の物理アドレスを示します。バッファアドレスの配列には制約はありませんが、以下の条件を除きます。すなわち、DMA は、RDES2 の値がフレームの開始を格納するために使用される場合には、そのアドレス生成のために設定された値を使用します。DMA は、フレームの開始の転送時には、RDES2[3/2/1:0] ビットを 0 として書き込み動作を実行しますが、フレームデータは実際のバッファアドレスポインタに従ってシフトされることに注意してください。DMA は、アドレスポインタがフレームの中間または最後の部分が格納されるバッファを示している場合、RDES2[3/2/1:0] (128/64/32 のバス幅に対応) を無視します。

**RTSL** : RDES0 の OWN ビットをクリアする前に、DMA は、対応する受信フレームについてキャプチャされたタイムスタンプの下位 32 ビットでこのフィールドを更新します (RBAP1 の値を上書きします)。このフィールドは、このフレームについてタイムスタンプが有効であり、ディスクリプタの最後のセグメント制御ビット (LS) がセットされている場合のみ、タイムスタンプの値を含みます。



● **RDES3 : 受信ディスクリプタワード 3**

RDES3 は、ディスクリプタ内の第 2 のデータバッファまたは次のディスクリプタのいずれかのアドレスポインタ、またはタイムスタンプデータを含みます。



ビット 31:0 **RBAP2 / RTSH** : 受信バッファ 2 アドレスポインタ (次のディスクリプタのアドレス) / 受信フレームタイムスタンプハイ

これらのビットは、2つの機能で使用されます。アプリケーションは、これらのビットを使用して、データを書き込むべきメモリ内の位置を DMA に知らせ、すべてのデータを転送した後、DMA は、これらのビットを使用して、タイムスタンプデータを返送できます。

**RBAP1** : ソフトウェアがこのディスクリプタを DMA から使用可能にすると (RDES0 の OWN ビットが 1 にセットされたとき)、これらのビットは、ディスクリプタリング構造が使用される時、バッファ 2 の物理アドレスを示します。セカンドアドレス連鎖 (RDES1 [24]) ビットがセットされた場合、このアドレスは、次のディスクリプタが存在する物理メモリのポインタを含みます。RDES1 [24] がセットされた場合、バッファ (次のディスクリプタ) アドレスポインタは、128、64、または 32 のバス幅に応じて、RDES3[3、2、または 1:0] = 0)、バス幅に整列されていなければなりません。(LSB は内部的に無視されます。) ただし、RDES1 [24] がリセットされた場合、RDES3 の値には制約はありませんが、以下の条件を除きます。すなわち、DMA は、RDES3 の値がフレームの開始を格納するために使用される時には、そのバッファアドレス生成のために設定された値を使用します。DMA は、アドレスポインタがフレームの中間または最後の部分が格納されるバッファを示している場合、RDES3[3、2、または 1:0] (128、64、32 のバス幅に対応) を無視します。

**RTSH**: RDES0 の OWN ビットをクリアする前に、DMA は、対応する受信フレームについてキャプチャされたタイムスタンプの上位 32 ビットでこのフィールドを更新します (RBAP2 の値を上書きします)。このフィールドは、このフレームについてタイムスタンプが有効であり、ディスクリプタの最終セグメントコントロールビット (LS) がセットされているケースでのみ、タイムスタンプを持ちます。

**IEEE1588 タイムスタンプを持つ Rx DMA ディスクリプタのフォーマット**

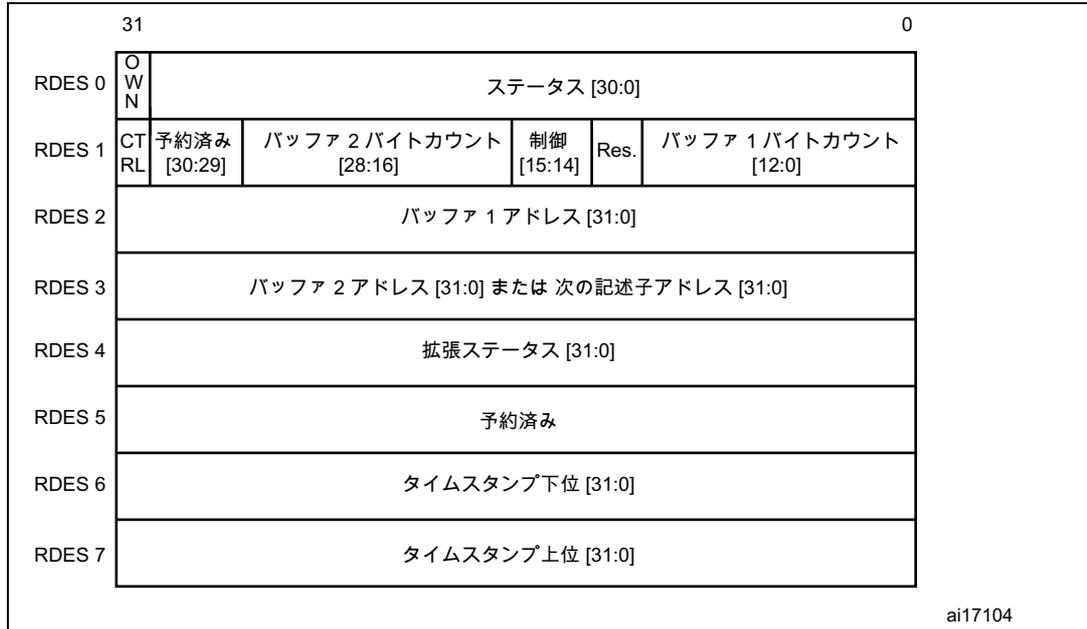
タイムスタンプがアクティブの場合 (TSE=1、ETH\_PTPTSCR ビット 0)、または IPv4 チェックサムオフロードがアクティブの場合 (IPCO=1、ETH\_MACCR ビット 10)、拡張ディスクリプタ (EDFE=1、ETHDMABMR ビット 7 で有効化) を使う必要があります。

拡張ディスクリプタは 8 個の 32 ビットワードから構成され、通常ディスクリプタサイズの 2 倍です。RDES0、RDES1、RDES2、RDES3 の定義は、通常受信ディスクリプタと同じです ([通常 Rx DMA ディスクリプタ](#)を参照)。RDES4 は拡張ステータスを、RDES6 と RDES7 はタイムスタンプをそれぞれ保持します。RDES4、RDES5、RDES6、RDES7 は次のように定義されます。

拡張ディスクリプタモードを選択する場合、ソフトウェアは 32 バイトのメモリ (8 DWORDS) を各ディスクリプタに割り当てる必要があります。タイムスタンプまたは IPv4 チェックサムオフロードを使用しない場合は、拡張ディスクリプタフォーマットを無効化でき、ソフトウェアは 16 バイトのデフォルトサイズを持つ通常ディスクリプタを使うことができます。



図 502. IEEE1588 タイムスタンプが有効な拡張受信ディスクリプタフィールドのフォーマット



- RDES4 : 受信ディスクリプタワード 4

下に示す拡張ステータスは、IPv4 チェックサムに関するステータスまたは RDES0 のビット 0 で指定されるタイムスタンプが存在する場合にのみ有効です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	PV	PFT	PMT					IPV6PR	IPV4PR	IPCB	IPPE	IPHE	IPPT																			
																		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **PV** : PTP バージョン

セットされている場合、受信 PTP メッセージは IEEE 1588 バージョン 2 フォーマットを使用していることを表示します。クリアされている場合は、バージョン 1 フォーマットを使用していることを表示します。これは、メッセージタイプが非ゼロの場合のみ有効です。

ビット 12 **PFT** : PTP フレームタイプ

セットされている場合、このビットは PTP メッセージがイーサネットを介して直接送信されることを表示します。このビットがクリアされていて、かつメッセージタイプが非ゼロの場合、PTP メッセージが UDP-IPv4 または UDP-IPv6 を使用して送信されることを表示します。IPv4 または IPv6 の情報は、ビット 6 とビット 7 から取得することができます。

ビット 11:8 **PMT** : PTP メッセージタイプ

これらのビットは、受信メッセージのタイプを表すようにエンコードされます。

- 0000 : 受信 PTP メッセージなし
- 0001 : SYNC (全クロックタイプ)
- 0010 : Follow\_Up (全クロックタイプ)
- 0011 : Delay\_Req (全クロックタイプ)
- 0100 : Delay\_Resp (全クロックタイプ)
- 0101 : Pdelay\_Req (ピア・ツー・ピアトランスペアレントクロック) またはアナウンス (通常または境界クロック)
- 0110 : Pdelay\_Resp (ピア・ツー・ピアトランスペアレントクロック) または管理 (通常または境界クロック)
- 0111 : Pdelay\_Resp\_Follow\_Up (ピア・ツー・ピアトランスペアレントクロック) またはシグナリング (通常または境界クロック)
- 1xxx - 予約済み

ビット 7 **IPV6PR** : IPv6 受信パケット

セットされている場合、このビットは受信パケットが IPv6 パケットであることを表示します。

ビット 6 **IPV4PR** : IPv4 受信パケット

セットされている場合、このビットは受信パケットが IPv4 パケットであることを表示します。

ビット 5 **IPCB** : IP チェックサムのパイパス

セットされている場合、このビットはチェックサムオフロードエンジンがバイパスされていることを表示します。

ビット 4 **IPPE** : IP ペイロードエラー

セットされている場合、このビットは、コアが計算した 16 ビット IP ペイロードチェックサム (すなわち、TCP、UDP、または ICMP チェックサム) が、受信セグメント内の対応するチェックサムフィールドに一致しないことを表示します。これは、TCP、UDP、または ICMP セグメント長が IP ヘッダフィールド内のペイロード長に一致しない場合にも送信されます。

ビット 3 **IPHE** : IP ヘッダエラー

セットされている場合、このビットは、アが計算した 16 ビット IPv4 ヘッダチェックサムが受信チェックサムバイトに一致しないこと、または IP データグラムバージョンがイーサネットタイプ値と矛盾することを表示します。

ビット 2:0 **IPPT** : IP ペイロードタイプ

IPv4 チェックサムオフロードがアクティブの場合 (IPCO=1、ETH\_MACCCR ビット 10)、これらのビットは、IP データグラム内のペイロードのタイプを表示します。IP ヘッダエラーまたはフラグメント化した IP が存在する場合にこれらのビットは "00" になります。

- 000 : 未知または IP ペイロードの処理なし
- 001 : UDP
- 010 : TCP
- 011 : ICMP
- 1xx : 予約済み

● **RDES5** : 受信ディスクリプタワード 5

予約済み

● **RDES6 : 受信ディスクリプタワード 6**

受信ディスクリプタがクローズされ、タイムスタンプが有効なとき、RDES6 に対して異なる意味を持つフィールドを次の表に示します。

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

ビット 31:0 **RTSL** : 受信フレームタイムスタンプロー

DMA は、このフィールドを、対応する受信フレームについてキャプチャされたタイムスタンプの下位 32 ビットで更新します。DMA は、最後のディスクリプタのステータスビット (RDES0[8]) によって示された受信フレームの最後のディスクリプタについてのみ、このフィールドを更新します。このフィールドと RDES7 の RTSH フィールドがすべて 1 を示している場合、タイムスタンプは破損しているものとして扱われなければなりません。

● **RDES7 : 受信ディスクリプタワード 7**

受信ディスクリプタがクローズされ、タイムスタンプが有効なとき、RDES7 に対して異なる意味を持つフィールドを次の表に示します。

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

ビット 31:0 **RTSH** : 受信フレームタイムスタンプハイ

DMA は、このフィールドを、対応する受信フレームについてキャプチャされたタイムスタンプの上位 32 ビットで更新します。DMA は、最後のディスクリプタのステータスビット (RDES0[8]) によって示された受信フレームの最後のディスクリプタについてのみ、このフィールドを更新します。このフィールドと RDES7 の RTSL フィールドがすべて 1 を示している場合、タイムスタンプは破損しているものとして扱われなければなりません。

## 38.6.9 DMA 割り込み

さまざまなイベントの結果として、割り込みが生成されることがあります。ETH\_DMASR レジスタは、割り込みの原因となる可能性があるすべてのビットを含んでいます。ETH\_DMAIER レジスタは、割り込みの原因になりえる各イベントのイネーブルビットを含んでいます。

割り込みには、ETH\_DMASR レジスタで説明されているように、正常と異常の 2 つのグループがあります。割り込みは、対応するビット位置に 1 を書き込むことによってクリアされます。有効にされていたグループ内のすべての割り込みがクリアされると、対応する要約ビットがクリアされます。MAC コアが割り込みアサートの原因である場合、ETH\_DMASR レジスタの TSTS または PMTS ビットのいずれかがハイにセットされます。

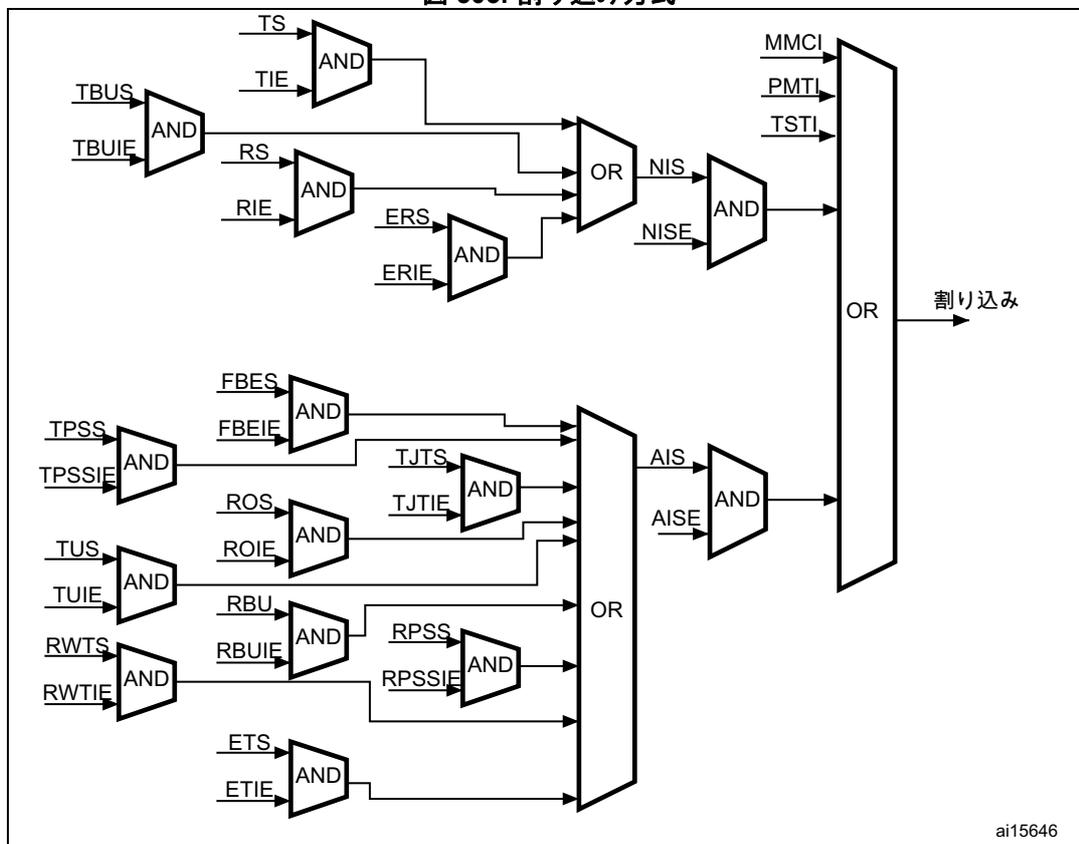
割り込みは待ち行列を作らないので、ドライバが割り込みに応答する前に別の割り込みイベントが発生した場合、追加の割り込みは生成されません。たとえば、受信割り込みビット (ETH\_DMASR レジスタ [6]) が STM32F75xxx および STM32F74xxx バッファに 1 つまたは複数のフレームが転送されたことを示したとします。ドライバは、最後に記録された位置から DMA によって所有される最初のディスクリプタまで、すべてのディスクリプタをスキャンする必要があります。

割り込みは、同時に発生した複数のイベントについて 1 つしか生成されません。ドライバは、ETH\_DMASR レジスタをスキャンして、割り込みの原因を調べなければなりません。ドライバが ETH\_DMASR レジスタの適切なビットをクリアした後は、新しい割り込みイベントが発生しない限り、割り込みが再び生成されることはありません。たとえば、コントローラが受信割り込み (ETH\_DMASR レジスタ [6]) を生成して、ドライバが ETH\_DMASR レジスタの読み出しを開始した



とします。次に、受信バッファ使用不可ビット (ETH\_DMASR レジスタ [7]) がセットされます。ドライバは、受信割り込みをクリアします。その場合でも、有効状態またはペンディング中の受信バッファ使用不可割り込みにより、新しい割り込みが生成されます。

図 503. 割り込み方式



## 38.7 イーサネット割り込み

イーサネットコントローラには、2つの割り込みベクタがあります。1つは、通常のイーサネット動作であり、もう1つはEXTIライン19に配置されたときにイーサネットウェイクアップイベント(ウェイクアップフレームまたはマジックパケット検出時)にのみ使用されます。

最初のイーサネットベクタは、**MAC 割り込み**および**DMA 割り込み**のセクションにリストされているような、MAC および DMA によって生成される割り込みのために確保されています。

2番目のベクタは、ウェイクアップイベント時にPMTによって生成される割り込みのために確保されています。EXTIライン19上のウェイクアップイベントのマッピングは、STM32F75xxx および STM32F74xxx を低電力モードから抜け出させ、割り込みを生成します。

EXTIライン19上でイーサネットウェイクアップイベントのマッピングが発生し、MAC PMT 割り込みが有効にされ、立ち上がりエッジでの検出が指定されたEXTIライン19割り込みも有効にされると、両方の割り込みが生成されます。

RSビット(ETH\_DMASRレジスタ)の柔軟な制御のために、ウォッチドッグタイマ(ETH\_DMARSWTRレジスタを参照)が用意されています。このウォッチドッグタイマがゼロでない値でプログラムされると、対応する受信ディスクリプタ(RDES1[31])では有効でないため、RxDMAが受信ステータスをアサートせずに、受信フレームのシステムメモリへの転送を完了すると、ウォッチドッグタイマが

起動します。このタイマがプログラムされた値に従ってタイムアウトになると、RSビットがセットされ、ETH\_DMAIER レジスタで対応する RIE が有効な場合、割り込みがアサートされます。このタイマは、フレームがメモリに転送される時、そのディスクリプタに対して有効にされているために RS がセットされると、タイムアウトになる前に無効にされます。

**注 :** **PMT 制御およびステータスレジスタが読み出されると、ウェイクアップフレーム受信割り込みフラグとマジックパケット受信割り込みフラグが自動的にクリアされます。ただし、これらのフラグのレジスタは CLK\_RX ドメインにあるので、この更新がファームウェアから見えるようになるまでに、かなりの時間がかかることがあります。この遅延は、RX クロックが遅く (10 Mbit モード)、AHB バスが高周波のときに、特に長くなります。**  
**PMT から CPU への割り込みリクエストは CLK\_RX ドメイン内の同じレジスタに基づくので、CPU は PMT\_CSR を読み出した後でも、誤って、割り込みルーチンを再び呼び出すことがあります。したがって、ファームウェアは、ウェイクアップフレーム受信ビットおよび、マジックパケット受信ビットにポーリングして、それらが "0" であることを確認したときのみ、割り込みサービスルーチンから抜ける必要があります。**

## 38.8 イーサネットレジスタの説明

ペリフェラルレジスタには、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位でアクセスできます。

### 38.8.1 MAC レジスタの説明

#### イーサネット MAC 設定レジスタ (ETH\_MACCR)

アドレスオフセット : 0x0000

リセット値 : 0x0000 8000

MAC 設定レジスタは、MAC の動作モードレジスタです。受信および送信動作モードを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	CSTF	予約済み	WD	JD	Res	Res	IFG	CSD	Res	FES	ROD	LM	DM	IPCO	RD	Res	APCS	BL	DC	TE	RE	Res	Res			
						r/w	予約済み	r/w	r/w			r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w			

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **ビット 25 CSTF:**タイプフレーム用 CRC ストリッピング

セットすると、フレームをアプリケーションに転送する前に Ether タイプ (0x0600 より大きいタイプフィールド) の全フレームの最後の 4 バイト (FCS) が排除され、ドロップされます。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **WD :** ウォッチドッグディセーブル

このビットがセットされているときは、MAC は、レシーバのウォッチドッグタイマを無効にし、最大 16,384 バイトまでフレームを受信できます。

このビットがリセットされると、MAC は 2,048 バイト以上のフレームは受信できず、それ以降の受信バイトを切り捨てます。



**ビット 22 JD** : ジャバディセーブル

このビットがセットされているときは、MAC は、トランスミッタのジャバタイマを無効にして、最大 16,384 バイトまでフレームを転送できます。

このビットがリセットされているときには、MAC は、アプリケーションが送信時に 2,048 バイトを超えて送信した場合、トランスミッタをオフにします。

ビット 21:20 予約済みであり、リセット値に保持する必要があります。

**ビット 19:17 IFG** : フレーム間ギャップ

これらのビットは、送信時のフレーム間ギャップの最小時間を制御します。

000 : 96 ビット時間

001 : 88 ビット時間

010 : 80 ビット時間

....

111 : 40 ビット時間

**注 :** 半二重モードでは、最小 IFG は、64 ビット時間 (IFG = 100) についてのみ設定できます。それより小さな値は考慮されません。

**ビット 16 CSD** : キャリヤセンスディセーブル

このビットがハイにセットされると、MAC トランスミッタは、半二重モードでのフレーム送信時に MII/CRS 信号を無視します。そのような送信時には、キャリヤの喪失やキャリヤなしによるエラーは生成されません。

このビットがローにセットされると、MAC トランスミッタは、キャリヤセンスによるエラーを生成し、送信を中止します。

ビット 15 予約済みであり、リセット値に保持する必要があります。

**ビット 14 FES** : ファストイーサネット速度

ファストイーサネット (MII) モードの速度を示します。

0 : 10 Mbit/s

1 : 100 Mbit/s

**ビット 13 ROD** : 自己受信ディセーブル

このビットがセットされた場合、MAC は、半二重モードでのフレームの受信を無効にします。

このビットがリセットされた場合、MAC は、送信中に PHY によって与えられるすべてのパケットを受け取ります。

このビットは、MAC が全二重モードで動作している場合は適用されません。

**ビット 12 LM** : ループバックモード

このビットがセットされた場合、MAC は、MII でループバックモードで動作します。送信クロックは内部でループバックされないため、ループバックが正しく機能するためには、MII 受信クロック入力 (RX\_CLK) が必要です。

**ビット 11 DM** : 全二重モード

このビットがセットされた場合、MAC は全二重モードで動作し、送信と受信を同時に行うことができます。

**ビット 10 IPCO** : IPv4 チェックサムオフロード

セットされた場合、このビットは、IPv4 チェックサムによる受信フレームペイロードの TCP/UDP/ICMP ヘッダのチェックを有効にします。このビットがリセットされた場合、レシーバでのチェックサムオフロード機能は無効にされ、対応する PCE および IP HCE ステータスビット (表 240 (1496 ページ) を参照) は常にクリアされます。

**ビット 9 RD** : リトライディセーブル

このビットがセットされると、MAC は送信を 1 回だけ試みます。MII でコリジョンが発生すると、MAC は現在のフレーム送信を無視して、送信フレームステータスで過剰コリジョンエラーによるフレーム中止を報告します。

このビットがリセットされた場合、MAC は、BL の設定に基づいて再試行を行います。

**注 :** このビットは、半二重モードでのみ適用されます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **APCS** : 自動パッド/CRC ストリッピング

このビットがセットされた場合、MAC は、受信フレームの長さフィールドの値が 1,500 バイト以下の場合のみ、フレームからパッド/FCS フィールドを削除します。長さフィールドが 1,501 バイト以上の受信フレームは、パッド/FCS フィールドを削除せずに、そのままアプリケーションに渡されます。このビットがリセットされると、MAC は、着信するすべてのフレームをそのまま通過させます。

ビット 6:5 **BL** : バックオフリミット

バックオフリミットは、コリジョンの後、再試行時に送信の試みを再スケジュールする前に、MAC が待機するスロット時間遅延 (1000 Mbit/s の場合は 4,096 ビット時間、10/100 Mbit/s の場合は 512 ビット時間) のランダム整数値 ( $r$ ) を指定します。

**注 :** このビットは、半二重モードでのみ適用されます。

00 :  $k = \min(n, 10)$

01 :  $k = \min(n, 8)$

10 :  $k = \min(n, 4)$

11 :  $k = \min(n, 1)$

ここで、 $n$  = 再送信試行。ランダム整数値  $r$  は、 $0 \leq r < 2^k$  の範囲の値を取ります。

ビット 4 **DC** : 遅延チェック

このビットがセットされた場合、MAC の遅延チェック機能が有効になります。MAC は、送信ステートマシンが 10/100 Mbit/s モードで 24,288 ビット時間以上遅延した場合、フレーム中止ステータスを発行して、送信フレームステータスの過剰遅延エラービットをセットします。遅延は、トランスミッタの送信準備ができたときに開始しますが、MII 上に有効な CRS (キャリヤセンス) 信号があるときには開始されません。遅延時間は累積されません。トランスミッタが 10,000 ビット時間遅延した後、送信を行い、コリジョンが発生し、バックオフし、さらにバックオフの完了後に再び遅延した場合、遅延タイムは 0 にリセットされ、再スタートします。

このビットがリセットされた場合、遅延チェック機能は無効であり、MAC は CRS 信号が無効になるまで遅延します。このビットは、半二重モードでのみ適用されます。

ビット 3 **TE** : トランスミッタ有効

このビットがセットされた場合、MAC の送信ステートマシンは、MII 上で送信を行うために有効になります。このビットがリセットされた場合、MAC の送信ステートマシンは、現在のフレームの送信完了後に無効になり、それ以上のフレームは送信されません。

ビット 2 **RE** : レシーバ有効

このビットがセットされた場合、MAC の受信ステートマシンは、MII からフレームを受信するために有効になります。このビットがリセットされた場合、MAC の受信ステートマシンは、現在のフレームの受信完了後に無効になり、MII からそれ以上のフレームを受信しません。

ビット 1:0 予約済みであり、リセット値に保持する必要があります。

## イーサネット MAC フレームフィルタレジスタ (ETH\_MACFFR)

アドレスオフセット : 0x0004

リセット値 : 0x0000 0000

MAC のフレームフィルタレジスタは、フレームを受信するためのフィルタ制御ビットを含みます。このレジスタの制御の一部は、MAC のアドレスチェックブロックに渡されて、アドレスフィルタリングの最初のレベルを実行します。フィルタリングの第 2 のレベルは、不良フレームのパスや制御フレームのパスなど、その他の制御に基づいて、着信フレームに対して実行されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RA	Res.	HPF	SAF	SAIF	PCF	BFD	PAM	DAIF	HM	HU	PM																				
rW																					rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

### ビット 31 RA : すべてを受信

このビットがセットされた場合、MAC レシーバは、アドレスフィルタをパスしたかどうかに関係なく、すべての受信フレームをアプリケーションに渡します。SA/DA フィルタリングの結果は、受信ステータスワードの対応するビットで更新されます。このビットがリセットされた場合、MAC レシーバは、SA/DA アドレスフィルタをパスしたフレームのみをアプリケーションに渡します。

ビット 30:11 予約済みであり、リセット値に保持する必要があります。

### ビット 10 HPF : ハッシュまたは完全フィルタ

このビットがセットされ、HM または HU ビットがセットされている場合、アドレスフィルタは、完全フィルタリングまたはハッシュフィルタリングのいずれかに一致したフレームを通過させます。

このビットがクリアされ、HU または HM ビットがセットされている場合、ハッシュフィルタに一致したフレームのみがアプリケーションに渡されます。

### ビット 9 SAF : ソースアドレスフィルタ

MAC コアは、受信したフレームの SA フィールドと、有効な SA レジスタでプログラムされた値とを比較します。比較結果が一致した場合、RxStatus ワードの SAMatch ビットがハイにセットされます。このビットがハイにセットされ、SA フィルタが失敗した場合、MAC はフレームをドロップします。このビットがリセットされた場合、MAC コアは、受信したフレームをアプリケーションに転送します。また、SA アドレス比較に応じて、RxStatus の更新した SA Match ビットも転送します。

### ビット 8 SAIF : ソースアドレス逆フィルタリング

このビットがセットされた場合、アドレスチェックブロックは、SA アドレス比較の際に逆フィルタリングモードで動作します。SA が SA レジスタと一致するフレームは、SA アドレスフィルタで失敗したのものとしてマークされます。

このビットがリセットされた場合、SA が SA レジスタと一致しないフレームが、SA アドレスフィルタで失敗したのものとしてマークされます。

**ビット 7:6 PCF** : 制御フレームパス

これらのビットは、すべての制御フレーム（ユニキャストおよびマルチキャスト PAUSE フレームを含む）の転送を制御します。PAUSE 制御フレームの処理は、フロー制御レジスタ [2] の RFCE にのみ依存することに注意してください。

00 : MAC は、すべての制御フレームがアプリケーションに届くのを妨げます。

01 : MAC はポーズ制御フレーム以外のすべての制御フレームをアプリケーションへ転送します。

10 : MAC は、アドレスフィルタに失敗した場合もすべての制御フレームをアプリケーションへ転送します。

11 : MAC は、アドレスフィルタをパスする制御フレームを転送します。

これらのビットは、すべての制御フレーム（ユニキャストおよびマルチキャスト PAUSE フレームを含む）の転送を制御します。PAUSE 制御フレームの処理は、フロー制御レジスタ [2] の RFCE にのみ依存することに注意してください。

00 または 01 : MAC は、すべての制御フレームがアプリケーションに届くのを妨げます。

10 : MAC は、アドレスフィルタに失敗した場合もすべての制御フレームをアプリケーションへ転送します。

11 : MAC は、アドレスフィルタをパスする制御フレームを転送します。

**ビット 5 BFD** : ブロードキャストフレームディセーブル

このビットがセットされると、アドレスフィルタは、着信するすべてのブロードキャストフレームをフィルタします。

このビットがリセットされると、アドレスフィルタは、受信したすべてのブロードキャストフレームを通過させます。

**ビット 4 PAM** : すべてのマルチキャストをパス

セットされた場合、このビットは、マルチキャストデスティネーションアドレス（デスティネーションアドレスフィールドの最初のビットが "1"）を持つすべての受信フレームがパスされることを示します。リセットされた場合、マルチキャストフレームのフィルタリングは、HM ビットに依存します。

**ビット 3 DAIF** : デスティネーションアドレス逆フィルタリング

このビットがセットされた場合、アドレスチェックブロックは、ユニキャストおよびマルチキャストフレームの両方について、DA アドレス比較の際に逆フィルタリングモードで動作します。

リセットされた場合、通常のフレームフィルタリングが行われます。

**ビット 2 HM** : ハッシュマルチキャスト

セットされた場合、MAC は、ハッシュテーブルに従って、受信したマルチキャストフレームのデスティネーションアドレスフィルタリングを実行します。

リセットされた場合、MAC は、マルチキャストフレームについて完全デスティネーションアドレスフィルタリングを実行します。すなわち、DA フィールドを DA レジスタでプログラムされた値と比較します。

**ビット 1 HU** : ハッシュユニキャスト

セットされた場合、MAC は、ハッシュテーブルに従って、ユニキャストフレームのデスティネーションアドレスフィルタリングを実行します。

リセットされた場合、MAC は、ユニキャストフレームについて完全デスティネーションアドレスフィルタリングを実行します。すなわち、DA フィールドを DA レジスタでプログラムされた値と比較します。

**ビット 0 PM** : プロミスキュアモード

このビットがセットされると、アドレスフィルタは、デスティネーションまたはソースアドレスに関係なく、着信するすべてのフレームを通過させます。PM がセットされると、受信ステータスワードの SA/DA フィルタ失敗ステータスビットが常にクリアされます。

## イーサネット MAC ハッシュテーブルハイレジスタ (ETH\_MACHTHR)

アドレスオフセット : 0x0008

リセット値 : 0x0000 0000

64 ビットのハッシュテーブルは、グループアドレスフィルタリングに使用されます。ハッシュフィルタリングの場合、着信フレームのデスティネーションアドレスの内容は、CRC ロジックを通してパスされ、CRC レジスタの上位 6 ビットがハッシュテーブルの内容のインデックスに使用されます。この CRC は、次の多項式によって符号化された 32 ビット値です (詳細については、[セクション 38.5.3 : MAC フレームの受信](#)を参照してください)。

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

最上位ビットは、使用されるレジスタ (ハッシュテーブルハイ/ハッシュテーブルロー) を決め、その他の 5 ビットは、レジスタ内のビットを決めます。ハッシュ値 0b0 0000 は、選択されたレジスタのビット 0 を、値 0b1 1111 は、選択されたレジスタのビット 31 を選択します。

たとえば、着信フレームの DA が 0x1F52 419C B6AF として受信された場合 (0x1F は MII インタフェースで受信された最初のバイトです)、内部で計算される 6 ビット、ハッシュ値は 0x2C であり、HTH レジスタのビット [12] がフィルタリングのためにチェックされます。着信フレームの DA が 0xA00A 9800 0045 として受信された場合、計算される 6 ビット、ハッシュ値は 0x07 であり、HTL レジスタのビット [7] がフィルタリングのためにチェックされます。

レジスタの対応するビット値が 1 の場合、フレームは受け入れられます。そうでない場合は、拒否されます。ETH\_MACFFR レジスタの PAM (すべてのマルチキャストをパス) ビットがセットされている場合、マルチキャストハッシュ値に関係なく、すべてのマルチキャストフレームが受け入れられます。

ハッシュテーブルハイレジスタは、マルチキャストハッシュテーブルの上位 32 ビットを含みます。

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

ビット 31:0 HTH : ハッシュテーブルハイ

このフィールドは、ハッシュテーブルの上位 32 ビットを含みます。

## イーサネット MAC ハッシュテーブルローレジスタ (ETH\_MACHTLR)

アドレスオフセット : 0x000C

リセット値 : 0x0000 0000

ハッシュテーブルローレジスタは、マルチキャストハッシュテーブルの下位 32 ビットを含みます。

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

ビット 31:0 HTL : ハッシュテーブルロー

このフィールドは、ハッシュテーブルの下位 32 ビットを含みます。

## イーサネット MAC MII アドレスレジスタ (ETH\_MACMIAR)

アドレスオフセット : 0x0010

リセット値 : 0x0000 0000

MII アドレスレジスタは、管理インターフェースを通じて外部 PHY の管理サイクルを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	PA					MR					Res.	CR			MW	MB																
																r/w		r/w	r/w	r/w	r/w	r/w	rc_w1									

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:11 **PA** : PHY アドレス

このフィールドは、可能な 32 の PHY デバイスのどれがアクセスされているかを示します。

ビット 10:6 **MR** : MII レジスタ

これらのビットは、選択された PHY デバイスの望ましい MII レジスタを選択します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:2 **CR** : クロック範囲

CR クロック範囲選択ビットは、HCLK の周波数を決め、MDC クロックの周波数を決定するために使用されます。

選択HCLK MDC Clock

000 60-100 MHz HCLK/42

001 100-150 MHz HCLK/62

010 20-35 MHz HCLK/16

011 35-60 MHz HCLK/26

100 150-168 MHz HCLK/102

101, 110, 111 予約済み-

ビット 1 **MW** : MII 書き込み

セットされた場合、このビットは、これが MII データレジスタを使用した書き込み動作であることを PHY に知らせます。このビットがセットされていない場合、これは、MII データレジスタにデータを入れる読み出し動作です。

ビット 0 **MB** : MII ビジー

このビットは、ETH\_MACMIAR および ETH\_MACMIIDR に書き込む前は、論理 0 として読み出されます。このビットは、ETH\_MACMIAR への書き込み時に 0 にリセットされる必要があります。PHY レジスタへのアクセス時には、このビットはアプリケーションによって 0b1 にセットされて、読み出された書き込みアクセスが進行中であることを示します。ETH\_MACMIIDR (MII データ) は、このビットが PHY 書き込み動作時に MAC によってクリアされるまで、有効に保たれなければなりません。ETH\_MACMIIDR は、このビットが PHY 読み出し動作時に MAC によってクリアされるまで、無効です。ETH\_MACMIAR (MII アドレス) には、このビットがクリアされるまで書き込むことができません。

## イーサネット MAC MII データレジスタ (ETH\_MACMIIDR)

アドレスオフセット : 0x0014

リセット値 : 0x0000 0000

MAC MII データレジスタは、ETH\_MACMIAR で指定されたアドレスにある PHY レジスタに書き込むデータを格納します。ETH\_MACMIIDR は、ETH\_MACMIAR によって指定されたアドレスにある PHY レジスタから読み出されたデータも格納します。

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 MD : MII データ

これは、管理読み出し動作の後で PHY から読み出された 16 ビットのデータ、または管理書き込み動作の前に PHY に書き込まれる 16 ビットデータを含みます。

## イーサネット MAC フロー制御レジスタ (ETH\_MACFCR)

アドレスオフセット : 0x0018

リセット値 : 0x0000 0000

フロー制御レジスタは、MAC による制御フレーム (ポーズコマンド) の生成と受信を制御します。ビジービットが "1" にセットされているレジスタに書き込むと、MAC はポーズ制御フレームを生成します。制御フレームのフィールドは、802.3x 仕様での規定に従って選択され、このレジスタのポーズ時間の値が制御フレームのポーズ時間フィールドで使用されます。ビジービットは、制御フレームがケーブルに転送されるまでセット状態を維持します。ホストは、レジスタに書き込む前に、ビジービットがクリアされていることを確認する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PT																Res.	ZQPD	Res.	PLT	UPFD	RFCE	TFCE	FCB/B PA								
rW								rW		rW	rW	rW	rW	rW	rc_w1/rw																

ビット 31:16 PT : ポーズ時間

このフィールドは、送信制御フレームのポーズ時間フィールドで使用される値を保持します。ポーズ時間ビットが MII クロックドメインに二重同期されるように設定されている場合、このレジスタへの連続書き込みは、デスティネーションクロックドメインの少なくとも 4 クロックサイクル後のみ実行されなければなりません。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 ZQPD : ゼロクオンタムポーズディセーブル

セットされた場合、このビットは、FIFO レイヤからのフロー制御信号のネゲート時の、ゼロクオンタムポーズ制御フレームの自動生成を無効にします。

このビットがリセットされた場合、ゼロクオンタムポーズ制御フレームの自動生成を伴う通常動作が有効になります。

ビット 6 予約済みであり、リセット値に保持する必要があります。

**ビット 5:4 PLT : ポーズロー閾値**

このフィールドは、ポーズフレームが自動的に再送信されるポーズタイムの閾値を設定します。この閾値は、常に、ビット [31:16] で設定されたポーズ時間未満でなければなりません。たとえば、PT = 100H (256 スロット時間) であり、PLT = 01 の場合、最初のポーズフレームの送信後、228 (256 - 28) スロット時間の時点で、第 2 のポーズフレームが自動的に送信されます。

**選択 閾値**

- 00 ポーズ時間 - 4 スロット時間
- 01 ポーズ時間 - 28 スロット時間
- 10 ポーズ時間 - 144 スロット時間
- 11 ポーズ時間 - 256 スロット時間

スロット時間は、MII インタフェース上で 512 ビット (64 バイト) を送信するための所要時間として定義されます。

**ビット 3 UPFD : ユニキャストポーズフレーム検出**

このビットがセットされた場合、MAC は、一意なマルチキャストアドレスを持つポーズフレームの検出に加えて、ETH\_MACA0HR および ETH\_MACA0LR レジスタで指定されたステーションのユニキャストアドレスを持つポーズフレームも検出します。

このビットがリセットされた場合、MAC は、802.3x 標準で規定されている一意なマルチキャストアドレスを持つポーズフレームのみを検出します。

**ビット 2 RFCE : 受信フロー制御イネーブル**

このビットがセットされた場合、MAC は、受信したポーズフレームをデコードして、指定された時間 (ポーズ時間) の間、トランスミッタを無効にします。

このビットがリセットされた場合、ポーズフレームのデコード機能は無効です。

**ビット 1 TFCE : 送信フロー制御イネーブル**

全二重モードでは、このビットがセットされた場合、MAC は、ポーズフレームを送信するためにフロー制御を有効にします。このビットがリセットされた場合、MAC のフロー制御動作は無効になり、MAC はポーズフレームを送信しません。

半二重モードでは、このビットがセットされた場合、MAC はバックプレッシャ機能を有効にします。このビットがリセットされた場合、バックプレッシャ機能は無効です。

**ビット 0 FCB/BPA : フロー制御ビジー/バックプレッシャ有効**

このビットは、全二重モードで、一ズ制御フレームを開始し、半二重モードでは、TFCE ビットがセットされている場合、バックプレッシャ機能を有効にします。

全二重モードでは、フロー制御レジスタに書き込む前に、このビットが 0 として読み出されなければなりません。ポーズ制御フレームを開始するには、アプリケーションはこのビットを 1 にセットする必要があります。制御フレームの転送時、このビットは、フレーム送信が進行中であることを示すために、セットされたままです。ポーズ制御フレームの送信完了後、MAC はこのビットを 0 にリセットします。フロー制御レジスタは、このビットがクリアされるまで書き込みが禁止されます。

半二重モードでは、このビットがセットされ (TFCE もセットされ) た場合、MAC コアによってバックプレッシャがアサートされます。バックプレッシャ時、MAC が新しいフレームを受信すると、トランスミッタは、JAM パターンの送出を開始し、結果としてコリジョンが発生します。MAC が全二重モードに設定されると、BPA は自動的に無効になります。

## イーサネット MAC VLAN タグレジスタ (ETH\_MAVLANTR)

アドレスオフセット : 0x001C

リセット値 : 0x0000 0000

VLAN タグレジスタは、VLAN フレームを識別する IEEE 802.1Q VLAN タグを含んでいます。MAC は受信フレームの 13 番目と 14 番目のバイト (長さ/タイプ) を 0x8100 と比較して、次の 2 バイトを VLAN タグと比較します。一致した場合は、受信フレームステータスの VLAN ビットがセットされます。フレームの有効長は、1518 バイトから 1522 バイトに増やされます。

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

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **VLANTC** : 12 ビット VLAN タグの比較

このビットがセットされた場合、比較およびフィルタリングには、完全な 16 ビット VLAN タグの代わりに 12 ビット VLAN 識別子が使用されます。VLAN タグのビット [11:0] が、受信した VLAN タグ付きフレーム内の対応するフィールドと比較されます。

このビットがリセットされた場合、受信した VLAN フレームの 15 番目と 16 番目のバイトが比較に使用されます。

ビット 15:0 **VLANTI** : VLAN タグ識別子 (受信フレーム用)

これは、VLAN フレームを識別するための 802.1Q VLAN タグを含み、VLAN フレームのために受信されるフレームの 15 および 16 番目のバイトと比較されます。ビット [15:13] はユーザの優先度であり、ビット [12] は正規フォーマットインジケータ (CFI) であり、ビット [11:0] は VLAN タグの VLAN 識別子 (VID) フィールドです。VLANTC ビットがセットされると、VID (ビット [11:0]) のみが比較に使用されます。

VLANTI (VLANTC がセットされている場合、VLANTI [11:0]) がすべてゼロの場合、MAC は、VLAN タグの比較のために 15 および 16 番目のバイトをチェックせず、タイプフィールドの値が 0x8100 のすべてのフレームを VLAN フレームとして宣言します。

## イーサネット MAC リモートウェイクアップフレームフィルタレジスタ (ETH\_MACRWUFR)

アドレスオフセット : 0x0028

リセット値 : 0x0000 0000

これは、リモートウェイクアップフレームフィルタレジスタがアプリケーションによって読み書きされるアドレスです。ウェイクアップフレームフィルタレジスタは、実際には、そのような (トランスペアレントではない) 8 つのフレームフィルタレジスタのポイントです。オフセット (0x0028) を持つこのレジスタへの 8 回の連続書き込み動作によって、すべてのウェイクアップフレームフィルタレジスタに書き込まれます。オフセット (0x0028) を持つこのレジスタへの 8 回の連続読み出し動作によって、すべてのウェイクアップフレームフィルタレジスタが読み出されます。このレジスタは 7 番目の MAC アドレスの上位 16 ビットを含みます。詳細については、[リモートウェイクアップフレームフィルタレジスタセクション](#)を参照してください。

図 504. イーサネット MAC リモートウェイクアップフレームフィルタレジスタ (ETH\_MACRWUFR)

ウェイクアップフレームフィルタレジスタ 0	フィルタ 0 バイトマスク							
ウェイクアップフレームフィルタレジスタ 1	フィルタ 1 バイトマスク							
ウェイクアップフレームフィルタレジスタ 2	フィルタ 2 バイトマスク							
ウェイクアップフレームフィルタレジスタ 3	フィルタ 3 バイトマスク							
ウェイクアップフレームフィルタレジスタ 4	RSVD	フィルタ 3 コマンド	RSVD	フィルタ 2 コマンド	RSVD	フィルタ 1 コマンド	RSVD	フィルタ 0 コマンド
ウェイクアップフレームフィルタレジスタ 5	フィルタ 3 オフセット		フィルタ 2 オフセット		フィルタ 1 オフセット		フィルタ 0 オフセット	
ウェイクアップフレームフィルタレジスタ 6	フィルタ 1 CRC - 16				フィルタ 0 CRC - 16			
ウェイクアップフレームフィルタレジスタ 7	フィルタ 3 CRC - 16				フィルタ 2 CRC - 16			

ai15648

## イーサネット MAC PMT 制御およびステータスレジスタ (ETH\_MACPMTCSR)

アドレスオフセット : 0x002C

リセット値 : 0x0000 0000

ETH\_MACPMTCSR は、ウェイクアップイベントのリクエストをプログラムし、ウェイクアップイベントを監視します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WFFRPR	Res.	GU	Res.	Res.	WFR	MPR	Res.	Res.	WFE	MPE	PD																				
rs																						rw			rc_r	rc_r			rw	rw	rs

ビット 31 **WFFRPR** : ウェイクアップフレームフィルタレジスタポインタのリセット

セットされた場合、リモートウェイクアップフレームフィルタレジスタのポインタを 0b000 にリセットします。1 クロックサイクル後に自動的にクリアされます。

ビット 30:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **GU** : グローバルユニキャスト

セットされた場合、MAC (DAF) アドレス認識によってフィルタリングされたすべてのユニキャストパケットをウェイクアップフレームとして有効にします。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **WFR** : ウェイクアップフレームを受信

セットされた場合、このビットは、ウェイクアップフレームの受信により、電源管理イベントが生成されたことを示します。このビットは、このレジスタの読み出しによってクリアされます。

ビット 5 **MPR** : マジックパケットを受信

セットされた場合、このビットは、マジックパケットの受信により、電源管理イベントが生成されたことを示します。このビットは、このレジスタの読み出しによってクリアされます。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。



**ビット 2 WFE** : ウェイクアップフレームイネーブル

セットされた場合、このビットは、ウェイクアップフレームの受信による電源管理イベントの生成を有効にします。

**ビット 1 MPE** : マジックパケットイネーブル

セットされた場合、このビットは、マジックパケットの受信による電源管理イベントの生成を有効にします。

**ビット 0 PD** : パワーダウン

このビットがセットされると、すべての受信フレームがドロップされます。このビットは、マジックパケットまたはウェイクアップフレームが受信されると自動的にクリアされ、パワーダウンモードが無効になります。このビットがクリアされた後で受信されたフレームは、アプリケーションに転送されます。このビットは、マジックパケットイネーブルビットまたはウェイクアップフレームイネーブルビットがハイにセットされているときのみ、セットされなければなりません。

## イーサネット MAC デバッグレジスタ (ETH\_MACDBGR)

アドレスオフセット : 0x0034

リセット値 : 0x0000 0000

このデバッグレジスタは、送信および受信データパスと FIFO のすべてのメインモジュールのステータスを提供します。すべてゼロのステータスは、MAC コアがアイドル状態にあり (さらに FIFO がエンブティ) であるため、データパスには動きがないことを表示します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TFF	TFNE	Res.	TFWA	TFRS	MTP	MTFCS	MMTEA	Res.	RFLL	Res.	RFRC	RFWRA	Res.	MSFRWCS	MMRPEA										
						ro	ro		ro	ro	ro	ro	ro	ro	Res.	Res.	Res.	Res.	Res.	Res.	ro	ro		ro	ro	ro		ro	ro	ro	

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

**ビット 25 TFF** : Tx FIFO フル

ハイの場合、Tx FIFO がフルであるため、転送するフレームをさらに受け取れないことを表示します。

**ビット 24 TFNE** : Tx FIFO非エンブティ

ハイの場合、Tx FIFO が非エンブティであるため、転送するデータが残っていることを表示します。

ビット 23 予約済みであり、リセット値に保持する必要があります。

**ビット 22 TFWA** : Tx FIFO 書き込みアクティブ

ハイの場合、Tx FIFO 書き込みコントローラがアクティブであり、データを Tx FIFO へ転送中であることを示します。

**ビット 21:20 TFRS** : Tx FIFO読み出しステータス

Tx FIFO読み出しコントローラの状態を表示します :

00 : アイドル状態

01 : 読み出し状態 (データを MAC トランスミッタへ転送中)

10 : MAC トランスミッタからの Tx ステータス待ち

11 : 受信 Tx ステータスの書き込み中、または Tx FIFO フラッシング中

**ビット 19 MTP** : MAC トランスミッタ停止中

ハイの場合、MAC トランスミッタがポーズ状態 (全二重モードの場合) であるため、転送フレームをスケジュールしないことを表示します。

ビット 18:17 **MTFCS** : MAC送信フレームコントローラステータス

MAC送信フレームコントローラの状態を表示します :

00 : アイドル

01 : 前のフレームのステータスまたは IFG/バックオフ時間の経過待ち

10 : ポーズ制御フレーム (全二重モード) の生成と送信

11 : 送信用に入力フレームを転送

ビット 16 **MMTEA** : MAC MII送信エンジンアクティブ

ハイの場合、MAC MII送信プロトコルエンジンがアクティブにデータを送信しているため、アイドル状態にないことを表示します。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **RFFL** : Rx FIFO フィルレベル

Rx FIFO フィルレベルのステータスを表します :

00 : RxFIFO エンプティ

01 : RxFIFO フィルレベルがフロー制御停止閾値より下

10 : RxFIFO フィルレベルがフロー制御開始閾値より上

11 : RxFIFO フル

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:5 **RFRCS** : Rx FIFO読み出しコントローラステータス

Rx FIFO読み出しコントローラのステータスを表します :

00 : アイドル状態

01 : フレームデータ読み出し中

10 : フレームステータス (またはタイムスタンプ) 読み出し中

11 : フレームデータとステータスをスフラッシュ中

ビット 4 **RFWRA** : Rx FIFO 書き込みコントローラアクティブ

ハイの場合、Rx FIFO 書き込みコントローラがアクティブであり、受信したフレームを FIFO へ転送中であることを表示します。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:1 **MSFRWCS** : MAC スモール FIFO読み出し/書き込みコントローラステータス

ハイの場合、これらのビットは、MAC 受信フレームコントローラモジュールの スモール FIFO読み出しおよび書き込みコントローラのそれぞれのアクティブ状態を表示します。

ビット 0 **MMRPEA** : MAC MII 受信プロトコルエンジンアクティブ

ハイの場合、MAC MII 受信プロトコルエンジンがアクティブにデータを受信しているため、アイドル状態にないことを表示します。

## イーサネット MAC 割り込みステータスレジスタ (ETH\_MACSR)

アドレスオフセット : 0x0038

リセット値 : 0x0000 0000

ETH\_MACSR レジスタの内容は、割り込みを生成できる MAC のイベントを識別します。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSTS	Res.	Res.	MMCTS	MMCRS	MMCS	PMTS	Res.	Res.	Res.
						rc_r			r	r	r	r			

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **TSTS** : タイムスタンプトリガステータス

このビットは、システム時間の値が 目標時間ハイおよびローレジスタで指定された値以上のときに、ハイにセットされます。このビットは、このレジスタの読み出しによってクリアされます。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **MMCTS** : MMC送信ステータス

このビットは、ETH\_MMCTIR レジスタで割り込みが発生されると、セットされます。このビットは、この割り込みレジスタ (ETH\_MMCTIR) のすべてのビットがクリアされたときにリクリアされます。

ビット 5 **MMCRS** : MMC 受信ステータス

このビットは、ETH\_MMCRIR レジスタで割り込みが生成されると、ハイにセットされます。このビットは、この割り込みレジスタ (ETH\_MMCRIR) のすべてのビットがクリアされたときにクリアされます。

ビット 4 **MMCS** : MMC ステータス

このビットは、ビット 6:5 のいずれかがハイにセットされたときに、ハイにセットされます。両方のビットがローのときのみクリアされます。

ビット 3 **PMTS** : PMT ステータス

このビットは、パワーダウンモードで、マジックパケット、または Wake-on-LAN フレームが受信されたときに、セットされます (ETH\_MACPMTCSR レジスタのビット 5 と 6 ([イーサネット MAC PMT 制御およびステータスレジスタ \(ETH\\_MACPMTCSR\) \(1551 ページ\)](#)) を参照)。このビットは、この最後のレジスタのビット [6:5] の両方が ETH\_MACPMTCSR レジスタの読み出し動作のためにクリアされたときにクリアされます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

## イーサネット MAC 割り込みマスクレジスタ (ETH\_MACIMR)

アドレスオフセット : 0x003C

リセット値 : 0x0000 0000

ETH\_MACIMR レジスタのビットによって、ETH\_MACCSR レジスタの対応するイベントによる割り込み信号をマスクできます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSTIM	Res.	Res.	Res.	Res.	Res.	PMTIM	Res.	Res.	Res.
						rw						rw			

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **TSTIM** : タイムスタンプトリガ割り込みマスク

セットされた場合、このビットは、タイムスタンプ割り込みの生成を無効にします。

ビット 8:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **PMTIM** : PMT 割り込みマスク

セットされた場合、このビットは、ETH\_MACCSR の PMT ステータスビットのセットによる割り込み信号のアサートを無効にします。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

## イーサネット MAC アドレス 0 ハイレジスタ (ETH\_MACA0HR)

アドレスオフセット : 0x0040

リセット値 : 0x8000 FFFF

MAC アドレス 0 ハイレジスタは、ステーションの最初の MAC アドレス 6 バイトのうち上位 16 ビットを保持します。MII インタフェース上で受信された最初の DA バイトは、MAC アドレスローレジスタの LS バイト (ビット [7:0]) に対応することに注意してください。たとえば、MII 上でデスティネーションアドレスとして 0x1122 3344 5566 が受信された場合、MAC アドレス 0 レジスタ [47:0] は 0x6655 4433 2211 と比較されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MO	Res.	MACA0H																													
1																rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **MO** : 常に 1。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **MACA0H** : MAC アドレス 0 ハイ [47:32]

このフィールドは、6 バイトの MAC アドレス 0 の上位 16 ビット (47:32) を含みます。これは、受信フレームのフィルタリングのためと、送信フロー制御 (ポーズ) フレームに MAC アドレスを挿入するために、MAC によって使用されます。

## イーサネット MAC アドレス 0 ローレジスタ (ETH\_MACA0LR)

アドレスオフセット : 0x0044

リセット値 : 0xFFFF FFFF

MAC アドレス 0 ローレジスタは、ステーションの最初の MAC アドレス 6 バイトのうち下位 32 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MACA0L																																
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **MACA0L** : MAC アドレス 0 ロー [31:0]

このフィールドは、6 バイトの MAC アドレス 0 の下位 32 ビットを含みます。これは、受信フレームのフィルタリングのためと、送信フロー制御 (ポーズ) フレームに MAC アドレスを挿入するために、MAC によって使用されます。

## イーサネット MAC アドレス 1 ハイレジスタ (ETH\_MACA1HR)

アドレスオフセット : 0x0048

リセット値 : 0x0000 FFFF

MAC アドレス 1 ハイレジスタは、ステーションの 2 番目の MAC アドレス 6 バイトの上位 16 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
AE	SA	MBC								Res.	MACA1H																					
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW								rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	

ビット 31 **AE** : アドレスイネーブル

このビットがセットされると、アドレスフィルタは、完全フィルタリングに MAC アドレス 1 を使用します。このビットがクリアされると、アドレスフィルタは、フィルタリングについてアドレスを無視します。

ビット 30 **SA** : ソースアドレス

このビットがセットされると、受信フレームの SA フィールドとの比較に MAC アドレス 1 [47:0] が使用されます。  
このビットがクリアされると、受信フレームの DA フィールドとの比較に MAC アドレス 1 [47:0] が使用されます。

ビット 29:24 **MBC** : マスクバイト制御

これらのビットは、MAC アドレス 1 の各バイトの比較のためのマスク制御ビットです。これらがハイにセットされると、MAC コアは、受信した DA/SA の対応するバイトと MAC アドレス 1 レジスタの内容とを比較しません。各ビットは、以下のようにバイトのマスキングを制御します。

- ビット 29 : ETH\_MACA1HR [15:8]
- ビット 28 : ETH\_MACA1HR [7:0]
- ビット 27 : ETH\_MACA1LR [31:24]
- ...
- ビット 24 : ETH\_MACA1LR [7:0]

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **MACA1H** : MAC アドレス 1 ハイ [47:32]

このフィールドは、2 番目の MAC アドレスの 6 バイトのうち上位 16 ビット (47:32) を含みます。

## イーサネット MAC アドレス 1 ローレジスタ (ETH\_MACA1LR)

アドレスオフセット : 0x004C

リセット値 : 0xFFFF FFFF

MAC アドレス 1 ローレジスタは、ステーションの 2 番目の MAC アドレス 6 バイトのうち下位 32 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MACA1L																																
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **MACA1L** : MAC アドレス 1 ロー [31:0]

このフィールドは、6 バイトの MAC アドレス 1 の下位 32 ビットを含みます。このフィールドの内容は、初期化プロセスの後、アプリケーションによってロードされるまで未定義です。

## イーサネット MAC アドレス 2 ハイレジスタ (ETH\_MACA2HR)

アドレスオフセット : 0x0050

リセット値 : 0x0000 FFFF

MAC アドレス 2 ハイレジスタは、ステーションの 2 番目の MAC アドレス 6 バイトの上位 16 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AE	SA	MBC								Res.	MACA2H																				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW								rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31 **AE** : アドレスイネーブル

このビットがセットされると、アドレスフィルタは、完全フィルタリングに MAC アドレス 2 を使用します。このビットがリセットされると、アドレスフィルタは、フィルタリングについてアドレスを無視します。

ビット 30 **SA** : ソースアドレス

このビットがセットされると、受信フレームの SA フィールドとの比較に MAC アドレス 2 [47:0] が使用されます。

このビットがクリアされると、受信フレームの DA フィールドとの比較に MAC アドレス 2 [47:0] が使用されます。

ビット 29:24 **MBC** : マスクバイト制御

これらのビットは、MAC アドレス 2 の各バイトの比較のためのマスク制御ビットです。これらのビットがハイにセットされると、MAC コアは、受信した DA/SA の対応するバイトと MAC アドレス 2 レジスタの内容とを比較しません。各ビットは、以下のようにバイトのマスクングを制御します。

- ビット 29 : ETH\_MACA2HR [15:8]
- ビット 28 : ETH\_MACA2HR [7:0]
- ビット 27 : ETH\_MACA2LR [31:24]
- ...
- ビット 24 : ETH\_MACA2LR [7:0]

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

**0MACA2H** : MAC アドレス 2 ハイ [47:32]

ビット 15:0 このフィールドは、MAC アドレス 2 の 6 バイトの上位 16 ビット (47:32) を含みます。



## イーサネット MAC アドレス 2 ローレジスタ (ETH\_MACA2LR)

アドレスオフセット : 0x0054

リセット値 : 0xFFFF FFFF

MAC アドレス 2 ローレジスタは、ステーションの 2 番目の MAC アドレス 6 バイトのうち下位 32 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MACA2L																																
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **MACA2L** : MAC アドレス 2 ロー [31:0]

このフィールドは、6 バイトの MAC アドレス 2 の下位 32 ビットを含みます。このフィールドの内容は、初期化プロセスの後、アプリケーションによってロードされるまで未定義です。

## イーサネット MAC アドレス 3 ハイレジスタ (ETH\_MACA3HR)

アドレスオフセット : 0x0058

リセット値 : 0x0000 FFFF

MAC アドレス 3 ハイレジスタは、ステーションの 2 番目の MAC アドレス 6 バイトの上位 16 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
AE	SA	MBC								Res.	MACA3H																						
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW								rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW		

ビット 31 **AE** : アドレスイネーブル

このビットがセットされると、アドレスフィルタは、完全フィルタリングに MAC アドレス 3 を使用します。このビットがクリアされると、アドレスフィルタは、フィルタリングについてアドレスを無視します。

ビット 30 **SA** : ソースアドレス

このビットがセットされると、受信フレームの SA フィールドとの比較に MAC アドレス 3 [47:0] が使用されます。  
このビットがクリアされると、受信フレームの DA フィールドとの比較に MAC アドレス 3 [47:0] が使用されます。

ビット 29:24 **MBC** : マスクバイト制御

これらのビットは、MAC アドレス 3 の各バイトの比較のためのマスク制御ビットです。これらのビットがハイにセットされると、MAC コアは、受信した DA/SA の対応するバイトと MAC アドレス 3 レジスタの内容とを比較しません。各ビットは、以下のようにバイトのマスキングを制御します。

- ビット 29 : ETH\_MACA3HR [15:8]
- ビット 28 : ETH\_MACA3HR [7:0]
- ビット 27 : ETH\_MACA3LR [31:24]
- ...
- ビット 24 : ETH\_MACA3LR [7:0]

ビット 23:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **MACA3H** : MAC アドレス 3 ハイ [47:32]

このフィールドは、6 バイトの MAC アドレス 3 の上位 16 ビット (47:32) を含みます。

# 参考資料

RM0385

イーサネット (ETH) : メディアアクセス制御 (MAC)、DMA コントローラ付き

## イーサネット MAC アドレス 3 ローレジスタ (ETH\_MACA3LR)

アドレスオフセット : 0x005C

リセット値 : 0xFFFF FFFF

MAC アドレス 3 ローレジスタは、ステーションの 2 番目の MAC アドレス 6 バイトのうち下位 32 ビットを保持します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MACA3L																																
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **MACA3L** : MAC アドレス 3 ロー [31:0]

このフィールドは、6 バイトの MAC アドレス 3 の下位 32 ビットを含みます。このフィールドの内容は、初期化プロセスの後、アプリケーションによってロードされるまで未定義です。



## 38.8.2 MMC レジスタの説明

### イーサネット MMC 制御レジスタ (ETH\_MMCCR)

アドレスオフセット : 0x0100

リセット値 : 0x0000 0000

イーサネット MMC 制御レジスタは、管理カウンタの動作モードを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MCFHP	MCP	MCF	ROR	CSR	CR																									
																										rW	rW	rW	rW	rW	rW

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

#### ビット 5 **MCFHP**:MMC カウンタフルハーフプリセット

MCFHP がローでビット 4 がセットされた場合、すべての MMC カウンタがほぼハーフ値にプリセットされます。また、すべてのフレームカウンタは 0x7FFF\_FFF0 (ハーフ - 16) にプリセットされます。MCFHP がハイでビット 4 がセットされた場合、すべての MMC カウンタがほぼフル値にプリセットされます。また、すべてのフレームカウンタは 0xFFFF\_FFF0 (フル - 16) にプリセットされます。

#### ビット 4 **MCP**:MMC カウンタプリセット

セットされた場合、上記のビット 5 のようにすべてのカウンタがほぼフルまたはほぼハーフに初期化またはプリセットされます。  
このビットは、1 クロックサイクル後に自動的にクリアされます。ビット 5 とともに、このビットは MMC カウンタがハーフフルまたはフルになっているために発生する割り込みのアサーションのデバッグやテストに有用です。

#### ビット 3 **MCF** : MMC カウンタフリーズ

セットされた場合、このビットは、すべての MMC カウンタを現在の値に凍結します。(このビットが 0 にクリアされるまで、どの MMC カウンタも、送信または受信フレームによって更新されません。MMC カウンタのいずれかが読み出し時リセットビットがセットされた状態で読み出された場合、そのカウンタはこのモードでクリアされます。)

#### ビット 2 **ROR** : 読み出し時リセット

このビットがセットされた場合、MMC カウンタは、読み出し後にゼロにリセットされます (リセット後、自動クリア)。カウンタは、最下位バイトレーン (ビット [7:0]) が読み出されると、クリアされます。

#### ビット 1 **CSR** : カウンタストップロールオーバー

このビットがセットされた場合、カウンタは、最大値に達した後もゼロにロールオーバーしません。

#### ビット 0 **CR** : カウンタリセット

セットされた場合、すべてのカウンタがリセットされます。このビットは、1 クロックサイクル後に自動的にクリアされます。

## イーサネット MMC 受信割り込みレジスタ (ETH\_MMCRIR)

アドレスオフセット : 0x0104

リセット値 : 0x0000 0000

イーサネット MMC 受信割り込みレジスタは、受信統計カウンタが最大値の半分に達したときに生成される割り込みを保持します。(カウンタの MSB がセットされます。) 32 ビット幅のレジスタです。割り込みビットは、割り込みの原因となったそれぞれの MMC カウンタが読み出されたときにクリアされます。割り込みビットをクリアするには、それぞれのカウンタの最下位バイトレーン (ビット [7:0]) が読み出されなければなりません。

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

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **RGUFS** : 受信良好ユニキャストフレームステータス

このビットは、受信された良好なユニキャストフレームのカウンタが最大値の半分に達したときにセットされます。

ビット 16:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **RFAES** : 受信フレームアライメントエラーステータス

このビットは、アライメントエラーのある受信フレームのカウンタが最大値の半分に達したときにセットされます。

ビット 5 **RFCES** : 受信フレーム CRC エラーステータス

このビットは、CRC エラーがある受信フレームのカウンタが最大値の半分に達したときにセットされます。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

## イーサネット MMC 送信割り込みレジスタ (ETH\_MMCTIR)

アドレスオフセット : 0x0108

リセット値 : 0x0000 0000

イーサネット MMC 送信割り込みレジスタは、送信統計カウンタが最大値の半分に達したときに生成される割り込みを保持します。(カウンタの MSB がセットされます。) 32 ビット幅のレジスタです。割り込みビットは、割り込みの原因となったそれぞれの MMC カウンタが読み出されたときにクリアされます。割り込みビットをクリアするには、それぞれのカウンタの最下位バイトレーン (ビット [7:0]) が読み出されなければなりません。

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

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **TGFS** : 送信良好フレームステータス

このビットは、送信された良好なフレームのカウンタが最大値の半分に達したときにセットされます。

ビット 20:16 予約済みであり、リセット値に保持する必要があります。



ビット 15 **TGFMSCS** : 複数のコリジョン後に送信された良好フレームのステータス  
このビットは、複数のコリジョン後に送信された良好なフレームのカウンタが最大値の半分に達したときにセットされます。

ビット 14 **TGFSCS** : 1 回のコリジョン後に送信された良好フレームのステータス  
このビットは、1 回のコリジョン後に送信された良好なフレームのカウンタが最大値の半分に達したときにセットされます。

ビット 13:0 予約済みであり、リセット値に保持する必要があります。

## イーサネット MMC 受信割り込みマスクレジスタ (ETH\_MMCRIMR)

アドレスオフセット : 0x010C

リセット値 : 0x0000 0000

イーサネット MMC 受信割り込みマスクレジスタは、受信統計カウンタが最大値の半分に達したときに生成される割り込みのマスクを保持します。(カウンタの MSB がセットされます。) 32 ビット幅のレジスタです。

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

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **RGUFM** : 受信良好ユニキャストフレームマスク  
このビットをセットすると、受信された良好なユニキャストフレームのカウンタが最大値の半分に達したときに割り込みをマスクします。

ビット 16:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **RFAEM** : 受信フレームアライメントエラーマスク  
このビットをセットすると、アライメントエラーがある受信フレームのカウンタが最大値の半分に達したときに割り込みをマスクします。

ビット 5 **RFCEM** : 受信フレーム CRC エラーマスク  
このビットをセットすると、CRC エラーがある受信フレームのカウンタが最大値の半分に達したときに割り込みをマスクします。

ビット 4:0 予約済みであり、リセット値に保持する必要があります。

## イーサネット MMC 送信割り込みマスクレジスタ (ETH\_MMCTIMR)

アドレスオフセット : 0x0110

リセット値 : 0x0000 0000

イーサネット MMC 送信割り込みマスクレジスタは、送信統計カウンタが最大値の半分に達したときに生成される割り込みのマスクを保持します。(カウンタの MSB がセットされます。) 32 ビット幅のレジスタです。

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

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **TGFM** : 送信良好フレームマスク

このビットをセットすると、送信された良好なフレームのカウンタが最大値の半分に達したときに割り込みをマスクします。

ビット 20:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TGFMSCM** : 複数のコリジョン後に送信された良好フレームのマスク

このビットをセットすると、複数のコリジョン後に送信された良好なフレームのカウンタが最大値の半分に達したときに割り込みをマスクします。

ビット 14 **TGFMSCM** : 1 回のコリジョン後に送信された良好フレームのマスク

このビットをセットすると、1 回のコリジョン後に送信された良好なフレームのカウンタが最大値の半分に達したときに割り込みをマスクします。

ビット 13:0 予約済みであり、リセット値に保持する必要があります。

## 1 回のコリジョン後に送信されたイーサネット MMC 良好フレームカウンタレジスタ (ETH\_MMCTGFSCCR)

アドレスオフセット : 0x014C

リセット値 : 0x0000 0000

このレジスタは、半二重モードで 1 回のコリジョン後に正常に送信されたフレームの数を含まれます。

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

ビット 31:0 **TGFSCC** : 1 回のコリジョン後に送信された良好フレームのカウンタ

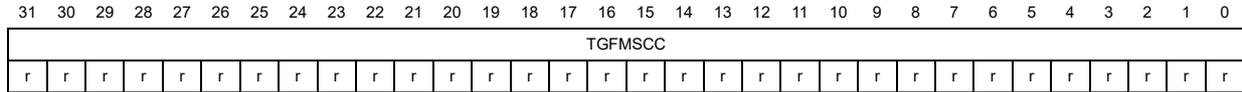
1 回のコリジョン後に送信された良好フレームのカウンタ。

## 複数のコリジョン後に送信されたイーサネット MMC 良好フレームカウンタレジスタ (ETH\_MMCTGFMSCCR)

アドレスオフセット : 0x0150

リセット値 : 0x0000 0000

このレジスタは、半二重モードで複数のコリジョン後に正常に送信されたフレームの数を含まます。



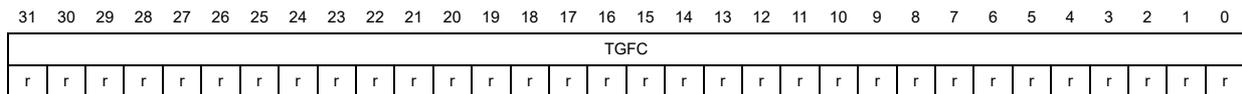
ビット 31:0 **TGFMSCC** : 複数のコリジョン後に送信された良好フレームのカウンタ  
 複数のコリジョン後に送信された良好フレームのカウンタ

## イーサネット MMC 送信良好フレームカウンタレジスタ (ETH\_MMCTGFCR)

アドレスオフセット : 0x0168

リセット値 : 0x0000 0000

このレジスタは、送信された良好フレームの数を含まます。



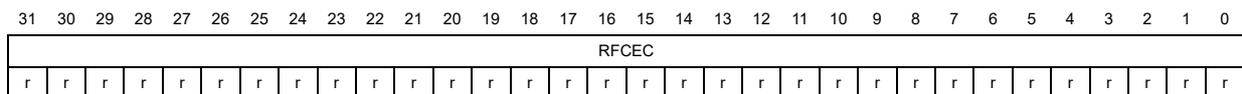
ビット 31:0 **TGFC** : 送信良好フレームカウンタ

## CRC エラーのあるイーサネット MMC 受信フレームカウンタレジスタ (ETH\_MMCRFCECR)

アドレスオフセット : 0x0194

リセット値 : 0x0000 0000

このレジスタは、CRC エラーがある受信フレームの数を含まます。



ビット 31:0 **RFCEC** : 受信フレーム CRC エラーカウンタ  
 CRC エラーのある受信フレームのカウンタ



## 38.8.3 IEEE 1588 タイムスタンプレジスタ

このセクションでは、IEEE 1588 標準に基づいて高精度なネットワーククロック同期機能をサポートするために必要なレジスタについて説明します。

### イーサネット PTP タイムスタンプ制御レジスタ (ETH\_PTPTSCR)

アドレスオフセット : 0x0700

リセット値 : 0x0000 2000

このレジスタは、タイムスタンプの生成および更新ロジックを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TSPFFMAE	TSCNT	TSSMRME	TSSEME	TSSIPV4FE	TSSIPV6FE	TSSPTPOEFE	TSPTPPSV2E	TSSSR	TSSARFE	Res.	Res.	TTSARU	TSITE	TSSTU	TSSTI	TSFCU	TSE													
													rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TSPFFMAE** : タイムスタンプ PTP フレームフィルタリング MAC アドレスイネーブル

セットされている場合、PTP をイーサネットで直接送信する際に、このビットは MAC アドレス (MAC アドレス 0 以外) を使って PTP フレームをフィルタします。

ビット 17:16 **TSCNT** : タイムスタンプクロックノードタイプ

次にクロックノードの使用可能なタイプを示します :

- 00 : 通常クロック
- 01 : 境界クロック
- 10 : エンドツーエンドトランスペアレントクロック
- 11 : ピア・ツー・ピアトランスペアレントクロック

ビット 15 **TSSMRME** : マスタイネーブルに関係するメッセージのタイムスタンプスナップショット

このビットがセットされている場合、マスタノードのみに関係するメッセージのスナップショットがとられます。クリアされている場合は、スレーブノードのみに関係するメッセージのスナップショットがとられます。これは、通常クロックノードと境界クロックノードに対してのみ有効です。

ビット 14 **TSSEME** : イベントメッセージイネーブルのタイムスタンプスナップショット

このビットがセットされている場合、イベントメッセージのみ (SYNC、Delay\_Req、Pdelay\_Req または Pdelay\_Resp) のタイムスタンプスナップショットが取られます。クリアされている場合は、アナウンス、管理、シグナリング以外の全メッセージのスナップショットがとられます。

ビット 13 **TSSIPV4FE** : IPv4 フレームイネーブルのタイムスタンプスナップショット

このビットがセットされている場合、IPv4 フレームのタイムスタンプスナップショットが取られます。

ビット 12 **TSSIPV6FE** : IPv6 フレームイネーブルのタイムスタンプスナップショット

このビットがセットされている場合、IPv6 フレームのタイムスタンプスナップショットが取られます。

ビット 11 **TSSPTPOEFE** : PTP オーバーイーサネットフレームイネーブルのタイムスタンプスナップショット

このビットがセットされている場合、イーサネットフレーム内に PTP メッセージを持つフレーム (PTP オーバーイーサネット) のタイムスタンプスナップショットもとられます。デフォルトでは、UDP-IP イーサネット PTP パケットのスナップショットがとられます。

ビット 10 **TSPTPPSV2E** : バージョン 2 フォーマットイネーブルのタイムスタンプ PTP パケットスヌーピング

このビットがセットされている場合、バージョン 2 フォーマットを使って PTP パケットがスヌープされます。このビットがクリアされている場合、バージョン 1 フォーマットを使って PTP パケットがスヌープされます。

**注 :** IEEE 標準 1588-2008 で指定される IEEE 1588 バージョン 1 およびバージョン 2 フォーマット (レビジョン IEEE STD. 1588-2002)。

ビット 9 **TSSSR** : タイムスタンプサブセカンドロールオーバー : デジタルまたはバイナリロールオーバー制御

このビットがセットされている場合、サブセカンドカウンタが値 0x3B9A C9FF (10 進で 999 999 999) に到達したときタイムスタンプローレジスタがロールオーバーし、タイムスタンプ (ハイ) セカンドをインクリメントさせます。

クリアされると、サブセカンドレジスタのロールオーバー値が 0x7FFF FFFF に到達します。サブセカンドのインクリメントは、PTP の基準クロック周波数とこのビット値に応じて正しく設定される必要があります。

ビット 8 **TSSARFE** : 全受信フレームイネーブルのタイムスタンプスナップショット

このビットがセットされている場合、コアが受信する全フレームに対してタイムスタンプスナップショットが有効化されます。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **TSARU** : タイムスタンプ仮数レジスタ更新

このビットがセットされた場合、タイムスタンプ加数レジスタの内容が、微調整のために PTP ブロックに更新されます。このビットは、更新が完了するとクリアされます。このレジスタビットの読み出し値がゼロでなければ、セットできません。

ビット 4 **TSITE** : タイムスタンプ割り込みトリガイネーブル

このビットがセットされた場合、システム時間が目標時間レジスタに書き込まれた値を超えると、タイムスタンプ割り込みが発生されます。タイムスタンプトリガ割り込みが発生されると、このビットはクリアされます。

ビット 3 **TSSTU** : タイムスタンプシステム時間更新

このビットがセットされた場合、システム時間は、タイムスタンプハイ更新およびタイムスタンプロー更新レジスタで指定された値で更新されます (加算または減算されます)。TSSTU ビットと TSSTI ビットの両方の読み出し値がゼロでなければ、このビットをセットできません。ハードウェアで更新が完了すると、このビットはクリアされます。

ビット 2 **TSSTI** : タイムスタンプシステム時間初期化

このビットがセットされた場合、システム時間は、タイムスタンプハイ更新およびタイムスタンプロー更新レジスタで指定された値で初期化されます (上書きされます)。このビットの読み出し値がゼロでなければ、セットできません。初期化が完了すると、このビットはクリアされます。

ビット 1 **TSFCU** : タイムスタンプ粗密更新

セットされた場合、このビットは、システムタイムスタンプが密更新法を使用して更新されることを示します。クリアされた場合、このビットは、システムタイムスタンプが粗更新法を使用して更新されることを示します。

ビット 0 **TSE** : タイムスタンプイネーブル

このビットがセットされると、送信および受信フレームのタイムスタンプが有効になります。このビットがクリアされると、タイムスタンプ機能は中断され、送信および受信フレームにタイムスタンプが追加されなくなります。維持されていたシステム時間が中断されるので、このビットをハイにセットした後は必ず、タイムスタンプ機能 (システム時間) の初期化を行う必要があります。

下の表に、クロック、イネーブルマスタ、イネーブルスナップショットのイベントメッセージレジスタ設定値に応じてスナップショットをとるメッセージを示します。

表 244. タイムスタンプスナップショットのレジスタビットに対する依存性

TSCNT (ビット 17:16)	TSSMRME (ビット 15) <sup>(1)</sup>	TSSEME (ビット 14)	スナップショットをとるメッセージ
00 または 01	X <sup>(2)</sup>	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
00 または 01	1	1	Delay_Req
00 または 01	0	1	SYNC
10	N/A	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
10	N/A	1	SYNC、Follow_Up
11	N/A	0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp
11	N/A	1	SYNC、Pdelay_Req、Pdelay_Resp

1. N/A = 該当なし。
2. X = don't care.

## イーサネット PTP サブセカンドインクリメントレジスタ (ETH\_PTPSSIR)

アドレスオフセット : 0x0704

リセット値 : 0x0000 0000

このレジスタは、サブセカンドレジスタがインクリメントされる 8 ビット値を含みます。粗更新モードでは (ETH\_PTPTSCR レジスタの TDFCU ビット)、HCLK クロックサイクルごとに、このレジスタの値がシステム時間に加算されます。密更新モードでは、加算器がオーバーフローするたびに、このレジスタの値がシステム時間に加算されます。

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **STSSI** : システム時間サブセカンドインクリメント

このレジスタでプログラムされた値が、更新のたびに、システム時間のサブセカンド値の内容に加算されます。

たとえば、20 ns の精度を得るには、この値は  $20 / 0.467 \approx 43$  (すなわち 0x2A) となります。

## イーサネット PTP タイムスタンプハイレジスタ (ETH\_PTPTSHR)

アドレスオフセット : 0x0708

リセット値 : 0x0000 0000

このレジスタは、上位 32 時間ビットを含みます。この読み出し専用レジスタは、システム時間の秒の値を含んでいます。タイムスタンプハイレジスタは、タイムスタンプローレジスタとともに、MAC によって維持されているシステム時間の現在値を示します。ただし、このレジスタは常時更新されています。

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

ビット 31:0 **STS** : システム時間の秒

このフィールドの値は、コアによって維持されているシステム時間の現在値を秒で示します。

## イーサネット PTP タイムスタンプローレジスタ (ETH\_PTPTSLR)

アドレスオフセット : 0x070C

リセット値 : 0x0000 0000

このレジスタは、下位の 32 時間ビットを含んでいます。この読み出し専用レジスタは、システム時間のサブセカンドの値を含んでいます。

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

ビット 31 **STPNS** : システム時間正または負符号

このビットは、正または負の時間値を示します。セットされた場合、このビットは、時間表示が負であることを示します。クリアされた場合は、時間表示が正であることを示します。システム時間は常に正であるべきなので、このビットは通常ゼロです。

ビット 30:0 **STSS** : システム時間サブセカンド

このフィールドの値は、0.46 ns 精度のサブセカンド表示です。

## イーサネット PTP タイムスタンプハイ更新レジスタ (ETH\_PTPTSHUR)

アドレスオフセット : 0x0710

リセット値 : 0x0000 0000

このレジスタは、システム時間値に書き込まれる時間、または加算あるいは減算される時間の上位 32 ビットを含みます。タイムスタンプハイ更新レジスタは、タイムスタンプ更新ローレジスタとともに、MAC によって維持されているシステム時間を初期化または更新します。これらのレジスタは両方とも、タイムスタンプ制御レジスタの TSSTI または TSSTU ビットをセットする前に書き込む必要があります。

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

ビット 31:0 **TSUS** : タイムスタンプ更新の秒

このフィールドの値は、システム時間が初期化される時間、またはシステム時間に加算される時間を秒で示します。

## イーサネット PTP タイムスタンプロー更新レジスタ (ETH\_PTPTSLUR)

アドレスオフセット : 0x0714

リセット値 : 0x0000 0000

このレジスタは、システム時間値に書き込まれる時間、または加算あるいは減算される時間の下位 32 ビットを含みます。

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

### ビット 31 TSUPNS : タイムスタンプ更新正または負符号

このビットは、正または負の時間値を示します。セットされた場合、このビットは、時間表示が負であることを示します。クリアされた場合は、時間表示が正であることを示します。TSSI がセットされた場合 (システム時間の初期化)、このビットはゼロでなければなりません。TSSTU がセットされているときに、このビットがセットされた場合、タイムスタンプ更新レジスタの値がシステム時間から減算されます。そうでない場合は、システム時間に加算されます。

### ビット 30:0 TSUSS : タイムスタンプ更新サブセカンド

このフィールドの値は、システム時間が初期化されるサブセカンド時間、またはシステム時間に加算されるサブセカンド時間を示します。この値の精度は 0.46 ns です (言い換えると、値 0x0000\_0001 は 0.46 ns です)。

## イーサネット PTP タイムスタンプ加数レジスタ (ETH\_PTPTSAR)

アドレスオフセット : 0x0718

リセット値 : 0x0000 0000

このレジスタは、クロック周波数をマスタクロック周波数に合わせて直線的に再調整するために、ソフトウェアによって使用されます。このレジスタ値は、システム時間が密更新モード (ETH\_PTPTSCR の TSFCU ビット) で設定されているときにのみ使用されます。このレジスタの内容は、クロックサイクルごとに 32 ビットの加算器に加算され、加算器がオーバーフローするたびにシステム時間が更新されます。

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

### ビット 31:0 TSA : タイムスタンプ加数

このレジスタは、時間同期を達成するために、アキュムレータレジスタに加算される 32 ビットの時間値を示します。

## イーサネット PTP 目標時間ハイレジスタ (ETH\_PTPTTHR)

アドレスオフセット : 0x071C

リセット値 : 0x0000 0000

このレジスタは、割り込みイベントの生成のためにシステム時間と比較される時間の上位 32 ビットを含みます。目標時間ハイレジスタは、目標時間ローレジスタとともに、システム時間がこれらのレジスタでプログラムされた値を超えたときに割り込みイベント (ETH\_PTPTSCR の TSARU ビット) をスケジュールするために使用されます。

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

ビット 31:0 **TTSH** : 目標時間スタンプハイ

このレジスタは、時間値を秒単位で格納します。タイムスタンプの値が両方の目標時間スタンプレジスタの値に等しいか、超えたとき、MAC は割り込みを生成します (有効な場合)。

## イーサネット PTP 目標時間ローレジスタ (ETH\_PTPTTLR)

アドレスオフセット : 0x0720

リセット値 : 0x0000 0000

このレジスタは、割り込みイベントの生成のためにシステム時間と比較される時間の下位 32 ビットを含みます。

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

ビット 31:0 **TTSL** : 目標時間スタンプロー

このレジスタは、時間値を (符号付き) ナノ秒単位で格納します。タイムスタンプの値が両方の目標時間スタンプレジスタの値に等しいか、超えたとき、MAC は割り込みを生成します (有効な場合)。



## 38.8.4 DMA レジスタの説明

このセクションでは、各 DMA レジスタのビットを定義します。アドレスがワード整列されている限り、非 32 ビットアクセスが許されます。

### イーサネット DMA バスモードレジスタ (ETH\_DMABMR)

アドレスオフセット : 0x1000

リセット値 : 0x0002 0101

このバスモードレジスタは、DMA のバス動作モードを確立します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	MB	AAB	FPM	USP	RDP				ED	PM	PBL				EDFE	DSL				DA	RS						
					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	rs														

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **MB** : 混合バースト

このビットがハイにセットされ、FB ビットがローの場合、AHB マスタインタフェースは INCR (未定義バースト) で長さが 16 を超えるすべてのバーストを開始します。このビットがクリアされると、バースト長 16 以下の固定のバースト転送 (INCRx および SINGLE) に戻ります。

ビット 25 **AAB** : アドレス整列ビット

このビットがハイにセットされ、FB ビットが 1 の場合、AHB インタフェースは、すべてのバーストを開始アドレスの LS ビットに整列して生成します。FB ビットが 0 に等しい場合、最初のバースト (データバッファの開始アドレスにアクセスする) は整列されませんが、それ以降のバーストはアドレスに整列されます。

ビット 24 **FPM** : 4xPBL モード

セットされた場合、このビットは、プログラムされた PBL 値 (ビット [22:17] およびビット [13:8]) を 4 倍にします。したがって、DMA は、PBL の値に応じて、最大 4、8、16、32、64、および 128 ビートでデータを転送します。

ビット 23 **USP** : 個別 PBL の使用

ハイにセットされた場合、ビット [22:17] で設定された値を PBL として使用するよう RxDMA が設定され、ビット [13:8] の PBL 値は TxDMA の動作にのみ適用されます。このビットがクリアされると、ビット [13:8] の PBL 値は両方の DMA エンジンに対して適用されます。

ビット 22:17 **RDP** : Rx DMA PBL

これらのビットは、1 回の RxDMA トランザクションで転送される最大ビート数を示します。これは、1 回のブロック読み出し/書き込み動作で使用される最大値です。RxDMA は、常に、ホストバス上でバースト転送を開始するたびに、RDP での指定に従ってバーストを試みます。RDP は、1、2、4、8、16、および 32 の許された値でプログラムできます。その他の値は、未定義の動作をもたらします。これらのビットは、USP がハイにセットされた場合にのみ有効であり、適用されます。

ビット 16 **FB** : 固定バースト

このビットは、AHB マスタインタフェースが固定バースト転送を行うかどうかを制御します。このビットがセットされると、AHB は、通常のバースト転送の開始時に、SINGLE、INCR4、INCR8、または INCR16 のみを使用します。このビットがリセットされると、AHB は、SINGLE および INCR バースト転送動作を使用します。



ビット 15:14 **PM** : Rx Tx 優先比率

**RxDMA** リクエストには、**TxDMA** リクエストに対して、以下の比率、先度が与えられます。

00 : 1:1  
01 : 2:1  
10 : 3:1  
11 : 4:1

これは、DA ビットがクリアされているときのみ有効です。

ビット 13:8 **PBL** : プログラマブルバースト長

これらのビットは、1 回の DMA トランザクションで転送される最大ビート数を示します。これは、1 回のブロック読み出し/書き込み動作で使用される最大値です。DMA は、常に、ホストバス上でバースト転送を開始するたびに、PBL での指定に従ってバーストを試みます。PBL は、1、2、4、8、16、および 32 の許された値でプログラムできます。その他の値は、未定義の動作をもたらします。USP がセットされると、この PBL 値は、TxDMA トランザクションにのみ適用されます。

PBL 値には、以下のような制約があります。

- 可能な最大ビート数 (PBL) は、Tx FIFO および Rx FIFO のサイズによって制限されます。
- FIFO には、サポートされる最大ビート数が FIFO の深さの半分であるという制約があります。
- PBL が送信 DMA と受信 DMA の両方に共通の場合、Rx FIFO と Tx FIFO の最小深さを考慮する必要があります。
- 範囲外の PBL 値をプログラムしないでください。システムが正常に動作しなくなる可能性があります。

ビット 7 **EDFE** : 拡張ディスクリプタフォーマット有効

このビットがセットされると、拡張ディスクリプタフォーマットが有効になり、ディスクリプタサイズが 32 バイト (8 個のダブルワード) に増加します。これはタイムスタンプがアクティブになったとき (TSE=1、ETH\_PTPTSCR ビット 0) または IPv4 チェックサムオフロードがアクティブになったとき (IPCO=1、ETH\_MACCCR ビット 10) に必要になります。

ビット 6:2 **DSL** : ディスクリプタスキップ長

このビットは、2 つの連結されていないディスクリプタ間でスキップするワード数を指定します。アドレススキッピングは、現在のディスクリプタの終わりから始まり、次のディスクリプタの始めまでです。DSL の値がゼロの場合、ディスクリプタテーブルは DMA によって連続的である、すなわち、リングモードであると解釈されます。

ビット 1 **DA** : DMA アービトレーション

0 : ビット [15:14] で与えられた Rx と Tx の優先比率によるラウンドロビン方式。  
1 : Rx が Tx に優先します。

ビット 0 **SR** : ソフトウェアリセット

このビットがセットされると、MAC DMA コントローラは、すべての MAC サブシステムの内部レジスタとロジックをリセットします。コアのクロックドメインのすべてでリセット操作が完了した後、自動的にクリアされます。コアのレジスタを再プログラムする前に、このビットの値は 0 でなければなりません。

## イーサネット DMA 送信ポール要求レジスタ (ETH\_DMATPDR)

アドレスオフセット : 0x1004

リセット値 : 0x0000 0000

このレジスタは、送信ディスクリプタリストのポーリングを DMA に指示するために、アプリケーションによって使用されます。送信ポーリング要求レジスタによって、送信 DMA は、現在のディスクリプタが DMA によって所有されているかどうかを確認できます。送信ポーリング要求コマンドは、TxDMA がサスペンドモードの場合に、TxDMA をウェイクアップするために与えられます。TxDMA は、送信フレームにアンダーフローエラーがあった場合、または送信 DMA が所有するディスクリプタが使用不能であった場合、サスペンドモードに移行します。このコマンドはいつでも発行することができ、TxDMA はホストメモリからの現在のディスクリプタの再フェッチを開始すると、リセットします。

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

ビット 31:0 TPD : 送信ポール要求

これらのビットに値が書き込まれると、DMA は、ETH\_DMACHTDR レジスタによって示される現在のディスクリプタを読み出します。そのディスクリプタが使用できない (ホストによって所有されている) 場合、送信はサスペンド状態に戻り、ETH\_DMASR レジスタのビット 2 がアサートされます。ディスクリプタが使用可能な場合は、送信が再開されます。

## イーサネット DMA 受信ポール要求レジスタ (ETH\_DMARPDR)

アドレスオフセット : 0x1008

リセット値 : 0x0000 0000

このレジスタは、受信ディスクリプタリストをポーリングするように DMA に指示するために、アプリケーションによって使用されます。受信ポーリング要求レジスタによって、受信 DMA は新しいディスクリプタを確認できます。このコマンドは、RxDMA をサスペンド状態からウェイクアップするために与えられます。RxDMA は、所有するディスクリプタが使用できない場合にのみ、サスペンドモードに移行することができます。

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

ビット 31:0 RPD : 受信ポール要求

これらのビットに値が書き込まれると、DMA は、ETH\_DMACHRDR レジスタによって示される現在のディスクリプタを読み出します。そのディスクリプタが使用できない (ホストによって所有されている) 場合、受信はサスペンド状態に戻り、ETH\_DMASR レジスタのビット 7 はアサートされません。ディスクリプタが使用可能な場合、受信 DMA は有効状態に戻ります。

## イーサネット DMA 受信ディスクリプタリストアドレスレジスタ (ETH\_DMARDLAR)

アドレスオフセット : 0x100C

リセット値 : 0x0000 0000

受信ディスクリプタリストアドレスレジスタは、受信ディスクリプタリストの開始アドレスを示します。ディスクリプタリストは STM32F75xxx および STM32F74xxx の物理メモリ空間にあり、ワード整列されていなければなりません。DMA は、対応する LS ビットをローにセットすることによって、内部でバス幅で整列されたアドレスに変換します。ETH\_DMARDLAR レジスタへの書き込みは、受信が停止されたときにのみ許されます。受信が停止された場合、受信開始コマンドが与えられる前に、ETH\_DMARDLAR レジスタに書き込まなければなりません。

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

ビット 31:0 SRL : 受信リストの開始

このフィールドは、受信ディスクリプタリストの最初のディスクリプタのベースアドレスを含みます。LSB ビット (32/64/128 ビットバス幅に対して [1/2/3:0] ビット) は、内部的に無視され、DMA によってすべてゼロとして扱われます。したがって、これらの LSB ビットは読み出し専用です。



## イーサネット DMA 送信ディスクリプタリストアドレスレジスタ (ETH\_DMATDLAR)

アドレスオフセット : 0x1010

リセット値 : 0x0000 0000

送信ディスクリプタリストアドレスレジスタは、送信ディスクリプタリストの開始アドレスを示します。ディスクリプタリストは STM32F75xxx および STM32F74xxx の物理メモリ空間にあり、ワード整列されていなければなりません。DMA は、対応する LSB をローにセットすることによって、内部でバス幅で整列されたアドレスに変換します。ETH\_DMATDLAR レジスタへの書き込みは、送信が停止されたときにのみ許されます。送信が停止されると、送信開始コマンドが与えられる前に、ETH\_DMATDLAR レジスタへの書き込みが可能になります。

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

ビット 31:0 **STL** : 送信リストの開始

このフィールドは、送信ディスクリプタリストの最初のディスクリプタのベースアドレスを含みます。LSB ビット (32/64/128 ビットバス幅に対して [1/2/3:0] ビット) は、内部的に無視され、DMAI によってすべてゼロとして扱われます。したがって、これらの LSB ビットは読み出し専用です。

## イーサネット DMA ステータスレジスタ (ETH\_DMASR)

アドレスオフセット : 0x1014

リセット値 : 0x0000 0000

ステータスレジスタは、DMA がアプリケーションに報告するすべてのステータスビットを含んでいます。ETH\_DMASR レジスタは、通常、割り込みサービスルーチンまたはポーリング中にソフトウェアドライバによって読み出されます。このレジスタのフィールドのほとんどは、ホストに割り込みます。ETH\_DMASR レジスタのビットは、読み出されてもクリアされません。ETH\_DMASR レジスタ [16:0] の (予約されていない) ビットに 1 を書き込むと、それらがクリアされますが、0 を書き込んでも効果はありません。各フィールド (ビット [16:0]) は、ETH\_DMAIER レジスタの適切なビットをマスクすることによってマスクできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TSTS	PMTS	MMCS	Res.	EBS			TPS			RPS			NIS	AIS	ERS	FBES	Res.	Res.	ETS	RWTS	RPS	RBUS	RS	TUS	ROS	TJTS	TBUS	TPSS	TS
		r	r	r		r	r	r	r	r	r	r	r	r	rc-w1	rc-w1	rc-w1	rc-w1			rc-w1										

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **TSTS** : タイムスタンプトリガステータス

このビットは、MAC コアのタイムスタンプジェネレータブロックの割り込みイベントを示します。ソフトウェアは、MAC コアのステータスレジスタを読み出して、ソース (ビット 9) をクリアすることによって、このビットを 0 にリセットする必要があります。このビットがハイのときには、割り込みが生成されます (有効な場合)。

ビット 28 **PMTS** : PMT ステータス

このビットは、MAC コアの PMT のイベントを示します。ソフトウェアは、MAC 内の対応するレジスタを読み出して、割り込みの正確な原因を調べ、ソースをクリアすることによって、このビットを 0 にリセットする必要があります。このビットがハイのとき、割り込みが生成されます (有効な場合)。

**ビット 27 MMCS** : MMC ステータス

このビットは、MAC コアの MMC のイベントを反映します。ソフトウェアは、MAC 内の対応するレジスタを読み出して、割り込みの正確な原因を調べ、割り込みのソースをクリアすることによって、このビットを 0 にする必要があります。このビットがハイのとき、割り込みが生成されます (有効な場合)。

ビット 26 予約済みであり、リセット値に保持する必要があります。

**ビット 25:23 EBS** : エラービットステータス

これらのビットは、バスエラー (AHB インタフェース上のエラー応答) の原因となったエラーのタイプを示します。致命的バスエラービット (ETH\_DMASR レジスタ [13]) がセットされたときのみ有効です。このフィールドは割り込みを生成しません。

ビット 231 TxDMA によるデータ転送中のエラー

0 RxDMA によるデータ転送中のエラー

ビット 24 1 読み出し転送中のエラー

0 書き込み転送中のエラー

ビット 25 1 ディスクリプタアクセス中のエラー

0 データバッファアクセス中のエラー

**ビット 22:20 TPS** : 送信プロセス状態

これらのビットは、送信 DMA FSM の状態を示します。このフィールドは割り込みを生成しません。

000 : 停止。送信リセットまたは送信停止コマンドが発行されました。

001 : 実行中。送信転送ディスクリプタをフェッチしています。

010 : 実行中。ステータスを待っています。

011 : 実行中。ホストメモリバッファからデータを読み出して、送信バッファ (TxFIFO) のキューに入れています。

100, 101 : 将来の使用のために予約済み。

110 : 中断。送信ディスクリプタが使用できないか、送信バッファがアンダーフローしています。

111 : 実行中。送信ディスクリプタをクローズしています。

**ビット 19:17 RPS** : 受信処理状態

これらのビットは、受信 DMA FSM の状態を示します。このフィールドは割り込みを生成しません。

000 : 停止。受信リセットまたは受信停止コマンドが発行されました。

001 : 実行中。受信転送ディスクリプタをフェッチしています。

010 : 将来の使用のために予約済み。

011 : 実行中。受信パケットを待っています。

100 : 停止中 : 受信ディスクリプタが使用不可。

101 : 実行中。受信ディスクリプタをクローズしています。

110 : 将来の使用のために予約済み。

111 : 実行中。受信パケットデータを受信バッファからホストメモリへ転送しています。

**ビット 16 NIS** : 通常割り込み要約

正常割り込み要約ビットの値は、ETH\_DMAIER レジスタで対応する割り込みビットが有効なとき、以下の論理 OR です。

- ETH\_DMASR [0] : 送信割り込み

- ETH\_DMASR [2] : 送信バッファ使用不可

- ETH\_DMASR [6] : 受信割り込み

- ETH\_DMASR [14] : アーリー受信割り込み

マスクされていないビットのみが正常割り込み要約ビットに影響します。

これはスティッキービットであり、NIS をセット状態にする対応ビットがクリアされるたびに、クリアされなければなりません (このビットに 1 を書き込むことによって)。

ビット 15 **AIS** : 異常割り込み要約

異常割り込み要約ビットの値は、ETH\_DMAIER レジスタで対応する割り込みビットが有効なとき、以下の論理 OR です。

- ETH\_DMASR [1] : 送信プロセスが停止されました。
- ETH\_DMASR [3] : 送信ジャバタイムアウト
- ETH\_DMASR [4] : 受信 FIFO オーバーフロー
- ETH\_DMASR [5] : 送信アンダーフロー
- ETH\_DMASR [7] : 受信バッファ使用不可
- ETH\_DMASR [8] : 受信処理停止
- ETH\_DMASR [9] : 受信ウォッチドッグタイムアウト
- ETH\_DMASR [10] : アーリー送信割り込み
- ETH\_DMASR [13] : 致命的バスエラー

マスクされていないビットのみが異常割り込み要約ビットに影響します。

これはスティッキービットであり、AIS をセット状態にする対応ビットがクリアされるたびにクリアされなければなりません。

ビット 14 **ERS** : アーリー受信ステータス

このビットは、DMA が最初のデータバッファをパケットで一杯にしたことを示します。受信割り込み ETH\_DMASR [6] は、このビットを自動的にクリアします。

ビット 13 **FBES** : 致命的バスエラーステータス

このビットは、[25:23] で詳細が示されるバスエラーが発生したことを示します。このビットがセットされると、対応する DMA エンジンは、すべてのバスアクセスを無効にします。

ビット 12:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **ETS** : アーリー送信ステータス

このビットは、送信フレームが完全に送信 FIFO に転送されたことを示します。

ビット 9 **RWTS** : 受信ウォッチドッグタイムアウトステータス

このビットは、2,048 バイトより長いフレームが受信されたときにアサートされます。

ビット 8 **RPSS** : 受信処理停止ステータス

このビットは、受信プロセスが停止状態に入ったときにアサートされます。

ビット 7 **RBUS** : 受信バッファ使用不可ステータス

このビットは、受信リスト内の次のディスクリプタがホストによって所有されていて、DMA は取得できないことを示します。受信プロセスは中断されます。受信ディスクリプタの処理を再開するには、ホストは、ディスクリプタの所有権を変更して、受信ポーリング要求コマンドを発行しなければなりません。受信ポーリング要求が発行されなかった場合、受信プロセスは、認識可能な次の着信フレームが受信されたときに再開されます。ETH\_DMASR [7] は、前の受信ディスクリプタが DMA によって所有されていたときのみセットされます。

ビット 6 **RS** : 受信ステータス

このビットは、フレームの受信が完了したことを示します。特定のフレームステータス情報がディスクリプタでポストされました。受信は実行中状態のままです。

ビット 5 **TUS** : 送信アンダーフローステータス

このビットは、フレーム送信中に送信バッファがアンダーフローしたことを示します。送信は中断され、アンダーフローエラー TDES0[1] がセットされます。

ビット 4 **ROS** : 受信オーバーフローステータス

このビットは、フレーム受信中に受信バッファ、オーバーフローしたことを示します。部分的なフレームがアプリケーションに転送された場合、RDES0[11] でオーバーフローステータスがセットされます。

ビット 3 **TJTS** : 送信ジャバタイムアウトステータス

このビットは、送信ジャバタイムアウトが時間切れになったことを示し、トランスミッタが過剰にアクティブであることを意味します。送信プロセスは中止され、停止状態になります。これにより、送信ジャバタイムアウト TDES0[14] フラグがアサートされます。

**ビット 2 TBUS : 送信バッファ使用不可ステータス**

このビットは、送信リスト内の次のディスクリプタがホストによって所有されていて、DMA は取得できないことを示します。送信は中断されます。ビット [22:20] は、送信プロセス状態の遷移の理由を示します。送信ディスクリプタの処理を再開するには、ホストは、ディスクリプタのビットの所有権を変更して、送信ポーリング要求コマンドを発行しなければなりません。

**ビット 1 TPSS : 送信プロセス停止ステータス**

このビットは、送信が停止されたときにセットされます。

**ビット 0 TS : 送信ステータス**

このビットは、フレーム送信が終了し、TDES1[31] が最初のディスクリプタでセットされたことを示します。

## イーサネット DMA 動作モードレジスタ (ETH\_DMAOMR)

アドレスオフセット : 0x1018

リセット値 : 0x0000 0000

動作モードレジスタは、送信および受信動作モードとコマンドを確立します。ETH\_DMAOMR レジスタは、DMA 初期化の一部として書きこまれる最後の CSR でなければなりません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTCEFD	RSF	DFRF	Res.	Res.	TSF	FTF	Res.	Res.	Res.	TTC			ST	Res.	Res.	Res.	Res.	Res.	FEF	FUGF	Res.	RTC		OSF	SR	Res.
					rw	rw	rw			rw	rs				rw	rw	rw	rw						rw	rw		rw	rw	rw	rw	

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

**ビット 26 DTCEFD : TCP/IP チェックサムエラーフレームのドロップングディセーブル**

このビットがセットされた場合、コアは、受信チェックサムオフロードエンジンによって検出されたエラーだけがあるフレームをドロップしません。そのようなフレームは、MAC によって受信され、イーサネットフレームにはエラー (FCS エラーも含む) がなく、カプセル化されたペイロードにのみエラーがあります。このビットがクリアされると、FEF ビットがリセットされていた場合、すべてのエラーフレームがドロップされます。

**ビット 25 RSF : 受信ストアアンドフォワード**

このビットがセットされた場合、完全なフレームが書き込まれた後、Rx FIFO からフレームが読み出され、RTC ビットは無視されます。このビットがクリアされると、Rx FIFO は、RTC ビットで指定された閾値に従って、カットスルーモードで動作します。

**ビット 24 DFRF : 受信フレームフラッシュディセーブル**

このビットがセットされた場合、RxDMA は、受信ディスクリプタ/バッファの使用不可によるフレームのフラッシュを行いません。このビットがクリアされると、通常はフラッシュが行われます。( [中断された受信プロセス \(1530 ページ\)](#) を参照)

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

**ビット 21 TSF : 送信ストアアンドフォワード**

このビットがセットされると、完全なフレームが送信 FIFO に存在しているときに送信が開始されます。このビットがセットされると、ETH\_DMAOMR レジスタのビット [16:14] によって指定された TTC の値は無視されます。

このビットがクリアされると、ETH\_DMAOMR レジスタのビット [16:14] によって指定された TTC の値が考慮されます。

このビットは、送信が停止されたときにのみ変更されなければなりません。



**ビット 20 FTF : フラッシュ送信 FIFO**

このビットがセットされた場合、送信 FIFO コントローラロジックはデフォルト値にリセットされ、TxFIFO 内のすべてのデータが消去されます。このビットは、フラッシュ動作が完了したときに内部的にクリアされます。動作モードレジスタは、このビットがクリアされるまで書き込みが禁止されます。

ビット 19:17 予約済みであり、リセット値に保持する必要があります。

**ビット 16:14 TTC : 送信閾値制御**

この 3 つのビットは、送信 FIFO の閾値レベルを制御します。送信は、送信 FIFO 内のフレームサイズが閾値より大きいときに開始されます。さらに、閾値より短い完全なフレームも送信されます。これらのビットは、TSF ビット (ビット 21) がクリアされたときのみ使用されます。

000 : 64

001 : 128

010 : 192

011 : 256

100 : 40

101 : 32

110 : 24

111 : 16

**ビット 13 ST : 送信開始/停止**

このビットがセットされた場合、送信は実行中状態に置かれ、DMA は、送信フレームの現在位置での送信リストをチェックします。ディスクリプタの取得は、リスト内の現在位置 (ETH\_DMATDLAR レジスタによって設定された送信リストのベースアドレス) から、または、以前に送信が停止されたときに保持された位置から試みられます。現在のディスクリプタが DMA によって所有されていない場合、送信はサスペンド状態に入り、送信バッファ使用不可ビット (ETH\_DMASR [2]) がセットされます。送信開始コマンドは、送信が停止されたときのみ有効となります。DMA ETH\_DMATDLAR レジスタをセットする前にコマンドが発行された場合、DMA の動作は予想できません。

このビットがクリアされると、送信プロセスは現在のフレームの送信を終了した後、停止状態に置かれます。送信リスト内の次のディスクリプタの位置が保存され、送信がリスタートされるときに現在位置となります。送信停止コマンドは、現在のフレームの送信が完了したとき、または送信がサスペンド状態にあるときのみ有効です。

ビット 12:8 予約済みであり、リセット値に保持する必要があります。

**ビット 7 FEF : エラーフレーム転送**

このビットがセットされた場合、ラントエラーフレームを除くすべてのフレームが DMA に転送されます。

このビットがクリアされると、Rx FIFO は、エラーステータス (CRC エラー、コリジョンエラー、ジャイアントフレーム、ウォッチドッグタイムアウト、オーバーフロー) のフレームをドロップします。ただし、フレームの開始バイト (書き込み) ポインタが読み出しコントローラ側にすでに転送されている場合 (閾値モードで)、フレームはドロップされません。Rx FIFO は、そのフレームの開始バイトが ARI バスで転送 (出力) されていない場合、エラーフレームをドロップします。

**ビット 6 FUGF : アンダーサイズの良いフレームの転送**

このビットがセットされた場合、Rx FIFO は、アンダーサイズのフレーム (エラーがなく、パッドバイトと CRC を含めて長さが 64 バイト未満のフレーム) を転送します。

このビットがクリアされると、Rx FIFO は、受信閾値より低い値であるために (RTC = 01 など)、フレームがすでに転送されていない限り、64 バイト未満のすべてのフレームをドロップします。

ビット 5 予約済みであり、リセット値に保持する必要があります。

**ビット 4:3 RTC : 受信閾値制御**

この 2 つのビットは、受信 FIFO の閾値レベルを制御します。DMA への転送 (リクエスト) は、受信 FIFO 内のフレームサイズが閾値より大きいときに開始されます。さらに、長さが閾値より短い完全なフレームは自動的に転送されます。

**注 :** 設定された受信 FIFO のサイズが 128 バイトの場合、値 11 は適用されないことに注意してください。

**注 :** これらのビットは、RSF ビットがゼロのときにのみ有効であり、RSF ビットが 1 にセットされたときは無視されます。

00 : 64  
01 : 32  
10 : 96  
11 : 128

**ビット 2 OSF : 2 番目のフレームに対する動作**

このビットがセットされた場合、このビットは、最初のフレームのステータスが取得される前でも、DMA に送信データの 2 番目のフレームを処理するように指示します。

**ビット 1 SR : 受信開始/停止**

このビットがセットされると、受信プロセスは実行中状態に置かれます。DMA は、受信リストからのディスクリプタの取得を試みて、受信フレームを処理します。ディスクリプタの取得は、リスト内の現在位置 (ETH\_DMATDLAR レジスタによって設定されたアドレス) から、または、以前に受信プロセスが停止されたときに保持された位置から試みられます。DMA によって所有されているディスクリプタがない場合、受信は中断され、受信バッファ使用不可ビット (ETH\_DMASR [7]) がセットされます。受信開始コマンドは、受信が停止されているときにのみ有効です。DMA ETH\_DMARDLAR レジスタをセットする前にコマンドが発行された場合、DMA の動作は予想できません。

このビットがクリアされると、RxDMA の動作は、現在のフレームの転送後に停止されます。受信リスト内の次のディスクリプタの位置が保存され、受信プロセスがリスタートされるときに現在位置となります。受信停止コマンドは、受信プロセスが実行中 (受信パケットを待機中) またはサスペンド状態にあるときにのみ有効です。

ビット 0 予約済みであり、リセット値に保持する必要があります。

## イーサネット DMA 割り込み有効レジスタ (ETH\_DMAIER)

アドレスオフセット : 0x101C

リセット値 : 0x0000 0000

割り込み有効レジスタは、ETH\_DMASR によって報告された割り込みを有効にします。ビットを 1 にセットすると、対応する割り込みが有効になります。すべての割り込みは、ハードウェアまたはソフトウェアリセット後に無効になります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NISE	AISE	ERIE	FBEIE	Res.	Res.	ETIE	RWTIE	RPSIE	RBUIE	RIE	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE														
															rW	rW	rW	rW			rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **NISE** : 通常割り込み要約イネーブル

このビットがセットされた場合、正常割り込みが有効になります。このビットがクリアされた場合、正常割り込みは無効です。このビットは、以下のビットを有効にします。

- ETH\_DMASR [0] : 送信割り込み
- ETH\_DMASR [2] : 送信バッファ使用不可
- ETH\_DMASR [6] : 受信割り込み
- ETH\_DMASR [14] : アーリー受信割り込み

ビット 15 **AISE** : 異常割り込み要約イネーブル

このビットがセットされると、異常割り込みが有効にされます。このビットがクリアされると、異常割り込みは無効です。このビットは、以下のビットを有効にします。

- ETH\_DMASR [1] : 送信プロセス停止
- ETH\_DMASR [3] : 送信ジャバタイムアウト
- ETH\_DMASR [4] : 受信オーバーフロー
- ETH\_DMASR [5] : 送信アンダーフロー
- ETH\_DMASR [7] : 受信バッファ使用不可
- ETH\_DMASR [8] : 受信処理停止
- ETH\_DMASR [9] : 受信ウォッチドッグタイムアウト
- ETH\_DMASR [10] : アーリー送信割り込み
- ETH\_DMASR [13] : 致命的バスエラー

ビット 14 **ERIE** : アーリー受信割り込みイネーブル

このビットが、正常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [16]) とともにセットされると、アーリー受信割り込みが有効になります。

このビットがクリアされると、アーリー受信割り込みは無効です。

ビット 13 **FBEIE** : 致命的バスエラー割り込みイネーブル

このビットが、異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされると、致命的バスエラー割り込みが有効になります。

このビットがクリアされると、致命的バスエラー割り込みは無効です。

ビット 12:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **ETIE** : アーリー送信割り込みイネーブル

このビットが、異常割り込み要約イネーブルビット (ETH\_DMAIER register[15]) とともにセットされると、アーリー送信割り込みが有効になります。

このビットがクリアされると、アーリー送信割り込みは無効です。

- ビット 9 RWTIE** : 受信ウォッチドッグタイムアウト割り込み有効  
このビットが、異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされると、受信ウォッチドッグタイムアウト割り込みが有効になります。  
このビットがクリアされると、受信ウォッチドッグタイムアウト割り込みは無効です。
- ビット 8 RPSIE** : 受信処理停止割り込みイネーブル  
このビットが、異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされると、受信プロセス停止割り込みが有効になります。このビットがクリアされると、受信プロセス停止割り込みは無効です。
- ビット 7 RBUIE** : 受信バッファ使用不可割り込みイネーブル  
このビットが、異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされた場合、受信バッファ使用不可割り込みが有効になります。  
このビットがクリアされると、受信バッファ使用不可割り込みは無効です。
- ビット 6 RIE** : 受信割り込みイネーブル  
このビットが正常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [16]) とともにセットされた場合、受信割り込みが有効になります。  
このビットがクリアされると、受信割り込みは無効です。
- ビット 5 TUIE** : アンダーフロー割り込みイネーブル  
このビットが異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされた場合、送信アンダーフロー割り込みが有効になります。  
このビットがクリアされると、アンダーフロー割り込みは無効です。
- ビット 4 ROIE** : オーバーフロー割り込みイネーブル  
このビットが異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされた場合、受信オーバーフロー割り込みが有効になります。  
このビットがクリアされると、オーバーフロー割り込みは無効です。
- ビット 3 TJTIE** : 送信ジャバタイムアウト割り込み有効  
このビットが異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされた場合、送信ジャバタイムアウト割り込みが有効になります。  
このビットがクリアされると、送信ジャバタイムアウト割り込みは無効です。
- ビット 2 TBUIE** : 送信バッファ使用不可割り込みイネーブル  
このビットが正常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [16]) とともにセットされた場合、送信バッファ使用不可割り込みが有効になります。  
このビットがクリアされると、送信バッファ使用不可割り込みは無効です。
- ビット 1 TPSIE** : 送信プロセス停止割り込みイネーブル  
このビットが異常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [15]) とともにセットされた場合、送信プロセス停止割り込みが有効になります。  
このビットがクリアされると、送信停止割り込みは無効です。
- ビット 0 TIE** : 送信割り込みイネーブル  
このビットが正常割り込み要約イネーブルビット (ETH\_DMAIER レジスタ [16]) とともにセットされた場合、送信割り込みが有効になります。  
このビットがクリアされると、送信割り込みは無効です。

イーサネット割り込みは、DMA ステータスレジスタの TSTS または PMTS ビットがアサートされ、対応する割り込みのマスクが解除されたとき、または、NIS/AIS ステータスビットがアサートされ、対応する割り込みイネーブルビット (NISE/AISE) が有効にされたときにのみ生成されます。





## イーサネット DMA 現在のホスト受信バッファアドレスレジスタ (ETH\_DMACHRBAR)

アドレスオフセット : 0x1054

リセット値 : 0x0000 0000

現在のホスト受信バッファアドレスレジスタは、DMA によって読み出されている現在の受信バッファを示します。

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

ビット 31:0 **HRBAP** : ホスト受信バッファアドレスポインタ

リセット時にクリアされます。ポインタは動作中に DMA によって更新されます。

### 38.8.5 イーサネットレジスタマップ

表 245 に、ETH レジスタマップとリセット値を示します。

表 245. イーサネットレジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	ETH_MACCCR	Res	Res	Res	Res	Res	Res	CSTF	Res	WD	JD	Res	Res	IFG	Res	CSD	Res	FES	ROD	LM	DM	IPCO	RD	Res	APCS	BL	DC	TE	RE	Res	Res						
	リセット値							0		0	0			0	0	0			0	0	0	0	0	0		0	0	0	0	0	0	0	0				
0x04	ETH_MACFFR	RA	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	HPF	SAF	Res	PCF	BFD	PAM	DAIF	HM	HU	PM					
	リセット値	0																					0	0	0	0	0	0	0	0	0	0	0				
0x08	ETH_MACHTHR	HTH[31:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x0C	ETH_MACHTLR	HTL[31:0]																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	ETH_MACMIAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res				
	リセット値																																				
0x14	ETH_MACMIADR	MD																																			
	リセット値																																				
0x18	ETH_MACFCR	PT															Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ZQPD	Res	PLT	UPFD	RFCE	TFCE	FCB/BPA
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																				
0x1C	ETH_MACVLANTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																																				
0x28	ETH_MACRWUFR	Frame filter reg0\Frame filter reg1\Frame filter reg2\Frame filter reg3\Frame filter reg4...\Frame filter reg7																																			
	リセット値	0																																			
0x2C	ETH_MACPMTCSR	WFERPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値	0																																			
0x34	ETH_MACDBGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res					
	リセット値																																				

表 245. イーサネットレジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x38	ETH_MACSR																	Res.	Res.	Res.	Res.	Res.	Res.	TSTS	Res.	Res.	Res.	MMCTS	MMCRS	MMCS	PMTS	Res.	Res.	Res.			
	リセット値																								0			0	0	0	0						
0x3C	ETH_MACIMR																	Res.	Res.	Res.	Res.	Res.	Res.	TSTIM	Res.	Res.	Res.	Res.	Res.	PMTIM	Res.	Res.	Res.				
	リセット値																							0						0							
0x40	ETH_MACA0HR	MO	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACA0H																			
	リセット値	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
0x44	ETH_MACA0LR	MACA0L																																			
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x48	ETH_MACA1HR	AE	SA	MBC[6:0]															MACA1H																		
	リセット値	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x4C	ETH_MACA1LR	MACA1L																																			
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
0x50	ETH_MACA2HR	AE	SA	MBC															MACA2H																		
	リセット値	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x54	ETH_MACA2LR	MACA2L																																			
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
0x58	ETH_MACA3HR	AE	SA	MBC															MACA3H																		
	リセット値	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
0x5C	ETH_MACA3LR	MACA3L																																			
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
0x100	ETH_MMCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCFHP	MCP	MCF	ROR	CSR	CR				
	リセット値																											0	0	0	0	0	0				
0x104	ETH_MMCRIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RGUFS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFAES	RFCES	Res.	Res.	Res.	Res.				
	リセット値																0											0	0								
0x108	ETH_MMCTIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.					
	リセット値																																				
0x10C	ETH_MMCRIMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RGUFM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFAEM	RFCEM	Res.	Res.	Res.	Res.				
	リセット値																0											0	0								
0x110	ETH_MMCTIMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGFM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.					
	リセット値																0																				
0x14C	ETH_MMCTGFSCCR	TGFSCC																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x150	ETH_MMCTGFMSCR	TGFMSCC																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x168	ETH_MMCTGFSCR	TGFC																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x194	ETH_MMCRFCECR	RFCEC																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x198	ETH_MMCRFAECR	RFAEC																																			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				



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

表 245. イーサネットレジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C4	ETH_MMCRGUFCR	RGUFC																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x700	ETH_PTPTSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSPFMAE	TSCNT	TSSMRME	TSSEME	TSSIPV4FE	TSSIPV6FE	TSSPTPOEFE	TSPTPPSV2E	TSSSR	TSSARFE	Res.	Res.	TTSARU	TSITE	TSSTU	TSSTI	TSFCU	TSE	
	リセット値														0	0	0	0	0	1	0	0	0	0				0	0	0	0	0	0
0x704	ETH_PTSSIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x708	ETH_PTPTSHR	STS[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70C	ETH_PTPTSLR	STPNS	STSS																														
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x710	ETH_PTPTSHUR	TSUS																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x714	ETH_PTPTSLUR	TSUPNS	TSUSS																														
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x718	ETH_PTPTSAR	TSA																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x71C	ETH_PTPTTHR	TTSH																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x720	ETH_PTPTTLR	TTSL																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x728	ETH_PTPTSSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																
0x1000	ETH_DMABMR	Res.	Res.	Res.	Res.	Res.	MB	AAB	FPM	USP	RDP				FB	PM	PBL				EDFE	DSL				DA	SR						
	リセット値						0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0x1004	ETH_DMATPDR	TPD																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1008	ETH_DMARPDR	RPD																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x100C	ETH_DMARDLAR	SRL																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1010	ETH_DMATDLAR	STL																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1014	ETH_DMASR	Res.	Res.	TSTS	PMTS	MMCS	Res.	EBS	TPS				RPS	NIS		AI5	ERS	FBES	Res.	Res.	ETS	RWTS	RPSS	RBUS	RS	TUS	ROS	TJTS	TBUS	TPSS	TS		
	リセット値			0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1018	ETH_DMAOMR	Res.	Res.	Res.	Res.	Res.	DTCEFD	RSF	DFRF	Res.	Res.	TSF	FTF	Res.	Res.	Res.	TTC		ST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値						0	0	0			0	0				0	0	0	0													
0x101C	ETH_DMAIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NISE	AISE	ERIE	FBEIE	Res.	Res.	ETIE	RWIE	RP5IE	RBUIE	RIE	TUIE	ROIIE	TJUIE	TBUIE	TP5IE	TIE
	リセット値																0	0	0	0			0	0	0	0	0	0	0	0	0	0	0
0x1020	ETH_DMAMFBOCR	Res.	Res.	Res.	OFOC	MFA										OMFC	MFC																
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 245. イーサネットレジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x1024	ETH_DMARSWTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSWTC													
	リセット値																										0	0	0	0	0	0	0	0					
0x1048	ETH_DMACHTDR	HTDAP																																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x104C	ETH_DMACHRDR	HRDAP																																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1050	ETH_DMACHTBAR	HTBAP																																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1054	ETH_DMACHRBAR	HRBAP																																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 39 HDMI-CEC コントローラ (HDMI-CEC)

### 39.1 概要

Consumer Electronics Control (CEC) は HDMI (高解像度マルチメディアインタフェース) 規格の一部です (付録 1)。

ユーザ環境ですべての多様な視聴覚製品に高品質な制御機能を提供するプロトコルで構成されています。ロースピードでの最小限の処理とメモリのオーバヘッドで、動作させるよう策定されています。

HDMI-CEC コントローラでは、このプロトコルに対するハードウェアサポートを提供しています。

### 39.2 HDMI-CEC コントローラの主な機能

- HDMI-CEC v1.4 仕様に準拠
- 2つのクロックソースオプションを用いた 32 kHz CEC カーネル
  - 固定プリスケラを用いた HSI RC オシレータ (HSI/488)
  - LSE オシレータ
- STOP モードの超低電力アプリケーションで動作
- 送信開始前に設定可能な信号フリータイム
  - CEC 状態と送信履歴に従って、ハードウェアによって自動的に実行される
  - ソフトウェアによって固定化 (7つのタイミングオプション)
- 設定可能なペリフェラルアドレス (OAR)
- リスンモードをサポート
  - CEC ラインに干渉することなく、OAR とは異なる転送先アドレスに送信された CEC メッセージの受信を有効化
- 設定可能な Rx 許容誤差マージン
  - 標準の許容誤差
  - 拡張された許容誤差
- 受信 – エラー検出
  - ビット立ち上がりエラー (BRE)、オプションによる受信の停止 (BRESTP)
  - ショートビット周期エラー (SBPE)
  - ロングビット周期エラー (LBPE)
- 設定可能なエラービット生成
  - BRE 検出 (BREGEN)
  - LBPE 検出 (LBPEGEN)
  - 常に SBPE 検出で生成される
- 送信エラー検出 (TXERR)
- アービトレーション喪失検出 (ARBLST)
  - 自動送信を再試行
- 送信アンダーラン検出 (TXUDR)
- 受信オーバーラン検出 (RXOVR)

## 39.3 HDMI-CEC の機能詳細

### 39.3.1 HDMI-CEC ピン

CEC バスは、デバイスからデータを送受信するために使用される 1 本の双方向ラインで構成されています。このバスは、27 kΩ プルアップレジスタを介して、+3.3 V 電源電圧に接続されています。ワイヤード AND 接続を使用するには、デバイスの出力段はオープンドレインまたはオープンコレクタである必要があります。

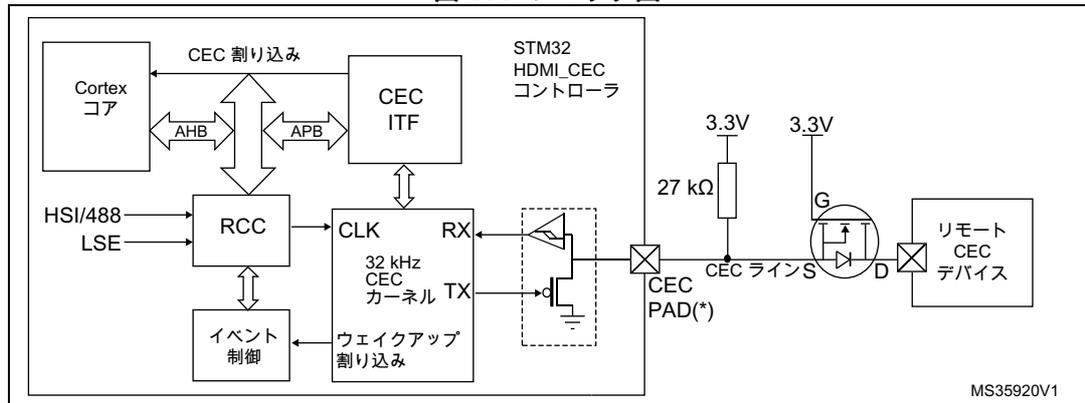
HDMI-CEC コントローラでは、CEC 双方向ラインをオルタネート機能オープンドレインとして設定されたものとみなし、標準 GPIO のオルタネート機能として管理します。27 kΩ プルアップを外部的に STM32 に追加する必要があります。

アプリケーションの電源をオフにした際に CEC バスと干渉しないよう、このような条件ではバスから CEC ピンを離す必要があります。これは、[図 505](#) に示すような MOS トランジスタを使用して実行できます。

表 246. HDMI ピン

名前	信号タイプ	説明
CEC	双方向	2つの状態： 1 = ハイインピーダンス 0 = ローインピーダンス 27 kΩ を外部的に追加する必要があります。

図 505. ブロック図



- 出力オープンドレインのオルタネート機能として設定された GPIO。
- 出力オープンドレインのオルタネート機能として設定される場合、シュミットトリガは有効なままとなります。

## 39.3.2 メッセージの説明

CEC ライン上のすべてのトランザクションは、イニシエータと 1 つまたは複数のフォローで構成されています。イニシエータは、メッセージの構造とデータを送信します。フォローは、すべてのデータを受信し、すべての確認応答ビットを設定します。

メッセージは単一フレームで送信されます。このフレームは、開始ビットとそれに続くヘッダブロックによって構成され、これにオプションとして OP コードとオペランドブロックの変数を含めることもできます。

これらのすべてのブロックは 8 ビットペイロードによって作成され (最上位ビットが最初に送信される)、続けてメッセージの終了 (EOM) ビット、および確認応答 (ACK) ビットによって作成されます。

EOM ビットはメッセージの最後のブロックにセットされ、その他のブロックではリセットされます。EOM が示された後でメッセージに追加のブロックが含まれている場合、このブロックは無視されません。EOM ビットは、他のデバイスがアクティブであることを確認するために、ヘッダブロックにセットして他のデバイスを「ピング」することができます。

確認応答ビットは、イニシエータによって常にハイインピーダンスにセットされています。これは、ヘッダの自己アドレスを読み出したフォロー、またはブロードキャストメッセージを拒否する必要があるフォローが、ローに駆動できるようにするためです。

ヘッダは、ソース論理アドレスフィールドおよび転送先論理アドレスフィールドで構成されています。ブロードキャストメッセージには、特殊アドレス 0xF が使用されている点に注意してください。

図 506. メッセージの構造

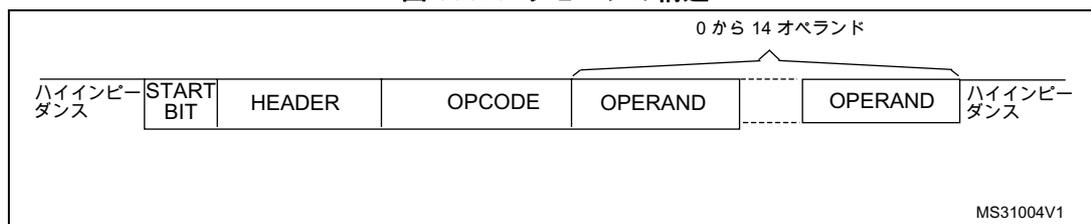
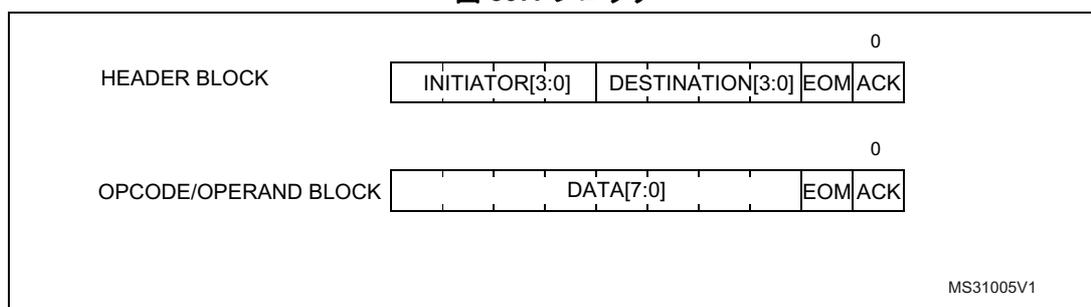


図 507. ブロック

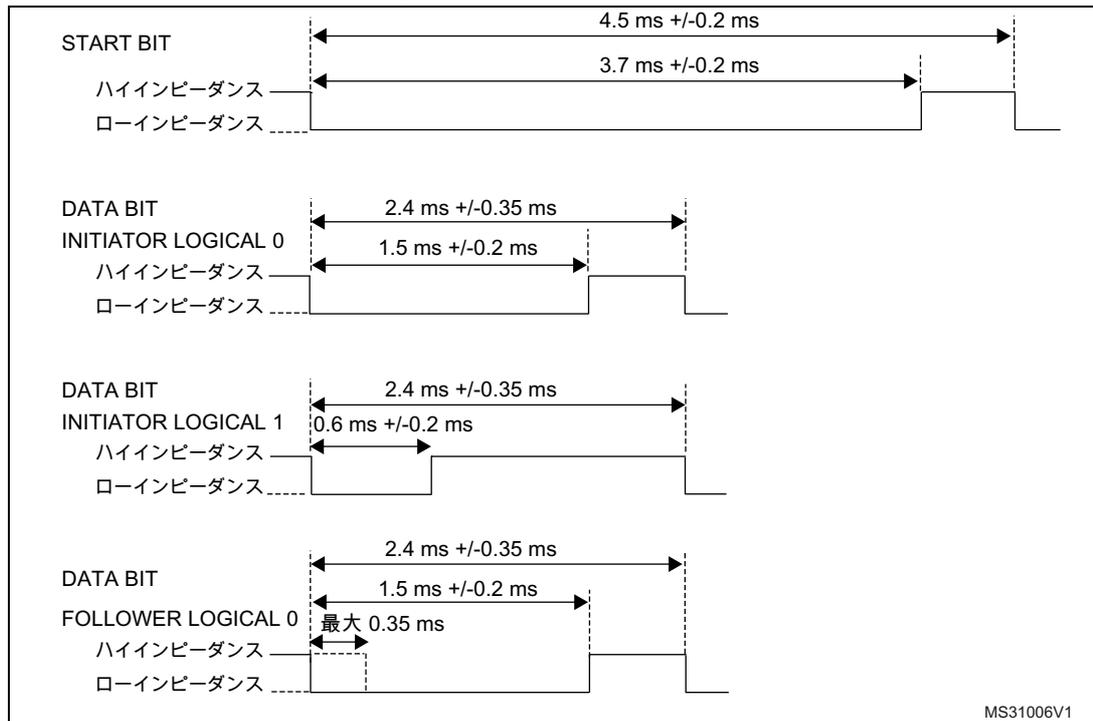


### 39.3.3 ビットタイミング

開始ビットの形式は一意で、メッセージの始まりを識別します。ロー時間および合計時間で検証されます。

メッセージの開始ビット後の残りのすべてのデータビットのタイミングは一定です。データビットの終わりのハイからローへの遷移は、CEC ラインがハイのままとなる最終ビットを除き、次のデータビットの始まりです。

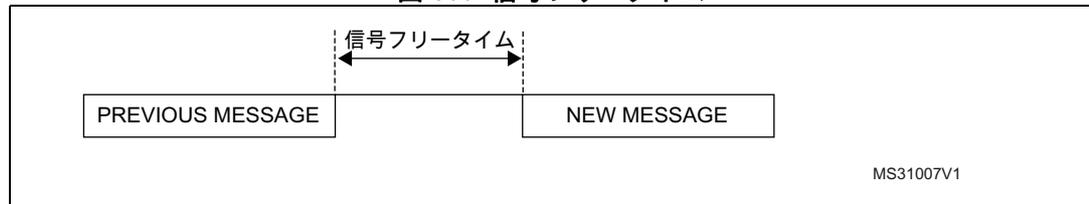
図 508. ビットタイミング



## 39.4 アービトレーション

CEC ラインにメッセージを送信または再送信する必要のあるすべてのデバイスは、いくつかのビット周期において無効であったことを確認する必要があります。信号フリータイムは、前のフレームの最終ビットからの開始時間として定義され、開始するデバイスと次の図に示す現在のステータスに依存します。

図 509. 信号フリータイム



1 度に使用できるイニシエータは 1 つのみであるため、同時に複数のイニシエータが送信を開始する場合に競合を避けるために、アービトレーション機構が提供されています。

CEC ラインアービトレーションは、開始ビットの先端から開始し、ヘッダブロック内のイニシエータアドレスビットの最後まで続けられます。この周期中、イニシエータは CEC ラインを監視し、このラインをハイインピーダンス状態に駆動する際に 0 に読み戻すと、アービトレーションを喪失したとみなして送信を停止し、フォロワになります。

図 510. アービトレーションフェーズ

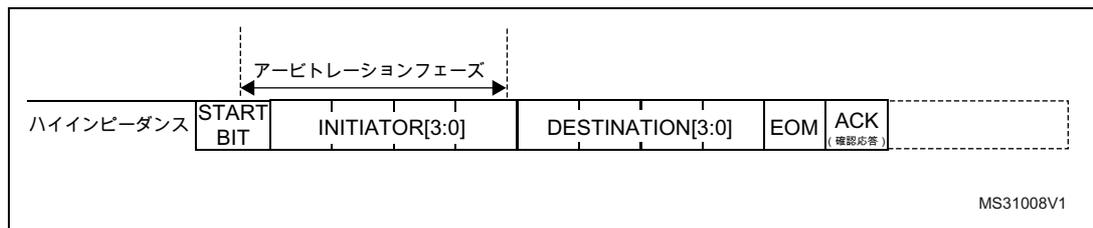
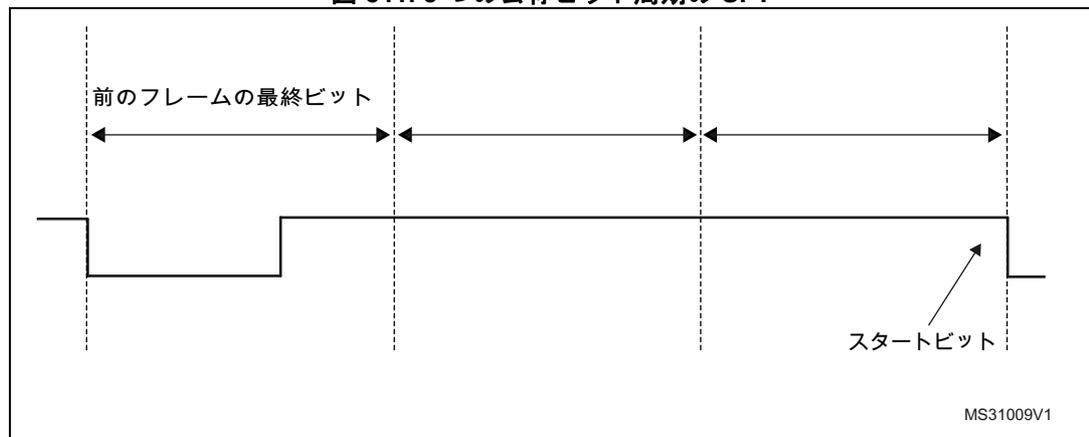


図 511 は、3 つの公称ビット周期の SFT の例を表します。

図 511. 3 つの公称ビット周期の SFT



送信を開始する前に、設定可能な時間枠がカウントされます。

SFT=0x0 設定では、HDMI-CEC デバイスが自動 SFT 計算を実行し、HDMI-CEC 規格に準拠していることを保証します。

- CEC が、送信が失敗した最後のバスイニシエータである場合、2.5 データビット周期
- CEC が新しいバスイニシエータである場合、4 データビット周期
- CEC が、送信が成功した最後のバスイニシエータである場合、6 データビット周期

これは、失敗した送信に最上位の優先順位、成功した送信の最後のイニシエータに最下位の優先順位を保証するために実行されます。

それ以外の場合は、固定されたタイミング値をカウントするために、SFT ビットを設定できる場合があります。可能な値は、0.5、1.5、2.5、3.5、4.5、5.5、6.5 データビット周期です。

## 39.4.1 SFT オプションビット

SFTOPT=0 の設定の場合、ソフトウェアによって送信開始コマンドがセットされると SFT のカウントが開始されます (TXSOM=1)。

SFTOPT=1 の場合、バスアイドル条件またはラインエラー条件が検出されると HDMI-CEC デバイスによって自動的に SFT のカウントが開始されます。TXSOM コマンドのセット時に SFT タイマが完了すると、送信は遅延なくただちに開始されます。SFT タイマが実行されたままである場合、システムは送信が開始される前にタイマが経過するまで待ちます。

SFTOPT=1 の場合、SFT タイマを開始させるバスイベント条件が、以下の場合に検出されます。

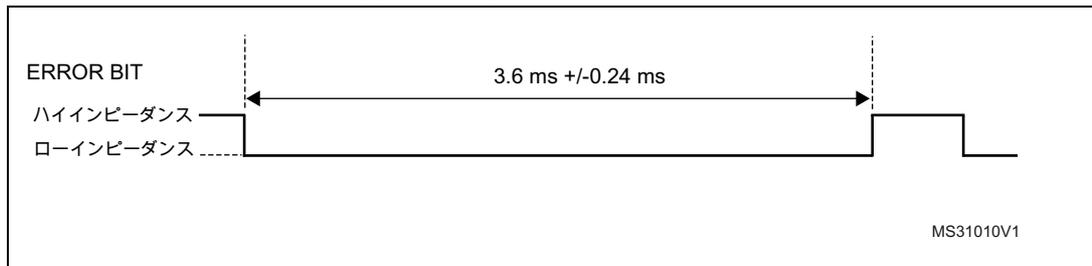
- 通常の送信／受信終了時に、TXEND/RXEND ビットはメッセージの最後のビット (ACK ビット) の最小公称データビット時間にセットされます。
- 送信エラーが検出された場合、TXERR 送信エラー検出時に SFT タイマが開始されます (TXERR=1)。
- CEC フォロワからの確認応答を喪失した場合、TXACKE ビットがセットされたとき、つまり ACK ビットの公称サンプリング時間に、SFT タイマが開始されます。
- 送信アンダーランエラーが発生した場合、ACK ビットの終了時に TXUDR ビットをセットすると SFT タイマが開始されます。
- 受信のアボートを意味する受信エラーが検証された場合、SFT タイマはエラー検出と同じ時間に開始されます。エラービットが生成された場合、エラービットの終了時に SFT のカウントが開始されます。
- 開始ビットに誤りがある場合や、アイドル状態からコード化されていないローインピーダンスバス状態のいずれかである場合、SFT タイマはバスがハイインピーダンスアイドル状態から戻るとすぐに再開されます。

## 39.5 エラー処理

### 39.5.1 ビットエラー

開始ビットを除くデータビットが無効であるとみなされる場合は、フォロウによってそのようなエラーが通知されることが期待されます。これには、公称データビット周期の 1.4 から 1.6 倍の CEC ライン上でロービット周期 (つまり、公称 3.6 ms) を生成して実行されます。

図 512. エラービットタイミング



### 39.5.2 メッセージエラー

メッセージは喪失したものとみなされ、以下の条件により再送信することができます。

- メッセージは、直接アドレス指定されたメッセージでは確認応答されません。
- メッセージは、ブロードキャストメッセージでは否定応答されます。
- 予期しないローインピーダンスが、CEC ライン上で検出されました (ラインエラー)。

CEC インタフェースがデータビットを受信する際に、3 種類のエラーフラグを検出できます。

### 39.5.3 ビット立ち上がりエラー (BRE)

ビット立ち上がりエラー (BRE) : 想定したウィンドウの外にビット立ち上がりエッジが検出されるとセットされます (図 513 を参照)。BRE フラグは BREIE=1 の場合にも CEC 割り込みを生成します。

BRE 検出の場合、BRESTP ビット値によってメッセージの受信を停止することができます。BREGEN ビットをセットすると、エラービットが生成されます。

BRESTP=1 のブロードキャストメッセージで BRE が検出されると、BREGEN=0 の場合でもエラービットが生成され、イニシエータによって失敗した送信の再試行が強制されます。エラービット生成は、BREGEN=0、BRDNOGEN=1 を設定することで無効化できます。

### 39.5.4 ショートビット周期エラー (SBPE)

SBPE は、ビット立ち下がりエッジが予想よりも早く検出された場合にセットされます (図 513 を参照)。SBPE フラグは SBPEIE=1 の場合にも CEC 割り込みを生成します。

エラービットは、SBPE エラー検出時に常にライン上に生成されます。エラービットは、リッスンモードのみがセットされ (LSTN=1)、以下の条件を満たしている場合には SBPE 検出時に生成されません。

- 直接アドレス指定されたメッセージは、SBPE とともに受信されます。
- ブロードキャストメッセージは、SBPE AND BRDNOGEN=1 とともに受信されます。

## 39.5.5 ロングビット周期エラー (LBPE)

LBPE は、有効なウィンドウでビット立ち下がりエッジが検出されない場合にセットされます(図 513 を参照)。LBPE フラグは LBPEIE=1 の場合にも CEC 割り込みを生成します。

LBPE は常に受信を停止し、LBPEGEN ビットがセットされるとライン上にエラービットが生成されます。

ブロードキャストメッセージで LBPE が検出されると、LBPEGEN=0 の場合でもエラービットが生成され、イニシエータによって失敗した送信の再試行が強制されます。エラービット生成は、LBPEGEN=0、BRDNOGEN=1 を設定することで無効化できます。

**注：** *BREGEN=1、BRESTOP=0 設定を避ける必要があります。*

図 513. エラー処理

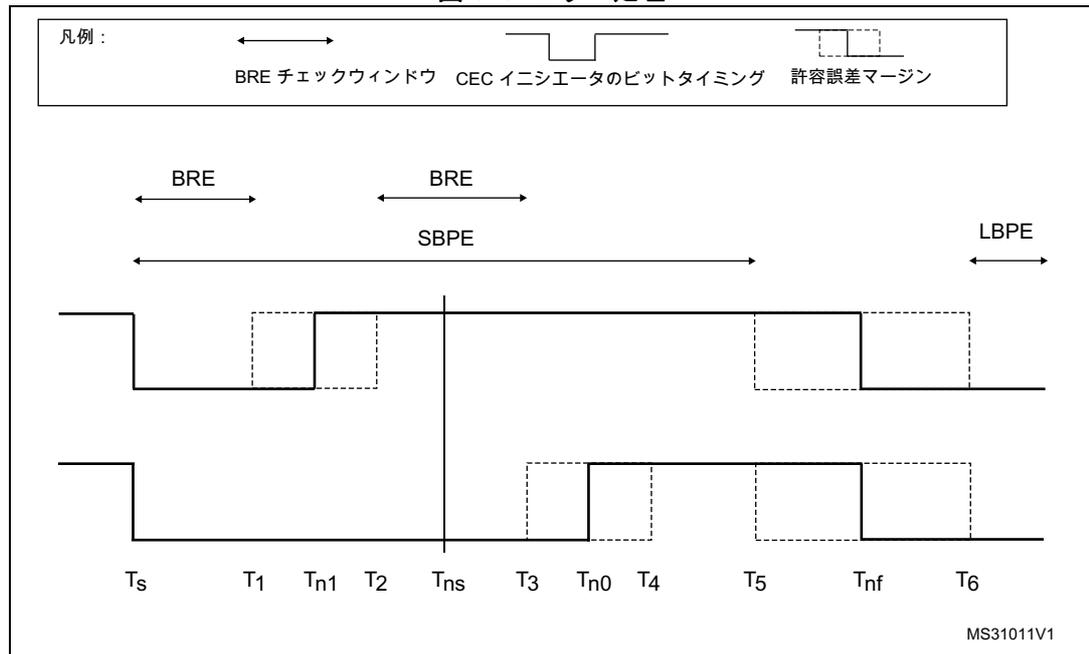


表 247. エラー処理タイミングパラメータ

時間	RXTOL	ms	説明
$T_s$	x	0	ビット開始イベント。
$T_1$	1	0.3	論理 1 を示すためにローからハイに移移する一番早い時間。
	0	0.4	
$T_{n1}$	x	0.6	論理 1 を示すためにローからハイに移移する公称時間。
$T_2$	0	0.8	論理 1 を示すためにローからハイに移移する一番遅い時間。
	1	0.9	
$T_{ns}$	x	1.05	公称サンプリング時間。
$T_3$	1	1.2	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番早い時間。
	0	1.3	
$T_{n0}$	x	1.5	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される公称時間。

時間	RXTOL	ms	説明
T <sub>4</sub>	0	1.7	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番遅い時間。
	1	1.8	
T <sub>5</sub>	1	1.85	後続のビットが開始される一番早い時間。
	0	2.05	
T <sub>nf</sub>	x	2.4	公称データビット周期。
T <sub>6</sub>	0	2.75	後続のビットが開始される一番遅い時間。
	1	2.95	

### 39.5.6 送信エラー検出 (TXERR)

CEC イニシエータでは、ハイインピーダンスの送信中でフォロウがビットをアサートしないと想定される場合に、CEC ラインでローインピーダンスを検出すると、TXERR フラグをセットします。TXERR フラグは TXERRIE=1 の場合にも CEC 割り込みを生成します。

TXERR のアサートによってメッセージの送信が停止されます。アプリケーションでは、失敗した送信を最大 5 回まで再試行することができます。

TXERR のチェックは、さまざまな CEC ラインの状態または RX 許容誤差設定によって、異なる方法で実行されます。

図 514. TXERR 検出

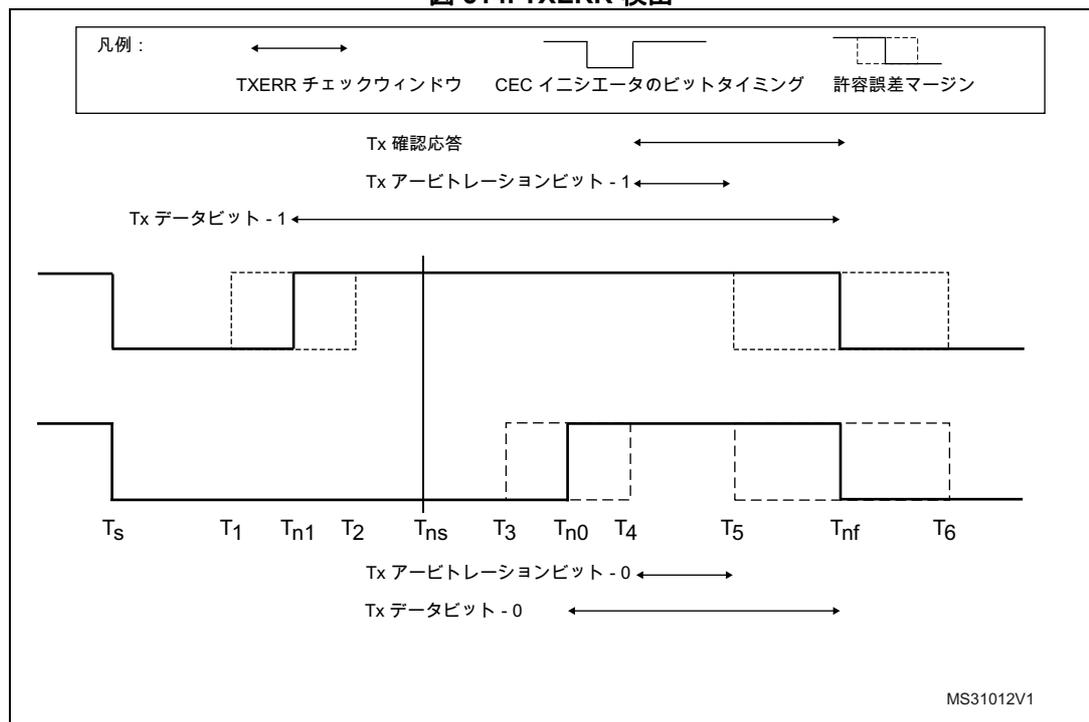


表 248. TXERR タイミングパラメータ

時間	RXTOL	ms	説明
$T_s$	x	0	ビット開始イベント。
$T_1$	1	0.3	論理 1 を示すためにローからハイに遷移する一番早い時間。
	0	0.4	
$T_{n1}$	x	0.6	論理 1 を示すためにローからハイに遷移する公称時間。
$T_2$	0	0.8	論理 1 を示すためにローからハイに遷移する一番遅い時間。
	1	0.9	
$T_{ns}$	x	1.05	公称サンプリング時間。
$T_3$	1	1.2	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番早い時間。
	0	1.3	
$T_{n0}$	x	1.5	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される公称時間。
$T_4$	0	1.7	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番遅い時間。
	1	1.8	
$T_5$	1	1.85	後続のビットが開始される一番早い時間。
	0	2.05	
$T_{nf}$	x	2.4	公称データビット周期。
$T_6$	0	2.75	後続のビットが開始される一番遅い時間。
	1	2.95	

## 39.6 HDMI-CEC 割り込み

次の場合に割り込みを生成できます。

- 受信ブロック転送が完了または受信エラーが発生した場合の受信時。
- 送信ブロック転送が完了または送信エラーが発生した場合の送信時。

表 249. HDMI-CEC 割り込み

割り込みイベント	イベントフラグ	イネーブル制御ビット
Rx バイト受信	RXBR	RXBRIE
受信終了	RXEND	RXENDIE
Rx オーバーラン	RXOVR	RXOVRIE
Rx ビット立ち上がりエラー	BRE	BREIE
Rx ショートビット周期エラー	SBPE	SBPEIE
Rx ロングビット周期エラー	LBPE	LBPEIE
Rx 喪失確認応答エラー	RXACKE	RXACKEIE
アービトレーション喪失	ARBLST	ARBLSTIE
Tx バイトリクエスト	TXBR	TXBRIE
送信終了	TXEND	TXENDIE
Tx バッファアンダーラン	TXUDR	TXUDRIE
Tx エラー	TXERR	TXERRIE
Tx 喪失確認応答エラー	TXACKE	TXACKEIE

## 39.7 HDMI-CEC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(59 ページ\)](#) を参照してください。

### 39.7.1 CEC 制御レジスタ (CEC\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TXEOM**:Tx メッセージの終了

TXEOM ビットは、CEC メッセージの最後のバイトの送信を指示するためにソフトウェアによってセットされません。

TXEOM は、TXSOM の場合と同時に、同じ条件下でハードウェアによってクリアされます。

0 : TXDR データバイトは EOM=0 で送信されます。

1 : TXDR データバイトは EOM=1 で送信されます。

**注 :** *TXEOM は、CECEN=1 のときにセットする必要があります。*

*TXEOM は、TXDR に送信データを書き込む前にセットする必要があります。*

*TXSOM=0 のときに TXEOM がセットされると、送信メッセージは 1 バイト (HEADER) のみで構成されず (PING メッセージ)。*

ビット 1 **TXSOM** : Tx メッセージの開始

TXSOM は、CEC メッセージの最初のバイトの送信を指示するためにソフトウェアによってセットされます。CEC メッセージが 1 バイトのみで構成される場合、TXEOM は TXSOM の前にセットする必要があります。

スタートビットは、SFT のカウント後に CEC ライン上で効果的に開始されます。メッセージの受信中に TXSOM がセットされると、送信は受信終了後に開始されます。

TXSOM は、送信アンダーラン (TXUDR=1)、否定応答 (TXACK=1)、および送信エラー (TXERR=1) が発生したときに、メッセージの最後のバイトが肯定応答 (TXEND=1) とともに送信された後で、ハードウェアによってクリアされます。また、CECEN=0 でもクリアされます。アービトラージ喪失 (ARBLST=1) の場合はクリアされず、送信は自動的に再試行されます。

TXSOM は、保留中や実行中の送信リクエストがあるかどうかをアプリケーションに知らせる、ステータスビットとしても使用できます。アプリケーションは、CECEN ビットをクリアすることで送信リクエストをいつでもアポートできます。

- 0 : 実行中の CEC 送信はありません。
- 1 : CEC 送信コマンド。

**注 :** TXSOM は、CECEN=1 のときにセットする必要があります。  
 TXSOM は、送信データを TXDR で使用できる場合にセットする必要があります。  
 自己のペリフェラルアドレスを含む HEADER の最初の 4 つのビットは、受信用としてのみ使用される CEC\_CFGR.OAR ではなく、TXDR[7:4] から取得されます。

### ビット 0 CECEN : CEC 有効

CECEN ビットは、ソフトウェアによってセット/クリアされます。CECEN=1 の場合にメッセージの受信が開始され、TXSOM 制御が有効化されます。CECEN=0 の場合に CEC ペリフェラルが無効化され、CEC\_CR レジスタのすべてのビットがクリアされ、実行中のすべての送受信がアポートされます。

- 0 : CEC ペリフェラルはオフです。
- 1 : CEC ペリフェラルはオンです。

## 39.7.2 CEC 設定レジスタ (CEC\_CFGR)

このレジスタは HDMI-CEC コントローラの設定に使用されます。

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

**注意 :** CECEN=0 の場合のみ、CEC\_CFGR に書き込む必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSTN	OAR[14:0]														
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SFT OPT	BRDNO GEN	LBPE GEN	BRE GEN	BRE STP	RX TOL	SFT[2:0]		
							rw	rw	rw	rw	rw	rw	rw		

### ビット 31 LSTN : リッスンモード

LSTN ビットは、ソフトウェアによってセット/クリアされます。

- 0 : CEC ペリフェラルでは、自己アドレス (OAR) に宛てられたメッセージのみを受信します。他の転送先に宛てられたメッセージは無視されます。ブロードキャストメッセージは常に受信されます。
- 1 : CEC ペリフェラルでは、肯定応答とともに自己アドレス (OAR) に宛てられたメッセージを受信します。他の転送先に宛てられたメッセージは受信されますが、CEC バスに干渉することはありません。確認応答は送信されません。

### ビット 30:16 OAR : 自己アドレス設定

OAR ビットは、受信モードで考慮する必要がある転送先論理アドレスを選択するために、ソフトウェアによってセットされます。各ビットをセットすると、特定のビット位置によって指定された CEC 論理アドレスが有効になります。

HEADER 受信終了時に、受信した転送先アドレスが有効化されたアドレスと比較されます。アドレスが一致する場合は、受信メッセージが確認応答され、受信されます。アドレスが一致しない場合は、受信メッセージはリッスンモード (LSTN=1) でのみ受信され、確認応答は送信されません。ブロードキャストメッセージは常に受信されます。

例 :

OAR = 0b000 0000 0010 0001 は、アドレス 0x0 および 0x5 が CEC で確認応答されることを意味します。結果として、これらのアドレスのいずれかにそれぞれ宛てられたメッセージが受信されます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **SFTOP** : SFT オプションビット

SFTOPT ビットは、ソフトウェアによってセット/クリアされます。

- 0 : ソフトウェアによって TXSOM がセットされると、SFT タイマが開始されます。
- 1 : メッセージの送受信の終了時に SFT タイマが自動的に開始されます。

ビット 7 **BRDNOGEN** : ブロードキャストでの生成エラービットの回避

BRDNOGEN ビットは、ソフトウェアによってセット/クリアされます。

- 0 : BRESTP=1 および BREGEN=0 のブロードキャストメッセージで BRE が検出されると、CEC ライン上にエラービットが生成されます。LBPEGEN=0 のブロードキャストメッセージで LBPE が検出されると、CEC ライン上にエラービットが生成されます。
- 1 : エラービットは、上記と同じ条件では生成されません。リッスンモードがセットされている場合、ブロードキャストメッセージで SBPE が検出されてもエラービットは生成されません。

ビット 6 **LBPEGEN** : ロングビット周期エラーでのエラービットの生成

LBPEGEN ビットは、ソフトウェアによってセット/クリアされます。

- 0 : LBPE 検出で、CEC ライン上にエラービットは生成されません。
- 1 : LBPE 検出で、CEC ライン上にエラービットが生成されます。

**注 :** *BRDNOGEN=0 の場合、LBPEGEN=0 の場合でもブロードキャストでは LBPE 検出時にエラービットが生成されます。*

ビット 5 **BREGEN** : ビット立ち上がりエラーでのエラービットの生成

BREGEN ビットは、ソフトウェアによってセット/クリアされます。

- 0 : BRE 検出で、CEC ライン上にエラービットは生成されません。
- 1 : BRE 検出で、CEC ライン上にエラービットが生成されます (BRESTP がセットされている場合)。

**注 :** *BRDNOGEN=0 の場合、BREGEN=0 の場合でもブロードキャストで BRESTP=1 の BRE 検出時にエラービットが生成されます。*

ビット 4 **BRESTP** : ビット立ち上がりエラーでの Rx 停止

BRESTP ビットは、ソフトウェアによってセット/クリアされます。

- 0 : BRE 検出で CEC メッセージの受信が停止されることはありません。データビットは 1.05 ms でサンプリングされます。
- 1 : BRE 検出でメッセージの受信が停止されます。

ビット 3 **RXTOL** : Rx 許容誤差

RXTOL ビットは、ソフトウェアによってセット/クリアされます。

- 0 : 標準許容誤差マージン
  - スタートビット : +/- 200  $\mu$ s 立ち上がり、 +/- 200  $\mu$ s 立ち下がり
  - データビット : +/- 200  $\mu$ s 立ち上がり、 +/- 350  $\mu$ s 立ち下がり
- 1 : 拡張された許容誤差
  - スタートビット : +/- 400  $\mu$ s 立ち上がり、 +/- 400  $\mu$ s 立ち下がり
  - データビット : +/- 300  $\mu$ s 立ち上がり、 +/- 500  $\mu$ s 立ち下がり

### ビット 2:0 SFT : 信号フリータイム

SFT ビットは、ソフトウェアによってセットされます。SFT=0x0 設定では、送信前に待った公称データビット周期数は、送信履歴に従って、ハードウェアによって支配されます。その他の設定では、SFT 番号はソフトウェアによって決定されます。

- 0x0
  - CEC が、送信が失敗した最後のバスイニシエータである場合、2.5 データビット周期 (ARBLST=1、TXERR=1、TXUDR=1、または TXACK= 1)
  - CEC が新しいバスイニシエータである場合、4 データビット周期
  - CEC が、送信が成功した最後のバスイニシエータである場合、6 データビット周期 (TXEOM=1)
- 0x1 : 0.5 公称データビット周期
- 0x2 : 1.5 公称データビット周期
- 0x3 : 2.5 公称データビット周期
- 0x4 : 3.5 公称データビット周期
- 0x5 : 4.5 公称データビット周期
- 0x6 : 5.5 公称データビット周期
- 0x7 : 6.5 公称データビット周期

## 39.7.3 CEC Tx データレジスタ (CEC\_TXDR)

アドレスオフセット : 0x8

リセット値 : 0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TXD[7:0]** : Tx データレジスタ

TXD は、送信されるデータバイトを含む書き込み専用のレジスタです。

**注 :** TXD は、TXSTART=1 のときに書き込む必要があります。

## 39.7.4 CEC Rx データレジスタ (CEC\_RXDR)

アドレスオフセット : 0xC

リセット値 : 0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RXD[7:0]** : Rx データレジスタ

RXD は読み出し専用で、CEC ラインから受信した最後のデータバイトを含みます。

## 39.7.5 CEC 割り込みおよびステータスレジスタ (CEC\_ISR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TX ACKE	TX ERR	TX UDR	TX END	TXBR	ARB LST	RX ACKE	LBPE	SBPE	BRE	RX OVR	RX END	RXBR
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

### ビット 12 TXACKE : Tx 喪失確認応答エラー

送信モードでは、TXACKE はハードウェアによってセットされ、アプリケーションに確認応答を受信していないことを知らせます。ブロードキャスト送信の場合、TXACKE はアプリケーションに否定応答を受信したことを知らせます。TXACKE はメッセージの送信をアボートし、TXSOM および TXEOM 制御をクリアします。

TXACKE はソフトウェアで 1 を書き込むことによってクリアされます。

### ビット 11 TXERR : Tx エラー

送信モードでは、CEC イニシエータがリリース中に CEC ラインでローインピーダンスを検出した場合に、ハードウェアによって TXERR がセットされます。TXERR はメッセージの送信をアボートし、TXSOM および TXEOM 制御をクリアします。

TXERR はソフトウェアで 1 を書き込むことによってクリアされます。

### ビット 10 TXUDR : Tx バッファアンダーラン

送信モードでは、次のバイト送信前の TXUDR のロードにアプリケーションが間に合わなかった場合に、ハードウェアによって TXUDR がセットされます。TXUDR はメッセージの送信をアボートし、TXSOM および TXEOM 制御ビットをクリアします。

TXUDR はソフトウェアで 1 を書き込むことによってクリアされます。

### ビット 9 TXEND : 送信終了

TXEND は、CEC メッセージの最後のバイトが正常に送信されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。TXEND は、TXSOM および TXEOM 制御ビットをクリアします。

TXEND はソフトウェアで 1 を書き込むことによってクリアされます。

### ビット 8 TXBR : Tx バイトリクエスト

TXBR は、次の送信データを TXDR に書き込む必要があることをアプリケーションに知らせるために、ハードウェアによってセットされます。TXBR は、現在送信されたバイトの 4 番目のビットが送信されたときにセットされます。アプリケーションでは、送信アンダーランエラー (TXUDR) が発生する前に、6 公称データビット周期内の TXDR に次のバイトを書き込む必要があります。

TXBR はソフトウェアで 1 を書き込むことによってクリアされます。

### ビット 7 ARBLST : アービトレーション喪失

ARBLST は、TXSOM コマンドに続くアービトレーション喪失イベントによって CEC デバイスが受信に切り換わることをアプリケーションに知らせるために、ハードウェアによってセットされます。ARBLST の原因は、先に開始される、またはより高い HEADER 優先順位で同時に開始される、競合するいずれかの CEC デバイスです。ARBLST のアサート後、TXSOM ビットでは次の送信試行を保留のままとします。

ARBLST はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 6 RXACKE** : Rx 喪失確認応答

受信モードでは、RXACKE はハードウェアによってセットされ、CEC ラインで確認応答を確認できなかったことをアプリケーションに知らせます。RXACKE はブロードキャストメッセージにのみ適用でき、またリスンモードでは直接アドレス指定されていないメッセージにも適用できます(転送先アドレスは OAR では無効)。RXACKE はメッセージの受信をアボートします。

RXACKE はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 5 LBPE** : Rx ロングビット周期エラー

ロングビット周期エラーのデータビット波形が検出された場合、LBPE はハードウェアによってセットされます。立ち下がリエッジが存在している場合、LBPE は、RXTOL が許容する最大ビット拡張許容誤差の最後にセットされます。LBPE は、常に CEC メッセージの受信を停止します。LBPE で、LBPEGEN=1 のときに CEC ライン上にエラービットが生成されます。ブロードキャストの場合は、LBPEGEN=0 の場合でもエラービットが生成されます。

LBPE はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 4 SBPE** : Rx ショートビット周期エラー

ショートビット周期エラーのデータビット波形が検出された場合、SBPE はハードウェアによってセットされます。SBPE は、想定された立ち下がリエッジの発生時にセットされます。SBPE で、CEC ライン上にエラービットが生成されます。

SBPE はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 3 BRE** : Rx ビット立ち上がりエラー

ビット立ち上がりエラーのデータビット波形が検出された場合、BRE はハードウェアによってセットされます。立ち上がりエッジが存在している場合、BRE は、立ち上がりエッジの誤配置が発生したとき、または RXTOL が許容する最大 BRE 許容誤差の最後にセットされます。BRESTOP=1 のとき、BRE ではメッセージの受信を停止します。BRE で、BREGEN=1 のときに CEC ライン上にエラービットが生成されます。

BRE はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 2 RXOVR** : Rx オーバーラン

新しいバイトの受信時に RXBR がクリアされていない場合や RXD に保存されていない場合、RXOVR はハードウェアによってセットされます。RXOVR アサートではメッセージの受信を停止するため、確認応答は送信されません。ブロードキャストの場合、否定応答が送信されます。

RXOVR はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 1 RXEND** : 受信終了

RXEND は、CEC ラインから CEC メッセージの最後のバイトを受信し、RXD バッファに格納したことをアプリケーションに知らせるために、ハードウェアによってセットされます。RXEND は、RXBR と同じ時間でセットされます。

RXEND はソフトウェアで 1 を書き込むことによってクリアされます。

**ビット 0 RXBR** : Rx バイト受信

RXBR ビットは、CEC ラインから新しいバイトを受信し、RXD バッファに格納したことをアプリケーションに知らせるために、ハードウェアによってセットされます。

RXBR はソフトウェアで 1 を書き込むことによってクリアされます。

## 39.7.6 CEC 割り込み有効レジスタ (CEC\_IER)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TXACK IE	TXERR IE	TX UDRIE	TXEND IE	TXBR IE	ARBLST IE	RXACK IE	LBPE IE	SBPE IE	BREIE	RXOVR IE	RXEND IE	RXBR IE
			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TXACKIE** : Tx 喪失確認応答エラー割り込み有効  
TXACKIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : TXACKIE 割り込み無効
- 1 : TXACKIE 割り込み有効

ビット 11 **TXERRIE** : Tx エラー割り込み有効  
TXERRIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : TXERRIE 割り込み無効
- 1 : TXERRIE 割り込み有効

ビット 10 **TXUDRIE** : Tx アンダーラン割り込み有効  
TXUDRIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : TXUDRIE 割り込み無効
- 1 : TXUDRIE 割り込み有効

ビット 9 **TXENDIE** : Tx メッセージの終了割り込み有効  
TXENDIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : TXENDIE 割り込み無効
- 1 : TXENDIE 割り込み有効

ビット 8 **TXBRIE** : Tx バイトリクエスト割り込み有効  
TXBRIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : TXBRIE 割り込み無効
- 1 : TXBRIE 割り込み有効

ビット 7 **ARBLSTIE** : アービトレーション喪失割り込み有効  
ARBLSTIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : ARBLSTIE 割り込み無効
- 1 : ARBLSTIE 割り込み有効

ビット 6 **RXACKIE** : Rx 喪失確認応答エラー割り込み有効  
RXACKIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : RXACKIE 割り込み無効
- 1 : RXACKIE 割り込み有効

ビット 5 **LBPEIE** : ロングビット周期エラー割り込み有効  
LBPEIE ビットは、ソフトウェアによってセット/クリアされます。

- 0 : LBPEIE 割り込み無効
- 1 : LBPEIE 割り込み有効

- ビット 4 **SBPEIE** : ショートビット周期エラー割り込み有効  
SBPEIE ビットは、ソフトウェアによってセット/クリアされます。  
0 : SBPE 割り込み無効  
1 : SBPE 割り込み有効
- ビット 3 **BREIE** : ビット立ち上がりエラー割り込み有効  
BREIE ビットは、ソフトウェアによってセット/クリアされます。  
0 : BRE 割り込み無効  
1 : BRE 割り込み有効
- ビット 2 **RXOVRIE** : Rx バッファオーバーラン割り込み有効  
RXOVRIE ビットは、ソフトウェアによってセット/クリアされます。  
0 : RXOVR 割り込み無効  
1 : RXOVR 割り込み有効
- ビット 1 **RXENDIE** : 受信終了割り込み有効  
RXENDIE ビットは、ソフトウェアによってセット/クリアされます。  
0 : RXEND 割り込み無効  
1 : RXEND 割り込み有効
- ビット 0 **RXBRIE** : Rx バイト受信割り込み有効  
RXBRIE ビットは、ソフトウェアによってセット/クリアされます。  
0 : RXBR 割り込み無効  
1 : RXBR 割り込み有効

**注意 :** (\*) **CECEN=0** の場合は、**CEC\_IER** に書き込む必要があります。

## 39.7.7 HDMI-CEC レジスタマップ

次の表に HDMI-CEC レジスタの一覧を示します。

表 250. HDMI-CEC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	CEC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																		
0x04	CEC_CFGR	LSTN	OAR[14:0]														Res.	Res.	Res.	SFTOPT	BRNOGEN	LBPEGEN	BREGEN	BRESTP	RXTOL	SFT[2:0]									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	CEC_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]									
	リセット値																									0	0	0	0	0	0	0	0	0	
0x0C	CEC_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]										
	リセット値																																		
0x10	CEC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																		
0x14	CEC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																		

レジスタ境界アドレスについては、[セクション 2.2.2 \(66 ページ\)](#) を参照してください。

## 40 デバッグサポート (DBG)

### 40.1 概要

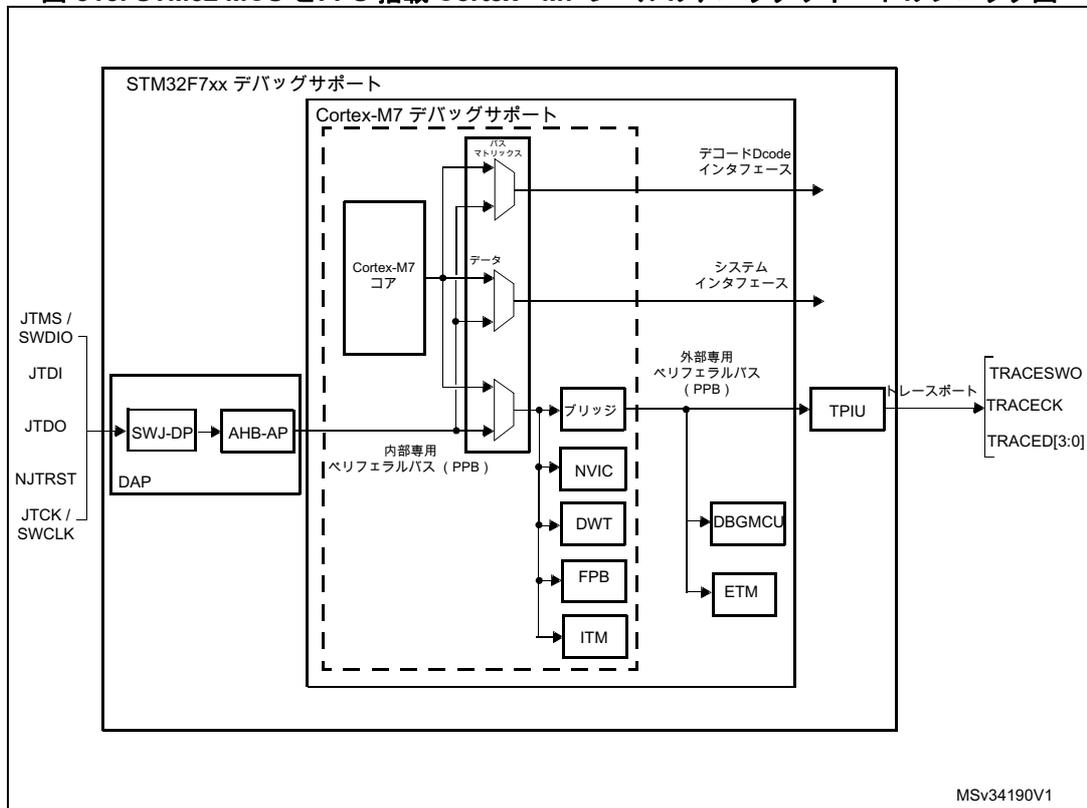
STM32F75xxx および STM32F74xxx は、FPU コア搭載 Cortex<sup>®</sup>-M7 を中心にして構築されており、高度なデバッグ機能のためのハードウェア拡張機能を含んでいます。デバッグ拡張機能によって、コアは、特定の命令フェッチ時（ブレークポイント）またはデータアクセス時（ウォッチポイント）に停止することができます。停止したとき、コアの内部状態とシステムの外部状態を調査することができます。調査が終わったら、コアとシステムを復元して、プログラム実行を再開することができます。

デバッグ機能は、STM32F75xxx および STM32F74xxx MCU への接続時とデバッグ時にデバッグによって使用されます。

デバッグ用に、次の 2 つのインターフェースを使用できます。

- シリアルワイヤ
- JTAG デバッグポート

図 515. STM32 MCU と FPU 搭載 Cortex<sup>®</sup>-M7 レベルのデバッグサポートのブロック図



注： FPU コア搭載 Cortex<sup>®</sup>-M7 に内蔵されているデバッグ機能は、ARM<sup>®</sup> CoreSight Components Technical Reference Manual のサブセットです。

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

FPU コア搭載 ARM<sup>®</sup> Cortex<sup>®</sup>-M7 は統合的なオンチップデバッグサポート機能を提供します。この機能は以下の要素で構成されます。

- SWJ-DP : シリアルワイヤ/JTAG デバッグポート
- AHP-AP : AHB アクセスポート
- ITM : 計測トレースマクロセル (Instrumentation Trace Macrocell)
- FPB : フラッシュパッチブレークポイント (Flash Patch Breakpoint)
- DWT : データウォッチポイントトリガ
- TPIU : トレースポートインタフェースユニット (Trace Port Interface Unit : 対応するピンが配置される大型パッケージで使用)
- ETM : エンベデッドトレースマクロセル (Embedded Trace Macrocell : 対応するピンが配置される大型パッケージで使用)

また、STM32F75xxx および STM32F74xxx専用の以下のデバッグ機能も内蔵されています。

- 柔軟性の高いデバッグピンの割り当て
- MCU デバッグボックス (低電力モードのサポート、ペリフェラルクロックの制御など)

**注 :** **FPU コア搭載 ARM<sup>®</sup> Cortex<sup>®</sup>-M7 でサポートされているデバッグ機能の詳細は、FPU 搭載 Cortex<sup>®</sup>-M7 Technical Reference Manual および CoreSight Components Technical Reference Manual (セクション 40.2 : ARM<sup>®</sup> リファレンス資料) を参照してください。**

## 40.2 ARM<sup>®</sup> リファレンス資料

- FPU 搭載Cortex<sup>®</sup>-M7 Technical Reference Manual (TRM)  
(1 ページの関連資料を参照)
- ARM<sup>®</sup> Debug Interface V5 Architecture Specification
- ARM<sup>®</sup> CoreSight Components Technical Reference Manual

## 40.3 SWJ デバッグポート (シリアルワイヤと JTAG)

STM32F75xxx および STM32F74xxx のコアには、シリアルワイヤ/JTAG デバッグポート (SWJ-DP) が組み込まれています。これは、JTAG-DP (5 ピン) インタフェースと SW-DP (2 ピン) インタフェースを組み合わせた ARM<sup>®</sup> 標準の CoreSight デバッグポートです。

- JTAG デバッグポート (JTAG-DP) は、AHP-AP ポートに 5 ピンの標準 JTAG インタフェースを提供します。
- シリアルワイヤデバッグポート (SW-DP) は、AHP-AP ポートに 2 ピン (クロック + データ) のインタフェースを提供します。

SWJ-DP では、SW-DP の 2 個の JTAG ピンは、JTAG-DP の 5 個の JTAG ピンの一部と多重化されています。

図 516. SWJ デバッグポート

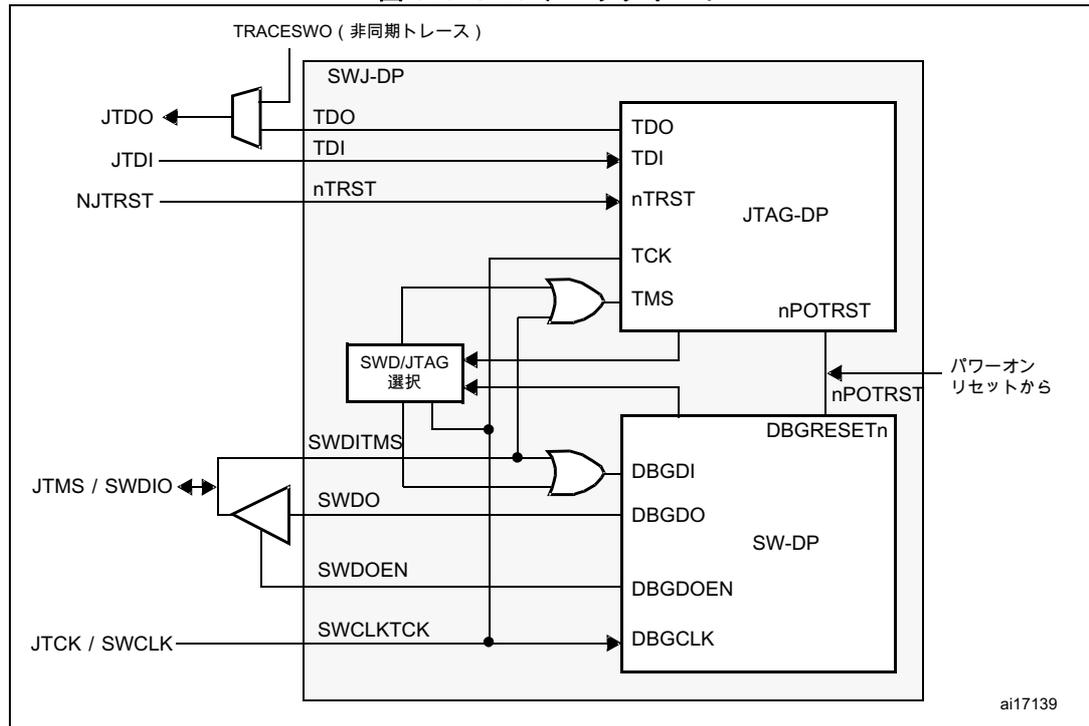


図 516 は、非同期 TRACE 出力 (TRACESWO) が TDO と多重化されていることを示します。つまり非同期トレースは、JTAG-DP ではなく、SW-DP にのみ使用できます。

### 40.3.1 JTAG-DP または SW-DP の選択メカニズム

デフォルトでは、JTAG デバッグポートが選択されています。

SW-DP に切り替えるには、デバッガホストは、TMS/TCK (それぞれ SWDIO/SWCLK に配置) に専用の JTAG シーケンスを提供して、JTAG-DP を無効にし SW-DP を有効にする必要があります。このように、SWCLK ピンと SWDIO ピンのみを使用して、SW-DP を有効にすることができます。

このシーケンスは、

1. TMS (SWDIO) =1 で 50 を超える TCK サイクルを送信します。
2. TMS (SWDIO) に 16 ビットシーケンス (0111100111100111) を MSB ファーストで送信します。
3. TMS (SWDIO) =1 で 50 を超える TCK サイクルを送信します。

## 40.4 ピン名とデバッグポートピン

STM32F75xxx および STM32F74xxx MCUは、使用できるピン数の異なるさまざまなパッケージに組み込まれています。このため、ピンを使用する一部の機能 (TPIU パラレル出力インタフェース) は、パッケージによって異なることがあります。

### 40.4.1 SWJ デバッグポートピン

5 個のピンが SWJ-DP 用の STM32F75xxx および STM32F74xxx からの出力として使用されます (汎用入出力のオルタネート機能)。これらのピンはすべてのパッケージで使用できます。

表 251. SWJ デバッグポートピン

SWJ-DP ピン名	JTAG デバッグポート		SW デバッグポート		ピン割り当て
	タイプ	説明	タイプ	デバッグ割り当て	
JTMS/SWDIO	I	JTAG テストモード選択	入出力	シリアルワイヤデータ入出力	PA13
JTCK/SWCLK	I	JTAG テストクロック	I	シリアルワイヤクロック	PA14
JTDI	I	JTAG テストデータ入力	-	-	PA15
JTDO/TRACESWO	O	JTAG テストデータ出力	-	非同期トレースが有効な場合は TRACESWO	PB3
NJTRST	I	JTAG テスト nReset	-	-	PB4

### 40.4.2 柔軟性の高い SWJ-DP ピンの割り当て

RESET (SYSRESETn または PORESETn) 後、SWJ-DP 用に使用される 5 個のピンはすべて、デバッグホストによってすぐに使用可能な専用ピンとして割り当てられます (なお、デバッグホストによって明示的にプログラミングされた場合を除いて、トレース出力は割り当てられません)。

ただし、STM32F75xxx および STM32F74xxx MCU は SWJ-DP ポートの一部またはすべてを無効にすることができますので、汎用 IO (GPIO) に使用する関連ピン (下表にグレーで示す) を解放することもできます。SWJ-DPポートピンを無効にする方法の詳細については、[セクション 6.3.2 : I/O ピンオルタネート機能マルチプレクサと配置](#)を参照してください。

表 252. 柔軟性の高い SWJ-DP ピンの割り当て

使用可能なデバッグポート	SWJ IO ピンの割り当て				
	PA13 / JTMS / SWDIO	PA14 / JTCK / SWCLK	PA15 / JTDI	PB3 / JTDO	PB4 / NJTRST
全 SWJ (JTAG-DP + SW-DP)、リセット状態	X	X	X	X	X
全 SWJ (JTAG-DP + SW-DP)、NJTRST なし	X	X	X	X	
JTAG-DP 無効、SW-DP 有効	X	X			
JTAG-DP 無効、SW-DP 無効					解放

**注：** APB ブリッジの書き込みバッファがフルのとき、GPIO\_AFR レジスタへの書き込みには APB 1 サイクル分が追加で必要になります。これは、コアの nTRST および TCK 入力信号でのクリーンレベルを保証するために、JTAGSW ピンの無効化が 2 サイクルで行われるからです。

- サイクル 1：コアへの JTAGSW 入力信号は 1 または 0 に (TRST、TDI、および TMS では 1 に、TCK では 0 に) 接続されます。
- サイクル 2：GPIO コントローラが SWJTAG 入出力ピンの制御 (方向、プルアップ/ダウン、シュミットトリガの有効化などの制御) 信号を受け取ります。

### 40.4.3 JTAG ピンでの内部プルアップ/プルダウン

JTAG 入力ピンは、デバッグモード機能を制御するためにフリップフロップに直結されます。したがって、JTAG 入力ピンをフロート状態にしないことが必要です。これらのフリップフロップの一部のクロックに直結される SWCLK/TCK ピンについては、特に注意が必要です。

入出力レベルを正しく制御するため、デバイスには内部プルアップ/プルダウンが JTAG 入力ピンに内蔵されています。

- NJTRST：内部プルアップ
- JTDI：内部プルアップ
- JTMS/SWDIO：内部プルアップ
- TCK/SWCLK：内部プルダウン

JTAG 入出力がユーザソフトウェアによって解放されると、GPIO コントローラが再び制御権を獲得します。リセット状態では、GPIO 制御レジスタは入出力をこれと同等の状態に設定します。

- NJTRST：AF 入力プルアップ
- JTDI：AF 入力プルアップ
- JTMS/SWDIO：AF 入力プルアップ
- JTCK/SWCLK：AF 入力プルダウン
- JTDO：AF 出力フローティング

ソフトウェアはこれらの入出力を標準の GPIO 信号として使用することができます。

**注：** JTAG IEEE 規格では、TDI、TMS、および nTRST にプルアップを追加することを推奨していますが、TCK に関しては特別な推奨はありません。ただし、JTCK に関しては、デバイスは内蔵プルダウンを必要とします。

プルアップとプルダウンを内蔵しているため、外部抵抗を追加する必要はありません。

## 40.4.4 シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法

シリアルワイヤ DP を使って GPIO を解放するには、ユーザソフトウェアが GPIO\_MODER レジスタで GPIO (PA15、PB3、および PB4) コンフィギュレーションモードを変更する必要があります。これによって PA15、PB3、および PB4 が解放され、GPIO として使用できるようになります。

デバッグ時には、ホストは次の動作を行います。

- システムリセット中に、すべての SWJ ピンが割り当てられます (JTAG-DP + SW-DP)。
- システムリセット中に、デバッグホストは、JTAG-DP から SW-DP に切り替える JTAG シーケンスを送信します。
- システムリセット中に、さらに、デバッグはベクタリセットにブレークポイントを設定します。
- システムリセットは解除され、コアは停止します。
- これ以降のすべてのデバッグ通信は、SW-DP を使用して行われます。残りの JTAG ピンは、ユーザソフトウェアによって GPIO として再割当てできます。

**注：** ユーザソフトウェアの設計に関しては、次の点に注意してください。

デバッグピンを解放するには、リセット後にユーザソフトウェアがピンを解放するまでの一定の期間、デバッグピンは、まず、入力プルアップ (nTRST、TMS、TDI)、プルダウン (TCK)、または出カトライステート (TDO) に設定されることに注意してください。

デバッグピン (JTAG、SW、または TRACE) が配置されると、IOPORT コントローラの対応する入出力ピンの設定を変更しても効果はありません。

## 40.5 STM32F75xxx および STM32F74xxx JTAG デバッグポートの接続

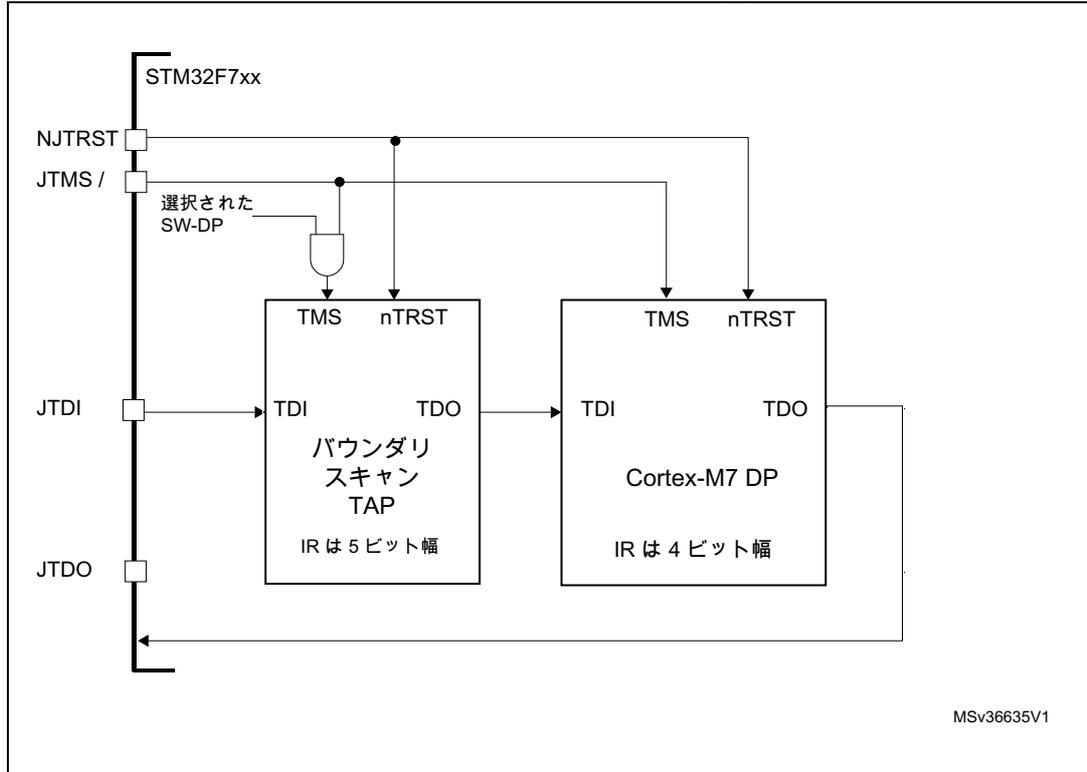
STM32F75xxx および STM32F74xxx MCUは、バウンダリスキャンデバッグポート (IR は 5 ビット幅) と FPU 搭載 Cortex<sup>®</sup>-M7 デバッグポート (IR は 4 ビット幅) というシリアル接続された 2 個の JTAG デバッグポートを内蔵しています。

デバッグ目的で FPU 搭載 Cortex<sup>®</sup>-M7 のデバッグポートにアクセスするには、

1. まず、バウンダリスキャンデバッグポートの BYPASS 命令をシフトする必要があります。
2. 次に、IR シフトごとに、スキャンチェーンには 9 (= 5 + 4) ビットが格納され、未使用のデバッグポート命令を BYPASS 命令によってシフトインする必要があります。
3. データシフトごとに、BYPASS モードにある未使用のデバッグポートは、データスキャンチェーンに 1 個のデータビットを追加します。

**注：** **重要：専用の ARM<sup>®</sup> JTAG シーケンスを使用してシリアルワイヤが選択されると、バウンダリスキャンデバッグポートは自動的に無効になります (JTMS は強制的にハイレベルになります)。**

図 517. JTAG デバッグポートの接続



## 40.6 ID コードとロック機構

STM32F75xxx および STM32F74xxx MCU には、内部にいくつかの ID コードがあります。ツール設計者は、外部 PPB メモリマップのアドレス 0xE0042000 に配置されている MCU デバイス ID コードを使用して、デバッグをロックすることを強く推奨します。

### 40.6.1 MCU デバイス ID コード

STM32F75xxx および STM32F74xxx MCU には MCU ID コードが内蔵されています。この ID は、ST 社製 MCU の部品番号とダイのリビジョンを識別します。これは DBG\_MCU 部品の一部であり、外部 PPB バスに配置されます ([セクション 40.16 \(1631 ページ\)](#) を参照)。このコードにアクセスするには、JTAG デバッグポート (4~5 本のピン)、SW デバッグポート (2 本のピン)、またはユーザソフトウェアを使用します。アクセスは MCU がシステムリセット中でも可能です。

デバッガ/プログラマツールでは、DEV\_ID(11:0) のみを識別に使用してください。

#### DBGMCU\_IDCODE

アドレス : 0xE004 2000

32 ビットアクセスのみサポートされます。読み出し専用。

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

ビット 31:16 **REV\_ID(15:0)** リビジョン識別子

このフィールドは、デバイスのリビジョンを示します。

0x1000 = リビジョン A

0x1001 = リビジョン Z

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DEV\_ID(11:0)** : デバイス識別子

デバイス ID は 0x449 です。

### 40.6.2 バウンダリスキャンデバッグポート

#### JTAG ID コード

STM32F75xxx および STM32F74xxx BSC (バウンダリスキャン) のデバッグポートは 0x06449041 に等しい JTAG ID コードを内蔵しています。

### 40.6.3 FPU 搭載Cortex<sup>®</sup>-M7 のデバッグポート

FPU 搭載 ARM<sup>®</sup> Cortex<sup>®</sup>-M7 のデバッグポートは JTAG ID コードを内蔵しています。この ID コードは ARM<sup>®</sup> のデフォルトであり、変更されていません。このコードは、JTAG デバッグポートからのみアクセスできます。

このコードは 0x5BA00477 です (FPU 搭載 Cortex<sup>®</sup>-M7 に対応、[セクション 40.2 : ARM<sup>®</sup> リファレンス資料](#)を参照)。

## 40.6.4 FPU 搭載Cortex<sup>®</sup>-M7 の JEDEC-106 ID コード

FPU 搭載 ARM<sup>®</sup> Cortex<sup>®</sup>-M7 は JEDEC-106 ID コードを内蔵しています。これは内部 PPB バスのアドレス 0xE00FF000\_0xE00FFFFF に配置された 4KB ROM テーブルに置かれています。

このコードは、JTAG デバッグポート (4~5 本のピン)、SW デバッグポート (2 本のピン)、またはユーザソフトウェアによってアクセスできます。

## 40.7 JTAG デバッグポート

標準的な JTAG ステートマシンは、4 ビット命令レジスタ (IR) と 5 個のデータレジスタを搭載しています (全詳細については、FPU搭載 Cortex<sup>®</sup>-M7 *Technical Reference Manual (TRM)* を参照し、参考として [セクション 40.2 : ARM<sup>®</sup> リファレンス資料](#) を参照してください)。

表 253. JTAG デバッグポートのデータレジスタ

IR[3:0]	データレジスタ	詳細
1111	BYPASS [1 ビット]	-
1110	IDCODE [32 ビット]	ID コード 0x06449041 (FPU 搭載 ARM <sup>®</sup> Cortex <sup>®</sup> -M7 の ID コード)
1010	DPACC [35 ビット]	<p>デバッグポートアクセスレジスタ</p> <p>デバッグポートを初期化し、デバッグポートレジスタへのアクセスを可能にします。</p> <ul style="list-style-type: none"> <li>- 入力データ転送時 :                      ビット 34:3 = DATA[31:0] = 書き込みリクエスト用に転送する 32 ビットデータ                      ビット 2:1 = A[3:2] = デバッグポートレジスタの 2 ビットアドレス                      ビット 0 = RnW = 読み出しリクエスト (1) または書き込みリクエスト (0)</li> <li>- 出力データ転送時 :                      ビット 34:3 = DATA[31:0] = 読み出しリクエストに続いて読み出される 32 ビットデータ                      ビット 2:0 = ACK[2:0] = 3 ビット確認応答 :                      010 = OK/FAULT                      001 = WAIT                      その他 = 予約済み</li> </ul> <p>A[3:2] ビットの説明については、<a href="#">表 254</a> を参照してください。</p>

表 253. JTAG デバッグポートのデータレジスタ (続き)

IR[3:0]	データレジスタ	詳細
1011	APACC [35 ビット]	<p>アクセスポートアクセスレジスタ アクセスポートを初期化し、アクセスポートレジスタへのアクセスを可能にします。</p> <ul style="list-style-type: none"> <li>- 入力データ転送時 : ビット 34:3 = DATA[31:0] = 書き込みリクエスト用にシフトインする 32 ビットデータ ビット 2:1 = A[3:2] = 2 ビットアドレス (サブアドレス AP レジスタ) ビット 0 = RnW = 読み出しリクエスト (1) または書き込みリクエスト (0)</li> <li>- 出力データ転送時 : ビット 34:3 = DATA[31:0] = 読み出しリクエストに続いて読み出される 32 ビットデータ ビット 2:0 = ACK[2:0] = 3 ビット確認応答 : 010 = OK/FAULT 001 = WAIT その他 = 予約済み</li> </ul> <p>次の項目の組み合わせとして、多くの AP レジスタ (AHB-AP を参照) をアドレス指定します。</p> <ul style="list-style-type: none"> <li>- シフトされた値 A[3:2]</li> <li>- DP SELECT レジスタの現在値</li> </ul>
1000	ABORT [35 ビット]	<p>アポートレジスタ</p> <ul style="list-style-type: none"> <li>- ビット 31:1 = 予約済み</li> <li>- ビット 0 = DAPABORT : DAP アポートを生成するには 1 を書き込みます</li> </ul>

表 254. シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ

アドレス	A[3:2] の値	説明
0x0	00	予約済みであり、リセット値に保持する必要があります。
0x4	01	DP CTRL/STAT レジスタ：次の目的で使用されます。 - システム/デバッグパワーアップのリクエスト - AP アクセス用の転送動作の設定 - プッシュ比較/プッシュ検証動作の制御 - ステータスフラグ（オーバーラン、パワーアップの確認応答）の読み出し
0x8	10	DP SELECT レジスタ：現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウの選択に使用されます。 - ビット 31:24 (APSEL)：現在の AP を選択します。 - ビット 23:8：予約済み - ビット 7:4 (APBANKSEL)：現在の AP でアクティブな 4 ワードレジスタウィンドウを選択します。 - ビット 3:0：予約済み
0xC	11	DP RDBUFF レジスタ：一連の動作のあとで（新しい JTAG-DP 動作をリクエストせずに）、デバッガが最終結果を得られるようにします。

## 40.8 SW デバッグポート

### 40.8.1 SW プロトコルの概要

この同期式シリアルプロトコルでは、次の 2 個のピンを使用します。

- SWCLK：ホストからターゲットへのクロック
- SWDIO：双方向

このプロトコルでは、2 バンクのレジスタ（DPACC レジスタと APACC レジスタ）の読み出し/書き込みが可能です。

ビットは、ワイヤ上を LSB ファーストで転送されます。

SWDIO を双方向管理するには、ボード上でラインがプルアップされている必要があります（100 KΩ が ARM® の推奨値）。

プロトコルで SWDIO の方向が変化するたびに、ラインがホストからもターゲットからも駆動されないターンアラウンド時間が挿入されます。このターンアラウンド時間のデフォルトは 1 ビット時間ですが、SWCLK 周波数の設定によって調整できます。

### 40.8.2 SW プロトコルシーケンス

各シーケンスは 3 つのフェーズで構成されます。

1. ホストによって送信されるパケットリクエスト（8 ビット）
2. ターゲットによって送信される確認応答（3 ビット）
3. ホストまたはターゲットによって送信されるデータ転送フェーズ（33 ビット）



表 255. パケットリクエスト (8 ビット)

ビット	名前	説明
0	Start	“1”である必要があります。
1	APnDP	0 : DP アクセス 1 : AP アクセス
2	RnW	0 : 書き込みリクエスト 1 : 読み出しリクエスト
4:3	A[3:2]	DP/AP レジスタのアドレスフィールド (表 254を参照)
5	Parity	先行するビットの 1 ビットパリティ
6	STOP	0
7	Park	ホストによって駆動されません。プルアップの効果で、ターゲットには「1」と読み出される必要があります。

DPACC レジスタと APACC レジスタの詳細については、FPU 搭載 Cortex®-M7 TRM を参照してください。

パケットリクエストの後には、必ずホストもターゲットもラインを駆動しないターンアラウンド時間 (デフォルトでは 1 ビット) が続きます。

表 256. ACK 応答 (3 ビット)

ビット	名前	説明
0~2	ACK (確認応答)	001 : FAULT 010 : WAIT 100 : OK

読み出しトランザクションの場合や、受信した ACK 応答が“WAIT” または“FAULT” の場合にのみ、ACK 応答の後にターンアラウンド時間が続く必要があります。

表 257. データ転送 (33 ビット)

ビット	名前	説明
0~31	WDATA または RDATA	書き込み/読み出しデータ
32	Parity	32 データビットの 1 ビットパリティ

読み出しトランザクションの場合にのみ、データ転送の後にターンアラウンド時間が続く必要があります。

## 40.8.3 SW-DP ステートマシン (リセット、アイドル状態、ID コード)

SW-DP のステートマシンには、SW-DP を識別する内部 ID コードがあります。これは JEP-106 規格に準じています。この ID コードは、ARM<sup>®</sup> のデフォルトコードであり、**0x5BA02477** (FPU 搭載 Cortex<sup>®</sup>-M7 に対応) がセットされています。

**注:** **SW-DP ステートマシンは、ターゲットがこの ID コードを読み出すまで非アクティブであることに注意してください。**

- パワーオンリセット後、または DP が JTAG から SWD に切り替えられた後、またはラインが 50 サイクルを超えてハイレベルにあった後では、SW-DP ステートマシンは RESET 状態になります。
- リセット状態のあと、ラインが 2 サイクル以上の間ローレベルであれば、SW-DP ステートマシンはアイドル状態になります。
- リセット状態のあとは、まずアイドル状態に入り、次に DP-SW ID CODE レジスタの読み出しアクセスを行う **必要があります**。そうしないと、ターゲットは、他のトランザクションに対して ACK 応答の“FAULT”を発行します。

SW-DP ステートマシンの詳細については、FPU 搭載 Cortex<sup>®</sup>-M7 TRM および CoreSight Components Technical Reference Manualを参照してください。

## 40.8.4 DP と AP の読み出し/書き込みアクセス

- DP への読み出しアクセスはポストされません。つまり、ターゲットは、ACK 応答が“OK”の場合はただちに応答し、ACK 応答が“WAIT”の場合は遅れて応答します。
- AP への読み出しアクセスはポストされます。つまり、アクセスの結果は次の転送時に返されます。次のアクセスが AP アクセスでない場合、結果を得るには DP-RDBUFF レジスタを読み出す必要があります。  
AP 読み出しアクセスが成功したかどうかを判断するため、DP-CTRL/STAT レジスタの READOK フラグは、AP 読み出しアクセスまたは RDBUFF 読み出しリクエストのたびに更新されます。
- SW-DP は、DP と AP の両方の書き込みに使用できる書き込みバッファを実装しているため、たとえ他のトランザクションが未処理であっても、書き込み動作を受け付けることができます。書き込みバッファがフルのとき、ターゲットの ACK 応答は“WAIT”です。例外として、IDCODE 読み出し、CTRL/STAT 読み出し、または ABORT 書き込みは、書き込みバッファがフルであっても受け付けられます。
- 非同期クロックドメイン SWCLK と HCLK によって、書き込みを内部的に有効にするには、書き込みトランザクション後 (パリティビット後) に SWCLK の 2 サイクルが余分に必要となります。これらのサイクルは、ラインをローレベルに駆動している間 (アイドル状態) に適用してください。  
これは、パワーアップリクエストのために CTRL/STAT の書き込みを行う際に特に重要です。パワーアップを必要とする次のトランザクションがただちに発生すると、そのトランザクションは失敗します。

## 40.8.5 SW-DP レジスタ

これらのレジスタへのアクセスは、APnDP = 0 のときに開始されます。

表 258. SW-DP レジスタ

A[3:2]	読み出し/ 書き込み	SELECT レジスタの CTRLSEL ビット	レジスタ	注
00	読み出し	-	IDCODE	製造者コードは ST 社のコードではありません。 0x5BA02477 (SW-DP を識別)。
00	書き込み	-	ABORT	-
01	読み出し/ 書き込み	0	DP CTRL/STAT	用途は以下のとおりです。 - システム/デバッグパワーアップのリクエスト - AP アクセス用の転送動作の設定 - ブッシュ比較/ブッシュ検証動作の制御 - ステータスフラグ (オーバーラン、パワーアップの確認応答) の読み出し
01	読み出し/ 書き込み	1	WIRE CONTROL	物理的なシリアルポートプロトコルの設定 (ターンアラウンド時間など) を行います。
10	読み出し	-	READ RESEND	元の AP 転送を反復しなくても、破壊されたデバッグ転送からの読み出しデータの復旧を可能にします。
10	書き込み	-	SELECT	現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウを選択します。
11	読み出し/ 書き込み	-	読み出し バッファ	AP アクセスはポストされるため、この読み出しバッファは効果的です (AP 読み出しリクエストの結果は、次の AP トランザクションで取得できる)。 この読み出しバッファは、新しいトランザクションを開始することなく、前回の読み出しの結果として AP から出力されるデータをキャプチャします。

## 40.8.6 SW-AP レジスタ

これらのレジスタへのアクセスは、APnDP = 1 のときに開始されます。

次の項目の組み合わせとして、多くの AP レジスタ (AHB-AP を参照) をアドレス指定します。

- シフトされた値 A[3:2]
- DP SELECT レジスタの現在値

## 40.9 AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効

**機能 :**

- システムアクセスはプロセッサステータスから独立しています。
- SW-DP または JTAG-DP が AHB-AP にアクセスします。
- AHB-AP はバスマトリックスの AHB マスタです。したがって、AHB-AP は ICode バスを除くすべてのデータバス (Dcode バス、システムバス、内部/外部の PPB バス) にアクセスできます。
- ビットバンドトランザクションがサポートされます。
- AHB-AP トランザクションは FPB を迂回します。

32 ビット AHB-AP レジスタのアドレスは 6 ビット幅 (最大 64 ワードまたは 256 バイト) であり、以下の構成となっています。

- c) ビット [7:4] = DP\_SELECT レジスタのビット [7:4] APBANKSEL
- d) ビット [3:2] = SW-DP 用の 35 ビットパケットリクエストの 2 ビットアドレス A(3:2)

FPU 搭載 Cortex®-M7 の AHB-AP は、9 個の 32 ビットレジスタを内蔵しています。

**表 259. FPU 搭載Cortex®-M7 AHB-AP レジスタ**

アドレス オフセット	レジスタ名	注
0x00	AHB-AP 制御およびステータスワード	AHB インタフェースを通じて転送を設定および制御します (サイズ、hprot、現在の転送のステータス、アドレスインクリメントタイプ)。
0x04	AHB-AP 転送アドレス	-
0x0C	AHB-AP データ読み出し/書き込み	-
0x10	AHB-AP バンクデータ 0	転送アドレスレジスタを書き換えずに、4 個のアラインドデータワードを直接配置します。
0x14	AHB-AP バンクデータ 1	
0x18	AHB-AP バンクデータ 2	
0x1C	AHB-AP バンクデータ 3	
0xF8	AHB-AP デバッグ ROM アドレス	デバッグインタフェースのベースアドレス
0xFC	AHB-AP ID レジスタ	-

詳細については、FPU 搭載 Cortex®-M7 TRM を参照してください。

## 40.10 コアデバッグ

コアデバッグはコアデバッグレジスタを通じてアクセスされます。これらのレジスタへのデバッグアクセスには、Advanced High-performance Bus (AHB-AP) ポートを使用します。プロセッサは、内部のプライベートペリフェラルバス (PPB) を介してこれらのレジスタに直接アクセスできます。

コアデバッグは 4 個のレジスタから構成されています。

表 260. コアデバッグレジスタ

レジスタ	説明
DHCSR	32 ビットのデバッグ停止制御/ステータスレジスタ : レジスタプロセッサの状態についてのステータス情報を提供し、コアデバッグを有効にし、プロセッサの停止とステップ実行を行います。
DCRSR	17 ビットのコアデバッグレジスタセクタレジスタ : データの転送先または転送元となるプロセッサレジスタを選択します。
DCRDR	32 ビットのコアデバッグレジスタデータレジスタ : DCRSR (セクタ) レジスタによって選択されたプロセッサとの間でレジスタの読み出しおよび書き込みに使用するデータを保持します。
DEMCR	32 ビットのコアデバッグ例外/モニタ制御レジスタ : ベクタキャッチとデバッグモニタの制御を行います。このレジスタには、TRACE を使用できるようにする <i>TRCENA</i> というビットがあります。

**注 :** **重要 :** これらのレジスタは、システムリセットによってはリセットされません。パワーオンリセットによってのみリセットされます。

詳細については、FPU 搭載 Cortex<sup>®</sup>-M7 の TRM を参照してください。

リセット時に停止させるには、以下の手順が必要です。

- デバッグ例外/モニタ制御レジスタのビット 0 (VC\_CORRESET) を有効にします。
- デバッグ停止制御/ステータスレジスタのビット 0 (C\_DEBUGEN) を有効にします。

## 40.11 システムリセット中のデバッグホスト接続機能

STM32F75xxx および STM32F74xxx MCU のリセットシステムは、次のリセットソースから構成されます。

- POR (パワーオンリセット) : パワーアップのたびに RESET をアサートします。
- 内部ウォッチドッグリセット
- ソフトウェアリセット
- 外部リセット

FPU 搭載 Cortex<sup>®</sup>-M7 では、デバッグ部のリセット (一般に PORRESETn) とそれ以外のリセット (SYSRESETn) が区別されます。

こうすれば、リセットベクタをフェッチするときにコアを停止させるようコアデバッグレジスタをプログラミングすることで、デバッグはシステムリセット中に接続が可能になります。これによってホストはシステムリセットを解除でき、コアは命令を実行しなくても、ただちに停止します。さらに、システムリセット中にデバッグ機能をプログラミングできます。

**注 :** システムリセット中にデバッグホストを接続する (リセットベクタにブレークポイントを設定する) ことを強く推奨します。

## 40.12 FPB (フラッシュパッチブレークポイント)

一般に、Cortex-M のアーキテクチャにおける FPB ユニットの機能は以下の通りです。

- ハードウェアブレークポイントを実装します。
- コード空間からシステム空間に、コードとデータをパッチします。この機能によって、コードメモリ空間にあるソフトウェアバグの訂正が可能になることがあります。

この場合、ソフトウェアパッチとハードウェアブレークポイントは同時に使用できません。

しかし、ペリカン FPB には次のようないくつかの大きな変化があります。

- フラッシュパッチはサポートされていません (FP\_REMAP レジスタがない)。
- コンパレータはすべて命令アドレス用です (最大 8 つの命令ブレークポイント)。
- ブレークポイントコンパレータのプログラマモデルは機能が拡張され、全アドレス範囲でハードウェアブレークポイントが可能となっています。

## 40.13 DWT (データウォッチポイントトリガ)

DWT ユニットの 4 個のコンパレータから構成されており、次の用途に設定できます。

- ハードウェアウォッチポイント
- ETM へのトリガ
- PC サンプラ
- データアドレスサンプラ

DWT は、プロファイル情報を提供することもできます。このため、次の値を得るためのカウンタにアクセスできます。

- クロックサイクル数
- フォールドされた命令数
- ロードストアユニット (LSU) の動作数
- スリープサイクル数
- CPI (命令あたりのクロック数)
- 割込みオーバーヘッドの回数

## 40.14 ITM (計測トレースマクロセル)

### 40.14.1 概要

ITM は、アプリケーション駆動のトレースソースであり、*printf* 方式のデバッグをサポートすることでオペレーティングシステム (OS) やアプリケーションのイベントをトレースし、システムの診断情報を発行します。ITM がパケットとして発行するトレース情報は、次のように生成できます。

- **ソフトウェアトレース**: ソフトウェアは、ITM スティムラレジスタに直接書き込みを行って、パケットを発行できます。
- **ハードウェアトレース**: DWT がパケットを生成し、ITM がそれを発行します。
- **タイムスタンプ**: タイムスタンプはパケットを基準にして発行されます。ITM は、タイムスタンプを生成するための 21 ビットカウンタを内蔵しています。カウンタは、FPU 搭載 Cortex<sup>®</sup>-M7 クロックまたは シリアルワイヤ ビューア (SWV) 出力のビットクロックレートによって駆動されます。

ITM によって発行されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから (TPIU を参照)、完全なパケットシーケンスをデバッガホストに出力します。

ITM をプログラミングまたは使用するには、その前に、デバッグ例外/モニタ制御レジスタの TRCEN ビットを有効にする必要があります。

### 40.14.2 タイムスタンプパケット、同期およびオーバーフローパケット

タイムスタンプパケットは、タイムスタンプ情報や一般的な制御および同期情報をコード化します。そのために使用される 21 ビットのタイムスタンプカウンタ (プリスケアラ付きの場合もあります) は、タイムスタンプパケットが発行されるたびにリセットされます。このカウンタは、CPU クロックまたは SWV クロックによって駆動できます。

同期パケットは 6 バイトの値 0x80\_00\_00\_00\_00\_00 から構成され、TPIU には 00 00 00 00 00 80 として LSB ファーストで発行されます。

同期パケットは、タイムスタンプ制御のためのパケットで、DWT トリガのたびに発行されます。

このため、DWT 制御レジスタの CYCCNTENA (ビット 0) をセットして、ITM をトリガするように DWT を設定する必要があります。さらに、ITM トレース制御レジスタのビット 2 (SYNCENA) もセットする必要があります。

**注：** **SYNENA ビットがセットされていない場合、DWT は TPIU への同期トリガを生成し、TPIU 同期パケットのみを送信し、ITM 同期パケットは送信しません。**

オーバーフローパケットは、データの書き込み時に FIFO がフルであったことを示す特殊なタイムスタンプパケットで構成されます。

表 261. 主な ITM レジスタ

アドレス	レジスタ	詳細
@E0000FB0	ITM ロックアクセス	他の ITM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
@E0000E80	ITM トレース制御	ビット 31-24 = 常に 0
		ビット 23 = ビジー
		ビット 22-16 = トレースデータのソースを識別する 7 ビットの ATB ID
		ビット 15-10 = 常に 0
		ビット 9:8 = TSPrescale = タイムスタンププリスケール
		ビット 7-5 = 予約済み
		ビット 4 = SWOENA = SWV 動作 (タイムスタンプカウンタを SWV クロックによって駆動) を有効にします。
		ビット 3 = DWTENA : DWT スティムラスを有効にします。
		ビット 2 = SYNCENA : DWT が同期トリガを生成して TPIU が同期パケットを発行できるようにするには、このビットを 1 にする必要があります。
		ビット 1 = TSENA (タイムスタンプ有効)
ビット 0 = ITMENA : ITM のグローバル有効ビット		
@E0000E40	ITM トレース特権	ビット 3 : ポート 31:24 のトレースを有効にするマスク
		ビット 2 : ポート 23:16 のトレースを有効にするマスク
		ビット 1 : ポート 15:8 のトレースを有効にするマスク
		ビット 0 : ポート 7:0 のトレースを有効にするマスク
@E0000E00	ITM トレース有効	各ビットは、トレースを生成するために対応するスティムラスポートを有効にします。
@E0000000- E000007C	スティムラスポート レジスタ 0-31	選択されたスティムラスポート (32 個まで) に 32 ビットデータを書き込んでトレース出力します。

## 設定例

TPIU に単純な値を出力するには、

- TPIU を設定し、DBGMCU\_CR を設定することで TRACE I/O を割り当てます ([セクション 40.17.2: TRACE ピンの割当て](#) および [セクション 40.16.3: デバッグMCU 設定レジスタ](#) を参照)。
- ITM ロックアクセスレジスタに 0xC5ACCE55 を書き込んで、ITM レジスタへの書き込みアクセスをロック解除します。
- ITM トレース制御レジスタに 0x00010005 を書き込んで、同期を有効にした状態で ITM を有効にし、ATB ID を 0x00 以外の値にします。
- ITM トレース有効レジスタに 0x1 を書き込んで、スティムラサポート 0 を有効にします。
- ITM トレース特権レジスタに 0x1 を書き込んで、スティムラサポート 7:0 をマスク解除します。
- スティムラサポートレジスタ 0 に出力値を書き込みます。これはソフトウェアで行うことができます (printf 機能を使用)。

## 40.15 ETM (組み込みトレースマクロセル)

### 40.15.1 概要

ETM はプログラム実行の再構築を有効にします。データはデータウォッチポイントおよびトレース (DWT) コンポーネントまたは計測トレースマクロセル (ITM) を使用してトレースされます。一方、命令は組み込みトレースマクロセル (ETM) を使用してトレースされます。

ETMは、情報をパケットとして送信し、埋め込みリソースによってトリガされます。これらのリソースは個別にプログラミングする必要があり、トリガソースの選択にはトリガイベントレジスタ (0xE0041008) を使用します。イベントは、単一イベント (アドレスコンパレータからのアドレスマッチ) または 2 つのイベント間の論理式とすることができます。トリガソースは、DWT モジュールの 4 番目のコンパレータの 1 つで、次のイベントを監視できます。

- クロックサイクルマッチング
- データアドレスマッチング

トリガリソースの詳細については、[セクション 40.13: DWT \(データウォッチポイントトリガ\)](#) を参照してください。

ETM によって送信されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから ([セクション 40.17: ペリカン TPIU \(トレースポートインタフェースユニット\)](#) を参照)、完全なパケットシーケンスをデバッグホストに出力します。

**注:** *注意: Cortex-M7 ETM は ARM ETM Architecture V4 に適合していますが、このプログラミングモデルは Cortex-M4 ETM のプログラミングモデル (ETM Architecture V3.5) と後方互換性がありません。*

### 40.15.2 信号プロトコル、パケットタイプ

これについては、第 6 章の ETM V4 Architecture Specification (IHI0064B) で説明されています。

## 40.15.3 主な ETM レジスタ

レジスタの詳細については、Pelican ETM Technical Reference Manual (DDI0494-2a) および ETM v4 Architecture Specification (IHI0064B) を参照してください。

## 40.15.4 設定例

TPIU に単純な値を出力するには、

- TRACE I/O の設定 : STM32F75xxx および STM32F74xxx デバッグ設定レジスタ (DBGMCU\_CR) の TRACE\_CLKINEN を有効にします。
- E000EDFC 01000000 への書き込み (SCS) : TRCENA をセットします。そうしないと、トレースレジスタへはアクセスできません。
- E00400F0 00000000 への書き込み (TPIU) : 同期ポートモードを選択します。
- E0040004 00000008 への書き込み (TPIU) : TPIU ポートサイズ = 4 を選択します。
- E0001020 002002CA への書き込み (WT) : PC 一致コンパレータ (PC = 0x2002CA)
- E0001024 00000000 への書き込み (DWT) : コンパレータへのマスク適用はしません。
- E0001028 00000008 への書き込み (DWT) : PC 一致時に ETM へのトリガを行います。

ETM :

- E0041004 00000000 への書き込み : ETM を無効化します。
- E004100C 00000003 からの読み出し : ETM はアイドル状態でなければなりません。
- E0041040 00000002 への書き込み : 命令トレースソース ID = 0x2
- E0041080 00000001 への書き込み : イベントを有効する ViewInst のリソースは「常に真」です。
- E004108C 000000FF への書き込み : 開始用のプロセッサのコンパレータを選択します。  
pc\_match0 (=> DWT 一致)
- E0041004 00000001 への書き込み : ETM を有効化します。

## 40.16 MCU デバッグコンポーネント (DBGMCU)

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

- 低電力モード
- ブレークポイントにおける、タイマ、ウォッチドッグ、I2C、および bxCAN のクロック制御
- TRACE ピンの割当て制御

### 40.16.1 低電力モードのデバッグサポート

低電力モードに入るには、WFI または WFE 命令を実行する必要があります。

MCU はいくつかの低電力モードを実装しており、CPU クロックを無効にしたり、CPU の消費電力を低減したりすることができます。

デバッグセッション中には、コアは FCLK や HCLK をオフにすることはできません。デバッグ時のデバッグ接続に必要なため、これらをアクティブな状態に保つ必要があります。MCU は、ユーザが低電力モードでソフトウェアをデバッグするための特殊な手段を備えています。

このため、デバッグホストは、最初にいくつかのデバッグ設定レジスタをセットして、低電力モード動作を変更する必要があります。

- SLEEP モードでは、DBGMCU\_CR レジスタの DBG\_SLEEP ビットをデバッグによって事前にセットする必要があります。これによって、HCLK には FCLK と同じクロックが供給されます (システムクロックはソフトウェアによって事前に設定されています)。
- STOP モードでは、DBG\_STOP ビットをデバッグによって事前にセットする必要があります。これによって、内部 RC オシレータが、STOP モードで FCLK と HCLK にクロックを供給できません。

## 40.16.2 タイマ、ウォッチドッグ、bxCAN、および I<sup>2</sup>C のデバッグサポート

ブレークポイントにおいては、以下に示すタイマのカウンタやウォッチドッグの動作方法を選択する必要があります。

- ブレークポイントの中でもカウントを継続できます。この動作は、たとえば、PWM がモータを制御しているときに一般的に必要です。
- ブレークポイントの中でカウントを停止できます。この動作はウォッチドッグ用に必要です。

bxCAN の場合、ブレークポイントにおいては受信レジスタの更新をブロックするように選択できます。

I<sup>2</sup>C の場合、ブレークポイントにおいては SMBUS タイムアウトをブロックするように選択できます。

## 40.16.3 デバッグ MCU 設定レジスタ

このレジスタを使用して、デバッグ中に MCU を設定できます。次の設定が可能です。

- 低電力モードのサポート
- タイマおよびウォッチドッグカウンタのサポート
- bxCAN 通信のサポート
- TRACE ピンの割当て

この DBGMCU\_CR は、アドレス 0xE0042004 にある外部 PPB バスに配置されます。

このレジスタは PORESET によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

デバッグホストがこれらの機能をサポートしない場合でも、ユーザソフトウェアによってこれらのレジスタへの書き込みが可能です。

## 40.16.4 DBGMCU\_CR レジスタ

アドレス : 0xE004 2004

32 ビットアクセスのみサポートされます。

POR リセット : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TRACE_	TRACE_	Res.	Res.	DBG_	DBG_	DBG_								
								MODE	CLKINE			STANDB	STOP	SLEEP	
								[1:0]	N			Y			
								rw	rw	rw		rw	rw	rw	

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:5 **TRACE\_MODE[1:0]** および **TRACE\_CLKINEN** : TRACE クロックおよびピンの割り当て制御

- TRACE\_CLKINEN = 0 の場合

TRACE\_MODE=xx : TRACE 出力は無効です (同期と非同期の両方)。

- TRACE\_CLKINEN = 1 の場合

- TRACE\_MODE[1:0] = 00 : 非同期トレースインタフェースはパッドレベルで有効 (シリアルワイヤモードを使用する場合、TRACESWO は TDO パッドでのみ使用可能) / 同期トレースインタフェースは有効です。

- TRACE\_MODE[1:0] ≠ 00 : 非同期トレースインタフェースは無効 / 同期トレースインタフェースは有効です。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG\_STANDBY** : デバッグ STANDBY モード

0 : (FCLK=オフ、HCLK=オフ) デジタル部全体が電源オフになります。

ソフトウェアから見て、STANDBY モードから抜けることは、リセットベクタのフェッチと同じになります (ただし、いくつかのステータスビットは MCU が STANDBY モードから再開していることを示す)。

1 : (FCLK=オン、HCLK=オン) この場合、デジタル部は電源オフ状態ではなく、FCLK と HCLK は引き続きアクティブ状態の内部 RC オシレータから供給されます。さらに、MCU は STANDBY モード中にシステムリセットを生成するため、STANDBY モードから抜けることはリセットからのフェッチと同じになります。

ビット 1 **DBG\_STOP** : デバッグ STOP モード

0 : (FCLK=オフ、HCLK=オフ) STOP モードでは、クロックコントローラがすべてのクロック (HCLK と FCLK を含む) を無効にします。STOP モードから抜けると、クロック設定はリセット後の場合と同じになります (CPU は 8 MHz の内部 RC オシレータ (HSI) から供給されます)。したがって、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります。

1 : (FCLK=オン、HCLK=オン) この場合、STOP モードに入ると、FCLK と HCLK は STOP モードでもアクティブ状態の内部 RC オシレータから供給されます。STOP モードから抜けるとき、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります (DBG\_STOP=0 の場合と同様です)。

ビット 0 **DBG\_SLEEP** : デバッグ SLEEP モード

0 : (FCLK=オン、HCLK=オフ) SLEEP モードでは、FCLK はソフトウェアによって事前に設定されたシステムクロックによって駆動され、HCLK は無効にされます。

SLEEP モードでは、クロックコントローラの設定はリセットされず、事前にプログラミングされた状態のままです。したがって、SLEEP モードから抜けるときに、ソフトウェアでクロックコントローラを再設定する必要はありません。

1 : (FCLK=オン、HCLK=オン) この場合、SLEEP モードに入ると、HCLK には FCLK と同じクロック (ソフトウェアによって事前に設定されたシステムクロック) が供給されます。

## 40.16.5 デバッグ MCU APB1 フリーズレジスタ (DBGMCU\_APB1\_FZ)

DBGMCU\_APB1\_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB2 ペリフェラルと関係のあるレジスタで、アドレス 0xE004 2008 にある外部 PPB バスに配置されています。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

アドレス : 0xE004 2008

32 ビットアクセスのみサポートされます。

パワーオンリセット (POR) : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	DBG_CAN2_STOP	DBG_CAN1_STOP	DBG_I2C4_SMBUS_TIMEOUT	DBG_I2C3_SMBUS_TIMEOUT	DBG_I2C2_SMBUS_TIMEOUT	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.
					rW	rW	rW	rW	rW	rW					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	DBG_LP1TIM1_STOP	DBG_TIM14_STOP	DBG_TIM13_STOP	DBG_TIM12_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

- ビット 31:27 予約済みであり、リセット値に保持する必要があります。
- ビット 26 **DBG\_CAN2\_STOP** : コア停止時にデバッグ CAN2 は停止  
0 : 通常モードと同じ動作です。  
1 : CAN2 受信レジスタは停止状態です。
- ビット 25 **DBG\_CAN1\_STOP** : コア停止時にデバッグ CAN2 は停止  
0 : 通常モードと同じ動作です。  
1 : CAN2 受信レジスタは停止状態です。
- ビット 24 **DBG\_I2C4\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止  
0 : 通常モードと同じ動作です。  
1 : SMBUS タイムアウトは凍結されます。
- ビット 23 **DBG\_I2C3\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止  
0 : 通常モードと同じ動作です。  
1 : SMBUS タイムアウトは凍結されます。
- ビット 22 **DBG\_I2C2\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止  
0 : 通常モードと同じ動作です。  
1 : SMBUS タイムアウトは凍結されます。
- ビット 21 **DBG\_I2C1\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止  
0 : 通常モードと同じ動作です。  
1 : SMBUS タイムアウトは凍結されます。
- ビット 20:13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **DBG\_IWDG\_STOP** : コア停止時にデバッグ独立型ウォッチドッグは停止  
0 : コアが停止しても独立型ウォッチドッグカウンタのクロックは継続されます。  
1 : コア停止時に独立型ウォッチドッグカウンタのクロックは停止します。
- ビット 11 **DBG\_WWDG\_STOP** : コア停止時にデバッグウィンドウ型ウォッチドッグは停止  
0 : コアが停止してもウィンドウ型ウォッチドッグカウンタのクロックは継続されます。  
1 : コア停止時にウィンドウ型ウォッチドッグカウンタのクロックは停止します。
- ビット 10 **DBG\_RTC\_STOP** : コア停止時は RTC 停止  
0 : コアが停止した場合も RTC カウンタのクロックは継続されます。  
1 : コア停止時に RTC カウンタのクロックは停止します。
- ビット 9 **DBG\_LPTIM1\_STOP** : コア停止時に LPTIM1 カウンタは停止  
0 : コアが停止しても LPTIM1 カウンタのクロックは供給されます。  
1 : コア停止時に LPTIM1 カウンタのクロックは停止します。
- ビット 8:0 **DBG\_TIMx\_STOP** : コア停止時に TIMx カウンタは停止 (x=2, 7, 12..14)  
0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。  
1 : コア停止時に当該タイマカウンタのクロックは停止します。

## 40.16.6 デバッグ MCU APB2 フリーズレジスタ (DBGMCU\_APB2\_FZ)

DBGMCU\_APB2\_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB2 ペリフェラルと関係のあるレジスタで、

アドレス 0xE004 200C にある外部 PPB バスに配置されています。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッガによって書き込みが可能です。

アドレス : 0xE004 200C

32 ビットアクセスのみサポートされます。

POR : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DBG_TIM11_STOP	DBG_TIM10_STOP	DBG_TIM9_S_TOP												
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DBG_TIM8_S_TOP	DBG_TIM1_S_TOP													
														rw	rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **DBG\_TIMx\_STOP** : コア停止時に TIMx カウンタは停止 (x=9..11)

0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。

1 : コア停止時に当該タイマカウンタのクロックは停止します。

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DBG\_TIM8\_STOP** : コア停止時に TIM8 カウンタは停止

0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。

1 : コア停止時に当該タイマカウンタのクロックは停止します。

ビット 0 **DBG\_TIM1\_STOP** : コア停止時に TIM1 カウンタは停止

0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。

1 : コア停止時に当該タイマカウンタのクロックは停止します。

## 40.17 ペリカン TPIU (トレースポートインタフェースユニット)

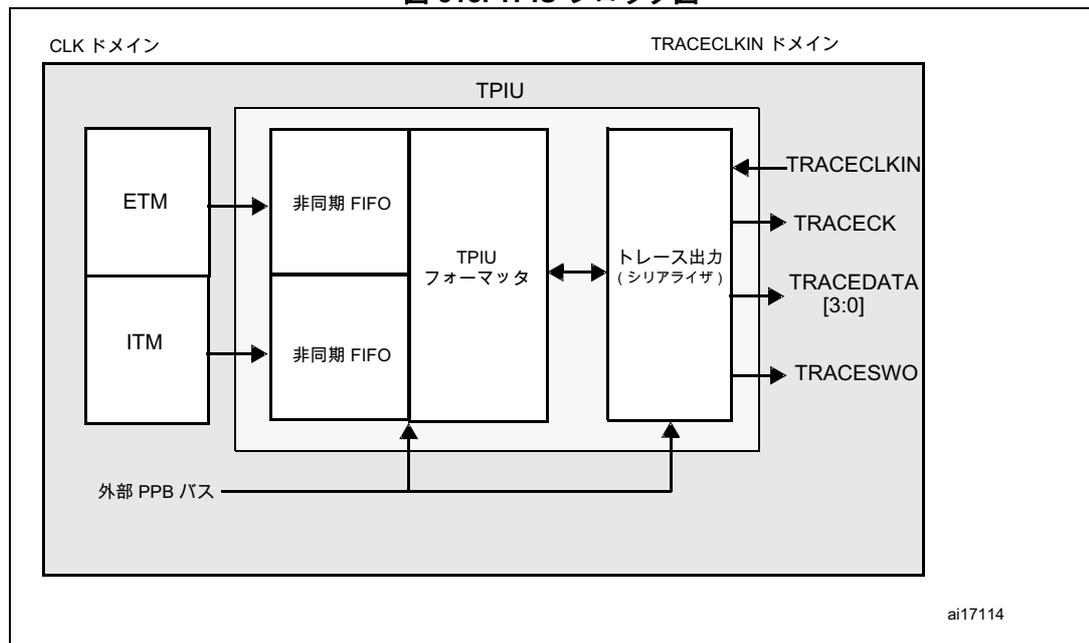
### 40.17.1 概要

TPIU は、ITM と ETM から受信するオンチップトレースデータ間のブリッジとして、および外部トレースキャプチャデバイスとしての役割を果たします。

出力データストリームによってカプセル化されたトレースソース ID は、トレースポートアナライザ (TPA) によってキャプチャされます。

コアには、特に低コストのデバッグ向けの単純な TPIU (特別バージョンの CoreSight TPIU から構成) が内蔵されています。

図 518. TPIU ブロック図



## 40.17.2 TRACE ピンの割当て

- 非同期モード  
非同期モードは、1 個のピンを追加で必要とし、すべてのパッケージで使用できます。非同期モードは、JTAG モードではなくシリアルワイヤモードを使用する場合にのみ使用できます。

表 262. 非同期 TRACE ピンの割当て

TPIU ピン名	TRACE 同期モード	
	タイプ	説明
TRACESWO	0	TRACE 非同期データ出力

- 同期モード  
同期モードは、データトレースサイズに応じて 2~6 本のピンを追加で必要とし、大型パッケージでのみ使用できます。さらに、JTAG モードとシリアルワイヤモードでも使用でき、非同期トレースよりも優れた帯域幅出力機能を提供します。

表 263. 同期 TRACE ピンの割当て

TPIU ピン名	TRACE 同期モード	
	タイプ	説明
TRACECK	0	TRACE クロック
TRACED[3:0]	0	TRACE 同期データ出力 1、2、または 4 とすることができます。

### TPIU TRACE ピンの割当て

デフォルトでは、これらのピンは割り当てられません。これらのピンを割り当てるには、TRACE\_CLKINEN ビットおよび TRACE\_MODE ビット (**MCU デバッグコンポーネント設定レジスタ**) をセットします。この設定はデバッグホストで行う必要があります。

さらに、割り当てるピン数は、トレースの設定（非同期/同期）によって異なります。

- **非同期モード**：1 本のピンが追加で必要です。
- **同期モード**：データトレースポートレジスタのサイズ（1、2、または 4）に応じて、2~5 本のピンが追加で必要です。
  - TRACECK
  - TRACED(0)、ポートサイズが 1、2、または 4 に設定された場合
  - TRACED(1)、ポートサイズが 2 または 4 に設定された場合
  - TRACED(2)、ポートサイズが 4 に設定された場合
  - TRACED(3)、ポートサイズが 4 に設定された場合

TRACE ピンを割り当てるには、デバッグホストは、デバッグ MCU 設定レジスタ (DBGMCU\_CR) の TRACE\_CLKINEN ビットと TRACE\_MODE[1:0] ビットをプログラムする必要があります。デフォルトでは、TRACE ピンは割り当てられません。

このレジスタは、外部 PPB に配置され、PORESET によってリセットされます（システムリセットではありません）。このレジスタは、システムリセット中にデバッグによる書き込みができます。

表 264. 柔軟性の高い TRACE ピン割り当て

DBGMCU_CR レジスタ		ピン割り当て:	TRACE IO ピンの割り当て						
TRACE_CLKINEN	TRACE_MODE [1:0]		JTDO/TRACE SWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]	
0	XX	トレースなし (デフォルト状態)	解放 <sup>(1)</sup>	-					
1	00	非同期トレース	TRACESWO	-	-	解放 (GPIO として使用可能)			
1	00 以外	同期トレース、1 ビット <sup>(2)</sup>	解放 <sup>(1)</sup>	TRACECK	TRACED[0]	-	-	-	
1	00 以外	同期トレース 2 ビット <sup>(2)</sup>		TRACECK	TRACED[0]	TRACED[1]	-	-	
1	00 以外	同期トレース 4 ビット <sup>(2)</sup>		TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]	

1. シリアルワイヤモードが使用されているときには解放されます。しかし、JTAG が使用されると、JTDO に割り当てられます。
2. TPIU レジスタのビット [3:0] (現在のポートサイズ) で選択します。

**注:** デフォルトでは、TPIU の TRACECLKIN 入力クロックは GND に接続されています。このクロックは、TRACE\_CLKINEN ビットがセットされてから 2 クロックサイクル後に HCLK に割り当てられません。

デバッグは、TPIU の SPP\_R (選択ピンプロトコル) レジスタの PROTOCOL[1:0] ビットに書き込むことによって、トレースモードをプログラミングする必要があります。

- PROTOCOL = 00 : トレースポートモード (同期モード)。
- PROTOCOL = 01 または 10 : シリアルワイヤ (マンチェスタまたは NRZ) モード (非同期モード)。  
デフォルト状態は 01 です。

次に、TPIU の CPSPS\_R (カレント同期化ポートサイズレジスタ) のビット [3:0] に書き込むことによって TRACE ポートサイズも設定します。

- 0x1 : 1 ピンの場合 (デフォルト状態)
- 0x2 : 2 ピンの場合
- 0x8 : 4 ピンの場合

## 40.17.3 TPIU フォーマッタ

このフォーマッタの目的は、ETM と ITM の両方から受け取る可能性のあるトレースデータを格納する 128 ビットのフレームを構築し、トレースパケットとエミッタ間の相関関係をトレースアナライザレベルにすることです。

フォーマッタプロトコルは、データを 16 バイトフレームで出力します。

- 7 バイトのデータ
- 8 バイトの混合バイトであり、以下の構成になっています。
  - 1 ビット (LSB) : データバイト (“0”) または ID バイト (“1”) であることを示します。
  - 7 ビット (MSB) : データまたはソース ID トレースの変更とすることができます。
- 1 バイトの補助ビット : 各ビットは 8 つの混合バイトのいずれかに対応します。
  - 対応するバイトがデータであった場合、このビットはデータのビット 0 を示します。
  - 対応するバイトが ID 変更であった場合、このビットはその ID 変更が有効になるタイミングを示します。

**注 :** *詳細については、ARM® CoreSight Architecture Specification v1.0 (ARM® IHI 0029B) を参照してください。*

## 40.17.4 TPIU フレーム同期パケット

TPIU は 2 種類の同期パケットを生成できます。

- フレーム同期パケット (フルワード同期パケット)  
これはワード 0x7F\_FF\_FF\_FF (LSB ファーストで発行) から構成されます。ID ソースコード 0x7F が使用されていない場合、このシーケンスは他のタイミングでは発生できません。  
このパケットはフレーム *間* で定期的に出力されます。  
連続モードでは、同期フレームが検出されると、TPA がこれらすべてのフレームを破棄する必要があります。
- ハーフワード同期パケット  
これはハーフワード 0x7F\_FF (LSB ファーストで発行) から構成されます。  
このパケットはフレーム *間または内* で定期的に出力されます。  
これらのパケットは連続モードでのみ生成され、TPA は TRACE ポートが IDLE モードである (TRACE はキャプチャされない) ことを検出できるようになります。TPA によって検出されたパケットは、破棄する必要があります。

## 40.17.5 同期フレームパケットの送信

コアの TPIU には、同期カウンタレジスタは実装されていません。したがって、同期トリガは DWT によってのみ生成できます。DWT 制御レジスタ (SYNCTAP[11:10] ビット) と DWT カレント PC サンプラサイクルカウントレジスタの説明を参照してください。

TPIU フレーム同期パケット (0x7F\_FF\_FF\_FF) は次の場合に発行されます。

- 毎回の TPIU リセット解除後。このリセットは、TRACECLKIN クロックの立ち上がりエッジに同期して解除されます。つまり、DBGMCU\_CFG レジスタの TRACE\_CLKINEN ビットがセットされると、このパケットが送信されます。この場合、ワード 0x7F\_FF\_FF\_FF の後に、フォーマットされたパケットは続きません。

- 毎回の DWT トリガ時 (DWT は事前に設定されているものとします)。次の 2 つの場合があります。
  - ITM の SYNENA ビットがリセットされた場合、ワード 0x7F\_FF\_FF\_FF だけが発行され、フォーマットされたストリームが続くことはありません。
  - ITM の SYNENA ビットがセットされた場合、TPIU によってフォーマット (トレースソース ID を追加) された ITM 同期パケットが続きます (0x80\_00\_00\_00\_00\_00)。

## 40.17.6 同期モード

トレースデータの出力サイズは 4、2、または 1 ピンから選択できます (TRACED[3:0])。

出力クロックはデバッグに出力されません (TRACECK)。

なお、TRACECLKIN は内部で駆動され、TRACE が使用される場合にのみ HCLK に接続されます。

**注:** *同期モードでは、必ずしも安定したクロック周波数を提供する必要はありません。*

TRACE I/O (TRACECK を含む) は TRACLKIN (HCLK と同等) の立ち上がりエッジで駆動されません。したがって、TRACECK の出力周波数は HCLK/2 になります。

## 40.17.7 非同期モード

これは 1 ピン (非同期出力ピン TRACESWO) のみを使用してトレースを出力する低コストな方法ですが、明らかにその帯域幅は制限されます。

一般に、シングル IO トレースモードは ITM トレース出力に適しています。また、非同期トレースの場合はフォーマッタが無効化されていますので、ETM と ITM のトレースストリームのマージ (合成) はできません。

SW-DP ピンを使用する場合、TRACESWO ピンは JTDO ピンと多重化されます。このように、この機能はすべての STM32F75xxx および STM32F74xxx パッケージで使用できます。

この非同期モードでは、TRACECLKIN に一定の周波数が要求されます。標準的な UART (NRZ) のキャプチャ機構では、5% の精度が必要です。マンチェスタエンコード方式では、10% までの誤差が許されます。

## 40.17.8 STM32F75xxx および STM32F74xxx内の TRACECLKIN 接続

STM32F75xxx および STM32F74xxxでは、この TRACECLKIN 入力は内部で HCLK に接続されます。つまり、非同期トレースモードでは、アプリケーションは、CPU 周波数が安定しているタイムフレームの使用のみに制限されています。

**注:** *重要: 非同期トレースを使用する場合は、以下のことに注意してください。*

*STM32F75xxx および STM32F74xxx MCU のデフォルトクロックは、内部 RC オシレータです。リセット中の周波数はリセット解除後の周波数とは異なります。これは、RC 較正はシステムリセット中のデフォルト動作であり、システムリセット解除のたびに更新されるからです。*

*したがって、トレースポートアナライザ (TPA) は、システムリセット中に (TRACE\_CLKINEN ビットによって) トレースを有効にするべきではありません。なぜなら、同期フレームパケットは、リセット解除後に送信されるトレースパケットとは異なるビット時間で発行されるからです。*

## 40.17.9 TPIU レジスタ

TPIU APB レジスタは、デバッグ例外およびモニタ制御レジスタ (DEMCR) の TRCENA ビットがセットされた場合にのみ読み出し/書き込みが可能です。そうでない場合、これらのレジスタはゼロとして読み出されます (このビットの出力は TPIU の PCLK を有効にします)。

**表 265. 重要な TPIU レジスタ**

アドレス	レジスタ	説明
0xE0040004	現在のポートサイズ	<p>トレースポートサイズの選択 :</p> <p>ビット 0 : ポートサイズ = 1</p> <p>ビット 1 : ポートサイズ = 2</p> <p>ビット 2 : ポートサイズ = 3、サポートされません</p> <p>ビット 3 : ポートサイズ = 4</p> <p>1 ビットのみセットする必要があります。デフォルトでは、ポートサイズは 1 ビットです。(0x00000001)</p>
0xE00400F0	選択ピンプロトコル	<p>トレースポートプロトコルの選択 :</p> <p>ビット 1:0=</p> <p>00 : 同期トレースポートモード</p> <p>01 : シリアルワイヤ出力 - マンチェスタ (デフォルト値)</p> <p>10 : シリアルワイヤ出力 - NRZ</p> <p>11 : 予約済み</p>
0xE0040304	フォーマッタおよびフラッシュ制御	<p>ビット 31-9 = 常に "0"</p> <p>ビット 8 = TriglIn = 常に "1" で、トリガが指定されたことを示します</p> <p>ビット 7-4 = 常に 0</p> <p>ビット 3-2 = 常に 0</p> <p>ビット 1 = EnFCont同期トレースモード (Select_Pin_Protocol レジスタのビット 1:0 = 00) では、このビットは強制的に "1" にされ、フォーマッタは自動的に連続モードで有効にされます。非同期モード (Select_Pin_Protocol レジスタのビット 1:0 &lt;&gt; 00) では、このビットの書き込みによって、フォーマッタを有効/無効にできます。</p> <p>ビット 0 = 常に 0</p> <p>このレジスタのデフォルト値は 0x102 です。</p> <p><b>注 :</b> 同期モードでは、TRACECTL ビンがチップ外に配置されないため、フォーマッタは常に連続モードで有効になります。このように、フォーマッタは、トレースパケットの転送元を識別するための制御パケットを挿入します。</p>
0xE0040300	フォーマッタおよびフラッシュステータス	<p>FPU 搭載 Cortex<sup>®</sup>-M7 では使用されず、常に 0x00000008 が読み出されます。</p>

### 40.17.10 設定例

- デバッグ例外およびモニタ制御レジスタ (DEMCR) の TRCENA ビットをセットします。
- TPIU カレントポートサイズレジスタに希望する値を書き込みます (デフォルトは 0x1 で 1 ビットのポートサイズ)。
- TPIU フォーマッタおよびフラッシュ制御レジスタに 0x102 (デフォルト値) を書き込みます。
- TPIU 選択ピンプロトコルレジスタに書き込んで同期/非同期モードを選択します。例 : 非同期 NRZ モード (UART に類似) の場合は 0x2 を書き込みます。
- DBGMCU 制御レジスタに 0x20 (IO\_TRACEN ビット) を書き込み、TRACE I/O を非同期モードとします。このとき、TPIU 同期パケット (FF\_FF\_FF\_7F) が発行されます。
- ITM を設定し、ITM スティムラスレジスタに書き込んで値を出力します。

## 40.18 DBG レジスタマップ

次の表にデバッグレジスタの一覧を示します。

表 266. DBG レジスタマップとリセット値

アドレス	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0xE004 2000	DBGMCU_IDCODE	REV_ID												Res.	Res.	Res.	Res.	DEV_ID																
	リセット値 <sup>(1)</sup>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X	
0xE004 2004	DBGMCU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM8_STOP	DBG_I2C2_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_MODE_	TRACE_	Res.	Res.	DBG_STANDBY	DBG_STOP	DBG_SLEEP		
	リセット値												0	0	0	0	0									0	0	0			0	0	0	
0xE004 2008	DBGMCU_APB1_FZ	Res.	Res.	Res.	Res.	Res.	Res.	DBG_CAN2_STOP	DBG_CAN1_STOP	DBG_I2C4_SMBUS_TIMEOUT	DBG_I2C3_SMBUS_TIMEOUT	DBG_I2C2_SMBUS_TIMEOUT	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_LPTIM1_STOP	DBG_RTC_STOP	DBG_TIM14_STOP	DBG_TIM13_STOP	DBG_TIM12_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
	リセット値							0	0	0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0
0xE004 200C	DBGMCU_APB2_FZ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM11_STOP	DBG_TIM10_STOP	DBG_TIM9_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM8_STOP	DBG_TIM1_STOP	
	リセット値													0	0	0																0	0	

- リセット値は製品によって異なります。詳細については、[セクション 40.6.1: MCU デバイス ID コード](#)を参照してください。

## 41 デバイス電子署名

電子署名は、フラッシュメモリ領域に格納され、JTAG/SWD または CPU を使用して読み出すことができます。電子署名には、出荷時にプログラミングされた識別データが含まれています。電子署名に含まれる出荷時にプログラムされた識別データを使用すれば、ユーザファームウェアやその他の外部デバイスは、そのインタフェースを STM32F75xxx および STM32F74xxx マイクロコントローラの特性に自動的に整合させることができます。

### 41.1 ユニークデバイス ID レジスタ (96 ビット)

このユニークデバイス識別子は、以下の用途に最適です。

- シリアル番号 (例: USB 文字列シリアル番号やその他のエンドアプリケーション) として使用
- 内部フラッシュメモリをプログラムする前に、このユニーク ID をソフトウェア暗号プリミティブやプロトコルと組み合わせて使用する際に、フラッシュメモリ内のコードのセキュリティを高めるためのセキュリティキーとして使用
- セキュアなブートプロセスなどの起動に使用

96 ビットのユニークデバイス識別子は、状況やデバイスの違いとは無関係にユニークなリファレンス番号を提供します。ユーザは、これらのビットを変更できません。

96 ビットのユニークデバイス識別子は、さまざまな方法で 1 バイト/ハーフワード/ワード単位で読み出し、カスタムアルゴリズムを使用して連結することもできます。

**ベースアドレス : 0x1FF0 F420**

アドレスオフセット : 0x00

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

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

ビット 31:0 **U\_ID(31:0)** : 31:0 ユニーク ID ビット

アドレスオフセット : 0x04

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID(63:48)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(47:32)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **U\_ID(63:32)** : 63:32 ユニーク ID ビット

アドレスオフセット : 0x08

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID(95:80)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(79:64)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 U\_ID(95:64) : 95:64 ユニーク ID ビット

## 41.2 フラッシュサイズ

ベースアドレス : 0x1FF0 F442

アドレスオフセット : 0x00

読み出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 F\_ID(15:0) : フラッシュメモリサイズ

このビットフィールドは、デバイスのフラッシュメモリサイズを KB 単位で示します。  
たとえば、0x0400 は 1024 KB に対応します。

## 41.3 パッケージデータレジスタ

ベースアドレス : 0x1FFF 7BF0

アドレスオフセット : 0x00

読み出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	PKG[2:0]			Res.							
					rw	rw	rw								

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 PKG[2:0] : パッケージタイプ

- 0x1xx : LQFP208 および TFBGA216 パッケージ
- 0x011 : LQFP176 および UFBGA176 パッケージ
- 0x010 : WLCSP143 および LQFP144 パッケージ
- 0x001 : LQFP100 パッケージ
- 0x000 : 予約済み

ビット 7:0 予約済みであり、リセット値に保持する必要があります。



## 索引







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





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

## 42 改版履歴

表 267. 文書改版履歴

日付	版	変更内容
2015 年 5 月 21 日	1	初版リリース。
2015 年 7 月 21 日	2	<p><b>FMC セクションの更新：</b></p> <p><a href="#">表 81：プログラ可能なNAND フラッシュのアクセスパラメータ</a>のメモリリセットアップ時間とメモリホールド最大値の更新。</p> <p>c7amba_fmc_V2_UserSpec の更新：</p> <ul style="list-style-type: none"> <li>- <a href="#">セクション：共通メモリ空間タイミングレジスタ 2.4 (FMC_PMEM)</a> の MEMSEY、MEMHOLD、MEMHIZ レジスタの更新。</li> <li>- <a href="#">セクション：属性メモリ空間タイミングレジスタ 2.4 (FMC_PATT)</a> の ATTSET、ATTHOLD、ATTHIZ レジスタの更新。</li> <li>- <a href="#">表 87：FMC レジスタマップ</a> および [x:x] を追加するビットフィールドの説明の更新。</li> </ul> <p>また、FMC_SDCR2 のビット 13 からビット 31 は予約済み。</p> <ul style="list-style-type: none"> <li>- <a href="#">セクション：SRAM/NOR 型フラッシュチップセレクト制御レジスタ 1..4 (FMC_BCR1..4)</a> CPSIZE[2:0] ビットの説明に 011 設定を追加して更新。</li> </ul> <p><b>SYSARCHI セクションの更新：</b></p> <ul style="list-style-type: none"> <li>- <a href="#">セクション 2.1.8：DMA メモリバス</a> および <a href="#">セクション 2.1.9：DMA ペリフェラルバス</a> に「内部フラッシュメモリ」を追加して更新。</li> <li>- <a href="#">セクション 2.1.6：CPU AHBS バス</a> を更新。</li> </ul> <p><b>フラッシュメモリセクションの更新：</b></p> <ul style="list-style-type: none"> <li>- <a href="#">セクション 3.3.1：フラッシュメモリの構成</a> を更新。</li> <li>- <a href="#">セクション 3.3.8：フラッシュ 割り込み</a> の FLASH_SR を FLASH_CR レジスタに置き換えて更新。</li> </ul> <p>ビットフィールドおよび <a href="#">表 22：GPIO レジスタマップとリセット値</a> で GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A..K) を削除して、c7amba_ioport_UserSpec を更新。</p> <p>c7amba_spi2s1_v3_x_UserSpec <a href="#">セクション 32.1：概要</a> の「全二重モードおよび」を隠して更新。</p> <p><b>LTDC セクションの更新：</b></p> <ul style="list-style-type: none"> <li>- レジスタビットの説明、レジスタマップ、および太字形式に [x:x] を追加して、すべてのレジスタを更新。</li> <li>- <a href="#">セクション 18.4.2：レイヤのプログラム可能なパラメータ</a> を更新。</li> <li>- <a href="#">セクション 18.7.4：LTDC 全幅設定レジスタ (LTDC_TWCR)</a> の TOTALW[11:0] ビットフィールドを更新。</li> <li>- <a href="#">セクション 18.7.5：LTDC グローバル制御レジスタ (LTDC_GCR)</a> のビット 29 の説明を更新。</li> </ul> <p><b>USART セクションの更新：</b></p> <ul style="list-style-type: none"> <li>- <a href="#">セクション 31.5.13：スマートカードモード</a> の RTO カウンタの開始に関する注を更新。</li> </ul>



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

表 267. 文書改版履歴 (続き)

日付	版	変更内容
2015 年 7 月 21 日	2 (続き)	<p>- 表 173 : <i>USART 割り込みリクエスト</i> からラインレシーバタイムアウトエラーのエラーを削除して更新。</p> <p><b>PWR セクションの更新 :</b> 更新 : 表 14、表 15、表 17、および 表 18 に「割り込み (WFI) またはイベント (WFE) が保留になっていない状態」を追加。</p> <p><b>RTC セクションの更新 :</b> - セクション : <i>ウェイクアップタイムのプログラミング</i> のポイント 3 を更新。ウェイクアップ自動再ロード値をプログラム。 - セクション 29.6.4 : <i>RTC 初期化とステータスレジスタ (RTC_ISR)</i> のビット 2 「WUTWF : ウェイクアップタイム書き込みフラグ」の説明を更新。</p> <p><b>DCMI セクションの更新 :</b> <i>セクション 17.1 : DCMI の概要</i> を更新。出力モードはサポートされません。</p>

## 重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST製品は、注文請書発行時点で有効なSTの販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してSTは一切の責任を負いません。

明示又は黙示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

STおよびSTロゴはSTMicroelectronicsの商標です。その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

© 2016 STMicroelectronics - All rights reserved

