



## 概要

このリファレンスマニュアルは、アプリケーション開発者を対象としています。STM32F303xB/C/D/E、STM32F303x6/x8、STM32F328x8、STM32F358xC、および STM32F398xE マイクロコントローラのメモリやペリフェラルの使用方法について、詳細に説明しています。特に指定がない限り、STM32F303xB/C/D/E、STM32F303x6/x8、STM32F328x8、STM32F358xC、および STM32F398xE デバイスは、ドキュメント全体にわたって STM32F3xx と呼ばれます。

STM32F3xx は、さまざまなメモリサイズ、パッケージ、およびペリフェラルを持つマイクロコントローラファミリ製品になります。

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

ARM FPU 搭載<sup>®</sup> CORTEX<sup>®</sup>-M4 コアについては、STM32F3xx/STM32F4xx プログラミングマニュアル (PM0214) を参照してください。

## 関連ドキュメント

- STM32F303xB/C、STM32F303xD/E、STM32F303x6/8、STM32F328x8、STM32F358xC、および STM32F398xE データシートは、当社ウェブサイト [www.st.com](http://www.st.com) から入手できます。
- STM32F3xx/F4xx Cortex<sup>®</sup>-M4 プログラミングマニュアル (PM0214) は、当社ウェブサイト [www.st.com](http://www.st.com) から入手できます。

## 目次

<b>1</b>	<b>本マニュアルの概要</b> .....	<b>43</b>
<b>2</b>	<b>このマニュアルにおける表記の規則</b> .....	<b>45</b>
2.1	レジスタに関する略記 .....	45
2.2	用語 .....	45
2.3	使用可能なペリフェラル .....	45
<b>3</b>	<b>システムおよびメモリの概要</b> .....	<b>46</b>
3.1	システムアーキテクチャ .....	46
3.1.1	S0 : I-バス .....	48
3.1.2	S1 : D-バス .....	48
3.1.3	S2 : S-バス .....	48
3.1.4	S3、S4 : DMA バス .....	48
3.1.5	バスマトリックス .....	49
3.2	メモリ構成 .....	50
3.2.1	概要 .....	50
3.2.2	メモリマップとレジスタ境界アドレス .....	50
3.3	内蔵 SRAM .....	57
3.3.1	パリティチェック .....	58
3.3.2	CCM SRAM 書き込み保護 .....	59
3.4	フラッシュメモリの概要 .....	60
3.5	ブート設定 .....	60
3.5.1	内蔵ブートローダ .....	61
<b>4</b>	<b>内蔵フラッシュメモリ</b> .....	<b>62</b>
4.1	フラッシュの主な機能 .....	62
4.2	フラッシュメモリの機能詳細 .....	62
4.2.1	フラッシュメモリの構成 .....	62
4.2.2	読み出し操作 .....	64
4.2.3	フラッシュのプログラムおよび消去操作 .....	66
4.3	メモリ保護 .....	72
4.3.1	読み出し保護 .....	72
4.3.2	書き込み保護 .....	74

4.3.3	オプションバイトブロックの書き込み保護	75
4.4	フラッシュ割り込み	75
4.5	フラッシュレジスタの説明	76
4.5.1	フラッシュアクセス制御レジスタ (FLASH_ACR)	76
4.5.2	フラッシュキーレジスタ (FLASH_KEYR)	76
4.5.3	フラッシュオプションキーレジスタ (FLASH_OPTKEYR)	77
4.5.4	フラッシュステータスレジスタ (FLASH_SR)	77
4.5.5	フラッシュ制御レジスタ (FLASH_CR)	78
4.5.6	フラッシュアドレスレジスタ (FLASH_AR)	79
4.5.7	オプションバイトレジスタ (FLASH_OBR)	80
4.5.8	書き込み保護レジスタ (FLASH_WRPR)	81
4.6	フラッシュレジスタマップ	81
<b>5</b>	<b>オプションバイトの説明</b>	<b>83</b>
<b>6</b>	<b>巡回冗長検査計算ユニット (CRC)</b>	<b>86</b>
6.1	概要	86
6.2	CRC の主な機能	86
6.3	CRC の機能説明	87
6.4	CRC レジスタ	88
6.4.1	データレジスタ (CRC_DR)	88
6.4.2	独立型データレジスタ (CRC_IDR)	89
6.4.3	制御レジスタ (CRC_CR)	89
6.4.4	CRC の初期値 (CRC_INIT)	90
6.4.5	CRC 多項式 (CRC_POL)	90
6.4.6	CRC レジスタマップ	91
<b>7</b>	<b>電源制御 (PWR)</b>	<b>92</b>
7.1	電源	92
7.1.1	独立した A/D および D/A コンバータ用電源と基準電圧	94
7.1.2	バッテリーバックアップドメイン	94
7.1.3	電圧レギュレータ	95
7.2	電源供給スーパバイザ	95
7.2.1	パワーオンリセット (POR) / パワーダウンリセット (PDR)	95
7.2.2	プログラム可能な電圧検出器 (PVD)	97
7.2.3	外部 NPOR 信号	97



7.3	低電力モード	98
7.3.1	システムクロックの低速化	99
7.3.2	ペリフェラルクロックゲーティング	99
7.3.3	SLEEP モード	99
7.3.4	STOP モード	100
7.3.5	STANDBY モード	102
7.3.6	低電力モードからの自動ウェイクアップ	104
7.4	電源制御レジスタ	105
7.4.1	電源制御レジスタ (PWR_CR)	105
7.4.2	電源制御/ステータスレジスタ (PWR_CSR)	106
7.4.3	PWR レジスタマップ	108
<b>8</b>	<b>ペリフェラル相互接続マトリックス</b>	<b>109</b>
8.1	概要	109
8.2	接続の一覧	109
8.3	相互接続の詳細	113
8.3.1	DMA 相互接続	113
8.3.2	ADC から ADC	113
8.3.3	ADC から TIM	113
8.3.4	TIM および EXTI から ADC	113
8.3.5	OPAMP から ADC	114
8.3.6	TS から ADC	114
8.3.7	VBAT から ADC	114
8.3.8	VREFINT から ADC	114
8.3.9	COMP から TIM	115
8.3.10	TIM から COMP	116
8.3.11	DAC から COMP	117
8.3.12	VREFINT から COMP	117
8.3.13	DAC から OPAMP	117
8.3.14	TIM から OPAMP	117
8.3.15	TIM から TIM	118
8.3.16	ブレーク入力ソースから TIM	119
8.3.17	HSE、HSI、LSE、LSI、MCO、RTC から TIM	119
8.3.18	TIM および EXTI から DAC	120
8.3.19	TIM から IRTIM	120

<b>9</b>	<b>リセットおよびクロック制御 (RCC)</b> .....	<b>121</b>
9.1	リセット .....	121
9.1.1	電源リセット .....	121
9.1.2	システムリセット .....	121
9.1.3	RTC ドメインリセット .....	122
9.2	クロック .....	123
9.2.1	HSE クロック .....	127
9.2.2	HSI クロック .....	128
9.2.3	PLL .....	129
9.2.4	LSE クロック .....	129
9.2.5	LSI クロック .....	130
9.2.6	システムクロック (SYSCLK) の選択 .....	130
9.2.7	クロックセキュリティシステム (CSS) .....	130
9.2.8	ADC クロック .....	131
9.2.9	RTC クロック .....	131
9.2.10	タイマ (TIMx) クロック .....	131
9.2.11	ウォッチドッグクロック .....	132
9.2.12	I2S クロック (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ) .....	132
9.2.13	クロック信号出力 .....	132
9.2.14	TIM16 を使用した内部/外部クロックの測定 .....	133
9.3	低電力モード .....	134
9.4	RCC レジスタ .....	135
9.4.1	クロック制御レジスタ (RCC_CR) .....	135
9.4.2	クロック設定レジスタ (RCC_CFGR) .....	137
9.4.3	クロック割り込みレジスタ (RCC_CIR) .....	141
9.4.4	APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR) .....	143
9.4.5	APB1 ペリフェラルリセットレジスタ (RCC_APB1RSTR) .....	145
9.4.6	AHB ペリフェラルクロック有効レジスタ (RCC_AHBENR) .....	147
9.4.7	APB2 ペリフェラルクロック有効レジスタ (RCC_APB2ENR) .....	149
9.4.8	APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR) .....	151
9.4.9	RTC ドメイン制御レジスタ (RCC_BDCR) .....	154
9.4.10	制御/ステータスレジスタ (RCC_CSR) .....	155
9.4.11	AHB ペリフェラルリセットレジスタ (RCC_AHBRSTR) .....	157
9.4.12	クロック設定レジスタ 2 (RCC_CFGR2) .....	158
9.4.13	クロック設定レジスタ 3 (RCC_CFGR3) .....	161
9.4.14	RCC レジスタマップ .....	165



<b>10</b>	<b>フレキシブルスタティックメモリコントローラ (FSMC) .....</b>	<b>167</b>
10.1	FMC の主な機能 .....	167
10.2	ブロック図 .....	168
10.3	AHB インタフェース .....	169
10.3.1	サポートされるメモリおよびトランザクション .....	169
10.4	外部デバイスアドレスマッピング .....	170
10.4.1	NOR/PSRAM アドレスマッピング .....	171
10.4.2	NAND フラッシュメモリ /PC カードアドレスマッピング .....	172
10.5	NOR 型フラッシュ / PSRAM コントローラ .....	173
10.5.1	外部メモリインタフェース信号 .....	174
10.5.2	サポートされるメモリおよびトランザクション .....	176
10.5.3	一般的なタイミング規則 .....	177
10.5.4	NOR フラッシュ /PSRAM コントローラ非同期トランザクション .....	178
10.5.5	同期トランザクション .....	195
10.5.6	NOR/PSRAM コントローラレジスタ .....	201
10.6	NAND 型フラッシュ /PC カードコントローラ .....	208
10.6.1	外部メモリインタフェース信号 .....	209
10.6.2	NAND 型フラッシュ /PC カードによってサポートされるメモリおよびトランザクション .....	211
10.6.3	NAND フラッシュメモリと PC カードのタイミング図 .....	211
10.6.4	NAND フラッシュ動作 .....	212
10.6.5	NAND フラッシュのプリウェイト機能 .....	213
10.6.6	エラー訂正コード (ECC) の計算 NAND 型フラッシュメモリ .....	214
10.6.7	PC カード / コンパクトフラッシュの動作 .....	215
10.6.8	NAND 型フラッシュ /PC カードコントローラレジスタ .....	217
10.7	FMC レジスタマップ .....	224
<b>11</b>	<b>汎用 I/O (GPIO) .....</b>	<b>226</b>
11.1	概要 .....	226
11.2	GPIO の主な機能 .....	226
11.3	GPIO の機能説明 .....	226
11.3.1	汎用 I/O (GPIO) .....	229
11.3.2	I/O ピンオルタネート機能マルチプレクサと配置 .....	229
11.3.3	I/O ポート制御レジスタ .....	230
11.3.4	I/O ポートデータレジスタ .....	230

11.3.5	I/O データのビット単位の操作	230
11.3.6	GPIO ロック機構	231
11.3.7	I/O オルタネート機能の入力/出力	231
11.3.8	外部割り込み/ウェイクアップライン	231
11.3.9	入力設定	231
11.3.10	出力設定	232
11.3.11	オルタネート機能設定	233
11.3.12	アナログ設定	234
11.3.13	HSE または LSE オシレータのピンを GPIO として使用	234
11.3.14	GPIO ピンを RTC 供給ドメインで使用	234
11.4	GPIO レジスタ	235
11.4.1	GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ H)	235
11.4.2	GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A ~ H)	235
11.4.3	GPIO ポート出カスピードレジスタ (GPIOx_OSPEEDR) (x = A ~ H)	236
11.4.4	GPIO ポートプルアップ/プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ H)	236
11.4.5	GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ H)	237
11.4.6	GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A ~ H)	237
11.4.7	GPIO ポートビットセット/リセットレジスタ (GPIOx_BSRR) (x = A ~ H)	238
11.4.8	GPIO ポート設定ロックレジスタ (GPIOx_LCKR)	238
11.4.9	GPIO オルタネート機能下位レジスタ (GPIOx_AFRL) (x = A ~ H)	239
11.4.10	GPIO オルタネート機能上位レジスタ (GPIOx_AFRH) (x = A ~ H)	240
11.4.11	GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A ~ H)	240
11.4.12	GPIO レジスタマップ	241
<b>12</b>	<b>システム設定コントローラ (SYSCFG)</b>	<b>243</b>
12.1	SYSCFG レジスタ	243
12.1.1	SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1)	243
12.1.2	SYSCFG CCM SRAM 保護レジスタ (SYSCFG_RCR)	246
12.1.3	SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1)	247
12.1.4	SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2)	248
12.1.5	SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG_EXTICR3)	250
12.1.6	SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4)	252
12.1.7	SYSCFG 設定レジスタ 2 (SYSCFG_CFGR2)	253
12.1.8	SYSCFG 設定レジスタ 3 (SYSCFG_CFGR3)	255
12.1.9	SYSCFG 設定レジスタ 4 (SYSCFG_CFGR4)	256



	12.1.10 SYSCFG レジスタマップ .....	259
<b>13</b>	<b>ダイレクトメモリアクセスコントローラ (DMA) .....</b>	<b>261</b>
13.1	概要 .....	261
13.2	DMA の主な機能 .....	261
13.3	DMA の実装 .....	262
13.4	DMA の機能説明 .....	263
13.4.1	DMA トランザクション .....	263
13.4.2	アービタ .....	263
13.4.3	DMA チャンネル .....	264
13.4.4	プログラム可能なデータ幅、データの整列、およびエンディアン .....	265
13.4.5	エラー管理 .....	266
13.4.6	DMA 割り込み .....	266
13.4.7	DMA リクエストマッピング .....	267
13.5	DMA レジスタ .....	274
13.5.1	DMA 割り込みステータスレジスタ (DMA_ISR) .....	274
13.5.2	DMA 割り込みフラグクリアレジスタ (DMA_IFCR) .....	275
13.5.3	DMA チャンネル x 設定レジスタ (DMA_CCRx) (x = 1 ~ 7、ここで x = チャンネル数) .....	276
13.5.4	DMA チャンネル x データ数レジスタ (DMA_CNDTRx) (x = 1 ~ 7、x = チャンネル番号) .....	278
13.5.5	DMA チャンネル x ペリフェラルアドレスレジスタ (DMA_CPARx) (x = 1 ~ 7、x = チャンネル番号) .....	278
13.5.6	DMA チャンネル x メモリアドレスレジスタ (DMA_CMARx) (x = 1 ~ 7、x = チャンネル番号) .....	279
13.5.7	DMA レジスタマップ .....	280
<b>14</b>	<b>割り込みとイベント .....</b>	<b>283</b>
14.1	ネスト化されたベクタ割り込みコントローラ (NVIC) .....	283
14.1.1	NVIC の主な機能 .....	283
14.1.2	SysTick 較正值レジスタ .....	283
14.1.3	割り込みベクタと例外ベクタ .....	283
14.2	拡張割り込み/イベントコントローラ (EXTI) .....	290
14.2.1	主な機能 .....	290
14.2.2	ブロック図 .....	291
14.2.3	ウェイクアップイベント管理 .....	291
14.2.4	非同期内部割り込み .....	292
14.2.5	機能詳細 .....	292



14.2.6	外部および内部の割り込み/イベントラインの配置	293
14.3	EXTI レジスタ	295
14.3.1	割り込みマスクレジスタ (EXTI_IMR1)	295
14.3.2	イベントマスクレジスタ (EXTI_EMR1)	295
14.3.3	立ち上がりトリガ選択レジスタ (EXTI_RTISR1)	296
14.3.4	立ち下がりトリガ選択レジスタ (EXTI_FTISR1)	296
14.3.5	ソフトウェア割り込みイベントレジスタ (EXTI_SWIER1)	297
14.3.6	ペンディングレジスタ (EXTI_PR1)	297
14.3.7	割り込みマスクレジスタ (EXTI_IMR2)	298
14.3.8	イベントマスクレジスタ (EXTI_EMR2)	298
14.3.9	立ち上がりトリガ選択レジスタ (EXTI_RTISR2)	299
14.3.10	立ち下がりトリガ選択レジスタ (EXTI_FTISR2)	299
14.3.11	ソフトウェア割り込みイベントレジスタ (EXTI_SWIER2)	300
14.3.12	ペンディングレジスタ (EXTI_PR2)	300
14.3.13	EXTI レジスタマップ	301
<b>15</b>	<b>アナログデジタルコンバータ (ADC)</b>	<b>303</b>
15.1	概要	303
15.2	ADC の主な機能	304
15.3	ADC の機能詳細	306
15.3.1	ADC ブロック図	306
15.3.2	ピンおよび内部信号	307
15.3.3	クロック	308
15.3.4	ADC1/2 および ADC3/4 の接続性	310
15.3.5	スレーブ AHB インタフェース	313
15.3.6	ADC 電圧レギュレータ (ADVREGEN)	313
15.3.7	シングルエンドおよび差動入力のチャンネル	313
15.3.8	較正 (ADCAL、ADCALDIF、ADCx_CALFACT)	314
15.3.9	ADC オン/オフ制御 (ADEN、ADDIS、ADRDY)	317
15.3.10	ADC 制御ビット書き込み時の制約	318
15.3.11	チャンネルの選択 (SQRx、JSQRx)	318
15.3.12	チャンネル単位でプログラム可能なサンプリング時間 (SMPR1、SMPR2)	319
15.3.13	シングル変換モード (CONT=0)	320
15.3.14	連続変換モード (CONT=1)	320
15.3.15	変換の開始 (ADSTART、JADSTART)	321
15.3.16	タイミング	322
15.3.17	実行中の変換の停止 (ADSTP、JADSTP)	322



15.3.18	外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN、JEXTSEL、JEXTEN) .....	324
15.3.19	インジェクトチャネルの管理 .....	328
15.3.20	不連続モード (DISCEN、DISCNUM、JDISCEN) .....	329
15.3.21	インジェクト変換のコンテキストのキュー .....	331
15.3.22	プログラム可能な分解能 (RES) - 高速変換モード .....	338
15.3.23	変換の終了、サンプリングフェーズの終了 (EOC、JEOC、EOSMP) .....	339
15.3.24	変換シーケンスの終了 (EOS、JEOS) .....	339
15.3.25	タイミング図の例 (シングル / 連続モード、ハードウェア / ソフトウェアトリガ) .....	339
15.3.26	データ管理 .....	341
15.3.27	動的低電力機能 .....	346
15.3.28	アナログウィンドウウォッチドッグ (AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx) .....	351
15.3.29	デュアル ADC モード .....	355
15.3.30	温度センサ .....	368
15.3.31	VBAT 電源監視 .....	370
15.3.32	内部電圧基準の監視 .....	370
15.4	ADC 割り込み .....	372
15.5	ADC レジスタ (ADC ごと) .....	373
15.5.1	ADC 割り込みおよびステータスレジスタ (ADCx_ISR、x=1..4) .....	373
15.5.2	ADC 割り込み有効レジスタ (ADCx_IER、x=1..4) .....	375
15.5.3	ADC 制御レジスタ (ADCx_CR、x=1..4) .....	377
15.5.4	ADC 設定レジスタ (ADCx_CFGR、x=1..4) .....	380
15.5.5	ADC サンプル時間レジスタ 1 (ADCx_SMPR1、x=1..4) .....	384
15.5.6	ADC サンプル時間レジスタ 2 (ADCx_SMPR2、x=1..4) .....	386
15.5.7	ADC ウォッチドッグ閾値レジスタ 1 (ADCx_TR1、x=1..4) .....	386
15.5.8	ADC ウォッチドッグ閾値レジスタ 2 (ADCx_TR2、x = 1..4) .....	387
15.5.9	ADC ウォッチドッグ閾値レジスタ 3 (ADCx_TR3、x=1..4) .....	388
15.5.10	ADC レギュラシーケンスレジスタ 1 (ADCx_SQR1、x=1..4) .....	389
15.5.11	ADC レギュラシーケンスレジスタ 2 (ADCx_SQR2、x=1..4) .....	390
15.5.12	ADC レギュラシーケンスレジスタ 3 (ADCx_SQR3、x=1..4) .....	392
15.5.13	ADC レギュラシーケンスレジスタ 4 (ADCx_SQR4、x=1..4) .....	393
15.5.14	ADC レギュラデータレジスタ (ADCx_DR、x=1..4) .....	394
15.5.15	ADC インジェクトシーケンスレジスタ (ADCx_JSQR、x=1..4) .....	395
15.5.16	ADC オフセットレジスタ (ADCx_OFRy、x=1..4) (y=1..4) .....	397
15.5.17	ADC インジェクトデータレジスタ (ADCx_JDRy、x=1..4、y = 1..4) .....	398
15.5.18	ADC アナログウォッチドッグ 2 設定レジスタ (ADCx_AWD2CR、x=1..4) ..	398

15.5.19	ADC アナログウォッチドッグ 3 設定レジスタ (ADCx_AWD3CR、x=1..4)	399
15.5.20	ADC 差動モード選択レジスタ (ADCx_DIFSEL、x=1..4)	399
15.5.21	ADC 較正係数 (ADCx_CALFACT、x=1..4)	400
15.6	ADC 共通レジスタ	401
15.6.1	ADC 共通ステータスレジスタ (ADCx_CSR、x=12 または 34)	401
15.6.2	ADC 共通制御レジスタ (ADCx_CCR、x=12 または 34)	403
15.6.3	デュアルモード用 ADC 共通レギュレータレジスタ (ADCx_CDR、x=12 または 34)	406
15.6.4	ADC レジスタマップ	406
<b>16</b>	<b>D/A コンバータ (DAC1 および DAC2)</b>	<b>410</b>
16.1	概要	410
16.2	DAC1/2 の主な機能	410
16.3	DAC 出力バッファイネーブル/DAC 出力スイッチ	412
16.4	DAC チャネルイネーブル	413
16.5	シングルモードの機能説明	413
16.5.1	DAC データフォーマット	413
16.5.2	DAC チャネル変換	414
16.5.3	DAC 出力電圧	415
16.5.4	DAC トリガ選択	415
16.6	デュアルモードの機能説明	416
16.6.1	DAC データフォーマット	416
16.6.2	デュアルモードでの DAC チャネル変換	416
16.6.3	デュアル変換モードの説明	417
16.6.4	DAC 出力電圧	420
16.6.5	DAC トリガ選択	421
16.7	ノイズ生成	421
16.8	三角波生成	422
16.9	DMA リクエスト	423
16.10	DAC レジスタ	424
16.10.1	DAC 制御レジスタ (DAC_CR)	424
16.10.2	DAC ソフトウェアトリガレジスタ (DAC_SWTRIGR)	428
16.10.3	DAC チャネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R1)	428
16.10.4	DAC チャネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L1)	429
16.10.5	DAC チャネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R1)	429



16.10.6	DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2) .....	429
16.10.7	DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L2) .....	430
16.10.8	DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R2) .....	430
16.10.9	デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD) .....	431
16.10.10	デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD) .....	431
16.10.11	デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD) .....	432
16.10.12	DAC チャンネル 1 データ出力レジスタ (DAC_DOR1) .....	432
16.10.13	DAC チャンネル 2 データ出力レジスタ (DAC_DOR2) .....	432
16.10.14	DAC ステータスレジスタ (DAC_SR) .....	433
16.10.15	DAC レジスタマップ .....	434
<b>17</b>	<b>コンパレータ (COMP) .....</b>	<b>436</b>
17.1	概要 .....	436
17.2	COMP の主な機能 .....	436
17.3	COMP の機能説明 .....	437
17.3.1	COMP ブロック図 .....	437
17.3.2	COMP ピンおよび内部信号 .....	439
17.3.3	COMP のリセットおよびクロック .....	441
17.3.4	コンパレータのロック機構 .....	441
17.3.5	ヒステリシス (STM32F303xB/C および STM32F358xC のみ) .....	441
17.3.6	コンパレータの出力のブランキング機能 .....	442
17.3.7	電力モード (STM32F303xB/C および STM32F358xC のみ) .....	442
17.4	COMP 割り込み .....	442
17.5	COMP レジスタ .....	443
17.5.1	COMP1 制御およびステータスレジスタ (COMP1_CSR) .....	443
17.5.2	COMP2 制御およびステータスレジスタ (COMP2_CSR) .....	445
17.5.3	COMP3 制御およびステータスレジスタ (COMP3_CSR) .....	447
17.5.4	COMP4 制御およびステータスレジスタ (COMP4_CSR) .....	450
17.5.5	COMP5 制御およびステータスレジスタ (COMP5_CSR) .....	452
17.5.6	COMP6 制御およびステータスレジスタ (COMP6_CSR) .....	455
17.5.7	COMP7 制御およびステータスレジスタ (COMP7_CSR) .....	457
17.5.8	COMP レジスタマップ .....	460

<b>18</b>	<b>オペアンプ (OPAMP) .....</b>	<b>462</b>
18.1	OPAMP の概要 .....	462
18.2	OPAMP の主な機能 .....	462
18.3	OPAMP の機能説明 .....	462
18.3.1	概要 .....	462
18.3.2	クロック .....	463
18.3.3	オペアンプとコンパレータの相互接続 .....	464
18.3.4	ADC 入力としての OPAMP 出力の使用 .....	466
18.3.5	較正 .....	466
18.3.6	タイマによって制御されたマルチプレクサモード .....	467
18.3.7	OPAMP モード .....	468
18.4	OPAMP レジスタ .....	472
18.4.1	OPAMP1 制御レジスタ (OPAMP1_CSR) .....	472
18.4.2	OPAMP2 制御レジスタ (OPAMP2_CSR) .....	474
18.4.3	OPAMP3 制御レジスタ (OPAMP3_CSR) .....	476
18.4.4	OPAMP4 制御レジスタ (OPAMP4_CSR) .....	479
18.4.5	OPAMP レジスタマップ .....	482
<b>19</b>	<b>タッチセンシングコントローラ (TSC) .....</b>	<b>483</b>
19.1	概要 .....	483
19.2	TSC の主な機能 .....	483
19.3	TSC の機能説明 .....	484
19.3.1	TSC ブロック図 .....	484
19.3.2	表面電荷移動取得の概要 .....	484
19.3.3	リセットおよびクロック .....	486
19.3.4	電荷移動取得シーケンス .....	487
19.3.5	スペクトル拡散機能 .....	488
19.3.6	最大カウントエラー .....	488
19.3.7	サンプリングコンデンサ I/O および チャネル I/O のモード選択 .....	489
19.3.8	取得モード .....	490
19.3.9	I/O ヒステリシスおよびアナログスイッチの制御 .....	490
19.4	TSC 低電力モード .....	491
19.5	TSC 割り込み .....	491
19.6	TSC レジスタ .....	492
19.6.1	TSC 制御レジスタ (TSC_CR) .....	492
19.6.2	TSC 割り込み有効レジスタ (TSC_IER) .....	494



19.6.3	TSC 割り込みクリアレジスタ (TSC_ICR) .....	495
19.6.4	TSC 割り込みステータスレジスタ (TSC_ISR) .....	496
19.6.5	TSC I/O ヒステリシス制御レジスタ (TSC_IOHCR) .....	496
19.6.6	TSC I/O アナログスイッチ制御レジスタ (TSC_IOASCR) .....	497
19.6.7	TSC I/O サンプリング制御レジスタ (TSC_IOSCR) .....	497
19.6.8	TSC I/O チャネル制御レジスタ (TSC_IOCCTSC_IOCCR) .....	498
19.6.9	TSC I/O グループ制御ステータスレジスタ (TSC_IQGCSR) .....	498
19.6.10	TSC I/O グループ x カウンタレジスタ (TSC_IQGXCR) (x = 1 ~ 8) .....	499
19.6.11	TSC レジスタマップ .....	500
<b>20</b>	<b>高機能制御タイマ (TIM1/TIM8/TIM20) .....</b>	<b>502</b>
20.1	TIM1/TIM8/TIM20 の概要 .....	502
20.2	TIM1/TIM8/TIM20 の主な特長 .....	502
20.3	TIM1/TIM8/TIM20 機能詳細 .....	504
20.3.1	タイムベースユニット .....	504
20.3.2	カウンタモード .....	506
20.3.3	繰り返しカウンタ .....	517
20.3.4	外部トリガ入力 .....	519
20.3.5	クロック選択 .....	520
20.3.6	キャプチャ/比較チャネル .....	524
20.3.7	入力キャプチャモード .....	527
20.3.8	PWM 入力モード .....	528
20.3.9	強制出力モード .....	529
20.3.10	出力比較モード .....	530
20.3.11	PWM モード .....	531
20.3.12	非対称 PWM モード .....	534
20.3.13	組み合わせ PWM モード .....	535
20.3.14	組み合わせ 3 相 PWM モード .....	536
20.3.15	相補出力とデッドタイム挿入 .....	537
20.3.16	ブ레이크機能の使用 .....	539
20.3.17	外部イベントによる OCxREF 信号のクリア .....	544
20.3.18	6 ステップ PWM 生成 .....	546
20.3.19	ワンパルスモード .....	547
20.3.20	再トリガ可能なワンパルスモード (OPM) .....	548
20.3.21	エンコーダインタフェースモード .....	549
20.3.22	UIF ビットの再配置 .....	551
20.3.23	タイマ入力 XOR 機能 .....	552

20.3.24	ホールセンサとのインタフェース	552
20.3.25	タイマの同期	555
20.3.26	ADC の同期	558
20.3.27	DMA バーストモード	559
20.3.28	デバッグモード	560
20.4	TIM1/TIM8/TIM20 レジスタ	561
20.4.1	TIM1/TIM8/TIM20 制御レジスタ 1 (TIMx_CR1)	561
20.4.2	TIM1/TIM8/TIM20 制御レジスタ 2 (TIMx_CR2)	562
20.4.3	TIM1/TIM8/TIM20 スレーブモード制御レジスタ (TIMx_SMCR)	565
20.4.4	TIM1/TIM8/TIM20 DMA / 割り込み有効レジスタ (TIMx_DIER)	567
20.4.5	TIM1/TIM8/TIM20 ステータスレジスタ (TIMx_SR)	569
20.4.6	TIM1/TIM8/TIM20 イベント生成レジスタ (TIMx_EGR)	571
20.4.7	TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	572
20.4.8	TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	576
20.4.9	TIM1/TIM8/TIM20 キャプチャ/比較有効レジスタ (TIMx_CCER)	578
20.4.10	TIM1/TIM8/TIM20 カウンタ (TIMx_CNT)	582
20.4.11	TIM1/TIM8/TIM20 プリスケーラ (TIMx_PSC)	582
20.4.12	TIM1/TIM8/TIM20 自動再ロードレジスタ (TIMx_ARR)	582
20.4.13	TIM1/TIM8/TIM20 繰り返しカウンタレジスタ (TIMx_RCR)	583
20.4.14	TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 1 (TIMx_CCR1)	583
20.4.15	TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 2 (TIMx_CCR2)	583
20.4.16	TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 3 (TIMx_CCR3)	584
20.4.17	TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 4 (TIMx_CCR4)	584
20.4.18	TIM1/TIM8/TIM20 ブレークおよびデッドタイムレジスタ (TIMx_BDTR)	585
20.4.19	TIM1/TIM8/TIM20 DMA 制御レジスタ (TIMx_DCR)	588
20.4.20	完全転送の TIM1/TIM8/TIM20 DMA アドレス (TIMx_DMAR)	589
20.4.21	TIM1/TIM8/TIM20 オプションレジスタ (TIMx_OR)	590
20.4.22	TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 3 (TIMx_CCMR3)	591
20.4.23	TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 5 (TIMx_CCR5)	592
20.4.24	TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 6 (TIMx_CCR6)	593
20.4.25	TIM1/TIM8/TIM20 レジスタマップ	594
<b>21</b>	<b>汎用タイマ (TIM2/TIM3/TIM4)</b>	<b>597</b>
21.1	TIM2/TIM3/TIM4 の概要	597
21.2	TIM2/TIM3/TIM4 の主な特長	597
21.3	TIM2/TIM3/TIM4 機能詳細	599
21.3.1	タイムベースユニット	599



21.3.2	カウンタモード	601
21.3.3	クロック選択	611
21.3.4	キャプチャ/比較チャンネル	615
21.3.5	入力キャプチャモード	617
21.3.6	PWM 入力モード	618
21.3.7	強制出力モード	619
21.3.8	出力比較モード	619
21.3.9	PWM モード	621
21.3.10	非対称 PWM モード	624
21.3.11	組み合わせ PWM モード	624
21.3.12	外部イベントによる OCxREF 信号のクリア	625
21.3.13	ワンパルスモード	627
21.3.14	再トリガ可能なワンパルスモード (OPM)	628
21.3.15	エンコーダインタフェースモード	629
21.3.16	UIF ビットの再配置	631
21.3.17	タイマ入力 XOR 機能	631
21.3.18	タイマと外部トリガの同期	632
21.3.19	タイマの同期	635
21.3.20	DMA バーストモード	640
21.3.21	デバッグモード	641
21.4	TIM2/TIM3/TIM4 レジスタ	642
21.4.1	TIMx 制御レジスタ 1 (TIMx_CR1)	642
21.4.2	TIMx 制御レジスタ 2 (TIMx_CR2)	644
21.4.3	TIMx スレーブモード制御レジスタ (TIMx_SMCR)	645
21.4.4	TIMx DMA / 割り込み有効レジスタ (TIMx_DIER)	648
21.4.5	TIMx ステータスレジスタ (TIMx_SR)	649
21.4.6	TIMx イベント生成レジスタ (TIMx_EGR)	651
21.4.7	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	652
21.4.8	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	656
21.4.9	TIMx キャプチャ/比較有効レジスタ (TIMx_CCER)	657
21.4.10	TIMx カウンタ (TIMx_CNT)	659
21.4.11	TIMx プリスケーラ (TIMx_PSC)	660
21.4.12	TIMx 自動再ロードレジスタ (TIMx_ARR)	660
21.4.13	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCR1)	661
21.4.14	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCR2)	661
21.4.15	TIMx キャプチャ/比較モードレジスタ 3 (TIMx_CCR3)	662
21.4.16	TIMx キャプチャ/比較モードレジスタ 4 (TIMx_CCR4)	662



21.4.17	TIMx DMA 制御レジスタ (TIMx_DCR) .....	663
21.4.18	完全転送の TIMx DMA アドレス (TIMx_DMAR) .....	663
21.4.19	TIMx レジスタマップ .....	664
<b>22</b>	<b>基本タイマ (TIM6/TIM7) : .....</b>	<b>666</b>
22.1	TIM6/TIM7 の概要 .....	666
22.2	TIM6/TIM7 の主な機能 .....	666
22.3	TIM6/TIM7 の機能詳細 .....	667
22.3.1	タイムベースユニット .....	667
22.3.2	カウントモード .....	669
22.3.3	UIF ビットの再配置 .....	672
22.3.4	クロックソース .....	672
22.3.5	デバッグモード .....	673
22.4	TIM6/TIM7 レジスタ .....	673
22.4.1	TIM6/TIM7 制御レジスタ 1 (TIMx_CR1) .....	673
22.4.2	TIM6/TIM7 制御レジスタ 2 (TIMx_CR2) .....	675
22.4.3	TIM6/TIM7 DMA / 割り込み有効レジスタ (TIMx_DIER) .....	675
22.4.4	TIM6/TIM7 のステータスレジスタ (TIMx_SR) .....	676
22.4.5	TIM6/TIM7 のイベント生成レジスタ (TIMx_EGR) .....	676
22.4.6	TIM6/TIM7 のカウンタ (TIMx_CNT) .....	676
22.4.7	TIM6/TIM7 プリスケラ (TIMx_PSC) .....	677
22.4.8	TIM6/TIM7 の自動再ロードレジスタ (TIMx_ARR) .....	677
22.4.9	TIM6/TIM7 レジスタマップ .....	678
<b>23</b>	<b>汎用タイマ (TIM15/TIM16/TIM17) .....</b>	<b>679</b>
23.1	TIM15/TIM16/TIM17 の概要 .....	679
23.2	TIM15 の主な特長 .....	679
23.3	TIM16/TIM17 の主な特長 .....	680
23.4	TIM15/TIM16/TIM17 機能詳細 .....	683
23.4.1	タイムベースユニット .....	683
23.4.2	カウンタモード .....	685
23.4.3	繰り返しカウンタ .....	689
23.4.4	クロック選択 .....	690
23.4.5	キャプチャ / 比較チャネル .....	692
23.4.6	入力キャプチャモード .....	695
23.4.7	PWM 入力モード (TIM15 の場合のみ) .....	696



23.4.8	強制出力モード	697
23.4.9	出力比較モード	697
23.4.10	PWM モード	698
23.4.11	組み合わせ PWM モード (TIM15 のみ)	699
23.4.12	相補出力とデッドタイム挿入	700
23.4.13	ブレーク機能の使用	703
23.4.14	ワンパルスモード	706
23.4.15	UIF ビットの再配置	707
23.4.16	タイマ入力 XOR 機能 (TIM15 のみ)	708
23.4.17	外部トリガ同期 (TIM15 のみ)	709
23.4.18	スレーブモード: リセットモードとトリガモードの組み合わせ (TIM15 のみ)	711
23.4.19	DMA バーストモード	711
23.4.20	タイマ同期 (TIM15)	713
23.4.21	デバッグモード	713
23.5	TIM15 レジスタ	714
23.5.1	TIM15 制御レジスタ 1 (TIM15_CR1)	714
23.5.2	TIM15 制御レジスタ 2 (TIM15_CR2)	715
23.5.3	TIM15 のスレーブモード制御レジスタ (TIM15_SMCR)	717
23.5.4	TIM15 DMA / 割り込み有効レジスタ (TIM15_DIER)	718
23.5.5	TIM15 ステータスレジスタ (TIM15_SR)	719
23.5.6	TIM15 のイベント生成レジスタ (TIM15_EGR)	721
23.5.7	TIM15 のキャプチャ/比較モードレジスタ 1 (TIM15_CCMR1)	722
23.5.8	TIM15 のキャプチャ/比較有効レジスタ (TIM15_CCER)	726
23.5.9	TIM15 のカウンタ (TIM15_CNT)	729
23.5.10	TIM15 のプリスケラ (TIM15_PSC)	729
23.5.11	TIM15 の自動再ロードレジスタ (TIM15_ARR)	729
23.5.12	TIM15 繰り返しカウンタレジスタ (TIM15_RCR)	730
23.5.13	TIM15 のキャプチャ/比較レジスタ 1 (TIM15_CCR1)	730
23.5.14	TIM15 のキャプチャ/比較レジスタ 2 (TIM15_CCR2)	731
23.5.15	TIM15 ブレークおよびデッドタイムレジスタ (TIM15_BDTR)	731
23.5.16	TIM15 DMA 制御レジスタ (TIM15_DCR)	733
23.5.17	完全転送の TIM15 DMA アドレス (TIM15_DMAR)	734
23.5.18	TIM15 レジスタマップ	734
23.6	TIM16/TIM17 レジスタ	736
23.6.1	TIM16/TIM17 制御レジスタ 1 (TIMx_CR1)	736
23.6.2	TIM16/TIM17 制御レジスタ 2 (TIMx_CR2)	737

23.6.3	TIM16/TIM17 DMA / 割り込み有効レジスタ (TIMx_DIER) .....	738
23.6.4	TIM16/TIM17 ステータスレジスタ (TIMx_SR) .....	739
23.6.5	TIM16/TIM17 イベント生成レジスタ (TIMx_EGR) .....	740
23.6.6	TIM16/TIM17 キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) .....	741
23.6.7	TIM16/TIM17 キャプチャ/比較有効レジスタ (TIMx_CCER) .....	744
23.6.8	TIM16/TIM17 カウンタ (TIMx_CNT) .....	747
23.6.9	TIM16/TIM17 プリスケーラ (TIMx_PSC) .....	747
23.6.10	TIM16/TIM17 自動再ロードレジスタ (TIMx_ARR) .....	747
23.6.11	TIM16/TIM17 繰り返しカウンタレジスタ (TIMx_RCR) .....	748
23.6.12	TIM16/TIM17 キャプチャ/比較レジスタ 1 (TIMx_CCR1) .....	748
23.6.13	TIM16/TIM17 ブレークおよびデッドタイムレジスタ (TIMx_BDTR) .....	749
23.6.14	TIM16/TIM17 DMA 制御レジスタ (TIMx_DCR) .....	751
23.6.15	完全転送の TIM16/TIM17 DMA アドレス (TIMx_DMAR) .....	751
23.6.16	TIM16 オプションレジスタ (TIM16_OR) .....	752
23.6.17	TIM16/TIM17 レジスタマップ .....	753
<b>24</b>	<b>赤外線インタフェース (IRTIM) .....</b>	<b>755</b>
<b>25</b>	<b>独立型ウォッチドッグ (IWDG) .....</b>	<b>756</b>
25.1	概要 .....	756
25.2	IWDG の主な機能 .....	756
25.3	IWDG の機能説明 .....	756
25.3.1	IWDG ブロック図 .....	756
25.3.2	ウィンドウオプション .....	757
25.3.3	ハードウェアウォッチドッグ .....	758
25.3.4	STOP および STANDBY モードでの動作 .....	758
25.3.5	レジスタのアクセス保護 .....	758
25.3.6	デバッグモード .....	758
25.4	IWDG レジスタ .....	759
25.4.1	キーレジスタ (IWDG_KR) .....	759
25.4.2	プリスケーラレジスタ (IWDG_PR) .....	760
25.4.3	再ロードレジスタ (IWDG_RLR) .....	761
25.4.4	ステータスレジスタ (IWDG_SR) .....	762
25.4.5	ウィンドウレジスタ (IWDG_WINR) .....	763
25.4.6	IWDG レジスタマップ .....	764



<b>26</b>	<b>システムウィンドウ型ウォッチドッグ (WWDG) .....</b>	<b>765</b>
26.1	概要 .....	765
26.2	WWDG の主な機能 .....	765
26.3	WWDG の機能説明 .....	765
26.3.1	ウォッチドッグの有効化 .....	766
26.3.2	ダウンカウンタの制御 .....	766
26.3.3	高度なウォッチドッグ割り込み機能 .....	766
26.3.4	ウォッチドッグタイムアウトをプログラムする方法 .....	767
26.3.5	デバッグモード .....	768
26.4	WWDG レジスタ .....	769
26.4.1	制御レジスタ (WWDG_CR) .....	769
26.4.2	設定レジスタ (WWDG_CFR) .....	770
26.4.3	ステータスレジスタ (WWDG_SR) .....	770
26.4.4	WWDG レジスタマップ .....	771
<b>27</b>	<b>リアルタイムクロック (RTC) .....</b>	<b>772</b>
27.1	概要 .....	772
27.2	RTC の主な機能 .....	773
27.3	RTC の機能説明 .....	774
27.3.1	RTC ブロック図 .....	774
27.3.2	RTC によって制御される GPIO .....	775
27.3.3	クロックとプリスケアラ .....	777
27.3.4	リアルタイムクロックとカレンダー .....	777
27.3.5	プログラム可能なアラーム .....	778
27.3.6	周期的自動ウェイクアップ .....	778
27.3.7	RTC の初期化と設定 .....	779
27.3.8	カレンダーの読み出し .....	780
27.3.9	RTC のリセット .....	781
27.3.10	RTC の同期 .....	782
27.3.11	RTC リファレンスクロック検出 .....	782
27.3.12	RTC の高精度デジタル較正 .....	783
27.3.13	タイムスタンプ機能 .....	785
27.3.14	タンパ検出 .....	785
27.3.15	較正クロック出力 .....	787
27.3.16	アラーム出力 .....	787

27.4	RTC 低電力モード	788
27.5	RTC 割り込み	788
27.6	RTC レジスタ	789
27.6.1	RTC 時刻レジスタ (RTC_TR)	789
27.6.2	RTC 日付レジスタ (RTC_DR)	790
27.6.3	RTC 制御レジスタ (RTC_CR)	791
27.6.4	RTC 初期化とステータスレジスタ (RTC_ISR)	794
27.6.5	RTC プリスケアラレジスタ (RTC_PRER)	797
27.6.6	RTC ウェイクアップタイムレジスタ (RTC_WUTR)	798
27.6.7	RTC アラーム A レジスタ (RTC_ALRMAR)	799
27.6.8	RTC アラーム B レジスタ (RTC_ALRMBR)	800
27.6.9	RTC 書き込み保護レジスタ (RTC_WPR)	801
27.6.10	RTC サブセカンドレジスタ (RTC_SSR)	801
27.6.11	RTC シフト制御レジスタ (RTC_SHIFTR)	802
27.6.12	RTC タイムスタンプ時刻レジスタ (RTC_TSTR)	803
27.6.13	RTC タイムスタンプ日付レジスタ (RTC_TSDR)	804
27.6.14	RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)	805
27.6.15	RTC 較正レジスタ (RTC_CALR)	806
27.6.16	RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)	807
27.6.17	RTC アラーム A サブセカンドレジスタ (RTC_ALRMSSR)	810
27.6.18	RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)	811
27.6.19	RTC バックアップレジスタ (RTC_BKPxR)	811
27.6.20	RTC レジスタマップ	812
<b>28</b>	<b>I2C (Inter-integrated circuit) インタフェース</b>	<b>814</b>
28.1	概要	814
28.2	I2C の主な機能	814
28.3	I2C の実装	815
28.4	I2C の機能詳細	815
28.4.1	I2C ブロック図	816
28.4.2	I2C クロックの要件	817
28.4.3	モード選択	817
28.4.4	I2C の初期化	819
28.4.5	ソフトウェアリセット	823
28.4.6	データ転送	824
28.4.7	I2C スレーブモード	826



28.4.8	I2C マスタモード	835
28.4.9	I2C_TIMINGR レジスタの設定例	847
28.4.10	SMBus 固有の機能	848
28.4.11	SMBus 初期化	851
28.4.12	SMBus : I2C_TIMEOUTR レジスタの設定例	853
28.4.13	SMBus スレーブモード	854
28.4.14	アドレス一致時に STOP モードからウェイクアップ	861
28.4.15	エラー条件	861
28.4.16	DMA リクエスト	863
28.4.17	デバッグモード	864
28.5	I2C 低電力モード	864
28.6	I2C 割り込み	864
28.7	I2C レジスタ	866
28.7.1	制御レジスタ 1 (I2C_CR1)	866
28.7.2	制御レジスタ 2 (I2C_CR2)	869
28.7.3	Own Address 1 レジスタ (I2C_OAR1)	872
28.7.4	Own Address 2 レジスタ (I2C_OAR2)	873
28.7.5	タイミングレジスタ (I2C_TIMINGR)	874
28.7.6	タイムアウトレジスタ (I2C_TIMEOUTR)	875
28.7.7	割り込みおよびステータスレジスタ (I2C_ISR)	876
28.7.8	割り込みクリアレジスタ (I2C_ICR)	878
28.7.9	PEC レジスタ (I2C_PECR)	879
28.7.10	受信データレジスタ (I2C_RXDR)	880
28.7.11	送信データレジスタ (I2C_TXDR)	880
28.7.12	I2C レジスタマップ	881

## 29 USART

### (Universal synchronous asynchronous receiver transmitter) . . . . . 883

29.1	概要	883
29.2	USART の主な機能	883
29.3	USART の拡張機能	884
29.4	USART の実装	885
29.5	USART の機能詳細	886
29.5.1	USART キャラクタの説明	888
29.5.2	USART トランスミッタ	890
29.5.3	USART レシーバ	892

29.5.4	USART ボーレート生成	898
29.5.5	クロック偏差に対する USART レシーバの許容誤差	900
29.5.6	USART 自動ボーレート検出	902
29.5.7	USART を使用したマルチプロセッサ通信	903
29.5.8	USART を使用した Modbus 通信	905
29.5.9	USART パリティ制御	906
29.5.10	USART LIN (Local Interconnection Network) モード	907
29.5.11	USART 同期モード	909
29.5.12	USART 単線半二重通信	912
29.5.13	USART スマートカードモード	912
29.5.14	USART IrDA SIR ENDEC ブロック	917
29.5.15	DMA モードでの USART 連続通信	919
29.5.16	USART を使用した RS232 ハードウェアフロー制御および RS485 ドライバ有効	921
29.5.17	USART を使用した STOP モードからのウェイクアップ	923
29.6	USART 低電力モード	925
29.7	USART 割り込み	925
29.8	USART レジスタ	927
29.8.1	制御レジスタ 1 (USART_CR1)	927
29.8.2	制御レジスタ 2 (USART_CR2)	930
29.8.3	制御レジスタ 3 (USART_CR3)	934
29.8.4	ボーレートレジスタ (USART_BRR)	938
29.8.5	ガード時間およびプリスケアラレジスタ (USART_GTPR)	938
29.8.6	レシーバタイムアウトレジスタ (USART_RTOR)	939
29.8.7	リクエストレジスタ (USART_RQR)	940
29.8.8	割り込みおよびステータスレジスタ (USART_ISR)	941
29.8.9	割り込みフラグクリアレジスタ (USART_ICR)	946
29.8.10	受信データレジスタ (USART_RDR)	947
29.8.11	送信データレジスタ (USART_TDR)	947
29.8.12	USART レジスタマップ	948

<b>30</b>	<b>シリアルペリフェラルインタフェース / I2S (SPI/I2S) .....</b>	<b>950</b>
30.1	概要 .....	950
30.2	SPI の主な機能 .....	950
30.3	I2S の主な機能 .....	951
30.4	SPI/I2S の実装 .....	951
30.5	SPI の機能説明 .....	952
30.5.1	概要 .....	952
30.5.2	マスタとスレーブの 1 対 1 の通信 .....	953
30.5.3	標準マルチスレーブ通信 .....	955
30.5.4	マルチマスタ通信 .....	956
30.5.5	スレーブ選択 (NSS) ピンの管理 .....	957
30.5.6	通信フォーマット .....	958
30.5.7	SPI の設定 .....	960
30.5.8	SPI を有効にする手順 .....	961
30.5.9	データの送受信手順 .....	961
30.5.10	SPI ステータスフラグ .....	971
30.5.11	SPI エラーフラグ .....	972
30.5.12	NSS パルスモード .....	973
30.5.13	TI モード .....	973
30.5.14	CRC 計算 .....	974
30.6	SPI 割り込み .....	976
30.7	I <sup>2</sup> S の機能説明 (STM32F303xB/C/D/E、STM32F358xC および STM32F398xE のみ) .....	977
30.7.1	I <sup>2</sup> S の概要 .....	977
30.7.2	I2S 全二重 .....	978
30.7.3	サポートされるオーディオプロトコル .....	979
30.7.4	起動に関する説明 .....	986
30.7.5	クロックジェネレータ .....	987
30.7.6	I <sup>2</sup> S マスタモード .....	989
30.7.7	I <sup>2</sup> S スレーブモード .....	991
30.7.8	I <sup>2</sup> S ステータスフラグ .....	993
30.7.9	I <sup>2</sup> S エラーフラグ .....	994
30.7.10	DMA の機能 .....	994
30.8	I <sup>2</sup> S 割り込み .....	995
30.9	SPI および I <sup>2</sup> S レジスタ .....	996
30.9.1	SPI 制御レジスタ 1 (SPIx_CR1) .....	996



30.9.2	SPI 制御レジスタ 2 (SPIx_CR2) .....	998
30.9.3	SPI ステータスレジスタ (SPIx_SR) .....	1001
30.9.4	SPI データレジスタ (SPIx_DR) .....	1002
30.9.5	SPI CRC 多項式レジスタ (SPIx_CRCPR) .....	1003
30.9.6	SPI Rx CRC レジスタ (SPIx_RXCRCR) .....	1004
30.9.7	SPI Tx CRC レジスタ (SPIx_TXCRCR) .....	1004
30.9.8	SPIx_I <sup>2</sup> S 設定レジスタ (SPIx_I2SCFGR) .....	1005
30.9.9	SPIx_I <sup>2</sup> S プリスケアラレジスタ (SPIx_I2SPR) .....	1007
30.9.10	SPI/I2S レジスタマップ .....	1008
<b>31</b>	<b>コントローラエリアネットワーク (bxCAN) .....</b>	<b>1009</b>
31.1	概要 .....	1009
31.2	bxCAN の主な機能 .....	1009
31.3	bxCAN の概要 .....	1010
31.3.1	CAN 2.0B アクティブコア .....	1010
31.3.2	制御レジスタ、ステータスレジスタ、設定レジスタ .....	1010
31.3.3	送信メールボックス .....	1010
31.3.4	受信フィルタ .....	1011
31.4	bxCAN 動作モード .....	1011
31.4.1	初期化モード .....	1011
31.4.2	通常モード .....	1012
31.4.3	スリープモード (低電力) .....	1012
31.5	テストモード .....	1013
31.5.1	サイレントモード .....	1013
31.5.2	ループバックモード .....	1014
31.5.3	サイレントモードとループバックモードの組み合わせ .....	1014
31.6	デバッグモードでの挙動 .....	1015
31.7	bxCAN 機能の説明 .....	1015
31.7.1	送信処理 .....	1015
31.7.2	タイムトリガ通信モード .....	1017
31.7.3	受信処理 .....	1017
31.7.4	識別子フィルタリング .....	1018
31.7.5	メッセージストレージ .....	1022
31.7.6	エラー管理 .....	1024
31.7.7	ビットタイミング .....	1024



31.8	bxCAN の割り込み	1027
31.9	CAN レジスタ	1028
31.9.1	レジスタのアクセス保護	1028
31.9.2	CAN 制御/ステータスレジスタ	1029
31.9.3	CAN メールボックスレジスタ	1039
31.9.4	CAN フィルタレジスタ	1046
31.9.5	bxCAN レジスタマップ	1050
<b>32</b>	<b>ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)</b>	<b>1054</b>
32.1	概要	1054
32.2	USB の主な機能	1054
32.3	USB の実装	1054
32.4	USART の機能詳細	1055
32.4.1	USB ブロックの説明	1056
32.5	プログラミングに関する考慮事項	1058
32.5.1	汎用 USB デバイスのプログラミング	1058
32.5.2	システムリセットとパワーオンリセット	1058
32.5.3	ダブルバッファエンドポイント	1063
32.5.4	アイソクロナス転送	1065
32.5.5	サスペンド/レジュームイベント	1067
32.6	USB レジスタ	1069
32.6.1	共通レジスタ	1069
32.6.2	バッファディスクリプタテーブル	1081
32.6.3	USB レジスタマップ	1085
<b>33</b>	<b>デバッグサポート (DBG)</b>	<b>1087</b>
33.1	概要	1087
33.2	ARM® リファレンス資料	1088
33.3	SWJ デバッグポート (シリアルワイヤと JTAG)	1088
33.3.1	JTAG-DP または SW-DP の選択メカニズム	1089
33.4	ピン名とデバッグポートピン	1089
33.4.1	SWJ デバッグポートピン	1090
33.4.2	柔軟性の高い SWJ-DP ピンの割り当て	1090
33.4.3	JTAG ピンでの内部プルアップ/プルダウン	1091

33.4.4	シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法	1092
33.5	STM32F3xxJTAG TAP 接続	1092
33.6	ID コードとロック機構	1093
33.6.1	MCU デバイス ID コード	1094
33.6.2	バウンダリスキャン TAP	1094
33.6.3	Cortex-M4 <sup>®</sup> F TAP	1094
33.6.4	Cortex-M4 <sup>®</sup> F JEDEC-106 ID コード	1095
33.7	JTAG デバッグポート	1095
33.8	SW デバッグポート	1097
33.8.1	SW プロトコルの概要	1097
33.8.2	SW プロトコルシーケンス	1097
33.8.3	SW-DP ステートマシン (リセット、アイドル状態、ID コード)	1098
33.8.4	DP と AP の読み出し/書き込みアクセス	1098
33.8.5	SW-DP レジスタ	1099
33.8.6	SW-AP レジスタ	1100
33.9	AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効	1100
33.10	コアデバッグ	1101
33.11	システムリセット中のデバッグホスト接続機能	1101
33.12	FPB (フラッシュパッチブレークポイント)	1102
33.13	DWT (データウォッチポイントトリガ)	1103
33.14	ITM (計測トレースマクロセル)	1103
33.14.1	概要	1103
33.14.2	タイムスタンプパケット、同期およびオーバーフローパケット	1103
33.15	ETM (組み込みトレースマクロセル)	1105
33.15.1	概要	1105
33.15.2	信号プロトコル、パケットタイプ	1105
33.15.3	主な ETM レジスタ	1106
33.15.4	設定例	1106
33.16	MCU デバッグコンポーネント (DBGMCU)	1106
33.16.1	低電力モードのデバッグサポート	1107
33.16.2	タイマ、ウォッチドッグ、bxCAN、および I <sup>2</sup> C のデバッグサポート	1107
33.16.3	デバッグ MCU 設定レジスタ	1107
33.16.4	デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ)	1109
33.16.5	デバッグ MCU APB2 凍結レジスタ (DBGMCU_APB2_FZ)	1111



33.17	TPIU (トレースポートインタフェースユニット)	1111
33.17.1	概要	1111
33.17.2	TRACE ピンの割当て	1112
33.17.3	TPIU フォーマッタ	1114
33.17.4	TPIU フレーム同期パケット	1114
33.17.5	同期フレームパケットの送信	1115
33.17.6	同期モード	1115
33.17.7	非同期モード	1115
33.17.8	STM32F3xx 内の TRACECLKIN 接続	1115
33.17.9	TPIU レジスタ	1116
33.17.10	設定例	1116
33.18	DBG レジスタマップ	1117
<b>34</b>	<b>デバイス電子署名</b>	<b>1118</b>
34.1	ユニークデバイス ID レジスタ (96 ビット)	1118
34.2	メモリサイズデータレジスタ	1119
34.2.1	フラッシュサイズデータレジスタ	1119
<b>35</b>	<b>改版履歴</b>	<b>1120</b>

## 表の一覧

表 1.	各製品で利用可能な機能一覧	43
表 2.	STM32F303xB/C および STM32F358xC ペリフェラルレジスタ境界アドレス	50
表 3.	STM32F303xD/E および STM32F398xE ペリフェラルレジスタ境界アドレス	52
表 4.	STM32F303x6/8 および STM32F328x8 ペリフェラルレジスタ境界アドレス	55
表 5.	CCM SRAM の構成	59
表 6.	ブートモード	60
表 7.	フラッシュモジュールの構成	63
表 8.	フラッシュメモリの読み出し保護ステータス	72
表 9.	アクセス状態対保護レベルと実行モード	74
表 10.	フラッシュ割り込みリクエスト	75
表 11.	フラッシュインタフェース - レジスタマップとリセット値	81
表 12.	オプションバイトのフォーマット	83
表 13.	オプションバイトの構成	83
表 14.	オプションバイトの説明	84
表 15.	CRC レジスタマップとリセット値	91
表 16.	低電力モードの概要	98
表 17.	Sleep-now	100
表 18.	Sleep-on-exit	100
表 19.	STOP モード	102
表 20.	STANDBY モード	103
表 21.	PWR レジスタマップとリセット値	108
表 22.	STM32F3xx ペリフェラル相互接続マトリックス	109
表 23.	ADCx アナログウォッチドッグへの TIM1/8/20_ETR の接続	113
表 24.	VREFOPAMPx から ADC チャンネル	114
表 25.	ADC 入力への OPAMP 出力	114
表 26.	タイマ入力へのコンパレータの出力	115
表 27.	コンパレータのブランキングソースとしてのタイマ出力選択	116
表 28.	コンパレータの反転入力としての DAC 出力選択	117
表 29.	OPAMP の非反転入力としての DAC 出力選択	117
表 30.	タイマの同期	118
表 31.	DAC 変換をトリガするタイマおよび EXTI 信号	120
表 32.	RCC レジスタマップとリセット値	165
表 33.	NOR/PSRAM バンク選択	171
表 34.	NOR/PSRAM 外部メモリアドレス	171
表 35.	NAND/PC カードメモリマッピングおよびタイミングレジスタ	172
表 36.	NAND バンク選択	172
表 37.	プログラム可能な NOR/PSRAM のアクセスパラメータ	174
表 38.	非マルチプレクス I/O NOR フラッシュメモリ	174
表 39.	16 ビットマルチプレクス I/O NOR フラッシュメモリ	175
表 40.	非マルチプレクス I/O PSRAM/SRAM	175
表 41.	16 ビットマルチプレクス I/O PSRAM	176
表 42.	NOR フラッシュ /PSRAM : サポートされているメモリとトランザクションの例	176
表 43.	FMC_BCRx ビットフィールド	179
表 44.	FMC_BTRx ビットフィールド	180
表 45.	FMC_BCRx ビットフィールド	181
表 46.	FMC_BTRx ビットフィールド	182
表 47.	FMC_BWTRx ビットフィールド	182
表 48.	FMC_BCRx ビットフィールド	184
表 49.	FMC_BTRx ビットフィールド	185



表 50.	FMC_BWTRx ビットフィールド	185
表 51.	FMC_BCRx ビットフィールド	187
表 52.	FMC_BTRx ビットフィールド	187
表 53.	FMC_BWTRx ビットフィールド	188
表 54.	FMC_BCRx ビットフィールド	189
表 55.	FMC_BTRx ビットフィールド	190
表 56.	FMC_BWTRx ビットフィールド	190
表 57.	FMC_BCRx ビットフィールド	192
表 58.	FMC_BTRx ビットフィールド	192
表 59.	FMC_BCRx ビットフィールド	197
表 60.	FMC_BTRx ビットフィールド	198
表 61.	FMC_BCRx ビットフィールド	199
表 62.	FMC_BTRx ビットフィールド	200
表 63.	プログラム可能な NAND フラッシュ /PC カードアクセスパラメータ	209
表 64.	8 ビット NAND 型フラッシュ	209
表 65.	16 ビット NAND 型フラッシュ	210
表 66.	16 ビット PC カード	210
表 67.	サポートされるメモリおよびトランザクション	211
表 68.	16 ビット PC- カード信号とアクセスタイプ	216
表 69.	ECC 結果関連ビット	223
表 70.	FMC レジスタマップ	224
表 71.	ポートビット設定表	228
表 72.	GPIO レジスタマップとリセット値	241
表 73.	SYSCFG レジスタマップとリセット値	259
表 74.	DMA の実装	262
表 75.	プログラム可能なデータ幅およびエンディアンの動作 (ビット PINC = MINC = 1 の場合)	265
表 76.	DMA 割り込みリクエスト	266
表 77.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE でのチャンネルごとの DMA1 リクエストの概要	270
表 78.	STM32F303x6/8 および STM32F328x8 のチャンネルごとの DMA1 リクエストの概要	270
表 79.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE のチャンネルごとの DMA2 リクエストの概要	273
表 80.	DMA レジスタマップとリセット値	280
表 81.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブル	283
表 82.	STM32F303x6/8 および STM32F328x8 ベクタテーブル	287
表 83.	外部割り込み/イベントコントローラのレジスタマップとリセット値	301
表 84.	ADC 外部チャンネルマッピング	304
表 85.	ADC 内部チャンネルの概要	305
表 86.	ADC 内部信号	307
表 87.	ADC ピン	307
表 88.	レギュラ外部トリガのトリガ極性の設定	324
表 89.	ADC1 (マスタ) および 2 (スレーブ) - レギュラチャンネルの外部トリガ	325
表 90.	ADC1 および ADC2 - インジェクトチャンネルの外部トリガ	326
表 91.	ADC3 および ADC4 - レギュラチャンネルの外部トリガ	327
表 92.	ADC3 および ADC4 - インジェクトチャンネルの外部トリガ	327
表 93.	TSAR タイミングは分解能に依存	338
表 94.	オフセット計算対データ分解能	341
表 95.	アナログウォッチドッグチャンネル選択	351
表 96.	アナログウォッチドッグ 1 の比較	352
表 97.	アナログウォッチドッグ 2 および 3 の比較	352
表 98.	ADC ごとの ADC 割り込み	372
表 99.	DELAY ビット対 ADC 分解能	405

表 100.	ADC グローバルレジスタマップ	406
表 101.	ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100、x=1..4)	407
表 102.	ADC レジスタマップとリセット値 (マスタおよびスレーブ ADC 共通レジスタ) オフセット =0x300、x=1 または 34)	409
表 103.	DAC1 ピン	412
表 104.	外部トリガ (DAC1)	415
表 105.	外部トリガ (DAC2)	415
表 106.	DAC レジスタマップ とリセット値	434
表 107.	コンパレータの入出力の概要	440
表 108.	COMP レジスタマップとリセット値	460
表 109.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE の専用 I/O との接続	463
表 110.	STM32F303x6/8 および STM32F328x8 の専用 I/O との接続	463
表 111.	OPAMP レジスタマップとリセット値	482
表 112.	取得シーケンスの概要	486
表 113.	スペクトル拡散偏差と AHB クロック周波数	488
表 114.	モードおよび IODEF ビットの値に応じた I/O の状態	489
表 115.	低電力モードが TSC に与える影響	491
表 116.	割り込み制御ビット	491
表 117.	TSC レジスタマップとリセット値	500
表 118.	タイマ出力と BRK/BRK2 入力の動作	543
表 119.	カウント方向とエンコーダ信号	550
表 120.	TIMx 内部トリガ接続	567
表 121.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット	581
表 122.	TIM1/TIM8/TIM20 レジスタマップとリセット値	594
表 123.	カウント方向とエンコーダ信号	630
表 124.	TIMx 内部トリガ接続	647
表 125.	標準 OCx チャンネルの出力制御ビット	659
表 126.	TIM2/TIM3/TIM4 レジスタマップとリセット値	664
表 127.	TIM6/TIM7 レジスタマップとリセット値	678
表 128.	TIMx 内部トリガ接続	718
表 129.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット	728
表 130.	TIM15 レジスタマップとリセット値	734
表 131.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット	746
表 132.	TIM16/TIM17 レジスタマップとリセット値	753
表 133.	IWDG レジスタマップとリセット値	764
表 134.	WWDG レジスタマップとリセット値	771
表 135.	RTC ピン PC13 の設定	776
表 136.	LSE ピン PC14 の設定	776
表 137.	LSE ピン PC15 の設定	776
表 138.	低電力モードが RTC に与える影響	788
表 139.	割り込み制御ビット	789
表 140.	RTC レジスタマップとリセット値	812
表 141.	STM32F3xxI2C の実装	815
表 142.	アナログフィルタとデジタルフィルタの比較	819
表 143.	I2C-SMBUS 仕様のデータのセットアップおよびホールド時間	822
表 144.	I2C 設定表	826
表 145.	I2C-SMBUS 仕様のクロックタイミング	837
表 146.	fI2CCLK = 8 MHz でのタイミング設定の例	847
表 147.	fI2CCLK = 48 MHz でのタイミング設定の例	847
表 148.	SMBus タイムアウト仕様	850
表 149.	SMBUS の PEC 設定	852



表 150.	さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{\text{TIMEOUT}} = 25 \text{ ms}$ )	853
表 151.	さまざまな I2CCLK 周波数での TIMEOUTB の設定例	853
表 152.	さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{\text{IDLE}} = 50 \text{ ms}$ )	853
表 153.	低電力モード	864
表 154.	I2C 割り込みリクエスト	864
表 155.	I2C レジスタマップとリセット値	881
表 156.	STM32F3xx USART の機能	885
表 157.	サンプリングされたデータからのノイズ検出	897
表 158.	16 倍または 8 倍のオーバーサンプリングの両方の場合の $f_{\text{CK}} = 72\text{MHz}$ でのプログラムされた ポーレートの誤差計算	900
表 159.	BRR [3:0] = 0000 のときの USART レシーバの許容誤差	901
表 160.	BRR[3:0] が 0000 でないときの USART レシーバの許容誤差	901
表 161.	フレームフォーマット	906
表 162.	低電力モードが USART に与える影響	925
表 163.	USART 割り込みリクエスト	925
表 164.	USART レジスタマップとリセット値	948
表 165.	STM32F303x6/8 および STM32F328x8 SPI 実装	951
表 166.	STM32F303xB/C/D/E、STM32F358xC および STM32F398xE SPI 実装	952
表 167.	SPI 割り込みリクエスト	976
表 168.	標準 8 MHz HSE を使用した場合のオーディオ周波数精度	989
表 169.	I <sup>2</sup> S 割り込みリクエスト	995
表 170.	SPI レジスタマップとリセット値	1008
表 171.	送信メールボックスの配置	1023
表 172.	受信メールボックスの配置	1023
表 173.	bxCAN レジスタマップとリセット値	1050
表 174.	STM32F3xx USB の実装	1054
表 175.	ダブルバッファリングバッファフラグの定義	1064
表 176.	パルクダブルバッファリングメモリバッファの用途	1064
表 177.	アイソクロナスメモリバッファの使用	1066
表 178.	レジャーイベント検出	1068
表 179.	受信ステータスエンコード	1080
表 180.	エンドポイントタイプエンコード	1080
表 181.	エンドポイントの種類の意味	1080
表 182.	送信ステータスエンコード	1080
表 183.	割り当てられるバッファメモリの定義	1084
表 184.	USB レジスタマップとリセット値	1085
表 185.	SWJ デバッグポートピン	1090
表 186.	柔軟性の高い SWJ-DP ピンの割り当て	1090
表 187.	JTAG デバッグポートのデータレジスタ	1095
表 188.	シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ	1096
表 189.	パケットリクエスト (8 ビット)	1097
表 190.	ACK 応答 (3 ビット)	1098
表 191.	データ転送 (33 ビット)	1098
表 192.	SW-DP レジスタ	1099
表 193.	Cortex-M4 <sup>®</sup> F AHB-AP レジスタ	1100
表 194.	コアデバッグレジスタ	1101
表 195.	主な ITM レジスタ	1104
表 196.	主な ETM レジスタ	1106
表 197.	非同期 TRACE ピンの割り当て	1112
表 198.	同期 TRACE ピンの割り当て	1112
表 199.	柔軟性の高い TRACE ピン割り当て	1113
表 200.	重要な TPIU レジスタ	1116



# 参考資料

RM0316

表の一覧

表 201.	DBG レジスタマップとリセット値.....	1117
表 202.	文書改版履歴.....	1120



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

## 図の一覧

図 1.	STM32F303xB/C および STM32F358xC システムアーキテクチャ	47
図 2.	STM32F303x6/8 および STM32F328x8 システムアーキテクチャ	47
図 3.	STM32F303xDxE および STM32F398xE のシステムアーキテクチャ	48
図 4.	プログラミング手順	67
図 5.	フラッシュメモリのページ消去手順	69
図 6.	フラッシュメモリの全体消去手順	70
図 7.	CRC 計算ユニットのブロック図	87
図 8.	電源の概要 (STM32F303x デバイス)	92
図 9.	電源の概要 (STM32F3x8 デバイス)	93
図 10.	パワーオンリセット/パワーダウンリセット波形	96
図 11.	PVD の閾値	97
図 12.	簡略化されたリセット回路図	122
図 13.	STM32F303xB/C および STM32F358xC クロックツリー	124
図 14.	STM32F303xDxE および STM32F398xE のクロックツリー	125
図 15.	STM32F303x6/8 および STM32F328x8 クロックツリー	126
図 16.	HSE/LSE クロックソース	127
図 17.	キャプチャモードにおける TIM16 を使用した周波数測定	133
図 18.	FMC ブロック図	168
図 19.	FMC メモリバンク	171
図 20.	モード 1 読み出しアクセス波形	178
図 21.	モード 1 書き込みアクセス波形	179
図 22.	モード A 読み出しアクセス波形	180
図 23.	モード A 書き込みアクセス波形	181
図 24.	モード 2 およびモード B 読み出しアクセス波形	183
図 25.	モード 2 書き込みアクセス波形	183
図 26.	モード B 書き込みアクセス波形	184
図 27.	モード C 読み出しアクセス波形	186
図 28.	モード C 書き込みアクセス波形	186
図 29.	モード D 読み出しアクセス波形	188
図 30.	モード D 書き込みアクセス波形	189
図 31.	マルチプレクス読み出しアクセス波形	191
図 32.	マルチプレクス書き込みアクセス波形	191
図 33.	読み出しアクセス中の非同期ウェイト波形	194
図 34.	書き込みアクセス中の非同期ウェイト波形	194
図 35.	待ち設定波形	196
図 36.	同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)	197
図 37.	同期マルチプレクス書き込みモード波形 - PSRAM (CRAM)	199
図 38.	共通メモリアクセスの NAND フラッシュ /PC カードコントローラ波形	212
図 39.	CE が無視されない NAND 型フラッシュへのアクセス	213
図 40.	I/O ポートビットの基本構成	227
図 41.	5 V トレラント I/O ポートビットの基本構成	227
図 42.	入力フローティング/プルアップ/プルダウン設定	232
図 43.	出力設定	233
図 44.	オルタネート機能設定	233
図 45.	ハイインピーダンスアナログ設定	234
図 46.	DMA ブロック図	262
図 47.	STM32F302xB/C/D/E および STM32F302x6/8 の DMA1 リクエストマッピング	268
図 48.	STM32F303x6/8 および STM32F328x8 DMA1 リクエストマッピング	269

図 49.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE での DMA2 リクエストマッピング	272
図 50.	外部割り込み/イベントブロック図	291
図 51.	外部割り込み/イベント GPIO の配置	293
図 52.	ADC ブロック図	306
図 53.	ADC クロック構成	309
図 54.	ADC1 および ADC2 の接続性	310
図 55.	ADC3 および ADC4 の接続性	312
図 56.	ADC 較正	315
図 57.	ADC 較正係数の更新	316
図 58.	シングルエンドチャンネルと差動チャンネルの混合	316
図 59.	ADC の有効化/無効化	317
図 60.	アナログ/デジタル変換時間	322
図 61.	実行中のレギュラ変換の停止	323
図 62.	実行中のレギュラ変換とインジェクト変換の停止	324
図 63.	ADC マスタと ADC スレーブ間でトリガを共有	325
図 64.	インジェクト変換の遅延時間	329
図 65.	コンテキストの JSQR キューの例 (シーケンスの変更)	332
図 66.	コンテキストの JSQR キューの例 (トリガの変更)	332
図 67.	変換前にオーバーフローが発生した場合のコンテキストの JSQR キューの例	333
図 68.	変換中にオーバーフローが発生した場合のコンテキストの JSQR キューの例	333
図 69.	空のキューがあるコンテキストの JSQR キューの例 (JQM=0 の場合)	334
図 70.	空のキューがあるコンテキストの JSQR キューの例 (JQM=1 の場合)	334
図 71.	JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0) 変換の実行中に JADSTP が発生した場合	335
図 72.	JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0) 変換の実行中に JADSTP が発生し、新しいトリガが発生した場合	335
図 73.	JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0) 変換の実行中に外部で JADSTP が発生した場合	336
図 74.	JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=1)	336
図 75.	ADDIS=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0)	337
図 76.	ADDIS=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=1)	337
図 77.	ソフトウェアトリガとハードウェアトリガを変更する場合のコンテキストの JSQR キューの例	338
図 78.	シーケンスのシングル変換、ソフトウェアトリガ	339
図 79.	シーケンスの連続変換、ソフトウェアトリガ	340
図 80.	シーケンスのシングル変換、ハードウェアトリガ	340
図 81.	シーケンスの連続変換、ハードウェアトリガ	340
図 82.	右詰め (オフセット無効、符号なしの値)	342
図 83.	右詰め (オフセット有効、符号付きの値)	342
図 84.	左詰め (オフセット無効、符号なしの値)	343
図 85.	左詰め (オフセット有効、符号付きの値)	343
図 86.	オーバーラン (OVR) の例	344
図 87.	AUTDLY=1、連続モードのレギュラ変換、ソフトウェアトリガ	347
図 88.	AUTDLY=1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN=0、JDISCEN=0)	348
図 89.	AUTDLY=1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN=1、JDISCEN=1)	349
図 90.	AUTDLY=1、インジェクト変換によって中断されたレギュラ連続変換	350
図 91.	自動インジェクトモードの AUTDLY=1 (JAUTO=1)	350
図 92.	アナログウォッチドッグによって保護される領域	351
図 93.	ADCy_AWDx_OUT 信号生成 (すべてのレギュラチャンネル)	353



図 94.	ADCy_AWDx_OUT 信号生成 (ソフトウェアによって AWDx フラグがクリアされない場合) . . . . .	354
図 95.	ADCy_AWDx_OUT 信号生成 (1 つのレギュラチャネル) . . . . .	354
図 96.	ADCy_AWDx_OUT 信号生成 (すべてのインジェクトチャネル) . . . . .	354
図 97.	デュアル ADC ブロック図 <sup>(1)</sup> . . . . .	356
図 98.	4 チャネルのインジェクト同時モード: デュアル ADC モード . . . . .	357
図 99.	16 チャネルのレギュラ同時モード: デュアル ADC モード . . . . .	359
図 100.	連続変換モードにおける 1 チャネルのインタリーブモード: デュアル ADC モード . . . . .	361
図 101.	シングル変換モードにおける 1 チャネルのインタリーブモード: デュアル ADC モード . . . . .	361
図 102.	インジェクションによるインタリーブ変換 . . . . .	362
図 103.	オルタネートトリガ: 各 ADC のインジェクトグループ . . . . .	363
図 104.	オルタネートトリガ: 不連続モードにおける 4 つのインジェクトチャネル (各 ADC) . . . . .	364
図 105.	オルタネート + レギュラ同時 . . . . .	365
図 106.	インジェクト変換中にトリガが発生した場合 . . . . .	365
図 107.	レギュラ同時モードにおける DMA リクエスト (MDMA=0b00) . . . . .	366
図 108.	レギュラ同時モードにおける DMA リクエスト (MDMA=0b10) . . . . .	367
図 109.	インタリーブモードにおける DMA リクエスト (MDMA=0b10) . . . . .	367
図 110.	温度センサチャネルのブロック図 . . . . .	369
図 111.	VBAT チャネルのブロック図 . . . . .	370
図 112.	VREFINT チャネルのブロック図 . . . . .	371
図 113.	DAC1 ブロック図 . . . . .	411
図 114.	DAC2 ブロック図 (STM32F303x6/8 および STM32F328 のみ) . . . . .	412
図 115.	シングル DAC チャネルモードのデータレジスタ . . . . .	413
図 116.	トリガ無効 (TEN = 0) 時の変換タイミング図 . . . . .	414
図 117.	デュアル DAC チャネルモードのデータレジスタ . . . . .	416
図 118.	DAC LFSR レジスタ計算アルゴリズム . . . . .	421
図 119.	LFSR 波形生成による DAC 変換 (SW トリガ有効) . . . . .	421
図 120.	DAC 三角波生成 . . . . .	422
図 121.	三角波生成による DAC 変換 (SW トリガ有効) . . . . .	422
図 122.	コンパレータ 1 および 2 のブロック図 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE) . . . . .	437
図 123.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE コンパレータ 7 のブロック図 . . . . .	438
図 124.	STM32F303x6/8 および STM32F328x8 コンパレータ 2/4/6 のブロック図 . . . . .	439
図 125.	コンパレータヒステリシス . . . . .	441
図 126.	コンパレータの出力のブランキング . . . . .	442
図 127.	STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE コンパレータとオペアンプの 相互接続 (パート 1) . . . . .	464
図 128.	STM32F303xB/C/D/E および STM32F358xC コンパレータとオペアンプの相互接続 (パート 2) . . . . .	465
図 129.	STM32F303x6/8 および STM32F328x8 コンパレータとオペアンプの接続 . . . . .	466
図 130.	タイマによって制御されたマルチプレクサモード . . . . .	468
図 131.	スタンダオンモード: 外部ゲイン設定モード . . . . .	469
図 132.	フォロア設定 . . . . .	470
図 133.	PGA モード、内部ゲイン設定 (x2/x4/x8/x16)、未使用の反転入力 . . . . .	471
図 134.	PGA モード、内部ゲイン設定 (x2/x4/x8/x16)、フィルタリング用の反転入力 . . . . .	471
図 135.	TSC ブロック図 . . . . .	484
図 136.	表面電荷移動アナログ I/O グループの構造 . . . . .	485
図 137.	サンプリングコンデンサの電圧の変化 . . . . .	486
図 138.	電荷移動取得シーケンス . . . . .	487
図 139.	スペクトル拡散の変動の原理 . . . . .	488
図 140.	高機能制御タイマのブロック図 . . . . .	503
図 141.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図 . . . . .	505
図 142.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図 . . . . .	505

図 143.	内部クロック分周比が 1 の場合のカウンタのタイミング図	507
図 144.	内部クロック分周比が 2 の場合のカウンタのタイミング図	507
図 145.	内部クロック分周比が 4 の場合のカウンタのタイミング図	508
図 146.	内部クロック分周比が N の場合のカウンタのタイミング図	508
図 147.	ARPE = 0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	509
図 148.	ARPE = 1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタの タイミング図	509
図 149.	内部クロック分周比が 1 の場合のカウンタのタイミング図	511
図 150.	内部クロック分周比が 2 の場合のカウンタのタイミング図	511
図 151.	内部クロック分周比が 4 の場合のカウンタのタイミング図	512
図 152.	内部クロック分周比が N の場合のカウンタのタイミング図	512
図 153.	繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図 (不使用)	513
図 154.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	514
図 155.	内部クロック分周比が 2 の場合のカウンタのタイミング図	515
図 156.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	515
図 157.	内部クロック分周比が N の場合のカウンタのタイミング図	516
図 158.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図	516
図 159.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	517
図 160.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	518
図 161.	外部トリガ入力ブロック	519
図 162.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	520
図 163.	TI2 外部クロックの接続例	521
図 164.	外部クロックモード 1 の制御回路	522
図 165.	外部トリガ入力ブロック	522
図 166.	外部クロックモード 2 の制御回路	523
図 167.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	524
図 168.	キャプチャ/比較チャンネル 1 メイン回路	525
図 169.	キャプチャ/比較チャンネル (チャンネル 1、同じくチャンネル 2 および 3) の出力ステージ	526
図 170.	キャプチャ/比較チャンネル (チャンネル 4) の出力ステージ	526
図 171.	キャプチャ/比較チャンネル (チャンネル 5、同じくチャンネル 6) の出力ステージ	527
図 172.	PWM 入力モードタイミング	529
図 173.	出力比較モード、OC1 の反転	531
図 174.	エッジアライン PWM 波形 (ARR=8)	532
図 175.	センターアライン PWM 波形 (ARR=8)	533
図 176.	50% デューティサイクルの 2 位相シフトされた PWM 信号の生成	534
図 177.	チャンネル 1 および 3 における組み合わせ PWM モード	536
図 178.	周期ごとの複数トリガパルスを持つ組み合わせ 3 相 PWM 信号	537
図 179.	デッドタイム挿入のある相補出力	538
図 180.	負のパルスより長い遅延があるときのデッドタイムの波形	538
図 181.	正のパルスより長い遅延があるときのデッドタイムの波形	539
図 182.	BKIN (OSSI = 1) でのブレークイベントに対するさまざまな出力の動作	542
図 183.	BKIN および BKIN2 ピンのアサート後の PWM 出力状態 (OSSI=1)	543
図 184.	BKIN アサート後の PWM 出力状態 (OSSI=0)	544
図 185.	TIMx_OCxREF のクリア	545
図 186.	6 ステップ生成 COM の例 (OSSR=1)	546
図 187.	ワンパルスモードの例	547
図 188.	再トリガ可能なワンパルスモード	549
図 189.	エンコーダインタフェースモードにおけるカウンタの動作例	550
図 190.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	551
図 191.	3 つの信号上のエッジ間の時間間隔の測定	552



図 192.	ホールセンサインタフェースの例	554
図 193.	リセットモードの制御回路	555
図 194.	ゲートモードの制御回路	556
図 195.	トリガモードの制御回路	557
図 196.	外部クロックモード 2 + トリガモードの制御回路	558
図 197.	汎用タイマのブロック図	598
図 198.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	600
図 199.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	600
図 200.	内部クロック分周比が 1 の場合のカウンタのタイミング図	601
図 201.	内部クロック分周比が 2 の場合のカウンタのタイミング図	602
図 202.	内部クロック分周比が 4 の場合のカウンタのタイミング図	602
図 203.	内部クロック分周比が N の場合のカウンタのタイミング図	603
図 204.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図	603
図 205.	ARPE=1 (TIMx_ARR はプリロードされる) のときの更新イベント時のカウンタのタイミング図	604
図 206.	内部クロック分周比が 1 の場合のカウンタのタイミング図	605
図 207.	内部クロック分周比が 2 の場合のカウンタのタイミング図	605
図 208.	内部クロック分周比が 4 の場合のカウンタのタイミング図	606
図 209.	内部クロック分周比が N の場合のカウンタのタイミング図	606
図 210.	繰り返しカウンタが使用されていないときの更新イベント時のカウンタのタイミング図	607
図 211.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	608
図 212.	内部クロック分周比が 2 の場合のカウンタのタイミング図	609
図 213.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	609
図 214.	内部クロック分周比が N の場合のカウンタのタイミング図	610
図 215.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図	610
図 216.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	611
図 217.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	612
図 218.	TI2 外部クロックの接続例	612
図 219.	外部クロックモード 1 の制御回路	613
図 220.	外部トリガ入力ブロック	614
図 221.	外部クロックモード 2 の制御回路	615
図 222.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	616
図 223.	キャプチャ/比較チャンネル 1 メイン回路	616
図 224.	キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ	617
図 225.	PWM 入力モードタイミング	619
図 226.	出力比較モード、OC1 の反転	620
図 227.	エッジアライン PWM 波形 (ARR=8)	622
図 228.	センターアライン PWM 波形 (ARR=8)	623
図 229.	50% デューティサイクルの 2 位相シフトされた PWM 信号の生成	624
図 230.	チャンネル 1 および 3 における組み合わせ PWM モード	625
図 231.	TIMx_OCxREF のクリア	626
図 232.	ワンパルスモードの例	627
図 233.	再トリガ可能なワンパルスモード	629
図 234.	エンコーダインタフェースモードにおけるカウンタの動作例	630
図 235.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	631
図 236.	リセットモードの制御回路	632
図 237.	ゲートモードの制御回路	633
図 238.	トリガモードの制御回路	634
図 239.	外部クロックモード 2 + トリガモードの制御回路	635
図 240.	マスタ/スレーブタイマの例	636
図 241.	TIM3 の OC1REF による TIM2 のゲート操作	637

図 242.	TIM3 の有効化による TIM2 のゲート操作	638
図 243.	TIM3 の更新による TIM2 のトリガ	638
図 244.	TIM3 の有効化による TIM2 のトリガ	639
図 245.	TIM3 と TIM2 を TIM3 TI1 入力でトリガします。	640
図 246.	基本タイマブロック図	666
図 247.	プリスケーラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	668
図 248.	プリスケーラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	668
図 249.	内部クロック分周比が 1 の場合のカウンタのタイミング図	669
図 250.	内部クロック分周比が 2 の場合のカウンタのタイミング図	670
図 251.	内部クロック分周比が 4 の場合のカウンタのタイミング図	670
図 252.	内部クロック分周比が N の場合のカウンタのタイミング図	671
図 253.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図	671
図 254.	ARPE = 1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図	672
図 255.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	673
図 256.	TIM15 ブロック図	681
図 257.	TIM16/TIM17 ブロック図	682
図 258.	プリスケーラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	684
図 259.	プリスケーラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	684
図 260.	内部クロック分周比が 1 の場合のカウンタのタイミング図	686
図 261.	内部クロック分周比が 2 の場合のカウンタのタイミング図	686
図 262.	内部クロック分周比が 4 の場合のカウンタのタイミング図	687
図 263.	内部クロック分周比が N の場合のカウンタのタイミング図	687
図 264.	ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図	688
図 265.	ARPE = 1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図	688
図 266.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	689
図 267.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	690
図 268.	TI2 外部クロックの接続例	691
図 269.	外部クロックモード 1 の制御回路	692
図 270.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	693
図 271.	キャプチャ/比較チャンネル 1 メイン回路	693
図 272.	キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ	694
図 273.	キャプチャ/比較チャンネル (TIM15 の場合、チャンネル 2) の出力ステージ	694
図 274.	PWM 入力モードタイミング	696
図 275.	出力比較モード、OC1 の反転	698
図 276.	エッジアライン PWM 波形 (ARR=8)	699
図 277.	チャンネル 1 および 2 における組み合わせ PWM モード	700
図 278.	デッドタイム挿入のある相補出力	701
図 279.	負のパルスより長い遅延があるときのデッドタイムの波形	701
図 280.	正のパルスより長い遅延があるときのデッドタイムの波形	702
図 281.	ブレークに対する出力の動作	705
図 282.	ワンパルスモードの例	706
図 283.	2 つの信号上のエッジ間の時間間隔の測定	708
図 284.	リセットモードの制御回路	709
図 285.	ゲートモードの制御回路	710
図 286.	トリガモードの制御回路	711
図 287.	TIM16 との IR 内部ハードウェア接続	755
図 288.	独立型ウォッチドッグのブロック図	756
図 289.	ウォッチドッグのブロック図	766

図 290.	ウィンドウ型ウォッチドッグのタイミング図	767
図 291.	RTC ブロック図	774
図 292.	I2C ブロック図	816
図 293.	I2C バスプロトコル	818
図 294.	セットアップおよびホールドタイミング	820
図 295.	I2C 初期化フローチャート	823
図 296.	データ受信	824
図 297.	データ送信	825
図 298.	スレーブ初期化フローチャート	829
図 299.	I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0)	830
図 300.	I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1)	831
図 301.	I2C スレーブトランスミッタの転送バス図	832
図 302.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0)	833
図 303.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1)	834
図 304.	I2C スレーブレシーバの転送バス図	834
図 305.	マスタクロック生成	836
図 306.	マスタ初期化フローチャート	838
図 307.	HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス	838
図 308.	HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス	839
図 309.	N≤255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート	840
図 310.	N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート	841
図 311.	I2C マスタトランスミッタの転送バス図	842
図 312.	N≤255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート	844
図 313.	N>255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート	845
図 314.	I2C マスタレシーバの転送バス図	846
図 315.	t <sub>LOW:SEXT</sub> 、t <sub>LOW:MEXT</sub> のタイムアウト間隔	850
図 316.	N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート	854
図 317.	SMBus スレーブトランスミッタの転送バス図 (SBC=1)	855
図 318.	N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート	856
図 319.	SMBus スレーブレシーバのバス転送図 (SBC=1)	857
図 320.	SMBus マスタトランスミッタのバス転送図	858
図 321.	SMBus マスタレシーバのバス転送図	860
図 322.	I2C 割り込みマッピング図	865
図 323.	USART のブロック図	887
図 324.	ワード長のプログラミング	889
図 325.	設定可能なストップビット	891
図 326.	送信時の TC/TXE の動作	892
図 327.	16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出	893
図 328.	データサンプリング (16 倍のオーバーサンプリング)	896
図 329.	データサンプリング (8 倍のオーバーサンプリング)	897
図 330.	アイドルライン検出を使用したミュートモード	904
図 331.	アドレスマーク検出を使用したミュートモード	905
図 332.	LIN モードでのブ레이크検出 (11 ビットブ레이크長、LBDL=1)	908
図 333.	LIN モードでのブ레이크検出とフレーミングエラー検出	909
図 334.	USART の同期送信の例	910
図 335.	USART データクロックタイミング図 (M ビット = 00)	910
図 336.	USART データクロックタイミング図 (M ビット = 01)	911
図 337.	RX データセットアップ/ホールド時間	911
図 338.	ISO 7816-3 非同期プロトコル	913
図 339.	ストップビット 1.5 個を使用したパリティエラー検出	914
図 340.	IrDA SIR ENDEC- ブロック図	918
図 341.	IrDA データ変調 (3/16) - 通常モード	919



図 342.	DMA を使用した送信	920
図 343.	DMA を使用した受信	921
図 344.	2 つの USART 間のハードウェアフロー制御	921
図 345.	RS232 RTS フロー制御	922
図 346.	RS232 CTS フロー制御	923
図 347.	USART 割り込みマッピング図	926
図 348.	SPI ブロック図	952
図 349.	全二重シングルマスタ/シングルスレーブアプリケーション	953
図 350.	半二重シングルマスタ/シングルスレーブアプリケーション	954
図 351.	単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ)	955
図 352.	マスタと 3 つの独立したスレーブ	956
図 353.	マルチマスタアプリケーション	957
図 354.	ハードウェア/ソフトウェアスレーブ選択管理	958
図 355.	データクロックのタイミング図	959
図 356.	データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置	960
図 357.	FIFO での送受信のデータのパッキング	964
図 358.	マスタの全二重通信	967
図 359.	スレーブの全二重通信	968
図 360.	CRC のあるマスタの全二重通信	969
図 361.	パックされたモードでのマスタの全二重通信	970
図 362.	モトローラ SPI マスタモードでの NSSP パルス生成	973
図 363.	TI モードでの転送	974
図 364.	I <sup>2</sup> S ブロック図	977
図 365.	I <sup>2</sup> S 全二重のブロック図	978
図 366.	I <sup>2</sup> S フィリップスプロトコルの波形 (16/32 ビットフル精度)	980
図 367.	I <sup>2</sup> S フィリップス規格の波形 (24 ビットフレーム)	980
図 368.	0x8EAA33 の送信	981
図 369.	0x8EAA33 の受信	981
図 370.	I <sup>2</sup> S フィリップス規格 (32 ビットパケットフレームに拡張された 16 ビット)	981
図 371.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例	981
図 372.	MSB 詰め 16 ビットまたは 32 ビット長フル精度	982
図 373.	MSB 詰め 24 ビットフレーム長	982
図 374.	32 ビットパケットフレームに拡張された MSB 詰め 16 ビット	983
図 375.	LSB 詰め 16 ビットまたは 32 ビットフル精度	983
図 376.	LSB 詰め 24 ビットフレーム長	983
図 377.	0x3478AE を送信するために必要な動作	984
図 378.	0x3478AE の受信に必要な動作	984
図 379.	32 ビットパケットフレームに拡張された LSB 詰め 16 ビット	984
図 380.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例	985
図 381.	PCM 規格の波形 (16 ビット)	985
図 382.	PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)	986
図 383.	マスタモードでの開始シーケンス	987
図 384.	オーディオサンプリング周波数の定義	988
図 385.	I <sup>2</sup> S クロックジェネレータのアーキテクチャ	988
図 386.	CAN ネットワークのトポロジ	1010
図 387.	bxCAN 動作モード	1013
図 388.	bxCAN (サイレントモード)	1014
図 389.	bxCAN (ループバックモード)	1014
図 390.	bxCAN (組み合わせモード)	1015
図 391.	送信メールボックスの状態	1016
図 392.	受信 FIFO の状態	1017



図 393.	フィルタバンクのスケール設定 - レジスタ構造	1020
図 394.	フィルタ番号の例	1021
図 395.	フィルタのメカニズム - 例	1022
図 396.	CAN エラー状態図	1023
図 397.	ビットタイミング	1025
図 398.	CAN フレーム	1026
図 399.	イベントフラグと割り込み生成	1027
図 400.	CAN メールボックスレジスタ	1039
図 401.	USB ペリフェラルのブロック図	1055
図 402.	パケットバッファ領域とバッファ記述テーブルの位置の例	1060
図 403.	ブロック図 - STM32 MCU および Cortex-M4 <sup>®</sup> F レベルのデバッグサポート Cortex-M4 <sup>®</sup> F	1087
図 404.	SWJ デバッグポート	1089
図 405.	JTAG TAP 接続	1093
図 406.	TPIU ブロック図	1112

## 1 本マニュアルの概要

表 1. 各製品で利用可能な機能一覧

ペリフェラル	STM32F303xB/C	STM32F303xD/E	STM32F358xC	STM32F398xE	STM32F303x6/8	STM32F328x6/8
セクション 9 : リセットおよびクロック制御 (RCC)	利用可能	利用可能	利用可能	利用可能	利用可能	利用可能
セクション 11 : 汎用 I/O (GPIO)	最大 87	最大 115	最大 86	最大 114	最大 52	最大 51
セクション 13 : ダイレクトメモリアクセスコントローラ (DMA)	DMA1&2	DMA1&2	DMA1&2	DMA1&2	DMA1	DMA1
セクション 15 : アナログデジタルコンバータ (ADC)	ADC1,2,3&4	ADC1,2,3&4	ADC1,2,3&4	ADC1,2,3&4	ADC1&2	ADC1&2
セクション 16 : D/A コンバータ (DAC1 および DAC2)	DAC1 チャネル 1&2	DAC1 チャネル 1&2	DAC1 チャネル 1&2	DAC1 チャネル 1&2	DAC1 チャネル 1&2 + DAC2 チャネル 1	DAC1 チャネル 1&2 + DAC2 チャネル 1
セクション 17 : コンパレータ (COMP)	Comp1,2,3,4,5,6&7	Comp1,2,3,4,5,6&7	Comp1,2,3,4,5,6&7	Comp1,2,3,4,5,6&7	Comp2,4&6	Comp2,4&6
セクション 18 : オペアンプ (OPAMP)	Opamp 1,2,3&4	Opamp 1,2,3&4	Opamp 1,2,3&4	Opamp 1,2,3&4	Opamp 2	Opamp 2
セクション 19 : タッチセンシングコントローラ (TSC)	最大 24	最大 24	最大 24	最大 24	最大 18	最大 17
セクション 20 : 高機能制御タイマ (TIM1/TIM8/TIM20)	TIM1 & 8	TIM1, TIM8 & TIM20	TIM1 & 8	TIM1, TIM8 & TIM20	TIM1	TIM1
セクション 21 : 汎用タイマ (TIM2/TIM3/TIM4)	TIM2,3&4	TIM2,3&4	TIM2,3&4	TIM2,3&4	TIM2&3	TIM2&3
セクション 23 : 汎用タイマ (TIM15/TIM16/TIM17)	TIM15,16&17	TIM15,16&17	TIM15,16&17	TIM15,16&17	TIM15,16&17	TIM15,16&17
セクション 22 : 基本タイマ (TIM6/TIM7) :	TIM6&7	TIM6&7	TIM6&7	TIM6&7	TIM6&7	TIM6&7



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

表 1. 各製品で利用可能な機能一覧 (続き)

ペリフェラル	STM32F303xB/C	STM32F303xD/E	STM32F358xC	STM32F398xE	STM32F303x6/8	STM32F328x6/8
セクション 24 : 赤外線インタフェース (IRTIM)	利用可能	利用可能	利用可能	利用可能	利用可能	利用可能
セクション 25 : 独立型ウォッチドッグ (IWDG)	利用可能	利用可能	利用可能	利用可能	利用可能	利用可能
セクション 26 : システムウィンドウ型ウォッチドッグ (WWDG)	利用可能	利用可能	利用可能	利用可能	利用可能	利用可能
セクション 27 : リアルタイムクロック (RTC)	利用可能	利用可能	利用可能	利用可能	利用可能	利用可能
セクション 28 : I2C (Inter-integrated circuit) インタフェース	I2C1, I2C2	I2C1, I2C2, I2C3	I2C1, I2C2	I2C1, I2C2, I2C3	I2C1	I2C1
セクション 29 : USART (Universal synchronous asynchronous receiver transmitter)	最大 5 つの USART	最大 5 つの USART	最大 5 つの USART	最大 5 つの USART	最大 3 つの USART	最大 3 つの USART
セクション 30 : シリアルペリフェラルインタフェース/I2S (SPI/I2S)	I2S と SPI1, SPI2 & SPI3	I2S と SPI1, SPI2, SPI3, SPI4	I2S と SPI1, SPI2, SPI3	I2S と SPI1, SPI2, SPI3, SPI4	SPI1	SPI1
セクション 31 : コントローラエリアネットワーク (bxCAN)	利用可能	利用可能	利用可能	利用可能	利用可能	利用可能
セクション 32 : ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)	利用可能	利用可能	利用不可	利用不可	利用不可	利用不可

## 2 このマニュアルにおける表記の規則

### 2.1 レジスタに関する略記

レジスタの説明では、次の略記が使用されます。

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

### 2.2 用語

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

- Cortex-M4<sup>®</sup>F コアは、1つのデバッグポートを内蔵しています。**SWD デバッグポート (SWD-DP)**には、シリアルワイヤデバッグ (SWD) プロトコルに基づいた2ピン (クロックとデータ) のインタフェースを搭載しています。Cortex-M4<sup>®</sup>F Technical Reference Manual を参照してください。
- **ワード**: 32 ビット長のデータ。
- **ハーフワード**: 16 ビット長のデータ。
- **バイト**: 8 ビット長のデータ。
- **IAP (アプリケーション内プログラミング)**: IAP は、ユーザプログラム実行中にマイクロコントローラのフラッシュメモリを再プログラムする機能です。
- **ICP (インサーキットプログラミング)**: ICP は、ユーザアプリケーションボードにデバイスが搭載された状態で、JTAG プロトコルや SWD プロトコルまたはブートローダを用いて、マイクロコントローラのフラッシュメモリをプログラムする機能です。
- **オプションバイト**: フラッシュメモリに格納された製品設定ビット。
- **OBL**: オプションバイトローダ。
- **AHB**: アドバンスドハイパフォーマンスバス。

### 2.3 使用可能なペリフェラル

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



## 3 システムおよびメモリの概要

### 3.1 システムアーキテクチャ

STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE メインシステムの構成は次のとおりです。

- 5つのマスタ：
  - Cortex<sup>®</sup>-M4 コアの I-bus
  - Cortex<sup>®</sup>-M4 コアの D-bus
  - Cortex<sup>®</sup>-M4 コアの S-bus
  - GP-DMA1 および GP-DMA2 (汎用 DMA)
- 7つ (STM32F303xDxE および STM32F398xE で 8つ) のスレーブ：
  - DCode 上の内蔵フラッシュメモリ
  - ICode 上の内蔵フラッシュメモリ
  - 最大 40 KB の内蔵 SRAM
  - 8 KB の内蔵 CCM SRAM (STM32F303xE および STM32F398xE で 16 KB CCM SRAM)
  - STM32F303xDxE および STM32F398xE の FMC
  - AHB - APBx (APB1 または APB2)、すべての APB ペリフェラルに接続
  - GPIO ポート専用の AHB
  - ADC 1、2、3、および 4

STM32F303x6/8 および STM32F328x8 メインシステムの構成は次のとおりです。

- 4つのマスタ：
  - Cortex<sup>®</sup>-M4 コアの I-bus
  - Cortex<sup>®</sup>-M4 コアの D-bus
  - Cortex<sup>®</sup>-M4 コアの S-bus
  - GP-DMA1 (汎用 DMA)
- 7つのスレーブ：
  - DCode 上の内蔵フラッシュメモリ
  - ICode 上の内蔵フラッシュメモリ
  - 最大 12 KB の内蔵 SRAM
  - 4 KB の内蔵 CCM SRAM
  - AHB - APBx (APB1 または APB2)、すべての APB ペリフェラルに接続
  - GPIO ポート専用の AHB
  - ADC 1 および 2


 1 に示すように、マルチレイヤ AHB バスアーキテクチャを使用して相互接続します。

図 1. STM32F303xB/C および STM32F358xC システムアーキテクチャ

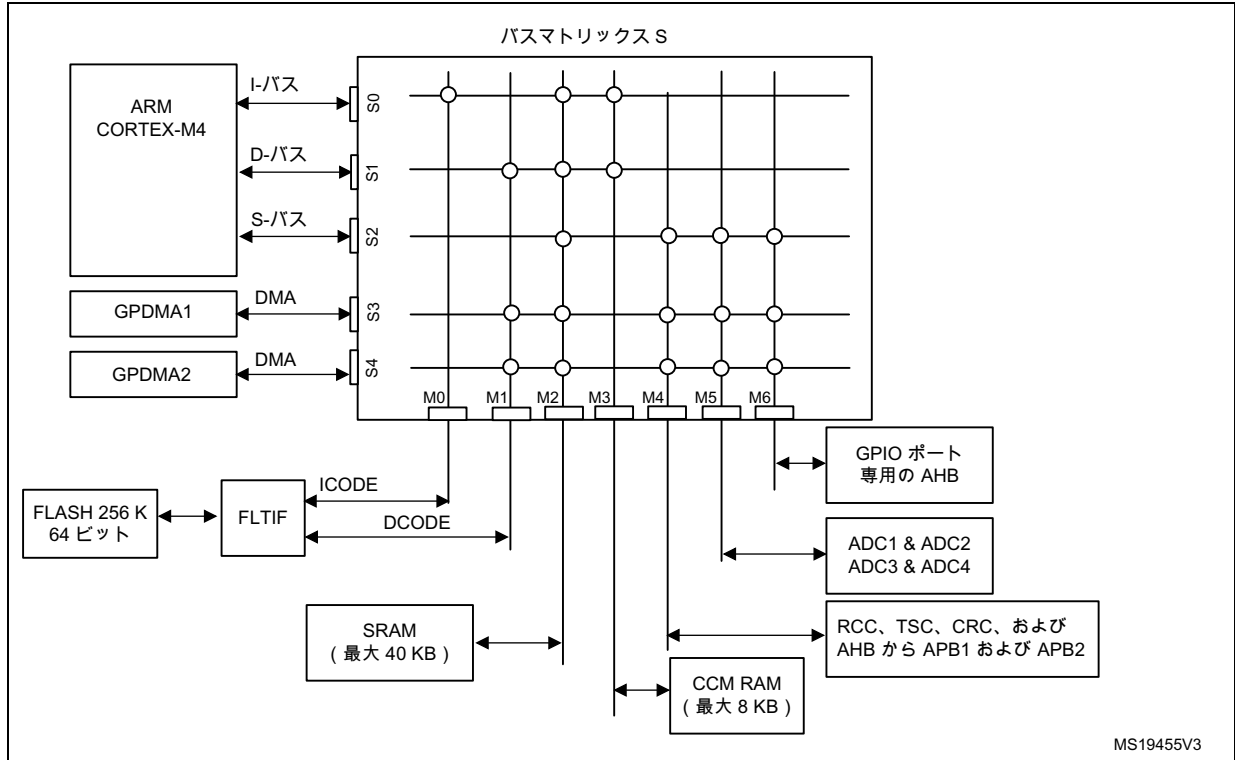


図 2. STM32F303x6/8 および STM32F328x8 システムアーキテクチャ

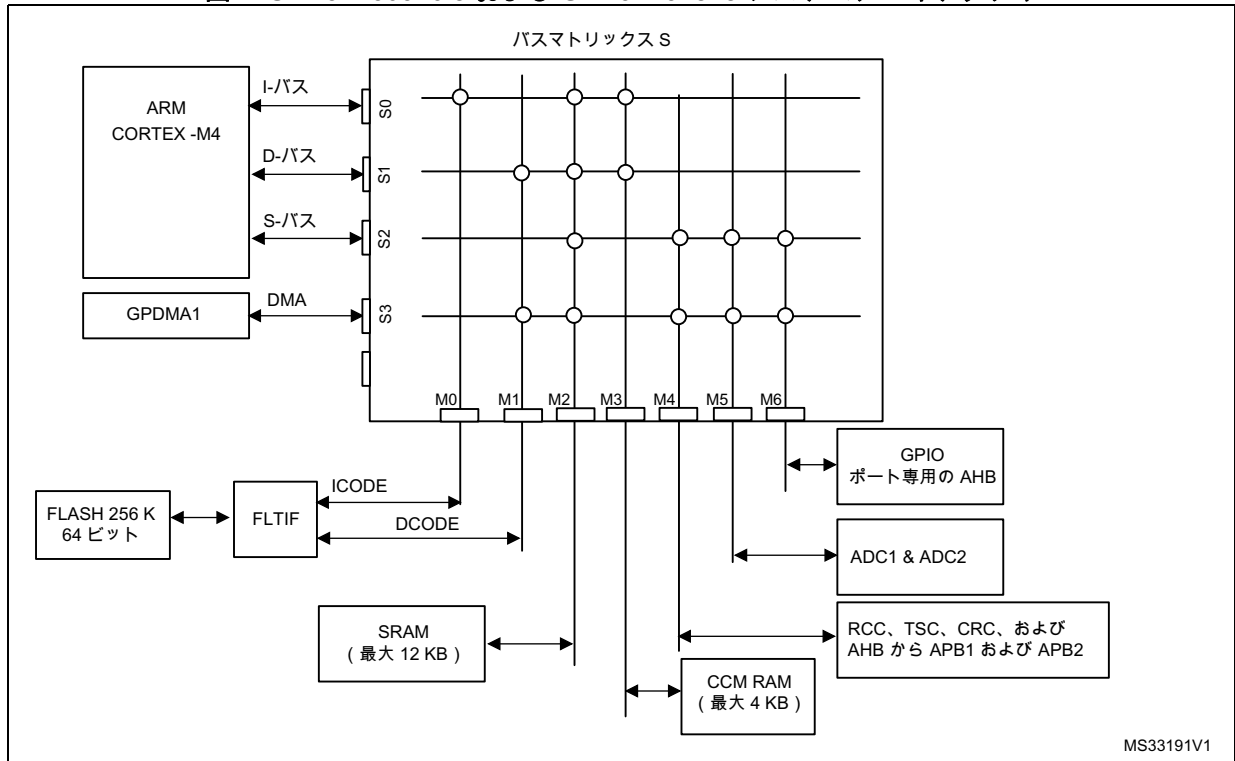
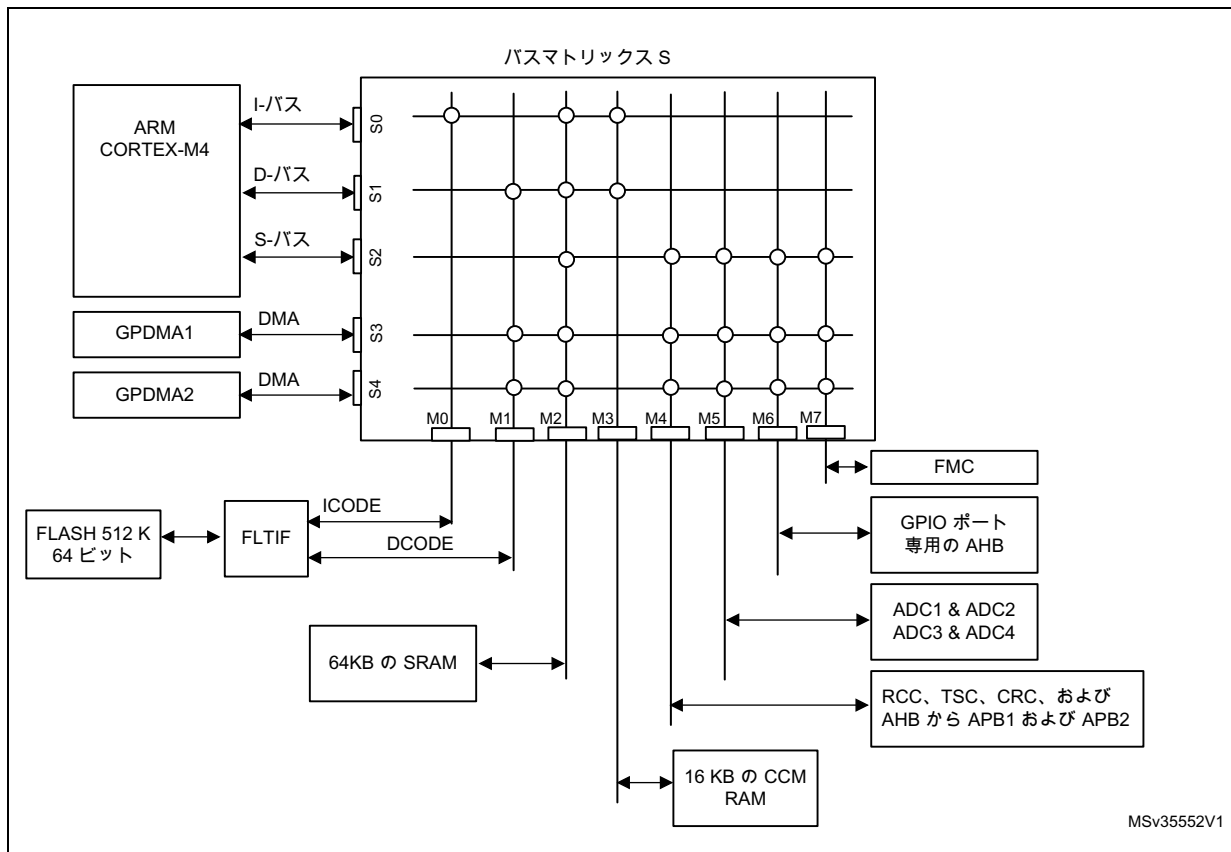


図 3. STM32F303xDxE および STM32F398xE のシステムアーキテクチャ



### 3.1.1 S0 : I-バス

このバスによって、Cortex<sup>®</sup>-M4 コアの命令バスが、バスマトリックスに接続されます。コアは、このバスを用いて命令をフェッチします。このバスの対象は、内蔵フラッシュメモリ、SRAM、および CCM SRAM です。

### 3.1.2 S1 : D-バス

このバスによって、Cortex<sup>®</sup>-M4 コアの DCode バス（リテラルロードとデバッグアクセス）が、バスマトリックスに接続されます。このバスの対象は、内蔵フラッシュメモリ、SRAM、および CCM SRAM です。

### 3.1.3 S2 : S-バス

このバスによって、Cortex<sup>®</sup>-M4 コアのシステムバスが、バスマトリックスに接続されます。このバスを使用してペリフェラルまたは SRAM 領域内のデータにアクセスします。このバスの対象は、SRAM、AHB - APB1/APB2 ブリッジ、AHB IO ポート、および ADC です。

### 3.1.4 S3、S4 : DMA バス

このバスは、異なるマスタのフラッシュ、SRAM、およびペリフェラルへのアクセスを管理するバスマトリックスに、DMA の AHB マスタインタフェースを接続します。



## 3.1.5 バスマトリックス

このバスマトリックスによって、マスタ間のアクセス調停を管理します。調停には、ラウンドロビン方式を使用します。バスマトリックスは 5 つのマスタ (CPU AHB、システムバス、DCode バス、ICode バス、DMA1&2 バス) と 7 つのスレーブ (FLITF、SRAM、CCM SRAM、AHB2GPIO、および AHB2APB1/2 ブリッジ、および ADC) で構成されています。

### AHB/APB ブリッジ

この 2 つの AHB/APB ブリッジでは、AHB および 2 つの APB バス間に完全同期接続を提供します。APB1 は 36 MHz に制限され、APB2 はフルスピードで動作します (72 MHz)。

このブリッジに接続されたペリフェラルのアドレスマッピングについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス \(50 ページ\)](#) を参照してください。

各デバイスのリセット後、すべてのペリフェラルクロックは無効になります (SRAM と FLITF は除く)。ペリフェラルを使用する前に、RCC\_AHBENR、RCC\_APB2ENR、または RCC\_APB1ENR レジスタでクロックを有効にする必要があります。

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

## 3.2 メモリ構成

### 3.2.1 概要

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

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

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

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

### 3.2.2 メモリマップとレジスタ境界アドレス

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

次の表に、デバイスで使用可能なペリフェラルの境界アドレスを示します。

表 2. STM32F303xB/C および STM32F358xC ペリフェラルレジスタ境界アドレス

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
AHB3	0x5000 0400 - 0x5000 07FF	1 K	ADC3 - ADC4	<a href="#">セクション 15.6.4 (406 ページ)</a>
	0x5000 0000 - 0x5000 03FF	1 K	ADC1 - ADC2	
	0x4800 1800 - 0x4FFF FFFF	~132 M	予約済み	
AHB2	0x4800 1400 - 0x4800 17FF	1 K	GPIOF	<a href="#">セクション 11.4.12 (241 ページ)</a>
	0x4800 1000 - 0x4800 13FF	1 K	GPIOE	
	0x4800 0C00 - 0x4800 0FFF	1 K	GPIOD	
	0x4800 0800 - 0x4800 0BFF	1 K	GPIOC	
	0x4800 0400 - 0x4800 07FF	1 K	GPIOB	
	0x4800 0000 - 0x4800 03FF	1 K	GPIOA	
	0x4002 4400 - 0x47FF FFFF	~128 M	予約済み	
AHB1	0x4002 4000 - 0x4002 43FF	1 K	TSC	<a href="#">セクション 19.6.11 (500 ページ)</a>
	0x4002 3400 - 0x4002 3FFF	3 K	予約済み	
	0x4002 3000 - 0x4002 33FF	1 K	CRC	<a href="#">セクション 6.4.6 (91 ページ)</a>
	0x4002 2400 - 0x4002 2FFF	3 K	予約済み	
	0x4002 2000 - 0x4002 23FF	1 K	フラッシュインタフェース	<a href="#">セクション 4.6 (81 ページ)</a>
	0x4002 1400 - 0x4002 1FFF	3 K	予約済み	
	0x4002 1000 - 0x4002 13FF	1 K	RCC	<a href="#">セクション 9.4.14 (165 ページ)</a>
	0x4002 0800 - 0x4002 0FFF	2 K	予約済み	
	0x4002 0400 - 0x4002 07FF	1 K	DMA2	<a href="#">セクション 13.5.7 (280 ページ)</a>
	0x4002 0000 - 0x4002 03FF	1 K	DMA1	

表 2. STM32F303xB/C および STM32F358xC ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
	0x4001 8000 - 0x4001 FFFF	32 K	予約済み	
APB2	0x4001 4C00 - 0x4001 7FFF	13 K	予約済み	
	0x4001 4800 - 0x4001 4BFF	1 K	TIM17	セクション 23.6.17 (753 ページ)
	0x4001 4400 - 0x4001 47FF	1 K	TIM16	
	0x4001 4000 - 0x4001 43FF	1 K	TIM15	セクション 23.5.18 (734 ページ)
	0x4001 3C00 - 0x4001 3FFF	1 K	予約済み	
	0x4001 3800 - 0x4001 3BFF	1 K	USART1	Section 25.7.12 on page 701
	0x4001 3400 - 0x4001 37FF	1 K	TIM8	セクション 20.4.25 (594 ページ)
	0x4001 3000 - 0x4001 33FF	1 K	SPI1	セクション 30.9.10 (1008 ページ)
	0x4001 2C00 - 0x4001 2FFF	1 K	TIM1	セクション 20.4.25 (594 ページ)
	0x4001 0800 - 0x4001 2BFF	9 K	予約済み	
	0x4001 0400 - 0x4001 07FF	1 K	EXTI	セクション 14.3.13 (301 ページ)
	0x4001 0000 - 0x4001 03FF	1 K	SYSCFG + COMP + OPAMP	セクション 12.1.10 (259 ページ)、 セクション 17.5.8 (460 ページ)、 セクション 18.4.5 (482 ページ)
	0x4000 7800 - 0x4000 FFFF	34 K	予約済み	
APB1	0x4000 7400 - 0x4000 77FF	1 K	DAC1	セクション 16.10.15 (434 ページ)
	0x4000 7000 - 0x4000 73FF	1 K	PWR	セクション 7.4.3 (108 ページ)
	0x4000 6C00 - 0x4000 6FFF	1 K	予約済み	
	0x4000 6800 - 0x4000 6BFF	1 K	予約済み	
	0x4000 6400 - 0x4000 67FF	1 K	bxCAN	セクション 31.9.5 (1050 ページ)
	0x4000 6000 - 0x4000 63FF	1 K	USB SRAM 512 バイト	セクション 32.6.3 (1085 ページ)
	0x4000 5C00 - 0x4000 5FFF	1 K	USB デバイス FS	
	0x4000 5800 - 0x4000 5BFF	1 K	I2C2	セクション 28.7.12 (881 ページ)
	0x4000 5400 - 0x4000 57FF	1 K	I2C1	
	0x4000 5000 - 0x4000 53FF	1 K	UART5	Section 25.7.12 on page 701
	0x4000 4C00 - 0x4000 4FFF	1 K	UART4	
	0x4000 4800 - 0x4000 4BFF	1 K	USART3	
	0x4000 4400 - 0x4000 47FF	1 K	USART2	
	0x4000 4000 - 0x4000 43FF	1 K	I2S3ext	セクション 30.9.10 (1008 ページ)
	0x4000 3C00 - 0x4000 3FFF	1 K	SPI3/I2S3	
	0x4000 3800 - 0x4000 3BFF	1 K	SPI2/I2S2	
	0x4000 3400 - 0x4000 37FF	1 K	I2S2ext	
	0x4000 3000 - 0x4000 33FF	1 K	IWDG	セクション 25.4.6 (764 ページ)
	0x4000 2C00 - 0x4000 2FFF	1 K	WWDG	セクション 26.4.4 (771 ページ)
	0x4000 2800 - 0x4000 2BFF	1 K	RTC	セクション 27.6.20 (812 ページ)
	0x4000 1800 - 0x4000 27FF	4 K	予約済み	

表 2. STM32F303xB/C および STM32F358xC ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
APB1	0x4000 1400 - 0x4000 17FF	1 K	TIM7	セクション 22.4.9 (678 ページ)
	0x4000 1000 - 0x4000 13FF	1 K	TIM6	
	0x4000 0C00 - 0x4000 0FFF	1 K	予約済み	
	0x4000 0800 - 0x4000 0BFF	1 K	TIM4	セクション 21.4.19 (664 ページ)
	0x4000 0400 - 0x4000 07FF	1 K	TIM3	
	0x4000 0000 - 0x4000 03FF	1 K	TIM2	
	0x2000 A000 - 3FFF FFFF	~512 M	予約済み	
	0x2000 0000 - 0x2000 9FFF	40 K	SRAM	-
	0x1FFF F800 - 0x1FFF FFFF	2 K	オプションバイト	-
	0x1FFF D800 - 0x1FFF F7FF	8 K	システムメモリ	-
	0x1000 2000 - 0x1FFF D7FF	~256 M	予約済み	
	0x1000 0000 - 0x1000 1FFF	8 K	CCM SRAM	-
	0x0804 0000 - 0x0FFF FFFF	~128 M	予約済み	
	0x0800 0000 - 0x0803 FFFF	256 K	メインフラッシュメモリ	-
	0x0004 0000 - 0x07FF FFFF	~128 M	予約済み	
	0x0000 000 - 0x0003 FFFF	256 K	メインフラッシュメモリ、システムメモリまたはSRAM (BOOT 設定に応じて)	-

表 3. STM32F303xD/E および STM32F398xE ペリフェラルレジスタ境界アドレス

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
AHB4	0xA000 0400 - 0xA000 0FFF	4 K	FMC 制御レジスタ	セクション 10.7 : FMC レジスタマップ
	0x8000 0400 - 0x9FFF FFFF	512 M	FMC バンク 3 および 4	
	0x6000 0000 - 0x7FFF FFFF	512 M	FMC バンク 1 および 2	
	0x5000 0800 - 0x5FFF FFFF	384M	予約済み	
AHB3	0x5000 0400 - 0x5000 07FF	1 K	ADC3 - ADC4	セクション 15.6.4 (406 ページ)
	0x5000 0000 - 0x5000 03FF	1 K	ADC1 - ADC2	
	0x4800 2000 - 0x4FFF FFFF	~132 M	予約済み	

表 3. STM32F303xD/E および STM32F398xE ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
AHB2	0x4800 1C00 - 0x4800 1FFF	1 K	GPIOH	セクション 11.4.12 (241 ページ)
	0x4800 1800 - 0x4800 1BFF	1 K	GPIOG	
	0x4800 1400 - 0x4800 17FF	1 K	GPIOF	
	0x4800 1000 - 0x4800 13FF	1 K	GPIOE	
	0x4800 0C00 - 0x4800 0FFF	1 K	GPIOD	
	0x4800 0800 - 0x4800 0BFF	1 K	GPIOC	
	0x4800 0400 - 0x4800 07FF	1 K	GPIOB	
	0x4800 0000 - 0x4800 03FF	1 K	GPIOA	
	0x4002 4400 - 0x47FF FFFF	~128 M	予約済み	
AHB1	0x4002 4000 - 0x4002 43FF	1 K	TSC	セクション 19.6.11 (500 ページ)
	0x4002 3400 - 0x4002 3FFF	3 K	予約済み	
	0x4002 3000 - 0x4002 33FF	1 K	CRC	セクション 6.4.6 (91 ページ)
	0x4002 2400 - 0x4002 2FFF	3 K	予約済み	
	0x4002 2000 - 0x4002 23FF	1 K	フラッシュインタフェース	セクション 4.6 (81 ページ)
	0x4002 1400 - 0x4002 1FFF	3 K	予約済み	
	0x4002 1000 - 0x4002 13FF	1 K	RCC	セクション 9.4.14 (165 ページ)
	0x4002 0800 - 0x4002 0FFF	2 K	予約済み	
	0x4002 0400 - 0x4002 07FF	1 K	DMA2	セクション 13.5.7 (280 ページ)
	0x4002 0000 - 0x4002 03FF	1 K	DMA1	
	0x4001 8000 - 0x4001 FFFF	32 K	予約済み	
APB2	0x4001 4C00 - 0x4001 4FFF	1 K	予約済み	
	0x4001 5400 - 0x4001 7FFF	11K	予約済み	
	0x4001 5000 - 0x4001 53FF	1 K	TIM20	セクション 23.6.17 (753 ページ)
	0x4001 4800 - 0x4001 4BFF	1 K	TIM17	
	0x4001 4400 - 0x4001 47FF	1 K	TIM16	
	0x4001 4000 - 0x4001 43FF	1 K	TIM15	セクション 23.5.18 (734 ページ)
	0x4001 3C00 - 0x4001 3FFF	1 K	SPI4	セクション 30.9.10 (1008 ページ)
	0x4001 3800 - 0x4001 3BFF	1 K	USART1	Section 25.7.12 on page 701
	0x4001 3400 - 0x4001 37FF	1 K	TIM8	セクション 20.4.25 (594 ページ)
	0x4001 3000 - 0x4001 33FF	1 K	SPI1	セクション 30.9.10 (1008 ページ)
	0x4001 2C00 - 0x4001 2FFF	1 K	TIM1	セクション 20.4.25 (594 ページ)
	0x4001 0800 - 0x4001 2BFF	9 K	予約済み	
	0x4001 0400 - 0x4001 07FF	1 K	EXTI	セクション 14.3.13 (301 ページ)
	0x4001 0000 - 0x4001 03FF	1 K	SYSCFG + COMP + OPAMP	セクション 12.1.10 (259 ページ)、セクション 17.5.8 (460 ページ)、セクション 18.4.5 (482 ページ)



表 3. STM32F303xD/E および STM32F398xE ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
	0x4000 7C00 - 0x4000 FFFF	33 K	予約済み	
APB1	0x4000 7800 - 0x4000 7BFF	1 K	I2C3	<a href="#">セクション 28.7.12 (881 ページ)</a>
	0x4000 7400 - 0x4000 77FF	1 K	DAC1	<a href="#">セクション 16.10.15 (434 ページ)</a>
	0x4000 7000 - 0x4000 73FF	1 K	PWR	<a href="#">セクション 7.4.3 (108 ページ)</a>
	0x4000 6C00 - 0x4000 6FFF	1 K	予約済み	
	0x4000 6800 - 0x4000 6BFF	1 K	予約済み	
	0x4000 6400 - 0x4000 67FF	1 K	bxCAN	<a href="#">セクション 31.9.5 (1050 ページ)</a>
	0x4000 6000 - 0x4000 63FF	1 K	USB/CAN SRAM	<a href="#">セクション 32.6.3 (1085 ページ)</a>
	0x4000 5C00 - 0x4000 5FFF	1 K	USB デバイス FS	
	0x4000 5800 - 0x4000 5BFF	1 K	I2C2	<a href="#">セクション 28.7.12 (881 ページ)</a>
	0x4000 5400 - 0x4000 57FF	1 K	I2C1	
	0x4000 5000 - 0x4000 53FF	1 K	UART5	<a href="#">Section 25.7.12 on page 701</a>
	0x4000 4C00 - 0x4000 4FFF	1 K	UART4	
	0x4000 4800 - 0x4000 4BFF	1 K	USART3	
	0x4000 4400 - 0x4000 47FF	1 K	USART2	
	0x4000 4000 - 0x4000 43FF	1 K	I2S3ext	<a href="#">セクション 30.9.10 (1008 ページ)</a>
	0x4000 3C00 - 0x4000 3FFF	1 K	SPI3/I2S3	
	0x4000 3800 - 0x4000 3BFF	1 K	SPI2/I2S2	
	0x4000 3400 - 0x4000 37FF	1 K	I2S2ext	
	0x4000 3000 - 0x4000 33FF	1 K	IWDG	<a href="#">セクション 25.4.6 (764 ページ)</a>
	0x4000 2C00 - 0x4000 2FFF	1 K	WWDG	<a href="#">セクション 26.4.4 (771 ページ)</a>
0x4000 2800 - 0x4000 2BFF	1 K	RTC	<a href="#">セクション 27.6.20 (812 ページ)</a>	
0x4000 1800 - 0x4000 27FF	4 K	予約済み		
APB1	0x4000 1400 - 0x4000 17FF	1 K	TIM7	<a href="#">セクション 22.4.9 (678 ページ)</a>
	0x4000 1000 - 0x4000 13FF	1 K	TIM6	
	0x4000 0C00 - 0x4000 0FFF	1 K	予約済み	
	0x4000 0800 - 0x4000 0BFF	1 K	TIM4	<a href="#">セクション 21.4.19 (664 ページ)</a>
	0x4000 0400 - 0x4000 07FF	1 K	TIM3	
	0x4000 0000 - 0x4000 03FF	1 K	TIM2	
0x2000 A000 - 3FFF FFFF	~512 M	予約済み		
0x2000 0000 - 0x2000 FFFF	64 K	SRAM	-	
0x1FFF F800 - 0x1FFF FFFF	2 K	オプションバイト	-	
0x1FFF D800 - 0x1FFF F7FF	8 K	システムメモリ	-	
0x1000 2000 - 0x1FFF D7FF	~256 M	予約済み		
0x1000 0000 - 0x1000 3FFF	16 K	CCM SRAM	-	
0x0808 0000 - 0x0FFF FFFF	~128 M	予約済み		

表 3. STM32F303xD/E および STM32F398xE ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
	0x0800 0000 - 0x0807 FFFF	512 K	メインフラッシュメモリ	-
	0x0008 0000 - 0x07FF FFFF	~128 M	予約済み	
	0x0000 000 - 0x0007 FFFF	512 K	メインフラッシュメモリ、システムメモリまたはSRAM (BOOT 設定に応じて)	-

表 4. STM32F303x6/8 および STM32F328x8 ペリフェラルレジスタ境界アドレス

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
AHB3	0x5000 0400 - 0x5000 07FF	1 K	予約済み	
	0x5000 0000 - 0x5000 03FF	1 K	ADC1 - ADC2	<a href="#">セクション 15.6.4 (406 ページ)</a>
	0x4800 1800 - 0x4FFF FFFF	~132 M	予約済み	
AHB2	0x4800 1400 - 0x4800 17FF	1 K	GPIOF	<a href="#">セクション 11.4.12 (241 ページ)</a>
	0x4800 1000 - 0x4800 13FF	1 K	予約済み	
	0x4800 0C00 - 0x4800 0FFF	1 K	GPIOD	
	0x4800 0800 - 0x4800 0BFF	1 K	GPIOC	
	0x4800 0400 - 0x4800 07FF	1 K	GPIOB	
	0x4800 0000 - 0x4800 03FF	1 K	GPIOA	
	0x4002 4400 - 0x47FF FFFF	~128 M	予約済み	
AHB1	0x4002 4000 - 0x4002 43FF	1 K	TSC	<a href="#">セクション 19.6.11 (500 ページ)</a>
	0x4002 3400 - 0x4002 3FFF	3 K	予約済み	
	0x4002 3000 - 0x4002 33FF	1 K	CRC	<a href="#">セクション 6.4.6 (91 ページ)</a>
	0x4002 2400 - 0x4002 2FFF	3 K	予約済み	
	0x4002 2000 - 0x4002 23FF	1 K	フラッシュインタフェース	<a href="#">セクション 4.6 (81 ページ)</a>
	0x4002 1400 - 0x4002 1FFF	3 K	予約済み	
	0x4002 1000 - 0x4002 13FF	1 K	RCC	<a href="#">セクション 9.4.14 (165 ページ)</a>
	0x4002 0400 - 0x4002 0FFF	3 K	予約済み	
	0x4002 0000 - 0x4002 03FF	1 K	DMA1	<a href="#">セクション 13.5.7 (280 ページ)</a>
	0x4001 8000 - 0x4001 FFFF	32 K	予約済み	

表 4. STM32F303x6/8 および STM32F328x8 ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
APB2	0x4001 4C00 - 0x4001 7FFF	13 K	予約済み	
	0x4001 4800 - 0x4001 4BFF	1 K	TIM17	セクション 23.6.17 (753 ページ)
	0x4001 4400 - 0x4001 47FF	1 K	TIM16	
	0x4001 4000 - 0x4001 43FF	1 K	TIM15	セクション 23.5.18 (734 ページ)
	0x4001 3C00 - 0x4001 3FFF	1 K	予約済み	
	0x4001 3800 - 0x4001 3BFF	1 K	USART1	Section 25.7.12 on page 701
	0x4001 3400 - 0x4001 37FF	1 K	予約済み	
	0x4001 3000 - 0x4001 33FF	1 K	SPI1	セクション 30.9.10 (1008 ページ)
	0x4001 2C00 - 0x4001 2FFF	1 K	TIM1	セクション 20.4.25 (594 ページ)
	0x4001 0800 - 0x4001 2BFF	9 K	予約済み	
	0x4001 0400 - 0x4001 07FF	1 K	EXTI	セクション 14.3.13 (301 ページ)
	0x4001 0000 - 0x4001 03FF	1 K	SYSCFG + COMP + OPAMP	セクション 12.1.10 (259 ページ)、 セクション 17.5.8 (460 ページ)、 セクション 18.4.5 (482 ページ)
	0x4000 9C00 - 0x4000 FFFF	25 K	予約済み	
APB1	0x4000 9800 - 0x4000 9BFF	1 K	DAC2	セクション 16.10.15 (434 ページ)
	0x4000 7800 - 0x4000 97FF	8 K	予約済み	
	0x4000 7400 - 0x4000 77FF	1 K	DAC1	セクション 16.10.15 (434 ページ)
	0x4000 7000 - 0x4000 73FF	1 K	PWR	セクション 7.4.3 (108 ページ)
	0x4000 6C00 - 0x4000 6FFF	1 K	予約済み	
	0x4000 6800 - 0x4000 6BFF	1 K	予約済み	
	0x4000 6400 - 0x4000 67FF	1 K	bxCAN	セクション 31.9.5 (1050 ページ)
	0x4000 5800 - 0x4000 63FF	3 K	予約済み	
	0x4000 5400 - 0x4000 57FF	1 K	I2C1	セクション 28.7.12 (881 ページ)
	0x4000 4C00 - 0x4000 53FF	2 K	予約済み	
	0x4000 4800 - 0x4000 4BFF	1 K	USART3	Section 25.7.12 on page 701
	0x4000 4400 - 0x4000 47FF	1 K	USART2	
	0x4000 3400 - 0x4000 43FF	4 K	予約済み	
	0x4000 3000 - 0x4000 33FF	1 K	IWDG	セクション 25.4.6 (764 ページ)
	0x4000 2C00 - 0x4000 2FFF	1 K	WWDG	セクション 26.4.4 (771 ページ)
	0x4000 2800 - 0x4000 2BFF	1 K	RTC	セクション 27.6.20 (812 ページ)
	0x4000 1800 - 0x4000 27FF	4 K	予約済み	
	0x4000 1400 - 0x4000 17FF	1 K	TIM7	セクション 22.4.9 (678 ページ)
	0x4000 1000 - 0x4000 13FF	1 K	TIM6	
	0x4000 0800 - 0x4000 0FFF	2K	予約済み	
0x4000 0400 - 0x4000 07FF	1 K	TIM3	セクション 21.4.19 (664 ページ)	
0x4000 0000 - 0x4000 03FF	1 K	TIM2	-	



表 4. STM32F303x6/8 および STM32F328x8 ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ (バイト)	ペリフェラル	ペリフェラルレジスタマップ
	0x2000 3000 - 3FFF FFFF	~512 M	予約済み	
	0x2000 0000 - 0x2000 2FFF	12 K	SRAM	-
	0x1FFF F800 - 0x1FFF FFFF	2 K	オプションバイト	-
	0x1FFF D800 - 0x1FFF F7FF	8 K	システムメモリ	-
	0x1000 1000 - 0x1FFF D7FF	~256 M	予約済み	
	0x1000 0000 - 0x1000 0FFF	4 K	CCM SRAM	-
	0x0804 0000 - 0x0FFF FFFF	~128 M	予約済み	
	0x0800 0000 - 0x0800 FFFF	64 K	メインフラッシュメモリ	-
	0x0001 0000 - 0x07FF FFFF	~128 M	予約済み	
	0x0000 000 - 0x0000 FFFF	64 K	メインフラッシュメモリ、システムメモリまたは SRAM (BOOT 設定に応じて)	-

### 3.3 内蔵 SRAM

STM32F303xB/C および STM32F358xC デバイスには、最大 48 KB の SRAM を搭載しています。この SRAM には、バイト、ハーフワード (16 ビット)、またはフルワード (32 ビット) によるアクセスが可能です。

- 最大 40 KB の SRAM には、最大システムクロック周波数 (ウェイトステートなし) でアドレス指定、つまり CPU と DMA の両方でのアクセスが可能です。
- 8 KB の CCM SRAM。これは、重要なルーチンを実行するため、またはデータをアクセスするために使用されます。また、CPU のみが使用できます。DMA アクセスは許可されていません。このメモリは、最大システムクロック周波数 (ウェイトステートなし) でアドレス指定できます。

STM32F303xD/E および STM32F398xE デバイスには、最大 80 KB の SRAM を搭載しています。この SRAM には、バイト、ハーフワード (16 ビット)、またはフルワード (32 ビット) によるアクセスが可能です。

- 最大 64 KB の SRAM には、最大システムクロック周波数 (ウェイトステートなし) でアドレス指定、つまり CPU と DMA の両方でのアクセスが可能です。
- 16 KB の CCM SRAM。これは、重要なルーチンを実行するため、またはデータをアクセスするために使用されます。また、CPU のみが使用できます。DMA アクセスは許可されていません。このメモリは、最大システムクロック周波数 (ウェイトステートなし) でアドレス指定できます。

STM32F303x6/8 および STM32F328x8 デバイスは、同じメモリを搭載しています。ただし、最大 16 KB の SRAM、最大 12 KB の SRAM、および 4 KB の CCM SRAM までのみです。

## 3.3.1 パリティチェック

STM32F303xB/C および STM32F358xC デバイスで、40 KB SRAM のパリティチェックは最初の 16 バイトに対してのみ実装されます。

SRAM パリティチェックはデフォルトで無効です。これは、ユーザが必要なときにオプションビットを使用して有効化します。

STM32F303x6/8 および STM32F328x8 デバイスでは、パリティチェックはすべての SRAM と CCM SRAM に対して実装されます。

STM32F303xD/E および STM32F398xE デバイスでは、パリティチェックは SRAM の最初の 32 KB と CCM SRAM 全体に対して実装されます。

パリティチェックに使用できるのは 4 ビットであるため (1 バイトあたり 1 ビット)、パリティチェックをサポートする SRAM のデータバス幅は 36 ビットです。これは、Class B や SIL 標準などで必要とされるメモリの信頼性を高めるためです。

パリティビットは、SRAM に書き込む際にデータおよびアドレス上で計算され、格納されます。その後、読み出すときに自動的にチェックされます。1 つのビットがフェイルすると、SRAM パリティチェックが有効化されている場合は、NMI が生成されます。同じエラーが、SYSCFG 設定レジスタ 2 (SYSCFG\_CFGR2) の SRAM\_PARITY\_LOCK に制御ビットを設定することにより、TIMER 20、1、8、15、16、および 17 のブレイク入力にリンクされている場合もあります。パリティエラーの場合、SRAM パリティエラーフラグ (SRAM\_PEF) は SYSCFG 設定レジスタ 2 (SYSCFG\_CFGR2) に設定されます。詳細については、SYSCFG 設定レジスタ 2 (SYSCFG\_CFGR2) を参照してください。

SYSCFG\_CFGR2 レジスタの BYP\_ADD\_PAR ビットは、ユーザがアドレス 0x2XXXXXXX (アドレス範囲 0x20000000-0x20002000 のアドレス) で RAM のコードをプログラミングし、ブート時に RAM からコードを実行する際に (RAM はアドレス 0x00 で再割り当てされる)、不要なパリティエラーの発生を防ぐために使用できます。

## 3.3.2 CCM SRAM 書き込み保護

CCM SRAM は、1 KB のページ単位で書き込み保護されています。

表 5. CCM SRAM の構成

ページ番号	開始アドレス	終了アドレス
ページ 0	0x1000 0000	0x1000 03FF
ページ 1	0x1000 0400	0x1000 07FF
ページ 2	0x1000 0800	0x1000 0BFF
ページ 3	0x1000 0C00	0x1000 0FFF
ページ 4 <sup>(1)</sup>	0x1000 1000	0x1000 13FF
ページ 5 <sup>(1)</sup>	0x1000 1400	0x1000 17FF
ページ 6 <sup>(1)</sup>	0x1000 1800	0x1000 1BFF
ページ 7 <sup>(1)</sup>	0x1000 1C00	0x1000 1FFF
ページ 8 <sup>(2)</sup>	0x1000 2000	0x1000 23FF
ページ 9 <sup>(2)</sup>	0x1000 2400	0x1000 27FF
ページ 10 <sup>(2)</sup>	0x1000 2800	0x1000 2BFF
ページ 11 <sup>(2)</sup>	0x1000 2C00	0x1000 2FFF
ページ 12 <sup>(2)</sup>	0x1000 3000	0x1000 33FF
ページ 13 <sup>(2)</sup>	0x1000 3400	0x1000 37FF
ページ 14 <sup>(2)</sup>	0x1000 3800	0x1000 3BFF
ページ 15 <sup>(2)</sup>	0x1000 3C00	0x1000 3FFF

1. STM32F303xB/C/D/E および STM32F358xC デバイスのみ

2. STM32F303xD/E および STM32F398xE デバイスのみ

書き込み保護は、SYSCFG ブロックの CCM SRAM 保護レジスタ (SYSCFG\_RCR) で有効にすることができます。これは、「1」を 1 回だけ書き込むメカニズムを持つレジスタで、ビットに「1」を書き込むことで、システムリセットでのみ削除/消去可能な書き込み保護を SRAM の当該ページに設定します。詳細については、SYSCFG のセクションを参照してください。

## 3.4 フラッシュメモリの概要

フラッシュメモリは2つの別個の物理領域で構成されます。

- メインフラッシュメモリブロック。必要に応じてアプリケーションプログラムとユーザデータを含みます。
- 情報ブロック。2つの部分で構成されます。
  - ハードウェアとメモリ保護のユーザ設定のオプションバイト。
  - 独自のブートローダコードを含むシステムメモリ。詳細については、[セクション 4：内蔵フラッシュメモリ](#)を参照してください。

フラッシュメモリの命令およびデータアクセスは、AHB バスを通して行います。プリフェッチブロックは、ICode バスを通して命令フェッチに使用されます。調停はフラッシュメモリインタフェースで行われ、DCode バス上のデータアクセスに優先順位が与えられます。また、フラッシュメモリ操作（プログラム/消去）を実行するために必要な、フラッシュレジスタを使って制御されるロジックも実装します。

## 3.5 ブート設定

STM32F3xx では、次の表に示すように、ユーザオプションバイトで BOOT0 ピンおよび nBOOT1 ビットによって3種類のブートモードを選択できます。

表 6. ブートモード

ブートモード選択		ブートモード	エイリアシング
nBOOT1	BOOT0		
x	0	メインフラッシュメモリ	メインフラッシュメモリがブート領域として選択されます。
1	1	システムメモリ	システムメモリがブート領域として選択されます。
0	1	内蔵 SRAM	(DCode バス上の) 内蔵 SRAM がブート領域として選択されます。

BOOT0 ピンと nBOOT1 ビットの両方の値は、リセット後、SYSCLK の4番目の立上りエッジでラッチされます。

ユーザは、nBOOT1 と BOOT0 をセットして必要なブートモードを指定します。BOOT0 ピンおよび nBOOT1 ビットも、STANDBY モードの終了時に再サンプリングされます。したがって、これらのピンは STANDBY モードのときでも必要なブートモード設定に保たれる必要があります。この起動遅延時間が終了すると、CPU はアドレス 0x0000 0000 からスタック最上位の値をフェッチし、0x0000 0004 のブートメモリからコード実行を開始します。選択したブートメモリに応じて、メインフラッシュメモリ、システムメモリ、または SRAM に、次のようにアクセスできます。

- メインフラッシュメモリからブート：メインフラッシュメモリのエイリアスがブートメモリ空間 (0x0000 0000) に作成されますが、引き続き元のメモリ空間 (0x0800 0000) からアクセスすることも可能です。言い換えると、フラッシュメモリの内容は、0x0000 0000 または 0x0800 0000 から始まるアドレスからアクセスできます。
- システムメモリからブート：システムメモリのエイリアスがブートメモリ空間 (0x0000 0000) に作成されますが、引き続き元のメモリ空間 (0x1FFF D800) からアクセスすることも可能です。
- 内蔵 SRAM からブート：SRAM のエイリアスがブートメモリ空間 (0x0000 0000) に作成されますが、引き続き元のメモリ空間 (0x2000 0000) からアクセスすることも可能です。

## 3.5.1 内蔵ブートローダ

内蔵ブートローダは、システムメモリに配置されるよう生産時に ST によってプログラムされています。次のいずれかを使用して、フラッシュメモリを再プログラムするために使用します。

- STM32F303xB/C デバイスの USART1 (PA9/PA10)、USART2 (PD5/PD6)、または USB (DFU)
- STM32F358xC デバイスの USART1 (PA9/PA10)、USART2 (PD5/PD6)、I2C1 (PB6/PB7)
- STM32F303x6/8 および STM32F328x8 デバイスの USART1 (PA9/PA10)、USART2 (PA2/PA3)、I2C1 (PB6/PB7)
- STM32F303xD/E デバイスの USART1 (PA9/PA10)、USART2 (PA2/PA3)、または USB (DFU)
- STM32F398xE の USART1 (PA9/PA10) または USART2 (PA2/PA3) または I2C1 (PB6/PB7) または I2C3 (PA8/PB5)

**注：** *詳細については、対応するデータシートを参照してください。*

## 4 内蔵フラッシュメモリ

### 4.1 フラッシュの主な機能

STM32F303xD/E で最大 512 KB のフラッシュメモリ、STM32F303xB/C および STM32F358xC デバイスで最大 256 KB のフラッシュメモリ、STM32F303x6/8 および STM32F328x8 デバイスで最大 64 KB のフラッシュメモリ。

- メモリ構成：
  - メインメモリブロック：  
STM32F303xD/E で 128 KB × 64 ビット、STM32F303xB/C および STM32F358xC デバイスで 64 KB × 64 ビット。  
STM32F303x6/8 および STM32F328x8 16 KB × 64 ビット デバイス。
  - 情報ブロック：  
1280 × 64 ビット

フラッシュメモリインタフェース (FLITF) の機能：

- プリフェッチバッファ機能付き読み出しインタフェース (2 × 64 ビットワード)
- オプションバイトローダ
- フラッシュのプログラム/消去操作
- 読み出し/書き込み保護
- 低電力モード

### 4.2 フラッシュメモリの機能詳細

#### 4.2.1 フラッシュメモリの構成

フラッシュメモリは、64 ビット幅のメモリセルとして構成されており、コードとデータの両方の定数を格納するために使用できます。

メモリ構成は、STM32F303xB/C および STM32F358xC デバイスでは 2 KB のページを 128 ページ、STM32F303xD/E では 2 KB のページを 256 ページ含むメインメモリブロックと、表 7 に示す情報ブロックに基づいています。STM32F303x6/8 および STM32F328x8 デバイスでは、メインメモリブロックに 2 KB のページが 32 ページが含まれます。

表 7. フラッシュモジュールの構成<sup>(1)</sup>

フラッシュ領域	フラッシュメモリアドレス	サイズ (バイト)	名前
メインメモリ	0x0800 0000 - 0x0800 07FF	2 K	ページ 0
	0x0800 0800 - 0x0800 0FFF	2 K	ページ 1
	0x0800 1000 - 0x0800 17FF	2 K	ページ 2
	0x0800 1800 - 0x0800 1FFF	2 K	ページ 3
	⋮	⋮	⋮
	⋮	⋮	⋮
	⋮	⋮	⋮
	⋮	⋮	⋮
	⋮	⋮	⋮
	0x0807 F800 - 0x0807 FFFF	2 K	ページ 255
情報ブロック	0x1FFF D800 - 0x1FFF F7FF	8 K	システムメモリ
	0x1FFF F800 - 0x1FFF F80F	16	オプションバイト
フラッシュメモリ インタフェース レジスタ	0x4002 2000 - 0x4002 2003	4	FLASH_ACR
	0x4002 2004 - 0x4002 2007	4	FLASH_KEYR
	0x4002 2008 - 0x4002 200B	4	FLASH_OPTKEYR
	0x4002 200C - 0x4002 200F	4	FLASH_SR
	0x4002 2010 - 0x4002 2013	4	FLASH_CR
	0x4002 2014 - 0x4002 2017	4	FLASH_AR
	0x4002 2018 - 0x4002 201B	4	予約済み
	0x4002 201C - 0x4002 201F	4	FLASH_OBR
	0x4002 2020 - 0x4002 2023	4	FLASH_WRPR

1. 予約済みのフラッシュプログラムメモリには、灰色が使われます。

情報ブロックは、2つの部分に分かれます。

- システムメモリは、システムメモリブートモードでデバイスをブートするために使用します。この領域は STMicroelectronics によって予約済みで、次のいずれかのインタフェースを介したフラッシュメモリの再プログラムに使用するブートローダを含みます。内蔵レギュレータがオンになっているデバイスの USART1、USART2、および内蔵レギュレータがオフになっているデバイスの USART または I2C。これはデバイスの製造時に ST によってプログラムされており、誤った書き込み/消去操作から保護されています。詳細については、[www.st.com](http://www.st.com) から入手可能な AN2606 を参照してください。
- オプションバイト



## 4.2.2 読み出し操作

内蔵フラッシュモジュールは、共通メモリ空間として直接アドレス指定できます。任意のデータ読み出し操作が専用の読み出し機構を通じてフラッシュモジュールの内容にアクセスし、リクエストされたデータを提供します。

読み出しインタフェースは、フラッシュメモリにアクセスするための読み出しコントローラと、CPU へのインタフェースとして使用する AHB インタフェースで構成されています。読み出しインタフェースの主なタスクは、フラッシュメモリから読み出すための制御信号を生成することと、CPU で必要とされるブロックをプリフェッチすることです。プリフェッチブロックは、ICode バスによる命令フェッチのみに使用されます。リテラルプールへのアクセスには DCode バスが使用されます。これらの 2 つのバスのターゲットは同じフラッシュメモリであるため、プリフェッチアクセスよりも DCode バスアクセスのほうが優先されます。

読み出しアクセスは、フラッシュアクセス制御レジスタ (FLASH\_ACR) を通じて管理される次のオプションで行うことができます。

- 命令フェッチ：高速 CPU 実行用のプリフェッチバッファ
- 遅延：正確な読み出し操作に必要なウェイトステート数 (0 から 2 の間)



## 命令フェッチ

Cortex®-M4 は ICode バスを通じて命令をフェッチし、DCode バスを通じてリテラルプール（コンスタント/データ）をフェッチします。プリフェッチブロックは、ICode バスアクセスの効率の向上を目的としています。

## プリフェッチバッファ

プリフェッチバッファの幅はブロック 2 つ分で、各ブロックは 8 バイトで構成されます。プリフェッチブロックは直接マッピングされています。ブロックは、そのサイズがフラッシュメモリの帯域幅と一致しているため、シングル読み出しでフラッシュメモリと完全に置き換えることができます。

このプリフェッチバッファの実装により、CPU の高速実行が可能になります。これは、CPU が一度に 1 ワードずつフェッチしますが、プリフェッチバッファに次のワードがすぐに使用可能な状態で、準備できているためです。これは、コードがジャンプのために 64 ビットの境界に揃っている場合、加速の割合が 2 倍になることを意味しています。

## プリフェッチコントローラ

プリフェッチコントローラは、プリフェッチバッファで使用可能な空間に応じてフラッシュメモリにアクセスするかどうかを決定します。コントローラは、プリフェッチバッファで空いているブロックが少なくとも 1 つある場合に、読み出しリクエストを開始します。

リセット後のプリフェッチバッファの状態はオンです。プリフェッチバッファのオン/オフを切り替える際に、AHB クロックのプリスケアラを使用しないでください（SYSCLK は HCLK と等しくなければなりません）。マイクロコントローラが内部 8 MHz の RC (HSI) オシレータで実行している間、プリフェッチバッファは初期化ルーチン中に通常オン/オフに切り替えられます。

**注：** *AHB クロックで 1 以外のプリスケアラを使用している場合は、プリフェッチバッファをオンにしておく必要があります (FLASH\_ACR[4]='1')。*

*システム内に使用可能な高周波クロックが存在しない場合は、HCLK (AHB クロック) のハーフサイクルでフラッシュメモリにアクセスすることができます。このモードは、フラッシュアクセス制御レジスタの制御ビットをセットすることによって選択できます。*

*AHB クロックで 1 以外のプリスケアラを使用している場合、ハーフサイクルアクセスは使用できません。*

## アクセスの遅延

フラッシュメモリを読み出すための制御信号を維持するには、フラッシュアクセス制御レジスタの LATENCY[2:0] ビットで、プリフェッチコントローラのクロック周期に対するフラッシュメモリのアクセス時間の比率をプログラムする必要があります。この値によってフラッシュメモリの制御信号の維持に必要なサイクル数を得ることができ、また要求されたデータを正しく読み出すことができます。リセット後、値はゼロになり、フラッシュメモリには 1 サイクルのみでアクセスできます。追加ウェイトステートは不要です。

## DCode インタフェース

DCode インタフェースは、CPU 側のシンプルな AHB インタフェースと、フラッシュアクセスコントローラのアービタにあるリクエストジェネレータで構成されています。DCode アクセスは、プリフェッチアクセスよりも優先されます。このインタフェースは、プリフェッチバッファのアクセス時間チューナーブロックを使用します。



## フラッシュアクセスコントローラ

このブロックは主に、プリフェッチ/ICode および DCode インタフェースの読み出しリクエスト間のシンプルなアービタです。

DCode インタフェースのリクエストは、その他のリクエストよりも優先されます。

### 4.2.3 フラッシュのプログラムおよび消去操作

STM32F3xx 内蔵フラッシュメモリは、インサーキットプログラミングまたはアプリケーション内プログラミングを使用してプログラム可能です。

**インサーキットプログラミング (ICP)** 方法は、フラッシュメモリの内容全体を更新するために使用します。マイクロコントローラにユーザアプリケーションをロードするために JTAG、SWD プロトコルやブートローダを使用します。ICP は、迅速かつ効率的な設計時の繰り返し操作を提供し、不要なパッケージ処理やデバイスのソケット処理を排除できます。

ICP の方法とは対照的に、**アプリケーション内プログラミング (IAP)** では、メモリにプログラミングデータをダウンロードするために、マイクロコントローラでサポートされる任意の通信インタフェース (I/O、USB、CAN、UART、I<sup>2</sup>C、SPI など) を使用できます。IAP を使用して、ユーザはアプリケーション実行中にフラッシュメモリを再プログラムできます。ただし、アプリケーションの一部は ICP を使用してフラッシュメモリ内に事前にプログラムする必要があります。

プログラムおよび消去操作は、次の 7 つのフラッシュレジスタを通じて管理されます。

- キーレジスタ (FLASH\_KEYR)
- オプションバイトキーレジスタ (FLASH\_OPTKEYR)
- フラッシュ制御レジスタ (FLASH\_CR)
- フラッシュステータスレジスタ (FLASH\_SR)
- フラッシュアドレスレジスタ (FLASH\_AR)
- オプションバイトレジスタ (FLASH\_OBR)
- 書き込み保護レジスタ (FLASH\_WRPR)

進行中のフラッシュメモリ操作は、CPU がフラッシュメモリにアクセスしないかぎり、CPU をブロックしません。

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

フラッシュメモリ上でのプログラムおよび消去操作 (書き込み/消去) では、内部 RC オシレータ (HSI) をオンにする必要があります。

### フラッシュメモリのロック解除

リセット後、FPEC は不要な書き込みまたは消去操作から保護されます。FLASH\_CR レジスタは、OBL の再ロードに使用する OBL LAUNCH ビットを除き、書き込みモードではアクセスできません。FLASH\_CR レジスタへのアクセスを開くには、FLASH\_KEYR レジスタにロック解除シーケンスを書き込む必要があります。シーケンスは FLASH\_KEYR レジスタへの 2 つの書き込み操作で構成されません。

1. KEY1 = 0x45670123 を書き込む
2. KEY2 = 0xCDEF89AB を書き込む

シーケンスを誤ると、次のリセットまで FPEC と FLASH\_CR レジスタがロックされます。

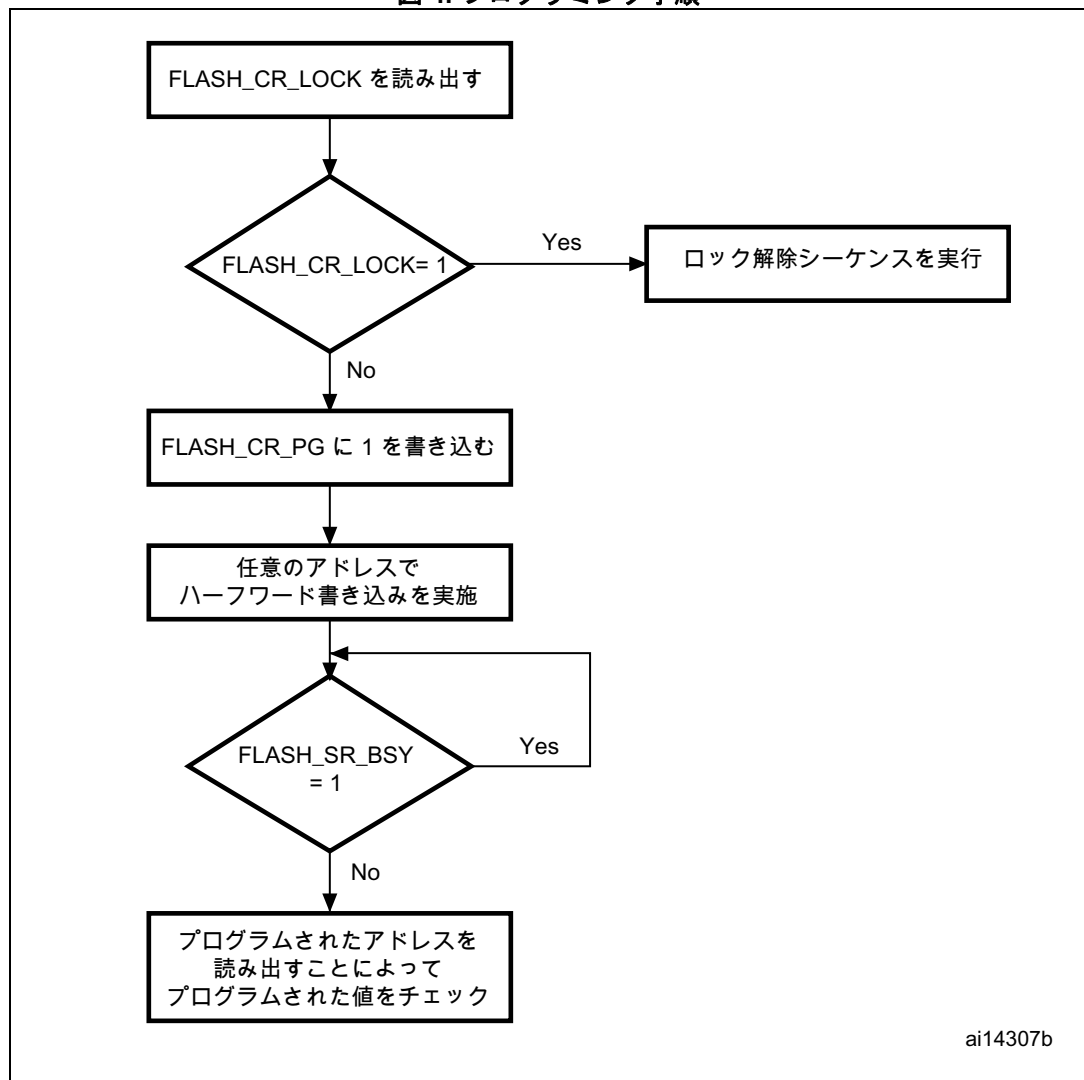
キーシーケンスを誤ると、バスエラーが検出され、ハードフォールト割り込みが発生します。これは最初の書き込みサイクルの後で KEY1 が一致しない場合、または 2 番目の書き込みサイクルで KEY1 が正しく書き込まれるが KEY2 が一致しない場合に行われます。

FPEC と FLASH\_CR レジスタは、ユーザソフトウェアで FLASH\_CR レジスタの LOCK ビットに 1 を書き込むと再びロックできます。

## メインフラッシュメモリプログラム

メインフラッシュメモリは、一度に 16 ビットずつプログラム可能です。プログラム操作は、FLASH\_CR レジスタの PG ビットがセットされた状態で、CPU によってメインフラッシュメモリアドレスにハーフワードが書き込まれた場合に開始します。ハーフワード長以外のデータの書き込みは、バスエラーの原因となり、ハードフォールト割り込みが発生します。

図 4. プログラミング手順



フラッシュメモリインタフェースではメインフラッシュメモリのアドレス指定された位置の値を事前に読み出し、それが消去されたことをチェックします。消去されていない場合、プログラム操作はスキップされ、FLASH\_SR レジスタの PGERR ビットによって警告が発行されます（唯一の例外は、0x0000 がプログラムされている場合です。この場合、位置は 0x0000 に正しくプログラムされ、

PGERR ビットはセットされていません)。メインフラッシュメモリのアドレス指定された位置が FLASH\_WRPRTERR レジスタによって書き込み保護されている場合、プログラム操作はスキップされ、FLASH\_SR レジスタの WRPRTERR ビットによって警告が発行されます。プログラム操作の終了は、FLASH\_SR レジスタの EOP ビットによって示されます。

標準モードのメインフラッシュメモリのプログラミングシーケンスは、次のようになっています。

1. FLASH\_SR レジスタの BSY ビットを確認し、進行中のメインフラッシュメモリ操作がないことを確認します。
2. FLASH\_CR レジスタの PG ビットをセットします。
3. 任意のアドレスでデータ書き込み（ハーフワード）を実施します。
4. FLASH\_SR レジスタの BSY ビットがリセットされるまで待ちます。
5. FLASH\_SR レジスタの EOP フラグ（プログラミング操作が成功した場合にセットされる）をチェックし、その後ソフトウェアによってクリアします。

**注：** FLASH\_SR レジスタの BSY ビットがセットされていると、書き込みモードではレジスタにアクセスできません。

## フラッシュメモリの消去

フラッシュメモリはページごとに、または完全（全体消去）に消去できます。

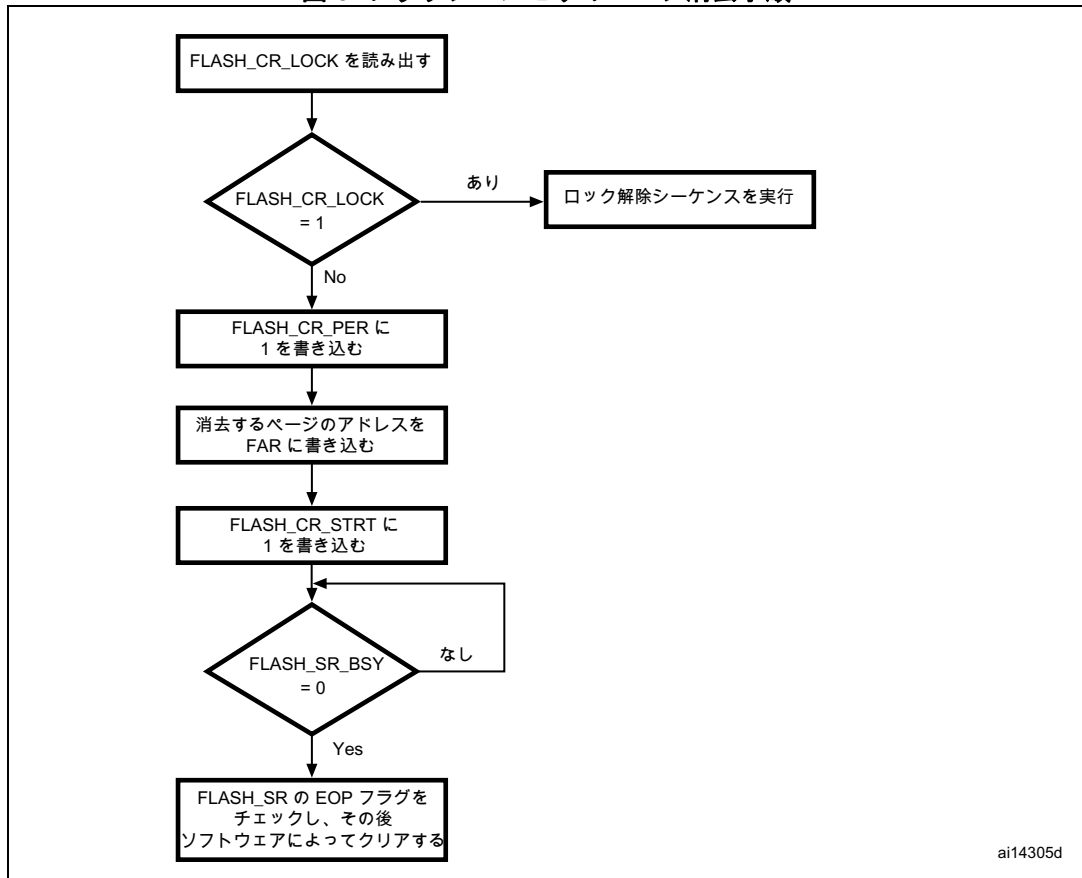
### ページ消去

ページを消去するには、次の手順に従います。

1. FLASH\_CR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH\_CR レジスタの PER ビットをセットします。
3. 消去するページを選択するために、FLASH\_AR レジスタをプログラムします。
4. FLASH\_CR レジスタの STRT ビットをセットします（下の注を参照）。
5. BSY ビットがリセットされるのを待ちます。
6. FLASH\_SR レジスタの EOP フラグ（消去操作が成功した場合にセットされる）をチェックし、その後ソフトウェアによってクリアします。
7. EOP フラグをクリアします。

**注：** STRT ビットをセットして、少なくとも 1 CPU サイクル後に BSY ビットが「0」と等しくなるかどうかについて、ソフトウェアでチェックを開始する必要があります。

図 5. フラッシュメモリのページ消去手順



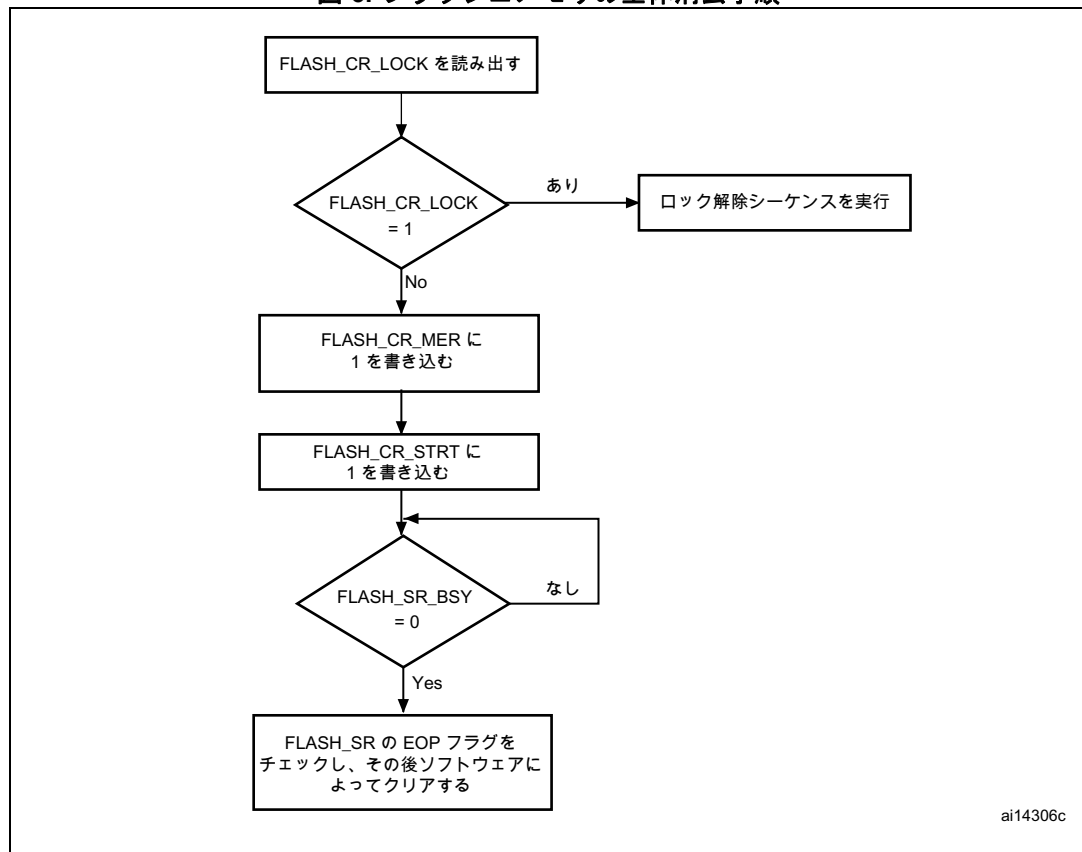
## 全体消去

全体消去コマンドは、フラッシュメモリのユーザページを完全に消去する場合に使用できます。情報ブロックは、この手順の影響を受けません。次のシーケンスを推奨します。

1. FLASH\_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH\_CR レジスタの MER ビットをセットします。
3. FLASH\_CR レジスタの STRT ビットをセットします（下の注を参照）。
4. BSY ビットがリセットされるのを待ちます。
5. FLASH\_SR レジスタの EOP フラグ（消去操作が成功した場合にセットされる）をチェックし、その後ソフトウェアによってクリアします。
6. EOP フラグをクリアします。

**注：** *STRT* ビットをセットして、少なくとも 1 CPU サイクル後に BSY ビットが「0」と等しくなるかどうかについて、ソフトウェアでチェックを開始する必要があります。

図 6. フラッシュメモリの全体消去手順



## オプションバイトのプログラミング

オプションバイトは、通常のユーザアドレスとは異なる方法でプログラムされます。オプションバイト数は、8 に制限されます（書き込み保護に 4 バイト、読み出し保護に 1 バイト、ハードウェア設定に 1 バイト、データストレージに 2 バイト）。FPEC のロック解除後、ユーザは FLASH\_OPTKEYR レジスタに同じ KEYS のセット（KEY1 および KEY2）を書き込むことでオプションバイトのプログラミングを許可する必要があります（キー値については [フラッシュメモリのロック解除](#) を参照）。次に、FLASH\_CR レジスタの OPTWRE ビットがハードウェアによってセットされ、ユーザは任意のフラッシュアドレスで、FLASH\_CR レジスタの OPTPG ビットをセットし、ハーフワードの書き込み操作を実施する必要があります。

本当に消去されたことをチェックするために、アドレス指定されたオプションバイトの値が最初に読み出されます。消去されていない場合、プログラム操作はスキップされ、FLASH\_SR レジスタの WRPRERR ビットによって警告が発行されます。プログラム操作の終了は、FLASH\_SR レジスタの EOP ビットによって示されます。

LSB 値は、プログラム操作の開始前に、MSB に自動的に補完されます。これにより、オプションバイトとその補数は常に正しくなります。

シーケンスは、次のとおりです。

- FLASH\_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
- FLASH\_CR レジスタの OPTWRE ビットをロック解除します。
- FLASH\_CR レジスタの OPTPG ビットをセットします。
- 任意のアドレスにデータ（ハーフワード）を書き込みます。
- BSY ビットがリセットされるのを待ちます。
- プログラムされた値を読み出し、確認します。

フラッシュメモリの読み出し保護オプションが保護された状態から保護されていない状態に変更された場合、読み出し保護オプションの再プログラミングの前にメインフラッシュメモリの全体消去が実施されます。読み出し保護オプション以外のオプションを変更したい場合は、全体消去は実施されません。読み出し保護オプションバイトが消去されていると、フラッシュメモリは保護状態となります。

## 消去手順

オプションバイトの消去シーケンス（OPTERASE）は、次のとおりです。

- FLASH\_SR レジスタの BSY ビットを読み出し、進行中のフラッシュメモリ操作がないことを確認します。
- FLASH\_CR レジスタの OPTWRE ビットをロック解除します。
- FLASH\_CR レジスタの OPTER ビットをセットします。
- FLASH\_CR レジスタの STRT ビットをセットします。
- BSY がリセットされるのを待ちます。
- 消去されたオプションバイトを読み出し、検証します。

## 4.3 メモリ保護

フラッシュメモリのユーザエリアは、信頼されないコードによる読み出しから保護できます。フラッシュメモリのページは、プログラムカウンタの内容の損失による不要な書き込みから保護することもできます。書き込み保護の単位は 2 ページです。

### 4.3.1 読み出し保護

読み出し保護は、RDP オプションバイトをセットし、新しい RDP オプションバイトを再ロードするシステムリセットを適用することで有効になります。

**注：** *デバッガが JTAG/SWD を通じて接続されている間に読み出し保護がセットされる場合、システムリセットの代わりに POR (パワーオンリセット) を適用してください。*

読み出し保護には、保護なし (レベル 0) から最大保護またはデバッグなし (レベル 2) までの 3 つのレベルがあります。

フラッシュメモリは、RDP オプションバイトとその補数に、表 8 に示す値のペアが含まれる場合に保護されます。

表 8. フラッシュメモリの読み出し保護ステータス

RDP バイト値	RDP 補数	読み出し保護レベル
0xAA	0x55	レベル 0 (ST 出荷時の設定)
0xAA または 0xCC を除く任意の値	0x55 および 0x33 以外の任意の値 (相補である必要はない)	レベル 1
0xCC	0x33	レベル 2

保護レベルにかかわらず、システムメモリ領域は読み出しアクセス可能です。プログラム/消去操作の場合は、アクセスできません。

#### レベル 0 : 保護なし

メインメモリのフラッシュ領域に、読み出し、プログラム、および消去操作を実行できます。すべての操作でオプションバイトにもアクセスできます。



## レベル 1：読み出し保護

これは、RDP オプションバイトが消去された場合のデフォルトの保護レベルです。RDP 値が 0xAA および 0xCC 以外の任意の値である場合に、または補数が正しくない場合でも、併せて定義されます。

- **ユーザモード**：ユーザモードで実行されるコードにより、すべての操作でメインメモリのフラッシュやオプションバイトにアクセスできます。
- **デバッグ、ブート RAM、およびブートローダモード**：デバッグモード、またはコードをブート RAM またはブートローダから実行している場合に、メインフラッシュメモリやバックアップレジスタ (RTC の RTC\_BKPxR) には完全にアクセスできません。これらのモードでは、簡単な読み出しアクセスでもバスエラーやハードフォールト割り込みが発生します。メインメモリはプログラム/消去保護されており、悪意のある、または許可されていないユーザがダンプルーチンによってユーザコードを再プログラミングするのを防ぎます。任意にプログラム/消去操作が行われると、フラッシュステータスレジスタ (FLASH\_SR) の PGERR フラグがセットされます。レベル 0 に戻すために RDP が値 0xAA に再プログラムされると、メインメモリのフラッシュの全体消去が実施され、バックアップレジスタ (RTC の RTC\_BKPxR) がリセットされます。

## レベル 2：デバッグなし

このレベルでは、保護レベル 1 が保証されます。さらに、Cortex<sup>®</sup>-M4 のデバッグ機能が無効化されます。結果として、デバッグポート、RAM からのブート (ブート RAM モード)、およびシステムメモリからのブート (ブートローダモード) は使用できなくなります。ユーザ実行モードでは、メインフラッシュメモリのすべての操作が許可されます。逆に、オプションバイトでは、読み出しおよびプログラム操作のみを実行できます。

オプションバイトは消去できません。さらに、RDP バイトはプログラムできません。そのため、レベル 2 は解除する方法がありません。これは、不可逆操作です。RDP バイトをプログラムしようとすると、Flash\_SR レジスタの保護エラーフラグ WRPRERR がセットされ、割り込みが生成されます。

**注：** リセット後、デバッグ機能は無効化されます。

**STMicroelectronics は、レベル 2 保護がセットされている不良部品には分析を実施できません。**

表 9. アクセス状態対保護レベルと実行モード

領域	保護レベル	ユーザ実行			デバッグ/RAM からのブート/ ローダからのブート		
		読み出し	書き込み	消去	読み出し	書き込み	消去
メインフラッシュ メモリ	1	あり	あり	あり	なし	なし	なし <sup>(3)</sup>
	2	あり	あり	あり	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>
システム メモリ <sup>(2)</sup>	1	あり	なし	なし	あり	なし	なし
	2	あり	なし	なし	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>
オプション バイト	1	あり	あり <sup>(3)</sup>	あり	あり	あり <sup>(3)</sup>	あり
	2	あり	あり <sup>(4)</sup>	なし	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>
バックアップ レジスタ	1	あり	あり	N/A	なし	なし	なし <sup>(5)</sup>
	2	あり	あり	N/A	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>

1. 保護レベル 2 が有効な場合、デバッグブート、RAM からのブート、およびシステムメモリからのブートは無効化されます。
2. システムメモリは、保護レベル (0、1、または 2) および実行モードにかかわらず、読み出しアクセスのみ可能です。
3. メインフラッシュメモリは、すべてのレベルの保護が無効化された状態 (0xAA) で RDP オプションバイトがプログラミングされると消去されます。
4. RDP バイトを除く、すべてのオプションバイトをプログラムできます。
5. バックアップレジスタは、RDP がレベル 1 からレベル 0 に変更された場合にのみ消去されます。

## 読み出し保護レベルの変更

レベル 0 からレベル 1 への移動は、RDP バイトの値を任意の値 (0xCC を除く) に変更することで、簡単に行うことができます。RDP バイトに 0xCC 値をプログラミングすることで、直接レベル 0 またはレベル 1 からレベル 2 に移動することができます。逆に、レベル 0 (保護なし) への変更は、メインフラッシュメモリの全体消去操作なしでは行うことができません。この全体消去は、RDP バイトに 0xAA がプログラムされると直ちに実施されます。

**注：** 全体消去コマンドが使用されると、バックアップレジスタ (RTC の RTC\_BKPxR) もリセットされます。

保護レベルの変更を確認するには、フラッシュ制御レジスタの OBL\_LAUNCH ビットを通じてオプションバイトを再ロードする必要があります。

### 4.3.2 書き込み保護

書き込み保護は、2 ページの単位で実装されています。これは、WRP[1:0] オプションバイトを設定し、FLASH\_CR レジスタの OBL\_LAUNCH ビットをセットして再ロードすることで有効化されます。

保護されたページでプログラムまたは消去操作が実施された場合、フラッシュメモリはフラッシュメモリステータスレジスタ (FLASH\_SR) の WRPRERR 保護エラーフラグを返します。

## 書き込み保護解除

書き込み保護を無効にするために、2つの応用事例が提供されています。

- ケース 1: 書き込み保護解除後に、読み出し保護を無効化します。
  - フラッシュメモリ制御レジスタ (FLASH\_CR) の OPTER ビットを使用して、オプションバイト領域全体を消去します。
  - メモリの保護を解除するために、RDP バイトにコード 0xAA をプログラムします。この操作では、強制的にメインフラッシュメモリを全体消去します。
  - フラッシュ制御レジスタ (FLASH\_CR) の OBL\_LAUNCH ビットをセットしてオプションバイト (および新しい WRP[3:0] バイト) を再ロードし、書き込み保護を無効化します。
- ケース 2: 書き込み保護解除後に読み出し保護はアクティブ状態を維持。ユーザブートローダによるアプリケーション内プログラミングで有効です。
  - フラッシュメモリ制御レジスタ (FLASH\_CR) の OPTER ビットを使用して、オプションバイト領域全体を消去します。
  - フラッシュ制御レジスタ (FLASH\_CR) の OBL\_LAUNCH ビットをセットしてオプションバイト (および新しい WRP[3:0] バイト) を再ロードし、書き込み保護を無効化します。

### 4.3.3 オプションバイトブロックの書き込み保護

オプションバイトは、デフォルトで常に読み出しアクセス可能であり、また書き込み保護されています。オプションバイトへの書き込みアクセス (プログラム/消去) には、OPTKEYR にキーのシーケンス (ロックと同じ) を書き込む必要があります。正しいキーのシーケンスによってオプションバイトへの書き込みアクセスができ、これは FLASH\_CR レジスタの OPTWRE のセットによって示されます。書き込みアクセスは、ソフトウェアを通じてビットをリセットすることで無効にできます。

## 4.4 フラッシュ割り込み

表 10. フラッシュ割り込みリクエスト

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

## 4.5 フラッシュレジスタの説明

フラッシュメモリレジスタは、32 ビットワードでアクセスする必要があります（ハーフワードおよびバイトアクセスは許可されません）。

### 4.5.1 フラッシュアクセス制御レジスタ（FLASH\_ACR）

アドレスオフセット：0x00  
リセット値：0x0000 0030

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

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **PRFTBS**：プリフェッチバッファステータス

このビットは、プリフェッチバッファのステータスを提供します。

0：プリフェッチバッファは無効です。

1：プリフェッチバッファは有効です。

ビット 4 **PRFTBE**：プリフェッチバッファ有効化

0：プリフェッチは無効です。

1：プリフェッチは有効です。

ビット 3 **HLFCYA**：フラッシュハーフサイクルアクセス有効化

0：ハーフサイクルは無効です。

1：ハーフサイクルは有効です。

ビット 2:0 **LATENCY[2:0]**：遅延

これらのビットは、フラッシュアクセス時間に対する HCLK 周期の割合を表します。

000：0 < HCLK ≤ 24 MHz の場合、0 ウェイトステート

001：24 MHz < HCLK ≤ 48 MHz の場合、1 ウェイトステート

010：48 < HCLK ≤ 72 MHz の場合、2 ウェイトステート

### 4.5.2 フラッシュキーレジスタ（FLASH\_KEYR）

アドレスオフセット：0x04

リセット値：xxxx xxxx

これらのビットは書き込み専用で、読み出すと 0 を返します。

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

ビット 31:0 **FKEYR**：フラッシュキー

これらのビットは、フラッシュをロック解除するキーを表します。

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

アドレスオフセット : 0x08

リセット値 : XXXX XXXX

すべてのレジスタビットは書き込み専用で、読み出すと 0 を返します。

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

ビット 31:0 **OPTKEYR** : オプションバイトキー  
 これらのビットは、OPTWRE をロック解除するキーを表します。

## 4.5.4 フラッシュステータスレジスタ (FLASH\_SR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOP	WRPRT ERR	Res.	PG ERR	Res.	BSY
										r/w	r/w		r/w		r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **EOP** : 操作終了

フラッシュ操作 (プログラム/消去) の完了時に、ハードウェアによってセットされま  
 す。1 を書き込むとリセットされます。

**注:** 成功したプログラムや消去操作が終了するたびに、EOP がアサートされます。

ビット 4 **WRPRTERR** : 書き込み保護エラー

フラッシュメモリの書き込み保護されたアドレスをプログラムすると、ハードウェアに  
 よってセットされます。

1 を書き込むとリセットされます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **PGERR** : プログラミングエラー

プログラムする前に、プログラム対象のアドレスに「0xFFFF」以外の値が含まれる場合、  
 ハードウェアによってセットされます。

1 を書き込むとリセットされます。

**注:** FLASH\_CR レジスタの STRT ビットはプログラム操作開始前にリセットする必要  
 があります。

ビット 1 予約済みであり、リセット値のままにしておかなければなりません。

ビット 0 **BSY** : ビジー

これは、フラッシュ操作が進行中であることを示します。フラッシュ操作の開始時にセッ  
 トされ、操作が終了するかエラーが発生するとリセットされます。



## 4.5.5 フラッシュ制御レジスタ (FLASH\_CR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OBL_LAUNCH	EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
		r/w	r/w		r/w	r/w		r/w	r/w	r/w	r/w		r/w	r/w	r/w

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **OBL\_LAUNCH** : オプションバイトローディングの強制

1 にセットすると、このビットではオプションバイトの再ロードを強制的に行います。この操作によって、システムリセットが生成されます。

0 : インアクティブ  
1 : アクティブ

ビット 12 **EOPIE** : 操作終了割込み有効化

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

0 : 割込み生成は無効です。  
1 : 割込み生成は有効です。

ビット 11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10 **ERRIE** : エラー割り込みイネーブル

このビットは、FLASH\_SR レジスタに PGERR/WRPRTERR がセットされた場合に、エラーによる割込み生成を有効にします。

0 : 割込み生成は無効です。  
1 : 割込み生成は有効です。

ビット 9 **OPTWRE** : オプションバイトの書き込み有効化

セットすると、オプションバイトをプログラムできます。このビットは、FLASH\_OPTKEYR レジスタに正しいキーシーケンスを書き込む際にセットされます。このビットは、ソフトウェアでリセットできます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **LOCK** : ロック状態

1 のみを書き込みます。このビットがセットされている場合は、フラッシュがロックされていることを示します。このビットは、ロック解除シーケンスが検出されると、ハードウェアによってリセットされます。

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

ビット 6 **STRT** : 開始

このビットがセットされると、ERASE 操作がトリガされます。このビットはソフトウェアによってのみセットされ、BSY ビットがリセットされるとリセットされます。

ビット 5 **OPTER** : オプションバイトの消去

オプションバイトの消去が選択されます。

ビット 4 **OPTPG** : オプションバイトのプログラミング

オプションバイトのプログラミングが選択されます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

- ビット 2 **MER** : 全体消去  
 選択したすべてのユーザページが消去されます。
- ビット 1 **PER** : ページ消去  
 選択したページが消去されます。
- ビット 0 **PG** : プログラミング  
 フラッシュプログラミングが選択されます。

## 4.5.6 フラッシュアドレスレジスタ (FLASH\_AR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

このレジスタは、現在使用しているアドレスと最後に使用したアドレスで、ハードウェアによって更新されます。ページ消去操作では、選択したページを示すためにソフトウェアで更新する必要があります。

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

ビット 31:0 **FAR** : フラッシュアドレス

プログラミングを選択した場合はプログラムするアドレスを選択し、ページの消去を選択した場合は消去するページを選択します。

**注:** *FLASH\_SR レジスタの BSY ビットがセットされたときに、このレジスタへの書き込みアクセスがブロックされます。*

## 4.5.7 オプションバイトレジスタ (FLASH\_OBR)

アドレスオフセット : 0x1C

リセット値 : 0XXXXXXXX0X

保護レベルの通知、オプションバイトおよびユーザオプションのローディング中のエラーを含みません。

このレジスタのリセット値はオプションバイトでプログラムした値に依存し、OPTERR ビットのリセット値はオプションバイトローディング中のオプションバイトとその相補との比較に依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Data1								Data0								Res.	SRAM_PE	VDDA_MONITOR	nBOOT1	Res.	nRST_STDBY	nRST_STOP	WDG_SW	Res.						RDPRT[1:0]	OPTERR			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r			r	r		r	r	r									r	r	r

ビット 31:24 Data1

ビット 23:16 Data0

ビット 15:8 **OBR** : ユーザオプションバイト

ビット 15 : 予約済みであり、リセット値に保持する必要があります。

ビット 14 : SRAM\_PE

ビット 13 : VDDA\_MONITOR

ビット 12 : nBOOT1

ビット 11 : 予約済みであり、リセット値に保持する必要があります。

ビット 10 : nRST\_STDBY

ビット 9 : nRST\_STOP

ビット 8 : WDG\_SW

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:1 **RDPRT[1:0]** : 読み出し保護レベルステータス

00 : 読み出し保護レベル 0 は有効です (ST 出荷時の設定)。

01 : 読み出し保護レベル 1 は有効です。

10 : 予約済み

11 : 読み出し保護レベル 2 は有効です。

**注 :** これらのビットは読み出し専用です。

ビット 0 **OPTERR** : オプションバイトロードエラー

セットされると、ロードされたオプションバイトとその補数が一致しないことが示されます。対応するバイトとその補数は、FLASH\_OBR または FLASH\_WRPR レジスタで 0xFF として読み出されます。

**注 :** このビットは読み出し専用です。



## 4.5.8 書き込み保護レジスタ (FLASH\_WRPR)

アドレスオフセット : 0x20

リセット値 : 0xFFFF FFFF

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

ビット 31:0 WRP : 書き込み保護

このレジスタには、OBL によってロードされた書き込み保護オプションバイトが含まれます。

これらのビットは読み出し専用です。

## 4.6 フラッシュレジスタマップ

表 11. フラッシュインタフェース - レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRFTBS	PRFTBE	HLFCYA	LATENCY [2:0]			
	リセット値																												1	1	0	0	0	0
0x004	FLASH_KEYR	FKEYR[31:0]																																
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x008	FLASH_OPTKEYR	OPTKEYR[31:0]																																
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x00C	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOP	WRPRERR	Res.	PGERR	Res.	BSY
	リセット値																												0	0		0		0
0x010	FLASH_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OBL_LAUNCH	EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG	
	リセット値																			0	0		0	0		1	0	0	0		0	0	0	
0x014	FLASH_AR	FAR[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



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

表 11. フラッシュインタフェース - レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x01C	FLASH_OBR	Data1								Data0								Res.	SRAM_PE	VDDA_MONITOR	nBOOT1	Res.	nRST_STDBY	nRST_STOP	WDG_SW	Res.								RDPRT[1:0]	OPTERR
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	x	x	x	x	
0x020	FLASH_WRPR	WRP[31:0]																																	
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 5 オプションバイトの説明

最大 8 つのオプションバイトがあります。これらは、アプリケーション要件によってエンドユーザが設定します。設定の例として、ウォッチドッグをハードウェアまたはソフトウェアモードに選択できます。

32 ビットワードは、オプションバイトで次に示すように分割されます。

表 12. オプションバイトのフォーマット

31-24	23-16	15:-8	7-0
相補オプションバイト 1	オプションバイト 1	相補オプションバイト 0	オプションバイト 0

情報ブロック内のこれらのバイトの構成は、表 13 に示すとおりです。

オプションバイトは、表 13 にリストされたメモリ位置またはオプションバイトレジスタ (FLASH\_OBR) から読み出すことができます。

**注：** *新しくプログラムされたオプションバイト (ユーザ、読み出し/書き込み保護) は、システムリセット後にロードされます。*

表 13. オプションバイトの構成

アドレス	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nData1	Data1	nData0	Data0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF F80C	nWRP3 <sup>(1)</sup>	WRP3 <sup>(1)</sup>	nWRP2 <sup>(1)</sup>	WRP2 <sup>(1)</sup>

1. STM32F303xB/C および STM32F358xC デバイスでのみ使用できます。STM32F303x6/8 および STM32F328x8 デバイスでは、リセット値のままにしておかなければなりません。

表 14. オプションバイトの説明

フラッシュ メモリアドレス	オプションバイト
0x1FFF F800	<p>ビット [31:24] : <b>nUSER</b></p> <p>ビット [23:16] : <b>USER</b> : ユーザオプションバイト (FLASH_OBR[15:8] に格納)</p> <p>このバイトは次の機能の設定に使用します。</p> <ul style="list-style-type: none"> <li>- ウォッチドッグイベントの選択 : ハードウェアまたはソフトウェア</li> <li>- STOP モードに移行するときにイベントをリセットします。</li> <li>- STANDBY モードに移行するときにイベントをリセットします。</li> </ul> <p>ビット 23 : 予約済み</p> <p>ビット 22 : <b>SRAM_PE</b></p> <p>SRAM ハードウェアパリティチェックはデフォルトで無効です。このビットによって、SRAM ハードウェアパリティチェックを有効にできます。</p> <ul style="list-style-type: none"> <li>0 : パリティチェックは有効です。</li> <li>1 : パリティチェックは無効です。</li> </ul> <p>ビット 21 : <b>VDDA_MONITOR</b></p> <p>このビットは、VDDA 外部電源でアナログ監視を選択します。</p> <ul style="list-style-type: none"> <li>0 : VDDA 電源供給スーパバイザは無効です。</li> <li>1 : VDDA 電源供給スーパバイザは有効です。</li> </ul> <p>ビット 20 : <b>nBOOT1</b></p> <p>このビットは、BOOT0 ピンと一緒にメインフラッシュメモリ、SRAM、またはシステムメモリからブートモードを選択します。 <a href="#">セクション 3.5 (60 ページ)</a> を参照してください。</p> <p>ビット 19 : 予約済みであり、リセット値のままにしておかなければなりません。</p> <p>ビット 18 : <b>nRST_STDBY</b></p> <ul style="list-style-type: none"> <li>0 : STANDBY モードに移行するときにリセットを生成します。</li> <li>1 : リセットは生成されません。</li> </ul> <p>ビット 17 : <b>nRST_STOP</b></p> <ul style="list-style-type: none"> <li>0 : STOP モードに移行するときにリセットを生成します。</li> <li>1 : リセットは生成されません。</li> </ul> <p>ビット 16 : <b>WDG_SW</b></p> <ul style="list-style-type: none"> <li>0 : ハードウェアウォッチドッグ</li> <li>1 : ソフトウェアウォッチドッグ</li> </ul> <p>ビット [15:8] : <b>nRDP</b></p> <p>ビット [7:0] : <b>RDP</b> : 読み出し保護オプションバイト</p> <p>このバイトの値はフラッシュメモリの保護レベルを定義します。</p> <ul style="list-style-type: none"> <li>0xAA : レベル0</li> <li>0XX (0xAA および 0xCC を除く) : レベル1</li> <li>0xCC : レベル2</li> </ul> <p>保護レベルは Flash_OBR フラッシュオプションバイトレジスタに格納されています (RDPRT ビット)。</p>
0x1FFF F804	<p><b>Datax</b> : ユーザデータストレージ用の 2 バイト。</p> <p>これらのアドレスは、オプションバイトプログラミング手順を使用してプログラミングできます。</p> <p>ビット [31:24] : <b>nData1</b></p> <p>ビット [23:16] : <b>データ 1</b> (FLASH_OBR[31:24] に格納)</p> <p>ビット [15:8] : <b>nData0</b></p> <p>ビット [7:0] : <b>データ 0</b> (FLASH_OBR[23:16] に格納)</p>

表 14. オプションバイトの説明 (続き)

フラッシュメモリアドレス	オプションバイト
0x1FFF F808	<p><b>WRP<sub>x</sub></b> : フラッシュメモリ書き込み保護オプションバイト</p> <p>ビット [31:24] : <b>nWRP1</b>                      ビット [23:16] : <b>WRP1</b> (FLASH_WRP[15:8] に格納)                      ビット [15:8] : <b>nWRP0</b>                      ビット [7:0] : <b>WRP0</b> (FLASH_WRP[7:0] に格納)</p> <p>0 : 書き込み保護がアクティブです。                      1 : 書き込み保護がアクティブになっていません。</p> <p>詳細については、<a href="#">セクション 4.3.2 : 書き込み保護</a>を参照してください。</p> <p><b>注 :</b> <b>WRP2</b> と <b>WRP3</b> が使用できない場合でも、それ以外はリセット値に保持する必要があります。</p>
0x1FFF F80C	<p><b>WRP<sub>x</sub></b> : フラッシュメモリ書き込み保護オプションバイト (STM32F303xB/C および STM32F358xC でのみ使用可能)</p> <p>ビット [31:24] : <b>nWRP3</b>                      ビット [23:16] : <b>WRP3</b> (FLASH_WRP[31:24] に格納)                      ビット [15:8] : <b>nWRP2</b>                      ビット [7:0] : <b>WRP2</b> (FLASH_WRP[23:16] に格納)</p> <p>ユーザオプションバイト WRP<sub>x</sub> の 1 ビットは、メインメモリブロックの 2 KB の 2 ページを保護するために使用されます。</p> <p>0 : 書き込み保護がアクティブです。                      1 : 書き込み保護がアクティブになっていません。</p> <p>メインフラッシュメモリ全体を保護するために、全部で 4 つのユーザオプションバイトが使用されています。</p> <p>WRP0 : 0 から 15 ページを書き込み保護します。                      WRP1 : 16 から 31 ページを書き込み保護します。                      WRP2 : 32 から 47 ページを書き込み保護します。<sup>(1)</sup>                      WRP3 : ビット 0-6 で 48 から 61 ページを書き込み保護し、ビット 7 で STM32F303xB/C および STM32F358xC の 62 から 127 ページ、STM32F303xD/E の 62 から 255 ページを書き込み保護します。</p>

1. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用できます。STM32F303x6/8 および STM32F328x8 デバイスでは、リセット値のままにしておかなければなりません。

システムリセットごとに、オプションバイトローダ (OBL) は情報ブロックを読み出して、オプションバイトレジスタ (FLASH\_OBR) と書き込み保護レジスタ (FLASH\_WRP) にデータを格納します。各オプションバイトにも情報ブロックに補数があります。オプションのローディング中に、オプションビットとその補数を確認することで、ローディングが正しく実行されたことを確認できます。そうでない場合は、オプションバイトエラー (OPTERR) が生成されます。比較エラーが発生した場合は、対応するオプションバイトは強制的に 0xFF になります。オプションバイトとその補数が両方も 0xFF (電氣的消去状態) と等しい場合に、比較回路は無効化されます。



## 6 巡回冗長検査計算ユニット (CRC)

### 6.1 概要

CRC (Cyclic Redundancy Check : 巡回冗長検査) 計算ユニットは、8、16、または 32 ビットデータワードと、ある生成多項式から、CRC コードを得るために使用されます。

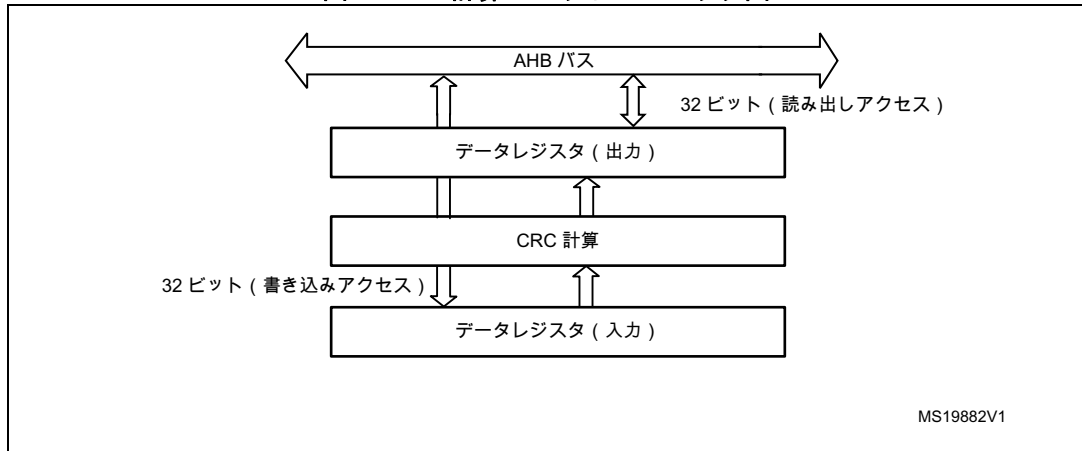
他のアプリケーションの中でも、CRC ベースのテクニックは、データ転送やストレージの整合性を確認するために使用されます。機能安全規格の範囲内では、このテクニックがフラッシュメモリの整合性を確認するひとつの手段となっています。CRC 計算ユニットは、実行時にソフトウェアのシグネチャ計算を支援します。リンク時に生成されて、特定のメモリ領域に保存されたりファレンスシグネチャと計算されたソフトウェアシグネチャが比較されます。

### 6.2 CRC の主な機能

- プログラム可能なサイズ (7、8、16、32 ビット) の完全にプログラム可能な多項式。
- サイズが 8、16、32 ビットのデータを取り扱います。
- プログラム可能な CRC の初期値
- シングル入力/出力 32 ビットデータレジスタ
- 計算時のバスのストールを避けるための入力バッファ
- データサイズが 32 ビットの場合、CRC の計算は AHB クロック 4 サイクル (HCLK) 以内に行われます。
- 汎用 8 ビットレジスタ (一時ストレージとして使用可能)
- I/O データの可逆性のオプション

## 6.3 CRC の機能説明

図 7. CRC 計算ユニットのブロック図



CRC 計算ユニットは、1つの 32 ビット読み出し/書き込みデータレジスタ (CRC\_DR) を持っています。このレジスタを使用して、新しいデータを入力し (書き込みアクセス)、前回の CRC 計算結果を保持します (読み出しアクセス)。

データレジスタへの書き込み操作のたびに、前回の CRC 値 (CRC\_DR に格納) と新しい値の組み合わせが作成されます。CRC 計算は、書き込まれるデータのフォーマットに応じ、32 ビットデータワード全体に対して、またはバイト単位で行われます。

CRC\_DR レジスタは、ワード、右詰め、右詰め、右詰めのバイトによってアクセスできます。他のレジスタについては、32 ビットアクセスのみ可能です。

計算の時間はデータ幅に依存します。

- 32 ビットの場合、4 AHB クロックサイクル
- 16 ビットの場合、2 AHB クロックサイクル
- 8 ビットの場合、1 AHB クロックサイクル

入力バッファを使うと、前回の CRC 計算によるウェイトステートを待つことなく、すぐに第 2 のデータを書き込むことができます。

与えられたバイト数に対する書き込みアクセス数を最小限に抑えるために、データサイズを動的に調節することができます。たとえば、5 バイトの CRC は、1 ワードの書き込みと、それに続く 1 バイトの書き込みで計算することができます。

入力データを逆にして、さまざまなエンディアンネス方式を管理することができます。逆転操作は、CRC\_CR レジスタの REV\_IN[1:0] ビットに応じて、8、16、および 32 ビットで行うことができます。

たとえば、入力データ 0x1A2B3C4D は、CRC 計算では以下のように使用されます。

- 0x58D43CB2 (バイト単位でビットが逆転)
- 0xD458B23C (ハーフワード単位でビットが逆転)
- 0xB23CD458 (フルワードでビットが逆転)

また、出力データも、CRC\_CR レジスタの REV\_OUT ビットをセットすることによって逆にすることができます。

操作はビットレベルで行われます。たとえば、出力データ 0x11223344 は 0x22CC4488 に変換されます。

CRC 計算機は、CRC\_CR レジスタの RESET 制御ビットを使用して、プログラム可能な値に初期化することができます (デフォルト値は 0xFFFFFFFF)。

CRC の初期値は、CRC\_INIT レジスタでプログラムすることができます。CRC\_DR レジスタは、CRC\_INIT レジスタの書き込みアクセス時に自動的に初期化されます。

CRC\_IDR レジスタを使用して、CRC 計算に関する一時的な値を保持することができます。このレジスタは、CRC\_CR レジスタの RESET ビットによる影響を受けません。

## 多項式のプログラミング可能性

多項式係数は CRC\_POL レジスタを介して完全にプログラム可能であり、多項式のサイズは CRC\_CR レジスタの POLYSIZE[1:0] ビットをプログラムすることにより、7、8、16、または 32 ビットに設定できます。偶多項式はサポートされていません。

CRC データが 32 ビット未満であれば、その値は CRC\_DR レジスタの下位ビットから読み出すことができます。

信頼できる CRC 計算結果を得るために、CRC 計算の実行中に多項式の値やサイズを変更することはできません。そのため、CRC 計算が進行中である場合、アプリケーションは計算をリセットするか、または CRC\_DR の読み出しを行ってから、多項式を変更しなければなりません。

多項式のデフォルト値は、CRC-32 (イーサネット) 多項式、0x4C11DB7。

## 6.4 CRC レジスタ

### 6.4.1 データレジスタ (CRC\_DR)

アドレスオフセット : 0x00

リセット値 : 0xFFFF FFFF

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

#### ビット 31:0 DR[31:0] : データレジスタビット

このレジスタを使用して、CRC 計算機に新しいデータを書き込みます。

読み出し時には、前回の CRC 計算結果を保持します。

データサイズが 32 ビット未満であれば、下位ビットを使用して正しい値の書き込み/読み出しを行います。



## 6.4.2 独立型データレジスタ (CRC\_IDR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:8 予約済み、クリア状態を保つ必要があります。

ビット 7:0 **IDR[7:0]** : 汎用 8 ビットデータレジスタビット

これらのビットは、1 バイトの一時的なストレージ領域として使用できます。

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

## 6.4.3 制御レジスタ (CRC\_CR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:8 予約済み、クリア状態を保つ必要があります。

ビット 7 **REV\_OUT** : 出力データを逆にします。

このビットは、出力データビット順序の反転を制御します。

0 : ビットの順序は変わりません。

1 : 出力フォーマットのビットが反転しています。

ビット 6:5 **REV\_IN[1:0]** : 入力データを逆にします。

これらのビットは、入力データビット順序の反転を制御します。

00 : ビットの順序は変わりません。

01 : バイト単位でビットが反転しています。

10 : ハーフワード単位でビットが反転しています。

11 : ワード単位でビットが反転しています。

ビット 4:3 **POLYSIZE[1:0]** : 多項式のサイズ

これらのビットは、多項式のサイズを制御します。

00 : 32 ビットの多項式

01 : 16 ビットの多項式

10 : 8 ビットの多項式

11 : 7 ビットの多項式

ビット 2:1 予約済み、クリア状態を保つ必要があります。

ビット 0 **RESET** : RESET ビット

このビットは、CRC 計算ユニットをリセットし、CRC\_INIT レジスタに格納された値にデータレジスタをセットするために、ソフトウェアによってセットされます。このビットはセットのみが可能で、ハードウェアによって自動的にクリアされます。

## 6.4.4 CRC の初期値 (CRC\_INIT)

アドレスオフセット : 0x10

リセット値 : 0xFFFF FFFF

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

ビット 31:0 **CRC\_INIT** : プログラム可能な CRC の初期値

このレジスタを使用して、CRC の初期値を書き込みます。

## 6.4.5 CRC 多項式 (CRC\_POL)

アドレスオフセット : 0x14

リセット値 : 0x04C11DB7

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

ビット 31:0 **POL[31:0]** : プログラム可能な多項式

このレジスタを使用して、CRC 計算に使用される多項式の係数を書き込みます。

多項式のサイズが 32 ビット未満であれば、下位ビットを使用して正しい値をプログラムする必要があります。

## 6.4.6 CRC レジスタマップ

表 15. CRC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	CRC_DR	DR[31:0]																																
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																	
0x08	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																	
0x10	CRC_INIT	CRC_INIT[31:0]																																
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	CRC_POL	多項式係数																																
	リセット値	0x04C11DB7																																

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



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

## 7 電源制御 (PWR)

### 7.1 電源

内蔵レギュレータが STM32F3xx デバイスに搭載されています。

- 内蔵レギュレータは、STM32F3xx MCU で有効化されます。  
STM32F303x6/8/B/C/D/E デバイスでは、2.0 V ~ 3.6 V の動作電源電圧 ( $V_{DD}$ ) と 2.0 V ~ 3.6 V のアナログ電源電圧 ( $V_{DDA}$ ) が必要です。内部の 1.8 V デジタル電源を供給するために、レギュレータが搭載されています。
- 内蔵レギュレータは、STM32F3x8 MCU で無効化されます。  
STM32F3x8 デバイスでは、1.8 V +/- 8% の動作電源電圧 ( $V_{DD}$ ) と 1.65 V ~ 3.6 V のアナログ電源電圧 ( $V_{DDA}$ ) が必要です。搭載されたレギュレータはオフの場合、レギュレータの出力は直接  $V_{DD}$  から供給されます。

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

図 8. 電源の概要 (STM32F303x デバイス)

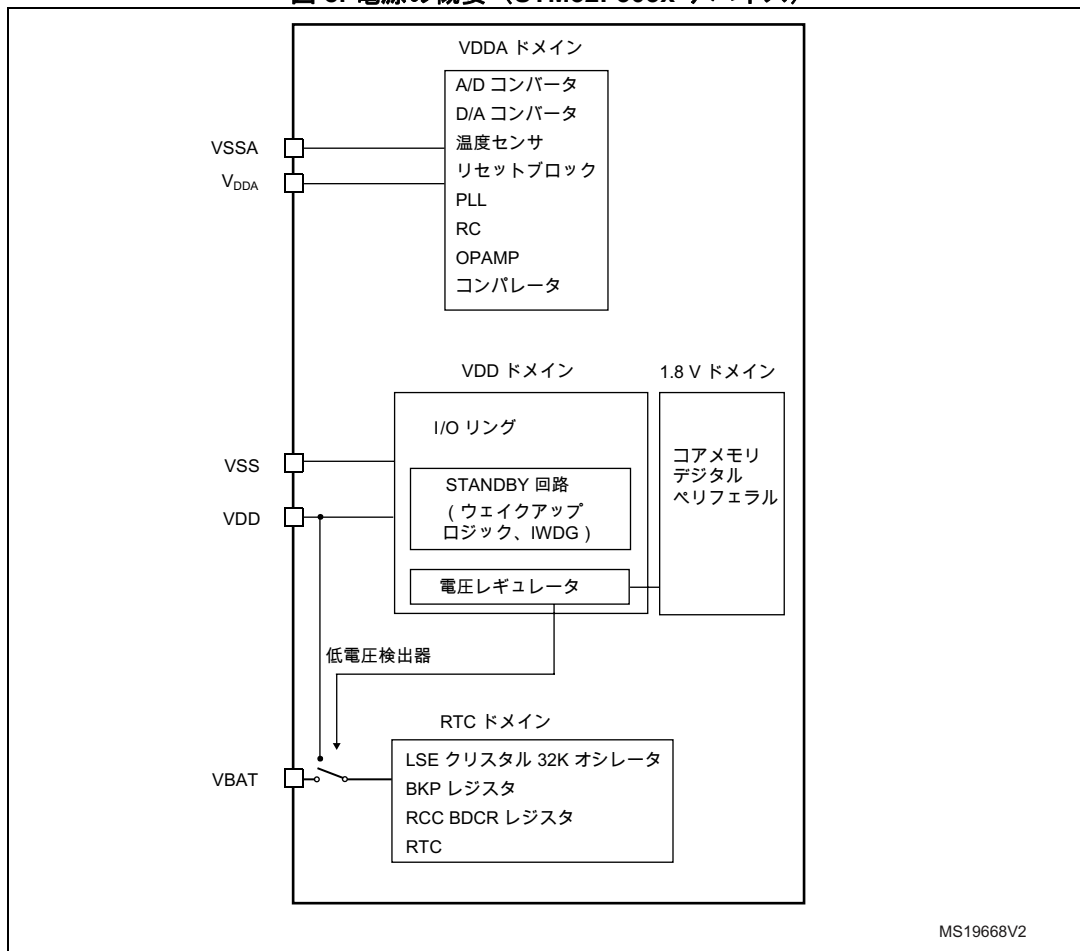
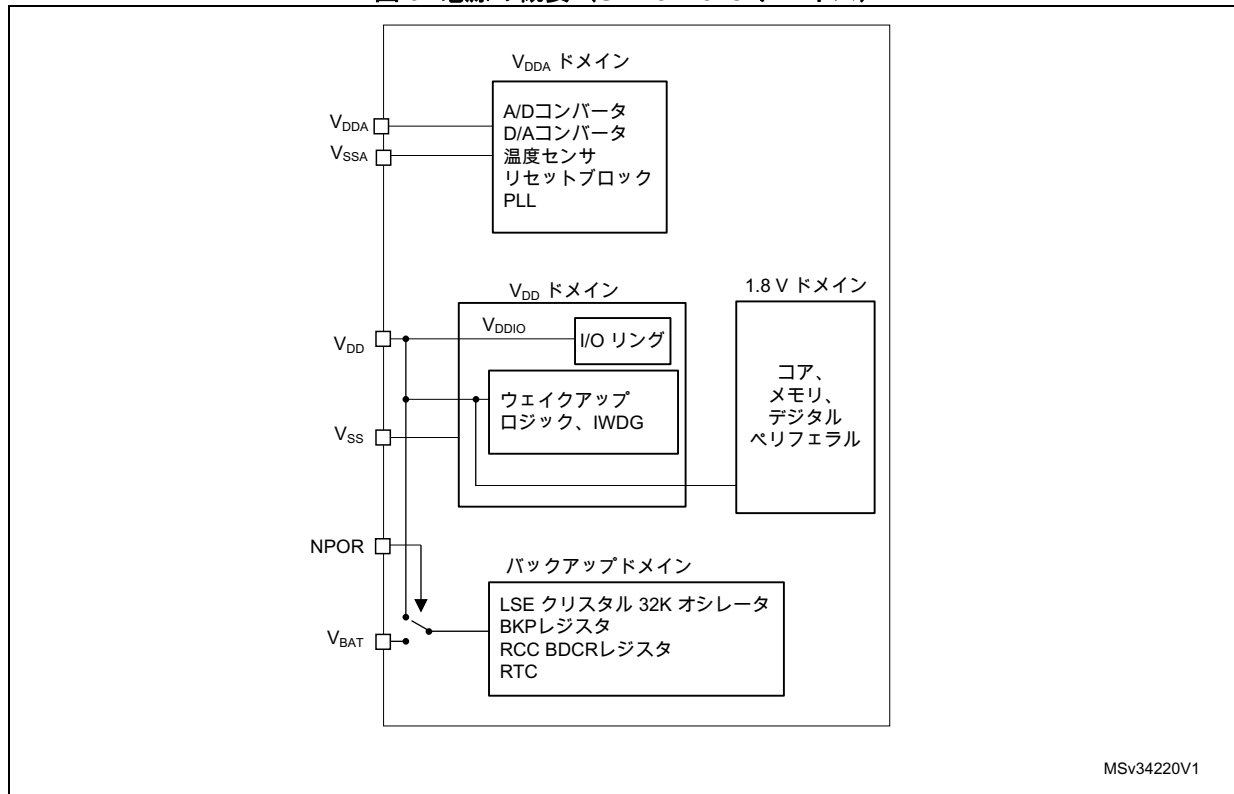


図 9. 電源の概要 (STM32F3x8 デバイス)



使用可能な電源電圧は、次のとおりです。

- $V_{DD}$  および  $V_{SS}$  : I/O およびコアの外部電源。  
供給電圧は  $V_{DD}$  および  $V_{SS}$  ピンを通じて外部より供給されます。 $V_{DD} = 2.0$  から  $3.6$  V (STM32F303x6/8/B/C/D/E デバイス) または  $1.8$  V  $\pm 8\%$  (STM32F3x8 デバイス)。  
1.8 V モードの外部電源を選択した場合、レギュレータの出力は直接  $V_{DD}$  から供給され、VDD18 ドメインはこれによって直接駆動します。  
 $V_{DD}$  は、常に  $V_{DDA}$  以下である必要があります。
- $V_{DD18} = 1.65$  から  $1.95$  V (VDD18 ドメイン) : デジタルコア、SRAM、およびフラッシュメモリの電源。  
 $V_{DD18}$  は、内部電圧レギュレータを通じて内部的に生成するか (STM32F303x6/8/B/C/D/E)、もしくはレギュレータのバイパス時に外部  $V_{DD}$  ピンから直接提供することができます (STM32F3x8)。
- $V_{DDA}$ 、 $V_{SSA} = 2.0$  から  $3.6$  V (STM32F303x6/8/B/C/D/E) または  $1.65$  から  $3.6$  V (STM32F3x8) : ADC、DAC、コンパレータ、オペアンプ、温度センサ、PLL、HSI 8 MHz オシレータ、LSI 40 kHz オシレータ、およびリセットブロックの外部電源。  
OPAMP および DAC が使用されているときには、 $V_{DDA}$  は  $2.4$  から  $3.6$  V の範囲である必要があります。  
ADC が使用されているときには、 $V_{DDA}$  は  $1.8$  から  $3.6$  V の範囲である必要があります。  
 $V_{DDA} < V_{DD} \sim 0.4$  V であることは禁じられています。この条件を満たすよう、外部ショットキーダイオードを  $V_{DD}$  と  $V_{DDA}$  の間に配置する必要があります。
- $V_{BAT} = 1.65$  から  $3.6$  V : RTC、LSE オシレータ、PC13 から PC15、およびバックアップレジスタのバックアップ電源 ( $V_{DD}$  が存在しない場合)。 $V_{DD}$  電源が存在する場合、内部電源スイッチはバックアップ電源を  $V_{DD}$  に切り替えます。 $V_{BAT}$  が使用されない場合、 $V_{DD}$  に接続する必要があります。

## 7.1.1 独立した A/D および D/A コンバータ用電源と基準電圧

変換の精度を向上させるため、ADC および DAC には独立した電源が供給されます。この電源は、PCB のノイズを防ぐために、個別にフィルタシールドすることができます。

ADC および DAC 用の電源入力は、独立した VDDA ピンから供給されます。ADC 電源のグラウンドは、独立した VSSA ピンとして用意されています。

### 144 および 100 ピンのパッケージの接続

低電圧入出力時の精度を確保するため、独立した外部基準電圧を VREF+ に接続できます。VREF+ は、アナログ入力 (ADC) または出力 (DAC) 信号のフルスケール値に該当する最大電圧です。

### 64 ピン、48 ピン、および 32 ピンのパッケージの接続

これらのパッケージでは、VREF+ および VREF- ピンは使用できません。これらのピンは、ADC 用の電源入力 (VDDA) とグラウンド (VSSA) に、それぞれ内部接続されています。

VDDA 電源/基準電圧は VDD 以上である場合があります。単電源を使用している場合、ノイズフリーの VDDA/基準電圧を確保するために、外部フィルタ回路を通じて VDDA を VDD に外部的に接続することができます。

VDDA が VDD と異なる場合、VDDA は VDD 以上である必要があります。パワーアップ/パワーダウン時に VDDA と VDD の間で安全な電位差を維持するために、VDD と VDDA の間で外部ショットキーダイオードを使用できます。許容可能な差の最大値に関してはデータシートを参照してください。

## 7.1.2 バッテリバックアップドメイン

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

VBAT ピンから RTC ユニット、LSE オシレータ、および PC13 から PC15 の I/O に電源が供給され、主電源がオフの場合でも RTC が動作できるようにします。VBAT 電源への切り替えは、リセットブロックに組み込まれているパワーダウンリセット回路によって制御されます。

---

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

---

外部バッテリーを使用しないアプリケーションでは、 $V_{BAT}$  を  $V_{DD}$  に外部接続し、デカップリング用の 100 nF のセラミックコンデンサを接続することを推奨します (詳細については、AN4206 を参照)。

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

- PC13、PC14、および PC15 は、GPIO ピンとして使用可能
- PC13、PC14、および PC15 は、RTC または LSE で設定可能 (セクション 27.3 : RTC の機能説明 (774 ページ) を参照)

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

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

- PC13、PC14、および PC15 は、RTC または LSE でのみ制御可能 (セクション 27.3 : RTC の機能説明 (774 ページ) を参照)

## 7.1.3 電圧レギュレータ

リセット後、電圧レギュレータは常に使用可能になります。レギュレータは、アプリケーションのモードに応じて、3 種類のモードで動作します。

- RUN モードでは、レギュレータは 1.8 V ドメイン (コア、メモリ、デジタルペリフェラル) にフル電力を供給します。
- STOP モードでは、レジスタと SRAM の内容を保持したまま、レギュレータは 1.8 V ドメインに低電力を供給します。
- STANDBY モードでは、レギュレータの電源がオフになります。STANDBY 回路と RTC ドメインを除き、レジスタと SRAM の内容は失われます。

STM32F3x8 デバイスでは、電圧レギュレータはバイパスされ、マイクロコントローラは公称電源電圧  $V_{DD} = 1.8 V \pm 8\%$  から電源供給されます。

## 7.2 電源供給スーパバイザ

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

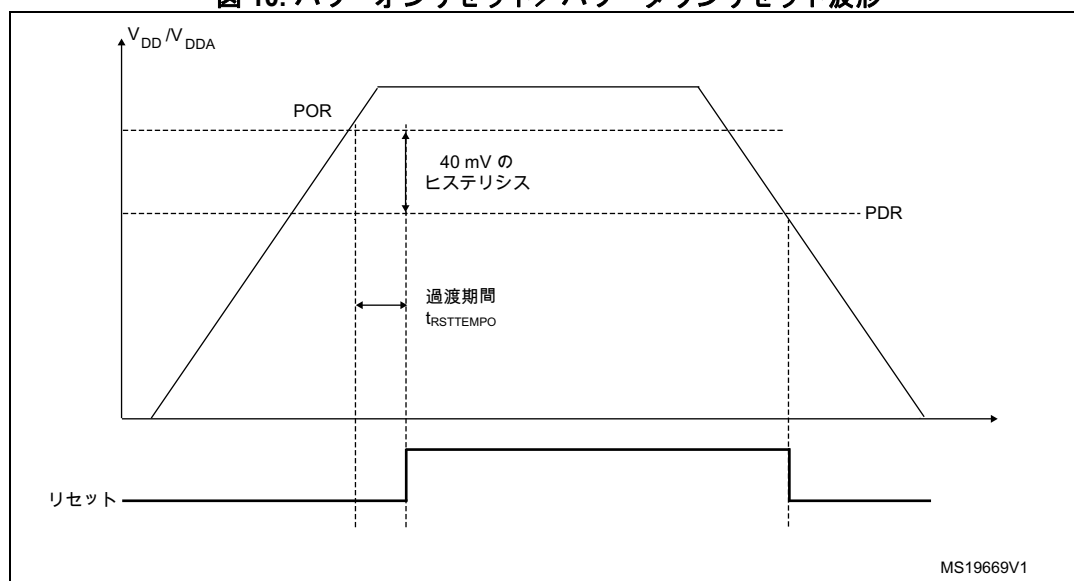
デバイスには、統合されたパワーオンリセット (POR) 回路とパワーダウンリセット (PDR) 回路があります。これらは常にアクティブで、2 V の閾値を上回る場合に正しい動作を保証します。

デバイスは、監視された電源電圧が規定の閾値  $V_{POR/PDR}$  を下回っている間は、外部のリセット回路を必要とせずに、リセットモードを維持します。

- POR では、 $V_{DD}$  電源電圧のみを監視します。起動フェーズ中、最初の入力力は  $V_{DDA}$  である必要があります。この値は  $V_{DD}$  以上である必要があります。
- PDR では、 $V_{DD}$  および  $V_{DDA}$  電源電圧を監視します。ただし、アプリケーションで  $V_{DDA}$  が  $V_{DD}$  以上になるように設計されている場合、消費電力を低減するために (専用の  $V_{DDA\_MONITOR}$  オプションビットをプログラムして)  $V_{DDA}$  電源供給スーパバイザを無効にすることができます。

POR/PDR 閾値の詳細については、データシートの電気的特性セクションの項を参照してください。

図 10. パワーオンリセット/パワーダウンリセット波形





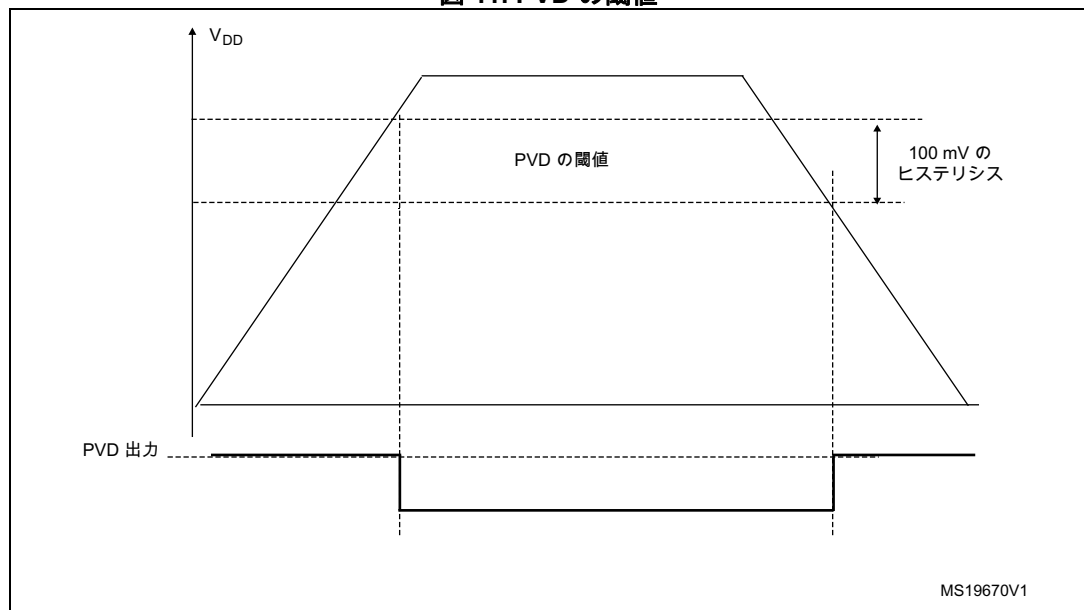
## 7.2.2 プログラム可能な電圧検出器 (PVD)

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

PVD は PVDE ビットをセットすることで有効になります。

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

図 11. PVD の閾値



注: STM32F3x8 デバイス ( $V_{DD} = 1.8 V \pm 8\%$ ) では、POR、PDR、および PVD 機能は使用できません。パワーオンリセット信号は NPOR ピンに印加されます。詳細については、次のセクションを参照してください。

## 7.2.3 外部 NPOR 信号

STM32F3x8 デバイスでは、PB2 の I/O は使用できません。パワーオンリセットで使用される NPOR 機能に置き換えられます。

パワーオンリセットを適切に行うために、 $V_{DD}$  の印加時に NPOR ピンをローにしておく必要があります。 $V_{DD}$  が安定している場合、リセット状態は次のいずれかの方法で終了できます。

- NPOR ピンをハイインピーダンスにします。NPOR ピンには、この入力を  $V_{DDA}$  に保持する内部プルアップがあります。
- または、プルアップレジスタを通じて外部的に  $V_{DDA}$  に接続し、ピンをハイレベルに保持します。

## 7.3 低電力モード

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

このデバイスは、次の 3 つの低電力モードを備えています。

- SLEEP モード (CPU クロックはオフ。NVIC、SysTick などの Cortex-M4<sup>®</sup>F コアペリフェラルを含むすべてのペリフェラルは動作状態を維持)
- STOP モード (すべてのクロックが停止)
- STANDBY モード (1.8 V ドメインの電源オフ)

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

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

表 16. 低電力モードの概要

モード名	エントリ	ウェイクアップ	1.8V ドメインクロックへの影響	V <sub>DD</sub> ドメインクロックへの影響	電圧レギュレータ
SLEEP (Sleep-now または Sleep-on- exit)	WFI	割り込み	CPU クロックオフ 他のクロックおよび アナログクロック ソースへの影響なし	なし	オン
	WFE	ウェイクアップ イベント			
STOP	PDDS ビット および LPDS ビット + SLEEPDEEP ビット + WFI または WFE	任意の EXTI ライン (EXTI レジスタで設 定) 受信イベントでの特 定の通信ペリフェラ ル (USART、I2C)	すべての 1.8V ドメ インクロックオフ	HSI および HSE オシレータ オフ	オンまたは低電力 モード ( <a href="#">電源制御レジスタ (PWR_CR)</a> に 従う)
STANDBY	PDDS ビット + SLEEPDEEP ビット + WFI または WFE	WKUP ピンの立ち 上がりエッジ、RTC アラーム、NRST ピン による外部リセット IWDG リセット			オフ

**注意：** レギュレータがオフの STM32F3x8 デバイスでは、STANDBY モードは使用できません。STOP モードは使用できます。ただし、レギュレータは使用されておらず、V<sub>DD</sub> はレギュレータ出力に外部から印加されているため、低電力モードの電圧レギュレータと RUN モードの電圧レギュレータの識別は意味を持ちません。

## 7.3.1 システムクロックの低速化

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

詳細については、[セクション 9.4.2 : クロック設定レジスタ \(RCC\\_CFGR\)](#) を参照してください。

## 7.3.2 ペリフェラルクロックゲーティング

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

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

ペリフェラルへのクロックゲーティングは、AHB ペリフェラルクロック有効レジスタ (RCC\_AHBENR)、APB1 ペリフェラルクロック有効レジスタ (RCC\_APB1ENR)、および APB2 ペリフェラルクロック有効レジスタ (RCC\_APB2ENR) によって制御されます。

## 7.3.3 SLEEP モード

### SLEEP モードへの移行

WFI (Wait For Interrupt) 命令または WFE (Wait for Event) 命令を実行することにより、SLEEP モードに移行します。SLEEP モードへの移行方法として、Cortex-M4<sup>®</sup>F システム制御レジスタの SLEEPONEXIT ビットによって、2 つのオプションが選択できます。

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

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

SLEEP モードへの移行方法の詳細については、[表 17](#) および [表 18](#) を参照してください。

### SLEEP モードの終了

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

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

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、Cortex-M4<sup>®</sup>F システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードで設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャネルペンディングビットをクリアする必要はありません。

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

SLEEP モードの終了方法の詳細については、表 17 および表 18 を参照してください。

表 17. Sleep-now

Sleep-now モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) - SLEEPDEEP = 0 および - SLEEPONEXIT = 0 Cortex-M4 <sup>®</sup> F システム制御レジスタを参照。
モードの終了	WFI 命令を使用して移行した場合： 割り込み：表 81：STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブルおよび表 82：STM32F303x6/8 および STM32F328x8 ベクタテーブルを参照してください。 WFE 命令を使用して移行した場合： ウェイクアップイベント：セクション 14.2.3：ウェイクアップイベント管理を参照してください。
ウェイクアップ遅延時間	なし

表 18. Sleep-on-exit

Sleep-on-exit	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) - SLEEPDEEP = 0 および - SLEEPONEXIT = 1 Cortex-M4 <sup>®</sup> F システム制御レジスタを参照。
モードの終了	割り込み：表 81：STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブルおよび表 82：STM32F303x6/8 および STM32F328x8 ベクタテーブルを参照。
ウェイクアップ遅延時間	なし

## 7.3.4 STOP モード

STOP モードは、ペリフェラルクロックゲーティングと組み合わせられた Cortex-M4<sup>®</sup>F のディープスリープ (deepsleep) モードに準拠しています。電圧レギュレータは、STM32F3xx デバイスで、通常モードまたは低電力モードのいずれかに設定できます。STM32F3x8 デバイスでは、レギュレータは使用されておらず、V<sub>DD</sub> はレギュレータ出力に外部から印加されているため、低電力モードの電圧レギュレータと RUN モードの電圧レギュレータの識別は意味を持ちません。STOP モードでは、1.8 V ドメインのすべてのクロックが停止し、PLL、HSI、HSE RC オシレータが停止します。SRAM とレジスタの内容は保持されます。

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

### STOP モードへの移行

STOP モードへの移行方法の詳細については、表 19 を参照してください。

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

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

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

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

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。参照: [セクション 25.3 : IWDG の機能説明](#)、[セクション 25 : 独立型ウォッチドッグ \(IWDG\)](#)
- リアルタイムクロック (RTC) : この設定は [RTC ドメイン制御レジスタ \(RCC\\_BDCR\)](#) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は [制御/ステータスレジスタ \(RCC\\_CSR\)](#) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は [RTC ドメイン制御レジスタ \(RCC\\_BDCR\)](#) の LSEON ビットで行います。

ADC や DAC は、STOP モードに移行する前に停止させない限り、STOP モード中でも電力を消費します。ADC を無効にするには、ADCx\_CR レジスタの ADDIS ビットをセットする必要があります。DAC を停止するには、DAC\_CR レジスタの ENx ビットに 0 を書き込む必要があります。

STOP モードの終了

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

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

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

表 19. STOP モード

STOP モード	説明
モードへの移行	<p>次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event)</p> <ul style="list-style-type: none"> <li>- Cortex-M4<sup>®</sup>F システム制御レジスタの SLEEPDEEP ビットをセット</li> <li>- 電源制御レジスタ (PWR_CR) の PDDS ビットをクリア</li> <li>- PWR_CR の LPDS ビットを設定することによって電圧レギュレータのモードを選択</li> </ul> <p><b>注:</b> STOP モードに移行するには、すべての EXTI ラインのペンディングビット (<a href="#">ペンディングレジスタ (EXTI_PR1)</a> 内)、すべてのペリフェラル割り込みのペンディングビット、および RTC アラームフラグをリセットする必要があります。そうしないと、STOP モード移行手順が無視され、プログラムが実行され続けます。</p> <p>STOP モードに移行する前にアプリケーションが外部オシレータ (外部クロック) を無効にする必要がある場合、最初にシステムクロックソースを HSI に切り替え、次いで HSEON ビットをクリアする必要があります。</p> <p>そうしないと、STOP モードに移行する前に HSEON ビットが 1 のままであった場合、STOP モードに移行するときに外部オシレータ (外部クロック) 障害を検出して誤動作を回避するべく、セキュリティシステム (CSS) 機能を有効にする必要があります。</p>
モードの終了	<p>WFI 命令を使用して移行した場合 :</p> <ul style="list-style-type: none"> <li>- 割り込みモードに設定されている任意の EXTI ライン (対応する EXTI 割り込みベクタが NVIC で有効になっている必要があります)。</li> <li>- ウェイクアップモードでプログラムした場合は、特定の通信ペリフェラル (USART、I2C) 割り込みの一部 (ペリフェラルはウェイクアップモードでプログラムする必要があり、対応する割り込みベクタが NVIC で有効になっている必要があります)。</li> </ul> <p><a href="#">表 81 : STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブル</a> および <a href="#">表 82 : STM32F303x6/8 および STM32F328x8 ベクタテーブル</a> を参照してください。</p> <p>WFE 命令を使用して移行した場合 :</p> <p>イベントモードに設定されている任意の EXTI ライン <a href="#">セクション 14.2.3 : ウェイクアップイベント管理 (291 ページ)</a> を参照してください。</p>
ウェイクアップ遅延時間	HSIRC ウェイクアップ時間 + 低電力モードからのレギュレータウェイクアップ時間

## 7.3.5 STANDBY モード

STANDBY モードでは、消費電力を最も少なくできます。このモードは、電圧レギュレータを無効にした状態の Cortex-M4<sup>®</sup>F のディープスリープ (deepsleep) モードに基づきます。結果として、1.8 V ドメインの電力がオフになります。PLL、HSI オシレータ、HSE オシレータもオフになります。RTC ドメインと STANDBY 回路のレジスタを除いて、SRAM とレジスタの内容は失われます ([図 8](#) を参照)。

**注意 :** *STM32F3x8 デバイスでは、STANDBY モードは使用できません。*

### STANDBY モードへの移行

STANDBY モードへの移行方法の詳細については、[表 20](#) を参照してください。

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

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リ

セット以外では停止できません。参照: [セクション 25.3: IWDG の機能説明](#)、[セクション 25: 独立型ウォッチドッグ \(IWDG\)](#)

- リアルタイムクロック (RTC): この設定は、RTC ドメイン制御レジスタ (RCC\_BDCR) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC): この設定は、制御/ステータスレジスタ (RCC\_CSR) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC): この設定は、RTC ドメイン制御レジスタ (RCC\_BDCR) の LSEON ビットで行います。

## STANDBY モードの終了

マイクロコントローラは、外部リセット (NRST ピン)、IWDG リセット、WKUP ピンの立ち上がりエッジ、または RTC アラームの立ち上がりエッジのいずれかが発生すると、STANDBY モードを終了します ([図 291: RTC ブロック図](#)を参照)。[電源制御/ステータスレジスタ \(PWR\\_CSR\)](#) を除くすべてのレジスタは、STANDBY モードからのウェイクアップ後にリセットされます。

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

STANDBY モードの終了方法の詳細については、[表 20](#) を参照してください。

表 20. STANDBY モード

STANDBY モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) - Cortex-M4 <sup>®</sup> F システム制御レジスタの SLEEPDEEP をセット - 電源制御レジスタ (PWR_CR) の PDDS ビットをセット - 電源制御/ステータスレジスタ (PWR_CSR) の WUF ビットをクリア
モードの終了	WKUP ピンの立ち上がりエッジ、RTC アラームイベントの立ち上がりエッジ、NRST ピンによる外部リセット、IWDG リセット
ウェイクアップ遅延時間	リセットフェーズ

## STANDBY モードにおける I/O の状態

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

- リセットパッド (有効)
- タンパまたは較正出力として設定されている場合の TAMPER ピン
- 有効な場合の WKUP ピン

## デバッグモード

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

ただし、DBGMCU\_CR レジスタの設定ビットをセットすることによって、低電力モードを多用しているときでも、ソフトウェアのデバッグを行うことができます。

## 7.3.6 低電力モードからの自動ウェイクアップ

RTC は、外部割り込み（自動ウェイクアップモード）に頼ることなく、低電力モードから MCU をウェイクアップするために使用できます。RTC を、一定の時間間隔で STOP モードや STANDBY モードからウェイクアップさせるためのプログラム可能なタイムベースとすることができます。この目的のため、*RTC ドメイン制御レジスタ (RCC\_BDCR)* の RTCSEL[1:0] ビットをプログラムすることによって、次の 3 つの代替 RTC クロックソースのうちから 2 つを選択できます。

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

RTC アラームイベントによって STOP モードからウェイクアップさせるには、次の設定が必要です。

- EXTI ライン 17 を立ち上がりエッジを検知するように設定します。
- RTC が RTC アラームを生成するように設定します。

STANDBY モードからウェイクアップするには、EXTI ライン 17 の設定は必要ありません。



## 7.4 電源制御レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

### 7.4.1 電源制御レジスタ (PWR\_CR)

アドレスオフセット : 0x00

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

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **DBP** : RTC ドメイン書き込み保護の無効化

リセット状態で、RTC およびバックアップレジスタは不要な書き込みアクセスから保護されます。これらのレジスタへの書き込みアクセスを可能にするには、このビットをセットする必要があります。

0 : RTC およびバックアップレジスタへのアクセスは無効です。

1 : RTC およびバックアップレジスタへのアクセスは有効です。

**注 :** 128 で分周した HSE が RTC クロックとして使用される場合、このビットは 1 にセットした状態を維持する必要があります。

ビット 7:5 **PLS[2:0]** : PVD レベル選択

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

000 : 2.2 V

001 : 2.3 V

010 : 2.4 V

011 : 2.5 V

100 : 2.6 V

101 : 2.7 V

110 : 2.8 V

111 : 2.9 V

注 :

1. 詳細については、データシートの電気的特性を参照してください。

2. PVD\_LOCK が有効化されると (CLASS B 保護のため)、PLS[2:0] ビットはプログラムできなくなります。

ビット 4 **PVDE** : 電源電圧検出器有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : PVD は無効です。

1 : PVD は有効です。

ビット 3 **CSBF** : STANDBY フラグのクリア

このビットは常に 0 として読み出されます。

0 : 影響なし。

1 : SBF STANDBY フラグのクリア (書き込み)

- ビット 2 **CWUF** : ウェイクアップフラグのクリア  
 このビットは常に 0 として読み出されます。  
 0 : 影響なし。  
 1 : WUF ウェイクアップフラグを 2 システムクロックサイクル後にクリアします (書き込み)。
- ビット 1 **PDDS** : パワーダウンディープスリープ  
 このビットは、ソフトウェアによってセット/クリアされます。LPDS ビットとの組み合わせで動作します。  
 0 : CPU がディープスリープに移行すると、STOP モードに移行します。レギュレータの状態は LPDS ビットによって変化します。  
 1 : CPU がディープスリープに移行すると、STANDBY モードに移行します。
- ビット 0 **LPDS** : 低電力ディープスリープ  
 このビットは、ソフトウェアによってセット/クリアされます。このビットは PDDS ビットとの組み合わせで動作します  
 0 : 電圧レギュレータは STOP モードの間もオン状態を保ちます。  
 1 : 電圧レギュレータは STOP モードの間は低電力モードになります。

## 7.4.2 電源制御/ステータスレジスタ (PWR\_CSR)

アドレスオフセット : 0x04

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	EWUP3 (1)	EWUP2	EWUP1	Res.	Res.	Res.	Res.	VREFINT RDYF	PVDO	SBF	WUF
					r/w	r/w	r/w					r	r	r	r

1. STM32F303x6/8 および STM32F328x8 では使用できません。

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **EWUP3** : WKUP3 ピンを有効化

このビットは、ソフトウェアによってセット/クリアされます。

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

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

**注 :** このビットは、システムリセットによってリセットされます。

ビット 9 **EWUP2** : WKUP2 ピンを有効化

このビットは、ソフトウェアによってセット/クリアされます。

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

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

**注 :** このビットは、システムリセットによってリセットされます。

## ビット 8 EWUP1 : WKUP1 ピンを有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : WKUP1 ピンは汎用 I/O として使用されます。この WKUP1 ピンでイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。

1 : WKUP1 ピンは STANDBY モードからのウェイクアップに使用され、強制的にプルダウン入力モードに設定されます (WKUP1 ピンの立ち上がりエッジで、システムは STANDBY モードからウェイクアップします)。

**注 :** このビットは、システムリセットによってリセットされます。

## ビット 7:4 予約済みであり、リセット値に保持する必要があります。

## ビット 3 VREFINTRDYF : V<sub>REFINT</sub> はレディ状態です。読み出し専用。このビットは、内部基準電圧の状態を示します。V<sub>REFINT</sub> がレディ状態になると、セットされます。V<sub>REFINT</sub> の安定化中にリセットされます。

**注 :** このフラグは、内部レギュレータをバイパスし、外部 NPOR ピンを使用する製品でのみ有効です。内部 POR は、リセットが解除される前に V<sub>REFINT</sub> が安定するまで待ちます。

## ビット 2 PVDO : PVD 出力

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

0 : V<sub>DD</sub>/V<sub>DDA</sub> は PLS[2:0] ビットで選択された PVD 閾値より高いです。

1 : V<sub>DD</sub>/V<sub>DDA</sub> は PLS[2:0] ビットで選択された PVD 閾値より低いです。

**注 :**

1. PVD は STANDBY モードによって停止します。したがって、このビットは STANDBY またはリセット後、PVDE ビットがセットされるまで 0 になります。
2. PVD が有効化され、PWR\_CR レジスタで設定されると、PVDO は外部割り込みコントローラを通じて割り込みを生成するために使用できます。
3. PVD\_LOCK が有効化されると (CLASS B 保護のため)、PVDO は無効にすることができなくなります。

## ビット 1 SBF : STANDBY フラグ

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

0 : デバイスは STANDBY モードではありません。

1 : デバイスは STANDBY モードです。

## ビット 0 WUF : ウェイクアップフラグ

このビットはハードウェアによってセットされ、システムリセットまたは電源制御レジスタ (PWR\_CR) に CWUF ビットをセットすることでクリアされます。

0 : ウェイクアップイベントは発生していません。

1 : WKUP ピンまたは RTC アラームからウェイクアップイベントを受信しました。

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

## 7.4.3 PWR レジスタマップ

次の表に PWR レジスタの一覧を示します。

表 21. PWR レジスタマップとリセット値

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

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 8 ペリフェラル相互接続マトリックス

### 8.1 概要

いくつかの STM32F3 ペリフェラルには内部相互接続があります。これらの相互接続を知っておくと、次のような利点があります。

- ペリフェラル間の自主的な通信
- ペリフェラル間の効率的な同期
- ソフトウェアの遅延をなくし、GPIO 設定を最小化
- 小さなパッケージでも使用可能な最適なピン数
- コネクタの使用を避け、余計なエネルギーを低減した最適な PCB を設計します。

### 8.2 接続の一覧

次の表に、ペリフェラル相互接続のマトリックスを示します。

表 22. STM32F3xx ペリフェラル相互接続マトリックス<sup>(1)</sup>

転送元	転送先																													
	DMA1	DMA2 <sup>(2)</sup>	ADC1	ADC2	ADC3 <sup>(2)</sup>	ADC4 <sup>(2)</sup>	COMP1 <sup>(2)</sup>	COMP2	COMP3 <sup>(2)</sup>	COMP4	COMP5 <sup>(2)</sup>	COMP6	COMP7 <sup>(2)</sup>	OPAMP1 <sup>(2)</sup>	OPAMP2	OPAMP3 <sup>(2)</sup>	OPAMP4 <sup>(2)</sup>	TIM1	TIM8	TIM15	TIM16	TIM17	TIM20	TIM2	TIM3	TIM4	DAC1	DAC2 <sup>(3)</sup>	IRTIM	
ADC1	X	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ADC2	X <sup>(3)</sup>	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
ADC3 <sup>(2)</sup>	-	X	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	X	-	-	-	-	-	-	-
ADC4 <sup>(2)</sup>	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-
COMP1 <sup>(2)</sup>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	X	X	X	-	-	-	-	-
COMP2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	X	X	X	-	-	-	-	-
COMP3 <sup>(2)</sup>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	X	X	X	-	-	-	-	-
COMP4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	X	-	X	X	-	-	-	-



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

表 22. STM32F3xx ペリフェラル相互接続マトリックス<sup>(1)</sup> (続き)

転送元	転送先																												
	DMA1	DMA2 <sup>(2)</sup>	ADC1	ADC2	ADC3 <sup>(2)</sup>	ADC4 <sup>(2)</sup>	COMP1 <sup>(2)</sup>	COMP2	COMP3 <sup>(2)</sup>	COMP4	COMP5 <sup>(2)</sup>	COMP6	COMP7 <sup>(2)</sup>	OPAMP1 <sup>(2)</sup>	OPAMP2	OPAMP3 <sup>(2)</sup>	OPAMP4 <sup>(2)</sup>	TIM1	TIM8	TIM15	TIM16	TIM17	TIM20	TIM2	TIM3	TIM4	DAC1	DAC2 <sup>(3)</sup>	IRTIM
COMP5 <sup>(2)</sup>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	X	X	X	X	X	-	-	-
COMP6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	X	X	-	X	-	-	-
COMP7 <sup>(2)</sup>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	X	X	X	-	-	-	-	-
OPAMP1 <sup>(2)</sup>	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OPAMP2	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OPAMP3 <sup>(2)</sup>	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
OPAMP4 <sup>(2)</sup>	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TIM1	X	-	X	X	X	X	X	X	X	-	-	-	X	X	X	X	X	-	X	-	-	-	X	X	X	X	-	-	-
SPI1	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TIM8	-	X	X	X	X	X	-	-	-	X	X	X	X	-	-	-	-	-	-	-	-	-	X	X	-	X	X	-	-
USART1	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPI4	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TIM15	X	-	X	X	X	X	-	-	-	X	X	X	X	-	-	-	-	X	-	-	-	-	X	-	X	-	X	X	-
TIM16	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	X
TIM17	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-	-	-	-	-	-	-	X

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

表 22. STM32F3xx ペリフェラル相互接続マトリックス<sup>(1)</sup> (続き)

転送元	転送先																												
	DMA1	DMA2 <sup>(2)</sup>	ADC1	ADC2	ADC3 <sup>(2)</sup>	ADC4 <sup>(2)</sup>	COMP1 <sup>(2)</sup>	COMP2	COMP3 <sup>(2)</sup>	COMP4	COMP5 <sup>(2)</sup>	COMP6	COMP7 <sup>(2)</sup>	OPAMP1 <sup>(2)</sup>	OPAMP2	OPAMP3 <sup>(2)</sup>	OPAMP4 <sup>(2)</sup>	TIM1	TIM8	TIM15	TIM16	TIM17	TIM20	TIM2	TIM3	TIM4	DAC1	DAC2 <sup>(3)</sup>	IRTIM
TIM20	-	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TIM2	X	-	X	X	X	X	X	X	X	-	X	X	-	-	-	-	-	X	X	X	-	-	-	-	X	X	X	X	-
TIM3	X	-	X	X	X	X	X	X	-	X	-	-	-	-	-	-	-	X	X	X	-	-	-	X	-	X	X	X	-
TIM4	X	-	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	-	X	X	X	-	X	-	-
TIM6	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	
TIM7	X	X	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	
SPI2/I2S	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SPI3/I2S	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
USART2	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
USART3	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UART4	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UART5	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I2C1	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I2C2	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DAC1	X	X	-	-	-	-	X	X	X	X	X	X	X	X	-	X	X	-	-	-	-	-	-	-	-	-	-	-	-
DAC2 (1)	X	-	-	-	-	-	-	X	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I2C3	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TS	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



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

表 22. STM32F3xx ペリフェラル相互接続マトリックス<sup>(1)</sup> (続き)

転送元	転送先																												
	DMA1	DMA2 <sup>(2)</sup>	ADC1	ADC2	ADC3 <sup>(2)</sup>	ADC4 <sup>(2)</sup>	COMP1 <sup>(2)</sup>	COMP2	COMP3 <sup>(2)</sup>	COMP4	COMP5 <sup>(2)</sup>	COMP6	COMP7 <sup>(2)</sup>	OPAMP1 <sup>(2)</sup>	OPAMP2	OPAMP3 <sup>(2)</sup>	OPAMP4 <sup>(2)</sup>	TIM1	TIM8	TIM15	TIM16	TIM17	TIM20	TIM2	TIM3	TIM4	DAC1	DAC2 <sup>(3)</sup>	IRTIM
VBAT	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vrefint	-	-	X	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	X	-	-	-	-	-	-
PVD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	X	-	-	-	-	-	-
SRAM パリティエラー	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	X	-	-	-	-	-	-
CPU ハードフォルト	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	-	-	X	-	-	-	-	-	-
HSE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
HSI	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
LSE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
LSI	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
MCO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-
RTC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-

- 「X」は相互接続があることを、「-」は相互接続がないことを意味しています。
- STM32F303x6/8 および STM32F328x8 は除きます。
- STM32F303x6/8 および STM32F328x8 のみ。



## 8.3 相互接続の詳細

### 8.3.1 DMA 相互接続

ハードウェア DMA リクエストはペリフェラルで管理されます。各ペリフェラル専用の DMA チャンネルの一覧を [セクション 13.4.7 : DMA リクエストマッピング](#) に示します。

### 8.3.2 ADC から ADC

ADC1 は ADC2 の変換の「スレーブ」開始をトリガする「マスタ」として使用できます。

ADC3 は ADC4 の変換の「スレーブ」開始をトリガする「マスタ」として使用できます。

デュアル ADC モードでは、マスタおよびスレーブ ADC の変換されたデータを並列で読み出すことができます。

デュアル ADC モードの説明については、[セクション 15.3.29 : デュアル ADC モード](#) を参照してください。

### 8.3.3 ADC から TIM

ADCx (x=1..4) では、ウォッチドッグ信号を通じて高機能制御タイマ (TIM1/TIM8/TIM20) にトリガイベントを提供できます。

ADC アナログウォッチドッグ設定の説明については、[セクション 15.3.28 : アナログウィンドウウォッチドッグ \(AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD\\_HTx, AWD\\_LTx, AWDx\)](#) を参照してください。

出力 (ADC から) は信号 ADCn\_AWDx\_OUT (ADC ごとにアナログウォッチドッグが 3 つあるため n = 1..4, x = 1..3) で行われ、入力 (タイマへ) は信号 TIMx\_ETR (外部トリガ) で行われます。

TIMx\_ETR は TIMx\_OR レジスタのビットを通じて ADCn\_AWDx\_OUT に接続されています。[セクション 20.4.21 : TIM1/TIM8/TIM20 オプションレジスタ \(TIMx\\_OR\)](#) を参照してください。

表 23. ADCx アナログウォッチドッグへの TIM1/8/20\_ETR の接続

	TIM1	TIM8	TIM20
ADC1	x	-	-
ADC2	-	x	-
ADC3	-	x	x
ADC4	x	-	x

### 8.3.4 TIM および EXTI から ADC

汎用タイマ (TIM2/TIM3/TIM4)、基本タイマ (TIM6/TIM7)、高機能制御タイマ (TIM1/TIM8/TIM20)、汎用タイマ (TIM15/TIM16/TIM17)、および EXTI は、ADC トリガイベントを生成するために使用できます。

出力 (タイマから) は、信号 TIMx\_TRGO、TIMx\_TRGO2、または TIMx\_CCx イベントで行われます。

入力 (ADC へ) は信号 EXT[15:0]、JEXT[15:0] で行われます。

タイマと ADC の間、または EXTI & ADC との接続については、次を参照してください。

- [表 89 : ADC1 \(マスタ\) および 2 \(スレーブ\) - レギュラチャネルの外部トリガ](#)
- [表 90 : ADC1 および ADC2 - インジェクトチャネルの外部トリガ](#)
- [表 91 : ADC3 および ADC4 - レギュラチャネルの外部トリガ](#)
- [表 92 : ADC3 および ADC4 - インジェクトチャネルの外部トリガ](#)

## 8.3.5 OPAMP から ADC

この場合、次の 2 つの相互接続のタイプがあります。

1. 内部 ADC チャンネルに OPAMP 出力基準電圧を接続します。この接続は OPAMP 較正用として使用できます。詳細については、OPAMP の章のセクション [較正](#) を参照してください。

[セクション 15.3.11 : チャンネルの選択 \(SQRx, JSQRx\)](#) これで、使用する実際の ADC チャンネルを提供できます。

**表 24. VREFOPAMPx から ADC チャンネル**

VREFOPAMPx	ADC チャンネル
VREFOPAMP1	ADC1_IN15
VREFOPAMP2	ADC2_IN17
VREFOPAMP3	ADC3_IN17
VREFOPAMP4	ADC4_IN17

2. OPAMPx 出力 (x = 1..4) は GPIO を通じて ADCy チャンネル (y = 1..4) に接続することができます。次の表の一覧を参照してください。 [セクション 18.3.4 : ADC 入力としての OPAMP 出力の使用](#) を参照してください。

**表 25. ADC 入力への OPAMP 出力**

OPAMPx 出力	ADC チャンネル	使用されるピン
OPAMP1_VOUT	ADC1_IN3	PA2
OPAMP2_VOUT	ADC2_IN3	PA6
OPAMP3_VOUT	ADC3_IN1	PB1
OPAMP4_VOUT	ADC4_IN3	PB12

## 8.3.6 TS から ADC

内部温度センサ (VTS) は、ADC1\_IN16 に内部接続されます。 [セクション 15.3.30 : 温度センサ](#) を参照してください。

## 8.3.7 VBAT から ADC

VBAT/2 出力電圧は、ADC1\_IN17 を使用して変換できます。この相互接続を、 [セクション 15.3.31 : VBAT 電源監視](#) に示します。

## 8.3.8 VREFINT から ADC

VREFINT は、4 つの ADC のチャンネル 18 に内部的に接続されます。これにより、 [セクション 15.3.32 : 内部電圧基準の監視](#) で説明しているように、値を監視することができます。

## 8.3.9 COMP から TIM

コンパレータの出力を異なるタイマ入力に内部的に変更することができます。

- 高速 PWM 停止のブレイク入力 1/2
- OCREF\_CLR 入力
- 入力キャプチャ

コンパレータの出力に接続する必要があるタイマ入力を選択するには、COMPx\_CSR レジスタのビットフィールド COMPxOUTSEL を使用します。

次の表に、タイマ入力に変更可能なすべてのコンパレータの出力の概要を示します。

表 26. タイマ入力へのコンパレータの出力

COMP 出力選択									
	TIM1	TIM8	TIM2	TIM3	TIM4	TIM15	TIM16	TIM17	TIM20
COMP1	TIM1_BRK_ACTH TIM1_BRK2 TIM1_OCrefClear TIM1_IC1	TIM8_BRK_ACTH TIM8_BRK2	TIM2_IC4 TIM2_OCrefClear	TIM3_IC1 TIM3_OCrefClear	N/A	N/A	N/A	N/A	TIM20_BRK_ACTH TIM20_BRK2
COMP2	TIM1_BRK_ACTH TIM1_BRK2 TIM1_OCrefClear TIM1_IC1	TIM8_BRK_ACTH TIM8_BRK2	TIM2_IC4 TIM2_OCrefClear	TIM3_IC1 TIM3_OCrefClear	N/A	N/A	N/A	N/A	TIM20_BRK_ACTH TIM20_BRK2 TIM20_OCrefClear
COMP3	TIM1_BRK_ACTH TIM1_BRK2 TIM1_OCrefClear	TIM8_BRK_ACTH TIM8_BRK2	TIM2_OCrefClear	TIM3_IC2	TIM4_IC1	TIM15_IC1 TIM15_BRK_ACTH	N/A	N/A	TIM20_BRK_ACTH TIM20_BRK2
COMP4	TIM1_BRK TIM1_BRK2	TIM8_BRK TIM8_BRK2 TIM8_OCrefClear	N/A	TIM3_IC3 TIM3_OCrefClear	TIM4_IC2	TIM15_OCrefClear TIM15_IC2	N/A	N/A	TIM20_BRK TIM20_BRK2
COMP5	TIM1_BRK_ACTH TIM1_BRK2	TIM8_BRK_ACTH TIM8_BRK2 TIM8_OCrefClear	TIM2_IC1	TIM3_OCrefClear	TIM4_IC3	N/A	TIM16_BRK_ACTH	TIM17_IC1	TIM20_BRK_ACTH TIM20_BRK2

表 26. タイマ入力へのコンパレータの出力 (続き)

COMP 出力選択									
	TIM1	TIM8	TIM2	TIM3	TIM4	TIM15	TIM16	TIM17	TIM20
COMP6	TIM1_BRK_ACTH TIM1_BRK2	TIM8_BRK_ACTH TIM8_BRK2 TIM8_OCrefClear	TIM2_IC2 TIM2_OCrefClear	N/A	TIM4_IC4	N/A	TIM16_OCrefClear TIM16_IC1	N/A	TIM20_BRK_ACTH TIM20_BRK2
COMP7	TIM1_BRK TIM1_BRK2 TIM1_OCrefClear TIM1_IC2	TIM8_BRK TIM8_BRK2 TIM8_OCrefClear	TIM2_IC3	N/A	N/A	N/A	N/A	TIM17_OCrefClear TIM17_BRK_ACTH	TIM20_BRK TIM20_BRK2

注: コンパレータの出力がタイマブレイク入力に内部接続するように設定されている場合、次の事を考慮する必要があります。

1/ COMP1/2/3/5/6 は、TIM1/8/20\_BRK\_ACTH (このブレイクはデジタルフィルタなしで常にアクティブハイ) に加え、TIM1/8/20\_BRK2 入力を制御するためにも使用できます。

2/ COMP4/7 は TIM1/8/20\_BRK および TIM1/8/20\_BRK2 入力 (他のコンパレータと同じ) を制御するために使用できます。

3/ COMP3/5/7 は、TIMx\_BRK\_ACTH (x はそれぞれ 15、16、17) を制御するために使用できます (このブレイクはデジタルフィルタなしで常にアクティブハイ)。

### 8.3.10 TIM から COMP

タイマ出力は、「COMPx\_CSR」レジスタの「COMPx\_BLANKING」ビットを使用して、コンパレータの出力のブランキング信号として選択できます。ブランキング機能の詳細については、[セクション 17.3.6: コンパレータの出力のブランキング機能](#)を参照してください。

表 27. コンパレータのブランキングソースとしてのタイマ出力選択

COMP のブランキングソース							
	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
TIM1	TIM1 OC5	TIM1 OC5	TIM1 OC5	-	-	-	TIM1 OC5
TIM8	-	-	-	TIM8 OC5	TIM8 OC5	TIM8 OC5	TIM8 OC5
TIM15	-	-	-	TIM15 OC1	-	TIM15 OC2	TIM15 OC2
TIM2	TIM2 OC3	TIM2 OC3	TIM2 OC4	-	-	TIM2 OC4	-
TIM3	TIM3 OC3	TIM3 OC3	-	TIM3 OC4	TIM3 OC3	-	-

## 8.3.11 DAC から COMP

コンパレータの反転入力、DAC チャネル出力である場合があります (DAC1\_CH1 または DAC1\_CH2)。デバイスが STM32F303x6/8 または STM32F328x8 である場合に、COMP2、COMP4、および COMP6 に DAC2\_CH1 を選択できます。

この選択は、「COMPx\_CSR」レジスタの「COMPxINMSEL」ビットに基づいて行われます。

次の表に、相互接続の一覧を示します。

表 28. コンパレータの反転入力としての DAC 出力選択

COMP 反転入力							
	COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
DAC1_CH1	X	X	X	X	X	X	X
DAC1_CH2	X	X	X	X	X	X	X
DAC2_CH1 <sup>(1)</sup>		X		X		X	

1. STM32F303x6/8 および STM32F328x8 のみ。

## 8.3.12 VREFINT から COMP

DAC チャネル出力の他に、「COMPx\_CSR」レジスタの「COMPxINMSEL」ビットを使用して、Vrefint (x1、x3/4、x1/2、x1/4) をコンパレータの反転入力として選択できます。

## 8.3.13 DAC から OPAMP

次の一覧表に示すように、DAC 出力は、OPAMP1 & OPAMP3 & OPAMP4 の非反転入力に内部接続しています。

表 29. OPAMP の非反転入力としての DAC 出力選択

	非反転入力		
	OPAMP1	OPAMP3	OPAMP4
DAC チャネル	DAC1_CH2	DAC1_CH2	DAC1_CH1

## 8.3.14 TIM から OPAMP

OPAMP の反転および非反転入力間の切り替えは、自動的に行うことができます。この自動切り替えは、OPAMP の入力マルチプレクサで発生する TIM1 CC6 出力によってトリガされます。この機能の詳細については、[セクション 18.3.6: タイマによって制御されたマルチプレクサモード](#)を参照してください。

## 8.3.15 TIM から TIM

タイマの同期や連携した動作のために、いくつかの STM32F3 タイマを内部で相互リンクすることができます。

マスターモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

さまざまな同期モードを持つ機能の説明については、次を参照してください。

- 高機能制御タイマ (TIM1/TIM8/TIM20) について: [セクション 20.3.25 : タイマの同期](#)
- 汎用タイマ (TIM2/TIM3/TIM4) について: [セクション 21.3.19 : タイマの同期](#)

スレーブモード選択には「SMS」ビットを使用します。次を参照してください。

- [セクション 20.4.3 : TIM1/TIM8/TIM20 スレーブモード制御レジスタ \(TIMx\\_SMCR\)](#)
- 汎用タイマ (TIM2/TIM3/TIM4) について: [セクション 21.4.3 : TIMx スレーブモード制御レジスタ \(TIMx\\_SMCR\)](#)
- 汎用タイマ (TIM2/TIM3/TIM4) について: [セクション 23.4.18 : スレーブモード: リセットモードとトリガモードの組み合わせ \(TIM15 のみ\)](#)

使用可能なマスタ/スレーブ接続を、内部トリガ接続を提供する次の一覧表に示します。

表 30. タイマの同期

		スレーブ						
		TIM1	TIM8	TIM20	TIM2	TIM3	TIM4	TIM15
マスタ	TIM1	-	TIM8_ITR0	TIM20_ITR0	TIM2_ITR0	TIM3_ITR0	TIM4_ITR0	-
	TIM8	-	-	TIM20_ITR1	TIM2_ITR1	-	TIM4_ITR3	-
	TIM2	TIM1_ITR1	TIM8_ITR1	-	-	TIM3_ITR1	TIM4_ITR1	TIM15_ITR0
	TIM3	TIM1_ITR2	TIM8_ITR3	-	TIM2_ITR2	-	TIM4_ITR2	TIM15_ITR1
	TIM4	TIM1_ITR3	TIM8_ITR2	TIM20_ITR2	TIM2_ITR3	TIM3_ITR3	-	-
	TIM15	TIM1_ITR0	-	TIM20_ITR3	-	TIM3_ITR2	-	-
	TIM16	-	-	-	-	-	-	TIM15_ITR2
	TIM17	TIM1_ITR3	-	-	-	-	-	TIM15_ITR3

## 8.3.16 ブレーク入力ソースから TIM

コンパレータの出力に加え、他のソースを一部のタイマ (TIM1/TIM8/TIM20/TIM15/TIM16/TIM17) の内部ブレークイベントのトリガとして使用できます。例：

- CSS によって生成されるクロック障害イベント (詳細については、[セクション 9.2.6: システムクロック \(SYSCLK\) の選択](#)を参照)
- PVD 出力 (詳細については、[セクション 7.2.2: プログラム可能な電圧検出器 \(PVD\)](#)を参照)
- SRAM パリティエラー信号 (詳細については、[セクション 3.3.1: パリティチェック](#)を参照)
- Cortex-M4 LOCKUP (ハードフォルト) 出力

上記のソースは、TIMx\_BRK\_ACTH 入力に内部接続することができます。  
x = 1、8、15、16、17、20。

ブレーク機能の目的は、タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。

この機能の詳細については、次を参照してください。

- 高機能制御タイマ (TIM1/TIM8/TIM20) について：[セクション 20.3.16: ブレーク機能の使用](#)
- 汎用タイマ (TIM15/TIM16/TIM17) について：[セクション 23.4.13: ブレーク機能の使用](#)

## 8.3.17 HSE、HSI、LSE、LSI、MCO、RTC から TIM

TIM16 は、内部/外部クロックソースの測定に使用できます。TIM16 チャンネル 1 入力キャプチャは、HSE/32、GPIO、RTC クロックに接続され、MCO (HSE、HSI、LSE、LSI、SYSCLK、PLLCLK、PLLCLK/2) の出力クロックに接続されます。

この選択は、TIM16\_OR レジスタの TI1\_RMP [1:0] ビットで行います。

これにより、HSI/LSI クロックを校正できます。

詳細については、[セクション 9.2.14: TIM16 を使用した内部/外部クロックの測定](#)を参照してください。

## 8.3.18 TIM および EXTI から DAC

タイマカウンタは DAC 変換のトリガとして使用できます。

TRGO イベントは変換をトリガする内部信号です。

DAC とタイマとの相互接続を次の一覧表に示します。

これについては、[セクション 16.5.4 : DAC トリガ選択](#)で説明されています。

表 31. DAC 変換をトリガするタイマおよび EXTI 信号

	DAC1	DAC2 <sup>(1)</sup>
TIM8	X	-
TIM2	X	X
TIM3	X	X
TIM4	X	-
TIM6	X	X
TIM7	X	X
TIM15	X	X
EXTI ライン 9	X	X

1. STM32F303x6/8 および STM32F328x8 デバイスのみ。

## 8.3.19 TIM から IRTIM

汎用タイマ (TIM16/TIM17) の出力チャネル TIMx\_OC1 は、赤外線信号出力の波形を生成するために使用されます。この機能については、[セクション 24 : 赤外線インタフェース \(IRTIM\)](#)で説明されています。



## 9 リセットおよびクロック制御 (RCC)

### 9.1 リセット

リセットには、システムリセット、電源リセット、RTC ドメインリセットの 3 種類があります。

#### 9.1.1 電源リセット

電源リセットは、次のいずれかのイベントが発生したときに生成されます。

1. パワーオン/パワーダウンリセット (POR/PDR リセット)
2. STANDBY モードの終了

電源リセットは、RTC ドメインを除くすべてのレジスタをリセット値にセットします ([図 8 \(92 ページ\)](#) を参照)。

#### 9.1.2 システムリセット

システムリセットは、クロックコントローラ CSR レジスタのリセットフラグと RTC ドメインのレジスタを除くすべてのレジスタをリセット値にセットします ([図 8 \(92 ページ\)](#) を参照)。

システムリセットは、次のイベントのいずれかの発生時に生成されます。

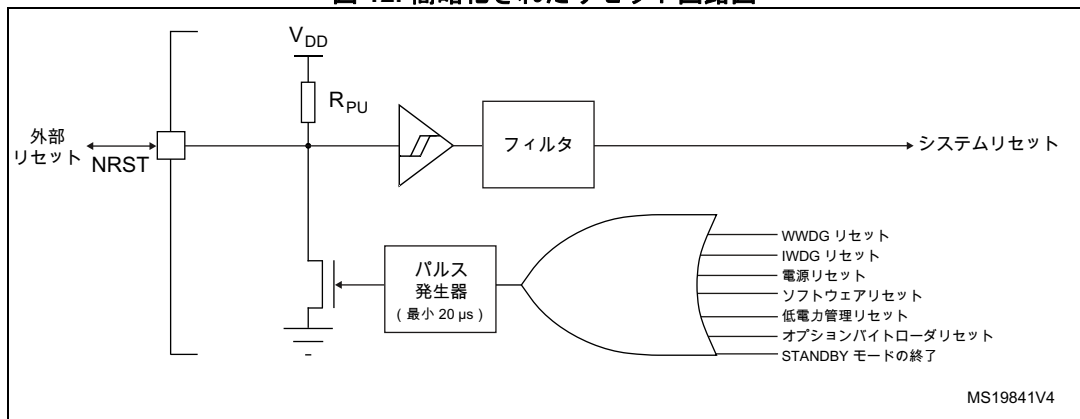
1. NRST ピンのローレベル (外部リセット)
2. ウィンドウ型ウォッチドッグイベント (WWDG リセット)
3. 独立型ウォッチドッグイベント (IWDG リセット)
4. ソフトウェアリセット (SW リセット) ([ソフトウェアリセット](#)を参照)
5. 低電力管理リセット ([低電力管理リセット](#)を参照)
6. オプションバイトローダリセット ([オプションバイトローダリセット](#)を参照)
7. 電源リセット

リセットソースは、制御/ステータスレジスタ (RCC\_CSR) のリセットフラグを確認することによって識別できます ([セクション 9.4.10 : 制御/ステータスレジスタ \(RCC\\_CSR\)](#) を参照)。

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

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

図 12. 簡略化されたリセット回路図



## ソフトウェアリセット

デバイス上でソフトウェアリセットを実行するには、Cortex-M4<sup>®</sup>F のアプリケーション割り込みおよびリセット制御レジスタの SYSRESETREQ ビットをセットする必要があります。詳細については、STM32F3xx/F4xx Cortex<sup>®</sup>-M4 プログラミングマニュアル (PM0214) を参照してください。

## 低電力管理リセット

低電力管理リセットを生成するには、2つの方法があります。

- STANDBY モードに移行するときにリセットを生成します。  
この種類のリセットは、ユーザオプションバイトの nRST\_STDBY ビットをリセットすることで有効になります。この場合、STANDBY モードへの遷移シーケンスが正常に実行されるたびに、STANDBY モードに移行する代わりにデバイスがリセットされます。
- STOP モードに移行するときにリセットを生成します：  
この種類のリセットは、ユーザオプションバイトの nRST\_STOP ビットをリセットすることで有効になります。この場合、STOP モードへの遷移シーケンスが正常に実行されるたびに、STOP モードに移行する代わりにデバイスがリセットされます。

ユーザオプションバイトの詳細については、[Section 4: Option byte](#)を参照してください。

## オプションバイトローダリセット

オプションバイトローダリセットは、FLASH\_CR レジスタの OBL\_LAUNCH ビット (ビット 13) がセットされると生成されます。このビットは、ソフトウェアによるオプションバイトローディングを起動するために使用されます。

### 9.1.3 RTC ドメインリセット

RTC ドメインには、RTC ドメインだけに影響する 2 種類の特定制のリセットがあります ([図 8 \(92 ページ\)](#))。

RTC ドメインリセットは、LSE オシレータ、RTC、バックアップレジスタ、および RCC [RTC ドメイン制御レジスタ \(RCC\\_BDCR\)](#) にも影響します。これは、次のいずれかのイベントが発生したときに生成されます。

- [RTC ドメイン制御レジスタ \(RCC\\_BDCR\)](#) の BDRST ビットをセットすることでトリガされるソフトウェアリセット
- V<sub>BAT</sub> がローのときに切断されると、V<sub>DD</sub> がパワーアップします。

バックアップレジスタは、次のいずれかのイベントが発生したときにリセットされます。

1. RTC タンパ検出イベント
2. 読み出し保護のレベル 1 からレベル 0 への変更

## 9.2 クロック

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

- HSI 8 MHz RC オシレータクロック
- HSE オシレータクロック
- PLL クロック

デバイスには、次の追加のクロックソースがあります。

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

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

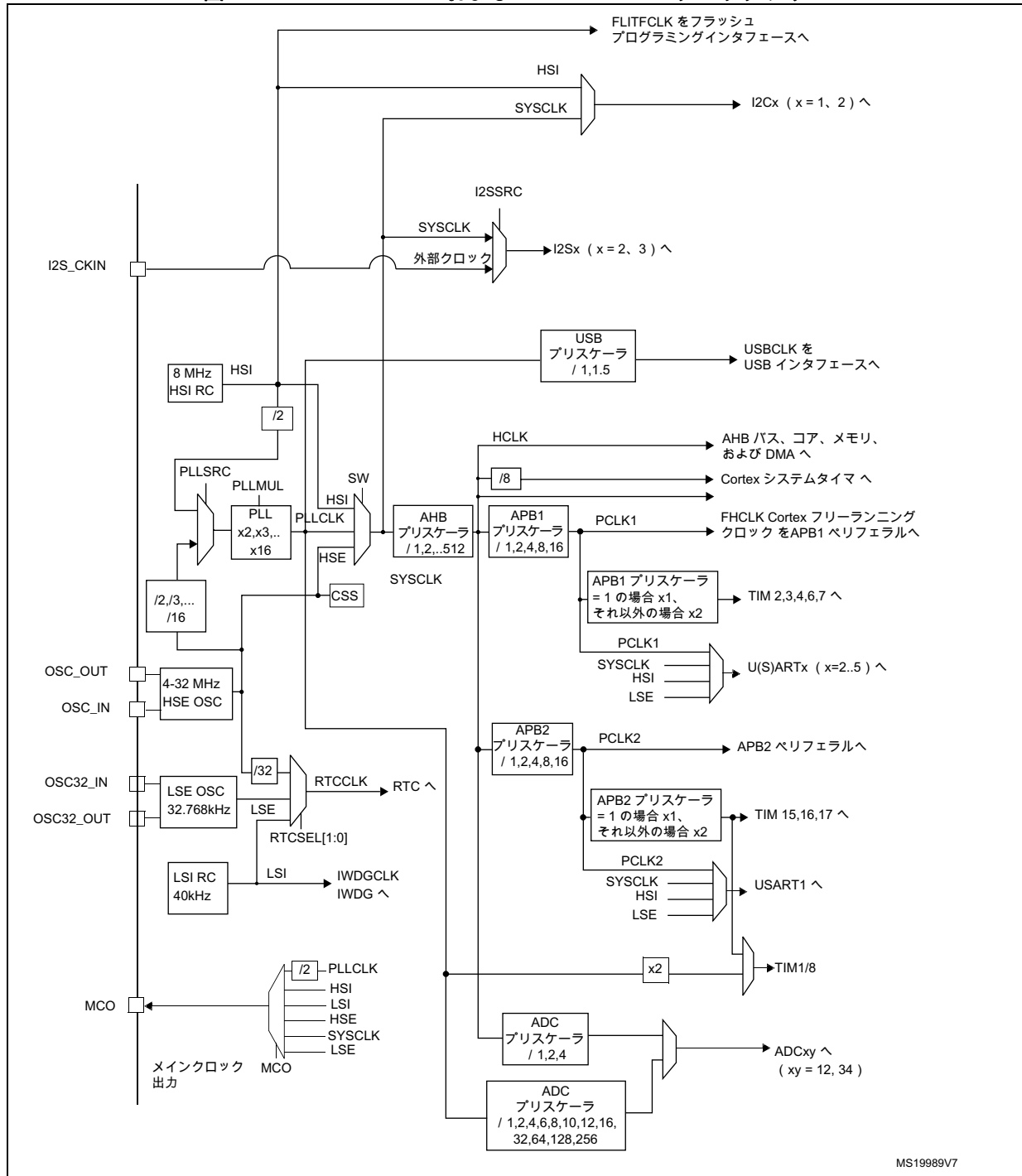
AHB 周波数、高速 APB (APB2)、および低速 APB (APB1) ドメインの設定に、複数のプリスケアラを使用できます。AHB および APB2 ドメインの最大周波数は 72 MHz です。APB1 ドメインの最大許容周波数は 36 MHz です。

すべてのペリフェラルクロックは、次を除いてそれらのバスクロックから生成されます (HCLK、PCLK1、または PCLK2)。

- 常に HSI クロックであるフラッシュメモリのプログラミングインタフェースクロック (FLITFCLK)
- PLL VCO (STM32F303xB/C/D/E デバイス) から生成された 48 MHz USB クロック
- 常に HSI クロックであるオプションバイトローダクロック
- PLL 出力から生成された ADC クロック。72 MHz に達することができ、その後 1、2、4、6、8、10、12、16、32、64、128、または 256 で分周できます。
- 次の 4 つのうちのいずれかのソースから生成される U(S)ART クロック (ソフトウェアによって選択される):
  - システムクロック
  - HSI クロック
  - LSE クロック
  - APB1 または APB2 クロック (USART に配置されている APB に応じて、PCLK1 または PCLK2)
- 次の 2 つのうちのいずれかのソースから生成される (ソフトウェアによって選択される) I2C1/2 (STM32F303xD/E および STM32F398xE の I2C1/2/3) クロック:
  - システムクロック
  - HSI クロック
- 外部専用クロックソースから生成できる I2S2 と I2S3 クロック
- LSE、LSI、または 32 分周された HSE クロックから生成される RTC クロック
- IWDG クロックは、常に LSI クロックから生成されます。

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

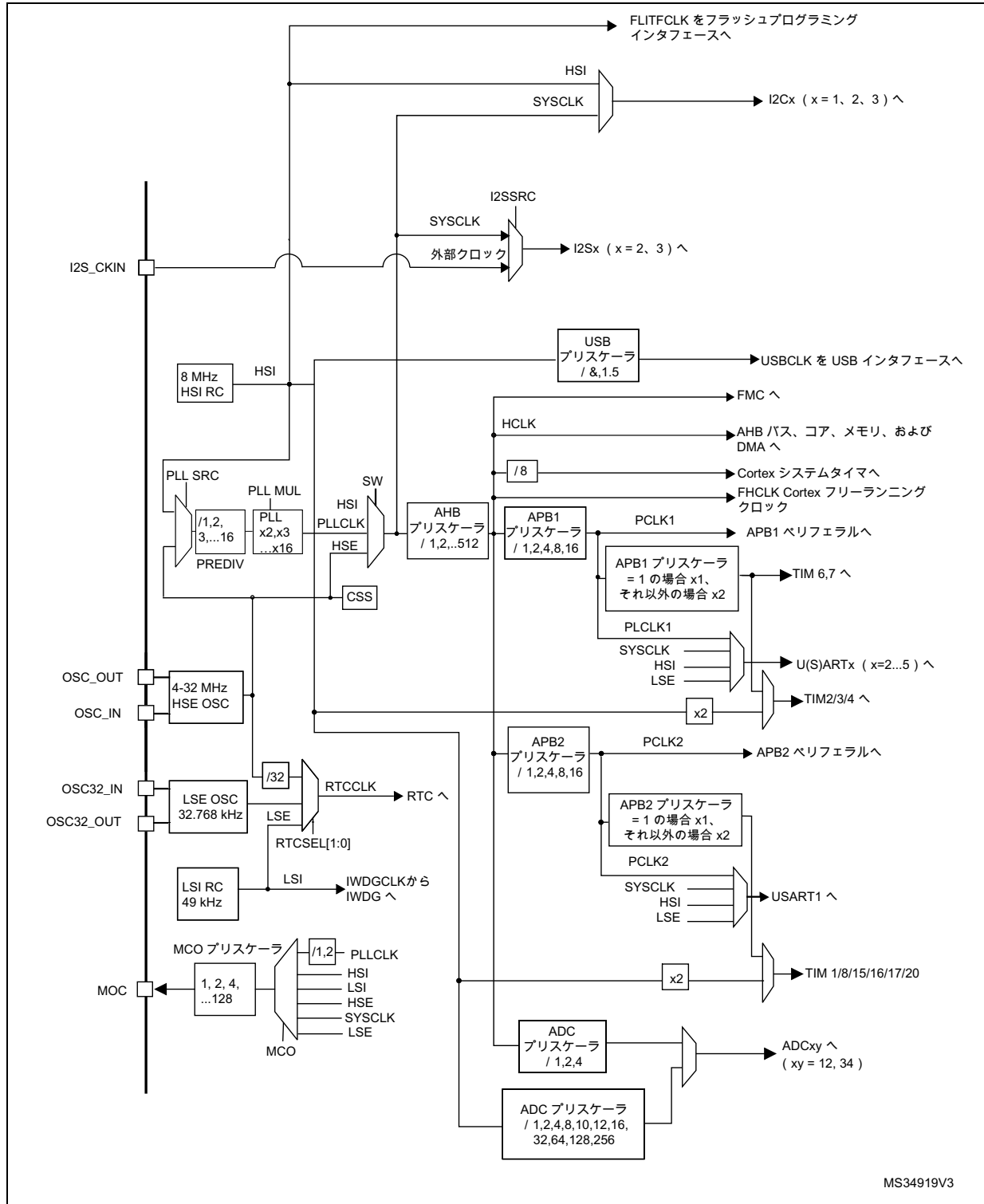
図 13. STM32F303xB/C および STM32F358xC クロックツリー



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

- TIM1 および TIM8 は、システムクロックソースが PLL で、AHB および APB2 プリスケーラが「1」にセットされている場合に、最大 144 MHz まで動作する PLLCLKx2 からクロック供給を受けられます。
- ADC クロックは、ADC バスインタフェースの AHB クロックから生成し、プログラム可能な係数 (1、2、または 4) で分周することができます。プログラム可能な係数が「1」の場合、AHB プリスケーラは「1」でなければなりません。

図 14. STM32F303xDxE および STM32F398xE のクロックツリー



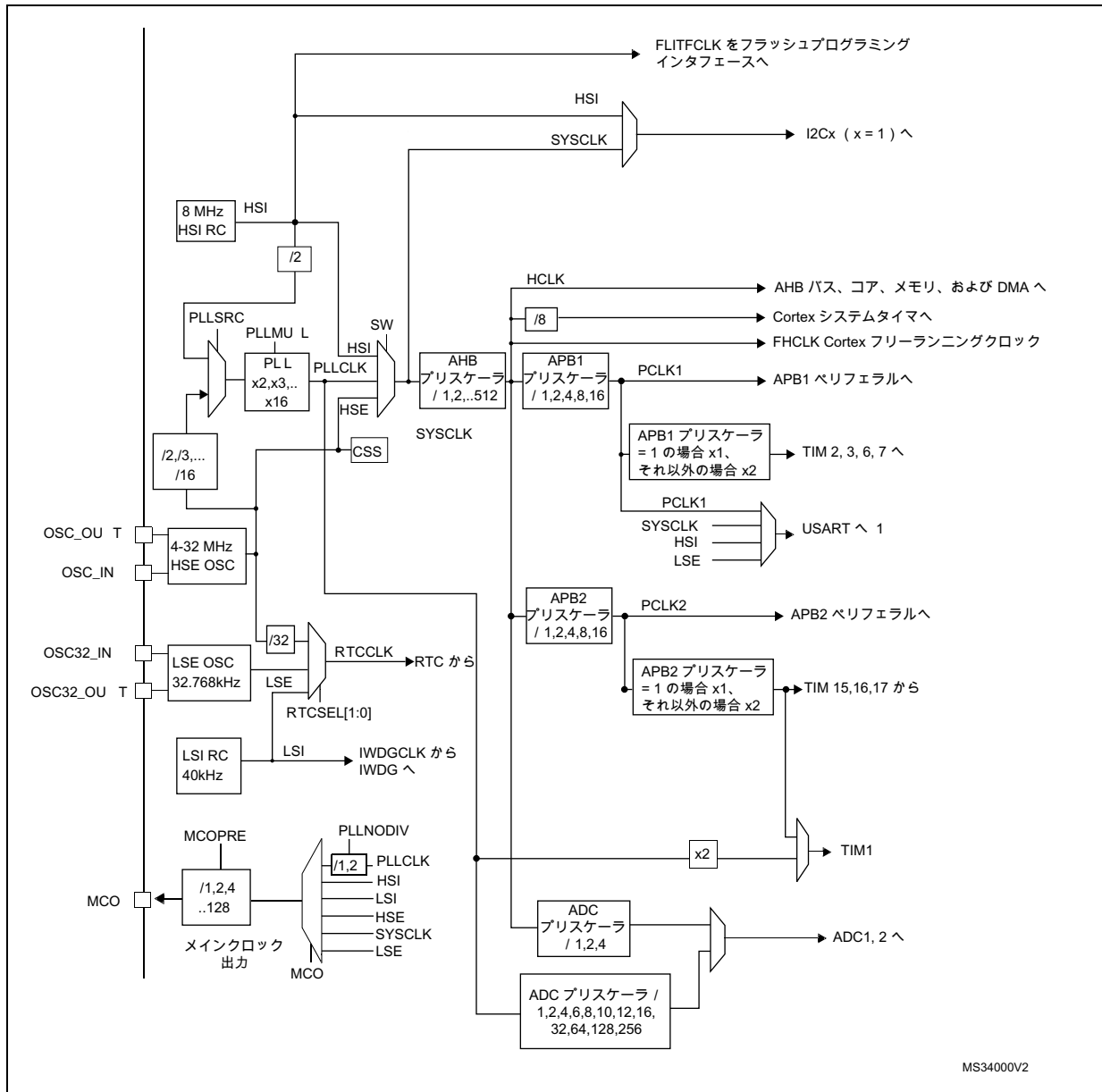
MS34919V3



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

1. 内部および外部クロックソース特性の詳細については、デバイスのデータシートの「電気的特性」のセクションを参照してください。
2. TIMx (x = 1/2/3/4/8/15/16/17/20) は、システムクロックソースが PLL で、AHB または APB2 サブシステムが累積で 2 を超えて分周されていない場合に、144 MHz で動作中の PLL からクロック供給を受けられます。
3. ADC クロックは、ADC バスインタフェースの AHB クロックから生成し、プログラム可能な係数 (1、2、または 4) で分周することができます。プログラム可能な係数が「1」の場合、AHB プリスケーラは「1」でなければなりません。

図 15. STM32F303x6/8 および STM32F328x8 クロックツリー



1. 内部および外部クロックソース特性の詳細については、デバイスのデータシートの「電気的特性」のセクションを参照してください。
2. TIM1 は、システムクロックソースが PLL で、AHB または APB2 サブシステムが累積で 2 を超えて分周されていない場合に、144 MHz で動作する PLL からクロック供給を受けられます。
3. ADC クロックは、ADC バスインタフェースの AHB クロックから生成し、プログラム可能な係数 (1、2、または 4) で分周することができます。プログラム可能な係数が「1」の場合、AHB プリスケーラは「1」でなければなりません。

FCLK は Cortex-M4<sup>®</sup>F のフリーランニングクロックとして動作します。詳細については、STM32F3xx/F4xx Cortex<sup>®</sup>-M4 プログラミングマニュアル (PM0214) を参照してください。

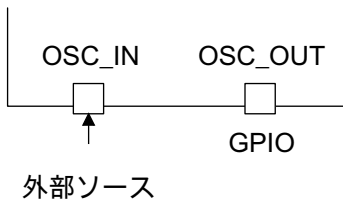
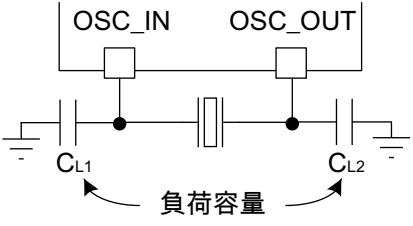
## 9.2.1 HSE クロック

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

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

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

図 16. HSE/LSE クロックソース

クロックソース	ハードウェア構成
外部クロック	 <p style="text-align: right;">MSv31915V1</p>
クリスタル/ セラミック発振子	 <p style="text-align: right;">MSv31916V1</p>

## 外部クリスタル／セラミック発振子 (HSE クリスタル)

4 から 32 MHz の外部オシレータには、メインクロックの周波数を非常に高い精度で生成できるという利点があります。

関連するハードウェア構成を図 16 に示します。詳細については、データシートの電気的特性のセクションを参照してください。

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

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

**注意：** HSE オシレータをオンにするには、HSEON ビットのセット後に内部安定カウンタで 512 HSE クロックパルスを検出する必要があります。デバイスにクリスタルや発振子が接続されていない場合でも、OSC\_IN ピンの過剰な外部ノイズによってオシレータが起動する場合があります。オシレータを起動すると、オフにするシーケンスを完了するために、別の 6 HSE クロックパルスが必要となります。何らかの理由で OSC\_IN ピン上の振動がなくなった場合、オシレータをオフにすることはできず、OSC ピンは他で使用できないようロックされ、不要な消費電力が発生します。このような状況を避けるには、このような場合にオシレータをオフにすることができるクロックセキュリティシステム (CSS) を常に有効にすることを強くお勧めします。

## 外部ソース (HSE バイパス)

このモードでは、外部クロックソースが必要です。最大 32 MHz までの周波数を使用できます。このモードを選択するには、**クロック制御レジスタ (RCC\_CR)** に HSEBYP および HSEON ビットをセットします。周波数 (データシートを参照) に応じて 40 ~ 60% までのデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC\_IN ピンを駆動する必要があり、その間、OSC\_OUT ピンは GPIO として使用できます。図 16 を参照してください。

## 9.2.2 HSI クロック

HSI クロック信号は、内部 8 MHz RC オシレータから生成され、システムクロックとして直接使用するか、PLL 入力として使用するために 2 分周できます。

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

### 較正

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

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

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

HSI 周波数変動の測定方法の詳細については、**セクション 9.2.14 : TIM16 を使用した内部／外部クロックの測定**を参照してください。



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

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

HSI 信号は、HSE クリスタルオシレータに障害がある場合のバックアップソース (補助クロック) としても使用できます。[セクション 9.2.7: クロックセキュリティシステム \(CSS\) \(130 ページ\)](#) を参照してください。

## 9.2.3 PLL

内部 PLL は、HSI、または HSE 出力クロック周波数を逡倍するために使用できます。[図 13](#) および **クロック制御レジスタ (RCC\_CR)** を参照してください。

PLL 設定 (入力クロックおよび逡倍数の選択) は、PLL を有効にする前に実行する必要があります。PLL が有効になると、これらのパラメータは変更できません。

PLL 設定を変更するには、次の手順に従います。

1. PLLON を 0 にセットして PLL を無効にします。
2. PLLRDY がクリアされるまで待ちます。PLL が完全に停止します。
3. 任意のパラメータを変更します。
4. PLLON を 1 にセットして PLL を再度有効にします。

**クロック割り込みレジスタ (RCC\_CIR)** で有効になっていれば、PLL がレディ状態になると割り込みを生成することができます。

PLL 出力周波数は、16 ~ 72 MHz の範囲でセットする必要があります。

## 9.2.4 LSE クロック

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

LSE クリスタルは、**RTC ドメイン制御レジスタ (RCC\_BDCR)** の LSEON ビットを使用してオン/オフの切り替えができます。安定性、短い起動時間、および低い電力消費のバランスを取るために、クリスタルオシレータの駆動能力は、**RTC ドメイン制御レジスタ (RCC\_BDCR)** の LSEDRV[1:0] ビットによって、実行時に変更できます。

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

**注意:** **LSE オシレータをオンにするには、LSEON ビットのセット後に内部安定カウンタで 4096 LSE クロックパルスを検出する必要があります。デバイスにクリスタルや発振子が接続されていない場合でも、OSC32\_IN ピンの過剰な外部ノイズによってオシレータが起動する場合があります。オシレータを起動すると、オフにするシーケンスを完了するために、別の 6 LSE クロックパルスが必要となります。何らかの理由で OSC\_IN ピン上の振動がなくなった場合、オシレータをオフにすることはできず、OSC32 ピンは他で使用できないようロックされ、不要な消費電力が発生します。このような状況を回復する唯一の方法は、ソフトウェアによって RTC ドメインをリセットすることです。**

## 外部ソース (LSE バイパス)

このモードでは、外部クロックソースが必要です。最大 1 MHz までの周波数を使用できます。このモードを選択するには、[RTC ドメイン制御レジスタ \(RCC\\_BDCR\)](#) に LSEBYP および LSEON ビットをセットします。50 % までのデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC\_32\_IN ピンを駆動する必要があり、その間、OSC\_32\_OUT ピンを GPIO として使用することができます。[図 16](#)を参照してください。

## 9.2.5 LSI クロック

LSI RC は、独立型ウォッチドッグ (IWDG) および RTC の STOP モードおよび STANDBY モードでの動作を可能にする低電力のクロックソースとして動作します。クロック周波数は約 40 kHz (30 kHz と 50 kHz の間) です。詳細については、データシートの電気的特性のセクションを参照してください。

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

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

## 9.2.6 システムクロック (SYSCLK) の選択

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

- HSI オシレータ
- HSE オシレータ
- PLL

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

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

## 9.2.7 クロックセキュリティシステム (CSS)

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

HSE クロックに異常が検出された場合、HSE オシレータは自動で無効になり、クロック障害イベントが高機能制御タイマ (TIM1/TIM8 および TIM15/16/17) のブレーク入力に送られます。また、ソフトウェアに異常を通知する割り込み (クロックセキュリティシステム割り込み (CSSI)) が生成されるので、MCU は復旧処理を行うことができます。CSSI は、Cortex-M4<sup>®</sup>F NMI (ノンマスカブル割り込み) 例外ベクタにリンクされています。

注： CSS を有効にすると、HSE クロックに障害が発生した場合に CSS 割り込みが発生し、NMI が自動的に生成されます。NMI は、CSS 割り込みペンディングビットがクリアされない限り、無限に実行されます。そのため、NMI 割り込みサービスルーチン (ISR) では、**クロック割り込みレジスタ (RCC\_CIR)** の CSSC ビットをセットして CSS 割り込みをクリアする必要があります。

HSE オシレータが直接的または間接的にシステムクロックとして使用されている場合（間接的とは、PLL 入力クロックとして使用され、その PLL クロックがシステムクロックとして使用されることを意味する）、障害が検出されると、システムクロックが HSI オシレータに切り替えられ、HSE オシレータは無効になります。障害発生時に HSE クロック（分周されているかいないかにかかわらず）がシステムクロックとして使用されている PLL のクロック入力であった場合には、PLL も無効になります。

## 9.2.8 ADC クロック

PLL 出力から生成された ADC クロック。72 MHz に達することができ、その後プリスケール値 1、2、4、6、8、10、12、16、32、64、128、または 256 で分周できます。これは、AHB クロックと非同期です。代わりに、ADC バスインタフェースの AHB クロックから ADC クロックを生成し、プログラム可能な係数 (1、2、または 4) で分周できます。プログラム可能な係数は、ADCx\_CCR の CKMODE ビットフィールドを使用して設定されます。

プログラムされた係数が「1」の場合、AHB プリスケールは「1」にセットする必要があります。

## 9.2.9 RTC クロック

RTCCLK クロックソースは HSE/32、LSE、または LSI クロックのいずれかです。これは、**RTC ドメイン制御レジスタ (RCC\_BDCR)** の RTCSEL[1:0] ビットをプログラムすることで選択できます。この選択は、RTC ドメインをリセットしない限り変更できません。RTC を正常に動作させるには、PCLK 周波数が常に RTCCLK 周波数以上になるようにシステムを設定する必要があります。

LSE クロックは RTC ドメインに属しますが、HSE と LSI クロックはそうではありません。そのため、次のようになります。

- LSE が RTC クロックとして使用されている場合：
  - $V_{DD}$  供給がオフになった場合でも、 $V_{BAT}$  の供給が保たれている限り、RTC は動作を続けます。
  - RTC は、システムリセット中でもクロック供給された状態、かつ機能的な状態のままです。
- LSI が RTC クロックとして使用されている場合：
  - $V_{DD}$  供給がオフになった場合、RTC の状態は保証されません。
- 32 分周された HSE クロックが RTC クロックとして使用されている場合：
  - $V_{DD}$  供給がオフになった場合や、内蔵されている電圧レギュレータがオフになった場合 (1.8 V 電源ドメインへの電力供給が停止にされた場合)、RTC の状態は保証されません。

## 9.2.10 タイマ (TIMx) クロック

### APB クロックソース

タイマクロック周波数は、ハードウェアによって自動で設定されます。これには、2 つのケースがあります。

1. APB プリスケールが 1 の場合、タイマクロック周波数は APB ドメインのクロックと同じ周波数に設定されます。
2. そうでない場合は、APB ドメインの周波数の 2 倍に設定されます。

## PLL クロックソース

PLL (PLLCLKx2) から発行されたクロックを TIMx に選択できます (このとき、STM32F303xB/C と STM32F358xC の場合  $x = 1, 8$ ; STM32F303xD/E の場合  $x = 1, 2, 3, 4, 8, 15, 16, 17, 20$ ; STM32F303x6/8 および STM32F328x8 の場合  $x = 1$ )。

STM32F303x6/8 および STM32F328x8 システムクロックソースが PLL の場合、この設定では最大 144 MHz の周波数を TIMx に供給できます。

この設定で：

- STM32F303xB/C および STM32F358xC では、AHB および APB2 プリスケーラは 1 にセットされます (AHB および APB2 クロックはシステムクロックに対して分周されません)。
- STM32F303xD/E STM32F303x6/8 および STM32F328x8 では、AHB または APB2 サブシステムクロックはシステムクロックに対して累積で 2 を超えて分周されません。

## 9.2.11 ウォッチドッグクロック

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

## 9.2.12 I2S クロック (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

I2S クロックは、I2S\_CKIN ピンに供給されたシステムクロックまたは外部クロックのいずれかです。I2S クロックソースの選択は、RCC\_CFGR レジスタのビット 23 (I2SSRC) を使用して実行されます。

## 9.2.13 クロック信号出力

マイクロコントローラクロック出力 (MCO) 機能では、外部 MCO ピンにクロックを出力することができます。対応する GPIO ポートの設定レジスタは、オルタネート機能モードに設定されている必要があります。5 つのクロック信号のうちの 1 つを MCO クロックとして選択できます。

- LSI
- LSE
- SYSCLK
- HSI
- HSE
- 2 分周された PLL クロック

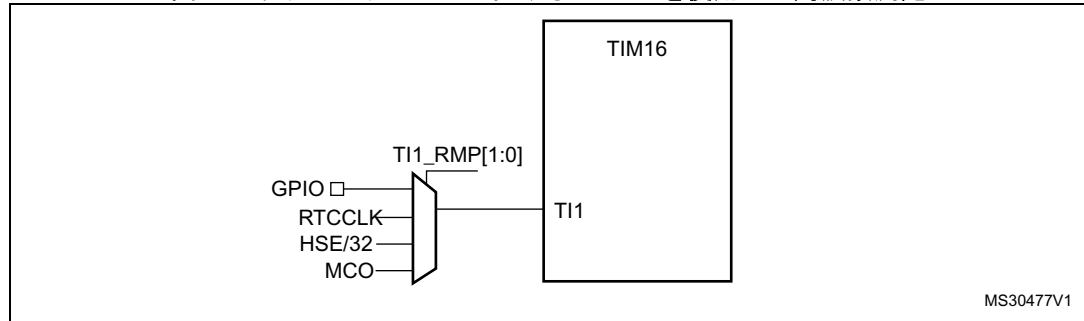
選択は [クロック設定レジスタ \(RCC\\_CFGR\)](#) の MCO[2:0] ビットによって制御されます。

STM32F303xD/E、STM32F303x6/8 および STM32F328x8、では、このレジスタの追加ビット PLLNODIV が、MCO への PLL クロック入力の分周バイパスを制御します。MCO 周波数は、クロック設定レジスタ (RCC\_CFGR) の MCOPRE[2..0] ビットで制御される設定可能な分周器によって低減できます。

## 9.2.14 TIM16 を使用した内部/外部クロックの測定

TIM16 チャンネル 1 の入力キャプチャを使用することにより、ボード上のすべてのクロックソースの周波数を間接的に測定することができます。図 17 を参照してください。

図 17. キャプチャモードにおける TIM16 を使用した周波数測定



タイマ 16 の入力キャプチャチャンネルは GPIO ラインまたは MCU の内部クロックです。この選択は、TIM16\_OR レジスタの TI1\_RMP [1:0] ビットで行います。以下を使用できます。

- TIM16 のチャンネル 1 は、GPIO に接続されます。デバイスのデータシートのオルタネート機能配置表を参照してください。
- TIM16 のチャンネル 1 は、RTCCLK に接続されます。
- TIM16 のチャンネル 1 は、HSE/32 クロックに接続されます。
- TIM16 のチャンネル 1 はマイクロコントローラクロック出力 (MCO) に接続されます。この選択は、クロック設定レジスタ (RCC\_CFGR) の MCO[2:0] ビットで制御されます。

### HSI の較正

MCO マルチプレクサを通じてチャンネル 1 の入力キャプチャに LSE を接続する主な目的は、HSI システムクロックを正確に測定できるようにすることです (これには、システムクロックソースとして HSI を使用する必要があります)。LSE 信号の連続したエッジ間の HSI クロックカウント数により、内部クロック周期が測定されます。LSE クリスタルの高い精度 (通常は数十 ppm) を利用することにより、同じ分解能で内部クロック周波数を判断でき、製造工程、温度、電圧に関連した周波数偏差に対してクロックソースを調整して補正することができます。

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

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

LSE を使用できない場合、可能な限り正確な較正を達成するには、HSE/32 を選択することをお勧めします。

### LSI の較正

LSI の較正は HSI と同じ方法で行いますが、リファレンスクロックを変更します。LSI クロックを TIM16 チャンネル 1 の入力キャプチャに接続する必要があります。次に、HSE をシステムクロックソースとして定義し、LSI 信号の連続したエッジ間のクロックカウント数により、内部低速クロック周期が測定されます。

この基本概念は、相対測定 (HSE/LSI 比など) ができることにあり、そのため、精度は 2 つのクロックソース間の比に密接にかかわっています。この比が大きいほど、測定精度は向上します。

## 9.3 低電力モード

APB ペリフェラルクロックおよび DMA クロックは、ソフトウェアで無効にできます。

SLEEP モードでは CPU クロックを停止します。メモリインタフェースクロック（フラッシュおよび RAM インタフェース）は、ソフトウェアによって SLEEP モード中に停止できます。AHB - APB ブリッジのクロックは、SLEEP モード中にこれらに接続されたペリフェラルのクロックがすべて無効になった場合に、ハードウェアによって無効化されます。

STOP モードでは、V18 ドメインのすべてのクロックを停止し、PLL と HSI および HSE オシレータを無効にします。

すべての U(S)ART と I2C では、MCU が STOP モードの場合でも、HSI オシレータを有効にすることができます（HSI が当該ペリフェラルのクロックソースとして選択されている場合）。

すべての U(S)ART は、システムが STOP モードの場合（LSE が当該ペリフェラルのクロックソースとして選択されている場合）で LSE オシレータが有効な場合（LSEON）に、LSE オシレータによる駆動もできます。ただし、LSE オシレータをオンにすることはできません。

STANDBY モードでは、V18 ドメインのすべてのクロックを停止し、PLL と HSI および HSE オシレータを無効にします。

CPU のディープスリープモードは、DBGMCU\_CR レジスタに DBG\_STOP または DBG\_STANDBY ビットをセットすることで、デバッグのためにオーバーライドすることができます。

割り込み（STOP モード）またはリセット（STANDBY モード）後にディープスリープからウェイクアップすると、HSI オシレータがシステムクロックとして選択されます。

フラッシュプログラム操作が実行中の場合、フラッシュインタフェースアクセスが終了してから、ディープスリープモードに移行します。APB ドメインにアクセス中の場合、APB アクセスが終了してから、ディープスリープモードに移行します。

## 9.4 RCC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

### 9.4.1 クロック制御レジスタ (RCC\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 XX83 (X は未定義)。

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

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

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **PLL RDY** : PLL クロックレディフラグ

ハードウェアによってセットされ、PLL がロック状態であることを示します。

0 : PLL アンロック

1 : PLL ロック

ビット 24 **PLLON** : PLL イネーブル

PLL を有効にするために、ソフトウェアによってセット/クリアされます。

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

0 : PLL オフ

1 : PLL ON

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CSSON** : クロックセキュリティシステム有効化

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

0 : クロック検出回路オフ

1 : クロック検出回路オン (HSE オシレータがレディである場合はクロック検出回路オン、そうでなければオフ)

ビット 18 **HSEBYP** : HSE クリスタルオシレータバイパス

外部クロックでオシレータをバイパスするために、ソフトウェアによってセット/クリアされます。デバイスで使用するには、HSEON ビットをセットして外部クロックを有効にする必要があります。HSEBYP ビットは、HSE オシレータが無効のときのみ、書き込みができます。

0 : HSE クリスタルオシレータはバイパスされません。

1 : HSE クリスタルオシレータが外部クロックによってバイパスされます。

**ビット 17 HSERDY** : HSE クロックレディフラグ

HSE オシレータが安定していることを示すために、ハードウェアによってセットされます。このビットでは HSEON リセット後に立ち上げるために、HSE オシレータクロックが 6 サイクル必要です。

- 0 : HSE オシレータはレディ状態ではありません。
- 1 : HSE オシレータはレディ状態です。

**ビット 16 HSEON** : HSE クロック有効化

ソフトウェアでセット/クリアされます。

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

- 0 : HSE オシレータオフ
- 1 : HSE オシレータオン

**ビット 15:8 HSICAL[7:0]** : HSI クロック較正

これらのビットは、スタートアップ時に自動的に初期化されます。

**ビット 7:3 HSITRIM[4:0]** : HSI クロックトリミング

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

デフォルト値は 16 で、HSICAL 値に追加すると、HSI を  $8 \text{ MHz} \pm 1\%$  にトリミングします。トリミングステップ ( $F_{\text{hsitrim}}$ ) は、2 つの連続する HSICAL ステップの間で、40 kHz 前後です。

ビット 2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 HSIRDY** : HSI クロックレディフラグ

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

- 0 : HSI オシレータはレディ状態ではありません。
- 1 : HSI オシレータはレディ状態です。

**ビット 0 HSION** : HSI クロック有効化

ソフトウェアでセット/クリアされます。

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

- 0 : HSI オシレータオフ
- 1 : HSI オシレータオン



## 9.4.2 クロック設定レジスタ (RCC\_CFGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLNO DIV	MCOPRE[2:1]		MCOF/ MCOP RE0	Res.	MCO[2:0]			I2SSSRC	USBPR E	PLLMUL[3:0]				PLL XTPRE	PLL SRC
r/w	r/w	r/w	r / r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLSR C <sup>(1)</sup>	Res.	PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r	r	r/w	r/w

1. STM32F303xD/E のみ

ビット 31 **PLLNO DIV** : MCO への PLL を分周しないこと (STM32F303x6/8 および STM32F328x8、STM32F303xDxE および STM32F398xE のみ)

このビットは、ソフトウェアによってセット/クリアされます。MCO への PLL の接続は 2 分周でスイッチオフされます。

0 : MCO の前で PLL を 2 分周

1 : MCO の前で PLL は分周されない

ビット 30:28 **MCOPRE** : マイクロコントローラクロック出力プリスケラ (STM32F303x6/8 および STM32F328x8、STM32F303xDxE および STM32F398xE のみ)

これらのビットは、ソフトウェアによってセット/クリアされます。MCO出力を有効にする前に、このプリスケラの変更を強く推奨します。

000 : MCO を 1 分周

001 : MCO を 2 分周

010 : MCO を 4 分周

.....

111 : MCO を 128 分周

ビット 28 **MCOF** : マイクロコントローラクロック出力フラグ (STM32F303xB/C および STM32F358xC のみ)

ハードウェアでセット/リセットされます。

MCO フィールドに新しい値が書き込まれたときにソフトウェアによってリセットされます。

新しい MCO ソースへのスイッチが有効であるときに、ハードウェアによってセットされます。

ビット 27 予約済みであり、リセット値に保持する必要があります。



ビット 26:24 **MCO** : マイクロコントローラクロック出力

ソフトウェアでセット/クリアされます。

000 : MCO 出力無効、MCO にクロックなし

001 : 予約済み

010 : LSI クロックの選択

011 : LSE クロックの選択

100 : システムクロック (SYSCLK) の選択

101 : HSI クロックの選択

110 : HSE クロックの選択

111 : PLL クロックの選択 (PLL NODIV ビットに応じて 1 または 2 で分周)

**注 :** このクロック出力では、起動時または MCO クロックソースの切り替え時に、切り捨てのサイクルがある場合があります。

ビット 23 **I2SSRC** : I2S 外部クロックソース選択 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

外部クロックで I2S2 および I2S3 をクロック供給するために、ソフトウェアによってセット/リセットされます。I2S2-3 クロックを有効にする前に、このビットを有効にする必要があります。

0 : システムクロックによってクロック供給された I2S2 および I2S3

1 : 外部クロックによってクロック供給された I2S2 および I2S3

ビット 22 **USBPRE** : USB プリスケアラ

このビットは、48 MHz USB クロックを生成するために、ソフトウェアによってセット/リセットされます。USB クロックを有効にする前に、これらを有効にする必要があります。

0 : PLL クロックを 1.5 分周

1 : PLL クロックは分周されない

ビット 21:18 **PLLMUL** : PLL 乗倍数

これらのビットは、PLL 乗倍数を定義するために、ソフトウェアによって書き込まれます。これらのビットに書き込めるのは、PLL が無効なときだけです。

注意 : PLL 出力周波数は 72 MHz を超えてはなりません。

0000 : PLL 入力クロック × 2

0001 : PLL 入力クロック × 3

0010 : PLL 入力クロック × 4

0011 : PLL 入力クロック × 5

0100 : PLL 入力クロック × 6

0101 : PLL 入力クロック × 7

0110 : PLL 入力クロック × 8

0111 : PLL 入力クロック × 9

1000 : PLL 入力クロック × 10

1001 : PLL 入力クロック × 11

1010 : PLL 入力クロック × 12

1011 : PLL 入力クロック × 13

1100 : PLL 入力クロック × 14

1101 : PLL 入力クロック × 15

1110 : PLL 入力クロック × 16

1111 : PLL 入力クロック × 16

ビット 17 **PLLXTPRE** : PLL 入力クロックの HSE 分周

このビットは、PLL の HSE 分周比を選択するために、ソフトウェアによってセット/クリアされます。これらに書き込めるのは、PLL が無効なときだけです。

**注 :** このビットはクロック設定レジスタ 2 (RCC\_CFGR2) の PREDIV の LSB と同じです (他の STM32 製品との互換性のため)。

0000 : PLL への HSE 入力は分周されません。

0001 : PLL への HSE 入力が 2 分周されます。

- ビット 16:15 **PLL SRC** : PLL 入力クロックソース (STM32F303xD/E および STM32F398xE のみ)  
PLL クロックソースを選択するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLL が無効なときだけです。  
00 : PREDIV1 エントリとして HSI/2 が使われ、PREDIV1 は強制的に 2 分周されます。  
01 : HSI が PREDIV1 エントリとして使用されます。  
10 : HSE が PREDIV1 エントリとして使用されます。  
11 : 予約済み。
- ビット 16 **PLL SRC** : PLL 入力クロックソース (STM32F303xB/C および STM32F358xC および STM32F303x6/8 および STM32F328x8 デバイス)  
PLL クロックソースを選択するために、ソフトウェアによってセット/クリアされます。このビットに書き込めるのは、PLL が無効なときだけです。  
0 : PLL 入力クロックとして HSI/2 を選択  
1 : PLL 入力クロックとして HSE/PREDIV を選択 ([セクション 9.4.12 : クロック設定レジスタ 2 \(RCC\\_CFGR2\) \(158 ページ\)](#) を参照)
- ビット 15 予約済みであり、STM32F303xB/C、STM32F358xC、STM32F303x6/8、および STM32F328x8 デバイスではリセット値を保持する必要があり、STM32F303xD/E では PLL クロックソースを選択するためにビット 16 とともに使用する必要があります。
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13:11 **PPRE2** : APB 高速プリスケアラ (APB2)  
APB2 クロック (PCLK) の分周比を制御するために、ソフトウェアによってセット/クリアされます。  
0xx : HCLK は分周されません。  
100 : HCLK は 2 分周されます。  
101 : HCLK は 4 分周されます。  
110 : HCLK は 8 分周されます。  
111 : HCLK は 16 分周されます。
- ビット 10:8 **PPRE1** : APB 低速プリスケアラ (APB1)  
APB1 クロック (PCLK) の分周比を制御するために、ソフトウェアによってセット/クリアされます。  
0xx : HCLK は分周されません。  
100 : HCLK は 2 分周されます。  
101 : HCLK は 4 分周されます。  
110 : HCLK は 8 分周されます。  
111 : HCLK は 16 分周されます。

### ビット 7:4 HPRE : HLCK プリスケーラ

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

- 0xxx : SYSCLK は分周されません。
- 1000 : SYSCLK は 2 分周されます。
- 1001 : SYSCLK は 4 分周されます。
- 1010 : SYSCLK は 8 分周されます。
- 1011 : SYSCLK は 16 分周されます。
- 1100 : SYSCLK は 64 分周されます。
- 1101 : SYSCLK は 128 分周されます。
- 1110 : SYSCLK は 256 分周されます。
- 1111 : SYSCLK は 512 分周されます。

**注 :** AHB クロックで 1 以外のプリスケーラを使用している場合は、プリフェッチバッファをオンにしておく必要があります。詳細については、[セクション 読み出し操作 \(64 ページ\)](#) を参照してください。

### ビット 3:2 SWS : システムクロックスイッチステータス

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

- 00 : HSI オシレータがシステムクロックとして使用されています。
- 01 : HSE オシレータがシステムクロックとして使用されています。
- 10 : PLL がシステムクロックとして使用されています。
- 11 : 適用されません。

### ビット 1:0 SW : システムクロックスイッチ

SYSCLK ソースを選択するために、ソフトウェアによってセット/クリアされます。

STOP および STANDBY モードを終了するとき、またはシステムクロックとして直接的または間接的に使用されている HSE オシレータに障害が発生したときに、強制的に HSI を選択するようにハードウェアによってクリアされます。

- 00 : HSI がシステムクロックとして選択されます。
- 01 : HSE がシステムクロックとして選択されます。
- 10 : PLL がシステムクロックとして選択されます。
- 11 : 設定禁止

## 9.4.3 クロック割り込みレジスタ (RCC\_CIR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

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

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

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

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

0 : 影響なし。

1 : CSSF フラグをクリアします。

ビット 22:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **PLL RDYC** : PLL レディ割り込みクリア

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

0 : 影響なし。

1 : PLL RDYF フラグをクリアします。

ビット 19 **HSE RDYC** : HSE レディ割り込みクリア

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

0 : 影響なし。

1 : HSE RDYF フラグをクリアします。

ビット 18 **HSI RDYC** : HSI レディ割り込みクリア

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

0 : 影響なし。

1 : HSI RDYF フラグをクリアします。

ビット 17 **LSE RDYC** : LSE レディ割り込みクリア

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

0 : 影響なし。

1 : LSE RDYF がクリアされます。

ビット 16 **LSI RDYC** : LSI レディ割り込みクリア

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

0 : 影響なし。

1 : LSI RDYF がクリアされます。

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **PLL RDYIE** : PLL レディ割り込み有効化

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

0 : PLL ロック割り込みは無効です。

1 : PLL ロック割り込みは有効です。



- ビット 11 **HSERDYIE** : HSE レディ割り込み有効化  
HSE オシレータの安定化によって発生する割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : HSE レディ割り込みは無効です。  
1 : HSE レディ割り込みは有効です。
- ビット 10 **HSIRDYIE** : HSI レディ割り込み有効化  
HSI オシレータの安定化によって発生する割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : HSI レディ割り込み無効  
1 : HSI レディ割り込み有効
- ビット 9 **LSERDYIE** : LSE レディ割り込み有効化  
LSE オシレータの安定化によって発生する割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : LSE レディ割り込みは無効です。  
1 : LSE レディ割り込みは有効です。
- ビット 8 **LSIRDYIE** : LSI レディ割り込み有効化  
LSI オシレータの安定化によって発生する割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : LSI レディ割り込みは無効です。  
1 : LSI レディ割り込みは有効です。
- ビット 7 **CSSF** : クロックセキュリティシステム割り込みフラグ  
HSE オシレータで障害が検出されたときに、ハードウェアによってセットされます。  
CSSC ビットをセットすることによって、ソフトウェアによってクリアされます。  
0 : HSE クロック障害によるクロックセキュリティ割り込みは発生していません。  
1 : HSE クロック障害によってクロックセキュリティ割り込みが発生しました。
- ビット 6:5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **PLLRDYF** : PLL レディ割り込みフラグ  
PLL がロックされ、PLLRDYIE がセットされているときに、ハードウェアによってセットされます。  
PLLRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。  
0 : PLL ロックによるクロックレディ割り込みは発生していません。  
1 : PLL ロックによるクロックレディ割り込みが発生しました。
- ビット 3 **HSERDYF** : HSE レディ割り込みフラグ  
HSE クロックが安定し、HSERDYIE がセットされているときに、ハードウェアによってセットされます。  
HSERDYC ビットをセットすることによって、ソフトウェアによってクリアされます。  
0 : HSE オシレータによるクロックレディ割り込みは発生していません。  
1 : HSE オシレータによるクロックレディ割り込みが発生しました。

ビット 2 **HSIRDYF** : HSI レディ割り込みフラグ

HSI クロックが安定し、HSION (クロック制御レジスタ (RCC\_CR) を参照) のセットに応じて HSIRDYDIE がセットされたときに、ハードウェアによってセットされます。HSION がセットされず、ただしクロックリクエストを通じてペリフェラルによって HSI オシレータが有効化された場合、このビットはセットされず、割り込みも生成されません。

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

- 0 : HSI オシレータによるクロックレディ割り込みは発生していません。
- 1 : HSI オシレータによるクロックレディ割り込みが発生しました。

ビット 1 **LSERDYF** : LSE レディ割り込みフラグ

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

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

- 0 : LSE オシレータによるクロックレディ割り込みは発生していません。
- 1 : LSE オシレータによるクロックレディ割り込みが発生しました。

ビット 0 **LSIRDYF** : LSI レディ割り込みフラグ

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

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

- 0 : LSI オシレータによるクロックレディ割り込みは発生していません。
- 1 : LSI オシレータによるクロックレディ割り込みが発生しました。

## 9.4.4 APB2 ペリフェラルリセットレジスタ (RCC\_APB2RSTR)

アドレスオフセット : 0x0C

リセット値 : 0x00000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM20 RST	Res.	TIM17 RST	TIM16 RST	TIM15 RST
													rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4RST	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYS CFG RST
rW	rW	rW	rW	rW											rW

ビット 31:2119 予約済みであり、リセット値に保持する必要があります。

ビット 20 **TIM20RST** : TIM20 タイマリセット (STM32F303xD/E および STM32F398xE デバイスのみ)

ソフトウェアでセット/クリアされます。

- 0 : 影響なし。
- 1 : TIM20 タイマをリセットします。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM17RST** : TIM17 タイマリセット

ソフトウェアでセット/クリアされます。

- 0 : 影響なし。
- 1 : TIM17 タイマをリセットします。



- ビット 17 **TIM16RST** : TIM16 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM16 タイマをリセットします。
- ビット 16 **TIM15RST** : TIM15 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM15 タイマをリセットします。
- ビット 15 **SPI4RST** : SPI4 リセット (STM32F303xD/E および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : SPI4 をリセットします。
- ビット 14 **USART1RST** : USART1 リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : USART1 をリセットします。
- ビット 13 **TIM8RST** : TIM8 タイマリセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM8 タイマをリセットします。
- ビット 12 **SPI1RST** : SPI1 リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : SPI1 をリセットします。
- ビット 11 **TIM1RST** : TIM1 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM1 タイマをリセットします。
- ビット 10:1 予約済みであり、リセット値に保持する必要があります。
- ビット 0 **SYSCFGRST** : SYSCFG、コンパレータおよびオペアンプのリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : SYSCFG、COMP、および OPAMP のリセット



## 9.4.5 APB1 ペリフェラルリセットレジスタ (RCC\_APB1RSTR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	I2C3 RST <sup>(1)</sup>	DAC1 RST	PWR RST	Res.	DAC2R ST	CAN RST	Res.	USB RST	I2C2 RST	I2C1 RST	UART5 RST	UART4 RST	USART3 RST	USART2 RST	Res.
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Res.	Res.	WWDG RST	Res.	Res.	Res.	Res.	Res.	TIM7 RST	TIM6 RST	Res.	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw			rw						rw	rw		rw	rw	rw

1. STM32F303xD/E デバイスのみ。

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **I2C3RST** : I2C3 リセット (STM32F303xD/E および STM32F398xE デバイスのみ)

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I2C3 をリセットします。

ビット 29 **DAC1RST** : DAC1 インタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : DAC1 インタフェースをリセットします。

ビット 28 **PWRRST** : 電源インタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : 電源インタフェースをリセットします。

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **DAC2RST** : DAC2 インタフェースリセット (STM32F303x6/8 および STM32F328x8 デバイスのみ)

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : DAC2 インタフェースをリセットします。

ビット 25 **CANRST** : CAN リセット

ソフトウェアでセット/クリアされます。

0 : CAN をリセットしません。

1 : CAN をリセットします。

ビット 24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23 **USBRST** : USB リセット (STM32F303xB/C/D/E デバイスのみ)

ソフトウェアでセット/クリアされます。

0 : USB をリセットしません。

1 : USB をリセットします。



- ビット 22 **I2C2RST** : I2C2 リセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : I2C2 をリセットします。
- ビット 21 **I2C1RST** : I2C1 リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : I2C1 をリセットします。
- ビット 20 **UART5RST** : UART5 リセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : UART5 をリセットします。
- ビット 19 **UART4RST** : UART4 リセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : UART4 をリセットします。
- ビット 18 **USART3RST** : USART3 リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : USART3 をリセットします。
- ビット 17 **USART2RST** : USART2 リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : USART2 をリセットします。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3RST** : SPI3 リセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : SPI3 および I2S3 をリセットします。
- ビット 14 **SPI2RST** : SPI2 リセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : SPI2 および I2S2 をリセットします。
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGRST** : ウィンドウ型ウォッチドッグリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : ウィンドウ型ウォッチドッグをリセットします。
- ビット 10:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **TIM7RST** : TIM7 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM7 をリセットします。

- ビット 4 **TIM6RST** : TIM6 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM6 をリセットします。
- ビット 3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **TIM4RST** : TIM4 タイマリセット (STM32F303xB/C および STM32F358xC デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM4 をリセットします。
- ビット 1 **TIM3RST** : TIM3 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM3 をリセットします。
- ビット 0 **TIM2RST** : TIM2 タイマリセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : TIM2 をリセットします。

## 9.4.6 AHB ペリフェラルクロック有効レジスタ (RCC\_AHBENR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0014

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

**注 :** **ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの値はソフトウェアによって読み出しできず、返される値は常に 0x0 になります。**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ADC34 EN	ADC12EN	Res.	Res.	Res.	TSCEN	IOPG EN <sup>(1)</sup>	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	IOPH EN <sup>(1)</sup>
		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC EN	FMC EN <sup>(1)</sup>	FLITF EN	Res.	SRAM EN	DMA2 EN	DMA1 EN
									rw	rw	rw		rw	rw	rw

1. STM32F303xDxE のみ。

- ビット 31:30 予約済みであり、リセット値に保持する必要があります。
- ビット 29 **ADC34EN** : ADC3 および ADC4 有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : ADC3 および ADC4 クロック無効  
1 : ADC3 および ADC4 クロック有効
- ビット 28 **ADC12EN** : ADC1 および ADC2 有効化  
ソフトウェアでセット/クリアされます。  
0 : ADC1 および ADC2 クロック無効  
1 : ADC1 および ADC2 クロック有効



ビット 27:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TSCEN** : タッチセンシングコントローラクロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : TSC クロック無効
- 1 : TSC クロック有効

ビット 23 **IOPGEN** : IO ポート G クロック有効化 (STM32F303xDxE のみ)  
ソフトウェアでセット/クリアされます。

- 0 : IO ポート G クロック無効
- 1 : IO ポート G クロック有効

ビット 22 **IOPFEN** : I/O ポート F クロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : I/O ポート F クロック無効
- 1 : I/O ポート F クロック有効

ビット 21 **IOPEEN** : I/O ポート E クロック有効化 (STM32F303xB/C および STM32F358xC デバイスのみ)  
ソフトウェアでセット/クリアされます。

- 0 : I/O ポート E クロック無効
- 1 : I/O ポート E クロック有効

ビット 20 **IOPDEN** : I/O ポート D クロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : I/O ポート D クロック無効
- 1 : I/O ポート D クロック有効

ビット 19 **IOPCEN** : I/O ポート C クロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : I/O ポート C クロック無効
- 1 : I/O ポート C クロック有効

ビット 18 **IOPBEN** : I/O ポート B クロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : I/O ポート B クロック無効
- 1 : I/O ポート B クロック有効

ビット 17 **IOPAEN** : I/O ポート A クロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : I/O ポート A クロック無効
- 1 : I/O ポート A クロック有効

ビット 16 **IOPHEN** : IO ポート H クロック有効化 (STM32F303xDxE のみ)  
ソフトウェアでセット/クリアされます。

- 0 : IO ポート H クロック無効
- 1 : IO ポート H クロック有効

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **CRCCEN** : CRC クロック有効化  
ソフトウェアでセット/クリアされます。

- 0 : CRC クロック無効
- 1 : CRC クロック有効

ビット 5 **FMCEN** : FMC クロック有効化 (STM32F303xDxE のみ)  
ソフトウェアでセット/クリアされます。

- 0 : FMC クロック無効
- 1 : FMC クロック有効

ビット 4 **FLITFEN** : FLITF クロック有効化

SLEEP モード時に FLITF クロックを無効/有効にするために、ソフトウェアによってセット/クリアされます。

- 0 : SLEEP モード時 FLITF クロック無効
- 1 : SLEEP モード時 FLITF クロック有効

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **SRAMEN** : SRAM インタフェースクロック有効

SLEEP モード時に SRAM インタフェースクロックを無効/有効にするために、ソフトウェアによってセット/クリアされます。

- 0 : SLEEP モード時 SRAM インタフェースクロック無効
- 1 : SLEEP モード時 SRAM インタフェースクロック有効

ビット 1 **DMA2EN** : DMA2 クロック有効化 (STM32F303xB/C および STM32F358xC デバイスのみ)

ソフトウェアでセット/クリアされます。

- 0 : DMA2 クロック無効
- 1 : DMA2 クロック有効

ビット 0 **DMA1EN** : DMA1 クロック有効化

ソフトウェアでセット/クリアされます。

- 0 : DMA1 クロック無効
- 1 : DMA1 クロック有効

## 9.4.7 APB2 ペリフェラルクロック有効レジスタ (RCC\_APB2ENR)

アドレス : 0x18

リセット値 : 0x0000 0000

アクセス : ワード、ハーフワード、およびバイトアクセス

APB2 ドメイン内のペリフェラルへのアクセス中にアクセスが発生する場合を除き、ノーウェイトステート。この場合、APB2 ペリフェラルへのアクセスが終了するまでウェイトステートが挿入されます。

**注 :** ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの値はソフトウェアによって読み出しできず、返される値は常に 0x0 になります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM20 EN	Res.	TIM17 EN	TIM16 EN	TIM15 EN
											rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI4EN	USART1EN	TIM8 EN	SPI1 EN	TIM1 EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYS CFGEN
rw	rw	rw	rw	rw											rw

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **TIM20EN** : TIM20 タイマクロック有効化 (STM32F303xD/E および STM32F398xE のみ)

ソフトウェアでセット/クリアされます。

- 0 : TIM20 タイマクロック無効
- 1 : TIM20 タイマクロック有効

ビット 19 予約済みであり、リセット値に保持する必要があります。



- ビット 18 **TIM17EN** : TIM17 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM17 タイマクロック無効  
1 : TIM17 タイマクロック有効
- ビット 17 **TIM16EN** : TIM16 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM16 タイマクロック無効  
1 : TIM16 タイマクロック有効
- ビット 16 **TIM15EN** : TIM15 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM15 タイマクロック無効  
1 : TIM15 タイマクロック有効
- ビット 15 **SPI4EN** : SPI4 クロック有効化 (STM32F303xD/E および STM32F398xE のみ)  
ソフトウェアでセット/クリアされます。  
0 : SPI4 クロック無効  
1 : SPI4 クロック有効
- ビット 14 **USART1EN** : USART1 クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : USART1 クロック無効  
1 : USART1 クロック有効
- ビット 13 **TIM8EN** : TIM8 タイマクロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : TIM8 タイマクロック無効  
1 : TIM8 タイマクロック有効
- ビット 12 **SPI1EN** : SPI1 クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : SPI1 クロック無効  
1 : SPI1 クロック有効
- ビット 11 **TIM1EN** : TIM1 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM1 タイマクロック無効  
1 : TIM1 タイマクロック有効
- ビット 10:1 予約済みであり、リセット値に保持する必要があります。
- ビット 0 **SYSCFGEN** : SYSCFG クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : SYSCFG クロック無効  
1 : SYSCFG クロック有効

## 9.4.8 APB1 ペリフェラルクロック有効レジスタ (RCC\_APB1ENR)

アドレス : 0x1C

リセット値 : 0x0000 0000

アクセス : ワード、ハーフワード、およびバイトアクセス

APB1 ドメイン内のペリフェラルへのアクセス中にアクセスが発生する場合を除き、ノーウェイトステート。この場合、ウェイトステートは APB1 ペリフェラルへのアクセスが終了するまで挿入されません。

**注 :** ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの値はソフトウェアによって読み出しできず、返される値は常に 0x0 になります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	I2C3 EN	DAC1 EN	PWR EN	Res.	DAC2 EN	CAN EN	Res.	USB EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Res.
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Res.	Res.	WWD GEN	Res.	Res.	Res.	Res.	Res.	TIM7EN	TIM6EN	Res.	TIM4EN	TIM3EN	TIM2 EN
rw	rw			rw						rw	rw		rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **I2C3EN** : I2C3 クロック有効化 (STM32F303xD/E デバイスのみ)

ソフトウェアでセット/クリアされます。

0 : I2C3 クロック無効

1 : I2C3 クロック有効

ビット 29 **DAC1EN** : DAC1 インタフェースクロック有効化

ソフトウェアでセット/クリアされます。

0 : DAC1 インタフェースクロック無効

1 : DAC1 インタフェースクロック有効

ビット 28 **PWREN** : 電源インタフェースクロック有効

ソフトウェアでセット/クリアされます。

0 : 電源インタフェースクロック無効

1 : 電源インタフェースクロック有効

ビット 27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **DAC2EN** : DAC2 インタフェースクロック有効化 (STM32F303x6/8 および STM32F328x8 デバイスのみ)

ソフトウェアでセット/クリアされます。

0 : DAC2 インタフェースクロック無効

1 : DAC2 インタフェースクロック有効

ビット 25 **CANEN** : CAN クロック有効化

ソフトウェアでセット/クリアされます。

0 : CAN クロック無効

1 : CAN クロック有効

ビット 24 予約済みであり、リセット値に保持する必要があります。



- ビット 23 **USBEN** : USB クロック有効化 (STM32F303xB/C/D/E デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : USB クロック無効  
1 : USB クロック有効
- ビット 22 **I2C2EN** : I2C2 クロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : I2C2 クロック無効  
1 : I2C2 クロック有効
- ビット 21 **I2C1EN** : I2C1 クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : I2C1 クロック無効  
1 : I2C1 クロック有効
- ビット 20 **UART5EN** : UART5 クロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : UART5 クロック無効  
1 : UART5 クロック有効
- ビット 19 **UART4EN** : UART4 クロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : UART4 クロック無効  
1 : UART4 クロック有効
- ビット 18 **USART3EN** : USART3 クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : USART3 クロック無効  
1 : USART3 クロック有効
- ビット 17 **USART2EN** : USART2 クロック有効化  
ソフトウェアでセット/クリアされます。  
0 : USART2 クロック無効  
1 : USART2 クロック有効
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **SPI3EN** : SPI3 クロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : SPI3 クロック無効  
1 : SPI3 クロック有効
- ビット 14 **SPI2EN** : SPI2 クロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : SPI2 クロック無効  
1 : SPI2 クロック有効
- ビット 13:12 予約済みであり、リセット値に保持する必要があります。
- ビット 11 **WWDGEN** : ウィンドウ型ウォッチドッグクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : ウィンドウ型ウォッチドッグクロック無効  
1 : ウィンドウ型ウォッチドッグクロック有効



ビット 10:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **TIM7EN** : TIM7 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM7 クロック無効  
1 : TIM7 クロック有効

ビット 4 **TIM6EN** : TIM6 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM6 クロック無効  
1 : TIM6 クロック有効

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TIM4EN** : TIM4 タイマクロック有効化 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)  
ソフトウェアでセット/クリアされます。  
0 : TIM4 クロック無効  
1 : TIM4 クロック有効

ビット 1 **TIM3EN** : TIM3 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM3 クロック無効  
1 : TIM3 クロック有効

ビット 0 **TIM2EN** : TIM2 タイマクロック有効化  
ソフトウェアでセット/クリアされます。  
0 : TIM2 クロック無効  
1 : TIM2 クロック有効

## 9.4.9 RTC ドメイン制御レジスタ (RCC\_BDCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0018h、RTC ドメインリセットによってリセットされます。

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

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

**注 :** **RTC ドメイン制御レジスタ (RCC\_BDCR) の LSEON、LSEBYP、RTCSEL、および RTCEN ビットは、RTC ドメインにあります。このため、リセット後、これらのビットは書き込み保護されるので、これらを変更するには、電源制御レジスタ (PWR\_CR) の DBP ビットをセットする必要があります。これらのビットは、RTC ドメインリセット後にのみリセットされず (セクション 9.1.3 : RTC ドメインリセット を参照)。内部または外部リセットは、これらのビットに影響しません。**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST
															rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	Res.	Res.	LSEDRV[1:0]		LSE BYP	LSE RDY	LSEON
rW						rW	rW				rW	rW	rW	r	rW

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **BDRST** : RTC ドメインソフトウェアリセット

ソフトウェアでセット/クリアされます。

0 : リセットは有効ではありません。

1 : RTC ドメイン全体をリセットします。

ビット 15 **RTCEN** : RTC クロック有効化

ソフトウェアでセット/クリアされます。

0 : RTC クロック無効

1 : RTC クロック有効

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

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

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

00 : クロックなし。

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

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

11 : 32 分周された HSE オシレータクロックが RTC クロックとして使用されます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:3 **LSEDRV[1:0]** : LSE オシレータの駆動能力

LSE オシレータの駆動能力を調整するためにソフトウェアによってセットおよびリセットされます。RTC ドメインのリセットはデフォルト値を復元します。

00 : 「Xtal モード」 低駆動

01 : 「Xtal モード」 中高駆動

10 : 「Xtal モード」 中低駆動

11 : 「Xtal モード」 高駆動 (リセット値)

**注 :** オシレータは、バイパスモードでない場合、Xtal モードです。

**ビット 2 LSEBYP** : LSE オシレータバイパス

デバッグモードのオシレータをバイパスするために、ソフトウェアによってセット/クリアされます。このビットに書き込めるのは、外部 32 kHz オシレータが無効なときだけです。

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

**ビット 1 LSERDY** : LSE オシレータはレディ状態です。

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

- 0 : LSE オシレータはレディ状態ではありません。
- 1 : LSE オシレータはレディ状態です。

**ビット 0 LSEON** : LSE オシレータイネーブル

ソフトウェアでセット/クリアされます。

- 0 : LSE オシレータオフ
- 1 : LSE オシレータオン

## 9.4.10 制御/ステータスレジスタ (RCC\_CSR)

アドレス : 0x24

リセット値 : 0x0C00 0000。システムリセットによってリセットされます。ただし、リセットフラグは電源リセット時のみリセットされます。

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

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IW WDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	OB LRSTF	RMVF	V18PW RRSTF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI RDY	LSION
														r	rw

**ビット 31 LPWRSTF** : 低電力リセットフラグ

低電力管理リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

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

低電力管理リセットの詳細については、[リセット](#)を参照してください。

**ビット 30 WWDRSTF** : ウィンドウ型ウォッチドッグリセットフラグ

ウィンドウ型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

- 0 : ウィンドウ型ウォッチドッグリセットは発生していません。
- 1 : ウィンドウ型ウォッチドッグリセットが発生しました。



- ビット 29 **IWDGRSTF** : 独立型ウィンドウウォッチドッグリセットフラグ  
 $V_{DD}$  ドメインから独立型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。  
0 : ウォッチドッグリセットは発生していません。  
1 : ウォッチドッグリセットが発生しました。
- ビット 28 **SFTRSTF** : ソフトウェアリセットフラグ  
ソフトウェアリセットが発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。  
0 : ソフトウェアリセットは発生していません。  
1 : ソフトウェアリセットが発生しました。
- ビット 27 **PORRSTF** : POR/PDR フラグ  
POR/PDR が発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。  
0 : POR/PDR は発生していません。  
1 : POR/PDR が発生しました。
- ビット 26 **PINRSTF** : PIN リセットフラグ  
NRST ピンに外部リセット信号が入力され、リセットが発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。  
0 : NRST ピンからのリセットは発生していません。  
1 : NRST ピンからリセットが発生しました。
- ビット 25 **OBLRSTF** : オプションバイトローダリセットフラグ  
OBL からリセットが発生したときに、ハードウェアによってセットされます。RMVF ビットに書き込むことによってクリアされます。  
0 : OBL からのリセットは発生していません。  
1 : OBL からリセットが発生しました。
- ビット 24 **RMVF** : リセットフラグ解除  
リセットフラグをクリアするために、ソフトウェアによってセットされます。  
0 : 影響なし。  
1 : リセットフラグをクリアします。
- ビット 23 **V18PWRRSTF** : 1.8 V ドメインのフラグをリセットします。  
1.8 V ドメインの POR/PDR が発生したときに、ハードウェアによってクリアされます。RMVF ビットに書き込むことによってクリアされます。  
0 : 1.8 V ドメインの POR/PDR リセットは発生していません。  
1 : 1.8 V ドメインの POR/PDR リセットが発生しました。
- 注 : STM32F3x8 製品では、このフラグは予約済みです。
- ビット 22:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **LSIRDY** : LSI オシレータはレディ状態です。  
LSI オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSION ビットがクリアされた後、LSIRDY は、LSI オシレータクロックの 3 サイクル後にローになります。  
0 : LSI オシレータはレディ状態ではありません。  
1 : LSI オシレータはレディ状態です。
- ビット 0 **LSION** : LSI オシレータイネーブル  
ソフトウェアでセット/クリアされます。  
0 : LSI オシレータオフ  
1 : LSI オシレータオン

## 9.4.11 AHB ペリフェラルリセットレジスタ (RCC\_AHBRSTR)

アドレス : 0x28

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ADC34 RST	ADC12 RST	Res.	Res.	Res.	TSC RST	IOPG RST <sup>(1)</sup>	IOPF RST	IOPE RST	IOPD RST	IOPC RST	IOPB RST	IOPA RST	IOPH RST <sup>(1)</sup>
		rW	rW				rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMC RST <sup>(1)</sup>	Res.	Res.	Res.	Res.	Res.
										rW					

1. STM32F303xDxE のみ。

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **ADC34RST** : ADC3 および ADC4 リセット (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

ソフトウェアでセット/クリアされます。

0 : ADC3 および ADC4 をリセットしません。

1 : ADC3 および ADC4 をリセットします。

ビット 28 **ADC12RST** : ADC1 および ADC2 リセット

ソフトウェアでセット/クリアされます。

0 : ADC1 および ADC2 をリセットしません。

1 : ADC1 および ADC2 をリセットします。

ビット 27:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TSCRST** : タッチセンシングコントローラリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : TSC をリセットします。

ビット 23 **IOPGRST** : I/O ポート G リセット (STM32F303xDxE のみ)

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I/O ポート G をリセットします。

ビット 22 **IOPFRST** : I/O ポート F リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I/O ポート F をリセットします。

ビット 21 **OPERST** : I/O ポート E リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I/O ポート E をリセットします。

ビット 20 **IOPDRST** : I/O ポート D リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I/O ポート D をリセットします。



- ビット 19 **IOPCRST** : I/O ポート C リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : I/O ポート C をリセットします。
- ビット 18 **IOPBRST** : I/O ポート B リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : I/O ポート B をリセットします。
- ビット 17 **IOPARST** : I/O ポート A リセット  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : I/O ポート A をリセットします。
- ビット 16 **IOPHRST** : I/O ポート H リセット (STM32F303xDxE のみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : I/O ポート H をリセットします。
- ビット 15:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **FMC RST** : FMC リセット (STM32F303xDxE のみ)  
ソフトウェアでセット/クリアされます。  
0 : 影響なし。  
1 : FMC をリセットします。
- ビット 4:0 予約済みであり、リセット値に保持する必要があります。

## 9.4.12 クロック設定レジスタ 2 (RCC\_CFGR2)

アドレス : 0x2C

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADC34PRES[4:0]				ADC12PRES[4:0]				PREDIV[3:0]					
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:9 **ADC34PRES** : ADC34 のプリスケアラ (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE のみ)

PLL クロックの分周比を ADC34 に制御するために、ソフトウェアによってセット/リセットされます。

0xxxx : ADC34 クロック無効、ADC34 では AHB クロックを使用可能

10000 : 1 分周された PLL クロック

10001 : 2 分周された PLL クロック

10010 : 4 分周された PLL クロック

10011 : 6 分周された PLL クロック

10100 : 8 分周された PLL クロック

10101 : 10 分周された PLL クロック

10110 : 12 分周された PLL クロック

10111 : 16 分周された PLL クロック

11000 : 32 分周された PLL クロック

11001 : 64 分周された PLL クロック

11010 : 128 分周された PLL クロック

11011 : 256 分周された PLL クロック

上記以外 : 256 分周された PLL クロック

ビット 8:4 **ADC12PRES** : ADC12 プリスケアラ

PLL クロックの分周比を ADC12 に制御するために、ソフトウェアによってセット/リセットされます。

0xxxx : ADC12 クロック無効、ADC12 では AHB クロックを使用可能

10000 : 1 分周された PLL クロック

10001 : 2 分周された PLL クロック

10010 : 4 分周された PLL クロック

10011 : 6 分周された PLL クロック

10100 : 8 分周された PLL クロック

10101 : 10 分周された PLL クロック

10110 : 12 分周された PLL クロック

10111 : 16 分周された PLL クロック

11000 : 32 分周された PLL クロック

11001 : 64 分周された PLL クロック

11010 : 128 分周された PLL クロック

11011 : 256 分周された PLL クロック

上記以外 : 256 分周された PLL クロック

### ビット 3:0 PREDIV : PREDIV の分周比

これらのビットは、PREDIV の分周比を選択するために、ソフトウェアによってセット/クリアされません。これらに書き込めるのは、PLL が無効なときだけです。

**注:** ビット 0 はクロック設定レジスタ (RCC\_CFGR) のビット 17 と同じビットです。そのため、クロック設定レジスタ (RCC\_CFGR) のビット 17 を変更すると、クロック設定レジスタ 2 (RCC\_CFGR2) のビット 0 も変更されます (他の STM32 製品との互換性のため)。

- 0000 : PLL への HSE 入力 は分周されません。
- 0001 : PLL への HSE 入力 が 2 分周されます。
- 0010 : PLL への HSE 入力 が 3 分周されます。
- 0011 : PLL への HSE 入力 が 4 分周されます。
- 0100 : PLL への HSE 入力 が 5 分周されます。
- 0101 : PLL への HSE 入力 が 6 分周されます。
- 0110 : PLL への HSE 入力 が 7 分周されます。
- 0111 : PLL への HSE 入力 が 8 分周されます。
- 1000 : PLL への HSE 入力 が 9 分周されます。
- 1001 : PLL への HSE 入力 が 10 分周されます。
- 1010 : PLL への HSE 入力 が 11 分周されます。
- 1011 : PLL への HSE 入力 が 12 分周されます。
- 1100 : PLL への HSE 入力 が 13 分周されます。
- 1101 : PLL への HSE 入力 が 14 分周されます。
- 1110 : PLL への HSE 入力 が 15 分周されます。
- 1111 : PLL への HSE 入力 が 16 分周されます。



## 9.4.13 クロック設定レジスタ 3 (RCC\_CFGR3)

アドレス : 0x30

リセット値 : 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TIM34 SW <sup>(1)</sup>	TIM2 SW <sup>(1)</sup>	UART5SW[1:0]		UART4SW[1:0]		USART3SW[1:0] <sup>(2)</sup>		USART2SW[1:0] <sup>(2)</sup>	
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM20 SW <sup>(1)</sup>	Res.	TIM17 SW <sup>(1)</sup>	Res.	TIM16 SW <sup>(1)</sup>	TIM15 SW <sup>(1)</sup>	TIM8S W	TIM1 SW	Res.	I2C3 SW <sup>(1)</sup>	I2C2 SW	I2C1 SW	Res.	Res.	USART1SW[1:0]	
r/w		r/w		r/w	r/w	r/w	r/w		r/w	r/w	r/w			r/w	r/w

1. STM32F303xDxE のみ。
2. STM32F303x6/8 および STM32F328x8 デバイスでは使用不可。

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **TIM34SW** : Timer34 クロックソースの選択

TIM34 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

このビットは、システムクロックソースが PLL で、AHB または APB2 サブシステムクロックが累積で 2 を超えて分周されていない場合のみ書き込み可能です。

このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

- 0 : PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)
- 1 : PLL VCO 出力 (最大 144 MHz で動作)

注 : STM32F303xDxE および STM32F398xE のみ。

ビット 24 **TIM2SW** : Timer2 クロックソースの選択

TIM2 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

このビットは、クロックシステムが PLL で、AHB または APB2 サブシステムクロックが累積で 2 を超えて分周されていない場合のみ書き込み可能です。

- 0 : PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)
- 1 : PLL VCO 出力 (最大 144 MHz で動作)

注 : STM32F303xDxE および STM32F398xE のみ。

ビット 23:22 **UART5SW[1:0]** : UART5 クロックソース選択 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

このビットは、UART5 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00 : PCLK が UART5 クロックソースとして選択されます (デフォルト)。
- 01 : システムクロック (SYSCLK) が UART5 クロックとして選択されます。
- 10 : LSE クロックが UART5 クロックとして選択されます。
- 11 : HSI クロックが UART5 クロックとして選択されます。

ビット 21:20 **UART4SW[1:0]**: UART4 クロックソース選択 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

このビットは、UART4 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00: PCLK が UART4 クロックソースとして選択されます (デフォルト)。
- 01: システムクロック (SYSCLK) が UART4 クロックとして選択されます。
- 10: LSE クロックが UART4 クロックとして選択されます。
- 11: HSI クロックが UART4 クロックとして選択されます。

ビット 19:18 **USART3SW[1:0]**: USART3 クロックソースの選択

このビットは、USART3 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00: PCLK が USART3 クロックソースとして選択されます (デフォルト)。
- 01: システムクロック (SYSCLK) が USART3 クロックとして選択されます。
- 10: LSE クロックが USART3 クロックとして選択されます。
- 11: HSI クロックが USART3 クロックとして選択されます。

注: USART2SW[1:0] は STM32F303x6/8 および STM32F328x8 では使用できません。

ビット 17:16 **USART2SW[1:0]**: USART2 クロックソースの選択

このビットは、USART2 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00: PCLK が USART2 クロックソースとして選択されます (デフォルト)。
- 01: システムクロック (SYSCLK) が USART2 クロックとして選択されます。
- 10: LSE クロックが USART2 クロックとして選択されます。
- 11: HSI クロックが USART2 クロックとして選択されます。

注: USART2SW[1:0] は STM32F303x6/8 および STM32F328x8 では使用できません。

ビット 15 **TIM20SW**: Timer20 クロックソースの選択

TIM20 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

このビットは、システムクロックソースが PLL で、AHBまたはAPB2 サブシステムクロックが累積で 2 を超えて分周されていない場合にのみ書き込み可能です。

このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

- 0: PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)
- 1: PLL VCO 出力 (最大 144 MHz で動作)

注: STM32F303xDxE のみ。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **TIM17SW**: Timer17 クロックソースの選択

TIM17 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

このビットは、システムクロックソースが PLL で、AHBまたはAPB2 サブシステムクロックが累積で 2 を超えて分周されていない場合にのみ書き込み可能です。

このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

- 0: PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)
- 1: PLL VCO 出力 (最大 144 MHz で動作)

注: STM32F303xDxE および STM32F398xE デバイスのみ。

ビット 12 予約済みであり、リセット値に保持する必要があります。

**ビット 11 TIM16SW** : Timer16 クロックソースの選択

TIM16 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

このビットは、システムクロックソースが PLL で、AHBまたはAPB2 サブシステムクロックが累積で 2 を超えて分周されていない場合にのみ書き込み可能です。

このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

0 : PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)

1 : PLL VCO 出力 (最大 144 MHz で動作)

注 : STM32F303xD/E および STM32F398xE デバイスのみ。

**ビット 10 TIM15SW** : Timer15 クロックソースの選択

TIM15 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

このビットは、システムクロックソースが PLL および AHB で、APB2 サブシステムクロックが累積で 2 を超えて分周されていない場合にのみ書き込み可能です。

このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

0 : PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)

1 : PLL VCO 出力 (最大 144 MHz で動作)

注 : STM32F303xD/E および STM32F398xE デバイスのみ。

**ビット 9 TIM8SW** : タイマ 8 クロックソース選択 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

TIM8 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

STM32F303xB/C および STM32F358xC では、クロックシステムが PLL で、分周されていない AHB および APB2 のサブシステムクロックをそのクロックシステムで使用している場合にのみこのビットの書き込みが可能です。

STM32F303xD/E および STM32F398xE では、システムクロックソースが PLL で、AHB または APB2 サブシステムクロックが累積で 2 を超えて分周されていない場合にのみこのビットの書き込みが可能です。このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

0 : PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)

1 : PLL VCO 出力 (最大 144 MHz で動作)

**ビット 8 TIM1SW** : Timer1 クロックソースの選択

TIM1 クロックソースを選択するために、ソフトウェアによってセット/リセットされます。

STM32F303xB/C および STM32F358xC では、クロックシステムが PLL で、分周されていない AHB および APB2 のサブシステムクロックがクロックシステムに対応している場合にのみこのビットの書き込みが可能です。

STM32F303x6/8/D/E および STM32F398xE では、システムクロックソースが PLL で、AHB または APB2 サブシステムクロックが累積で 2 を超えて分周されていない場合にのみこのビットの書き込みが可能です。

このビットは、前回の状態を終了する際にハードウェアによってリセットされます (新しいスイッチが必要な場合は、もう一度ビットをセットする必要があります)。

0 : PCLK2 クロック (プリスケールされた場合は倍の周波数) (デフォルト)

1 : PLL VCO 出力 (最大 144 MHz で動作)

ビット 7 : 予約済みであり、リセット値に保持する必要があります。

**ビット 6 I2C3SW** : I2C3 クロックソース選択 (STM32F303xD/E デバイスのみ)

このビットは、I2C3 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : HSI クロック が I2C3 クロックソースとして選択されます (デフォルト)。

1 : SYSCLK クロックが I2C3 クロックとして選択されます。



ビット 5 **I2C2SW** : I2C2 クロックソース選択 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)

このビットは、I2C2 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : HSIクロック が I2C2 クロックソースとして選択されます (デフォルト)。
- 1 : SYSCLK クロックが I2C2 クロックとして選択されます。

ビット 4 **I2C1SW** : I2C1 クロックソースの選択

このビットは、I2C1 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : HSIクロック が I2C1 クロックソースとして選択されます (デフォルト)。
- 1 : SYSCLK クロックが I2C1 クロックとして選択されます。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **USART1SW[1:0]** : USART1 クロックソースの選択

このビットは、USART1 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00 : PCLK が USART1 クロックソースとして選択されます (デフォルト)。
- 01 : システムクロック (SYSCLK) が USART1 クロックとして選択されます。
- 10 : LSE クロックが USART1 クロックとして選択されます。
- 11 : HSI クロックが USART1 クロックとして選択されます。

## 9.4.14 RCC レジスタマップ

次の表に、RCC レジスタマップとリセット値を示します。

表 32. RCC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RCC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値							0	0					0	0	0	0	x	x	x	x	x	x	x	x	1	0	0	0	0	Res.	HSIRDY	HSION	
0x04	RCC_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	RCC_CIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値									0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	RCC_APB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	RCC_APB1RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	RCC_AHBENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	RCC_APB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	RCC_APB1ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



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

表 32. RCC レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x20	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC SEL [1:0]	Res.	Res.	Res.	Res.	LSE DRV [1:0]	Res.	LSEBYP	LSELDY	LSEON	
	リセット値																	0	0						0	0			1	1	0	0	0	
0x24	RCC_CSR	LPWRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	OBLRSTF	RMVF	V18PWRSTF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値	0	0	0	0	0	0	0	0	0																							0	0
0x28	RCC_AHBRSTR	Res.	Res.	ADC34RST <sup>(2)</sup>	ADC12RST	Res.	Res.	Res.	TSCRST	IOGPRST <sup>(3)</sup>	IOFPRST	IOPERST <sup>(2)</sup>	IODRST	IOPCRST	IOBRST	IOPARST	IOPHRST <sup>(3)</sup>	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCPRST <sup>(3)</sup>	Res.	Res.	Res.	Res.	
	リセット値			0	0				0	0	0	0	0	0	0	0	0											0						
0x2C	RCC_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x30	RCC_CFGR3	Res.	Res.	Res.	Res.	Res.	Res.	TIM34SW <sup>(3)</sup>	TIM2SW <sup>(3)</sup>	UART5SW[1:0] <sup>(2)</sup>	UART4SW[1:0] <sup>(2)</sup>	USART3SW[1:0]	USART2SW[1:0]	TIM20SW <sup>(3)</sup>	Res.	Res. (1)	Res.	Res. (1)	Res. (1)	Res. (1)	TIM8SW <sup>(2)</sup>	TIM1SW	Res.	I2C3SW <sup>(3)</sup>	I2C2SW	I2C1SW	Res.	Res.	USART1SW[1:0]	Res.	Res.	Res.	Res.	
	リセット値									0	0	0	0									0	0											

1. STM32F303xB/C および STM32F358xC デバイスのみ。
2. STM32F303x6/8 および STM32F328x8 デバイスのみ。
3. STM32F303xD/E デバイスのみ。

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 10 フレキシブルスタティックメモリコントローラ (FSMC)

**注：** *STM32F303xD/E およびSTM32F398xE デバイスにのみFSMCがあります。*

フレキシブルスタティックメモリコントローラ (FSMC) には、次の2つのメモリコントローラが含まれています。

- NOR/PSRAM メモリコントローラ
- NAND/PC カードメモリコントローラ

このメモリコントローラはフレキシブルメモリコントローラ (FMC) とも呼ばれています。

### 10.1 FMC の主な機能

FMC 機能ブロックは、同期および非同期のスタティックメモリ、16 ビット PC カードメモリとのインタフェースを構成します。主要目的は、

- AHB トランザクションを該当する外部デバイスプロトコルへ翻訳すること
- 外部メモリデバイスのアクセス時間条件を満たすようにすること

すべての外部メモリが、アドレス信号、データ信号、制御信号をコントローラと共用します。各外部デバイスを固有のチップセレクトを使ってアクセスします。FMC は外部デバイスへ一度に1回だけアクセスします。

FMC コントローラの主要機能は次の通りです。

- 次のスタティックメモリにマップされたデバイスとのインタフェース
  - スタティックランダムアクセスメモリ (SRAM)
  - NOR 型フラッシュメモリ/OneNAND フラッシュメモリ
  - PSRAM (4 メモリバンク)
  - 16 ビット PC カード互換デバイス
  - 最大 8 K バイトのデータをチェックできる ECC ハードウェア付きの 2 バンクの NAND フラッシュメモリ
- NOR フラッシュメモリ、PSRAM などの同期デバイスに対するアクセスを高速化するバーストモードのサポート
- 非同期アクセスと同期アクセス用のプログラム可能な連続クロック出力
- 8 ビット、16 ビット幅のデータバス
- 各メモリバンクに対する独立したチップセレクト制御
- メモリバンクごとに独立した設定
- PSRAM デバイス、SRAM デバイスで使用する書き込みイネーブルとバイトレーン選択出力
- 外部非同期ウェイト制御
- 16 x 33 ビット長の書き込みデータ FIFO
- 16 x 30 ビット長の書き込みアドレス FIFO

FMC は、16 x 33 ビット長の書き込みデータ FIFO と、16 x 30 ビット長の書き込みアドレス FIFO の2つの書き込み FIFO を内蔵。

- 書き込みデータ FIFO は、メモリへ書き込む AHB データ (最大 32 ビット) と、AHB 転送 (バーストモードまたは非シーケンシャルモード) 用の 1 ビットを格納

- 書き込みアドレス FIFO は、AHB アドレス (最大 28 ビット) と AHB データサイズ (最大 2 ビット) を格納。バーストモードで動作する場合、ページ境界 (PSRAM) を跨ぐとき以外は、開始アドレスのみを格納。この場合、AHBバーストは 2 つの FIFO エントリに分割されます。

開始時に、ユーザーアプリケーションから FMC ピンを設定する必要があります。アプリケーションで使用しない FMC I/O ピンは、他の用途に使用することができます。

外部デバイスタイプおよびその特性を定義する FMC レジスタは、通常、起動時に設定され、次のリセットまたは電源投入まで変更されません。ただし、設定値は何時でも変更することができます。

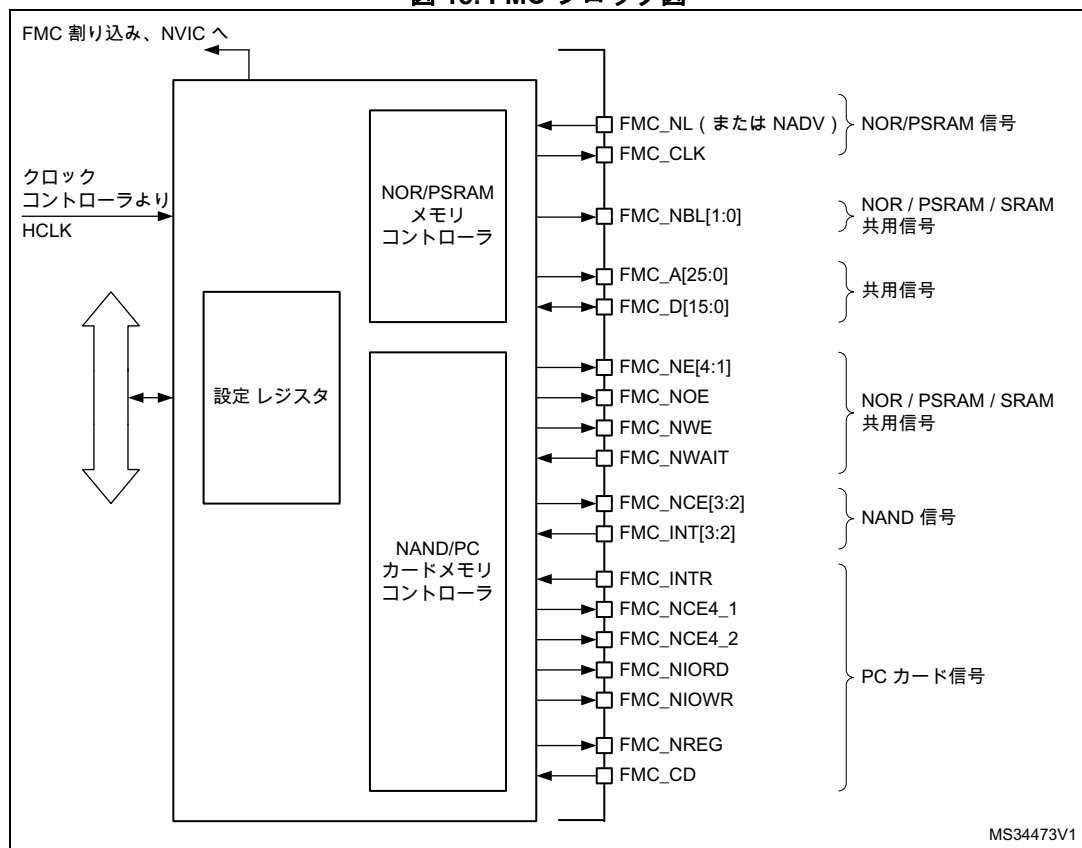
## 10.2 ブロック図

FMC は 次のメインブロックで構成されます。

- AHB インタフェース (FMC 設定レジスタを含む)
- NOR フラッシュ /PSRAM/SRAM コントローラ
- NAND 型フラッシュ /PC カードコントローラ
- 外部デバイスインタフェース

図 18 にブロック図を示します。

図 18. FMC ブロック図





## 10.3 AHB インタフェース

AHB スレーブインタフェースは、内部 CPU および他のバスマスタペリフェラルから外部スタティックメモリへのアクセスを可能にします。

AHB トランザクションは、外部デバイスプロトコルに変換されます。特に、選択された外部メモリが 16 または 8 ビット幅の場合、AHB 上の 32 ビット幅のトランザクションは、16 または 8 ビットの連続した複数のアクセスに分割されます。FMC チップセレクト (FMC\_NEX) は、拡張モードが有効になっている際にアクセスモード D の場合を除いて、複数の連続アクセス間にトグルしません。

FMC は、次の条件で AHB エラーを発生します。

- 有効化していない FMC バンク (バンク1~4) の読み出しまたは書き込みを行ったとき。
- FMC\_BCRx レジスタの FACCEN ビットがクリアされている状態で NOR フラッシュバンクの読み出しまたは書き込みを行ったとき。
- FMC\_CD (カード存在検出) 入力ピンがロー状態のとき、PC カードバンクの読み出しまたは書き込みを行ったとき。

この AHB エラーの影響は、読み出し/書き込みアクセスを試みた AHB マスタに依存します。

- アクセスが FPU CPU を持つ Cortex™-M4 から行われた場合は、ハードフォルト割り込みが発生します。
- アクセスが DMA コントローラから行われた場合は、DMA 転送エラーが発生し、対応する DMA チャンネルが自動的に無効化されます。

AHB クロック (HCLK) は、FMC のリファレンスクロックです。

### 10.3.1 サポートされるメモリおよびトランザクション

#### 一般的なトランザクション規則

リクエストされる AHB トランザクションのデータサイズは、アクセスされる外部デバイスのデータ幅が固定であっても、8、16、または 32 ビット幅です。このため、転送に矛盾が生じることがあります。

したがって、次のようないくつかの単純なトランザクション規則を守る必要があります。

- AHB トランザクションサイズとメモリデータサイズが等しい場合：
 

この場合は問題ありません。
- AHB トランザクションサイズがメモリサイズより大きい場合：
 

この場合、FMC は、AHB トランザクションを外部データ幅に合わせて、連続した小さなメモリアクセスに分割します。FMC チップセレクト (FMC\_NEX) は、複数の連続アクセス間にトグルしません。
- AHB トランザクションサイズがメモリサイズより小さい場合：
 

外部デバイスのタイプによっては、転送に一貫性がなくなる場合があります：

  - バイト選択機能を持つデバイス (SRAM、ROM、PSRAM) に対するアクセス
 

この場合、FMC は、読み出し/書き込みトランザクションを許可して、バイトレーン NBL[1:0] を通じて正しいデータにアクセスします。

書き込み対象バイトは NBL[1:0] によりアドレス指定されます。

すべてのメモリバイトが読み出され (読み出しトランザクション中 NBL[1:0] はローに駆動されます)、不要なバイトは無視されます。
  - バイト選択機能を持たないデバイス (NOR および NAND フラッシュメモリ) に対するアクセス
 

この状況は、16 ビット幅のフラッシュメモリへのバイトアクセスが要求されたときに発生します。バイトモードでデバイスにアクセスできないため (フラッシュメモリに対しては

16 ビットワードの読み出し/書き込みのみが可能)、書き込みトランザクションと読み出しトランザクションが可能です (コントローラは 16 ビットメモリワード全体を読み出して、必要なバイトのみを使用)。

## 設定レジスタ

FMC は、レジスタのセットとして設定することができます。NOR フラッシュ/PSRAM コントローラレジスタの詳細については、[セクション 10.5.6](#)を参照してください。NAND フラッシュ/PC カードレジスタの詳細については、[セクション 10.6.8](#)を参照してください。

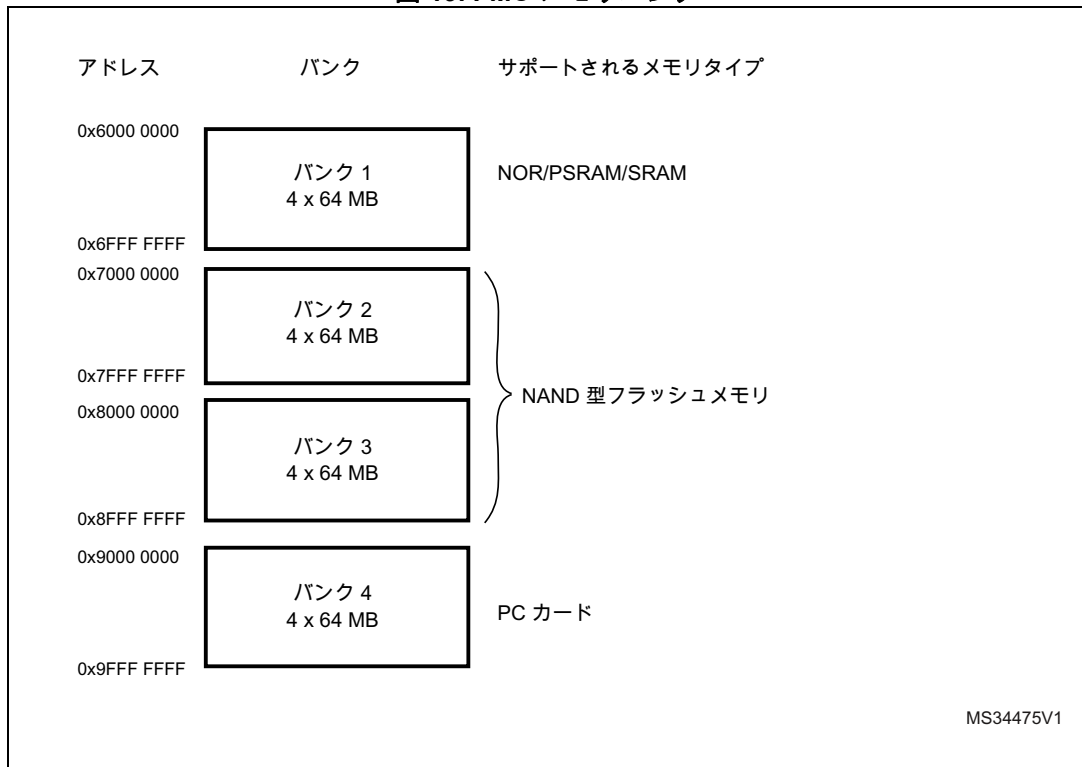
## 10.4 外部デバイスアドレスマッピング

FMC から見ると、外部メモリは、それぞれ 256 MB の固定サイズのバンクに分けられています ([図 19](#)を参照)。

- バンク 1 は、4 つまでの NOR フラッシュメモリまたは PSRAM メモリデバイスのアドレス指定に使用されます。このバンクは、4 つの専用チップセレクトを持つ 4 つの NOR/PSRAM サブバンクに分割されます。
  - バンク 1 - NOR/PSRAM 1
  - バンク 1 - NOR/PSRAM 2
  - バンク 1 - NOR/PSRAM 3
  - バンク 1 - NOR/PSRAM 4
- バンク 2 と 3 は、NAND フラッシュメモリデバイスのアドレス指定に使用されます (バンクごとに 1 つのデバイス)。
- バンク 4 は、PC カードのアドレス指定に使用されます。

各バンクで使用されるメモリのタイプは、ユーザアプリケーションから設定レジスタを使って設定することができます。

図 19. FMC メモリバンク



## 10.4.1 NOR/PSRAM アドレスマッピング

HADDR[27:26] ビットは、表 33 に示される 4 つのメモリバンクの 1 つを選択するために使用されます。

表 33. NOR/PSRAM バンク選択

HADDR[27:26] <sup>(1)</sup>	選択されるバンク
00	バンク 1 - NOR/PSRAM 1
01	バンク 1 - NOR/PSRAM 2
10	バンク 1 - NOR/PSRAM 3
11	バンク 1 - NOR/PSRAM 4

1. HADDR は、外部メモリに変換される内部 AHB アドレスラインです。

HADDR[25:0] ビットは、外部メモリアドレスを含みます。HADDR はバイトアドレスですが、メモリはワードレベルでアドレス指定されるので、メモリに対して実際に発行されるアドレスは、次の表に示されるように、メモリのデータ幅に応じて変わります。

表 34. NOR/PSRAM 外部メモリアドレス

メモリ幅 <sup>(1)</sup>	メモリに発行されるデータアドレス	最大メモリ容量 (ビット)
8 ビット	HADDR[25:0]	64 Mバイト * 8 = 512 M ビット
16 ビット	HADDR[25:1] >> 1	64 Mバイト / 2 * 16 = 512 M ビット

- 外部メモリが 16 ビット幅の場合、FSMC は、内部で HADDR[25:1] を使用して、外部メモリ FMC\_A[24:0] に対するアドレスを生成します。  
外部メモリの幅にかかわらず、FMC\_A[0] は外部メモリアドレス A[0] に接続されます。

## NOR 型フラッシュ/PSRAM のラップサポート

同期メモリに対するラップバーストモードはサポートされていません。メモリは、不定長のリニアバーストモードに構成する必要があります。

### 10.4.2 NAND フラッシュメモリ/PC カードアドレスマッピング

この場合 3 つのバンクが使用可能であり、それぞれが表 35 に示されるメモリ領域に分割されます。

表 35. NAND/PC カードメモリマッピングおよびタイミングレジスタ

開始アドレス	終了アドレス	FMC バンク	メモリ空間	タイミングレジスタ
0x9C00 0000	0x9FFF FFFF	バンク 4 - PC カード	I/O	FMC_PIO4 (0xB0)
0x9800 0000	0x9BFF FFFF		属性	FMC_PATT4 (0xAC)
0x9000 0000	0x93FF FFFF		共通	FMC_PMEM4 (0xA8)
0x8800 0000	0x8BFF FFFF	バンク 3 - NAND フラッシュ	属性	FMC_PATT3 (0x8C)
0x8000 0000	0x83FF FFFF		共通	FMC_PMEM3 (0x88)
0x7800 0000	0x7BFF FFFF	バンク 2 - NAND フラッシュ	属性	FMC_PATT2 (0x6C)
0x7000 0000	0x73FF FFFF		共通	FMC_PMEM2 (0x68)

NAND 型フラッシュメモリでは、共通および属性メモリ空間は、下位 256 KB に位置する 3 つのセクションに分割されます (表 36 を参照)。

- データセクション (共通/属性メモリ空間の最初の 64 KB)
- コマンドセクション (共通/属性メモリ空間の 2 番目の 64 KB)
- アドレスセクション (共通/属性メモリ空間の次の 128 KB)

表 36. NAND バンク選択

セクション名	HADDR[17:16]	アドレス範囲
アドレスセクション	1X	0x020000-0x03FFFF
コマンドセクション	01	0x010000-0x01FFFF
データセクション	00	0x000000-0x00FFFF

アプリケーションソフトウェアは、3 つのセクションを使用して、NAND 型フラッシュメモリにアクセスします。

- NAND フラッシュメモリにコマンドを送信するときは、ソフトウェアからコマンドセクションの任意のメモリ位置にコマンドの値を書き込む必要があります。**
- 読み出しまたは書き込みを行う必要がある NAND フラッシュのアドレスを指定するには、ソフトウェアは、アドレスセクションの任意のメモリ位置にアドレス値を書き込む必要があります。** アドレスは 4 または 5 バイト長なので (実際のメモリサイズに依存)、アドレス全体を指定するには、アドレスセクションへのいくつかの連続した書き込みが必要です。

- データの読み出しまたは書き込みを行うときは、ソフトウェアは、データセクションの任意のメモリ位置からデータを読み出すか、書き込む必要があります。

NAND 型フラッシュメモリはアドレスを自動的にインクリメントするので、連続したメモリ位置にアクセスするには、データセクションのアドレスをインクリメントする必要はありません。

## 10.5 NOR 型フラッシュ/PSRAM コントローラ

FMC は、以下のタイプのメモリを駆動するのに適した信号タイミングを発生します。

- 非同期 SRAM および ROM
  - 8 ビット
  - 16 ビット
- PSRAM (Cellular RAM)
  - 非同期モード
  - 同期アクセスに対するバーストモード
  - マルチプレクスまたは非マルチプレクス
- NOR フラッシュメモリ
  - 非同期モード
  - 同期アクセスに対するバーストモード
  - マルチプレクスまたは非マルチプレクス

FMC は、バンクごとに 1 つのチップセレクト信号 NE[4:1] を出力します。その他の信号 (アドレス、データ、および制御) はすべて、共有されます。

FMC は、次のようなプログラム可能なタイミングにより、広範囲なデバイスをサポートしています。

- プログラム可能なウェイトステート (15 まで)
- プログラム可能なバスターンアラウンドサイクル (15 まで)
- プログラム可能な出カインエーブルおよび書き込みインエーブル遅延 (15 まで)
- 独立した読み出しおよび書き込みタイミングとプロトコルにより、広範囲なメモリおよびタイミングをサポート
- プログラム可能な連続クロック (FMC\_CLK) 出力

FMC クロック (FMC\_CLK) は HCLK クロックの約数。このクロックは、同期アクセス中にのみ、または FMC\_BCR1 レジスタの CCKEN ビット設定に応じて非同期アクセスおよび同期アクセス中に、選択した外部デバイスへ供給することができます。

- CCLKEN ビットがリセットされている場合、FMC は同期アクセス (読み出し/書き込みトランザクション) 中にのみクロック (CLK) を生成します。
- CCLKEN ビットがセットされている場合、FMC は非同期アクセス中および同期アクセス中に連続クロックを生成します。FMC\_CLK 連続クロックを生成するためには、バンク 1 を同期モードに設定する必要があります ([セクション 10.5.6 : NOR/PSRAM コントローラレジスタ](#)を参照)。すべての同期メモリに対して同じクロックが使用されるため、連続出カクロックを生成して同期アクセスを実行する場合、AHB データサイズはメモリデータ幅 (MWID) に一致する必要があります。そうしないと、AHB データトランザクションに応じて、FMC\_CLK 周波数が変化します (FMC\_CLK 分周比の式については、[セクション 10.5.5 : 同期トランザクション](#)を参照)。

各バンクのサイズは 64 MB に固定されます。各バンクは、専用レジスタによって設定されます ([セクション 10.5.6 : NOR/PSRAM コントローラレジスタ](#)を参照)。

プログラム可能なメモリパラメータは、アクセスタイミング (表 37を参照) を含み、ウェイト管理をサポートします (バーストモードでの PSRAM および NOR フラッシュへのアクセス)。

表 37. プログラム可能な NOR/PSRAM のアクセスパラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
アドレスセットアップ	アドレスセットアップフェーズ時間	非同期	AHB クロック サイクル (HCLK)	0	15
アドレスホールド	アドレスホールドフェーズ時間	非同期、マルチプレクス I/O	AHB クロック サイクル (HCLK)	1	15
データセットアップ	データセットアップフェーズ時間	非同期	AHB クロック サイクル (HCLK)	1	256
バスターン	バスターンアラウンドフェーズ時間	非同期および同期読み出し/書き込み	AHB クロック サイクル (HCLK)	0	15
クロック分周比	1つのメモリクロックサイクル (CLK) に必要な AHB クロック サイクル (HCLK) 数	同期	AHB クロック サイクル (HCLK)	2	16
データ遅延	バーストの最初のデータの前にメモリに発行されるクロックサイクル数	同期	メモリクロック サイクル (CLK)	2	17

## 10.5.1 外部メモリインタフェース信号

表 38、表 39 および 表 40 に、NOR フラッシュメモリ、SRAM、および PSRAM へのインタフェースで一般に使用される信号を示します。

注: 接頭辞「N」は、アクティブローの信号を表します。

### NOR フラッシュメモリ、非マルチプレクス I/O

表 38. 非マルチプレクス I/O NOR フラッシュメモリ

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:0]	O	アドレスバス
D[15:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインエーブル
NWE	O	書き込みインエーブル
NL(= NADV)	O	ラッチインエーブル (この信号は、一部の NOR 型フラッシュデバイスではアドレス有効 (NADV) と呼ばれます)。
NWAIT	I	FMC への NOR フラッシュウェイト入力信号

最大容量は、512 M ビット (26 アドレスライン) です。

## NOR フラッシュメモリ、16 ビットマルチプレクス I/O

表 39. 16 ビットマルチプレクス I/O NOR フラッシュメモリ

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4
NOE	O	出カインーブル
NWE	O	書き込みインーブル
NL(= NADV)	O	ラッチインーブル (この信号は、一部の NOR 型フラッシュデバイスではアドレス有効 (NADV) と呼ばれます。)
NWAIT	I	FMC への NOR フラッシュウェイト入力信号

最大容量は 512 M ビットです。

## PSRAM/SRAM、非マルチプレクス I/O

表 40. 非マルチプレクス I/O PSRAM/SRAM

FMC 信号名	I/O	機能
CLK	O	クロック (PSRAM 同期アクセス専用)
A[25:0]	O	アドレスバス
D[15:0]	I/O	双方向データバス
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (Cellular RAM すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出カインーブル
NWE	O	書き込みインーブル
NL(= NADV)	O	PSRAM 入力の場合のみアドレス有効 (メモリ信号名 : NADV)
NWAIT	I	FMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出カバイト 0 およびバイト 1 制御 (上位および下位バイトインーブル)

最大容量は 512 M ビットです。

## PSRAM、16 ビットマルチプレクス I/O

表 41. 16 ビットマルチプレクス I/O PSRAM

FMC 信号名	I/O	機能
CLK	O	クロック (同期アクセス用)
A[25:16]	O	アドレスバス
AD[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス (16 ビットアドレス A[15:0] とデータ D[15:0] がデータバスでマルチプレクスされています)
NE[x]	O	チップセレクト、x = 1..4 (PSRAM (Cellular RAM すなわち CRAM) では NCE と呼ばれます。)
NOE	O	出力イネーブル
NWE	O	書き込みイネーブル
NL(= NADV)	O	アドレス有効 PSRAM 入力 (メモリ信号名: NADV)
NWAIT	I	FMC への PSRAM ウェイト入力信号
NBL[1:0]	O	バイトレーン出力バイト 0 およびバイト 1 制御 (上位および下位バイトイネーブル)

最大容量は、512 M ビット (26 アドレスライン) です。

### 10.5.2 サポートされるメモリおよびトランザクション

以下の表 42 に、NOR フラッシュメモリ、PSRAM、SRAM に対してメモリデータバスが 16 ビット幅の場合にサポートされているデバイス、アクセスモード、トランザクションの例を示します。FMC では許容されない (すなわち非サポート) トランザクションは、この例で灰色表示しています。

表 42. NOR フラッシュ /PSRAM : サポートされているメモリとトランザクションの例

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	許可/ 禁止	コメント
NOR 型 フラッシュ (マルチプレクス I/O と非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割
	非同期 ページ	R	-	16	N	モードはサポートされていません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-



表 42. NOR フラッシュ /PSRAM : サポートされているメモリとトランザクションの例 (続き)

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	許可/ 禁止	コメント
PSRAM (マルチプレクス I/O および 非マルチ プレクス I/O)	非同期	R	8	16	Y	-
	非同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割
	非同期 ページ	R	-	16	N	モードはサポートされていません。
	同期	R	8	16	N	-
	同期	R	16	16	Y	-
	同期	R	32	16	Y	-
	同期	W	8	16	Y	バイトレーン NBL[1:0] の使用
	同期	W	16/32	16	Y	-
SRAM と ROM	非同期	R	8/16	16	Y	-
	非同期	W	8/16	16	Y	バイトレーン NBL[1:0] の使用
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割、 バイトレーン NBL[1:0] の使用

## 10.5.3 一般的なタイミング規則

### 信号の同期

- すべてのコントローラ出力信号は、内部クロック (HCLK) の立ち上がりエッジで変化します。
- 同期モード (読み出しまたは書き込み) では、すべての出力信号が HCLK の立ち上がりエッジで変化します。CLKDIV の値によらず、すべての信号は次のように変化します。
  - NOEL/NWEL/ NEL/NADVL/ NADVH /NBLL/ アドレスの有効出力は、FMC\_CLK クロックの立ち下がりエッジで変化します。
  - NOEH/ NWEH/ NEH/ NOEH/NBLH/ アドレスの無効出力は、FMC\_CLK クロックの立ち上がりエッジで変化します。



## 10.5.4 NOR フラッシュ /PSRAM コントローラ非同期トランザクション

### 非同期スタティックメモリ (NOR フラッシュ、PSRAM、SRAM)

- 信号は、内部クロック HCLK によって同期されます。このクロックはメモリには発行されません。
- FMC は、NOE 信号をネゲートする前に、常にデータをサンプリングします。これにより、メモリのデータホールドタイミングに関する制約が満たされます (チップイネーブルハイからデータ遷移までの最小時間は通常 0 ns です)。
- 拡張モードを有効にした場合 (FMC\_BCRx レジスタの EXTMOD ビットをセット)、最大 4 つの拡張モード (A、B、C、D) が使用できます。書き込み動作および読み出し動作で、A、B、C、D モードをミックスすることができます。例えば、読み出し動作をモード A で実行し、書き込み動作をモード B で実行することができます。
- 拡張モードを無効にした場合 (FMC\_BCRx レジスタで EXTMOD ビットをリセット)、FMC は次のようにモード 1 またはモード 2 で動作することができます。
  - SRAM/PSRAM メモリタイプを選択した場合 (FMC\_BCRx レジスタで MTYP = 0x0 または 0x01)、モード 1 がデフォルトモードです。
  - NOR メモリタイプを選択した場合 (FMC\_BCRx レジスタで MTYP = 0x10)、モード 2 がデフォルトモードです。

### モード 1 - SRAM/PSRAM (CRAM)

次の図に、サポートされているモードに対する読み出しトランザクションと書き込みトランザクションを示します。それに続いて、FMC\_BCRx レジスタ、および FMC\_BTRx/FMC\_BWTRx レジスタの必要とされる設定を示します。

図 20. モード 1 読み出しアクセス波形

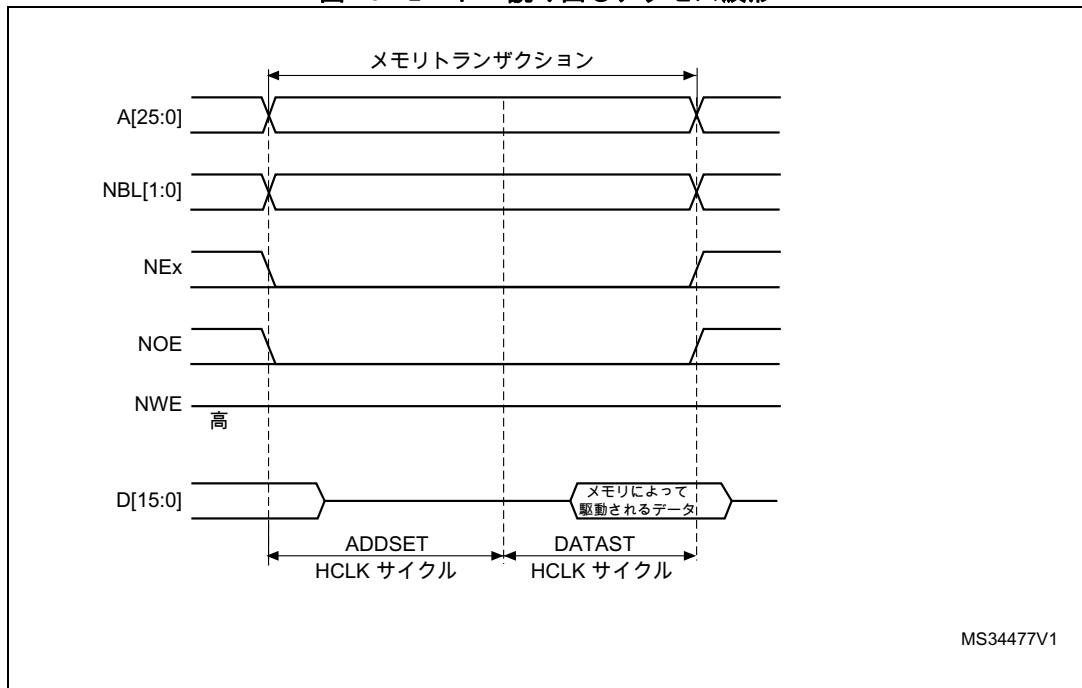
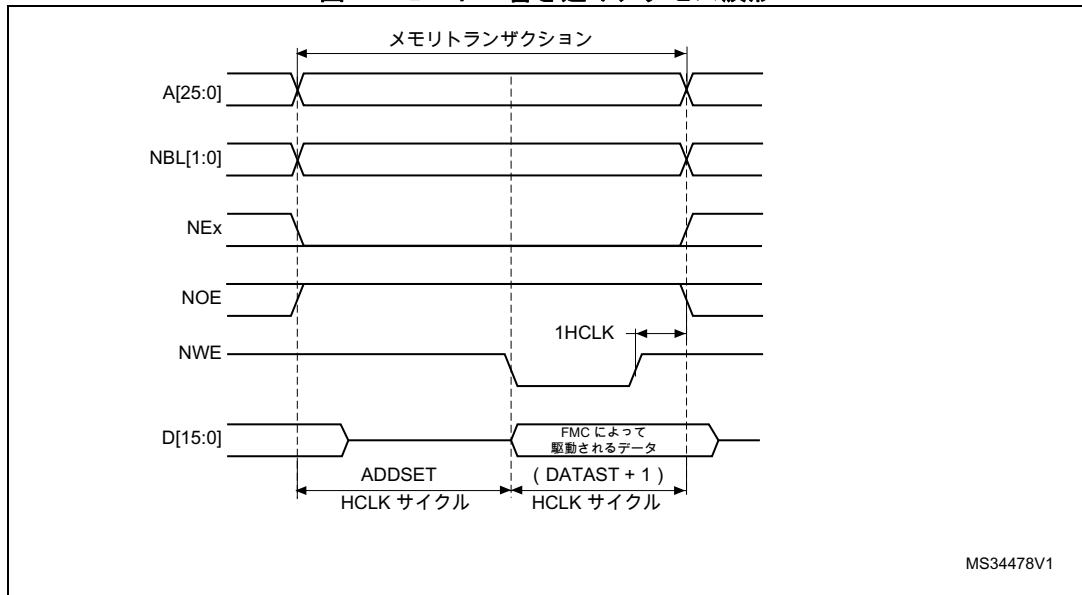


図 21. モード 1 書き込みアクセス波形



書き込みトランザクションの最後の 1 HCLK サイクルは、NWE の立ち上がりエッジ後のアドレスおよびデータホールド時間の保証に役立ちます。この HCLK サイクルがあるため、DATAST の値はゼロより大きくなければなりません (DATAST > 0)。

表 43. FMC\_BCRx ビットフィールド

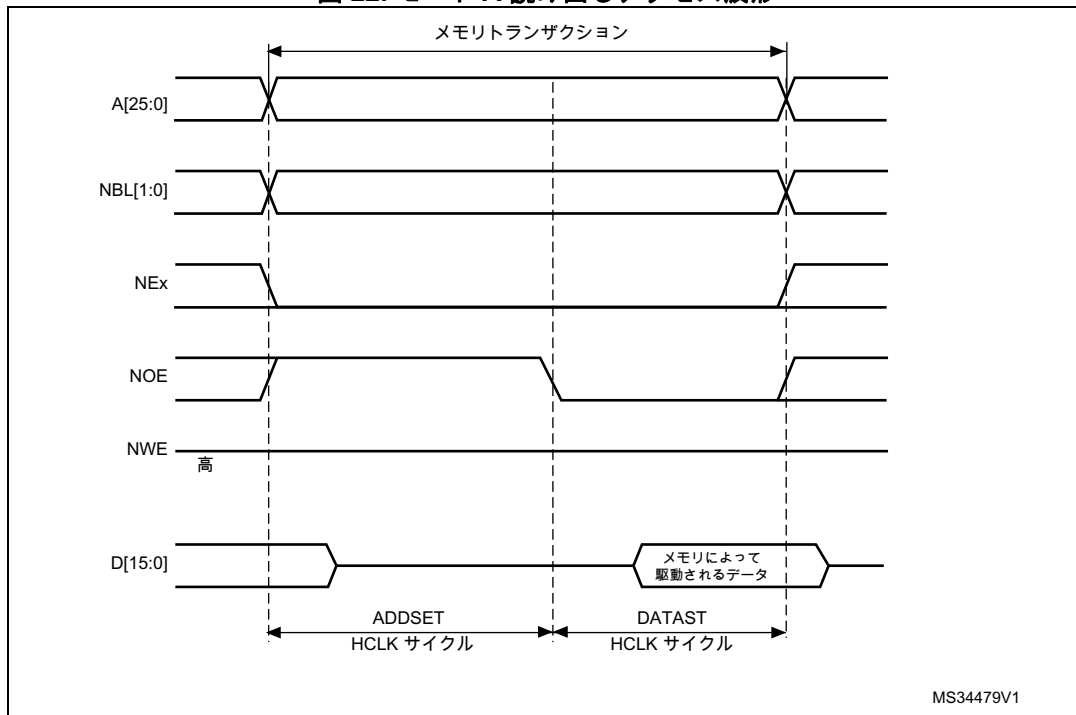
ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	予約済み	0x0
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5-4	MWID	必要に応じて設定します。
3-2	MTYP	必要に応じて、0x2 (NOR フラッシュメモリ) を除きます。
1	MUXE	0x0
0	MBKEN	0x1

表 44. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	無視
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	2 番目のアクセスフェーズの時間 (書き込みアクセスの場合は DATAST + 1 HCLK サイクル、読み出しアクセスの場合は DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

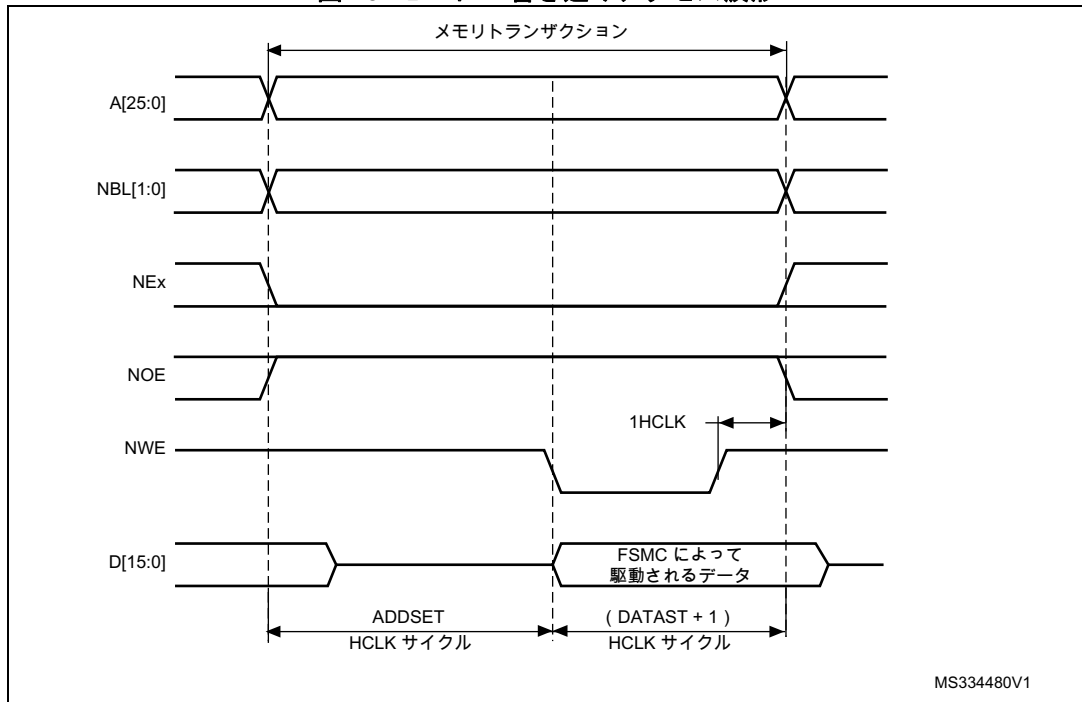
## モード A - SRAM/PSRAM (CRAM) OE トグル

図 22. モード A 読み出しアクセス波形



1. NBL[1:0] は読み出しアクセス中ローレベルに駆動されます。

図 23. モード A 書き込みアクセス波形



MS334480V1

モード 1 と比較した場合の相違点は、NOE のトグルリングと、独立した読み出しおよび書き込みのタイミングです。

表 45. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	予約済み	0x0
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	無視
5-4	MWID	必要に応じて設定します。
3-2	MTYP	必要に応じて、0x2 (NOR フラッシュメモリ) を除きます。

表 45. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
1	MUXEN	0x0
0	MBKEN	0x1

表 46. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

表 47. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。 ADDSET の最小値は 0。

## モード 2/B - NOR 型フラッシュ

図 24. モード 2 およびモード B 読み出しアクセス波形

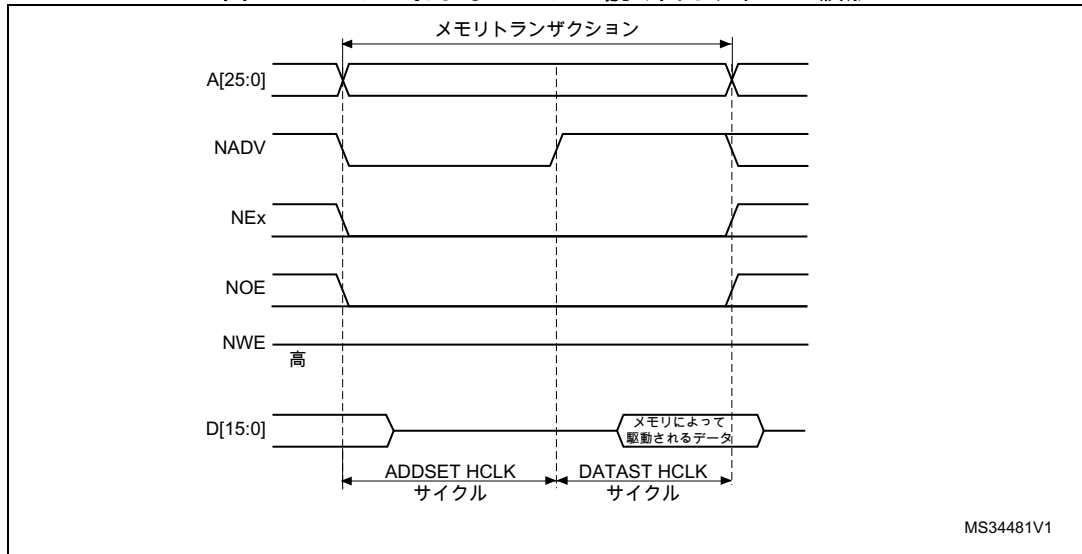


図 25. モード 2 書き込みアクセス波形

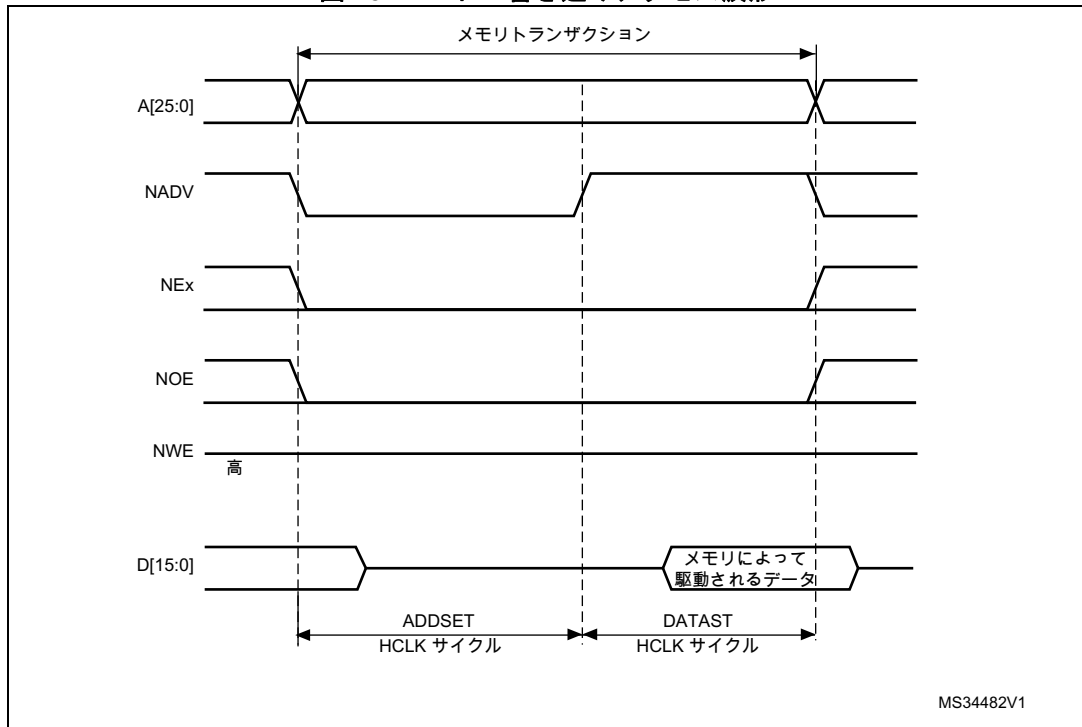
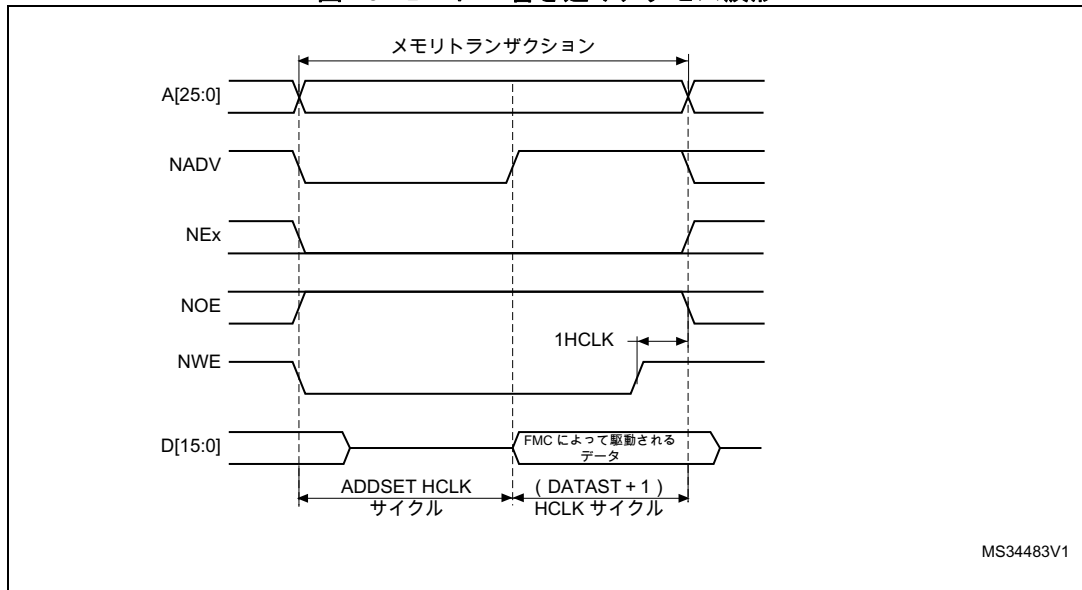


図 26. モード B 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NWE のトグルリングと拡張モードがセットされたとき（モード B）の独立した読み出しおよび書き込みのタイミングです。

表 48. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0（非同期モードで無効）
18:16	予約済み	0x0
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	モード B では 0x1、モード 2 では 0x0
13	WAITEN	0x0（非同期モードで無効）
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x2（NOR 型フラッシュメモリ）
1	MUXEN	0x0
0	MBKEN	0x1



表 49. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	拡張モードがセットされている場合は 0x1
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAS	読み出しアクセスに対する 2 番目のアクセスフェーズの時間 (DATAS HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 50. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	拡張モードがセットされている場合は 0x1
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAS	書き込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAS HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

注: FMC\_BWTRx レジスタは、拡張モードがセットされている場合 (モード B) のみ有効です。それ以外の場合には、その内容は意味を持ちません。

## モード C - NOR フラッシュ - OE トグリング

図 27. モード C 読み出しアクセス波形

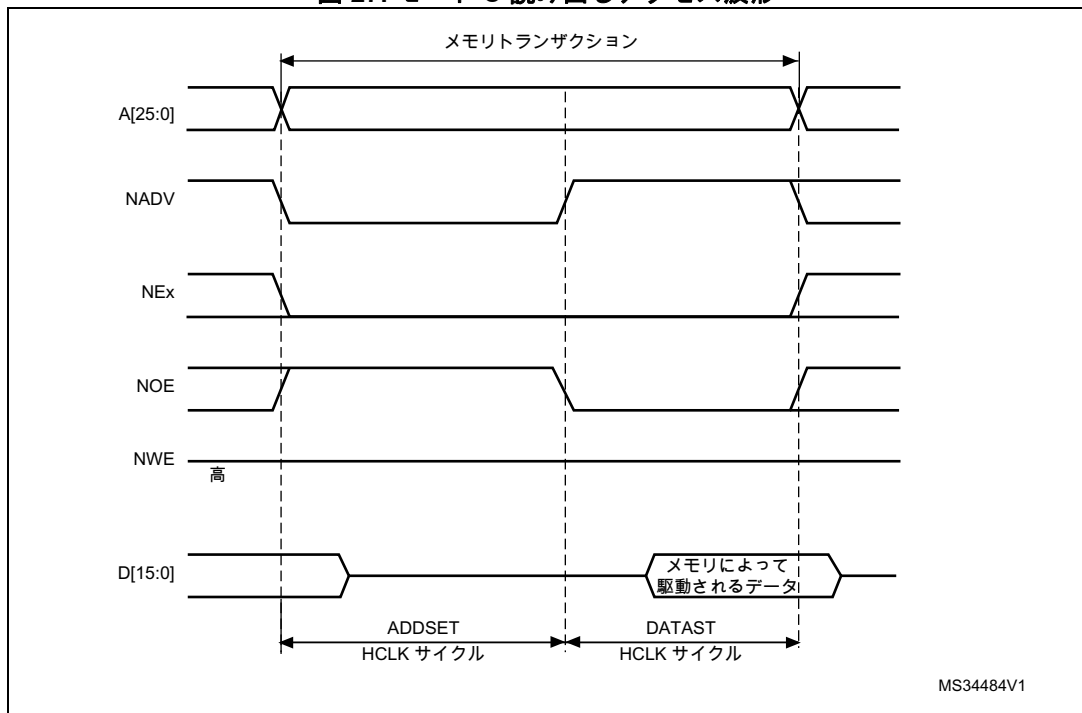
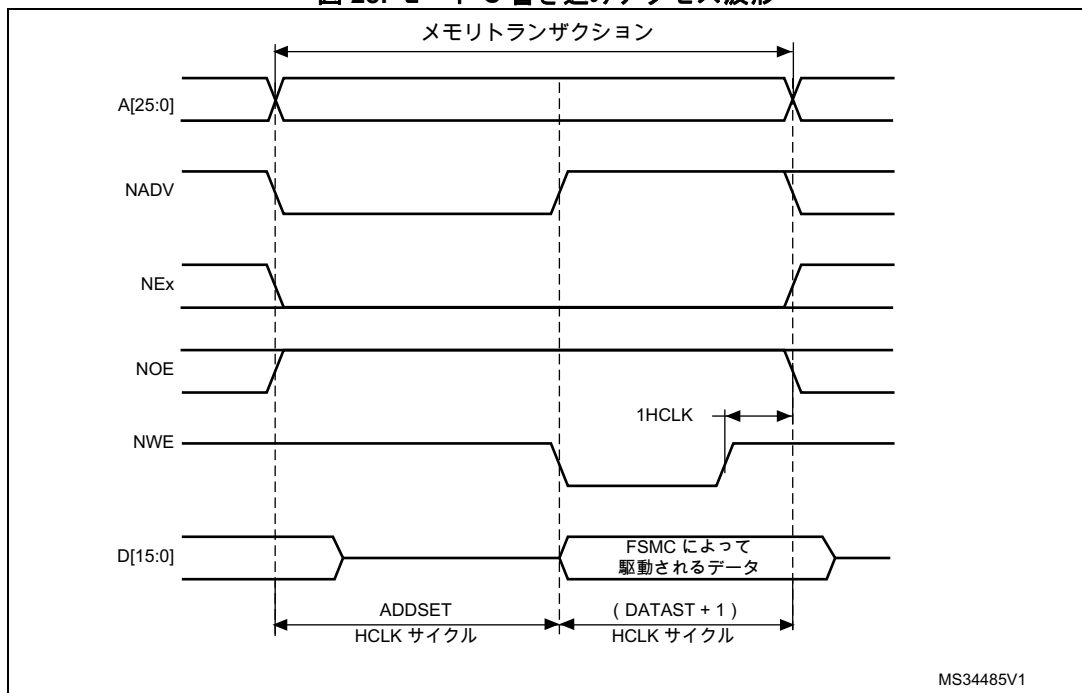


図 28. モード C 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NOE のトグリングと、独立した読み出しおよび書き込みのタイミングです。

表 51. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	予約済み	0x0
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x02 (NOR 型フラッシュメモリ)
1	MUXEN	0x0
0	MBKEN	0x1

表 52. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	0x0
23-20	CLKDIV	0x0
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	無視
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

表 53. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29:28	ACCMOD	0x2
27:24	DATLAT	無視
23:20	CLKDIV	無視
19:16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15:8	DATAST	書き込みに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7:4	ADDHLD	無視
3:0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 0。

## モード D - 拡張アドレスによる非同期アクセス

図 29. モード D 読み出しアクセス波形

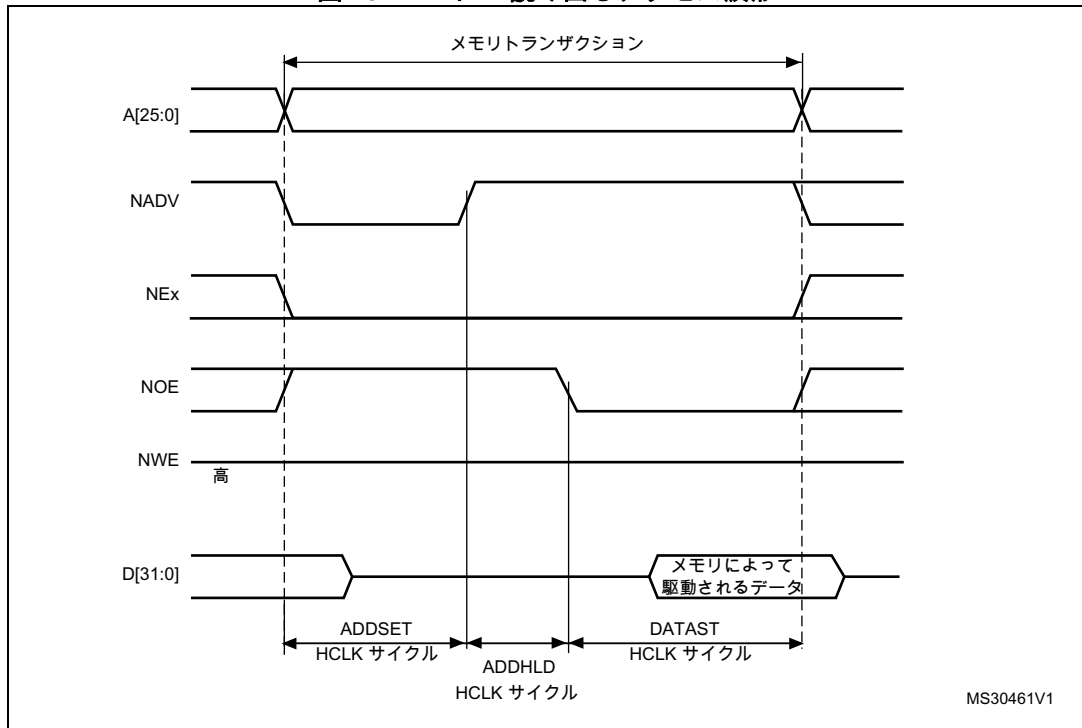
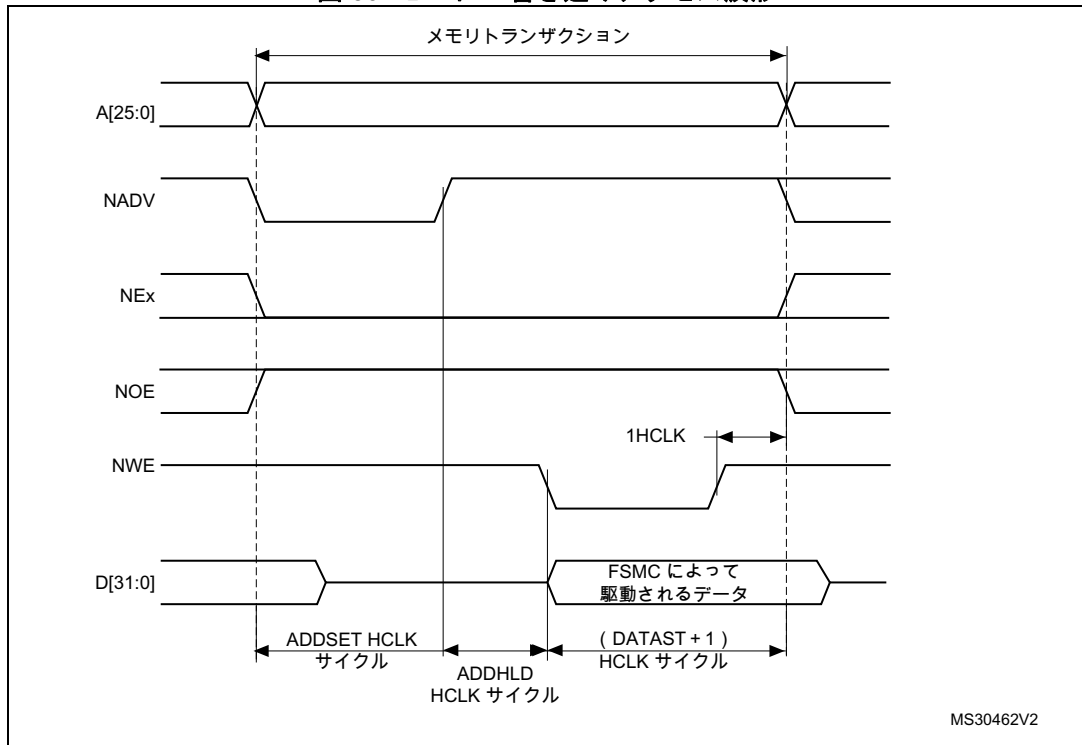


図 30. モード D 書き込みアクセス波形



モード 1 と比較した場合の相違点は、NADV が変化した後にはトグルを続ける NOE のトグルと独立した読み出し書き込みタイミングです。

表 54. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	予約済み	0x0
15	ASYNCWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x1
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5-4	MWID	必要に応じて設定します。

表 54. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
3-2	MTYP	必要に応じて設定します。
1	MUXEN	0x0
0	MBKEN	0x1

表 55. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	読み出しに対する 2 番目のアクセスフェーズの時間 (DATAST HCLK サイクル)。
7-4	ADDHLD	読み出しにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3-0	ADDSET	読み出しに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

表 56. FMC\_BWTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	書き込みアクセスに対する 2 番目のアクセスフェーズの時間 (DATAST1 HCLK サイクル)
7-4	ADDHLD	書き込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)
3-0	ADDSET	書き込みアクセスに対する最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。

## マルチプレクスモード - NOR 型フラッシュメモリに対するマルチプレクス非同期アクセス

図 31. マルチプレクス読み出しアクセス波形

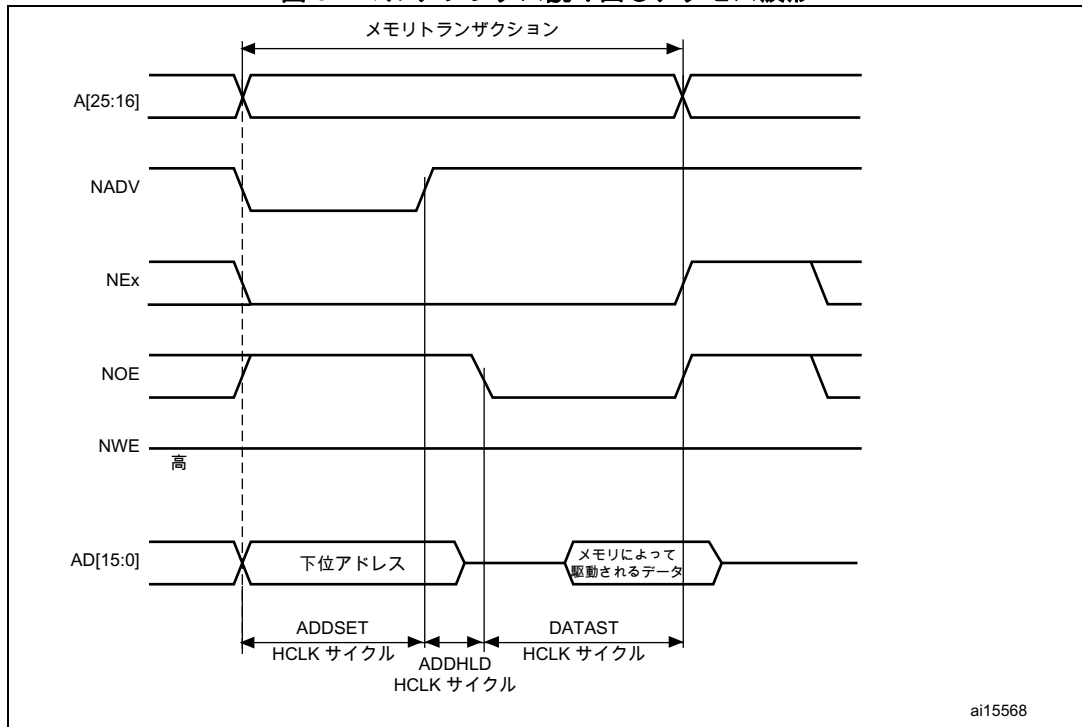
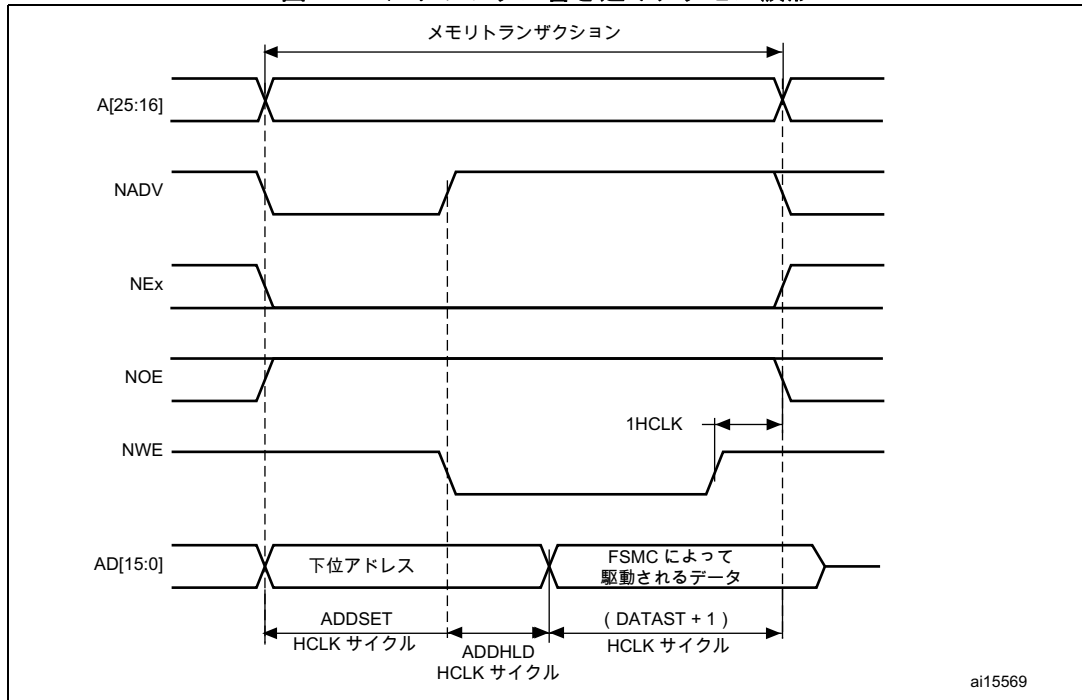


図 32. マルチプレクス書き込みアクセス波形



モード D との相違点は、データバス上の下位アドレスバイトのドライブです。

表 57. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x0 (非同期モードで無効)
18:16	予約済み	0x0
15	ASYNCAWAIT	メモリがこの機能をサポートする場合は 1 にセットします。そうでない場合は、0 のままにしておきます。
14	EXTMOD	0x0
13	WAITEN	0x0 (非同期モードで無効)
12	WREN	必要に応じて設定します。
11	WAITCFG	無視
10	WRAPMOD	0x0
9	WAITPOL	ビット 15 が 1 の場合のみ、意味を持ちます。
8	BURSTEN	0x0
7	予約済み	0x1
6	FACCEN	0x1
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x2 (NOR 型フラッシュメモリ)
1	MUXEN	0x1
0	MBKEN	0x1

表 58. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	無視
23-20	CLKDIV	無視
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DAST	2 番目のアクセスフェーズの時間 (読み出しアクセスの場合は DAST HCLK サイクル、書き込みアクセスの場合は DAST+1 HCLK サイクル)。
7-4	ADDHLD	書き込みアクセスにおける中間アクセスフェーズの時間 (ADDHLD HCLK サイクル)。
3-0	ADDSET	最初のアクセスフェーズの時間 (ADDSET HCLK サイクル)。ADDSET の最小値は 1。



## 非同期アクセスにおけるウェイト管理

非同期メモリが、データの受け取りまたは提供の準備がまだできていないことを知らせるために、WAIT 信号をアサートした場合、FMC\_BCRx レジスタの ASYNCWAIT ビットをセットする必要があります。

WAIT 信号がアクティブな場合 (WAITPOL ビットに依存して、ハイまたはロー)、DATAST ビットによってプログラムされる 2 番目のアクセスフェーズ (データセットアップフェーズ) は、WAIT ビットが非アクティブになるまで延長されます。データセットアップフェーズとは異なり、ADDSET および ADDHLD ビットによってプログラムされる最初のアクセスフェーズ (アドレスセットアップおよびアドレスホールドフェーズ) は、WAIT に対応しないので、延長されません。

データセットアップフェーズは、メモリトランザクションが終了する 4 HCLK サイクル前に WAIT が検出されるようにプログラムする必要があります。次のようなケースを考慮する必要があります。

1. メモリが、次のようにトグルする NOE/NWE に合わせた WAIT 信号をアサートする場合。

$$\text{DATAST} \geq (4 \times \text{HCLK}) + \text{max\_wait\_assertion\_time}$$

2. メモリが、NEx (またはトグルしない NOE/NWE) に合わせた WAIT 信号をアサートする場合。  
次の条件が成立するならば、

$$\text{max\_wait\_assertion\_time} > \text{address\_phase} + \text{hold\_phase}$$

以下の通りです。

$$\text{DATAST} \geq (4 \times \text{HCLK}) + (\text{max\_wait\_assertion\_time} - \text{address\_phase} - \text{hold\_phase})$$

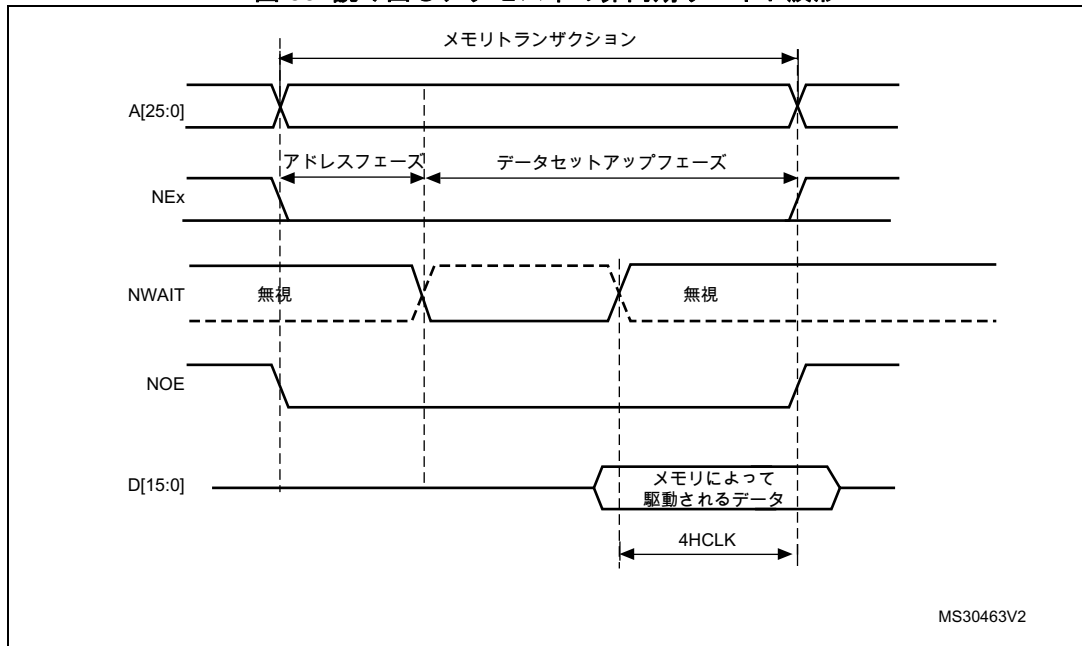
それ以外の場合は、

$$\text{DATAST} \geq 4 \times \text{HCLK}$$

ここで、max\_wait\_assertion\_time は、NEx/NOE/NWE がローになったときにメモリが WAIT 信号をアサートするために必要な最大時間です。

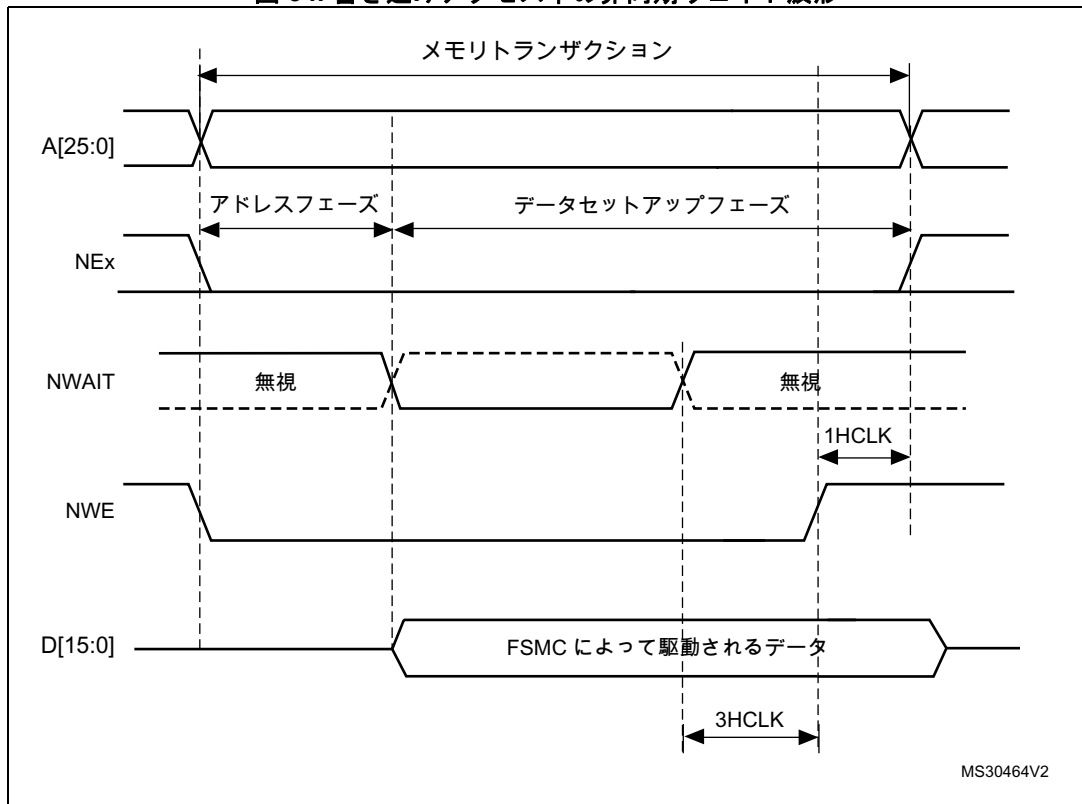
図 33 と 図 34 に、非同期メモリによって WAIT が解除された後、メモリアクセスフェーズに追加される HCLK クロックのサイクル数を示します (上記のケースに無関係)。

図 33. 読み出しアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FMC\_BCRx レジスタの WAITPOL ビット設定値に依存します。

図 34. 書き込みアクセス中の非同期ウェイト波形



1. NWAIT 極性は、FMC\_BCRx レジスタの WAITPOL ビット設定値に依存します。

## 10.5.5 同期トランザクション

メモリクロック FMC\_CLK は、HCLK の約数です。この値は CLKDIV 値と MWID/ AHB データサイズに依存し、次式で与えられます。

$$\text{FMC\_CLK divider ratio} = \max(\text{CLKDIV} + 1, \text{MWID}(\text{AHB data size}))$$

WID サイズが16 ビットまたは 8 ビットの場合、FMC\_CLK の分周比は常に、設定された CLKDIV 値で決まります。

例：

- CLKDIV = 1、MWID = 16 ビット、AHB データサイズ = 8 ビットの場合、FMC\_CLK = HCLK / 2

NOR 型フラッシュメモリは、NADV のアサーションから CLK がハイになるまでの最小時間を指定します。この制約を満たすために、FMC は、同期アクセスの最初の内部クロックサイクルでは (NADV アサーションの前)、クロックをメモリに供給しません。これにより、メモリクロックの立ち上がりエッジは、NADV ローパルスの中間で発生します。

### データ遅延と NOR メモリ遅延

データ遅延は、データをサンプリングする前のウェイトサイクル数です。DATLAT の値は、NOR 型フラッシュ設定レジスタで指定された遅延の値と一致しなければなりません。FMC は、NADV がローのときのクロックサイクルをデータ遅延カウントに含めません。

**注意：** 一部の NOR フラッシュメモリは、データ遅延カウントに NADV ローサイクルを含めるので、NOR フラッシュの遅延と FMC DATLAT パラメータの厳密な関係は、次のいずれかです。

- NOR 型フラッシュ遅延 = (DATLAT + 2) CLK クロックサイクル、または
- NOR 型フラッシュ遅延 = (DATLAT + 3) CLK クロックサイクル

最近のメモリの中には、遅延フェーズにおいて NWAIT をアサートするものがあります。そのような場合には、DATLAT を最小値に設定することができます。結果として、FMC はデータをサンプリングして、データが有効であるかどうか評価するのに十分な時間、ウェイトします。このように、FMC は、メモリが遅延から抜けたことを検出し、実際のデータが処理されます。

その他のメモリは、遅延中には NWAIT をアサートしません。この場合、FMC とメモリの両方について、遅延が正しく設定されなければなりません。そうしないと、正しいデータの代わりに無効なデータが取られたり、有効なデータがメモリサイクルの初期フェーズで失われることとなります。

### シングルバースト転送

選択されたバンクが同期アクセスのバーストモードに設定されている場合、例えば、16 ビットメモリで AHB シングルバーストトランザクションがリクエストされると、FMC は長さ 1 (AHB 転送が 16 ビットの場合) または 2 (AHB 転送が 32 ビットの場合) のバーストトランザクションを実行して、最後のデータがストロープされると、チップセレクトをデアサートします。

このような転送はサイクル数の点で、非同期読み出し動作に比較して最も効率的ではありません。しかし、時間がかかるメモリアクセスモードの再プログラムには、まず、ランダム非同期アクセスが必要です。

### ウェイト管理

同期 NOR フラッシュメモリでは、NWAIT がプログラムされた遅延期間 ((DATLAT+2) CLK サイクルに対応) 後に評価されます。

NWAIT がアクティブの場合 (WAITPOL = 0 ではローレベル、WAITPOL = 1 ではハイレベル)、ウェイト状態は NWAIT が非アクティブ (WAITPOL = 0 ではハイレベル、WAITPOL = 1 ではローレベル) になるまでが挿入されます。



NWAIT が非アクティブのときには、データはただちに（ビット WAITCFG=1）または次のクロックエッジで（ビット WAITCFG=0）、有効とみなされます。

NWAIT 信号によるウェイトステートの挿入時には、コントローラはメモリへのクロックパルスの送信を続け、チップセレクトと出カインーブル信号を維持します。データは有効と見なしません。

バーストモードでの NOR フラッシュの NWAIT 信号には、2 つのタイミング設定があります。

- フラッシュメモリは、ウェイトステートの 1 データサイクル前に NWAIT 信号をアサートします（リセット後のデフォルト）。
- フラッシュメモリは、ウェイトステートの間に NWAIT 信号をアサートします。

FMC は FMC\_BCRx レジスタ (x = 0..3) の WAITCFG ビットを使って、各チップセレクトに対する両 NOR フラッシュウェイトステート設定をサポートします。

図 35. 待ち設定波形

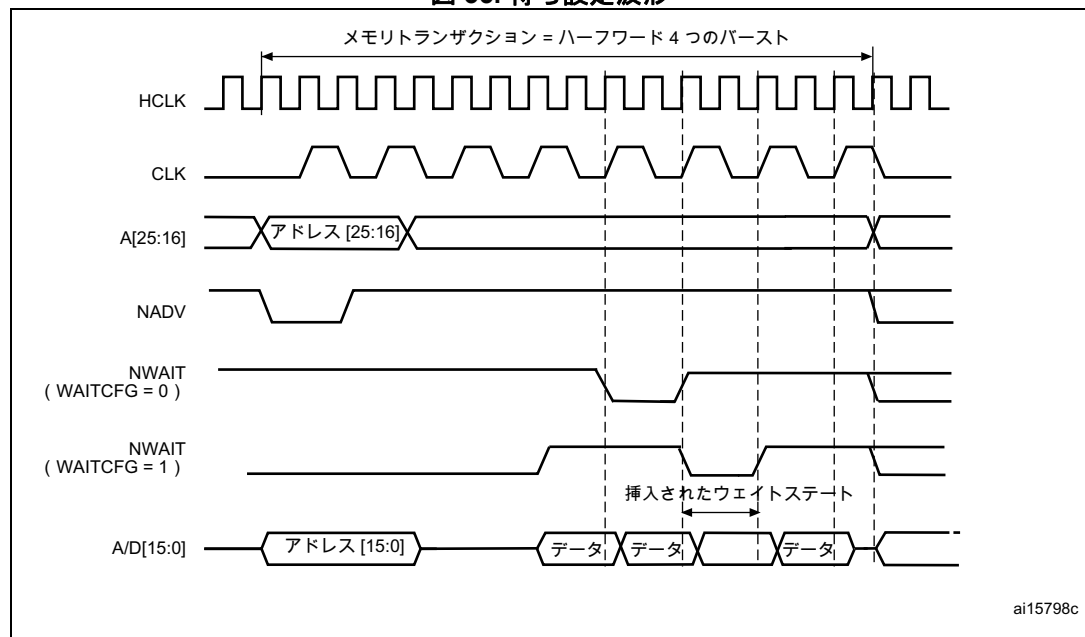
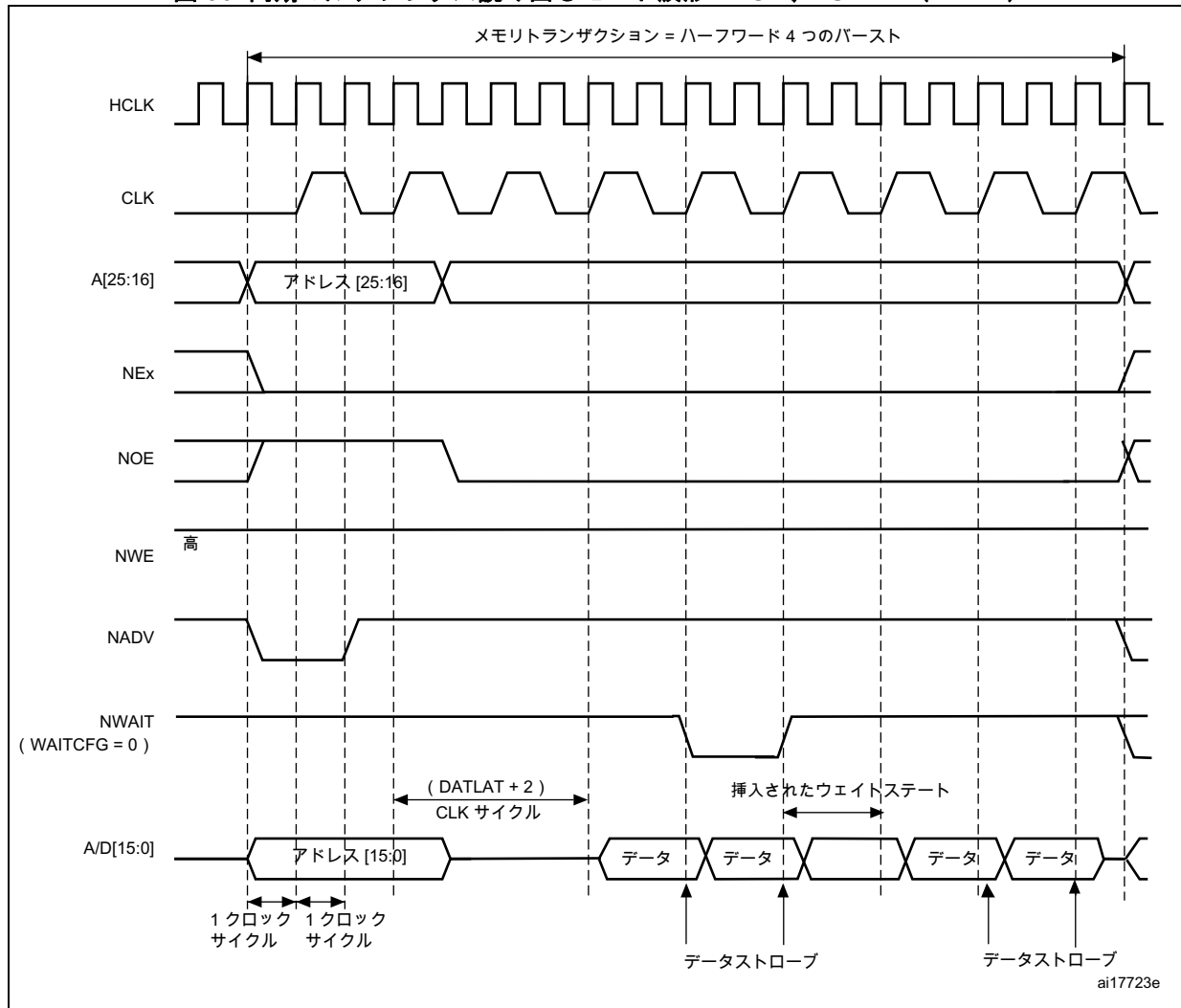


図 36. 同期マルチプレクス読み出しモード波形 - NOR、PSRAM (CRAM)



1. バイトレーン出力 (NBL は示されていません。NOR アクセス時にはハイに保たれ、PSRAM (CRAM) アクセス時にはローに保たれます。)

表 59. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-21	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	同期読み出し時は影響しません。
18-15	予約済み	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。
12	WREN	同期読み出し時は影響しません。
11	WAITCFG	メモリに応じて設定します。
10	WRAPMOD	0x0

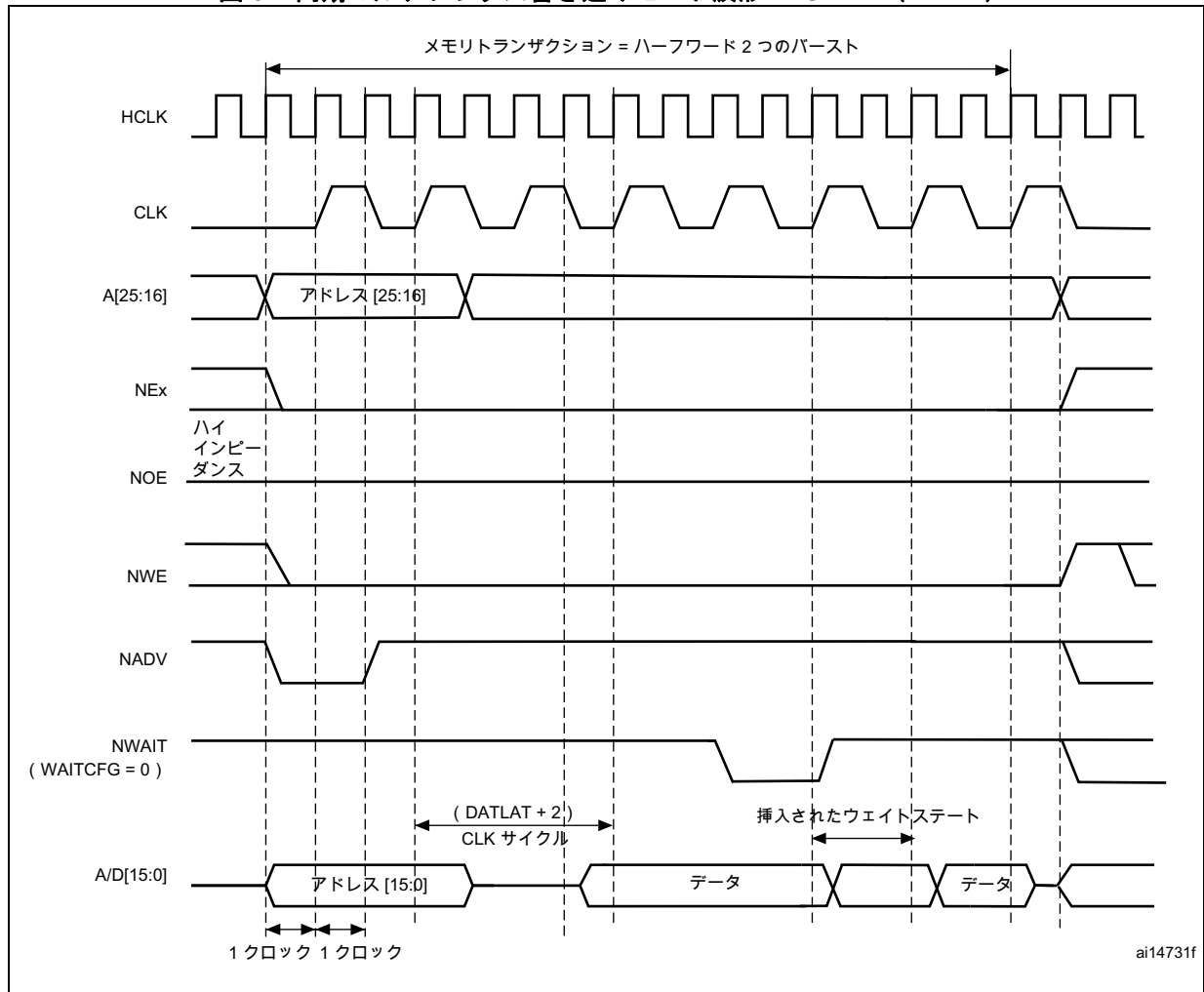
表 59. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	0x1
7	予約済み	0x1
6	FACCEN	メモリサポートに従って設定します (NOR 型フラッシュメモリ)
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x1 または 0x2
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 60. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31:30	予約済み	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	データ遅延
27-24	DATLAT	データ遅延
23-20	CLKDIV	0x0 CLK = HCLK の場合 (サポートしてません) 0x1 の場合、CLK=2×HCLK となります。 ..
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	無視
7-4	ADDHLD	無視
3-0	ADDSET	無視

図 37. 同期マルチプレクス書き込みモード波形 - PSRAM (CRAM)



1. メモリは、NWAIT 信号を 1 サイクル前に発行しなければならないので、WAITCFG を 0 にプログラムする必要があります。
2. バイトレーン (NBL) 出力は、図に示されておらず、NEx がアクティブな間はローに保たれます。

表 61. FMC\_BCRx ビットフィールド

ビット番号	ビット名	設定値
31-20	予約済み	0x000
20	CCLKEN	必要に応じて設定します。
19	CBURSTRW	0x1
18-15	予約済み	0x0
14	EXTMOD	0x0
13	WAITEN	メモリがこの機能をサポートする場合は 1 にセット、そうでなければ 0 のままとします。
12	WREN	0x1
11	WAITCFG	0x0
10	WRAPMOD	0x0

表 61. FMC\_BCRx ビットフィールド (続き)

ビット番号	ビット名	設定値
9	WAITPOL	メモリに応じて設定します。
8	BURSTEN	同期読み出し時は影響しません。
7	予約済み	0x1
6	FACCEN	メモリのサポート状況に応じて設定します。
5-4	MWID	必要に応じて設定します。
3-2	MTYP	0x1
1	MUXEN	必要に応じて設定します。
0	MBKEN	0x1

表 62. FMC\_BTRx ビットフィールド

ビット番号	ビット名	設定値
31-30	予約済み	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	データ遅延
23-20	CLKDIV	0x0 CLK = HCLK の場合 (サポートしてません) 0x1 の場合、CLK=2×HCLK となります。
19-16	BUSTURN	NEx ハイから NEx ローまでの時間 (BUSTURN HCLK)
15-8	DATAST	無視
7-4	ADDHLD	無視
3-0	ADDSET	無視



## 10.5.6 NOR/PSRAM コントローラレジスタ

### SRAM/NOR 型フラッシュチップセレクト制御レジスタ 1.4 (FMC\_BCR1.4)

アドレスオフセット :  $8 * (x - 1)$ ,  $x = 1..4$

リセット値 : バンク 1 は 0x0000 30DB、バンク 2~4 は 0x0000 30D2

このレジスタは、SRAM、PSRAM、および NOR 型フラッシュメモリで使用される各メモリバンクの制御情報を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCLKEN	CBURSTRW	Res.	Res.	Res.	ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Res.	FACCEN	MWID		MTYP	MUXEN	MBKEN	
											rW	rW				rW	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW	rW	

ビット 31 : 21 予約済みであり、リセット値に保持する必要があります。

#### ビット 20 CCLKEN : 連続クロック有効化

このビットは、外部メモリデバイスに対する FMC\_CLK クロック出力を有効化します。

0 : FMC\_CLK は同期メモリアクセス (読み出し/書き込みトランザクション) 時にのみ発生されます。FMC\_CLK クロック比は、FMC\_BCRx レジスタに設定した CLKDIV 値により指定されます (リセット後のデフォルト)。

1 : FMC\_CLK は、非同期アクセスと同期アクセス時に連続的に発生されます。CCLKEN がセットされると、FMC\_CLK クロックがアクティブになります。

**注 :** FMC\_BCR2.4 レジスタの CCLKEN ビットは無視されます。このビットは FMC\_BCR1 レジスタからのみ有効化できます。FMC\_CLK 連続クロックを発生するときは、バンク 1 を同期モードに設定する必要があります。

**注 :** CCLKEN ビットがセットされている場合、FMC\_CLK クロック比は FMC\_BTR1 レジスタの CLKDIV 値で指定されます。FMC\_BWTR1 の CLKDIV は、無視されます。

**注 :** 同期モードを使用し、かつ CCLKEN ビットがセットされる場合、バンク 1 以外のバンクに接続される同期メモリはすべて、同じクロックで駆動されます (他のバンクに対する FMC\_BTR2.4 レジスタと FMC\_BWTR2.4 レジスタの CLKDIV 値は無効です)。

#### ビット 19 CBURSTRW : 書き込みバースト有効化

バーストモードでの PSRAM (CRAM) 動作では、このビットは書き込み動作時に同期アクセスを可能にします。同期読み出しアクセスに対するイネーブルビットは、FMC\_BCRx レジスタの BURSTEN ビットです。

0 : 書き込み動作は、常に非同期モードで実行されます。

1 : 書き込み動作は、同期モードで実行されます。

ビット 18:16 予約済みであり、リセット値に保持する必要があります。

#### ビット 15 ASYNCWAIT : 非同期転送中のウエイト信号

このビットは、非同期プロトコル時でも、FMC が NWAIT 信号を有効化/無効化できるようにします。

0 : 非同期プロトコル動作中は、NWAIT 信号は考慮されません (リセット後のデフォルト)。

1 : 非同期プロトコル動作中に NWAIT 信号が考慮されます。

**ビット 14 EXTMOD : 拡張モード有効化**

このビットを使うと、FMC が FMC\_BWTR レジスタで非同期アクセスの書き込みタイミングをプログラムできるため、読み出しと書き込みで異なるタイミングが可能になります。

- 0 : FMC\_BWTR レジスタ内部の値は考慮されません (リセット後のデフォルト)。
- 1 : FMC\_BWTR レジスタ内部の値が考慮されます。

**注 :** 拡張モードを無効化した場合、FMC は次のようにモード1 またはモード2 で動作することができます。

- モード1は、SRAM/PSRAM メモリタイプを選択した場合 (MTYP = 0x0 または 0x01) のデフォルトモードです。
- モード2は、NOR メモリタイプを選択した場合 (MTYP = 0x10) のデフォルトモードです。

**ビット 13 WAITEN : ウェイトイネーブルビット**

このビットは、同期モードでフラッシュメモリをアクセスするとき NWAIT 信号を使用したウェイト状態の挿入を有効化/無効化します。

- 0 : NWAIT 信号は無効です (レベルは考慮されず、プログラムされたフラッシュ遅延時間後にウェイト状態は挿入されません)。
- 1 : NWAIT 信号は有効です (プログラムされた遅延時間後にレベルが考慮され、アサートされた場合に、ウェイト状態が挿入されます) (リセット後のデフォルト)。

**ビット 12 WREN : 書き込みイネーブルビット**

このビットは、FMC によるバンクへの書き込み動作の有効/無効を示します。

- 0 : FMC によるバンクへの書き込み動作は無効にされています。AHB エラーが報告されます。
- 1 : FMC によるバンクへの書き込み動作は有効にされています (リセット後のデフォルト)。

**ビット 11 WAITCFG : ウェイトタイミング設定**

NWAIT 信号は、メモリからのデータが有効か、または同期モードでフラッシュメモリをアクセスするときウェイト状態を挿入すべきかを表示します。この設定ビットは、ウェイト状態の1クロック前、またはウェイト状態中に、メモリによって NWAIT がアサートされるかどうかを定義します。

- 0 : NWAIT 信号は、ウェイト状態の1データサイクル前にアクティブです (リセット後のデフォルト)。
- 1 : NWAIT 信号は、ウェイト状態の間アクティブです (PSRAM には不使用)。

**ビット 10 WRAPMOD : ラップトバーストモードサポート**

コントローラが AHB バーストラップアクセスを2つのリニアアクセスに分割するかどうかを定義します。バーストモードでメモリにアクセスするときのみ有効です。

- 0 : ダイレクトラップトバーストは無効です (リセット後のデフォルト)。
- 1 : ダイレクトラップトバーストは有効です。

**注 :** このビットは、CPU および DMA がラッピングバースト転送を生成できない場合無効です。

**ビット 9 WAITPOL : ウェイト信号極性ビット**

同期モードまたは非同期モードで使用されるメモリからのウェイト信号の極性を定義します。

- 0 : NWAIT はアクティブローです (リセット後のデフォルト)。
- 1 : NWAIT はアクティブハイです。

**ビット 8 BURSTEN : バーストイネーブルビット**

このビットは、読み出し動作での同期アクセスを有効化/無効化します。これは、バーストモードでの同期メモリ動作に対してのみ有効です。

- 0 : バーストアクセスモードは無効です (リセット後のデフォルト)。読み出しアクセスは非同期モードで実行されます。
- 1 : バーストモードを有効化。読み出しアクセスは同期モードで実行されます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

**ビット 6 FACCEN : フラッシュアクセス有効化**

NOR 型フラッシュメモリアクセス動作を有効にします。

- 0 : 対応する NOR 型フラッシュメモリアクセスは無効です。
- 1 : 対応する NOR フラッシュメモリアクセスは有効です (リセット後のデフォルト)。

**ビット 5:4 MWID : メモリデータバス幅。**

外部メモリデバイスの幅を定義します。すべてのタイプのメモリに対して有効です。

- 00 : 8 ビット
- 01 : 16 ビット (リセット後のデフォルト)
- 10 : 予約済み。使用しないでください。
- 11 : 予約済み。使用しないでください。

**ビット 3:2 MTYP : メモリタイプ**

対応するメモリバンクに接続される外部メモリのタイプを定義します。

- 00 : SRAM (バンク 2...4 のリセット後のデフォルト)
- 01 : PSRAM (CRAM)
- 10 : NOR 型フラッシュ /OneNAND フラッシュ (バンク 1 のリセット後のデフォルト)
- 11 : 予約済み

**ビット 1 MUXEN : アドレス / データマルチプレクシングイネーブルビット**

このビットがセットされているときには、データバス上でアドレスとデータがマルチプレクスされます。NOR および PSRAM メモリでのみ有効です。

- 0 : アドレス/データはマルチプレクスされません。
- 1 : アドレス/データはデータバス上でマルチプレクスされます (リセット後のデフォルト)。

**ビット 0 MBKEN : メモリバンクイネーブルビット。**

メモリバンクを有効にします。リセット後は、バンク 1 は有効であり、その他はすべて無効です。無効なバンクにアクセスすると、AHB バス上で ERROR が発生します。

- 0 : 対応するメモリバンクは無効です
- 1 : 対応するメモリバンクは有効です。

## SRAM/NOR 型フラッシュチップセレクトタイミングレジスタ 1.4 (FMC\_BTR1..4)

アドレスオフセット :  $0x04 + 8 * (x - 1)$ ,  $x = 1..4$

リセット値 : 0x0FFF FFFF

リセット値 : 0x0FFF FFFF

このレジスタは、SRAM、PSRAM、および NOR 型フラッシュメモリで使用される各メモリバンクの制御情報を含みます。FMC\_BCRx レジスタの EXTMOD ビットがセットされている場合、このレジスタは、書き込みと読み出しアクセスに分割されます。すなわち、読み出しアクセスを設定するためのレジスタ (このレジスタ) と、書き込みアクセスを設定するためのレジスタ (FMC\_BWTRx レジスタ) の 2 つのレジスタが使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ACCMOD	DATLAT				CLKDIV				BUSTURN				DATAST				ADHLD				ADDSET								
			r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w



ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **ACCMOD** : アクセスモード

タイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FMC\_BCRx レジスタの EXTMOD ビットが 1 のときのみ考慮されます。

- 00 : アクセスモード A
- 01 : アクセスモード B
- 10 : アクセスモード C
- 11 : アクセスモード D

ビット 27:24 **DATLAT** : (下のビット説明参照) 同期メモリのデータ遅延

読み出し/書き込みバーストモードを有効にした同期 アクセスの場合 (BURSTEN / CBURSTRW ビットをセット)、最初のデータの読み書きの前にメモリに発行するメモリクロックサイクル数 (+2) を定義します。

このタイミングパラメータは、HCLK 周期ではなく、FMC\_CLK 周期で表されます。

非同期アクセスでは、この値は無視されます。

- 0000 : 最初のバーストアクセスでは、2 CLK クロックサイクルのデータ遅延
- 1111 : 最初のバーストアクセスでは、17 CLK クロックサイクルのデータ遅延 (リセット後のデフォルト)

ビット 23:20 **CLKDIV** : クロック分周比 (FMC\_CLK 信号)

FMC\_CLK クロック出力信号の周期を定義します。HCLK サイクル数で表されます。

- 0000 : 予約済み
- 0001 : FMC\_CLK 周期 = 2 × HCLK 周期
- 0010 : FMC\_CLK 周期 = 3 × HCLK 周期
- 1111 : FMC\_CLK 周期 = 16 × HCLK 周期 (リセット後のデフォルト値)

非同期 NOR 型フラッシュ、SRAM、または PSRAM アクセスでは、この値は無視されます。

**注 :** **FMC\_CLK 分周比の式については、セクション 10.5.5 : 同期トランザクション** を参照してください。

**ビット 19:16 BUSTURN[3:0] : バスターンアラウンドフェーズ時間**

これらのビットは、書き込みから読み出しまで（および読み出しから書き込みまで）のトランザクションの終わりに遅延を追加するためにソフトウェアで書き込まれます。この遅延を使うと、連続トランザクション間の最小時間（NEx ハイから NEx ローまでの tEHEL）と、読み出しアクセス後にメモリがデータバスを開放するために要する最大時間（tEHQZ）を次のように一致させることができます。プログラムされたバスターンアラウンド遅延は、非同期読み出しトランザクション（マルチプレクスまたはモード D）や書き込みトランザクションと、その他のスタティックバンクの同期または非同期の読み書きの間に挿入されます。読み出し時、バンクを同じものまたは別のものにできますが、書き込み時は、マルチプレクスまたはモード D を除いて別にできます。

一部のケースでは、プログラムされた BUSTURN の値にかかわらず、バスターンアラウンド遅延は固定されます。

次のとおりです。

- バスターンアラウンド遅延は、マルチプレクスモードおよびモード D を除いて同じスタティックメモリバンクに向けた 2 つの連続した非同期書き込み転送の間に挿入されません。
- 次の間に 1 FMC クロックサイクルのバスターンアラウンド遅延があります。
  - マルチプレクスモードおよびモード D を除いて同じスタティックメモリバンクに向けた 2 つの連続した非同期読み出し転送
  - マルチプレクスモードおよびモード D を除いて同じスタティックバンクまたはダイナミックバンクへの非同期または同期書き込みに対する非同期読み出し
  - 非同期読み出し（モード 1、2、A、B、C）と、別のスタティックバンクからの読み出し
- 次の間に 2 FMC クロックサイクルのバスターンアラウンド遅延があります。
  - 同じバンクへの 2 つの連続した同期書き込み（バーストまたはシングル）
  - スタティックメモリバンクへの同期書き込み（バーストまたはシングル）アクセスと、非同期書き込みまたは読み出し転送（読み出しの場合バンクは同じものまたは別のものにできます）
  - 2 つの連続した同期読み出し（バーストまたはシングル）、それに続く別のスタティックメモリバンクへの同期または非同期読み出しや書き込み
- 次の間に 3 FMC クロックサイクルのバスターンアラウンド遅延があります。
  - 別のスタティックバンクへの 2 つの連続した同期書き込み（バーストまたはシングル）同じまたは別のバンクへの同期書き込み（バーストまたはシングル）アクセスと同期読み出し

0000 : BUSTURN フェーズ時間 = 0 HCLK クロックサイクル追加

...

1111 : BUSTURN フェーズ時間 = 15× HCLK クロックサイクル（リセット後のデフォルト）

**ビット 15:8 DATAST** : データフェーズ時間

これらのビットは、非同期アクセスで 사용되는データフェーズの時間を定義するためにソフトウェアにより書き込まれます (図 20 ~ 図 32 を参照)。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 \* HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 \* HCLK クロックサイクル

...

1111 1111 : DATAST フェーズ時間 = 255 \* HCLK クロックサイクル (リセット後のデフォルト値)

各メモリタイプおよびアクセスモードでのデータフェーズ時間については、それぞれの図 (図 20 ~ 図 32) を参照してください。

例: モード 1、書き込みアクセス、DATAST=1 : データフェーズ時間 = DATAST+1 = 2 HCLK クロックサイクル

**注:** 同期アクセスでは、この値は無視されます。

**ビット 7:4 ADDHLD** : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアから書き込まれ (図 20 ~ 図 32 を参照)、モード D またはマルチプレクスアクセスで使用されます。

0000 : 予約済み

0001 : ADDHLD フェーズ時間 = 1 \* HCLK クロックサイクル

0010 : ADDHLD フェーズ時間 = 2 \* HCLK クロックサイクル

...

1111 : ADDHLD フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスホールドフェーズ時間については、それぞれの図 (図 20 ~ 図 32) を参照してください。

**注:** 同期アクセスでは、この値は使用されず、アドレスホールドフェーズは、常に 1 メモリクロック周期の長さです。

**ビット 3:0 ADDSET** : アドレスセットアップフェーズ時間

これらのビットは、アドレスセットアップフェーズの時間を定義するために、ソフトウェアによって書き込まれ (図 20 ~ 図 32 を参照)、SRAM、ROM、非同期 NOR フラッシュで使用されます。

0000 : ADDSET フェーズ時間 = 0 \* HCLK クロックサイクル

...

1111 : ADDSET フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

各アクセスモードでのアドレスセットアップフェーズ時間については、それぞれの図 (図 20 ~ 図 32) を参照してください。

**注:** 同期アクセスでは、この値は無視されます。

マルチプレクスモードとモード D では、ADDSET の最小値は 1 です。

**注:**

PSRAM (CRAM) の遅延は、内部更新のために可変です。したがって、これらのメモリは、必要に応じて遅延を延長するために、遅延フェーズ全体について NWAIT 信号を発行します。

SPRAM (CRAM) では、フィールド DATLAT を 0 に設定しなければなりません。こうすると、FMC は、すぐに遅延フェーズを終了して、メモリからの NWAIT 信号のサンプリングを開始し、メモリがレディ状態になると、読み出しまたは書き込みを開始します。

この方法は、古い世代のフラッシュメモリと異なり、NWAIT 信号を発行する最新世代の同期フラッシュメモリでも使用できます (使用中の特定のフラッシュメモリのデータシートを参照してください)。

## SRAM/NOR 型フラッシュ書き込みタイミングレジスタ 1.4 (FMC\_BWTR1..4)

アドレスオフセット :  $0x104 + 8 * (x - 1)$ ,  $x = 1..4$

リセット値 :  $0x0FFF\ FFFF$

このレジスタには各メモリバンクの制御情報が格納されます。SRAM、PSRAM、NOR フラッシュメモリに使用されます。FMC\_BCRx レジスタの EXTMOD ビットがセットされているときには、このレジスタは書き込みアクセスについてアクティブになります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ACCMOD		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN				DATASET						ADDHLD				ADDSET					
		r/w	r/w									r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **ACCMOD** : アクセスモード。

次のタイミング図に示されているように、非同期アクセスモードを指定します。これらのビットは、FMC\_BCRx レジスタの EXTMOD ビットが 1 にセットされている場合のみ考慮されます。

- 00 : アクセスモード A
- 01 : アクセスモード B
- 10 : アクセスモード C
- 11 : アクセスモード D

ビット 27:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:16 **BUSTURN** : バスターンアラウンドフェーズ時間

プログラムされたバスターンアラウンド遅延は、非同期書き込み転送と、その他のスタティックバンクの同期または非同期の読み書き転送の間に挿入されます。読み出し時、バンクを同じものまたは別のものにできますが、書き込み時は、マルチプレクスまたはモード D を除いて別にできます。

一部のケースでは、プログラムされた BUSTURN の値にかかわらず、バスターンアラウンド遅延は固定されます。

次のように固定されます。

- バスターンアラウンド遅延は、マルチプレクスモードおよびモード D を除いて同じスタティックメモリバンクに向けた 2 つの連続した非同期書き込み転送の間に挿入されません。
- 次の間に 2 FMC クロックサイクルのバスターンアラウンド遅延があります。
  - 同じバンクへの 2 つの連続した同期書き込み (バーストまたはシングル)
  - スタティックメモリバンクへの同期書き込み (バーストまたはシングル) 転送と、非同期書き込みまたは読み出し転送
- 次の間に 3 FMC クロックサイクルのバスターンアラウンド遅延があります。
  - 別のスタティックバンクへの 2 つの連続した同期書き込み (バーストまたはシングル) 同じまたは別のバンクへの同期書き込み (バーストまたはシングル) 転送と同期読み出し

0000 : BUSTURN フェーズ時間 = 0 HCLK クロックサイクル追加

...

1111 : BUSTURN フェーズ時間 = 15 HCLK クロックサイクル追加 (リセット後のデフォルト値)



**ビット 15:8 DATAST** : データフェーズ時間

これらのビットは、非同期 SRAM、PSRAM、NOR フラッシュメモリアクセスで使用されるデータフェーズの時間を定義するために、ソフトウェアによって書き込まれます (図 20 ~ 図 32 を参照)。

0000 0000 : 予約済み

0000 0001 : DATAST フェーズ時間 = 1 \* HCLK クロックサイクル

0000 0010 : DATAST フェーズ時間 = 2 \* HCLK クロックサイクル

...

1111 1111 : DATAST フェーズ時間 = 255 \* HCLK クロックサイクル (リセット後のデフォルト値)

**ビット 7:4 ADDHLD** : アドレスホールドフェーズ時間

これらのビットは、アドレスホールドフェーズの時間を定義するためにソフトウェアによって書き込まれ (図 20 ~ 図 32 を参照)、非同期マルチプレクスアクセスで使用されます。

0000 : 予約済み

0001 : ADDHLD フェーズ時間 = 1 \* HCLK クロックサイクル

0010 : ADDHLD フェーズ時間 = 2 \* HCLK クロックサイクル

...

1111 : ADDHLD フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

注: 同期 NOR 型フラッシュアクセスでは、この値は使用されず、アドレスホールドフェーズは常に 1 フラッシュクロック周期の長さです。

**ビット 3:0 ADDSET** : アドレスセットアップフェーズ時間。

これらのビットは、アドレスセットアップフェーズの時間を HCLK サイクル数で定義するためにソフトウェアによって書き込まれ (図 20 ~ 図 32 を参照)、非同期アクセスで使用されます。

0000 : ADDSET フェーズ時間 = 0 \* HCLK クロックサイクル

...

1111 : ADDSET フェーズ時間 = 15 \* HCLK クロックサイクル (リセット後のデフォルト値)

注: 同期アクセスでは、この値は使用されず、アドレスセットアップフェーズは常に 1 フラッシュクロック周期の長さです。マルチプレクスモードでは、ADDSET の最小値は 1 です。

## 10.6 NAND 型フラッシュ /PC カードコントローラ

FMC は、以下のタイプのデバイスを駆動するのに適した信号タイミングを発生します。

- 8 ビットと 16 ビットの NAND フラッシュメモリ
- 16 ビット PC カード互換デバイス

NAND フラッシュ /PC カードコントローラは、3 つの外部バンク (バンク 2、3、4) を制御することができます。

- バンク 2 およびバンク 3 は、NAND 型フラッシュデバイスをサポートします。
- バンク 4 は、PC カードデバイスをサポートします。

各バンクは、専用レジスタによって設定されます (セクション 10.6.8)。プログラム可能なメモリパラメータは、アクセスタイミング (表 63 を参照) と ECC 構成を含みます。



表 63. プログラム可能な NAND フラッシュ /PC カードアクセスパラメータ

パラメータ	機能	アクセスモード	単位	最小値	最大値
メモリセットアップ時間	コマンドアサーション前にアドレスをセットアップするために必要なクロックサイクル (HCLK) 数	読み出し／書き込み	AHB クロックサイクル (HCLK)	1	255
メモリウェイト	コマンドアサーションの最小時間 (HCLK クロックサイクル数)	読み出し／書き込み	AHB クロックサイクル (HCLK)	2	256
メモリホールド	コマンドアサーション解除後に、アドレスを (書き込みアクセス時はデータも) 保持する必要があるクロックサイクル数 (HCLK)。	読み出し／書き込み	AHB クロックサイクル (HCLK)	1	254
メモリデータバスハイインピーダンス	書き込みアクセス開始後にデータバスがハイインピーダンス状態に保たれるクロックサイクル数 (HCLK)	書き込み	AHB クロックサイクル (HCLK)	0	255

## 10.6.1 外部メモリインタフェース信号

次の表に、通常、NAND フラッシュメモリおよびPC カードとのインタフェースに使用される信号を示します。

注： 接頭辞「N」は、アクティブローの信号を表します。

### 8 ビット NAND フラッシュメモリ

表 64. 8 ビット NAND 型フラッシュ

FMC 信号名	I/O	機能
A[17]	O	NAND 型フラッシュアドレスラッチイネーブル (ALE) 信号
A[16]	O	NAND 型フラッシュコマンドラッチイネーブル (CLE) 信号
D[7:0]	I/O	8 ビットマルチプレクス、双方向アドレス/データバス
NCE[x]	O	チップセレクト、x=2、3
NOE(= NRE)	O	出カイネーブル (メモリ信号名：読み出しイネーブル、NRE)
NWE	O	書き込みイネーブル
NWAIT/INT[3:2]	I	FMC への NAND フラッシュレディ/ビジー入力信号

FMC は必要なだけのアドレスサイクルを管理できるので、容量についての理論上の限界はありません。

## 16 ビット NAND フラッシュメモリ

表 65. 16 ビット NAND 型フラッシュ

FMC 信号名	I/O	機能
A[17]	O	NAND 型フラッシュアドレスラッチイネーブル (ALE) 信号
A[16]	O	NAND 型フラッシュコマンドラッチイネーブル (CLE) 信号
D[15:0]	I/O	16 ビットマルチプレクス、双方向アドレス/データバス
NCE[x]	O	チップセレクト、x = 2、3
NOE(= NRE)	O	出カインーブル (メモリ信号名: 読み出しイネーブル、NRE)
NWE	O	書き込みイネーブル
NWAIT/INT[3:2]	I	FMC への NAND フラッシュレディ/ビジー入力信号

FMC は必要なだけのアドレスサイクルを管理できるので、容量についての理論上の限界はありません。

表 66. 16 ビット PC カード

FMC 信号名	I/O	機能
A[10:0]	O	アドレスバス
NIORD	O	I/O 空間の出カインーブル
NIOWR	O	I/O 空間の書き込みイネーブル
NREG	O	アクセスが共通または属性空間のいずれかを示すレジスタ信号
D[15:0]	I/O	双方向データバス
NCE4_1	O	チップセレクト 1
NCE4_2	O	チップセレクト 2 (アクセスが 16 ビットか 8 ビットかを示す)
NOE	O	共通および属性空間での出カインーブル
NWE	O	共通および属性空間での書き込みイネーブル
NWAIT	I	FMC への PC カードウェイト入力信号 (メモリ信号名 IORDY)
INTR	I	FMC への PC カード割り込み (割り込みを発生できる PC カードのみ)
CD	I	PC カードの有無の検出。アクティブハイ。CD がローの間に PC カードバンクへのアクセスが行われた場合、AHB エラーが生成されます。 <a href="#">セクション 10.3: AHB インタフェース</a> を参照してください。

## 10.6.2 NAND 型フラッシュ /PC カードによってサポートされるメモリおよびトランザクション

表 67 に、サポートされるデバイス、アクセスモード、およびトランザクションを示します。NAND フラッシュ /PC カードコントローラによって許可されない (またはサポートされない) トランザクションは、灰色で示されています。

表 67. サポートされるメモリおよびトランザクション

デバイス	モード	読み出し/ 書き込み	AHB データ サイズ	メモリ データ サイズ	設定許可/ 設定禁止	コメント
NAND 8 ビット	非同期	R	8	8	Y	-
	非同期	W	8	8	Y	-
	非同期	R	16	8	Y	2 つの FMC アクセスに分割
	非同期	W	16	8	Y	2 つの FMC アクセスに分割
	非同期	R	32	8	Y	4 つの FMC アクセスに分割
	非同期	W	32	8	Y	4 つの FMC アクセスに分割
NAND 16 ビット	非同期	R	8	16	Y	-
	非同期	W	8	16	N	-
	非同期	R	16	16	Y	-
	非同期	W	16	16	Y	-
	非同期	R	32	16	Y	2 つの FMC アクセスに分割
	非同期	W	32	16	Y	2 つの FMC アクセスに分割

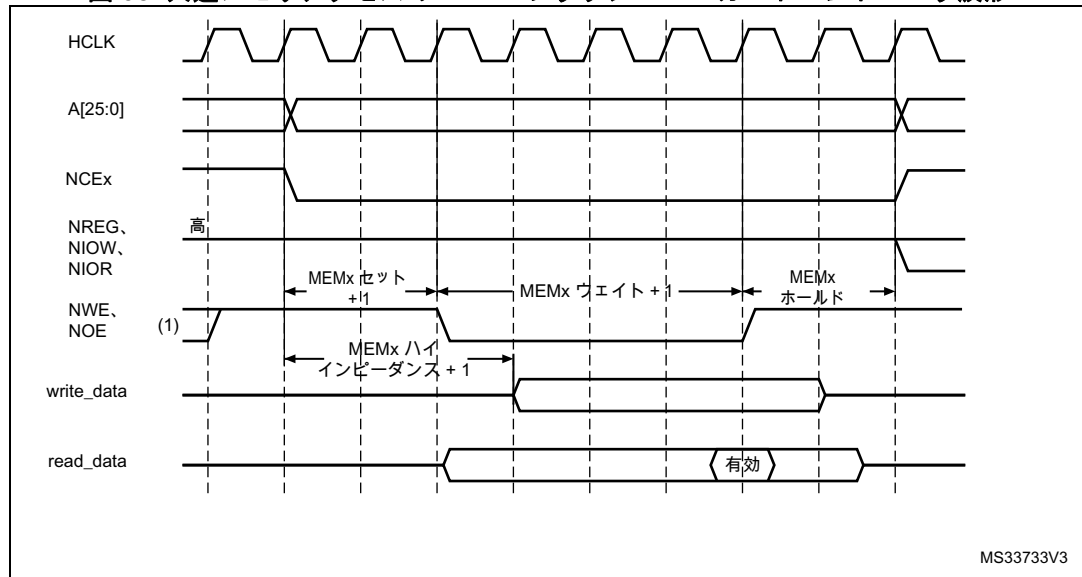
## 10.6.3 NAND フラッシュメモリと PC カードのタイミング図

PC カード/コンパクトフラッシュ、および NAND 型フラッシュメモリの各バンクは、レジスタのセットを通じて管理されます。

- 制御レジスタ : FMC\_PCRx
- 割り込みステータスレジスタ : FMC\_SRx
- ECC レジスタ : FMC\_ECCRx
- 共通メモリ空間用タイミングレジスタ : FMC\_PMEMx
- 属性メモリ空間用タイミングレジスタ : FMC\_PATTx
- I/O スペース用タイミング・レジスタ : FMC\_PIOx

各タイミング構成レジスタは、PC カード/コンパクトフラッシュ、または NAND フラッシュアクセスの 3 つのフェーズの HCLK サイクル数を定義するための 3 つのパラメータと、書き込みの際にデータバスの駆動を開始するタイミングを定義する 1 つのパラメータを格納します。図 38 は、共通メモリアccessのタイミングパラメータの定義を示します。属性メモリ空間と I/O (PC カードのみ) メモリ空間のアクセスタイミングが同様であることがわかります。

図 38. 共通メモリアクセスの NAND フラッシュ / PC カードコントローラ波形



- 書き込みアクセス時、NOE はハイ（非アクティブ）に保たれます。読み出しアクセス時、NWE はハイ（非アクティブ）に保たれます。
- 書き込みアクセスの場合、ホールドフェーズ遅延は（MEMHOLD）HCLK サイクルとなり、読み出しアクセスの場合、（MEMHOLD + 2）HCLK サイクルとなります。

## 10.6.4 NAND フラッシュ動作

NAND フラッシュデバイスのコマンドラッチイネーブル（CLE）およびアドレスラッチイネーブル（ALE）信号は、FMC コントローラのアドレス信号によって駆動されます。これは、NAND フラッシュメモリにコマンドまたはアドレスを送信するには、CPU は、CPU のメモリ空間内の特定のアドレスに書き込みを行う必要があることを意味します。

NAND フラッシュデバイスからの一般的なページ読み出し動作には、次のステップが必要です。

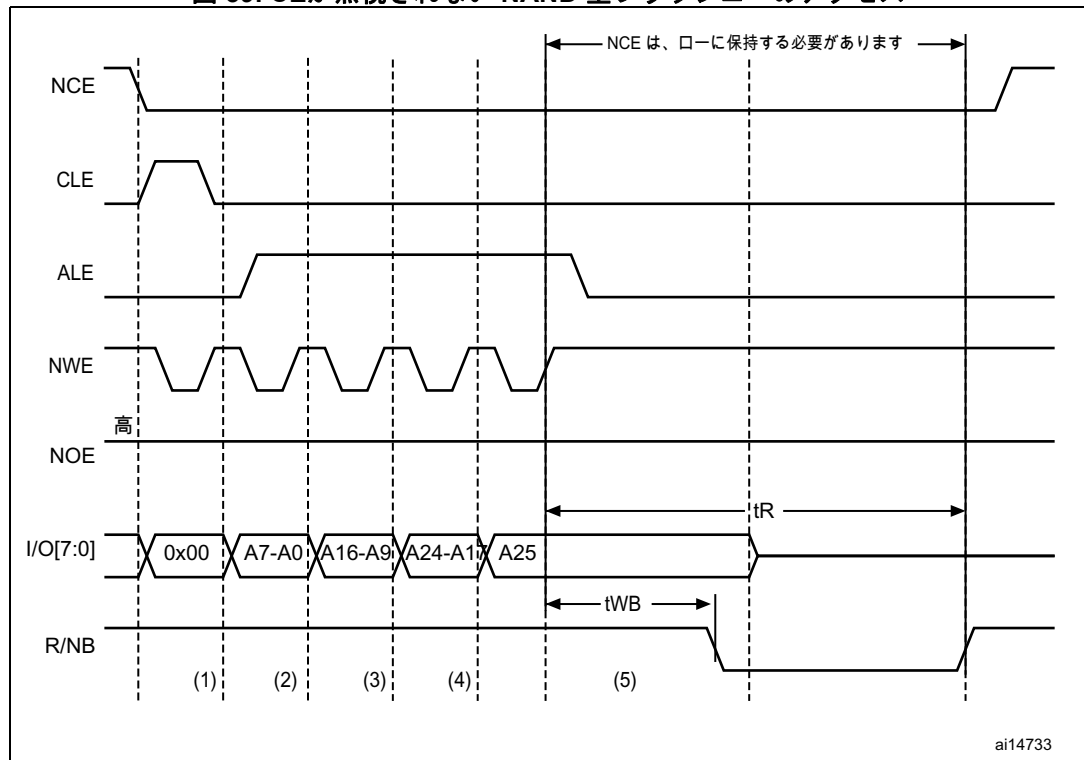
- FMC\_PCRx レジスタと FMC\_PMEMx レジスタ（デバイスによっては、FMC\_PATTx も。セクション 10.6.5 : NAND フラッシュのプリウエイト機能を参照）を、NAND フラッシュメモリの特長（NAND フラッシュのデータバス幅に対する PWID ビット、PTYP = 1、必要に応じて PWAITEN = 0 または 1、タイミング設定についてはセクション 10.4.2 : NAND フラッシュメモリ/PC カードアドレスマッピングを参照）に従って設定することで、対応するメモリバンクのプログラムおよび有効化を行います。
- CPU は、1 つのフラッシュコマンドバイトと等しいデータバイト（たとえば、Samsung の NAND フラッシュデバイスの場合は 0x00）で、共通メモリ空間にバイト書き込みを行います。NAND フラッシュの LE 入力は、書き込みストロブ（NWE 上のローパルス）の間はアクティブであり、書き込まれたバイトは、NAND フラッシュによってコマンドとして認識されます。コマンドがメモリデバイスにラッチされたら、次のページ読み出し動作のために書き込む必要はありません。
- CPU は、4 バイト（より小容量のデバイスの場合は 3 バイト）を共通メモリまたは属性空間に書き込むことによって、読み出し操作の開始アドレス（STARTAD）を送信できます（STARTAD[7:0]、STARTAD[16:9]、STARTAD[24:17]、および 64 M ビット x 8 ビット NAND フラッシュの場合は最後に STARTAD[25]）。NAND 型フラッシュデバイスの ALE 入力は、書き込みストロブの間アクティブであり（NWE 上のローパルス）、書き込まれたバイトは、読み出し動作の開始アドレスとして解釈されます。属性メモリ空間を使用すると、FMC の異なるタイミング設定を使用でき、一部の NAND フラッシュメモリで必要とされるプリウエイト機能を実装できます（詳細については、セクション 10.6.5 : NAND フラッシュのプリウエイト機能を参照してください）。

6. コントローラは、NAND フラッシュメモリがレディ (R/NB 信号がハイ) になるのを待ってから、同じメモリバンクまたは別のメモリバンクへの新しいアクセスを開始します。待っている間、コントローラは NCE 信号をアクティブ (ロー) に保ちます。
7. CPU は、NAND フラッシュページ (データフィールド+スペアフィールド) をバイト単位で読み出すために、共通メモリ空間からバイト読み出し動作を行います。
8. 次の NAND フラッシュページは、CPU コマンドやアドレス書き込み動作なしに読み出すことができます。これは次の 3 つの方法で行うことができます。
  - ステップ 5 に記述された動作を実行します。
  - ステップ 3 から動作をリスタートすることによって、新しいランダムアドレスにアクセスできます。
  - ステップ 2 からリスタートすることによって、NAND 型フラッシュデバイスに新しいコマンドを送信できます。

## 10.6.5 NAND フラッシュのプリウエイト機能

一部の NAND フラッシュデバイスでは、アドレスの最後の部分を書き込んだ後、コントローラは R/NB 信号を待つ必要があります。(図 39 を参照。)

図 39. CEが無視されない NAND 型フラッシュへのアクセス



1. CPU はアドレス 0x7001 0000 にバイト 0x00 を書き込みました。
2. CPU はアドレス 0x7002 0000 にバイト A7~A0 を書き込みました。
3. CPU はアドレス 0x7002 0000 にバイト A16~A9 を書き込みました。
4. CPU はアドレス 0x7002 0000 にバイト A24~A17 を書き込みました。
5. CPU はアドレス 0x7002 0000 にバイト A25 を書き込みました : FMC は FMC\_PATT2 のタイミング定義を使用して書き込みアクセスを実行します。ここで  $ATTHOLD \geq 7$  (ただし、 $(7+1) * HCLK = 112 \text{ ns} > t_{WB \text{ max}}$ )。これにより、NCE は R/NB がローから再びハイになるまでローを維持します (NCE が無視されない NAND 型フラッシュメモリの場合のみ必要)。

この機能が必要な場合、MEMHOLD 値をプログラムすることにより  $t_{WB}$  タイミングを確実に満たすことができます。ただし、NAND フラッシュメモリに対するすべての CPU 読み出しアクセスでは、(MEMHOLD+ 2) HCLK サイクル、CPU 書き込みアクセスでは、(MEMHOLD) HCLK サイクルのホールド遅延が NWE 信号の立ち上がりエッジと次のアクセスとの間に挿入されます。

このタイミング制約に対処するためには、 $t_{WB}$  タイミングに応じた ATTHOLD 値でタイミングレジスタをプログラムし、MEMHOLD 値を最小値のままにしておくことで、属性メモリ空間を使用できます。この場合、CPU は NAND フラッシュのすべての読み出しおよび書き込みアクセスには共通メモリ空間を使用する必要があります。ただし、NAND フラッシュデバイスに最後のアドレスバイトを書き込むときには、属性メモリ空間に書き込む必要があります。

## 10.6.6 エラー訂正コード (ECC) の計算 NAND 型フラッシュメモリ

FMC の PC カードコントローラには、メモリバンクごとに 1 つずつ、2 つのエラー訂正コード計算ハードウェアブロックがあります。これらのハードウェアブロックは、ソフトウェアで ECC を処理する場合、ホスト CPU の負荷を削減します。

この 2 つの ECC ブロックは同一であり、それぞれバンク 2 とバンク 3 に接続されます。結果として、バンク 4 に接続されたメモリでは、ハードウェア ECC 計算を使用できません。

FMC に実装されたエラー訂正コード (ECC) のアルゴリズムでは、NAND フラッシュメモリに対する 256、512、1024、2048、4096、または 8192 バイトの読み出しまたは書き込みごとに、1 ビットのエラー訂正と 2 ビットのエラー検出を実行できます。ハミングコーディングアルゴリズムを採用し、行と列のパリティ計算を行います。

ECC モジュールは、NAND フラッシュメモリバンクがアクティブになるたびに、NAND フラッシュデータバスおよび読み出し/書き込み信号 (NCE と NWE) を監視します。

ECC は次のように動作します。

- NAND フラッシュメモリのバンク 2 またはバンク 3 へアクセスするときは、D[15:0] バス上のデータがラッチされて、ECC 計算に使用されます。
- NAND フラッシュメモリの他のアドレスをアクセスするときは、ECC ロジックはアイドル状態となり、動作しません。このため、NAND フラッシュメモリに対するコマンドまたはアドレスを定義する書き込み動作は、ECC 計算で考慮されません。

目的のバイト数がホスト CPU によって NAND フラッシュメモリから読み出されるか、NAND フラッシュメモリに書き込まれたら、FMC\_ECCR2/3 レジスタを読み出して、計算値を取得する必要があります。読み出し後は、ECCEN ビットを"0"にリセットすることによってクリアされます。新しいデータクロックを計算するには、FMC\_PCR2/3 レジスタの ECCEN ビットを 1 にセットする必要があります。

ECC 計算を行うときは：

1. FMC\_PCR2/3 レジスタで ECCEN ビットを有効にします。
2. NAND 型フラッシュメモリページヘータを書き込みます。NAND ページへ書き込む際、ECC ブロックは ECC 値を計算します。
3. FMC\_ECCR2/3 レジスタにある ECC 値を読み出しして、変数内に格納します。
4. ECCEN ビットをクリアし、FMC\_PCR2/3 レジスタで有効にした後、書き込んだデータを NAND ページから読み出します。NAND ページを読み出す際、ECC ブロックは ECC 値を計算します。
5. FMC\_ECCR2/3 レジスタにある新しい ECC 値を読み出します。
6. 2 つの ECC 値が同じ場合、訂正は不要です。その他の場合、ECC エラーが存在するためソフトウェア訂正ルーチンがエラーが訂正できるか否かの情報を返します。

## 10.6.7 PC カード/コンパクトフラッシュの動作

### アドレス空間とメモリアクセス

FMC は、コンパクトフラッシュデバイスまたは PC カードをメモリモードおよび I/O モードでサポートします (True IDE モードはサポートされません)。

コンパクトフラッシュカードと PC カードは、次の 3 つのメモリ空間から構成されます。

- 共通メモリ空間
- 属性空間
- I/O メモリ空間

nCE2 および nCE1 ピン (それぞれ FMC\_NCE4\_2 および FMC\_NCE4\_1) は、カードを選択し、バイト動作が行われているか、ワード動作が行われているかを示します。nCE2 は、D15-8 上の奇数バイトにアクセスし、nCE1 は、A0=0 の場合は D7-0 の偶数バイト、A0=1 の場合は D7-0 上の奇数バイトにアクセスします。nCE2 と nCE1 の両方ともローの場合は、D15-0 上のフルワードがアクセスされます。

メモリ空間は、nCE2/nCE1 と nREG のローアサーションと組み合わせて、読み出しアクセスの場合は nOE を、書き込みアクセスの場合は nWE を、それぞれローにアサートすることによって選択されます。

- メモリアクセス時にピン nREG=1 の場合は、共通メモリ空間が選択されます。
- メモリアクセス時にピン nREG=0 の場合は、属性メモリ空間が選択されます。

I/O 空間は、nCE2/nCE1 と組み合わせて、読み出しアクセスの場合は nIORD を、書き込みアクセスの場合は nIOWR をそれぞれアサートすることによって選択されます (メモリ空間の場合は nOE/nWE を使います)。I/O 空間にアクセスするときには、nREG もローにアサートされなければならないことに注意してください。

16 ビット PC カードでは、次の 3 種類のアクセスが可能です。

- データストレージのための共通メモリ空間へのアクセスは、偶数アドレスでは 8 ビットアクセス、または 16 ビット AHB アクセスが可能です。

奇数アドレスでの 8 ビットアクセスはサポートされないため、nCE2 がローに駆動されないことに注意してください。32 ビット AHB リクエストは、2 つの 16 ビットメモリアクセスに変換されます。

- PC カードが構成情報を格納している属性メモリ空間へのアクセスは、偶数アドレスでの 8 ビット AHB アクセスに限られます。

16 ビット AHB アクセスは 1 つの 8 ビットメモリ転送に変換されることに注意してください。nCE1 はローにアサートされ、nCE2 はハイにアサートされ、D7-D0 上の偶数バイトのみが有効になります。32 ビット AHB アクセスは 2 つの 8 ビットメモリ転送に変換される代わりに、nCE1 はローにアサートされ、nCE2 はハイにアサートされ、偶数バイトのみが有効になります。

- I/O 空間へのアクセスでは、8 ビットまたは 16 ビットの AHB アクセスが可能です。

表 68. 16 ビット PC- カード信号とアクセスタイプ

nCE2	nCE1	nREG	noE/nWE	nIORD/nIOWR	A10	A9	A7-1	A0	空間	アクセスタイプ	許可/禁止
1	0	1	0	1	X	X	X-X	X	共通メモリ空間	D7-D0 上のバイトの読み出し／書き込み	YES
0	1	1	0	1	X	X	X-X	X		D15-D8 上のバイトの読み出し／書き込み	サポートされません
0	0	1	0	1	X	X	X-X	0		D15-D0 上のワードの読み出し／書き込み	YES
X	0	0	0	1	0	1	X-X	0	属性空間	設定レジスタの読み出しまたは書き込み	YES
X	0	0	0	1	0	0	X-X	0		CIS (カード情報構造) の読み出しまたは書き込み	YES
1	0	0	0	1	X	X	X-X	1	属性空間	無効な読み出しまたは書き込み (奇数アドレス)	YES
0	1	0	0	1	X	X	X-X	x		無効な読み出しまたは書き込み (奇数アドレス)	YES
1	0	0	1	0	X	X	X-X	0	I/O 空間	D7-0 上の偶数バイトの読み出し	YES
1	0	0	1	0	X	X	X-X	1		D7-0 上の奇数バイトの読み出し	YES
1	0	0	1	0	X	X	X-X	0		D7-0 上の偶数バイトの書き込み	YES
1	0	0	1	0	X	X	X-X	1		D7-0 上の奇数バイトの書き込み	YES
0	0	0	1	0	X	X	X-X	0		D15-0 上のワードの読み出し	YES
0	0	0	1	0	X	X	X-X	0		D15-0 上のワードの書き込み	YES
0	1	0	1	0	X	X	X-X	X		D15-8 上の奇数バイトの読み出し	サポートされません
0	1	0	1	0	X	X	X-X	X		D15-8 上の奇数バイトの書き込み	サポートされません

FMC バンク 4 では、[セクション 10.4.2 : NAND フラッシュメモリ/PC カードアドレスマッピング](#) および [表 35: NAND/PC カードメモリマッピングおよびタイミングレジスタ](#) で説明する 3 つのメモリ空間へのアクセスが可能です。



## ウェイト機能

コンパクトフラッシュや PC カードは、FMC\_PCRx レジスタの PWAITEN ビットを通じてウェイト機能が有効な場合、nOE/nWE または nIORD/nIOWR が有効になった後に nWAIT 信号をアサートすることによって、MEMWAITx/ATTWAITx/IOWAITx ビットでプログラムされたアクセスフェーズの長さを延長するように FMC に要求することができます。nWAIT のアサートを正しく検出するためには、MEMWAITx/ATTWAITx/IOWAITx ビットを次のようにプログラムする必要があります。

$$xxWAITx \geq 4 + \frac{\text{max\_wait\_assertion\_time}}{\text{HCLK}}$$

ここで、max\_wait\_assertion\_time は、nOE/nWE または nIORD/nIOWR がローになってから nWAIT がローになるまでの最大時間です。

WAIT のアサーション後、FMC は、WAIT フェーズを 4 HCLK クロックサイクルだけ延長します。

## 10.6.8 NAND 型フラッシュ /PC カードコントローラレジスタ

### PC カード/NAND フラッシュ制御レジスタ 2..4 (FMC\_PCR2..4)

アドレスオフセット : 0x40 + 0x20 \* (x - 1)、x = 2..4

リセット値 : 0x0000 0018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS			TAR				TCLR				Res.	Res.	ECCEN	PWID		PTYP	PBKEN	PWAITEN	Res.
												rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:17 **ECCPS** : ECC ページサイズ

外部 ECC のページサイズを定義します :

- 000 : 256 バイト
- 001 : 512 バイト
- 010 : 1024 バイト
- 011 : 2048 バイト
- 100 : 4096 バイト
- 101 : 8192 バイト

ビット 16:13 **TAR** : ALE から RE 間の遅延

ALE ローから RE ローまでの時間を AHB クロックサイクル (HCLK) 数で設定します。時間は、t\_ar = (TAR + SET + 2) \* THCLK です。THCLK は HCLK クロック周期です。

- 0000 : 1 HCLK サイクル (デフォルト)
- 1111 : 16 HCLK サイクル

**注 :** SET は、アドレス指定される空間に応じて、MEMSET または ATTSET です。

ビット 12:9 **TCLR** : CLE から RE 間の遅延

CLE ローから RE ローまでの時間を AHB クロックサイクル (HCLK) 数で設定します。時間は、t\_clr = (TCLR + SET + 2) \* THCLK です。THCLK は HCLK クロック周期です。

- 0000 : 1 HCLK サイクル (デフォルト)
- 1111 : 16 HCLK サイクル

**注 :** SET は、アドレス指定される空間に応じて、MEMSET または ATTSET です。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。



ビット 6 **ECEN** : ECC 計算ロジックイネーブルビット

- 0 : ECC ロジックは無効であり、リセットされています (リセット後のデフォルト)。
- 1 : ECC ロジックは有効です。

ビット 5:4 **PWID** : データバス幅。

外部メモリデバイスの幅を定義します。

- 00 : 8 ビット
- 01 : 16 ビット (リセット後のデフォルト) この値は PC カードに対して必須です。
- 10 : 予約済み。使用しないでください。
- 11 : 予約済み。使用しないでください。

ビット 3 **PTYP** : メモリタイプ

対応するメモリバンクに接続されたデバイスのタイプを定義します。

- 0 : PC カード、コンパクトフラッシュ、CF+ または PCMCIA
- 1 : NAND 型フラッシュ (リセット後のデフォルト)

ビット 2 **PBKEN** : PC カード/NAND フラッシュメモリバンクイネーブルビット。

メモリバンクを有効にします。無効なメモリバンクにアクセスすると、AHB バスの ERROR になります。

- 0 : 対応するメモリバンクは無効です (リセット後のデフォルト)。
- 1 : 対応するメモリバンクは有効です。

ビット 1 **PWAITEN** : ウェイト機能イネーブルビット

PC カード/NAND 型フラッシュメモリバンクのウェイト機能を有効にします。

- 0 : 無効
- 1 : 有効

**注 :** PC カードの場合、ウェイト機能を有効にするときには、MEMWAITx/ATTWAITx/IOWAITx ビットを次の値にプログラムする必要があります。

$$xxWAITx^3 + max\_wait\_assertion\_time/HCLK$$

ここで、max\_wait\_assertion\_time は、nOE/nWE または nIORD/nIOWR がローになってから NWAIT がローになるまでの最大時間です。

ビット 0 予約済み。

## FIFO ステータスおよび割り込みレジスタ 2..4 (FMC\_SR2..4)

アドレスオフセット :  $0x44 + 0x20 * (x-1)$ ,  $x = 2..4$

リセット値 : 0x0000 0040

このレジスタは、FIFO ステータスと割り込みに関する情報を格納します。FMC には、AHB の 16 ワードまでのデータをメモリに書き込んで格納するときに使用される FIFO があります。

これは、FMC が FIFO をメモリへ転送すると同時に、FIFO に迅速に書き込んで、FMC 以外のペリフェラルへのトランザクションに AHB を解放するために使用されます。これらのレジスタビットの 1 つは、ECC のために、FIFO のステータスを示します。

ECC は、データのメモリへの書き込み中に計算されます。正しい ECC を読み出すため、ソフトウェアは FIFO がエンプティになるまで待つ必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEMPT	IFEN	ILEN	IFEN	IFS	ILS	IRS
																									r	rw	rw	rw	rw	rw	rw

ビット 31:7 予約済みであり、リセット値に保持する必要があります。

- ビット 6 **FEMPT** : FIFO エンプティ  
 FIFO のステータスを表示する読み取り専用ビット  
 0 : FIFO 非エンプティ  
 1 : FIFO エンプティ
- ビット 5 **IFEN** : 割り込み立ち下がリエッジ検出イネーブルビット  
 0 : 割り込み立ち下がリエッジ検出リクエストは無効  
 1 : 割り込み立ち下がリエッジ検出リクエストは有効
- ビット 4 **ILEN** : 割り込みハイレベル検出イネーブルビット  
 0 : 割り込みハイレベル検出リクエストは無効  
 1 : 割り込みハイレベル検出リクエストは有効
- ビット 3 **IREN** : 割り込み立ち上がりエッジ検出イネーブルビット  
 0 : 割り込み立ち上がりエッジ検出リクエストは無効  
 1 : 割り込み立ち上がりエッジ検出リクエストは有効
- ビット 2 **IFS** : 割り込み立ち下がリエッジステータス  
 このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。  
 0 : 割り込み立ち下がリエッジは発生しませんでした。  
 1 : 割り込み立ち下がリエッジが発生しました。  
**注 :** ソフトウェアがこのビットに 1 を書き込むとセットされます。
- ビット 1 **ILS** : 割り込みハイレベルステータス  
 このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。  
 0 : 割り込みハイレベルは発生しませんでした。  
 1 : 割り込みハイレベルが発生しました。
- ビット 0 **IRS** : 割り込み立ち上がりエッジステータス  
 このフラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。  
 0 : 割り込み立ち上がりエッジは発生しませんでした。  
 1 : 割り込み立ち上がりエッジが発生しました。  
**注 :** ソフトウェアがこのビットに 1 を書き込むとセットされます。

## 共通メモリ空間タイミングレジスタ 2..4 (FMC\_PMEM2..4)

アドレスオフセット : アドレス :  $0x48 + 0x20 * (x - 1)$ ,  $x = 2..4$

リセット値 :  $0xFCFC\ FCFC$

各 FMC\_PMEMx ( $x = 2..4$ ) 読み出し/書き込みレジスタは、PC カードまたは NAND フラッシュメモリバンク x のタイミング情報を格納します。この情報は、16 ビット PC カード/コンパクトフラッシュの共通メモリ空間をアクセスするとき、またはコマンド、アドレス書き込みアクセスおよびデータ読み出し/書き込みアクセスのための NAND フラッシュをアクセスするときに使われます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
MEMHIZx								MEMHOLDx								MEMWAITx								MEMSETx															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



### ビット 31:24 MEMHIZx : 共通メモリ x データバスハイインピーダンス時間

ソケット x 上の共通メモリ空間への PC カード/NAND フラッシュによる書き込みアクセスの開始後に、データバスをハイインピーダンスに保つ HCLK クロックサイクル数を定義します。これは、書き込みトランザクションに対してのみ有効です。

0000 0000 : (0x00) 0 HCLK サイクル (PC カード) / 1 HCLK サイクル (NAND フラッシュ)  
 1111 1110 : (0xFF) 255 HCLK サイクル (PC カードの場合) / 256 HCLK サイクル (NAND フラッシュの場合)  
 1111 1111 : 予約済み

### ビット 23:16 MEMHOLDx : 共通メモリ x ホールド時間

ソケット x 上の共通メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドのネゲート (NWE、NOE) 後にアドレスを (書き込みアクセス時はデータも) 保持する書き込みアクセス時の HCLK クロックサイクル数と読み出しアクセス時の HCLK (+2) クロックサイクル数を定義します。

0000 0000 : 予約済み  
 0000 0001 : 書き出しアクセスの場合 1 HCLK サイクル / 読み出しアクセスの場合 3 HCLK サイクル  
 1111 1110 : 書き出しアクセスの場合 254 HCLK サイクル / 読み出しアクセスの場合 256 HCLK サイクル  
 1111 1111 : 予約済み。

### ビット 15:8 MEMWAITx : 共通メモリ x ウェイト時間

ソケット x 上の共通メモリ空間への PC カード/NAND 型フラッシュによる読み出しまたは書き込みアクセスの際に、コマンド (NEW、NOE) をアサートするための HCLK (+1) クロックサイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサーションの時間は延長されます。

0000 0000 : 予約済み  
 0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)  
 1111 1111 : 256 HCLK サイクル (+カードが NWAIT をデアサートすることによって導入されるウェイトサイクル)

### ビット 7:0 MEMSETx : 共通メモリ x セットアップ時間

ソケット x 上の共通メモリ空間への PC カード/NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドアサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロックサイクル数を定義します。

0000 0000 : 1 HCLK サイクル (PC カードの場合) / HCLK サイクル (NAND 型フラッシュの場合)  
 1111 1110 : 255 HCLK サイクル (PC Card の場合) / 257 HCLK サイクル (NAND フラッシュの場合)  
 1111 1111 : 予約済み

## 属性メモリ空間タイミングレジスタ 2..4 (FMC\_PATT2..4)

アドレスオフセット :  $0x4C + 0x20 * (x - 1)$ ,  $x = 2..4$

リセット値 : 0xFCFC FCFC

各 FMC\_PATTx ( $x = 2..4$ ) 読み出し / 書き込みレジスタは、PC カード / コンパクトフラッシュまたは NAND フラッシュメモリバンク x のタイミング情報を含みます。この情報は、最後のアドレス書き込みアクセスの際に、タイミングが以前のアクセスのタイミングと異なる必要がある場合に、PC カード / コンパクトフラッシュの属性メモリ空間へ 8 ビットアクセスや NAND フラッシュにアクセスするために使用されます (レディ / ビジー管理については、[セクション 10.6.5 : NAND フラッシュのブリウエイト機能](#)を参照)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATTHIZ								ATTHOLD								ATTWAIT								ATTSET							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	

**ビット 31:24 ATTHIZ** : 属性メモリ x データバスハイインピーダンス時間

ソケット x 上の属性メモリ空間への PC カード/NAND フラッシュによる書き込みアクセスの開始後に、データバスをハイインピーダンスに保つ HCLK クロックサイクル数を定義します。これは、書き込みトランザクションに限り有効です。

0000 0000 : 0 HCLK サイクル  
1111 1110 : 255 HCLK サイクル  
1111 1111 : 予約済み

**ビット 23:16 ATTHOLD** : 属性メモリ x ホールド時間

ソケット上の属性メモリ空間への NAND フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドのネゲート (NWE、NOE) 後にアドレスを (書き込みアクセス時はデータも) 保持する書き込みアクセス時の HCLK クロックサイクル数と読み出しアクセス時の HCLK (+2) クロックサイクル数を定義します。

0000 0000 : 予約済み  
0000 0001 : 書き出しアクセスの場合 1 HCLK サイクル/読み出しアクセスの場合 3 HCLK サイクル  
1111 1110 : 書き出しアクセスの場合 254 HCLK サイクル/読み出しアクセスの場合 256 HCLK サイクル  
1111 1111 : 予約済み。

**ビット 15:8 ATTWAIT** : 属性メモリ x ウェイト時間

ソケット x 上の属性メモリ空間への PC カード/NAND 型フラッシュによる読み出しまたは書き込みアクセスの際に、コマンド (NEW、NOE) をアサートするための HCLK (+1) クロックサイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサーションの時間は延長されます。

0000 0000 : 予約済み  
0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)  
1111 1111 : 256 HCLK サイクル (+カードが NWAIT をデアサートすることによって導入されるウェイトサイクル)

**ビット 7:0 ATTSET** : 属性メモリ x セットアップ時間

ソケット x 上の属性メモリ空間への PC カード/NAND 型フラッシュによる読み出しまたは書き込みアクセスの際に、コマンドアサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロックサイクル数を定義します。

0000 0000 : 1 HCLK サイクル  
1111 1110 : 255 HCLK サイクル  
1111 1111 : 予約済み。

## I/O空間タイミングレジスタ 4 (FMC\_PIO4)

アドレスオフセット : 0xB0

リセット値 : 0xFCFCFCFC

FMC\_PIO4 読み出し/書き込みレジスタは、16 ビット PC カード/コンパクトフラッシュの I/O 空間へのアクセスするためのタイミング情報を格納します。

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

ビット 31:24 **IOHIZx** : I/O x データバスハイインピーダンス時間

ソケット x 上の I/O 空間への PC カード書き込みアクセスの開始後に、データバスをハイインピーダンスに保つ HCLK クロックサイクル数を定義します。これは、書き込みトランザクションに限り有効です。

0000 0000 : 0 HCLK サイクル

1111 1111 : 255 HCLK サイクル

ビット 23:16 **IOHOLDx** : I/O x ホールド時間

ソケット x 上の I/O 空間への PC カードによる読み出しまたは書き込みアクセスの際に、コマンドのアサーション解除 (NWE、NOE) 後にアドレスを (書き込みアクセス時はデータも) 保持する HCLK クロックサイクル数を定義します。

0000 0000 : 予約済み

0000 0001 : 1 HCLK サイクル

1111 1111 : 255 HCLK サイクル

ビット 15:8 **IOWAITx** : I/O x ウェイト時間

ソケット x 上の I/O 空間への PC カードによる読み出しまたは書き込みアクセスの際に、コマンド (SMNWE、SMNOE) をアサートするための HCLK (+1) クロックサイクルの最小数を定義します。プログラムされた HCLK 値の後にウェイト信号 (NWAIT) がアクティブ (ロー) の場合、コマンドアサーションの時間は延長されます。

0000 0000 : 予約済み。この値は使用しないでください。

0000 0001 : 2 HCLK サイクル (+NWAIT のデアサーションによって導入されるウェイトサイクル)

1111 1111 : 256 HCLK サイクル (+カードが NWAIT をデアサートすることによって導入されるウェイトサイクル)

ビット 7:0 **IOSETx** : I/O x セットアップ時間

ソケット x 上の I/O 空間への PC カードによる読み出しまたは書き込みアクセスの際に、コマンドアサーション (NWE、NOE) 前にアドレスをセットアップするための HCLK (+1) クロックサイクル数を定義します。

0000 0000 : 1 HCLK サイクル

1111 1111 : 256 HCLK サイクル (リセット後のデフォルト値)

## ECC リザルトレジスタ 2/3 (FMC\_ECCR2/3)

アドレスオフセット :  $0x54 + 0x20 * (x - 1)$ ,  $x = 2$  または  $3$

リセット値 :  $0x0000\ 0000$

これらのレジスタは、FSMC コントローラの ECC 計算モジュール (NAND フラッシュメモリバンクごとに 1 つのモジュール) によって計算された現在のエラー訂正コード値を格納します。CPU が、NAND フラッシュメモリページ上の正しいアドレスからデータを読み出すときには ( [セクション 10.6.6: エラー訂正コード \(ECC\) の計算 NAND 型フラッシュメモリ](#) を参照)、NAND フラッシュメモリに対して読み書きされたデータは、ECC 計算モジュールによって自動的に処理されます。X バイトを読み出した後 (FMC\_PCRx レジスタの ECCPS フィールドに従って)、CPU は FMC\_ECCx レジスタから計算された ECC 値を読み出す必要があります。次に、これらの計算されたパリティデータがスぺア領域に記録されているパリティ値と一致するか検証して、ページが有効か否か、訂正が必要か否かを確認します。FMC\_ECCRx レジスタは、読み出し後、ECCEN ビットを "0" にセットすることによってクリアする必要があります。新しいデータブロックを計算するときは、ECCEN ビットを "1" にセットする必要があります。

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

ビット 31:0 **ECCx** : ECC 結果

このフィールドは、ECC 計算ロジックによって計算された値を格納します。 [表 69](#) に、これらのビットフィールドの値を示します。

**表 69. ECC 結果関連ビット**

ECCPS[2:0]	ページサイズ (バイト数)	ECC ビット
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

## 10.7 FMC レジスタマップ

次の表に FMC レジスタの一覧を示します。

表 70. FMC レジスタマップ

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x00	FMC_BCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTMOD	EXTMOD	WAITEN	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	予約済み	FACCEN	MWID		MTYP	MUXEN	MBKEN				
0x08	FMC_BCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTMOD	EXTMOD	WAITEN	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	予約済み	FACCEN	MWID		MTYP	MUXEN	MBKEN				
0x10	FMC_BCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTMOD	EXTMOD	WAITEN	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	予約済み	FACCEN	MWID		MTYP	MUXEN	MBKEN				
0x18	FMC_BCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTMOD	EXTMOD	WAITEN	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	予約済み	FACCEN	MWID		MTYP	MUXEN	MBKEN				
0x04	FMC_BTR1	Res.	Res.	ACCMOD		DATLAT		CLKDIV		BUSTURN																												
0x0C	FMC_BTR2	Res.	Res.	ACCMOD		DATLAT		CLKDIV		BUSTURN																												
0x14	FMC_BTR3	Res.	Res.	ACCMOD		DATLAT		CLKDIV		BUSTURN																												
0x1C	FMC_BTR4	Res.	Res.	ACCMOD		DATLAT		CLKDIV		BUSTURN																												
0x104	FMC_BWTR1	Res.	Res.	ACCMOD																																		
0x10C	FMC_BWTR2	Res.	Res.	ACCMOD																																		
0x114	FMC_BWTR3	Res.	Res.	ACCMOD																																		
0x11C	FMC_BWTR4	Res.	Res.	ACCMOD																																		
0x60	FMC_PCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS	TAR		TCLR		Res.	Res.	ECCEN	PWID		PTYP	PBKEN	PWAITEN	Res.						
0x80	FMC_PCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS	TAR		TCLR		Res.	Res.	ECCEN	PWID		PTYP	PBKEN	PWAITEN	Res.						
0xA0	FMC_PCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS	TAR		TCLR		Res.	Res.	ECCEN	PWID		PTYP	PBKEN	PWAITEN	Res.						
0x64	FMC_SR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0x84	FMC_SR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
0xA4	FMC_SR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x68	FMC_PMEM2	MEMHIZx				MEMHOLDx				MEMWAITx				MEMSETx																								
0x88	FMC_PMEM3	MEMHIZx				MEMHOLDx				MEMWAITx				MEMSETx																								
0xA8	FMC_PMEM4	MEMHIZx				MEMHOLDx				MEMWAITx				MEMSETx																								



表 70. FMC レジスタマップ (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x6C	FMC_PATT2	ATTHIZx				ATTHOLDx				ATTWAITx				ATTSETx																			
0x8C	FMC_PATT3	ATTHIZx				ATTHOLDx				ATTWAITx				ATTSETx																			
0xAC	FMC_PATT4	ATTHIZx				ATTHOLDx				ATTWAITx				ATTSETx																			
0xB0	FMC_PIO4	IOHIZx				IOHOLDx				IOWAITx				IOSETx																			
0x74	FMC_ECCR2	ECCx																															
0x94	FMC_ECCR3	ECCx																															

## 11 汎用 I/O (GPIO)

### 11.1 概要

各汎用 I/O ポートにはそれぞれ 4 つの 32 ビット設定レジスタ (GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR)、2 つの 32 ビットデータレジスタ (GPIOx\_IDR、GPIOx\_ODR)、32 ビットセット/リセットレジスタ (GPIOx\_BSRR)、32 ビットロックレジスタ (GPIOx\_LCKR)、2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx\_AFRH、GPIOx\_AFLR) があります。

### 11.2 GPIO の主な機能

- 出力状態：プッシュプルまたはオープンドレイン + プルアップ/プルダウン
- 出力データレジスタ (GPIOx\_ODR) またはペリフェラル (オルタネート機能出力) からの出力データ
- 各 I/O のスピード選択
- 入力状態：フローティング、プルアップ/プルダウン、アナログ
- 入力データレジスタ (GPIOx\_IDR) またはペリフェラル (オルタネート機能入力) への入力データ
- GPIOx\_ODR へのビット単位の書き込みアクセス用のビットセット/リセットレジスタ (GPIOx\_BSRR)
- STM32F303xD/E および STM32F398xE のポート A、B、C、D、E、F、G、H、STM32F303xB/C および STM32F358xC デバイスの A、B、および D I/O 設定、および STM32F303x6/8 および STM32F328x8 デバイスのポート A、B、C、D、および F を凍結するロック機構 (GPIOx\_LCKR)。
- アナログ機能
- オルタネート機能選択レジスタ
- 2 クロックサイクルごとに変化可能な高速トグル
- 柔軟性の高いピンの多重化により、I/O ピンを GPIO またはいくつかのペリフェラル機能の 1 つとして使用可能

### 11.3 GPIO の機能説明

各 I/O ポートの特定のハードウェア特性については、データシートに記載されています。汎用 I/O (GPIO) ポートの各ポートビットは、ソフトウェアによって以下の動作モードを個別に設定できます。

- 入力フローティング
- 入力プルアップ
- 入力プルダウン
- アナログ
- プルアップまたはプルダウン機能を持つ出力オープンドレイン
- プルアップまたはプルダウン機能を持つ出力プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能オープンドレイン

各 I/O ポートビットは自由にプログラム可能ですが、I/O ポートレジスタには 32 ビットワード、ハーフワード、またはバイト単位でアクセスする必要があります。GPIOx\_BSRR および GPIOx\_BRR レジスタを使用すると、任意の GPIOx\_ODR レジスタに不可分な読み出し/変更アクセスを行うことができます。これにより、読み出しと変更アクセスの間に IRQ が発生するリスクを回避できます。

図 40 および 図 41 は、標準 I/O ポートビットと 5V トレラント I/O ポートビットの基本構成をそれぞれ示します。表 71 には、実行可能なポートビットの設定を示します。

図 40. I/O ポートビットの基本構成

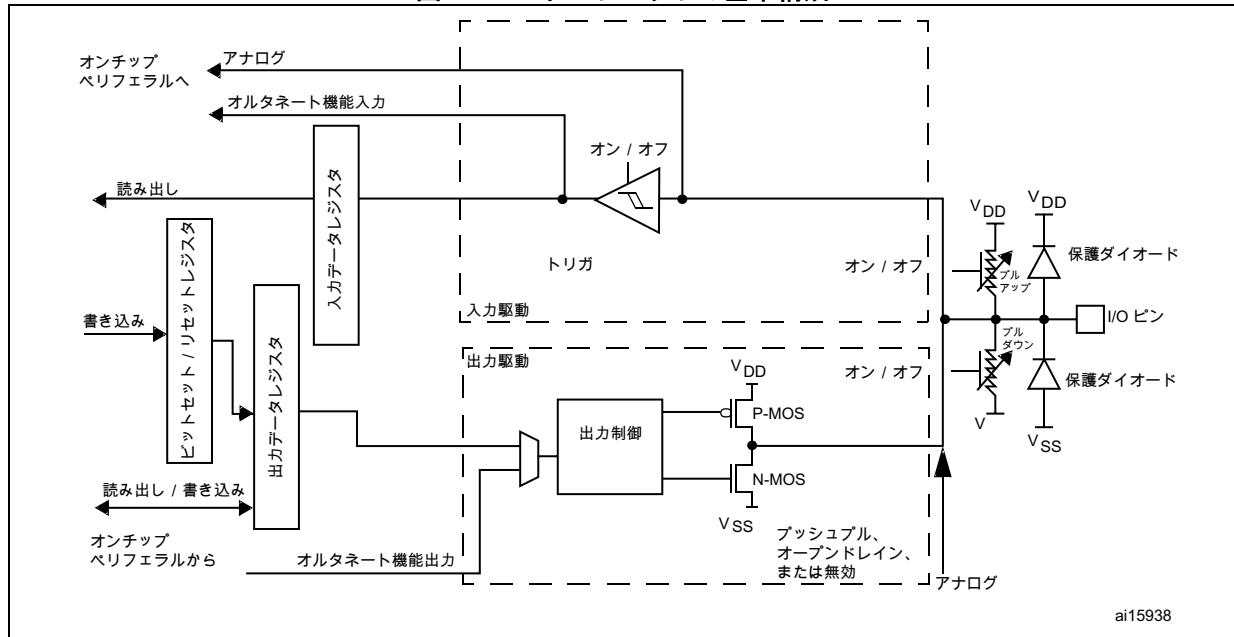
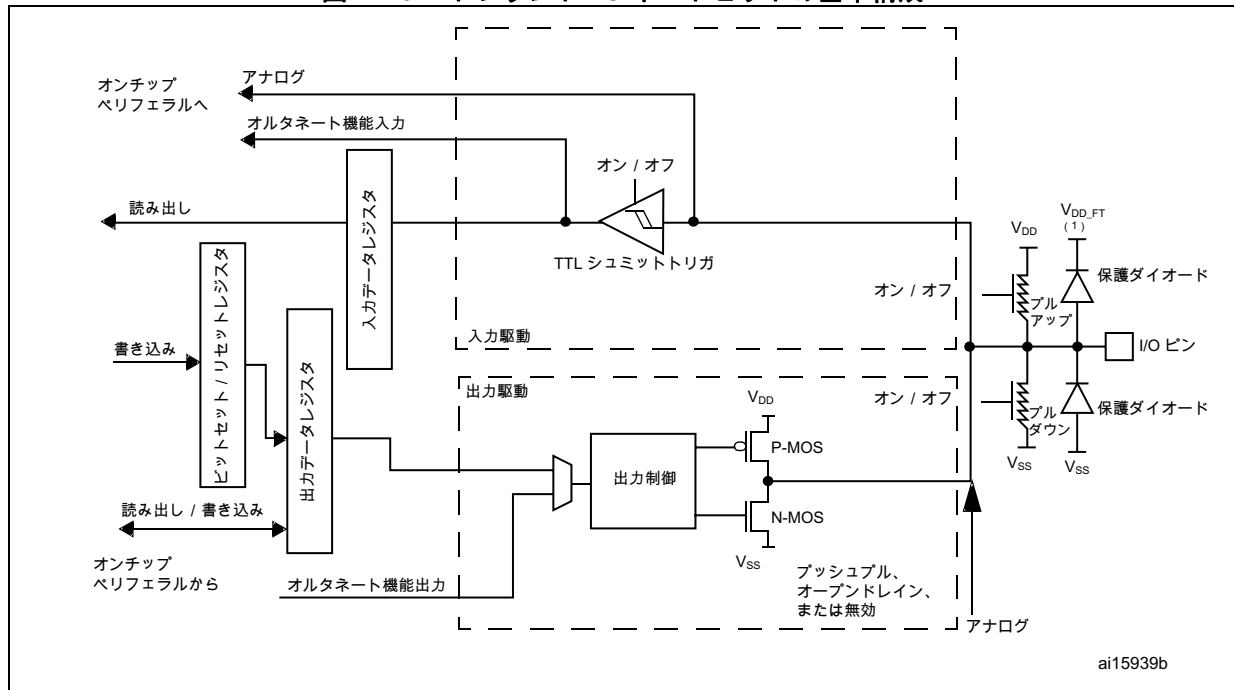


図 41. 5V トレラント I/O ポートビットの基本構成



1.  $V_{DD\_FT}$  は、5V トレラント I/O に固有の電位であり、 $V_{DD}$  とは異なります。

表 71. ポートビット設定表<sup>(1)</sup>

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [1:0]		PUPDR(i) [1:0]		I/O 設定	
01	0	SPEED [1:0]		0	0	GP 出力	PP
	0			0	1	GP 出力	PP + PU
	0			1	0	GP 出力	PP + PD
	0			1	1	予約済み	
	1			0	0	GP 出力	OD
	1			0	1	GP 出力	OD + PU
	1			1	0	GP 出力	OD + PD
	1			1	予約済み (GP 出力 OD)		
10	0	SPEED [1:0]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	予約済み	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	予約済み		
00	x	x	x	0	0	入力	フローティング
	x	x	x	0	1	入力	PU
	x	x	x	1	0	入力	PD
	x	x	x	1	1	予約済み (入力フローティング)	
11	x	x	x	0	0	入力/出力	アナログ
	x	x	x	0	1	予約済み	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 汎用、PP = プッシュプル、PU = プルアップ、PD = プルダウン、OD = オープンドレイン、AF = オルタネート機能

## 11.3.1 汎用 I/O (GPIO)

リセット中とリセット直後は、オルタネート機能は有効ではなく、ほとんどの I/O ポートは入力フローティングモードに設定されています。

リセット後、デバッグピンはオルタネート機能のプルアップ/プルダウンに設定されています。

- PA15 : JTDI プルアップ
- PA14 : JTCK/SWCLK プルダウン
- PA13 : JTMS/SWDIO プルアップ
- PB4 : NJTRST プルアップ
- PB3 : JTDO/TRACESWO

ピンが出力として設定されている場合、出力データレジスタ (GPIOx\_ODR) に書き込まれた値が I/O ピンに出力されます。出力ドライバをプッシュプルモードまたはオープンドレインモード (ローレベルのみが駆動され、ハイレベルはハイインピーダンス) で使用することができます。

入力データレジスタ (GPIOx\_IDR) は、AHB クロックサイクルごとに、I/O ピン上のデータをキャプチャします。

すべての GPIO ピンに、内部ウィークプルアップ抵抗とウィークプルダウン抵抗があり、GPIOx\_PUPDR レジスタの値によってこれらを有効化/無効化できます。

## 11.3.2 I/O ピンオルタネート機能マルチプレクサと配置

デバイスの I/O ピンは、マルチプレクサを介してオンボードのペリフェラル/モジュールに接続され、一度に 1 つのペリフェラルオルタネート機能 (AF) のみが 1 つの I/O ピンに接続可能となっています。この方法により、同じ I/O ピンを共有するペリフェラル間での競合を無くすることができます。

各 I/O ピンは、最大 16 のオルタネート機能入力 (AF0~AF15) を持つマルチプレクサを内蔵しており、GPIOx\_AFRL (ピン 0~7) と GPIOx\_AFRH (ピン 8~15) レジスタを介して設定することができます。

- リセット後、マルチプレクサの選択はオルタネート機能 0 (AF0) です。I/O は、GPIOx\_MODER レジスタを通してオルタネート機能モードで設定されます。
- 各ピンに固有のオルタネート機能割り当てについての詳細は、デバイスデータシートに記載されています。

この柔軟性の高い I/O 多重化アーキテクチャに加え、各ペリフェラルではオルタネート機能がそれぞれの I/O ピンに配置されており、さらに小型のパッケージで使用できるペリフェラルの数を最適化します。

I/O を任意の設定で使用するには、次の手順に従います。

- **デバッグ機能**: 各デバイスのリセット後、これらのピンはデバッグホストによってすぐに使用可能なオルタネート機能ピンとして割り当てられます。
- **GPIO**: 必要とする I/O を、GPIOx\_MODER レジスタで出力、入力、またはアナログとして設定します。
- **ペリフェラルオルタネート機能**
  - I/O を GPIOx\_AFRL または GPIOx\_AFRH レジスタのどちらかで必要とする AFx に接続します。
  - タイプ、プルアップ/プルダウン、出力スピードをそれぞれ GPIOx\_OTYPER、GPIOx\_PUPDR、GPIOx\_OSPEEDER レジスタで選択します。
  - 必要とする I/O を、GPIOx\_MODER レジスタでオルタネート機能として設定します。

- **追加機能：**

- ADC、DAC、OPAMP、および COMP については、GPIOx\_MODER レジスタのアナログモードで必要とする I/O を設定し、ADC、DAC、OPAMP、および COMP レジスタで必要な機能を設定します。
- RTC、WKUPx、オシレータなどの追加機能については、関連する RTC、PWR、および RCC レジスタで必要な機能を設定します。これらの機能は、標準の GPIO レジスタの設定よりも優先されます。

オルタネート機能 I/O ピンの配置に関する詳細は、デバイスデータシートの「オルタネート機能配置」表を参照してください。

### 11.3.3 I/O ポート制御レジスタ

各 GPIO ポートには 32 ビットメモリマップド制御レジスタが 4 つ (GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR) あり、最大 16 個の I/O を設定します。GPIOx\_MODER レジスタは I/O モード (入力、出力、AF、アナログ) を選択するために使用されます。GPIOx\_OTYPER および GPIOx\_OSPEEDR レジスタは、出力タイプ (プッシュプルまたはオープンドレイン) および速度を選択するために使用されます。I/O の方向がどちらであっても、GPIOx\_PUPDR レジスタは、プルアップ/プルダウンを選択するために使用されます。

### 11.3.4 I/O ポートデータレジスタ

各 GPIO には、16 ビットメモリマップドデータレジスタが 2 つあります。入力データレジスタ (GPIOx\_IDR) と出力データレジスタ (GPIOx\_ODR) です。GPIOx\_ODR は出力されるデータを格納し、読み出し/書き込みアクセスが可能です。I/O から入力されるデータは読み出し専用の入力データレジスタ (GPIOx\_IDR) に格納されます。

レジスタの説明は、[セクション 11.4.5 : GPIO ポート入力データレジスタ \(GPIOx\\_IDR\) \(x = A ~ H\)](#) および [セクション 11.4.6 : GPIO ポート出力データレジスタ \(GPIOx\\_ODR\) \(x = A ~ H\)](#) を参照してください。

### 11.3.5 I/O データのビット単位の操作

ビットセット/リセットレジスタ (GPIOx\_BSRR) は 32 ビットレジスタで、アプリケーションによる出力データレジスタ (GPIOx\_ODR) のビット単位のセット/リセットを可能にします。ビットセット/リセットレジスタは GPIOx\_ODR の 2 倍のサイズです。

GPIOx\_ODR の各ビットには GPIOx\_BSRR の 2 つの制御ビット BS(i) と BR(i) が対応します。ビット BS(i) に 1 を書き込むと、対応する ODR(i) ビットが**セット**されます。ビット BR(i) に 1 を書き込むと、対応する ODR(i) ビットが**リセット**されます。

GPIOx\_BSRR のいかなるビットに 0 を書き込んでも GPIOx\_ODR の対応するビットには影響しません。仮に、GPIOx\_BSRR のビットに対してセットおよびリセットの両方を実行しようとした場合、セット動作が優先されます。

GPIOx\_BSRR レジスタを使用した GPIOx\_ODR 内の個々のビットの変更には、1 回限りの効果しかなく、GPIOx\_ODR ビットを固定するものではありません。GPIOx\_ODR のビットは常に直接アクセスすることができます。GPIOx\_BSRR レジスタによって、ビット単位の不可分操作を行うことができます。

GPIOx\_ODR をビットレベルでプログラムする場合は、ソフトウェアで割り込みを無効にする必要はありません。1 回の不可分な AHB 書き込みアクセスで 1 ビットまたは複数ビットを変更することができます。

## 11.3.6 GPIO ロック機構

GPIOx\_LCKR レジスタへ特定の書き込みシーケンスを行うことにより、GPIO 制御レジスタをロックすることができます。ロックされるレジスタは、GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR、GPIOx\_AFRL、GPIOx\_AFRH です。

GPIOx\_LCKR レジスタに書き込むには、特定の書き込み／読み出しシーケンスを行う必要があります。このレジスタのビット 16 に適切な LOCK シーケンスを行う場合、LCKR[15:0] の値を使用して I/O の設定を固定します（この書き込みシーケンス中、LCKR[15:0] の値は同じである必要があります）。あるポートビットに LOCK シーケンスが行われると、次の MCU リセットまたはペリフェラルリセットまで、そのポートビットの値を変更できなくなります。GPIOx\_LCKR の各ビットによって、制御レジスタ（GPIOx\_MODER、GPIOx\_OTYPER、GPIOx\_OSPEEDR、GPIOx\_PUPDR、GPIOx\_AFRL、GPIOx\_AFRH）の対応するビットが停止されます。

GPIOx\_LCKR ビット 16 を [15:0] ビットと同時に設定する必要があるため、この LOCK シーケンス（[セクション 11.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx\\_LCKR\)](#) を参照）は、GPIOx\_LCKR レジスタへのワード（32 ビット長）アクセスを使用してのみ実行できます。

詳細については、[セクション 11.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx\\_LCKR\)](#) の LCKR レジスタの説明を参照してください。

## 11.3.7 I/O オルタネート機能の入力／出力

各 I/O が使用できるオルタネート機能入力／出力の 1 つを選択するため、2 つのレジスタが用意されています。これらのレジスタを使用し、必要に応じて、アプリケーションでオルタネート機能を他のピンに接続することができます。

つまり、GPIOx\_AFRL および GPIOx\_AFRH オルタネート機能レジスタを使用していくつかの使用可能なペリフェラル機能が、各 GPIO に多重化されることとなります。こうして各 I/O に使用可能な機能のどれか 1 つをアプリケーションによって選択できます。AF 選択信号はオルタネート機能入力およびオルタネート機能出力に共通なので、任意の I/O が持つオルタネート機能入力／出力に対し 1 つのチャンネルが選択されます。

どの機能が各 GPIO ピンに多重化されているかについてはデバイスデータシートを参照してください。

## 11.3.8 外部割り込み／ウェイクアップライン

すべてのポートに外部割り込み機能があります。外部割り込みラインを使用するには、ポートを入力モードに設定する必要があります。[セクション 14.2 : 拡張割り込み／イベントコントローラ \(EXTI\)](#) および [セクション 14.2.3 : ウェイクアップイベント管理](#) を参照してください。

## 11.3.9 入力設定

I/O ポートが入力としてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力の有効になります。
- GPIOx\_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。


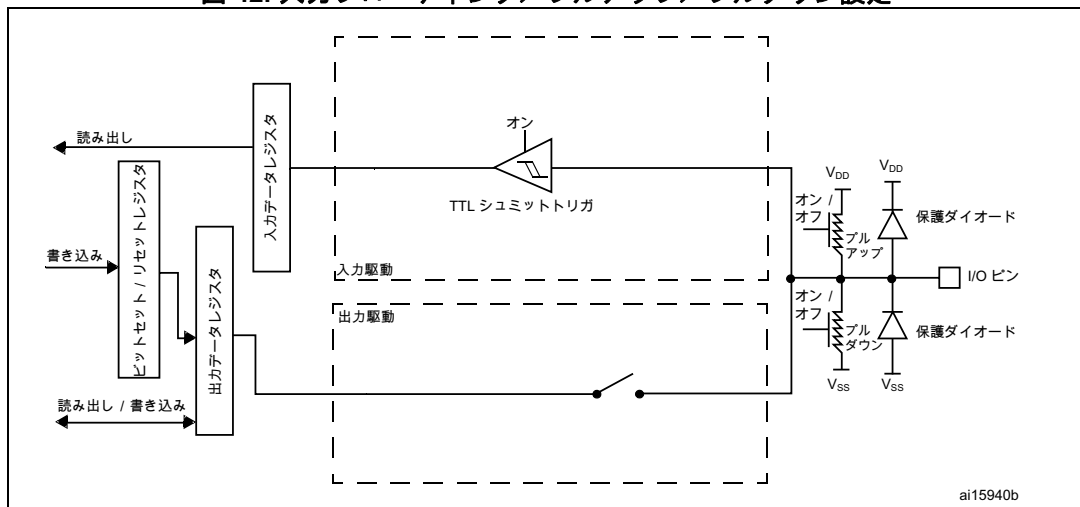
 [42](#) は、I/O ポートビットの入力設定を示しています。

図 42. 入力フローティング/プルアップ/プルダウン設定



## 11.3.10 出力設定

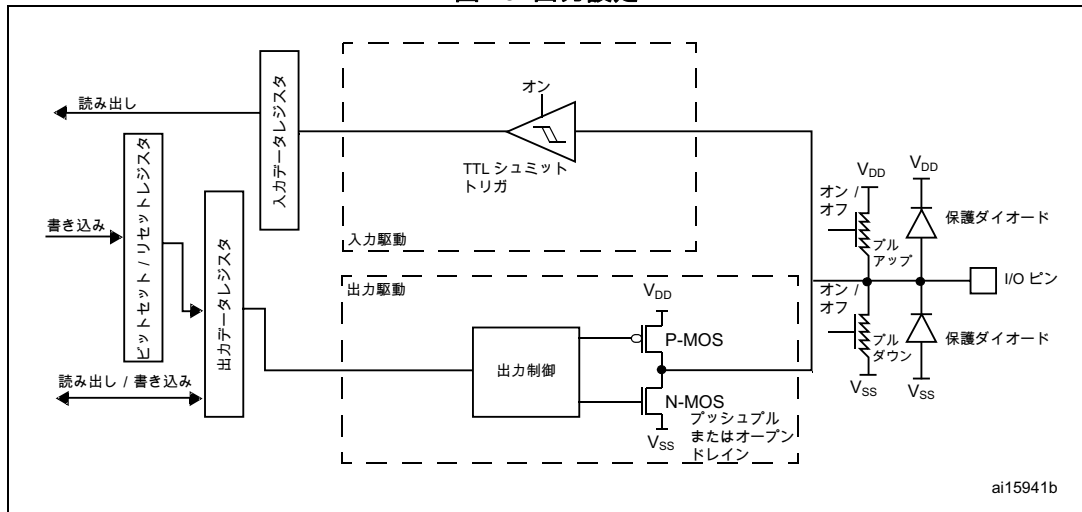
I/O ポートが出力としてプログラムされた場合、

- 出力バッファが有効になります。
  - オープンドレインモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときはポートはハイインピーダンス状態のままです (P-MOS が有効になることはない)。
  - プッシュプルモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときは P-MOS が有効になります。
- シュミットトリガ入力があります。
- GPIOx\_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。
- 出力データレジスタの読み出しアクセスによって、最後に書き込まれたデータが得られます。

図 43 は、I/O ポートビットの出力設定を示しています。



図 43. 出力設定



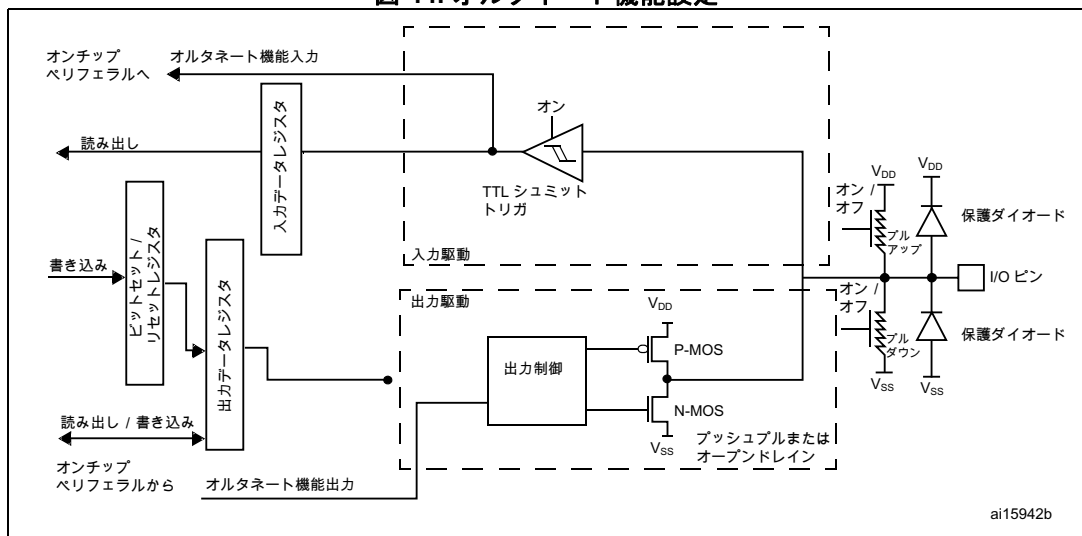
## 11.3.11 オルタネート機能設定

I/O ポートがオルタネート機能としてプログラムされた場合、

- 出力バッファをオープンドレインまたはプッシュプルモードに設定することができます。
- 出力バッファが、ペリフェラル (トランスミッタイネーブルおよびデータ) からの信号で駆動されます。
- シュミットトリガ入力が有効になります。
- ウィークプルアップ抵抗およびプルダウン抵抗が有効になるか否かは、GPIOx\_PUPDR レジスタの値によって決まります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

図 44 は、I/O ポートビットのオルタネート機能設定を示しています。

図 44. オルタネート機能設定



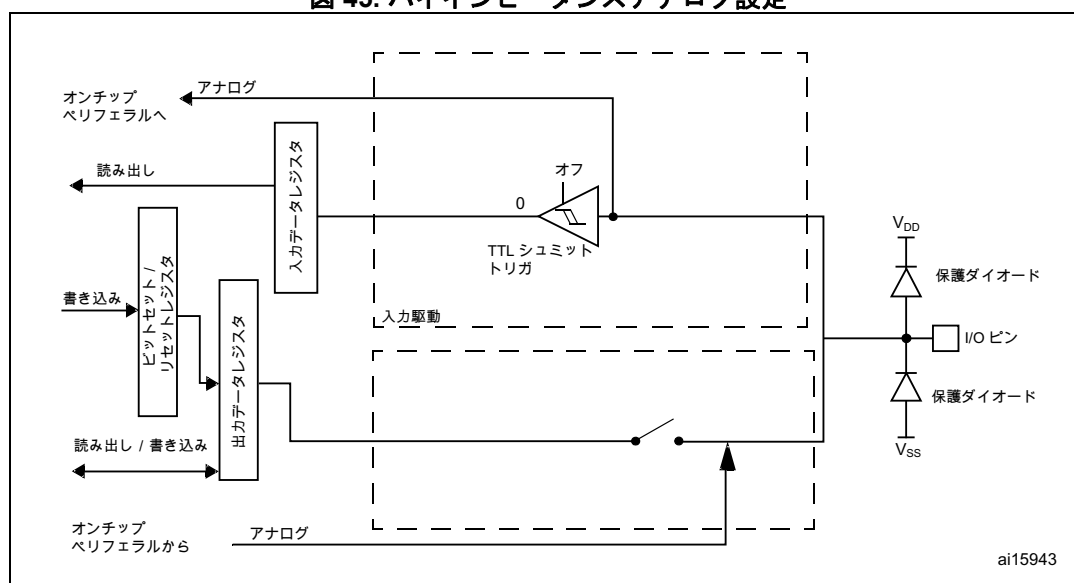
## 11.3.12 アナログ設定

I/O ポートがアナログとしてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力は無効になり、I/O ピンのどのようなアナログ値に対しても消費電力をゼロに抑えます。シュミットトリガ回路の出力は、常に“0”になります。
- ウィークプルアップ抵抗およびプルダウン抵抗はハードウェアによって無効にされます。
- 入力データレジスタの読み出しアクセスを行うと、値“0”が得られます。

図 45 は、I/O ポートビットのハイインピーダンスアナログ入力設定を示しています。

図 45. ハイインピーダンスアナログ設定



## 11.3.13 HSEまたはLSE オシレータのピンを GPIO として使用

HSE または LSE オシレータがスイッチオフされた場合（リセット後のデフォルト状態）、関連のオシレータピンを通常の GPIO として使用することができます。

HSE または LSE オシレータがスイッチオンされた場合（RCC\_CSR レジスタの HSEON または LSEON ビットを設定することで）、オシレータは関連ピンを制御しますが、これらのピンの GPIO 設定は無効です。

オシレータがユーザ外部クロックモードに設定されている場合、ピンのみがクロック入力のために確保されますが、OSC\_OUT または OSC32\_OUT ピンは通常の GPIO として使用することができます。

## 11.3.14 GPIO ピンを RTC 供給ドメインで使用

コア供給ドメインの電源がオフになったとき（デバイスが STANDBY モードに移行したとき）、PC13/PC14/PC15 の GPIO が機能しなくなります。この場合、それらの GPIO 設定が RTC 設定によってバイパスされなければ、これらのピンはアナログ入力モードに設定されます。

RTC による I/O 制御の詳細は、[セクション 27.3:RTC の機能説明 \(774 ページ\)](#) を参照してください。

## 11.4 GPIO レジスタ

このセクションには、GPIO レジスタの詳細な説明が記載されています。

レジスタビット、レジスタアドレスオフセット、リセット値の概要は、表 72 を参照してください。

ペリフェラルレジスタはワード、ハーフワード、バイトのいずれかのモードで書き込むことができます。

### 11.4.1 GPIO ポートモードレジスタ (GPIOx\_MODER) (x = A ~ H)

アドレスオフセット : 0x00

リセット値 :

- ポート A 0xA800 0000
- ポート B 0x0000 0280
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット  $2y+1:2y$  MODERy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O モードを設定するために、ソフトウェアによって書き込まれます。

- 00 : 入力モード (リセット状態)
- 01 : 汎用出力モード
- 10 : オルタネート機能モード
- 11 : アナログモード

**注 :** STM32F303xB/xC および STM32F358x デバイスでは、GPIOF\_MODER のビット 10 および 11 は予約済みであり、リセット状態に保持する必要があります。

### 11.4.2 GPIO ポート出力タイプレジスタ (GPIOx\_OTYPER) (x = A ~ H)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **OTy** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O 出力タイプを設定するために、ソフトウェアによって書き込まれます。

- 0 : 出力プッシュプル (リセット状態)
- 1 : 出力オープンドレイン

### 11.4.3 GPIO ポート出力スピードレジスタ (GPIOx\_OSPEEDR) (x = A ~ H)

アドレスオフセット : 0x08

リセット値 :

- ポート A 0x0C00 0000
- ポート B 0x0000 00C0
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3 [1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 2y+1:2y **OSPEEDRy[1:0]** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O の出力スピードを設定するために、ソフトウェアによって書き込まれます。

- x0 : ロースピード
- 01 : ミディアムスピード
- 11 : ハイスピード

**注 :** 周波数仕様、およびスピード別の電源や負荷条件については、デバイスデータシートを参照してください。

### 11.4.4 GPIO ポートプルアップ/プルダウンレジスタ (GPIOx\_PUPDR) (x = A ~ H)

アドレスオフセット : 0x0C

リセット値 :

- ポート A 0x6400 0000
- ポート B 0x0000 0100
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 2y+1:2y PUPDRy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O のプルアップまたはプルダウンを設定するために、ソフトウェアによって書き込まれます。

- 00 : プルアップ/プルダウンなし
- 01 : プルアップ
- 10 : プルダウン
- 11 : 予約済み

## 11.4.5 GPIO ポート入力データレジスタ (GPIOx\_IDR) (x = A ~ H)

アドレスオフセット : 0x10

リセット値 : 0x0000 XXXX (X は未定義)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 IDRy : ポート入力データビット (y = 0~15)

これらのビットは読み出し専用です。これらのビットには、対応する I/O ポートの入力値が格納されています。

## 11.4.6 GPIO ポート出力データレジスタ (GPIOx\_ODR) (x = A ~ H)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 ODRy : ポート出力データビット (y = 0~15)

これらのビットは、ソフトウェアによって読み出し/書き込みができます。

**注 :** 最小なビット単位のセット/リセットのために、GPIOx\_BSRR または GPIOx\_BRR レジスタ (x = A ~ F) への書き込みによって、ODR ビットを個々にセット/リセットすることができます。



## 11.4.7 GPIO ポートビットセット/リセットレジスタ (GPIOx\_BSRR) (x = A ~ H)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 **BRy** : ポート x リセットビット y (y=0~15)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODRx ビットの値は変化しません。

1 : 対応する ODRx ビットをリセットします。

**注 :** **BSx** ビットと **BRx** ビットの両方がセットされた場合、**BSx** ビットが優先されます。

ビット 15:0 **BSy** : ポート x セットビット y (y=0~15)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODRx ビットの値は変化しません。

1 : 対応する ODRx ビットをセットします。

## 11.4.8 GPIO ポート設定ロックレジスタ (GPIOx\_LCKR)

STM32F303xB/C および STM32F358xC デバイスでは x= A、B、D、STM32F303x6/8 および STM32F328x8 デバイスでは x= A、B、C、D、F、および STM32F303xD/E デバイスでは x= A、B、C、D、E、F、G、H です。

このレジスタは、ビット 16 (LCKK) に正しい書き込みシーケンスが行われたときに、ポートビットの設定をロックするために使用されます。ビット [15:0] の値は、GPIO の設定をロックするために使用されます。書き込みシーケンスの間は、LCKR[15:0] の値を変更することはできません。あるポートビットに LOCK シーケンスが適用されると、次の MCU リセットまたはペリフェラルリセットまで、このポートビットの値を変更できなくなります。

**注 :** **GPIOx\_LCKR** レジスタへの書き込みには特定の書き込みシーケンスが使われます。このロックシーケンスではワードアクセス (32 ビット長) のみ可能です。

各ロックビットによって、特定の設定レジスタ (制御レジスタおよびオルタネート機能レジスタ) が固定されます。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

**ビット 16 LCKK** : ロックキー

このビットは常時読み出しができます。ロックキー書き込みシーケンスを使用しなければ変更できません。

0 : ポート設定ロックキーは無効です。

1 : ポート設定ロックキーは有効です。GPIOx\_LCKR レジスタは、次の MCU リセットまたはペリフェラルリセットまでロックされます。

ロックキー書き込みシーケンス :

書き込み LCKR[16] = "1" + LCKR[15:0]

書き込み LCKR[16] = "0" + LCKR[15:0]

書き込み LCKR[16] = "1" + LCKR[15:0]

読み出し LCKR

読み出し LCKR[16] = "1" (この読み出し操作は任意だが、ロックが有効であることを確認できる。)

**注 :** **ロックキー書き込みシーケンスの間は、LCK[15:0] の値を変更することはできません。**

**ロックシーケンス中にエラーが発生すると、ロックは中止されます。**

**ポートの任意のビットの最初のロックシーケンスの後、次の MCU リセットまたはペリフェラルリセットまでは、LCKK ビットのいかなる読み出しアクセスに対しても、“1”が返されます。**

**ビット 15:0 LCKy** : ポート x ロックビット y (y= 0~15)

これらのビットは読み出し／書き込みができますが、書き込めるのは LCKK ビットが“0”のときだけです。

0 : ポート設定はロックされません。

1 : ポート設定はロックされます。

## 11.4.9 GPIO オルタネート機能下位レジスタ (GPIOx\_AFRL) (x = A ~ H)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

**ビット 31:0 AFRy[3:0]** : ポート x ピン y (y= 0~7) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRy 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15



## 11.4.10 GPIO オルタネート機能上位レジスタ (GPIOx\_AFRH) (x = A ~ H)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **AFRy[3:0]** : ポート x ピン y (y = 8~15) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRy 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15

## 11.4.11 GPIO ポートビットリセットレジスタ (GPIOx\_BRR) (x=A ~ H)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 予約済み

ビット 15:0 **BRy** : ポート x リセットビット y (y = 0 ~ 15)

これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。

0 : 対応する ODx ビットの値は変化しません。

1 : 対応する ODx ビットをリセットします。



## 11.4.12 GPIO レジスタマップ

次の表に、GPIO レジスタマップとリセット値を示します。

表 72. GPIO レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	GPIOB_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0x00	GPIOx_MODER (x = C ~ H)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	GPIOx_OTYPER (x = A ~ H)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOA_OSPEEDR	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	リセット値	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOB_OSPEEDR	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0x08	GPIOx_OSPEEDR (x = C ~ H)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	GPIOA_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	リセット値	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



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

表 72. GPIO レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C	GPIOB_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0x10	GPIOx_IDR (x = A ~ H)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	リセット値																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	GPIOx_ODR (x = A ~ H)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (x = A ~ H)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	GPIOx_LCKR (x = <sup>(1)</sup> を参照)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFR1 (x = C ~ H)	AFRLAFR7[3:0]			AFRLAFR6[3:0]			AFRLAFR5[3:0]			AFRLAFR4[3:0]			AFRLAFR3[3:0]			AFRLAFR2[3:0]			AFRLAFR1[3:0]			AFRLAFR0[3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (x = A ~ H)	AFRHAFR15[3:0]			AFRHAFR14[3:0]			AFRHAFR13[3:0]			AFRHAFR12[3:0]			AFRHAFR11[3:0]			AFRHAFR10[3:0]			AFRHAFR9[3:0]			AFRHAFR8[3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	GPIOx_BRR (x = A ~ H)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. STM32F303xB/C および STM32F358xC では A、B、D、STM32F303x6/8 および STM32F328x8 では A、B、C、D、F、および STM32F303xD/E では A、B、C、D、E、F、G、H です。

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



## 12 システム設定コントローラ (SYSCFG)

STM32F3xx デバイスは設定レジスタ群を持っています。システム設定コントローラの主な目的は次の通りです。

- 一部の I/O ポート上での I<sup>2</sup>C Fm+ の有効化/無効化
- TIM16、TIM17、TIM6、DAC1\_CH1、および DAC1\_CH2、TIM7、および ADC4 の一部の DMA トリガソースの、異なる DMA チャンネル (STM32F303x6/8 および STM32F328x8 の SPI1、I2C1、ADC2 も含む) への再配置
- コード領域の先頭に配置されたメモリの再配置
- 外部割り込みラインのGPIO への接続の管理
- TIM1 ITR3 ソースの再配置
- USB 割り込みラインの再配置
- DAC1 および DAC2 トリガの再配置
- 堅牢性の管理機能
- エンコーダモードの設定
- CCM SRAM ページ保護

### 12.1 SYSCFG レジスタ

#### 12.1.1 SYSCFG 設定レジスタ 1 (SYSCFG\_CFGR1)

このレジスタは、メモリ再配置時の特定の設定に使用されます。

アドレス 0x0000 0000 でアクセス可能なメモリのタイプを設定するために、2 つのビットが使用されます。これらのビットを用いて、ソフトウェアによる物理的な再配置を選択すると、BOOT ピンおよびオプションビット設定はバイパスされます。

リセット後、これらのビットは、BOOT ピン (BOOT0) およびオプションビット (BOOT1) で選択された値になります。

アドレスオフセット : 0x00

リセット値 : 0x7C00 000X (X は BOOT0 ピンおよび BOOT1 オプションビットで選択されたメモリモード)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FPU_IE[5..0]						Res.	I2C3_FMP	ENCODER_MODE	I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP	
r/w	r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC2_CH1_DMA_RMP <sup>(1)</sup>	TIM7_DAC2_DMA_RMP	TIM6_DAC1_DMA_RMP	TIM17_DMA_RMP	TIM16_DMA_RMP	Res.	Res.	ADC2_DMA_RMP	DAC_TRIG_RMP	TIM1_ITR3_RMP	USB_IT_RMP	Res.	Res.	MEM_MODE <sup>(2)</sup>	MEM_MODE	
r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w	r/w				r/w	r/w

1. STM32F303x6/8 および STM32F328x8 のみ。
2. STM32F303xD/E および STM32F398xE デバイスのみ。



ビット 31:26 **FPU\_IE[5..0]** : 浮動小数点演算装置割り込みイネーブルビット

FPU\_IE[5] : 不正確割り込みイネーブル  
FPU\_IE[4] : 入力正常割り込みイネーブル  
FPU\_IE[3] : オーバーフロー割り込みイネーブル  
FPU\_IE[2] : アンダーフロー割り込みイネーブル  
FPU\_IE[1] : 0 分周割り込みイネーブル  
FPU\_IE[0] : 無効操作割り込みイネーブル

ビット 25 : 予約済みであり、リセット値に保持する必要があります。

ビット 24 **I2C3\_FMP** : I2C3 高速モードプラス駆動機能有効化 (STM32F303xD/E デバイスのみ)

このビットは、ソフトウェアによってセット/クリアされます。AF 選択ビットを通じて選択された I2C3 ピンの Fm+ を有効にします。

0 : AF 選択ビットを通じて選択された I2C3 ピンで、Fm+ モードは無効です。

1 : AF 選択ビットを通じて選択された I2C3 ピンで、Fm+ モードが有効になります。

ビット 23:22 **ENCODER\_MODE** : エンコーダモード

このビットは、ソフトウェアによってセット/クリアされます。

00 : 変更なし。

01 : TIM2 IC1 および TIM2 IC2 はそれぞれ TIM15 IC1 および TIM15 IC2 に接続されています。

10 : TIM3 IC1 および TIM3 IC2 はそれぞれ TIM15 IC1 および TIM15 IC2 に接続されています。

11 : TIM4 IC1 および TIM4 IC2 はそれぞれ TIM15 IC1 および TIM15 IC2 に接続されています (デバイスのみ)。

STM32F303xB/C および STM32F358xC

ビット 21 **I2C2\_FMP** : I2C2 高速モードプラス駆動機能有効化 (STM32F303xB/C および STM32F358xC デバイスのみ)

このビットは、ソフトウェアによってセット/クリアされます。AF 選択ビットを通じて選択された I2C2 ピンの Fm+ を有効にします。

0 : AF 選択ビットを通じて選択された I2C2 ピンで、Fm+ モードは無効です。

1 : AF 選択ビットを通じて選択された I2C2 ピンで、Fm+ モードが有効になります。

ビット 20 **I2C1\_FMP** : I2C1 Fm+ 駆動機能有効化

このビットは、ソフトウェアによってセット/クリアされます。AF 選択ビットを通じて選択された I2C1 ピンの Fm+ を有効にします。

0 : AF 選択ビットを通じて選択された I2C1 ピンで、Fm+ モードは無効です。

1 : AF 選択ビットを通じて選択された I2C1 ピンで、Fm+ モードが有効になります。

ビット 19:16 **I2C\_PbX\_FMP** : パッド上の Fm+ 駆動機能有効化

これらのビットは、ソフトウェアによってセット/クリアされます。各ビットは、PB6、PB7、PB8、および PB9 の I/O で I<sup>2</sup>C Fm+ モードを有効にします。

0 : PBx ピンは標準モード (Sm) で動作します (x = 6 ~ 9)。

1 : PBx ピンで I<sup>2</sup>C Fm+ モードを有効にし、速度制御をバイパスします。

ビット 15 **DAC2\_CH1\_DMA\_RMP** : DAC2 チャンネル 1 DMA 再配置 (STM32F303x6/8 および STM32F328x8 デバイスのみ)

このビットは、ソフトウェアによってセット/クリアされます。DAC2 チャンネル 1 の DMA リクエストの再配置を制御します。

0 : 再配置なし

1 : 再配置 (DAC2\_CH1 の DMA リクエストは DMA1 チャンネル 5 に配置される)

**注 :** STM32F303x6/8 および STM32F328x8 では、このビットをセットする必要があります。

ビット 14 **TIM7\_DAC1\_CH2\_DMA\_RMP** : TIM7 および DAC チャンネル 2 DMA 再配置

このビットは、ソフトウェアによってセット/クリアされます。TIM7(UP) および DAC チャンネル 2 の DMA リクエストの再配置を制御します。

0 : 再配置なし (TIM7\_UP および DAC\_CH2 の DMA リクエストは、STM32F303xB/C および STM32F358xC デバイスの DMA2 チャンネル 4 に配置される)

1 : 再配置 (TIM7\_UP および DAC\_CH2 の DMA リクエストは、DMA1 チャンネル 4 に配置される)

**注 :** *STM32F303x6/8 および STM32F328x8 製品には DMA2 がいないため、このビットをセットする必要があります。*

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **TIM6\_DAC1\_CH1\_DMA\_RMP** : TIM6 および DAC チャンネル 1 DMA 再配置

このビットは、ソフトウェアによってセット/クリアされます。TIM6 (UP) および DAC チャンネル 1 の DMA リクエストの再配置を制御します。

0 : 再配置なし (TIM6\_UP および DAC\_CH1 の DMA リクエストは、STM32F303xB/C および STM32F358xC の DMA2 チャンネル 3 に配置される)

1 : 再配置 (TIM6\_UP および DAC\_CH1 の DMA リクエストは、DMA1 チャンネル 3 に配置される)

**注 :** *STM32F303x6/8 および STM32F328x8 製品には DMA2 がいないため、このビットをセットする必要があります。*

ビット 12 **TIM17\_DMA\_RMP** : TIM17 の DMA リクエスト再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。TIM17 の DMA リクエストの再配置を制御します。

0 : 再配置なし (DMA1 チャンネル 1 に配置された TIM17\_CH1 と TIM17\_UP の DMA リクエスト)

1 : 再配置 (DMA1 チャンネル 7 に配置された TIM17\_CH1 と TIM17\_UP の DMA リクエスト)

ビット 11 **TIM16\_DMA\_RMP** : TIM16 の DMA リクエスト再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。TIM16 の DMA リクエストの再配置を制御します。

0 : 再配置なし (DMA1 チャンネル 3 に配置された TIM16\_CH1 と TIM16\_UP の DMA リクエスト)

1 : 再配置 (DMA1 チャンネル 6 に配置された TIM16\_CH1 と TIM16\_UP の DMA リクエスト)

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ADC2\_DMA\_RMP** : ADC2 の DMA 再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。ADC24 の DMA リクエストの再配置を制御します。

0 : 再配置なし (ADC24 の DMA リクエストは DMA2 チャンネル 1 および 2 に配置される)

1 : 再配置 (ADC24 の DMA リクエストは DMA2 チャンネル 3 および 4 に配置される)

ビット 7 **DAC1\_TRIG\_RMP** : DAC トリガ再配置 (TSEL = 001 の場合) このビットは、ソフトウェアによってセット/クリアされます。DAC トリガソースの配置を制御します。

0 : 再配置なし (STM32F303xB/C および STM32F358xC デバイスの DAC トリガは TIM8\_TRGO)

1 : 再配置 (DAC トリガは TIM3\_TRGO)

ビット 6 **TIM1\_ITR3\_RMP** : タイマ 1 ITR3 選択

このビットは、ソフトウェアによってセット/クリアされます。TIM1 ITR3 の配置を制御します。

0 : 再配置なし (STM32F303xB/C および STM32F358xC デバイスの TIM1\_ITR3 = TIM4\_TRGO)

1 : 再配置 (TIM1\_ITR3 = TIM17\_OC)

ビット 5 **USB\_IT\_RMP** : USB 割り込み再配置 (STM32F303xB/C/D/E デバイスのみ)

このビットは、ソフトウェアによってセット/クリアされます。USB 割り込み配置を制御します。

0 : USB\_HP、USB\_LP、および USB\_WAKEUP 割り込みは、それぞれ割り込みライン 19、20、および 42 に配置されます。

1 : USB\_HP、USB\_LP、および USB\_WAKEUP 割り込みは、それぞれ割り込みライン 74、75、および 76 に配置されます。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **MEM\_MODE** : メモリ配置選択ビット

このビットは、ソフトウェアによってセット/クリアされます。アドレス 0x0000 0000 のメモリ内部配置を制御します。リセット後、これらのビットは、BOOT0 ピンと BOOT1 オプションビットで選択されたメモリ配置を行います。

0x0 : 0x0000 0000 にメインフラッシュメモリがマッピングされます。

001 : 0x0000 0000 にシステムフラッシュメモリがマッピングされます。

011 : (D-Code バス上の) 内蔵 SRAM が 0x0000 0000 に配置されます。

1xx : FMC バンク (最初の 2 つのバンクのみ) (STM32F303xD/E でのみ使用可能)

## 12.1.2 SYSCFG CCM SRAM 保護レジスタ (SYSCFG\_RCR)

CCM SRAM のサイズは 8 KB で、STM32F303xB/C および STM32F358xC デバイスでは 8 ページ (各 1 KB) で構成されます。CCM SRAM のサイズは 4 KB で、STM32F303x6/8 および STM32F328x8 デバイスでは 4 ページ (各 1 KB) で構成されます。CCM SRAM のサイズは 16 KB で、STM32F303xD/E デバイスでは 16 ページ (各 1 KB) で構成されます。

各ページを書き込み保護可能です。

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAGE15 _WP <sup>(1)</sup>	PAGE14 _WP <sup>(1)</sup>	PAGE13 _WP <sup>(1)</sup>	PAGE12 _WP <sup>(1)</sup>	PAGE11 _WP <sup>(1)</sup>	PAGE10 _WP <sup>(1)</sup>	PAGE9 _WP <sup>(1)</sup>	PAGE8 _WP <sup>(1)</sup>	PAGE7 _WP <sup>(2)</sup>	PAGE6 _WP <sup>(2)</sup>	PAGE5 _WP <sup>(2)</sup>	PAGE4 _WP <sup>(2)</sup>	PAGE 3_WP	PAGE 2_WP	PAGE 1_WP	PAGE 0_WP
												rW	rW	rW	rW

1. STM32F303xD/E および STM32F398xE のみ。

2. STM32F303xB/C/D/E デバイスのみ。

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **PAGEx\_WP** (x = 0 から 15) : CCM SRAM ページ書き込み保護ビット

これらのビットは、ソフトウェアによってセットされます。これらは、システムリセットによるのみクリアされます。

0 : ページ x の書き込み保護は無効です。

1 : ページ x の書き込み保護は有効です。

## 12.1.3 SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG\_EXTICR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **EXTI3[3:0]** : EXTI 3 設定ビット

これらのビットは、EXTI3 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[3] ピン
- x001 : PB[3] ピン
- x010 : PC[3] ピン
- x011 : PD[3] ピン
- x100 : PE[3] ピン
- x101 : PF[3] ピン
- x110 : PG[3] ピン
- 他の設定 : 予約済み

ビット 11:8 **EXTI2[3:0]** : EXTI 2 設定ビット

これらのビットは、EXTI2 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[2] ピン
- x001 : PB[2] ピン
- x010 : PC[2] ピン
- x011 : PD[2] ピン
- x100 : PE[2] ピン
- x101 : PF[2] ピン
- x110 : PG[2] ピン
- x111 : PH[2] ピン
- 他の設定 : 予約済み

ビット 7:4 **EXTI1[3:0]** : EXTI 1 設定ビット

これらのビットは、EXTI1 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[1] ピン
- x001 : PB[1] ピン
- x010 : PC[1] ピン
- x011 : PD[1] ピン
- x100 : PE[1] ピン
- x101 : PF[1] ピン
- x110 : PG[1] ピン
- x111 : PH[1] ピン
- 他の設定 : 予約済み

ビット 3:0 **EXTI0[3:0]** : EXTI 0 設定ビット

これらのビットは、EXTI0 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[0] ピン
- x001 : PB[0] ピン
- x010 : PC[0] ピン
- x011 : PD[0] ピン
- x100 : PE[0] ピン
- x101 : PF[0] ピン
- x110 : PG[0] ピン
- x111 : PH[0] ピン

注 :  他の設定 : 予約済み

## 12.1.4 SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG\_EXTICR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **EXTI7[3:0]** : EXTI 7 設定ビット

これらのビットは、EXTI7 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[7] ピン

x001 : PB[7] ピン

x010 : PC[7] ピン

x011 : PD[7] ピン

x100 : PE[7] ピン

x101 : PF[7] ピン

x110 : PG[7] ピン

他の設定 : 予約済み

ビット 11:8 **EXTI6[3:0]** : EXTI 6 設定ビット

これらのビットは、EXTI6 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[6] ピン

x001 : PB[6] ピン

x010 : PC[6] ピン

x011 : PD[6] ピン

x100 : PE[6] ピン

x101 : PF[6] ピン

x110 : PG[6] ピン

他の設定 : 予約済み

ビット 7:4 **EXTI5[3:0]** : EXTI 5 設定ビット

これらのビットは、EXTI5 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[5] ピン

x001 : PB[5] ピン

x010 : PC[5] ピン

x011 : PD[5] ピン

x100 : PE[5] ピン

x101 : PF[5] ピン

x110 : PG[5] ピン

他の設定 : 予約済み

ビット 3:0 **EXTI4[3:0]** : EXTI 4 設定ビット

これらのビットは、EXTI4 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[4] ピン

x001 : PB[4] ピン

x010 : PC[4] ピン

x011 : PD[4] ピン

x100 : PE[4] ピン

x101 : PF[4] ピン

x110 : PG[4] ピン

他の設定 : 予約済み

**注 :** *上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。*



## 12.1.5 SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG\_EXTICR3)

アドレスオフセット : 0x10

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **EXTI11[3:0]** : EXTI 11 設定ビット

これらのビットは、EXTI11 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[11] ピン
- x001 : PB[11] ピン
- x010 : PC[11] ピン
- x011 : PD[11] ピン
- x100 : PE[11] ピン
- x101 : PF[11] ピン
- x110 : PG[11] ピン
- 他の設定 : 予約済み

**ビット 11:8 EXTI10[3:0] : EXTI 10 設定ビット**

これらのビットは、EXTI10 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[10] ピン  
x001 : PB[10] ピン  
x010 : PC[10] ピン  
x011 : PD[10] ピン  
x100 : PE[10] ピン  
x101 : PF[10] ピン  
x110 : PG[10] ピン  
他の設定 : 予約済み

**ビット 7:4 EXTI9[3:0] : EXTI 9 設定ビット**

これらのビットは、EXTI9 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[9] ピン  
x001 : PB[9] ピン  
x010 : PC[9] ピン  
x011 : PD[9] ピン  
x100 : PE[9] ピン  
x101 : PF[9] ピン  
x110 : PG[9] ピン  
他の設定 : 予約済み

**ビット 3:0 EXTI8[3:0] : EXTI 8 設定ビット**

これらのビットは、EXTI8 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[8] ピン  
x001 : PB[8] ピン  
x010 : PC[8] ピン  
x011 : PD[8] ピン  
x100 : PE[8] ピン  
x101 : PF[8] ピン  
x110 : PG[8] ピン  
他の設定 : 予約済み

**注 :** *上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。*

## 12.1.6 SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG\_EXTICR4)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **EXTI15[3:0]** : EXTI15 設定ビット

これらのビットは、EXTI15 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[15] ピン
- x001 : PB[15] ピン
- x010 : PC[15] ピン
- x011 : PD[15] ピン
- x100 : PE[15] ピン
- x101 : PF[15] ピン
- x110 : PG[15] ピン
- 他の設定 : 予約済み

ビット 11:8 **EXTI14[3:0]** : EXTI14 設定ビット

これらのビットは、EXTI14 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[14] ピン
- x001 : PB[14] ピン
- x010 : PC[14] ピン
- x011 : PD[14] ピン
- x100 : PE[14] ピン
- x101 : PF[14] ピン
- x110 : PG[14] ピン
- 他の設定 : 予約済み

ビット 7:4 **EXTI13[3:0]** : EXTI13 設定ビット

これらのビットは、EXTI13 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[13] ピン
- x001 : PB[13] ピン
- x010 : PC[13] ピン
- x011 : PD[13] ピン
- x100 : PE[13] ピン
- x101 : PF[13] ピン
- x110 : PG[13] ピン
- 他の設定 : 予約済み

ビット 3:0 **EXTI12[3:0]** : EXTI12 設定ビット

これらのビットは、EXTI12 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[12] ピン
- x001 : PB[12] ピン
- x010 : PC[12] ピン
- x011 : PD[12] ピン
- x100 : PE[12] ピン
- x101 : PF[12] ピン
- x110 : PG[12] ピン
- 他の設定 : 予約済み

**注 :** *上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。*

## 12.1.7 SYSCFG 設定レジスタ 2 (SYSCFG\_CFGR2)

アドレスオフセット : 0x18

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM_PEF	Res.	Res.	Res.	BYP_ADDR_PAR	Res.	PVD_LOCK	SRAM_PARITY_LOCK	LOCKUP_LOCK
							rc_w1				rw		rw	rw	rw



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

ビット 31:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8 **SRAM\_PEF** : SRAM パリティエラーフラグ

このビットは、SRAM パリティエラーが検出されたときに、ハードウェアによってセットされます。ソフトウェアで「1」を書き込むことによってクリアされます。

0 : SRAM パリティエラーは検出されていません。

1 : SRAM パリティエラーが検出されました。

ビット 7:5 予約済みであり、リセット値のままにしておかなければなりません。

ビット 4 **BYP\_ADDR\_PAR** : パリティ計算によるバイパスアドレスビット 29

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。ユーザがアドレス 0x2XXXXXXX (アドレス範囲 0x20000000-0x20002000 のアドレス) にある RAM にコードを書き込み、ブート時に RAM からコードを実行する際に (RAM はアドレス 0x00 に再配置される)、不要なパリティエラーの発生を防ぐために使用できます。この場合、読み出し動作はパリティエラーが発生する範囲 0x00000000-0x00002000 から実行されます (アドレス上のパリティは異なる)。

0 : パリティ計算時のアドレスのビット 29 を考慮して、RAM ロード操作を実行します。

1 : パリティ計算時のアドレスのビット 29 を考慮せずに、RAM ロード操作を実行します。

ビット 3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2 **PVD\_LOCK** : PVD ロックイネーブルビット

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。これは、TIM1/8/15/16/17 のブレーク入力への PVD 接続と、PWR\_CR レジスタの PVDE と PLS[2:0] を有効にし、ロックするために使用できます。

0 : PVD 割り込みは、TIM1/8/15/16/17 のブレーク入力に接続されていません。PVDE および PLS[2:0] ビットは、アプリケーションでプログラムできます。

1 : PVD 割り込みは、TIM1/8/15/16/17 のブレーク入力に接続されています。PVDE および PLS[2:0] ビットは読み出し専用。

ビット 1 **SRAM\_PARITY\_LOCK** : SRAM パリティロックビット

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。これは、SRAM パリティエラー信号の TIM1/8/15/16/17 のブレーク入力への接続を有効にし、ロックするために使用できます。

0 : SRAM パリティエラー信号は TIM1/8/15/16/17 のブレーク入力から切断されています。

1 : SRAM パリティエラー信号は TIM1/8/15/16/17 のブレーク入力に接続されています。

ビット 0 **LOCKUP\_LOCK** : Cortex<sup>®</sup>-M4 LOCKUP (ハードフォルト) 出力イネーブルビット

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。これは、Cortex<sup>®</sup>-M4 LOCKUP (ハードフォルト) 出力の TIM1/15/16/17 ブレーク入力への接続を有効にし、ロックするために使用できます。

0 : Cortex<sup>®</sup>-M4 LOCKUP 出力は TIM1/8/15/16/17 のブレーク入力から切断されています。

1 : Cortex<sup>®</sup>-M4 LOCKUP 出力は TIM1/8/15/16/17 のブレーク入力に接続されています。

## 12.1.8 SYSCFG 設定レジスタ 3 (SYSCFG\_CFGR3)

注: このレジスタは、STM32F303x6/x8 および STM32F328 デバイスでのみ使用できます。

アドレスオフセット: 0x50

システムリセット値: 0x0000 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	ADC2_DMA_RMP	I2C1_TX_DMA_RMP	I2C1_RX_DMA_RMP	SPI1_TX_DMA_RMP	SPI1_RX_DMA_RMP					
						rw	rw	rw	rw	rw					

ビット 31:10 予約済みであり、リセット値のままにしておかなければなりません。

ビット 9 **ADC2\_DMA\_RMP[1]**: ADC2 の DMA コントローラ再配置ビット

0: DMA2 に配置された ADC2

1: DMA1 に配置された ADC2

ビット 8 **ADC2\_DMA\_RMP[0]**: ADC2 の DMA チャネル再配置ビット

0: DMA1 チャネル 2 に配置された ADC2

1: DMA1 チャネル 4 に配置された ADC2

ビット 7:6 **I2C1\_TX\_DMA\_RMP**: I2C1\_TX の DMA 再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。I2C1\_TX が配置された DMA1 チャネルを定義します。

00: DMA1 CH6 に配置された I2C1\_TX

01: DMA1 CH2 に配置された I2C1\_TX

10: DMA1 CH4 に配置された I2C1\_TX

11: DMA1 CH6 に配置された I2C1\_TX

ビット 5:4 **I2C1\_RX\_DMA\_RMP** : I2C1\_RX の DMA 再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。I2C1\_RX が配置された DMA1 チャンネルを定義します。

- 00 : DMA1 CH7 に配置された I2C1\_RX
- 01 : DMA1 CH3 に配置された I2C1\_RX
- 10 : DMA1 CH5 に配置された I2C1\_RX
- 11 : DMA1 CH7 に配置された I2C1\_RX

ビット 3:2 **SPI1\_TX\_DMA\_RMP** : SPI1\_TX の DMA 再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。SPI1\_TX が配置された DMA1 チャンネルを定義します。

- 00 : DMA1 CH3 に配置された SPI1\_TX
- 01 : DMA1 CH5 に配置された SPI1\_TX
- 10 : DMA1 CH7 に配置された SPI1\_TX
- 11 : DMA1 CH3 に配置された SPI1\_TX

ビット 1:0 **SPI1\_RX\_DMA\_RMP** : SPI1\_RX の DMA 再配置ビット

このビットは、ソフトウェアによってセット/クリアされます。SPI1\_RX が配置された DMA1 チャンネルを定義します。

- 00 : DMA1 CH2 に配置された SPI1\_RX
- 01 : DMA1 CH4 に配置された SPI1\_RX
- 10 : DMA1 CH6 に配置された SPI1\_RX
- 11 : DMA1 CH2 に配置された SPI1\_RX

## 12.1.9 SYSCFG 設定レジスタ 4 (SYSCFG\_CFGR4)

**注 :** このレジスタは、STM32F303xD/E および STM32F398xE デバイスでのみ使用できます。

**SYSCFG\_CFGR4** は、ADC のトリガ (主に新しい TIM20 イベント) を再配置するために追加されました。

アドレスオフセット : 0x48

システムリセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ADC34_JEXT14_RMP	ADC34_JEXT11_RMP	ADC34_JEXT5_RMP	ADC34_EXT15_RMP	ADC34_EXT6_RMP	ADC34_EXT5_RMP	ADC12_JEXT13_RMP	ADC12_JEXT6_RMP	ADC12_JEXT3_RMP	ADC12_EXT15_RMP	ADC12_EXT13_RMP	ADC12_EXT5_RMP	ADC12_EXT3_RMP	ADC12_EXT2_RMP
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



ビット 31:14 予約済みであり、リセット値のままにしておかなければなりません。

ビット 13 **ADC34\_JEXT14\_RMP** - ADC34 のインジェクトチャンネル JEXT14 の入力トリガを制御します。

0: トリガソースは TIM7\_TRGO です。

1: トリガソースは TIM20\_CC2 です。

ビット 12 **ADC34\_JEXT11\_RMP** - ADC34 のインジェクトチャンネル JEXT11 の入力トリガを制御します。

0: トリガソースは TIM1\_CC3 です。

1: トリガソースは TIM20\_TRGO2 です。

ビット 11 **ADC34\_JEXT5\_RMP** - ADC34 のインジェクトチャンネル JEXT5 の入力トリガを制御します。

0: トリガソースは TIM4\_CC3 です。

1: トリガソースは TIM20\_TRGO です。

ビット 10 **ADC34\_EXT15\_RMP** - ADC34 のレギュラチャンネル EXT15 の入力トリガを制御します。

0: トリガソースは TIM2\_CC1 です。

1: トリガソースは TIM20\_CC1 です。

ビット 9 **ADC34\_EXT6\_RMP** - ADC34 のレギュラチャンネル EXT6 の入力トリガを制御します。

0: トリガソースは TIM4\_CC1 です。

1: トリガソースは TIM20\_TRGO2 です。

ビット 8 **ADC34\_EXT5\_RMP** - ADC34 のレギュラチャンネル EXT5 の入力トリガを制御します。

0: 「0」でリセットした場合、トリガソースは EXTI ライン 2 です。

1: トリガソースは TIM20\_TRGO です。

ビット 7 **ADC12\_JEXT13\_RMP** - ADC12 のインジェクトチャンネルの入力トリガを制御します。

JEXT13:

0: トリガソースは TIM3\_CC1 です。

1: トリガソースは TIM20\_CC4 です。

ビット 6 **ADC12\_JEXT6\_RMP** - ADC12 のインジェクトチャンネル JEXT6 の入力トリガを制御します。

0: トリガソースは EXTI ライン 15 です。

1: トリガソースは TIM20\_TRGO2 です。

ビット 5 **ADC12\_JEXT3\_RMP** - ADC12 のインジェクトチャンネル EXT3 の入力トリガを制御します。

0: トリガソースは TIM2\_CC1 です。

1: トリガソースは TIM20\_TRGO です。

ビット 4 **ADC12\_EXT15\_RMP** - ADC12 のレギュラチャンネル EXT15 の入力トリガを制御します。

0: トリガソースは TIM3\_CC4 です。

1: トリガソースは TIM20\_CC3 です。

ビット 3 **ADC12\_EXT13\_RMP** - ADC12 のレギュラチャンネル EXT13 の入力トリガを制御します。

0: トリガソースは TIM6\_TRGO です。

1: トリガソースは TIM20\_CC2 です。

- ビット 2 **ADC12\_EXT5\_RMP** - ADC12 のレギュラチャネル EXT5 の入力トリガを制御します。
- 0 : トリガソースは TIM4\_CC4 です。
  - 1 : トリガソースは TIM20\_CC1 です。
- ビット 1 **ADC12\_EXT3\_RMP** - ADC12 のレギュラチャネル EXT3 の入力トリガを制御します。
- 0 : トリガソースは TIM2\_CC2 です。
  - 1 : トリガソースは TIM20\_TRGO2 です。
- ビット 0 **ADC12\_EXT2\_RMP** - ADC12 のレギュラチャネル EXT2 の入力トリガを制御します。
- 0 : トリガソースは TIM1\_CC3 です。
  - 1 : トリガソースは TIM20\_TRGO です。

## 12.1.10 SYSCFG レジスタマップ

次の表に、SYSCFG レジスタマップとリセット値を示します。

表 73. SYSCFG レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	SYSCFG_CFGR1	FPU_IE[5..0]							Res.	Res.	ENCODER_MODE [1:0]				I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP	DAC2_CH1_DMA_RMP	TIM7_DAC2_DMA_RMP	TIM6_DAC1_DMA_RMP	TIM17_DMA_RMP	TIM16_DMA_RMP	Res.	Res.	ADC24_DMA_RMP	DAC_TRIG_RMP	TIM1_ITR3_RMP	USB_IT_RMP	Res.	Res.	Res.	MEM_MODE
	リセット値	1	1	1	1	1	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0					X	X
0x04	SYSCFG_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAGE[15:0]_WP																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	SYSCFG_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[3:0]			EXTI2[3:0]			EXTI1[3:0]			EXTI0[3:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	SYSCFG_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7[3:0]			EXTI6[3:0]			EXTI5[3:0]			EXTI4[3:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	SYSCFG_EXTICR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11[3:0]			EXTI10[3:0]			EXTI9[3:0]			EXTI8[3:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	SYSCFG_EXTICR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15[3:0]			EXTI14[3:0]			EXTI13[3:0]			EXTI12[3:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	SYSCFG_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																								0				0			0		0	0
..	..																																		
0x50	SYSCFG_CFGR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																								1	0	0	0	0	0	0	0	0	0	0



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

表 73. SYSCFG レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	SYSCFG_CFGR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC34_JEXT14_RMP	ADC34_JEXT11_RMP	ADC34_JEXT5_RMP	ADC34_EXT15_RMP	ADC34_EXT6_RMP	ADC34_EXT5_RMP	ADC12_JEXT13_RMP	ADC12_JEXT6_RMP	ADC12_JEXT3_RMP	ADC12_EXT15_RMP	ADC12_EXT13_RMP	ADC12_EXT5_RMP	ADC12_EXT3_RMP	ADC12_EXT2_RMP
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 13 ダイレクトメモリアクセスコントローラ (DMA)

### 13.1 概要

DMA (Direct Memory Access : ダイレクトメモリアクセスコントローラ) は、ペリフェラルとメモリ間、およびメモリ間で高速データ転送を行うために使用します。DMA により、CPU を動作させずに、データを高速で移動することができます。これにより、CPU リソースを他の操作のためにあけておくことができます。

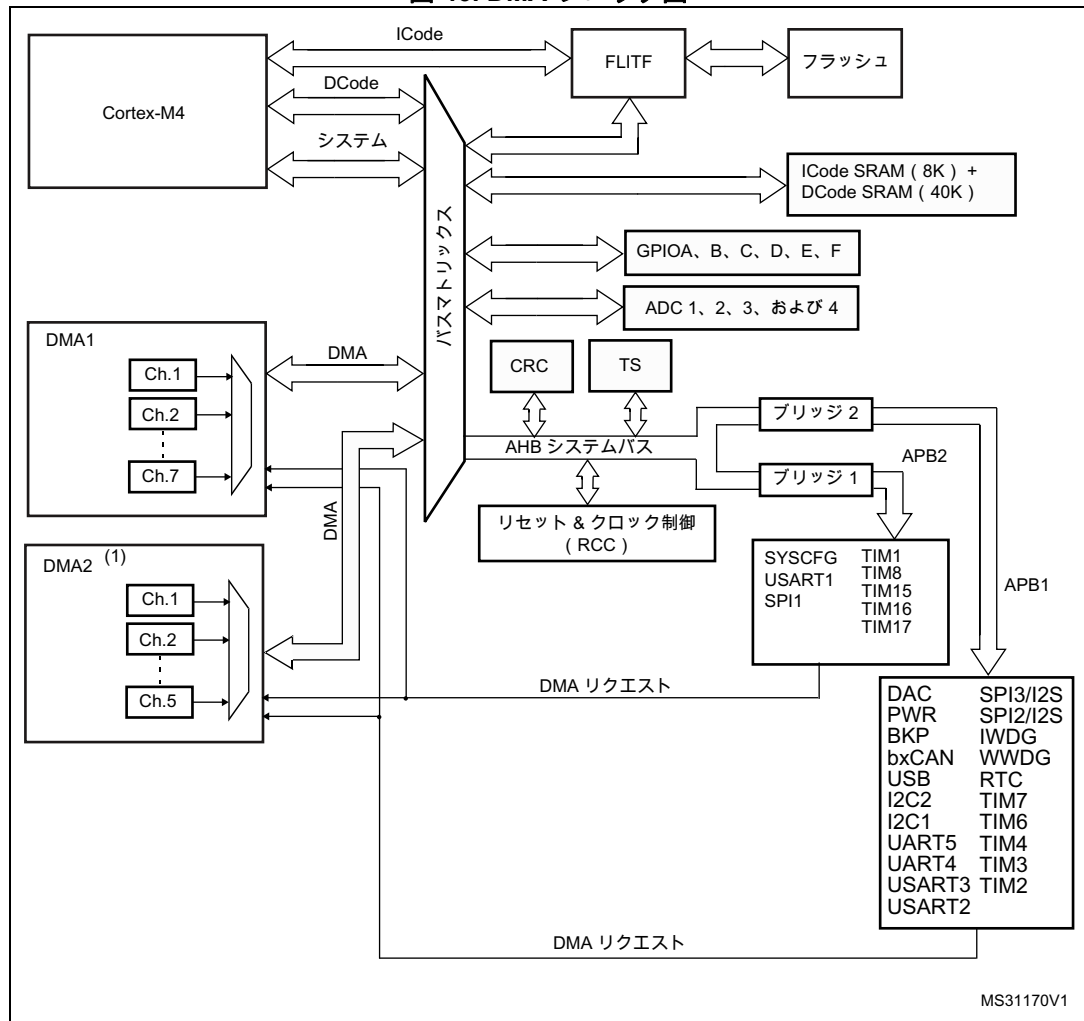
STM32F303xB/C/D/E、STM32F398xC、および STM32F398xE デバイスには合計 12 本のチャンネルを持つ 2 つの DMA コントローラがあります。STM32F303x6/8 および STM32F328x8 には、7 本のチャンネルを持つ 1 DMA コントローラがあります。それぞれのチャンネルは、1 つ以上のペリフェラルからのメモリアセスリクエストを管理する役割を担っています。また、DMA リクエスト間の優先順位を操作するためのアービタも内蔵しています。

### 13.2 DMA の主な機能

- 1STM32F302xB/C/D/E および STM32F302x6/8 デバイスの 12 本の個別に設定可能なチャンネル (リクエスト) および STM32F303x6/8 および STM32F328x8 デバイスの 7 本の個別に設定可能なチャンネル (リクエスト)
- 各チャンネルは、専用のハードウェア DMA リクエストに接続されており、ソフトウェアトリガもサポートしています。この設定はソフトウェアで行われます。
- 1 つの DMA チャンネルからのリクエスト間の優先順位はソフトウェアでプログラムできます (最高、高、中、低の 4 レベル)。またはレベルが等しい場合はハードウェアによって決まります (リクエスト 1 はリクエスト 2 よりも優先、など)。
- 転送元および転送先の転送サイズ (バイト、ハーフワード、ワード) は個別に設定され、パッキング/アンパッキングをエミュレートします。転送元/転送先のアドレスは、データサイズに基づいて整列させてください。
- サークュラバッファ管理をサポートします。
- 各チャンネルに対する単一割り込みリクエストにおいて、3 つのイベントフラグ (DMA 1/2 転送、DMA 転送完了、DMA 転送エラー) の論理和がとられます。
- メモリ間の転送
- ペリフェラルからメモリ、メモリからペリフェラル、およびペリフェラル間の転送
- フラッシュ、SRAM、APB、AHB ペリフェラルに対して、転送元および転送先としてアクセス
- プログラム可能な転送データ数 : 最大65535

以下にブロック図を示します。

図 46. DMA ブロック図



1. DMA2 は STM32F303x6/8 および STM32F328x8 デバイスでは使用できません。

## 13.3 DMA の実装

このマニュアルでは、DMA1 に実装されているすべての機能について説明しています。DMA2 でサポートされているチャンネル数は DMA1 よりも少ないですが、他の点ではまったく同等です。

表 74. DMA の実装

機能	DMA1	DMA2 <sup>(1)</sup>
DMA チャンネル数	7	5

1. DMA2 は STM32F303x6/8 および STM32F328x8 デバイスでは使用できません。

## 13.4 DMA の機能説明

DMA コントローラは、システムバスを Cortex-M4<sup>®</sup>F コアと共有することでダイレクトメモリ転送を行います。DMA リクエストは、CPU と DMA のターゲット転送先が同じである場合に（メモリまたはペリフェラル）、多少のバスサイクルの間、CPU のシステムバスへのアクセスを停止することがあります。バスマトリックスはラウンドロビンスケジューリングを実装します。これにより、CPU のシステムバス帯域幅（メモリとペリフェラルの両方への）の少なくとも半分を確保できます。

### 13.4.1 DMA トランザクション

イベントの後、ペリフェラルは DMA コントローラにリクエスト信号を送信します。DMA コントローラは、チャンネルの優先順位に応じて、リクエストを処理します。DMA コントローラがペリフェラルにアクセスするとすぐに、DMA コントローラはペリフェラルに確認応答を送信します。ペリフェラルは、DMA コントローラからの確認応答を受け取るとすぐに、そのリクエストを解除します。ペリフェラルによってリクエストが無効にされると、DMA コントローラは確認応答を解除します。さらにリクエストがある場合、ペリフェラルは次のトランザクションを開始できます。

要約すると、各 DMA 転送は次の 3 つの動作で構成されています。

- ペリフェラルデータレジスタまたは現在の内部ペリフェラル/メモリアドレスレジスタを介してアドレス指定されたメモリ位置からのデータのロード。最初の転送に使用される開始アドレスは、DMA\_CPARx または DMA\_CMARx レジスタでプログラムされたペリフェラル/メモリのベースアドレスです。
- ペリフェラルデータレジスタまたは現在の内部ペリフェラル/メモリアドレスレジスタを介してアドレス指定されたメモリ位置にロードされたデータの格納。最初の転送に使用される開始アドレスは、DMA\_CPARx または DMA\_CMARx レジスタでプログラムされたペリフェラル/メモリのベースアドレスです。
- まだ実行される予定のトランザクションの数を保持している DMA\_CNDTRx レジスタのポストデクリメント。

### 13.4.2 アービタ

アービタは、チャンネルリクエストをその優先順位に基づいて管理し、ペリフェラル/メモリアクセスシーケンスを起動します。

優先順位は 2 段階で管理されます。

- ソフトウェア：各チャンネルの優先順位は、DMA\_CCRx レジスタで設定できます。4 つのレベルがあります。
  - 最優先
  - 高優先
  - 中優先
  - 低優先
- ハードウェア：2 つのリクエストのソフトウェア優先順位レベルが同じである場合、小さな番号のチャンネルが大きな番号のチャンネルよりも優先されます。たとえば、チャンネル 2 はチャンネル 4 よりも優先されます。

## 13.4.3 DMA チャンネル

各チャンネルは、ある固定アドレスにあるペリフェラルレジスタとメモリアドレスの間の DMA 転送を処理します。転送データ数（最大 65535）はプログラム可能です。転送データ項目数を格納しているレジスタは、トランザクションの終了ごとにデクリメントされます。

### プログラム可能なデータサイズ

ペリフェラルおよびメモリの転送データサイズは、DMA\_CCRx レジスタの PSIZE および MSIZE ビットを介して完全にプログラム可能です。

### ポインタのインクリメント

ペリフェラルおよびメモリのポインタは、DMA\_CCRx レジスタの PINC および MINC ビットに応じて、各転送後に任意で自動的にポストインクリメントすることができます。インクリメントモードを有効にした場合、次の転送アドレスは、選択されたデータサイズに応じて 1、2、または 4 だけインクリメントされた前回の転送アドレスとなります。最初の転送アドレスは、DMA\_CPARx/DMA\_CMARx レジスタでプログラムされたアドレスとなります。転送動作中は、これらのレジスタは最初にプログラムされた値を保持します。現在の転送アドレス（現在の内部ペリフェラル/メモリアドレスレジスタ内）に対するソフトウェアによるアクセスはできません。

チャンネルがノンサーキュラモードに設定されている場合、転送終了後（転送すべきデータ項目数がゼロに達したとき）、DMA リクエストは処理されません。新たな転送すべきデータ項目数を DMA\_CNDTRx レジスタに再ロードするには、DMA チャンネルを無効にする必要があります。

**注：** *DMA チャンネルが無効にされると、DMA レジスタはリセットされません。DMA チャンネルレジスタ (DMA\_CCRx、DMA\_CPARx、DMA\_CMARx) は、チャンネル設定フェーズでプログラムされた初期値のままです。*

サーキュラモードでは、転送終了後、DMA\_CNDTRx レジスタには最初にプログラムした値が自動的に再ロードされます。現在の内部アドレスレジスタには、DMA\_CPARx/DMA\_CMARx レジスタからのベースアドレス値が再ロードされます。

### チャンネル設定手順

DMA チャンネル x (x はチャンネル番号) の設定は、以下のシーケンスで行います。

1. DMA\_CPARx レジスタに、ペリフェラルレジスタアドレスをセットします。ペリフェラルイベント後は、メモリとの転送に使用されるこのアドレスとデータをやり取りします。
2. DMA\_CMARx レジスタにメモリアドレスをセットします。ペリフェラルイベント後は、このメモリに対してデータを読み書きします。
3. 転送すべきデータの総数を DMA\_CNDTRx レジスタに設定します。毎回のペリフェラルイベント後に、この値はデクリメントされます。
4. DMA\_CCRx レジスタの PL[1:0] ビットを使用して、チャンネルの優先順位を設定します。
5. データ転送方向、サーキュラモード、ペリフェラルとメモリのインクリメントモード、ペリフェラルとメモリのデータサイズ、1/2 転送やフル転送後の割り込みを DMA\_CCRx レジスタに設定します。
6. DMA\_CCRx レジスタの ENABLE ビットをセットすることでチャンネルを有効にします。

チャンネルが有効になると、そのチャンネルに接続されているペリフェラルからの DMA リクエストを処理することができます。

バイトの半分が転送されると、1/2 転送フラグ (HTIF) がセットされ、1/2 転送割り込みイネーブルビット (HTIE) がセットされている場合には、割り込みが生成されます。転送が終了すると、転送完了フラグ (TCIF) がセットされ、転送完了割り込みイネーブルビット (TCIE) がセットされている場合には、割り込みが生成されます。



## サーキュラモード

サーキュラモードを使用すると、サーキュラバッファや連続したデータフロー（ADC スキャンモードなど）を処理することができます。この機能は、DMA\_CCRx レジスタの CIRC ビットを使用して有効にできます。サーキュラモードが有効になると、転送されるデータ項目数はチャンネル設定フェーズでプログラムされた初期値が自動的に再ロードされ、DMA リクエストの処理が続行されます。

## メモリ間モード

DMA チャンネルは、ペリフェラルからのリクエストによってトリガされなくても機能します。このモードは、メモリ間モードと呼ばれます。

DMA\_CCRx レジスタの MEM2MEM ビットがセットされると、チャンネルは、ソフトウェアが DMA\_CCRx レジスタの Enable ビット (EN) をセットすることによってそのチャンネルを有効にする。すぐに、転送を開始します。DMA\_CNDTRx レジスタがゼロに達すると、転送は停止します。メモリ間モードとサーキュラモードを同時に使用することはできません。

### 13.4.4 プログラム可能なデータ幅、データの整列、およびエンディアン

PSIZE と MSIZE が等しくない場合、DMA は、表 75 : プログラム可能なデータ幅およびエンディアンの動作 (ビット PINC = MINC = 1 の場合) に記載されているとおりに必要なデータの整列を行います。

表 75. プログラム可能なデータ幅およびエンディアンの動作 (ビット PINC = MINC = 1 の場合)

転送元ポートの幅	転送先ポートの幅	転送するデータ項目の数 (NDT)	転送元の内容: アドレス/データ	転送動作	転送先の内容: アドレス/データ
8	8	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: B0[7:0] を 0x0 で読み出し、次に B0[7:0] を 0x0 に書き込みます。 2: B1[7:0] を 0x1 で読み出し、次に B1[7:0] を 0x1 に書き込みます。 3: B2[7:0] を 0x2 で読み出し、次に B2[7:0] を 0x2 に書き込みます。 4: B3[7:0] を 0x3 で読み出し、次に B3[7:0] を 0x3 に書き込みます。	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3
8	16	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: B0[7:0] を 0x0 で読み出し、次に 00B0[15:0] を 0x0 に書き込みます。 2: B1[7:0] を 0x1 で読み出し、次に 00B1[15:0] を 0x2 に書き込みます。 3: B3[7:0] を 0x2 で読み出し、次に 00B2[15:0] を 0x4 に書き込みます。 4: B4[7:0] を 0x3 で読み出し、次に 00B3[15:0] を 0x6 に書き込みます。	@0x0 / 00B0 @0x2 / 00B1 @0x4 / 00B2 @0x6 / 00B3
8	32	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1: B0[7:0] を 0x0 で読み出し、次に 000000B0[31:0] を 0x0 に書き込みます。 2: B1[7:0] を 0x1 で読み出し、次に 000000B1[31:0] を 0x4 に書き込みます。 3: B3[7:0] を 0x2 で読み出し、次に 000000B2[31:0] を 0x8 に書き込みます。 4: B4[7:0] を 0x3 で読み出し、次に 000000B3[31:0] を 0xC に書き込みます。	@0x0 / 000000B0 @0x4 / 000000B1 @0x8 / 000000B2 @0xC / 000000B3
16	8	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: B1B0[15:0] を 0x0 で読み出し、次に B0[7:0] を 0x0 に書き込みます。 2: B3B2[15:0] を 0x2 で読み出し、次に B2[7:0] を 0x1 に書き込みます。 3: B5B4[15:0] を 0x4 で読み出し、次に B4[7:0] を 0x2 に書き込みます。 4: B7B6[15:0] を 0x6 で読み出し、次に B6[7:0] を 0x3 に書き込みます。	@0x0 / B0 @0x1 / B2 @0x2 / B4 @0x3 / B6
16	16	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: B1B0[15:0] を 0x0 で読み出し、次に B1B0[15:0] を 0x0 に書き込みます。 2: B3B2[15:0] を 0x2 で読み出し、次に B3B2[15:0] を 0x2 に書き込みます。 3: B5B4[15:0] を 0x4 で読み出し、次に B5B4[15:0] を 0x4 に書き込みます。 4: B7B6[15:0] を 0x6 で読み出し、次に B7B6[15:0] を 0x6 に書き込みます。	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6
16	32	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1: B1B0[15:0] を 0x0 で読み出し、次に 0000B1B0[31:0] を 0x0 に書き込みます。 2: B3B2[15:0] を 0x2 で読み出し、次に 0000B3B2[31:0] を 0x4 に書き込みます。 3: B5B4[15:0] を 0x4 で読み出し、次に 0000B5B4[31:0] を 0x8 に書き込みます。 4: B7B6[15:0] を 0x6 で読み出し、次に 0000B7B6[31:0] を 0xC に書き込みます。	@0x0 / 0000B1B0 @0x4 / 0000B3B2 @0x8 / 0000B5B4 @0xC / 0000B7B6
32	8	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: B3B2B1B0[31:0] を 0x0 で読み出し、次に B0[7:0] を 0x0 に書き込みます。 2: B7B6B5B4[31:0] を 0x4 で読み出し、次に B4[7:0] を 0x1 に書き込みます。 3: BBBAB9B8[31:0] を 0x8 で読み出し、次に B8[7:0] を 0x2 に書き込みます。 4: BFBEBDBC[31:0] を 0xC で読み出し、次に BC[7:0] を 0x3 に書き込みます。	@0x0 / B0 @0x1 / B4 @0x2 / B8 @0x3 / BC
32	16	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: B3B2B1B0[31:0] を 0x0 で読み出し、次に B1B0[7:0] を 0x0 に書き込みます。 2: B7B6B5B4[31:0] を 0x4 で読み出し、次に B5B4[7:0] を 0x1 に書き込みます。 3: BBBAB9B8[31:0] を 0x8 で読み出し、次に B9B8[7:0] を 0x2 に書き込みます。 4: BFBEBDBC[31:0] を 0xC で読み出し、次に BDBC[7:0] を 0x3 に書き込みます。	@0x0 / B1B0 @0x2 / B5B4 @0x4 / B9B8 @0x6 / BDBC
32	32	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1: B3B2B1B0[31:0] を 0x0 で読み出し、次に B3B2B1B0[31:0] を 0x0 に書き込みます。 2: B7B6B5B4[31:0] を 0x4 で読み出し、次に B7B6B5B4[31:0] を 0x4 に書き込みます。 3: BBBAB9B8[31:0] を 0x8 で読み出し、次に BBBAB9B8[31:0] を 0x8 に書き込みます。 4: BFBEBDBC[31:0] を 0xC で読み出し、次に BFBEBDBC[31:0] を 0xC に書き込みます。	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC



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

## バイトまたはハーフワード書き込み動作をサポートしていない AHB ペリフェラルへの対応

DMA が AHB のバイトまたはハーフワード書き込み動作を起動すると、データは HWDATA[31:0] バスの未使用のレーンに複製されます。そのため、使用中の AHB スレーブペリフェラルがバイト/ハーフワード書き込み動作をサポートしておらず (HSIZE がペリフェラルで使用されていない)、かつエラーが 1 件も出ていない場合は、以下の2つの例に示されるように、DMA は 32 HWDATA ビットを書き込みます。

- ハーフワード "0xABCD" を書き込むには、DMA は HWDATA バスを "0xABCDABCD" (HSIZE = ハーフワード) にセットします。
- バイト "0xAB" を書き込むには、DMA は HWDATA バスを "0xABABABAB" (HSIZE = バイト) にセットします。

AHB/APB ブリッジが AHB 32 ビットスレーブペリフェラルである (HSIZE の値は考慮せず) と仮定した場合、以下のようなやり方で、AHB バイトまたはハーフワードのあらゆる動作を 32 ビット APB 動作に変換します。

- データ "0xB0" を 0x0 (または 0x1, 0x2, 0x3) に書き込む AHB バイト書き込み動作は、データ "0xB0B0B0B0" を 0x0 に書き込む APB ワード書き込み動作に変換されます。
- データ "0xB1B0" を 0x0 (または 0x2) に書き込む AHB ハーフワード書き込み動作は、データ "0xB1B0B1B0" を 0x0 に書き込む APB ワード書き込み動作に変換されます。

たとえば、APB バックアップレジスタ (32 ビットアドレス境界に整列された 16 ビットレジスタ) に書き込むには、ソフトウェアで転送元のメモリサイズ (MSIZE) を 「16 ビット」 に、転送先のペリフェラルサイズ (PSIZE) を 「32 ビット」 に設定する必要があります。

### 13.4.5 エラー管理

DMA 転送エラーは、予約済みアドレス空間に対する読み出しや書き込みによって発生する可能性があります。DMA 読み出しまたは書き込みアクセス中に DMA 転送エラーが発生した場合、障害のあるチャンネルは、ハードウェアが対応するチャンネル設定レジスタ (DMA\_CCRx) の該当する EN ビットをクリアすることにより、自動的に無効化されます。DMA\_IFR レジスタのチャンネルの転送エラー割り込みフラグ (TEIF) がセットされ、DMA\_CCRx レジスタの転送エラー割り込みイネーブルビット (TEIE) がセットされている場合には、割り込みが生成されます。

### 13.4.6 DMA 割り込み

割り込みは、DMA チャンネルごとの 1/2 転送、転送完了、転送エラーに対して生成されます。高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 76. DMA 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
1/2 転送	HTIF	HTIE
転送完了	TCIF	TCIE
転送エラー	TEIF	TEIE

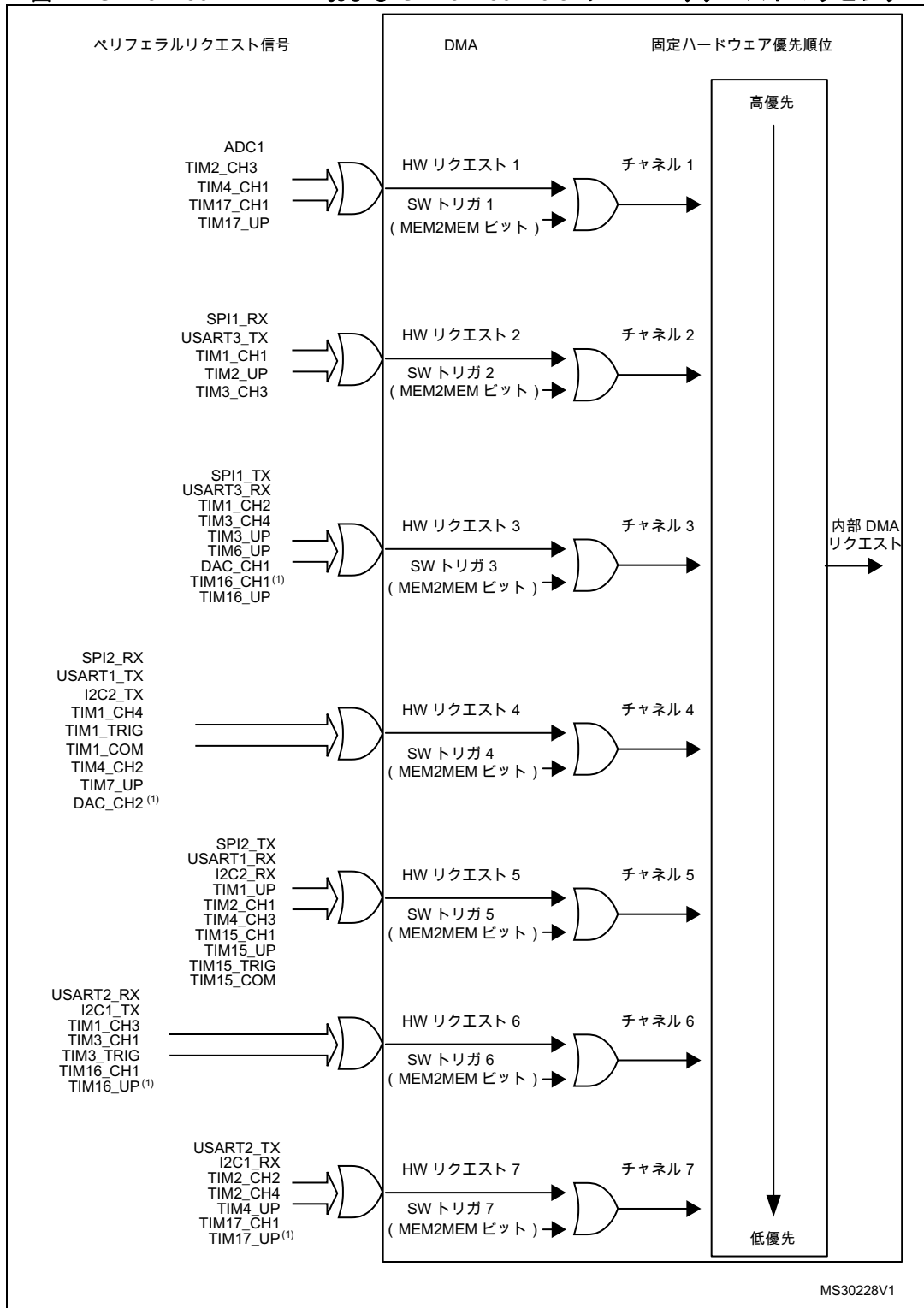
## 13.4.7 DMA リクエストマッピング

### DMA1 コントローラ

ペリフェラルからのハードウェアリクエスト (TIMx (x=1 ~ 4, 6, 7, 15 ~ 17)、ADC1、ADC2、SPI1、SPI2/I2S、I2Cx (x=1, 2)、DAC1\_Channel[1,2]、DAC2\_Channel[1]、および USARTx (x=1 ~ 3)) は、DMA1 を入力する前に単純に論理和がとられます。これは、1 つのチャンネルでは、一度に 1 つのリクエストのみ有効にする必要があることを意味しています。図 47 : STM32F302xB/C/D/E および STM32F302x6/8 の DMA1 リクエストマッピング および 図 48 : STM32F303x6/8 および STM32F328x8 DMA1 リクエストマッピングを参照してください。

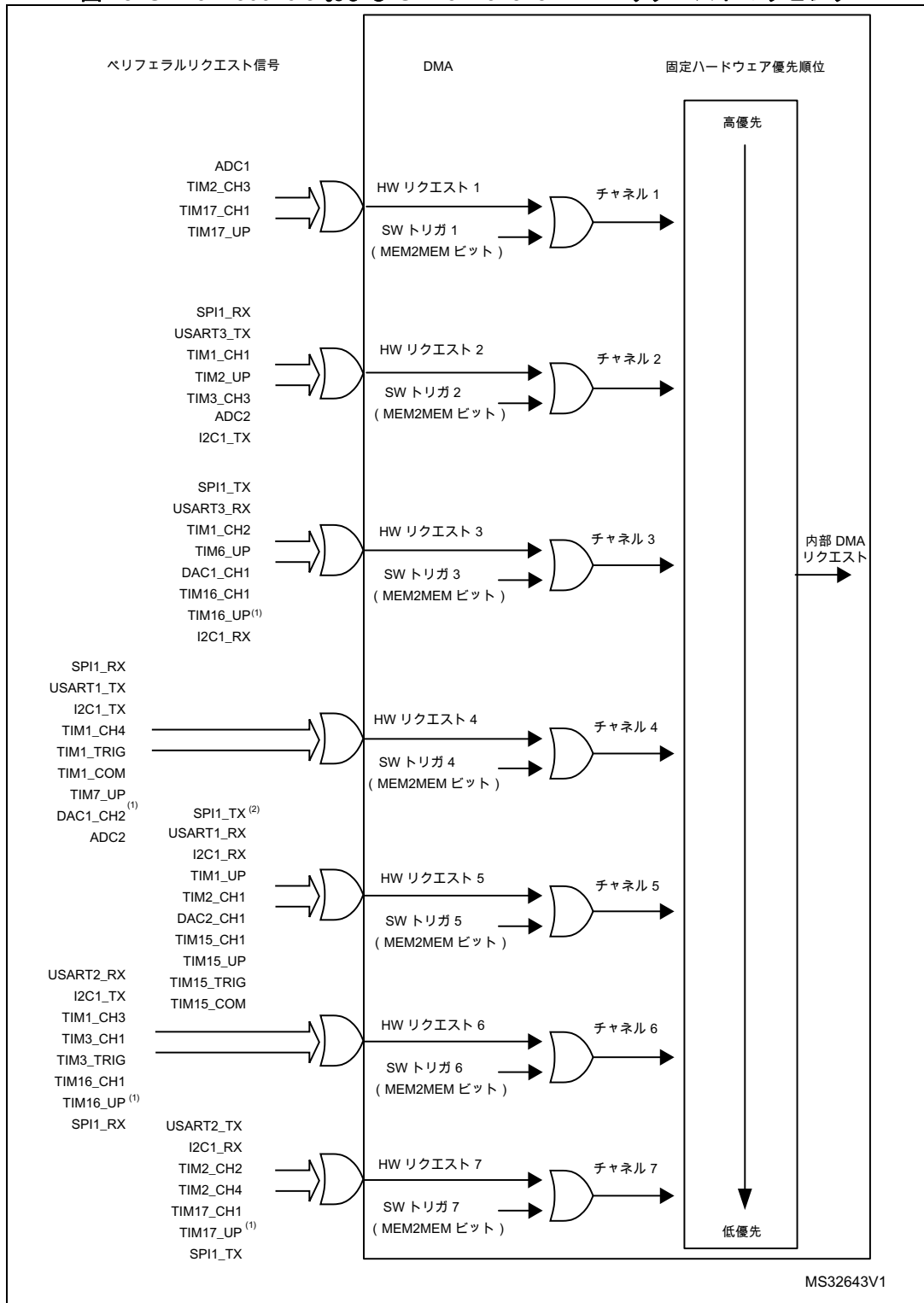
ペリフェラル DMA リクエストは、対応するペリフェラルのレジスタの DMA 制御ビットをプログラムすることにより、個別に有効/無効にできます。

図 47. STM32F302xB/C/D/E および STM32F302x6/8 の DMA1 リクエストマッピング



1. DMA リクエストは、SYSCFG\_CFGR1 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 12.1.1: SYSCFG 設定レジスタ 1 \(SYSCFG\\_CFGR1\) \(243 ページ\)](#) を参照してください。

図 48. STM32F303x6/8 および STM32F328x8 DMA1 リクエストマッピング



1. TIM6\_UP、DAC1\_CH1、TIM7\_UP、DAC1\_CH2、TIM16\_CH1、TIM16\_UP、TIM17\_CH1、TIM17\_UP、DAC2\_CH1、I2C1、SPI1 および DMA リクエストは、SYSCFG\_CFGR1 または SYSCFG\_CFGR3 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 12.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG\\_CFGR1\)](#) (243 ページ) および [セクション 12.1.8 : SYSCFG 設定レジスタ 3 \(SYSCFG\\_CFGR3\)](#) (255 ページ) を参照してください。



2. SYSCFG 設定レジスタ 2 (SYSCFG\_CFGR2) の SPI1\_TX\_DMA\_RMP[1:0] ビットは、チャンネル 5 および 7 で SPI1\_TX を再配置できます。

表 77. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE でのチャンネルごとの DMA1 リクエストの概要

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5	Channel6	Channel7
ADC	ADC1	-	-	-	-	-	-
SPI	-	SPI1_RX	SP1_TX	SPI2_RX	SPI2_TX	-	-
USART	-	USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I2C	I2C3_TX <sup>(1)</sup>	I2C3_RX <sup>(1)</sup>	-	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
TIM1	-	TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
TIM2	TIM2_CH3	TIM2_UP	-	-	TIM2_CH1	-	TIM2_CH2 TIM2_CH4
TIM3	-	TIM3_CH3	TIM3_CH4 TIM3_UP	-	-	TIM3_CH1 TIM3_TRIG	-
TIM4	TIM4_CH1	-	-	TIM4_CH2	TIM4_CH3	-	TIM4_UP
TIM6/DAC	-	-	TIM6_UP DAC_CH1 <sup>(2)</sup>	-	-	-	-
TIM7/DAC	-	-	-	TIM7_UP DAC_CH2 <sup>(2)</sup>	-	-	-
TIM15	-	-	-	-	TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM	-	-
TIM16	-	-	TIM16_CH1 TIM16_UP	-	-	TIM16_CH1 TIM16_UP <sup>(2)</sup>	-
TIM17	TIM17_CH1 TIM17_UP	-	-	-	-	-	TIM17_CH1 TIM17_UP <sup>(2)</sup>

1. STM32F303xD/E のみで使用できます。
2. DMA リクエストは、SYSCFG\_CFGR1 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、セクション 12.1.1 : SYSCFG 設定レジスタ 1 (SYSCFG\_CFGR1) (243 ページ) を参照してください。

表 78. STM32F303x6/8 および STM32F328x8 のチャンネルごとの DMA1 リクエストの概要

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5	チャンネル 6	チャンネル 7
ADC	ADC1	ADC2	-	ADC2 <sup>(1)</sup>	-	-	-
SPI	-	SPI1_RX	SP1_TX	SPI1_RX <sup>(1)</sup>	SPI1_TX <sup>(1)</sup>	SPI1_RX <sup>(1)</sup>	SPI1_TX <sup>(1)</sup>
USART	-	USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I2C	-	I2C1_TX <sup>(1)</sup>	I2C1_RX <sup>(1)</sup>	I2C1_TX <sup>(1)</sup>	I2C1_RX <sup>(1)</sup>	I2C1_TX	I2C1_RX
TIM1	-	TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
TIM2	TIM2_CH3	TIM2_UP	-	-	TIM2_CH1	-	TIM2_CH2 TIM2_CH4

表 78. STM32F303x6/8 および STM32F328x8 のチャンネルごとの DMA1 リクエストの概要

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5	チャンネル 6	チャンネル 7
TIM3	-	TIM3_CH3	TIM3_CH4 TIM3_UP	-	-	TIM3_CH1 TIM3_TRIG	-
TIM6/DAC	-	-	TIM6_UP DAC1_CH1 <sup>(1)</sup>	-	-	-	-
TIM7/DAC	-	-	-	TIM7_UP DAC2_CH2 <sup>(1)</sup>	-	-	-
DAC	-	-	-	-	DAC2_CH1 <sup>(1)</sup>	-	-
TIM15	-	-	-	-	TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM	-	-
TIM16	-	-	TIM16_CH1 TIM16_UP	-	-	TIM16_CH1 TIM16_UP <sup>(1)</sup>	-
TIM17	TIM17_CH1 TIM17_UP	-	-	-	-	-	TIM17_CH1 TIM17_UP <sup>(1)</sup>

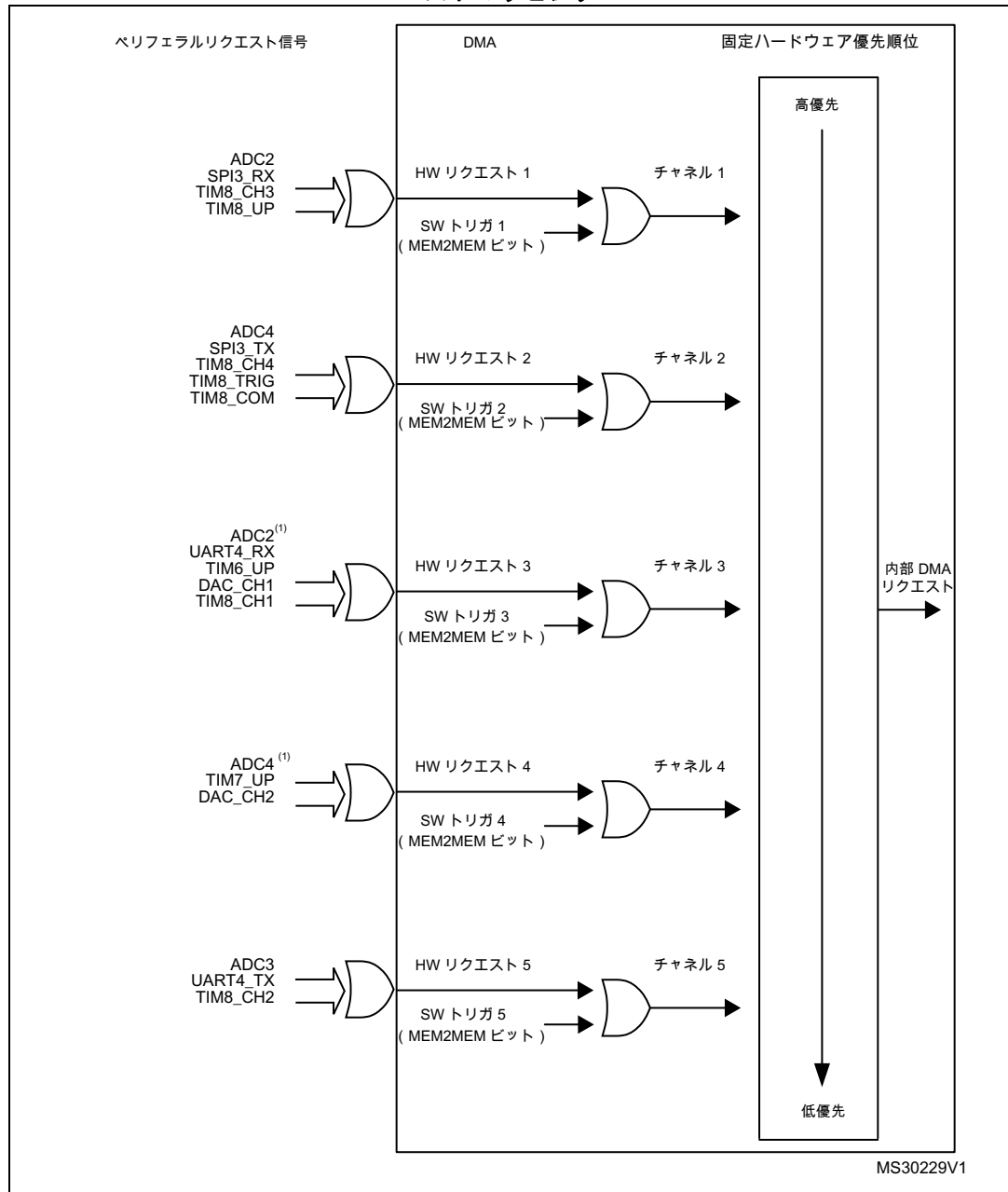
1. DMA リクエストは、SYSCFG\_CFGR1 または SYSCFG3 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 12.1.1: SYSCFG 設定レジスタ 1 \(SYSCFG\\_CFGR1\) \(243 ページ\)](#) および [セクション 12.1.8: SYSCFG 設定レジスタ 3 \(SYSCFG\\_CFGR3\) \(255 ページ\)](#) を参照してください。

## DMA2 コントローラ

ペリフェラル (TIMx (x= 6、7、8)、ADCx (x=2、3、4)、SPI/I2S3、UART4、DAC\_Channel[1,2]) からの 5 つのリクエストは、DMA2 に入力する前に単純に論理和がとられます。これは、一度に 1 つのリクエストのみを有効にする必要があることを意味します。[図 49: STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE での DMA2 リクエストマッピング](#) を参照してください。

ペリフェラル DMA リクエストは、対応するペリフェラルのレジスタの DMA 制御ビットをプログラムすることにより、個別に有効/無効にできます。

図 49. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE での DMA2 リクエストマッピング



1. DMA リクエストは、SYSCFG\_CFGR1 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 12.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG\\_CFGR1\) \(243 ページ\)](#) を参照してください。



表 79 に、チャンネルごとの DMA リクエストを示します。

表 79. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE のチャンネルごとの DMA2 リクエストの概要

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5
ADC	ADC2	ADC4	ADC2 <sup>(1)</sup>	ADC4 <sup>(1)</sup>	ADC3
SPI	SPI3_RX	SPI3_TX	-	SPI4_RX <sup>(2)</sup>	SPI4_TX <sup>(2)</sup>
UART4	-	-	UART4_RX	-	UART4_TX
TIM6/DAC	-	-	TIM6_UP DAC_CH1	-	-
TIM7/DAC	-	-	-	TIM7_UP DAC_CH2	-
TIM8	TIM8_CH3 TIM8_UP	TIM8_CH4 TIM8_TRIG TIM8_COM	TIM8_CH1	-	TIM8_CH2
TIM20 <sup>(2)</sup>	TIM20_CH1	TIM20_CH2	TIM20_CH3 TIM20_UP	TIM20_CH4 TIM20_TRIG TIM20_COM	-

1. DMA リクエストは、SYSCFG\_CFGR1 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 12.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG\\_CFGR1\) \(243 ページ\)](#) を参照してください。
2. STM32F303xD/E のみで使用できます。

## 13.5 DMA レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位でアクセスできます。

### 13.5.1 DMA 割り込みステータスレジスタ (DMA\_ISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27、23、19、**TEIFx** : チャネル x 転送エラーフラグ (x = 1 ~ 7)

15、11、7、3 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA\_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャネル x に転送エラーは発生していません。

1 : チャネル x に転送エラー (TE) が発生しました。

ビット 26、22、18、**HTIFx** : チャネル x 1/2 転送フラグ (x = 1 ~ 7)

14、10、6、2 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA\_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャネル x に 1/2 転送 (HT) イベントは発生していません。

1 : チャネル x に 1/2 転送 (HT) イベントが発生しました。

ビット 25、21、17、**TCIFx** : チャネル x 転送完了フラグ (x = 1 ~ 7)

13、9、5、1 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA\_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャネル x の転送完了 (TC) イベントは発生していません。

1 : チャネル x に転送完了 (TC) イベントが発生しました。

ビット 24、20、16、**GIFx** : チャネル x グローバル割り込みフラグ (x = 1 ~ 7)

12、8、4、0 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA\_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャネル x に TE、HT、または TC イベントは発生していません。

1 : チャネル x に TE、HT、または TC イベントが発生しました。

## 13.5.2 DMA 割り込みフラグクリアレジスタ (DMA\_IFCR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
				w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27、23、19、**CTEIFx** : チャネル x 転送エラークリア (x = 1 ~ 7)

15、11、7、3 このビットは、ソフトウェアによってセット/クリアされます。

0 : 影響なし。

1 : DMA\_ISR レジスタの対応する TEIF フラグをクリアします。

ビット 26、22、18、**CHTIFx** : チャネル x 1/2 転送クリア (x = 1 ~ 7)

14、10、6、2 このビットは、ソフトウェアによってセット/クリアされます。

0 : 影響なし。

1 : DMA\_ISR レジスタの対応する HTIF フラグをクリアします。

ビット 25、21、17、**CTCIFx** : チャネル x 転送完了クリア (x = 1 ~ 7)

13、9、5、1 このビットは、ソフトウェアによってセット/クリアされます。

0 : 影響なし。

1 : DMA\_ISR レジスタの対応する TCIF フラグをクリアします。

ビット 24、20、16、**CGIFx** : チャネル x グローバル割り込みクリア (x = 1 ~ 7)

12、8、4、0 このビットは、ソフトウェアによってセット/クリアされます。

0 : 影響なし。

1 : DMA\_ISR レジスタの GIF、TEIF、HTIF、および TCIF フラグをクリアします。

## 13.5.3 DMA チャンネル x 設定レジスタ (DMA\_CCRx) (x = 1 ~ 7、ここで x = チャンネル数)

アドレスオフセット : 0x08 + 0d20 x (チャンネル番号 - 1)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MEM2 MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **MEM2MEM** : メモリ間モード

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : メモリ間モードは無効です。
- 1 : メモリ間モードは有効です。

ビット 13:12 **PL[1:0]** : チャンネル優先順位レベル

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 低
- 01 : 中
- 10 : 高
- 11 : 最優先

ビット 11:10 **MSIZE[1:0]** : メモリサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 8 ビット
- 01 : 16 ビット
- 10 : 32 ビット
- 11 : 予約済み

ビット 9:8 **PSIZE[1:0]** : ペリフェラルサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 8 ビット
- 01 : 16 ビット
- 10 : 32 ビット
- 11 : 予約済み

ビット 7 **MINC** : メモリインクリメントモード

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : メモリインクリメントモードは無効です。
- 1 : メモリインクリメントモードは有効です。

ビット 6 **PINC** : ペリフェラルインクリメントモード

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : ペリフェラルインクリメントモードは無効です。
- 1 : ペリフェラルインクリメントモードは有効です。

- ビット 5 **CIRC** : サーキュラモード  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : サーキュラモードは無効です。  
1 : サーキュラモードは有効です。
- ビット 4 **DIR** : データ転送方向  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : ペリフェラルから読み出します。  
1 : メモリから読み出します。
- ビット 3 **TEIE** : 転送エラー割り込み有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TE 割り込みは無効です。  
1 : TE 割り込みは有効です。
- ビット 2 **HTIE** : 1/2 転送割り込みイネーブル  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : HT 割り込みは無効です。  
1 : HT 割り込みは有効です。
- ビット 1 **TCIE** : 転送完了割り込み有効化  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : TC 割り込みは無効です。  
1 : TC 割り込みは有効です。
- ビット 0 **EN** : チャネルイネーブル  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : チャネルは無効です。  
1 : チャネルは有効です。

## 13.5.4 DMA チャンネル x データ数レジスタ (DMA\_CNDTRx) (x = 1 ~ 7、x = チャンネル番号)

アドレスオフセット :  $0x0C + 0d20 \times$  (チャンネル番号 - 1)

リセット値 :  $0x0000\ 0000$

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **NDT[15:0]** : 転送データ項目の数

転送データ数 : 0~65535 このレジスタは、チャンネルが無効な場合にのみ書き込みできます。チャンネルが有効になると、このレジスタは読み出し専用になり、送信されるべき残りのバイトを指し示します。このレジスタは、各 DMA 転送後にデクリメントされます。

転送が完了すると、このレジスタは、ゼロのままにすることもできれば、チャンネルが自動再ロードモードに設定されている場合には、事前にプログラムされた値を自動的に再ロードすることもできます。

このレジスタの値が 0 の場合、チャンネルが有効であるか否かにかかわらず、トランザクションは処理できません。

## 13.5.5 DMA チャンネル x ペリフェラルアドレスレジスタ (DMA\_CPARx) (x = 1 ~ 7、x = チャンネル番号)

アドレスオフセット :  $0x10 + 0d20 \times$  (チャンネル番号 - 1)

リセット値 :  $0x0000\ 0000$

チャンネルが有効なときには、このレジスタへは書き込みを *行わない* てください。

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

ビット 31:0 **PA[31:0]** : ペリフェラルアドレス

データの読み出し/書き込みが行われるペリフェラルデータレジスタのベースアドレス。

PSIZE が 01 (16 ビット) のとき、PA[0] ビットは無視されます。アクセスは、自動的にハーフワードアドレスに整列されます。

PSIZE が 10 (32 ビット) のとき、PA[1:0] は無視されます。アクセスは、自動的にワードアドレスに整列されます。

## 13.5.6 DMA チャンネル x メモリアドレスレジスタ (DMA\_CMARx) (x = 1 ~ 7、x = チャンネル番号)

アドレスオフセット :  $0x14 + 0d20 \times$  (チャンネル番号 - 1)

リセット値 :  $0x0000\ 0000$

チャンネルが有効なときには、このレジスタへは書き込みを行わないでください。

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

ビット 31:0 **MA[31:0]** : メモリアドレス

データの読み出し／書き込みが行われるメモリ領域のベースアドレス。

MSIZE が 01 (16 ビット) のとき、MA[0] ビットは無視されます。アクセスは、自動的にハーフワードアドレスに整列されます。

MSIZE が 10 (32ビット) のとき、MA[1:0] は無視されます。アクセスは、自動的にワードアドレスに整列されます。

## 13.5.7 DMA レジスタマップ

次の表に、DMA レジスタマップとリセット値を示します。

表 80. DMA レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DMA_ISR	Res	Res	Res	Res	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	DMA_IFCR	Res	Res	Res	Res	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	DMA_CCR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DMA_CNDTR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																
0x10	DMA_CPAR1	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	DMA_CMAR1	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x1C	DMA_CCR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DMA_CNDTR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																
0x24	DMA_CPAR2	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	DMA_CMAR2	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x30	DMA_CCR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	MSIZE [1:0]	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	DMA_CNDTR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	リセット値																																
0x38	DMA_CPAR3	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	DMA_CMAR3	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



表 80. DMA レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
0x44	DMA_CCR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x48	DMA_CNDTR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x4C	DMA_CPAR4	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	DMA_CMAR4	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x58	DMA_CCR5	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5C	DMA_CNDTR5	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x60	DMA_CPAR5	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x64	DMA_CMAR5	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x68	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x6C	DMA_CCR6	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x70	DMA_CNDTR6	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x74	DMA_CPAR6	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x78	DMA_CMAR6	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x7C	予約済み	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x80	DMA_CCR7	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM2MEM	PL [1:0]	MSIZE [1:0]	PSIZE [1:0]	PSIZE [1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x84	DMA_CNDTR7	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:0]														
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	



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

表 80. DMA レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x88	DMA_CPAR7	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	DMA_CMAR7	MA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90 - 0xA7	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。

## 14 割り込みとイベント

### 14.1 ネスト化されたベクタ割り込みコントローラ (NVIC)

#### 14.1.1 NVIC の主な機能

- 74 のマスク可能な割り込みチャンネル (16 本の FPU 割り込みラインを持つ Cortex<sup>®</sup>-M4 は数に含まれていない)
- 16 のプログラム可能な優先レベル (4 ビットの割り込み優先順位を使用)
- 遅延時間の少ない例外および割り込み処理
- 電源管理制御
- システム制御レジスタの実装

NVIC とプロセッサコアのインタフェースは密に結合され、割り込み処理の遅延時間を小さくし、遅れて到着した割り込みを効率的に処理できます。

コア例外を含むすべての割り込みは、NVIC によって管理されます。例外と NVIC プログラミングの詳細については、Cortex<sup>®</sup>-M4 製品のプログラミングマニュアル PM0214 を参照してください。

#### 14.1.2 SysTick 較正值レジスタ

SysTick 較正值は 9000 にセットされますので、SysTick クロックを 9 MHz (最大  $f_{HCLK}/8$ ) に設定した状態での基準タイムベースは 1 ms になります。

#### 14.1.3 割り込みベクタと例外ベクタ

表 81 は、STM32F303xB/C および STM32F358xC デバイスのベクタテーブルです。表 82 は、STM32F303x6/8 および STM32F328x8 デバイスのベクタテーブルです。

表 81. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブル

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	-	-	-	予約済み	0x0000 0000
-	-3	固定	リセット	リセット	0x0000 0004
-	-2	固定	NMI	ノンマスク可能割り込み。RCC クロックセキュリティシステム (CSS) は NMI ベクタにリンクされます。	0x0000 0008
-	-1	固定	HardFault	あらゆる種類の異常	0x0000 000C
-	0	設定可能	MemManage	メモリ管理	0x0000 0010
-	1	設定可能	BusFault	プリフェッチ異常、メモリアクセス異常	0x0000 0014
-	2	設定可能	UsageFault	未定義命令または無効状態	0x0000 0018
-	-	-	-	予約済み	0x0000 001C~ 0x0000 0028
-	3	設定可能	SVCall	SWI 命令によるシステムサービスコール	0x0000 002C
-	5	設定可能	PendSV	ペンディング可能なシステムサービスリクエスト	0x0000 0038
-	6	設定可能	SysTick	システムティックタイマ	0x0000 003C



表 81. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
0	7	設定可能	WWDG	ウィンドウ型ウォッチドッグ割り込み	0x0000 0040
1	8	設定可能	PVD	PVD 検出割り込み (EXTI ライン 16 経由)	0x0000 0044
2	9	設定可能	TAMPER_STAMP	タンパおよびタイムスタンプ割り込み (EXTI ライン 19 経由)	0x0000 0048
3	10	設定可能	RTC_WKUP	RTC ウェイクアップタイマ割り込み (EXTI ライン 20 経由)	0x0000 004C
4	11	設定可能	FLASH	フラッシュグローバル割り込み	0x0000 0050
5	12	設定可能	RCC	RCC グローバル割り込み	0x0000 0054
6	13	設定可能	EXTI0	EXTI ライン 0 割り込み	0x0000 0058
7	14	設定可能	EXTI1	EXTI ライン 1 割り込み	0x0000 005C
8	15	設定可能	EXTI2_TS	EXTI ライン 2 およびタッチセンシング割り込み	0x0000 0060
9	16	設定可能	EXTI3	EXTI ライン 3	0x0000 0064
10	17	設定可能	EXTI4	EXTI ライン 4	0x0000 0068
11	18	設定可能	DMA1_Channel1	DMA1 チャンネル 1 割り込み	0x0000 006C
12	19	設定可能	DMA1_Channel2	DMA1 チャンネル 2 割り込み	0x0000 0070
13	20	設定可能	DMA1_Channel3	DMA1 チャンネル 3 割り込み	0x0000 0074
14	21	設定可能	DMA1_Channel4	DMA1 チャンネル 4 割り込み	0x0000 0078
15	22	設定可能	DMA1_Channel5	DMA1 チャンネル 5 割り込み	0x0000 007C
16	23	設定可能	DMA1_Channel6	DMA1 チャンネル 6 割り込み	0x0000 0080
17	24	設定可能	DMA1_Channel7	DMA1 チャンネル 7 割り込み	0x0000 0084
18	25	設定可能	ADC1_2	ADC1 および ADC2 グローバル割り込み	0x0000 0088
19 <sup>(1)</sup>	26	設定可能	USB_HP/CAN_TX	USB 高優先/CAN_TX 割り込み	0x0000 008C
20 <sup>(1)</sup>	27	設定可能	USB_LP/CAN_RX0	USB 低優先/CAN_RX0 割り込み	0x0000 0090
21	28	設定可能	CAN_RX1	CAN_RX1 割り込み	0x0000 0094
22	29	設定可能	CAN_SCE	CAN_SCE 割り込み	0x0000 0098
23	30	設定可能	EXTI9_5	EXTI ライン [9:5] 割り込み	0x0000 009C
24	31	設定可能	TIM1_BRK/TIM15	TIM1 ブレーク/TIM15 グローバル割り込み	0x0000 00A0
25	32	設定可能	TIM1_UP/TIM16	TIM1 更新/TIM16 グローバル割り込み	0x0000 00A4
26	33	設定可能	TIM1_TRG_COM/TIM17	TIM1 トリガおよびコムテーション/TIM17 割り込み	0x0000 00A8
27	34	設定可能	TIM1_CC	TIM1 キャプチャ/比較割り込み	0x0000 00AC
28	35	設定可能	TIM2	TIM2 グローバル割り込み	0x0000 00B0
29	36	設定可能	TIM3	TIM3 グローバル割り込み	0x0000 00B4
30	37	設定可能	TIM4	TIM4 グローバル割り込み	0x0000 00B8
31	38	設定可能	I2C1_EV	I2C1 イベント割り込みおよび EXTI ライン 23 割り込み	0x0000 00BC
32	39	設定可能	I2C1_ER	I2C1 エラー割り込み	0x0000 00C0

表 81. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
33	40	設定可能	I2C2_EV	I2C2 イベント割り込みおよび EXTI ライン 24 割り込み	0x0000 00C4
34	41	設定可能	I2C2_ER	I2C2 エラー割り込み	0x0000 00C8
35	42	設定可能	SPI1	SPI1 グローバル割り込み	0x0000 00CC
36	43	設定可能	SPI2	SPI2 グローバル割り込み	0x0000 00D0
37	44	設定可能	USART1	USART1 グローバル割り込みおよび EXTI ライン 25	0x0000 00D4
38	45	設定可能	USART2	USART2 グローバル割り込みおよび EXTI ライン 26	0x0000 00D8
39	46	設定可能	USART3	USART3 グローバル割り込みおよび EXTI ライン 28	0x0000 00DC
40	47	設定可能	EXTI15_10	EXTI ライン [15:10] 割り込み	0x0000 00E0
41	48	設定可能	RTC_ALARM	RTC アラーム割り込み	0x0000 00E4
42 <sup>(1)</sup>	49	設定可能	USBWakeUp	サスペンドからの USB ウェイクアップ (EXTI ライン 18)	0x0000 00E8
43	50	設定可能	TIM8_BRK	TIM8 ブレーク割り込み	0x0000 00EC
44	51	設定可能	TIM8_UP	TIM8 更新割り込み	0x0000 00F0
45	52	設定可能	TIM8_TRG_COM	TIM8 トリガおよびコムティーション割り込み	0x0000 00F4
46	53	設定可能	TIM8_CC	TIM8 キャプチャ/比較割り込み	0x0000 00F8
47	54	設定可能	ADC3	ADC3 グローバル割り込み	0x0000 00FC
48	55	設定可能	FMC <sup>(2)</sup>	FMC グローバル割り込み	0x0000 0100
49	56		予約済み		0x0000 0104
50	57		予約済み		0x0000 0108
51	58	設定可能	SPI3	SPI3 グローバル割り込み	0x0000 010C
52	59	設定可能	UART4	UART4 グローバルおよび EXTI ライン 34 割り込み	0x0000 0110
53	60	設定可能	UART5	UART5 グローバルおよび EXTI ライン 35 割り込み	0x0000 0114
54	61	設定可能	TIM6_DAC	TIM6 グローバルおよび DAC1 アンダーラン割り込み	0x0000 0118
55	62	設定可能	TIM7	TIM7 グローバル割り込み	0x0000 011C
56	63	設定可能	DMA2_Channel1	DMA2 チャンネル 1 グローバル割り込み	0x0000 0120
57	64	設定可能	DMA2_Channel2	DMA2 チャンネル 2 グローバル割り込み	0x0000 0124
58	65	設定可能	DMA2_Channel3	DMA2 チャンネル 3 グローバル割り込み	0x0000 0128
59	66	設定可能	DMA2_Channel4	DMA2 チャンネル 4 グローバル割り込み	0x0000 012C
60	67	設定可能	DMA2_Channel5	DMA2 チャンネル 5 グローバル割り込み	0x0000 0130
61	68	設定可能	ADC4	ADC4 グローバル割り込み	0x0000 0134
62	69	-	予約済み		0x0000 0138
63	70	-	予約済み		0x0000 013C



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

表 81. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
64	71	設定可能	COMP1_2_3	COMP1、COMP2 および COMP3 割り込み (EXTI ライン 21、22 および 29 割り込みの組み合わせ)	0x0000 0140
65	72	設定可能	COMP4_5_6	COMP4、COMP5 および COMP6 割り込み (EXTI ライン 30、31 および 32 割り込みの組み合わせ)	0x0000 0144
66	73	設定可能	COMP7	COMP7 割り込み (EXTI ライン 33 割り込みの組み合わせ)	0x0000 0148
67	74	-	予約済み		0x0000 014C
68	75	-	予約済み		0x0000 0150
69	76	-	予約済み		0x0000 0154
70	77	-	予約済み		0x0000 0158
71	78	-	予約済み		0x0000 015C
72	79	設定可能	I2C3_EV <sup>(2)</sup>	I2C3 イベント割り込み	0x0000 0160
73	80	設定可能	I2C3_ER <sup>(2)</sup>	I2C3 エラー割り込み	0x0000 0164
74	81	設定可能	USB_HP	USB 高優先割り込み	0x0000 0168
75	82	設定可能	USB_LP	USB 低優先割り込み	0x0000 016C
76	83	設定可能	USB_WakeUp_RMP (注 1 参照)	サスペンドからの USB ウェイクアップおよび EXTI ライン 18	0x0000 0170
77	84	設定可能	TIM20_BRK <sup>(2)</sup>	TIM20 ブレーク割り込み	0x0000 0174
78	85	設定可能	TIM20_UP <sup>(2)</sup>	TIM20 更新割り込み	0x0000 0178
79	86	設定可能	TIM20_TRG_COM <sup>(2)</sup>	TIM20 トリガおよびコミュニケーション割り込み	0x0000 017C
80	87	設定可能	TIM20_CC <sup>(2)</sup>	TIM20 キャプチャ/比較割り込み	0x0000 0180
81	88	設定可能	FPU	浮動小数点割り込み	0x0000 0184
82	89	-	-	予約済み	0x0000 0188
83	90	-	-	予約済み	0x0000 018C
84	91	設定可能	-	SPI4 SPI4 グローバル割り込み <sup>(2)</sup>	0x0000 0190

1. セクション 12.1.1 : SYSCFG 設定レジスタ 1 (SYSCFG\_CFGR1) (243 ページ) で USB\_IT\_RMP ビットを設定することで、割り込みライン 74、75、76 それぞれに USB 割り込み (USB\_HP、USB\_LP、USB\_WKUP) の再配置が可能です。
2. 使用できるのは STM32F303xD/E でのみです。

表 82. STM32F303x6/8 および STM32F328x8ベクタテーブル

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	-	-	-	予約済み	0x0000 0000
-	-3	固定	リセット	リセット	0x0000 0004
-	-2	固定	NMI	ノンマスカブル割り込み。RCC クロックセキュリティシステム (CSS) は NMI ベクタにリンクされます。	0x0000 0008
-	-1	固定	HardFault	あらゆる種類の異常	0x0000 000C
-	0	設定可能	MemManage	メモリ管理	0x0000 0010
-	1	設定可能	BusFault	ブリフェッチ異常、メモリアクセス異常	0x0000 0014
-	2	設定可能	UsageFault	未定義命令または無効状態	0x0000 0018
-	-	-	-	予約済み	0x0000 001C~ 0x0000 0028
-	3	設定可能	SVCall	SWI 命令によるシステムサービスコール	0x0000 002C
-	5	設定可能	PendSV	ペンディング可能なシステムサービスリクエスト	0x0000 0038
-	6	設定可能	SysTick	システムティックタイマ	0x0000 003C
0	7	設定可能	WWDG	ウィンドウ型ウォッチドッグ割り込み	0x0000 0040
1	8	設定可能	PVD	PVD 検出割り込み (EXTI ライン 16 経由)	0x0000 0044
2	9	設定可能	TAMPER_STAMP	タンパおよびタイムスタンプ割り込み EXTI ライン 19 経由	0x0000 0048
3	10	設定可能	RTC_WKUP	RTC ウェイクアップタイマ割り込み (EXTI ライン 20 経由)	0x0000 004C
4	11	設定可能	FLASH	フラッシュグローバル割り込み	0x0000 0050
5	12	設定可能	RCC	RCC グローバル割り込み	0x0000 0054
6	13	設定可能	EXTI0	EXTI ライン 0 割り込み	0x0000 0058
7	14	設定可能	EXTI1	EXTI ライン 1 割り込み	0x0000 005C
8	15	設定可能	EXTI2_TS	EXTI ライン 2 およびタッチセンシング割り込み	0x0000 0060
9	16	設定可能	EXTI3	EXTI ライン 3	0x0000 0064
10	17	設定可能	EXTI4	EXTI ライン 4	0x0000 0068
11	18	設定可能	DMA1_Channel1	DMA1 チャンネル 1 割り込み	0x0000 006C
12	19	設定可能	DMA1_Channel2	DMA1 チャンネル 2 割り込み	0x0000 0070
13	20	設定可能	DMA1_Channel3	DMA1 チャンネル 3 割り込み	0x0000 0074
14	21	設定可能	DMA1_Channel4	DMA1 チャンネル 4 割り込み	0x0000 0078
15	22	設定可能	DMA1_Channel5	DMA1 チャンネル 5 割り込み	0x0000 007C
16	23	設定可能	DMA1_Channel6	DMA1 チャンネル 6 割り込み	0x0000 0080
17	24	設定可能	DMA1_Channel7	DMA1 チャンネル 7 割り込み	0x0000 0084
18	25	設定可能	ADC1_2	ADC1 および ADC2 グローバル割り込み	0x0000 0088
19	26	設定可能	CAN_TX	CAN_TX 割り込み	0x0000 008C
20	27	設定可能	CAN_RX0	CAN_RX0 割り込み	0x0000 0090



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

表 82. STM32F303x6/8 および STM32F328x8ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
21	28	設定可能	CAN_RX1	CAN_RX1 割り込み	0x0000 0094
22	29	設定可能	CAN_SCE	CAN_SCE 割り込み	0x0000 0098
23	30	設定可能	EXTI9_5	EXTI ライン [9:5] 割り込み	0x0000 009C
24	31	設定可能	TIM1_BRK/TIM15	TIM1 ブレーク/TIM15 グローバル割り込み	0x0000 00A0
25	32	設定可能	TIM1_UP/TIM16	TIM1 更新/TIM16 グローバル割り込み	0x0000 00A4
26	33	設定可能	TIM1_TRG_COM /TIM17	TIM1 トリガおよびコムテーション/TIM17 割り込み	0x0000 00A8
27	34	設定可能	TIM1_CC	TIM1 キャプチャ /比較割り込み	0x0000 00AC
28	35	設定可能	TIM2	TIM2 グローバル割り込み	0x0000 00B0
29	36	設定可能	TIM3	TIM3 グローバル割り込み	0x0000 00B4
30	37	-	予約済み		0x0000 00B8
31	38	設定可能	I2C1_EV	I2C1 イベント割り込みおよび EXTI ライン 23 割り込み	0x0000 00BC
32	39	設定可能	I2C1_ER	I2C1 エラー割り込み	0x0000 00C0
33	40	-	予約済み		0x0000 00C4
34	41	-	予約済み		0x0000 00C8
35	42	-	SPI1	SPI1 グローバル割り込み	0x0000 00CC
36	43	-	予約済み		0x0000 00D0
37	44	設定可能	USART1	USART1 グローバル割り込みおよび EXTI ライン 25	0x0000 00D4
38	45	設定可能	USART2	USART2 グローバル割り込みおよび EXTI ライン 26	0x0000 00D8
39	46	設定可能	USART3	USART3 グローバル割り込みおよび EXTI ライン 28	0x0000 00DC
40	47	設定可能	EXTI15_10	EXTI ライン [15:10] 割り込み	0x0000 00E0
41	48	設定可能	RTC_ALARM	RTC アラーム割り込み	0x0000 00E4
42	49	-	予約済み		0x0000 00E8
43	50	-	予約済み		0x0000 00EC
44	51	-	予約済み		0x0000 00F0
45	52	-	予約済み		0x0000 00F4
46	53	-	予約済み		0x0000 00F8
47	54	-	予約済み		0x0000 00FC
48	55	-	予約済み		0x0000 0100
49	56	-	予約済み		0x0000 0104
50	57	-	予約済み		0x0000 0108
51	58	-	予約済み		0x0000 010C
52	59	-	予約済み		0x0000 0110
53	60	-	予約済み		0x0000 0114



表 82. STM32F303x6/8 および STM32F328x8ベクタテーブル (続き)

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
54	61	設定可能	TIM6_DAC1	TIM6 グローバルおよび DAC1 アンダーラン割り込み	0x0000 0118
55	62	設定可能	TIM7_DAC2	TIM7 グローバルおよび DAC2 アンダーラン割り込み	0x0000 011C
56	63	-	予約済み		0x0000 0120
57	64	-	予約済み		0x0000 0124
58	65	-	予約済み		0x0000 0128
59	66	-	予約済み		0x0000 012C
60	67	-	予約済み		0x0000 0130
61	68	-	予約済み		0x0000 0134
62	69	-	予約済み		0x0000 0138
63	70	-	予約済み		0x0000 013C
64	71	設定可能	COMP2	COMP2 割り込み (EXTI ライン 22 割り込みの組み合わせ)	0x0000 0140
65	72	設定可能	COMP4_6	COMP4 および COMP6 割り込み (EXTI ライン 30 および 32 割り込みの各組み合わせ)	0x0000 0144
66	73	-	予約済み		0x0000 0148
67	74	-	予約済み		0x0000 014C
68	75	-	予約済み		0x0000 0150
69	76	-	予約済み		0x0000 0154
70	77	-	予約済み		0x0000 0158
71	78	-	予約済み		0x0000 015C
72	79	-	予約済み		0x0000 0160
73	80	-	予約済み		0x0000 0164
74	81	-	予約済み		0x0000 0168
75	82	-	予約済み		0x0000 016C
76	83	-	予約済み		0x0000 0170
77	84	-	予約済み		0x0000 0174
78	85	-	予約済み		0x0000 0178
79	86	-	予約済み		0x0000 017C
80	87	-	予約済み		0x0000 0180
81	88	設定可能	FPU	浮動小数点割り込み	0x0000 0184



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

## 14.2 拡張割り込み／イベントコントローラ (EXTI)

拡張割り込み／イベントコントローラ (EXTI) は、外部および内部で非同期的に発生するイベント／割り込みを管理し、CPU／割り込みコントローラへのイベントリクエスト、およびパワーマネージャへのウェイクアップリクエストを生成します。

EXTI を使用することで、外部／内部イベントラインを最大 36 本 (28 本の外部イベントラインおよび 8 本の内部イベントライン) まで管理することが可能です。

各外部割り込みラインのアクティブエッジは個別に選択可能です。一方で、内部割り込みのアクティブエッジは常に立ち上がりエッジです。割り込みは保留のまま残すことができます。外部割り込みの場合、ステータスレジスタはインスタンス化され、割り込みのソースを示します。イベントは常に単純なパルスで、コアウェイクアップのトリガに使用されます。内部割り込みの場合、保留ステータスは生成したペリフェラルによって保証され、特別なフラグは必要ありません。各入力ラインは、割り込みまたはイベント生成時に個別にマスクすることができます。また、内部ラインは STOP モードのみサンプリングされます。このコントローラは、専用レジスタに書き込むことで、対応するハードウェアイベントラインによって多重化されているソフトウェアによる (唯一の) 外部イベントのエミュレーションを可能にします。

### 14.2.1 主な機能

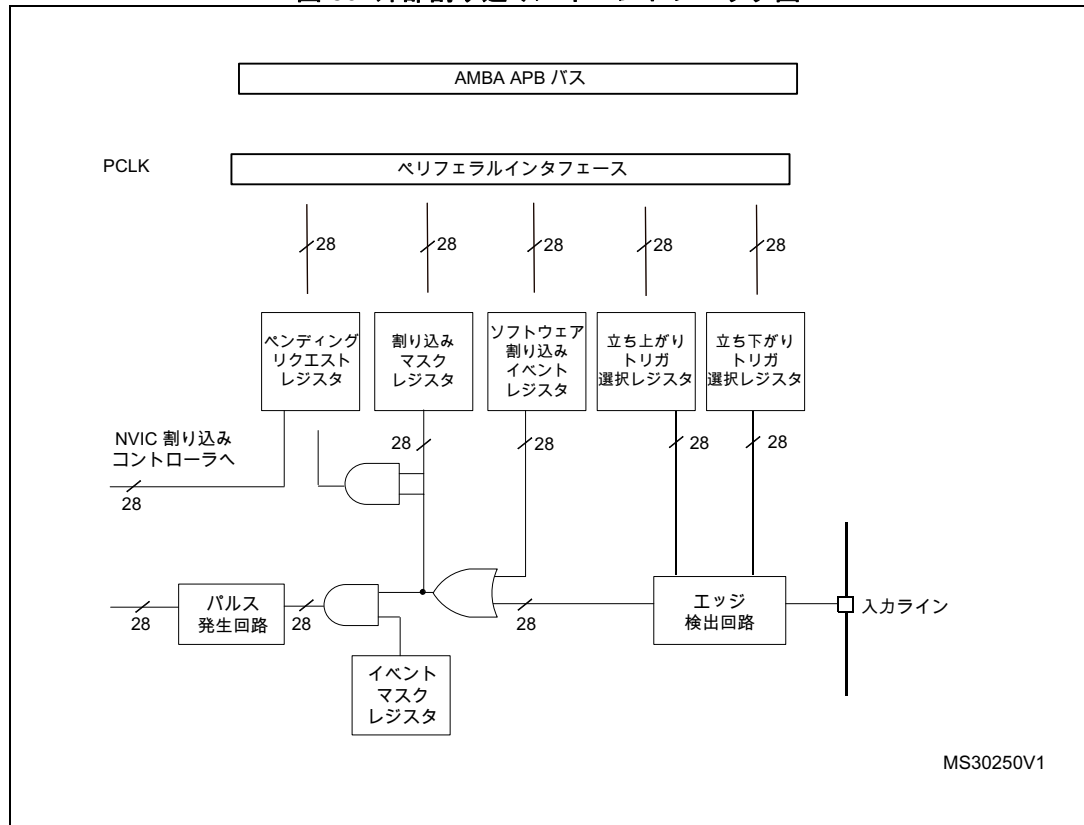
EXTI の主な機能は以下のとおりです。

- 最大 36 のイベント／割り込みリクエストの生成をサポート
- 外部イベントリクエストまたは内部イベントリクエストとしての各ラインの独立した設定
- 各イベント／割り込みラインの個別マスク
- システムが STOP モードでない場合に、内部ラインを自動的に無効化
- 外部イベント／割り込みラインの個別トリガ
- 外部割り込みライン専用のステータスビット
- すべての外部イベントリクエストのエミュレーション

## 14.2.2 ブロック図

以下に拡張割り込み／イベントブロック図を示します。

図 50. 外部割り込み／イベントブロック図



## 14.2.3 ウェイクアップイベント管理

STM32F3xx デバイスは、コアをウェイクアップするために、外部イベントや内部イベントを処理することができます (WFE)。ウェイクアップイベントは、次のいずれかによって生成できます。

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、Cortex<sup>®</sup>-M4 システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、EXTI ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャンネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードに設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャンネルペンディングビットをクリアする必要はありません。

## 14.2.4 非同期内部割り込み

一部の通信ペリフェラル (UART、I2C) は、システムが RUN モードにあるときにイベントを生成することができ、STOP モードにあるときもシステムを STOP モードからウェイクアップさせることが可能になります。

そのためには、ペリフェラルは、(APB クロックなどのシステムクロックに) 同期されたイベントと非同期のイベントの両方を生成することが求められます。

## 14.2.5 機能詳細

外部割り込みラインで割り込みを生成するには、割り込みラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、割り込みマスクレジスタの対応するビットに "1" を書き込むことによって割り込みリクエストを有効にします。選択されたエッジが外部割り込みラインで発生すると、割り込みリクエストが生成されます。割り込みラインに対応するペンディングビットもセットされます。このリクエストは、ペンディングレジスタに "1" を書き込むことによってリセットされます。

内部割り込みラインの場合、立ち上がりエッジが常にアクティブエッジです。割り込みは割り込みマスクレジスタのデフォルト値によって有効化され、ペンディングレジスタには対応するペンディングビットはセットされていません。

イベントを生成するには、イベントラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、イベントマスクレジスタの対応するビットに "1" を書き込むことによってイベントリクエストを有効にします。選択されたエッジがイベントラインで発生すると、イベントパルスが生成されます。イベントラインに対応するペンディングビットはセットされません。

外部ラインの場合、割り込み/イベントリクエストは、ソフトウェア割り込み/イベントレジスタに "1" を書き込むことによって、ソフトウェアによって生成することもできます。

**注：** *内部ラインに関連する割り込みやイベントは、システムが STOP モードのときのみトリガできます。システムがまだ動作している場合は、割り込み/イベントは生成されません。*

### ハードウェア割り込みの選択

ラインを割り込みソースとして設定するには、次の手順を使用します。

- EXTI\_IMR レジスタの対応するマスクビットを設定します。
- 割り込みラインのトリガ選択ビットを設定します (EXTI\_RTISR および EXTI\_FTISR)。
- いずれかの EXTI ラインからの割り込みを正しく処理できるように、EXTI にマップされた NVIC IRQ チャネルを制御するイネーブルビットとマスクビットを設定します。

### ハードウェアイベントの選択

ラインをイベントソースとして設定するには、次の手順を使用します。

- EXTI\_EMR レジスタの対応するマスクビットを設定します。
- イベントラインのトリガ選択ビットを設定します (EXTI\_RTISR および EXTI\_FTISR)。

### ソフトウェア割り込み/イベントの選択

外部ラインのいずれかを、ソフトウェア割り込み/イベントラインとして設定できます。ソフトウェア割り込みの生成には、次の手順を使用します。

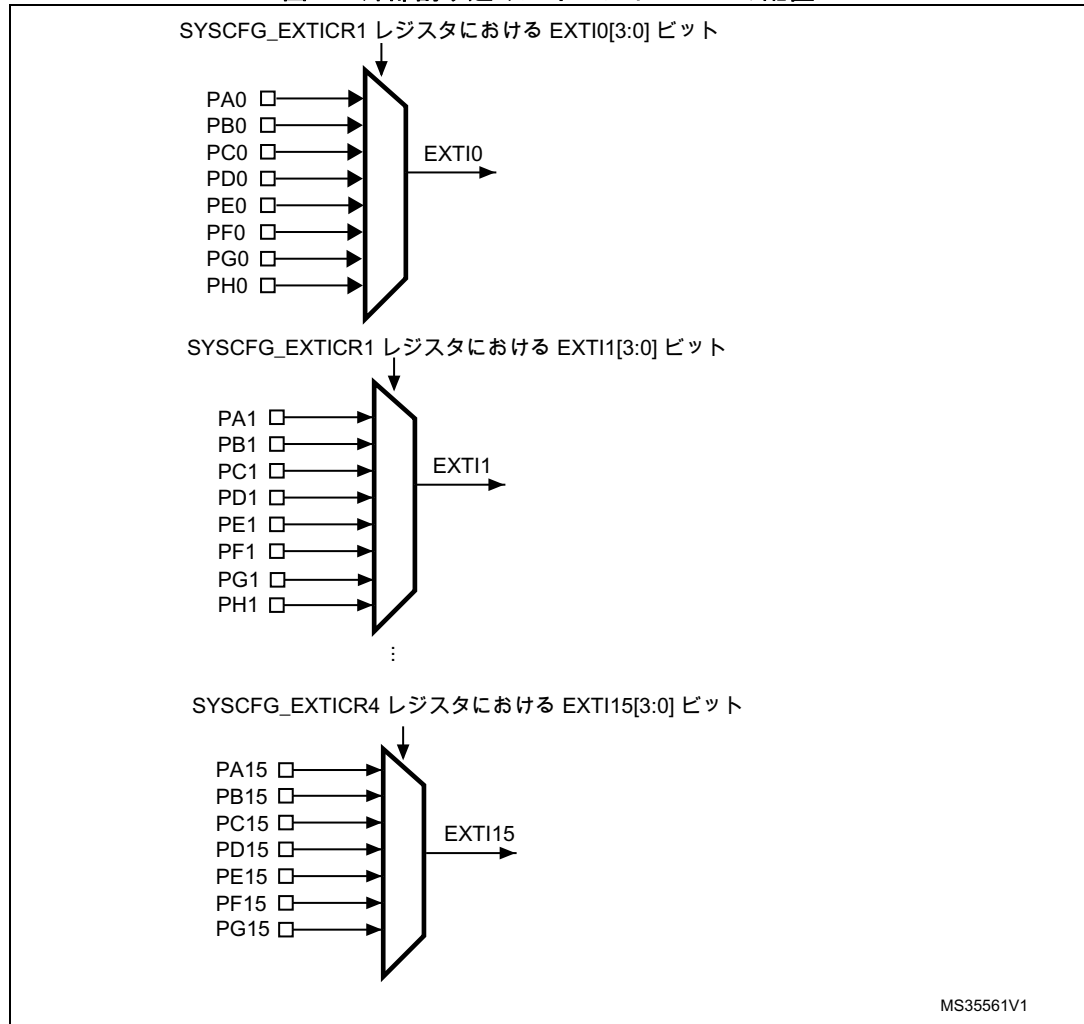
- 対応するマスクビット (EXTI\_IMR、EXTI\_EMR) を設定します。
- ソフトウェア割り込みレジスタの必要なビットをセットします (EXTI\_SWIER)。

## 14.2.6 外部および内部の割り込み／イベントラインの配置

36本の割り込み／イベントラインが使用できます。内訳は、内部が8本（予約済みのものも含む）、外部が28本です。

GPIOは、次のように、16本の外部割り込み／イベントラインに接続されます。

図 51. 外部割り込み／イベント GPIO の配置



残りのラインは、以下のように接続されます。

- EXTI ライン 16 は PVD 出力に接続されます。
- EXTI ライン 17 は RTC アラームイベントに接続されます。
- EXTI ライン 18 は USB デバイス FS ウェイクアップイベントに接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 19 は RTC のタンパおよびタイムスタンプに接続されます。
- EXTI ライン 20 は RTC ウェイクアップタイマに接続されます。
- EXTI ライン 21 はコンパレータ 1 出力に接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 22 はコンパレータ 2 出力に接続されます。
- EXTI ライン 23 は I2C1 ウェイクアップに接続されます。
- EXTI ライン 24 は I2C2 ウェイクアップに接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 25 は USART1 ウェイクアップに接続されます。
- EXTI ライン 26 は USART2 ウェイクアップに接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 27 は I2C3 ウェイクアップに接続されます (STM32F303xD/E および STM32F398xE デバイス)。
- EXTI ライン 28 は USART3 ウェイクアップに接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 29 はコンパレータ 3 出力に接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 30 はコンパレータ 4 出力に接続されます。
- EXTI ライン 31 はコンパレータ 5 出力に接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 32 はコンパレータ 6 出力に接続されます。
- EXTI ライン 33 はコンパレータ 7 出力に接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 34 は UART4 ウェイクアップに接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。
- EXTI ライン 35 は UART5 ウェイクアップに接続されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイス)。

**注：** EXTI ライン 23、24、25、26、27、28、34 および 35 は内部です。

## 14.3 EXTI レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

### 14.3.1 割り込みマスクレジスタ (EXTI\_IMR1)

アドレスオフセット : 0x00

リセット値 : 0x1F80 0000 (下の注を参照)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR31	MR30	MR29	MR28	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **MRx** : 外部/内部ライン x の割り込みマスク

0 : ライン x からの割り込みリクエストはマスクされます。

1 : ライン x からの割り込みリクエストはマスクされません。

**注 :** デフォルトで割り込みを有効にするために、内部ライン (23、24、25、26、27 および 28) のリセット値は“1”にセットされます。

### 14.3.2 イベントマスクレジスタ (EXTI\_EMR1)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR31	MR30	MR29	MR28	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **MRx** : 外部/内部ライン x のイベントマスク

0 : ライン x からのイベントリクエストはマスクされます。

1 : ライン x からのイベントリクエストはマスクされません。



## 14.3.3 立ち上がりトリガ選択レジスタ (EXTI\_RTISR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TR31	TR30	TR29	Res.	Res.	Res.	Res.	Res.	Res.	TR22	TR21	TR20	TR19	TR18	TR17	TR16
r/w	r/w	r/w							r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:29 **TRx** : ライン x の立ち上がりトリガイベント設定ビット (x = 31~29)

0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。

1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

ビット 28:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **TRx** : ライン x の立ち上がりトリガイベント設定ビット (x = 22~0)

0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。

1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

**注 :** 外部ウェイクアップラインはエッジトリガされています。これらのライン上でグリッチが生成されないようにする必要があります。EXTI\_RTISR レジスタへの書き込み中に外部割り込みラインで立ち上がりエッジが発生した場合、ペンディングビットはセットされません。

同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

## 14.3.4 立ち下がりトリガ選択レジスタ (EXTI\_FTISR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TR31	TR30	TR29	Res.	Res.	Res.	Res.	Res.	Res.	TR22	TR21	TR20	TR19	TR18	TR17	TR16
r/w	r/w	r/w							r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:29 **TRx** : ライン x の立ち下がりトリガイベント設定ビット (x = 31~29)

0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です

1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です

ビット 28:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **TRx** : ライン x の立ち下がりトリガイベント設定ビット (x = 22~0)

0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です

1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です



注： 外部ウェイクアップラインはエッジトリガされています。これらのライン上でグリッチが生成されないようにする必要があります。EXTI\_FTSR レジスタへの書き込み中に外部割り込みラインで立ち下がリエッジが発生した場合、ペンディングビットはセットされません。  
 同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

### 14.3.5 ソフトウェア割り込みイベントレジスタ (EXTI\_SWIER1)

アドレスオフセット：0x10  
 リセット値：0x0000 0000

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

ビット 31 : 29 **SWIERx** : ライン x のソフトウェア割り込み (x = 31~29)

割り込みが EXTI\_IMR のこのライン上で有効になっている場合、このビットが“0”のときに“1”を書き込むと EXTI\_PR の対応するペンディングビットがセットされるため、割り込みリクエストが生成されます。

EXTI\_PR レジスタの対応するビットをクリアする (対応ビットに“1”を書き込む) と、このビットはクリアされます。

ビット 28:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **SWIERx** : ライン x のソフトウェア割り込み (x = 22~0)

割り込みが EXTI\_IMR のこのライン上で有効になっている場合、このビットが“0”のときに“1”を書き込むと EXTI\_PR の対応するペンディングビットがセットされるため、割り込みリクエストが生成されます。

EXTI\_PR レジスタの対応するビットをクリアする (対応ビットに“1”を書き込む) と、このビットはクリアされます。

### 14.3.6 ペンディングレジスタ (EXTI\_PR1)

アドレスオフセット：0x14  
 リセット値：定義されていません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PR31	PR30	PR29	Res.	Res.	Res.	Res.	Res.	Res.	PR22	PR21	PR20	PR19	PR18	PR17	PR16
rc_w1	rc_w1	rc_w1							rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1



ビット 31:29 **PRx** : ライン x のペンディングビット (x = 31~29)

- 0 : トリガリクエストは発生していません。
- 1 : 選択されたトリガリクエストが発生しました。

このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。このビットは、ビットに“1”を書き込むことによってクリアされます。

ビット 28:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **PRx** : ライン x のペンディングビット (x = 22~0)

- 0 : トリガリクエストは発生していません。
- 1 : 選択されたトリガリクエストが発生しました。

このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。このビットは、ビットに“1”を書き込むことによってクリアされます。

### 14.3.7 割り込みマスクレジスタ (EXTI\_IMR2)

アドレスオフセット : 0x20

リセット値 : 0xFFFF FFFC (下の注を参照)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR35	MR34	MR33	MR32
												r/w	r/w	r/w	r/w

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **MRx** : 外部/内部ライン x の割り込みマスク、x = 32~35

- 0 : ライン x からの割り込みリクエストはマスクされます。
- 1 : ライン x からの割り込みリクエストはマスクされません。

**注 :** 内部ライン (EXTI ライン 34 および 35) のリセット値および予約済みラインは“1”にセットされます。

### 14.3.8 イベントマスクレジスタ (EXTI\_EMR2)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR35	MR34	MR33	MR32
												r/w	r/w	r/w	r/w

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **MRx** : 外部/内部ライン x のイベントマスク、x = 32~35

- 0 : ライン x からのイベントリクエストはマスクされます。
- 1 : ライン x からのイベントリクエストはマスクされません。

## 14.3.9 立ち上がりトリガ選択レジスタ (EXTI\_RTISR2)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR33	TR32
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TRx** : ライン x の立ち上がりトリガイベント設定ビット (x = 32、33)

- 0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。
- 1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

**注 :** 外部ウェイクアップラインはエッジトリガされています。これらのライン上でグリッチが生成されないようにする必要があります。EXTI\_RTISR レジスタへの書き込み中に外部割り込みラインで立ち上がりエッジが発生した場合、ペンディングビットはセットされません。

同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

## 14.3.10 立ち下がりトリガ選択レジスタ (EXTI\_FTISR2)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR33	TR32
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TRx** : ライン x の立ち下がりトリガイベント設定ビット (x = 32、33)

- 0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です
- 1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です



注： 外部ウェイクアップラインはエッジトリガされています。これらのライン上でグリッチが生成されないようにする必要があります。EXTI\_FTSR レジスタへの書き込み中に外部割り込みラインで立ち下がリエッジが発生した場合、ペンディングビットはセットされません。  
同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

### 14.3.11 ソフトウェア割り込みイベントレジスタ (EXTI\_SWIER2)

アドレスオフセット：0x30  
リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER33	SWIER32
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 SWIERx：ライン x のソフトウェア割り込み (x = 32, 33)

割り込みが EXTI\_IMR のこのライン上で有効になっている場合、このビットが“0”のときに“1”を書き込むと EXTI\_PR の対応するペンディングビットがセットされるため、割り込みリクエストが生成されます。

EXTI\_PR レジスタの対応するビットをクリアする (対応ビットに“1”を書き込む) と、このビットはクリアされます。

### 14.3.12 ペンディングレジスタ (EXTI\_PR2)

アドレスオフセット：0x34  
リセット値：定義されていません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR33	PR32
														rc_w1	rc_w1

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 PRx：ライン x のペンディングビット (x = 32, 33)

0：トリガリクエストは発生していません。

1：選択されたトリガリクエストが発生しました。

このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。このビットは、ビットに“1”を書き込むことによってクリアされます。

## 14.3.13 EXTI レジスタマップ

次の表に、EXTI レジスタマップとリセット値を示します。

表 83. 外部割り込み/イベントコントローラのレジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	EXTI_IMR1	MR[31:0]																																
	リセット値	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR1	MR[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_RTISR1	TR[31:29]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[22:0]																							
	リセット値	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	EXTI_FTISR1	TR[31:29]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[22:0]																							
	リセット値	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	EXTI_SWIER1	SWIER [31:29]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER[22:0]																							
	リセット値	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	EXTI_PR1	PR [31:29]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[22:0]																							
	リセット値	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	EXTI_IMR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR35	MR34	MR33	MR32
	リセット値																													1	1	0	0	
0x24	EXTI_EMR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR35	MR34	MR33	MR32
	リセット値																													0	0	0	0	
0x28	EXTI_RTISR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR33	TR32	
	リセット値																															0	0	
0x2C	EXTI_FTISR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR33	TR32	
	リセット値																															0	0	
0x30	EXTI_SWIER2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER33	SWIER32	
	リセット値																															0	0	



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

表 83. 外部割り込み/イベントコントローラのレジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	EXTI_PR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																0

レジスタ境界アドレスについては、[セクション 3.2.2: メモリマップとレジスタ境界アドレス](#) を参照してください。

## 15 アナログデジタルコンバータ (ADC)

### 15.1 概要

このセクションでは、最大 4 つ実装されている ADC について説明します。

- ADC1 および ADC2 は密接に結合されており、デュアルモードで動作できません (ADC1 はマスターです)。
- ADC3 および ADC4 は密接に結合されており、デュアルモードで動作できません (ADC3 はマスターです)。

各 ADC は、12 ビットの逐次比較型アナログデジタルコンバータで構成されています。

各 ADC には、最大 19 の多重化チャネルがあります。さまざまなチャネルの A/D 変換は、シングル、連続、スキャン、または不連続モードで行うことができます。ADC の結果は、左詰めまたは右詰めでの 16 ビットのデータレジスタに格納されます。

ADC は AHB バスに配置され、データを高速で処理できます。

アナログウォッチドッグ機能により、入力電圧が、ユーザ定義の上限値または下限値から逸脱していないかを、アプリケーションで検出することができます。

低周波数で非常に低い消費電力を可能にするために、効率的な低電力モードが実装されています。

**注：** *STM32F303x6/8 および STM32F328x8 デバイスには、ADC1 と ADC2 のみがあります。*

## 15.2 ADC の主な機能

- ハイパフォーマンス機能
  - 最大 4つの ADC、それぞれデュアルモードで動作可能
  - 次の表に、ADC で使用可能な異なる外部チャンネルを示します。

表 84. ADC 外部チャンネルマッピング

デバイス	ADC1	ADC2	ADC3	ADC4
STM32F303xB/C	10	12	15	13
STM32F358	10	11	15	13
STM32F303x6/8	11	14	N/A	N/A
STM32F328	11	13	N/A	N/A
STM32F303xD/E	11	13	15	13
STM32F398xE	11	12	15	13

- 12、10、8、または 6 ビットに設定可能な分解能
- ADC 変換時間 :
  - 高速チャンネル : 12 ビット分解能で 0.19  $\mu$ s (5.1 Ms/s)
  - 低速チャンネル : 12 ビット分解能で 0.21  $\mu$ s (4.8 Ms/s)
- ADC 変換時間は AHB バスクロック周波数から独立
- 分解能の減少による高速変換時間 : 10 ビット分解能で 0.16  $\mu$ s
- シングルエンドまたは差動入力 (チャンネルごとにプログラム可能) を管理可能
- 高速データ処理を可能にする AHB スレーブバスインタフェース
- 自己較正
- チャンネル単位でプログラム可能なサンプリング時間
- 最大 4つのインジェクトチャンネル (アナログ入力をレギュラまたはインジェクトチャンネルへ完全に割り当て設定可能)
- 高速コンテキスト切り替えを可能にするためにインジェクトチャンネルの内容を準備するハードウェアによる支援
- 組み込みのデータコヒーレンシによるデータ配置
- データは、レギュラチャンネル変換において、GP-DMA で管理可能
- 4つのインジェクトチャンネル専用のデータレジスタ
- 低電力機能
  - 低周波数での動作時に ADC 消費電流を低減するための速度適応型低電力モード
  - 最適な ADC 性能を維持しながら、低バス周波数を適用可能 (AHB バスクロック周波数にかかわらず、高速チャンネルで 0.19  $\mu$ s 変換時間を維持可能)
  - 低 AHB バスクロック周波数アプリケーション (自動遅延モード) で ADC オーバーランを避けるための自動制御を提供
- 4つの ADC それぞれに対する外部アナログ入力チャンネル :
  - 専用の GPIO パッドから最大 5つの高速チャンネル
  - 専用の GPIO パッドから最大 11つの低速チャンネル
- さらに、ADC で使用可能な専用の内部チャンネルがあります。次の表の一覧を参照してください。:



表 85. ADC 内部チャネルの概要

製品	ADC1	ADC2	ADC3	ADC4	ADC 内部チャネルの合計
STM32F303xB/C/D/E、 STM32F358、および STM32F398xE	<ul style="list-style-type: none"> <li>- 温度センサに接続されたチャネル x 1</li> <li>- VBAT/2 に接続されたチャネル x 1</li> <li>- VREFINT に接続されたチャネル x 1</li> <li>- OPAMP1 基準電圧出力 (VREFOPAMP1) に接続されたチャネル x 1</li> </ul>	<ul style="list-style-type: none"> <li>- VREFINT に接続されたチャネル x 1</li> <li>- OPAMP2 基準電圧出力 (VREFOPAMP2) に接続されたチャネル x 1</li> </ul>	<ul style="list-style-type: none"> <li>- VREFINT に接続されたチャネル x 1</li> <li>- OPAMP3 基準電圧出力 (VREFOPAMP3) に接続されたチャネル x 1</li> </ul>	<ul style="list-style-type: none"> <li>- VREFINT に接続されたチャネル x 1</li> <li>- OPAMP4 基準電圧出力 (VREFOPAMP4) に接続されたチャネル x 1</li> </ul>	7
STM32F303x6/8 および STM32F328	<ul style="list-style-type: none"> <li>- 温度センサに接続されたチャネル x 1</li> <li>- VBAT/2 に接続されたチャネル x 1</li> <li>- VREFINT に接続されたチャネル x 1</li> <li>- OPAMP1 基準電圧出力 (VREFOPAMP1) に接続されたチャネル x 1</li> </ul>	<ul style="list-style-type: none"> <li>- VREFINT に接続されたチャネル x 1</li> <li>- OPAMP2 基準電圧出力 (VREFOPAMP2) に接続されたチャネル x 1</li> </ul>	N/A	N/A	5

- 変換開始は、次のように開始できます。
  - レギュラ変換とインジェクト変換の両方について、ソフトウェアによって開始
  - レギュラ変換とインジェクト変換の両方について、極性が設定可能なハードウェアトリガによって開始 (内部タイマイベントまたは GPIO 入力イベント)
- 変換モード
  - 各 ADC では単一チャネルの変換、または一連のチャネルをスキャンできます。
  - シングルモードは、選択された入力をトリガごとに1回変換します。
  - 連続モードは、選択された入力を連続的に変換します。
  - 不連続モード
- デュアル ADC モード
- (レギュラまたはインジェクト) 変換終了時、(レギュラまたはインジェクト) シーケンス変換終了時、アナログウォッチドッグ 1、2、3、またはオーバーランイベント時に割り込みを生成
- ADC ごとの 3 つのアナログウォッチドッグ
- ADC 電源仕様 : 1.80 V から 3.6 V
- ADC 入力電圧範囲 :  $V_{REF-} \leq V_{IN} \leq V_{REF+}$

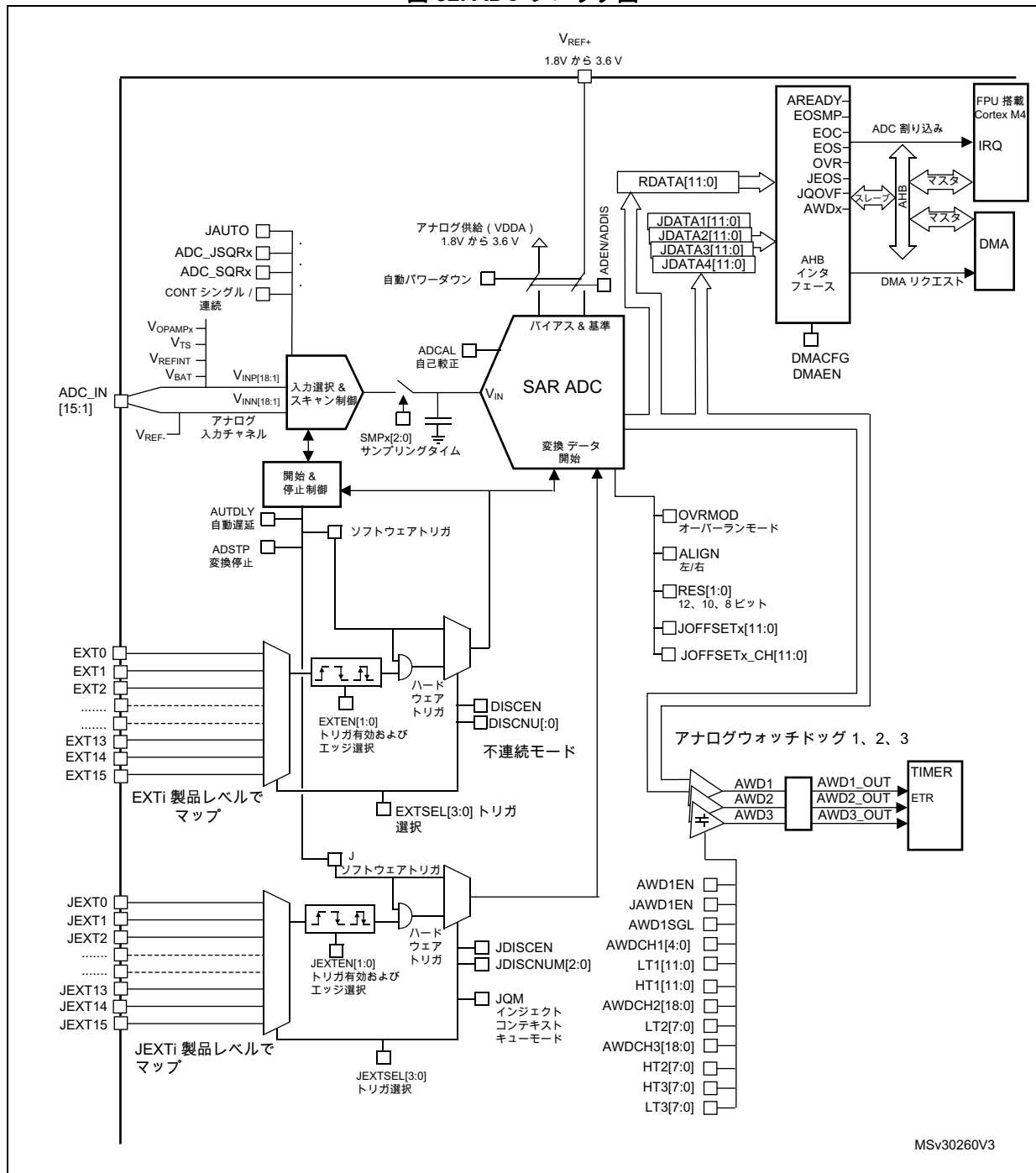
図 52 に、1 つの ADC のブロック図を示します。

## 15.3 ADC の機能詳細

### 15.3.1 ADC ブロック図

図 52 に ADC ブロック図を、表 87 に ADC ピンの説明を示します。

図 52. ADC ブロック図



## 15.3.2 ピンおよび内部信号

表 86. ADC 内部信号

内部信号名	信号タイプ	説明
EXT[15:0]	入力	レギュラ変換の最大 16 個の外部トリガ入力 (オンチップタイマに接続可能)。これらの入力は、ADC マスタと ADC スレーブの間で共有されています。
JEXT[15:0]	入力	インジェクト変換の最大 16 個の外部トリガ入力 (オンチップタイマに接続可能)。これらの入力は、ADC マスタと ADC スレーブの間で共有されています。
ADC1_AWDx_OUT ADC2_AWDx_OUT ADC3_AWDx_OUT ADC4_AWDx_OUT	出力	オンチップタイマに接続された内部アナログウォッチドッグの出力信号 (x = アナログウォッチドッグの数 1、2、3)
V <sub>REFOPAMP1</sub>	入力	内部オペアンプ 1 からの基準電圧出力
V <sub>REFOPAMP2</sub>	入力	内部オペアンプ 2 からの基準電圧出力
V <sub>REFOPAMP3</sub>	入力	内部オペアンプ 3 からの基準電圧出力
V <sub>REFOPAMP4</sub>	入力	内部オペアンプ 4 からの基準電圧出力
V <sub>TS</sub>	入力	内部温度センサからの出力電圧
V <sub>REFINT</sub>	入力	内部基準電圧からの出力電圧
V <sub>BAT</sub>	入力電源供給	外部バッテリー電圧供給

表 87. ADC ピン

名前	信号タイプ	コメント
V <sub>REF+</sub>	入力、アナログ基準電圧正	ADC のハイ側 / 正基準電圧 $1.8\text{ V} \leq V_{\text{REF}+} \leq V_{\text{DDA}}$
V <sub>DDA</sub>	入力、アナログ電源供給	V <sub>DDA</sub> に等しいアナログ電源供給 : $1.8\text{ V} \leq V_{\text{DDA}} \leq 3.6\text{ V}$
V <sub>REF-</sub>	入力、アナログ基準電圧負	ADC のロー側 / 負基準電圧 $V_{\text{REF}-} = V_{\text{SSA}}$
V <sub>SSA</sub>	入力、アナログ供給グラウンド	V <sub>SS</sub> に等しいアナログ電源供給のグラウンド
V <sub>INP</sub> [18:1]	各 ADC の正のアナログ入力チャネル	外部チャネル ADC_IN <i>i</i> または内部チャネルのいずれかに接続
V <sub>INN</sub> [18:1]	各 ADC の負のアナログ入力チャネル	V <sub>REF-</sub> または外部チャネル ADC_IN <i>i-1</i> に接続
ADCx_IN[16:1]	外部アナログ入力信号	最大 16 アナログ入力チャネル (x = ADC 数 = 1、2、3、または 4) : - 5 つの高速チャネル - 10 の低速チャネル

## 15.3.3 クロック

### デュアルクロックドメインアーキテクチャ

デュアルクロックドメインアーキテクチャは、各 ADC クロックが AHB バスクロックから独立していることを意味します。

2 つの ADC (マスタおよびスレーブ) の入力クロックは、2 つのクロックソースから選択できます (図 53 : ADC クロック構成を参照)。

- a) ADC クロックは、「ADCxy\_CK (xy=12 または 34)」という名前の AHB クロックとは非同期の、独立した特定のクロックソースです。

最大 72 MHz (PLL 出力) で供給するように RCC で設定できます。ADC12\_CK および ADC34\_CK の生成の詳細については、RCC のセクションを参照してください。

この構成を選択するには、ADCx\_CCR レジスタのビット CKMODE[1:0] をリセットする必要があります。

- b) ADC クロックは、ADC バスインタフェースの AHB クロックから生成し、プログラム可能な係数 (1、2、または 4) で分周することができます。このモードでは、プログラム可能な分周比を選択できます (ビット CKMODE[1:0] に従って /1、2、または 4)。

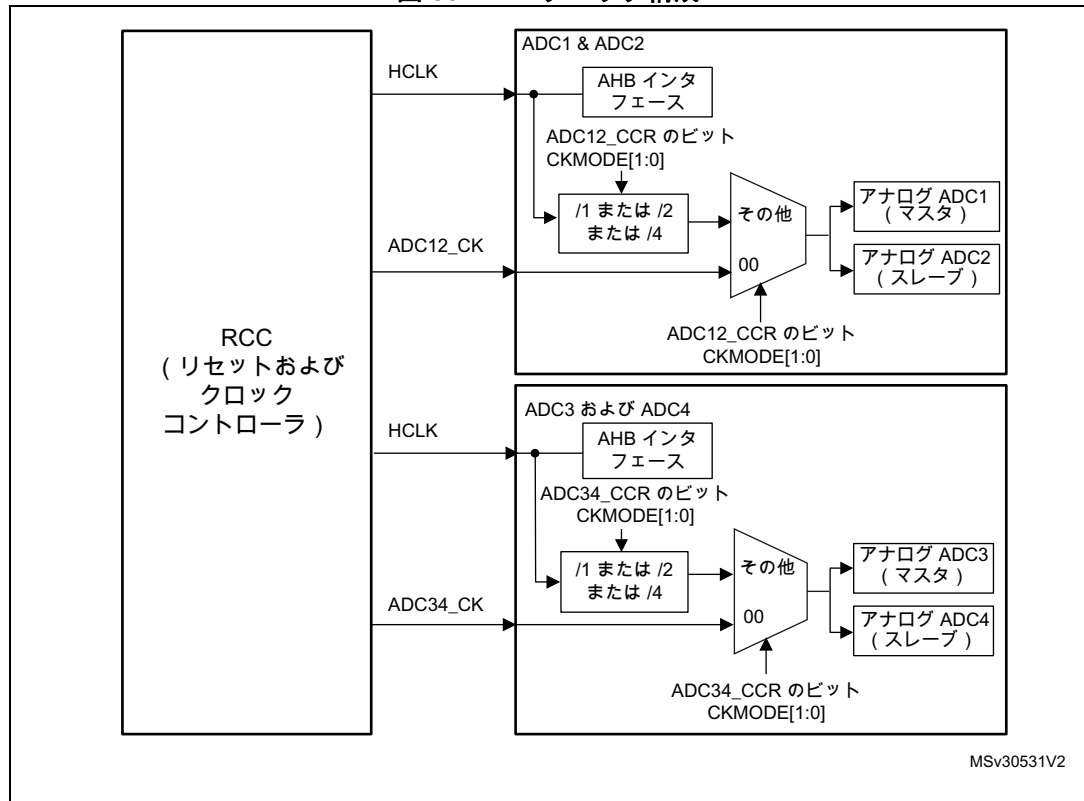
この構成を選択するには、ADCx\_CCR レジスタのビット CKMODE[1:0] が「00」以外の値である必要があります。

**注：** ソフトウェアでは、RCC の AHB プリスケーラが 1 にセットされている場合のみ CKMODE[1:0]=01 を書き込むことでオプション b) を使用できません (AHB クロックのデューティサイクルは、この設定では 50% である必要があります)。

オプション a) には、選択された AHB クロック構成にかかわらず、最大の ADC クロック周波数に到達できるという利点があります。ADC クロックは、最終的には 1、2、4、6、8、12、16、32、64、128、256 の比率で分周できます。このとき、レジスタ RCC\_CFGR2 の ADCxPRES[4:0] ビットで設定されたプリスケーラを使用します (セクション 9 : リセットおよびクロック制御 (RCC) を参照)。

オプション b) には、クロックドメインの再同期を回避するという利点があります。これは、ADC がタイマによってトリガされるときと、アプリケーションが ADC の確実に精密なトリガを必要とする場合に便利です (そうしないと、トリガ時の不確実性は、2 つのクロックドメイン間の再同期によって高まります)。

図 53. ADC クロック構成



1. HCLK、ADC12\_CK、および ADC34\_CK の生成の詳細については、RCC のセクションを参照してください。

## ADC クロックと AHB クロックの間におけるクロック比の制約

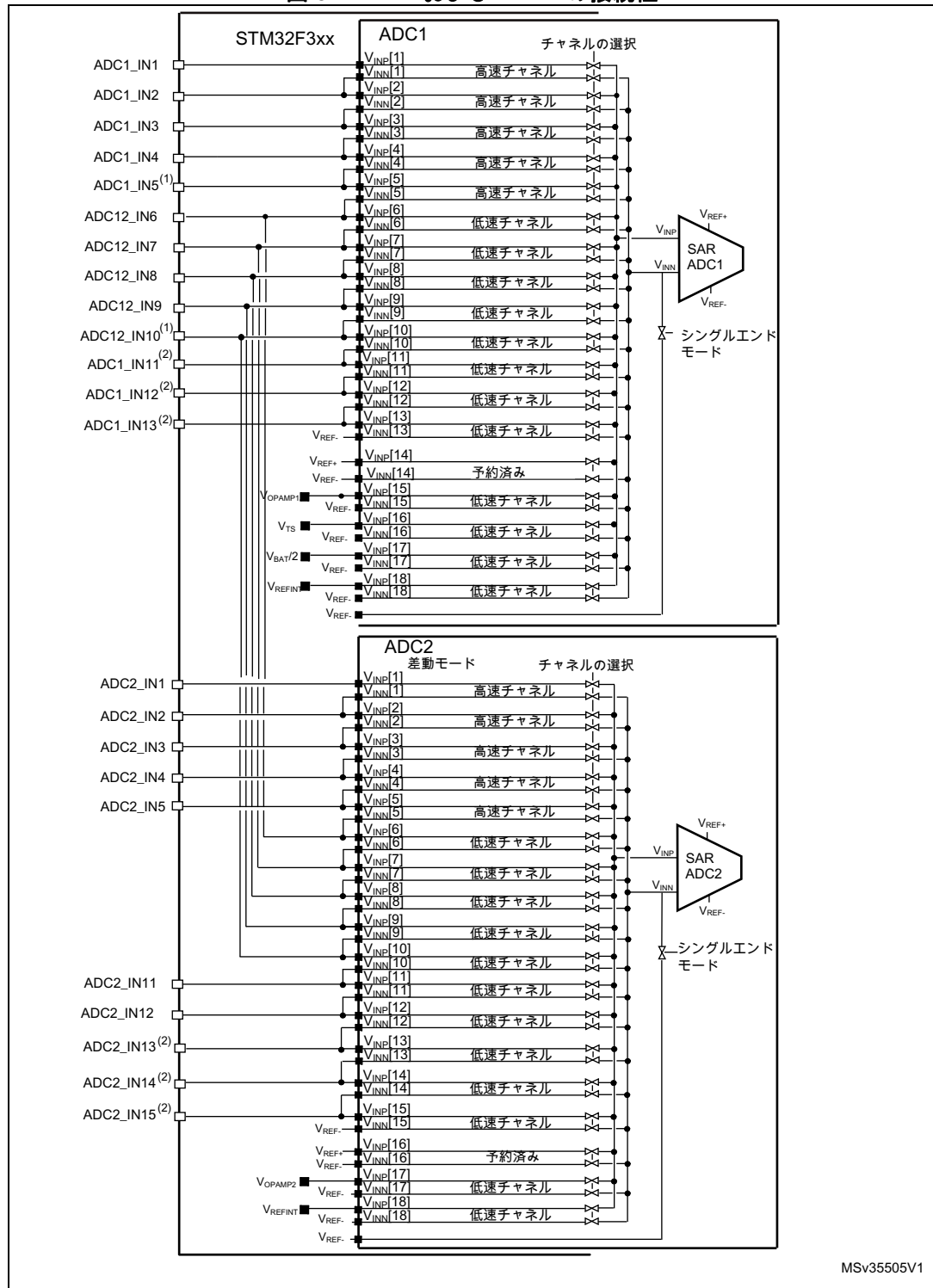
通常、ADC クロックと AHB クロックの間で注意すべき制約はありませんが、インジェクトチャンネルがプログラムされている場合は例外です。この場合、次の比率に注意する必要があります。

- すべてのチャンネルの分解能が 12 ビットまたは 10 ビットである場合、 $F_{HCLK} \geq F_{ADC} / 4$
- いくつかのチャンネルの分解能が 8 ビットである場合（それを下回る分解能のチャンネルがない場合）、 $F_{HCLK} \geq F_{ADC} / 3$
- いくつかのチャンネルの分解能が 6 ビットである場合、 $F_{HCLK} \geq F_{ADC} / 2$

## 15.3.4 ADC1/2 および ADC3/4 の接続性

ADC1 および ADC2 (ADC3 および ADC4も同様) は密接に結合されており、いくつかの外部チャンネルを共有しています。図 54 および 図 55 を参照してください。

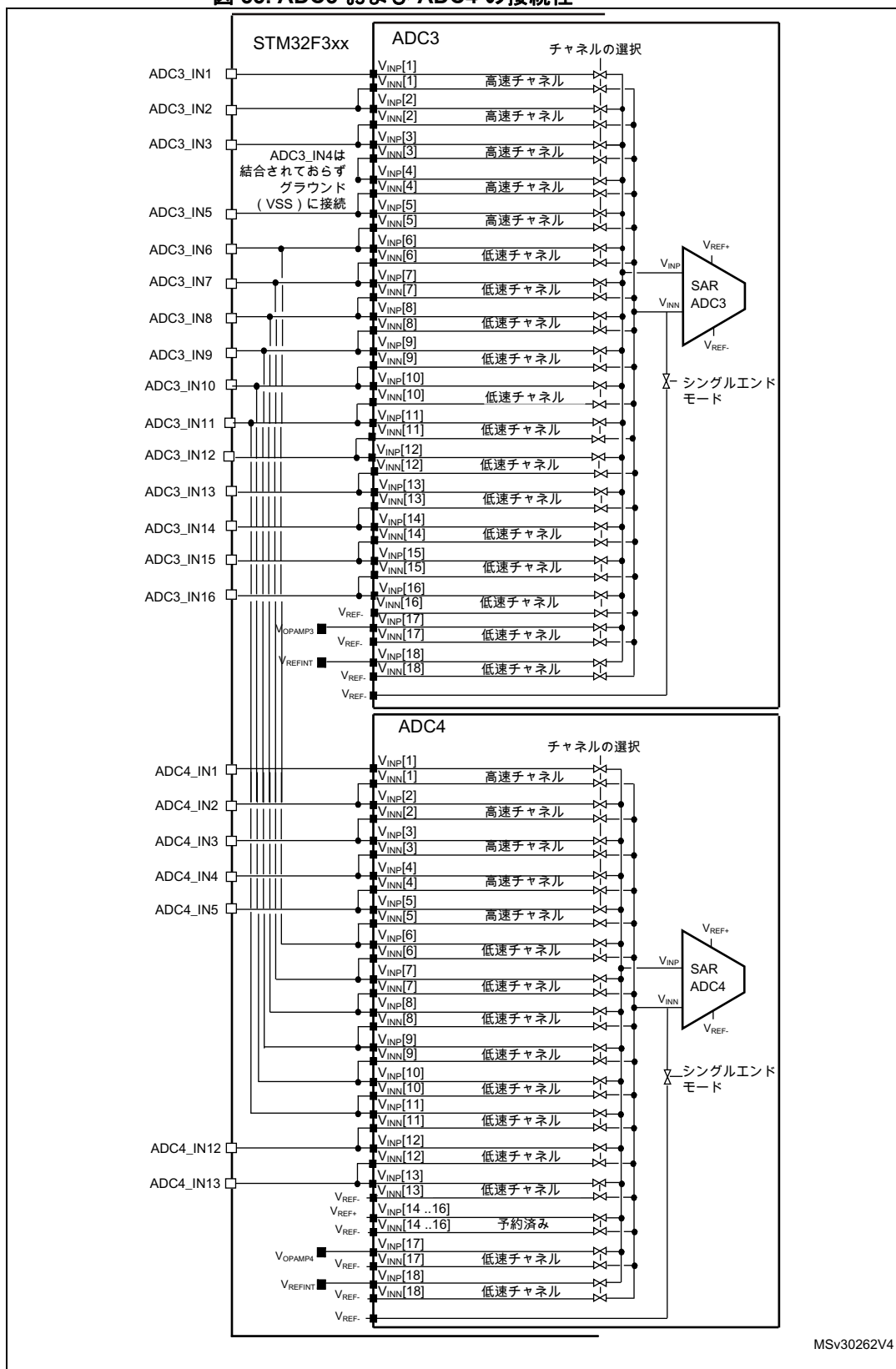
図 54. ADC1 および ADC2 の接続性



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

1. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ。
2. STM32F303x6/8 および STM32F328x8 デバイスのみ。

図 55. ADC3 および ADC4 の接続性



MSV30262V4



## 15.3.5 スレーブ AHB インタフェース

ADC は、制御/ステータスレジスタおよびデータアクセスのために AHB スレーブポートを実装しています。AHB インタフェースの機能を次に示します。

- ワード (32 ビット) アクセス
- シングルサイクルレスポンス
- レジスタへのすべての読み出し/書き込みアクセスに対してゼロウェイトステートの応答。

AHB スレーブインタフェースでは分割/再試行リクエストをサポートしておらず、また AHB エラーも発生しません。

## 15.3.6 ADC 電圧レギュレータ (ADVREGEN)

ADC 動作を開始するためには次のシーケンスが必要です。

1. ADC 内部電圧レギュレータを有効にします (ADC 電圧レギュレータイネーブルシーケンスを参照)。
2. ソフトウェアは、較正を起動するか ADC を有効化する前に、ADC 電圧レギュレータの起動時間の間待つ必要があります ( $T_{\text{ADCVREG\_STUP}}$ )。この過渡期間は、ソフトウェアによって実装する必要があります。最悪のケースのプロセス/温度/電源で、 $T_{\text{ADCVREG\_STUP}}$  は 10  $\mu\text{s}$  に等しくなります。

ADC の動作が完了すると、ADC は無効化されます (ADEN=0)。

ADC 電圧レギュレータを無効にすることで節電できます (ADC 電圧レギュレータの無効化シーケンスを参照してください)。

**注：** *内部電圧レギュレータが無効なときには、内部アナログ較正が保持されます。*

### ADVREG 有効化シーケンス

ADC 電圧レギュレータを有効にするには、以下のシーケンスを実行します。

1. ADVREGEN[1:0] ビットを「10」(無効状態、リセット状態) から「00」に変更します。
2. ADVREGEN[1:0] ビットを「00」から「01」(有効状態) に変更します。

### ADVREG 無効化シーケンス

ADC 電圧レギュレータを無効にするには、以下のシーケンスを実行します。

1. ADVREGEN[1:0] ビットを「01」(有効状態) から「00」に変更します。
2. ADVREGEN[1:0] ビットを「00」から「10」(無効状態) に変更します。

## 15.3.7 シングルエンドおよび差動入力のチャネル

ADCx\_DIFSEL レジスタのビット DIFSEL[15:1] に書き込むことで、チャネルを、シングルエンド入力または差動入力のいずれかに設定できます。この設定は、ADC が無効 (ADEN=0) のときに書き込む必要があります。DIFSEL[18:16] がシングルエンドチャネル (内部チャネルのみ) に固定され、常に 0 として読み出される点に注意してください。

シングルエンド入力モードで、チャネル「i」に変換されるアナログ電圧は、外部電圧 ADC\_INi (正の入力) と  $V_{\text{REF}}$  (負の入力) の差です。

差動入力モードで、チャネル「i」に変換されるアナログ電圧は、外部電圧 ADC\_INi (正の入力) と ADC\_INi+1 (負の入力) の差です。

各 ADC への入力チャネルの接続方法の詳細については、[図 54 : ADC1 および ADC2 の接続性 \(310 ページ\)](#) および [図 55 : ADC3 および ADC4 の接続性 \(312 ページ\)](#) を参照してください。

**注意：** 差動入力モードでチャンネル「i」を設定する場合、負の入力電圧はADC\_INi+1 に接続されます。その結果、チャンネル「i+1」はシングルエンドモードや差動モードでは使用できなくなり、この先変換するために設定してはいけません。

ADC1 と ADC2 では一部のチャンネルを共有しています (ADC3 および ADC4 も同様)。これにより、ほかの ADC のチャンネルが使用不可になる場合があります。唯一の例外は、ADC マスタとスレーブのインタリーブモードです。

例: ADC1\_IN5 を差動入力モードに設定すると、ADC12\_IN6 が使用不可になります。その場合、ADC1 と ADC2 の両方のチャンネル 6 を変換してはなりません。

**注：** ADC1 のチャンネル 16、17、および 18、および ADC2、ADC3、および ADC4 のチャンネル 17 および 18 は内部アナログチャンネルに接続されていて、内部的にシングルエンド入力設定に固定されます (対応するビット DIFSEL[i] は常にゼロです)。ADC1 のチャンネル 15 も内部チャンネルであり、ユーザは対応するビット DIFSEL[15] をゼロに設定する必要があります。

## 15.3.8 較正 (ADCAL、ADCALDIF、ADCx\_CALFACT)

各 ADC は、ADC のパワーオン/オフシーケンスを含むすべての較正シーケンスを駆動する、自動較正手順を提供しています。この処理時、ADC は、次の ADC のパワーオフまで ADC に内部で適用される、7 ビット幅の較正係数を計算します。較正処理時、アプリケーションでは ADC を使用してはならず、較正が完了するまで待つ必要があります。

較正は、すべての ADC 動作に先立って行われます。較正は、プロセスまたはバンドギャップのばらつきによりチップごとに異なるオフセットエラーを削除します。

シングルエンド入力変換に適用される較正係数は、差動入力変換に適用される係数とは異なります。

- シングルエンド入力変換に適用される較正を起動する前に、ADCALDIF=0 を書き込みます。
- 差動入力変換に適用される較正を起動する前に、ADCALDIF=1 を書き込みます。

較正は、ビット ADCAL=1 をセットすることによって、ソフトウェアによって開始されます。較正は、ADC が無効のとき (ADEN=0 のとき) だけ開始できます。すべての較正シーケンス時、ADCAL ビットは 1 のままです。較正が完了すると、ハードウェアによってクリアされます。このとき、関連する較正係数は、アナログ ADC と、ADCx\_CALFACT レジスタのビット CALFACT\_S[6:0] または CALFACT\_D[6:0] に内部的に格納されます (シングルエンドであるか差動入力較正であるかによって異なる)。

ADC が無効の場合 (ADEN=0)、内部アナログ較正が保持されます。ただし、ADC が長時間無効である場合は、ADC を再度有効にする前に、新しい較正サイクルを実行することを推奨します。

ADC が無効の場合 (ADEN=0)、内部アナログ較正が保持されます。ADC 動作条件が変化したときには ( $V_{REF+}$  の変化が ADC オフセットのばらつきの主因であり、 $V_{DDA}$  と温度変化はそれほど影響しません)、較正サイクルを再実行することが推奨されます。

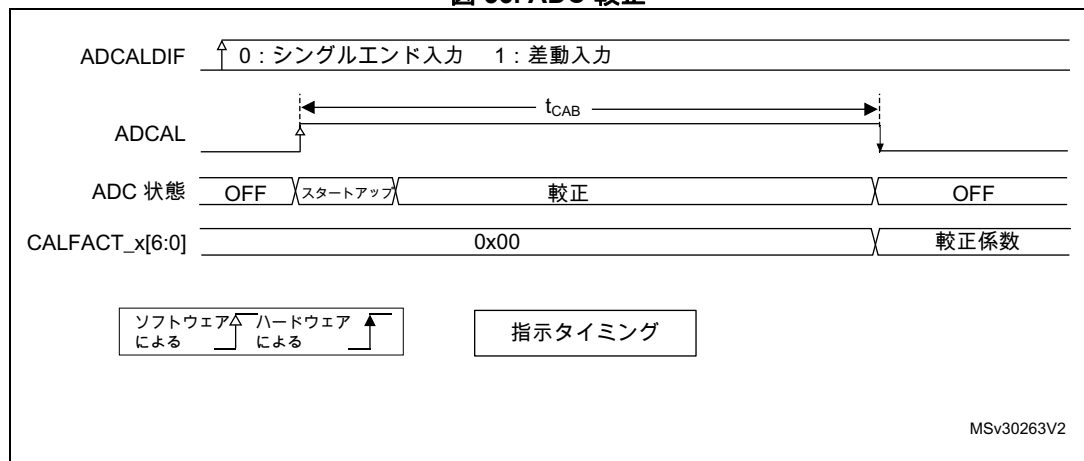
内部アナログ較正は ADC の電源がオフになるたびに失われます (たとえば、製品が STANDBY または VBAT モードになった場合)。この場合、ADC の再較正に時間をかけるのを避けるためには、ソフトウェアが前回の較正時に較正係数を保存していた場合にかぎり、再較正せずに ADCx\_CALFACT レジスタに較正係数を再度書き込むことができます。

ADC が有効であり、変換中でない場合 (ADEN=1 かつ ADSTART=0 かつ JADSTART=0)、較正係数を書き込むことができます。その場合、次の変換開始時に、較正係数がアナログ ADC に自動的にインジェクトされます。このローディングは透過的であり、変換開始のサイクル遅延は増加しません。

## ソフトウェアでの ADC の較正手順

1. ADVREGEN[1:0]=01 であり、ADC 電圧レギュレータの起動時間が経過していることを確認します。
2. ADEN=0 であることを確認します。
3. ADCALDIF=0 (シングルエンド入力) または ADCALDIF=1 (差動入力) にセットして、この較正の入力モードを選択します。
4. ADCAL=1 にセットします。
5. ADCAL=0 になるまで待ちます。
6. 較正係数を ADCx\_CALFACT レジスタから読み出すことができます。

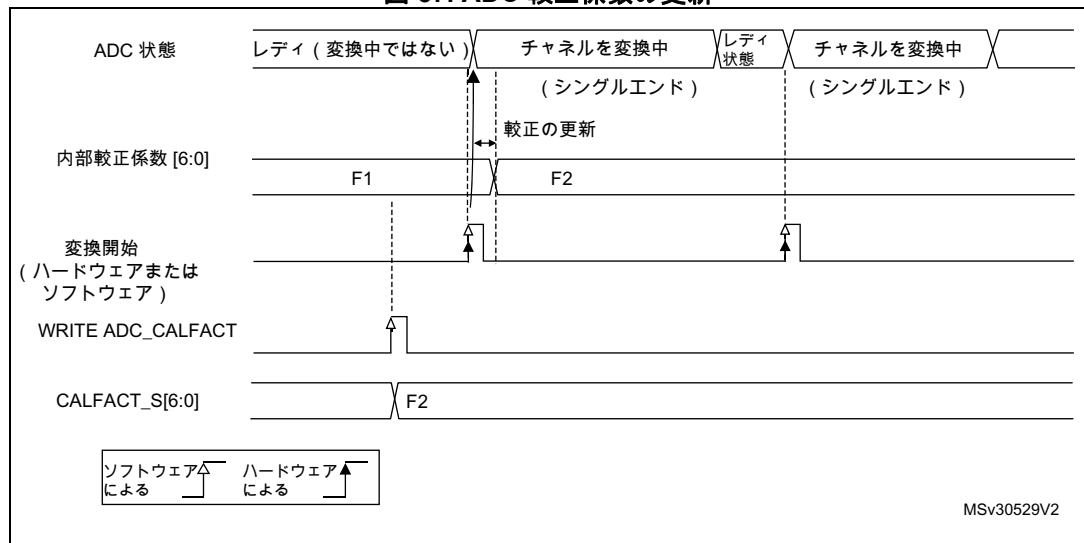
図 56. ADC 較正



## ソフトウェアでの ADC への較正係数の再インジェクト手順

1. ADEN=1 かつ ADSTART=0 かつ JADSTART=0 である (ADC が有効で変換中でない) ことを確認します。
2. 新しい較正係数で CALFACT\_S および CALFACT\_D を書き込みます。
3. 変換を起動すると、較正係数がアナログ ADC にインジェクトされます。これは、内部アナログ較正係数が、シングルエンド入力チャンネルの場合はビット CALFACT\_S、差動入力チャンネルの場合はビット CALFACT\_D に格納された較正係数と異なる場合にのみ有効です。

図 57. ADC 較正係数の更新

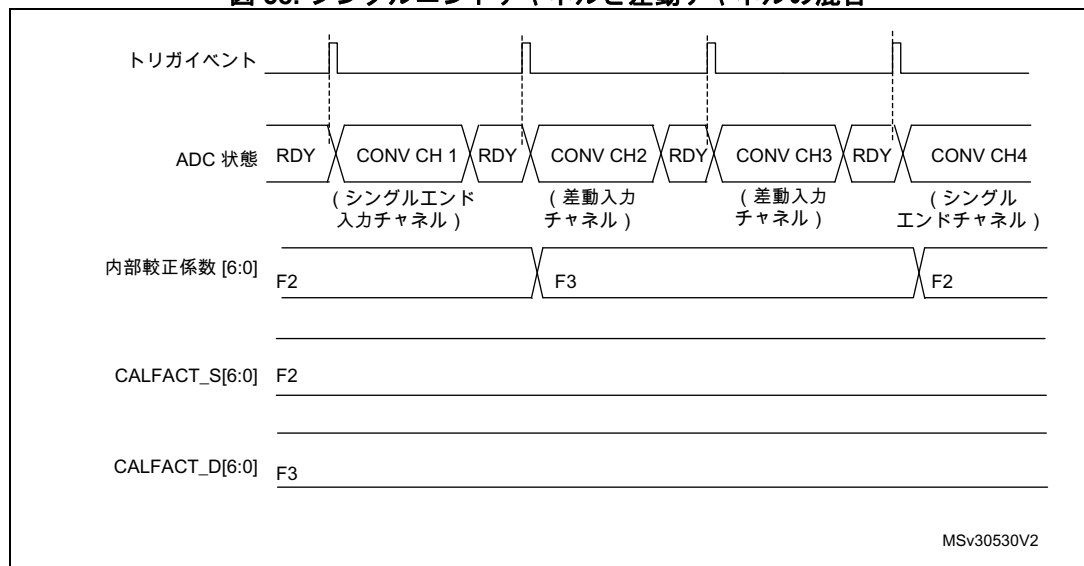


## シングル ADC によるシングルエンドおよび差動アナログ入力の変換

ADC で、差動入力とシングルエンド入力の両方を変換することを想定している場合、ADCALDIF=0 と ADCALDIF=1 の 2 つの較正を実行する必要があります。手順は、次のとおりです。

1. ADC を無効にします。
2. シングルエンド入力モードで ADC を較正します (ADCALDIF=0)。これにより、レジスタ CALFACT\_S[6:0] が更新されます。
3. 差動入力モードで ADC を較正します (ADCALDIF=1)。これにより、レジスタ CALFACT\_D[6:0] が更新されます。
4. ADC を有効化し、チャンネルを設定して変換を起動します。シングルエンド入力チャンネルから差動入力チャンネル（またはその逆）に切り替えられるたびに、アナログ ADC では自動的に較正が行われます。

図 58. シングルエンドチャンネルと差動チャンネルの混合



## 15.3.9 ADC オン/オフ制御 (ADEN, ADDIS, ADRDY)

最初に、[セクション 15.3.6 : ADC 電圧レギュレータ \(ADVREGEN\)](#) で説明している手順に従ってください。

一度 ADVREGEN[1:0] = 01 になると ADC を有効化することができ、[図 59](#)にあるように、ADC は正確な変換を開始する前に、安定時間  $t_{STAB}$  を必要とします。ADC は、次の 2 つの制御ビットによって有効化または無効化されます。

- ADC は、ADEN=1 によって有効化されます。ADC の動作準備ができると、ADRDY フラグがセットされます。
- ADC は ADDIS=1 によって無効化されます。アナログ ADC が実質的に無効になると、ADEN および ADDIS はハードウェアによって自動的にクリアされます。

その後、ADSTART=1 をセットすることによって ([セクション 15.3.18 : 外部トリガおよびトリガ極性での変換 \(EXTSEL, EXTEN, JEXTSEL, JEXTEN\)](#) を参照)、またはトリガが有効な場合に外部トリガイベントが発生したときに、レギュラ変換を開始できます。

JADSTART=1 をセットすることによって、またはインジェクトリガが有効な場合に外部インジェクトトリガイベントが発生したときに、インジェクト変換を開始できます。

### ソフトウェアでの ADC の有効化手順

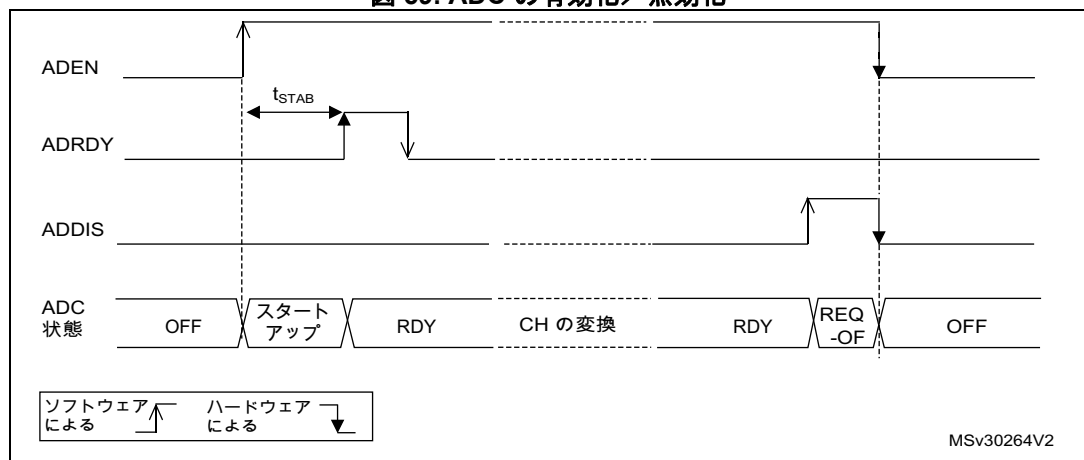
1. ADEN=1 にセットします。
2. ADRDY=1 になるまで待ちます (ADRDY は ADC 起動時間後にセットされます)。これは、対応する割り込みを使用することで実行できます (ADRDYIE=1 をセットします)。

**注 :** **ADCAL=1 の間、および、ADCAL ビットがハードウェアによってクリアされた後の 4 ADC クロックサイクルの間は、ADEN ビットをセットすることはできません (校正の終了)。**

### ソフトウェアでの ADC の無効化手順

1. ADSTART=0 かつ JADSTART=0 であることを確認し、変換が実行中でないことを確認します。必要な場合は、ADSTP=1 および JADSTP=1 にセットして、実行中のレギュラ変換とインジェクト変換を停止し、ADSTP=0 および JADSTP=0 になるまで待ちます。
2. ADDIS=1 にセットします。
3. アプリケーションによって必要とされる場合、ADEN=0 になり、アナログ ADC が実質的に無効になるまで待ちます (ADEN=0 になると、ADDIS は自動的にリセットされます)。

図 59. ADC の有効化/無効化



## 15.3.10 ADC 制御ビット書き込み時の制約

ソフトウェアでは、ADC が無効である場合のみ (ADEN は 0 である必要があります)、ADC クロック (RCC セクションを参照)、ADCx\_DIFSEL レジスタの制御ビット DIFSEL、および ADCx\_CR レジスタの制御ビット ADCAL と ADEN を設定して有効化するために、RCC 制御ビットを書き込むことができます。

ADC が有効であり、ADC を無効にするための保留中のリクエストがない場合のみ (ADEN=1 かつ ADDIS=0 である必要があります)、ソフトウェアは ADCx\_CR レジスタの制御ビット ADSTART、JADSTART、および ADDIS を書き込むことができます。

ADCx\_CFGR、ADCx\_SMPRx、ADCx\_TRx、ADCx\_SQRx、ADCx\_JDRy、ADCx\_OFRy、ADCx\_OFCHR、および ADCx\_IER レジスタのその他の制御ビットについては、次のとおりです。

- レギュラ変換の設定に関連する制御ビットについては、ADC が有効 (ADEN=1) であり、実行中のレギュラ変換がない (ADSTART = 0) 場合のみ、ソフトウェアでこれらを書き込むことができます。
- インジェクト変換の設定に関連する制御ビットについては、ADC が有効 (ADEN=1) であり、実行中のインジェクト変換がない (JADSTART = 0) 場合のみ、ソフトウェアでこれらを書き込むことができます。

ADC が有効であり、最終的に変換される場合、および ADC を無効にするための保留中のリクエストがない場合のみ (ADSTART = 1 または JADSTART = 1、かつ ADDIS = 0)、ソフトウェアは ADCx\_CR レジスタの制御ビット ADSTP または JADSTP を書き込むことができます。

ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもレジスタ ADCx\_JSQR を書き込むことができます。

**注:** これらの禁止された書き込みアクセスを防ぐハードウェア保護はありません。ADC の挙動は不明な状態になる場合があります。この状況を回復するには、ADC を無効にする必要があります (ADEN=0 と ADCx\_CR レジスタのすべてのビットをクリアします)。

## 15.3.11 チャネルの選択 (SQRx、JSQRx)

ADC ごとに最大 18 の多重化チャネルがあります。

- GPIO パッドからの最大 5 つの高速アナログ入力 (ADC\_IN1..5)
- GPIO パッドからの最大 10 個の低速アナログ入力 (ADC\_IN5..15) 製品によっては、すべてを GPIO パッドで使用できるわけではありません。
- ADC1 は、4 つの内部アナログ入力に接続されています。
  - ADC1\_IN15 =  $V_{REFOPAMP1}$  = オペアンプ 1 の基準電圧 (STM32F303xB/C および STM32F358xC)
  - ADC1\_IN16 =  $V_{TS}$  = 温度センサ
  - ADC1\_IN17 =  $V_{BAT}/2 = V_{BAT}$  チャネル
  - ADC1\_IN18 =  $V_{REFINT}$  = 内部基準電圧 (ADC2\_IN18、ADC3\_IN18、および ADC4\_IN18 にも接続)
- ADC2\_IN17 =  $V_{REFOPAMP2}$  = オペアンプ 2 の基準電圧
- ADC3\_IN17 =  $V_{REFOPAMP3}$  = オペアンプ 3 の基準電圧 (STM32F303xB/C/D/E および STM32F358C)
- ADC4\_IN17 =  $V_{REFOPAMP4}$  = オペアンプ 4 の基準電圧 (STM32F303xB/C/D/E および STM32F358C)

**警告：** ユーザは、同時に  $V_{REFINT}$  を変換できるのは 4 つの ADC のうちの 1 つのみであることを確認しておく必要があります (複数の ADC を使用して同時に  $V_{REFINT}$  を変換することは禁止されています)。

**注：** 内部アナログチャンネルのうちの 1 つを変換するには、最初に  $ADCx\_CCR$  レジスタのビット  $VREFEN$ 、 $TSEN$ 、または  $VBATEN$  をプログラムして、対応するアナログソースを有効化する必要があります。

変換は、レギュラとインジェクトの 2 つのグループに構成することができます。各グループは、任意のチャンネルに対して任意の順序で行うことができる一連の変換で構成されます。たとえば、次のような順で変換順を設定することができます： $ADC\_IN3$ 、 $ADC\_IN8$ 、 $ADC\_IN2$ 、 $ADC\_IN2$ 、 $ADC\_IN0$ 、 $ADC\_IN2$ 、 $ADC\_IN2$ 、 $ADC\_IN15$ 。

- レギュラグループは、最大 16 の変換で構成されます。レギュラチャンネルと変換シーケンス内での順序は、 $ADCx\_SQR$  レジスタで選択する必要があります。レギュラグループの合計変換数は、 $ADCx\_SQR1$  レジスタの  $L[3:0]$  ビットに書き込む必要があります。
- インジェクトグループは、最大 4 つの変換で構成されます。インジェクトチャンネルと変換シーケンス内での順序は、 $ADCx\_JSQR$  レジスタで選択する必要があります。インジェクトグループの合計変換数は、 $ADCx\_JSQR$  レジスタの  $L[1:0]$  ビットに書き込む必要があります。

レギュラ変換が実行される可能性がある場合は、 $ADCx\_SQR$  レジスタを変更しないでください。この場合、最初に  $ADSTP=1$  を書き込んで、ADC レギュラ変換を停止する必要があります (セクション 15.3.17 : 実行中の変換の停止 ( $ADSTP$ 、 $JADSTP$ ) を参照)。

インジェクト変換の実行中は、動作中に  $ADCx\_JSQR$  レジスタを変更することができます。セクション 15.3.21 : インジェクト変換のコンテキストのキューを参照してください。

## 15.3.12 チャンネル単位でプログラム可能なサンプリング時間 (SMPR1、SMPR2)

変換を開始する前に、ADC は測定する電圧ソースと ADC の内蔵サンプリングコンデンサの間の直接接続を確立する必要があります。このサンプリング時間は、入力電圧ソースが内蔵コンデンサを入力電圧レベルまでチャージできるだけの十分な長さが必要です。

各チャンネルは、 $ADCx\_SMPR1$  および  $ADCx\_SMPR2$  レジスタの  $SMP[2:0]$  ビットをプログラムすることで異なるサンプリング時間でサンプリングできます。したがって、次のサンプリング時間の値から選択することができます。

- $SMP = 000$  : 1.5 ADC クロックサイクル
- $SMP = 001$  : 2.5 ADC クロックサイクル
- $SMP = 010$  : 4.5 ADC クロックサイクル
- $SMP = 011$  : 7.5 ADC クロックサイクル
- $SMP = 100$  : 19.5 ADC クロックサイクル
- $SMP = 101$  : 61.5 ADC クロックサイクル
- $SMP = 110$  : 181.5 ADC クロックサイクル
- $SMP = 111$  : 601.5 ADC クロックサイクル

合計変換時間は、次のように計算されます。

$$T_{conv} = \text{サンプリング時間} + 12.5 \text{ ADC クロックサイクル}$$



例 :

$F_{ADC\_CLK} = 72 \text{ MHz}$ 、サンプリング時間 = 1.5 ADC クロックサイクル :

$T_{conv} = (1.5 + 12.5) \text{ ADC クロックサイクル} = 14 \text{ ADC クロックサイクル} = 0.194 \mu\text{s}$  (高速チャンネル)

ADC は、ステータスビット EOSMP をセットすることによって、サンプリングフェーズの終了を示します (レギュラ変換のみ)。

## 高速および低速チャンネルのサンプリング時間の制約

各チャンネルで、SMP[2:0] ビットはデータシートの ADC 特性セクションで指定された最小サンプリング時間を考慮してプログラムする必要があります。

### 15.3.13 シングル変換モード (CONT=0)

シングル変換モードでは、ADC はすべてのチャンネルを一度変換します。CONT ビットが 0 の時、次のいずれかによってこのモードが開始します。

- ADCx\_CR レジスタの ADSTART ビットをセットします (レギュラチャンネル)。
- ADCx\_CR レジスタの JADSTART ビットをセットします (インジェクトチャンネル)。
- 外部ハードウェアトリガイベント (レギュラまたはインジェクトチャンネル)

レギュラシーケンス内で、各変換の完了後、

- 変換データは 16 ビットの ADCx\_DR レジスタに格納されます。
- EOC (レギュラ変換の完了) フラグがセットされます。
- EOCIE ビットがセットされている場合、割り込みが生成されます。

インジェクトシーケンス内で、各変換の完了後、

- 変換データは 4 つの 16 ビット ADCx\_JDRy レジスタのうちの 1 つに格納されます。
- JEOP (インジェクト変換の完了) フラグがセットされます。
- JEOPIC ビットがセットされている場合、割り込みが生成されます。

レギュラシーケンスの完了後、

- EOS (レギュラシーケンスの完了) フラグがセットされます。
- EOSIE ビットがセットされている場合、割り込みが生成されます。

インジェクトシーケンスの完了後、

- JEOP (インジェクトシーケンスの完了) フラグがセットされます。
- JEOPIC ビットがセットされている場合、割り込みが生成されます。

次に、ADC は、新しい外部レギュラトリガまたは外部インジェクトトリガが発生するか、ADSTART または JADSTART ビットが再びセットされるまで停止します。

**注 :** 単一チャンネルを変換するには、長さが 1 のシーケンスをプログラムします。

### 15.3.14 連続変換モード (CONT=1)

このモードはレギュラチャンネルにのみ適用されます。

連続変換モードでは、ソフトウェアまたはハードウェアのレギュラトリガイベントが発生すると、ADC はチャンネルのレギュラ変換を一度すべて実行し、自動的に再起動して、シーケンスの各変換を連続的に実行します。CONT ビットが 1 の時、外部トリガまたは ADCx\_CR レジスタの ADSTART ビットをセットすることによって、このモードが開始されます。



レギュラシーケンス内で、各変換の完了後、

- 変換データは 16 ビットの ADCx\_DR レジスタに格納されます。
- EOC (変換完了) フラグがセットされます。
- EOCIE ビットがセットされている場合、割り込みが生成されます。

変換シーケンスの完了後、

- EOS (シーケンス完了) フラグがセットされます。
- EOSIE ビットがセットされている場合、割り込みが生成されます。

次に、新しいシーケンスをすぐに再開して、ADC は変換シーケンスを連続的に繰り返します。

**注：**

**単一チャンネルを変換するには、長さが1のシーケンスをプログラムします。**

**不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方をセットすることは禁じられています。**

**インジェクトチャンネルを連続して変換することはできません。唯一の例外は、連続モードに設定されたレギュラチャンネルの後にインジェクトチャンネルを自動的に変換するように設定されている (JAUTO ビットを使用) 場合です。自動インジェクションモードのセクションを参照してください。**

## 15.3.15 変換の開始 (ADSTART、JADSTART)

ソフトウェアは、ADSTART=1 をセットすることによって ADC レギュラ変換を開始します。

ADSTART がセットされると、変換は、

- EXTEN = 0x0 (ソフトウェアトリガ) の場合、すぐに開始します。
- EXTEN ≠ 0x0 の場合、選択されたレギュラハードウェアトリガの次のアクティブエッジで開始します。

ソフトウェアは、JADSTART = 1 をセットすることによって ADC インジェクト変換を開始します。

JADSTART がセットされると、変換は、

- JEXTEN = 0x0 (ソフトウェアトリガ) の場合、すぐに開始します。
- JEXTEN ≠ 0x0 の場合、選択されたインジェクトハードウェアトリガの次のアクティブエッジで開始します。

**注：**

**自動インジェクションモード (JAUTO=1) では、ADSTART ビットを使用してレギュラ変換を開始し、続いて自動インジェクト変換を行います (JADSTART はクリアされたままでなければなりません)。**

ADSTART および JADSTART では、ADC の動作が実行中かどうかを示す情報も提供しています。ADSTART=0 および JADSTART=0 が両方とも真であり、ADC がアイドルであることを示しているときには、ADC を再設定できます。

次のときに、ADSTART はハードウェアによってクリアされます。

- シングルモード時のソフトウェアのレギュラトリガ (CONT=0、EXTSEL=0x0)
  - レギュラ変換シーケンスの終了時に (EOS のアサート)、または DISCEN = 1 の場合のサブグループ処理の終了時
- すべての場合 (CONT=x、EXTSEL=x)
  - ソフトウェアによってアサートされた ADSTP 手順の実行後

**注：**

**連続モード (CONT=1) では、シーケンスは自動的に再起動されるので、EOS がアサートされても、ADSTART はハードウェアによってクリアされません。**

**シングルモードでハードウェアトリガが選択されたとき (CONT=0 かつ EXTSEL ≠ 0x00)、次のハードウェアトリガイベントで ADSTART の再リセットを必要としないソフトウェアを支援するために、**



**EOS がアサートされても、ADSTART はハードウェアによってクリアされません。これにより、さらなるハードウェアトリガの見落としを防ぐことができます。**

次のときに、JADSTART はハードウェアによってクリアされます。

- シングルモード時のソフトウェアのインジェクトトリガ (JEXTSEL=0x0)
  - インジェクト変換シーケンスの終了時に (JEOS のアサート)、または JDISCEN = 1 の場合のサブグループ処理の終了時
- すべての場合 (JEXTSEL=x)
  - ソフトウェアによってアサートされた JADSTP 手順の実行後

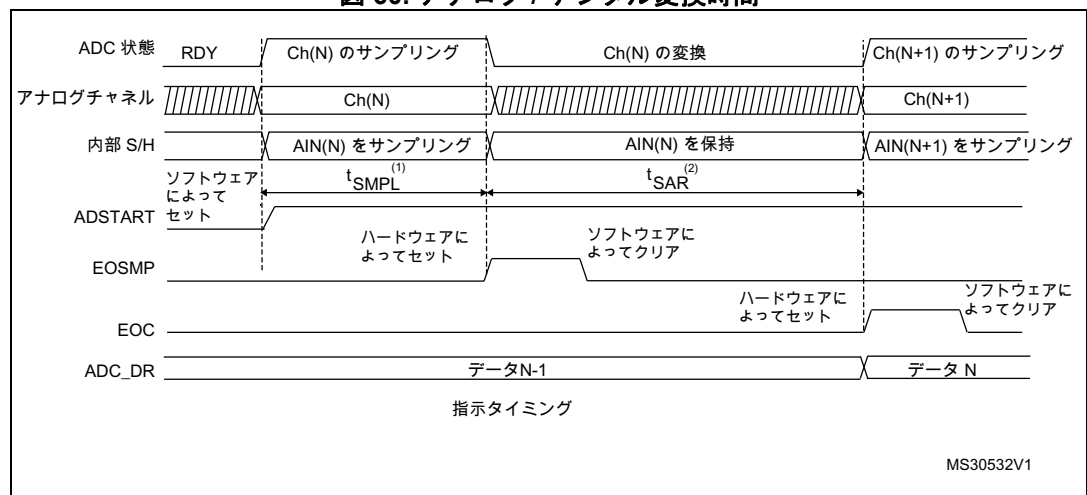
## 15.3.16 タイミング

変換の開始から変換の終了までの経過時間は、設定されたサンプリング時間に逐次比較時間 (データ分解能に依存) を加えた合計です。

$$T_{ADC} = T_{SMPL} + T_{SAR} = [ 1.5 |_{min} + 12.5 |_{12bit} ] \times T_{ADC\_CLK}$$

$$T_{ADC} = T_{SMPL} + T_{SAR} = 20.83 \text{ ns } |_{min} + 173.6 \text{ ns } |_{12bit} = 194.4 \text{ ns } ( F_{ADC\_CLK} = 72 \text{ MHz の場合 )$$

図 60. アナログ / デジタル変換時間



1.  $T_{SMPL}$  は SMP[2:0] に依存
2.  $T_{SAR}$  は RES[2:0] に依存

## 15.3.17 実行中の変換の停止 (ADSTP、JADSTP)

ソフトウェアは、ADSTP=1 をセットすることによって実行中のレギュラ変換を、また JADSTP=1 をセットすることによって実行中のインジェクト変換を停止することができます。

変換を停止すると、実行中の ADC 動作がリセットされます。これによって ADC 動作が再設定され (チャンネル選択やトリガの変更など)、新しい動作の準備ができます。

レギュラ変換の実行中に、インジェクト変換を停止することもできる点に注意してください (その逆も可能)。これにより、たとえば、レギュラ変換の実行中にインジェクト変換シーケンスやトリガを再設定することができます (その逆も可能)。

ADSTP ビットがソフトウェアによってセットされると、実行中のレギュラ変換はアボートされ、部分的な結果は破棄されます (ADCx\_DR レジスタは現在の変換では更新されません)。

JADSTP ビットがソフトウェアによってセットされると、実行中のインジェクト変換はアボートされ、部分的な結果は破棄されます (ADCx\_JDRy レジスタは現在の変換では更新されません)。スキャンシーケンスも中止され、リセットされます (ADC を再起動すると、新しいシーケンスが再開されることを意味します)。

この手順が完了すると、ADSTP/ADSTART ビット (レギュラ変換の場合)、または JADSTP/JADSTART ビット (インジェクト変換の場合) がハードウェアによってクリアされ、ソフトウェアは新しい変換を開始する前に ADSTART = 0 (または JADSTART = 0) になるまで待つ必要があります。

**注：** *自動インジェクションモード (JAUTO=1) では、ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアボートされます (JADSTP を使用してはいけません)。*

図 61. 実行中のレギュラ変換の停止

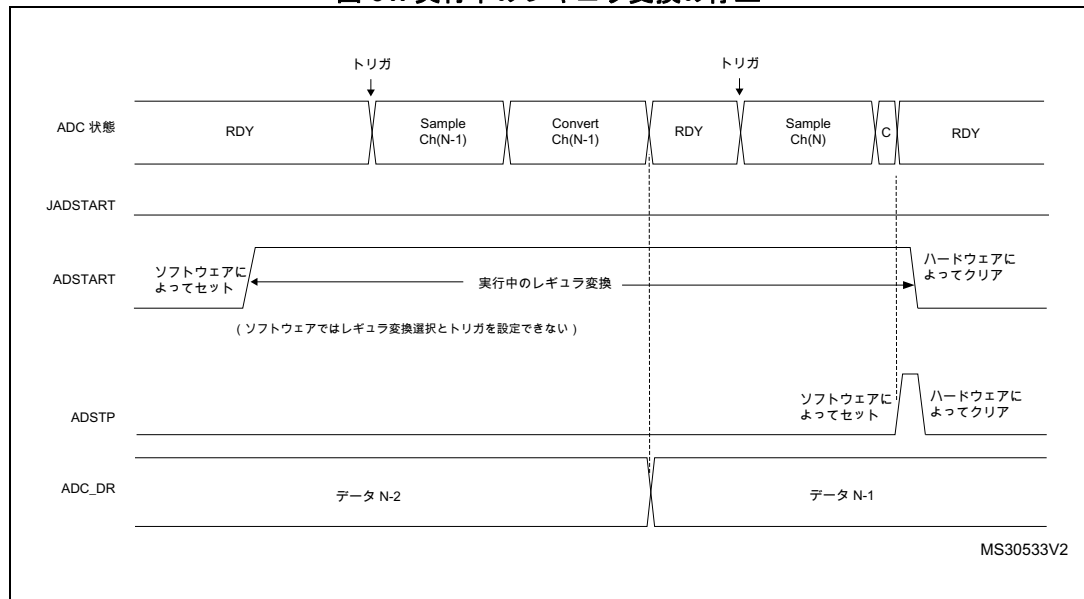


図 62. 実行中のレギュラ変換とインジェクト変換の停止



### 15.3.18 外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN、JEXTSEL、JEXTEN)

変換または変換シーケンスは、ソフトウェアによって、または外部イベント（タイマキャプチャや入力ピンなど）によってトリガできます。EXTEN[1:0] 制御ビット（レギュラ変換の場合）または JEXTEN[1:0] ビット（インジェクト変換の場合）が 0b00 以外の場合、外部イベントは選択した極性で変換をトリガすることができます。

レギュラトリガ選択は、一度ソフトウェアでビット ADSTART=1 をセットすると有効になります。インジェクトトリガ選択は、一度ソフトウェアでビット JADSTART=1 をセットすると有効になります。

変換中に発生したハードウェアトリガは無視されます。

- ビット ADSTART=0 の場合、発生したレギュラハードウェアトリガは無視されます。
- ビット JADSTART=0 の場合、発生したインジェクトハードウェアトリガは無視されます。

表 88 に EXTEN[1:0] と JEXTEN[1:0] の値とトリガ極性の対応を示します。

表 88. レギュラ外部トリガのトリガ極性の設定

EXTEN[1:0] / JEXTEN[1:0]	転送元
00	ハードウェアトリガ検出が無効になり、ソフトウェアトリガ検出が有効になります。
01	立ち上がりエッジで検出するハードウェアトリガ
10	立ち下がりエッジで検出するハードウェアトリガ
11	立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

- 注： レギュラトリガの極性は動作中に変更することはできません。
- 注： インジェクトトリガの極性は動作中に予想して変更できます。 [セクション 15.3.21：インジェクト変換のコンテキストのキュー](#)を参照してください。

EXTSEL[3:0] および JEXTSEL[3:0] 制御ビットでは、レギュラおよびインジェクトグループの変換をトリガできるイベントを 16 のイベントから選択します。

レギュラグループ変換は、インジェクトトリガによって中断することが可能です。

- 注： レギュラトリガの選択は動作中に変更することはできません。インジェクトトリガの選択は動作中に予想して変更できます。 [セクション 15.3.21：インジェクト変換のコンテキストのキュー \(331 ページ\)](#)を参照してください。

図 63 に示すように、各 ADC マスタは ADC スレーブと同じ入カトリガを共有しています。

図 63. ADC マスタと ADC スレーブ間でトリガを共有

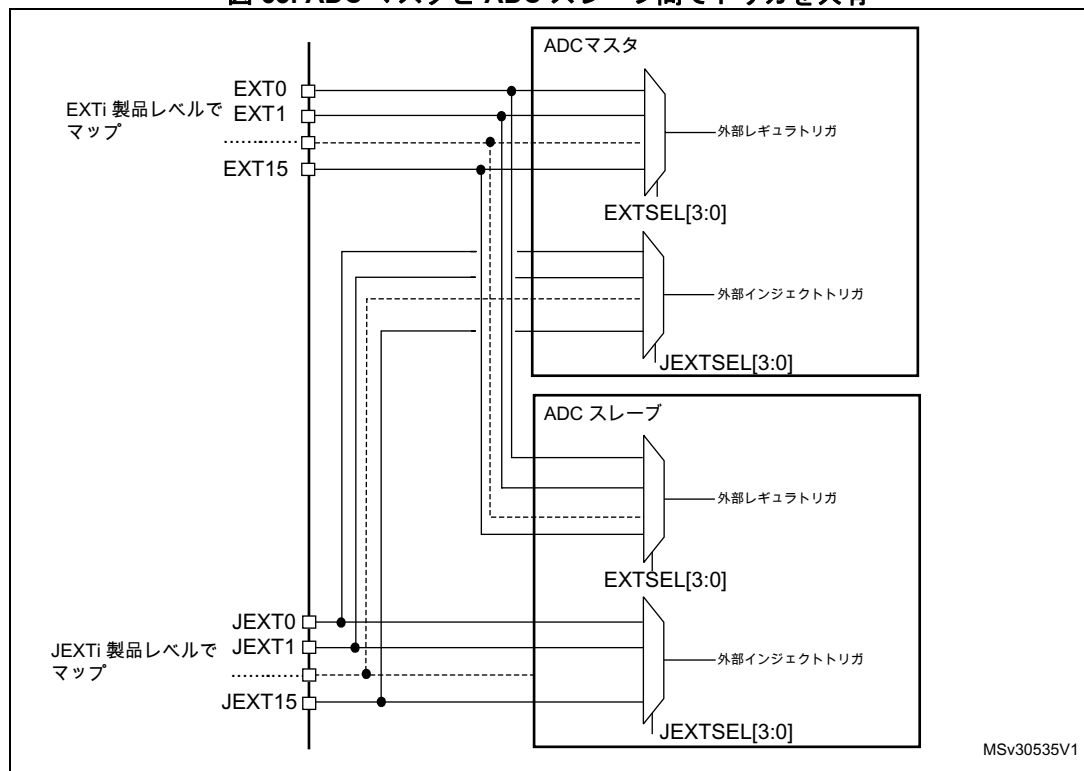


表 89 から 表 92 では、レギュラ変換とインジェクト変換に使用できる 4 つの ADC の全ての外部トリガを示します。

表 89. ADC1 (マスタ) および 2 (スレーブ) - レギュラチャネルの外部トリガ

名前	転送元	タイプ	EXTSEL[3:0]
EXT0	TIM1_CC1 イベント	オンチップタイマからの内部信号	0000
EXT1	TIM1_CC2 イベント	オンチップタイマからの内部信号	0001
EXT2	TIM1_CC3 イベントまたは TIM20_TRGO イベント <sup>(1)</sup>	オンチップタイマからの内部信号	0010
EXT3	TIM2_CC2 イベントまたは TIM20_TRGO2 <sup>(1)</sup>	オンチップタイマからの内部信号	0011

表 89. ADC1 (マスタ) および 2 (スレーブ) - レギュラチャネルの外部トリガ (続き)

名前	転送元	タイプ	EXTSEL[3:0]
EXT4	TIM3_TRGO イベント	オンチップタイマからの内部信号	0100
EXT5	TIM4_CC4 イベントまたは TIM20_CC1 <sup>(1)</sup>	オンチップタイマからの内部信号	0101
EXT6	EXTI ライン 11	外部ピン	0110
EXT7	TIM8_TRGO イベント	オンチップタイマからの内部信号	0111
EXT8	TIM8_TRGO2 イベント	オンチップタイマからの内部信号	1000
EXT9	TIM1_TRGO イベント	オンチップタイマからの内部信号	1001
EXT10	TIM1_TRGO2 イベント	オンチップタイマからの内部信号	1010
EXT11	TIM2_TRGO イベント	オンチップタイマからの内部信号	1011
EXT12	TIM4_TRGO イベント	オンチップタイマからの内部信号	1100
EXT13	TIM6_TRGO イベントまたは TIM20_CC2 <sup>(1)</sup>	オンチップタイマからの内部信号	1101
EXT14	TIM15_TRGO イベント	オンチップタイマからの内部信号	1110
EXT15	TIM3_CC4 イベントまたは TIM20_CC3 <sup>(1)</sup>	オンチップタイマからの内部信号	1111

1. STM32F303xD/E および STM32F398xE デバイスのみ

表 90. ADC1 および ADC2 - インジェクトチャネルの外部トリガ

名前	転送元	タイプ	JEXTSEL[3..0]
JEXT0	TIM1_TRGO イベント	オンチップタイマからの内部信号	0000
JEXT1	TIM1_CC4 イベント	オンチップタイマからの内部信号	0001
JEXT2	TIM2_TRGO イベント	オンチップタイマからの内部信号	0010
JEXT3	TIM2_CC1 イベントまたは TIM20_TRGO <sup>(1)</sup>	オンチップタイマからの内部信号	0011
JEXT4	TIM3_CC4 イベント	オンチップタイマからの内部信号	0100
JEXT5	TIM4_TRGO イベント	オンチップタイマからの内部信号	0101
JEXT6	EXTI ライン 15 または TIM20_GRG02 <sup>(1)</sup>	外部ピン	0110
JEXT7	TIM8_CC4 イベント	オンチップタイマからの内部信号	0111
JEXT8	TIM1_TRGO2 イベント	オンチップタイマからの内部信号	1000
JEXT9	TIM8_TRGO イベント	オンチップタイマからの内部信号	1001
JEXT10	TIM8_TRGO2 イベント	オンチップタイマからの内部信号	1010
JEXT11	TIM3_CC3 イベント	オンチップタイマからの内部信号	1011
JEXT12	TIM3_TRGO イベント	オンチップタイマからの内部信号	1100
JEXT13	TIM3_CC1 イベントまたは TIM20_CC4 <sup>(1)</sup>	オンチップタイマからの内部信号	1101
JEXT14	TIM6_TRGO イベント	オンチップタイマからの内部信号	1110
JEXT15	TIM15_TRGO イベント	オンチップタイマからの内部信号	1111

1. STM32F303xD/E および STM32F398xE デバイスのみ

表 91. ADC3 および ADC4 - レギュラチャネルの外部トリガ

名前	転送元	タイプ	EXTSEL[3..0]
EXT0	TIM3_CC1 イベント	オンチップタイマからの内部信号	0000
EXT1	TIM2_CC3 イベント	オンチップタイマからの内部信号	0001
EXT2	TIM1_CC3 イベント	オンチップタイマからの内部信号	0010
EXT3	TIM8_CC1 イベント	オンチップタイマからの内部信号	0011
EXT4	TIM8_TRGO イベント	オンチップタイマからの内部信号	0100
EXT5	EXTI ライン 2 または TIM20_TRGO <sup>(1)</sup>	外部ピン	0101
EXT6	TIM4_CC1 イベントまたは TIM20_TRGO2 <sup>(1)</sup>	オンチップタイマからの内部信号	0110
EXT7	TIM2_TRGO イベント	オンチップタイマからの内部信号	0111
EXT8	TIM8_TRGO2 イベント	オンチップタイマからの内部信号	1000
EXT9	TIM1_TRGO イベント	オンチップタイマからの内部信号	1001
EXT10	TIM1_TRGO2 イベント	オンチップタイマからの内部信号	1010
EXT11	TIM3_TRGO イベント	オンチップタイマからの内部信号	1011
EXT12	TIM4_TRGO イベント	オンチップタイマからの内部信号	1100
EXT13	TIM7_TRGO イベント	オンチップタイマからの内部信号	1101
EXT14	TIM15_TRGO イベント	オンチップタイマからの内部信号	1110
EXT15	TIM2_CC1 イベントまたは TIM20_CC1 <sup>(1)</sup>	オンチップタイマからの内部信号	1111

1. STM32F303xD/E および STM32F398xE デバイスのみ

表 92. ADC3 および ADC4 - インジェクトチャネルの外部トリガ

名前	転送元	タイプ	JEXTSEL[3..0]
JEXT0	TIM1_TRGO イベント	オンチップタイマからの内部信号	0000
JEXT1	TIM1_CC4 イベント	オンチップタイマからの内部信号	0001
JEXT2	TIM4_CC3 イベント	オンチップタイマからの内部信号	0010
JEXT3	TIM8_CC2 イベント	オンチップタイマからの内部信号	0011
JEXT4	TIM8_CC4 イベント	オンチップタイマからの内部信号	0100
JEXT5	TIM4_CC3 イベントまたは TIM20_TRGO <sup>(1)</sup>	オンチップタイマからの内部信号	0101
JEXT6	TIM4_CC4 イベント	オンチップタイマからの内部信号	0110
JEXT7	TIM4_TRGO イベント	オンチップタイマからの内部信号	0111
JEXT8	TIM1_TRGO2 イベント	オンチップタイマからの内部信号	1000
JEXT9	TIM8_TRGO イベント	オンチップタイマからの内部信号	1001
JEXT10	TIM8_TRGO2 イベント	オンチップタイマからの内部信号	1010
JEXT11	TIM1_CC3 イベントまたは TIM20_TRGO2 <sup>(1)</sup>	オンチップタイマからの内部信号	1011
JEXT12	TIM3_TRGO イベント	オンチップタイマからの内部信号	1100
JEXT13	TIM2_TRGO イベント	オンチップタイマからの内部信号	1101



表 92. ADC3 および ADC4 - インジェクトチャネルの外部トリガ (続き)

名前	転送元	タイプ	JEXTSEL[3..0]
JEXT14	TIM7_TRGO イベント	オンチップタイマからの内部信号	1110
JEXT15	TIM15_TRGO イベントまたは TIM20_CC2 <sup>(1)</sup>	オンチップタイマからの内部信号	1111

1. STM32F303xD/E および STM32F398xE デバイスのみ

**注:** 2 つのトリガソースを使用できる場合は、SYSCFG\_CFGR4 レジスタの対応するビットを使用して選択することができます。

## 15.3.19 インジェクトチャネルの管理

### トリガインジェクションモード

トリガインジェクションを使用するには、ADCx\_CFGR レジスタの JAUTO ビットがクリアされていなければなりません。

- 外部トリガによって、または、ADCx\_CR レジスタの ADSTART ビットをセットすることによって、レギュラチャネルグループの変換を開始します。
- レギュラチャネルグループの変換中に外部インジェクトトリガが発生した場合、または ADCx\_CR レジスタの JADSTART ビットがセットされた場合、現在の変換はリセットされ、インジェクトチャネルシーケンススイッチが起動されます(すべてのインジェクトチャネルが一度変換されます)。
- その後、レギュラチャネルグループのレギュラ変換が、最後に中断されたレギュラ変換から再開されます。
- インジェクト変換中にレギュライベントが発生した場合、インジェクト変換は中断されず、インジェクトシーケンスの終了時にレギュラシーケンスが実行されます。図 64 に対応するタイミング図を示します。

**注:** トリガインジェクションを使用する場合は、トリガイベントの間隔がインジェクションシーケンスより長くなるようにしなければなりません。たとえば、シーケンスの長さが 28 ADC クロックサイクルの場合 (1.5 クロック周期のサンプリング時間を持つ 2 つの変換)、トリガの最小間隔は 29 ADC クロックサイクルでなければなりません。

### 自動インジェクションモード

ADCx\_CFGR レジスタの JAUTO ビットがセットされている場合、レギュラグループチャネルの変換後、インジェクトグループチャネルが自動的に変換されます。これを使用して、ADCx\_SQR レジスタおよび ADCx\_JSQR レジスタにプログラミングされた最大 20 回の変換シーケンスを実行することができます。

このモードでは、レギュラ変換を開始し、続いてインジェクト変換を開始するために ADCx\_CR レジスタの ADSTART ビットをセットする必要があります (JADSTART はクリアされたままでなければなりません)。ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアポートされません (JADSTP ビットを使用することはできません)。

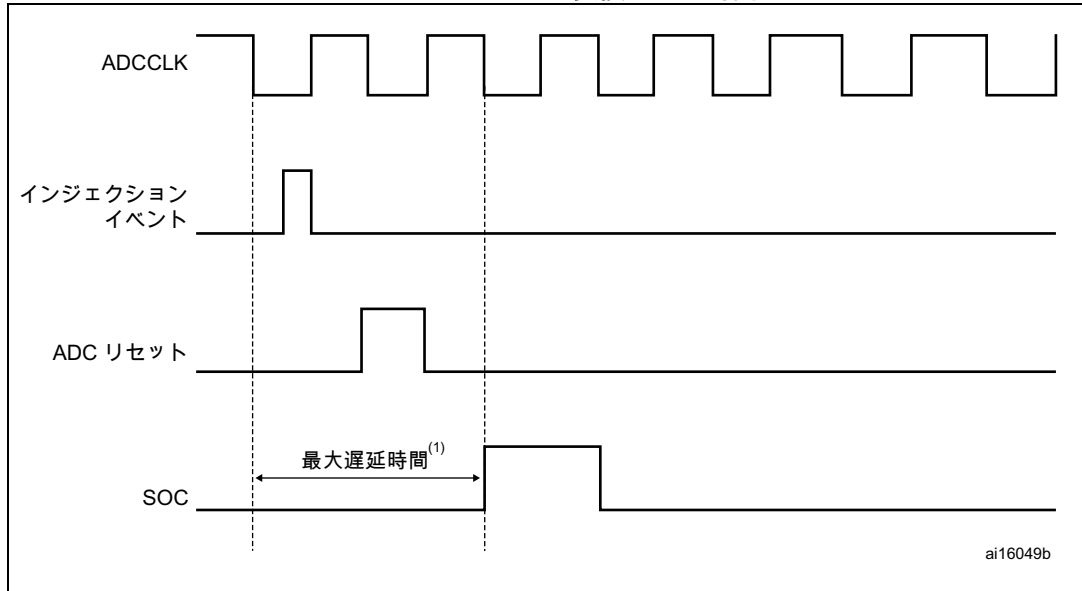
このモードでは、インジェクトチャネルの外部トリガは無効でなければなりません。

JAUTO ビットに加えて CONT ビットもセットされている場合、レギュラチャネルに続いてインジェクトチャネルが連続して変換されます。



注： 自動インジェクトモードと不連続モードを同時に使用することはできません。  
 JAUTO モードで、レギュラシーケンスのデータをエクスポートする際に DMA を使用する場合は、サーキュラモードでプログラムする必要があります (DMA\_CCRx レジスタの CIRC ビットをセット)。CIRC ビットがリセットされると (シングルショットモード)、DMA 転送完了イベントによって JAUTO シーケンスが停止されます。

図 64. インジェクト変換の遅延時間



1. 最大遅延時間の値は、STM32F3xx のデータシートの電気特性に記載されています。

## 15.3.20 不連続モード (DISCEN、DISCNUM、JDISCEN)

### レギュラグループモード

このモードは、ADCx\_CFGR レジスタの DISCEN ビットをセットすることによって有効になります。

これを使用して、ADCx\_SQR レジスタで選択された変換シーケンスの一部である短い n 回 ( $n \leq 8$ ) の変換シーケンス (サブグループ) を変換できます。n の値は、ADCx\_CFGR レジスタの DISCNUM[2:0] ビットに書き込むことによって指定されます。

外部トリガが発生すると、ADCx\_SQR レジスタで選択された次の n 回の変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADCx\_SQR1 レジスタの L[3:0] ビットによって定義されます。

例 :

- DISCEN=1、n=3、変換されるチャネル = 1、2、3、6、7、8、9、10、11
  - 1 番目のトリガ : 変換されるチャネルは 1、2、3 です (各変換で EOC イベントが生成されます)。
  - 2 番目のトリガ : 変換されるチャネルは 6、7、8 です (各変換で EOC イベントが生成されます)。
  - 3 番目のトリガ : 変換されるチャネルは 9、10、11 です (各変換で EOC イベントが生成されます)。EOS イベントはチャネル 11 の変換後に生成されます。
  - 4 番目のトリガ : 変換されるチャネルは 1、2、3 です (各変換で EOC イベントが生成されます)。
  - ...
- DISCEN=0、変換されるチャネル = 1、2、3、6、7、8、9、10、11
  - 1 番目のトリガ : シーケンス全体、すなわち、チャネル 1、次いで 2、3、6、7、8、9、10、および 11 が変換されます。各変換後に EOC イベントが生成され、最後の交換後には EOS イベントも生成されます。
  - 後続のすべてのトリガイイベントによって、シーケンス全体が再起動されます。

**注 :** レギュラグループが不連続モードで変換される際には、ロールオーバーは発生しません (シーケンスの最後のサブグループの変換は n 回以下です)。

すべてのサブグループが変換されると、次のトリガでは、最初のサブグループの変換が開始されます。上の例では、4 番目のトリガで最初のサブグループのチャネル 1、2、および 3 が再変換されます。

不連続モードと連続モードの両方を有効にすることはできません。この場合 (DISCEN=1、CONT=1 の場合)、ADC は連続モードが無効化されたかのように挙動します。

## インジェクトグループモード

このモードは、ADCx\_CFGR レジスタの JDISCEN ビットをセットすることによって有効になります。ADCx\_JSQR レジスタで選択されたシーケンスを、外部インジェクトトリガイイベント後に 1 チャネルずつ変換します。これは、「n」が 1 に固定されたレギュラチャネルの不連続モードに相当します。

外部トリガが発生すると、ADCx\_JSQR レジスタで選択された次のチャネル変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADCx\_JSQR レジスタの JL[1:0]ビットによって定義されます。

例 :

- JDISCEN=1、変換されるチャネル = 1、2、3
  - 1 番目のトリガ : チャネル 1 が変換されます (JEOC イベントが生成されます)。
  - 2 番目のトリガ : チャネル 2 が変換されます (JEOC イベントが生成されます)。
  - 3 番目のトリガ : チャネル 3 が変換され、JEOC イベントと JEOS イベントが生成されます。
  - ...

**注 :** すべてのインジェクトチャネルが変換されると、次のトリガでは、最初のインジェクトチャネルの変換が開始されます。上の例では、4 番目のトリガで最初のインジェクトチャネル 1 が再変換されます。

自動インジェクトモードと不連続モードの両方を同時に使用することはできません。JAUTO がセットされている場合には、DISCEN ビットと JDISCEN ビットはクリアされている必要があります。

## 15.3.21 インジェクト変換のコンテキストのキュー

コンテキストのキューは、次のインジェクト変換シーケンスに最大 2 つのコンテキストを予測して実装されます。

このコンテキストの構成は次のとおりです。

- インジェクトトリガの設定 (ADCx\_JSQR レジスタの JEXTEN[1:0] ビット および JEXTSEL[3:0] ビット)
- インジェクトシーケンスの定義 (ADCx\_JSQR レジスタの JSQx[4:0] ビット および JL[1:0] ビット)

コンテキストのすべてのパラメータは 1 つの ADCx\_JSQR レジスタ 内に定義され、このレジスタには最大 2 セットのパラメータのバッファ化を可能にする 2 つのバッファのキューが実装されます。

- JSQR レジスタは、インジェクト変換の実行中など、いつでも書き込むことができます。
- JSQR レジスタに書き込まれる各データは、コンテキストのキューに格納されます。
- 最初に、キューは空であり、JSQR レジスタへの最初の書き込みアクセスによってコンテキストがすぐに変更され、ADC ではインジェクトトリガを受信する準備ができます。
- 一度インジェクトシーケンスが完了すると、キューは消費され、キューに格納された次の JSQR パラメータに従ってコンテキストが変化します。この新しいコンテキストは、次のインジェクト変換シーケンスに適用されます。
- キューのオーバーフローは、キューがフル状態のときに JSQR レジスタに書き込むと発生します。このオーバーフローは、フラグ JQOVF のアサーションによって示されます。オーバーフローが発生すると、オーバーフローを作成した JSQR レジスタの書き込みアクセスは無視され、コンテキストのキューは変更されません。JQOVFIE ビットがセットされている場合は、割り込みを生成することができます。
- キューが空になると、ADCx\_CFGR レジスタの制御ビット JQM の値に応じて、2 つの挙動が考えられます。
  - JQM=0 の場合、キューは ADC を有効化した直後に空になりますが、実行動作中に空になることはありません。キューは常に最後のアクティブコンテキストを維持し、さらに有効なインジェクトシーケンスの開始を最後のアクティブコンテキストに応じて処理します。
  - JQM=1 の場合、キューはインジェクトシーケンス終了後、またはキューのフラッシュ時に空になります。このとき、キュー内にコンテキストはなく、インジェクトソフトウェアとハードウェアトリガは無効になります。そのため、ソフトウェアが新しいインジェクトコンテキストを JSQR レジスタに再度書き込むまで、さらなるハードウェアまたはソフトウェアインジェクトトリガは無視されます。
- JSQR レジスタを読み出すと、その時点でアクティブな現在の JSQR コンテキストが返されます。JSQR コンテキストが空である場合、JSQR は 0x0000 として読み出されます。
- キューは、JADSTP=1 をセットしてインジェクト変換を停止した場合、または ADDIS=1 をセットして ADC を無効化した場合にフラッシュされます。
  - JQM=0 の場合、キューは最後のアクティブコンテキストによって維持されます。
  - JQM=1 の場合、キューは空になり、トリガは無視されます。

**注：** *不連続モードで設定された場合 (ビット JDISCEN=1)、コンテキストを変更し、キューを消費するのはインジェクトシーケンスの最後のトリガのみです。1 番目のトリガではキューを消費するだけです*

が、次の不連続モードの例に示すように、その他のトリガは引き続き有効なトリガとなります (両方のコンテキストにおいて長さは3)。

- 1 番目のトリガ、不連続。シーケンス 1 : コンテキスト 1 を消費、1 番目の変換を実行
- 2 番目のトリガ、不連続。シーケンス 1 : 2 番目の変換
- 3 番目のトリガ、不連続。シーケンス 1 : 3 番目の変換
- 4 番目のトリガ、不連続。シーケンス 2 : コンテキスト 2 を消費、1 番目の変換を実行
- 5 番目のトリガ、不連続。シーケンス 2 : 2 番目の変換
- 6 番目のトリガ、不連続。シーケンス 2 : 3 番目の変換

### トリガまたはシーケンスのコンテキストを変更する場合の挙動

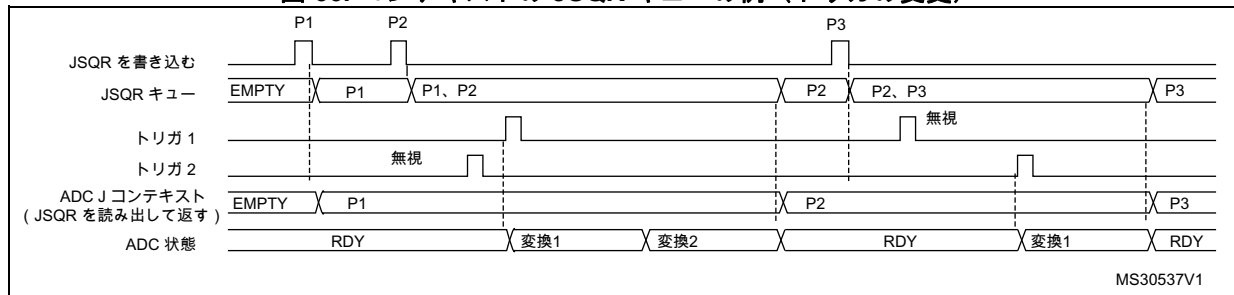
図 65 と 図 66 に、シーケンスまたはトリガを変更する際のコンテキストのキューの挙動を示します。

図 65. コンテキストの JSQR キューの例 (シーケンスの変更)



1. パラメータ :
- P1 : 3 つの変換のシーケンス、ハードウェアトリガ 1
  - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P3 : 4 つの変換のシーケンス、ハードウェアトリガ 1

図 66. コンテキストの JSQR キューの例 (トリガの変更)

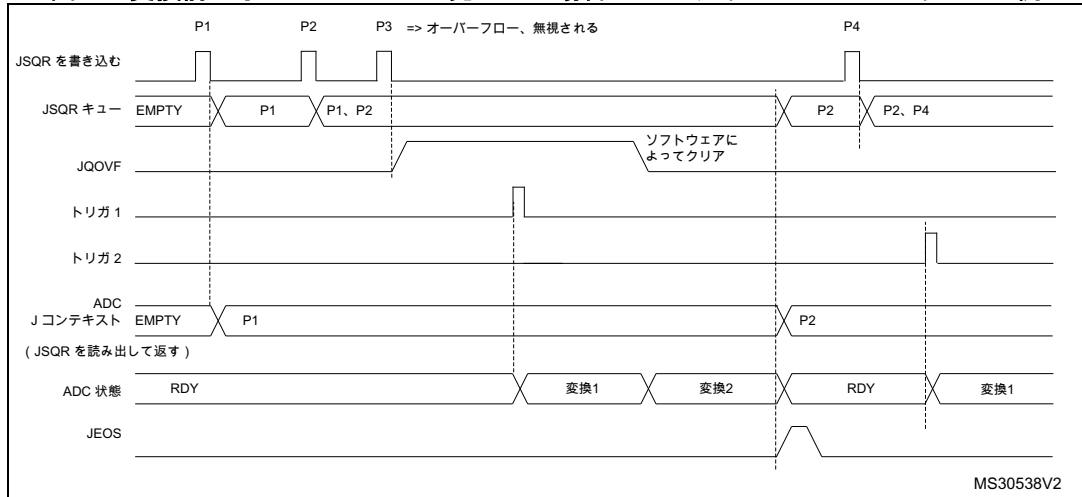


1. パラメータ :
- P1 : 2 つの変換のシーケンス、ハードウェアトリガ 1
  - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 2
  - P3 : 4 つの変換のシーケンス、ハードウェアトリガ 1

## コンテキストのキュー：キューのオーバーフローが発生した場合の挙動

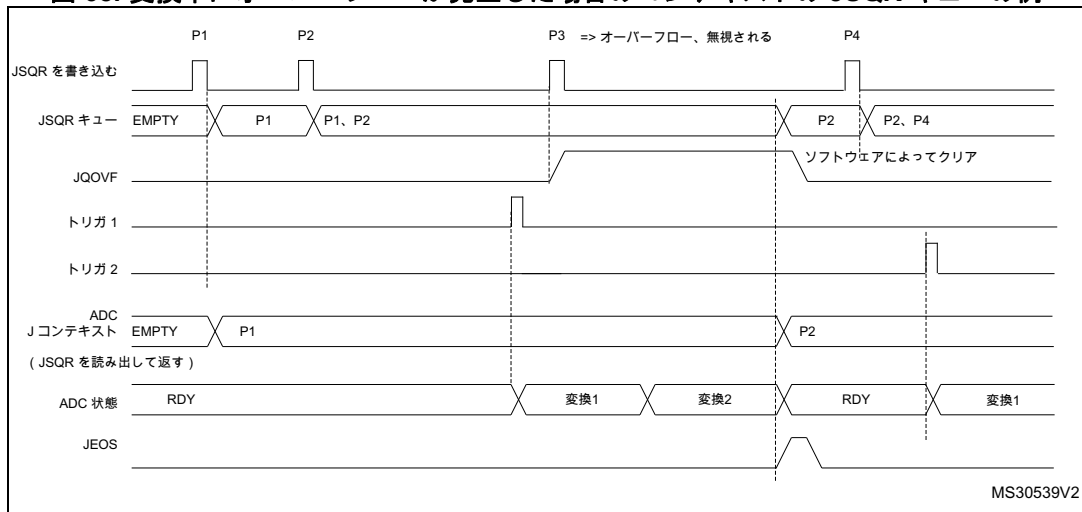
図 67 と 図 68 に、変換前または変換中にオーバーフローが発生した場合のコンテキストのキューの挙動を示します。

図 67. 変換前にオーバーフローが発生した場合のコンテキストの JSQR キューの例



1. パラメータ：
- P1：2 つの変換のシーケンス、ハードウェアトリガ 1
  - P2：1 つの変換のシーケンス、ハードウェアトリガ 2
  - P3：3 つの変換のシーケンス、ハードウェアトリガ 1
  - P4：4 つの変換のシーケンス、ハードウェアトリガ 1

図 68. 変換中にオーバーフローが発生した場合のコンテキストの JSQR キューの例



1. パラメータ：
- P1：2 つの変換のシーケンス、ハードウェアトリガ 1
  - P2：1 つの変換のシーケンス、ハードウェアトリガ 2
  - P3：3 つの変換のシーケンス、ハードウェアトリガ 1
  - P4：4 つの変換のシーケンス、ハードウェアトリガ 1

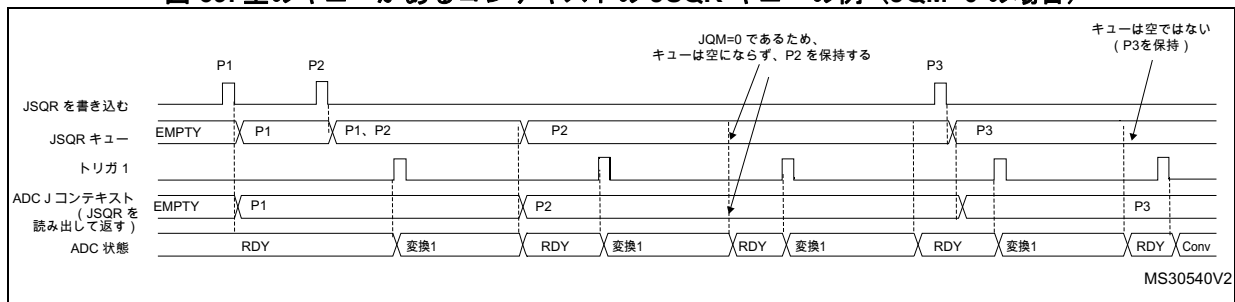
次のようにキューのオーバーフローを管理することを推奨します。

- JSQR レジスタへの各 P コンテキストの書き込み後、JQOVF フラグ は書き込みが無視されたかどうかを示します (割り込みを生成することができます)。
- 一度前のコンテキスト P2 のフラグ JEOS がセットされた場合のみ、3 番目のコンテキスト (P3) を書き込んでキューのオーバーフローを避けます。これにより、前のコンテキストが消費され、キューがフル状態ではないことを確認できます。

### コンテキストのキュー：キューが空になった場合の挙動

図 69 と 図 70 に、JQM=0 または 1 の場合にキューが空になる際のコンテキストのキューの挙動を示します。

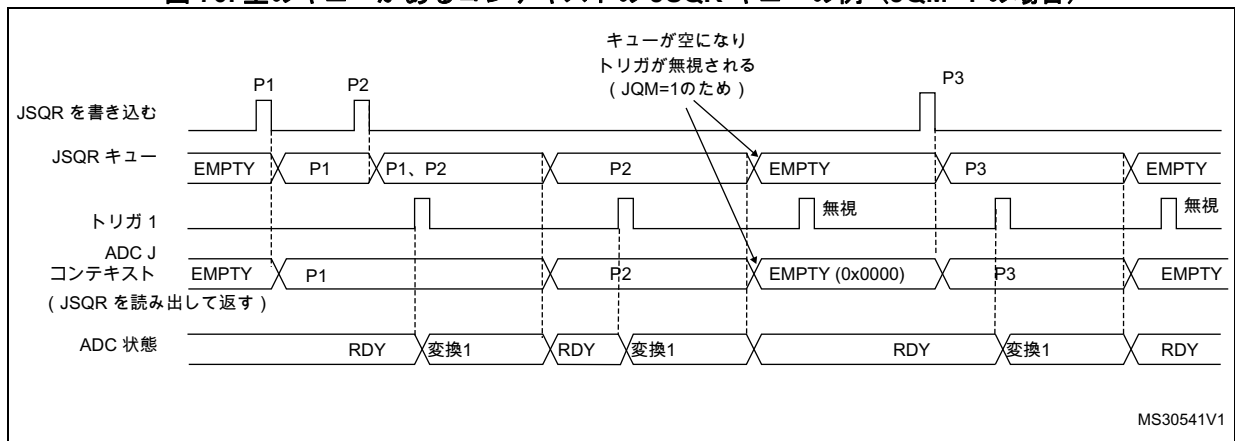
図 69. 空のキューがあるコンテキストの JSQR キューの例 (JQM=0 の場合)



1. パラメータ :  
 P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

**注 :** P3 を書き込む際、コンテキストはすぐに変更されます。ただし、内部での再同期のため、遅延が発生します。また、P3 を書き込んだ後または前にトリガが発生すると、コンテキスト P2 を考慮して変換が発生する可能性があります。この状況を避けるために、ユーザはすぐに適用される新しいコンテキストを書き込む際に ADC トリガが発生していないことを確認する必要があります。

図 70. 空のキューがあるコンテキストの JSQR キューの例 (JQM=1 の場合)

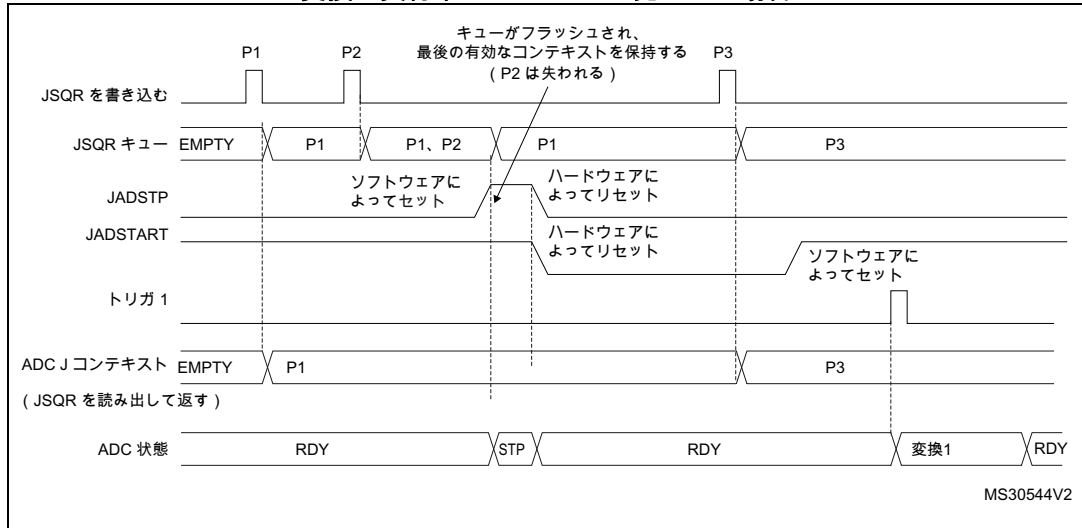


1. パラメータ :  
 P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

## コンテキストのキューのフラッシュ

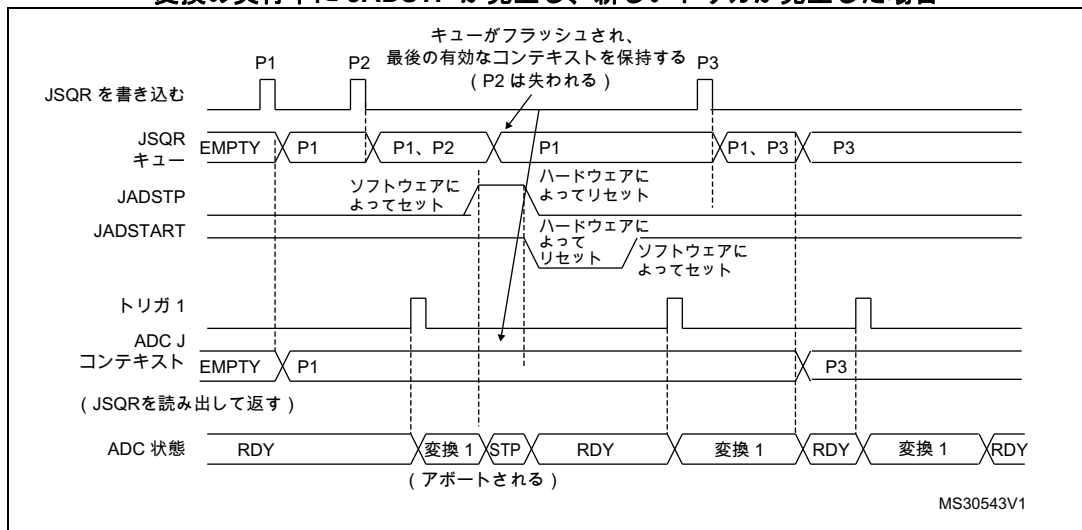
次の図に、キューがフラッシュされた場合のさまざまな状況におけるコンテキストのキューの挙動を示します。

**図 71. JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0) 変換の実行中に JADSTP が発生した場合**



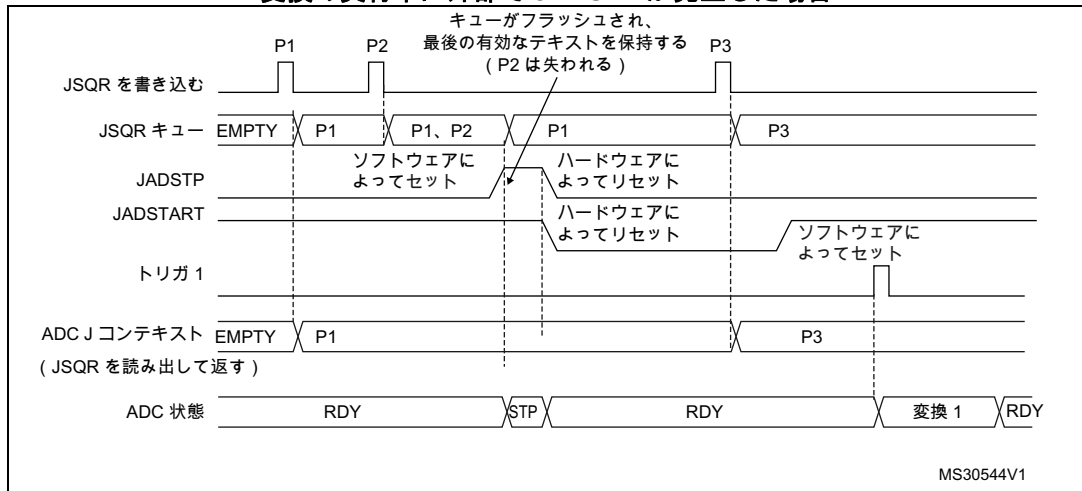
- パラメータ :  
 P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

**図 72. JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0) 変換の実行中に JADSTP が発生し、新しいトリガが発生した場合**



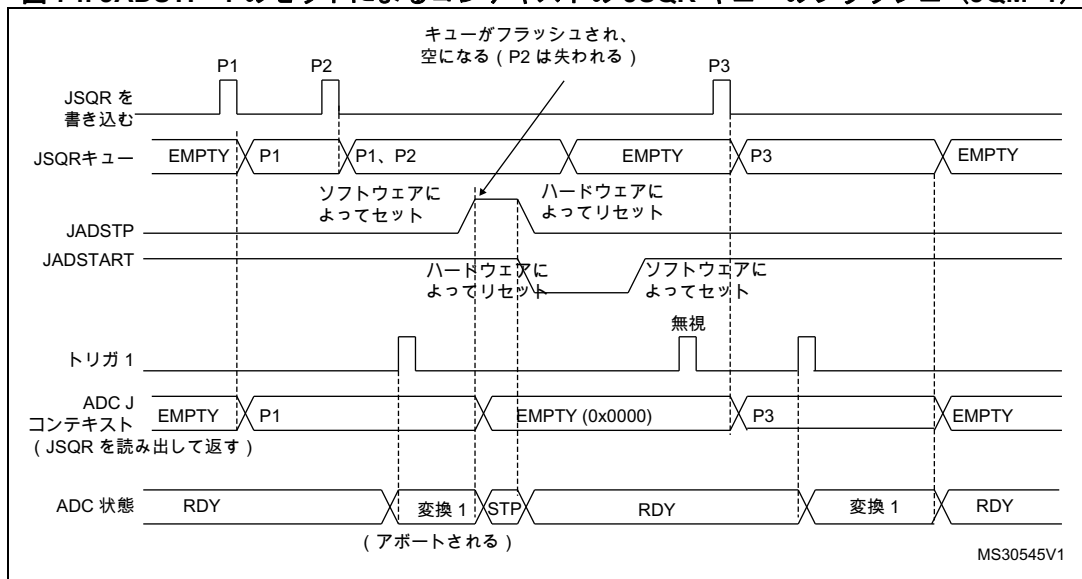
- パラメータ :  
 P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1  
 P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

図 73. JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0)  
変換の実行中に外部で JADSTP が発生した場合



1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

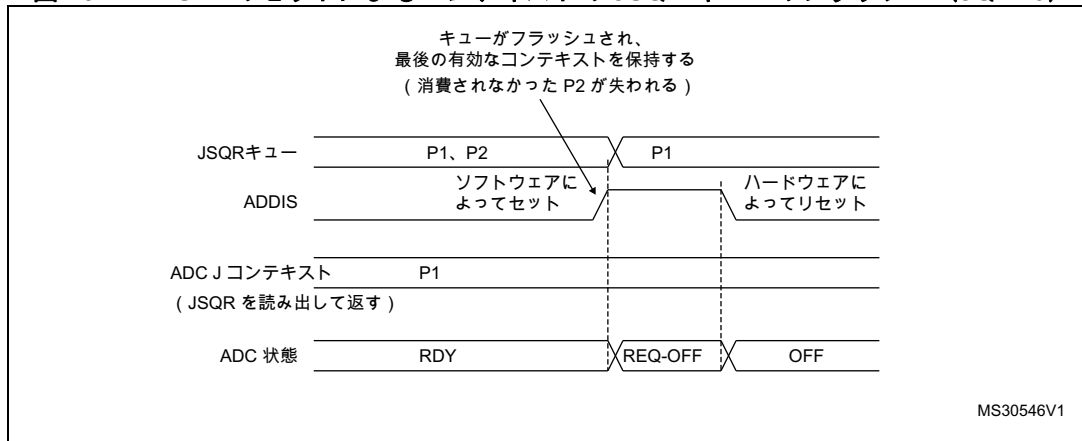
図 74. JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=1)



1. パラメータ :
- P1 : 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P2 : 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P3 : 1 つの変換のシーケンス、ハードウェアトリガ 1

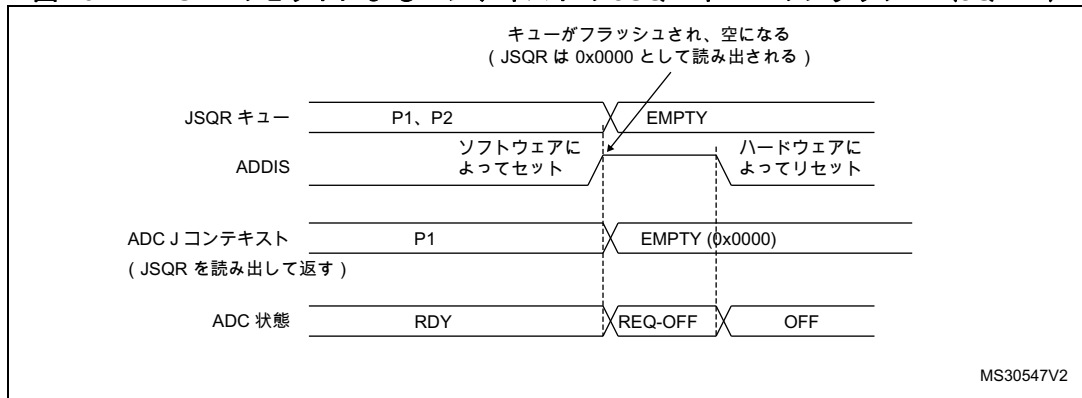


図 75. ADDIS=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0)



1. パラメータ:
- P1: 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P2: 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P3: 1 つの変換のシーケンス、ハードウェアトリガ 1

図 76. ADDIS=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=1)



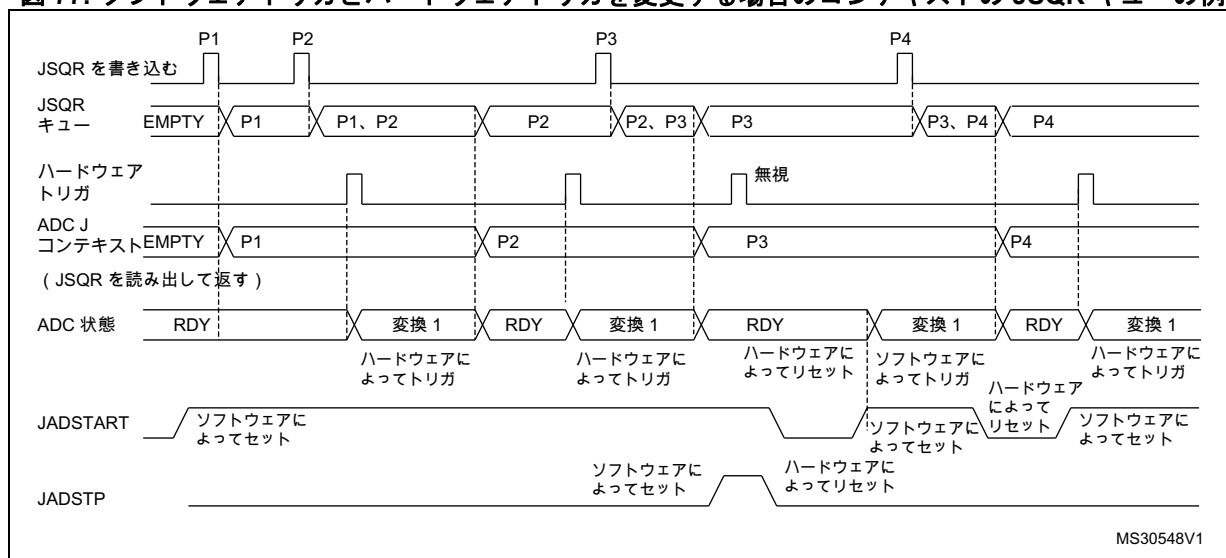
1. パラメータ:
- P1: 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P2: 1 つの変換のシーケンス、ハードウェアトリガ 1
  - P3: 1 つの変換のシーケンス、ハードウェアトリガ 1

## ハードウェアからソフトウェア (またはその逆) インジェクトトリガへのコンテキストの変更

ハードウェアトリガからソフトウェアインジェクトトリガにコンテキストを変更する場合、最後のハードウェアトリガ変換後に JADSTP=1 をセットして、インジェクト変換を停止する必要があります。これは、ソフトウェアトリガを再度有効にするために必要です (ソフトウェアインジェクト変換を開始するには JADSTART の立ち上がりエッジが必要です)。図 77 を参照してください。

ソフトウェアトリガからハードウェアインジェクトトリガにコンテキストを変更する場合、最後のソフトウェアトリガ後に JADSTART=1 をセットして、ハードウェアトリガを有効にする必要があります。図 77 を参照してください。

図 77. ソフトウェアトリガとハードウェアトリガを変更する場合のコンテキストの JSQR キューの例



- パラメータ:
  - P1: 1つの変換のシーケンス、ハードウェアトリガ (JEXTEN ≠ 0x0)
  - P2: 1つの変換のシーケンス、ハードウェアトリガ (JEXTEN ≠ 0x0)
  - P3: 1つの変換のシーケンス、ソフトウェアトリガ (JEXTEN = 0x0)
  - P4: 1つの変換のシーケンス、ハードウェアトリガ (JEXTEN ≠ 0x0)

### 15.3.22 プログラム可能な分解能 (RES) - 高速変換モード

ADC の分解能を下げることによって、高速変換を行うことができます。

分解能は、制御ビット RES[1:0] をプログラムすることによって、12、10、8、または 6 ビットに設定できます。図 82、図 83、図 84、および 図 85 に、分解能とデータの整列に対する変換結果のフォーマットを示します。

分解能を下げることによって変換時間を高速にでき、高いデータ精度を必要としないアプリケーションに有効です。表 93 のように、これによって逐次比較ステップの間に変換時間は短縮されます。

表 93.  $T_{SAR}$  タイミングは分解能に依存

RES (ビット)	$T_{SAR}$ (ADC クロック サイクル)	$T_{SAR}$ (ns) ( $F_{ADC}=72$ MHz 時)	$T_{ADC}$ (ADC クロック サイクル) (サンプリング時間 = 1.5 ADC クロック サイクル)	$T_{ADC}$ (ns) ( $F_{ADC}=72$ MHz 時)
12	12.5 ADC クロック サイクル	173.6 ns	14 ADC クロック サイクル	194.4 ns
10	10.5 ADC クロック サイクル	145.8 ns	12 ADC クロック サイクル	166.7 ns
8	8.5 ADC クロック サイクル	118.0 ns	10 ADC クロック サイクル	138.9 ns
6	6.5 ADC クロック サイクル	90.3 ns	8 ADC クロック サイクル	111.1 ns

### 15.3.23 変換の終了、サンプリングフェーズの終了 (EOC、JEOC、EOSMP)

ADC は各レギュラ変換の終了 (EOC) イベントと各インジェクト変換の終了 (JEOC) イベントをアプリケーションに通知します。

ADC は、新しいレギュラ変換データが ADCx\_DR レジスタで使用可能になるとすぐに EOC フラグをセットします。EOCIE ビットがセットされている場合は、割り込みを生成することができます。EOC フラグは、ソフトウェアによって 1 を書き込むことによって、または ADCx\_DR を読み出すことによってクリアされます。

ADC は、新しいインジェクト変換データが ADCx\_JDRy レジスタの 1 つで使用可能になるとすぐに JEOC フラグをセットします。JEOCIE ビットがセットされている場合は、割り込みを生成することができます。JEOC フラグは、ソフトウェアによって 1 を書き込むことによって、または対応する ADCx\_JDRy レジスタを読み出すことによってクリアされます。

さらに、ADC は、ステータスビット EOSMP をセットすることによって、サンプリングフェーズの終了を示します (レギュラ変換のみ)。EOSMP フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。EOSMPIE ビットがセットされている場合は、割り込みを生成することができます。

### 15.3.24 変換シーケンスの終了 (EOS、JEOS)

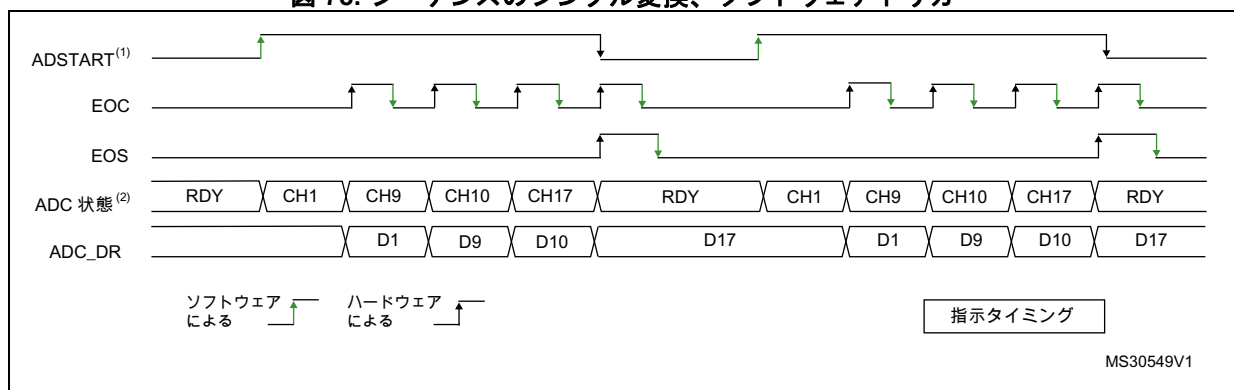
ADC は各レギュラシーケンスの終了 (EOC) と各インジェクトシーケンスの終了 (JEOC) イベントをアプリケーションに通知します。

ADC は、レギュラ変換シーケンスの最後のデータが ADCx\_DR レジスタで使用可能になるとすぐに、EOS フラグをセットします。EOSIE ビットがセットされている場合は、割り込みを生成することができます。EOS フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

ADC は、インジェクト変換シーケンスの最後のデータが完了するとすぐに、JEOS フラグをセットします。JEOSIE ビットがセットされている場合は、割り込みを生成することができます。JEOS フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

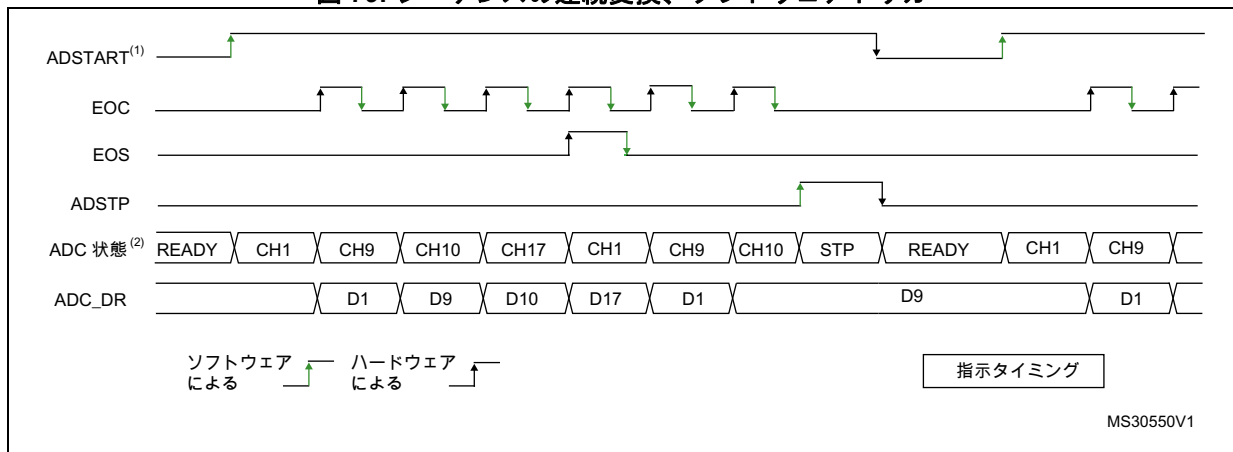
### 15.3.25 タイミング図の例 (シングル / 連続モード、ハードウェア / ソフトウェアトリガ)

図 78. シーケンスのシングル変換、ソフトウェアトリガ



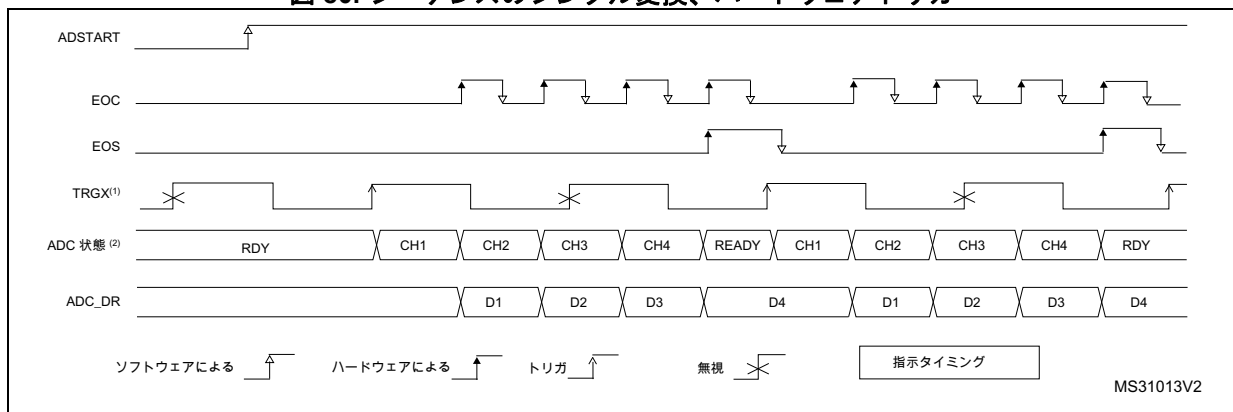
- EXTEN=0x0、CONT=0
- 選択されたチャネル = 1、9、10、17 ; AUTDLY=0

図 79. シーケンスの連続変換、ソフトウェアトリガ



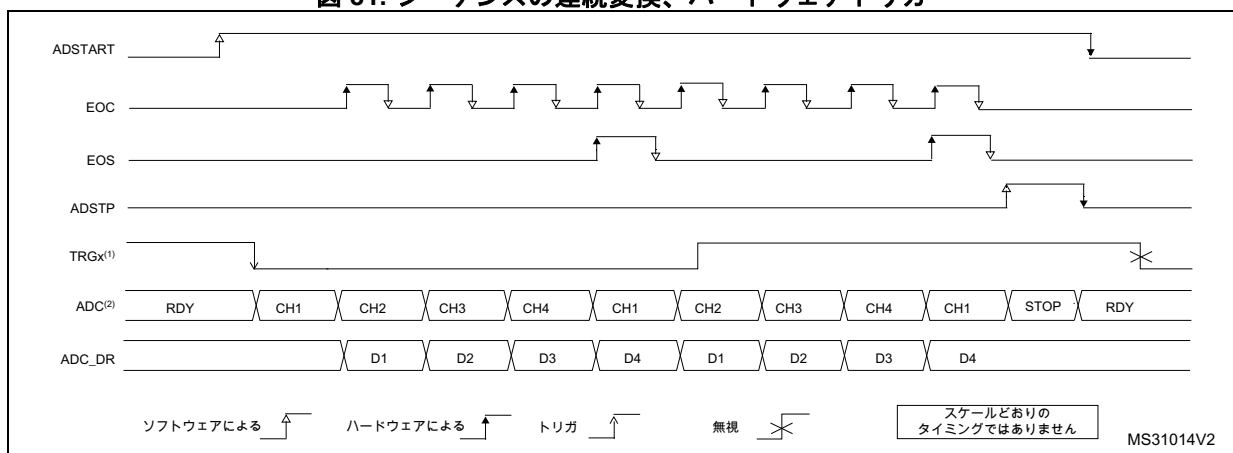
- EXTEN=0x0、CONT=1
- 選択されたチャンネル = 1、9、10、17 ; AUTDLY=0

図 80. シーケンスのシングル変換、ハードウェアトリガ



- TRGX (オーバー周波数) がトリガソースとして選択されます (EXTEN = 01、CONT = 0)。
- 選択されたチャンネル = 1、2、3、4 ; AUTDLY=0。

図 81. シーケンスの連続変換、ハードウェアトリガ



- TRGX がトリガソースとして選択されます (EXTEN = 10、CONT = 1)。
- 選択されたチャンネル = 1、2、3、4 ; AUTDLY=0。

## 15.3.26 データ管理

### データレジスタ、データの配置およびデータオフセット (ADCx\_DR、OFFSETy、OFFSETy\_CH、ALIGN)

#### データおよび配置

各レギュラ変換チャンネルの終了時 (EOC イベントの発生時)、変換データの結果は 16 ビット幅の ADCx\_DR データレジスタに格納されます。

各インジェクト変換チャンネルの終了時 (JEOC イベントの発生時)、変換データの結果は対応する 16 ビット幅の ADCx\_JDRy データレジスタに格納されます。

ADCx\_CFGR レジスタの ALIGN ビットは、変換後に格納されるデータの配置を選択します。データは、[図 82](#)、[図 83](#)、[図 84](#)、および [図 85](#) に示すように、右詰めまたは左詰めに配置できます。

特殊なケース：左詰めの場合、分解能が 6 ビットに設定されている場合を除き、データはハーフワード単位に配置されます。この場合、[図 84](#) および [図 85](#) に示すように、データはバイト単位に配置されます。

#### オフセット

オフセット y (y=1、2、3、4) は、ADCx\_OFRy レジスタにビット OFFSETy\_EN=1 をセットすることでチャンネルに適用されます。オフセットを適用するチャンネルは、ADCx\_OFRy レジスタの OFFSETy\_CH[4:0] ビットにプログラムされます。この場合、変換された値は OFFSETy[11:0] ビットに書き込まれたユーザ定義のオフセットによって減算されます。結果は負の値になることがあり、そのため読み出しデータは符号付きで、SEXT ビットは拡張符号値を表します。

[表 96](#) に、アナログウォッチドッグ 1 で可能なすべての分解能での比較方法を示します。

表 94. オフセット計算対データ分解能

分解能 (ビット RES[1:0])	元の変換データと オフセット間での減算：		結果	コメント
	元の変換データ、 左詰め	オフセット		
00：12 ビット	DATA[11:0]	OFFSET[11:0]	符号付き 12 ビットデータ	-
01：10 ビット	DATA[11:2],00	OFFSET[11:0]	符号付き 10 ビットデータ	ユーザは OFFSET[1:0] を "00" に設定する 必要があります。
10：8 ビット	DATA[11:4],0000	OFFSET[11:0]	符号付き 8 ビットデータ	ユーザは OFFSET[3:0] を "0000" に設定 する必要があります。
11：6 ビット	DATA[11:6],0000 00	OFFSET[11:0]	符号付き 6 ビットデータ	ユーザは OFFSET[5:0] を "000000" に設 定する必要があります。

チャンネル "i" に対して ADCx\_DR (レギュラチャンネル) または ADCx\_JDRy (インジェクトチャンネル、y=1、2、3、4) からデータを読み出す場合：

- 対応するチャンネルのオフセットの 1 つが有効 (ビット OFFSETy\_EN=1) である場合、読み出しデータは符号付きです。
- このチャンネルの 4 つのオフセットに有効なものがない場合、読み出しデータは符号なしです。

[図 82](#)、[図 83](#)、[図 84](#)、および [図 85](#) に、符号付きおよび符号なしのデータの配置を示します。

図 82. 右詰め (オフセット無効、符号なしの値)

12 ビットデータ															
ビット 15				ビット 7								ビット 0			
0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
10 ビットデータ															
ビット 15					ビット 7								ビット 0		
0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
8 ビットデータ															
ビット 15						ビット 7								ビット 0	
0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
6 ビットデータ															
ビット 15						ビット 7								ビット 0	
0	0	0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0

MS31015V1

図 83. 右詰め (オフセット有効、符号付きの値)

12 ビットデータ															
ビット 15				ビット 7								ビット 0			
SEXT	SEXT	SEXT	SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
10 ビットデータ															
ビット 15					ビット 7								ビット 0		
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
8 ビットデータ															
ビット 15						ビット 7								ビット 0	
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D7	D6	D5	D4	D3	D2	D1	D0
6 ビットデータ															
ビット 15						ビット 7								ビット 0	
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D5	D4	D3	D2	D1	D0

MS31016V1

図 84. 左詰め (オフセット無効、符号なしの値)

12ビットデータ															
ビット 15								ビット 7				ビット 0			
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
10ビットデータ															
ビット 15						ビット 7				ビット 0					
D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0
8ビットデータ															
ビット 15				ビット 7				ビット 0							
D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0
6ビットデータ															
ビット 15								ビット 7				ビット 0			
0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0

MS31017V1

図 85. 左詰め (オフセット有効、符号付きの値)

12ビットデータ															
ビット 15								ビット 7				ビット 0			
SEXT	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0
10ビットデータ															
ビット 15						ビット 7				ビット 0					
SEXT	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0
8ビットデータ															
ビット 15				ビット 7				ビット 0							
SEXT	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0
6ビットデータ															
ビット 15								ビット 7				ビット 0			
SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	SEXT	D5	D4	D3	D2	D1	D0	0

MS31018V1

## ADC オーバーラン (OVR、OVRMOD)

オーバーランフラグ (OVR) は、新しい変換データが使用可能になる前に、(CPU または DMA によって) レギュラ変換データが読み出されなかったときに、バッファオーバーランイベントを通知します。

OVR フラグは、新しい変換が完了した時点で EOC フラグが 1 のままであった場合にセットされます。ビット OVRIE=1 の場合、割り込みを生成することができます。

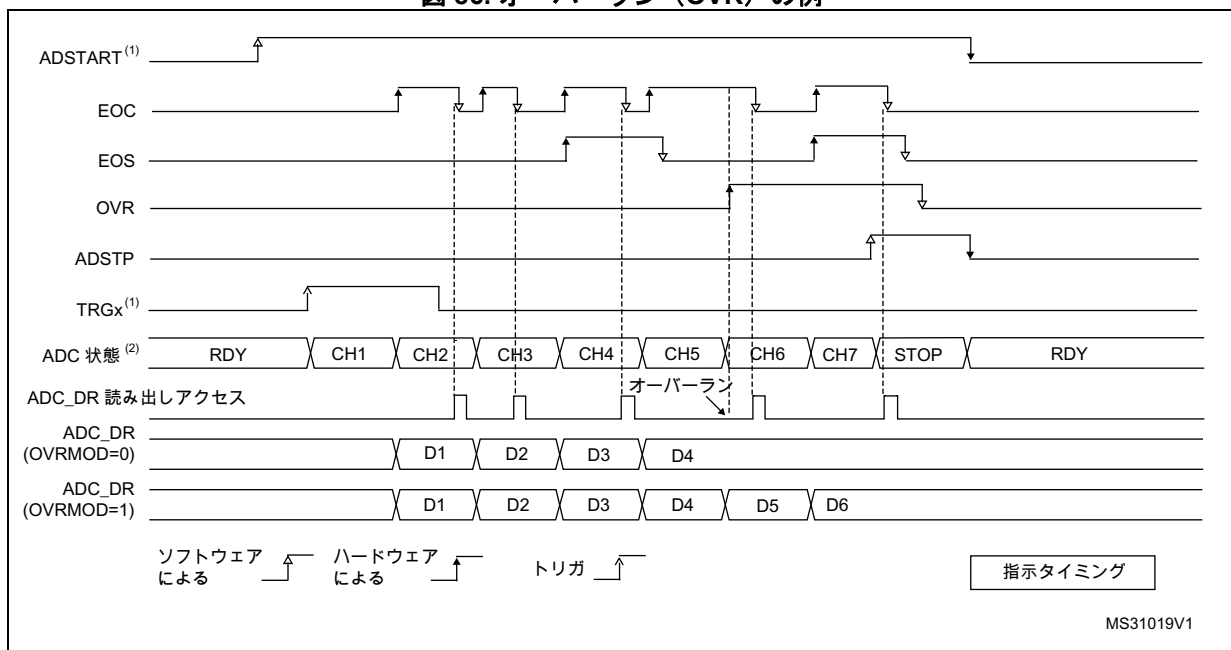
オーバーラン条件が発生すると、ADC は動作し続け、ソフトウェアがビット ADSTP=1 をセットすることによってシーケンスの停止およびリセットを決めるまで、変換を続行できます。

OVR フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

制御ビット OVRMOD をプログラムすることによって、オーバーランイベントが発生したときにデータが保存されるか上書きされるかを設定できます。

- OVRMOD=0: オーバーランイベントが発生しても、データレジスタは上書きされません。古いデータは保持され、新しい変換は破棄され、失われます。OVR が 1 のままの場合、その後の変換は行われますが、結果データは破棄されます。
- OVRMOD=1: データレジスタは最後の変換結果で上書きされ、以前の未読データは失われます。OVR が 1 のままの場合、その後の変換は通常どおり動作し、ADCx\_DR レジスタは常に最新の変換データを含みます。

図 86. オーバーラン (OVR) の例



注: インジェクトチャンネルでは、4 つのインジェクトチャンネルにそれぞれ専用のデータレジスタがあるため、オーバーランを検出する機能はありません。

## DMA を使用しない変換シーケンスの管理

変換が十分遅い場合、ソフトウェアで変換シーケンスを処理することができます。この場合、ソフトウェアは EOC フラグと関連の割り込みを使用して、各データを処理する必要があります。変換が終了するごとに、EOC はセットされ、ADCx\_DR レジスタを読み出すことができます。オーバーランイベントをエラーとして管理するには、OVRMOD を 0 に設定する必要があります。



## オーバーランなしでの DMA を使用しない変換の管理

毎回データの読み出しをせずに ADC に 1 つまたは複数のチャンネルを変換させると便利な場合があります (たとえば、アナログウォッチドッグがある場合)。この場合、OVRMOD ビットを 1 に設定する必要があります、ソフトウェアは OVR フラグを無視する必要があります。オーバーランイベントが発生しても ADC は変換を続行し、ADCx\_DR レジスタは常に最新の変換を含みます。

## DMA を使用した変換の管理

変換されたチャンネルの値は特定のデータレジスタに格納されるので、複数のチャンネルの変換には DMA の使用が便利です。これによって、ADCx\_DR レジスタにすでに格納されているデータの損失を防ぐことができます。

DMA モードが有効なとき (ADCx\_CFGR レジスタの DMAEN ビットがシングル ADC モードで 1 にセットされている場合、またはデュアル ADC モードで MDMA が 0b00 以外に設定されている場合)、各チャンネルの変換後、DMA リクエストが生成されます。これにより、変換データを ADCx\_DR レジスタからソフトウェアで選択した場所へ転送することができます。

これにもかかわらず、DMA が DMA 転送リクエストを時間内に処理できなかったためにオーバーランが発生した場合 (OVR=1)、ADC は DMA リクエストの生成を停止し、新しい変換に対応するデータは DMA によって転送されません。これは、RAM に転送されるすべてのデータを有効とみなすことができることを意味します。

OVRMOD ビットの設定に応じて、データは保存または上書きされます ([セクション : ADC オーバーラン \(OVR、OVRMOD\)](#) を参照してください)。

DMA 転送リクエストは、ソフトウェアが OVR ビットをクリアするまでブロックされます。

アプリケーションの用途に応じて 2 つの DMA モードがあり、シングル ADC モードでは ADCx\_CFGR レジスタの DMACFG ビットで設定され、デュアル ADC モードでは ADCx\_CCR レジスタの DMACFG ビットで設定されます。

- DMA ワンショットモード (DMACFG=0)。このモードは、DMA が固定数のデータを転送するようにプログラムされた場合に適しています。
- DMA サーキュラモード (DMACFG=1)。このモードは、DMA をサーキュラモードでプログラムする場合に適しています。

## DMA ワンショットモード (DMACFG=0)

このモードでは、ADC は新しい変換データが使用可能になるたびに DMA 転送リクエストを生成し、変換が再び開始された場合でも、DMA が最後の DMA 転送に達すると (DMA\_EOT 割り込みが発生すると) (DMA の段落を参照)、DMA リクエストの生成を停止します。

DMA 転送完了後 (DMA コントローラで設定されたすべての転送後) :

- ADC データレジスタの内容が停止されます。
- 実行中の変換は中止され、その部分的な結果は破棄されます。
- DMA コントローラに対する新しい DMA リクエストは発行されません。これによって、開始された変換がある場合のオーバーランエラーの生成を回避します。
- スキャンシーケンスは中止され、リセットされます。
- DMA は停止します。

## DMA サーキュラモード (DMACFG=1)

このモードでは、DMA が最後の DMA 転送に達した場合でも、ADC は新しい変換データがデータレジスタで使用可能になるたびに DMA 転送リクエストを生成します。これにより、DMA を連続的なアナログ入力データストリームを処理するようにサーキュラモードに設定できます。

## 15.3.27 動的低電力機能

### 自動遅延変換モード (AUTDLY)

ADC は、AUTDLY 設定ビットによって制御されている自動遅延変換モードを実装しています。自動遅延変換は、ソフトウェアを単純化する場合だけでなく、ADC オーバーランが発生するリスクのある低周波数のクロックで動作しているアプリケーションのパフォーマンスを最適化する場合に便利です。

AUTDLY=1 の場合、新しい変換は同じグループの前のデータがすべて処理された場合にのみ開始できます。

- レギュラ変換の場合：ADCx\_DR レジスタの読み出し後、または EOC ビットのクリア後 (図 87 を参照)
- インジェクト変換の場合：JEOS ビットのクリア後 (図 88 を参照)

これは、データを読み出すシステムの速度に ADC の速度を自動的に適応させる方法です。

遅延は、各レギュラ変換後 (DISCEN=0 か 1 かにかかわらず)、およびインジェクト変換の各シーケンス後 (JDISCEN=0 か 1 かにかかわらず) に挿入されます。

**注：** 遅延は、インジェクトシーケンスの各変換の間には挿入されません (最後の変換を除く)。

変換時、この遅延中に発生した (同じ変換グループに対する) ハードウェアトリガイベントは無視されます。

**注：** ソフトウェアトリガは、変換を再開するために、この遅延中に ADSTART ビットまたは JADSTART ビットをセットできるようにしている点で、ハードウェアトリガとは異なります。新しい変換を起動する前にデータを読み出すかどうかは、ソフトウェアに依存します。

遅延は、異なるグループの変換 (レギュラ変換に続いてインジェクト変換、またはその逆) の間には挿入されません。

- レギュラ変換の自動遅延中にインジェクトトリガが発生すると、すぐにインジェクト変換が開始されます (図 88 を参照)。
- インジェクトシーケンスが完了すると、新しいレギュラ変換を開始する前に、ADC は前のレギュラ変換の遅延を待ちます (終了していない場合) (図 90 を参照)。

自動インジェクトモード (JAUTO=1) では挙動が少し異なり、前のインジェクト変換シーケンスの自動遅延が終了した場合 (JEOS がクリアされた場合) にのみ新しいレギュラ変換を開始することができます。これは、新しいシーケンスを開始する前に、ソフトウェアが特定のシーケンスのデータをすべて読み出すことができるようにするためです (図 91 を参照)。

自動遅延モード (JAUTO=1、CONT=1、および AUTDLY=1) と組み合わせられた連続自動インジェクションモードの変換を停止するには、次の手順に従います。

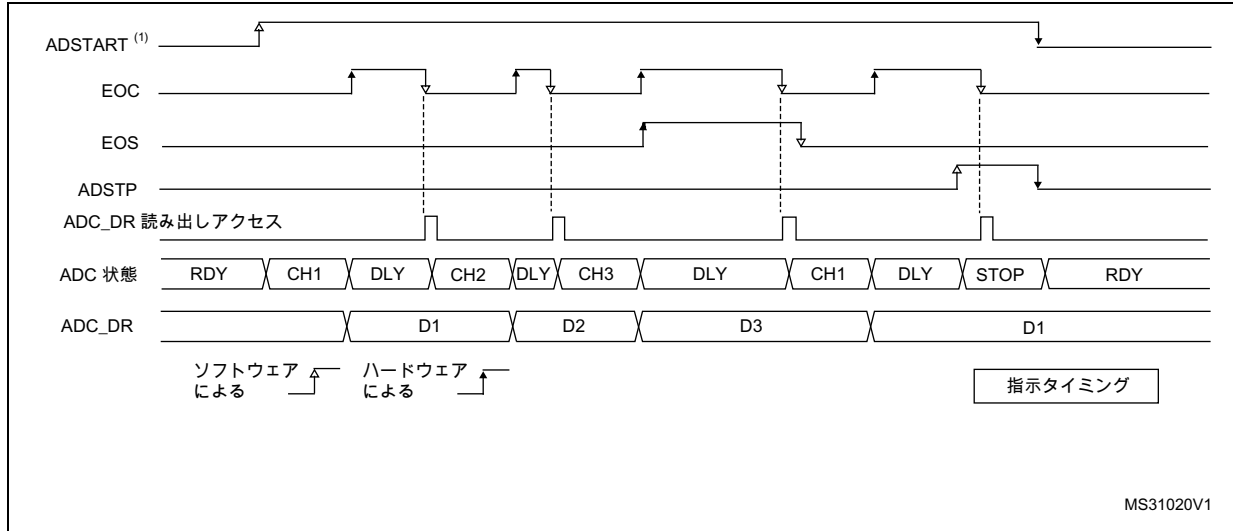
1. JEOS=1 になるまで (リスタートされる変換がなくなるまで) 待ちます。
2. JEOS をクリアします。
3. ADSTP=1 にセットします。
4. レギュラデータを読み出します。

この手順に従わない場合、ADSTP がセットされた後で JEOS がクリアされると、新しいレギュラシーケンスが再開されます。

AUTDLY モードでは、すでに実行中のレギュラシーケンス中、またはシーケンスの最後のレギュラ変換後の遅延中に発生した場合、ハードウェアレギュラトリガイベントは無視されます。ただし、この遅延後に発生した場合は、後続の遅延のインジェクトシーケンス中に発生した場合でも、保留とみなされます。その後、インジェクトシーケンスの遅延終了時に変換が開始されます。

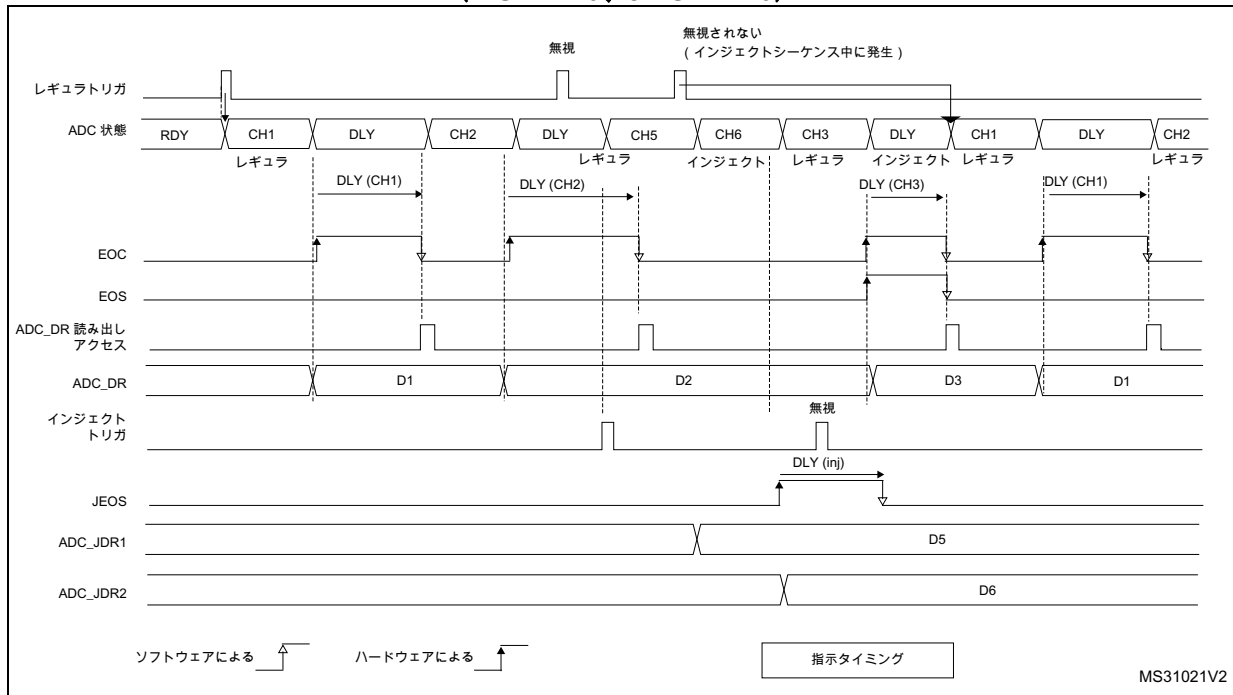
AUTDLY モードでは、すでに実行中のインジェクトシーケンス中、またはシーケンスの最後のインジェクト変換後の遅延中に発生した場合、ハードウェアインジェクトトリガイベントは無視されません。

図 87. AUTDLY=1、連続モードのレギュラ変換、ソフトウェアトリガ



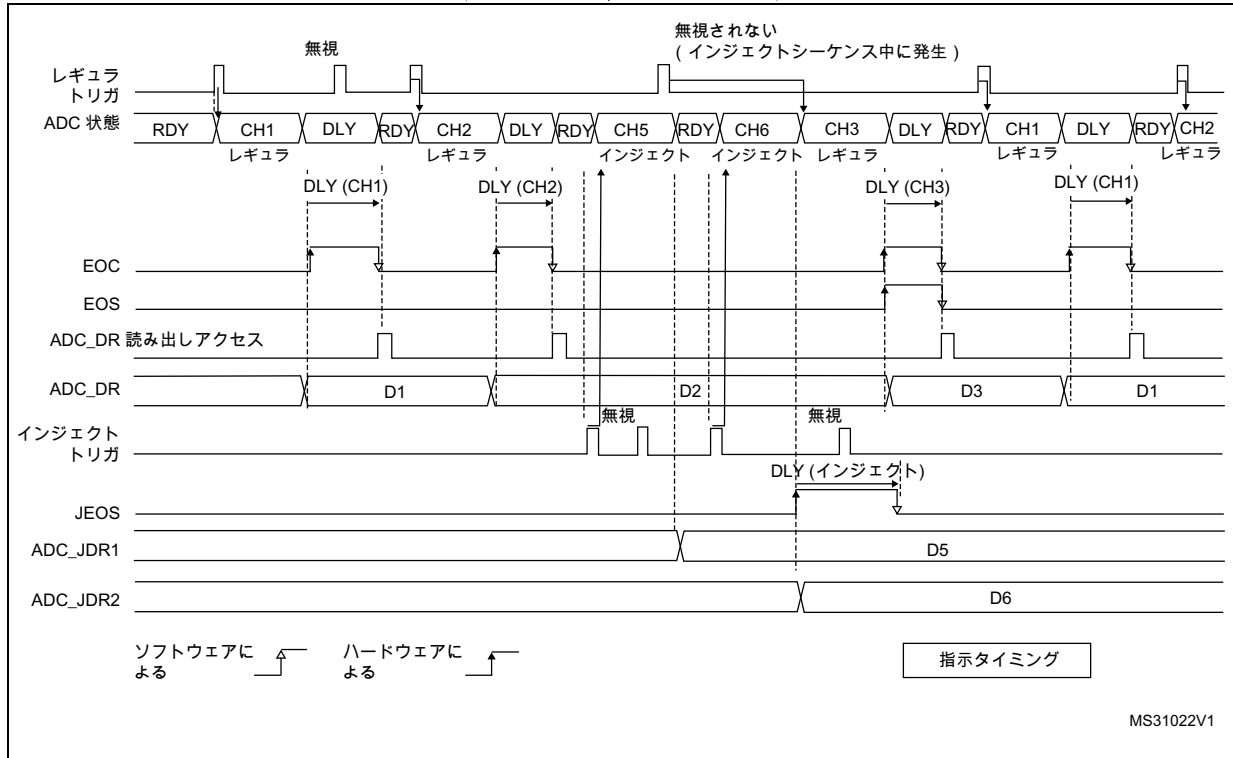
1. AUTDLY=1
2. レギュラ設定 : EXTEN=0x0 (ソフトウェアトリガ)、CONT=1、CHANNELS = 1、2、3
3. インジェクト設定 : 無効

図 88. AUTDLY=1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN=0、JDISCEN=0)



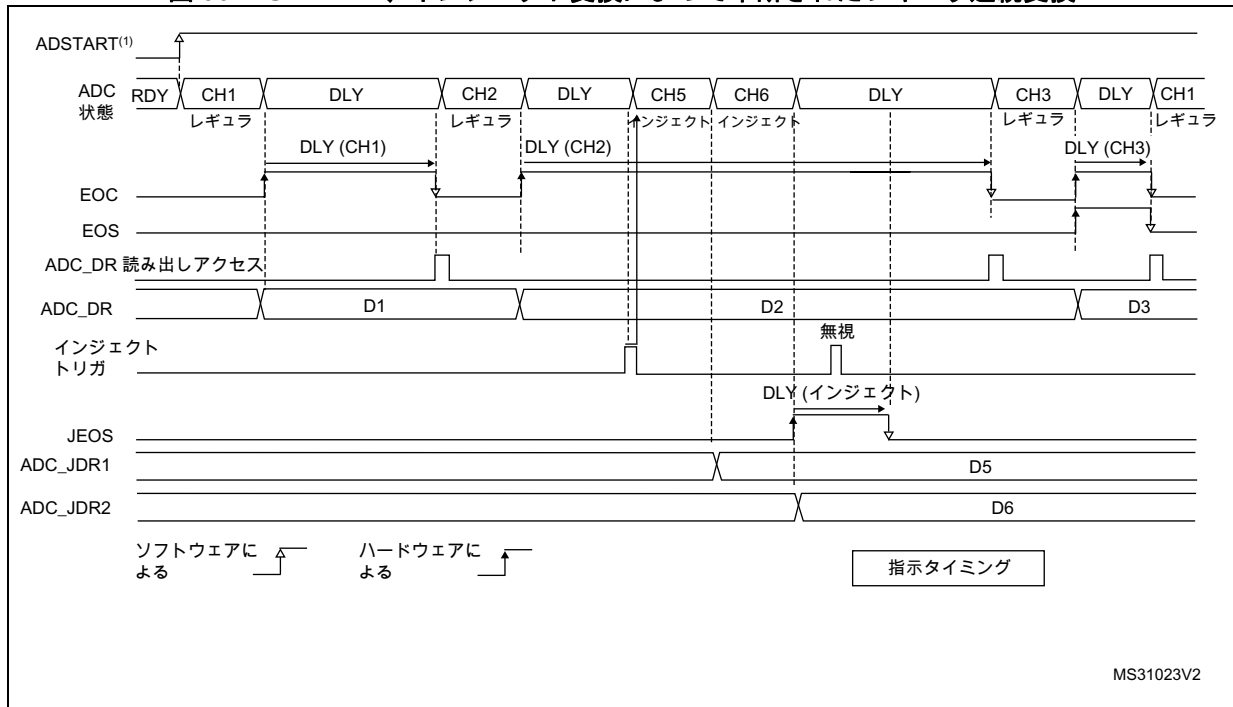
1. AUTDLY=1
2. レギュラ設定 : EXTEN=0x1 (ハードウェアトリガ)、CONT=0、DISCEN=0、CHANNELS = 1、2、3
3. インジェクト設定 : JEXTEN=0x1 (ハードウェアトリガ)、JDISCEN=0、CHANNELS = 5、6

図 89. AUTDLY=1、インジェクト変換によって中断されたレギュラハードウェア変換 (DISCEN=1、JDISCEN=1)



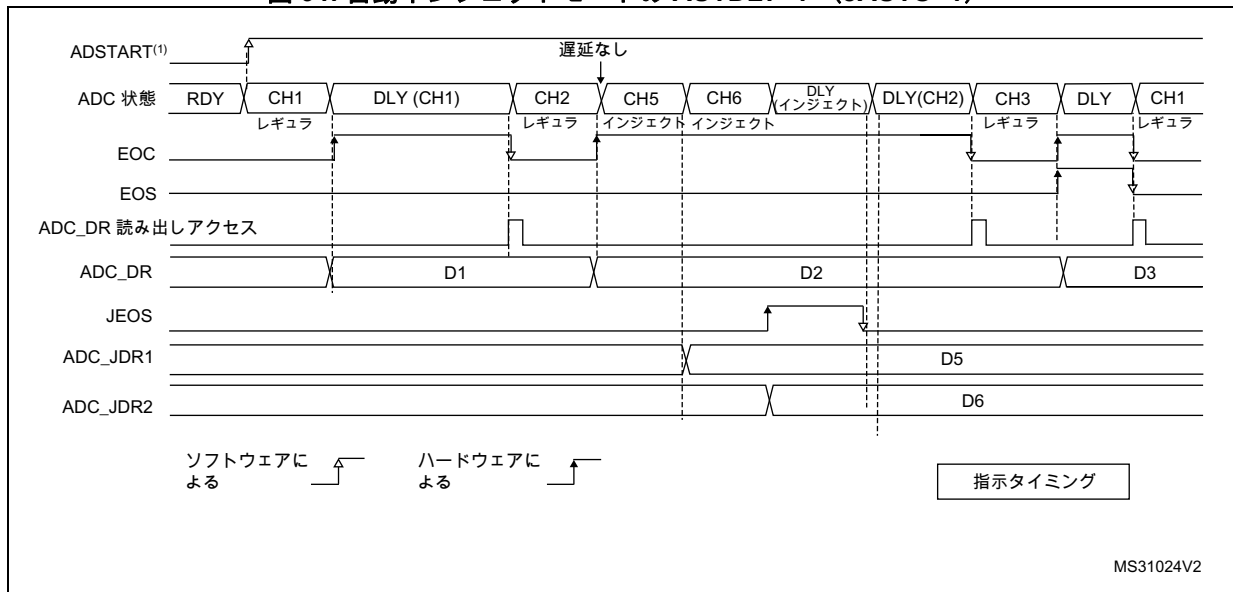
1. AUTDLY=1
2. レギュラ設定 : EXTEN=0x1 (ハードウェアトリガ)、CONT=0、DISCEN=1、DISCNUM=1、CHANNELS = 1、2、3
3. インジェクト設定 : JEXTEN=1x1 (ハードウェアトリガ)、JDISCEN=0、CHANNELS = 5、6

図 90. AUTDLY=1、インジェクト変換によって中断されたレギュラ連続変換



1. AUTDLY=1
2. レギュラ設定 : EXTEN=0x0 (ソフトウェアトリガ)、CONT=1、DISCEN=0、CHANNELS = 1、2、3
3. インジェクト設定 : JEXTEN=0x1 (ハードウェアトリガ)、JDISCEN=0、CHANNELS = 5、6

図 91. 自動インジェクトモードの AUTDLY=1 (JAUTO=1)

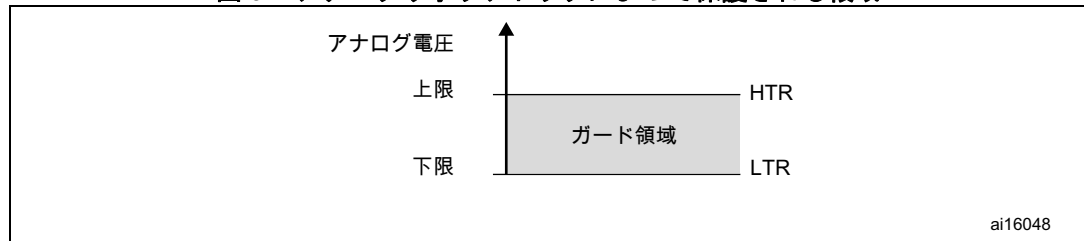


1. AUTDLY=1
2. レギュラ設定 : EXTEN=0x0 (ソフトウェアトリガ)、CONT=1、DISCEN=0、CHANNELS = 1、2
3. インジェクト設定 : JAUTO=1、CHANNELS = 5、6

## 15.3.28 アナログウィンドウウォッチドッグ (AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD\_HTx、AWD\_LTx、AWDx)

3 つの AWD アナログウォッチドッグは、設定された電圧範囲 (ウィンドウ) 内に一部のチャンネルが留まっているかどうかを監視します。

図 92. アナログウォッチドッグによって保護される領域



### AWDx フラグと割り込み

ADCx\_IER レジスタの AWDxIE (x=1、2、3) をセットすることによって、3 つのアナログウォッチドッグでそれぞれ割り込みを有効にできます。

AWDx (x=1、2、3) フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

ADC 変換結果は配置される前に低閾値および高閾値と比較されます。

### アナログウォッチドッグ 1 の説明

AWD アナログウォッチドッグ 1 は、ADCx\_CFGR レジスタの AWD1EN ビットをセットすることによって有効になります。ウォッチドッグは、選択された 1 つのチャンネルまたはすべての有効チャンネル<sup>(1)</sup>が設定された電圧範囲 (ウィンドウ) 内にとどまっているかどうかを監視します。

表 95 に、1 つまたは複数のチャンネル上でアナログウォッチドッグを有効にするためにどのようにして ADCx\_CFGR レジスタを設定するかを示します。

表 95. アナログウォッチドッグチャンネル選択

アナログウォッチドッグによって保護されるチャンネル	AWD1SGL ビット	AWD1EN ビット	JAWD1EN ビット
なし	x	0	0
すべてのインジェクトチャンネル	0	0	1
すべてのレギュラチャンネル	0	1	0
すべてのレギュラおよびインジェクトチャンネル	0	1	1
1 つの <sup>(1)</sup> インジェクトチャンネル	1	0	1
1 つの <sup>(1)</sup> レギュラチャンネル	1	1	0
1 つの <sup>(1)</sup> レギュラまたはインジェクトチャンネル	1	1	1

1. AWD1CH[4:0] ビットによって選択されます。チャンネルは、適切なレギュラまたはインジェクトシーケンスで変換されるようにプログラムする必要があります。

アナログウォッチドッグ (AWD1) ステータスビットは、ADC によって変換されたアナログ電圧が低閾値を下回るか、高閾値を上回る場合にセットされます。

これらの閾値は、アナログウォッチドッグ 1 の ADCx\_TR1 レジスタの HT1[11:0] ビットおよび LT1[11:0] ビットでプログラムされます。12 ビット未満の分解能でデータを変換するときには (RES[1:0] ビットに従って)、内部比較は常に 12 ビット全体の元の変換データ (左詰め) に対して実行されるため、プログラムされた閾値の LSB はクリアされたままである必要があります。

表 96 に、アナログウォッチドッグ 1 に可能なすべての分解能での比較方法を示します。

表 96. アナログウォッチドッグ 1 の比較

分解能 (RES[1:0] ビット)	アナログウォッチドッグ比較:		コメント
	元の変換データ、 左詰め <sup>(1)</sup>	閾値	
00 : 12 ビット	DATA[11:0]	LT1[11:0] および HT1[11:0]	-
01 : 10 ビット	DATA[11:2],00	LT1[11:0] および HT1[11:0]	ユーザは LT1[1:0] と HT1[1:0] を 00 に設定する必要があります。
10 : 8 ビット	DATA[11:4],0000	LT1[11:0] および HT1[11:0]	ユーザは LT1[3:0] と HT1[3:0] を 0000 に設定する必要があります。
11 : 6 ビット	DATA[11:6],000000	LT1[11:0] および HT1[11:0]	ユーザは LT1[5:0] と HT1[5:0] を 000000 に設定する必要があります。

1. 配置計算の前、かつオフセットを適用する前に、元の変換データに対してウォッチドッグ比較が行われます (比較対象のデータは符号なし)。

### アナログウォッチドッグ 2 および 3 の説明

2 番目と 3 番目のアナログウォッチドッグはより柔軟で、AWDxCH[18:1] (x=2, 3) の対応するビットをプログラムすることにより、いくつかの選択されたチャンネルをガードすることができます。

対応するウォッチドッグは、AWDxCH[18:0] (x=2, 3) の任意のビットがセットされると有効になります。

これらは 8 ビットの分解能に制限され、閾値の最上位の 8 ビットのみを HTx[7:0] と LTx[7:0] にプログラムすることができます。表 97 に、可能なすべての分解能での比較方法を示します。

表 97. アナログウォッチドッグ 2 および 3 の比較

分解能 (ビット RES[1:0])	アナログウォッチドッグ比較:		コメント
	元の変換データ、 左詰め <sup>(1)</sup>	閾値	
00 : 12 ビット	DATA[11:4]	LTx[7:0] および HTx[7:0]	DATA[3:0] はこの比較に適していません
01 : 10 ビット	DATA[11:4]	LTx[7:0] および HTx[7:0]	DATA[3:2] はこの比較に適していません
10 : 8 ビット	DATA[11:4]	LTx[7:0] および HTx[7:0]	-
11 : 6 ビット	DATA[11:6],00	LTx[7:0] および HTx[7:0]	ユーザは LTx[1:0] と HTx[1:0] を 00 に設定する必要があります

1. 配置計算の前、かつオフセットを適用する前に、元の変換データに対するウォッチドッグ比較が行われます (比較対象のデータは符号なし)。



## ADCy\_AWDx\_OUT 信号出力生成

各アナログウォッチドッグは、一部のオンチップタイマに ETR 入力 (外部トリガ) が直接接続された内部ハードウェア信号 ADCy\_AWDx\_OUT (y=ADC の数、x=ウォッチドッグの数) と関連付けられています。ETR として ADCy\_AWDx\_OUT 信号を選択する方法を理解するには、オンチップタイマのセクションを参照してください。

ADCy\_AWDx\_OUT は、関連付けられたアナログウォッチドッグが有効になった時にアクティブ化されます。

- ADCy\_AWDx\_OUT は、ガードされた変換がプログラムされた閾値の外にある場合にセットされます。
- ADCy\_AWDx\_OUT は、次のガードされた変換がプログラムされた閾値内にある場合、終了後にリセットされます (次のガードされた変換がプログラムされた閾値の外にある場合は 1 のまま維持されます)。
- ADCy\_AWDx\_OUT は ADC を無効にする場合にもリセットされます (ADDIS=1 に設定する場合)。レギュラまたはインジェクト変換の停止 (ADSTP=1 または JADSTP=1 に設定) は、ADCy\_AWDx\_OUT の生成にまったく影響しない点に注意してください。

**注:** AWDx フラグは、ハードウェアによってセットされ、ソフトウェアによってリセットされます。AWDx フラグは、ADCy\_AWDx\_OUT の生成にまったく影響しません (ソフトウェアによって AWDx フラグがクリアされず、このフラグが 1 のまま維持される間、ADCy\_AWDx\_OUT をトグルできます)。

図 93. ADCy\_AWDx\_OUT 信号生成 (すべてのレギュラチャネル)

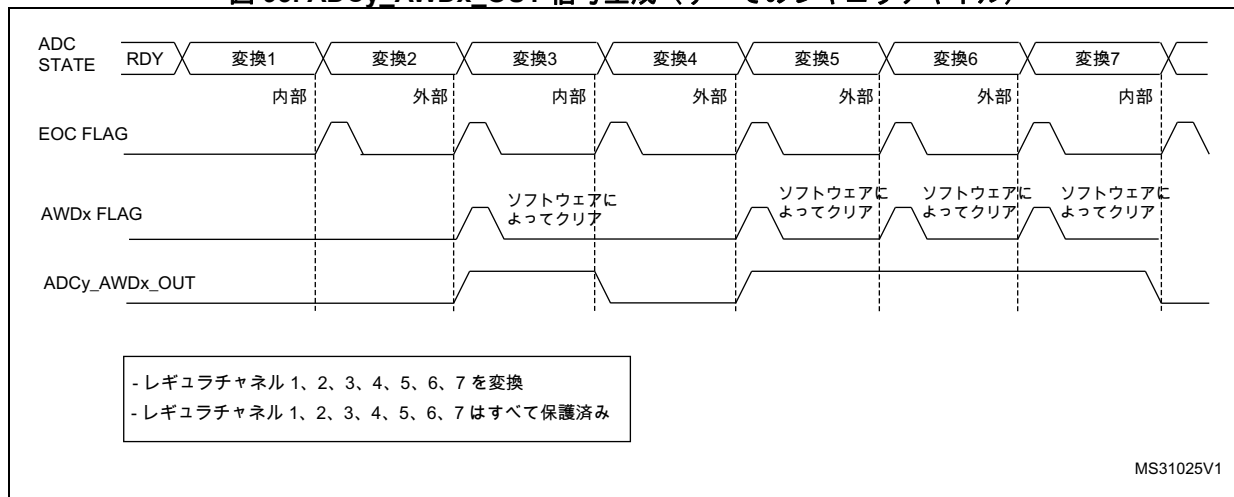


図 94. ADCy\_AWDx\_OUT 信号生成 (ソフトウェアによってAWDx フラグがクリアされない場合)

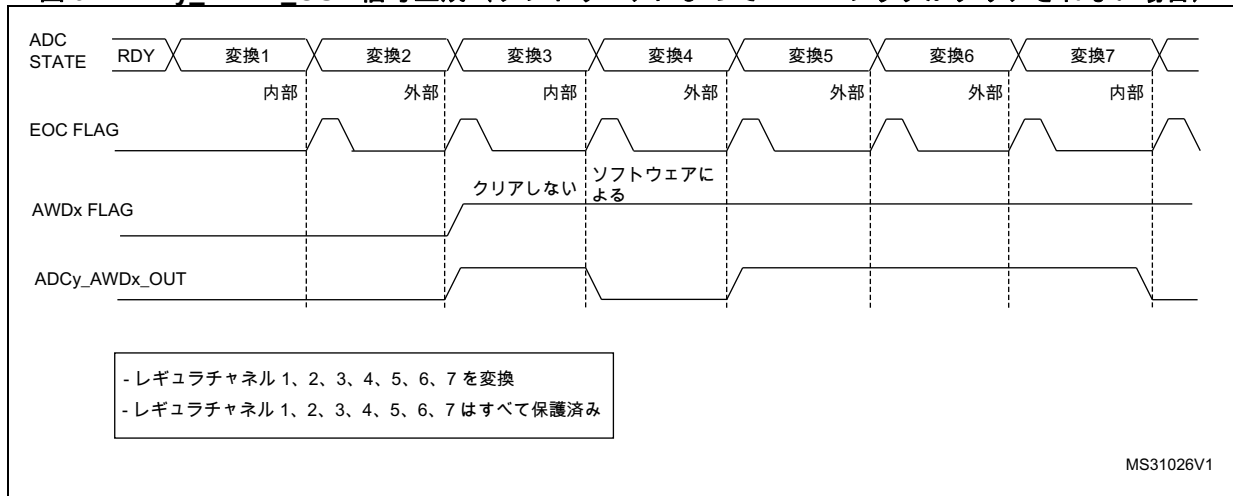


図 95. ADCy\_AWDx\_OUT 信号生成 (1つのレギュラチャネル)

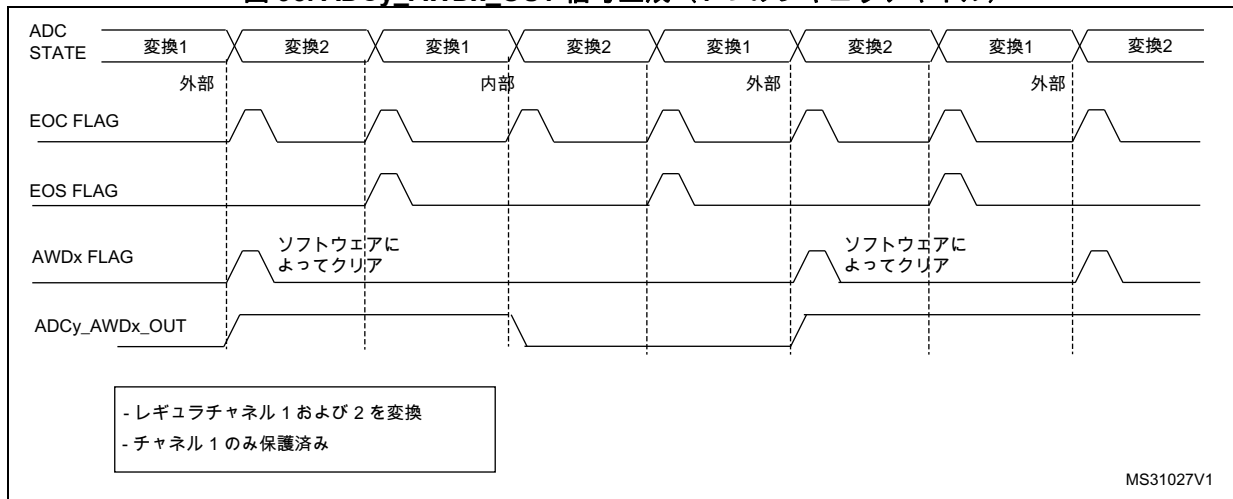
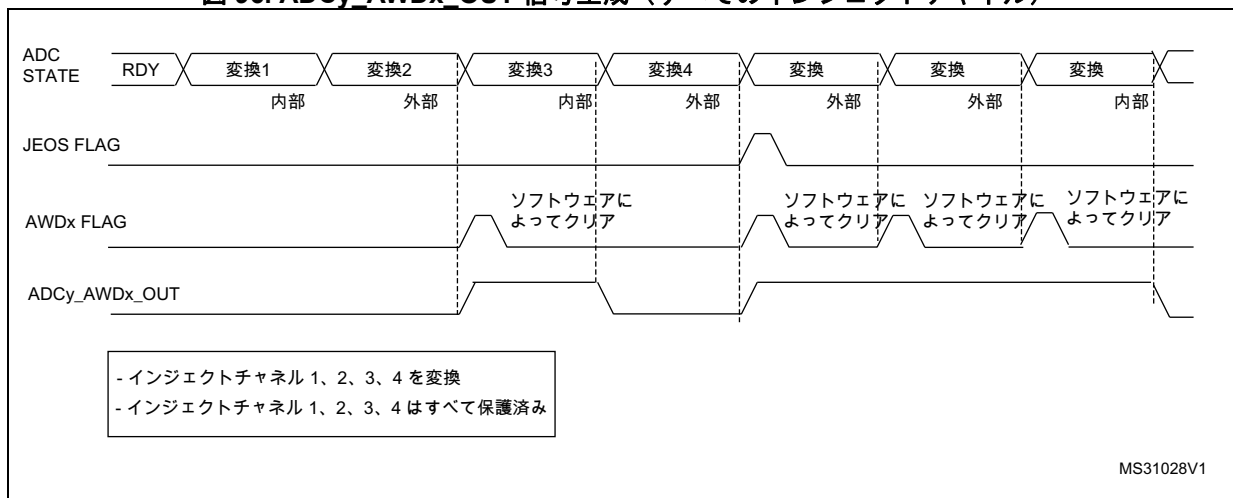


図 96. ADCy\_AWDx\_OUT 信号生成 (すべてのインジェクトチャネル)



## 15.3.29 デュアル ADC モード

2 つ以上の ADC を持つデバイスでは、デュアル ADC モードを使用することができます (図 97 を参照)。

- ADC1 および ADC2 は、デュアルモードで一緒に使用できます (ADC1 がマスタ)。
- ADC3 および ADC4 は、デュアルモードで一緒に使用できます (ADC3 がマスタ)。

デュアル ADC モードでは、変換の開始は ADCx\_CCR レジスタの DUAL[4:0] ビットによって選択されたモードに応じて、ADCx マスタにより ADC スレーブに交互または同時にトリガされます。

次の 4 つのモードを備えています。

- インジェクト同時モード
- レギュラ同時モード
- インタリーブモード
- オルタネートトリガモード

これらのモードを次のように組み合わせて使用することも可能です。

- インジェクト同時モード + レギュラ同時モード
- レギュラ同時モード + オルタネートトリガモード

デュアル ADC モードで (ADCx\_CCR レジスタの DUAL[4:0] ビットがゼロ以外の場合)、ADCx\_CFGR レジスタの CONT、AUTDLY、DISCEN、DISCNUM[2:0]、JDISCEN、JQM、JAUTO ビットは、マスタおよびスレーブ ADC の間で共有されます。スレーブ ADC のビットは、常にマスタ ADC の対応するビットと等しくなります。

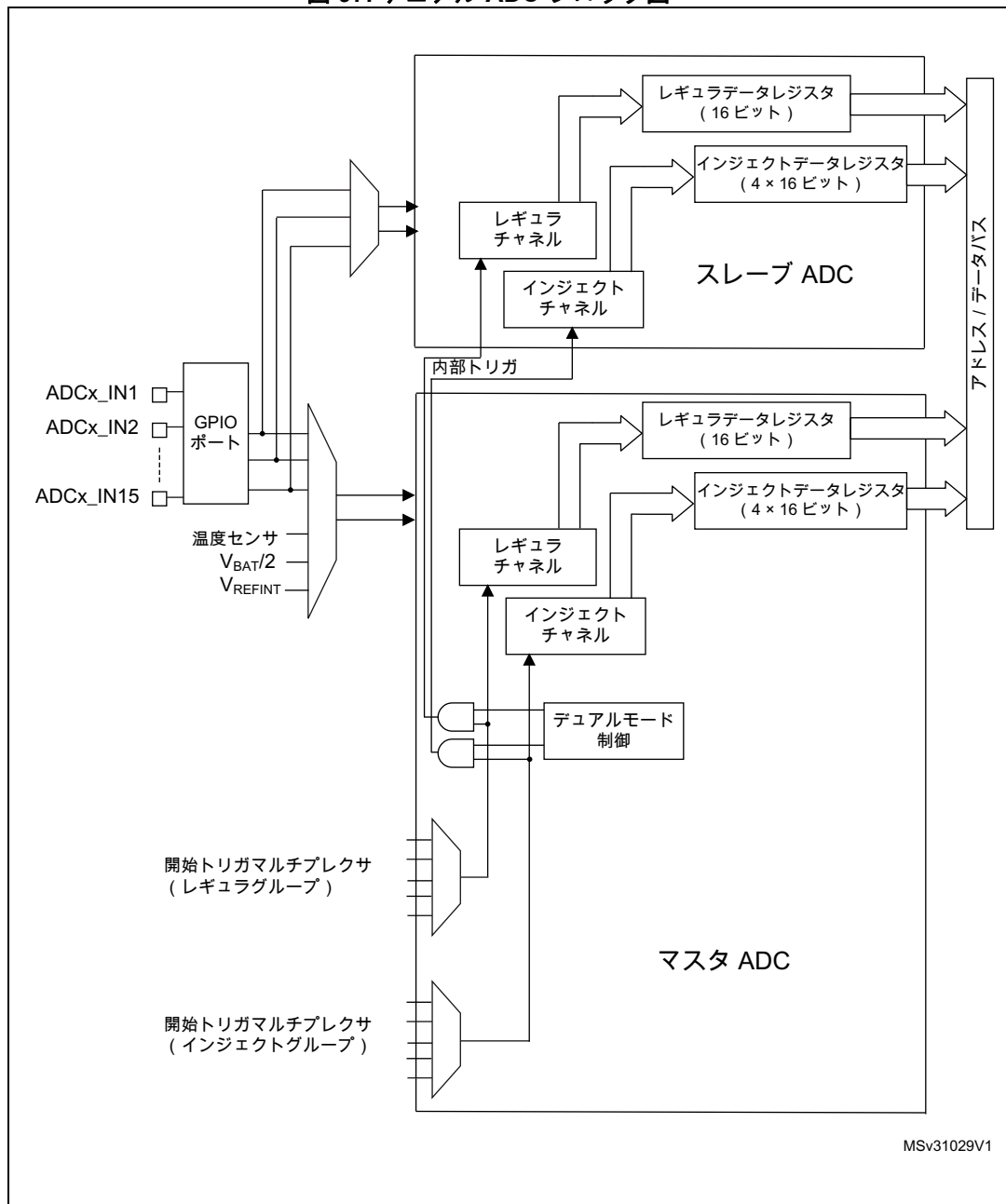
デュアルモードで変換を開始するには、ソフトウェアまたはハードウェアトリガやレギュラまたはインジェクトトリガを設定するために、マスタ ADC の EXTEN、EXTSEL、JEXTEN、JEXTSEL ビットのみをプログラムする必要があります (スレーブ ADC の EXTEN[1:0] および JEXTEN[1:0] ビットは無視されます)。

レギュラ同時モードまたはインタリーブモードでは、マスタ ADC の ADSTART または ADSTP ビットを一度セットすると、スレーブ ADC の対応するビットも自動的にセットされます。ただし、スレーブ ADC の ADSTART または ADSTP ビットは、必ずしもマスタ ADC ビットと同時にクリアされません。

インジェクト同時モードまたはオルタネートトリガモードでは、マスタ ADC の JADSTART または JADSTP ビットを一度セットすると、スレーブ ADC の対応するビットも自動的にセットされます。ただし、スレーブ ADC の JADSTART または JADSTP ビットは、必ずしもマスタ ADC ビットと同時にクリアされません。

デュアル ADC モードでは、ADC 共通データレジスタ (ADCx\_CDR) を読み出すことで、マスタおよびスレーブ ADC の変換データを並列で読み出すことができます。ステータスビットも、デュアルモードステータスレジスタ (ADCx\_CSR) を読み出すことで、並列で読み出すことができます。

図 97. デュアル ADC ブロック図 (1)



1. スレーブ ADC にも外部トリガは存在しますが、この図には表示していません。
2. ADC 共通データレジスタ (ADCx\_CDR) にはマスタとスレーブの両方の ADC レギュラ変換データが格納されます。

## インジェクト同時モード

このモードは、ビット DUAL[4:0]=00101 をプログラムすることによって選択されます。

このモードは、インジェクトチャンネルグループを変換します。外部トリガのソースは、マスタ ADC のインジェクトグループマルチプレクサ (ADCx\_JSQR レジスタの JEXTSEL[3:0] ビットによって選択) から供給されます。

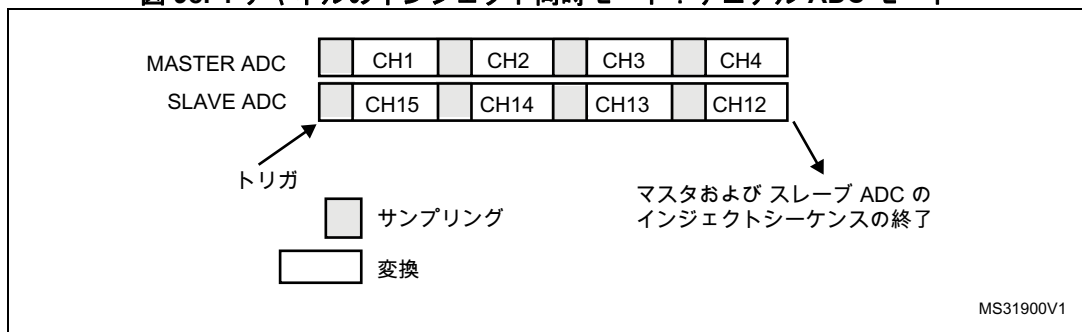
**注：** 2 つの ADC の同じチャンネルを変換しないでください (同じチャンネルを変換する場合、2 つの ADC のサンプリング時間が重なってはいけません)。

同時モードでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

レギュラ変換であれば、1 つまたはすべての ADC で行うことができます。その場合、それらは互いに独立していてインジェクトイベントが発生すると中断されます。それら (レギュラ変換) はインジェクト変換グループ終了時に再開されます。

- マスタ ADC での変換イベントのインジェクトシーケンスの終了時 (JEOS)、変換データはマスタ ADCx\_JDRy レジスタに格納され、JEOS 割り込みが生成されます (有効な場合)。
- スレーブ ADC での変換イベントのインジェクトシーケンスの終了時 (JEOS)、変換データはスレーブ ADCx\_JDRy レジスタに格納され、JEOS 割り込みが生成されます (有効な場合)。
- マスタインジェクトシーケンスの長さがスレーブインジェクトシーケンスの長さと同じ場合 (例: 図 98)、ソフトウェアでは 2 つの JEOS 割り込みのうちの一つのみを有効にすることができます (例: マスタ JEOS)、(マスタ ADCx\_JDRy とスレーブ ADCx\_JDRy レジスタの) 両方の変換データを読み出すことができます。

図 98.4 チャンネルのインジェクト同時モード：デュアル ADC モード



JDISCEN=1 の場合、インジェクトシーケンスの各同時変換にはインジェクトトリガイイベントが発生する必要があります。

このモードは、AUTDLY モードと組み合わせることができます。

- 変換の同時インジェクトシーケンスの終了後、マスタとスレーブADCの両方の JEOS ビットがクリアされた場合 (遅延フェーズ) にのみ、新しいインジェクトトリガイイベントは受け入れられます。実行中のインジェクトシーケンスで発生したすべての新しいインジェクトトリガイイベントと、これに関連する遅延フェーズは無視されます。
- マスタ ADC のレギュラ変換シーケンスが終了すると、マスタデータレジスタ (ADCx\_DR) が読み出された場合にのみ、マスタ ADC の新しいレギュラトリガイイベントが受け入れられます。実行中のレギュラシーケンスでマスタ ADC に対して発生したすべての新しいレギュラトリガイイベントと、これに関連する遅延フェーズは無視されます。スレーブ ADC で発生するレギュラシーケンスでも同じ挙動を示します。

## 独立インジェクト変換に対応したレギュラ同時モード

このモードは、ビット DUAL[4:0]=00110 をプログラムすることによって選択されます。

このモードは、レギュラチャンネルグループに対して行われます。外部トリガのソースは、マスタ ADC のレギュラグループマルチプレクサ (ADCx\_CFGR レジスタの EXTSEL[3:0] ビットによって選択) から供給されます。同時トリガは、スレーブ ADC に供給されます。

このモードでは、独立インジェクト変換がサポートされます。(マスタまたはスレーブでの) インジェクションリクエストにより、現在の同時変換がアボートされます。これは、インジェクト変換の完了後に再開されます。

**注：** 2 つの ADC の同じチャンネルを変換しないでください (同じチャンネルを変換する場合、2 つの ADC のサンプリング時間が重なってはいけません)。

レギュラ同時モードでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち変換時間が長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。

データが読み出せるタイミングは、割り込みによってソフトウェアに通知されます。

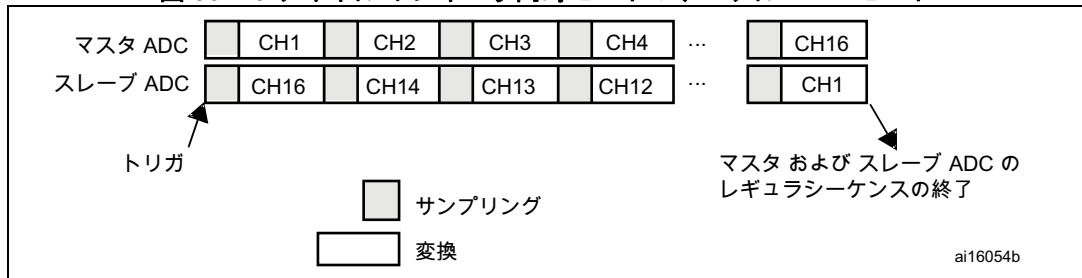
- マスタ ADC での各変換イベントの終了時 (EOS)、マスタ EOC 割り込みが生成され (EOCIE が有効な場合)、ソフトウェアではマスタ ADC の ADCx\_DR を読み出すことができます。
- スレーブ ADC での各変換イベントの終了時 (EOS)、スレーブ EOC 割り込みが生成され (EOCIE が有効な場合)、ソフトウェアではスレーブ ADC の ADCx\_DR を読み出すことができます。
- マスタレギュラシーケンスの長さがスレーブレギュラシーケンスの長さと同じ場合 (例：[図 99](#))、ソフトウェアでは 2 つの EOC 割り込みのうちの 1 つのみを有効にすることができます (例：マスタ EOC)、共通データレジスタ (ADCx\_CDR) から両方の変換データを読み出すことができます。

DMA を使用してレギュラデータを読み出すことができます。次の 2 つの方法が可能です。

- 2 つの DMA チャンネルを使用します (1 つはマスタ用、もう 1 つはスレーブ用)。この場合、MDMA[1:0] ビットをクリア状態に保つ必要があります。
  - マスタから ADCx\_DR を読み出すために、DMA のマスタ ADC チャンネルを設定します。DMA リクエストは、マスタ ADC の各 EOC イベントで生成されます。
  - スレーブから ADCx\_DR を読み出すために、DMA のスレーブ ADC チャンネルを設定します。DMA リクエストは、スレーブ ADC の各 EOC イベントで生成されます。
- MDMA モードを使用して、他の用途のために 1 つの DMA チャンネルを残します。
  - MDMA[1:0]=0b10 または 0b11 を設定します (分解能による)。
  - DMA チャンネルを 1 つ使用します (マスタのうちの 1 つ)。共通 ADC レジスタ (ADCx\_CDR) を読み出すために、DMA のマスタ ADC チャンネルを設定します。
  - マスタおよびスレーブの両方の EOC イベントが発生するたびに DMA リクエストが 1 つ生成されます。このとき、スレーブ ADC の変換データを ADCx\_CDR 32 ビットレジスタの上位ハーフワードで使用でき、またマスタ ADC の変換データを ADCx\_CCR レジスタの下位ハーフワードで使用できます。
  - EOC フラグは、DMA が ADCx\_CCR レジスタを読み出すと両方もクリアされます。

**注：** MDMA モード (MDMA[1:0]=0b10 または 0b11) で、ユーザはマスタのシーケンスとスレーブのシーケンスに同じ変換数をプログラムする必要があります。そうしないと、残りの変換では DMA リクエストは生成されません。

図 99. 16 チャンルのレギュラ同時モード：デュアル ADC モード



DISCEN=1 の場合、レギュラシーケンスの各「n」回の同時変換でレギュラトリガイイベントが発生する必要があります（「n」は DISCNUM で定義される）。

このモードは、AUTDLY モードと組み合わせることができます。

- シーケンスの同時変換の終了後、シーケンスの次の変換は、共通データレジスタ ADCx\_CDR（またはマスタ ADC のレギュラデータレジスタ）が読み出された場合（遅延フェーズ）にのみ開始されます。
- 変換の同時レギュラシーケンスの終了後、新しいレギュラトリガイイベントは共通データレジスタ (ADCx\_CDR) が読み出された場合（遅延フェーズ）にのみ受け入れられます。実行中のレギュラシーケンスで発生したすべての新しいレギュラトリガイイベントと、これに関連する遅延フェーズは無視されます。

AUTDLY モードと組み合わせたレギュラ同時モードでデータを処理するために、DMA を使用することができます（このとき、マルチ DMA モードが使われるものとします）。MDMA ビットは 0b10 または 0b11 にセットする必要があります。

レギュラ同時モードを AUTDLY モードと組み合わせる場合、ユーザは次の内容を確認する必要があります。

- マスタのシーケンスの変換数が、スレーブの変換数と同じであること。
- シーケンスが同時変換されるたび、スレーブ ADC の変換の長さがマスタ ADC の変換の長さよりも短いこと。シーケンスの長さは、変換するチャンネルの数および各チャンネルのサンプリングタイムと分解能に依存する点に注意してください。

**注：** このレギュラ同時モードと AUTDLY モードの組み合わせは、レギュラチャンネルのみがプログラムされている場合に制限されます。この組み合わせモードでインジェクトチャンネルをプログラムすることは禁止されています。

## 独立インジェクト変換に対応したインタリーブモード

このモードは、ビット DUAL[4:0]=00111 をプログラムすることによって選択されます。

このモードは、レギュラグループ（通常は 1 つのチャンネル）でのみ開始できます。外部トリガソースは、マスタ ADC のレギュラチャンネルマルチプレクサから供給されます。

外部トリガが発生した後、

- マスタ ADC はすぐに開始します。
- スレーブ ADC は、マスタ ADC のサンプリングフェーズが完了した後で、数 ADC クロックサイクルの遅延後に開始します。

インタリーブモードにおける 2 つの変換間の最小遅延は ADCx\_CCR レジスタの DELAY ビットで設定します。遅延のカウントは、マスタ変換のサンプリングフェーズ終了後に開始します。これにより、ADC は相補 ADC が依然として入力をサンプリングしている場合、変換を開始できません (与えられた時間に 1 つの ADC のみ入力信号をサンプリングすることができます)。

- DELAY に設定可能な最小値は 1 です。これにより、マスタ ADC サンプリングフェーズのアナログスイッチを開いてスレーブ ADC サンプリングフェーズのアナログスイッチを閉じるまでに、最低 1 つのサイクル時間を確保できます。
- 最大 DELAY 値は選択した分解能に対するサイクル数と同じです。ただし、ほかの ADC が依然として入力をサンプリングしているときに ADC が変換を開始しないよう、ユーザはこの遅延を適切に計算する必要があります。

マスタとスレーブの両方の ADC で CONT ビットがセットされている場合、両方の ADC の選択されたレギュラチャネルが連続的に変換されます。

データを読み出すタイミングは、割り込みによってソフトウェアに通知されます。

- マスタ ADC での各変換イベント (EOS) の終了時、マスタ EOC 割り込みが生成され (EOCIE が有効な場合)、ソフトウェアではマスタ ADC の ADCx\_DR を読み出すことができます。
- スレーブ ADC での各変換イベント (EOS) の終了時、スレーブ EOC 割り込みが生成され (EOCIE が有効な場合)、ソフトウェアではスレーブ ADC の ADCx\_DR を読み出すことができます。

**注：** *スレーブの EOC 割り込みのみを有効にして共通データレジスタ (ADCx\_CDR) を読み出すことができます。ただしこの場合、ユーザは変換時間が適合しており、それによって新しいマスタ変換をリスタートする前に、シーケンス内では常にマスタ変換に続いてスレーブ変換が実行されることを確認する必要があります。*

DMA を使用してレギュラデータを読み出すことができます。次の 2 つの方法が可能です。

- 2 つの DMA チャンネルを使用します (1 つはマスタ用、もう 1 つはスレーブ用)。この場合、ビット MDMA[1:0] をクリア状態に保つ必要があります。
  - マスタから ADCx\_DR を読み出すために、DMA のマスタ ADC チャンネルを設定します。DMA リクエストは、マスタ ADC の各 EOC イベントで生成されます。
  - スレーブから ADCx\_DR を読み出すために、DMA のスレーブ ADC チャンネルを設定します。DMA リクエストは、スレーブ ADC の各 EOC イベントで生成されます。
- MDMAモードを使用して、DMA チャンネルを 1 つ確保します。
  - MDMA[1:0]=0b10 または 0b11 を設定します (分解能による)。
  - DMA チャンネルを 1 つ使用します (マスタのうちの 1 つ)。共通 ADC レジスタ (ADCx\_CDR) を読み出すために、DMA のマスタ ADC チャンネルを設定します。
  - マスタおよびスレーブの両方の EOC イベントが発生するたびに DMA リクエストが 1 つ生成されます。このとき、スレーブ ADC の変換データを ADCx\_CDR 32 ビットレジスタの上位ハーフワードで使用でき、またマスタ ADC の変換データを ADCx\_CCR レジスタの下位ハーフワードで使用できます。
  - EOC フラグは、DMA が ADCx\_CCR レジスタを読み出すと両方ともクリアされます。



図 100. 連続変換モードにおける 1 チャンルのインタリーブモード：デュアル ADC モード

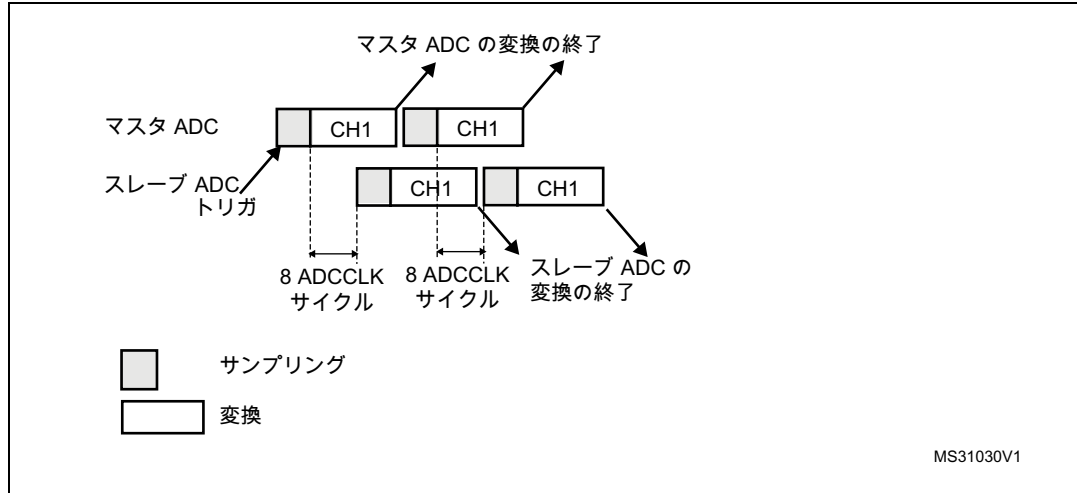
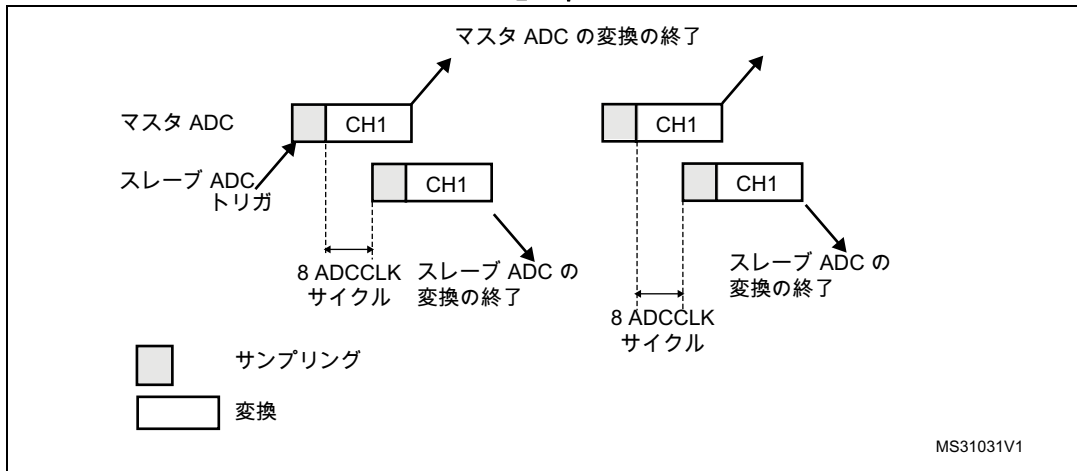


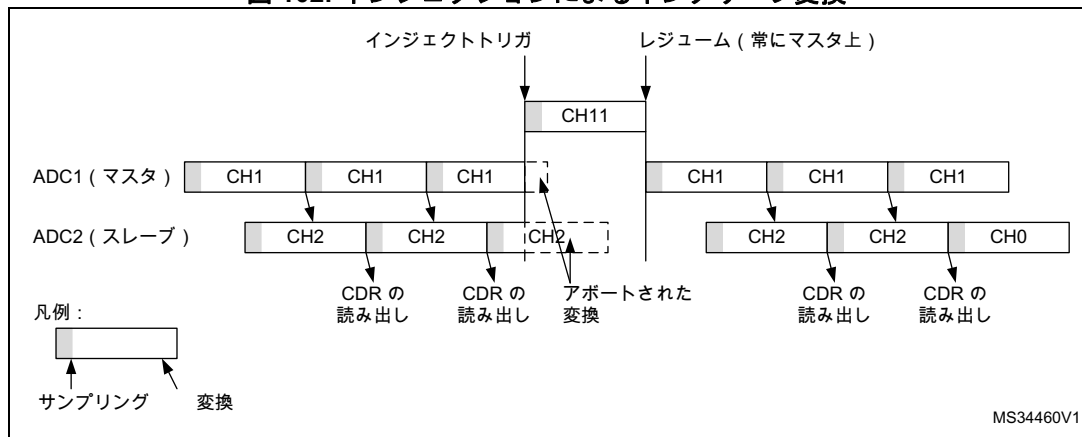
図 101. シングル変換モードにおける 1 チャンルのインタリーブモード：デュアル ADC モード



DISCEN=1 の場合、レギュラシーケンスの各「n」回の同時変換（「n」は DISCNUM で定義される）でレギュラトリガイベントが発生する必要があります。

このモードでは、インジェクト変換がサポートされます。（マスタまたはスレーブでの）インジェクションが完了すると、マスタとスレーブの両方のレギュラ変換がアボートされ、マスタからシーケンスが再開されます（以下の [図 102](#) を参照）。

図 102. インジェクションによるインタリーブ変換



## オルタネートトリガモード

このモードは、ビット DUAL[4:0]=01001 をプログラムすることによって選択されます。

このモードは、インジェクトグループでのみ開始できます。外部トリガのソースは、マスタ ADC のインジェクトグループマルチプレクサです。

このモードは、ハードウェアトリガの選択時にのみ使用できます。JEXTEN を 0x0 にセットすることはできません。

インジェクト不連続モードは無効です (両方の ADC で JDISCEN=0)。

- 最初のトリガが発生すると、グループ内のすべてのインジェクトされたマスタ ADC チャンネルが変換されます。
- 2 番目のトリガが発生すると、グループ内のすべてのインジェクトされたスレーブ ADC チャンネルが変換されます。
- 以下同様です。

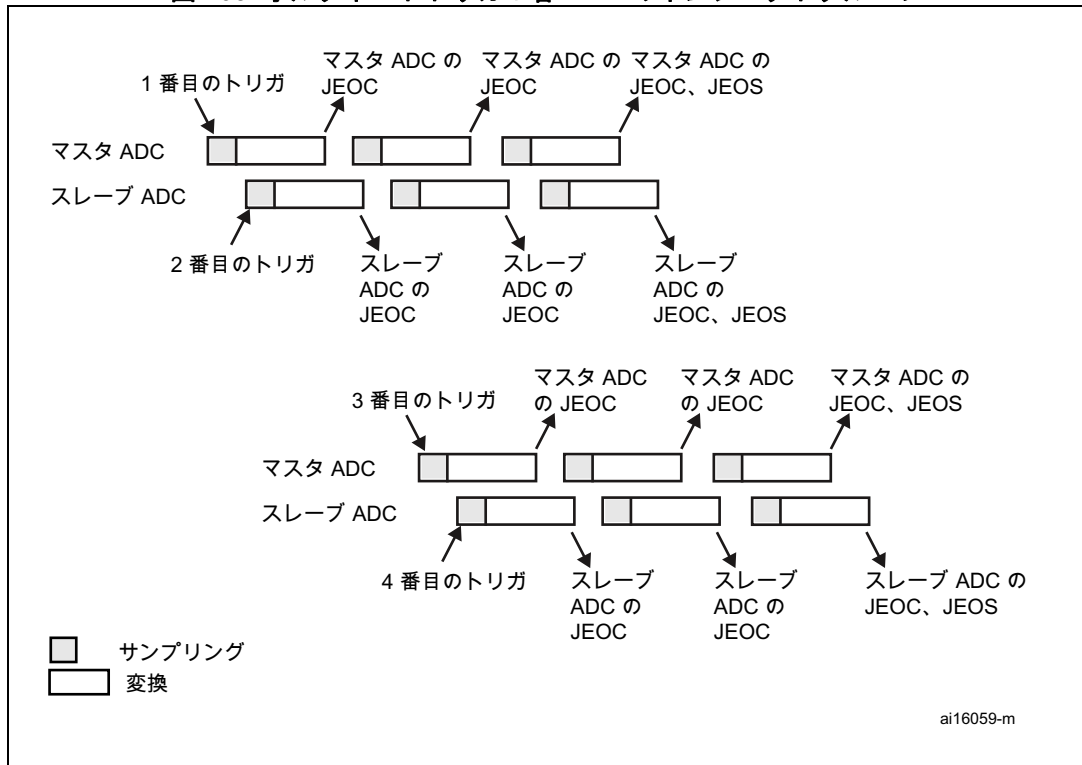
グループ内のマスタ ADC のすべてのインジェクトチャンネルが変換されると、JEOS 割り込み (有効な場合) が生成されます。

グループ内のスレーブ ADC のすべてのインジェクトチャンネルが変換されると、JEOS 割り込み (有効な場合) が生成されます。

JEOC 割り込み (有効な場合) は、各インジェクト変換後に生成することもできます。

グループ内すべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、グループ内のマスタ ADC のインジェクトチャンネルを変換することによって、オルタネートトリガプロセスがリスタートします。

図 103. オルタネートトリガ：各 ADC のインジェクトグループ



注：レギュラ変換であれば1 つまたはすべての ADC で有効にすることができます。この場合、レギュラ変換は互いに独立しています。ADC がインジェクト変換を行う必要があるとき、レギュラ変換は中断されます。インジェクト変換が終了すると(レギュラ変換が)再開されます。  
 2 つのトリガイベント間の間隔は1 ADC クロック期間と同等か長い必要があります。同じ ADC で変換を開始する2 つのトリガイベント間の最小間隔はシングル ADC モードと同じです。

インジェクト不連続モードは有効です (両方の ADC で JDISCEN=1)。

マスタとスレーブの両方の ADC に対してインジェクト不連続モードが有効な場合、

- 最初のトリガが発生すると、マスタ ADC の最初のインジェクトチャンネルが変換されます。
- 2 番目のトリガが発生すると、スレーブ ADC の最初のインジェクトチャンネルが変換されます。
- 以下同様です。

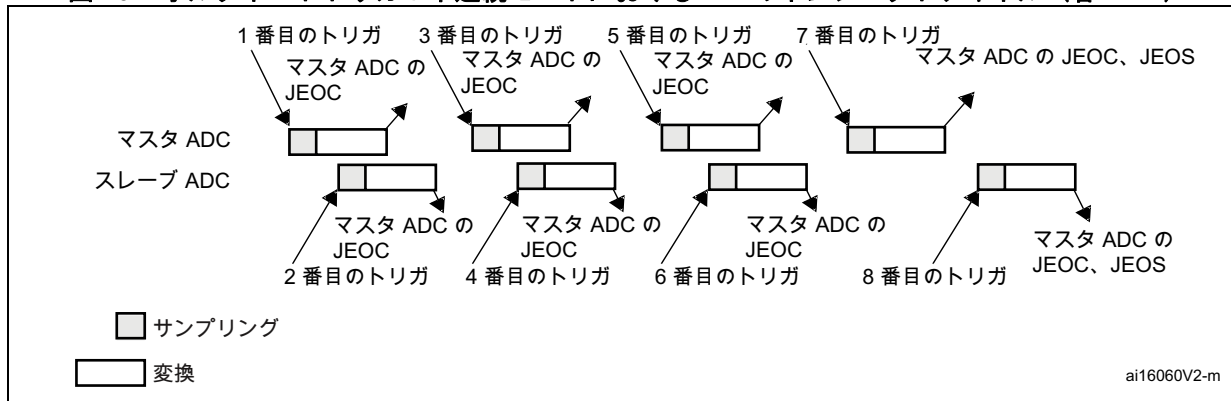
グループ内のマスタ ADC のすべてのインジェクトチャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

グループ内のスレーブ ADC のすべてのインジェクトチャンネルが変換されると、JEOC 割り込み (有効な場合) が生成されます。

JEOC 割り込み (有効な場合) は、各インジェクト変換後に生成することもできます。

グループ内のすべてのインジェクトチャンネルが変換された後で別の外部トリガが発生した場合は、オルタネートトリガプロセスがリスタートします。

図 104. オルタネートトリガ：不連続モードにおける 4 つのインジェクトチャンネル (各 ADC)



### レギュラ/インジェクト同時モードの組み合わせ

このモードは、ビット DUAL[4:0]=00001 をプログラムすることによって選択されます。

レギュラグループの同時変換を中断して、インジェクトグループの同時変換を開始することができます。

**注：** *レギュラ/インジェクト同時モードの組み合わせでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち変換時間が長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。*

### レギュラ同時 + オルタネートトリガモードの組み合わせ

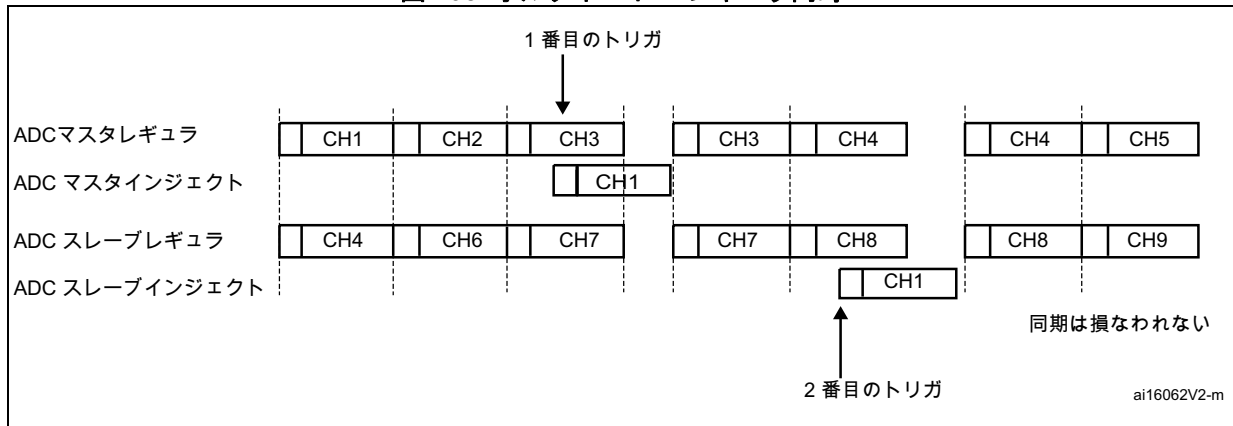
このモードは、ビット DUAL[4:0]=00010 をプログラムすることによって選択されます。

レギュラグループの同時変換を中断して、インジェクトグループのオルタネートトリガ変換を開始することができます。図 105 は、同時レギュラ変換に割り込むオルタネートトリガの動作を示します。

インジェクトオルタネート変換は、インジェクトイベント後、ただちに開始されます。すでにレギュラ変換が実行中の場合、インジェクト変換後の同期を確保するために、すべての (マスタ/スレーブ) ADC のレギュラ変換は停止し、インジェクト変換の終了と同期して再開されます。

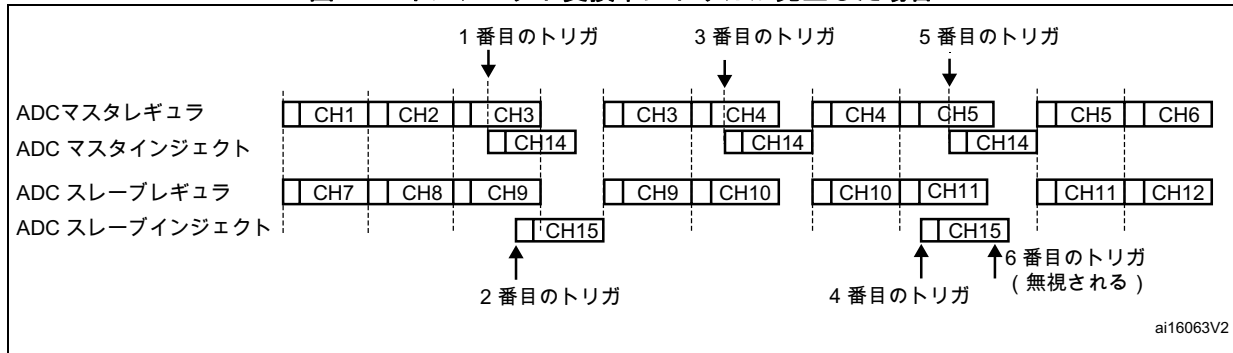
**注：** *レギュラ同時 + オルタネートトリガモードの組み合わせでは、同じ長さのシーケンスを変換するか、トリガ間隔が 2 つのシーケンスのうち変換時間が長い方のシーケンスより長くなるようにしなければなりません。そうしないと、長いシーケンスの ADC が前の変換を完了する前に、短いシーケンスの ADC がリスタートすることがあります。*

図 105. オルタネート + レギュラ同時



レギュラ変換に割り込んだインジェクト変換の最中にトリガが発生した場合はオルタネートトリガが処理されます。図 106に、この場合の挙動を示します（関連付けられたオルタネート変換は完了していないため、6番目のトリガは無視されます）。

図 106. インジェクト変換中にトリガが発生した場合

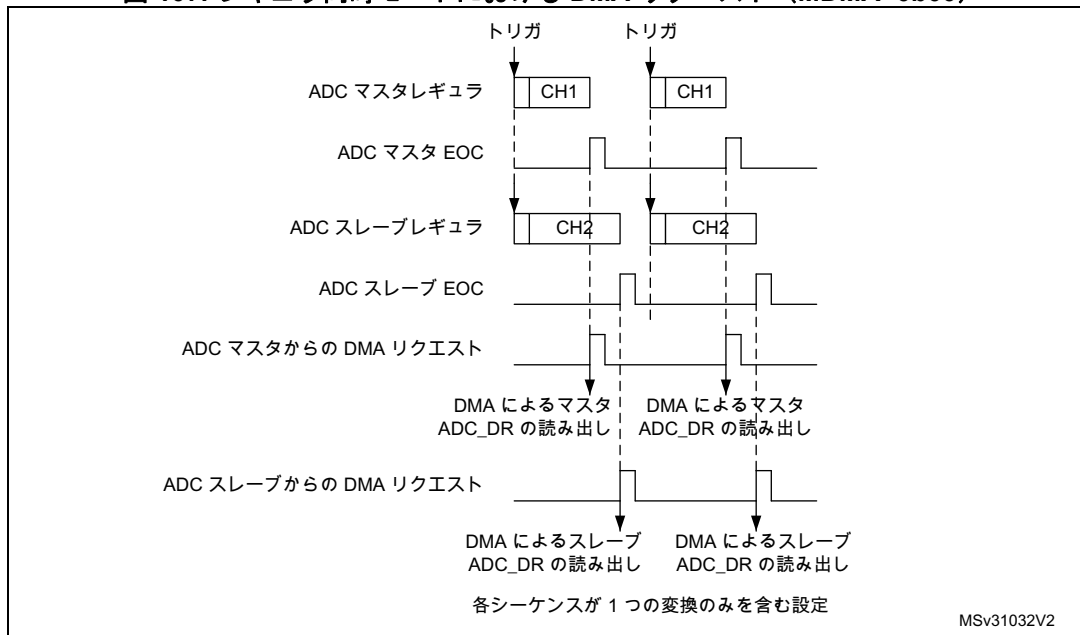


## デュアル ADC モードにおける DMA リクエスト

すべてのデュアル ADC モードでは、シングルモードの場合と同様、データを転送するために2つの DMA チャンネル（マスタ用1つとスレーブ用1つ）を使用することができます（図 107：レギュラ同

時モードにおける DMA リクエスト (MDMA=0b00) を参照)。

図 107. レギュラ同時モードにおける DMA リクエスト (MDMA=0b00)



レギュラ同時モードおよびインターリーブモードでは、1つのDMAチャンネルを確保し、シングルDMAチャンネルを使用して両方のデータを転送することもできます。そのためには、ADCx\_CCRレジスタでMDMAビットを設定する必要があります。

- **MDMA=0b10** : マスタおよびスレーブの両方の EOC イベントが発生するたびに DMA リクエストが 1 つ生成されます。このとき、2 つのデータアイテムが使用可能です。また、32 ビットレジスタ ADCx\_CDR には、2 つの ADC で変換されたデータアイテムを表す 2 つのハーフワードが含まれます。スレーブ ADC データは上位ハーフワードを使用し、マスタ ADC データは下位ハーフワードを使用します。  
このモードは、分解能が 10 ビットまたは 12 ビットの場合に、インターリーブモードおよびレギュラ同時モードで使用されます。

例 :

インターリーブデュアルモード: 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。

最初の DMA リクエスト: `ADCx_CDR[31:0] = SLV_ADCx_DR[15:0] | MST_ADCx_DR[15:0]`

2 番目の DMA リクエスト : `ADCx_CDR[31:0] = SLV_ADCx_DR[15:0] | MST_ADCx_DR[15:0]`

図 108. レギュラ同時モードにおける DMA リクエスト (MDMA=0b10)

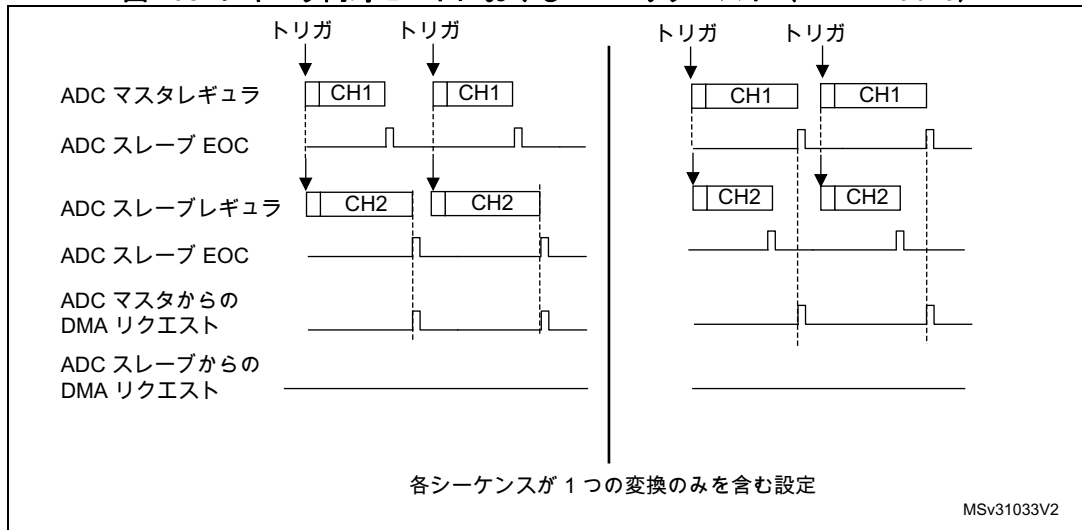
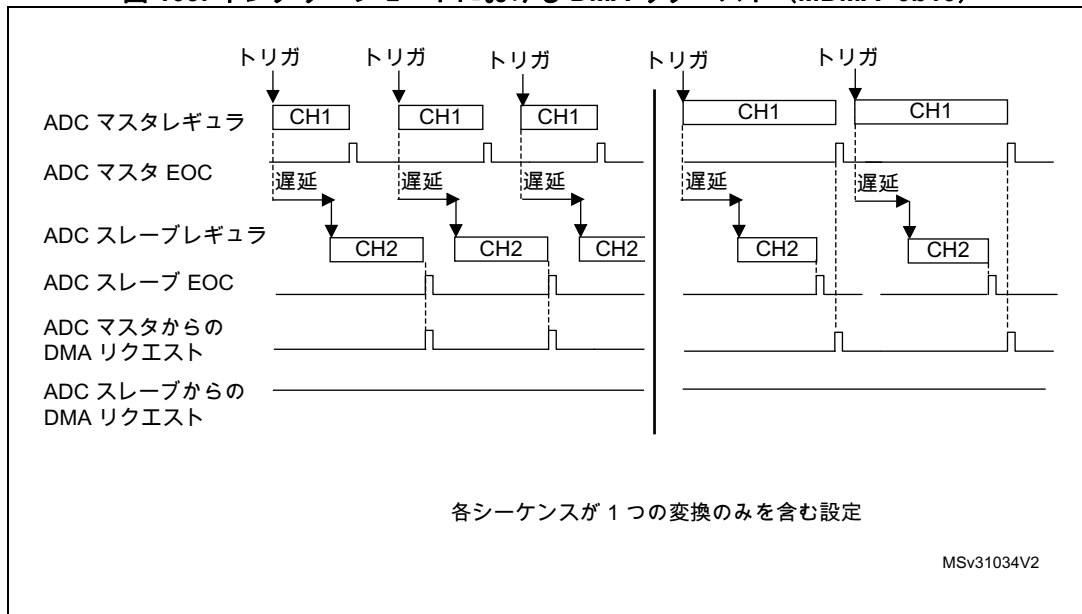


図 109. インタリーブモードにおける DMA リクエスト (MDMA=0b10)



**注：** MDMA モードを使用する場合は、マスタおよびスレーブ変換時間を適切に設定するよう注意する必要があります。これにより、新しい変換が使用可能になる前に DMA リクエストを生成して両方のデータ (マスタ + スレーブ) の読み出しを処理できます。

- **MDMA=0b11** : このモードは MDMA=0b10 に類似しています。唯一の違いは、DMA リクエストごとに (データ項目は 2 つ使用可能)、2 つの ADC 変換データを表す 2 バイトがハーフワードとして転送されます。

このモードは、分解能が 6 ビットまたは 8 ビットでデータが符号なしである場合に、インタリーブモードおよびレギュラ同時モードで使用されます (関連するすべてのチャンネルについて、オフセットは無効である必要があります)。

**例：**

インタリーブデュアルモード : 2 つのデータ項目が使用可能になるたびに DMA リクエストが 1 つ生成されます。

最初の DMA リクエスト : `ADCx_CDR[15:0] = SLV_ADCx_DR[7:0] | MST_ADCx_DR[7:0]`

2 番目の DMA リクエスト : `ADCx_CDR[15:0] = SLV_ADCx_DR[7:0] | MST_ADCx_DR[7:0]`

## オーバーラン検出

デュアル ADC モード (DUAL[4:0] が 0b00000 以外の場合) で、ADC のうちの 1 つにオーバーランが検出された場合、RAM に転送されたすべてのデータが有効であることを保証するため DMA リクエストの発行を停止します (この動作は MDMA の設定にかかわらず発生します)。ある ADC のデータレジスタに有効なデータが格納されていることによりその ADC に対応する EOC ビットがセットされたままになることがあります。

## MDMA モード選択時の DMA ワンショットモード/DMA サーキュラモード

MDMA モードが選択されている場合 (0b10 または 0b11)、[セクション : DMA を使用した変換の管理](#) で説明しているように、DMA ワンショットモード/サーキュラモードのどちらかを選択するために ADCx\_CCR レジスタの DMACFG ビット も設定する必要があります。(マスタおよびスレーブ ADCx\_CFGR の DMACFG ビットは無関係です)。

## デュアル ADC モードにおける変換の停止

デュアル ADC モードにおける ADC の変換を両方とも停止するには、マスタ ADC の制御ビットに ADSTP/JADSTP をセットする必要があります。デュアル ADC モードには、スレーブ ADC のその他の ADSTP 制御ビットの効果はありません。

両方の ADC が実質的に無効になると、マスタおよびスレーブ ADC の両方の ADSTART/JADSTART ビット がハードウェアによってクリアされます。

## 15.3.30 温度センサ

温度センサを使用して、デバイスの接合温度 (T<sub>J</sub>) を測定できます。温度センサは、センサの出力電圧をデジタル値に変換するために 入力チャンネルに内部接続されます。使用しないときには、センサをパワーダウンモードにすることができます。

[図 110](#) に、温度センサおよび ADC 間の接続のブロック図を示します。

温度センサの出力電圧は、温度に比例して変化します。このラインのオフセットは、プロセスのばらつきにより、チップごとに異なります (チップ間で最大 45 °C)。

較正されていない内部温度センサは、温度の絶対値の代わりに温度変化を検出するアプリケーションに適しています。温度センサの測定精度を高めるために、生産時に ST によって各デバイスの較正值がシステムメモリに格納されています。



製造プロセス中に、温度センサと内部電圧基準の較正データがシステムメモリ領域に格納されます。ユーザアプリケーションはこれらを読み出して、温度センサまたは内部基準の精度の向上に使用できます。詳細については、STM32F3xx データシートを参照してください。

## 主な特長

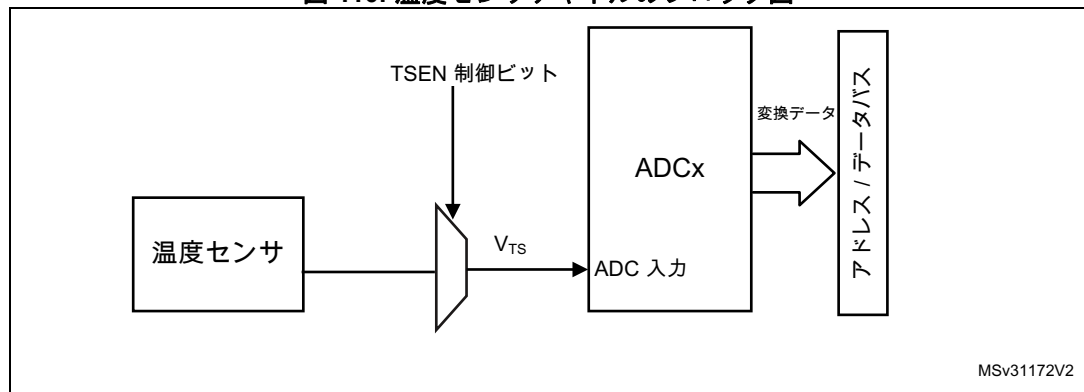
- サポートしている温度範囲：-40 ~ 125 °C
- 精度：±2 °C

温度センサは、センサの出力電圧をデジタル値に変換する ADC1\_IN16 入力チャンネルに内部接続されます。内部温度センサの変換時に適用するサンプリング時間の値については、STM32F3xx データシートの電気的特性のセクションを参照してください。

使用しないときには、センサをパワーダウンモードにすることができます。

図 110 に、温度センサのブロック図を示します。

図 110. 温度センサチャンネルのブロック図



注： 温度センサの電圧  $V_{TS}$  の変換を有効にするには、TSEN ビットをセットする必要があります。

## 温度の読み出し

温度センサを使用するには：

1. ADC1\_IN16 入力チャンネルを選択します (適切なサンプリング時間を使用)。
2. 適切なサンプリング時間をプログラムします (STM32F3xx データシートの電気的特性のセクションを参照)。
3. ADC1\_CCR レジスタの TSEN ビットをセットして、温度センサをパワーダウンモードからウェイクアップします。
4. ADC 変換を開始します。
5. ADC データレジスタの  $V_{TS}$  データを読み出します。
6. 次の式を使用して、実際の温度を計算します。

$$\text{温度 (摂氏)} = \{(V_{25} - V_{TS}) / \text{Avg\_Slope}\} + 25$$

ここで、

- $V_{25}$  = 25° C における  $V_{TS}$  値
- Avg\_Slope = 温度対  $V_{TS}$  曲線の平均傾斜 (mV/°C または  $\mu\text{V}/^\circ\text{C}$ )

$V_{25}$  および Avg\_Slope の実際の値については、データシートの電気的特性のセクションを参照してください。

注： センサがパワーダウンモードからウェイクアップして、正しいレベルで  $V_{TS}$  を出力できるようになるまでには時間がかかります (スタートアップ時間)。ADC にも起動後のスタートアップ時間があるので、遅延を最小にするには、ADEN ビットと TSEN ビットを同時にセットしてください。

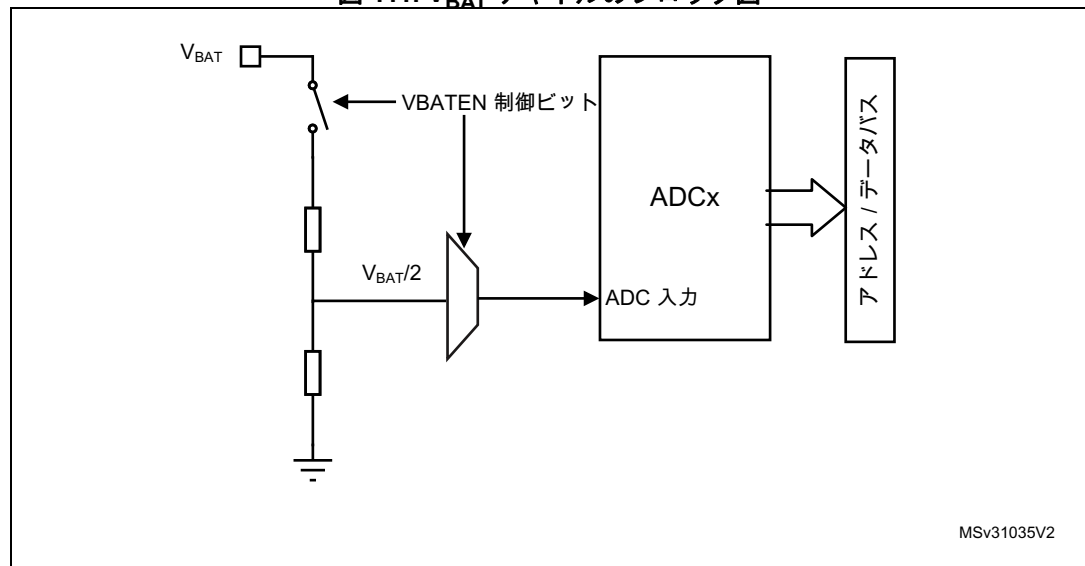
### 15.3.31 $V_{BAT}$ 電源監視

ADC12\_CCR レジスタの VBATEN ビットはバッテリー電圧に切り替えるために使用します。 $V_{BAT}$  電圧は  $V_{DDA}$  より高くなることがあるので、ADC の正しい動作を確保するために、 $V_{BAT}$  ピンはブリッジ分圧回路に内部接続されています。このブリッジは、VBATEN ビットがセットされると自動的に有効になり、 $V_{BAT}/2$  を ADC1\_IN17 入力チャンネルに接続します。結果として、変換されたデジタル値は  $V_{BAT}$  電圧の半分です。不要なバッテリー消費を避けるには、ADC 変換を行うときだけ、ブリッジ分圧回路を有効にすることが推奨されます。

$V_{BAT}/2$  電圧の変換時に適用するサンプリング時間の値については、STM32F3xx データシートの電気的特性を参照してください。

図 111 に、 $V_{BAT}$  センシング機能のブロック図を示します。

図 111.  $V_{BAT}$  チャンネルのブロック図



注： 内部チャンネル ADC1\_IN17 ( $V_{BATEN}$ ) の変換を有効にするには、VBATEN ビットをセットする必要があります。

### 15.3.32 内部電圧基準の監視

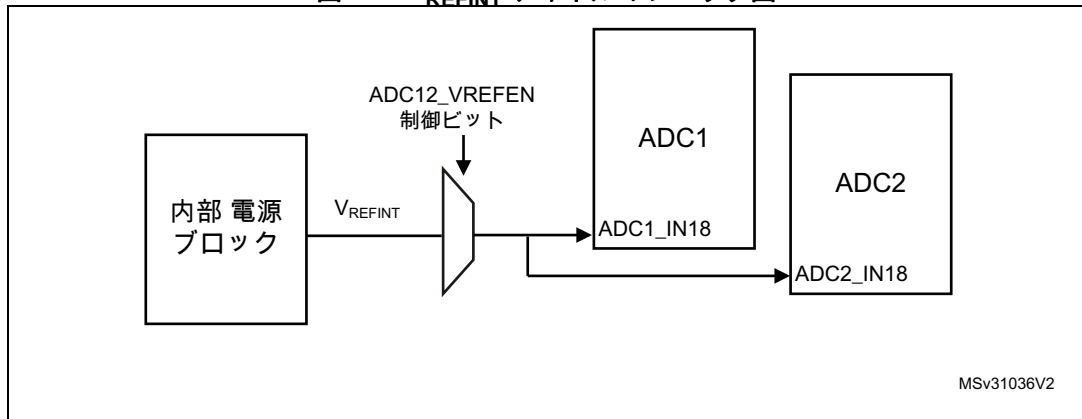
ADC  $V_{REF+}$  電圧レベルを評価するための基準点を持つために、内部電圧基準 ( $V_{REFINT}$ ) を監視することができます。

内部電圧基準は、4 つの ADC の入力チャンネル 18 (ADCx\_IN18) に内部的に接続されます。

内部電圧基準の変換時に適用するサンプリング時間の値については、STM32F3xx データシートの電気的特性のセクションを参照してください。

図 111 に、 $V_{REFINT}$  センシング機能のブロック図を示します。

図 112. V<sub>REFINT</sub> チャンネルのブロック図



注: 内部チャンネル ADC1\_IN18 または ADC2\_IN18 (V<sub>REFINT</sub>) の変換を有効にするには、ADC12\_CCR レジスタに VREFEN ビットをセットする必要があります。

内部チャンネル ADC3\_IN18 または ADC4\_IN18 (V<sub>REFINT</sub>) の変換を有効にするには、ADC34\_CCR レジスタに VREFEN ビットをセットする必要があります。

### 内部基準電圧を使用した実際の V<sub>D<sub>DA</sub></sub> 電圧の計算

マイクロコントローラに印加される V<sub>D<sub>DA</sub></sub> 電源電圧は、ばらつきがあり、または正確にはわかりません。組み込まれた内部電圧基準 (V<sub>REFINT</sub>) と、製造プロセス時に V<sub>D<sub>DA</sub></sub> = 3.3 V で ADC によって得られた較正データを使用して、実際の V<sub>D<sub>DA</sub></sub> 電圧レベルを評価することができます。

デバイスに印加される実際の V<sub>D<sub>DA</sub></sub> 電圧は、次の式で求められます。

$$V_{DDA} = 3.3 \text{ V} \times VREFINT\_CAL / VREFINT\_DATA$$

ここで、

- VREFINT\_CAL は、VREFINT の較正值です。
- VREFINT\_DATA は、ADC によって変換された実際の VREFINT の出力値です。

### 供給に相対的な ADC 測定値から絶対電圧値への変換

ADC は、アナログ電源と変換されるチャンネルに印加される電圧との比に対応するデジタル値を提供するように設計されています。ほとんどのアプリケーションの使用事例では、この比を V<sub>D<sub>DA</sub></sub> に依存しない電圧に変換する必要があります。V<sub>D<sub>DA</sub></sub> が既知であり、ADC によって変換された値が右詰めされるアプリケーションでは、次の式を使用して、この絶対値を求めることができます。

$$V_{CHANNELx} = \frac{V_{DDA}}{FULL\_SCALE} \times ADCx\_DATA$$

V<sub>D<sub>DA</sub></sub> 値が既知ではないアプリケーションの場合、内部電圧基準を使用する必要があります。V<sub>D<sub>DA</sub></sub> をセクション 内部基準電圧を使用した実際の V<sub>D<sub>DA</sub></sub> 電圧の計算 に記載されている式で置き換えることができます、次の式を使用できます。

$$V_{CHANNELx} = \frac{3.3 \text{ V} \times VREFINT\_CAL \times ADCx\_DATA}{VREFINT\_DATA \times FULL\_SCALE}$$

ここで、

- VREFINT\_CAL は、VREFINT の較正值です。
- ADCx\_DATA は、チャンネル x で ADC によって測定された値です (右詰め)。
- VREFINT\_DATA は、ADC によって変換された実際の VREFINT の出力値です。
- FULL\_SCALE は、ADC 出力の最大デジタル値です。たとえば、12 ビット分解能では  $2^{12} - 1 = 4095$  になり、8 ビット分解能では  $2^8 - 1 = 255$  になります。

**注：** ADC 測定が 12 ビット右詰め以外の出力形式を使用して行われる場合、計算を行う前に、すべてのパラメータを互換性のある形式に変換しておく必要があります。

## 15.4 ADC 割り込み

各 ADC に、割り込みを生成することができます。

- ADC パワーアップ後、ADC の準備ができたとき (ADRDY フラグ)
- レギュラグループの任意の変換終了時 (EOC フラグ)
- レギュラグループの任意の変換シーケンス終了時 (EOS フラグ)
- インジェクトグループの任意の変換終了時 (JEOC フラグ)
- インジェクトグループの任意の変換シーケンス終了時 (JEOS フラグ)
- アナログウォッチドッグ検出の発生時 (AWD1、AWD2、および AWD3 フラグ)
- サンプリングフェーズ終了の発生時 (EOSMP フラグ)
- データオーバーランの発生時 (OVR フラグ)
- インジェクトシーケンスのコンテキストキューのオーバーフロー時 (JQOVF フラグ)

高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 98. ADC ごとの ADC 割り込み

割り込みイベント	イベントフラグ	イネーブル制御ビット
ADC レディ	ADRDY	ADRDYIE
レギュラグループの変換終了	EOC	EOCIE
レギュラグループの変換シーケンスの終了	EOS	EOSIE
インジェクトグループの変換終了	JEOC	JEOCIE
インジェクトグループの変換シーケンスの終了	JEOS	JEOSIE
アナログウォッチドッグ 1 のステータスビットのセット	AWD1	AWD1IE
アナログウォッチドッグ 2 のステータスビットのセット	AWD2	AWD2IE
アナログウォッチドッグ 3 のステータスビットのセット	AWD3	AWD3IE
サンプリングフェーズの終了	EOSMP	EOSMPIE
オーバーラン	OVR	OVRIE
インジェクトコンテキストキューのオーバーフロー	JQOVF	JQOVFIE

## 15.5 ADC レジスタ (ADC ごと)

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

**注：** *STM32F303x6/8 および STM32F328x8 デバイスには、ADC1 と ADC2 のみがあります。*

### 15.5.1 ADC 割り込みおよびステータスレジスタ (ADCx\_ISR、x=1..4)

アドレスオフセット：0x00

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **JQOVF**：インジェクトコンテキストキューのオーバーフロー

このビットは、コンテキストのインジェクトキューのオーバーフローが発生したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。詳細については、[セクション 15.3.21：インジェクト変換のコンテキストのキュー](#) を参照してください。

0：インジェクトコンテキストキューのオーバーフローは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1：インジェクトコンテキストキューのオーバーフローが発生しました。

ビット 9 **AWD3**：アナログウォッチドッグ 3 フラグ

このビットは、変換された電圧が、ADCx\_TR3 レジスタのフィールド LT3[7:0] および HT3[7:0] でプログラムされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0：アナログウォッチドッグ 3 イベントは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1：アナログウォッチドッグ 3 イベントが発生しました。

ビット 8 **AWD2**：アナログウォッチドッグ 2 フラグ

このビットは、変換された電圧が、ADCx\_TR2 レジスタのフィールド LT2[7:0] および HT2[7:0] でプログラムされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0：アナログウォッチドッグ 2 イベントは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1：アナログウォッチドッグ 2 イベントが発生しました。

ビット 7 **AWD1**：アナログウォッチドッグ 1 フラグ

このビットは、変換された電圧が、ADCx\_TR1 レジスタのフィールド LT1[11:0] および HT1[11:0] でプログラムされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0：アナログウォッチドッグ 1 イベントは発生していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされています）。

1：アナログウォッチドッグ 1 イベントが発生しました。



**ビット 6 JEOS** : インジェクトチャネルのシーケンス終了フラグ

このビットは、グループ内のすべてのインジェクトチャネル変換が終了したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : インジェクト変換シーケンスは完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : インジェクト変換シーケンスが完了しました。

**ビット 5 JEOC** : インジェクトチャネルの変換終了フラグ

このビットは、チャネルの各インジェクト変換の終了時に、新しいデータが対応する ADCx\_JDRy レジスタで使用可能になったときに、ハードウェアによってセットされます。ソフトウェアによって 1 を書き込むことによって、または対応する ADCx\_JDRy レジスタを読み出すことによってクリアされます。

0 : インジェクトチャネル変換は完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : インジェクトチャネル変換が完了しました。

**ビット 4 OVR** : ADC オーバーラン

このビットは、レギュラチャネルでオーバーランが発生したときにハードウェアによってセットされ、EOC フラグがすでにセットされているときに新しい変換が完了したことを意味します。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : オーバーランは発生していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : オーバーランが発生しました。

**ビット 3 EOS** : レギュラシーケンス終了フラグ

このビットは、レギュラチャネルシーケンスの変換が終了したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : レギュラ変換シーケンスは完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : レギュラ変換シーケンスが完了しました。

**ビット 2 EOC** : 変換終了フラグ

このビットは、チャネルの各レギュラ変換の終了時に、新しいデータが ADCx\_DR レジスタで使用可能になったときに、ハードウェアによってセットされます。ソフトウェアによって 1 を書き込むことによって、または ADCx\_DR レジスタを読み出すことによってクリアされます。

0 : レギュラチャネル変換は完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : レギュラチャネル変換が完了しました。

**ビット 1 EOSMP** : サンプリング終了フラグ

このビットは、任意のチャネル (レギュラチャネルのみ) の変換中、サンプリングフェーズの終了時にハードウェアによってセットされます。

0 : サンプリングフェーズは終了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : サンプリングフェーズは終了しました。

**ビット 0 ADRDY** : ADC レディ

このビットは、ADC が有効にされた後 (ビット ADEN=1)、ADC が変換リクエストを受け入れる準備ができた状態に達したときに、ハードウェアによってセットされます。

ソフトウェアで 1 を書き込むことによってクリアされます。

0 : ADC はまだ変換を開始する準備ができていません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされています)。

1 : ADC は変換を開始する準備ができました。

## 15.5.2 ADC 割り込み有効レジスタ (ADCx\_IER, x=1..4)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQ	AWD3	AWD2	AWD1	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMP	ADRDY
					OVFIE	IE	IE	IE						IE	IE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **JQOVFIE** : インジェクトコンテキストキューのオーバーフロー割り込みイネーブル

このビットは、インジェクトコンテキストキューのオーバーフローの割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : インジェクトコンテキストキューのオーバーフロー割り込みは無効です。

1 : インジェクトコンテキストキューのオーバーフロー割り込みは有効です。JQOVF ビットがセットされると、割り込みが生成されます。

**注 :** ソフトウェアは、**JADSTART=0** である (インジェクト変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 9 **AWD3IE** : アナログウォッチドッグ 3 割り込みイネーブル

このビットは、アナログウォッチドッグ 3 割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : アナログウォッチドッグ 3 割り込みは無効です。

1 : アナログウォッチドッグ 3 割り込みは有効です。

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 8 **AWD2IE** : アナログウォッチドッグ 2 割り込みイネーブル

このビットは、アナログウォッチドッグ 2 割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : アナログウォッチドッグ 2 割り込みは無効です。

1 : アナログウォッチドッグ 2 割り込みは有効です。

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 7 **AWD1IE** : アナログウォッチドッグ 1 割り込みイネーブル

このビットは、アナログウォッチドッグ 1 割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : アナログウォッチドッグ 1 割り込みは無効です。

1 : アナログウォッチドッグ 1 割り込みは有効です。

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

- ビット 6 JEOSIE** : インジェクト変換シーケンス終了割り込みイネーブル  
このビットは、インジェクト変換シーケンス終了時の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : JEOS 割り込みは無効です。  
1 : JEOS 割り込みは有効です。JEOS ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**JADSTART=0** である (インジェクト変換が実行中でない) ときのみ、このビットに書き込むことができます。
- ビット 5 JEOCIE** : インジェクト変換終了割り込みイネーブル  
このビットは、インジェクト変換終了時の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : JEOC 割り込みは無効です。  
1 : JEOC 割り込みは有効です。JEOC ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**JADSTART=0** である (インジェクト変換が実行中でない) ときのみ、このビットに書き込むことができます。
- ビット 4 OVRIE** : オーバーラン割り込みイネーブル  
このビットは、レギュラ変換のオーバーラン割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : オーバーラン割り込みは無効です。  
1 : オーバーラン割り込みは有効です。OVR ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。
- ビット 3 EOSIE** : レギュラ変換シーケンス終了割り込みイネーブル  
このビットは、レギュラ変換シーケンス終了時の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : EOS 割り込みは無効です。  
1 : EOS 割り込みは有効です。EOS ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。
- ビット 2 EOCIE** : レギュラ変換終了割り込みイネーブル  
このビットは、レギュラ変換終了時の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : EOC 割り込みは無効です。  
1 : EOC 割り込みは有効です。EOC ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。
- ビット 1 EOSMIE** : レギュラ変換のサンプリング終了フラグ割り込みイネーブル  
このビットは、レギュラ変換のサンプリングフェーズ終了時の割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : EOSMP 割り込みは無効です。  
1 : EOSMP 割り込みは有効です。EOSMP ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。
- ビット 0 ADRDYIE** : ADC レディ割り込みイネーブル  
このビットは、ADC レディ割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。  
0 : ADRDY 割り込みは無効です。  
1 : ADRDY 割り込みは有効です。ADRDY ビットがセットされると、割り込みが生成されます。  
**注** : ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、このビットに書き込むことができます。



## 15.5.3 ADC 制御レジスタ (ADCx\_CR、x=1..4)

アドレスオフセット : 0x08

リセット値 : 0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD CAL	ADCA LDIF	ADVREGEN[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs	r/w	r/w	r/w												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JAD STP	AD STP	JAD START	AD START	AD DIS	AD EN
										rs	rs	rs	rs	rs	rs

### ビット 31 ADCAL : ADC 較正

このビットは、ADC の較正を開始するためにソフトウェアによってセットされます。最初にADCALDIF ビットをプログラムして、この較正がシングルエンドまたは差動入力モードに適用されるかどうかを確認します。

較正の完了後、ハードウェアによってクリアされます。

0 : 較正が完了しました。

1 : ADC を較正するには、1 を書き込みます。1 として読み出されたときには、較正が実行中であることを意味します。

**注 :** ソフトウェアは、ADEN=0 のときだけ ADCAL をセットして、較正を起動することができます。

**注 :** ソフトウェアは、ADEN=1 かつ ADSTART=0 かつ JADSTART=0 である (ADC が有効であり、変換中ではない) のときだけ、ADCx\_CALFACT を書き込むことによって較正係数を更新できます。

### ビット 30 ADCALDIF : 較正の差動モード

このビットは、較正のシングルエンド入力または差動入力モードを設定するために、ソフトウェアによってセット / クリアされます。

0 : ADCAL を書き込むと、シングルエンド入力モードで較正が起動されます。

1 : ADCAL を書き込むと、差動入力モードで較正が起動されます。

**注 :** ソフトウェアは、ADC が無効で較正されていないときだけ (ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0)、このビットを書き込むことができます。

### ビット 29:28 ADVREGEN[1:0] : ADC 電圧レギュレータイネーブル

これらのビットは、ADC 電圧レギュレータを有効にするために、ソフトウェアによってセットされます。

較正の起動や ADC の有効化などの操作を実行する前に、まずは ADC 電圧レギュレータを有効にし、ソフトウェアはレギュレータの起動時間まで待つ必要があります。

00 : ADC 電圧レギュレータの状態を有効から無効 (またはその逆) にする際に必要な中間状態です。

01 : ADC 電圧レギュレータは有効です。

10 : ADC 電圧レギュレータは無効です (リセット状態)。

11 : 予約済み

ADC 電圧レギュレータの有効化および無効化シーケンスの詳細については、[セクション 15.3.6 : ADC 電圧レギュレータ \(ADVREGEN\)](#) を参照してください。

**注 :** ソフトウェアは、ADC が無効のときのみ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0)、このビットフィールドをプログラムできます。

ビット 27:6 予約済みであり、リセット値に保持する必要があります。



### ビット 5 JADSTP : ADC インジェクト変換停止コマンド

このビットは、実行中のインジェクト変換を停止および破棄するためにソフトウェアによってセットされます (JADSTP コマンド)。

変換が効果的に破棄され、ADC インジェクトシーケンスとトリガを再設定できる場合に、ハードウェアによってクリアされます。これで、ADC は新しいインジェクト変換の開始 (JADSTART コマンド) を受け入れる準備ができます。

0 : ADC インジェクト変換停止コマンドは実行中ではありません。

1 : 実行中の ADC インジェクト変換を停止するには、1 を書き込みます。1 として読み出されたときには、ADSTP コマンドが実行中であることを意味します。

**注 :** ソフトウェアは、JADSTART=1 かつ ADDIS=0 である (ADC が有効であり、最終的にインジェクト変換が変換され、ADC を無効にする保留中のリクエストがない) ときだけ、JADSTP をセットできます。

**注 :** 自動インジェクションモード (JAUTO=1) では、ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアボートされます (JADSTP は使用しないでください)。

### ビット 4 ADSTP : ADC レギュラ変換停止コマンド

このビットは、実行中のレギュラ変換を停止および破棄するためにソフトウェアによってセットされます (ADSTP コマンド)。

変換が効果的に破棄され、ADC レギュラシーケンスとトリガを再設定できる場合に、ハードウェアによってクリアされます。これで、ADC は新しいレギュラ変換の開始 (ADSTART コマンド) を受け入れる準備ができます。

0 : ADC レギュラ変換停止コマンドは実行中ではありません。

1 : 実行中の ADC レギュラ変換を停止するには、1 を書き込みます。1 として読み出されたときには、ADSTP コマンドが実行中であることを意味します。

**注 :** ソフトウェアは、ADSTART=1 かつ ADDIS=0 である (ADC が有効であり、最終的にレギュラ変換が変換され、ADC を無効にする保留中のリクエストがない) ときだけ、ADSTP をセットできます。

**注 :** 自動インジェクションモード (JAUTO=1) では、ADSTP ビットをセットすると、レギュラ変換とインジェクト変換の両方がアボートされます (JADSTP は使用しないでください)。

**注 :** デュアル ADC レギュラ同時モードとインタリーブモードでは、レギュラ変換を停止するためにマスタ ADC の ADSTP ビットを使用する必要があります。その他の ADSTP ビットはインアクティブです。

### ビット 3 JADSTART : ADC インジェクト変換開始

このビットは、インジェクトチャネルの ADC 変換を開始するためにソフトウェアによってセットされます。設定ビット JEXTEN に応じて、変換はただちに開始されるか (ソフトウェアトリガ設定) またはインジェクトハードウェアトリガイベントが発生 (ハードウェアトリガ設定) したときに開始されます。

次のときに、ハードウェアによってクリアされます。

– シングル変換モードでは、ソフトウェアトリガが選択されたとき (JEXTSEL=0x0) : インジェクト変換シーケンス終了 (JEOS) フラグのアサート時。

– すべての場合 : JADSTP コマンドの実行後、JADSTP ビットがハードウェアによってクリアされると同時に。

0 : ADC インジェクト変換は実行中ではありません。

1 : インジェクト変換を開始するには、1 を書き込みます。1 として読み出されたときには、ADC が動作中であり、最終的にインジェクトチャネルが変換されることを意味します。

**注 :** ソフトウェアは、ADEN=1 かつ ADDIS=0 である (ADC が有効であり、ADC を無効にする保留中のリクエストがない) ときだけ、JADSTART をセットできます。

**注 :** 自動インジェクションモード (JAUTO=1) では、ADSTART ビットのセットによりレギュラおよび自動インジェクト変換を開始します (JADSTART はクリアされたままでなければなりません)。

## ビット 2 ADSTART : ADC レギュラ変換開始

このビットは、レギュラチャネルの ADC 変換を開始するためにソフトウェアによってセットされます。設定ビット EXTEN に応じて、変換はただちに開始されるか (ソフトウェアトリガ設定) またはレギュラハードウェアトリガイベントが発生したときに (ハードウェアトリガ設定) 開始されます。

次のときに、ハードウェアによってクリアされます。

– シングル変換モードでは、ソフトウェアトリガが選択されたとき (EXTSEL=0x0) : レギュラ変換シーケンス終了 (EOS) フラグのアサート時。

– すべての場合 : ADSTP コマンドの実行後、ADSTP ビットがハードウェアによってクリアされると同時に。

0 : ADC レギュラ変換は実行中ではありません。

1 : レギュラ変換を開始するには、1 を書き込みます。1 として読み出されたときには、ADC が動作中であり、最終的にレギュラチャネルが変換されることを意味します。

**注 :** ソフトウェアは、ADEN=1 かつ ADDIS=0 (ADC が有効であり、ADC を無効にする保留中のリクエストがない) のときだけ、ADSTART をセットできます。

**注 :** 自動インジェクションモード (JAUTO=1) では、設定ビット ADSTART を使用してレギュラおよび自動インジェクト変換を開始します (JADSTART はクリアされたままでなければなりません)。

## ビット 1 ADDIS : ADC 無効化コマンド

このビットは、ADC を無効にして (ADDIS コマンド)、パワーダウン状態 (OFF 状態) にするためにソフトウェアによってセットされます。

ADC が効果的に無効化されると、ハードウェアによってクリアされます (ADEN もこの時点でハードウェアによってクリアされます)。

0 : ADDIS コマンドは実行中ではありません。

1 : ADC を無効にするには、1 を書き込みます。1 として読み出されたときには、ADDIS コマンドが実行中であることを意味します。

**注 :** ソフトウェアは、ADEN=1 で ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) のときのみ、ADDIS をセットできます。

## ビット 0 ADEN : ADC イネーブル制御

このビットは、ADC を有効にするために、ソフトウェアによってセットされます。フラグ ADRDY がセットされると、ADC は動作する準備ができています。

ADDIS コマンドの実行後、ADC が無効になるとハードウェアによってクリアされます。

0 : ADC は無効です (OFF 状態)。

1 : ADC を有効にするには、1 を書き込みます。

**注 :** ソフトウェアは、1 にセットする必要がある ADVREGEN ビットを除き、ADCx\_CR レジスタのすべてのビットが 0 のとき (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0) だけ、ADEN をセットできます (また、ソフトウェアは、電圧レギュレータの起動時間まで待っていないければなりません)。

## 15.5.4 ADC 設定レジスタ (ADCx\_CFGR, x=1..4)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWD1CH[4:0]					JAUTO	JAWD1 EN	AWD1 EN	AWD1S GL	JQM	JDISC EN	DISCNUM[2:0]			DISC EN
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AUT DLY	CONT	OVR MOD	EXTEN[1:0]		EXTSEL[3:0]			ALIGN	RES[1:0]		Res.	DMA CFG	DMA EN	
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:26 **AWD1CH[4:0]** : アナログウォッチドッグ 1 チャンネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグによって保護される入力チャンネルを選択します。

00000 : 予約済み (アナログ入力チャンネル 0 は未配置)

00001 : AWD1 によって監視されている ADC アナログ入力チャンネル 1

.....

10010 : AWD1 によって監視されている ADC アナログ入力チャンネル 18

その他。予約済み。使用できません。

**注 :** **AWD1CH** で選択したチャンネルは、**SQRi** または **JSQRi** レジスタでも選択する必要があります。

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 25 **JAUTO** : 自動インジェクトグループ変換

このビットは、レギュラグループ変換の後、自動インジェクトグループ変換を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : 自動インジェクトグループ変換は無効です。

1 : 自動インジェクトグループ変換は有効です。

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (レギュラ変換とインジェクト変換がどちらも実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアルモードが有効 (**ADCx\_CCR** レジスタの **DUAL** ビットが 0 以外) である場合、スレーブ ADC の **JAUTO** ビットへの書き込みができなくなり、その内容はマスタ ADC の **JAUTO** ビットと等しくなります。

ビット 24 **JAWD1EN** : アナログウォッチドッグ 1 有効 (インジェクトチャンネル)

このビットは、ソフトウェアによってセット/クリアされます。

0 : インジェクトチャンネルのアナログウォッチドッグ 1 は無効です。

1 : インジェクトチャンネルのアナログウォッチドッグ 1 は有効です。

**注 :** ソフトウェアは、**JADSTART=0** (インジェクト変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 23 **AWD1EN** : アナログウォッチドッグ 1 有効 (レギュラチャンネル)

このビットは、ソフトウェアによってセット/クリアされます。

0 : レギュラチャンネルのアナログウォッチドッグ 1 は無効です。

1 : レギュラチャンネルのアナログウォッチドッグ 1 は有効です。

**注 :** ソフトウェアは、**ADSTART=0** (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 22 **AWD1SGL** : ウォッチドッグ 1 を単一チャンネルまたはすべてのチャンネルで有効にします。

このビットは、AWD1CH[4:0] ビットによって指定されたチャンネルまたはすべてのチャンネルに対するアナログウォッチドッグを有効にするために、ソフトウェアによってセット/クリアされます。

0 : すべてのチャンネルでアナログウォッチドッグ 1 は有効です。

1 : 単一チャンネルでアナログウォッチドッグ 1 は有効です。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 21 **JQM** : JSQR キューモード

このビットは、ソフトウェアによってセット/クリアされます。

空のキューの管理方法について定義します。

0 : JSQR モード 0 : キューは空にならず、JSQR に最後に書き込んだ設定を保持します。

1 : JSQR モード 1 : キューは空である場合もありますが、このとき、インジェクトシーケンスのソフトウェア/ハードウェアトリガは、最後の有効なインジェクトシーケンスの完了後、すぐに両方とも内部的に無効化されます。

詳細については、[セクション 15.3.21 : インジェクト変換のコンテキストのキュー](#)を参照してください。

**注 :** ソフトウェアは、JADSTART=0 (インジェクト変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアルモードが有効 (ADCx\_CCR レジスタの DUAL ビットが 0 以外) である場合、スレーブ ADC の JQM ビットへの書き込みができなくなり、その内容はマスタ ADC の JQM ビットと等しくなります。

ビット 20 **JDISCEN** : インジェクトチャンネルの不連続モード

このビットは、グループのインジェクトチャンネルに対する不連続モードを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : インジェクトチャンネルの不連続モードは無効です。

1 : インジェクトチャンネルの不連続モードは有効です。

**注 :** ソフトウェアは、JADSTART=0 (インジェクト変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** 自動インジェクトモードと不連続モードの両方を同時に使用することはできません。JAUTO のセット時には、ビット DISCEN と JDISCEN はクリアされたままでなければなりません。

**注 :** デュアルモードが有効 (ADCx\_CCR レジスタの DUAL ビットが 0 以外) である場合、スレーブ ADC の JDISCEN ビットへの書き込みができなくなり、その内容はマスタ ADC の JDISCEN ビットと等しくなります。

ビット 19:17 **DISCNUM[2:0]** : 不連続モードチャンネルカウント

これらのビットは、外部トリガの受信後に、不連続モードで変換されるレギュラチャンネルの数を定義するために、ソフトウェアによって書き込まれます。

000 : 1 チャンネル

001 : 2 チャンネル

...

111 : 8 チャンネル

**注 :** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注 :** デュアルモードが有効 (ADCx\_CCR レジスタの DUAL ビットが 0 以外) である場合、スレーブ ADC の DISCNUM[2:0] ビットへの書き込みができなくなり、その内容はマスタ ADC の DISCNUM[2:0] ビットと等しくなります。

**ビット 16 DISCEN** : レギュラチャネルの不連続モード

このビットは、レギュラチャネルの不連続モードを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : レギュラチャネルの不連続モードは無効です。
- 1 : レギュラチャネルの不連続モードは有効です。

**注 :** 不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方をセットすることは禁じられています。

**注 :** 自動インジェクトモードと不連続モードの両方を同時に使用することはできません。JAUTO のセット時には、ビット DISCEN と JDISCEN はクリアされたままでなければなりません。

**注 :** ソフトウェアは、ADSTART=0 (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアルモードが有効 (ADCx\_CCR レジスタのDUALビットが0以外) である場合、スレーブADCのDISCENビットへの書き込みができなくなり、その内容はマスタADCのDISCENビットと等しくなります。

ビット 15 予約済みであり、リセット値に保持する必要があります。

**ビット 14 AUTDLY** : 遅延変換モード

このビットは、自動遅延変換モードを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : 自動遅延変換モードはオフです。
- 1 : 自動遅延変換モードはオンです。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアルモードが有効 (ADCx\_CCR レジスタのDUALビットが0以外) である場合、スレーブADCのAUTDLYビットへの書き込みができなくなり、その内容はマスタADCのAUTDLYビットと等しくなります。

**ビット 13 CONT** : レギュラ変換のシングル/連続変換モード

このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされた場合、それがクリアされるまで連続的にレギュラ変換が行われます。

- 0 : シングル変換モード
- 1 : 連続変換モード

**注 :** 不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方をセットすることは禁じられています。

**注 :** ソフトウェアは、ADSTART=0 (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアルモードが有効 (ADCx\_CCR レジスタのDUALビットが0以外) である場合、スレーブADCのCONTビットへの書き込みができなくなり、その内容はマスタADCのCONTビットと等しくなります。

**ビット 12 OVRMOD** : オーバーランモード

このビットは、ソフトウェアによってセット/クリアされ、データオーバーランを管理する方法を設定します。

- 0 : オーバーランが検出されたとき、ADCx\_DR レジスタの古いデータが保存されます。
- 1 : オーバーランが検出されたとき、ADCx\_DR レジスタは最後の変換結果で上書きされます。

**注 :** ソフトウェアは、ADSTART=0 (レギュラ変換が実行中でない) ときのみ、このビットに書き込むことができます。

**ビット 11:10 EXTEN[1:0]** : レギュラチャネルの外部トリガ有効および極性選択

これらのビットは、外部トリガ極性を選択しレギュラグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

- 00 : ハードウェアトリガ検出は無効です (変換はソフトウェアによって起動できます)。
- 01 : 立ち上がりエッジでハードウェアトリガを検出します。
- 10 : 立ち下がりエッジでハードウェアトリガを検出します。
- 11 : 立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

**注 :** ソフトウェアは、ADSTART=0 (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**ビット 9:6 EXTSEL[3:0]** : レギュラグループの外部トリガ選択

これらのビットは、レギュラグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : イベント 0  
0001 : イベント 1  
0010 : イベント 2  
0011 : イベント 3  
0100 : イベント 4  
0101 : イベント 5  
0110 : イベント 6  
0111 : イベント 7  
...  
1111 : イベント 15

**注 :** ソフトウェアは、ADSTART=0 (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**ビット 5 ALIGN** : データの配置

このビットは、右詰めまたは左詰めを選択するために、ソフトウェアによってセット/クリアされます。[図 : データレジスタ、データの配置およびデータオフセット \(ADCx\\_DR、OFFSETy、OFFSETy\\_CH、ALIGN\)](#) を参照してください。

0 : 右詰め  
1 : 左詰め

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

**ビット 4:3 RES[1:0]** : データ分解能

これらのビットは、変換の分解能を選択するために、ソフトウェアによって書き込まれます。

00 : 12 ビット  
01 : 10 ビット  
10 : 8 ビット  
11 : 6 ビット

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **DMACFG** : ダイレクトメモリアクセス設定

このビットは、2つの DMA 動作モードを選択するためにソフトウェアによってセット/クリアされ、DMAEN=1 のときのみ有効です。

0 : DMA ワンショットモードが選択されています。

1 : DMA サーキュラモードが選択されています。

詳細については、[セクション : DMA を使用した変換の管理](#)を参照してください。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアル ADC モードでは、このビットは無関係で、ADCx\_CCR レジスタの制御ビット DMACFG に置き換えられます。

ビット 0 **DMAEN** : ダイレクトメモリアクセス有効

このビットは、DMA リクエストの生成を有効にするために、ソフトウェアによってセット/クリアされます。これにより、GP-DMA を使用して変換データを自動的に管理できます。詳細については、[セクション : DMA を使用した変換の管理](#)を参照してください。

0 : DMA は無効です。

1 : DMA は有効です。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

**注 :** デュアル ADC モードでは、このビットは無関係で、ADCx\_CCR レジスタの制御ビット MDMA[1:0] に置き換えられます。

## 15.5.5 ADC サンプル時間レジスタ 1 (ADCx\_SMPR1、x=1..4)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			Res.	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w			



ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:3 **SMPx[2:0]** : チャネル x サンプリング時間選択

これらのビットは、各チャネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプルサイクル中は、チャネル選択ビットを変更することはできません。

000 : 1.5 ADC クロックサイクル

001 : 2.5 ADC クロックサイクル

010 : 4.5 ADC クロックサイクル

011 : 7.5 ADC クロックサイクル

100 : 19.5 ADC クロックサイクル

101 : 61.5 ADC クロックサイクル

110 : 181.5 ADC クロックサイクル

111 : 601.5 ADC クロックサイクル

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 2:0 予約済み

## 15.5.6 ADC サンプル時間レジスタ 2 (ADCx\_SMPR2, x=1..4)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15_0		SMP14[2:0]		SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26:0 **SMPx[2:0]** : チャネル x サンプリング時間選択

これらのビットは、各チャンネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプリングサイクル中は、チャンネル選択ビットを変更することはできません。

- 000 : 1.5 ADC クロックサイクル
- 001 : 2.5 ADC クロックサイクル
- 010 : 4.5 ADC クロックサイクル
- 011 : 7.5 ADC クロックサイクル
- 100 : 19.5 ADC クロックサイクル
- 101 : 61.5 ADC クロックサイクル
- 110 : 181.5 ADC クロックサイクル
- 111 : 601.5 ADC クロックサイクル

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

## 15.5.7 ADC ウォッチドッグ閾値レジスタ 1 (ADCx\_TR1, x=1..4)

アドレスオフセット : 0x20

リセット値 : 0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1[11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT1[11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 HT1[11:0]: アナログウォッチドッグ 1 高閾値

これらのビットは、アナログウォッチドッグ 1 の高閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 15.3.28: アナログウィンドウウォッチドッグ \(AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD\\_HTx, AWD\\_LTx, AWDx\)](#) を参照してください。

**注:** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 LT1[11:0]: アナログウォッチドッグ 1 低閾値

これらのビットは、アナログウォッチドッグ 1 の低閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 15.3.28: アナログウィンドウウォッチドッグ \(AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD\\_HTx, AWD\\_LTx, AWDx\)](#) を参照してください。

**注:** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

## 15.5.8 ADC ウォッチドッグ閾値レジスタ 2 (ADCx\_TR2, x = 1..4)

アドレスオフセット: 0x24

リセット値: 0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[7:0]							
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 HT2[7:0]: アナログウォッチドッグ 2 高閾値

これらのビットは、アナログウォッチドッグ 2 の高閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 15.3.28: アナログウィンドウウォッチドッグ \(AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD\\_HTx, AWD\\_LTx, AWDx\)](#) を参照してください。

**注:** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 LT2[7:0]: アナログウォッチドッグ 2 低閾値

これらのビットは、アナログウォッチドッグ 2 の低閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 15.3.28: アナログウィンドウウォッチドッグ \(AWD1EN, JAWD1EN, AWD1SGL, AWD1CH, AWD2CH, AWD3CH, AWD\\_HTx, AWD\\_LTx, AWDx\)](#) を参照してください。

**注:** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

## 15.5.9 ADC ウォッチドッグ閾値レジスタ 3 (ADCx\_TR3, x=1..4)

アドレスオフセット : 0x28

リセット値 : 0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **HT3[7:0]** : アナログウォッチドッグ 3 高閾値

これらのビットは、アナログウォッチドッグ 3 の高閾値を定義するために、ソフトウェアによって書き込まれます。

[セクション 15.3.28 : アナログウィンドウウォッチドッグ \(AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD\\_HTx、AWD\\_LTx、AWDx\)](#) を参照してください。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **LT3[7:0]** : アナログウォッチドッグ 3 低閾値

これらのビットは、アナログウォッチドッグ 3 の低閾値を定義するために、ソフトウェアによって書き込まれます。

このウォッチドッグでは、LT3 の 8 ビットを変換データの最上位の 8 ビットと比較します。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

## 15.5.10 ADC レギュラシーケンスレジスタ 1 (ADCx\_SQR1, x=1..4)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w		r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ2[3:0]			Res.	SQ1[4:0]					Res.	Res.	L[3:0]				
r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w	r/w

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **SQ4[4:0]** : レギュラシーケンスの 4 番目の変換

これらのビットには、レギュラ変換シーケンス内の 4 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注 :** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注 :** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **SQ3[4:0]** : レギュラシーケンスの 3 番目の変換

これらのビットには、レギュラ変換シーケンス内の 3 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注 :** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注 :** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16:12 **SQ2[4:0]** : レギュラシーケンスの 2 番目の変換

これらのビットには、レギュラ変換シーケンス内の 2 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注 :** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注 :** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ1[4:0]** : レギュラシーケンスの 1 番目の変換

これらのビットには、レギュラ変換シーケンス内の 1 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 5:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **L[3:0]** : レギュラチャンネルシーケンス長

これらのビットは、レギュラチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

0000 : 1 変換

0001 : 2 変換

...

1111 : 16 変換

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

## 15.5.11 ADC レギュラシーケンスレジスタ 2 (ADCx\_SQR2, x=1..4)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4]
			r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w		r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ7[3:0]			Res.	SQ6[4:0]					Res.	SQ5[4:0]					
r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **SQ9[4:0]** : レギュラシーケンスの 9 番目の変換

これらのビットには、レギュラ変換シーケンス内の 9 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **SQ8[4:0]** : レギュラシーケンスの 8 番目の変換

これらのビットには、レギュラ変換シーケンス内の 8 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16:12 **SQ7[4:0]** : レギュラシーケンスの 7 番目の変換

これらのビットには、レギュラ変換シーケンス内の 7 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ6[4:0]** : レギュラシーケンスの 6 番目の変換

これらのビットには、レギュラ変換シーケンス内の 6 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **SQ5[4:0]** : レギュラシーケンスの 5 番目の変換

これらのビットには、レギュラ変換シーケンス内の 5 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、**ADSTART=0** である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

## 15.5.12 ADC レギュラシーケンスレジスタ 3 (ADCx\_SQR3, x=1..4)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4]
			r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w		r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ12[3:0]			Res.	SQ11[4:0]					Res.	SQ10[4:0]					
r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28:24 **SQ14[4:0]** : レギュラシーケンスの 14 番目の変換

これらのビットには、レギュラ変換シーケンス内の 14 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:18 **SQ13[4:0]** : レギュラシーケンスの 13 番目の変換

これらのビットには、レギュラ変換シーケンス内の 13 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16:12 **SQ12[4:0]** : レギュラシーケンスの 12 番目の変換

これらのビットには、レギュラ変換シーケンス内の 12 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ11[4:0]** : レギュラシーケンスの 11 番目の変換

これらのビットには、レギュラ変換シーケンス内の 11 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **SQ10[4:0]** : レギュラシーケンスの 10 番目の変換

これらのビットには、レギュラ変換シーケンス内の 10 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。



## 15.5.13 ADC レギュラシーケンスレジスタ 4 (ADCx\_SQR4, x=1..4)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
					r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w

ビット 31:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **SQ16[4:0]** : レギュラシーケンスの 16 番目の変換

これらのビットには、レギュラ変換シーケンス内の 16 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **SQ15[4:0]** : レギュラシーケンスの 15 番目の変換

これらのビットには、レギュラ変換シーケンス内の 15 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

## 15.5.14 ADC レギュラデータレジスタ (ADCx\_DR、x=1..4)

アドレスオフセット : 0x40

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RDATA[15:0]** : レギュラデータ変換

これらのビットは読み出し専用です。これらは、最後の変換されたレギュラチャネルの変換結果を含んでいます。  
[セクション 15.3.26 : データ管理](#) に示すように、データは左詰めまたは右詰めされています。

## 15.5.15 ADC インジェクトシーケンスレジスタ (ADCx\_JSQR、x=1..4)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:2]			
	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
JSQ2[1:0]		Res.	JSQ1[4:0]				JEXTEN[1:0]			JEXTSEL[3:0]			JL[1:0]			
rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:26 JSQ4[4:0] : インジェクトシーケンスの 4 番目の変換

これらのビットには、インジェクト変換シーケンス内の 4 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 25 予約済みであり、リセット値に保持する必要があります。

ビット 24:20 JSQ3[4:0] : インジェクトシーケンスの 3 番目の変換

これらのビットには、インジェクト変換シーケンス内の 3 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18:14 JSQ2[4:0] : インジェクトシーケンスの 2 番目の変換

これらのビットには、インジェクト変換シーケンス内の 2 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 JSQ1[4:0] : インジェクトシーケンスの 1 番目の変換

これらのビットには、インジェクト変換シーケンス内の 1 番目として割り当てられたチャンネル番号 (1..18) がソフトウェアによって書き込まれます。

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

**注:** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

### ビット 7:6 JEXTEN[1:0]: インジェクトチャネルの外部トリガ有効および極性選択

これらのビットは、外部トリガ極性を選択しインジェクトグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00: ハードウェアトリガ検出は無効です (変換はソフトウェアによって起動できます)。

01: 立ち上がりエッジでハードウェアトリガを検出します。

10: 立ち下がりエッジでハードウェアトリガを検出します。

11: 立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

**注:** JQM=1 で、コンテキストのキューが空である場合、インジェクトシーケンスのソフトウェアおよびハードウェアトリガは両方とも内部的に無効化されます (セクション 15.3.21: インジェクト変換のコンテキストのキューを参照)。

### ビット 5:2 JEXTSEL[3:0]: インジェクトグループの外部トリガ選択

これらのビットは、インジェクトグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000: イベント 0

0001: イベント 1

0010: イベント 2

0011: イベント 3

0100: イベント 4

0101: イベント 5

0110: イベント 6

0111: イベント 7

...

1111: イベント 15

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

### ビット 1:0 JL[1:0]: インジェクトチャネルシーケンス長

これらのビットは、インジェクトチャネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

00: 1 変換

01: 2 変換

10: 3 変換

11: 4 変換

**注:** ソフトウェアは、ADC が有効である場合 (ADEN=1)、いつでもこれらのビットを書き込むことができます。

## 15.5.16 ADC オフセットレジスタ (ADCx\_OFBy, x=1..4) (y=1..4)

アドレスオフセット : 0x60, 0x64, 0x68, 0x6C

リセット値 : 0x0000 0000

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

ビット 31 **OFFSETy\_EN** : オフセット y イネーブル

このビットは、OFFSETy[11:0] ビットにプログラムされたオフセットを有効または無効にするために、ソフトウェアによって書き込まれます。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 30:26 **OFFSETy\_CH[4:0]** : データオフセット y のチャンネル選択

このビットは、OFFSETy[11:0] ビットにプログラムされたオフセットが適用されるチャンネルを定義するために、ソフトウェアによって書き込まれます。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

**注 :** アナログ入力チャンネル 0 は配置されていません。値「00000」は使用すべきではありません。

ビット 25:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **OFFSETy[11:0]** : OFFSETy\_CH[4:0] ビットにプログラムされたチャンネルのデータオフセット y

これらのビットは、チャンネル (レギュラまたはインジェクトのどちらか) を変換するときに元の変換データから引かれるオフセット y を定義するために、ソフトウェアによって書き込まれます。データオフセット y を適用するチャンネルは、ビット OFFSETy\_CH[4:0] ビットにプログラムする必要があります。変換結果は、ADCx\_DR レジスタ (レギュラ変換) または ADCx\_JDRyi レジスタ (インジェクト変換) から読み出すことができます。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

**注 :** 複数のオフセット (OFFSETy) が同じチャンネルを指している場合、減算時には最小の x を持つオフセットのみが考慮されます。

**例 :** OFFSET1\_CH[4:0]=4 かつ OFFSET2\_CH[4:0]=4 である場合、これはチャンネル 4 を変換するときに減算される OFFSET1[11:0] です。

## 15.5.17 ADC インジェクトデータレジスタ (ADCx\_JDRy, x=1..4, y= 1..4)

アドレスオフセット : 0x80 - 0x8C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **JDATA[15:0]** : インジェクトデータ

これらのビットは読み出し専用です。これらのビットには、インジェクトチャンネル y の変換結果が格納されます。[セクション 15.3.26: データ管理](#) に示すように、データは左詰めまたは右詰めされています。

## 15.5.18 ADC アナログウォッチドッグ 2 設定レジスタ (ADCx\_AWD2CR, x=1..4)

アドレスオフセット : 0xA0

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:16]		
													r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CH[15:1]															Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:1 **AWD2CH[18:1]** : アナログウォッチドッグ 2 チャンネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグ 2 によって保護される入力チャンネルを有効化して選択します。

AWD2CH[i] = 0 : ADC アナログ入力チャンネル i は AWD2 によって監視されません。

AWD2CH[i] = 1 : ADC アナログ入力チャンネル i は AWD2 によって監視されます。

AWD2CH[18:1] = 000..0 である場合、アナログウォッチドッグ 2 は無効になります。

**注 :** **AWD2CH** で選択したチャンネルは、**SQRi** または **JSQRi** レジスタでも選択する必要があります。

**注 :** ソフトウェアは、**ADSTART=0** かつ **JADSTART=0** である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

## 15.5.19 ADC アナログウォッチドッグ 3 設定レジスタ (ADCx\_AWD3CR、x=1..4)

アドレスオフセット : 0xA4

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:16]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3CH[15:1]															Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:1 **AWD3CH[18:1]** : アナログウォッチドッグ 3 チャンネル選択

これらのビットは、ソフトウェアによってセット/クリアされます。アナログウォッチドッグ 3 によって保護される入力チャンネルを有効化して選択します。

AWD3CH[i] = 0 : ADC アナログ入力チャンネル i は AWD3 によって監視されません。

AWD3CH[i] = 1 : ADC アナログ入力チャンネル i は AWD3 によって監視されます。

AWD3CH[18:1] = 000..0 である場合、アナログウォッチドッグ 3 は無効になります。

**注 :** AWD3CH で選択したチャンネルは、SQRi または JSQRi レジスタでも選択する必要があります。

**注 :** ソフトウェアは、ADSTART=0 かつ JADSTART=0 である (変換が実行中でない) ときのみ、これらのビットに書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。

## 15.5.20 ADC 差動モード選択レジスタ (ADCx\_DIFSEL、x=1..4)

アドレスオフセット : 0xB0

リセット値 : 0x0000 0000

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



ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:16 **DIFSEL[18:16]**: チャンネル (18 ~ 16) の差動モード

これらのビットは読み出し専用です。これらのチャンネルは強制的にシングルエンド入力モード (シングルエンド I/O ポートまたは内部チャンネルのどちらかに接続) になります。

ビット 15:1 **DIFSEL[15:1]**: チャンネル (15 ~ 1) の差動モード

これらのビットは、ソフトウェアによってセット/クリアされます。これにより、チャンネルの設定をシングルエンドモードまたは差動モードから選択できます。

DIFSEL[i] = 0 : ADC アナログ入力チャンネル i はシングルエンドモードで設定されます。

DIFSEL[i] = 1 : ADC アナログ入力チャンネル i は差動モードで設定されます。

**注:** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0)、これらのビットを書き込むことができます。

**注:** ADC1\_DIFSEL[15] (内部シングルエンドチャンネルに接続) はクリアされたままにする必要があります。

ビット 0 予約済みであり、リセット値に保持する必要があります。

### 15.5.21 ADC 較正係数 (ADCx\_CALFACT、x=1..4)

アドレスオフセット: 0xB4

リセット値: 0x0000 0000

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

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:16 **CALFACT\_D[6:0]**: 差動モードでの較正係数

これらのビットは、ハードウェアまたはソフトウェアによって書き込まれます。

差動入力較正が完了すると、ハードウェアによって較正係数で更新されます。

ソフトウェアは、これらのビットに新しい較正係数を書き込むことができます。新しい較正係数がアナログ ADC に格納されている現在のものと異なる場合は、新しい差動較正の起動時に適用されます。

**注:** ソフトウェアは、ADEN=1、ADSTART=0、および JADSTART=0 のときのみ (ADC が有効であり、較正中でなく、変換中でない)、これらのビットを書き込むことができます。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **CALFACT\_S[6:0]**: シングルエンドモードの較正係数

これらのビットは、ハードウェアまたはソフトウェアによって書き込まれます。

シングルエンド入力較正が完了すると、ハードウェアによって較正係数で更新されます。

ソフトウェアは、これらのビットに新しい較正係数を書き込むことができます。新しい較正係数がアナログ ADC に格納されている現在のものと異なる場合は、新しいシングルエンド較正の起動時に適用されます。

**注:** ソフトウェアは、ADEN=1、ADSTART=0、および JADSTART=0 のときのみ (ADC が有効であり、較正中でなく、変換中でない)、これらのビットを書き込むことができます。



## 15.6 ADC 共通レジスタ

これらのレジスタは、マスタおよびスレーブ ADC に共通の制御およびステータスレジスタを定義します。

- ADC1 (マスタ) および ADC2 (スレーブ) に関連する1つのレジスタセット
- STM32F303xB/C および STM32F358xC デバイスで使用可能な ADC3 (マスタ) および ADC4 (スレーブ) に関連する1つのレジスタセット

### 15.6.1 ADC 共通ステータスレジスタ (ADCx\_CSR、x=12 または 34)

アドレスオフセット: 0x00 (このオフセットアドレスはマスタ ADC ベースアドレス + 0x300 に対する相対値です。)

リセット値: 0x0000 0000

このレジスタにより、別の ADC のステータスビットの状態を確認することができます。ただし、これは読み出し専用で、別の ADC のステータスビットをクリアすることはできません。その代わりに、対応する ADCx\_SR レジスタに 0 を書き込むことによって各ステータスビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	JQOVF_SLV	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV
スレーブ ADC															
					r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
マスタ ADC															
					r	r	r	r	r	r	r	r	r	r	r

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **JQOVF\_SLV**: スレーブ ADC のインジェクトコンテキストキューのオーバーフローフラグ

このビットは対応する ADCx\_ISR レジスタの JQOVF ビットのコピーです。

ビット 25 **AWD3\_SLV**: スレーブ ADC のアナログウォッチドッグ 3 フラグ

このビットは対応する ADCx\_ISR レジスタの AWD3 ビットのコピーです。

ビット 24 **AWD2\_SLV**: スレーブ ADC のアナログウォッチドッグ 2 フラグ

このビットは対応する ADCx\_ISR レジスタの AWD2 ビットのコピーです。

ビット 23 **AWD1\_SLV**: スレーブ ADC のアナログウォッチドッグ 1 フラグ

このビットは対応する ADCx\_ISR レジスタの AWD1 ビットのコピーです。

ビット 22 **JEOS\_SLV**: スレーブ ADC のインジェクトシーケンスの終了フラグ

このビットは対応する ADCx\_ISR レジスタの JEOS ビットのコピーです。

ビット 21 **JEOC\_SLV**: スレーブ ADC のインジェクト変換の終了フラグ

このビットは対応する ADCx\_ISR レジスタの JEOC ビットのコピーです。

ビット 20 **OVR\_SLV**: スレーブ ADC のオーバーランフラグ

このビットは対応する ADCx\_ISR レジスタの OVR ビットのコピーです。

ビット 19 **EOS\_SLV**: スレーブ ADC のレギュラシーケンスの終了フラグ

このビットは対応する ADCx\_ISR レジスタの EOS ビットのコピーです。

- ビット 18 **EOC\_SLV** : スレーブ ADC のレギュラ変換の終了フラグ  
このビットは対応する ADCx\_ISR レジスタの EOC ビットのコピーです。
- ビット 17 **EOSMP\_SLV** : スレーブ ADC のサンプリングフェーズの終了フラグ  
このビットは対応する ADCx\_ISR レジスタの EOSMP ビットのコピーです。
- ビット 16 **ADRDY\_SLV** : スレーブ ADC レディ  
このビットは対応する ADCx\_ISR レジスタの ADRDY ビットのコピーです。
- ビット 15:11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **JQOVF\_MST** : マスタ ADC のインジェクトコンテキストキューのオーバーフローフラグ  
このビットは対応する ADCx\_ISR レジスタの JQOVF ビットのコピーです。
- ビット 9 **AWD3\_MST** : マスタ ADC のアナログウォッチドッグ 3 フラグ  
このビットは対応する ADCx\_ISR レジスタの AWD3 ビットのコピーです。
- ビット 8 **AWD2\_MST** : マスタ ADC のアナログウォッチドッグ 2 フラグ  
このビットは対応する ADCx\_ISR レジスタの AWD2 ビットのコピーです。
- ビット 7 **AWD1\_MST** : マスタ ADC のアナログウォッチドッグ 1 フラグ  
このビットは対応する ADCx\_ISR レジスタの AWD1 ビットのコピーです。
- ビット 6 **JEOS\_MST** : マスタ ADC のインジェクトシーケンスの終了フラグ  
このビットは対応する ADCx\_ISR レジスタの JEOS ビットのコピーです。
- ビット 5 **JEOC\_MST** : マスタ ADC のインジェクト変換の終了フラグ  
このビットは対応する ADCx\_ISR レジスタの JEOC ビットのコピーです。
- ビット 4 **OVR\_MST** : マスタ ADC のオーバーランフラグ  
このビットは対応する ADCx\_ISR レジスタの OVR ビットのコピーです。
- ビット 3 **EOS\_MST** : マスタ ADC のレギュラシーケンスの終了フラグ  
このビットは対応する ADCx\_ISR レジスタの EOS ビットのコピーです。
- ビット 2 **EOC\_MST** : マスタ ADC のレギュラ変換の終了フラグ  
このビットは対応する ADCx\_ISR レジスタの EOC ビットのコピーです。
- ビット 1 **EOSMP\_MST** : マスタ ADC のサンプリングフェーズの終了フラグ  
このビットは対応する ADCx\_ISR レジスタの EOSMP ビットのコピーです。
- ビット 0 **ADRDY\_MST** : マスタ ADC レディ  
このビットは対応する ADCx\_ISR レジスタの ADRDY ビットのコピーです。

## 15.6.2 ADC 共通制御レジスタ (ADCx\_CCR, x=12 または 34)

アドレスオフセット : 0x08 (このオフセットアドレスはマスタ ADC のベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBAT EN	TS EN	VREF EN	Res.	Res.	Res.	Res.	CKMODE[1:0]	
							r/w	r/w	r/w					r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDMA[1:0]		DMA CFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]				
r/w	r/w	r/w		r/w	r/w	r/w	r/w				r/w	r/w	r/w	r/w	r/w

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **VBATEN** : V<sub>BAT</sub> イネーブル

このビットは、V<sub>BAT</sub> チャンネルを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : V<sub>BAT</sub> チャンネルは無効です。

1 : V<sub>BAT</sub> チャンネルは有効です。

**注 :** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、およびADEN=0)、このビットを書き込むことができます。

ビット 23 **TSEN** : 温度センサ有効

このビットは、温度センサチャンネルを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : 温度センサチャンネルは無効です。

1 : 温度センサチャンネルは有効です。

**注 :** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、およびADEN=0)、このビットを書き込むことができます。

ビット 22 **VREFEN** : V<sub>REFINT</sub> 有効

このビットは、V<sub>REFINT</sub> チャンネルを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : V<sub>REFINT</sub> チャンネルは無効です。

1 : V<sub>REFINT</sub> チャンネルは有効です。

**注 :** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、およびADEN=0)、このビットを書き込むことができます。

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

### ビット 17:16 **CKMODE[1:0]** : ADC クロックモード

これらのビットは、ADC クロックスキーム（マスタとスレーブの両方の ADC に共通）を定義するために、ソフトウェアによってセット/クリアされます。

00 : CK\_ADCx (x=123) (非同期クロックモード)。製品レベルで生成されます ([Section 6: Reset and clock control \(RCC\)](#) を参照)。

01 : HCLK/1 (同期クロックモード)。この設定は、AHB クロックプリスケアラが 1 にセットされた場合で (RCC\_CFGR レジスタの HPRE[3:0] = 0xxx)、システムクロックが 50% のデューティサイクルを持つ場合にのみ有効にする必要があります。

10 : HCLK/2 (同期クロックモード)

11 : HCLK/4 (同期クロックモード)

すべての同期クロックモードにおいて、タイマトリガから変換開始までの遅延にジッタはありません。

**注：** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0)、これらのビットを書き込むことができます。

### ビット 15:14 **MDMA[1:0]** : デュアル ADC モードのためのダイレクトメモリアクセスモード

このビットフィールドは、ソフトウェアによってセット/クリアされます。詳細については、DMA コントローラのセクションを参照してください。

00 : MDMA モードは無効です。

01 : 予約済み

10 : MDMA モードは、12 ビットおよび 10 ビットの分解能で有効です。

11 : MDMA モードは、8 ビットおよび 6 ビットの分解能で有効です。

**注：** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

### ビット 13 **DMACFG** : DMA 設定 (デュアル ADC モード)

このビットは、2 つの DMA 動作モードを選択するためにソフトウェアによってセット/クリアされ、DMAEN=1 のときのみ有効です。

0 : DMA ワンショットモードが選択されています。

1 : DMA サーキュラモードが選択されています。

詳細については、[セクション : DMA を使用した変換の管理](#) を参照してください。

**注：** ソフトウェアは、ADSTART=0 である (レギュラ変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

ビット 12 予約済みであり、リセット値に保持する必要があります。

ビット 11:8 **DELAY** : 2つのサンプリングフェーズ間の遅延

ソフトウェアでセット/クリアされます。これらのビットはデュアルインターリーブモードで使用します。ADC 分解能の値対 DELAY ビット値については、表 99 を参照してください。

**注:** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、およびADEN=0)、これらのビットを書き込むことができます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DUAL[4:0]** : デュアル ADC モード選択

これらのビットは、動作モードを選択するために、ソフトウェアによって書き込まれます。

すべての ADC が単独で動作します。

00000 : 独立モード

00001 から 01001 : デュアルモード、マスタおよびスレーブ ADCは協調して動作

00001 : レギュラ同時 + インジェクト同時モードの組み合わせ

00010 : レギュラ同時 + オルタネートトリガモードの組み合わせ

00011 : インターリーブモード + インジェクト同時モードの組み合わせ

00100 : 予約済み

00101 : インジェクト同時モードのみ

00110 : レギュラ同時モードのみ

00111 : インターリーブモードのみ

01001 : オルタネートトリガモードのみ

他のすべての組み合わせは予約済みであり、プログラムすることはできません。

**注:** ソフトウェアは、ADC が無効なときだけ (ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0、およびADEN=0)、これらのビットを書き込むことができます。

表 99. DELAY ビット対 ADC 分解能

DELAY ビット	12 ビット分解能	10 ビット分解能	8 ビット分解能	6 ビット分解能
0000	1 * T <sub>ADC_CLK</sub>	1 * T <sub>ADC_CLK</sub>	1 * T <sub>ADC_CLK</sub>	1 * T <sub>ADC_CLK</sub>
0001	2 * T <sub>ADC_CLK</sub>	2 * T <sub>ADC_CLK</sub>	2 * T <sub>ADC_CLK</sub>	2 * T <sub>ADC_CLK</sub>
0010	3 * T <sub>ADC_CLK</sub>	3 * T <sub>ADC_CLK</sub>	3 * T <sub>ADC_CLK</sub>	3 * T <sub>ADC_CLK</sub>
0011	4 * T <sub>ADC_CLK</sub>	4 * T <sub>ADC_CLK</sub>	4 * T <sub>ADC_CLK</sub>	4 * T <sub>ADC_CLK</sub>
0100	5 * T <sub>ADC_CLK</sub>	5 * T <sub>ADC_CLK</sub>	5 * T <sub>ADC_CLK</sub>	5 * T <sub>ADC_CLK</sub>
0101	6 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
0110	7 * T <sub>ADC_CLK</sub>	7 * T <sub>ADC_CLK</sub>	7 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
0111	8 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
1000	9 * T <sub>ADC_CLK</sub>	9 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
1001	10 * T <sub>ADC_CLK</sub>	10 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
1010	11 * T <sub>ADC_CLK</sub>	10 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
1011	12 * T <sub>ADC_CLK</sub>	10 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>
その他	12 * T <sub>ADC_CLK</sub>	10 * T <sub>ADC_CLK</sub>	8 * T <sub>ADC_CLK</sub>	6 * T <sub>ADC_CLK</sub>

## 15.6.3 デュアルモード用 ADC 共通レギュラデータレジスタ (ADCx\_CDR、x=12 または 34)

アドレスオフセット : 0x0C (このオフセットアドレスはマスタ ADC のベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

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

ビット 31:16 **RDATA\_SLV[15:0]** : スレーブ ADC のレギュラデータ

デュアルモードでは、これらのビットにはスレーブ ADC のレギュラデータが格納されます。 [セクション 15.3.29 : デュアル ADC モード](#) を参照してください。

データの配置は、 [セクション : データレジスタ、データの配置およびデータオフセット \(ADCx\\_DR、OFFSETy、OFFSETy\\_CH、ALIGN\)](#) に示すように適用されています。

ビット 15:0 **RDATA\_MST[15:0]** : マスタ ADC のレギュラデータ

デュアルモードでは、これらのビットにはマスタ ADC のレギュラデータが格納されます。 [セクション 15.3.29 : デュアル ADC モード](#) を参照してください。

データの配置は、 [セクション : データレジスタ、データの配置およびデータオフセット \(ADCx\\_DR、OFFSETy、OFFSETy\\_CH、ALIGN\)](#) に示すように適用されています。

MDMA=0b11 モードでは、ビット 15:8 には SLV\_ADC\_DR[7:0] が、ビット 7:0 には MST\_ADC\_DR[7:0] が含まれています。

## 15.6.4 ADC レジスタマップ

次の表は ADC レジスタの一覧です。

**注 :** *STM32F303x6/8 および STM32F328x8 デバイスには、ADC1 と ADC2 のみがあります。*

**表 100. ADC グローバルレジスタマップ**

オフセット	レジスタ
0x000 - 0x04C	マスタ ADCx (ADC1 または ADC3)
0x050 - 0x0FC	予約済み
0x100 - 0x14C	スレーブ ADCx (ADC2 または ADC4)
0x118 - 0x1FC	予約済み
0x200 - 0x24C	予約済み
0x250 - 0x2FC	予約済み
0x300 - 0x308	マスタおよびスレーブ ADC に共通のレジスタ (ADC12 または ADC34)

表 101. ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100、x=1..4)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	ADCx_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY				
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0			
0x04	ADCx_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE				
	リセット値																						0	0	0	0	0	0	0	0	0	0	0	0			
0x08	ADCx_CR	ADCAL	ADCALDIF	ADVREGEN[1:0]																								JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN				
	リセット値	0	0	1	0																								0	0	0	0	0	0			
0x0C	ADCx_CFGR	Res.	AWD1CH[4:0]				JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM [2:0]		DISCEN	Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]			EXTSEL [3:0]			ALIGN	RES [1:0]		Res.	DMACFG	DMAEN						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	予約済み	Res.																																			
0x14	ADCx_SMPR1	Res.	Res.	SMP9 [2:0]		SMP8 [2:0]		SMP7 [2:0]		SMP6 [2:0]		SMP5 [2:0]		SMP4 [2:0]		SMP3 [2:0]		SMP2 [2:0]		SMP1 [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x18	ADCx_SMPR2	Res.	Res.	Res.	Res.	Res.	SMP18 [2:0]		SMP17 [2:0]		SMP16 [2:0]		SMP15 [2:0]		SMP14 [2:0]		SMP13 [2:0]		SMP12 [2:0]		SMP11 [2:0]		SMP10 [2:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	予約済み	Res.																																			
0x20	ADCx_TR1	Res.	Res.	Res.	Res.	HT1[11:0]											Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT1[11:0]										
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1	1																			
0x24	ADCx_TR2	Res.	Res.	Res.	Res.	HT2[7:0]											Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]										
	リセット値					1	1	1	1	1	1	1	1	1	1																						
0x28	ADCx_TR3	Res.	Res.	Res.	Res.	HT3[7:0]											Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]										
	リセット値					1	1	1	1	1	1	1	1	1	1																						
0x2C	予約済み	Res.																																			
0x30	ADCx_SQR1	Res.	Res.	SQ4[4:0]				Res.	SQ3[4:0]				Res.	SQ2[4:0]				Res.	SQ1[4:0]				Res.	L[3:0]													
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x34	ADCx_SQR2	Res.	Res.	SQ9[4:0]				Res.	SQ8[4:0]				Res.	SQ7[4:0]				Res.	SQ6[4:0]				Res.	SQ5[4:0]													
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x38	ADCx_SQR3	Res.	Res.	SQ14[4:0]				Res.	SQ13[4:0]				Res.	SQ12[4:0]				Res.	SQ11[4:0]				Res.	SQ10[4:0]													
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x3C	ADCx_SQR4	Res.	Res.	SQ16[4:0]				Res.	SQ15[4:0]				Res.	SQ14[4:0]				Res.	SQ13[4:0]				Res.	SQ12[4:0]													
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x40	ADCx_DR	Res.	Res.	レギュラ RDATA[15:0]																																	
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x44-0x48	予約済み	Res.																																			
0x4C	ADCx_JSQR	Res.	JSQ4[4:0]				Res.	JSQ3[4:0]				Res.	JSQ2[4:0]				Res.	JSQ1[4:0]				Res.	JEXTEN[1:0]				JEXTSEL [3:0]				JL[1:0]						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				



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

表 101. ADC レジスタマップと各 ADC のリセット値 (マスタ ADC のオフセットは 0x000、スレーブ ADC のオフセットは 0x100、x=1..4) (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x50-0x5C	予約済み	Res.																																			
0x60	ADCx_OFR1	OFFSET1_EN	OFFSET1_CH[4:0]				Res.														OFFSET1[11:0]																
	リセット値	0	0	0	0	0	0																														
0x64	ADCx_OFR2	OFFSET2_EN	OFFSET2_CH[4:0]				Res.														OFFSET2[11:0]																
	リセット値	0	0	0	0	0	0																														
0x68	ADCx_OFR3	OFFSET3_EN	OFFSET3_CH[4:0]				Res.														OFFSET3[11:0]																
	リセット値	0	0	0	0	0	0																														
0x6C	ADCx_OFR4	OFFSET4_EN	OFFSET4_CH[4:0]				Res.														OFFSET4[11:0]																
	リセット値	0	0	0	0	0	0																														
0x70-0x7C	予約済み	Res.																																			
0x80	ADCx_JDR1	Res.														JDATA1[15:0]																					
	リセット値	0																																			
0x84	ADCx_JDR2	Res.														JDATA2[15:0]																					
	リセット値	0																																			
0x88	ADCx_JDR3	Res.														JDATA3[15:0]																					
	リセット値	0																																			
0x8C	ADCx_JDR4	Res.														JDATA4[15:0]																					
	リセット値	0																																			
0x8C-0x9C	予約済み	Res.																																			
0xA0	ADCx_AWD2CR	Res.														AWD2CH[18:1]														Res.							
	リセット値	0																																			
0xA4	ADCx_AWD3CR	Res.														AWD3CH[18:1]														Res.							
	リセット値	0																																			
0xA8-0xAC	予約済み	Res.																																			
0xB0	ADCx_DIFSEL	Res.														DIFSEL[18:1]														Res.							
	リセット値	0																																			
0xB4	ADCx_CALFACT	Res.											CALFACT_D[6:0]						Res.											CALFACT_S[6:0]							
	リセット値	0																																			



表 102. ADC レジスタマップとリセット値 (マスタおよびスレーブ ADC 共通レジスタ)  
オフセット=0x300、x=1 または 34)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADCx_CSR	Res.	Res.	Res.	Res.	Res.	JQOVF_SLV	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV	Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
	リセット値							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x04	予約済み	Res.																															
0x08	ADCx_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREFEN	Res.	Res.	Res.	Res.	CKMODE[1:0]	MDMA[1:0]	DMACFG	Res.	DELAY[3:0]			Res.	Res.	Res.	DUAL[4:0]							
	リセット値								0	0	0					0	0	0	0		0						0	0	0	0	0		
0x0C	ADCx_CDR	RDATA_SLV[15:0]															RDATA_MST[15:0]																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 16 D/A コンバータ (DAC1 および DAC2)

### 16.1 概要

DAC モジュールは、12 ビットの電圧出力デジタルアナログコンバータです。DAC は、8 または 12 ビットモードで設定でき、DMA コントローラと組み合わせて使用することもできます。12 ビットモードでは、データを左詰め右詰めどちらにも配置できます。入力基準電圧、 $V_{REF+}$  (ADC と共用) が使用できます。より高い電流駆動力を得るために、任意で出力バッファを有効にすることができます。

### 16.2 DAC1/2 の主な機能

STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE では、出力バッファを持つ 12 ビット DAC チャンネルを 2 個実装しています。

STM32F303x6/8 および STM32F328x8 では、12 ビット DAC チャンネルを 3 個実装しています (出力バッファを持つ DAC チャンネル 1 個と出力バッファを持たない DAC チャンネル 2 個)。

- DAC1 には DAC チャンネルが 2 個実装されています。
  - DAC1 チャンネル 1 (出力は DAC1\_OUT1)
  - DAC1 チャンネル 2 (出力は DAC1\_OUT2)

2 つのチャンネルは、両方のチャンネルが同期更新操作のためにグループ化されているときには、独立または同時に使用することができます (デュアルモード)。

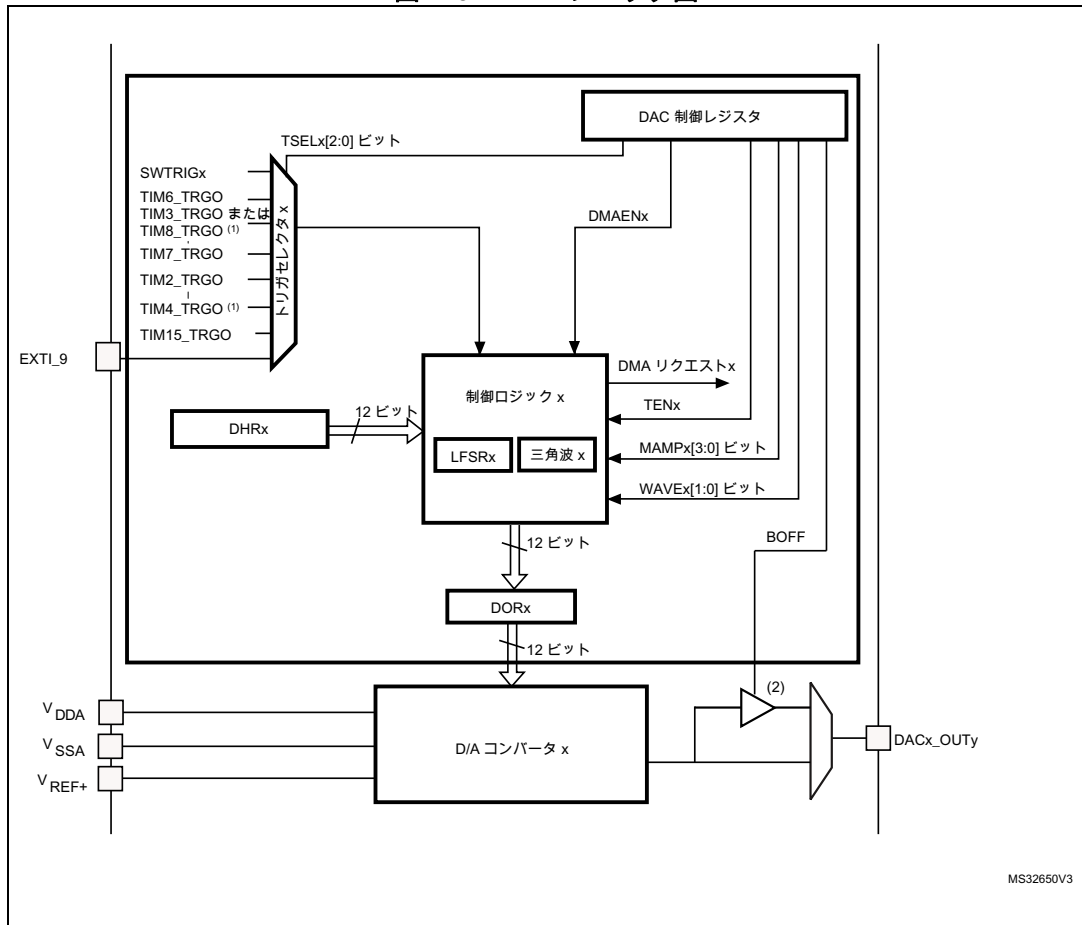
- DAC2 では、出力が DAC2\_OUT1 である 1 つのチャンネル (DAC2 チャンネル 1) のみが実装されています (STM32F303x6/8 および STM32F328x8 デバイスのみ)。

DAC の主な機能は以下のとおりです。

- 12 ビットモードでのデータの左詰めまたは右詰め
- 同期更新機能
- ノイズ波生成 (DAC1 のみ)
- 三角波生成 (DAC1 のみ)
- 独立または同時変換 (デュアルモードのみ)
- 各チャンネルでの DMA 利用
- DMA アンダーランエラー検出
- 変換外部トリガ
- プログラム可能な内部バッファ
- 入力基準電圧、 $V_{DDA}$

図 113 および 図 114 は DAC1 および DAC2 チャンネルのブロック図を、表 103 はピンの概要を示します。

図 113. DAC1 ブロック図



1. TIM8\_TRGO および TIM4\_TRGO は、STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用可能です。
2. STM32F303x6/8 および STM32F328 では、DAC1 チャンネル 2 に出力バッファはありません。代わりに、DAC1\_OUT2 を対応する I/O (PA5) に接続することができるスイッチがあります (DAC2 ブロック図を参照)。

図 114. DAC2 ブロック図 (STM32F303x6/8 および STM32F328 のみ)

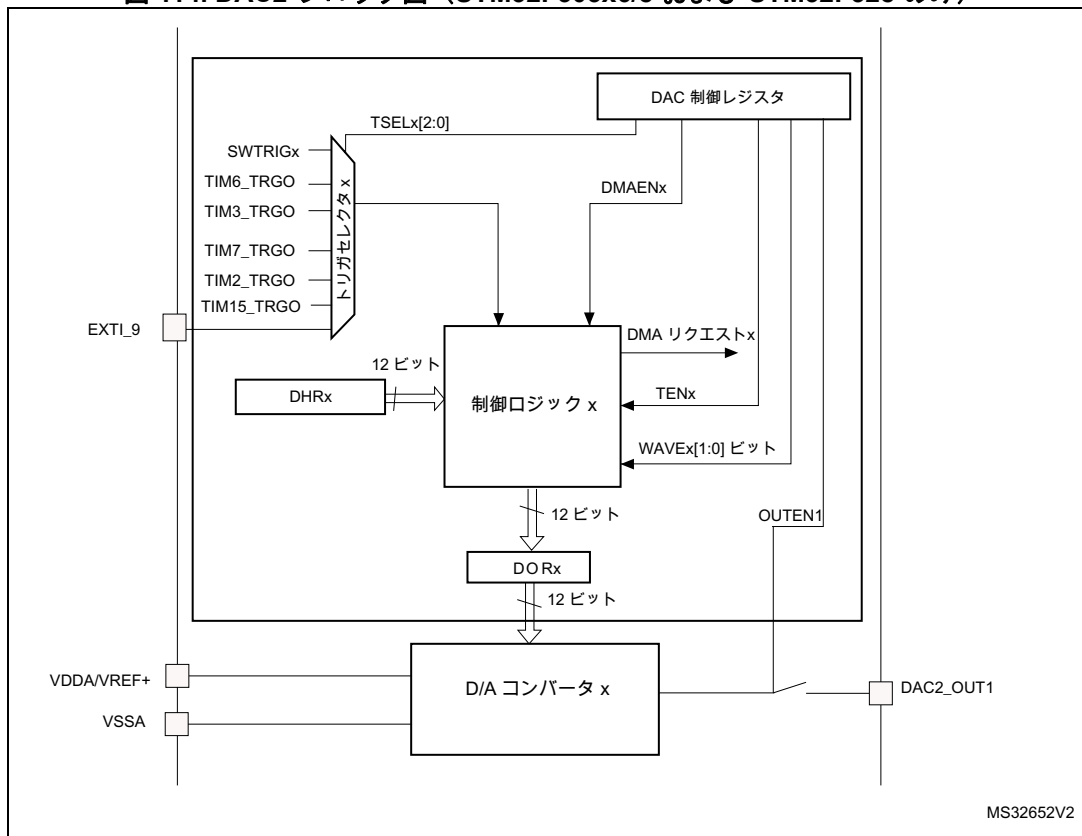


表 103. DAC1 ピン

名前	信号タイプ	説明
V <sub>REF+</sub> <sup>(1)</sup>	入力、アナログ基準電圧正	DAC のハイレベル/正基準電圧
V <sub>DDA</sub>	入力、アナログ電源供給	アナログ電源供給
V <sub>SSA</sub>	入力、アナログ供給グラウンド	アナログ電源供給のグラウンド
DAC1_OUT1/2 DAC2_OUT1	アナログ出力信号	DACx チャネル y アナログ出力

1. STM32F303x6/8 および STM32F328 では、VDDA および VREF+ は内部的に接続されています。

**注:** DACx チャネル y が有効になると、対応する GPIO ピン (PA4、PA5 または PA6) が自動的にアナログコンバータ出力 (DACx\_OUTy) に接続されます。寄生消費を防ぐために、PA4、PA5、または PA6 ピンはまずアナログ (AIN) として設定してください。

## 16.3 DAC 出力バッファイネーブル/DAC 出力スイッチ

出力バッファを持つ DAC1 チャネル 1 および DAC1 チャネル 2 は、DAC1\_OUT1/2 出力における出力インピーダンスを減らし、外部オペアンプがなくても外部負荷を直接駆動するために使用できます。この機能は、STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE で使用できます。

STM32F303x6/8 および STM32F328 には、出力バッファを持つ DAC1 チャネル 1 が内蔵されています。DAC1 チャネル 2 には出力バッファはありません。代わりに、DAC1\_OUT2 を対応する I/O (PA5)

に接続することができるスイッチがあります。スイッチは、DAC\_CR レジスタの OUTEN2 ビットを通じて有効/無効にできます。DAC2 チャンネル 1 には出力バッファはありません。代わりに、DAC2\_OUT1 を対応する I/O (PA6) に接続することができるスイッチがあります。スイッチは、DAC\_CR レジスタの OUTEN1 ビットを通じて有効/無効にできます。

DAC1 チャンネル出力バッファは、DAC\_CR レジスタの BOFF1 ビットを通じて有効/無効にできます。

## 16.4 DAC チャンネルイネーブル

各 DAC チャンネルは、DAC\_CR レジスタの対応する ENx ビットをセットすることによって起動できます。各 DAC チャンネルは、スタートアップ時間  $t_{WAKEUP}$  後に有効になります。

**注:** ENx ビットは、アナログ DAC チャンネル x のマクロセルのみを有効にします。DAC チャンネル x デジタルインタフェースは、ENx ビットがリセットされた場合でも有効になります。

## 16.5 シングルモードの機能説明

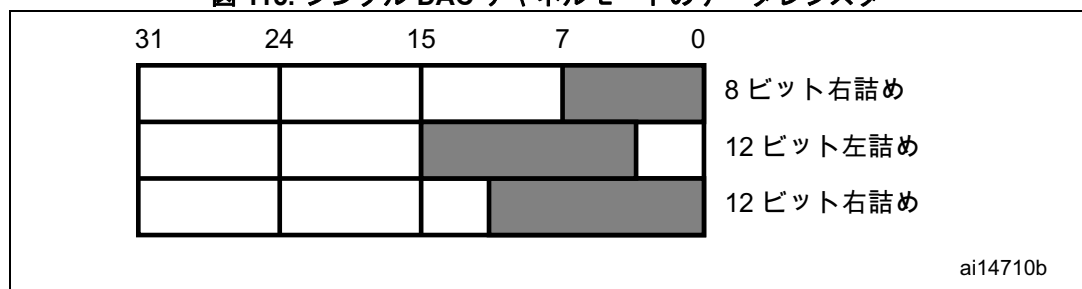
### 16.5.1 DAC データフォーマット

この場合、次の 3 つの設定が可能です。

- 8 ビット右詰め: データは、DAC\_DHR8Rx [7:0] ビット (DHRx [11:4] ビットに格納) にソフトウェアによってロードされること。
- 12 ビット左詰め: データは、DAC\_DHR12Lx [15:4] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。
- 12 ビット右詰め: データは、DAC\_DHR12Rx [11:0] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。

ユーザによって書き込まれたデータは、ロードされた DAC\_DHRyyyx レジスタに応じて、シフトされてから、DHRx (メモリマップされない内部レジスタであるデータ保持レジスタ x) に格納されます。その後、DHRx レジスタは自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、DORx レジスタにロードされます。

図 115. シングル DAC チャンネルモードのデータレジスタ



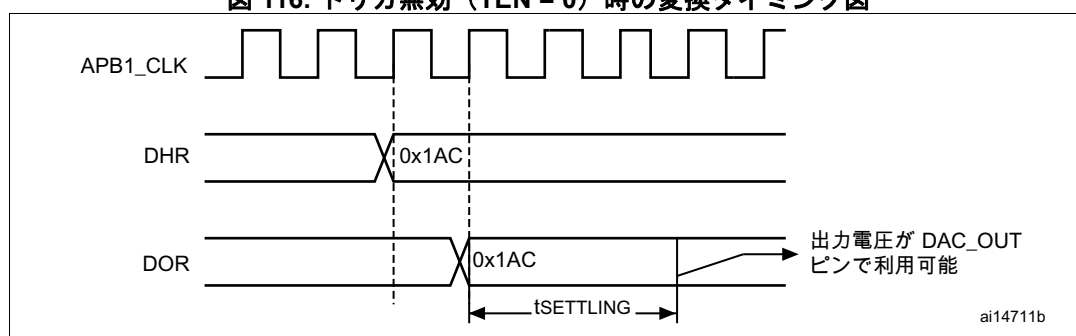
## 16.5.2 DAC チャンネル変換

DAC\_DORx に直接書き込むことはできませんので、DAC\_DHRx レジスタをロードする (DAC\_DHR8Rx、DAC\_DHR12Lx、DAC\_DHR12Rx への書き込み) ことによって、DAC チャンネル x へのデータ転送を行う必要があります。

DAC\_DHRx レジスタに格納されたデータは、ハードウェアトリガが選択されていない (DAC\_CR レジスタの TENx ビットがリセットされている) 場合に、1 APB1 クロックサイクル後に DAC\_DORx レジスタに自動的に転送されます。ただし、ハードウェアトリガが選択されている (DAC\_CR レジスタの TENx ビットがセットされている) ときには、トリガが発生すると、転送は 3 PCLK1 クロックサイクル後に行われます。

DAC\_DORx に DAC\_DHRx の内容がロードされると、電源電圧とアナログ出力負荷に応じて決定される t<sub>SETTLING</sub> 時間後にアナログ出力電圧が使用可能になります。

図 116. トリガ無効 (TEN = 0) 時の変換タイミング図



### 1 つの LFSR 生成による独立トリガ

DAC をこの変換モード ([セクション 16.7: ノイズ生成](#) を参照) に設定するには、次の手順が必要です。

1. DAC チャンネルトリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットをセットすることによって、トリガソースを設定します。
3. DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DAC\_DHRx レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、DAC チャンネルデータをロードします。

DAC チャンネル x トリガが発生すると、同じマスクを持つ LFSRx カウンタが DHRx レジスタに加算され、合計が DAC\_DORx に転送されます (3 APB クロックサイクル後)。その後、LFSRx カウンタが更新されます。

### 1 つの三角波生成による独立トリガ

DAC をこの変換モード ([セクション 16.8: 三角波生成](#) を参照) に設定するには、次の手順が必要です。

1. DAC チャンネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットをセットすることによって、トリガソースを設定します。
3. DAC チャンネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. DAC チャンネル x データを、目的の DAC\_DHRx レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、ロードします。

DAC チャネル x トリガが発生すると、同じ三角波振幅を持つ DAC チャネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC\_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャネル x の三角波カウンタが更新されます。

## 16.5.3 DAC 出力電圧

デジタル入力は、0 から  $V_{REF+}$  までのリニア変換で出力電圧に変換されます。

各 DAC チャネルピンのアナログ出力電圧は、次の式によって求められます。

$$DAC_{output} = V_{REF+} \times \frac{DOR}{4096}$$

## 16.5.4 DAC トリガ選択

TENx 制御ビットがセットされている場合、外部イベント (タイマカウンタ、外部割り込みラインなど) によって変換をトリガできます。表 104 に示すように、考えられるイベントのうち、どのイベントが変換をトリガするかは TSELx[2:0] 制御ビットによって決まります。

表 104. 外部トリガ (DAC1)

転送元	タイプ	TSEL[2:0]
TIM6_TRGO イベント	オンチップタイマからの内部信号	000
TIM3_TRGO イベント <sup>(1)</sup> または タイマ 8 TRGO イベント <sup>(2)</sup>		001 <sup>(2)</sup>
TIM7_TRGO イベント		010
TIM15_TRGO イベント		011
TIM2_TRGO イベント		100
TIM4_TRGO イベント		101
EXTI ライン 9	外部ピン	110
SWTRIG	ソフトウェア制御ビット	111

- DAC1 トリガソースとして TIM3\_TRGO イベントを選択するには、SYSCFG\_CFGR1 レジスタに DAC\_TRIG\_RMP ビットをセットする必要があります。
- TSEL = 001 の場合、DAC トリガは SYSCFG\_CFGR1 レジスタの DAC\_TRIG\_RMP ビットを使用して選択されます。このビットがクリアされると、DAC トリガはタイマ 8 TRGO イベントになります。このビットがセットされると、DAC トリガはタイマ 3 TRGO イベントになります。

表 105. 外部トリガ (DAC2)

転送元	タイプ	TSEL[2:0]
TIM6_TRGO イベント	オンチップタイマからの内部信号	000
TIM3_TRGO イベント		001
TIM7_TRGO イベント		010
TIM15_TRGO イベント		011
TIM2_TRGO イベント		100
EXTI ライン 9	外部ピン	110
SWTRIG	ソフトウェア制御ビット	111

DAC インタフェースが選択されたタイマ TRGO 出力または選択された外部割り込みライン 9 の立ち上がりエッジを検出するたびに、DAC\_DHRx レジスタに最後に格納されたデータが DAC\_DORx レジスタに転送されます。DAC\_DORx レジスタは、トリガが発生してから 3 APB1 サイクル後に更新されます。

ソフトウェアトリガが選択されている場合、変換は、SWTRIG ビットがセットされると開始されます。SWTRIG ビットは、DAC\_DHRx レジスタの内容が DAC\_DORx レジスタにロードされると、ハードウェアによってリセットされます。

**注：** ENx ビットがセットされているときには、TSELx[2:0] ビットを変更することはできません。ソフトウェアトリガが選択されているときには、DAC\_DHRx レジスタから DAC\_DORx レジスタへの転送は、わずか 1 APB1 クロックサイクルで行われます。

## 16.6 デュアルモードの機能説明

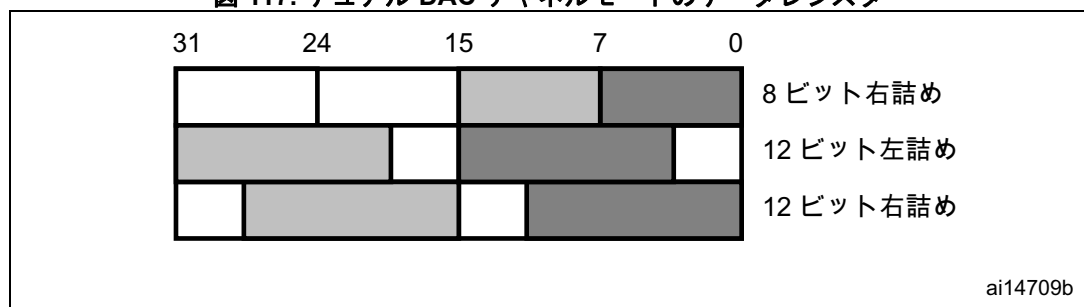
### 16.6.1 DAC データフォーマット

デュアル DAC チャンネルモードでは、3 つの設定が可能です。

- 8 ビット右詰め：DAC チャンネル 1 のデータは、DAC\_DHR8RD [7:0] ビット (DHR1[11:4] ビットに格納) に、DAC チャンネル 2 のデータは、DAC\_DHR8RD [15:8] ビット (DHR2[11:4] ビットに格納) にロードされます。
- 12 ビット左詰め：DAC チャンネル 1 のデータは、DAC\_DHR12LD [15:4] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC\_DHR12LD [31:20] ビット (DHR2[11:0] ビットに格納) にロードされます。
- 12 ビット右詰め：DAC チャンネル 1 のデータは、DAC\_DHR12RD [11:0] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC\_DHR12LD [27:16] ビット (DHR2[11:0] ビットに格納) にロードされます。

ロードされた DAC\_DHRyyyD レジスタに応じて、ユーザによって書き込まれたデータは、シフトされてから、DHR1 および DHR2 (データ保持レジスタ、メモリマップされない内部レジスタ) に格納されます。その後、DHR1 および DHR2 レジスタは、自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、それぞれ DOR1 および DOR2 レジスタにロードされます。

図 117. デュアル DAC チャンネルモードのデータレジスタ



### 16.6.2 デュアルモードでの DAC チャンネル変換

デュアルモードでの DAC チャンネル変換は、シングルモードの場合と同様に行われます (セクション 16.5.2 を参照)。ただし、データは DAC\_DHR8Rx、DAC\_DHR12Lx、DAC\_DHR12Rx、DAC\_DHR8RD、DAC\_DHR12LD、または DAC\_DHR12RD に書き込んでロードする必要があります。



## 16.6.3 デュアル変換モードの説明

同時に 2 つの DAC チャンネルを必要とするアプリケーションで、バスのバンド幅を有効に使用するために、DHR8RD、DHR12RD、および DHR12LD の 3 つのデュアルレジスタが搭載されています。両方の DAC チャンネルを同時に駆動するには、一意なレジスタアクセスが必要です。

2 つの DAC チャンネルとこれらのデュアルレジスタを使用することで、11 の変換モードが使用可能です。すべての変換モードは、必要な場合には、別の DHRx レジスタを使っても利用できます。

すべてのモードについて、以下の節で説明します。

DAC 変換を駆動する APB バス (APB または APB1) の詳細については、[セクション 16.5.2 : DAC チャンネル変換](#)を参照してください。

### 波形生成なしの独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、DHR1 レジスタが DAC\_DOR1 に転送されます (3 APB クロックサイクル後)。

DAC チャンネル 2 トリガが発生すると、DHR2 レジスタが DAC\_DOR2 に転送されます (3 APB クロックサイクル後)。

### 1 つの LFSR 生成による独立トリガ

DAC をこの変換モードに設定するには ([セクション 16.7 : ノイズ生成](#)を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、デュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャンネル 2 トリガが発生すると、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 異なる LFSR 生成による独立トリガ

DAC をこの変換モードに設定するには( [セクション 16.7: ノイズ生成](#)を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. 2 つの DAC チャネル WAVEx[1:0] ビットを“01”に設定し、MAMP1[3:0] ビットと MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャネルデータをロードします。

DAC チャネル 1 トリガが発生すると、MAMP1[3:0] によって設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャネル 2 トリガが発生すると、MAMP2[3:0] によって設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 1 つの三角波生成による独立トリガ

DAC をこの変換モードに設定するには( [セクション 16.8: 三角波生成](#)を参照)、次の手順が必要です。

1. DAC チャネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. DAC チャネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. DAC チャネル x データを、目的の DAC\_DHRx レジスタ

DAC 変換を駆動する APB バス (APB または APB1) の詳細については、 [セクション 16.5.2: DAC チャネル変換](#) を参照してください。

DAC チャネル x トリガが発生すると、同じ三角波振幅を持つ DAC チャネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC\_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャネル x の三角波カウンタが更新されます。

## 異なる三角波生成による独立トリガ

DAC をこの変換モードに設定するには( [セクション 16.8: 三角波生成](#)を参照)、次の手順が必要です。

1. DAC チャネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. DAC チャネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで異なる最大振幅値を設定します。
4. DAC チャネル x データを、目的の DAC\_DHRx レジスタ

DAC チャネル x トリガが発生すると、MAMPx[3:0] によって設定された三角波振幅を持つ DAC チャネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC\_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャネル x の三角波カウンタが更新されます。

## 同時ソフトウェア開始

DAC をこの変換モードに設定するには、次の手順が必要です。

1. 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) に、デュアル DAC チャンネルデータをロードします。

この設定では、1 APB クロックサイクルです。

## 波形生成なしの同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) に、デュアル DAC チャンネルデータをロードします。

トリガが発生すると、DHR1 および DHR2 レジスタが DAC\_DOR1 と DAC\_DOR2 にそれぞれ転送されます (3 APB クロックサイクル後)。

## 1 つの LFSR 生成による同時トリガ

DAC をこの変換モードに設定するには ( [セクション 16.7: ノイズ生成](#) を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、デュアル DAC チャンネルデータをロードします。

トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。同時に、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

## 異なる LFSR 生成による同時トリガ

DAC をこの変換モードに設定するには ( [セクション 16.7: ノイズ生成](#) を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 2 つの DAC チャンネルの WAVEx[1:0] ビットを“01”に設定し、MAMP1[3:0] と MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DAC\_DHR12RD、DAC\_DHR12LD、または DAC\_DHR8RD) にデュアル DAC チャンネルデータをロードします。

トリガが発生すると、MAMP1[3:0] で設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC\_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

同時に、MAMP2[3:0] で設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC\_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。



## 1つの三角波生成による同時トリガ

DAC をこの変換モードに設定するには( [セクション 16.8: 三角波生成](#)を参照)、次の手順が必要です。

1. DAC チャンネル x トリガイネーブルビット TEN1x をセットします。
2. TSELx[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. DAC チャンネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. 目的の DAC\_DHRx レジスタに、DAC チャンネル x データをロードします。

トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC\_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャンネル x の三角波カウンタが更新されます。

## 異なる三角波生成による同時トリガ

DAC をこの変換モードに設定するには( [セクション 16.8: 三角波生成](#)を参照)、次の手順が必要です。

1. DAC チャンネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットに同じ値をセットすることによって、DAC チャンネル x に同じトリガソースを設定します。
3. DAC チャンネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで異なる最大振幅値を設定します。
4. 目的の DAC\_DHRx レジスタに、DAC チャンネル x データをロードします。

トリガが発生すると、MAMPx[3:0] によって設定された三角波振幅を持つ DAC チャンネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC\_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャンネル x の三角波カウンタが更新されます。

## 16.6.4 DAC 出力電圧

[セクション 16.5.3: DAC 出力電圧](#)を参照してください。

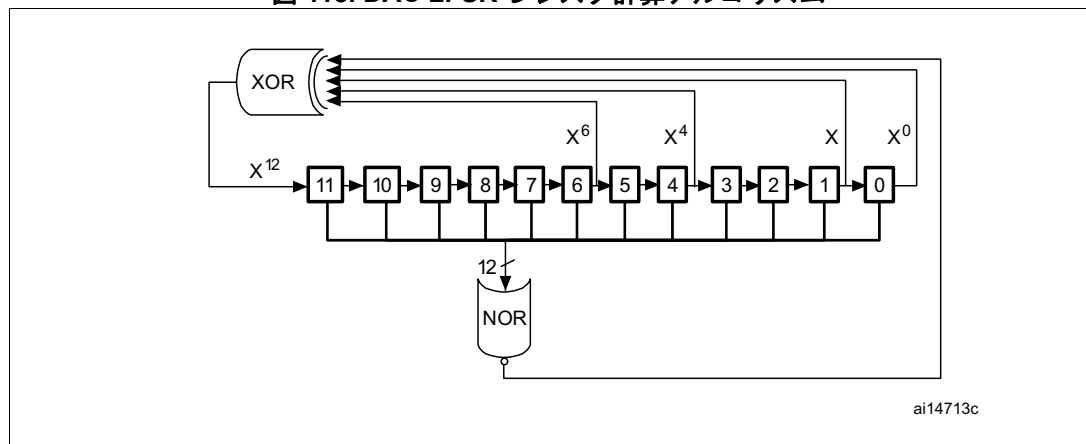
## 16.6.5 DAC トリガ選択

セクション 16.5.4 : DAC トリガ選択を参照してください。

## 16.7 ノイズ生成

リニアフィードバックシフトレジスタ (LFSR) を使用して、可変振幅の擬似ノイズを生成することができます。DAC ノイズ生成を選択するには、WAVEx[1:0] に“01”をセットします。LFSRにプリロードされる値は 0xAAA です。このレジスタは、各トリガイベントの 3 APB クロックサイクル後に、以下の特定の計算アルゴリズムに従って更新されます。

図 118. DAC LFSR レジスタ計算アルゴリズム

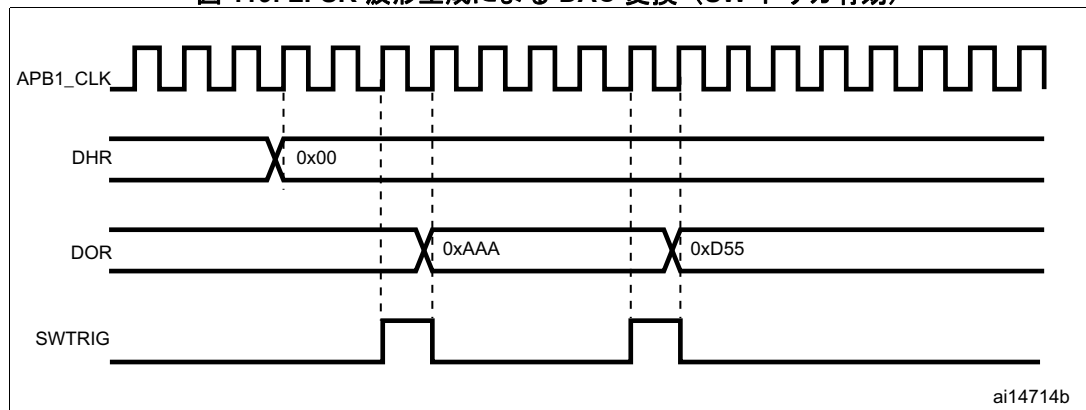


LFSR 値は、DAC\_CR レジスタの MAMPx[3:0] ビットによって部分的または全体的にマスクでき、オーバーフローなしに DAC\_DHRx の内容に加算され、DAC\_DORx レジスタに格納されます。

LFSR が 0x0000 の場合、“1”がインジェクトされます (アンチロックアップメカニズム)。

WAVEx[1:0] ビットをリセットすることによって、LFSR 波形生成をリセットできます。

図 119. LFSR 波形生成による DAC 変換 (SW トリガ有効)



注： ノイズ生成のためには、DAC\_CR レジスタの TENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

## 16.8 三角波生成

DC または低周波数信号上に、小さな振幅の三角波を追加することが可能です。DAC 三角波生成を選択するには、WAVEx[1:0] を“10” にセットします。振幅は、DAC\_CR レジスタのMAMPx[3:0] ビットを介して設定されます。内部三角波カウンタは、各トリガイベントの 3 APB クロックサイクル後にインクリメントされます。このカウンタの値は、オーバーフローなしに DAC\_DHRx レジスタに加えられ、合計は DAC\_DORx レジスタに格納されます。三角波カウンタは、MAMPx[3:0] ビットによって定義された最大振幅以上になるまでインクリメントされます。設定された振幅に達すると、カウンタは 0 にデクリメントされ、再びインクリメントが開始されます。

WAVEx[1:0] ビットをリセットすることによって、三角波生成をリセットすることができます。

図 120. DAC 三角波生成

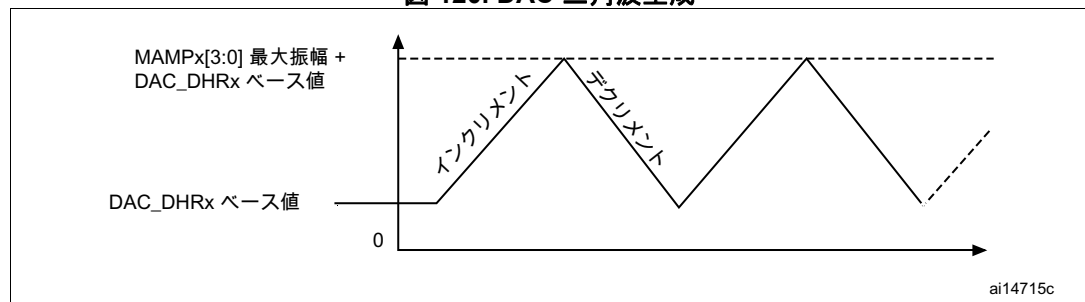
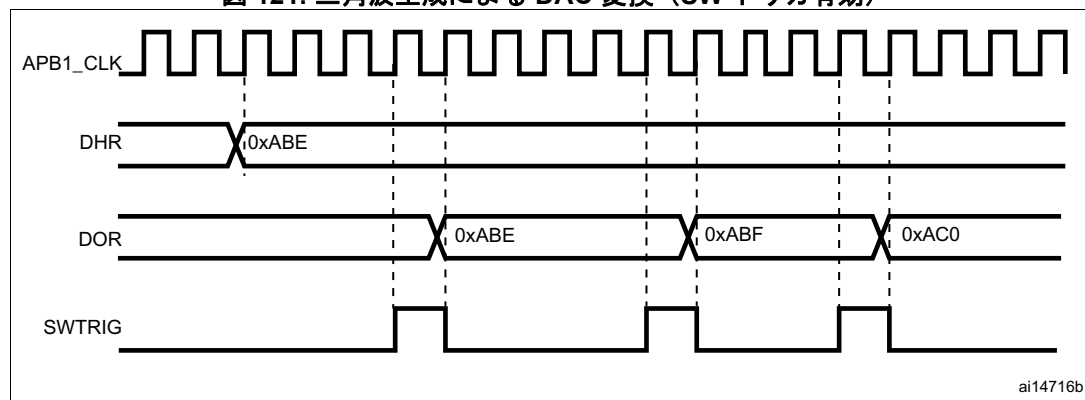


図 121. 三角波生成による DAC 変換 (SW トリガ有効)



**注:** 三角波生成のためには、DAC\_CR レジスタのTENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

DAC を有効にするには、その前に MAMPx[3:0] ビットを設定する必要があります。そうしないと、これらのビットは変更できません。

## 16.9 DMA リクエスト

各 DAC チャンネルは、DMA 機能を備えています。DAC チャンネルの DMA リクエストは、2 つの DMA チャンネルを使用して処理されます。

DAC DMA リクエストは、DMAENx ビットがセットされているときに、外部トリガ（ソフトウェアトリガでなく）が発生したときに生成されます。その場合、DAC\_DHRx レジスタの値が DAC\_DORx レジスタに転送されます。

デュアルモードでは、両方の DMAENx ビットがセットされている場合、2 つの DMA リクエストが生成されます。1 つの DMA リクエストしか必要ない場合には、対応する DMAENx ビットだけをセットしてください。これにより、アプリケーションは 1 つの DMA リクエストと一意な DMA チャンネルを使用して、両方の DAC チャンネルをデュアルモードで管理することができます。

### DMA アンダーラン

DAC DMA リクエストはキューされないため、最初の外部トリガに対する確認応答が受信される（最初のリクエスト）前に 2 番目の外部トリガが発生すると、新しいリクエストは発行されず、DAC\_SR レジスタの DMA チャンネル x アンダーランフラグ DMAUDRx がセットされてエラー状態を報告します。続いて DMA データ転送が無効になり、その後の DMA リクエストは処理されません。DAC チャンネル x は、古いデータを変換し続けます。

ソフトウェアでは、“1”を書き込むことによって DMAUDRx フラグをクリアし、使用された DMA ストリームの DMAEN ビットをクリアし、DMA と DAC のチャンネル x を再初期化して転送を正しくリスタートさせてください。また、DAC トリガ変換周波数を変更するか、または DMA の負荷を軽減して、新しい DMA アンダーランを回避してください。最後に、DMA データ転送と変換トリガを有効にすることによって DAC 変換を再開することができます。

各 DAC チャンネルでは、DAC\_CR レジスタの対応する DMAUDRIEx ビットが有効にされた場合、割り込みも生成されます。

## 16.10 DAC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

### 16.10.1 DAC 制御レジスタ (DAC\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAU DRIE2	DMA EN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2 /OUTE N2	EN2
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAU DRIE1	DMA EN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[2:0]			TEN1	BOFF1 /OUTE N1	EN1
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DMAUDRIE2** : DAC チャンネル2 DMA アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル2 DMA アンダーラン割り込みは無効です。

1 : DAC チャンネル2 DMA アンダーラン割り込みは有効です。

**注 :** このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 28 **DMAEN2** : DAC チャンネル2 DMA イネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル2 DMA モードは無効です。

1 : DAC チャンネル2 DMA モードは有効です。

**注 :** このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 27:24 **MAMP2[3:0]** : DAC チャンネル 2 マスク/振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

0000 : LFSR/三角波振幅のアンマスクビット 0 は 1 に等しい。

0001 : LFSR/三角波振幅のアンマスクビット [1:0] は 3 に等しい。

0010 : LFSR/三角波振幅のアンマスクビット [2:0] は 7 に等しい。

0011 : LFSR/三角波振幅のアンマスクビット [3:0] は 15 に等しい。

0100 : LFSR/三角波振幅のアンマスクビット [4:0] は 31 に等しい。

0101 : LFSR/三角波振幅のアンマスクビット [5:0] は 63 に等しい。

0110 : LFSR/三角波振幅のアンマスクビット [6:0] は 127 に等しい。

0111 : LFSR/三角波振幅のアンマスクビット [7:0] は 255 に等しい。

1000 : LFSR/三角波振幅のアンマスクビット [8:0] は 511 に等しい。

1001 : LFSR/三角波振幅のアンマスクビット [9:0] は 1023 に等しい。

1010 : LFSR/三角波振幅のアンマスクビット [10:0] は 2047 に等しい。

≥1011 : LFSR/三角波振幅のアンマスクビット [11:0] は 4095 に等しい。

**注 :** これらのビットは、波形生成がサポートされるときにデュアルモードでのみ使用できます。それ以外の場合は予約済みであり、リセット値に保持する必要があります。



ビット 23:22 **WAVE2[1:0]** : DAC チャンネル 2 ノイズ/三角波生成イネーブル

これらのビットは、ソフトウェアによってセット/リセットされます。

- 00 : 波形生成は無効です。
- 01 : ノイズ波生成は有効です。
- 1x : 三角波生成は有効です。

**注 :** ビット  $TEN2=1$  (DAC チャンネル2 トリガ有効) の場合のみ使用されます。

これらのビットは、波形生成がサポートされる時にデュアルモードでのみ使用できます。それ以外の場合は予約済みであり、リセット値に保持する必要があります。

ビット 21:19 **TSEL2[2:0]** : DAC チャンネル2 トリガ選択

これらのビットは、DAC チャンネル2 をトリガするために使用される外部イベントを選択します。

- 000 : タイマ 6 TRGO イベント
- 001 : SYSCFG\_CFGR1 レジスタの DAC\_TRIG\_RMP ビットの値に応じたタイマ 3 またはタイマ 8 TRGO イベント
- 010 : タイマ 7 TRGO イベント
- 011 : タイマ 15 TRGO イベント
- 100 : タイマ 2 TRGO イベント
- 101 : タイマ 4 TRGO イベント
- 110 : EXTI ライン 9
- 111 : ソフトウェアトリガ

**注 :** ビット  $TEN2=1$  (DAC チャンネル2 トリガ有効) の場合のみ使用されます。

このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 18 **TEN2** : DAC チャンネル2 トリガイネーブル

このビットは、DAC チャンネル2 トリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 2 トリガは無効であり、DAC\_DHRx レジスタに書き込まれたデータは、1 APB1 クロックサイクル後に DAC\_DOR2 レジスタに転送されます。
- 1 : DAC チャンネル 2 トリガは有効であり、DAC\_DHRx レジスタから転送されたデータは、3 APB1 クロックサイクル後に DAC\_DOR2 レジスタに転送されます。

**注 :** ソフトウェアトリガが選択されているときには、DAC\_DHRx から DAC\_DOR2 レジスタへの転送には、わずか 1 APB1 クロックサイクルで行われます。

**注 :** このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 17 **STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE :**

**BOFF2** : DAC チャンネル2 出力バッファディセーブル

このビットは、DAC チャンネル2 出力バッファを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル2 出力バッファは有効です。
- 1 : DAC チャンネル2 出力バッファは無効です。

**注 :** このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

**STM32F303x6/8 および STM32F328x8 DAC1 :**

**OUTEN2** : DAC チャンネル 2 出力スイッチイネーブル

このビットは、DAC チャンネル 2 出力スイッチを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 2 出力スイッチは無効です。
- 1 : DAC チャンネル 2 出力スイッチは有効です。

**ビット 16 EN2** : DAC チャンネル2 イネーブル

このビットは、DAC チャンネル2 を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル2 は無効です。
- 1 : DAC チャンネル2 は有効です。

**注 :** このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

**ビット 13 DMAUDRIE1** : DAC チャンネル 1 DMA アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 1 DMA アンダーラン割り込みは無効です。
- 1 : DAC チャンネル 1 DMA アンダーラン割り込みは有効です。

**ビット 12 DMAEN1** : DAC チャンネル 1 DMA イネーブル

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 1 DMA モードは無効です。
- 1 : DAC チャンネル 1 DMA モードは有効です。

**ビット 11:8 MAMP1[3:0]** : DAC チャンネル 1 マスク/振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

- 0000 : LFSR/三角波振幅のアンマスクビット 0 は 1 に等しい。
- 0001 : LFSR/三角波振幅のアンマスクビット [1:0] は 3 に等しい。
- 0010 : LFSR/三角波振幅のアンマスクビット [2:0] は 7 に等しい。
- 0011 : LFSR/三角波振幅のアンマスクビット [3:0] は 15 に等しい。
- 0100 : LFSR/三角波振幅のアンマスクビット [4:0] は 31 に等しい。
- 0101 : LFSR/三角波振幅のアンマスクビット [5:0] は 63 に等しい。
- 0110 : LFSR/三角波振幅のアンマスクビット [6:0] は 127 に等しい。
- 0111 : LFSR/三角波振幅のアンマスクビット [7:0] は 255 に等しい。
- 1000 : LFSR/三角波振幅のアンマスクビット [8:0] は 511 に等しい。
- 1001 : LFSR/三角波振幅のアンマスクビット [9:0] は 1023 に等しい。
- 1010 : LFSR/三角波振幅のアンマスクビット [10:0] は 2047 に等しい。
- ≥ 1011 : LFSR/三角波振幅のアンマスクビット [11:0] は 4095 に等しい。

**注 :** これらのビットは、波形生成機能がサポートされるときにのみ使用できます。それ以外の場合は予約済みであり、リセット値に保持する必要があります。

**ビット 7:6 WAVE1[1:0]** : DAC チャンネル 1 ノイズ/三角波生成イネーブル

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 波形生成は無効です。
- 01 : ノイズ波生成は有効です。
- 1x : 三角波生成は有効です。

**注 :** ビット  $TEN1 = 1$  (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

**注 :** これらのビットは、波形生成機能がサポートされるときにのみ使用できます。それ以外の場合は予約済みであり、リセット値に保持する必要があります。

**ビット 5:3 TSEL1[2:0]** : DAC チャンネル 1 トリガ選択

これらのビットは、DAC チャンネル 1 をトリガするために使用される外部イベントを選択します。

000 : タイマ 6 TRGO イベント

001 : SYSCFG\_CFGR1 レジスタの DAC\_TRIG\_RMP ビットの値に応じたタイマ 3 またはタイマ 8 TRGO イベント

010 : タイマ 7 TRGO イベント

011 : タイマ 15 TRGO イベント

100 : タイマ 2 TRGO イベント

101 : タイマ 4 TRGO イベント

110 : EXTI ライン 9

111 : ソフトウェアトリガ

**注:** ビット  $TEN1 = 1$  (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

**ビット 2 TEN1** : DAC チャンネル 1 トリガイネーブル

このビットは、DAC チャンネル 1 トリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 トリガは無効であり、DAC\_DHRx レジスタに書き込まれたデータは、1 APB1 クロックサイクル後に DAC\_DOR1 レジスタに転送されます。

1 : DAC チャンネル 1 トリガは有効であり、DAC\_DHRx レジスタから転送されたデータは、3 APB1 クロックサイクル後に DAC\_DOR1 レジスタに転送されます。

**注:** ソフトウェアトリガが選択されているときには、DAC\_DHRx レジスタから DAC\_DOR1 レジスタへの転送は、わずか 1 APB1 クロックサイクルで行われます。

**ビット 1 DAC1 :**

**BOFF1** : DAC チャンネル 1 出力バッファディセーブル

このビットは、DAC チャンネル 1 出力バッファを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 出力バッファは有効です。

1 : DAC チャンネル 1 出力バッファは無効です。

**DAC2 : (STM32F303x6/8 および STM32F328x8 のみ)**

**OUTEN1** : DAC チャンネル 1 出力スイッチイネーブル

このビットは、DAC チャンネル 1 出力スイッチを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 出力スイッチは無効です。

1 : DAC チャンネル 1 出力スイッチは有効です。

**ビット 0 EN1** : DAC チャンネル 1 イネーブル

このビットは、DAC チャンネル 1 を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル 1 は無効です。

1 : DAC チャンネル 1 は有効です。

## 16.10.2 DAC ソフトウェアトリガレジスタ (DAC\_SWTRIGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
														w	w

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SWTRIG2** : DAC チャンネル2 ソフトウェアトリガ

このビットは、ソフトウェアトリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : ソフトウェアトリガは無効です。

1 : ソフトウェアトリガは有効です。

**注 :** このビットは、DAC\_DHR2 レジスタの値が DAC\_DOR2 レジスタにロードされると、ハードウェアによってクリアされます (1 APB1 クロックサイクル後)。

このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 0 **SWTRIG1** : DAC チャンネル 1 ソフトウェアトリガ

このビットは、ソフトウェアトリガを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : ソフトウェアトリガは無効です。

1 : ソフトウェアトリガは有効です。

**注 :** このビットは、DAC\_DHR1 レジスタの値が DAC\_DOR1 レジスタにロードされると、ハードウェアによってクリアされます (1 APB1 クロックサイクル後)。

## 16.10.3 DAC チャンネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC\_DHR12R1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.10.4 DAC チャンネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC\_DHR12L1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												v	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 16.10.5 DAC チャンネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC\_DHR8R1)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 8 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.10.6 DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC\_DHR12R2)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.10.7 DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC\_DHR12L2)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 16.10.8 DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC\_DHR8R2)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DACC2DHR[7:0]** : DAC チャンネル2 8ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.10.9 デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC\_DHR12RD)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.10.10 デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC\_DHR12LD)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w				

ビット 31:20 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。



## 16.10.11 デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC\_DHR8RD)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DACC2DHR[7:0]** : DAC チャンネル2 8ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル1 の 8ビット右詰めデータ

これらのビットは、DAC チャンネル1 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

## 16.10.12 DAC チャンネル 1 データ出力レジスタ (DAC\_DOR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DOR[11:0]** : DAC チャンネル1 データ出力

これらのビットは読み出し専用であり、DAC チャンネル1 のデータ出力を含みます。

## 16.10.13 DAC チャンネル 2 データ出力レジスタ (DAC\_DOR2)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				DACC2DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r



ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DOR[11:0]** : DAC チャンネル2 データ出力

これらのビットは読み出し専用であり、DAC チャンネル2 のデータ出力を含みます。

## 16.10.14 DAC ステータスレジスタ (DAC\_SR)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rc_w1													

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DMAUDR2** : DAC チャンネル 2 DMA アンダーランフラグ

このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。

0 : DAC チャンネル2 に DMA アンダーランエラー条件は発生しませんでした。

1 : DAC チャンネル2 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル2 変換を駆動しています)。

**注 :** このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 28:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **DMAUDR1** : DAC チャンネル 1 の DMA アンダーランフラグ

このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。

0 : DAC チャンネル 1 に DMA アンダーランエラー条件は発生しませんでした。

1 : DAC チャンネル1 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル 1 変換を駆動しています)。

ビット 12:0 予約済みであり、リセット値に保持する必要があります。

## 16.10.15 DAC レジスタマップ

表 106 に DAC レジスタの要約を示します。

表 106. DAC レジスタマップ とリセット値

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DAC_CR	Res.	Res.	DMAUDRIE2	DMAEN2		MAMP2[3:0]			WAVE2[1:0]		TSEL2[2:0]		TEN2	BOFF2	EN2	Res.	Res.	DMAUDRIE1	DMAEN1		MAMP1[3:0]		WAVE1[1:0]		TSEL1[2:0]		TEN1	BOFF1	EN1			
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	DAC_SWTRIGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																															0	0
0x08	DAC_DHR12R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																							0	0	0	0	0	0	0	0	0	0
0x0C	DAC_DHR12L1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	DAC_DHR8R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																0
0x14	DAC_DHR12R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																							0	0	0	0	0	0	0	0	0	0
0x18	DAC_DHR12L2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	DAC_DHR8R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																0
0x20	DAC_DHR12RD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x24	DAC_DHR12LD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x28	DAC_DHR8RD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	DAC_DOR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																							0	0	0	0	0	0	0	0	0	0
0x30	DAC_DOR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																							0	0	0	0	0	0	0	0	0	0

表 106. DAC レジスタマップ (続き) とリセット値 (続き)

オフセット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	DAC_SR	Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値			0																0													

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。

## 17 コンパレータ (COMP)

### 17.1 概要

STM32F302xB/C/D/E および STM32F302x6/8 は 7 個の汎用コンパレータを内蔵しており、それらはスタンダードデバイス (すべての端子を I/O で使用可能) として使用するか、タイマと組み合わせることができます。STM32F303x6/8 および STM32F328x8 は 3 個の汎用コンパレータ (COMP2、COMP4、および COMP6) を内蔵しています。

コンパレータは以下のようなさまざまな機能に使用できます。

- アナログ信号によってトリガされる低電力モードからのウェイクアップ
- アナログ信号調節
- DAC と タイマからの PWM 出力を組み合わせた場合のサイクルごとの電流制御ループ

### 17.2 COMP の主な機能

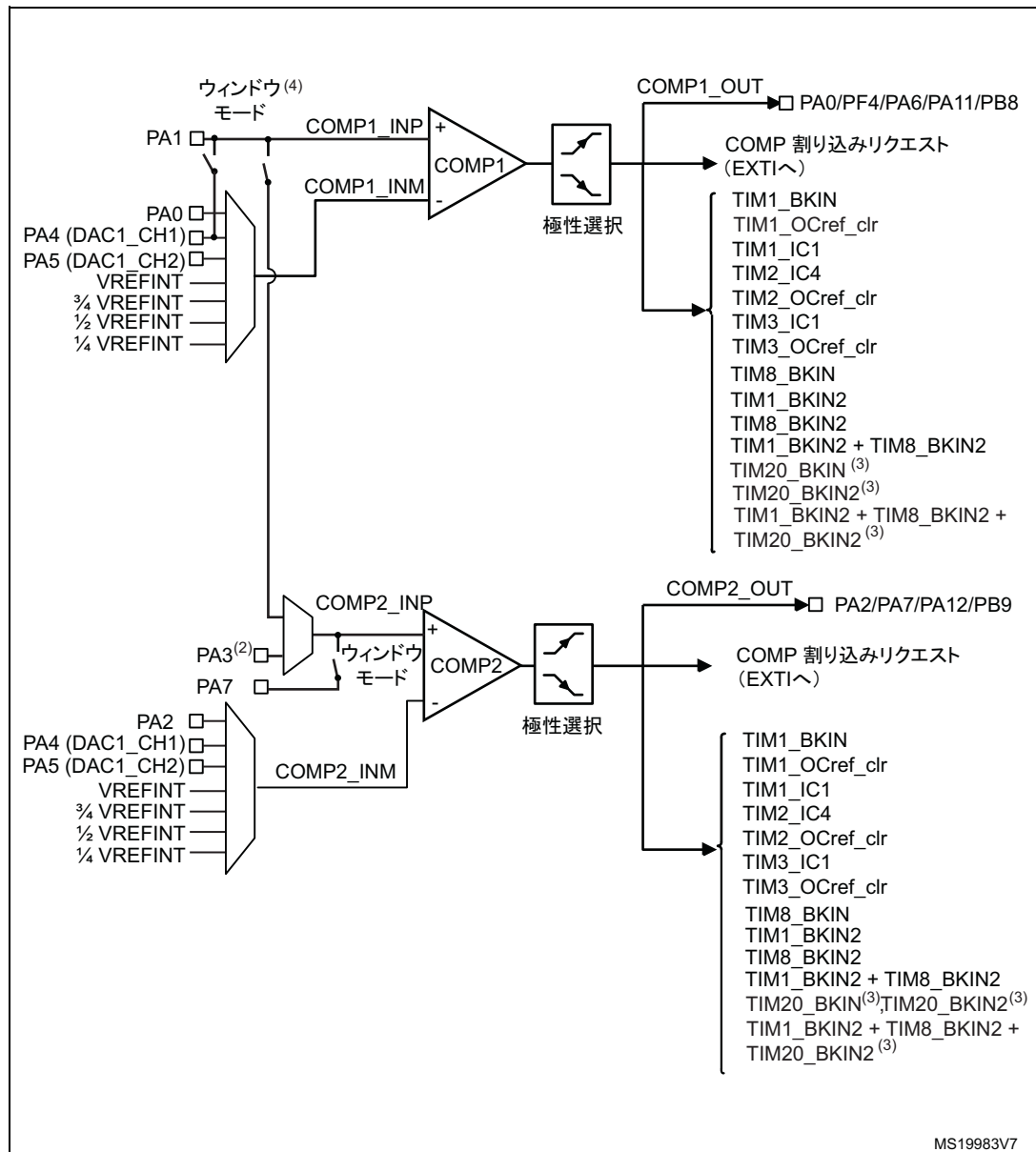
- レールツーレールコンパレータ
- 各コンパレータは、電圧を柔軟に選択できるように、次のような正入力および設定可能な負入力を備えています。
  - マルチプレクス I/O ピン
  - STM32F303x6/8 および STM32F328x8 デバイスの DAC1 チャンネル 1、DAC1 チャンネル 2、DAC2 チャンネル 1、および STM32F302xB/C/D/E および STM32F302x6/8 デバイスの DAC1 チャンネル 1、DAC1 チャンネル 2
  - スケラ (バッファ付き分圧器) が提供する内部基準電圧および 3 つの約数 (1/4、1/2、3/4)
- プログラム可能なヒステリシス (STM32F303xB/C および STM32F358xC のみ)
- プログラム可能なスピード/消費電力 (STM32F303xB/C および STM32F358xC のみ)
- 出力先を I/O またはトリガに使用するタイマ入力に変更することができます。
  - キャプチャイベント
  - OCREF\_CLR イベント (サイクルごとの電流制御)
  - 高速 PWM 停止のブレーキイベント
- COMP1/COMP2、COMP3/COMP4、および COMP5/COMP6 コンパレータをウィンドウコンパレータで結合することができます。これは、STM32F303xB/C および STM32F358xC デバイスにのみ適用されます。COMP7 ではウィンドウモードをサポートしていません。
- コンパレータはブランキングソースとともに出力されます。
- 各コンパレータは、SLEEP および STOP モードからのウェイクアップ (EXTI コントローラ経由) を備えた割り込み生成機能を持っています。

## 17.3 COMP の機能説明

### 17.3.1 COMP ブロック図

コンパレータのブロック図を図 122: コンパレータ 1 および 2 のブロック図 (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE) および 図 123: STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE コンパレータ 7 のブロック図に示します。

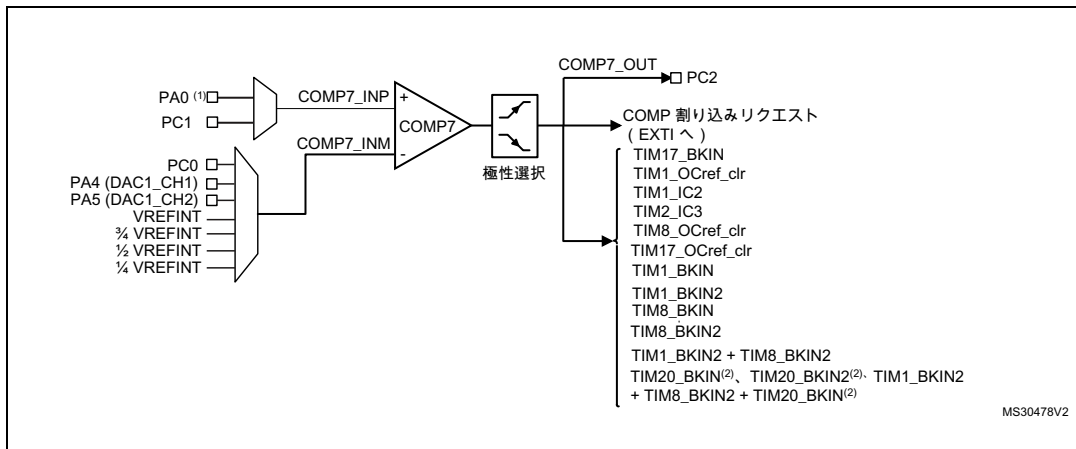
図 122. コンパレータ 1 および 2 のブロック図  
(STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE)



1. コンパレータ 1 から 6 の完全なブロック図については、[セクション 18: オペアンプ \(OPAMP\)](#) を参照してください。ここでは、コンパレータ 1 から 6 とオペアンプの間にあるすべてのブロック図と相互接続を示します。
2. STM32F303xB/C および STM32F358xC のみ。
3. STM32F303xDxE のみ。

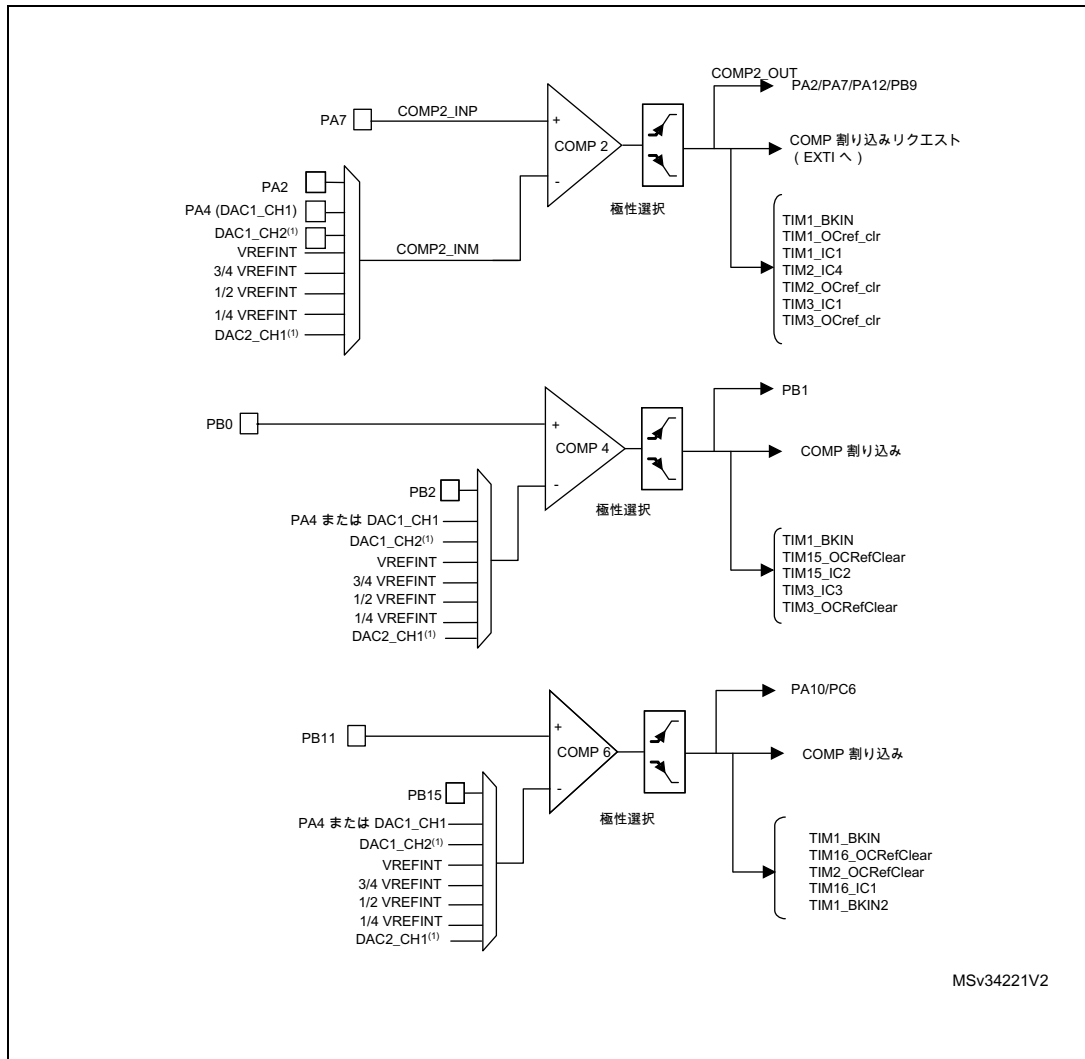
4. ウィンドウモードは、STM32F303xD/E および STM32F398xE ではサポートされていません。

図 123. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE コンパレータ 7 のブロック図



1. PA0 は STM32F303xD/E の COMP7\_INP には使用できません。
2. STM32F303xD/E および STM32F398xE のみ。

図 124. STM32F303x6/8 および STM32F328x8 コンパレータ 2/4/6 のブロック図



1. STM32F303x6/8 および STM32F328x8 デバイスでは、DAC1\_CH2 および DAC2\_CH1 出力は直接接続されているため、PA5 および PA6 は COMPx\_INM (x=2, 4, 6) 入力では使用できません。DAC1\_OUT2 および DAC2\_OUT1 がコンパレータの非反転入力に内部的に接続している場合、DAC1\_OUT および DAC2\_OUT が配置された I/O (PA5 および PA6) を GPIO として使用できます。

## 17.3.2 COMP ピンおよび内部信号

コンパレータ入力として使用される I/O は、GPIO レジスタのアナログモードで設定する必要があります。

コンパレータの出力は、データシートの「オルタネート機能配置」表に示されているオルタネート機能チャンネルを使用して I/O に接続することができます。

次の表に、コンパレータの入出力として使用できる I/O を示します。

出力先を、以下の目的に使用される各種タイマ入力に内部で変更することも可能です。

- BKIN および BKIN2 入力を使用した PWM 信号の緊急停止
- OCREF\_CLR 入力を使用したサイクルごとの電流制御
- タイミング測定のための入力キャプチャ



コンパレータの出力先を内部および外部から一斉に変更することが可能です。

表 107. コンパレータの入出力の概要

	コンパレータの入出力						
	COMP1 <sup>(3)</sup>	COMP2	COMP3 <sup>(3)</sup>	COMP4	COMP5 <sup>(3)</sup>	COMP6	COMP7 <sup>(3)</sup>
コンパレータの反転入力：内部信号への接続	DAC1_CH1 DAC1_CH2 DAC2_CH1 <sup>(1)</sup> Vrefint ¾ Vrefint ½ Vrefint ¼ Vrefint						
I/O に接続されたコンパレータの入力 (+: 非反転入力、 -: 反転入力)	+: PA1 -: PA0	+: PA3 <sup>(2)</sup> +: PA7 -: PA2	-: PB12 -: PD15 <sup>(3)</sup> +: PB14 +: PD14 <sup>(2)</sup>	+: PB0 +: PE7 <sup>(2)</sup> -: PB2 -: PE8 <sup>(3)</sup>	-: PB10 -: PD13 <sup>(3)</sup> +: PB13 +: PD12 <sup>(2)</sup>	+: PB11 +: PD11 <sup>(3)</sup> -: PB15 -: PD10 <sup>(3)</sup>	+: PC1 +: PA0 <sup>(2)</sup> -: PC0
コンパレータの出力 (モータ制御保護)	T1BKIN T1BKIN2 T8BKIN <sup>(3)</sup> T8BKIN2 <sup>(3)</sup> T1BKIN2+ T8BKIN2 <sup>(3)</sup> TIM20BKIN <sup>(4)</sup> TIM20BKIN2 <sup>(4)</sup> TIM1BKIN2 + TIM8BKIN2 + TIM20BKIN2 <sup>(4)</sup>						
I/O での出力	PA0 PF4 PA6 PA11 PB8	PA2 PA7 <sup>(2)</sup> PA12 PB9	PC8 PA8	PB1	PC7 PA9	PA10 PC6	PC2
内部信号への出力	TIM1_OCrefClear TIM1_IC1 TIM2_IC4 TIM2_OCrefClear TIM3_IC1 TIM3_OCrefClear	TIM1_OCrefClear TIM2_OCrefClear TIM3_IC2 TIM4_IC1 <sup>(3)</sup> TIM15_IC1 TIM15_BKIN	TIM8_OCrefClear <sup>(3)</sup> TIM3_IC3 TIM3_OCrefClear TIM4_IC2 <sup>(3)</sup> TIM15_OCrefClear TIM15_IC2	TIM8_OCrefClear <sup>(3)</sup> TIM2_IC1 TIM3_OCrefClear TIM4_IC3 <sup>(3)</sup> TIM16_BKIN TIM17_IC1	TIM8_OCrefClear <sup>(3)</sup> TIM2_IC2 TIM2_OCrefClear TIM16_OCrefClear TIM16_IC1 TIM4_IC4 <sup>(3)</sup>	TIM1_OCrefClear TIM8_OCrefClear <sup>(3)</sup> TIM2_IC3 TIM1_IC2 TIM17_OCrefClear TIM17_BKIN	

1. STM32F303x6/8 および STM32F328x8 デバイスのみ。
2. STM32F303xB/C および STM32F358xC デバイスのみ
3. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ。
4. STM32F303xD/E および STM32F398xE デバイスのみ



### 17.3.3 COMP のリセットおよびクロック

クロックコントローラによって提供される COMP クロックは、PCLK2 (APB2 クロック) と同期しています。

RCC コントローラには、クロックイネーブル制御ビットは提供されていません。コンパレータにクロックソースを使用するには、RCC コントローラに SYSCFG クロックイネーブル制御ビットをセットする必要があります。

**注:** **重要: 極性選択ロジックおよびポートへの出力先変更は、PCLK2 クロックから個別に機能します。これにより、コンパレータは STOP モードでも機能することができます。**

### 17.3.4 コンパレータのロック機構

コンパレータは、過電流保護や熱保護などの安全上の目的で使用されます。特定の機能安全要件があるようなアプリケーションの場合、万が一誤ったレジスタへのアクセスやプログラムカウンタの破壊が起こった場合に、コンパレータのプログラミングが絶対に変更されないようにすることが必要です。

このような目的で、コンパレータの制御レジスタおよびステータスレジスタを書き込み保護することができます (読み出し専用)。

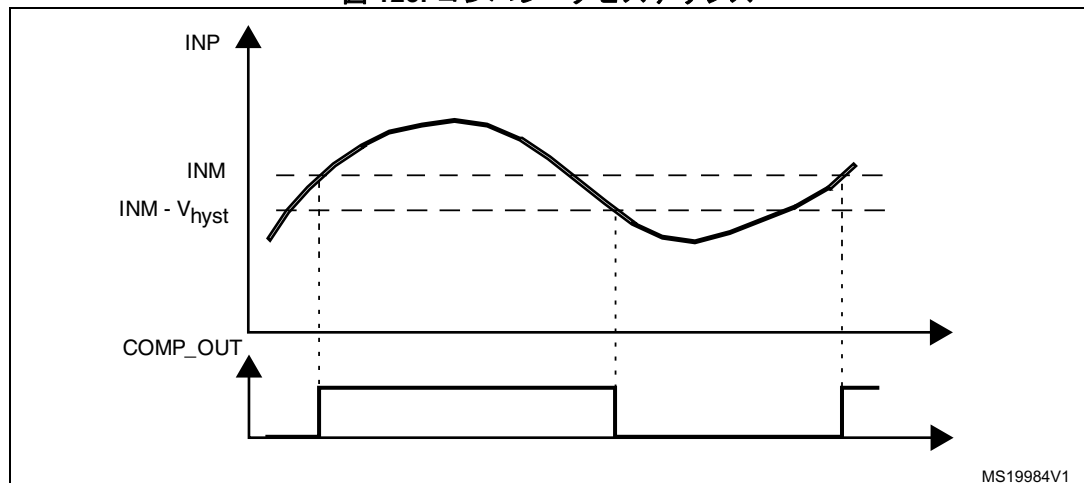
プログラミングが完了したら、COMPx\_CSR の 30:0 ビットを使用して、COMPxLOCK ビットを 1 にセットすることができます。これにより、COMPxLOCK ビットを含む COMPx\_CSR レジスタ全体が読み出し専用になります。

書き込み保護は MCU リセットによってのみリセット可能です。

### 17.3.5 ヒステリシス (STM32F303xB/C および STM32F358xC のみ)

コンパレータには、ノイズの多い信号での疑似出力遷移を避けるために、プログラム可能なヒステリシスが含まれています。不要な場合は、ヒステリシスを無効にすることができます (低電力モードを終了する場合など)。これにより、外部コンポーネントを使用してヒステリシス値を強制できます。

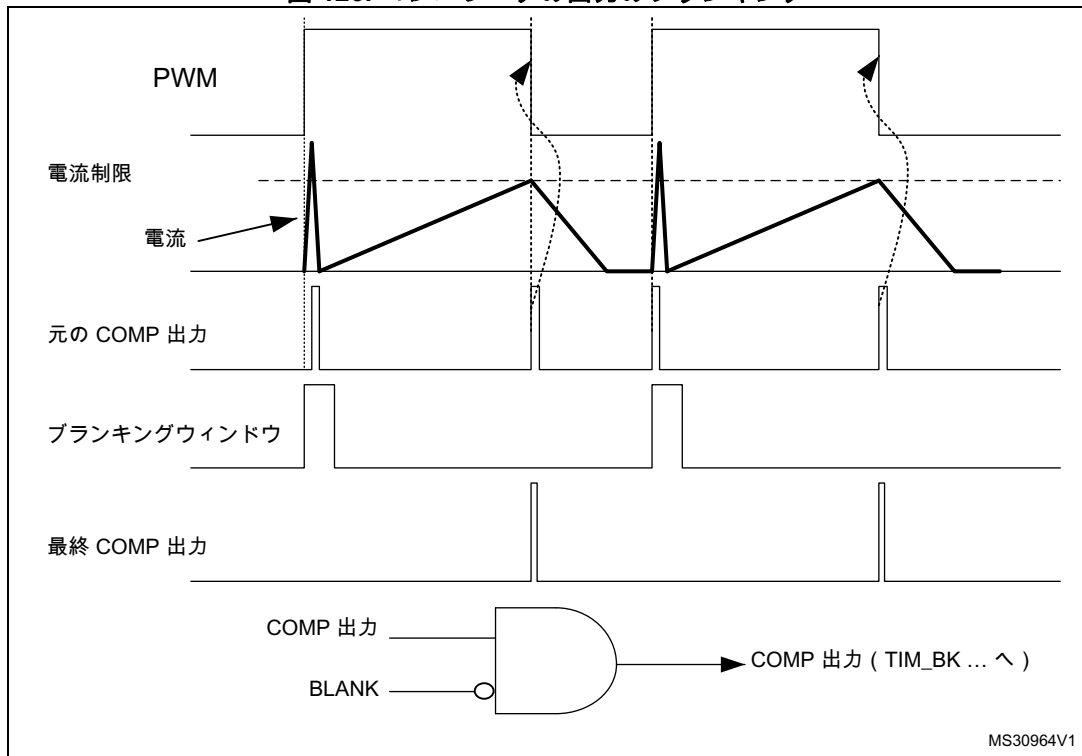
図 125. コンパレータヒステリシス



## 17.3.6 コンパレータの出力のブランキング機能

ブランキング機能の目的は、PWM 周期の開始時に短絡電流スパイクでトリップするために現在のレギュレーションを防ぐことです（通常は電源スイッチのアンチパラレルダイオードのリカバリ電流）。タイマ出力比較信号であるブランキングウィンドウの選択から構成されています。選択はソフトウェアで行われます（考えられるブランキング信号のコンパレータレジスタの説明を参照）。次に、必要なコンパレータの出力を供給するために、相補ブランキング信号とコンパレータの出力の論理積が取られます。次の図に示す例を参照してください。

図 126. コンパレータの出力のブランキング



## 17.3.7 電力モード (STM32F303xB/C および STM32F358xC のみ)

コンパレータの電力消費と伝搬遅延を調節して、特定のアプリケーションに最適なトレードオフを実現させることができます。COMPx\_CSR レジスタの COMPxMODE[1:0] ビットは次のようにプログラムできます。

- 00 : 高速 / フル電力
- 01 : 中速 / 中電力
- 10 : 低速 / 低電力
- 11 : 超低速 / 超低電力

## 17.4 COMP 割り込み

コンパレータの出力は拡張割り込み / イベントコントローラに内部的に接続されます。各コンパレータには専用の EXTI ラインがあり、割り込みまたはイベントを生成することができます。低電力モードを終了するときにも同じ方法が使用されます。

詳細については、割り込みおよびイベントのセクションを参照してください。

## 17.5 COMP レジスタ

### 17.5.1 COMP1 制御およびステータスレジスタ (COMP1\_CSR)

注: このレジスタは、STM32F303xB/C/D/E、STM32F358xC、およびSTM32F398xE のみ使用可能です。

アドレスオフセット: 0x1C

リセット値: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP1 LOCK	COMP1 OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP1_BLANKING			COMP1HYST [1:0] <sup>(1)</sup>	
rwo	r										rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP1 POL	Res.	COMP1OUTSEL				Res.	Res.	Res.	COMP1INMSEL[2:0]			COMP1MODE [1:0] <sup>(1)</sup>		COMP1_INP_DAC	COMP1EN
rw		rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw

1. STM32F303xB/C および STM32F358xC のみ。STM32F303xD/E では使用できません。

#### ビット 31 COMP1LOCK: コンパレータ 1 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP1\_CSR レジスタを読み出し専用にすることができます。

- 0: COMP1\_CSR は読み書き用です。
- 1: COMP1\_CSR は読み出し専用です。

#### ビット 30 COMP1OUT: コンパレータ 1 出力

この読み出し専用ビットはコンパレータ 1 の出力状態のコピーです。

- 0: 出力はロー (反転入力を下回る非反転入力)
- 1: 出力はハイ (反転入力を上回る非反転入力)

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

#### ビット 20:18 COMP1\_BLANKING: コンパレータ 1 ブランキングソース

これらのビットによってコンパレータ 1 出力のブランキングを制御するタイマ出力を選択します。

- 000: ブランキングなし
- 001: TIM1 OC5 がブランキングソースとして選択されます。
- 010: TIM2 OC3 がブランキングソースとして選択されます。
- 011: TIM3 OC3 がブランキングソースとして選択されます。
- 他の設定: 予約済み

#### ビット 17:16 COMP1HYST[1:0] コンパレータ 1 ヒステリシス

STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE では、これらのビットがヒステリシスレベルを制御します。

- 00: ヒステリシスなし
- 01: 低ヒステリシス
- 10: 中ヒステリシス
- 11: 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303xD/E、STM32F303x6/8、および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。



ビット 15 **COMP1POL** : コンパレータ 1 出力の極性

このビットは、コンパレータ 1 出力を反転させるために使用します。

- 0 : 出力は反転されません。
- 1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **COMP1OUTSEL[3:0]** : コンパレータ 1 出力選択

これらのビットによってコンパレータ 1 出力に接続するタイマ入力を選択します。

- 0000 : 選択なし
- 0001 : (BRK\_ACTH) タイマ 1 ブレーク入力
- 0010 : (BRK2) タイマ 1 ブレーク入力 2
- 0011 : タイマ 8 ブレーク入力 1
- 0100 : タイマ 8 ブレーク入力 2
- 0101 : タイマ 1 ブレーク入力 2 + タイマ 8 ブレーク入力 2
- 0110 : タイマ 1 OCrefclear 入力
- 0111 : タイマ 1 入力キャプチャ 1
- 1000 : タイマ 2 入力キャプチャ 4
- 1001 : タイマ 2 OCrefclear 入力
- 1010 : タイマ 3 入力キャプチャ 1
- 1011 : タイマ 3 OCrefclear 入力
- 1100 : タイマ 20 ブレーク入力 1 (注)
- 1101 : タイマ 20 ブレーク入力 2 (注)
- 1110 : タイマ 1 ブレーク入力 2 + タイマ 8 ブレーク入力 2 + タイマ 20 ブレーク入力 2 (注)
- 1111 : 予約済み。

注 : STM32F303xD/E デバイスのみ。

ビット 9:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **COMP1INMSEL[2:0]** : コンパレータ 1 反転入力選択

これらのビットにより、コンパレータ 1 の反転入力に接続されたソースを選択できます。

- 000 : Vrefint の 1/4
- 001 : Vrefint の 1/2
- 010 : Vrefint の 3/4
- 011 : Vrefint
- 100 : PA4 または DAC1 出力 (有効な場合)
- 101 : PA5 または DAC2 出力 (有効な場合)
- 110 : PA0
- 111 : 予約済み

ビット 3:2 **COMP1MODE[1:0]** : コンパレータ 1 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 1 の動作モードを制御し、スピード/消費電力を調節できます。

- 00 : ハイスピード
- 01 : ミディウムスピード
- 10 : 低電力
- 11 : 超低電力

ビット 1 **COMP1\_INP\_DAC** : DAC 出力へのコンパレータ 1 非反転入力接続

このビットは、コンパレータ 1 非反転入力 (PA0) と DAC 出力 I/O (PA4) 間のスイッチを閉じます。

- 0 : スイッチを開きます。
- 1 : スイッチを閉じます。

**注 :** このスイッチは、COMP1 非反転入力 (高抵抗スイッチ) などの信号をハイインピーダンス入力に変更することのみを目的としています。

ビット 0 **COMP1EN** : コンパレータ 1 イネーブル

このビットは COMP1 のオン/オフを切り替えます。

- 0 : コンパレータ 1 無効
- 1 : コンパレータ 1 有効

## 17.5.2 COMP2 制御およびステータスレジスタ (COMP2\_CSR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP2LOCK	COMP2OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP2INMSEL[3] <sup>(1)</sup>	Res.	COMP2_BLANKING[2:0]			COMP2HYST [1:0] <sup>(2)</sup>	
rw	r								rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP2POL	Res.	COMP2OUTSEL[3:0]				COMP2WINMODE <sup>(3)</sup>	Res.	COMP2INPSEL <sup>(2)</sup>	COMP2INMSEL[2:0]			COMP2MODE [1:0] <sup>(2)</sup>		Res.	COMP2EN
rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw		rw

1. STM32F303x6/8 および STM32F328x8 のみ。
2. STM32F303xB/C および STM32F358xC デバイスのみ。
3. STM32F303x6/8/D/E、STM32F328x8、および STM32F398xE デバイスでは使用不可。

ビット 31 **COMP2LOCK** : コンパレータ 2 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP2\_CSR レジスタを読み出し専用にすることができます。

- 0 : COMP2\_CSR は読み書き用です。
- 1 : COMP2\_CSR は読み出し専用です。

ビット 30 **COMP2OUT** : コンパレータ 2 出力

この読み出し専用ビットはコンパレータ 1 の出力状態のコピーです。

- 0 : 出力はロー (反転入力を下回る非反転入力)
- 1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **COMP2INMSEL[3]** : コンパレータ 2 反転入力選択。このビットは、F303x6/x8 および F328xx でのみ使用できます。

これは、COMP の反転入力を設定するために、ビット [6..4] とともに使用されます。



ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20:18 **COMP2\_BLANKING[2:0]** : コンパレータ 2 出力ブランキングソース

これらのビットによってコンパレータ 1 出力のブランキングを制御するタイマ出力を選択します。

000 : ブランキングなし

001 : TIM1 OC5 がブランキングソースとして選択されます。

010 : TIM2 OC3 がブランキングソースとして選択されます。

011 : TIM3 OC3 がブランキングソースとして選択されます。

他の設定 : 予約済み

ビット 17:16 **COMP2HYST[1:0]** : コンパレータ 2 ヒステリシス

STM32F303xB/C および STM32F358xC では、これらのビットがヒステリシスレベルを制御します。

00 : ヒステリシスなし

01 : 低ヒステリシス

10 : 中ヒステリシス

11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303xD/E、STM32F303x6/8、および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。

ビット 15 **COMP2POL** : コンパレータ 2 出力の極性

このビットは、コンパレータ 2 出力を反転させるために使用します。

0 : 出力は反転されません。

1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **COMP2OUTSEL[3:0]** : コンパレータ 2 出力選択

これらのビットによってコンパレータ 2 出力に接続するタイマ入力を選択します。

0000 : 選択なし

0001 : (BRK\_ACTH) タイマ 1 ブレーク入力

0010 : (BRK2) タイマ 1 ブレーク入力 2

0011 : (BRK\_ACTH) タイマ 8 ブレーク入力

0100 : (BRK2) タイマ 8 ブレーク入力 2 (専用)

0101 : タイマ 1 ブレーク入力 2 + タイマ 8 ブレーク入力 2 (専用)

0110 : タイマ 1 OCREF\_CLR 入力

0111 : タイマ 1 入力キャプチャ 1

1000 : タイマ 2 入力キャプチャ 4

1001 : タイマ 2 OCREF\_CLR 入力

1010 : タイマ 3 入力キャプチャ 1

1011 : タイマ 3 OCrefclear 入力

1100 : タイマ 20 ブレーク入力を選択 (STM32F303xDxE のみ)

1101 : タイマ 20 ブレーク 2 入力を選択 (STM32F303xDxE のみ)

1110 : タイマ 1 ブレーク 2 またはタイマ 8 ブレーク 2 またはタイマ 20 ブレーク 2 (STM32F303xDxE のみ)

1111 : タイマ 20 OCrefClear 入力を選択 (STM32F303xDxE のみ)

ビット 9 **COMP2WINMODE** : コンパレータ 2 ウィンドウモード (STM32F303xB/C および STM32F358xC デバイスのみ)

このビットは、ウィンドウモードを選択します。コンパレータの 2 つの非反転入力は、コンパレータ 1 (PA1) の非反転入力を共有しています。

0 : コンパレータ 1 および 2 をウィンドウモードで使用できません。

1 : コンパレータ 1 および 2 をウィンドウモードで使用できます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **COMP2INPSEL** : コンパレータ 2 非反転入力選択 (STM32F303xB/C および STM32F358xC デバイスのみ)

- 0 : PA7 が選択されています。
- 1 : PA3 が選択されています。

**注 :** STM32F303x6/8, STM32F303xDxE, および STM32F328 では、このビットは予約済みです。ビット 7 に書き込まれた値にかかわらず、PA3 では COMP2\_VINP を使用できます。

ビット 6:4 **COMP2INMSEL[2:0]** : コンパレータ 2 反転入力選択

これらのビットは、ビット 22 と組み合わせて、コンパレータ 2 の反転入力に接続されたソースを選択できます。

- 0000 : Vrefint の 1/4
- 0001 : Vrefint の 1/2
- 0010 : Vrefint の 3/4
- 0011 : Vrefint
- 0100 : PA4 または DAC1\_CH1 出力 (有効な場合)  
STM32F303xB/C および STM32F358xC :
- 0101 : PA5 または DAC1\_CH2 出力 (有効な場合)  
STM32F303x6/8 および STM32F328x8 :
- 0101 : DAC1\_CH2 出力
- 0110 : PA2
- 1000 DAC2\_CH1 出力
- 残りの組み合わせ : 予約済み

ビット 3:2 **COMP2MODE[1:0]** : コンパレータ 2 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 2 の動作モードを制御し、スピード/消費電力を調節できます。

- 00 : ハイスピード
- 01 : ミディアムスピード
- 10 : 低電力
- 11 : 超低電力

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **COMP2EN** : コンパレータ 2 イネーブル

このビットは COMP2 のオン/オフを切り替えます。

- 0 : コンパレータ 2 無効
- 1 : コンパレータ 2 有効

## 17.5.3 COMP3 制御およびステータスレジスタ (COMP3\_CSR)

**注 :** このレジスタは、STM32F302xB/C/D/E および STM32F302x6/8 でのみ使用可能です。

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP3LOCK	COMP3OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP3_BLANKING			COMP3HYST [1:0] <sup>(1)</sup>	
rwo	r										rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP3POL	Res.	COMP3OUTSEL				Res.	Res.	COMP3INPSEL (1)	COMP3INMSEL[2:0]			COMP3MODE [1:0] <sup>(1)</sup>		Res.	COMP3EN
rw		rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		rw

1. STM32F303xB/C および STM32F358xC のみ。



**ビット 31 COMP3LOCK** : コンパレータ 3 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP3\_CSR レジスタを読み出し専用にすることができます。

- 0 : COMP3\_CSR は読み書き用です。
- 1 : COMP3\_CSR は読み出し専用です。

**ビット 30 COMP3OUT** : コンパレータ 3 出力

この読み出し専用ビットはコンパレータ 3 の出力状態のコピーです。

- 0 : 出力はロー (反転入力を下回る非反転入力)
- 1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

**ビット 20:18 COMP3\_BLANKING** : コンパレータ 3 ブランキングソース

これらのビットによってコンパレータ 3 出力のブランキングを制御するタイマ出力を選択します。

- 000 : ブランキングなし
- 001 : TIM1 OC5 がブランキングソースとして選択されます。
- 010 : 予約済み。
- 011 : TIM2 OC4 がブランキングソースとして選択されます。
- 他の設定 : 予約済み

**ビット 17:16 COMP3HYST[1:0]** コンパレータ 3 ヒステリシス

これらのビットは、ヒステリシスレベルを制御します (STM32F303xB/C および STM32F358x のみ)。

- 00 : ヒステリシスなし
- 01 : 低ヒステリシス
- 10 : 中ヒステリシス
- 11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303xD/E、STM32F303x6/8、および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。

**ビット 15 COMP3POL** : コンパレータ 3 出力の極性

このビットは、コンパレータ 3 出力を反転させるために使用します。

- 0 : 出力は反転されません。
- 1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。



ビット 13:10 **COMP3OUTSEL[3:0]** : コンパレータ 3 出力選択

これらのビットは、COMP3\_LOCK がセットされていない場合、ソフトウェアによってセット/クリアされます。これらのビットによってコンパレータ 3 出力に接続するタイマ入力を選択します。

- 0000 : タイマ入力なし
- 0001 : (BRK\_ACTH) タイマ 1 ブレーク入力
- 0010 : (BRK2) タイマ 1 ブレーク入力 2
- 0011 : (BRK\_ACTH) タイマ 8 ブレーク入力
- 0100 : (BRK2) タイマ 8 ブレーク入力 2
- 0101 : タイマ 1 ブレーク入力 2 または タイマ 8 ブレーク入力 2
- 0110 : タイマ 1 OCrefclear 入力
- 0111 : タイマ 4 入力キャプチャ 1
- 1000 : タイマ 3 入力キャプチャ 2
- 1001 : タイマ 2 OCrefclear 入力
- 1010 : タイマ 15 入力キャプチャ 1
- 1011 : タイマ 15 ブレーク入力
- 1100 = タイマ 20 ブレーク入力を選択 (STM32F303xD/E のみ)
- 1101 = タイマ 20 ブレーク 2 入力を選択 (STM32F303xD/E のみ)
- 1110 = タイマ 1 ブレーク 2 または タイマ 8 ブレーク 2 または タイマ 20 ブレーク 2 (STM32F303xD/E のみ)

残りの組み合わせ : 予約済み

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **COMP3INPSEL** : コンパレータ 3 非反転入力選択

- 0 : PB14
- 1 : PD14

**注 :** STM32F303xD/E では、このビットは予約済みです。COMP3\_VINP は PB14 で使用できます。

ビット 6:4 **COMP3INMSEL[2:0]** : コンパレータ 1 反転入力選択

これらのビットにより、コンパレータ 3 の反転入力に接続されたソースを選択できます。

- 000 : Vrefint の 1/4
- 001 : Vrefint の 1/2
- 010 : Vrefint の 3/4
- 011 : Vrefint
- 100 : PA4 または DAC1 出力 (有効な場合)
- 101 : PA5 または DAC2 出力 (有効な場合)
- 110 : PD15
- 111 : PB12

ビット 3:2 **COMP3MODE[1:0]** : コンパレータ 3 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 3 の動作モードを制御し、スピード/消費を調節できます。

- 00 : 超低電力
- 01 : 低電力
- 10 : ミディアムスピード
- 11 : ハイスピード

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **COMP3EN** : コンパレータ 3 イネーブル

このビットは COMP3 のオン/オフを切り替えます。

- 0 : コンパレータ 3 無効
- 1 : コンパレータ 3 有効



## 17.5.4 COMP4 制御およびステータスレジスタ (COMP4\_CSR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP4LOCK	COMP4OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP4INMSEL[3] <sup>(1)</sup>	Res.	COMP4_BLANKING[2:0]			COMP4HYST[1:0] <sup>(2)</sup>	
rw	r								rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP4POL	Res.	COMP4OUTSEL[3:0]				COMP4WINMODE <sup>(3)</sup>	Res.	COMP4INPSEL <sup>(2)</sup>	COMP4INMSEL[2:0]			COMP4MODE[1:0] <sup>(2)</sup>		Res.	COMP4EN
rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw		rw

1. STM32F303x6/8 および STM32F328x8 のみ。
2. STM32F303xB/C および STM32F358xC のみ。
3. STM32F303x6/8/D/E、STM32F328x8、および STM32F398xE デバイスでは使用不可。

ビット 31 **COMP4LOCK** : コンパレータ 4 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP4\_CSR レジスタを読み出し専用にすることができます。

- 0 : COMP4\_CSR は読み書き用です。
- 1 : COMP4\_CSR は読み出し専用です。

ビット 30 **COMP4OUT** : コンパレータ 4 出力

この読み出し専用ビットはコンパレータ 4 の出力状態のコピーです。

- 0 : 出力はロー (反転入力を下回る非反転入力)
- 1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **COMP4INMSEL[3]** : コンパレータ 4 反転入力選択。このビットは、F303x6/x8 および F328xx でのみ使用できます。これは、COMP の反転入力を設定するために、ビット [6..4] とともに使用されます。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20:18 **COMP4\_BLANKING** : コンパレータ 4 ブランキングソース

これらのビットによってコンパレータ 4 出力のブランキングを制御するタイマ出力を選択します。

- 000 : ブランキングなし
  - 001 : TIM3 OC4 がブランキングソースとして選択されます。
  - 010 : TIM8 OC5 がブランキングソースとして選択されます (STM32F303xB/C および STM32F358xC デバイスのみ)。
  - 011 : TIM15 OC1 がブランキングソースとして選択されます。
- その他の設定 : 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **COMP4HYST[1:0]** : コンパレータ 4 ヒステリシス

STM32F303xB/C および STM32F358xC では、これらのビットがヒステリシスレベルを制御します。

- 00 : ヒステリシスなし
- 01 : 低ヒステリシス
- 10 : 中ヒステリシス
- 11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303xD/E、STM32F303x6/8、および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。

ビット 15 **COMP4POL** : コンパレータ 4 出力の極性

このビットは、コンパレータ 4 出力を反転させるために使用します。

- 0 : 出力は反転されません。
- 1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **COMP4OUTSEL[3:0]** : コンパレータ 4 出力選択

これらのビットによってコンパレータ 4 出力に接続するタイマ入力を選択します。

- 0000 : タイマ入力なしの選択
  - 0001 : (BRK) タイマ 1 ブレーク入力
  - 0010 : (BRK2) タイマ 1 ブレーク入力 2
  - 0011 : (BRK) タイマ 8 ブレーク入力 (STM32F303xB/C および STM32F358xC デバイスのみ)
  - 0100 : (BRK2) タイマ 8 ブレーク入力 2 (STM32F303xB/C および STM32F358xC デバイスのみ)
  - 0101 : タイマ 1 ブレーク入力 2 または タイマ 8 ブレーク入力 2 (STM32F303xB/C および STM32F358xC デバイスのみ)
  - 0110 : タイマ 3 入力キャプチャ 3
  - 0111 : タイマ 8 OCrefclear 入力 (STM32F303xB/C および STM32F358xC デバイスのみ)
  - 1000 : タイマ 15 入力キャプチャ 2
  - 1001 : タイマ 4 入力キャプチャ 2 (STM32F303xB/C および STM32F358xC のみ)
  - 1010 : タイマ 15 OCREF\_CLR 入力
  - 1011 : タイマ 3 OCrefclear 入力
  - 1100 = タイマ 20 ブレーク入力を選択 (STM32F303xD/E のみ)
  - 1101 = タイマ 20 ブレーク 2 入力を選択 (STM32F303xD/E のみ)
  - 1110 = タイマ 1 ブレーク 2 または タイマ 8 ブレーク 2 または タイマ 20 ブレーク 2 (STM32F303xD/E のみ)
- 残りの組み合わせ : 予約済み

ビット 9 **COMP4WINMODE** : コンパレータ 4 ウィンドウモード (STM32F303xB/C および STM32F358xC デバイスのみ)

このビットは、ウィンドウモードを選択します。2 つの非反転入力コンパレータ 3 および 4 は、コンパレータ 3 (PB14 または PD14) の非反転入力を共有しています。

- 0 : コンパレータ 3 および 4 をウィンドウモードで使用できません。
- 1 : コンパレータ 3 および 4 をウィンドウモードで使用できます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **COMP4INPSEL** : コンパレータ 4 非反転入力選択 (STM32F303xB/C および STM32F358xC のみ)

- 0 : PB0
- 1 : PE7

**注 :** STM32F303x6/8、STM32F303xDxE、および STM32F328 では、このビットは予約済みです。COMP4\_VINP は PB0 で使用できます。



ビット 6:4 **COMP4INMSEL[3:0]** : コンパレータ 4 反転入力選択

これらのビットにより、コンパレータ 4 の反転入力に接続されたソースを選択できます。

- 0000 : Vrefint の 1/4
- 0001 : Vrefint の 1/2
- 0010 : Vrefint の 3/4
- 0011 : Vrefint
- 0100 : PA4 または DAC1\_CH1 出力 (有効な場合)
- :
- 0101 : PA5 または DAC1\_CH2 出力 (有効な場合)
- STM32F303x6/8 および STM32F328x8 :
- 0101 : DAC1\_CH2 出力
- 0110 : PE8
- 0111 : PB2
- 1000 : DAC2\_CH1 出力
- 残りの組み合わせ : 予約済み

ビット 3:2 **COMP4MODE[1:0]** : コンパレータ 1 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 4 の動作モードを制御し、スピード/消費を調節できます。

- 00 : 超低電力
- 01 : 低電力
- 10 : ミディアムスピード
- 11 : ハイスピード

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **COMP4EN** : コンパレータ 4 イネーブル

このビットは COMP4 のオン/オフを切り替えます。

- 0 : コンパレータ 4 無効
- 1 : コンパレータ 4 有効

## 17.5.5 COMP5 制御およびステータスレジスタ (COMP5\_CSR)

**注 :** このレジスタは、STM32F302xB/C/D/E および STM32F302x6/8 でのみ使用可能です。

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP5LOCK	COMP5OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP5_BLANKING			COMP5HYST [1:0] <sup>(1)</sup>	
rw	r										rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP5POL	Res.	COMP5OUTSEL				Res.	Res.	COMP5INPSEL (1)	COMP5INMSEL[2:0]			COMP5MODE [1:0] <sup>(1)</sup>		Res.	COMP5EN
rw		rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		rw

1. STM32F303xB/C のみ。

**ビット 31 COMP5LOCK** : コンパレータ 5 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP5\_CSR レジスタを読み出し専用にすることができます。

0 : COMP5\_CSR は読み書き用です。

1 : COMP5\_CSR は読み出し専用です。

**ビット 30 COMP5OUT** : コンパレータ 5 出力

この読み出し専用ビットはコンパレータ 5 の出力状態のコピーです。

0 : 出力はロー (反転入力を下回る非反転入力)

1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

**ビット 20:18 COMP5\_BLANKING** : コンパレータ 5 ブランキングソース

これらのビットによってコンパレータ 5 出力のブランキングを制御するタイマ出力を選択します。

000 : ブランキングなし

001 : 予約済み。

010 : TIM8 OC5 がブランキングソースとして選択されます。

011 : TIM3 OC3 がブランキングソースとして選択されます。

他の設定 : 予約済み

**ビット 17:16 COMP5HYST[1:0]** コンパレータ 5 ヒステリシス

これらのビットは、ヒステリシスレベルを制御します (STM32F303xB/C および STM32F358xC デバイス のみ)。

00 : ヒステリシスなし

01 : 低ヒステリシス

10 : 中ヒステリシス

11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303xD/E、STM32F303x6/8、および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。

**ビット 15 COMP5POL** : コンパレータ 5 出力の極性

このビットは、コンパレータ 5 出力を反転させるために使用します。

0 : 出力は反転されません。

1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **COMP5OUTSEL[3:0]** : コンパレータ 5 出力選択

これらのビットによってコンパレータ 5 出力に接続するタイマ入力を選択します。

- 0000 : タイマ入力なしの選択
  - 0001 : (BRK\_ACTH) タイマ 1 ブレーク入力
  - 0010 : (BRK2) タイマ 1 ブレーク入力 2
  - 0011 : (BRK\_ACTH) タイマ 8 ブレーク入力
  - 0100 : (BRK2) タイマ 8 ブレーク入力 2
  - 0101 : タイマ 1 ブレーク入力 2 または タイマ 8 ブレーク入力 2
  - 0110 : タイマ 2 入力キャプチャ 1
  - 0111 : タイマ 8 OCrefclear 入力
  - 1000 : タイマ 17 入力キャプチャ 1
  - 1001 : タイマ 4 入力キャプチャ 3
  - 1010 : タイマ 16 ブレーク入力
  - 1011 : タイマ 3 OCrefclear 入力
  - 1100 = タイマ 20 ブレーク入力を選択 (STM32F303xDxE のみ)
  - 1101 = タイマ 20 ブレーク 2 入力を選択 (STM32F303xDxE のみ)
  - 1110 = タイマ 1 ブレーク 2 または タイマ 8 ブレーク 2 または タイマ 20 ブレーク 2 (STM32F303xDxE のみ)
- 残りの組み合わせ : 予約済み

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **COMP5INPSEL** : コンパレータ 5 非反転入力選択

- 0 : PD12
- 1 : PB13

**注 :** STM32F303xDxE では、このビットは予約済みです。COMP5\_VINP は PB13 で使用できます。

ビット 6:4 **COMP5INSEL[2:0]** : コンパレータ 5 反転入力選択

これらのビットにより、コンパレータ 5 の反転入力に接続されたソースを選択できます。

- 000 : Vrefint の 1/4
- 001 : Vrefint の 1/2
- 010 : Vrefint の 3/4
- 011 : Vrefint
- 100 : PA4 または DAC1 出力 (有効な場合)
- 101 : PA5 または DAC2 出力 (有効な場合)
- 110 : PD13
- 111 : PB10

**注 :** STM32F303xDxE では、このビットは予約済みです。COMP5\_VINP は PB13 で使用できます。

ビット 3:2 **COMP5MODE[1:0]** : コンパレータ 5 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 5 の動作モードを制御し、スピード/消費を調節できます。

- 00 : 超低電力
- 01 : 低電力
- 10 : ミディアムスピード
- 11 : ハイスピード

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **COMP5EN** : コンパレータ 5 イネーブル

このビットは COMP5 のオン/オフを切り替えます。

- 0 : コンパレータ 5 無効
- 1 : コンパレータ 5 有効

## 17.5.6 COMP6 制御およびステータスレジスタ (COMP6\_CSR)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP6LOCK	COMP6OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP6INMSEL[3] <sup>(1)</sup>	Res.	COMP6_BLANKING[2:0]			COMP6HYST[1:0] <sup>(2)</sup>	
rw	r								rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP6POL	Res.	COMP6OUTSEL[3:0]				COMP6WINMODE <sup>(3)</sup>	Res.	COMP6INPSEL <sup>(2)</sup>	COMP6INMSEL[2:0]			COMP6MODE[1:0] <sup>(2)</sup>		Res.	COMP6EN
rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw		rw

1. STM32F303x6/8 および STM32F328 のみ。
2. STM32F303xB/C および STM32F358xC デバイスのみ。
3. STM32F303x6/8/D/E、STM32F328x8、および STM32F398xE デバイスでは使用不可。

ビット 31 **COMP6LOCK** : コンパレータ 6 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP6\_CSR レジスタを読み出し専用にすることができます。

- 0 : COMP6\_CSR は読み書き用です。
- 1 : COMP6\_CSR は読み出し専用です。

ビット 30 **COMP6OUT** : コンパレータ 6 出力

この読み出し専用ビットはコンパレータ 6 の出力状態のコピーです。

- 0 : 出力はロー (反転入力を下回る非反転入力)
- 1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **COMP6INMSEL[3]**: コンパレータ 6 反転入力選択。このビットは、F303x6/x8 および F328xx でのみ使用できます。これは、COMP の反転入力を設定するために、ビット [6..4] とともに使用されます。

ビット 21 予約済みであり、リセット値に保持する必要があります。

ビット 20:18 **COMP6\_BLANKING** : コンパレータ 6 ブランキングソース

これらのビットによってコンパレータ 6 出力のブランキングを制御するタイマ出力を選択します。

- 000 : ブランキングなし
- 001 : 予約済み
- 010 : TIM8 OC5 がブランキングソースとして選択されます (STM32F303xB/C および STM32F358xC デバイスのみ)。
- 011 : TIM2 OC4 がブランキングソースとして選択されます。
- 100 : TIM15 OC2 がブランキングソースとして選択されます。
- 他の設定 : 予約済み

ブランキング信号はアクティブハイです (コンパレータの出力信号をマスクします)。コンパレータとブランキング信号極性を正しくプログラムするかどうかはユーザに委ねられています。

ビット 17:16 **COMP6HYST[1:0]** : コンパレータ 6 ヒステリシス

STM32F303xB/C および STM32F358xC では、これらのビットがヒステリシスレベルを制御します。

00 : ヒステリシスなし

01 : 低ヒステリシス

10 : 中ヒステリシス

11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303x6/8/D/E および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。

ビット 15 **COMP6POL** : コンパレータ 6 出力の極性

このビットは、コンパレータ 6 出力を反転させるために使用します。

0 : 出力は反転されません。

1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **COMP6OUTSEL[3:0]** : コンパレータ 6 出力選択

これらのビットによってコンパレータ 6 出力に接続するタイマ入力を選択します。

0000 : タイマ入力なし

0001 : (BRK\_ACTH) タイマ 1 ブレーク入力

0010 : (BRK2) タイマ 1 ブレーク入力 2

0011 : (BRK\_ACTH) タイマ 8 ブレーク入力 (STM32F303xB/C および STM32F358xC デバイスのみ)

0100 : (BRK2) タイマ 8 ブレーク入力 2 (STM32F303xB/C および STM32F358xC デバイスのみ)

0101 : タイマ 1 ブレーク入力 2 またはタイマ 8 ブレーク入力 2 (STM32F303xB/C および STM32F358xC デバイスのみ)

0110 : タイマ 2 入力キャプチャ 2

0111 : タイマ 8 OCreffieldclear 入力 (STM32F303xB/C および STM32F358xC デバイスのみ)

1000 : タイマ 2 OCREFIELD\_CLR 入力

1001 : タイマ 16 OCREFIELD\_CLR 入力

1010 : タイマ 16 入力キャプチャ 1

1011 : タイマ 4 入力キャプチャ 4 (STM32F303xB/C および STM32F358xC のみ)

1100 = タイマ 20 ブレーク入力を選択 (STM32F303xDxE のみ)

1101 = タイマ 20 ブレーク 2 入力を選択 (STM32F303xDxE のみ)

1110 = タイマ 1 ブレーク 2 またはタイマ 8 ブレーク 2 またはタイマ 20 ブレーク 2 (STM32F303xDxE のみ)

残りの組み合わせ : 予約済み

ビット 9 **COMP6WINMODE** : コンパレータ 6 ウィンドウモード (STM32F303xB/C/D/E および STM32F358xC デバイスのみ)

このビットは、ウィンドウモードを選択します。2 つの非反転入力コンパレータ 6 は、コンパレータ 5 (PD12 または PB13) の非反転入力を共有しています。

0 : コンパレータ 5 および 6 をウィンドウモードで使用できません。

1 : コンパレータ 5 および 6 をウィンドウモードで使用できます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **COMP6INPSEL** : コンパレータ 6 非反転入力選択 (STM32F303x6/8 および STM32F358x のみ)

0 : PD11

1 : PB11

**注 :** STM32F303x6/x8/D/E および STM32F328x8 では、このビットは予約済みです。ビット 7 に書き込まれた値にかかわらず、PB11 では COMP6\_VINP を使用できます。



ビット 6:4 **COMP6INMSEL[2:0]** : コンパレータ 6 反転入力選択

これらのビットにより、コンパレータ 6 の反転入力に接続されたソースを選択できます。

- 0000 : Vrefint の 1/4
- 0001 : Vrefint の 1/2
- 0010 : Vrefint の 3/4
- 0011 : Vrefint
- 0100 : PA4 または DAC1\_CH1 出力 (有効な場合)
- STM32F303xB/C および STM32F358xC :
- 0101 : PA5 または DAC1\_CH2 出力 (有効な場合)
- STM32F303x6/8 および STM32F328x8 :
- 0101 : DAC1\_CH2 出力
- 0110 : PD10
- 0111 : PB15
- 1000 : DAC2\_CH1
- 残りの組み合わせ : 予約済み

ビット 3:2 **COMP6MODE[1:0]** : コンパレータ 6 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 6 の動作モードを制御し、スピード/消費を調節できます。

- 00 : 超低電力
- 01 : 低電力
- 10 : ミディアムスピード
- 11 : ハイスピード

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **COMP6EN** : コンパレータ 6 イネーブル

このビットは COMP6 のオン/オフを切り替えます。

- 0 : コンパレータ 6 無効
- 1 : コンパレータ 6 有効

## 17.5.7 COMP7 制御およびステータスレジスタ (COMP7\_CSR)

**注 :** このレジスタは、STM32F302xB/C/D/E および STM32F302x6/8 でのみ使用可能です。

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP7LOCK	COMP7OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP7_BLANKING			COMP7HYST [1:0] <sup>(1)</sup>	
rw	r										rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP7POL	Res.	COMP7OUTSEL				Res.	Res.	COMP7INPSEL (1)	COMP7INMSEL[2:0]			COMP7MODE [1:0] <sup>(1)</sup>		Res.	COMP7EN
rw		rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		rw

1. STM32F303xB/C および STM32F358xC デバイスのみ

**ビット 31 COMP7LOCK** : コンパレータ 7 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

COMP7\_CSR レジスタを読み出し専用にすることができます。

0 : COMP7\_CSR は読み書き用です。

1 : COMP7\_CSR は読み出し専用です。

**ビット 30 COMP7OUT** : コンパレータ 7 出力

この読み出し専用ビットはコンパレータ 7 の出力状態のコピーです。

0 : 出力はロー (反転入力を下回る非反転入力)

1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:21 予約済みであり、リセット値に保持する必要があります。

**ビット 20:18 COMP7\_BLANKING** : コンパレータ 7 ブランキングソース

これらのビットによってコンパレータ 7 出力のブランキングを制御するタイマ出力を選択します。

000 : ブランキングなし

001 : TIM1 OC5 がブランキングソースとして選択されます。

010 : TIM8 OC5 がブランキングソースとして選択されます。

011 : 予約済み

100 : TIM15 OC2 がブランキングソースとして選択されます。

他の設定 : 予約済み

**ビット 17:16 COMP7HYST[1:0]** コンパレータ 7 ヒステリシス

これらのビットは、ヒステリシスレベルを制御します。(STM32F303xB/C および STM32F358xC デバイスのみ。)

00 : ヒステリシスなし

01 : 低ヒステリシス

10 : 中ヒステリシス

11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

STM32F303xD/E、STM32F303x6/8、および STM32F328x8 では、これらのビットは予約済みであり、リセット値に保持する必要があります。

**ビット 15 COMP7POL** : コンパレータ 7 出力の極性

このビットは、コンパレータ 7 出力を反転させるために使用します。

0 : 出力は反転されません。

1 : 出力は反転されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **COMP7OUTSEL[3:0]** : コンパレータ 7 出力選択

これらのビットによってコンパレータ 7 出力に接続するタイマ入力を選択します。

- 0001 : (BRK) タイマ 1 ブレーク入力
  - 0010 : (BRK2) タイマ 1 ブレーク入力 2
  - 0011 : (BRK) タイマ 8 ブレーク入力
  - 0100 : (BRK2) タイマ 8 ブレーク入力 2
  - 0101 : タイマ 1 ブレーク入力 2 + タイマ 8 ブレーク入力 2
  - 0110 : タイマ 1 OCrefclear 入力
  - 0111 : タイマ 8 OCrefclear 入力
  - 1000 : タイマ 2 入力キャプチャ 3
  - 1001 : タイマ 1 入力キャプチャ 2
  - 1010 : タイマ 17 OCrefclear 入力
  - 1011 : タイマ 17 ブレーク入力
  - 1100 = タイマ 20 ブレーク入力を選択 (STM32F303xDxE のみ)
  - 1101 = タイマ 20 ブレーク 2 入力を選択 (STM32F303xDxE のみ)
  - 1110 = タイマ 1 ブレーク 2 またはタイマ 8 ブレーク 2 またはタイマ 20 ブレーク 2 (STM32F303xDxE のみ)
- 残りの組み合わせ : 予約済み

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **COMP7INPSEL** : コンパレータ 7 非反転入力選択

- 0 : PA0
- 1 : PC1

**注 :** STM32F303x6/x8/D/E および STM32F328x では、このビットは予約済みです。COMP7\_VINP は PC1 で使用できません。

ビット 6:4 **COMP7INSEL[2:0]** : コンパレータ 7 反転入力選択

これらのビットにより、コンパレータ 7 の反転入力に接続されたソースを選択できます。

- 000 : Vrefint の 1/4
- 001 : Vrefint の 1/2
- 010 : Vrefint の 3/4
- 011 : Vrefint
- 100 : PA4 または DAC1 出力 (有効な場合)
- 101 : PA5 または DAC2 出力 (有効な場合)
- 110 : PC0
- 111 : 予約済み

ビット 3:2 **COMP7MODE[1:0]** : コンパレータ 7 モード (STM32F303xB/C および STM32F358xC デバイスのみ)

これらのビットはコンパレータ 7 の動作モードを制御し、スピード/消費を調節できます。

- 00 : 超低電力
- 01 : 低電力
- 10 : ミディアムスピード
- 11 : ハイスピード

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **COMP7EN** : コンパレータ 7 イネーブル

このビットは COMP7 のオン/オフを切り替えます。

- 0 : コンパレータ 7 無効
- 1 : コンパレータ 7 有効



## 17.5.8 COMP レジスタマップ

次の表にコンパレータレジスタの一覧を示します。

表 108. COMP レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C	COMP1_CSR	COMP1LOCK	COMP1OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPx_BLANKING[2:0]	Res.	COMP1HYST[1:0]	COMP1POL	Res.	Res.	COMP1OUT SEL [3:0]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値	0	0											0	0	0	0	0			0	0	0	0				0	0	0	0	0	0
0x20	COMP2_CSR	COMP2LOCK	COMP2OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP2INMSEL[3]	Res.	Res.	COMP2_BLANKING	Res.	Res.	COMP2POL	Res.	Res.	COMP2OUT SEL[3:0]	COMP2WINMODE	Res.	Res.	Res.	Res.	COMP2INSEL	COMP2INMSEL[2:0]	Res.	Res.	COMP2MODE[1:0]	COMP2_INP_DAC	COMP2EN	
	リセット値	0	0								0			0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	COMP3_CSR	COMP3LOCK	COMP3OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP3_BLANKING	Res.	COMP3HYST[1:0]	COMP3POL	Res.	Res.	COMP3OUT SEL[3:0]	Res.	Res.	Res.	Res.	Res.	COMP3INSEL	COMP3INMSEL[2:0]	Res.	Res.	COMP3MODE[1:0]	COMP3EN		
	リセット値	0	0											0	0	0	0	0			0	0	0	0		0	0	0	0	0	0	0	0
0x28	COMP4_CSR	COMP4LOCK	COMP4OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP4INMSEL[3]	Res.	Res.	COMP4_BLANKING	Res.	Res.	COMP4POL	Res.	Res.	COMP4OUT SEL[3:0]	COMP4WINMODE	Res.	Res.	Res.	Res.	COMP4INSEL	COMP4INMSEL[2:0]	Res.	Res.	COMP4MODE[1:0]	COMP4EN		
	リセット値	0	0								0			0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	COMP5_CSR	COMP5LOCK	COMP5OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP5_BLANKING	Res.	COMP5HYST[1:0]	COMP5POL	Res.	Res.	COMP5OUT SEL[3:0]	Res.	Res.	Res.	Res.	Res.	COMP5INSEL	COMP5INMSEL[2:0]	Res.	Res.	COMP5MODE[1:0]	COMP5EN		
	リセット値	0	0											0	0	0	0	0			0	0	0	0		0	0	0	0	0	0	0	0
0x30	COMP6_CSR	COMP6LOCK	COMP6OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP6INMSEL[3]	Res.	Res.	COMP6_BLANKING	Res.	Res.	COMP6POL	Res.	Res.	COMP6OUT SEL[3:0]	COMP6WINMODE	Res.	Res.	Res.	Res.	COMP6INSEL	COMP6INMSEL[2:0]	Res.	Res.	COMP6MODE[1:0]	COMP6EN		
	リセット値	0	0								0			0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	COMP7_CSR	COMP7LOCK	COMP7OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP7_BLANKING	Res.	COMP7HYST[1:0]	COMP7POL	Res.	Res.	COMP7OUT SEL[3:0]	Res.	Res.	Res.	Res.	Res.	COMP7INSEL	COMP7INMSEL[2:0]	Res.	Res.	COMP7MODE[1:0]	COMP7EN		
	リセット値	0	0											0	0	0	0	0			0	0	0	0		0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



## 18 オペアンプ (OPAMP)

### 18.1 OPAMP の概要

STM32F303xB/C/D/E および STM32F358xC デバイスは 4 つのオペアンプ OPAMP1、OPAMP2、OPAMP3、および OPAMP4 を内蔵し、STM32F303x6/8 および STM32F328x8 デバイスは 1 つのオペアンプ OPAMP2 を内蔵しています。これらは、スタンドアロンアンプまたはフォロア/プログラム可能なゲインアンプのどちらかとして使用できます。

オペアンプ出力は測定のため ADC チャンネルに内部接続されています。

### 18.2 OPAMP の主な機能

- レールツーレール入出力
- 低オフセット電圧
- スタンドアロンオペアンプまたはプログラム可能なゲインアンプ (PGA) として設定できる機能
- すべての端子へのアクセス
- 反転入力および非反転入力での入力マルチプレクサ
- タイマでトリガでき、PWM 信号と同期可能な入力マルチプレクサ

### 18.3 OPAMP の機能説明

#### 18.3.1 概要

すべての OPAMP の非反転入力には 1 つの 4:1 マルチプレクサがあり、また反転入力には 1 つの 2:1 マルチプレクサがあります。

反転および非反転入力は、OPAMPx\_CSR レジスタの VM\_SEL および VP\_SEL ビットを使用して選択されます。

OPAMP 入出力として使用される I/O は、GPIO レジスタのアナログモードで設定する必要があります。

専用 I/O との接続については、以下の表の一覧と [図 127](#)、[図 128](#)、および [図 129](#)を参照してください。

表 109. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE の専用 I/O との接続

OPAMP1 反転入力	OPAMP1 非反転入力	OPAMP2 反転入力	OPAMP2 非反転入力	OPAMP3 反転入力	OPAMP3 非反転入力	OPAMP4 反転入力	OPAMP4 非反転入力
PA3 (VM1)	PA1 (VP0)	PA5 (VM1)	PA7 (VP0)	PB2 (VM1)	PB0 (VP0)	PB10 (VM0)	PB13 (VP0)
PC5 (VM0)	PA7 (VP1)	PC5 (VM0)	PD14 (VP1)	PB10 (VM0)	PB13 (VP1)	PD8 (VM1)	PD11 (VP1)
-	PA3 (VP2)	-	PB0 (VP2)	-	PA1 (VP2)	-	PA4 (VP2)
-	PA5 (VP3)	-	PB14 (VP3)	-	PA5 (VP3)	-	PB11 (VP3)

表 110. STM32F303x6/8 および STM32F328x8 の専用 I/O との接続

OPAMP2 反転入力	OPAMP2 非反転入力
PA5 (VM1)	PA7 (VP0)
PC5 (VM0)	PD14 (VP1)
-	PB0 (VP2)

## 18.3.2 クロック

クロックコントローラによって提供される OPAMP クロックは、PCLK2 (APB2 クロック) と同期しています。RCC コントローラには、クロックイネーブル制御ビットは提供されていません。OPAMP にクロックソースを使用するには、RCC コントローラで SYSCFG クロックイネーブル制御ビットをセットする必要があります。

## 18.3.3 オペアンプとコンパレータの相互接続

オペアンプとコンパレータ間の相互接続はモータ制御アプリケーションで有用です。この接続の一覧を以下の図に示します。

図 127. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE コンパレータとオペアンプの相互接続 (パート 1)

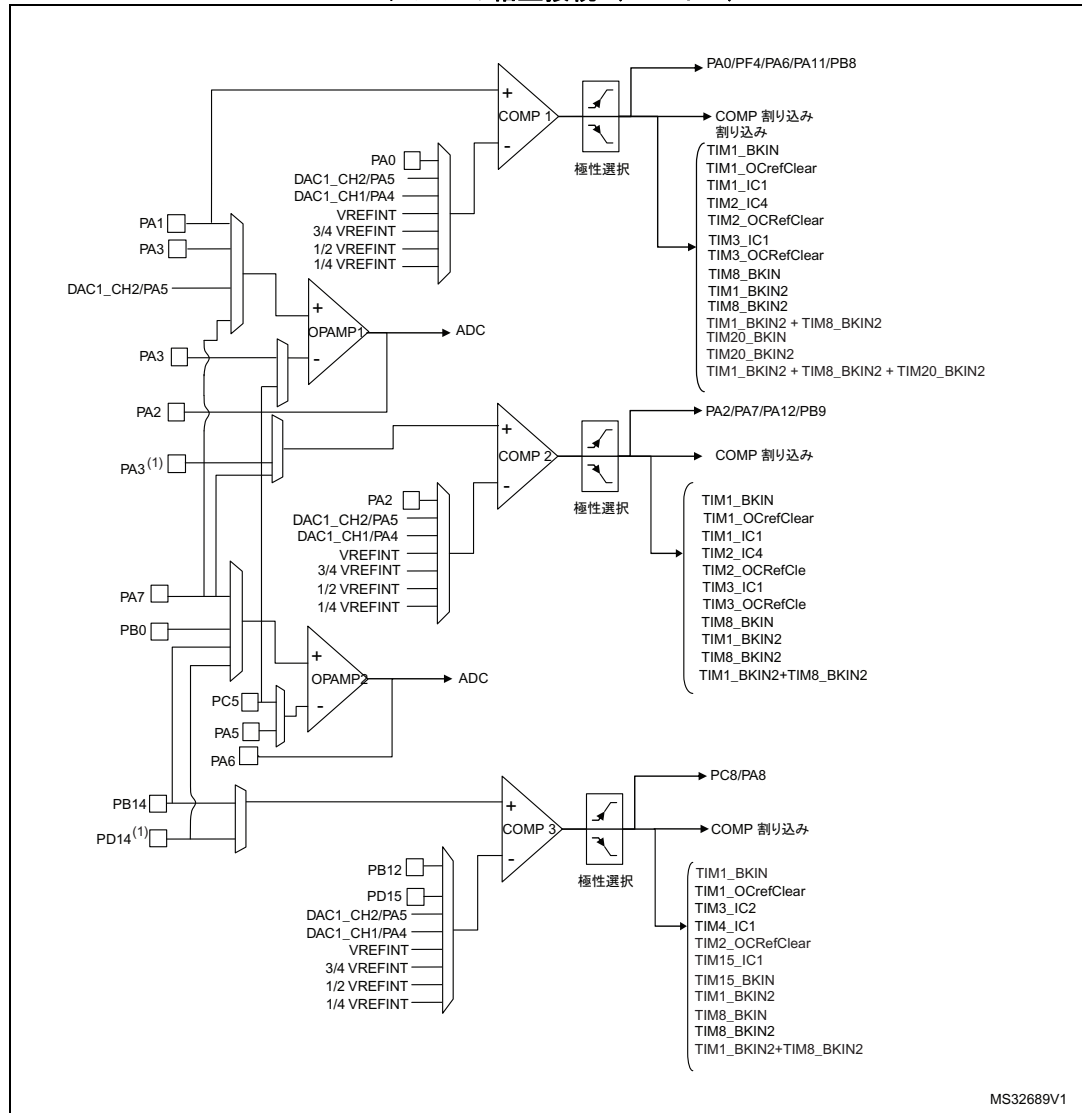
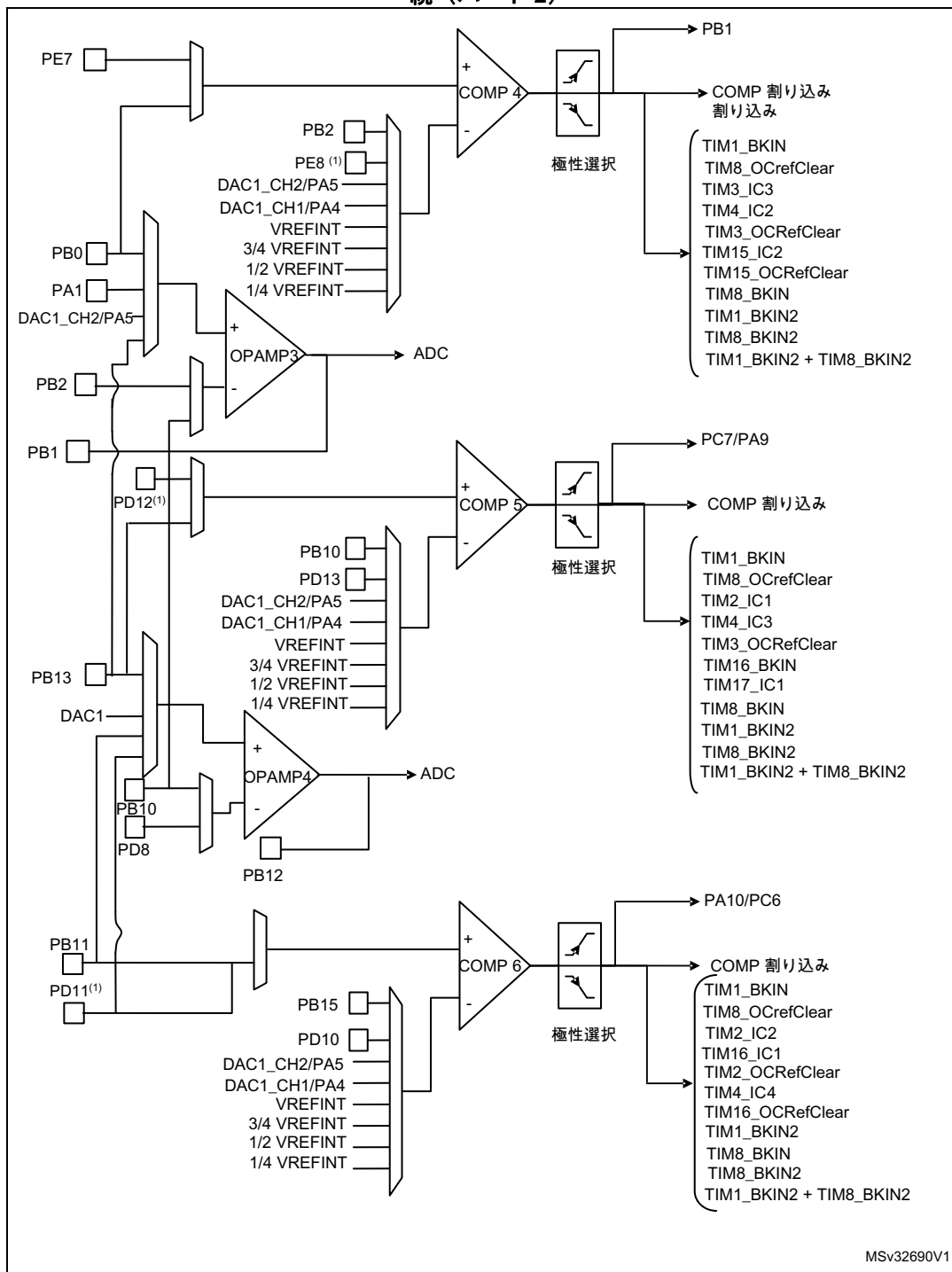


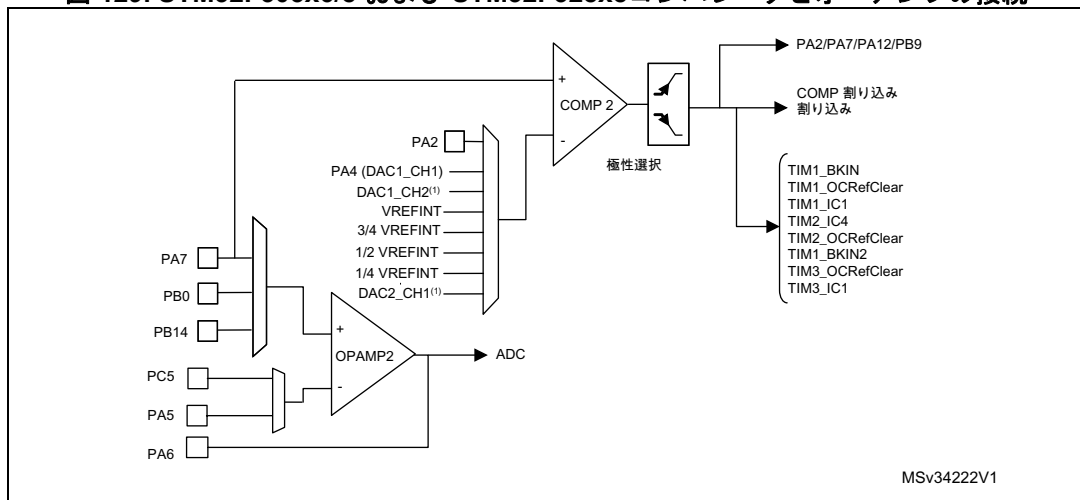


図 128. STM32F303xB/C/D/E および STM32F358xC コンパレータとオペアンプの相互接続 (パート 2)



1. STM32F303xB/C および STM32F358 デバイスのみ。

図 129. STM32F303x6/8 および STM32F328x8コンパレータとオペアンプの接続



1. DAC1\_CH2 および DAC2\_CH1 出力は直接接続されているため、PA5 および PA6 は COMP2\_INM の入力として使用できません。GPIO として使用できます。

### 18.3.4 ADC 入力としての OPAMP 出力の使用

OPAMP 出力を ADC 入力として使用するには、オペアンプが有効であり、ADC では OPAMP 出力チャネル番号が使われている必要があります。

- OPAMP1 では、ADC1 チャンネル 3 が使われています (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE のみ)。
- OPAMP2 では、ADC2 チャンネル 3 が使われています。
- OPAMP3では、ADC3 チャンネル 1 が使われています (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE のみ)。
- OPAMP4 では、ADC4 チャンネル 3 が使われています (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE のみ)。

### 18.3.5 較正

OPAMP インタフェースでは、トリミングされたオフセット値を 4 つのオペアンプに送信し続けます。起動時、これらの値はプリセットされた「出荷時」トリミング値に初期化されます。

また、各オペアンプオフセットを、ユーザがトリミングすることもできます。

OPAMP 制御レジスタの USER\_TRIM ビットを使用して、「出荷時」の値から「ユーザ」がトリミングした値に切り替えることができます。このビットは起動時にリセットされます (「出荷時」の値がオペアンプに送信されます)。

OPAMP の レールツーレール入力ステージは 2 つの差動ペアで構成されます。

- ペアのうちの 1 つは NMOS トランジスタで構成されます。
- ペアのうちの 1 つは PMOS トランジスタで構成されます。

これらの 2 つのペアは独立しているため、トリミング手順ではそれぞれ個別に較正します。NMOS 差分ペアのオフセットは TRIMOFFSETN ビットで較正され、PMOS 差分ペアのオフセットは TRIMOFFSETP ビットで較正されます。

NMOS 差分ペアを較正するには、CALON=1 および CALSEL=11 でなければなりません。この場合、内部高電圧基準 ( $0.9 \times V_{DDA}$ ) が生成され、相互接続された反転および非反転 OPAMP 入力に印加されます。OPAMP の両方の入力に印加された電圧を測定できます (OPAMP 基準電圧は、TSTREF ビット

トを通じて出力でき、ADC チャンネルに内部的に接続できます。[セクション 15: アナログデジタルコンバータ \(ADC\) \(303 ページ\)](#)を参照)。ソフトウェアによって OPAMP 制御レジスタの TRIMOFFSETN ビットを 0x00 からインクリメントし、OPAMP 制御レジスタの OUTCAL ビットが 1 から 0 に変わる最初の値にする必要があります。OUTCAL ビットがリセットされると、オフセットが正しく較正されるので対応するトリミング値を格納する必要があります。

PMOS 差動ペアの較正も同様の手順で実行されますが、2 つの違いがあります。TRIMOFFSETP ビットフィールドが使用される点と、CALSEL ビットを「01」にプログラムする必要がある点です (内部低電圧基準 ( $0.1 \times V_{DDA}$ ) が生成され、相互接続された反転および非反転 OPAMP 入力に印加されます)。

**注:** **較正モード中、正しい OUTCAL 値を取得するには、トリミング値 (TRIMOFFSETP または TRIMOFFSETN) の書き込みと OUTCAL 値の読み出しの間で OFFTRIMmax 遅延時間 (データシートの電気的特性のセクションで指定) が経過したことを確認してください。**

NMOS 差分ペアを較正するには、以下のソフトウェア手順を使用します。

1. OPAMPxEN ビットをセットして、OPAMP を有効にします。
2. USERTRIM ビットをセットして、ユーザによるオフセットトリミングを有効にします。
3. CALON ビットをセットして、VM と VP を内部基準電圧に接続します。
4. CALSEL を 11 にセットします (OPAMP 内部基準 =  $0.9 \times V_{DDA}$ )。
5. ループして、TRIMOFFSETN 値をインクリメントします。ループを終了するには、OUTCAL ビットをリセットする必要があります。この場合、TRIMOFFSETN 値を格納する必要があります。

同じソフトウェア手順を CALSEL = 01 の PMOS 差分ペア較正に適用する必要があります (OPAMP 内部基準 =  $0.1 V_{DDA}$ )。

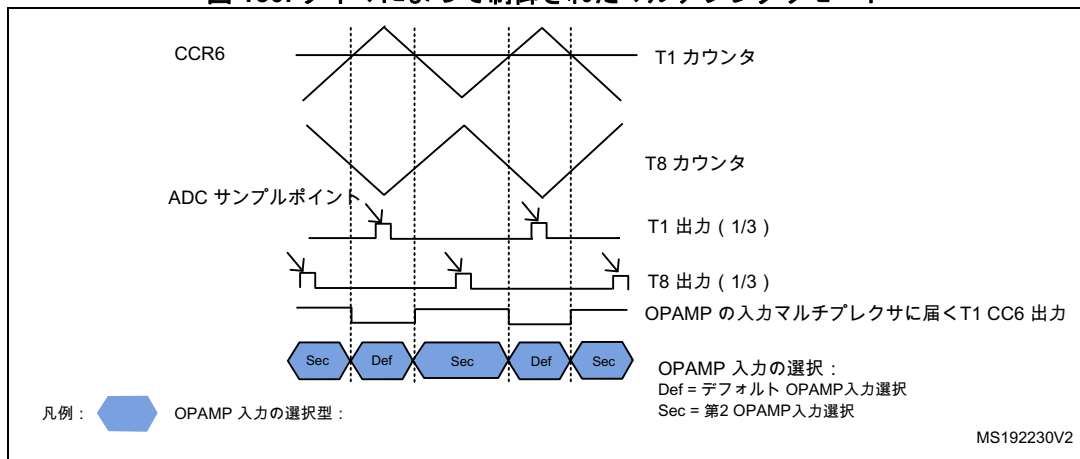
## 18.3.6 タイマによって制御されたマルチプレクサモード

OPAMP の反転および非反転入力を、自動で選択できます。この場合、1 つの入力から別の入力へ自動で切り替わります。この自動切り替えは、OPAMP の入力マルチプレクサで発生する TIM1 CC6 出力によってトリガされます。

最初のモータに次いで 2 番目のモータで 3 つのフェーズの電流を瞬時に測定する必要があるデュアルモータ制御で有用です。

自動切り替えは、OPAMP 制御レジスタの TCM\_EN ビットをセットすることによって有効になります。反転および非反転入力の選択には、OPAMP 制御レジスタの VPS\_SEL および VMS\_SEL ビットフィールドを使用します。TCM\_EN ビットがクリアされると、OPAMP 制御レジスタの VP\_SEL および VM\_SEL ビットフィールドに応じて入力を選択されます。

図 130. タイマによって制御されたマルチプレクサモード



## 18.3.7 OPAMP モード

すべてのオペアンプの入出力に、端子からアクセスできます。アンプは複数の設定環境で使用できます。

- スタンドアロンモード (外部ゲイン設定モード)
- フォロア設定モード
- PGA モード

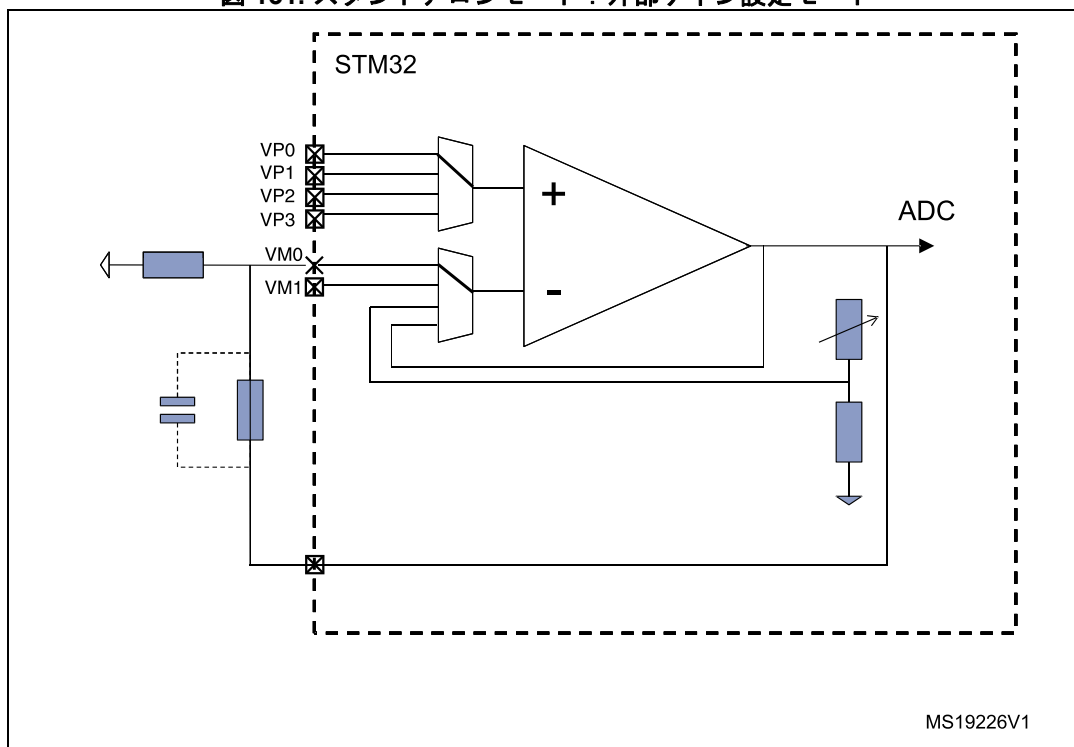
重要: 出力インピーダンスを最小限に抑えるために、アンプ出力ピンは出力パッドに直接接続されています。アンプが PGA として設定され、ADC チャンネルにのみ接続されていても、汎用 I/O として使用することはできません。

**注:** 信号のインピーダンスを、(ソースの抵抗性電圧降下が原因で) 重大な乱れをつくる入力リークが発生しないレベル以下に維持する必要があります。詳細については、データシートの電気的特性のセクションを参照してください。

### スタンドアロンモード (外部ゲイン設定モード)

外部ゲイン設定モードにより、アンプ設定とフィードバックネットワークを柔軟に選択できるようになります。このモードは、反転入力を 2 つの使用可能な I/O のうちの 1 つに接続するために、OPAMPx\_CR レジスタの VM\_SEL ビットに 00 または 01 を書き込むことによって有効になります。

図 131. スタンドアロンモード：外部ゲイン設定モード

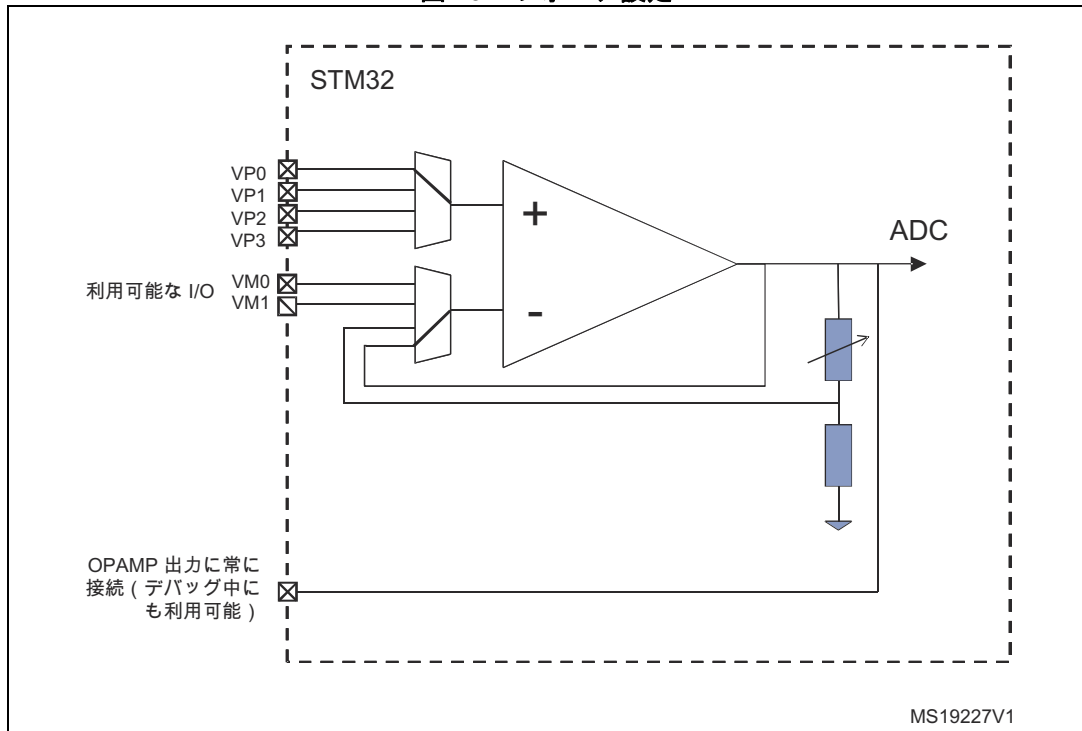


1. この図に反転設定の例を示します。コンパレータモードなど、ほかの任意のオプションを使用することもできます。

## フォロア設定モード

OPAMPx\_CR レジスタの VM\_SEL ビットを 11 に設定することで、アンプをフォロアとして設定できます。これにより、たとえば比較的高いインピーダンスで信号をバッファリングすることができます。この場合、反転入力フリーで、対応するポートを汎用 I/O として使用できます。

図 132. フォロア設定



1. この図に反転設定の例を示します。コンパレータモードなど、ほかの任意のオプションを使用することもできます。

## プログラム可能なゲインアンプモード

プログラム可能なゲインアンプ (PGA) モードは、OPAMPx\_CR レジスタの VM\_SEL ビットに 10 を書き込むことにより有効になります。ゲインは PGA\_GAIN ビットを使用してセットされます。このビットは、2 から 16 の範囲のゲインでは 0x00..0x11 にセットする必要があります。

この場合、反転入力、ゲインを設定する内蔵抵抗ネットワークの中心点に内部的に接続されます。[図 133 : PGA モード、内部ゲイン設定 \(x2/x4/x8/x16\)、未使用の反転入力](#)に、このモードでの内部接続を示します。

PGA モードの代替オプションにより、抵抗ネットワークの中心点を、非反転入力に接続された I/O のうちの 1 つに送ることができます。これは、OPAMPx\_CR レジスタの PGA\_GAIN ビットを使用して有効にできます。

- 10xx の値によりゲインが設定され、中心点を利用可能な 2 つの入力のうちの 1 つに接続します。
- 11xx の値によりゲインが設定され、中心点を利用可能な 2 番目の入力に接続します。

この機能は、たとえば PGA にローパスフィルタを追加する場合に使用できます。[図 134 : PGA モード、内部ゲイン設定 \(x2/x4/x8/x16\)、フィルタリング用の反転入力](#)を参照してください。ゲインを変更するとカットオフ周波数も変更される点に注意してください (抵抗ネットワークの要素の詳細については、データシートの電気的特性のセクションを参照)。

図 133. PGA モード、内部ゲイン設定 (x2/x4/x8/x16)、未使用の反転入力

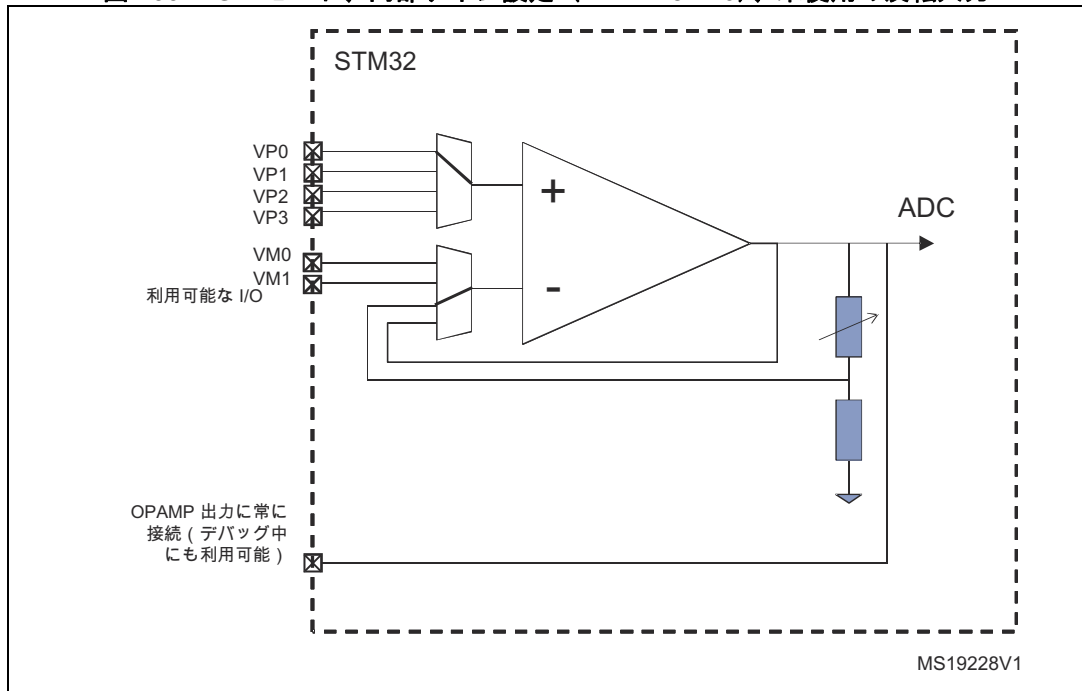
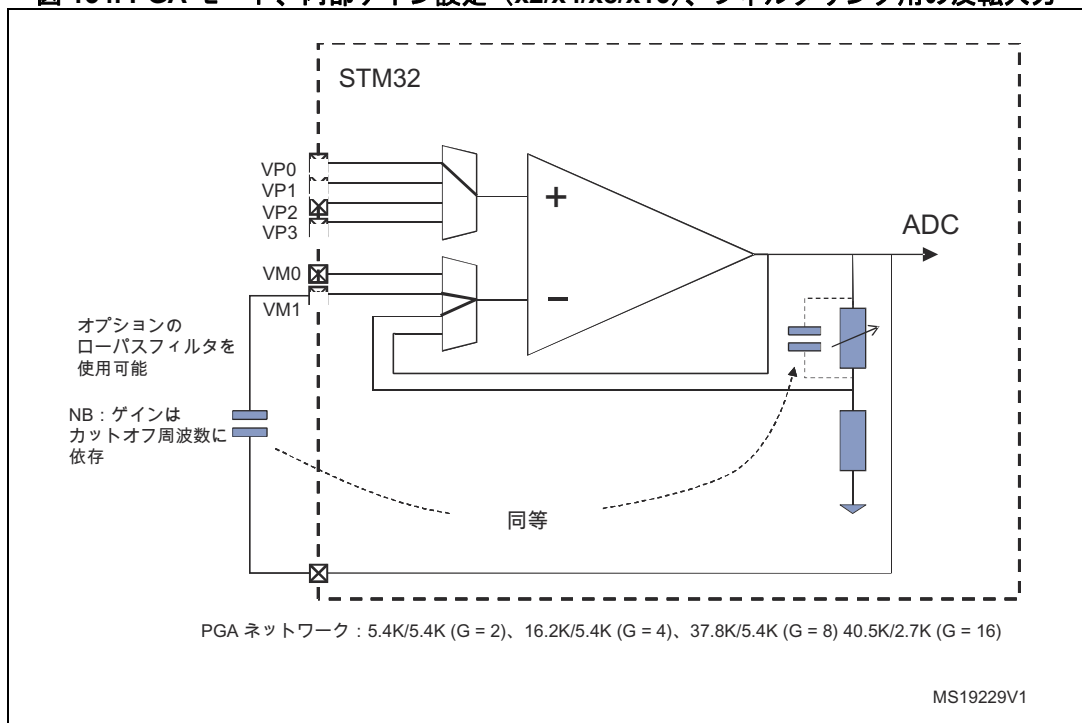


図 134. PGA モード、内部ゲイン設定 (x2/x4/x8/x16)、フィルタリング用の反転入力



## 18.4 OPAMP レジスタ

### 18.4.1 OPAMP1 制御レジスタ (OPAMP1\_CSR)

**注：** このレジスタは、STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用可能です。

アドレスオフセット：0x38

リセット値：0xXXXX 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUTCAL	TSTR EF	TRIMOFFSETN				TRIMOFFSETP				USER_ TRIM	PGA_GAIN			
rw	r	rw	rw				rw				rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGA_GAIN		CALSEL	CAL ON	VPS_SEL	VMS_SE L	TCM_ EN	VM_SEL	Res.	VP_SEL		FORCE _VP	OPAMP 1EN			
rw		rw	rw	rw	rw	rw	rw		rw		rw	rw			

#### ビット 31 LOCK：OPAMP 1 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

このビットは、OPAMP1\_CSR レジスタを読み出し専用を設定するために使用します。

0：OPAMP1\_CSR は読み書き用です。

1：OPAMP1\_CSR は読み出し専用です。

#### ビット 30 OUTCAL：

OPAMP が較正中にコンパレータとして使用される場合は OPAMP 出力ステータスフラグです。

0：非反転 < 反転

1：非反転 > 反転

#### ビット 29 TSTREF：

このビットは、ソフトウェアによってセット/クリアされます。内部基準電圧の出力に使用されます (V<sub>REFOPAMP1</sub>)。

0：V<sub>REFOPAMP1</sub> が出力されます。

1：V<sub>REFOPAMP1</sub> は出力されません。

#### ビット 28:24 TRIMOFFSETN：オフセットトリミング値 (NMOS)

#### ビット 23:19 TRIMOFFSETP：オフセットトリミング値 (PMOS)

#### ビット 18 USER\_TRIM：ユーザートリミング有効

このビットは、OPAMP オフセットの設定に使用します。

0：ユーザートリミングは無効です。

1：ユーザートリミングは有効です。



**ビット 17:14 PGA\_GAIN** : PGA モードのゲイン

0X00 = 非反転ゲイン = 2  
0X01 = 非反転ゲイン = 4  
0X10 = 非反転ゲイン = 8  
0X11 = 非反転ゲイン = 16  
1000 = 非反転ゲイン = 2 - VM0 に接続された内部フィードバック  
1001 = 非反転ゲイン = 4 - VM0 に接続された内部フィードバック  
1010 = 非反転ゲイン = 8 - VM0 に接続された内部フィードバック  
1011 = 非反転ゲイン = 16 - VM0 に接続された内部フィードバック  
1100 = 非反転ゲイン = 2 - VM1 に接続された内部フィードバック  
1101 = 非反転ゲイン = 4 - VM1 に接続された内部フィードバック  
1110 = 非反転ゲイン = 8 - VM1 に接続された内部フィードバック  
1111 = 非反転ゲイン = 16 - VM1 に接続された内部フィードバック

**ビット 13:12 CALSEL** : 較正選択

このビットは、ソフトウェアによってセット/クリアされます。CALON = 1 または FORCE\_VP = 1 の場合に、内部基準電圧の生成に使用するオフセット較正バスを選択するために使用されます。

00 =  $V_{REFOPAMP} = 3.3\% V_{DDA}$   
01 =  $V_{REFOPAMP} = 10\% V_{DDA}$   
10 =  $V_{REFOPAMP} = 50\% V_{DDA}$   
11 =  $V_{REFOPAMP} = 90\% V_{DDA}$

**ビット 11 CALON** : 較正モード有効

このビットは、ソフトウェアによってセット/クリアされます。OPAMP 内部基準電圧に VM および VP を接続する較正モードを有効にするために使用します。

0 : 較正モードは無効です。  
1 : 較正モードは有効です。

**ビット 10:9 VPS\_SEL** : OPAMP1 非反転入力 の 2 次選択

これらのビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP1 非反転入力を選択するために使用します。

00 : PA7 が OPAMP1 非反転入力として使用されます。  
01 : PA5 が OPAMP1 非反転入力として使用されます。  
10 : PA3 が OPAMP1 非反転入力として使用されます。  
11 : PA1 が OPAMP1 非反転入力として使用されます。

**ビット 8 VMS\_SEL** : OPAMP1 反転入力 の 2 次選択

このビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP1 反転入力を選択するために使用します。

0 : PC5 (VM0) が OPAMP1 反転入力として使用されます。  
1 : PA3 (VM1) が OPAMP1 反転入力として使用されます。

**ビット 7 TCM\_EN** : タイマ制御マルチプレクスモード有効

このビットは、ソフトウェアによってセット/クリアされます。反転および非反転入力のデフォルト選択 (VP\_SEL および VM\_SEL) と 2 次選択 (VPS\_SEL および VMS\_SEL) の切り替えを自動的に制御するために使用します。

**ビット 6:5 VM\_SEL** : OPAMP1 反転入力選択

これらのビットは、ソフトウェアによってセット/クリアされます。OPAMP1 反転入力を選択するために使用します。

00 : PC5 (VM0) が OPAMP1 反転入力として使用されます。  
01 : PA3 (VM1) が OPAMP1 反転入力として使用されます。  
10 : レジスタフィードバック出力 (PGA モード)  
11 : フォロアモード

ビット 4 予約済みであり、リセット値に保持する必要があります。

### ビット 3:2 VP\_SEL : OPAMP1 非反転入力選択

これらのビットは、ソフトウェアによってセット/クリアされます。OPAMP1 非反転入力を選択するために使用します。

- 00 : PA7 が OPAMP1 非反転入力として使用されます。
- 01 : PA5 が OPAMP1 非反転入力として使用されます。
- 10 : PA3 が OPAMP1 非反転入力として使用されます。
- 11 : PA1 が OPAMP1 非反転入力として使用されます。

### ビット 1 FORCE\_VP :

このビットは、較正された基準電圧を強制的に非反転入力に印加し、外部接続を無効にします。

- 0 : 通常動作モード。非反転入力は入力に接続されます。
- 1 : 較正モード。非反転入力は較正基準電圧に接続されます。

### ビット 0 OPAMP1EN : OPAMP1 イネーブル

このビットは、ソフトウェアによってセット/クリアされます。OPAMP1 を有効にするために使用します。

- 0 : OPAMP1 は無効です。
- 1 : OPAMP1 は有効です。

## 18.4.2 OPAMP2 制御レジスタ (OPAMP2\_CSR)

アドレスオフセット : 0x3C

リセット値 : 0xXXXX 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUTCAL	TSTREF	TRIMOFFSETN				TRIMOFFSETP				USERTRIM	PGA_GAIN			
rw	r	rw	rw				rw				rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGA_GAIN		CALSEL	CALON	VPS_SEL	VMS_SEL	TCM_EN	VM_SEL	Res.	VP_SEL		FORCE_VP	OPAMP2EN			
rw		rw	rw	rw	rw	rw	rw		rw		rw	rw			

### ビット 31 LOCK : OPAMP 2 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

このビットは、OPAMP2\_CSR レジスタを読み出し専用を設定するために使用します。

- 0 : OPAMP2\_CSR は読み書き用です。
- 1 : OPAMP2\_CSR は読み出し専用です。

### ビット 30 OUTCAL :

OPAMP が較正中にコンパレータとして使用される場合は OPAMP 出カステータスフラグです。

- 0 : 非反転 < 反転
- 1 : 非反転 > 反転

### ビット 29 TSTREF :

このビットは、ソフトウェアによってセット/クリアされます。内部基準電圧の出力に使用されます (V<sub>REFOPAMP2</sub>)。

- 0 : V<sub>REFOPAMP2</sub> が出力されます。
- 1 : V<sub>REFOPAMP2</sub> は出力されません。

ビット 28:24 TRIMOFFSETN : オフセットトリミング値 (NMOS)

ビット 23:19 TRIMOFFSETP : オフセットトリミング値 (PMOS)

**ビット 18 USER\_TRIM** : ユーザトリミング有効

このビットは、OPAMP オフセットの設定に使用します。

0 : ユーザトリミングは無効です。

1 : ユーザトリミングは有効です。

**ビット 17:14 PGA\_GAIN** : PGA モードのゲイン

0X00 = 非反転ゲイン = 2

0X01 = 非反転ゲイン = 4

0X10 = 非反転ゲイン = 8

0X11 = 非反転ゲイン = 16

1000 = 非反転ゲイン = 2 - VM0 に接続された内部フィードバック

1001 = 非反転ゲイン = 4 - VM0 に接続された内部フィードバック

1010 = 非反転ゲイン = 8 - VM0 に接続された内部フィードバック

1011 = 非反転ゲイン = 16 - VM0 に接続された内部フィードバック

1100 = 非反転ゲイン = 2 - VM1 に接続された内部フィードバック

1101 = 非反転ゲイン = 4 - VM1 に接続された内部フィードバック

1110 = 非反転ゲイン = 8 - VM1 に接続された内部フィードバック

1111 = 非反転ゲイン = 16 - VM1 に接続された内部フィードバック

**ビット 13:12 CALSEL** : 較正選択

このビットは、ソフトウェアによってセット/クリアされます。CALON = 1 または FORCE\_VP = 1 の場合に、内部基準電圧の生成に使用するオフセット較正バスを選択するために使用されます。

00 =  $V_{REFOPAMP} = 3.3\% V_{DDA}$

01 =  $V_{REFOPAMP} = 10\% V_{DDA}$

10 =  $V_{REFOPAMP} = 50\% V_{DDA}$

11 =  $V_{REFOPAMP} = 90\% V_{DDA}$

**ビット 11 CALON** : 較正モード有効

このビットは、ソフトウェアによってセット/クリアされます。OPAMP 内部基準電圧に VM および VP を接続する較正モードを有効にするために使用します。

0 : 較正モードは無効です。

1 : 較正モードは有効です。

**ビット 10:9 VPS\_SEL** : OPAMP2 非反転入力の 2 次選択

これらのビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP2 非反転入力を選択するために使用します。

00 : PD14 が OPAMP2 非反転入力として使用されます (STM32F303xB/C および STM32F358C デバイスのみ)

01 : PB14 が OPAMP2 非反転入力として使用されます。

10 : PB0 が OPAMP2 非反転入力として使用されます。

11 : PA7 が OPAMP2 非反転入力として使用されます。

**ビット 8 VMS\_SEL** : OPAMP2 反転入力の 2 次選択

このビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP2 反転入力を選択するために使用します。

0 : PC5 (VM0) が OPAMP2 反転入力として使用されます。

1 : PA5 (VM1) が OPAMP2 反転入力として使用されます。

**ビット 7 TCM\_EN** : タイマ制御マルチプレクスモード有効

このビットは、ソフトウェアによってセット/クリアされます。反転および非反転入力のデフォルト選択 (VP\_SEL および VM\_SEL) と 2 次選択 (VPS\_SEL および VMS\_SEL) の切り替えを自動的に制御するために使用します。

### ビット 6:5 VM\_SEL : OPAMP2 反転入力選択

これらのビットは、ソフトウェアによってセット/クリアされます。OPAMP2 反転入力を選択するために使用します。

- 00 : PC5 (VM0) が OPAMP2 反転入力として使用されます。
- 01 : PA5 (VM1) が OPAMP2 反転入力として使用されます。
- 10 : レジスタフィードバック出力 (PGA モード)
- 11 : フォロアモード

ビット 4 予約済みであり、リセット値に保持する必要があります。

### ビット 3:2 VP\_SEL : OPAMP2 非反転入力選択

これらのビットは、ソフトウェアによってセット/リセットされます。OPAMP2 非反転入力を選択するために使用します。

- 00 : PD14 が OPAMP2 非反転入力として使用されます (STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ)
- 01 : PB14 が OPAMP2 非反転入力として使用されます。
- 10 : PB0 が OPAMP2 非反転入力として使用されます。
- 11 : PA7 が OPAMP2 非反転入力として使用されます。

### ビット 1 FORCE\_VP :

このビットは、較正された基準電圧を強制的に非反転入力に印加し、外部接続を無効にします。

- 0 : 通常動作モード。非反転入力は入力に接続されます。
- 1 : 較正モード。非反転入力は較正基準電圧に接続されます。

### ビット 0 OPAMP2EN : OPAMP2 イネーブル

このビットは、ソフトウェアによってセット/クリアされます。OPAMP2 を選択するために使用します。

- 0 : OPAMP2 は無効です。
- 1 : OPAMP2 は有効です。

## 18.4.3 OPAMP3 制御レジスタ (OPAMP3\_CSR)

**注 :** このレジスタは、STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用できます。

アドレスオフセット : 0x40

リセット値 : 0xXXXX 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT CAL	TSTR EF	TRIMOFFSETN				TRIMOFFSETP				USER TRIM	PGA_GAIN			
rw	r	rw	rw				rw				rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGA_GAIN		CALSEL	CAL ON	VPS_SEL	VMS_SEL	TCM_EN	VM_SEL	Res.	VP_SEL		FORCE_VP	OPAMP 3EN			
rw		rw	rw	rw	rw	rw	rw		rw		rw	rw			

**ビット 31 LOCK** : OPAMP 3 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

このビットは、OPAMP3\_CSR レジスタを読み出し専用を設定するために使用します。

0 : OPAMP3\_CSR は読み書き用です。

1 : OPAMP3\_CSR は読み出し専用です。

**ビット 30 OUTCAL** :

OPAMP が較正中にコンパレータとして使用される場合は OPAMP 出力ステータスフラグです。

0 : 非反転 < 反転

1 : 非反転 > 反転

**ビット 29 TSTREF** :

このビットは、ソフトウェアによってセット/クリアされます。内部基準電圧の出力に使用されます ( $V_{REFOPAMP3}$ )。

0 :  $V_{REFOPAMP3}$  が出力されます。

1 :  $V_{REFOPAMP3}$  は出力されません。

**ビット 28:24 TRIMOFFSETN** : オフセットトリミング値 (NMOS)**ビット 23:19 TRIMOFFSETP** : オフセットトリミング値 (PMOS)**ビット 18 USER\_TRIM** : ユーザトリミング有効

このビットは、OPAMP オフセットの設定に使用します。

0 : ユーザトリミングは無効です。

1 : ユーザトリミングは有効です。

**ビット 17:14 PGA\_GAIN** : PGA モードのゲイン

0X00 = 非反転ゲイン = 2

0X01 = 非反転ゲイン = 4

0X10 = 非反転ゲイン = 8

0X11 = 非反転ゲイン = 16

1000 = 非反転ゲイン = 2 - VM0 に接続された内部フィードバック

1001 = 非反転ゲイン = 4 - VM0 に接続された内部フィードバック

1010 = 非反転ゲイン = 8 - VM0 に接続された内部フィードバック

1011 = 非反転ゲイン = 16 - VM0 に接続された内部フィードバック

1100 = 非反転ゲイン = 2 - VM1 に接続された内部フィードバック

1101 = 非反転ゲイン = 4 - VM1 に接続された内部フィードバック

1110 = 非反転ゲイン = 8 - VM1 に接続された内部フィードバック

1111 = 非反転ゲイン = 16 - VM1 に接続された内部フィードバック

**ビット 13:12 CALSEL** : 較正選択

このビットは、ソフトウェアによってセット/クリアされます。CALON = 1 または FORCE\_VP = 1 の場合に、内部基準電圧の生成に使用するオフセット較正バスを選択するために使用されます。

00 =  $V_{REFOPAMP} = 3.3\% V_{DDA}$

01 =  $V_{REFOPAMP} = 10\% V_{DDA}$

10 =  $V_{REFOPAMP} = 50\% V_{DDA}$

11 =  $V_{REFOPAMP} = 90\% V_{DDA}$

**ビット 11 CALON** : 較正モード有効

このビットは、ソフトウェアによってセット/クリアされます。内部基準電圧に VM および VP を接続する較正モードを有効にできます。

0 : 較正モードは無効です。

1 : 較正モードは有効です。

**ビット 10:9 VPS\_SEL : OPAMP3 非反転入力 の 2 次選択**

これらのビットは、ソフトウェアによってセット/リセットされます。TCM\_EN = 1 の場合に OPAMP3 非反転入力を選択するために使用します。

- 00 : PB13 が OPAMP3 非反転入力として使用されます。
- 01 : PA5 が OPAMP3 非反転入力として使用されます。
- 10 : PA1 が OPAMP3 非反転入力として使用されます。
- 11 : PB0 が OPAMP3 非反転入力として使用されます。

**ビット 8 VMS\_SEL : OPAMP3 反転入力 の 2 次選択**

このビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP3 反転入力を選択するために使用します。

- 0 : PB10 (VM0) が OPAMP3 反転入力として使用されます。
- 1 : PB2 (VM1) が OPAMP3 反転入力として使用されます。

**ビット 7 TCM\_EN : タイマ制御マルチプレクスモード有効**

このビットは、ソフトウェアによってセット/クリアされます。反転および非反転入力のデフォルト選択 (VP\_SEL および VM\_SEL) と 2 次選択 (VPS\_SEL および VMS\_SEL) の切り替えを自動的に制御するために使用します。

**ビット 6:5 VM\_SEL : OPAMP3 反転入力選択**

これらのビットは、ソフトウェアによってセット/リセットされます。OPAMP3 反転入力を選択するために使用します。

- 00 : PB10 (VM0) が OPAMP3 反転入力として使用されます。
- 01 : PB2 (VM1) が OPAMP3 反転入力として使用されます。
- 10 : レジスタフィードバック出力 (PGA モード)
- 11 : フォロアモード

ビット 4 予約済みであり、リセット値に保持する必要があります。

**ビット 3:2 VP\_SEL : OPAMP3 非反転入力選択**

これらのビットは、ソフトウェアによってセット/クリアされます。OPAMP3 非反転入力を選択するために使用します。

- 00 : PB13 が OPAMP3 非反転入力として使用されます。
- 01 : PA5 が OPAMP3 非反転入力として使用されます。
- 10 : PA1 が OPAMP3 非反転入力として使用されます。
- 11 : PB0 が OPAMP3 非反転入力として使用されます。

**ビット 1 FORCE\_VP :**

このビットは、較正された基準電圧を強制的に非反転入力に印加し、外部接続を無効にします。

- 0 : 通常動作モード。非反転入力は入力に接続されます。
- 1 : 較正モード。非反転入力は較正基準電圧に接続されます。

**ビット 0 OPAMP3EN : OPAMP3 イネーブル**

このビットは、ソフトウェアによってセット/クリアされます。OPAMP3 を有効にするために使用します。

- 0 : OPAMP3 は無効です。
- 1 : OPAMP3 は有効です。

## 18.4.4 OPAMP4 制御レジスタ (OPAMP4\_CSR)

**注:** このレジスタは、STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用できます。

アドレスオフセット : 0x44

リセット値 : 0xFFFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT CAL	TSTREF	TRIMOFFSETN				TRIMOFFSETP				USER_TRIM	PGA_GAIN			
rw	r	rw	rw				rw				rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGA_GAIN		CALSEL	CALON	VPS_SEL	VMS_SEL	TCM_EN	VM_SEL	Res.	VP_SEL		FORCE_VP	OPAMP4EN			
rw		rw	rw	rw	rw	rw	rw		rw		rw	rw			

### ビット 31 LOCK : OPAMP 4 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

このビットは、OPAMP4\_CSR レジスタを読み出し専用に変更するために使用します。

0 : OPAMP4\_CSR は読み書き用です。

1 : OPAMP4\_CSR は読み出し専用です。

### ビット 30 OUTCAL :

OPAMP が較正中にコンパレータとして使用される場合は OPAMP 出力ステータスフラグです。

0 : 非反転 < 反転

1 : 非反転 > 反転

### ビット 29 TSTREF :

このビットは、ソフトウェアによってセット/クリアされます。内部基準電圧の出力に使用されます (V<sub>REFOPAMP4</sub>)。

0 : V<sub>REFOPAMP4</sub> が出力されます。

1 : V<sub>REFOPAMP4</sub> は出力されません。

ビット 28:24 TRIMOFFSETN : オフセットトリミング値 (NMOS)

ビット 23:19 TRIMOFFSETP : オフセットトリミング値 (PMOS)

### ビット 18 USER\_TRIM : ユーザートリミング有効

このビットは、OPAMP オフセットの設定に使用します。

0 : ユーザートリミングは無効です。

1 : ユーザートリミングは有効です。

**ビット 17:14 PGA\_GAIN** : PGA モードのゲイン

0X00 = 非反転ゲイン = 2  
0X01 = 非反転ゲイン = 4  
0X10 = 非反転ゲイン = 8  
0X11 = 非反転ゲイン = 16  
1000 = 非反転ゲイン = 2 - VM0 に接続された内部フィードバック  
1001 = 非反転ゲイン = 4 - VM0 に接続された内部フィードバック  
1010 = 非反転ゲイン = 8 - VM0 に接続された内部フィードバック  
1011 = 非反転ゲイン = 16 - VM0 に接続された内部フィードバック  
1100 = 非反転ゲイン = 2 - VM1 に接続された内部フィードバック  
1101 = 非反転ゲイン = 4 - VM1 に接続された内部フィードバック  
1110 = 非反転ゲイン = 8 - VM1 に接続された内部フィードバック  
1111 = 非反転ゲイン = 16 - VM1 に接続された内部フィードバック

**ビット 13:12 CALSEL** : 較正選択

このビットは、ソフトウェアによってセット/クリアされます。CALON = 1 または FORCE\_VP = 1 の場合に、内部基準電圧の生成に使用するオフセット較正バスを選択するために使用されます。

00 = VREFOPAMP = 3.3% V<sub>DDA</sub>  
01 = VREFOPAMP = 10% V<sub>DDA</sub>  
10 = VREFOPAMP = 50% V<sub>DDA</sub>  
11 = VREFOPAMP = 90% V<sub>DDA</sub>

**ビット 11 CALON** : 較正モード有効

このビットは、ソフトウェアによってセット/クリアされます。OPAMP 内部基準電圧に VM および VP を接続する較正モードを有効にするために使用します。

0 : 較正モードは無効です。  
1 : 較正モードは有効です。

**ビット 10:9 VPS\_SEL** : OPAMP4 非反転入力の 2 次選択

これらのビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP4 非反転入力を選択するために使用します。

00 : PD11 が OPAMP4 非反転入力として使用されます。  
01 : PB11 が OPAMP4 非反転入力として使用されます。  
10 : PA4 が OPAMP4 非反転入力として使用されます。  
11 : PB13 が OPAMP4 非反転入力として使用されます。

**ビット 8 VMS\_SEL** : OPAMP4 反転入力の 2 次選択

このビットは、ソフトウェアによってセット/クリアされます。TCM\_EN = 1 の場合に OPAMP4 反転入力を選択できます。

0 : PB10 (VM0) が OPAMP4 反転入力として使用されます。  
1 : PD8 (VM1) が OPAMP4 反転入力として使用されます。

**ビット 7 TCM\_EN** : タイマ制御マルチプレクスモード有効

このビットは、ソフトウェアによってセット/クリアされます。反転および非反転入力のデフォルト選択 (VP\_SEL および VM\_SEL) と 2 次選択 (VPS\_SEL および VMS\_SEL) の切り替えを自動的に制御するために使用します。

**ビット 6:5 VM\_SEL** : OPAMP4 反転入力選択

これらのビットは、ソフトウェアによってセット/リセットされます。OPAMP4 反転入力を選択するために使用します。

00 : PB10 (VM0) が OPAMP4 反転入力として使用されます。  
01 : PD8 (VM1) が OPAMP4 反転入力として使用されます。  
10 : レジスタフィードバック出力 (PGA モード)  
11 : フォロアモード

ビット 4 予約済みであり、リセット値に保持する必要があります。



**ビット 3:2 VP\_SEL** : OPAMP4 非反転入力選択

これらのビットは、ソフトウェアによってセット/クリアされます。OPAMP4 非反転入力を選択するために使用します。

00 : PD11 が OPAMP4 非反転入力として使用されます。

01 : PB11 が OPAMP4 非反転入力として使用されます。

10 : PA4 が OPAMP4 非反転入力として使用されます。

11 : PB13 が OPAMP4 非反転入力として使用されます。

**ビット 1 FORCE\_VP** :

この説明は、本書の将来のバージョンで追加されます。

**ビット 0 OPAMP4EN** : OPAMP4 イネーブル

このビットは、ソフトウェアによってセット/クリアされます。OPAMP4 を有効にできます。

0 : OPAMP4 は無効です。

1 : OPAMP4 は有効です。

## 18.4.5 OPAMP レジスタマップ

次の表に OPAMP レジスタの一覧を示します。

表 111. OPAMP レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x38	OPAMP1_CSR	LOCK	OUTCAL	TSTREF			TRIMOFFSETN					TRIMOFFSETP			USER_TRIM		PGA_GAIN			CALSEL		CALON	VPS_SEL	VMS_SEL	TCM_LEN	VM_SEL		Res.		VP_SEL	FORCE_VP	OPAMP1EN	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	OPAMP2_CSR	LOCK	OUTCAL	TSTREF			TRIMOFFSETN					TRIMOFFSETP			USER_TRIM		PGA_GAIN			CALSEL		CALON	VPS_SEL	VMS_SEL	TCM_LEN	VM_SEL		Res.		VP_SEL	FORCE_VP	OPAMP2EN	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	OPAMP3_CSR	LOCK	OUTCAL	TSTREF			TRIMOFFSETN					TRIMOFFSETP			USER_TRIM		PGA_GAIN			CALSEL		CALON	VPS_SEL	VMS_SEL	TCM_LEN	VM_SEL		Res.		VP_SEL	FORCE_VP	OPAMP3EN	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	OPAMP4_CSR	LOCK	OUTCAL	TSTREF			TRIMOFFSETN					TRIMOFFSETP			USER_TRIM		PGA_GAIN			CALSEL		CALON	VPS_SEL	VMS_SEL	TCM_LEN	VM_SEL		Res.		VP_SEL	FORCE_VP	OPAMP4EN	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 19 タッチセンシングコントローラ (TSC)

### 19.1 概要

タッチセンシングコントローラを使用することで、どんなアプリケーションにも簡単に静電容量検出機能を追加することができます。静電容量検出技術とは、誘電材料（ガラス、プラスチックなど）により直接触れることがないように保護されている電極の近くに近づけた指を検出することができる技術です。指（または何らかの導電性物体）によって発生する静電容量の変化は、表面電荷移動取得原理に基づき、実証済みの方法で測定されます。

タッチセンシングコントローラは、STMTouch タッチ検出ファームウェアライブラリによってフルサポートされています。この自由に利用できるライブラリにより、タッチ検出機能がエンドアプリケーションに確実に実装されます。

### 19.2 TSC の主な機能

タッチセンシングコントローラは、以下のような主な機能を備えています。

- 実績のある安定した表面電荷移動取得原理を採用。
- 最大 24 本の静電容量検出チャンネルに対応。
- 最大 8 本の静電容量検出チャンネルの同時取得により、非常に優れた応答時間を実現。
- スペクトル拡散機能により、ノイズの多い環境でのシステムの安定性が向上。
- 電荷移動取得シーケンスのハードウェア完全管理
- プログラム可能な電荷移動周波数
- プログラム可能なサンプリングコンデンサの I/O ピン
- プログラム可能なチャンネルの I/O ピン
- 最大カウント値をプログラムできるので、チャンネル障害時の取得時間の延長を回避。
- 割り込み機能を持つ専用の取得完了フラグおよび最大カウントエラーフラグ
- 最高 3 本の静電容量検出チャンネルに対して 1 個のサンプリングコンデンサとすることで、システム構成部品の個数を削減。
- 近接、タッチキー、リニア、およびロータリタイプのタッチセンサに対応。
- STMTouch タッチ検出ファームウェアライブラリで動作。

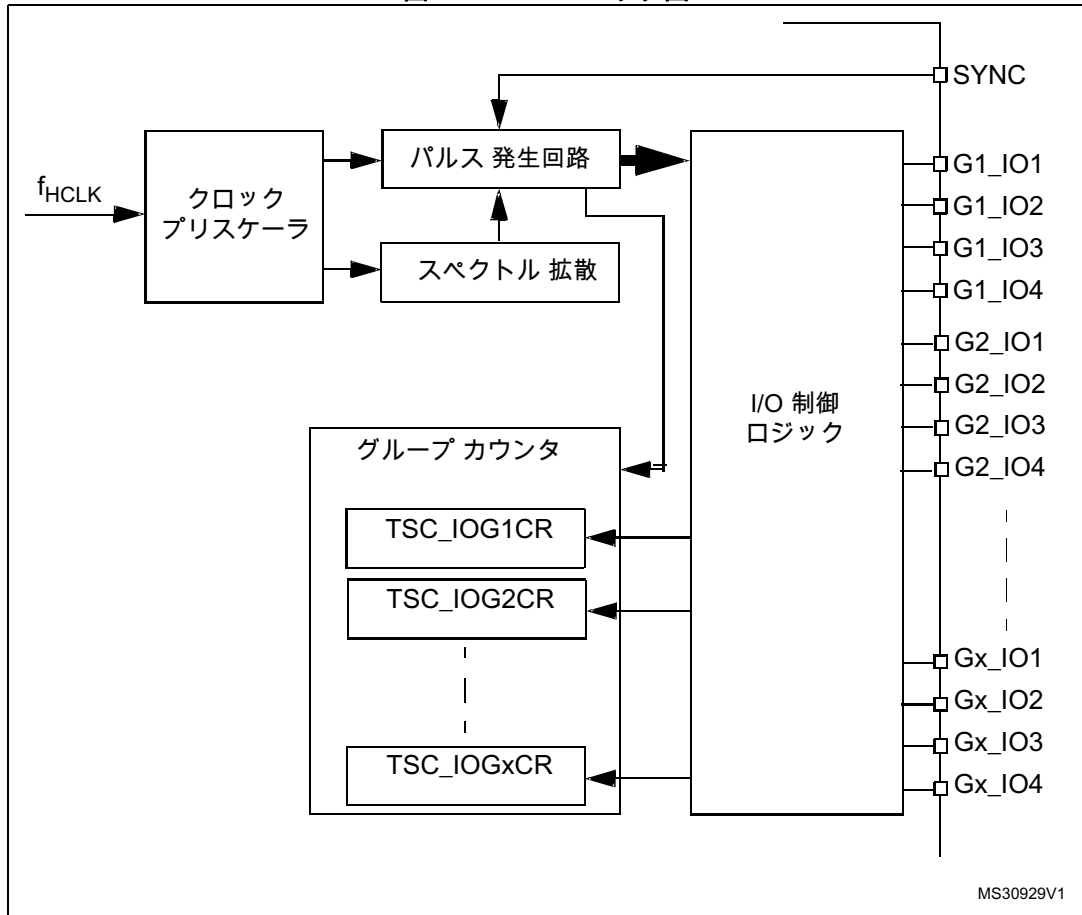
**注：** 静電容量検出チャンネルの本数は、パッケージのサイズ、および使用可能な I/O があるかどうかによって決まります。

## 19.3 TSC の機能説明

### 19.3.1 TSC ブロック図

タッチセンシングコントローラのブロック図を [図 135 : TSC ブロック図](#) に示します。

図 135. TSC ブロック図



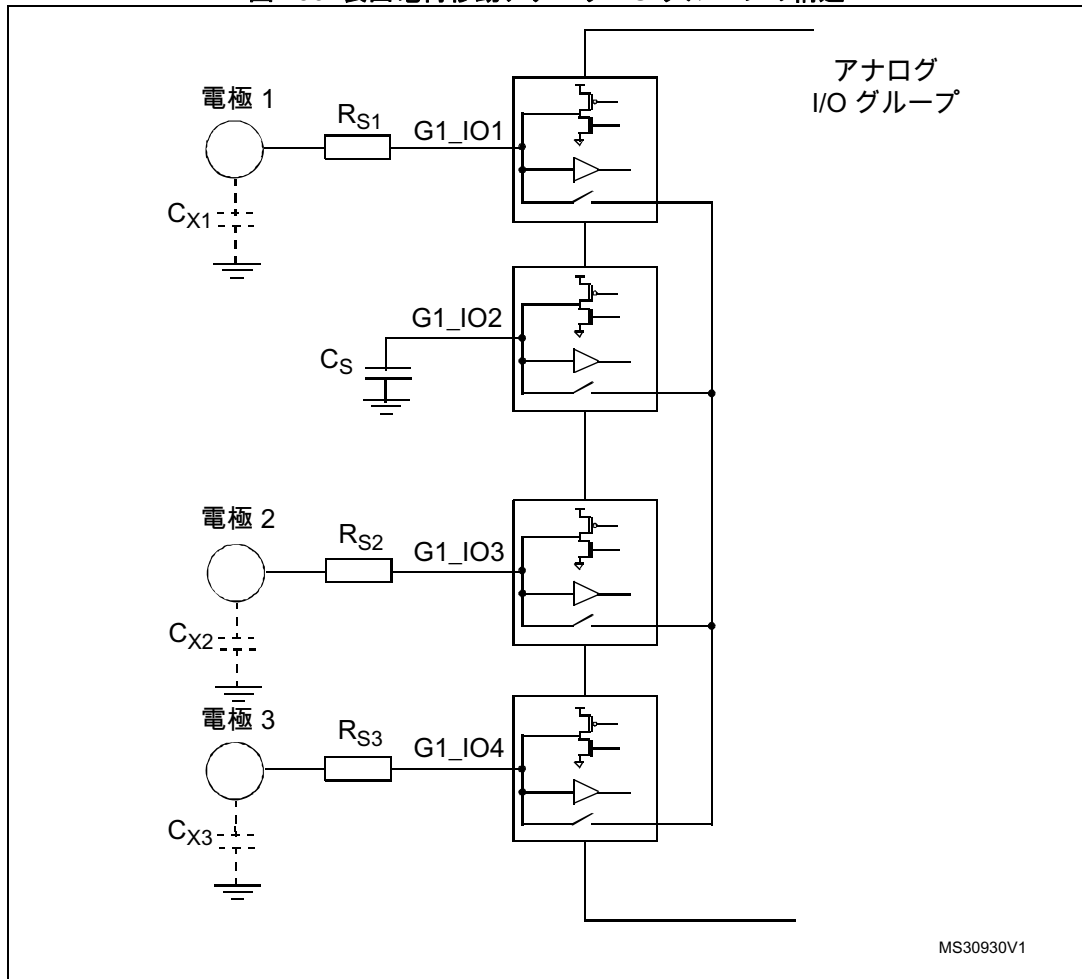
### 19.3.2 表面電荷移動取得の概要

表面電荷移動取得は、静電容量を計測するための実績のある、安定した効果的な方法です。この方法では、シングルエンド電極型で動作するために最低限必要な数の外部コンポーネントが使用されます。この取得は、4つのGPIOから成るアナログI/Oグループを中心に設計されています（[図 136](#)を参照）。いくつかのアナログI/Oグループを使用することができ、それにより複数の静電容量検出チャンネルを同時に取得し、サポートできる静電容量検出チャンネルの数を増やすことができます。1つのアナログI/Oグループ内で行われる静電容量検出チャンネルの取得はシーケンシャルです。

GPIOのうちの一つは、サンプリングコンデンサ  $C_S$  専用です。1つのアナログI/Oグループでは、一度に1つのサンプリングコンデンサI/Oだけを有効にしてください。

残りのGPIOは電極に使用され、一般にチャンネルと呼ばれます。特定のニーズ（近接検出など）については、1つのアナログI/Oグループで同時に1つ以上のチャンネルを有効にすることができます。

図 136. 表面電荷移動アナログ I/O グループの構造



注:  $Gx\_IOy$  の  $x$  はアナログ I/O グループの番号、 $y$  は選択したグループ内での GPIO の番号を表します。

表面電荷移動取得の基本動作は、電極の静電容量 ( $C_X$ ) を充電することと、蓄積された電荷の一部をサンプリングコンデンサ ( $C_S$ ) へ移動することです。このシーケンスは、 $C_S$  の電圧が所定の閾値 (弊社の場合、 $V_{IH}$ ) に達するまで繰り返されます。閾値に達するために必要な電荷移動の回数は、電極の静電容量の大きさで直接表されます。

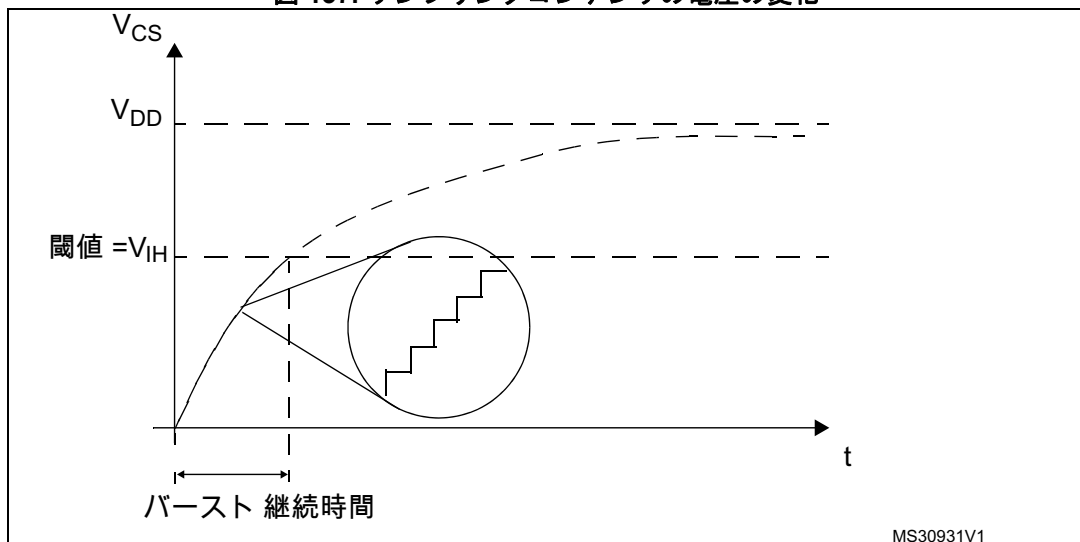
表 112 に、静電容量検出チャンネル 1 の電荷移動取得シーケンスの詳細を示します。 $C_S$  の電圧が所定の閾値に達するまで、状態 3~7 が繰り返されます。その他のチャンネルの取得にも同じシーケンスが適用されます。電極のシリアルレジスタ  $R_S$  によって、この方法の ESD (静電気放電) イミュニティが向上します。

表 112. 取得シーケンスの概要

状態	G1_IO1 (電極)	G1_IO2 (サンプリング)	G1_IO3 (電極)	G1_IO4 (電極)	状態の説明
1	入力 フローティング (アナログ スイッチ閉)	出力オープン ドレインロー (アナログ スイッチ閉)	入力フローティング (アナログスイッチ閉)		すべての $C_X$ および $C_S$ を放電。
2	入力フローティング				デッドタイム
3	出力プッシュ プルハイ	入力フローティング			$C_{X1}$ を充電。
4	入力フローティング				デッドタイム
5	入力フローティング (アナログスイッチ閉)		入力フローティング		$C_{X1}$ から $C_S$ への 電荷移動
6	入力フローティング				デッドタイム
7	入力フローティング				$C_S$ の電圧を測定。

サンプリングコンデンサ  $C_S$  の電圧の経時変化を以下に示します。

図 137. サンプリングコンデンサの電圧の変化



### 19.3.3 リセットおよびクロック

TSC クロックソースは AHB クロック (HCLK) です。2 個のプログラム可能なプリスケアラを使用し、パルス発生回路およびスペクトル拡散内部クロックを生成します。

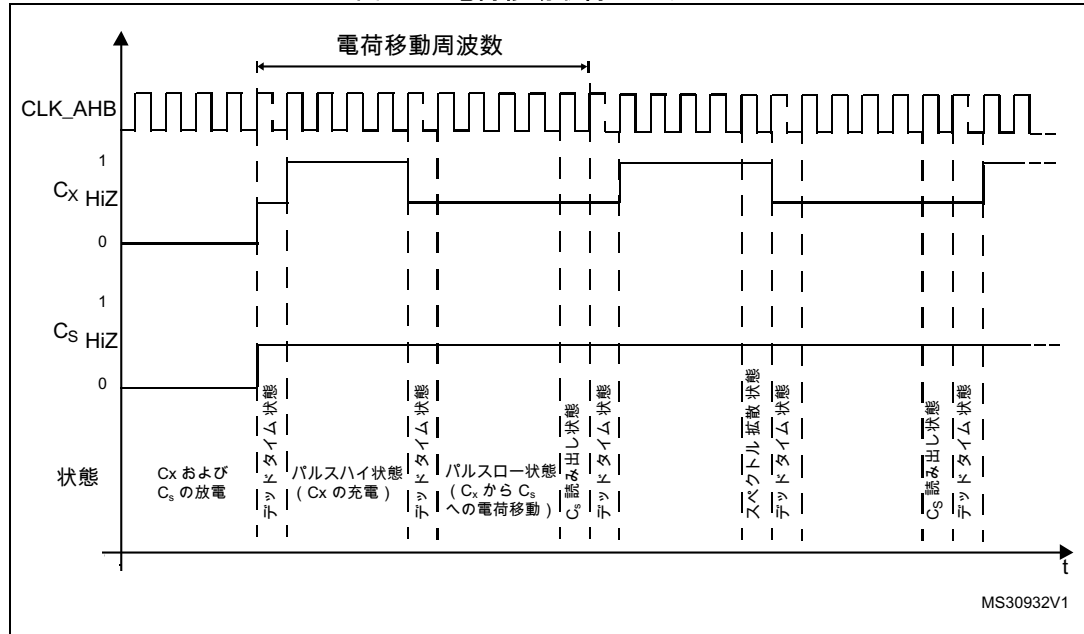
- パルス発生回路クロック (PGCLK) は、TSC\_CR レジスタの PGpsc[2:0] ビットを使用して定義されます。
- スペクトル拡散クロック (SSCLK) は、TSC\_CR レジスタの SSPsc ビットを使用して定義されます。

リセットおよびクロックコントローラ (RCC) は、タッチセンシングコントローラのクロックを有効にし、このペリフェラルをリセットするための専用ビットを提供します。詳細については、[セクション 9: リセットおよびクロック制御 \(RCC\)](#) を参照してください。

## 19.3.4 電荷移動取得シーケンス

電荷移動取得シーケンスの例を [図 138](#) に示します。

図 138. 電荷移動取得シーケンス



より高い柔軟性を得るために、電荷移動周波数は完全に設定可能です。パルスのハイ状態 (Cx の充電) およびパルスのロー状態 (Cx から Cs への電荷の移動) のどちらも、継続時間は TSC\_CR レジスタの CTPH[3:0] ビットおよび CTPL[3:0] ビットを使用して定義することができます。パルスのハイ状態およびロー状態の継続時間の標準範囲は、500 ns~2 μs です。電極の静電容量を正確に測定できるようにするには、Cx が常にフル充電となるように、パルスのハイ状態の継続時間を設定する必要があります。

電荷移動取得シーケンスを最適化するため、パルスのハイ状態とロー状態の間にデッドタイムが挿入されます。デッドタイムとは、サンプリングコンデンサの I/O とチャネルの I/O の両方が入力フローティング状態にある時間のことです。この状態の継続時間は HCLK の 2 周期です。

スペクトル拡散機能が有効であれば、パルスのハイ状態の最後に SSCLK クロックの周期の変数が追加されます。

サンプリングコンデンサの I/O の読み出しは、Cs の電圧が所定の閾値に達したかどうかを確認するために、パルスのロー状態の最後に行われ、その継続時間は HCLK の 1 周期分です。

**注：** 次の TSC 制御レジスタ設定は禁止されています。

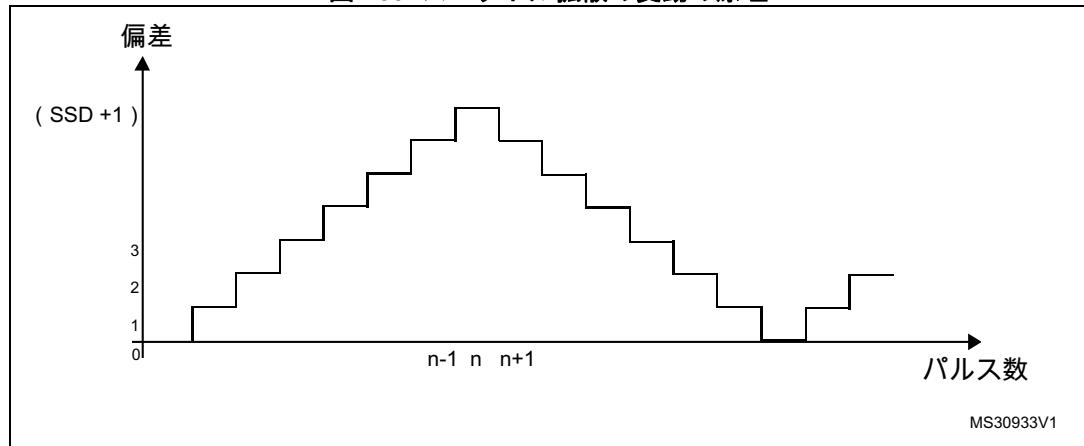
- ビット PGPSC を“0”に、ビット CTPL を“0”にセットすること
- ビット PGPSC を“0”に、ビット CTPL を“1”にセットすること
- ビット PGPSC を“1”に、ビット CTPL を“0”にセットすること

## 19.3.5 スペクトル拡散機能

スペクトル拡散機能により、電荷移動周波数の変動を生成することができます。これは、ノイズの多い環境において電荷移動取得の安定性を向上させ、さらに誘導放出を低減するために行われます。周波数の最大変動幅は、公称電荷移動周期の 10~50% です。たとえば、公称電荷移動周波数が 250 kHz (4 μs) の場合、典型的なスペクトル拡散偏差は 10% (400 ns) であり、この値から最小電荷移動周波数 (227 kHz 以下) が導き出されます。

実際には、スペクトル拡散とは、以下に示す原理を用いて、SSCLK 周期の変数を パルスのハイ状態に追加することを言います。

図 139. スペクトル拡散の変動の原理



次の表に、異なるHCLK 設定での最大周波数偏差を示します。

表 113. スペクトル拡散偏差と AHB クロック周波数

f <sub>HCLK</sub>	スペクトル拡散の刻み	スペクトル拡散最大偏差
24 MHz	41.6 ns	10666.6 ns
48 MHz	20.8 ns	5333.3 ns

スペクトル拡散機能は、TSC\_CR レジスタの SSE ビットを使用して有効/無効にできます。周波数偏差は、TSC\_CR レジスタの SSPSC および SSD[6:0] ビットを介して、デバイス HCLK クロック周波数および選択された電荷移動周波数を収容できるように設定することも可能です。

## 19.3.6 最大カウントエラー

最大カウントエラーは、静電容量検出チャンネルの障害により取得時間が長くなるのを防ぎます。アナログ I/O グループのカウンタに対して最大カウント値を指定するのがこのエラーです。この最大カウント値は TSC\_CR レジスタの MCV[2:0] ビットを使用して指定されます。取得グループのカウンタがこの最大値に到達するとすぐに、処理中の取得は停止し、取得完了 (EOAF ビット) および最大カウントエラー (MCEF ビット) の両方のフラグがセットされます。対応する取得完了 (EOAIE ビット) や最大カウントエラー (MCEIE ビット) の割り込みイネーブルビットがセットされている場合は、割り込みも生成することができます。



## 19.3.7 サンプリングコンデンサ I/O および チャネル I/O のモード選択

タッチセンシングコントローラによる GPIO の制御を可能にするには、標準の GPIO レジスタおよび GPIOxAFR レジスタを介して、対応するオルタネート機能を有効にする必要があります。

TSC によって制御される GPIO モードは、TSC\_IOSCR および TSC\_IOCCR レジスタを使用して定義されます。

処理中の取得がない場合、タッチセンシングコントローラによって制御されるすべての I/O はデフォルト状態です。取得の処理中は、未使用 I/O (サンプリングコンデンサ I/O としても、チャネル I/O としても定義されない) のみがデフォルト状態です。TSC\_CR レジスタの IODEF ビットは、デフォルト状態にある I/O の設定を定義します。次の表にモードに応じた I/O の設定を示します。

**表 114. モードおよび IODEF ビットの値に応じた I/O の状態**

IODEF ビット	取得ステータス	未使用 I/O モード	電極 I/O モード	サンプリングコンデンサ I/O モード
0 (出力 プッシュプルロー)	なし	出力 プッシュプルロー	出力 プッシュプルロー	出力プッシュプル ロー
0 (出力 プッシュプルロー)	処理中	出力 プッシュプルロー	-	-
1 (入力 フローティング)	なし	入力フローティング	入力フローティング	入力フローティング
1 (入力 フローティング)	処理中	入力フローティング	-	-

### 未使用 I/O モード

未使用 I/O は、TSC ペリフェラルによって制御される GPIO に対応しますが、電極 I/O としても、サンプリングコンデンサ I/O としても定義されません。

### サンプリングコンデンサ I/O モード

TSC ペリフェラルによるサンプリングコンデンサ I/O の制御を可能にするには、まず対応する GPIO をオルタネート出力オープンドレインモードにセットし、次に TSC\_IOSCR レジスタ の対応する Gx\_IOy ビットをセットする必要があります。

1 つのアナログ I/O グループでは、一度に 1 つのサンプリングコンデンサのみを有効にしてください。

### チャネル I/O モード

TSC ペリフェラルによるチャネル I/O の制御を可能にするには、まず対応する GPIO をオルタネート出力プッシュプルモードにセットし、次に TSC\_IOCCR レジスタ の対応する Gx\_IOy ビットをセットする必要があります。

より高い等価電極表面が要求される、あるいは取得プロセスを高速化する近接検出では、同じアナログ I/O グループに属するいくつかのチャネルを有効にし、同時に取得することが可能です。

**注：** **取得フェーズの間、および TSC ペリフェラルのオルタネート機能が有効でない場合でも、TSC\_IOSCR または TSC\_IOCCR ビットがセットされたらすぐに、タッチセンシングコントローラによって、対応する GPIO アナログスイッチが自動的に制御されます。**



## 19.3.8 取得モード

タッチセンシングコントローラには 2 種類の取得モードがあります。

- 通常の取得モード: TSC\_CR レジスタの START ビットがセットされるとすぐに取得を開始します。
- 同期取得モード: TSC\_CR レジスタの START ビットをセットすることで取得が有効になりますが、取得の開始は、立ち下がり/立ち上がりエッジおよび SYNC 入力ピンのハイレベルが検出されたときのみです。このモードは、CPU 負荷を追加することなく、静電容量検出チャネルの取得を外部信号に同期させる場合に役立ちます。

TSC\_I0GCSR レジスタの GxE ビットで、どのアナログ I/O グループを有効 (対応するカウンタがカウントされる状態) にするかを指定します。無効なアナログ I/O グループの C<sub>S</sub> 電圧は監視されず、このグループは取得完了フラグのトリガに加わりません。ただし、無効なアナログ I/O グループにいくつかのチャネルが含まれる場合は、それらはパルスを出力します。

有効なアナログ I/O グループの C<sub>S</sub> 電圧が所定の閾値に達すると、TSC\_I0GCSR レジスタの対応する GxS ビットがセットされます。すべての有効なアナログ I/O グループの取得が完了した (すべての有効なアナログ I/O グループのすべての GxS ビットがセットされた) 時点で、TSC\_ISR レジスタの EOAF フラグがセットされます。TSC\_IER レジスタの EOAIIE ビットがセットされている場合、割り込みリクエストが生成されます。

最大カウントエラーが検出された場合、処理中の取得は停止し、TSC\_ISR レジスタの EOAF フラグと MCEF フラグの両方がセットされます。対応するビット (TSCIER レジスタの EOAIIE および MCEIE ビット) がセットされていれば、両方のイベントに対して割り込みリクエストを生成することができます。最大カウントエラーが検出された場合、有効なアナログ I/O グループの残りの GxS ビットはセットされませんので注意してください。

割り込みフラグをクリアするには、TSC\_ICR レジスタの対応する EOAIIC ビットと MCEIC ビットをセットする必要があります。

アナログ I/O グループのカウンタは、新しい取得が開始された時点でクリアされます。カウンタは、取得完了と同時に、対応するチャネルで生成された電荷移動サイクルの数に更新されます。

## 19.3.9 I/O ヒステリシスおよびアナログスイッチの制御

より高い柔軟性を提供するために、タッチセンシングコントローラを使用することで、各 Gx\_I0y のシュミットトリガヒステリシスおよびアナログスイッチを制御することもできます。タッチセンシングコントローラが有効であると仮定すれば、I/O 制御モードの状態 (標準の GPIO レジスタまたはその他のペリフェラルによって制御されている) にかかわらず、この制御は実行可能です。この制御は、異なる取得シーケンスを実行したり、他の目的のために実行する際に役立つことがあります。

システムの誤検出を防ぐ目的で、TSC\_I0HCR レジスタの対応する Gx\_I0y ビットをリセットすることによって、TSC によって制御される GPIO のシュミットトリガヒステリシスを無効にする必要があります。

## 19.4 TSC 低電力モード

表 115. 低電力モードが TSC に与える影響

モード	説明
SLEEP	影響なし。 TSC 割り込みによって、デバイスは SLEEP モードを終了します。
STOP	TSC レジスタは停止状態です。
STANDBY	STOP または STANDBY モードを終了するまで、TSC は動作を停止します。

## 19.5 TSC 割り込み

表 116. 割り込み制御ビット

割り込みイベント	有効化 制御ビット	イベント フラグ	フラッグ ビットの クリア	SLEEP モードの終了	STOPモード の終了	STANDBY モードの終了
取得完了	EOAIE	EOAIF	EOAIC	あり	なし	なし
最大カウントエラー	MCEIE	MCEIF	MCEIC	あり	なし	なし

## 19.6 TSC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

### 19.6.1 TSC 制御レジスタ (TSC\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTPH[3:0]				CTPL[3:0]				SSD[6:0]						SSE	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV[2:0]			IODEF	SYNC POL	AM	START	TSCE
r/w	r/w	r/w	r/w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:28 **CTPH[3:0]** : 電荷移動パルスハイ

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、電荷移動パルスのハイ状態 (C<sub>X</sub>の充電) の継続時間を定義します。

0000 : 1x t<sub>PGCLK</sub>

0001 : 2x t<sub>PGCLK</sub>

...

1111 : 16x t<sub>PGCLK</sub>

**注 :** 取得処理中はこれらのビットを変更しないでください。

ビット 27:24 **CTPL[3:0]** : 電荷移動パルスロー

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、電荷移動パルスのロー状態 (C<sub>X</sub> から C<sub>S</sub>への電荷の移動) の継続時間を定義します。

0000 : 1x t<sub>PGCLK</sub>

0001 : 2x t<sub>PGCLK</sub>

...

1111 : 16x t<sub>PGCLK</sub>

**注 :** 取得処理中はこれらのビットを変更しないでください。

**注 :** 一部の設定は禁止されています。詳細については、[セクション 19.3.4 : 電荷移動取得シーケンス](#) を参照してください。

ビット 23:17 **SSD[6:0]** : スペクトル拡散偏差

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、SSCLK クロックの周期の変数を電荷移動パルスのハイ状態に追加する際のスペクトル拡散偏差を定義します。

0000000 : 1x t<sub>SSCLK</sub>

0000001 : 2x t<sub>SSCLK</sub>

...

1111111 : 128x t<sub>SSCLK</sub>

**注 :** 取得処理中はこれらのビットを変更しないでください。

**ビット 16 SSE** : スペクトル拡散イネーブル

このビットは、スペクトル拡散機能を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : スペクトル拡散機能は無効です。
- 1 : スペクトル拡散機能は有効です。

**注:** 取得処理中はこのビットを変更しないでください。

**ビット 15 SSPSC** : スペクトル拡散プリスケアラ

このビットは、ソフトウェアによってセット/クリアされます。このビットは、スペクトル拡散クロック (SSCLK) の生成に使用される AHB クロック分周器を選択します。

- 0 :  $f_{HCLK}$
- 1 :  $f_{HCLK} / 2$

**注:** 取得処理中はこのビットを変更しないでください。

**ビット 14:12 PGPSC[2:0]** : パルス発生回路プリスケアラ

これらのビットはソフトウェアによってセット/クリアされます。これらのビットは、パルス発生回路クロック (PGCLK) の生成に使用される AHB クロック分周器を選択します。

- 000 :  $f_{HCLK}$
- 001 :  $f_{HCLK} / 2$
- 010 :  $f_{HCLK} / 4$
- 011 :  $f_{HCLK} / 8$
- 100 :  $f_{HCLK} / 16$
- 101 :  $f_{HCLK} / 32$
- 110 :  $f_{HCLK} / 64$
- 111 :  $f_{HCLK} / 128$

**注:** 取得処理中はこれらのビットを変更しないでください。

**注:** 一部の設定は禁止されています。詳細については、[セクション 19.3.4 : 電荷移動取得シーケンス](#)を参照してください。

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

**ビット 7:5 MCV[2:0]** : 最大カウント値

これらのビットは、ソフトウェアによってセット/クリアされます。これらのビットは、最大カウントエラーが生成される前に生成され得る電荷移動パルスの最大数を定義します。

- 000 : 255
- 001 : 511
- 010 : 1023
- 011 : 2047
- 100 : 4095
- 101 : 8191
- 110 : 16383
- 111 : 予約済み

**注:** 取得処理中はこれらのビットを変更しないでください。

**ビット 4 IODEF** : I/O デフォルトモード

このビットは、ソフトウェアによってセット/クリアされます。このビットは、処理中の取得がない場合に、すべての TSC I/O の設定を定義します。処理中の取得がある場合、このビットは、すべての未使用 I/O (サンプリングコンデンサ I/O やチャネル I/O として定義されていない) の設定を定義します。

- 0 : I/O は強制的に出カプッシュプルローにされます。
- 1 : I/O は入力フローティング状態です。

**注:** 取得処理中はこのビットを変更しないでください。

**ビット 3 SYNCPOL** : 同期ピンの極性

このビットは、同期入力ピンの極性を選択するために、ソフトウェアによってセット/クリアされます。

- 0 : 立ち下がりがエッジのみ
- 1 : 立ち上がりエッジおよびハイレベル



**ビット 2 AM** : 取得モード

このビットは、取得モードを選択するために、ソフトウェアによってセット/クリアされます。

0 : 通常の取得モード (START ビットがセットされた直後に取得を開始)

1 : 同期取得モード (START ビットがセットされ、かつ選択された信号が SYNC 入力ピンで検出された場合に、取得を開始)

**注 :** 取得処理中はこのビットを変更しないでください。

**ビット 1 START** : 新しい取得を開始します。

このビットは、新しい取得を開始するために、ソフトウェアによってセットされます。このビットは、取得が完了した直後にハードウェアによって、または処理中の取得をキャンセルするためにソフトウェアによってクリアされます。

0 : 取得は開始されていません。

1 : 新しい取得を開始します。

**ビット 0 TSCE** : タッチセンシングコントローライネーブル

このビットは、タッチセンシングコントローラを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : タッチセンシングコントローラは無効です。

1 : タッチセンシングコントローラは有効です。

**注 :** タッチセンシングコントローラが無効の場合、TSC レジスタの設定は無効です。

## 19.6.2 TSC 割り込み有効レジスタ (TSC\_IER)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 MCEIE** : 最大カウントエラー割り込みイネーブル

このビットは、最大カウントエラーの割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : 最大カウントエラー割り込みは無効です。

1 : 最大カウントエラー割り込みは有効です。

**ビット 0 EOAIE** : 取得完了割り込みイネーブル

このビットは、取得完了割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : 取得完了割り込みは無効です。

1 : 取得完了割り込みは有効です。

## 19.6.3 TSC 割り込みクリアレジスタ (TSC\_ICR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **MCEIC** : 最大カウントエラー割り込みクリア

このビットは、最大カウントエラーフラグをクリアするためにソフトウェアによってセットされ、フラグがリセットされた時点でハードウェアによってクリアされます。“0”を書き込んでも、ビットの値は変化しません。

0 : 影響なし。

1 : TSC\_ISR レジスタの対応する MCEF をクリアします。

ビット 0 **EOAIC** : 取得完了割り込みクリア

このビットは、取得完了フラグをクリアするためにソフトウェアによってセットされ、フラグがリセットされた時点でハードウェアによってクリアされます。“0”を書き込んでも、ビットの値は変化しません。

0 : 影響なし。

1 : TSC\_ISR レジスタの対応する EOAF をクリアします。

## 19.6.4 TSC 割り込みステータスレジスタ (TSC\_ISR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **MCEF** : 最大カウントエラーフラグ

このビットは、アナログ I/O グループのカウンタが指定された最大カウント値に達すると、ハードウェアによってセットされます。このビットは、ソフトウェアで TSC\_ICR レジスタの MCEIC ビットに“1”を書き込むことによってクリアされます。

- 0 : 最大カウントエラー (MCE) は検出されていません。
- 1 : 最大カウントエラー (MCE) が検出されました。

ビット 0 **EOAF** : 取得完了フラグ

このビットは、すべての有効なグループの取得が完了した (すべての有効なアナログ I/O グループのすべての GxS ビットがセットされた、または最大カウントエラーが検出された) 時点で、ハードウェアによってセットされます。このビットは、ソフトウェアで TSC\_ICR レジスタの EOAIIC ビットに“1”を書き込むことによってクリアされます。

- 0 : 取得は処理中または開始されていません。
- 1 : 取得は完了しました。

## 19.6.5 TSC I/O ヒステリシス制御レジスタ (TSC\_IOHCR)

アドレスオフセット : 0x10

リセット値 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **Gx\_IOy** : Gx\_IOy シュミットトリガヒステリシスモード

これらのビットは、Gx\_IOy シュミットトリガヒステリシスを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : Gx\_IOy シュミットトリガヒステリシスは無効です。
- 1 : Gx\_IOy シュミットトリガヒステリシスは有効です。

**注 :** これらのビットは、I/O 制御モードの状態にかかわらず (標準の GPIO レジスタによって制御されていても)、I/O シュミットトリガヒステリシスを制御します。



## 19.6.6 TSC I/O アナログスイッチ制御レジスタ (TSC\_IOASCR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **Gx\_IOy** : Gx\_IOy アナログスイッチイネーブル

これらのビットは、Gx\_IOy アナログスイッチを有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : Gx\_IOy アナログスイッチは無効 (開) です。

1 : Gx\_IOy アナログスイッチは有効 (閉) です。

**注 :** これらのビットは、I/O 制御モードの状態にかかわらず (標準の GPIO レジスタによって制御されていても)、I/O アナログスイッチを制御します。

## 19.6.7 TSC I/O サンプリング制御レジスタ (TSC\_IOSCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **Gx\_IOy** : Gx\_IOy サンプリングモード

これらのビットは、Gx\_IOy をサンプリングコンデンサ I/O として設定するために、ソフトウェアによってセット/クリアされます。1つのアナログ I/O グループでは、1つの I/O だけをサンプリングコンデンサとして定義してください。

0 : Gx\_IOy は未使用です。

1 : Gx\_IOy はサンプリングコンデンサとして使用されます。

**注 :** 取得処理中はこれらのビットを変更しないでください。

取得フェーズの間、および TSC ベリフェラルのオルタネート機能が有効でない場合でも、TSC\_IOSCR ビットがセットされたらすぐに、タッチセンシングコントローラによって、対応する GPIO アナログスイッチが自動的に制御されます。

## 19.6.8 TSC I/O チャンネル制御レジスタ (TSC\_IOCCTRSC\_IOCOCR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **Gx\_IOy** : Gx\_IOy チャンネルモード

これらのビットは、Gx\_IOy をチャンネル I/O として設定するために、ソフトウェアによってセット/クリアされます。

0 : Gx\_IOy は未使用です。

1 : Gx\_IOy はチャンネルとして使用されます。

**注 :** 取得処理中はこれらのビットを変更しないでください。

取得フェーズの間、およびTSC パリフェラルのオルタネート機能が有効でない場合でも、TSC\_IOCOCR ビットがセットされたらすぐに、タッチセンシングコントローラによって、対応するGPIO アナログスイッチが自動的に制御されます。

## 19.6.9 TSC I/O グループ制御ステータスレジスタ (TSC\_IIOGCSR)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8S	G7S	G6S	G5S	G4S	G3S	G2S	G1S
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8E	G7E	G6E	G5E	G4E	G3E	G2E	G1E
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **GxS** : アナログ I/O グループ x ステータス

これらのビットは、対応する有効なアナログ I/O グループ x で取得が完了した時点で、ハードウェアによってセットされます。これらのビットは、新しい取得が開始した時点で、ハードウェアによってクリアされます。

0 : アナログ I/O グループ x での取得は、処理中または開始されていません。

1 : アナログ I/O グループ x での取得は完了しました。

**注 :** 最大カウンタエラーが検出された場合、有効なアナログ I/O グループの残りの GxS ビットはセットされません。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **GxE** : アナログ I/O グループ x イネーブル

これらのビットは、対応するアナログ I/O グループ x での取得を有効 (カウンタがカウントされる状態) にするために、ソフトウェアによってセット/クリアされます。

0 : アナログ I/O グループ x での取得は無効です。

1 : アナログ I/O グループ x での取得は有効です。

## 19.6.10 TSC I/O グループ x カウンタレジスタ (TSC\_IIOGxCR) (x = 1~8)

アドレスオフセット :  $0x30 + 0x04 \times$  アナログ I/O グループ番号

リセット値 : 0x0000 0000

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

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **CNT[13:0]** : カウンタ値

これらのビットは、アナログ I/O グループ x の取得を完了させる ( $C_S$  の電圧が閾値に達する) ために、そこで生成された電荷移動サイクルの数を表します。

## 19.6.11 TSC レジスタマップ

表 117. TSC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x0000	TSC_CR	CTPH[3:0]				CTPL[3:0]				SSD[6:0]						SSE	SSPSC		PGPSC[2:0]		Res.	Res.	Res.	Res.	MCV [2:0]		IODEF	SYNCPOL	AM	START	TSCE					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0004	TSC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE		
	リセット値																																0	0		
0x0008	TSC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC		
	リセット値																																0	0		
0x000C	TSC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIF	EOAIF		
	リセット値																																0	0		
0x0010	TSC_IOHCR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1			
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x0014	予約済み																																			
0x0018	TSC_IOASCR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x001C	予約済み																																			
0x0020	TSC_IOSCR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0024	予約済み																																			
0x0028	TSC_IOCRR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x002C	予約済み																																			
0x0030	TSC_IQGCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																			
0x0034	TSC_IQG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]															
	リセット値																				0															
0x0038	TSC_IQG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]															
	リセット値																				0															

表 117. TSC レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x003C	TSC_I0G3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	TSC_I0G4CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	TSC_I0G5CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	TSC_I0G6CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	TSC_I0G7CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	TSC_I0G8CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



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

## 20 高機能制御タイマ (TIM1/TIM8/TIM20)

### 20.1 TIM1/TIM8/TIM20 の概要

高機能制御タイマ (TIM1/TIM8/TIM20) は、プログラム可能なプリスケラによって駆動される 16 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM、デッドタイムを挿入した相補 PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

高機能制御タイマ (TIM1/TIM8/TIM20) と汎用タイマ (TIMx) は、互いに独立しており、リソースを共有しません。これらのタイマは、[セクション 20.3.25: タイマの同期](#)に示すように、相互に同期させることができます。

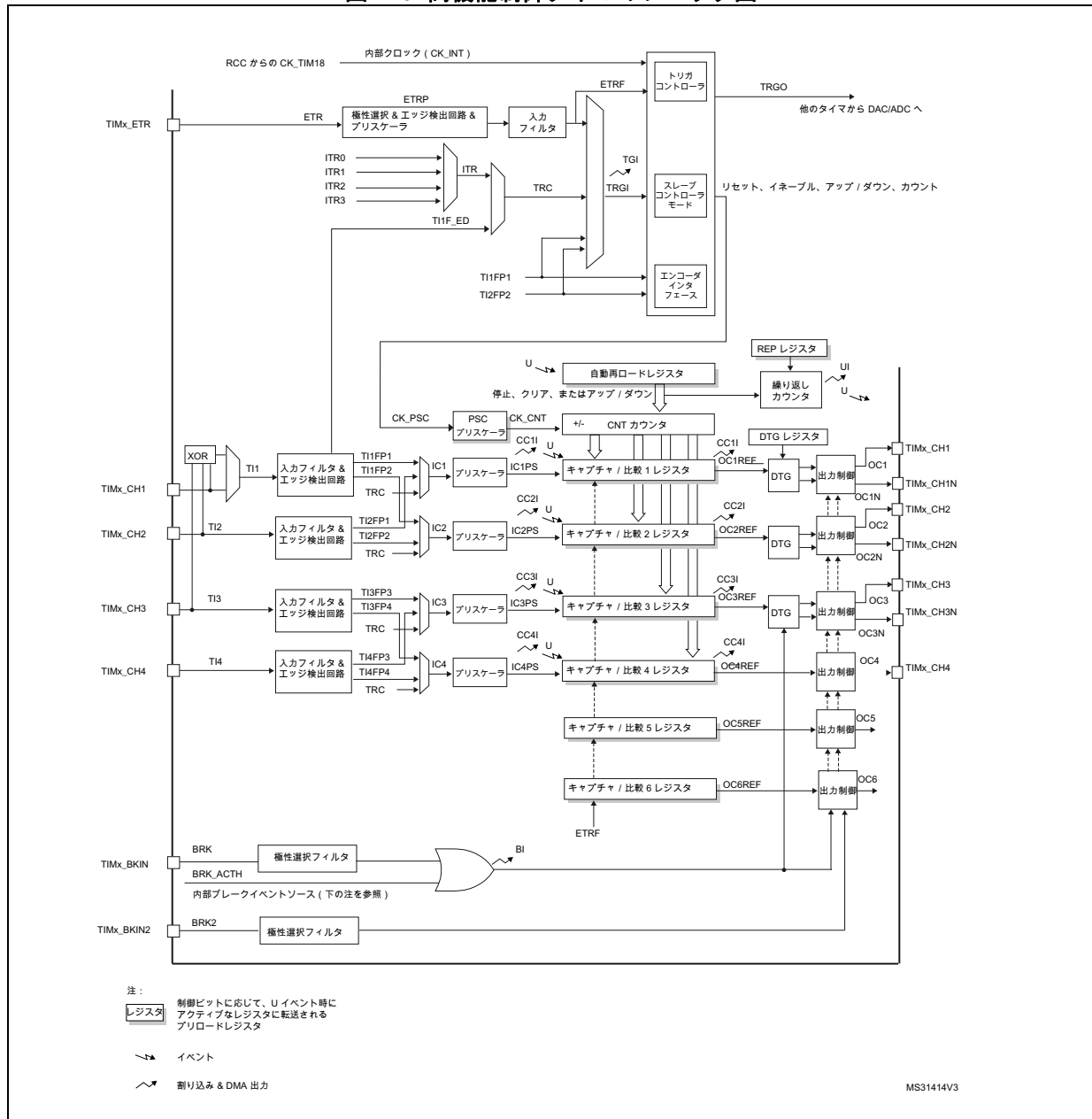
**注:** *TIM8 は STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスで使用できます。TIM20 は、STM32F303xD/E および STM32F398xE デバイスでのみ使用できます。*

### 20.2 TIM1/TIM8/TIM20 の主な特長

TIM1/TIM8/TIM20 タイマの主な機能は、次のとおりです。

- 16 ビットのアップカウンタ、ダウンカウンタ、アップ/ダウン自動再ロードカウンタ。
- 16 ビットのプログラム可能なプリスケラ。カウンタクロック周波数を 1 から 65536 の間で分周でき、分周比の動作中の変更も可能。
- 次の機能を持つ、最大 6 つの独立チャンネル。
  - 入力キャプチャ (ただしチャンネル 5 および 6)
  - 出力比較
  - PWM 生成 (エッジアラインモードとセンターアラインモード)
  - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をユーザが選択可能な安全な設定にする 2 つのブレイク入力。
- 以下のイベント時の割り込み/DMA 生成。
  - 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
  - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
  - 入力キャプチャ
  - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 140. 高機能制御タイマのブロック図



- 内部ブレークイベントソースは次のいずれかです。
  - CSS によって生成されたクロック障害イベント。CSS の詳細については、[セクション 9.2.7: クロックセキュリティシステム \(CSS\)](#) を参照してください。
  - PVD 出力
  - SRAM パリティエラー信号
  - Cortex-M4<sup>®</sup>F LOCKUP (ハードフォルト) 出力
  - COMP1/2/3/5/6 出力

## 20.3 TIM1/TIM8/TIM20機能詳細

### 20.3.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC)
- 自動再ロードレジスタ (TIMx\_ARR)
- 繰り返しカウンタレジスタ (TIMx\_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー (またはダウンカウント時はアンダーフロー) に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときのみ、プリスケアラ出力 CK\_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx\_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 141](#) と [図 142](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。



図 141. プリスケーラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

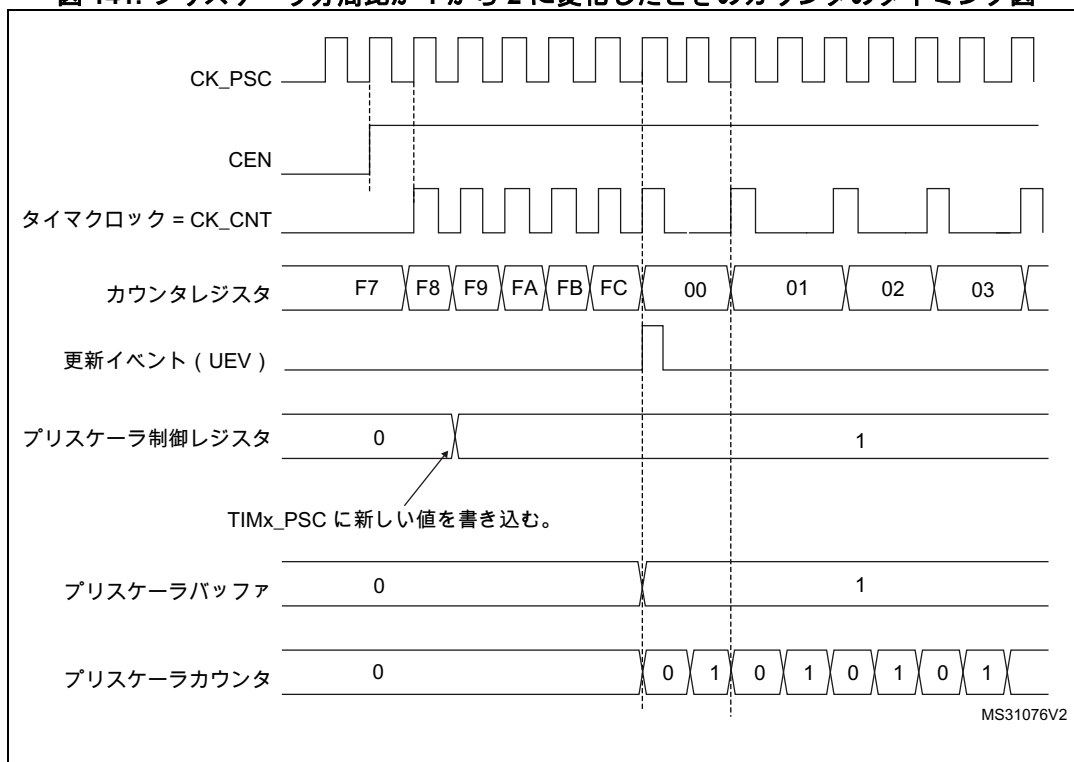
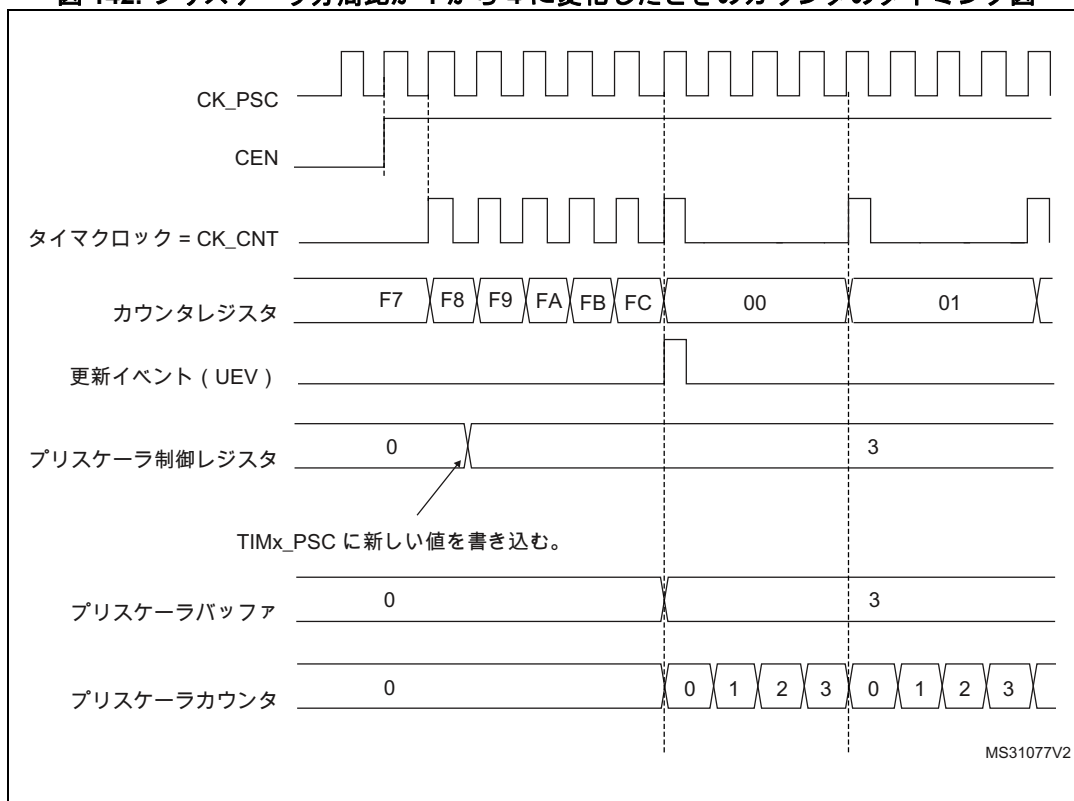


図 142. プリスケーラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 20.3.2 カウンタモード

### アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx\_RCR) + 1 までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 143. 内部クロック分周比が 1 の場合のカウンタのタイミング図

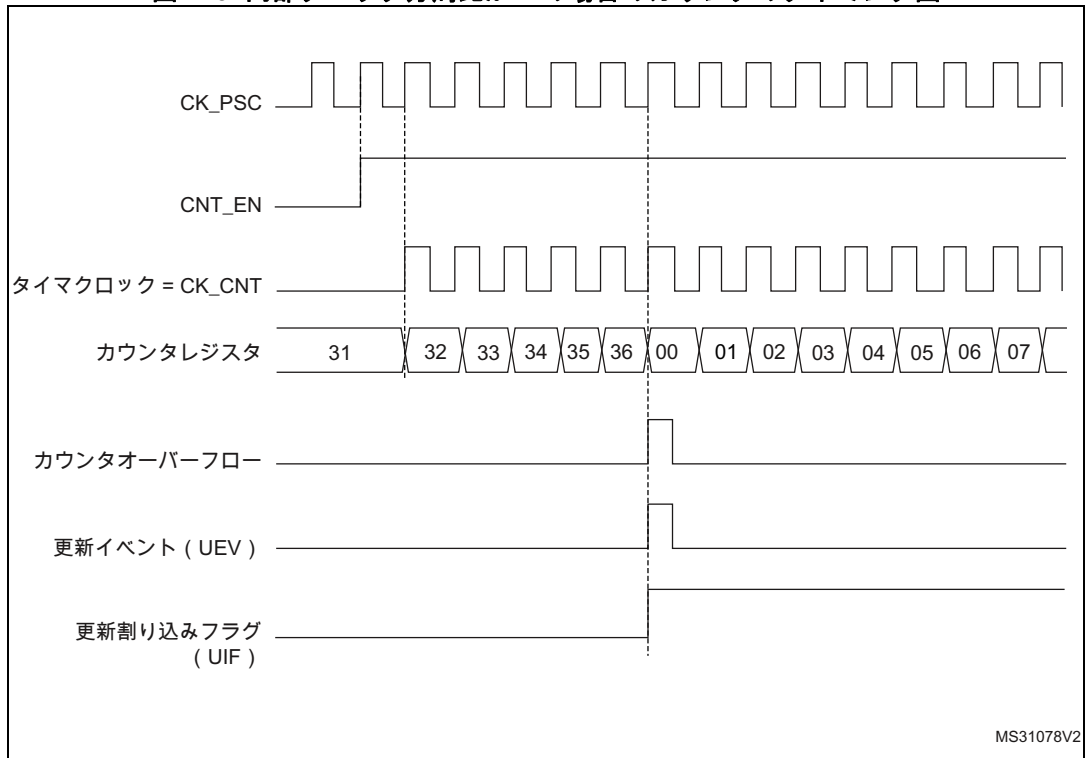


図 144. 内部クロック分周比が 2 の場合のカウンタのタイミング図

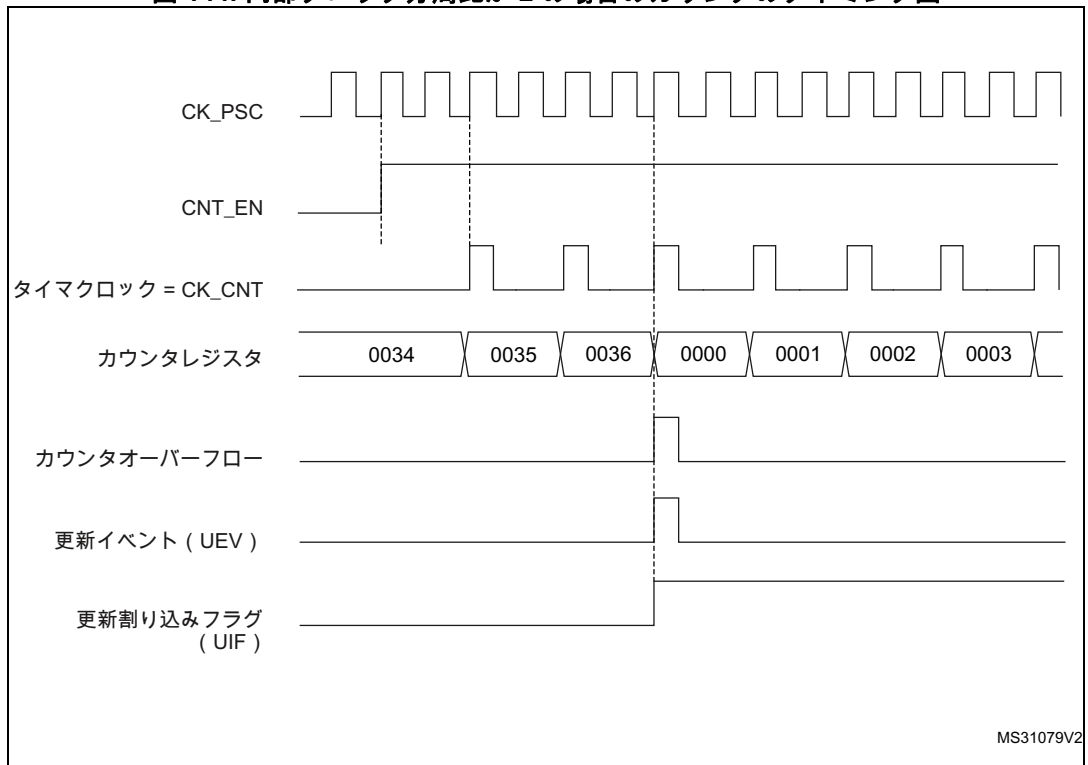


図 145. 内部クロック分周比が 4 の場合のカウンタのタイミング図

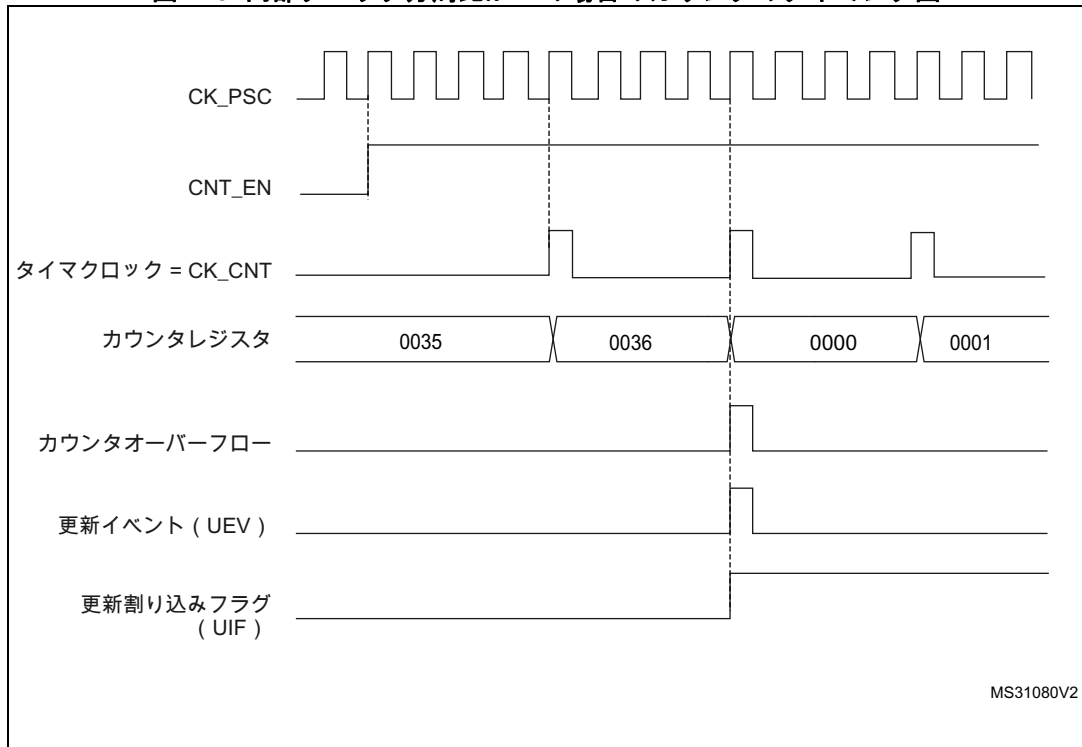


図 146. 内部クロック分周比が N の場合のカウンタのタイミング図

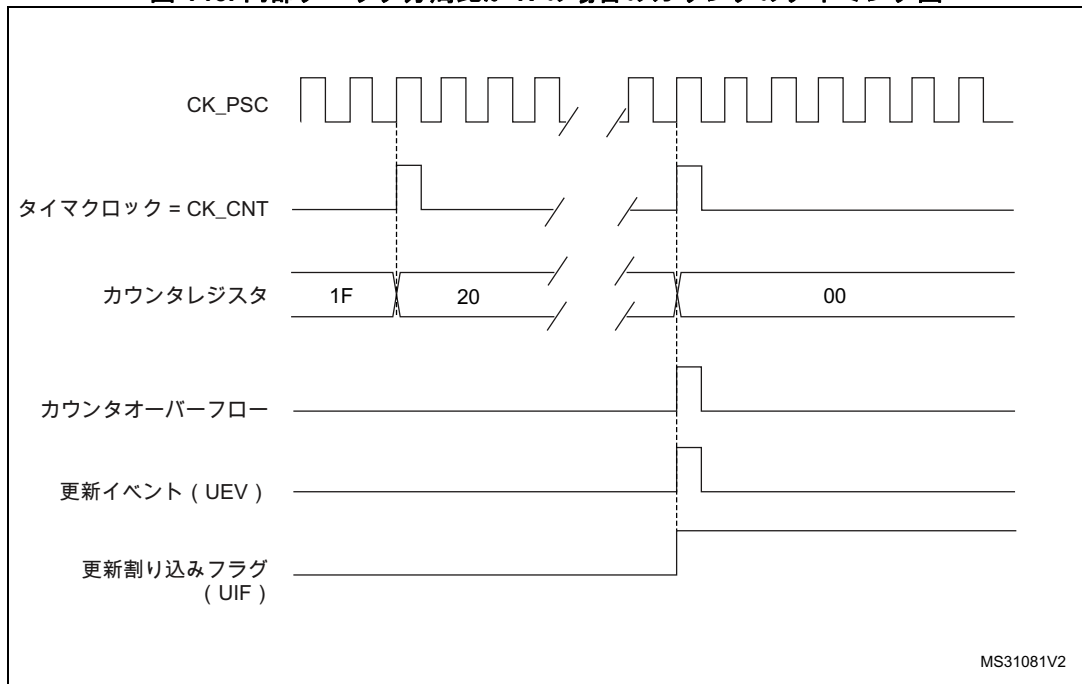


図 147. ARPE = 0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

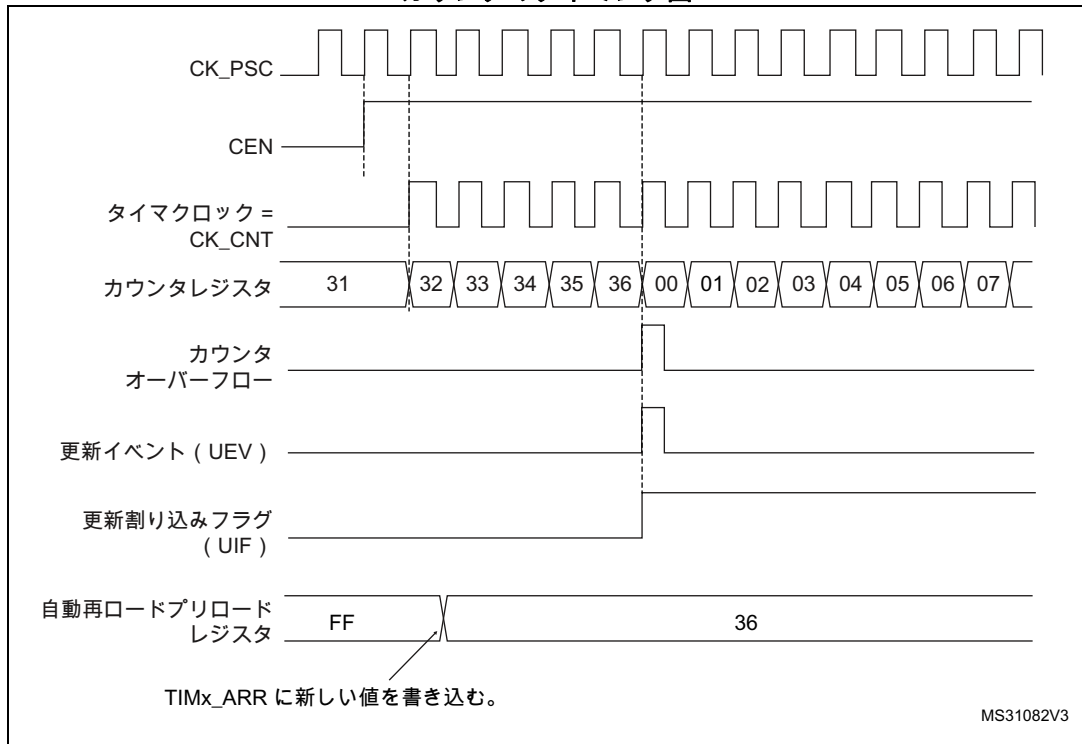
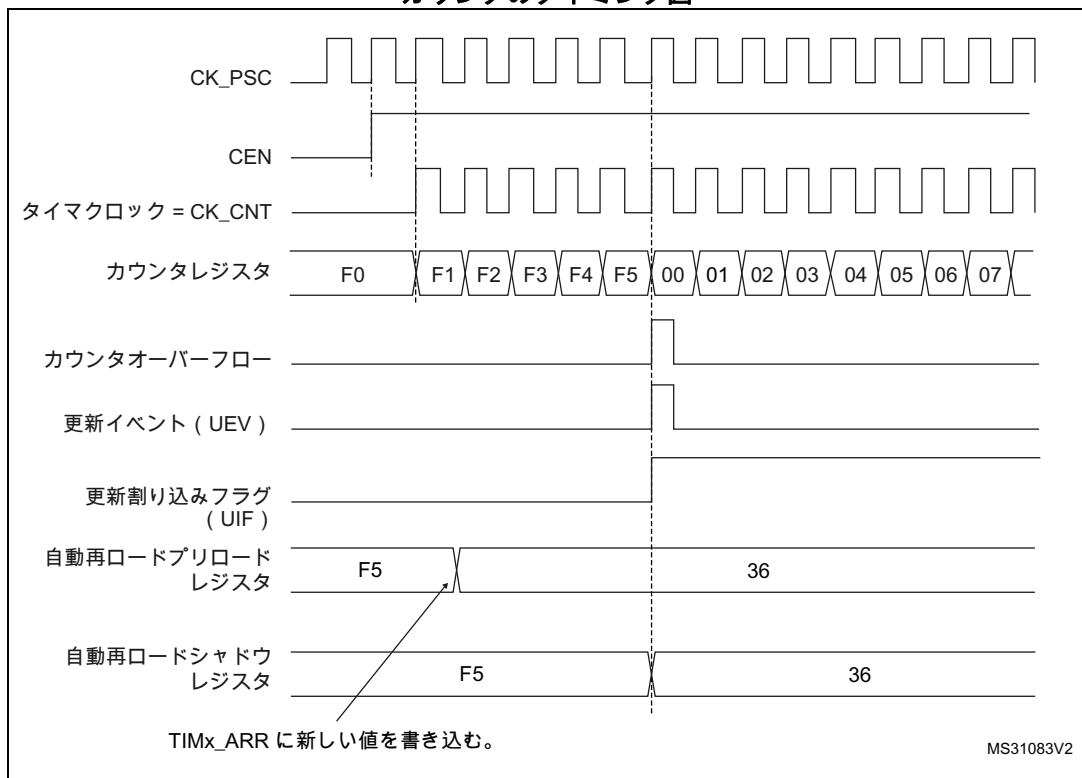


図 148. ARPE = 1 (TIMx\_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



## ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx\_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx\_RCR) + 1 までダウンカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのアンダーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 149. 内部クロック分周比が 1 の場合のカウンタのタイミング図

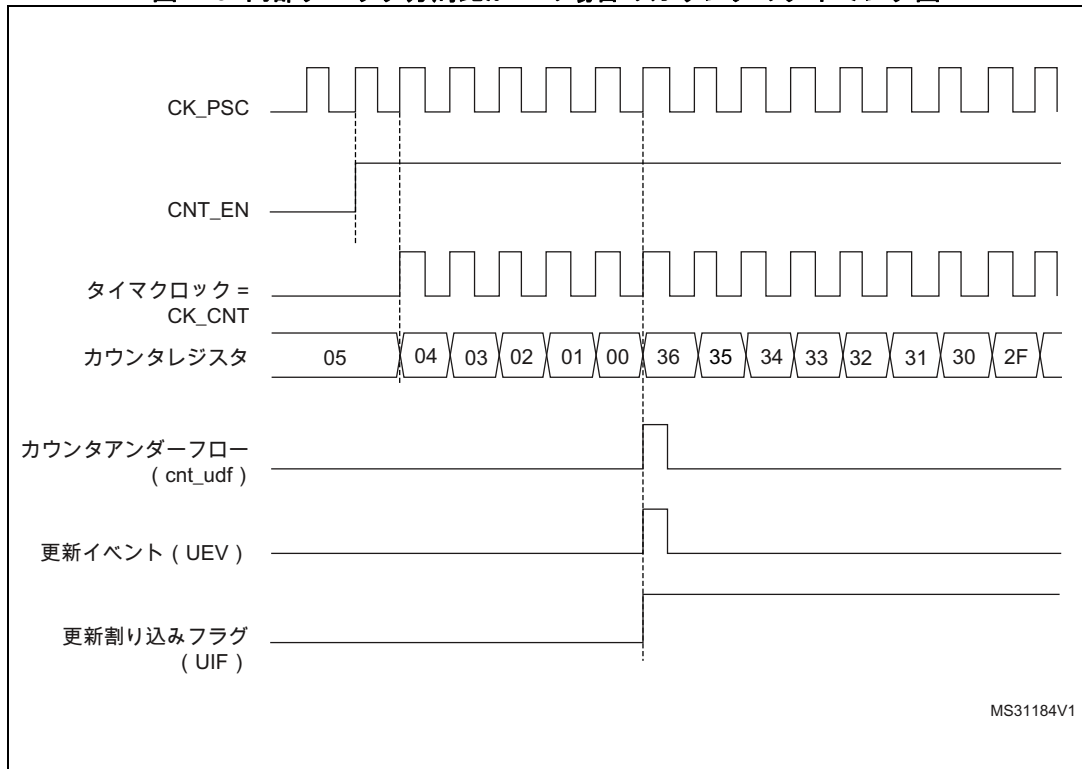


図 150. 内部クロック分周比が 2 の場合のカウンタのタイミング図

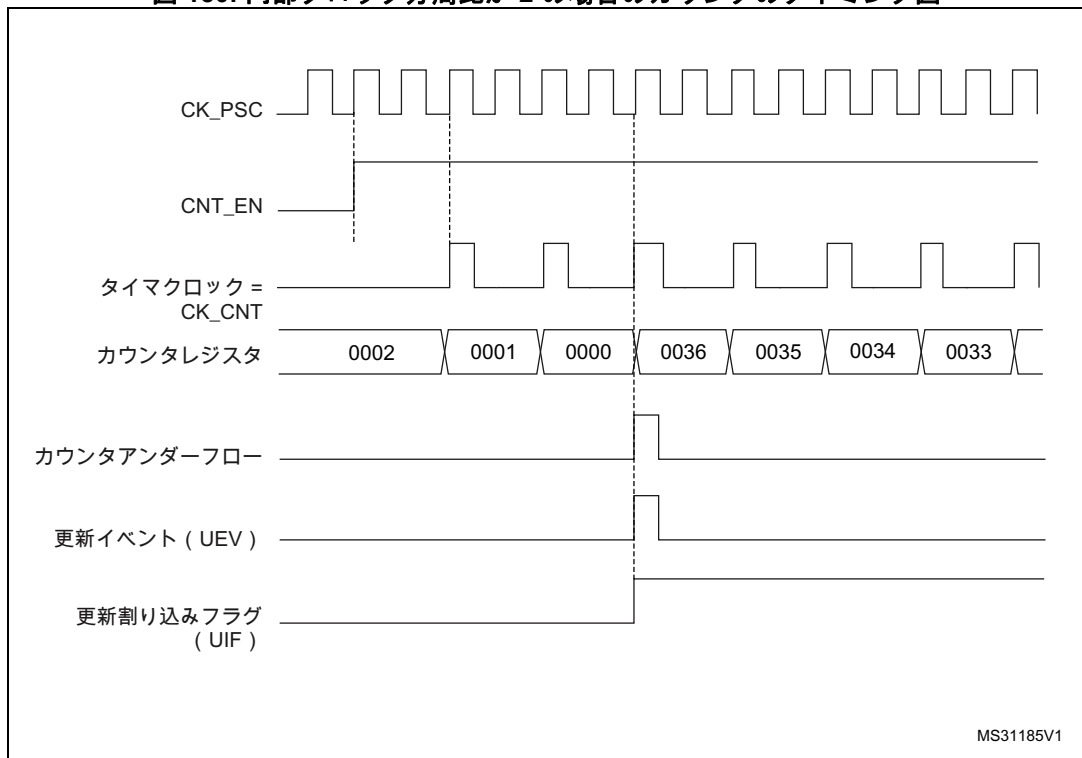


図 151. 内部クロック分周比が 4 の場合のカウンタのタイミング図

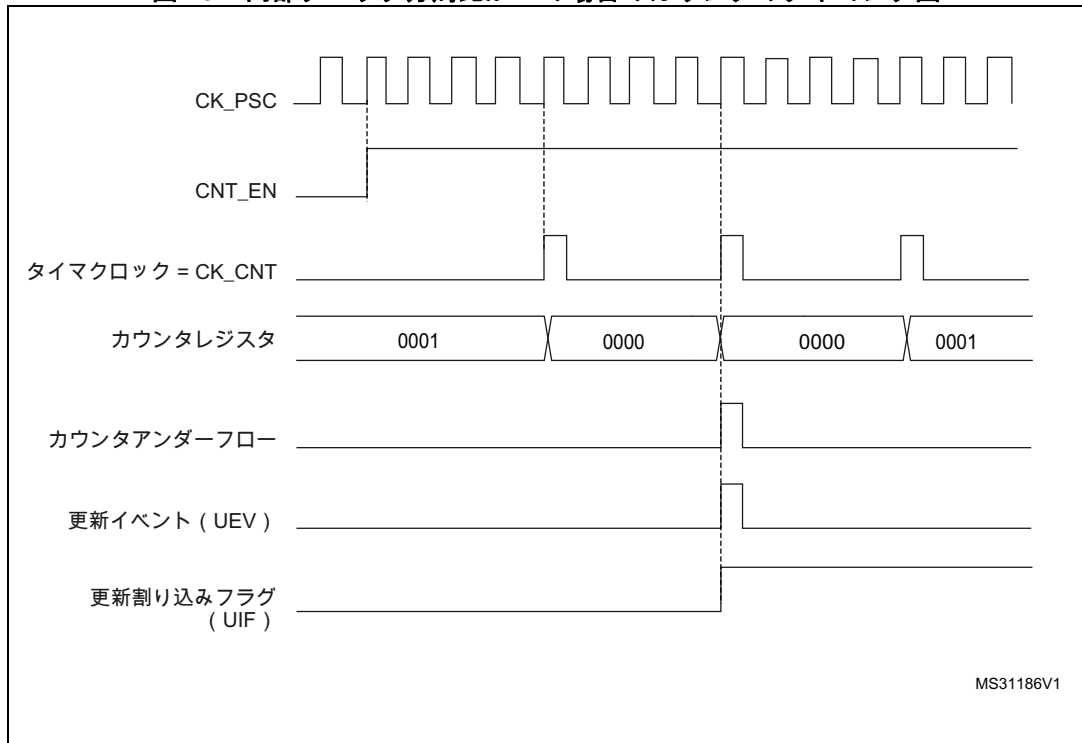


図 152. 内部クロック分周比が N の場合のカウンタのタイミング図

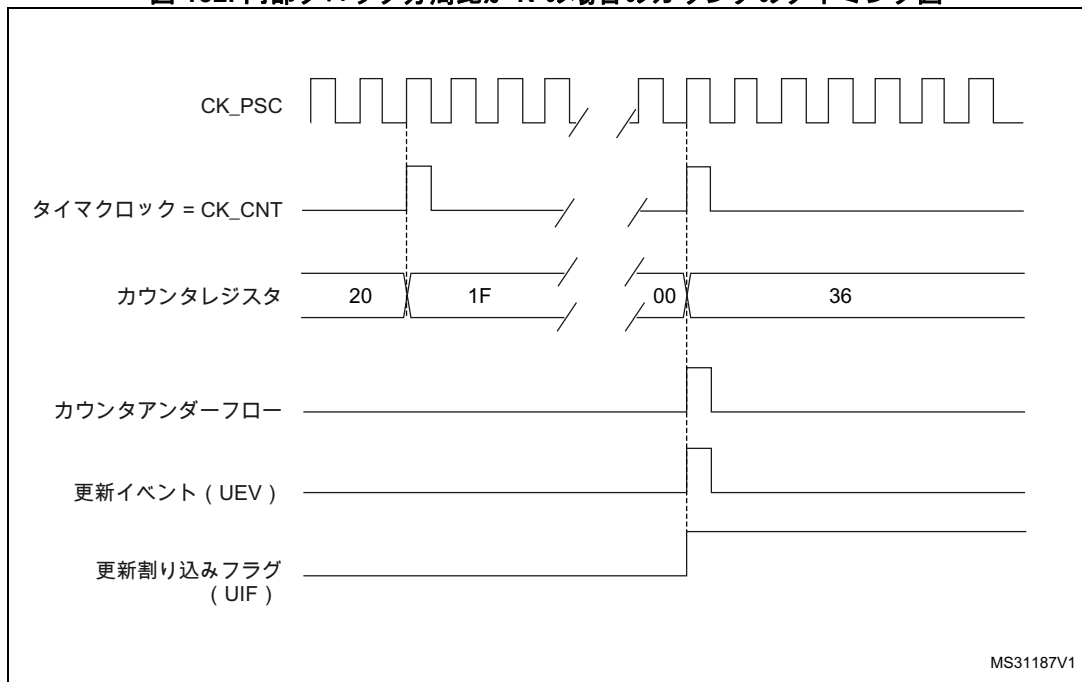
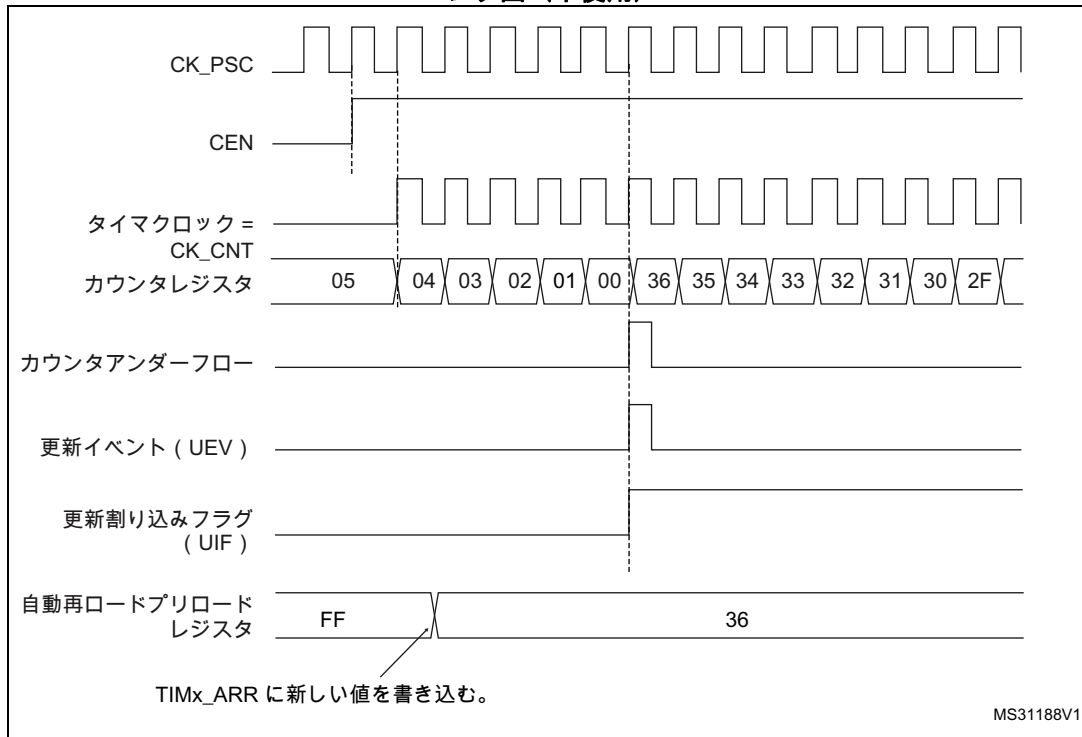




図 153. 繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図 (不使用)



## センターアラインモード (アップ/ダウンカウンタ)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センターアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS=11) にセットされます。

このモードでは、TIMx\_CR1 レジスタの方向ビット (DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合、UG ビットをセットすると UEV 更新イベントが生成されますが、UIF フラグはセットされません (した

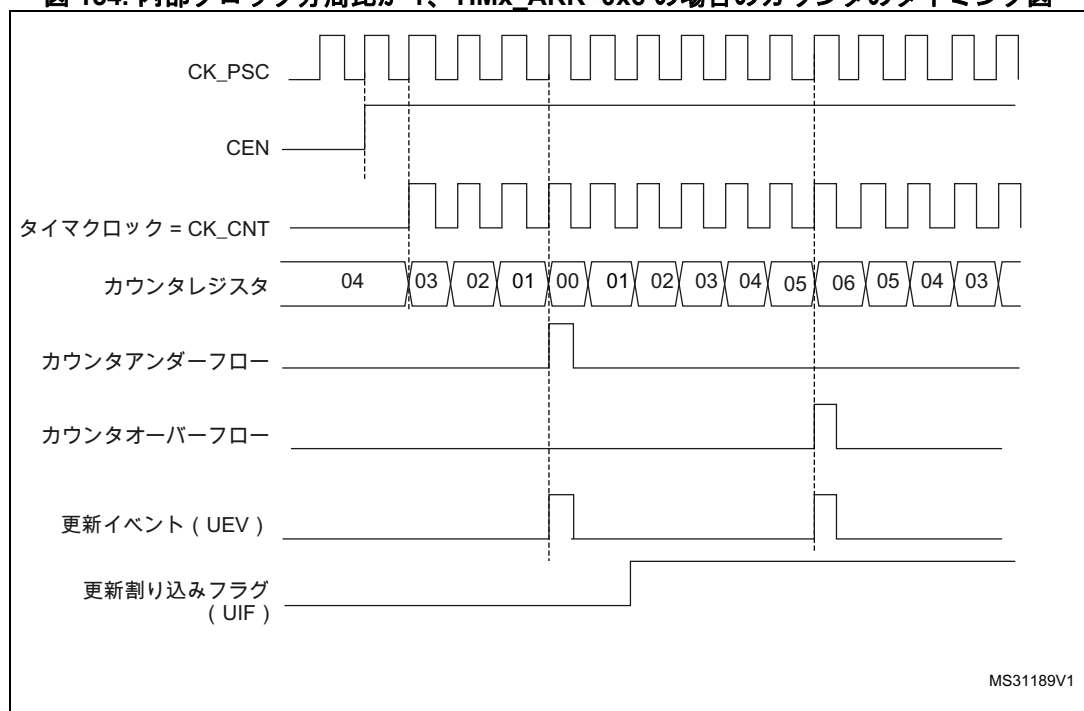
がって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 154. 内部クロック分周比が 1、TIMx\_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 20.4: TIM1/TIM8/TIM20 レジスタ](#) を参照)。

図 155. 内部クロック分周比が 2 の場合のカウンタのタイミング図

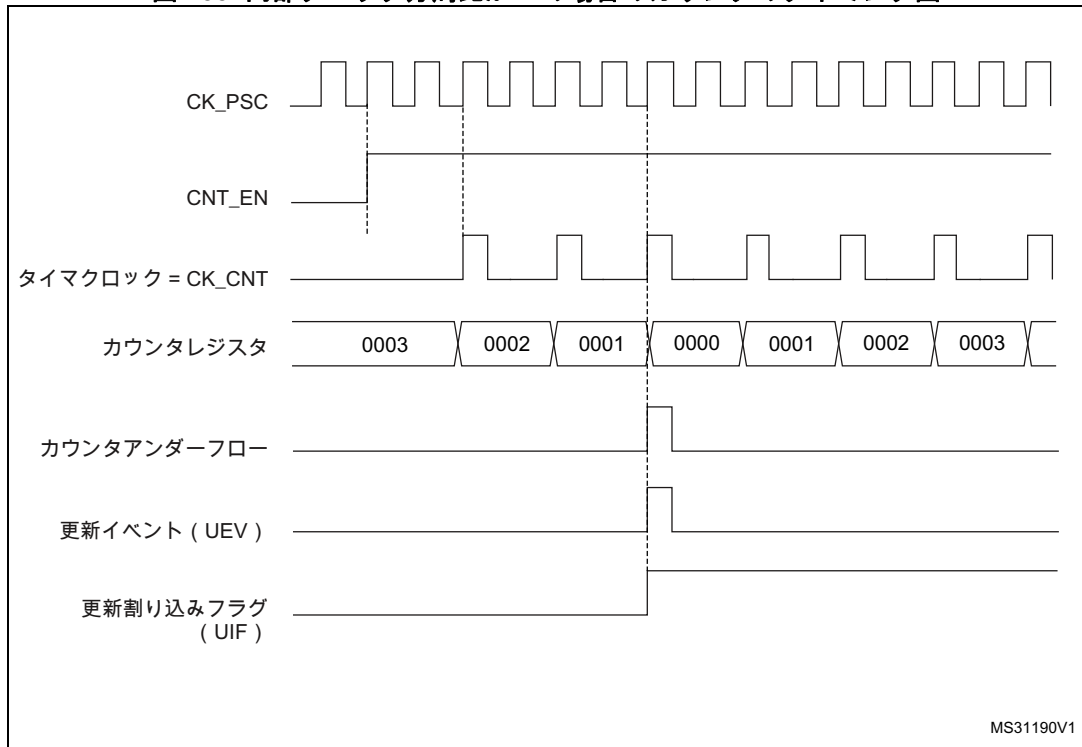


図 156. 内部クロック分周比が 4、TIMx\_ARR=0x36 の場合のカウンタのタイミング図

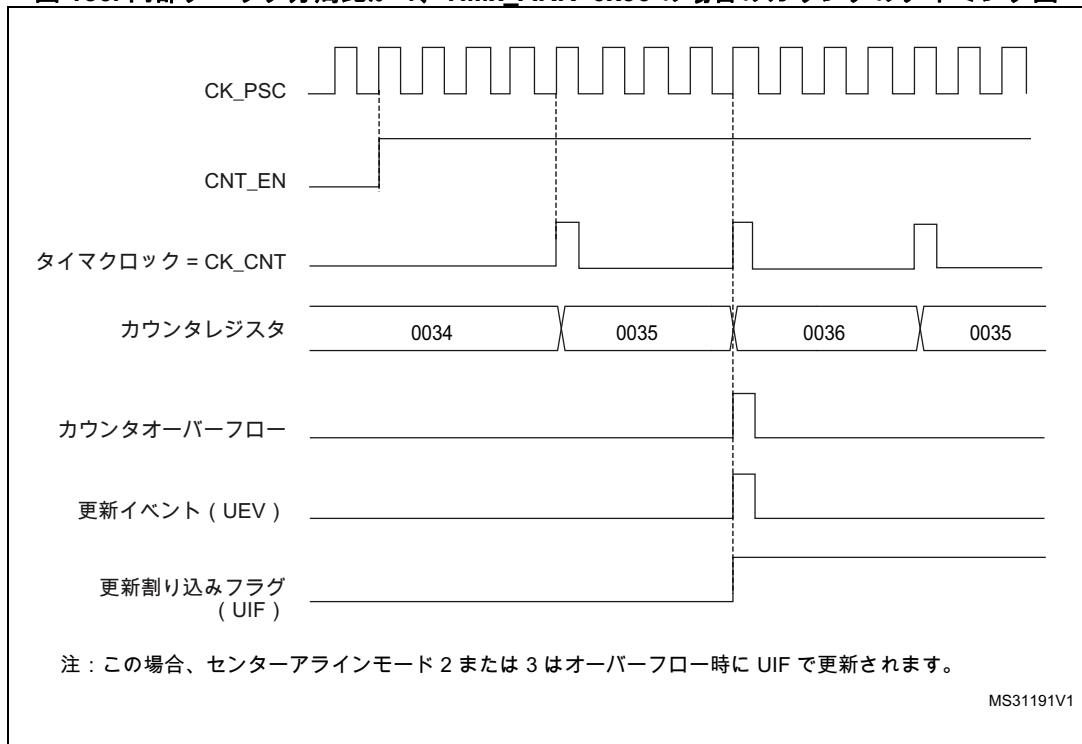


図 157. 内部クロック分周比が N の場合のカウンタのタイミング図

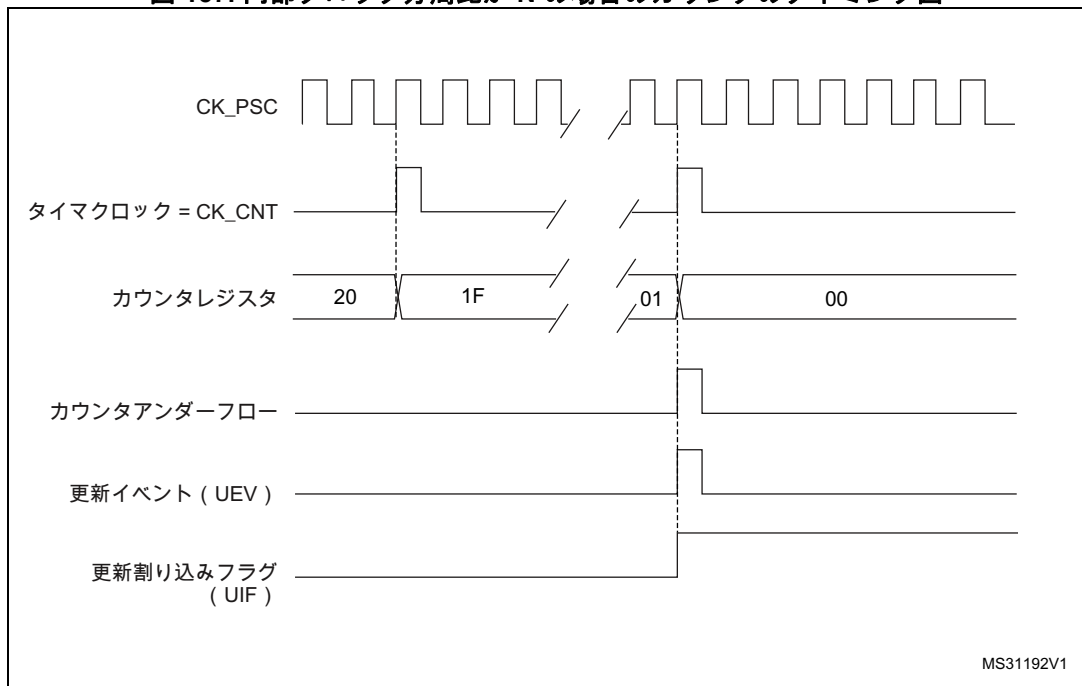


図 158. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図

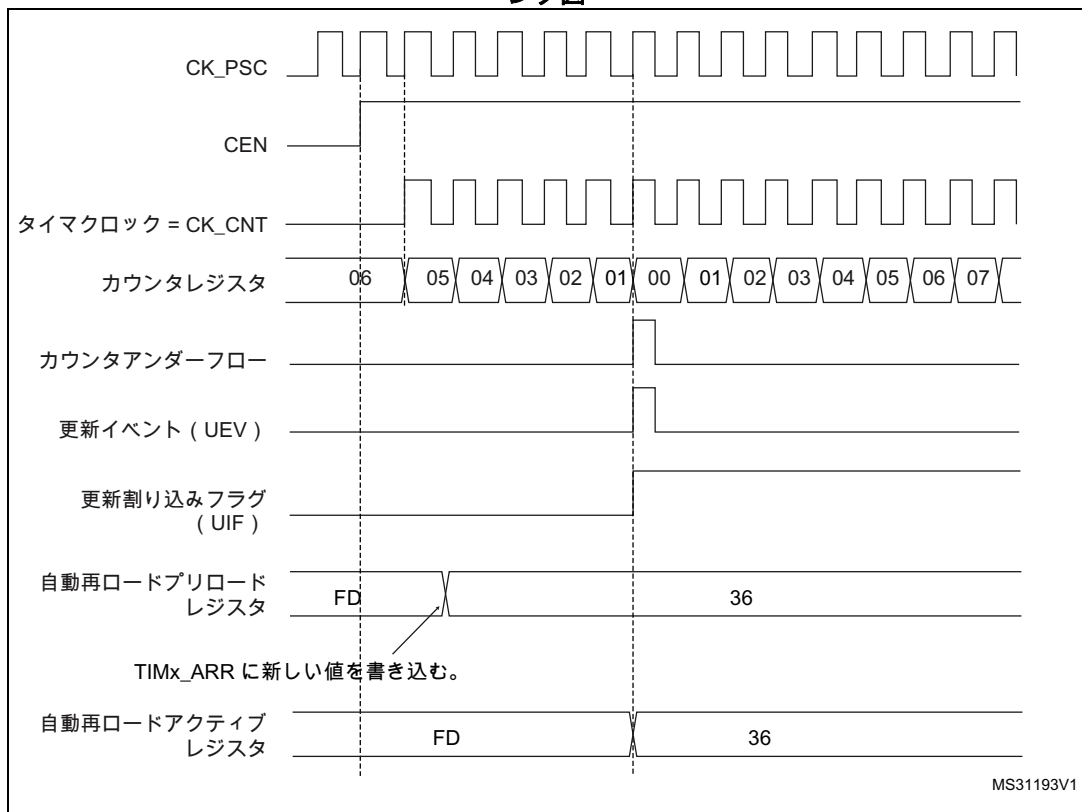
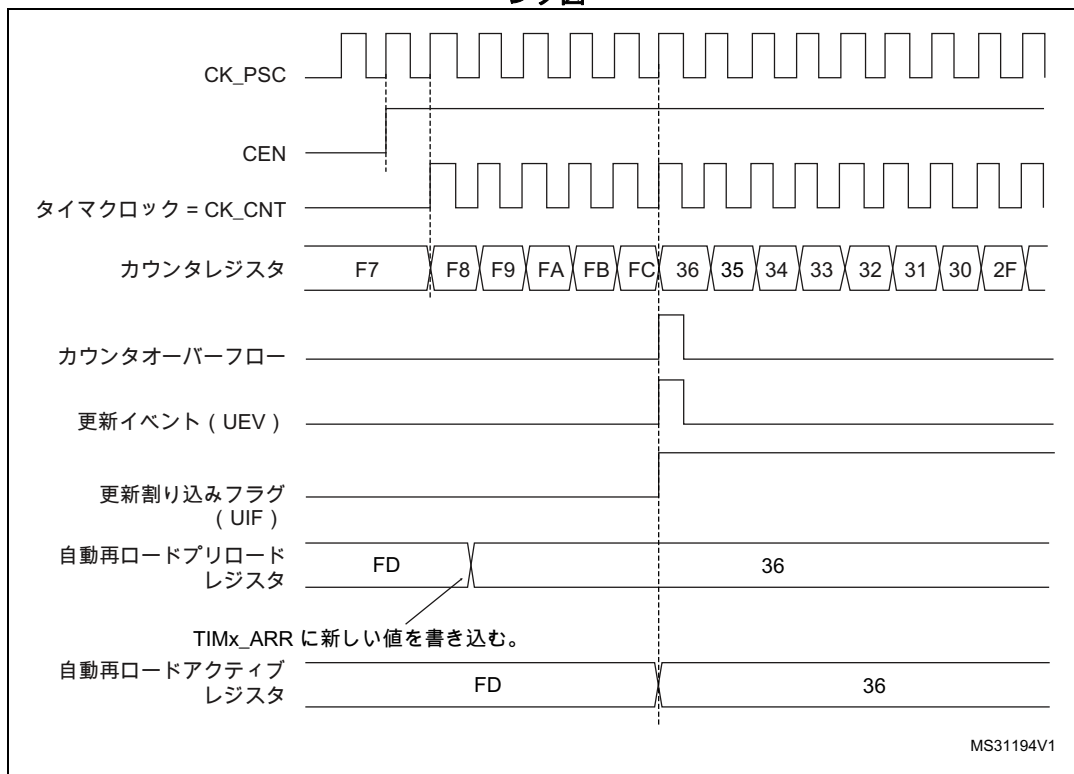


図 159. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



## 20.3.3 繰り返しカウンタ

セクション 20.3.1: タイムベースユニットに、カウンタオーバーフロー/アンダーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰り返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx\_RCR 繰り返しカウンタレジスタの値を N とすると、N+1 回目のカウンタオーバーフローまたはアンダーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx\_ARR 自動再ロードレジスタ、TIMx\_PSC プリスケアラレジスタ、比較モードの TIMx\_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

繰り返しカウンタは、次の場合にデクリメントします。

- アップカウントモードで、カウンタオーバーフローごと
- ダウンカウントモードで、カウンタアンダーフローごと
- センターアラインモードで、カウンタオーバーフローとカウンタアンダーフローごと最大繰り返し回数は 32768 PWM サイクルに限られますが、PWM 周期ごとにデューティサイクルを 2 回更新することが可能になります。センターアラインモードで比較レジスタの値を PWM 周期あたり 1 回のみ更新するときには、パターンが対称なので、最大精度は  $2xT_{ck}$  です。

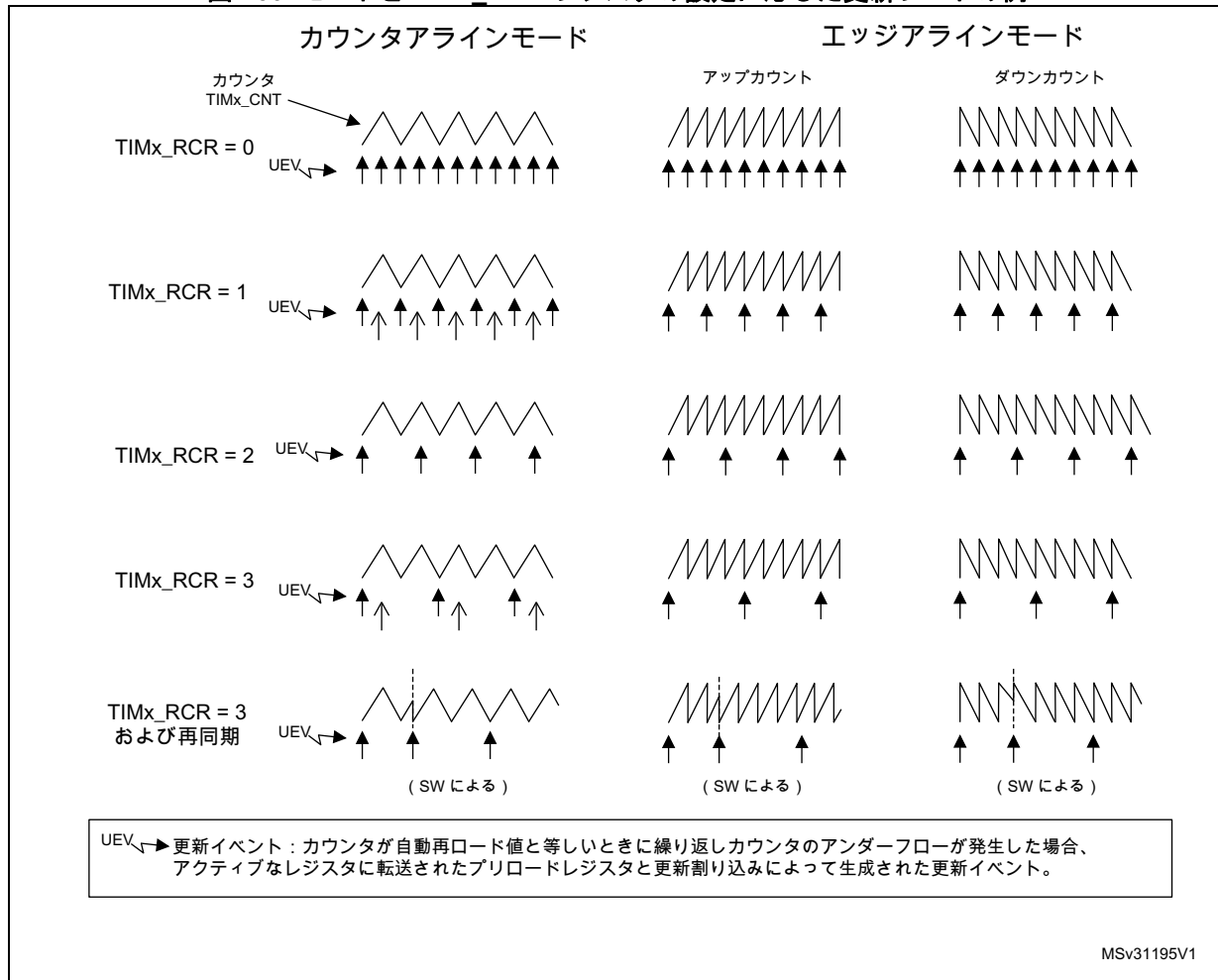
繰り返しダウンカウンタは自動再ロードタイプです。繰り返しの回数は、TIMx\_RCR レジスタの値によって定義されたとおりに維持されます (図 160 を参照してください)。ソフトウェアによって (TIMx\_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰り返しカウンタの値にかかわらず常にイベントが発生し、繰り返しカウンタに TIMx\_RCR レジスタの内容が再ロードされます。

センターアラインモードでは、RCR が奇数の場合、RCR レジスタが書き込まれたタイミングおよびカウンタが開始されたタイミングに応じてオーバーフローまたはアンダーフロー時に更新イベント

が発生します。カウントの開始前に RCR が書き込まれた場合は、オーバーフローで、UEV が発生します。カウントの開始後に RCR が書き込まれた場合は、アンダーフローで UEV が発生します。

たとえば、RCR = 3 の場合、RCR の書き込みタイミングに応じて 4 回目のオーバーフローイベントまたはアンダーフローイベントごとに UEV が発生します。

図 160. モードと TIMx\_RCR レジスタの設定に応じた更新レートの例



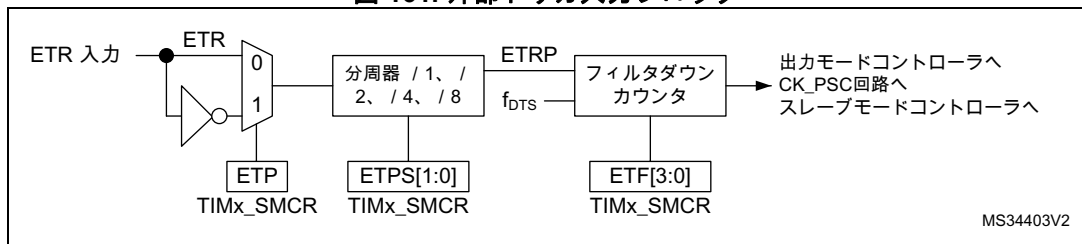
## 20.3.4 外部トリガ入力

タイマには外部トリガ入力 ETR 機能があります。以下の目的で使用できます。

- 外部クロック (外部クロックモード 2、[セクション 20.3.5](#) を参照)
- スレープモードのトリガ ([セクション 20.3.25](#) を参照)
- サイクルごとの電流調整の PWM リセット入力 ([セクション 20.3.7](#) を参照)

以下の [図 161](#) は、ETR の入力条件付けについて説明しています。入力の極性は、TIMxSMCR レジスタの ETP ビットで定義されています。トリガは ETPS[1:0] ビットフィールドでプログラムされた分周器でプリスケールし、ETF[3:0] ビットフィールドでデジタル的にフィルタリングすることができます。

図 161. 外部トリガ入力ブロック



## 20.3.5 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

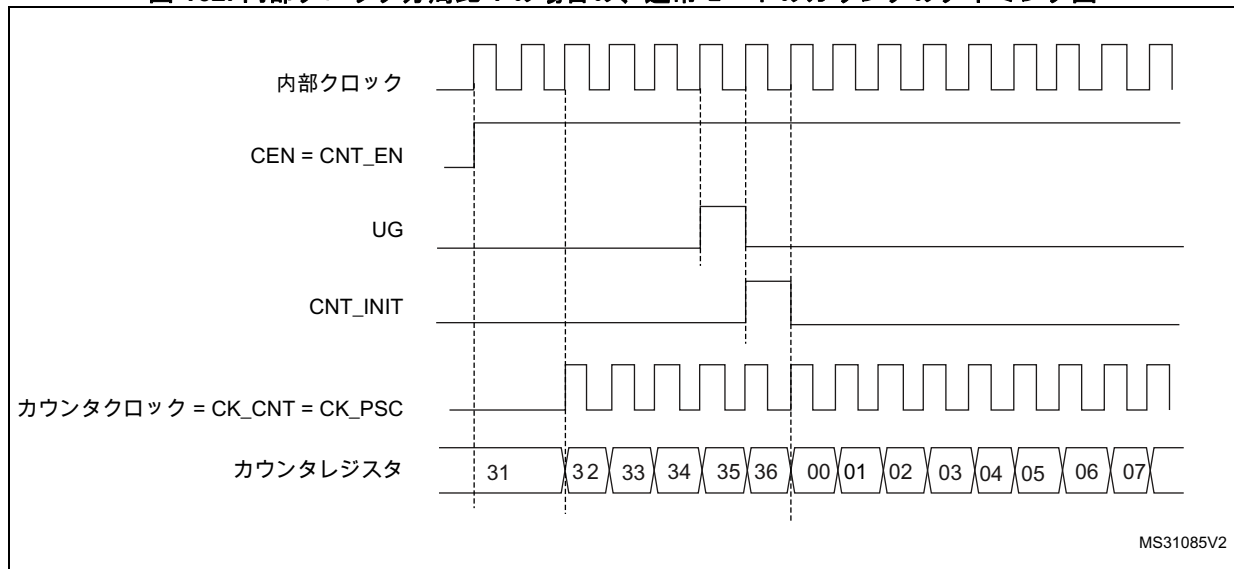
- 内部クロック (CK\_INT)
- 外部クロックモード 1: 外部入力ピン
- 外部クロックモード 2: 外部トリガ入力 ETR
- エンコーダモード

### 内部クロックソース (CK\_INT)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx\_CR1 レジスタ)、および UG ビット (TIMx\_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK\_INTが供給されます。

図 162 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

図 162. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図

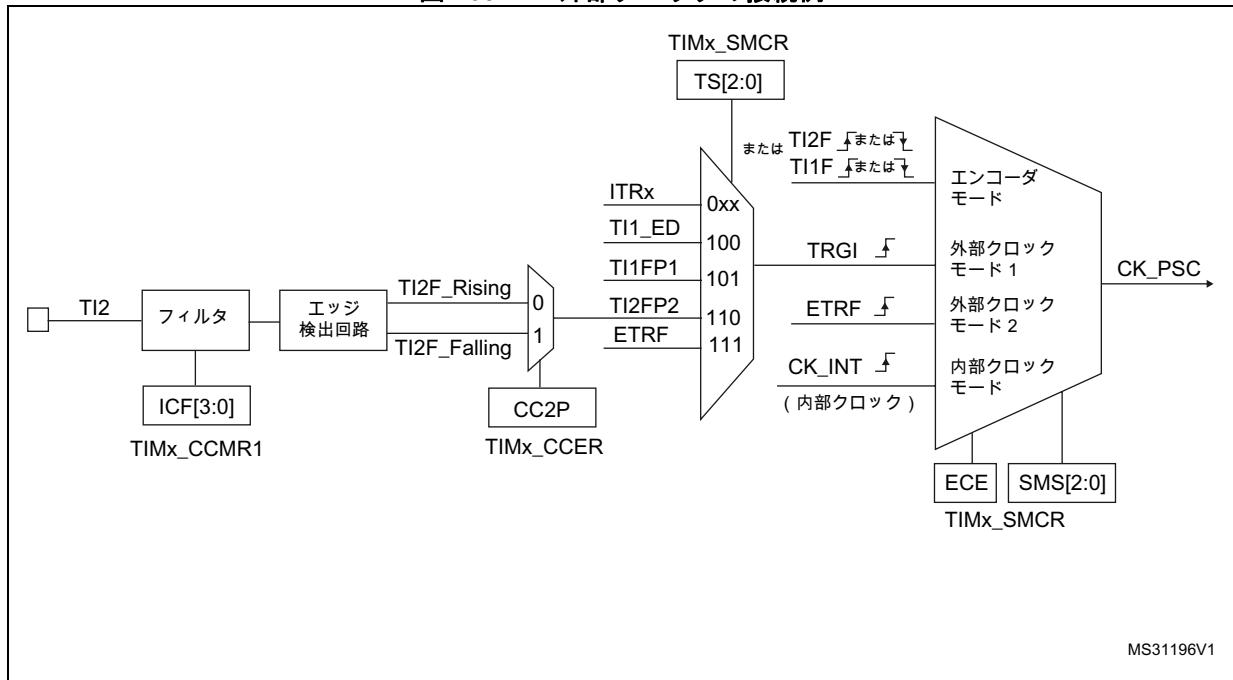


### 外部クロックソースモード 1

このモードは TIMx\_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりにエッジでカウントすることができます。



図 163. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

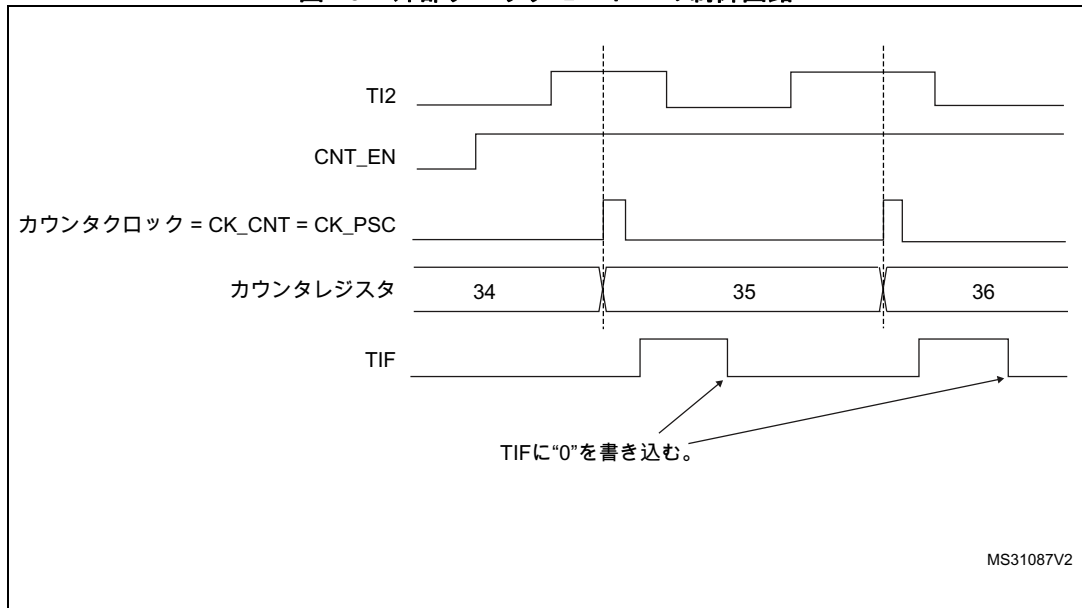
1. TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx\_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。
3. CC2P=0 と CC2NP=0 を TIMx\_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx\_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx\_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

**注：** キャプチャプリスケラはトリガには使用されないため、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 164. 外部クロックモード 1 の制御回路



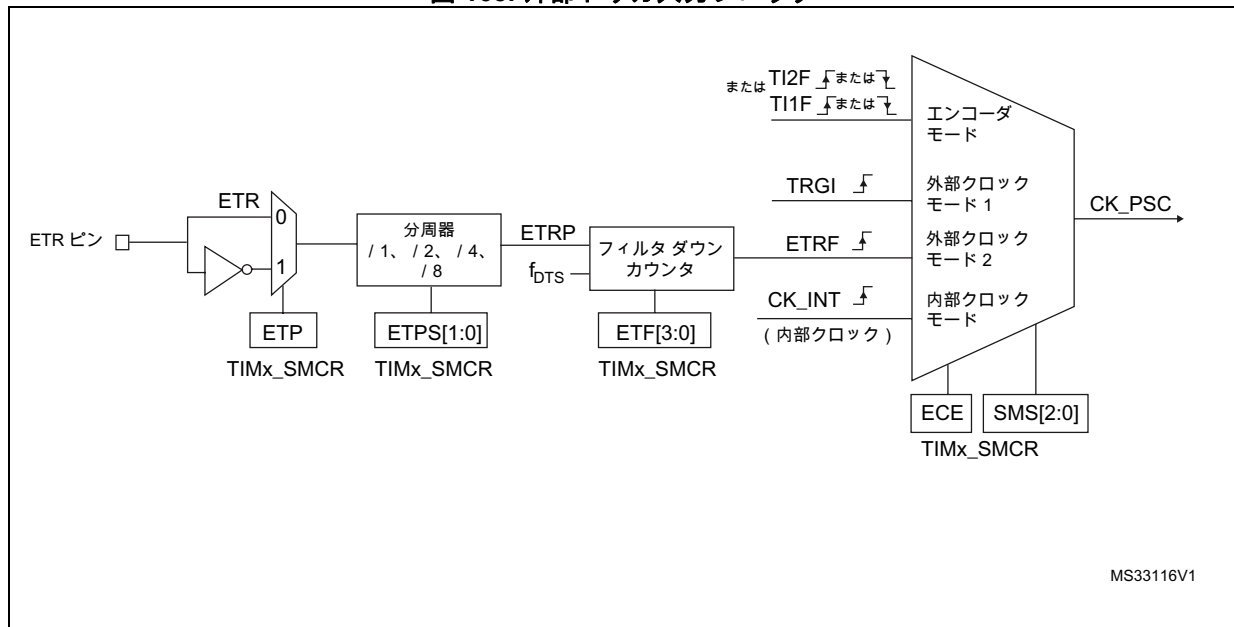
## 外部クロックソースモード 2

このモードは、TIMx\_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 165 に、外部トリガ入力ブロックの概要を示します。

図 165. 外部トリガ入力ブロック



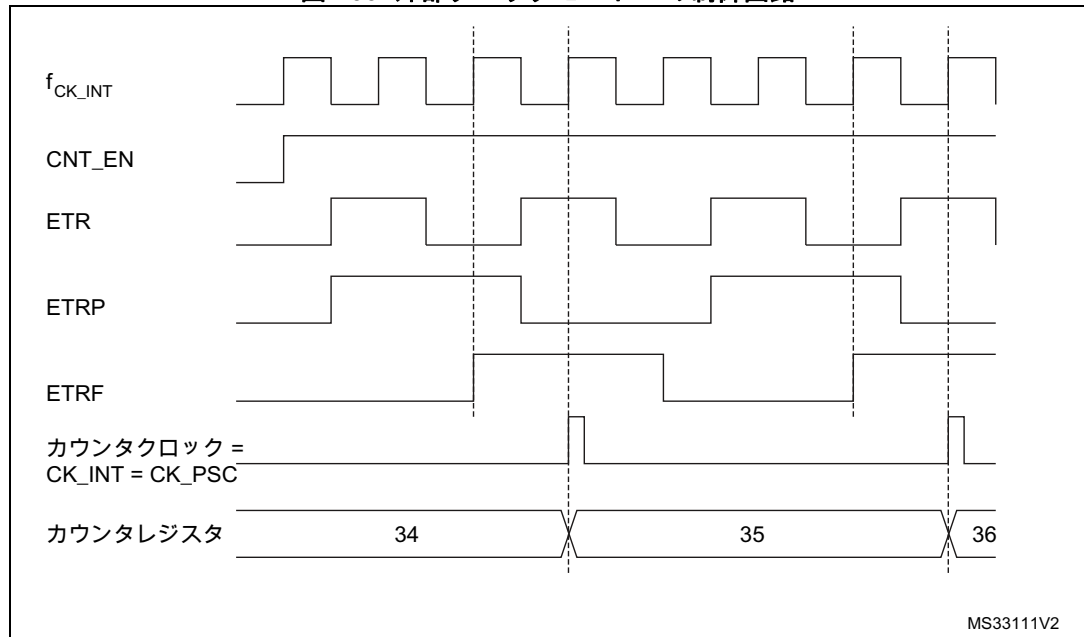
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx\_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx\_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx\_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx\_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 166. 外部クロックモード 2 の制御回路



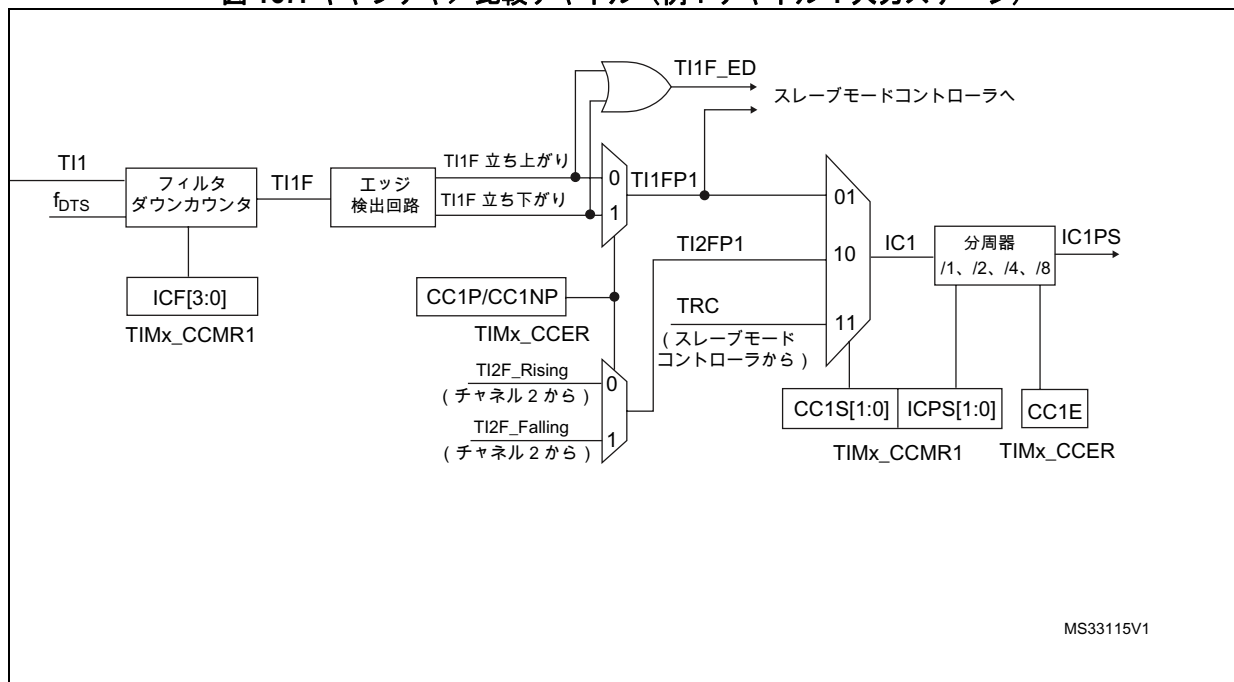
## 20.3.6 キャプチャ／比較チャンネル

各キャプチャ／比較チャンネルは、キャプチャ／比較レジスタ (シャドウレジスタを含む)、キャプチャの入カステージ (チャンネル 5 および 6 を除くデジタルフィルタ、マルチプレクサ、プリスケアラ)、および出カステージ (比較回路と出力制御) から構成されています。

図 167 から 図 170 に、1 つのキャプチャ／比較チャンネルの概要を示します。

入カステージは、対応する Tlx 入カをサンプリングして、フィルタリングを行った TlxF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入カとして、またはキャプチャコマンドとして使用される信号 (TlxFPx) を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 167. キャプチャ／比較チャンネル (例: チャンネル 1 入カステージ)



出カステージは、OCxRef (アクティブハイ) として使用される中間波形を生成します。OCxRef はアクティブハイです。信号の極性は最終出力に影響を与えます。

図 168. キャプチャ/比較チャンネル 1 メイン回路

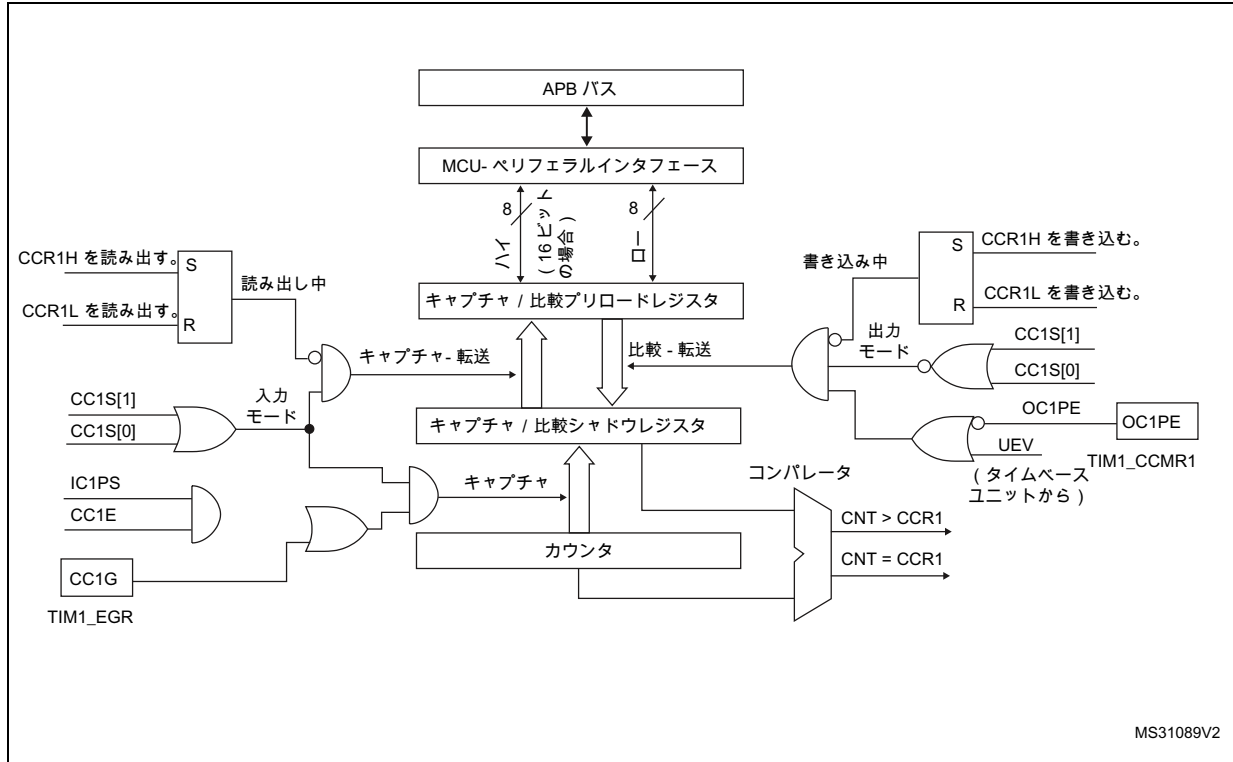
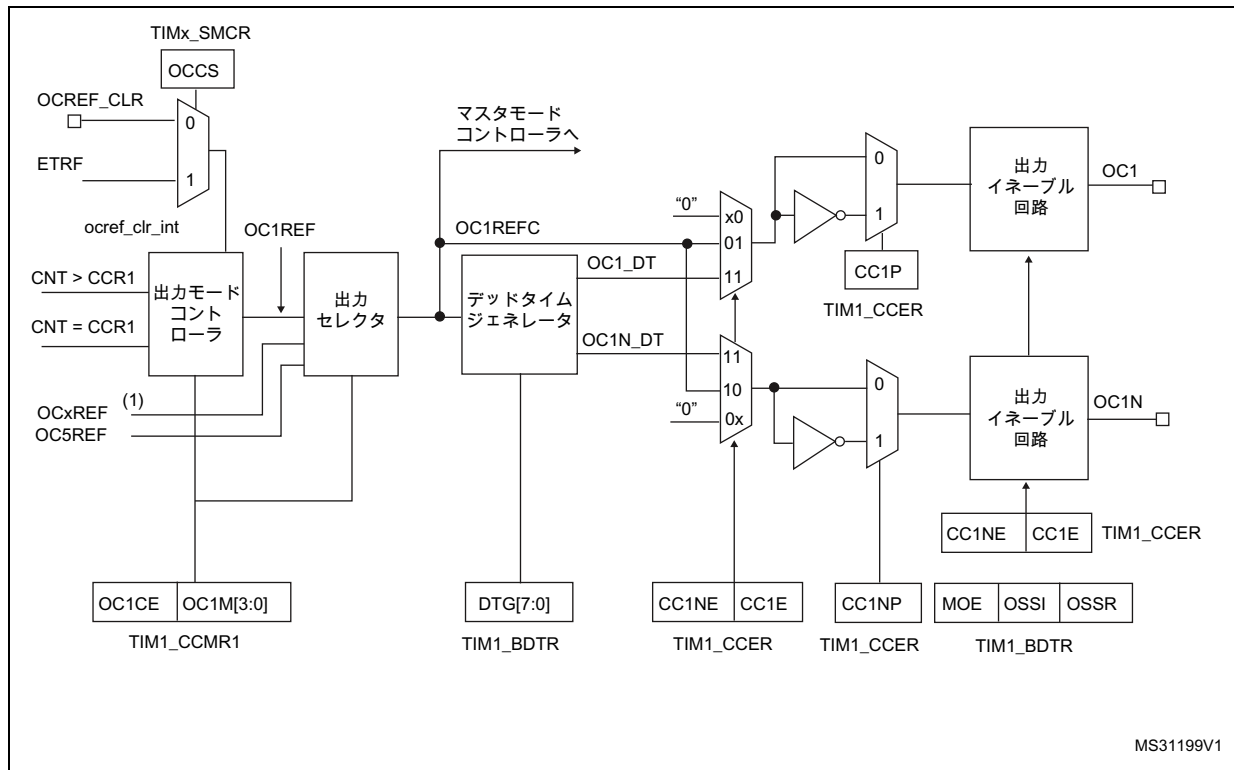


図 169. キャプチャ/比較チャンネル (チャンネル 1、同じくチャンネル 2 および 3) の出力ステージ



1. OCxREF、ここで x は相補チャンネルのランク

図 170. キャプチャ/比較チャンネル (チャンネル 4) の出力ステージ

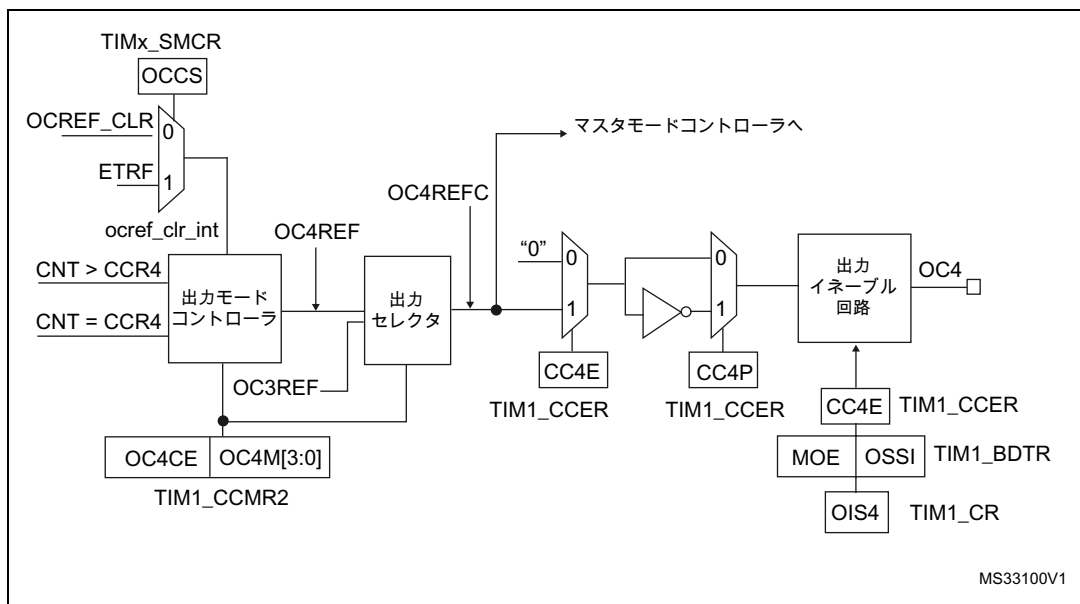
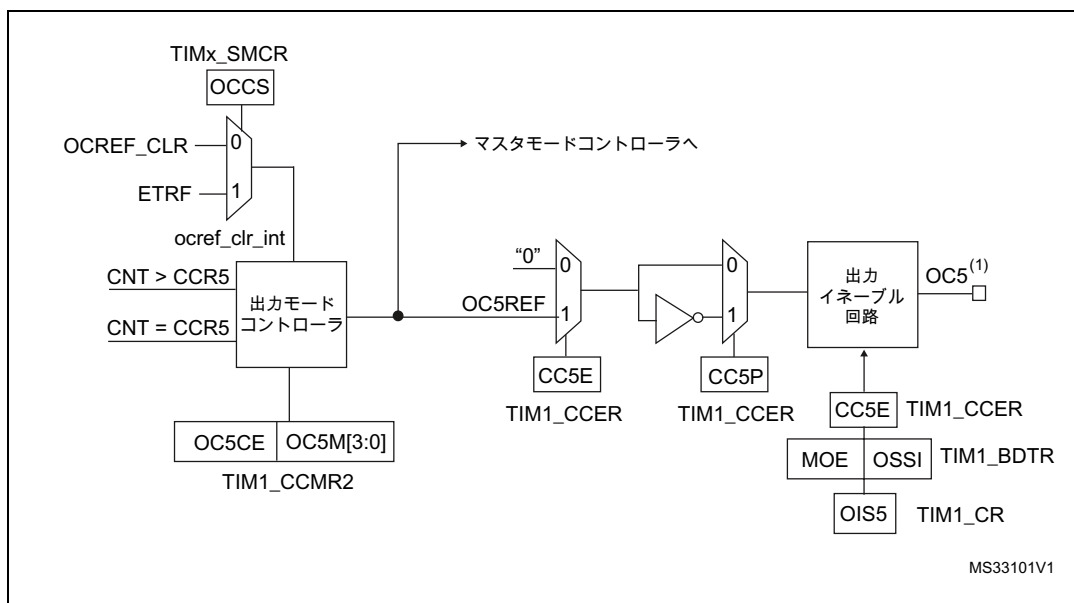


図 171. キャプチャ/比較チャンネル (チャンネル 5、同じくチャンネル 6) の出力ステージ



1. 外部的には使用できません。

キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

## 20.3.7 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx\_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx\_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx\_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx\_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx\_CCR1 は TI1 入力とリンクされていなければならないため、このためには TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx\_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数  $f_{DTS}$  でサンプリング)。この場合、TIMx\_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。

- TI1 チャンネルのアクティブ変化のエッジを選択します。このためには、TIMx\_CCER レジスタの CC1P ビットと CC1NP ビットに“0”を書き込みます（この場合、立ち上がりエッジの選択）。
- 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
- TIMx\_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要な場合は、TIMx\_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx\_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

**注：** IC 割り込みと DMA リクエストは、TIMx\_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

## 20.3.8 PWM 入力モード

このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

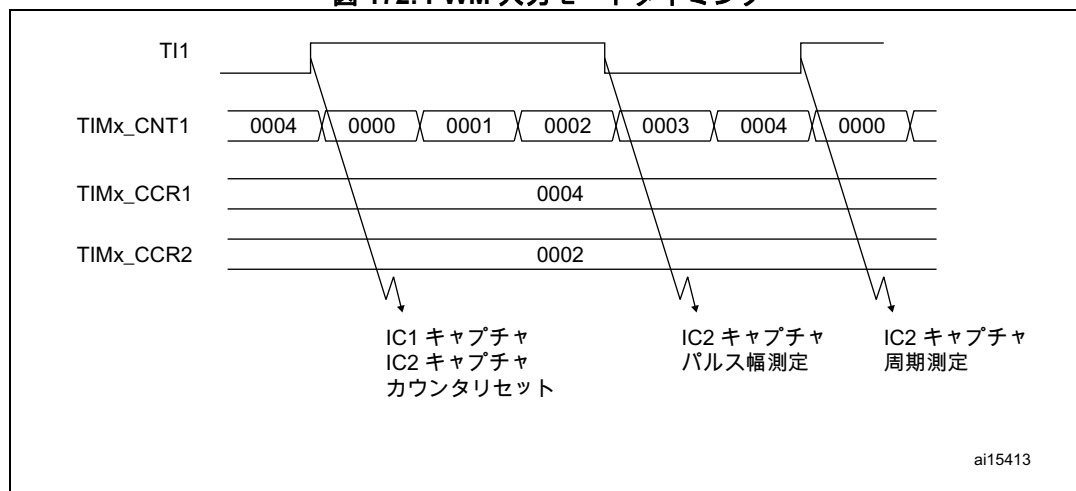
- 2 つの ICx 信号が同じ Tlx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。



たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます (手順は、CK\_INT 周波数とプリスケール値によって、若干異なることがあります)。

- TIMx\_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx\_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx\_CCR1 のキャプチャとカウンタクリアの両方に使用)。
- TIMx\_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
- CC2P ビットと CC2NP ビットに CC2P/CC2NP=“10”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx\_CCR2 のキャプチャに使用されます)。
- TIMx\_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx\_SMCR レジスタの SMS ビットに 0100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx\_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 172. PWM 入力モードタイミング



## 20.3.9 強制出力モード

出力モード (TIMx\_CCMRx レジスタの CCxS ビット =00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx\_CCMRx レジスタの OCxM ビットに 0101 を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例 : CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx\_CCMRx レジスタの OCxM ビットに“0100”を書き込むことによって、強制的にローにできます。



いずれにしても、TIMx\_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みやDMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

## 20.3.10 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。マイクロコントロール内でチャンネル 5 および 6 のみが使用可能である場合に、チャンネル 1 から 4 を出力できません (たとえば、合成波形生成または ADC トリガのため)。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx\_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=0000)、アクティブにセットされるか (OCxM=0001)、非アクティブにセットされるか (OCxM=0010)、または反転されます (OCxM=0011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx\_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx\_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx\_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信しません (DMA リクエスト選択には、TIMx\_CR2 レジスタの CCDS ビットが使用されます)。

TIMx\_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx\_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

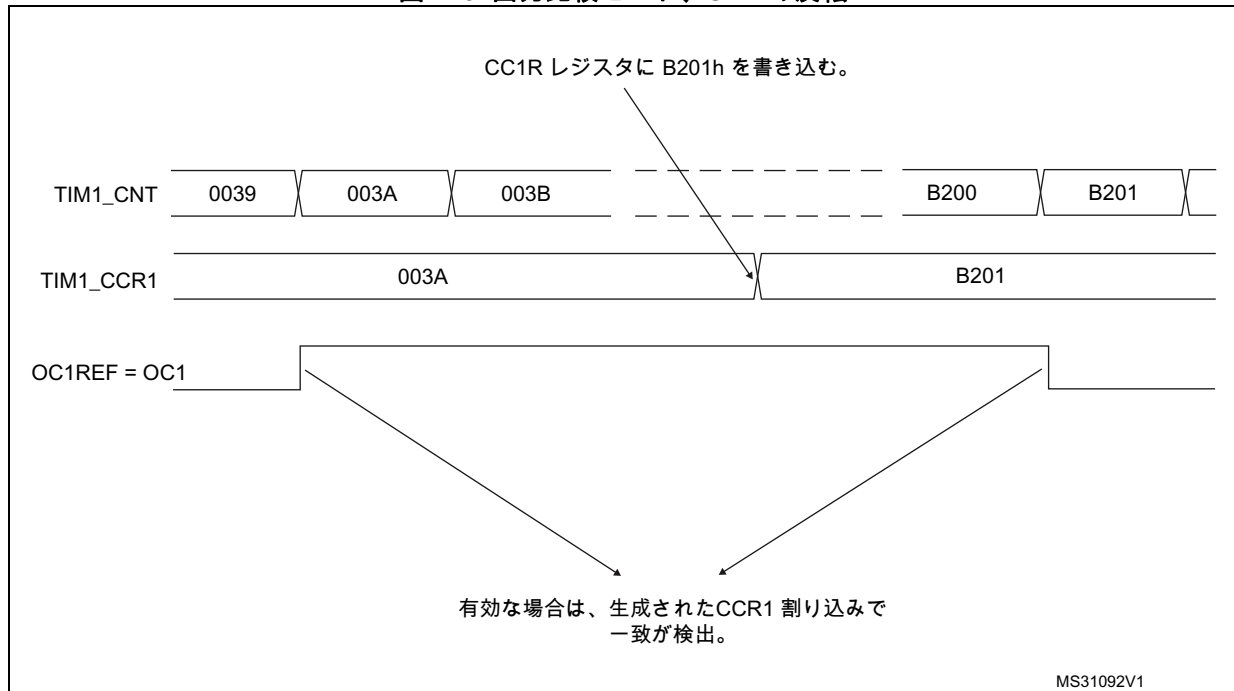
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは、単一パルスを出力するためにも使用できません (ワンパルスモード)。

### 手順

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx\_ARR レジスタと TIMx\_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
  - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに 0011 を書き込みます。
  - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
  - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
  - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
5. TIMx\_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx\_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx\_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 173](#) に示します。

図 173. 出力比較モード、OC1 の反転



## 20.3.11 PWM モード

パルス幅変調(PWM)モードでは、TIMx\_ARR レジスタの値によって決められた周波数と TIMx\_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“0110” (PWM モード 1) または“0111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx\_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx\_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx\_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx\_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx\_CCER および TIMx\_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx\_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx\_CNT と TIMx\_CCRx が常に比較されて、TIMx\_CCRx ≤ TIMx\_CNT または TIMx\_CNT ≤ TIMx\_CCRx がどうか判断されます (カウントの方向によります)。

タイマは、TIMx\_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

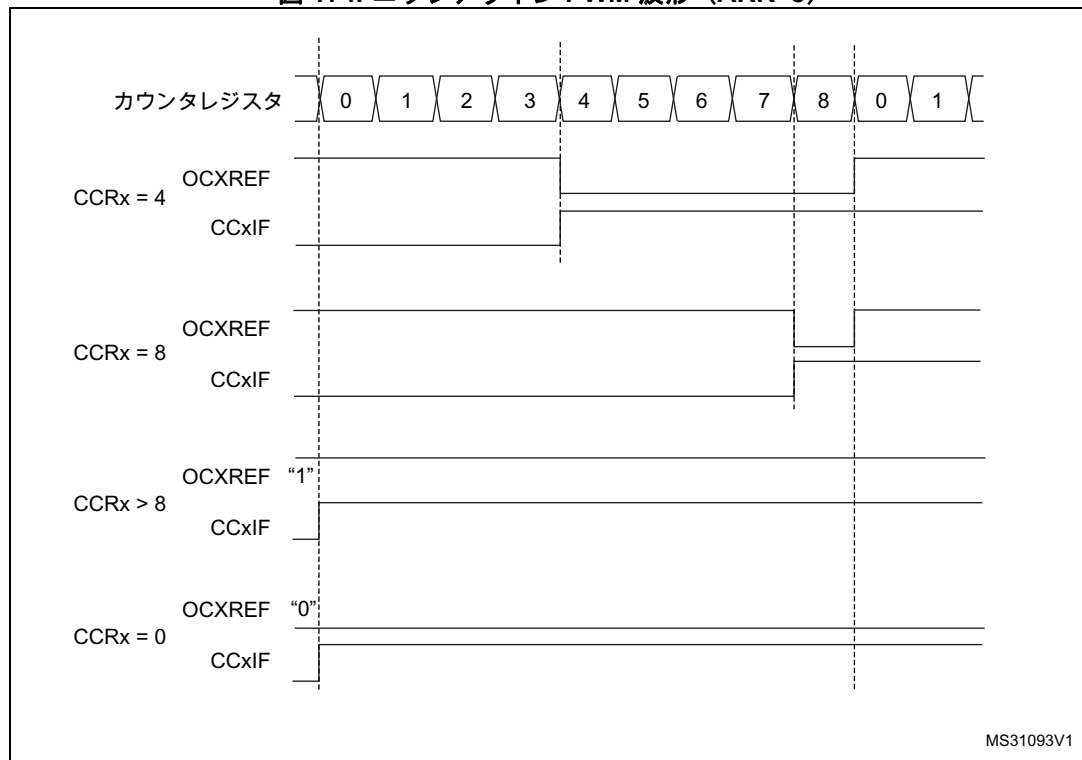
## PWM エッジアラインモード

- アップカウント構成

TIMx\_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード \(506 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx\_CNT < TIMx\_CCRx の間はハイに、そうでない場合はローになります。TIMx\_CCRx の比較値が自動再ロード値 (TIMx\_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 174](#) に TIMx\_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 174. エッジアライン PWM 波形 (ARR=8)



- ダウンカウント構成

TIMx\_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード \(510 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx\_CNT > TIMx\_CCRx の間はローであり、そうでない場合はハイになります。TIMx\_CCRx の比較値が TIMx\_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0% の PWM 信号を生成することはできません。

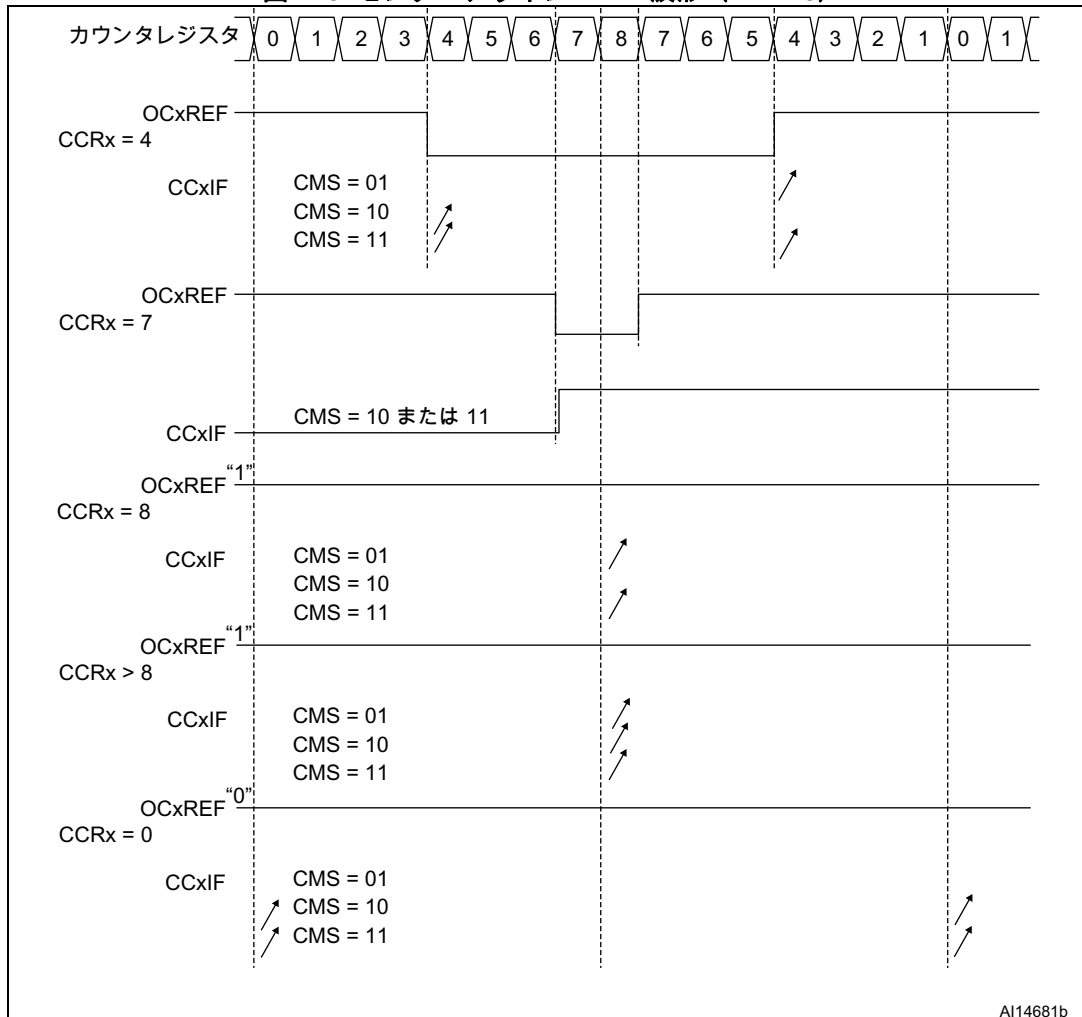
## PWM センターアラインモード

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”でないときにアクティブです (その他すべての構成は、OCxRef/OCx 信号に対して同じ効果を持ちます)。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx\_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターアラインモード \(アップ/ダウンカウント\) \(513 ページ\)](#) を参照してください。

図 175 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx\_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx\_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 175. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx\_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。

- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
  - 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx\_CNT > TIMx\_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
  - カウンタに 0 または TIMx\_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx\_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

### 20.3.12 非対称 PWM モード

非対称モードでは、プログラム可能な位相シフトによって 2 つのセンターアライン PWM 信号の生成を可能にします。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや位相シフトは TIMx\_CCRx レジスタペアで決定されます。1 つ目のレジスタがアップカウント時の PWM を制御し、2 つ目のレジスタがダウンカウント時の PWM を制御することで、PWM は PWM ハーフサイクルごとに調整されます。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

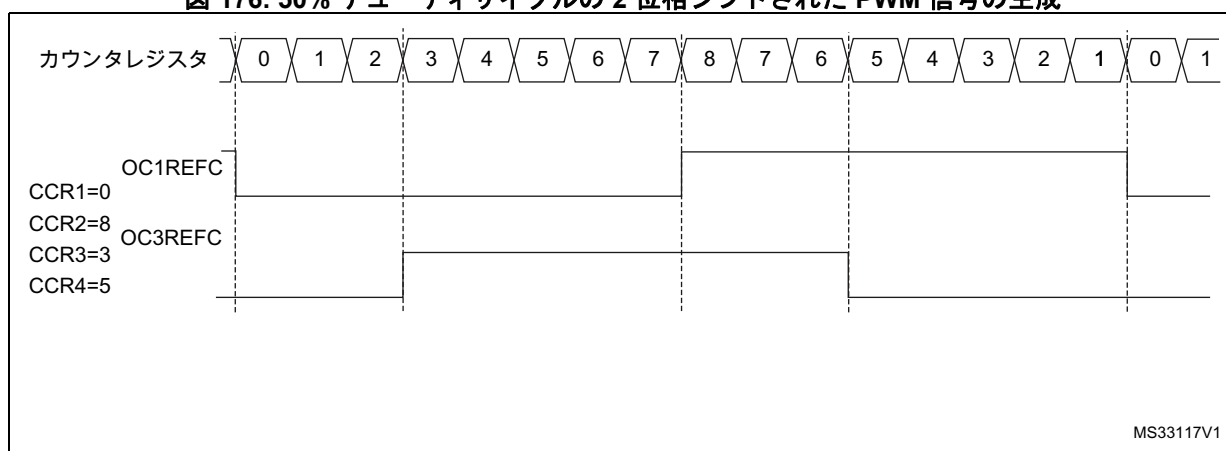
非対称 PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1110” (非対称 PWM モード 1) または“1111” (非対称 PWM モード 2) を書き込むことによって、2 チャネルごとに選択できます (CCR レジスタペアごとに 1 つの OCx 出力)。

**注：** OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

特定のチャネルが非対称の PWM チャネルとして使用されると、その相補チャネルも使用できます。たとえば、OC1REFC 信号がチャネル 1 (非対称 PWM モード 1) に生成されると、チャネル 2 の OC2REF 信号、または非対称 PWM モード 1 の結果として得られる OC2REFC 信号を出力できます。

図 176 は、非対称 PWM モードを使用して生成される信号の例を表します (チャネル 1 から 4 は非対称 PWM モード 1 として設定されます)。これにより、デッドタイムジェネレータとともにフルブリッジ位相シフト DC-DC コンバータを制御できます。

図 176. 50% デューティサイクルの 2 位相シフトされた PWM 信号の生成



MS33117V1

## 20.3.13 組み合わせ PWM モード


組み合わせ PWM モードでは、2つのエッジラインまたはセンターアライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2つの TIMx\_CCRx レジスタで決定されます。結果として得られる信号 OCxREFC は、2つの PWM 基準信号の OR または AND による論理結合から成ります。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

組み合わせ PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1100” (組み合わせ PWM モード 1) または“1101” (組み合わせ PWM モード 2) を書き込むことによって、2チャンネルごとに選択できます (CCR レジスタペアごとに 1つの OCx 出力)。

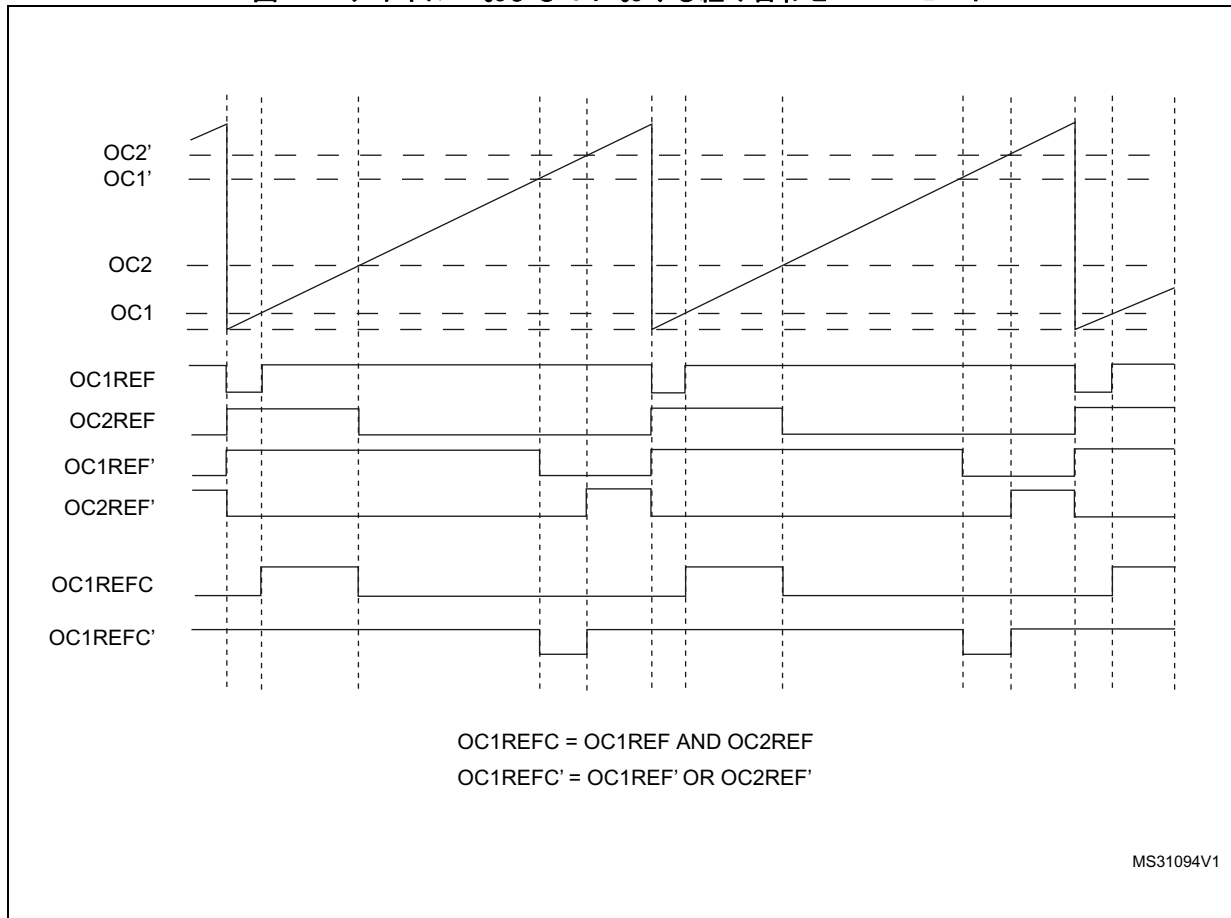
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、相補チャンネルを反対の PWM モードに設定する必要があります (たとえば、1つを組み合わせ PWM モード 1、もう 1つを組み合わせ PWM モード 2 にします)。

**注:** OCxM[3:0] ビットフィールドは互換性を確保するために 2つのパーツに分割され、最上位ビットと 3つの最下位ビットとは隣接していません。

 177 は、次の設定で取得可能な非対称 PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合
- チャンネル 3 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 4 が PWM モード 1 で設定されている場合

図 177. チャンネル 1 および 3 における組み合わせ PWM モード



## 20.3.14 組み合わせ 3 相 PWM モード

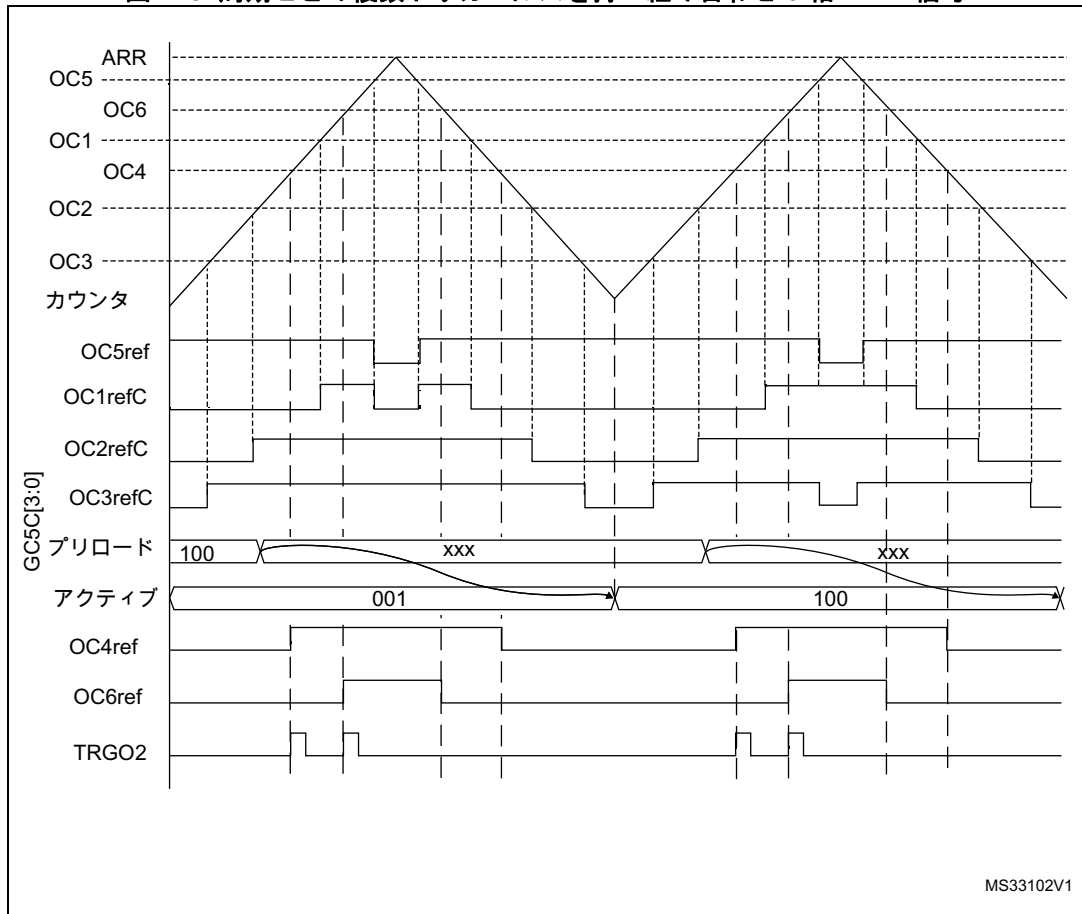
組み合わせ 3 相 PWM モードでは、パルスの中で論理積を取った単一のプログラム可能な信号とともに 1 つから 3 つのセンターアライン PWM 信号を生成できます。結果として得られる組み合わせ信号の定義には、OC5REF 信号が使用されます。TIMx\_CCR5 の 3 ビット GC5C[3:1] では、OC5REF を組み合わせる基準信号を選択できます。結果として得られる信号 OCxREFC は、2 つの PWM 基準信号の AND による論理結合から生成されます。

- GC5C1 がセットされると、OC1REFC は TIMx\_CCR1 および TIMx\_CCR5 によって制御されます。
- GC5C2 がセットされると、OC2REFC は TIMx\_CCR2 および TIMx\_CCR5 によって制御されます。
- GC5C3 がセットされると、OC3REFC は TIMx\_CCR3 および TIMx\_CCR5 によって制御されます。

組み合わせ 3 相 PWM モードは、少なくとも 3 ビット GC5C[3:1] の 1 つをセットすることでチャンネル 1 から 3 で個別に選択できます。



図 178. 周期ごとの複数トリガパルスを持つ組み合わせ 3 相 PWM 信号



TRGO2 波形は、特定の 3 相 PWM 信号での ADC の同期方法を示します。詳細については、[セクション 20.3.26 : ADC の同期](#)を参照してください。

## 20.3.15 相補出力とデッドタイム挿入

高機能制御タイマ (TIM1/TIM8/TIM20) は、2 つの相補信号を出力して、出力時のスイッチオンおよびスイッチオフを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性 (レベルシフタの内在的な遅延、電源スイッチによる遅延など) に応じて調整する必要があります。

出力の極性 (主出力 OCx または補 OCxN) は、出力ごとに独自に選択できます。これは TIMx\_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 OCx および OCxN は、TIMx\_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx\_BDTR レジスタと TIMx\_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、[表 121: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット \(581 ページ\)](#)を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるときに)、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッ

ドタイムジェネレータがあります。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点が異なります。
- OCxN 出力信号は、立ち上がりエッジが基準波形の立ち下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (OCx または OCxN) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 OCxREF との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 179. デッドタイム挿入のある相補出力

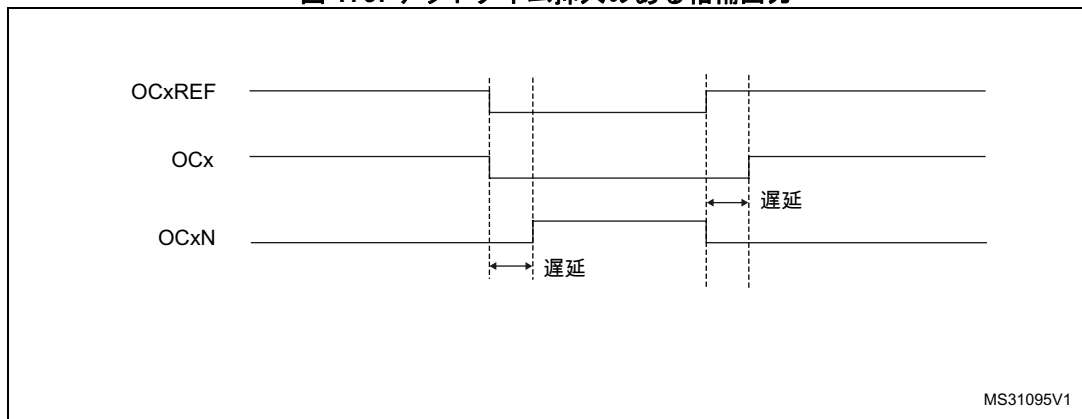


図 180. 負のパルスより長い遅延があるときのデッドタイムの波形

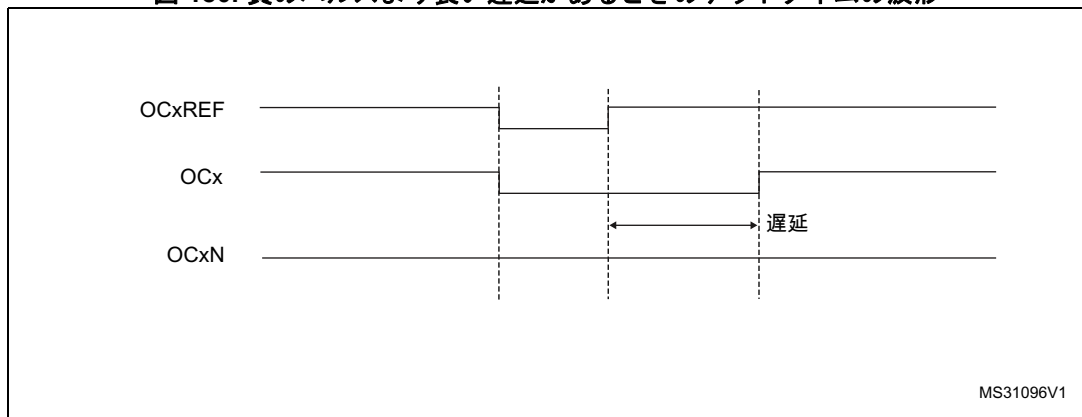
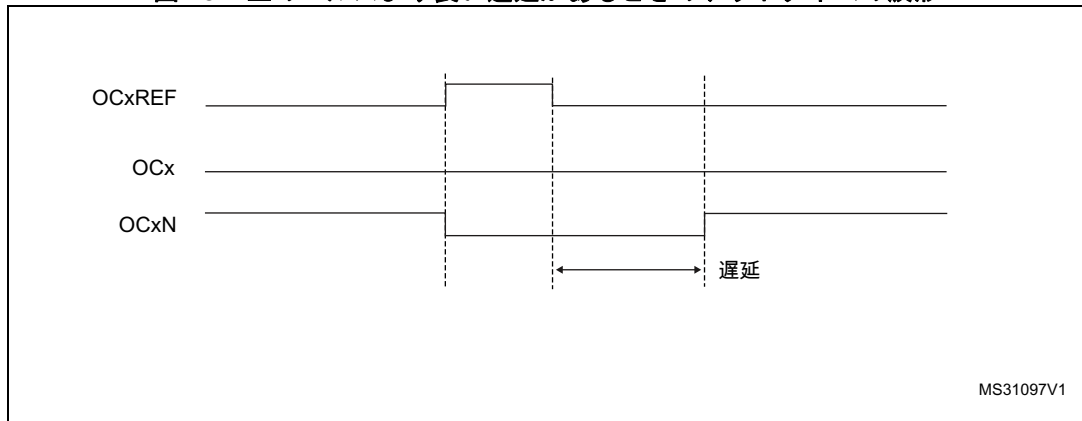


図 181. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャンネルで同じであり、TIMx\_BDTR レジスタの DTG ビットでプログラミングできます。遅延計算については、[セクション 20.4.18 : TIM1/TIM8/TIM20 ブレークおよびデッドタイムレジスタ \(TIMx\\_BDTR\)](#) を参照してください。

### OCxREF 信号の OCx または OCxN へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx\_CCER レジスタの CCxE ビットおよび CCxNE ビットを構成することによって、OCxREF 信号を OCx 出力または OCxN 出力にリダイレクトできます。

これにより、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定することができます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができます。

**注：** OCxN のみが有効なときには (CCxE=0, CCxNE=1)、相補にならず、OCxREF がハイレベルとなるとアクティブになります。たとえば、CCxNP=0 の場合は、OCxN=OCxRef です。他方、OCx と OCxN の両方が有効なときには (CCxE=CCxNE=1)、OCxREF がハイになると OCx はアクティブになり、OCxREF がローのときには、OCxN は補信号であり、アクティブになります。

### 20.3.16 ブレーク機能の使用

ブレーク機能の目的は、TIM1 および TIM8 タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。2つのブレーク入力は通常、パワーステージおよび3相インバータの異常出力に接続されています。アクティブ化すると、ブレーク回路は PWM 出力を遮断し、強制的に事前定義された安全な状態に移行させます。

ブレーク機能を使用しているときには、出カイナーブル信号とインアクティブレベルは追加の制御ビット (TIMx\_BDTR レジスタの MOE、OSSI、および OSSR ビットと TIMxCR2 レジスタの OISx および OISxN ビット) に応じて変更されます。ただし、OCx および OCxN 出力の両方を同時にアクティブレベルに設定することはできません。詳細については、[表 121: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット \(581 ページ\)](#) を参照してください。

BRK のソースは以下のいずれかです。

- BKIN ピンに接続された外部ソース
- 内部ソース : COMP4/7 出力

BRK\_ACTH のソースは以下のいずれかです。

- CSS によって生成されたクロック障害イベント。CSS の詳細については、[セクション 9.2.7: クロックセキュリティシステム \(CSS\)](#) を参照してください。
- COMP1/2/3/5/6 出力
- PVD 出力
- SRAM パリティエラー信号
- Cortex-M4<sup>®</sup>F LOCKUP (ハードフォルト) 出力

BRK2 のソースは以下のいずれかです。

- BKIN2 ピンに接続された外部ソース
- コンパレータ出力からの内部ソース

BRK2 で結果として得られる信号は、BKIN2 ピンの外部信号とコンパレータ出力の間の OR です (BRK2 イベントソースとして選択された場合)。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。TIMx\_BDTR レジスタの BKE および BKE2 ビットをセットすることによって、ブレーク機能を有効にできます。ブレーク入力の極性は、同じレジスタの BKP および BKP2 ビットを設定することによって選択できます。BKEx と BKPx は、同時に変更できます。BKEx および BKPx ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がリエッジは非同期のことがあるので、実際の信号 (出力に作用する信号) と同期制御ビット (TIMx\_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延 (ダミー命令) を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

ブレークは、任意のブレーク入力 (BRK、BRK2、BRK\_ACTH)、BRK、および BRK2 で生成できます。

- プログラム可能な極性 (TIMx\_BDTR レジスタの BKPx ビット)
- プログラム可能なイネーブルビット (TIMx\_BDTR レジスタの BKEx ビット)
- 誤ったイベントを避けるためのプログラム可能なフィルタ (TIMx\_BDTR レジスタの BKxF[3:0] ビット)

BRK\_ACTH に接続されると、フィルタ機能は使用できず、極性は常にアクティブハイになります。

ブレークイベントは、TIMx\_EGR レジスタで BG および B2G ビットを使用して、ソフトウェアによって生成することもできます。

**注:** *非同期 (クロックなし) 動作は、プログラム可能なフィルタが無効な場合にのみ保証されます。有効になっている場合は、必ずブレークイベントが処理されるように、フェイルセーフクロックモード (たとえば、内部 PLL や CSS を使用) を使用する必要があります。*

ブレークが1つ発生すると (ブレーク入力の1つで選択されたレベル)、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態またはアイドル状態になるか、GPIO コントローラへの制御が解除されます (OSSI ビットで選択)。これは、MCU オシレータがオフの場合に有効です。
- 各出力チャネルは、MOE=0 になったとき、TIMx\_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマは出力の制御 (GPIO コントローラによって引き継がれた) を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには：
  - － 出力は、まずインアクティブ状態に置かれます (極性に依存します)。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
  - － タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください (約 2 CK\_TIM クロックサイクル)。
  - － OSSI=0 の場合、タイマは出力の制御 (ハイインピーダンス状態を強制する GPIO コントローラによって引き継がれた) を解除し、そうでない場合、イネーブル出力はハイのままか、CCxE または CCxNE ビットのどちらかがハイになったときにハイになります。
- ブレーク状態フラグ (TIMx\_SR レジスタの BIF および B2IF ビット) がセットされます。TIMx\_DIER レジスタの BIE ビットがセットされている場合、割り込みが生成されます。TIMx\_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx\_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント (UEV) で再び自動的にセットされます。たとえば、これを使用してレギュレーションを行うことができます。そうでない場合、MOE ビットはアプリケーションが再び“1”をセットするまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

**注：** *ブレーク入力は、信号レベルに対してアクティブです。このため、ブレーク入力がアクティブな間は、MOE をセットできません (自動的にも、ソフトウェアによっても)。この間、ステータスフラグ BIF および B2IF をクリアできません。*

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ (デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性) を固定することができます。TIMx\_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 20.4.18: TIM1/TIM8/TIM20 ブレークおよびデッドタイムレジスタ \(TIMx\\_BDTR\)](#) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。


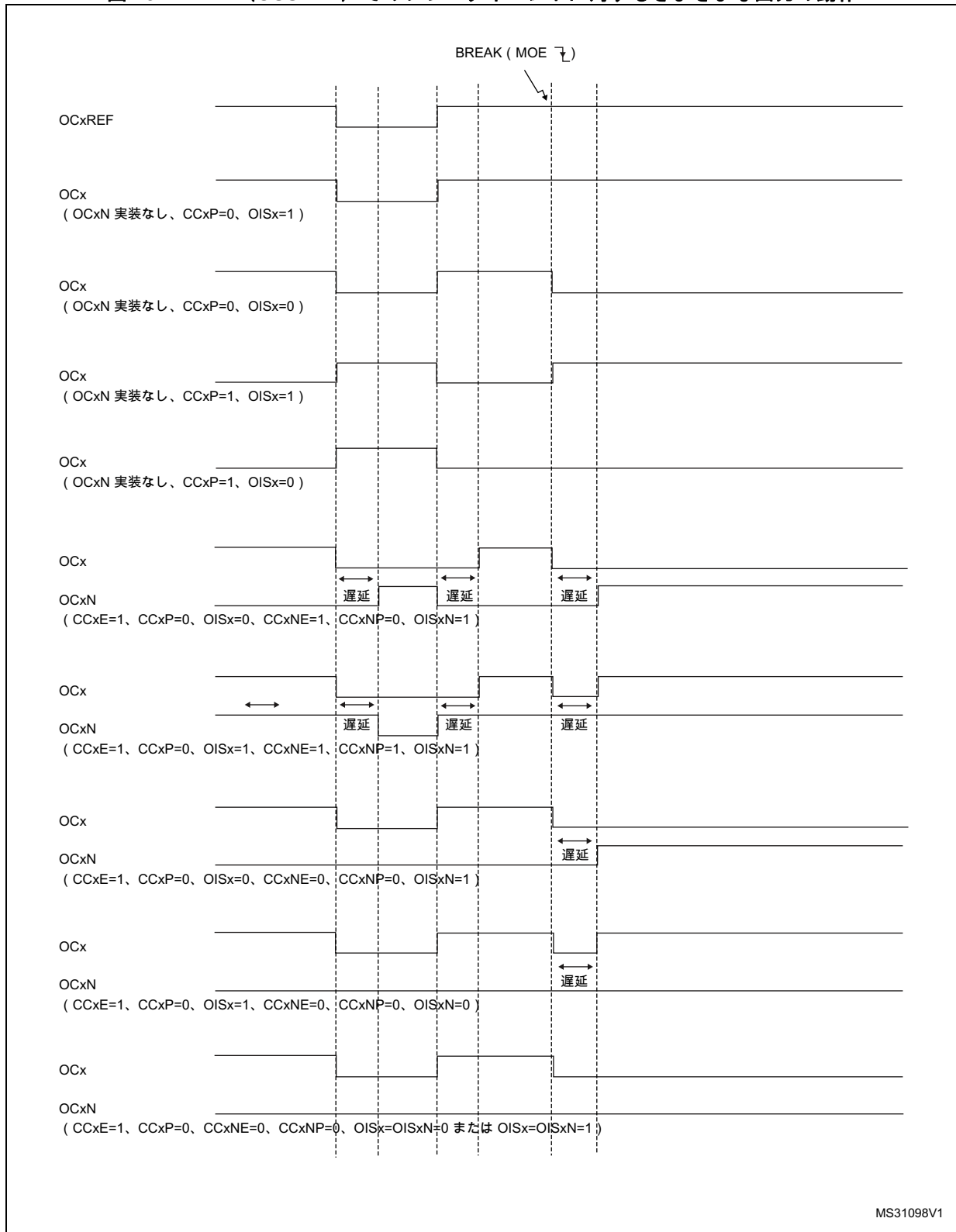
 [182](#) に、ブレークに対する出力の動作例を示します。

図 182. BKIN (OSSI = 1) でのブレークイベントに対するさまざまな出力の動作



2つのブレーク入力は、タイマ出力で異なる動作を示します。

- BRK 入力は、無効化（インアクティブ状態）するか、PWM 出力を強制的に事前定義した安全な状態に移行できます。
- BRK2 は、PWM 出力の無効化（インアクティブ状態）のみ可能です。

表 118 に示すように、BRK の優先順位は BRK2 入力よりも高いです。

注： BRK2 は  $OSSR = OSSI = 1$  の場合にのみ使用してください。

表 118. タイマ出力と BRK/BRK2 入力の動作

BRK	BRK2	タイマ出力状態	通常の使用例	
			OCxN 出力 (ローサイドスイッチ)	OCx 出力 (ハイサイドスイッチ)
アクティブ	X	<ul style="list-style-type: none"> <li>- インアクティブから強制される出力状態（デッドタイム後）</li> <li>- <math>OSSI = 0</math> の場合、出力は無効（GPIO ロジックが制御を引き継ぐ）</li> </ul>	デッドタイム挿入後 ON	OFF
インアクティブ	アクティブ	インアクティブ	OFF	OFF

図 183 では、BKIN および BKIN2 入力で信号がアクティブな場合の、OCx および OCxN 出力の動作の例を示します。この場合、両方の出力がアクティブハイ極性になります（TIMx\_CCER レジスタの  $CCxP = CCxNP = 0$ ）。

図 183. BKIN および BKIN2 ピンのアサート後の PWM 出力状態（ $OSSI=1$ ）

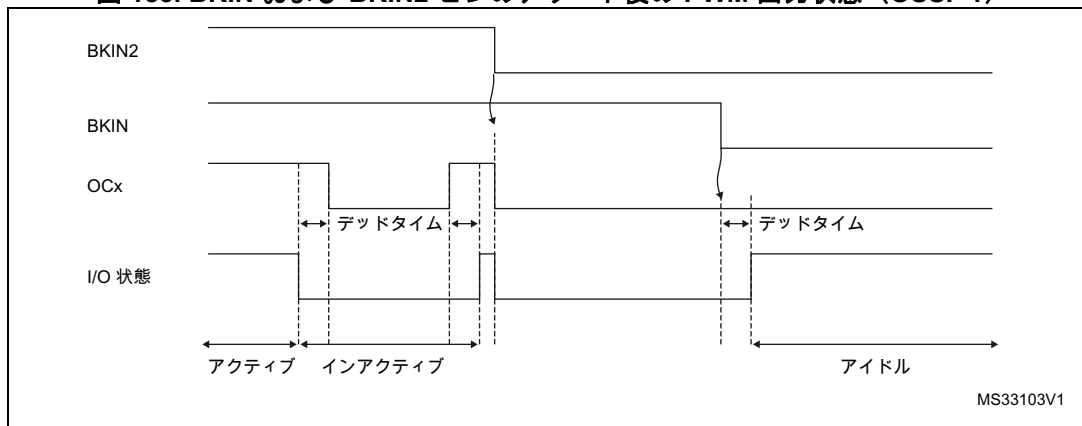
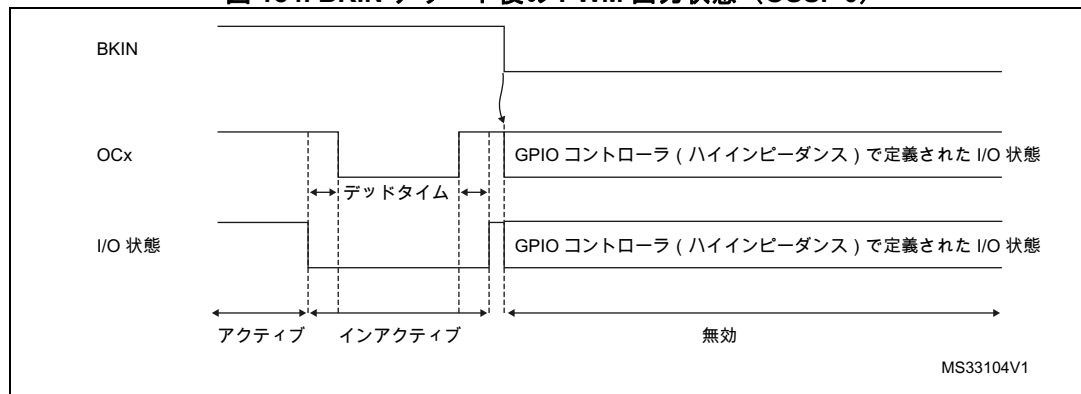


図 184. BKIN アサート後の PWM 出力状態 (OSSI=0)



### 20.3.17 外部イベントによる OCxREF 信号のクリア

特定のチャンネルの OCxREF 信号は ocref\_clr\_int 入力にハイレベルを適用するとクリアされます (対応する TIMx\_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする)。OCxREF は、次の更新イベント (UEV) が発生するまで、ローレベルを保ちます。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。ocref\_clr\_int 入力は、TIMx\_SMCR レジスタで OCCS ビットを設定することで、OCREF\_CLR 入力と ETRF (フィルタ後の ETR) の間で選択できます。

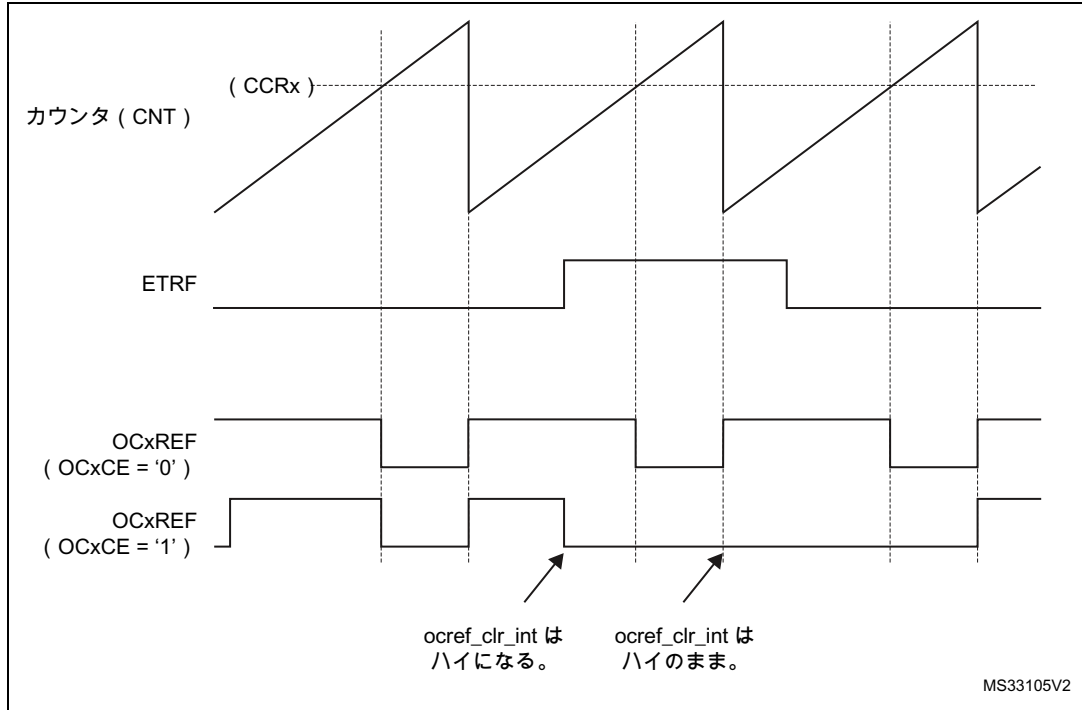
ETRF が選択された場合、ETRF は次のように設定する必要があります。

1. 外部トリガプリスケータはオフ状態に維持します (TIMx\_SMCR レジスタの ETPS[1:0] ビットを“00”にセット)。
2. 外部クロックモード 2 を無効にします (TIMx\_SMCR レジスタの ECE ビットを“0”にセット)。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、ユーザのニーズに応じて設定できます。

図 185 に、イネーブルビット OCxCE の両方の値について、ETRF 入力が高レベルになったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。



図 185. TIMx OCxREF のクリア



注： 100% デューティサイクルのPWMの場合 (CCRx > ARRの場合)、次のカウンタオーバーフローでOCxREFが再度有効になります。

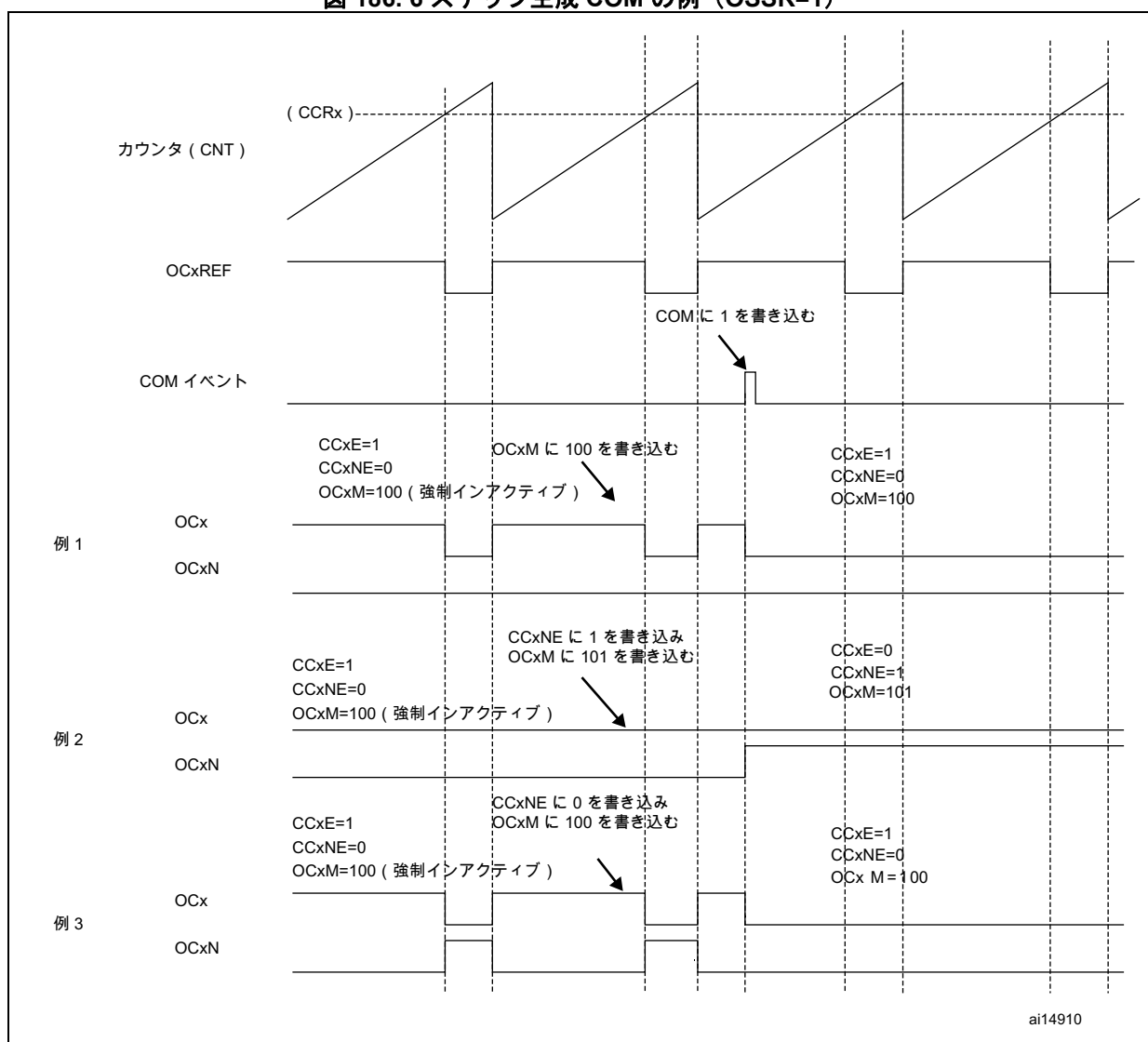
## 20.3.18 6 ステップ PWM 生成

チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転換イベントでシャドウビットにコピーされます。これにより、次のステップの構成をあらかじめプログラミングして、すべてのチャンネルの構成を同時に変更することができます。COM は、TIMx\_EGR レジスタの COM ビットをセットすることによってソフトウェアによって、またはハードウェアによって (TRGI 立ち上がりエッジで) 生成することができます。

フラグは、COM イベントが発生したときにセットされ (TIMx\_SR レジスタの COMIF ビット)、これによって割り込み (TIMx\_DIER レジスタの COMIE ビットがセットされている場合) または DMA リクエスト (TIMx\_DIER レジスタの COMDE ビットがセットされている場合) を生成できます。

図 186 に、COM イベントが発生したときの OCx と OCxN 出力の動作を、3 種類のプログラミング構成の例で示します。

図 186. 6 ステップ生成 COM の例 (OSSR=1)



## 20.3.19 ワンパルスモード

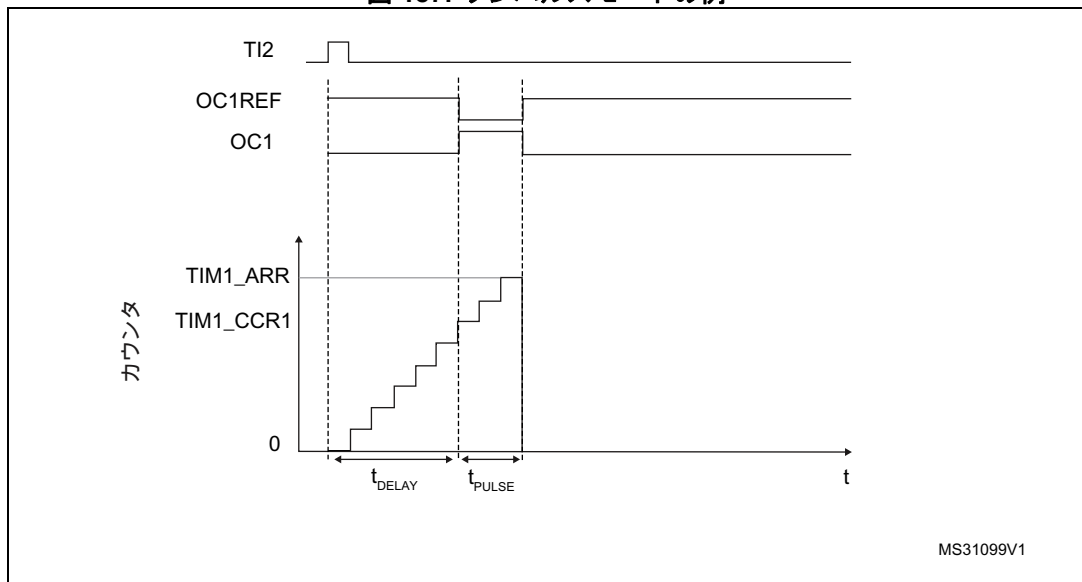
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 :  $CNT < CCRx \leq ARR$  (特に、 $0 < CCRx$ )
- ダウンカウント時 :  $CNT > CCRx$

図 187. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が  $t_{PULSE}$  の正のパルスを遅延時間  $t_{DELAY}$  後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx\_CCER レジスタで CC2P=“0”と CC2NP=“0”を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx\_SMCR レジスタの TS ビットに 110 を書き込みます。
- TI2FP2 を使用してカウンタを開始します。このためには、TIMx\_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- $t_{\text{DELAY}}$  は、TIMx\_CCR1 レジスタに書き込まれた値によって定義されます。
- $t_{\text{PULSE}}$  は、自動再ロード値と比較値の差 (TIMx\_ARR - TIMx\_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx\_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx\_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx\_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx\_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので (シングルモード)、TIMx\_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx\_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース : OCx 高速イネーブル :

ワンパルスモードでは、Tlx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ( $t_{\text{DELAY min}}$ ) が制限されます。

最小遅延で波形を出力したい場合は、TIMx\_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

## 20.3.20 再トリガ可能なワンパルスモード (OPM)

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 20.3.19](#) で説明する再トリガ不可能なワンパルスモードについて、次のような違いがあります。

- パルスはトリガが発生し次第開始します (プログラム可能な遅延はありません)。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

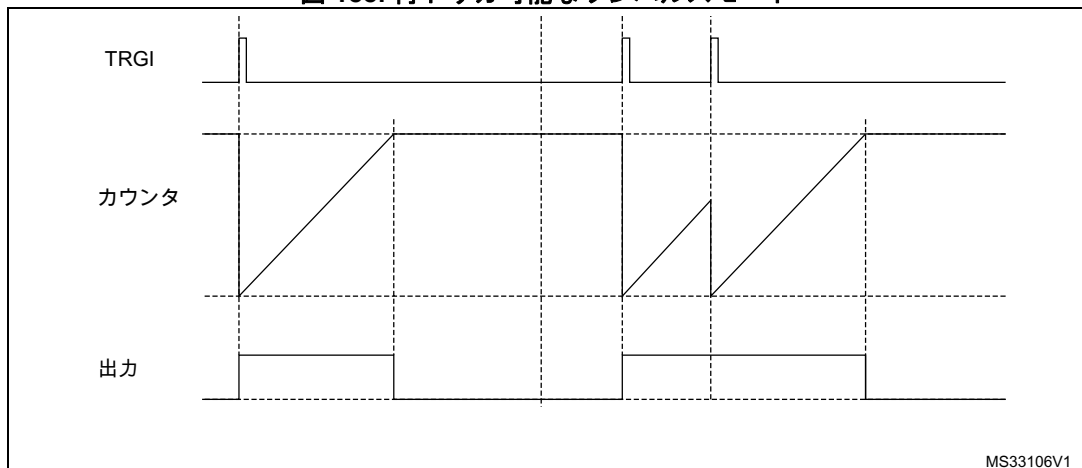
タイマはスレーブモードである必要があり、このときビットは TIMx\_SMCR レジスタで SMS[3:0] = “1000” (リセットモードとトリガモードの組み合わせ)、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が “1000” または “1001” にセットされています。

タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります (ARR レジスタによってパルス長がセットされます)。タイマをダウンカウントモードで設定した場合、CCRx は ARR 以上である必要があります。

**注 :** OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

このモードをセンターライン PWM モードと組み合わせて使用することはできません。TIMx\_CR1 では、CMS[1:0] = 00 にする必要があります。

図 188. 再トリガ可能なワンパルスモード



## 20.3.21 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx\_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS="001" を、TI1 エッジのみをカウントしている場合は SMS="010" を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS="011" を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx\_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2つの入力 TI1 と TI2 は、直交エンコーダとのインタフェースに使用されます。表 119 を参照してください。カウンタのクロック供給は、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択の後には TI1 と TI2、フィルタされず反転されない場合は TI1FP1=TI1、フィルタされず反転されない場合は TI2FP2=TI2) での有効な遷移ごとに行われます。ただし、カウンタは有効である (TIMx\_CR1 レジスタの CEN ビットに "1" が書き込まれている) ことが前提です。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx\_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx\_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx\_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、繰り返しカウンタ、トリガ出力の機能は通常どおりに機能を続けます。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタは直交エンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (TI1 と TI2 は同時に切り替わらないと想定しています)。

表 119. カウント方向とエンコーダ信号

アクティブ エッジ	他方の信号の レベル (TI2 に 対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみ カウント	ハイ	ダウン	アップ	カウントなし	カウントなし
	ロー	アップ	ダウン	カウントなし	カウントなし
TI2 のみ カウント	ハイ	カウントなし	カウントなし	アップ	ダウン
	ロー	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の 両方をカウント	ハイ	ダウン	アップ	アップ	ダウン
	ロー	アップ	ダウン	ダウン	アップ

直交エンコーダは、外部インターフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 189 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx\_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S="01" (TIMx\_CCMR2 レジスタ、TI1FP2 は TI2 に配置)
- CC1P="0"、CC1NP="0" (TIMx\_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1)
- CC2P="0"、CC2NP="0" (TIMx\_CCER レジスタ、TI1FP2 非反転、TI1FP2=TI2)
- SMS="011" (TIMx\_SMCR レジスタ、両方の入力立ち上がり立ち下がり両エッジでアクティブ)
- CEN="1" (TIMx\_CR1 レジスタ、カウンタ有効)

図 189. エンコーダインターフェースモードにおけるカウンタの動作例

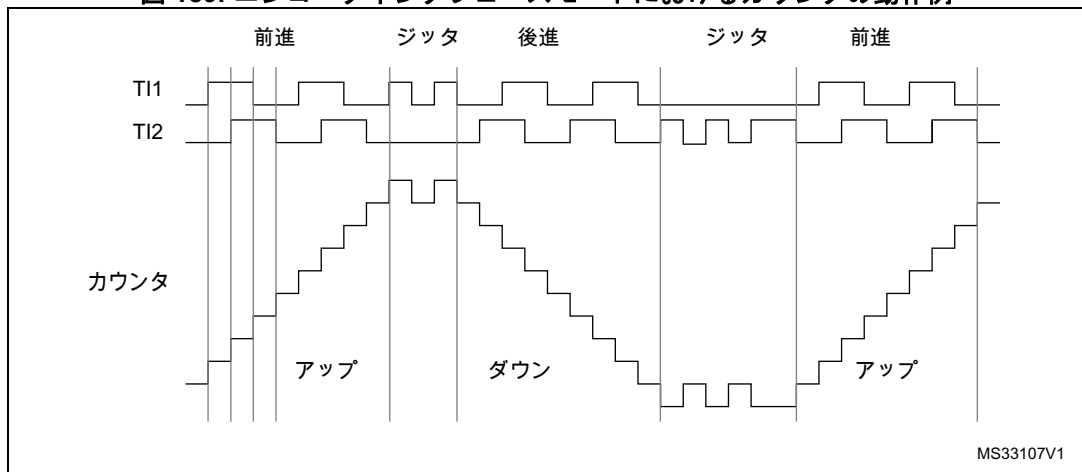
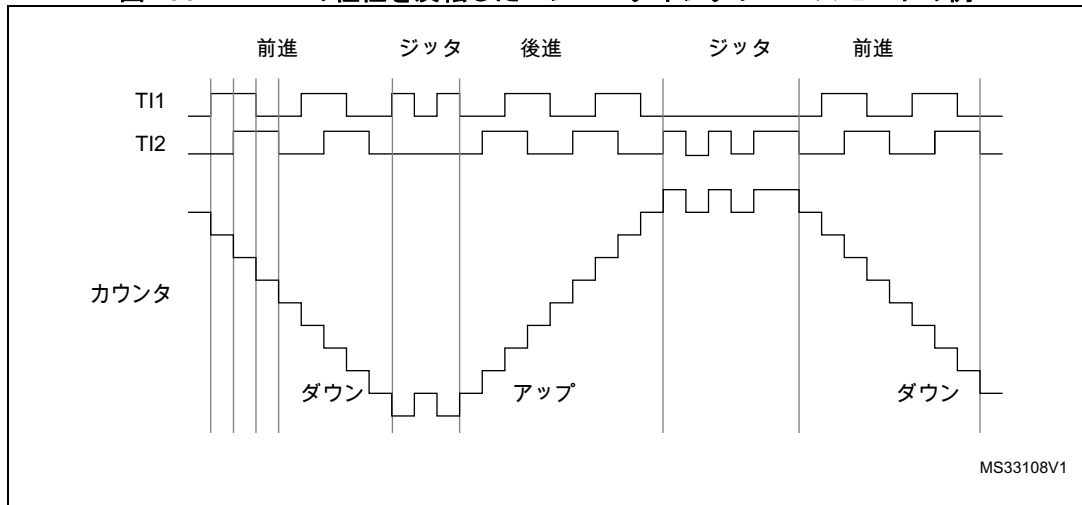


図 190 に、TI1FP1 の極性を反転したときのカウンタの動作を示します (上記と同じ設定ですが、CC1P="1")。

図 190. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします (キャプチャ信号は周期的でなければならず、別のタイマによって生成できます)。使用可能なときには、DMA リクエストを通じて値を読み出すことも可能です。

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。バックグラウンドタスク (カウンタの読み出し) と中断 (更新の中断) との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読み出しアクセス時に UIFCPY フラグによって上書きされます (カウンタの最上位ビットには書き込みモード時のみアクセス可能)。

## 20.3.22 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。特殊なケースでは、バックグラウンドタスク (カウンタの読み出し) と中断 (更新の中断) との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

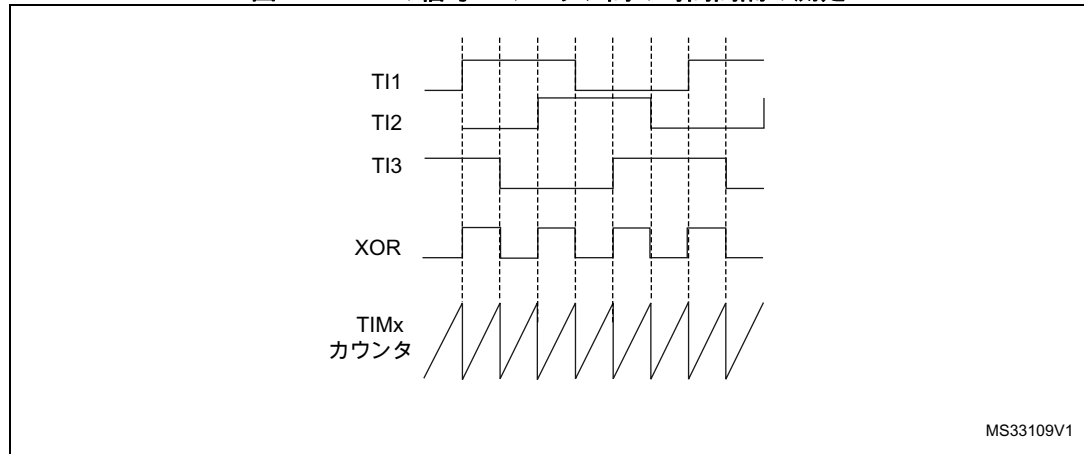
UIF と UIFCPY フラグのアサートの間には、遅延はありません。

## 20.3.23 タイマ入力 XOR 機能

TIMx\_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 3 つの入力ピン TIMx\_CH1、TIMx\_CH2、および TIMx\_CH3 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。次の [図 191](#) に示すように、2 つの入力信号上のエッジ間の間隔を測定するのに便利です。

図 191. 3 つの信号上のエッジ間の時間間隔の測定



MS33109V1

## 20.3.24 ホールセンサとのインターフェース

これは、高機能制御タイマ (TIM1 または TIM8) を使用して PWM 信号を生成し、モータと [図 192](#) で「インタフェースタイマ」と記されている別のタイマ TIMx (TIM2、TIM3、または TIM4) を駆動することによって実現します。「インタフェースタイマ」は、XOR を通じて TI1 入力チャンネル (TIMx\_CR2 レジスタの TI1S ビットをセットすることで選択できます) に接続された 3 つのタイマ入力ピン (CC1、CC2、CC3) をキャプチャします。

スレーブモードコントローラはリセットモードに設定され、スレーブ入力は TI1F\_ED です。したがって、3 つの入力のいずれかが反転するごとに、カウンタは 0 からカウントをリスタートします。これが、ホール入力の変化によってトリガされるタイムベースとなります。

「インタフェースタイマ」上で、キャプチャ/比較チャンネル 1 がキャプチャモードで設定され、キャプチャ信号は TRC です ([図 167: キャプチャ/比較チャンネル \(例: チャンネル 1 入カステージ\) \(524 ページ\)](#) を参照)。キャプチャされた値は、入力の 2 回の変化の間の経過時間に対応し、モータの速度情報を与えます。

「インタフェースタイマ」を出力モードで使用して、(COM イベントをトリガすることで) 高機能制御タイマ (TIM1 または TIM8) のチャンネルの設定を変更するパルスを生成することができます。TIM1 タイマは、モータを駆動する PWM 信号を生成するために使用されます。このためには、プログラミングした遅延の後に正パルスが生成されるように (出力比較モードまたは PWM モードで) インタフェースタイマチャンネルをプログラミングする必要があります。このパルスは、TRGO 出力を通じて高機能制御タイマ (TIM1 または TIM8) に送られます。



例：TIMx タイマの 1 つに接続されているホール入力に変化するたびに、プログラミングした遅延の後に高機能制御タイマ TIM1 の PWM 設定を変更するとします。

- 3 つのタイマ入力を TI1 入力チャンネルに OR 接続します。このためには、TIMx\_CR2 レジスタの TI1S ビットに“1”を書き込みます。
- タイムベースをプログラムします。このためには、TIMx\_ARR に最大値を書き込み、TI1 の変化でカウンタがクリアされるようにします。最大カウンタ時間がセンサの 2 回の変化の間の時間より長くなるように、プリスケアラを設定します。
- チャンネル 1 をキャプチャモード (TRC 選択) にプログラムします。すなわち、TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。必要な場合は、デジタルフィルタをプログラムすることもできます。
- チャンネル 2 を PWM 2 モードにプログラミングし、希望の遅延を指定します。このためには、TIMx\_CCMR1 レジスタの OC2M ビットに“111”を、CC2S ビットに“00”を書き込みます。
- TRGO 上のトリガ出力として OC2REF を選択します。このためには、TIMx\_CR2 レジスタの MMS ビットに“101”を書き込みます。

高機能制御タイマ TIM1 で、トリガ入力として適切な ITR 入力を選択する必要があり、タイマが PWM 信号を生成するようにプログラミングし、キャプチャ/比較制御信号がプリロードされ (TIMx\_CR2 レジスタの CCPC=1)、COM イベントがトリガ入力によって制御されなければなりません (TIMx\_CR2 レジスタの CCUS=1)。PWM 制御ビット (CCxE、OCxM) は、COM イベント後に次のステップのために書き込まれます (これは、OC2REF の立ち上がりエッジによって生成される割り込みサブルーチンで行うことができます)。


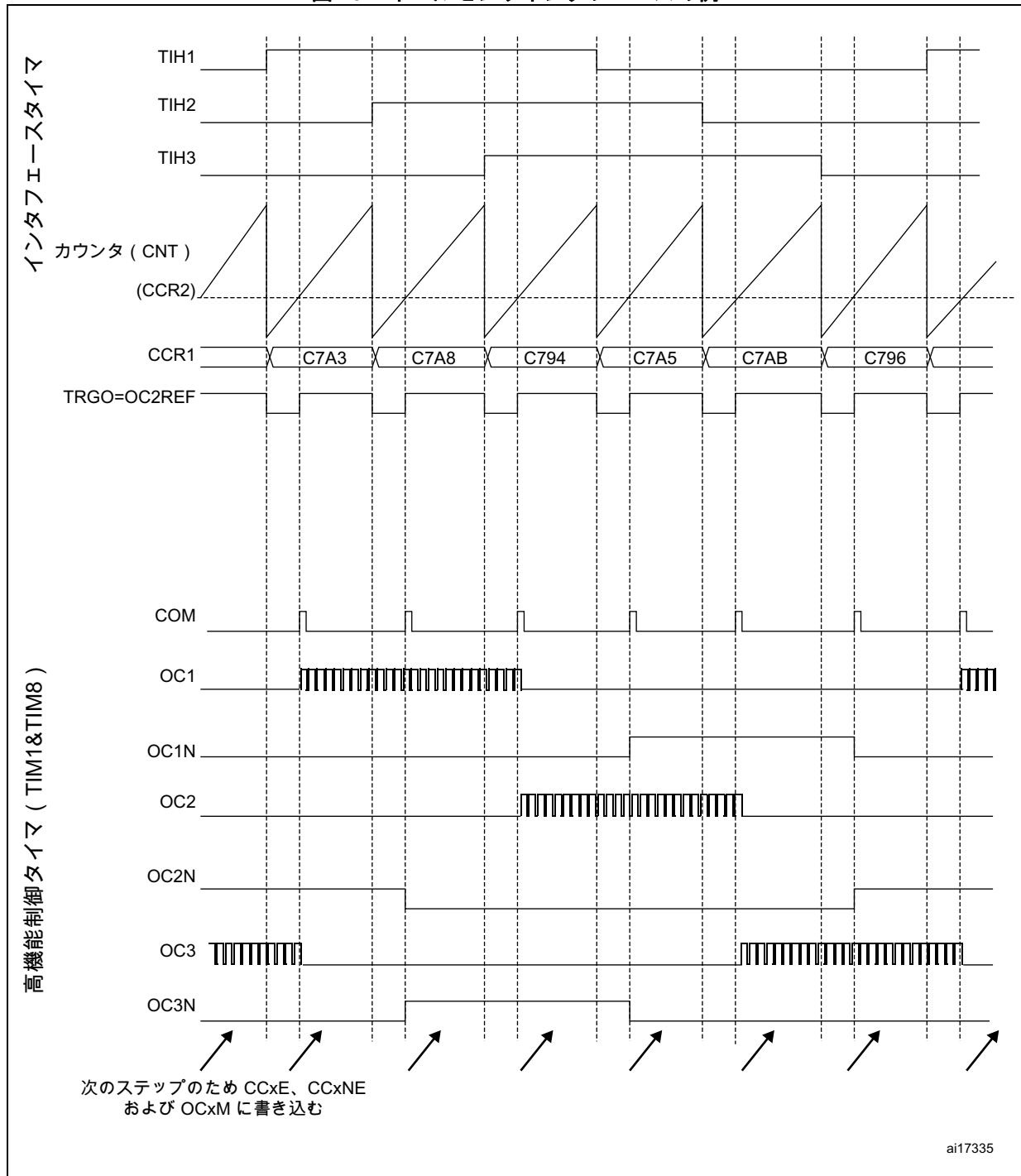
 192 に、この例を示します。

図 192. ホールセンサインタフェースの例



## 20.3.25 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。これらのタイマは、いくつかのモードで同期させることができます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

### スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx\_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx\_ARR、TIMx\_CCRx) が更新されます。

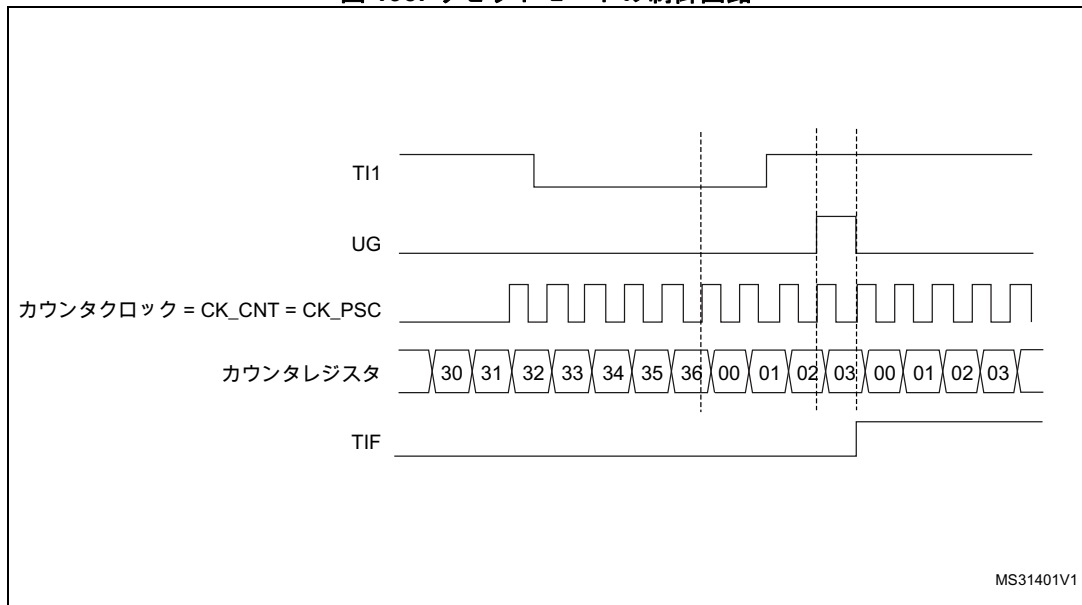
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01)。TIMx\_CCER レジスタに CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx\_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx\_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx\_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx\_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 193. リセットモードの制御回路



## スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

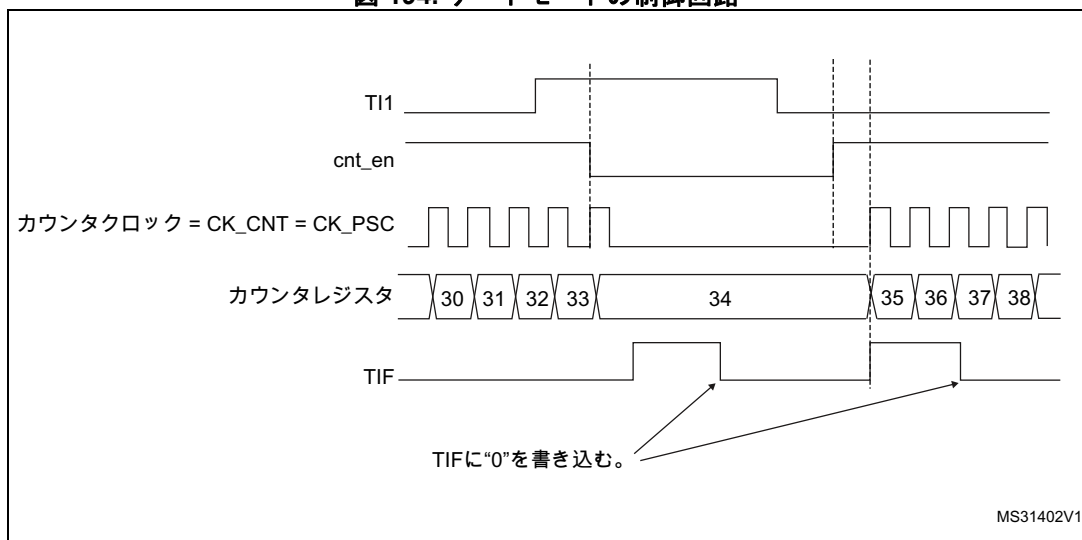
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01 ビット)。TIMx\_CCER レジスタで CC1P=1 と CC1NP=0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx\_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx\_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx\_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 194. ゲートモードの制御回路



## スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

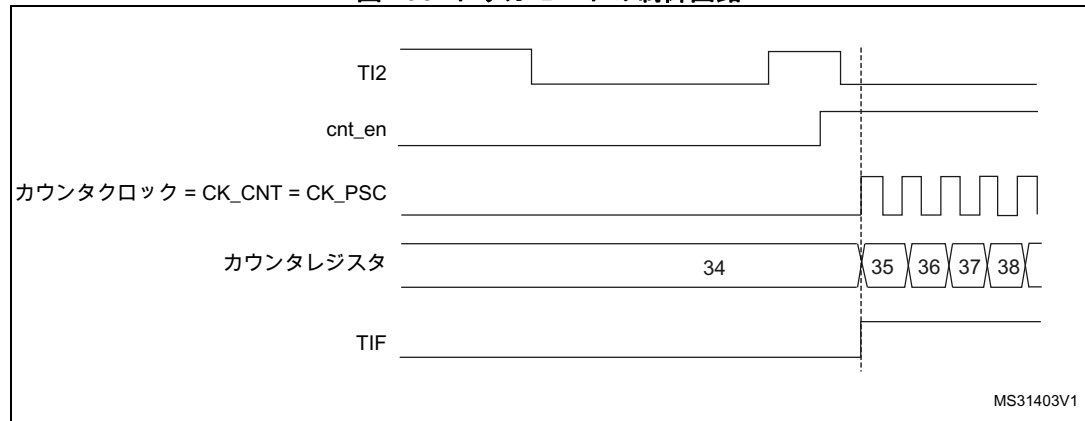
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx\_CCMR1 レジスタの CC2S=01)。TIMx\_CCER レジスタで CC2P=1 と CC2NP=0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 195. トリガモードの制御回路



## スレーブモード：リセットモードとトリガモードの組み合わせ

この場合、選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

## スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます (ただし、外部クロックモード 1 とエンコーダモードは除きます)。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます (リセットモード、ゲートモード、またはトリガモード)。TIMx\_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

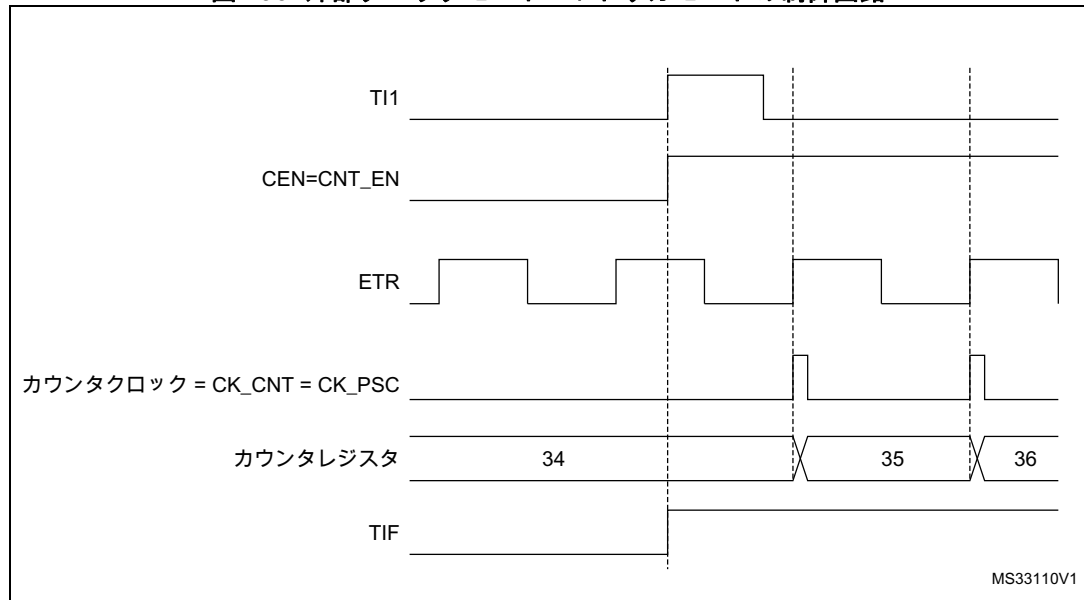
次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

- TIMx\_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
  - ETF = 0000 : フィルタなし
  - ETPS=00 : プリスケール無効
  - ETP=0 : ETR の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
  - IC1F=0000 : フィルタなし。
  - キャプチャプリスケールはトリガには使用されないため、設定する必要はありません。
  - TIMx\_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
  - TIMx\_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 196. 外部クロックモード 2+トリガモードの制御回路



**注：** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

### 20.3.26 ADC の同期

タイマでは、リセットイベント、イネーブルイベント、比較イベントなどのさまざまな内部信号で ADC トリガイベントを生成できます。以下のような内部エッジ検出から発行されたパルスを生成することもできます。

- OC4ref の立ち上がりおよび立ち下がりエッジ
- OC5ref の立ち上がりエッジまたは OC6ref の立ち下がりエッジ

トリガは、ADC にリダイレクトされる TRGO2 内部ラインで発行されます。使用できるイベントは全部で 16 個あり、TIMx\_CR2 レジスタの MMS2[3:0] ビットを使用して選択できます。

3 相モータ駆動のアプリケーションの例については、[図 178 \(537 ページ\)](#) を参照してください。

**注：** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

**注：** ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

## 20.3.27 DMA バーストモード

TIMx タイマには、1つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、タイマの一部をソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx\_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト (バースト) を開始します。TIMx\_DMAR レジスタへの各書き込みは、実際にタイマレジスタの1つにリダイレクトされます。

TIMx\_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数 (ハーフワードまたはバイト) です。

TIMx\_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx\_CR1

00001 : TIMx\_CR2

00010 : TIMx\_SMCR

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
  - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
  - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
  - 転送データ数 = 3 とします (下の注を参照)。
  - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。  
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタのUDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注 :

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

## 20.3.28 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M4<sup>®</sup>F コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

安全のため、カウンタが停止すると (DBG\_TIMx\_STOP = 1)、出力は無効になります (MOE ビットのリセット時と同じ)。通常強制的にハイインピーダンスにするために、出力を強制的にインアクティブ状態にするか (OSSI ビット = 1)、GPIO コントローラで制御することができません (OSSI ビット = 0)。

詳細については、[セクション 33.16.2: タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。



## 20.4 TIM1/TIM8/TIM20 レジスタ

レジスタの説明で使用されている略語のリストを参照してください。

### 20.4.1 TIM1/TIM8/TIM20 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFRE MAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (ETR、Tlx) によって使用されるデッドタイムおよびサンプリングクロック ( $t_{DTS}$ ) との間の分周比を示します。

00 :  $t_{DTS}=t_{CK\_INT}$

01 :  $t_{DTS}=2*t_{CK\_INT}$

10 :  $t_{DTS}=4*t_{CK\_INT}$

11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:5 **CMS[1:0]** : センターアラインモード選択

00 : エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01 : センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10 : センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。

11 : センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャンネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

**注 :** カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

0 : カウンタはアップカウンタとして使用されます。

1 : カウンタはダウンカウンタとして使用されます。

**注 :** このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

### ビット 3 OPM : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

### ビット 2 URS : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。

### ビット 1 UDIS : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

### ビット 0 CEN : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

**注 :** 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

## 20.4.2 TIM1/TIM8/TIM20 制御レジスタ 2 (TIM<sub>x</sub>\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								r/w	r/w	r/w	r/w		r/w		r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **MMS2[3:0]** : マスタモード選択 2

これらのビットにより、選択される同期 (TRGO2) について、ADC に情報を送信できるようになります。組み合わせは、次のとおりです。

0000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO2) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO2 信号は実際のリセットより遅延します。

0001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO2) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO2 には遅延が存在します (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

0010 : **更新** - 更新イベントがトリガ出力 (TRGO2) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。

0011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる (すでにハイであった場合も)、トリガ出力は正のパルスを送信します (TRGO2)。

0100 : **比較** - OC1REF 信号がトリガ出力 (TRGO2) として使用されます。

0101 : **比較** - OC2REF 信号がトリガ出力 (TRGO2) として使用されます。

0110 : **比較** - OC3REF 信号がトリガ出力 (TRGO2) として使用されます。

0111 : **比較** - OC4REF 信号がトリガ出力 (TRGO2) として使用されます。

1000 : **比較** - OC5REF 信号がトリガ出力 (TRGO2) として使用されます。

1001 : **比較** - OC6REF 信号がトリガ出力 (TRGO2) として使用されます。

1010 : **パルス比較** - OC4REF の立ち上がりエッジまたは立ち下がりエッジによって、TRGO2 にパルスが生成されます。

1011 : **パルス比較** - OC6REF の立ち上がりエッジまたは立ち下がりエッジによって、TRGO2 にパルスが生成されます。

1100 : **パルス比較** - OC4REF または OC6REF の立ち上がりエッジによって、TRGO2 にパルスが生成されます。

1101 : **パルス比較** - OC4REF の立ち上がりエッジまたは OC6REF の立ち下がりエッジによって、TRGO2 にパルスが生成されます。

1110 : **パルス比較** - OC5REF または OC6REF の立ち上がりエッジによって、TRGO2 にパルスが生成されます。

1111 : **パルス比較** - OC5REF の立ち上がりエッジまたは OC6REF の立ち下がりエッジによって、TRGO2 にパルスが生成されます。

**注 :** **スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。**

ビット 19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **OIS6** : 出力アイドル状態 6 (OC6 出力)

OIS1 ビットの説明を参照してください。

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **OIS5** : 出力アイドル状態 5 (OC5 出力)

OIS1 ビットの説明を参照してください。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **OIS4** : 出力アイドル状態 4 (OC4 出力)

OIS1 ビットの説明を参照してください。

ビット 13 **OIS3N** : 出力アイドル状態 3 (OC3N 出力)

OIS1N ビットの説明を参照してください。

ビット 12 **OIS3** : 出力アイドル状態 3 (OC3 出力)

OIS1 ビットの説明を参照してください。



- ビット 11 **OIS2N** : 出力アイドル状態 2 (OC2N 出力)  
OIS1N ビットの説明を参照してください。
- ビット 10 **OIS2** : 出力アイドル状態 2 (OC2 出力)  
OIS1 ビットの説明を参照してください。
- ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)  
0 : MOE=0 のとき、デッドタイム後に OC1N=0  
1 : MOE=1 のとき、デッドタイム後に OC1N=0  
**注 :** このビットは、**LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。**
- ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)  
0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)  
1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)  
**注 :** このビットは、**LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。**
- ビット 7 **TI1S** : TI1 選択  
0 : TIMx\_CH1 ピンが TI1 入力に接続されます。  
1 : TIMx\_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。
- ビット 6:4 **MMS[1:0]** : マスタモード選択  
これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。  
000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。  
001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、GEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在しません (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。  
010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。  
011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる時 (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。(TRGO)  
100 : 比較 - OC1REF 信号がトリガ出力 (TRGO) として使用されます。  
101 : 比較 - OC2REF 信号がトリガ出力 (TRGO) として使用されます。  
110 : 比較 - OC3REF 信号がトリガ出力 (TRGO) として使用されます。  
111 : 比較 - OC4REF 信号がトリガ出力 (TRGO) として使用されます。  
**注 :** **スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。**
- ビット 3 **CCDS** : キャプチャ/比較 DMA 選択  
0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。  
1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ/比較制御更新選択

0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。

1 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ/比較プリロード制御

0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。

1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込みの後、これらのビットは、転換イベント (COM) が発生した時にのみ更新されます (CCUS ビットに応じて、COMG ビットがセットまたは TRGI で立ち上がりエッジが検出されたとき)。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

## 20.4.3 TIM1/TIM8/TIM20 スレーブモード制御レジスタ (TIMx\_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]		OCCS	SMS[2:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SMS[3]** : スレーブモード選択 - ビット 3

SMS 説明を参照 - ビット 2:0

ビット 15 **ETP** : 外部トリガ極性

このビットは、ETR と  $\overline{\text{ETR}}$  のいずれがトリガ動作に使用されるかを選択します。

0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。

1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブです。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

0 : 外部クロックモード 2 は無効です。

1 : 外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

**注 :** 1 : ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=111)。

2 : 外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません (TS ビットが 111 でないことが必要)。

3 : 外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力は ETRF です。

### ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、TIMxCLK 周波数の 1/4 までに制限されます。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。

- 00 : プリスケアラオフ
- 01 : ETRP 周波数は 2 分周されます。
- 10 : ETRP 周波数は 4 分周されます。
- 11 : ETRP 周波数は 8 分周されます。

### ビット 11:8 **ETF[3:0]** : 外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

- 0001 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2
- 0010 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4
- 0011 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8
- 0100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2
- 0101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2
- 0110 :  $f_{SAMPLING} = f_{DTS}/4$ 、N = 6
- 0111 :  $f_{SAMPLING} = f_{DTS}/4$ 、N = 8
- 1000 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8
- 1001 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8
- 1010 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16
- 1011 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16
- 1100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16
- 1101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32
- 1110 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32
- 1111 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

### ビット 7 **MSM** : マスタ/スレーブモード

- 0 : 影響なし。
- 1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

### ビット 6:4 **TS[2:0]** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)
- 100 : TI1 エッジ検出回路 (TI1F\_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)
- 111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、[表 120 : TIMx 内部トリガ接続 \(567 ページ\)](#) を参照してください。

**注 :** **設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。**

### ビット 3 **OCCS** : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

- 0 : OCREF\_CLR\_INT は、OCREF\_CLR 入力に接続されています。
- 1 : OCREF\_CLR\_INT は、ETRF に接続されています。

ビット 2:0 SMS : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001 : エンコーダモード 1 - カウンタは、TI2FP2 のレベルに応じて、TI1FP1 のエッジでカウントアップ/ダウンします。

0010 : エンコーダモード 2 - カウンタは、TI1FP1 のレベルに応じて、TI2FP2 のエッジでカウントアップ/ダウンします。

0011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。

0100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

1000 以上のコード : 予約済み。

**注 :** トリガ入力として TI1F\_ED が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F\_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

**注 :** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 120. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM15	TIM2	TIM3	TIM4 または TIM17 <sup>(1)</sup>
TIM8	TIM1	TIM2	TIM4	TIM3
TIM20	TIM1	TIM8	TIM4	TIM15

1. TIM1\_ITR3 の選択は、SYSCFG\_CFGR1 レジスタのビット 6 を使用して行います。

## 20.4.4 TIM1/TIM8/TIM20 DMA/割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE	
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル  
0 : トリガ DMA リクエストは無効です。  
1 : トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル  
0 : COM DMA リクエストは無効です。  
1 : COM DMA リクエストは有効です。

ビット 12 **CC4DE** : キャプチャ/比較 4 DMA リクエストイネーブル  
0 : CC4 DMA リクエストは無効です。  
1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ/比較 3 DMA リクエストイネーブル  
0 : CC3 DMA リクエストは無効です。  
1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ/比較 2 DMA リクエストイネーブル  
0 : CC2 DMA リクエストは無効です。  
1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ/比較 1 DMA リクエストイネーブル  
0 : CC1 DMA リクエストは無効です。  
1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル  
0 : 更新 DMA リクエストは無効です。  
1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割り込みイネーブル  
0 : ブレーク割り込みは無効です。  
1 : ブレーク割り込みは有効です。

ビット 6 **TIE** : トリガ割り込みイネーブル  
0 : トリガ割り込みは無効です。  
1 : トリガ割り込みは有効です。

ビット 5 **COMIE** : COM 割り込みイネーブル  
0 : COM 割り込みは無効です。  
1 : COM 割り込みは有効です。

ビット 4 **CC4IE** : キャプチャ/比較 4 割り込みイネーブル  
0 : CC4 割り込みは無効です。  
1 : CC4 割り込みは有効です。

ビット 3 **CC3IE** : キャプチャ/比較 3 割り込みイネーブル  
0 : CC3 割り込みは無効です。  
1 : CC3 割り込みは有効です。

ビット 2 **CC2IE** : キャプチャ/比較 2 割り込みイネーブル  
0 : CC2 割り込みは無効です。  
1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル  
0 : CC1 割り込みは無効です。  
1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル  
0 : 更新割り込みは無効です。  
1 : 更新割り込みは有効です。



## 20.4.5 TIM1/TIM8/TIM20 ステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	B1F	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CC6IF** : 比較 6 割り込みフラグ

CC1IF の説明を参照してください。(注 : チャネル 6 は出力としてのみ設定できます。)

ビット 16 **CC5IF** : 比較 5 割り込みフラグ

CC1IF の説明を参照してください。(注 : チャネル 5 は出力としてのみ設定できます。)

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8 **B2IF** : ブレーク 2 割り込みフラグ

このフラグは、ブレーク 2 入力が入力アクティブになると、ハードウェアによってセットされます。ブレーク 2 入力が入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク 2 入力が入力アクティブレベルが検出されました。TIMx\_DIER レジスタの BIE=1 の場合、割り込みが生成されます。

ビット 7 **B1F** : ブレーク割り込みフラグ

このフラグは、ブレーク入力が入力アクティブになると、ハードウェアによってセットされます。ブレーク入力が入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク入力が入力アクティブレベルが検出されました。TIMx\_DIER レジスタの BIE=1 の場合、割り込みが生成されます。

### ビット 6 TIF : トリガ割り込みフラグ

このフラグは、トリガイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされず、ソフトウェアによってクリアされます。

- 0 : トリガイベントは発生していません。
- 1 : トリガ割り込みが保留中です。

### ビット 5 COMIF : COM 割り込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE, CCxNE, OCxM - が更新されたとき)。ソフトウェアによってクリアされます。

- 0 : COM イベントは発生していません。
- 1 : COM 割り込みがペンディング中です。

### ビット 4 CC4IF : キャプチャ/比較 4 割り込みフラグ

CC1IF の説明を参照してください。

### ビット 3 CC3IF : キャプチャ/比較 3 割り込みフラグ

CC1IF の説明を参照してください。

### ビット 2 CC2IF : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

### ビット 1 CC1IF : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合** : このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます (センターアラインモードでは、例外もあります。TIMx\_CR1 レジスタの CMS ビットの説明を参照してください)。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。TIMx\_CCR1 の内容が TIMx\_ARR の内容より大きいときには、カウンタオーバーフロー時 (アップカウントおよびアップ/ダウンカウントモードの場合)、またはアンダーフロー時 (ダウンカウントモードの場合) に CC1IF ビットはハイになります。

**CC1 チャンネルが入力として設定されている場合** : このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

### ビット 0 UIF : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフローまたはアンダーフロー (繰り返しカウンタ=0 の場合の更新)、および TIMx\_CR1 レジスタの UDIS=0 の場合。
- TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき (セクション 20.4.3: TIM1/TIM8/TIM20 スレーブモード制御レジスタ (TIMx\_SMCR) を参照)。

## 20.4.6 TIM1/TIM8/TIM20 イベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **B2G** : ブレーク 2 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレーク 2 イベントが生成されます。MOE ビットがクリアされ、B2IF フラグがセットされます。有効な場合は、関連する割り込が発生します。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx\_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

**注 :** このビットは、相補出力を持つチャネルでのみ機能します。

ビット 4 **CC4G** : キャプチャ / 比較 4 イベント生成

CC1G の説明を参照してください。

ビット 3 **CC3G** : キャプチャ / 比較 3 イベント生成

CC1G の説明を参照してください。

ビット 2 **CC2G** : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

### ビット 1 CC1G : キャプチャ/比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

#### CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

#### CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

### ビット 0 UG : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx\_ARR) をとります。

## 20.4.7 TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]			IC2PSC[1:0]			IC1F[3:0]			IC1PSC[1:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

### 出力比較モード :

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OC2M[3]** : 出力比較 2 モード - ビット 3

ビット 14:12 の OC2M 説明を参照

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

Bits16 **OC1M[3]** : 出力比較 1 モード - ビット 3

ビット 6:4 の OC1M 説明を参照

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC2S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。**

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : OC1Ref は ocref\_clr\_int の影響を受けません。

1 : OC1Ref は ocref\_clr\_int のハイレベルが検出されるとクリアされます (OCREF\_CLR 入力または ETRF 入力)。

### ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

0001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

0010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

0011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

0100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

0101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 OR です。

1101 : 組み合わせ PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 AND です。

1110 : 非対称 PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

1111 : 非対称 PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

**注 :** これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

**注 :** PWM モードでは、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとりまします。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注 :** 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2: PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** CC2S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。

ビット 7:4 IC1F[3:0] : 入力キャプチャ 1 フィルタ

このビットフィールドは、T11 入力をサンプリングする周波数と、T11 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

0001 :  $f_{SAMPLING} = f_{CK\_INT}$ , N = 2

0010 :  $f_{SAMPLING} = f_{CK\_INT}$ , N = 4

0011 :  $f_{SAMPLING} = f_{CK\_INT}$ , N = 8

0100 :  $f_{SAMPLING} = f_{DTS}$ , N = 2

0101 :  $f_{SAMPLING} = f_{DTS}$ , N = 2

0110 :  $f_{SAMPLING} = f_{DTS}/4$ , N = 6

0111 :  $f_{SAMPLING} = f_{DTS}/4$ , N = 8

1000 :  $f_{SAMPLING} = f_{DTS}$ , N = 8

1001 :  $f_{SAMPLING} = f_{DTS}$ , N = 8

1010 :  $f_{SAMPLING} = f_{DTS}$ , N = 16

1011 :  $f_{SAMPLING} = f_{DTS}$ , N = 16

1100 :  $f_{SAMPLING} = f_{DTS}$ , N = 16

1101 :  $f_{SAMPLING} = f_{DTS}$ , N = 32

1110 :  $f_{SAMPLING} = f_{DTS}$ , N = 32

1111 :  $f_{SAMPLING} = f_{DTS}$ , N = 32

ビット 3:2 IC1PSC : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E = 0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は T11 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は T12 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

## 20.4.8 TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 2 (TIMx\_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

上記の CCMR1 レジスタの説明を参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							r/w								r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE.	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w



## 出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OC4M[3]** : 出力比較 4 モード - ビット 3

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **OC3M[3]** : 出力比較 3 モード - ビット 3

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC3S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケール

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。



ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケール

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC3S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときのみ書き込み可能です。

## 20.4.9 TIM1/TIM8/TIM20 キャプチャ/比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										r/w	r/w			r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **CC6P** : キャプチャ/比較 6 出力極性

CC1P の説明を参照してください。

ビット 20 **CC6E** : キャプチャ/比較 6 出カインーブル

CC1E の説明を参照してください。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CC5P** : キャプチャ/比較 5 出力極性

CC1P の説明を参照してください。

ビット 16 **CC5E** : キャプチャ/比較 5 出カインーブル

CC1E の説明を参照してください。

ビット 15 **CC4NP** : キャプチャ/比較 4 相補出力極性

CC1NP の説明を参照してください。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC4P** : キャプチャ/比較 4 出力極性

CC1P の説明を参照してください。

ビット 12 **CC4E** : キャプチャ/比較 4 出カインーブル

CC1E の説明を参照してください。

ビット 11 **CC3NP** : キャプチャ/比較 3 相補出力極性

CC1NP の説明を参照してください。

ビット 10 **CC3NE** : キャプチャ/比較 3 相補出カインーブル

CC1NE の説明を参照してください。

ビット 9 **CC3P** : キャプチャ/比較 3 出力極性  
CC1P の説明を参照してください。

ビット 8 **CC3E** : キャプチャ/比較 3 出カインーブル  
CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ/比較 2 相補出力極性  
CC1NP の説明を参照してください。

ビット 6 **CC2NE** : キャプチャ/比較 2 相補出カインーブル  
CC1NE の説明を参照してください。

ビット 5 **CC2P** : キャプチャ/比較 2 出力極性  
CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ/比較 2 出カインーブル  
CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性  
**CC1 チャンネルが出力として設定されている場合 :**  
0 : OC1N はアクティブハイです。  
1 : OC1N はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :**  
このビットは、TI1FP1とTI2FP1の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

**注 :** このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 2 **CC1NE** : キャプチャ/比較 1 相補出カインーブル  
0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。  
1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1NE アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

**CC1 チャンネルが出力として設定されている場合 :**

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :** CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

00 : 非反転/立ち上がりエッジ。この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

01 : 反転/立ち下がりエッジ。この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ。この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。この設定をエンコーダモードで使用することはできません。

**注 :** このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx\_BDTR レジスタの LOCK ビット)。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル

**CC1 チャンネルが出力として設定されている場合 :**

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :** このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

**注 :** 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1E アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

表 121. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット

制御ビット					出力状態 <sup>(1)</sup>	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	X	0	0	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=0、OCxN=0	
		0	0	1	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP
		0	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP	出力無効 (タイマによって駆動され ない: ハイインピーダンス) OCxN=0
		X	1	1	OCREF + 極性 + デッドタイム	OCREF に対する相補 (OCREF では なく) + 極性 + デッドタイム
		1	0	1	オフ状態 (インアクティブ状態 で出力有効) OCx=CCxP	OCxREF + 極性 OCxN = OCxREF xor CCxNP
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP	オフ状態 (インアクティブ状態で出 力有効) OCxN=CCxNP
0	1	X	X	X	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=CCxP、OCxN=CCxNP	
			0	0		
			0	1	オフ状態 (インアクティブ状態で出力有効)	
			1	0	非同期: OCx=CCxP、OCxN=CCxNP (BRK または BRK2 がトリガ された場合)	
			1	1	クロックがある場合 (これは BRK がトリガされている場合にのみ有 効): アクティブな状態で OISx と OISxN が OCx と OCxN にそれぞ れ対応しないとみなされる場合 (ハーフブリッジ設定でスイッチを 駆動した場合に短絡の原因となる)、デッドタイム後に OCx=OISx お よび OCxN=OISxN となります。 注: BRK2 は OSSI = OSSR = 1 の場合にのみ使用できます。	

1. チャンネルの両方の出力が使用されないとき (GPIO が制御を引き継いだ場合)、OISx、OISxN、CCxP、および CCxNP はクリアされたまま  
でなければなりません。

注: 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OCx および OCxN  
チャンネルの状態と、GPIO レジスタに依存します。

## 20.4.10 TIM1/TIM8/TIM20 カウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー。TIMxCR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

## 20.4.11 TIM1/TIM8/TIM20 プリスケーラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **PSC[15:0]** : プリスケーラ値

カウンタクロック周波数 (CK\_CNT) は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケーラレジスタにロードされる値を含みます (更新イベントには、TIMx\_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

## 20.4.12 TIM1/TIM8/TIM20 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **ARR[15:0]** : プリスケーラ値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 20.3.1 : タイムベースユニット \(504 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 20.4.13 TIM1/TIM8/TIM20 繰り返しカウンタレジスタ (TIMx\_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REP[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 REP[15:0] : 繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセッティングできます。

REP\_CNTに関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U\_RC のみ、REP\_CNT に REP 値がロードされるので、TIMx\_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) は次のことを意味します。

エッジアラインモードでは、PWM 周期の数

センターアラインモードでは、PWM の 1/2 周期の数

## 20.4.14 TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CCR1[15:0] : キャプチャ/比較 1 値

**CC1 チャンネルが出力として設定されている場合** : CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値（プリロード値）です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**CC1 チャンネルが入力として設定されている場合** : CR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 20.4.15 TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 2 (TIMx\_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW



ビット 15:0 **CCR2[15:0]** : キャプチャ/比較 2 値

**CC2 チャンネルが出力として設定されている場合** : CCR2 は、実際のキャプチャ/比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 2 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC2 出力に送信される値を含みます。

**CC2 チャンネルが入力として設定されている場合** : CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

### 20.4.16 TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 3 (TIMx\_CCR3)

アドレスオフセット : 0x3C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR3[15:0]** : キャプチャ / 比較値

**CC3 チャンネルが出力として設定されている場合** : CCR3 は、実際のキャプチャ/比較 3 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 3 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC3 出力に送信される値を含みます。

**CC3 チャンネルが入力として設定されている場合** : CCR3 は、最後の入力キャプチャ 3 イベント (IC3) によって転送されたカウンタ値です。

### 20.4.17 TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 4 (TIMx\_CCR4)

アドレスオフセット : 0x40

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



ビット 15:0 **CCR4[15:0]** : キャプチャ / 比較値

**CC4 チャンネルが出力として設定されている場合** : CCR4 は、実際のキャプチャ / 比較 4 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ / 比較 4 レジスタにコピーされます。

アクティブキャプチャ / 比較レジスタは、カウンタ TIMx\_CNT と比較され、OC4 出力に送信される値を含みます。

**CC4 チャンネルが入力として設定されている場合** : CCR4 は、最後の入力キャプチャ 4 イベント (IC4) によって転送されたカウンタ値です。

## 20.4.18 TIM1/TIM8/TIM20 ブレークおよびデッドタイムレジスタ (TIMx\_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

**注 :** ビット **BK2P**、**BK2E**、**BK2F[3:0]**、**BKF[3:0]**、**AOE**、**BKP**、**BKE**、**OSSI**、**OSSR**、および **DTG[7:0]** は、**LOCK** 設定に応じて書き込みがロックされるので、**TIMx\_BDTR** レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **BK2P** : ブレーク 2 極性

0 : ブレーク入力 BRK2 はアクティブラーです。

1 : ブレーク入力 BRK2 はアクティブハイです。

**注 :** このビットは、**LOCK** レベル 1 がプログラムされている場合 (**TIMx\_BDTR** レジスタの **LOCK** ビット)、変更できません。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 24 **BK2E** : ブレーク 2 イネーブル

0 : ブレーク入力 BRK2 は無効です。

1 : ブレーク入力 BRK2 は有効です。

**注 :** **BRK2** は **OSSR = OSSI = 1** の場合にのみ使用してください。

**注 :** このビットは、**LOCK** レベル 1 がプログラムされているときには変更できません (**TIMx\_BDTR** レジスタの **LOCK** ビット)。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

### ビット 23:20 BK2F[3:0] : ブレーク 2 フィルタ

このビットフィールドは、BRK2 入力をサンプリングする周波数と、BRK2 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、BRK2 は非同期として動作します。

0001 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 2

0010 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 4

0011 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 8

0100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 2

0101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 2

0110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N=6

0111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N=8

1000 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1001 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1010 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1011 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

### ビット 19:16 BKF[3:0] : ブレークフィルタ

このビットフィールドは、BRK 入力をサンプリングする周波数と、BRK に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、BRK は非同期として動作します。

0001 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 2

0010 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 4

0011 :  $f_{\text{SAMPLING}} = f_{\text{CK\_INT}}$ 、N = 8

0100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 2

0101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 2

0110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N=6

0111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ 、N=8

1000 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1001 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 8

1010 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1011 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1100 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 16

1101 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1110 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

1111 :  $f_{\text{SAMPLING}} = f_{\text{DTS}}$ 、N = 32

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

ビット 15 MOE : メイン出力イネーブル

このビットは、ブレーク入力の 1 つがアクティブになると、ハードウェアによって非同期にクリアされます (BRK または BRK2)。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : ブレーク 2 イベントへの対応。OC および OCN 出力は無効です。

ブレークイベントへの対応、または MOE が 0 に書き込まれた場合 : OC および OCN 出力が無効か、OSSI ビットによって強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx\_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 20.4.9 : TIM1/TIM8/TIM20 キャプチャ/比較有効レジスタ \(TIMx\\_CCER\)](#))。

ビット 14 AOE : 自動出力イネーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力 BRK および BRK2 のいずれもがアクティブでない場合)。

**注 :** このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 BKP : ブレーク極性

0 : ブレーク入力 BRK はアクティブローです。

1 : ブレーク入力 BRK はアクティブハイです。

**注 :** このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 BKE : ブレークイネーブル

0 : ブレーク入力 (BRK および CCS クロック障害イベント) は無効です。

1 : ブレーク入力 (BRK および CCS クロック障害イベント) は有効です。

**注 :** このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 20.4.9 : TIM1/TIM8/TIM20 キャプチャ/比較有効レジスタ \(TIMx\\_CCER\)](#))。

0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する GPIO ロジックによって引き継がれます)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります (出力は引き続きタイマで制御される)。

**注 :** このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

### ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE = 0 のとき、ブレークイベントまたはソフトウェアの書き込みにより、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 20.4.9 : TIM1/TIM8/TIM20 キャプチャ/比較有効レジスタ \(TIMx\\_CCER\)](#))。

0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する GPIO ロジックによって引き継がれます)。

1 : インアクティブのとき、OC/OCN 出力はまず強制的にインアクティブレベルにされ、次にデッドタイム後に強制的にアイドルレベルにされます。タイマは出力の制御を保持します。

**注 :** このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

### ビット 9:8 LOCK[1:0] : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 = TIMx\_BDTR レジスタの DTG ビット、TIMx\_CR2 レジスタの OISx および OISxN ビット、および TIMx\_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込めなくなります。

11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCMR レジスタの OCxM および OCxPE ビット) が書き込めなくなります。

**注 :** LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx\_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

### ビット 7:0 DTG[7:0] : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ , ここで  $t_{dtg}=t_{DTS}$ .

$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ , ここで  $T_{dtg}=2 \times t_{DTS}$ .

$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ , ここで  $T_{dtg}=8 \times t_{DTS}$ .

$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ , ここで  $T_{dtg}=16 \times t_{DTS}$ .

例 :  $T_{DTS}=125ns$  (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16  $\mu s$  から 31750 ns (250 ns 単位)

32  $\mu s$  から 63  $\mu s$  (1  $\mu s$  単位)

64  $\mu s$  から 126  $\mu s$  (2  $\mu s$  単位)

**注 :** このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

## 20.4.19 TIM1/TIM8/TIM20 DMA 制御レジスタ (TIMx\_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送長 (タイマは、TIMx\_DMAR アドレスに対して読み出しましたは書き込みアクセスが行われるときにバースト転送を認識します)、つまり転送回数を指定します。転送は、ハーフワードまたはバイトです (以下の例を参照)。

- 00000 : 1 回転送
- 00001 : 2 回転送
- 00010 : 3 回転送
- ...
- 10001 : 18 回転送

例 : 次の転送を考えます : DBL = 7 バイトかつ DBA = TIM2\_CR1。

- DBL = 7 バイトおよび DBA = TIM2\_CR1 が転送するバイトのアドレスを表す場合、転送のアドレスは次の式で与えられます。

(TIMx\_CR1 アドレス) + DBA + (DMA インデックス)、ここで DMA インデックス = DBL

この例では、(TIMx\_CR1 アドレス) + DBA に 7 バイトが追加され、データのコピー元/コピー先アドレスが与えられます。この場合、転送は、以下のアドレスから始めて、7 つのレジスタに対して行われます。(TIMx\_CR1 アドレス) + DBA

DMA データサイズの設定に応じて、いくつかのケースが想定されます。

- DMA データサイズをハーフワードで設定した場合、7 つのレジスタにそれぞれ 16 ビットのデータが転送されます。
- DMA データサイズをバイトで設定した場合も、データは 7 つのレジスタに転送されます (最初のレジスタには最初の MSB バイトが含まれ、2 番目のレジスタには最初の LSB バイトが含まれるなど、以降同様)。タイマへの転送で、DMA によって転送されるデータサイズを指定する必要もあります。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

- 00000 : TIMx\_CR1、
- 00001 : TIMx\_CR2、
- 00010 : TIMx\_SMCR
- ...

## 20.4.20 完全転送の TIM1/TIM8/TIM20 DMA アドレス (TIMx\_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMAB[15:0]															
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しましたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります : (TIMx\_CR1 アドレス) + (DBA + DMA インデックス) x 4。

ここで、TIMx\_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx\_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx\_DCR 内で設定)。



## 20.4.21 TIM1/TIM8/TIM20 オプションレジスタ (TIMx\_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_ETR_ADC4_RMP または TIM8_ETR_ADC3_RMP または TIM20_ETR_ADC4_RMP	TIM1_ETR_ADC1_RMP または TIM8_ETR_ADC2_RMP または TIM20_ETR_ADC3_RMP		
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3:2 **TIM1\_ETR\_ADC4\_RMP[1:0]** : TIM1\_ETR\_ADC4 再配置機能

- 00 : TIM1\_ETR は、どの AWD (アナログウォッチドッグ) にも接続されていません。
- 01 : TIM1\_ETR が ADC4 AWD1 入力に接続されます。
- 10 : TIM1\_ETR が ADC4 AWD2 入力に接続されます。
- 11 : TIM1\_ETR が ADC4 AWD3 入力に接続されます。

**注 :** ADC4 AWD とほかの TIM1\_ETR ソース信号の「論理和」がとられます。そのため、ソフトウェアのその他のソース (入力ピン) を無効にする必要があります。

**TIM8\_ETR\_ADC3\_RMP[1:0]** : TIM8\_ETR\_ADC3 再配置機能

- 00 : TIM8\_ETR は、どの AWD (アナログウォッチドッグ) にも接続されていません。
- 01 : TIM8\_ETR が ADC3 AWD1 に接続されます。
- 10 : TIM8\_ETR が ADC3 AWD2 に接続されます。
- 11 : TIM8\_ETR が ADC3 AWD3 に接続されます。

**注 :** ADC3 AWD とほかの TIM8\_ETR ソース信号の「論理和」がとられます。そのため、ソフトウェアのその他のソース (入力ピン) を無効にする必要があります。

**TIM20\_ETR\_ADC4\_RMP[1:0]** : TIM20\_ETR\_ADC4 再配置機能

- 00 : TIM20\_ETR は、どの AWD (アナログウォッチドッグ) にも接続されていません。
- 01 : TIM20\_ETR が ADC4 AWD1 入力に接続されます。
- 10 : TIM20\_ETR が ADC4 AWD2 入力に接続されます。
- 11 : TIM20\_ETR が ADC4 AWD3 入力に接続されます。

**注 :** ADC4 AWD とほかの TIM20\_ETR ソース信号の「論理和」がとられます。そのため、ソフトウェアのその他のソース (入力ピン) を無効にする必要があります。

ビット 1:0 **TIM1\_ETR\_ADC1\_RMP[1:0]** : TIM1\_ETR\_ADC1 再配置機能

- 00 : TIM1\_ETR は、どの AWD にも接続されていません。
- 01 : TIM1\_ETR が ADC1 AWD1 入力に接続されます。
- 10 : TIM1\_ETR が ADC1 AWD2 入力に接続されます。
- 11 : TIM1\_ETR が ADC1 AWD3 入力に接続されます。

**注 :** **ADC1 AWD とほかの TIM1\_ETR ソース信号の「論理和」がとられます。そのため、ソフトウェアのその他のソース (入力ピン) を無効にする必要があります。**

**TIM8\_ETR\_ADC2\_RMP[1:0]** : TIM8\_ETR\_ADC2 再配置機能

- 00 : TIM8\_ETR は、どの AWD にも接続されていません。
- 01 : TIM8\_ETR が ADC2 AWD1 入力に接続されます。
- 10 : TIM8\_ETR が ADC2 AWD2 入力に接続されます。
- 11 : TIM8\_ETR が ADC2 AWD3 入力に接続されます。

**注 :** **ADC3 AWD とほかの TIM20\_ETR ソース信号の「論理和」がとられます。そのため、ソフトウェアのその他のソース (入力ピン) を無効にする必要があります。**

**TIM20\_ETR\_ADC3\_RMP[1:0]** : TIM20\_ETR\_ADC3 再配置機能

- 00 : TIM20\_ETR は、どの AWD (アナログウォッチドッグ) にも接続されていません。
- 01 : TIM20\_ETR が ADC3 AWD1 入力に接続されます。
- 10 : TIM20\_ETR が ADC3 AWD2 入力に接続されます。
- 11 : TIM20\_ETR が ADC3 AWD3 入力に接続されます。

**注 :** **ADC3 AWD とほかの TIM20\_ETR ソース信号の「論理和」がとられます。そのため、ソフトウェアのその他のソース (入力ピン) を無効にする必要があります。**

## 20.4.22 TIM1/TIM8/TIM20 キャプチャ/比較モードレジスタ 3 (TIMx\_CCMR3)

アドレスオフセット : 0x54

リセット値 : 0x0000 0000

上記の CCMR1 レジスタの説明を参照してください。チャンネル 5 および 6 は出力としてのみ設定できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							r/w								r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE.	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w	r/w	r/w	r/w		

### 出力比較モード

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **OC6M[3]** : 出力比較 6 モード - ビット 3

ビット 23:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **OC5M[3]** : 出力比較 5 モード - ビット 3

ビット 15 **OC6CE** : 出力比較 6 クリアイネーブル

ビット 14:12 **OC6M** : 出力比較 6 モード

ビット 11 **OC6PE** : 出力比較 6 プリロードイネーブル

ビット 10 **OC6FE** : 出力比較 6 高速イネーブル



- ビット 9:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **OC5CE** : 出力比較 5 クリアイネーブル
- ビット 6:4 **OC5M** : 出力比較 5 モード
- ビット 3 **OC5PE** : 出力比較 5 プリロードイネーブル
- ビット 2 **OC5FE** : 出力比較 5 高速イネーブル
- ビット 1:0 予約済みであり、リセット値に保持する必要があります。

## 20.4.23 TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 5 (TIMx\_CCR5)

アドレスオフセット : 0x58

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r/w	r/w	r/w													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 31 **GC5C3** : チャンネル 5 およびチャンネル 3 のグループ化

チャンネル 3 出力におけるひずみ :

0 : OC5REF の OC3REFC への影響はありません。

1 : OC3REFC は、OC3REFC と OC5REF の論理 AND です。

このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR2 でプリロード機能を選択している場合)。

**注 : このひずみを組み合わせ PWM 信号に適用することもできます。**
- ビット 30 **GC5C2** : チャンネル 5 およびチャンネル 2 のグループ化

チャンネル 2 出力におけるひずみ :

0 : OC5REF の OC2REFC への影響はありません。

1 : OC2REFC は、OC2REFC と OC5REF の論理 AND です。

このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR1 でプリロード機能を選択している場合)。

**注 : このひずみを組み合わせ PWM 信号に適用することもできます。**



ビット 29 **GC5C1** : チャネル 5 およびチャネル 1 のグループ化

チャネル 1 出力におけるひずみ :

0 : OC5REF の OC1REFC5 への影響はありません。

1 : OC1REFC は、OC1REFC と OC5REF の論理 AND です。

このビットは、直ちに有効にするか、プリロードしておいて更新イベント後に考慮にすることができます (TIMxCCMR1 でプリロード機能を選択している場合)。

**注 :** このひずみを組み合わせ PWM 信号に適用することもできます。

ビット 28:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CCR5[15:0]** : キャプチャ/比較 5 値

CCR5 は、実際のキャプチャ/比較 5 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR3 レジスタの OC5PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 5 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC5 出力に送信される値を含みます。

## 20.4.24 TIM1/TIM8/TIM20 キャプチャ/比較レジスタ 6 (TIMx\_CCR6)

アドレスオフセット : 0x5C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR6[15:0]** : キャプチャ/比較 6 値

CCR6 は、実際のキャプチャ/比較 6 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR3 レジスタの OC6PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 6 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC6 出力に送信される値を含みます。

## 20.4.25 TIM1/TIM8/TIM20 レジスタマップ

TIM1/TIM8/TIM20 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 122. TIM1/TIM8/TIM20 レジスタマップとリセット値

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]	ARPE	CMS [1:0]	DIR	OPM	URS	UDIS	CEN				
	リセット値																					0	0	0	0	0	0	0	0	0	0	0			
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]	CCDS	CCUS	Res.	CCPC				
	リセット値										0	0	0	0		0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x08	TIMx_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]	ETP	ECE	ETP S [1:0]	ETF[3:0]	MSM	TS[2:0]	OCCS	SMS[2:0]										
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																		
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																		
0x18	TIMx_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値								0																										
	TIMx_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
リセット値																																			
0x1C	TIMx_CCMR2 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値								0																										
	TIMx_CCMR2 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
リセット値																																			
0x20	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																		
0x24	TIMx_CNT	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値	0																																	

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

表 122. TIM1/TIM8/TIM20 レジスタマップとリセット値 (続き)

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																		
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x30	TIMx_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x34	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x38	TIMx_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x3C	TIMx_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x40	TIMx_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x44	TIMx_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]			BKF[3:0]			MOE	AOE	BKP	BKE	OSSR	OSSI	LOC K [1:0]	DT[7:0]													
	リセット値							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x48	TIMx_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]											
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0				
0x4C	TIMx_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x50	TIMx_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																																	0	0	0



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

表 122. TIM1/TIM8/TIM20 レジスタマップとリセット値 (続き)

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x54	TIMx_CCMR3 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]		OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]		OC5PE	OC5FE	Res.	Res.		
	リセット値								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	TIMx_CCR5	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	リセット値	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	TIMx_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 21 汎用タイマ (TIM2/TIM3/TIM4)

### 21.1 TIM2/TIM3/TIM4 の概要

この汎用タイマは、プログラマブルなプリスケラによって駆動される 16 ビットまたは 32 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較と PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 21.3.19: タイマの同期](#)に示すように、相互に同期させることができます。

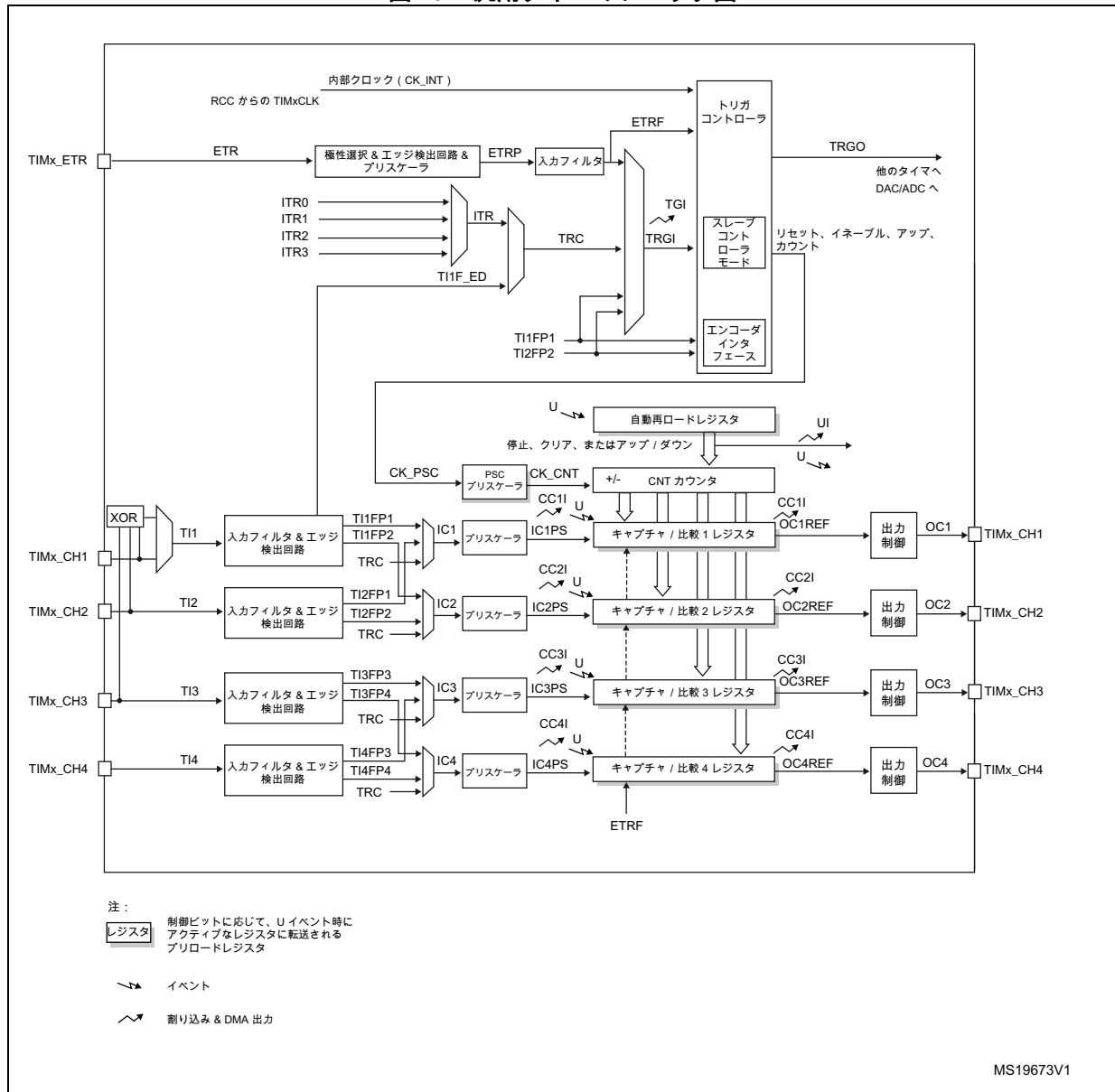
**注:** *TIM4 は STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用できません。*

### 21.2 TIM2/TIM3/TIM4 の主な特長

汎用 TIMx タイマの主な機能は、次のとおりです。

- 16 ビット (TIM3 および TIM4) または 32 ビット (TIM2) のアップ、ダウン、アップ/ダウン自動再ロードカウンタ。
- カウンタクロック周波数を、1 から 65535 の間で分周する 16 ビットプログラム可能プリスケラ。
- 次の機能を持つ、最大 4 つの独立チャンネル。
  - 入力キャプチャ
  - 出力比較
  - PWM 生成 (エッジアラインモードとセンターアラインモード)
  - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み/DMA 生成。
  - 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
  - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
  - 入力キャプチャ
  - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 197. 汎用タイマのブロック図



## 21.3 TIM2/TIM3/TIM4機能詳細

### 21.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビット/32 ビットカウンタです。カウンタは、カウントアップ、カウントダウン、またはカウントアップとカウントダウンの両方を行います。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC) :
- 自動再ロードレジスタ (TIMx\_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー (またはダウンカウント時はアンダーフロー) に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK\_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

実際のカウンタイネーブル信号 CNT\_EN は、CEN の 1 クロックサイクル後にセットされます。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビット/32 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 198](#) と [図 199](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 198. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

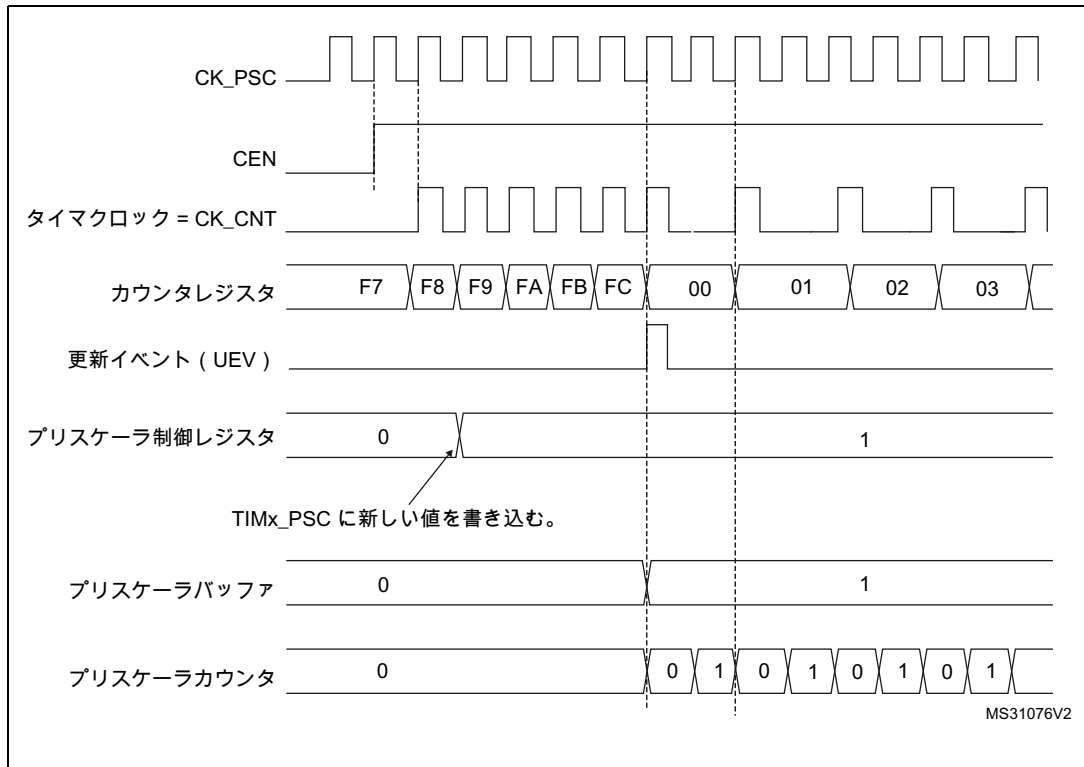
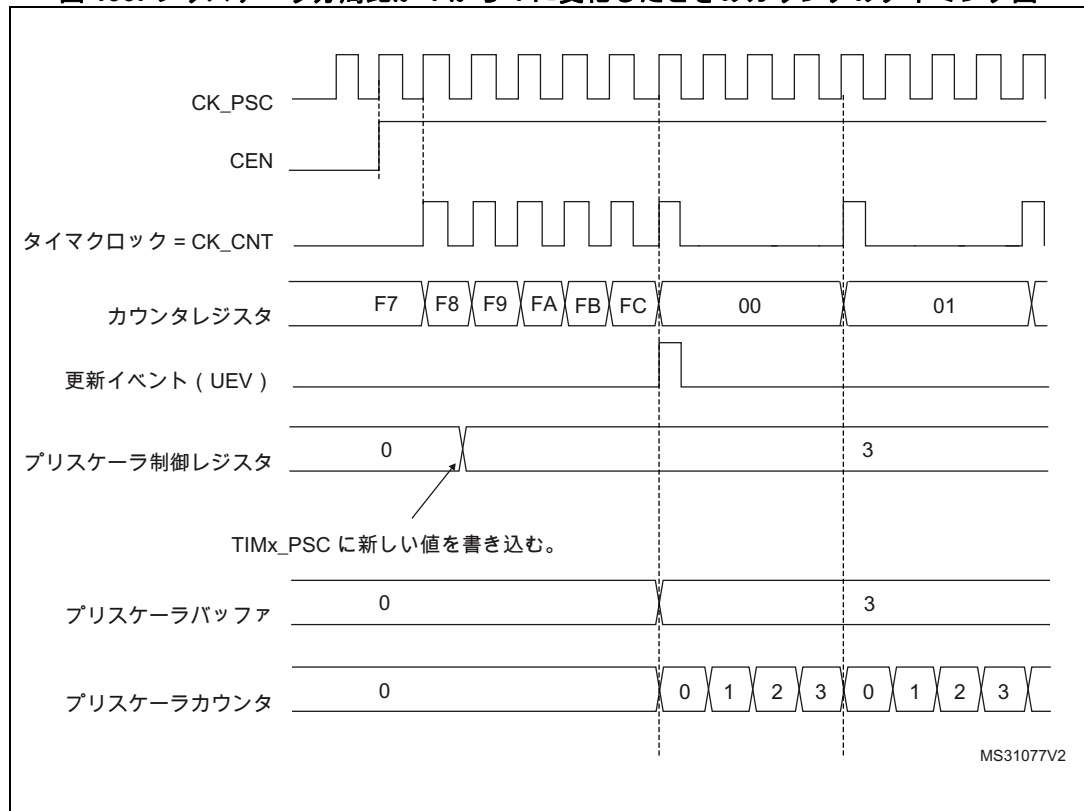


図 199. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図





## 21.3.2 カウンタモード

### アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、(ソフトウェアで、または、スレープモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることで生成できます。

UEV イベントは、TIMx\_CR1 レジスタの UDIS ビットをセットすることにより、ソフトウェアで無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 200. 内部クロック分周比が 1 の場合のカウンタのタイミング図

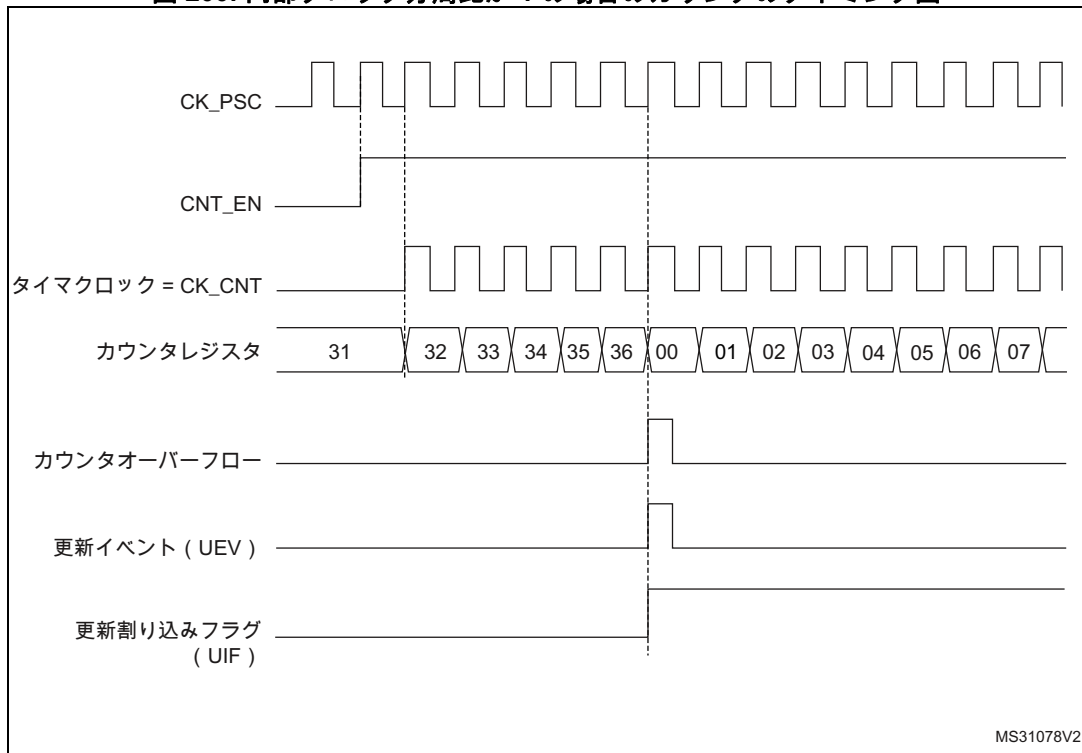


図 201. 内部クロック分周比が 2 の場合のカウンタのタイミング図

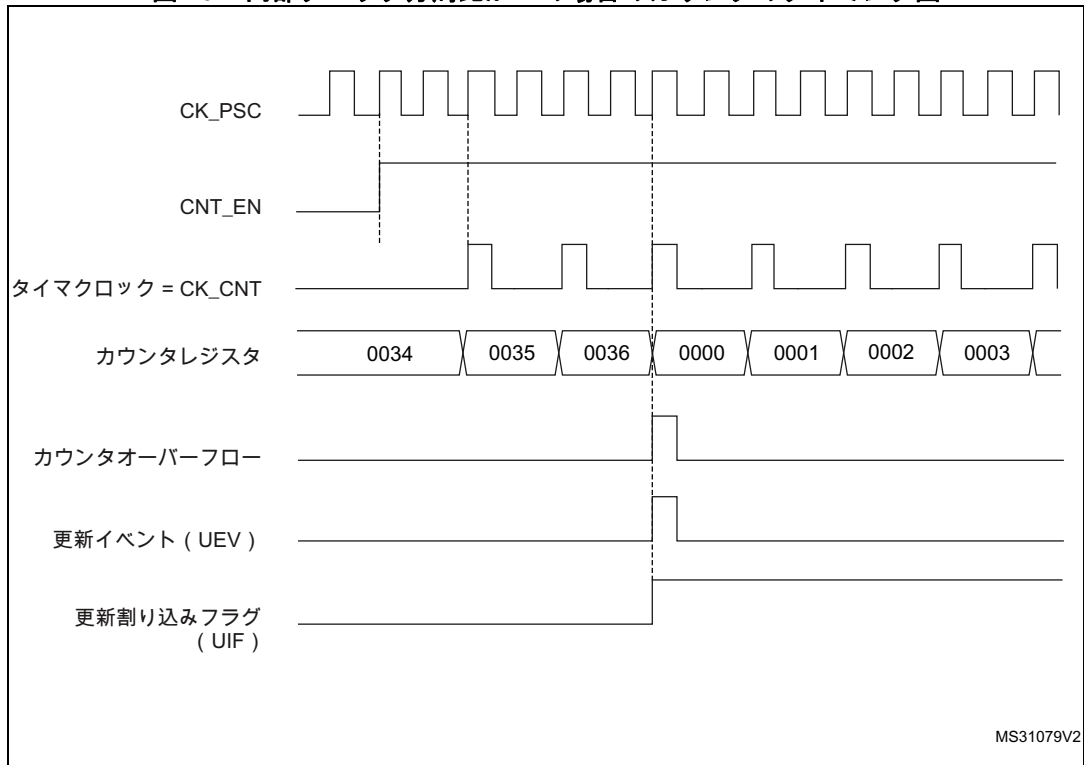


図 202. 内部クロック分周比が 4 の場合のカウンタのタイミング図

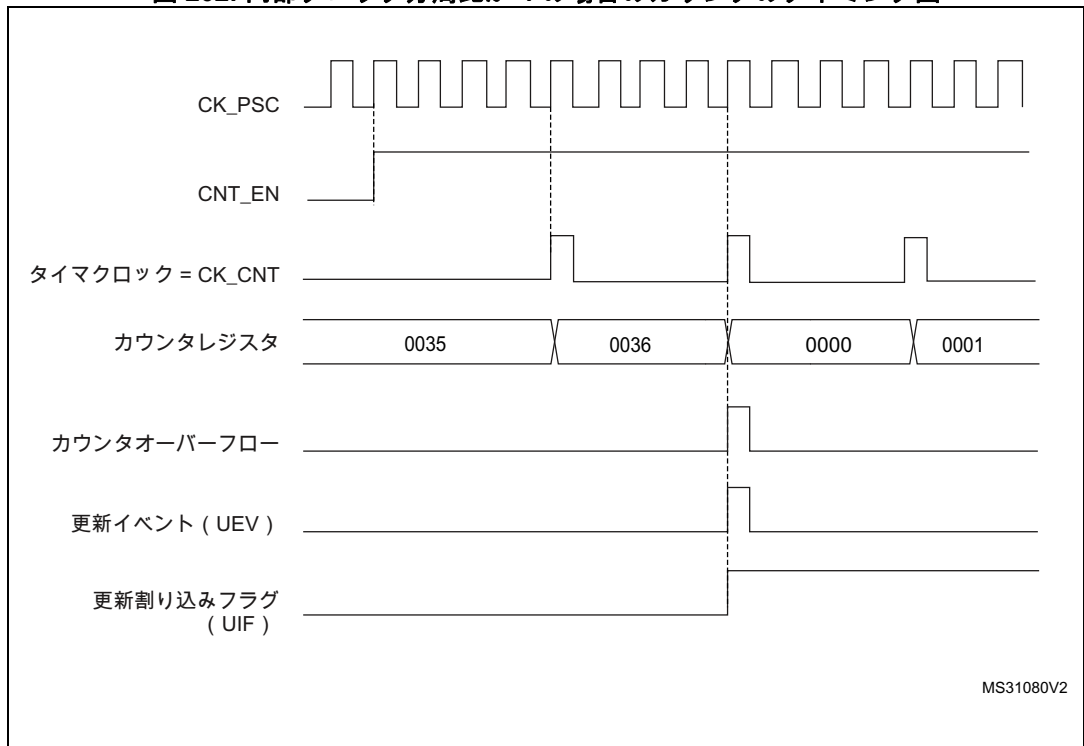


図 203. 内部クロック分周比が N の場合のカウンタのタイミング図

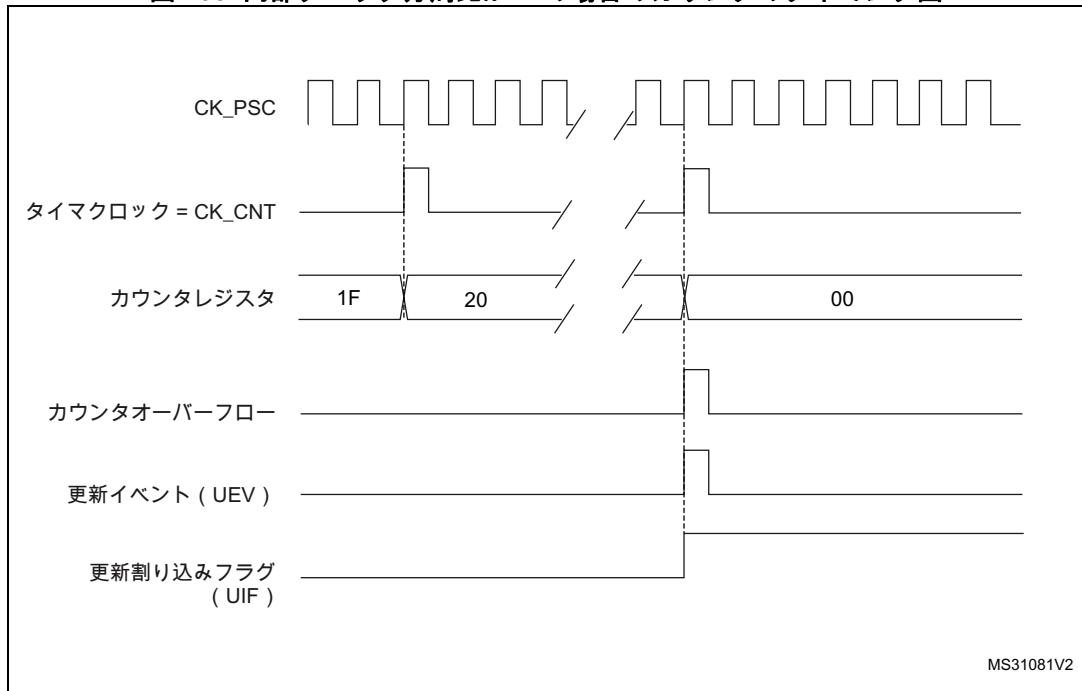


図 204. ARPE=0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

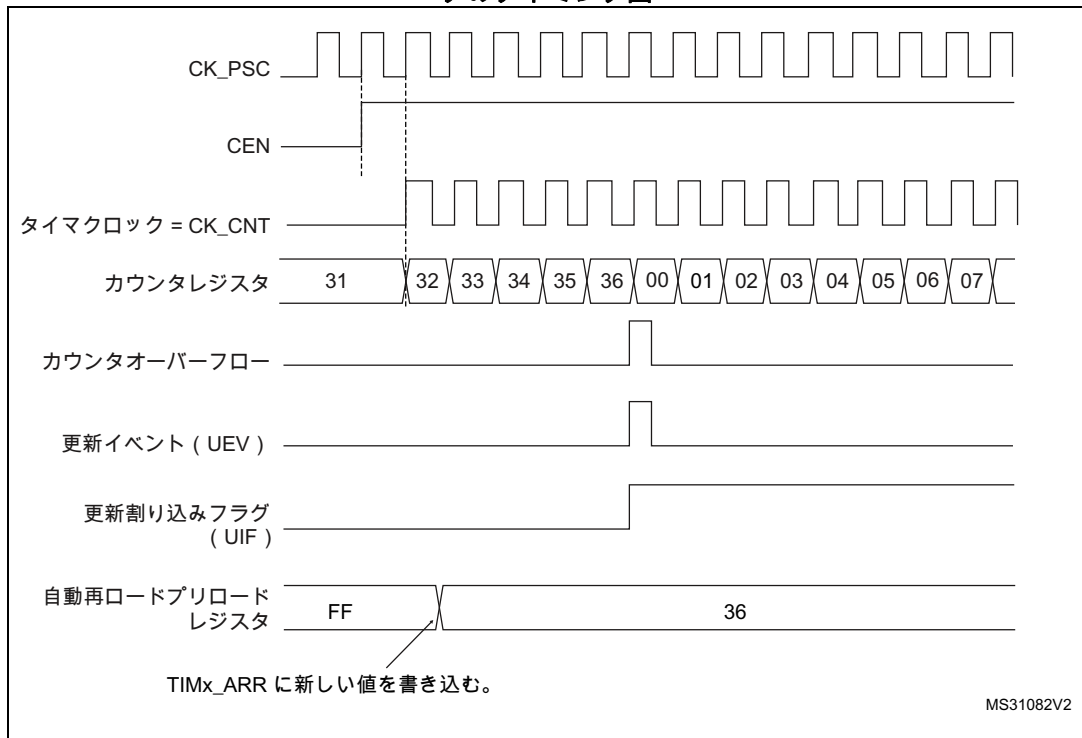
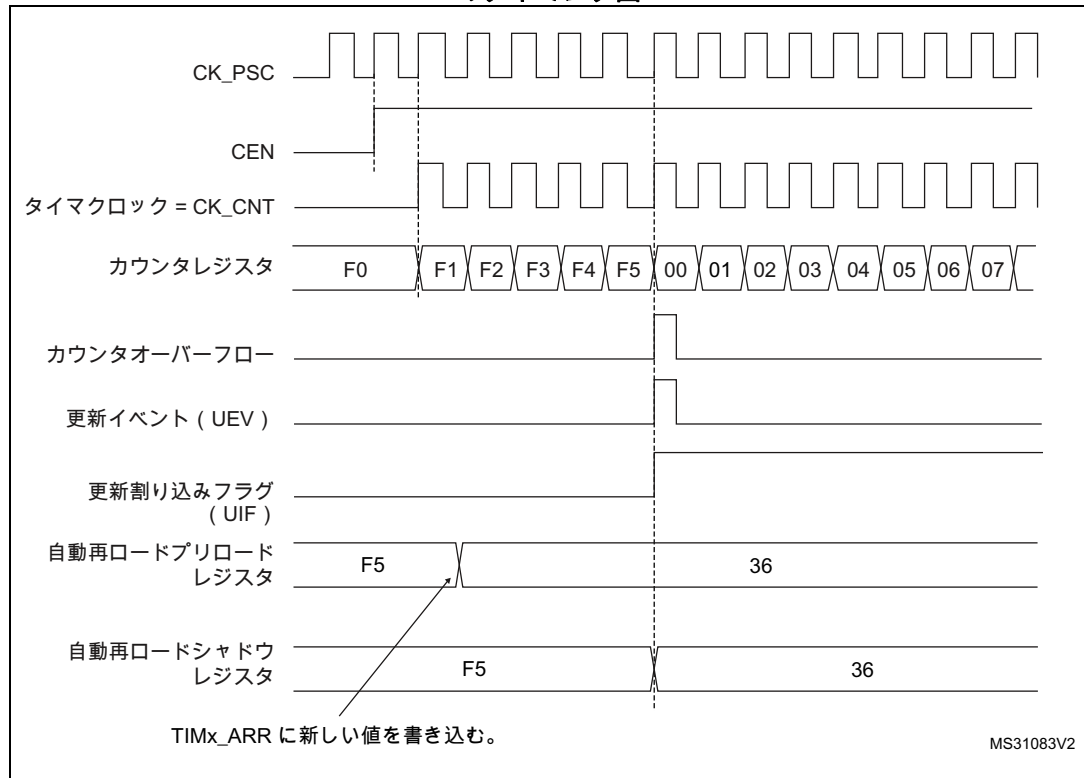


図 205. ARPE=1 (TIMx\_ARR はプリロードされる) のときの更新イベント時のカウンタのタイミング図



## ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx\_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

更新イベントは、カウンタアンダーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることにより生成できます。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。その後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

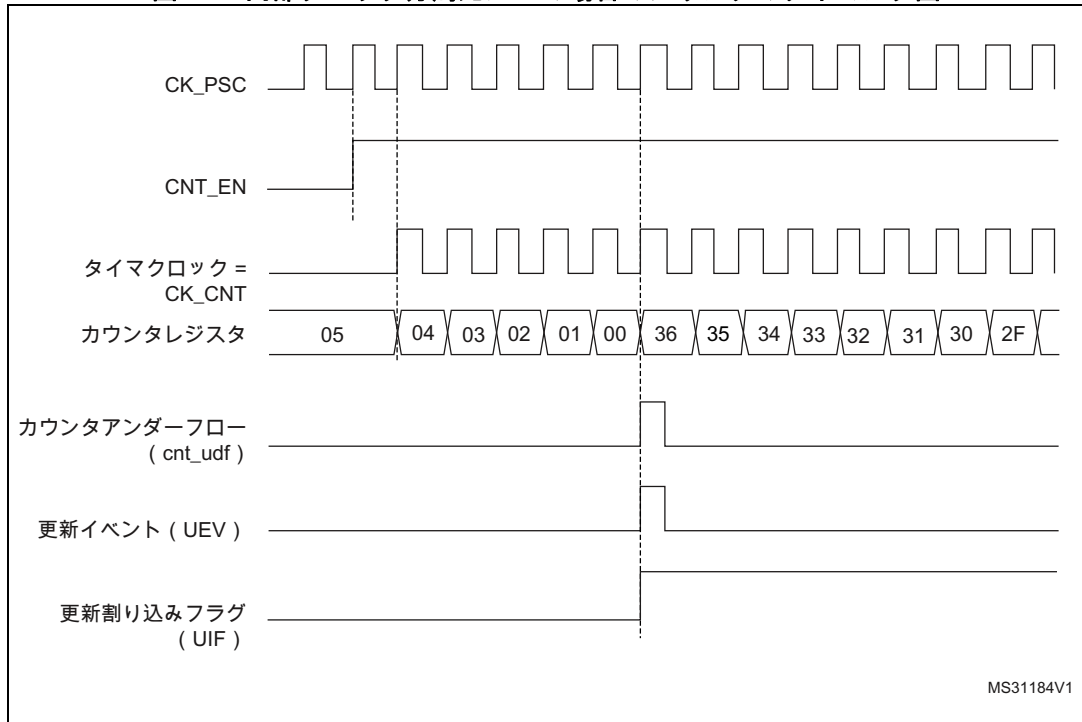
さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

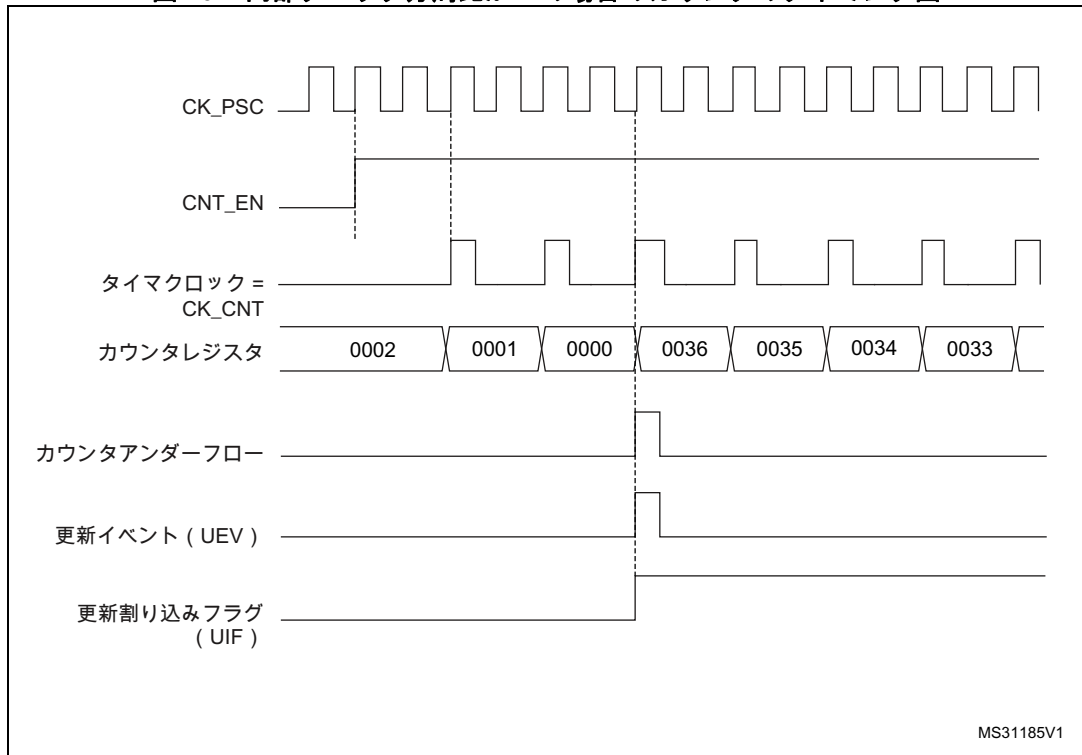
以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 206. 内部クロック分周比が 1 の場合のカウンタのタイミング図



MS31184V1

図 207. 内部クロック分周比が 2 の場合のカウンタのタイミング図



MS31185V1

図 208. 内部クロック分周比が 4 の場合のカウンタのタイミング図

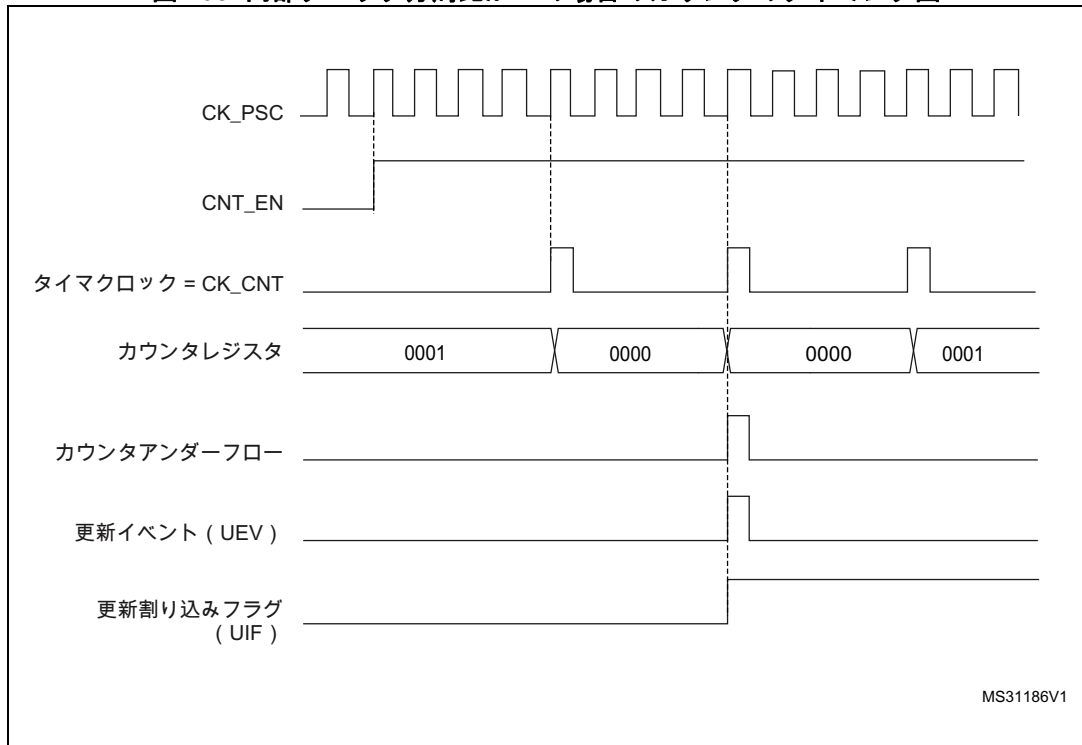


図 209. 内部クロック分周比が N の場合のカウンタのタイミング図

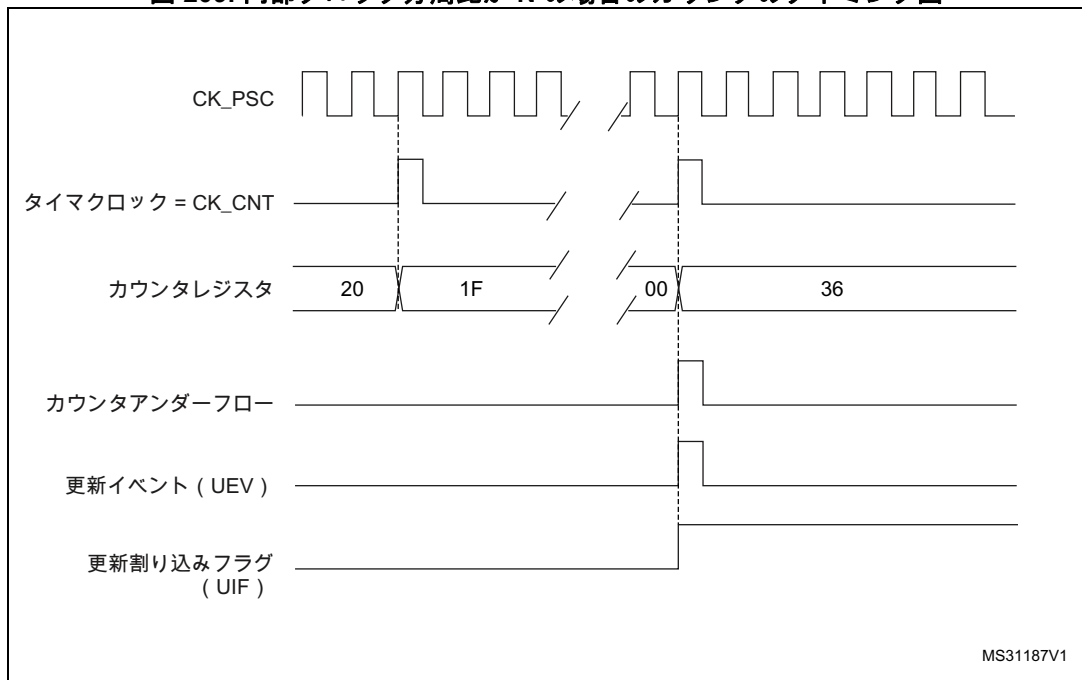
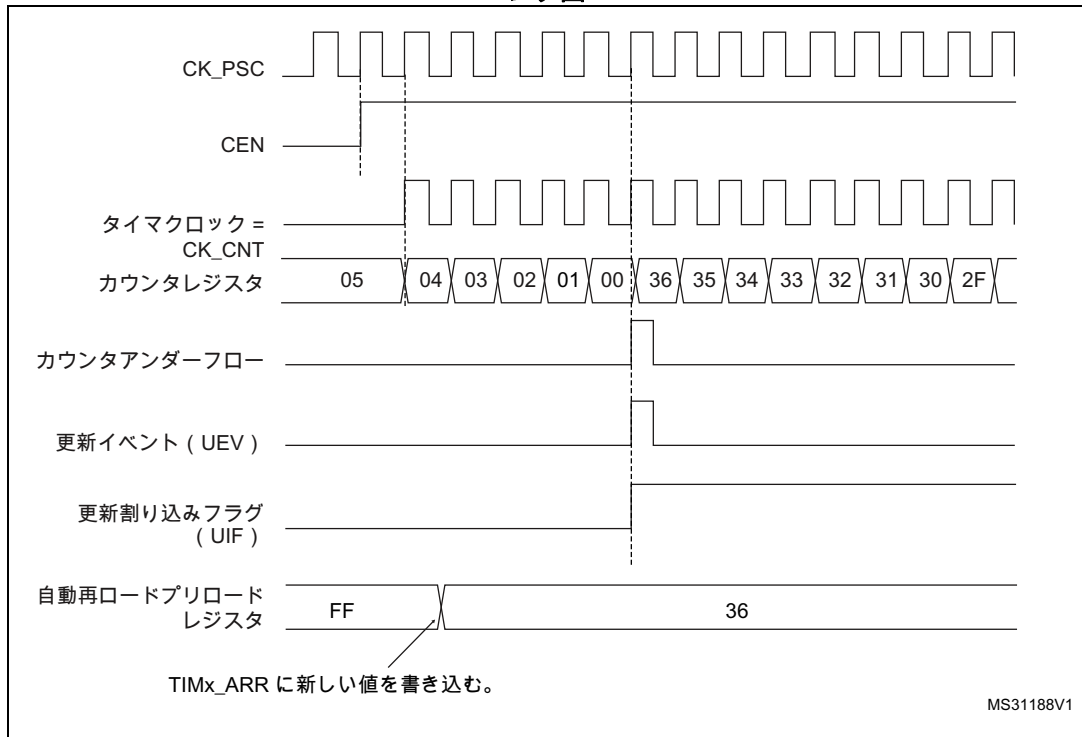


図 210. 繰り返しカウンタが使用されていないときの更新イベント時のカウンタのタイミング図



## センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS=01)、カウンタがカウントアップするとき (センターアラインモード 2、CMS=10)、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS=11) にセットされます。

このモードでは、方向ビット (TIMx\_CR1 レジスタの DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (した

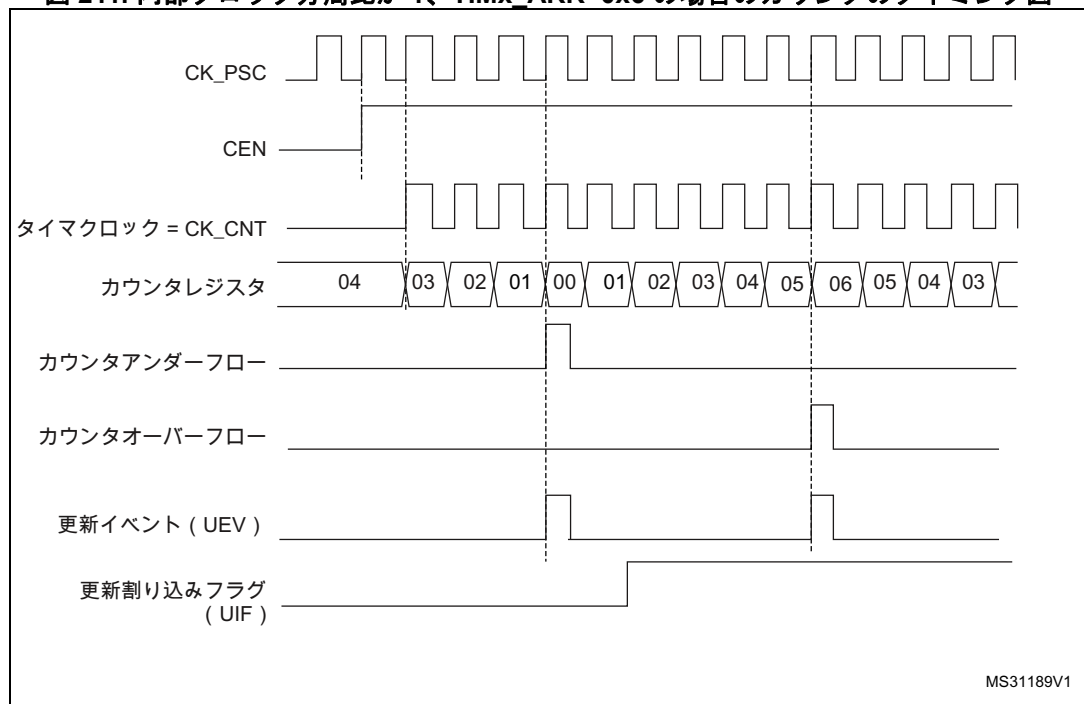
がって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx\_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 211. 内部クロック分周比が 1、TIMx\_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターアラインモード 1 が使用されています (詳細については、[セクション 21.4.1: TIMx 制御レジスタ 1 \(TIMx\\_CR1\)](#) (642 ページ) を参照)。



図 212. 内部クロック分周比が 2 の場合のカウンタのタイミング図

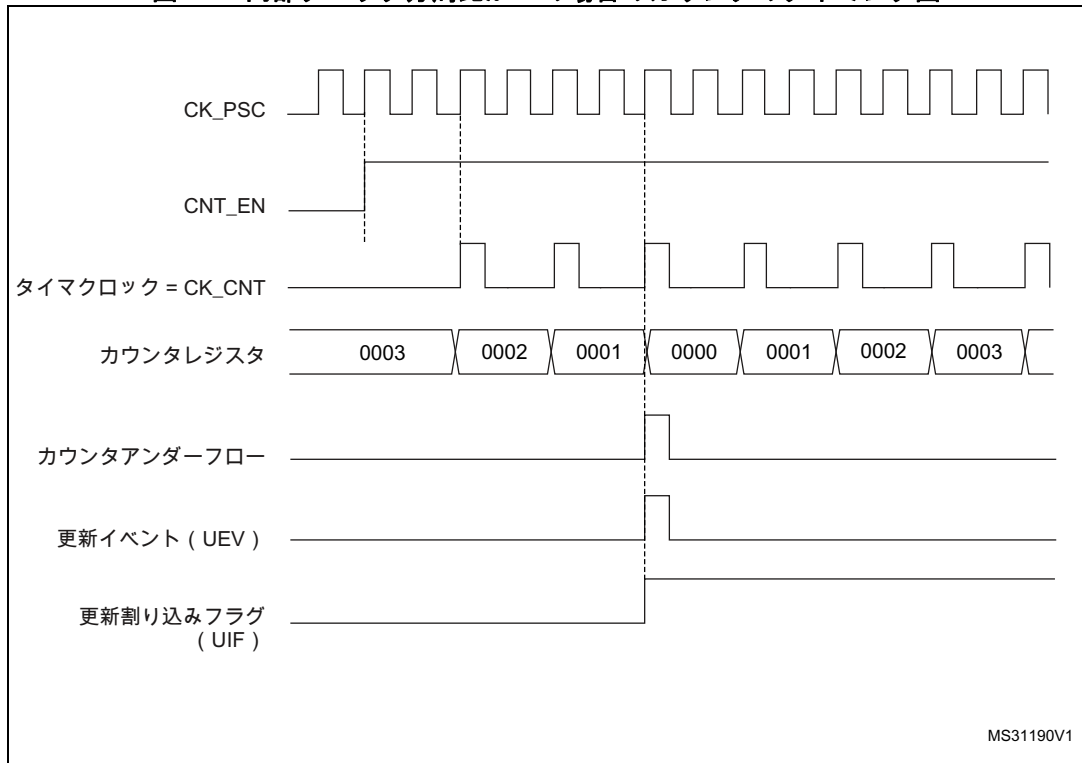
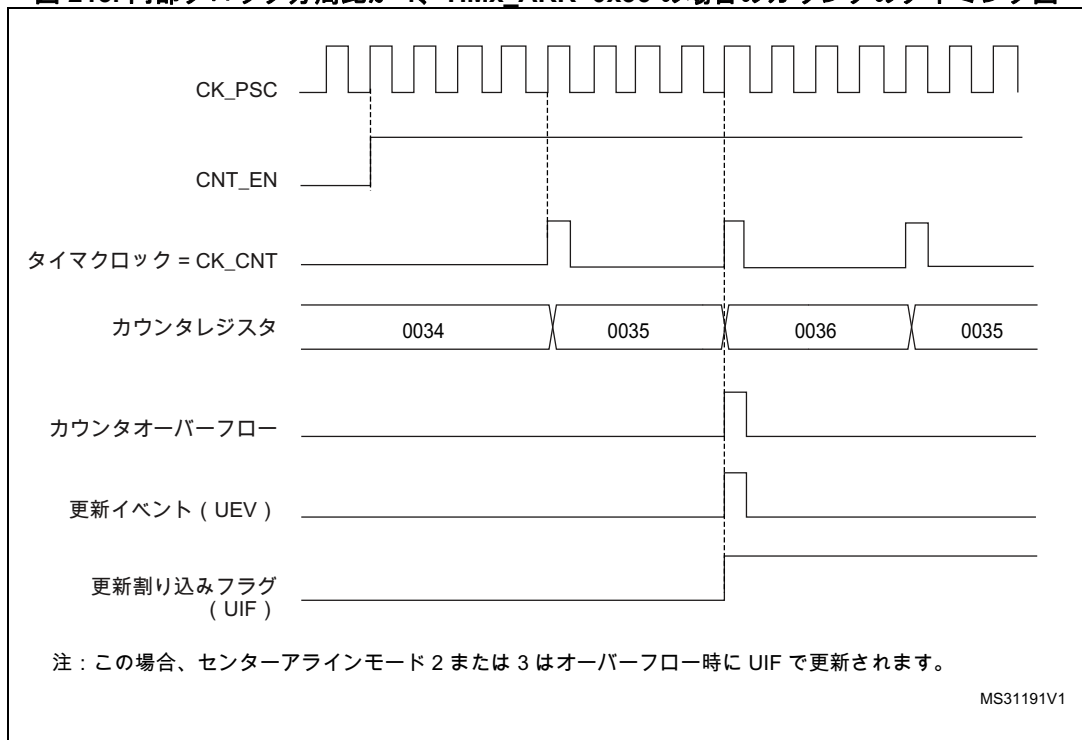


図 213. 内部クロック分周比が 4、TIMx\_ARR=0x36 の場合のカウンタのタイミング図



1. センターアラインモード 2 または 3 が使用され、オーバーフロー時に UIF がセットされます。

図 214. 内部クロック分周比が N の場合のカウンタのタイミング図

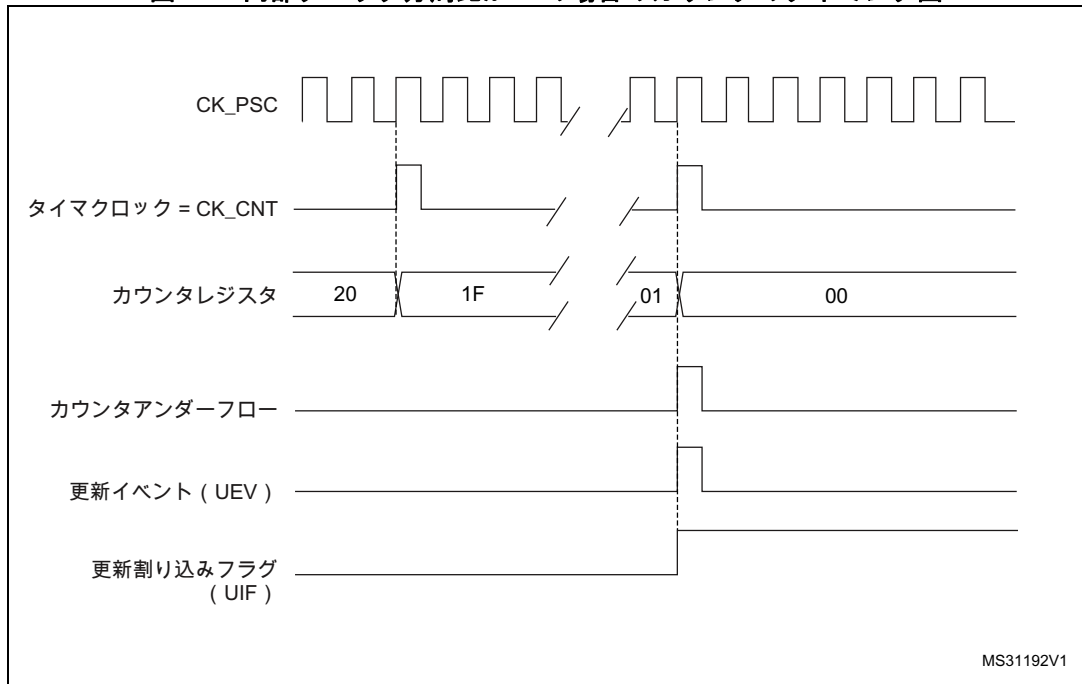


図 215. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図

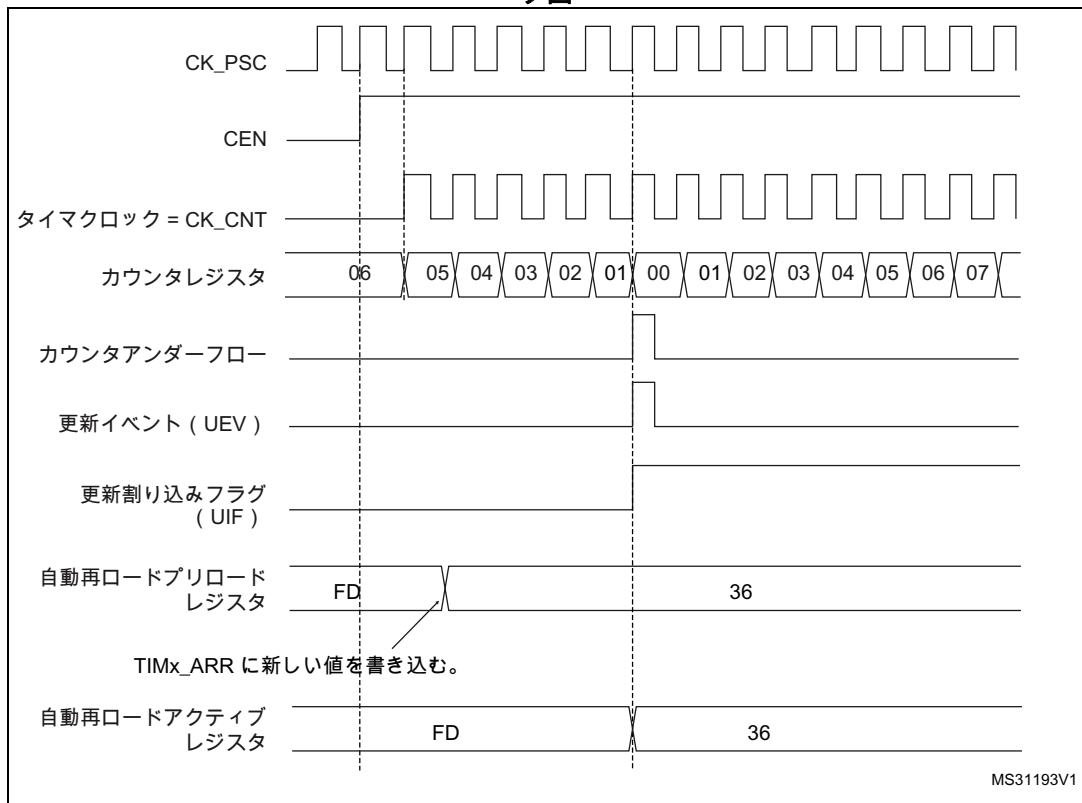
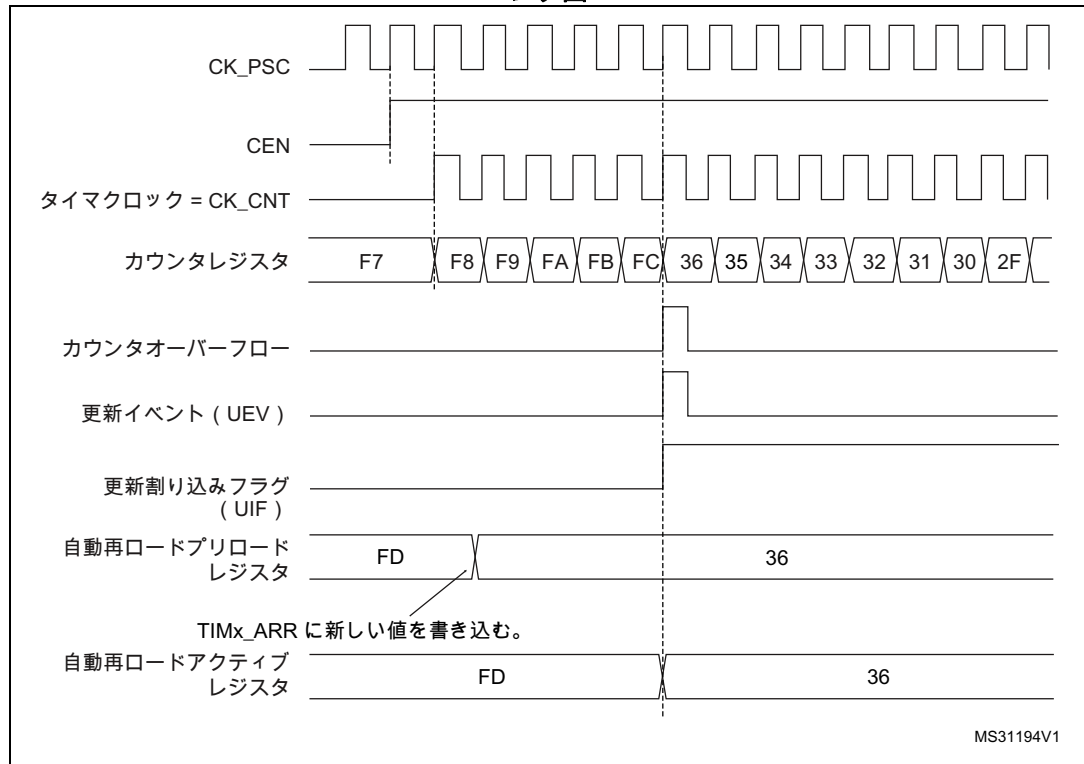


図 216. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



## 21.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

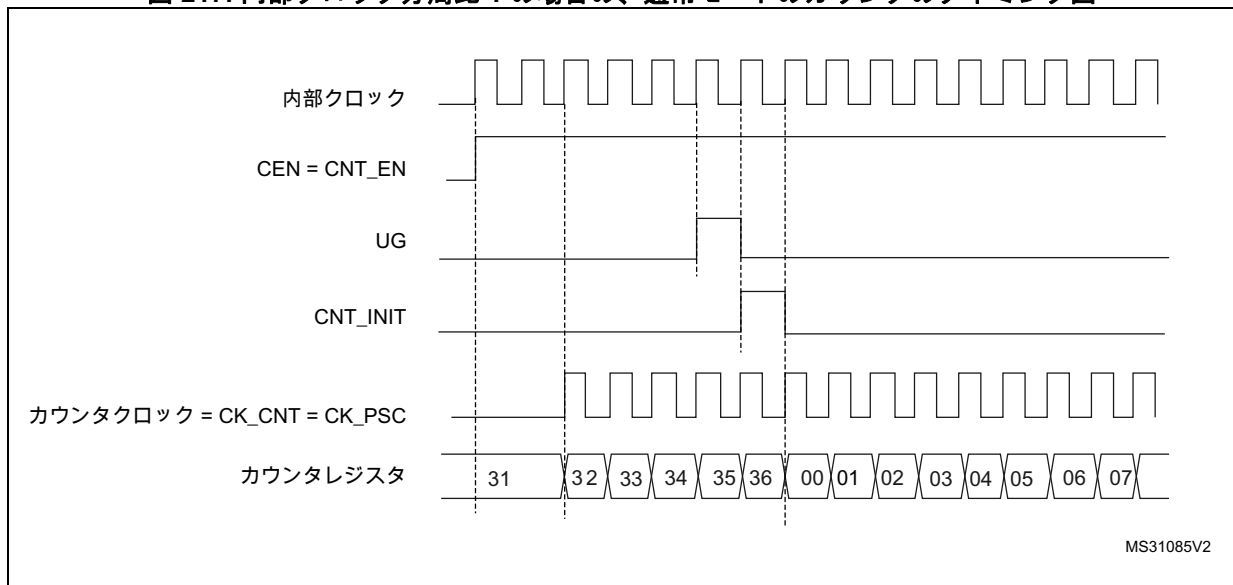
- 内部クロック (CK\_INT)
- 外部クロックモード 1: 外部入力ピン (Tix)
- 外部クロックモード 2: 外部トリガ入力 (ETR)
- 内部トリガ入力 (ITRx): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 13 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する \(636 ページ\)](#) を参照してください。

### 内部クロックソース (CK\_INT)

スレーブモードコントローラが無効の場合 (TIMx\_SMCR レジスタの SMS=000)、CEN、DIR ビット (TIMx\_CR1 レジスタ) と UG ビット (TIMx\_EGR レジスタ) が実際の制御ビットであり、ソフトウェアでのみ変更できます (自動的にクリアされたままの UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK\_INT が供給されます。

図 217 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

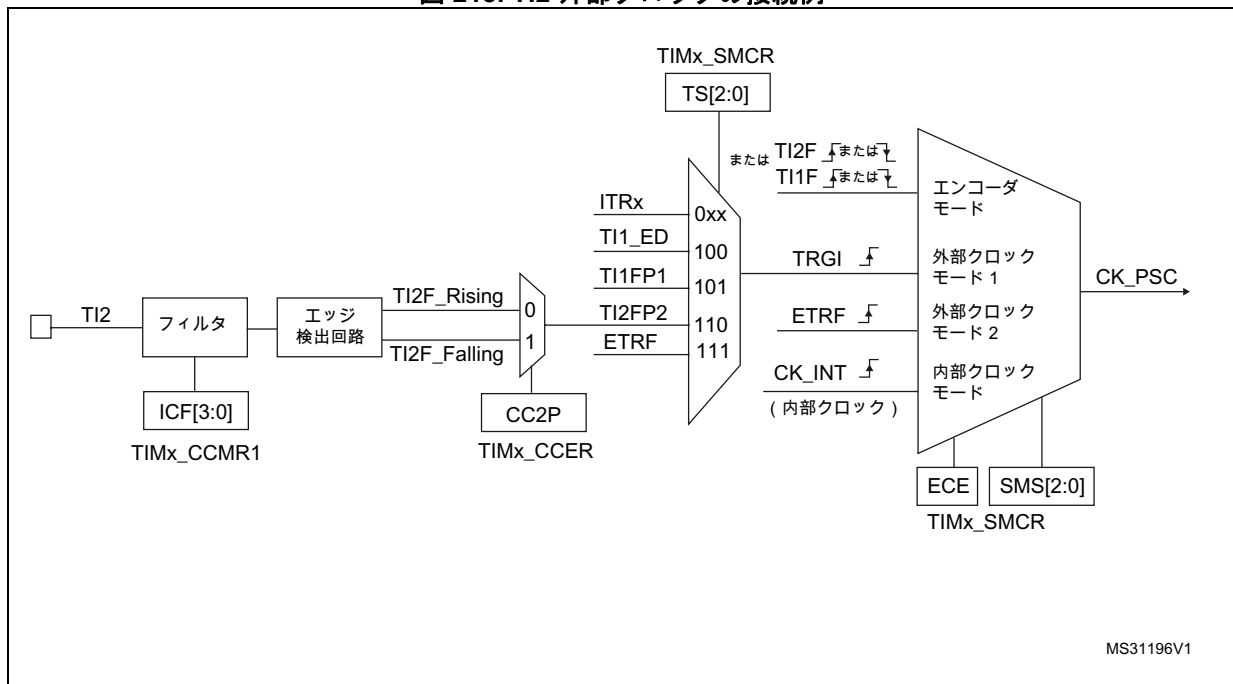
図 217. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



## 外部クロックソースモード 1

このモードは TIMx\_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 218. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx\_CCMR1 レジスタの CC2S ビットに「01」を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx\_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。

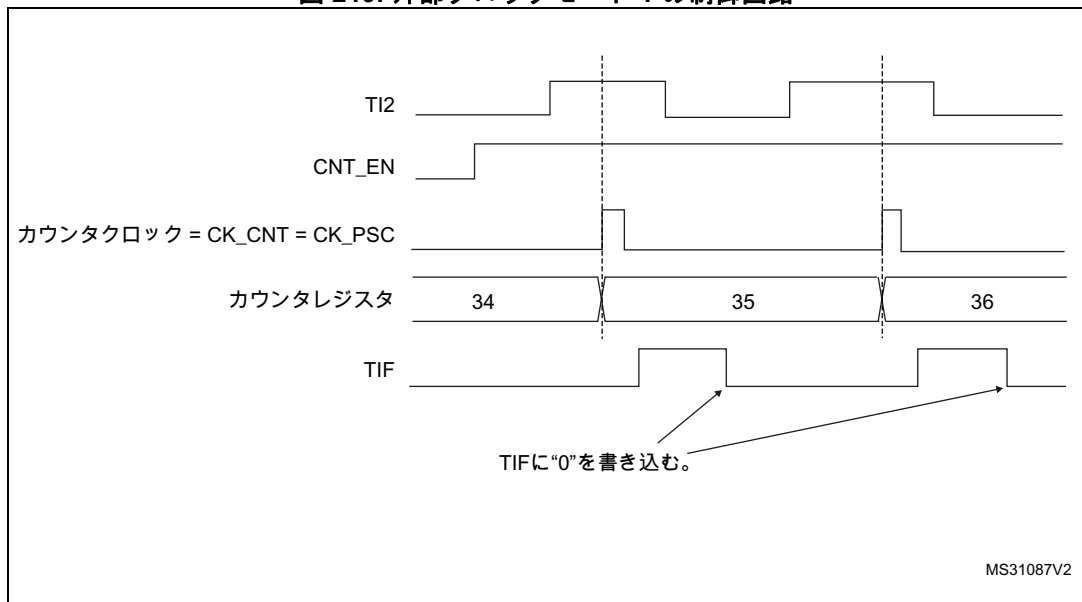
**注:** キャプチャプリスケアラはトリガには使用されないため、設定は不要です。

3. CC2P=0、CC2NP=0、および CC2NP=0 を TIMx\_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx\_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。
6. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 219. 外部クロックモード 1 の制御回路



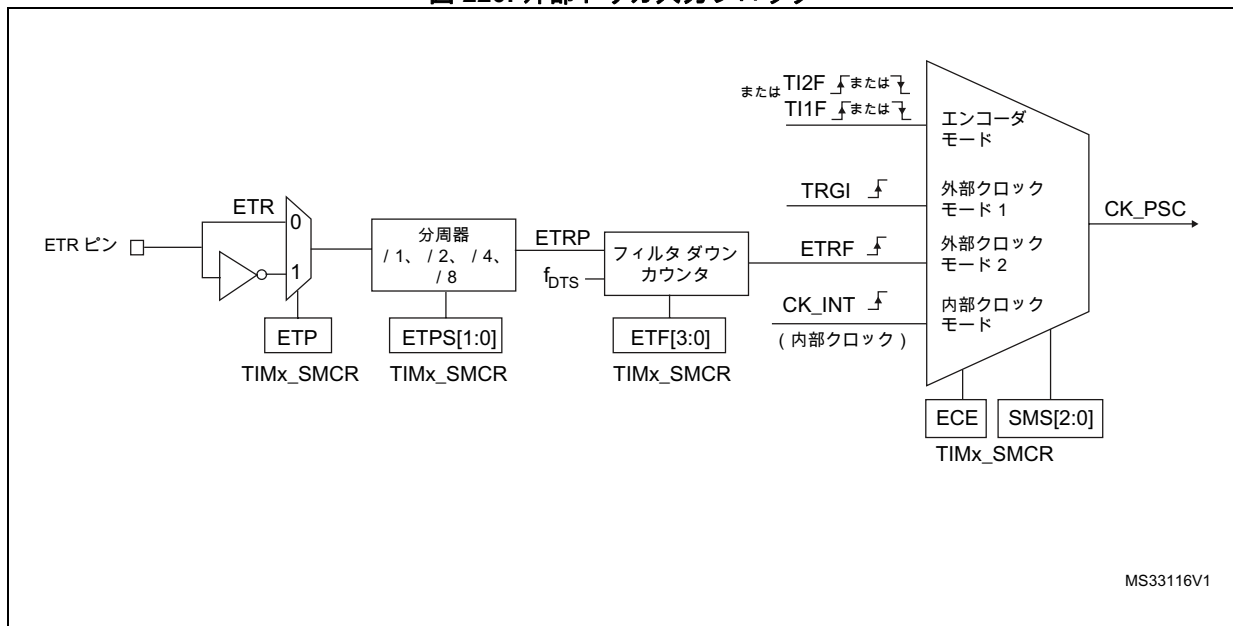
## 外部クロックソースモード 2

このモードは、TIMx\_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 220 に、外部トリガ入力ブロックの概要を示します。

図 220. 外部トリガ入力ブロック



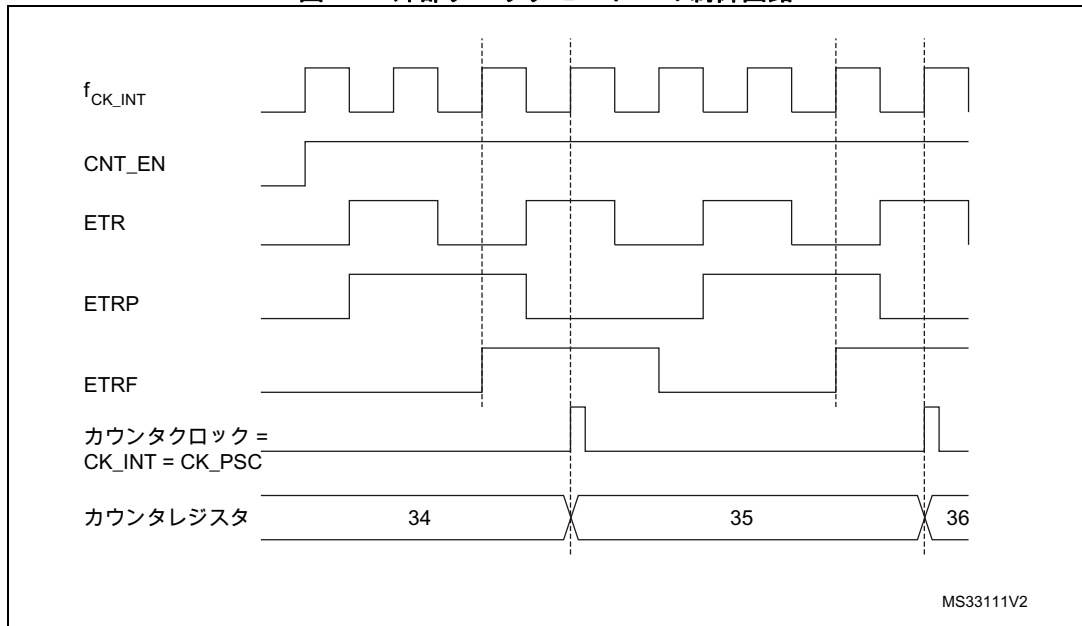
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx\_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx\_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx\_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx\_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 221. 外部クロックモード 2 の制御回路



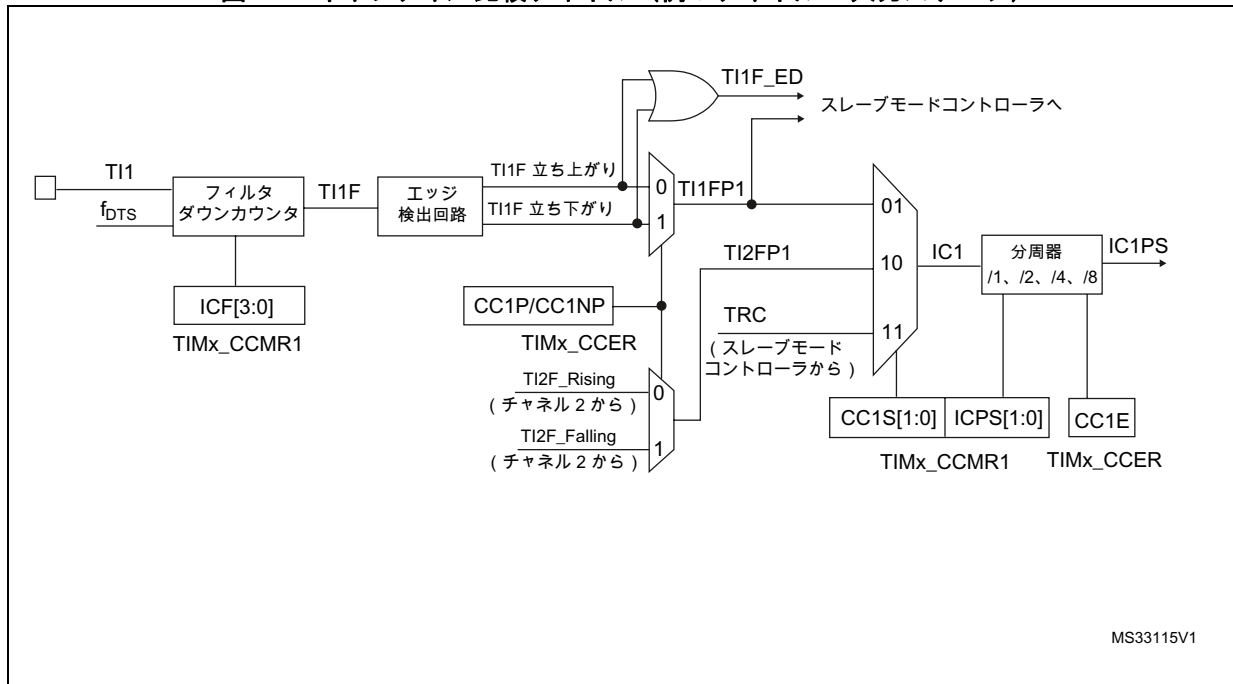
## 21.3.4 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケラ）、および出力カステージ（比較回路と出力制御）から構成されています。

次の図に、キャプチャ／比較チャネルの概要を示します。

入カステージは、対応する TIX 入力をサンプリングして、フィルタリングを行った TIXF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIXFPx) を生成します。この信号はプリスケラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 222. キャプチャ/比較チャンネル (例: チャンネル 1 入カステージ)



出力ステージは、OCxRef (アクティブハイ) として使用される中間波形を生成します。OCxRef はアクティブハイです。信号の極性は最終出力に影響を与えます。

図 223. キャプチャ/比較チャンネル 1 メイン回路

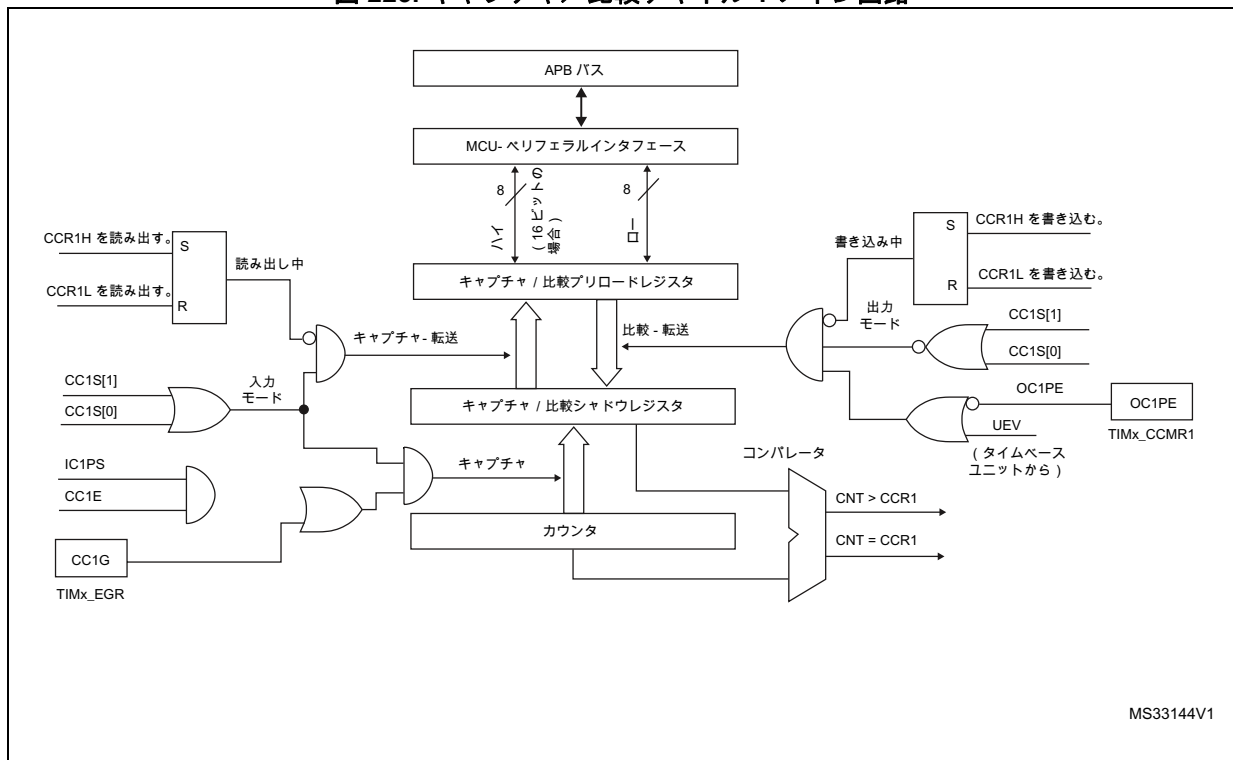
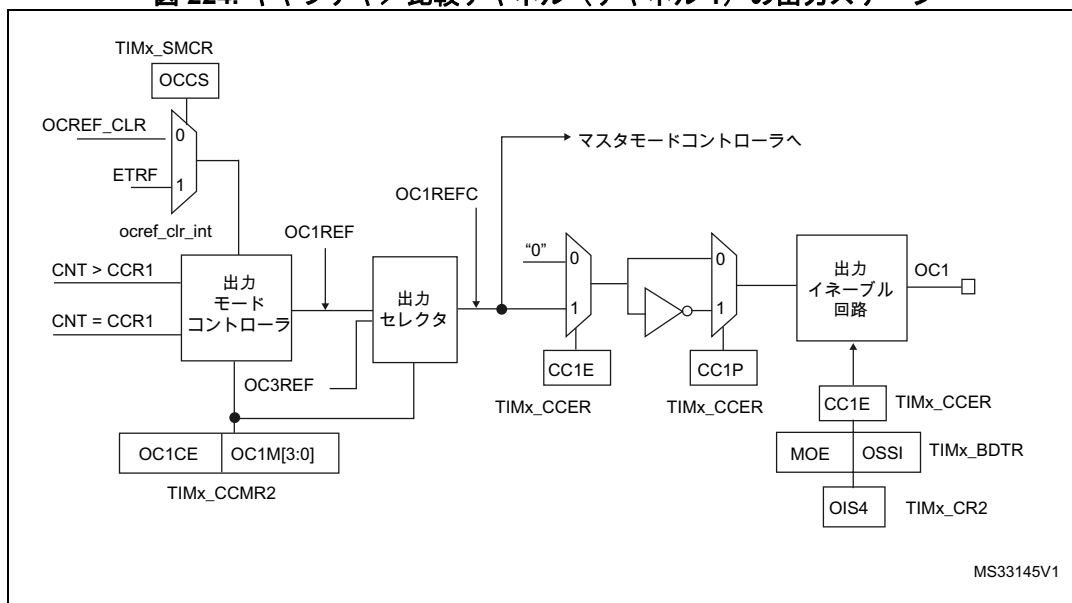




図 224. キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ



キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

### 21.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx\_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx\_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx\_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx\_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx\_CCR1 は TI1 入力とリンクされていなければならないので、このためには TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx\_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数  $f_{DTS}$  でサンプリング)。この場合、TIMx\_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx\_CCER レジスタの CC1P、CC1NP、および CC1NP ビットに“000”を書き込みます (この場合、立ち上がりエッジの選択)。



4. 入力プリスケアラをプログラムします。今回の例では、有効な信号変化ごとにキャプチャを行いたいため、プリスケアラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに 00 を書き込みます)。
5. TIMx\_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
6. 必要な場合は、TIMx\_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx\_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

**注：** IC 割り込みと DMA リクエストは、TIMx\_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

## 21.3.6 PWM 入力モード

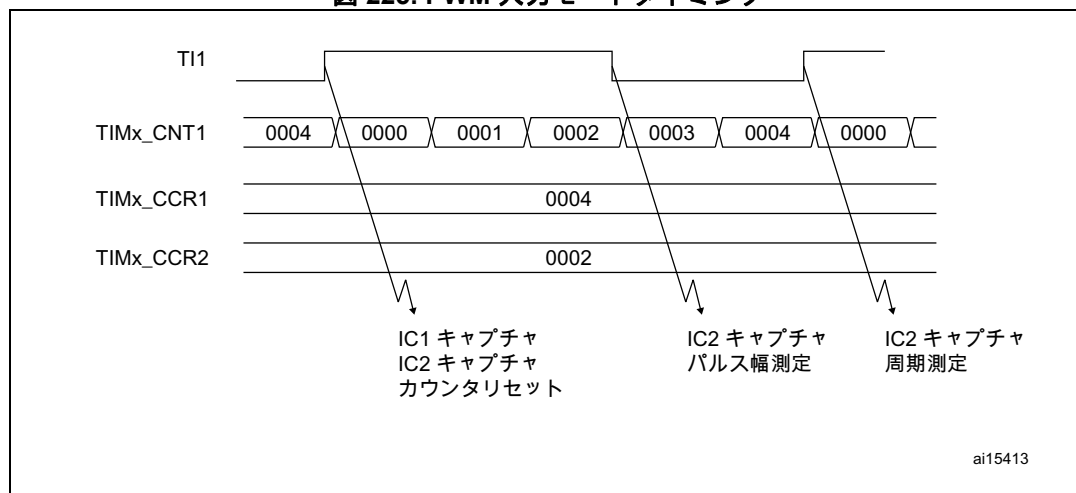
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2 つの ICx 信号が同じ Tlx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます (手順は、CK\_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

1. TIMx\_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx\_CCR1 のアクティブ入力を選択します。
2. CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブ極性を選択します (TIMx\_CCR1 のキャプチャとカウンタクリアの両方に使用します)。
3. TIMx\_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
4. CC2P ビットに“1”を、CC2NP ビットに“0”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx\_CCR2 のキャプチャに使用されます)。
5. TIMx\_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
6. TIMx\_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
7. TIMx\_CCER レジスタの CC1E と CC2E ビットに 1 を書き込むことによって、キャプチャを有効にします。

図 225. PWM 入力モードタイミング



1. TI1FP1 と TI2FP2 のみがスレーブモードコントローラに接続されているので、PWM 入力モードは TIMx\_CH1/TIMx\_CH2 信号でのみ使用できます。

## 21.3.7 強制出力モード

このモード (TIMx\_CCMRx レジスタの CCxS=00) では、各出力比較信号 (OCxREF、そして OCx) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx\_OCMRx レジスタの OCxM ビットに 101 を書き込みます。これにより、OCxREF は強制的にハイレベルになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例 : CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

TIMx\_CCMRx レジスタの OCxM ビットに 100 を書き込むことによって、OCxREF 信号を強制的にローにできます。

いずれにしても、TIMx\_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、出力比較モードのセクションで説明します。

## 21.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx\_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx\_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx\_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx\_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信しません (DMA リクエスト選択には、TIMx\_CR2 レジスタの CCDS ビットが使用されます)。

TIMx\_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx\_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

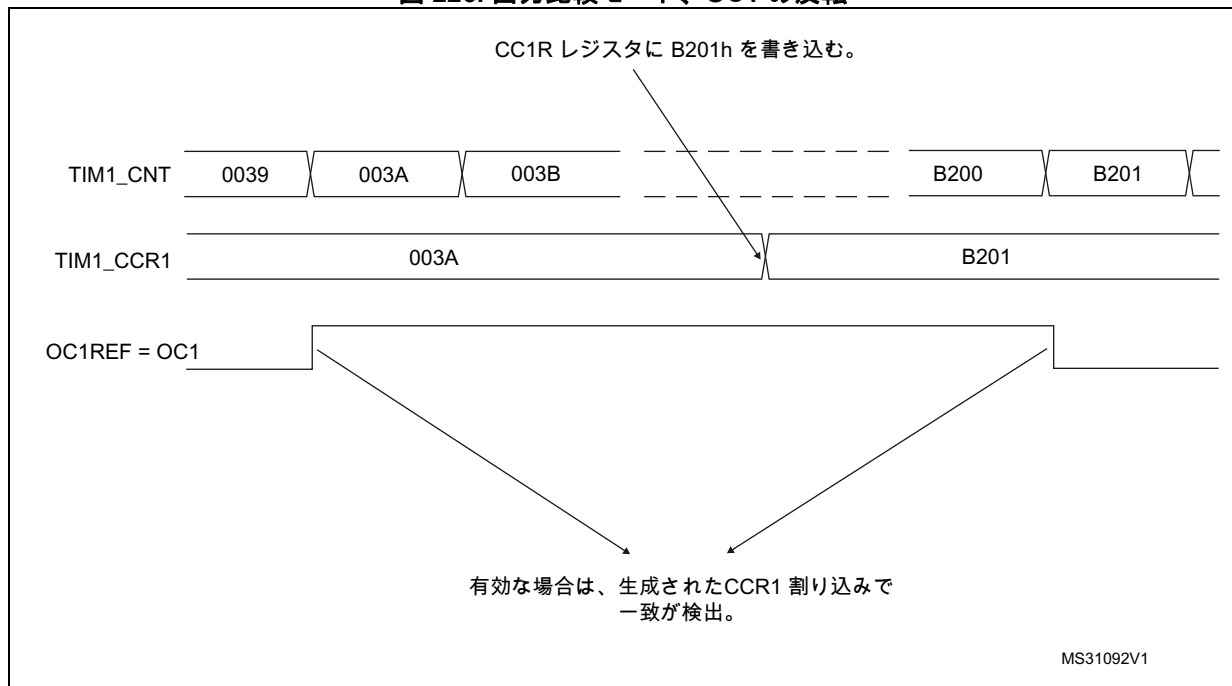
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

## 手順

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx\_ARR レジスタと TIMx\_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエスト/DMA リクエストを生成する場合は、CCxIE ビット/CCxDE ビットをセットします。
4. 出力モードを選択します。たとえば、CNT が CCRx と一致したときに OCx 出力をトグルし、CCRx プリロードを使用せず、OCx が有効でアクティブハイのときには、OCxM=011、OCxPE=0、CCxP=0、CCxE=1 を書き込みます。
5. TIMx\_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアで TIMx\_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx\_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 226](#) に示します。

図 226. 出力比較モード、OC1 の反転



## 21.3.9 PWM モード

パルス幅変調(PWM)モードでは、TIMx\_ARR レジスタの値によって決められた周波数と TIMx\_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、個々のチャンネル (OCx 出力ごとに PWM 1 波形) で、TIMx\_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) や“111” (PWM モード 2) を書き込むことで、独自に選択できません。TIMx\_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx\_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx\_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx\_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx\_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx\_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx\_CNT と TIMx\_CCRx が常に比較されて、TIMx\_CCRx ≤ TIMx\_CNT または TIMx\_CNT ≤ TIMx\_CCRx かどうか判断されます (カウントの方向によります)。ただし、OCREF\_CLR 機能 (OCREF は、次の PWM 周期までは ETR 信号を通じて外部イベントによってクリアできる) に従って、OCREF 信号は次の場合にのみアサートされます。

- 比較結果が変化したとき、または
- 出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) が停止構成 (比較なし、OCxM=000) から PWM モードの 1 つ (OCxM=110 または 111) へ切り替えられたとき。

タイマの動作中は、ソフトウェアで強制的に PWM になります。

タイマは、TIMx\_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

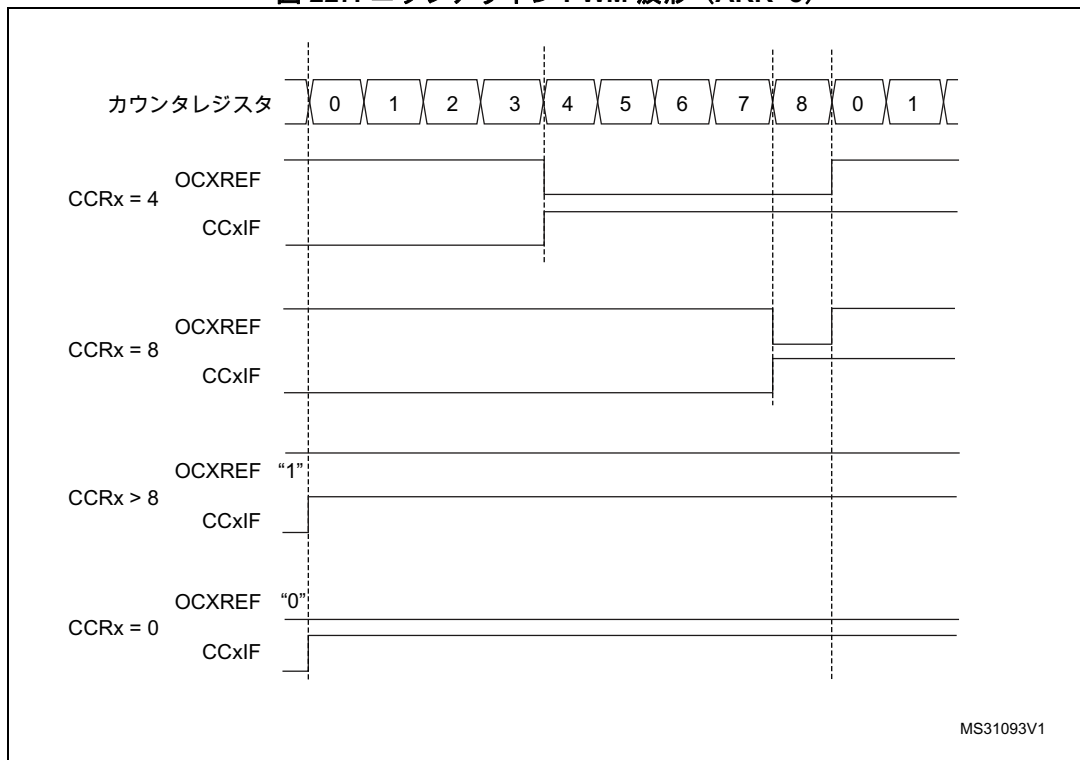
### PWM エッジアラインモード

アップカウント構成

TIMx\_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード \(601 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx\_CNT < TIMx\_CCRx の間はハイに、そうでない場合はローになります。TIMx\_CCRx の比較値が自動再ロード値 (TIMx\_ARR レジスタの) より大きい場合、OCxREF は“1”です。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 227](#) に TIMx\_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 227. エッジライン PWM 波形 (ARR=8)



## ダウンカウント構成

TIMx\_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード \(604 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx\_CNT > TIMx\_CCRx の間はローであり、そうでない場合はハイになります。TIMx\_CCRx の比較値が TIMx\_ARR の自動再ロード値より大きい場合、ocxref は 100% です。このモードでは、PWM 信号を生成することはできません。

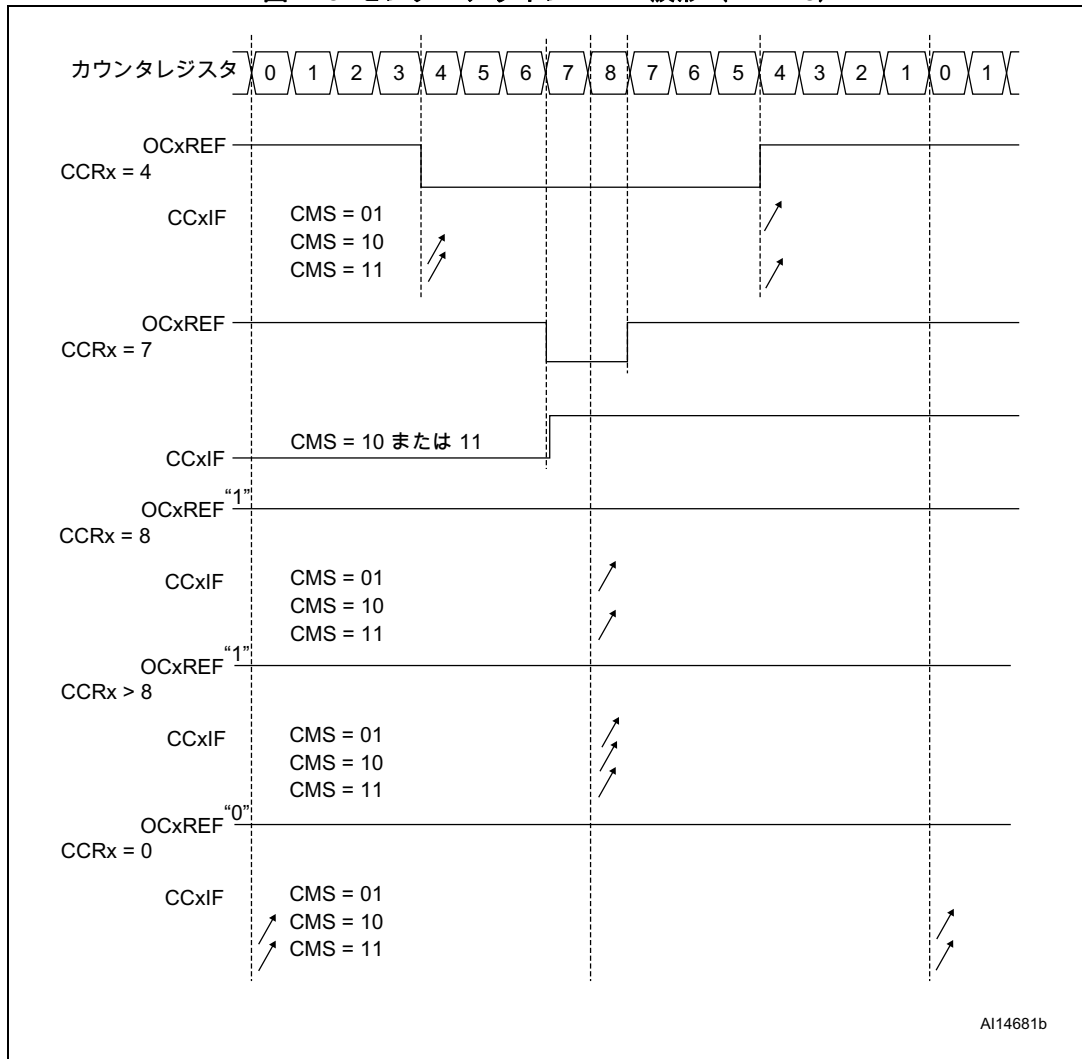
## PWM センターラインモード

センターラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”でないときにアクティブです（その他すべての設定は、OCxRef/OCx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx\_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターラインモード \(アップ/ダウンカウント\) \(607 ページ\)](#) を参照してください。

図 228 に、次の条件でのセンターライン PWM 波形の例を示します。

- TIMx\_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx\_CR1 レジスタの CMS=01 で選択されたセンターラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 228. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント :

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx\_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
  - 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx\_CNT > TIMx\_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
  - カウンタに 0 または TIMx\_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx\_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

## 21.3.10 非対称 PWM モード

非対称モードでは、プログラム可能な位相シフトによって2つのセンターアライン PWM 信号の生成を可能にします。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや位相シフトは TIMx\_CCRx レジスタペアで決定されます。1つ目のレジスタがアップカウント時の PWM を制御し、2つ目のレジスタがダウンカウント時の PWM を制御することで、PWM は PWM ハーフサイクルごとに調整されます。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

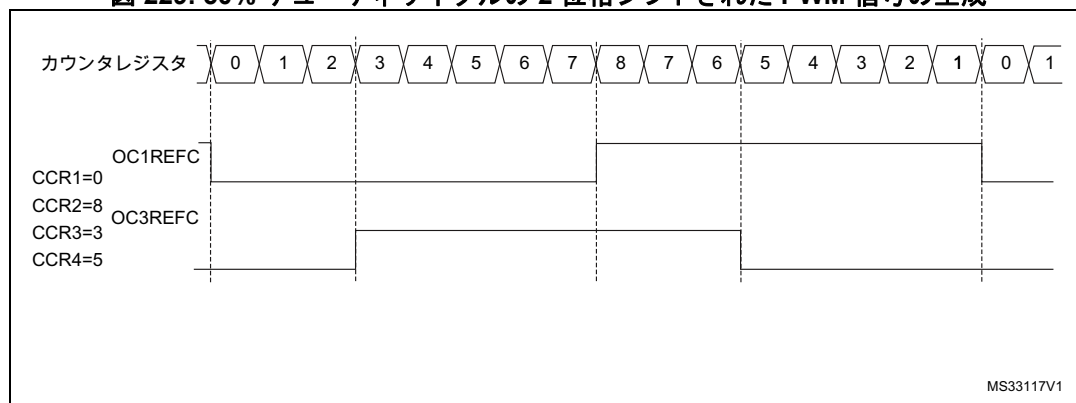
非対称 PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1110” (非対称 PWM モード 1) または“1111” (非対称 PWM モード 2) を書き込むことによって、2チャンネルごとに選択できます (CCR レジスタペアごとに1つの OCx 出力)。

**注：** OCxM[3:0] ビットフィールドは互換性を確保するために2つのパーツに分割され、最上位ビットと3つの最下位ビットとは隣接していません。

特定のチャンネルが非対称の PWM チャンネルとして使用されると、その2次チャンネルも使用できます。たとえば、OC1REFC 信号がチャンネル 1 (非対称 PWM モード 1) に生成されると、チャンネル 2 の OC2REF 信号、または非対称 PWM モード 2 の結果として得られる OC2REFC 信号を出力できます。

[図 229](#) は、非対称 PWM モードを使用して生成される信号の例を表します (チャンネル 1 から 4 は非対称 PWM モード 1 として設定されます)。

図 229. 50% デューティサイクルの 2 位相シフトされた PWM 信号の生成



## 21.3.11 組み合わせ PWM モード

組み合わせ PWM モードでは、2つのエッジアラインまたはセンターアライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は2つの TIMx\_CCRx レジスタで決定されます。結果として得られる信号 OCxREFC は、2つの PWM 基準信号の OR または AND による論理結合から成ります。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 によって制御されます。
- OC3REFC (または OC4REFC) は、TIMx\_CCR3 および TIMx\_CCR4 によって制御されます。

組み合わせ PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1100” (組み合わせ PWM モード 1) または“1101” (組み合わせ PWM モード 2) を書き込むことによって、2チャンネルごとに選択できます (CCR レジスタペアごとに1つの OCx 出力)。



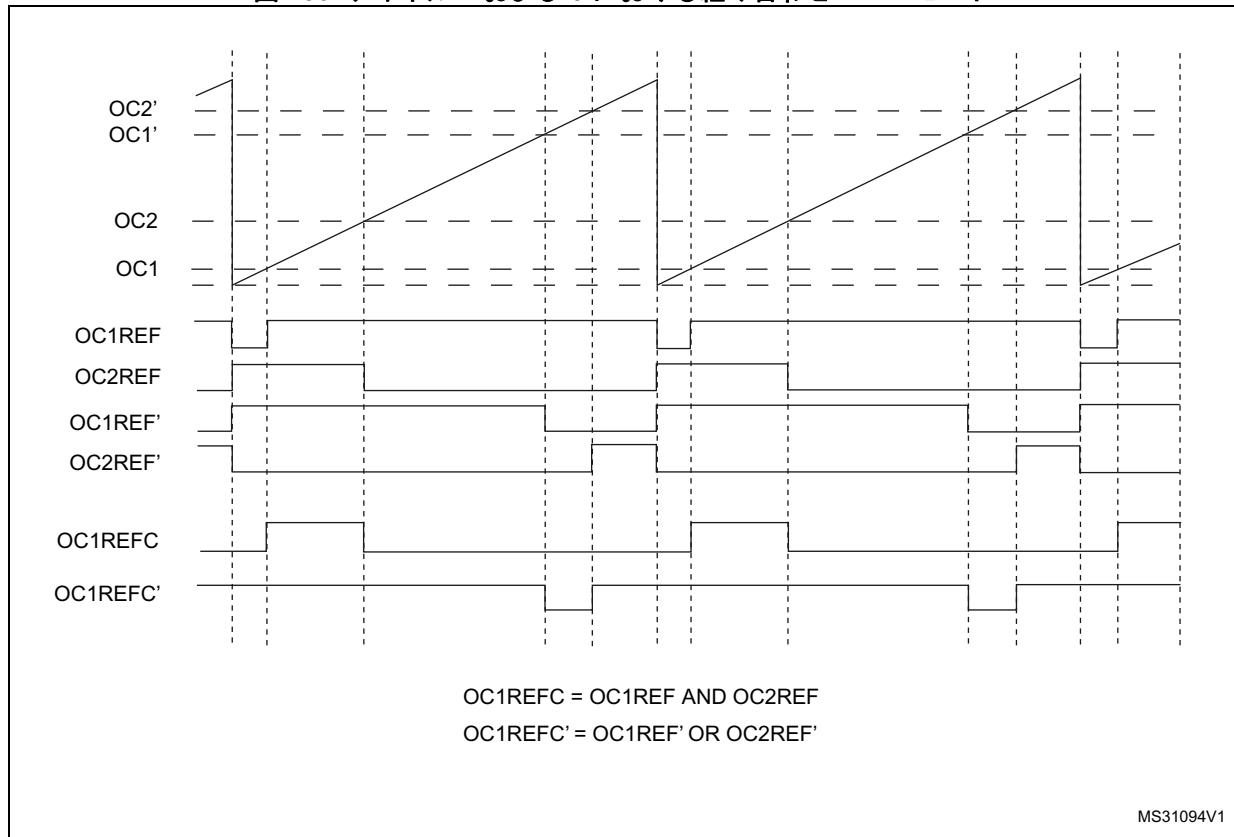
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、2 次チャンネルを反対の PWM モードに設定する必要があります (たとえば、1 つを組み合わせ PWM モード 1、もう 1 つを組み合わせ PWM モード 2 にします)。

**注:** OCxM[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。

図 230 は、次の設定で取得可能な非対称 PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合
- チャンネル 3 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 4 が PWM モード 1 で設定されている場合

図 230. チャンネル 1 および 3 における組み合わせ PWM モード



### 21.3.12 外部イベントによる OCxREF 信号のクリア

特定のチャンネルの OCxREF 信号は ocref\_clr\_int 入力にハイレベルを適用するとクリアされます (対応する TIMx\_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする)。OCxREF は、次の更新イベント (UEV) が発生するまで、ローレベルを保ちます。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

OCREF\_CLR\_INPUT は、TIMx\_SMCR レジスタで OCCS ビットを設定することで、OCREF\_CLR 入力と ETRF (フィルタ後の ETR) の間で選択できます。

特定のチャンネルの OCxREF 信号は ETRF 入力をハイレベルとする (対応する TIMx\_CCMRx レジスタの OCxCE イネーブルビットを 1 にセットする) ことでリセットすることができます。OCxREF は、次の更新イベント (UEV) が発生するまで、ローレベルを保ちます。

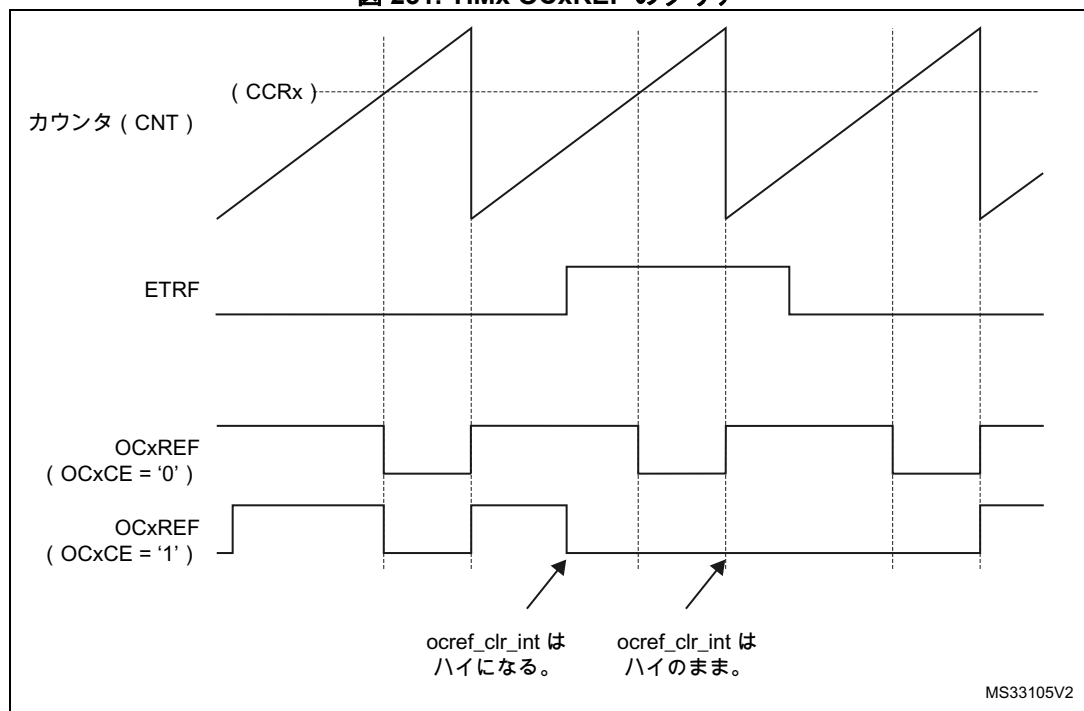
この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

たとえば、OCxREF 信号を比較回路の出力に接続して、電流処理に使用することができます。この場合、ETRF は次のように設定する必要があります。

1. 外部トリガプリスケアラをオフに保つ必要があります。すなわち、TIMx\_SMCR レジスタのビット ETPS[1:0] が 00 にクリアされます。
2. 外部クロックモード 2 を無効にする必要があります。すなわち、TIM1\_SMCR レジスタのビット ECE が 0 にクリアされます。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて設定できます。

図 231 に、OCxCE イネーブルビットの両方の値について、ETRF 入力が高くなったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 231. TIMx OCxREF のクリア



注: 100% デューティサイクルの PWM の場合 (CCRx > ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

## 21.3.13 ワンパルスモード

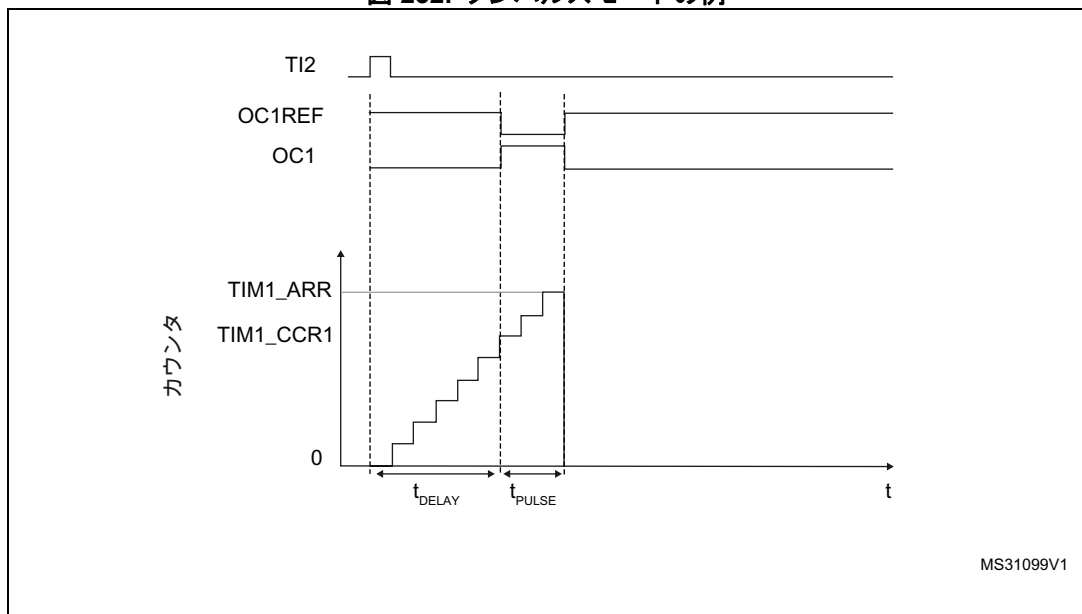
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- $CNT < CCRx \leq ARR$  (特に、 $0 < CCRx$ )

図 232. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が  $t_{PULSE}$  の正のパルスを遅延時間  $t_{DELAY}$  後に生成することもできます。

Ti2FP2 をトリガ 1 として使用します。

- TIMx\_CCMR1 レジスタに IC2S=01 を書き込むことによって、Ti2FP2 を Ti2 に配置します。
- Ti2FP2 は、立ち上がりエッジを検出して、TIMx\_CCER レジスタで CC2P=「0」と CC2NP=「0」を書き込みます。
- Ti2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx\_SMCR レジスタの TS ビットに 110 を書き込みます。
- Ti2FP2 を使用してカウンタを開始するために、TIMx\_SMCR レジスタの SMS ビットに「110」(トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- $t_{\text{DELAY}}$  は、TIMx\_CCR1 レジスタに書き込まれた値によって定義されます。
- $t_{\text{PULSE}}$  は、自動再ロード値と比較値の差 (TIMx\_ARR - TIMx\_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx\_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。オプションで、TIMx\_CCMR1 レジスタの OC1PE=1 と TIMx\_CR1 レジスタの ARPE=1 を書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx\_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので (シングルモード)、TIMx\_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx\_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

### 特殊なケース : OCx 高速イネーブル :

ワンパルスモードでは、Tix 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ( $t_{\text{DELAY min}}$ ) が制限されます。

最小遅延で波形を出力したい場合は、TIMx\_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較を考慮せずにトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

## 21.3.14 再トリガ可能なワンパルスモード (OPM)

このモードでは、トリガに応じてカウンタを開始して、プログラム可能な長さのパルスを生成できます。ただし、[セクション 21.3.13](#)で説明する再トリガ不可能なワンパルスモードについて、次のような違いがあります。

- パルスはトリガが発生し次第開始します (プログラム可能な遅延はありません)。
- パルスは、前のトリガが完了する前に新しいトリガが発生すると拡張されます。

タイマはスレーブモードである必要があり、このときビットは TIMx\_SMCR レジスタで SMS[3:0] = 「1000」 (リセットモードとトリガモードの組み合わせ)、および再トリガ可能な OPM モード 1 または 2 で OCxM[3:0] が「1000」または「1001」にセットされています。

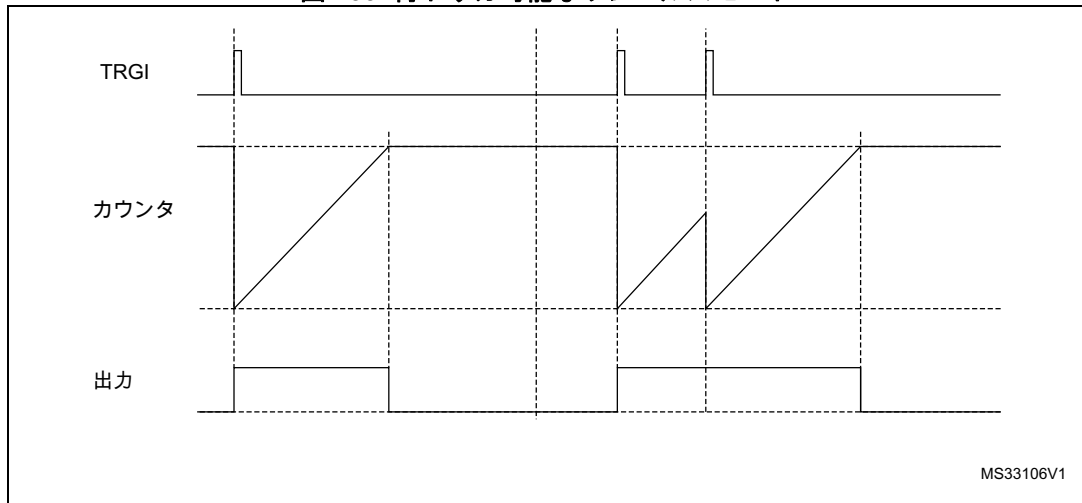
タイマをアップカウントモードで設定した場合、対応する CCRx を 0 にセットする必要があります (ARR レジスタによってパルス長がセットされます)。タイマをダウンカウントモードで設定した場合、CCRx は ARR 以上である必要があります。

**注 :** **再トリガ可能なワンパルスモードでは、CCxIF フラグは意味を持ちません。**

**OCxM[3:0] および SMS[3:0] ビットフィールドは互換性を確保するために 2 つのパーツに分割され、最上位ビットと 3 つの最下位ビットとは隣接していません。**

**このモードをセンターライン PWM モードと組み合わせて使用することはできません。TIMx\_CR1 では、CMS[1:0] = 00 にする必要があります。**

図 233. 再トリガ可能なワンパルスモード



MS33106V1

## 21.3.15 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx\_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS="001" を、TI1 エッジのみをカウントしている場合は SMS="010" を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS="011" を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx\_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。CC1NP と CC2NP はクリア状態に維持する必要があります。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。表 123 を参照してください。カウンタのクロックは、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択後は TI1 および TI2。フィルタされず、反転されない場合は TI1FP1=TI1、フィルタされず、反転されない場合は TI2FP2=TI2) の有効な変化によって駆動されます。ただし、カウンタ有効なことが前提となります (TIMx\_CR1 レジスタの CEN ビットが "1")。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx\_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx\_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx\_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、およびトリガ出力機能は、通常動作を続けます。

このモードでは、カウンタは直交エンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (TI1 と TI2 は同時に切り替わらないと想定しています)。

表 123. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみ カウント	ハイ	ダウン	アップ	カウントなし	カウントなし
	ロー	アップ	ダウン	カウントなし	カウントなし
TI2 のみ カウント	ハイ	カウントなし	カウントなし	アップ	ダウン
	ロー	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の 両方をカウント	ハイ	ダウン	アップ	アップ	ダウン
	ロー	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 234 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S= 01 (TIMx\_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S= 01 (TIMx\_CCMR2 レジスタ、TI2FP2 は TI2 に配置)
- CC1P = 0、CC1NP = 0、IC1F = 0000 (TIMx\_CCER レジスタ、TI1FP1 非反転、TI1FP1 = TI1)
- CC2P = 0、CC2NP = 0、IC2F = 0000 (TIMx\_CCER レジスタ、TI2FP2 非反転、TI2FP2 = TI2)
- SMS=011 (TIMx\_SMCR レジスタ、両方の入力が立ち上がり立ち下がりの両エッジでアクティブ)
- CEN=1 (TIMx\_CR1 レジスタ、カウンタ有効)

図 234. エンコーダインタフェースモードにおけるカウンタの動作例

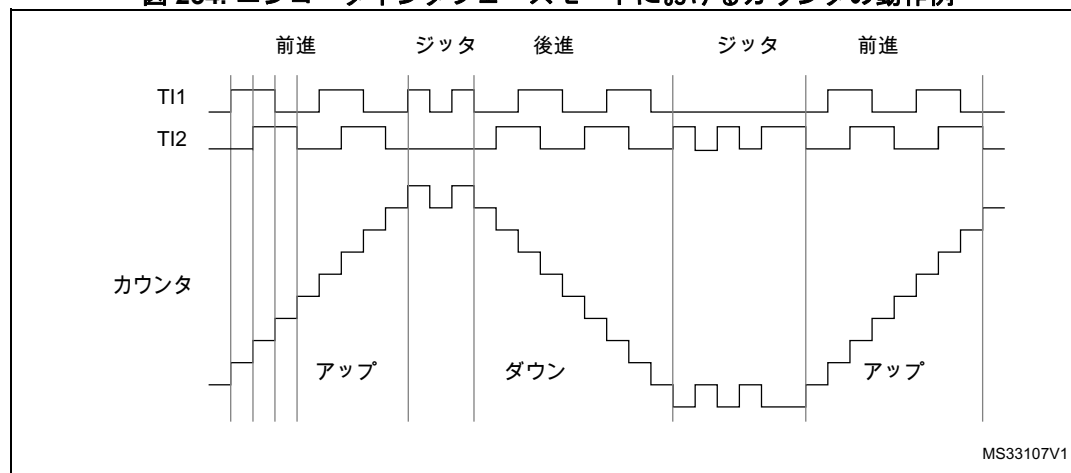
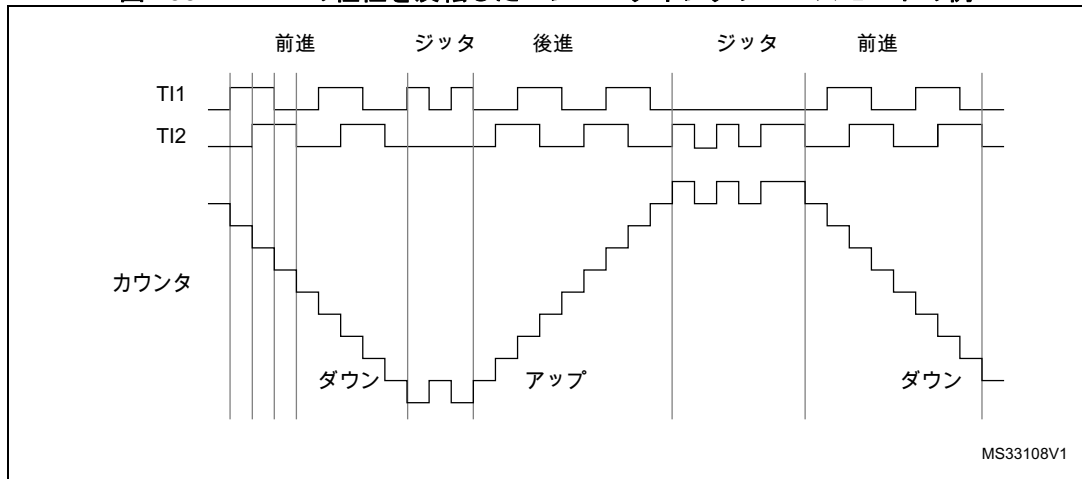


図 235 に、TI1FP1 の極性を反転したときのカウンタの動作を示します (上記と同じ設定ですが、CC1P=1)。

図 235. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならず、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

### 21.3.16 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。バックグラウンドタスク（カウンタの読み出し）と中断（更新の中断）との間で共有されている処理などによって生じる競合状態を避けることで、角速度の計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

32 ビットのタイマの実装で、IUFREMAP ビットがセットされている場合、カウンタのビット 31 は読み出しアクセス時に UIFCPY フラグによって上書きされます（カウンタの最上位ビットには書き込みモード時のみアクセス可能）。

### 21.3.17 タイマ入力 XOR 機能

TIM1xx\_CR2 レジスタの TI1S ビットを使用して、チャンネル 1 の入力フィルタを、TIMx\_CH1 から TIMx\_CH3 までの 3 つの入力ピンを組み合わせさせた XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。

この機能をホールセンサのインタフェースに使用した例を [セクション 20.3.24 : Interfacing with Hall sensors \(558 ページ\)](#) に示します。

## 21.3.18 タイマと外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

### スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx\_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx\_ARR、TIMx\_CCRx) が更新されます。

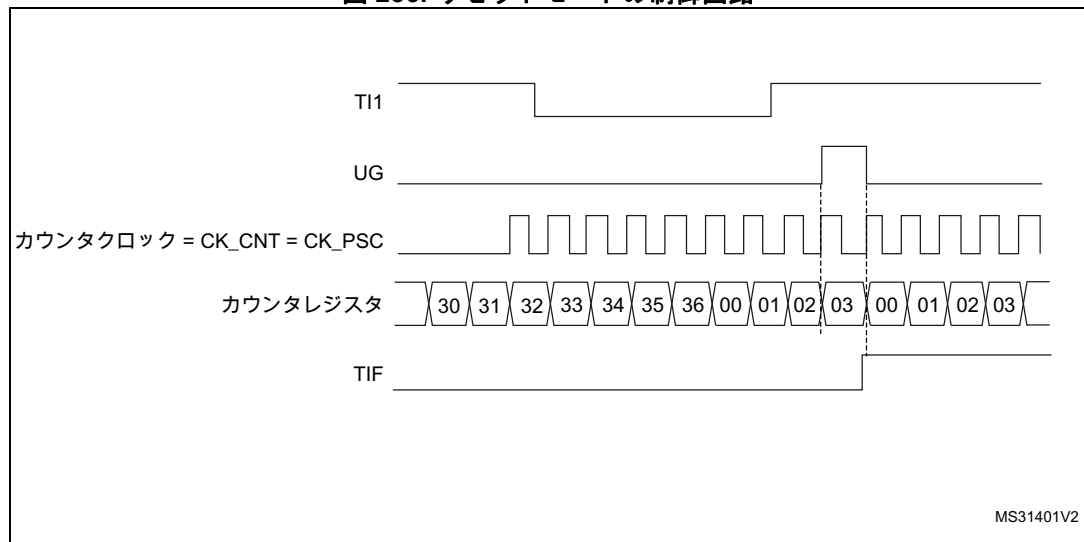
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01)。TIMx\_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx\_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx\_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx\_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 236. リセットモードの制御回路





## スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

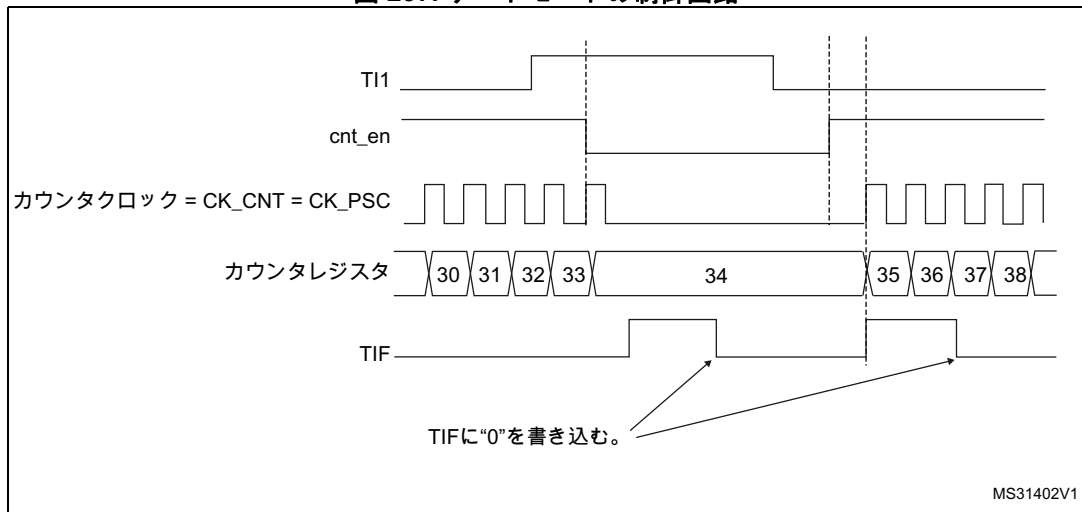
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

1. TI1 の高レベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01 ビット)。TIMx\_CCER レジスタで CC1P = 1 と CC1NP = 0 を書き込んで、極性を有効にします (そして、高レベルのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 が高レベルになると内部クロックでカウントを開始して、TI1 が低レベルになると停止します。TIMx\_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 237. ゲートモードの制御回路



1. ゲートモードはエッジではなくレベルに対して動作するため、CCxP=CCxNP=1 の設定 (立ち上がり立ち下がり両エッジの検出) はゲートモードでは意味がありません。

**注：** ゲートモードはエッジではなくレベルに対して動作するため、CCxP=CCxNP=1 の設定 (立ち上がり立ち下がり両エッジの検出) はゲートモードでは意味がありません。

## スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

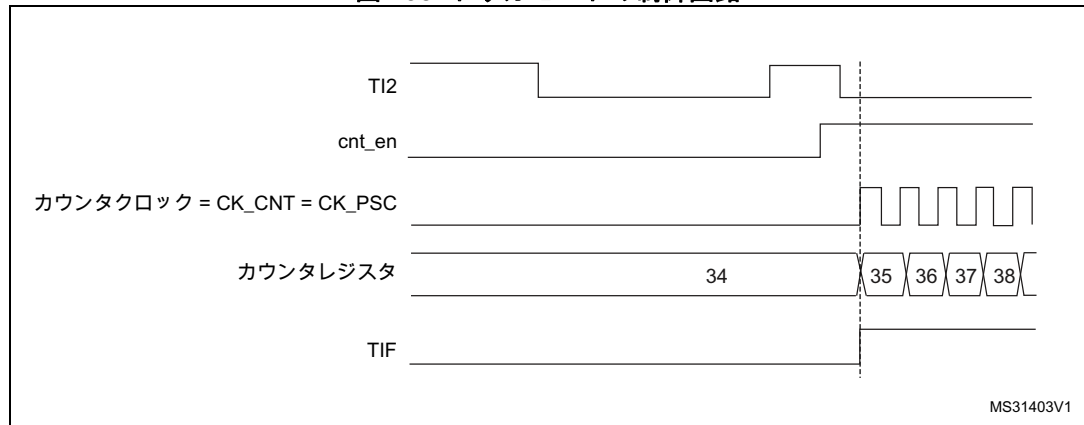
1. TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC2S=01)。TIMx\_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします (そして、高レベルのみを検出します)。

- TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 238. トリガモードの制御回路



### スレーブモード：リセットモードとトリガモードの組み合わせ

この場合、選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

### スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます (ただし、外部クロックモード 1 とエンコーダモードは除きます)。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます (リセットモード、ゲートモード、およびトリガモードで動作している場合)。TIMx\_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

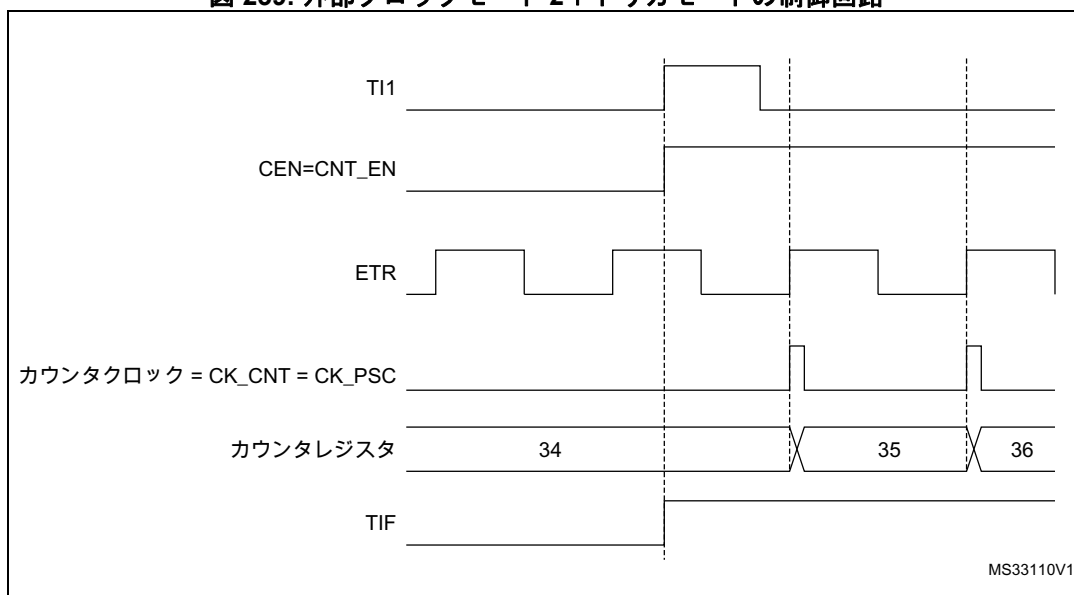
次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

1. TIMx\_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
  - ETF = 0000 : フィルタなし
  - ETPS=00 : プリスケール無効
  - ETP=0 : ETR の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。
2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
  - IC1F=0000 : フィルタなし。
  - キャプチャプリスケールはトリガには使用されないため、設定する必要はありません。
  - TIMx\_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
  - TIMx\_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
3. TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 239. 外部クロックモード 2+トリガモードの制御回路

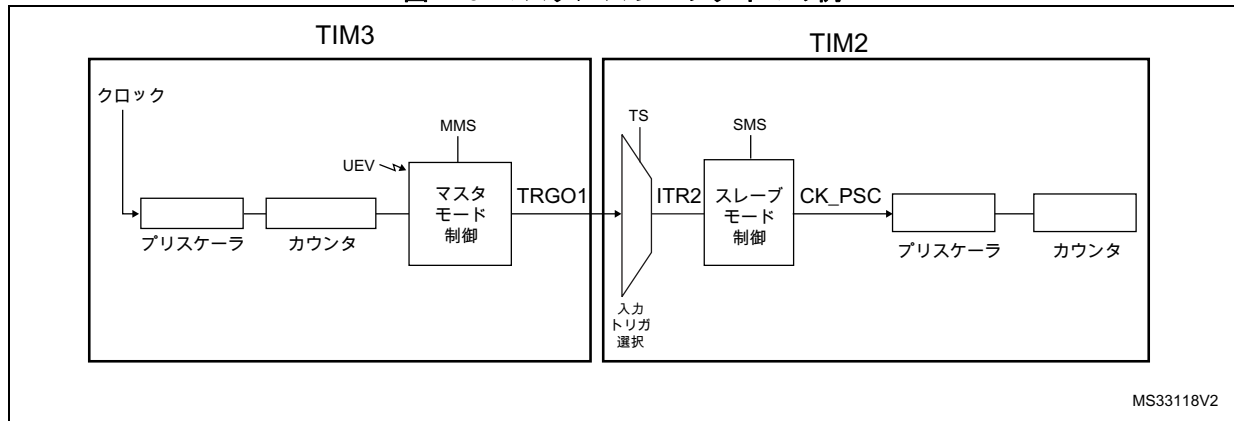


## 21.3.19 タイマの同期

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。マスターモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

図 240: マスタ/スレーブタイマの例に、トリガ選択およびマスターモード選択ブロックの概要を示します。

図 240. マスタ/スレーブタイマの例



## タイマを別のタイマのプリスケラとして使用する

たとえば、TIM3 が TIM2 のプリスケラとして動作するように設定できます。図 240 を参照してください。このためには、次の操作を行います。

1. 更新イベント UEV ごとに定期的なトリガ信号を出力するように、TIM 3 をマスタモードに設定します。TIM3\_CR2 レジスタの MMS=010 を書き込んだ場合、更新イベントが生成されるたびに、TRGO で立ち上がりエッジが出力されます。
2. TIM3 の TRGO 出力を TIM2 に接続するには、ITR2 を内部トリガとして使用して、TIM2 をスレーブモードで設定する必要があります。このためには、TIM2\_SMCR レジスタの TS ビットを使用します (TS=010 を書き込みます)。
3. 次に、スレーブモードコントローラを外部クロックモード 1 にします (TIM2\_SMCR レジスタの SMS=111 を書き込みます)。これにより TIM2 は、TIM3 の定期的なトリガ信号の立ち上がりエッジ (TIM3 カウンタのオーバーフローに対応しています) をクロックとして動作します。
4. 最後に、それぞれの CEN ビット (TIMx\_CR1 レジスタ) をセットすることによって、両方のタイマを有効にする必要があります。

**注：** TIM3 のトリガ出力として OCx が選択された場合 (MMS=1xx)、その立ち上がりエッジが TIM2 カウンタのクロックとして使用されます。

## タイマを使用して別のタイマを有効にする

この例では、TIM2 の有効化を、タイマ 3 の出力比較 1 で制御します。接続については、図 240 を参照してください。TIM2 は、TIM3 の OC1REF がハイのときのみ、分周された内部クロックをカウントします。両方のカウンタクロック周波数は、CK\_INT をプリスケラで 3 分周したものです ( $f_{CK\_CNT} = f_{CK\_INT}/3$ )。

1. TIM 3 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=100)。
2. TIM3 の OC1REF 波形を設定します (TIM3\_CCMR1 レジスタ)。
3. TIM3 から入力トリガを受け取るように TIM2 を設定します (TIM2\_SMCR レジスタの TS=010)。
4. TIM2 をゲートモードに設定します (TIM2\_SMCR レジスタの SMS=101)。
5. CEN ビット (TIM2\_CR1 レジスタ) に「1」を書き込んで、TIM2 を有効にします。
6. CEN ビット (TIM3\_CR1 レジスタ) に「1」を書き込んで、TIM3 を開始します。

**注：** カウンタ 2 のクロックはカウンタ 1 と同期しないので、このモードは TIM2 カウンタのイネーブル信号にのみ影響します。

図 241. TIM3 の OC1REF による TIM2 のゲート操作

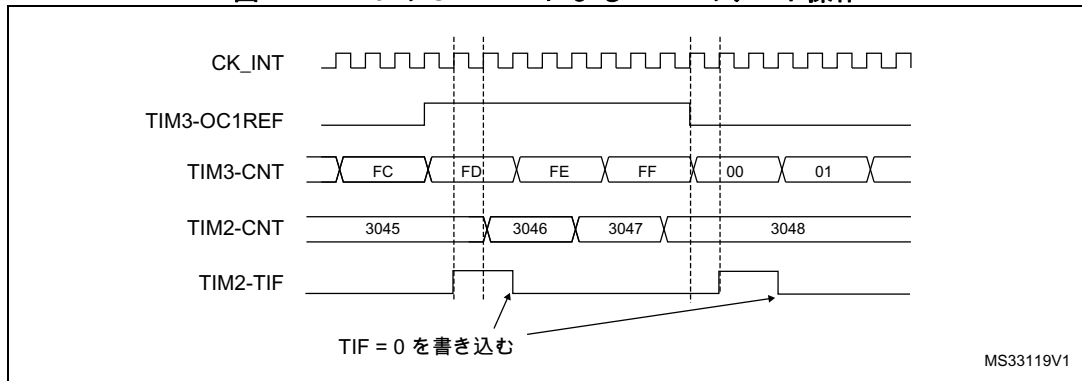
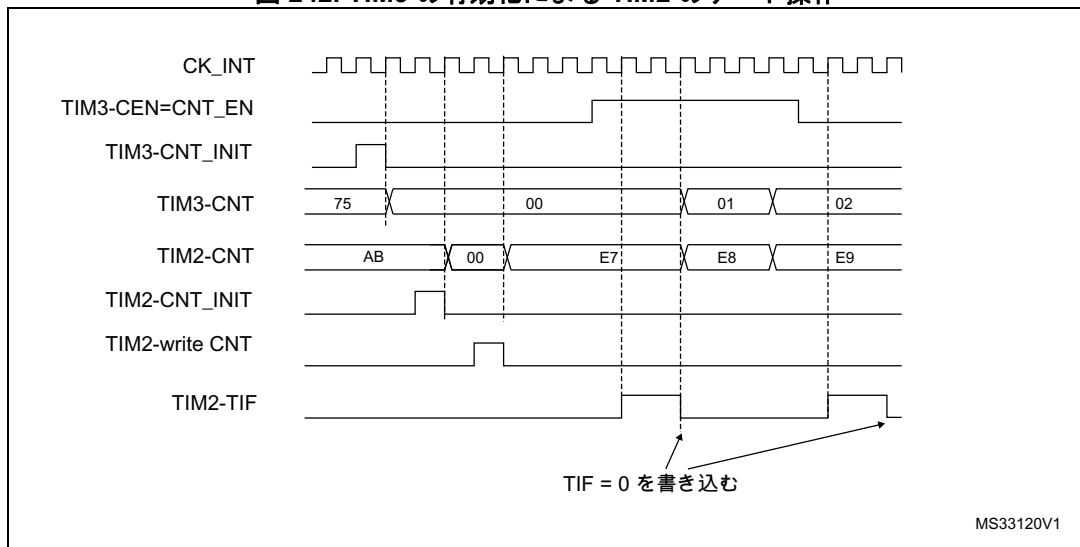


図 241 の例では、TIM2 のカウンタとプリスケアラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。TIM3 を開始する前に両方のタイマをリセットすることによって、特定の値から開始することが可能です。この場合、タイマカウンタに任意の値を書き込むことができます。TIMx\_EGR レジスタの UG ビットを使用して、ソフトウェアで容易にタイマをリセットできます。

次の例では（図 242 を参照）、TIM3 と TIM2 を同期させます。TIM3 はマスタであり、0 からカウントを開始します。TIM2 はスレーブであり、0xE7 から開始します。プリスケアラの分周比は両方のタイマで同じです。TIM3\_CR1 レジスタの CEN ビットに「0」を書き込むことによって TIM3 を無効にすると、TIM2 は停止します。

1. TIM3 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=100)。
2. TIM3 の OC1REF 波形を設定します (TIM3\_CCMR1 レジスタ)。
3. TIM3 から入力トリガを受け取るように TIM2 を設定します (TIM2\_SMCR レジスタの TS=010)。
4. TIM2 をゲートモードに設定します (TIM2\_SMCR レジスタの SMS=101)。
5. UG ビット (TIM3\_EGR レジスタ) に「1」を書き込んで、TIM3 をリセットします。
6. UG ビット (TIM2\_EGR レジスタ) に「1」を書き込んで、TIM2 をリセットします。
7. TIM2 カウンタ (TIM2\_CNTR) に「0xE7」を書き込んで、TIM2 を 0xE7 に初期化します。
8. CEN ビット (TIM2\_CR1 レジスタ) に「1」を書き込んで、TIM2 を有効にします。
9. CEN ビット (TIM3\_CR1 レジスタ) に「1」を書き込んで、TIM3 を開始します。
10. CEN ビット (TIM3\_CR1 レジスタ) に「0」を書き込んで、TIM3 を停止します。

図 242. TIM3 の有効化による TIM2 のゲート操作

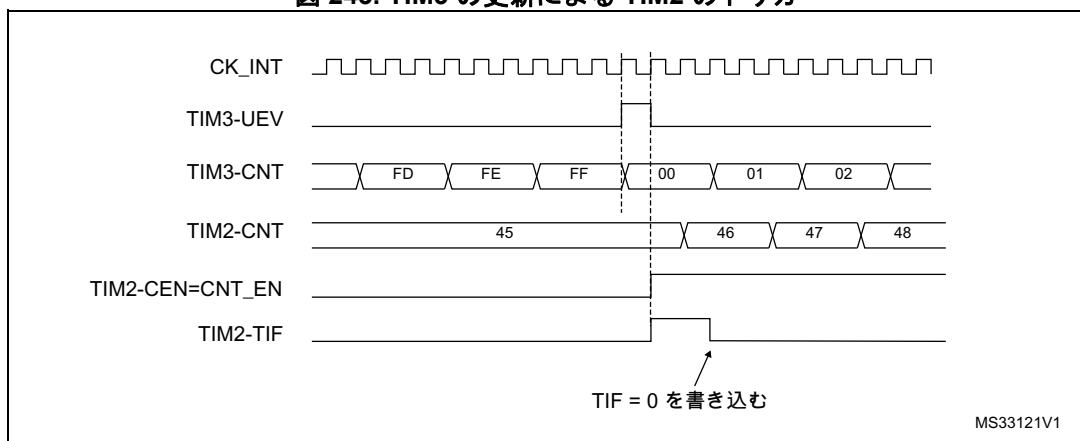


## タイマを使用して別のタイマを開始する

この例では、タイマ 3 の更新イベントによってタイマ 2 の有効化を設定します。接続については、[図 240](#) を参照してください。タイマ 1 によって更新イベントが生成されると、タイマ 2 は、分周された内部クロックで現在値（ゼロである必要はありません）からカウントを開始します。タイマ 2 がトリガ信号を受信すると、その CEN ビットが自動的にセットされ、カウンタは TIM2\_CR1 レジスタの CEN ビットに“0”が書き込まれるまでカウントします。両方のカウンタクロック周波数は、CK\_INT をプリスケータで 3 分周したものです ( $f_{CK\_CNT} = f_{CK\_INT}/3$ )。

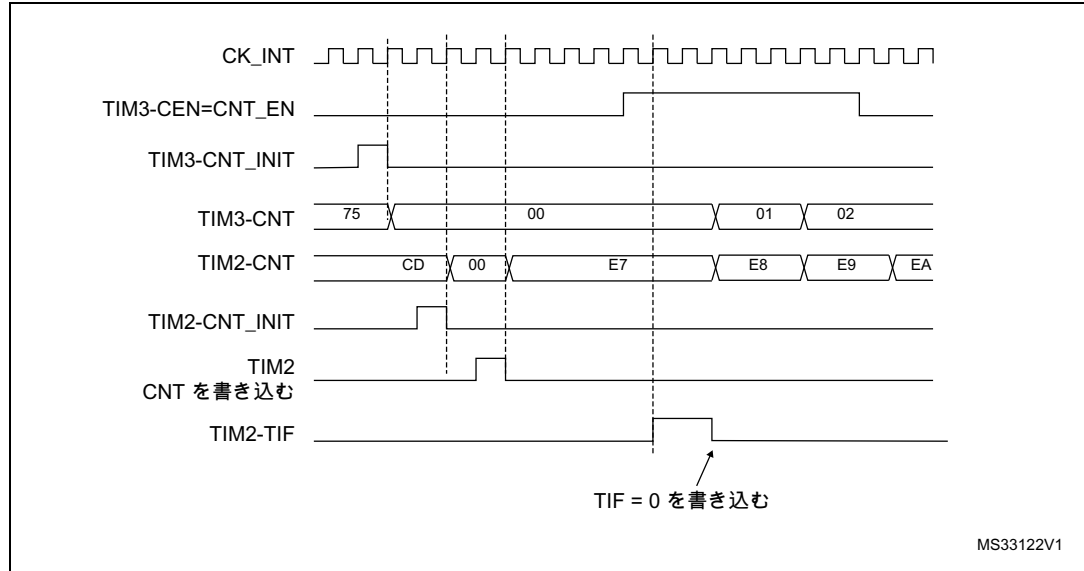
1. TIM3 をマスタモードに設定して、その更新イベント (UEV) をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=010)。
2. TIM3 の周期を設定します (TIM3\_ARR レジスタ)。
3. TIM3 から入力トリガを受け取るように TIM2 を設定します (TIM2\_SMCR レジスタの TS=010)。
4. TIM2 をトリガモードに設定します (TIM2\_SMCR レジスタの SMS=110)。
5. CEN ビット (TIM3\_CR1 レジスタ) に“1”を書き込んで、TIM3 を開始します。

図 243. TIM3 の更新による TIM2 のトリガ



前の例と同じように、カウントを開始する前に両方のカウンタを初期化することができます。図 244 は 図 243 と同じ設定ですが、ゲートモードではなくトリガモードでの動作を示します (TIM2\_SMCR レジスタの SMS=110)。

図 244. TIM3 の有効化による TIM2 のトリガ



外部トリガに対応して 2 つのタイマを同期して開始します。

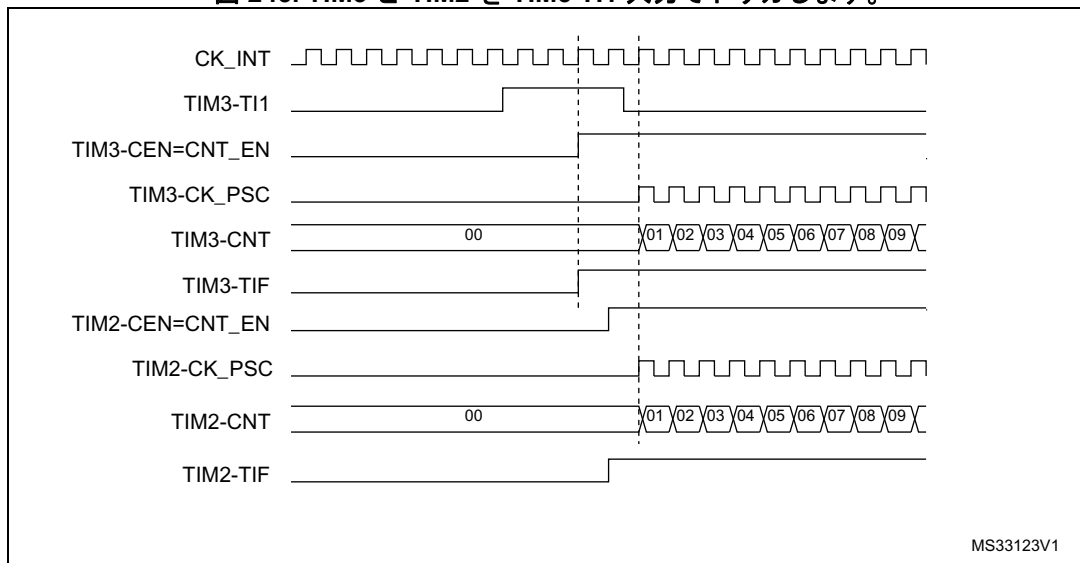
この例では、T11 入力の立ち上がりエッジで TIM3 を有効にし、TIM3 の有効で、TIM2 を有効にします。接続については、図 240 を参照してください。カウンタの同時性を確保するため、TIM3 はマスタ/スレーブモードに設定する必要があります (T11 に対してはスレーブ、TIM2 に対してはマスタ)。

1. TIM3 をマスタモードに設定して、その有効化をトリガ出力として送信します (TIM3\_CR2 レジスタの MMS=001)。
2. TIM3 をスレーブモードに設定して、T11 から入力トリガを受け取るようにします (TIM3\_SMCR レジスタの TS=100)。
3. TIM3 をトリガモードに設定します (TIM3\_SMCR レジスタの SMS=110)。
4. MSM=1 を書き込むことによって TIM3 をマスタ/スレーブモードに設定します (TIM3\_SMCR レジスタ)。
5. TIM3 から入力トリガを受け取るように TIM2 を設定します (TIM2\_SMCR レジスタの TS=000)。
6. TIM2 をトリガモードに設定します (TIM2\_SMCR レジスタの SMS=110)。

T11 (TIM3) で立ち上がりエッジが発生すると、両方のカウンタが同時に内部クロックによるカウントを開始し、両方の TIF フラグがセットされます。

**注:** この例では、両方のタイマが開始前に初期化されます (それぞれの UG ビットをセットすることによって)。両方のカウンタは 0 から開始しますが、カウンタレジスタ (TIMx\_CNT) に書き込むことによって、容易にオフセットを挿入できます。マスタ/スレーブモードでは、TIM3 の CNT\_EN と CK\_PSC の間に遅延が挿入されます。

図 245. TIM3 と TIM2 を TIM3 TI1 入力でトリガします。



注：スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

### 21.3.20 DMA バーストモード

TIMx タイマには、1つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、タイマの一部をソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx\_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト (バースト) を開始します。TIMx\_DMAR レジスタへの各書き込みは、実際にタイマレジスタの1つにリダイレクトされます。

TIMx\_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数 (ハーフワードまたはバイト) です。

TIMx\_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx\_CR1

00001 : TIMx\_CR2

00010 : TIMx\_SMCR

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。



1. 対応する DMA チャンネルを次のように設定します。
  - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
  - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
  - 転送データ数 = 3 とします (下の注を参照)。
  - サークュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。  
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタのUDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注 :

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

## 21.3.21 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M4<sup>®</sup>F コアは停止状態)、TIMx カウンタは、DBGMCU モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、停止します。詳細については、[セクション 33.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 21.4 TIM2/TIM3/TIM4 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 21.4.1 TIMx 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF RE-MAP	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
				r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デジタルフィルタ (ETR、Tlx) によって使用されるサンプリングクロックとの間の分周比を示します。

00:  $t_{DTS} = t_{CK\_INT}$

01:  $t_{DTS} = 2 \times t_{CK\_INT}$

10:  $t_{DTS} = 4 \times t_{CK\_INT}$

11: 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:5 **CMS** : センターアラインモード選択

00: エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。

01: センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。

10: センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。

11: センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

**注:** カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

0 : カウンタはアップカウンタとして使用されます。

1 : カウンタはダウンカウンタとして使用されます。

**注:** このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

**ビット 3 OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

**ビット 2 URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。

**ビット 1 UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケラは再初期化されます。

**ビット 0 CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
- 1 : カウンタは有効です。

**注 :** 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

## 21.4.2 TIMx 制御レジスタ 2 (TIMx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								r/w	r/w	r/w	r/w	r/w			

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TI1S** : TI1 選択

0 : TIMx\_CH1 ピンが TI1 入力に接続されます。

1 : TIMx\_CH1, CH2, および CH3 ピンが TI1 入力に接続されます (XOR 接続)。参照先 : [セクション 20.3.24 : ホールセンサとのインタフェース \(552 ページ\)](#)

ビット 6:4 **MMS** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、GEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在します (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる時 (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。(TRGO)

100 : 比較 - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : 比較 - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

110 : 比較 - OC3REF 信号がトリガ出力 (TRGO) として使用されます。

111 : 比較 - OC4REF 信号がトリガ出力 (TRGO) として使用されます。

**注 :** **スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。**

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2:0 予約済みであり、リセット値に保持する必要があります。

## 21.4.3 TIMx スレーブモード制御レジスタ (TIMx\_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SMS[3]** : スレーブモード選択 - ビット 3

SMS 説明を参照 - ビット 2:0

ビット 15 **ETP** : 外部トリガ極性

このビットは、ETR と  $\overline{\text{ETR}}$  のいずれがトリガ動作に使用されるかを選択します。

0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。

1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブになります。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

0 : 外部クロックモード 2 は無効です。

1 : 外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

1 : ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=111)。

2 : 外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません (TS ビットが 111 でないことが必要)。

3 : 外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力は ETRF です。

ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、最大でも CK\_INT 周波数の 1/4 でなければなりません。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するとき役に立ちます。

00 : プリスケアラオフ

01 : ETRP 周波数は 2 分周されます。

10 : ETRP 周波数は 4 分周されます。

11 : ETRP 周波数は 8 分周されます。

**ビット 11:8 ETF[3:0] : 外部トリガフィルタ**

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

0001 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=2

0010 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=4

0011 :  $f_{SAMPLING}=f_{CK\_INT}$ , N=8

0100 :  $f_{SAMPLING}=f_{DTS}/2$ , N=6

0101 :  $f_{SAMPLING}=f_{DTS}/2$ , N=8

0110 :  $f_{SAMPLING}=f_{DTS}/4$ , N=6

0111 :  $f_{SAMPLING}=f_{DTS}/4$ , N=8

1000 :  $f_{SAMPLING}=f_{DTS}/8$ , N=6

1001 :  $f_{SAMPLING}=f_{DTS}/8$ , N=8

1010 :  $f_{SAMPLING}=f_{DTS}/16$ , N=5

1011 :  $f_{SAMPLING}=f_{DTS}/16$ , N=6

1100 :  $f_{SAMPLING}=f_{DTS}/16$ , N=8

1101 :  $f_{SAMPLING}=f_{DTS}/32$ , N=5

1110 :  $f_{SAMPLING}=f_{DTS}/32$ , N=6

1111 :  $f_{SAMPLING}=f_{DTS}/32$ , N=8

**ビット 7 MSM : マスタ/スレーブモード**

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)。予約済み
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)。予約済み
- 100 : TI1 エッジ検出回路 (TI1F\_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)
- 111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、表 124:TIMx 内部トリガ接続(647 ページ)を参照してください。

**注:** 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 **OCCS** : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

- 0 : OCREF\_CLR\_INT は、OCREF\_CLR 入力に接続されています。
- 1 : OCREF\_CLR\_INT は、ETRF に接続されています。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力を選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

- 0000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。
- 0001 : エンコーダモード 1 - カウンタは、TI2FP2 のレベルに応じて、TI1FP1 のエッジでカウントアップ/ダウンします。
- 0010 : エンコーダモード 2 - カウンタは、TI1FP1 のレベルに応じて、TI2FP2 のエッジでカウントアップ/ダウンします。
- 0011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。
- 0100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
- 0101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。
- 0110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。
- 0111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。
- 1000 : リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力の立ち上がりエッジ (TRGI) カウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

**注:** トリガ入力として TI1F\_ED が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F\_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

**注:** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 124. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8



## 21.4.4 TIMx DMA／割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rW		rW	rW	rW	rW	rW		rW		rW	rW	rW	rW	rW

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IE** : キャプチャ／比較 4 割り込みイネーブル

0 : CC4 割り込みは無効です。

1 : CC4 割り込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割り込みイネーブル

0 : CC3 割り込みは無効です。

1 : CC3 割り込みは有効です。



ビット 2 **CC2IE** : キャプチャ/比較 2 割り込みイネーブル

- 0 : CC2 割り込みは無効です。
- 1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル

- 0 : CC1 割り込みは無効です。
- 1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

- 0 : 更新割り込みは無効です。
- 1 : 更新割り込みは有効です。

## 21.4.5 TIMx ステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ  
CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ  
CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ  
CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ  
このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。"0"を書き込むことによってソフトウェアによってクリアされます。  
0 : オーバーキャプチャは検出されていません。  
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIF** : トリガ割り込みフラグ  
このフラグは、トリガイイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。  
0 : トリガイイベントは発生していません。  
1 : トリガ割り込みが保留中です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IF** : キャプチャ/比較 4 割り込みフラグ  
CC1IF の説明を参照してください。

ビット 3 **CC3IF** : キャプチャ/比較 3 割り込みフラグ  
CC1IF の説明を参照してください。



ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ  
CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合** : このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます。センターアラインモード (TIMx\_CR1 レジスタの CMS ビットの説明を参照) および再トリガ可能なワンパルスモードでは、例外もあります。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。

**CC1 チャンネルが入力として設定されている場合** : このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (選択された極性に一致するエッジが IC1 で検出されました)。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

オーバーフローまたはアンダーフロー時 (TIM2 から TIM4)、および TIMx\_CR1 レジスタの UDIS=0 である場合。

TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイイベントによって CNT が再初期化されたとき (同期制御レジスタの説明を参照)。

## 21.4.6 TIMx イベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									W		W	W	W	W	W

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx\_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4G** : キャプチャ/比較 4 生成

CC1G の説明を参照してください。

ビット 3 **CC3G** : キャプチャ/比較 3 生成

CC1G の説明を参照してください。

ビット 2 **CC2G** : キャプチャ/比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ/比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ/比較イベントが生成されます。

**CC1 チャンネルが出力として設定されている場合 :**

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

**CC1 チャンネルが入力として設定されている場合 :**

カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx\_ARR) をとります。

## 21.4.7 TIMx キャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
							Res.								Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]		OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]		OC1PE	OC1FE	CC1S[1:0]			
IC2F[3:0]		IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]		IC1PSC[1:0]		CC1S[1:0]					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

### 出力比較モード

ビット 31:25 予約済み、常に 0 として読み出されます。

ビット 24 **OC2M[3]** : 出力比較 2 モード - ビット 3

ビット 23:17 予約済み、常に 0 として読み出されます。

ビット 16 **OC1M[3]** : 出力比較 1 モード - ビット 3

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 6:4 の OC1M 説明を参照

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC2S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : OC1Ref は ETRF 入力の影響を受けません。

1 : OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

0001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

0010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

0011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

0100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

0101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

0110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合は非アクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

0111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx\_CNT > TIMx\_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

1000 : 再トリガ可能な OPM モード 1 - アップカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。

1001 : 再トリガ可能な OPM モード 2 - アップカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはインアクティブです。その後、PWM モード 2 と同様に比較が行われ、チャンネルは次の更新時に再びインアクティブになります。ダウンカウントモードでは、TRGI 信号でトリガイベントを検出するまでチャンネルはアクティブです。その後、PWM モード 1 と同様に比較が行われ、チャンネルは次の更新時に再びアクティブになります。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 OR です。

1101 : 組み合わせ PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 AND です。

1110 : 非対称 PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

1111 : 非対称 PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、カウンタがカウントアップするときに OC1REF を出力し、カウントダウンするときに OC2REF を出力します。

**注 :** 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2: PWM モードでは、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときのみ、OCREF のレベルが変化します。

**ビット 3 OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

- 注 :**
- 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。
  - 2: PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

**ビット 2 OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

**ビット 1:0 CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

- 注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済み、常に 0 として読み出されます。

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケーラ

**ビット 9:8 CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

- 注 :** CC2S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

0001 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2

0010 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4

0011 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8

0100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2

0101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2

0110 :  $f_{SAMPLING} = f_{DTS}/4$ 、N=6

0111 :  $f_{SAMPLING} = f_{DTS}/4$ 、N=8

1000 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8

1001 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8

1010 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1011 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

1110 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

1111 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E=0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。**

## 21.4.8 TIMx キャプチャ/比較モードレジスタ 2 (TIMx\_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000

上記の CCMR1 レジスタの説明を参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							Res.								Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

### 出力比較モード

ビット 31:25 予約済み、常に 0 として読み出されます。

ビット 24 **OC4M[3]** : 出力比較 2 モード - ビット 3

ビット 23:17 予約済み、常に 0 として読み出されます。

ビット 16 **OC3M[3]** : 出力比較 1 モード - ビット 3

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

OC1M の説明 (TIMx\_CCMR1 レジスタのビット 6:4) を参照

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード

OC1M の説明 (TIMx\_CCMR1 レジスタのビット 6:4) を参照



ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC3S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済み、常に 0 として読み出されます。

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケール

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC4S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC4E=0) のときのみ書き込み可能です。

ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケール

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC3S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC3E=0) のときのみ書き込み可能です。

## 21.4.9 TIMx キャプチャ/比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
r/w		r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w	r/w		r/w	r/w



- ビット 15 **CC4NP** : キャプチャ/比較 4 出力極性  
CC1NP の説明を参照してください。
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **CC4P** : キャプチャ/比較 4 出力極性  
CC1P の説明を参照してください。
- ビット 12 **CC4E** : キャプチャ/比較 4 出カインェブル。  
CC1E の説明を参照してください。
- ビット 11 **CC3NP** : キャプチャ/比較 3 出力極性  
CC1NP の説明を参照してください。
- ビット 10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **CC3P** : キャプチャ/比較 3 出力極性  
CC1P の説明を参照してください。
- ビット 8 **CC3E** : キャプチャ/比較 3 出カインェブル。  
CC1E の説明を参照してください。
- ビット 7 **CC2NP** : キャプチャ/比較 2 出力極性  
CC1NP の説明を参照してください。
- ビット 6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **CC2P** : キャプチャ/比較 2 出力極性  
CC1P の説明を参照してください。
- ビット 4 **CC2E** : キャプチャ/比較 2 出カインェブル。  
CC1E の説明を参照してください。
- ビット 3 **CC1NP** : キャプチャ/比較 1 出力極性  
**CC1 チャンネルが出力として設定されている場合** : この場合、CC1NP はクリアされたままでなければなりません。  
**CC1 チャンネルが入力として設定されている場合** : このビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

**CC1 チャンネルが出力として設定されている場合 :**

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :** CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。

00 : 非反転/立ち上がりエッジ

回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。

01 : 反転/立ち下がりエッジ

回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ

回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル。

**CC1 チャンネルが出力として設定されている場合 :**

0 : オフ - OC1 はアクティブではありません。

1 : オン - OC1 信号は、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :** このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

表 125. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx=0, OCx_EN=0)
1	OCx=OCxREF + 極性, OCx_EN=1

**注 :** 標準 OCx チャンネルに接続されている外部 IO ピンの状態は、OCx チャンネルの状態と、GPIO および AFIO レジスタに依存します。

## 21.4.10 TIMx カウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31] または UIFCPY	CNT[30:16] (タイマに依存)														
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w



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

ビット 31 値は TIMx\_CR1 の UIFREMAP に依存します。

UIFREMAP = 0 の場合 :

**CNT[31]** : カウンタ値の最上位ビット (TIM2)

他のタイマで予約済み

UIFREMAP = 1 の場合 :

**UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー

ビット 30:16 **CNT[30:16]** : カウンタ値の最上位部分 (TIM2)

ビット 15:0 **CNT[15:0]** : カウンタ値の最下位部分

## 21.4.11 TIMx プリスケーラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **PSC[15:0]** : プリスケーラ値

カウンタクロック周波数 CK\_CNT は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケーラレジスタにロードされる値を含みます (更新イベントには、TIMx\_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

## 21.4.12 TIMx 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0xFFFF FFFF

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

ビット 31:16 **ARR[31:16]** : 自動再ロード値上位ビット (TIM2)

ビット 15:0 **ARR[15:0]** : 自動再ロードプリスケーラ値下位ビット

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 21.3.1 : タイムベースユニット \(599 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 21.4.13 TIMx キャプチャ/比較モードレジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR1[31:16]** : キャプチャ/比較 1 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値下位ビット

**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**チャンネル CC1 が入力として設定されている場合 :**

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 21.4.14 TIMx キャプチャ/比較モードレジスタ 2 (TIMx\_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR2[31:16]** : キャプチャ/比較 2 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR2[15:0]** : キャプチャ/比較 2 値下位ビット

**CC2 チャンネルが出力として設定されている場合 :**

CCR2 は、実際のキャプチャ/比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR1 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 2 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC2 出力に送信される値を含みます。

**CC2 チャンネルが入力として設定されている場合 :**

CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。



## 21.4.15 TIMx キャプチャ/比較モードレジスタ 3 (TIMx\_CCR3)

アドレスオフセット : 0x3C

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR3[31:16]** : キャプチャ/比較 3 値上位ビット (TIM2 および TIM5)

ビット 15:0 **CCR3[15:0]** : キャプチャ/比較値下位ビット

**CC3 チャンネルが出力として設定されている場合 :**

CCR3 は、実際のキャプチャ/比較 3 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 3 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC3 出力に送信される値を含みます。

**チャンネル CC3 が入力として設定されている場合 :**

CCR3 は、最後の入力キャプチャ 3 イベント (IC3) によって転送されたカウンタ値です。

## 21.4.16 TIMx キャプチャ/比較モードレジスタ 4 (TIMx\_CCR4)

アドレスオフセット : 0x40

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16] (タイマに依存)															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **CCR4[31:16]** : キャプチャ/比較 4 値上位ビット (TIM2)

ビット 15:0 **CCR4[15:0]** : キャプチャ/比較値下位ビット

1. **CC4 チャンネルが出力として設定されている場合 (CC4S ビット) :**

CCR4 は、実際のキャプチャ/比較 4 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 4 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC4 出力に送信される値を含みます。

2. **CC4 チャンネルが入力として設定されている場合 (TIMx\_CCMR4 レジスタの CC4S ビット) :**

CCR4 は、最後の入力キャプチャ 4 イベント (IC4) によって転送されたカウンタ値です。

## 21.4.17 TIMx DMA 制御レジスタ (TIMx\_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			r/w	r/w	r/w	r/w	r/w				r/w	r/w	r/w	r/w	r/w

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送回数 (タイマは、TIMx\_DMAR アドレスに対して読み出したまたは書き込みアクセスが行われるときにバースト転送を認識します) を指定します。

- 00000 : 1 回転送
- 00001 : 2 回転送、
- 00010 : 3 回転送、
- ...
- 10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

- 例 :
- 00000 : TIMx\_CR1
  - 00001 : TIMx\_CR2
  - 00010 : TIMx\_SMCR
  - ...

例 : 次の転送を考えます : DBL = 7 回転送 かつ DBA = TIMx\_CR1。この場合、転送は、TIMx\_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

## 21.4.18 完全転送の TIMx DMA アドレス (TIMx\_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

DMAB[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **DMAB[15:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(TIMx\_CR1 \text{ アドレス}) + (DBA + DMA \text{ インデックス}) \times 4$$

ここで、TIMx\_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx\_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx\_DCR 内で設定)。



## 21.4.19 TIMx レジスタマップ

TIMx レジスタは、次の表のようにマップされます。

表 126. TIM2/TIM3/TIM4レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIMx_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIFREMAP	Res	Res	Res	ARPE	CMS [1:0]	DIR	OPM	URS	UDIS	CEN		
	リセット値																									0	0	0	0	0	0	0	0	
0x04	TIMx_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	T11S	MMS[2:0]	CCDS	Res	Res	Res			
	リセット値																									0	0	0	0	0				
0x08	TIMx_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SMS[3]	ETP	ECE	ETPS [1:0]	Res	Res	Res	Res	Res	MSM	TS[2:0]	OCSS	Res	Res	Res			
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0C	TIMx_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res	TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE	
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	TIMx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	リセット値																											0	0	0	0	0	0	
0x14	TIMx_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TG	Res	CC4G	CC3G	CC2G	CC1G	UG	
	リセット値																										0	0	0	0	0	0	0	
0x18	TIMx_CCMR1 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2CE	OC2M [2:0]	Res	Res	Res	Res	Res	OC1CE	OC1M [2:0]	OC1PE	OC1FE	OC1S [1:0]				
	リセット値									0								0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	TIMx_CCMR1 入力キャプチャモード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]	IC2 PSC [1:0]	CC2S [1:0]	Res	Res	Res	IC1F[3:0]	IC1 PSC [1:0]	CC1S [1:0]						
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0			
0x1C	TIMx_CCMR2 出力比較モード	Res	Res	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	O24CE	OC4M [2:0]	Res	Res	Res	Res	Res	OC3CE	OC3M [2:0]	OC3PE	OC3FE	CC3S [1:0]				
	リセット値									0								0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	TIMx_CCMR2 入力キャプチャモード	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F[3:0]	IC4 PSC [1:0]	CC4S [1:0]	Res	Res	Res	IC3F[3:0]	IC3 PSC [1:0]	CC3S [1:0]						
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0			
0x20	TIMx_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res		
	リセット値																																	







## 22 基本タイマ (TIM6/TIM7) :

### 22.1 TIM6/TIM7 の概要

基本タイマ TIM6 および TIM7 は、プログラマブルなプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

このタイマは、タイムベース生成を目的とした汎用タイマとして使用できますが、特にデジタルアナログコンバータ (DAC) の駆動にも使用されます。実際、これらのタイマは内部で DAC に接続されており、トリガ出力を通じて駆動できます。

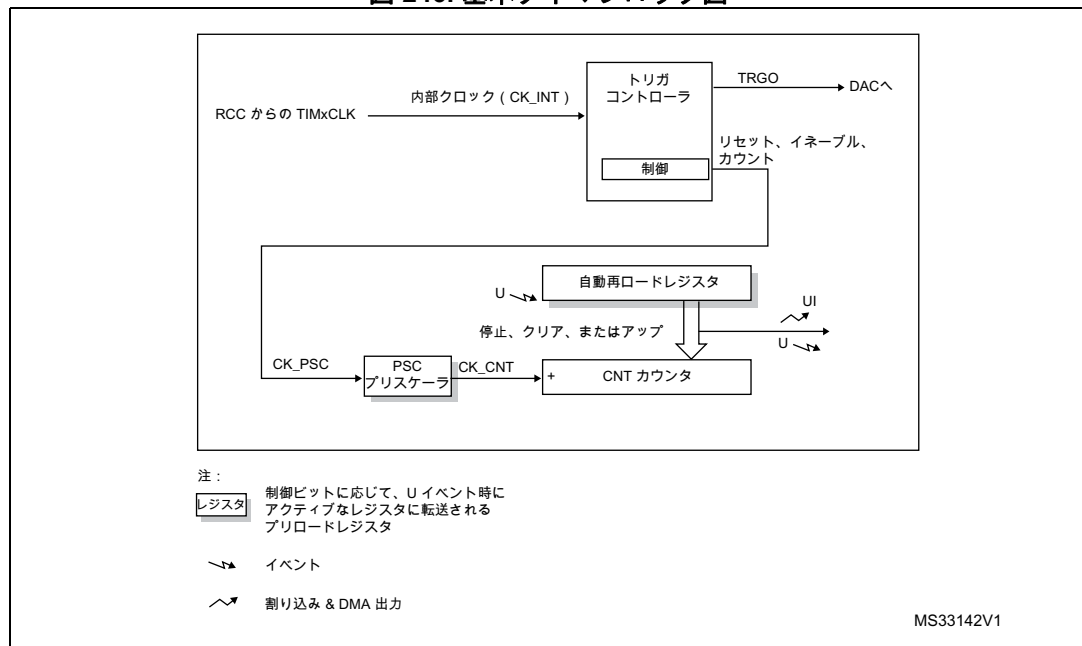
タイマは完全に独立していて、いかなるリソースも共有しません。

### 22.2 TIM6/TIM7 の主な機能

基本タイマ (TIM6/TIM7) の機能は、次のとおりです。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケラ (動作中も変更可能) で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- DAC をトリガする同期回路
- 更新イベント時の割り込み/DMA 生成 : カウンタオーバーフロー

図 246. 基本タイマブロック図



## 22.3 TIM6/TIM7 の機能詳細

### 22.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC)
- 自動再ロードレジスタ (TIMx\_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きはプリロードレジスタへのアクセスとなります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント UEV ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 である場合、カウンタがオーバーフロー値に達すると、更新イベントが送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときのみ、プリスケアラ出力 CK\_CNT から供給されます。

実際のカウンタイネーブル信号 CNT\_EN は、CEN の 1 クロックサイクル後にセットされます。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。TIMx\_PSC 制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 247](#) と [図 248](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 247. プリスケータ分周比が 1 から 2 に変化したときのカウンタのタイミング図

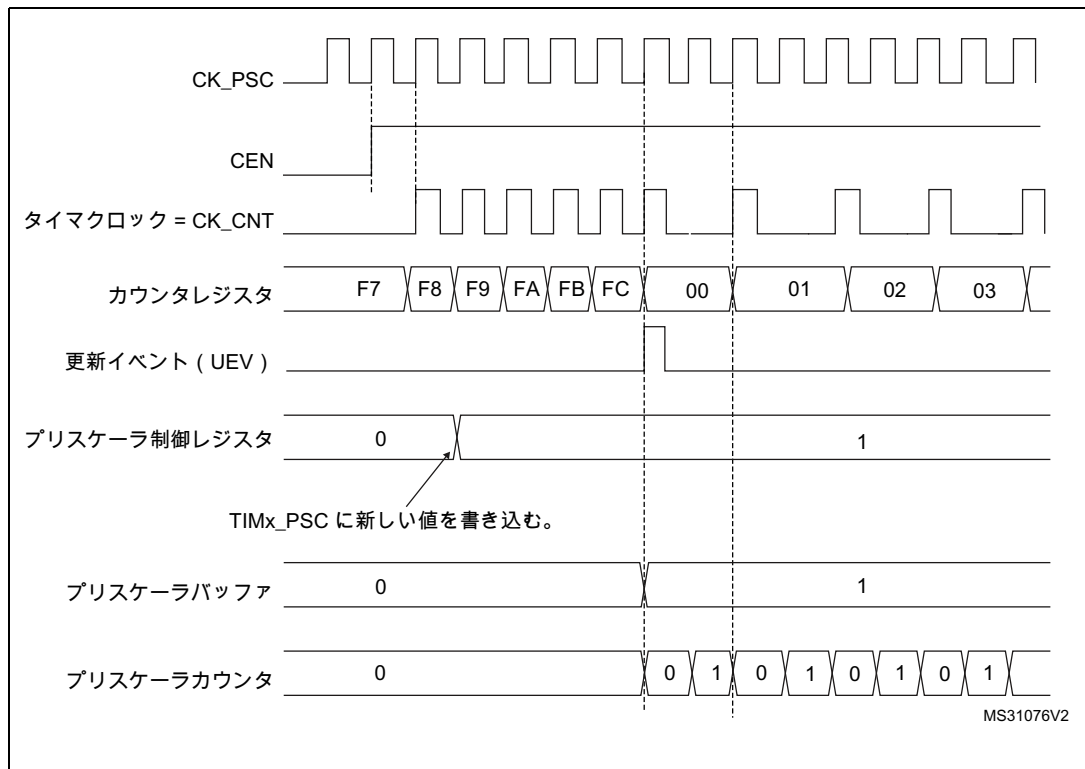
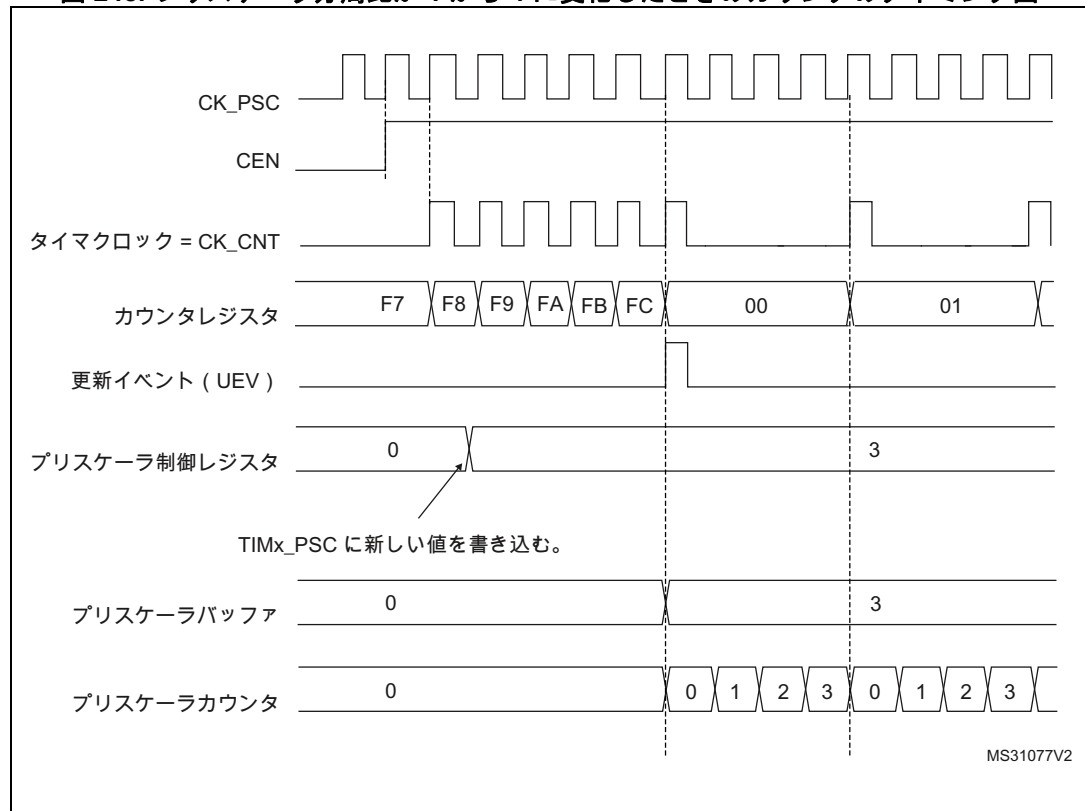


図 248. プリスケータ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 22.3.2 カウントモード

カウンタは、0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントした後、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、TIMx\_EGR レジスタの UG ビットをセットすることによって (ソフトウェアで、または、スレーブモードコントローラを使用して) 生成できます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。このようにすると、UDIS ビットに 0 が書き込まれるまで更新イベントは発生しませんが、カウンタとプリスケアラカウンタは両方とも 0 からリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS (更新リクエスト選択) ビットがセットされている場合、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (割り込みや DMA リクエストは送信されません)。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットの設定に応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 249. 内部クロック分周比が 1 の場合のカウンタのタイミング図

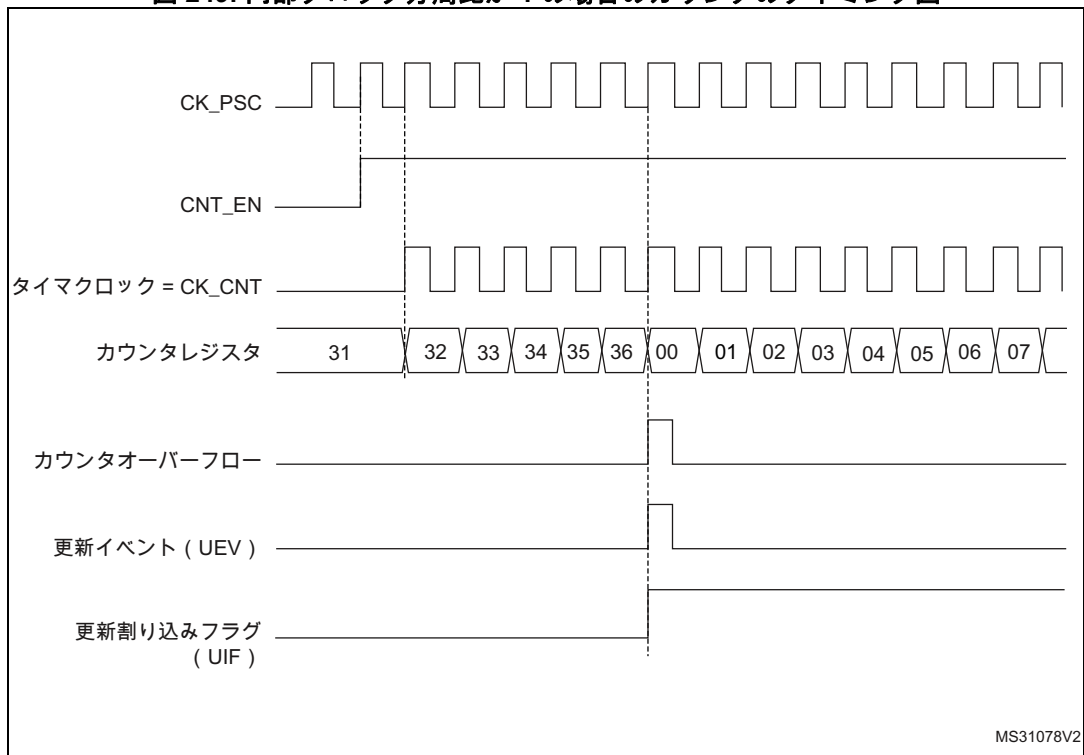


図 250. 内部クロック分周比が 2 の場合のカウンタのタイミング図

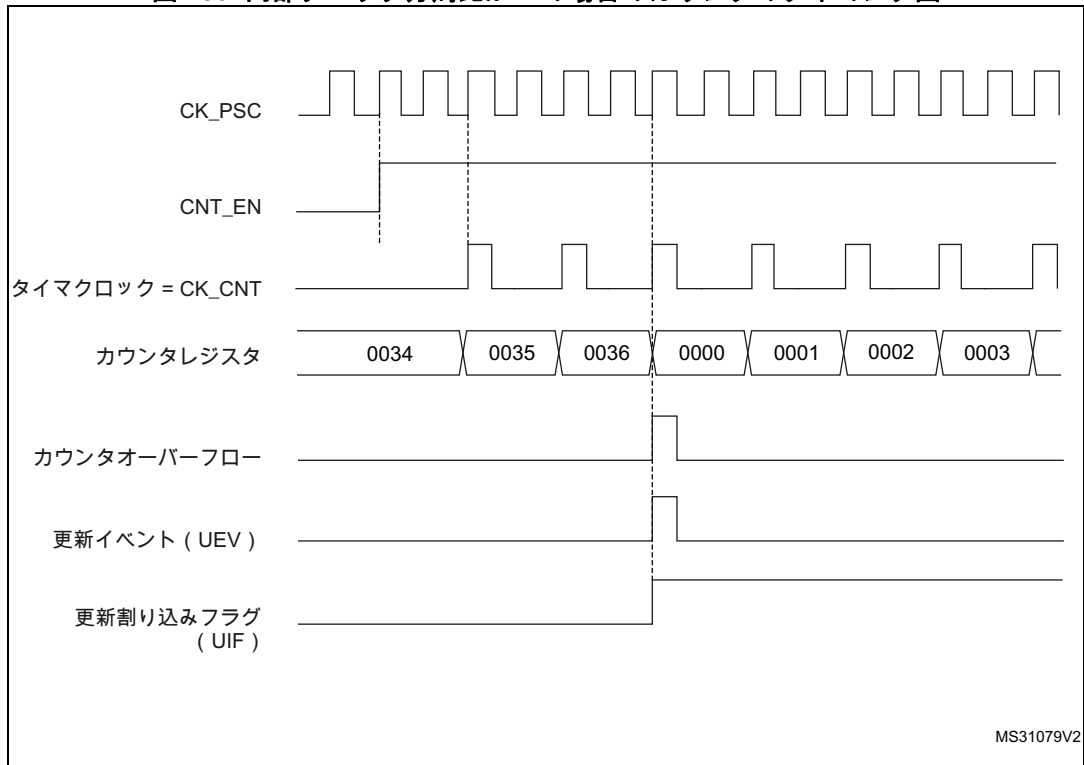


図 251. 内部クロック分周比が 4 の場合のカウンタのタイミング図

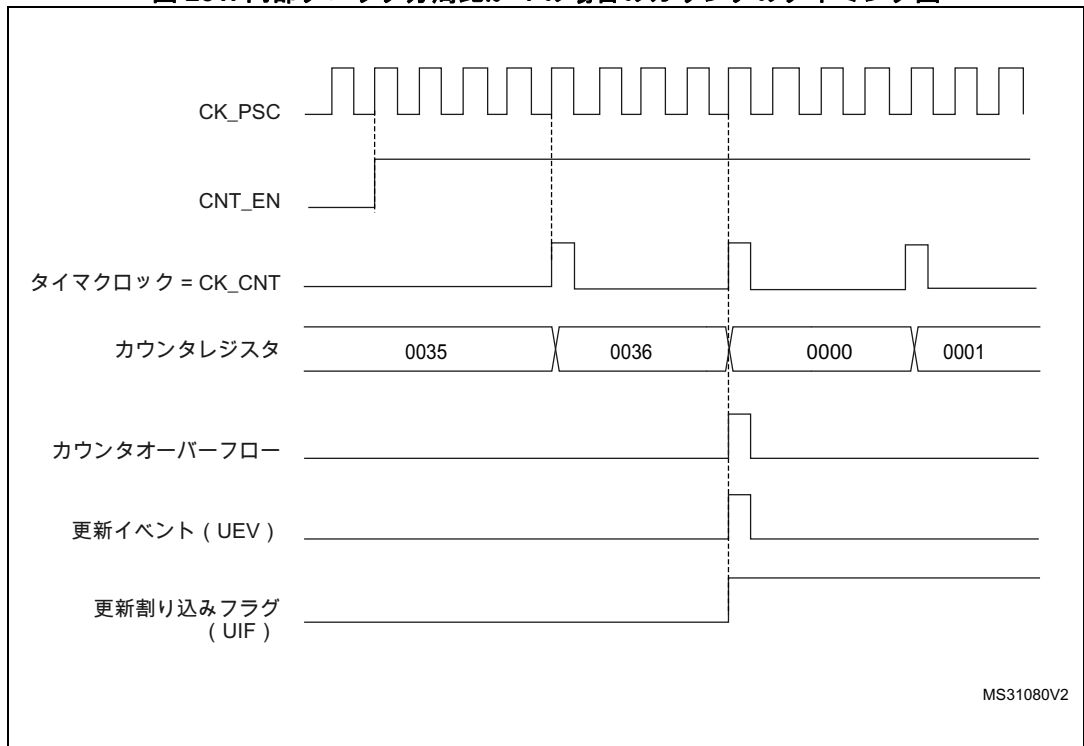


図 252. 内部クロック分周比が N の場合のカウンタのタイミング図

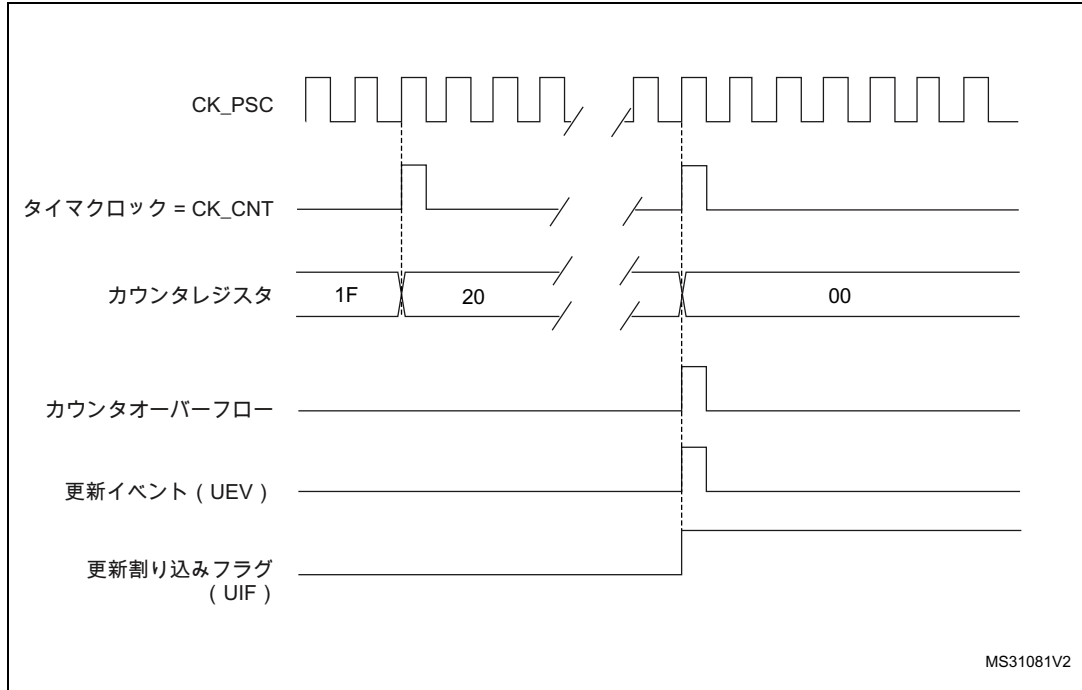


図 253. ARPE=0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

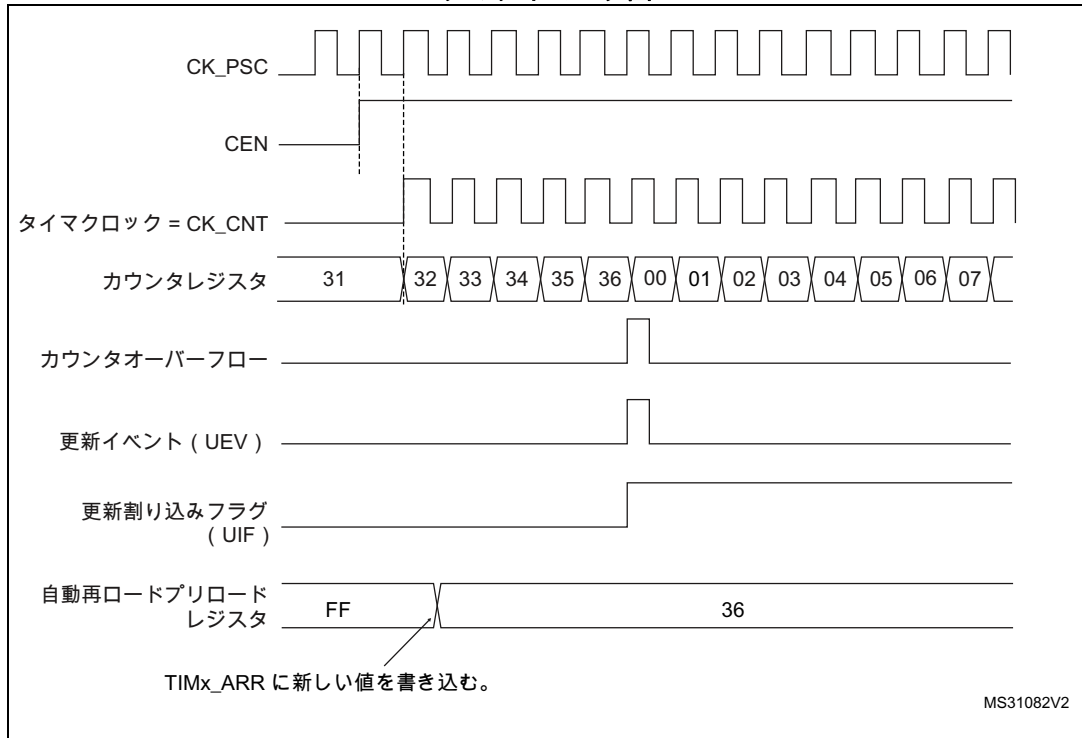
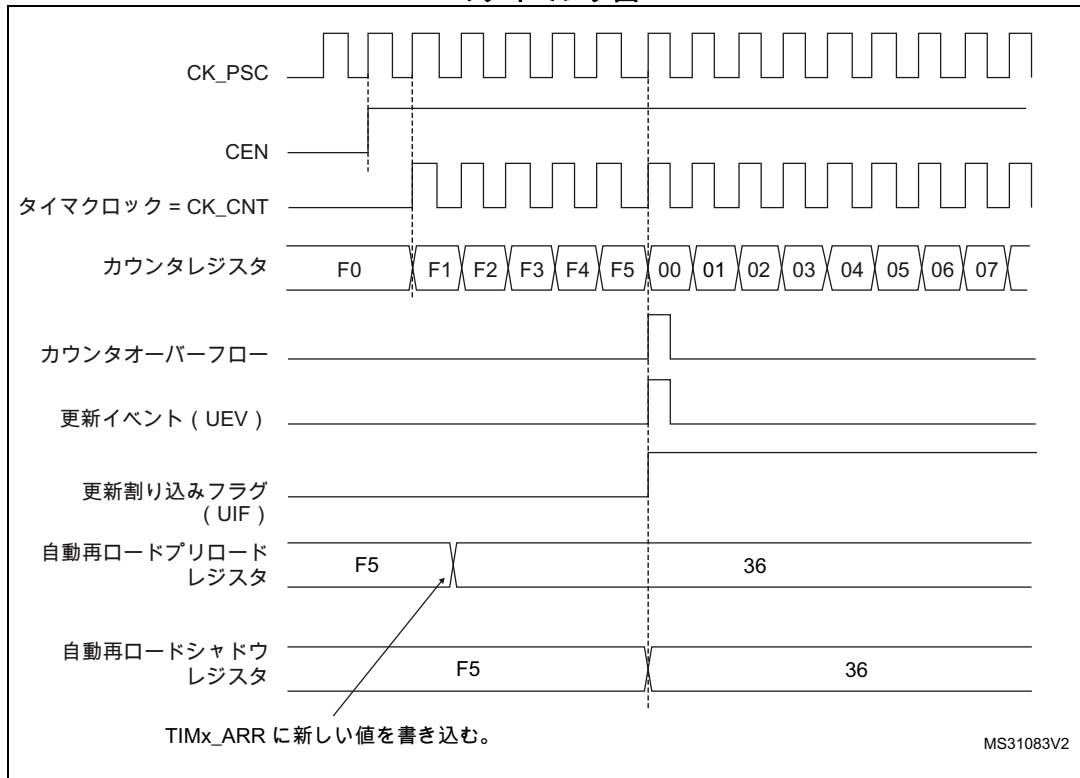


図 254. ARPE = 1 (TIMx\_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



### 22.3.3 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができます。特定のケースでは、バックグラウンドタスク (カウンタの読み出し) と割り込み (更新の割り込み) との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

UIF と UIFCPY フラグのアサートの間には、遅延はありません。

### 22.3.4 クロックソース

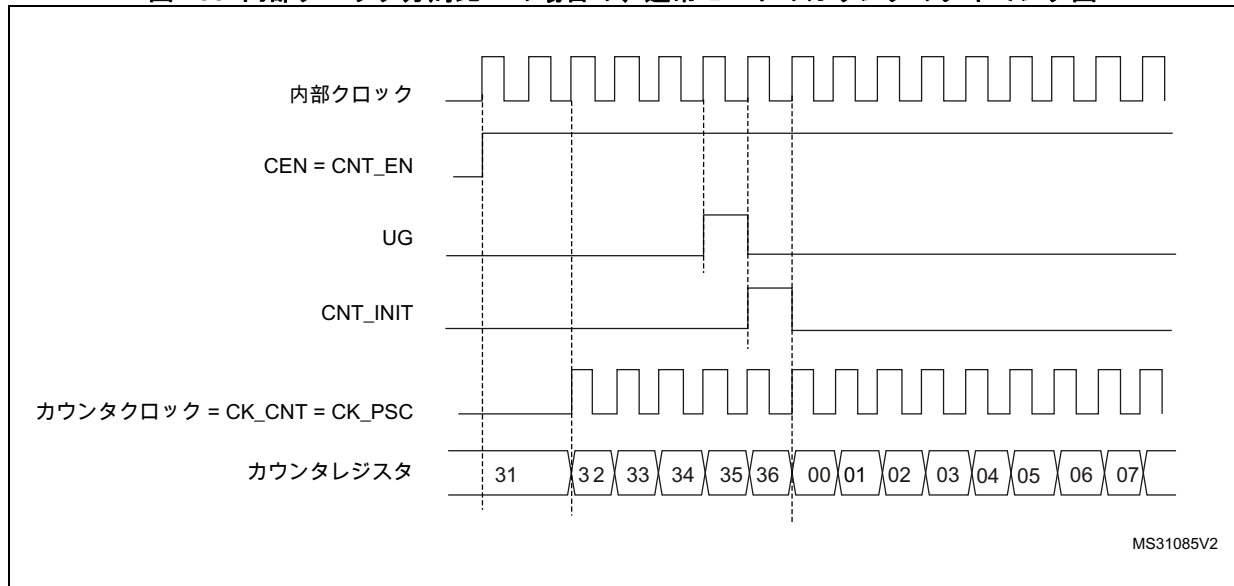
カウンタクロックは、内部クロック (CK\_INT) ソースから供給されます。

TIMx\_CR1 レジスタの CEN ビットと TIMx\_EGR レジスタの UG ビットは実際の制御ビットであり、ソフトウェアによってのみ変更できます (ただし、自動的にクリア状態が保持される UG ビットを除く)。CEN ビットに 1 が書き込まれると、プリスケラにはクロックとして内部クロック CK\_INT が供給されます。

図 255 に、プリスケラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。



図 255. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



MS31085V2

## 22.3.5 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M4<sup>®</sup>F コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 33.16.2: タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 22.4 TIM6/TIM7 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#)を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 22.4.1 TIM6/TIM7 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF RE-MAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw				rw				rw	rw	rw	rw

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10:8 予約済みであり、リセット値に保持する必要があります。



ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC) はそれぞれの値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

**注 :** ゲートモードは、CEN ビットが事前にソフトウェアでセットされている場合にのみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

## 22.4.2 TIM6/TIM7 制御レジスタ 2 (TIMx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **MMS** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (TRGO) を選択するために使用します。組み合わせは、次のとおりです。

000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが発生したとき (スレーブモードコントローラがリセットモードに設定されているとき) TRGO 信号は実際のリセットから遅れて発生します。

001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除いて、TRGO に遅延が存在します (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。

**注 :** *スレーブタイマまたは ADC のクロックは、マスタタイマからイベントを受信する前に有効化する必要があります。マスタタイマからトリガを受信している間は動作中に変更しないでください。*

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

## 22.4.3 TIM6/TIM7 DMA/割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

- 0 : 更新 DMA リクエストは無効です。
- 1 : 更新 DMA リクエストは有効です。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIE** : 更新割り込みイネーブル

- 0 : 更新割り込みは無効です。
- 1 : 更新割り込みは有効です。



## 22.4.4 TIM6/TIM7 のステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- TIMx\_CR1 レジスタの UDIS = 0 であり、繰り返しカウンタ値でオーバーフローまたはアンダーフローが発生したとき。

- TIMx\_CR1 レジスタの URS = 0 かつ UDIS = 0 の場合に、TIMx\_EGR レジスタの UG ビットを使用して、ソフトウェアで CNT が再初期化されたとき。

## 22.4.5 TIM6/TIM7 のイベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : タイマカウンタを再初期化して、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

## 22.4.6 TIM6/TIM7 のカウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

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

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー。TIMx\_CR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

## 22.4.7 TIM6/TIM7 プリスケーラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

PSC[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケーラ値

カウンタクロック周波数 CK\_CNT は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブなプリスケーラレジスタにロードされる値を含みます

(更新イベントには、TIMx\_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

## 22.4.8 TIM6/TIM7 の自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0xFFFF

ARR[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **ARR[15:0]** : プリスケーラ値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 22.3.1: タイムベースユニット \(667 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 22.4.9 TIM6/TIM7 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 127. TIM6/TIM7 レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JIFREMAP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OPM	URS	UDIS	CEN	
	リセット値																						0				0				0	0	0	0		
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS [2:0]	Res.	Res.	Res.	Res.	Res.			
	リセット値																											0	0	0						
0x08	予約済み																																			
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																																		0	UIE
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			UIF
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			UG
0x18-0x20	予約済み																																			
0x24	TIMx_CNT	UJFCPY または Res.															CNT[15:0]																			
	リセット値	0																																		
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

## 23 汎用タイマ (TIM15/TIM16/TIM17)

### 23.1 TIM15/TIM16/TIM17 の概要

TIM15/TIM16/TIM17 タイマは、プログラム可能なプリスケラによって駆動される 16 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM、デッドタイムを挿入した相補 PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

TIM15/TIM16/TIM17 タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 23.4.20 : タイマ同期 \(TIM15\)](#) に示すように、相互に同期させることができます。

### 23.2 TIM15 の主な特長

TIM15 には以下の機能があります。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケラ (動作中も変更可能) で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- 次の機能を持つ、最大 2 つの独立チャンネル。
  - 入力キャプチャ
  - 出力比較
  - PWM 生成 (エッジモード)
  - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力 (チャンネル 1 のみ)
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブ레이크入力。
- 以下のイベント時の割り込み/DMA 生成。
  - 更新: カウンタオーバーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
  - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
  - 入力キャプチャ
  - 出力比較
  - ブ레이크入力 (割り込みリクエスト)

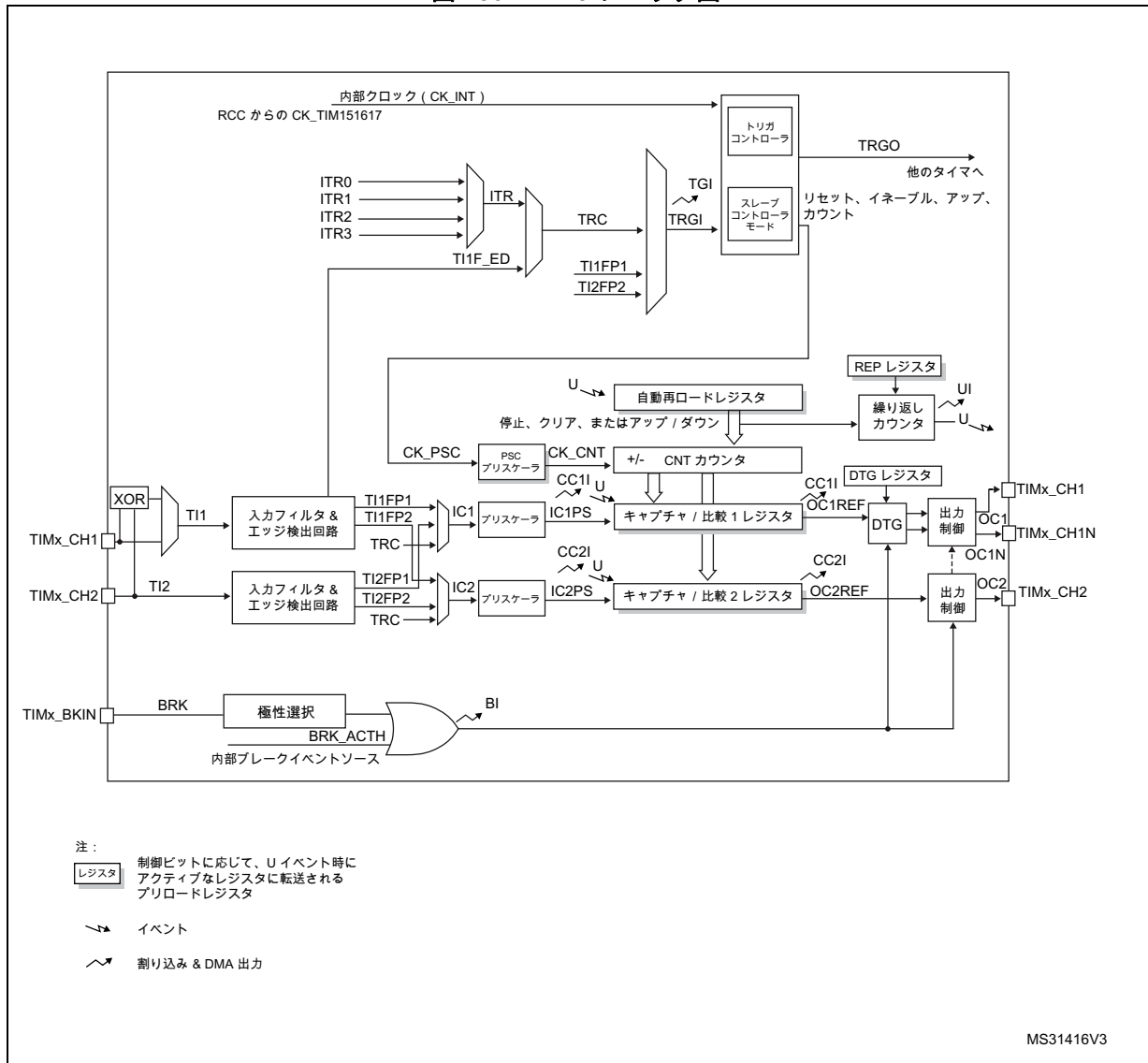
## 23.3 TIM16/TIM17 の主な特長

TIM16/TIM17 タイマには以下の機能があります。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ（動作中も変更可能）で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- 次の機能を持つ、1 チャンネルタイマ。
  - 入力キャプチャ
  - 出力比較
  - PWM 生成（エッジアラインモード）
  - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレーク入力。
- 以下のイベント時の割り込み/DMA 生成。
  - 更新：カウンタオーバーフロー
  - トリガイベント（カウンタの開始、停止、初期化、または内部/外部トリガによるカウント）
  - 入力キャプチャ
  - 出力比較
  - ブレーク入力

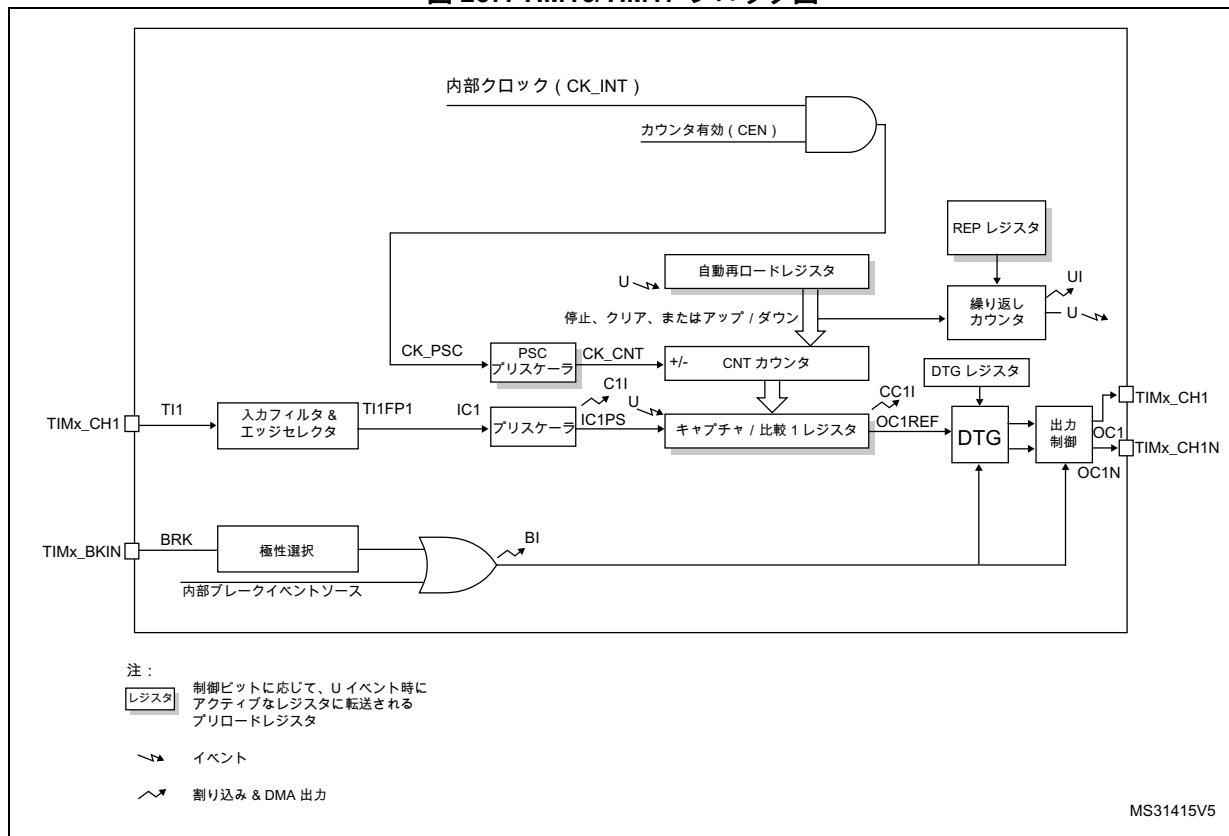


図 256. TIM15 ブロック図



- 内部ブレークイベントソースは次のいずれかです。
  - CSS によって生成されたクロック障害イベント。CSS の詳細については、[セクション 9.2.7: クロックセキュリティシステム \(CSS\)](#) を参照してください。
  - PVD 出力
  - SRAM パリティエラー信号
  - Cortex-M4<sup>®</sup>F LOCKUP (ハードフォルト) 出力
  - COMP 出力

図 257. TIM16/TIM17 ブロック図



- 内部ブレークイベントソースは次のいずれかです。
  - CSS によって生成されたクロック障害イベント。CSS の詳細については、[セクション 9.2.7: クロックセキュリティシステム \(CSS\)](#) を参照してください。
  - PVD 出力
  - SRAM パリティエラー信号
  - Cortex-M4<sup>®</sup>F LOCKUP (ハードフォルト) 出力
  - COMP 出力

## 23.4 TIM15/TIM16/TIM17機能詳細

### 23.4.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx\_CNT)
- プリスケアラレジスタ (TIMx\_PSC)
- 自動再ロードレジスタ (TIMx\_ARR)
- 繰り返しカウンタレジスタ (TIMx\_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx\_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx\_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx\_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK\_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx\_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

#### プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx\_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 258](#) と [図 259](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 258. プリスケータ分周比が 1 から 2 に変化したときのカウンタのタイミング図

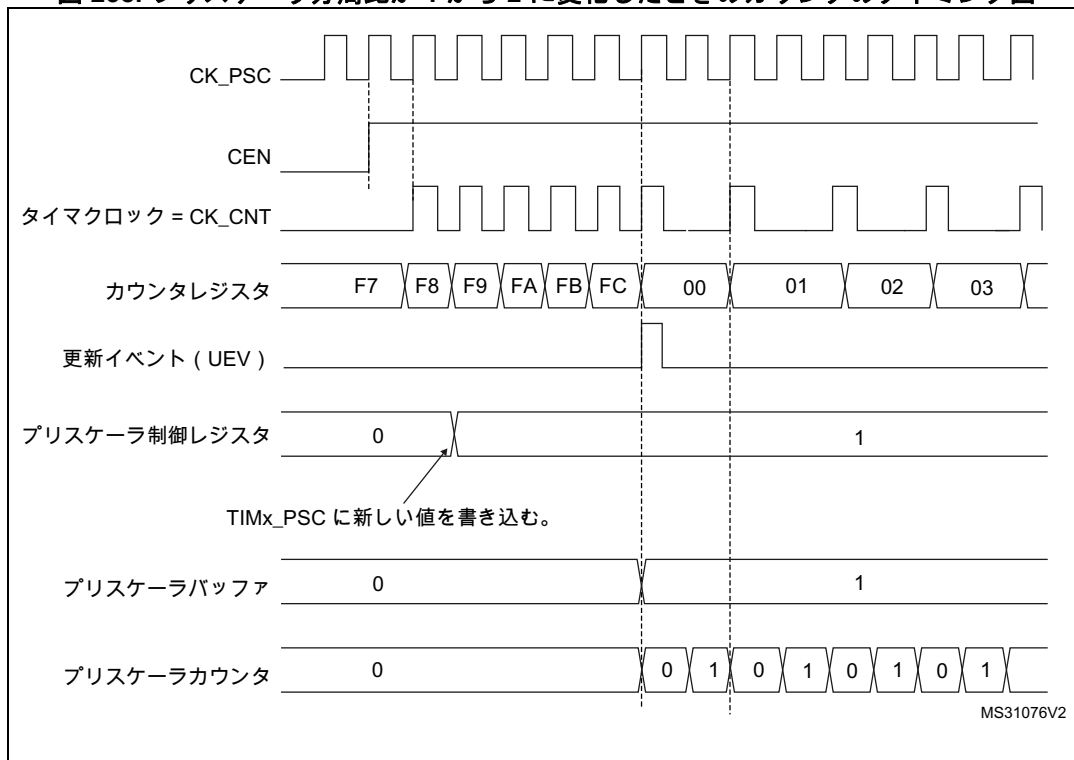
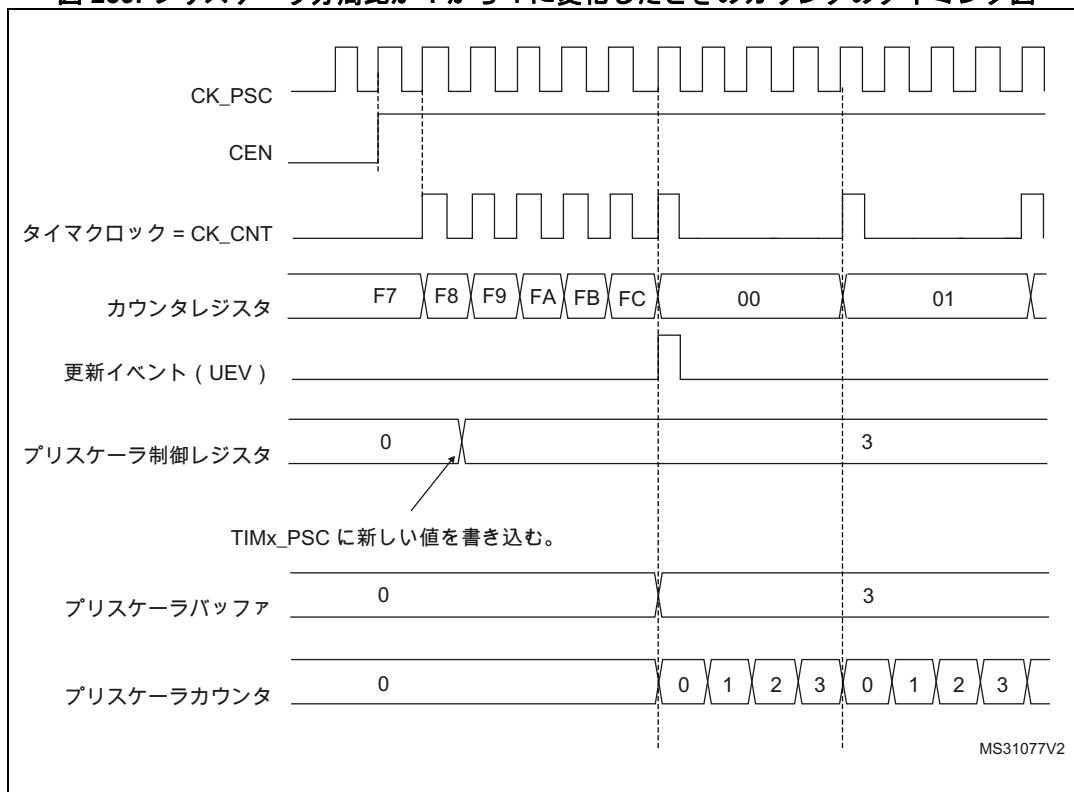


図 259. プリスケータ分周比が 1 から 4 に変化したときのカウンタのタイミング図



## 23.4.2 カウンタモード

### アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx\_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx\_RCR) までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx\_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx\_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx\_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx\_PSC レジスタの内容) が再びロードされません。

以下の図は、TIMx\_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 260. 内部クロック分周比が 1 の場合のカウンタのタイミング図

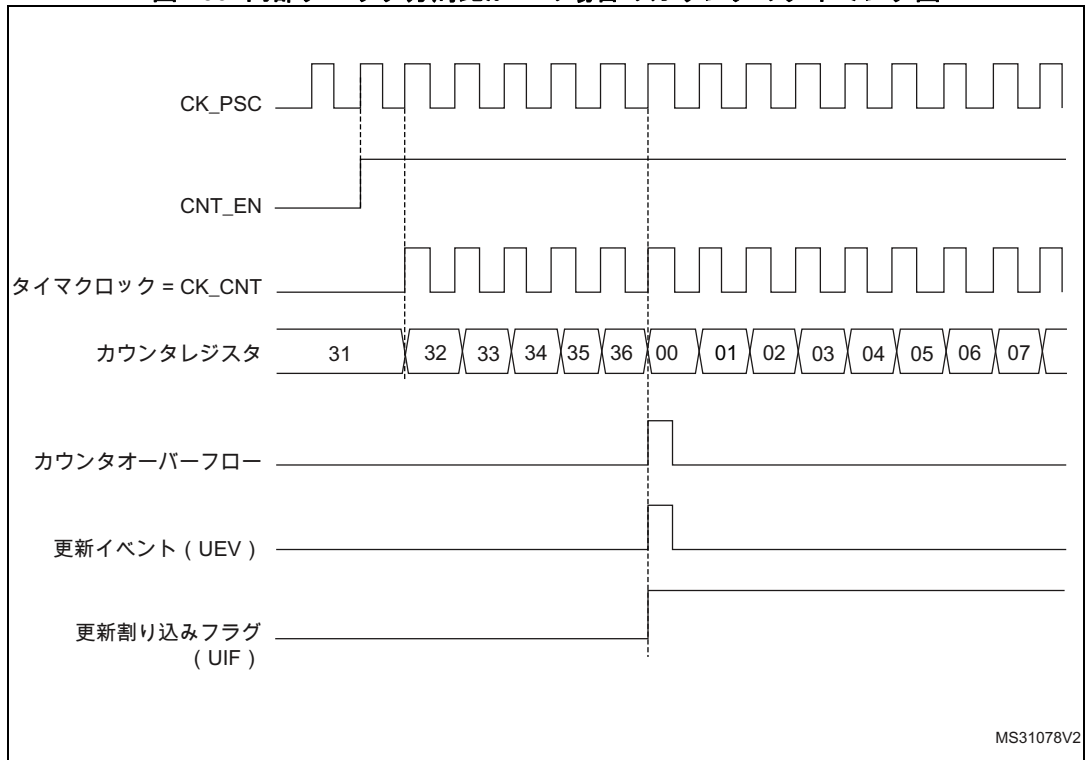


図 261. 内部クロック分周比が 2 の場合のカウンタのタイミング図

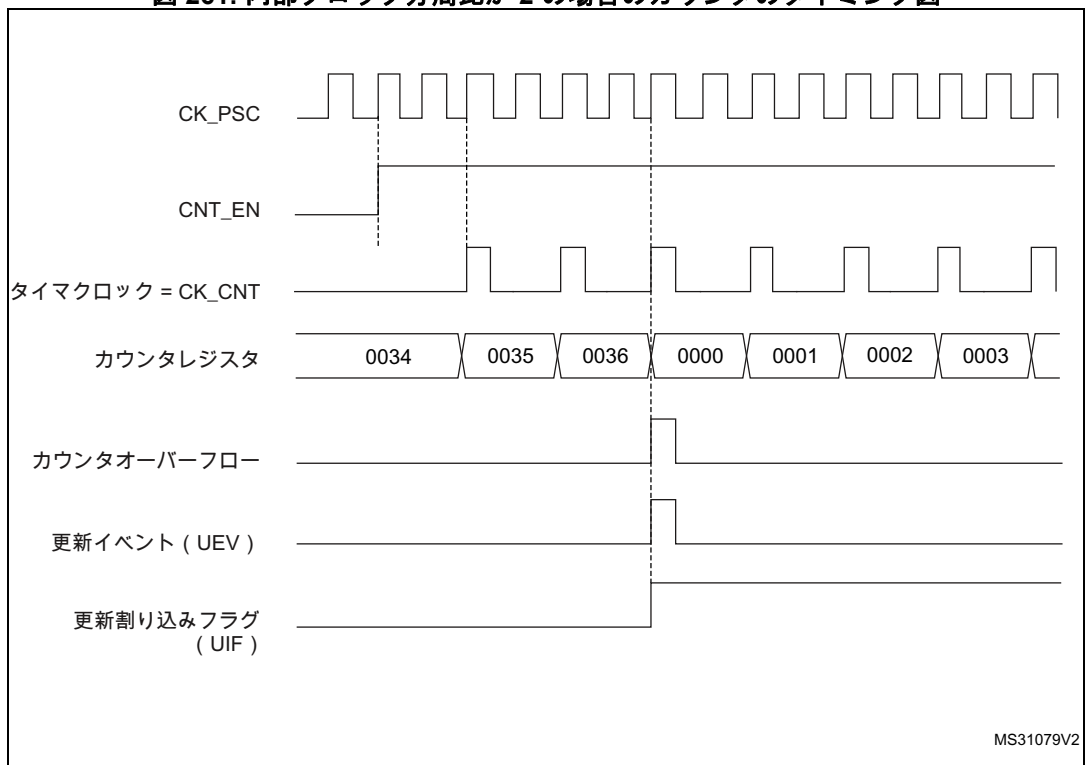


図 262. 内部クロック分周比が 4 の場合のカウンタのタイミング図

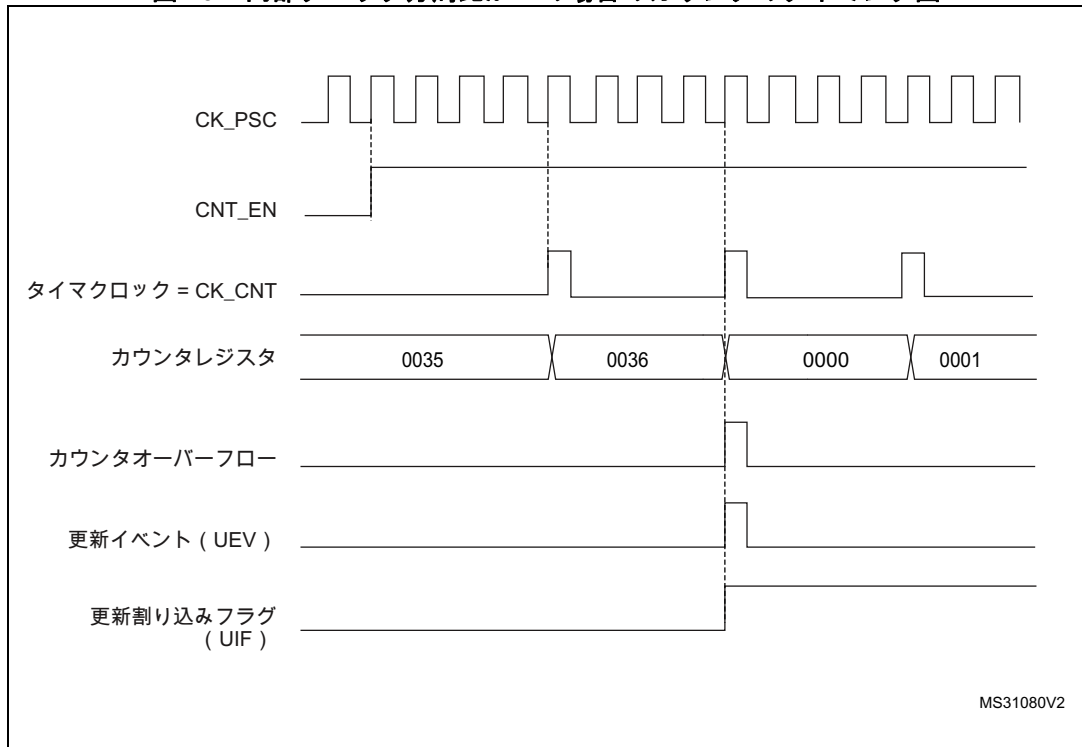


図 263. 内部クロック分周比が N の場合のカウンタのタイミング図

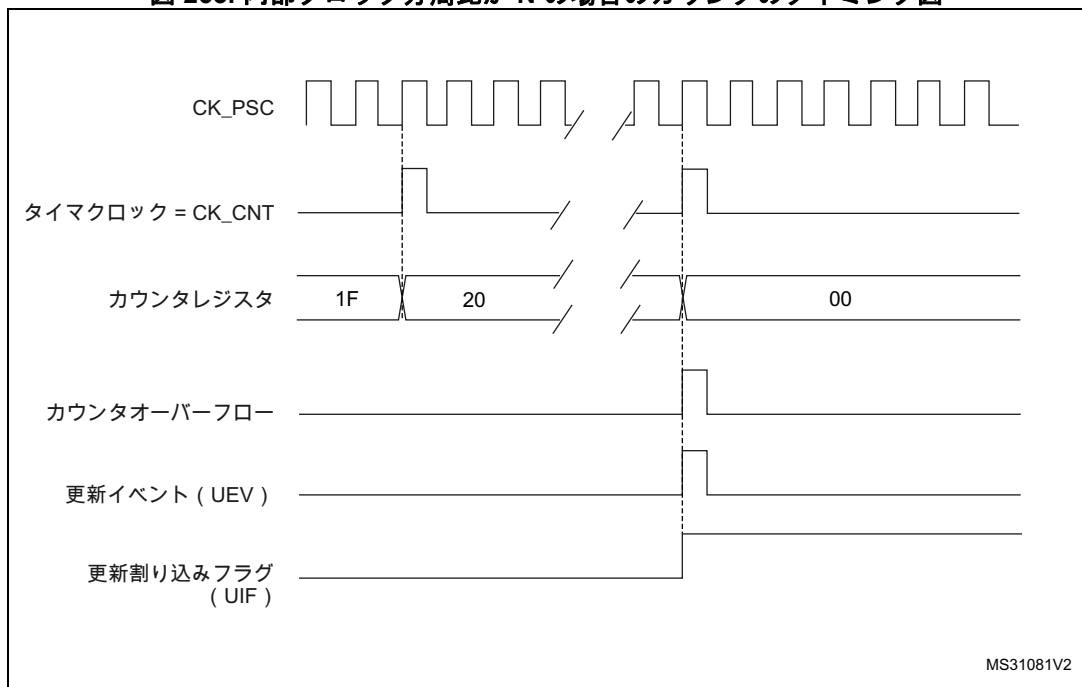


図 264. ARPE=0 (TIMx\_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

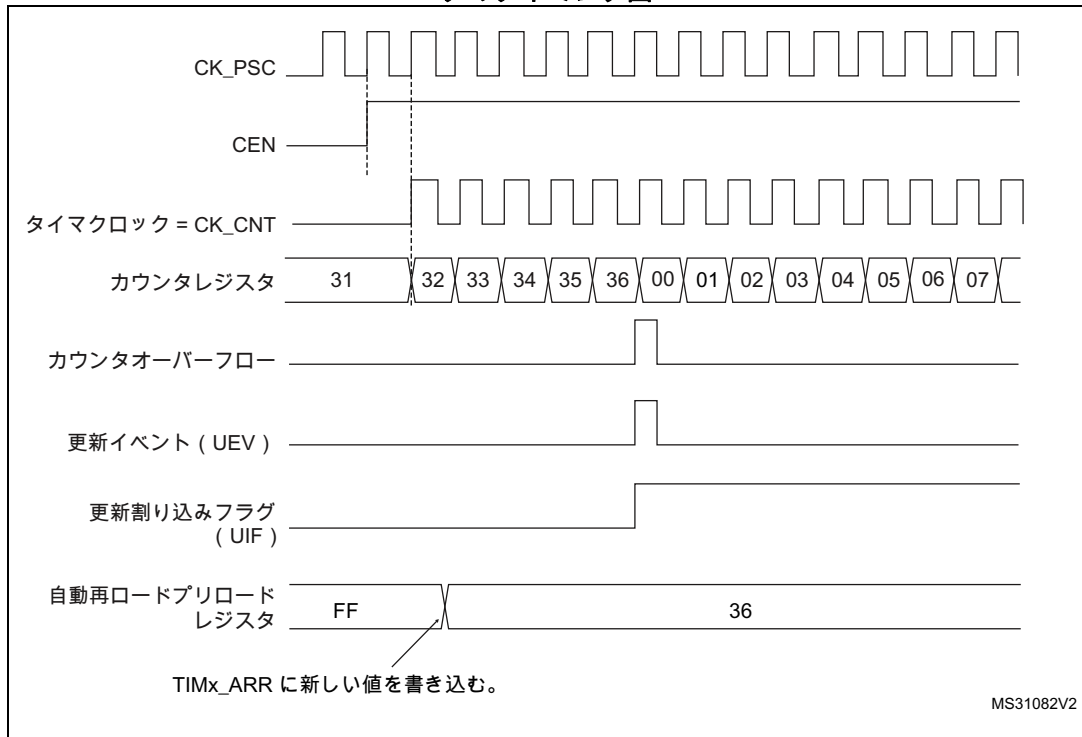
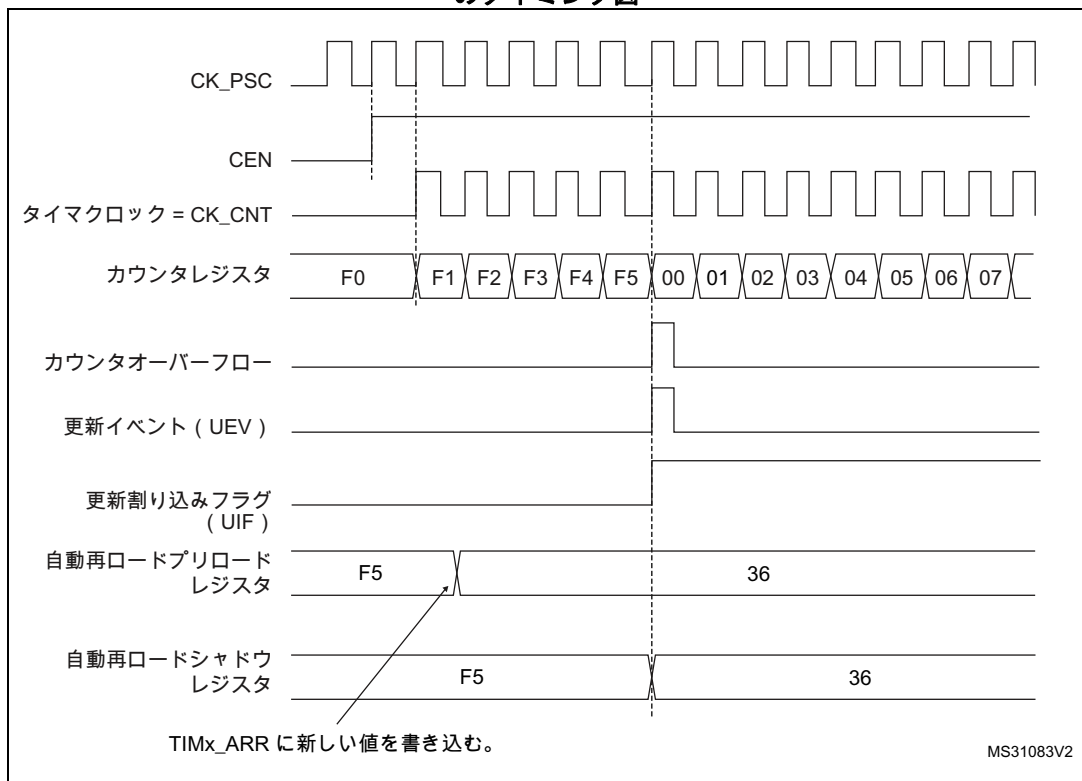


図 265. ARPE = 1 (TIMx\_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図





## 23.4.3 繰り返しカウンタ

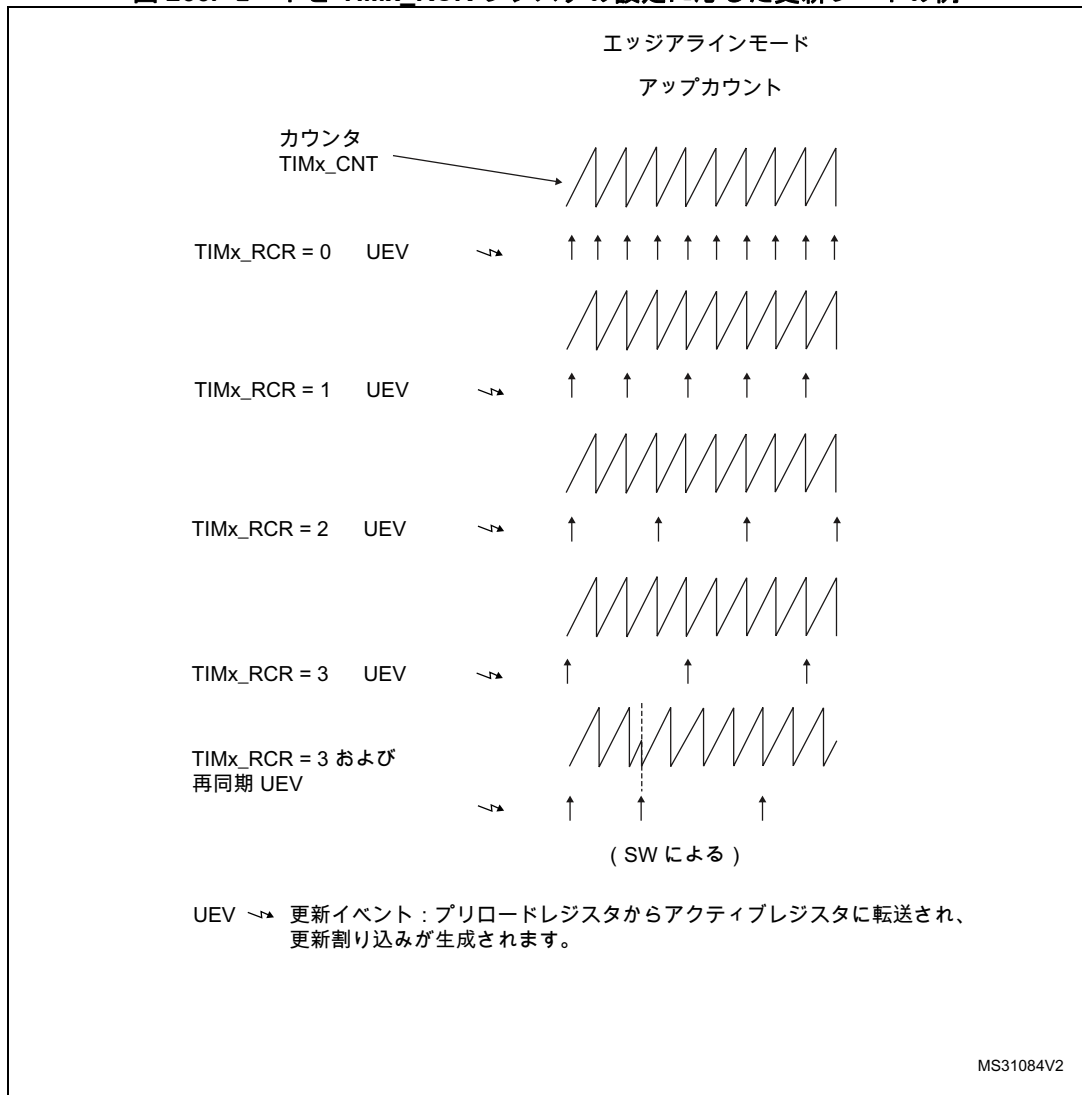
セクション 23.4.1: タイムベースユニットに、カウンタオーバーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰り返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx\_RCR 繰り返しカウンタレジスタの値を N とすると、N 回目のカウンタオーバーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx\_ARR 自動再ロードレジスタ、TIMx\_PSC プリスケアラレジスタ、比較モードの TIMx\_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

繰り返しカウンタは、カウンタオーバーフローごとにデクリメントされます。

繰り返しダウンカウンタは自動再ロードタイプです。繰り返しの回数は、TIMx\_RCR レジスタの値によって定義されたとおりに維持されます (図 266 を参照してください)。ソフトウェアによって (TIMx\_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰り返しカウンタの値にかかわらず直ちにイベントが発生し、繰り返しカウンタに TIMx\_RCR レジスタの内容が再ロードされます。

図 266. モードと TIMx\_RCR レジスタの設定に応じた更新レートの例



## 23.4.4 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

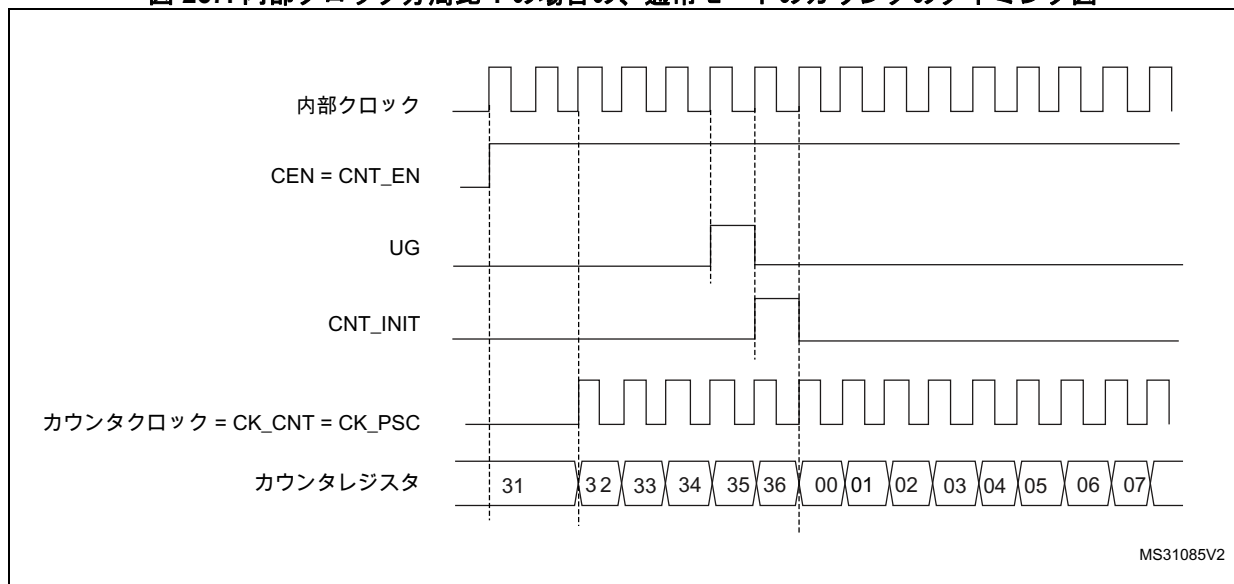
- 内部クロック (CK\_INT)
- 外部クロックモード 1: 外部入力ピン
- 内部トリガ入力 (ITRx) (TIM15 の場合のみ) : あるタイマを別のタイマのプリスケアラとして使用します。たとえば、TIM1 が TIM15 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する \(636 ページ\)](#) を参照してください。

### 内部クロックソース (CK\_INT)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN (TIMx\_CR1 レジスタ)、および UG ビット (TIMx\_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK\_INT が供給されます。

[図 267](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

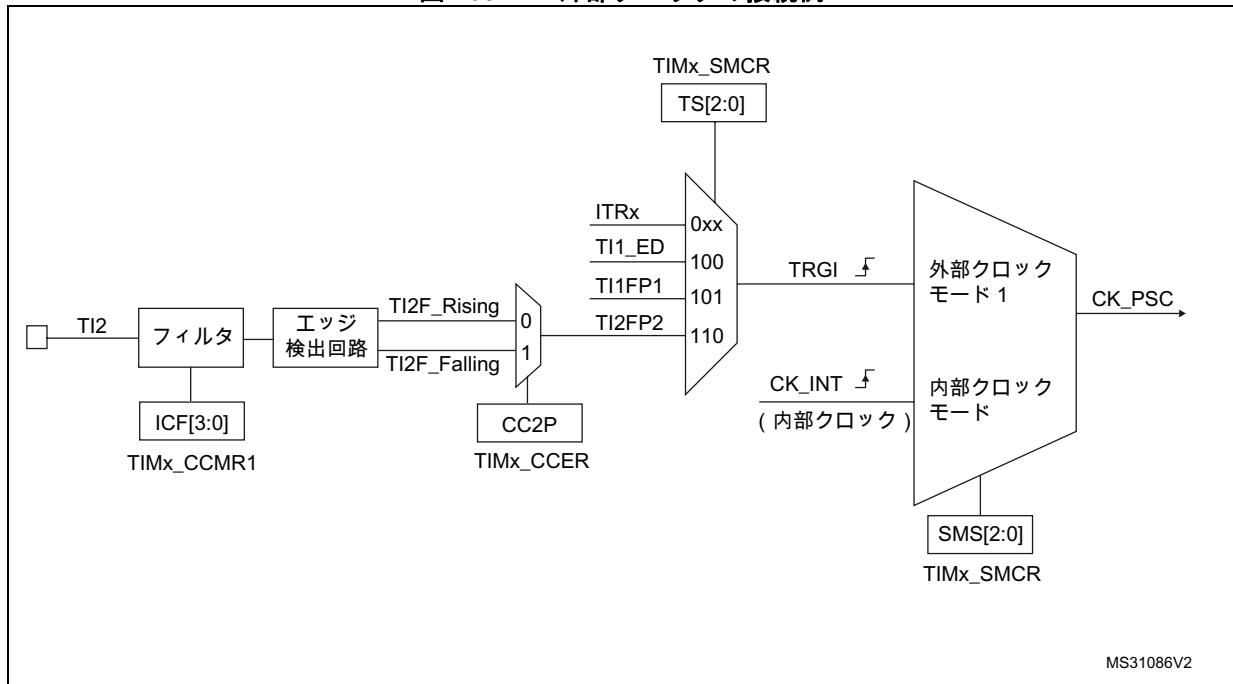
図 267. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



### 外部クロックソースモード 1

このモードは TIMx\_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 268. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

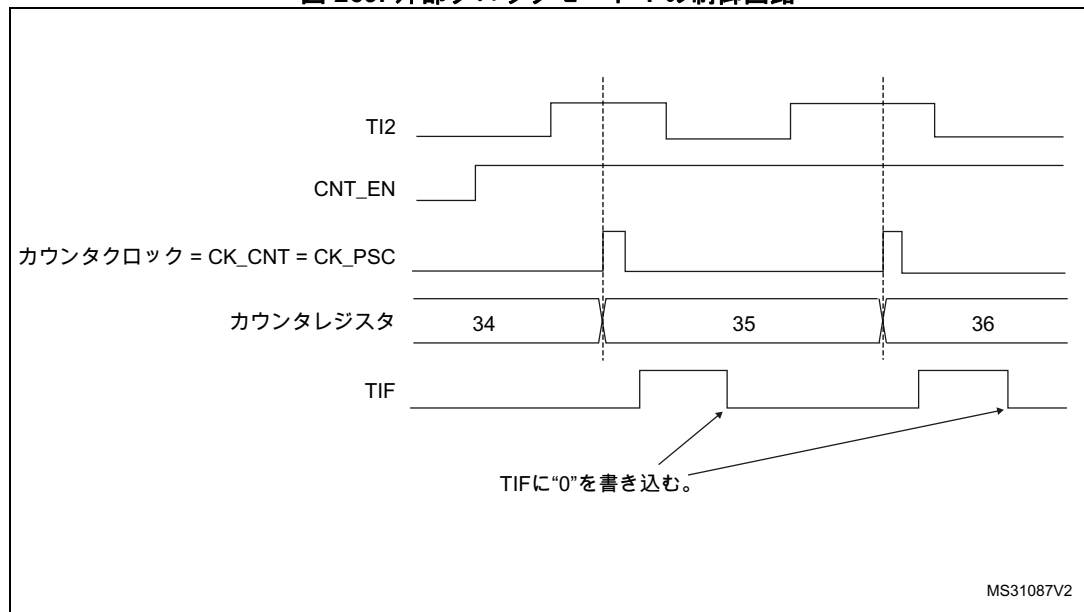
1. TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx\_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。
3. CC2P=0 を TIMx\_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx\_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx\_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

**注：** キャプチャプリスケアラはトリガには使用されないため、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウントクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 269. 外部クロックモード 1 の制御回路



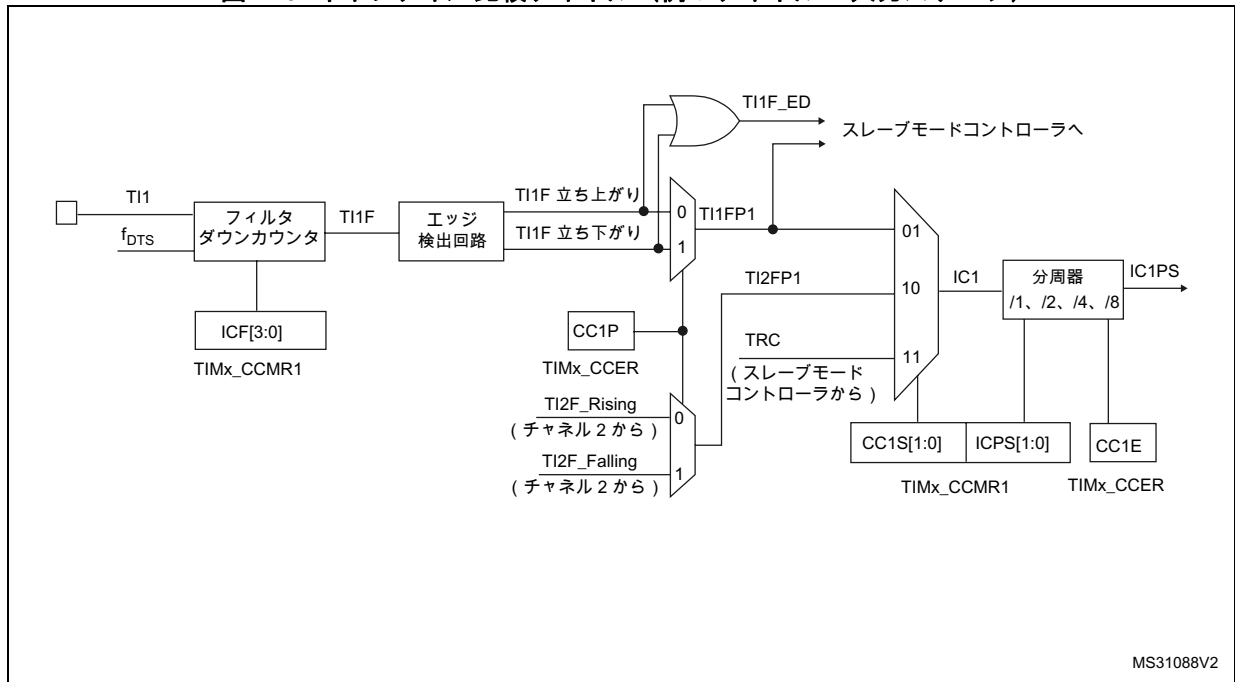
## 23.4.5 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケラ）、および出力カステージ（比較回路と出力制御）から構成されています。

図 270 から 図 273 に、1 つのキャプチャ／比較チャネルの概要を示します。

入カステージは、対応する T1x 入カをサンプリングして、フィルタリングを行った T1xF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入カとして、またはキャプチャコマンドとして使用される信号 (T1xFPx) を生成します。この信号はプリスケラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 270. キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)



出力ステージは、OCxRef (アクティブハイ) として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 271. キャプチャ/比較チャンネル 1 メイン回路

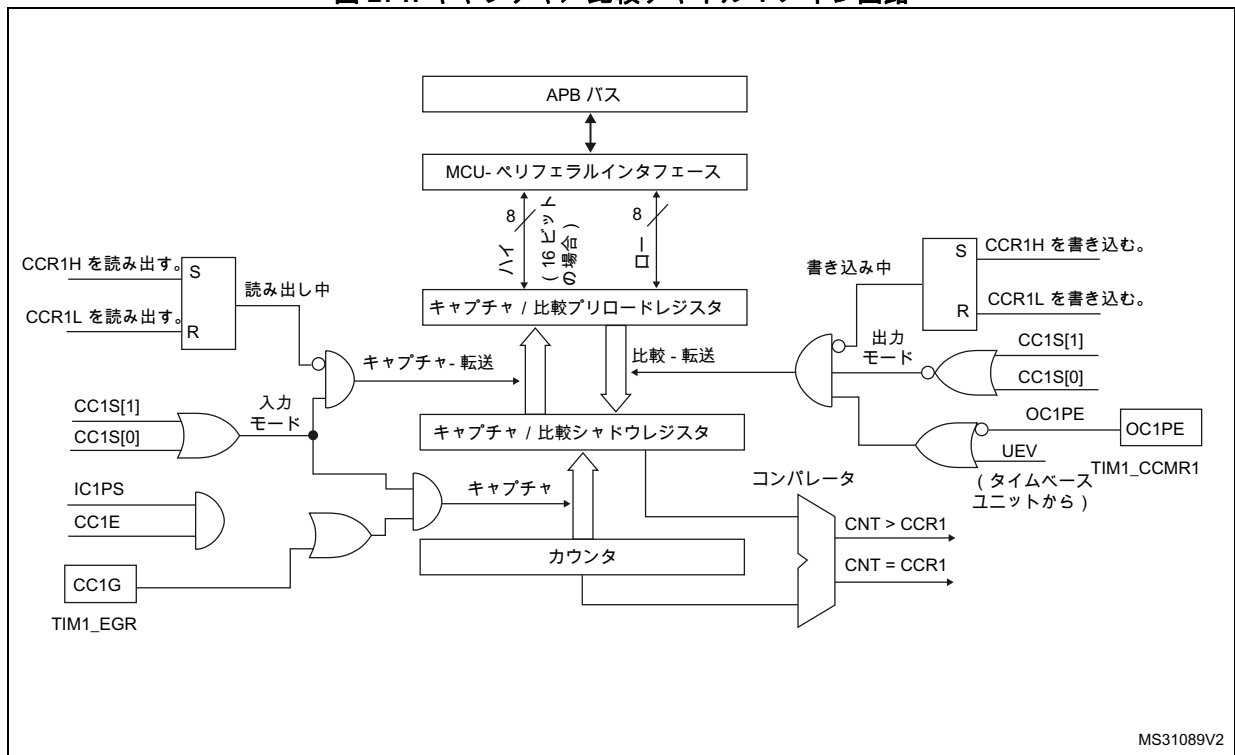
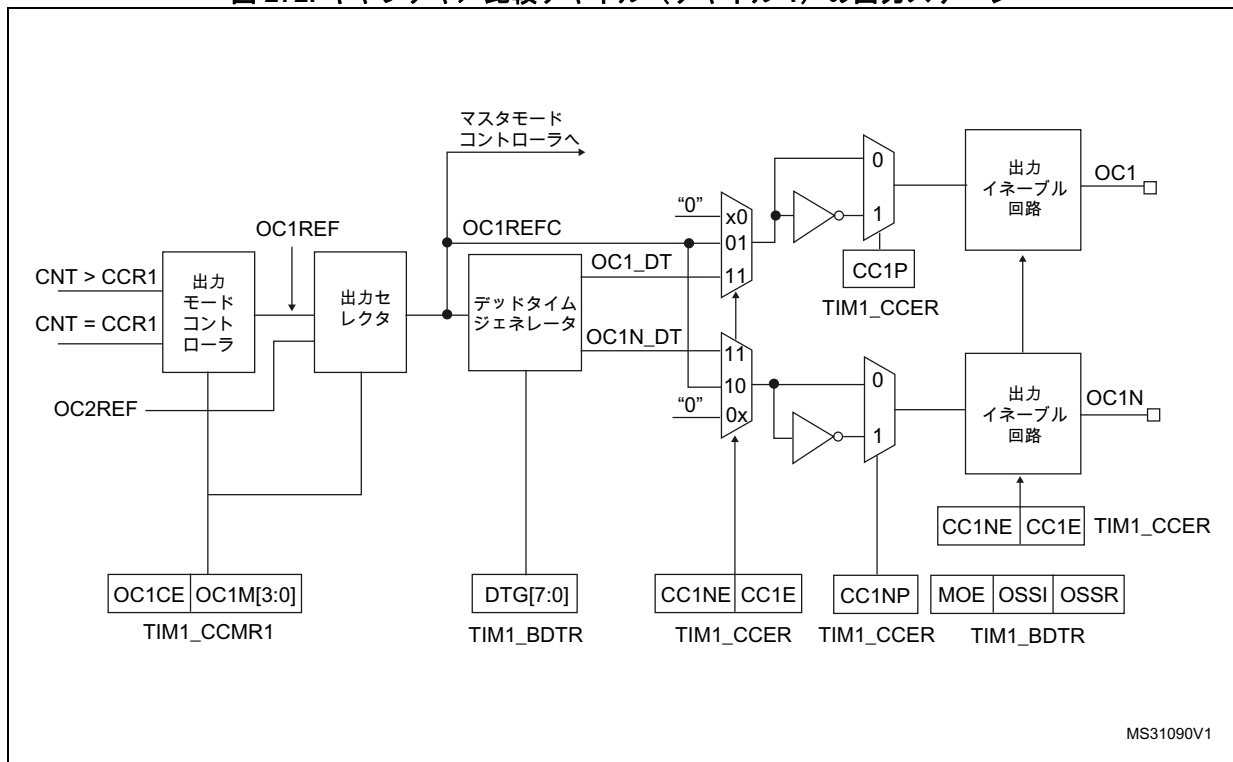
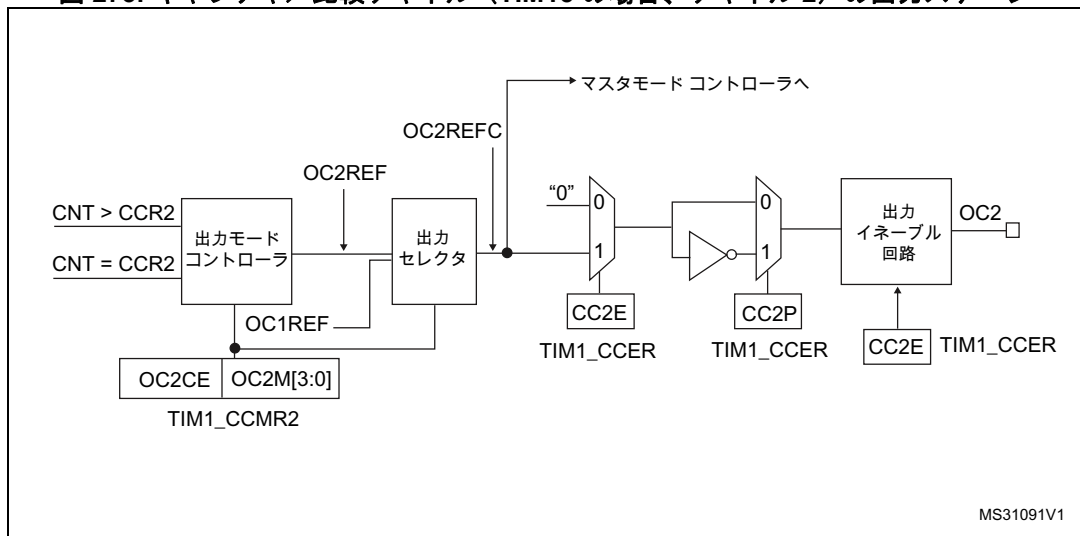


図 272. キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ



MS31090V1

図 273. キャプチャ/比較チャンネル (TIM15 の場合、チャンネル 2) の出力ステージ



MS31091V1

キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

## 23.4.6 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx\_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx\_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx\_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx\_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx\_CCR1 は TI1 入力とリンクされていなければならないため、このためには TIMx\_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx\_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最低でも内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数  $f_{DTS}$  でサンプリング)。この場合、TIMx\_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx\_CCER レジスタの CC1P ビットに 0 を書き込みます (この場合、立ち上がりエッジの選択)。
4. 入力プリスケラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいため、プリスケラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
5. TIMx\_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
6. 必要な場合は、TIMx\_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx\_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

**注:** IC 割り込みと DMA リクエストは、TIMx\_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

## 23.4.7 PWM 入力モード (TIM15 の場合のみ)

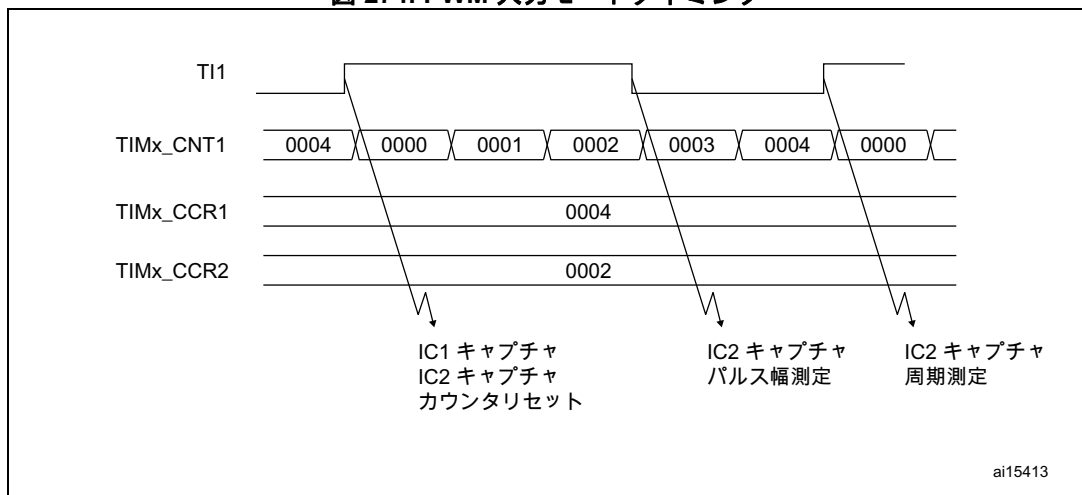
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます (手順は、CK\_INT 周波数とプリスケール値によって、若干異なることがあります)。

1. TIMx\_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx\_CCR1 のアクティブ入力を選択します。
2. CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx\_CCR1 のキャプチャとカウンタクリアの両方に使用)。
3. TIMx\_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
4. CC2P ビットと CC2NP ビットに“1”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx\_CCR2 のキャプチャに使用されます)。
5. TIMx\_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
6. TIMx\_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
7. TIMx\_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 274. PWM 入力モードタイミング



1. TI1FP1 と TI2FP2 のみがスレーブモードコントローラに接続されているので、PWM 入力モードは TIMx\_CH1/TIMx\_CH2 信号でのみ使用できます。



## 23.4.8 強制出力モード

出力モード (TIMx\_CCMRx レジスタの CCxS ビット =00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルとするには、対応する TIMx\_OCMRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx\_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx\_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

## 23.4.9 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx\_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx\_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx\_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx\_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx\_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx\_CR2 レジスタの CCDS ビットが使用されます)。

TIMx\_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx\_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

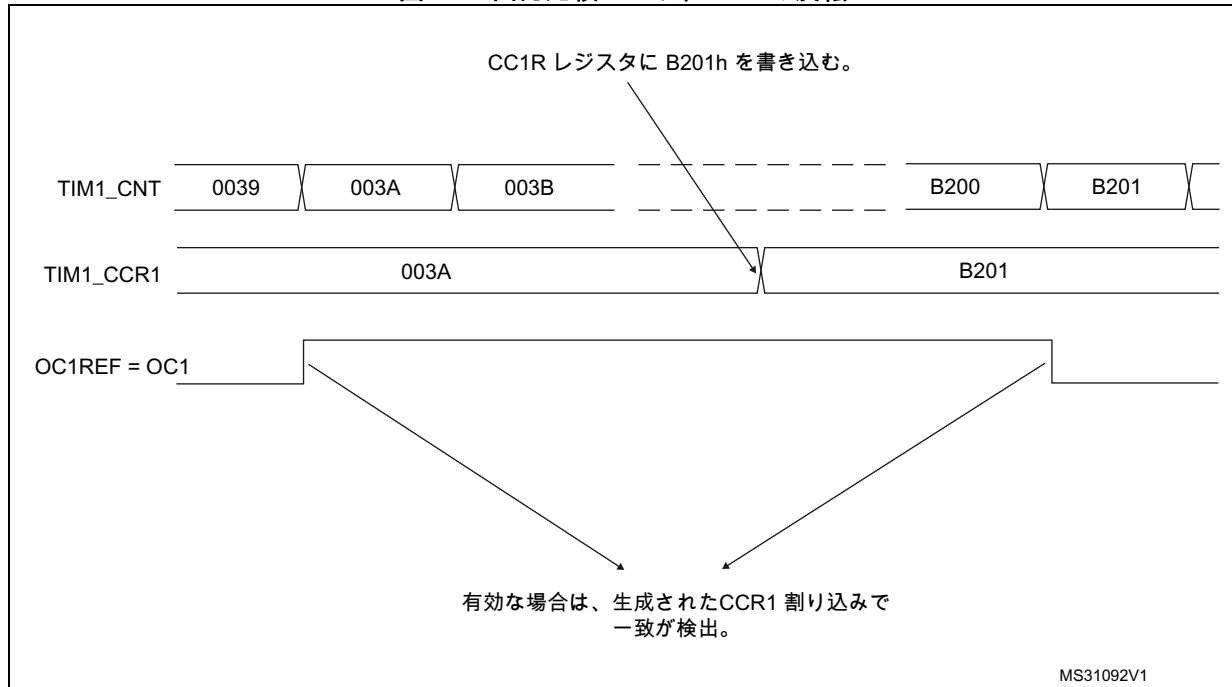
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルス出力するためにも使用できます (ワンパルスモード)。

### 手順

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx\_ARR レジスタと TIMx\_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例:
  - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに 011 を書き込みます。
  - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
  - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
  - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
5. TIMx\_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアによって TIMx\_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx\_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 274](#) に示します。

図 275. 出力比較モード、OC1 の反転



## 23.4.10 PWM モード

パルス幅変調 (PWM) モードでは、TIMx\_ARR レジスタの値によって決められた周波数と TIMx\_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx\_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx\_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモード) で有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx\_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

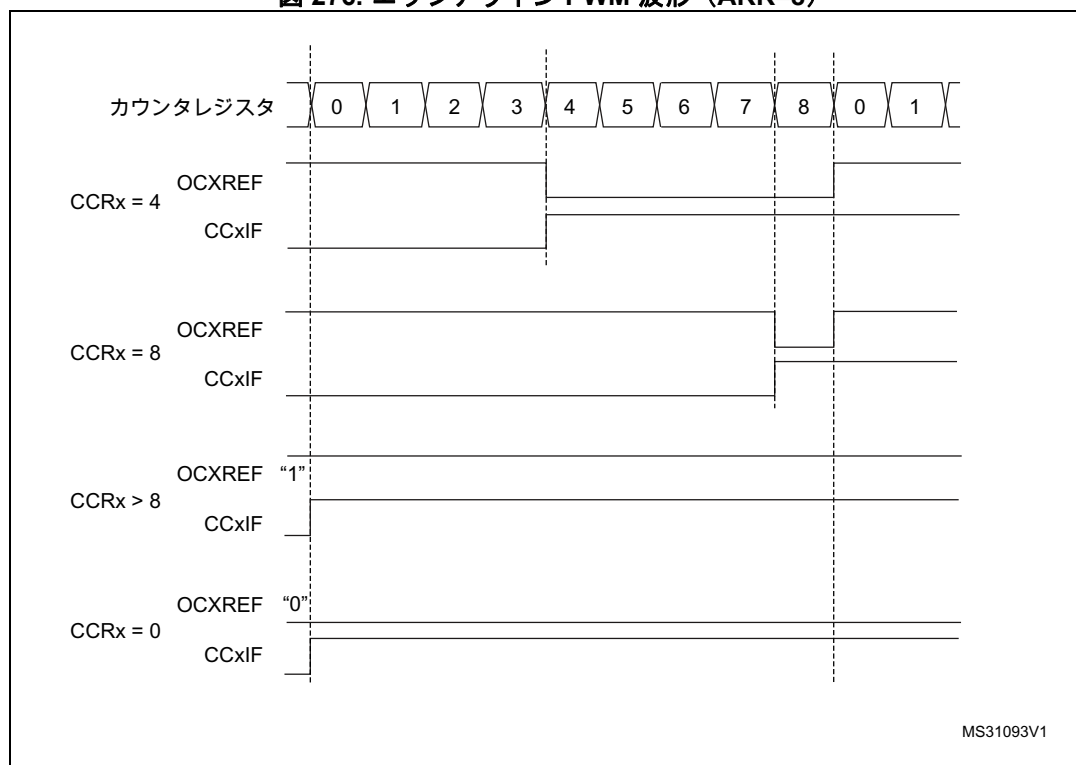
OCx 極性は、TIMx\_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx\_CCER および TIMx\_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx\_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx\_CNT と TIMx\_CCRx が常に比較されて、TIMx\_CCRx ≤ TIMx\_CNT または TIMx\_CNT ≤ TIMx\_CCRx がどうか判断されます (カウントの方向によります)。

TIM15/TIM16/TIM17 はアップカウント動作でのみ使用可能です。アップカウントモード (685 ページ) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx\_CNT < TIMx\_CCRx の間はハイに、そうでない場合はローになります。TIMx\_CCRx の比較値が自動再ロード値 (TIMx\_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。図 276 に TIMx\_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 276. エッジアライン PWM 波形 (ARR=8)



## 23.4.11 組み合わせ PWM モード (TIM15 のみ)

組み合わせ PWM モードでは、2つのエッジアラインまたはセンターアライン PWM 信号を生成でき、それぞれのパルス間に遅延および位相シフトをプログラムできます。周波数が TIMx\_ARR レジスタの値で決定されるのに対し、デューティサイクルや遅延は 2つの TIMx\_CCRx レジスタで決定されず。結果として得られる信号 OCxREFC は、2つの PWM 基準信号の OR または AND による論理結合から成ります。

- OC1REFC (または OC2REFC) は、TIMx\_CCR1 および TIMx\_CCR2 レジスタによって制御されます。

組み合わせ PWM モードは、TIMx\_CCMRx レジスタの OCxM ビットに“1100” (組み合わせ PWM モード 1) または“1101” (組み合わせ PWM モード 2) を書き込むことによって、2チャンネルごとに選択できます (CCR レジスタペアごとに 1つの OCx 出力)。

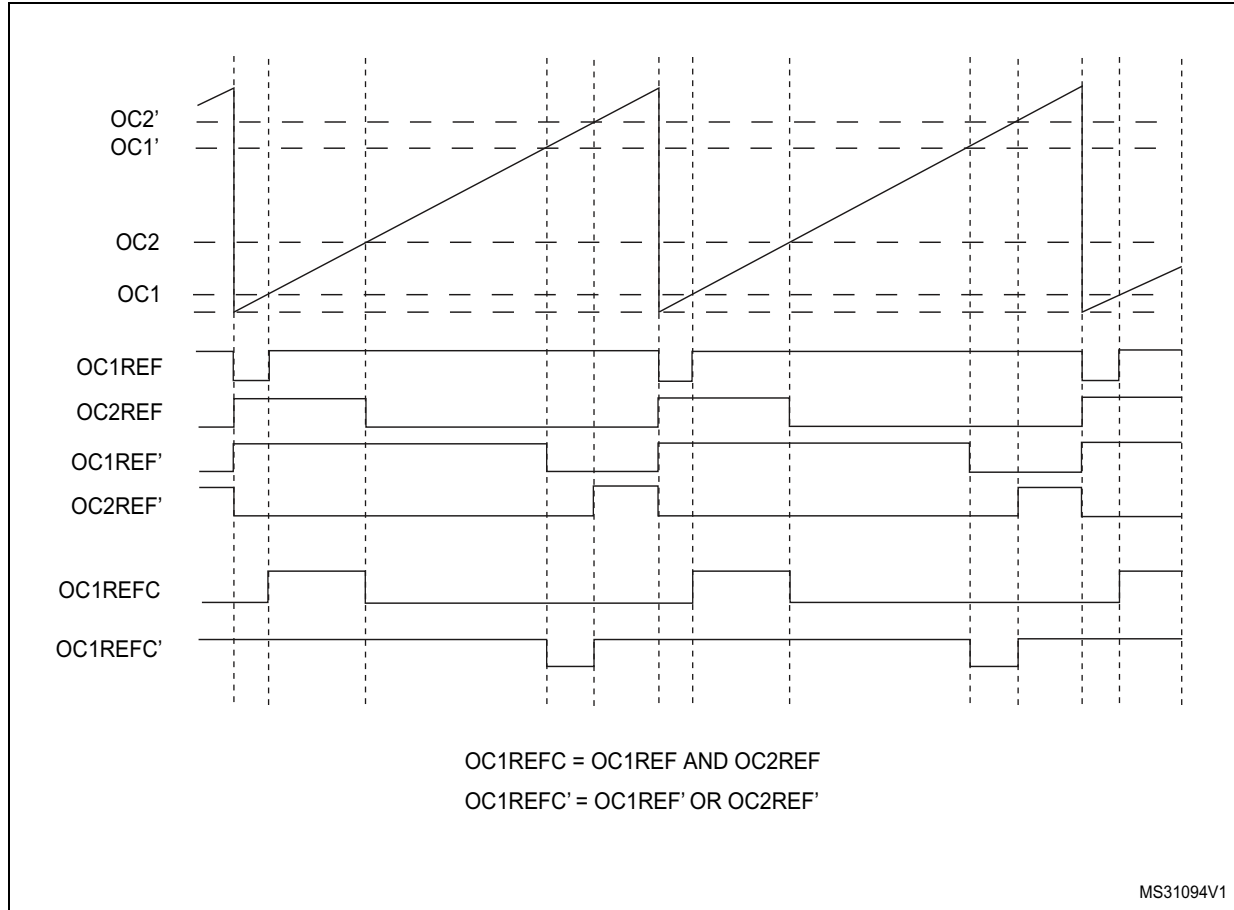
特定のチャンネルが組み合わせ PWM チャンネルとして使用されている場合、相補チャンネルを反対の PWM モードに設定する必要があります (たとえば、1つを組み合わせ PWM モード 1、もう 1つを組み合わせ PWM モード 2 にします)。

**注:** OCxM[3:0] ビットフィールドは互換性を確保するために 2つのパーツに分割され、最上位ビットと 3つの最下位ビットとは隣接していません。

図 277 は、次の設定で取得可能な非対称 PWM モードを使用して生成される信号の例を表します。

- チャンネル 1 が組み合わせ PWM モード 2 で設定されている場合
- チャンネル 2 が PWM モード 1 で設定されている場合

図 277. チャンネル 1 および 2 における組み合わせ PWM モード



## 23.4.12 相補出力とデッドタイム挿入

TIM15/TIM16/TIM17 汎用タイマは、1 つの相補信号を出力して、出力時のスイッチオフおよびスイッチオンを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性（レベルシフタの内在的な遅延、電源スイッチによる遅延など）に応じて調整する必要があります。

出力の極性（主出力 OCx または補 OCxN）は、出力ごとに独自に選択できます。これは TIMx\_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 OCx および OCxN は、TIMx\_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx\_BDTR レジスタと TIMx\_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、表 129: [ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット \(728 ページ\)](#) を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるときに)、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッ

ドタイムジェネレータがあります。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点が異なります。
- OCxN 出力信号は、立ち上がりエッジが基準波形の立ち下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (OCx または OCxN) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 OCxREF との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 278. デッドタイム挿入のある相補出力

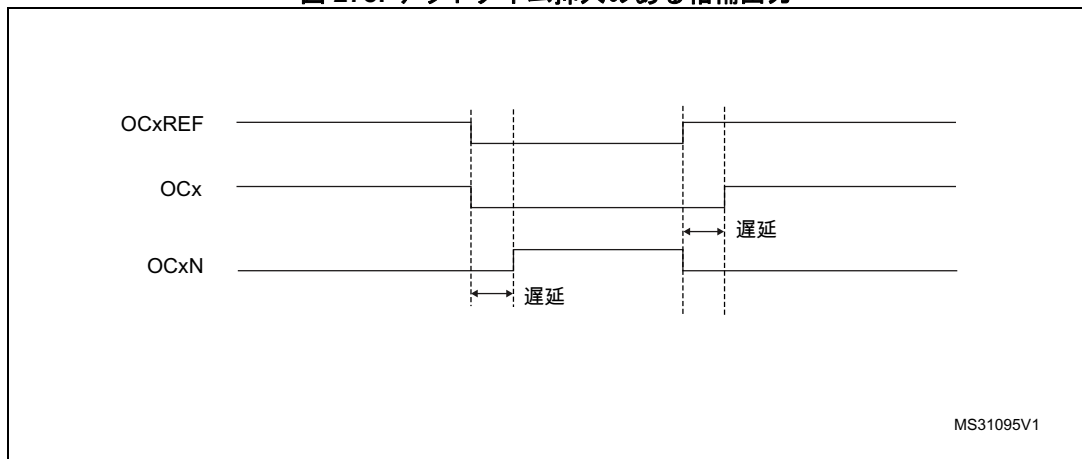


図 279. 負のパルスより長い遅延があるときのデッドタイムの波形

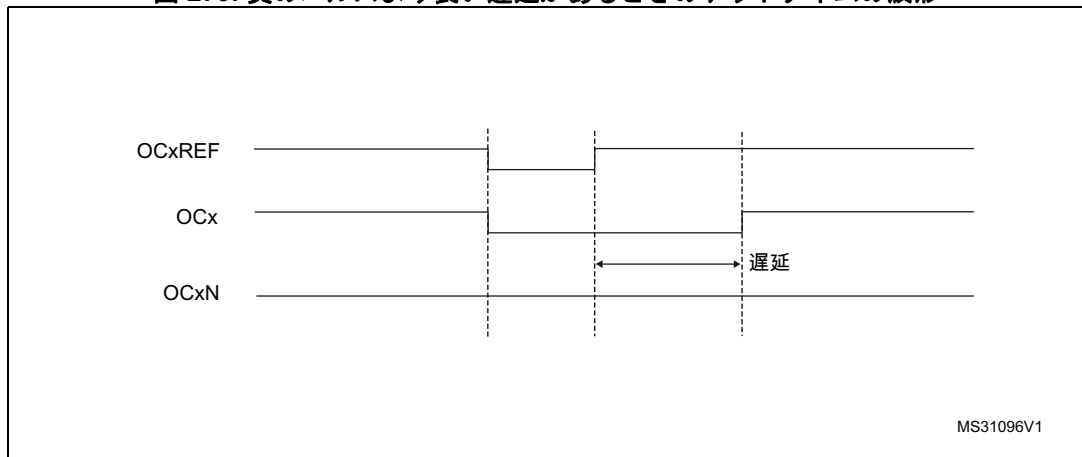
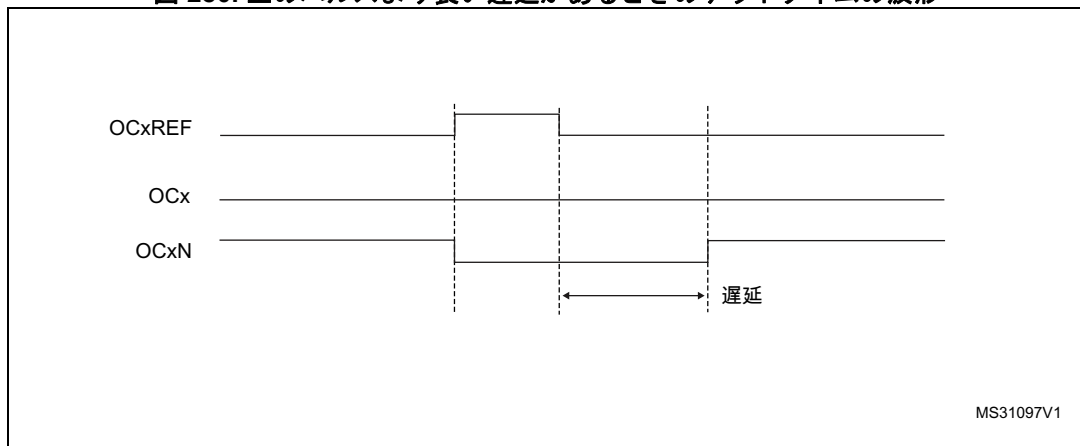


図 280. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャンネルで同じであり、TIMx\_BDTR レジスタの DTG ビットでプログラミングできます。遅延計算については、[セクション 23.5.15 : TIM15 ブレークおよびデッドタイムレジスタ \(TIM15\\_BDTR\) \(731 ページ\)](#) を参照してください。

## OCxREF 信号の OCx または OCxN へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx\_CCER レジスタの CCxE ビットおよび CCxNE ビットを構成することによって、OCxREF 信号を OCx 出力または OCxN 出力にリダイレクトできます。

これにより、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定することができます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができます。

**注：** OCxN のみが有効なときには (CCxE=0, CCxNE=1)、相補にならず、OCxREF がハイレベルとなるとアクティブになります。たとえば、CCxNP=0 の場合は、OCxN=OCxRef です。他方、OCx と OCxN の両方が有効なときには (CCxE=CCxNE=1)、OCxREF がハイになると OCx はアクティブになり、OCxREF がローのときには、OCxN は補信号であり、アクティブになります。

## 23.4.13 ブレーク機能の使用

ブレーク機能の目的は、TIM15/TIM16/TIM17 タイマによって生成される PWM 信号によって駆動する電源スイッチを保護することです。ブレーク入力は通常、パワーステージおよび 3 相インバータの異常出力に接続されています。アクティブ化すると、ブレーク回路は PWM 出力を遮断し、強制的に事前定義された安全な状態に移行させます。

ブレーク機能を使用しているときには、出カインエーブル信号とインアクティブレベルは追加の制御ビット (TIMx\_BDTR レジスタの MOE、OSSI、および OSSI ビットと TIMxCR2 レジスタの OISx および OISxN ビット) に応じて変更されます。ただし、OCx および OCxN 出力の両方を同時にアクティブレベルに設定することはできません。詳細については、[表 129: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット \(728 ページ\)](#) を参照してください。

ブレークソースは次のいずれかです。

- BKIN ピンに接続された外部ソース (BRK に内部的に接続)
- 内部ソース (BRK\_ACTH に内部的に接続):
  - CSS によって生成されたクロック障害イベント。CSS の詳細については、[セクション 9.2.7: クロックセキュリティシステム \(CSS\)](#) を参照してください。
  - コンパレータからの出力
  - PVD 出力
  - SRAM パリティエラー信号
  - Cortex<sup>®</sup>-M4 LOCKUP (ハードフォルト) 出力

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。ブレーク機能は、TIMx\_BDTR レジスタの BKE ビットをセットすることによって有効になります。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。BKE および BKP ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がりエッジは非同期のことがあるので、実際の信号 (出力に作用する信号) と同期制御ビット (TIMx\_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延 (ダミー命令) を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

ブレークは、次を持つ BRK 入力によって生成されます。

- プログラム可能な極性 (TIMx\_BDTR レジスタの BKP ビット)
- プログラム可能なインエーブルビット (TIMx\_BDTR レジスタの BKE ビット)

TIMx\_EGR レジスタの BG ビットを使用してソフトウェアでブレークイベントを生成することもできます。

ブレークが発生すると (ブレーク入力で選択されたレベル)、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態またはアイドル状態になるか、AFIO コントローラへの制御が解除されます (OSSI ビットで選択)。これは、MCU オシレータがオフの場合も同様です。
- 各出力チャンネルは、MOE=0 になったとき、TIMx\_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマは出力の制御 (AFIO コントローラによって引き継がれた) を解除し、そうでない場合、インエーブル出力はハイのままです。
- 相補出力が使用されているときには:
  - 出力は、まずリセット状態のインアクティブ状態に置かれます (極性に依存します)。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。

- タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください (約 2 CK\_TIM クロックサイクル)。
- OSSl=0 の場合、タイマはイネーブル出力 (ハイインピーダンス状態を強制する AFIO コントローラによって引き継がれた) を解除し、そうでない場合、イネーブル出力はハイのままか、CCxE または CCxNE ビットのどちらかがハイになったときにハイになります。
- ブレーク状態フラグ (TIMx\_SR レジスタの BIF ビット) がセットされます。TIMx\_DIER レジスタの BIE ビットがセットされている場合は、割り込みを生成できます。TIMx\_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx\_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント UEV で再び自動的にセットされます。これを使用して、たとえば、レギュレーションを行うことができます。そうでない場合、MOE ビットは、再び 1 が書き込まれるまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

**注：** ブレーク入力は、信号レベルに対して動作します。このため、ブレーク入力アクティブな間は、MOE をセットできません (自動的にも、ソフトウェアによっても)。この間、ステータスフラグ BIF をクリアできません。

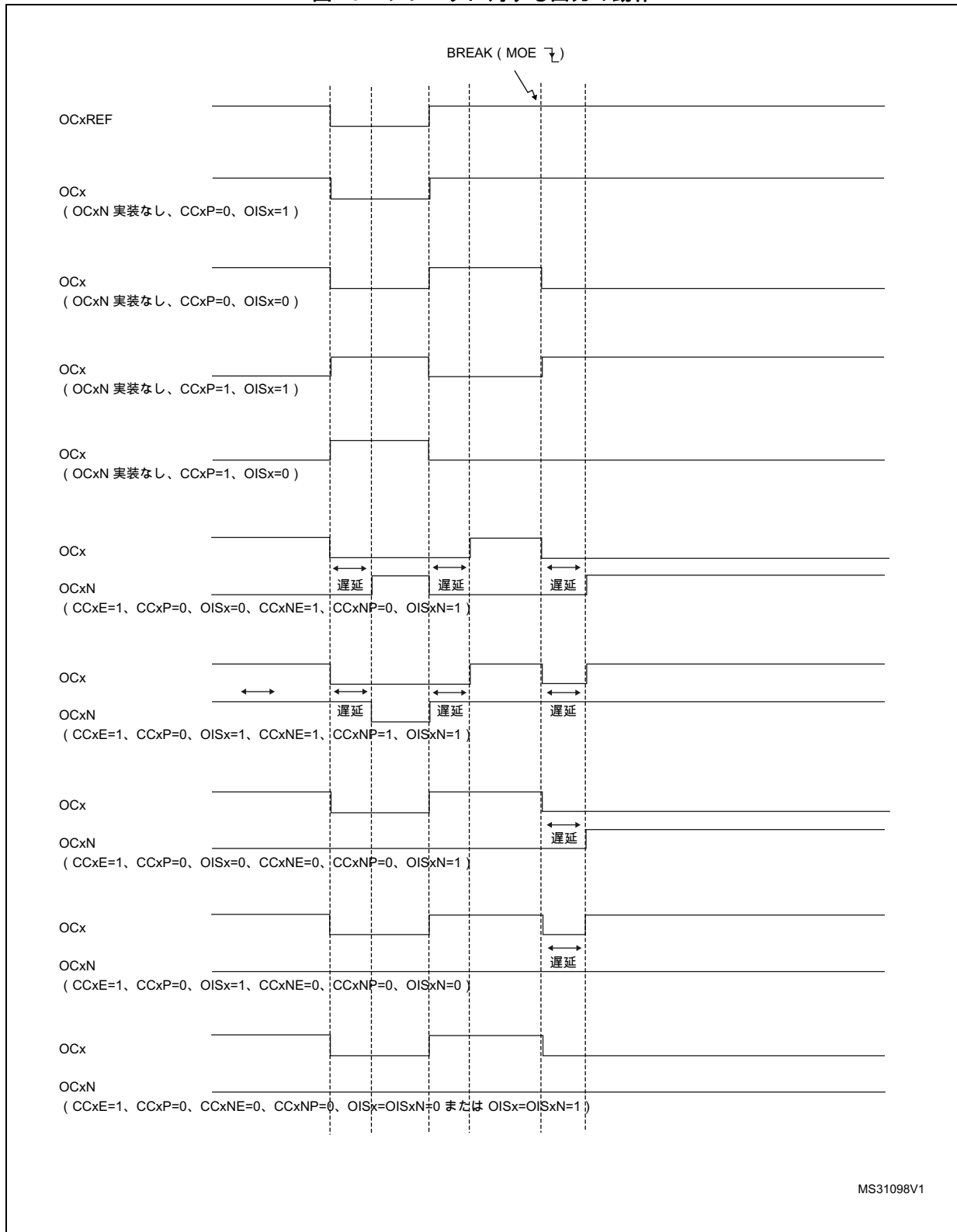
ブレークは、BRK 入力によって生成でき、BRK はプログラミング可能な極性を持ち、TIMx\_BDTR レジスタの BKE がイネーブルビットです。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ (デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性) を固定することができます。TIMx\_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。セクション 23.5.15: TIM15 ブレークおよびデッドタイムレジスタ (TIM15\_BDTR) (731 ページ) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。

図 281 に、ブレークに対する出力の動作例を示します。



図 281. ブレークに対する出力の動作



## 23.4.14 ワンパルスモード

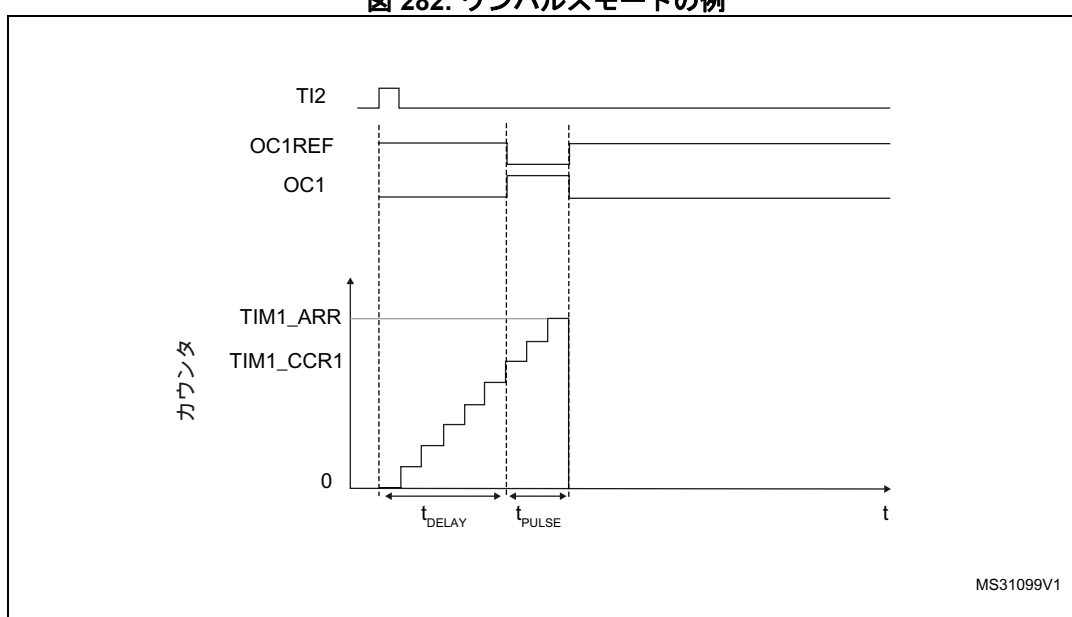
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- $CNT < CCRx \leq ARR$  (特に、 $0 < CCRx$ )

図 282. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が  $t_{PULSE}$  の正のパルスを遅延時間  $t_{DELAY}$  後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

1. TIMx\_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
2. TI2FP2 は、立ち上がりエッジを検出して、TIMx\_CCER レジスタで CC2P=“0”と CC2NP=“0”を書き込みます。
3. TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx\_SMCR レジスタの TS ビットに“110”を書き込みます。
4. TI2FP2 を使用してカウンタを開始します。このためには、TIMx\_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- $t_{DELAY}$  は、TIMx\_CCR1 レジスタに書き込まれた値によって定義されます。
- $t_{PULSE}$  は、自動再ロード値と比較値の差 (TIMx\_ARR - TIMx\_CCR1) によって定義されます。

- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx\_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx\_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx\_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイベントを待ちます。この例では、CC1P に“0”を書き込みます。

必要なパルスは 1 つだけなので、TIMx\_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します（カウンタが自動再ロード値に達して、“0”に戻る時点）。

特殊なケース：OCx 高速イネーブル

ワンパルスモードでは、TIMx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ( $t_{\text{DELAY min}}$ ) が制限されます。

最小遅延で波形を出力したい場合は、TIMx\_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF（および OCx）は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

## 23.4.15 UIF ビットの再配置

TIMx\_CR1 レジスタの IUFREMAP ビットでは、タイマカウンタレジスタのビット 31 (TIMxCNT[31]) に更新割り込みフラグ (UIF) の連続コピーを強制します。これにより、UIFCPY フラグによって示されたカウンタ値と潜在的なロールオーバー条件を分割できないものとして読み取ることができません。特定のケースでは、バックグラウンドタスク（カウンタの読み出し）と割り込み（更新の割り込み）との間で共有されている処理などによって生じる競合状態を避けることで、計算が容易になります。

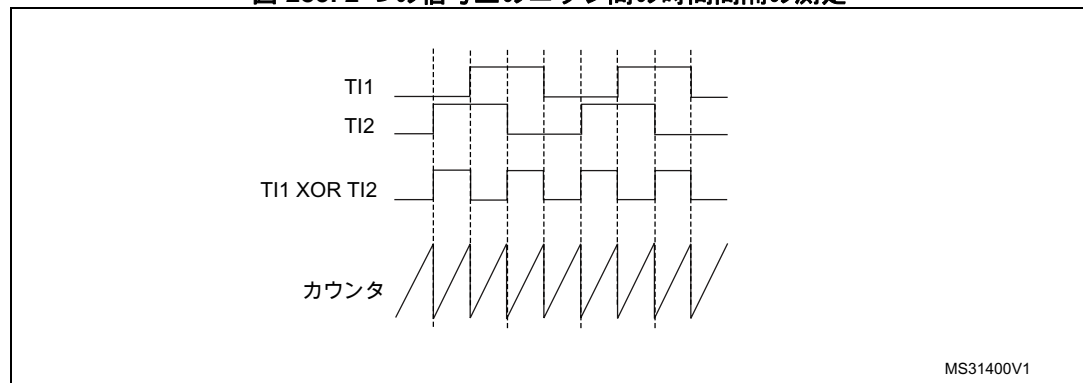
UIF と UIFCPY フラグのアサートの間には、遅延はありません。

## 23.4.16 タイマ入力 XOR 機能 (TIM15 のみ)

TIMx\_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 2 つの入力ピン TIMx\_CH1 および TIMx\_CH2 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。次の [図 283](#) に示すように、2 つの入力信号上のエッジ間の間隔を測定するのに便利です。

図 283. 2 つの信号上のエッジ間の時間間隔の測定



MS31400V1

## 23.4.17 外部トリガ同期 (TIM15 のみ)

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。

TIM15 タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

### スレープモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx\_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx\_ARR、TIMx\_CCRx) が更新されます。

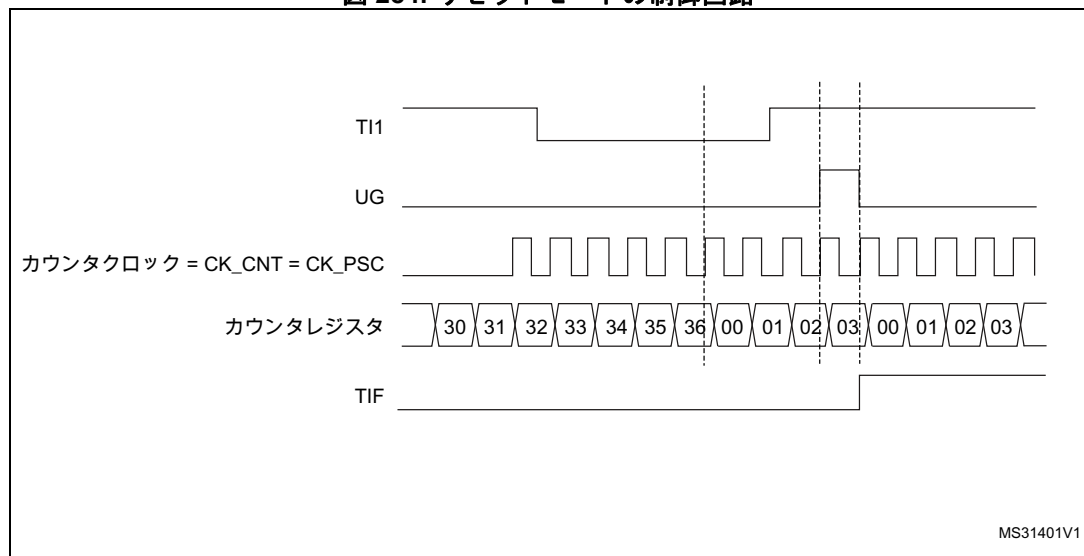
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01)。TIMx\_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx\_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx\_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx\_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 284. リセットモードの制御回路



## スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

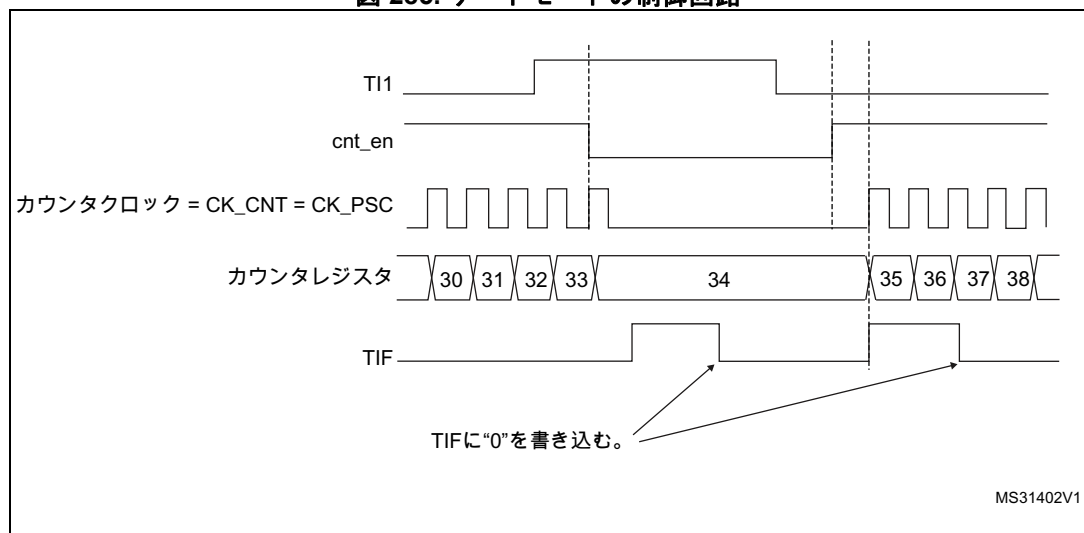
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

1. TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx\_CCMR1 レジスタの CC1S=01 ビット)。TIMx\_CCER レジスタで CC1P = 1 と CC1NP = 0 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx\_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 285. ゲートモードの制御回路



## スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

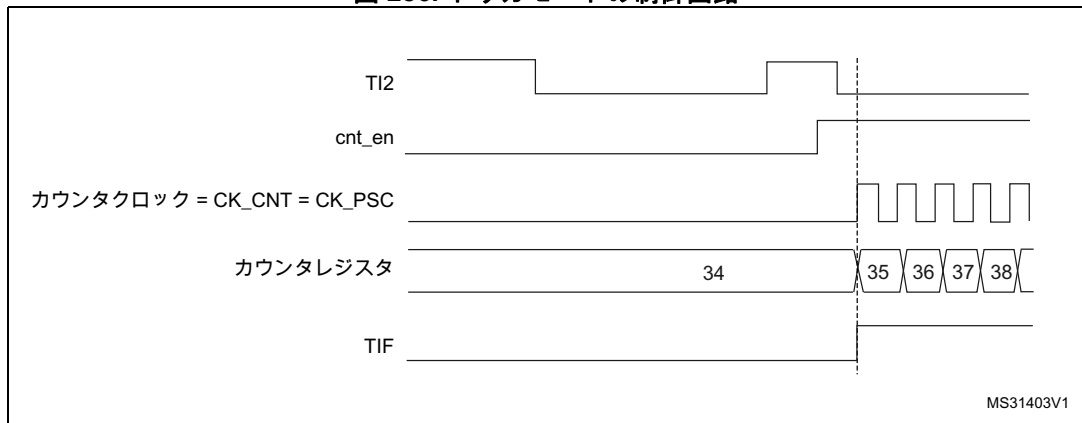
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

1. TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F=0000 のままにしておきます)。キャプチャプリスケラはトリガには使用されないため、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx\_CCMR1 レジスタの CC2S=01)。TIMx\_CCER レジスタで CC2P に「1」、CC2NP に「0」を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
2. TIMx\_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx\_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 286. トリガモードの制御回路



### 23.4.18 スレーブモード：リセットモードとトリガモードの組み合わせ (TIM15 のみ)

この場合、選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成し、カウンタを開始します。

このモードはワンパルスモードで使用します。

### 23.4.19 DMA バーストモード

TIMx タイマには、1 つのイベントで多重 DMA リクエストを生成する機能があります。主な目的は、いくつかのタイマレジスタをソフトウェアのオーバーヘッドなく複数回再プログラムできるようにすることです。複数のレジスタを連続して一定の時間間隔で読み出すために使用することもできます。

DMA コントローラの転送先は一意で、仮想レジスタ TIMx\_DMAR を示している必要があります。特定のタイマイベントで、タイマは一連の DMA リクエスト (バースト) を開始します。TIMx\_DMAR レジスタへの各書き込みは、実際にタイマレジスタの 1 つにリダイレクトされます。

TIMx\_DCR レジスタの DBL[4:0] ビットによって、DMA バースト長がセットされます。タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します。つまり、転送数 (ハーフワードまたはバイト) です。

TIMx\_DCR レジスタの DBA[4:0] ビットは、DMA 転送の DMA ベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx\_CR1、

00001 : TIMx\_CR2、

00010 : TIMx\_SMCR

たとえば、更新イベント時に CCRx レジスタ値の内容を更新するためにタイマ DMA バースト機能を使用します (x = 2, 3, 4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
  - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
  - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
  - 転送データ数 = 3 とします (下の注を参照)。
  - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。  
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタの UDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注 :

この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。



## 23.4.20 タイマ同期 (TIM15)

タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。詳細については、[セクション 21.3.19 : タイマの同期](#)を参照してください。

**注：** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

## 23.4.21 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M4<sup>®</sup>F コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG\_TIMx\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 33.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。

## 23.5 TIM15 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1](#) を参照してください。

### 23.5.1 TIM15 制御レジスタ 1 (TIM15\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rW		rW	rW	rW				rW	rW	rW	rW

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (TIx) によって使用されるデッドタイムおよびサンプリングクロック ( $t_{DTS}$ ) との間の分周比を示します。

00 :  $t_{DTS} = t_{CK\_INT}$

01 :  $t_{DTS} = 2 * t_{CK\_INT}$

10 :  $t_{DTS} = 4 * t_{CK\_INT}$

11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

**ビット 2 URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みを生成します (有効な場合)。

**ビット 1 UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレープモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。ただし、UG ビットがセットされた場合や、スレープモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケラは再初期化されます。

**ビット 0 CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

**注 :** 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできません。

## 23.5.2 TIM15 制御レジスタ 2 (TIM15\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rW	rW	rW	rW	rW	rW	rW	rW	rW		rW

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

**ビット 10 OIS2** : 出力アイドル状態 2 (OC2 出力)

0 : MOE = 0 のとき、OC2=0

1 : MOE = 1 のとき、OC2=0

**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIM<sub>x</sub>\_BKR レジスタの LOCK ビット)、変更できません。

**ビット 9 OIS1N** : 出力アイドル状態 1 (OC1N 出力)

0 : MOE=0 のとき、デッドタイム後に OC1N=0

1 : MOE=1 のとき、デッドタイム後に OC1N=0

**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIM<sub>x</sub>\_BKR レジスタの LOCK ビット)。



ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)

0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx\_BKR レジスタの LOCK ビット)。

ビット 7 **TI1S** : TI1 選択

0 : TIMx\_CH1 ピンが TI1 入力に接続されます。

1 : TIMx\_CH1 および CH2 ピンが TI1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[1:0]** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx\_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT\_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、GEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在しません (TIMx\_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケラとして使用できます。

011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる時 (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。(TRGO)

100 : **比較** - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : **比較** - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ/比較制御更新選択

0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。

1 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ/比較プリロード制御

0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。

1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込みの後、これらのビットは、転換イベント (COM) が発生した時にのみ更新されます (CCUS ビットに応じて、COMG ビットがセットまたは TRGI で立ち上がりエッジが検出されたとき)。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

## 23.5.3 TIM15 のスレーブモード制御レジスタ (TIM15\_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SMS[3]** : スレーブモード選択 - ビット 3

SMS 説明を参照 - ビット 2:0

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS[2:0]** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

000 : 内部トリガ 0 (ITR0)

001 : 内部トリガ 1 (ITR1)

010 : 内部トリガ 2 (ITR2)

011 : 内部トリガ 3 (ITR3)

100 : TI1 エッジ検出回路 (TI1F\_ED)

101 : フィルタタイマ入力 1 (TI1FP1)

110 : フィルタタイマ入力 2 (TI2FP2)

各タイマでの ITRx の詳細については、表 128 : TIMx 内部トリガ接続 (718 ページ) を参照してください。

**注 :** 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

0000: スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

0001: 予約済み

0010: 予約済み

0011: 予約済み

0100: リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

0101: ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

0110: トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

0111: 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

1000: リセットモードとトリガモードの組み合わせ - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成してカウンタを開始します。

その他: 予約済み

**注:** トリガ入力として **T11F\_ED** が選択されている場合 (**TS="100"**)、ゲートモードを使用することはできません。**T11F\_ED** は **T11F** の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

**注:** スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 128. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM15	TIM2	TIM3	TIM16 OC1	TIM17 OC1

## 23.5.4 TIM15 DMA/割り込み有効レジスタ (TIM15\_DIER)

アドレスオフセット: 0x0C

リセット値: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	r/w	r/w			r/w	r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE**: トリガ DMA リクエストイネーブル

0: トリガ DMA リクエストは無効です。

1: トリガ DMA リクエストは有効です。

ビット 13 **COMDE**: COM DMA リクエストイネーブル

0: COM DMA リクエストは無効です。

1: COM DMA リクエストは有効です。

ビット 12:11 予約済みであり、リセット値に保持する必要があります。

- ビット 10 **CC2DE** : キャプチャ/比較 2 DMA リクエストイネーブル  
 0 : CC2 DMA リクエストは無効です。  
 1 : CC2 DMA リクエストは有効です。
- ビット 9 **CC1DE** : キャプチャ/比較 1 DMA リクエストイネーブル  
 0 : CC1 DMA リクエストは無効です。  
 1 : CC1 DMA リクエストは有効です。
- ビット 8 **UDE** : 更新 DMA リクエストイネーブル  
 0 : 更新 DMA リクエストは無効です。  
 1 : 更新 DMA リクエストは有効です。
- ビット 7 **BIE** : ブレーク割り込みイネーブル  
 0 : ブレーク割り込みは無効です。  
 1 : ブレーク割り込みは有効です。
- ビット 6 **TIE** : トリガ割り込みイネーブル  
 0 : トリガ割り込みは無効です。  
 1 : トリガ割り込みは有効です。
- ビット 5 **COMIE** : COM 割り込みイネーブル  
 0 : COM 割り込みは無効です。  
 1 : COM 割り込みは有効です。
- ビット 4:3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **CC2IE** : キャプチャ/比較 2 割り込みイネーブル  
 0 : CC2 割り込みは無効です。  
 1 : CC2 割り込みは有効です。
- ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル  
 0 : CC1 割り込みは無効です。  
 1 : CC1 割り込みは有効です。
- ビット 0 **UIE** : 更新割り込みイネーブル  
 0 : 更新割り込みは無効です。  
 1 : 更新割り込みは有効です。

## 23.5.5 TIM15 ステータスレジスタ (TIM15\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

- ビット 15:11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ  
 CC1OF の説明を参照してください。
- ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ  
 このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。  
 0 : オーバキャプチャは検出されていません。  
 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。
- ビット 8 予約済みであり、リセット値に保持する必要があります。



**ビット 7 BIF** : ブレーク割り込みフラグ

このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。

- 0 : ブレークイベントは発生していません。
- 1 : ブレーク入力アクティブレベルが検出されました。

**ビット 6 TIF** : トリガ割り込みフラグ

このフラグは、トリガイベント時 (スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力アクティブエッジが検出されたとき、またはゲートモードが選択されている場合は、両方のエッジが検出されたとき) にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

- 0 : トリガイベントは発生していません。
- 1 : トリガ割り込みが保留中です。

**ビット 5 COMIF** : COM 割り込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき)。ソフトウェアによってクリアされます。

- 0 : COM イベントは発生していません。
- 1 : COM 割り込みがペンディング中です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

**ビット 2 CC2IF** : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

**ビット 1 CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合** : このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : 一致していません。
- 1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。TIMx\_CCR1 の内容が TIMx\_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

**CC1 チャンネルが入力として設定されている場合** : このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

- 0 : 入力キャプチャは発生していません。
- 1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

**ビット 0 UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

- 0 : 更新は発生していません。
- 1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。
  - 繰り返しカウンタ値に関するオーバーフロー (繰り返しカウンタ=0 の場合の更新)、および TIMx\_CR1 レジスタの UDIS=0 の場合。
  - TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
  - TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき ( [セクション 23.5.3 : TIM15 のスレーブモード制御レジスタ \(TIM15\\_SMCR\)](#) を参照)。



## 23.5.6 TIM15 のイベント生成レジスタ (TIM15\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			w	w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx\_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2G** : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

**CC1 チャンネルが出力として設定されている場合 :**

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

**CC1 チャンネルが入力として設定されている場合 :**

カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

## 23.5.7 TIM15 のキャプチャ/比較モードレジスタ 1 (TIM15\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							Res.								Res.
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]			IC2PSC[1:0]					IC1F[3:0]			IC1PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

## 出力比較モード :

ビット 31:25 予約済み。常に 0 として読み出されます。

ビット 24 **OC2M[3]** : 出力比較 2 モード - ビット 3

ビット 23:17 予約済み。常に 0 として読み出されます。

ビット 16 **OC1M[3]** : 出力比較 1 モード - ビット 3

ビット 6:4 の OC1M 説明を参照

ビット 15 予約済み。常に 0 として読み出されます。

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC2S** は、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

0 : OC1Ref は OCREF\_CLR 入力の影響を受けません。

1 : OC1Ref は OCREF\_CLR 入力のハイレベルが検出されるとクリアされます。

## ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000 : 停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません。

0001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

0010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

0011 : 反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

0100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

0101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

0110 : PWM モード 1 - チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

0111 : PWM モード 2 - チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

1000 : 予約済み。

1001 : 予約済み。

1010 : 予約済み。

1011 : 予約済み。

1100 : 組み合わせ PWM モード 1 - OC1REF は、PWM モード 1 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 OR です。

1101 : 組み合わせ PWM モード 2 - OC1REF は、PWM モード 2 と同様に挙動します。OC1REFC は、OC1REF と OC2REF との論理 AND です。

1110 : 予約済み。

1111 : 予約済み。

**注 :** 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードでは、比較結果が変化したとき、または出力比較モードが停止モードから PWM モードに変更されたときのみ、OCREF のレベルが変化します。

3 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときのみプリロードから新しい値をとります。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われず。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注 :** 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 入力キャプチャモード

ビット 31:16 予約済み。常に 0 として読み出されます。

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** CC2S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC2E=0) のときのみ書き込み可能です。



ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

- 0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。
- 0001 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2
- 0010 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4
- 0011 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8
- 0100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2
- 0101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2
- 0110 :  $f_{SAMPLING} = f_{DTS}/4$ 、N=6
- 0111 :  $f_{SAMPLING} = f_{DTS}/4$ 、N=8
- 1000 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8
- 1001 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8
- 1010 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16
- 1011 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16
- 1100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16
- 1101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32
- 1110 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32
- 1111 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。プリスケアラは、CC1E = 0 (TIMx\_CCER レジスタ) になるとリセットされます。

- 00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。
- 01 : キャプチャは、2 イベントごとに行われます。
- 10 : キャプチャは、4 イベントごとに行われます。
- 11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向 (入力 / 出力) と、使用される入力を定義します。

- 00 : CC1 チャンネルは出力として設定されます。
- 01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。
- 10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。
- 11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

**注 :** **CC1S** ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

## 23.5.8 TIM15 のキャプチャ / 比較有効レジスタ (TIM15\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rW		rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CC2NP** : キャプチャ / 比較 2 相補出力極性  
CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ/比較 2 出力極性  
CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ/比較 2 出力イネーブル  
CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性  
CC1 チャンネルが出力として設定されている場合 :  
0 : OC1N はアクティブハイです。  
1 : OC1N はアクティブローです。  
CC1 チャンネルが入力として設定されている場合 :  
このビットは、TI1FP1とTI2FP1の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

**注 :**

- このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。
- 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 2 **CC1NE** : キャプチャ/比較 1 相補出力イネーブル  
0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1Nおよび CC1E ビットによって決まります。  
1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性  
**CC1 チャンネルが出力として設定されている場合 :**  
0 : OC1 はアクティブハイです。  
1 : OC1 はアクティブローです。  
**CC1 チャンネルが入力として設定されている場合 :** CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 の極性を選択します。  
00: 非反転/立ち上がりエッジ。この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。  
01: 反転/立ち下がりエッジ。この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 が反転されず (ゲートモードでのトリガ動作)。  
10: 予約済み。この設定は使用しないでください。  
11: 非反転/両エッジ。この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

**注 :**

- このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx\_BDTR レジスタの LOCK ビット)。
- 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット0 **CC1E** : キャプチャ/比較 1 出力イネーブル

**CC1 チャンネルが出力として設定されている場合 :**

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :** このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

**表 129. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット**

制御ビット					出力状態 <sup>(1)</sup>	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	X	0	0	出力無効 (タイマによって駆動されない : ハイインピーダンス) OCx=0 OCxN=0, OCxN_EN=0	
		0	0	1	出力無効 (タイマによって駆動されない : ハイインピーダンス) OCx=0	OCxREF + 極性 OCxN=OCxREF XOR CCxNP
		0	1	0	OCxREF + 極性 OCx=OCxREF XOR CCxP	出力無効 (タイマによって駆動されない : ハイインピーダンス) OCxN=0
		X	1	1	OCREF + 極性 + デッドタイム	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム
		1	0	1	オフ状態 (インアクティブ状態で出力有効) OCx=CCxP	OCxREF + 極性 OCxN=OCxREF XOR CCxNP
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP, OCx_EN=1	オフ状態 (インアクティブ状態で出力有効) OCxN=CCxNP, OCxN_EN=1
0	1	X	X	X	出力無効 (タイマによって駆動されない : ハイインピーダンス) OCx=CCxP, OCxN=CCxNP	
			0	0	オフ状態 (インアクティブ状態で出力有効)	
			0	1	非同期 : OCx=CCxP, OCxN=CCxNP	
			1	0	クロックが存在する場合 : デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCx と OCxN の両方に対応しないことを前提とします。	
			1	1		

1. チャンネルの両方の出力が使用されないとき (GPIO コントローラが制御を引き継いだ場合)、OISx、OISxN、CCxP、および CCxNP ビットはクリアされたままでなければなりません。

**注 :** **相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OCx および OCxN チャンネルの状態と、AFIO レジスタに依存します。**



## 23.5.9 TIM15 のカウンタ (TIM15\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

## 23.5.10 TIM15 のプリスケアラ (TIM15\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 (CK\_CNT) は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx\_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

## 23.5.11 TIM15 の自動再ロードレジスタ (TIM15\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **ARR[15:0]** : プリスケアラ値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 23.4.1 : タイムベースユニット \(683 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。



## 23.5.12 TIM15 繰り返しカウンタレジスタ (TIM15\_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]** : 繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP\_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U\_RC のみ、REP\_CNT に REP 値がロードされるので、TIMx\_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) はエッジアラインモードで PWM 周期の数を意味します。

## 23.5.13 TIM15 のキャプチャ/比較レジスタ 1 (TIM15\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値

**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値（プリロード値）です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**CC1 チャンネルが入力として設定されている場合 :**

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 23.5.14 TIM15 のキャプチャ/比較レジスタ 2 (TIM15\_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR2[15:0]** : キャプチャ/比較 2 値

**CC2 チャンネルが出力として設定されている場合 :**

CCR2 は、実際のキャプチャ/比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx\_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 2 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較され、OC2 出力に送信される値を含みます。

**CC2 チャンネルが入力として設定されている場合 :**

CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

## 23.5.15 TIM15 ブレークおよびデッドタイムレジスタ (TIM15\_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

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

**注 :** **AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] ビットは、LOCK 設定に応じて書き込みがロックされるので、TIMx\_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。**

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **MOE** : メイン出力イネーブル

このビットは、ブレーク入力 that アクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : OC および OCN 出力が無効か、OSSI ビットによって強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx\_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 23.5.8 : TIM15 のキャプチャ/比較有効レジスタ \(TIM15\\_CCER\) \(726 ページ\)](#))。

ビット 14 **AOE** : 自動出力イネーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力 that アクティブでない場合)。

**注 :** **このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。**



ビット 13 **BKP** : ブレーク極性

- 0 : ブレーク入力 BRK はアクティブローです。
- 1 : ブレーク入力 BRK はアクティブハイです。

**注 :** 1 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

2 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 **BKE** : ブレークイネーブル

- 0 : ブレーク入力 (BRK および CCS クロック障害イベント) は無効です。
- 1 : ブレーク入力 (BRK および CCS クロック障害イベント) は有効です。

このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

**注 :** このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 **OSSR** : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 23.5.8 : TIM15 のキャプチャ/比較有効レジスタ (TIM15\_CCER) (726 ページ))。

0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する AFIO ロジックによって引き継がれます)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります (出力は引き続きタイマで制御される)。

**注 :** このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

ビット 10 **OSSI** : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 23.5.8 : TIM15 のキャプチャ/比較有効レジスタ (TIM15\_CCER) (726 ページ))。

0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、まず強制的にアイドルレベルになります (OC/OCN イネーブル出力信号 = 1)。

**注 :** このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

ビット 9:8 **LOCK[1:0]** : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 = TIMx\_BDTR レジスタの DTG ビット、TIMx\_CR2 レジスタの OISx および OISxN ビット、および TIMx\_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込みできなくなります。

11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCMR レジスタの OCxM および OCxPE ビット) が書き込みできなくなります。

**注 :** LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx\_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

DTG[7:5]=0xx => DT=DTG[7:0]x t<sub>dtg</sub>、ここで t<sub>dtg</sub>=t<sub>DTS</sub>  
 DTG[7:5]=10x => DT=(64+DTG[5:0])x t<sub>dtg</sub>、ここで T<sub>dtg</sub>=2x t<sub>DTS</sub>  
 DTG[7:5]=110 => DT=(32+DTG[4:0])x t<sub>dtg</sub>、ここで T<sub>dtg</sub>=8x t<sub>DTS</sub>  
 DTG[7:5]=111 => DT=(32+DTG[4:0])x t<sub>dtg</sub>、ここで T<sub>dtg</sub>=16x t<sub>DTS</sub>  
 例 : T<sub>DTS</sub>=125ns (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。  
 0 から 15875 ns (125 ns 単位)  
 16 μs から 31750 ns (250 ns 単位)  
 32 μs から 63 μs (1 μs 単位)  
 64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

## 23.5.16 TIM15 DMA 制御レジスタ (TIM15\_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのフィールドは、DMA 転送長 (タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します) を指定します。

00000 : 1 回転送  
 00001 : 2 回転送、  
 00010 : 3 回転送、  
 ...  
 10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのフィールドは、DMA 転送のベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :  
 00000 : TIMx\_CR1、  
 00001 : TIMx\_CR2、  
 00010 : TIMx\_SMCR  
 ...

## 23.5.17 完全転送の TIM15 DMA アドレス (TIM15\_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(TIMx\_CR1 \text{ アドレス}) + (DBA + \text{DMA インデックス}) \times 4$$

ここで、TIMx\_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx\_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx\_DCR 内で設定)。

## 23.5.18 TIM15 レジスタマップ

TIM15 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 130. TIM15 レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	TIM15_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UJFREMAP	Res	Res	CKD [1:0]	ARPE	Res	Res	Res	Res	OPM	URS	UDIS	CEN		
	リセット値																						0		0	0	0				0	0	0	0		
0x04	TIM15_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	OIS2	OIS1N	OIS1	T11S	MMS[2:0]			CCDS	CCUS	Res	CCPC			
	リセット値																							0	0	0	0	0	0	0	0	0		0		
0x08	TIM15_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SMS[3]	Res	Res	Res	Res	Res	Res	Res	Res	MSM	TS[2:0]			Res	SMS[2:0]					
	リセット値																0										0	0	0			0	0	0		
0x0C	TIM15_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	Res	Res	Res	Res	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res	Res	Res	CC2IE	CC1IE	UIE
	リセット値																			0	0				0	0	0	0	0	0				0	0	0
0x10	TIM15_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC2OF	CC1OF	Res	BIF	TIF	COMIF	Res	Res	Res	CC2IF	CC1IF	UIF
	リセット値																								0	0			0	0	0				0	0
0x14	TIM15_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BG	TG	COMG	Res	Res	Res	CC2G	CC1G	UG	
	リセット値																											0	0	0				0	0	0

表 130. TIM15 レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x18	TIM15_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]			
	リセット値								0								0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x20	TIM15_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F [3:0]			IC2PSC [1:0]	CC2S [1:0]		IC1F [3:0]			IC1PSC [1:0]	CC1S [1:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x20	TIM15_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																									0	0	0	0	0	0	0	0		
0x24	TIM15_CNT	UIFCPY または Res.																CNT[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x28	TIM15_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x2C	TIM15_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																	
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x30	TIM15_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																										0	0	0	0	0	0	0		
0x34	TIM15_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x38	TIM15_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x44	TIM15_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x48	TIM15_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																																		
0x4C	TIM15_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



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

## 23.6 TIM16/TIM17 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

### 23.6.1 TIM16/TIM17 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF REM- AP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rW		rW	rW	rW				rW	rW	rW	rW

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **UIFREMAP** : UIF ステータスビットの再配置

0 : 再配置なし。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされません。

1 : 再配置は有効です。UIF ステータスビットは TIMx\_CNT レジスタのビット 31 にコピーされます。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK\_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (Tix) によって使用されるデッドタイムおよびサンプリングクロック (t<sub>DTS</sub>) との間の分周比を示します。

00 : t<sub>DTS</sub>=t<sub>CK\_INT</sub>

01 : t<sub>DTS</sub>=2\*t<sub>CK\_INT</sub>

10 : t<sub>DTS</sub>=4\*t<sub>CK\_INT</sub>

11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx\_ARR レジスタはバッファされません。

1 : TIMx\_ARR レジスタはバッファされます。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します (有効な場合)。これらのイベントは、次のとおりです。

- カウンタオーバーフロー/アンダーフロー

- UG ビットのセット

- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みまたは DMA リクエストを生成します (有効な場合)。



**ビット 1 UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー/アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

パツファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR<sub>x</sub>) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

**ビット 0 CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

**注 :** 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできません。

## 23.6.2 TIM16/TIM17 制御レジスタ 2 (TIM<sub>x</sub>\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

**ビット 9 OIS1N** : 出力アイドル状態 1 (OC1N 出力)

0 : MOE=0 のとき、デッドタイム後に OC1N=0

1 : MOE=1 のとき、デッドタイム後に OC1N=0

**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIM<sub>x</sub>\_BKR レジスタの LOCK ビット)。

**ビット 8 OIS1** : 出力アイドル状態 1 (OC1 出力)

0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

**注 :** このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIM<sub>x</sub>\_BKR レジスタの LOCK ビット)。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

**ビット 3 CCDS** : キャプチャ/比較 DMA 選択

0 : CC<sub>x</sub> DMA リクエストは、CC<sub>x</sub> イベントが発生すると送信されます。

1 : CC<sub>x</sub> DMA リクエストは、更新イベントが発生すると送信されます。

**ビット 2 CCUS** : キャプチャ/比較制御更新選択

0 : キャプチャ/比較制御ビットがプリロードされる際には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。

1 : キャプチャ/比較制御ビットがプリロードされる際には (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。



ビット 0 **CCPC** : キャプチャ / 比較プリロード制御

0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。

1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込み後は、COM ビットがセットされた場合にのみ更新されます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

## 23.6.3 TIM16/TIM17 DMA / 割り込み有効レジスタ (TIMx\_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
	rw	rw				rw	rw	rw		rw				rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル

0 : COM DMA リクエストは無効です。

1 : COM DMA リクエストは有効です。

ビット 12:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1DE** : キャプチャ / 比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割り込みイネーブル

0 : ブレーク割り込みは無効です。

1 : ブレーク割り込みは有効です。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **COMIE** : COM 割り込みイネーブル

0 : COM 割り込みは無効です。

1 : COM 割り込みは有効です。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ / 比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

## 23.6.4 TIM16/TIM17 ステータスレジスタ (TIMx\_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC10F	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC10F** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BIF** : ブレーク割り込みフラグ

このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク入力アクティブレベルが検出されました。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **COMIF** : COM 割り込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE, CCxNE, OCxM - が更新されたとき)。ソフトウェアによってクリアされます。

0 : COM イベントは発生していません。

1 : COM 割り込みがペンディング中です。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

**CC1 チャンネルが出力として設定されている場合 :**

このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx\_CNT の内容が TIMx\_CCR1 レジスタの内容と一致しました。TIMx\_CCR1 の内容が TIMx\_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

**CC1 チャンネルが入力として設定されている場合 :**

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx\_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

**ビット 0 UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフロー (繰り返しカウンタ=0 の場合の更新)、および TIMx\_CR1 レジスタの UDIS=0 の場合。
- TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx\_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx\_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき (セクション 23.5.3 : TIM15 のスレーブモード制御レジスタ (TIM15\_SMCR) を参照)。

## 23.6.5 TIM16/TIM17 イベント生成レジスタ (TIMx\_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								w		w				w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

**ビット 7 BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 予約済みであり、リセット値に保持する必要があります。

**ビット 5 COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

**注 :** このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4:2 予約済みであり、リセット値に保持する必要があります。

**ビット 1 CC1G** : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

**CC1 チャンネルが出力として設定されている場合 :**

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

**CC1 チャンネルが入力として設定されている場合 :**

カウンタの現在値が TIMx\_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。

## 23.6.6 TIM16/TIM17 キャプチャ/比較モードレジスタ 1 (TIMx\_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	OC1M [3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
																r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]		
									IC1F[3:0]			IC1PSC[1:0]				
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	



## 出力比較モード：

ビット 31:17 予約済み。常に 0 として読み出されます。

ビット 16 **OC1M[3]**：出力比較 1 モード (ビット 3)

ビット 15:8 予約済み

ビット 7 **OC1CE**：出力比較 1 クリアイネーブル

0：OC1Ref は OCREF\_CLR 入力の影響を受けません。

1：OC1Ref は OCREF\_CLR 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 **OC1M[2:0]**：出力比較 1 モード (ビット 2～0)

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

0000：停止 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT との間の比較結果は出力に影響しません。

0001：一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にハイになります。

01010：一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx\_CNT がキャプチャ/比較レジスタ 1 (TIMx\_CCR1) と一致したときに、強制的にローになります。

0011：反転 - TIMx\_CNT = TIMx\_CCR1 のとき、OC1REF は反転します。

0100：強制非アクティブレベル - OC1REF は強制的にローになります。

0101：強制アクティブレベル - OC1REF は強制的にハイになります。

0110：PWM モード 1 - チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

0111：PWM モード 2 - チャンネル 1 は、TIMx\_CNT < TIMx\_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

その他の値：予約済み

**注：** 1：これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2：PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときのみ、OCREF のレベルが変化します。

ビット 3 **OC1PE**：出力比較 1 プリロードイネーブル

0：TIMx\_CCR1 のプリロードレジスタは無効です。TIMx\_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1：TIMx\_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx\_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

**注：** 1：これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2：PWM モードは、ワンパルスモード (TIMx\_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE**：出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0：CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1：トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。**

## 入力キャプチャモード

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは  $f_{DTS}$  で行われます。

0001 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2

0010 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4

0011 :  $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8

0100 :  $f_{SAMPLING} = f_{DTS}/2$ 、N =

0101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 2

0110 :  $f_{SAMPLING} = f_{DTS}/4$ 、N=6

0111 :  $f_{SAMPLING} = f_{DTS}/4$ 、N=8

1000 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8

1001 :  $f_{SAMPLING} = f_{DTS}$ 、N = 8

1010 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1011 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1100 :  $f_{SAMPLING} = f_{DTS}$ 、N = 16

1101 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

1110 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

1111 :  $f_{SAMPLING} = f_{DTS}$ 、N = 32

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx\_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ /比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx\_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

**注 :** **CC1S ビットは、チャンネルがオフ (TIMx\_CCER レジスタの CC1E=0) のときのみ書き込み可能です。**

## 23.6.7 TIM16/TIM17 キャプチャ/比較有効レジスタ (TIMx\_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												r/w	r/w	r/w	r/w

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 :

- 0 : OC1N はアクティブハイです。
- 1 : OC1N はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

このビットは、TI1FP1とTI2FP1 の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

- 注 :**
1. このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx\_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。
  2. 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。



ビット 2 **CC1NE** : キャプチャ/比較 1 相補出力イネーブル

0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

ビット 1 **CC1P** : キャプチャ/比較 1 出力極性

**CC1 チャンネルが出力として設定されている場合 :**

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

**CC1 チャンネルが入力として設定されている場合 :**

CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 の極性を選択します。

00 : 非反転/立ち上がりエッジ。この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

01 : 反転/立ち下がりエッジ。この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 が反転されます (ゲートモードでのトリガ動作)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ。この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

**注 :** 1. このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx\_BDTR レジスタの LOCK ビット)。

2. 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx\_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 0 **CC1E** : キャプチャ/比較 1 出力イネーブル

**CC1 チャンネルが出力として設定されている場合 :**

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

**CC1 チャンネルが入力として設定されている場合 :**

このビットによって、カウンタ値のキャプチャ/比較レジスタ 1 (TIMx\_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

表 131. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット

制御ビット					出力状態 <sup>(1)</sup>	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	X	0	0	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCx=0	OCxREF + 極性 OCxN=OCxREF XOR CCxNP
		0	1	0	OCxREF + 極性 OCx=OCxREF XOR CCxP	出力無効 (タイマによって駆動されない: ハイインピーダンス) OCxN=0
		X	1	1	OCREF + 極性 + デッドタイム	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム
		1	0	1	オフ状態 (インアクティブ状態で出力有効) OCx=CCxP	OCxREF + 極性 OCxN=OCxREF XOR CCxNP
		1	1	0	OCxREF + 極性 OCx=OCxREF XOR CCxP、 OCx_EN=1	オフ状態 (インアクティブ状態で出力有効) OCxN=CCxNP、OCxN_EN=1
0	0	X	X	X	出力無効 (タイマによって駆動されない: ハイインピーダンス)	
			0	0	OCx=CCxP、OCxN=CCxNP	
	1		0	1	オフ状態 (インアクティブ状態で出力有効)	
			1	0	非同期: OCx=CCxP、OCxN=CCxNP	
			1	1	クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCx と OCxN の両方に対応しないことを前提とします。	

1. チャンネルの両方の出力が使用されないとき (GPIO コントローラが制御を引き継いだ場合)、OISx、OISxN、CCxP、および CCxNP ビットはクリアされたままでなければなりません。

**注:** 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OCx および OCxN チャンネルの状態と、AFIO レジスタに依存します。

## 23.6.8 TIM16/TIM17 カウンタ (TIMx\_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31 **UIFCPY** : UIF コピー

このビットは TIMx\_ISR レジスタの UIF ビットの読み出し専用コピー。TIMx\_CR1 の UIFREMAP ビットがリセットされると、ビット 31 は予約済みで、0 で読み出されます。

ビット 30:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **CNT[15:0]** : カウンタ値

## 23.6.9 TIM16/TIM17 プリスケーラ (TIMx\_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケーラ値

カウンタクロック周波数 (CK\_CNT) は  $f_{CK\_PSC} / (PSC[15:0] + 1)$  に等しいです。

PSC は、更新イベントごとにアクティブプリスケーラレジスタにロードされる値を含みます (更新イベントには、TIMx\_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

## 23.6.10 TIM16/TIM17 自動再ロードレジスタ (TIMx\_ARR)

アドレスオフセット : 0x2C

リセット値 : 0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **ARR[15:0]** : プリスケーラ値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 23.4.1 : タイムベースユニット \(683 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

## 23.6.11 TIM16/TIM17 繰り返しカウンタレジスタ (TIMx\_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]** : 繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP\_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U\_RC のみ、REP\_CNT に REP 値がロードされるので、TIMx\_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) はエッジアラインモードで PWM 周期の数を意味します。

## 23.6.12 TIM16/TIM17 キャプチャ/比較レジスタ 1 (TIMx\_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]** : キャプチャ/比較 1 値

**CC1 チャンネルが出力として設定されている場合 :**

CCR1 は、実際のキャプチャ/比較 1 レジスタにロードされる値（プリロード値）です。

TIMx\_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ/比較 1 レジスタにコピーされます。

アクティブキャプチャ/比較レジスタは、カウンタ TIMx\_CNT と比較されて、OC1 出力に送信される値を含みます。

**CC1 チャンネルが入力として設定されている場合 :**

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

## 23.6.13 TIM16/TIM17 ブレークおよびデッドタイムレジスタ (TIMx\_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

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

**注 :** AOE、BKP、BKE、OSSI、OSSR、およびDTG[7:0] ビットは、LOCK 設定に応じて書き込みがロックされるので、TIMx\_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 MOE : メイン出カインーブル

このビットは、ブレーク入力アクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : OC および OCN 出力が無効か、OSSI ビットによって強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのインーブルビット (TIMx\_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN インーブルの説明を参照してください (セクション 23.5.8 : TIM15 のキャプチャ/比較有効レジスタ (TIM15\_CCER) (726 ページ))。

ビット 14 AOE : 自動出カインーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力アクティブでない場合)。

**注 :** このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 BKP : ブレーク極性

0 : ブレーク入力 BRK はアクティブローです。

1 : ブレーク入力 BRK はアクティブハイです。

**注 :** 1.このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx\_BDTR レジスタの LOCK ビット)、変更できません。

2.このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 BKE : ブレークインーブル

0 : ブレーク入力 (BRK および CCS クロック障害イベント) は無効です。

1 : ブレーク入力 (BRK および CCS クロック障害イベント) は有効です。

**注 :** 1.このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)。

2.このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 **OSSR** : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 23.5.8 : TIM15 のキャプチャ/比較有効レジスタ \(TIM15\\_CCER\) \(726 ページ\)](#))。

0 : インアクティブのとき、OC/OCN 出力は無効です (タイマは出力の制御を解除し、ハイインピーダンス状態を強制する AFIO ロジックによって引き継がれます)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります (出力は引き続きタイマで制御される)。

**注 :** このビットは、**LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)**。

ビット 10 **OSSI** : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 23.5.8 : TIM15 のキャプチャ/比較有効レジスタ \(TIM15\\_CCER\) \(726 ページ\)](#))。

0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、まず強制的にアイドルレベルになります (OC/OCN イネーブル出力信号 = 1)。

**注 :** このビットは、**LOCK レベル 2 がプログラムされているときには変更できません (TIMx\_BDTR レジスタの LOCK ビット)**。

ビット 9:8 **LOCK[1:0]** : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 = TIMx\_BDTR レジスタの DTG ビット、TIMx\_CR2 レジスタの OISx および OISxN ビット、および TIMx\_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込めなくなります。

11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx\_CCMR レジスタの OCxM および OCxPE ビット) が書き込めなくなります。

**注 :** **LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx\_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。**

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ 、ここで  $t_{dtg}=t_{DTS}$

$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ 、ここで  $T_{dtg}=2 \times t_{DTS}$

$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで  $T_{dtg}=8 \times t_{DTS}$

$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで  $T_{dtg}=16 \times t_{DTS}$

例 :  $T_{DTS}=125\text{ns}$  (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16  $\mu\text{s}$  から 31750 ns (250 ns 単位)

32  $\mu\text{s}$  から 63  $\mu\text{s}$  (1  $\mu\text{s}$  単位)

64  $\mu\text{s}$  から 126  $\mu\text{s}$  (2  $\mu\text{s}$  単位)

**注 :** このビットフィールドは、**LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx\_BDTR レジスタの LOCK ビット)**。

## 23.6.14 TIM16/TIM17 DMA 制御レジスタ (TIMx\_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA パースト長

この 5 ビットのフィールドは、DMA 転送長 (タイマは、TIMx\_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにパースト転送を認識します)、つまり転送回数を指定します。転送は、ハーフワードまたはバイトです (以下の例を参照)。

00000 : 1 回転送  
 00001 : 2 回転送、  
 00010 : 3 回転送、  
 ...  
 10001 : 18 回転送。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのフィールドは、DMA 転送のベースアドレスを指定します (TIMx\_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx\_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :  
 00000 : TIMx\_CR1、  
 00001 : TIMx\_CR2、  
 00010 : TIMx\_SMCR  
 ...

例 : 次の転送を考えます : DBL = 7 転送 かつ DBA = TIMx\_CR1。この場合、転送は、TIMx\_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

## 23.6.15 完全転送の TIM16/TIM17 DMA アドレス (TIMx\_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA パーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(TIMx\_CR1 \text{ アドレス}) + (DBA + \text{DMA インデックス}) \times 4$$

ここで、TIMx\_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx\_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx\_DCR 内で設定)。



## 23.6.16 TIM16 オプションレジスタ (TIM16\_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1RMP	
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TI1\_RMP** : タイマ 16 入力 1 接続

このビットは、ソフトウェアによってセット/クリアされます。

00 : TIM16 の TI1 は、GPIO に接続されます。

01 : TIM16 の TI1 は、RTC\_clock に接続されます。

10 : TIM16 の TI1 は、HSE/32 に接続されます。

11 : TIM16 の TI1 は、MCO に接続されます。



## 23.6.17 TIM16/TIM17 レジスタマップ

TIM16/TIM17 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 132. TIM16/TIM17レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	Res.	CKD [1:0]	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN			
	リセット値																						0			0	0				0	0	0	0		
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC	
	リセット値																								0	0						0	0		0	
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE	
	リセット値																			0	0					0	0	0		0				0	0	
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			
0x18	TIMx_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			
	TIMx_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
リセット値																																				
0x20	TIMx_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			
0x24	TIMx_CNT	UIFCPY または Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0																																		
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																			



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

表 132. TIM16/TIM17レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																		
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x30	TIMx_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]										
	リセット値																									0	0	0	0	0	0	0	0	0		
0x34	TIMx_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x44	TIMx_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOK [1:0]	DT[7:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x48	TIMx_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]										
	リセット値																					0	0	0	0	0				0	0	0	0	0		
0x4C	TIMx_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]																		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x50	TIM16_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T11_RMP [1:0]
	リセット値																																			0

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。

## 24 赤外線インタフェース (IRTIM)

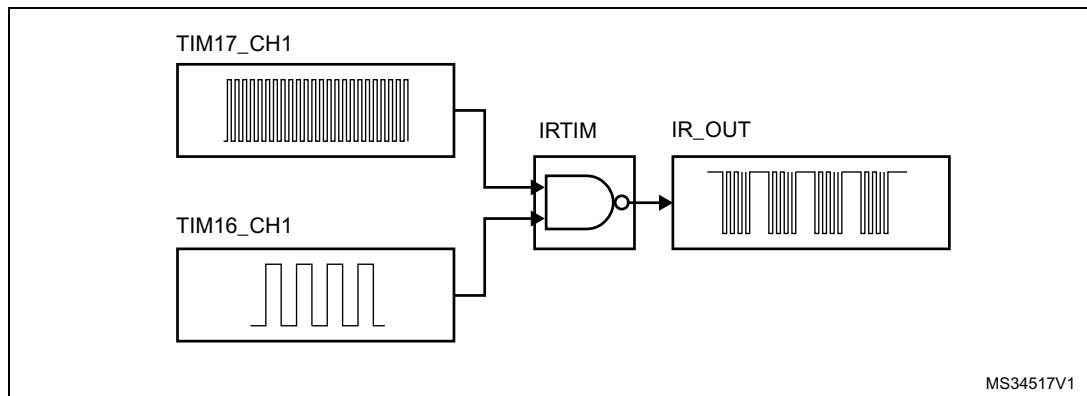
このデバイスではリモートコントロールで赤外線インタフェース (IRTIM) を使用できます。リモートコントロール機能を実行するために、赤外線 LED と併せて使用できます。

図 287 に示すように、TIM16 との間で内部接続を使用します。

赤外線リモートコントロール信号を生成するには、正しい波形を生成するために IR インタフェースを有効にし、TIM16 チャンネル 1 (TIM16\_OC1) を適切に設定する必要があります。

赤外線レシーバは、基本的な入力キャプチャモードで簡単に実装できます。

図 287. TIM16 との IR 内部ハードウェア接続



標準の IR パルス変調モードはすべて 2 つのタイマ出力比較チャンネルをプログラミングすることで取得できます。

TIM16 は包絡線変調を生成し、は高周波キャリア信号を生成するために使用されます。

赤外線機能は IR\_OUT ピンから出力されます。この機能の有効化は、GPIOx\_AFRx レジスタを通じ、関連するオルタネート機能ビットを有効にすることで行われます。

大電流シンク LED ドライバ機能 (PB9 ピンでのみ使用可能) の有効化は SYSCFG\_CFGR1 レジスタの I2C\_PB9\_FMP ビットを通じて行われ、赤外線 LED を直接制御するために必要な大電流を流すために使用されます。

## 25 独立型ウォッチドッグ (IWDG)

### 25.1 概要

デバイスは、内蔵ウォッチドッグペリフェラルを搭載しており、使用上、高い安全レベル、タイミングの正確さ、および柔軟性を兼ね備えています。独立型ウォッチドッグペリフェラルは、ソフトウェア障害による誤動作を検出および解決し、カウンタが所定のタイムアウト値に達すると、システムリセットをトリガします。

独立型ウォッチドッグ (IWDG) は、独自の低速クロック (LSI) によってクロック供給されるので、メインクロックに障害があってもアクティブなままです。

IWDG は、メインアプリケーションの外部で、完全に独立したプロセスとして実行するウォッチドッグが必要な場合に最適ですが、タイミング精度が低いという制約があります。ウィンドウ型ウォッチドッグの詳細については、[セクション 26 \(765 ページ\)](#) を参照してください。

### 25.2 IWDG の主な機能

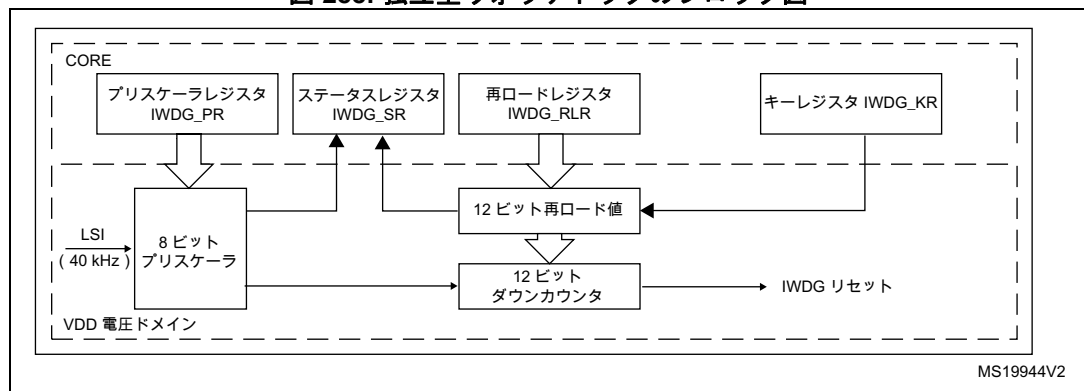
- フリーランニングダウンカウンタ
- 独立した RC オシレータからのクロック供給 (STANDBY および STOP モードで動作可能)
- 条件付きリセット
  - ダウンカウンタの値が 0x000 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
  - ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合)。

### 25.3 IWDG の機能説明

#### 25.3.1 IWDG ブロック図

図 288 に、独立型ウォッチドッグモジュールの機能ブロックを示します。

図 288. 独立型ウォッチドッグのブロック図



**注：** ウォッチドッグ機能は、STOP モードおよび STANDBY モードでも機能する CORE 電圧ドメインに実装されています。

キーレジスタ (IWDG\_KR) に値 0x0000 CCCC が書き込まれることによって独立型ウォッチドッグが開始すると、カウンタはリセット値 0xFFFF からカウントダウンを開始します。カウント値の終わり (0x000) に達すると、リセット信号が生成されます (IWDG\_reset)。

IWDG\_KR レジスタにキー値 0x0000 AAAA が書き込まれると、IWDG\_RLR の値がカウンタに再ロードされ、ウォッチドッグのリセットが防止されます。

## 25.3.2 ウィンドウオプション

IWDG は、IWDG\_WINR レジスタに適切なウィンドウをセットすることによって、ウィンドウ型ウォッチドッグとしても機能します。

カウンタがウィンドウレジスタ (IWDG\_WINR) に格納された値より大きい間に再ロード操作が行われると、リセットが生成されます。

IWDG\_WINR のデフォルト値は 0x0000 0FFF です。この値が更新されない場合は、ウィンドウオプションは無効にされます。

ウィンドウ値が変わるとすぐに再ロード操作が行われ、ダウンカウンタを IWDG\_RLR 値にリセットし、次の再ロードを生成するためのサイクル数計算を容易にします。

### ウィンドウオプションが有効な場合の IWDG の設定

1. IWDG\_KR レジスタに 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. IWDG\_KR レジスタに 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. IWDG\_PR を 0 から 7 までプログラムすることによって、IWDG のプリスケアラに書き込みを行います。
4. 再ロードレジスタ (IWDG\_RLR) に書き込みます。
5. レジスタが更新されるのを待ちます (IWDG\_SR = 0x0000 0000)。
6. ウィンドウレジスタ IWDG\_WINR に書き込みます。これにより、カウンタ値 IWDG\_RLR が自動的にリフレッシュされます。

**注：** ウィンドウ値を書き込むことで、IWDG\_SR が“0x0000 0000”にセットされた時点でカウンタ値を RLR でリフレッシュすることができます。

### ウィンドウオプションが無効な場合の IWDG の設定

ウィンドウオプションが使用されていない場合、IWDG は以下のように設定することができます。

1. IWDG\_KR レジスタに 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. IWDG\_KR レジスタに 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. IWDG\_PR を 0 から 7 までプログラムすることによって、IWDG のプリスケアラに書き込みを行います。
4. 再ロードレジスタ (IWDG\_RLR) に書き込みます。
5. レジスタが更新されるのを待ちます (IWDG\_SR = 0x0000 0000)。
6. カウンタ値を IWDG\_RLR (IWDG\_KR = 0x0000 AAAA) でリフレッシュします。

## 25.3.3 ハードウェアウォッチドッグ

デバイスのオプションビットを使って「ハードウェアウォッチドッグ」機能が有効化されると、ウォッチドッグは電源投入時に自動的に有効になり、カウンタがカウントの終わりに達する前にソフトウェアによってキーレジスタへ書き込まれない限り、またはダウンカウンタがウィンドウ内に再ロードされた場合は、リセットを生成します。

## 25.3.4 STOP および STANDBY モードでの動作

一度起動すると、IWDG は停止できません。

## 25.3.5 レジスタのアクセス保護

IWDG\_PR、IWDG\_RLR、および IWDG\_WINR レジスタへの書き込みアクセスは保護されます。これらを変更するには、まず、IWDG\_KR レジスタにコード 0x0000 5555 を書き込む必要があります。このレジスタに別の値を書き込むと、シーケンスがブレイクされ、レジスタへのアクセスが再び保護されます。これは、再ロード操作 (0x0000 AAAA の書き込み) であることを意味します。ステータスレジスタは、プリスケアラの更新、あるいはダウンカウンタ再ロード値やウィンドウ値の更新が行われていることを示すために使用されます。

## 25.3.6 デバッグモード

マイクロコントローラがデバッグモードになると (コアは停止状態)、IWDG カウンタは、DBG モジュールの DBG\_IWDG\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 33.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#) を参照してください。

## 25.4 IWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 25.4.1 キーレジスタ (IWDG\_KR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされる)

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **KEY[15:0]** : キー値 (書き込み専用、読み出しでは 0x0000)

これらのビットには、ソフトウェアによって一定間隔でキー値 0xAAAA が書き込まれなければなりません。そうしないと、カウンタが 0 に達した時点でウォッチドッグがリセットを生成します。

キー値 0x5555 を書き込むことによって、IWDG\_PR、IWDG\_RLR、および IWDG\_WINR レジスタへのアクセスが可能になります ([セクション 25.3.5 : レジスタのアクセス保護](#)を参照)。

キー値 CCCCh を書き込むと、ウォッチドッグが開始します (ハードウェアウォッチドッグオプションが選択されている場合を除く)。

## 25.4.2 プリスケーラレジスタ (IWDG\_PR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **PR[2:0]** : プリスケーラ分周回路

これらのビットは、書き込みアクセス保護されています ([セクション 25.3.5: レジスタのアクセス保護](#)を参照)。カウンタクロックを供給するプリスケーラ分周回路を選択するようにソフトウェアで書き込まれます。プリスケーラ分周回路を変更できるようにするには、IWDG\_SR レジスタの PVU ビットをリセットする必要があります。

- 000 : 4 分周
- 001 : 8 分周
- 010 : 16 分周
- 011 : 32 分周
- 100 : 64 分周
- 101 : 128 分周
- 110 : 256 分周
- 111 : 256 分周

**注 :** このレジスタを読み出すと、VDD 電圧ドメインからプリスケーラ値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG\_SR レジスタの PVU ビットがリセットされているときのみとなります。



## 25.4.3 再ロードレジスタ (IWDG\_RLR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **RL[11:0]** : ウォッチドッグカウンタ再ロード値

これらのビットは、書き込みアクセス保護されています ([セクション 25.3.5](#)を参照)。IWDG\_KR レジスタに値 0xAAAA が書き込まれるたびにウォッチドッグカウンタにロードされる値を定義するために、ソフトウェアで書き込まれます。ウォッチドッグカウンタは、この値からカウントダウンします。タイムアウトまでの時間は、この値とクロックプリスケアラによって決まります。タイムアウトに関する詳細はデータシートを参照してください。

再ロード値を変更できるようにするには、IWDG\_SR レジスタの RVU ビットをリセットする必要があります。

**注 :** このレジスタを読み出すと、VDD 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG\_SR レジスタの RVU ビットがリセットされているときのみとなります。

## 25.4.4 ステータスレジスタ (IWDG\_SR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされません)

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

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **WVU** : ウォッチドッグカウンタウィンドウ値の更新

このビットは、ウィンドウ値の更新が進行中であることを示すために、ハードウェアによってセットされます。V<sub>DD</sub> 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

ウィンドウ値は、WVU ビットがリセットされているときのみ更新できます。

このビットは、一般のウィンドウが 1 の場合のみ生成されます。

ビット 1 **RVU** : ウォッチドッグカウンタ再ロード値の更新

このビットは、再ロード値の更新が進行中であることを示すために、ハードウェアによってセットされます。V<sub>DD</sub> 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

再ロード値は、RVU ビットがリセットされているときのみ更新できます。

ビット 0 **PVU** : ウォッチドッグプリスケアラ値の更新

このビットは、プリスケアラ値の更新が進行中であることを示すために、ハードウェアによってセットされます。V<sub>DD</sub> 電圧ドメインでプリスケアラ更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

プリスケアラ値は、PVU ビットがリセットされているときのみ更新できます。

**注 :** *複数の再ロード値、プリスケアラ値、またはウィンドウ値がアプリケーションで使用される場合は、それぞれ、再ロード値を変更する前に RVU ビットがリセットされるまで待つか、プリスケアラ値を変更する前に PVU ビットがリセットされるまで待つか、またはウィンドウ値を変更する前に WVU ビットがリセットされるまで待つ必要があります。ただし、プリスケアラ値、再ロード値、またはウィンドウ値を更新した後は、RVU、PVU、または WVU がリセットされるのを待たずに、コード実行を続けることができます (低電力モードに入った場合を除く)。*

## 25.4.5 ウィンドウレジスタ (IWDG\_WINR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **WIN[11:0]** : ウォッチドッグカウンタウィンドウ値

これらのビットは、書き込みアクセス保護されています ([セクション 25.3.5](#)を参照)。これらのビットは、ダウンカウンタと比較されるウィンドウ値の上限を含みます。

リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、

0x0 よりも大きい間にダウンカウンタを再ロードする必要があります。

再ロード値を変更できるようにするには、IWDG\_SR レジスタの WVU ビットをリセットする必要があります。

**注 :** このレジスタを読み出すと、V<sub>DD</sub> 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG\_SR レジスタの WVU ビットがリセットされているときのみとなります。

## 25.4.6 IWDG レジスタマップ

次の表に、IWDG レジスタマップとリセット値を示します。

表 133. IWDG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	IWDG_KR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	IWDG_PR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR[2:0]
	リセット値																															0	0
0x08	IWDG_RLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RL[11:0]										
	リセット値																						1	1	1	1	1	1	1	1	1	1	1
0x0C	IWDG_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WVU
	リセット値																															0	0
0x10	IWDG_WINR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WIN[11:0]										
	リセット値																						1	1	1	1	1	1	1	1	1	1	1

レジスタ境界アドレスについては、[セクション 3.2.2: メモリマップとレジスタ境界アドレス](#) を参照してください。

## 26 システムウィンドウ型ウォッチドッグ (WWDG)

### 26.1 概要

システムウィンドウ型ウォッチドッグ (WWDG) は、通常、外部の影響や予期しない論理条件などによって発生し、アプリケーションプログラムを正常なシーケンスから逸脱させるソフトウェア障害の発生を検出するために使用されます。ウォッチドッグ回路は、T6 ビットがクリアされる前にプログラムがダウンカウンタの内容をリフレッシュしない限り、プログラムされた時間の経過後に MCU リセットを生成します。MCU リセットは、ダウンカウンタがウィンドウレジスタ値に達する前に 7 ビットのダウンカウンタの値 (制御レジスタ内) がリフレッシュされた場合にも生成されます。このことは、限られた時間枠 (time-window) の間にカウンタがリフレッシュされなければならないことを意味します。

WWDG クロックは、APB1 クロックから分周され、また設定可能な時間枠 (time-window) があるので、これをプログラムしてアプリケーション動作の異常な進み・遅れを検出できます。

WWDG は、正確な時間枠内で反応するウォッチドッグが必要なアプリケーションに適しています。

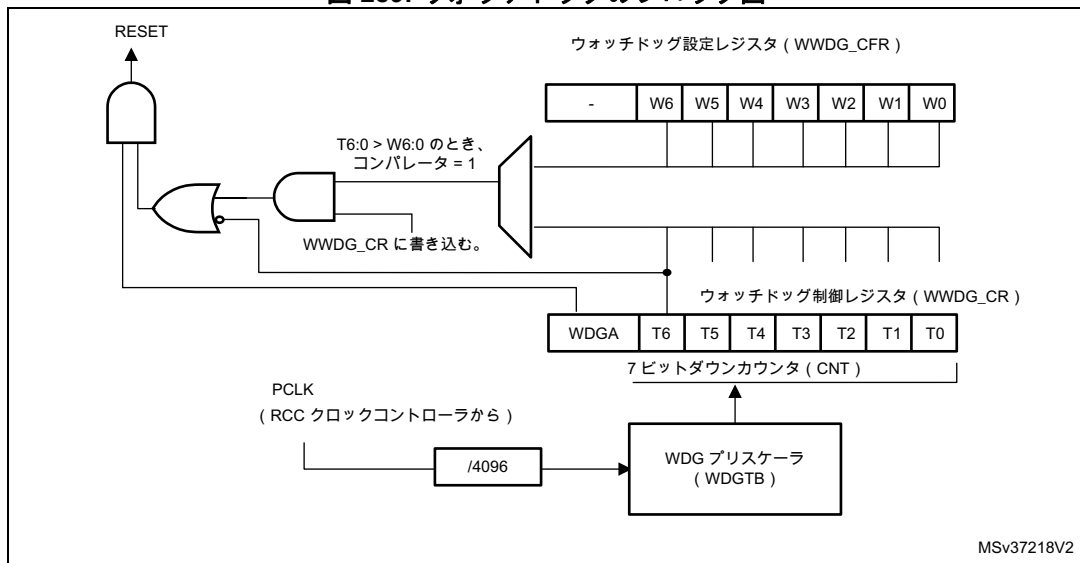
### 26.2 WWDG の主な機能

- プログラム可能なフリーランニングダウンカウンタ
- 条件付きリセット
  - ダウンカウンタの値が 0x40 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
  - ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合) ([図 290](#)を参照)。
- 早期ウェイクアップ割り込み (EWI) : ダウンカウンタが 0x40 になったときにトリガ (有効であり、ウォッチドッグがアクティブな場合)

### 26.3 WWDG の機能説明

ウォッチドッグが有効 (WWDG\_CR レジスタの WDGA ビットがセットされている) な場合、7 ビットのダウンカウンタ (T[6:0] ビット) が 0x40 に達して 0x3F にデクリメントされた (T6 がクリアされた) 時点で、リセットを開始します。カウンタがウィンドウレジスタに格納された値より大きい間にソフトウェアがカウンタを再ロードした場合にも、リセットが生成されます。

図 289. ウォッチドッグのブロック図



アプリケーションプログラムは、通常動作時には定期的に WWDG\_CR レジスタへの書き込みを行って、MCU リセットを防ぐ必要があります。この操作は、カウンタの値がウィンドウレジスタの値より小さいとき、かつ 0x3F より高いときに限られます。WWDG\_CR レジスタに格納される値は、0xFF から 0xC0 の間でなければなりません。

## 26.3.1 ウォッチドッグの有効化

ウォッチドッグはリセット後は常に無効です。これを有効にするには、WWDG\_CR レジスタの WDGA ビットをセットします。この後は、リセット以外の方法でウォッチドッグを無効にすることはできません。

## 26.3.2 ダウンカウンタの制御

このダウンカウンタはフリーランニングであり、ウォッチドッグが無効状態であってもカウントダウンを続けます。ウォッチドッグを有効にするときには、T6 ビットをセットして、ただちにリセットが生成されるのを防ぐ必要があります。

T[5:0] ビットは、ウォッチドッグがリセットを生成するまでの時間遅延を表すインクリメント数を含みます。このタイミングは、WWDG\_CR レジスタへの書き込み時のプリスケアラの状態が不明なので最小値から最大値の間で変化します (図 290 参照)。設定レジスタ (WWDG\_CFR) は、ウィンドウの上限值を含みます。リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、0x3F よりも大きい間にダウンカウンタを再ロードする必要があります。図 290 に、ウィンドウ型ウォッチドッグのプロセスを示します。

**注:** T6 ビットを使用して、ソフトウェアリセットを生成することができます (WDGA ビットはセット、T6 ビットはクリアされる)。

## 26.3.3 高度なウォッチドッグ割り込み機能

実際にリセットが生成される前に特定の安全処理やデータロギングを実施する必要がある場合は、早期ウェイクアップ割り込み (EWI) が使用できます。EWI 割り込みは、WWDG\_CFR レジスタの EWI ビットをセットすることによって有効になります。ダウンカウンタ値が 0x40 に到達すると、EWI 割り込みが生成され、対応する割り込みサービスルーチン (ISR) を使用してデバイスをリセットする前に特定の処理 (通信やデータロギングなど) をトリガすることができます。

アプリケーションによっては、EWI 割り込みを使用して、WWDG リセットを生成せずにソフトウェアのシステムチェックやシステム復旧/グレースフルデグラデーションを管理することができます。この場合、対応する割り込みサービスルーチン (ISR) で WWDG カウンタを再ロードし、WWDG リセットを回避してから必要な操作をトリガしてください。

EWI 割り込みは、WWDG\_SR レジスタの EWIF ビットに“0”を書き込むことによってクリアされません。

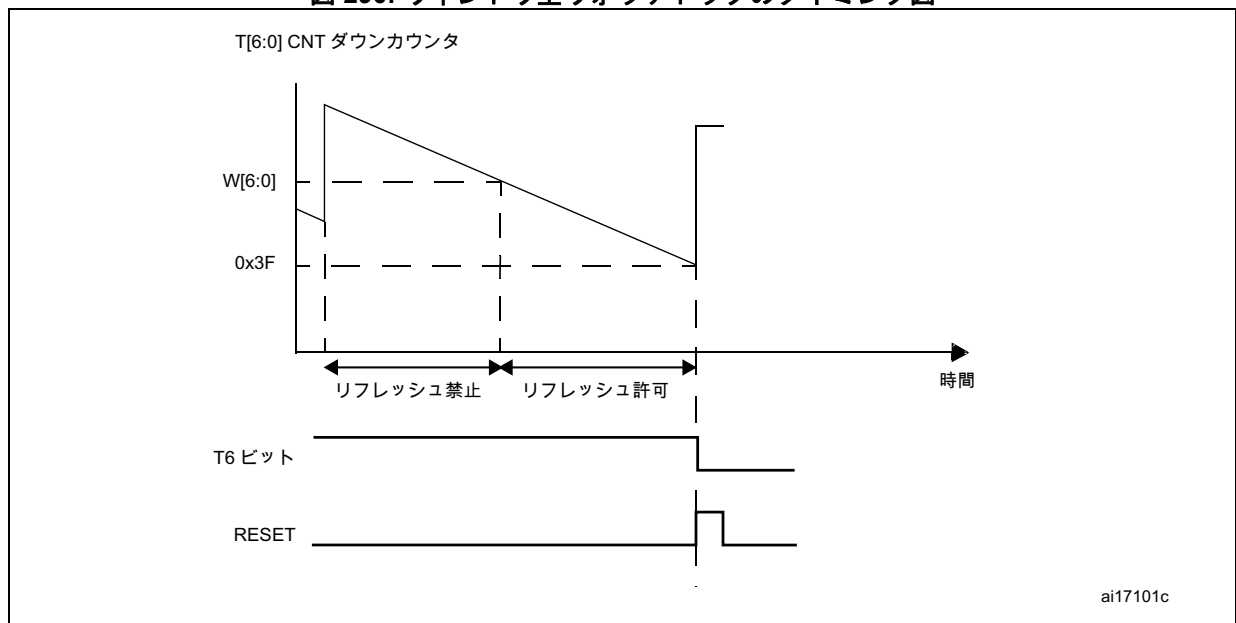
**注：** 優先順位の高いタスクにおけるシステムロックなどによって EWI 割り込みが使用できない場合、最終的には WWDG リセットが生成されます。

## 26.3.4 ウォッチドッグタイムアウトをプログラムする方法

図 290 の式を使用して、WWDG のタイムアウトを計算することができます。

**警告：** WWDG\_CR レジスタに書き込むときには、ただちにリセットされるのを防ぐために、常に T6 ビットに 1 を書き込んでください。

図 290. ウィンドウ型ウォッチドッグのタイミング図



タイムアウト値は次の式で算出されます。

$$t_{\text{WWDG}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{WDGTB}[1:0]} \times (\text{T}[5:0] + 1) \quad (\text{ms})$$

ここで、

$t_{\text{WWDG}}$  : WWDG タイムアウト

$t_{\text{PCLK1}}$  : APB1 クロック周期の測定値 (ms)

4096 : 内部分周器に対応する値

たとえば、APB1 周波数が 48 MHz と等しい場合、WDGTB[1:0] は 3 にセットされ、T[5:0] は 63 にセットされます。

$$t_{\text{WWDG}} = 1/48000 \times 4096 \times 2^3 \times (63 + 1) = 43.69 \text{ ms}$$

$t_{\text{WWDG}}$  の最小値と最大値については、データシートを参照してください。

## 26.3.5 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex-M4<sup>®</sup>Fコアは停止状態)、WWDG カウンタは、DBG モジュールの DBG\_WWDG\_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 33.16.2 : タイマ、ウォッチドッグ、bxCAN、および I2C のデバッグサポート](#)を参照してください。



## 26.4 WWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 26.4.1 制御レジスタ (WWDG\_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 007F

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDGA** : 有効化ビット

このビットは、ソフトウェアでセットされ、リセット後はハードウェアによってのみクリアされます。WDGA = 1 のとき、ウォッチドッグはリセットを生成できます。

- 0 : ウォッチドッグは無効です。
- 1 : ウォッチドッグは有効です。

ビット 6:0 **T[6:0]** : 7 ビットカウンタ (MSB から LSB まで)

これらのビットは、ウォッチドッグカウンタの値を含みます。(4096 x 2<sup>WDGTB[1:0]</sup>) PCLK サイクルごとにデクリメントされます。0x40 に達して 0x3F にデクリメントされると (T6 がクリアされると)、リセットが生成されます。

## 26.4.2 設定レジスタ (WWDG\_CFR)

アドレスオフセット : 0x04

リセット値 : 0x0000 007F

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

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **EWI** : 早期ウェイクアップ割り込み

このビットがセットされているときには、カウンタの値が 0x40 に達したときに割り込みが発生します。この割り込みは、リセット後にハードウェアによってのみクリアされます。

ビット 8:7 **WDGTB[1:0]** : タイマーベース

プリスケアラのタイムベースは、次のように変更できます。

00 : CK カウンタクロック (PCLK/4096) 1 分周

01 : CK カウンタクロック (PCLK/4096) 2 分周

10 : CK カウンタクロック (PCLK/4096) 4 分周

11 : CK カウンタクロック (PCLK/4096) 8 分周

ビット 6:0 **W[6:0]** : 7 ビットウィンドウ値

これらのビットは、ダウンカウンタと比較されるウィンドウ値を含みます。

## 26.4.3 ステータスレジスタ (WWDG\_SR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EWIF** : 早期ウェイクアップ割り込みフラグ

このビットは、カウンタの値が 0x40 に達したときにハードウェアによってセットされます。“0”を書き込んでソフトウェアでクリアする必要があります。“1”を書き込んでも、ビットの値は変化しません。このビットは、割り込みが有効でない場合にもセットされます。

## 26.4.4 WWDG レジスタマップ

次の表に、WWDG のレジスタマップとリセット値を示します。

表 134. WWDG レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	WWDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]							
	リセット値																									0	1	1	1	1	1	1	1	
0x04	WWDG_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB1	W[6:0]								
	リセット値																							0	0	0	0	0	1	1	1	1	1	1
0x08	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF		
	リセット値																																	0

レジスタ境界アドレスについては、[セクション 3.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。



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

## 27 リアルタイムクロック (RTC)

### 27.1 概要

RTC は、あらゆる低電力モードを管理する自動ウェイクアップ機能を提供します。

本リアルタイムクロック (RTC) は、独立した BCD タイマ/カウンタです。RTC は、プログラム可能なアラーム割り込み機能を備えた時刻クロック/カレンダーを搭載しています。

また、割り込み機能を備えたプログラム可能な周期的ウェイクアップフラグも搭載しています。

2 つの 32 ビットレジスタには、2 進化 10 進数形式 (BCD) で表現した秒、分、時 (12 時間または 24 時間形式)、曜日、日、月、年が含まれています。サブセカンドの値もバイナリ形式で利用できます。

28 日、29 日 (うるう年)、30 日、31 日の補正は、自動的に行われます。サマータイム補正も行われます。

サブセカンド、秒、分、時、曜日、日付のプログラム可能なアラームを備えた 32 ビットレジスタが追加されています。

クリスタルオシレータ精度の偏差を補正するために、デジタル較正機能が利用可能です。

バックアップドメインリセット後、すべての RTC レジスタは、起こりうる不要な書き込みアクセスから保護されます。

供給電圧が動作範囲内にある間は、デバイスのステータス (実行モード、低電力モード、またはリセット中) に関係なく、RTC が停止することはありません。

## 27.2 RTC の主な機能

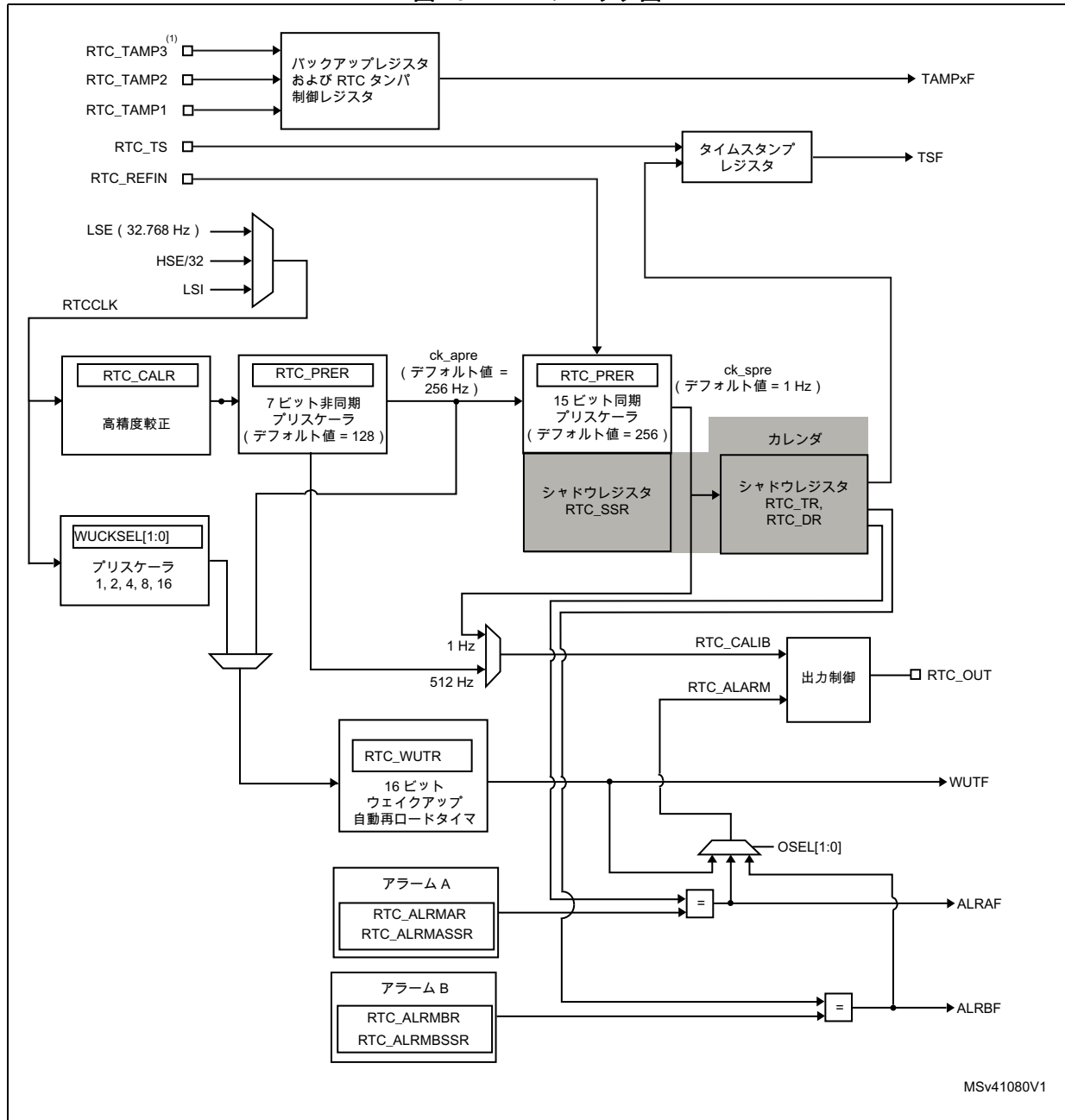
RTC ユニットの主な機能を以下に示します (図 291 : RTC ブロック図を参照)。

- サブセカンド、秒、分、時 (12 または 24 時間形式)、曜日、日、月、年に対応するカレンダー。
- ソフトウェアでプログラム可能なサマータイム補正。
- 割り込み機能を備えたプログラム可能なアラーム。アラームは、カレンダー項目のどの組み合わせでもトリガ可能。
- 自動ウェイクアップ割り込みをトリガする周期的なフラグを生成する自動ウェイクアップユニット。
- リファレンスクロック検出 : より正確な秒のクロックソース (50 または 60 Hz) の使用で、カレンダーの精度を向上。
- サブセカンドシフト機能を使用する外部クロックとの正確な同期。
- デジタル較正回路 (周期的なカウンタ修正) : 数秒の較正時間範囲で得られる 0.95 ppm の精度。
- イベントを保存するタイムスタンプ機能。
- 設定可能なフィルタおよび内部プルアップのあるタンパ検出イベント。
- マスク可能な割り込み/イベント :
  - アラーム A
  - アラーム B
  - ウェイクアップ割り込み
  - タイムスタンプ
  - タンパ検出
- 16 バックアップレジスタ。

## 27.3 RTC の機能説明

### 27.3.1 RTC ブロック図

図 291. RTC ブロック図



1. RTC\_TAMP3 は STM32F303x6/8 および STM32F328 デバイスでは使用できません。

RTC には次の要素が含まれます。

- 2本のアラーム
- I/O からの3つのタンパイベント
  - タンパ検出でバックアップレジスタは消去されます。
- I/O からの1つのタイムスタンプイベント
- タンパイベント検出によってタイムスタンプイベントを生成できます。
- 16 x 32 ビットバックアップレジスタ (STM32F303xB/C) および 5 x 32 ビットバックアップレジスタ (STM32F303x6/8)
  - VDD 電源が遮断された場合にVBATによって電源が供給される RTC ドメインに、バックアップレジスタ (RTC\_BKPxR) は搭載されています。
- オルタネート機能出力：次の2つの出力のうち1つを選択する RTC\_OUTです。
  - RTC\_CALIB：512 Hz または 1Hz のクロック出力 (LSE 周波数 32.768 kHz の場合)。この出力は、RTC\_CR レジスタの COE ビットをセットして有効にします。
  - RTC\_ALARM：この出力は、アラーム A、アラーム B、またはウェイクアップ出力を選択する RTC\_CR レジスタの OSEL[1:0] ビットを設定することで有効にします。
- オルタネート機能入力：
  - RTC\_TS：タイムスタンプイベント
  - RTC\_TAMP1：タンパ1 イベント検出
  - RTC\_TAMP2：タンパ2 イベント検出
  - RTC\_TAMP3：タンパ3 イベント検出
  - RTC\_REFIN：50 または 60 Hz のリファレンスクロック入力

## 27.3.2 RTC によって制御される GPIO

RTC\_OUT、RTC\_TS、および RTC\_TAMP1 は、同一ピン (PC13) 上に配置されます。

RTC\_ALARM 出力の選択は次のように RTC\_TAFPCR レジスタを通して行います。PC13VALUE ビットを使用して、RTC\_ALARM 出力をプッシュプルモードとオープンドレインモードのどちらで設定するかを選択します。

PC13 が RTC オルタネート機能として使用されない場合、RTC\_TAFPCR の PC13MODE ビットを設定することで、出力を強制的にプッシュプルモードに設定できます。これにより、PC13VALUE ビットから出力データ値が与えられます。この場合、PC13 の出力のプッシュプル状態とデータ値は STANDBY モードでも維持されます。

出力形式は [表 135](#) に示す優先順位に従います。

PC14 と PC15 が LSE オシレータとして使用されない場合、RTC\_TAFPCR レジスタの PC14MODE ビットと PC15MODE ビットをそれぞれ設定することで、出力を強制的にプッシュプルモードに設定できます。これにより、PC14VALUE と PC15VALUE から出力データ値が与えられます。この場合、PC14 および PC15 の出力のプッシュプル状態とデータ値は STANDBY モードでも維持されます。

出力形式は [表 136](#) および [表 137](#) に示す優先順位に従います。

表 135. RTC ピン PC13 の設定<sup>(1)</sup>

ピンの設定と機能	RTC_ALARM 出力有効	RTC_CALIB 出力有効	RTC_TAMP1 入力有効	RTC_TS 入力有効	PC13MODE ビット	PC13VALUE ビット
RTC_ALARM 出力 OD	1	無視	無視	無視	無視	0
RTC_ALARM 出力 PP	1	無視	無視	無視	無視	1
RTC_CALIB 出力 PP	0	1	無視	無視	無視	無視
RTC_TAMP1 入力フロー テイング	0	0	1	0	無視	無視
RTC_TS および RTC_TAMP1 入力フロー テイング	0	0	1	1	無視	無視
RTC_TS 入力 フローテイング	0	0	0	1	無視	無視
出力 PP に固定	0	0	0	0	1	PC13 出力 データ値
ウェイクアップ ピンまたは標準 GPIO	0	0	0	0	0	無視

1. OD : オープンドレイン、PP : プッシュプル

表 136. LSE ピン PC14 の設定<sup>(1)</sup>

ピンの設定と機能	RCC_BDCR レジスタの LSEON ビット	RCC_BDCR レジスタの LSEBYP ビット	PC14MODE ビット	PC14VALUE ビット
LSE オシレータ	1	0	無視	無視
LSE バイパス	1	1	無視	無視
出力 PP に固定	0	無視	1	PC14 出力データ値
標準 GPIO	0	無視	0	無視

1. OD : オープンドレイン、PP : プッシュプル

表 137. LSE ピン PC15 の設定<sup>(1)</sup>

ピンの設定と機能	RCC_BDCR レジスタの LSEON ビット	RCC_BDCR レジスタの LSEBYP ビット	PC15MODE ビット	PC15VALUE ビット
LSE オシレータ	1	0	無視	無視
出力 PP に固定	1	1	1	PC15 出力データ値
	0	無視		
標準 GPIO	0	無視	0	無視

1. OD : オープンドレイン、PP : プッシュプル



## 27.3.3 クロックとプリスケアラ

RTC クロックソース (RTCCLK) は、LSE クロック、LSI オシレータクロック、HSE クロックのうちから、クロックコントローラを介して選択されます。RTC クロックソースの設定に関する詳細は、[セクション 9 : リセットおよびクロック制御 \(RCC\)](#) を参照してください。

プログラム可能なプリスケアラステージで、カレンダーの更新に使用する 1 Hz のクロックを生成します。消費電力を最少に抑えるため、プリスケアラは以下に示す 2 つのプログラム可能なプリスケアラに分割されます ([図 291 : RTC ブロック図](#)を参照)。

- RTC\_PRER レジスタの PREDIV\_A ビットで設定される 7 ビットの非同期プリスケアラ
- RTC\_PRER レジスタの PREDIV\_S ビットで設定される 15 ビットの同期プリスケアラ

**注 :** **両方のプリスケアラを使用する場合は、非同期プリスケアラを高い値に設定して消費を最低限に抑えることをお勧めします。**

LSE 周波数 32.768 kHz で 1 Hz (ck\_spre) の内部クロック周波数を得るため、非同期プリスケアラ分周比は 128、同期プリスケアラの分周比は 256 に設定されます。

最低分周比は 1、最大分周比は  $2^{22}$  です。

これは、約 4 MHz の最大入力周波数に相当します。

$f_{ck\_apre}$  は、次の式で与えられます。

$$f_{CK\_APRE} = \frac{f_{RTCCLK}}{PREDIV\_A + 1}$$

ck\_apre クロックは、サブセカンドダウンカウンタであるバイナリ RTC\_SSR にクロックを供給するために使用されます。値がゼロになると、RTC\_SSR は、PREDIV\_S の内容で再ロードされます。

$f_{ck\_apre}$  は、次の式で与えられます。

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

ck\_spre クロックは、カレンダーの更新に、または 16 ビットウェイクアップ自動再ロードタイマのタイムベースとして使用できます。短いタイムアウト期間を得るため、16 ビットウェイクアップ自動再ロードタイマを、プログラム可能な 4 ビット非同期プリスケアラで分周した RTCCLK で動作させることもできます (詳細は [セクション 27.3.6 : 周期的自動ウェイクアップ](#) を参照)。

## 27.3.4 リアルタイムクロックとカレンダー

RTC カレンダーの時刻および日付レジスタには、PCLK (APB クロック) と同期するシャドウレジスタからアクセスします。同期持続の待ち時間を避けるため、これらのレジスタに直接アクセスすることもできます。

- サブセカンド用 RTC\_SSR
- 時刻用 RTC\_TR
- 日付用 RTC\_DR

RTCCLK 2 サイクルごとに現在のカレンダー値がシャドウレジスタにコピーされ、RTC\_ISR レジスタの RSF ビットがセットされます ([セクション 27.6.4 : RTC 初期化とステータスレジスタ \(RTC\\_ISR\)](#) を参照)。STOP モードおよび STANDBY モードでは、コピーは行われません。これらのモードが終了すると、RTCCLK 2 サイクル以内にシャドウレジスタが更新されます。

アプリケーションが、カレンダーレジスタを読み出す際、実際にはシャドウレジスタの内容にアクセスします。RTC\_CR レジスタの BYPSHAD 制御ビットをセットすることにより、カレンダーレジスタに直接アクセスできます。デフォルトでは、このビットはクリアされており、ユーザはシャドウレジスタにアクセスします。

RTC\_SSR、RTC\_TR または RTC\_DR レジスタを BYPSHAD = 0 の状態で読み出す際は、APB クロックの周波数 ( $f_{APB}$ ) は、RTC クロック ( $f_{RTCCLK}$ ) の周波数の 7 倍以上でなければなりません。

シャドウレジスタは、システムリセットによってリセットされます。

## 27.3.5 プログラム可能なアラーム

RTC ユニットは、以下に示すプログラム可能なアラーム、アラーム A およびアラーム B を搭載しています。以下に示すのはアラーム A に関する説明ですが、アラーム B についても同様です。

プログラム可能なアラーム機能は、RTC\_CR レジスタの ALRAE ビットを通じて有効にします。ALRAF は、カレンダーのサブセカンド、秒、分、時、日または曜日がそれぞれアラームレジスタ RTC\_ALRMSSR および RTC\_ALRMAR にプログラムされている値と一致する場合は 1 にセットされます。各カレンダー項目は、RTC\_ALRMAR レジスタの MSKx ビットおよび RTC\_ALRMSSR レジスタの MASKSSx ビットで個別に選択できます。アラームの割り込みは、RTC\_CR レジスタの ALRAIE ビットを通じて有効にします。

**注意：** *秒の項目が選択されている (RTC\_ALRMAR で MSK1 ビットがリセットされている) 場合、正しい動作を保証するため、RTC\_PRER レジスタでセットされる同期プリスケアラの分周比は 3 以上でなければなりません。*

アラーム A および アラーム B (RTC\_CR レジスタの OSEL[0:1] ビットで有効になっている場合) は、RTC\_ALARM 出力に送ることができます。RTC\_ALARM 出力の極性は、RTC\_CR レジスタの POL ビットを通じて設定できます。

## 27.3.6 周期的自動ウェイクアップ

周期的ウェイクアップフラグは、16 ビットのプログラム可能な自動再ロードダウンカウンタによって生成されます。ウェイクアップタイマの範囲は 17 ビットまで拡張できます。

ウェイクアップ機能は、RTC\_CR レジスタの WUTE ビットを通じて有効にします。

ウェイクアップタイマクロック入力には、次のものが使用できます。

- 2、4、8、または 16 分周した RTC クロック (RTCCLK)
 

RTCCLK が LSE (32.768kHz) である場合、最小分解能 61  $\mu$ s で、ウェイクアップ割り込み周期を 122  $\mu$ s から 32 s の範囲で設定できます。
- ck\_spre (通常は 1 Hz の内部クロック)
 

ck\_spre 周波数が 1 Hz の場合、1 秒の分解能でウェイクアップ時間を 1 秒 からおよそ 36 時間までの範囲で設定できます。このプログラム可能な広い時間範囲は、2 つの部分に分かれます。

  - WUCKSEL[2:1] = 10 の場合は 1 秒から 18 時間、
  - WUCKSEL[2:1] = 11 の場合は約 18 時間から 36 時間です。後者の場合、16 ビットカウンタの現在値に 216 が加算されます。初期化シーケンスが完了すると ([ウェイクアップタイマのプログラミング \(780 ページ\)](#) を参照)、タイマがカウントダウンを開始します。ウェイクアップ機能が有効な場合、低電力モードでもカウントダウンはアクティブのままとなります。さらに、カウンタがゼロに到達すると、RTC\_ISR レジスタの WUTF フラグがセットされ、ウェイクアップカウンタが再ロード値 (RTC\_WUTR レジスタ値) で自動的に再ロードされます。

その後、WUTF フラグはソフトウェアでクリアする必要があります。

RTC\_CR2 レジスタの WUTIE ビットをセットして周期的ウェイクアップ割り込みを有効にすると、デバイスは低電力モードを終了できます。

周期的なウェイクアップフラグは、RTC\_CR レジスタの OSEL[0:1] ビットを通じて有効になっている場合に限り、RTC\_ALARM 出力に送ることができます。RTC\_ALARM 出力の極性は、RTC\_CR レジスタの POL ビットを通じて設定できます。

低電力モード (SLEEP、STOP、STANDBY) と同様に、システムリセットもウェイクアップタイムには影響しません。

## 27.3.7 RTC の初期化と設定

### RTC レジスタアクセス

RTC レジスタは、32 ビットのレジスタです。APB インターフェイスは、RTC レジスタアクセスに 2 ウェイトステートを挿入します。ただし、BYP SHAD = 0 のときのカレンダーシャドウレジスタへの読み出しアクセスは除きます。

### RTC レジスタ書き込み保護

システムリセット後、RTC レジスタは、PWR\_CR レジスタの DBP ビットをクリアすることによって、意図しない書き込みアクセスから保護されます (電源制御のセクションを参照)。RTC レジスタ書き込みアクセスを可能にするには、DBP ビットをセットする必要があります。

バックアップドメインリセット後、すべての RTC レジスタは書き込み保護されます。RTC レジスタへの書き込みは、書き込み保護レジスタ RTC\_WPR にキーを書き込むことにより有効になります。

RTC\_TAFCR、RTC\_BKPxR、および RTC\_ISR[13:8] を除くすべての RTC レジスタの書き込み保護を解除するには、次のステップが必要です。

1. RTC\_WPR レジスタに“0xCA”を書き込みます。
2. RTC\_WPR レジスタに“0x53”を書き込みます。

誤ったキーを書き込むと、書き込み保護が再度アクティブになります。

保護メカニズムは、システムリセットの影響を受けません。

### カレンダーの初期化と設定

時間形式やプリスケアラ設定を含むカレンダー時刻と日付の初期値をプログラムするには、次のシーケンスが必要です。

1. RTC\_ISR レジスタで INIT ビットを 1 にセットして、初期化モードに入ります。このモードでは、カレンダーカウンタが停止し、その値を更新することができます。
2. RTC\_ISR レジスタの INITF ビットをポーリングします。INITF が 1 にセットされると、初期化フェーズモードに入ります。これには RTCCLK クロック約 2 サイクルを必要とします (クロック同期のため)。
3. カレンダーカウンタのための 1 Hz クロックを生成するには、RTC\_PRER レジスタで両方のプリスケアラ分周比をプログラムします。
4. シャドウレジスタ (RTC\_TR および RTC\_DR) に時刻と日付の初期値をロードし、RTC\_CR レジスタの FMT ビットを介して時間形式 (12 時間または 24 時間) を設定します。
5. INIT ビットをクリアして初期化モードを終了します。その後、カレンダーカウンタの実際の値が自動的にロードされ、4 RTCCLK クロックサイクル後にカウントが再開します。

初期化シーケンスが完了すると、カレンダーがカウントを開始します。

**注：** システムリセット後、アプリケーションは RTC\_ISR レジスタの INITS フラグを読み出し、カレンダーが初期化されたか否かを確認できるようになります。このフラグが 0 であれば、カレンダーの年の項目がバックアップドメインリセットデフォルト値 (0x00) にセットされているため、初期化されていません。

初期化後にカレンダーを読み出すには、まずソフトウェアで RTC\_ISR レジスタの RSF フラグがセットされていることを確認する必要があります。

## サマータイム

サマータイム管理は、RTC\_CR レジスタの SUB1H ビット、ADD1H ビット、BKP ビットを介して行われます。

SUB1H または ADD1H を使用すると、ソフトウェアは初期化手順を踏まずに 1 度の操作で、カレンダーから 1 時間引いたり足したりすることができます。

さらに、ソフトウェアは BKP ビットを使用してこの操作を記憶することができます。

## アラームのプログラミング

プログラム可能なアラームをプログラムまたは更新するには、同様な手順を踏む必要があります。以下に示すのはアラーム A の手順ですが、アラーム B についても同様です。

1. RTC\_CR の ALRAE をクリアしてアラーム A を無効にします。
2. アラーム A レジスタ (RTC\_ALRMASR/RTC\_ALRMAR) をプログラムします。
3. RTC\_CR レジスタで ALRAE をセットしてアラーム A を再び有効にします。

**注：** RTC\_CR レジスタの各変更は、クロック同期のため RTCCLK クロック約 2 サイクル後に有効になります。

## ウェイクアップタイマのプログラミング

ウェイクアップタイマ自動再ロード値 (RTC\_WUTR の WUT[15:0]) の設定または変更には、次の手順が必要です。

1. RTC\_CR の WUTE をクリアしてウェイクアップタイマを無効にします。
2. RTC\_ISR の WUTWF がセットされ、ウェイクアップ自動再ロードカウンタおよび WUCKSEL[2:0] ビットへのアクセスが許可されていることが確認されるまで WUTWF をポーリングします。これには RTCCLK クロック約 2 サイクルを必要とします (クロック同期のため)。
3. ウェイクアップ自動再ロード値 WUT[15:0] およびウェイクアップクロック選択 (RTC\_CR の WUCKSEL[2:0] ビット) をプログラムします。RTC\_CR で WUTE をセットしてタイマを再び有効にします。ウェイクアップタイマがカウントダウンを再開します。WUTWF ビットは、クロックの同期化により、WUTE クリア後、2 RTCCLK クロックサイクルまでクリアされます。

## 27.3.8 カレンダーの読み出し

### RTC\_CR レジスタの BYPSHAD 制御ビットがクリアされている場合

RTC カレンダーレジスタ (RTC\_SSR、RTC\_TR、および RTC\_DR) を正しく読み出すには、APB1 クロック周波数 ( $f_{PCLK}$ ) が RTC クロック周波数 ( $f_{RTCCLK}$ ) の 7 倍以上でなければなりません。これにより、同期メカニズムの安全な動作が保証されます。

APB1 クロック周波数が RTC クロック周波数の 7 倍未満である場合、ソフトウェアによってカレンダー時間と日付のレジスタを 2 回読み出す必要があります。RTC\_TR の 2 回目の読み出しが 1 回目の

読み出しと同じ結果であれば、データが正しいことが保証されます。同じでない場合は、3 回目の読み出しアクセスを行う必要があります。どの場合も、APB1 クロック周波数は必ず RTC クロック周波数以上でなければなりません。

RTC\_ISR レジスタの RSF ビットは、カレンダーレジスタが RTC\_SSR、RTC\_TR、および RTC\_DR シャドウレジスタにコピーされるたびにセットされます。コピーは、2 RTCCLK サイクルごとに行われます。3 つの値における一貫性を保証するため、RTC\_SSR または RTC\_TR のどちらかを読み出すと、高次カレンダーシャドウレジスタの値は RTC\_DR が読み出されるまでロックされます。ソフトウェアが 2 RTCCLK サイクル未満の間隔でカレンダーの読み出しアクセスを行う場合、最初のカレンダー読み出し後に RSF をソフトウェアでクリアする必要があり、その後ソフトウェアは、RSF ビットがセットされるまで待ってから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを再読み出す必要があります。

低電力モード (STOP または STANDBY) からのウェイクアップ後は、RSF をソフトウェアでクリアする必要があります。その後、ソフトウェアは、いま一度 RSF がセットされるまで待ってから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを再読み出す必要があります。

RSF ビットは、ウェイクアップ後にクリアする必要がありますが、低電力モードに入る前には、その必要はありません。

システムリセット後、ソフトウェアは RSF がセットされるまで待ってから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを読み出す必要があります。実際、システムリセットがかかると、シャドウレジスタはデフォルト値にリセットされます。

初期化 ([カレンダーの初期化と設定 \(779 ページ\)](#)) を参照) 後、ソフトウェアは RSF がセットされるまで待ってから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを読み出す必要があります。

同期 ([セクション 27.3.10 : RTC の同期](#)) を参照) 後、ソフトウェアは RSF がセットされるまで待ってから、RTC\_SSR、RTC\_TR、および RTC\_DR レジスタを読み出す必要があります。

## RTC\_CR レジスタ (バイパスシャドウレジスタ) の BYPSHAD 制御ビットがセットされている場合

カレンダーレジスタを読み出すと、カレンダーカウンタの値が直接与えられるため、RSF ビットがセットされるのを待つ必要はありません。シャドウレジスタは低電力モード (STOP または STANDBY) では更新されないため、低電力モード終了後に特にこのような読み出しが有用です。

BYPSHAD ビットが 1 にセットされている場合、レジスタへの 2 回の読み出しアクセス間で RTCCLK エッジが発生した場合は、さまざまなレジスタ間で互いに不整合が起きる場合があります。さらに、読み出し操作中に RTCCLK エッジが発生した場合、レジスタの 1 つが不正な値となる場合があります。ソフトウェアはすべてのレジスタを 2 回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要があります。その代わりに、ソフトウェアはカレンダーレジスタの最下位の数値を 2 回比較するだけで構いません。

**注 :** *BYPSHAD = 1 の間、カレンダーレジスタの読み出し命令が完了するには 1 APB サイクルだけ余計に必要となります。*

### 27.3.9 RTC のリセット

カレンダーシャドウレジスタ (RTC\_SSR、RTC\_TR、RTC\_DR) および RTC ステータスレジスタ (RTC\_ISR) の一部のビットは、利用可能なすべてのシステムリセットリソースによってデフォルト値にリセットされます。

逆に、次のレジスタはバックアップドメインリセットによってそれぞれのデフォルト値にリセットされ、システムリセットの影響は受けません。RTC の現在のカレンダーレジスタ、RTC 制御レジスタ (RTC\_CR)、プリスケールレジスタ (RTC\_PRER)、RTC 較正レジスタ (RTC\_CALR)、RTC シフト



レジスタ (RTC\_SHIFTR)、RTC タイムスタンプレジスタ (RTC\_TSSSR, RTC\_TSTR および RTC\_TSDR)、RTC タンパおよびオルタネート機能設定レジスタ (RTC\_TAFCR)、RTC バックアップレジスタ (RTC\_BKPxR)、ウェイクアップタイマレジスタ (RTC\_WUTR)、アラーム A とアラーム B レジスタ (RTC\_ALRMASR/RTC\_ALRMAR と RTC\_ALRMBSSR/RTC\_ALRMBR)。

さらに、LSE クロックによって駆動されている際にリセットソースがバックアップドメインリセットと異なる場合、システムリセットがかかっても RTC は動作を続けます (システムリセットの影響を受けない RTC クロックソースのリストの詳細については、リセットおよびクロックコントローラの RTC クロックセクションを参照してください)。バックアップドメインリセットが発生すると、RTC は停止し、すべての RTC レジスタがリセット値にセットされます。

## 27.3.10 RTC の同期

RTC は、高精度でリモートクロックと同期できます。サブセカンド項目 (RTC\_SSR または RTC\_TSSSR) を読み出すと、リモートクロックによって維持されている時刻と RTC 間の正確なオフセットが計算できます。その後、RTC\_SHIFTR を使用してほんの一瞬クロックを「シフト」することによって RTC を調整し、このオフセットを取り除くことができます。

RTC\_SSR には、同期プリスケアラのカウンタの値が入っています。これにより、RTC によって維持されている正確な時刻を計算でき、その分解能は  $1/(\text{PREDIV}_S + 1)$  秒です。その結果、同期プリスケアラ値 (PREDIV\_S[14:0]) を増加させることにより分解能を改善できます。許可されている最大分解能 (32768 Hz クロックで 30.52  $\mu$ s) は、PREDIV\_S を 0x7FFF にセットすることにより得られます。

ただし、PREDIV\_S を増加させるということは、同期プリスケアラの出力を 1 Hz に維持するため PREDIV\_A を減らす必要があることを意味します。このように、非同期プリスケアラの出力周波数が増加すると、RTC の動的消費電力が増加する場合があります。

RTC は、RTC シフト制御レジスタ (RTC\_SHIFTR) を使って微調整できます。RTC\_SHIFTR に書き込むことにより、 $1/(\text{PREDIV}_S + 1)$  秒の分解能で、クロックを最大 1 秒だけシフト (遅れ/進み) させることができます。このシフト操作の本質は、同期プリスケアラのカウンタ SS[15:0] に SUBFS[14:0] 値を加算することであり、この操作はクロックを遅らせることになります。同時に ADD1S ビットがセットされた場合、1 秒追加すると同時に秒の小数部を差し引くことになるため、クロックを進めることになります。

**注意：** シフト操作を始める前に、ユーザは SS[15] = 0 であることを確認し、オーバーフローが発生しないようにする必要があります。

RTC\_SHIFTR レジスタへの書き込みによってシフト操作が始まるとすぐに、シフト操作が保留中であることを示す SHPF フラグがハードウェアによってセットされます。このビットは、シフト操作が完了するとすぐに、ハードウェアによってクリアされます。

**注意：** この同期機能はリファレンスクロック検出機能とは両立できません。具体的には、REFCKON = 1 のときにファームウェアから RTC\_SHIFTR への書き込みはできません。

## 27.3.11 RTC リファレンスクロック検出

RTC カレンダの更新は、リファレンスクロックである RTC\_REFIN に同期させることができます。通常は商用電源 (50 または 60 Hz) です。RTC\_REFIN リファレンスクロックには、32.768 kHz LSE クロックより高い精度が必要です。RTC\_REFIN 検出を有効にした際 (RTC\_CR の REFCKON ビットが 1 にセット)、カレンダは引き続き LSE クロックによって駆動され、RTC\_REFIN はカレンダ更新周波数 (1 Hz) による誤差の補正に使用されます。

各 1 Hz クロックエッジは、一番近い RTC\_REFIN クロックエッジ (所与の時間枠内に見つかった場合) と比較されます。ほとんどの場合、2 つのクロックエッジは正しく整列しています。LSE クロックが不正確なために 1 Hz のクロックがずれた場合、RTC は 1 Hz のクロックを少しシフトさせ、そ

の後の 1 Hz のクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダーはリファレンスクロックと同様に正確になります。

RTC は、32.768 kHz クォーツから生成される 256 Hz クロック (ck\_apre) を使用して、リファレンスクロックソースがあるかどうかを検出します。検出は各カレンダー更新 (1 秒ごと) 程度の時間枠で行われます。最初のリファレンスクロックエッジを検出する際、この時間枠は ck\_apre 7 周期に等しくなります。その後のカレンダー更新では、ck\_apre 3 周期より短い時間枠が使用されます。

リファレンスクロックがこの時間枠内で検出されるたびに、ck\_apre クロックを出力する非同期プリスケラは強制的に再ロードされます。プリスケラは同時に再ロードされるので、リファレンスクロックおよび 1 Hz のクロックが整列するタイミングには影響しません。クロックが整列していない場合、後の 1 Hz クロックエッジは、リファレンスクロックと整列するように再ロードによって少しシフトされます。

リファレンスクロックが停止した (ck\_apre 3 周期の枠内でリファレンスクロックエッジが発生しない) 場合、カレンダーは LSE クロックのみを基準にして更新が継続されます。その後 RTC は ck\_spre エッジを中心として ck\_apre 7 周期という広い検出時間枠でリファレンスクロックを待ちます。

基準クロック検出を有効にした場合、PREDIV\_A および PREDIV\_S を以下に示すそれぞれのデフォルト値にセットする必要があります。

- PREDIV\_A = 0x007F
- PREDIV\_S = 0x00FF

**注:** *RTC\_REFIN クロック検出は、STANDBY モードでは利用できません。*

## 27.3.12 RTC の高精度デジタル較正

RTC 周波数の精度は、-487.1~+488.5 ppm の範囲で、分解能約 0.954 ppm でデジタル的に較正できます。周波数の修正は、一連の微調整 (個々の RTCCLK パルスの追加や削除) によって行われます。このような調整は、短い期間で観測された場合でも RTC が十分に較正されるように、かなり広範に分散して行われます。

この高精度デジタル較正は、入力周波数が 32768 Hz の場合、RTCCLK 約 2<sup>20</sup> パルスのサイクルまたは 32 秒の間で行われます。このサイクルは、RTCCLK によって駆動される 20 ビットカウンタ、cal\_cnt[19:0] によって維持されます。

高精度較正レジスタ (RTC\_CALR) によって、32 秒サイクル中にマスクされる RTCCLK クロックサイクル数を指定します。

- CALM[0] ビットを 1 にセットすると、32 秒サイクルの中でちょうど 1 パルスがマスクされます。
- CALM[1] ビットを 1 にセットすると、さらに 2 サイクルがマスクされます。
- CALM[2] ビットを 1 にセットすると、さらに 4 サイクルがマスクされます。
- CALM[8] ビットを 1 にセットするまで続けると、256 クロックがマスクされます。

**注:** *CALM[8:0] (RTC\_CALR) によって、32 秒サイクル中にマスクされる RTCCLK パルス数を指定します。CALM[0] ビットを 1 にセットすると、cal\_cnt[19:0] = 0x80000 になった時点で、32 秒サイクル中でちょうど 1 パルスがマスクされます。CALM[1] = 1 では、さらに 2 サイクルがマスクされ (cal\_cnt = 0x40000 および 0xC0000)、CALM[2] = 1 では、さらに 4 サイクルがマスクされ (cal\_cnt = 0x20000/0x60000/0xA0000/0xE0000)、CALM[8] = 1 まで続けると、256 クロックがマスクされ (cal\_cnt = 0xXX800)。*

CALM では、細かい分解能で RTC 周波数を最大 487.1 ppm 負の方向に調整することができ、CALP ビットでは周波数を 488.5 ppm 正の方向に調整することができます。CALP を 1 にセットすることにより、実質上は、RTCCLK 2<sup>11</sup> サイクルごとに、RTCCLK パルスが 1 パルス追加で挿入されます。すなわち、32 秒サイクルごとに 512 クロックが追加されることとなります。



CALM を CALP と合わせて使用すると、32 秒サイクルの間に RTCCLK -511 から +512 サイクルまでのオフセットが追加でき、これは約 0.954 ppm の分解能で較正範囲 -487.1~+488.5 ppm に換算されます。

有効較正周波数 (FCAL) を入力周波数 (FRTCCLK) に対して求める計算式は次のとおりです。

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$

## PREDIV\_A < 3 の場合の較正

非同期プリスケアラ値 (RTC\_PRER レジスタの PREDIV\_A ビット) が 3 未満の場合、CALP ビットを 1 にセットすることはできません。CALP がすでに 1 にセットされていて、PREDIV\_A ビットが 3 未満の値にセットされた場合、CALP の設定値は無視され、CALP が 0 に設定された場合と同じように較正されます。

PREDIV\_A が 3 未満の状態では較正を実施するには、各秒のカウントが 8 RTCCLK クロックサイクル分早められるように同期プリスケアラ値 (PREDIV\_S) を小さくする必要があります。これは 32 秒毎に 256 クロックサイクル追加することに相当します。結果として、CALM ビットのみを使用して 32 秒周期の間に 256 クロックパルスから 256 クロックパルス (243.3 から 244.1 ppm の較正範囲に相当)を追加することができます。

公称 RTCCLK 周波数が 32768 Hz で、PREDIV\_A が 1 (分周比 2) の場合、PREDIV\_S を 16383 ではなく 16379 (4 少ない) にセットする必要があります。また、PREDIV\_A が 0 の場合、PREDIV\_S を 32767 ではなく 32759 (8 少ない) にセットする必要がありますので注意して下さい。

PREDIV\_S をこのように減少させた場合、較正された入力クロックの有効周波数の

式は次のようになります。

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

この場合、RTCCLK が正確に 32768.00 Hz であれば、CALM[7:0] が 0x100 (CALM 設定範囲の中間値) と等しくなるのが正しい設定です。

## RTC 較正值の確認

RTC の精度は、RTCCLK の正確な周波数を測定し、正しい CALM 値および CALP 値を計算することにより保証されます。オプションの 1 Hz 出力が搭載されており、アプリケーションによって RTC 精度の測定と確認を行うことができます。

ある時間間隔で RTC の周波数を精密に測定すると、デジタル較正サイクルを測定周期とどのように合わせているかにより、測定期間中に最大 2 RTCCLK クロックサイクルの測定誤差が生じます。

ただし、この測定誤差は、測定周期が較正サイクル周期と同じ長さであれば排除できます。この場合、観測される唯一の誤差はデジタル較正の分解能による誤差となります。

- デフォルトでは、較正サイクル周期は 32 秒です。

このモードを使用して正確に 32 秒で 1 Hz 出力の精度を測定すると、その精度は 0.477 ppm (較正分解能の制限により 32 秒で 0.5 RTCCLK サイクル) 以内となることが保証されます。

- RTC\_CALR レジスタの CALW16 ビットを 1 にセットして、較正サイクル周期を強制的に 16 秒にすることができます。

この場合、RTC 精度は最大誤差 0.954 ppm (16 秒で 0.5 RTCCLK サイクル) で 16 秒間で測定できます。ただし、較正分解能が下がるため、長期的な RTC 精度もまた 0.954 ppm に下がります。CALW16 が 1 にセットされると、CALM[0] ビットは 0 のままとなります。

- RTC\_CALR レジスタの CALW8 ビットを 1 にセットして、較正サイクル周期を強制的に 8 秒にすることができます。



この場合、RTC 精度は最大誤差 1.907 ppm (8 秒で 0.5 RTCCLK サイクル) で 8 秒で測定できます。長期的な RTC 精度もまた 1.907 ppm に下がります。CALW8 が 1 にセットされると、CALM[1:0] ビットは 00 のままとなります。

## 動作中の再較正

次の処理を実施することにより、RTC\_ISR/INITF = 0 の間でも、較正レジスタ (RTC\_CALR) を動作中に更新することができます。

1. RTC\_ISR/RECALPF (再較正保留フラグ) をポーリングします。
2. このフラグが 0 にセットされている場合は、必要に応じて新しい値を RTC\_CALR に書き込みます。すると、RECALPF が自動的に 1 にセットされます。
3. RTC\_CALR への書き込み動作後 ck\_apre 3 サイクル以内に、新しい較正設定が有効になります。

## 27.3.13 タイムスタンプ機能

タイムスタンプは、RTC\_CR レジスタの TSE ビットを 1 にセットすることにより有効になります。

RTC\_TS ピンでタイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC\_TSSSR、RTC\_TSTR、RTC\_TSDR) にカレンダーが保存されます。

また、タイムスタンプイベントが発生すると、RTC\_ISR レジスタのタイムスタンプフラグビット (TSF) がセットされます。

RTC\_CR レジスタの TSIE ビットをセットすることにより、タイムスタンプイベントが発生したときに割り込みが生成されます。

タイムスタンプフラグ (TSF) がすでにセットされている間に新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) がセットされ、タイムスタンプレジスタ (RTC\_TSTR および RTC\_TSDR) は、その前のイベントの結果を維持します。

**注：** *同期処理のため、TSF はタイムスタンプイベント発生から ck\_apre 2 サイクル後にセットされます。一方、TSOVF のセットに遅延はありません。これは、2 つのタイムスタンプイベントの発生したタイミングが近い場合、TSF がまだ“0”であっても TSOVF が“1”と検出される可能性があることを意味します。よって、TSOVF のポーリングは TSF がセットされた後に実施することをお勧めします。*

**注意：** *TSF ビットがクリアされたと思われた直後にタイムスタンプイベントが発生した場合、TSF および TSOVF ビットの両方がセットされます。同時に発生するタイムスタンプイベントのマスキングを回避するため、TSF がすでに“1”と読み出されていない限りは、アプリケーションによる“0”クリアの処理を行ってはなりません。*

オプション機能として、タンパイベントによってタイムスタンプイベントを記録することもできます。TAMPTS 制御ビットの詳細については、[セクション 27.6.16 : RTC タンパおよびオルタネート機能設定レジスタ \(RTC\\_TAFCR\)](#) を参照してください。

## 27.3.14 タンパ検出

RTC\_TAMPx 入力イベントは、フィルタを使用してエッジ検出またはレベル検出のどちらにも設定できます。

タンパ検出は次の目的で設定することができます。

- RTC バックアップレジスタを消去する。
- STOP モードおよび STANDBY モードからのウェイクアップが可能な割り込みを生成する。



## RTC バックアップレジスタ

バックアップレジスタ (RTC\_BKPxR) は、システムリセットや STANDBY モードからのウェイクアップではリセットされません。

バックアップレジスタは、タンパ検出イベント発生時にリセットされます ([セクション 27.6.19: RTC バックアップレジスタ \(RTC\\_BKPxR\)](#) および [タンパ検出の初期化 \(786 ページ\)](#) を参照。またはフラッシュの読み出し保護がレベル 1 からレベル 0 に変わったとき)。

## タンパ検出の初期化

各入力は、RTC\_TAFPCR レジスタにおいて該当する TAMPxE ビットを 1 にセットすることにより有効にできます。

各 RTC\_TAMPx タンパ検出入力は、RTC\_ISR レジスタの TAMPxF フラグに関連付けられています。

TAMPxF フラグは、ピン上でタンパイベントが発生した後にアサートされます。その際の遅延時間を以下に示します。

- TAMPFLT = 0x0 以外の場合、ck\_apre 3 サイクル (フィルタを使ったレベル検出)
- TAMPTS = 1 の場合、ck\_apre 3 サイクル (タンパイベント時のタイムスタンプ)
- TAMPFLT = 0x0 (エッジ検出) および TAMPTS = 0 の場合、遅延なし

TAMPxF がセットされている場合に、この周期中に同一ピンで発生した新たなタンパイベントを検出することはできません。

RTC\_TAFPCR レジスタの TAMPIE ビットをセットすることにより、タンパ検出イベント発生時に割り込みが生成されます。

## タンパイベント時のタイムスタンプ

TAMPTS を“1”にセットすると、すべてのタンパイベントがタイムスタンプイベントを発生させるようになります。この場合、通常のタイムスタンプイベント発生時と同様に TSF ビットまたは TSOVF ビットが RTC\_ISR でセットされます。TSF または TSOVF がセットされるのと同時に、影響を受けるタンパフラグレジスタ、TAMPxF がセットされます。

## タンパ入力でのエッジ検出

TAMPFLT ビットが“00”の場合、該当する TAMPxTRG ビットに応じて立ち上がりエッジまたは立ち下がりエッジが観測されると、RTC\_TAMPx ピンがタンパ検出イベントを生成します。エッジ検出を選択すると、RTC\_TAMPx 入力の内部プルアップ抵抗が無効になります。

**注意：** *タンパ検出イベントを確実に検出するため、RTC\_TAMPx ピンが有効になる前にタンパ検出イベントが発生した場合でも検出できるように、エッジ検出に使用される信号は対応する TAMPxE ビットと論理積がとられます。*

- TAMPxTRG = 0 の場合：タンパ検出が有効になる (TAMPxE ビットが 1 にセットされる) 前に RTC\_TAMPx オルタネート機能がすでにハイのとき、TAMPxE がセットされた後に RTC\_TAMPx に立ち上がりエッジ入力がなくとも、RTC\_TAMPx 入力が有効になるとすぐにタンパイベントが検出されます。
- TAMPxTRG = 1 の場合：タンパ検出が有効になる前に RTC\_TAMPx オルタネート機能がすでにローのとき、RTC\_TAMPx 入力が有効になるとすぐに (TAMPxE がセットされた後に RTC\_TAMPx に立ち下がりエッジ入力がなくとも)、タンパイベントが検出されます。

タンパイベントが検出されクリアされた後に、バックアップレジスタ (RTC\_BKPxR) を再プログラムする場合には、事前に、RTC\_TAMPx オルタネート機能を無効にしてから再度有効にする (TAMPxE を 1 にセット) 必要があります。これによって、RTC\_TAMPx 入力の値がタンパ検出を示している間に、アプリケーションがバックアップレジスタにデータを書き込むのを防ぎます。これは、RTC\_TAMPx オルタネート機能の入力でのレベル検出に相当します。

**注：** タンパ検出は、V<sub>DD</sub> 電源がオフのときでも有効です。バックアップレジスタの不必要なリセットを避けるには、RTC\_TAMPx オルタネート機能が設定されているピンを外部で適切な信号レベルに接続しておく必要があります。

## RTC\_TAMPx 入力でのフィルタを使ったレベル検出

フィルタを使ったレベル検出は、TAMPFLT を 0 以外の値にセットすることにより行われます。タンパ検出イベントは、(TAMPFLT に応じて) 2、4 または 8 回のいずれかの連続したサンプルが TAMPxTRG ビットで指定するレベルで観測されたときに生成されます。

RTC\_TAMPx 入力は、TAMPPUDIS が 1 にセットされて無効な状態になっていない限り、その状態がサンプリングされる前に I/O の内部プルアップ抵抗でプリチャージされています。プリチャージの継続時間は TAMPPRCH ビットによって決定され、RTC\_TAMPx 入力ピンにおけるより大きな容量を持たせることができます。

タンパ検出の遅延時間と、プルアップによる電力消費との間のトレードオフは、TAMPFREQ を使用してレベル検出のサンプリング周波数を決定することにより、最適化できます。

**注：** プルアップ抵抗の電気的特性については、データシートを参照してください。

## 27.3.15 較正クロック出力

RTC\_CR レジスタで COE ビットが 1 にセットされると、RTC\_CALIB デバイス出力にリファレンスクロックが供給されます。

RTC\_CR レジスタの COSEL ビットがリセットされ、かつ PREDIV\_A = 0x7F である場合、RTC\_CALIB 周波数は  $f_{\text{RTCCLK}}/64$  です。これは 32.768 kHz の RTCCLK 周波数に対する 512 Hz の較正出力に相当します。立ち下がりがエッジには軽いジッタがあるため、RTC\_CALIB のデューティサイクルは不規則になります。したがって、立ち上がりがエッジの使用が推奨されます。

COSEL がセットされ、かつ "PREDIV\_S+1" がゼロ以外の 256 の倍数である場合 (すなわち、PREDIV\_S[7:0] = 0xFF)、RTC\_CALIB 周波数は  $f_{\text{RTCCLK}}/(256 * (\text{PREDIV\_A}+1))$  となります。これは、RTCCLK 周波数が 32.768 kHz で、プリスケラデフォルト値 (PREDIV\_A = 0x7F、PREDIV\_S = 0xFF) に対する 1 Hz の較正出力に相当します。

**注：** RTC\_CALIB または RTC\_ALARM 出力が選択されると、RTC\_OUT ピンは自動的に出力オルタネート機能に設定されます。

## 27.3.16 アラーム出力

RTC\_CR レジスタの OSEL[1:0] 制御ビットを使用してアラームオルタネート機能出力、RTC\_ALARM を有効にし、出力となる機能を選択します。これらの機能は、RTC\_ISR レジスタの該当するフラグの内容を反映します。

RTC\_CR の POL 制御ビットは、POL が 1 にセットされているときに選択されたフラグビットの逆が出力されるよう、出力の極性を決定します。

### アラームオルタネート機能出力

RTC\_ALARM ピンは、制御ビット ALARMOUTTYPE (RTC\_TAFCR レジスタ) を使用して、出力オープンドレインまたは出力プッシュプルに設定できます。



注: **RTC\_ALARM** 出力が有効になると、この設定は、**RTC\_CALIB** (COE ビットは無視され、クリアされたまま) の設定よりも優先されます。  
**RTC\_CALIB** または **RTC\_ALARM** 出力が選択されると、**RTC\_OUT** ピンは自動的に出力オルタネート機能に設定されます。

## 27.4 RTC 低電力モード

表 138. 低電力モードが RTC に与える影響

モード	説明
SLEEP	影響なし。 RTC 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンパイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスは STOP モードから復帰します。
STANDBY	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンパイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスは STANDBY モードから復帰します。

## 27.5 RTC 割り込み

すべての RTC 割り込みは、NVIC コントローラに接続されています。[セクション 14.3 : EXTI レジスタ](#)を参照してください。

RTC アラーム割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで RTC アラームイベントに対応する NVIC ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で **RTC\_ALARM\_IRQ** チャンネルを設定し、有効にします。
3. RTC が RTC アラームを生成するように設定します。

RTC タンパ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで RTC タンパイベントに対応する NVIC ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で **RTC\_TAMP\_STAMP\_IRQ** チャンネルを設定し、有効にします。
3. RTC が RTC タンパイベントを検出するように設定します。

RTC タイムスタンプ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードで RTC タイムスタンプイベントに対応する NVIC ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で **RTC\_TAMP\_STAMP\_IRQ** チャンネルを設定し、有効にします。
3. RTC タイムスタンプイベントを検出するよう RTC を設定します。

ウェイクアップタイマ割り込みを有効にするには、次のシーケンスが必要です。

1. 割り込みモードでウェイクアップタイマ割り込みイベントに対応する NVIC ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
2. NVIC で **RTC\_WKUP\_IRQ** チャンネルを設定し、有効にします。
3. RTC が RTC ウェイクアップタイマイベントを検出するように設定します。

表 139. 割り込み制御ビット

割り込みイベント	イベントフラグ	有効化制御ビット	SLEEPモードの終了	STOPモードの終了	STANDBYモードの終了
アラーム A	ALRAF	ALRAIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
アラーム B	ALRBF	ALRBIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TS 入力 (タイムスタンプ)	TSF	TSIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TAMP1 入力検出	TAMP1F	TAMPIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TAMP2 入力検出	TAMP2F	TAMPIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
RTC_TAMP3 入力検出	TAMP3F	TAMPIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>
ウェイクアップタイマ割り込み	WUTF	WUTIE	あり	あり <sup>(1)</sup>	あり <sup>(1)</sup>

1. STOP モードおよび STANDBY モードからのウェイクアップは、RTC クロックソースが LSE または LSI のときのみ可能です。

## 27.6 RTC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

### 27.6.1 RTC 時刻レジスタ (RTC\_TR)

RTC\_TR は、カレンダー時刻シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。 [カレンダーの初期化と設定 \(779 ページ\)](#) および [カレンダーの読み出し \(780 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、 [RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x00

バックアップドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位



ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 27.6.2 RTC 日付レジスタ (RTC\_DR)

RTC\_DR は、カレンダー日付シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。[カレンダーの初期化と設定 \(779 ページ\)](#) および [カレンダーの読み出し \(780 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x04

バックアップ ドメインリセット値 : 0x0000 2101

システムリセット : BYPSHAD = 0 の場合、0x0000 2101 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:20 **YT[3:0]** : BCD 形式での年の十の位

ビット 19:16 **YU[3:0]** : BCD 形式での年の一の位

ビット 15:13 **WDU[2:0]** : 曜日

000 : 禁止  
001 : 月曜日  
...  
111 : 日曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

## 27.6.3 RTC 制御レジスタ (RTC\_CR)

アドレスオフセット : 0x08

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COE	OSEL[1:0]			POL	COSEL	BKP	SUB1H	ADD1H
								r/w	r/w	r/w		r/w	r/w	r/w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPS HAD	REFCKON	TSEDGE	WUCKSEL[2:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **COE** : 較正出力イネーブル

このビットは、RTC\_CALIB 出力を有効にします。

- 0 : 較正出力は無効です。
- 1 : 較正出力は有効です。

ビット 22:21 **OSEL[1:0]** : 出力選択

これらのビットは、RTC\_ALARM 出力に送られるフラグの選択に使用します。

- 00 : 出力は無効です。
- 01 : アラーム A 出力は有効です。
- 10 : アラーム B 出力は有効です。
- 11 : ウェイクアップ出力は有効です。

ビット 20 **POL** : 出力極性

このビットは、RTC\_ALARM 出力の極性の設定に使用します。

- 0 : ALRAF/ALRBF/WUTF がアサートされると、このピンがハイになります (OSEL[1:0] に応じて)。
- 1 : ALRAF/ALRBF/WUTF がアサートされると、このピンがローになります (OSEL[1:0] に応じて)。

ビット 19 **COSEL** : 較正出力選択

COE = 1 のとき、このビットによって RTC\_CALIB に出力される信号を選択します。

- 0 : 較正出力は 512 Hz です。
- 1 : 較正出力は 1 Hz です。

これらの周波数は、RTCCLK が 32.768 kHz で、プリスケアラがデフォルト値 (PREDIV\_A = 127 および PREDIV\_S = 255) の場合に有効です。 [セクション 27.3.15 : 較正クロック出力](#) を参照してください。

ビット 18 **BKP** : バックアップ

このビットは、サマータイムの変更を実施したか否かを記憶しておくため、ユーザが書き込むことができます。

ビット 17 **SUB1H** : 1 時間差し引き (冬時間変更)

このビットを初期化モード以外の人にセットすると、現在時刻が 0 でない場合にカレンダー時刻から 1 時間を差し引きます。このビットは常に 0 として読み出されます。

現在時間が 0 のときにこのビットをセットしても、影響はありません。

- 0 : 影響なし。
- 1 : 現在時刻から 1 時間差し引きます。これは、冬時間変更で使用できます。

**ビット 16 ADD1H** : 1 時間加算 (サマータイム変更)

このビットを初期化モード以外のときにセットすると、カレンダー時刻に 1 時間加算します。このビットは常に 0 として読み出されます。

0 : 影響なし。

1 : 現在時刻に 1 時間加算します。これは、サマータイム変更に使用できます。

**ビット 15 TSIE** : タイムスタンプ割り込みイネーブル

0 : タイムスタンプ割り込みは無効です。

1 : タイムスタンプ割り込みは有効です。

**ビット 14 WUTIE** : ウェイクアップタイマ割り込みイネーブル

0 : ウェイクアップタイマ割り込みは無効です。

1 : ウェイクアップタイマ割り込みは有効です。

**ビット 13 ALRBIE** : アラーム B 割り込みイネーブル

0 : アラーム B 割り込みは無効です。

1 : アラーム B 割り込みは有効です。

**ビット 12 ALRAIE** : アラーム A 割り込みイネーブル

0 : アラーム A 割り込みは無効です。

1 : アラーム A 割り込みは有効です。

**ビット 11 TSE** : タイムスタンプイネーブル

0 : タイムスタンプは無効です。

1 : タイムスタンプは有効です。

**ビット 10 WUTE** : ウェイクアップタイマイネーブル

0 : ウェイクアップタイマは無効です。

1 : ウェイクアップタイマは有効です。

**ビット 9 ALRBE** : アラーム B イネーブル

0 : アラーム B は無効です。

1 : アラーム B は有効です。

**ビット 8 ALRAE** : アラーム A イネーブル

0 : アラーム A は無効です。

1 : アラーム A は有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

**ビット 6 FMT** : 時間形式

0 : 24 時間 / 日形式

1 : AM / PM 時間形式

**ビット 5 BYPSHAD** : シャドウレジスタをバイパスします。

0 : カレンダー値は (RTC\_SSR、RTC\_TR、RTC\_DR から読み出す場合)、シャドウレジスタから取得され、これらは 2 RTCCLK サイクルごとに 1 回更新されます。

1 : カレンダー値は (RTC\_SSR、RTC\_TR、RTC\_DR から読み出す場合)、カレンダーカウンタから直接取得されます。

**注 :** APB1 クロックの周波数が RTCCLK の 7 倍未満である場合、BYPSHAD は“1”にセットする必要があります。



ビット 4 **REFCKON** : RTC\_REFIN リファレンスクロック検出イネーブル (50 または 60 Hz)

- 0 : RTC\_REFIN 検出は無効です。
- 1 : RTC\_REFIN 検出は有効です。

**注 :** *PREDIV\_S は 0x00FF である必要があります。*

ビット 3 **TSEDGE** : タイムスタンプイベントアクティブエッジ

- 0 : RTC\_TS 入力の立ち上がりエッジによってタイムスタンプイベントを生成します。
  - 1 : RTC\_TS 入力の立ち下がりエッジによってタイムスタンプイベントを生成します。
- 不要な TSF 設定を回避するため、TSEDGE が変化した場合には TSE をリセットする必要があります。

ビット 2:0 **WUCKSEL[2:0]** : ウェイクアップクロック選択

- 000 : RTC/16 クロックが選択されます。
- 001 : RTC/8 クロックが選択されます。
- 010 : RTC/4 クロックが選択されます。
- 011 : RTC/2 クロックが選択されます。
- 10x : ck\_spre (通常は 1 Hz) クロックが選択されます。
- 11x : ck\_spre (通常は 1 Hz) クロックが選択され、 $2^{16}$  が WUT カウンタ値に加算されます (下記注を参照)。

**注 :** *初期化モード (RTC\_ISR/INITF = 1) の場合のみ、このレジスタのビット 7、6、4 が書き込めます。*  
*WUT = ウェイクアップユニットカウンタ値  $WUT = (0x0000 \sim 0xFFFF) + 0x10000$  (WUCKSEL[2:1] = 11 の場合追加されます。)*  
*このレジスタのビット 2~0 は、RTC\_CR WUTE ビット = 0 かつ RTC\_ISR WUTWF ビット = 1 の場合にのみ書き込めます。*  
*カレンダーの時間項目のインクリメント中は時間を変更しないことが推奨されます。カレンダーの時間項目のインクリメントがマスクされる可能性があるためです。*  
*ADD1H および SUB1H の変更は、次の秒から有効になります。*  
*このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。*

## 27.6.4 RTC 初期化とステータスレジスタ (RTC\_ISR)

このレジスタは、書き込み保護されています (RTC\_ISR[13:8] ビットを除く)。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x0C

バックアップ ドメインリセット値 : 0x0000 0007

システムリセット : 0 にクリアされる INIT、INITF、RSF ビット以外は影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRB WF	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

ビット 31:17 予約済みであり、リセット値のままにしておかなければなりません。

ビット 16 **RECALPF** : 再較正保留フラグ

ソフトウェアによって RTC\_CALR レジスタに書き込みが行われると、RECALPF ステータスフラグが自動的に“1”にセットされ、RTC\_CALR レジスタがブロックされたことを示します。新たな較正設定が認識されると、このビットは“0”に戻ります。[動作中の再較正](#)を参照してください。

ビット 15 **TAMP3F** : RTC\_TAMP3 検出フラグ

このフラグは、RTC\_TAMP3 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 14 **TAMP2F** : RTC\_TAMP2 検出フラグ

このフラグは、RTC\_TAMP2 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 13 **TAMP1F** : RTC\_TAMP1 検出フラグ

このフラグは、RTC\_TAMP1 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。

ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 12 **TSOVF** : タイムスタンプオーバーフローフラグ

このフラグは、TSF が既にセットされている間にタイムスタンプイベントが発生したときに、ハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。そうしないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 11 **TSF** : タイムスタンプフラグ

このフラグは、タイムスタンプイベントが発生したときに、ハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 10 **WUTF** : ウェイクアップタイマフラグ

このフラグは、ウェイクアップ自動再ロードカウンタが 0 に到達したときに、ハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

**ビット 9 ALRBF** : アラーム B フラグ

このフラグは、時刻/日付レジスタ (RTC\_TR および RTC\_DR) がアラーム B レジスタ (RTC\_ALRMBR) と一致したときにハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

**ビット 8 ALRAF** : アラーム A フラグ

このフラグは、時刻/日付レジスタ (RTC\_TR および RTC\_DR) がアラーム A レジスタ (RTC\_ALRMAR) と一致したときにハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

**ビット 7 INIT** : 初期化モード

0 : フリーランニングモード

1 : 時刻と日付レジスタ (RTC\_TR と RTC\_DR)、およびプリスケアラレジスタ (RTC\_PRER) のプログラムに使用する初期化モードです。INIT がリセットされると、カウンタは停止し、新しい値からカウントし始めます。

**ビット 6 INITF** : 初期化フラグ

このビットが 1 にセットされると、RTC は初期化状態となり、時刻、日付およびプリスケアラレジスタが更新できます。

0 : カレンダーレジスタを更新できません。

1 : カレンダーレジスタを更新できます。

**ビット 5 RSF** : レジスタ同期フラグ

このビットは、カレンダーレジスタがシャドウレジスタ (RTC\_SSRx、RTC\_TRx および RTC\_DRx) にコピーされるたびにハードウェアによってセットされます。このビットは、シフト操作が保留中 (SHPF = 1) に初期化モードで、またはバイパスシャドウレジスタモード (BYP SHAD = 1) で、ハードウェアによってクリアされます。このビットは、ソフトウェアでクリアすることもできます。

初期化モードでソフトウェアまたはハードウェアによってクリアされます。

0 : カレンダーシャドウレジスタはまだ同期していません。

1 : カレンダーシャドウレジスタは同期しています。

**ビット 4 INITS** : 初期化ステータスフラグ

このビットは、カレンダーの年の項目が 0 ではないとき (バックアップ ドメインリセット状態) にハードウェアによってセットされます。

0 : カレンダーは初期化されていません。

1 : カレンダーは初期化されています。

**ビット 3 SHPF** : シフト操作保留

0 : 保留中のシフト操作はありません。

1 : 保留中のシフト操作があります。

このフラグは、RTC\_SHIFTR への書き込みによってシフト操作が開始された直後に、ハードウェアによってセットされます。該当するシフト操作が実行されると、ハードウェアによってクリアされます。SHPF ビットに書き込んでも影響はありません。

**ビット 2 WUTWF** : ウェイクアップタイマ書き込みフラグ

このビットは、ハードウェアによって WUTE ビットが RTC\_CR で 0 にセットされた後で 2 RTCCLK サイクルまでセットされ、WUTE ビットが 1 にセットされた後で 2 RTCCLK サイクルまでクリアされます。ウェイクアップタイマ値は、WUTE ビットがクリアされ、WUTWF がセットされたときに変更されます。

0 : ウェイクアップタイマ設定は更新できません。

1 : ウェイクアップタイマ設定は更新できます。

**ビット 1 ALRBWF** : アラーム B 書き込みフラグ

このビットは、RTC\_CR で ALRBIE ビットが 0 にセットされた後、アラーム B 値が変更可能な時にハードウェアによってセットされます。

初期化モードでハードウェアによってクリアされます。

0 : アラーム B は更新できません。

1 : アラーム B は更新できます。

**ビット 0 ALRAWF** : アラーム A 書き込みフラグ

このビットは、RTC\_CR で ALRAE ビットが 0 にセットされた後、アラーム A 値が変更可能な時にハードウェアによってセットされます。

初期化モードでハードウェアによってクリアされます。

0 : アラーム A は更新できません。

1 : アラーム A は更新できます。

**注 :** **ALRAF、ALRBF、WUTF、および TSF ビットは、0 にプログラムされてから 2 APB クロック サイクル後にクリアされます。**

## 27.6.5 RTC プリスケーラレジスタ (RTC\_PRER)

このレジスタは、必ず初期化モードで書き込む必要があります。初期化は、2 回の書き込みアクセスに分けて行う必要があります。 [カレンダーの初期化と設定 \(779 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、 [RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x10

バックアップ ドメインリセット値 : 0x007F 00FF

システムリセット : 影響なし

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

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22:16 **PREDIV\_A[6:0]** : 非同期プリスケーラ分周比

非同期分周比です。

$$ck\_apre \text{ 周波数} = RTCCLK \text{ 周波数} / (PREDIV\_A + 1)$$

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **PREDIV\_S[14:0]** : 同期プリスケーラ分周比

同期分周比です。

$$ck\_spre \text{ 周波数} = ck\_apre \text{ 周波数} / (PREDIV\_S + 1)$$

## 27.6.6 RTC ウェイクアップタイムレジスタ (RTC\_WUTR)

このレジスタは、RTC\_ISR の WUTWF が 1 にセットされているときのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x14

バックアップ ドメインリセット値 : 0x0000 FFFF

システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **WUT[15:0]** : ウェイクアップ自動再ロード値ビット

ウェイクアップタイムが有効 (WUTE が 1 にセット) なとき、ck\_wut の (WUT[15:0] + 1) サイクル毎に WUTF フラグがセットされます。ck\_wut の周期は、RTC\_CR レジスタの WUCKSEL[2:0] ビットで選択します。

WUCKSEL[2] = 1 のとき、ウェイクアップタイムは 17 ビットとなり、WUCKSEL[1] が事実上タイムに再ロードされる最上位ビットである WUT[16] となります。

WUTF の最初のアサートは、WUTE がセットされてから ck\_wut の (WUT+1) サイクル後に発生します。WUCKSEL[2:0] = 011 (RTCCLK/2) のときに WUT[15:0] を 0x0000 にセットすることはできません。

## 27.6.7 RTC アラーム A レジスタ (RTC\_ALRMAR)

このレジスタは、RTC\_ISR の ALRAWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x1C

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]		SU[3:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31 **MSK4** : アラーム A 日付マスク

- 0 : 日付/曜日が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では日付/曜日を無視します。

ビット 30 **WDSEL** : 曜日選択

- 0 : DU[3:0] は日付の一の位を表します。
- 1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位

ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

ビット 23 **MSK3** : アラーム A 時マスク

- 0 : 時が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では時を無視します。

ビット 22 **PM** : AM/PM 表記

- 0 : AM または 24 時間形式
- 1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 **MSK2** : アラーム A 分マスク

- 0 : 分が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では分を無視します。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 **MSK1** : アラーム A 秒マスク

- 0 : 秒が一致すると、アラーム A がセットされます。
- 1 : アラーム A の比較では秒を無視します。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 27.6.8 RTC アラーム B レジスタ (RTC\_ALRMBR)

このレジスタは、RTC\_ISR の ALRBWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x20

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
MSK4		WDSEL		DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MSK2		MNT[2:0]		MNU[3:0]				MSK1	ST[2:0]		SU[3:0]						
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		

ビット 31 **MSK4** : アラーム B 日付マスク

- 0 : 日付/曜日が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では日付/曜日を無視します。

ビット 30 **WDSEL** : 曜日選択

- 0 : DU[3:0] は日付の一の位を表します。
- 1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位

ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

ビット 23 **MSK3** : アラーム B 時マスク

- 0 : 時が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では時を無視します。

ビット 22 **PM** : AM/PM 表記

- 0 : AM または 24 時間形式
- 1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 **MSK2** : アラーム B 分マスク

- 0 : 分が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では分を無視します。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 **MSK1** : アラーム 秒マスク

- 0 : 秒が一致すると、アラーム B がセットされます。
- 1 : アラーム B の比較では秒を無視します。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位



## 27.6.9 RTC 書き込み保護レジスタ (RTC\_WPR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **KEY** : 書き込み保護キー

このバイトはソフトウェアで書き込まれます。

このバイトを読み出すと常に 0x00 が返されます。

RTC レジスタの書き込み保護解除方法については、[RTC レジスタ書き込み保護](#) を参照してください。

## 27.6.10 RTC サブセカンドレジスタ (RTC\_SSR)

アドレスオフセット : 0x28

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS** : サブセカンド値

SS[15:0] は、同期プリスケアラのカウンタ内の値です。秒の小数部は、下の式によって与えられます。

秒の小数部 = ( PREDIV\_S - SS ) / ( PREDIV\_S + 1 )

**注 :** **SS** は、シフト操作後に限り、PREDIV\_S より大きな値となる場合があります。この場合、正確な時刻/日付は、RTC\_TR/RTC\_DR で示される値よりも 1 秒少ない値となります。

## 27.6.11 RTC シフト制御レジスタ (RTC\_SHIFTR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x2C

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **ADD1S** : 1 秒加算

0 : 影響なし。

1 : 時計/カレンダーに 1 秒加算します。

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC\_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

この機能は、SUBFS (下記説明を参照) と共に使用されることを想定しており、不可分操作で、効果的に時計に秒の小数部を加算することを目的としています。

ビット 30:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:0 **SUBFS** : 秒の小数部差し引き

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC\_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

SUBFS に書き込まれた値は、同期プリスケアラのカウンタに加算されます。このカウンタはカウントダウンして行くので、この操作によって、次の式で求める値が効果的にクロックから差し引き(遅延)されます。

$$\text{遅れ (秒)} = \text{SUBFS} / (\text{PREDIV}_S + 1)$$

ADD1S 機能が SUBFS と共に用いられた場合、秒の小数部を効果的にクロックに加算する (クロックを進める) ことができ、実際のクロックの進みは次の式のとおりとなります。

$$\text{進み (秒)} = (1 - (\text{SUBFS} / (\text{PREDIV}_S + 1)))$$

**注 :** **SUBFS に書き込むことにより RSF はクリアされます。その後、ソフトウェアが RSF = 1 まで待つことにより、シャドウレジスタがシフトされた時刻で更新されていることが確実にあります。**

## 27.6.12 RTC タイムスタンプ時刻レジスタ (RTC\_TSTR)

このレジスタの内容は、RTC\_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x30

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

## 27.6.13 RTC タイムスタンプ日付レジスタ (RTC\_TSDR)

このレジスタの内容は、RTC\_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x34

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:13 **WDU[1:0]** : 曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

## 27.6.14 RTC タイムスタンプサブセカンドレジスタ (RTC\_TSSSR)

このレジスタの内容は、RTC\_ISR/TSF がセットされている場合にのみ有効です。また、RTC\_ISR/TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x38

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS** : サブセカンド値

SS[15:0] は、タイムスタンプイベントが発生したときの同期プリスケアラのカウンタの値です。

## 27.6.15 RTC 較正レジスタ (RTC\_CALR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(779 ページ\)](#) を参照してください。

アドレスオフセット : 0x3C

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
r/w	r/w	r/w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15 **CALP** : RTC 周波数を 488.5 ppm 増加

0 : RTCCLK パルスは加えられません。

1 : RTCCLK の  $2^{11}$  パルスごとに 1 パルス効果的に挿入されます (周波数が 488.5 ppm 増加)。

この機能は、CALM と共に使用されることを想定しており、カレンダーの周波数を高分解能で下げることができます。入力周波数が 32768 Hz の場合、32 秒枠の間に追加される RTCCLK パルスの数は次のように算出されます :  $(512 * CALP) - CALM$

[セクション 27.3.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 14 **CALW8** : 8 秒較正サイクル周期の使用

CALW8 が "1" にセットされると、8 秒較正サイクル周期が選択されます。

**注 :** **CALW8 = 1 の場合、CALM[1:0] は "00" に固定されます。** [セクション 27.3.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 13 **CALW16** : 16 秒較正サイクル周期の使用

CALW16 が "1" にセットされると、16 秒較正サイクル周期が選択されます。CALW8 = 1 の場合、このビットを "1" にセットすることはできません。

**注 :** **CALW16 = 1 の場合、CALM[0] は "0" に固定されます。** [セクション 27.3.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 12:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8:0 **CALM[8:0]** : 較正マイナス

RTCCLK  $2^{20}$  パルス (入力周波数が 32768 Hz の場合 32 秒) 内の CALM をマスクすることによって、カレンダーの周波数が下げられます。この方法により、カレンダーの周波数を 0.9537 ppm の分解能で下げることができます。

カレンダーの周波数を上げるには、この機能を CALP と共に使用する必要があります。[セクション 27.3.12 : RTC の高精度デジタル較正 \(783 ページ\)](#) を参照してください。

## 27.6.16 RTC タンパおよびオルタネート機能設定レジスタ (RTC\_TAFCR)

アドレスオフセット : 0x40

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC15 MODE	PC15 VALUE	PC14 MODE	PC14 VALUE	PC13 MODE	PC13 VALUE	Res.	Res.
								r/w	r/w	r/w	r/w	r/w	r/w		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMPP UDIS	TAMPPRCH [1:0]	TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPT S	TAMP3 TRG <sup>(1)</sup>	TAMP3 E <sup>(1)</sup>	TAMP2 TRG	TAMP2 E	TAMPIE	TAMP1 TRG	TAMP1 E	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w	r/w	r/w	

1. RTC\_TAMP3 は STM32F303x6/8 および STM32F328 デバイスでは使用できません。

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **PC15MODE** : PC15 モード

0 : PC15 は、GPIO 設定レジスタによって制御されます。そのため、PC15 は STANDBY モードでローティング状態になります。

1 : LSE が無効の場合、PC15 は強制的にプッシュプル出力に設定されます。

ビット 22 **PC15VALUE** : PC15 値

LSE が無効で PC15MODE = 1 の場合、PC15VALUE が PC15 出力データを設定します。

ビット 21 **PC14MODE** : PC14 モード

0 : PC14 は、GPIO 設定レジスタによって制御されます。そのため、PC14 は STANDBY モードでローティング状態になります。

1 : LSE が無効の場合、PC14 は強制的にプッシュプル出力に設定されます。

ビット 20 **PC14VALUE** : PC14 値

LSE が無効で PC14MODE = 1 の場合、PC14VALUE が PC14 出力データを設定します。

ビット 19 **PC13MODE** : PC13 モード

0 : PC13 は、GPIO 設定レジスタによって制御されます。そのため、PC13 は STANDBY モードでローティング状態になります。

1 : RTC オルタネート機能がすべて無効の場合、PC13 は強制的にプッシュプル出力に設定されます。

ビット 18 **PC13VALUE** : RTC\_ALARM 出力タイプ/PC13 値

PC13 を使用して RTC\_ALARM を出力した場合、PC13VALUE が出力設定を設定します。

0 : RTC\_ALARM はオープンドレイン出力になります。

1 : RTC\_ALARM はプッシュプル出力になります。

RTC 代替機能がすべて無効で PC13MODE = 1 の場合、PC13VALUE が PC13 出力データを設定します。

ビット 17:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **TAMPPUDIS** : RTC\_TAMPx プルアップディセーブル

このビットにより、毎回のサンプリング前に各 RTC\_TAMPx ピンをプリチャージするかどうか決定します。

0 : サンプリング前に RTC\_TAMPx ピンをプリチャージします (内部プルアップを有効化)。

1 : RTC\_TAMPx ピンのプリチャージを無効化します。



**ビット 14:13 TAMPPRCH[1:0] : RTC\_TAMPx プリチャージ時間**

これらのビットにより、各サンプリングの前にプルアップを有効化している時間を決定します。TAMPPRCH は、各 RTC\_TAMPx 入力に対して有効です。

- 0x0 : 1 RTCCLK サイクル
- 0x1 : 2 RTCCLK サイクル
- 0x2 : 4 RTCCLK サイクル
- 0x3 : 8 RTCCLK サイクル

**ビット 12:11 TAMPFLT[1:0] : RTC\_TAMPx フィルタカウント**

これらのビットにより、タンパイベントをアクティブにするのに必要な指定のレベル (TAMP\*TRG) での連続サンプリングの数を決定します。TAMPFLT は、各 RTC\_TAMPx 入力に対して有効です。

- 0x0 : RTC\_TAMPx 入力 that アクティブレベル (RTC\_TAMPx 入力における内部プルアップ無し) に遷移するときのエッジで、タンパイベントがアクティブになります。
- 0x1 : アクティブレベルでの連続した 2 回のサンプリングの後、タンパイベントがアクティブになります。
- 0x2 : アクティブレベルでの連続した 4 回のサンプリングの後、タンパイベントがアクティブになります。
- 0x3 : アクティブレベルでの連続した 8 回のサンプリングの後、タンパイベントがアクティブになります。

**ビット 10:8 TAMPFREQ[2:0] : タンパサンプリング周波数**

各 RTC\_TAMPx 入力 that サンプリングされる周波数を決定します。

- 0x0 : RTCCLK / 32768 (RTCCLK = 32768 Hz の場合 1 Hz)
- 0x1 : RTCCLK / 16384 (RTCCLK = 32768 Hz の場合 2 Hz)
- 0x2 : RTCCLK / 8192 (RTCCLK = 32768 Hz の場合 4 Hz)
- 0x3 : RTCCLK / 4096 (RTCCLK = 32768 Hz の場合 8 Hz)
- 0x4 : RTCCLK / 2048 (RTCCLK = 32768 Hz の場合 16 Hz)
- 0x5 : RTCCLK / 1024 (RTCCLK = 32768 Hz の場合 32 Hz)
- 0x6 : RTCCLK / 512 (RTCCLK = 32768 Hz の場合 64 Hz)
- 0x7 : RTCCLK / 256 (RTCCLK = 32768 Hz の場合 128 Hz)

**ビット 7 TAMPTS : タンパ検出イベント時のタイムスタンプの有効化**

- 0 : タンパ検出イベントがあっても、タイムスタンプは保存されません。
- 1 : タンパ検出イベント時、タイムスタンプが保存されます。

RTC\_CR レジスタで TSE = 0 であっても TAMPTS は有効です。

**ビット 6 TAMP3TRG : RTC\_TAMP3 入力のアクティブレベル**

TAMPFLT != 00 の場合

- 0 : RTC\_TAMP3 入力 that ローのままのとき、タンパ検出イベント that トリガされます。
- 1 : RTC\_TAMP3 入力 that ハイのままのとき、タンパ検出イベント that トリガされます。

TAMPFLT = 00 の場合

- 0 : RTC\_TAMP3 入力の立ち上がりエッジ with タンパ検出イベント that トリガされます。
- 1 : RTC\_TAMP3input の立ち下りエッジ with タンパ検出イベント that トリガされます。

**ビット 5 TAMP3E : RTC\_TAMP3 検出イネーブル**

- 0 : RTC\_TAMP3 入力検出は無効です。
- 1 : RTC\_TAMP3 入力検出は有効です。

**ビット 4 TAMP2TRG : RTC\_TAMP2 入力のアクティブレベル**

TAMPFLT != 00 の場合

- 0 : RTC\_TAMP2 入力 that ローのままのとき、タンパ検出イベント that トリガされます。
- 1 : RTC\_TAMP2 入力 that ハイのままのとき、タンパ検出イベント that トリガされます。

TAMPFLT = 00 の場合

- 0 : RTC\_TAMP2 入力の立ち上がりエッジ with タンパ検出イベント that トリガされます。
- 1 : RTC\_TAMP2 入力の立ち下りエッジ with タンパ検出イベント that トリガされます。

**ビット 3 TAMP2E : RTC\_TAMP2 入力検出イネーブル**

- 0 : RTC\_TAMP2 検出は無効です。
- 1 : RTC\_TAMP2 検出は有効です。



ビット 2 **TAMPIE** : タンパ割り込みイネーブル

- 0 : タンパ割り込みは無効です。
- 1 : タンパ割り込みは有効です。

ビット 1 **TAMP1TRG** : RTC\_TAMP1 入力のアクティブレベル

TAMPFLT != 00 の場合

- 0 : RTC\_TAMP1 入力がローのままのとき、タンパ検出イベントがトリガされます。
- 1 : RTC\_TAMP1 入力がハイのままのとき、タンパ検出イベントがトリガされます。

TAMPFLT = 00 の場合

- 0 : RTC\_TAMP1 入力の立ち上がりエッジでタンパ検出イベントがトリガされます。
- 1 : RTC\_TAMP1 入力の立ち下がりエッジでタンパ検出イベントがトリガされます。

ビット 0 **TAMP1E** : RTC\_TAMP1 入力検出イネーブル

- 0 : RTC\_TAMP1 検出は無効です。
- 1 : RTC\_TAMP1 検出は有効です。

**注意 :** **TAMPFLT = 0 の場合、TAMP1E ビットは、TAMP1TRG が変更されたらリセットし、TAMP1F の不要なセットを回避する必要があります。**

## 27.6.17 RTC アラーム A サブセカンドレジスタ (RTC\_ALRMASR)

このレジスタは、RTC\_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込み可能です。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護を参照してください。[RTC レジスタ書き込み保護 \(779 ページ\)](#)

アドレスオフセット : 0x44

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 MASKSS[3:0] : このビットから始まる最上位ビットのマスク

0 : アラーム A に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

1 : アラーム A の比較では SS[14:1] を無視します。SS[0] のみ比較されます。

2 : アラーム A の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。

3 : アラーム A の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。

...

12 : アラーム A の比較では SS[14:12] を無視します。SS[11:0] が比較されます。

13 : アラーム A の比較では SS[14:13] を無視します。SS[12:0] が比較されます。

14 : アラーム A の比較では SS[14] を無視します。SS[13:0] が比較されます。

15 : アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。

同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 SS[14:0] : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

## 27.6.18 RTC アラーム B サブセカンドレジスタ (RTC\_ALRMBSSR)

このレジスタは、RTC\_CR レジスタの ALRBE がリセットされた場合、または初期化モードの場合にのみ書き込みめます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[セクション : RTC レジスタ書き込み保護](#)を参照してください。

アドレスオフセット : 0x48

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **MASKSS[3:0]** : このビットから始まる最上位ビットのマスク

0x0 : アラーム B に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。

0x1 : アラーム B の比較では SS[14:1] を無視します。SS[0] のみ比較されます。

0x2 : アラーム B の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。

0x3 : アラーム B の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。

...

0xC : アラーム B の比較では SS[14:12] を無視します。SS[11:0] が比較されます。

0xD : アラーム B の比較では SS[14:13] を無視します。SS[12:0] が比較されます。

0xE : アラーム B の比較では SS[14] を無視します。SS[13:0] が比較されます。

0xF : アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。

同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **SS[14:0]** : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム B をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

## 27.6.19 RTC バックアップレジスタ (RTC\_BKPxR)

アドレスオフセット : 0x50 から 0x8C

バックアップ ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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



ビット 31:0 BKP[31:0]

アプリケーションはこれらのレジスタに対してデータの読み書きをすることができます。

これらのレジスタは、V<sub>DD</sub> がオフになった場合、V<sub>BAT</sub> によって電源が供給されるため、システムリセットによりリセットされず、デバイスが低電力モードで動作する場合、レジスタの内容は有効なまま保持されます。

このレジスタは、TAMPxP = 1 である限り、あるいはフラッシュ読み出し保護が無効なときは、タンパ検出イベント時にリセットされます。

### 27.6.20 RTC レジスタマップ

表 140. RTC レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	RTC_TR	Res	Res	Res	Res	Res	Res	Res	Res	Res	PM	HT [1:0]	HU[3:0]			Res	MNT[2:0]			MNU[3:0]			Res	ST[2:0]			SU[3:0]										
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	RTC_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	YT[3:0]			YU[3:0]			WDU[2:0]		MT	MU[3:0]			Res	Res	DT [1:0]			DU[3:0]									
	リセット値										0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1		0	0	0	0	0	1			
0x08	RTC_CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	COE	OSE L [1:0]	POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]						
	リセット値										0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x0C	RTC_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RECALPF	TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTF	ALRBF	ALRAF				
	リセット値																0		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1		
0x10	RTC_PRER	Res	Res	Res	Res	Res	Res	Res	Res	Res	PREDIV_A[6:0]						PREDIV_S[14:0]																				
	リセット値										1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
0x14	RTC_WUTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUT[15:0]																			
	リセット値																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x1C	RTC_ALRMAR	MSK4	WDSSEL	DT [1:0]	DU[3:0]			MSK3	PM	HT [1:0]	HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK1	ST[2:0]		SU[3:0]														
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x20	RTC_ALRMBR	MSK4	WDSSEL	DT [1:0]	DU[3:0]			MSK3	PM	HT [1:0]	HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK2	ST[2:0]		SU[3:0]														
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x24	RTC_WPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY										
	リセット値																										0	0	0	0	0	0	0	0	0	0	
0x28	RTC_SSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SS[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x2C	RTC_SHIFTR	ADD1S	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SUBFS[14:0]																		
	リセット値	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

表 140. RTC レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	RTC_TSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]	HU[3:0]			Res.	MNT[2:0]			MNU[3:0]			Res.	ST[2:0]		SU[3:0]							
	リセット値										0	0	0	0	0	0	0										0	0	0	0	0	0	0
0x34	RTC_TSDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[1:0]		MT	MU[3:0]			Res.	Res.	DT[1:0]		DU[3:0]					
	リセット値																		0	0	0	0					0	0	0	0	0	0	0
0x38	RTC_TSSSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	RTC_CALR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
	リセット値																	0	0	0					0	0	0	0	0	0	0	0	0
0x40	RTC_TAFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC15MODE	PC15MODE	PC14VALUE	PC14MODE	PC13VALUE	PC13VALUE	Res.	Res.	TAMPPUDIS	TAMPPRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]		TAMPTS	TAMP3TRG	TAMP9E	TAMP2TRG	TAMP2E	TAMP1E	TAMP1TRG	TAMP1E
	リセット値										0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	RTC_ALRMASR	Res.	Res.	Res.	Res.	MASKSS [3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]													
	リセット値					0	0	0	0											0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	RTC_ALRMBSR	Res.	Res.	Res.	Res.	MASKSS [3:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]													
	リセット値					0	0	0	0											0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50 to 0x8C	RTC_BKP0R	BKP[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	to RTC_BKP15R	BKP[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



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

## 28 I2C (Inter-integrated circuit) インタフェース

### 28.1 概要

I<sup>2</sup>C (Inter-integrated circuit) バスインタフェースは、マイクロコントローラとシリアル I<sup>2</sup>C バス間の通信を処理します。マルチマスタ機能を備え、すべての I<sup>2</sup>C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準モード (Sm)、高速モード (Fm)、および高速モードプラス (Fm+) をサポートします。

また、SMBus (System Management Bus) および PMBus (Power Management Bus) と互換性があります。

DMA を使用して、CPU の負荷を軽減できます。

### 28.2 I2C の主な機能

- I<sup>2</sup>C バス仕様 rev03 との互換性：
  - スレーブおよびマスタモード
  - マルチマスタ機能
  - 標準モード (最大 100 kHz)
  - 高速モード (最大 400 kHz)
  - 高速モードプラス (最大 1 MHz)
  - 7ビットおよび 10 ビットアドレッシングモード
  - 複数の 7 ビットスレーブアドレス (2 つのアドレス、1 つは設定可能なマスク付き)
  - すべての 7 ビットアドレス確認応答モード
  - 同報 (General call) コール
  - プログラム可能なセットアップおよびホールド時間
  - 使いやすいイベント管理
  - クロックストレッチオプション
  - ソフトウェアリセット
- DMA 機能付きの 1 バイトバッファ
- プログラム可能なアナログおよびデジタルノイズフィルタ

製品の実装によっては、次の追加機能も使用できます ([セクション 28.3 : I2C の実装](#) を参照) :

- SMBus 仕様 rev 2.0 との互換性：
  - ハードウェア PEC (Packet Error Checking) の生成と ACK 制御による確認
  - コマンドおよびデータ確認応答制御
  - アドレス解決プロトコル (ARP) サポート
  - ホストおよびデバイスのサポート
  - SMBus アラート
  - タイムアウトおよびアイドル条件の検出
- PMBus rev 1.1 標準との互換性
- 独立したクロック : 独立したクロックソースの選択により、I2C の通信速度は PCLK の再プログラミングから独立
- アドレス一致時に STOP モードからウェイクアップ

## 28.3 I2C の実装

このマニュアルでは、I2C1、I2C2、および I2C3 に実装されているすべての機能について説明しています。STM32F3xx デバイスでは、I2C1、I2C2、および I2C3 (STM32F303xD/E) は同一で、次の表に示すように、すべての機能を実装しています。

表 141. STM32F3xxI2C の実装

I2C の機能 <sup>(1)</sup>	I2C1	I2C2 <sup>(2)</sup>	I2C3 <sup>(3)</sup>
7 ビットアドレスモード	X	X	X
10 ビットアドレスモード	X	X	X
標準モード (最大 100 kbit/s)	X	X	X
高速モード (最大 400 kbit/s)	X	X	X
20mA 出力駆動 I/O 搭載高速モードプラス (最大 1 Mbit/s)	X	X	X
独立クロック	X	X	X
SMBus	X	X	X
STOP モードからのウェイクアップ	X	X	X

1. X: サポートされています。
2. I2C2 は使用可能です。
3. I2C3 は STM32F303xD/E でのみ使用可能です。

## 28.4 I2C の機能詳細

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式 (およびその逆) に変換します。割り込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン (SDA) とクロックピン (SCL) によって I2C バスに接続されます。標準 (最大 100 kHz)、高速モード (最大 400 kHz)、または高速モードプラス (最大 1 MHz) の I<sup>2</sup>C バスで接続できます。

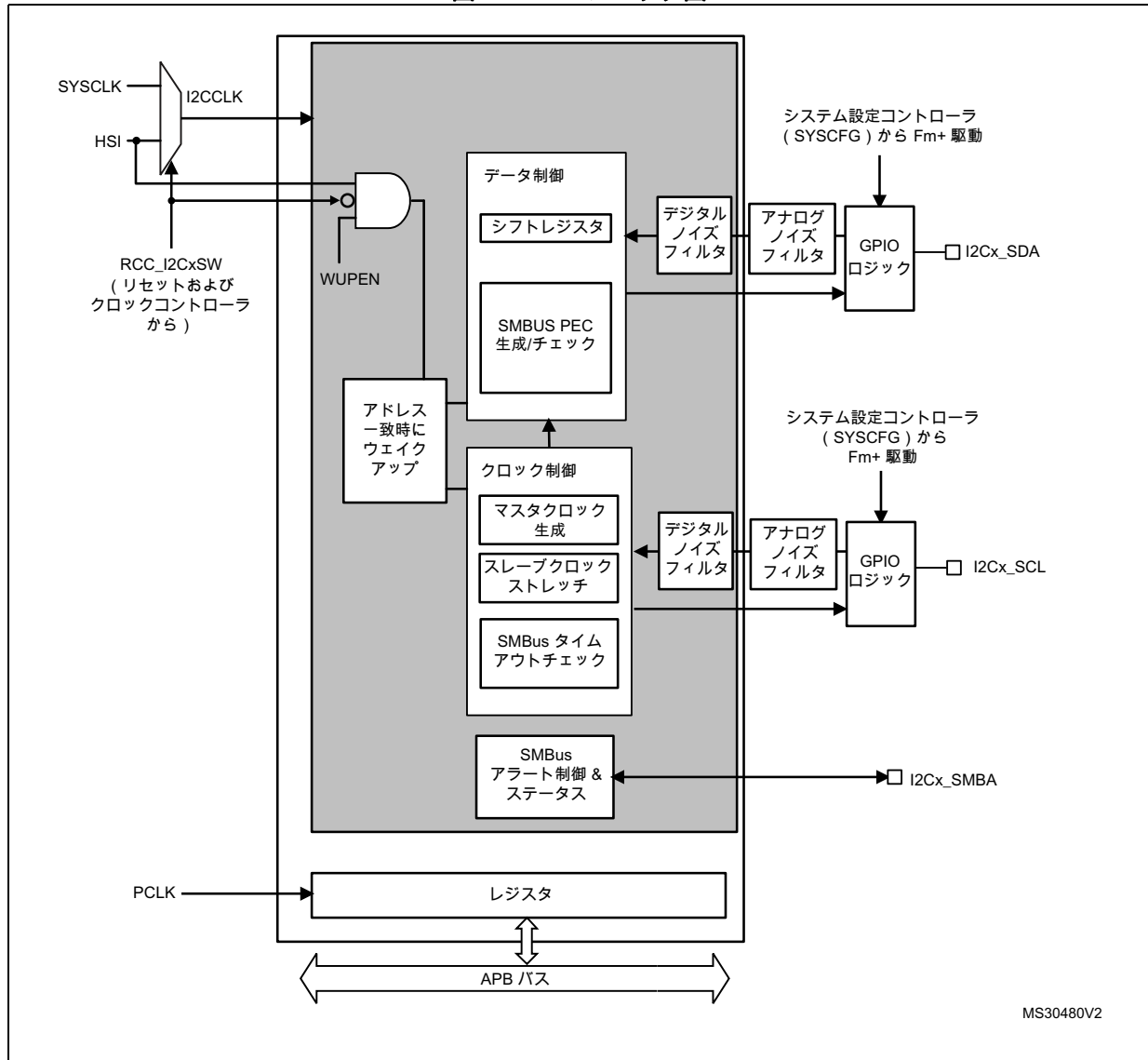
このインタフェースは、データピン (SDA) とクロックピン (SCL) によって SMBus に接続することもできます。

SMBus 機能がサポートされる場合、追加の SMBus アラートピン (SMBA) オプションも使用できます。

### 28.4.1 I2C ブロック図

I2C インタフェースのブロック図を [図 292](#) に示します。

図 292. I2C ブロック図



I2C は、独立したクロックソースによってクロック供給されるため、I2C は PCLK 周波数から独立して動作できます。

この独立したクロックソースは、次の 2 つのクロックソースから選択できます：

- HSI : 高速内部オシレータ (デフォルト値)
- SYSCLK : システムクロック

詳細については、[セクション 9 : リセットおよびクロック制御 \(RCC\)](#) を参照してください。

I2C I/O は、高速モードプラス動作のための 20 mA 出力電流駆動をサポートします。これを有効にするには、[セクション 12.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG\\_CFGR1\)](#) の SCL および SDA の駆動機能制御ビットをセットします。



## 28.4.2 I2C クロックの要件

I2C カーネルは I2CCLK によってクロック供給されます。

I2CCLK の周期  $t_{I2CCLK}$  は、次の条件を満たす必要があります。

$$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4 \text{ and } t_{I2CCLK} < t_{HIGH}$$

ここで：

$t_{LOW}$  : SCL ロー時間、および  $t_{HIGH}$  : SCL ハイ時間

$t_{filters}$  : 有効なときには、アナログフィルタとデジタルフィルタによる遅延の合計。

アナログフィルタの遅延は、最大 260 ns です。デジタルフィルタの遅延は、 $DNF \times t_{I2CCLK}$  です。

PCLK のクロック周期  $t_{PCLK}$  は、次の条件を満たす必要があります。

$$t_{PCLK} < 4/3 t_{SCL}$$

$t_{SCL}$  : SCL 周期

**注意：** I2C カーネルが PCLK によってクロック供給されると、PCLK は  $t_{I2CCLK}$  の条件を満たす必要があります。

## 28.4.3 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます：

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成したときにはスレーブからマスタへ、アービトレーションの喪失または STOP 生成が発生したときにはマスタからスレーブへ自動的に切り替わるため、マルチマスタ機能を使用できます。

### 通信の流れ

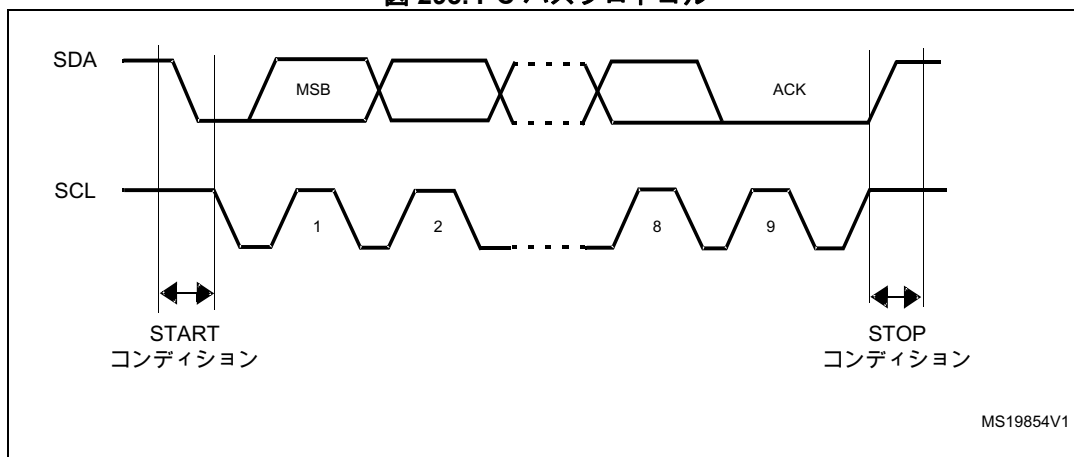
マスタモードでは、I2C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス (7 または 10 ビット) と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効または無効にできます。予約済みの SMBus アドレスもソフトウェアによって有効にできます。

データとアドレスは、MSB ファーストの 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト (7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト) にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります。次の図を参照してください。

図 293. I<sup>2</sup>C バスプロトコル



確認応答 (Acknowledge) は、ソフトウェアによって有効または無効にできます。I2C インタフェースのアドレスは、ソフトウェアによって選択できます。

## 28.4.4 I2C の初期化

### ペリフェラルの有効化と無効化

I2C ペリフェラルクロックは、クロックコントローラで設定し、有効にする必要があります ([セクション 9: リセットおよびクロック制御 \(RCC\)](#) を参照)。

そして、I2C\_CR1 レジスタの PE ビットをセットすることによって、I2C を有効にできます。

I2C が無効なときには (PE=0)、I<sup>2</sup>C はソフトウェアリセットを実行します。詳細については、[セクション 28.4.5: ソフトウェアリセット](#) を参照してください。

### ノイズフィルタ

I2C\_CR1 レジスタの PE ビットをセットすることによって I2C ペリフェラルを有効にする前に、必要な場合は、ノイズフィルタを設定する必要があります。デフォルトでは、SDA および SCL 入力にアナログノイズフィルタがあります。このアナログフィルタは I<sup>2</sup>C 仕様に準拠しており、高速モードおよび高速モードプラスで最大 50 ns のパルス幅を持つスパイクを抑制します。ANFOFF ビットをセットすることによって、このアナログフィルタを無効にし、I2C\_CR1 レジスタの DNF[3:0] ビットを設定することによってデジタルフィルタを選択することができます。

デジタルフィルタが有効なときには、SCL または SDA ラインのレベルは、DNF x I2CCLK 周期より長く安定していた場合のみ、内部で変更されます。これにより、プログラム可能な 1 ~ 15 I2CCLK 周期の長さを持つスパイクを抑制できます。

表 142. アナログフィルタとデジタルフィルタの比較

	アナログフィルタ	デジタルフィルタ
抑制されるスパイクのパルス幅	≥ 50 ns	長さを 1 ~ 15 I2C ペリフェラルクロックにプログラム可能
利点	STOP モードで使用可能	<ul style="list-style-type: none"> <li>- プログラム可能な長さ: 追加のフィルタリング機能対標準要件</li> <li>- 安定した長さ</li> </ul>
欠点	温度、電圧、プロセスのばらつき	デジタルフィルタが有効なときには、アドレス一致時の STOP モードからのウェイクアップは使用できない

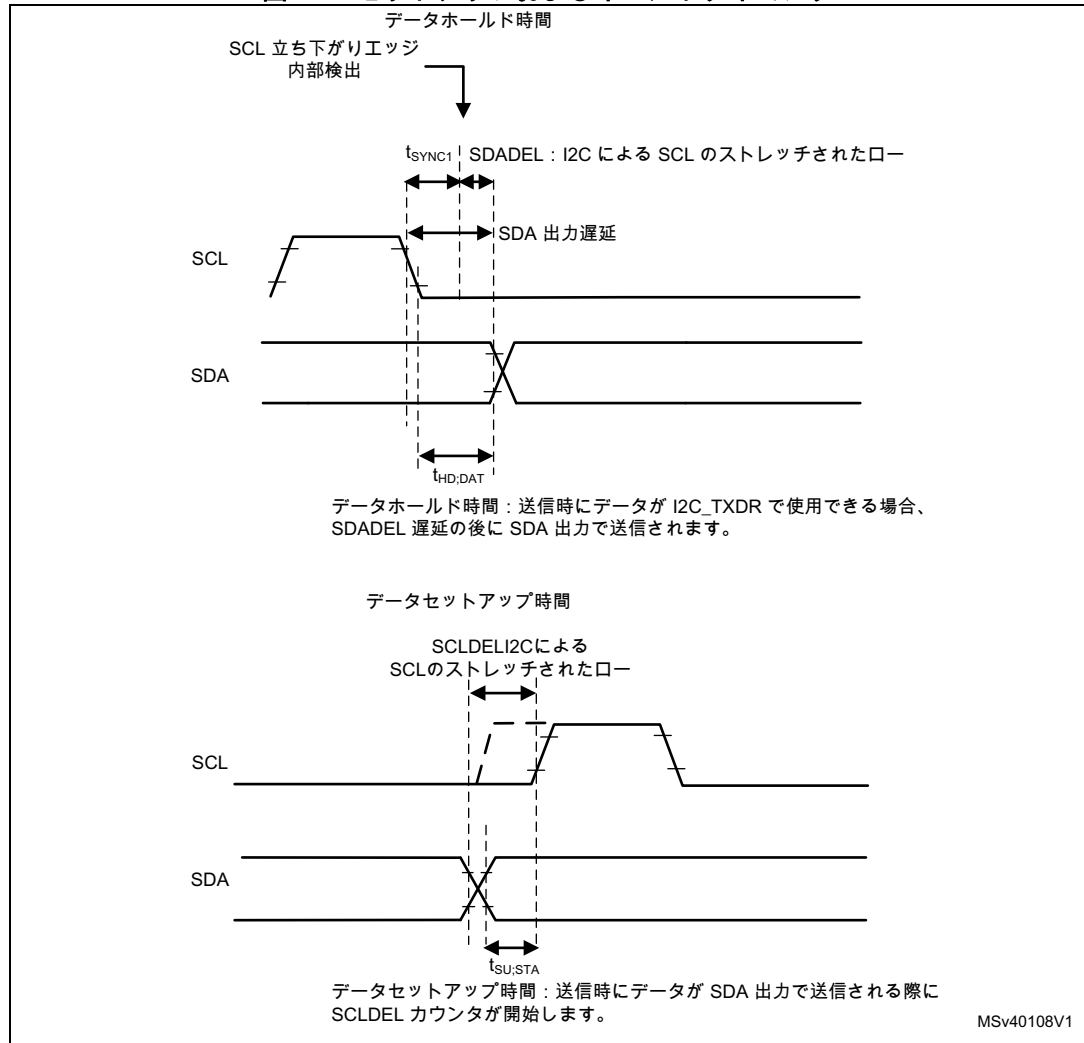
**注意:** I2C が有効なときには、フィルタ構成の変更はできません。

## I2C のタイミング

マスタおよびスレーブモードで正しいデータホールドおよびセットアップ時間が使用されるのを保証するためには、タイミングを設定する必要があります。これを行うには、I2C\_TIMINGR レジスタの PRESC[3:0]、SCLDEL[3:0]、および SDADEL[3:0] ビットをプログラムします。

STM32CubeMX ツールは、I2C 設定ウィンドウの I2C\_TIMINGR コンテンツを計算し、提供します。

図 294. セットアップおよびホールドタイミング



- SCL 立ち下がりエッジが内部で検出されると、SDA 出力を送信する前に遅延が挿入されます。この遅延は、 $t_{SDADEL} = SDADEL \times t_{PRESC} + t_{I2CCCLK}$  であり  $t_{PRESC} = (PRESC+1) \times t_{I2CCCLK}$  です。 $t_{SDADEL}$  はホールド時間  $t_{HD;DAT}$  に影響を与えます。

SDA 出力遅延の合計は、次のとおりです：

$$t_{SYNC1} + \{[SDADEL \times (PRESC+1) + 1] \times t_{I2CCCLK}\}$$

$t_{SYNC1}$  の長さは、次のパラメータに依存します。

- SCL 立ち下がり傾斜
- アナログフィルタが有効なときの入力遅延： $t_{AF(min)} < t_{AF} < t_{AF(max)}$  ns
- デジタルフィルタが有効なときの入力遅延： $t_{DNF} = DNF \times t_{I2CCCLK}$
- SCL と I2CCCLK クロックの同期による遅延 (2 ~ 3 I2CCCLK 周期)

SCL 立ち下がりエッジの未定義の領域をブリッジするためには、SDADEL を次のようにプログラムする必要があります：

$$\{t_{r(max)} + t_{HD;DAT(min)} - t_{AF(min)} - [(DNF+3) \times t_{I2CCCLK}]\} / \{(PRESC+1) \times t_{I2CCCLK}\} \leq SDADEL$$

$$SDADEL \leq \{t_{HD;DAT(max)} - t_{AF(max)} - [(DNF+4) \times t_{I2CCCLK}]\} / \{(PRESC+1) \times t_{I2CCCLK}\}$$

**注：**  $t_{AF(min)} / t_{AF(max)}$  は、アナログフィルタが有効なときのみ、等式に含まれます。 $t_{AF}$  の値については、デバイスのデータシートを参照してください。

最大  $t_{HD;DAT}$  は、標準モード、高速モード、および高速モードプラスで 3.45  $\mu$ s、0.9  $\mu$ s、および 0.45  $\mu$ s ですが、遷移時間による  $t_{VD;DAT}$  の最大値より短い必要があります。この最大値を満たす必要があるのは、デバイスが SCL 信号の LOW 周期 ( $t_{LOW}$ ) をストレッチしない場合だけです。クロックが SCL をストレッチする場合、クロックをリリースする前に、データがセットアップ時間まで有効である必要があります。

SDA 立ち上がりエッジは、通常、最悪ケースであり、この場合、前の等式は次のようになります：

$$SDADEL \leq \{t_{VD;DAT(max)} - t_r(max) - 260 \text{ ns} - [(DNF+4) \times t_{I2CCCLK}]\} / \{(PRESC+1) \times t_{I2CCCLK}\}$$

**注：** **NOSTRETCH=0 のときには、SCLDEL の値に従って、デバイスはセットアップ時間を保証するために SCL ローをストレッチするので、この条件に違反することがあります。**

$t_r$ 、 $t_r$ 、 $t_{HD;DAT}$ 、および  $t_{VD;DAT}$  の標準値については、表 143 : I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を参照してください。

- I2C\_TXDR レジスタにデータがまだ書き込まれていないためスレーブがクロックをストレッチする必要があった場合に、 $t_{SDADEL}$  遅延の後、または SDA 出力の送信後、SCL ラインはセットアップ時間中、ローレベルに保たれます。このセットアップ時間は、 $t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$  であり  $t_{PRESC} = (PRESC+1) \times t_{I2CCCLK}$  です。

$t_{SCLDEL}$  は、セットアップ時間  $t_{SU;DAT}$  に影響を与えます。

SDA 遷移 (立ち上がりエッジは通常、最悪のケース) の未定義の領域をブリッジするためには、SCLDEL を次のようにプログラムする必要があります：

$$\{[t_r(max) + t_{SU;DAT(min)}] / [(PRESC+1)] \times t_{I2CCCLK}\} - 1 \leq SCLDEL$$

$t_r$  および  $t_{SU;DAT}$  の標準値については、表 143 : I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を参照してください。

使用される SDA および SCL 遷移時間の値は、アプリケーションの値です。標準から最大値を使用すると、SDADEL と SCLDEL の計算の制約が増えますが、アプリケーションにかかわらず、この機能を使用できます。

**注：** 各クロックパルスで、SCL 立ち下がリエッジの検出後、I2C マスタまたはスレーブは、送信モードおよび受信モード両方で、最低  $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCCLK}$  の間、SCL ローをストレッチします。送信モードでは、SDADEL カウンタが終了したときに I2C\_TXDR にデータがまだ書き込まれていない場合、I2C は次のデータが書き込まれるまで SCL ローのストレッチを続けます。その後、新しいデータ MSB が SDA 出力で送信され、SCLDEL カウンタが開始し、SCL ローのストレッチを続け、データセットアップ時間を保証します。

スレーブモードで NOSTRETCH=1 の場合、SCL はストレッチされません。そのため、SDADEL はこのようにしてプログラムし、十分なセットアップ時間を保証する必要があります。

表 143. I<sup>2</sup>C-SMBUS 仕様のデータのセットアップおよびホールド時間

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
t <sub>HD;DAT</sub>	データホールド時間	0	-	0	-	0	-	0.3	-	μs
t <sub>VD;DAT</sub>	データ有効時間	-	3.45	-	0.9	-	0.45	-	-	
t <sub>SU;DAT</sub>	データセットアップ時間	250	-	100	-	50	-	250	-	ns
t <sub>r</sub>	SDA および SCL 信号の立ち上がり時間	-	1000	-	300	-	120	-	1000	
t <sub>f</sub>	SDA および SCL 信号の立ち下がり時間	-	300	-	300	-	120	-	300	

また、マスタモードでは、I2C\_TIMINGR レジスタの PRESC[3:0]、SCLH[7:0]、および SCLL[7:0] ビットをプログラムすることによって、SCL クロックのハイおよびローレベルを設定する必要があります。

- SCL 立ち下がリエッジが内部で検出されると、SCL 出力をリリースする前に遅延が挿入されます。この遅延は、 $t_{SCLL} = (SCLL+1) \times t_{PRESC}$  であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCCLK}$  です。  
t<sub>SCLL</sub> は、SCL ロー時間 t<sub>LOW</sub> に影響を与えます。
- SCL 立ち上がりエッジが内部で検出されると、SCL 出力を強制的にローレベルにする前に遅延が挿入されます。この遅延は、 $t_{SCLH} = (SCLH+1) \times t_{PRESC}$  であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCCLK}$  です。t<sub>SCLH</sub> は、SCLハイ時間 t<sub>HIGH</sub> に影響を与えます。

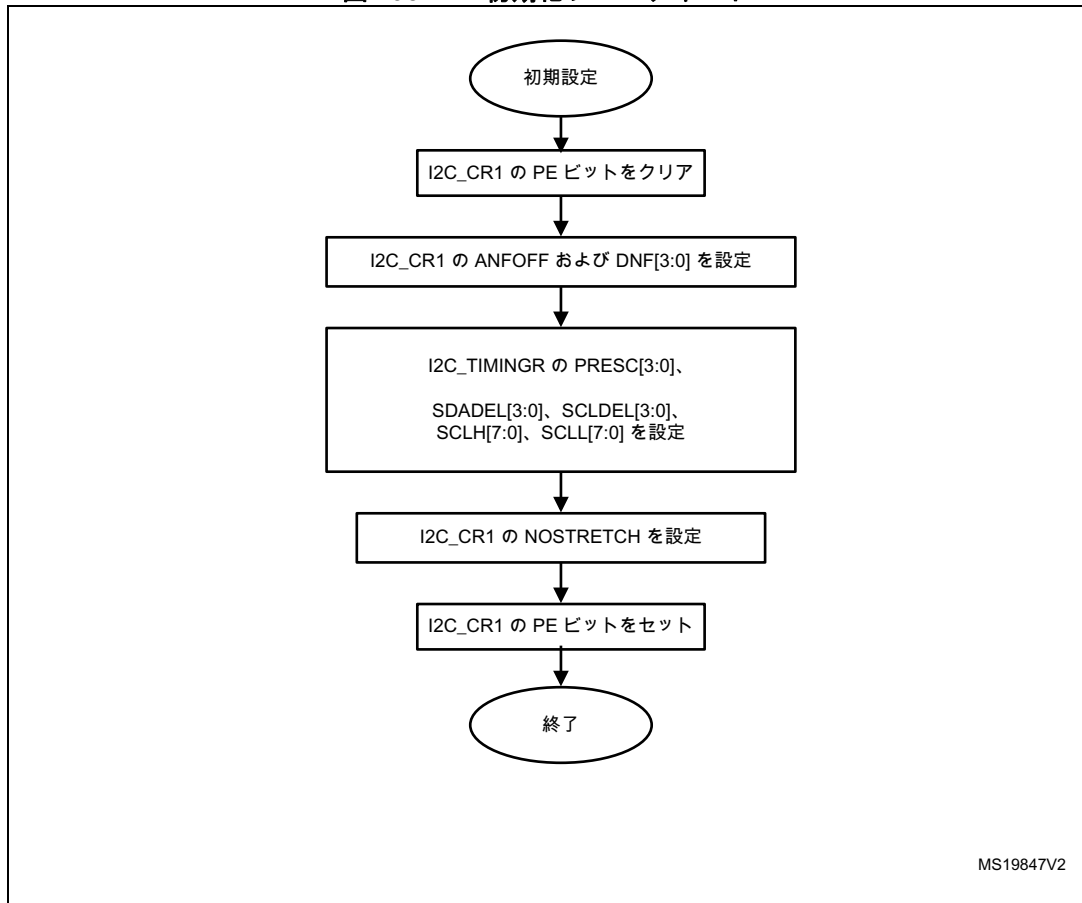
詳細については、[I2C マスタ初期化](#) を参照してください。

**注意：** I2C が有効なときには、[タイミング構成の設定はできません](#)。

ペリフェラルを有効にする前に、I2C スレーブ NOSTRETCH モードも設定する必要があります。詳細については、[I2C スレーブ初期化](#) を参照してください。

**注意：** I2C が有効なときには、[NOSTRETCH 構成の変更はできません](#)。

図 295. I2C 初期化フローチャート



## 28.4.5 ソフトウェアリセット

ソフトウェアリセットを行うには、I2C\_CR1 レジスタの PE ビットをクリアします。その場合、I2C のライン SCL および SDA がリリースされます。内部状態マシンがリセットされ、通信制御ビットとステータスビットがリセット値に戻ります。構成レジスタは影響を受けません。

影響を受けるレジスタのビットは、以下のとおりです：

1. I2C\_CR2 レジスタ：START、STOP、NACK
2. I2C\_ISR レジスタ：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO、OVR

SMBus 機能がサポートされるときには、以下も影響を受けます：

1. I2C\_CR2 レジスタ：PECBYTE
2. I2C\_ISR レジスタ：PECERR、TIMEOUT、ALERT

ソフトウェアリセットを実行するためには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たなければならない。このためには、次のソフトウェアシーケンスを書き込みます：- PE=0 を書き込む - PE=0 を確認する - PE=1 を書き込む

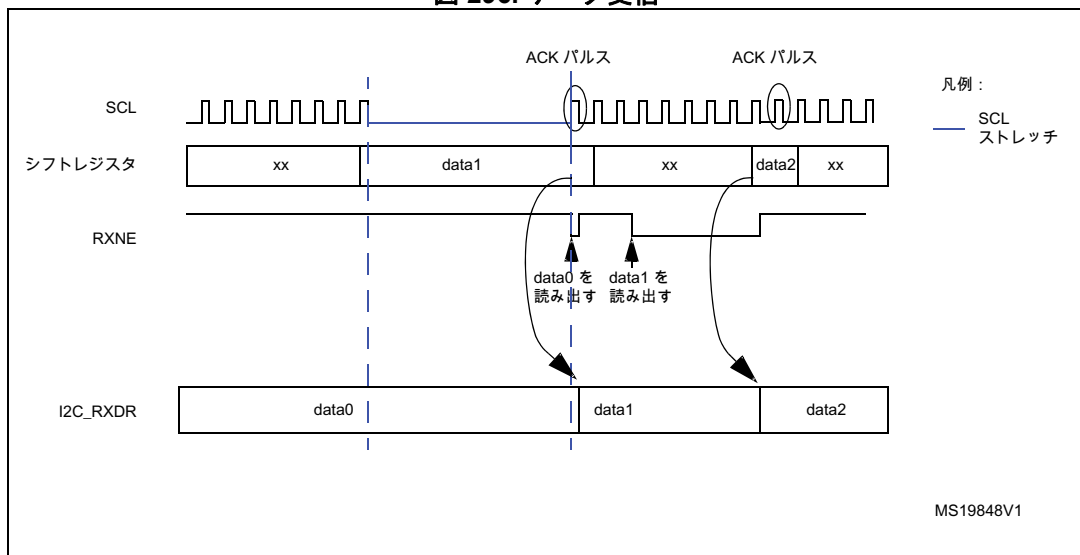
### 28.4.6 データ転送

データ転送は、送受信データレジスタとシフトレジスタを通じて管理されます。

#### 受信

SDA 入力はシフトレジスタに送られます。8 番目の SCL パルスの後 (完全なデータバイトの受信後)、シフトレジスタは、I2C\_RXDR レジスタが空の場合 (RXNE=0)、このレジスタにコピーされます。RXNE=1 の場合、すなわち、前に受信されたデータバイトがまだ読み出されていなかった場合、SCL ラインは I2C\_RXDR が読み出されるまでストレッチされます。ストレッチは、8 番目と 9 番目の SCL パルスの間 (確認応答パルスの前) に挿入されます。

図 296. データ受信

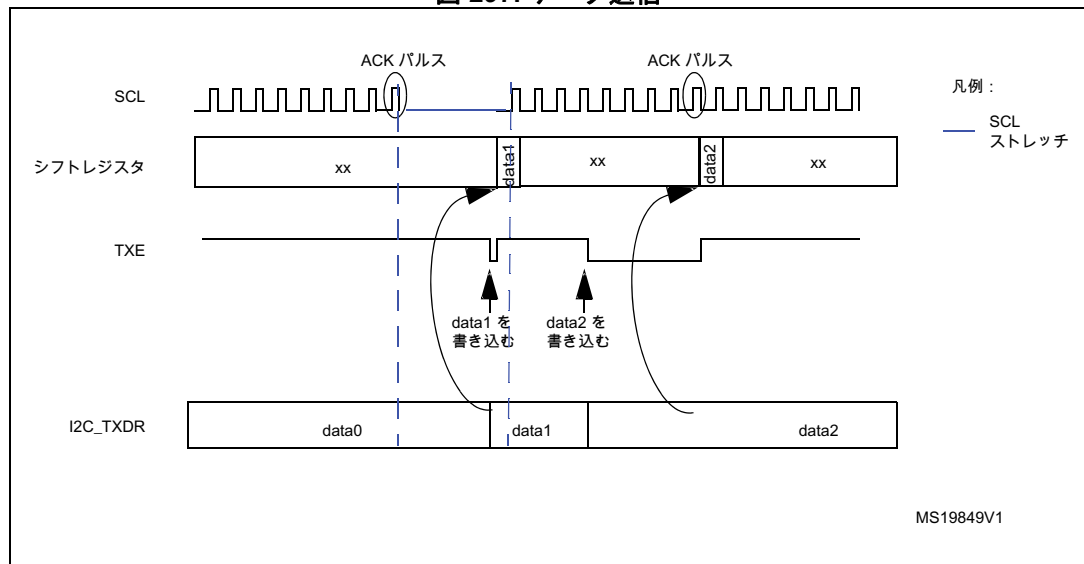




## 送信

I2C\_TXDR レジスタが空 (TXE=0) でない場合、その内容が 9 番目の SCL パルス (確認応答パルス) の後、シフトレジスタにコピーされます。次に、シフトレジスタの内容が SDA ラインにシフトアウトされます。TXE=1 の場合、すなわち、I2C\_TXDR にデータがまだ書き込まれていない場合、SCL ラインは I2C\_TXDR に書き込まれるまでストレッチされます。ストレッチは、9 番目の SCL パルスの後で行われます。

図 297. データ送信



## ハードウェア転送管理

次のようにさまざまなモードでバイト転送を管理し、通信をクローズするために、I2C にはハードウェアにバイトカウンタが組み込まれています。

- マスタモードでの NACK、STOP、および ReSTART 生成
- スレーブレシーバモードでの ACK 制御
- SMBus 機能がサポートされているときの PEC 生成/確認

バイトカウンタは、マスタモードでは常に使用されます。デフォルトでは、スレーブモードでは無効ですが、I2C\_CR2 レジスタの SBC (スレーブバイト制御) ビットをセットすることによって、ソフトウェアにより有効にできます。

転送されるバイト数は、I2C\_CR2 レジスタの NBYTES[7:0] ビットフィールドでプログラムされます。転送バイト数 (NBYTES) が 255 より大きい場合、またはレシーバが受信データバイトの確認応答値を制御したい場合には、I2C\_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。このモードでは、NBYTES でプログラムされたバイト数が転送されると、TCR フラグがセットされ、TCIE がセットされている場合は割り込みが生成されます。SCL は、TCR フラグがセットされている間、ストレッチされます。TCR は、NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

NBYTES カウンタに最後のバイト数が再ロードされたときには、RELOAD ビットがクリアされる必要があります。

マスタモードで RELOAD=0 のときには、カウンタは 2 つのモードで使用できます：

- **自動終了モード** (I2C\_CR2 レジスタの AUTOEND = "1")。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、マスタは STOP コンディションを自動的に送信します。
- **ソフトウェア終了モード** (I2C\_CR2 レジスタの AUTOEND = 0)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、ソフトウェアアクションが求められます。TC フラグがセットされ、TCIE ビットがセットされている場合は割り込みが生成されます。SCL 信号は、TC フラグがセットされている間、ストレッチされます。TC フラグは、I2C\_CR2 レジスタの START または STOP ビットがセットされたときに、ソフトウェアによってクリアされます。マスタが RESTART コンディションを送信したいときには、このモードを使用する必要があります。

**注意：** AUTOEND ビットは、RELOAD ビットがセットされているときには効果がありません。

表 144. I2C 設定表

機能	SBC ビット	RELOAD ビット	AUTOEND ビット
マスタ Tx/Rx NBYTES + STOP	x	0	1
マスタ Tx/Rx + NBYTES + RESTART	x	0	0
スレーブ Tx/Rx すべての受信バイトに ACK	0	x	x
スレーブ Rx および ACK 制御	1	1	x

## 28.4.7 I2C スレーブモード

### I2C スレーブ初期化

スレーブモードで動作するには、少なくとも 1 つのスレーブアドレスを有効にする必要があります。2 つのレジスタ I2C\_OAR1 と I2C\_OAR2 を使用して、スレーブ専用アドレス OA1 および OA2 をプログラムできます。

- OA1 は、I2C\_OAR1 レジスタの OA1MODE ビットをセットすることによって、7 ビットモード (デフォルト) または 10 ビットアドレッシングモードに設定できます。

OA1 を有効にするには、I2C\_OAR1 レジスタの OA1EN ビットをセットします。

- 追加のスレーブアドレスが必要な場合は、2 番目のスレーブアドレス OA2 を設定できます。I2C\_OAR2 レジスタの OA2MSK[2:0] ビットを設定することによって、最大 7 つの OA2 LSB をマスクできます。したがって、OA2MSK が 1 から 6 まで設定された場合、OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]、または OA2[7] のみが受信アドレスと比較されます。OA2MSK が 0 に等しくなくなるとすぐに、OA2 のアドレスコンパレータは、確認応答されない I2C 予約済みアドレス (0000 XXX および 1111 XXX) を除外します。OA2MSK=7 の場合、受信されたすべてのアドレスが確認応答されます (予約済みアドレスを除く)。OA2 は常に 7 ビットアドレスです。

これらの予約済みアドレスは、特定のイネーブルビットによって有効化された場合、I2C\_OAR1 または I2C\_OAR2 レジスタが OA2MSK=0 でプログラムされた場合、確認応答できます。

OA2 を有効にするには、I2C\_OAR2 レジスタの OA2EN ビットをセットします。

- 同報アドレスは、I2C\_CR1 レジスタの GCEN ビットをセットすることで有効になります。

I2C が有効アドレスの 1 つによって選択されると、ADDR 割り込みステータスフラグがセットされ、ADDRIE ビットがセットされている場合は割り込みが生成されます。

デフォルトでは、スレーブはクロックストレッチ機能を使用し、必要ときには、ソフトウェアアクションを実行するために、SCL 信号をローレベルでストレッチすることを意味します。マスタがクロックストレッチをサポートしない場合、I2C\_CR1 レジスタの NOSTRETCH=1 で I2C を設定する必要があります。

ADDR 割り込みの受信後、いくつかのアドレスが有効な場合は、I2C\_ISR レジスタの ADDCODE[6:0] ビットを読み出して、一致するアドレスを確認する必要があります。転送方向を知るために、DIR フラグも確認する必要があります。

## スレーブクロックストレッチ (NOSTRETCH = 0)

デフォルトモードでは、I2C スレーブは次の状況で SCL クロックをストレッチします：

- ADDR フラグがセットされると：受信アドレスは有効なスレーブアドレスの 1 つと一致します。このストレッチは、ADDRCF ビットをセットすることによりソフトウェアによって ADDR フラグがクリアされたときにリリースされます。
- 送信時、前のデータ送信が完了し、新しいデータが I2C\_TXDR レジスタに書き込まれなかった場合、または ADDR フラグがクリアされたときに (TXE=1)、最初のデータバイトが書き込まれていなかった場合。このストレッチは、データが I2C\_TXDR レジスタに書き込まれたときにリリースされます。
- 受信時、I2C\_RXDR レジスタがまだ読み出されておらず、新しいデータ受信が完了したとき。このストレッチは、I2C\_RXDR が読み出されたときにリリースされます。
- スレーブバイト制御モードおよび再ロードモード (SBC=1 および RELOAD=1) で TCR = 1 のとき、すなわち、最後データバイトが転送されたとき。このストレッチは、NBYTES[7:0] フィールドにゼロ以外の値を書き込むことによって TCR がクリアされたときにリリースされます。
- SCL 立ち下がリエッジの検出後、I2C は、 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCLK}$  の間、SCL ローをストレッチします。

## クロックストレッチなしのスレーブ (NOSTRETCH = 1)

I2C\_CR1 レジスタの NOSTRETCH = 1 のとき、I2C スレーブは SCL 信号をストレッチしません。

- ADDR フラグがセットされている間、SCL クロックはストレッチされません。
- 送信時、転送に対応する最初の SCL パルスが発生する前に、I2C\_TXDR レジスタにデータが書き込まれる必要があります。そうでない場合、アンダーランが発生し、I2C\_ISR レジスタで OVR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。OVR フラグは、最初のデータ送信が開始し、STOPF ビットがまだセットされている (クリアされていない) ときにもセットされます。したがって、次の転送で送信される最初のデータを書き込んだ後、STOPF フラグをクリアすることによって、最初の送信データについても、OVR ステータスが提供できます。
- 受信時、次のデータバイトの 9 番目の SCL パルス (ACK パルス) が発生する前に、I2C\_RXDR レジスタからデータが読み出される必要があります。そうでない場合、オーバーランが発生し、I2C\_ISR レジスタの OVR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## スレーブバイト制御モード

スレーブ受信モードでバイト ACK 制御を可能にするためには、I2C\_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。これは、SMBus 標準に準拠する必要があります。

スレーブ受信モードでバイト ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD=1)。各バイトの制御を得るには、ADDR 割り込みサブルーチンで NBYTES を 0x1 に初期化し、各受信バイト後に 0x1 に再ロードする必要があります。バイトが受信されると、TCR ビットがセットされ、8 番目と 9 番目の SCL パルスの中で、SCL 信号ローをストレッチします。I2C\_RXDR レジスタからデータを読み出すことができ、その後、I2C\_CR2 レジスタの ACK ビットを設定することによって、確認応答するかどうかを決定できます。SCL ストレッチは、NBYTES をゼロ以外の値にプログラムすることによってリリースされ、確認応答または非確認応答が送信され、次のバイトを受信できます。

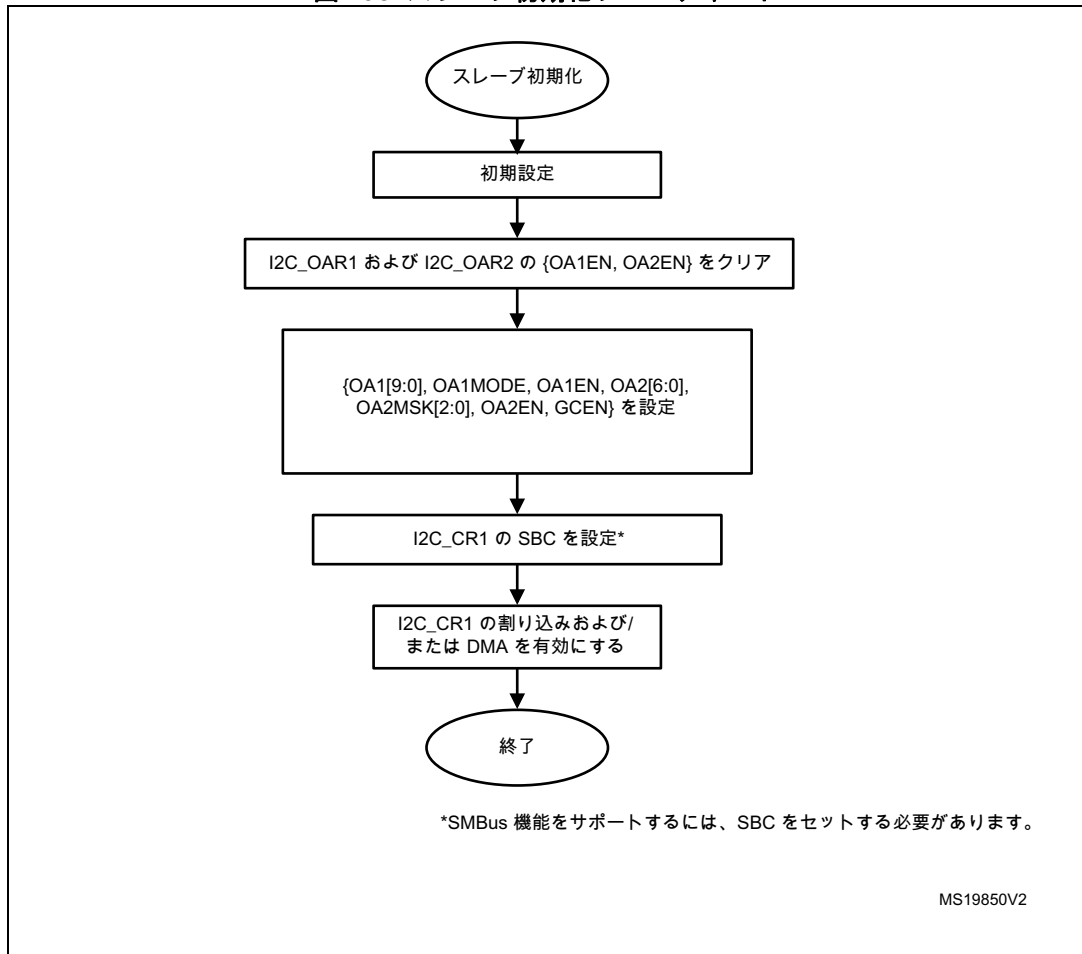
NBYTES に 0x1 より大きい値をロードでき、この場合、受信フローは NBYTES データ受信時、継続します。

**注：** SBC ビットは、I2C が無効なとき、またはスレーブがアドレス指定されていないとき、または ADDR=1 のときに設定する必要があります。

RELOAD ビットの値は、ADDR=1 のとき、または TCR=1 のときに変更できます。

**注意：** スレーブバイト制御モードは、NOSTRETCH モードと互換性がありません。NOSTRETCH=1 のときに SBC をセットすることはできません。

図 298. スレーブ初期化フローチャート



## スレーブトランスマッタ

I2C\_TXDR レジスタが空になると、送信割り込みステータス (TXIS) が生成されます。I2C\_CR1 レジスタの TXIE ビットがセットされている場合は、割り込みが生成されます。

TXIS ビットは、I2C\_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

NACK が受信されると、I2C\_ISR レジスタの NACKF ビットがセットされ、I2C\_CR1 レジスタの NACKIE ビットがセットされていた場合は割り込みが生成されます。マスタが STOP または RESTART コンディションを実行できるように、スレーブは SCL および SDA ラインを自動的にリリースします。TXIS ビットは、NACK 受信時にはセットされません。

STOP が受信され、I2C\_CR1 レジスタの STOPIE ビットがセットされると、I2C\_ISR レジスタの STOPF フラグがセットされ、割り込みが生成されます。ほとんどのアプリケーションでは、SBC は通常、0 にプログラムされます。この場合、スレーブアドレスが受信されたときに (ADDR=1)、TXE = 0 であった場合、I2C\_TXDR レジスタの内容を最初のデータバイトとして送信するか、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C\_TXDR レジスタをフラッシュするかを選択できます。

スレーブバイト制御モード (SBC=1) では、送信バイト数をアドレス一致割り込みサブルーチンの NBYTES でプログラムする必要があります (ADDR=1)。この場合、転送中の TXIS イベントの数は、NBYTES でプログラムされた値に対応します。



**注意：** NOSTRETCH=1 のとき、SCL クロックは ADDR フラグがセットされている間はストレッチされない  
ので、最初のデータバイトをプログラムするために ADDR サブルーチンで I2C\_TXDR レジスタの  
内容をフラッシュすることはできません。最初に送信されるデータバイトは、I2C\_TXDR レジスタで前  
もってプログラムされている必要があります。

- このデータは、前の送信メッセージの最後の TXIS イベントで書き込まれたデータでもかまいません。
- このデータバイトが送信データバイトでない場合、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C\_TXDR レジスタをフラッシュできます。アドレスの確認応答に続いて、最初のデータ送信が開始する前にこれらが実行されることを保証するためには、STOPF ビットのクリアは、これらのアクションの後でのみ行う必要があります。

最初のデータ送信が開始したときに STOPF がまだセットされていた場合、アンダーランエラーが生成されます (OVR フラグがセットされます)。

TXIS イベントが必要な場合 (送信割り込みまたは送信 DMA リクエスト)、TXIS イベントを生成するためには、TXE ビットに加えて TXIS ビットもセットする必要があります。

図 299. I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0)

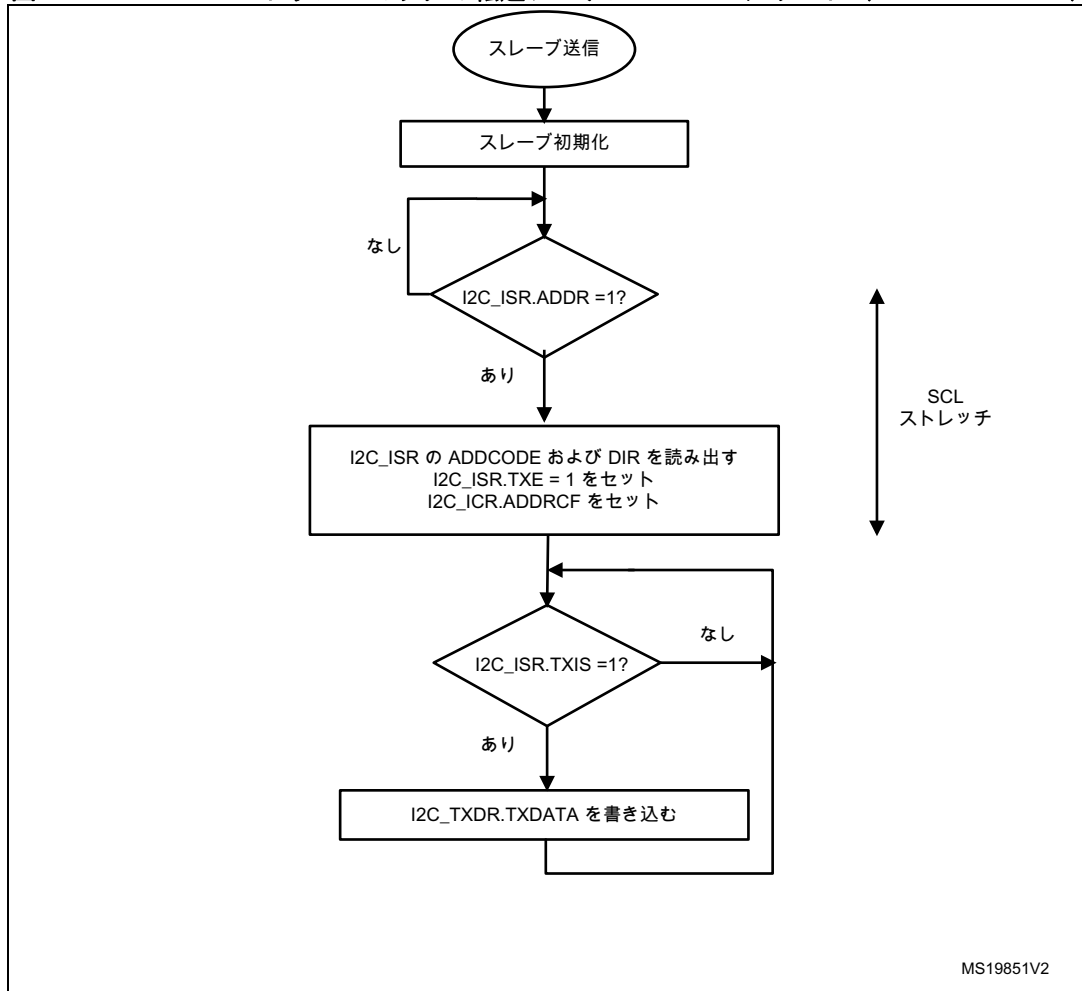


図 300. I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1)

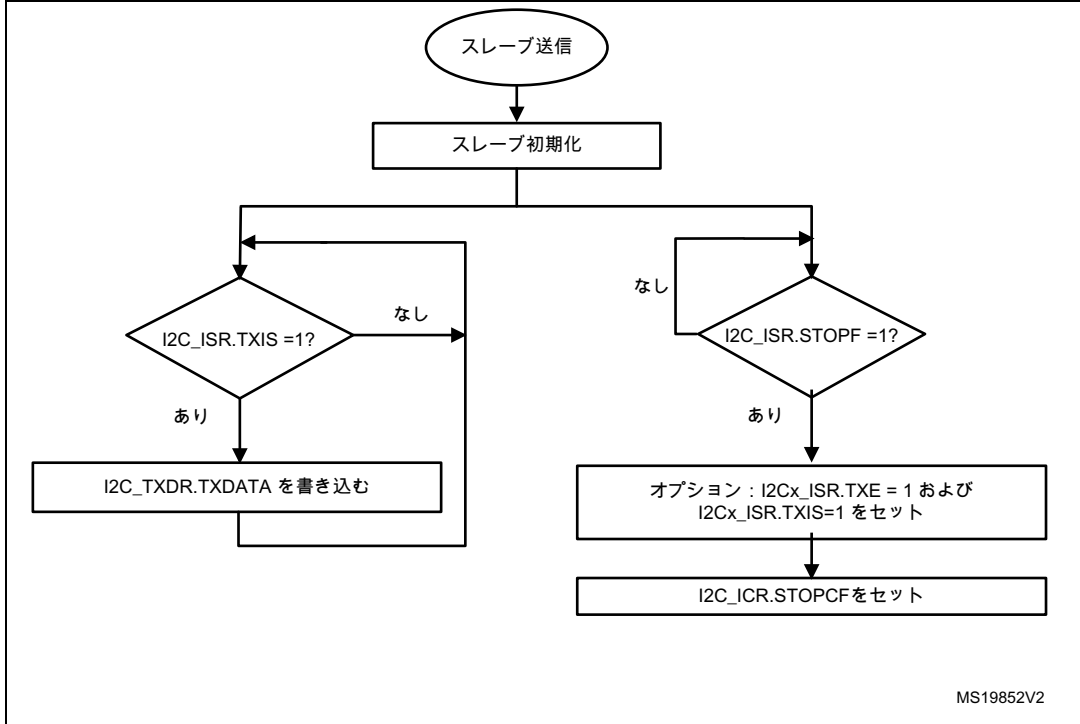
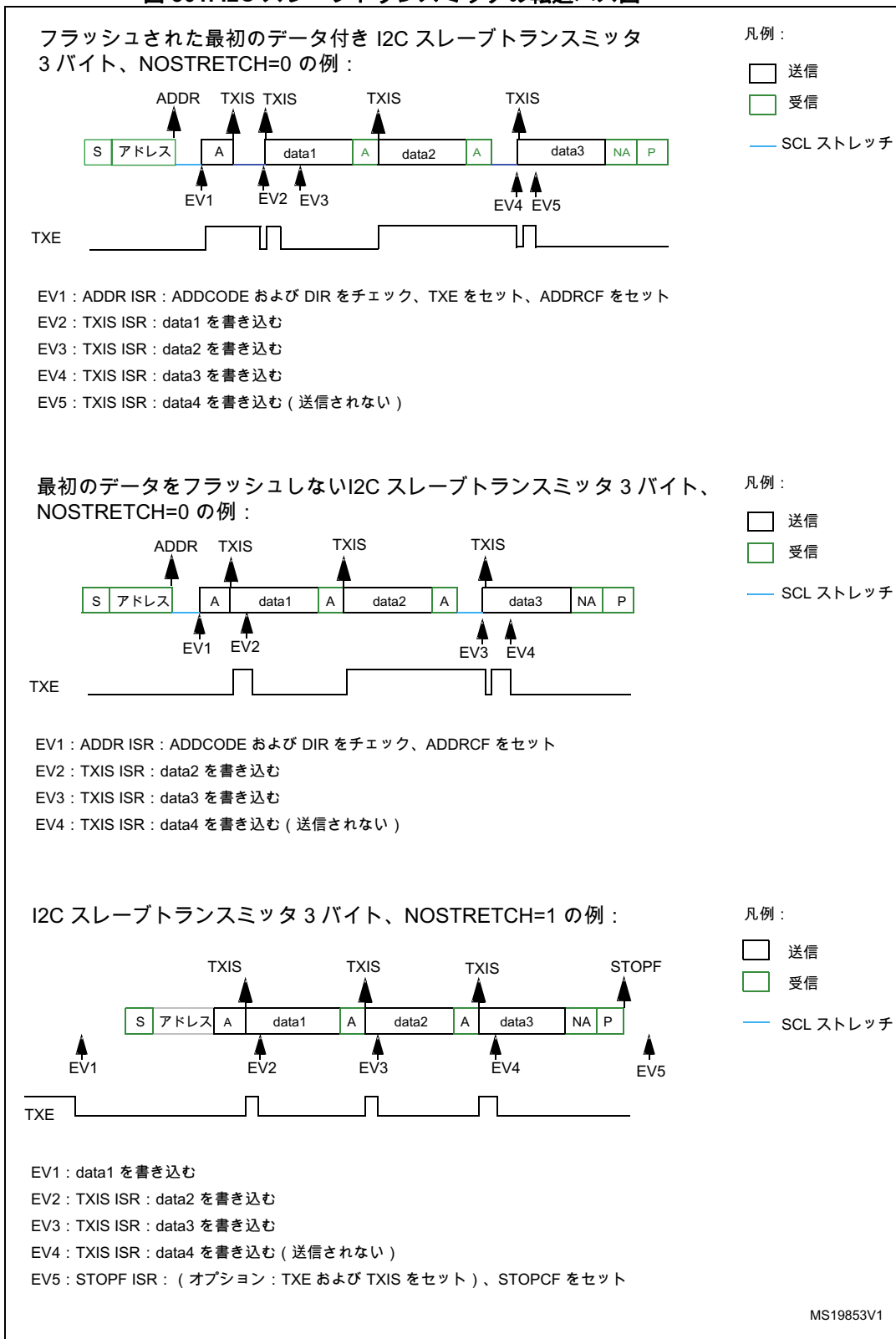


図 301. I2C スレーブトランスミッタの転送バス図



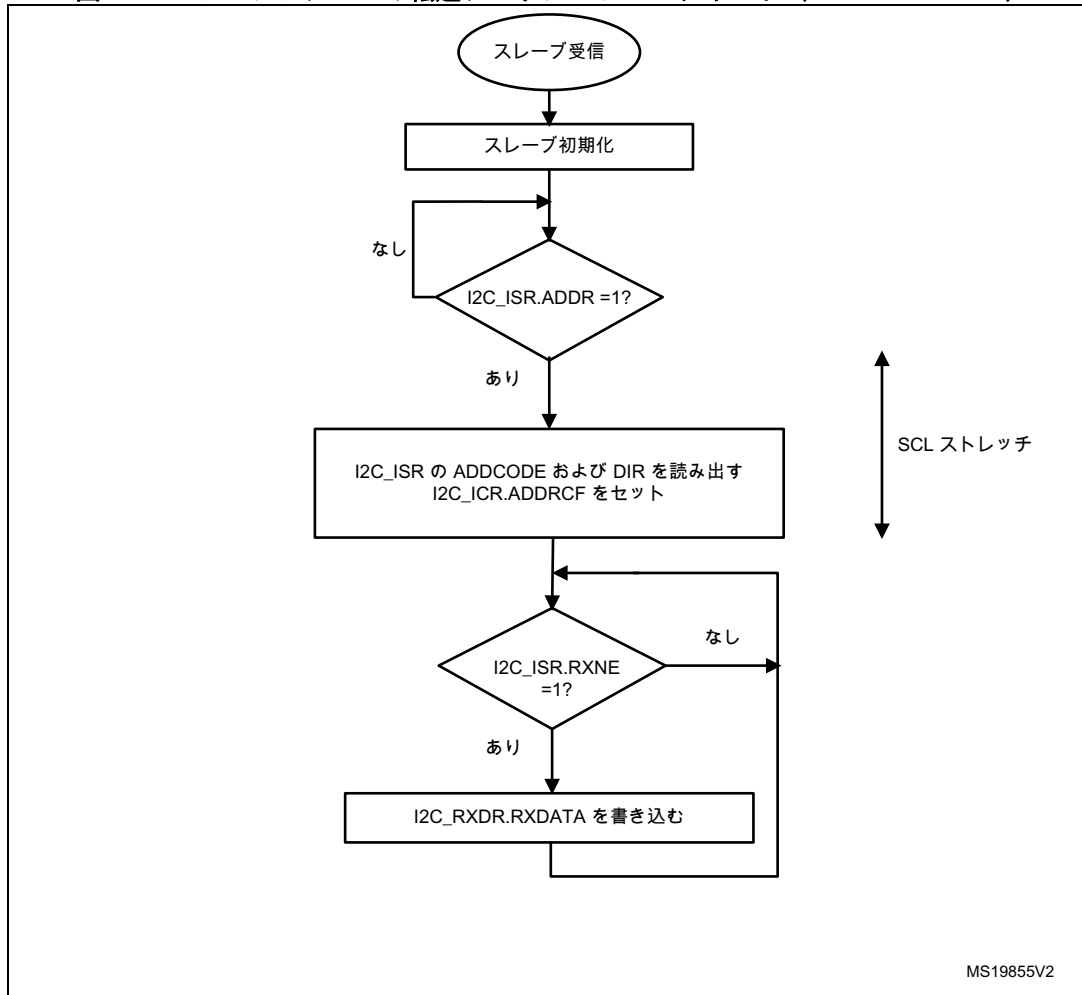


## スレープレシーバ

I2C\_RXDR がフルのときには、I2C\_ISR の RXNE がセットされ、I2C\_CR1 の RXIE がセットされている場合は割り込みが生成されます。RXNE は、I2C\_RXDR が読み出されたときにクリアされます。

STOP が受信され、I2C\_CR1 レジスタの STOPIE ビットがセットされると、I2C\_ISR の STOPF がセットされ、割り込みが生成されます。

図 302. スレープレシーバの転送シーケンスフローチャート (NOSTRETCH=0)



MS19855V2

図 303. スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1)

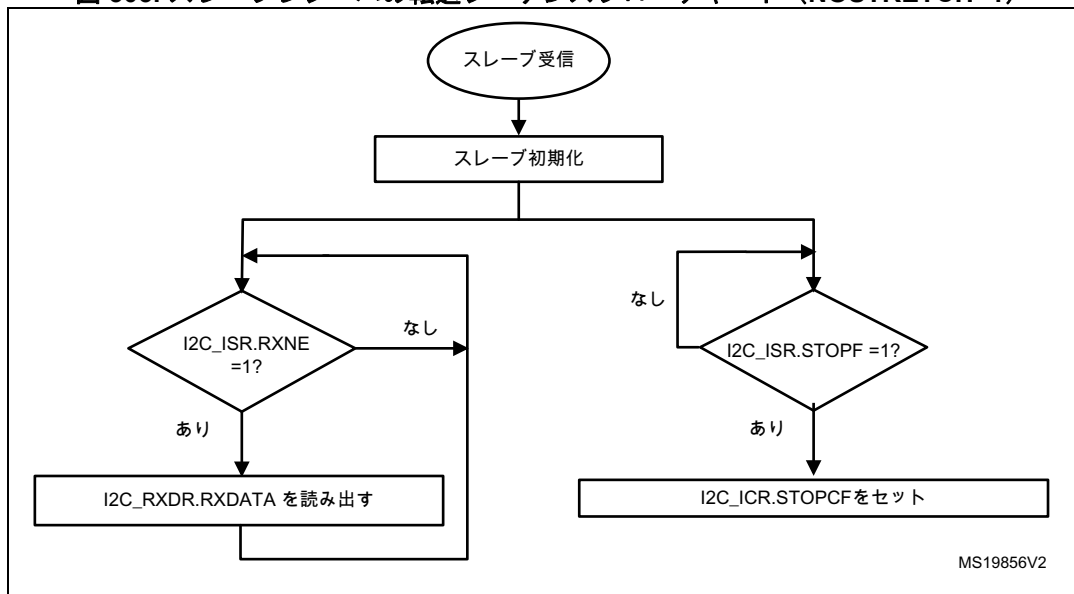
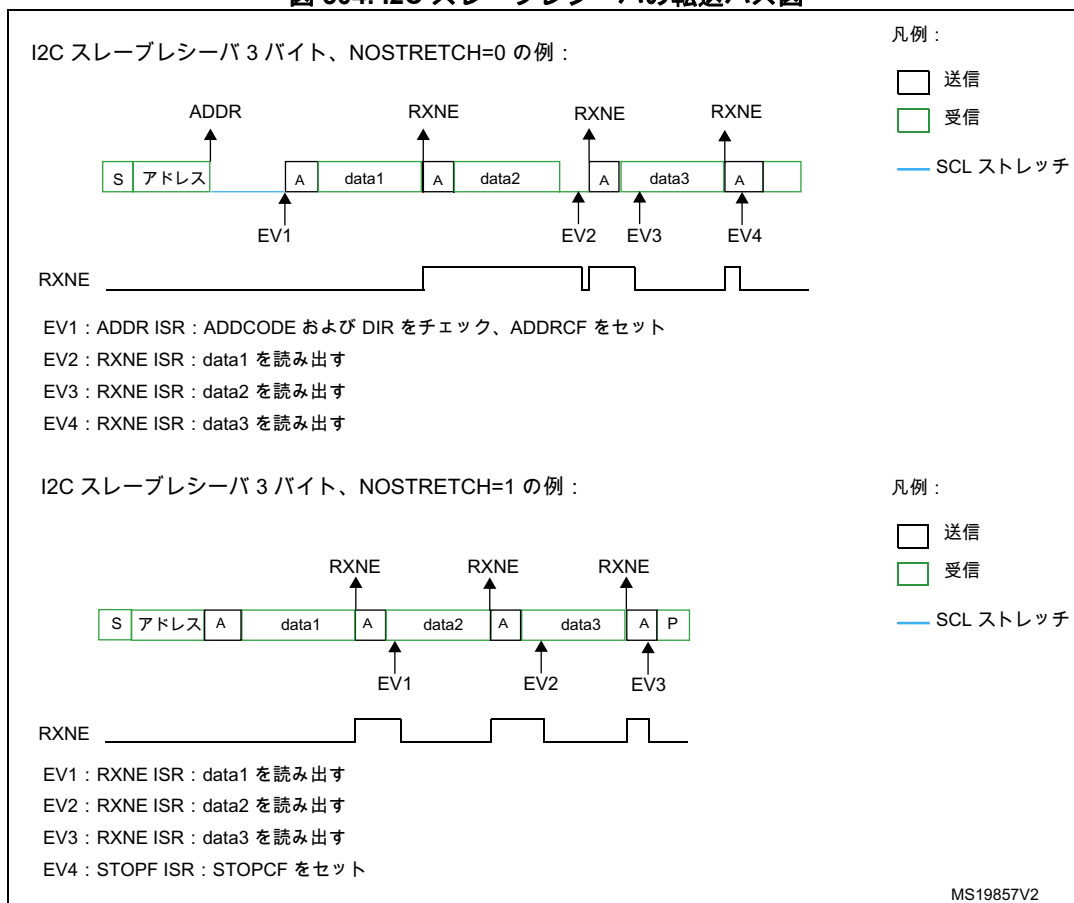


図 304. I2C スレーブレシーバの転送バス図



## 28.4.8 I2C マスタモード

### I2C マスタ初期化

ペリフェラルを有効にする前に、I2C\_TIMINGR レジスタの SCLH および SCLL ビットをセットすることによって、I2C マスタクロックを設定する必要があります。

STM32CubeMX ツールは、I2C 設定ウィンドウの I2C\_TIMINGR コンテンツを計算し、提供します。

マルチマスタ環境とスレーブクロックストレッチをサポートするために、クロック同期メカニズムが実装されています。

クロック同期を可能にするために：

- クロックのローレベルは SCLL カウンタを使用してカウントされ、SCL ローレベル内部検出から開始されます。
- クロックのハイレベルは SCLH カウンタを使用してカウントされ、SCL ハイレベル内部検出から開始されます。

I2C は、SCL 立ち下がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延  $t_{\text{SYNC1}}$  の後に SCL ローレベルを検出します。SCLL カウンタが I2C\_TIMINGR レジスタの SCLL[7:0] ビットでプログラムされた値に達すると、I2C は SCL をハイレベルにリリースします。

I2C は、SCL 立ち上がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延  $t_{\text{SYNC2}}$  の後に SCL ハイレベルを検出します。SCLH カウンタが I2C\_TIMINGR レジスタの SCLH[7:0] ビットでプログラムされた値に達すると、I2C は SCL をローレベルにします。

結果として、マスタクロック周期は次のとおりです：

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}+1) + (\text{SCLL}+1)] \times (\text{PRESC}+1) \times t_{\text{I2CCLK}}\}$$

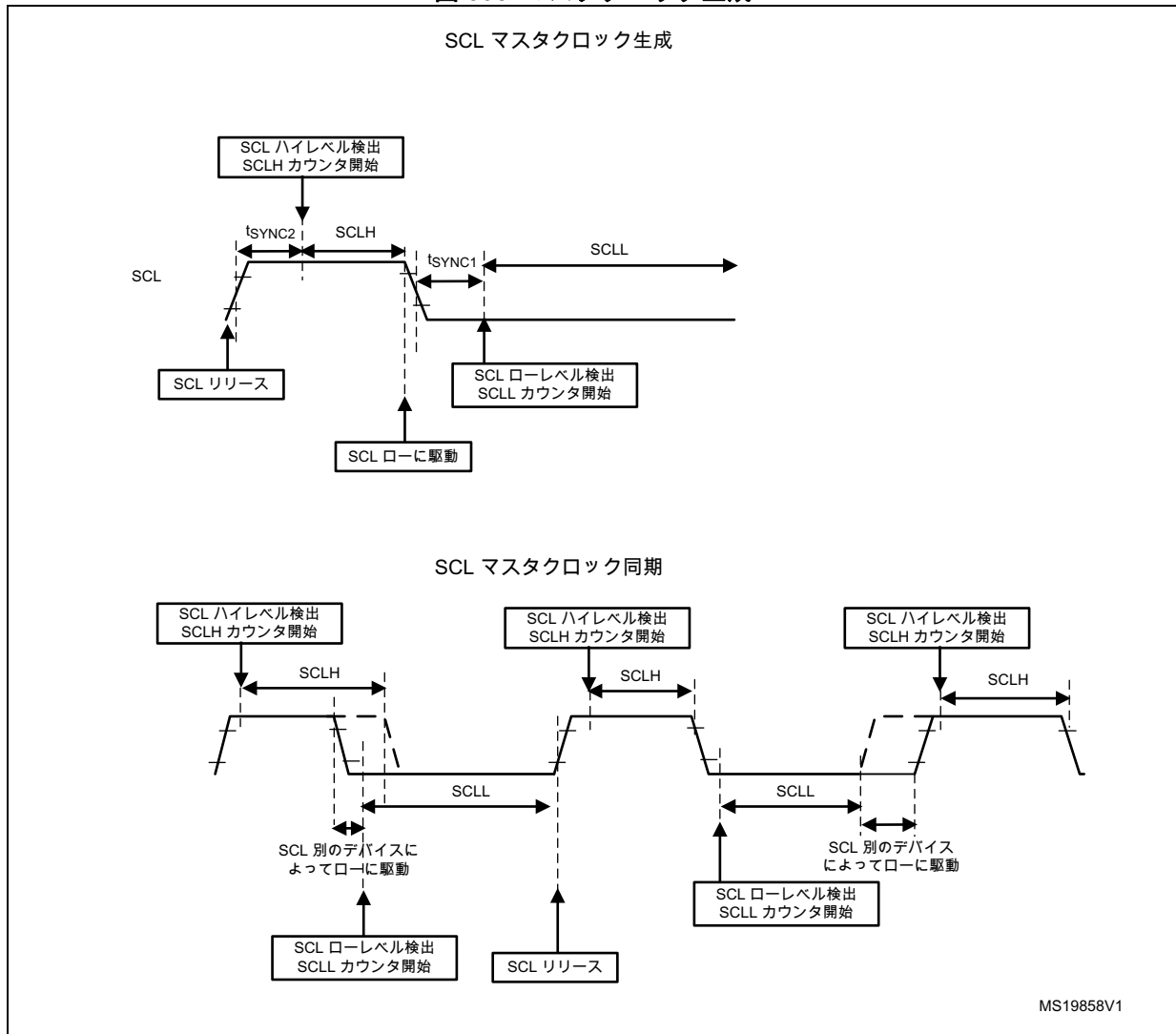
$t_{\text{SYNC1}}$  の長さは、次のパラメータに依存します：

- SCL 立ち下がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延：DNF  $\times t_{\text{I2CCLK}}$
- I2CCLK クロックとの SCL 同期による遅延（2 ~ 3 I2CCLK 周期）

$t_{\text{SYNC2}}$  の長さは、次のパラメータに依存します：

- SCL 立ち上がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延：DNF  $\times t_{\text{I2CCLK}}$
- I2CCLK クロックとの SCL 同期による遅延（2 ~ 3 I2CCLK 周期）

図 305. マスタクロック生成



**注意 :** I<sup>2</sup>C または SMBus 準拠のためには、マスタクロックは次のタイミングを満たす必要があります :

表 145. I<sup>2</sup>C-SMBUS 仕様のクロックタイミング

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
$f_{SCL}$	SCL クロック周波数		100		400		1000		100	kHz
$t_{HD:STA}$	(反復) START コンディションのホールド時間	4.0	-	0.6		0.26	-	4.0	-	μs
$t_{SU:STA}$	反復 START コンディションのセットアップ時間	4.7	-	0.6		0.26	-	4.7	-	μs
$t_{SU:STO}$	STOP コンディションのセットアップ時間	4.0	-	0.6		0.26	-	4.0	-	μs
$t_{BUF}$	STOP コンディションと START コンディションの間のバスフリー時間	4.7	-	1.3		0.5	-	4.7	-	μs
$t_{LOW}$	SCL クロックのロー周期	4.7	-	1.3		0.5	-	4.7	-	μs
$t_{HIGH}$	SCL クロックの周期	4.0	-	0.6		0.26	-	4.0	50	μs
$t_r$	SDA および SCL 信号の立ち上がり時間	-	1000	-	300		120	-	1000	ns
$t_f$	SDA および SCL 信号の立ち下がり時間	-	300	-	300		120	-	300	ns

注: **SCLL** は、 $t_{BUF}$  および  $t_{SU:STA}$  タイミングの生成にも使用されます。

**SCLH** は、 $t_{HD:STA}$  および  $t_{SU:STO}$  タイミングの生成にも使用されます。

I2C\_TIMINGR 設定と I2CCLK 周波数の例については、[セクション 28.4.9: I2C\\_TIMINGR レジスタの設定例](#) を参照してください。

### マスタ通信の初期化 (アドレスフェーズ)

通信を初期化するためには、I2C\_CR2 レジスタでアドレス指定されたスレーブについて次のパラメータをプログラムする必要があります。

- アドレッシングモード (7 ビットまたは 10 ビット) : ADD10
- 送信されるスレーブアドレス : SADD[9:0]
- 転送方向 : RD\_WRN
- 10 ビットアドレスが読み出される場合 : HEAD10R ビット。HEAD10R を設定して、完全なアドレスシーケンスが送信されなければならないか、ヘッダのみ (方向の変更の場合) かを示す必要があります。
- 転送されるバイト数 : NBYTES[7:0]。バイト数が 255 バイト以上の場合、NBYTES[7:0] に 0xFF を書き込む必要があります。

次に、I2C\_CR2 レジスタの START ビットをセットする必要があります。START ビットがセットされているとき、上記のすべてのビットを変更することはできません。

その場合、マスタは、バスがフリーである (BUSY = 0) ことを検出すると、 $t_{BUF}$  の遅延後に、自動的に START コンディションとスレーブアドレスを送信します。

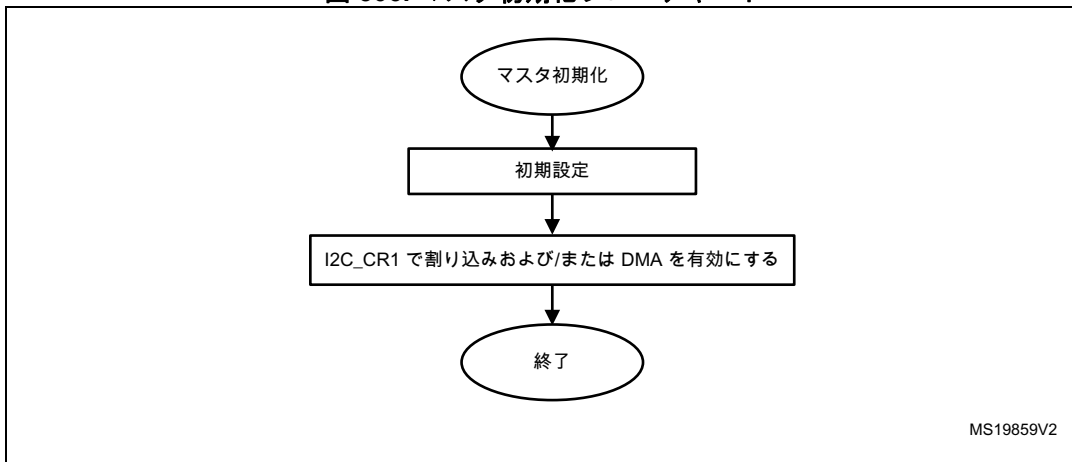


アービトレーション喪失の場合、マスタはスレーブモードに自動的に切り替えて、スレーブとしてアドレス指定された場合は専用アドレスを確認応答できます。

**注：** START ビットは、スレーブアドレスがバスに送信されたとき、受信した確認応答値にかかわらず、ハードウェアによってリセットされます。START ビットは、アービトレーション喪失が発生した場合にも、ハードウェアによってリセットされます。START ビットがセットされているときに、I2C がスレーブとしてアドレス指定された場合 (ADDR=1)、I2C はスレーブモードに切り替わり、ADDRCF ビットがセットされたときに START ビットがクリアされます。

**注：** 反復スタートコンディションにも同じ手順が適用されます。この場合、BUSY=1 です。

図 306. マスタ初期化フローチャート



## 10 ビットアドレススレーブをアドレス指定するマスタレシーバの初期化

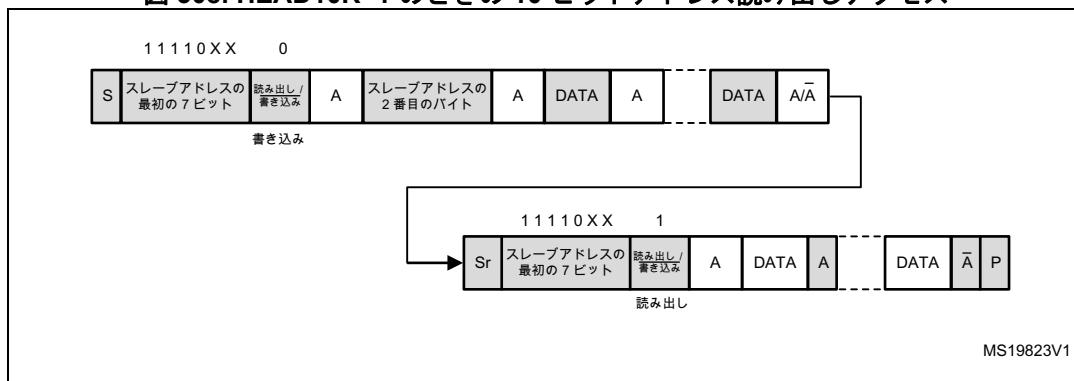
- スレーブアドレスが 10 ビット形式の場合、I2C\_CR2 レジスタの HEAD10R ビットをクリアすることによって、完全な読み出しシーケンスを送信することができます。この場合、マスタは、START ビットがセットされた後、次のような完全なシーケンスを自動的に送信します：(Re) START + スレーブアドレス 10 ビットヘッダ書き込み + スレーブアドレスの 2 番目のバイト + RStart + スレーブアドレス 10 ビットヘッダ読み出し。

図 307. HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス



- マスタが 10 ビットアドレススレーブをアドレス指定して、このスレーブアドレスにデータを送信した後、同じスレーブからデータを読み出す場合には、まず、マスタ送信フローが行われる必要があります。その場合、反復開始が、HEAD10R=1 で設定された 10 ビットスレーブアドレスでセットされます。この場合、マスタは次のシーケンスを送信します：ReStart + スレーブアドレス 10 ビットヘッダ読み出し。

図 308. HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス



## マスタトランスマッタ

書き込み転送の場合、ACK が受信されたときの9 番目の SCL パルス後、各バイトの送信後にTXIS フラグがセットされます。

I2C\_CR1 レジスタの TXIE ビットがセットされている場合、TXIS イベント時に割り込みが生成されます。このフラグは、I2C\_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

転送中の TXIS イベントの数は、NBYTES[7:0] でプログラムされた値に対応します。送信されるデータバイト数の合計が 255 より大きい場合、I2C\_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

TXIS フラグは、NACK 受信時にはセットされません。

- RELOAD=0 でNBYTES データが転送されたとき：
  - 自動終了モード (AUTOEND=1) では、STOP が自動的に送信されます。
  - ソフトウェア終了モード (AUTOEND=0) では、TC フラグがセットされ、ソフトウェアアクションを実行するために SCL ラインがローでストレッチされます：
 

正しいスレーブアドレス設定と転送バイト数で I2C\_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションがバスに送信されます。

I2C\_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。
- NACK が受信された場合：TXIS フラグはセットされず、NACK 受信後、自動的に STOP コンディションが送信されます。I2C\_ISR レジスタの NACKF フラグがセットされ、NACKIE ビットがセットされていた場合は割り込みが生成されます。

図 309. N≤255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート

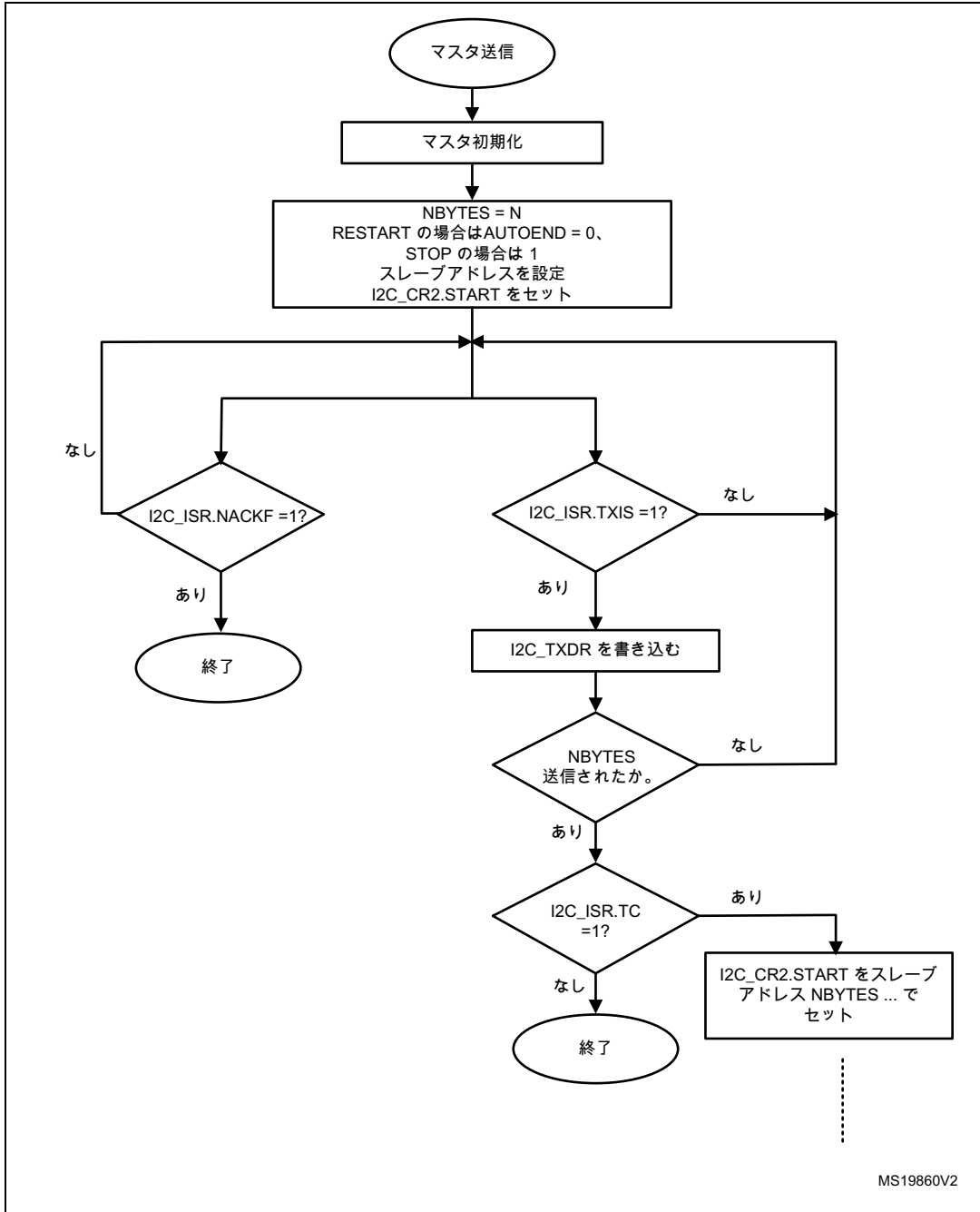
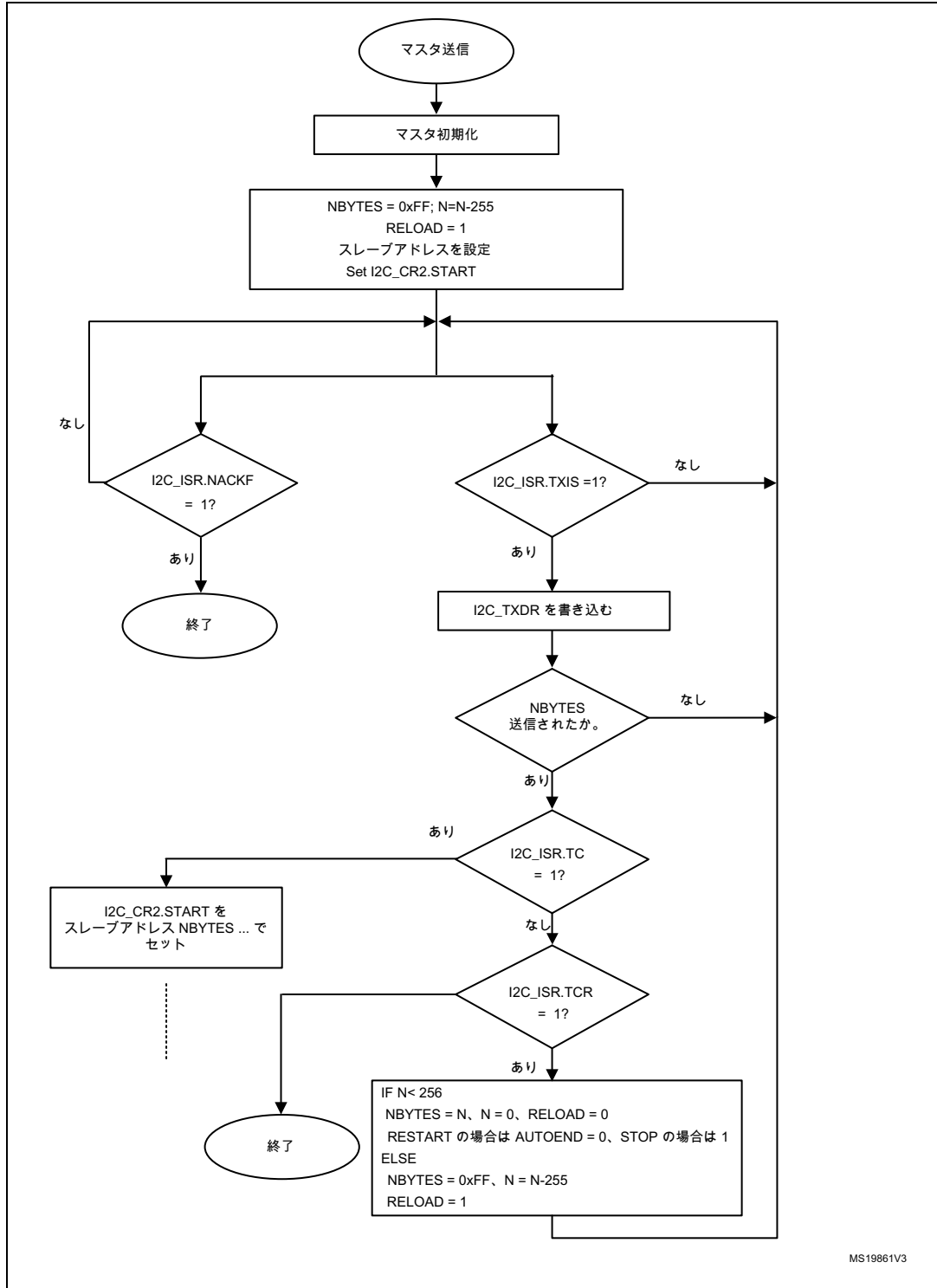


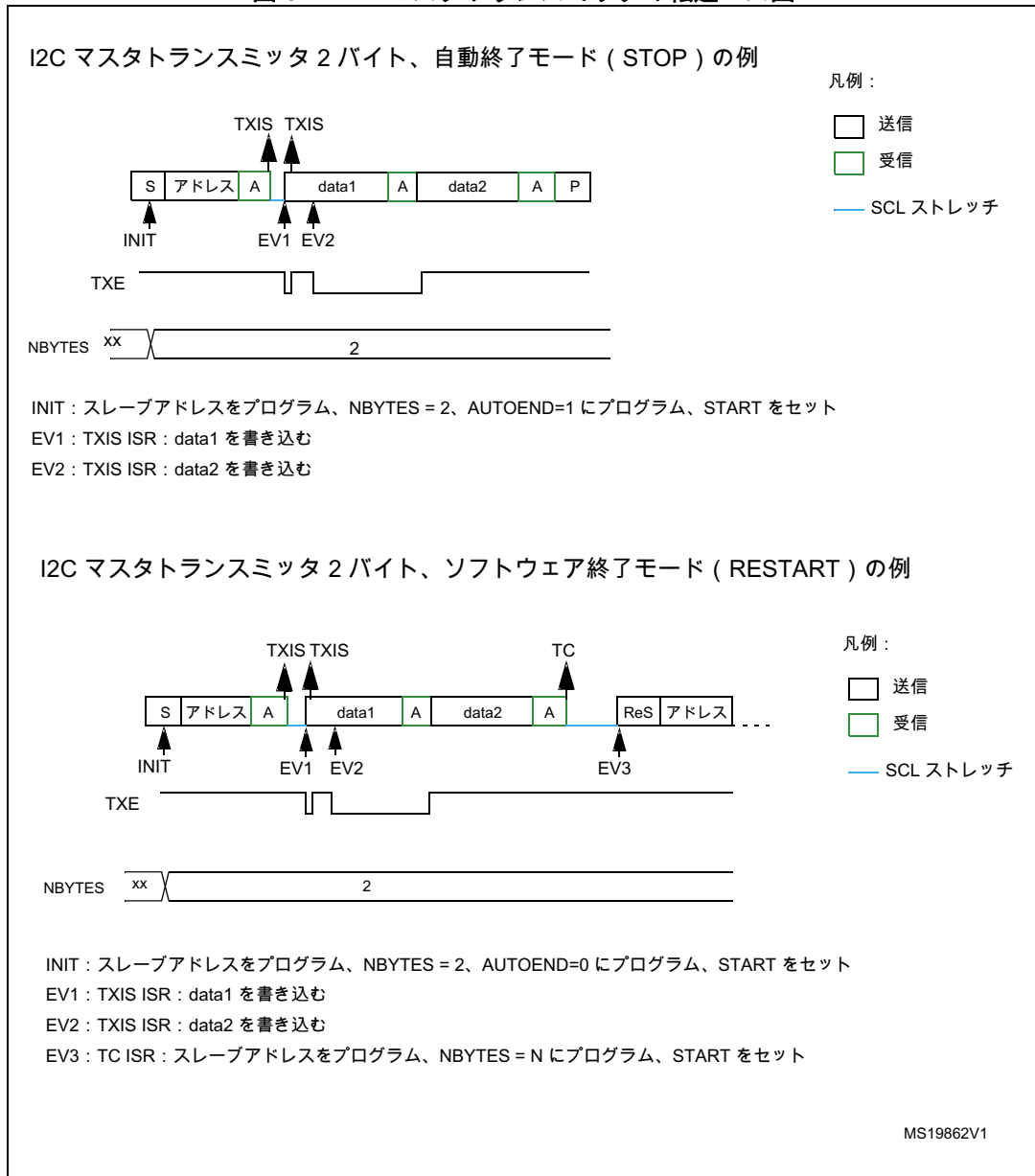


図 310. N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート



MS19861V3

図 311. I2C マスタトランスミッタの転送バス図



## マスタレシーバ

読み出し転送の場合、各バイトの受信後や8番目の SCL パルス後に RXNE フラグがセットされます。I2C\_CR1 レジスタの RXIE ビットがセットされている場合、RXNE イベント時に割り込みが生成されます。このフラグは、I2C\_RXDR が読み出されたときにクリアされます。

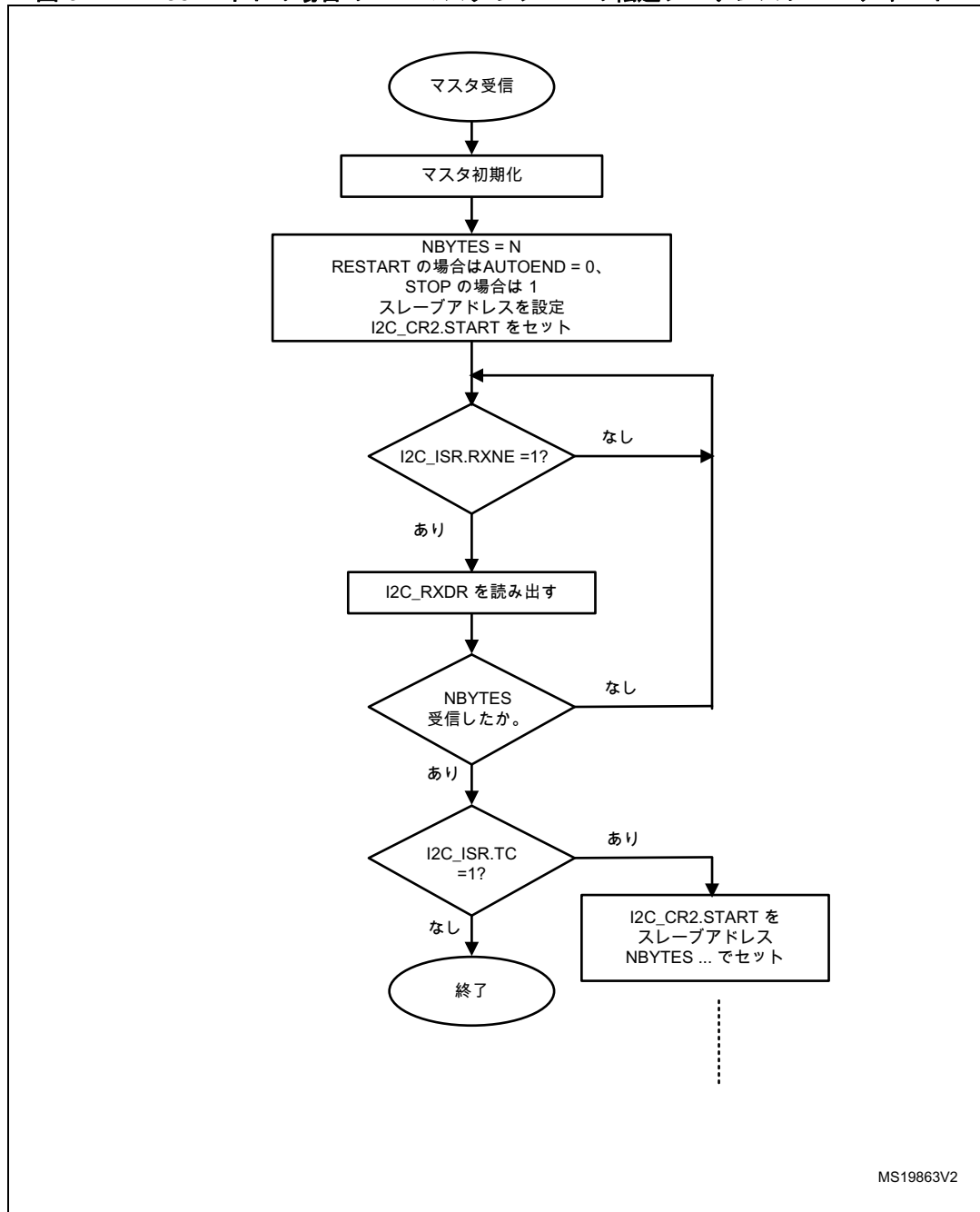
受信されるデータバイト数の合計が 255 より大きい場合、I2C\_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES[7:0] データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

- RELOAD=0 で NBYTES[7:0] データが転送されたとき：
  - 自動終了モード (AUTOEND=1) では、最後の受信バイト後に NACK および STOP が自動的に送信されます。
  - ソフトウェア終了モード (AUTOEND=0) では、最後の受信バイト後に NACK が自動的に送信され、TC フラグがセットされ、ソフトウェアアクションを実行できるように、SCL ラインがローでストレッチされます。

正しいスレーブアドレス設定と転送バイト数で I2C\_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションとスレーブアドレスがバスに送信されます。

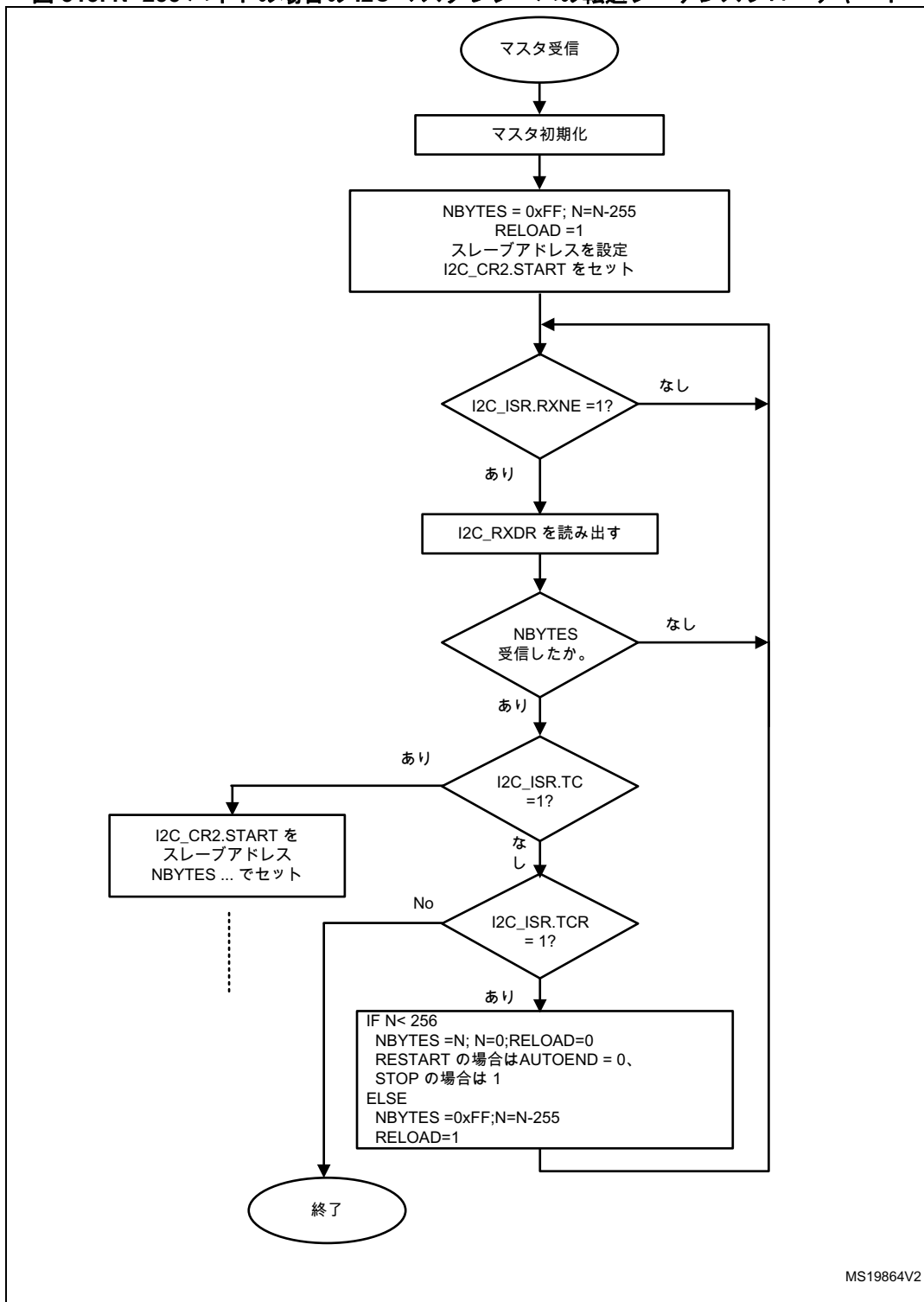
I2C\_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。

図 312. N ≤ 255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート



MS19863V2

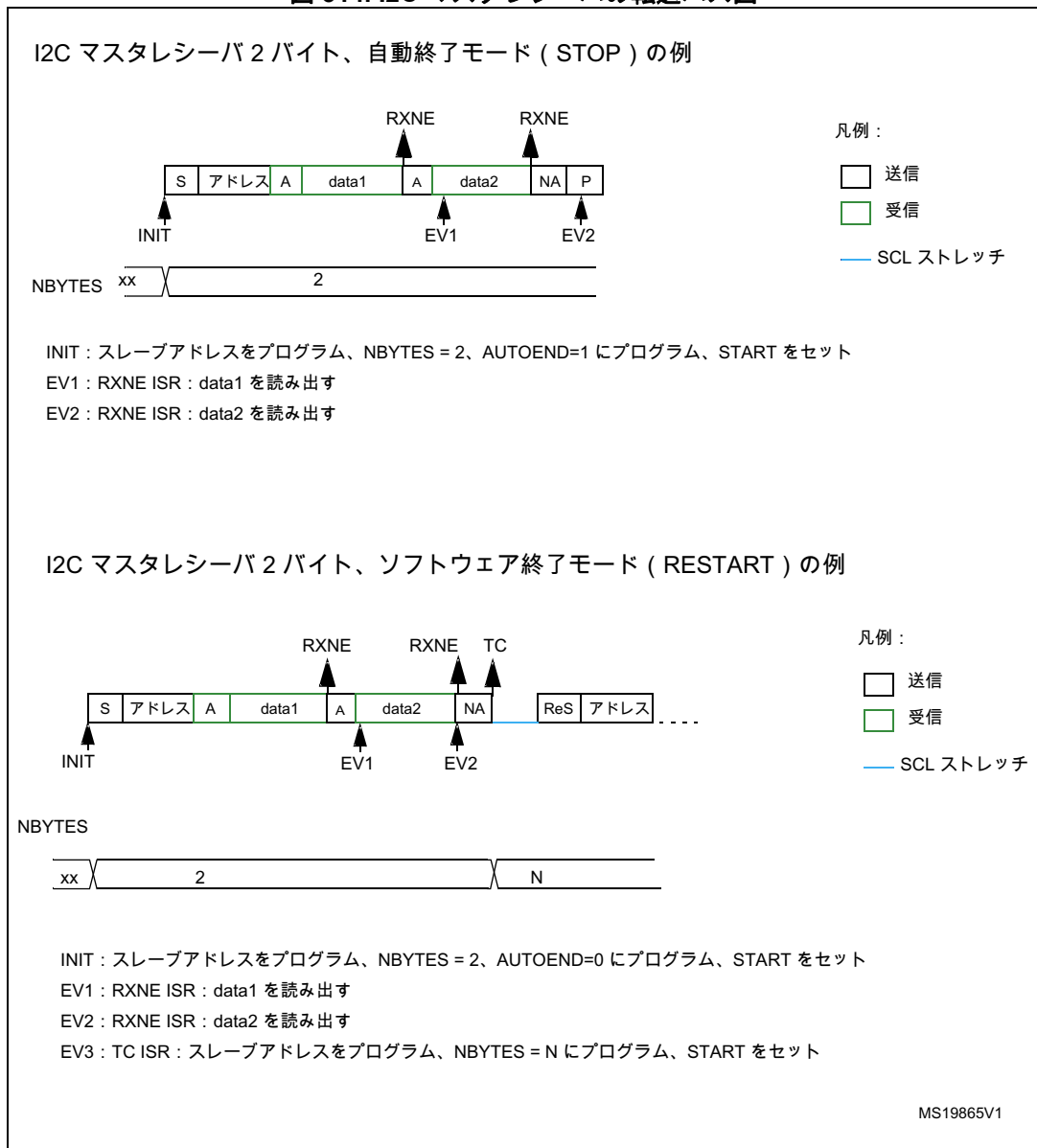
図 313. N>255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート



MS19864V2



図 314. I2C マスタレシーバの転送バス図



## 28.4.9 I2C\_TIMINGR レジスタの設定例

下の表に、I<sup>2</sup>C 仕様準拠したタイミングを得るための I2C\_TIMINGR をプログラムする方法の例を示します。より正確な設定値を得るには、アプリケーションノート I<sup>2</sup>C タイミング設定ツール (AN4235) および該当するソフトウェア STSW-STM32126 を参照してください。

表 146. f<sub>I2CCLK</sub> = 8 MHz でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t <sub>SCLL</sub>	200x250 ns = 50 μs	20x250 ns = 5.0 μs	10x125 ns = 1250 ns	7x125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t <sub>SCLH</sub>	196x250 ns = 49 μs	16x250 ns = 4.0 μs	4x125ns = 500ns	4x125 ns = 500 ns
t <sub>SCL</sub> <sup>(1)</sup>	~100 μs <sup>(2)</sup>	~10 μs <sup>(2)</sup>	~2500 ns <sup>(3)</sup>	~2000 ns <sup>(4)</sup>
SDADEL	0x2	0x2	0x1	0x0
t <sub>SDADEL</sub>	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t <sub>SCLDEL</sub>	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

1. SCL 周期 t<sub>SCL</sub> は、SCL 内部検出遅延のため、t<sub>SCLL</sub> + t<sub>SCLH</sub> より大きくなります。t<sub>SCL</sub> として示されている値は例にすぎません。
2. t<sub>SYNC1</sub> + t<sub>SYNC2</sub> 最小値は、4 × t<sub>I2CCLK</sub> = 500 ns です。例：t<sub>SYNC1</sub> + t<sub>SYNC2</sub> = 1000 ns
3. t<sub>SYNC1</sub> + t<sub>SYNC2</sub> 最小値は、4 × t<sub>I2CCLK</sub> = 500 ns です。例：t<sub>SYNC1</sub> + t<sub>SYNC2</sub> = 750 ns
4. t<sub>SYNC1</sub> + t<sub>SYNC2</sub> 最小値は、4 × t<sub>I2CCLK</sub> = 500 ns です。例：t<sub>SYNC1</sub> + t<sub>SYNC2</sub> = 655 ns

表 147. f<sub>I2CCLK</sub> = 48 MHz でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
t <sub>SCLL</sub>	200 x 250 ns = 50 μs	20 x 250 ns = 5.0 μs	10 x 125 ns = 1250 ns	4 x 125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
t <sub>SCLH</sub>	196 x 250 ns = 49 μs	16 x 250 ns = 4.0 μs	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns
t <sub>SCL</sub> <sup>(1)</sup>	~100 μs <sup>(2)</sup>	~10 μs <sup>(2)</sup>	~2500 ns <sup>(3)</sup>	~875 ns <sup>(4)</sup>
SDADEL	0x2	0x2	0x3	0x0
t <sub>SDADEL</sub>	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	3 x 125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t <sub>SCLDEL</sub>	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. SCL 周期 t<sub>SCL</sub> は、SCL 内部検出遅延のため、t<sub>SCLL</sub> + t<sub>SCLH</sub> より大きくなります。t<sub>SCL</sub> として示されている値は例にすぎません。
2. t<sub>SYNC1</sub> + t<sub>SYNC2</sub> 最小値は、4 × t<sub>I2CCLK</sub> = 83.3 ns です。例：t<sub>SYNC1</sub> + t<sub>SYNC2</sub> = 1000 ns
3. t<sub>SYNC1</sub> + t<sub>SYNC2</sub> 最小値は、4 × t<sub>I2CCLK</sub> = 83.3 ns です。例：t<sub>SYNC1</sub> + t<sub>SYNC2</sub> = 750 ns



4.  $t_{\text{SYNC1}} + t_{\text{SYNC2}}$  最小値は、 $4 \times t_{\text{I2CCLK}} = 83.3 \text{ ns}$  です。例:  $t_{\text{SYNC1}} + t_{\text{SYNC2}} = 250 \text{ ns}$

### 28.4.10 SMBus 固有の機能

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 28.3 : I2C の実装](#) を参照してください。

#### 概要

システム管理バス (SMBus) は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I<sup>2</sup>C の動作原理に基づきます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。

このペリフェラルは、SMBUS 仕様 rev 2.0 (<http://smbus.org>) と互換性があります。

システム管理バス仕様では、3 種類のデバイスを規定しています。

- スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。
- マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。
- ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ/スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

このペリフェラルは、マスタまたはスレーブデバイスとして、また、ホストとしても設定できます。

#### SMBUS は、I<sup>2</sup>C 仕様 rev 2.1 に基づきます。

#### バスプロトコル

特定のデバイスについて、11 の可能なコマンドプロトコルがあります。デバイスは、11 のプロトコルの一部または全部を使用して通信できます。プロトコルは、Quick Command、Send Byte、Receive Byte、Write Byte、Write Word、Read Byte、Read Word、Process Call、Block Read、Block Write、および Block Write-Block Read Process Call です。これらのプロトコルは、ユーザのソフトウェアによって実装してください。

これらのプロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

#### アドレス解決プロトコル (ARP)

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス割り当てを目的とする各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子 (UDID) を実装する必要があります。128 ビットの番号がソフトウェアによって実装されます。

このペリフェラルは、アドレス解決プロトコル (ARP) をサポートします。SMBus デバイスのデフォルトアドレス (0b11001100 001) は、I2C\_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。ARP コマンドは、ユーザのソフトウェアによって実装してください。

ARP サポートのために、スレーブモードでアービトレーションも行われます。

SMBus アドレス解決プロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。



## 受信コマンドおよびデータ確認応答制御

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C\_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、[スレーブバイト制御モード \(828 ページ\)](#) を参照してください。

## Host Notify プロトコル

このペリフェラルは、I2C\_CR1 レジスタの SMBHEN ビットをセットすることによって、Host Notify (ホスト通知) プロトコルをサポートします。この場合、ホストは SMBus ホストアドレス (0b0001 000) を確認応答します。

このプロトコルが使用されると、デバイスはマスタとして動作し、ホストはスレーブとして動作します。

## SMBus アラート

SMBus ALERT オプション信号がサポートされます。スレーブ専用デバイスは、通信したいホストの SMBALERT# ピンを通じてホストに信号を送信します。ホストは、割り込みを処理し、アラート応答アドレス (0b0001 100) を通じて全 SMBALERT# デバイスに同時にアクセスします。SMBALERT# をローに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。

スレーブデバイスとして設定されたとき (SMBHEN=0)、I2C\_CR1 レジスタの ALERTEN ビットをセットすることによって、SMBA ピンはローに引き下げられます。同時に、アラート応答アドレスが有効になります。

ホストとして設定されたとき (SMBHEN=1)、SMBA ピンで立ち下がりエッジが検出され、ALERTEN=1 のとき、I2C\_ISR レジスタの ALERT フラグがセットされます。I2C\_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。ALERTEN=0 のときには、外部 SMBA ピンがローの場合でも、ALERT ラインはハイとみなされます。

SMBus ALERT ピンが不要な場合には、ALERTEN=0 の場合、SMBA ピンを標準 GPIO として使用できません。

## パケットエラーチェック

信頼性と通信の堅牢性を向上させるために、SMBus 仕様にパケットエラーチェックメカニズムが導入されました。パケットエラーチェックは、各メッセージ転送の終わりにパケットエラーコード (PEC) を付加することによって実装されます。PEC は、すべてのメッセージバイト (アドレスと読み出し/書き込みビットを含む) に対して  $C(x) = x_8 + x^2 + x + 1$  CRC-8 多項式を使用して計算されます。

ペリフェラルはハードウェア PEC 計算機が組み込まれ、受信バイトがハードウェアによって計算された PEC に一致しないときには自動的に非確認応答を送信できます。

## タイムアウト

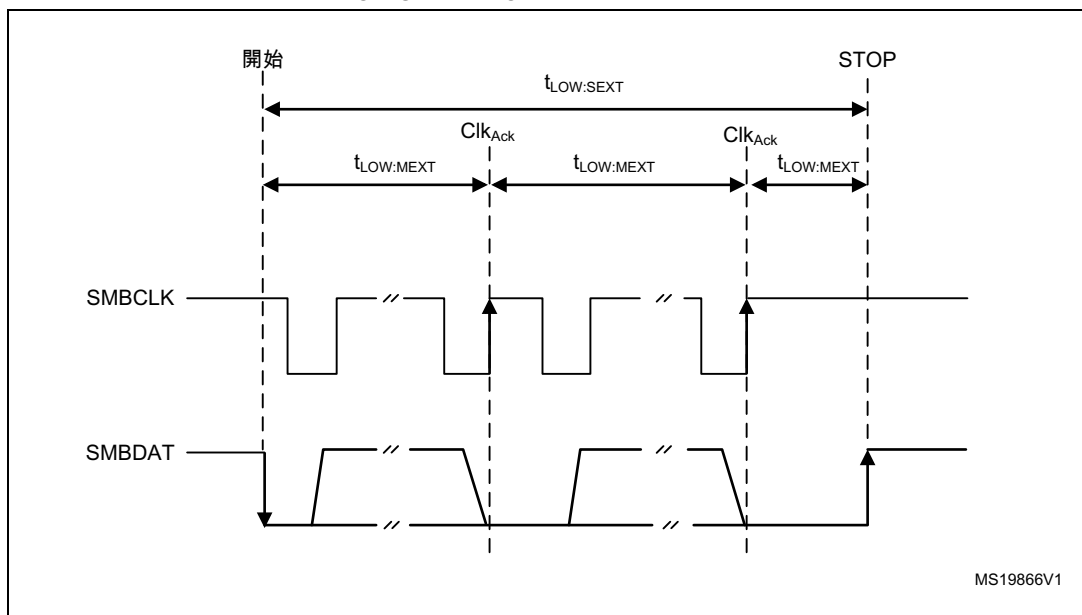
このペリフェラルは、SMBus 仕様バージョン 2.0 で定義された 3 つのタイムアウトに準拠するために、ハードウェアタイマが組み込まれています。

表 148. SMBus タイムアウト仕様

記号	パラメータ	リミット		単位
		最小値	最大値	
$t_{\text{TIMEOUT}}$	クロックロータイムアウト検出	25	35	ms
$t_{\text{LOW:SEXT}}^{(1)}$	累積クロックロー延長時間 (スレーブデバイス)	-	25	ms
$t_{\text{LOW:MEXT}}^{(2)}$	累積クロックロー延長時間 (マスタデバイス)	-	10	ms

- $t_{\text{LOW:SEXT}}$  は、特定のスレーブデバイスが初めての START から STOP までの 1 つのメッセージのクロックサイクルを延長できる累積時間です。別のスレーブデバイスまたはマスタもクロックを延長して、合計のクロックロー延長時間が  $t_{\text{LOW:SEXT}}$  より大きくなる場合があります。したがって、このパラメータは、スレーブデバイスをフルスピードのマスタの単独のターゲットとして測定されます。
- $t_{\text{LOW:MEXT}}$  は、マスタがメッセージの各バイト内のクロックサイクルを START-to-ACK、ACK-to-ACK、または ACK-to-STOP から定義に従って延長できる累積時間です。スレーブデバイスまたは別のマスタもクロックを延長して、合計のクロックロー時間が  $t_{\text{LOW:MEXT}}$  より大きくなる場合があります。したがって、このパラメータは、フルスピードスレーブデバイスをマスタの単独のターゲットとして測定されます。

図 315.  $t_{\text{LOW:SEXT}}$ 、 $t_{\text{LOW:MEXT}}$  のタイムアウト間隔



MS19866V1

## バスアイドル検出

マスタは、クロックおよびデータ信号が  $t_{\text{HIGH,MAX}}$  より大きい  $t_{\text{IDLE}}$  の間ハイであった場合、バスはフリーであるとみなすことができます (表 145 : I2C-SMBUS 仕様のクロックタイミングを参照してください)

このタイミングパラメータは、マスタがバスに動的に追加し、SMBCLK または SMBDAT ラインで状態遷移を検出しなかった可能性のあるコンディションをカバーします。この場合、マスタは十分に長い時間待って、転送が進行中でないことを確認する必要があります。このペリフェラルは、ハードウェアバスアイドル検出をサポートします。

## 28.4.11 SMBus 初期化

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 28.3 : I2C の実装を参照してください。

SMBus 通信を行うためには、I2C 初期化に加えて、他にも特定の初期化を行う必要があります。

### 受信コマンドおよびデータ確認応答制御 (スレーブモード)

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C\_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、スレーブバイト制御モード (828 ページ) を参照してください。

### 特定アドレス (スレーブモード)

必要な場合は、特定の SMBus アドレスを有効にしてください。詳細については、バスアイドル検出 (851 ページ) を参照してください。

- SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C\_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。
- SMBus ホストアドレス (0b0001 000) は、I2C\_CR1 レジスタの SMBHEN ビットをセットすることによって有効になります。
- アラート応答アドレス (0b0001100) は、I2C\_CR1 レジスタの ALERTEN ビットをセットすることで有効になります。

### パケットエラーチェック

PEC 計算を有効にするには、I2C\_CR1 レジスタの PECEN ビットをセットします。その場合、PEC 転送はハードウェアバイトカウンタ (I2Cx\_CR2 レジスタの NBYTES[7:0]) を使用して管理されます。PECEN ビットは、I2C を有効にする前に設定する必要があります。

PEC 転送はハードウェアバイトカウンタによって管理されるので、スレーブモードで SMBus とインタフェースするときには SBC ビットをセットする必要があります。PEC は、PECBYTE ビットがセットされ、RELOAD ビットがクリアされたとき、NBYTES-1 データの転送後に転送されます。RELOAD がセットされた場合、PECBYTE は効果がありません。

**注意 :** I2C が有効なときには、PECEN 設定の変更はできません。

表 149. SMBUS の PEC 設定

モード	SBC ビット	RELOAD ビット	AUTOEND ビット	PECBYTE ビット
マスタ Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
マスタ Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
スレーブ Tx/Rx と PEC	1	0	x	1

## タイムアウト検出

タイムアウト検出は、I2C\_TIMEOUTR レジスタの TIMOUTEN および TEXTEN ビットをセットすることによって有効になります。SMBus 仕様バージョン 2.0 で指定された最大時間の前にタイムアウトを検出するようにタイマをプログラムする必要があります。

- t<sub>TIMEOUT</sub> チェック

t<sub>TIMEOUT</sub> チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] ビットを t<sub>TIMEOUT</sub> パラメータをチェックするためにタイマ再ロード値でプログラムする必要があります。SCL ローレベルタイムアウトを検出するためには、TIDLE ビットを 0 に設定する必要があります。

その場合、タイマは、I2C\_TIMEOUTR レジスタの TIMOUTEN をセットすることによって有効になります。

SCL が (TIMEOUTA+1) × 2048 × t<sub>I2CCLK</sub> より長い時間、ローに設定された場合、I2C\_ISR レジスタの TIMEOUT フラグがセットされます。

表 150 : さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 t<sub>TIMEOUT</sub> = 25 ms) を参照してください。

**注意 :** **TIMEOUTEN ビットがセットされているときには、TIMEOUTA[11:0] ビットおよび TIDLE ビットの設定変更はできません。**

- t<sub>LOW:SEXT</sub> および t<sub>LOW:MEXT</sub> チェック

ペリフェラルがマスタとして設定されているか、スレーブとして設定されているかに応じて、12 ビットの TIMEOUTB タイマは、スレーブの場合は t<sub>LOW:SEXT</sub> をチェックするために、マスタの場合は t<sub>LOW:MEXT</sub> をチェックするために、設定する必要があります。標準では最大値のみが規定されているので、両方について同じ値を選ぶことができます。

その場合、タイマは、I2C\_TIMEOUTR レジスタの TEXTEN ビットをセットすることによって有効になります。

SMBus ペリフェラルが、(TIMEOUTB+1) × 2048 × t<sub>I2CCLK</sub> より長い時間および **バスアイドル検出 (851 ページ)** セクションで述べられているタイムアウト間隔で、累積 SCL ストレッチを実行した場合、I2C\_ISR レジスタの TIMEOUT フラグがセットされます。

表 151 : さまざまな I2CCLK 周波数での TIMEOUTB の設定例を参照してください。

**注意 :** **TEXTEN ビットがセットされているときには、TIMEOUTB 設定変更はできません。**

## バスアイドル検出

t<sub>IDLE</sub> チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] フィールドを t<sub>IDLE</sub> パラメータを得るためにタイマ再ロード値でプログラムする必要があります。SCL および SDA ハイレベルタイムアウトを検出するためには、TIDLE ビットを 1 に設定する必要があります。

その場合、タイマは、I2C\_TIMEOUTR レジスタの TIMEOUTEN ビットをセットすることによって有効になります。

SCL および SDA の両方のラインが (TIMEOUTA+1) × 4 × t<sub>I2CCLK</sub> より長い間ハイのままであった場合、I2C\_ISR レジスタの TIMEOUT フラグがセットされます。

表 152 : さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値  $t_{IDLE} = 50 \text{ ms}$ ) を参照してください。

**注意 :** TIMEOUTEN がセットされているときに、TIMEOUTA および TIDLE 設定を変更することはできません。

## 28.4.12 SMBus : I2C\_TIMEOUTR レジスタの設定例

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 28.3 : I2C の実装を参照してください。

- $t_{TIMEOUT}$  の最大時間を 25 ms に設定 :

表 150. さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値  $t_{TIMEOUT} = 25 \text{ ms}$ )

$f_{I2CCLK}$	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	$t_{TIMEOUT}$
8 MHz	0x61	0	1	$98 \times 2048 \times 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 \times 2048 \times 62.5 \text{ ns} = 25 \text{ ms}$
48 MHz	0x249	0	1	$586 \times 2048 \times 20.08 \text{ ns} = 25 \text{ ms}$

- $t_{LOW:SEXT}$  および  $t_{LOW:MEXT}$  の最大時間を 8 ms に設定 :

表 151. さまざまな I2CCLK 周波数での TIMEOUTB の設定例

$f_{I2CCLK}$	TIMEOUTB[11:0] ビット	TEXTEN ビット	$t_{LOW:EXT}$
8 MHz	0x1F	1	$32 \times 2048 \times 125 \text{ ns} = 8 \text{ ms}$
16 MHz	0x3F	1	$64 \times 2048 \times 62.5 \text{ ns} = 8 \text{ ms}$
48 MHz	0xBB	1	$188 \times 2048 \times 20.08 \text{ ns} = 8 \text{ ms}$

- $t_{IDLE}$  の最大時間を 50  $\mu\text{s}$  に設定

表 152. さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値  $t_{IDLE} = 50 \text{ ms}$ )

$f_{I2CCLK}$	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	$t_{TIDLE}$
8 MHz	0x63	1	1	$100 \times 4 \times 125 \text{ ns} = 50 \text{ } \mu\text{s}$
16 MHz	0xC7	1	1	$200 \times 4 \times 62.5 \text{ ns} = 50 \text{ } \mu\text{s}$
48 MHz	0x257	1	1	$600 \times 4 \times 20.08 \text{ ns} = 50 \text{ } \mu\text{s}$

## 28.4.13 SMBus スレーブモード

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 28.3 : I2C の実装を参照してください。

I2C スレーブ転送管理 (セクション 28.4.7 : I2C スレーブモードを参照) に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

### SMBus スレーブトランスミッタ

IP が SMBus で使用されるときには、SBCは、プログラムされたデータバイト数の終わりの PEC 送信を可能にするため、1 にプログラムする必要があります。PECBYTE ビットがセットされているときには、NBYTES[7:0] でプログラムされたバイト数には PEC 送信が含まれます。その場合、TXIS 割り込みの合計数は NBYTES-1 であり、NBYTES-1 データ転送後にマスタが追加のバイトをリクエストした場合、I2C\_PECR レジスタの内容が自動的に送信されます。

**注意 :** PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 316. N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート

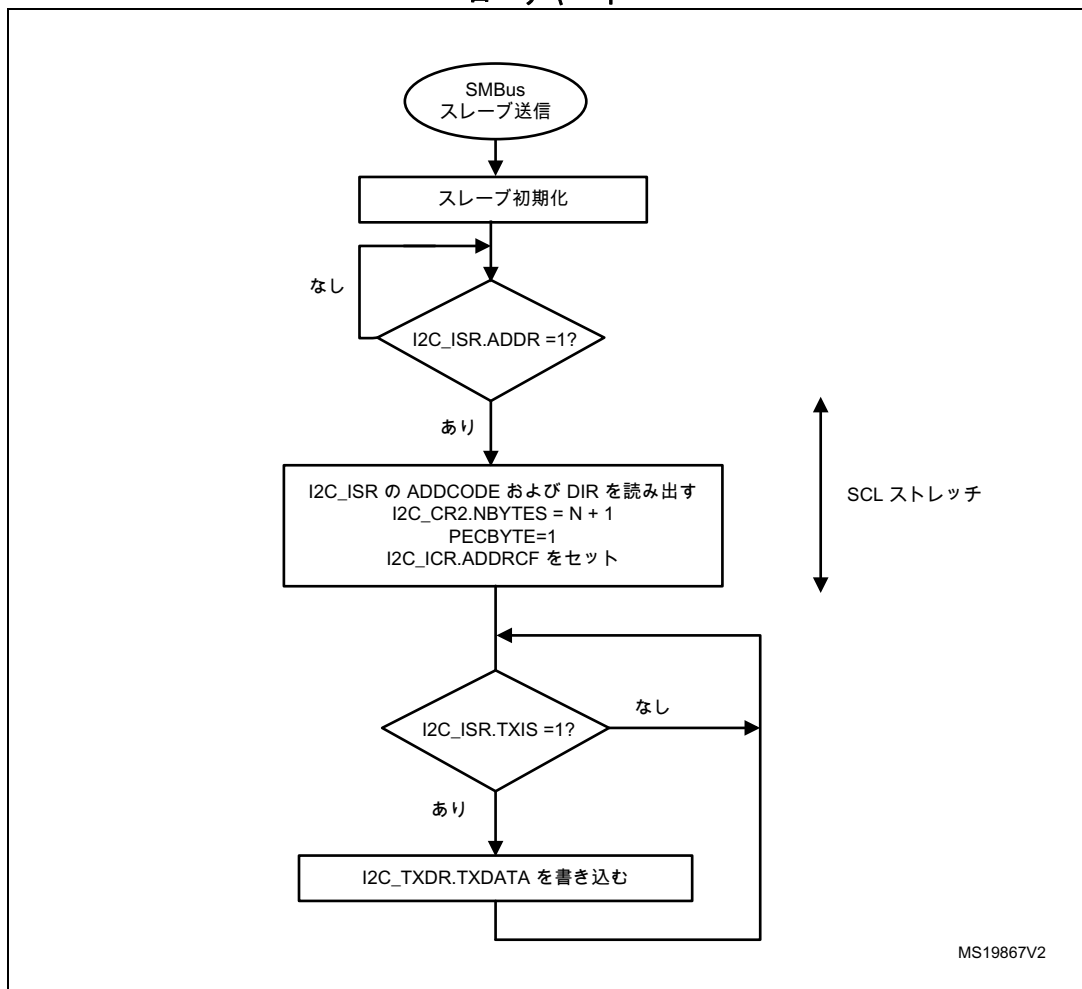
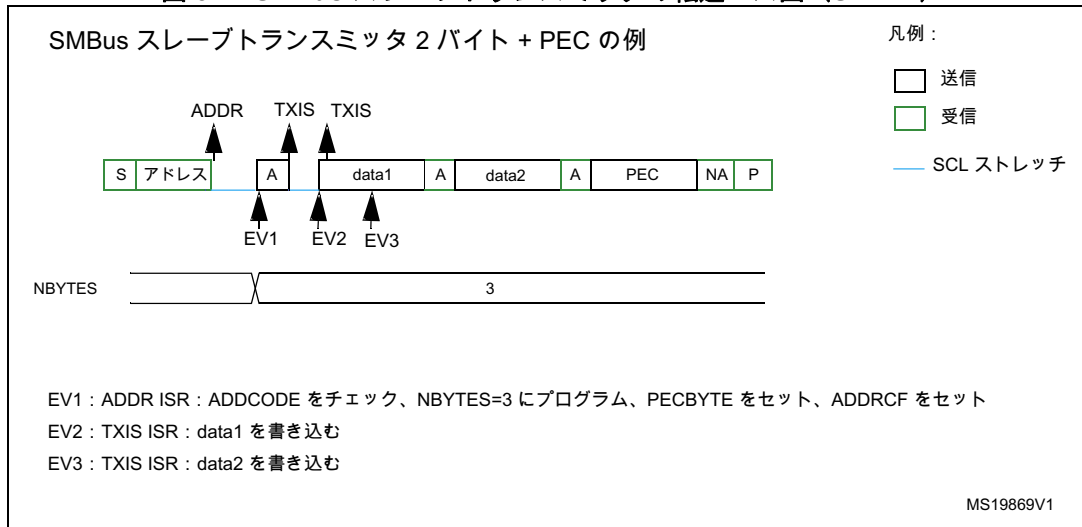


図 317. SMBus スレーブトランスミッタの転送バス図 (SBC=1)



## SMBus スレーブレシーバ

I2C が SMBus モードで使用される際には、SBC は、プログラムされたデータバイト数の終わりの PEC チェックを可能にするため、1 にプログラムする必要があります。各バイトの ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD=1)。詳細については、[スレーブバイト制御モード \(828 ページ\)](#) を参照してください。

PEC バイトをチェックするためには、RELOAD ビットをクリアして、PECBYTE ビットをセットする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが内部 I2C\_PECR レジスタの内容と比較されます。ACK ビットの値にかかわらず、比較が一致しなかった場合は NACK が自動的に生成され、比較が一致した場合は ACK が自動的に生成されます。PEC バイトが受信されると、他のデータと同様に I2C\_RXDR レジスタにコピーされ、RXNE フラグがセットされます。

PEC 不一致の場合、PECERR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

ACK ソフトウェア制御が不要な場合は、PECBYTE=1 をプログラムし、同じ書き込み操作で NBYTES を連続フローで受信するバイト数にプログラムします。NBYTES-1 が受信された後、次の受信バイトが PEC であるかどうかチェックされます。

**注意：** PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 318. N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート

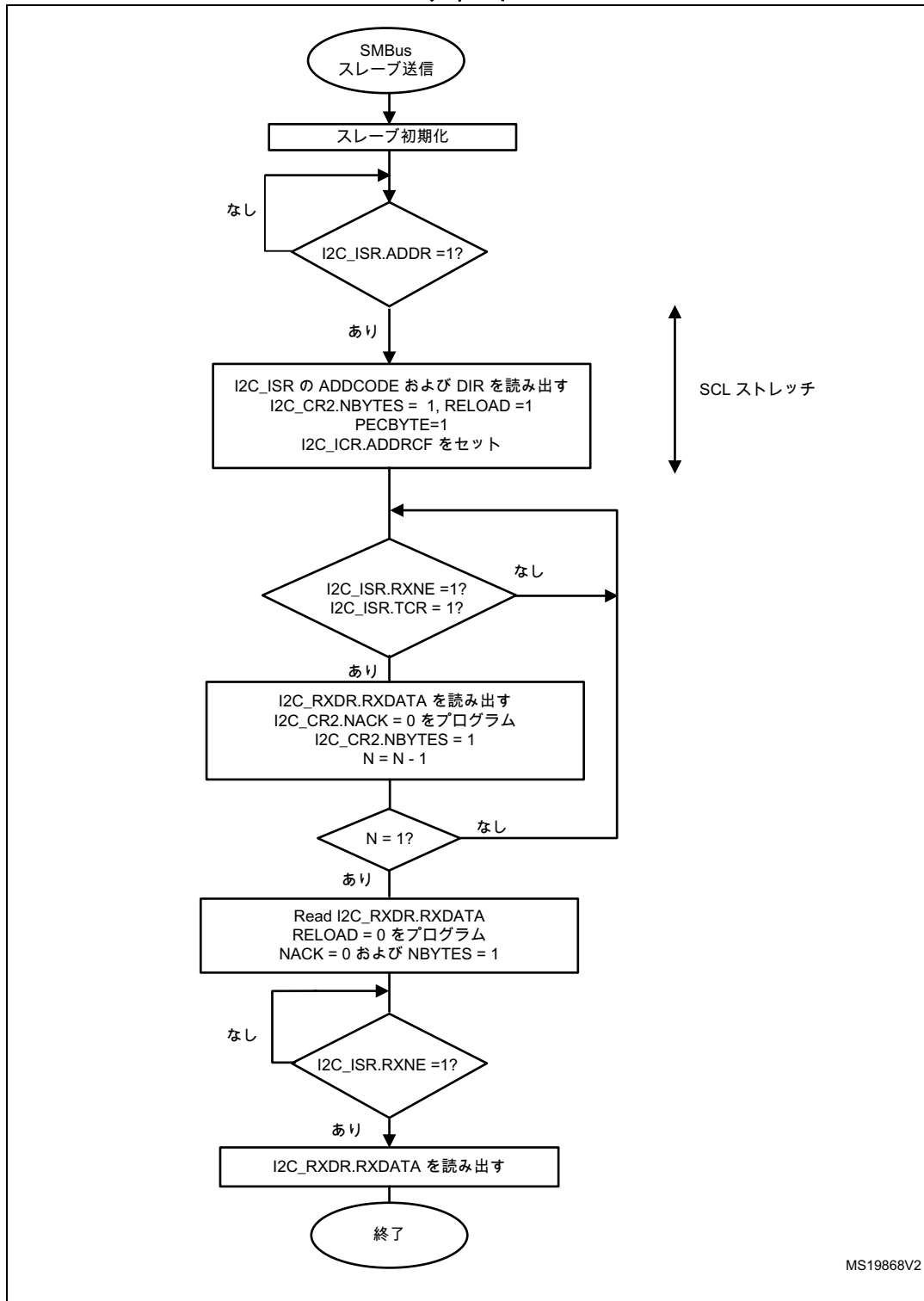
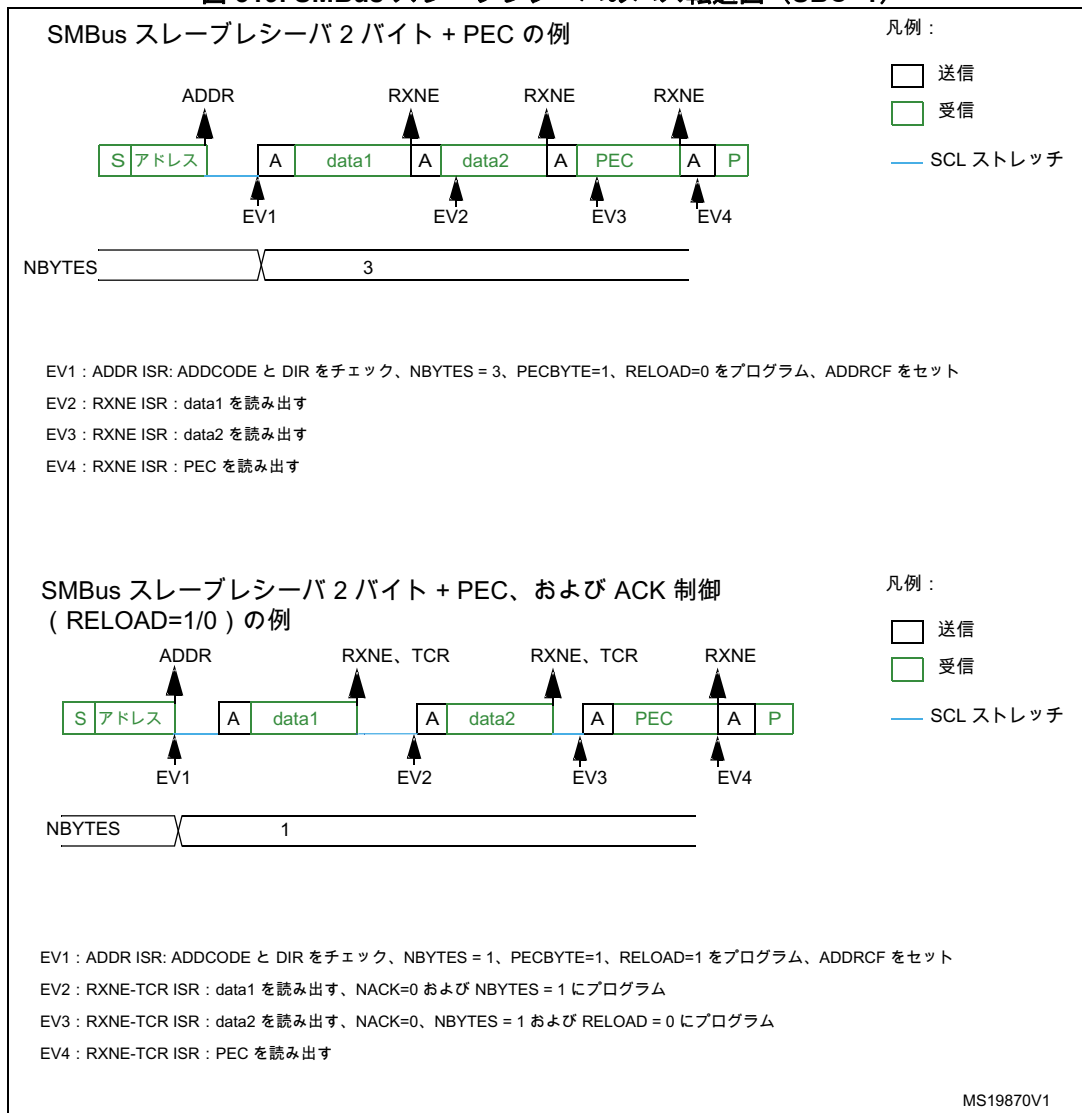




図 319. SMBus スレーブレシーバのバス転送図 (SBC=1)



このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 28.3 : I2C の実装](#) を参照してください。

I2C マスタ転送管理 ([セクション 28.4.8 : I2C マスタモード](#) を参照) に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

## SMBus マスタトランスミッタ

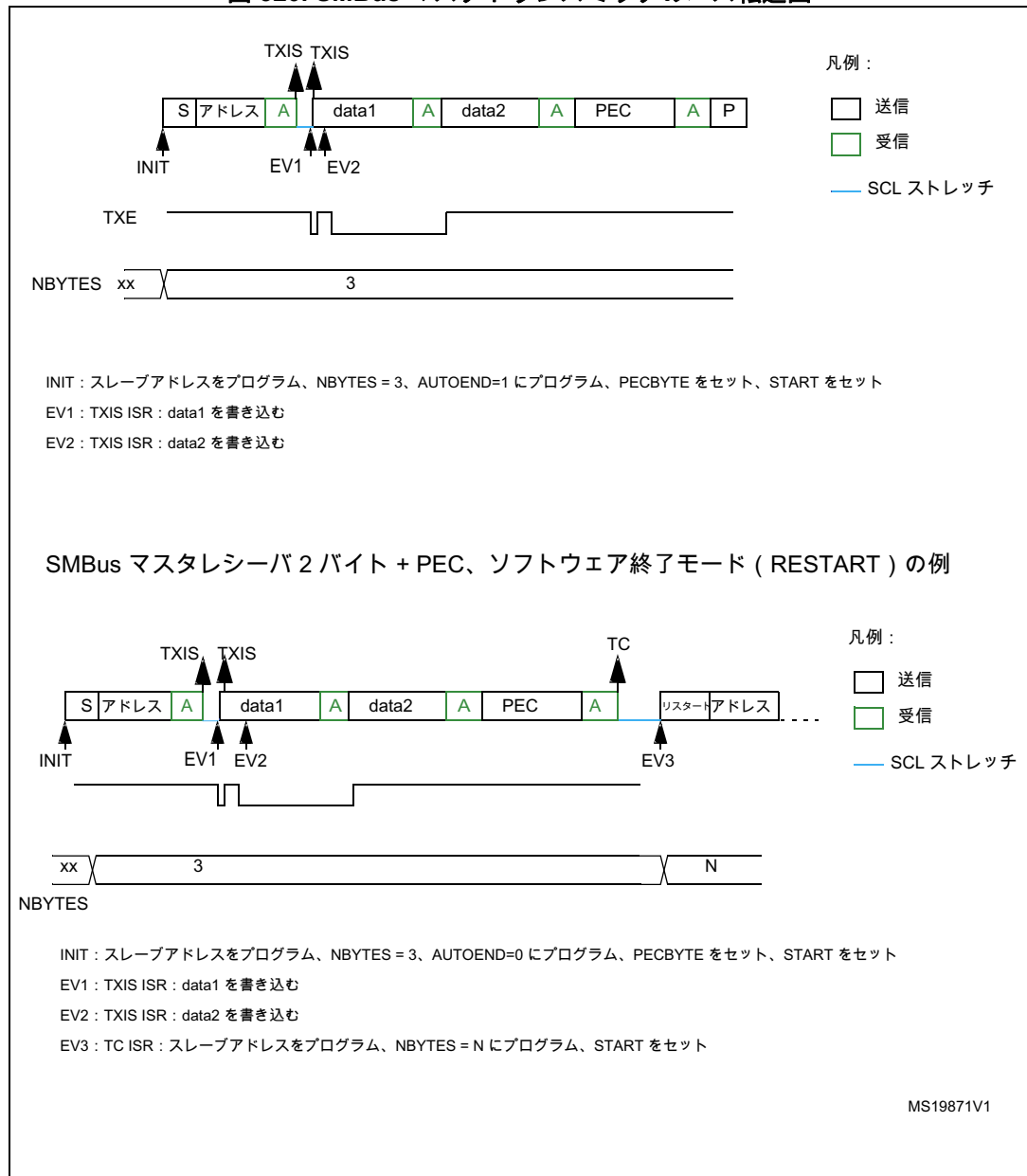
SMBus マスタが PEC を送信したいときには、START ビットをセットする前に、PECBYTE ビットをセットする必要があり、バイト数を NBYTES[7:0] フィールドでプログラムする必要があります。この場合、TXIS 割り込みの合計数は NBYTES-1 になります。したがって、NBYTES=0x1 のときに PECBYTE ビットがセットされた場合、I2C\_PECR レジスタの内容が自動的に送信されます。

SMBus マスタが PEC 後に STOP コンディションを送信したい場合は、自動終了モードを選択してください (AUTOEND=1)。この場合、PEC 送信に続いて、STOP コンディションが自動的に送信されます。

SMBus マスタが PEC 後に RESTART コンディションを送信したい場合は、ソフトウェアモードを選択してください (AUTOEND=0)。この場合、NBYTES-1 が送信されると、PEC 送信後に I2C\_PECR レジスタの内容が送信され、TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションを TC 割り込みサブルーチンでプログラムする必要があります。

**注意：** PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。

図 320. SMBus マスタトランスミッタのバス転送図



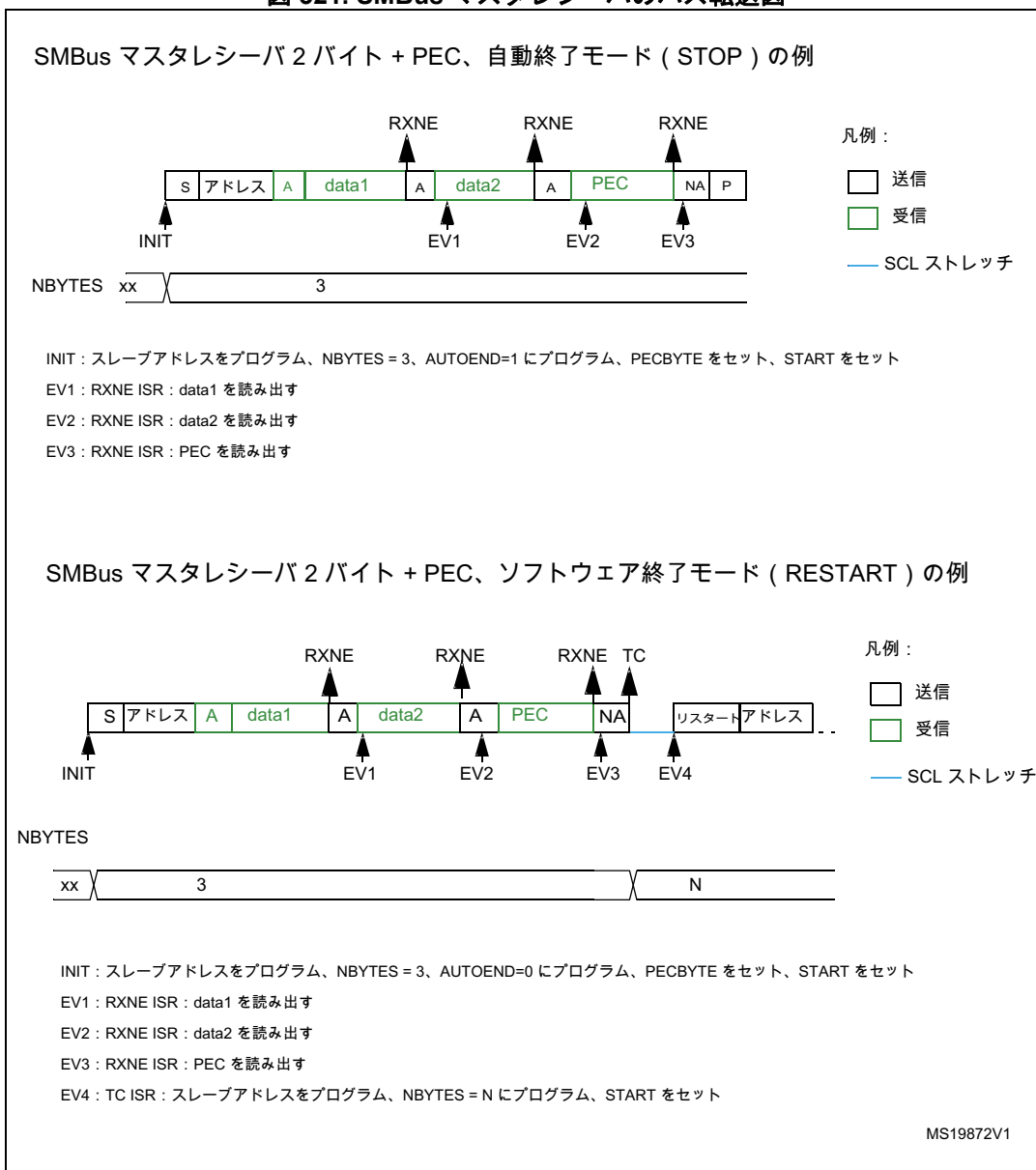
## SMBus マスタレシーバ

SMBus マスタが転送終了時に PEC を受信してから STOP を受信したいときには、自動終了モードを選択できます (AUTOEND=1)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが I2C\_PECR レジスタの内容と自動的に照合されます。PEC バイトに対して NACK 応答が与えられた後、STOP コンディションが送信されます。

SMBus マスタが転送終了時に PEC バイトを受信してから RESTART を受信したいときには、ソフトウェアモードを選択する必要があります (AUTOEND=0)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが I2C\_PECR レジスタの内容と自動的に照合されます。PEC バイト受信後に TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションは、TC 割り込みサブルーチンでプログラムできます。

**注意：** *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 321. SMBus マスタレシーバのバス転送図



## 28.4.14 アドレス一致時に STOP モードからウェイクアップ

このセクションは、STOP モードからのウェイクアップ機能がサポートされるときにのみ適用されます。[セクション 28.3 : I2C の実装](#) を参照してください。

I2C は、アドレス指定されたとき、MCU を STOP モードからウェイクアップできます (APB クロックはオフ)。すべてのアドレッシングモードがサポートされます。

STOP モードからのウェイクアップを有効にするには、I2C\_CR1 レジスタの WUPEN ビットをセットします。STOP モードからのウェイクアップを可能にするには、HSI オシレータを I2CCLK のクロックソースとして選択する必要があります。

STOP モード中、HSI はオフです。START が検出されると、I2C インタフェースは HSI をオンに切り替えて、HSI がウェイクアップするまで SCL ローをストレッチします。

HSI は、アドレス受信に使用されます。

アドレス一致の場合、I2C は、MCU のウェイクアップ時間の間、SCLローをストレッチします。ストレッチは、ADDR フラグがソフトウェアによってクリアされたときにリリースされ、転送は通常通りに続行されます。

アドレスが一致しなかった場合、HSI は再びオフになり、MCU はウェイクアップしません。

**注 :** I2C クロックがシステムクロックの場合、または WUPEN = 0 の場合、START 受信後も HSI オシレータはオンになりません。

**ADDR 割り込みによってのみ、MCU をウェイクアップできます。したがって、I2C がマスタとして、または ADDR フラグのセット後にアドレス指定されたスレーブとして転送を行っているときには、STOP モードに入らないでください。これを管理するには、ADDR 割り込みルーチンで SLEEPDEEP ビットをクリアして、STOPF フラグのセット後のみ再びオンにセットします。**

**注意 :** デジタルフィルタは、STOP モードからのウェイクアップ機能と互換性がありません。DNF ビットが 0 でない場合、WUPEN ビットをセットしても効果はありません。

**注意 :** この機能は、I2C クロックソースが HSI オシレータのときのみ使用できます。

**注意 :** STOP モードからのウェイクアップ機能の正しい動作を保証するには、クロックストレッチを有効にする必要があります (NOSTRETCH=0)。

**注意 :** STOP モードからのウェイクアップが無効な場合 (WUPEN=0)、STOP モードに入る前に、I2C ペリフェラルを無効にする必要があります (PE=0)。

## 28.4.15 エラー条件

以下は、通信エラーを引き起こす可能性のあるエラー条件です。

### バスエラー (BERR)

バスエラーは、START または STOP コンディションが検出され、複数の 9 SCL クロックパルス後になかったときに検出されます。START または STOP コンディションは、SCL がハイとのときに SDA エッジが発生した場合に検出されます。

バスエラーフラグは、I2C がマスタまたはアドレス指定されたスレーブとして転送に関与する場合のみ (すなわち、スレーブモードのアドレスフェーズでないとき)、セットされます。

スレーブモードで START または RESTART の誤配置が検出された場合、I2C は、正しい START コンディションの場合と同様に、アドレス認識状態に入ります。

バスエラーが検出されると、I2C\_ISR レジスタの BERR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。



## アービトレーション喪失 (ARLO)

アービトレーション喪失は、SDA ラインでハイレベルが送信されたが、SCL 立ち上がりエッジでローレベルがサンプリングされたときに検出されます。

- マスタモードでは、アービトレーション喪失は、アドレスフェーズ、データフェーズ、およびデータ確認応答フェーズで検出されます。この場合、SDA および SCL ラインはリリースされ、START 制御ビットがハードウェアによってクリアされ、マスタは自動的にスレーブモードに切り替わります。
- スレーブモードでは、アービトレーション喪失は、データフェーズとデータ確認応答フェーズで検出されます。この場合、転送は中止され、SCL および SDA ラインがリリースされます。

アービトレーション喪失が検出されると、I2C\_ISR レジスタの ARLO フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## オーバーラン/アンダーランエラー (OVR)

オーバーランまたはアンダーラインエラーは、スレーブモードで NOSTRETCH=1 のとき、および次のときに検出されます：

- 受信時、新しいバイトが受信され、RXDR レジスタがまだ読み出されていないとき。新しい受信バイトは失われ、新しいバイトへの応答として NACK が自動的に送信されます。
- 送信時：
  - STOPF=1 のときには、最初のデータバイトが送信されなければなりません。TXE=0、0xFF の場合、I2C\_TXDR レジスタの内容が送信され、そうでない場合は送信されません。
  - 新しいバイトが送信されるべきときに、I2C\_TXDR レジスタがまだ書き込まれていなかった場合、0xFF が送信されます。

オーバーランまたはアンダーランエラーが検出されると、I2C\_ISR レジスタの OVR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## パケットエラーチェックエラー (PECERR)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 28.3 : I2C の実装](#) を参照してください。

PEC エラーは、受信した PEC バイトが I2C\_PECR レジスタの内容と一致しなかったときに検出されます。正しくない PEC の受信後、NACK が自動的に送信されます。

PEC エラーが検出されると、I2C\_ISR レジスタの PECERR フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## タイムアウトエラー (TIMEOUT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 28.3 : I2C の実装](#) を参照してください。

タイムアウトエラーは、次のような条件で発生します：

- TIDLE=0 であり、SCL が TIMEOUTA[11:0] ビットで定義された時間だけローのままであった場合：これは SMBus タイムアウトの検出に使用されます。
- TIDLE=1 であり、SDA および SCL が TIMEOUTA[11:0] ビットで定義された時間だけハイのままであった場合：これはバスアイドル状態の検出に使用されます。
- マスタ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合 (SMBus  $t_{LOW:MEXT}$  パラメータ)。
- スレーブ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合 (SMBus  $t_{LOW:SEXT}$  パラメータ)。

マスタモードでタイムアウト違反が検出されると、STOP コンディションが自動的に送信されます。

スレーブモードでタイムアウト違反が検出されると、SDA および SCL ラインが自動的にリリースされます。

タイムアウトエラーが検出されると、I2C\_ISR レジスタの TIMEOUT フラグがセットされ、I2C\_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

## アラート (ALERT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 28.3 : I2C の実装](#) を参照してください。

ALERT フラグは、I2C インタフェースがホストとして設定され (SMBHEN=1)、アラートピン検出が有効であり (ALERTEN=1)、SMBA ピンで立ち下がりがエッジが検出されたときにセットされます。I2C\_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。

## 28.4.16 DMA リクエスト

### DMA を使用した送信

送信について DMA (Direct Memory Access) を有効にするには、I2C\_CR1 レジスタの TXDMAEN ビットをセットします。TXIS ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 13 : ダイレクトメモリアクセスコントローラ \(DMA\) \(261 ページ\)](#)) を参照) を使用して設定された SRAM 領域から I2C\_TXDR レジスタにロードされます。

データのみが DMA で転送されます。

- マスタモード : 初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます (送信されたスレーブアドレスを DMA で転送することはできません)。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。 [マスタトランスミッタ \(839 ページ\)](#) を参照してください。
- スレーブモードでは :
  - NOSTRETCH=0 では、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
  - NOSTRETCH=1 では、アドレス一致イベントの前に DMA を初期化する必要があります。
- SMBus をサポートする場合 : PEC 転送は NBYTES カウンタによって管理されます。 [SMBus スレーブトランスミッタ \(854 ページ\)](#) および [SMBus マスタトランスミッタ \(857 ページ\)](#) を参照してください。

**注 :** DMA が送信に使用される場合、TXIE ビットが有効である必要はありません。

### DMA を使用した受信

受信について DMA (Direct Memory Access) を有効にするには、I2C\_CR1 レジスタの RXDMAEN ビットをセットします。RXNE ビットがセットされているときには、データは、I2C\_RXDR レジスタから DMA ペリフェラル ([セクション 13 : ダイレクトメモリアクセスコントローラ \(DMA\) \(261 ページ\)](#)) を参照) を使用して設定された SRAM 領域にロードされます。データのみ (PEC を含む) が DMA で転送されます。

- マスタモード、初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。



- NOSTRETCH=0 のスレーブモードでは、すべてのデータが DMA を使用して転送される時には、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
- SMBus がサポートされる場合（[セクション 28.3 : I2C の実装](#) を参照）：PEC 転送は NBYTES カウンタによって管理されます。[SMBus スレーブレシーバ \(855 ページ\)](#) および [SMBus マスタレシーバ \(859 ページ\)](#) を参照してください。

注： DMA が受信に使用される場合、RXIE ビットが有効である必要はありません。

## 28.4.17 デバッグモード

マイクロコントローラがデバッグモードに入ると（コア停止）、DBG モジュールの DBG\_I2Cx\_SMBUS\_TIMEOUT 設定ビットに応じて、SMBus タイムアウトは、通常の動作を続行するか、あるいは停止します。

## 28.5 I2C 低電力モード

表 153. 低電力モード

モード	説明
SLEEP	影響なし。 I2C 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	I2C レジスタの内容は保たれます。
STANDBY	I2C ペリフェラルはパワーダウンされ、STANDBY 終了後に再初期化する必要があります。

## 28.6 I2C 割り込み

次の表に、I2C 割り込みリクエストの一覧を示します。

表 154. I2C 割り込みリクエスト

割り込みイベント	イベントフラグ	イベントフラグ/ 割り込みのクリア方法	割り込み有効制御 ビット
受信バッファノットエンプティ	RXNE	I2C_RXDR レジスタを 読み出す	RXIE
送信バッファ割り込みステータス	TXIS	I2C_TXDR レジスタに 書き込む	TXIE
STOP 検出割り込みフラグ	STOPF	STOPCF=1 を書き込む	STOPIE
転送完了再ロード	TCR	I2C_CR2 の NBYTES[7:0] に 0 以外 を書き込む	TCIE
転送完了	TC	START=1 または STOP=1 を書き込む	
アドレス一致	ADDR	ADDRCF=1 を書き込む	ADDRIE
NACK 受信	NACKF	NACKCF=1 を書き込む	NACKIE



表 154. I2C 割り込みリクエスト (続き)

割り込みイベント	イベントフラグ	イベントフラグ/ 割り込みのクリア方法	割り込み有効制御 ビット
バスエラー	BERR	BERRCF=1 を書き込む	ERRIE
アービトレーション喪失	ARLO	ARLOCF=1 を書き込む	
オーバーラン/アンダーラン	OVR	OVRCF=1 を書き込む	
PEC エラー	PECERR	PECERRCF=1 を書き込む	
タイムアウト/ $t_{LOW}$ エラー	TIMEOUT	TIMEOUTCF=1 を書き込む	
SMBus アラート	ALERT	ALERTCF=1 を書き込む	

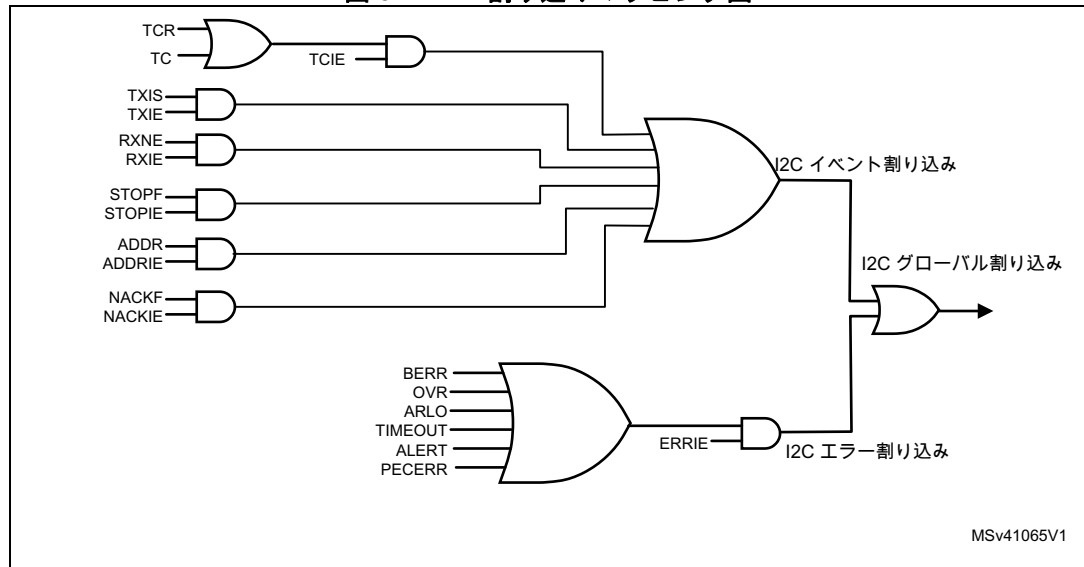
製品実装に応じて、これらの割り込みイベントのすべてで同じ割り込みベクトルを共有するか (I2C グローバル割り込み)、2つの割り込みベクトル (I2C イベント割り込みと I2C エラー割り込み) にグループ化することができます。詳細については、セクション: 割り込みベクタと例外ベクタを参照してください。

I2C 割り込みを有効にするには、次のシーケンスが必要です。

1. NVIC で I2C IRQ チャンネルを設定し、有効にします。
2. 割り込みを生成するように I2C を設定します。

I2C ウェイクアップイベントは EXTI コントローラに接続されます (セクション 14.2: 拡張割り込み/イベントコントローラ (EXTI) を参照)。

図 322. I2C 割り込みマッピング図



## 28.7 I2C レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタは、ワード (32 ビット) 単位でアクセスされます。

### 28.7.1 制御レジスタ 1 (I2C\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times \text{PCLK1} + 6 \times \text{I2CCCLK}$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERT EN	SMBD EN	SMBH EN	GCEN	WUPE N	NOSTR ETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	Res.	ANF OFF	DNF			ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE	
rw	rw		rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **PECEN** : PEC イネーブル

- 0 : PEC 計算は無効です。
- 1 : PEC 計算は有効です。

**注 :** *SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。*

ビット 22 **ALERTEN** : SMBus アラート有効

**デバイスモード (SMBHEN=0) :**

- 0 : SMBA ピンをハイにリリースし、アラート応答アドレスヘッダを無効にします。0001100x の後に NACK が続きます。
- 1 : SMBA ピンをローに駆動し、アラート応答アドレスヘッダを有効にします : 0001100x の後に ACK が続きます。

**ホストモード (SMBHEN=1) :**

- 0 : SMBus アラートピン (SMBA) はサポートされません。
- 1 : SMBus アラートピン (SMBA) はサポートされます。

**注 :** *ALERTEN=0 のときには、SMBA ピンを標準 GPIO として使用できます。*

*SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。*

ビット 21 **SMBDEN** : SMBus デバイスデフォルトアドレス有効

- 0 : デバイスデフォルトアドレス無効。アドレス 0b1100001x は NACK されます。
- 1 : デバイスデフォルトアドレス有効。アドレス 0b1100001x は ACK されます。

**注 :** *SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。*

- ビット 20 **SMBHEN** : SMBus ホストアドレス有効
- 0 : ホストアドレス無効。アドレス 0b0001000x は NACK されます。
  - 1 : ホストアドレス有効。アドレス 0b0001000x は ACK されます。
- 注 :** **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。**
- ビット 19 **GCEN** : 同報イネーブル
- 0 : 同報は無効です。アドレス 0b00000000 は NACK されます。
  - 1 : 同報は有効です。アドレス 0b00000000 は ACK されます。
- ビット 18 **WUPEN** : STOP モードからのウェイクアップ有効
- 0 : STOP モードからのウェイクアップ無効。
  - 1 : STOP モードからのウェイクアップ有効。
- 注 :** **STOP モードからのウェイクアップ機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。**
- 注 :** **WUPEN は、DNF = 0000 のときのみセットできます。**
- ビット 17 **NOSTRETCH** : クロックストレッチ無効
- このビットは、スレープモードでのクロックストレッチを無効にするために使用されます。マスタモードではクリアされたままでなければなりません。
- 0 : クロックストレッチ有効
  - 1 : クロックストレッチ無効
- 注 :** **このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。**
- ビット 16 **SBC** : スレープバイト制御
- このビットは、スレープモードでのハードウェアバイト制御を有効にするために使用されます。
- 0 : スレープバイト制御無効
  - 1 : スレープバイト制御有効
- ビット 15 **RXDMAEN** : DMA 受信リクエスト有効
- 0 : DMA モードは受信に無効
  - 1 : DMA モードは受信に有効
- ビット 14 **TXDMAEN** : DMA 送信リクエスト有効
- 0 : DMA モードは送信に無効
  - 1 : DMA モードは送信に有効
- ビット 13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **ANFOFF** : アナログノイズフィルタ OFF
- 0 : アナログノイズフィルタ有効
  - 1 : アナログノイズフィルタ無効
- 注 :** **このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。**
- ビット 11:8 **DNF[3:0]** : デジタルノイズフィルタ
- これらのビットは、SDA および SCL 入力のデジタルノイズフィルタを設定するために使用されます。デジタルフィルタは、最大  $DNF[3:0] * t_{I2CCLK}$  の長さのスパイクを除去します。
- 0000 : デジタルフィルタ無効
  - 0001 : デジタルフィルタは有効であり、最大  $1 t_{I2CCLK}$  の除去能力を持ちます。
  - ...
  - 1111 : デジタルフィルタは有効であり、最大  $15 t_{I2CCLK}$  の除去能力を持ちます。
- 注 :** **アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。このフィルタは、I2C が無効 (PE = 0) のときのみプログラムできます。**

ビット 7 **ERRIE** : エラー割り込み有効

0 : エラー検出割り込み無効

1 : エラー検出割り込み有効

**注 :** 次のようなエラーが発生すると、割り込みが生成されます :

**アービトレーション喪失 (ARLO)**

**バスエラー検出 (BERR)**

**オーバーラン/アンダーラン (OVR)**

**タイムアウト検出 (TIMEOUT)**

**PEC エラー検出 (PECERR)**

**アラートピンイベント検出 (ALERT)**

ビット 6 **TCIE** : 転送完了割り込み有効

0 : 転送完了割り込み無効

1 : 転送完了割り込み有効

**注 :** 次のようなイベントが発生すると、割り込みが生成されます :

**転送完了 (TC)**

**転送完了再ロード (TCR)**

ビット 5 **STOPIE** : STOP 検出割り込み有効

0 : STOP 検出 (STOPF) 割り込み無効

1 : STOP 検出 (STOPF) 割り込み有効

ビット 4 **NACKIE** : 非確認応答受信割り込み有効

0 : 非確認応答 (NACKF) 受信割り込み無効

1 : 非確認応答 (NACKF) 受信割り込み有効

ビット 3 **ADDRIE** : アドレス一致割り込み有効 (スレーブのみ)

0 : アドレス一致 (ADDR) 割り込み無効

1 : アドレス一致 (ADDR) 割り込み有効

ビット 2 **RXIE** : RX 割り込み有効

0 : 受信 (RXNE) 割り込み無効

1 : 受信 (RXNE) 割り込み有効

ビット 1 **TXIE** : TX 割り込み有効

0 : 送信 (TXIS) 割り込み無効

1 : 送信 (TXIS) 割り込み有効

ビット 0 **PE** : ペリフェラルは有効です。

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

**注 :** **PE=0** のとき、**I2C SCL** および **SDA** ラインはリリースされます。内部ステートマシンおよびステータスビットはリセット値に戻されます。クリアされたときには、**PE** は少なくとも 3 APB クロックサイクルの間、ローに保たれる必要があります。

## 28.7.2 制御レジスタ 2 (I2C\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rW	rW	rW							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_W RN	SADD[9:0]									
rs	rs	rs	rW	rW	rW	rW									

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **PECBYTE** : パケットエラーチェックバイト

このビットはソフトウェアによってセットされ、PEC が転送されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、また、PE=0 のとき、ハードウェアによってクリアされます。

0 : PEC 転送なし。

1 : PEC 送信/受信がリクエストされます。

**注 :** このビットに 0 を書き込んでも、効果はありません。

このビットは、RELOAD がセットされているときには効果はありません。

このビットは、SBC=0 のとき、スレーブモードでは効果はありません。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。**

ビット 25 **AUTOEND** : 自動終了モード (マスタモード)

このビットは、ソフトウェアによってセット/クリアされます。

0 : ソフトウェア終了モード : NBYTES データが転送されると TC フラグがセットされ、SCL ローをストレッチします。

1 : 自動終了モード : NBYTES データが転送されると、STOP コンディションが自動的に送信されます。

**注 :** このビットは、スレーブモードまたは RELOAD ビットがセットされているときには効果がありません。

ビット 24 **RELOAD** : NBYTES 再ロードモード

このビットは、ソフトウェアによってセット/クリアされます。

0 : 転送は、NBYTES データの転送後、完了します (STOP または RESTART が続きます)。

1 : 転送は、NBYTES データの転送後に完了しません (NBYTES が再ロードされます)。NBYTES データが転送されると TCR フラグがセットされ、SCL ローをストレッチします。

ビット 23:16 **NBYTES[7:0]** : バイト数

送受信されるバイト数は、ここでプログラムされます。このフィールドは、SBC=0 のスレーブモードでは効果がありません。

**注 :** START ビットがセットされているときに、これらのビットを変更することはできません。



### ビット 15 NACK : NACK 生成 (スレーブモード)

このビットはソフトウェアによってセットされ、NACK が送信されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、または PE=0 のとき、ハードウェアによってクリアされます。

0 : 現在の受信バイト後に ACK が送信されます。

1 : 現在の受信バイト後に NACK が送信されます。

**注 :** このビットに 0 を書き込んでも、効果はありません。

このビットは、スレーブモードでのみ使用されます : マスタレシーバモードでは、NACK ビットの値にかかわらず、STOP または RESTART コンディション前の最後のバイト後に NACK が自動的に生成されます。

スレーブレシーバ NOSTRETCH モードでオーバーランが発生すると、NACK ビットの値にかかわらず、NACK が自動的に生成されます。

ハードウェア PEC チェックが有効なとき (PECBYTE=1)、PEC 確認応答値は NACK 値に依存しません。

### ビット 14 STOP : STOP 生成 (マスタモード)

このビットはソフトウェアによってセットされ、STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

**マスタモード :**

0 : STOP 生成なし。

1 : 現在のバイト転送後の STOP 生成。

**注 :** このビットに 0 を書き込んでも、効果はありません。

### ビット 13 START : START 生成

このビットはソフトウェアによってセットされ、START とアドレスシーケンスが送信された後、アービトレーション喪失によって、タイムアウトエラー検出によって、または PE=0 のときに、ハードウェアによってクリアされます。I2C\_ICR レジスタの ADDRDCF ビットに "1" を書き込むことによって、ソフトウェアによってクリアすることもできます。

0 : START 生成なし。

1 : RESTART/START 生成 :

- I2C がすでにマスタモードであり、AUTOEND=0 の場合、このビットをセットすると、RELOAD=0 のとき、NBYTES 転送の終了後に REPEATED START コンディションが生成されます。

- そうでない場合、このビットをセットすると、バスがフリーになると、START コンディションが生成されます。

**注 :** このビットに 0 を書き込んでも、効果はありません。

バスが BUSY の場合、または I2C がスレーブモードの場合でも、START ビットをセットできません。

このビットは、RELOAD がセットされているときには効果がありません。

### ビット 12 HEAD10R : 読み出し方向のみの 10 ビットアドレスヘッダ (マスタレシーバモード)

0 : マスタは完全な 10 ビットスレーブアドレス読み出しシーケンスを送信します : START + 2 バイトの書き込み方向の 10 ビットアドレス + RESTART + 読み出し方向の 10 ビットアドレスの最初の 7 ビット。

1 : マスタは 10 ビットアドレスの最初の 7 ビットのみを送信し、その後読み出し方向を送信します。

**注 :** START ビットがセットされているときに、このビットを変更することはできません。

### ビット 11 ADD10 : 10 ビットアドレッシングモード (マスタモード)

0 : マスタは 7 ビットアドレッシングモードで動作します。

1 : マスタは 10 ビットアドレッシングモードで動作します。

**注 :** START ビットがセットされているときに、このビットを変更することはできません。

### ビット 10 RD\_WRN : 転送方向 (マスタモード)

0 : マスタは書き込み転送をリクエストします。

1 : マスタは読み出し転送をリクエストします。

**注 :** START ビットがセットされているときに、このビットを変更することはできません。

- ビット 9:8 **SADD[9:8]** : スレーブアドレスビット 9:8 (マスタモード)  
**7 ビットアドレッシングモード (ADD10=0)** :  
これらのビットは無視されます。  
**10 ビットアドレッシングモード (ADD10=1)** :  
これらのビットには、送信されるスレーブアドレスのビット 9:8 を書き込みます。  
**注:** *START ビットがセットされているときに、これらのビットを変更することはできません。*
- ビット 7:1 **SADD[7:1]** : スレーブアドレスビット 7:1 (マスタモード)  
**7 ビットアドレッシングモード (ADD10=0)** :  
これらのビットには、送信される 7 ビットのスレーブアドレスを書き込みます。  
**10 ビットアドレッシングモード (ADD10=1)** :  
これらのビットには、送信されるスレーブアドレスのビット 7:1 を書き込みます。  
**注:** *START ビットがセットされているときに、これらのビットを変更することはできません。*
- ビット 0 **SADD0** : スレーブアドレスビット 0 (マスタモード)  
**7 ビットアドレッシングモード (ADD10=0)** :  
このビットは無視されます。  
**10 ビットアドレッシングモード (ADD10=1)** :  
このビットには、送信されるスレーブアドレスのビット 0 を書き込みます。  
**注:** *START ビットがセットされているときに、これらのビットを変更することはできません。*

## 28.7.3 Own Address 1 レジスタ (I2C\_OAR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]							OA1[0]
rw					rw	rw		rw							rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA1EN** : Own Address 1 有効

- 0 : Own Address 1 無効。受信されたスレーブアドレス OA1 は NACK されます。
- 1 : Own Address 1 有効。受信されたスレーブアドレス OA1 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **OA1MODE** Own Address 1 10 ビットモード

- 0 : Own Address 1は 7 ビットアドレスです。
- 1 : Own Address 1は 10 ビットアドレスです。

**注 :** このビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 9:8 **OA1[9:8]** : インタフェースアドレス

- 7 ビットアドレッシングモード : 無視されます。
- 10 ビットアドレッシングモード : アドレスのビット 9:8。

**注 :** これらのビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 7:1 **OA1[7:1]** : インタフェースアドレス

アドレスのビット 7:1

**注 :** これらのビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 0 **OA1[0]** : インタフェースアドレス

- 7 ビットアドレッシングモード : 無視されます。
- 10 ビットアドレッシングモード : アドレスのビット 0。

**注 :** このビットは、OA1EN=0 のときのみ書き込むことができます。



## 28.7.4 Own Address 2 レジスタ (I2C\_OAR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]						Res.	
rw					rw			rw							

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA2EN** : Own Address 2 有効

- 0 : Own Address 2 無効。受信されたスレーブアドレス OA2 は NACK されます。
- 1 : Own Address 2 有効。受信されたスレーブアドレス OA2 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **OA2MSK[2:0]** : Own Address 2 マスク

- 000 : マスクなし。
- 001 : OA2[1] はマスクされ、無視されます。OA2[7:2] のみ比較されます。
- 010 : OA2[2:1] はマスクされ、無視されます。OA2[7:3] のみ比較されます。
- 011 : OA2[3:1] はマスクされ、無視されます。OA2[7:4] のみ比較されます。
- 100 : OA2[4:1] はマスクされ、無視されます。OA2[7:5] のみ比較されます。
- 101 : OA2[5:1] はマスクされ、無視されます。OA2[7:6] のみ比較されます。
- 110 : OA2[6:1] はマスクされ、無視されます。OA2[7] のみ比較されます。
- 111 : OA2[7:1] はマスクされ、無視されます。比較は行われず、すべての (予約済みを除く) 7 ビット受信アドレスが確認応答されます。

**注 :** これらのビットは、OA2EN=0 のときのみ書き込むことができます。

**OA2MSK が 0 でなくなると、予約済み I2C アドレス (0b0000xxx および 0b1111xxx) は、比較が一致した場合でも確認応答されません。**

ビット 7:1 **OA2[7:1]** : インタフェースアドレス

アドレスのビット 7:1

**注 :** これらのビットは、OA2EN=0 のときのみ書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。



## 28.7.5 タイミングレジスタ (I2C\_TIMINGR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

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

ビット 31:28 **PRESC[3:0]** : タイミングプリスケール

このフィールドは、データのセットアップおよびホールドカウンタ (*I2C のタイミング (820 ページ)* を参照) と SCL ハイおよびローレベルカウンタ (*I2C マスタ初期化 (835 ページ)* を参照) に使用されるクロック周期  $t_{PRESC}$  を生成して、I2CCLK をプリスケールするために使用されます。

$$t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$$

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **SCLDEL[3:0]** : データセットアップ時間

このフィールドは、SDA エッジと SCL 立ち上がりエッジの間に遅延  $t_{SCLDEL}$  を生成するために使用されます。マスタモードおよびスレープモード (NOSTRETCH = 0) では、SCL ラインは  $t_{SCLDEL}$  の間、ローをストレッチします。

$$t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$$

**注 :**  $t_{SCLDEL}$  は、 $t_{SU:DAT}$  タイミングを生成するために使用されます。

ビット 19:16 **SDADEL[3:0]** : データホールド時間

このフィールドは、SCL 立ち下がりエッジと SDA エッジの間に遅延  $t_{SDADEL}$  を生成するために使用されます。マスタモードおよびスレープモード (NOSTRETCH = 0) では、SCL ラインは  $t_{SCADEL}$  の間、ローをストレッチします。

$$t_{SDADEL} = SDADEL \times t_{PRESC}$$

**注 :**  $SDADEL$  は、 $t_{HD:DAT}$  タイミングを生成するために使用されます。

ビット 15:8 **SCLH[7:0]** : SCL ハイ周期 (マスタモード)

このフィールドは、マスタモードで SCL ハイ周期を生成するために使用されます。

$$t_{SCLH} = (SCLH+1) \times t_{PRESC}$$

**注 :**  $SCLH$  は、 $t_{SU:STO}$  および  $t_{HD:STA}$  タイミングを生成するためにも使用されます。

ビット 7:0 **SCLL[7:0]** : SCL ロー周期 (マスタモード)

このフィールドは、マスタモードで SCL ロー周期を生成するために使用されます。

$$t_{SCLL} = (SCLL+1) \times t_{PRESC}$$

**注 :**  $SCLL$  は、 $t_{BUF}$  および  $t_{SU:STA}$  タイミングを生成するためにも使用されます。

**注 :** このレジスタは、I2C が無効 (PE=0) のときに設定する必要があります。

**注 :** STM32CubeMX ツールは、I2C 設定ウィンドウの I2C\_TIMINGR コンテンツを計算し、提供します。

## 28.7.6 タイムアウトレジスタ (I2C\_TIMEOUTR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大  $2 \times PCLK1 + 6 \times I2CCLK$  です。

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

ビット 31 **TEXTEN** : 拡張クロックタイムアウト有効

0 : 拡張クロックタイムアウト検出は無効です。

1 : 拡張クロックタイムアウト検出は有効です。  $t_{LOW:EXT}$  を超える累積 SCL ストレッチが I2C インタフェースによって行われると、タイムアウトエラーが検出されます (TIMEOUT=1)。

ビット 30:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **TIMEOUTB[11:0]** : バスタイムアウト B

このフィールドは、累積クロック拡張タイムアウトを設定するために使用されます :

マスタモードでは、マスタ累積クロックロー拡張時間 ( $t_{LOW:MEXT}$ ) が検出されます。

スレーブモードでは、スレーブ累積クロックロー拡張時間 ( $t_{LOW:SEXT}$ ) が検出されます。

$$t_{LOW:EXT} = (TIMEOUTB+1) \times 2048 \times t_{2CCLK}$$

**注 :** これらのビットは、**TEXTEN=0** のときのみ書き込むことができます。

ビット 15 **TIMOUTEN** : クロックタイムアウト有効

0 : SCL タイムアウト検出は無効です。

1 : SCL タイムアウト検出は有効です。SCL が  $t_{TIMEOUT}$  (**TIDLE=0**) を超えてローであるか、 $t_{IDLE}$  (**TIDLE=1**) を超えてハイであった場合、タイムアウトエラーが検出されます (TIMEOUT=1)。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TIDLE** : アイドルクロックタイムアウト検出

0 : **TIMEOUTA** は、SCL ロータイムアウトの検出に使用されます。

1 : **TIMEOUTA** は、SCL と SDA の両方のハイタイムアウト (バスアイドル条件) の検出に使用されます。

**注 :** このビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

ビット 11:0 **TIMEOUTA[11:0]** : バスタイムアウト A

このフィールドは、以下を設定するために使用されます :

- **TIDLE=0** のときの SCL ロータイムアウト条件  $t_{TIMEOUT}$

$$t_{TIMEOUT} = (TIMEOUTA+1) \times 2048 \times t_{2CCLK}$$

- **TIDLE=1** のときのバスアイドル条件 (SCL と SDA の両方のハイ)

$$t_{IDLE} = (TIMEOUTA+1) \times 4 \times t_{2CCLK}$$

**注 :** これらのビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

**注 :** **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 28.3 : I2C の実装を参照してください。**

## 28.7.7 割り込みおよびステータスレジスタ (I2C\_ISR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0001

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]						DIR	
								r						r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:17 **ADDCODE[6:0]** : アドレス一致コード (スレーブモード)

これらのビットは、アドレス一致イベントが発生したときに (ADDR = 1)、受信したアドレスで更新されます。

10 ビットアドレスの場合、ADDCODE は 10 ビットのヘッダとその後のアドレスの 2 つの MSB を示します。

ビット 16 **DIR** : 転送方向 (スレーブモード)

このフラグは、アドレス一致イベントが発生したときに (ADDR=1)、更新されます。

0 : 書き込み転送、スレーブはレシーバモードになります。

1 : 読み出し転送、スレーブはトランスミッタモードになります。

ビット 15 **BUSY** : バスビジー

このフラグは、バスで通信が進行中であることを示します。START コンディションが検出されたときに、ハードウェアによってセットされます。STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERT** : SMBus アラート

このフラグは、SMBHEN=1 (SMBus ホスト設定)、ALERTEN=1、および SMBALERT イベント (立ち下がりエッジ) が SMBA ピンで検出されたときに、ハードウェアによってセットされます。ALERTCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。**

ビット 12 **TIMEOUT** : タイムアウトまたは  $t_{LOW}$  検出フラグ

このフラグは、タイムアウトまたは拡張クロックタイムアウトが発生したときに、ハードウェアによってセットされます。TIMEOUTCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。**

ビット 11 **PECERR** : 受信時の PEC エラー

このフラグは、受信した PEC が PEC レジスタの内容に一致しないときに、ハードウェアによってセットされます。正しくない PEC の受信後、NACK が自動的に送信されます。PECCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

**SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。**

ビット 10 **OVR** : オーバーラン/アンダーラン (スレーブモード)

このフラグは、NOSTRETCH=1 のスレーブモードで、オーバーラン/アンダーランエラーが発生したときに、ハードウェアによってセットされます。OVRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 9 **ARLO** : アービトレーション喪失

このフラグは、アービトレーション喪失の場合に、ハードウェアによってセットされます。ARLOCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 8 **BERR** : バスエラー

このフラグは、ペリフェラルが転送に関与しているため、START または STOP コンディションの誤配置が検出されたときに、ハードウェアによってセットされます。このフラグは、スレーブモードのアドレスフェーズではセットされません。BERRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 7 **TCR** : 転送完了再ロード

このフラグは、RELOAD=1 および NBYTES データが転送されたときに、ハードウェアによってセットされます。NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

**このフラグは、マスタモード、または SBC ビットがセットされているときのスレーブモードでのみ使用されます。**

ビット 6 **TC** : 転送完了 (マスタモード)

このフラグは、RELOAD=0、AUTOEND=0、および NBYTES データが転送されたときに、ハードウェアによってセットされます。START ビットまたは STOP ビットがセットされたときに、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 5 **STOPF** : STOP 検出フラグ

このフラグは、バス上で STOP コンディションが検出され、ペリフェラルがこの転送に関与しているときに、ハードウェアによってセットされます :

- マスタとして。ただし、STOP コンディションがペリフェラルによって生成される場合。
- または、スレーブとして。ただし、ペリフェラルがこの転送中にアドレス指定されていた場合。

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

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 4 **NACKF** : 非確認応答受信フラグ

このフラグは、バイト送信後に NACK を受信したときに、ハードウェアによってセットされます。NACKCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 3 **ADDR** : アドレス一致 (スレーブモード)

このビットは、受信したスレーブアドレスが有効なスレーブアドレスの 1 つに一致したときに、ハードウェアによってセットされます。ADDRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。



ビット 2 **RXNE** : 受信データレジスタノットエンプティ (レシーバ)

このビットは、受信データが I2C\_RXDR レジスタにコピーされ、読み出す準備ができたときに、ハードウェアによってセットされます。I2C\_RXDR が読み出されたときにクリアされます。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 1 **TXIS** : 送信割り込みステータス (トランスミッタ)

このビットは、I2C\_TXDR レジスタが空であり、送信データを I2C\_TXDR レジスタに書き込む必要があるときに、ハードウェアによってセットされます。次の送信データが I2C\_TXDR レジスタに書き込まれたときにクリアされます。

このビットは、NOSTRETCH=1 のときのみ、ソフトウェアによって 1 を書き込んで、TXIS イベントを生成することができます (TXIE=1 の場合に割り込み、または TXDMAEN=1 の場合に DMA リクエスト)。

**注:** このビットは、PE=0 のとき、ハードウェアによってクリアされます。

ビット 0 **TXE** : 送信データレジスタエンプティ (トランスミッタ)

このビットは、I2C\_TXDR レジスタが空のときに、ハードウェアによってセットされます。次の送信データが I2C\_TXDR レジスタに書き込まれたときにクリアされます。

このビットは、ソフトウェアによって 1 を書き込んで、送信データレジスタ I2C\_TXDR をフラッシュできます。

**注:** このビットは、PE=0 のとき、ハードウェアによってセットされます。

### 28.7.8 割り込みクリアレジスタ (I2C\_ICR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERT CF	TIM OUTCF	PECCF	OVRDCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDR CF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERTCF** : アラートフラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの ALERT フラグがクリアされます。

**注:** SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。

ビット 12 **TIMOUTCF** : タイムアウト検出フラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの TIMEOUT フラグがクリアされます。

**注:** SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。

ビット 11 **PECCF** : PEC エラーフラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの PECERR フラグがクリアされます。

**注:** SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 28.3 : I2C の実装を参照してください。

ビット 10 **OVRDCF** : オーバーラン/アンダーランフラグクリア

このビットに 1 を書き込むと、I2C\_ISR レジスタの OVR フラグがクリアされます。

- ビット 9 **ARLOCF** : アービトレーション喪失フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの ARLO フラグがクリアされます。
- ビット 8 **BERRCF** : バスエラーフラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの BERRF フラグがクリアされます。
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **STOPCF** : STOP 検出フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの STOPF フラグがクリアされます。
- ビット 4 **NACKCF** : 非確認応答フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの ACKF フラグがクリアされます。
- ビット 3 **ADDRCF** : アドレス一致フラグクリア  
このビットに 1 を書き込むと、I2C\_ISR レジスタの ADDR フラグがクリアされます。このビットに 1 を書き込むと、I2C\_CR2 レジスタの START ビットもクリアされます。
- ビット 2:0 予約済みであり、リセット値に保持する必要があります。

## 28.7.9 PEC レジスタ (I2C\_PECR)

アドレスオフセット : 0x20  
リセット値 : 0x0000 0000  
アクセス : ウェイト状態なし

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

- ビット 31:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7:0 **PEC[7:0]** パケットエラーチェックレジスタ  
PECEN=1 のとき、このフィールドは内部 PEC を含みます。  
PEC は、PE=0 のとき、ハードウェアによってクリアされます。

**注 :** **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 28.3 : I2C の実装 を参照してください。**



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

## 28.7.10 受信データレジスタ (I2C\_RXDR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RXDATA[7:0]** 8 ビット受信データ  
I<sup>2</sup>C バスから受信したデータバイト。

## 28.7.11 送信データレジスタ (I2C\_TXDR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TXDATA[7:0]** 8 ビット送信データ  
I<sup>2</sup>C バスに送信されるデータバイト。

**注 :** これらのビットは、TXE=1 のときのみ書き込むことができます。



## 28.7.12 I2C レジスタマップ

次の表に、I2C のレジスタマップとリセット値を示します。

表 155. I2C レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0	I2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECFEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]			ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE	
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x4	I2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]							NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]										
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0xC	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]				Res.				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	I2C_TIMINGR	PRESC[3:0]			Res.	Res.	Res.	SCLDEL[3:0]			SDADEL[3:0]			SCLH[7:0]				SCLL[7:0]															
	リセット値	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	I2C_TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]										TIMOUTEN	Res.	TIDLE	TIMEOUTA[11:0]														
	リセット値	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	I2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]						DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE	
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
0x1C	I2C_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMEOUTCF	PECFCF	OVRFCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	
	リセット値																			0	0	0	0	0	0			0	0	0			
0x20	I2C_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]						
	リセット値																									0	0	0	0	0	0	0	
0x24	I2C_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]						
	リセット値																									0	0	0	0	0	0	0	

表 155. I2C レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x28	I2C_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]													
	リセット値																										0	0	0	0	0	0	0	0					

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。

## 29 USART (Universal synchronous asynchronous receiver transmitter)

### 29.1 概要

USART (Universal Synchronous Asynchronous Receiver Transmitter) を使用すると、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟に全二重データ交換を行うことができます。USART は、プログラム可能なボーレートジェネレータを使用して、非常に広範囲のボーレートを提供します。

USART は、マルチプロセッサ通信に加え、同期式単方向通信と半二重単線通信をサポートします。また、LIN (Local Interconnect Network)、スマートカードプロトコル、および IrDA (Infrared Data Association) SIR ENDEC 仕様、およびモデム操作 (CTS/RTS) もサポートします。

マルチバッファ設定で DMA (直接メモリアクセス) を使用することによって、高速データ通信が可能です。

### 29.2 USART の主な機能

- 全二重非同期通信
- NRZ 標準フォーマット (マーク/スペース)
- 16 倍または 8 倍に設定可能なオーバーサンプリング方式によって、速度とクロック誤差の間の柔軟性を実現
- 最大 9 Mbit/s (クロック周波数が 72 MHz のとき) まで共通にプログラム可能な送信および受信ボーレートと 8 倍のオーバーサンプリング
- 以下を許可するデュアルクロックドメイン：
  - USART 機能と STOP モードからのウェイクアップ
  - PCLK 再プログラミングから独立した便利なボーレートプログラミング
- 自動ボーレート検出
- プログラム可能なデータワード長 (7 または 8 または 9 ビット)
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 設定可能なストップビット (1 または 2 個のストップビット)
- 同期通信のための同期モードとクロック出力
- 単線半二重通信
- DMA を使用した連続通信
- 送受信バイトは集中型 DMA を使用して専用 SRAM にバッファリング
- トランスミッタとレシーバ用に個別の有効ビット
- 送信と受信の信号極性を個別に制御
- スワップ可能な Tx/Rx ピン設定
- モデムと RS-485 トランシーバのハードウェアフロー制御
- 通信制御/エラー検出フラグ
- パリティ制御：
  - パリティビットの送信
  - 受信したデータバイトのパリティ検査

- 14 の割り込みソース (フラグ付き)
- マルチプロセッサ通信  
アドレスが一致しない場合、USART はミュートモードに入ります。
- ミュートモードからのウェイクアップ (アイドルライン検出またはアドレスマーク検出による)

## 29.3 USART の拡張機能

- LIN マスタの同期ブ레이크送信機能と LIN スレーブのブ레이크検出機能
  - USART が LIN 用にハードウェア設定されている場合、13 ビットのブ레이크生成と 10/11 ビットのブ레이크検出
- 通常モードで 3/16 ビット期間をサポートする IrDA SIR エンコーダデコーダ
- スマートカードモード
  - ISO/IEC 7816-3 標準で定義されているスマートカードの T=0 および T=1 非同期プロトコルをサポート
  - スマートカード動作用に 0.5 個および 1.5 個のストップビット
- Modbus 通信のサポート
  - タイムアウト機能
  - CR/LF キャラクタ認識

## 29.4 USART の実装

表 156. STM32F3xx USART の機能

USART のモード/機能 <sup>(1)</sup>	STM32F303xB/C および STM32F358xC			STM32F303xD/E および STM32F398xE			STM32F303x6/8 および STM32F328x8	
	USART1/ USART2/ USART3	UART4	UART5	USART1/ USART2/ USART3	UART4	UART5	USART1	USART2/ USART3
モデムのハードウェアフロー制御	X	-	-	X	-	-	X	X
DMA を使用した連続通信	X	X	-	X	X	-	X	X
マルチプロセッサ通信	X	X	X	X	X	X	X	X
同期モード	X	-	-	X	-	-	X	X
スマートカードモード	X <sup>(2)(3)</sup>	-	-	X <sup>(4)</sup>	-	-	X <sup>(4)</sup>	-
単線半二重通信	X	X	X	X	X	X	X	X
Ir SIR ENDEC ブロック	X	X	X	X	X	X	X	-
LIN モード	X	X	X	X	X	X	X	-
デュアルクロックドメインと STOP モードからのウェイクアップ	X	X	X	X	X	X	X	-
レシーバタイムアウト割り込み	X	X	X	X	X	X	X	-
Modbus 通信	X	X	X	X	X	X	X	-
自動ポーレート検出	X (4 モード)	-	-	X (4 モード)	-	-	X (4 モード)	-
ドライバインネーブル	X	-	-	X	-	-	X	X
USART データ長	8 および 9 ビット			7、8、および 9 ビット			7、8、および 9 ビット	

1. X: サポートされています。
2. UE ビット = 0 のとき、CK 出力は無効です。
3. STM32F303xB/C の場合、次の制限があります。USART がスマートカードモードで使用されており、カードが Answer To Reset の後でデフォルトの通信パラメータを使用できず、クロック停止にサポートしていない場合、CK でカードにクロック供給できません。この原因は、一部のパラメータを再プログラミングしている間、USART とクロック出力を無効にする必要があるためです。
4. UE ビット値に関係なく、CLKEN = 1 の場合に CK は常に使用可能です。

## 29.5 USART の機能詳細

USART の双方向通信には、少なくとも 2 本のピンが必要です。すなわち、受信データ入力 (RX) と送信データ出力 (TX) です。

- **RX** : 受信データ入力。  
これはシリアルデータ入力です。データ復旧にはオーバーサンプリング技術が使用され、有効な入力データとノイズを区別しています。
- **TX** : 送信データ出力。  
トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、送信すべきデータがないとき、TX ピンはハイレベルになります。単線 およびスマートカードモードでは、この I/O はデータの送受信に使用されます。

シリアルデータは、通常の USART モードでは、これらのピンを通じて送受信されます。フレームは、以下で構成されます。

- 送受信前のアイドルライン
- スタートビット
- データワード (7、8、または 9 ビット) LSBファースト
- フレームの完了を示す 0.5 個、1 個、1.5 個、2 個のストップビット
- USART インタフェースはボーレートジェネレータを使用
- ステータスレジスタ (USART\_ISR)
- 受信および送信データレジスタ (USART\_RDR、USART\_TDR)
- ボーレートレジスタ (USART\_BRR)
- ガードタイムレジスタ (USART\_GTPR) - スマートカードモードの場合

各ビットの定義については、[セクション 29.8 : USART レジスタ \(927 ページ\)](#) を参照してください。

同期モードおよびスマートカードモードでインタフェースするには、以下のピンが必要です。

- **CK** : クロック出力。このピンは、SPI マスタモードに対応する同期送信用のトランスミッタデータクロックを出力します (スタートビットとストップビットのクロックパルスはなく、ソフトウェアオプションで最後のデータビットのクロックパルスを送信します)。これと並行して、RX でデータを同期受信できます。これを使用して、シフトレジスタを持つペリフェラルを制御できます。クロックの位相と極性は、ソフトウェアでプログラム可能です。スマートカードモードでは、CK 出力はスマートカードにクロックを供給できます。

RS232 ハードウェアフロー制御モードでは、以下のピンが必要です。

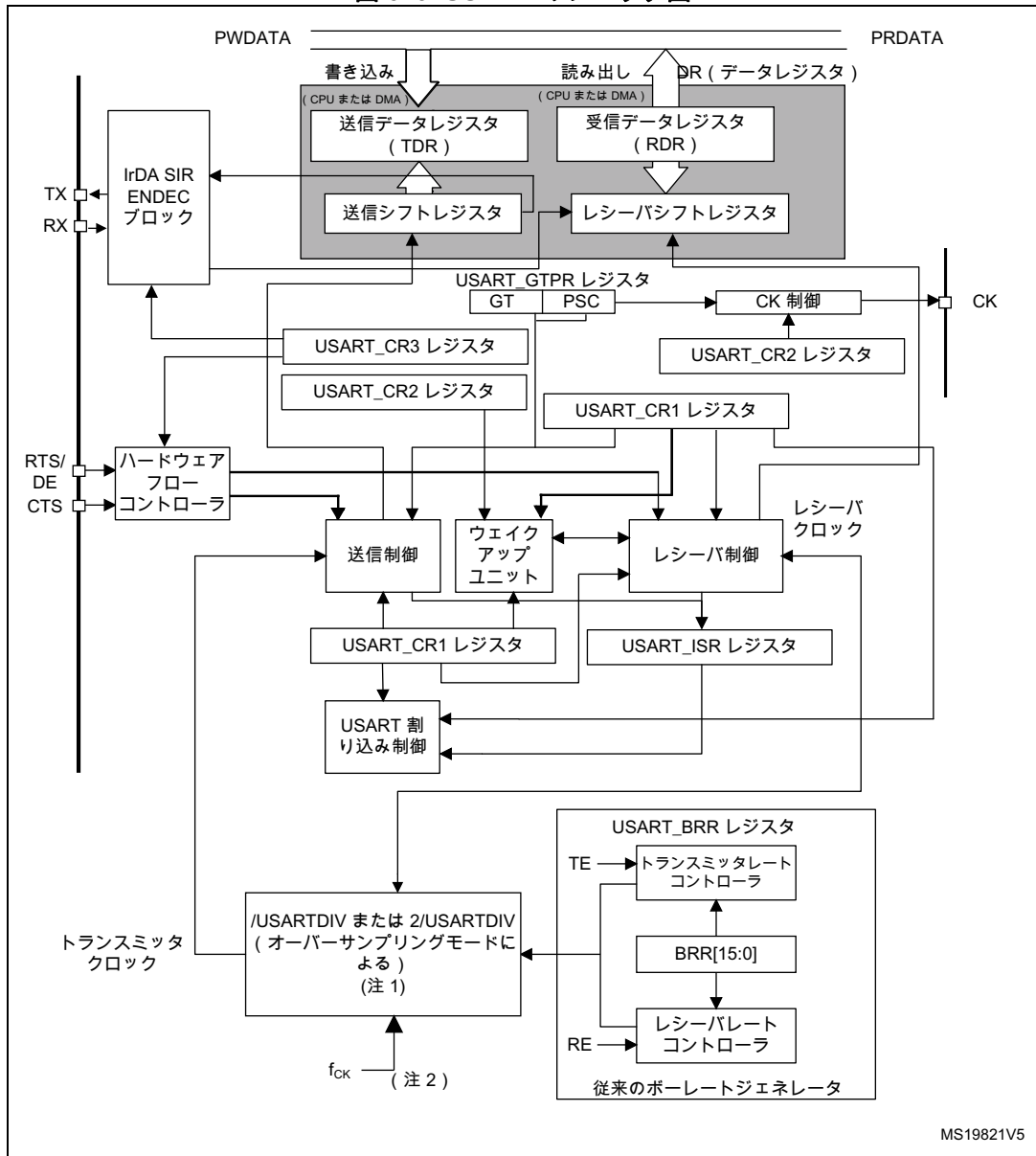
- **CTS** : Clear To Send は、ハイレベルのとき、現在の転送の終わりにデータ送信をブロックします。
- **RTS** : Request To Send は、ローレベルのとき、USART がデータを受信する準備ができたことを示します。

RS485 ハードウェアフロー制御モードでは、以下のピンが必要です。

- **DE** : Driver Enable は、外部トランシーバの送信モードを有効にします。

**注 :** **DE と RTS は同じピンを共有します。**

図 323. USART のブロック図



MS19821V5

1. USART\_BRR レジスタの USARTDIV のコーディングの詳細については、[セクション 29.5.4: USART ポーレート生成](#) を参照してください。
2. f<sub>ck</sub> は、f<sub>LSE</sub>、f<sub>HSI</sub>、f<sub>PCLK</sub>、f<sub>SYS</sub> のいずれかです。

## 29.5.1 USART キャラクタの説明

ワード長は、USART\_CR1 レジスタの M[1:0] ビットをプログラムすることによって、7、8、または 9 ビットとして選択できます (図 324 を参照)。

- 7 ビットのキャラクタ長 : M[1:0] = 10
- 8 ビットのキャラクタ長 : M[1:0] = 00
- 9 ビットのキャラクタ長 : M[1:0] = 01

**注 :** 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ボーレート (0x7F および 0x55 フレーム検出) はサポートされません。7 ビットモードは、一部の USART でのみサポートされます。

デフォルトでは、信号 (TX または RX) はスタートビットの処理中ではロー状態です。また、ストップビットの処理中にはハイ状態です。

これらの値は、極性設定制御により、各信号について個別に反転できます。

**アイドルキャラクタ**は、すべてが「1」のフレームとして解釈されます (「1」の数にはストップビットの数が含まれます)。

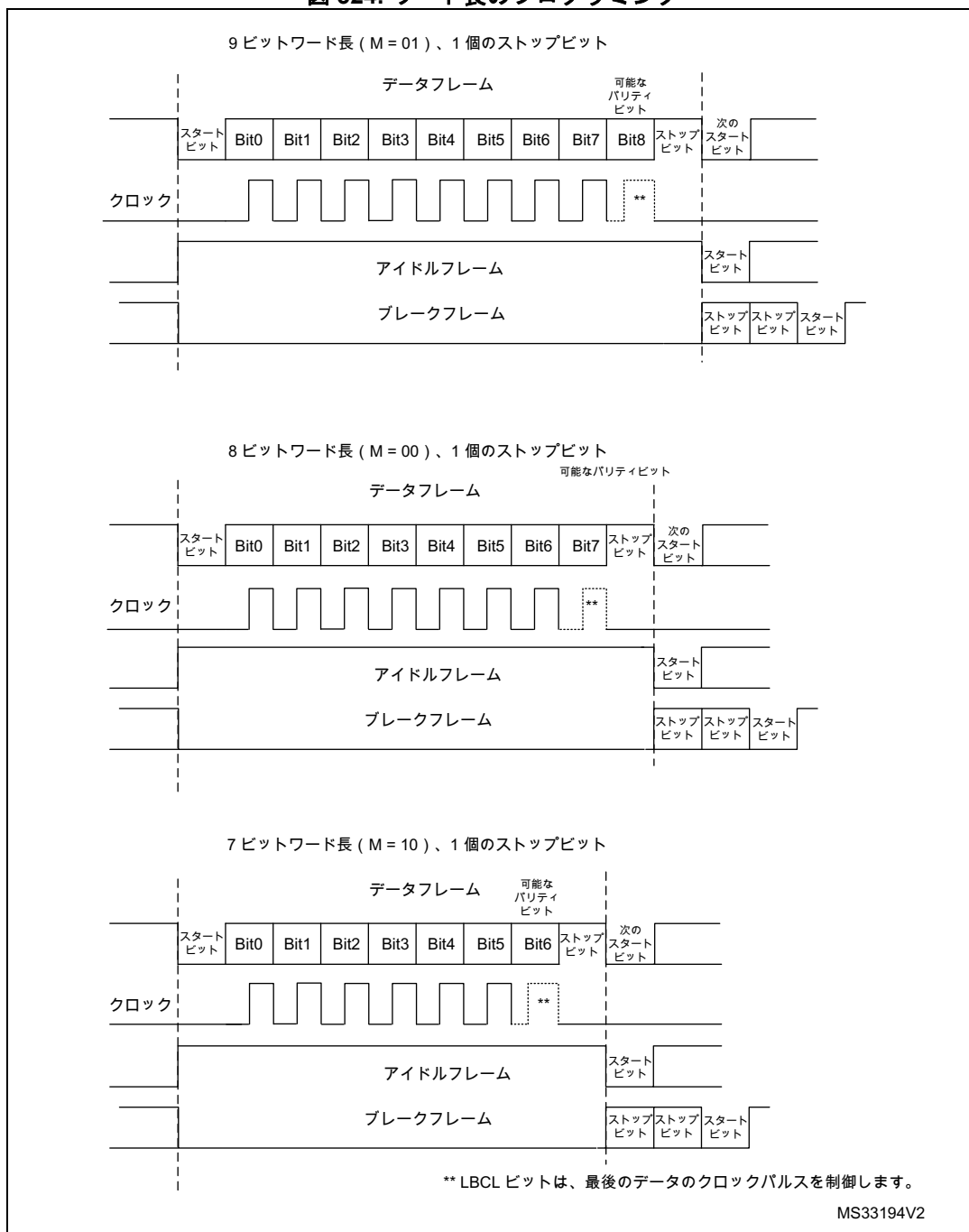
**ブレイクキャラクタ**は、フレーム周期中に「0」を受信することと解釈されます。ブレイクフレームの終了時、トランスミッタは 2 個のストップビットを挿入します。

送信と受信は、共通のボーレートジェネレータによって駆動され、それぞれのクロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。



図 324. ワード長のプログラミング



## 29.5.2 USART トランスミッタ

トランスミッタは、M ビットのステータスに応じて、7、8、または 9 ビットのデータワードを送信できます。トランスミッタ機能を有効にするには、送信イネーブルビット (TE) をセットする必要があります。送信シフトレジスタ内のデータは TX ピンで出力され、対応するクロックパルスは CK ピンで出力されます。

### キャラクタ送信

USART 送信時、データは LSB ファースト (デフォルト設定) で TX ピンにシフトアウトされます。このモードでは、USART\_TDR レジスタは、内部バスと送信シフトレジスタの間のバッファ (TDR) で構成されます (図 323 を参照)。

各キャラクタの前には、スタートビット (1 ビット周期、論理レベルロー) があります。キャラクタは、設定可能な数のストップビットで終端されます。

USART では、0.5 個、1 個、1.5 個および 2 個のストップビットがサポートされます。

**注:** **送信データを USART\_TDR に書き込む前に、TE ビットをセットする必要があります。**  
**データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、ポーレートカウンタが停止されるため、TX ピンのデータが破壊されます。送信中のデータは失われます。**

**TE ビットが有効になると、アイドルフレームが送信されます。**

### 設定可能なストップビット

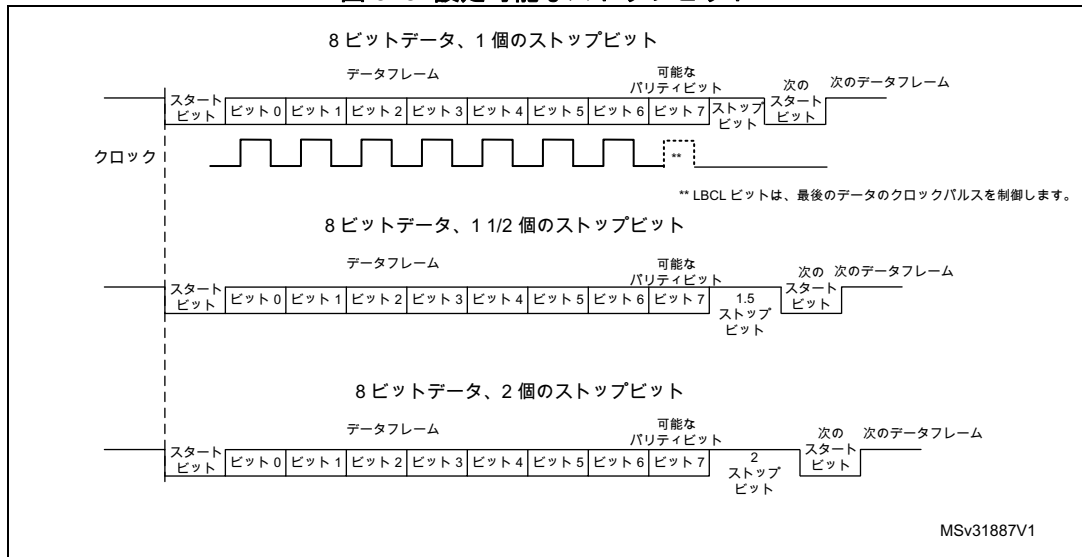
各キャラクタとともに送信されるストップビットの数は、制御レジスタ 2 のビット 13、12 でプログラミングできます。

- **1 個のストップビット:** ストップビット数のデフォルト値です。
- **2 個のストップビット:** 通常の USART モード、単線モード、およびモデムモードでサポートされます。
- **ストップビット 1, 5 個:** スマートカードモードで使用されます。
- **0.5 個のストップビット:** スマートカードモードでのデータの受信に使用されます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信は、10 個のロービット (M[1:0] = 00 のとき) または 11 個のロービット (M[1:0] = 01 のとき) または 9 個のロービット (M[1:0] = 10 のとき) の後に 2 個のストップビットが続きます (図 325 を参照)。長いブ레이크 (9/10/11 個のロービットを超える長さのブ레이크) を送信することはできません。

図 325. 設定可能なストップビット



## キャラクタ送信手順

1. ワード長を定義するには、USART\_CR1 の M ビットをプログラムします。
2. USART\_BRR レジスタを使用して、希望するボーレートを選択します。
3. USART\_CR2 レジスタでストップビットの数をプログラミングします。
4. USART\_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USART\_CR3 レジスタの DMA 有効 (DMAT) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
6. USART\_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。
7. 送信するデータを USART\_TDR レジスタに書き込みます (これによって TXE ビットがクリアされます)。シングルバッファの場合、送信される各データにこれを繰り返します。
8. USART\_TDR レジスタに最後のデータを書き込んだ後、TC=1 になるまで待ちます。これは、最後のフレームの送信が完了したことを示します。これは、たとえば最後の送信内容の破壊を避けるために、USART が無効になったり HALT モードに入ったりする場合に必要です。

## 1 バイト通信

TXE ビットは、常に送信データレジスタへの書き込みによってクリアされます。

TXE ビットは、ハードウェアによってセットされ、次を示します。

- データは USART\_TDR レジスタからシフトレジスタへ移動され、データ送信が開始しています。
- USART\_TDR レジスタは空です。
- 次のデータを、前のデータに上書きせずに、USART\_TDR レジスタに書き込みます。

TXEIE ビットがセットされている場合、このフラグは割り込みを生成します。

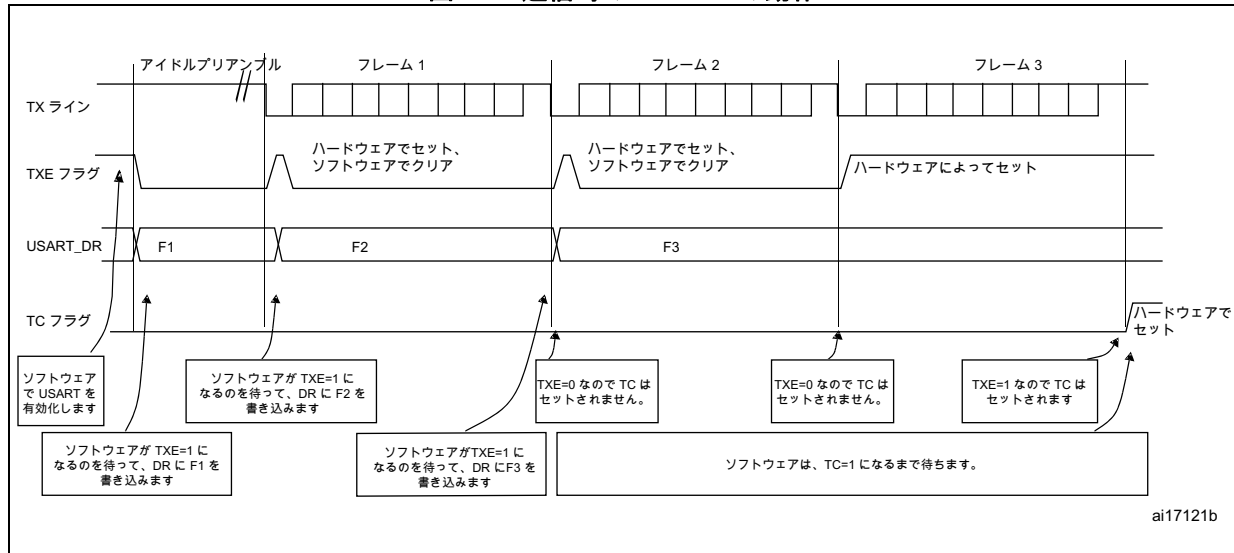
送信が行われている場合、USART\_TDR レジスタへの書き込み命令によってデータが TDR レジスタに格納され、次に、データが現在進行中の送信の最後にシフトレジスタにコピーされます。

送信が行われていないときには、USART\_TDR レジスタへの書き込み命令によってデータがシフトレジスタに格納され、データ送信が開始され、TXE ビットがセットされます。

フレームが送信され (ストップビットの後)、TXE ビットがセットされると、TC ビットはハイレベルになります。USART\_CR1 レジスタの TCIE ビットがセットされると、割り込みが生成されます。

USART\_TDR レジスタに最後のデータを書き込んだ後は、USART を無効にしたり、マイクロコントローラを低電力モードにする前に TC=1 になるまで待つ必要があります (図 326 : 送信時の TC/TXE の動作を参照)。

図 326. 送信時の TC/TXE の動作



## ブ레이크キャラクタ

SBKRQ ビットをセットすると、ブ레이크キャラクタが送信されます。ブ레이크フレーム長は、M ビットに依存します (図 324 を参照)。

SBKRQ ビットに 1 が書き込まれた場合、現在のキャラクタ送信の完了後、TX ラインにブ레이크キャラクタが送信されます。SBKF ビットは書き込み操作によってセットされ、ブ레이크キャラクタが完了すると (ブ레이크キャラクタの後のストップビット時に) ハードウェアによってリセットされます。USART は、次のフレームのスタートビットが正しく認識されるように、ブ레이크フレームの終わりに 2 ビットの長さの間、ロジック 1 信号 (STOP) を挿入します。

アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブ레이크キャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを待つ必要があります。

## アイドルキャラクタ

TE ビットをセットすると、USART は、最初のデータフレームの前にアイドルフレームを送信します。

## 29.5.3 USART レシーバ

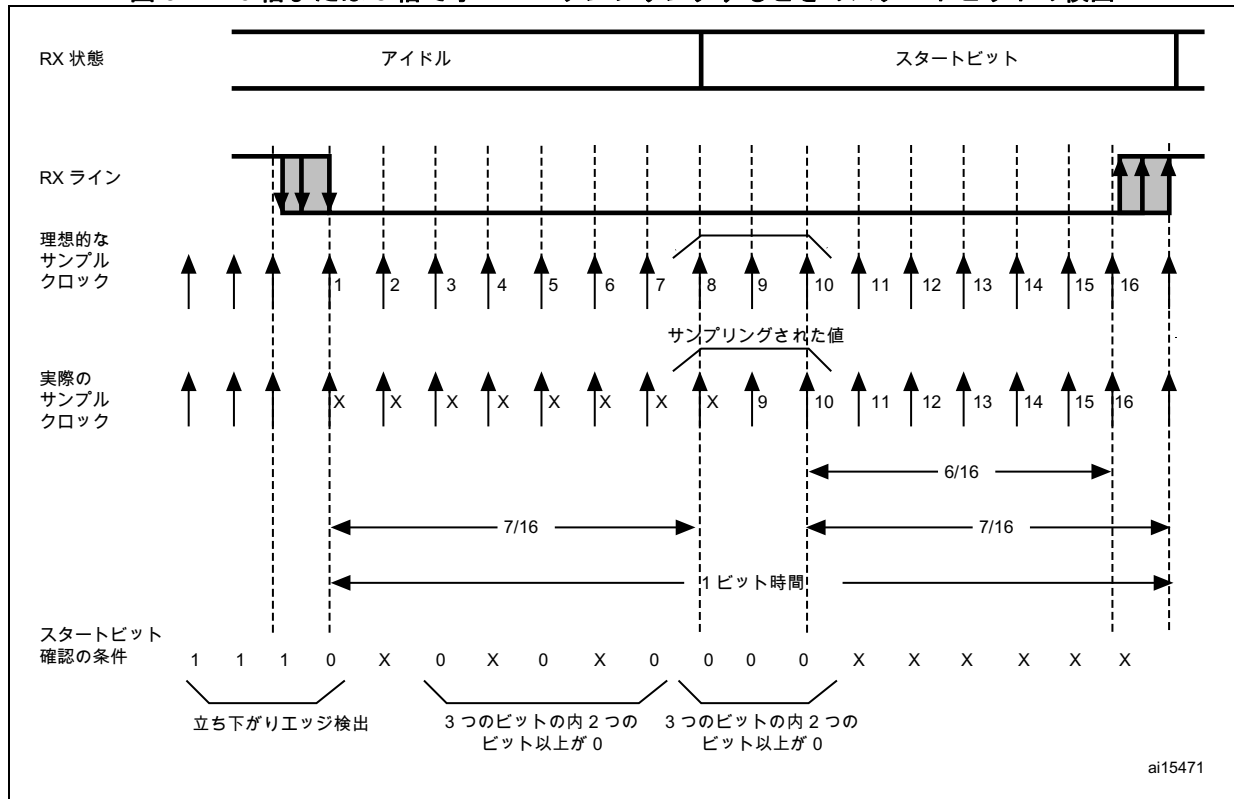
USART は、USART\_CR1 レジスタの M ビットに応じて、7、8 または 9 ビットのデータワードを受信できます。

### スタートビット検出

スタートビット検出シーケンスは、16 倍または 8 倍でオーバーサンプリングするときと同じです。

USART では、サンプルの特定シーケンスが認識されると、スタートビットが検出されます。このシーケンスは、1 1 1 0 X 0 X 0 X 0 X 0 X 0 です。

図 327. 16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出



注：シーケンスが完了していない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻って（フラグはセットされません）、立ち下がりエッジを待ちます。

サンプリングされた3つのビットが0の場合（第3、第5、および第7ビットでの最初のサンプリングで3つのビットが0であり、第8、第9、および第10ビットでの2回目のサンプリングでも3つのビットが0）、スタートビットが確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）。

次の場合、スタートビットは確認されますが（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）、NF ノイズフラグがセットされます。

- a) 両方のサンプリングについて、サンプリングされた3ビットのうちの2ビットが0の場合（第3、第5、および第7ビットのサンプリングと、第8、第9、および第10ビットのサンプリング）。

または

- b) いずれか一方のサンプリングで（第3、第5、および第7ビットでのサンプリングまたは第8、第9、および第10ビットでのサンプリング）、3つのビットのうち2つが0の場合。

a. または b. のいずれの条件も満たされない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻ります（フラグはセットされません）。

## キャラクタの受信

USART の受信時には、データは RX ピンを通じて LSB ファーストでシフトインされます。このモードでは、USART\_RDR レジスタは、内部バスと受信シフトレジスタの間のバッファ (RDR) で構成されます。

### キャラクタ受信手順

1. ワード長を定義するには、USART\_CR1 の M ビットをプログラムします。
2. ボーレートレジスタ USART\_BRR を使用して、希望するボーレートを選択します。
3. USART\_CR2 レジスタでストップビットの数をプログラミングします。
4. USART\_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USART\_CR3 レジスタの DMA 有効 (DMAR) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
6. USART\_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

キャラクタが受信されると、

- RXNE ビットは、シフトレジスタの内容が RDR レジスタに転送されたことを示すためにセットされます。言い換えると、データは受信され、読み出し可能です (関連するエラーフラグも同様です)。
- RXNEIE ビットがセットされていた場合、割り込みが生成されます。
- 受信中にフレームエラー、ノイズまたはオーバーランエラーが検出された場合、エラーフラグをセットできます。RXNE によって PE フラグもセットできます。
- マルチバッファでは、RXNE はバイト受信ごとにセットされ、受信データレジスタの DMA 読み出しによってクリアされます。
- シングルバッファモードでは、RXNE ビットのクリアは、ソフトウェアによる USART\_RDR レジスタの読み出しによって行われます。RXNE フラグは、USART\_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE ビットをクリアする必要があります。

## ブレイクキャラクタ

ブレイクキャラクタを受信すると、USART はブレイクキャラクタをフレーミングエラーとして処理します。

## アイドルキャラクタ

アイドルフレームが検出されると、受信データキャラクタの場合と同じ手順が行われ、IDLEIE ビットがセットされていた場合は割り込みが生成されます。

## オーバーランエラー

RXNE ビットがリセットされていないときにキャラクタを受信すると、オーバーランエラーが発生します。RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。

RXNE フラグは、バイトを受信するたびにセットされます。次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

- ORE ビットがセットされます。
- RDR の内容は失われません。USART\_RDR への読み出しが行われると、前回のデータが読み出されます。
- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXNEIE ビットがセットされているか、EIE ビットがセットされている場合、割り込みが生成されます。
- ORE ビットは、ICR レジスタの ORECF ビットをセットすることによってリセットされます。

**注：** ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。この場合、次の 2 つの可能性がります。

- RXNE=1 の場合、最後の有効なデータが受信レジスタ RDR に格納され、読み出すことができます。

- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR には読み出すべきものがないことを意味します。このケースは、有効な最後のデータが RDR で読み出されると同時に新しい（そして失われた）データが受信されると発生します。

## クロックソースと適切なオーバーサンプリング方式の選択

クロックソースの選択は、クロック制御システムを通じて行われます（リセットおよびクロック制御 (RCC) のセクションを参照）。クロックソースは、USART を有効にする（UE ビットをセットする）前に選ぶ必要があります。

クロックソースの選択は、2 つの基準に従って行われる必要があります。

- USART を低電力モードで使用できること
- 通信速度

クロックソース周波数は、 $f_{CK}$  です。

デュアルクロックドメインと STOP モードからのウェイクアップがサポートされるときには、クロックソースは次のソースの 1 つにできます。PCLK（デフォルト）、LSE、HSI、または SYSCLK。そうでない場合、USART のクロックソースは PCLK です。

LSE または HSI をクロックソースとして選択すると、USART は MCU が低電力モードのときにデータを受信できます。受信データとウェイクアップモードの選択に応じて、USART より必要なときに MCU をウェイクアップし、ソフトウェアが USART\_RDR レジスタを読み出すことによって、または DMA によって受信データを転送します。

他のクロックソースの場合、USART 通信を可能にするためには、システムを有効にする必要があります。

通信速度の範囲（特に最大通信速度）もクロックソースによって決まります。

レシーバは有効な受信データとノイズを区別して、データを復旧するユーザ設定可能なさまざまなオーバーサンプリング技術を実装しています。これにより、最大通信速度とノイズ/クロック精度の耐性の間でトレードオフができます。

オーバーサンプリング方式は、USART\_CR1 レジスタの OVER8 ビットをプログラムすることによって選択でき、ボーレートクロックの 16 倍または 8 倍にすることができます（[図 328](#) および [図 329](#)）。



アプリケーションに応じて、

- 高速 (最大  $f_{CK}/8$ ) を達成するには、8 倍 (OVER8=1) のオーバーサンプリングを選択します。この場合、クロック偏差に対するレシーバの最大許容誤差は軽減されます ([セクション 29.5.5: クロック偏差に対する USART レシーバの許容誤差 \(900 ページ\)](#) を参照)。
- クロック偏差に対するレシーバの許容誤差を増やすには、16 倍 (OVER8=0) のオーバーサンプリングを選択します。この場合、最大速度は  $f_{CK}/16$  に制限されます。 $f_{CK}$  はクロックソース周波数です。

論理レベルの評価方法を選択するには、USART\_CR3 レジスタの ONEBIT ビットをプログラミングします。これには 2 つのオプションがあります。

- 受信されたビットの中央にある 3 つのサンプルの多数決。この場合、多数決に使用された 3 つのサンプルが等しくないとき、NF ビットがセットされます。
- 受信されたビットの中央にある 1 つのサンプル。

アプリケーションに応じて、

- ノイズの多い環境で操作するときは 3 つのサンプルの多数決方式 (ONEBIT=0) を選択します。そしてノイズが検出された場合 ([図 157](#) を参照) は、サンプリング中にグリッチが発生していることとなりますので、そのデータを除去します。
- ラインがノイズフリーであるときは、1 つのサンプルによる方式 (ONEBIT=1) を選択し、クロック偏差に対するレシーバの許容誤差を増やします ([セクション 29.5.5: クロック偏差に対する USART レシーバの許容誤差 \(900 ページ\)](#) を参照)。この場合、NF ビットはセットされません。

フレーム内でノイズが検出された場合 :

- RXNE ビットの立ち上がりエッジで、NF ビットがセットされます。
- 無効なデータがシフトレジスタから USART\_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USART\_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

NF ビットは、ICR レジスタの NFCF ビットをセットすることによってリセットされます。

**注 :** 8 倍のオーバーサンプリングは、LIN、スマートカード、および IrDA の各モードでは使用できません。これらのモードでは、OVER8 ビットはハードウェアによって 0 に固定されています。

図 328. データサンプリング (16 倍のオーバーサンプリング)

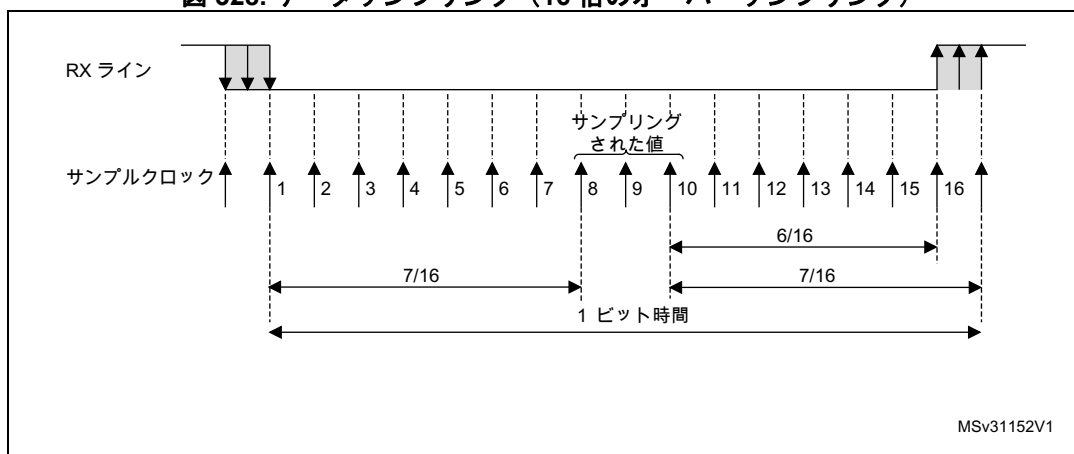
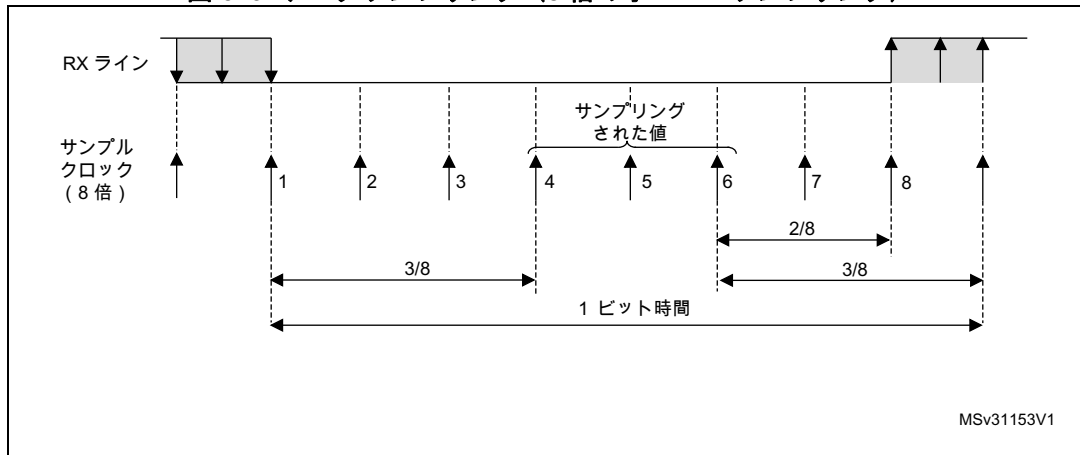




図 329. データサンプリング (8 倍のオーバーサンプリング)



MSv31153V1

表 157. サンプリングされたデータからのノイズ検出

サンプリングされた値	NE ステータス	受信ビットの値
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

## フレーミングエラー

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合、フレーミングエラーが検出されます。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから USART\_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USART\_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

USART\_ICR レジスタの FECF に 1 を書き込むことによって、FE ビットがリセットされます。

## 受信時の設定可能なストップビット

受信するストップビット数は、制御レジスタ 2 の制御ビットを通じて設定でき、通常モードでは 1 または 2、スマートカードモードでは 0.5 または 1.5 にできます。

- **0.5 個のストップビット (スマートカードモードでの受信)** : 0.5 個のストップビットでは、サンプリングは行われません。したがって、0.5 個のストップビットが選択されている場合、フレーミングエラーやブレイクフレームは検出されません。
- **1 個のストップビット** : ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
- **1.5 個のストップビット (スマートカードモード)** : スマートカードモードでの送信時は、デバイスは、データが正しく送信されたことをチェックする必要があります。したがって、レシーバブロックを有効にし (USART\_CR1 レジスタの RE=1)、ストップビットをチェックして、スマートカードがパリティエラーを検出したかどうかをテストする必要があります。パリティエラーが発生した場合、スマートカードはサンプリング時のデータ信号を強制的にローレベルにします (これは、フレーミングエラーとしてフラグされる NACK 信号です)。その後、1.5 個のストップビットの最後に、RXNE ビットとともに FE フラグがセットされます。ストップビット 1.5 個のサンプリングは、16 番目、17 番目、および 18 番目のサンプルで (ストップビットの開始から 1 ボークロック周期後に) 行われます。1.5 個のストップビットは、2 つの部分に分解できます。すなわち、何も起こらない 0.5 ボークロック周期と、途中でサンプリングが行われる通常の 1 ストップビット周期です。詳細については、[セクション 29.5.13 : USART スマートカードモード \(912 ページ\)](#) を参照してください。
- **2 個のストップビット** : ストップビット 2 個のサンプリングは、最初のストップビットの 8 番目、9 番目、および 10 番目のサンプルで行われます。最初のストップビットでフレーミングエラーが検出された場合、フレーミングエラーフラグがセットされます。2 番目のストップビットでは、フレーミングエラーの検査は行われません。RXNE フラグは、最初のストップビットの最後でセットされます。

## 29.5.4 USART ボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートは、USART\_BRR レジスタでプログラムされたものと同じ値に設定されます。

### 式 1 : 標準 USART のボーレート (SPI モードを含む) (OVER8 = 0 または 1)

16 倍のオーバーサンプリングの場合、等式は次のとおりです。

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{\text{USARTDIV}}$$

8 倍のオーバーサンプリングの場合、計算式は次のとおりです。

$$\text{Tx/Rx ボー} = \frac{2 \times f_{\text{CK}}}{\text{USARTDIV}}$$

### 式 2 : スマートカード、LIN、および IrDA モードのボーレート (OVER8 = 0)

スマートカード、LIN、および IrDA モード。16 倍のオーバーサンプリングのみをサポート :

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{\text{USARTDIV}}$$

USARTDIV は、符号なしの固定小数点数であり、USART\_BRR レジスタでコード化されます。

- OVER8 = 0 のとき、BRR = USARTDIV です。
- OVER8 = 1 のとき、
  - BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。
  - BRR[3] は、クリアされたままにする必要があります。
  - BRR[15:4] = USARTDIV[15:4]

**注：** ボーカウンタは、USART\_BRR への書き込み後、ポーレジスタの新しい値によって更新されます。したがって、通信中はポーレートレジスタの値を変更しないでください。

16 倍または 8 倍のオーバーサンプリングの場合、USARTDIV は 0d16 以上である必要があります。

## USART\_BRR レジスタの値から USARTDIV を得る方法

### 例 1

$f_{CK} = 8 \text{ MHz}$  で 9600 ボーを得るには、

- 16 倍のオーバーサンプリングの場合：  
USARTDIV =  $8\,000\,000/9600$   
BRR = USARTDIV = 833d = 0341h
- 8 倍のオーバーサンプリングの場合：  
USARTDIV =  $2 * 8\,000\,000/9600$   
USARTDIV = 1666,66 (1667d = 683h)  
BRR[3:0] = 3h >> 1 = 1h  
BRR = 0x681

### 例 2

$f_{CK} = 48 \text{ MHz}$  で 921.6 kbaud を得るには、

- 16 倍のオーバーサンプリングの場合：  
USARTDIV =  $48\,000\,000/921\,600$   
BRR = USARTDIV = 52d = 34h
- 8 倍のオーバーサンプリングの場合：  
USARTDIV =  $2 * 48\,000\,000/921\,600$   
USARTDIV = 104 (104d = 68h)  
BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h  
BRR = 0x64

表 158. 16 倍または 8 倍のオーバーサンプリングの両方の場合の  $f_{CK} = 72\text{MHz}$  でのプログラムされたボーレートの誤差計算<sup>(1)</sup>

ボーレート		16 倍のオーバーサンプリング (OVER8=0)			8 倍のオーバーサンプリング (OVER8=1)		
S.No	目標	実際	BRR	誤差 (%) = (計算値 - 目標値) ボーレート / 目標 ボーレート	実際	BRR	誤差
1	2.4 KBps	2.4 KBps	0x7530	0	2.4 KBps	0xEA60	0
2	9.6 KBps	9.6 KBps	0x1D4C	0	9.6 KBps	0x3A94	0
3	19.2 KBps	19.2 KBps	0xEA6	0	19.2 KBps	0x1D46	0
4	38.4 KBps	38.4 KBps	0x753	0	38.4 KBps	0xEA3	0
5	57.6 KBps	57.6 KBps	0x4E2	0	57.6 KBps	0x9C2	0
6	115.2 KBps	115.2 KBps	0x271	0	115.2 KBps	0x4E1	0
7	230.4 KBps	230.03KBps	0x139	0.16	230.4 KBps	0x270	0
8	460.8 KBps	461.54 KBps	0x9C	0.16	460.06KBps	0x134	0.16
9	921.6 KBps	923.08KBps	0x4E	0.16	923.07 KBps	0x96	0.16
10	2 MBps	2 MBps	0x24	0	2 MBps	0x44	0
11	3 MBps	3 MBps	0x18	0	3 MBps	0x30	0
12	4 MBps	4MBps	0x12	0	4 MBps	0x22	0
13	5 MBps	N/A	N/A	N/A	4965.51KBps	0x16	0.69
14	6 MBps	N/A	N/A	N/A	6 MBps	0x14	0
15	7MBps	N/A	N/A	N/A	6857.14KBps	0x12	2
16	9MBps	N/A	N/A	N/A	9MBps	0x10	0

1. CPU クロック周波数が低いほど、ボーレートの精度も低下します。ボーレートの達成可能な上限は、これらのデータによって決定できます。

## 29.5.5 クロック偏差に対する USART レシーバの許容誤差

USART の非同期レシーバは、クロックシステムの合計偏差が USART レシーバの許容誤差未満の場合のみ、正しく動作します。合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタのローカルオシレータの偏差も含みます)
- DQUANT : レシーバのボーレート量子化に起因する誤差
- DREC : レシーバのローカルオシレータの偏差
- DTCL : 送信ラインに起因する偏差 (一般には、ローからハイへの遷移タイミングとハイからローへの遷移タイミングの間に非対称性をもたらす可能性のあるトランシーバに起因)

$$DTRA + DQUANT + DREC + DTCL + DWU < \text{USART receiver's 許容誤差}$$

ここで、

DWU は、STOP モードからのウェイクアップが使用されたときのサンプリングポイントの偏差によるエラーです。

M[1:0] = 01 の場合 :

$$DWU = \frac{t_{WUUSART}}{11 \times Tbit}$$

M[1:0] = 00 の場合 :

$$DWU = \frac{t_{WUUSART}}{10 \times Tbit}$$

M[1:0] = 10 の場合 :

$$DWU = \frac{t_{WUUSART}}{9 \times Tbit}$$

$t_{WUUSART}$  は、ウェイクアップイベントと、クロック（ペリフェラルによるリクエスト）とレギュレータの動作可能状態を検出する間の時間です。

USART レシーバは、次の選択に応じて、表 159 および 表 160 で指定された最大許容偏差まで、データを正しく受信できます。

- USART\_CR1 レジスタの M ビットで定義された 9、10、または 11 ビットキャラクタ長
- USART\_CR1 レジスタの OVER8 ビットによって定義された 8 倍または 16 倍のオーバーサンプリング
- USART\_BRR レジスタのビット BRR[3:0] が 0000 に等しいかどうか。
- データのサンプリングに 1 ビットを使用するか 3 ビットを使用するか (USART\_CR3 レジスタの ONEBIT ビットの値に依存)。

表 159. BRR [3:0] = 0000 のときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 160. BRR[3:0] が 0000 でないときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注 : 表 159 および 表 160 で指定されたデータは、M ビット = 00 のとき、受信フレームに正確に 10 ビット持続時間のアイドルフレームが含まれる特殊なケースで、若干異なることがあります (M ビット = 01 のときには 11 ビット持続時間、または M ビット = 10 のときには 9 ビット持続時間)。



## 29.5.6 USART 自動ボーレート検出

USART は、1 キャラクタの受信に基づいて、USART\_BRR レジスタ値を検出し、自動的にセットすることができます。自動ボーレート検出は、2 つの状況で便利です。

- システムの通信速度が事前に分かっていないとき。
- システムが比較的低い精度のクロックソースを使用している場合、このメカニズムによって、クロック偏差を測定しなくても、正しいボーレートを求めることができます。

クロックソース周波数は、予期される通信速度と互換性がなければなりません（16 倍のオーバーサンプリングのとき、ボーレートは  $f_{CK}/65535$  から  $f_{CK}/16$  までの範囲内です。8 倍のオーバーサンプリングのとき、ボーレートは  $f_{CK}/65535$  から  $f_{CK}/8$  までの範囲内です）。

自動ボーレート検出を有効にする前に、自動ボーレート検出モードを選択する必要があります。キャラクタパターンに基づいて、いくつかのモードがあります。

これらのモードは、USART\_CR2 レジスタの ABRMOD[1:0] フィールドで選択できます。これらの自動ボーレートモードでは、同期データ受信中にボーレートが数回測定され、各測定値が前回の測定値と比較されます。

以下のモードがあります。

- **モード 0** : 1 のビットで始まるキャラクタ。この場合、USART はスタートビットの時間を測定します（立ち下がリエッジから立ち上がりエッジまで）。
- **モード 1** : 10xx ビットパターンで始まるキャラクタ。この場合、USART はスタートと最初のデータビットの時間を測定します。低速な信号傾斜の場合の精度を高めるために、測定は立ち下がリエッジから立ち上がりエッジまでで行われます。
- **モード 2** : 0x7F キャラクタフレーム (LSB ファーストモードでは 0x7F キャラクタ、または MSB ファーストモードでは 0xFE)。この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 6 の終了時に更新されます（立ち下がリエッジから立ち上がりエッジまで行われた測定に基づいて、BR6）。ビット 0 からビット 6 までが BRs でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。
- **モード 3** : 0x55 キャラクタフレーム。この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 0 の終了時に更新されます（立ち下がリエッジから立ち上がりエッジまで行われた測定に基づいて、BR0）、最後にビット 6 (BR6) の終了時に更新されます。ビット 0 は BRs でサンプリングされ、ビット 1 からビット 6 までが BR0 でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。

並行して、RX ラインの中間遷移ごとに別のチェックが行われます。RX の遷移がレシーバと十分に同期していない場合はエラーが生成されます（レシーバは、ビット 0 で計算されたボーレートに基づきます）。

自動ボーレート検出を有効にする前に、ゼロ以外のボーレート値を書き込むことによって、USART\_BRR レジスタを初期化する必要があります。

自動ボーレート検出を有効にするには、USART\_CR2 レジスタの ABREN ビットをセットします。USART は、RX ラインの最初のキャラクタを待ちます。自動ボーレート動作の完了は、USART\_ISR レジスタの ABRF フラグのセットによって示されます。ラインにノイズが多い場合、正しいボーレート検出を保証できません。この場合、BRR 値が破損して、ABRE エラーフラグがセットされることがあります。また、通信速度が自動ボーレート検出の範囲と互換性がない場合にも発生します（ビット時間が 16 から 65536 までのクロック周期でなく（16 倍のオーバーサンプリング）、8 から 65536 までのクロック周期でない（8 倍のオーバーサンプリング））。

RXNE 割り込みは、操作の終了を知らせます。

その後いつでも、ABRF フラグをリセットすることによって（0 を書き込むことによって）、自動ボーレート検出を再開できます。

注： 自動ポーレート操作中に USART が無効化された場合 (UE=0)、BRR 値が破損することがあります。

## 29.5.7 USART を使用したマルチプロセッサ通信

マルチプロセッサ通信では、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの LINEN ビット
- USART\_CR3 レジスタの HDSEL、IREN、および SCEN ビット

USART とのマルチプロセッサ通信が可能です (ネットワーク内で複数の USART を接続して)。たとえば、1 つの USART をマスタとして、その TX 出力を別の USART の RX 入力に接続することができます。別の USART はスレーブであり、それぞれの TX 出力の論理積をとった上でマスタの RX 入力に接続します。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する USART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードにできます。ミュートモード機能を使用するためには、USART\_CR1 レジスタの MME ビットをセットする必要があります。

ミュートモードでは、

- 受信ステータスビットはセットできません。
- 受信割り込みはすべて禁止されます。
- USART\_ISR レジスタの RWU ビットは 1 にセットされます。特定の条件下では、USART\_RQR レジスタの MMRQ ビットを通じて、RWU をハードウェアまたはソフトウェアによって自動的に制御できます。

USART は、USART\_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

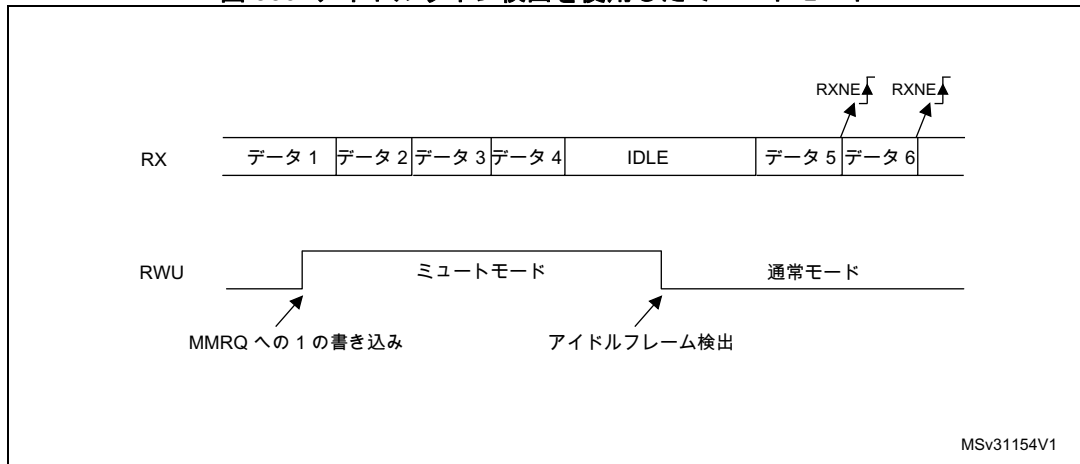
- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

## アイドルライン検出 (WAKE=0)

MMRQ ビットに 1 が書き込まれ、RWU が自動的にセットされたときには、USART はミュートモードに入ります。

USART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、USART\_ISR レジスタの IDLE ビットはセットされません。アイドルライン検出を使用したミュートモードの動作例を [図 330](#) に示します。

図 330. アイドルライン検出を使用したミュートモード



**注：** IDLE キャラクタがすでに経過しているときに MMRQ がセットされた場合は、ミュートモードに入りません (RWU はセットされません)。

ラインが IDLE のときに USART が有効にされた場合、1 IDLE フレーム後にアイドル状態が検出されます (1 キャラクタフレームの受信後だけでなく)。

## 4 ビット/7 ビットアドレスマーク検出 (WAKE=1)

このモードでは、MSB が 1 の場合、バイトはアドレスとして認識され、そうでない場合はデータとみなされます。アドレスバイトのうち、ターゲットレシーバのアドレスは 4 または 7 LSB です。7 または 4 ビットアドレス検出の選択は、ADDM7 ビットを使用して行われます。この 4 ビット/7 ビットワードは、レシーバによって、USART\_CR2 レジスタの ADD ビットでプログラムされたレシーバの自己アドレスと比較されます。

**注：** 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) で行われます。

プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、USART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。USART がミュートモードに入ったときには、このアドレスバイトに対して RXNE フラグはセットされず、割り込みも DMA リクエストも発行されません。

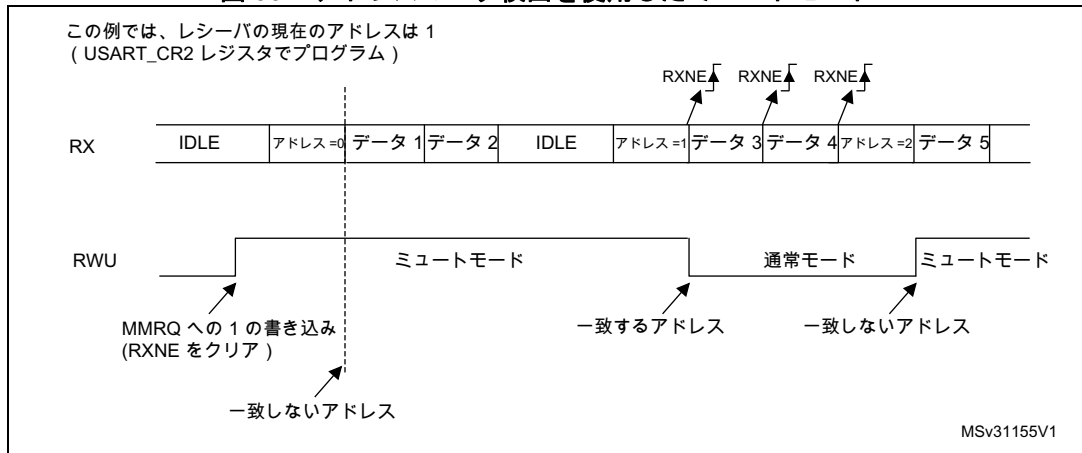
MMRQ ビットに 1 が書き込まれたときにも、USART はミュートモードに入ります。この場合、RWU ビットも自動的にセットされます。

プログラムされたアドレスに一致するアドレスキャラクタを受信すると、USART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE ビットがセットされません。

アドレスマーク検出を使用したミュートモードの動作例を [図 331](#) に示します。



図 331. アドレスマーク検出を使用したミュートモード



## 29.5.8 USART を使用した Modbus 通信

USART は、Modbus/RTU および Modbus/ASCII プロトコルの実装に対する基本的サポートを備えています。Modbus/RTU は、半二重のブロック転送プロトコルです。プロトコルの制御部分（アドレス認識、ブロック整合性制御、およびコマンド解釈）は、ソフトウェアで実装する必要があります。

USART は、ソフトウェアに負荷をかけず、他のリソースを使用せずに、ブロック検出の終了に対する基本的サポートを提供します。

### Modbus/RTU

このモードでは、1つのブロックの終了は2キャラクタ時間を超える「サイレンス」（アイドルライン）によって認識されます。この機能は、プログラム可能なタイムアウト機能を通じて実装されます。

タイムアウト機能と割り込みは、USART\_CR2 レジスタの RTOEN ビットと、USART\_CR1 レジスタの RTOIE を通じて有効にする必要があります。2キャラクタ時間のタイムアウトに対応する値（たとえば、22 x ビット時間）を RTO レジスタでプログラムする必要があります。最後のストップビットの受信後、この時間にわたって受信ラインがアイドルのときには、割り込みが生成されて、現在のブロック受信が完了したことをソフトウェアに知らせます。

### Modbus/ASCII

このモードでは、ブロックの終了は特定の (CR/LF) キャラクタシーケンスによって認識されます。USART は、キャラクター一致機能を使用して、このメカニズムを管理します。

ADD[7:0] フィールドで LF ASCII コードをプログラムし、キャラクター一致割り込みを有効にすることによって (CMIE=1)、LF が受信されたときにソフトウェアに通知し、ソフトウェアは DMA バッファの CR/LF をチェックできます。

## 29.5.9 USART パリティ制御

パリティ制御（送信中のパリティビット生成と受信中のパリティ検査）を有効にするには、USART\_CR1 レジスタの PCE ビットをセットします。M ビットによって定義されたフレーム長に応じて、可能な USART フレームフォーマットを表 161 に示します。

表 161. フレームフォーマット

M ビット	PCE ビット	USART フレーム <sup>(1)</sup>
00	0	SB 8 ビットデータ STB
00	1	SB 7 ビットデータ PB STB
01	0	SB 9 ビットデータ STB
01	1	SB 8 ビットデータ PB STB
10	0	SB 7 ビットデータ STB
10	1	SB 6 ビットデータ PB STB

1. 凡例：SB：スタートビット、STB：ストップビット、PB：パリティビット。データレジスタでは、PB は常に MSB 位置を取ります（M ビットの値に応じて、9、8、または 7 番目）。

### 偶数パリティ

パリティビットは、6、7、または 8 LSB ビット（M ビットの値に応じて）とパリティビットから構成されるフレーム内で「1」の数が偶数になるように計算されます。

たとえば、データ = 00110101 であり、4 ビットがセットされた場合、偶数パリティが選択された場合（USART\_CR1 の PS ビット = 0）、パリティビットは 0 になります。

### 奇数パリティ

パリティビットは、6、7、または 8 LSB ビット（M ビットの値に応じて）とパリティビットから構成されるフレーム内で「1」の数が奇数になるように計算されます。

たとえば、データ = 00110101 であり、4 ビットがセットされた場合、奇数パリティが選択された場合（USART\_CR1 の PS ビット = 1）、パリティビットは 1 になります。

### 受信中のパリティチェック

パリティチェックに失敗した場合、USART\_ISR レジスタの PE フラグがセットされ、USART\_CR1 レジスタの PEIE ビットがセットされている場合は割り込みが生成されます。PE フラグは、USART\_ICR レジスタの PECF に 1 を書き込むことによってクリアされます。

### 送信中のパリティ生成

USART\_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます（偶数パリティが選択された場合（PS=0）は偶数個の「1」、奇数パリティが選択された場合（PS=1）は奇数個の「1」）。

## 29.5.10 USART LIN (Local Interconnection Network) モード

このセクションは、LIN モードがサポートされるときにのみ適用されます。[セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。

LIN モードを選択するには、USART\_CR2 レジスタの LINEN ビットをセットします。LIN モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの STOP[1:0] および CLKEN ビット、
- USART\_CR3 レジスタの SCEN、HDSSEL、および IREN ビット。

### LIN 送信

LIN マスタ送信の場合、[セクション 29.5.2 : USART トランスミッタ](#) に説明されている手順を適用する必要があります。通常の USART 送信と同じですが、次のような違いがあります。

- 8 ビットのワード長を設定するには M ビットをクリアします。
- LIN モードに入るには、LINEN ビットをセットします。この場合、SBKRQ ビットをセットすると、13 個の“0” ビットがブレークキャラクタとして送信されます。その後、値“1”の 2 ビットが送信され、次の START 検出が可能になります。

### LIN 受信

LIN モードが有効になると、ブレーク検出回路が有効になります。この検出は、通常の USART レシーバとは完全に独立しています。アイドル状態時やフレームの処理中には、発生のたびにブレークが検出できます。

レシーバが有効になると (USART\_CR1 レジスタの RE=1)、RX 入力の START 信号を探します。スタートビットの検出方法は、ブレークキャラクタやデータの検索方法と同じです。スタートビットが検出された後、データの場合とまったく同様に次のビットがサンプリングされます (8、9、および 10 番目のサンプル)。10 個 (USART\_CR2 レジスタの LBDL=0) または 11 個 (USART\_CR2 レジスタの LBDL=1) の連続したビットが“0”として検出され、その後デリミタキャラクタが続く場合、USART\_ISR レジスタの LBDF フラグがセットされます。LBDIE ビットが 1 の場合、割り込みが生成されます。ブレークを確認する前に、RX ラインがハイレベルに戻ったことを知らせるデリミタが検査されます。

この 10 または 11 が発生する前に“1”がサンプリングされた場合、ブレーク検出回路は現在の検出をキャンセルし、再びスタートビットを検索します。

LIN モードが無効にされた場合 (LINEN=0)、レシーバは、ブレーク検出を考慮することなく、通常の USART として機能し続けます。

LIN モードが有効にされた場合 (LINEN=1)、フレーミングエラーが発生 (つまり、ブレークフレームの場合と同様に、“0”の位置でストップビットを検出) すると、レシーバは停止し、ブレーク検出回路が“1” (ブレークワードが完全でなかった場合) またはデリミタキャラクタ (ブレークが検出された場合) を受信するまで停止状態を維持します。

ブレーク検出回路ステートマシンの動作とブレークフラグを [図 332 : LIN モードでのブレーク検出 \(11 ビットブレーク長、LBDL=1\) \(908 ページ\)](#) に示します。

ブレークフレームの例を [図 333 : LIN モードでのブレーク検出とフレーミングエラー検出 \(909 ページ\)](#) に示します。

図 332. LIN モードでのブレイク検出 (11 ビットブレイク長、LBDL=1)

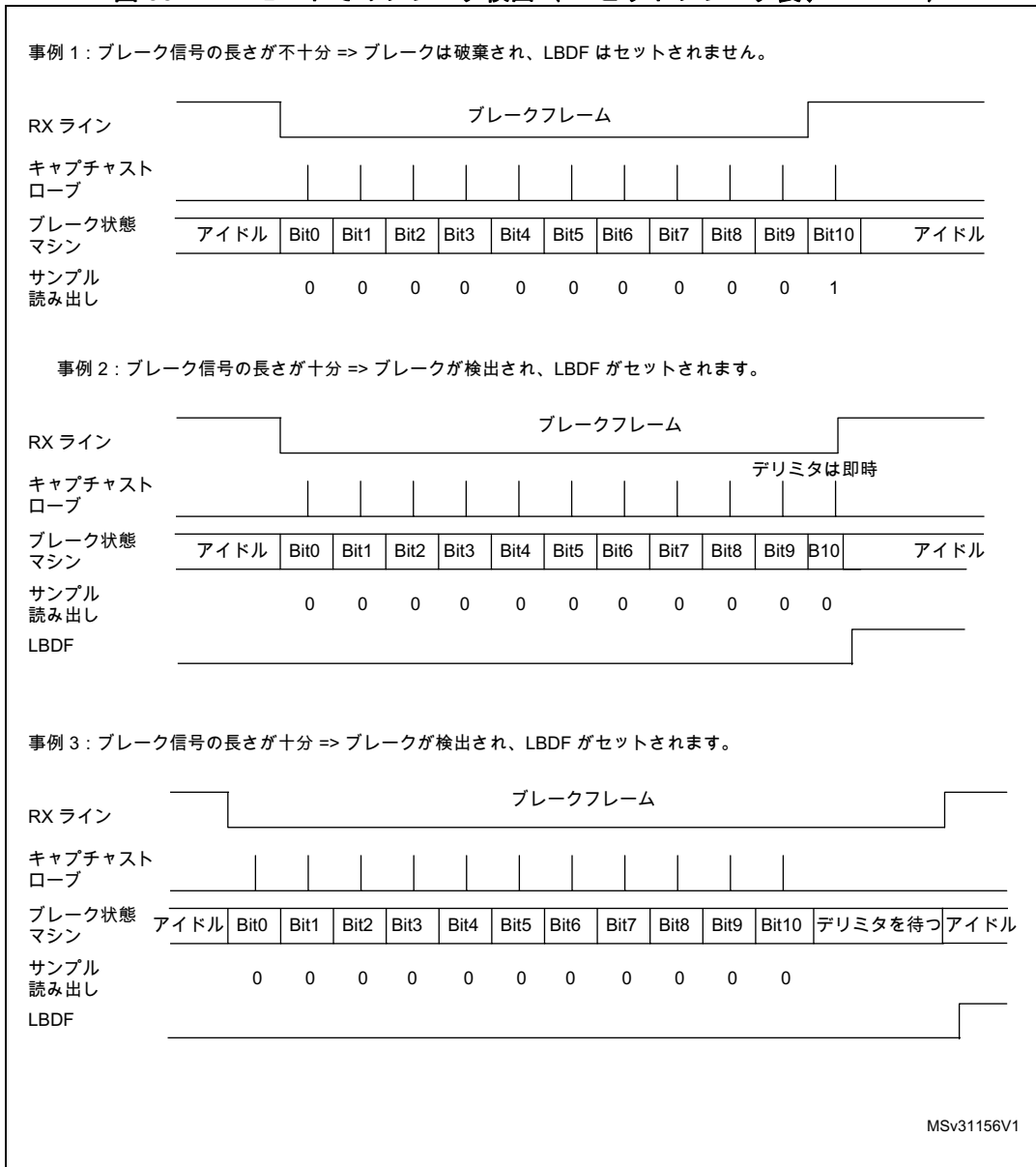
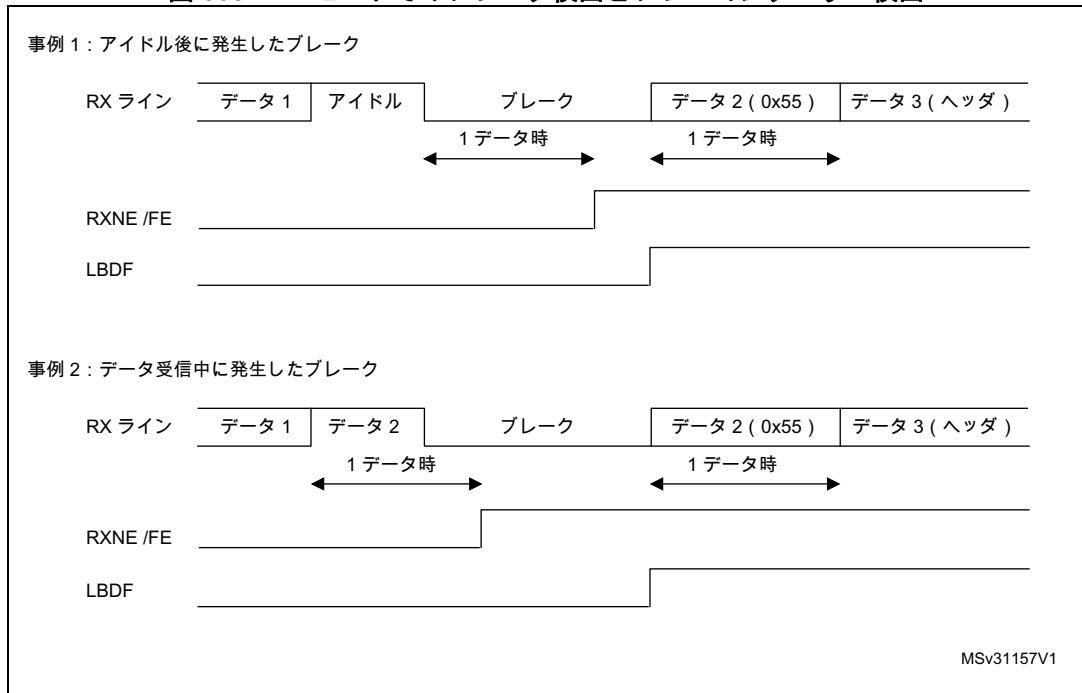


図 333. LIN モードでのブレーク検出とフレーミングエラー検出



## 29.5.11 USART 同期モード

同期モードを選択するには、USART\_CR2 レジスタの CLKEN ビットに 1 を書き込みます。同期モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの LINEN ビット
- USART\_CR3 レジスタの SCEN、HDSEL、および IREN ビット

このモードでは、USART を使用して、双方向同期シリアル通信をマスタモードで制御できます。CK ピンは USART トランスミッタクロックの出力です。スタートビットとストップビットの処理中には、CK ピンにクロックパルスは送信されません。USART\_CR2 レジスタの LBCL ビットの状態によっては、有効な最後のデータビット（アドレスマーク）の処理中にクロックパルスが生成されることもあります。USART\_CR2 レジスタの CPOL ビットは、クロック極性を選択するために使用され、USART\_CR2 レジスタの CPHA ビットは、外部クロックの位相を選択するために使用されます (図 334、図 335、および図 336 を参照)。

アイドル時、プリアンブル処理時、およびブレーク送信時には、外部 CK クロックは起動されません。

同期モードでは、USART トランスミッタは非同期モードの場合とまったく同じように機能します。しかし、CPOL と CPHA に基づいて CK が TX と同期すると、TX 上のデータが同期します。

このモードでは、USART レシーバは非同期モードの場合とは異なる動作をします。RE=1 の場合、データはオーバーサンプリングなしで、CK (CPOL と CPHA に応じて立ち上がりまたは立ち下がりエッジ) でサンプリングされます。ポーレート (1/16 ビット持続時間) に応じたセットアップ時間とホールド時間を守る必要があります。

**注:** CK ピンは TX ピンと連携して動作します。したがって、クロックが供給されるのは、トランスミッタが有効であり (TE=1)、データが送信中 (データレジスタ USART\_TDR への書き込み) の場合に限られます。つまり、データ送信を行わずに同期データを受信することはできません。

USART が無効にされたときには (UE=0)、クロックパルスが正常に機能するように、LBCL、CPOL、および CPHA ビットを選択する必要があります。



図 334. USART の同期送信の例

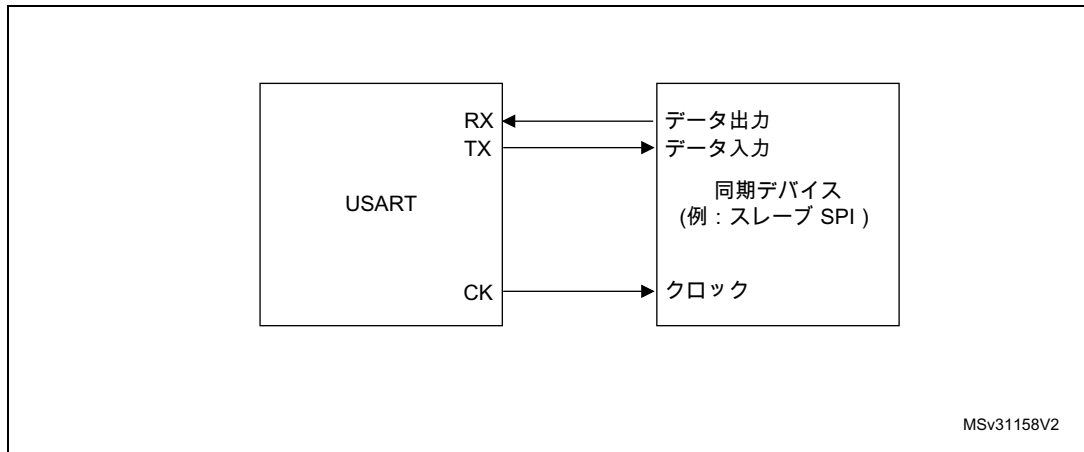


図 335. USART データクロックタイミング図 (M ビット = 00)

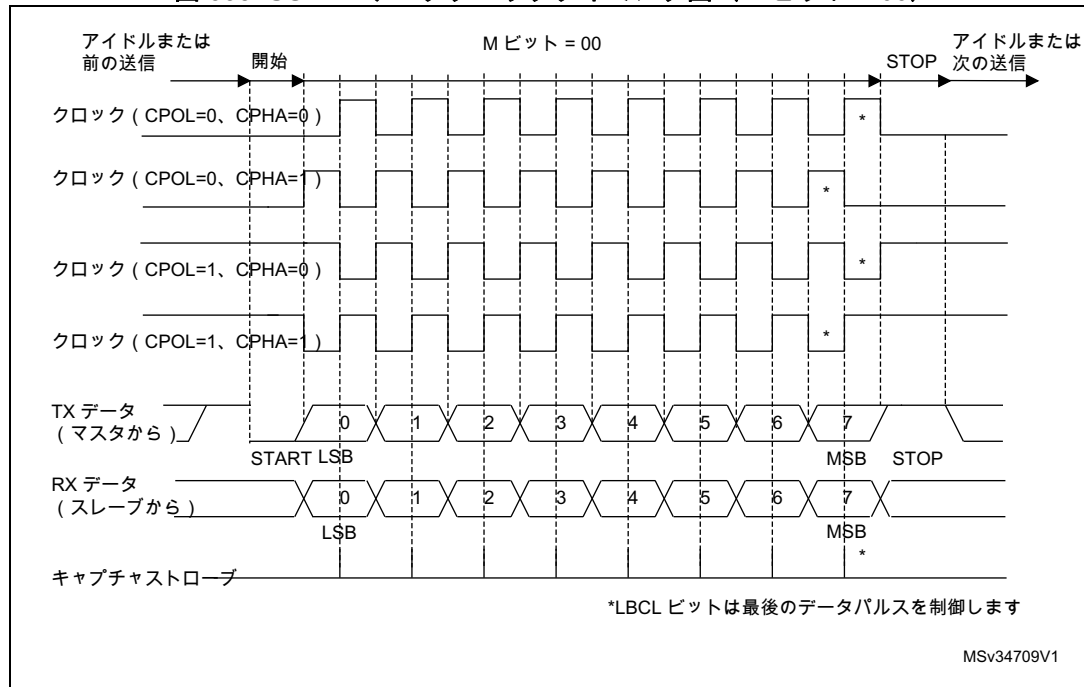


図 336. USART データクロックタイミング図 (M ビット = 01)

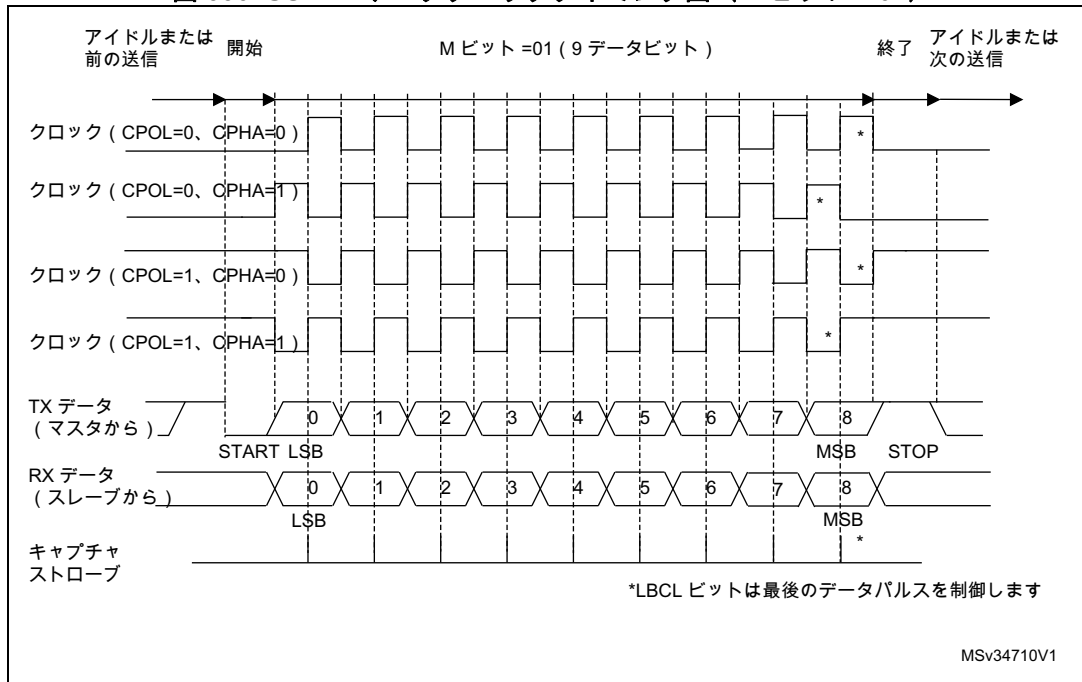
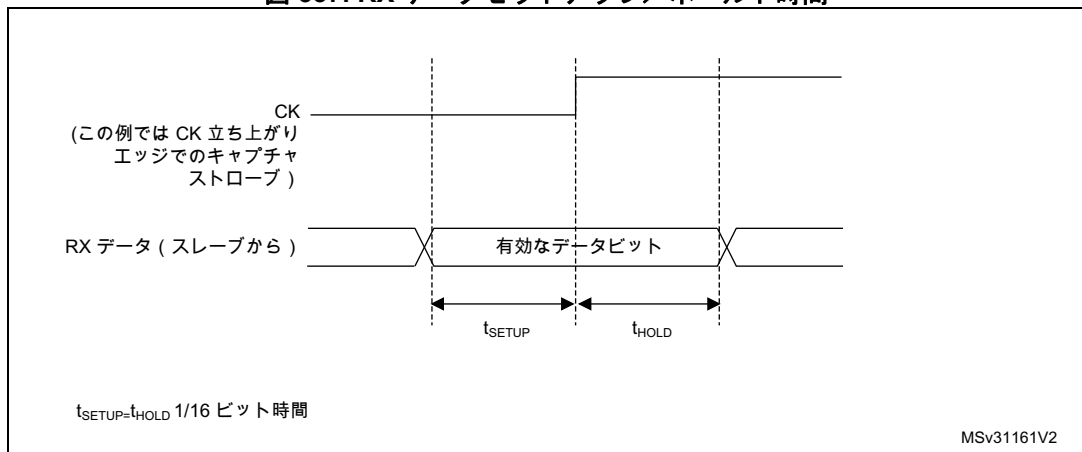


図 337. RX データセットアップ/ホールド時間



注： **スマートカードモードでは CK の機能が異なります。詳細については、[セクション 29.5.13 : USART スマートカードモード](#)を参照してください。**

## 29.5.12 USART 単線半二重通信

単線半二重モードを選択するには、USART\_CR3 レジスタの HDSEL ビットをセットします。このモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの LINEN および CLKEN ビット
- USART\_CR3 レジスタの SCEN および IREN ビット

USART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、USART\_CR3 レジスタの制御ビット HDSEL で行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が外部プルアップ付きの代替機能オープンドレインとして設定されるように、I/O を設定する必要があります。

この点を除くと、通信プロトコルは通常の USART モードと同じです。ラインの競合はソフトウェアによって管理する必要があります（たとえば、集中型アービタを使用して）。特に、送信がハードウェアによってブロックされることはなく、TE ビットがセットされている間は、データレジスタにデータが書き込まれるとすぐに、送信が続行されます。

## 29.5.13 USART スマートカードモード

このセクションは、スマートカードモードがサポートされるときにのみ適用されます。 [セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。

スマートカードモードを選択するには、USART\_CR3 レジスタの SCEN ビットをセットします。スマートカードモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの LINEN ビット
- USART\_CR3 レジスタの HDSEL および IREN ビット

さらに、スマートカードにクロックを供給するために CLKEN ビットをセットすることもできます。

スマートカードインタフェースは、ISO 7816-3 標準で定義されたスマートカードの非同期プロトコルをサポートするように設計されています。T=0（キャラクタモード）と T=1（ブロックモード）の両方がサポートされます。

USART は次のように設定してください。

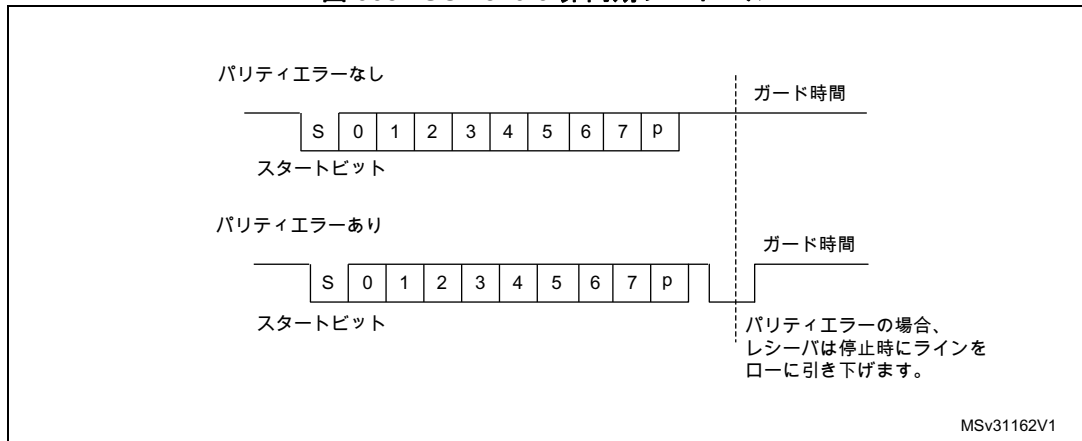
- 8 ビット+パリティ：ワード長が 8 ビットにセットされ、および USART\_CR1 レジスタの PCE=1。
- 1.5 個のストップビット：USART\_CR2 レジスタの STOP=11。受信にはストップビット 0.5 個を選択することも可能です。

T=0（キャラクタ）モードでは、パリティエラーはガードタイム中の各キャラクタの終わりに示されます。

[図 338](#) に、パリティエラーの有無によるデータラインの状況の変化の例を示します。



図 338. ISO 7816-3 非同期プロトコル



スマートカードと接続されると、USART の TX 出力は、スマートカードによっても駆動される双方ラインを駆動します。TX ピンは、オープンドレインとして設定される必要があります。

スマートカードモードは、単線半二重通信プロトコルを実装します。

- 送信シフトレジスタからのデータの送信は、少なくとも 1/2 ボークロックの遅れが保証されません。通常動作では、満杯の送信シフトレジスタは、次のボークロックエッジでシフト動作を開始します。スマートカードモードでは、この送信は、保証された 1/2 ボークロック分だけさらに遅れます。
- 送信時、スマートカードがパリティエラーを検出した場合には、ラインをローに駆動することによって (NACK)、この状態を USART に知らせます。この NACK 信号 (1 ボークロックの間、送信ラインをローに引き下げ) は、1.5 個のストップビットが組み込まれたトランスミッタ側にフレーミングエラーを引き起こします。USART は、プロトコルに従って、データの自動再送信を処理できます。再試行回数は、SCARCNT ビットフィールドでプログラムされます。プログラムされた再試行回数後も USART が NACK を受信し続けた場合は、送信を停止して、エラーをフレーミングエラーとして通知します。TXE ビットは、USART\_RQR レジスタの TXFRQ ビットを使用してセットできます。
- 送信時のスマートカード自動再試行: USART による NACK の検出と反復キャラクタのスタートビットの間に 2.5 ボー周期の遅延が挿入されます。最後の反復キャラクタの受信終了時、ただちに TC ビットがセットされます (ガードタイムはありません)。ソフトウェアで再び繰り返したい場合は、規格によって指定されている 2 ボー周期以上を確保する必要があります。
- 1.5 個のストップビット周期でプログラムされたフレームの受信時にパリティエラーが検出された場合、受信フレームの完了後 1 ボークロック周期間、送信ラインがローに引き下げられます。これは、USART に送信されたデータが正しく受信されなかったことをスマートカードに知らせるためです。NACK 制御ビットがセットされている場合、パリティエラーはレシーバによって "NACK" され、そうでない場合、NACK は送信されません (T=1 モードで使用されます)。受信したキャラクタにエラーがあった場合、RXNE/受信 DMA リクエストは有効になりません。プロトコルの仕様に従って、スマートカードは同じキャラクタを再送信する必要があります。SCARCNT ビットフィールドで指定された最大試行回数後も、受信したキャラクタにエラーがあった場合、USART は NACK の送信を停止して、エラーをパリティエラーとして通知します。
- 受信時のスマートカード自動再試行: USART がカードを NACK したが、カードがキャラクタを繰り返さなかった場合、BUSY フラグはセットされたままです。

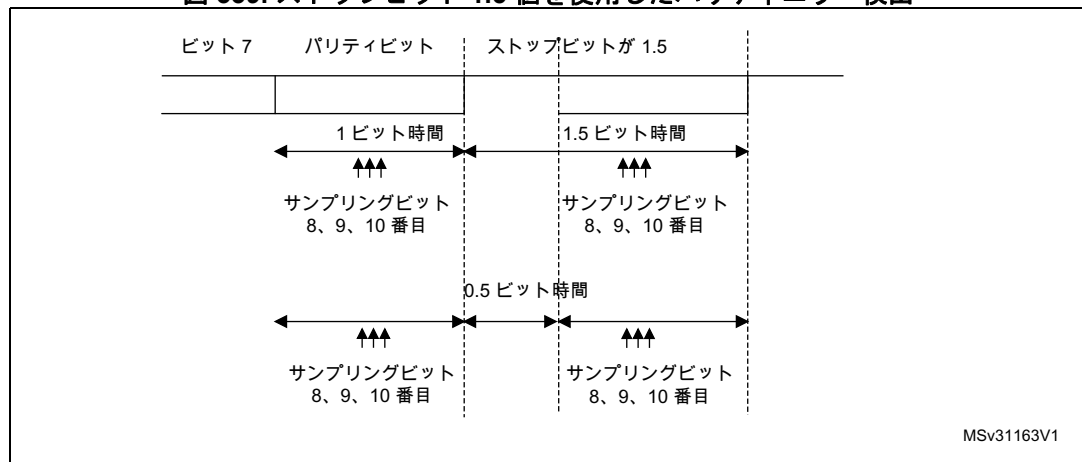
- 送信時、USART は 2 つの連続するキャラクタの間にガードタイム（ガードタイムレジスタでプログラム）を挿入します。ガードタイムは前のキャラクタのストップビット後に測定されるので、GT[7:0] レジスタを目的の CGT（7816-3 仕様で定義されている Character Guard Time）から 12（1 キャラクタの時間）を引いた値にプログラムする必要があります。
- TC フラグのアサーションは、ガードタイムレジスタをプログラムすることによって遅らせることができます。通常動作では、TC がアサートされるのは、送信シフトレジスタが空であり、他に未処理の送信リクエストがない場合です。スマートカードモードでは、空の送信シフトレジスタは、ガードタイムカウンタをトリガして、ガードタイムレジスタにプログラミングされた値までカウントアップします。この間、TC は強制的にローレベルに保たれます。ガードタイムカウンタがプログラミングされた値に達すると、TC がハイにアサートされます。
- TC フラグのネゲートは、スマートカードモードの影響を受けません。
- レシーバからの NACK によってトランスミッタ端でフレーミングエラーが検出された場合、トランスミッタの受信ブロックは、この NACK をスタートビットとして検出しません。ISO プロトコルによれば、受信される NACK の期間は 1 または 2 ボークロック周期です。
- レシーバ側では、パリティエラーが検出されて NACK が送信された場合、レシーバはこの NACK をスタートビットとして検出しません。

**注：** スマートカードモードでは、ブレイクキャラクタは意味を持ちません。フレーミングエラー発生時のデータ 0x00 は、ブレイクではなくデータとして処理されます。

TE ビットをトグルするとき、アイドルフレームは送信されません。アイドルフレームは、他の設定では定義されますが、ISO プロトコルでは定義されていません。

図 339 に、USART による NACK 信号のサンプリング方法を示します。この例では、USART はデータを送信中であり、ストップビットが 1.5 個組み込まれています。データと NACK 信号の整合性を検査するために、USART のレシーバ部が有効にされます。

図 339. ストップビット 1.5 個を使用したパリティエラー検出



USART は、CK 出力を通じてスマートカードにクロックを供給できます。スマートカードモードでは、CK は通信に関係せず、5 ビットのプリスケラを通じて単に内部のペリフェラル入力クロックから取得されます。この分周比は、プリスケラレジスタ USART\_ で設定されます。CK 周波数は、 $f_{CK}/2$  から  $f_{CK}/62$  までの範囲でプログラムでき、 $f_{CK}$  はペリフェラル入力クロックです。

## ブロックモード (T=1)

T=1 (ブロック) モードでは、パリティエラー送信は、USART\_CR3 レジスタの NACK ビットをクリアすることによって無効化されます。

ブロックモードでスマートカードからの読み出しをリクエストするときには、ソフトウェアは USART\_CR2 レジスタの RTOEN ビットをセットすることによってレシーバタイムアウト機能を有効にし、RTOR レジスタの RTO ビットフィールドを BWT (ブロックウェイトタイム) - 11 の値にプログラムする必要があります。この期間が終了する前にカードからの応答が受信されなかった場合、RTOF フラグがセットされ、タイムアウト割り込みが生成されます (USART\_CR1 レジスタの RTOIE ビットがセットされていた場合)。この期間が終了する前に最初のキャラクタが受信された場合は、RXNE 割り込みによって示されます。

**注:** *ブロックモードのスマートカードからの読み出しに USART を DMA モードで使用するときでも、RXNE 割り込みを有効にする必要があります。並行して、DMA は最初の受信バイトの後でのみ有効にする必要があります。*

2 つの連続するキャラクタの間で最大ウェイトタイムの自動チェックを行うには、最初のキャラクタの受信後 (RXNE 割り込み)、RTOR レジスタの RTO ビットフィールドを CWT (キャラクタウェイトタイム) - 11 の値にプログラムする必要があります。この時間は、ボータイム単位で表されます。スマートカードが前のキャラクタの終了後、CWT 未満の時間内に新しいキャラクタを送信しなかった場合、USART は RTOF フラグと割り込み (RTOIE ビットがセットされているとき) によって、これをソフトウェアに通知します。

**注:** *RTO カウンタはカウントを開始します。*

- STOP = 00 の場合、ストップビットの最後から
- STOP = 10 の場合、2 番目のストップビットの最後から
- STOP = 11 の場合、ストップビットの最初から 1 ビット時間後
- STOP = 01 の場合、ストップビットの最初から

*スマートカードプロトコルの定義にあるように、BWT/CWT 値は最後のキャラクタの開始 (スタートビット) から定義されます。RTO レジスタは、最後のキャラクタ自体の長さを考慮して、それぞれ BWT - 11 または CWT - 11 にプログラムする必要があります。*

ブロック長カウンタは、USART が受信するすべてのキャラクタをカウントするために使用されます。このカウンタは、USART の送信時 (TXE=0) にリセットされます。ブロックの長さは、スマートカードによってブロックの 3 番目のバイト (プロローグフィールド) で伝えられます。この値を USART\_RTOR レジスタの BLEN フィールドでプログラムする必要があります。DMA モードを使用するときには、ブロックの開始前に、このレジスタフィールドを最小値 (0x0) にプログラムする必要があります。この値では、4 番目の受信キャラクタの後に割り込みが生成されます。ソフトウェアは LEN フィールド (3 番目のバイト) を読み出す必要があり、その値は受信バッファから読み出される必要があります。

割り込み駆動受信モードでは、ブロックの長さはソフトウェアによって、または BLEN 値をプログラムすることによってチェックできます。ただし、ブロックの開始前に、BLEN の最大値 (0xFF) をプログラムすることができます。実際の値は、3 番目のキャラクタの受信後にプログラムされます。

ブロックが LRC 水平冗長検査 (1 エピローグバイト) を使用している場合は、BLEN=LEN です。ブロックが CRC メカニズム (2 エピローグバイト) を使用している場合は、BLEN=LEN+1 をプログラムする必要があります。合計ブロック長 (プロローグ、エピローグ、および情報フィールドを含む) は、BLEN+4 に等しくなります。ブロックの終わりは EOBFF フラグと割り込み (EOBIE ビットがセットされているとき) によってソフトウェアに通知されます。

ブロック長エラーの場合、ブロックの終わりは RTO 割り込みによって通知されます (キャラクタウェイトタイムオーバーフロー)。



注： エラーチェックコード (LRC/CRC) は、ソフトウェアによって計算/確認されなければなりません。

## ダイレクトおよびインバースコンベンション

スマートカードプロトコルは、ダイレクトとインバースの 2 つコンベンションを定義しています。

ダイレクトコンベンションは、LSB ファースト、論理ビットの値 1 がラインの H 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=0, DATAINV=0 (デフォルト値) をプログラムする必要があります。

インバースコンベンションは、MSB ファースト、論理ビットの値 1 が単線の L 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=1, DATAINV=1 をプログラムする必要があります。

注： 論理データ値が反転されると (0=H, 1=L)、パリティビットも同じように反転されます。

カードのコンベンションを認識するために、カードは最初のキャラクタ TS を ATR (Answer To Reset) フレームの最初のキャラクタとして送信します。TS には、LHHL LLL LLH と LHHL HHH LLH の 2 つのパターンが使えます。

- (H) LHHL LLL LLH は、インバースコンベンションをセットアップします。状態 L が値 1 にエンコードされ、モーメント 2 は最上位ビットを含みます (MSB ファースト)。インバースコンベンションによってデコードされると、送受信されたバイトは 3F に等しくなります。
- (H) LHHL HHH LLH は、ダイレクトコンベンションをセットアップします。状態 H が値 1 にエンコードされ、モーメント 2 は最下位ビットを含みます (LSB ファースト)。ダイレクトコンベンションによってデコードされると、送受信されたバイトは 3B に等しくなります。

キャラクタパリティは、2 から 10 までの 9 個のモーメントに 1 にセットされた偶数個のビットがあったときに正しいとみなされます。

USART はカードが使用するコンベンションを知らないので、いずれのパターンであるかを認識して、それに従って動作できる必要があります。パターン認識はハードウェアでは行われず、ソフトウェアシーケンスによって行われます。さらに、USART がダイレクトコンベンション (デフォルト) で設定され、カードがインバースコンベンションで応答した場合、TS = LHHL LLL LLH になり、USART が受信したキャラクタは 03 になり、パリティは奇数です。

したがって、TS パターン認識には 2 つの方式を使用できます。

### 方式 1

USART は、標準スマートカードモード/ダイレクトコンベンションでプログラムされます。この場合、TS パターンの受信によってパリティエラー割り込みと、カードに対するエラー信号が生成されます。

- パリティエラー割り込みは、カードがダイレクトコンベンションで正しく応答しなかったことをソフトウェアに知らせます。ソフトウェアは、USART をインバースコンベンションで再プログラムします。
- エラー信号に対して、カードは同じ TS キャラクタを再試行し、再プログラムされた USART によって今度は正しく受信されます。

または、パリティエラー割り込みに応答して、ソフトウェアは USART を再プログラムし、カードに対して新しいリセットコマンドを生成してから、TS を再び待つこともできます。

## 方式 2

USART は 9 ビット/パリティなしモード、ビット反転なしでプログラムされます。このモードでは、次のように 2 つの TS パターンのいずれかを受信します。

(H) LHHH LLL LLH = 0x103 -> インバースコンベンションを選択

(H) LHHH LLL LLH = 0x103 -> ダイレクトコンベンションを選択

ソフトウェアは受信されたキャラクタをこの 2 つのパターンと照合して、いずれかに一致した場合は、それに応じて、次のキャラクタ受信に備えて USART をプログラムします。

2 つのうちのどちらも認識されなかった場合、ネゴシエーションを再開するためにカードリセットが生成されます。

## 29.5.14 USART IrDA SIR ENDEC ブロック

このセクションは、IrDA モードがサポートされるときにのみ適用されます。[セクション 29.4:USART の実装 \(885 ページ\)](#) を参照してください。

IrDA モードを選択するには、USART\_CR3 レジスタの IREN ビットをセットします。IrDA モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART\_CR2 レジスタの LINEN、STOP、および CLKEN ビット
- USART\_CR3 レジスタの SCEN および HDSEL ビット

IrDA SIR 物理層は、ロジック 0 を赤外光パルスとして表現する RZI (Return to Zero, Inverted) 変調方式の使用を指定します ([図 340](#) を参照)。

SIR 送信エンコーダは、USART からの NRZ (Non Return to Zero) 送信ビットストリーム出力を変調します。出力パルスストリームは、外部の出力ドライバと赤外線 LED に送信されます。SIR ENDEC の場合、USART は最大 115.2 Kbps のビットレートしかサポートしません。通常モードでは、送信されるパルス幅は、ビット周期の 3/16 と指定されます。

SIR 受信デコーダは、赤外線検出回路からの RZ (Return to Zero) ビットストリームを復調し、受信した NRZ シリアルビットストリームを USART に出力します。デコーダの入力は、アイドル状態のノーマルハイレベル (マーク状態) です。送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。デコーダ入力がローレベルのとき、スタートビットが検出されます。

- IrDA は半二重通信プロトコルです。トランスミッタがビジーである場合 (USART が IrDA エンコーダにデータを送信しているとき)、IrDA デコーダは IrDA 受信ライン上にあるすべてのデータを無視します。レシーバがビジーである (USART が IrDA デコーダからデコードされたデータを受信している) 場合、IrDA は USART から IrDA への TX 上のデータをエンコードしません。データの受信中は、送信データの破壊を防ぐために、送信を避けてください。
- 0 はハイパルスとして送信され、1 は 0 として送信されます。通常モードでは、パルスの幅は、選択されたビット周期の 3/16 と規定されます ([図 341](#) を参照)。
- SIR デコーダは、IrDA 準拠の受信信号を USART 用のビットストリームに変換します。
- SIR 受信ロジックは、ハイ状態を論理値 1 とみなし、ローパルスを論理値 0 とみなします。
- 送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。SIR 出力は、アイドル時にロー状態になります。

- IrDA 仕様では、1.41 us より大きなパルスを受け入れる必要があります。受け入れられるパルス幅は、プログラム可能です。レシーバ側のグリッチ検出回路は、PSC 2 周期 (PSC は USART\_GTPR でプログラムされたプリスケール値) より小さな幅のパルスをフィルタします。PSC 1 周期より小さな幅のパルスは常に拒否されますが、1 周期以上2 周期未満の幅のパルスは受け入れられることも、拒否されることもあります。2 周期より大きな幅のパルスは、パルスとして受け入れられます。PSC=0 のとき、IrDA エンコーダ/デコーダは機能しません。
- レシーバは、低電力トランスミッタと通信できます。
- IrDA モードでは、USART\_CR2 レジスタのストップビットを「1 ストップビット」に設定する必要があります。

## IrDA 低電力モード

### トランスミッタ

低電力モードでは、パルス幅はビット周期の 3/16 に維持されません。代わりに、パルス幅は低電力ボーレート (最小で 1.42 MHz) の 3 倍となります。

一般に、この値は 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz) です。低電力モードのプログラム可能な分周器は、この値を得るためにシステムクロックを分周します。

### レシーバ

低電力モードでの受信は、通常モードでの受信と同様です。グリッチ検出の場合、USART は 1 PSC よりも短いパルスを破棄する必要があります。有効なローレベルは、その期間が IrDA 低電力ボーレート (USART\_GTPR の PSC 値) の 2 周期分を超える場合にのみ受け入れられます。

**注 :** PSC 2 周期未満 1 周期以上の幅のパルスは、拒否されることも、拒否されないこともあります。  
レシーバのセットアップ時間は、ソフトウェアで管理してください。IrDA 物理層仕様では、送信と受信の間に最小 10 ms の遅延を指定しています (IrDA は半二重プロトコルです)。

図 340. IrDA SIR ENDEC- ブロック図

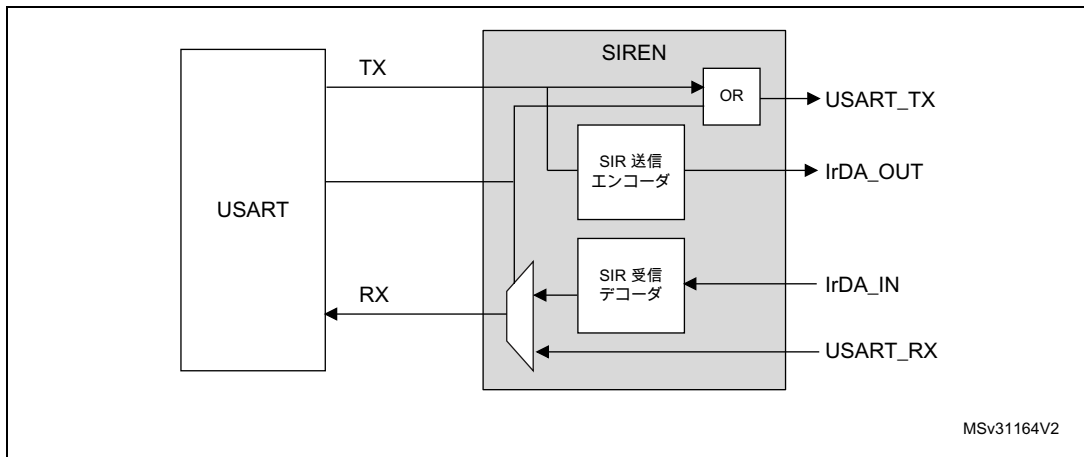
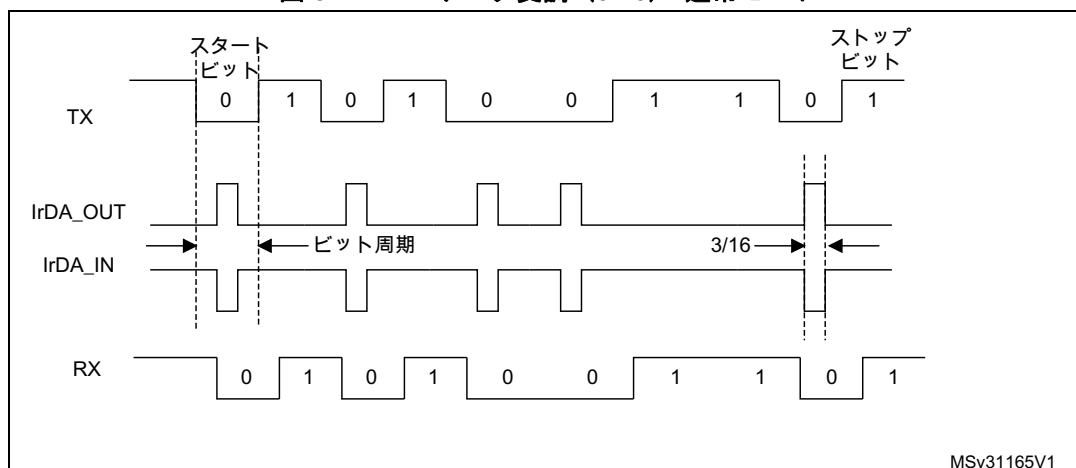


図 341. IrDA データ変調 (3/16) - 通常モード



## 29.5.15 DMA モードでの USART 連続通信

USART は、DMA を使用して連続通信を行うことができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

**注：** *DMA モードがサポートされるかどうかについては、[セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。DMA がサポートされない場合は、[セクション 29.5.2 : USART トランスミッタ](#) または [セクション 29.5.3 : USART レシーバ](#) での説明に従って USART を使用してください。連続通信を行うには、USART\_ISR レジスタの TXE/ RXNE フラグをクリアします。*

### DMA を使用した送信

DMA モードでの送信を有効にするには、USART\_CR3 レジスタの DMAT ビットをセットします。TXE ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 13 : ダイレクトメモリアクセスコントローラ \(DMA\) \(261 ページ\)](#)) を参照) を使用して設定された SRAM 領域から USART\_TDR レジスタにロードされます。DMA チャンネルを USART 送信用に割り付けるには、次の手順を実行します (x はチャンネル番号を示します)

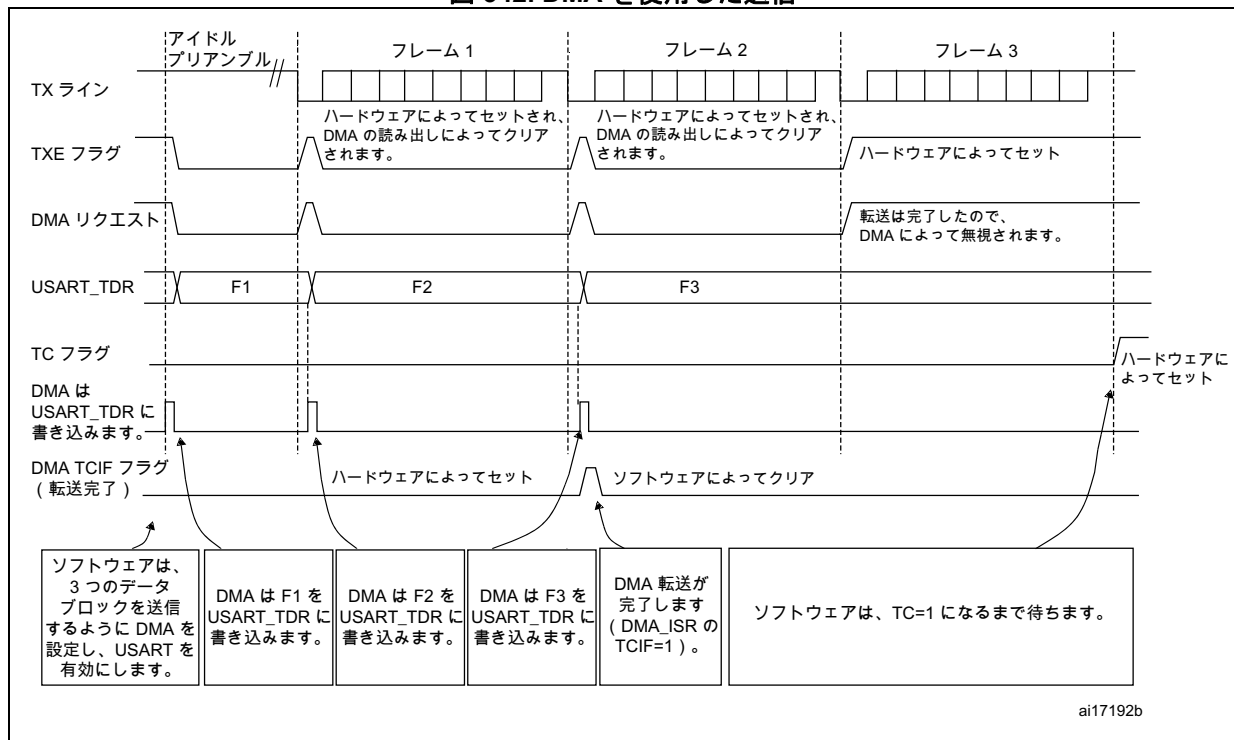
1. DMA 制御レジスタに USART\_TDR レジスタのアドレスを書き込み、これを転送先として設定します。データは、各 TXE イベント後にメモリからこのアドレスに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送元として設定します。データは、各 TXE イベント後に、このメモリ領域から USART\_TDR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. USART\_ICR レジスタの TCCF ビットをセットすることによって、USART\_ISR レジスタの TC フラグをクリアします。
7. DMA レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと (DMA\_ISR レジスタの TCIF フラグがセットされます)、TC フラグを観察して USART 通信の完了を確認することができます。これは、USART を無効にしたり STOP モードに入ったりする前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、すべてのデータ転

送中、クリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

図 342. DMA を使用した送信



## DMA を使用した受信

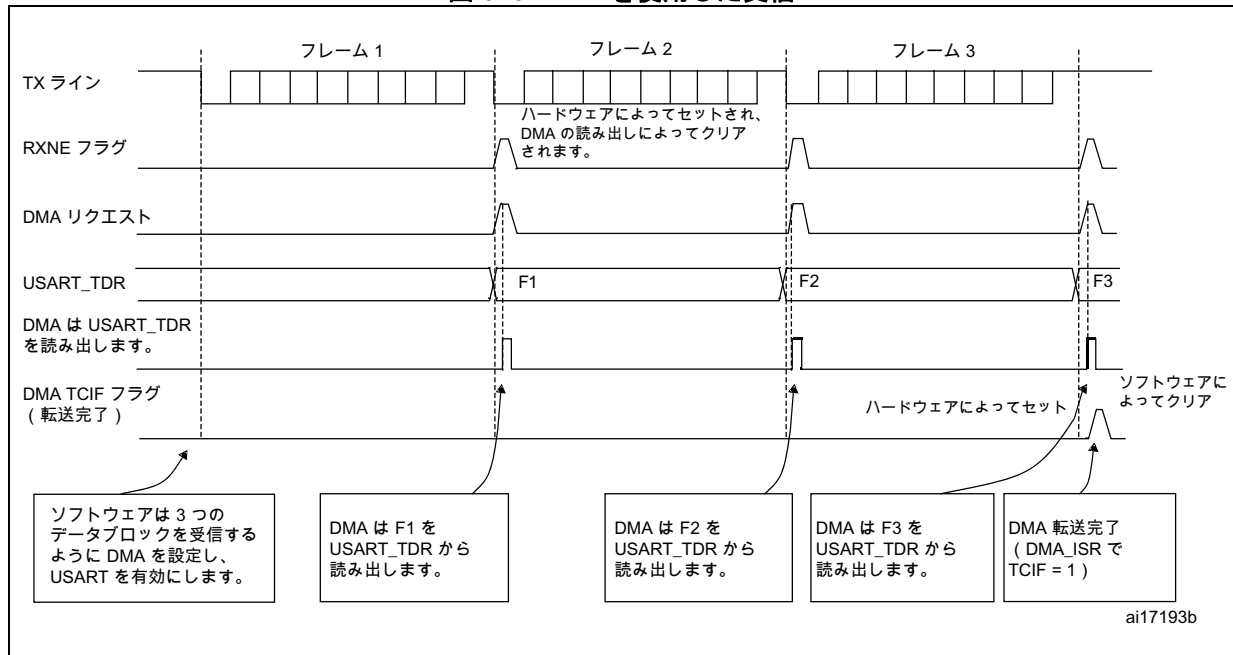
DMA モードでの受信を有効にするには、USART\_CR3 レジスタの DMAR ビットをセットします。データは、データバイトが受信されると、USART\_RDR レジスタから、DMA ペリフェラル ([セクション 13: ダイレクトメモリアクセスコントローラ \(DMA\) \(261 ページ\)](#)) を参照) を使用して設定された SRAM 領域にロードされます。DMA チャンネルを USART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに USART\_RDR レジスタのアドレスを書き込み、これを転送元として設定します。データは、各 RXNE イベント後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送先として設定します。データは、各 RXNE イベント後に、USART\_RDR からこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割り込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。



図 343. DMA を使用した受信



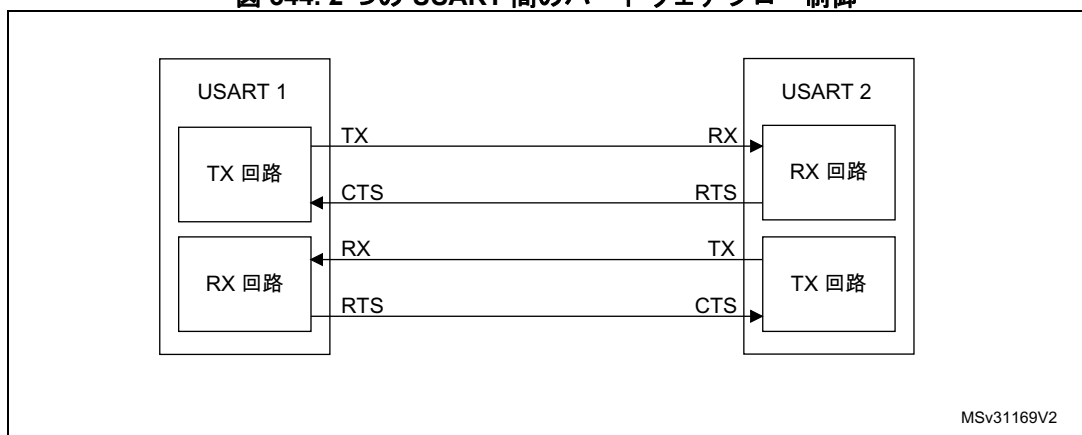
## マルチバッファ通信における割り込み生成とエラーフラグ

マルチバッファ通信でトランザクション中にエラーが発生した場合、現在のバイトの後にエラーフラグがアサートされます。割り込み有効フラグがセットされている場合、割り込みが生成されます。1バイト受信においてRXNEとともにアサートされるフレーミングエラー、オーバーランエラー、およびノイズフラグに関しては、別のエラーフラグ割り込み有効ビット (USART\_CR3 レジスタのEIEビット) があり、これがセットされている場合、いずれかのエラーが発生すると、現在のバイトの後に割り込みが有効になります。

## 29.5.16 USART を使用した RS232 ハードウェアフロー制御および RS485 ドライバ有効

CTS 入力と RTS 出力を使用すると、2つのデバイス間でシリアルデータフローを制御できます。図 344 に、このモードで2つのデバイスを接続する方法を示します。

図 344. 2つの USART 間のハードウェアフロー制御

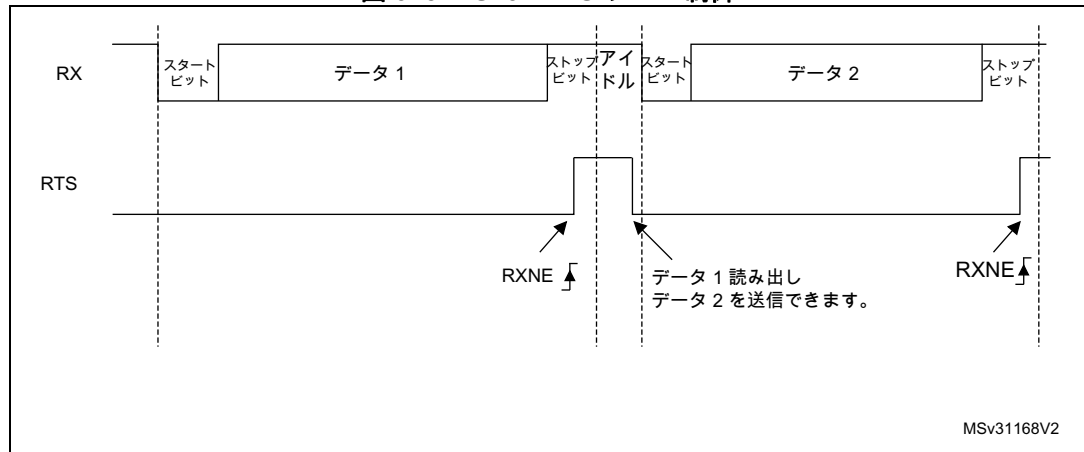


RS232 RTS と CTS のフロー制御は、USART\_CR3 レジスタの RTSE ビットと CTSE ビットにそれぞれ 1 を書き込むことによって、個別に有効にできます。

## RS232 RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、USART レシーバが新しいデータを受信可能である限り、RTS がアサートされます (ローレベル接続)。受信レジスタが満杯になると RTS がネゲートされ、現在のフレームの終わりに送信が停止する予定であることを示します。図 345 に、RTS フロー制御が有効な場合の通信例を示します。

図 345. RS232 RTS フロー制御

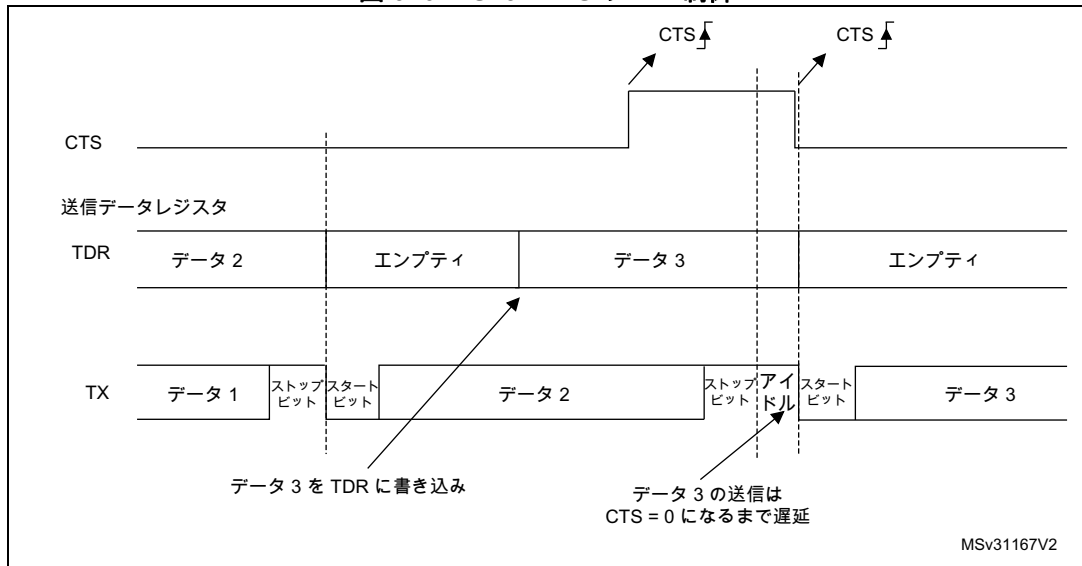


## RS232 CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、CTS 入力をチェックしてから、次のフレームを送信します。CTS がアサートされた場合 (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE=0 の場合)。そうでない場合、送信は行われません。送信中に CTS がネゲートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、CTS 入力が入ると、CTSIF ステータスビットはハードウェアによってただちに自動的にセットされます。このビットは、レシーバの通信準備ができているかどうかを示します。USART\_CR3 レジスタの CTSIE ビットがセットされている場合、割り込みが生成されます。図 346 に、CTS フロー制御が有効な場合の通信例を示します。

図 346. RS232 CTS フロー制御



注： **正しい動作のために、CTS は、現在のキャラクタの終了の少なくとも 3 USART クロックソース周期前にアサートする必要があります。さらに、2 x PCLK 周期より短いパルスでは CTSCF フラグがセットされない場合があることに注意してください。**

## RS485 ドライバ有効

ドライバ有効機能を有効にするには、USART\_CR3 制御レジスタのビット DEM をセットします。これにより、DE (Driver Enable) 信号によって外部トランシーバ制御を有効にできます。アサーション時間は、DE 信号の有効化から START ビットの開始までの時間です。USART\_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。ネゲート時間は、送信メッセージの最後のストップビットの終了から DE 信号の無効化までの時間です。USART\_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。DE 信号の極性は、USART\_CR3 制御レジスタの DEP ビットを使用して設定できます。

USART では、DEAT および DEDT はサンプル時間単位（オーバーサンプリングレートに応じて 1/8 または 1/16 ビット持続時間）で表されます。

## 29.5.17 USART を使用した STOP モードからのウェイクアップ

USART は、UESM ビットがセットされ、USART クロックが HSI または LSE にセットされているとき、MCU を STOP モードからウェイクアップできます（リセットおよびクロック制御 (RCC) のセクションを参照してください）。

STOP モードからの MCU のウェイクアップ は、標準の RXNE 割り込みを使用して行うことができます。この場合、STOP モードに入る前に RXNEIE ビットをセットする必要があります。

または、WUS ビットフィールドによって、特定の割り込みを選択することもできます。

MCU を STOP モードからウェイクアップできるようにするためには、STOP モードに入る前に、USART\_CR1 制御レジスタの UESM ビットをセットする必要があります。

ウェイクアップイベントが検出されると、ハードウェアによって WUF フラグがセットされ、WUFIE ビットがセットされていた場合はウェイクアップ割り込みが生成されます。

- 注：
- STOP モードに入る前に、ユーザは USART が転送を行っていないことを確認する必要があります。BUSY フラグでは、受信中に STOP モードに入らないことを保証できません。
  - WUF フラグは、MCU が STOP モードか、アクティブモードかに関係なく、ウェイクアップイベントが検出されたときにセットされます。
  - 初期化とレシーバの有効化の直後に STOP モードに入るときには、REACK ビットをチェックして、USART が実際に有効であることを確認する必要があります。
  - 受信に DMA が使用されるときには、STOP モードに入る前に無効化し、STOP モードの終了時に再び有効にする必要があります。
  - STOP モードからのウェイクアップは、すべてのモードで使用できるわけではありません。たとえば、SPI はマスタモードでのみ動作するので、SPI モードでは機能しません。

## STOP モードでのミュートモードの使用

USART は、STOP モードに入る前にミュートモードになります。

- アイドル検出は STOP モードでは機能しないので、アイドル検出時にミュートモードからウェイクアップすることはできません。
- アドレス一致によるミュートモードからのウェイクアップが使用される場合、STOP モードからのウェイクアップのソースもアドレス一致でなければなりません。STOP モードに入るときに RXNE フラグがセットされる場合、アドレス一致によって STOP からウェイクアップしても、インタフェースはミュートモードのままです。
- USART が START ビット検出時に MCU を STOP モードからウェイクアップするように設定された場合、WUF フラグはセットされますが、RXNE フラグはセットされません。

## USART クロックソースが HSI クロックであるときに最大 USART ボーレートを決定して STOP モードから正しくウェイクアップを許可

STOP モードから正しいウェイクアップを可能にする最大ボーレートは、次の要素の影響を受けません。

- デバイスのデータシートに示されたパラメータ  $t_{WUUSART}$
- [セクション 29.5.5 : クロック偏差に対する USART レシーバの許容誤差](#) に示された USART レシーバの許容誤差

この例では OVER8 = 0、M ビット = 10、ONEBIT = 1、BRR [3:0] = 0000 です。

この条件では、[表 159 : BRR \[3:0\] = 0000 のときの USART レシーバの許容誤差](#) によると、USART レシーバの許容誤差は 4.86 % です。

$DTRA + DQUANT + DREC + DTCL + DWU < \text{USART レシーバの許容誤差}$

$$DWU \max = t_{WUUSART} / (9 \times \text{Tbit Min})$$

$$\text{Tbit Min} = t_{WUUSART} / (9 \times DWU \max)$$

パラメータ DTRA、DQUANT、DREC および DTCL が 0% であるときに理想的なケースを考えた場合、DWU max は 4.86 % です。実際には、最低 HSI 精度を考慮する必要があります。

ここでは、HSI 精度 = 1%、 $t_{WUUSART} = 3.125 \mu\text{s}$  (RUN モードでメインレギュレータがあり、STOP モードからウェイクアップする場合) を考えます。

$$DWU \max = 4.86 \% - 1 \% = 3.86 \%$$

$$\text{Tbit Min} = 3.125 \mu\text{s} / (9 \times 3.86 \%) = 9 \mu\text{s}$$

この条件では、STOP モードから正しいウェイクアップを可能にする最大ボーレートは、 $1/9 \mu\text{s} = 111 \text{Kbaud}$  です。

## 29.6 USART 低電力モード

表 162. 低電力モードが USART に与える影響

モード	説明
SLEEP	影響はありません。USART 割り込みによって、デバイスは SLEEP モードを終了します。
STOP	USART は、UESM ビットがセットされ、USART クロックが HSI または LSE にセットされているとき、MCU を STOP モードからウェイクアップできます。STOP モードからの MCU のウェイクアップは、標準の RXNE または WUF 割り込みを使用して行うことができます。
STANDBY	USART はパワーダウンされ、デバイスが STANDBY モードを終了したときに再初期化する必要があります。

## 29.7 USART 割り込み

表 163. USART 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
送信データレジスタエンプティ	TXE	TXEIE
CTS 割り込み	CTSIF	CTSIE
送信完了	TC	TCIE
受信データレジスタノットエンプティ (データの読み出し可能)	RXNE	RXNEIE
オーバーランエラー検出	ORE	
アイドルライン検出	IDLE	IDLEIE
パリティエラー	PE	PEIE
LIN ブレーク	LBDF	LBDIE
マルチバッファ通信におけるノイズフラグ、オーバーランエラー、およびフレーミングエラー。	NF または ORE または FE	EIE
キャラクター一致	CMF	CMIE
レシーパタイムアウト	RTOF	RTOIE
ブロックの終了	EOBF	EOBIE
STOP モードからのウェイクアップ	WUF <sup>(1)</sup>	WUFIE

1. WUF 割り込みは、STOP モードでのみアクティブです。

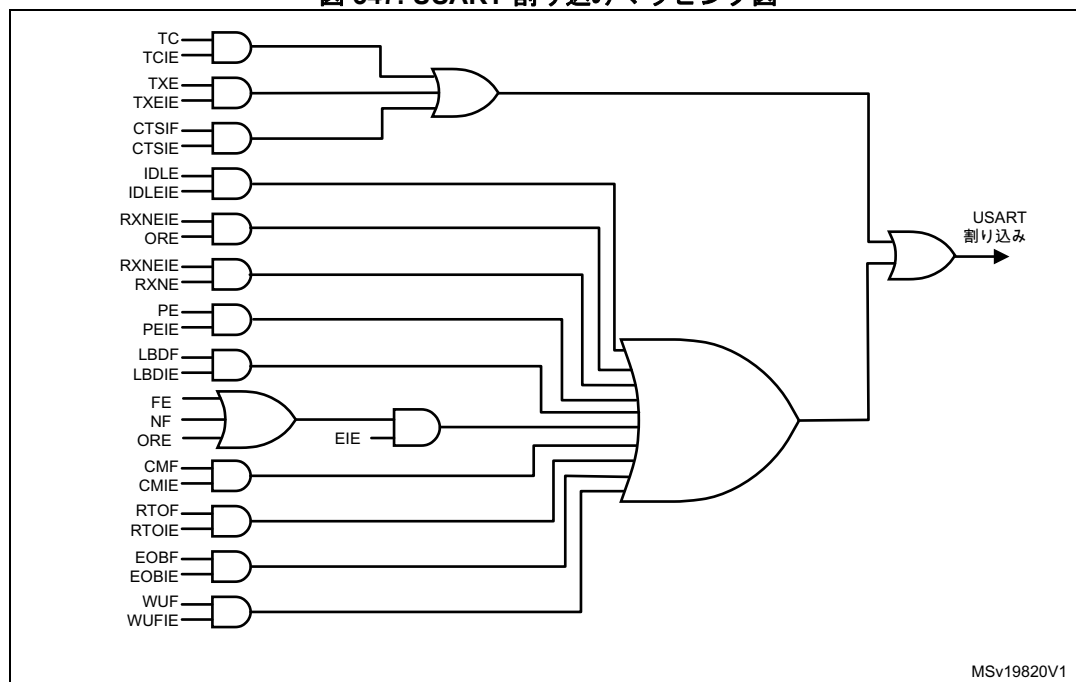
USART 割り込みイベントは、同じ割り込みベクターに接続されます (図 347 を参照)。

- 送信時: 送信完了、Clear to Send、送信データレジスタエンプティ、またはフレーミングエラー (スマートカードモード) 割り込み。
- 受信時: アイドルライン検出、オーバーランエラー、受信データレジスタノットエンプティ、パリティエラー、LIN ブレーク検出、ノイズフラグ、フレーミングエラー、キャラクター一致など。

これらのイベントは、対応する有効制御ビットがセットされている場合に割り込みを生成します。



図 347. USART 割り込みマッピング図



MSV19820V1

## 29.8 USART レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

### 29.8.1 制御レジスタ 1 (USART\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	M1	EOBIE	RTOIE	DEAT[4:0]				DEDT[4:0]					
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **M1** : ワード長

このビットとビット 12 (M0) によって、ワード長が決まります。ソフトウェアによってセット/クリアされます。

M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個

M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個

M[1:0] = 10: スタートビット 1 個、データビット 7 個、ストップビット n 個

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ポーレート (0x7F および 0x55 フレーム検出) はサポートされません。

ビット 27 **EOBIE** : ブロック終了割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの EOBIF フラグがセットされると、USART 割り込みが生成されます。

**注 :** USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。

ビット 26 **RTOIE** : レシーバタイムアウト割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの RTOIF フラグがセットされると、USART 割り込みが生成されます。

**注 :** USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 29.4 : USART の実装 \(885 ページ\)](#)。

ビット 25:21 **DEAT[4:0]** : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット持続時間) で表されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。[セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。



**ビット 20:16 DEDT[4:0]** : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット持続時間) で表されます。

DEDT 時間中に USART\_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 29.4 : USART の実装 (885 ページ) を参照してください。**

**ビット 15 OVER8** : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **LIN、IrDA、およびモードでは、このビットは常にクリア状態に保つ必要があります。**

**ビット 14 CMIE** : キャラクター致割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの CMF ビットがセットされると、USART 割り込みが生成されます。

**ビット 13 MME** : ミュートモード有効

このビットは、USART のミュートモード機能を有効にします。セットされると、USART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット/クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

**ビット 12 M0** : ワード長

このビットとビット 28 (M1) によって、ワード長が決まります。ソフトウェアによってセット/クリアされます。ビット 28 (M1) の説明を参照してください。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 11 WAKE** : レシーバウェイクアップ方式

このビットによって、ミュートモードからの USART のウェイクアップ方式が決まります。ソフトウェアによってセット/クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 10 PCE** : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合はビット 9、M=0 の場合はビット 8) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 9 PS** : パリティ選択

このビットは、パリティの生成/検出が有効である (PCE ビットがセットされている) とき、奇数パリティ/偶数パリティを選択します。ソフトウェアによってセット/クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。



**ビット 8 PEIE** : PE 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの PE=1 のときには、USART 割り込みが生成されます。

**ビット 7 TXEIE** : 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの TXE=1 のときには、USART 割り込みが生成されます。

**ビット 6 TCIE** : 転送完了割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの TC=1 のときには、USART 割り込みが生成されます。

**ビット 5 RXNEIE** : RXNE 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの ORE=1 または RXNE=1 のときには、USART 割り込みが生成されます。

**ビット 4 IDLEIE** : IDLE 割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの IDLE=1 のときには、USART 割り込みが生成されます。

**ビット 3 TE** : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット/クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

**注 :** 送信時、TE ビットに 0 パルス (0 の後に 1) を与えると、現在のワードの後にプリアンブル (アイドルライン) を送信します (スマートカードモードの場合を除きます)。アイドルキャラクタを生成するためには、すぐには TE に 1 を書き込まないでください。必要な時間を確保するために、ソフトウェアは USART\_ISR レジスタの TEACK ビットをポーリングできます。

スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

**ビット 2 RE** : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット/クリアされます。  
 0 : レシーバは無効です。  
 1 : レシーバは有効であり、スタートビットの検索が開始されます。

**ビット 1 UESM** : STOP モードでの USART 有効

このビットがクリアされると、USART は MCU を STOP モードからウェイクアップできません。  
 このビットがセットされると、USART は MCU を STOP モードからウェイクアップできますが、USART クロック選択が RCC にて HSI または LSE であることが条件です。  
 このビットは、ソフトウェアによってセット/クリアされます。  
 0 : USART は STOP モードから MCU をウェイクアップできません。  
 1 : USART は STOP モードから MCU をウェイクアップできます。この機能がアクティブなとき、USART のクロックソースは HSI または LSE でなければなりません (リセットおよびクロック制御 (RCC) を参照)。

**注 :** STOP モードに入る直前に UESM ビットをセットし、STOP モードの終了時にクリアすることが推奨されます。

USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 0 UE** : USART 有効

このビットがクリアされると、USART プリスケアラと出力はただちに停止され、現在の操作は破棄されます。USART の設定は保たれますが、USART\_ISR のステータスフラグはすべてそれぞれのデフォルト値にセットされます。このビットは、ソフトウェアによってセット/クリアされます。  
 0 : USART プリスケアラと出力は無効であり、低電力モードです。  
 1 : USART は有効です。

**注 :** ラインにエラーを生成せずに低電力モードに入るためには、TE ビットをリセットする必要があり、ソフトウェアは USART\_ISR の TC ビットがセットされるのを待ってから、UE ビットをリセットする必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャンネルを無効にする必要があります。

## 29.8.2 制御レジスタ 2 (USART\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD[1:0]		ABREN	MSBFI RST	DATAINV	TXINV	RXINV
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	Res.	Res.	Res.	Res.
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w				

**ビット 31:28 ADD[7:4] : USART ノードのアドレス**

このビットフィールドは、認識される USART ノードのアドレスまたはキャラクタコードを指定します。これは、マルチプロセッサ通信において、7 ビットアドレスマーク検出によるウェイクアップのために、ミュートモードまたは STOP モード時に使用されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。通常の受信時、ミュートモードが無効なときに（たとえば、Modbus プロトコルのブロック終了検出）、キャラクタ検出のために使用することができます。この場合、受信されたキャラクタ全体 (8 ビット) が ADD[7:0] 値と比較され、一致した場合は CMF フラグがセットされます。このビットフィールドは、受信が無効のとき (RE=0) または USART が無効のとき (UE=0) のみ、書き込むことができます。

**ビット 27:24 ADD[3:0] : USART ノードのアドレス**

このビットフィールドは、認識される USART ノードのアドレスまたはキャラクタコードを指定します。これは、マルチプロセッサ通信において、アドレスマーク検出によるウェイクアップのために、ミュートモードまたは STOP モード時に使用されます。このビットフィールドは、受信が無効のとき (RE=0) または USART が無効のとき (UE=0) のみ、書き込むことができます。

**ビット 23 RTOEN : レシーバタイムアウト有効**

このビットは、ソフトウェアによってセット/クリアされます。

0 : レシーバタイムアウト機能は無効です。

1 : レシーバタイムアウト機能は有効です。

この機能が有効なとき、RTOR (レシーバタイムアウトレジスタ) でプログラムされた時間にわたって RX ラインがアイドル (受信なし) であった場合、USART\_ISR レジスタの RTOF フラグがセットされます。

**注 :** USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 22:21 ABRMOD[1:0] : 自動ポーレートモード**

これらのビットは、ソフトウェアによってセット/クリアされます。

00 : スタートビットの測定がポーレートの検出に使用されます。

01 : 立ち下がりがエッジから立ち下がりがエッジまでの測定。(受信されたフレームはシングルビット = 1 で始まらなければならない、その場合、フレーム = Start10xxxxxx)

10 : 0x7F フレーム検出

11 : 0x55 フレーム検出

このビットフィールドは、ABREN=0 または USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** DATAINV=1 および/または MSBFIRST=1 の場合、パターンはライン上で同じである必要があります(たとえば、MSBFIRST の場合は 0xAA)。

USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 20 ABREN : 自動ポーレート有効**

このビットは、ソフトウェアによってセット/クリアされます。

0 : 自動ポーレート検出は無効です。

1 : 自動ポーレート検出は有効です。

**注 :** USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 19 MSBFIRST : MSBファースト**

このビットは、ソフトウェアによってセット/クリアされます。

0 : スタートビットに続いて、データはデータビット 0 から順に送受信されます。

1 : スタートビットに続いて、データは MSB (ビット 7/8/9) から順に送受信されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 18 DATAINV** : バイナリデータ反転

このビットは、ソフトウェアによってセット/クリアされます。

0 : データレジスタからの論理データは正/ダイレクトロジックで送受信されます。(1=H、0=L)

1 : データレジスタからの論理データは、負/インバースロジックで送受信されます。(1=L、0=H) パリティビットも反転されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 17 TXINV** : TX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット/クリアされます。

0 : TX ピン信号は標準ロジックレベルを使用して機能します ( $V_{DD} = 1$ /アイドル、Gnd=0/マーク)。

1 : TX ピン信号値は反転されます。(V<sub>DD</sub> =0/マーク、Gnd=1/アイドル)。

これにより、TX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 16 RXINV** : RX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット/クリアされます。

0 : RX ピン信号は標準ロジックレベルを使用して機能します ( $V_{DD} = 1$ /アイドル、Gnd=0/マーク)。

1 : RX ピン信号値は反転されます。(V<sub>DD</sub> =0/マーク、Gnd=1/アイドル)。

これにより、RX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 15 SWAP** : TX/RX ピンのスワップ

このビットは、ソフトウェアによってセット/クリアされます。

0 : TX/RX ピンは標準ピンアウトでの定義に従って使用されます。

1 : TX および RX ピンの機能はスワップされます。これにより、別の USART へのクロスワイヤ接続の場合に動作できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 14 LINEN** : LIN モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : LIN モードは無効です。

1 : LIN モードは有効です。

LIN モードでは、USART\_RQR レジスタの SBKRQ ビットを使用して LIN 同期ブレーク (下位 13 ビット) を送信し、LIN 同期ブレークを検出することができます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** *USART が LIN モードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

**ビット 13:12 STOP[1:0]** : STOP ビット

このビットは、ストップビットのプログラミングに使用します。

00 : 1 個のストップビット

01 : 0.5 個のストップビット

10 : 2 個のストップビット

11 : 1.5 個のストップビット

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 11 CLKEN** : クロック有効

このビットによって、CK ピンを有効にできます。

0 : CK ピンは無効です。

1 : CK ピンは有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** *同期モードまたはスマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

- 注: UE ビット値に関係なく、CLKEN = 1 で CK が常に使用可能な場合にスマートカードに CK クロックを正しく供給するには、次のステップを順守する必要があります。
- UE = 0
  - SCEN = 1
  - GTPR 設定(PSC を設定する必要がある場合は、USART\_GTPR レジスタへのシングルアクセスで PSC と GT を設定することを推奨)
  - CLKEN = 1
  - UE = 1

## ビット 10 CPOL : クロック極性

このビットによって、同期モードにおける CK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して動作し、希望するクロック/データ関係になるようにします。

0: 送信ウィンドウの外で、CK ピンはローレベルを維持します。

1: 送信ウィンドウの外で、CK ピンはハイレベルを維持します。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

- 注: 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

## ビット 9 CPHA : クロック位相

このビットは、同期モードでの CK ピンのクロック出力の位相を選択するために使用されます。CPOL ビットと連携して動作し、希望するクロック/データ関係になるようにします (図 335 および 図 336 を参照)。

0: 最初のクロック遷移が最初のデータキャプチャエッジです。

1: 2 番目のクロック遷移が最初のデータキャプチャエッジです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

- 注: 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

## ビット 8 LBCL : 最終ビットのクロックパルス

このビットは、同期モードで送信される最終データビット (MSB) に関連するクロックパルスを、CK ピンに出力する必要があるかどうかを選択するために使用されます。

0: 最終データビットのクロックパルスは、CK ピンに出力されません。

1: 最終データビットのクロックパルスは、CK ピンに出力されます。

- 注意: 最終ビットは、USART\_CR1 レジスタの M ビットによって選択された 7 または 8 または 9 ビットフォーマットに応じて送信された 7 番目または 8 番目または 9 番目のデータビットです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

- 注: 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 7 予約済みであり、リセット値に保持する必要があります。

## ビット 6 LBDIE : LIN ブレーク検出割り込み有効

ブレーク割り込みマスクです (ブレークデリミタを使用したブレーク検出)。

0: 割り込みは禁止されています。

1: USART\_ISR レジスタの LBDF=1 のときには、割り込みが生成されます。

- 注: LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 5 LBDL : LIN ブレーク検出長

このビットでは、10 ビットと 11 ビットのブレーク検出を選択します。

0 : 10 ビットブレーク検出

1 : 11 ビットブレーク検出

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 4 ADDM7 : 7 ビットアドレス検出 / 4 ビットアドレス検出

このビットは、4 ビットアドレス検出と 7 ビットアドレス検出の選択に使用されます。

0 : 4 ビットアドレス検出

1 : 7 ビットアドレス検出 (8 ビットデータモード)

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) に対して行われます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

**注:** トランスミッタが有効なときには、3 つのビット (CPOL、CPHA、LBCL) に書き込まないでください。

### 29.8.3 制御レジスタ 3 (USART\_CR3)

アドレスオフセット : 0x08

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFIE	WUS			SCARCNT2[0]			Res.
									rW	rW	rW	rW	rW	rW		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSSEL	IRLP	IREN	EIE	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	v	v	rW	rW	rW	rW	

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 WUFIE : STOP モードからのウェイクアップ割り込み有効

このビットは、ソフトウェアによってセット / クリアされます。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの WUF=1 のときには、USART 割り込みが生成されます。

**注:** WUFIE は、STOP モードに入る前にセットする必要があります。

WUF 割り込みは、STOP モードでのみアクティブです。

USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 21:20 **WUS[1:0]** : STOP モードからのウェイクアップ割り込みフラグ選択

このビットフィールドは、WUF を有効にするイベントを指定します (STOP モードからのウェイクアップフラグ)。

00 : WUF はアドレス一致時に有効になります (ADD[7:0] および ADDM7 による定義に従って)。

01 : 予約済み

10 : WuF はスタートビット検出時に有効になります。

11 : WUF は RXNE 時に有効になります。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。**

ビット 19:17 **SCARCNT[2:0]** : スマートカード自動再試行カウンタ

このビットフィールドは、スマートカードにおける送受信の再試行回数を指定します。

送信モードでは、送信エラーが生成されるまでの送信の自動再試行回数を指定します (FE ビットをセット)。

受信モードでは、受信エラーが生成されるまでの受信の試行エラー回数を指定します (RXNE および PE ビットをセット)。

このビットフィールドは、USART が無効 (UE=0) のときのみプログラムする必要があります。

USART が有効になると (UE=1)、このビットフィールドは再送信を停止するために 0x0 にのみ書き込み可能です。

0x0 : 再送信無効 - 送信モードでの自動再送信禁止

0x1 から 0x7 : 自動再送信試行回数 (信号エラーの生成前)

**注:** **スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。**

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **DEP** : ドライバ有効極性選択

0 : DE 信号はアクティブハイです。

1 : DE 信号はアクティブローです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 29.4 : USART の実装 (885 ページ) を参照してください。**

ビット 14 **DEM** : ドライバ有効モード

このビットにより、DE 信号によって外部トランシーバ制御を有効にできます。

0 : DE 機能は無効です。

1 : DE 機能は有効です。DE 信号は RTS ピンで出力されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 29.4 : USART の実装 (885 ページ) を参照してください。**

ビット 13 **DDRE** : 受信エラー時 DMA 無効

0 : 受信エラーの場合、DMA は無効になります。対応するエラーフラグはセットされますが、RXNE は 0 に保たれ、オーバーランを防ぎます。結果として、DMA リクエストはアサートされないため、エラーのあるデータは転送されず (DMA リクエストなし)、次の正しい受信データが転送されます (スマートカードモードで使用される)。

1 : 受信エラーの後、DMA は無効化されます。対応するエラーフラグと RXNE がセットされます。エラーフラグがクリアされるまで、DMA リクエストはマスクされます。つまり、ソフトウェアは、まず、DMA リクエストを無効にするか (DMAR=0)、RXNE をクリアしてから、エラーフラグをクリアする必要があります。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** **受信エラーは、パリティエラー、フレーミングエラー、またはノイズエラーです。**

**ビット 12 OVRDIS** : オーバーラン無効

このビットは、受信オーバーラン検出を無効にするために使用されます。

0 : オーバーランエラーフラグ、ORE は、受信データが読み出される前に新しいデータを受信したときにセットされます。

1 : オーバーラン機能は無効です。RXNE フラグがまだセットされている間に新しいデータを受信した場合、

ORE フラグはセットされず、新しく受信されたデータが USART\_RDR レジスタの前の内容に上書きされます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** この制御ビットにより、データを読み出さずに通信フローをチェックできます。

**ビット 11 ONEBIT** : 1 サンプルビット方式有効

このビットによって、サンプル方式を選択できます。1 サンプルビット方式が選択されると、ノイズ検出フラグ (NF) が無効になります。

0 : 3 サンプルビット方式

1 : 1 サンプルビット方式

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ONEBIT の機能はデータビットのみに適用され、スタートビットには適用されません。

**ビット 10 CTSIE** : CTS 割り込み有効

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタの CTSIF=1 のときには、割り込みが生成されます。

**注 :** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 9 CTSE** : CTS 有効

0 : CTS ハードウェアフロー制御が無効です。

1 : CTS モードが有効です。データは CTS 入力のアサート (0 に関係) されている場合にのみ転送されます。データの送信中に CTS 入力にネゲートされた場合、送信は停止前に完了します。CTS がネゲートされている間にデータがデータレジスタに書き込まれた場合、CTS がアサートされるまで送信は延期されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 8 RTSE** : RTS 有効

0 : RTS ハードウェアフロー制御が無効です。

1 : RTS 出力は有効であり、レシーババッファにスペースがあるときにのみ、データがリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。データを受信できるとき、RTS 出力がアサートされます (0 にプルされます)。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 7 DMAT** : DMA 有効トランスミッタ

このビットは、ソフトウェアでセット/クリアされます。

1 : DMA モードは送信に有効です。

0 : DMA モードは送信に無効です。

**ビット 6 DMAR** : DMA 有効レシーバ

このビットは、ソフトウェアでセット/クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。



**ビット 5 SCEN** : スマートカードモード有効

このビットはスマートカードモードを有効にするために使用します。

0 : スマートカードモードが無効です。

1 : スマートカードモードが有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** *USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

**ビット 4 NACK** : スマートカード NACK 有効

0 : パリティエラーの際の NACK 転送が無効です。

1 : パリティエラー時の NACK 転送が有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** *USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

**ビット 3 HDSEL** : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**ビット 2 IRLP** : IrDA 低電力

このビットは、通常と低電力の IrDA モードの選択に使用されます。

0 : 通常モード

1 : 低電力モード

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** *IrDA モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

**ビット 1 IREN** : IrDA モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : IrDA は無効です。

1 : IrDA は有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注 :** *IrDA モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

**ビット 0 EIE** : エラー割り込みイネーブル

エラー割り込み有効ビットは、フレーミングエラー、オーバーランエラー、またはノイズフラグ (USART\_ISR レジスタの FE=1 または ORE=1 または NF=1) の場合に割り込み生成を有効にするために必要です。

0 : 割り込みは禁止されています。

1 : USART\_ISR レジスタで FE=1、ORE=1、または NF=1 になると、割り込みが生成されます。

## 29.8.4 ボーレートレジスタ (USART\_BRR)

このレジスタは、USART が無効 (UE=0) のときのみ書き込むことができます。自動ボーレート検出モードでハードウェアによって自動的に更新されます。

アドレスオフセット : 0x0C

リセット値 : 0x0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **BRR[15:4]**

$$BRR[15:4] = USARTDIV[15:4]$$

ビット 3:0 **BRR[3:0]**

OVER8 = 0 のとき、 $BRR[3:0] = USARTDIV[3:0]$ 。

OVER8 = 1 のとき、

$BRR[2:0] = USARTDIV[3:0]$  であり、右に 1 ビットシフトされます。

$BRR[3]$  は、クリアされたままにする必要があります。

## 29.8.5 ガード時間およびプリスケアラレジスタ (USART\_GTPR)

アドレスオフセット : 0x10

リセット値 : 0x0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **GT[7:0]** : ガード時間値

このビットフィールドは、ガード時間値をボークロック周期数でプログラムするために使用します。これはスマートカードモードで使用します。このガード時間値の後は転送完了フラグがセットされます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 7:0 **PSC[7:0]** : プリスケアラ値

**IrDA 低電力および IrDA 通常モード :**

PSC[7:0] = IrDA 通常および低電力ポーレート

USART クロックソースを分周して低電力周波数を得るためのプリスケアラのプログラミングに使用します。

ソースクロックは、レジスタに与えられた値 (上位 8 ビット) で分周されます。

00000000 : 予約済み - この値はプログラミングしないでください。

00000001 : クロックソースは 1 で分周されます。

00000010 : クロックソースは 2 で分周されます。

...

**スマートカードモード :**

PSC[4:0] : プリスケアラ値

USART クロックソースを分周してスマートカードのクロックを提供するプリスケアラのプログラミングに使用します。

レジスタで指定された値 (上位 5 ビット) を 2 倍して、ソースクロック周波数の分周比を求めます。

00000 : 予約済み - この値はプログラミングしないでください。

00001 : クロックソースは 2 で分周されます。

00010 : クロックソースは 4 で分周されます。

00011 : クロックソースは 6 で分周されます。

...

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

**注:** スマートカードモードが使用される場合、ビット [7:5] はクリアされたままにする必要があります。

スマートカードモードや IrDA モードがサポートされない場合、このビットフィールドは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

## 29.8.6 レシーバタイムアウトレジスタ (USART\_RTOR)

アドレスオフセット : 0x14

リセット値 : 0x0000

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



ビット 31:24 BLEN[7:0] : ブロック長

このビットフィールドは、受信時のスマートカード T=1 のブロック長を指定します。この値は、情報文字の数 + エピローグフィールドの長さ (1-LEC/2-CRC) - 1 と等しくなります。

例 :

BLEN = 0 -> 0 情報文字 + LEC

BLEN = 1 -> 0 情報文字 + CRC

BLEN = 255 -> 254 情報文字 + CRC (合計 256 文字)

スマートカードモードでは、TXE=0 のときにブロック長カウンタがリセットされます。

このビットフィールドは、他のモードでも使用できます。この場合、RE=0 (受信無効) のとき、および/または EOBCF ビットが 1 に書き込まれたときに、ブロック長カウンタがリセットされます。

**注 :** この値は、ブロック受信の開始後にプログラムできます (プロローグフィールドの LEN 文字のデータを使用)。必ず受信したブロックにつき一度だけプログラムするようにしてください。

ビット 23:0 RTO[23:0] : レシーバタイムアウト値

このビットフィールドは、レシーバタイムアウト値をビット持続時間の数で指定します。

標準モードでは、最後の受信キャラクタの後、RTO 値を超える間、新しいスタートビットが検出されなかった場合、RTOF フラグがセットされます。

スマートカードモードでは、この値は CWT および BWT を実装するために使用されます。詳細については、スマートカードのセクションを参照してください。

この場合、タイムアウト測定は最後の受信キャラクタのスタートビットから始めて行われます。

**注 :** この値は、受信キャラクタごとにプログラムされる必要があります。

**注 :** RTOR は、動作中に書き込むことができます。新しい値がカウンタ以下の場合、RTOF フラグがセットされます。

レシーバタイムアウト機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって 0x00000000 に強制的に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

## 29.8.7 リクエストレジスタ (USART\_RQR)

アドレスオフセット : 0x18

リセット値 : 0x0000

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

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **TXFRQ** : 送信データー掃リクエスト

このビットに 1 を書き込むと、TXE フラグがセットされます。

これにより、送信データを破棄できます。このビットは、エラー (NACK) によりデータが送信されなかった場合、および USART\_ISR レジスタで FE フラグがアクティブである場合に、スマートカードモードでのみ使用する必要があります。

USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。

ビット 3 **RXFRQ** : 受信データー掃リクエスト

このビットに 1 を書き込むと、RXNE フラグがクリアされます。

これにより、受信したデータを読み出さずに破棄して、オーバーラン条件を避けることができます。

ビット 2 **MMRQ** : ミュートモードリクエスト

このビットに 1 を書き込むと、USART はミュートモードになり、RWU フラグがセットされます。

ビット 1 **SBKRQ** : ブレーク送信リクエスト

このビットに 1 を書き込むと、SBKF フラグがセットされ、送信マシンが使用可能になるとすぐに、ラインで BREAK を送信するリクエストが発行されます。

**注:** アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブレークキャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを待つ必要があります。

ビット 0 **ABRRQ** : 自動ポーレートリクエスト

このビットに 1 を書き込むと、USART\_ISR の ABRF フラグがリセットされ、次の受信データフレームでの自動ポーレート測定をリクエストします。

**注:** USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 29.4 : USART の実装 \(885 ページ\)](#) を参照してください。

## 29.8.8 割り込みおよびステータスレジスタ (USART\_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0200 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	Res.	EOBF	RTOF	CTS	CTSIF	LBDIF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:23 予約済みであり、リセット値に保持する必要があります。



ビット 22 **REACK** : 受信有効確認応答フラグ

このビットは、受信有効値が USART によって考慮されるときに、ハードウェアによってセット/リセットされます。

STOP モードからのウェイクアップがサポートされる時、STOP モードに入る前に USART が受信できる状態であることを REACK フラグを使用して確認できます。

ビット 21 **TEACK** : 送信有効確認応答フラグ

このビットは、送信有効値が USART によって考慮されるときに、ハードウェアによってセット/リセットされます

USART\_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20 **WUF** : STOP モードからのウェイクアップフラグ

このビットは、ウェイクアップイベントが検出されたときに、ハードウェアによってセットされます。イベントは、WUS ビットフィールドによって定義されます。USART\_ICR レジスタの WUCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR3 レジスタの WUFIE=1 である場合、割り込みが生成されます。

**注：** *UESM がクリアされると、WUF フラグもクリアされます。*

*WUF 割り込みは、STOP モードでのみアクティブです。*

*USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

ビット 19 **RWU** : レシーバのミュートモードからのウェイクアップ

このビットは、USART がミュートモードかどうかを示します。ウェイクアップ/ミュートシーケンスが認識されたときに、ハードウェアによってクリア/セットされます。ミュートモード制御シーケンス (アドレスまたは IDLE) は、USART\_CR1 レジスタの WAKE ビットによって選択されます。

IDLE モードでのウェイクアップが選択されたとき、このビットは USART\_RQR レジスタの MMRQ ビットに 1 を書き込むことによって、ソフトウェアによってのみセットできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

ビット 18 **SBKF** : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。USART\_RQR レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタは送信されません。

1 : ブレークキャラクタは送信されます。

ビット 17 **CMF** : キャラクター一致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。USART\_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR1 レジスタの CMIE=1 の場合、割り込みが生成されます。

0 : キャラクター一致は検出されていません。

1 : キャラクター一致が検出されました。

ビット 16 **BUSY** : ビジーフラグ

このビットは、ハードウェアによってセット/リセットされます。RX ラインで通信中 (スタートビットの検出時) はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : USART はアイドルです (受信なし)。

1 : 受信中です。

**ビット 15 ABRF** : 自動ポーレートフラグ

このビットは、自動ポーレートがセットされたときにハードウェアによってセットされ (RXNE もセットされ、RXNEIE=1 の場合は割り込みが生成されます)、または、自動ポーレート操作が成功せずに完了したときにセットされます (ABRE=1) (この場合、ABRE、RXNE、および FE もセットされます)。新しい自動ポーレート検出をリクエストするために、USART\_RQR レジスタの ABRRQ に 1 を書き込むことによって、ソフトウェアによってクリアされます。

**注:** *USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

**ビット 14 ABRE** : 自動ポーレートエラー

このビットは、ポーレート測定が失敗した場合に、ハードウェアによってセットされます (範囲外のポーレートまたはキャラクタ比較の失敗)。

USART\_CR3 レジスタの ABRRQ ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

**注:** *USART が自動ポーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

ビット 13 予約済みであり、リセット値に保持する必要があります。

**ビット 12 EOCF** : ブロック終了フラグ

このビットは、完全なブロックが受信されたときに、ハードウェアによってセットされます (たとえば、T=1 スマートカードモード)。検出は、受信バイト数が BLEN + 4 以上である場合に行われます (ブロックの開始時から、プロローグを含む)。

USART\_CR2 レジスタの EOBIE=1 である場合、割り込みが生成されます。

USART\_ICR レジスタの EOBCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ブロック終了に達していません。

1 : ブロック終了 (文字数) に達しました。

**注:** *スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。*

**ビット 11 RTOF** : レシーバタイムアウト

このビットは、RTOR レジスタでプログラムされたタイムアウト値が通信なしで経過したときに、ハードウェアによってセットされます。USART\_ICR レジスタの RTOCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR1 レジスタの RTOIE=1 の場合、割り込みが生成されます。

スマートカードモードでは、タイムアウトは CWT または BWT タイミングに対応します。

0 : タイムアウト値に達していません。

1 : データを受信せずにタイムアウト値に達しました。

**注:** *時間が RTOR レジスタでプログラムされた値に等しい場合、2 つのキャラクタが分離され、RTOF はセットされません。この時間がこの値に 2 サンプル時間 (オーバーサンプリング方式によって 2/16 または 2/8) を加えた値を超える場合、RTOF フラグがセットされます。*

*カウンタは RE=0 の場合でもカウントしますが、RTOF は RE=1 のときのみセットされます。RE がセットされたときにタイムアウトがすでに経過していた場合、RTOF はセットされます。*

*USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

**ビット 10 CTS** : CTS フラグ

このビットは、ハードウェアによってセット/リセットされます。CTS 入力ピンのステータスの反転コピーです。

0 : CTS ラインはセットされました。

1 : CTS ラインはリセットされました。

**注:** *ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。*

**ビット 9 CTSIF** : CTS 割り込みフラグ

このビットは、CTSE ビットがセットされていた場合、CTS 入力がトグルしたときにハードウェアによってセットされます。USART\_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

また、USART\_CR3 レジスタで CTSIE=1 であれば、割り込みが生成されます。

0 : CTS ステータスラインでの変更はありません。

1 : CTS ステータスラインでの変更がありました。

**注 :** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

**ビット 8 LBDF** : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。USART\_ICR レジスタの LBDCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR2 レジスタの LBDIE=1 である場合、割り込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

**注 :** USART が LIN モードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

**ビット 7 TXE** : 送信データレジスタエンプティ

このビットは、USART\_TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。このビットは、USART\_TDR レジスタへの書き込みによってクリアされます。

TXE フラグは、USART\_RQR レジスタの TXFRQ に 1 を書き込んでクリアし、データを破棄することもできます (スマートカード T=0 モードでの送信失敗の場合のみ)。

USART\_CR1 レジスタの TXEIE ビット =1 の場合、割り込みが生成されます。

0 : データはシフトレジスタに転送されません。

1 : データはシフトレジスタに転送されます。

**注 :** このビットは、シングルバッファ送信時に使用されます。

**ビット 6 TC** : 送信完了

データを含むフレームの送信が完了し、TXE がセットされている場合、このビットはハードウェアによってセットされます。USART\_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。USART\_ICR レジスタの TCCF に 1 を書き込むことによって、または USART\_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。

0 : 送信は完了していません。

1 : 送信は完了しています。

**注 :** TE ビットがリセットされ、送信中でなかった場合、TC ビットはただちにセットされます。

**ビット 5 RXNE** : 読み出しデータレジスタノットエンプティ

このビットは、RDR シフトレジスタの内容が USART\_RDR レジスタに転送されると、ハードウェアによってセットされます。このビットは、USART\_RDR レジスタへの読み出しによってクリアされます。RXNE フラグは、USART\_RDR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。

USART\_CR1 レジスタの RXNEIE=1 の場合、割り込みが生成されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。



**ビット 4 IDLE** : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。USART\_CR1 レジスタの IDLEIE=1 である場合、割り込みが生成されます。USART\_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

**注 :** *RXNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。*

*ミュートモードが有効な場合 (MME=1)、USART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。*

**ビット 3 ORE** : オーバーランエラー

このビットは、RXNE=1 のときに、シフトレジスタで現在受信中のデータを RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USART\_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR1 レジスタの RXNEIE=1 または EIE=1 の場合、割り込みが生成されます。

0 : オーバーランエラーはありません。

1 : オーバーランエラーが検出されました。

**注 :** *このビットがセットされると、RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割り込みが生成されます。*

*USART\_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。*

**ビット 2 NF** : START ビットノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。USART\_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

**注 :** *このビットは、割り込みを生成する RXNE ビットと同時に出現するため、割り込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NF フラグがセットされた場合、割り込みが生成されます。*

**注 :** *ラインがノイズフリーであるとき、NF フラグを無効にして、ONEBIT ビットに 1 をプログラミングして偏差に対する USART の許容誤差を増加させることができます (セクション 29.5.5 : クロック偏差に対する USART レシーバの許容誤差 (900 ページ) を参照)。*

**ビット 1 FE** : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。USART\_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでは、送信時、成功せずに (カードがデータフレームを NACK) 最大送信試行回数に達すると、このビットがセットされます。

USART\_CR1 レジスタの EIE=1 の場合、割り込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

**ビット 0 PE** : パリティエラー

このビットは、レシーバモードでパリティエラーが発生したときに、ハードウェアによってセットされます。USART\_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USART\_CR1 レジスタの PEIE=1 の場合、割り込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

## 29.8.9 割り込みフラグクリアレジスタ (USART\_ICR)

アドレスオフセット : 0x20

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											rc_w1			rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EOBCF	RTOCF	Res.	CTSCF	LBDCF	Res.	TCCF	Res.	IDLECF	ORECF	NCF	FECF	PECF
			rc_w1	rc_w1		rc_w1	rc_w1		rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **WUCF** : STOP モードからのウェイクアップフラグクリア

このビットに 1 を書き込むと、USART\_ISR レジスタの WUF フラグがクリアされます。

**注:** USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CMCF** : キャラクター一致フラグクリア

このビットに 1 を書き込むと、USART\_ISR レジスタの CMF フラグがクリアされます。

ビット 16:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **EOBCF** : ブロック終了クリアフラグ

このビットに 1 を書き込むと、USART\_ISR レジスタの EOCIF フラグがクリアされます。

**注:** USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 11 **RTOCF** : レシーバタイムアウトフラグクリア

このビットに 1 を書き込むと、USART\_ISR レジスタの RTOF フラグがクリアされます。

**注:** USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CTSCF** : CTS フラグクリア

このビットに 1 を書き込むと、USART\_ISR レジスタの CTSIF フラグがクリアされます。

**注:** ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 8 **LBDCF** : LIN ブレーク検出クリアフラグ

このビットに 1 を書き込むと、USART\_ISR レジスタの LBDF フラグがクリアされます。

**注:** LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 29.4 : USART の実装 (885 ページ) を参照してください。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TCCF** : 送信完了フラグクリア

このビットに 1 を書き込むと、USART\_ISR レジスタの TC フラグがクリアされます。

ビット 5 予約済みであり、リセット値に保持する必要があります。

- ビット 4 **IDLECF** : アイドルライン検出フラグクリア  
このビットに 1 を書き込むと、USART\_ISR レジスタの IDLE フラグがクリアされます。
- ビット 3 **ORECF** : オーバーランエラーフラグクリア  
このビットに 1 を書き込むと、USART\_ISR レジスタの ORE フラグがクリアされます。
- ビット 2 **NCF** : ノイズ検出フラグクリア  
このビットに 1 を書き込むと、USART\_ISR レジスタの NF フラグがクリアされます。
- ビット 1 **FECF** : フレーミングエラーフラグクリア  
このビットに 1 を書き込むと、USART\_ISR レジスタの FE フラグがクリアされます。
- ビット 0 **PECF** : パリティエラーフラグクリア  
このビットに 1 を書き込むと、USART\_ISR レジスタの PE フラグがクリアされます。

## 29.8.10 受信データレジスタ (USART\_RDR)

アドレスオフセット : 0x24

リセット値 : 未定義

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **RDR[8:0]** : 受信データ値

受信データキャラクタを含みます。

RDR レジスタは、入カシフトレジスタと内部バスとの間にパラレルインタフェースを提供します (図 323 を参照)。

パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

## 29.8.11 送信データレジスタ (USART\_TDR)

アドレスオフセット : 0x28

リセット値 : 未定義

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

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **TDR[8:0]** : 送信データ値

送信されるデータキャラクタを含みます。

TDR レジスタは、内部バスと出カシフトレジスタとの間にパラレルインタフェースを提供します (図 323 を参照)。

パリティを有効にして (USART\_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。

**注:** このレジスタは、TXE=1 のときのみ書き込むことができます。

## 29.8.12 USART レジスタマップ

次の表に、USART のレジスタマップとリセット値を示します。

表 164. USART レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	USART_CR1	Res	Res	Res	M1	EOBIE	RTOIE	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UE		
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	USART_CR2	ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD1	ABRMOD0	ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]	CLKEN	CPOL	CPHA	LBCL	Res	LBIDIE	LBIDL	ADDM7	Res	Res	Res	Res		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	USART_CR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUFIE	WUS	SCARCNT[2:0]			Res	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSSEL	IRLP	IREN	EIE	
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	USART_BRR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BRR[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	USART_GTPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GT[7:0]					PSC[7:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	USART_RTOR	BLEN[7:0]							RTO[23:0]																									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	USART_RQR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ	
	リセット値																												0	0	0	0	0	0
0x1C	USART_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	Res	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

表 164. USART レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x20	USART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	EOBCF	RTOCF	Res.	CTSCF	LBDCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NCF	FECF	PECF				
	リセット値												0			0						0	0	0	0	0	0	0	0	0	0	0	0	0					
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]														
	リセット値																								X	X	X	X	X	X	X	X	X	X	X				
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]														
	リセット値																								X	X	X	X	X	X	X	X	X	X	X				

レジスタ境界アドレスについては、[セクション 3.2 \(50 ページ\)](#) を参照してください。

## 30 シリアルペリフェラルインタフェース/I2S (SPI/I2S)

### 30.1 概要

SPI/I<sup>2</sup>S インタフェースを使用して、SPI プロトコルまたは I<sup>2</sup>S オーディオプロトコルに基づき外部デバイスと通信することができます。SPI または I<sup>2</sup>S モードはソフトウェアによって選択可能です。デバイスのリセット後は、デフォルトで SPI モトローラモードが選択されます。

SPI (シリアルペリフェラルインタフェース) プロトコルは、外部デバイスとの半二重、全二重、および単方向の同期シリアル通信をサポートしています。このインタフェースはマスタとして設定することも可能で、その場合、外部スレーブデバイスに通信クロック (SCK) を供給します。このインタフェースは、マルチマスタ設定で動作することもできます。

I<sup>2</sup>S プロトコルも、同期シリアル通信インタフェースです。スレーブまたはマスタモードで、全二重および半二重通信として動作することができます。フィリップス I<sup>2</sup>S 規格、MSB/LSB 詰め規格、PCM 規格など、4 つのオーディオ規格に対応できます。

**注：** *STM32F303x6/8 および STM32F328x8 に I2S はありません。*

### 30.2 SPI の主な機能

- マスタまたはスレーブ動作
- 3 本のラインでの全二重同期転送
- 2 本のラインでの半二重同期転送 (双方向データライン有り)
- 2 本のラインでの単方向同期転送 (単方向データライン有り)
- 4 ~ 16 ビットのデータサイズ選択
- マルチマスタモード機能
- 8 個のマスタモードポーレートプリスケアラ (最大周波数  $f_{PCLK}/2$ )
- スレーブモード周波数 (最大周波数  $f_{PCLK}/2$ )
- マスタとスレーブの両方に対するハードウェア/ソフトウェアによる NSS 管理 : マスタ/スレーブ動作の動的切り替え
- クロックの極性と位相をプログラム可能
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 専用の送受信フラグ (割り込み機能付き)
- SPI バスビジステータスフラグ
- SPI モトローラモードをサポート
- ハードウェア CRC 機能による信頼性の高い通信 :
  - Tx モードでは CRC 値を最終バイトとして送信可能
  - 最終受信バイトに対する CRC エラーの自動チェック
- マスタモードの障害、オーバーランの各フラグ (割り込み機能付き)
- CRC エラーフラグ
- DMA 機能付きの 2 つの 32 ビット内蔵 Rx および Tx FIFO
- SPI TI モードをサポート

## 30.3 I2S の主な機能

- 全二重通信
- 半二重通信 (トランスミッタまたはレシーバのみ)
- マスタまたはスレーブ動作
- 正確なオーディオサンプリング周波数 (8~192 kHz) を実現するプログラム可能な 8 ビットのリニアプリスケアラ
- 16、24、または 32 ビットのデータフォーマット
- パケットフレームはオーディオチャネルによって 16 ビット (16 ビットデータフレーム) または 32 ビット (16、24、32 ビットデータフレーム) に固定。
- プログラム可能なクロック極性 (定常状態)
- スレーブ送信モードのアンダーランフラグ、受信モード (マスタおよびスレーブ) のオーバーランフラグ、受信モードと送信モード (スレーブの場合のみ) のフレームエラーフラグ
- 送受信用の 16 ビットレジスタ (両チャンネルサイドに対して 1 個のデータレジスタ)
- 以下の I<sup>2</sup>S プロトコルをサポート。
  - フィリップス I<sup>2</sup>S 規格
  - MSB 詰め規格 (左詰め)
  - LSB 詰め規格 (右詰め)
  - PCM 規格 (16 ビットチャンネルフレーム、または 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームでの、ショートおよびロングフレーム同期付き)
- データ方向は常に MSB ファースト。
- 送受信 (16 ビット幅) 用の DMA 機能
- 外部オーディオコンポーネントを駆動するためのマスタクロックを出力可能。周波数比は、 $256 \times F$  ( $F_S$  はオーディオサンプリング周波数) に固定。
- I<sup>2</sup>S (I2S2 と I2S3) クロックは、I2S\_CKIN ピンに割り当てられた外部クロックから取得可能

## 30.4 SPI/I2S の実装

このマニュアルでは、SPI1、SPI2、SPI3、および SPI4 に実装されているすべての機能について説明しています。SPI1 と SPI4 は、I<sup>2</sup>S モードを除くすべての機能をサポートしています。

**注：** STM32F303x6/8 および STM32F328x8 デバイスでは、SPI1 のみ使用可能です。

表 165. STM32F303x6/8 および STM32F328x8 SPI 実装

SPI の機能 <sup>(1)</sup>	SPI1
ハードウェア CRC 計算	X
Rx/Tx FIFO	X
NSS パルスモード	X
I2S モード	-
TI モード	X

1. X：サポートされています。

表 166. STM32F303xB/C/D/E、STM32F358xC および STM32F398xE SPI 実装

SPI の機能 <sup>(1)</sup>	SPI1	SPI2	SPI3	SPI4 <sup>(2)</sup>
ハードウェア CRC 計算	X	X	X	X
Rx/Tx FIFO	X	X	X	X
NSS パルスモード	X	X	X	X
I2S モード	-	X	X	-
TI モード	X	X	X	X

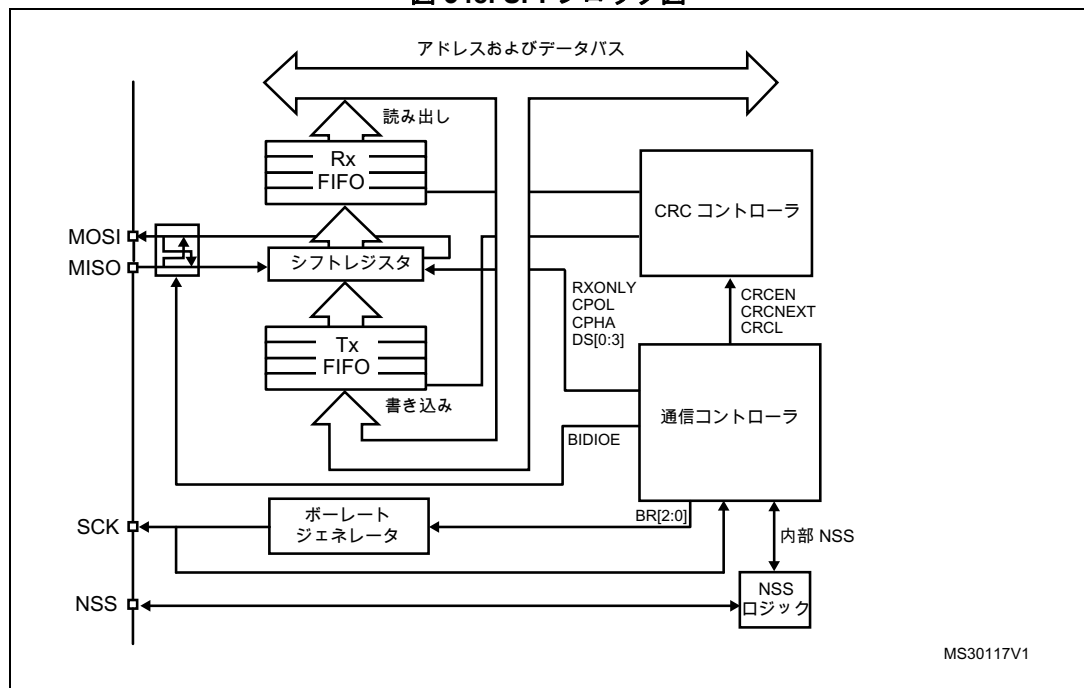
1. X: サポートされています。
2. SPI4 は STM32F303xD/E でのみ使用可能です。

## 30.5 SPI の機能説明

### 30.5.1 概要

SPI では、MCU と外部デバイス間の同期シリアル通信が可能です。アプリケーションソフトウェアは、ステータスフラグをポーリングするか、または専用の SPI 割り込みを使用することで、通信を管理することができます。SPI の主要要素およびそれらの相互作用を以下のブロック図 (図 348) に示します。

図 348. SPI ブロック図



MS30117V1



4本のI/Oピンが外部デバイスとのSPI通信専用に使われます。

- **MISO** : マスターイン/スレーブアウトデータ。一般に、このピンは、スレーブモードではデータの送信に、マスタモードではデータの受信に使われます。
- **MOSI** : マスターアウト/スレーブインデータ。一般に、このピンは、マスタモードではデータの送信に、スレーブモードではデータの受信に使われます。
- **SCK** : SPI マスタではシリアルクロックの出力に、SPI スレーブでは入力に使われます。
- **NSS** : スレーブ選択用のピンです。このピンは、SPI および NSS の設定に応じて、以下のいずれかに使えます。
  - 個々の通信用スレーブデバイスを選択する
  - データフレームを同期させる
  - 複数のマスタ間での競合を検出する

詳細は、[セクション 30.5.5 : スレーブ選択 \(NSS\) ピンの管理](#)を参照してください。

SPIバスを使用することで、1つのマスタデバイスと1つ以上のスレーブデバイスとの間で通信することができます。バスは2本以上の線から成り、1本はクロック信号用、その他はデータの同期転送用です。SPIノード間でのデータ交換とそれらのスレーブ選択信号管理に応じて、その他の信号を追加することができます。

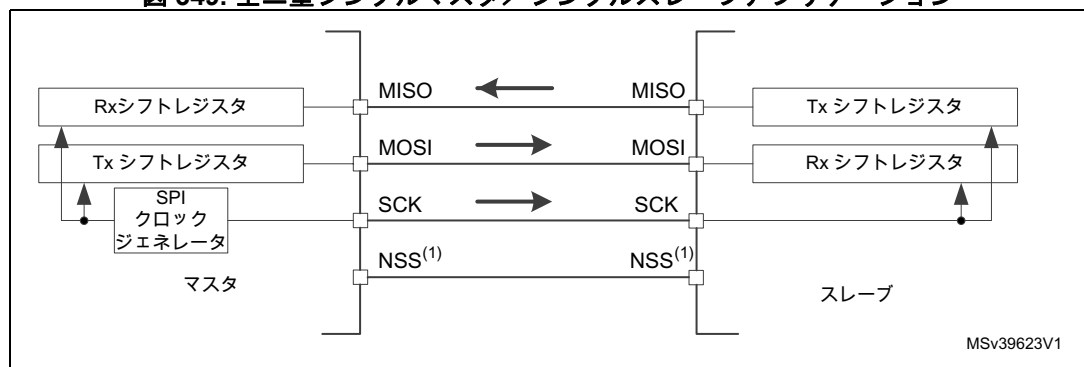
## 30.5.2 マスタとスレーブの1対1の通信

SPIを使用することで、MCUは対象となるデバイスやアプリケーション要件に応じたさまざまな設定で通信ができます。これらの設定には、2または3本の線（ソフトウェアNSS管理あり）、あるいは3または4本の線（ハードウェアNSS管理あり）が使われます。通信は常にマスタによって開始されます。

### 全二重通信

SPIは、デフォルトで全二重通信に設定されます。この設定では、マスタおよびスレーブのシフトレジスタは、MOSIピンとMISOピンの間に2本の単方向ラインを介してリンクされます。SPI通信の間、データはマスタから供給されるSCKクロックのエッジに同期してシフトされます。マスタは、送信すべきデータをMOSIライン経由でスレーブに送信し、MISOライン経由でスレーブからデータを受信します。データフレーム転送が完了した（すべてのビットがシフトされた）時点で、マスタとスレーブの間で情報が交換されます。

図 349. 全二重シングルマスタ/シングルスレーブアプリケーション

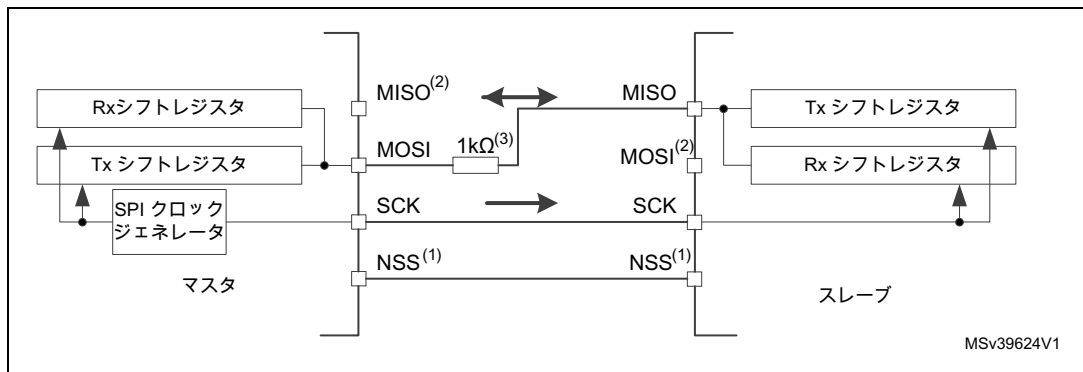


1. NSSピンを使用して、マスタとスレーブ間のハードウェア制御フローを実現できます。オプションで、ペリフェラルによってピンを未使用状態のままにできます。そのとき、マスタとスレーブ両方に対して内部操作を行う必要があります。詳細については、[セクション 30.5.5 : スレーブ選択 \(NSS\) ピンの管理](#)を参照してください。

## 半二重通信

SPIx\_CR1 レジスタの BIDIMODE ビットをセットすることで、SPI は半二重モードで通信できます。この設定では、1 本の交差接続ラインを使用して、マスタとスレーブのシフトレジスタを互いにリンクさせます。この通信中に、データは SCK クロックのエッジに同期して、シフトレジスタ間でシフトされます。シフトの方向は、マスタとスレーブの両方が SPIx\_CR1 レジスタの BDIOE ビットを使用して相互に選択された転送方向となります。この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、他のアプリケーションで使用でき、GPIO として機能します。

図 350. 半二重シングルマスタ/シングルスレーブアプリケーション



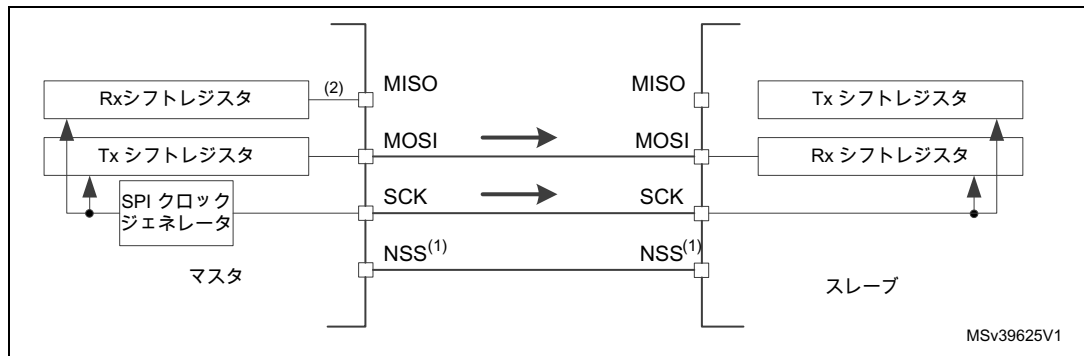
1. NSS ピンを使用して、マスタとスレーブ間のハードウェア制御フローを実現できます。オプションで、ペリフェラルによってピンを未使用状態のままにできます。そのとき、マスタとスレーブ両方に対して内部操作を行う必要があります。詳細については、[セクション 30.5.5: スレーブ選択 \(NSS\) ピンの管理](#) を参照してください。
2. この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、GPIO として使用できます。
3. 双方向モードで動作している 2 つのノード間で通信方向を同期しないよう変更した場合、重大な状況が発生し、新しいトランスミッタが共通データラインにアクセスしている状態で、これまでのトランスミッタはライン上で逆の値を保持し続けます（この値は SPI 設定と通信データの影響を受けます）。そのとき、両方のノードが競合し、次のノードが対応する方向設定に変更するまで、共有データラインで一時的に逆の出力レベルも供給されます。このモードでは MISO ピンと MOSI ピンの間に直列抵抗を挿入して、出力を保護し、この状況で流れる電流を制限することをお奨めします。

## 単方向通信

SPI は、SPIx\_CR2 レジスタの RXONLY ビットを使用して送信専用または受信専用を設定することにより、単方向モードで通信できます。この設定では、マスタとスレーブのシフトレジスタ間の転送に使用するのは 1 ラインのみです。残りの MISO ピンと MOSI ピンのペアは通信には使用されず、標準の GPIO として使用できます。

- **送信専用モード (RXONLY = 0) の場合**：設定は全二重の場合と同じです。アプリケーションは、未使用の入力ピンでキャプチャされた情報を無視する必要があります。このピンは標準の GPIO として使用できます。
- **受信専用モード (RXONLY = 1) の場合**：アプリケーションにて、RXONLY ビットをセットすることによって、SPI 出力機能を無効にできます。スレーブ設定では、MISO 出力が無効化され、ピンを GPIO として使用することができます。スレーブ選択信号がアクティブな間は、スレーブは MOSI ピンからデータを受信し続けます（[30.5.4: マルチマスタ通信](#) を参照）。データバッファの設定に応じて、受信データイベントが出現します。マスタ設定では、MOSI 出力が無効化され、ピンを GPIO として使用することができます。SPI が有効である間はクロック信号が生成され続けます。クロックを停止させる唯一の方法は、クロックの設定に応じて、RXONLY ビットまたは SPE ビットをクリアし、MISO ピンからの受信パターンが終了し、データバッファ構造への書き込みが行われるまで待つことです。

図 351. 単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ)



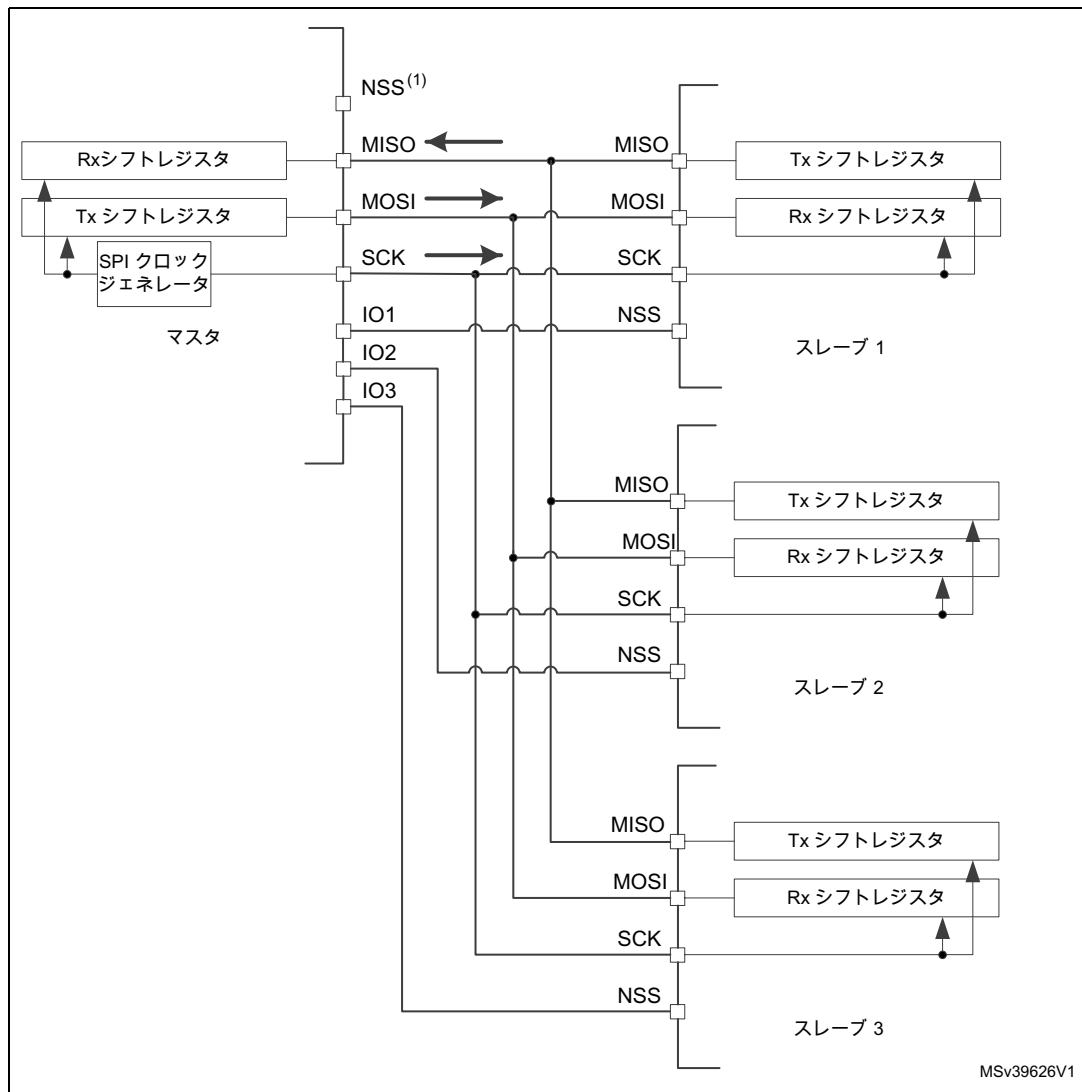
1. NSS ピンを使用して、マスタとスレーブ間のハードウェア制御フローを実現できます。オプションで、ペリフェラルによってピンを未使用状態のままにできます。そのとき、マスタとスレーブ両方に対して内部操作を行う必要があります。詳細については、[セクション 30.5.5: スレーブ選択 \(NSS\) ピンの管理](#)を参照してください。
2. トランスミッタ Rx シフトレジスタの入力で予期しない入力情報がキャプチャされます。トランスミッタ受信フローに関連するすべてのイベントは、標準の送信専用モード (たとえば、OVF フラグ) では無視する必要があります。
3. この設定では、両方のMISO ピンを GPIO として使用できます。

**注:** **すべての単方向通信は、トランザクション方向の設定を固定して (双方向モードは BDIO ビットが変化しない限り有効)、別の半二重通信に置き換えることができます。**

### 30.5.3 標準マルチスレーブ通信

2 つ以上の独立したスレーブがある設定の場合、マスタは GPIO ピンを使用して、各スレーブのチップセレクトラインを管理します ([図 352](#)を参照)。マスタは、スレーブの NSS 入力に接続されている GPIO をローレベルにプルダウンすることによって、スレーブの 1 つを選択する必要があります。これを行うことにより、標準マスタと専用スレーブの通信が確立します。

図 352. マスタと 3 つの独立したスレーブ



1. NSS ピンはこの設定でのマスタ側では使用しません。MODF エラーを防ぐために内部的に管理 (SSM=1、SSI=1) する必要があります。
2. スレーブの MISO ピンは相互接続されているので、すべてのスレーブにおいて、その MISO ピンの GPIO 設定をオルタネート機能オープンドレインとしてセットする必要があります ( [セクション 11.3.7 : I/O オルタネート機能の入力/出力 \(231 ページ\)](#) を参照)。

## 30.5.4 マルチマスタ通信

SPI バスはマルチマスタ機能に主に対応して設計されていない限り、同時にバスをマスタしようとする 2 ノード間での潜在的な競合を検出する搭載機能を使用できます。この検出では、NSS ピンをハードウェア入力モードで設定して使用します。

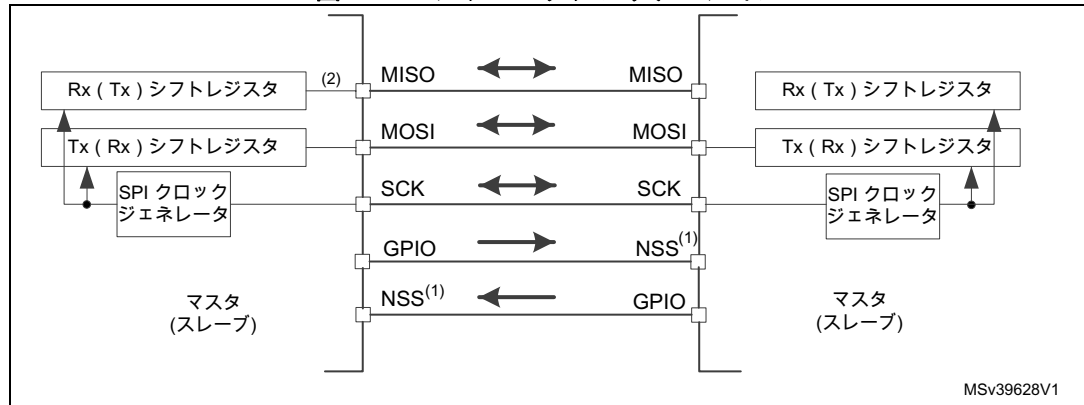
このモードで動作している 2 つ以上の SPI ノードを接続することは、1 度に共通データラインで出力を適用できるノードは 1 つだけであるため不可能です。

ノードが非アクティブの場合、両方のノードはデフォルトでスレーブモードのままとなります。1 つのノードがバスの制御を担うと、マスタモードに切り替わり、専用の GPIO ピンを通じてもう一方のノードのスレーブ選択入力のアクティブレベルを適用します。セッションが完了すると、アクティブ

スレーブ選択信号が解放され、バスを一時的にマスタするノードがパッシブスレーブモードに戻り、次のセッション開始を待機します。

同時に両方のノードのマスタリングリクエストが発生する可能性がある場合、バス競合イベントが発生します（モードフォールト MODF イベントを参照）。その場合、いくつかの簡単なアービトレーションプロセスを適用できます（たとえば、両方のノードに適用する事前定義した個別タイムアウトで次の試行を延長します）。

図 353. マルチマスタアプリケーション



1. NSS ピンは両方のノードのハードウェア入力モードで設定されます。パッシブノードがスレーブとして設定されるため、そのアクティブレベルによって MISO ライン出力制御が可能になります。

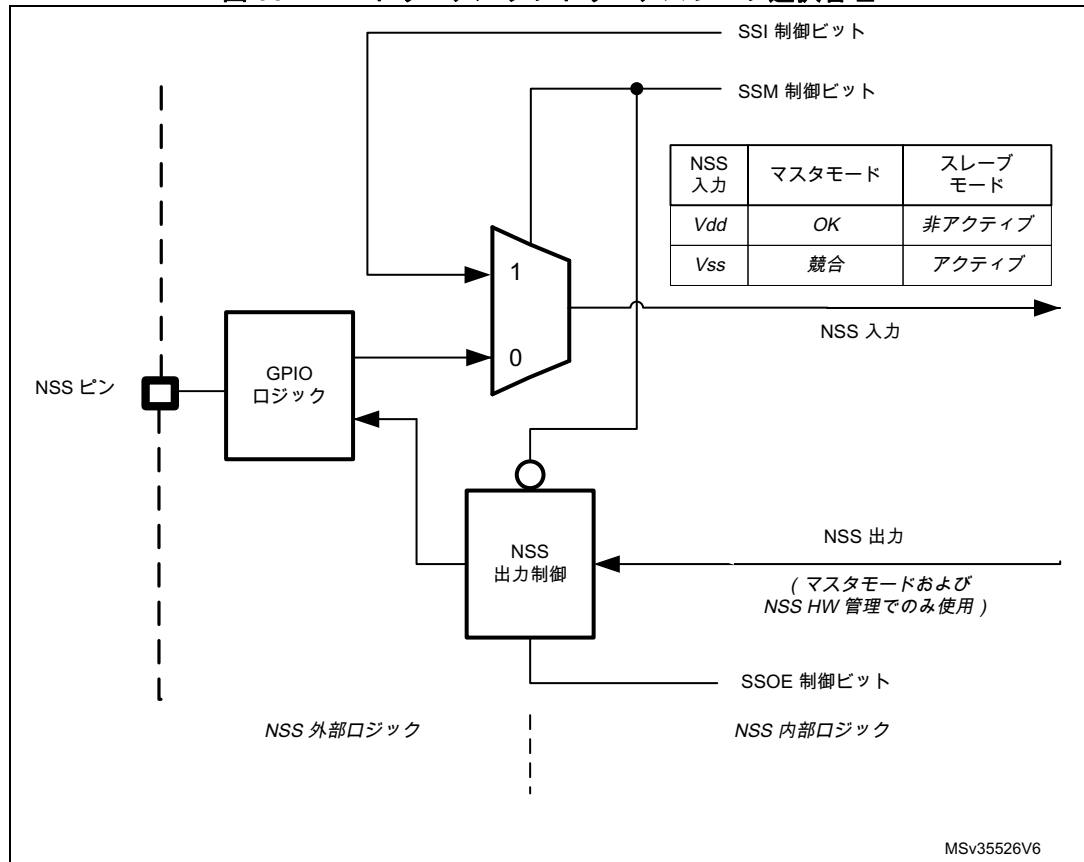
## 30.5.5 スレーブ選択 (NSS) ピンの管理

スレーブモードでは、NSS は標準の「チップセレクト」入力として機能し、スレーブをマスタと通信させます。マスタモードでは、NSS は出力としても入力としても使用できます。入力として使用する場合は、NSS はマルチマスタのバスの衝突を未然に防ぎ、出力として使用する場合は 1 つのスレーブのスレーブ選択信号を駆動させることができます。

ハードウェアまたはソフトウェアのスレーブ選択管理は、SPIx\_CR1 レジスタの SSM ビットを使用して、以下のようにセットすることができます。

- **ソフトウェア NSS 管理 (SSM = 1)** : この設定では、スレーブ選択情報は SPIx\_CR1 レジスタの SSI ビットの値によって内部で駆動されます。外部 NSS ピンは他のアプリケーションで使用できます。
- **ハードウェア NSS 管理 (SSM = 0)** : この場合、2 通りの設定が可能です。次のどちらの設定を使用するかは、NSS 出力設定 (SPIx\_CR1 レジスタの SSOE ビット) によって決まります。
  - **NSS 出力が有効な場合 (SSM = 0, SSOE = 1)** : この設定は、MCU がマスタとしてセットされている場合にのみ使用します。NSS ピンはハードウェアによって管理されます。NSS 信号は、SPI がマスタモードで有効になる (SPE = 1) とすぐにローレベルに駆動され、SPI が無効化される (SPE = 0) までローレベルに保たれます。パルスは、NSS パルスモードが有効になると連続通信間で生成できます (NSSP = 1)。この NSS 設定では、SPI はマルチマスタ設定で機能させることはできません。
  - **NSS 出力が無効な場合 (SSM = 0, SSOE = 0)** : マイクロコントローラがバスでマスタとして機能している場合、この設定によりマルチマスタ機能が可能になります。このモードで、NSS ピンがローレベルにプルダウンされた場合、SPI はマスタモードのフォールト状態に入り、デバイスは自動的にスレーブモードに再設定されます。スレーブモードでは、NSS ピンは標準の「チップセレクト」入力として機能し、NSS ラインがローレベルの間はスレーブが選択されます。

図 354. ハードウェア/ソフトウェアスレープ選択管理



## 30.5.6 通信フォーマット

SPI 通信中は受信と送信の操作が同時に行われます。シリアルクロック (SCK) は、データライン上で行われる情報のシフトとサンプリングを同期させます。通信フォーマットは、クロック位相、クロック極性、およびデータフレームフォーマットに応じて決定されます。マスターデバイスとスレーブデバイスの通信を可能にするには、双方が同じ通信フォーマットに従う必要があります。

### クロックの位相および極性の制御

SPIx\_CR1 レジスタの CPOL ビットと CPHA ビットを使用することによって、考えられる4つのタイミングの関係をソフトウェアで選択できます。CPOL (クロック極性) ビットは、データが転送されていないときのクロックのアイドル状態の値を制御します。このビットは、マスターモードとスレーブモードの両方に影響を与えます。CPOL がリセットされると、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされると、SCK ピンはハイレベルのアイドル状態になります。

CPHA ビットがセットされると、SCK ピンの2番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがリセットされていれば立ち下がりエッジ、CPOL ビットがセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。CPHA ビットがリセットされている場合、SCK ピンの1番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがセットされていれば立ち下がりエッジ、CPOL ビットがリセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。

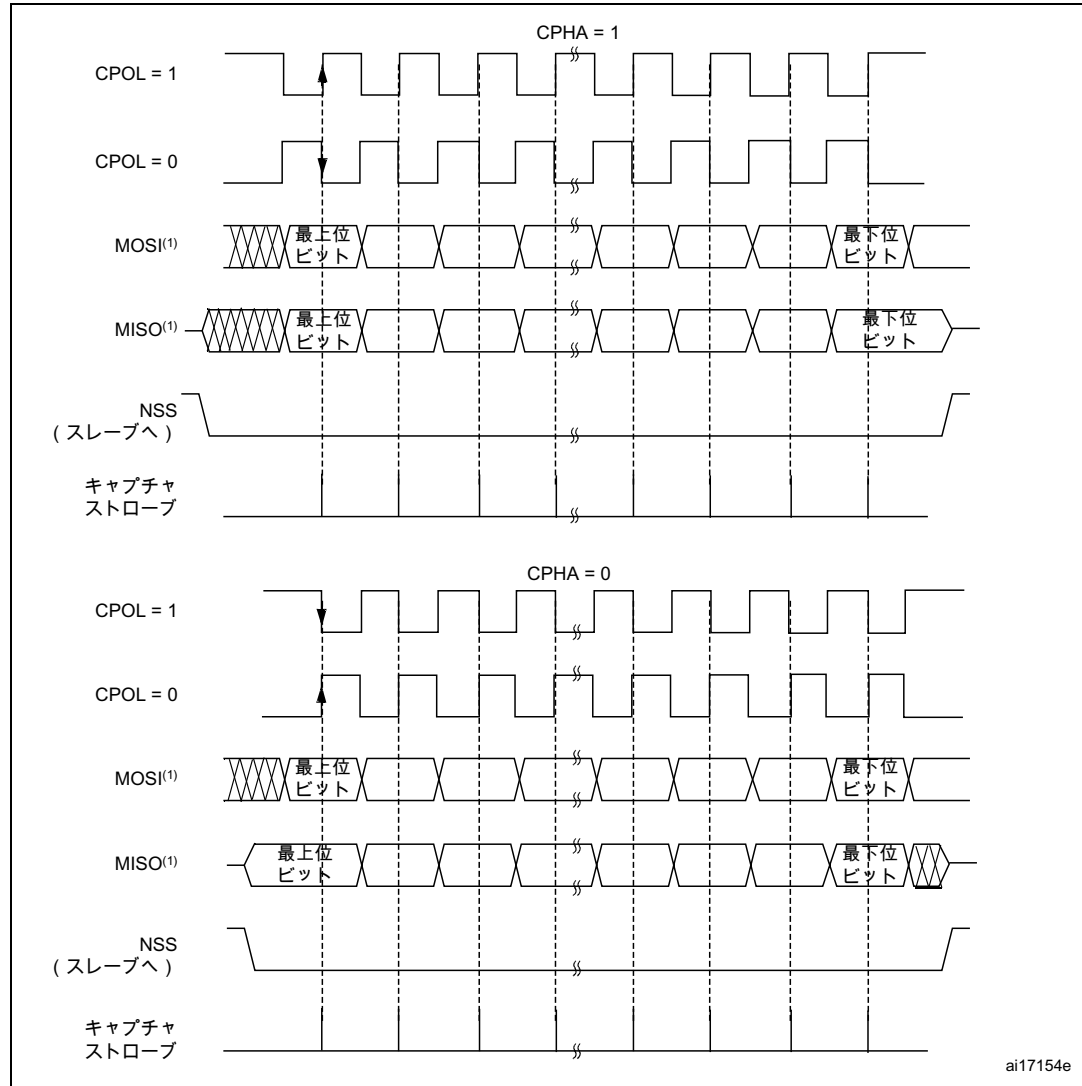
CPOL (クロック極性) ビットと CPHA (クロック位相) ビットの組み合わせによって、データキャプチャのクロックエッジを選択できます。

図 355は、CPHA ビットと CPOL ビットの 4 つの組み合わせによる SPI 全二重転送を示しています。

注： CPOL または CPHA ビットを変更する前に、SPE ビットをリセットすることによって、SPI を無効にする必要があります。

SCK のアイドル状態は、SPIx\_CR1 レジスタで (CPOL = 1 なら SCK のプルアップ、CPOL = 0 なら SCK のプルダウンによって) 選択された極性に一致する必要があります。

図 355. データクロックのタイミング図

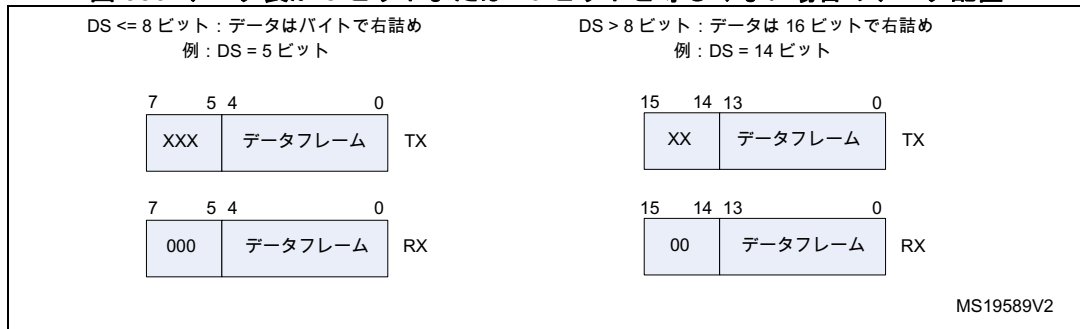


1. データビットの順序は LSBFIRST ビットの設定値に依存します。

## データフレームフォーマット

LSBFIRST ビットの値に応じて、SPI シフトレジスタを設定することで、シフトをMSB ファーストまたは LSB ファーストに設定することができます。データフレームサイズは、DS ビットを使用して選択できます。4 ビットから 16 ビット長までの間にセットでき、この設定は送信と受信の両方に適用できます。選択されたデータフレームサイズにかかわらず、FIFO への読み出しアクセスは FRXTH レベルに整列されている必要があります。SPIx\_DR レジスタにアクセスすると、データフレームはバイト (データがバイトに適合する場合) またはハーフワード (図 356 を参照) のいずれかに常に右詰めされます。通信中、データフレーム内のビットのみがクロック供給され、送信されます。

図 356. データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置



**注 :** 最小データ長は 4 ビットです。4 ビット未満のデータ長が選択された場合は、8 ビットのデータフレームサイズに固定されます。

## 30.5.7 SPI の設定

設定手順は、マスタとスレーブではほぼ同じです。特定のモードの設定については、それぞれのモードに関するセクションを参照してください。標準通信を初期化する必要があるときは、以下の手順を実行します。

1. 適切な GPIO レジスタに書き込みを行います。MOSI ピン、MISO ピン、SCK ピンの GPIO 設定を行います。
2. SPI\_CR1 レジスタに書き込みを行います。
  - a) BR[2:0] ビットを使用して、シリアルクロックボーレートを設定します (注 : 4)。
  - b) CPOL ビットと CPHA ビットの組み合わせを設定して、データ転送とシリアルクロックの 4 つの関係のうちの一つを定義します (CPHA は NSSP モードでクリアする必要があります)。(注 : 2)。
  - c) RXONLY または BIDIMODE、および BIDIOE を設定することによって (RXONLY と BIDIMODE は同時にセットできません)、単方向または半二重モードを選択します。
  - d) LSBFIRST ビットを設定して、フレームフォーマットを定義します (注 : 2)。
  - e) CRC が必要な場合は (SCK クロック信号がアイドル状態のとき)、CRCL と CRCEN ビットを設定します。
  - f) SSM と SSI を設定します (注 : 2 および 3)。
  - g) MSTR ビットを設定します (マルチマスタ NSS 設定では、MODF エラーを防ぐためにマスタが設定されている場合、NSS での競合を避けること)。
3. 以下のように、SPI\_CR2 レジスタに書き込みを行います。
  - a) 転送のデータ長を選択するために DS[3:0] ビットを選択します。
  - b) SSOE を設定します (注 : 1、2、3)。
  - c) TI プロトコルが必要な場合は FRF ビットをセットします (TI モードで NSSP ビットをクリアされたままにする)。
  - d) 2 つのデータユニット間で NSS パルスモードが必要な場合は NSSP ビットをセットします (NSSP モードで CHPA と TI ビットをクリアされたままにする)。
  - e) FRXTH ビットを設定します。RXFIFO 閾値は、SPIx\_DR レジスタの読み出しアクセスサイズに揃える必要があります。
  - f) DMA がパックモードで使用される場合は、LDMA\_TX および LDMA\_RX ビットを初期化します。
4. SPI\_CRCPR レジスタに書き込みを行います。必要に応じて CRC 多項式を設定します。
5. 適切な DMA レジスタに書き込みを行います。DMA ストリームが使用されている場合は、DMA



レジスタに SPI Tx および Rx 専用の DMA ストリームを設定します。

- 注：
- (1) このステップはスレーブモードでは必要ありません。
  - (2) このステップは TI モードでは必要ありません。
  - (3) このステップは NSSP モードでは必要ありません。
  - (4) このステップは、スレーブモードにて TI モードで動作している場合を除き、スレーブモードでは必要ありません。

## 30.5.8 SPI を有効にする手順

マスタがクロックを送信する前に、SPI スレーブを有効にすることを推奨します。さもなければ、望ましくないデータ送信が発生することがあります。スレーブのデータレジスタは、マスタとの通信を開始する前に、送信データをすでに格納していなければなりません（通信クロックの 1 番目のエッジに、またはクロック信号が連続的なきは現在の通信の最後の前に）。SPI スレーブが有効になる前に、SCK 信号を選択された極性に対応するアイドル状態のレベルに安定させる必要があります。

全二重（または送信専用モード）のマスタは、SPI が有効で TXFIFO がエンプティでない場合、または次の TXFIFO への書き込み時に通信を開始します。

あらゆるマスタ受信専用モードにおいて（RXONLY = 1 または BIDIMODE = 1、および BIDIOE = 0）、SPI が有効になるとすぐに、マスタは通信を開始し、クロックは動作を開始します。

DMA を処理するには、該当するセクションを参照してください。

## 30.5.9 データの送受信手順

### RXFIFO および TXFIFO

すべての SPI データトランザクションは 32 ビット埋め込み FIFO を通過します。これにより、SPI は連続フローで動作できるようになり、またデータフレームサイズが短い場合にオーバーランを防ぐことができます。各方向にはそれぞれ TXFIFO と RXFIFO と呼ばれる固有の FIFO があります。これらの FIFO は、CRC 計算を有効にした状態で、受信専用モード（スレーブまたはマスタ）を除くすべての SPI モードで使用されます（[セクション 30.5.14 : CRC 計算](#)を参照）。

FIFO の扱いはデータ変換モード（二重、単方向）、データフレームフォーマット（フレーム内のビット数）、FIFO データレジスタで実行されるアクセスサイズ（8 ビットまたは 16 ビット）、および FIFO アクセス時にデータパッキングが使用されるかどうかに従います（[セクション 30.5.13 : TI モード](#)を参照）。

SPIx\_DR レジスタへの読み出しアクセスからは、RXFIFO に保管された、まだ読み出されていない一番古い値が返されます。SPIx\_DR への書き込みアクセスでは、送信キューの最後に TXFIFO に書き込まれるデータを保管します。読み出しアクセスは、必ず SPIx\_CR2 レジスタの FRXTH ビットによって設定された RXFIFO 閾値に揃える必要があります。FTLV[1:0] および FRLVL[1:0] ビットは、両方の FIFO について現在の占有レベルを示します。

SPIx\_DR レジスタへの読み出しアクセスは RXNE イベントで管理する必要があります。このイベントは、データが RXFIFO に保管され、閾値（FRXTH ビットで定義される）に達した場合にトリガされます。RXNE がクリアされると、RXFIFO はエンプティであるとみなされます。同じように、送信するデータフレームの書き込みアクセスは TXE イベントで管理されます。このイベントは、TXFIFO レベルが容量の半分以下である場合にトリガされます。そうでない場合、TXE はクリアされ、TXFIFO がフルであるとみなされます。このように、データフレームフォーマットが 8 ビットを超えない場合、TXFIFO では最大 3 つのデータフレームのみを保管できるのに対し、RXFIFO では最大 4 つを保管することができます。この差異は、ソフトウェアが 16 ビットモードで TXFIFO により多くのデータの

書き込みを試みた場合に、すでに TXFIFO に保管されている 3x 8 ビットデータフレームが破損する可能性を防ぎます。TXE イベントと RXNE イベントの両方を割り込みによってポーリングまたは処理できます。図 358 から 図 361 までを参照してください。

データ交換を管理するもう一つの方法は、DMA を使用することです（[セクション 13.2 : DMA の主な機能](#)を参照）。

RXFIFO がフルのときに次のデータを受信した場合、オーバーランイベントが発生します（[セクション 30.5.10 : SPI ステータスフラグの OVR フラグの説明](#)を参照）。オーバーランイベントは割り込みによってポーリングまたは処理できます。

セットされる BSY ビットは、現在のデータフレームの進行中のトランザクションを示します。クロック信号が流れ続けているときは、BSY フラグはマスタ側のデータフレーム間でセットされたままになりますが、スレーブの各データフレーム転送間の 1 つの SPI クロックの最小時間においてローレベルになります。

## シーケンス処理

いくつかのデータフレームを単一シーケンスに渡してメッセージを完成させることができます。送信が有効な場合、シーケンスはマスタ側の TXFIFO に何らかのデータが存在する場合に開始し、その間続行します。TXFIFO がエンプティになるまで、マスタによってクロック信号が供給され続け、その後追加のデータを待つことを停止します。

半二重 (BIDIMODE=1、BIDIOE=0) または単方向 (BIDIMODE=0、RXONLY=1) の受信専用モードでは、SPI が有効化され、受信専用モードがアクティブ化されると、直ちにマスタによってシーケンスが開始されます。マスタによってクロック信号が供給されますが、この信号はマスタが SPI または受信専用モードを無効にするまで停止しません。マスタは、クロック信号が停止するまでデータフレームを受信し続けます。

マスタはあらゆるトランザクションを連続モードで供給できる (SCK 信号は連続的) 一方で、データフローおよびその内容をいつでも処理できるスレーブ機能を優先する必要があります。必要に応じて、マスタは通信速度を下げ、より低速のクロックか、または十分な遅延を含む個別のフレーム/データセッションを供給する必要があります。次の 2 点に注意してください。SPI モードのマスタまたはスレーブに対するアンダーフローエラー信号はありません。また、スレーブからのデータは、たとえスレーブがそれらのデータを時間内に適切に準備できない場合でも、常にマスタによってトランザクション処理されます。スレーブが DMA を使用することが特に望ましいのは、データフレームが短く、バスが高速の場合です。

マルチスレーブシステムにおいて通信用のスレーブを 1 つだけ選択するには、各シーケンスを NSS パルス内に収める必要があります。単一のスレーブシステムでは NSS によってスレーブを制御する必要はありませんが、スレーブを各データ転送シーケンスの開始と同期させるために、ここにもパルスを供給することをお勧めします。NSS は、ソフトウェアとハードウェアの両方で管理できます（[セクション 30.5.5 : スレーブ選択 \(NSS\) ピンの管理](#)を参照）。

セットされた BSY ビットは、進行中のデータフレームトランザクションを示します。専用のフレームトランザクションが終了すると、RXNE フラグが立てられます。最後のビットは単にサンプリングされ、すべてのデータフレームが RXFIFO に保管されます。

## SPI を無効にする手順

SPI を無効にする場合は、本段落に記載されている無効化手順に従ってください。この手順は、ペリフェラルクロックが停止し、システムが低電力モードに入る前に行うことが重要です。この場合、進行中のトランザクションが破壊されることがあります。モードによっては、この無効化手順が連続通信を停止させる唯一の方法です。

全二重または送信専用モードでは、マスタは、転送するデータの供給を停止した時点でいかなるトランザクションも終了することができます。この場合、クロックは最後のデータトランザクション後に

停止します。奇数のデータフレームがダミーバイトの変換を防ぐためにトランザクション処理された場合、パッキングモードでは特に注意する必要があります (データパッキングセクションを参照)。これらのモードで SPI を無効化する前に、ユーザは標準的な無効化手順に従う必要があります。フレームトランザクションの進行中または次のデータフレームを TXFIFO に保管しているときにマスタトランスミッタで SPI を無効化した場合、SPI の動作は保証されません。

マスタがいずれかの受信専用モードに設定されている場合、連続クロックを停止する唯一の方法は SPE=0 にしてペリフェラルを無効にすることです。これは、最初のビットのサンプリング時間の間、および最後のビット転送が開始される前における最後のデータフレームトランザクション内の特定の時間枠内で発生する必要があります (予測されるデータフレームの総数を受信し、最後の有効データフレーム後に追加の「ダミー」データの読み出しを防ぐため)。このモードで SPI を無効にするには、特定の手順に従う必要があります。

SPI が無効の場合、受信済みで読み出されていないデータは RXFIFO に保管されたままになり、次に SPI を有効にする際に、新しいシーケンスを開始する前に処理する必要があります。未読のデータを防ぐには、SPI を無効にする際に RXFIFO がエンプティであることを確認してください。これは、正しい無効化手順を使用して、またはペリフェラルリセット専用の固有レジスタを制御してソフトウェアリセットを行い、すべての SPI レジスタを初期化することで確認できます (RCC\_APB1RSTR レジスタの SPIIRST ビットを参照)。

標準的な無効化手順は、送信セッションが完全に終わったかどうかをチェックするために、FTLV[1:0] および BSY フラグの状態をプルして行われます。このチェックは、たとえば以下に示すように、進行中のトランザクションの終わりを識別する必要があるような特別な場合にも行うことができます。

- NSS 信号がソフトウェアによって管理されており、マスタはスレーブに適切な NSS パルスの終わりを提供する必要がある場合
- 最後のデータフレームまたは CRC フレームのトランザクションがまだペリフェラルバスで進行している間に、DMA または FIFO からのトランザクションのストリームが完了した場合

正しい無効化手順を以下に示します (受信専用モードが使用されている場合を除く)。

1. FTLVL[1:0] = 00 (送信するデータがなくなる) まで待ちます。
2. BSY = 0 (最後のデータフレームが処理される) まで待ちます。
3. SPI を無効にします (SPE = 0)。
4. FRLVL[1:0] = 00 (受信したデータをすべて読み出す) までデータを読み出します。

受信専用モードの場合の正しい無効化手順を以下に示します。

1. 最後のデータフレームの進行中に特定の時間枠内で SPI を無効 (SPE = 0) にすることにより、受信フローへの割り込みを行います。
2. BSY = 0 (最後のデータフレームが処理される) まで待ちます。
3. FRLVL[1:0] = 00 (受信したデータをすべて読み出す) までデータを読み出します。

**注:** *パッキングモードが使用され、8 ビット以下 (1 バイトに適合) のフォーマットの奇数のデータフレームを受信する必要がある場合は、FRLVL[1:0] = 01 のときに FRXTH をセットしなければなりません。これにより、最後の奇数データフレームを読み出し、良好な FIFO ポインタの整列を維持するための RXNE イベントが生成されます。*

## データパッキング

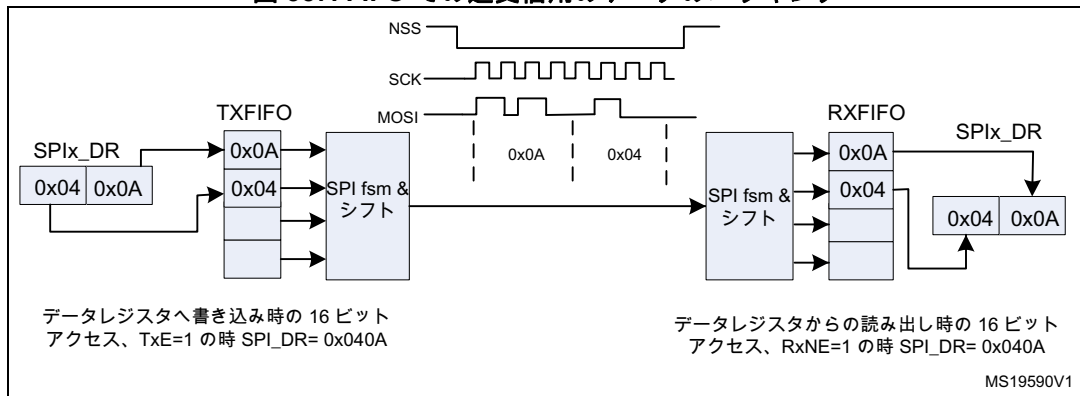
データフレームサイズが 1 バイト (8 ビット以下) に適合する場合、SPIx\_DR レジスタで 16 ビットの読み出しまたは書き込みアクセスが実行されたときに、自動的にデータパッキングが使用されます。この場合、二重データフレームパターンが並行して処理されます。最初に、SPI はアクセスしたワードの LSB に保管されたパターンを使用して動作します。次に、MSB に保管された残り半分を使用して動作します。図 357 に、データパッキングモードのシーケンス処理の例を示します。2 つのデータフレームは、トランスミッタの SPIx\_DR レジスタに単一の 16 ビットがアクセスした後で送信されます。このシーケンスでは、RXFIFO 閾値が 16 ビット (FRXTH=0) にセットされている場合に、



レシーバで 1 つの RXNE イベントのみを生成できます。この場合、この単一の RXNE イベントへのレスポンスとして、レシーバは SPIx\_DR の単一の 16 ビット読み出しにより、両方のデータフレームにアクセスする必要があります。RxFIFO 閾値設定および以下の読み出しアクセスは、常にレシーバ側に整理させる必要があります。整理されていない場合は、データを失う可能性があります。

これらの奇数の「1 バイトに適合」するデータフレームを処理する必要がある場合、特定の問題が発生します。トランスミッタ側では、SPIx\_DR への 8 ビットアクセスを持つ任意の奇数シーケンスの最後のデータフレームを書き込むだけで十分です。レシーバでは、RXNE イベントを生成するために、フレームの奇数シーケンスで受信した最後のデータフレームの Rx\_FIFO 閾値レベルを変更する必要があります。

図 357. FIFO での送受信のデータのパッキング



## DMA (ダイレクトメモリアクセス) を使用する通信

最高速度で動作し、オーバーランを回避するために必要なデータレジスタの読み出し/書き込み処理を容易にするために、SPI は簡単なリクエスト/確認応答プロトコルを実現する DMA 機能を備えています。

SPIx\_CR2 レジスタの TXE または RXNE イネーブルビットをセットすると、DMA アクセスがリクエストされます。TxバッファとRxバッファには、別々のリクエストを発行する必要があります。

- 送信では、TXE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx\_DR レジスタに書き込みます。
- 受信では、RXNE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx\_DR レジスタを読み出します。

図 358 から 図 361 までを参照してください。

SPI がデータの送信にのみ使用される場合、SPI Tx DMA チャンネルのみを有効にすることができます。この場合、受信したデータは読み出されないため、OVR フラグがセットされます。SPI がデータの受信にのみ使用される場合、SPI Rx DMA チャンネルのみを有効にすることができます。

送信モードで、DMA がすべての送信データを書き込んだとき (DMA\_ISR レジスタの TCIF フラグがセットされます)、BSY フラグを監視することで SPI 通信の完了を確認できます。最後の送信内容の破壊を避けるために、SPI を無効にする前、または STOP モードに入る前にこの操作を行う必要があります。ソフトウェアは、まず FTLVL[1:0] = 00 になるまで待つから、BSY=0 になるまで待つ必要があります。

DMA を使用して通信を開始する場合、DMA チャンネルの管理によるエラーイベントを防ぐために、以下の手順を順に実行する必要があります。

1. DMA Rx が使用される場合、SPI\_CR2 レジスタの RXDMAEN ビットの DMA Rx バッファを有効にします。
2. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを有効にします。
3. DMA Tx が使用される場合、SPI\_CR2 レジスタの TXDMAEN ビットにて DMA Tx バッファを有効にします。
4. SPE ビットをセットして、SPI を有効にします。

通信を終了するには、以下の手順を順に実行する必要があります。

1. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを無効にします。
2. SPI 無効化手順に従って SPI を無効にします。
3. DMA Tx および(または) DMA Rx が使用されている場合、SPI\_CR2 レジスタの TXDMAEN ビットおよび RXDMAEN ビットをクリアすることにより、DMA Tx バッファおよび Rx バッファを無効にします。

## DMA によるパッキング

DMA (SPIx\_CR2 レジスタでセットされた TXDMAEN および RXDMAEN) で転送が管理される場合、パッキングモードは SPI TX および SPI RX DMA チャンネルで設定される PSIZE 値によって、自動的に有効/無効になります。DMA チャンネルの PSIZE 値が 16 ビットと等しい場合、SPI データサイズは 8 ビット以下で、パッキングモードは有効になります。その後、DMA は SPIx\_DR レジスタへの書き込み動作を管理します。

データパッキングモードを使用する場合で転送データ数が 2 の倍数でない場合は、LDMA\_TX/LDMA\_RX ビットをセットする必要があります。SPI では、最後の DMA 転送の送受信に 1 つのデータのみを考慮します (詳細については、[データパッキング \(963 ページ\)](#) を参照)。

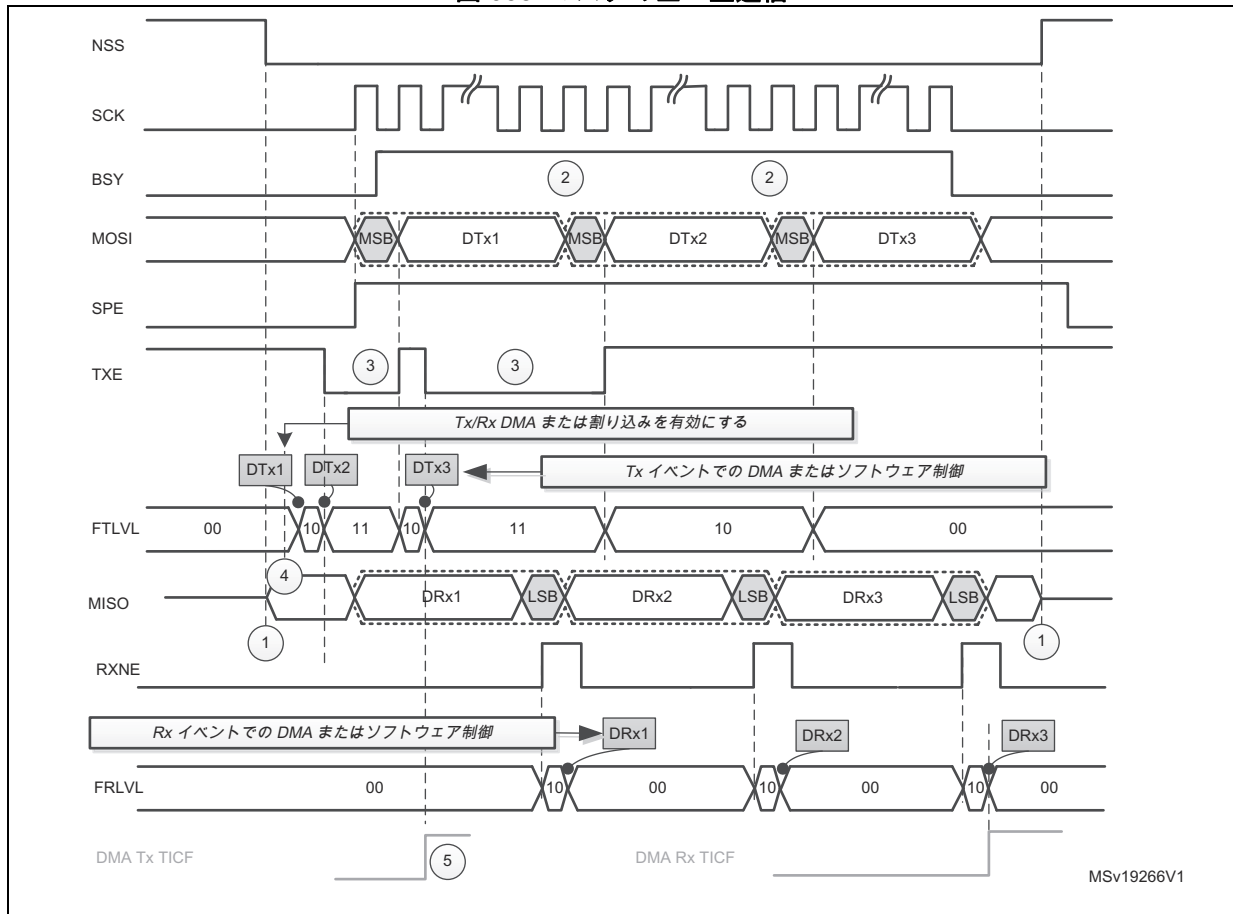
## 通信図

このセクションでは、一部の標準的なタイミング構成について説明します。これらの構成は、SPI イベントの処理方法（ポーリング、割り込み、または DMA）にかかわらず有効です。単純にするには、ここでは LSBFIRST=0、CPOL=0、および CPHA=1 設定を共通の前提として使用します。DMA ストリームの完全な設定は提供されません。

以下の番号付された注は、[図 358 \(967 ページ\)](#) から [図 361 \(970 ページ\)](#) に共通です。

- スレーブは、NSS がアクティブで SPI が有効になると MISO ラインを制御し始め、そのいずれかがリリースされるとラインから切断されます。事前にマスタ専用のデータを準備するには、トランザクションを開始する前にスレーブに十分な時間を与える必要があります。マスタでは、SPI が有効な場合のみ、SPI ペリフェラルが MOSI および SCK 信号で（また、ときに NSS 信号でも）制御できるようになります。SPI が無効の場合、SPI ペリフェラルは GPIO ロジックから切断され、これによりこれらのラインのレベルは GPIO 設定にのみ依存します。
- マスタでは、通信（クロック信号）が連続的な場合に、BSY はフレーム間でアクティブなままとなります。スレーブでは、BSY 信号は必ずフレーム間で最低でも 1 クロックサイクルは無効になります。
- TXE 信号は TXFIFO がフルの場合にのみクリアされます。
- DMA アービトレーションプロセスは TXDMAEN ビットのセット後すぐに開始します。TXE 割り込みは TXEIE のセット後すぐに生成されます。TXE 信号がアクティブレベルにあると、TxFIFO へのデータ転送が開始され、TxFIFO がフルまたは DMA 転送が完了するまで続きます。
- 送信するデータをすべて TxFIFO に適合することができる場合、SPI バスでの通信が開始される前に DMA Tx TCIF フラグを立てることができます。このフラグは、常に SPI トランザクションが完了する前に立てられます。
- パッケージの CRC 値は、SPIx\_TxCRCR および SPIx\_RxCRCR レジスタで、フレームごとに連続的に計算されます。CRC 情報はすべてのデータパッケージが完了した後で処理されます。これは、自動的に DMA で（必ず Tx チャンネルを処理するフレーム数にセットすること）、またはソフトウェアで（必ず最後のデータフレーム処理中に CRCNEXT を処理すること）行われます。SPIx\_TxCRCR で計算された CRC 値がトランスミッタによって簡単に送信されるのに対し、受信した CRC 情報は RxFIFO に読み込まれ、SPIx\_RxCRCR レジスタの内容と比較されます（差異がある場合は、ここで CRC エラーフラグが立てられます）。このため、ユーザは注意してこの情報を FIFO から一掃する必要があります。これには、RxFIFO のすべての保管された内容を読み出すソフトウェアか、Rx チャンネルの適切なデータフレーム数がプリセットされている場合は DMA を使用します（データフレーム数 + CRC フレーム数）（前提条件の例の設定を参照）。
- データがパックされたモードで、TxE および RxNE イベントはペアにされ、データフレーム数が偶数になるまで FIFO への各読み出し／書き込みアクセスは 16 ビット幅となります。TxFIFO が  $\frac{3}{4}$  フルの場合、FTLVL ステータスは FIFO フルレベルのままです。このため、最後の奇数のデータフレームを TxFIFO が  $\frac{1}{2}$  フルになる前に格納することはできません。このフレームは 8 ビットアクセスを持つ TxFIFO に格納されます。これは、ソフトウェアで、または LDMA\_TX 制御がセットされている場合は自動的に DMA で行われます。
- パックされたモードで最後の奇数のデータフレームを受信するには、最後のデータフレームが処理された時に Rx 閾値を 8 ビットに変更する必要があります。これは、ソフトウェアで FRXTH=1 に設定するか、LDMA\_RX がセットされている場合は自動的に DMA 内部信号で行われます。

図 358. マスタの全二重通信



マスタの全二重通信の前提条件の例：

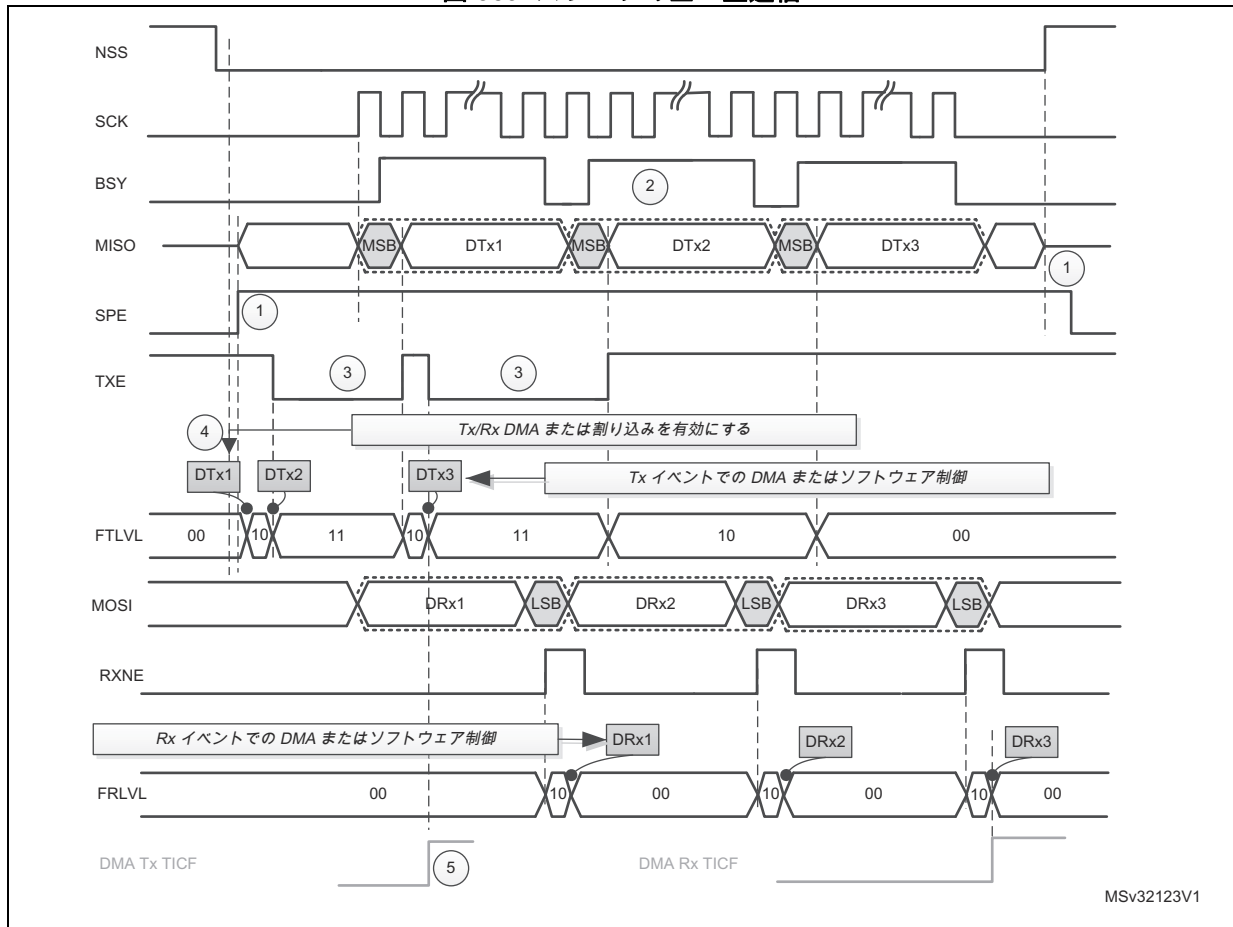
- データサイズ > 8 ビット

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(966 ページ\)](#) を参照してください。

図 359. スレープの全二重通信



スレープの全二重通信の前提条件の例：

- データサイズ > 8 ビット

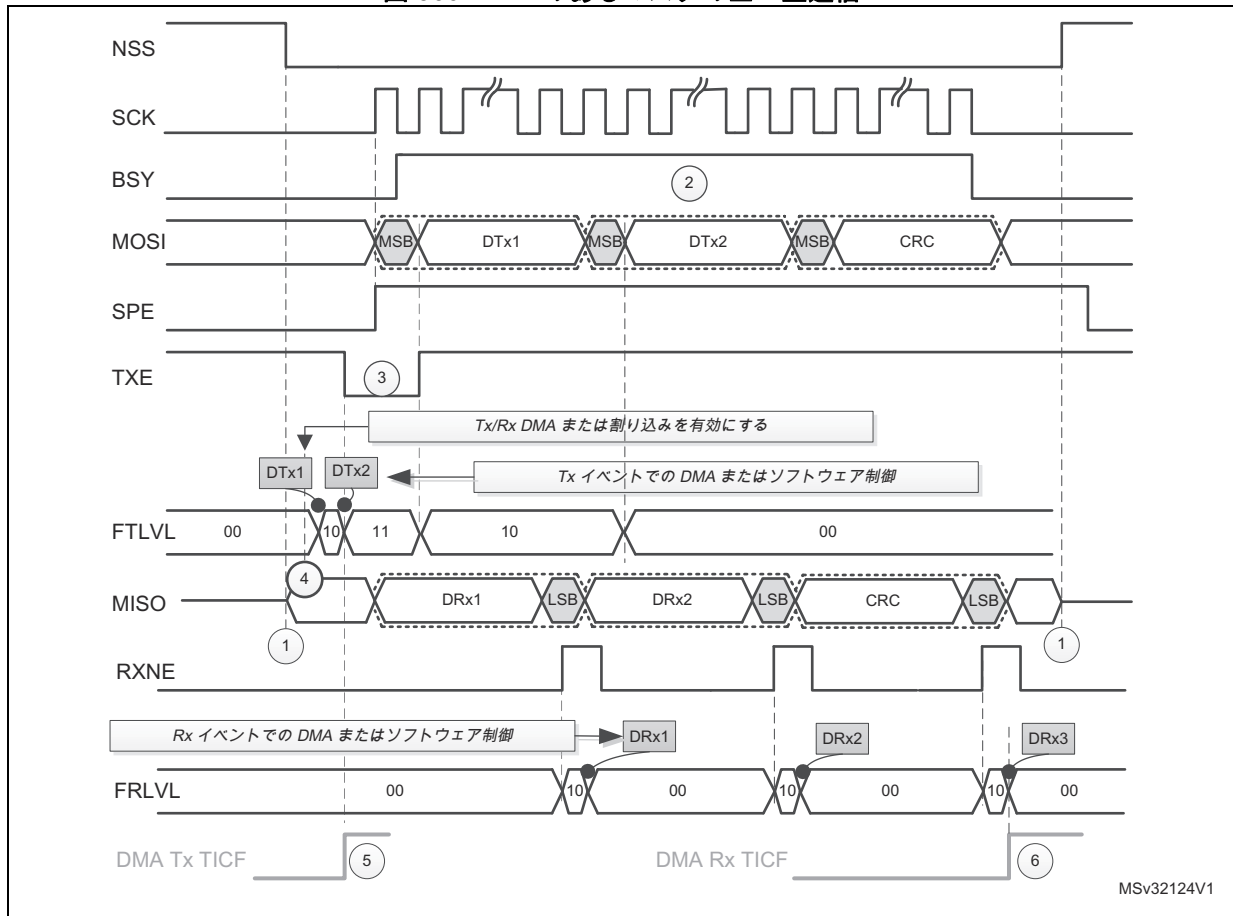
DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(966 ページ\)](#) を参照してください。



図 360. CRC のあるマスタの全二重通信



CRC のあるマスタの全二重通信の前提条件の例 :

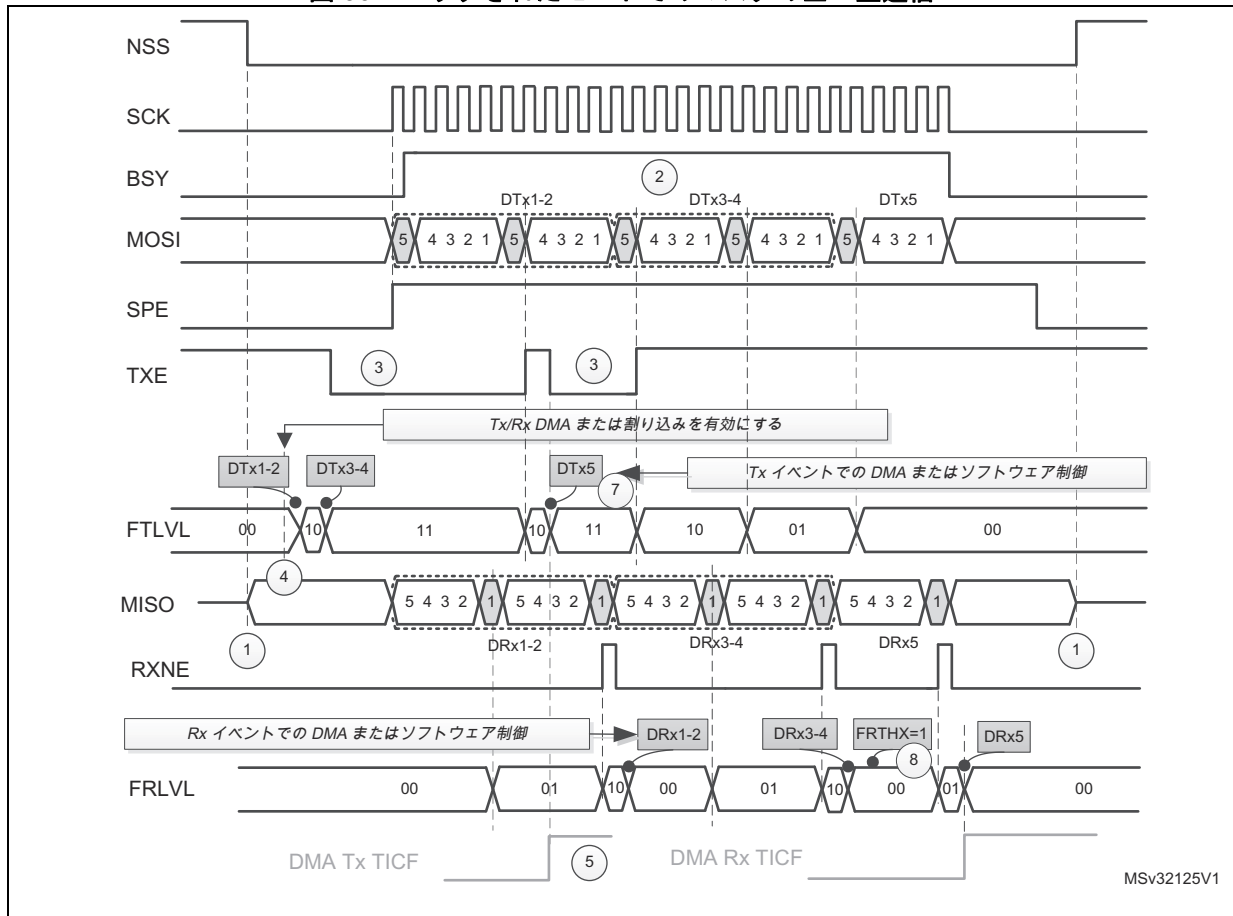
- データサイズ = 16 ビット
- CRC は有効

DMA を使用する場合 :

- DMA でトランザクションされる Tx フレーム数を 2 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(966 ページ\)](#) を参照してください。

図 361. パックされたモードでのマスタの全二重通信



パックされたモードでのマスタの全二重通信の前提条件の例：

- データサイズ = 5 ビット
- 読み出し／書き込み FIFO は、ほとんど 16 ビットアクセスで実行されます。
- FRXTH=0

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット
- Tx と Rx DMA チャンネルの両方の PSIZE を 16 ビットにセット
- LDMA\_TX=1 および LDMA\_RX=1

共通の前提条件および注の詳細については、通信図 (966 ページ) を参照してください。

## 30.5.10 SPI ステータスフラグ

アプリケーションが SPI バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

### Tx バッファエンプティフラグ (TXE)

TXE フラグは、送信 TXFIFO に送信データを格納するための十分なスペースがある場合にセットされます。TXE フラグは TXFIFO レベルにリンクされます。フラグはハイになり、TXFIFO レベルが FIFO の深さの 1/2 以下になってもハイを維持します。SPIx\_CR2 レジスタの TXEIE ビットがセットされている場合は、割り込みを生成できます。このビットは、TXFIFO レベルが 1/2 を超えると自動的にクリアされます。

### Rx バッファノットエンプティ (RXNE)

SPIx\_CR2 レジスタの FRXTH ビット値によって、RXNE フラグがセットされます。

- FRXTH をセットすると RXNE はハイになり、RXFIFO レベルが 1/4 (8 ビット) 以上になってもハイを維持します。
- FRXTH をクリアすると RXNE はハイになり、RXFIFO レベルが 1/2 (16 ビット) 以上になってもハイを維持します。

SPIx\_CR2 レジスタの RXNEIE ビットがセットされている場合は、割り込みを生成できます。

RXNE は、上記の条件が真ではなくなった場合にハードウェアによって自動的にクリアされます。

### ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます (このフラグへの書き込みは無効)。

BSY フラグがセットされると、SPI 上でデータ転送が進行中であることを示します (SPI バスはビジー)。

BSY は、転送終了を検出するために特定のモードで使用できます。これにより、低電力モード (ペリフェラルにクロックを供給しない) に入る前に、ソフトウェアで SPI やそのペリフェラルクロックを無効にすることができます。これによって、最後の転送データの破壊を回避します。

BSY フラグは、マルチマスタシステムでの書き込み衝突の回避にも役立ちます。

BSY フラグは次のいずれかの条件下でクリアされます。

- SPI が正常に無効にされたとき
- マスタモードで、障害が検出 (MODF ビットが 1 にセットされます) されたとき
- マスタモードで、データ送信が終了し、送信準備ができていない新しいデータがないとき
- スレーブモードで、各データ転送間で少なくとも SPI の 1 クロックサイクルの間、BSY フラグが 0 にセットされているとき

**注：** 次の送信がすぐにマスタで処理される場合 (マスタが受信専用モードに設定されているか、その送信 FIFO がエンプティでない場合)、通信は連続的で、BSY フラグはマスタ側での送信の間、“1”にセットされたままになります。スレーブとは異なりますが、必ず (BSY フラグを使用する代わりに) TXE フラグと RXNE フラグを使用して、データの送受信の処理を行うことを推奨します。

## 30.5.11 SPI エラーフラグ

次のいずれかのエラーフラグがセットされていて、ERRIE ビットをセットすることにより割り込みが有効になっている場合、SPI 割り込みが生成されます。

### オーバーランフラグ (OVR)

マスタまたはスレーブでデータを受信し、RXFIFO に受信データを格納するための十分なスペースがない場合に、オーバーラン状態が発生します。これは、ソフトウェアまたは DMA が RXFIFO に格納された前の受信データを読み出すために必要な時間が十分になかったか、データストレージのための空間が制限されている場合に発生する可能性があります。たとえば、RXFIFO は CRC が受信専用モードでのみ有効な場合は使用できないため、この場合、受信バッファは単一データフレームバッファに制限されます ([セクション 30.5.14 : CRC 計算](#)を参照)。

オーバーラン条件が発生すると、RXFIFO にある前の値を新しく受信した値で上書きすることはできません。新しく受信した値は破棄され、それ以降に送信されたすべてのデータは失われます。OVR ビットをクリアするには、SPI\_DR レジスタを読み出し、続けて SPI\_SR レジスタを読み出しアクセスを行います。

### モードフォールト (MODF)

モードフォールトは、マスタデバイスが内部 NSS 信号 (NSS ハードウェアモードでは NSS ピン、NSS ソフトウェアモードでは SSI ビット) をローレベルにプルダウンしたときに発生します。これにより、MODF ビットが自動的にセットされます。マスタモードフォールトは、SPI インタフェースに次のような影響を与えます。

- MODF ビットがセットされ、ERRIE ビットがセットされている場合は SPI 割り込みが生成されます。
- SPE ビットがクリアされます。これによって、デバイスからのすべての出力がブロックされ、SPI インタフェースが無効になります。
- MSTR ビットがクリアされ、デバイスは強制的にスレーブモードになります。

MODF ビットをクリアするには、次のソフトウェアシーケンスを実行します。

1. MODF ビットがセットされている間、SPIx\_SR レジスタを読み出し/書き込みアクセスを行います。
2. 次に、SPIx\_CR1 レジスタに書き込みを行います。

複数の MCU で構成されるシステムでスレーブ間の競合を避けるには、MODF ビットをクリアするシーケンス中、NSS ピンをハイレベルにプルアップする必要があります。このクリアシーケンスの後、SPE ビットと MSTR ビットは、元の状態に戻すことができます。安全のため、MODF ビットがセットされている間、ハードウェアは SPE ビットと MSTR ビットのセットを許可しません。スレーブデバイスでは、MODF ビットはセットできません。ただし、前回のマルチマスタ競合の結果としてセットする場合は例外です。

### CRC エラー (CRCERR)

このフラグを使用して、SPIx\_CR1 レジスタの CRCEN ビットがセットされているときに受信された値の有効性を検証します。シフトレジスタに受信された値が、レシーバである SPIx\_RXCRCR の値と一致しなかった場合、SPIx\_SR レジスタの CRCERR フラグがセットされます。フラグはソフトウェアによってクリアされます。

### TI モードフレームフォーマットエラー (FRE)

SPI がスレーブモードで動作し、かつ TI モードプロトコルに準拠した設定となっている場合、通信の進行中に NSS パルスが発生すると、TI モードフレームフォーマットエラーが検出されます。このエラーが発生すると、SPIx\_SR レジスタの FRE フラグがセットされます。エラー発生時には SPI は無

効にされず、この NSS パルスは無視されます。SPI は次の NSS パルスを待ってから新規の転送を開始します。このエラーの検出により 2 バイトのデータが失われるため、データは破壊される可能性があります。

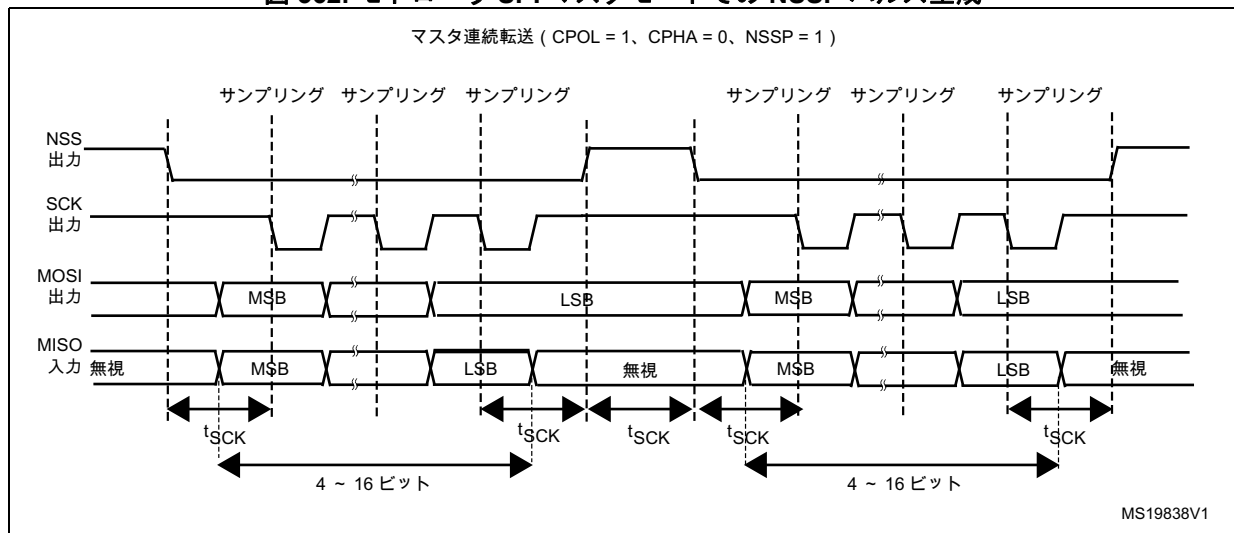
SPIx\_SR レジスタを読み出すと、FRE フラグがクリアされます。ERRIE ビットがセットされている場合、NSS エラー検出時に割り込みが生成されます。この場合、データの一貫性が保証されなくなるため、SPI を無効にする必要があり、またスレーブ SPI が再び有効化された場合は、マスタによって通信を再起動する必要があります。

### 30.5.12 NSS パルスモード

このモードは SPIx\_CR2 レジスタの NSSP ビットで有効化され、SPI インタフェースが 1 番目のエッジにキャプチャのある モトローラ SPI マスタ (FRF=0) として設定されている場合にのみ有効になります (SPIx\_CR1 CPHA = 0、CPOL 設定は無視されます)。有効にすると、最低 1 つのクロック周期の間、NSS がハイレベルを維持する場合に、2 つの連続したデータフレーム転送の間で NSS パルスが生成されます。このモードでは、スレーブでデータをラッチできます。NSSP パルスモードは、シングルマスタ - スレーブペアのアプリケーション用に設計されています。

図 362 に、NSSP パルスモードが有効な場合の NSS ピンの管理の図を示します。

図 362. モトローラ SPI マスタモードでの NSSP パルス生成



注： CPOL = 0 の場合に同様の動作が発生します。この場合、サンプリングエッジは SCK の立ち上がりエッジで、NSS のアサートおよびネゲートはこのサンプリングエッジを参照します。

### 30.5.13 TI モード

#### マスタモードでの TI プロトコル

SPI インタフェースは TI プロトコルと互換性があります。SPIx\_CR2 レジスタの FRF ビットを使って、SPI をこのプロトコルに準拠させるように設定することができます。

SPIx\_CR1 レジスタにセットされる値によらず、クロックの極性と位相は TI プロトコル要件に必ず適合します。NSS 管理も TI プロトコルに固有なものになります。これにより、この場合の SPIx\_CR1 レジスタと SPIx\_CR2 レジスタによる NSS 管理の設定 (SSM, SSI, SSOE) ができなくなります。

スレーブモードでは、SPI ボーレートプリスケラを使用して、現在のトランザクションが終了した時点で、MISO ピンの状態がハイインピーダンスに変化するタイミングを制御します (図 363 を参

照)。任意のボーレートが使用できるため、このタイミングを非常に柔軟に決定することができます。ただし、ボーレートは外部マスタクロックボーレートに設定されるのが一般的です。MISO 信号がハイインピーダンス ( $t_{\text{release}}$ ) になるまでの遅延は、内部再同期と SPIx\_CR1 レジスタの BR[2:0] ビットで設定されたボーレート値によって変わります。この値は次式で求められます：

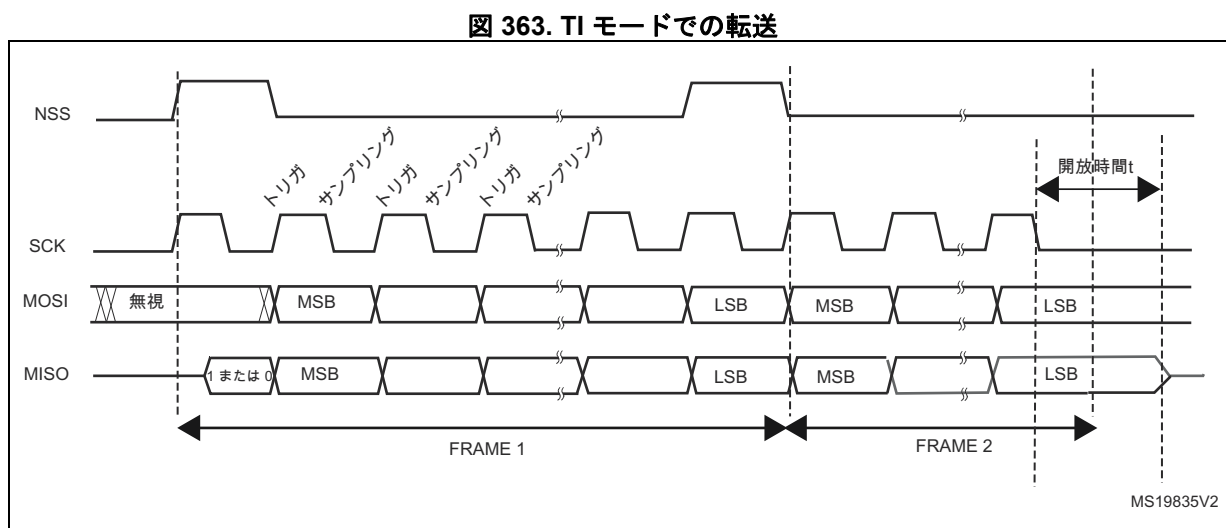
$$\frac{t_{\text{baud\_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud\_rate}}}{2} + 6 \times t_{\text{pclk}}$$

スレーブがデータフレームトランザクション中に NSS パルスの誤配置を検出すると、TIFRE フラグがセットされます。

データサイズが 4 ビットまたは 5 ビットである場合、全二重モードまたは送信専用モードのマスタは、最下位ビット後に追加された 1 つまたは複数のダミーデータビットを持つプロトコルを使用します。TI NSS パルスは、各周期の最下位ビットではなく、このダミービットクロックサイクル上に生成されます。

この機能はモトローラの SPI 通信には使用できません (FRF ビットを 0 に設定)。

**図 363 : TI モードでの転送** TI モードが選択されているときの SPI の通信波形を示します。



## 30.5.14 CRC 計算

2 つの CRC 計算機が、送信データおよび受信データの信頼性をチェックするために実装されています。SPI は、フレームデータ長とは別に、8 ビットまたは 16 ビットに固定される CRC8 または CRC16 を計算します。その他のデータフレーム長では、CRC は使用できません。

### CRC の原理

CRC 計算は、SPI が有効 (SPE = 1) になる前に、SPIx\_CR1 レジスタの CRCEN ビットをセットすることによって有効になります。CRC 値は、各ビットに対して奇数のプログラム可能な多項式の値を使用して計算されます。計算は、SPIx\_CR1 レジスタの CPHA ビットと CPOL ビットによって定義されるサンプリングクロックエッジで行われます。計算された CRC 値は、データブロックの最後のみならず、CPU または DMA によって管理される転送に関しても、自動的にチェックされます。受信データをもとに内部で計算された CRC 値とトランスミッタが送信した CRC 値の間に不一致が検出された場合、データ破壊エラーを示すために CRCERR フラグがセットされます。CRC 計算を処理する正しい手順は、SPI の設定および選択された転送管理によって変わります。

注： 多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。

## CPU によって管理される CRC 転送

通信が開始され、最後のデータフレームが SPIx\_DR レジスタで送信または受信されるまで正常に続きます。次に、CRC フレームトランザクションが現在処理中のデータフレームトランザクションの後に行われることを示すために、SPIx\_CR1 レジスタのCRCNEXT ビットをセットする必要があります。CRCNEXT ビットは、最後のデータフレームトランザクションの終了前にセットしてください。CRCのトランザクション中は CRC 計算は行われません。

受信した CRC は、データバイトまたはワードのように RXFIFO に格納されます。このため、CRC モードでは、受信バッファを一度に 1 データフレームずつ受信するために使用する単一の 16 ビットバッファとみなす必要があります。

CRC フォーマットトランザクションは、通常データシーケンスの最後に通信を行うためにデータフレームをもう 1 つ受け取ります。ただし、16 ビット CRC によってチェックされた 8 ビットデータフレームを設定する場合、CRC 全体を送信するにはフレームがあと 2 つ必要です。

最後の CRC データを受信すると、受信した値と SPIx\_RXCRC レジスタの値を比較する自動チェックが行われます。ソフトウェアは、SPIx\_SR レジスタの CRCERR フラグをチェックして、データ転送の内容が破壊されているか否かを判断する必要があります。ソフトウェアは、CRCERR フラグに“0”を書き込んでクリアします。

CRC 受信後、CRC 値は RXFIFO に格納され、RXNE フラグをクリアするために SPIx\_DR レジスタを読み出す必要があります。

## DMA によって管理される CRC 転送

SPI 通信が CRC 通信と DMA モードで有効化される場合、CRC の送受信は通信の最後に自動で行われます（ただし、受信専用モードで CRC データを読み出す場合を除く）。CRCNEXT ビットはソフトウェアで処理する必要はありません。SPI 送信用DMA チャンネルのカウンタは、転送するデータフレーム数からCRC フレームを除いてセットする必要があります。レシーバ側では、受信した CRC 値はトランザクションの終了時に DMA によって自動的に処理されますが、RXFIFO から受信した CRC 情報は常にレシーバにロードされますので、ユーザは注意してそれを一掃してください。全二重モードでは、受信 DMA チャンネルのカウンタを、受信するデータフレームの数（CRC を含む）にセットすることができます。つまり、たとえば 16 ビット CRC でチェックされた 8 ビットデータフレームの特定の例を、次の式で表すことができます。

$$\text{DMA\_RX} = \text{Numb\_of\_data} + 2$$

受信専用モードでは、DMA 受信チャンネルカウンタには転送されたデータ量のみが含まれます（CRC 計算を除く）。次に、DMA からの全体の転送に基づき、すべての CRC 値はこのモードでは単一のバッファとして動作するため、ソフトウェアによって FIFO から読み戻す必要があります。

転送中に内容の破壊が生じた場合、データと CRC の転送の最後に SPIx\_SR レジスタの CRCERR フラグがセットされます。

パッキングモードが使用され、データ数が奇数の場合は、LDMA\_RX ビットを管理する必要があります。

## SPIx\_TXCRC および SPIx\_RXCRC の値のリセット

CRC フェーズのあとで新しいデータがサンプリングされた場合、SPIx\_TXCRC 値および SPIx\_RXCRC 値は自動的にクリアされます。これにより、データを中断することなく転送するために、DMA サーキュラモードが使用可能になります（受信専用モードでは使用不可）。いくつかのデータブロックは中間の CRC チェックフェーズによってカバーされます。

通信中に SPI が無効化された場合は、次のシーケンスに従う必要があります。

1. SPI を無効にします。
2. CRCEN ビットをクリアします。
3. CRCEN ビットを有効にします。
4. SPI を有効にします。

**注：** SPI がスレープモードのとき、SPE ビットの値に関係なく、CRCEN がセットされるとすぐに、CRC 計算機は SCK スレープ入カクロックの影響を受けます。CRC 計算の間違いを回避するために、ソフトウェアは、クロックが安定している（定常状態にある）ときのみ、CRC 計算を有効にする必要があります。SPI インタフェースがスレープとして設定されている場合、データフェーズと CRC フェーズの間で NSS 内部信号をローレベルに維持する必要があります。

## 30.6 SPI 割り込み

SPI 通信中、割り込みは次のイベントによって生成できます。

- ロードする準備ができていない TXFIFO の送信
- 受信 RXFIFO で受信したデータ
- マスタモードフォールト
- オーバーランエラー
- TI フレームフォーマットエラー
- CRC プロトコルエラー

割り込みは個別に有効または無効にできます。

表 167. SPI 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
ロードする準備ができていない TXFIFO の送信	TXE	TXEIE
RXFIFO で受信したデータ	RXNE	RXNEIE
マスタモードフォールトイベント	MODF	ERRIE
オーバーランエラー	OVR	
TI フレームフォーマットエラー	FRE	
CRC プロトコルエラー	CRCERR	

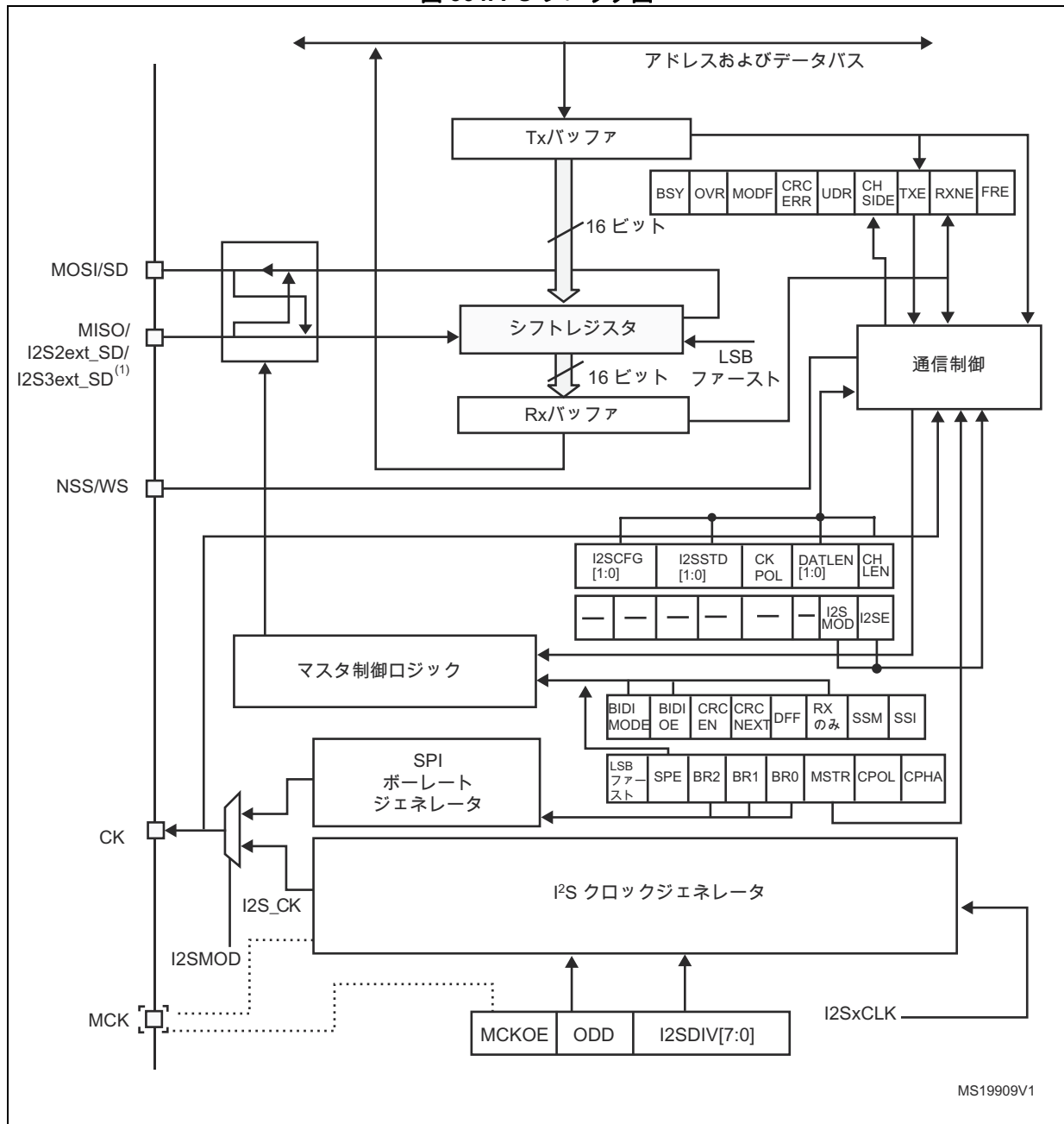


## 30.7 I<sup>2</sup>S の機能説明 (STM32F303xB/C/D/E、STM32F358xC および STM32F398xE のみ)

### 30.7.1 I<sup>2</sup>S の概要

I<sup>2</sup>S のブロック図を [図 364](#) に示します。

図 364. I<sup>2</sup>S ブロック図



1. I2S2ext\_SD と I2S3ext\_SD は、I2S 全二重 モードを制御する拡張 SD ピンです。

SPI はオーディオ I<sup>2</sup>S インタフェースとして機能することができます。それには、SPIx\_I2SCFGR レジスタの I2SMOD ビットをセットして I<sup>2</sup>S 機能を有効にしてください。このインタフェースは、主に SPI と同じピン、フラグ、および割り込みを使用します。

I<sup>2</sup>S と SPI は、以下の3つのピンを共有します。

- SD : MOSI ピンに配置され、2つの時間多重化データチャネルを送受信します (半二重モードのみ)。
- WS : NSS ピンに配置され、マスタモードではデータ制御信号の出力、スレーブモードでは入力です。
- CK : SCK ピンに配置され、マスタモードではシリアルクロック出力、スレーブモードではシリアルクロック入力です。
- I2S2ext\_SD および I2S3ext\_SD : I2S 全二重モードを制御する追加ピン (MISO ピンに配置)。

外部オーディオデバイスにマスタクロック出力が必要な場合、追加のピンを使用できます。

- MCK : 別々に配置され、I<sup>2</sup>S がマスタモードに設定されている (かつ、SPIx\_I2SPR レジスタの MCKOE ビットがセットされている) とき、 $256 \times f_s$  と等しい ( $f_s$  はオーディオサンプリング周波数) 設定済みの周波数で生成されたこの追加クロックを出力するために使用されます。

I<sup>2</sup>S は、マスタモードに設定されているとき、専用のクロックジェネレータを使用して通信クロックを生成します。このクロックジェネレータは、マスタクロック出力のソースでもあります。I<sup>2</sup>S モードでは、2つの追加レジスタを使用できます。1つはクロックジェネレータ設定 SPIx\_I2SPR にリンクされ、もう1つは汎用 I<sup>2</sup>S 設定レジスタ SPIx\_I2SCFGR (オーディオ規格、スレーブ/マスタモード、データフォーマット、パケットフレーム、クロック極性など) です。

SPIx\_CR1 レジスタとすべての CRC レジスタは、I<sup>2</sup>S モードでは使用されません。同様に、SPIx\_CR2 レジスタの SSOE ビットと SPIx\_SR レジスタの MODF および CRCERR ビットも使用されません。

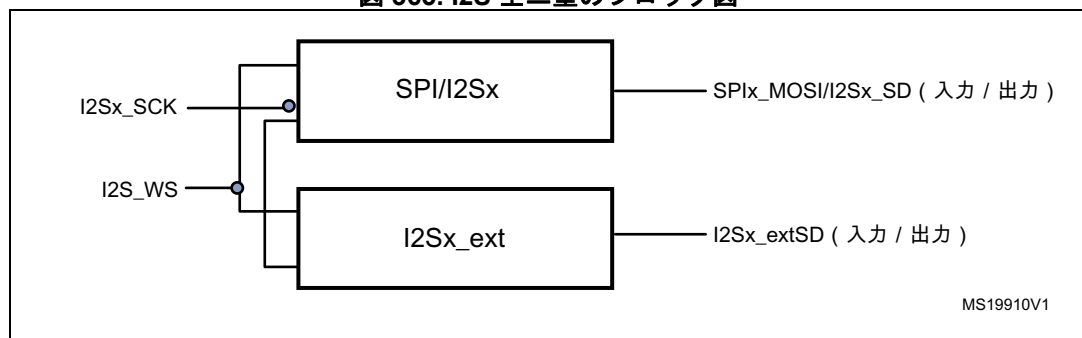
I<sup>2</sup>S は、16ビット幅モードでのデータ転送に同じ SPI レジスタ (SPIx\_DR) を使用します。

## 30.7.2 I2S 全二重

I2S 全二重モードをサポートするため、I2S2 と I2S3 の他に拡張 I2S (I2S2\_ext、I2S3\_ext) と呼ぶ2つの追加 I<sup>2</sup>S インスタンスが利用可能です (図 365 を参照)。そのため、最初の I2S 全二重インタフェースは I2S2 と I2S2\_ext に、2つ目の全二重インタフェースは I2S3 と I2S3\_ext に、それぞれ基づいています。

**注 :** I2S2\_ext と I2S3\_ext は全二重モードでのみ使用されます。

図 365. I2S 全二重のブロック図



1. x は 2 または 3 です。

I2Sx はマスタモードで動作することができます。その結果：

- 半二重モードでは、I2Sx だけが SCK と WS を出力することができます。
- 全二重モードでは、I2Sx だけが I2S2\_ext と I2S3\_ext への SCK と WS を供給することができます。

拡張 I2S (I2Sx\_ext) は、全二重モードでのみ使用することができます。I2Sx\_ext は、常にスレーブモードで動作します。

I2Sx と I2Sx\_ext はどちらも、トランスミッタまたはレシーバとして設定することができます。

### 30.7.3 サポートされるオーディオプロトコル

4 線バスでは、一般に 2 つのチャンネル (右チャンネルと左チャンネル) で時間多重化されたオーディオデータのみを処理する必要があります。しかしながら、送受信用には 1 つの 16 ビットレジスタしかありません。したがって、各チャンネルサイドに対応する適切な値をデータレジスタに書き込んだり、データレジスタからデータを読み出して SPIx\_SR レジスタの CHSIDE ビットをチェックして対応するチャンネルを識別したりすることは、ソフトウェアの責任です。左チャンネルは常に最初に送信され、その後で右チャンネルが送信されます (CHSIDE は PCM プロトコルには無関係です)。

4 つのデータおよびパケットフレームを使用できます。データは次のフォーマットで送信されます。

- 16 ビットフレームにパックされた 16 ビットデータ
- 16 ビットフレームにパックされた 32 ビットデータ
- 24 ビットフレームにパックされた 32 ビットデータ
- 32 ビットフレームにパックされた 32 ビットデータ

32 ビットパケットに拡張された 16 ビットデータを使用するとき、最初の 16 ビット (MSB) は上位ビットであり、LSB の 16 ビットは、ソフトウェア動作や DMA リクエストを必要とせずに (1 回の読み込み/書き込み動作のみで) 強制的に 0 にされます。

24 ビットと 32 ビットのデータフレームは、SPIx\_DR レジスタとの間で 2 回の CPU 読み出しまたは書き込み動作 (あるいはアプリケーションにとって DMA が望ましい場合は 2 回の DMA 動作) を必要とします。特に 24 ビットのデータフレームの場合、8 つの下位ビットはハードウェアによって 0 ビットで 32 ビットに拡張されます。

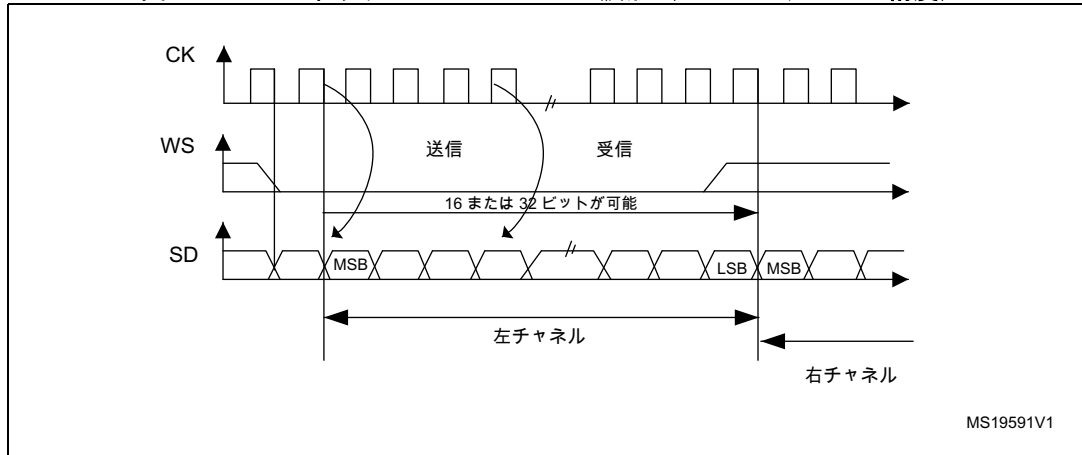
すべてのデータフォーマットと通信規格に対して、最上位ビットは常に最初に送信されます (MSB ファースト)。

I<sup>2</sup>S インタフェースは、SPIx\_I2SCFGR レジスタの I2SSTD[1:0] と PCMSYNC ビットを使用して設定可能な 4 つのオーディオ規格をサポートします。

## フィリップス I<sup>2</sup>S 規格

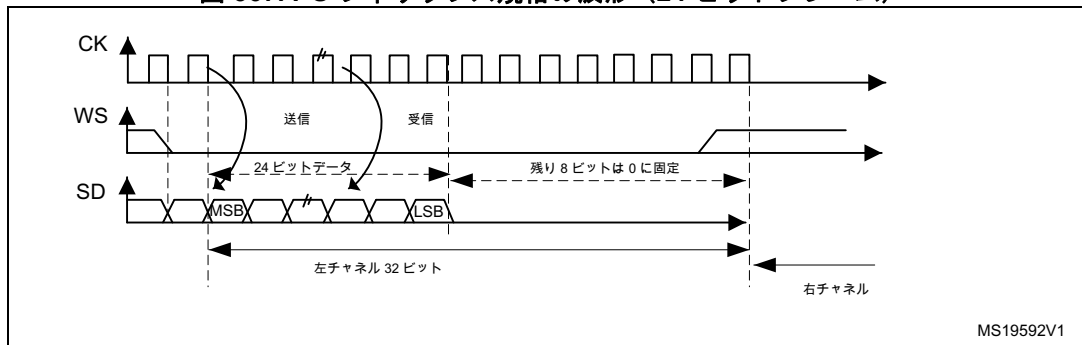
この規格では、どのチャンネルが送信されているかを示すために WS 信号を使用します。この信号が有効になってから 1 CK クロックサイクル後に最初のビット (MSB) が使用可能になります。

図 366. I<sup>2</sup>S フィリップスプロトコルの波形 (16/32 ビットフル精度)



データは、CK の立ち下がりエッジでラッチされ (トランスミッタの場合)、立ち上がりエッジで読み出されます (レシーバの場合)。WS 信号も CK の立ち下がりエッジでラッチされます。

図 367. I<sup>2</sup>S フィリップス規格の波形 (24 ビットフレーム)

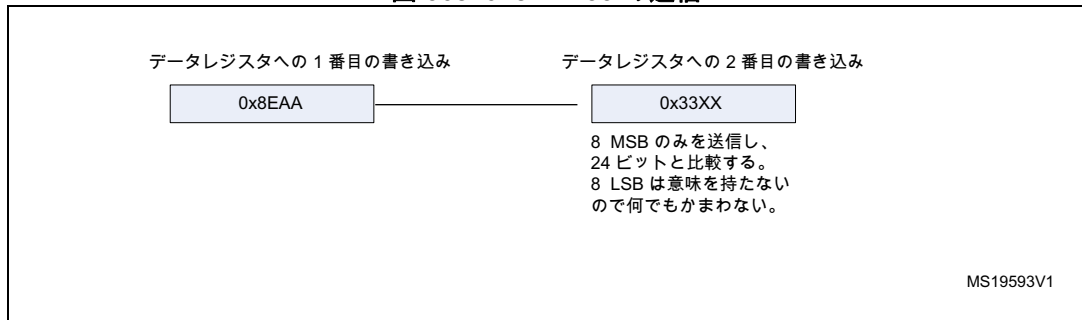


このモードでは、SPIx\_DR レジスタに対して 2 回の書き込みまたは読み出し動作が必要です。

- 送信モードの場合

0x8EAA33 を送信する必要がある場合 (24 ビット) :

図 368. 0x8EAA33 の送信



- 受信モードの場合  
データ 0x8EAA33 が受信される場合 :

図 369. 0x8EAA33 の受信

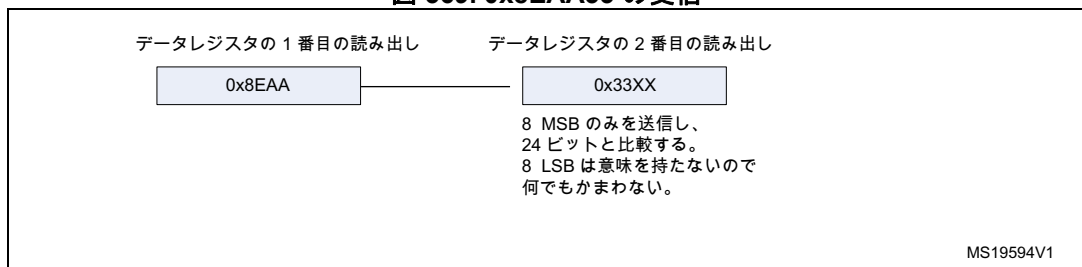
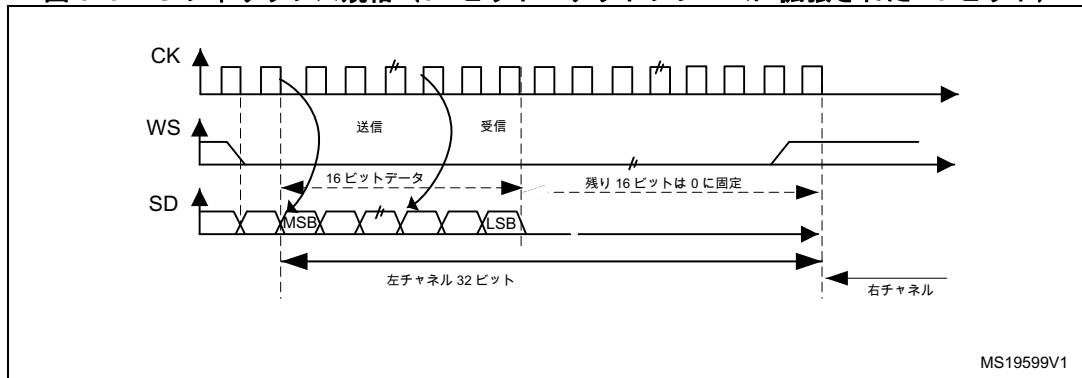


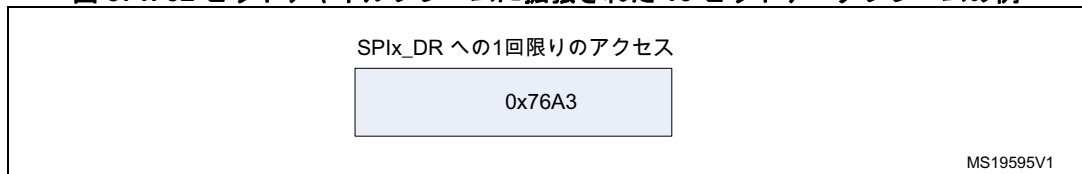
図 370. I<sup>2</sup>S フィリップス規格 (32 ビットパケットフレームに拡張された 16 ビット)



I<sup>2</sup>S 設定フェーズで、32 ビットチャンネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx\_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x76A30000) であるとき、[図 371](#) に示す動作が要求されます。

図 371. 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームの例



送信では、SPIx\_DR に MSB が書き込まれるたびに、TXE フラグがセットされ、可能ならば、SPIx\_DR レジスタに新しい送信値をロードするために割り込みが生成されます。これはハードウェアによって行われるため、0x0000 がまだ送信されていなくても行われます。

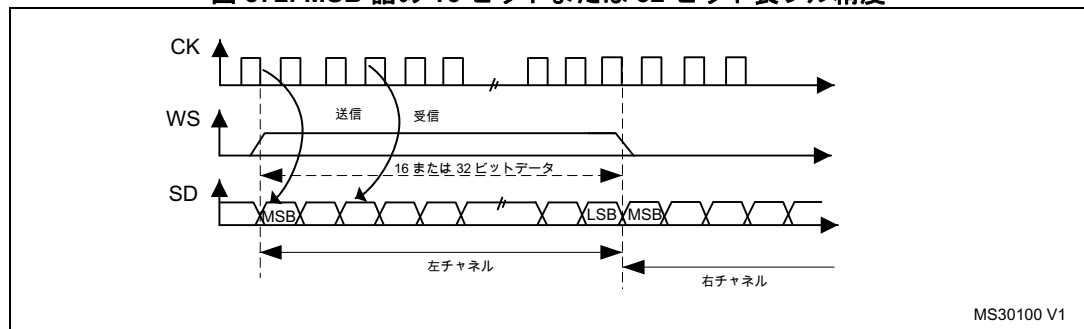
受信では、RXNE フラグがセットされ、可能ならば、最初の 16 MSB ハーフワードの受信時にその割り込みが生成されます。

このように、2 回の書き込みまたは読み出し動作の間にはより多くの時間が設けられるため、アンダーランまたはオーバーラン状態（データ転送の方向に依存）を避けられます。

## MSB 詰め規格

この規格では、WS 信号は最初のデータビット（最上位ビット）と同時に生成されます。

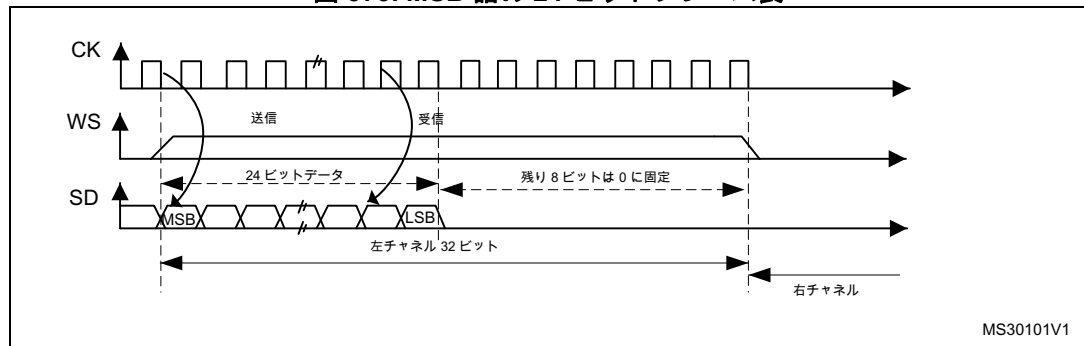
図 372. MSB 詰め 16 ビットまたは 32 ビット長フル精度



MS30100 V1

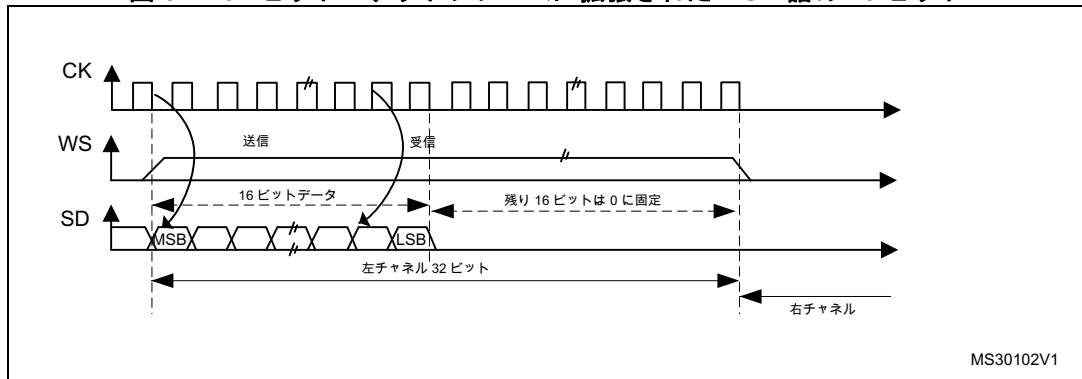
データは、CK の立ち下がりエッジでラッチされ（トランスミッタの場合）、立ち上がりエッジで読み出されます（レシーバの場合）。

図 373. MSB 詰め 24 ビットフレーム長



MS30101V1

図 374. 32 ビットパケットフレームに拡張された MSB 詰め 16 ビット



### LSB 詰め規格

この規格は、MSB 詰め規格と似ています（16 ビットと 32 ビットのフル精度フレームフォーマットに違いはありません）。

入出力信号のサンプリングは I2S フィリップス規格と同じです。

図 375. LSB 詰め 16 ビットまたは 32 ビットフル精度

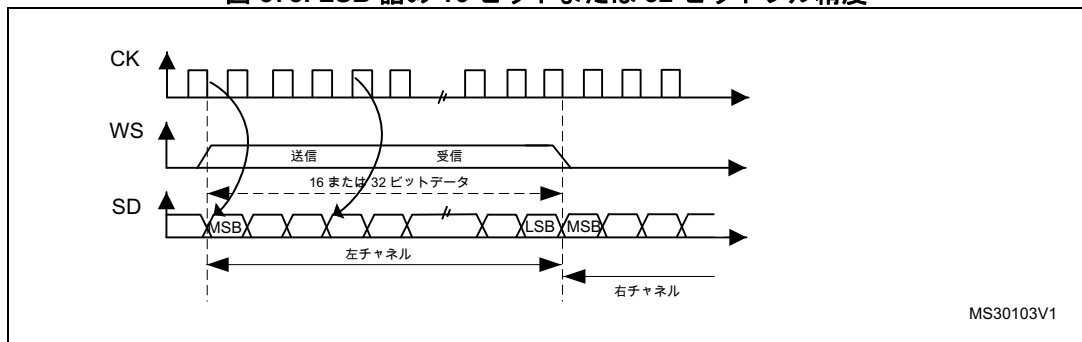
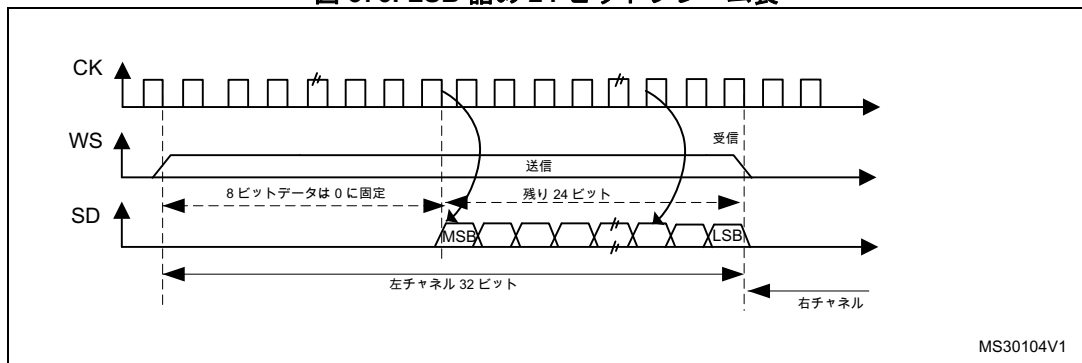


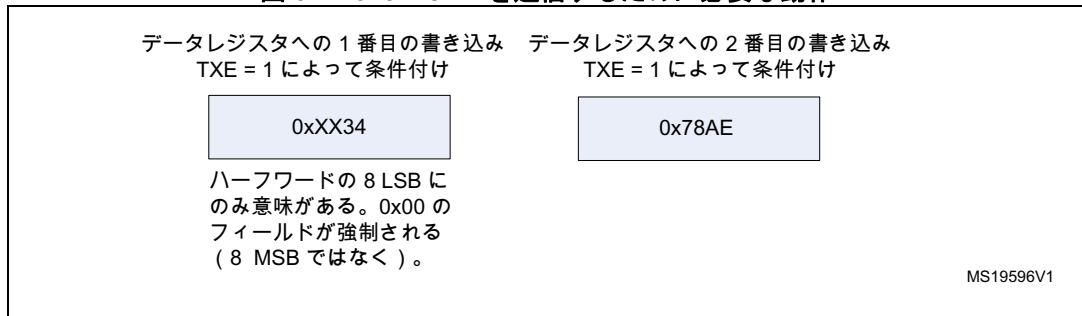
図 376. LSB 詰め 24 ビットフレーム長



- 送信モードの場合

データ 0x3478AE を送信する必要がある場合、ソフトウェアまたは DMA によって SPIx\_DR レジスタへの 2 回の書き込み動作が必要です。この動作を次に示します。

図 377. 0x3478AE を送信するために必要な動作



- 受信モードの場合

データ 0x3478AE が受信される場合、RXNE イベントごとに、SPIx\_DR レジスタから連続する 2 回の読み出し動作が必要です。

図 378. 0x3478AE の受信に必要な動作

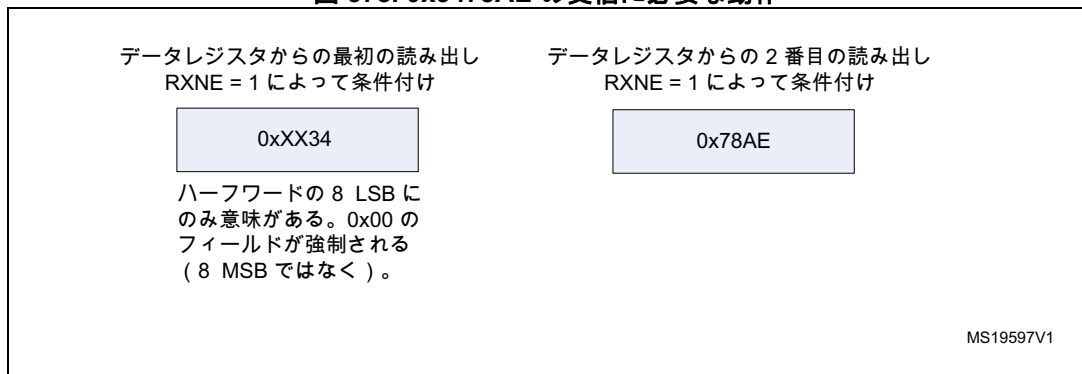
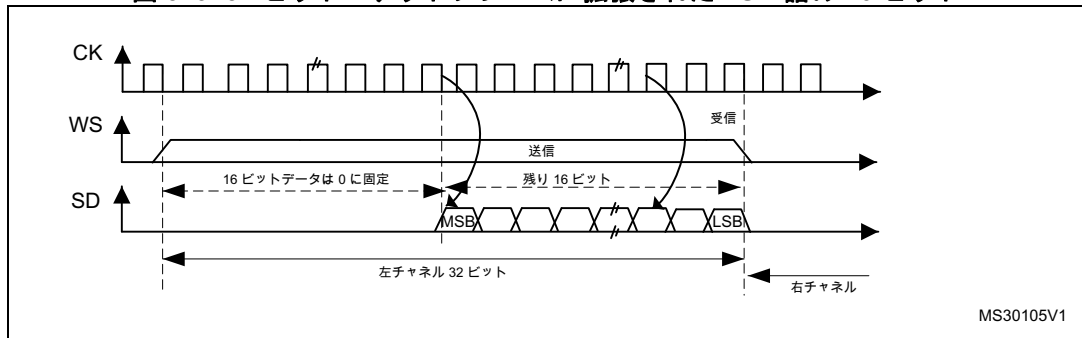


図 379. 32 ビットパケットフレームに拡張された LSB 詰め 16 ビット

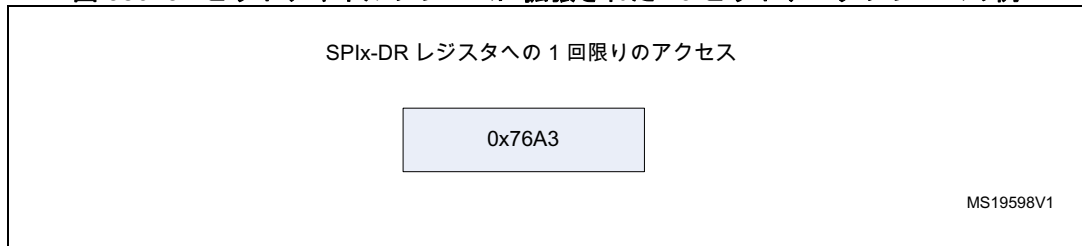


I<sup>2</sup>S 設定フェーズで、32 ビットチャンネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx\_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。この場合、それはハーフワード MSB に相当します。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x0000 76A3) であるとき、[図 380](#) に示す動作が要求されます。



図 380. 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームの例



送信モードでは、TXE イベントが発生すると、アプリケーションは送信するデータ（この場合は 0x76A3）を書き込む必要があります。0x000 フィールドが最初に送信されます（32 ビット拡張部）。有効なデータ（0x76A3）が SD に送信されると、すぐに TXE フラグが再びセットされます。

受信モードでは、0x0000 フィールドではない有効なハーフワードが受信されると、すぐに RXNE がアサートされます。

このように、アンダーランやオーバーランの状態を防ぐために、2 回の書き込みまたは読み出し動作の間には、より多くの時間が設けられています。

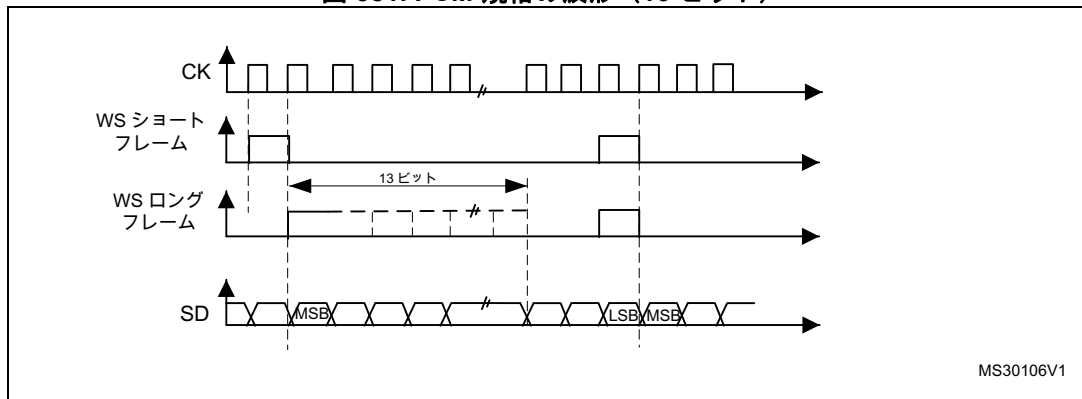
## PCM 規格

PCM 規格では、チャンネルサイド情報を使用する必要はありません。SPIx\_I2SCFGR レジスタの PCMSYNC ビットを使用して、2 つの PCM モード（ショートおよびロングフレーム）の使用および設定が可能です。

PCM モードで、出力信号（WS、SD）は CK 信号の立ち上がりエッジでサンプリングされます。入力信号（WS、SD）は立ち下がりエッジでキャプチャされます。

CK および WS は、マスタモードで出力として設定されることに注意してください。

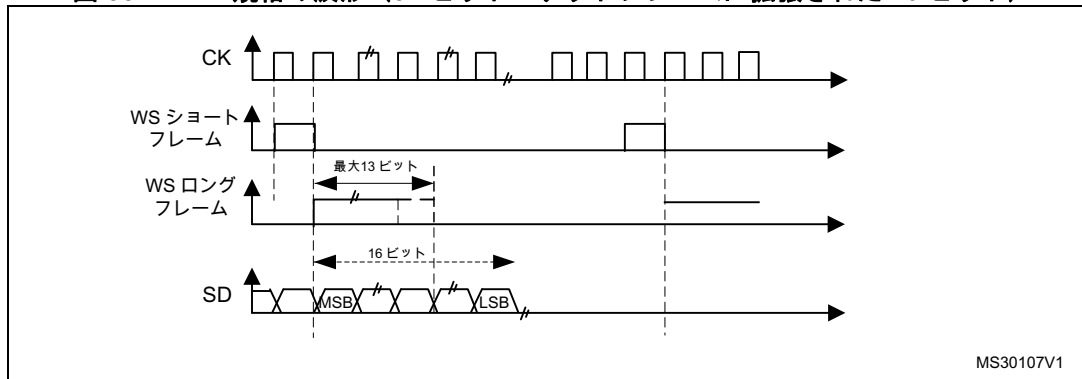
図 381. PCM 規格の波形（16 ビット）



ロングフレーム同期では、WS 信号のアサクション時間はマスタモードで 13 ビットに固定されています。

ショートフレーム同期では、WS 同期信号の長さは、わずか 1 サイクルです。

図 382. PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)

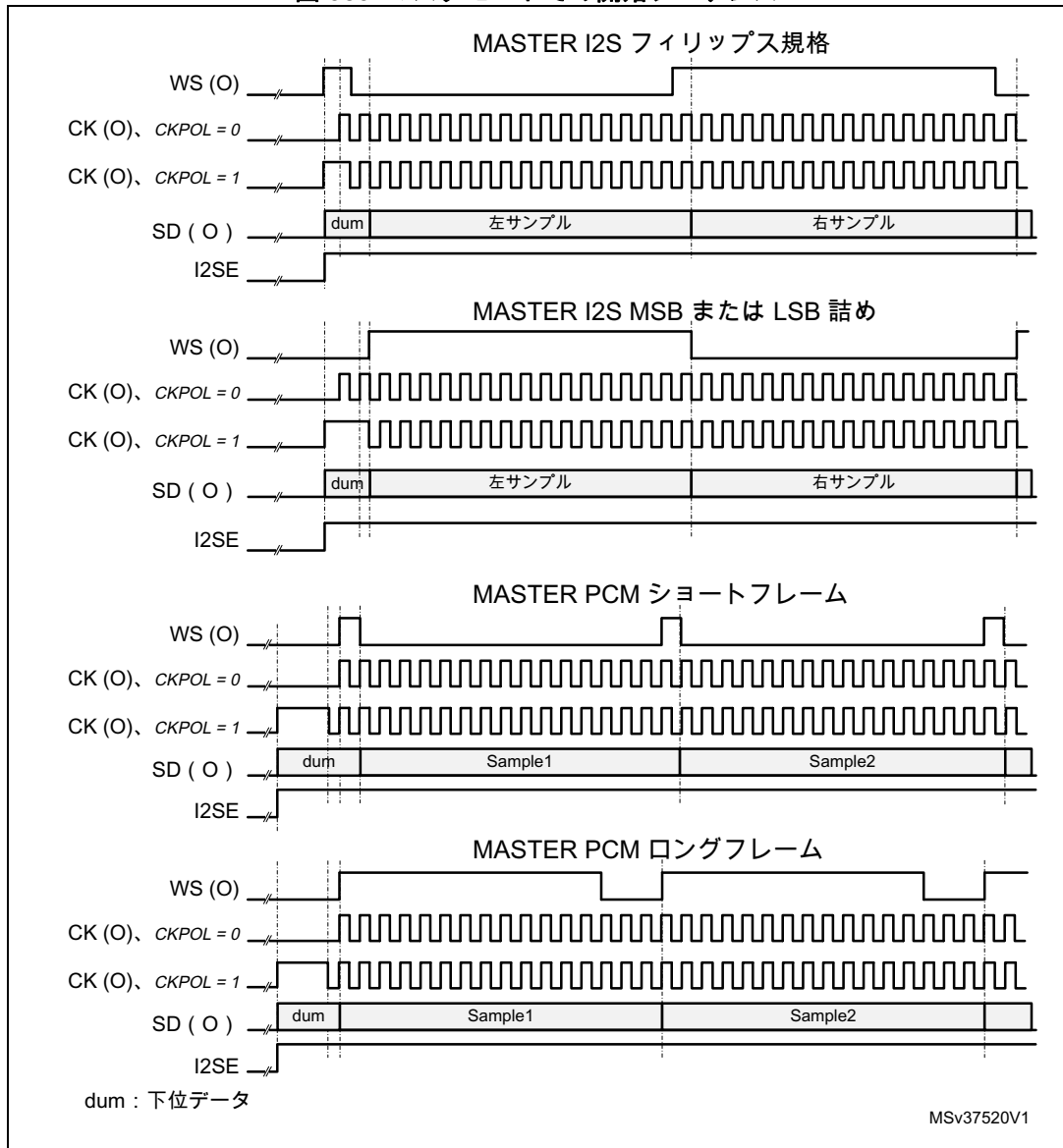


注： 2つのモード（マスタとスレーブ）と2つの同期（ショートとロング）に関しては、スレーブモードでも、連続した2つのデータ（したがって2つの同期信号）間のビット数を（SPIx\_I2SCFGR レジスタの DATLEN および CHLEN ビットで）指定する必要があります。

## 30.7.4 起動に関する説明

図 383 は、SPI/I2S が有効化された場合 (I2SE ビット) の MASTER モードでのシリアルインタフェースの処理について示しています。生成された信号での CKPOL の影響についても示します。

図 383. マスタモードでの開始シーケンス



スレーブモードの場合、ユーザは WS がアクティブになる前にオーディオインタフェースを有効にする必要があります。これは、I2S フィリップス規格が WS = 1、またはその他の規格が WS = 0 の場合に、I2SE ビットを 1 にセットする必要があることを意味しています。

### 30.7.5 クロックジェネレータ

I<sup>2</sup>S ビットレートにより、I<sup>2</sup>S データライン上のデータフローと I<sup>2</sup>S クロック信号周波数が決まります。

I<sup>2</sup>S ビットレート = チャンネルあたりのビット数 × チャンネル数 × オーディオサンプリング周波数

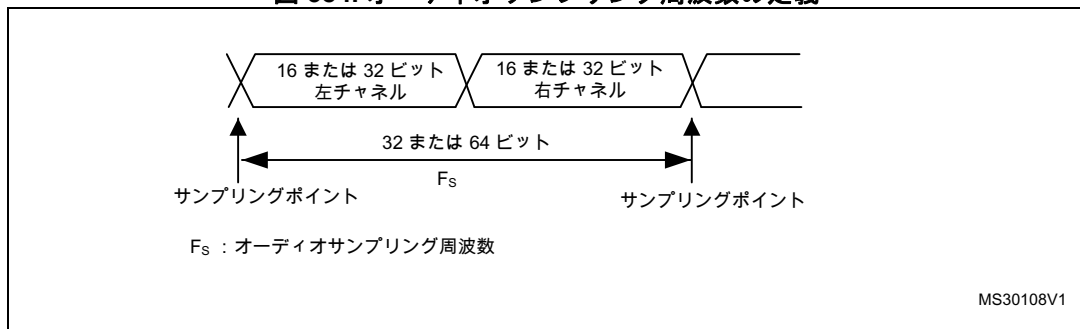
16 ビットオーディオ、左チャンネルおよび右チャンネルの場合、I<sup>2</sup>S ビットレートは次のように算出されます。

$$I^2S \text{ ビットレート} = 16 \times 2 \times f_s$$

パケット長が 32 ビットの場合、I<sup>2</sup>S ビットレート = 32 × 2 × f<sub>s</sub> となります。

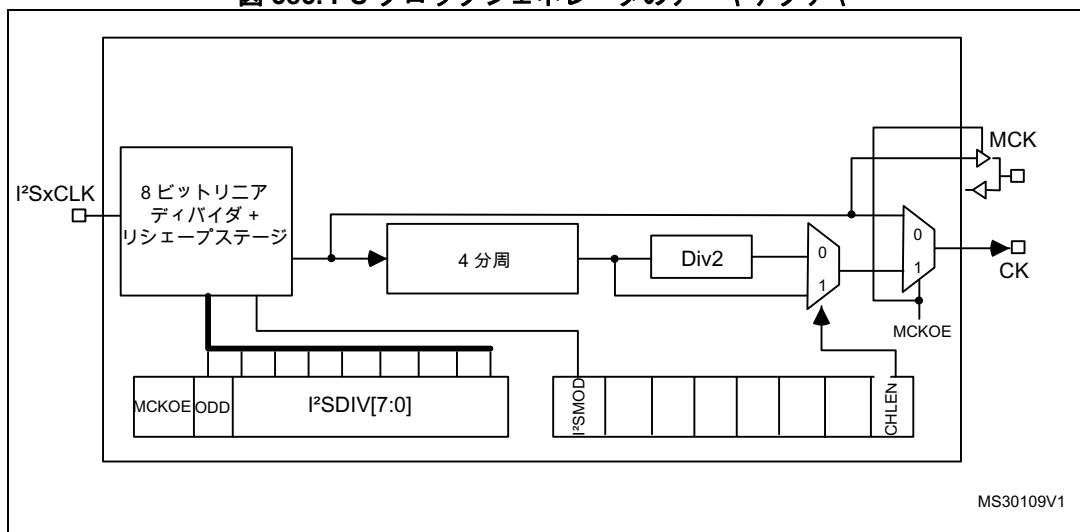


図 384. オーディオサンプリング周波数の定義



マスタモードが設定された場合、希望するオーディオ周波数で通信するために、特定の措置を講じてリニアディバイダを適切にプログラムする必要があります。

図 385. I<sup>2</sup>S クロックジェネレータのアーキテクチャ



1. x は 2 または 3 です。

図 385 に、通信クロックのアーキテクチャを示します。デフォルトでは、I2Sx クロックは常にシステムクロックです。高品質のオーディオ性能を実現するため、I2SxCLK クロックソースとしては、外部クロック (I2S\_CKIN ピンに割り付け) を使用することができます。セクション 9.4.2: クロック設定レジスタ (RCC\_CFGR) を参照してください。

オーディオサンプリング周波数は、192 KHz、96 kHz、または 48 kHz が可能です。希望する周波数を達成するには、次の式に基づいてリニアディバイダをプログラムする必要があります。

マスタクロックが生成される (SPIx\_I2SPR レジスタの MCKOE ビットをセットされる) 場合

$$f_s = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8] \dots \text{チャンネルフレームが 16 ビット幅の場合}$$

$$f_s = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4] \dots \text{チャンネルフレームが 32 ビット幅の場合}$$

マスタクロックが無効にされた場合 (MCKOE ビットをクリア)

$$f_s = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャンネルフレームが 16 ビット幅の場合}$$

$$f_s = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャンネルフレームが 32 ビット幅の場合}$$

表 168 に、さまざまなクロック設定に対する精度値の例を示します。

**注：** 最適クロック精度を可能にするその他の設定も可能です。

表 168. 標準 8 MHz HSE を使用した場合のオーディオ周波数精度<sup>(1)</sup>

SYSCLK (MHz)	I2S_DIV		I2S_ODD		MCLK	ターゲット f <sub>s</sub> (Hz)	実際の f <sub>s</sub> (Hz)		誤差	
	16 ビット	32 ビット	16 ビット	32 ビット			16 ビット	32 ビット	16 ビット	32 ビット
72	11	6	1	0	なし	96000	97826.09	93750	1.90%	2.34%
72	23	11	1	1	なし	48000	47872.34	48913.04	0.27%	1.90%
72	25	13	1	0	なし	44100	44117.65	43269.23	0.04%	1.88%
72	35	17	0	1	なし	32000	32142.86	32142.86	0.44%	0.44%
72	51	25	0	1	なし	22050	22058.82	22058.82	0.04%	0.04%
72	70	35	1	0	なし	16000	15675.75	16071.43	0.27%	0.45%
72	102	51	0	0	なし	11025	11029.41	11029.41	0.04%	0.04%
72	140	70	1	1	なし	8000	8007.11	7978.72	0.09%	0.27%
72	3	3	0	0	あり	48000	46875	46875	2.34%	2.34%
72	3	3	0	0	あり	44100	46875	46875	6.29%	6.29%
72	9	9	0	0	あり	32000	31250	31250	2.34%	2.34%
72	6	6	1	1	あり	22050	21634.61	21634.61	1.88%	1.88%
72	9	9	0	0	あり	16000	15625	15625	2.34%	2.34%
72	13	13	0	0	あり	11025	10817.30	10817.30	1.88%	1.88%
72	17	17	1	1	あり	8000	8035.71	8035.71	0.45%	0.45%

1. この表の値は、さまざまなクロック設定に対するほんの一例です。最適クロック精度を可能にするその他の設定も可能です。

## 30.7.6 I<sup>2</sup>S マスタモード

I<sup>2</sup>S は次のように設定することができます：

- 送信用または受信用のマスタモード (I2Sx を使用した半二重モード)
- 送受信用のマスタモード (I2Sx および I2Sx\_ext を使用した全二重モード)

つまり、シリアルクロックは、ワードセレクト信号 WS だけでなく、CK ピン上でも生成されます。マスタクロック (MCK) は、SPIx\_I2SPR レジスタの MCKOE ビットで、出力するかしないかを制御できます。

### 手順

1. 適切なオーディオサンプリング周波数に到達するシリアルクロックポーレートを定義するため、SPIx\_I2SPR レジスタの I2SDIV[7:0] ビットを選択します。SPIx\_I2SPR レジスタの ODD ビットも定義する必要があります。
2. 通信クロックの一定したレベルを定義するために、CKPOL ビットを選択します。外部の DAC/ADC オーディオコンポーネントにマスタクロック MCK を供給する必要がある場合、SPIx\_I2SPR レジスタの MCKOE ビットをセットします (I2SDIV と ODD の値は、MCK 出力の状態に応じて計算する必要があります。詳細については、[セクション 30.7.5 : クロックジェネレータ](#)を参照)。
3. SPIx\_I2SCFGR レジスタの I2SMOD ビットをセットして I<sup>2</sup>S 機能を有効にし、I2SSTD[1:0] と PCMSYNC ビットにより I<sup>2</sup>S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してチャンネルあたりのビット数を、それぞれ選択します。SPIx\_I2SCFGR レジスタの I2SCFG[1:0] ビットにより I<sup>2</sup>S マスタモードと方向 (トランスミッタまたはレシーバ) も選択します。



4. 必要な場合は、SPIx\_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
5. SPIx\_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

WS と CK は出力モードに設定されます。SPIx\_I2SPR の MCKOE ビットがセットされている場合、MCK も出力です。

## 送信シーケンス

送信シーケンスは、Txバッファにハーフワードが書き込まれたときに開始されます。

Txバッファに書き込まれる最初のデータは、左チャンネルのデータに対応すると想定します。データが Txバッファからシフトレジスタに転送されると、TXE がセットされ、右チャンネルに対応するデータを Txバッファに書き込む必要があります。CHSIDE フラグは、どのチャンネルが送信されるかを示します。CHSIDE フラグは TXE がハイレベルになったときに更新されるため、このフラグは TXE フラグがセットされた場合に意味を持ちます。

完全なフレームとは、左チャンネルのデータ送信と、それに続く右チャンネルのデータ送信であるとみなす必要があります。左チャンネルのみが送信される部分的フレームは実現できません。

データハーフワードは、最初のビット送信時に 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx\_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

選択された I<sup>2</sup>S 規格モードに応じた書き込み動作の詳細については、[セクション 30.7.3: サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx\_DR レジスタに書きこむ必要があります。

I2SE をクリアすることによって I<sup>2</sup>S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

## 受信シーケンス

動作モードは、ポイント 3 を除いて、送信モードの場合と同じです ([セクション 30.7.6: I<sup>2</sup>S マスタモード](#)に示す手順を参照)。ポイント 3 では、I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットの packets によって受信されます。つまり、Rxバッファが満杯になるたびに RXNE フラグがセットされ、SPIx\_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャンネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャンネルまたは左チャンネルに受信されることがあります。

RXNE ビットは、SPIx\_DR レジスタの読み出しによってクリアされます。

CHSIDE は毎回の受信後に更新されます。CHSIDE は、I<sup>2</sup>S セルによって生成される WS 信号に反応します。

選択された I<sup>2</sup>S 規格モードに応じた読み出し動作の詳細については、[セクション 30.7.3: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx\_CR2 レジスタの ERRIE ビットがセットされている場合、割り込みが生成されてエラーを知らせます。

I<sup>2</sup>S をスイッチオフするには、I<sup>2</sup>S が新しいデータ転送を開始することなく転送サイクルを適切に完了できるように、特定の動作が要求されます。そのシーケンスは、データ長とチャンネル長の設定、および選択したオーディオプロトコルモードに依存します。ケース別の説明

- 32 ビットチャンネル長に拡張された 16 ビットデータ長(DATLEN = 00、CHLEN = 1)、LSB 詰めモードを使用 (I2SSTD = 10)
  - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
  - b) 次に I<sup>2</sup>S の 17 クロックサイクルを待ちます (ソフトウェアループを使用)。
  - c) I<sup>2</sup>S を無効にします (I2SE = 0)。
- 32 ビットチャンネル長に拡張された 16 ビットデータ長(DATLEN = 00 および CHLEN = 1)、MSB 詰め、I<sup>2</sup>S または PCM モード (それぞれ、I2SSTD = 00、I2SSTD = 01、または I2SSTD = 11)
  - a) 最後の RXNE を待ちます。
  - b) 次に I<sup>2</sup>S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
  - c) I<sup>2</sup>S を無効にします (I2SE = 0)。
- DATLEN と CHLEN のその他すべての組み合わせについては、I2SSTD ビットを通じて選択したオーディオモードが何であれ、次のシーケンスを実行して I<sup>2</sup>S をスイッチオフします。
  - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
  - b) 次に I<sup>2</sup>S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
  - c) I<sup>2</sup>S を無効にします (I2SE = 0)。

**注：** 転送時、BSY フラグはローレベルに保持されます。

## 30.7.7 I<sup>2</sup>S スレーブモード

I2S は次のように設定することができます：

- 送信用または受信用のスレーブモード (I2Sx を使用した半二重モード)
- 送受信用のスレーブモード (I2Sx および I2Sx\_ext を使用した全二重モード)

動作モードは、主として I<sup>2</sup>S マスタ設定に関して述べたものと同じ規則に従います。スレーブモードでは、I<sup>2</sup>S インタフェースによって生成されるクロックはありません。クロックと WS 信号は、I<sup>2</sup>S インタフェースに接続された外部マスタから入力されます。したがって、ユーザがクロックを設定する必要はありません。

設定ステップを次に示します。

1. SPIx\_I2SCFGR レジスタの I2SMOD ビットをセットして I<sup>2</sup>S モードを選択し、I2SSTD[1:0] ビットにより I<sup>2</sup>S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してフレームのチャンネルあたりのビット数を、それぞれ選択します。SPIx\_I2SCFGR レジスタの I2SCFG[1:0] ビットにより、スレーブのモード (送信または受信) も選択します。
2. 必要な場合は、SPIx\_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
3. SPIx\_I2SCFGR レジスタの I2SE ビットをセットする必要があります (下の注を参照)。

**注：** 外部マスタで WS ラインを I2S プロトコルを選択した場合はハイレベル、LSB 詰めまたは MSB 詰めモードを選択した場合はローレベルにセットした後、I2S スレーブを有効化する必要があります。

### 送信シーケンス

送信シーケンスは、外部マスタデバイスがクロックを送信したときと、NSS\_WS 信号がデータの転送を要求したときに開始されます。スレーブを有効にしなければ、外部マスタは通信を開始しません。I<sup>2</sup>S データレジスタは、マスタが通信を開始する前に、ロードする必要があります。



I<sup>2</sup>S、MSB 詰め、および LSB 詰めモードの場合、データレジスタに書き込まれる最初のデータは、左チャンネルのデータに対応します。通信が開始されると、データはTxバッファからシフトレジスタに転送されます。次に、右チャンネルのデータを I<sup>2</sup>S データレジスタに書き込むように要求するために、TXE フラグがセットされます。

CHSIDE フラグは、どのチャンネルが送信されるかを示します。スレーブモードでは、マスタ送信モードに比べて、CHSIDE は外部マスタからの WS 信号に反応します。つまり、スレーブが最初のデータの送信準備をしなければ、マスタはクロックを生成できません。WS アサーションは、最初に送信される左チャンネルに対応します。

**注：** I2SE は、マスタの最初のクロックが CK ラインに到達するよりも、少なくとも 2 PCLK サイクル前に書き込まれる必要があります。

データハーフワードは、最初のビット送信時に内部バスから 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx\_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

なお、Txバッファへの書き込みの前に、TXE フラグが 1 であることを確認する必要があります。

選択された I<sup>2</sup>S 規格モードに応じた書き込み動作の詳細については、[セクション 30.7.3: サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx\_DR レジスタに書きこむ必要があります。次のデータ通信の最初のクロックエッジよりも前にデータが SPIx\_DR レジスタに書き込まれない場合、アンダーランフラグがセットされ、割り込みが生成されることがあります。これによって、転送データに誤りがあることがソフトウェアに知らされます。SPIx\_CR2 レジスタの ERRIE ビットがセットされた場合、SPIx\_SR レジスタの UDR フラグがハイレベルになると割り込みが生成されます。この場合、I<sup>2</sup>S をスイッチオフし、左チャンネルからデータ転送をリスタートする必要があります。

I2SE をクリアすることによって I<sup>2</sup>S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

## 受信シーケンス

動作モードは、ポイント 1 を除いて、送信モードの場合と同じです ([セクション 30.7.7: I<sup>2</sup>S スレーブモード](#)に示す手順を参照)。ポイント 1 では、SPIx\_I2SCFGR レジスタの I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットのパケットによって受信されます。つまり、Rxバッファがフルになるたびに SPIx\_SR レジスタの RXNE フラグがセットされ、SPIx\_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャンネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャンネルまたは左チャンネルに受信されることがあります。

CHSIDE フラグは、SPIx\_DR レジスタから読み出されるデータが受信されるたびに更新されます。このフラグは、外部マスタコンポーネントによって管理される外部 WS ラインに反応します。

RXNE ビットは、SPIx\_DR レジスタの読み出しによってクリアされます。

選択された I<sup>2</sup>S 規格モードに応じた読み出し動作の詳細については、[セクション 30.7.3: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx\_CR2 レジスタの ERRIE ビットがセットされた場合、エラーを示すために割り込みが生成されます。



受信モードで I<sup>2</sup>S をスイッチオフするには、最後の RXNE = 1 を受信した直後に I2SE をクリアする必要があります。

**注：** 外部マスタコンポーネントには、オーディオチャネルを介して 16 ビットまたは 32 ビットの packets でデータを送受信する機能が必要です。

## 30.7.8 I<sup>2</sup>S ステータスフラグ

アプリケーションが I<sup>2</sup>S バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

### ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます (このフラグへの書き込みは無効)。このフラグは I<sup>2</sup>S の通信層の状態を示します。

BSY がセットされると、I<sup>2</sup>S が通信中でビジーであることを示します。マスタ受信モード (I2SCFG=11) には 1 つの例外があり、BSY フラグは受信時にローレベルに保持されます。

ソフトウェアが I<sup>2</sup>S を無効にする必要がある場合、BSY フラグは転送の終わりを検出するために役立ちます。これによって、最後の転送データの破壊を回避します。このため、下記の手順を厳守する必要があります。

BSY フラグは転送が開始されるとセットされます。ただし、I<sup>2</sup>S がマスタ受信モードにある場合を除きます。

BSY フラグをクリアするタイミング

- 転送が完了したとき (ただし、通信が連続的だと思われるマスタ送信モードの場合を除く)
- I<sup>2</sup>S が無効化されたとき

通信が連続的な場合

- マスタ送信モードでは、BSY フラグはすべての転送期間を通じてハイレベルに保持されます。
- スレーブモードでは、BSY フラグは、各転送間で I<sup>2</sup>S の 1 クロックサイクルの間ローレベルになります。

**注：** 各データの送受信の処理には BSY フラグを使用しないでください。代わりに、TXE フラグと RXNE フラグを使用することをお勧めします。

### Txバッファエンプティフラグ (TXE)

このフラグがセットされると、Txバッファはエンプティ (空) であり、次に送信するデータをバッファにロードできることを示します。送信されるデータがすでに Txバッファに格納されているとき、TXE フラグはリセットされます。TXE フラグは、I<sup>2</sup>S が無効にされている (I2SE ビットがリセット) ときにもリセットされます。

### Rxバッファノットエンプティ (RXNE)

このフラグがセットされると、Rxバッファに有効な受信データがあることを示します。このフラグは、SPIx\_DR レジスタが読み出されるとリセットされます。

### チャンネルサイドフラグ (CHSIDE)

送信モードでは、このフラグは TXE がハイレベルになるとリフレッシュされます。このフラグは、SD 上の転送データが属するチャンネルサイドを示します。スレーブ送信モードでアンダーランエラーイベントが発生した場合、このフラグは信頼できないため、通信を再開する前に、I<sup>2</sup>S をスイッチオフし、さらにスイッチオンする必要があります。



受信モードでは、このフラグは SPIx\_DR にデータが受信されるとリフレッシュされます。このフラグは、どちらのチャンネルサイドからデータが受信されたかを示します。なお、エラー (OVR など) が発生した場合、このフラグは無意味になるため、I<sup>2</sup>S を無効にし、さらに有効にすることによってリセットする必要があります (変更が必要な場合は設定する)。

このフラグは、PCM 規格では意味を持ちません (ショートとロングフレームの両モード)。

SPIx\_SR の OVR または UDR フラグがセットされ、SPIx\_CR2 の ERRIE ビットもセットされると、割り込みが生成されます。この割り込みをクリアするには、割り込みソースをいったんクリアした後で、SPIx\_SR ステータスレジスタを読み出します。

## 30.7.9 I<sup>2</sup>S エラーフラグ

I<sup>2</sup>S セルには 3 つのエラーフラグがあります。

### アンダーランフラグ (UDR)

スレーブ送信モードでは、ソフトウェアが SPIx\_DR にまだ値をロードしていない間に、データ送信用の最初のクロックが現れると、このフラグがセットされます。このフラグは、SPIx\_I2SCFGR レジスタの I2SMOD ビットがセットされると使用できます。SPIx\_CR2 レジスタの ERRIE ビットがセットされている場合は、割り込みを生成できます。

UDR ビットは、SPIx\_SR レジスタの読み出し動作によってクリアされます。

### オーバーランフラグ (OVR)

このフラグがセットされるのは、データが受信され、前のデータが SPIx\_DR レジスタからまだ読み出されていないときです。結果として、受信データは失われます。SPIx\_CR2 の ERRIE ビットがセットされている場合、割り込みが生成されることがあります。

この場合、受信バッファの内容は、送信側のデバイスからの新しい受信データによって更新されません。SPIx\_DR レジスタへの読み出し動作によって、前に正しく受信されたデータが返されます。それ以降に送信されたすべてのハーフワードは失われます。

OVR ビットをクリアするには、SPIx\_DR レジスタを読み出し、続けて SPIx\_SR レジスタに読み出しアクセスを行います。

### フレームエラーフラグ (FRE)

このフラグは、I<sup>2</sup>S がスレーブモードに設定された場合にのみハードウェアによってセットすることができます。このフラグは、スレーブが WS ラインの変化を想定していないときに外部マスタが WS ラインを変化させようとしている場合にセットされます。同期が失われた場合、この状態から回復し、外部マスタデバイスと I<sup>2</sup>S スレーブデバイスを再同期させるためには、次のステップに従います。

1. I<sup>2</sup>S を無効にします。
2. 正しいレベルが WS ラインで検出されたとき (WS ラインは I<sup>2</sup>S モードではハイレベル、MSB 詰めまたは LSB 詰めあるいは PCM モードではローレベル)、再度有効にします。

マスタデバイスとスレーブデバイスの間の同期外れは、CK 通信クロック上または WS フレーム同期ライン上のノイズの多い環境に起因する可能性があります。ERRIE ビットをセットすれば、エラー割り込みを生成させることができます。同期外れフラグ (FRE) は、ステータスレジスタを読み出すときに、ソフトウェアでクリアされます。

## 30.7.10 DMA の機能

I<sup>2</sup>S モードでは、DMA は SPI モードとまったく同じように機能します。異なる点は、I<sup>2</sup>S モードではデータ転送保護システムがないため、CRC 機能を使用できない点のみです。

## 30.8 I<sup>2</sup>S 割り込み

表 169 に I<sup>2</sup>S 割り込みのリストを示します。

表 169. I<sup>2</sup>S 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
送信バッファエンプティフラグ	TXE	TXEIE
受信バッファノットエンプティフラグ	RXNE	RXNEIE
オーバーランエラー	OVR	ERRIE
アンダーランエラー	UDR	
フレームエラーフラグ	FRE	

## 30.9 SPI および I<sup>2</sup>S レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。さらに、SPI\_DR へは8 ビット単位でアクセスできます。

### 30.9.1 SPI 制御レジスタ 1 (SPIx\_CR1)

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15 **BIDIMODE**：双方向データモードイネーブル。このビットは、共通の双方向データラインを 1 本使用して、半二重通信を有効にします。双方向モードがアクティブのときは、RXONLY ビットをクリアされたままにします。

- 0：2 線単方向データモードを選択します。
- 1：1 線双方向データモードを選択します。

**注：** このビットは I<sup>2</sup>S モードでは使用しません。

ビット 14 **BIDIOE**：双方向モードでの出力イネーブル

双方向モードでの転送方向は、このビットと BIDIMODE ビットを組み合わせで選択します。

- 0：出力は無効です（受信専用モード）。
- 1：出力は有効です（送信専用モード）。

**注：** マスタモードでは MOSI ピンが使用され、スレーブモードでは MISO ピンが使用されます。このビットは I<sup>2</sup>S モードでは使用しません。

ビット 13 **CRCEN**：ハードウェア CRC 計算イネーブル

- 0：CRC 計算は無効です。
- 1：CRC 計算は有効です。

**注：** 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。このビットは I<sup>2</sup>S モードでは使用しません。

ビット 12 **CRCNEXT**：送信 CRC Next

- 0：次の送信値は Tx バッファから送信されます。
- 1：次の送信値は Tx CRC レジスタから送信されます。

**注：** このビットは、最後のデータが SPIx\_DR レジスタに書き込まれた直後に書き込む必要があります。

このビットは I<sup>2</sup>S モードでは使用しません。

ビット 11 **CRCL**：CRC 長

このビットは、CRC 長をセットするために、ソフトウェアによってセット/クリアされます。

- 0：8 ビットの CRC 長
- 1：16 ビットの CRC 長

**注：** 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 10 RXONLY** : 受信専用モードイネーブル

このビットは、データ受信専用の単方向ラインを 1 本使用して、単方向通信を有効にします。受信専用モードがアクティブのときは、BIDIMODE ビットをクリアされたままにします。このビットはマルチスレーブシステムでも役立ちます。そのシステムでは、この特定のスレーブはアクセスされず、アクセスされたスレーブからの出力は破壊されません。

0 : 全二重 (送受信)

1 : 出力は無効です (受信専用モード)。

**注:** このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 9 SSM** : ソフトウェアスレーブ管理

SSM ビットがセットされているとき、NSS ピンの入力は SSI ビットからの値に置き換えられます。

0 : ソフトウェアスレーブ管理は無効です。

1 : ソフトウェアスレーブ管理は有効です。

**注:** このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 8 SSI** : 内部スレーブ選択

このビットは、SSM ビットがセットされているときにのみ有効です。このビットの値は強制的に NSS ピンに設定され、NSS ピンの I/O 値は無視されます。

**注:** このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 7 LSBFIRST** : フレームフォーマット

0 : データは MSB ファーストとともに送信/受信されます。

1 : データは LSB ファーストとともに送信/受信されます。

**注:** 1. このビットは、通信中には変更しないでください。

2. このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 6 SPE** : SPI イネーブル

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

**注:** SPI を無効にするときは、SPI を無効にする手順 (962 ページ) に記載されている手順に従ってください。

このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 5:3 BR[2:0]** : ポーレート制御

000 :  $f_{HCLK} / 2$

001 :  $f_{HCLK} / 4$

010 :  $f_{HCLK} / 8$

011 :  $f_{HCLK} / 16$

100 :  $f_{HCLK} / 32$

101 :  $f_{HCLK} / 64$

110 :  $f_{HCLK} / 128$

111 :  $f_{HCLK} / 256$

**注:** これらのビットは、通信中には変更しないでください。

このビットは I<sup>2</sup>S モードでは使用しません。

ビット 2 **MSTR** : マスタ選択

- 0 : スレーブ設定
- 1 : マスタ設定

**注 :** このビットは、通信中には変更しないでください。  
このビットは I<sup>2</sup>S モードでは使用しません。

Bit1 **CPOL** : クロック極性

- 0 : アイドル時に CK は 0 となります。
- 1 : アイドル時に CK は 1 となります。

**注 :** このビットは、通信中には変更しないでください。  
このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

ビット 0 **CPHA** : クロック位相

- 0 : 最初のクロック遷移が最初のデータキャプチャエッジです。
- 1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

**注 :** このビットは、通信中には変更しないでください。  
このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

## 30.9.2 SPI 制御レジスタ 2 (SPIx\_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXTH	DS [3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **LDMA\_TX** : 送信の最後の DMA 転送

このビットは、DMA で送信するデータの合計数が奇数であるか偶数であるかを定義するために、データパッキングモードで使用されます。これは、SPIx\_CR2 レジスタの TXDMAEN ビットがセットされている場合で、パッキングモードが使用されている場合にのみ意味を持ちます (データ長 =< 8 ビット、SPIx\_DR への書き込みアクセスは 16 ビット幅)。SPI が無効化された場合に書き込む必要があります (SPIx\_CR1 レジスタで SPE = 0)。

- 0 : 転送データ項目の数は偶数です。
- 1 : 転送データ項目の数は奇数です。

**注 :** **CRCEN** ビットがセットされている場合、[SPI を無効にする手順 \(962 ページ\)](#) を参照してください。  
このビットは I<sup>2</sup>S モードでは使用しません。

ビット 13 **LDMA\_RX** : 受信の最後の DMA 転送

このビットは、DMA で受信するデータの合計数が奇数であるか偶数であるかを定義するために、データパッキングモードで使用されます。これは、SPIx\_CR2 レジスタの RXDMAEN ビットがセットされている場合で、パッキングモードが使用されている場合にのみ意味を持ちます (データ長 =< 8 ビット、SPIx\_DR への書き込みアクセスは 16 ビット幅)。SPI が無効化された場合に書き込む必要があります (SPIx\_CR1 レジスタで SPE = 0)。

- 0 : 転送データ項目の数は偶数です。
- 1 : 転送データ項目の数は奇数です。

**注 :** **CRCEN** ビットがセットされている場合、[SPI を無効にする手順 \(962 ページ\)](#) を参照してください。  
このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 12 FRXTH** : FIFO 受信閾値

このビットは、RXNE イベントをトリガする RXFIFO の閾値をセットするために使用されます。

0 : FIFO レベルが 1/2 (16 ビット) 以上である場合に RXNE イベントが生成されます。

1 : FIFO レベルが 1/4 (8 ビット) 以上である場合に RXNE イベントが生成されます。

**注:** このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 11:8 DS [3:0]** : データサイズ

以下のビットは SPI 転送のデータ長を設定します。

0000 : 未使用

0001 : 未使用

0010 : 未使用

0011 : 4 ビット

0100 : 5 ビット

0101 : 6 ビット

0110 : 7 ビット

0111 : 8 ビット

1000 : 9 ビット

1001 : 10 ビット

1010 : 11 ビット

1011 : 12 ビット

1100 : 13 ビット

1101 : 14 ビット

1110 : 15 ビット

1111 : 16 ビット

ソフトウェアが「未使用」値のいずれかの書き込みを試みた場合、値は強制的に「0111」(8 ビット)になります。

**注:** このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 7 TXEIE** : Txバッファエンプティ割り込みイネーブル

0 : TXE 割り込みはマスクされます。

1 : TXE 割り込みはマスクされません。TXE フラグがセットされたとき、割り込みリクエストの生成に使用されます。

**ビット 6 RXNEIE** : Rxバッファノットエンプティ割り込みイネーブル

0 : RXNE 割り込みはマスクされます。

1 : RXNE 割り込みはマスクされません。RXNE フラグがセットされたとき、割り込みリクエストの生成に使用されます。

**ビット 5 ERRIE** : エラー割り込みイネーブル

このビットは、エラー状態が発生したとき (SPI モードでは CRCERR、OVR、MODF ; TI モードでは FRE ; I<sup>2</sup>S モードでは UDR、OVR、FRE)、割り込みの生成を制御します。

0 : エラー割り込みはマスクされます。

1 : エラー割り込みは有効です。

**ビット 4 FRF** : フレームフォーマット

0 : SPI モトローラモード

1 : SPI TI モード

**注:** このビットは、SPI が無効 (SPE=0) のときにのみ書き込む必要があります。

このビットは I<sup>2</sup>S モードでは使用しません。

**ビット 3 NSSP** : NSS パルス管理

このビットは、マスタモードでのみ使用されます。これにより、SPI は連続転送中に 2 つの連続したデータ間で NSS パルスを生成できます。単一のデータ転送の場合、転送後、NSS ピンは強制的にハイレベルになります。

CPHA = '1' の場合も FRF = '1' の場合も意味を持ちません。

0 : NSS パルスなし

1 : NSS パルス発生回路

**注:** 1.このビットは、SPI が無効 (SPE=0) のときのみ書き込む必要があります。  
2.このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 2 SSOE** : SS 出カインーブル

0 : マスタモードで SS 出力は無効にされ、SPI インタフェースはマルチマスタ設定で機能できます。

1 : SPI インタフェースが有効であるとき、マスタモードで SS 出力は有効です。SPI インタフェースはマルチマスタ環境では機能できません。

**注:** このビットは I<sup>2</sup>S モードおよび SPI TI モードでは使用しません。

**ビット 1 TXDMAEN** : Txバッファ DMA イネーブル

このビットがセットされると、TXE フラグがセットされるたびに DMA リクエストが生成されます。

0 : Txバッファ DMA は無効です。

1 : Txバッファ DMA は有効です。

**ビット 0 RXDMAEN** : Rxバッファ DMA イネーブル

このビットがセットされると、RXNE フラグがセットされるたびに DMA リクエストが生成されます。

0 : Rxバッファ DMA は無効です。

1 : Rxバッファ DMA は有効です。



## 30.9.3 SPI ステータスレジスタ (SPIx\_SR)

アドレスオフセット : 0x08

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTLVL[1:0]		FRLVL[2:0]		FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0	r	r	r	r

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:11 **FTLVL[1:0]** : FIFO 送信レベル

これらのビットは、ハードウェアによってセット/クリアされます。

- 00 : FIFO エンプティ
- 01 : 1/4 FIFO
- 10 : 1/2 FIFO
- 11 : FIFO フル (FIFO 閾値が 1/2 より大きい場合、FULL とみなす)

**注 :** これらのビットは I<sup>2</sup>S モードでは使用されません。

ビット 10:9 **FRLVL[1:0]** : FIFO 受信レベル

これらのビットは、ハードウェアによってセット/クリアされます。

- 00 : FIFO エンプティ
- 01 : 1/4 FIFO
- 10 : 1/2 FIFO
- 11 : FIFO フル

**注 :** これらのビットは、CRC 計算が有効なときは I<sup>2</sup>S モードや SPI 受信専用モードでは使用されません。

ビット 8 **FRE** : フレームフォーマットエラー

このフラグは、TI スレーブモードと I<sup>2</sup>S スレーブモードの SPI で使用されます。[セクション 30.5.11 : SPI エラーフラグ](#)および[セクション 30.7.9 : I<sup>2</sup>S エラーフラグ](#)を参照してください。

このフラグは、ハードウェアによってセットされ、SPIx\_SR がソフトウェアによって読み出されるとリセットされます。

- 0 : フレームフォーマットエラーはありません。
- 1 : フレームフォーマットエラーが発生しました。

ビット 7 **BSY** : ビジーフラグ

- 0 : SPI (または I2S) はビジー状態ではありません。
  - 1 : SPI (または I2S) が通信ビジー状態であるか、または Tx バッファが空ではありません。
- このフラグはハードウェアによってセット/クリアされます。

**注 :** **BSY フラグを使用する際は注意が必要です。** [セクション 30.5.10 : SPI ステータスフラグ](#)および [SPI を無効にする手順 \(962 ページ\)](#) を参照してください。

ビット 6 **OVR** : オーバーランフラグ

- 0 : オーバーランは発生していません。
- 1 : オーバーランが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[I<sup>2</sup>S エラーフラグ \(994 ページ\)](#) を参照してください。

ビット 5 **MODF** : モードフォールト

- 0 : モードフォールトは発生していません。
- 1 : モードフォールトが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[セクション : モードフォールト \(MODF\) \(972 ページ\)](#) を参照してください。

**注：** このビットは I<sup>2</sup>S モードでは使用しません。

ビット 4 **CRCERR** : CRC エラーフラグ

- 0 : 受信した CRC 値が SPIx\_RXCRCR 値と一致します。
- 1 : 受信した CRC 値が SPIx\_RXCRCR 値と一致しません。

このフラグは、ハードウェアによってセットされ、ソフトウェアによって 0 を書き込むことでクリアされます。

**注：** このビットは I<sup>2</sup>S モードでは使用しません。

ビット 3 **UDR** : アンダーランフラグ

- 0 : アンダーランは発生していません。
- 1 : アンダーランが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[I<sup>2</sup>S エラーフラグ \(994 ページ\)](#) を参照してください。

**注：** このビットは SPI モードでは使用しません。

ビット 2 **CHSIDE** : チャネルサイド

- 0 : 左チャネルを送信する必要があるか、または受信が行われました。
- 1 : 右チャネルを送信する必要があるか、または受信が行われました。

**注：** このビットは SPI モードでは使用しません。PCM モードでは意味を持ちません。

ビット 1 **TXE** : 送信バッファエンプティ

- 0 : Tx バッファは空ではありません。
- 1 : Tx バッファは空です。

ビット 0 **RXNE** : 受信バッファノットエンプティ

- 0 : Rx バッファは空です。
- 1 : Rx バッファは空ではありません。

## 30.9.4 SPI データレジスタ (SPIx\_DR)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **DR[15:0]** : データレジスタ

受信したデータまたは送信されるデータ

このデータレジスタは、Rx および Tx FIFO 間のインタフェースとして使用できます。データレジスタが読み出されると、データレジスタへの書き込みが Tx FIFO にアクセスしている間に、Rx FIFO にアクセスされます ([セクション 30.5.9 : データの送受信手順](#)を参照)。

**注：** データは常に右詰めです。未使用のビットは、レジスタへの書き込み時に無視され、レジスタの読み出し時にゼロとして読み出されます。Rx 閾値設定は、常に現在使用中の読み出しアクセスに対応している必要があります。

## 30.9.5 SPI CRC 多項式レジスタ (SPIx\_CRCPR)

アドレスオフセット : 0x10

リセット値 : 0x0007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **CRCPOLY[15:0]** : CRC 多項式レジスタ

このレジスタは、CRC 計算用の多項式を格納します。

CRC 多項式 (0007h) は、このレジスタのリセット値です。必要に応じて、別の多項式を設定することができます。

**注：** 多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。

## 30.9.6 SPI Rx CRC レジスタ (SPIx\_RXCRCR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **RxCRC[15:0]** : Rx CRC レジスタ

CRC 計算が有効なとき、RxCRC[15:0] ビットには、その後に受信したバイトから算出された CRC 値が格納されています。このレジスタは、SPIx\_CR1 レジスタの CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPIx\_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPIx\_CR1 の CRCL ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPIx\_CR1 レジスタの CRCL ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

**注:** BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。

これらのビットは I<sup>2</sup>S モードでは使用されません。

## 30.9.7 SPI Tx CRC レジスタ (SPIx\_TXCRCR)

アドレスオフセット : 0x18

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **TxCRC[15:0]** : Tx CRC レジスタ

CRC 計算が有効なとき、TxCRC[15:0] ビットには、その後に送信されたバイトから算出された CRC 値が格納されます。このレジスタは、SPIx\_CR1 の CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPIx\_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPIx\_CR1 の CRCL ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPIx\_CR1 レジスタの CRCL ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

**注:** BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。

これらのビットは I<sup>2</sup>S モードでは使用されません。

## 30.9.8 SPIx\_I2S 設定レジスタ (SPIx\_I2SCFGR)

アドレスオフセット : 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	I2SMOD	I2SE	I2SCFG		PCMSYNC	Res.	I2SSTD		CKPOL	DATLEN		CHLEN
				r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:12 予約済み。ハードウェアによって 0 に固定されています。

ビット 11 **I2SMOD** : I2S モード選択

0 : SPI モードが選択されます。

1 : I2S モードが選択されます。

**注 :** このビットは、SPI または I2S が無効なときに設定してください。

ビット 10 **I2SE** : I2S イネーブル

0 : I2S ペリフェラルは無効です。

1 : I2S ペリフェラルは有効です。

**注 :** このビットは SPI モードでは使用しません。

ビット 9:8 **I2SCFG** : I2S 設定モード

00 : スレーブ - 送信

01 : スレーブ - 受信

10 : マスタ - 送信

11 : マスタ - 受信

**注 :** これらのビットは、I2S が無効なときに設定してください。

これらは SPI モードでは使用しません。

ビット 7 **PCMSYNC** : PCM フレーム同期

0 : ショートフレーム同期

1 : ロングフレーム同期

**注 :** このビットは、I2SSTD = 11 (PCM 規格使用) の場合にのみ意味を持ちます。

SPI モードでは使用しません。

ビット 6 予約済みであり、ハードウェアによって 0 に固定されています。

ビット 5:4 **I2SSTD** : I2S 規格選択

00 : フィリップス I2S 規格

01 : MSB 詰め規格 (左詰め)

10 : LSB 詰め規格 (右詰め)

11 : PCM 規格

I2S 規格に関する詳細は、[セクション 30.7.3 \(979 ページ\)](#) を参照してください。

**注 :** 正しい動作のためには、これらのビットは、I2S が無効のときに設定してください。

これらは SPI モードでは使用しません。

ビット 3 **CKPOL** : インアクティブ状態のクロック極性

0 : I<sup>2</sup>S クロックのインアクティブ状態はローレベルです。

1 : I<sup>2</sup>S クロックのインアクティブ状態はハイレベルです。

**注 :** 正しい動作のためには、このビットは、I<sup>2</sup>S が無効のときに設定してください。

**SPI モードでは使用しません。**

**CKPOL ビットは、SD 信号および WS 信号を受信または送信するために使用される CK エッジ感度には影響を与えません。**

ビット 2:1 **DATLEN** : 転送されるデータ長

00 : 16 ビットデータ長

01 : 24 ビットデータ長

10 : 32 ビットデータ長

11 : 設定禁止

**注 :** 正しい動作のためには、これらのビットは、I<sup>2</sup>S が無効のときに設定してください。

**これらは SPI モードでは使用しません。**

ビット 0 **CHLEN** : チャネル長 (オーディオチャネルごとのビット数)

0 : 16 ビット幅

1 : 32 ビット幅

ビット書き込み動作は、DATLEN=00 のときにのみ意味を持ちます。そうでない場合、書き込まれた値とは無関係に、チャネル長はハードウェアによって 32 ビットに固定されます。

**注 :** 正しい動作のためには、このビットは、I<sup>2</sup>S が無効のときに設定してください。

**SPI モードでは使用しません。**

## 30.9.9 SPIx\_I2S プリスケアラレジスタ (SPIx\_I2SPR)

アドレスオフセット : 0x20

リセット値 : 0000 0010 (0x0002)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV[7:0]							
						rw	rw	rw							

ビット 15:10 予約済み。ハードウェアによって 0 に固定されています。

ビット 9 **MCKOE** : マスタクロック出カインエーブル

- 0 : マスタクロック出力は無効です。
- 1 : マスタクロック出力は有効です。

**注 :** このビットは、I<sup>2</sup>S が無効なときに設定してください。これらのビットは、I<sup>2</sup>S がマスタモードのときのみ使用します。

**SPI モードでは使用しません。**

ビット 8 **ODD** : プリスケアラの奇数分周比

- 0 : 真のディバイダ値 = I2SDIV \* 2
- 1 : 真のディバイダ値 = (I2SDIV \* 2) + 1

[セクション 30.7.4 \(986 ページ\)](#) を参照してください。

**注 :** このビットは、I<sup>2</sup>S が無効なときに設定してください。これらのビットは、I<sup>2</sup>S がマスタモードのときのみ使用します。

**SPI モードでは使用しません。**

ビット 7:0 **I2SDIV[7:0]** : I<sup>2</sup>S リニアプリスケアラ

I2SDIV [7:0] = 0 または I2SDIV [7:0] = 1 は禁止されている値です。

[セクション 30.7.4 \(986 ページ\)](#) を参照してください。

**注 :** これらのビットは、I<sup>2</sup>S が無効なときに設定してください。これらのビットは、I<sup>2</sup>S がマスタモードのときのみ使用します。

**これらは SPI モードでは使用しません。**

## 30.9.10 SPI/I2S レジスタマップ

表 170 に、SPI/I2S レジスタマップとリセット値を示します。

表 170. SPI レジスタマップとリセット値

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	SPIx_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]		MSTR	CPOL	CPHA		
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	SPIx_CR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]			TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN		
	リセット値																		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	
0x08	SPIx_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FTLVL[1:0]	FRLVL[1:0]			FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE		
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	1	0	
0x0C	SPIx_DR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DR[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	SPIx_CRCPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CRCPOLY[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x14	SPIx_RXCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RxCRC[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	SPIx_TXCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TxCRC[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	SPIx_I2SCFGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	I2SMOD	I2SE	I2SCFG	PCMSYNC			Res	I2SSTD	CKPOL	DATLEN	CHLEN				
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0		
0x20	SPIx_I2SPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MCKOE	ODD	I2SDIV									
	リセット値																							0	0	0	0	0	0	0	0	0	1	0

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。



## 31 コントローラエリアネットワーク (bxCAN)

### 31.1 概要

**Basic Extended CAN** ペリフェラル (略称 **bxCAN**) は CAN ネットワークのインタフェースです。CAN プロトコルバージョン 2.0A および B をサポートしています。小さな CPU 負荷で多くの受信メッセージを効率よく取り扱えるように設計されています。また、送信メッセージの優先順位に関する要件も満たしています。

安全性が重視されるアプリケーションの場合、CAN コントローラは、CAN タイムトリガ通信オプションをサポートするすべてのハードウェア機能を提供します。

### 31.2 bxCAN の主な機能

- CAN プロトコルバージョン 2.0 A、B Active のサポート
- ビットレート：最大 1 Mbit/s
- タイムトリガ通信オプションのサポート

#### 送信

- 3 つの送信メールボックス
- 設定可能な送信優先順位
- SOF 送信時のタイムスタンプ

#### 受信

- 2 つの 3 ステージ受信 FIFO
- スケーラブルフィルタバンク：
  - 14 個のフィルタバンク
- 識別子リスト機能
- 設定可能な FIFO オーバーラン
- SOF 受信時のタイムスタンプ

#### タイムトリガ通信オプション

- 自動再送信モードの無効化
- 16 ビットのフリーランニングタイマ
- タイムスタンプを最後の 2 データバイトで送信

#### 管理

- マスク可能な割り込み
- 固有のアドレス空間でのソフトウェア効率の高いメールボックス配置

## 31.3 bxCAN の概要

今日の CAN アプリケーションでは、ネットワークのノード数が増加し、複数のネットワークがゲートウェイを介してリンクされることもあります。一般に、システム内の（そして各ノードが扱う）メッセージ数は大幅に増加しています。アプリケーションメッセージに加えて、ネットワーク管理や診断のメッセージも導入されています。

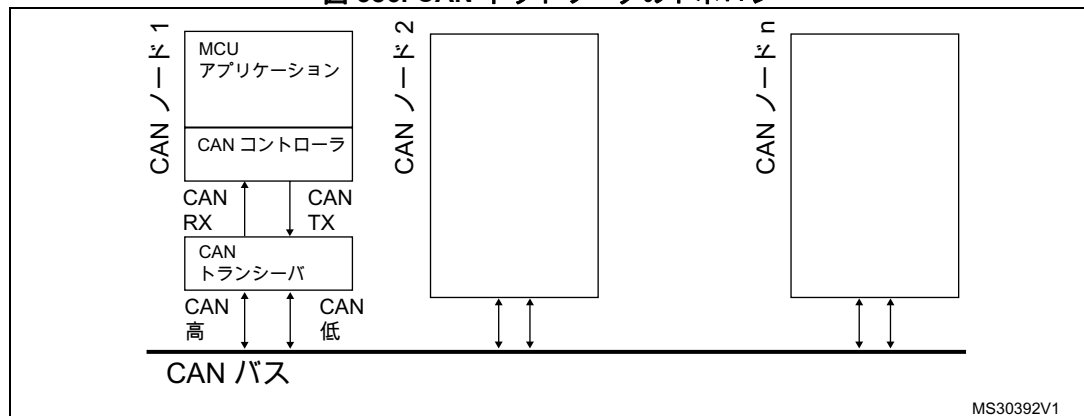
- 各種のメッセージを扱うには、高度なフィルタ機構が要求されます。

さらに、アプリケーションタスクはより多くの CPU 時間を必要とするため、メッセージ受信に起因するリアルタイム制約を減らす必要があります。

- 受信 FIFO 方式では、CPU は、メッセージを失うことなく長時間にわたってアプリケーションタスクに専念できます。

標準の CAN ドライバに基づく標準 HLP (Higher Layer Protocol : 上位層プロトコル) は、CAN コントローラとの効率的なインタフェースを必要とします。

図 386. CAN ネットワークのトポロジ



### 31.3.1 CAN 2.0B アクティブコア

bxCAN モジュールは、CAN メッセージの送受信を完全に自律的に処理します。標準識別子 (11 ビット) と拡張識別子 (29 ビット) は、ハードウェアによって完全にサポートされています。

### 31.3.2 制御レジスタ、ステータスレジスタ、設定レジスタ

アプリケーションは、これらのレジスタを以下の目的で使用します。

- CAN パラメータ (ポーレートなど) の設定
- 送信リクエスト
- 受信処理
- 割り込み管理
- 診断情報の入手

### 31.3.3 送信メールボックス

ソフトウェアがメッセージをセットアップするために、3 つの送信メールボックスが用意されています。最初に送信されるメールボックスは、送信スケジューラによって決定されます。

## 31.3.4 受信フィルタ

bxCAN は、受信メッセージからソフトウェアが必要なものを選び出し、他を破棄するために、14 個のスケラブルで設定可能な識別子フィルタバンクを備えています。

### 受信 FIFO

ハードウェアは、受信メッセージを格納するために 2 つの受信 FIFO を使用します。各 FIFO には、3 つの完全なメッセージを格納できます。FIFO の管理はすべてハードウェアによって行われます。

## 31.4 bxCAN 動作モード

bxCAN には、初期化、通常、およびスリープという 3 つの主な動作モードがあります。ハードウェアリセットの後、bxCAN は消費電力を低減するためにスリープモードに入り、CANTX では内部プルアップがアクティブとなります。ソフトウェアは、CAN\_MCR レジスタの INRQ または SLEEP ビットをセットすることによって、bxCAN に初期化またはスリープモードに入るように要求します。いずれかのモードに入ると、bxCAN は、CAN\_MSR レジスタの INAK または SLAK ビットをセットすることによってそれを確定し、内部プルアップは無効になります。INAK も SLAK もセットされていない場合、bxCAN は通常モードです。通常モードに入る前に、bxCAN は必ず CAN バスで同期する必要があります。同期のために、bxCAN は、CAN バスがアイドル状態になる（つまり、CANRX 上で 11 個の連続したリセツピビットが観測される）のを待ちます。

### 31.4.1 初期化モード

ハードウェアが初期化モードにある間に、ソフトウェアの初期化も行うことができます。このモードに入るには、ソフトウェアは CAN\_MCR レジスタの INRQ ビットをセットし、ハードウェアが CAN\_MSR レジスタの INAK ビットをセットしてこのリクエストを確認するのを待ちます。

初期化モードから出るには、ソフトウェアで INRQ ビットをクリアします。INAK ビットがハードウェアによってクリアされると、bxCAN は初期化モードを終了します。

初期化モードでは、CAN バスとの間のすべてのメッセージ転送は停止し、CAN バス出力 CANTX のステータスはリセツピ（ハイ）になります。

初期化モードに入っても、設定レジスタの内容は一切変化しません。

CAN コントローラを初期化するには、ソフトウェアでビットタイミング (CAN\_BTR) および CAN オプション (CAN\_MCR) レジスタの設定を行う必要があります。

CAN フィルタバンクに関連付けられたレジスタ (モード、スケール、FIFO 割当て、アクティベーション、およびフィルタ値) を初期化するには、ソフトウェアが FINIT ビット (CAN\_FMR) をセットする必要があります。フィルタの初期化は、初期化モード以外でも行うことができます。

**注：** FINIT=1 のとき、CAN 受信は無効化されます。

フィルタの値は、CAN\_FA1R レジスタの関連するフィルタアクティベーションビットを無効にすることによっても変更できます。

フィルタバンクを使用しない場合は、これを非アクティブのままにしておく (対応する FACT ビットをクリアしておく) ことを推奨します。

## 31.4.2 通常モード

初期化が終了すると、ソフトウェアはハードウェアに対し、CAN バスで同期して送受信を開始できるように通常モードに入るように要求する必要があります。

この通常モードに入るリクエストは、CAN\_MCR レジスタの INRQ ビットをクリアすることによって行われます。bxCAN は通常モードに入り、11 個の連続したリセッショビットの発生 (バスアイドル状態) を待って CAN バス上のデータ転送と同期すると、バス動作への参加準備完了となります。通常モードへの切り替えは、CAN\_MSR レジスタの INAK ビットをクリアすることによって、ハードウェアによって確認されます。

フィルタ値の初期化は初期化モードとは別に行いますが、フィルタがアクティブでない (対応する FACTx ビットがクリアされている) ときに行う必要があります。フィルタのスケールとモードの設定は、通常モードに入る前に行う必要があります。

## 31.4.3 スリープモード (低電力)

消費電力を低減するため、bxCAN にはスリープモードと呼ばれる低電力モードがあります。ソフトウェアのリクエストで CAN\_MCR レジスタの SLEEP ビットをセットすると、このモードに入ります。このモードでは bxCAN のクロックは停止しますが、ソフトウェアから bxCAN のメールボックスにアクセスすることは可能です。

bxCAN がスリープモードのときにソフトウェアが INRQ ビットをセットして初期化モードに入るように要求する場合には、SLEEP ビットもクリアする必要があります。

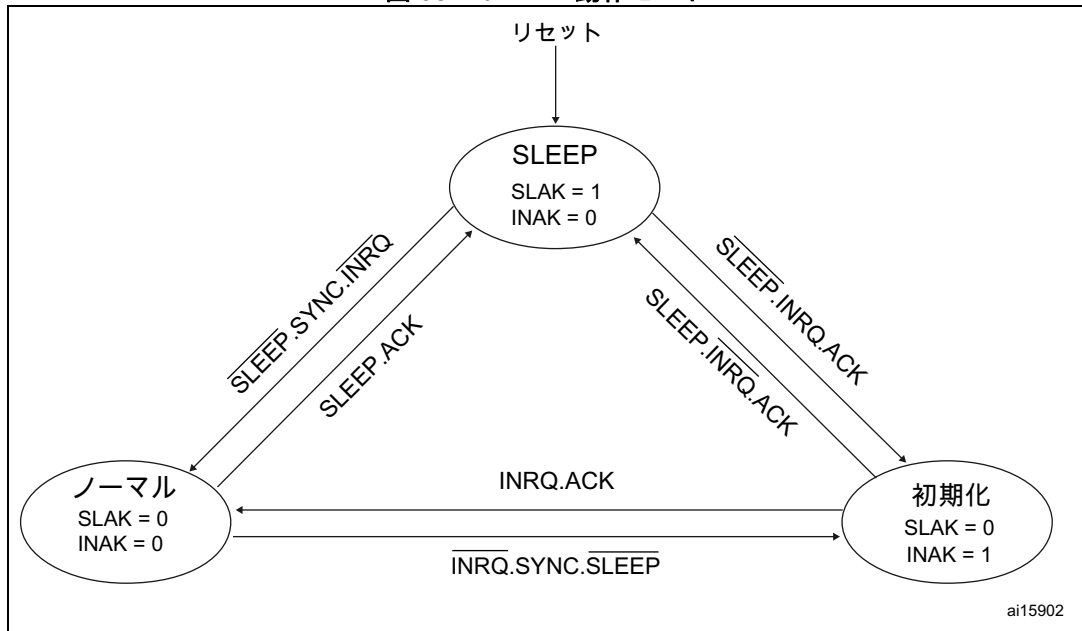
bxCAN がスリープモードからウェイクアップするには、ソフトウェアで SLEEP ビットをクリアするか、CAN バス動作を検出する必要があります。

CAN バス動作を検出すると、ハードウェアは、SLEEP ビットをクリアしてウェイクアップシーケンスを自動的に実行します (CAN\_MCR レジスタの AWUM ビットがセットされている場合)。AWUM ビットがクリアされている場合、スリープモードから出るにはウェイクアップ割り込み発生時にソフトウェアが SLEEP ビットをクリアする必要があります。

**注:** **ウェイクアップ割り込みが有効である (CAN\_IER レジスタの WKUIE ビットがセットされている) 場合には、bxCAN がウェイクアップシーケンスを自動的に実行する場合でも、CAN バス動作の検出時にウェイクアップ割り込みが生成されます。**

SLEEP ビットがクリアされた後、bxCAN が CAN バスに同期するとスリープモードを終了します (図 387: bxCAN 動作モードを参照)。SLAK ビットがハードウェアによってクリアされると、スリープモードを終了します。

図 387. bxCAN 動作モード



1. ACK = ハードウェアが CAN\_MSR レジスタの INAK または SLAK ビットをセットすることによってリクエストを確認する待ち状態。
2. SYNC = CAN バスがアイドル状態になる (つまり、11 個の連続したリセッパビットが CANRX で観測される) までの bxCAN の待ち状態。

## 31.5 テストモード

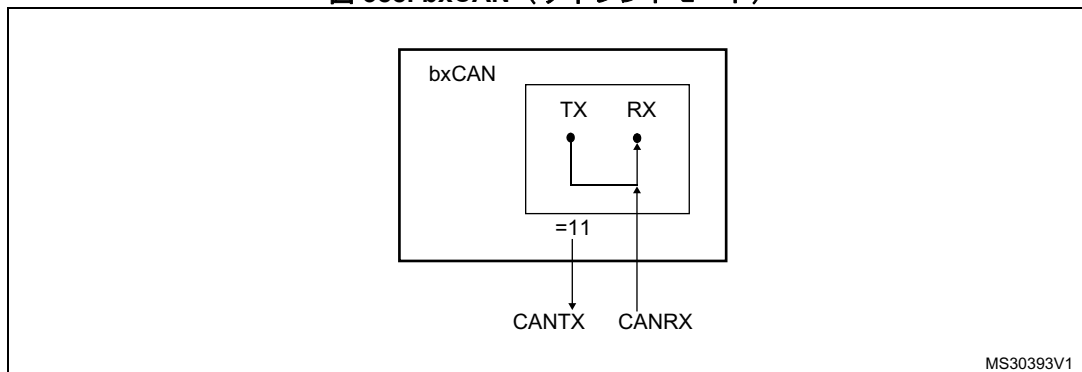
テストモードは、CAN\_BTR レジスタの SILM ビットと LBKM ビットによって選択できます。これらのビットは、bxCAN が初期化モードにある間に設定する必要があります。テストモードが選択された後で通常モードに戻るには、CAN\_MCR レジスタの INRQ ビットをリセットする必要があります。

### 31.5.1 サイレントモード

bxCAN をサイレントモードにするには、CAN\_BTR レジスタの SILM ビットをセットします。

サイレントモードでは、bxCAN は有効なデータフレームやリモートフレームを受信できますが、CAN バスではリセッパビットのみを送信するため、送信を開始することはできません。bxCAN がドミナントビット (ACK ビット、オーバロードフラグ、アクティブエラーフラグ) を送信する必要がある場合、CAN バスがリセッパ状態にとどまることがあっても CAN コアがこのドミナントビットを監視できるように、このビットは内部でルート変更されます。サイレントモードを使用すると、ドミナントビット (ACK ビット、エラーフレーム) の送信による影響を受けずに、CAN バス上のトラフィックを解析できます。

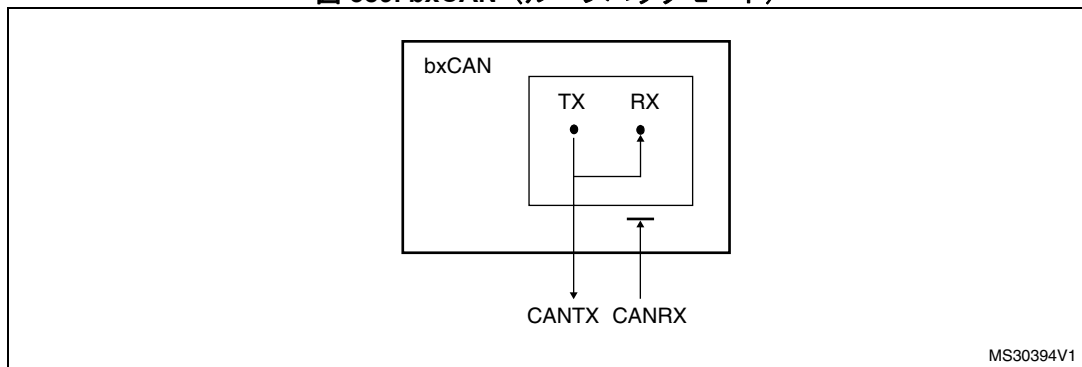
図 388. bxCAN (サイレントモード)



## 31.5.2 ループバックモード

bxCAN をループバックモードに設定するには、CAN\_BTR レジスタの LBKM ビットをセットします。ループバックモードでは、bxCAN は自分が送信したメッセージを受信メッセージとして扱い、それらを受信メールボックスに格納します（受信フィルタを通過した場合）。

図 389. bxCAN (ループバックモード)

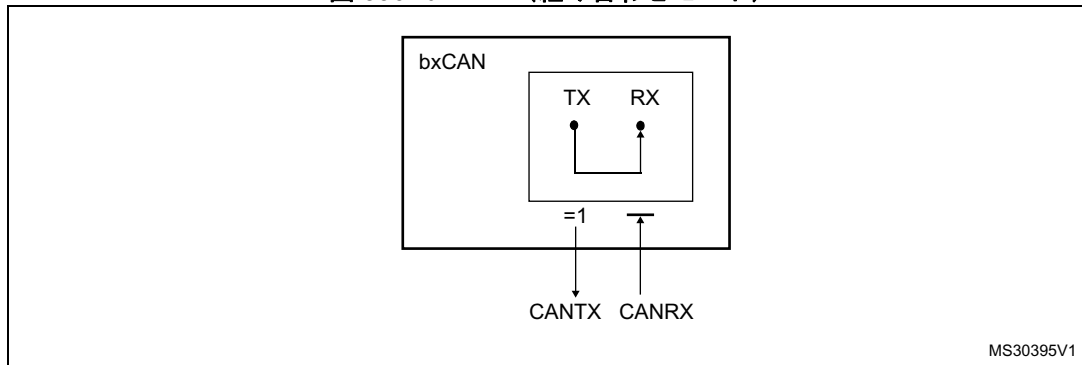


このモードは、セルフテスト機能のために用意されています。外部イベントから独立させるために、ループバックモードでは、CAN コアは ACK エラーを無視します（データ/リモートフレームの ACK スロットではドミナントビットはサンプリングされません）。このモードでは、bxCAN は Tx 出力から Rx 入力へ内部フィードバックを行います。bxCAN は、CANRX 入力ピンの実際の値を無視します。送信メッセージは CANTX ピンで監視できます。

## 31.5.3 サイレントモードとループバックモードの組み合わせ

CAN\_BTR レジスタの LBKM ビットと SILM ビットをセットすることによって、ループバックモードとサイレントモードを組み合わせることもできます。このモードは「ホットセルフテスト」に使用できます。つまり、CANTX および CANRX ピンに接続されている実行中の CAN システムに影響を与えずに、ループバックモードと同様に bxCAN をテストできます。このモードでは、CANRX ピンは bxCAN から切断され、CANTX ピンはリセツピに保持されます。

図 390. bxCAN (組み合わせモード)



## 31.6 デバッグモードでの挙動

マイクロコントローラがデバッグモードになると (Cortex-M4<sup>®</sup>F コア 停止)、次の状態に応じて、bxCAN は通常どおりに動作を続けるか、または停止します。

- CAN\_MCR の DBF ビット。詳細については、[セクション 31.9.2 : CAN 制御/ステータスレジスタ](#)を参照してください。

## 31.7 bxCAN 機能の説明

### 31.7.1 送信処理

メッセージを送信するには、アプリケーションは、**EMPTY** 状態の送信メールボックスを 1 つ選択し、識別子、データ長コード (DLC)、およびデータを設定してから、CAN\_TiRxR レジスタの対応する TXRQ ビットをセットすることによって、送信を要求する必要があります。メールボックスが **EMPTY** 状態でなくなると、ソフトウェアはこのメールボックスレジスタへの書き込みができなくなります。TXRQ ビットがセットされると、メールボックスはただちに **PENDING** 状態になり、最高優先順位のメールボックスになるまで待機します。[送信優先順位](#)を参照してください。メールボックスの優先順位が最高になると、メールボックスは送信の **SCHEDULED** 状態になります。スケジュール済みメールボックスのメッセージの送信は、CAN バスがアイドル状態になると開始されます (**TRANSMIT** 状態になります)。メールボックスの内容が正常に送信されると、そのメールボックスは再び **EMPTY** 状態になります。ハードウェアは、CAN\_TSR レジスタの RQCP ビットと TXOK ビットをセットして、送信が正常に終了したことを示します。

送信が失敗した場合、その原因がアービトレーションロストである場合は CAN\_TSR レジスタの ALST ビットによって、または、送信エラー検出である場合は TERR ビットによって示されます。

### 送信優先順位

識別子による優先順位

複数の送信メールボックスが保留中のときには、送信順序は、メールボックスに格納されているメッセージの識別子によって決まります。CAN プロトコルのアービトレーションに従って、識別子の値が最小のメッセージの優先順位が最高になります。識別子の値が同じ場合には、メールボックス番号の小さい方が優先的にスケジュールされます。

送信リクエストの順序による優先順位

送信メールボックスは、CAN\_MCR レジスタの TXFP ビットをセットすることによって、送信 FIFO として設定できます。このモードでは、優先順位は送信リクエストの順序によって決まります。

このモードは分割送信の場合に非常に役立ちます。

## アポート

ユーザは、CAN\_TSR レジスタの ABRQ ビットをセットすることによって、送信リクエストをアポートできます。**PENDING** または **SCHEDULED** 状態では、メールボックスはただちにアポートされます。メールボックスが **TRANSMIT** 状態のときにアポートリクエストが出されると、次の 2 つの結果につながります。メールボックスの送信が正常に行われた場合、CAN\_TSR レジスタの TXOK ビットがセットされ、メールボックスは **EMPTY** 状態になります。送信に失敗した場合、メールボックスは **SCHEDULED** 状態になり、送信がアポートされ、TXOK ビットがクリアされて、メールボックスは **EMPTY** 状態になります。いずれの場合にも、現在の送信が終了するとメールボックスは最低でも再び **EMPTY** 状態になります。

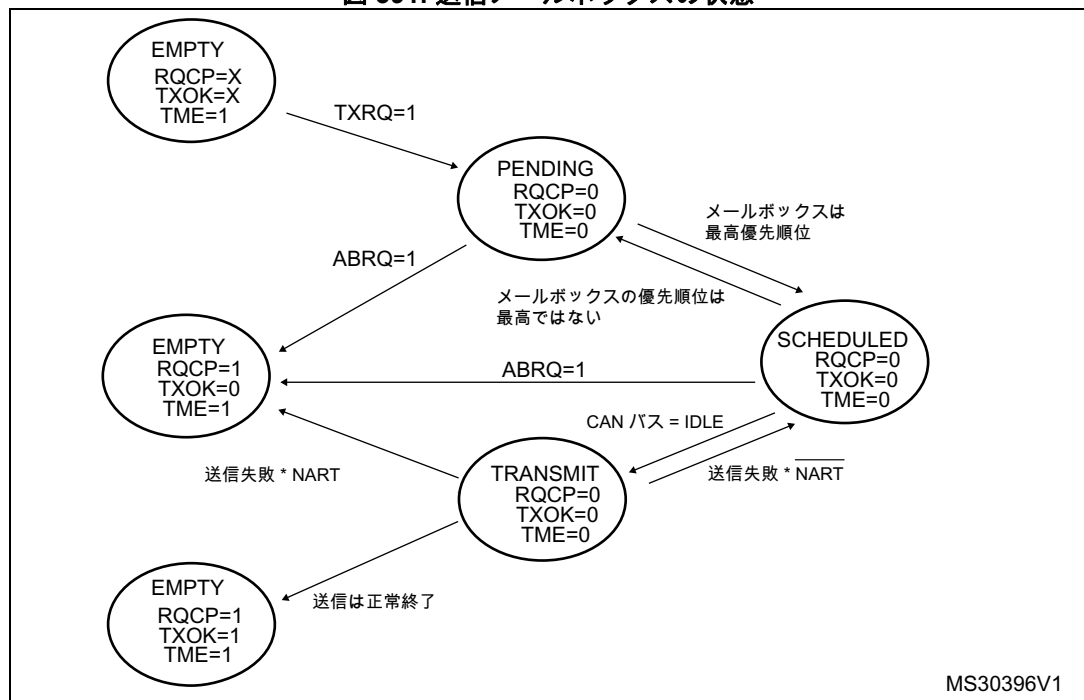
## 自動再送信禁止モード

このモードは、CAN 規格のタイムトリガ通信オプションの要件を満たすために実装されています。ハードウェアをこのモードに設定するには、CAN\_MCR レジスタの NART ビットをセットする必要があります。

このモードでは、個々の送信は一度だけ開始されます。最初の試みがアービトレーションロスやエラーのために失敗した場合、ハードウェアはメッセージ送信を自動的にリスタートしません。

最初の送信試行の終わりに、ハードウェアはそのリクエストが完了したものとみなし、CAN\_TSR レジスタの RQCP ビットをセットします。送信の結果は、CAN\_TSR レジスタの TXOK、ALST、および TERR ビットによって示されます。

図 391. 送信メールボックスの状態





## 31.7.2 タイムトリガ通信モード

このモードでは、CAN ハードウェアの内部カウンタがアクティブになり、このカウンタを使用して受信/送信メールボックスの CAN\_RDTxR/CAN\_TDTxR レジスタに格納されるタイムスタンプ値を生成します。内部カウンタは、CAN ビット時間ごとにインクリメントされます ([セクション 31.7.7: ビットタイミング](#)を参照)。内部カウンタは、送受信ともに、フレーム開始 (SOF) ビットのサンプルポイントでキャプチャされます。

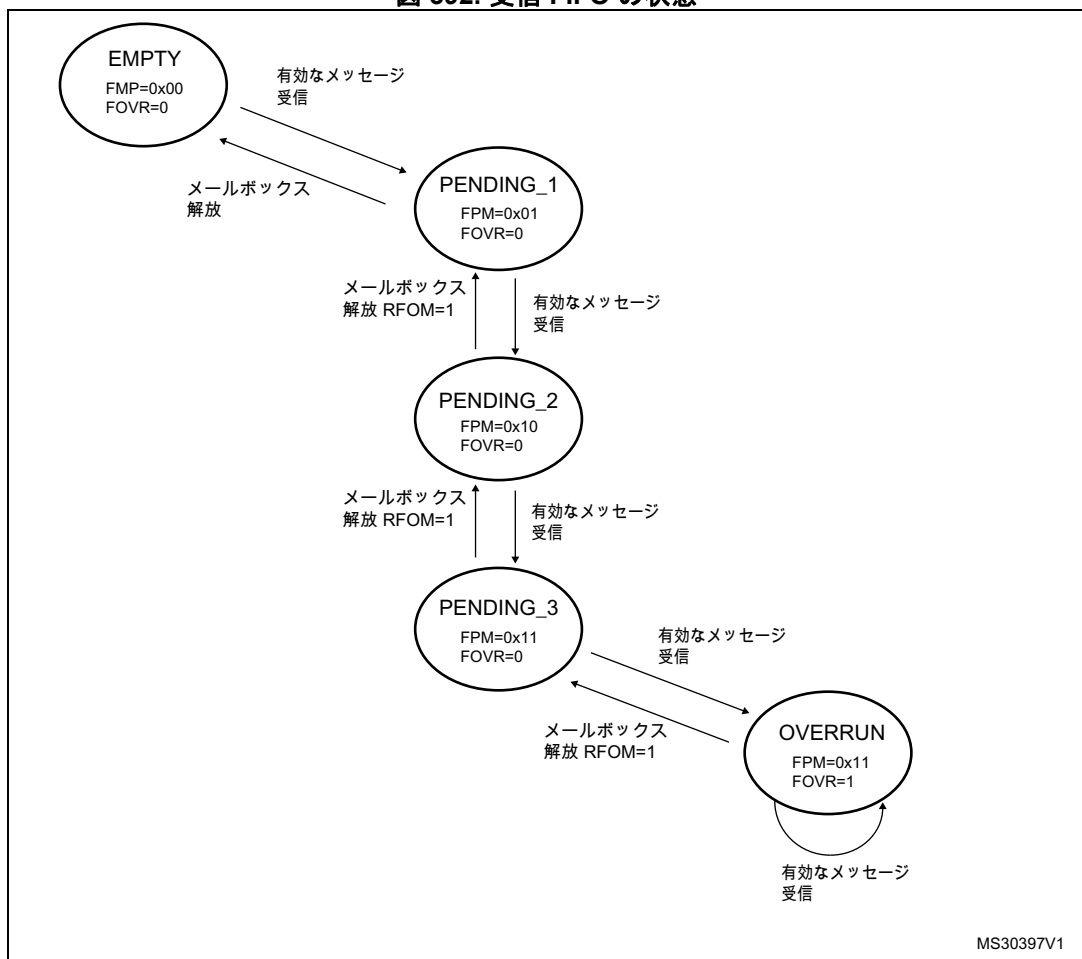
## 31.7.3 受信処理

CAN メッセージの受信には、3 つのメールボックスが FIFO として用意されています。CPU 負荷を軽減し、ソフトウェアを単純化し、データの一貫性を保証するために、FIFO の管理はすべてハードウェアによって行われます。アプリケーションは、FIFO 出力メールボックスを通じて、FIFO に格納されたメッセージにアクセスします。

### 有効なメッセージ

受信メッセージは、CAN プロトコルに従って正しく受信され (EOF フィールドの最後から 2 番目のビットまでエラーなし)、かつ、識別子フィルタリングを正常に通過した場合に、有効であるとみなされます ([セクション 31.7.4: 識別子フィルタリング](#)を参照)。

図 392. 受信 FIFO の状態



## FIFO 管理

**EMPTY** 状態から始まり、受信した最初の有効なメッセージが FIFO に格納されると、FIFO は **PENDING\_1** 状態になります。ハードウェアは、CAN\_RFR レジスタの FMP[1:0] ビットを値 01b に設定して、このイベントを通知します。このメッセージは FIFO 出力メールボックスからアクセスできます。ソフトウェアは、このメールボックスの内容を読み出し、CAN\_RFR レジスタの RFOM ビットをセットすることによってメールボックスを解放します。FIFO は再び **EMPTY** 状態になります。その間に新しい有効なメッセージを受信すると、FIFO は **PENDING\_1** 状態にとどまり、出力メールボックスから新しいメッセージにアクセスできます。

アプリケーションがメールボックスを解放しない場合には、次の有効なメッセージが FIFO に格納され、FIFO は **PENDING\_2** 状態 (FMP[1:0] = 10b) になります。この格納プロセスは、次の有効なメッセージに対しても繰り返され、FIFO は **PENDING\_3** 状態 (FMP[1:0] = 11b) になります。この時点で、メールボックスが次の有効なメッセージを格納できるように、ソフトウェアは、RFOM ビットをセットして、出力メールボックスを解放する必要があります。解放しないと、受信した次の有効なメッセージが失われます。

[セクション 31.7.5 : メッセージストレージ](#)も参照してください。

## オーバーラン

FIFO が **PENDING\_3** 状態 (すなわち、3 個のメールボックスが満杯) になると、次の有効なメッセージの受信で **OVERRUN** 状態となり、メッセージが失われます。ハードウェアは、CAN\_RFR レジスタの FOVR ビットをセットすることによって、オーバーラン状態を通知します。どのメッセージが失われるかは、FIFO の設定に依存します。

- FIFO ロック機能が無効 (CAN\_MCR レジスタの RFLM ビットがクリア) である場合、最後に FIFO に格納されたメッセージが新しい受信メッセージによって上書きされます。この場合、アプリケーションは常に最新のメッセージにアクセスできます。
- FIFO ロック機能が有効 (CAN\_MCR レジスタの RFLM ビットがセット) である場合、最新のメッセージが破棄され、ソフトウェアは FIFO 内にある 3 個の最も古いメッセージにアクセスできます。

## 受信関連の割り込み

メッセージが FIFO に格納されると、FMP[1:0] ビットが更新され、割り込みリクエストが生成されます (CAN\_IER レジスタの FMPIE ビットがセットされている場合)。

FIFO が満杯になると (3 番目のメッセージが格納されると)、CAN\_RFR レジスタの FULL ビットがセットされ、割り込みが生成されます (CAN\_IER レジスタの FFIE ビットがセットされている場合)。

オーバーラン状態になると、FOVR ビットがセットされ、割り込みが生成されます (CAN\_IER レジスタの FOVIE ビットがセットされている場合)。

## 31.7.4 識別子フィルタリング

CAN プロトコルでは、メッセージの識別子は、ノードのアドレスには関連付けられておらず、メッセージの内容に関連したものとなっています。したがって、トランスミッタは、そのメッセージをすべてのレシーバにブロードキャスト (一斉送信) します。メッセージを受信すると、受信ノードは、識別子の値に基づいてこのメッセージがソフトウェアに必要なものかどうかを判断します。このメッセージが必要な場合は、SRAM にコピーされます。そうでない場合、このメッセージは、ソフトウェアによる介入なしに破棄される必要があります。

この要件を満たすため、bxCAN コントローラは 28 個の設定可能でスケラブルなフィルタバンク (27-0) をアプリケーションに提供します。それ以外のデバイスでは、ソフトウェアに必要なメッセー

ジのみを受信できるように、bxCAN コントローラは 14 個の設定可能でスケーラブルなフィルタバンク (13-0) をアプリケーションに提供します。このハードウェアフィルタリングによって、ソフトウェアでフィルタリングを行う場合に必要となる CPU リソースを節約できます。各フィルタバンク x は、2 個の 32 ビットレジスタ (CAN\_FxR0 と CAN\_FxR1) で構成されています。

## スケーラブルな幅

フィルタをアプリケーションニーズに合わせて最適化/適応するため、各フィルタバンクは個別に拡大/縮小できます。フィルタのスケールに応じて、フィルタバンクは以下のフィルタを提供します。

- 1 個の 32 ビットフィルタ : STDID[10:0]、EXTID[17:0]、IDE、および RTR ビット用。
- 2 個の 16 ビットフィルタ : STDID[10:0]、RTR、IDE、および EXTID[17:15] ビット用。

 393 を参照してください。

さらに、これらのフィルタはマスクモードまたは識別子リストモードに設定できます。


## マスクモード

マスクモードでは、識別子レジスタは、識別子のどのビットを「一致が必要」として扱い、どのビットを「無視」として扱うかを指定するマスクレジスタに関連付けられます。

## 識別子リストモード

識別子リストモードでは、マスクレジスタが識別子レジスタとして使用されます。このように、識別子とマスクを定義する代わりに、2 つの識別子を指定することで、単一識別子の数を倍にします。受信識別子のすべてのビットが、フィルタレジスタで指定されたビットと一致する必要があります。

## フィルタバンクのスケールとモードの設定

フィルタバンクは、対応する CAN\_FMR レジスタによって設定されます。フィルタバンクを設定するには、CAN\_FAR レジスタの FACT ビットをクリアしてフィルタバンクを無効にする必要があります。フィルタのスケールは、CAN\_FS1R レジスタの対応する FSCx ビットによって設定されます ( 393 を参照)。対応するマスク/識別子レジスタの識別子リストまたは識別子マスクモードは、CAN\_FMR レジスタの FBMx ビットによって設定されます。

識別子のグループをフィルタするには、マスクモードでマスク/識別子レジスタを設定します。

単一識別子を選択するには、識別子リストモードでマスク/識別子レジスタを設定します。

アプリケーションで使用されないフィルタは、非アクティブのままにしておきます。

フィルタバンク内の各フィルタには、各フィルタバンクのモードとスケールに応じて 0 から最大値までの番号 (フィルタ番号) が付けられます。

フィルタの設定については  393 を参照してください。

図 393. フィルタバンクのスケール設定 - レジスタ構造



## 該当フィルタインデックス

メッセージは、FIFO で受信するとアプリケーションからアクセス可能になります。通常、アプリケーションのデータは SRAM のいずれかの位置にコピーされます。データを正しい位置にコピーするには、アプリケーションが識別子によってデータを識別する必要があります。この操作を省き、SRAM へのアクセスを容易にするために、CAN コントローラは該当フィルタインデックス (Filter Match Index) を搭載しています。

このインデックスは、フィルタの優先順位規則に従って、メッセージとともにメールボックスに格納されます。したがって、受信した各メッセージには関連付けられた該当フィルタインデックスがあります。

該当フィルタインデックスは、次の 2 つの方法で使用できます。

- 該当フィルタインデックスと予期される値のリストを比較します。
- 該当フィルタインデックスを配列のインデックスとして使用し、データの格納先にアクセスします。

マスクされていないフィルタについては、ソフトウェアが識別子を比較する必要はありません。

フィルタがマスクされている場合、ソフトウェアは、マスクされたビットのみを比較します。

フィルタ番号のインデックス値では、フィルタバンクのアクティブ状態を考慮に入れません。さらに、FIFO ごとに 1 つずつ、2 つの独立した番号付け方式を使用します。図 394 の例を参照してください。

図 394. フィルタ番号の例

フィルタバンク	FIFO0	フィルタ番号	フィルタバンク	FIFO1	フィルタ番号
0	ID リスト (32 ビット)	0 1	2	ID マスク (16 ビット)	0 1
1	ID マスク (32 ビット)	2	4	ID リスト (32 ビット)	2 3
3	ID リスト (16 ビット)	3 4 5 6	7	非アクティブ ID リスト (16 ビット)	4 5
5	非アクティブ ID リスト (32 ビット)	7 8	8	ID マスク (16 ビット)	6 7
6	ID マスク (16 ビット)	9 10	10	非アクティブ ID リスト (16 ビット)	8 9 10 11
9	ID リスト (32 ビット)	11 12	11	ID リスト (32 ビット)	12 13
13	ID マスク (32 ビット)	13	12	ID マスク (32 ビット)	14

ID = 識別子

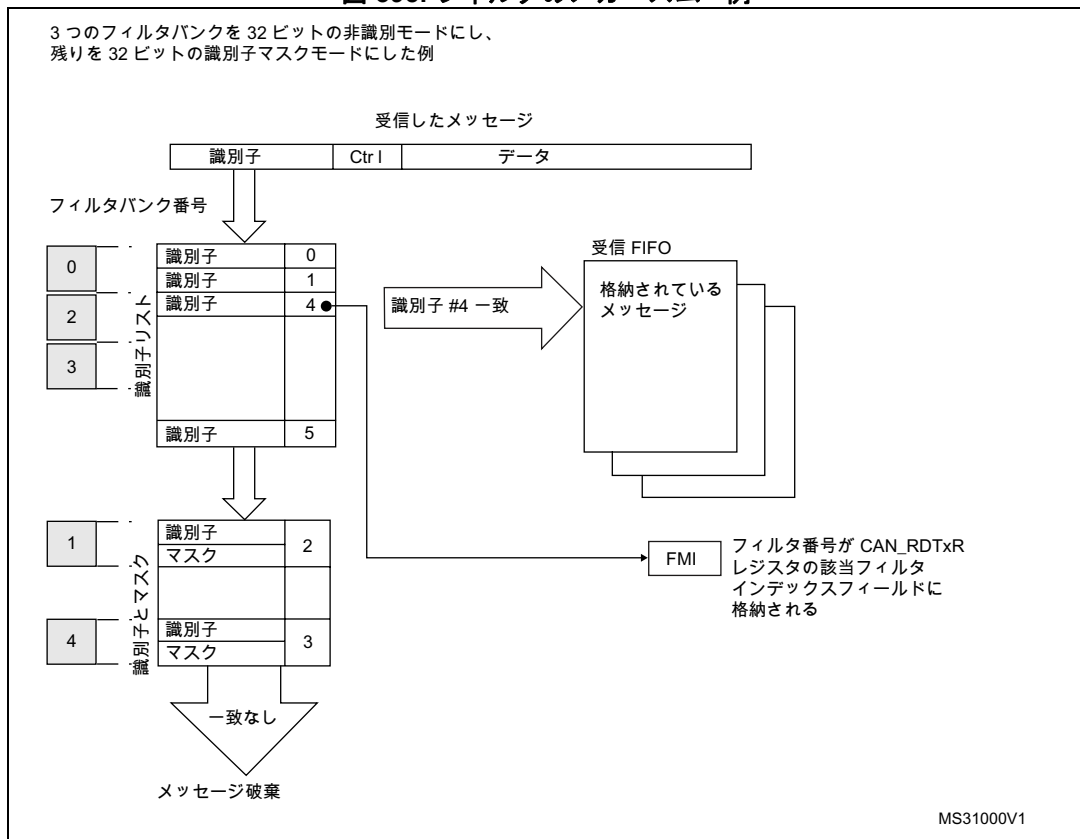
MS30399V1

## フィルタの優先順位規則

フィルタの組み合わせによっては、1 つの識別子が複数のフィルタを正常に通過することがあります。この場合、受信メールボックスに格納される該当フィルタ値は、次の優先順位規則に基づいて選択されます。

- 32 ビットフィルタは 16 ビットフィルタよりも優先されます。
- 同じスケールのフィルタでは、識別子マスクモードよりも識別子リストモードが優先されます。
- スケールとモードが同じフィルタでは、フィルタ番号によって優先順位が与えられます (番号が小さいほど優先順位が高くなります)。

図 395. フィルタのメカニズム - 例



上の例は bxCAN のフィルタリングの原理を示しています。メッセージを受信すると、その識別子は、まず識別子リストモードで設定されたフィルタと比較されます。一致する項目があれば、メッセージは関連する FIFO に格納され、一致したフィルタのインデックスが該当フィルタインデックスに格納されます。この例では、識別子は識別子 #4 と一致するため、メッセージの内容と FMI 2 が FIFO に格納されます。

一致する項目がなければ、受信メッセージの識別子は、次にマスクモードで設定されたフィルタと比較されます。

この識別子がフィルタ内のどの識別子とも一致しない場合、メッセージはハードウェアによって破棄され、ソフトウェアが介入することはありません。

## 31.7.5 メッセージストレージ

CAN メッセージに関するソフトウェアとハードウェアのインターフェースは、メールボックスによって実装されます。メールボックスには、識別子、データ、制御、ステータス、タイムスタンプ情報など、メッセージに関するすべての情報が格納されます。

### 送信メールボックス

ソフトウェアは、送信しようとするメッセージを空の送信メールボックスにセットアップします。送信のステータスは、ハードウェアによって CAN\_TSR レジスタに示されます。

表 171. 送信メールボックスの配置

送信メールボックスのベースアドレスのオフセット	レジスタ名
0	CAN_TlRxR
4	CAN_TDTxR
8	CAN_TDLxR
12	CAN_TDHxR

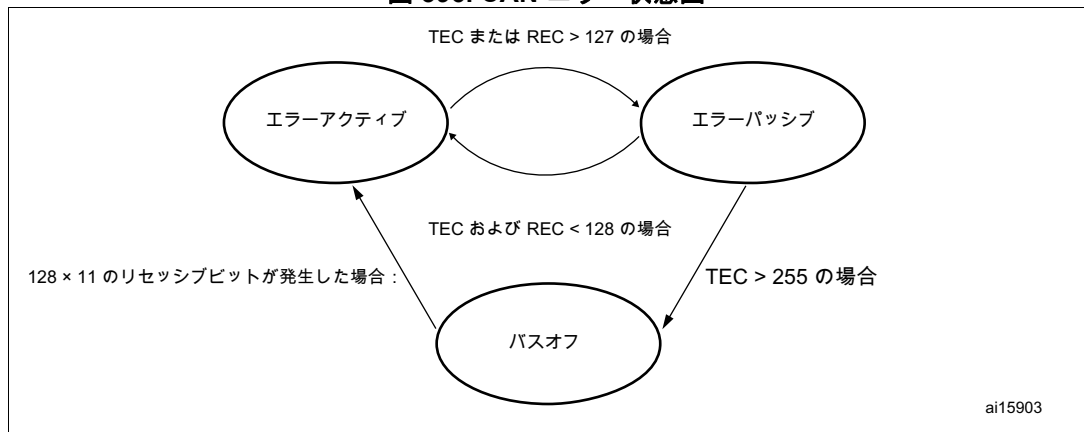
## 受信メールボックス

メッセージを受信すると、ソフトウェアは FIFO 出力メールボックスを通してそのメッセージにアクセスできます。ソフトウェアが読み出しなどのメッセージの処理を終えると、ソフトウェアは、CAN\_RFR レジスタの RFOM ビットによって FIFO 出力メールボックスを解放し、次の受信メッセージにアクセスできるようにする必要があります。該当フィルタインデックスは、CAN\_RDTxR レジスタの MFMI フィールドに格納されます。16 ビットタイムスタンプの値は、CAN\_RDTxR レジスタの TIME[15:0] フィールドに格納されます。

表 172. 受信メールボックスの配置

受信メールボックスのベースアドレスのオフセット (バイト)	レジスタ名
0	CAN_RlRxR
4	CAN_RDTxR
8	CAN_RDLxR
12	CAN_RDHxR

図 396. CAN エラー状態図



## 31.7.6 エラー管理

CAN プロトコルに記述されているエラー管理は、エラー条件によってインクリメント/デクリメントされる送信エラーカウンタ (CAN\_ESR レジスタの TEC 値) と受信エラーカウンタ (CAN\_ESR レジスタの REC 値) を使用して、すべてハードウェアによって処理されます。TEC および REC の管理の詳細については、CAN 規格を参照してください。

ソフトウェアは、この 2 つのカウンタを読み出してネットワークの安定性を判断できます。さらに、CAN ハードウェアは、現在のエラーステータスに関する詳細な情報を CAN\_ESR レジスタで提供します。CAN\_IER レジスタ (ERRIE ビットなど) によって、ソフトウェアは、エラー検出時の割り込み生成をきわめて柔軟に設定できます。

### バスオフからの復帰

TEC が 255 を超えるとバスオフ状態になります。この状態は CAN\_ESR レジスタの BOFF ビットによって示されます。バスオフ状態では、bxCAN はメッセージの送受信を行うことができません。

CAN\_MCR レジスタの ABOM ビットに応じて、bxCAN は、自動的に、またはソフトウェアのリクエストに基づいて、バスオフ状態から復帰します (再びエラーアクティブになります)。しかし、いずれの場合も、bxCAN は、少なくとも CAN 規格で定められた復帰シーケンス (CANRX で観測される 11 個の連続したリセッショビットの発生 128 回) の間は待機する必要があります。

ABOM がセットされると、bxCAN は、バスオフ状態に入った後で自動的に復帰シーケンスを開始します。

ABOM がクリアされている場合、ソフトウェアは、bxCAN に対して初期化モードに入って出るようにリクエストして、復帰シーケンスを開始する必要があります。

**注:** *初期化モードでは、bxCAN は CANRX 信号を監視しないため、復帰シーケンスを完了できません。復帰するには、bxCAN は通常モードでなければなりません。*

## 31.7.7 ビットタイミング

ビットタイミング回路は、シリアルバスラインを監視し、スタートビットエッジでの同期と後続エッジでの再同期によって、サンプリングとサンプリングポイントの調整を行います。

この動作は、公称ビット時間を次のように 3 つのセグメントに分割するとわかりやすく説明できます。

- **同期セグメント (SYNC\_SEG)**: この時間セグメント内にビット変化が生じることが想定されます。この時間の長さは 1 タイムクオンタム ( $1 \times t_q$ ) に固定されています。
- **ビットセグメント 1 (BS1)**: サンプルポイントの位置を定義します。これには、CAN 規格の PROP\_SEG と PHASE\_SEG1 が含まれます。セグメントの時間は 1~16 タイムクオンタムの間でプログラミング可能ですが、ネットワークのさまざまなノード間の周波数の差に起因する正の位相ドリフトを補正するために、自動的に延長されることがあります。
- **ビットセグメント 2 (BS2)**: 送信ポイントの位置を定義します。これは、CAN 規格の PHASE\_SEG2 に相当します。セグメントの時間は 1~8 タイムクオンタムの間でプログラミング可能ですが、負の位相ドリフトを補正するために、自動的に短縮されることもあります。

再同期ジャンプ幅 (SJW) は、ビットセグメントの延長/短縮量の上限を定め、1~4 タイムクオンタムの間でプログラミング可能です。

有効なエッジとは、コントローラ自体はリセッショビットを送信しないという条件で、あるビット時間において、ドミナントからリセッシブへのバスレベルの最初の遷移と定義されます。

有効なエッジが SYNC\_SEG ではなく BS1 で検出された場合、サンプルポイントが遅くなるように、BS1 は最大で SJW 延長されます。



逆に、有効なエッジが SYNC\_SEG ではなく BS2 で検出された場合、送信ポイントが早くなるように、BS2 は最大で SJW 短縮されます。

プログラミングエラーに対する安全対策として、ビットタイミングレジスタ (CAN\_BTR) は、デバイスが STANDBY モードの場合にのみ設定可能です。

**注：** CAN ビットタイミングと再同期メカニズムの詳細については、ISO11898 規格を参照してください。

図 397. ビットタイミング

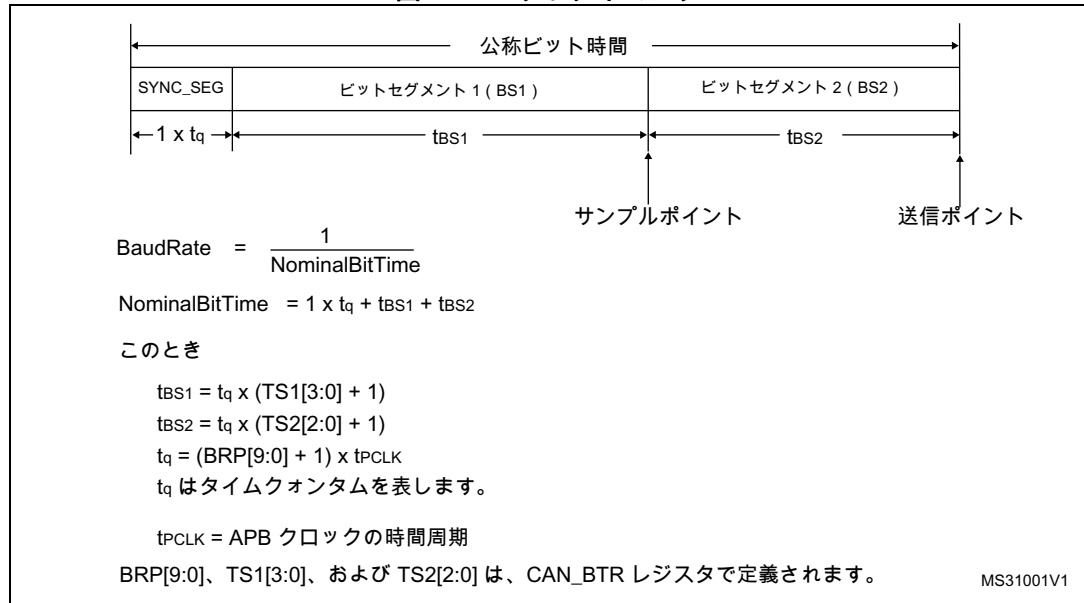
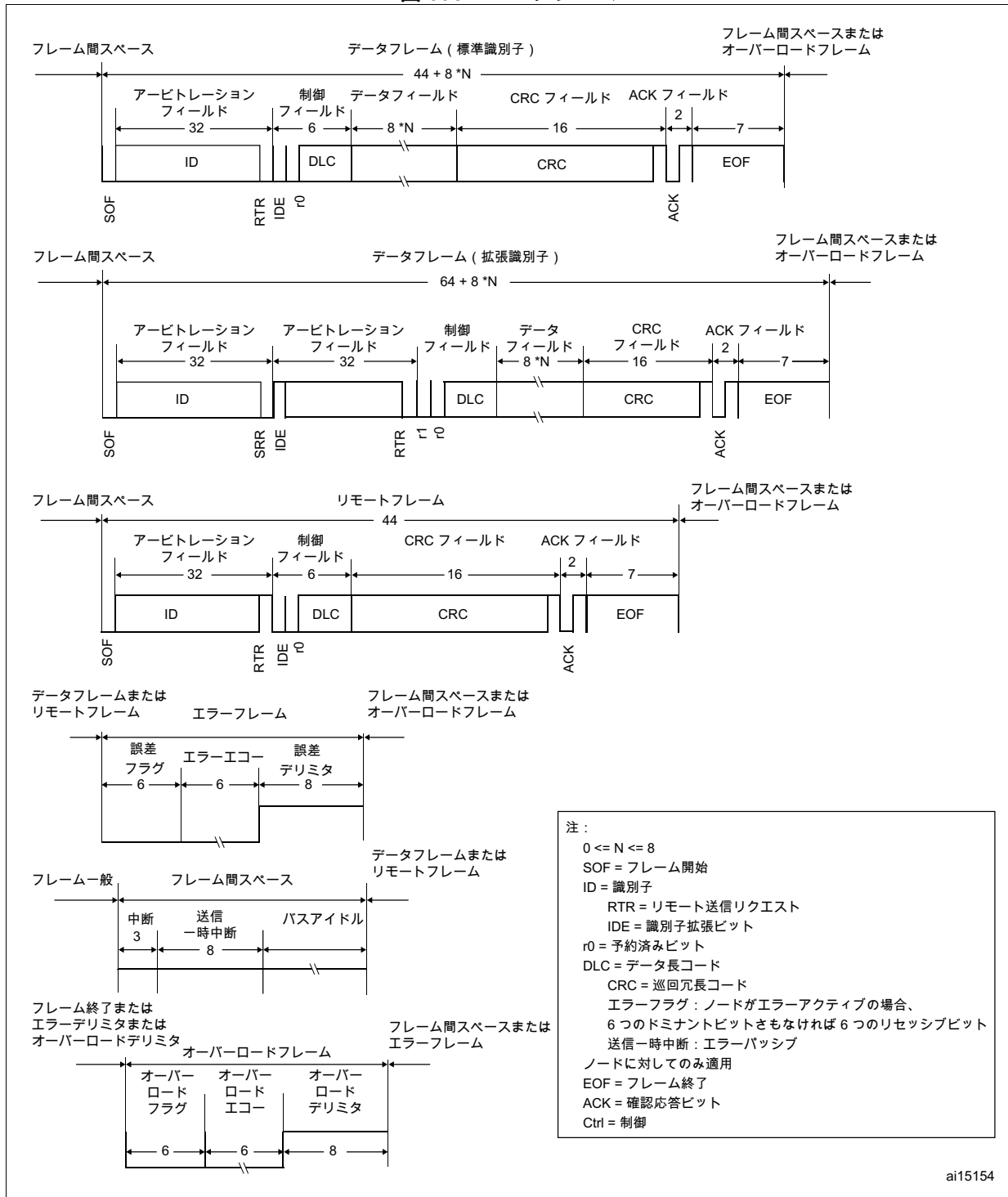


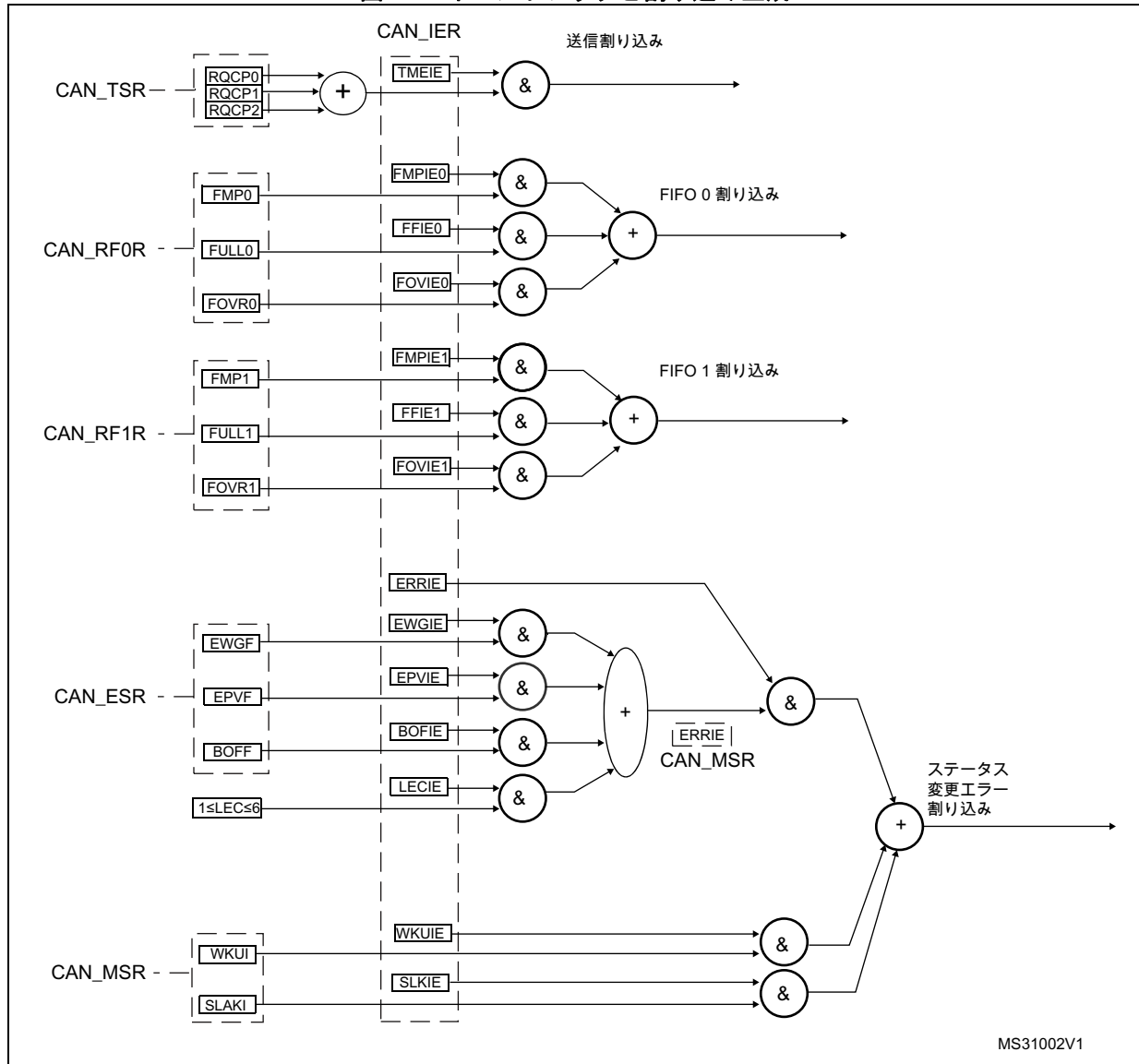
図 398. CAN フレーム



## 31.8 bxCAN の割り込み

bxCAN には 4 つの割り込みベクタが確保されています。各割り込みソースは、CAN 割り込み有効レジスタ (CAN\_IER) によって個別に有効/無効にできます。

図 399. イベントフラグと割り込み生成



- **送信割り込み**は次のイベントによって生成できます。
  - 送信メールボックス 0 が空になり、CAN\_TSR レジスタの RQCP0 ビットがセットされません。
  - 送信メールボックス 1 が空になり、CAN\_TSR レジスタの RQCP1 ビットがセットされません。
  - 送信メールボックス 2 が空になり、CAN\_TSR レジスタの RQCP2 ビットがセットされません。
- **FIFO 0 割り込み**は次のイベントによって生成できます。
  - 新しいメッセージの受信、CAN\_RF0R レジスタの FMP0 ビットが“00”ではない場合。
  - FIFO0 の満杯状態、CAN\_RF0R レジスタの FULL0 ビットがセットされる場合。
  - FIFO0 のオーバーラン状態、CAN\_RF0R レジスタの FOVR0 ビットがセットされる場合。
- **FIFO 1 割り込み**は次のイベントによって生成できます。
  - 新しいメッセージの受信、CAN\_RF1R レジスタの FMP1 ビットが“00”ではない場合。
  - FIFO1 の満杯状態、CAN\_RF1R レジスタの FULL1 ビットがセットされる場合。
  - FIFO1 のオーバーラン状態、CAN\_RF1R レジスタの FOVR1 ビットがセットされる場合。
- **エラー/ステータス変更割り込み**は次のイベントによって生成できます。
  - エラー条件：エラー条件の詳細については、CAN エラーステータスレジスタ (CAN\_ESR) を参照してください。
  - ウェイクアップ条件：CAN Rx 信号で SOF が観測されたとき。
  - スリープモードへの遷移。

## 31.9 CAN レジスタ

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

### 31.9.1 レジスタのアクセス保護

特定の設定レジスタに誤ったアクセスを行うと、ハードウェアが CAN ネットワーク全体の動作を一時的に妨害することがあります。このため、ソフトウェアが CAN\_BTR レジスタを変更できるのは、CAN ハードウェアが初期化モードのときに限られます。

誤ったデータを送信しても CAN ネットワークレベルでは問題は生じませんが、アプリケーションには重大な影響を与える可能性があります。ソフトウェアが送信メールボックスを変更できるのは、メールボックスが EMPTY 状態のときに限られます。図 391: 送信メールボックスの状態を参照してください。

フィルタ値を変更するには、関連するフィルタバンクを非アクティブにするか、FINIT ビットをセットします。さらに、CAN\_FMR、CAN\_FSR、および CAN\_FFR レジスタにあるフィルタ設定 (スケール、モード、および FIFO 割当て) の変更は、CAN\_FMR レジスタでフィルタ初期化モードがセットされている (FINIT = 1) 場合に限られます。

## 31.9.2 CAN 制御／ステータスレジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 2.1](#) を参照してください。

### CAN マスタ制御レジスタ (CAN\_MCR)

アドレスオフセット : 0x00

リセット値 : 0x0001 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBF
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ
rs								rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **DBF** : デバッグ凍結

0 : デバッグ時に CAN が機能します。

1 : デバッグ時に CAN の送受信機能が凍結されます。受信 FIFO は、まだ正常にアクセス／制御できます。

ビット 15 **RESET** : bxCAN ソフトウェアマスタリセット

0 : 通常動作。

1 : bxCAN のマスタリセットを強制します。リセット後は、スリープモードがアクティブになります (FMP ビットと CAN\_MCR レジスタはリセット値に初期化されます)。このビットは自動的に 0 にリセットされます。

ビット 14:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TTCM** : タイムトリガ通信モード

0 : タイムトリガ通信モードは無効です。

1 : タイムトリガ通信モードは有効です。

**注 :** [タイムトリガ通信モードの詳細については、セクション 31.7.2 : タイムトリガ通信モードを参照してください。](#)

ビット 6 **ABOM** : 自動バスオフ管理

このビットは、バスオフ状態から出るときの CAN ハードウェアの動作を制御します。

0 : 11 個のリセツピットが 128 回観測され、ソフトウェアが CAN\_MCR レジスタの INRQ ビットをまずセットし、クリアすると、ソフトウェアリクエストでバスオフ状態から出ます。

1 : 11 個のリセツピットが 128 回観測されると、ハードウェアによって自動的にバスオフ状態から出ます。

バスオフ状態の詳細については、[セクション 31.7.6 : エラー管理](#)を参照してください。

ビット 5 **AWUM** : 自動ウェイクアップモード

このビットは、スリープモードでのメッセージ受信時の CAN ハードウェアの動作を制御します。

0 : ソフトウェアリクエストで CAN\_MCR レジスタの SLEEP ビットをクリアしてスリープモードから出ます。

1 : CAN メッセージの検出時にハードウェアによって自動的にスリープモードから出ます。

CAN\_MCR レジスタの SLEEP ビットと CAN\_MSR レジスタの SLAK ビットはハードウェアによってクリアされます。

**ビット 4 NART** : 自動再送信禁止

- 0 : CAN ハードウェアは、CAN 規格に基づいてメッセージを正常に送信するまで、メッセージを自動的に再送信します。
- 1 : 送信結果 (正常、エラー、アービトレーションロスト) にかかわらず、メッセージは 1 回だけ送信されます。

**ビット 3 RFLM** : 受信 FIFO ロックモード

- 0 : 受信 FIFO はオーバーラン時にロックされません。受信 FIFO が満杯になると、次の受信メッセージは前のメッセージに上書きされます。
- 1 : 受信 FIFO はオーバーランに対してロックされます。受信 FIFO が満杯になると、次の受信メッセージは破棄されます。

**ビット 2 TXFP** : 送信 FIFO 優先順位

- このビットは、複数のメールボックスが同時にペンディング状態のときの送信順序を制御します。
- 0 : 優先順位はメッセージの識別子によって決まります。
- 1 : 優先順位はリクエスト順に (時系列に) 決まります。

**ビット 1 SLEEP** : スリープモードリクエスト

- このビットは、CAN ハードウェアがスリープモードに入るように要求するために、ソフトウェアによってセットされます。現在の CAN 動作 (CAN フレームの送受信) が完了すると、すぐにスリープモードに入ります。
- ソフトウェアによってこのビットがクリアされると、スリープモードを終了します。
- AWUM ビットがセットされていて、CAN Rx 信号で SOF ビットが検出されると、このビットはハードウェアによってクリアされます。
- このビットはリセット後にセットされます。CAN はスリープモードで開始します。

**ビット 0 INRQ** : 初期化リクエスト

- ソフトウェアによってこのビットがクリアされると、ハードウェアが通常モードに切り替わります。Rx 信号で 11 個の連続したリセツピットが観測されると、CAN ハードウェアが同期し、送受信の準備ができます。ハードウェアは、CAN\_MSR レジスタの INAK ビットをクリアすることによって、このイベントを通知します。
- このビットは、CAN ハードウェアに初期化モードに入るようにリクエストするために、ソフトウェアによってセットされます。ソフトウェアが INRQ ビットをセットすると、CAN ハードウェアは現在の CAN 動作 (送受信) が完了するのを待ってから、初期化モードに入ります。ハードウェアは、CAN\_MSR レジスタの INAK ビットをセットすることによって、このイベントを通知します。

## CAN マスタステータスレジスタ (CAN\_MSR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0C02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	SLAKI	WKUI	ERRI	SLAK	INAK
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **RX** : CAN Rx 信号

**CAN\_RX** ピンの実際の値を監視します。

ビット 10 **SAMP** : 最終サンプルポイント

最後のサンプルポイントにおける RX の値 (受信した現在のビット値) です。

ビット 9 **RXM** : 受信モード

CAN ハードウェアは、現在はレシーバです。

ビット 8 **TXM** : 送信モード

CAN ハードウェアは、現在はトランスミッタです。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **SLAKI** : スリープ確認割り込み

SLKIE=1 の場合、このビットは、bxCAN がスリープモードに入ったことを通知するために、ハードウェアによってセットされます。CAN\_IER レジスタの SLKIE ビットがセットされている場合にこのビットがセットされると、ステータス変更割り込みが生成されます。

SLAK がクリアされると、このビットはソフトウェアまたはハードウェアによってクリアされます。

**注:** **SLKIE=0 の場合、SLAKI ビットのポーリングはできません。この場合、SLAK ビットはポーリングできません。**

ビット 3 **WKUI** : ウェイクアップ割り込み

このビットは、CAN ハードウェアがスリープモードのときに SOF ビットが検出されたことを通知するために、ハードウェアによってセットされます。CAN\_IER レジスタの WKUIE ビットがセットされている場合、このビットをセットするとステータス変更割り込みが生成されます。

このビットはソフトウェアによってクリアされます。

ビット 2 **ERRI** : エラー割り込み

エラー検出時に CAN\_ESR のビットがセットされており、CAN\_IER で対応する割り込みが有効になっている場合、このビットはハードウェアによってセットされます。CAN\_IER レジスタの ERRIE ビットがセットされている場合、このビットをセットするとステータス変更割り込みが生成されます。

このビットはソフトウェアによってクリアされます。

ビット 1 **SLAK** : スリープ確認応答

このビットは、ハードウェアによってセットされ、CAN ハードウェアがスリープモードにあることをソフトウェアに知らせます。このビットは、ソフトウェアからのスリープモードリクエスト (CAN\_MCR レジスタの SLEEP ビットをセット) に対する確認応答になります。

このビットは、CAN ハードウェアがスリープモードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセッショビットを観測する必要があります。

**注:** **CAN\_MCR レジスタの SLEEP ビットがクリアされると、スリープモードから出るプロセスがトリガされます。SLEEP ビットのクリアについては、CAN\_MCR レジスタの AWUM ビットの説明を参照してください。**

ビット 0 **INAK** : 初期化確認応答

このビットは、CAN ハードウェアが初期化モードにあることをソフトウェアに知らせるために、ハードウェアによってセットされます。このビットは、ソフトウェアからの初期化リクエスト (CAN\_MCR レジスタの INRQ ビットをセット) に対する確認応答になります。

このビットは、CAN ハードウェアが初期化モードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセッショビットを観測する必要があります。

## CAN 送信ステータスレジスタ (CAN\_TSR)

アドレスオフセット : 0x08

リセット値 : 0x1C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2
r	r	r	r	r	r	r	r	rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	TERR0	ALST0	TXOK0	RQCP0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

ビット 31 **LOW2** : メールボックス 2 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 2 が最低優先順位の時、このビットがハードウェアによってセットされます。

ビット 30 **LOW1** : メールボックス 1 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 1 が最低優先順位の時、このビットがハードウェアによってセットされます。

ビット 29 **LOW0** : メールボックス 0 最低優先順位フラグ

複数のメールボックスが送信待ち状態であり、メールボックス 0 が最低優先順位の時、このビットがハードウェアによってセットされます。

**注 :** ペンディング状態のメールボックスが1 つしかないとき、LOW[2:0] ビットは 0 にセットされません。

ビット 28 **TME2** : 送信メールボックス 2 エンプティ

このビットは、メールボックス 2 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 27 **TME1** : 送信メールボックス 1 エンプティ

このビットは、メールボックス 1 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 26 **TME0** : 送信メールボックス 0 エンプティ

このビットは、メールボックス 0 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。

ビット 25:24 **CODE[1:0]** : メールボックスコード

少なくとも 1 個の送信メールボックスが空き状態の場合、このコード値は、次の空き状態の送信メールボックスの番号になります。

すべての送信メールボックスが待ち状態の場合、このコード値は、最低優先順位の送信メールボックスの番号になります。

ビット 23 **ABRQ2** : メールボックス 2 アポートリクエスト

対応するメールボックスへの送信リクエストをアポートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 22:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **TERR2** : メールボックス 2 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。



ビット 18 **ALST2** : メールボックス 2 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。

ビット 17 **TXOK2** : メールボックス 2 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0 : 前回の送信は失敗しました。

1 : 前回の送信は成功しました。

このビットは、メールボックス 2 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 391 を参照してください。

ビット 16 **RQCP2** : メールボックス 2 リクエスト完了

最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN\_TMD2R レジスタの TXRQ2 ビットをセット)。

このビットをクリアするとメールボックス 2 のすべてのステータスビット (TXOK2、ALST2、および TERR2) がクリアされます。

ビット 15 **ABRQ1** : メールボックス 1 アボートリクエスト

対応するメールボックスへの送信リクエストをアボートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **TERR1** : メールボックス 1 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。

ビット 10 **ALST1** : メールボックス 1 アービトレーションロスト

このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。

ビット 9 **TXOK1** : メールボックス 1 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0 : 前回の送信は失敗しました。

1 : 前回の送信は成功しました。

このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 391 を参照してください。

ビット 8 **RQCP1** : メールボックス 1 リクエスト完了

最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN\_TI1R レジスタの TXRQ1 ビットをセット)。

このビットをクリアするとメールボックス 1 のすべてのステータスビット (TXOK1、ALST1、および TERR1) がクリアされます。

ビット 7 **ABRQ0** : メールボックス 0 アボートリクエスト

対応するメールボックスへの送信リクエストをアボートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **TERR0** : メールボックス 0 送信エラー

このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。



- ビット 2 ALST0** : メールボックス 0 アービトレーションロスト  
このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。
- ビット 1 TXOK0** : メールボックス 0 送信OK  
ハードウェアは、送信を試みた後で毎回このビットを更新します。  
0 : 前回の送信は失敗しました。  
1 : 前回の送信は成功しました。  
このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。 [図 391](#)を参照してください。
- ビット 0 RQCP0** : メールボックス 0 リクエスト完了  
最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN\_TIOER レジスタの TXRQ0 ビットをセット)。  
このビットをクリアするとメールボックス 0 のすべてのステータスビット (TXOK0、ALST0、および TERR0) がクリアされます。

## CAN 受信 FIFO 0 レジスタ (CAN\_RF0R)

アドレスオフセット : 0x0C  
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]	
										rs	rc_w1	rc_w1		r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

### ビット 5 RFOM0 : FIFO 0 出力メールボックスの解放

FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。  
このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。

### ビット 4 FOVR0 : FIFO 0 オーバーラン

このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。  
このビットはソフトウェアによってクリアされます。

### ビット 3 FULL0 : FIFO 0 フル

FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。  
このビットはソフトウェアによってクリアされます。

### ビット 2 予約済みであり、リセット値に保持する必要があります。

### ビット 1:0 FMP0[1:0] : FIFO 0 メッセージペンディング

これらのビットは、受信 FIFO 内でペンディング状態になっているメッセージの数を示します。FMP は、ハードウェアが FIFO に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM0 ビットをセットして出力メールボックスを解放するたびに減少します。

## CAN 受信 FIFO 1 レジスタ (CAN\_RF1R)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]	
										rs	rc_w1	rc_w1		r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **RFOM1** : FIFO 1 出力メールボックスの解放

FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。

このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。

ビット 4 **FOVR1** : FIFO 1 オーバーラン

このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。

このビットはソフトウェアによってクリアされます。

ビット 3 **FULL1** : FIFO 1 フル

FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。

このビットはソフトウェアによってクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **FMP1[1:0]** : FIFO 1 メッセージペンディング

これらのビットは、受信 FIFO1 内でペンディング状態になっているメッセージの数を示します。

FMP1 は、ハードウェアが FIFO1 に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM1 ビットをセットして出力メールボックスを解放するたびに減少します。

## CAN 割り込み有効レジスタ (CAN\_IER)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Res.	Res.	Res.	LEC IE	BOF IE	EPV IE	EWG IE	Res.	FOV IE1	FF IE1	FMP IE1	FOV IE0	FF IE0	FMP IE0	TME IE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw



ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **SLKIE** : スリープ割り込み有効

- 0 : SLAKI ビットがセットされると、割り込みは生成されません。
- 1 : SLAKI ビットがセットされると、割り込みが生成されます。

ビット 16 **WKUIE** : ウェイクアップ割り込み有効

- 0 : WKUI ビットがセットされると、割り込みは生成されません。
- 1 : WKUI ビットがセットされると、割り込みが生成されます。

ビット 15 **ERRIE** : エラー割り込みイネーブル

- 0 : CAN\_ESR レジスタでエラー条件がペンディング状態のとき、割り込みは生成されません。
- 1 : CAN\_ESR レジスタでエラー条件がペンディング状態のとき、割り込みが生成されます。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **LECIE** : 最終エラーコード割り込み有効

- 0 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされる時、ERRI ビットはセットされません。
- 1 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされると、ERRI ビットがセットされます。

ビット 10 **BOFIE** : バスオフ割り込み有効

- 0 : BOFF がセットされる時、ERRI ビットはセットされません。
- 1 : BOFF がセットされると、ERRI ビットがセットされます。

ビット 9 **EPVIE** : エラーパッシブ割り込み有効

- 0 : EPVF がセットされる時、ERRI ビットはセットされません。
- 1 : EPVF がセットされると、ERRI ビットがセットされます。

ビット 8 **EWGIE** : エラー警告割り込み有効

- 0 : EWGF がセットされる時、ERRI ビットはセットされません。
- 1 : EWGF がセットされると、ERRI ビットがセットされます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FOVIE1** : FIFO オーバーラン割り込み有効

- 0 : FOVR がセットされる時、割り込みは生成されません。
- 1 : FOVR がセットされると、割り込みが生成されます。

ビット 5 **FFIE1** : FIFO フル割り込み有効

- 0 : FULL ビットがセットされる時、割り込みは生成されません。
- 1 : FULL ビットがセットされると、割り込みが生成されます。

ビット 4 **FMPIE1** : FIFO メッセージペンディング割り込み有効

- 0 : FMP[1:0] ビットの状態が 00b でないとき、割り込みは生成されません。
- 1 : FMP[1:0] ビットの状態が 00b でないとき、割り込みが生成されます。

ビット 3 **FOVIE0** : FIFO オーバーラン割り込み有効

- 0 : FOVR ビットがセットされる時、割り込みは生成されません。
- 1 : FOVR ビットがセットされると、割り込みが生成されます。

ビット 2 **FFIE0** : FIFO フル割り込み有効

- 0 : FULL ビットがセットされる時、割り込みは生成されません。
- 1 : FULL ビットがセットされると、割り込みが生成されます。

ビット 1 **FMPIE0** : FIFO メッセージペンディング割り込み有効

- 0 : FMP[1:0] ビットの状態が 00b でない時、割り込みは生成されません。
- 1 : FMP[1:0] ビットの状態が 00b でない時、割り込みが生成されます。

ビット 0 **TMEIE** : 送信メールボックスエンプティ割り込み有効

- 0 : RQCPx ビットがセットされる時、割り込みは生成されません。
- 1 : RQCPx ビットがセットされると、割り込みが生成されます。

注: [セクション 31.8 : bxCAN の割り込みを参照してください。](#)

## CAN エラーステータスレジスタ (CAN\_ESR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

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

ビット 31:24 **REC[7:0]** : 受信エラーカウンタ

CAN プロトコルの障害隔離メカニズムの実装部です。受信時にエラーがあった場合、このカウンタは、CAN 規格に定められたエラー条件に応じて、1 または 8 だけインクリメントされます。受信が正常に行われるたびに、このカウンタは 1 だけデクリメントされます (カウンタの値が 128 を超えた場合は、120 にリセットされます)。カウンタ値が 127 を超えると、CAN コントローラはエラーパッシブ状態に入ります。

ビット 23:16 **TEC[7:0]** : 9 ビット送信エラーカウンタの最下位バイト

CAN プロトコルの障害隔離メカニズムの実装部です。

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **LEC[2:0]** : 最終エラーコード

このフィールドはハードウェアによってセットされ、最後に CAN バスで検出されたエラーのエラー条件を示すコードを保持しています。メッセージがエラーなしで転送 (受信または送信) されている場合には、このフィールドは "0" にクリアされます。

LEC[2:0] ビットは、ソフトウェアによって値 0b111 にセットできます。ビットはハードウェアによって更新され、現在の通信ステータスを示します。

- 000 : エラーなし
- 001 : スタッフエラー
- 010 : フォームエラー
- 011 : ACK エラー
- 100 : ビットリセッシブエラー
- 101 : ビットドミナントエラー
- 110 : CRC エラー
- 111 : ソフトウェアによってセットされます

ビット 3 予約済みであり、リセット値に保持する必要があります。



**ビット 2 BOFF** : バスオフフラグ

このビットは、バスオフ状態に入るとハードウェアによってセットされます。バスオフ状態に入るのは、TEC オーバーフローのとき (255 を超えたとき) です。 [セクション 31.7.6 \(1024 ページ\)](#) を参照してください。

**ビット 1 EPVF** : エラーパッシブフラグ

このビットは、エラーパッシブ限界に達したときに (受信エラーカウンタまたは送信エラーカウンタ > 127)、ハードウェアによってセットされます。

**ビット 0 EWGF** : エラー警告フラグ

このビットは、警告限界に達したときに、ハードウェアによってセットされます (受信エラーカウンタまたは送信エラーカウンタ ≥ 96)。

## CAN ビットタイミングレジスタ (CAN\_BTR)

アドレスオフセット : 0x1C

リセット値 : 0x0123 0000

このレジスタは、CAN ハードウェアが初期化モードのときに、ソフトウェアによってのみアクセスできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
SILM	LBKM	Res.	Res.	Res.	Res.	SJW[1:0]		Res.	TS2[2:0]			TS1[3:0]				
r/w	r/w					r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	Res.	Res.	Res.	Res.	Res.	BRP[9:0]										
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

**ビット 31 SILM** : サイレントモード (デバッグ用)

0 : 通常動作

1 : サイレントモード

**ビット 30 LBKM** : ループバックモード (デバッグ用)

0 : ループバックモードは無効です。

1 : ループバックモードは有効です。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。

**ビット 25:24 SJW[1:0]** : 再同期ジャンプ幅

これらのビットは、再同期を行うために CAN ハードウェアがビットを伸縮できる最大タイムクオンタム数を定義します。

$$t_{RJW} = t_q \times (SJW[1:0] + 1)$$

ビット 23 予約済みであり、リセット値に保持する必要があります。

**ビット 22:20 TS2[2:0]** : 時間セグメント 2

これらのビットは、時間セグメント 2 に含まれるタイムクオンタム数を定義します。

$$t_{BS2} = t_q \times (TS2[2:0] + 1)$$

ビット 19:16 **TS1[3:0]** : 時間セグメント 1

これらのビットは、時間セグメント 1 に含まれるタイムクオンタム数を定義します。

$$t_{BS1} = t_q \times (TS1[3:0] + 1)$$

ビットタイミングの詳細については、[セクション 31.7.7 : ビットタイミング \(1024 ページ\)](#) を参照してください。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **BRP[9:0]** : ボーレートプリスケアラ

これらのビットは、タイムクオンタムの長さを定義します。

$$t_q = (BRP[9:0] + 1) \times t_{PCLK}$$

### 31.9.3 CAN メールボックスレジスタ

ここでは、送受信メールボックスのレジスタについて説明します。詳細なレジスタ配置については、[セクション 31.7.5 : メッセージストレージ \(1022 ページ\)](#) を参照してください。

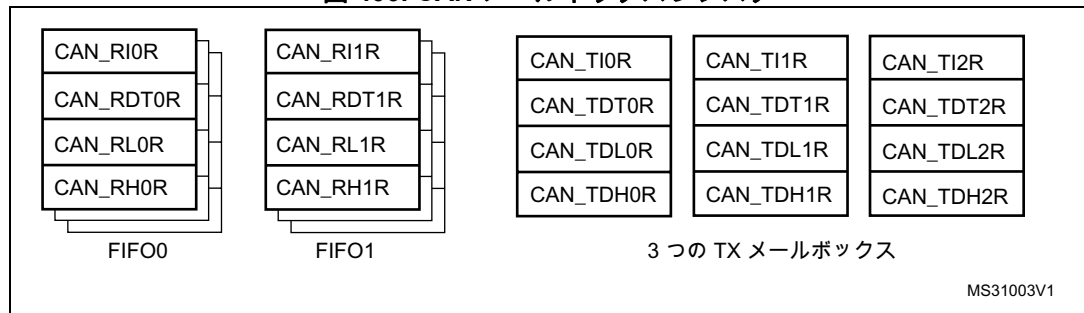
送信メールボックスと受信メールボックスのレジスタは同じですが、次の点が異なります。

- CAN\_RDTxR レジスタの FMI フィールド。
- 受信メールボックスは常に書き込み保護されます。
- 送信メールボックスは空のとき (CAN\_TSR レジスタの対応する TME ビットがセットされているとき) のみ、書き込み可能です。

送信メールボックスは 3 つ、受信メールボックスは 2 つあります。各受信メールボックスでは、3 種類の深さのレベルの FIFO にアクセスでき、FIFO 内の最も古い受信メッセージにのみアクセスできます。

各メールボックスは 4 つのレジスタから構成されています。

図 400. CAN メールボックスレジスタ



## CAN 送信メールボックス識別子レジスタ (CAN\_TlXR) (x=0..2)

アドレスオフセット : 0x180、0x190、0x1A0

リセット値 : 0xFFFF XXXX (ただし、ビット 0、TXRQ = 0)

メールボックスが送信待ち (TMEx リセット) のとき、すべての送信レジスタは書き込み保護されます。

このレジスタは、送信リクエスト制御 (ビット 0、リセット値 0) も実装します。

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

ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子

IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。

ビット 20:3 **EXID[17:0]** : 拡張識別子

拡張識別子の最下位ビットです。

ビット 2 **IDE** : 識別子拡張

このビットは、メールボックス内のメッセージの識別子のタイプを定義します。

0 : 標準識別子。

1 : 拡張識別子。

ビット 1 **RTR** : リモート送信リクエスト

0 : データフレーム

1 : リモートフレーム

ビット 0 **TXRQ** : 送信メールボックスリクエスト

対応するメールボックスに送信を要求するために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。



## CAN メールボックスデータ長制御/タイムスタンプレジスタ (CAN\_TDTxR) (x = 0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x184、0x194、0x1A4

リセット値 : 0xXXXXX XXXX

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

### ビット 31:16 TIME[15:0] : メッセージタイムスタンプ

このフィールドは、SOF 送信時にキャプチャされた 16 ビットタイマの値を含みます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

### ビット 8 TGT : 送信グローバルタイム

このビットは、ハードウェアがタイムトリガ通信モードにある (CAN\_MCR レジスタの TTCM ビットがセットされている) ときにのみアクティブとなります。

0 : タイムスタンプ TIME[15:0] は送信されません。

1 : タイムスタンプ TIME[15:0] の値は、8 バイトメッセージの最後の 2 データバイトで送信されます。

このとき、データバイト 7 の TIME[7:0] とデータバイト 6 の TIME[15:8] は、CAN\_TDHxR[31:16] レジスタ (DATA6[7:0]、DATA7[7:0]) に書き込まれたデータを置換します。これらの 2 バイトを CAN バス経由で送信するには、DLC を 8 としてプログラミングする必要があります。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

### ビット 3:0 DLC[3:0] : データ長コード

このフィールドは、データフレーム内のデータバイト数、またはリモートフレームリクエストのデータバイト数を定義します。

メッセージは、DLC フィールドの値に応じて、0~8 個のデータバイトを含むことができます。

## CAN メールボックスデータ下位レジスタ (CAN\_TDLxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x188、0x198、0x1A8

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 **DATA3[7:0]** : データバイト 3  
メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]** : データバイト 2  
メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]** : データバイト 1  
メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]** : データバイト 0  
メッセージのデータバイト 0 です。  
メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

## CAN メールボックスデータ上位レジスタ (CAN\_TDHxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x18C、0x19C、0x1AC

リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 31:24 **DATA7[7:0]** : データバイト 7  
 メッセージのデータバイト 7 です。  
**注:** このメッセージの TGT と TTCM がアクティブな場合、DATA7 と DATA6 はタイムスタンプの値で置換されます。
- ビット 23:16 **DATA6[7:0]** : データバイト 6  
 メッセージのデータバイト 6 です。
- ビット 15:8 **DATA5[7:0]** : データバイト 5  
 メッセージのデータバイト 5 です。
- ビット 7:0 **DATA4[7:0]** : データバイト 4  
 メッセージのデータバイト 4 です。

### CAN 受信 FIFO メールボックス識別子レジスタ (CAN\_RIxR) (x=0..1)

アドレスオフセット : 0x1B0、0x1C0  
 リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

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

- ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子  
 IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。
- ビット 20:3 **EXID[17:0]** : 拡張識別子  
 拡張識別子の最下位ビットです。
- ビット 2 **IDE** : 識別子拡張  
 このビットは、メールボックス内のメッセージの識別子のタイプを定義します。  
 0 : 標準識別子。  
 1 : 拡張識別子。
- ビット 1 **RTR** : リモート送信リクエスト  
 0 : データフレーム  
 1 : リモートフレーム
- ビット 0 予約済みであり、リセット値に保持する必要があります。



## CAN 受信 FIFO メールボックスデータ長制御/タイムスタンプレジスタ (CAN\_RDTxR) (x = 0..1)

アドレスオフセット : 0x1B4、0x1C4  
リセット値 : 0xXXXX XXXX

すべての受信レジスタは書き込み保護されます。

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

ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ

このフィールドは、SOF 検出時にキャプチャされた 16 ビットタイマの値を含みます。

ビット 15:8 **FMI[7:0]** : 該当フィルタインデックス

このレジスタは、メールボックスに格納されたメッセージが通過したフィルタの索引を含みます。識別子フィルタリングの詳細については、[セクション 31.7.4 : 識別子フィルタリング \(1018 ページ\)](#) の該当フィルタインデックス項を参照してください。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **DLC[3:0]** : データ長コード

このフィールドは、データフレームに格納されるデータバイト数 (0~8) を定義します。リモートフレームリクエストの場合、このフィールドは 0 です。

## CAN 受信 FIFO メールボックスデータ下位レジスタ (CAN\_RDLxR) (x=0..1)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x1B8、0x1C8  
リセット値 : 0xXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA3[7:0]** : データバイト 3  
 メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]** : データバイト 2  
 メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]** : データバイト 1  
 メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]** : データバイト 0  
 メッセージのデータバイト 0 です。  
 メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

### CAN 受信 FIFO メールボックスデータ上位レジスタ (CAN\_RDHxR) (x=0..1)

アドレスオフセット : 0x1BC、0x1CC

リセット値 : 0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA7[7:0]** : データバイト 7  
 メッセージのデータバイト 3 です。

ビット 23:16 **DATA6[7:0]** : データバイト 6  
 メッセージのデータバイト 2 です。

ビット 15:8 **DATA5[7:0]** : データバイト 5  
 メッセージのデータバイト 1 です。

ビット 7:0 **DATA4[7:0]** : データバイト 4  
 メッセージのデータバイト 0 です。



## 31.9.4 CAN フィルタレジスタ

### CAN フィルタマスタレジスタ (CAN\_FMR)

アドレスオフセット : 0x200

リセット値 : 0x2A1C 0E01

このレジスタのすべてのビットは、ソフトウェアによってセット/クリアされます。

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

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **FINIT** : フィルタ初期化モード

フィルタバンクの初期化モード

0 : アクティブフィルタモード。

1 : フィルタの初期化モード。

### CAN フィルタモードレジスタ (CAN\_FM1R)

アドレスオフセット : 0x204

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN\_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注 : [図 393 : フィルタバンクのスケール設定 - レジスタ構造 \(1020 ページ\)](#) を参照してください。

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **FBMx** : フィルタモード

フィルタ x のレジスタのモードです。

0 : フィルタバンク x の 2 つの 32 ビットレジスタは識別子マスクモードになっています。

1 : フィルタバンク x の 2 つの 32 ビットレジスタは識別子リストモードになっています。

注 : **ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。**

## CAN フィルタスケールレジスタ (CAN\_FS1R)

アドレスオフセット : 0x20C

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN\_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **FSCx** : フィルタスケール設定

これらのビットは、フィルタ 13 - 0 のスケール設定を定義します。

0 : デュアル 16 ビットスケール設定

1 : シングル 32 ビットスケール設定

**注 :** ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

**注 :** [図 393 : フィルタバンクのスケール設定 - レジスタ構造 \(1020 ページ\)](#) を参照してください。

## CAN フィルタ FIFO 割当てレジスタ (CAN\_FFA1R)

アドレスオフセット : 0x214

リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN\_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **FFAx** : フィルタ x のフィルタ FIFO 割り当て

このフィルタを通過するメッセージは、特定の FIFO に保存されます。

0 : フィルタが FIFO 0 に割り当てられます。

1 : フィルタが FIFO 1 に割り当てられます。

**注 :** ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

## CAN フィルタアクティベーションレジスタ (CAN\_FA1R)

アドレスオフセット : 0x21C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FACT1 3	FACT1 2	FACT1 1	FACT1 0	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0
		r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **FACTx** : アクティブなフィルタ

ソフトウェアがこのビットをセットしてフィルタ x をアクティブにします。フィルタ x レジスタ (CAN\_FxR[0:7]) を変更するには、FACTx ビットをクリアするか、CAN\_FMR レジスタの FINIT ビットをセットする必要があります。

0 : フィルタ x はアクティブではありません。

1 : フィルタ x はアクティブです。

**注 :** ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。



## CAN フィルタバンク i レジスタ x (CAN\_FiRx) (i = 0..13, x = 1, 2)

アドレスオフセット : 0x240 から 0x2AC

リセット値 : 0xFFFF XXXX

14 個のフィルタバンク (i= 0~13) が存在します。各フィルタバンク i は、2 個の 32 ビットレジスタ (CAN\_FiR[2:1]) から構成されています。

このレジスタを変更できるのは、CAN\_FAxR レジスタの FACTx ビットがクリアされている場合か、CAN\_FMR レジスタの FINIT ビットがセットされている場合に限られます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

すべての設定において :

ビット 31:0 **FB[31:0]** : フィルタビット

### 識別子

レジスタの各ビットは、予期される識別子の対応するビットのレベルを指定します。

0 : ドミナントビットが予期されます。

1 : リセシブビットが予期されます。

### マスク

レジスタの各ビットは、関連する識別子レジスタのビットが、予期される識別子の対応するビットに一致する必要があるかどうかを指定します。

0 : 無視 (このビットは比較に使用されません)

1 : 一致が必要 (受信識別子のビットのレベルは、フィルタの対応する識別子レジスタで指定されたレベルと同じでなければなりません)

**注 :** [フィルタのスケールとモードの設定に応じて、各レジスタの機能は異なることがあります。フィルタ配置、機能の説明、およびマスクレジスタの関連については、\[セクション 31.7.4 : 識別子フィルタリング \\(1018 ページ\\)\]\(#\) を参照してください。](#)

マスクモードにおけるマスク/識別子レジスタのビット配置は、[識別子リストモードの場合と同じ](#)です。

[フィルタバンクのレジスタ配置/アドレスについては、\[表 173 \\(1050 ページ\\)\]\(#\) を参照してください。](#)



表 173. bxCAN レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x188	CAN_TDL0R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x18C	CAN_TDH0R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x190	CAN_TI1R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE	RTR	TXRQ			
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
0x194	CAN_TDT1R	TIME[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																	
0x198	CAN_TDL1R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x19C	CAN_TDH1R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1A0	CAN_TI2R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE	RTR	TXRQ			
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
0x1A4	CAN_TDT2R	TIME[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]				
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x																	
0x1A8	CAN_TDL2R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1AC	CAN_TDH2R	DATA7[7:0]							DATA6[7:0]							DATA5[7:0]							DATA4[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1B0	CAN_RI0R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE	RTR	Res			
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x1B4	CAN_RDT0R	TIME[15:0]															FMI[7:0]							Res.	Res.	Res.	Res.	DLC[3:0]					
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x									
0x1B8	CAN_RDL0R	DATA3[7:0]							DATA2[7:0]							DATA1[7:0]							DATA0[7:0]										
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x



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



表 173. bxCAN レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x240	CAN_F0R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x244	CAN_F0R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x248	CAN_F1R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x24C	CAN_F1R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
⋮	⋮	⋮																															
0x318	CAN_F27R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x31C	CAN_F27R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X



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

## 32 ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)

USB は STM32F303xB/C/D/E デバイスでのみ使用可能です。STM32F3xx デバイスに USB はありません。

### 32.1 概要

USB ペリフェラルは、フルスピードの USB 2.0 バスと APB1 バスの間のインタフェースを実装します。USB サスペンド/レジュームがサポートされるため、節電のためにデバイスのクロックを停止できます。

### 32.2 USB の主な機能

- USB 仕様バージョン 2.0 フルスピード準拠
- 1 から 8 まで設定可能なエンドポイント数
- 巡回冗長検査 (CRC) の生成/チェック、NRZI (Non-return-to-zero Inverted) 方式のエンコード/デコード、およびビットスタッフィング
- アイソクロナス転送サポート
- ダブルバッファバルク/アイソクロナスエンドポイントサポート
- USB サスペンド/レジューム操作
- フレームロッククロックパルス生成

製品の実装によっては、次の追加機能も使用できます ( [セクション 32.3 : USB の実装](#) を参照 ) :

- USB 2.0 リンク電源管理サポート

### 32.3 USB の実装

[表 174](#)に、デバイスでの USB 実装を示します。

表 174. STM32F3xx USB の実装

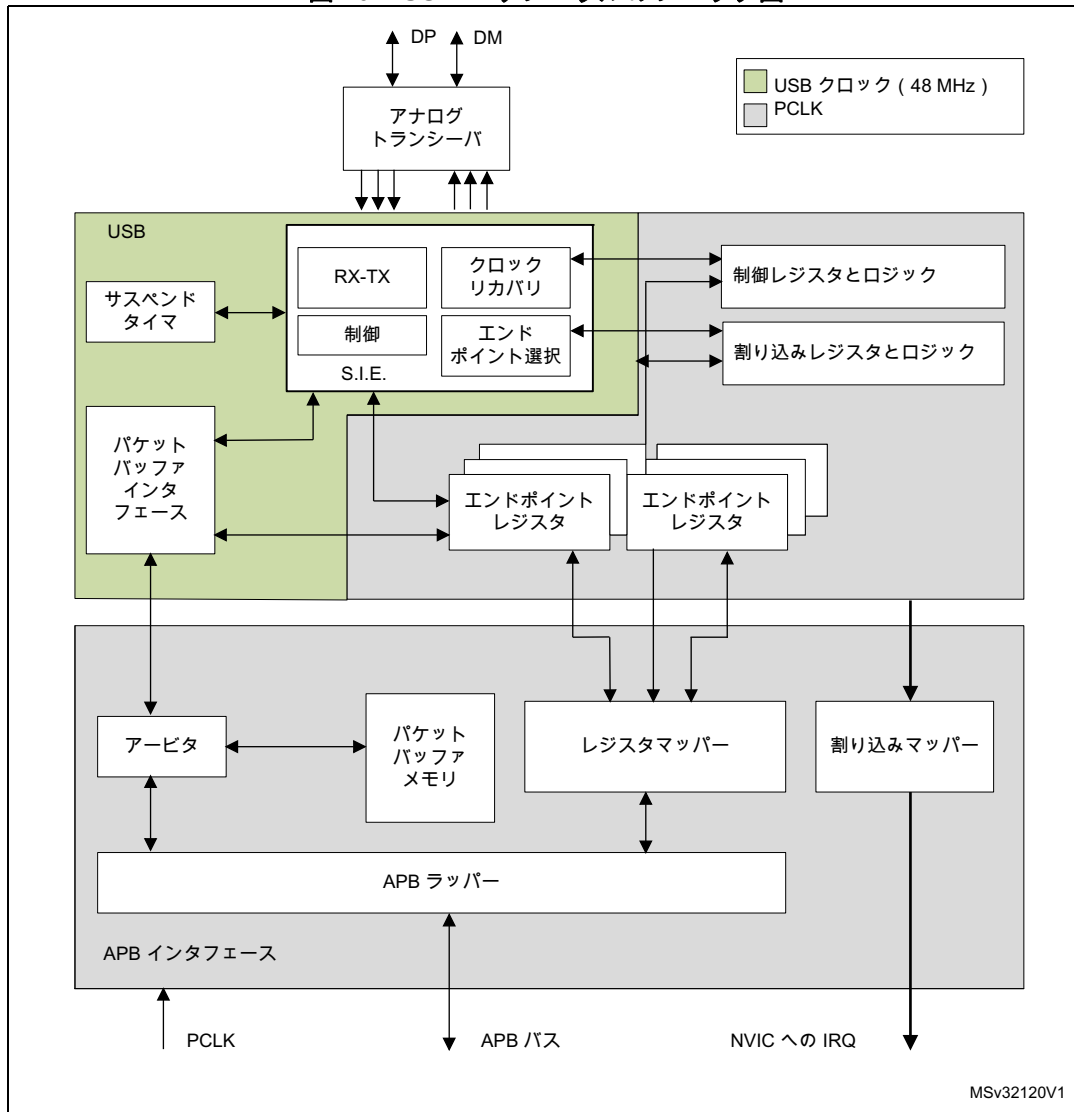
USB の機能 <sup>(1)</sup>	STM32F303xB/C	STM32F303xD/E <sup>(2)</sup>
エンドポイント数	8	8
専用パケットバッファメモリ SRAM のサイズ	512 バイト	1024 バイト <sup>(3)</sup>
専用パケットバッファメモリ SRAM のアクセス方式	1 x 16 ビット/ワード	2 x 16 ビット/ワード
USB 2.0 リンク電源管理 (LPM) のサポート	-	X

1. X : サポートされています
2. STM32F303xD/E には、USB 仕様バージョン 2.0 に準拠したフルスピード USB デバイスペリフェラルが組み込まれています。USB インタフェースには、フルスピード (12 Mbit/s) 機能インタフェースが実装されており、USB 2.0 リンク電源管理のサポートも付いています。ソフトウェアで設定可能な、最大1KBのパケットメモリ (有効化している場合 CAN ペリフェラルに 256 バイト使用) を持つエンドポイント設定があり、サスペンド/レジュームもサポートしています。
3. RCC\_APB1ENR レジスタで CAN ペリフェラルクロックが有効なときには、最初の768 のみが USB で使用可能であり、残りの 256 バイトは CAN によって使用されます。

## 32.4 USART の機能詳細

図 401 に、USB ペリフェラルのブロック図を示します。

図 401. USB ペリフェラルのブロック図



USB ペリフェラルは、ホスト PC とマイクロコントローラによって実装される機能の間に USB 準拠接続を提供します。ホスト PC とシステムメモリ間のデータ転送は、USB ペリフェラルによって直接アクセスされる専用パケットバッファメモリを通じて行われます。この専用メモリのサイズは最大 512 バイトであり、最大 16 の単方向または 8 つの双方向エンドポイントを使用できます。USB ペリフェラルは USB ホストとインタフェースして、トークンパケットを検出し、データの送受信を処理し、USB 標準の要求に従ってハンドシェイクパケットを処理します。トランザクションのフォーマット化は、CRC の生成とチェックも含めて、ハードウェアによって行われます。

各エンドポイントは、エンドポイント関連のメモリ領域の位置、その大きさ、または送信バイト数を示すバッファ記述ブロックに関連付けられます。有効な機能/エンドポイントのペアのトークンが USB ペリフェラルによって検出されると、関連するデータ転送が行われます（必要であり、エンドポイントが設定されている場合）。USB ペリフェラルによってバッファされたデータは、内部 16 ビットレジスタに配置され、専用バッファへのメモリアクセスが行われます。すべてのデータが転送され

ると、必要な場合には、転送方向に従って、適切なハンドシェイクパケットが USB 経由で生成され、または転送の方向によって期待されます。

トランザクションの終了時には、エンドポイント固有の割り込みが生成され、ステータスレジスタを読み出すか、別の割り込み応答ルーチンを使用します。マイクロコントローラは、以下を決めることができます。

- 処理するエンドポイント
- エラーが発生した場合に行うトランザクションのタイプ (ビットスタッフィング、フォーマット、CRC、プロトコル、ACK の欠如、オーバーラン/アンダーランなど)

アイソクロナス転送と高スループットバルク転送に対する特別なサポートが提供され、ダブルバッファ私用が実装されることにより、USB ペリフェラルが使用できるバッファが常にあり、マイクロコントローラ用のバッファが別にあります。

必要なときには、制御レジスタに書き込むことによって、ユニットを低電力モード (SUSPEND モード) にできます。このとき、すべての静的消費電流が回避され、USB クロックをスローダウンまたは停止できます。低電力モード中に USB 入力でアクティビティが検出されると、デバイスは非同期でウェイクアップします。特殊な割り込みソースをウェイクアップラインに直接接続することにより、システムは通常のクロック生成をただちに再開し、直接クロックの開始/停止をサポートできます。

## 32.4.1 USB ブロックの説明

USB ペリフェラルは USB インタフェースに関するすべての機能を実装し、これには以下のブロックが含まれます。

- シリアルインタフェースエンジン (SIE) : このブロックの機能には、同期パターン認識、ビットスタッフィング、CRC 生成およびチェック、PID 確認/生成、およびハンドシェイク評価が含まれます。USB トランシーバとインタフェースし、ローカルデータストレージ用のパケットバッファインタフェースによって提供される仮想バッファを使用する必要があります。このユニットは、フレーム開始 (SOF)、USB\_Reset、データエラーなどの USB ペリフェラルイベントと、送信終了、パケットの正しい受信などのエンドポイント関連イベントに従って信号を生成し、これらの信号は割り込みの生成に使用されます。
- タイマ : このブロックは、フレーム開始ロッククロックパルスを生成し、トラフィックが 3 ms の間、受信されなかったときにはグローバルサスペンドを (ホストから) 検出します。
- パケットバッファインタフェース : このブロックは、ローカルメモリを管理して、送信と受信の両方のためのバッファのセットを柔軟に実装します。SIE から着信したリクエストに従って適切なバッファを選び、エンドポイントレジスタによって示されるメモリアドレスに配置します。パケットの終わりまで、各バイトの交換後にアドレスをインクリメントして、交換バイト数を追跡し、バッファが最大容量までオーバーランするのを防ぎます。
- エンドポイント関連レジスタ : 各エンドポイントには、エンドポイントのタイプと現在のステータスを含んだレジスタが関連付けられます。単方向/シングルバッファエンドポイントの場合、1つのレジスタで2つのエンドポイントを実装できます。レジスタの数は8つであり、最大16の単方向/シングルバッファまたは最大7つのダブルバッファエンドポイントを任意の組み合わせで使用できます。たとえば、USB ペリフェラルをプログラムして、4つのダブルバッファエンドポイントと8つのシングルバッファ/単方向エンドポイントを使用できます。
- 制御レジスタ : これらは USB ペリフェラル全体のステータスに関する情報を含むレジスタであり、レジュームやパワーダウンなどの USB イベントを強制するために使用されます。
- 割り込みレジスタ : これらは割り込みマスクとイベントのレコードを含みます。割り込み理由、割り込みステータスの問い合わせや、保留中の割り込みのステータスをクリアするために使用できます。

**注 :** \* エンドポイント 0 は、常にシングルバッファモードでの転送の制御に使用されます。



USB ペリフェラルは APB1 バスに APB1 インタフェースを通じて接続され、以下のブロックを含みます。

- パケットメモリ：これは、パケットバッファを物理的に含むローカルメモリです。パケットバッファインタフェースによって使用して、データ構造を作成し、アプリケーションソフトウェアから直接アクセスできます。パケットメモリのサイズは、最大 512 バイトであり、256 個の 16 ビットハーフワードで構成されます。
- アービタ：このブロックは、APB1 バスおよび USB インタフェースから着信するリクエストを受け入れます。APB1 アクセスに優先順位を付けることによってコンフリクトを解決するとともに、常にメモリ帯域幅の半分をすべての USB 転送を完了するために確保しています。このタイムデュプレックス方式によって仮想デュアルポート SRAM が実装され、USB トランザクション中のメモリアccessを可能にします。この方式により、任意の長さのマルチワード APB1 転送も可能です。
- レジスタマッパー：このブロックは、USB ペリフェラルのさまざまなバイト長およびビット長のレジスタを APB1 によってアドレス指定され構造化された 16 ビット長のハーフワードセットに集めます。
- APB1 ラッパー：これは、メモリとレジスタに APB1 へのインタフェースを提供します。また、USB ペリフェラル全体を APB1 アドレス空間にマップします。
- 割り込みマッパー：このブロックは、USB イベントによって割り込みを生成する方法を選択し、それらを 3 つの異なる NVIC のラインにマップするために使用されます。
  - USB 低優先割り込み (チャンネル 20)：すべての USB イベント (正しい転送、USB リセットなど) でトリガされます。ファームウェアによって、割り込みソースをチェックしてから割り込みを処理する必要があります。
  - USB 高優先割り込み (チャンネル 19)：可能な限り高い転送速度を達成するアイソクロナス転送およびダブルバッファバルク転送の正しい転送イベントによってのみトリガされます。
  - USB ウェイクアップ割り込み (チャンネル 42)：USB サスペンドモードからのウェイクアップイベントによってトリガされます。

## 32.5 プログラミングに関する考慮事項

以下のセクションでは、アプリケーションソフトウェア開発を容易にするために、USB ペリフェラルとアプリケーションプログラムの間の予期される相互作用について説明します。

### 32.5.1 汎用 USB デバイスのプログラミング

このパートでは、USB 準拠動作を得るためにアプリケーションソフトウェアに必要な主なタスクについて説明します。ほとんどの汎用 USB イベントに関連するアクションが考慮に入れられており、ダブルバッファエンドポイントとアイソクロナス転送という特殊ケースについて、それぞれのパラグラフで説明します。システムリセットを別にすると、アクションは常に USB ペリフェラルによって開始され、以下に述べる USB イベントの 1 つによって駆動されます。

### 32.5.2 システムリセットとパワーオンリセット

システムリセットおよびパワーオンリセットの際、アプリケーションソフトウェアが最初に行うべき操作は、必要なクロック信号のすべてを USB ペリフェラルに与えることであり、その後、リセット信号をネゲートして、レジスタにアクセスできるようにしなければなりません。初期化シーケンス全体は、以下のとおりです。

最初のステップとして、アプリケーションソフトウェアは、デバイスクロック管理ロジックによって提供される関連の制御ビットを使用して、レジスタマクロセルクロックを有効にし、マクロセル固有のリセット信号をネゲートする必要があります。

その後、CNTR レジスタの PDWN ビットを使用して、USB トランシーバに関連するデバイスのアナログ部分をオンにする必要があります。これには特別な処理が必要です。このビットは、ポートトランシーバに供給される内部電圧基準をオンにすることを目的としています。この回路にはスタートアップ時間が定義されており（データシートで指定されている  $t_{STARTUP}$ ）、この間の USB トランシーバの動作は定義されていません。したがって、CNTR レジスタの PDWN ビットをセットした後、この時間だけ待つ必要があります。その後、USB 部分のリセット条件を解除します（CNTR レジスタの FRES ビットをクリアすることによって）。ISTR レジスタをクリアすると、他のマクロセル動作が有効になる前に、誤って保留中の割り込みが削除されます。

システムリセット時には、マイクロコントローラは必要なすべてのレジスタとパケットバッファ記述テーブルを初期化して、USB ペリフェラルが割り込みとデータ転送を正しく生成できるようにする必要があります。エンドポイント固有ではないレジスタはすべて、アプリケーションソフトウェアの必要に応じて初期化する必要があります（有効にする割り込みの選択、パケットバッファのアドレスの選択など）。USB リセットの場合、その後、プロセスが続行します（以下のパラグラフを参照）。

#### USB リセット (RESET 割り込み)

このイベントが発生すると、USB ペリフェラルは、前のパラグラフで説明した初期化後のシステムリセットと同じ条件になります。すなわち、すべてのエンドポイントレジスタで通信が無効になります（USB ペリフェラルはパケットに応答しません）。USB リセットイベントに対して、USB 機能を有効にして、USB アドレスを 0 にし、デフォルトの制御エンドポイントのみを実装します（エンドポイントアドレスも 0）。これは、USB\_DADDR レジスタの機能有効 (EF) ビットをセットし、EP0R レジスタと関連するパケットバッファを初期化することによって行われます。USB エニュメレーションプロセス時、ホストは一意的なアドレスをこのデバイスに割り当て、これが USB\_DADDR レジスタの ADD[6:0] ビットに書き込まれなければならない、他に必要なエンドポイントを設定する必要があります。

RESET 割り込みが受信されると、アプリケーションソフトウェアは、割り込みをトリガしたリセットシーケンスの終了から 10 ms 以内に、USB 機能 0 のデフォルトのエンドポイントを再び有効にする必要があります。

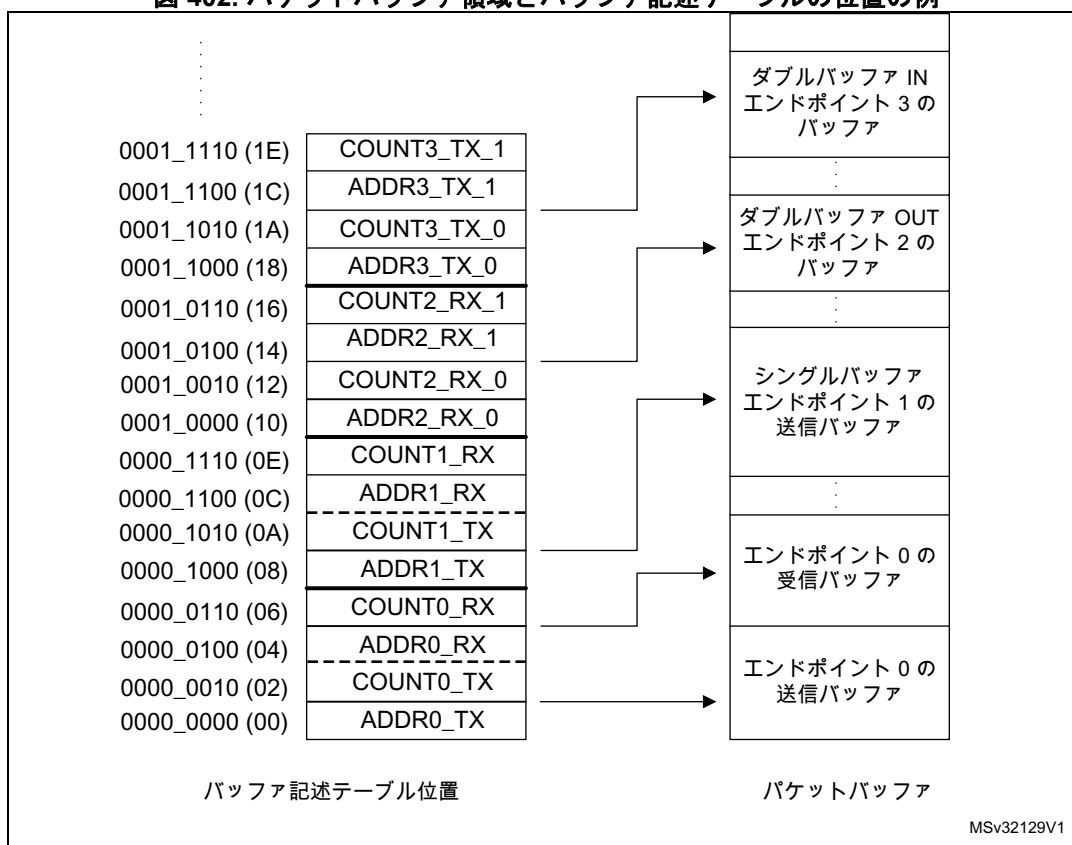
## パケットバッファの構造と用途

各双方向エンドポイントは、ホストとの間でデータを送受信できます。受信データは、そのエンドポイント用に予約された専用のメモリバッファに格納され、別のメモリバッファにはエンドポイントによって送信されるデータが格納されます。このメモリへのアクセスは、パケットバッファインタフェースブロックによって行われ、これはメモリアクセスリクエストを配信し、確認応答を待ちます。マイクロコントローラもパケットバッファメモリにアクセスする必要があるため、アクセスコンフリクトを避けるためにアービトレーションロジックが使用され、APB1 サイクルの半分がマイクロコントローラのアクセスに使用され、残りの半分が USB ペリフェラルのアクセスに使用されます。このように、パケットメモリがデュアルポート SRAM であるかのように両方のエージェントが動作でき、マイクロコントローラが連続アクセスを実行中でも、コンフリクトを気にする必要がありません。USB ペリフェラルロジックは、専用クロックを使用します。この専用クロックの周波数は、USB 標準の要件により 48 MHz に固定されており、これは APB1 バスへのインタフェースに使用されるクロックと異なる場合があります。APB1 クロック周波数が USB ペリフェラルのクロック周波数より高い、または低い場合、さまざまなクロック設定が可能です。

**注：** *USB データレートとパケットメモリインタフェース要件により、APB1 クロックは、データのオーバーラン/アンダーラン問題を回避するために、少なくとも 10 MHz 以上の周波数でなければなりません。*

各エンドポイントには 2 つのパケットバッファが関連付けられます (通常は、送信用に 1 つ、受信用に 1 つ)。バッファの位置とサイズはバッファ記述テーブルで指定されるので、バッファはパケットメモリ内の任意の場所に配置でき、このテーブルも USB\_BTABLE レジスタによって示されるパケットメモリ内のアドレスに配置されます。各テーブルエントリがエンドポイントレジスタに関連付けられ、4 つの 16 ビットハードワードで構成されるため、テーブル開始アドレスは常に 8 バイトの境界に揃えられます (USB\_BTABLE レジスタの下位 3 ビットは常に 000)。バッファディスクリプタテーブルのエントリについては、[セクション 32.6.2: バッファディスクリプタテーブル](#) で説明します。エンドポイントが単方向であり、アイソクロナスでもダブルバッファバルクでもない場合、必要なパケットバッファは 1 つだけです (サポートされる転送方向に応じた 1 つのバッファ)。サポートされない転送方向または未使用のエンドポイントに関連するテーブル位置は、ユーザが使用できます。アイソクロナスおよびダブルバッファバルクエンドポイントには、パケットバッファの特別な取り扱いがあります (それぞれ、[セクション 32.5.4: アイソクロナス転送](#) および [セクション 32.5.3: ダブルバッファエンドポイント](#) を参照)。バッファ記述テーブルのエントリとパケットバッファ領域の関係を [図 402](#) に示します。

図 402. パケットバッファ領域とバッファ記述テーブルの位置の例



各パケットバッファは、始めから終わりまで、受信または送信のいずれかに使用されます。USB ペリフェラルは、割り当てられたメモリバッファに隣接するメモリ位置の内容を変更ことはありません。割り当てられたバッファ長を超えるパケットが受信された場合 (バッファオーバーラン条件)、使用可能な最期の位置までしか、データはメモリにコピーされません。

## エンドポイントの初期化

エンドポイントを初期化するための最初のステップは、ADDRn\_TX/ADDRn\_RX レジスタに適切な値を書き込んで、送信データがすでに使用可能になったことと、受信データをバッファできることを USB ペリフェラルに知らせることです。USB\_EPnR レジスタの EP\_TYPE ビットをエンドポイントのタイプに応じてセットし、最終的には EP\_KIND ビットを使用して、特に必要な機能を有効にする必要があります。送信側では、USB\_EPnR レジスタの STAT\_TX ビットを使用してエンドポイントを有効にする必要があります。受信の場合、STAT\_RX ビットをセットして受信を有効にする必要があります。BL\_SIZE および NUM\_BLOCK フィールドを使用して、割り当てられたバッファサイズを COUNTn\_RX に書き込む必要があります。単方向エンドポイントは、アイソクロナスおよびダブルバッファバルクエンドポイントを除き、サポートされる方向に関するビットとレジスタのみを初期化する必要があります。送信または受信が有効になると、レジスタ USB\_EPnR と位置 ADDRn\_TX/ADDRn\_RX、COUNTn\_TX/COUNTn\_RX (それぞれ) をアプリケーションソフトウェアで変更しないでください。これらの値はハードウェアが直接変更できます。データ転送操作が完了し、CTR 割り込みイベントによって通知されると、再びアクセスできるようになり、新しい操作を再び有効にできます。

## IN パケット (データ送信)

IN トークンパケットを受信するとき、受信したアドレスが設定済みの有効なエンドポイントと一致した場合、USB ペリフェラルは、アドレス指定されたエンドポイントに関連するバッファディスクリプタテーブルエントリ内の ADDRn\_TX および COUNTn\_TX 位置にアクセスします。これらの位置の内容は、内部 16 ビットレジスタ ADDR および COUNT に格納されます (ソフトウェアからはアクセスできません)。パケットメモリが再びアクセスされ、最初の送信バイトを読み出し ([パケットバッファの構造と用途 \(1059 ページ\)](#) を参照)、USB\_EPnR のビット DTOG\_TX に従って、DATA0 または DATA1 PID の送信を開始します。PID が完了すると、バッファメモリから読み出された最初のバイトが出力シフトレジスタにロードされ、USB バスで送信されます。最後のデータバイトが送信された後、計算された CRC が送信されます。アドレス指定されたエンドポイントが有効でなかった場合は、USB\_EPnR レジスタの STAT\_TX ビットに従って、NAK または STALL ハンドシェイクパケットがデータパケットの代わりに送信されます。

ADDR 内部レジスタは、現在のバッファメモリ位置を指すポインタとして使用され、COUNT は、残りの送信バイト数をカウントするために使用されます。パケットバッファメモリから読み出された各ハーフワードは、最下位バイトから順に USB バスで送信されます。送信バッファメモリは、COUNTn\_TX/2 ハーフワードの場合、ADDRn\_TX によって示されるアドレスから順に読み出されます。送信パケットが奇数個のバイトで構成される場合、最後にアクセスされたハーフワードの下半分のみが使用されます。

ホストが ACK を受信した場合、USB\_EPnR レジスタは次のように更新されます。DTOG\_TX ビットがトグルされ、STAT\_TX=10 (NAK) をセットすることによってエンドポイントが無効化され、ビット CTR\_TX がセットされます。アプリケーションソフトウェアは、まず、USB\_ISTR レジスタの EP\_ID および DIR ビットを調べることによって、マイクロコントローラの処理を要求しているエンドポイントを識別する必要があります。CTR\_TX イベントの処理は、割り込みビットをクリアすることから始まります。次に、アプリケーションソフトウェアは、送信データでいっぱい別のバッファを準備し、COUNTn\_TX テーブル位置を次の転送時に送信されるバイト数で更新し、最後に、STAT\_TX を 11 (VALID) にセットして、送信を再び有効にします。STAT\_TX ビットが 10 (NAK) に等しいときには、そのエンドポイントにアドレス指定された IN リクエストは NAK され、フロー制御条件を示します。USB ホストは、成功するまで、トランザクションを再試行します。CTR 割り込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された 2 番目の IN トランザクションの通知が失われるのを避けるためには、一連の操作を上記の順序で実行する必要があります。

## OUT および SETUP パケット (データ受信)

この 2 つのトークンは、USB ペリフェラルによっておおむね同じように扱われます。SETUP パケットの取り扱いにおける違いについては、制御転送に関する以下のパラグラフで詳しく述べます。OUT/SETUP PID を受信するとき、アドレスが有効なエンドポイントと一致した場合、USB ペリフェラルは、アドレス指定されたエンドポイントに関連するバッファディスクリプタテーブルエントリ内の ADDRn\_RX および COUNTn\_RX 位置の内容にアクセスします。ADDRn\_RX の内容は、内部レジスタ ADDR に直接格納されます。COUNT がリセットされ、BL\_SIZE および NUM\_BLOCK ビットフィールドの値が COUNTn\_RX 内に読み出されると、それを使用して BUF\_COUNT が初期化され、これは、バッファオーバーラン条件のチェックに使用される内部 16 ビットカウンタです (これらの内部レジスタはすべて、ソフトウェアからはアクセスできません)。その後、USB ペリフェラルによって受信されるデータバイトは、ハーフワード単位でパケット化され (最初の受信バイトが最下位バイトとして格納されます)、内部 ADDR レジスタに含まれるアドレスから順にパケットバッファに転送され、各バイト転送により、BUF\_COUNT はデクリメントされ、COUNT はインクリメントされます。DATA パケットの終わりが検出されると、受信された CRC の正しさがテストされ、受信時にエラーが発生しなかった場合のみ、ACK ハンドシェイクパケットが送信元ホストに送り返されます。

CRC が正しくなかった場合、または他の種類のエラー (ビットスタッフ違反、フレームエラーなど) が発生した場合、データバイトは、少なくともエラー検出ポイントまではパケットメモリにコピーされますが、ACK パケットは送信されず、USB\_ISTR レジスタの ERR ビットがセットされます。ただ

し、この場合、通常はソフトウェアのアクションは不要です。USB ペリフェラルは受信エラーから回復し、次のトランザクションを処理する準備ができた状態を保ちます。アドレス指定されたエンドポイントが有効でなかった場合は、USB\_EPnR レジスタの STAT\_RX ビットに従って、NAK または STALL ハンドシェイクパケットがデータパケットの代わりに送信され、データは受信メモリバッファに書き込まれません。

受信メモリバッファの位置は、ADDRn\_RX に含まれるアドレスから、CRC を含めた受信データパケット長に対応するバイト数 (データペイロード長 + 2) だけ、または、BL\_SIZE および NUM\_BLOCK によって定義された割り当て済みの最後のメモリ位置のいずれか先に来る方まで書き込まれます。このように、USB ペリフェラルは、割り当てられた受信メモリバッファ領域の終わりを超えて書き込むことはありません。データパケットペイロードの長さ (アプリケーションによって使用された実際のバイト数) が割り当てられたバッファより大きい場合、USB ペリフェラルはバッファオーバーラン条件を検出します。この場合、通常の ACK の代わりに STALL ハンドシェイクが送信されて、問題をホストに通知し、割り込みは生成されず、トランザクションは失敗したとみなされます。

ACK ハンドシェイクパケットを送信することにより、トランザクションが正常に完了すると、内部 COUNT レジスタがバッファ記述テーブルエントリ内の COUNTn\_RX 位置にコピーされ、影響を受けなかった BL\_SIZE および NUM\_BLOCK フィールドはそのままに保たれ (通常は再書き込みの必要はありません)、USB\_EPnR レジスタが次のように更新されます。DTOG\_RX ビットがトグルされ、STAT\_RX=10 (NAK) をセットすることによってエンドポイントが無効化され、ビット CTR\_RX がセットされます。エラーまたはバッファオーバーランによってトランザクションが失敗した場合、上記のアクションはいずれも実行されません。アプリケーションソフトウェアは、まず、USB\_ISTR レジスタの EP\_ID および DIR ビットを調べることによって、マイクロコントローラの処理を要求しているエンドポイントを識別する必要があります。CTR\_RX イベントの処理は、まず、トランザクションのタイプを決定することによって行われます (USB\_EPnR レジスタの SETUP ビット)。アプリケーションソフトウェアは、割り込みフラグビットをクリアして、処理されるエンドポイントに関連するバッファ記述テーブルエントリ内の COUNTn\_RX 位置を読み出して、受信バイト数を取得する必要があります。受信データが処理された後、アプリケーションは USB\_EPnR の STAT\_RX ビットを 11 (有効) にセットして、次のトランザクションを有効にする必要があります。STAT\_RX ビットが 10 (NAK) に等しいときには、そのエンドポイントにアドレス指定された OUT リクエストは NAK され、フロー制御条件を示します。USB ホストは、成功するまで、トランザクションを再実行します。CTR 割り込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された 2 番目の OUT トランザクションの通知が失われるのを避けるためには、一連の操作を上記の順序で実行する必要があります。

### 制御転送

制御転送は、SETUP トランザクション、すべて同じ方向のゼロ個以上のデータステージ、およびステータスステージ (反対方向のゼロバイトの転送) で構成されます。SETUP トランザクションは制御エンドポイントのみによって処理され、OUT トランザクション (データ受信) と非常に似ていますが、制御転送を初期化するために、アドレス指定されたエンドポイントレジスタの DTOG\_TX および DTOG\_RX ビットの値がそれぞれ 1 と 0 にセットされる点と、ソフトウェアが後続のトランザクションが IN か OUT かを SETUP の内容に応じて決定できるように、STAT\_TX と STAT\_RX の両方が 10 (NAK) にセットされる点が異なります。制御エンドポイントは CTR\_RX イベントごとに USB\_EPnR レジスタの SETUP ビットをチェックして、通常の OUT トランザクションと SETIP トランザクションを区別する必要があります。USB デバイスは、SETUP ステージで転送されたデータを解釈することによって、データステージの数と方向を判断でき、エラーの場合にはトランザクションを STALL する必要があります。そのためには、最後以外のすべてのデータステージで、未使用の方向を STALL にセットして、ホストが転送方向を反転するのが速過ぎた場合にはステータスステージとして STALL されるようにする必要があります。

最後のデータステージを有効にするときには、反対方向を NAK にセットして、ホストが (ステータスステージを実行するために) 転送方向をすぐに反転した場合、制御操作の完了を待つようにする必要があります。制御操作が正常に完了した場合、ソフトウェアは NAK を VALID に変更し、そうでな

い場合は STALL に変更します。同時に、ステータスステージが OUT の場合、STATUS\_OUT (USB\_EPnR レジスタの EP\_KIND) ビットをセットして、ステータストランザクションが非ゼロのデータで実行された場合はエラーが生成されるようにする必要があります。ステータストランザクションが処理されると、アプリケーションは STATUS\_OUT ビットをクリアし、STAT\_RX を VALID にセットし (新しいコマンドを受け入れるため)、STAT\_TX を NAK にセットします (次のセットアップの直後のステータスステージを遅延するため)。

USB 仕様では、SETUP パケットには ACK 以外のハンドシェイクでは応答できず、結果として、以前に発行されたコマンドをアポートして新しいコマンドを開始すると規定されているため、USB ロジックでは、制御エンドポイントはホストから受信した SETUP トークンに対して NAK または STALL パケットで応答することはできません。

STAT\_RX ビットが 01 (STALL) または 10 (NAK) にセットされ、SETUP トークンが受信されると、USB はデータを受け入れ、要求されたデータ転送を実行し、ACK ハンドシェイクを返送します。そのエンドポイントに以前に発行された CTR\_RX リクエストがあり、アプリケーションによってまだ確認応答されていない場合 (すなわち、CTR\_RX ビットが以前に完了した受信からセットされたままになっている場合)、USB は SETUP トランザクションを破棄し、状態にかかわらずハンドシェイクパケットで応答せず、受信エラーをシミュレートして、ホストが SETIP トークンを再び送信しなければならないようにします。これは、CTR\_RX 割り込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された SETUP トランザクションの通知が失われるのを避けるために行われます。

### 32.5.3 ダブルバッファエンドポイント

USB 標準によって定義されているすべてのエンドポイントタイプは、さまざまなトラフィックモデルを表し、各種のデータ転送操作の典型的な要件を記述しています。データの大部分をホスト PC と USB 機能の間で転送する必要があるときには、バルクエンドポイントタイプが最適のモデルです。ホストはバルクトランザクションをスケジューリングして、フレーム内の使用可能な帯域幅のすべてをいっぱいにして、アドレス指定された USB 機能がバルクトランザクションを処理できる限り、実際の転送速度を最大化します。次のトランザクションが着信したときに、USB 機能が前のトランザクションでビジーであった場合は、NAK ハンドシェイクで応答し、ホスト PC は、USB 機能が処理できるようになるまで、同じトランザクションを再び発行するので、再送信によって帯域幅が占有されるため、実際の転送速度は低下します。このため、バルクエンドポイントでは、「ダブルバッファリング」と呼ばれる専用機能を使用できます。

「ダブルバッファリング」が有効なときには、データトグルシーケンシングが使用されて、USB ペリフェラルが要求されたデータ転送を実行するために使用するバッファが選択され、送信と受信の両方のパケットメモリ領域を使用して、トランザクションの成功ごとのバッファスワッピングを管理するため、アプリケーションは常に完全なバッファを使用でき、USB ペリフェラルは他方のバッファをいっぱいにします。たとえば、「受信」ダブルバッファバルクエンドポイント宛ての OUT トランザクション時、1つのバッファが USB ホストから着信する新しいデータで満たされている間、もう1つのバッファはマイクロコントローラソフトウェアが使用できます (「送信」ダブルバッファバルクエンドポイントと IN トランザクションでも同様です)。

スワップバッファ管理には、アドレスポイントと割り当て済みメモリバッファの長さをホストする 4 つすべてのバッファ記述テーブル位置を使用する必要があるため、ダブルバッファバルクエンドポイントの実装に使用される USB\_EPnR レジスタは、強制的に単方向として使用されます。したがって、1つの STAT ビットのみを 00 (無効) 以外の値にセットする必要があります。ダブルバッファバルクエンドポイントが受信について有効な場合は STAT\_RX、ダブルバッファバルクエンドポイントが送信について有効な場合は STAT\_TX。ダブルバッファバルクエンドポイントを受信と送信の両方について有効にする必要がある場合は、2つの USB\_EPnR レジスタを使用する必要があります。

ダブルバッファリング機能を利用して、可能な限り高い転送速度を達成するには、前の章で説明したエンドポイントフロー制御構造を変更して、トランザクションの正常官僚の旅ではなく、USB ペリフェラルとアプリケーションの間でバッファコンフリクトが発生したときのみ、エンドポイントの



ステータスを NAK に切り替える必要があります。USB ペリフェラルによって現在使用されているメモリバッファは、エンドポイント方向に関連する DTOG ビットによって定義されます。すなわち、「受信」ダブルバッファバルクエンドポイントの場合は DTOG\_RX (USB\_EPnR レジスタのビット 14)、または「送信」ダブルバッファバルクエンドポイントの場合は DTOG\_TX (USB\_EPnR レジスタのビット 6) です。新しいフロー制御方式を実装するには、USB ペリフェラルはアプリケーションソフトウェアによって使用されているパケットバッファを知り、コンフリクトに注意する必要があります。USB\_EPnR レジスタには 2 つの DTOG ビットがありますが、データおよびバッファシーケンシングのために USB ペリフェラルによって使用されるのは 1 つだけであり (ダブルバッファリング機能に必要な単方向成約のため)、もう 1 つはアプリケーションソフトウェアが、使用中のバッファを示すために使用できます。この新しいバッファフラグは、SW\_BUF と呼ばれます。次の表で、「送信」および「受信」ダブルバッファバルクエンドポイントの場合の USB\_EPnR レジスタのビットと DTOG/SW\_BUF 定義の対応を説明します。

**表 175. ダブルバッファリングバッファフラグの定義**

バッファフラグ	「送信」エンドポイント	「受信」エンドポイント
DTOG	DTOG_TX (USB_EPnR ビット 6)	DTOG_RX (USB_EPnR ビット 14)
SW_BUF	USB_EPnR ビット 14	USB_EPnR ビット 6

USB ペリフェラルによって現在使用されているメモリバッファは、DTOG バッファフラグによって定義され、アプリケーションソフトウェアによって現在使用されているバッファは、SW\_BUF バッファフラグによって示されます。バッファフラグの値と使用されているパケットバッファの関係は、どちらの場合も同じであり、次の表に示されています。

**表 176. バルクダブルバッファリングメモリバッファの用途**

エンドポイントタイプ	DTOG	SW_BUF	USB ペリフェラルによって使用されるパケットバッファ	アプリケーションソフトウェアによって使用されるパケットバッファ
IN	0	1	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル位置。
	1	0	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル位置	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。
	0	0	なし <sup>(1)</sup>	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。
	1	1	なし <sup>(1)</sup>	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。
OUT	0	1	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置。	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置。
	1	0	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置。	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置。
	0	0	なし <sup>(1)</sup>	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置。
	1	1	なし <sup>(1)</sup>	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置。

1. NAK ステータスのエンドポイント。



バルクエンドポイントのダブルバッファリング機能は、次のようにして有効化されます。

- USB\_EPnR レジスタの EP\_TYPE ビットフィールドに 00 を書き込み、エンドポイントをバルクとして定義します。
- 同じレジスタの EP\_KIND ビットを 1 (DBL\_BUF) にセットします。

アプリケーションソフトウェアは、最初に使用されるバッファに応じて、DTOG および SW\_BUF ビットを初期化する責任があります。これは、この 2 つのビットが持つ特殊なトグル専用プロパティを考慮して行う必要があります。DBL\_BUF のセット後に発生した初めてのトランザクションの終了により、ダブルバッファバルクエンドポイントの特殊なフロー制御がトリガされ、DBL\_BUF がセットされたままである限り、このエンドポイントにアドレス指定された他のすべてのトランザクションについて、これが使用されます。各トランザクションの終了時、有効化された方向に応じて、アドレス指定されたエンドポイント USB\_EPnR レジスタの CTR\_RX または CTR\_TX ビットがセットされます。同時に、USB\_EPnR レジスタの影響を受けた DTOG ビットがハードウェアによってトグルされ、USB ペリフェラルのバッファスワッピングを完全にソフトウェアから独立させます。通常のトランザクション、および DBL\_BUF のセット後の初めてのトランザクションと違って、STAT ビットペアはトランザクション終了の影響を受けず、値は 11 (有効) のままです。ただし、新しいトランザクションのトークンパケットが受信されると、実際のエンドポイントステータスは、USB ペリフェラルとアプリケーションソフトウェアの間のバッファコンフリクトが検出されたときには 10 (NAK) としてマスクされます (この条件は DTOG と SW\_BUF が同じ値を持つことによって示されます。表 176 (1064 ページ) を参照)。アプリケーションソフトウェアは、CTR イベント通知に応答して、割り込みフラグをクリアし、完了したトランザクションの要求された処理を開始します。アプリケーションパケットバッファの仕様が終了すると、ソフトウェアは SW\_BUF ビットに 1 を書き込んでトグルし、そのバッファが使用可能になったことを USB ペリフェラルに通知します。このように、NAK されるトランザクションの数は、トランザクションデータのアプリケーション得ラポレーション時間によってのみ制限されます。エラーポレーション時間が USB バスでのトランザクションの完了に必要な時間より短かった場合、フロー制御による再送信は行われず、実際の転送速度はホスト PC によってのみ制限されます。

アプリケーションソフトウェアは、11 (有効) とは異なる明示的なステータスを、関連する USB\_EPnR レジスタの STAT ビットペアに書き込むことによって、ダブルバッファバルクエンドポイントのために実装される特殊なフロー制御をオーバーライドできます。この場合、USB ペリフェラルは、バッファ使用条件に関係なく、プログラムされたエンドポイントステータスを常に使用します。

## 32.5.4 アイソクロナス転送

USB 標準は、固定された正確なデータ生成/消費周波数を必要とするフルスピードペリフェラルをサポートし、この種類のトラフィックを「アイソクロナス」として定義しています。このデータの典型的な例は、オーディオサンプル、圧縮ビデオストリーム、および一般に、供給される周波数の正確さについて厳密な要件を持つ、あらゆる種類のサンプルデータです。エニューメレーションフェーズでエンドポイントが「アイソクロナス」として定義されると、ホストは必要な帯域幅をフレームに割り当て、エンドポイントの方向に応じて、各フレームに正確に 1 つの IN または OUT パケットを供給します。帯域幅要件を制限するために、アイソクロナストラフィックについては、失敗したトランザクションの再送信はできません。このため、アイソクロナストランザクションにはハンドシェイクフェーズはなく、ACK パケットは予期されず、データパケット後に送信されません。同じ理由で、アイソクロナス転送はデータトグルシーケンシングをサポートせず、常に DATA0 PID を使用してデータパケットを開始します。

エンドポイントのアイソクロナス動作は、USB\_EPnR レジスタの EP\_TYPE ビットを 10 にセットすることによって選択されます。ハンドシェイクフェーズがないため、STAT\_RX/STAT\_TX ビットの唯一の正当な値は 00 (無効) と 11 (有効) であり、他の値の場合は、USB 標準に準拠しない結果になります。アイソクロナスエンドポイントはダブルバッファリングを実装して、アプリケーションソフトウェア開発を容易にし、「送信」と「受信」の両方のパケットメモリ領域を使用して、トランザクションの成功のたびにバッファスワッピングを管理し、アプリケーションは常に完全なバッファを使用でき、USB ペリフェラルはもう 1 つのバッファを満たします。



USB ペリフェラルによって現在使用されているメモリバッファは、エンドポイント方向に関する DTOG ビットによって、表 177 に従って定義されます (「受信」アイソクロナスエンドポイントの場合は DTOG\_RX、「送信」アイソクロナスエンドポイントの場合は DTOG\_TX であり、両方とも、関連する USB\_EPnR レジスタ)。

表 177. アイソクロナスメモリバッファの使用

エンドポイントタイプ	DTOG ビットの値	使用されるパケットバッファ (USB ペリフェラル)	使用されるパケットバッファ (アプリケーションソフトウェア)
IN	0	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル位置
	1	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル位置	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置
OUT	0	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置
	1	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置

ダブルバッファアイソクロナスエンドポイントで発生すると、アイソクロナスエンドポイントの実装に使用される USB\_EPnR レジスタは強制的に単方向として使用されます。アイソクロナスエンドポイントを受信と送信の両方について有効にする必要がある場合は、2 つの USB\_EPnR レジスタを使用する必要があります。

アプリケーションソフトウェアは、最初に使用されるバッファに応じて、DTOG ビットを初期化する責任があります。これは、この 2 つのビットが持つ特殊なトグル専用プロパティを考慮して行う必要があります。各トランザクションの終了時、有効化された方向に応じて、アドレス指定されたエンドポイント USB\_EPnR レジスタの CTR\_RX または CTR\_TX ビットがセットされます。同時に、USB\_EPnR レジスタの影響を受けた DTOG ビットがハードウェアによってトグルされ、バッファスワッピングを完全にソフトウェアから独立させます。STAT ビットペアはトランザクションの完了に影響を受けません。アイソクロナス転送ではハンドシェイクフェーズがないため、フロー制御は不可能であり、エンドポイントは常に 11 (有効) のままです。CRC エラーまたはバッファオーバーラン条件がアイソクロナス OUT 転送時に発生した場合、常に正しいトランザクションとみなされ、常に CTR\_RX イベントをトリガします。ただし、CRC エラーが発生すると、常に USB\_ISTR レジスタの ERR ビットがセットされて、データ破損の可能性をソフトウェアに通知します。

## 32.5.5 サスペンド/レジュームイベント

USB 標準では、SUSPEND と呼ばれる特殊なペリフェラル状態が定義され、USB バスからの平均電流は 2.5 mA より大きくなってはなりません。この要件はバスパワードデバイスにとって基本的な重要性を持ちますが、セルフパワードデバイスは、この厳格な消費電力成約を順守する必要はありません。サスペンドモードでは、ホスト PC は 3 ms を超えて USB バスにトラフィックを送信しないことによって通知を送信します。SOF パケットは、通常動作では 1 ms ごとに送信されなければならないので、USB ペリフェラルは 3 回連続の SOF パケットの欠落をホスト PC からのサスペンドリクエストとして検出し、USB\_ISTR レジスタの SUSP ビットを 1 にセットし、有効な場合は割り込みが生成されます。デバイスがサスペンドされると、いわゆる RESUME シーケンスによって通常操作が復元され、これはホスト PC から、またはペリフェラルそのものから直接開始できますが、常にホスト PC によって終了されます。サスペンドされた USB ペリフェラルは、常に RESET シーケンスを検出して、通常の USB リセットイベントとして、このイベントに対応できなければなりません。

USB ペリフェラルをサスペンドするために使用される実際の手順は、デバイスの構成によって合計消費量を削減するために必要なアクションが異なるため、デバイスに依存します。

USB ペリフェラルの SUSP 通知に回答するアプリケーションソフトウェアルーチンの USB 関連の局面を中心にして、典型的なサスペンド手順の簡単な説明を以下に示します。

1. USB\_CNTR レジスタの FSUSP ビットを 1 にセットします。このアクションによって、USB ペリフェラル内のサスペンドモードが有効になります。サスペンドモードが有効になると、USB がサスペンドされている間に SUSP 割り込みが発行されるのを回避するために、SOF 受信時のチェックが無効になります。
2. USB ペリフェラルとは異なるブロックの静的消費電力を除去または削減します。
3. USB\_CNTR レジスタの LP\_MODE ビットを 1 にセットして、アナログ USB トランシーバの静的消費電力を除去しつつ、レジュームアクティビティを検出できるままにしておきます。
4. オプションで、外部オシレータとデバイス PLL をオフにして、デバイス内部のアクティビティを停止します。

デバイスが SUSPEND モード中に USB イベントが発生したときには、RESUME 手順が呼び出されて、通常のチェックと復元し、通常の USB 動作を復元する必要があります。ウェイクイベントが USB リセットシーケンスである時には、このプロセスが 10 ms 以内で終了するように特に注意が必要です (詳細については、「ユニバーサルシリアルバス仕様」を参照してください)。USB ペリフェラルのサスペンド中にレジュームまたはリセットシーケンスが開始されると、USB\_CNTR レジスタの LP\_MODE ビットが非同期でクリアされます。このイベントによって WKUP 割り込みがトリガされた場合でも、システムクロックの再開による長い遅延のため、割り込み応答ルーチンの使用は慎重に評価する必要があります。公称クロックの再有効化の前の遅延を短くするには、サスペンド手順の終了直後にレジューム手順を行って、システムクロックが再開したらすぐにコードが実行されるようにします。システムをウェイクアップする際の ESD 放電またはその他の種類のノイズを防止するために (サスペンドモードの終了は、非同期イベントです)、サスペンド中にデータラインステータスの適切なアナログフィルタを有効にします。フィルタ幅は約 70 ns です。

以下は、レジューム手順で行うべきアクションのリストです。

1. オプションで、外部オシレータおよびデバイス PLL をオンにします。
2. USB\_CNTR レジスタの FSUSP ビットをクリアします。
3. レジュームをトリガしたイベントを識別する必要がある場合は、USB\_FNR レジスタのビット RXDP および RXDM を [表 178](#) に従って使用でき、この表には、あらゆる場合のソフトウェアのアクションもリストされています。必要な場合には、上記のビットのステータスを監視することによって、レジュームまたはリセットシーケンスの終了を検出でき、これらが「10」設定 (アイドルバス状態を表します) に達したことをチェックします。さらに、リセットシーケンスの終了時には、USB\_ISTR レジスタの RESET ビットが 1 にセットされ、有効な場合は割り込みが発行され、通常通りに処理されます。

表 178. レジュームイベント検出

[RXDP,RXDM] ステータス	ウェイクアップイベント	必要なレジュームソフトウェアアクション
"00"	ルートのリセット	なし
"10"	なし (バス上のノイズ)	サスペンドモードに戻る
"01"	ルートのレジューム	なし
"11"	不可 (バス上のノイズ)	サスペンドモードに戻る

デバイスは、USB プロトコルに直接関係しない特定のイベントに対する応答として、サスペンドモードを終了しなければならない場合があります (たとえば、マウスの移動はシステム全体をウェイクアップします)。この場合、USB\_CNTR レジスタの RESUME ビットを 1 にセットし、1 ms から 15 ms までのインターバル後に 0 にリセットすることによって、レジュームシーケンスを開始できます (このインターバルは、システムクロックが公称周波数で動作しているときには 1 ms 周期で発生する ESOF 割り込みによって計測できます)。RESUME ビットがクリアされると、レジュームシーケンスはホスト PC によって完了され、その終了は、USB\_FNR レジスタの RXDP および RXDM ビットを使用して監視できます。

**注:** RESUME ビットは、USB ペリフェラルがサスペンドモードにされた (USB\_CNTR レジスタの FSUSP ビットを 1 にセットした) 後でのみ使用する必要があります。

## 32.6 USBレジスタ

USB ペリフェラルレジスタは、以下のグループに分けることができます。

- 共通レジスタ：割り込みおよび制御レジスタ。
- エンドポイントレジスタ：エンドポイント設定およびステータス
- バッファディスクリプタテーブル：データバッファの配置に使用されたパケットメモリの位置

すべてのレジスタアドレスは、USB ペリフェラルレジスタのベースアドレス 0x4000 5C00 に対するオフセットとして表されますが、バッファディスクリプタテーブル位置は USB\_BTABLE レジスタによって指定されたアドレスから始まります。すべてのレジスタアドレスが 16 ビットであっても 32 ビットワード境界に整列されます。「1 x 16 ビット/ワード」のアクセス方式を持つデバイスでは、0x4000 6000 から始まる箇所にあるパケットバッファメモリ位置にアクセスするために同じアドレス整列を使用します。

レジスタの説明で使用されている略語のリストについては、[セクション 2.1 \(45 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

### 32.6.1 共通レジスタ

これらのレジスタは USB ペリフェラルの全般的動作に影響し、動作モード、割り込み処理、デバイスアドレス、およびホスト PC によって更新される現在のフレーム番号へのアクセス付与を定義します。

#### USB 制御レジスタ (USB\_CNTR)

アドレスオフセット：0x40

リセット値：0x0003

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR M	PMAOVR M	ERR M	WKUP M	SUSP M	RESET M	SOF M	ESOF M	L1REQ M	Res M	L1 RESUME	RE SUME	F SUSP	LP_ MODE	PDW N	F RES
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w

ビット 15 **CTRM**：正しい転送割り込みマスク

0：正しい転送 (CTR) 割り込みは無効です。

1：CTR 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 14 **PMAOVRM**：パケットメモリ領域のオーバー/アンダーラン割り込みマスク

0：PMAOVR 割り込みは無効です。

1：PMAOVR 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 13 **ERRM**：エラー割り込みマスク

0：ERR 割り込みは無効です。

1：ERR 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 12 **WKUPM**：ウェイクアップ割り込みマスク

0：WKUP 割り込みは無効です。

1：WKUP 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。



### ビット 11 **SUSPM** : サスペンドモード割り込みマスク

0 : サスペンドモードリクエスト (SUSP) 割り込みは無効です。

1 : SUSP 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

### ビット 10 **RESETM** : USB リセット割り込みマスク

0 : RESET 割り込みは無効です。

1 : RESET 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

### ビット 9 **SOFM** : フレーム開始割り込みマスク

0 : SOF 割り込みは無効です。

1 : SOF 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

### ビット 8 **ESOFM** : 予期されるフレーム開始割り込みマスク

0 : 予期されるフレーム開始 (ESOF) 割り込みは無効です。

1 : ESOF 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

### ビット 7 **L1REQM** : LPM L1 状態リクエスト割り込みマスク

0 : LPM L1 状態リクエスト (L1REQ) 割り込みマスク無効

1 : L1REQ 割り込みは有効であり、USB\_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

**注 :** LPM がサポートされない場合、このビットは実装されておらず、予約済みとみなされます。セクション 32.3 : USB の実装を参照してください。

### ビット 6 予約済み。

### ビット 5 **L1RESUME** : LPM L1 レジュームリクエスト

マイクロコントローラはこのビットをセットして、ホストに LPM L1 レジューム信号を送信できます。このビットは、信号の送信終了後にハードウェアによってクリアされます。

**注 :** LPM がサポートされない場合、このビットは実装されておらず、予約済みとみなされます。セクション 32.3 : USB の実装を参照してください。

### ビット 4 **RESUME** : レジュームリクエスト

マイクロコントローラはこのビットをセットして、ホストにレジューム信号を送信できます。USB 仕様に従って、ホスト PC がレジュームシーケンスを終了まで駆動する準備ができてから 1 ms 以上 15 ms 以下で有効にする必要があります。

### ビット 3 **FSUSP** : 強制サスペンド

ソフトウェアは、SUSP 割り込みが受信されたときにこのビットをセットする必要があり、USB ペリフェラルが 3 ms 間、トラフィックを受信しなかったときに発行されます。

0 : 影響はありません。

1 : サスペンドモードに入ります。アナログトランシーバのクロックと静的消費電流は影響を受けません。サスペンド消費電力が要件の場合 (バスパワーデバイス)、アプリケーションソフトウェアは FSUSP の後、以下の説明に従って LP\_MODE ビットをセットする必要があります。

**ビット 2 LP\_MODE** : 低電力モード

このモードは、サスペンドモードの電力制約から、外部プルアップレジスタに供給される電力を除き、すべての静的消費電流を回避されるときに使用されます。USB サスペンド条件の消費電力要件を満たすために、アプリケーションがすべてのシステムクロックを停止するか、周波数を削減する準備ができたときに、この条件に入るべきです。サスペンドモード中に USB アクティビティ (WKUP イベント) が発生すると、このビットは非同期でリセットされます (ソフトウェアによってリセットすることもできます)。

- 0 : 低電力モードではありません。
- 1 : 低電力モードに入ります。

**ビット 1 PDWN** : パワーダウン

このビットは、何らかの理由で USB ペリフェラルを完全に無効にする必要がある場合、USB 関連のすべてのアナログ部品を完全にオフにするために使用されます。このビットがセットされると、USB ペリフェラルはトランシーバから切断され、使用できなくなります。

- 0 : パワーダウンを終了します。
- 1 : パワーダウンモードに入ります。

**ビット 0 FRES** : 強制的 USB リセット

- 0 : USB リセットをクリアします。
- 1 : USB に RESET 信号が送られたときと同様に、USB ペリフェラルを強制的にリセットします。ソフトウェアがこのビットをクリアするまで、USB ペリフェラルは RESET 状態に保たれます。有効な場合は、「USB-RESET」割り込みが生成されます。

## USB 割り込みステータスレジスタ (USB\_ISTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMA OVR	ERR	WKUP	SUSP	RESET	SOF	ESOF	L1REQ	Res.	Res.	DIR	EP_ID[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0			r	r	r	r	r

このレジスタはすべての割り込みソースのステータスを含むため、アプリケーションソフトウェアは割り込みリクエストの原因になったイベントを判断できます。

このレジスタの上位部分はシングルビットを含み、そのそれぞれが特定のイベントを表します。これらのビットは、関連するイベントが発生したときにハードウェアによってセットされます。USB\_CNTR レジスタの対応するビットがセットされている場合、汎用割り込みリクエストが生成されます。各ビットを調べる割り込みルーチンは、必要なすべてのアクションを実行し、最後に処理済みビットをクリアします。そのいずれかがクリアされていない場合、割り込みはまだ保留中であるとみなされ、割り込みラインは再びハイに保たれます。複数のビットが同時にセットされた場合、1 つの割り込みだけが生成されます。

エンドポイントトランザクションの完了は、割り込み応答遅延を軽減するために、さまざまな方法で処理できます。CTR ビットは、エンドポイントがトランザクションを正常に完了するとすぐにハードウェアによってセットされ、USB\_CNTR の対応するビットがセットされていた場合は汎用割り込みリクエストが生成されます。エンドポイント専用割り込み条件は、USB\_CNTR レジスタの CTRM ビットとは無関係に有効化されます。どちらの割り込み条件も、ソフトウェアが対応する USB\_EPnR レジスタの保留中のビットをクリアするまでアクティブなままです (CTR ビットは実際には読み出し専用ビットです)。エンドポイント関連の割り込みの場合、ソフトウェアは、トランザクション方向 (DIR) および EP\_ID 読み出し専用ビットを使用して、最後の割り込みリクエストを発行して、対応する割り込みサービスルーチンを呼び出したエンドポイントを特定できます。



ユーザは、ソフトウェアが割り込みサービスルーチンで USB\_ISTR ビットをチェックする順序を指定することによって、同時に保留中の USB\_ISTR ビットの相対的優先順位を選ぶことができます。イベントに関連し、処理済みのビットだけがクリアされます。サービスルーチンの終了時、残りの条件を処理するために、別の割り込みがリクエストされます。

ビットの誤ったクリアを避けるために、ロード命令付きでクリアすることが推奨され、その場合、変更してはならないすべてのビットに 1 が書き込まれ、クリアすべきすべてのビットに 0 が書き込まれます (これらのビットはソフトウェアによってのみクリアできます)。読み出し-変更-書き込みのサイクルは避けるべきです。読み出し操作と書き込み操作の間に、別のビットがハードウェアによってセットされる可能性があり、マイクロプロセッサにイベントを処理する時間ができる前に、次の書き込みによってクリアされる可能性があるためです。

以下に、各ビットについて詳しく説明します。

### ビット 15 CTR : 正しい転送

このビットは、エンドポイントがトランザクションを正常に完了したことを示すために、ハードウェアによってセットされます。DIR および EP\_ID ビットを使用して、ソフトウェアは割り込みをリクエストしたエンドポイントを判断できます。このビットは読み出し専用です。

### ビット 14 PMAOVR : パケットメモリ領域のオーバー/アンダーラン

このビットは、マイクロコントローラに USB メモリリクエストに対応する時間がなくなった場合にセットされます。USB ペリフェラルは、このイベントを次のように処理します。受信時、ACK ハンドシェイクパケットは送信されず、送信時、送信ストリームのビットスタッフエラーが強制的に発行されます。いずれの場合も、ホストはトランザクションを再試行します。通常操作では、PMAOVR 割り込みは発生しません。失敗したトランザクションはホストによって再試行されるので、アプリケーションソフトウェアには、この割り込み処理中にデバイス操作をスピードアップして、次のトランザクションの差異施行に備える機会があります。ただし、アイソクロナス転送では (アイソクロナストランザクションは再試行されません)、データの喪失につながるため、これは行われません。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

### ビット 13 ERR : 誤差

このフラグは、下記にリストされているエラーのいずれかが発生したときにセットされます。  
NANS : 応答なし (No ANSwer)。ホスト応答がタイムアウトになりました。  
CRC : 巡回冗長検査 (Cyclic Redundancy Check) エラー。トークンまたはデータで受信された CRC の 1 つが正しくありませんでした。  
BST : ビットスタッフエラー。ビットスタッフエラーが、PID、データ、または CRC で検出されました。  
FVIO : フレーミングフォーマット違反。非標準のフレームが受信されました (正しい位置にない EOP、正しくないトークンシーケンスなど)。  
USB ペリフェラルと PC ホストは、エラーの場合、完全にトランスペアレントな方法で再送信を管理するため、USB ソフトウェアは、通常、エラーを無視できます。この割り込みは、ソフトウェア開発フェーズで、または USB バスでの送信品質の監視に役立ち、問題の可能性をユーザに知らせることができます (コネクタの緩み、ノイズが多い環境、USB ケーブルの導体の破損など)。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

### ビット 12 WKUP : ウェイクアップ

このビットは、サスペンドモード中に、USB ペリフェラルをウェイクアップするアクティビティが検出されたときにハードウェアによって 1 にセットされます。このイベントは CTRLR レジスタの LP\_MODE ビットを非同期でクリアし、USB\_WAKEUP ラインを有効にし、レジュームプロセスの開始に関して残りのデバイス (ウェイクアップユニットなど) に通知するために使用されます。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。



**ビット 11 SUSP** : サスペンドモードリクエスト

このビットは、3 ms にわたってトラフィックが受信されず、USB バスからのサスペンドモードリクエストを示したときに、ハードウェアによってセットされます。サスペンド条件チェックは、USB リセットの直後に有効化され、サスペンドモードがアクティブなとき (FSUSP=1)、レジャームシーケンスの終了まで、ハードウェアによって無効にされます。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

**ビット 10 RESET** : USB リセットリクエスト

USB ペリフェラルが入力でアクティブな USB RESET 信号を検出したときにセットされます。USB ペリフェラルは、RESET に対応して、内部プロトコル状態マシンをリセットし、USB\_CNTR レジスタの RESETM 有効ビットがセットされている場合、割り込みが生成されます。RESET ビットがクリアされるまで、受信と送信は無効です。すべての設定レジスタがリセットされるわけではありません。マイクロコントローラはこれらのレジスタを明示的にクリアする必要があります (これは、RESET 割り込みの安全な配信を確保し、RESET の直前のトランザクションを完了できるようにするためです)。機能アドレスおよびエンドポイントレジスタは、USB リセットイベントによってリセットされます。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

**ビット 9 SOF** : フレーム開始

このビットは、新しい USB フレームの開始を知らせ、SOF パケットが USB バス経由で着信したときにセットされます。割り込みサービスルーチンによって SOF イベントを監視して、USB ホストへの 1 ms の同期イベントを実行し、SOF パケット受信時に更新される USB\_FNR レジスタを安全に読み出すことができます (これはアイソクロナスアプリケーションに便利です)。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

**ビット 8 ESOF** : 予期されたフレーム開始

このビットは、予期された SOF パケットが受信されなかったときにハードウェアによってセットされます。ホストは 1 ms ごとに SOF パケットを送信しますが、ハブが正しく受信しなかった場合、サスペンドタイマがこの割り込みを発行します。3 回連続して ESOF 割り込みが生成され (すなわち、3 つの SOF パケットが失われ)、その間にトラフィックが発生しなかった場合、SUSP 割り込みが生成されます。このビットは、サスペンドタイマがまだロックされていないときに SOF パケットの喪失が発生したときにセットされます。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

**ビット 7 L1REQ** : LPM L1 状態リクエスト

このビットは、L1 状態を入力する LPM コマンドが正常に受信され、確認応答された場合に、ハードウェアによってセットされます。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

**注:** LPM がサポートされない場合、このビットは実装されておらず、予約済みとみなされます。セクション 32.3 : USB の実装を参照してください。

ビット 6:5 予約済み。

**ビット 4 DIR** : トランザクションの方向

このビットは、割り込みリクエストを生成した成功したトランザクションの方向に従って、ハードウェアによって書き込まれます。

DIR ビット =0 の場合、割り込みをかけたエンドポイントに関連する USB\_EPnR レジスタの CTR\_TX ビットがセットされます。割り込みの原因となったトランザクションは、IN タイプです (USB ペリフェラルからホスト PC へのデータ送信)。

DIR ビット =1 の場合、割り込みをかけたエンドポイントに関連する USB\_EPnR レジスタの CTR\_RX ビットまたは CTR\_TX と CTR\_RX の両方がセットされます。割り込みの原因となったトランザクションは OUT タイプであるか (ホスト PC からのデータを USB ペリフェラルが受信)、または 2 つの保留中のトランザクションが処理を待っています。

これは割り込みを保留中にした方向を表すので、アプリケーションソフトウェアはこの情報を使用して、トリガしたトランザクションに関連する USB\_EPnR ビットにアクセスできます。このビットは読み出し専用です。

**ビット 3:0 EP\_ID[3:0]** : エンドポイント識別子

これらのビットは、割り込みリクエストを生成したエンドポイント番号に従って、ハードウェアによって書き込まれます。複数のエンドポイントトランザクションが保留中の場合、ハードウェアは、次のように定義される最も高い優先順位を持つエンドポイントに関連するエンドポイント識別子を書き込みます。優先順位の順に、2 つのエンドポイントセットが定義されます。アイソクロナスおよびダブルバッファバブルクエンドポイントが最初に考慮され、次に、その他のエンドポイントが調べられます。同じセットの複数のエンドポイントが割り込みをリクエストしている場合、最下位リクエストエンドポイントレジスタに従って、USB\_ISTR レジスタの EP\_ID ビットが割り当てられ、EP0R が最も高い優先順位を持ち、その後 EP1R が続き、以下同様です。アプリケーションソフトウェアは、この優先順位方式に従って、レジスタを各エンドポイントに割り当てて、同時発生したエンドポイントリクエストを適切に順序づけることができます。これらのビットは読み出し専用です。

## USB フレーム番号レジスタ (USB\_FNR)

アドレスオフセット : 0x48

リセット値 : 0x0XXX (X は未定義)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP	RXDM	LCK	LSOF[1:0]		FN[10:0]										
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

**ビット 15 RXDP** : 受信データ + ラインステータス

このビットを使用して、受信データプラスアップストリームポートデータラインのステータスを観測できます。サスペンド終了ルーチン時に使用して、ウェイクアップイベントの決定を容易にできます。

**ビット 14 RXDM** : 受信データ - ラインステータス

このビットを使用して、受信データマイナスアップストリームポートデータラインのステータスを観測できます。サスペンド終了ルーチン時に使用して、ウェイクアップイベントの決定を容易にできます。

ビット 13 **LCK** : ロック

このビットは、USB リセット条件の終了後、または USB レジュームシーケンスの終了後に、少なくとも 2 つ連続した SOF パケットが受信されたときに、ハードウェアによってセットされます。ロックされると、USB リセットまたは USB サスペンドイベントが発生するまで、フレームタイムはこの状態に保たれます。

ビット 12:11 **LSOF[1:0]** : SOF の喪失

これらのビットは、ESOF 割り込みが生成されたときに、ハードウェアによって書き込まれ、失われた連続 SOF パケットの数をカウントします。SOF パケットの受信時、これらのビットはクリアされます。

ビット 10:0 **FN[10:0]** : フレーム番号

このビットフィールドは、最後に受信された SOF パケットに含まれる 11 ビットのフレーム番号を含みます。フレーム番号は、ホストによって送信されたフレームごとにインクリメントされ、アイソクロナス転送の場合に便利です。このビットフィールドは、SOF 割り込みの生成時に更新されます。

## USB デバイスアドレス (USB\_DADDR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済み

ビット 7 **EF** : 機能有効

このビットは、USB デバイスを有効にするために、ソフトウェアによってセットされます。このデバイスのアドレスは、次の ADD[6:0] ビットに含まれます。このビットが 0 の場合、USB\_EPnR レジスタの設定にかかわらず、トランザクションは処理されません。

ビット 6:0 **ADD[6:0]** : デバイスアドレス

これらのビットは、エnumerationプロセス時にホスト PC によって割り当てられた USB 機能を含みます。必要なエンドポイントへのトランザクションを処理するためには、関連する USB\_EPnR レジスタのこのフィールドとエンドポイントアドレス (EA) フィールドの両方とも、USB トークンに含まれる情報に一致する必要があります。

## バッファテーブルアドレス (USB\_BTABLE)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTABLE[15:3]													Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW			



ビット 15:3 **BTABLE[15:3]** : バッファテーブル

これらのビットは、専用パケットメモリ内のバッファ割り当てテーブルの開始アドレスを含みます。このテーブルは、各エンドポイントバッファ位置とサイズを記述し、8 バイトの境界に揃えられなければなりません (3 つの最下位ビットが常に 0)。このデバイスにアドレス指定されたトランザクションの開始ごとに、USB ペリフェラルはアドレス指定されたエンドポイントに関連するこのテーブルの要素を読み出して、バッファ開始位置とバッファサイズを取得します ([パケットバッファの構造と用途 \(1059 ページ\)](#) を参照)。

ビット 2:0 予約済み、ハードウェアによって 0 に固定されています。

## LPM 制御およびステータスレジスタ (USB\_LPMCSR)

アドレスオフセット : 0x54

リセット値 : 0x0000

**注 :** **LPM がサポートされない場合、このビットは実装されておらず、予約済みとみなされます。セクション 32.3 : USB の実装を参照してください。**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]			Res.	REM WAKE	LPM ACK	LPM EN	
								r				r	rw	rw	

ビット 15:8 予約済み。

ビット 7:4 **BESL[3:0]** : BESL 値

これらのビットは、最後に ACK された LPM トークンとともに受信した BESL 値を含みます。

ビット 3 **REM WAKE** : bRemoteWake 値

このビットは、最後に ACK された LPM トークンとともに受信した bRemoteWake 値を含みます。

ビット 2 予約済み

ビット 1 **LPMACK** : LPM トークン確認応答イネーブル

0 : 有効な LPM トークンは NYET です。

1 : 有効な LPM トークンは ACK です。

NYET/ACK は、LPM トランザクションが成功した場合にのみ返されます。

EXT トークンと LPM トークンの両方にエラーがない場合 (それ以外の場合は ERROR)

有効な bLinkState = 0001B (L1) を受信した場合 (それ以外の場合は STALL)

ビット 0 **LPMEN** : LPM サポートイネーブル

このビットは、USB デバイス内の LPM サポートを有効にするために、ソフトウェアによってセットされます。このビットが 0 の場合は、LPM トランザクションは処理されません。

## エンドポイント固有レジスタ

これらのレジスタの数は、USB ペリフェラルが処理する設計になっているエンドポイント数に応じて変化します。USB ペリフェラルは、最大 8 つの双方向エンドポイントをサポートします。各 USB デバイスは、制御エンドポイントをサポートしなければならず、そのアドレス (EA ビット) は 0 にセットされなければなりません。複数のエンドポイントが有効であり、同じエンドポイント番号値を持つ場合、USB ペリフェラルの動作は未定義です。各エンドポイントについて、USB\_EPnR レジスタはエンドポイント固有の情報を格納できます。

## USB エンドポイント n レジスタ (USB\_EPnR)、n=[0..7]

アドレスオフセット : 0x00 から 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_RX	DTOG_RX	STAT_RX[1:0]		SETUP	EP_TYPE[1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX[1:0]		EA[3:0]			
rc_w0	t	t	t	r	rw	rw	rw	rc_w0	t	t	t	rw	rw	rw	rw

USB リセットが USB バスから受信されたとき、または CTRL レジスタのビット FRES を通じて強制されたときにもリセットされますが、USB リセットイベントの直前の正しいパケット通知の喪失を回避するために、CTR\_RX および CTR\_TX ビットは変更されません。各エンドポイントに USB\_EPnR レジスタがあり、n はエンドポイント識別子です。

これらのレジスタでは読み出し-変更-書き込みのサイクルは避けるべきです。読み出し操作と書き込み操作の間に、いくつかのビットがハードウェアによってセットされる可能性があり、CPU に変更を検出する時間ができる前に、次の書き込みによって変更される可能性があるためです。この目的のために、この問題の影響を受けるすべてのビットが「不変」の値を持ち、変更が不要なときに使用される必要があります。これらのレジスタはロード命令付きで変更することが推奨され、この場合、ハードウェアによって変更できるすべてのビットには「不変」値が書き込まれます。

### ビット 15 CTR\_RX : 受信の場合の正しい転送

このビットは、このエンドポイントで OUT/SETUP トランザクションが正常に完了したときに、ハードウェアによってセットされます。ソフトウェアのみがこのビットをクリアできます。USB\_CNTR レジスタの CTRM ビットがセットされた場合、エンドポイント関連の割り込み条件（常に有効化されます）とともに汎用割り込み条件が生成されます。発生したトランザクションのタイプ（OUT または SETUP）は、下記の SETUP ビットから決めることができます。

NAK または STALL ハンドシェイクで終了したトランザクションでは、プロトコルエラーやデータトグル不一致の場合のように、実際にはデータは転送されていないため、このビットはセットされません。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んで効果はありません。

### ビット 14 DTOG\_RX : 受信転送の場合のデータトグル

エンドポイントがアイソクロナスでない場合、このビットは次の受信データパケットのデータトグルビット（0=DATA0、1=DATA1）の予期される値を含みます。データ PID 値が一致するデータパケット受信後、ACK ハンドシェイクが USB ホストに送信されたときに、ハードウェアはこのビットをトグルします。エンドポイントが制御エンドポイントとして定義された場合、ハードウェアはこのエンドポイントにアドレス指定された SETUP PID の受信時に、このビットをクリアします。

エンドポイントがダブルバッファリングを使用している場合、このビットはパケットバッファスワッピングをサポートするためにも使用されます（[セクション 32.5.3 : ダブルバッファエンドポイント](#)を参照）。

エンドポイントがアイソクロナスの場合、この種類のエンドポイントではデータトグルは使用されず、DATA0 パケットのみが送信されるため、このビットはパケットバッファスワッピングをサポートするためにのみ使用されます（[セクション 32.5.4 : アイソクロナス転送](#)を参照）。アイソクロナス転送ではハンドシェイクは使用されないため、ハードウェアはデータパケット受信の終了直後に、このビットをトグルします。

このビットは、値を初期化するため（エンドポイントが制御エンドポイントでないときは必須）、または特定のデータトグル/パケットバッファ使用のために、ソフトウェアによってトグルすることもできます。アプリケーションソフトウェアが 0 を書き込むと、DTOG\_RX の値は変更されませんが、1 を書き込むと、ビット値がトグルされます。このビットは読み出し/書き込みですが、1 を書き込むことによるのみ、トグルが可能です。



**ビット 13:12 STAT\_RX[1:0] : 受信転送の場合のステータスビット**

これらのビットは、[表 179 : 受信ステータスエンコード \(1080 ページ\)](#) にリストされているエンドポイントステータスに関する情報を含みます。これらのビットは、ソフトウェアによってトグルして、値を初期化することができます。アプリケーションソフトウェアが 0 を書き込むと、値は変更されませんが、1 を書き込むと、ビット値がトグルされます。ハードウェアは、このエンドポイントにアドレス指定された OUT または SETUP (制御のみ) トランザクションに対応する正しい転送が発生したとき (CTR\_RX=1)、STAT\_RX ビットを NAK にセットするので、ソフトウェアは新しいトランザクションに確認応答する前に受信データを処理する時間があります。

ダブルバッファバルクエンドポイントは、バッファ可用性条件に基づいてステータスを制御する特殊なトランザクションフロー制御を実装します ([セクション 32.5.3 : ダブルバッファエンドポイント](#) を参照)。

エンドポイントがアイソクロナスとして定義された場合、そのステータスは「VALID」または「DISABLED」のみなので、ハードウェアはトランザクションの成功後にエンドポイントのステータスを変更できません。ソフトウェアがアイソクロナスエンドポイントの STAT\_RX ビットを STALL または NAK にセットした場合、USB ペリフェラルの動作は未定義です。これらのビットは読み出し/書き込みですが、1 を書き込むことによるのみ、トグルが可能です。

**ビット 11 SETUP : セットアップトランザクション完了**

このビットは読み出し専用であり、最後に完了したトランザクションが SETUP のときにハードウェアによってセットされます。このビットは制御エンドポイントについてのみ値を変更します。成功した受信トランザクションの場合 (CTR\_RX イベント)、これを調べて、発生したトランザクションのタイプを判断する必要があります。割り込みサービスルーチンを次の着信トークンによる SETUP ビットの変更から保護するために、このビットは CTR\_RX ビットが 1 の間は停止され、CTR\_RX が 0 のときに状態が変更されます。このビットは読み出し専用です。

**ビット 10:9 EP\_TYPE[1:0] : エンドポイントタイプ**

これらのビットは、[表 180 : エンドポイントタイプエンコード \(1080 ページ\)](#) に記載されているように、このエンドポイントの動作を設定します。エンドポイント 0 は常に制御エンドポイントでなければならず、各 USB 機能にはアドレス 0 を持つ制御エンドポイントが少なくとも 1 つ必要ですが、必要な場合は、他の制御エンドポイントを使用することもできます。制御エンドポイントだけが SETUP トランザクションを処理し、他の種類のエンドポイントによって無視されます。SETUP トランザクションは NAK または STALL で応答することはできません。制御エンドポイントが NAK として定義された場合、SETUP トランザクションが受信されたときには、受信方向では USB ペリフェラルは応答せず、受信エラーをシミュレートします。制御エンドポイントが受信方向で STALL として定義された場合、SETUP パケットは受け入れられ、データを送信し、CTR 割り込みを発行します。エンドポイントが制御エンドポイントの場合でも、OUT トランザクションの受信は通常通りに処理されます。

バルクおよび割り込みエンドポイントの動作はほぼ同様ですが、EP\_KIND 設定ビットを使用して使用可能な特殊な機能のみが異なります。

アイソクロナスエンドポイントの使用については、[セクション 32.5.4 : アイソクロナス転送](#) で説明します。

**ビット 8 EP\_KIND : エンドポイントの種類**

このビットの意味は、EP\_TYPE ビットによって設定されたエンドポイントのタイプに依存します。[表 181](#) はさまざまな意味を要約します。

DBL\_BUF : このビットは、このバルクエンドポイントのダブルバッファリング機能を有効にするために、ソフトウェアによってセットされます。ダブルバッファバルクエンドポイントの使用については、[セクション 32.5.3 : ダブルバッファエンドポイント](#) で説明します。

STATUS\_OUT : このビットは、ステータスアウトトランザクションが予期されることを示すためにソフトウェアによってセットされます。この場合、1 個以上のデータバイトを含んでいるすべての OUT トランザクションは、ACK ではなく STALL で応答されます。このビットを使用して、制御転送時のプロトコルエラーに対するアプリケーションの堅牢性を向上させることができ、その使用は制御エンドポイントのみを対象としています。STATUS\_OUT がリセットされると、OUT トランザクションは必要に応じて任意の数のバイトを持つことができます。

## ビット 7 CTR\_TX : 送信の場合の正しい転送

このビットは、このエンドポイントで IN トランザクションが正常に完了したときに、ハードウェアによってセットされます。ソフトウェアのみがこのビットをクリアできます。USB\_CNTR レジスタの CTRM ビットがセットされた場合、エンドポイント関連の割り込み条件 (常に有効化されます) とともに汎用割り込み条件が生成されます。

NAK または STALL ハンドシェイクで終了したトランザクションでは、プロトコルエラーやデータトグル不一致の場合のように、実際にはデータは転送されていないため、このビットはセットされません。このビットは読み出し/書き込みですが、0 のみを書き込むことができます。

## ビット 6 DTOG\_TX : 送信転送の場合のデータトグル

エンドポイントがアイソクロナスでない場合、このビットは次の送信データパケットのデータトグルビット (0=DATA0、1=DATA1) の必要な値を含みます。データパケットの送信後、USB ホストから ACK ハンドシェイクが受信されると、ハードウェアはこのビットをトグルします。エンドポイントが制御エンドポイントとして定義された場合、このエンドポイントにアドレス指定された SETUP PID の受信時に、ハードウェアはこのビットを 1 にセットします。

エンドポイントがダブルバッファ機能を使用している場合、このビットはパケットバッファスワッピングをサポートするためにも使用されます ([セクション 32.5.3 : ダブルバッファエンドポイント](#) を参照)。

エンドポイントがアイソクロナスの場合、この種類のエンドポイントではデータトグルは使用されず、DATA0 パケットのみが送信されるため、このビットはパケットバッファスワッピングをサポートするためにのみ使用されます ([セクション 32.5.4 : アイソクロナス転送](#) を参照)。アイソクロナス転送ではハンドシェイクは使用されないため、ハードウェアはデータパケット送信の終了直後に、このビットをトグルします。

このビットは、値を初期化するため (エンドポイントが制御エンドポイントでないときは必須)、または特定のデータトグル/パケットバッファ使用のために、ソフトウェアによってトグルすることもできます。アプリケーションソフトウェアが 0 を書き込むと、DTOG\_TX の値は変更されませんが、1 を書き込むと、ビット値がトグルされます。このビットは読み出し/書き込みですが、1 を書き込むことによるのみ、トグルが可能です。

## ビット 5:4 STAT\_TX[1:0] : 送信転送の場合のステータスビット

これらのビットは、[表 182](#) にリストされているエンドポイントのステータスに関する情報を含みます。これらのビットは、ソフトウェアによってトグルして、値を初期化できます。アプリケーションソフトウェアが 0 を書き込むと、値は変更されませんが、1 を書き込むと、ビット値がトグルされます。ハードウェアは、このエンドポイントにアドレス指定された IN または SETUP (制御のみ) トランザクションに対応する正しい転送が発生したとき (CTR\_TX=1)、STAT\_TX ビットを NAK にセットします。次に、ソフトウェアが次の送信データセットを準備するのを待ちます。

ダブルバッファバルクエンドポイントは、バッファ可用性条件に基づいてステータスを制御する特殊なトランザクションフロー制御を実装します ([セクション 32.5.3 : ダブルバッファエンドポイント](#) を参照)。

エンドポイントがアイソクロナスとして定義された場合、そのステータスは「VALID」または「DISABLED」のみです。したがって、ハードウェアは、トランザクションの成功後にエンドポイントのステータスを変更することはできません。ソフトウェアがアイソクロナスエンドポイントの STAT\_TX ビットを STALL または NAK にセットした場合、USB ペリフェラルの動作は未定義です。これらのビットは読み出し/書き込みですが、1 を書き込むことによるのみ、トグルが可能です。

## ビット 3:0 EA[3:0] : エンドポイントアドレス

ソフトウェアは、このエンドポイント宛てのトランザクションを識別するために使用される 4 ビットのアドレスをこのフィールドに書き込む必要があります。値は、対応するエンドポイントを有効にする前に書き込まれる必要があります。

表 179. 受信ステータスエンコード

STAT_RX[1:0]	意味
00	<b>DISABLED</b> : このエンドポイントにアドレス指定されたすべての受信リクエストは無視されます。
01	<b>STALL</b> : エンドポイントは停止していて、すべての受信リクエストは STALL ハンドシェイクで終了します。
10	<b>NAK</b> : エンドポイントは NAK され、すべての受信リクエストは NAK ハンドシェイクで終了します。
11	<b>VALID</b> : このエンドポイントは受信に有効です。

表 180. エンドポイントタイプエンコード

EP_TYPE[1:0]	意味
00	バルク
01	制御
10	ISO
11	割り込み

表 181. エンドポイントの種類の意味

EP_TYPE[1:0]		EP_KIND の意味
00	バルク	DBL_BUF
01	制御	STATUS_OUT
10	ISO	未使用
11	割り込み	未使用

表 182. 送信ステータスエンコード

STAT_TX[1:0]	意味
00	<b>DISABLED</b> : このエンドポイントにアドレス指定されたすべての送信リクエストは無視されます。
01	<b>STALL</b> : エンドポイントは停止していて、すべての送信リクエストは STALL ハンドシェイクで終了します。
10	<b>NAK</b> : エンドポイントは NAK され、すべての送信リクエストは NAK ハンドシェイクで終了します。
11	<b>VALID</b> : このエンドポイントは送信に有効です。



## 32.6.2 バッファディスクリプタテーブル

バッファディスクリプタテーブルはパケットバッファメモリ内にありますが、そのエントリは、USB マクロセルとデバイス間でデータを交換するためのパケットバッファの位置とサイズを設定する追加レジスタとみなすことができます。

「1 x 16 ビット/ワード」のアクセス方式を持つデバイスでは、すべてのパケットメモリ位置は、USB\_BTABLE レジスタおよびバッファ記述テーブル位置の USB ペリフェラルで使用される実際のメモリ位置アドレスではなく、32 ビット整列アドレスを使用する APB によってアクセスされます。

次のページでは、「1 x 16 ビット/ワード」のアクセス方式を持つデバイスに関する 2 つのアドレス位置 (パケットメモリのアクセス時にアプリケーションソフトウェアによって使用されるアドレス位置と、USB ペリフェラルアクセスに関連するローカルアドレス位置) について示します。パケットメモリのアクセス時にアプリケーションソフトウェアによって使用される正しいメモリアドレス値を取得するには、実際のメモリ位置アドレスを 2 倍にする必要があります。

「2 x 16 ビット/ワード」のアクセス方式を持つデバイスでは、アプリケーションソフトウェアによって使用されるアドレス位置は、USB ペリフェラルアクセスに関連するローカルアドレス位置と同じになります。これらのデバイスでのパケットメモリへのアクセスは、バイト (8 ビット) またはハーフワード (16 ビット) アクセスによってのみ行う必要があります。ワード (32 ビット) アクセスは許されません。

最初のパケットメモリ位置は、0x4000 6000 にあります。USB\_EPnR レジスタに関連するバッファディスクリプタテーブルのエントリについて、以下に説明します。

パケットバッファとバッファディスクリプタテーブルの使用に関する詳細については、[パケットバッファの構造と用途 \(1059 ページ\)](#) を参照してください。

### 送信バッファアドレス n (USB\_ADDRn\_TX)

アドレスオフセット (「1 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*16

アドレスオフセット (「2 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*8

USB ローカルアドレス : [USB\_BTABLE] + n\*8

**注 :** **IN 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB\_ADDRn\_TX\_0 と呼ばれます。**

**OUT 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB\_ADDRn\_RX\_0 と呼ばれます。**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn_TX[15:1]															-
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	-

ビット 15:1 **ADDRn\_TX[15:1]** : 送信バッファアドレス

これらのビットは、USB\_EPnR レジスタに関連するエンドポイントによって、次の IN トークンで送信されるデータを含んでいるパケットバッファの開始アドレスを指します。

ビット 0 パケットメモリはハーフワード長であり、すべてのパケットバッファがハーフワードで揃えられている必要があるため、常に 0 が書き込まれる必要があります。



## 送信バイトカウント n (USB\_COUNTn\_TX)

アドレスオフセット (「1 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*16 + 4

アドレスオフセット (「2 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*8 + 2

USB ローカルアドレス : [USB\_BTABLE] + n\*8 + 2

**注 :** *IN* 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は **USB\_COUNTn\_TX\_0** と呼ばれます。

*OUT* 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は **USB\_COUNTn\_RX\_0** と呼ばれます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	COUNTn_TX[9:0]									
						rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:10 USB 仕様によりパケットサイズは 1023 バイトに制限されるので、これらのビットは使用されません。これらの値は、USB ペリフェラルによって考慮されません。

ビット 9:0 **COUNTn\_TX[9:0]** : 送信バイトカウント

これらのビットは、USB\_EPnR レジスタに関連するエンドポイントによって、次の IN トークンで送信されるバイト数を含みます。

## 受信バッファアドレス n (USB\_ADDRn\_RX)

アドレスオフセット (「1 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*16 + 8

アドレスオフセット (「2 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*8 + 4

USB ローカルアドレス : [USB\_BTABLE] + n\*8 + 4

**注 :** *OUT* 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は **USB\_ADDRn\_RX\_1** と呼ばれます。

*IN* 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は **USB\_ADDRn\_TX\_1** と呼ばれます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ADDRn_RX[15:1]															-	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	-

ビット 15:1 **ADDRn\_RX[15:1]** : 送信バッファアドレス

これらのビットは、USB\_EPnR レジスタに関連するエンドポイントによって、次の OUT/SETUP トークンで受信されるデータを含んでいるパケットバッファの開始アドレスを指します。

ビット 0 パケットメモリはハーフワード長であり、すべてのパケットバッファがハーフワードで揃えられている必要があるため、常に 0 が書き込まれる必要があります。

## 受信バイトカウント n (USB\_COUNTn\_RX)

アドレスオフセット (「1 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*16 + 12

アドレスオフセット (「2 x 16 ビット/ワード」のアクセス方式) : [USB\_BTABLE] + n\*8 + 6

USB ローカルアドレス : [USB\_BTABLE] + n\*8 + 6

**注 :** *OUT 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB\_COUNTn\_RX\_1 と呼ばれます。*

*IN 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB\_COUNTn\_TX\_1 と呼ばれます。*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLSIZE	NUM_BLOCK[4:0]						COUNTn_RX[9:0]									
r/w	r/w	r/w	r/w	r/w	r/w	r	r	r	r	r	r	r	r	r	r	r

このテーブル位置は、両方ともパケット受信時に必要とされる 2 つの値を格納するために使用されます。最上位ビットは割り当てられるバッファサイズの定義を含み、バッファオーバーフロー検出を可能にし、この位置の最下位部分には、受信終了時に USB ペリフェラルによって実際の受信バイト数を書き込まれます。使用可能なビット数が制限されているため、バッファサイズは割り当てられたメモリブロック数を使用して表され、細かい粒度/小さいバッファと粗い粒度/大きなバッファの間のトレードオフにより、ブロックサイズを選択できます。割り当てられたバッファのサイズは、エンドポイントディスクリプタの一部であり、通常、エニュメレーションプロセス時に maxPacketSize パラメータ値にしたって定義されます (「ユニバーサルシリアルバス仕様」を参照)。

### ビット 15 BL\_SIZE : ブロックサイズ

このビットは、割り当てられたバッファ領域を定義するために使用されるメモリブロックのサイズを選択します。

- BL\_SIZE=0 の場合、メモリブロックは 2 バイト長であり、これは、ハーフワード長のメモリで可能な最小ブロックです。このブロックサイズでは、割り当てられるバッファサイズは 2 ~ 62 バイトです。
- BL\_SIZE=1 の場合、メモリブロックは 32 バイト以上であり、USB 仕様によって定義された最大パケット長に達することができます。このブロックサイズでは、割り当てられるバッファサイズは、理論上、32 ~ 1024 バイトであり、USB 標準仕様で可能な最長のパケットサイズです。ただし、適用可能なサイズは、使用可能なバッファメモリによって制限されます。

### ビット 14:10 NUM\_BLOCK[4:0] : ブロック数

これらのビットは、このパケットバッファに割り当てられるメモリブロック数を定義します。実際の割り当てメモリの量は、表 183 に示されているように、BL\_SIZE の値に依存します。

### ビット 9:0 COUNTn\_RX[9:0] : 受信バイトカウント

これらのビットは、USB\_EPnR レジスタに関連するエンドポイントによって、最後の OUT/SETUP トランザクションで受信されたバイト数を含みます。

表 183. 割り当てられるバッファメモリの定義

値 NUM_BLOCK[4:0]	メモリ割り当て (BL_SIZE=0 の場合)	メモリ割り当て (BL_SIZE=1 の場合)
0 ('00000)	設定禁止	32 バイト
1 ('00001)	2 バイト	64 バイト
2 ('00010)	4 バイト	96 バイト
3 ('00011)	6 バイト	128 バイト
...	...	...
14 ('01110)	28 バイト	480 バイト
15 ('01111)	30 バイト	N/A
16 ('10000)	32 バイト	N/A
...	...	...
29 ('11101)	58 バイト	N/A
30 ('11110)	60 バイト	N/A
31 ('11111)	62 バイト	N/A

## 32.6.3 USB レジスタマップ

次の表に、USB レジスタマップとリセット値を示します。

表 184. USB レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	USB_EP0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	USB_EP1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x08	USB_EP2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x0C	USB_EP3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	USB_EP4R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x14	USB_EP5R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x18	USB_EP6R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	USB_EP7R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x20-0x3F	予約済み																																				
0x40	USB_CNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR	PMAOVRM	ERRM	WKUPM	SUSPM	RESSTM	SOFM	ESOFM	L1REQM	Res.	L1RESUMIE	RESUME	FSUSP	LPMODE	PDWEN	FRES			
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1		
0x44	USB_ISTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR	PMAOVR	ERR	WKUP	SUSP	RESET	SOF	ESOF	L1REQ	Res.	Res.	DIR	EP_ID[3:0]						
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x48	USB_FNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDP	RXDM	LCK	LSOF [1:0]	FN[10:0]														
	リセット値																		0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x			
0x4C	USB_DADDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD[6:0]											
	リセット値																									0	0	0	0	0	0	0	0	0	0		



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

表 184. USB レジスタマップとリセット値 (続き)

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	USB_BTABLE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BTABLE[15:3]										Res.	Res.	Res.			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	USB_LPMCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]			REMWAKE	Res.	Res.	Res.		
	リセット値																										0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 3.2.2 \(50 ページ\)](#) を参照してください。

## 33 デバッグサポート (DBG)

### 33.1 概要

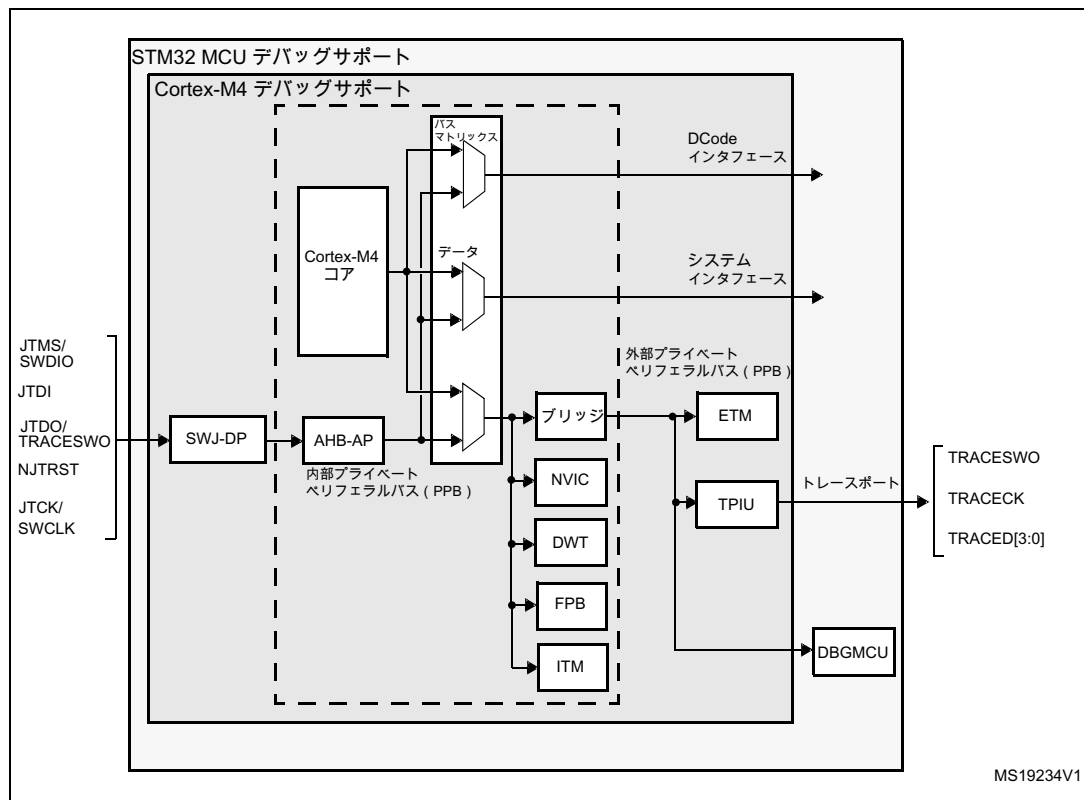
STM32F3xx は、Cortex-M4<sup>®</sup>F コアを中心にして構築されており、高度なデバッグ機能のためのハードウェア拡張機能を含んでいます。デバッグ拡張機能によって、コアは、特定の命令フェッチ時（ブレークポイント）またはデータアクセス時（ウォッチポイント）に停止することができます。停止したとき、コアの内部状態とシステムの外部状態を調査することができます。調査が終わったら、コアとシステムを復元して、プログラム実行を再開することができます。

デバッグ機能は、STM32F3xx MCU への接続時とデバッグ時にデバッガによって使用されます。

デバッグ用に、次の 2 つのインタフェースを使用できます。

- シリアルワイヤ
- JTAG デバッグポート

図 403. ブロック図 - STM32 MCU および Cortex-M4<sup>®</sup>F レベルのデバッグサポート Cortex-M4<sup>®</sup>F



注： Cortex<sup>®</sup>-M4 コアに内蔵されているデバッグ機能は、ARM<sup>®</sup> CoreSight Design Kit のサブセットです。

ARM® Cortex-M4®F コアは、統合的なオンチップデバッグサポート機能を提供します。この機能は以下の要素で構成されます。

- SWJ-DP : シリアルワイヤ/JTAG デバッグポート
- AHP-AP : AHB アクセスポート
- ITM : 計測トレースマクロセル (Instrumentation Trace Macrocell)
- FPB : フラッシュパッチブレークポイント (Flash Patch Breakpoint)
- DWT : データウォッチポイントトリガ
- TPIU : トレースポートインタフェースユニット (Trace Port Interface Unit : 対応するピンが配置される大型パッケージで使用)
- ETM : エンベデッドトレースマクロセル (Embedded Trace Macrocell : 対応するピンが配置される STM32F303xB/C および STM32F358xC デバイスの大型パッケージでのみ使用)

また、STM32F3xx 専用の以下のデバッグ機能も内蔵されています。

- 柔軟性の高いデバッグピンの割り当て
- MCU デバッグボックス (低電力モードのサポート、ペリフェラルクロックの制御など)

**注 :** ARM® Cortex-M4®F コアでサポートされるデバッグ機能の詳細については、FPU 搭載 Cortex®-M4-r0p1 Technical Reference Manual および CoreSight Design Kit-r0p1 TRM を参照してください(セクション 33.2 : ARM® リファレンス資料を参照)。

## 33.2 ARM® リファレンス資料

- Cortex-M4®F r0p1 Technical Reference Manual (TRM)  
<http://infocenter.arm.com> から入手可能です。
- ARM® Debug Interface V5
- ARM® CoreSight Design Kit revision r0p1 Technical Reference Manual

## 33.3 SWJ デバッグポート (シリアルワイヤと JTAG)

STM32F3xx のコアには、シリアルワイヤ/JTAG デバッグポート (SWJ-DP) が組み込まれています。これは、JTAG-DP (5 ピン) インタフェースと SW-DP (2 ピン) インタフェースを組み合わせた ARM® 標準の CoreSight デバッグポートです。

- JTAG デバッグポート (JTAG-DP) は、AHP-AP ポートに 5 ピンの標準 JTAG インタフェースを提供します。
- シリアルワイヤデバッグポート (SW-DP) は、AHP-AP ポートに 2 ピン (クロック + データ) のインタフェースを提供します。

SWJ-DP では、SW-DP の 2 個の JTAG ピンは、JTAG-DP の 5 個の JTAG ピンの一部と多重化されています。



図 404. SWJ デバッグポート

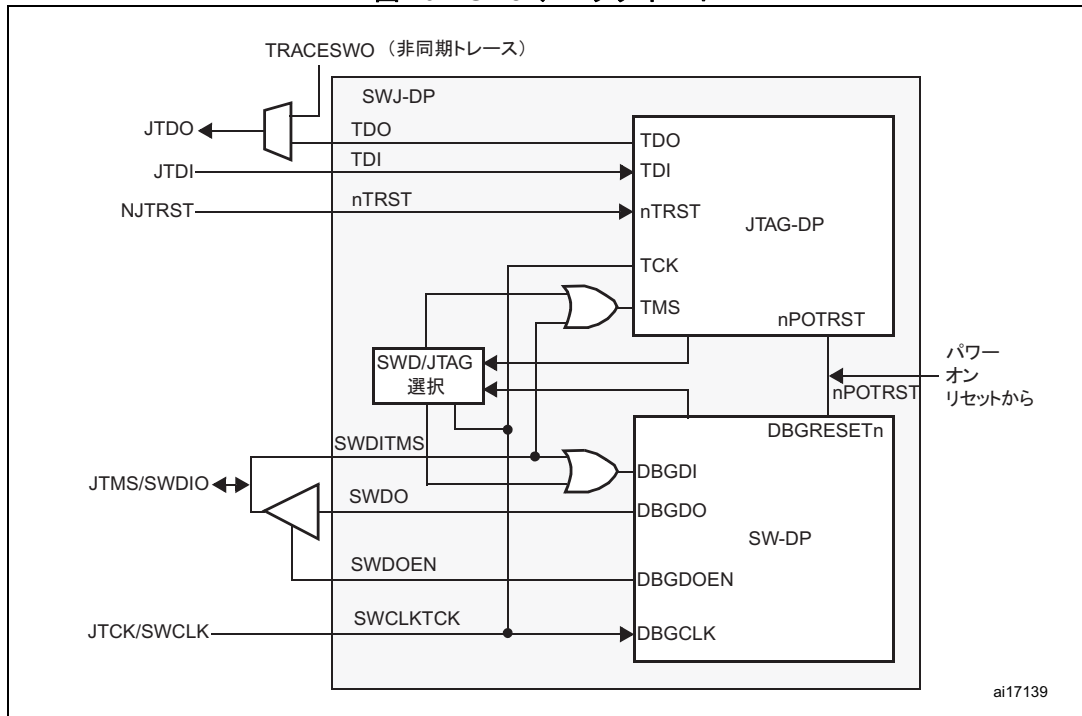


図 404 は、非同期 TRACE 出力 (TRACESWO) が TDO と多重化されていることを示します。つまり非同期トレースは、JTAG-DP ではなく、SW-DP にのみ使用できます。

### 33.3.1 JTAG-DP または SW-DP の選択メカニズム

デフォルトでは、JTAG デバッグポートが選択されています。

SW-DP に切り替えるには、デバッガホストは、TMS/TCK (それぞれ SWDIO/SWCLK に配置) に専用の JTAG シーケンスを提供して、JTAG-DP を無効にし SW-DP を有効にする必要があります。このように、SWCLK ピンと SWDIO ピンのみを使用して、SW-DP を有効にすることができます。

このシーケンスは、

1. TMS (SWDIO) =1 で 50 を超える TCK サイクルを送信します。
2. TMS (SWDIO) に 16 ビットシーケンス (0111100111100111) を MSB ファーストで送信します。
3. TMS (SWDIO) =1 で 50 を超える TCK サイクルを送信します。

### 33.4 ピン名とデバッグポートピン

STM32F3xx MCU は、使用できるピン数の異なるさまざまなパッケージに組み込まれています。このため、ピンを使用する一部の機能 (ETM) は、パッケージによって異なることがあります。

## 33.4.1 SWJ デバッグポートピン

5 個のピンが SWJ-DP 用の STM32F3xx からの出力として使用されます (汎用入出力のオルタネート機能)。これらのピンはすべてのパッケージで使用できます。

表 185. SWJ デバッグポートピン

SWJ-DP ピン名	JTAG デバッグポート		SW デバッグポート		ピン割り当て
	タイプ	説明	タイプ	デバッグ割り当て	
JTMS/SWDIO	I	JTAG テストモード選択	IO	シリアルワイヤデータ入出力	PA13
JTCK/SWCLK	I	JTAG テストクロック	I	シリアルワイヤクロック	PA14
JTDI	I	JTAG テストデータ入力	-	-	PA15
JTDO/TRACESWO	O	JTAG テストデータ出力	-	非同期トレースが有効な場合は TRACESWO	PB3
NJTRST	I	JTAG テスト nReset	-	-	PB4

## 33.4.2 柔軟性の高い SWJ-DP ピンの割り当て

RESET (SYSRESETn または PORESETn) 後、SWJ-DP 用に使用される 5 個のピンはすべて、デバッグホストによってすぐに使用可能な専用ピンとして割り当てられます (なお、デバッグホストによって明示的にプログラミングされた場合を除いて、トレース出力は割り当てられません)。

ただし、SWJ-DP ポートの一部またはすべてを無効にすることができるため、それにより汎用 I/O (GPIO) に使用する関連ピンを解放することもできます (下表にグレーで示す)。SWJ-DP ポートピンを無効にする方法の詳細については、[セクション 11.3.2 : I/O ピンオルタネート機能マルチプレクサと配置](#)を参照してください。

表 186. 柔軟性の高い SWJ-DP ピンの割り当て

使用可能なデバッグポート	SWJ IO ピンの割り当て				
	PA13 / JTMS / SWDIO	PA14 / JTCK / SWCLK	PA15 / JTDI	PB3 / JTDO	PB4 / NJTRST
全 SWJ (JTAG-DP + SW-DP)、リセット状態	X	X	X	X	X
全 SWJ (JTAG-DP + SW-DP)、NJTRST なし	X	X	X	X	
JTAG-DP 無効、SW-DP 有効	X	X			
JTAG-DP 無効、SW-DP 無効					解放

**注:** APB ブリッジの書き込みバッファがフルのとき、AFIO\_MAPR レジスタへの書き込みには APB 1 サイクル分が追加で必要になります。これは、コアの nTRST および TCK 入力信号でのクリーンレベルを保証するために、JTAGSW ピンの無効化が 2 サイクルで行われるからです。

- サイクル 1: コアへの JTAGSW 入力信号は 1 または 0 に (nTRST、TDI、および TMS では 1 に、TCK では 0 に) 接続されます。
- サイクル 2: GPIO コントローラが SWJTAG 入出力ピンの制御 (方向、プルアップ/ダウン、シュミットトリガの有効化などの制御) 信号を受け取ります。

## 33.4.3 JTAG ピンでの内部プルアップ/プルダウン

JTAG 入力ピンは、デバッグモード機能を制御するためにフリップフロップに直結されます。したがって、JTAG 入力ピンをフロート状態にしないことが必要です。これらのフリップフロップの一部のクロックに直結される SWCLK/TCK ピンについては、特に注意が必要です。

入出力レベルを正しく制御するため、デバイスには内部プルアップ/プルダウンが JTAG 入力ピンに内蔵されています。

- NJTRST : 内部プルアップ
- JTDI : 内部プルアップ
- JTMS/SWDIO : 内部プルアップ
- TCK/SWCLK : 内部プルダウン

JTAG 入出力がユーザソフトウェアによって解放されると、GPIO コントローラが再び制御権を獲得します。リセット状態では、GPIO 制御レジスタは入出力をこれと同等の状態に設定します。

- NJTRST : 入力プルアップ
- JTDI : 入力プルアップ
- JTMS/SWDIO : 入力プルアップ
- JTCK/SWCLK : 入力プルダウン
- JTDO : 入力フローティング

ソフトウェアはこれらの入出力を標準の GPIO 信号として使用することができます。

**注 :** *JTAG IEEE 規格では、TDI、TMS、および nTRST にプルアップを追加することを推奨していますが、TCK に関しては特別な推奨はありません。ただし、JTCK に関しては、デバイスは内蔵プルダウンを必要とします。*

*プルアップとプルダウンを内蔵しているため、外部抵抗を追加する必要はありません。*

## 33.4.4 シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法

シリアルワイヤ DP を使って一部の GPIO を解放するには、ユーザソフトウェアが GPIO\_MODER レジスタで GPIO (PA15、PB3、および PB4) コンフィギュレーションモードを変更する必要があります。これによって PA15、PB3、および PB4 が解放され、GPIO として使用できるようになります。

デバッグ時には、ホストは次の動作を行います。

- システムリセット中に、すべての SWJ ピンが割り当てられます (JTAG-DP + SW-DP)。
- システムリセット中に、デバッガホストは、JTAG-DP から SW-DP に切り替える JTAG シーケンスを送信します。
- システムリセット中に、さらに、デバッガはベクタリセットにブレークポイントを設定します。
- システムリセットは解除され、コアは停止します。
- これ以降のすべてのデバッグ通信は、SW-DP を使用して行われます。残りの JTAG ピンは、ユーザソフトウェアによって GPIO として再割当てできます。

**注：** ユーザソフトウェアの設計に関しては、次の点に注意してください。

デバッグピンを解放するには、リセット後にユーザソフトウェアがピンを解放するまでの一定の期間、デバッグピンは、まず、入力プルアップ (nTRST、TMS、TDI)、プルダウン (TCK)、または出カトライステート (TDO) に設定されることに注意してください。

デバッグピン (JTAG、SW、または TRACE) が配置されると、IOPORT コントローラの対応する入出力ピンの設定を変更しても効果はありません。

## 33.5 STM32F3xxJTAG TAP 接続

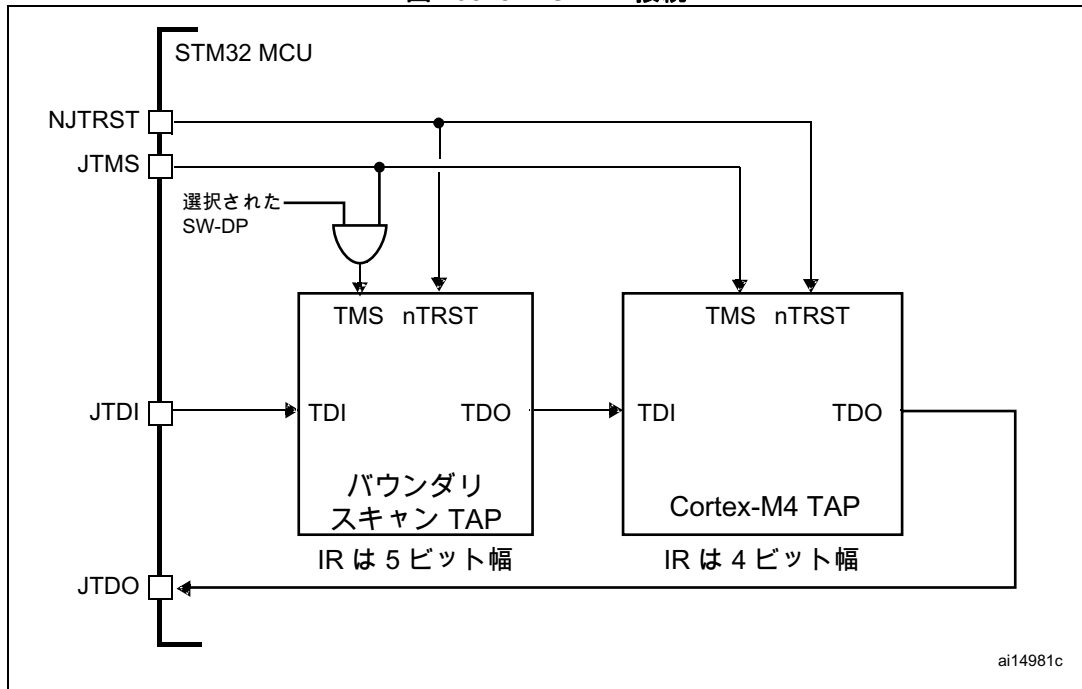
STM32F3xx MCU は、バウンダリスキャン TAP (IR は 5 ビット幅) と Cortex-M4<sup>®</sup>F TAP (IR は 4 ビット幅) という、シリアル接続された 2 個の JTAG TAP を内蔵しています。

デバッグ目的で Cortex-M4<sup>®</sup>F の TAP にアクセスするには、

1. まず、バウンダリスキャン TAP の BYPASS 命令をシフトする必要があります。
2. 次に、IR シフトごとに、スキャンチェーンには 9 (=5+4) ビットが格納され、未使用の TAP 命令を BYPASS 命令によってシフトインする必要があります。
3. データシフトごとに、BYPASS モードにある未使用の TAP は、データスキャンチェーンに 1 個のデータビットを追加します。

**注：** **重要：** 専用の ARM<sup>®</sup> JTAG シーケンスを使用してシリアルワイヤが選択されると、バウンダリスキャン TAP は自動的に無効になります (JTMS は強制的にハイレベルになります)。

図 405. JTAG TAP 接続



## 33.6 ID コードとロック機構

STM32F3xx MCU には、内部にいくつかの ID コードがあります。ツール設計者は、外部 PPB メモリマップのアドレス 0xE0042000 に配置されている MCU デバイス ID コードを使用して、デバッグをロックすることを強く推奨します。

## 33.6.1 MCU デバイス ID コード

STM32F3xx MCU には MCU ID コードが内蔵されています。この ID は、ST 社製 MCU の部品番号とダイのリビジョンを識別します。これは DBG\_MCU 部品の一部であり、外部 PPB バスに配置されず ([セクション 33.16 \(1106 ページ\)](#) を参照)。このコードにアクセスするには、JTAG デバッグポート (4~5 本のピン)、SW デバッグポート (2 本のピン)、またはユーザソフトウェアを使用します。アクセスは MCU がシステムリセット中でも可能です。

### DBGMCU\_IDCODE

アドレス : 0xE004 2000

32 ビットアクセスのみサポートされます。読み出し専用。

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

ビット 31:16 **REV\_ID(15:0)** リビジョン識別子

このフィールドは、デバイスのリビジョンを示します。

0x1001 : リビジョン Z

0x1003 : リビジョン Y

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DEV\_ID(11:0)** : デバイス識別子

このフィールドは、デバイスとそのリビジョンを示します。

デバイス ID は以下のとおりです。

STM32F303xB/C および STM32F358 デバイスの場合は 0x422。

STM32F303x6/8 および STM32F328 デバイスの場合は 0x438。

STM32F303xD/E および STM32F398xE デバイスの場合は 0x446。

## 33.6.2 バウンダリスキャン TAP

### JTAG ID コード

STM32F3xx BSC (バウンダリスキャン) の TAP は、0x06432041 に等しい JTAG ID コードを内蔵しています。

## 33.6.3 Cortex-M4<sup>®</sup>F TAP

ARM<sup>®</sup> Cortex-M4<sup>®</sup>F の TAP は JTAG ID コードを内蔵しています。この ID コードは ARM<sup>®</sup> のデフォルトであり、変更されていません。このコードは、JTAG デバッグポートからのみアクセスできます。このコードは **0x4BA00477** です (Cortex-M4<sup>®</sup>F r0p1 に対応、[セクション 33.2 : ARM<sup>®</sup> リファレンス資料](#)を参照)。

デバッガ/プログラマツールでは、DEV\_ID(11:0) のみを識別に使用してください。

## 33.6.4 Cortex-M4<sup>®</sup>F JEDEC-106 ID コード

ARM<sup>®</sup> Cortex-M4<sup>®</sup>F は JEDEC-106 ID コードを内蔵しています。これは内部 PPB バスのアドレス 0xE00FF000\_0xE00FFFFF に配置された 4KB ROM テーブルに置かれています。

このコードは、JTAG デバッグポート (4~5 本のピン)、SW デバッグポート (2 本のピン)、またはユーザソフトウェアによってアクセスできます。

## 33.7 JTAG デバッグポート

標準的な JTAG ステートマシンは、4 ビット命令レジスタ (IR) と 5 個のデータレジスタを搭載しています (全詳細については、Cortex-M4<sup>®</sup>Fr0p1 *Technical Reference Manual (TRM)* を参照し、参考として[セクション 33.2 : ARM<sup>®</sup> リファレンス資料](#)を参照)。

表 187. JTAG デバッグポートのデータレジスタ

IR(3:0)	データレジスタ	詳細
1111	BYPASS [1 ビット]	
1110	IDCODE [32 ビット]	ID コード 0x3BA00477 (ARM <sup>®</sup> Cortex-M4 <sup>®</sup> F r0p1 の ID コード)
1010	DPACC [35 ビット]	<p>デバッグポートアクセスレジスタ</p> <p>デバッグポートを初期化し、デバッグポートレジスタへのアクセスを可能にします。</p> <ul style="list-style-type: none"> <li>- 入力データ転送時 :                      ビット 34:3 = DATA[31:0] = 書き込みリクエスト用に転送する 32 ビットデータ                      ビット 2:1 = A[3:2] = デバッグポートレジスタの 2 ビットアドレス                      ビット 0 = RnW = 読み出しリクエスト (1) または書き込みリクエスト (0)</li> <li>- 出力データ転送時 :                      ビット 34:3 = DATA[31:0] = 読み出しリクエストに続いて読み出される 32 ビットデータ                      ビット 2:0 = ACK[2:0] = 3 ビット確認応答 :                      010 = OK/FAULT                      001 = WAIT                      その他 = 予約済み</li> </ul> <p>A[3:2] ビットの説明については、<a href="#">表 188</a> を参照してください。</p>

表 187. JTAG デバッグポートのデータレジスタ (続き)

IR(3:0)	データレジスタ	詳細
1011	APACC [35 ビット]	<p>アクセスポートアクセスレジスタ アクセスポートを初期化し、アクセスポートレジスタへのアクセスを可能にします。</p> <ul style="list-style-type: none"> <li>- 入力データ転送時 : ビット 34:3 = DATA[31:0] = 書き込みリクエスト用にシフトインする 32 ビットデータ ビット 2:1 = A[3:2] = 2 ビットアドレス (サブアドレス AP レジスタ) ビット 0 = RnW = 読み出しリクエスト (1) または書き込みリクエスト (0)</li> <li>- 出力データ転送時 : ビット 34:3 = DATA[31:0] = 読み出しリクエストに続いて読み出される 32 ビットデータ ビット 2:0 = ACK[2:0] = 3 ビット確認応答 : 010 = OK/FAULT 001 = WAIT その他 = 予約済み</li> </ul> <p>次の項目の組み合わせとして、多くの AP レジスタ (AHB-AP を参照) をアドレス指定します。</p> <ul style="list-style-type: none"> <li>- シフトされた値 A[3:2]</li> <li>- DP SELECT レジスタの現在値</li> </ul>
1000	ABORT [35 ビット]	<p>アポートレジスタ</p> <ul style="list-style-type: none"> <li>- ビット 31:1 = 予約済み</li> <li>- ビット 0 = DAPABORT : DAP アポートを生成するには 1 を書き込みます</li> </ul>

表 188. シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ

アドレス	A[3:2] の値	説明
0x0	00	予約済みであり、リセット値に保持する必要があります。
0x4	01	<p>DP CTRL/STAT レジスタ : 次の目的で使用されます。</p> <ul style="list-style-type: none"> <li>- システム/デバッグパワーアップのリクエスト</li> <li>- AP アクセス用の転送動作の設定</li> <li>- プッシュ比較/プッシュ検証動作の制御</li> <li>- ステータスフラグ (オーバーラン、パワーアップの確認応答) の読み出し</li> </ul>
0x8	10	<p>DP SELECT レジスタ : 現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウの選択に使用されます。</p> <ul style="list-style-type: none"> <li>- ビット 31:24 (APSEL) : 現在の AP を選択します。</li> <li>- ビット 23:8 : 予約済み</li> <li>- ビット 7:4 (APBANKSEL) : 現在の AP でアクティブな 4 ワードレジスタウィンドウを選択します。</li> <li>- ビット 3:0 : 予約済み</li> </ul>
0xC	11	<p>DP RDBUFF レジスタ : 一連の動作のあとで (新しい JTAG-DP 動作をリクエストせずに)、デバッガが最終結果を得られるようにします。</p>



## 33.8 SW デバッグポート

### 33.8.1 SW プロトコルの概要

この同期式シリアルプロトコルでは、次の 2 個のピンを使用します。

- SWCLK : ホストからターゲットへのクロック
- SWDIO : 双方向

このプロトコルでは、2 バンクのレジスタ (DPACC レジスタと APACC レジスタ) の読み出し/書き込みが可能です。

ビットは、ワイヤ上を LSB ファーストで転送されます。

SWDIO を双方向管理するには、ボード上でラインがプルアップされている必要があります (ARM® の推奨値は 100 kΩ)。

プロトコルで SWDIO の方向が変化するたびに、ラインがホストからもターゲットからも駆動されないターンアラウンド時間が挿入されます。このターンアラウンド時間のデフォルトは 1 ビット時間ですが、SWCLK 周波数の設定によって調整できます。

### 33.8.2 SW プロトコルシーケンス

各シーケンスは 3 つのフェーズで構成されます。

1. ホストによって送信されるパケットリクエスト (8 ビット)
2. ターゲットによって送信される確認応答 (3 ビット)
3. ホストまたはターゲットによって送信されるデータ転送フェーズ (33 ビット)

表 189. パケットリクエスト (8 ビット)

ビット	名前	説明
0	Start	"1"である必要があります。
1	APnDP	0 : DP アクセス 1 : AP アクセス
2	RnW	0 : 書き込みリクエスト 1 : 読み出しリクエスト
4:3	A[3:2]	DP/AP レジスタのアドレスフィールド (表 188を参照)
5	Parity	先行するビットの 1 ビットパリティ
6	STOP	0
7	Park	ホストによって駆動されません。プルアップの効果で、ターゲットには「1」と読み出される必要があります。

DPACC レジスタと APACC レジスタの詳細については、Cortex-M4®F r0p1 TRM を参照してください。

パケットリクエストの後には、必ずホストもターゲットもラインを駆動しないターンアラウンド時間 (デフォルトでは 1 ビット) が続きます。

表 190. ACK 応答 (3 ビット)

ビット	名前	説明
0~2	ACK	001 : FAULT 010 : WAIT 100 : OK

読み出しトランザクションの場合や、受信した ACK 応答が"WAIT" または"FAULT" の場合にのみ、ACK 応答の後にターンアラウンド時間が続く必要があります。

表 191. データ転送 (33 ビット)

ビット	名前	説明
0~31	WDATA または RDATA	書き込み/読み出しデータ
32	Parity	32 データビットの 1 ビットパリティ

読み出しトランザクションの場合にのみ、データ転送の後にターンアラウンド時間が続く必要があります。

### 33.8.3 SW-DP ステートマシン (リセット、アイドル状態、ID コード)

SW-DP のステートマシンには、SW-DP を識別する内部 ID コードがあります。これは JEP-106 規格に準じています。この ID コードは、ARM® のデフォルトコードであり、0x1BA01477 (Cortex-M4®F r0p1 に対応) がセットされています。

**注:** SW-DP ステートマシンは、ターゲットがこの ID コードを読み出すまで非アクティブであることに注意してください。

- パワーオンリセット後、または DP が JTAG から SWD に切り替えられた後、またはラインが 50 サイクルを超えてハイレベルにあった後では、SW-DP ステートマシンは RESET 状態になります。
- リセット状態のあと、ラインが 2 サイクル以上の間ローレベルであれば、SW-DP ステートマシンはアイドル状態になります。
- リセット状態のあとは、まずアイドル状態に入り、次に DP-SW ID CODE レジスタの読み出しアクセスを行う **必要があります**。そうしないと、ターゲットは、他のトランザクションに対して ACK 応答の"FAULT"を発行します。

SW-DP ステートマシンの詳細については、Cortex-M4®F r0p1 TRM および CoreSight Design Kit r0p1 TRM を参照してください。

### 33.8.4 DP と AP の読み出し/書き込みアクセス

- DP への読み出しアクセスはポストされません。つまり、ターゲットは、ACK 応答が"OK"の場合はただちに応答し、ACK 応答が"WAIT"の場合は遅れて応答します。
- AP への読み出しアクセスはポストされます。つまり、アクセスの結果は次の転送時に返されず、次のアクセスが AP アクセスでない場合、結果を得るには DP-RDBUFF レジスタを読み出す必要があります。  
AP 読み出しアクセスが成功したかどうかを判断するため、DP-CTRL/STAT レジスタの READOK フラグは、AP 読み出しアクセスまたは RDBUFF 読み出しリクエストのたびに更新されます。
- SW-DP は、DP と AP の両方の書き込みに使用できる書き込みバッファを実装しているため、たとえ他のトランザクションが未処理であっても、書き込み動作を受け付けることができます。書

き込みバッファがフルのとき、ターゲットの ACK 応答は"WAIT"です。例外として、IDCODE 読み出し、CTRL/STAT 読み出し、または ABORT 書き込みは、書き込みバッファがフルであっても受け付けられます。

- 非同期クロックドメイン SWCLK と HCLK によって、書き込みを内部的に有効にするには、書き込みトランザクション後（パリティビット後）に SWCLK の 2 サイクルが余分に必要となります。これらのサイクルは、ラインをローレベルに駆動している間（アイドル状態）に適用してください。  
これは、パワーアップリクエストのために CTRL/STAT の書き込みを行う際に特に重要です。パワーアップを必要とする次のトランザクションがただちに発生すると、そのトランザクションは失敗します。

### 33.8.5 SW-DP レジスタ

これらのレジスタへのアクセスは、APnDP = 0 のときに開始されます。

表 192. SW-DP レジスタ

A[3:2]	読み出し/ 書き込み	SELECT レジスタの CTRLSEL ビット	レジスタ	注
00	読み出し		IDCODE	製造者コードは ST 社のコード <b>0x2BA01477</b> ではありません (SW-DP を指定)。
00	書き込み		ABORT	
01	読み出し/ 書き込み	0	DP CTRL/STAT	用途は以下のとおりです。 - システム/デバッグパワーアップのリクエスト - AP アクセス用の転送動作の設定 - プッシュ比較/プッシュ検証動作の制御 - ステータスフラグ（オーバーラン、パワーアップの確認応答）の読み出し
01	読み出し/ 書き込み	1	WIRE CONTROL	物理的なシリアルポートプロトコルの設定（ターンアラウンド時間など）を行います。
10	読み出し		READ RESEND	元の AP 転送を反復しなくても、破壊されたデバッグ転送からの読み出しデータの復旧を可能にします。
10	書き込み		SELECT	現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウを選択します。
11	読み出し/ 書き込み		読み出しバッファ	AP アクセスはポストされるため、この読み出しバッファは効果的です (AP 読み出しリクエストの結果は、次の AP トランザクションで取得できる)。 この読み出しバッファは、新しいトランザクションを開始することなく、前回の読み出しの結果として AP から出力されるデータをキャプチャします。

## 33.8.6 SW-AP レジスタ

これらのレジスタへのアクセスは、APnDP = 1 のときに開始されます。

次の項目の組み合わせとして、多くの AP レジスタ (AHB-AP を参照) をアドレス指定します。

- シフトされた値 A[3:2]
- DP SELECT レジスタの現在値

## 33.9 AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効

機能：

- システムアクセスはプロセッサステータスから独立しています。
- SW-DP または JTAG-DP が AHB-AP にアクセスします。
- AHB-AP はバスマトリックスの AHB マスタです。したがって、AHB-AP は ICode バスを除くすべてのデータバス (Dcode バス、システムバス、内部/外部の PPB バス) にアクセスできます。
- ビットバンドトランザクションがサポートされます。
- AHB-AP トランザクションは FPB を迂回します。

32 ビット AHB-AP レジスタは 6 ビット幅 (最大 64 ワードまたは 256 バイト) であり、以下の構成となっています。

- d) ビット [7:4] = DP\_SELECT レジスタのビット [7:4] APBANKSEL
- e) ビット [3:2] = SW-DP 用の 35 ビットパケットリクエストの 2 ビットアドレス A(3:2)

Cortex-M4<sup>®</sup>F の AHB-AP は、9 個の 32 ビットレジスタを内蔵しています。

表 193. Cortex-M4<sup>®</sup>F AHB-AP レジスタ

アドレス オフセット	レジスタ名	注
0x00	AHB-AP 制御およびステータス ワード	AHB インタフェースを通じて転送を設定および制御します (サイズ、hprot、現在の転送のステータス、アドレスインクリ メントタイプ)。
0x04	AHB-AP 転送アドレス	
0x0C	AHB-AP データ読み出し/書き込 み	
0x10	AHB-AP バンクデータ 0	転送アドレスレジスタを書き換えずに、4 個のアラインドデー タワードを直接配置します。
0x14	AHB-AP バンクデータ 1	
0x18	AHB-AP バンクデータ 2	
0x1C	AHB-AP バンクデータ 3	
0xF8	AHB-AP デバッグ ROM アドレス	デバッグインタフェースのベースアドレス
0xFC	AHB-AP ID レジスタ	

詳細については、Cortex-M4<sup>®</sup>F r0p1 TRM を参照してください。

## 33.10 コアデバッグ

コアデバッグはコアデバッグレジスタを通じてアクセスされます。これらのレジスタへのデバッグアクセスには、Advanced High-performance Bus (AHB-AP) ポートを使用します。プロセッサは、内部のプライベートペリフェラルバス (PPB) を介してこれらのレジスタに直接アクセスできます。

コアデバッグは 4 個のレジスタから構成されています。

表 194. コアデバッグレジスタ

レジスタ	説明
DHCSR	32 ビットのデバッグ停止制御/ステータスレジスタ : レジスタプロセッサの状態についてのステータス情報を提供し、コアデバッグを有効にし、プロセッサの停止とステップ実行を行います。
DCRSR	17 ビットのデバッグコアレジスタセクタレジスタ : データの転送先または転送元となるプロセッサレジスタを選択します。
DCRDR	32 ビットのデバッグコアレジスタデータレジスタ : DCRSR (セクタ) レジスタによって選択されたプロセッサとの間でレジスタの読み出しおよび書き込みに使用するデータを保持します。
DEMCR	32 ビットのデバッグ例外/モニタ制御レジスタ : ベクタキャッチとデバッグモニタの制御を行います。このレジスタには、TRACE を使用できるようにする <i>TRCENA</i> というビットがあります。

**注 :** **重要 :** これらのレジスタは、システムリセットによってはリセットされません。パワーオンリセットによってのみリセットされます。

詳細については、Cortex-M4<sup>®</sup>F r0p1 TRM を参照してください。

リセット時に停止させるには、以下の手順が必要です。

- デバッグ例外/モニタ制御レジスタのビット 0 (VC\_CORRESET) を有効にします。
- デバッグ停止制御/ステータスレジスタのビット 0 (C\_DEBUGEN) を有効にします。

## 33.11 システムリセット中のデバッグホスト接続機能

STM32F3xx MCU のリセットシステムは、次のリセットソースから構成されます。

- POR (パワーオンリセット) : パワーアップのたびに RESET をアサートします。
- 内部ウォッチドッグリセット
- ソフトウェアリセット
- 外部リセット

Cortex-M4<sup>®</sup>F では、デバッグ部のリセット (一般に PORRESETn) とそれ以外のリセット (SYSRESETn) が区別されます。

こうすれば、リセットベクタをフェッチするときにコアを停止させるようコアデバッグレジスタをプログラミングすることで、デバッグはシステムリセット中に接続が可能になります。これによってホストはシステムリセットを解除でき、コアは命令を実行しなくても、ただちに停止します。さらに、システムリセット中にデバッグ機能をプログラミングできます。

**注 :** **システムリセット中にデバッグホストを接続する (リセットベクタにブレークポイントを設定する) ことを強く推奨します。**



## 33.12 FPB (フラッシュパッチブレイクポイント)

FPB ユニットは、

- ハードウェアブレイクポイントを実装します。
- コード空間からシステム空間に、コードとデータをパッチします。この機能によって、コードメモリ空間にあるソフトウェアバグの訂正が可能になることがあります。

ソフトウェアパッチとハードウェアブレイクポイントは同時に使用できません。

FPB は以下から構成されます。

- コード空間からのリテラルロードと照合し、システム空間内の対応する領域に再配置する 2 個のリテラルコンパレータ。
- コード空間からの命令フェッチと照合する 6 個の命令コンパレータ。命令コンパレータを使用すると、システム空間内の対応する領域に再配置したり、コアへのブレイクポイント命令を生成したりできます。

## 33.13 DWT (データウォッチポイントトリガ)

DWT ユニットの 4 個のコンパレータから構成されており、次の用途に設定できます。

- ハードウェアウォッチポイント
- ETM へのトリガ
- PC サンプラ
- データアドレスサンプラ

DWT は、プロファイル情報を提供することもできます。このため、次の値を得るためのカウンタにアクセスできます。

- クロックサイクル数
- フォールドされた命令数
- ロードストアユニット (LSU) の動作数
- スリープサイクル数
- CPI (命令あたりのクロック数)
- 割込みオーバーヘッドの回数

## 33.14 ITM (計測トレースマクロセル)

### 33.14.1 概要

ITM は、アプリケーション駆動のトレースソースであり、*printf* 方式のデバッグをサポートすることでオペレーティングシステム (OS) やアプリケーションのイベントをトレースし、システムの診断情報を発行します。ITM がパケットとして発行するトレース情報は、次のように生成できます。

- **ソフトウェアトレース**: ソフトウェアは、ITM スティムラスレジスタに直接書き込みを行って、パケットを発行できます。
- **ハードウェアトレース**: DWT がパケットを生成し、ITM がそれを発行します。
- **タイムスタンプ**: タイムスタンプはパケットを基準にして発行されます。ITM は、タイムスタンプを生成するための 21 ビットカウンタを内蔵しています。カウンタは、Cortex-M4<sup>®</sup>F クロックまたはシリアルワイヤビュア (SWV) 出力のビットクロックレートによって駆動されます。

ITM によって発行されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから (TPIU を参照)、完全なパケットシーケンスをデバッグホストに出力します。

ITM をプログラミングまたは使用するには、その前に、デバッグ例外/モニタ制御レジスタの TRCEN ビットを有効にする必要があります。

### 33.14.2 タイムスタンプパケット、同期およびオーバーフローパケット

タイムスタンプパケットは、タイムスタンプ情報や一般的な制御および同期情報をコード化します。そのために使用される 21 ビットのタイムスタンプカウンタ (プリスケアラ付きの場合もあります) は、タイムスタンプパケットが発行されるたびにリセットされます。このカウンタは、CPU クロックまたは SWV クロックによって駆動できます。

同期パケットは 6 バイトの値 0x80\_00\_00\_00\_00\_00 から構成され、TPIU には 00 00 00 00 00 80 として LSB ファーストで発行されます。

同期パケットは、タイムスタンプ制御のためのパケットで、DWT トリガのたびに発行されます。



このため、DWT 制御レジスタの CYCCNTENA (ビット 0) をセットして、ITM をトリガするように DWT を設定する必要があります。さらに、ITM トレース制御レジスタのビット 2 (SYNCENA) もセットする必要があります。

**注：** **SYNENA ビットがセットされていない場合、DWT は TPIU への同期トリガを生成し、TPIU 同期パケットのみを送信し、ITM 同期パケットは送信しません。**

オーバーフローパケットは、データの書き込み時に FIFO がフルであったことを示す特殊なタイムスタンプパケットで構成されます。

表 195. 主な ITM レジスタ

アドレス	レジスタ	詳細
@E0000FB0	ITM ロックアクセス	他の ITM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
@E0000E80	ITM トレース制御	ビット 31-24 = 常に 0
		ビット 23 = ビジー
		ビット 22-16 = トレースデータのソースを識別する 7 ビットの ATB ID
		ビット 15-10 = 常に 0
		ビット 9:8 = TSPrescale = タイムスタンププリスケール
		ビット 7-5 = 予約済み
		ビット 4 = SWOENA = SWV 動作 (タイムスタンプカウンタを SWV クロックによって駆動) を有効にします。
		ビット 3 = DWTENA : DWT スティムラスを有効にします。
		ビット 2 = SYNCENA : DWT が同期トリガを生成して TPIU が同期パケットを発行できるようにするには、このビットを 1 にする必要があります。
		ビット 1 = TSENA (タイムスタンプ有効)
ビット 0 = ITMENA : ITM のグローバル有効ビット		
@E0000E40	ITM トレース特権	ビット 3 : ポート 31:24 のトレースを有効にするマスク
		ビット 2 : ポート 23:16 のトレースを有効にするマスク
		ビット 1 : ポート 15:8 のトレースを有効にするマスク
		ビット 0 : ポート 7:0 のトレースを有効にするマスク
@E0000E00	ITM トレース有効	各ビットは、トレースを生成するために対応するスティムラスポートを有効にします。
@E0000000- E000007C	スティムラスポート レジスタ 0-31	選択されたスティムラスポート (32 個まで) に 32 ビットデータを書き込んでトレース出力します。



## 設定例

TPIU に単純な値を出力するには、

- TPIU を設定し、DBGMCU\_CR を設定することで TRACE I/O を割り当てます ([セクション 33.17.2: TRACE ピンの割当て](#)および[セクション 33.16.3: デバッグMCU 設定レジスタ](#)を参照)。
- ITM ロックアクセスレジスタに 0xC5ACCE55 を書き込んで、ITM レジスタへの書き込みアクセスをロック解除します。
- ITM トレース制御レジスタに 0x00010005 を書き込んで、同期を有効にした状態で ITM を有効にし、ATB ID を 0x00 以外の値にします。
- ITM トレース有効レジスタに 0x1 を書き込んで、スティムラサポート 0 を有効にします。
- ITM トレース特権レジスタに 0x1 を書き込んで、スティムラサポート 7:0 をマスク解除します。
- スティムラサポートレジスタ 0 に出力値を書き込みます。これはソフトウェアで行うことができます (printf 機能を使用)。

## 33.15 ETM (組み込みトレースマクロセル)

ETM は、STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスでのみ使用できます。

### 33.15.1 概要

ETM はプログラム実行の再構築を有効にします。データはデータウォッチポイントおよびトレース (DWT) コンポーネントまたは計測トレースマクロセル (ITM) を使用してトレースされます。一方、命令は組み込みトレースマクロセル (ETM) を使用してトレースされます。

ETMは、情報をパケットとして送信し、埋め込みリソースによってトリガされます。これらのリソースは個別にプログラミングする必要があり、トリガソースの選択にはトリガイベントレジスタ (0xE0041008) を使用します。イベントは、単一イベント (アドレスコンパレータからのアドレスマッチ) または 2 つのイベント間の論理式とすることができます。トリガソースは、DWT モジュールの 4 番目のコンパレータの 1 つで、次のイベントを監視できます。

- クロックサイクルマッチング
- データアドレスマッチング

トリガリソースの詳細については、[セクション 33.13: DWT \(データウォッチポイントトリガ\)](#) を参照してください。

ETM によって送信されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから ([セクション 33.17: TPIU \(トレースポートインタフェースユニット\)](#) を参照)、完全なパケットシーケンスをデバッグホストに出力します。

### 33.15.2 信号プロトコル、パケットタイプ

これについては、ARM IHI 0014N ドキュメントの第 7 章「ETMv3 信号プロトコル」を参照してください。



## 33.15.3 主な ETM レジスタ

レジスタの詳細については、ARM IHI 0014N 仕様の第 3 章を参照してください。

表 196. 主な ETM レジスタ

アドレス	レジスタ	詳細
0xE0041FB0	ETM ロックアクセス	他の ETM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
0xE0041000	ETM 制御	このレジスタは、ETM の一般的な動作（たとえば、トレースを有効にする方法）を制御します。
0xE0041010	ETM ステータス	このレジスタは、トレースおよびトリガロジックの現在のステータスに関する情報を提供します。
0xE0041008	ETM トリガイベント	このレジスタは、トリガを制御するイベントを定義します。
0xE004101C	ETM トレース有効制御	このレジスタは、どのコンパレータが選択されるかを定義します。
0xE0041020	ETM トレース有効イベント	このレジスタは、トレース有効化イベントを定義します。
0xE0041024	ETM トレース開始/停止	このレジスタは、トレースをそれぞれ開始および停止するためにトリガソースによって使用されるトレースを定義します。

## 33.15.4 設定例

TPIU に単純な値を出力するには、

- TPIU を設定し、I/O\_TRACEN を有効にして STM32F3xx デバッグ設定レジスタで TRACE I/O を割り当てます。
- ETM ロックアクセスレジスタに 0xC5ACCE55 を書き込んで、ETM レジスタへの書き込みアクセスをロック解除します。
- 制御レジスタに 0x00001D1E を書き込みます（トレースを設定します）。
- トリガイベントレジスタに 0000406F を書き込みます（トリガイベントを定義します）。
- トレース有効イベントレジスタに 0000006F を書き込みます（開始/停止するイベントを定義します）。
- トレース開始/停止レジスタに 00000001 を書き込みます（トレースを有効にします）。
- ETM 制御レジスタに 0000191E を書き込みます（設定の終了）。

## 33.16 MCU デバッグコンポーネント (DBGMCU)

MCU デバッグコンポーネントは、デバッガによる以下のサポート機能を支援します。

- 低電力モード
- ブレークポイントにおける、タイマ、ウォッチドッグ、I2C、および bxCAN のクロック制御
- TRACE ピンの割当て制御

## 33.16.1 低電力モードのデバッグサポート

低電力モードに入るには、WFI または WFE 命令を実行する必要があります。

MCU はいくつかの低電力モードを実装しており、CPU クロックを無効にしたり、CPU の消費電力を低減したりすることができます。

デバッグセッション中には、コアは FCLK や HCLK をオフにすることはできません。デバッグ時のデバッグ接続に必要なため、これらをアクティブな状態に保つ必要があります。MCU は、ユーザが低電力モードでソフトウェアをデバッグするための特殊な手段を備えています。

このため、デバッグホストは、最初にいくつかのデバッグ設定レジスタをセットして、低電力モード動作を変更する必要があります。

- SLEEP モードでは、DBGMCU\_CR レジスタの DBG\_SLEEP ビットをデバッグによって事前にセットする必要があります。これによって、HCLK には FCLK と同じクロックが供給されます (システムクロックはソフトウェアによって事前に設定されています)。
- STOP モードでは、DBG\_STOP ビットをデバッグによって事前にセットする必要があります。これによって、内部 RC オシレータが、STOP モードで FCLK と HCLK にクロックを供給できます。

## 33.16.2 タイマ、ウォッチドッグ、bxCAN、および I<sup>2</sup>C のデバッグサポート

ブレークポイントにおいては、以下に示すタイマのカウンタやウォッチドッグの動作方法を選択する必要があります。

- ブレークポイントの中でもカウントを継続できます。この動作は、たとえば、PWM がモータを制御しているときに一般的に必要です。
- ブレークポイントの中でカウントを停止できます。この動作はウォッチドッグ用に必要です。

bxCAN の場合、ブレークポイントにおいては受信レジスタの更新をブロックするように選択できます。

I<sup>2</sup>C の場合、ブレークポイントにおいては SMBUS タイムアウトをブロックするように選択できます。

## 33.16.3 デバッグ MCU 設定レジスタ

このレジスタを使用して、デバッグ中に MCU を設定できます。次の設定が可能です。

- 低電力モードのサポート
- タイマおよびウォッチドッグカウンタのサポート
- bxCAN 通信のサポート
- TRACE ピンの割当て

この DBGMCU\_CR は、アドレス 0xE0042004 にある外部 PPB バスに配置されます。

このレジスタは PORESET によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

デバッグホストがこれらの機能をサポートしない場合でも、ユーザソフトウェアによってこれらのレジスタへの書き込みが可能です。

### DBGMCU\_CR

アドレス : 0xE004 2004

32 ビットアクセスのみサポートされます。

POR リセット : 0x0000 0000 (システムリセットではリセットされません)



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_	TRACE			DBG_	DBG_	DBG_	
								MODE	_			STAND	STOP	SLEEP	
								[1:0]	IOEN			BY			
								rw	rw	rw		rw	rw	rw	

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:5 **TRACE\_MODE[1:0]** および **TRACE\_IOEN** : TRACE ピン割り当て制御

– TRACE\_IOEN=0 の場合 :

TRACE\_MODE=xx : TRACE ピンは割り当てられません (デフォルト)。

– TRACE\_IOEN=1 の場合 :

– TRACE\_MODE=00 : 非同期モードの TRACE ピン割り当てです。

– TRACE\_MODE=01 : TRACEDATA サイズが 1 の同期モードの TRACE ピン割り当てです。

– TRACE\_MODE=10 : TRACEDATA サイズが 2 の同期モードの TRACE ピン割り当てです。

– TRACE\_MODE=11 : TRACEDATA サイズが 4 の同期モードの TRACE ピン割り当てです。

**注 :** *STM32F303x6/8 および STM32F328x8 デバイスでは、同期トレースは使用できません。ビット 7:5 は予約済みであり、0 のままにしておかなければなりません。*

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG\_STANDBY** : デバッグ STANDBY モード

0 : (FCLK=オフ、HCLK=オフ) デジタル部全体が電源オフになります。

ソフトウェアから見て、STANDBY モードから抜けることは、リセットベクタのフェッチと同じになります (ただし、いくつかのステータスビットは MCU が STANDBY モードから再開していることを示す)。

1 : (FCLK=オン、HCLK=オン) この場合、デジタル部は電源オフ状態ではなく、FCLK と HCLK は引き続きアクティブ状態の内部 RC オシレータから供給されます。さらに、MCU は STANDBY モード中にシステムリセットを生成するため、STANDBY モードから抜けることはリセットからのフェッチと同じになります。

ビット 1 **DBG\_STOP** : デバッグ STOP モード

0 : (FCLK=オフ、HCLK=オフ) STOP モードでは、クロックコントローラがすべてのクロック (HCLK と FCLK を含む) を無効にします。STOP モードから抜けると、クロック設定はリセット後の場合と同じになります (CPU は 8 MHz の内部 RC オシレータ (HSI) から供給されます)。したがって、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります。

1 : (FCLK=オン、HCLK=オン) この場合、STOP モードに入ると、FCLK と HCLK は STOP モードでもアクティブ状態の内部 RC オシレータから供給されます。STOP モードから抜けるとき、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります (DBG\_STOP=0 の場合と同様です)。

ビット 0 **DBG\_SLEEP** : デバッグ SLEEP モード

0 : (FCLK=オン、HCLK=オフ) SLEEP モードでは、FCLK はソフトウェアによって事前に設定されたシステムクロックによって駆動され、HCLK は無効にされます。

SLEEP モードでは、クロックコントローラの設定はリセットされず、事前にプログラミングされた状態のままです。したがって、SLEEP モードから抜けるときに、ソフトウェアでクロックコントローラを再設定する必要はありません。

1 : (FCLK=オン、HCLK=オン) この場合、SLEEP モードに入ると、HCLK には FCLK と同じクロック (ソフトウェアによって事前に設定されたシステムクロック) が供給されます。

## 33.16.4 デバッグ MCU APB1 フリーズレジスタ (DBGMCU\_APB1\_FZ)

DBGMCU\_APB1\_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB1 ペリフェラルに関する設定が可能です。

- タイマクロックカウンタの凍結
- I2C SMBUS タイムアウトの凍結
- ウィンドウ型ウォッチドッグおよび独立型ウォッチドッグのカウンタの凍結サポート

この DBGMCU\_APB1\_FZ は、アドレス 0xE0042008 にある外部 PPB バスに配置されます。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

アドレス : 0xE004 2008

32 ビットアクセスのみサポートされます。

パワーオンリセット (POR) : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DBG_I2C3_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	DBG_CAN_STOP	Res.	Res.	DBG_I2C2_SMBUS_TIMEOUT <sup>(1)</sup>	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.
						r/w			r/w	r/w					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	Res.	Res.	Res.	DBG_TIM7_STOP	DBG_TIM6_STOP	Res.	DBG_TIM4_STOP <sup>(1)</sup>	DBG_TIM3_STOP	DBG_TIM2_STOP
			r/w	r/w	r/w					r/w	r/w		r/w	r/w	r/w

1. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ。

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **DBG\_I2C3\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。

ビット 25 **DBG\_CAN\_STOP** : コア停止時にデバッグ CAN は停止

- 0 : 通常モードと同じ動作です。
- 1 : CAN2 受信レジスタは停止状態です。

ビット 24:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DBG\_I2C2\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止 (STM32F303xB/C および STM32F358xC デバイスでのみ使用可能)

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。



ビット 21 **DBG\_I2C1\_SMBUS\_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止  
0 : 通常モードと同じ動作です。  
1 : SMBUS タイムアウトは凍結されます。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DBG\_IWDG\_STOP** : コア停止時にデバッグ独立型ウォッチドッグは停止  
0 : コアが停止しても独立型ウォッチドッグカウンタのクロックは継続されます。  
1 : コア停止時に独立型ウォッチドッグカウンタのクロックは停止します。

ビット 11 **DBG\_WWDG\_STOP** : コア停止時にデバッグウィンドウ型ウォッチドッグは停止  
0 : コアが停止してもウィンドウ型ウォッチドッグカウンタのクロックは継続されます。  
1 : コア停止時にウィンドウ型ウォッチドッグカウンタのクロックは停止します。

ビット 10 **DBG\_RTC\_STOP** : コア停止時にデバッグ RTC は停止  
0 : コアが停止しても当該 RTC カウンタのクロックは供給されます。  
1 : コア停止時に当該 RTC カウンタのクロックは停止します。

ビット 9:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **DBG\_TIM7\_STOP** : コア停止時に TIM7 カウンタは停止  
0 : コアが停止しても TIM7 カウンタのクロックは供給されます。  
1 : コア停止時に TIM7 カウンタのクロックは停止します。

ビット 4 **DBG\_TIM6\_STOP** : コア停止時に TIM6 カウンタは停止  
0 : コアが停止しても TIM6 カウンタのクロックは供給されます。  
1 : コア停止時に TIM6 カウンタのクロックは停止します。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG\_TIM4\_STOP** : コア停止時に TIM4 カウンタは停止 (STM32F303xB/C および STM32F358xC デバイスでのみ使用可能)  
0 : コアが停止しても TIM4 カウンタのクロックは供給されます。  
1 : コア停止時に TIM4 カウンタのクロックは停止します。

ビット 1 **DBG\_TIM3\_STOP** : コア停止時に TIM3 カウンタは停止  
0 : コアが停止しても TIM3 カウンタのクロックは供給されます。  
1 : コア停止時に TIM3 カウンタのクロックは停止します。

ビット 0 **DBG\_TIM2\_STOP** : コア停止時に TIM2 カウンタは停止  
0 : コアが停止しても TIM2 カウンタのクロックは供給されます。  
1 : コア停止時に TIM2 カウンタのクロックは停止します。

## 33.16.5 デバッグ MCU APB2 凍結レジスタ (DBGMCU\_APB2\_FZ)

DBGMCU\_APB2\_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB2 ペリフェラルに関する設定が可能です。

- タイマクロックカウンタの凍結

アドレス 0xE004 200C にある外部 PPB バスに配置されています。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

アドレス : 0xE004 200C

32 ビットアクセスのみサポートされます。

POR : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM20_STOP <sup>(1)</sup>	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP	DBG_TIM8_STOP <sup>(2)</sup>	DBG_TIM1_STOP
											r/w	r/w	r/w	r/w	r/w

1. STM32F303xD/E および STM32F398xE でのみ使用できます。
2. STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE デバイスのみ。

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **DBG\_TIMx\_STOP** : コア停止時に TIMx カウンタは停止 (x=1、8、15 .. 17)

0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。

1 : コア停止時に当該タイマカウンタのクロックは停止します。

**注 :** ビット 1 およびビット 5 は STM32F303x6/8 および STM32F328x8 で予約済みです。

## 33.17 TPIU (トレースポートインタフェースユニット)

### 33.17.1 概要

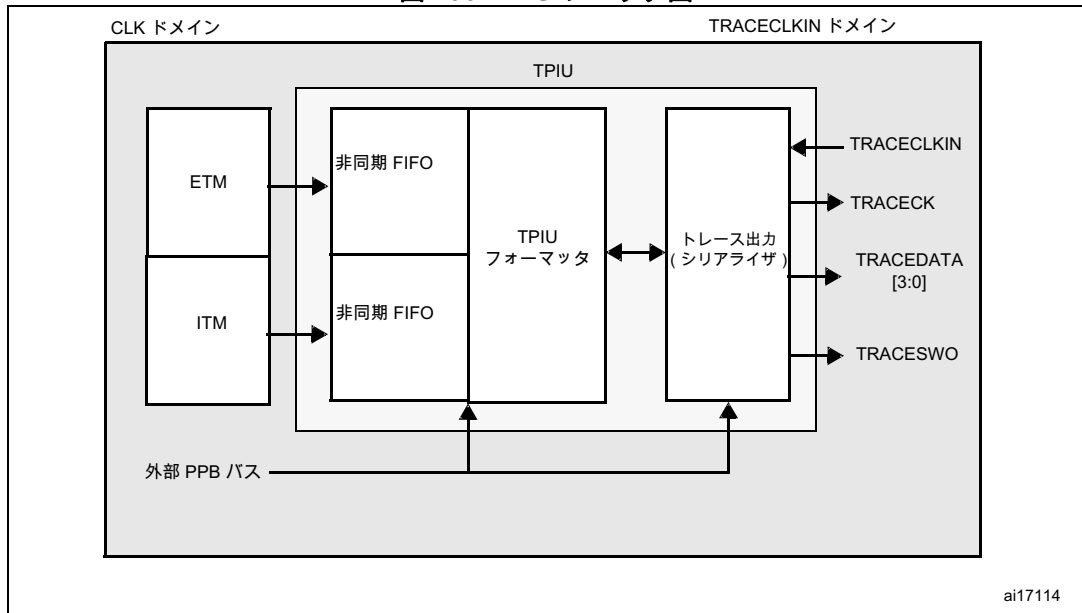
TPIU は、ITM と ETM からのオンチップトレースデータ間のブリッジとしての役割を果たします。

出力データストリームによってカプセル化されたトレースソース ID は、トレースポートアナライザ (TPA) によってキャプチャされます。

コアには、特に低コストのデバッグ向けの単純な TPIU (特別バージョンの CoreSight TPIU から構成) が内蔵されています。



図 406. TPIU ブロック図



### 33.17.2 TRACE ピンの割当て

- 非同期モード  
非同期モードは、1 個のピンを追加で必要とし、すべてのパッケージで使用できます。非同期モードは、JTAG モードではなくシリアルワイヤモードを使用する場合にのみ使用できます。

表 197. 非同期 TRACE ピンの割当て

TPIU ピン名	TRACE 同期モード		STM32F3xx ピン割り当て
	タイプ	説明	
TRACESWO	○	TRACE 非同期データ出力	PB3

- 同期モード  
同期モードは、データトレースサイズに応じて 2~6 本のピンを追加で必要とし、大型パッケージでのみ使用できます。さらに、JTAG モードとシリアルワイヤモードでも使用でき、非同期トレースよりも優れた帯域幅出力機能を提供します。

表 198. 同期 TRACE ピンの割当て

TPIU ピン名	TRACE 同期モード		STM32F3xx ピン割り当て
	タイプ	説明	
TRACECK	○	TRACE クロック	PE2
TRACED[3:0]	○	TRACE 同期データ出力 1、2、または 4 とすることができます。	PE[6:3]



## TPIU TRACE ピンの割当て

デフォルトでは、これらのピンは割り当てられません。これらのピンを割り当てるには、**MCU デバッグコンポーネント設定レジスタ**の TRACE\_IOEN および TRACE\_MODE ビットをセットします。この設定はデバッグホストで行う必要があります。

さらに、割り当てるピン数は、トレースの設定（非同期／同期）によって異なります。

- **非同期モード**：1本のピンが追加が必要です。
- **同期モード**（STM32F3xxでのみ使用可能）：データトレースポートレジスタのサイズ（1、2、または4）に応じて、2～5本のピンが追加が必要です。
  - TRACECK
  - TRACED(0)、ポートサイズが1、2、または4に設定された場合
  - TRACED(1)、ポートサイズが2または4に設定された場合
  - TRACED(2)、ポートサイズが4に設定された場合
  - TRACED(3)、ポートサイズが4に設定された場合

TRACE ピンを割り当てるには、デバッグホストは、デバッグ MCU 設定レジスタ (DBGMCU\_CR) の TRACE\_IOEN ビットと TRACE\_MODE[1:0] ビットをプログラミングする必要があります。デフォルトでは、TRACE ピンは割り当てられません。

このレジスタは、外部 PPB に配置され、PORESET によってリセットされます（システムリセットではありません）。このレジスタは、システムリセット中にデバッグによる書き込みができます。

表 199. 柔軟性の高い TRACE ピン割り当て

DBGMCU_CR レジスタ		ピン割当て：	TRACE IO ピンの割当て					
TRACE_IOEN	TRACE_MODE [1:0]		PB3 / JTDO / TRACESWO	PE2 / TRACECK	PE3 / TRACED[0]	PE4 / TRACED[1]	PE5 / TRACED[2]	PE6 / TRACED[3]
0	XX	トレースなし (デフォルト状態)	解放 <sup>(1)</sup>	-	-	-	-	-
1	00	非同期トレース	TRACESWO	-	-	解放 (GPIO として使用可能)		
1	01	同期トレース、1ビット	解放 <sup>(1)</sup>	TRACECK	TRACED[0]	-	-	-
1	10	同期トレース、2ビット		TRACECK	TRACED[0]	TRACED[1]	-	-
1	11	同期トレース、4ビット		TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

1. シリアルワイヤモードが使用されているときには解放されます。しかし、JTAG が使用されると、JTDO に割り当てられます。

**注：** デフォルトでは、TPIU の TRACECLKIN 入カロックは GND に接続されています。このクロックは、TRACE\_IOEN ビットがセットされてから 2 クロックサイクル後に HCLK に割り当てられます。



デバッグは、TPIU の SPP\_R (選択ピンプロトコル) レジスタの PROTOCOL[1:0] ビットに書き込むことによって、トレースモードをプログラミングする必要があります。

- PROTOCOL = 00 : トレースポートモード (同期モード)。
- PROTOCOL = 01 または 10 : シリアルワイヤ (マンチェスタまたは NRZ) モード (非同期モード)。  
デフォルト状態は 01 です。

次に、TPIU の CPSPS\_R (カレント同期化ポートサイズレジスタ) のビット [3:0] に書き込むことによって TRACE ポートサイズも設定します。

- 0x1 : 1 ピンの場合 (デフォルト状態)
- 0x2 : 2 ピンの場合
- 0x8 : 4 ピンの場合

### 33.17.3 TPIU フォーマッタ

フォーマッタプロトコルは、データを 16 バイトフレームで出力します。

- 7 バイトのデータ
- 8 バイトの混合バイトであり、以下の構成になっています。
  - 1 ビット (LSB) : データバイト ("0") または ID バイト ("1") であることを示します。
  - 7 ビット (MSB) : データまたはソース ID トレースの変更とすることができます。
- 1 バイトの補助ビット : 各ビットは 8 つの混合バイトのいずれかに対応します。
  - 対応するバイトがデータであった場合、このビットはデータのビット 0 を示します。
  - 対応するバイトが ID 変更であった場合、このビットはその ID 変更が有効になるタイミングを示します。

**注 :** *詳細については、ARM® CoreSight Architecture Specification v1.0 (ARM® IHI 0029B) を参照してください。*

### 33.17.4 TPIU フレーム同期パケット

TPIU は 2 種類の同期パケットを生成できます。

- フレーム同期パケット (フルワード同期パケット)  
これはワード 0x7F\_FF\_FF\_FF (LSB ファーストで発行) から構成されます。ID ソースコード 0x7F が使用されていない場合、このシーケンスは他のタイミングでは発生できません。  
このパケットはフレーム *間* で定期的に出力されます。  
連続モードでは、同期フレームが検出されると、TPA がこれらすべてのフレームを破棄する必要があります。
- ハーフワード同期パケット  
これはハーフワード 0x7F\_FF (LSB ファーストで発行) から構成されます。  
このパケットはフレーム *間または内* で定期的に出力されます。  
これらのパケットは連続モードでのみ生成され、TPA は TRACE ポートが IDLE モードである (TRACE はキャプチャされない) ことを検出できるようになります。TPA によって検出されたパケットは、破棄する必要があります。

## 33.17.5 同期フレームパケットの送信

コアの TPIU には、同期カウンタレジスタは実装されていません。したがって、同期トリガは DWT によってのみ生成できます。DWT 制御レジスタ (SYNCTAP[11:10] ビット) と DWT カレント PC サンプラサイクルカウンタレジスタの説明を参照してください。

TPIU フレーム同期パケット (0x7F\_FF\_FF\_FF) は次の場合に発行されます。

- 毎回の TPIU リセット解除後。このリセットは、TRACECLKIN クロックの立ち上がりエッジに同期して解除されます。つまり、DBGMCU\_CFG レジスタの TRACE\_IOEN ビットがセットされると、このパケットが送信されます。この場合、ワード 0x7F\_FF\_FF\_FF の後に、フォーマットされたパケットは続きません。
- 毎回の DWT トリガ時 (DWT は事前に設定されているものとします)。次の 2 つの場合があります。
  - ITM の SYNENA ビットがリセットされた場合、ワード 0x7F\_FF\_FF\_FF だけが発行され、フォーマットされたストリームが続くことはありません。
  - ITM の SYNENA ビットがセットされた場合、TPIU によってフォーマット (トレースソース ID を追加) された ITM 同期パケットが続きます (0x80\_00\_00\_00\_00\_00)。

## 33.17.6 同期モード

トレースデータの出力サイズは 4、2、または 1 ピンから選択できます TRACED(3:0)

出力クロックはデバッグに出力されます (TRACECK)。

なお、TRACECLKIN は内部で駆動され、TRACE が使用される場合にのみ HCLK に接続されます。

**注:** *同期モードでは、必ずしも安定したクロック周波数を提供する必要はありません。*

TRACE I/O (TRACECK を含む) は TRACECLKIN (HCLK と同等) の立ち上がりエッジで駆動されます。したがって、TRACECK の出力周波数は HCLK/2 になります。

## 33.17.7 非同期モード

これは 1 ピン (非同期出力ピン TRACESWO) のみを使用してトレースを出力する低コストな方法です。明らかにその帯域幅は制限されます。

SW-DP ピンを使用する場合、TRACESWO ピンは JTDO ピンと多重化されます。このように、この機能はすべての STM32F3xx パッケージで使用できます。

この非同期モードでは、TRACECLKIN に一定の周波数が要求されます。標準的な UART (NRZ) のキャプチャ機構では、5% の精度が必要です。マンチェスタエンコード方式では、10% までの誤差が許されます。

## 33.17.8 STM32F3xx 内の TRACECLKIN 接続

STM32F3xx では、この TRACECLKIN 入力は内部で HCLK に接続されます。つまり、非同期トレースモードでは、アプリケーションは、CPU 周波数が安定しているタイムフレームのみを使用できます。

**注:** *重要: 非同期トレースを使用する場合は、以下のことに注意してください。*

*STM32F3xx MCU のデフォルトクロックは、内部 RC オシレータです。リセット中の周波数はリセット解除後の周波数とは異なります。これは、RC 較正はシステムリセット中のデフォルト動作であり、システムリセット解除のたびに更新されるからです。*

*したがって、トレースポートアナライザ (TPA) は、システムリセット中に (TRACE\_IOEN ビットによって) トレースを有効にするべきではありません。なぜなら、同期フレームパケットは、リセット解除後に送信されるトレースパケットとは異なるビット時間で発行されるからです。*



## 33.17.9 TPIU レジスタ

TPIU APB レジスタは、デバッグ例外およびモニタ制御レジスタ (DEMCR) の TRCENA ビットがセットされた場合にのみ読み出し/書き込みが可能です。そうでない場合、これらのレジスタはゼロとして読み出されます (このビットの出力は TPIU の PCLK を有効にします)。

表 200. 重要な TPIU レジスタ

アドレス	レジスタ	説明
0xE0040004	現在のポートサイズ	<p>トレースポートサイズの選択 :</p> <p>ビット 0 : ポートサイズ = 1                      ビット 1 : ポートサイズ = 2                      ビット 2 : ポートサイズ = 3、サポートされません                      ビット 3 : ポートサイズ = 4</p> <p>1 ビットのみセットする必要があります。デフォルトでは、ポートサイズは 1 ビットです。(0x00000001)</p>
0xE00400F0	選択ピンプロトコル	<p>トレースポートプロトコルの選択 :</p> <p>ビット 1:0=</p> <p>00 : 同期トレースポートモード                      01 : シリアルワイヤ出力 - マンチェスタ (デフォルト値)                      10 : シリアルワイヤ出力 - NRZ                      11 : 予約済み</p>
0xE0040304	フォーマッタおよびフラッシュ制御	<p>ビット 31-9 = 常に 0                      ビット 8 = TriglIn = 常に "1" で、トリガが指定されたことを示します                      ビット 7-4 = 常に 0                      ビット 3-2 = 常に 0</p> <p>ビット 1 = EnFCont同期トレースモード (Select_Pin_Protocol レジスタのビット 1:0 = 00) では、このビットは強制的に "1" にされ、フォーマッタは自動的に連続モードで有効にされます。非同期モード (Select_Pin_Protocol レジスタのビット 1:0 &lt;&gt; 00) では、このビットの書き込みによって、フォーマッタを有効/無効にできます。</p> <p>ビット 0 = 常に 0</p> <p>このレジスタのデフォルト値は 0x102 です。</p> <p>注 : 同期モードでは、TRACECTL ピンがチップ外に配置されないため、フォーマッタは常に連続モードで有効になります。このように、フォーマッタは、トレースパケットの転送元を識別するための制御パケットを挿入します。</p>
0xE0040300	フォーマッタおよびフラッシュステータス	<p>Cortex-M4<sup>®</sup>F では使用されず、常に 0x00000008 が読み出されます。</p>

## 33.17.10 設定例

- デバッグ例外およびモニタ制御レジスタ (DEMCR) の TRCENA ビットをセットします。
- TPIU カレントポートサイズレジスタに希望する値を書き込みます (デフォルトは 0x1 で 1 ビットのポートサイズ)。
- TPIU フォーマッタおよびフラッシュ制御レジスタに 0x102 (デフォルト値) を書き込みます。
- TPIU 選択ピンプロトコルレジスタに書き込んで同期/非同期モードを選択します。例 : 非同期 NRZ モード (UART に類似) の場合は 0x2 を書き込みます。
- DBGMCU 制御レジスタに 0x20 (IO\_TRACEEN ビット) を書き込み、TRACE I/O を非同期モードとします。このとき、TPIU 同期パケット (FF\_FF\_FF\_7F) が発行されます。
- ITM を設定し、ITM スティムラスレジスタに書き込んで値を出力します。

## 33.18 DBG レジスタマップ

次の表にデバッグレジスタの一覧を示します。

表 201. DBG レジスタマップとリセット値

アドレス	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0xE0042000	DBGMCU_IDCODE	REV_ID												DEV_ID																					
	リセット値 <sup>(1)</sup>	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X		
0xE0042004	DBGMCU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値																										0	0	0						
0xE004 2008	DBGMCU_APB1_FZ	Res.	DBG_I2C3_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	DBG_CAN_STOP	Res.	Res.	DBG_I2C2_SMBUS_TIMEOUT	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	Res.	Res.	DBG_TIM20_STOP	DBG_TIM7_STOP	DBG_TIM6_STOP	Res.	Res.	Res.	Res.		
	リセット値		0					0			0	0									0	0	0					0	0	0					
0xE004 200C	DBGMCU_APB2_FZ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																		

1. リセット値は製品によって異なります。詳細については、[セクション 33.6.1 : MCU デバイス ID コード](#)を参照してください。



## 34 デバイス電子署名

デバイス電子署名は、フラッシュメモリモジュールのシステムメモリ領域に格納され、デバッグインタフェースまたは CPU を使用して読み出すことができます。電子署名に含まれる出荷時にプログラムされた識別データおよび較正データを使用すれば、ユーザファームウェアやその他の外部デバイスは、STM32F3xx マイクロコントローラの特性に自動的に整合させることができます。

### 34.1 ユニークデバイス ID レジスタ (96 ビット)

このユニークデバイス識別子は、以下の用途に最適です。

- シリアル番号（例：USB 文字列シリアル番号やその他のエンドアプリケーション）として使用
- 内部フラッシュメモリをプログラムする前に、このユニーク ID をソフトウェア暗号プリミティブやプロトコルと組み合わせて使用する際に、フラッシュメモリ内のコードのセキュリティを高めるためのセキュリティキーの一部として使用
- セキュアなブートプロセスなどの起動に使用

96 ビットのユニークデバイス識別子は、状況やデバイスの違いとは無関係にユニークなリファレンス番号を提供します。ユーザは、これらのビットを変更できません。

ベースアドレス：0x1FFF F7AC

アドレスオフセット：0x00

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

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

ビット 31:0 **UID[31:0]** : BCD 形式で表されたウェーハ上の X および Y 座標

アドレスオフセット : 0x04

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:8 **UID[63:40]** : LOT\_NUM[23:0]

ロット番号 (ASCII で符号化)

ビット 7:0 **UID[39:32]** : WAF\_NUM[7:0]

ウェーハ番号 (8 ビットの符号なし番号)

アドレスオフセット : 0x08

読み出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **UID[95:64]** : LOT\_NUM[55:24]

ロット番号 (ASCII で符号化)

## 34.2 メモリサイズデータレジスタ

### 34.2.1 フラッシュサイズデータレジスタ

ベースアドレス : 0x1FFF F7CC

アドレスオフセット : 0x00

読み出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **FLASH\_SIZE[15:0]** : フラッシュメモリサイズ

このビットフィールドは、デバイスのフラッシュメモリサイズを KB 単位で示します。

たとえば、0x040 は 64 KB に対応します。



## 35 改版履歴

表 202. 文書改版履歴

日付	版	変更内容
2013 年 12 月 16 日	1	初版リリース。
2013 年 3 月 8 日	2	<p><b>内蔵 SRAM およびフラッシュメモリ :</b>  <i>セクション 1.1 : システムアーキテクチャ</i> および <i>セクション 2.3.1 : パリティチェック</i> を更新。  <i>セクション 1.5.7 : オプションバイトレジスタ (FLASH_OBR)</i> の RDPRT の説明を更新。<i>表 8 : オプションバイトの説明</i> のアドレス 0x1FFF F80C の WRP3 を更新。  <i>セクション 1.6 : フラッシュレジスタマップ</i> の FLASH_SR レジスタを更新。  <i>表 8 : オプションバイトの説明</i> を更新。</p> <p><b>PWR :</b>  <i>図 1 : 電源の概要 (STM32F303x デバイス)</i>、<i>セクション 1.1 : 電源の概要</i> および <i>セクション 1.1.1 : 独立した A/D および D/A コンバータ用電源と基準電圧</i> を更新。  <i>セクション 1.1.2 : バッテリバックアップドメインの AN2586 ~ AN4206</i> を変更。  <i>セクション 1.4.2 : 電源制御/ステータスレジスタ (PWR_CSR)</i> で EWUP3 を追加、EWUP2 の説明を更新、<b>VREFINTRDYF</b> ビットを追加。</p> <p><b>RCC :</b>  <i>表 1 : RCC レジスタマップとリセット値</i> に USART3EN、UART4EN、および UART5EN を追加。  <i>セクション 1.2.5 : LSI クロック</i> の LSI クロックの最大周波数を 50 kHz に変更。<i>セクション 1.2.12 : I2S クロック (STM32F303xB/C および STM32F358xC のみ)</i> を追加。  <i>セクション 1.4.2 : クロック設定レジスタ (RCC_CFGR)</i>、<i>セクション 1.4.4 : APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR)</i> の SYSCFGRST、<i>セクション 1.4.5 : APB1 ペリフェラルリセットレジスタ (RCC_APB2RSTR)</i> の SPI2RST/SPI3RST、および <i>セクション 1.4.6 : AHB ペリフェラルクロック有効レジスタ (RCC_AHBENR)</i> の ADC34EN/ADC12 EN を更新。  <i>セクション 1.4.4 : APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR)</i> の APB を APB2 に置換。  <i>セクション 1.4.5 : APB1 ペリフェラルリセットレジスタ (RCC_APB1RSTR)</i> および <i>セクション 1.4.2 : クロック設定レジスタ (RCC_CFGR)</i> を更新。  <i>セクション 1.4.14 : RCC レジスタマップ</i> を更新。</p>



表 202. 文書改版履歴 (続き)

日付	版	変更内容
2013 年 3 月 8 日	2 (続き)	<p><b>GPIO :</b></p> <p>表 23 : GPIO レジスタマップとリセット値の GPIOA_OSPEEDR および GPIOB_OSPEEDR のリセット値を更新。</p> <p>セクション 9.3.1 : 汎用 I/O (GPIO) の JTMS/SWDAT を JTMS/SWDIO に置換。</p> <p>セクション 9.4.1 : GPIO ポートモードレジスタ (GPIOx_MODER) (x =A ~ F) に GPIOF_MODER に関する注を追加。</p> <p>セクション 9.4.12 : GPIO レジスタマップ: GPIOA_MODER リセット値を更新、GPIOB_MODER、GPIOA_OSPEEDR、GPIOB_OSPEEDR、GPIOB_PUPDR を追加、GPIOA_PUPDR のリセット値を更新、GPIOx_LCKR レジスタを更新。</p> <p><b>SYSCFG :</b></p> <p>セクション 1.1.1 : SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1)、セクション 1.1.6 : SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4) および セクション 1.1.7 : SYSCFG 設定レジスタ 2 (SYSCFG_CFGR2) を更新。</p> <p><b>DMA :</b></p> <p>図 22 : DMA ブロック図を更新。</p> <p>図 11.3.7 : DMA リクエストマッピングを変更。</p> <p><b>割り込みとイベント :</b></p> <p>表 1 : STM32F302xB/C ベクタテーブルを更新。</p> <p>EXTI_IMR2、EXTI_EMR2、EXTI_RTISR2、EXTI_FTSR2: 0x2C、EXTI_SWIER2、および EXTI_PR2 のアドレスオフセットを変更。</p> <p>セクション 14.3.4 および セクション 14.3.10 の EXTI_FTSR1 に関する注を更新。</p> <p>セクション 1.3.7 : 割り込みマスクレジスタ (EXTI_IMR2) の注を更新。</p> <p>セクション 1.3.13 : EXTI レジスタマップ</p> <p><b>ADC :</b></p> <p>セクション 1.1 : 概要を更新。図 1 : ADC ブロック図、図 3 : ADC1 および ADC2 接続性、および 図 4 : ADC3 および ADC4 接続性を更新。セクション 1.3.5 : スレーブ AHB インタフェースを追加。</p> <p>セクション 1.3.7 : シングルエンドおよび差動入力のチャネルの注意および注を更新。セクション 1.3.8 : 較正 (ADCAL、ADCALDIF、ADC_CALFACT) を変更。</p> <p>ADC_CALFACT_S と ADC_CALFACT_D をそれぞれ CALFACT_S と CALFACT_D に変更。</p>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2013 年 3 月 8 日	2 (続き)	<p><b>ADC (続き)</b>            セクション: シングル ADC によるシングルエンドおよび差動アナログ入力の変換のステップ 3 を更新。            表: ADC3 &amp; ADC4 - レギュラチャネルの外部トリガの表名を訂正。            セクション: チャネル単位でプログラム可能なサンプリング時間 (SMPR1, SMPR2) の高速チャネルに関する最小サンプリング時間の表および関連する例を削除し、データシートに参照を使用。            セクション: 実行中の変換の停止 (ADSTP、JADSTP) を更新。            セクション: 自動インジェクションモードに注を追加。            セクション: DMA を使用した変換の管理のダブルバッファモードを削除。            図: JADSTP=1 のセットによるコンテキストの JSQR キューのフラッシュ (JQM=0) を更新。実行中の変換以外で JADSTP が発生するケースおよび            セクション: Disabling the queue を追加。            セクション: 自動遅延変換モード (AUTDLY)、セクション: 独立インジェクト変換に対応したレギュラ同時モードを更新。</p> <p><b>DAC :</b>            表 50 : 外部トリガの TIM5 を TIM15 に、AIEC ライン 9 を EXTI ライン 9 に変更。DAC_CR のビット 21:19 の説明で TIM5 を TIM15 に変更。</p> <p><b>COMP :</b>            COMPx_CSR レジスタ (x = 1 ~ 7) の COMPx_OUT_TIM_SEL を COMPxOUTSEL に変更 (セクション 17 を参照)。            セクション 15.5.1 : COMP1 制御/ステータスレジスタ (COMP1_CSR) の COMP1SW1 を COMP1_INP_DAC に置換。            セクション 15.5.2 : COMP2 制御/ステータスレジスタ (COMP2_CSR) の COMP2_WINDOW_MODE を COMP2WINMODE に置換。</p> <p><b>OPAMP :</b>            セクション 1 : オペアンプの AOPx を OPAMPx に置換。            セクション 1.3 : OPAMP の機能説明の概要を更新 および セクション 1.3.2 : クロックを追加。            OPAMPx_CSR レジスタ (セクション 18.4.1 ~ セクション 18.4.4) により、OPAMPx_CSR レジスタ (x = 1 ~ 4) の OUT-CAL、CAL_SEL、および CAL_ON を OUTCAL、CALSEL、CALON に変更。TSTREF のアクセス権を「rw」にセット。OPAMPx_EN ビットを OPAMPxEN に名称変更。            VOPAMPx をすべて VREFOPAMPx に置換。</p>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2013 年 3 月 8 日	2 (続き)	<p><b>高機能制御タイマ (TIM1/8) :</b>  <a href="#">図 1 : 高機能制御タイマのブロック図</a>を訂正。  <a href="#">セクション 1.3.15 : ブレーク機能の使用</a>を更新。  <a href="#">セクション 1.3.15 : ブレーク機能の使用</a>を更新。</p> <p><b>汎用タイマ (TIM2/3/4)</b>  <a href="#">セクション 1.4.3 : TIMx スレーブモード制御レジスタ (TIMx_SMCR)</a>を更新。<a href="#">セクション 1.4.5 : TIMx ステータスレジスタ (TIMx_SR)</a>の UIF ビットを変更。<a href="#">セクション 1.4.6 : TIMx イベント生成レジスタ (TIMx_EGR)</a>および<a href="#">セクション 1.4.9 : TIMx キャプチャ/比較有効レジスタ (TIMx_CCER)</a>を更新。</p> <p><b>汎用タイマ (TIM15/16/17)</b>  <a href="#">図 2 : TIM16 および TIM17 ブロック図</a>を更新。  <a href="#">セクション 1.4.13 : ブレーク機能の使用</a>を更新。</p> <p><b>RTC</b>            セクション全体の「パワーオンリセット」を「バックアップドメインリセット」に変更。  <a href="#">セクション 24.7.15 : RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)</a>を更新。</p> <p><b>I2C :</b>  <a href="#">図 212 : セットアップおよびホールドタイミング</a>を更新。  <a href="#">表 75 : アナログフィルタとデジタルフィルタの比較</a>を更新。  <a href="#">図 228 : N&gt;255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート</a>を訂正。<a href="#">表 76 : I2C-SMBUS 仕様のデータのセットアップおよびホールド時間</a>でパラメータ「データホールド時間」の最大値を削除し、行「データ有効時間」を追加。  <a href="#">セクション 25.5 : I2C 低電力モード</a>を追加。  <a href="#">セクション 25.4.15 : アドレス一致時に STOP モードからウェイクアップ</a>に注意を追加。  <a href="#">セクション 24.7 : I2C デバッグモード</a>を <a href="#">セクション 28.4.17</a>に移動し、<b>デバッグモード</b>に名称変更。  <a href="#">セクション 25.7.7 : 割り込みおよびステータスレジスタ (I2Cx_ISR)</a>の ARLO ビットの定義を変更。</p> <p><b>SPI/I2S :</b>  <a href="#">セクション 27.6 : SPI 割り込み</a>の CRC エラーを削除。</p> <p><b>USART :</b>  <a href="#">セクション 25.5.10: LIN (local interconnection network) モード (658 ページ)</a>を更新。  <a href="#">セクション 29.8.8 : 割り込みおよびステータスレジスタ (USART_ISR) (941 ページ)</a>のビット 19 (RWU) の注を削除。  <a href="#">表 99 : USART の機能</a>を更新して、USART5 の DMA を削除。</p>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2013 年 3 月 8 日	2 (続き)	<p><a href="#">セクション 25.7.7</a> : リクエストレジスタ (<code>USART_RQR</code>) で、ビット 2 MMRQ の「～は RWU フラグをリセットします」を「RWU フラグをセットします」に置換。</p> <p><a href="#">セクション 25.5.4</a> : ボーレート生成に「スマートカード、LIN、および IrDA モードでは、16 倍のオーバーサンプリングのみをサポート」を追加</p> <p><a href="#">図 198</a> : ワード長のプログラミングでストップビットを訂正して更新</p> <p><b>TSC :</b></p> <p><a href="#">セクション 28.6.2</a> : <b>TSC 割り込み有効レジスタ (<code>TSC_IER</code>)</b> の「パワーオンリセット値」を「リセット値」に置換。</p> <p><b>CAN</b></p> <p><a href="#">図 317</a> : ビットタイミングを更新</p> <p><b>DEBUG :</b></p> <p><a href="#">図 3</a> : <b>JTAG TAP 接続</b>を更新。</p>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2014 年 4 月 25 日	3	<p>「STM32F302xx、STM32F303xx、STM32F313xx」から「STM32F303xB/xC、STM32F303x4/x6/x8、STM32F328xx および STM32F358xx」にドキュメントの範囲を変更。</p> <p>マニュアルのセクションの順番を変更。</p> <p><b>セクション 1 :</b> Overview of the manual を追加。</p> <p><b>セクション 1 :</b> このマニュアルにおける表記の規則の「常に 0 として読み出されます (w_r0)」を削除</p> <p><b>CRC :</b></p> <p><b>セクション 5.2 : CRC の主な機能</b>の CRC v2 完全にプログラム可能な多項式の 1 つ目の反復を削除。</p> <p><b>内蔵 SRAM およびフラッシュメモリ :</b></p> <p><b>表 2 :</b> フラッシュメモリの読み出し保護ステータスを更新。</p> <p><b>セクション 3.3 : 内蔵 SRAM</b></p> <p><b>DAC :</b></p> <p>STM32F303x4/6/8 デバイスの DAC2 を追加。</p> <p><b>セクション 14.3.5 :</b> DAC 出力電圧の「VDDA」を「VREF+」に置換。</p> <p><b>COMP :</b></p> <p><b>セクション 15.5 :</b> COMP レジスタの COMPxOUTSEL ビットの説明に「(BRK)」と「(BRK2)」を追加。</p> <p><b>セクション 15.3.1 :</b> COMP ブロック図の図を更新。</p> <p><b>OPAMP :</b></p> <p><b>図 3 :</b> STM32F303x6/8 および STM32F328x8 コンパレータとオペアンプの接続を更新。</p> <p><b>DMA :</b></p> <p><b>表 24 :</b> STM32F303x6/8 および STM32F328x8 の DMA1 リクエストマッピングを追加。</p> <p><b>表 29 :</b> STM32F303x6/8 および STM32F328x8 でのチャンネルごとの DMA1 リクエストの概要を追加。</p> <p><b>システム設定コントローラ :</b></p> <p><b>セクション 1.1.1 :</b> SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1) の STM32F303x4/6/8 の専用ビットを追加。</p> <p><b>セクション 1.1.8 :</b> SYSCFG 設定レジスタ 3 (SYSCFG_CFGR3) を追加。</p> <p><b>ADC :</b></p> <p><b>セクション 1.2 :</b> ADC の主な機能を更新。</p> <p><b>セクション 1.3.6 :</b> ADC 電圧レギュレータ (ADVREGEN) を更新。</p> <p><b>セクション 1.3.3 :</b> クロックを更新。</p>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2014 年 4 月 25 日	3 (続き)	<p><a href="#">セクション 1.3.11</a> : チャネル選択 (SQRx、JSQRx) を更新。</p> <p><a href="#">セクション 1.3.31</a> : VBAT 電源監視の ADC1_IN18 を ADC1_IN17 に変更。ADCx_CR の DEEPPWD および ADVREGEN ビットを ADVREGEN[1:0] に置換。</p> <p><a href="#">セクション 1.6.2</a> : ADC 共通制御レジスタ (ADCx_CCR、x=12 または 34) を更新。</p> <p><b>割込み :</b></p> <p><a href="#">表 1</a> : STM32F302xB/C ベクタテーブルを追加。</p> <p><a href="#">セクション 1.3.6</a> : ペンディングレジスタ (<a href="#">EXTI_PR1</a>) および <a href="#">セクション 1.3.12</a> : ペンディングレジスタ (<a href="#">EXTI_PR2</a>) の「またはエッジ検出回路の感度を変更すると、」を削除し、<a href="#">セクション 1.3.5</a> : ソフトウェア割り込みイベントレジスタ (<a href="#">EXTI_SWIER1</a>) および <a href="#">セクション 1.3.11</a> : ソフトウェア割り込みイベントレジスタ (<a href="#">EXTI_SWIER2</a>) のビットの説明を更新。</p> <p><a href="#">セクション 1.2.1</a> : 主な特長のプログラミングマニュアルのリファレンセンテンスを更新</p> <p><b>PWR :</b></p> <p><a href="#">セクション 1.4.1</a> : 電源制御レジスタ (PWR_CR) のビット 0 「WUF」の説明を更新。</p> <p><a href="#">表 4</a> : STOP モードのモード開始の注を更新</p> <p><a href="#">図 2</a> : 電源の概要 (STM32F3x8 デバイス) を追加。</p> <p><b>RCC :</b></p> <p><a href="#">セクション 1.4.13</a> : クロック設定レジスタ 3 (RCC_CFGR3) の I2CxSW ビットの説明の「PCLK」を「SYSCLK」に置換。</p> <p>注 <sup>(2)</sup> を変更 (<a href="#">図 2</a> : STM32F303xB/C および STM32F358xC クロックツリー)。</p> <p><b>高性能制御タイマ (TIM1/8) :</b></p> <p><a href="#">セクション 1.3.15</a> : <a href="#">ブレーク機能の使用</a> を変更。</p> <p><b>IRTIM :</b></p> <p><a href="#">図 203</a> : TIM16 と TIM17 との IR 内部ハードウェア接続の TIM16 および TIM17 をスワップ。</p> <p><b>RTC :</b></p> <p><a href="#">図 207</a> : STM32F03x、STM32F04x、および STM32F05x デバイスの RTC ブロック図の WUT 入力クロックを更新。</p> <p><a href="#">セクション 24.7.4</a> : RTC 初期化とステータスレジスタ (RTC_ISR) のビット SHPF の読み出しパラメータとクリアパラメータを訂正。</p> <p><b>I2C :</b></p> <p>t<sub>HD;DAT</sub> を更新 (<a href="#">表 76</a> : I2C-SMBUS 仕様のデータのセットアップおよびホールド時間)。</p> <p><a href="#">I2C のタイミング</a> セクションで 50 ns を t<sub>AF(min)</sub> に、260 ns を t<sub>AF(max)</sub> に置換。</p> <p>すべてのレジスタのウェイトステート情報にアクセスの段落を追加。</p> <p><b>USART :</b></p> <p><a href="#">セクション 25.5.6</a> : 自動ポーレート検出のモード 2 および 3 を更新</p> <p><a href="#">セクション 25.7.7</a> : リクエストレジスタ (USART_RQR) の TXFRQ の説明を訂正。</p> <p>USARTDIV に関する <a href="#">注</a> : の変更 (<a href="#">セクション 26.5.4</a> : <a href="#">ポーレート生成</a>)</p>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2014 年 8 月 18 日	4	<p>システムおよびメモリの概要</p> <ul style="list-style-type: none"> <li>- セクション 3.1 : システムアーキテクチャ</li> </ul> <p>システム設定コントローラ (SYSCFG)</p> <ul style="list-style-type: none"> <li>- セクション 10.1.8 : SYSCFG 設定レジスタ 3 (SYSCFG_CFGR3)</li> </ul> <p>アナログデジタルコンバータ (ADC)</p> <ul style="list-style-type: none"> <li>- セクション 13.3.18 : 外部トリガおよびトリガ極性での変換 (EXTSEL, EXTEN, JEXTSEL, JEXTEN)</li> <li>- 図 30 : ADC1 および ADC2 接続性</li> <li>- 図 31 : ADC3 および ADC4 接続性</li> </ul> <p>D/A コンバータ (DAC)</p> <ul style="list-style-type: none"> <li>- 図 89 : DAC1 ブロック図</li> </ul> <p>オペアンプ</p> <ul style="list-style-type: none"> <li>- 図 104 : STM32F303xB/C および STM32F358xC コンパレータとオペアンプの相互接続 (パート 1)</li> <li>- 図 105 : STM32F303xB/C および STM32F358xC コンパレータとオペアンプの相互接続 (パート 2)</li> </ul> <p>コンパレータ</p> <ul style="list-style-type: none"> <li>- セクション 15.5.2 : COMP2 制御/ステータスレジスタ (COMP2_CSR)</li> </ul> <p>D/A コンバータ (DAC)</p> <ul style="list-style-type: none"> <li>- 図 89 : DAC1 ブロック図</li> </ul> <p>シリアルペリフェラルインタフェース/I2S (SPI/I2S)</p> <ul style="list-style-type: none"> <li>- 表 116 : STM32F303xB/C および STM32F358xC SPI 実装</li> </ul> <p>USART (Universal synchronous asynchronous receiver transmitter)</p> <ul style="list-style-type: none"> <li>- 機能リスト</li> <li>- 表 107 : STM32F3xx USART の機能</li> <li>- セクション 27.5.2 : トランスミッタ</li> <li>- セクション 27.7.1 : 制御レジスタ 1 (USARTx_CR1)</li> </ul>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2015 年 1 月 23 日	5	<p>適用範囲を STM32F303xD/E まで拡張。  <a href="#">セクション 8 : ペリフェラル相互接続マトリックス</a>を追加。                      以下を更新 :  <b>表紙</b>  <b>フレキシブルメモリコントローラ (FMC)</b>                      - 章の追加 (STM32F303xD/E のみ適用)。  <b>システムおよびメモリの概要</b>                      - <a href="#">セクション 3.1 : システムアーキテクチャ</a>  <b>内蔵フラッシュメモリ</b>                      - <a href="#">セクション 4.1 : フラッシュの主な機能</a>                      - <a href="#">セクション 4.2.1 : フラッシュメモリの構成</a>  <b>リセットおよびクロック制御 (RCC)</b>                      - <a href="#">セクション 9.2.10 : タイマ (TIMx) クロック</a>                      - <a href="#">セクション 9.4.2 : クロック設定レジスタ (RCC_CFGR)</a>  <b>システム設定コントローラ (SYSCFG)</b>                      - <a href="#">セクション 12.1.1 : SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1)</a>                      - <a href="#">セクション 12.1.2 : SYSCFG CCM SRAM 保護レジスタ (SYSCFG_RCR)</a>                      - <a href="#">セクション 12.1.8 : SYSCFG 設定レジスタ 3 (SYSCFG_CFGR3)</a>  <b>ダイレクトメモリアクセスコントローラ (DMA)</b>                      - <a href="#">表 79 : STM32F303xB/C/D/E, STM32F358xC, および STM32F398xE のチャンネルごとの DMA2 リクエストの概要</a>  <b>割り込みとイベント</b>                      - <a href="#">表 81 : STM32F303xB/C/D/E, STM32F358xC, および STM32F398xE ベクタテーブル</a>  <b>アナログデジタルコンバータ (ADC)</b>                      - <a href="#">表 : ADC1 (マスタ) &amp; 2 (スレーブ) - レギュラチャネルの外部トリガ</a>                      - <a href="#">表 : ADC1 &amp; ADC2 - インジェクトチャネルの外部トリガ</a>                      - <a href="#">表 : ADC3 &amp; ADC4 - レギュラチャネルの外部トリガ</a>                      - <a href="#">表 : ADC3 &amp; ADC4 - インジェクトチャネルの外部トリガ</a></p>



表 202. 文書改版履歴 (続き)

日付	版	変更内容
2015 年 1 月 23 日	5 (続き)	<p>コンパレータ (COMP)</p> <ul style="list-style-type: none"> <li>- セクション 17.2 : COMP の主な機能</li> <li>- 図 123 : STM32F303xB/C/D/E、STM32F358xC、および STM32F398xE コンパレータ 7 のブロック図</li> <li>- 表 107 : コンパレータの入出力の概要</li> </ul> <p>セクション 17.5 : COMP レジスタ</p> <p>高機能制御タイマ (TIM1/TIM8/TIM20)</p> <ul style="list-style-type: none"> <li>- STM32F303xD/E に TIM20 を導入</li> </ul> <p>I2C (Inter-integrated circuit) インタフェース</p> <ul style="list-style-type: none"> <li>- 表 141 : STM32F3xxI2C の実装</li> </ul> <p>USART (Universal synchronous asynchronous receiver transmitter)</p> <ul style="list-style-type: none"> <li>- 表 156 : STM32F3xx USART の機能</li> </ul> <p>シリアルペリフェラルインタフェース/I2S (SPI/I2S)</p> <ul style="list-style-type: none"> <li>- 表 166 : STM32F303xB/C/D/E、STM32F358xC および STM32F398xE SPI 実装</li> </ul> <p>ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)</p> <ul style="list-style-type: none"> <li>- 表 : STM32F3xx USB 実装</li> </ul>



表 202. 文書改版履歴 (続き)

日付	版	変更内容
2015 年 8 月 25 日	6	<p><b>リセットおよびクロック制御 (RCC)</b></p> <ul style="list-style-type: none"> <li>- セクション 9.4.13 : クロック設定レジスタ 3 (RCC_CFGR3) でこのドキュメントに含まれる STM32F3xx に UARTSW2[1:0] および UARTSW3[1:0] が使用可能であることを示す注を追加</li> <li>- セクション 9.4.10 : 制御/ステータスレジスタ (RCC_CSR) のビット 23 (V18PWRSTF) の名称および説明を更新</li> </ul> <p><b>フレキシブルスタティックメモリコントローラ (FSMC)</b></p> <ul style="list-style-type: none"> <li>- セクションを「スタティックメモリコントローラ」に名称変更</li> <li>- セクション : SRAM/NOR 型フラッシュチップセレクトタイミングレジスタ 1..4 (FMC_BTR1..4) のビット BUSTURN の説明を更新</li> <li>- 表 37 : プログラム可能な NOR/PSRAM のアクセスパラメータのメモリセットアップ時間とメモリホールドの最大値を更新</li> <li>- セクション : 共通メモリ空間タイミングレジスタ 2.4 (FMC_PMEM2..4) の FMC_PMEM レジスタの MEMSET、MEMHOLD および MEMHIZ ビットの説明を更新</li> <li>- セクション : 属性メモリ空間タイミングレジスタ 2.4 (FMC_PATT2..4) の FMC_PATT レジスタの ATTSET、ATTHOLD および ATTHIZ ビットの説明を更新</li> <li>- セクション : SRAM/NOR 型フラッシュチップセレクトタイミングレジスタ 1..4 (FMC_BTR1..4) の FMC_BTR1 ~ 4 レジスタの BURSTRUN ビットの説明を更新</li> </ul> <p><b>アナログデジタルコンバータ (ADC)</b></p> <ul style="list-style-type: none"> <li>- 表 84 : ADC 外部チャネルマッピングの STM32F303x6/8 および STM32F328 について、ADC1 および ADC2 チャネルの数を更新</li> </ul> <p><b>高機能制御タイマ (TIM1/TIM8/TIM20)</b></p> <ul style="list-style-type: none"> <li>- セクション 20.4.3 : TIM1/TIM8/TIM20 スレープモード制御レジスタ (TIMx_SMCR) のビット SMS の説明</li> </ul> <p><b>基本タイマ (TIM6/TIM7)</b></p> <ul style="list-style-type: none"> <li>- セクション 22.4.2 : TIM6/TIM7 制御レジスタ 2 (TIMx_CR2) のビット MMS の説明を更新</li> </ul> <p><b>汎用タイマ (TIM15/16/17)</b></p> <ul style="list-style-type: none"> <li>- セクション 23.6.6 : TIM16/TIM17 キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1) の IC1F[3:0] ビットの説明を更新</li> </ul> <p><b>USART (Universal synchronous asynchronous receiver transmitter)</b></p> <ul style="list-style-type: none"> <li>- 表 156 : STM32F3xx USART の機能の UART5 の DMA サポートを更新</li> </ul> <p><b>コントローラエリアネットワーク (bxCAN)</b></p> <ul style="list-style-type: none"> <li>- t<sub>CAN</sub> を t<sub>q</sub> に置換</li> </ul> <p><b>ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)</b></p> <ul style="list-style-type: none"> <li>- LPM レジスタの説明を追加</li> </ul>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2016 年 5 月 11 日	7	<p><b>I2C2 セクションの更新:</b></p> <ul style="list-style-type: none"> <li>- 図 294: セットアップおよびホールドタイミングを更新。</li> <li>- セクション 28.4.4: I2C の初期化を更新して注を追加し、セクション: I2C のタイミングを更新</li> <li>- セクション 28.7.5: タイミングレジスタ (I2C_TIMINGR) の SCLDEL[3:0] および SDADEL[3:0] ビットの説明を更新</li> <li>- 「STM32CubeMX ツールは、I2C 設定ウィンドウの I2C_TIMIGR コンテンツを計算し、提供します」の文章を追加してセクション 28.4.4: I2C の初期化、セクション 28.4.8: I2C マスタモードおよびセクション 28.7.5: タイミングレジスタ (I2C_TIMINGR) を更新</li> </ul> <p><b>タッチセンシングコントローラセクションの更新:</b></p> <ul style="list-style-type: none"> <li>- TSC 制御レジスタ設定の禁止に関する注を追加してセクション 19.3.4: 電荷移動取得シーケンスを更新</li> <li>- CTPL[3:0] ビットおよび PGPSC[2:0] ビットの注を追加してセクション 19.6.1: TSC 制御レジスタ (TSC_CR) を更新</li> </ul> <p><b>USART セクションの更新:</b></p> <ul style="list-style-type: none"> <li>- USART クロックソースが HSI クロックであるときにSTOP モードから正しくウェイクアップを可能にする最大 USART ポーレートを決定するの段落を追加してセクション 29.5.17: USART を使用した STOP モードからのウェイクアップを更新</li> <li>- nCTS を CTS に、nRTS を RTS に、SCLK を CK に置換して USART ドキュメント全体を更新</li> <li>- セクション 29.8.9: 割り込みフラグクリアレジスタ (USART_ICR) の「w」を「rc_wl」に置換して更新</li> <li>- セクション 29.8.8: 割り込みおよびステータスレジスタ (USART_ISR) の RTOF フィールドの USARTx_CR2 を USARTx_CR1 に置換して更新</li> <li>- 「ONEBIT」ビット 11 の説明に注を追加してセクション 29.8.3: 制御レジスタ 3 (USART_CR3) を更新</li> <li>- USART_regname のレジスタ名 USARTx_regname を変更してセクション 29: USART (Universal synchronous asynchronous receiver transmitter) を更新</li> </ul> <p><b>RTC セクションの更新:</b></p> <ul style="list-style-type: none"> <li>- 図 291: RTC ブロック図の WUCKSEL ビットを更新</li> <li>- セクション 27.3.9: RTC のリセットの LSE によってクロック供給される RTC のケースを追加</li> <li>- 注を追加して図 291: RTC ブロック図を更新</li> <li>- TAMP3 ビットに注、レジスタに TAMP2E および TAMP2TRG を黒色で追加してセクション 27.6.16: RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR) を更新</li> </ul>



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

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2016年5月11日	7 (続き)	<p><b>RCC セクションの更新:</b></p> <ul style="list-style-type: none"> <li>- セクション 9.4.9: RTC ドメイン制御レジスタ (RCC_BDCR) を更新 (LSEDRV[1:0] ビット 「01」と「10」の組み合わせをスワップ)</li> <li>- メインクロック出力の生成に LSE 入力を追加して <a href="#">図 13</a>: STM32F303xB/C および STM32F358xC クロックツリー、<a href="#">図 14</a>: STM32F303xDxE および STM32F398xE のクロックツリーおよび <a href="#">図 15</a>: STM32F303x6/8 および STM32F328x8 クロックツリーを更新</li> <li>- RTC クロックが LSE の場合に「RTC は、システムリセット中でもクロック供給された状態、かつ機能的な状態のままです。」を追加して <a href="#">セクション 9.2.9: RTC クロック</a> を更新</li> <li>- PLLCLK がタイマクロックソースである場合に x2 係数を TIM2/3/4 に追加して <a href="#">図 14</a>: STM32F303xDxE および STM32F398xE のクロックツリーを更新</li> <li>- 「USARTx (x=1、2、3)」を「USART1」に置換して <a href="#">図 15</a>: STM32F303x6/8 および STM32F328x8 クロックツリーを更新</li> <li>- USBPRES を USBPRE に名称変更し、ビット 22 USBPRE の説明を追加して <a href="#">セクション 9.4.2: クロック設定レジスタ (RCC_CFGR)</a> を更新</li> </ul> <p><b>TIMER セクションの更新:</b></p> <ul style="list-style-type: none"> <li>- 「IC2S=01」を「CC2S=01」に変更して <a href="#">セクション 21.3.13: ワンパルスモード</a> を更新</li> <li>- タイトルに (TIM15 のみ) を追加して <a href="#">セクション 23.4.18: スレープモード: リセットモードとトリガモードの組み合わせ (TIM15 のみ)</a> を更新。</li> <li>- ビット 7 「予約済み」を OC1CE に置換して <a href="#">セクション 23.5.7: TIM15 のキャプチャ/比較モードレジスタ 1 (TIM15_CCMR1)</a> および <a href="#">セクション 23.5.18: TIM15 レジスタマップ</a> を更新</li> <li>- ビット 7 「予約済み」を OC1CE に置換して <a href="#">セクション 23.6.6: TIM16/TIM17 キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)</a> および <a href="#">セクション 23.6.17: TIM16/TIM17 レジスタマップ</a> を更新</li> <li>- <a href="#">図 20: 高機能制御タイマ (TIM1/TIM8/TIM20)</a>、<a href="#">セクション 21: 汎用タイマ (TIM2/TIM3/TIM4)</a>、<a href="#">セクション 22: 基本タイマ (TIM6/TIM7)</a>: および <a href="#">セクション 23: 汎用タイマ (TIM15/TIM16/TIM17)</a> の PSC[15:0] ビットの説明を更新</li> <li>- <a href="#">セクション 20.4.5: TIM1/TIM8/TIM20 ステータスレジスタ (TIMx_SR)</a> および <a href="#">セクション 20.4.25: TIM1/TIM8/TIM20 レジスタマップ</a> の CC5IF および CC6IF のビット名を更新</li> </ul> <p><b>FMC セクションの更新:</b></p> <ul style="list-style-type: none"> <li>- 「NOE 信号をネゲートする」を載せて <a href="#">セクション 10.5.4: NOR フラッシュ/PSRAM コントローラ非同期トランザクション</a> を更新</li> <li>- 注を追加して <a href="#">セクション: FIFO ステータスおよび割り込みレジスタ 2.4 (FMC_SR2.4)</a> のビット 0 (IRS) およびビット 2 (IFS) を更新</li> <li>- 新しい段落を追加して <a href="#">セクション: SRAM/NOR 型フラッシュチップセレクトタイミングレジスタ 1.4 (FMC_BTR1.4)</a> を更新</li> <li>- 新しい段落を追加して <a href="#">セクション: SRAM/NOR 型フラッシュ書き込みタイミングレジスタ 1.4 (FMC_BWTR1.4)</a> を更新</li> <li>- 「MEMxHIZ」を「MEMxHIZ+1」に置換し、注 2 を追加して <a href="#">図 38: 共通メモリアクセスの NAND フラッシュ/PC カードコントローラ波形</a> を更新</li> </ul>

表 202. 文書改版履歴 (続き)

日付	版	変更内容
2016 年 5 月 11 日	7 (続き)	<ul style="list-style-type: none"> <li>- セクション 10.6.5 : NAND フラッシュのプリウェイト機能を更新。</li> <li>- 共通メモリ空間タイミングレジスタ 2.4 (FMC_PMEM2.4) の MEMHOLD[7:0] の説明を更新</li> <li>- 属性メモリ空間タイミングレジスタ 2.4 (FMC_PATT2.4) の ATTHOLD[7:0] の説明を更新</li> <li>- セクション 10.3 : AHB インタフェースを更新。</li> <li>- NWE 立ち下がりエッジに訂正して 図 32: マルチプレクス書き込みアクセス波形を更新</li> </ul> <p><b>内蔵フラッシュメモリの更新 :</b></p> <ul style="list-style-type: none"> <li>- SYSCLK を HCLK に置換して セクション 4.5.1: フラッシュアクセス制御レジスタ (FLASH_ACR) ビット LATENCY[2:0] を更新</li> </ul> <p><b>ADC セクションの更新 :</b></p> <ul style="list-style-type: none"> <li>- セクション 15.3.3 : クロックの注で、オプション a) を「オプション b) に置換し、「または 10」を削除して更新</li> </ul> <p><b>オペアンプセクション (OPAMP) :</b></p> <ul style="list-style-type: none"> <li>- 表 109 : STM32F303xB/C/D/E, STM32F358xC, および STM32F398xE の専用 I/O との接続を更新。</li> </ul> <p><b>コンパレータセクションの更新 :</b></p> <ul style="list-style-type: none"> <li>- 注 4 を追加して 図 122 : コンパレータ 1 および 2 のブロック図 (STM32F303xB/C/D/E, STM32F358xC, および STM32F398xE) を更新</li> <li>- 注 3 およびビット 9 の説明を変更して セクション 17.5.2: COMP2 制御およびステータスレジスタ (COMP2_CSR)、セクション 17.5.4: COMP4 制御およびステータスレジスタ (COMP4_CSR) および セクション 17.5.6: COMP6 制御およびステータスレジスタ (COMP6_CSR) を更新</li> </ul> <p><b>割り込みとイベントセクションの更新 :</b></p> <ul style="list-style-type: none"> <li>- ライン 26/28/29/31/33 を更新および EXTI ラインの注を追加して セクション 14.2.6 : 外部および内部の割り込み/イベントラインの配置を更新</li> </ul> <p><b>USB セクションの更新 :</b></p> <ul style="list-style-type: none"> <li>- 「および STM32F358xC」を削除して 表 174 : STM32F3xx USB の実装を更新</li> </ul> <p><b>DEBUG セクションの更新 :</b></p> <ul style="list-style-type: none"> <li>- セクション 33.6.1: MCU デバイス ID コードの DBGMCU_IDCODE の説明を更新。</li> </ul>

## 索引

### A

ADCx_AWD2CR	398
ADCx_AWD3CR	399
ADCx_CALFACT	400
ADCx_CCR	403
ADCx_CDR	406
ADCx_CFGR	380
ADCx_CR	377
ADCx_CSR	401
ADCx_DIFSEL	399
ADCx_DR	394
ADCx_IER	375
ADCx_ISR	373
ADCx_JDRy	398
ADCx_JSQR	395
ADCx_OFRy	397
ADCx_SMPR1	384
ADCx_SMPR2	386
ADCx_SQR1	389
ADCx_SQR2	390
ADCx_SQR3	392
ADCx_SQR4	393
ADCx_TR1	386
ADCx_TR2	387
ADCx_TR3	388

### C

CAN_BTR	1038
CAN_ESR	1037
CAN_FA1R	1048
CAN_FFA1R	1047
CAN_FiRx	1049
CAN_FM1R	1046
CAN_FMR	1046
CAN_FS1R	1047
CAN_IER	1035
CAN_MCR	1029
CAN_MSR	1030
CAN_RDHxR	1045
CAN_RDLxR	1044
CAN_RDTxR	1044
CAN_RF0R	1034
CAN_RF1R	1035
CAN_RlRx	1043
CAN_TDHxR	1042
CAN_TDLxR	1042
CAN_TDTxR	1041

CAN_TlRx	1040
CAN_TSR	1032
COMP1_CSR	443
COMP2_CSR	445
COMP3_CSR	447
COMP4_CSR	450
COMP5_CSR	452
COMP6_CSR	455
COMP7_CSR	457
CRC_CR	89
CRC_DR	88
CRC_IDR	89
CRC_INIT	90
CRC_POL	90

### D

DAC_CR	424
DAC_DHR12L1	429
DAC_DHR12L2	430
DAC_DHR12LD	431
DAC_DHR12R1	428
DAC_DHR12R2	429
DAC_DHR12RD	431
DAC_DHR8R1	429
DAC_DHR8R2	430
DAC_DHR8RD	432
DAC_DOR1	432
DAC_DOR2	432
DAC_SR	433
DAC_SWTRIGR	428
DBGMCU_APB1_FZ	1109
DBGMCU_APB2_FZ	1111
DBGMCU_CR	1107
DBGMCU_IDCODE	1094
DMA_CCRx	276
DMA_CMARx	279
DMA_CNDTRx	278
DMA_CPARx	278
DMA_IFCR	275
DMA_ISR	274

### E

EXTI_EMR	295, 298
EXTI_FTSR	296, 299
EXTI_IMR	295, 298
EXTI_PR	297, 300
EXTI_RTSTR	296, 299



EXTI_SWIER .....	297, 300	OPAMP4_CSR .....	479
<b>F</b>		<b>P</b>	
FLASH_ACR .....	76	PWR_CR .....	105
FLASH_CR .....	78	PWR_CSR .....	106
FLASH_KEYR .....	76		
FLASH_OPTKEYR .....	77	<b>R</b>	
FLASH_SR .....	77	RCC_AHBENR .....	147
FMPI2C_ISR .....	876	RCC_AHBSTR .....	157
FSMC_BCR1..4 .....	201	RCC_APB1ENR .....	151
FSMC_BTR1..4 .....	203	RCC_APB1RSTR .....	145
FSMC_BWTR1..4 .....	207	RCC_APB2ENR .....	149
		RCC_APB2RSTR .....	143
<b>G</b>		RCC_BDCR .....	154
GPIOx_AFRH .....	240	RCC_CFGR .....	137
GPIOx_AFRL .....	239	RCC_CFGR2 .....	158
GPIOx_BRR .....	240	RCC_CFGR3 .....	161
GPIOx_BSRR .....	238	RCC_CIR .....	141
GPIOx_IDR .....	237	RCC_CR .....	135
GPIOx_LCKR .....	238	RCC_CSR .....	155
GPIOx_MODER .....	235	RTC_ALRMAR .....	799
GPIOx_ODR .....	237	RTC_ALRMBR .....	800
GPIOx_OSPEEDR .....	236	RTC_ALRMBSSR .....	811
GPIOx_OTYPER .....	235	RTC_BKPxR .....	811
GPIOx_PUPDR .....	236	RTC_CALR .....	806
		RTC_CR .....	791
<b>I</b>		RTC_DR .....	790
I2C_CR1 .....	866	RTC_ISR .....	794
I2C_CR2 .....	869	RTC_PRER .....	797
I2C_ICR .....	878	RTC_SHIFTR .....	802
I2C_ISR .....	876	RTC_SSR .....	801
I2C_OAR1 .....	872	RTC_TAFCR .....	807
I2C_OAR2 .....	873	RTC_TR .....	789
I2C_PECR .....	879	RTC_TSDR .....	804
I2C_RXDR .....	880	RTC_TSSSR .....	805
I2C_TIMEOUTR .....	875	RTC_TSTR .....	803
I2C_TIMINGR .....	874	RTC_WPR .....	801
I2C_TXDR .....	880	RTC_WUTR .....	798
I2Cx_CR2 .....	869		
IWDG_KR .....	759	<b>S</b>	
IWDG_PR .....	760	SPIx_CR1 .....	996
IWDG_RLR .....	761	SPIx_CR2 .....	998
IWDG_SR .....	762	SPIx_CRCPR .....	1003
IWDG_WINR .....	763	SPIx_DR .....	1002
		SPIx_I2SCFGR .....	1005
<b>O</b>		SPIx_I2SPR .....	1007
OPAMP1_CSR .....	472	SPIx_RXCR .....	1004
OPAMP2_CSR .....	474	SPIx_SR .....	1001
OPAMP3_CSR .....	476	SPIx_TXCR .....	1004
		SYSCFG_EXTICR1 .....	247

SYSCFG\_EXTICR2 ..... 248  
 SYSCFG\_EXTICR3 ..... 250  
 SYSCFG\_EXTICR4 ..... 252  
 SYSCFG\_MEMRMP ..... 243, 255-256

**T**

TIM15\_ARR ..... 729  
 TIM15\_BDTR ..... 731  
 TIM15\_CCER ..... 726  
 TIM15\_CCMR1 ..... 722  
 TIM15\_CCR1 ..... 730  
 TIM15\_CCR2 ..... 731  
 TIM15\_CNT ..... 729  
 TIM15\_CR1 ..... 714  
 TIM15\_CR2 ..... 715  
 TIM15\_DCR ..... 733  
 TIM15\_DIER ..... 718  
 TIM15\_DMAR ..... 734  
 TIM15\_EGR ..... 721  
 TIM15\_PSC ..... 729  
 TIM15\_RCR ..... 730  
 TIM15\_SMCR ..... 717  
 TIM15\_SR ..... 719  
 TIM16\_OR ..... 752  
 TIMx\_ARR ..... 582, 660, 677, 747  
 TIMx\_BDTR ..... 585, 749  
 TIMx\_CCER ..... 578, 657, 744  
 TIMx\_CCMR1 ..... 572, 652, 741  
 TIMx\_CCMR2 ..... 576, 656  
 TIMx\_CCMR3 ..... 591  
 TIMx\_CCR1 ..... 583, 661, 748  
 TIMx\_CCR2 ..... 583, 661  
 TIMx\_CCR3 ..... 584, 662  
 TIMx\_CCR4 ..... 584, 662  
 TIMx\_CCR5 ..... 592  
 TIMx\_CCR6 ..... 593  
 TIMx\_CNT ..... 582, 659, 676, 747  
 TIMx\_CR1 ..... 561, 642, 673, 736  
 TIMx\_CR2 ..... 562, 644, 675, 737  
 TIMx\_DCR ..... 588, 663, 751  
 TIMx\_DIER ..... 567, 648, 675, 738  
 TIMx\_DMAR ..... 589, 663, 751  
 TIMx\_EGR ..... 571, 651, 676, 740  
 TIMx\_OR ..... 590  
 TIMx\_PSC ..... 582, 660, 677, 747  
 TIMx\_RCR ..... 583, 748  
 TIMx\_SMCR ..... 565, 645  
 TIMx\_SR ..... 569, 649, 676, 739  
 TSC\_CR ..... 492  
 TSC\_ICR ..... 495  
 TSC\_IER ..... 494

TSC\_IOASCR ..... 497  
 TSC\_IOCCR ..... 498  
 TSC\_IOGCSR ..... 498  
 TSC\_IOGxCR ..... 499  
 TSC\_IOHCR ..... 496  
 TSC\_IOSCR ..... 497  
 TSC\_ISR ..... 496

**U**

USART\_BRR ..... 938  
 USART\_CR1 ..... 927  
 USART\_CR2 ..... 930  
 USART\_CR3 ..... 934  
 USART\_GTPR ..... 938  
 USART\_ICR ..... 946  
 USART\_ISR ..... 941  
 USART\_RDR ..... 947  
 USART\_RQR ..... 940  
 USART\_RTOR ..... 939  
 USART\_TDR ..... 947  
 USB\_ADDRn\_RX ..... 1082  
 USB\_ADDRn\_TX ..... 1081  
 USB\_BTABLE ..... 1075  
 USB\_CNTR ..... 1069  
 USB\_COUNTn\_RX ..... 1083  
 USB\_COUNTn\_TX ..... 1082  
 USB\_DADDR ..... 1075  
 USB\_EPnR ..... 1077  
 USB\_FNR ..... 1074  
 USB\_ISTR ..... 1071  
 USB\_LPMCSR ..... 1076

**W**

WWDG\_CFR ..... 770  
 WWDG\_CR ..... 769  
 WWDG\_SR ..... 770



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



## 重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST製品は、注文請書発行時点で有効なSTの販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してSTは一切の責任を負いません。

明示又は黙示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

STおよびSTロゴはSTMicroelectronicsの商標です。その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

© 2017 STMicroelectronics - All rights reserved

