



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。STM32F401xB/C および STM32F401xD/E のマイクロコントローラのメモリやペリフェラルを使用する方法について、詳しく説明しています。

STM32F401xB/C および STM32F401xD/E 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の STM32F401xx ファミリ製品になります。

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

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

関連ドキュメント

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

- STM32F401xB/C データシート
- STM32F401xD/E データシート
- FPU 搭載 ARM®-M4 コアについては、FPU 搭載 Cortex®-M4 STM32F3xx/F4xxx Cortex®-M4 プログラミングマニュアル (PM0214) を参照してください。

目次

1	このマニュアルにおける表記の規則	34
1.1	レジスタに関する略記	34
1.2	用語	35
1.3	使用可能なペリフェラル	35
2	メモリとバスのアーキテクチャ	36
2.1	システムアーキテクチャ	36
2.1.1	I-バス	37
2.1.2	D-バス	37
2.1.3	S-バス	37
2.1.4	DMA メモリバス	37
2.1.5	DMA ペリフェラルバス	37
2.1.6	バスマトリックス	37
2.1.7	AHB-APB ブリッジ (APB)	37
2.2	メモリ構成	38
2.3	メモリマップ	38
2.3.1	内蔵 SRAM	40
2.3.2	フラッシュメモリの概要	40
2.3.3	ビットバンディング	40
2.4	ブート設定	41
3	内蔵フラッシュメモリインタフェース	44
3.1	概要	44
3.2	主な特長	44
3.3	内蔵フラッシュメモリ割込み (STM32F401xB/C および STM32F401xD/E)	45
3.4	読出しインタフェース	46
3.4.1	CPU クロック周波数とフラッシュメモリ読出し時間との関係	46
3.4.2	適応型リアルタイムメモリアクセラレータ (ART Accelerator™)	47
3.5	消去操作とプログラム操作	49
3.5.1	フラッシュ制御レジスタのアンロック	49
3.5.2	プログラム／消去の並列処理	49
3.5.3	消去	50

3.5.4	プログラミング	51
3.5.5	割込み	52
3.6	オプションバイト	52
3.6.1	ユーザオプションバイトの説明	52
3.6.2	ユーザオプションバイトのプログラミング	54
3.6.3	読出し保護 (RDP)	54
3.6.4	書込み保護	56
3.6.5	独自仕様コード読出し保護 (PCROP)	57
3.7	OTP (One-time programmable) バイト	59
3.8	フラッシュインタフェースレジスタ	60
3.8.1	フラッシュアクセス制御レジスタ (FLASH_ACR)	60
3.8.2	フラッシュキーレジスタ (FLASH_KEYR)	61
3.8.3	フラッシュオプションキーレジスタ (FLASH_OPTKEYR)	61
3.8.4	フラッシュステータスレジスタ (FLASH_SR)	62
3.8.5	フラッシュ制御レジスタ (FLASH_CR)	63
3.8.6	フラッシュオプション制御レジスタ (FLASH_OPTCR)	65
3.8.7	フラッシュインタフェースレジスタマップ	67
4	CRC 計算ユニット	68
4.1	CRC の概要	68
4.2	CRC の主な機能	68
4.3	CRC の機能説明	68
4.4	CRC レジスタ	69
4.4.1	データレジスタ (CRC_DR)	69
4.4.2	独立型データレジスタ (CRC_IDR)	69
4.4.3	制御レジスタ (CRC_CR)	70
4.4.4	CRC レジスタマップ	70
5	電源コントローラ (PWR)	71
5.1	電源	71
5.1.1	独立した A/D コンバータ用電源と基準電圧	72
5.1.2	バッテリーバックアップドメイン	72
5.1.3	電圧レギュレータ	73
5.2	電源供給スーパバイザ	74
5.2.1	パワーオンリセット (POR) / パワーダウンリセット (PDR)	74
5.2.2	ブラウンアウトリセット (BOR)	75

5.2.3	プログラム可能な電圧検出器 (PVD)	76
5.3	低電力モード	76
5.3.1	システムクロックの低速化	78
5.3.2	ペリフェラルクロックゲーティング	78
5.3.3	SLEEP モード	79
5.3.4	STOP モード	80
5.3.5	STANDBY モード	82
5.3.6	デバイスをウェイクアップさせるための RTC 代替機能の プログラミング (STOP および STANDBY モードから)	84
5.4	電源制御レジスタ	87
5.4.1	PWR 電源制御レジスタ (PWR_CR)	87
5.4.2	PWR 電源制御/ステータスレジスタ (PWR_PWR)	89
5.5	PWR レジスタマップ	90
6	リセットおよびクロック制御 (STM32F401xB/C および STM32F401xD/E 向け) (RCC)	91
6.1	リセット	91
6.1.1	システムリセット	91
6.1.2	電源リセット	92
6.1.3	バックアップドメインリセット	93
6.2	クロック	93
6.2.1	HSE クロック	96
6.2.2	HSI クロック	97
6.2.3	PLL の設定	97
6.2.4	LSE クロック	98
6.2.5	LSI クロック	98
6.2.6	システムクロック (SYSCLK) の選択	98
6.2.7	クロックセキュリティシステム (CSS)	99
6.2.8	RTC/AWU クロック	99
6.2.9	ウォッチドッグクロック	100
6.2.10	クロック信号出力	100
6.2.11	TIM5/TIM11 を使用した内部/外部クロックの測定	100
6.3	RCC レジスタ	103
6.3.1	RCC クロック制御レジスタ (RCC_CR)	103
6.3.2	RCC PLL 設定レジスタ (RCC_PLLCFGR)	105
6.3.3	RCC クロック設定レジスタ (RCC_CFGR)	107
6.3.4	RCC クロック割込みレジスタ (RCC_CIR)	110

6.3.5	RCC AHB1 ペリフェラルリセットレジスタ (RCC_AHB1RSTR) ...	112
6.3.6	RCC AHB2 ペリフェラルリセットレジスタ (RCC_AHB2RSTR) ...	114
6.3.7	RCC APB1 ペリフェラルリセットレジスタ (RCC_APB1RSTR) ...	114
6.3.8	RCC APB2 ペリフェラルリセットレジスタ (RCC_APB2RSTR) ...	116
6.3.9	RCC AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1ENR)	118
6.3.10	RCC AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2ENR)	119
6.3.11	RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR)	119
6.3.12	RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2ENR)	122
6.3.13	低電力モードにおける RCC AHB1 ペリフェラルクロック 有効レジスタ (RCC_AHB1LPENR)	124
6.3.14	低電力モードにおける RCC AHB2 ペリフェラルクロック 有効レジスタ (RCC_AHB2LPENR)	125
6.3.15	低電力モードにおける RCC APB1 ペリフェラルクロック 有効レジスタ (RCC_APB1LPENR)	126
6.3.16	低電力モードにおける RCC APB2 ペリフェラルクロック 有効レジスタ (RCC_APB2LPENR)	128
6.3.17	RCC バックアップドメイン制御レジスタ (RCC_BDCR)	130
6.3.18	RCC クロック制御およびステータスレジスタ (RCC_CSR)	131
6.3.19	RCC スペクトル拡散クロック生成レジスタ (RCC_SSCGR)	133
6.3.20	RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR)	134
6.3.21	RCC 専用クロック設定レジスタ (RCC_DCKCFGR)	136
6.3.22	RCC レジスタマップ	137
7	システム設定コントローラ (SYSCFG)	139
7.1	I/O 補正セル	139
7.2	SYSCFG レジスタ	139
7.2.1	SYSCFG メモリ再配置レジスタ (SYSCFG_MEMRMP)	139
7.2.2	SYSCFG ペリフェラルモード設定レジスタ (SYSCFG_PMC)	140
7.2.3	SYSCFG 外部割込み設定レジスタ 1 (SYSCFG_EXTICR1)	141
7.2.4	SYSCFG 外部割込み設定レジスタ 2 (SYSCFG_EXTICR2)	141
7.2.5	SYSCFG 外部割込み設定レジスタ 3 (SYSCFG_EXTICR3)	142
7.2.6	SYSCFG 外部割込み設定レジスタ 4 (SYSCFG_EXTICR4)	142

7.2.7	補正セル制御レジスタ (SYSCFG_CMPCR)	143
7.2.8	SYSCFG レジスタマップ	144
8	汎用 I/O (GPIO)	145
8.1	GPIO の概要	145
8.2	GPIO の主な機能	145
8.3	GPIO の機能説明	146
8.3.1	汎用 I/O (GPIO)	148
8.3.2	I/O ピンマルチプレクサとマッピング	148
8.3.3	I/O ポート制御レジスタ	150
8.3.4	I/O ポートデータレジスタ	151
8.3.5	I/O データのビット単位の操作	151
8.3.6	GPIO ロック機構	151
8.3.7	I/O オルタネート機能の入力／出力	152
8.3.8	外部割込み／ウェイクアップライン	152
8.3.9	入力設定	152
8.3.10	出力設定	153
8.3.11	オルタネート機能設定	153
8.3.12	アナログ設定	154
8.3.13	OSC32_IN/OSC32_OUT ピンの GPIO PC14/PC15 ポートピンとしての使用	155
8.3.14	OSC_IN/OSC_OUT ピンの GPIO PH0/PH1 ポートピンとしての 使用	155
8.3.15	RTC 機能の選択	155
8.4	GPIO レジスタ	157
8.4.1	GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ E、H)	157
8.4.2	GPIO ポート出カタイプレジスタ (GPIOx_OTYPER) (x = A ~ E、H)	157
8.4.3	GPIO ポート出カスピードレジスタ (GPIOx_OSPEEDR) (x = A ~ E、H)	158
8.4.4	GPIO ポートプルアップ／プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ E、H)	158
8.4.5	GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ E、H)	159
8.4.6	GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A ~ E、H)	159
8.4.7	GPIO ポートビットセット／リセットレジスタ (GPIOx_BSRR) (x = A ~ E、H)	160

8.4.8	GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ E、H)	160
8.4.9	GPIO オルタネート機能下位レジスタ (GPIOx_AFRL) (x = A ~ E、H)	161
8.4.10	GPIO オルタネート機能上位レジスタ (GPIOx_AFRH) (x = A ~ E、H)	162
8.4.11	GPIO レジスタマップ	162
9	DMA コントローラ (DMA)	164
9.1	DMA の概要	164
9.2	DMA の主な機能	164
9.3	DMA の機能説明	166
9.3.1	概要	166
9.3.2	DMA トランザクション	167
9.3.3	チャンネル選択	168
9.3.4	アービタ	169
9.3.5	DMA ストリーム	169
9.3.6	転送元、転送先、および転送モード	170
9.3.7	ポインタのインクリメント	173
9.3.8	サーキュラモード	174
9.3.9	ダブルバッファモード	174
9.3.10	プログラム可能なデータ幅、パッキング／アンパッキング、 エンディアン形式	175
9.3.11	シングル転送とバースト転送	177
9.3.12	FIFO	178
9.3.13	DMA 転送の完了	181
9.3.14	DMA 転送の中断	181
9.3.15	フローコントローラ	182
9.3.16	実現可能な DMA 設定の概要	183
9.3.17	ストリーム設定手順	183
9.3.18	エラー管理	184
9.4	DMA 割込み	185
9.5	DMA レジスタ	186
9.5.1	DMA ロー割込みステータスレジスタ (DMA_LISR)	186
9.5.2	DMA ハイ割込みステータスレジスタ (DMA_HISR)	187
9.5.3	DMA ロー割込みフラグクリアレジスタ (DMA_LIFCR)	188
9.5.4	DMA ハイ割込みフラグクリアレジスタ (DMA_HIFCR)	188
9.5.5	DMA ストリーム x 設定レジスタ (DMA_SxCR) (x=0 ~ 7)	189

9.5.6	DMA ストリーム x データ数レジスタ (DMA_HIFCR) (x=0 ~ 7)	192
9.5.7	DMA ストリーム x ペリフェラルアドレスレジスタ (DMA_SxPAR) (x=0 ~ 7)	193
9.5.8	DMA ストリーム x メモリ 0 アドレスレジスタ (DMA_SxM0AR) (x=0 ~ 7)	193
9.5.9	DMA ストリーム x メモリ 1 アドレスレジスタ (DMA_SxM1AR) (x=0 ~ 7)	194
9.5.10	DMA ストリーム x FIFO 制御レジスタ (DMA_SxFCR) (x=0 ~ 7)	194
9.5.11	DMA レジスタマップ	196
10	割込みとイベント	200
10.1	ネスト化されたベクタ割込みコントローラ (NVIC)	200
10.1.1	NVIC の機能	200
10.1.2	SysTick 較正值レジスタ	200
10.1.3	割込みベクタと例外ベクタ	200
10.2	外部割込み／イベントコントローラ (EXTI)	200
10.2.1	EXTI の主な機能	203
10.2.2	EXTI ブロック図	203
10.2.3	ウェイクアップイベント管理	204
10.2.4	機能詳細	204
10.2.5	外部割込み／イベントラインの配置	205
10.3	EXTI レジスタ	206
10.3.1	割込みマスクレジスタ (EXTI_IMR)	206
10.3.2	イベントマスクレジスタ (EXTI_EMR)	206
10.3.3	立ち上がりトリガ選択レジスタ (EXTI_RTSTR)	207
10.3.4	立ち下がりトリガ選択レジスタ (EXTI_FTSTR)	207
10.3.5	ソフトウェア割込みイベントレジスタ (EXTI_SWIER)	208
10.3.6	ペンディングレジスタ (EXTI_PR)	208
10.3.7	EXTI レジスタマップ	209
11	アナログデジタルコンバータ (ADC)	210
11.1	ADC の概要	210
11.2	ADC の主な機能	210
11.3	ADC の機能詳細	211
11.3.1	ADC のオン／オフ制御	212
11.3.2	ADC クロック	212

11.3.3	チャンネル選択	212
11.3.4	シングル変換モード	213
11.3.5	連続変換モード	213
11.3.6	タイミング図	214
11.3.7	アナログウォッチドッグ	214
11.3.8	スキャンモード	215
11.3.9	インジェクトチャンネルの管理	215
11.3.10	不連続モード	216
11.4	データの配置	218
11.5	チャンネル単位でプログラム可能なサンプリング時間	219
11.6	外部トリガによる変換およびトリガ極性	219
11.7	高速変換モード	221
11.8	データ管理	222
11.8.1	DMA の使用	222
11.8.2	DMA を使用しない変換シーケンスの管理	222
11.8.3	DMA およびオーバーラン検出を使用しない変換	223
11.9	温度センサ	223
11.10	バッテリー残量監視	224
11.11	ADC 割込み	225
11.12	ADC レジスタ	226
11.12.1	ADC ステータスレジスタ (ADC_SR)	226
11.12.2	ADC 制御レジスタ 1 (ADC_CR1)	227
11.12.3	ADC 制御レジスタ 2 (ADC_CR2)	229
11.12.4	ADC サンプル時間レジスタ 1 (ADC_SMPR1)	231
11.12.5	ADC サンプル時間レジスタ 2 (ADC_SMPR2)	232
11.12.6	ADC インジェクトチャンネルデータオフセットレジスタ x (ADC_JOFRx) (x=1..4)	232
11.12.7	ADC ウォッチドッグ高閾値レジスタ (ADC_HTR)	233
11.12.8	ADC ウォッチドッグ低閾値レジスタ (ADC_LTR)	233
11.12.9	ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)	234
11.12.10	ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)	234
11.12.11	ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)	235
11.12.12	ADC インジェクトシーケンスレジスタ (ADC_JSQR)	236
11.12.13	ADC インジェクトデータレジスタ x (ADC_JDRx) (x= 1..4)	237
11.12.14	ADC レギュラデータレジスタ (ADC_DR)	237
11.12.15	ADC 共通制御レジスタ (ADC_CCR)	238

11.12.16	ADC レジスタマップ	239
12	高機能制御タイマ (TIM1)	241
12.1	TIM1 の概要	241
12.2	TIM1 の主な機能	242
12.3	TIM1 の機能詳細	244
12.3.1	タイムベースユニット	244
12.3.2	カウンタモード	245
12.3.3	繰り返しカウンタ	253
12.3.4	クロック選択	255
12.3.5	キャプチャ／比較チャンネル	258
12.3.6	入力キャプチャモード	260
12.3.7	PWM 入力モード	261
12.3.8	強制出力モード	262
12.3.9	出力比較モード	262
12.3.10	PWM モード	263
12.3.11	相補出力とデッドタイム挿入	266
12.3.12	ブレーク機能の使用	268
12.3.13	外部イベントによる OCxREF 信号のクリア	271
12.3.14	6 ステップ PWM 生成	272
12.3.15	ワンパルスモード	273
12.3.16	エンコーダインタフェースモード	274
12.3.17	タイマ入力 XOR 機能	276
12.3.18	ホールセンサとのインタフェース	277
12.3.19	TIMx と外部トリガの同期	279
12.3.20	タイマの同期	282
12.3.21	デバッグモード	282
12.4	TIM1 レジスタ	283
12.4.1	TIM1 制御レジスタ 1 (TIMx_CR1)	283
12.4.2	TIM1 制御レジスタ 2 (TIMx_CR2)	284
12.4.3	TIM1 のスレーブモード制御レジスタ (TIMx_SMCR)	287
12.4.4	TIM1 DMA／割込み有効レジスタ (TIMx_DIER)	289
12.4.5	TIM1 のステータスレジスタ (TIMx_SR)	290
12.4.6	TIM1 のイベント生成レジスタ (TIMx_EGR)	292
12.4.7	TIM1 のキャプチャ／比較モードレジスタ 1 (TIMx_CCMR1)	293
12.4.8	TIM1 のキャプチャ／比較モードレジスタ 2 (TIMx_CCMR2)	296
12.4.9	TIM1 のキャプチャ／比較有効レジスタ (TIMx_CCER)	297

12.4.10	TIM1 のカウンタ (TIMx_CNT)	301
12.4.11	TIM1 のプリスケアラ (TIMx_PSC)	301
12.4.12	TIM1 の自動再ロードレジスタ (TIMx_ARR)	301
12.4.13	TIM1 繰り返しカウンタレジスタ (TIMx_RCR)	302
12.4.14	TIM1 のキャプチャ／比較レジスタ 1 (TIMx_CCR1)	302
12.4.15	TIM1 のキャプチャ／比較レジスタ 2 (TIMx_CCR2)	303
12.4.16	TIM1 のキャプチャ／比較レジスタ 3 (TIMx_CCR3)	303
12.4.17	TIM1 のキャプチャ／比較レジスタ 4 (TIMx_CCR4)	304
12.4.18	TIM1 ブレークおよびデッドタイムレジスタ (TIMx_BDTR)	304
12.4.19	TIM1 DMA 制御レジスタ (TIMx_DCR)	306
12.4.20	完全転送用の TIM1 DMA アドレス (TIMx_DMAR)	307
12.4.21	TIM1 レジスタマップ	308
13	汎用タイマ (TIM2 から TIM5)	310
13.1	TIM2 から TIM5 の概要	310
13.2	TIM2 から TIM5 の主な機能	310
13.3	TIM2 から TIM5 の機能説明	311
13.3.1	タイムベースユニット	311
13.3.2	カウンタモード	313
13.3.3	クロック選択	322
13.3.4	キャプチャ／比較チャネル	325
13.3.5	入力キャプチャモード	326
13.3.6	PWM 入力モード	327
13.3.7	強制出力モード	328
13.3.8	出力比較モード	328
13.3.9	PWM モード	329
13.3.10	ワンパルスモード	333
13.3.11	外部イベントによる OCxREF 信号のクリア	334
13.3.12	エンコーダインタフェースモード	335
13.3.13	タイマ入力 XOR 機能	337
13.3.14	タイマと外部トリガの同期	337
13.3.15	タイマの同期	341
13.3.16	デバッグモード	345
13.4	TIM2 から TIM5 のレジスタ	346
13.4.1	TIMx 制御レジスタ 1 (TIMx_CR1)	346
13.4.2	TIMx 制御レジスタ 2 (TIMx_CR2)	348
13.4.3	TIMx スレーブモード制御レジスタ (TIMx_SMCR)	349

13.4.4	TIMx DMA / 割込み有効レジスタ (TIMx_DIER)	351
13.4.5	TIMx ステータスレジスタ (TIMx_SR)	352
13.4.6	TIMx イベント生成レジスタ (TIMx_EGR)	354
13.4.7	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	355
13.4.8	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	358
13.4.9	TIMx キャプチャ/比較有効レジスタ (TIMx_CCER)	359
13.4.10	TIMx カウンタ (TIMx_CNT)	361
13.4.11	TIMx プリスケアラ (TIMx_PSC)	361
13.4.12	TIMx 自動再ロードレジスタ (TIMx_ARR)	362
13.4.13	TIMx キャプチャ/比較モードレジスタ 1 (TIMx_CCR1)	362
13.4.14	TIMx キャプチャ/比較モードレジスタ 2 (TIMx_CCR2)	363
13.4.15	TIMx キャプチャ/比較モードレジスタ 3 (TIMx_CCR3)	363
13.4.16	TIMx キャプチャ/比較モードレジスタ 4 (TIMx_CCR4)	364
13.4.17	TIMx DMA 制御レジスタ (TIMx_DCR)	364
13.4.18	完全転送の TIMx DMA アドレス (TIMx_DMAR)	365
13.4.19	TIM2 オプションレジスタ (TIM2_OR)	366
13.4.20	TIM5 オプションレジスタ (TIM5_OR)	366
13.4.21	TIMx レジスタマップ	367
14	汎用タイマ (TIM9 から TIM11)	369
14.1	TIM9/10/11 の概要	369
14.2	TIM9/10/11 の主な機能	369
14.2.1	TIM9 の主な機能	369
14.2.2	TIM10/TIM11 の主な機能	370
14.3	TIM9 から TIM11 の機能詳細	372
14.3.1	タイムベースユニット	372
14.3.2	カウンタモード	373
14.3.3	クロック選択	376
14.3.4	キャプチャ/比較チャネル	378
14.3.5	入力キャプチャモード	380
14.3.6	PWM 入力モード (TIM9 の場合のみ)	381
14.3.7	強制出力モード	382
14.3.8	出力比較モード	382
14.3.9	PWM モード	383
14.3.10	ワンパルスモード	384
14.3.11	TIM9 外部トリガ同期	386
14.3.12	タイマ同期 (TIM9)	388

14.3.13	デバッグモード	388
14.4	TIM9 レジスタ	389
14.4.1	TIM9 制御レジスタ 1 (TIMx_CR1)	389
14.4.2	TIM9 のスレーブモード制御レジスタ (TIMx_SMCR)	390
14.4.3	TIM9 割込み有効レジスタ (TIMx_DIER)	391
14.4.4	TIM9 ステータスレジスタ (TIMx_SR)	392
14.4.5	TIM9 のイベント生成レジスタ (TIMx_EGR)	394
14.4.6	TIM9 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	395
14.4.7	TIM9 のキャプチャ/比較有効レジスタ (TIMx_CCER)	398
14.4.8	TIM9 のカウンタ (TIMx_CNT)	399
14.4.9	TIM9 プリスケアラ (TIMx_PSC)	399
14.4.10	TIM9 自動再ロードレジスタ (TIMx_ARR)	399
14.4.11	TIM9 キャプチャ/比較レジスタ 1 (TIMx_CCR1)	400
14.4.12	TIM9 キャプチャ/比較レジスタ 2 (TIMx_CCR2)	400
14.4.13	TIM9 レジスタマップ	401
14.5	TIM10/11 レジスタ	402
14.5.1	TIM10/11 制御レジスタ 1 (TIMx_CR1)	402
14.5.2	TIM10/11 割込み有効レジスタ (TIMx_DIER)	403
14.5.3	TIM ステータスレジスタ (TIMx_SR)	403
14.5.4	TIM のイベント生成レジスタ (TIMx_EGR)	404
14.5.5	TIM10/11 のキャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	405
14.5.6	TIM10/11 のキャプチャ/比較有効レジスタ (TIMx_CCER)	408
14.5.7	TIM10/11 のカウンタ (TIMx_CNT)	409
14.5.8	TIM10/11 プリスケアラ (TIMx_PSC)	409
14.5.9	TIM10/11 自動再ロードレジスタ (TIMx_ARR)	409
14.5.10	TIM10/11 キャプチャ/比較レジスタ 1 (TIMx_CCR1)	410
14.5.11	TIM11 オプションレジスタ 1 (TIM11_OR)	410
14.5.12	TIM10/11 レジスタマップ	411
15	独立型ウォッチドッグ (IWDG)	413
15.1	IWDG の概要	413
15.2	IWDG の主な機能	413
15.3	IWDG の機能説明	413
15.3.1	ハードウェアウォッチドッグ	413
15.3.2	レジスタのアクセス保護	413
15.3.3	デバッグモード	414

15.4	IWDG レジスタ	415
15.4.1	キーレジスタ (IWDG_KR)	415
15.4.2	プリスケアラレジスタ (IWDG_PR)	415
15.4.3	再ロードレジスタ (IWDG_RLR)	416
15.4.4	ステータスレジスタ (IWDG_SR)	416
15.4.5	IWDG レジスタマップ	417
16	ウィンドウ型ウォッチドッグ (WWDG)	418
16.1	WWDG の概要	418
16.2	WWDG の主な機能	418
16.3	WWDG の機能説明	418
16.4	ウォッチドッグタイムアウトをプログラムする方法	420
16.5	デバッグモード	421
16.6	WWDG レジスタ	422
16.6.1	制御レジスタ (WWDG_CR)	422
16.6.2	設定レジスタ (WWDG_CFR)	423
16.6.3	ステータスレジスタ (WWDG_SR)	423
16.6.4	WWDG レジスタマップ	424
17	リアルタイムクロック (RTC)	425
17.1	概要	425
17.2	RTC の主な機能	426
17.3	RTC の機能説明	427
17.3.1	クロックとプリスケアラ	427
17.3.2	リアルタイムクロックとカレンダー	428
17.3.3	プログラム可能なアラーム	428
17.3.4	周期的自動ウェイクアップ	429
17.3.5	RTC の初期化と設定	430
17.3.6	カレンダーの読出し	431
17.3.7	RTC のリセット	432
17.3.8	RTC の同期	433
17.3.9	RTC リファレンスクロック検出	433
17.3.10	RTC の簡易デジタル較正	434
17.3.11	RTC の高精度デジタル較正	435
17.3.12	タイムスタンプ機能	437
17.3.13	タンパ検出	438

17.3.14	校正クロック出力	439
17.3.15	アラーム出力	439
17.4	RTC と低電力モード	440
17.5	RTC 割込み	440
17.6	RTC レジスタ	442
17.6.1	RTC 時刻レジスタ (RTC_TR)	442
17.6.2	RTC 日付レジスタ (RTC_DR)	443
17.6.3	RTC 制御レジスタ (RTC_CR)	444
17.6.4	RTC 初期化とステータスレジスタ (RTC_ISR)	447
17.6.5	RTC プリスケアラレジスタ (RTC_PRER)	449
17.6.6	RTC ウェイクアップタイマレジスタ (RTC_WUTR)	450
17.6.7	RTC 校正レジスタ (RTC_CALIBR)	450
17.6.8	RTC アラーム A レジスタ (RTC_ALRMAR)	451
17.6.9	RTC アラーム B レジスタ (RTC_ALRMBR)	452
17.6.10	RTC 書込み保護レジスタ (RTC_WPR)	453
17.6.11	RTC サブセカンドレジスタ (RTC_SSR)	454
17.6.12	RTC シフト制御レジスタ (RTC_SHIFTR)	455
17.6.13	RTC タイムスタンプ時刻レジスタ (RTC_TSTR)	456
17.6.14	RTC タイムスタンプ日付レジスタ (RTC_TSDR)	456
17.6.15	RTC タイムスタンプサブセカンドレジスタ (RTC_TSSTR)	457
17.6.16	RTC 校正レジスタ (RTC_CALR)	458
17.6.17	RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)	459
17.6.18	RTC アラーム A サブセカンドレジスタ (RTC_ALRMASR)	461
17.6.19	RTC アラーム B サブセカンドレジスタ (RTC_ALRMBSSR)	462
17.6.20	RTC バックアップレジスタ (RTC_BKPxR)	463
17.6.21	RTC レジスタマップ	463
18	I²C (Inter-integrated circuit) インタフェース	465
18.1	I ² C の概要	465
18.2	I ² C の主な特長	465
18.3	I ² C の機能説明	466
18.3.1	モード選択	466
18.3.2	I ² C スレーブモード	468
18.3.3	I ² C マスタモード	471
18.3.4	エラー条件	477
18.3.5	プログラム可能なノイズフィルタ	478

18.3.6	SDA/SCL ライン制御	479
18.3.7	SMBus	479
18.3.8	DMA リクエスト	482
18.3.9	パケットエラーチェック	483
18.4	I ² C 割込み	484
18.5	I ² C デバッグモード	486
18.6	I ² C レジスタ	486
18.6.1	I ² C 制御レジスタ 1 (I2C_CR1)	486
18.6.2	I ² C 制御レジスタ 2 (I2C_CR2)	488
18.6.3	I ² C 自己アドレスレジスタ 1 (I2C_OAR1)	490
18.6.4	I ² C 自己アドレスレジスタ 2 (I2C_OAR2)	490
18.6.5	I ² C データレジスタ (I2C_DR)	491
18.6.6	I ² C ステータスレジスタ 1 (I2C_SR1)	491
18.6.7	I ² C ステータスレジスタ 2 (I2C_SR2)	495
18.6.8	I ² C クロック制御レジスタ (I2C_CCR)	496
18.6.9	I ² C TRISE レジスタ (I2C_TRISE)	497
18.6.10	I ² C FLTR レジスタ (I2C_FLTR)	498
18.6.11	I ² C のレジスタマップ	499

19	USART (Universal synchronous asynchronous receiver transmitter)	500
19.1	USART の概要	500
19.2	USART の主な機能	500
19.3	USART の機能詳細	501
19.3.1	USART キャラクタの説明	504
19.3.2	トランスミッタ	505
19.3.3	レシーバ	508
19.3.4	フラクショナルボーレート生成	513
19.3.5	クロック偏差に対する USART レシーバの許容誤差	522
19.3.6	マルチプロセッサ通信	523
19.3.7	パリティ制御	525
19.3.8	LIN (Local Interconnection Network) モード	526
19.3.9	USART 同期モード	528
19.3.10	単線半二重通信	530
19.3.11	スマートカード	531
19.3.12	Ir SIR ENDEC ブロック	533
19.3.13	DMA を使用した連続通信	535

19.3.14	ハードウェアフロー制御	537
19.4	USART 割込み	539
19.5	USART モード設定	540
19.6	USART レジスタ	540
19.6.1	ステータスレジスタ (USART_SR)	540
19.6.2	データレジスタ (USART_DR)	543
19.6.3	ボーレートレジスタ (USART_BRR)	543
19.6.4	制御レジスタ 1 (USART_CR1)	544
19.6.5	制御レジスタ 2 (USART_CR2)	546
19.6.6	制御レジスタ 3 (USART_CR3)	547
19.6.7	ガード時間およびプリスケアラレジスタ (USART_GTPR)	549
19.6.8	USART レジスタマップ	550
20	シリアルペリフェラルインタフェース (SPI)	551
20.1	SPI の概要	551
20.2	SPI および I ² S の主な機能	552
20.2.1	SPI の機能	552
20.2.2	I ² S の機能	553
20.3	SPI の機能説明	554
20.3.1	概要	554
20.3.2	SPI のスレーブモード設定	557
20.3.3	SPI のマスタモード設定	560
20.3.4	SPI の半二重通信設定	562
20.3.5	データの送受信手順	562
20.3.6	CRC 計算	569
20.3.7	ステータスフラグ	571
20.3.8	SPI の無効化	572
20.3.9	DMA (ダイレクトメモリアクセス) を使用する SPI 通信	573
20.3.10	エラーフラグ	575
20.3.11	SPI 割込み	576
20.4	I ² S の機能説明	577
20.4.1	I ² S の概要	577
20.4.2	I ² S 全二重	578
20.4.3	サポートされるオーディオプロトコル	579
20.4.4	クロックジェネレータ	587
20.4.5	I ² S マスタモード	589

20.4.6	I ² S スレーブモード	591
20.4.7	ステータスフラグ	592
20.4.8	エラーフラグ	593
20.4.9	I ² S 割込み	594
20.4.10	DMA の機能	594
20.5	SPI および I ² S レジスタ	595
20.5.1	SPI 制御レジスタ 1 (SPI_CR1) (I ² S モードでは使用しません)	595
20.5.2	SPI 制御レジスタ 2 (SPI_CR2)	597
20.5.3	SPI ステータスレジスタ (SPI_SR)	598
20.5.4	SPI データレジスタ (SPI_DR)	599
20.5.5	SPI CRC 多項式レジスタ (SPI_CRCPR) (I ² S モードでは使用しません)	600
20.5.6	SPI RX CRC レジスタ (SPI_RXCRCR) (I ² S モードでは使用しません)	600
20.5.7	SPI TX CRC レジスタ (SPI_TXCRCR) (I ² S モードでは使用しません)	601
20.5.8	SPI_I ² S 設定レジスタ (SPI_I2SCFGR)	601
20.5.9	SPI_I ² S プリスケアラレジスタ (SPI_I2SPR)	603
20.5.10	SPI レジスタマップ	604
21	セキュアデジタル入力／出力インタフェース (SDIO)	605
21.1	SDIO の主な機能	605
21.2	SDIO バストポロジ	605
21.3	SDIO の機能詳細	607
21.3.1	SDIO アダプタ	609
21.3.2	SDIO APB2 インタフェース	619
21.4	カード機能詳細	620
21.4.1	カード識別モード	620
21.4.2	カードリセット	620
21.4.3	動作電圧範囲の検証	620
21.4.4	カード識別プロセス	621
21.4.5	ブロック書込み	622
21.4.6	ブロック読出し	622
21.4.7	ストリームアクセス、ストリーム書込み、およびストリーム読出し (MultiMediaCard のみ)	623
21.4.8	消去：グループ消去とセクタ消去	624
21.4.9	ワイドバス選択または選択解除	624
21.4.10	保護管理	625

21.4.11	カードステータスレジスタ	628
21.4.12	SD ステータスレジスタ	631
21.4.13	SD I/O モード	635
21.4.14	コマンドおよびレスポンス	636
21.5	レスポンスフォーマット	639
21.5.1	R1 (ノーマルレスポンスコマンド)	639
21.5.2	R1b	639
21.5.3	R2 (CID、CSD レジスタ)	640
21.5.4	R3 (OCR レジスタ)	640
21.5.5	R4 (高速 I/O)	641
21.5.6	R4b	641
21.5.7	R5 (割込みリクエスト)	642
21.5.8	R6	642
21.6	SDIO I/O カード固有の操作	643
21.6.1	SDIO_D2 信号による SDIO I/O 読出しウェイト操作	643
21.6.2	SDIO_CK の停止による SDIO 読出しウェイト操作	643
21.6.3	SDIO サスペンド/レジューム操作	643
21.6.4	SDIO 割込み	644
21.7	CE-ATA 固有の操作	644
21.7.1	コマンド完了信号無効	644
21.7.2	コマンド完了信号有効	644
21.7.3	CE-ATA 割込み	644
21.7.4	CMD61 のアボート	644
21.8	HW フロー制御	645
21.9	SDIO レジスタ	645
21.9.1	SDIO 電源制御レジスタ (SDIO_POWER)	645
21.9.2	SDIO クロック制御レジスタ (SDIO_CLKCR)	646
21.9.3	SDIO 引数レジスタ (SDIO_ARG)	647
21.9.4	SDIO コマンドレジスタ (SDIO_CMD)	648
21.9.5	SDIO コマンドレスポンスレジスタ (SDIO_RESPCMD)	649
21.9.6	SDIO レスポンス 1 ~ 4 レジスタ (SDIO_RESPx)	649
21.9.7	SDIO データタイマレジスタ (SDIO_DTIMER)	650
21.9.8	SDIO データ長レジスタ (SDIO_DLEN)	650
21.9.9	SDIO データ制御レジスタ (SDIO_DCTRL)	651
21.9.10	SDIO データカウンタレジスタ (SDIO_DCOUNT)	652
21.9.11	SDIO ステータスレジスタ (SDIO_STA)	653

21.9.12	SDIO 割込みクリアレジスタ (SDIO_ICR)	654
21.9.13	SDIO マスクレジスタ (SDIO_MASK)	656
21.9.14	SDIO FIFO カウンタレジスタ (SDIO_FIFOCNT)	659
21.9.15	SDIO データ FIFO レジスタ (SDIO_FIFO)	659
21.9.16	SDIO レジスタマップ	660
22	USB On-The-Go フルスピード (OTG_FS)	661
22.1	OTG_FS の概要	661
22.2	OTG_FS の主な機能	661
22.2.1	標準機能	662
22.2.2	ホストモード機能	662
22.2.3	ペリフェラルモード機能	663
22.3	OTG_FS の機能説明	663
22.3.1	OTG フルスピードコア	664
22.3.2	フルスピード OTG PHY	664
22.4	OTG デュアルロールデバイス (DRD)	665
22.4.1	ID ラインの検出	665
22.4.2	HNP デュアルロールデバイス	665
22.4.3	SRP デュアルロールデバイス	666
22.5	USB ペリフェラル	666
22.5.1	SRP 対応ペリフェラル	667
22.5.2	ペリフェラルの状態	667
22.5.3	ペリフェラルエンドポイント	668
22.6	USB ホスト	670
22.6.1	SRP 対応ホスト	671
22.6.2	USB ホストの状態	671
22.6.3	ホストチャネル	673
22.6.4	ホストスケジューラ	674
22.7	SOF トリガ	675
22.7.1	ホストの SOF	675
22.7.2	ペリフェラルの SOF	676
22.8	電力オプション	676
22.9	OTG_FS_HFIR レジスタの動的更新	677
22.10	USB データ FIFO	677
22.11	ペリフェラル FIFO アーキテクチャ	678
22.11.1	ペリフェラル Rx FIFO	678

22.11.2	ペリフェラル Tx FIFO	679
22.12	ホスト FIFO アーキテクチャ	679
22.12.1	ホスト Rx FIFO	679
22.12.2	ホスト Tx FIFO	680
22.13	FIFO RAM の割り当て	680
22.13.1	デバイスモード	680
22.13.2	ホストモード	681
22.14	USB システムのパフォーマンス	681
22.15	OTG_FS 割込み	682
22.16	OTG_FS 制御／ステータスレジスタ	684
22.16.1	CSR メモリマップ	685
22.16.2	OTG_FS グローバルレジスタ	690
22.16.3	ホストモードレジスタ	711
22.16.4	デバイスモードレジスタ	721
22.16.5	OTG_FS パワーおよびクロックゲーティング制御レジスタ (OTG_FS_PCGCTL)	742
22.16.6	OTG_FS レジスタマップ	743
22.17	OTG_FS プログラミングモデル	751
22.17.1	コアの初期化	751
22.17.2	ホストの初期化	752
22.17.3	デバイスの初期化	752
22.17.4	ホストプログラミングモデル	753
22.17.5	デバイスプログラミングモデル	769
22.17.6	動作モデル	771
22.17.7	最悪ケースの応答時間	787
22.17.8	OTG プログラミングモデル	789
23	デバッグサポート (DBG)	795
23.1	概要	795
23.2	ARM® リファレンス資料	796
23.3	SWJ デバッグポート (シリアルワイヤと JTAG)	796
23.3.1	JTAG-DP または SW-DP の選択メカニズム	797
23.4	ピン名とデバッグポートピン	797
23.4.1	SWJ デバッグポートピン	798
23.4.2	柔軟性の高い SWJ-DP ピンの割り当て	798
23.4.3	JTAG ピンでの内部プルアップ／プルダウン	799

23.4.4	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	800
23.5	STM32F401xB/C および STM32F401xD/E の JTAG TAP 接続	800
23.6	ID コードとロック機構	802
23.6.1	MCU デバイス ID コード	802
23.6.2	バウンダリスキャン TAP	803
23.6.3	FPU 搭載 Cortex®-M4 TAP	803
23.6.4	FPU 搭載 Cortex®-M4 JEDEC-106 ID コード	803
23.7	JTAG デバッグポート	803
23.8	SW デバッグポート	805
23.8.1	SW プロトコルの概要	805
23.8.2	SW プロトコルシーケンス	805
23.8.3	SW-DP ステートマシン（リセット、アイドル状態、ID コード）	807
23.8.4	DP と AP の読み出し／書き込みアクセス	807
23.8.5	SW-DP レジスタ	808
23.8.6	SW-AP レジスタ	808
23.9	AHB-AP（AHB アクセスポート）- JTAG-DP と SW-DP の両方に有効	809
23.10	コアデバッグ	810
23.11	システムリセット中のデバッグホスト接続機能	811
23.12	FPB（フラッシュパッチブレークポイント）	811
23.13	DWT（データウォッチポイントトリガ）	812
23.14	ITM（計測トレースマクロセル）	812
23.14.1	概要	812
23.14.2	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	812
23.15	ETM（組み込みトレースマクロセル）	814
23.15.1	概要	814
23.15.2	信号プロトコル、パケットタイプ	814
23.15.3	主な ETM レジスタ	815
23.15.4	設定例	815
23.16	MCU デバッグコンポーネント（DBGMCU）	815
23.16.1	低電力モードのデバッグサポート	816
23.16.2	タイマ、ウォッチドッグ、および I ² C のデバッグサポート	816
23.16.3	デバッグ MCU 設定レジスタ	816
23.16.4	デバッグ MCU APB1 凍結レジスタ（DBGMCU_APB1_FZ）	818
23.16.5	デバッグ MCU APB2 フリーズレジスタ（DBGMCU_APB2_FZ）	819

23.17	TPIU（トレースポートインタフェースユニット）	820
23.17.1	概要	820
23.17.2	TRACE ピンの割当て	820
23.17.3	TPIU フォーマッタ	822
23.17.4	TPIU フレーム同期パケット	822
23.17.5	同期フレームパケットの送信	823
23.17.6	同期モード	823
23.17.7	非同期モード	823
23.17.8	TRACECLKIN 接続 (STM32F401xB/C および STM32F401xD/E 内)	823
23.17.9	TPIU レジスタ	824
23.17.10	設定例	824
23.18	DBG レジスタマップ	825
24	デバイス電子署名	826
24.1	ユニークデバイス ID レジスタ（96 ビット）	826
24.2	フラッシュサイズ	827

表の一覧

表 1.	STM32F401xB/C および STM32F401xD/E のレジスタ境界アドレス	38
表 2.	ブートモード	41
表 3.	メモリ割当てとブートモード／物理的な再割当て STM32F401xB/C	42
表 4.	メモリ割当てとブートモード／物理的な再割当て STM32F401xD/E	43
表 5.	フラッシュモジュールの構成 (STM32F401xB/C および STM32F401xD/E)	45
表 6.	CPU クロック (HCLK) 周波数によるウェイトステート数	46
表 7.	プログラム／消去の並列処理	50
表 8.	フラッシュ割込みリクエスト	52
表 9.	オプションバイトの構成	52
表 10.	オプションバイトの説明	52
表 11.	アクセスと読出し保護レベル	56
表 12.	OTP エリアの構成	59
表 13.	フラッシュレジスタマップとリセット値	67
表 14.	CRC 計算ユニットのレジスタマップとリセット値	70
表 15.	低電力モードの概要	78
表 16.	Sleep-now への移行と終了	79
表 17.	Sleep-on-exit への移行と終了	79
表 18.	STOP モードの動作モード	80
表 19.	STOP モードへの移行と終了	81
表 20.	STANDBY モードへの移行と終了	83
表 21.	PWR - レジスタマップとリセット値	90
表 22.	RCC レジスタマップとリセット値 STM32F401xB/C および STM32F401xD/E	137
表 23.	SYSCFG レジスタマップとリセット値	144
表 24.	ポートビット設定表	147
表 25.	柔軟性の高い SWJ-DP ピンの割り当て	149
表 26.	RTC 追加機能	156
表 27.	GPIO レジスタマップとリセット値	162
表 28.	DMA1 リクエストマッピング (STM32F401xB/C および STM32F401xD/E)	168
表 29.	DMA2 リクエストマッピング (STM32F401xB/C および STM32F401xD/E)	169
表 30.	転送元および転送先アドレス	170
表 31.	ダブルバッファモード (DBM=1) における転送元および転送先アドレスレジスタ	175
表 32.	パッキング／アンパッキングおよびエンディアン動作 (ビット PINC=MINC=1)	176
表 33.	PSIZE と MSIZE に対する NDT に関する制限事項	176
表 34.	FIFO 閾値設定	179
表 35.	実現可能な DMA 設定	183
表 36.	DMA 割込みリクエスト	185
表 37.	DMA レジスタマップとリセット値	196
表 38.	STM32F401xB/CSTM32F401xD/E のベクタテーブル	200
表 39.	外部割込み／イベントコントローラのレジスタマップとリセット値	209
表 40.	ADC ピン	212
表 41.	アナログウォッチドッグチャネル選択	215
表 42.	トリガ極性の設定	219
表 43.	レギュラチャネルの外部トリガ	220
表 44.	インジェクトチャネルの外部トリガ	221
表 45.	ADC 割込み	225
表 46.	ADC グローバルレジスタマップ	239
表 47.	ADC レジスタマップと各 ADC のリセット値	239
表 48.	ADC レジスタマップとリセット値 (共通 ADC レジスタ)	240

表 49.	カウント方向とエンコーダ信号	275
表 50.	TIMx 内部トリガ接続	288
表 51.	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	300
表 52.	TIM1 レジスタマップとリセット値	308
表 53.	カウント方向とエンコーダ信号	336
表 54.	TIMx 内部トリガ接続	351
表 55.	標準 OCx チャンネルの出力制御ビット	361
表 56.	TIM2 から TIM5 のレジスタマップとリセット値	367
表 57.	TIMx 内部トリガ接続	391
表 58.	標準 OCx チャンネルの出力制御ビット	399
表 59.	TIM9 レジスタマップとリセット値	401
表 60.	標準 OCx チャンネルの出力制御ビット	408
表 61.	TIM10/11 レジスタマップとリセット値	411
表 62.	最小/最大 IWDG タイムアウト期間 (32 kHz) (LSI)	414
表 63.	IWDG レジスタマップとリセット値	417
表 64.	30 MHz (f_{PCLK1}) でのタイムアウトの最大値および最小値	421
表 65.	WWDG レジスタマップとリセット値	424
表 66.	低消費電力モードが RTC に与える影響	440
表 67.	割込み制御ビット	441
表 68.	RTC レジスタマップとリセット値	463
表 69.	Thd:dat (max) に準拠した最大 DNF[3:0] 値	478
表 70.	SMBus と I ² C	480
表 71.	I ² C 割込みリクエスト	484
表 72.	I ² C レジスタマップとリセット値	499
表 73.	サンプリングされたデータからのノイズ検出	512
表 74.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8$ MHz または $f_{PCLK} = 12$ MHz)、16 倍のオーバーサンプリング	515
表 75.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8$ MHz または $f_{PCLK} = 12$ MHz)、8 倍のオーバーサンプリング	516
表 76.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16$ MHz または $f_{PCLK} = 24$ MHz)、16 倍のオーバーサンプリング	516
表 77.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16$ MHz または $f_{PCLK} = 24$ MHz)、8 倍のオーバーサンプリング	517
表 78.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8$ MHz または $f_{PCLK} = 16$ MHz)、16 倍のオーバーサンプリング	518
表 79.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8$ MHz または $f_{PCLK} = 16$ MHz)、8 倍のオーバーサンプリング	518
表 80.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30$ MHz または $f_{PCLK} = 60$ MHz)、16 倍のオーバーサンプリング	519
表 81.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30$ MHz または $f_{PCLK} = 60$ MHz)、8 倍のオーバーサンプリング	520
表 82.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42$ MHz または $f_{PCLK} = 84$ MHz)、16 倍のオーバーサンプリング	521
表 83.	プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42$ MHz または $f_{PCLK} = 84$ MHz)、8 倍のオーバーサンプリング	522
表 84.	USART レシーバの許容誤差 (DIV_Fraction が 0 の場合)	523
表 85.	USART レシーバ許容誤差 (DIV_Fraction が 0 でない場合)	523
表 86.	フレームフォーマット	525
表 87.	USART 割込みリクエスト	539
表 88.	USART モード設定	540
表 89.	USART レジスタマップとリセット値	550
表 90.	SPI 割込みリクエスト	576

表 91.	オーディオ周波数精度 (PLL M VCO = 1 MHz または 2 MHz の場合)	588
表 92.	I ² S 割込みリクエスト	594
表 93.	SPI レジスタマップとリセット値	604
表 94.	SDIO I/O 定義	609
表 95.	コマンドフォーマット	613
表 96.	ショートレスポンスフォーマット	613
表 97.	ロングレスポンスフォーマット	614
表 98.	コマンドパスステータスフラグ	614
表 99.	データトークンフォーマット	617
表 100.	送信 FIFO ステータスフラグ	618
表 101.	受信 FIFO ステータスフラグ	619
表 102.	カードステータス	629
表 103.	SD ステータス	631
表 104.	速度クラスコードフィールド	632
表 105.	移動パフォーマンスフィールド	633
表 106.	AU_SIZE フィールド	633
表 107.	最大 AU サイズ	633
表 108.	消去サイズフィールド	634
表 109.	消去タイムアウトフィールド	634
表 110.	消去オフセットフィールド	634
表 111.	ブロック指向書込みコマンド	637
表 112.	ブロック指向書込み保護コマンド	637
表 113.	消去コマンド	638
表 114.	I/O モードコマンド	638
表 115.	カードのロック	638
表 116.	アプリケーション固有コマンド	639
表 117.	R1 レスポンス	639
表 118.	R2 レスポンス	640
表 119.	R3 レスポンス	640
表 120.	R4 レスポンス	641
表 121.	R4b レスポンス	641
表 122.	R5 レスポンス	642
表 123.	R6 レスポンス	642
表 124.	レスポンスタイプと SDIO_RESPx レジスタ	649
表 125.	SDIO レジスタマップ	660
表 126.	コアグローバル制御およびステータスレジスタ (CSR)	685
表 127.	ホストモード制御およびステータスレジスタ (CSR)	686
表 128.	デバイスモード制御およびステータスレジスタ	687
表 129.	データ FIFO (DFIFO) アクセスレジスタマップ	689
表 130.	パワーおよびクロックゲーティング制御およびステータスレジスタ	689
表 131.	TRDT 値	695
表 132.	ソフト切断のための最短時間	723
表 133.	OTG_FS レジスタマップとリセット値	743
表 134.	SWJ デバッグポートピン	798
表 135.	柔軟性の高い SWJ-DP ピンの割り当て	798
表 136.	JTAG デバッグポートのデータレジスタ	803
表 137.	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	805
表 138.	パケットリクエスト (8 ビット)	806
表 139.	ACK 応答 (3 ビット)	806
表 140.	データ転送 (33 ビット)	806
表 141.	SW-DP レジスタ	808

表 142.	FPU 搭載 Cortex®-M4 AHB-AP レジスタ	809
表 143.	コアデバッグレジスタ	810
表 144.	主な ITM レジスタ	813
表 145.	主な ETM レジスタ	815
表 146.	非同期 TRACE ピンの割当て	820
表 147.	同期 TRACE ピンの割当て	821
表 148.	柔軟性の高い TRACE ピン割り当て	821
表 149.	重要な TPIU レジスタ	824
表 150.	DBG レジスタマップとリセット値	825
表 151.	文書改版履歴	833
表 152.	日本語版文書改版履歴	835

図の一覧

図 1.	システムアーキテクチャ	36
図 2.	システムアーキテクチャ内のフラッシュメモリインタフェース接続 (STM32F401xB/C および STM32F401xD/E)	44
図 3.	連続 32 ビット命令の実行	48
図 4.	RDP レベル	56
図 5.	PCROP レベル	58
図 6.	CRC 計算ユニットのブロック図	68
図 7.	電源の概要	71
図 8.	パワーオンリセット/パワーダウンリセット波形	74
図 9.	BOR の閾値	75
図 10.	PVD の閾値	76
図 11.	簡略化されたリセット回路図	92
図 12.	クロックツリー	94
図 13.	HSE/LSE クロックソース	96
図 14.	入力キャプチャモードにおける TIM5 を使用した周波数測定	101
図 15.	入力キャプチャモードにおける TIM11 を使用した周波数測定	102
図 16.	5 V トレラント I/O ポートビットの基本構成	146
図 17.	オルタネート機能の選択: STM32F401xB/C および STM32F401xD/E	150
図 18.	入力フローティング/プルアップ/プルダウン設定	152
図 19.	出力設定	153
図 20.	オルタネート機能設定	154
図 21.	ハイインピーダンスアナログ設定	154
図 22.	DMA ブロック図	166
図 23.	2 つの DMA コントローラのシステム実装 (STM32F401xB/C および STM32F401xD/E)	167
図 24.	チャネル選択	168
図 25.	“ペリフェラルからメモリ”モード	171
図 26.	“メモリからペリフェラル”モード	172
図 27.	メモリ間モード	173
図 28.	FIFO の構造	178
図 29.	外部割込み/イベントコントローラのブロック図	203
図 30.	外部割込み/イベント GPIO の配置	205
図 31.	1 つの ADC ブロック図	211
図 32.	タイミング図	214
図 33.	アナログウォッチドッグによって保護される領域	214
図 34.	インジェクト変換の遅延時間	216
図 35.	12 ビットデータの右詰め	218
図 36.	12 ビットデータの左詰め	218
図 37.	6 ビットデータの左詰め	218
図 38.	温度センサおよび VREFINT チャネルのブロック図	223
図 39.	高機能制御タイマのブロック図	243
図 40.	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	245
図 41.	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	245
図 42.	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	246
図 43.	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	246
図 44.	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	247
図 45.	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	247

図 46.	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	247
図 47.	ARPE=1 の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	248
図 48.	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	249
図 49.	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	249
図 50.	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	249
図 51.	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	250
図 52.	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	250
図 53.	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	251
図 54.	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	252
図 55.	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	252
図 56.	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	252
図 57.	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	253
図 58.	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	253
図 59.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	254
図 60.	内部クロック分周比 1 の場合の、通常モードの制御回路	255
図 61.	TI2 外部クロックの接続例	255
図 62.	外部クロックモード 1 の制御回路	256
図 63.	外部トリガ入力ブロック	257
図 64.	外部クロックモード 2 の制御回路	257
図 65.	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)	258
図 66.	キャプチャ/比較チャンネル 1 メイン回路	258
図 67.	キャプチャ/比較チャンネル (チャンネル 1 から 3) の出力ステージ	259
図 68.	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	259
図 69.	PWM 入力モードタイミング	261
図 70.	出力比較モード、OC1 の反転	263
図 71.	エッジアライン PWM 波形 (ARR=8)	264
図 72.	センターアライン PWM 波形 (ARR=8)	265
図 73.	デッドタイム挿入のある相補出力	266
図 74.	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	267
図 75.	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	267
図 76.	ブレークに対する出力の動作	270
図 77.	TIMx_OCxREF のクリア	271
図 78.	6 ステップ生成 COM の例 (OSSR=1)	272
図 79.	ワンパルスモードの例	273
図 80.	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	276
図 81.	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	276
図 82.	ホールセンサインタフェースの例	278
図 83.	リセットモードの制御回路	279
図 84.	ゲートモードの制御回路	280
図 85.	トリガモードの制御回路	281
図 86.	外部クロックモード 2 + トリガモードの制御回路	282
図 87.	汎用タイマのブロック図	311
図 88.	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	312
図 89.	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	313
図 90.	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	314
図 91.	内部クロック分周比が 2 の場合のカウンタのタイミング図	314
図 92.	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	314

図 93.	内部クロック分周比が N の場合のカウンタのタイミング図	315
図 94.	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	315
図 95.	ARPE=1 (TIMx_ARR はプリロードされる) の場合の更新イベント時の カウンタのタイミング図	316
図 96.	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	317
図 97.	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	317
図 98.	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	317
図 99.	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	318
図 100.	更新イベント時のカウンタのタイミング図	318
図 101.	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	319
図 102.	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	320
図 103.	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	320
図 104.	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	320
図 105.	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	321
図 106.	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	321
図 107.	内部クロック分周比 1 の場合の、通常モードの制御回路	322
図 108.	TI2 外部クロックの接続例	322
図 109.	外部クロックモード 1 の制御回路	323
図 110.	外部トリガ入力ブロック	324
図 111.	外部クロックモード 2 の制御回路	324
図 112.	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)	325
図 113.	キャプチャ/比較チャンネル 1 メイン回路	325
図 114.	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	326
図 115.	PWM 入力モードタイミング	328
図 116.	出力比較モード、OC1 の反転	329
図 117.	エッジアライン PWM 波形 (ARR=8)	330
図 118.	センターアライン PWM 波形 (ARR=8)	332
図 119.	ワンパルスモードの例	333
図 120.	TIMx_OCxREF のクリア	335
図 121.	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	336
図 122.	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	337
図 123.	リセットモードの制御回路	338
図 124.	ゲートモードの制御回路	339
図 125.	トリガモードの制御回路	339
図 126.	外部クロックモード 2 + トリガモードの制御回路	340
図 127.	マスタ/スレーブタイマの例	341
図 128.	タイマ 1 の OC1REF によるタイマ 2 のゲート操作	342
図 129.	タイマ 1 の有効化によるタイマ 2 のゲート操作	343
図 130.	タイマ 1 の更新によるタイマ 2 のトリガ	344
図 131.	タイマ 1 の有効化によるタイマ 2 のトリガ	344
図 132.	タイマ 1 の TI1 入力によるタイマ 1 および 2 のトリガ	345
図 133.	汎用タイマのブロック図 (TIM9)	370
図 134.	汎用タイマのブロック図 (TIM10/11)	371
図 135.	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	373
図 136.	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	373
図 137.	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	374
図 138.	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	374
図 139.	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	375
図 140.	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	375

図 141.	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	375
図 142.	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	376
図 143.	内部クロック分周比 1 の場合の、通常モードの制御回路	377
図 144.	TI2 外部クロックの接続例	377
図 145.	外部クロックモード 1 の制御回路	378
図 146.	キャプチャ/比較チャネル (例: チャネル 1 入力ステージ)	378
図 147.	キャプチャ/比較チャネル 1 メイン回路	379
図 148.	キャプチャ/比較チャネル (チャネル 1) の出力ステージ	379
図 149.	PWM 入力モードタイミング	381
図 150.	出力比較モード、OC1 の反転	383
図 151.	エッジアライン PWM 波形 (ARR=8)	384
図 152.	ワンパルスモードの例	385
図 153.	リセットモードの制御回路	386
図 154.	ゲートモードの制御回路	387
図 155.	トリガモードの制御回路	388
図 156.	独立型ウォッチドッグのブロック図	414
図 157.	ウォッチドッグのブロック図	418
図 158.	ウィンドウ型ウォッチドッグのタイミング図	420
図 159.	RTC ブロック図	427
図 160.	I ² C バスプロトコル	467
図 161.	I ² C ブロック図	468
図 162.	スレーブトランスミッタの転送シーケンス	470
図 163.	スレーブ受信の転送シーケンス図	471
図 164.	マスタ送信の転送シーケンス図	474
図 165.	マスタ受信の転送シーケンス図	476
図 166.	I ² C 割込みマッピング図	485
図 167.	USART のブロック図	503
図 168.	ワード長のプログラミング	504
図 169.	設定可能なストップビット	506
図 170.	送信時の TC/TXE の動作	507
図 171.	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	508
図 172.	データサンプリング (16 倍のオーバーサンプリング)	511
図 173.	データサンプリング (8 倍のオーバーサンプリング)	512
図 174.	アイドルライン検出を使用したミュートモード	524
図 175.	アドレスマーク検出を使用したミュートモード	525
図 176.	LIN モードでのブレーク検出 (11 ビットブレーク長、LBDL=1)	527
図 177.	LIN モードでのブレーク検出とフレーミングエラー検出	528
図 178.	USART の同期送信の例	529
図 179.	USART データクロックタイミング図 (M=0)	529
図 180.	USART データクロックタイミング図 (M=1)	530
図 181.	RX データセットアップ/ホールド時間	530
図 182.	ISO 7816-3 非同期プロトコル	531
図 183.	ストップビット 1.5 個を使用したパリティエラー検出	532
図 184.	IrDA SIR ENDEC- ブロック図	534
図 185.	IrDA データ変調 (3/16) - 通常モード	534
図 186.	DMA を使用した送信	535
図 187.	DMA を使用した受信	536
図 188.	2 つの USART 間のハードウェアフロー制御	537
図 189.	RTS フロー制御	537
図 190.	CTS フロー制御	538

図 191.	USART 割込みマッピング図	539
図 192.	SPI ブロック図	554
図 193.	シングルマスタ/シングルスレーブアプリケーション	555
図 194.	データクロックのタイミング図	557
図 195.	TI モード - スレーブモード、シングル転送	559
図 196.	TI モード - スレーブモード、連続転送	559
図 197.	TI モード - マスタモード、シングル転送	561
図 198.	TI モード - マスタモード、連続転送	561
図 199.	マスタ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合	565
図 200.	スレーブ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合	565
図 201.	マスタ送信専用モードでの TXE/BSY 動作 (BIDIMODE=0 および RXONLY=0) 連続転送の場合	566
図 202.	スレーブ送信専用モードでの TXE/BSY 動作 (BIDIMODE=0 および RXONLY=0) 連続転送の場合	567
図 203.	受信専用モードでの RXNE 動作 (BIDIRMODE=0 および RXONLY=1) 連続転送の場合	568
図 204.	送信時の TXE/BSY 動作 (BIDIRMODE=0 および RXONLY=0) 不連続転送の場合	569
図 205.	DMA を使用した送信	574
図 206.	DMA を使用した受信	574
図 207.	TI モードフレームフォーマットエラー検出	576
図 208.	I ² S ブロック図	577
図 209.	I ² S 全二重のブロック図	578
図 210.	フィリップス I ² S プロトコルの波形 (16/32 ビットフル精度、CPOL = 0)	580
図 211.	フィリップス I ² S 規格の波形 (24 ビットフレーム、CPOL = 0)	580
図 212.	0x8EAA33 の送信	581
図 213.	0x8EAA33 の受信	581
図 214.	フィリップス I ² S 規格 (CPOL=0 で 32 ビットパケットフレームに拡張された 16 ビット)	581
図 215.	例	582
図 216.	CPOL=0 で MSB 詰め 16 ビットまたは 32 ビットフル精度長	582
図 217.	CPOL=0 で MSB 詰め 24 ビットフレーム長	582
図 218.	CPOL=0 で MSB 詰め 16 ビットを 32 ビットパケットフレームに拡張	583
図 219.	CPOL=0 で LSB 詰め 16 ビットまたは 32 ビットフル精度	583
図 220.	CPOL=0 で LSB 詰め 24 ビットフレーム長	584
図 221.	0x3478AE を送信するために必要な動作	584
図 222.	0x3478AE の受信に必要な動作	584
図 223.	CPOL=0 で LSB 詰め 16 ビットを 32 ビットパケットフレームに拡張	585
図 224.	32 ビットパケットフレームに拡張された LSB 詰め 16 ビットの例	585
図 225.	PCM 規格の波形 (16 ビット)	586
図 226.	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)	586
図 227.	オーディオサンプリング周波数の定義	587
図 228.	I ² S クロックジェネレータのアーキテクチャ	587
図 229.	SDIO 「no response」と「no data」操作	606
図 230.	SDIO (複数) ブロック読出し操作	606
図 231.	SDIO (複数) ブロック書込み操作	606
図 232.	SDIO 順次読出し操作	607
図 233.	SDIO 順次書込み操作	607
図 234.	SDIO ブロック図	608
図 235.	SDIO アダプタ	609
図 236.	制御ユニット	610

図 237.	SDIO アダプタコマンドパス	611
図 238.	コマンドパスステートマシン (CPSM)	612
図 239.	SDIO コマンド転送	612
図 240.	データパス	615
図 241.	データパスステートマシン (DPSM)	616
図 242.	OTG フルスピードブロック図	663
図 243.	OTG A-B デバイス接続	665
図 244.	USB ペリフェラル専用接続	667
図 245.	USB ホスト専用接続	671
図 246.	SOF 接続性	675
図 247.	OTG_FS_HFIR の動的更新	677
図 248.	デバイスモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング	678
図 249.	ホストモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング	679
図 250.	割込みの階層	683
図 251.	CSR メモリマップ	685
図 252.	送信 FIFO 書込みタスク	754
図 253.	受信 FIFO 読出しタスク	755
図 254.	通常のバルク/コントロール OUT/SETUP およびバルク/コントロール IN トランザクション	756
図 255.	バルク/コントロール IN トランザクション	760
図 256.	通常インタラプト OUT/IN トランザクション	762
図 257.	通常のアイソクロナス OUT/IN トランザクション	766
図 258.	受信 FIFO パケット読出し	772
図 259.	SETUP パケットの処理	774
図 260.	バルク OUT トランザクション	780
図 261.	TRDT 最大タイミングのケース	789
図 262.	A デバイスの SRP	790
図 263.	B デバイスの SRP	791
図 264.	A デバイスの HNP	792
図 265.	B デバイスの HNP	793
図 266.	ブロック図 - STM32 MCU および FPU 搭載 Cortex®-M4 レベルのデバッグサポート ..	795
図 267.	SWJ デバッグポート	797
図 268.	JTAG TAP 接続	801
図 269.	TPIU ブロック図	820

1 このマニュアルにおける表記の規則

1.1 レジスタに関する略記

レジスタの説明では、次の略記が使用されます。

読出し／ 書込み (rw)	これらのビットは、ソフトウェアによる読出しと書込みができます。
読出し専用 (r)	これらのビットは、ソフトウェアによる読出しのみが可能です。
書込み専用 (w)	このビットは、ソフトウェアによる書込みのみが可能です。ビットを読み出すと、リセット値が返されます。
読出し／ 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。
読出し／ 読出しによる 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。
読出し／ セット (rs)	このビットは、ソフトウェアによって読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。
読出し専用 書込み トリガ (rt_w)	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。“0”または“1”を書き込むと、イベントがトリガされますが、ビットの値は変化しません。
反転 (t)	このビットは、ソフトウェアによって“1”を書き込むことで反転だけです。“0”を書き込んでも、ビットの値は変化しません。
予約済み (Res.)	予約済みビットであり、リセット値に保持する必要があります。

1.2 用語

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

- CPU コアは、2 つのデバッグポートを内蔵しています。
 - JTAG デバッグポート (JTAG-DP) には、Joint Test Action Group (JTAG) のプロトコルに基づいた 5 ピンの標準インタフェースを搭載しています。
 - SWD デバッグポート (SWD-DP) には、Serial Wire Debug (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。
JTAG および SWD のプロトコルについては、FPU 搭載 Cortex[®]-M4 テクニカルリファレンスマニュアルを参照してください。
- ワード : 32 ビット長のデータ/命令。
- ハーフワード : 16 ビット長のデータ/命令。
- バイト : 8 ビット長のデータ。
- ダブルワード : 64 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。
- I-Code : このバスによって、CPU コアの命令バスがフラッシュメモリの命令インタフェースに接続されます。このバスを経由して、プリフェッチが行われます。
- D-Code : このバスによって、CPU の D-code バス (リテラルロードとデバッグアクセス) がフラッシュメモリのデータインタフェースに接続されます。
- 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spbスバス。
- CPU : FPU 搭載 Cortex[®]-M4 コアを参照。

1.3 使用可能なペリフェラル

さまざまな STM32F401xx シリーズで使用可能な各ペリフェラルとそのインスタンスの数については、STM32F401xB/C および STM32F401xD/E のデータシートを参照してください。

2 メモリとバスのアーキテクチャ

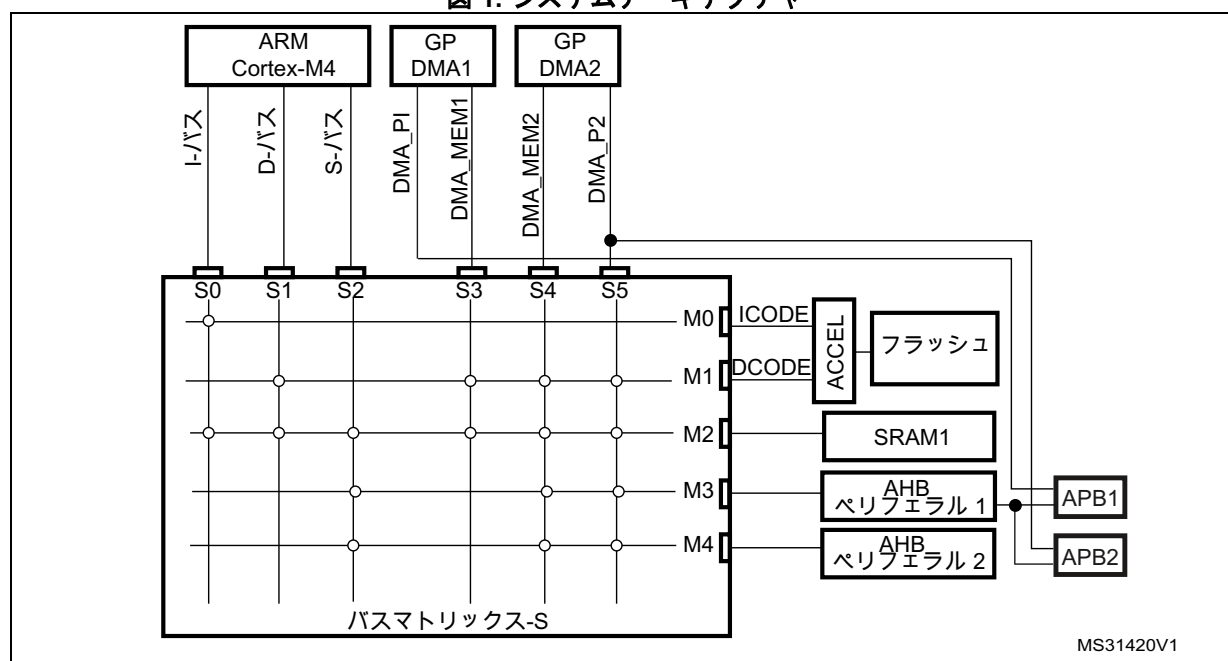
2.1 システムアーキテクチャ

STM32F401xB/C および STM32F401xD/E では、メインシステムは、以下のものを相互接続する 32 ビットマルチレイヤ AHB バスマトリックスで構成されています。

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

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

図 1. システムアーキテクチャ



1. STM32F401xB/C : 128 KB / 256 KB フラッシュと 64 KB SRAM を搭載。
STM32F401xD/E : 384 KB / 512 KB フラッシュと 96 KB SRAM を搭載。

2.1.1 I-バス

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

2.1.2 D-バス

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

2.1.3 S-バス

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

2.1.4 DMA メモリバス

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

2.1.5 DMA ペリフェラルバス

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

2.1.6 バスマトリックス

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

2.1.7 AHB-APB ブリッジ（APB）

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

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

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

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

2.2 メモリ構成

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ワードの最上位バイトとみなされます。

周辺回路のレジスタの詳細なマッピングについては、該当する章を参照してください。

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けられています。

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。該当する製品データシートのメモリマッピング図を参照してください。

2.3 メモリマップ

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。表 1 に、STM32F401xB/C および STM32F401xD/E のデバイスで使用可能なペリフェラルの境界アドレスを示します。

表 1. STM32F401xB/C および STM32F401xD/E のレジスタ境界アドレス

境界アドレス	ペリフェラル	バス	レジスタマップ
0x5000 0000 - 0x5003 FFFF	USB OTG FS	AHB2	セクション 22.16.6:743 ページのOTG_FS レジスタマップ
0x4002 6400 - 0x4002 67FF	DMA2	AHB1	セクション 9.5.11 : 196 ページのDMA レジスタマップ
0x4002 6000 - 0x4002 63FF	DMA1		
0x4002 3C00 - 0x4002 3FFF	フラッシュインタフェース レジスタ		セクション 3.8 : 60 ページのフラッシュインタフェース レジスタ
0x4002 3800 - 0x4002 3BFF	RCC		セクション 6.3.22 : 137 ページのRCC レジスタマップ
0x4002 3000 - 0x4002 33FF	CRC		セクション 4.4.4 : 70 ページのCRC レジスタマップ
0x4002 1C00 - 0x4002 1FFF	GPIOH		セクション 8.4.11 : 162 ページのGPIO レジスタマップ
0x4002 1000 - 0x4002 13FF	GPIOE		
0x4002 0C00 - 0x4002 0FFF	GIPOD		
0x4002 0800 - 0x4002 0BFF	GPIOC		
0x4002 0400 - 0x4002 07FF	GPIOB		
0x4002 0000 - 0x4002 03FF	GPIOA		



表 1. STM32F401xB/C および STM32F401xD/E のレジスタ境界アドレス（続き）

境界アドレス	ペリフェラル	バス	レジスタマップ
0x4001 4800 - 0x4001 4BFF	TIM11	APB2	セクション 14.5.12:411 ページのTIM10/11 レジスタマップ
0x4001 4400 - 0x4001 47FF	TIM10		
0x4001 4000 - 0x4001 43FF	TIM9		セクション 14.4.13 : 401 ページのTIM9 レジスタマップ
0x4001 3C00 - 0x4001 3FFF	EXTI		セクション 10.3.7 : 209 ページのEXTI レジスタマップ
0x4001 3800 - 0x4001 3BFF	SYSCFG		セクション 7.2.8 : SYSCFG レジスタマップ
0x4001 3400 - 0x4001 37FF	SPI4		セクション 20.5.10 : 604 ページのSPI レジスタマップ
0x4001 3000 - 0x4001 33FF	SPI1		
0x4001 2C00 - 0x4001 2FFF	SDIO		セクション 21.9.16 : 660 ページのSDIO レジスタマップ
0x4001 2000 - 0x4001 23FF	ADC1		セクション 11.12.16 : 239 ページのADC レジスタマップ
0x4001 1400 - 0x4001 17FF	USART6		セクション 19.6.8:550 ページのUSART レジスタマップ
0x4001 1000 - 0x4001 13FF	USART1		
0x4001 0000 - 0x4001 03FF	TIM1		セクション 12.4.21 : 308 ページのTIM1 レジスタマップ
0x4000 7000 - 0x4000 73FF	PWR	APB1	セクション 5.5 : 90 ページのPWR レジスタマップ
0x4000 5C00 - 0x4000 5FFF	I2C3		セクション 18.6.11 : 499 ページのI ² C のレジスタマップ
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 4400 - 0x4000 47FF	USART2		セクション 19.6.8:550 ページのUSART レジスタマップ
0x4000 4000 - 0x4000 43FF	I2S3ext		セクション 20.5.10 : 604 ページのSPI レジスタマップ
0x4000 3C00 - 0x4000 3FFF	SPI3/I2S3		
0x4000 3800 - 0x4000 3BFF	SPI2/I2S2		
0x4000 3400 - 0x4000 37FF	I2S2ext		セクション 15.4.5 : 417 ページのIWDG レジスタマップ
0x4000 3000 - 0x4000 33FF	IWDG		
0x4000 2C00 - 0x4000 2FFF	WWDG		セクション 16.6.4:424 ページのWWDG レジスタマップ
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers		セクション 17.6.21 : 463 ページのRTC レジスタマップ
0x4000 0C00 - 0x4000 0FFF	TIM5		セクション 13.4.21 : 367 ページのTIMx レジスタマップ
0x4000 0800 - 0x4000 0BFF	TIM4		
0x4000 0400 - 0x4000 07FF	TIM3		
0x4000 0000 - 0x4000 03FF	TIM2		

2.3.1 内蔵 SRAM

STM32F401xB/C デバイスには 64 KB、STM32F401xD/E には 96 KB のシステム SRAM を搭載しています。

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

CPU は、SRAM からのブートが選択されている場合、または物理的な再割当てが選択されている場合に、システムバスまたは I-Code／D-Code バスを通じて内蔵 SRAM にアクセスできます (SYSCFG コントローラの [セクション 7.2.1: SYSCFG メモリ再配置レジスタ \(SYSCFG_MEMRMP\)](#))。SRAM 実行のパフォーマンスを最大にするには、物理的な再割当てを選択します（ブートまたはソフトウェア選択）。

2.3.2 フラッシュメモリの概要

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

フラッシュメモリは、次のように構成されています。

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

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

2.3.3 ビットバンディング

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

STM32F4xx デバイスでは、ペリフェラルレジスタも SRAM もビットバンド領域に割り当てられますので、1 回のビットバンドの書込みと読出し操作が許可されます。操作は FPU 搭載 Cortex®-M4 アクセスでのみ可能であり、他のバスマスタ（DMA など）からはできません。

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

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

ここで、

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

例

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

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

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

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

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

2.4 ブート設定

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

STM32F4xx では、[表 2](#) に示すように、BOOT[1:0] ピンによって 3 種類のブートモードを選択できます。

表 2. ブートモード

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

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

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

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

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

内蔵ブートローダ

次のシリアルインタフェースのいずれかを使用してフラッシュメモリを再プログラムするには、内蔵ブートローダモードを使用します。

- USART1 (PA9/PA10)
- USART2 (PD5/PD6)
- I2C1 (PB6/PB7)
- I2C2 (PB10/PB3)
- I2C3 (PA8/PB4)
- SPI1 (PA4/PA5/PA6/PA7)
- SPI2 (PB12/PB13/PB14/PB15)
- SPI3 (PA15/PC10/PC11/PC12)
- デバイスモードで USB OTG フルスピード (PA11/12) (DFU : デバイスファームウェアアップグレード)

USART ペリフェラルは内部 16 MHz オシレータ (HSI) 周波数で動作しますが、USB OTG フルスピードは、1 MHz の倍数である外部クロック (HSE) (4~26 MHz) が必要です。

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

STM32F401xB/C および STM32F401xD/E の物理的な再割当て

ブートピンを選択したら、アプリケーションソフトウェアはコード領域でアクセス可能なメモリを変更できます (このようにするとシステムバスの代わりに ICode バスを使ってコードを実行できます)。この変更は、SYSCFG コントローラで [セクション 7.2.1: SYSCFG メモリ再配置レジスタ \(SYSCFG_MEMRMP\)](#) をプログラミングすることによって実行できます。

その後、次のメモリが再割当て可能となります。

- メインフラッシュメモリ
- システムメモリ
- 内蔵 SRAM1

表 3. メモリ割当てとブートモード/物理的な再割当て STM32F401xB/C

アドレス	メインフラッシュメモリでのブート/再割当て	内蔵 SRAM でのブート/再割当て	システムメモリでのブート/再割当て
0x2000 0000 - 0x2000 FFFF	SRAM1 (64 KB)	SRAM1 (64 KB)	SRAM1 (64 KB)
0x1FFF 0000 - 0x1FFF 77FF	システムメモリ	システムメモリ	システムメモリ
0x0804 0000 - 0x1FFE FFFF	予約済み	予約済み	予約済み
0x0800 0000 - 0x0803 FFFF	フラッシュメモリ	フラッシュメモリ	フラッシュメモリ
0x0400 000 - 0x07FF FFFF	予約済み	予約済み	予約済み
0x0000 0000 - 0x0003 FFFF ⁽¹⁾	フラッシュ (256 KB)、エイリアス	SRAM1 (64 KB)、エイリアス	システムメモリ (30 KB)、エイリアス

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

表 4. メモリ割当てとブートモード／物理的な再割当て STM32F401xD/E

アドレス	メインフラッシュメモリでの ブート／再割当て	内蔵 SRAM での ブート／再割当て	システムメモリでの ブート／再割当て
0x2000 0000 - 0x2001 7FFF	SRAM1 (96 KB)	SRAM1 (96KB)	SRAM1 (96KB)
0x1FFF 0000 - 0x1FFF 77FF	システムメモリ	システムメモリ	システムメモリ
0x0804 0000 - 0x1FFE FFFF	予約済み	予約済み	予約済み
0x0800 0000 - 0x0807 FFFF	フラッシュメモリ	フラッシュメモリ	フラッシュメモリ
0x0400 000 - 0x07FF FFFF	予約済み	予約済み	予約済み
0x0000 0000 - 0x0007 FFFF ⁽¹⁾	フラッシュ (512 KB)、 エイリアス	SRAM1 (96 KB)、 エイリアス	システムメモリ (30 KB)、 エイリアス

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

3 内蔵フラッシュメモリインタフェース

3.1 概要

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

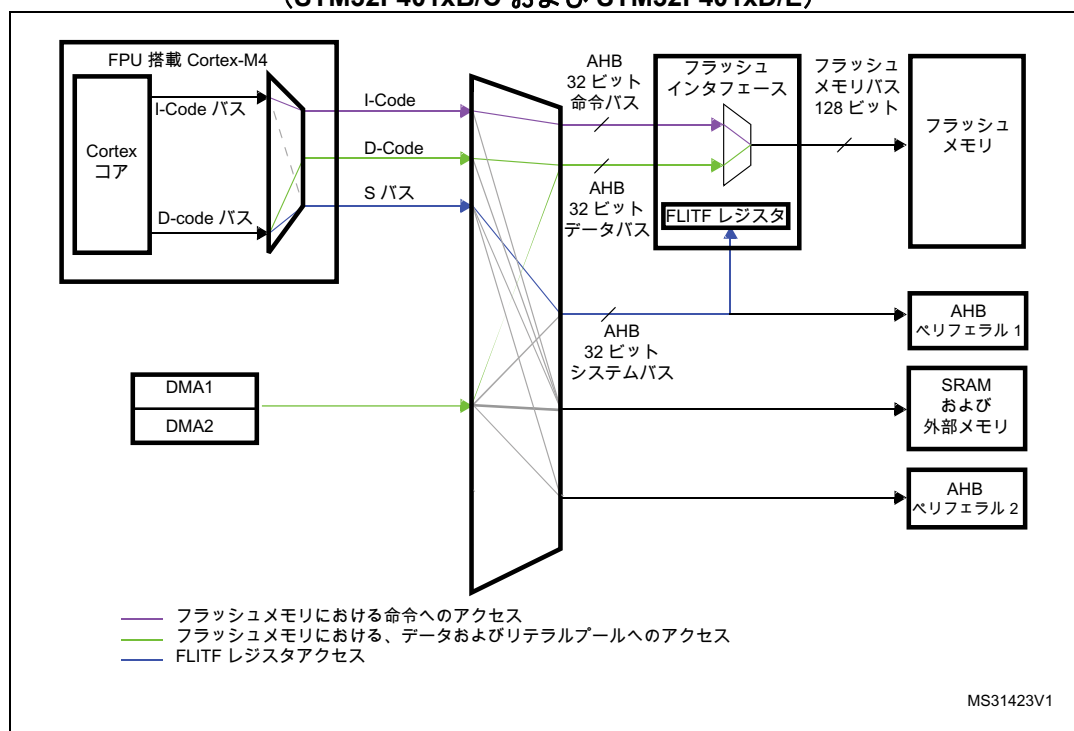
また、命令プリフェッチおよびキャッシュラインでコードの実行を加速します。

3.2 主な特長

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

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

図 2. システムアーキテクチャ内のフラッシュメモリインタフェース接続
(STM32F401xB/C および STM32F401xD/E)



3.3 内蔵フラッシュメモリ割込み（STM32F401xB/C および STM32F401xD/E）

フラッシュメモリは、以下のような主な特徴を備えています。

- STM32F401xB/C では最大容量 256 KB、STM32F401xD/E では最大容量 512 KB
- 128 ビット幅のデータ読出し
- バイト、ハーフワード、ワード、ダブルワードでの書込み
- セクタ消去と全体消去
- メモリ構成

フラッシュメモリは、次のように構成されています。

- 16 KB のセクタ 4 つ、64 KB のセクタ 1 つ、128 KB のセクタ 1 つ（STM32F401xB/C）、または 128 KB のセクタ 3 つ（STM32F401xD/E）に分割されたメインメモリブロック
- システムメモリブートモードでデバイスがブートするシステムメモリ
- ユーザデータ用の 512 バイト OTP（one-time programmable）
OTP エリアには、該当する OTP データブロックをロックするための追加の 16 バイトがあります。
- デバイスが STANDBY または STOP モードのときに読出し／書込み保護、BOR レベル、ウォッチドッグソフトウェア／ハードウェアおよびリセットを設定するオプションバイト
- 低電力モード（詳細についてはリファレンスマニュアルの電源管理(PWR)のセクションを参照）

表 5. フラッシュモジュールの構成（STM32F401xB/C および STM32F401xD/E）

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

3.4 読出しインタフェース

3.4.1 CPU クロック周波数とフラッシュメモリ読出し時間との関係

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

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

- VOS[1:0] = "0x01" のとき、 f_{HCLK} の最大値 = 60 MHz

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

表 6. CPU クロック（HCLK）周波数によるウェイトステート数

ウェイトステート (WS) (LATENCY)	HCLK (MHz)			
	電圧範囲 2.7 V - 3.6 V	電圧範囲 2.4 V - 2.7 V	電圧範囲 2.1 V - 2.4 V	電圧範囲 1.71 V - 2.1 V
0 WS (1 CPU サイクル)	$0 < HCLK \leq 30$	$0 < HCLK \leq 24$	$0 < HCLK \leq 18$	$0 < HCLK \leq 16$
1 WS (2 CPU サイクル)	$30 < HCLK \leq 60$	$24 < HCLK \leq 48$	$18 < HCLK \leq 36$	$16 < HCLK \leq 32$
2 WS (3 CPU サイクル)	$60 < HCLK \leq 84$	$48 < HCLK \leq 72$	$36 < HCLK \leq 54$	$32 < HCLK \leq 48$
3 WS (4 CPU サイクル)		$72 < HCLK \leq 84$	$54 < HCLK \leq 72$	$48 < HCLK \leq 64$
4 WS (5 CPU サイクル)	-	-	$72 < HCLK \leq 84$	$64 < HCLK \leq 80$
5 WS (6 CPU サイクル)	-	-	-	$80 < HCLK \leq 84$

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

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

CPU 周波数の増加

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

CPU 周波数の減少

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

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

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

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

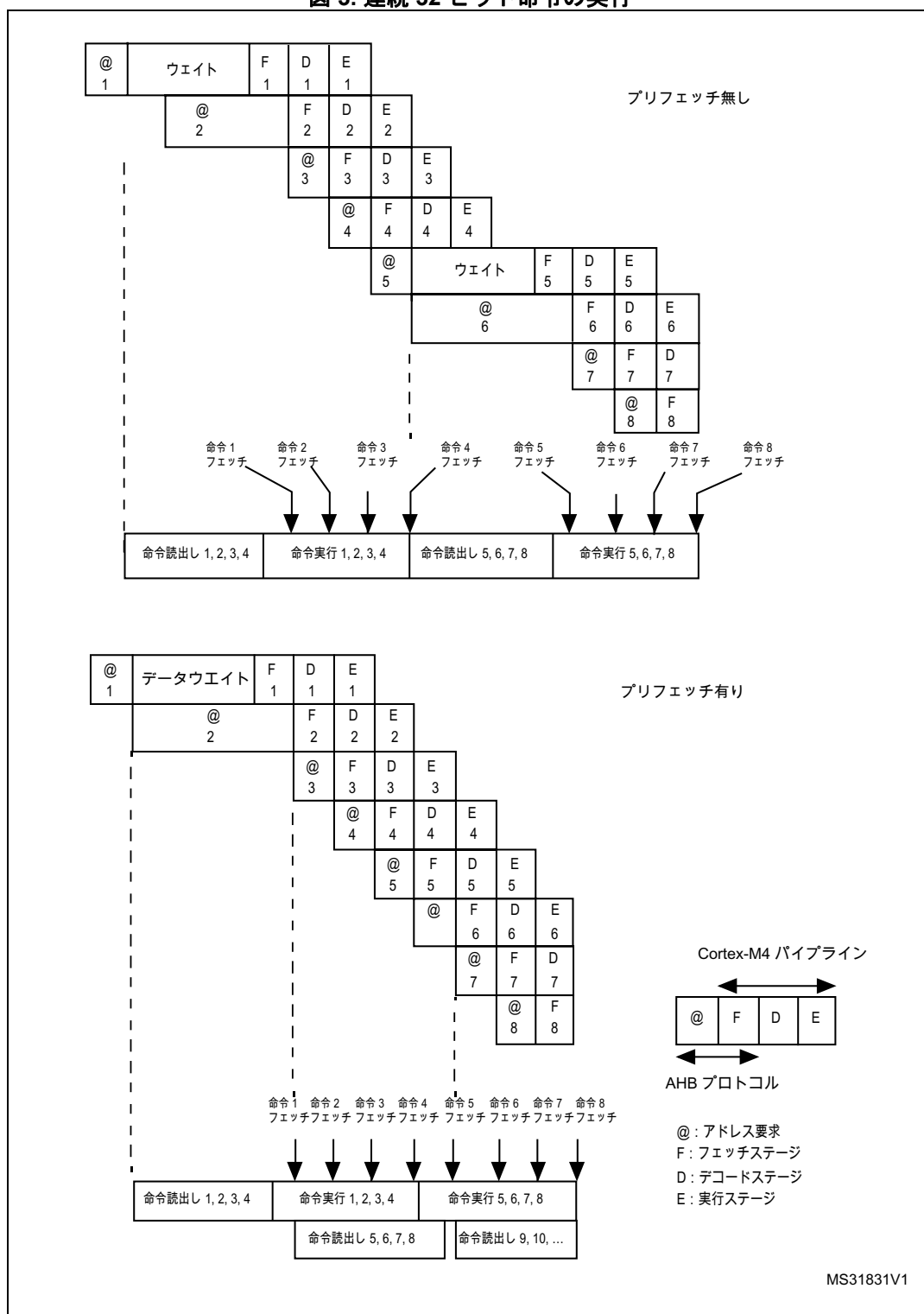
プロセッサのパフォーマンスを最大限に引き出すため、アクセラレータは128 ビットのフラッシュメモリからのプログラム実行速度を上昇させる命令プリフェッチキューと、ブランチキャッシュを搭載しています。CoreMark ベンチマークによると、ART アクセラレータによって達成されるパフォーマンスは、最大 CPU 周波数 84 MHz におけるフラッシュメモリからの 0 ウェイトステートプログラム実行と同等となります。

命令プリフェッチ

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

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

図 3. 連続 32 ビット命令の実行



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

命令キャッシュメモリ

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

データ管理

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

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

注： ユーザー設定セクタのデータは、キャッシュできません。

3.5 消去操作とプログラム操作

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

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

3.5.1 フラッシュ制御レジスタのアンロック

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

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

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

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

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

3.5.2 プログラム/消去の並列処理

並列処理サイズは、FLASH_CR レジスタの PSIZE フィールドで設定します。これはフラッシュメモリに書込み操作が発生するたびにプログラムされるバイト数を表します。PSIZE は、供給電圧や、外

部 V_{pp} 電源が使用されているかどうかによって制限されます。したがって、プログラム／消去操作の前に FLASH_CR レジスタで正しく設定しておく必要があります。

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

表 7 は、正しい PSIZE 値を記載しています。

表 7. プログラム／消去の並列処理

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

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

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

3.5.3 消去

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

セクタ消去

セクタを消去するには、次の手順に従います。

- FLASH_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
- FLASH_CR レジスタで SER ビットをセットし、メインメモリブロックで 5 個（STM32F401xB/C の場合）および 7 個（STM32F401xD/E の場合）あるセクタの中から消去するセクタ（SNB）を選択します。
- FLASH_CR レジスタの STRT ビットをセットします。
- BSY ビットがクリアされるのを待ちます。

全体消去

全体消去の実施には、次のシーケンスを推奨します。

- 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 MER 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 ビットがクリアされるのを待ちます。

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

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

3.5.4 プログラミング

標準プログラミング

フラッシュメモリのプログラミングシーケンスは、次のようになっています。

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

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

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

プログラミングエラー

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

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

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

プログラミングとキャッシュ

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

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

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

3.5.5 割込み

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

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

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

この場合、FLASH_SR レジスタでエラー割込み有効ビット (ERRIE) がセットされると、割込みが生成され、FLASH_SR レジスタで操作エラービット (OPERR) がセットされます。

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

表 8. フラッシュ割込みリクエスト

割込みイベント	イベントフラグ	有効制御ビット
操作終了	EOP	EOPIE
書き込み保護エラー	WRPERR	ERRIE
プログラミングエラー	PGAERR, PGPERR, PGSERR	ERRIE

3.6 オプションバイト

3.6.1 ユーザオプションバイトの説明

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

表 9. オプションバイトの構成

アドレス	[63:16]	[15:0]
0x1FFF C000	予約済み	ROP およびユーザオプションバイト (RDP & USER)
0x1FFF C008	予約済み	書き込み保護 nWRP ビット (セクタ 5 (STM32F401xB/C の場合))、(セクタ 0 から 7 (STM32F401xD/E の場合))

表 10. オプションバイトの説明

オプションバイト (ワード、アドレス 0x1FFF C000)	
RDP : 読出し保護オプションバイトです。 読出し保護は、フラッシュメモリに格納されているソフトウェアコードの保護に使用します。	
ビット 15:8	0xAA : レベル 0、保護なし 0xCC : レベル 2、チップ保護 (デバッグと RAM からのブート機能は無効) その他 : レベル 1、メモリの読出し保護 (デバッグ機能は制限付き)

表 10. オプションバイトの説明（続き）

USER : ユーザオプションバイト このバイトは次の機能の設定に使用します。 ウォッチドッグイベントの選択 : ハードウェアまたはソフトウェア STOP モードに入るときのリセットイベント STANDBY モードに入るときのリセットイベント	
ビット 7	nRST_STDBY 0 : STANDBY モードに入るときにリセットを生成します。 1 : リセットは生成されません。
ビット 6	nRST_STOP 0 : STOP モードに入るときにリセットを生成します。 1 : リセットは生成されません。
ビット 5	WDG_SW 0 : ハードウェアウォッチドッグ 1 : ソフトウェアウォッチドッグ
ビット 4	0x0 : 使用しません。STM32F401xB/C では、このビットのデフォルト値"0"から変更しないでください。
ビット 3:2	BOR_LEV : BOR リセットレベル このビットには、リセットをアクティブにしたりリセットを解除したりする供給レベル閾値が含まれています。 フラッシュメモリに新しい BOR レベルをプログラムするように書き込むことができます。 00 : BOR レベル 3 (VBOR3)、ブラウンアウト閾値レベル 3 01 : BOR レベル 2 (VBOR2)、ブラウンアウト閾値レベル 2 10 : BOR レベル 1 (VBOR1)、ブラウンアウト閾値レベル 1 11 : BOR オフ、POR/PDR リセット閾値レベルが適用されます。 BOR 特性の詳細については、製品データシートの「電気的特性」のセクションを参照してください。
ビット 1:0	0x1 : 未使用
オプションバイト（ワード、アドレス 0x1FFF C008）	
ビット 15	SPRMOD : nWPRi ビットの保護モードの選択 0 : nWPRi ビットをセクタ i の書き込み保護に使用します（デフォルト）。 1 : nWPRi ビットをセクタ i の PCROP 保護に使用します（セクタ）。
ビット 14:6	予約済み
nWRP : フラッシュメモリ書き込み保護オプションバイト セクタ 0 から 5 まで (STM32F401xB/C)、セクタ 0 から 7 まで (STM32F401xD/E) が書き込み保護可能です。	
ビット 5:0	nWRPi SPRMOD がリセットされている場合（デフォルト値） : 0 : セクタ i で書き込み保護がアクティブになっています。 1 : セクタ i で書き込み保護がアクティブになっていません。 SPRMOD がセットされている場合（アクティブ） : 0 : セクタ i で PCROP 保護がアクティブになっていません。 1 : セクタ i で PCROP 保護がアクティブになっています。

3.6.2 ユーザオプションバイトのプログラミング

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

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

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

ユーザオプションバイトの変更

ユーザオプション値を変更するには、次のシーケンスに従います。

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

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

3.6.3 読出し保護 (RDP)

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

- レベル 0：読出し保護なし

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

- レベル 1：読出し保護有効

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

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

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

- レベル 2：デバッグ／チップ読出し無効

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

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

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

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

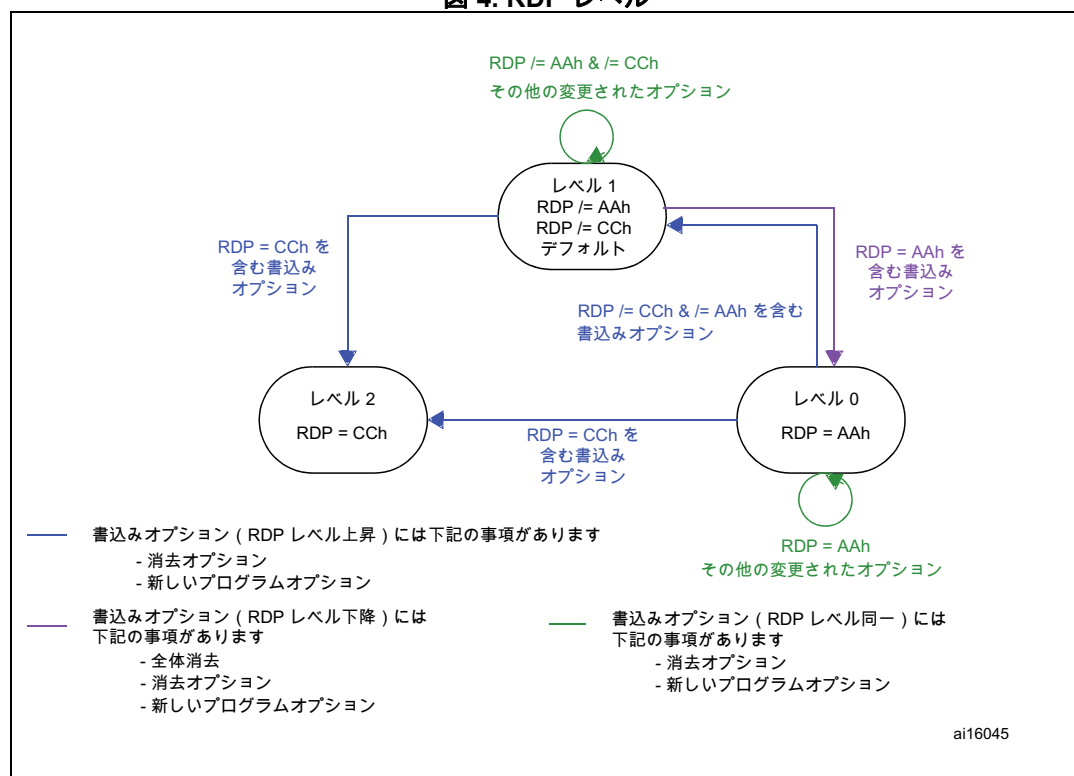
表 11. アクセスと読出し保護レベル

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

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

図 4 は、ある RDP レベルから別の RDP レベルへの遷移を示しています。

図 4. RDP レベル



3.6.4 書き込み保護

フラッシュメモリでは、最大 5 個 (STM32F401xB/C の場合) または最大 7 個 (STM32F401xD/E の場合) までのユーザセクタをプログラムカウンタコンテキストの損失による不要な書き込み操作から保護できます。書き込み保護 nWRPi ビット ($0 \leq i \leq 7$) が FLASH_OPTCR レジスタでローのときは、該当するセクタを消去したりプログラムしたりすることができません。結果として、セクタが 1 つ書き込み保護されていると全体消去が実施できません。

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

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

書き込み保護エラーフラグ

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

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

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

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

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

3.6.5 独自仕様コード読出し保護（PCROP）

フラッシュメモリユーザセクタ（0 から 5（STM32F401xB/C の場合）または最大 7 まで（STM32F401xD/E の場合））は、独自仕様の読出し保護（PCROP）を使って D-バス読出しアクセスから保護することができます。

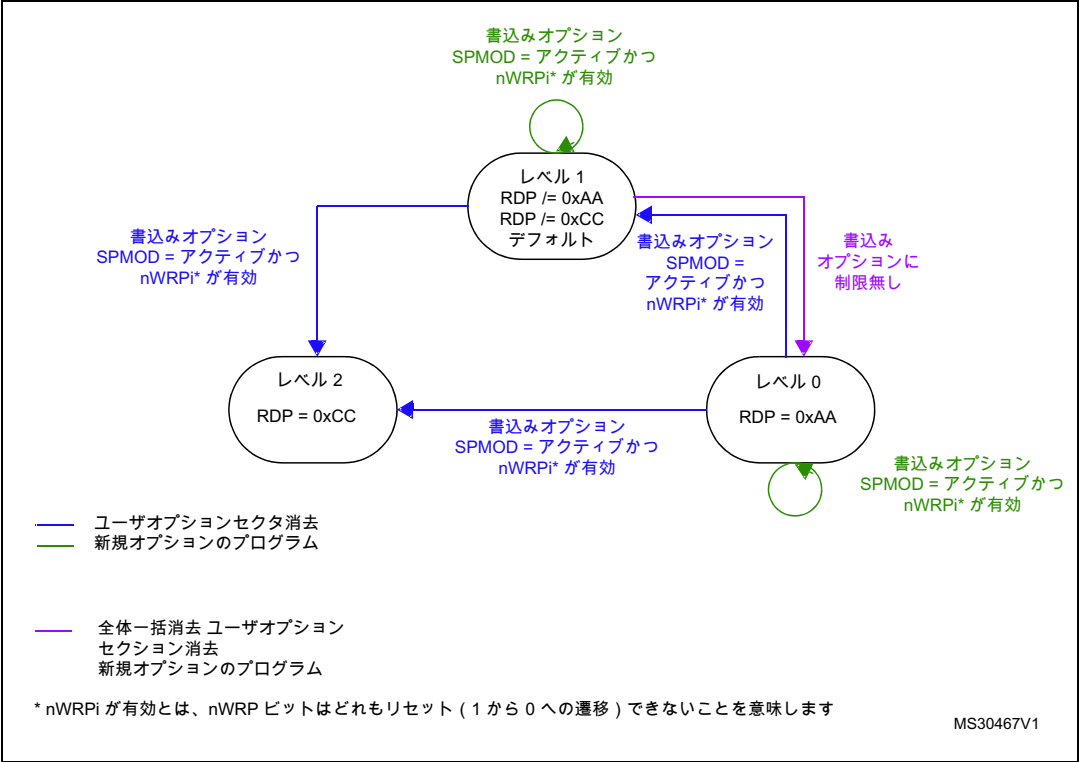
PCROP 保護は、FLASH_CR レジスタの SPRMOD オプションビットを使って次のように選択されます。

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

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

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

図 5. PCROP レベル



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

注： PCROP モードがアクティブ (SPRMOD = 1) のときは、nWRPi ビットのアクティブな値が反転します。SPRMOD = 1 で nWRPi = 1 のときは、バンク 1 のユーザセクタ i、バンク 2 のユーザセクタ i がそれぞれ読出し／書込み保護 (PCROP) されます。

3.7 OTP (One-time programmable) バイト

表 12 は、OTP エリアの構成を示しています。

表 12. OTP エリアの構成

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

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

3.8 フラッシュインタフェースレジスタ

3.8.1 フラッシュアクセス制御レジスタ (FLASH_ACR)

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

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

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

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

ビット 31:13 予約済み、クリア状態を保つ必要があります。

ビット 12 **DCRST** : データキャッシュのリセット

0 : データキャッシュはリセットされません。

1 : データキャッシュはリセットされます。

このビットに書き込めるのは、D キャッシュが無効のときだけです。

ビット 11 **ICRST** : 命令キャッシュのリセット

0 : 命令キャッシュはリセットされません。

1 : 命令キャッシュはリセットされます。

このビットに書き込めるのは、I キャッシュが無効のときだけです。

ビット 10 **DCEN** : データキャッシュ有効

0 : データキャッシュは無効です。

1 : データキャッシュは有効です。

ビット 9 **ICEN** : 命令キャッシュ有効

0 : 命令キャッシュは無効です。

1 : 命令キャッシュは有効です。

ビット 8 **PRFTEN** : プリフェッチ有効化

0 : プリフェッチは無効です。

1 : プリフェッチは有効です。

ビット 7:4 予約済み、クリア状態を保つ必要があります。

ビット 3:0 **LATENCY** : 遅延

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

0000 : ウェイトステート 0 個

0001 : ウェイトステート 1 個

0010 : ウェイトステート 2 個

-

-

-

1110 : ウェイトステート 14 個

1111 : ウェイトステート 15 個

3.8.2 フラッシュキーレジスタ (FLASH_KEYR)

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

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードアクセス

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

ビット 31:0 **FKEYR** : FPEC キー

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

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

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

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

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワードアクセス

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

ビット 31:0 **OPTKEYR** : オプションバイトキー

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

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

3.8.4 フラッシュステータスレジスタ (FLASH_SR)

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

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

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

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

ビット 31:17 予約済み、クリア状態を保つ必要があります。

ビット 16 **BSY** : ビジー

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

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

ビット 15:9 予約済み、クリア状態を保つ必要があります。

ビット 8 **RDERR** : 読出し保護エラー (pcrop)

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

ビット 7 **PGSERR** : プログラミングシーケンスエラー

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

ビット 6 **PGPERR** : プログラミング並列処理エラー

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

ビット 5 **PGAERR** : プログラミング配置エラー

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

ビット 4 **WRPERR** : 書込み保護エラー

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

ビット 3:2 予約済み、クリア状態を保つ必要があります。

ビット 1 **OPERR** : 操作エラー

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

ビット 0 **EOP** : 操作終了

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

3.8.5 フラッシュ制御レジスタ（FLASH_CR）

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

アドレスオフセット : 0x10

リセット値 : 0x8000 0000

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

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

ビット 31 **LOCK** : ロック状態

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

ビット 30:26 予約済み、クリア状態を保つ必要があります。

ビット 25 **ERRIE** : エラー割込みイネーブル

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

ビット 24 **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 23:17 予約済み、クリア状態を保つ必要があります。

ビット 16 **STRT** : スタート

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

ビット 15:10 予約済み、クリア状態を保つ必要があります。

ビット 9:8 **PSIZE** : プログラムサイズ

このビットは、プログラムの並列処理を選択します。

00 プログラム x8

01 プログラム x16

10 プログラム x32

11 プログラム x64

ビット 7 予約済み、クリア状態を保つ必要があります。

ビット 6:3 **SNB** : セクタ番号

このビットは、消去するセクタを選択します。

0000 セクタ 0

0001 セクタ 1

...

0101 セクタ 5

0110 セクタ 6 (STM32F401xD/E デバイスのみ)

0111 セクタ 7 (STM32F401xE デバイスのみ)

1000 設定禁止

...

1011 設定禁止

1100 ユーザ固有セクタ

1101 ユーザ設定セクタ

1110 設定禁止

1111 設定禁止

ビット 2 **MER** : 全体消去

すべてのユーザセクタに対して消去がアクティブになります。

ビット 1 **SER** : セクタ消去

セクタ消去がアクティブになります。

ビット 0 **PG** : プログラミング

フラッシュプログラミングがアクティブになります。

3.8.6 フラッシュオプション制御レジスタ (FLASH_OPTCR)

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

アドレスオフセット : 0x14

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPR MOD	予約済み							nWRP[7:6] ⁽¹⁾		nWRP[5:0]					
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP[7:0]								nRST_STDBY	nRST_STOP	WDG_SW	予約済み	BOR_LEV		OPTSTRT	OPTLOCK
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rs	rs

1. STM32F401xD/E デバイスでのみ使用できます。

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

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

ビット 30:24 予約済み、クリア状態を保つ必要があります。

ビット 23:22 **nWRP[7:6]** : 書き込み保護なし

- STM32F401xB/C の場合 :
予約済み、クリア状態を保つ必要があります。
- STM32F401xD/E の場合 :
このビットには、リセット後のセクタ 6 および 7 の書き込み保護および読出し保護 (PCROP) オプションバイトの値が含まれています。(詳細については、ビット 21:16 を参照してください。

ビット 21:16 **nWRP[5:0]** : 書き込み保護なし

- このビットには、リセット後のセクタの書き込み保護オプションバイトの値が含まれています。フラッシュメモリに新しい書き込み保護値をプログラムするよう書き込むことができます。
- 0 : 選択したセクタで書き込み保護がアクティブになっています。
- 1 : 選択したセクタで書き込み保護がアクティブになっていません。
- このビットには、リセット後のセクタ 0 から 5 の書き込み保護および読出し保護 (PCROP) オプションバイトの値が含まれています。フラッシュメモリに新しい書き込み保護または PCROP 値をプログラムするよう書き込むことができます。
- SPRMOD がリセットされている場合
 - 0 : セクタ i で書き込み保護がアクティブになっています。
 - 1 : セクタ i で書き込み保護がアクティブになっていません。
- SPRMOD がセットされている場合
 - 0 : セクタ i で PCROP 保護がアクティブになっていません。
 - 1 : セクタ i で PCROP 保護がアクティブになっています。

ビット 15:8 **RDP** : 読出し保護

- このビットには、リセット後の読出し保護オプションレベルの値が含まれています。フラッシュメモリに新しい読出し保護値をプログラムするよう書き込むことができます。
- 0xAA : レベル 0、読出し保護はアクティブではありません。
- 0xCC : レベル 2、チップ読出し保護がアクティブです。
- その他 : レベル 1、メモリの読出し保護がアクティブです。

ビット 7:5 USER : ユーザオプションバイト

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

ビット 7 : nRST_STDBY

ビット 6 : nRST_STOP

ビット 5 : WDG_SW

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

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

ビット 3:2 BOR_LEV : BOR リセットレベル

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

00 : BOR レベル 3 (VBOR3)、ブラウンアウト閾値レベル 3

01 : BOR レベル 2 (VBOR2)、ブラウンアウト閾値レベル 2

10 : BOR レベル 1 (VBOR1)、ブラウンアウト閾値レベル 1

11 : BOR オフ、POR/PDR リセット閾値レベルが適用されます。

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

ビット 1 OPTSTRT : オプション開始

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

ビット 0 OPTLOCK : オプションロック

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

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

3.8.7 フラッシュインタフェースレジスタマップ

表 13. フラッシュレジスタマップとリセット値

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

4 CRC 計算ユニット

4.1 CRC の概要

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

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

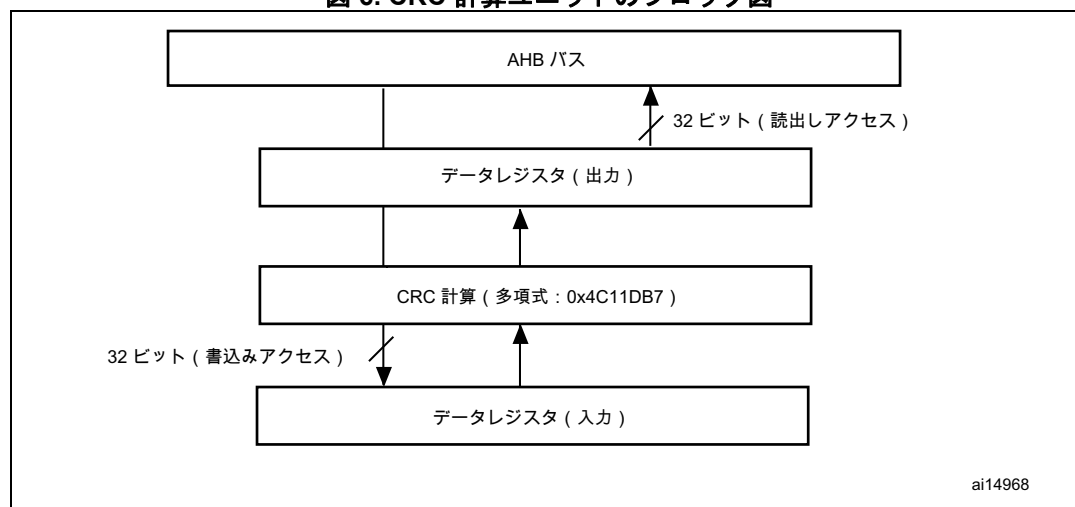
4.2 CRC の主な機能

- 以下のCRC-32 (Ethernet) 多項式を使用します。0x4C11DB7

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

図 6 にブロック図を示します。

図 6. CRC 計算ユニットのブロック図



4.3 CRC の機能説明

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

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

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

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

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

4.4 CRC レジスタ

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

4.4.1 データレジスタ（CRC_DR）

アドレスオフセット：0x00

リセット値：0xFFFF FFFF

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

ビット 31:0 **データレジスタビット**

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

4.4.2 独立型データレジスタ（CRC_IDR）

アドレスオフセット：0x04

リセット値：0x0000 0000

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **汎用 8 ビットデータレジスタビット**

1 バイトの一時的なストレージとして使用できます。
このレジスタは、CRC_CR レジスタの RESET ビットによって生成される CRC リセットの影響を受けません。

4.4.3 制御レジスタ（CRC_CR）

アドレスオフセット：0x08

リセット値：0x0000 0000

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

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **RESET** ビット

CRC 計算ユニットをリセットして、データレジスタを 0xFFFF FFFF にセットします。
このビットはセットのみができ、ハードウェアによって自動的にクリアされます。

4.4.4 CRC レジスタマップ

次の表に、CRC レジスタマップとリセット値を示します。

表 14. CRC 計算ユニットのレジスタマップとリセット値

オフ セット	レジスタ	31-24	23-16	15-8	7	6	5	4	3	2	1	0
0x00	CRC_DR	データレジスタ										
	リセット値	0xFFFF FFFF										
0x04	CRC_IDR	予約済み			独立型データレジスタ							
	リセット値				0x00							
0x08	CRC_CR	予約済み										RESET
	リセット値											0

5 電源コントローラ (PWR)

5.1 電源

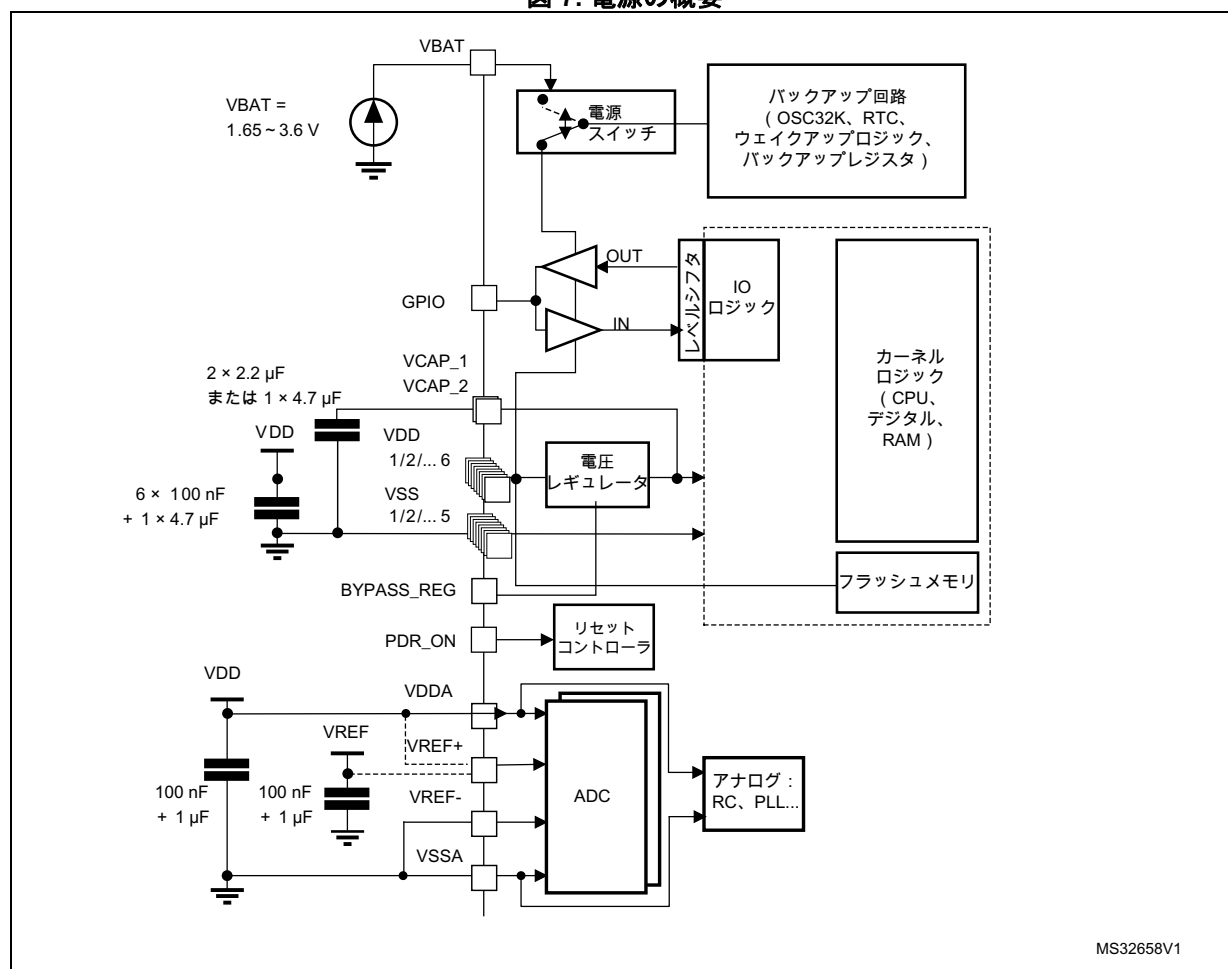
主電源の供給方法は 2 種類あります。

- $V_{DD} = 1.7 \sim 3.6 \text{ V}$: 無効化された内蔵レギュレータと I/O 用外部電源供給で、 V_{DD} ピンを介した外部供給です。 V_{DD} および PDR_ON ピンに接続された 外部電源供給スーパーバイザを使用する必要があります。
- $V_{DD} = 1.8 \sim 3.6 \text{ V}$: 内蔵レギュレータ (有効の場合) と I/O 用外部電源供給で、 V_{DD} ピンを介した外部供給です。

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

注 : 動作電源範囲に応じて、一部のペリフェラルは限定された機能と性能で使用されます。詳細については、STM32F4xx データシートの「一般動作条件」のセクションを参照してください。

図 7. 電源の概要



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

5.1.1 独立した A/D コンバータ用電源と基準電圧

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

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

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

5.1.2 バッテリバックアップドメイン

バックアップドメインの説明

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

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

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

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

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

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

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

- PC14 と PC15 は、GPIO または LSE ピンとして使用できます。
- PC13 を GPIO として使用したり、追加機能を設定することができます (このピン設定の詳細は [表 26 : RTC 追加機能](#) を参照してください)。

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

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

- PC14 と PC15 は、LSE ピンとしてのみ使用できます。
- PC13 を RTC 追加機能ピンとして使用できます (このピン設定の詳細は表 26 : RTC 追加機能を参照してください)。

バックアップドメインアクセス

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

- RTC および RTC バックアップレジスタへのアクセス
1. RCC_APB1ENR レジスタの PWREN ビットをセットして、電源インタフェースクロックを有効にします (セクション 6.3.11 : RCC APB1 ペリフェラルクロック有効レジスタ (RCC_APB1ENR) を参照)。
 2. セクション 5.4.1 の DBP ビットをセットして、バックアップドメインへのアクセスを有効にします。
 3. RTC クロックソースを選択します : セクション 6.2.8 : RTC/AWU クロックを参照。
 4. セクション 6.3.17 : RCC バックアップドメイン制御レジスタ (RCC_BDCR) の RTCEN [15] ビットをプログラムすることで、RTC クロックを有効にします。

RTC および RTC バックアップレジスタ

本リアルタイムクロック (RTC) は、独立した BCD タイマ/カウンタです。この RTC には、時刻クロック/カレンダー、プログラム可能なアラーム割込みを 2 本、割込み機能を備えたプログラム可能な周期的ウェイクアップフラグを搭載しています。この RTC には、20 個のバックアップデータレジスタ (80 バイト) を搭載しており、タンパ検出イベントが発生するとリセットされます。詳細については、セクション 17 : リアルタイムクロック (RTC) を参照してください。

5.1.3 電圧レギュレータ

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

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

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

- **RUN モード**では、レギュレータは 1.2 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。このモードでは、レギュレータ出力電圧 (約 1.2 V) をソフトウェアによって異なる電圧値にスケールリングできます。

スケール 2 またはスケール 3 が PWR_CR レジスタの VOS[1:0] ビットを介して設定できます (スケール 1 はこの製品では使用しません)。リセット後、VOS レジスタはスケール 2 にセットされます。PLL がオフの場合、電圧レギュレータは、VOS レジスタの内容に関係なく、スケール 3 にセットされます。VOS レジスタの内容は、PLL が有効化され、HSI または HSE がクロックソースとして選択されている場合のみ考慮されます。

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

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

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

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

注 : 詳細については、STM32F401xB/C データシートの電圧レギュレータのセクションを参照してください。

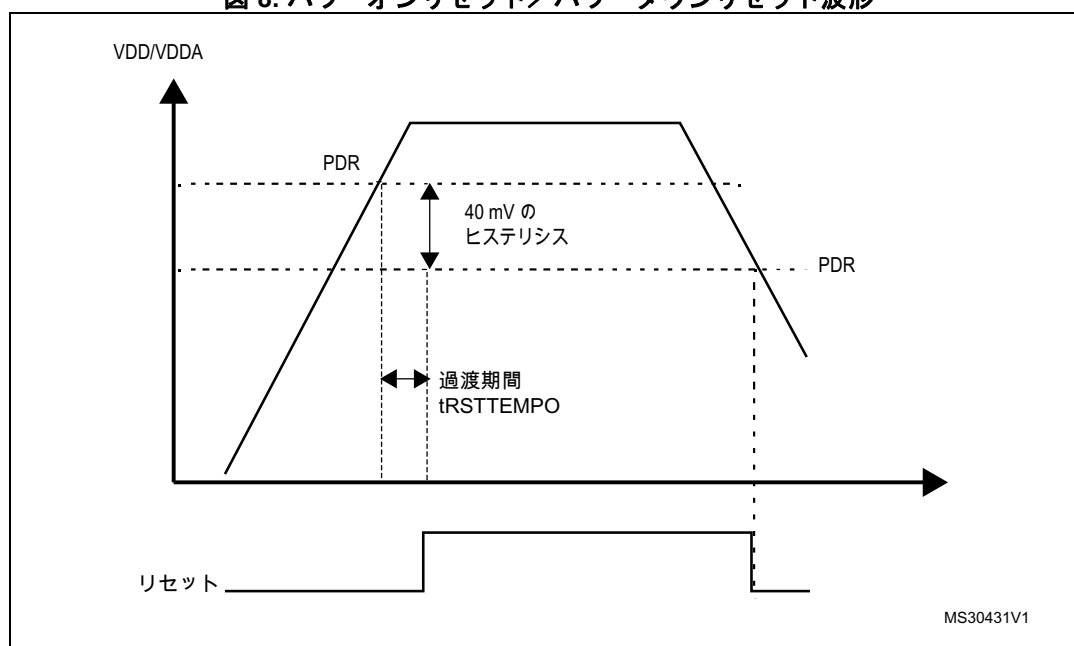
5.2 電源供給スーパバイザ

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

このデバイスには、POR/PDR 統合回路が搭載されています。

1.8 V 未満のデバイスを使用するには、PDR_ON ピンを使用して内部電源供給スーパバイザをオフにする必要があります (STM32F401xB/C データシートの電源供給スーパバイザのセクションを参照)。デバイスは、 V_{DD}/V_{DDA} が規定の閾値 $V_{POR/PDR}$ を下回っている間は、外部のリセット回路を必要とせずに、リセットモードを維持します。POR/PDR 閾値の詳細については、データシートの電気的特性の項を参照してください。

図 8. パワーオンリセット/パワーダウンリセット波形



5.2.2 ブラウンアウトリセット (BOR)

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

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

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

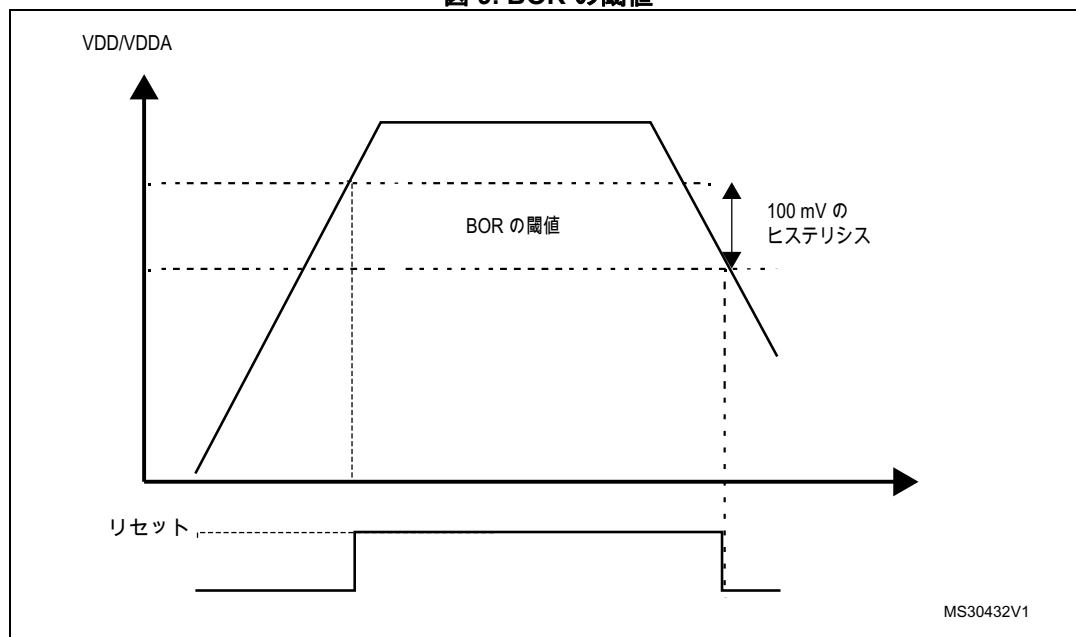
注： BOR 特性の詳細については、デバイスデータシートの「電気的特性」のセクションを参照してください。

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

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

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

図 9. BOR の閾値



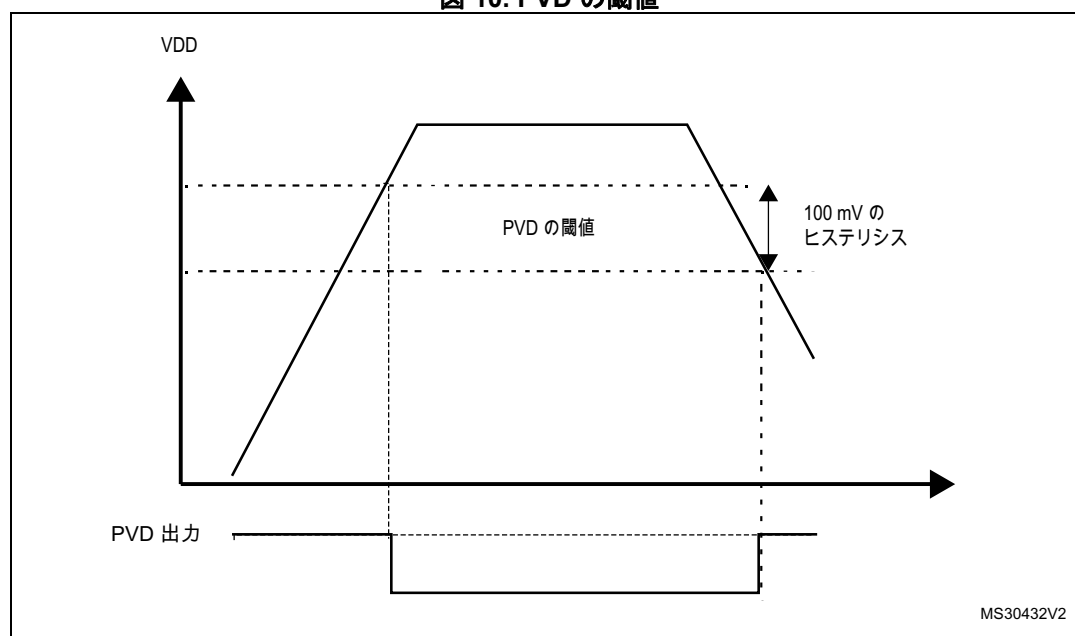
5.2.3 プログラム可能な電圧検出器 (PVD)

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

PVD は PVDE ビットをセットすることで有効になります。

PWR 電源制御/ステータスレジスタ (PWR_PWR) の PVDO フラグを使用して、 V_{DD} が PVD の閾値よりも高いか低いか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 10. PVD の閾値



5.3 低電力モード

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

このデバイスは、次の 3 つの低電力モードを備えています。

- SLEEP モード (FPU 搭載 Cortex®-M4 コアは停止、ペリフェラルは動作状態を維持)
- STOP モード (すべてのクロックが停止)
- STANDBY モード (1.2 V ドメインの電源オフ)

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

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

低電力モードへの移行

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

低電力モードの終了

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

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

- NVIC IRQ 割込み :

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

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

- イベント

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

MCU は、外部リセット (NRST ピン)、IWDG リセット、有効な WKUPx ピンのうちの 1 つの立ち上がりエッジ、または RTC イベントのいずれかが発生すると、低電力の STANDBY モードを終了します (図 159 : RTC ブロック図を参照)。

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

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

表 15. 低電力モードの概要

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

5.3.1 システムクロックの低速化

RUN モードでは、プリスケアラ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、PCLK1、PCLK2) 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 6.3.3: RCC クロック設定レジスタ \(RCC_CFGR\)](#) を参照してください。

5.3.2 ペリフェラルクロックゲーティング

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

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

ペリフェラルへのクロックゲーティングは、AHB1 ペリフェラルクロック有効レジスタ (RCC_AHB1ENR)、AHB2 ペリフェラルクロック有効レジスタ (RCC_AHB2ENR) によって制御されます ([セクション 6.3.9: RCC AHB1 ペリフェラルクロック有効レジスタ \(RCC_AHB1ENR\)](#)、[セクション 6.3.10: RCC AHB2 ペリフェラルクロック有効レジスタ \(RCC_AHB2ENR\)](#)、を参照)。

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

5.3.3 SLEEP モード

SLEEP モードへの移行

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

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

SLEEP モードの終了

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

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

表 16. Sleep-now への移行と終了

Sleep-now モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0 FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
	次の条件下での割り込みサービスルーチン (ISR) 終了時 – SLEEPDEEP = 0 および – SLEEPONEXIT = 1 FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
モードの終了	WFI または ISR 終了を使用して移行した場合 割り込み：参照先： 表 38：STM32F401xB/CSTM32F401xD/Eのベクタテーブル WFE 命令を使用して移行し、SEVONPEND = 0 の場合： ウェイクアップイベント： セクション 10.2.3: ウェイクアップイベント管理 を参照してください。 WFE 命令を使用して移行し、SEVONPEND = 1 の場合： 割り込みイベント (NVIC で無効化されている場合も含む)： 表 38: STM32F401xB/CSTM32F401xD/Eのベクタテーブル またはウェイクアップイベント (セクション 10.2.3: ウェイクアップイベント管理 を参照)を参照してください。
ウェイクアップ遅延時間	なし

表 17. Sleep-on-exit への移行と終了

Sleep-on-exit	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0 FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
	次の条件下での割り込みサービスルーチン (ISR) 終了時 – SLEEPDEEP = 0 および – SLEEPONEXIT = 1 FPU 搭載 Cortex®-M4 システム制御レジスタを参照。
モードの終了	割り込み：参照先： 表 38：STM32F401xB/CSTM32F401xD/Eのベクタテーブル
ウェイクアップ遅延時間	なし

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

PWR_CR レジスタの一部の設定によって、電力消費をさらに低減できます。フラッシュメモリが低電力モードで動作している場合、STOP モードからのウェイクアップの際に、さらなる起動時間が必要になります (表 18 : STOPモードの動作モードおよびセクション 5.4.1 : PWR 電源制御レジスタ (PWR_CR) を参照)。

表 18. STOPモードの動作モード

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

STOP モードへの移行

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

STOP モードへの移行方法の詳細については、表 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 LPDS ビット (PWR 電源制御レジスタ (PWR_CR)) 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 15.3](#)、[セクション 15 : 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 [セクション 6.3.17 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は [セクション 6.3.18 : RCC クロック制御およびステータスレジスタ \(RCC_CSR\)](#) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は [セクション 6.3.17 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEON ビットで行います。

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

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

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

STOP モードの終了

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

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

割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 モードへの移行と終了

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

表 19. STOP モードへの移行と終了 (続き)

STOP モード	説明
モードの終了	<p>WFI または ISR 終了を使用して移行した場合</p> <p>割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 割込みベクタが NVIC で有効になっている必要があります)。割込みソースは、外部割込みまたはウェイクアップ機能を備えたペリフェラルになることがあります。表 38 : STM32F401xB/CSTM32F401xD/E のベクタテーブルを参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 0 の場合</p> <p>イベントモードに設定されている任意の EXTI ライン セクション 10.2.3 : 204 ページのウェイクアップイベント管理 を参照してください。</p> <p>WFE 命令を使用して移行し、SEVONPEND = 1 の場合 :</p> <ul style="list-style-type: none"> 割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 割込みベクタが NVIC で無効になっている場合も含む)。割込みソースは、外部割込みまたはウェイクアップ機能を備えたペリフェラルになることがあります。表 38 : STM32F401xB/CSTM32F401xD/E のベクタテーブルを参照してください。 ウェイクアップイベント : セクション 10.2.3 : 204 ページのウェイクアップイベント管理 を参照。
ウェイクアップ遅延時間	表 18 : STOP モードの動作モード

5.3.5 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 FPU 搭載 Cortex®-M4 のディープスリープ (deepsleep) モードに基づきます。結果として、1.2 V ドメインの電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。バックアップドメインのレジスタ (RTC レジスタ、RTC バックアップレジスタ) と STANDBY 回路を除いて、SRAM とレジスタの内容は失われます (図 7 を参照)。

STANDBY モードへの移行

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

STANDBY モードへの移行方法の詳細については、表 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 15.3](#)、[セクション 15 : 独立型ウォッチドッグ \(IWDG\)](#)
- リアルタイムクロック (RTC) : この設定は、バックアップドメイン制御レジスタ (RCC_BDCR) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI RC) : この設定は、制御/ステータスレジスタ (RCC_CSR) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE OSC) : この設定は、バックアップドメイン制御レジスタ (RCC_BDCR) の LSEON ビットで行います。

STANDBY モードの終了

STANDBY モードは、[セクション 5.4.2: PWR 電源制御/ステータスレジスタ \(PWR_PWR\)](#) を参照) は MCU が STANDBY モードにあったことを示します。PWR_CR を除くすべてのレジスタは、STANDBY モードからのウェイクアップ後にリセットされます。

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

表 20. STANDBY モードへの移行と終了

STANDBY モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットをセット – 電源制御レジスタ (PWR_CR) の PDDS ビットをセット – 電源制御レジスタ (PWR_CR) の CWUF ビットをクリア – 選択されたウェイクアップソース (RTC アラーム A、RTC アラーム B、RTC ウェイクアップ、タンパ、タイムスタンプフラグ) に対応する RTC フラグをクリア
	次の条件下での割り込みサービスルーチン (ISR) 終了時 – FPU 搭載 Cortex®-M4 システム制御レジスタの SLEEPDEEP ビットをセット – SLEEPONEXIT = 1 – 電源制御レジスタ (PWR_CR) の PDDS ビットをセット – 電源制御/ステータスレジスタ (PWR_SR) 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 (アラーム A またはアラーム B)、RTC ウェイクアップイベント、タンパイベント、タイムスタンプイベント、NRST ピンによる外部リセット、IWDG リセット
ウェイクアップ遅延時間	リセットフェーズ

STANDBY モードにおける I/O の状態

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

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

デバッグモード

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

ただし、DBGMCU_CR レジスタの設定ビットをセットすることによって、低電力モードを多用しているときでも、ソフトウェアのデバッグを行うことができます。詳細については、[セクション 23.16.1: 低電力モードのデバッグサポート](#) を参照してください。

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

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

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

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

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

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

この目的のため、[セクション 6.3.17 : RCC バックアップドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1 : 0] ビットをプログラムすることによって、次の 3 つの代替 RTC クロックソースのうちから 2 つを選択できます。

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

STOP モードからデバイスをウェイクアップさせるための RTC 代替機能

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

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

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

RTC 代替機能の安全なウェイクアップフラグクリアシーケンス

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

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

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

- RTC タイムスタンプを使用してデバイスを低電力モードからウェイクアップさせる場合は、次の設定が必要です。
 - a) RTC タイムスタンプ割込みを無効にします (RTC_CR レジスタの TSIE ビット)。
 - b) RTC タイムスタンプ (TSF) フラグをクリアします。
 - c) PWR ウェイクアップ (WUF) フラグをクリアします。
 - d) RTC タイムスタンプ割込みを有効にします。
 - e) 再び低電力モードに移行します。

5.4 電源制御レジスタ

5.4.1 PWR 電源制御レジスタ (PWR_CR)

アドレスオフセット : 0x00

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

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

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

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

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

00 : 予約済み (スケール 3 モードが選択)

01 : スケール 3 モード

10 : スケール 2 モード

11 : 予約済み (スケール 2 モードが選択)

ビット 13 **ADCD1** :

0 : 影響はありません。

1 : このビットの使い方の詳細は、AN4073 を参照してください。

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

ビット 12 予約済みであり、リセット値に保持する必要があります。

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

0 : デバイスが STOP モードのとき、メインレギュレータは電圧スケール 3 になります。

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

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

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

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

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

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

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

1 : デバイスが STOP モードのときフラッシュメモリはパワーダウン

ビット 8 DBP : バックアップドメイン書き込み保護の無効化

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

0 : RTC および RTC バックアップレジスタへのアクセス

1 : RTC および RTC バックアップレジスタへのアクセス

ビット 7:5 PLS[2:0] : PVD レベル選択

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

000 : 2.2 V

001 : 2.3 V

010 : 2.4 V

011 : 2.5 V

100 : 2.6 V

101 : 2.7 V

110 : 2.8 V

111 : 2.9 V

注 : 詳細については、データシートの電気的特性を参照してください。

ビット 4 PVDE : 電源電圧検出器有効化

このビットは、ソフトウェアによってセット/クリアされます。

0 : PVD は無効です。

1 : PVD は有効です。

ビット 3 CSBF : STANDBY フラグのクリア

このビットは常に 0 として読み出されます。

0 : 影響はありません。

1 : SBF STANDBY フラグのクリア (書き込み)

ビット 2 CWUF : ウェイクアップフラグのクリア

このビットは常に 0 として読み出されます。

0 : 影響はありません。

1 : WUF ウェイクアップフラグを 2 システムクロックサイクル後にクリアします。

ビット 1 PDDS : パワーダウンディープスリープ

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

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

1 : CPU がディープスリープに移行すると、STANDBY モードに移行します。

ビット 0 LPDS : 低電力ディープスリープ

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

0 : 電圧レギュレータは STOP モードの間もオン状態を保ちます。

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

5.4.2 PWR 電源制御／ステータスレジスタ (PWR_PWR)

アドレスオフセット : 0x04

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

このレジスタを読み出すには、通常の APB 読出しに対して、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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	VOS RDY	予約済み				BRE	EWUP	予約済み				BRR	PVDO	SBF	WUF
	r					rw	rw					r	r	r	r

- ビット 31:15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **VOSRDY** : レギュレータ電圧スケーリング出力選択レディビット
- 0 : レディでない状態
 - 1 : レディ状態
- ビット 13:10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **BRE** : バックアップレギュレータ有効化
- このビットがセットされると、バックアップレギュレータ (バックアップドメインの内容を保持するために使用) が有効になります。BRE がリセットされるとバックアップレギュレータはオフになります。いったんセットされると、アプリケーションは、バックアップレギュレータレディフラグ (BRR) が設定され、バックアップレジスタに書き込まれたデータが STANDBY モードおよび V_{BAT} モードにおいて保持されることが示されるのを待つ必要があります。
- 0 : バックアップレギュレータは無効
 - 1 : バックアップレギュレータは有効
- 注 :** システムリセットや電源リセットによって、デバイスが STANDBY モードからウェイクアップするとき、このビットはリセットされません。
- ビット 8 **EWUP** : WKUP ピンを有効にします。
- 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 : WKUP ピンは汎用 I/O として使用されます。この WKUP 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 : WKUP 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 (WKUP 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 **BRR** : バックアップレギュレータレディ
- ハードウェアによってセットされ、バックアップレギュレータが動作可能状態であることを示します。
- 0 : バックアップレギュレータは動作可能状態ではありません。
 - 1 : バックアップレギュレータは動作可能状態です。
- 注 :** システムリセットや電源リセットによって、デバイスが STANDBY モードからウェイクアップするとき、このビットはリセットされません。

- ビット 2 **PVDO** : PVD 出力
- このビットは、ハードウェアによってセット/クリアされます。PVDE ビットによって PVD が有効化されている場合のみ有効です。
- 0 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より高いです。
- 1 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より低いです。
- 注 : PVD は STANDBY モードによって停止します。したがって、このビットは STANDBY またはリセット後、PVDE ビットがセットされるまで 0 になります。
- ビット 1 **SBF** : STANDBY フラグ
- このビットはハードウェアによってセットされ、POR/PDR (パワーオンリセット/パワーダウンリセット)、または PWR_CR レジスタの CSBF ビットをセットすることによってのみクリアされます。
- 0 : デバイスは STANDBY モードではありません。
- 1 : デバイスは STANDBY モードです。
- ビット 0 **WUF** : ウェイクアップフラグ
- このビットはハードウェアによってセットされ、システムリセットまたは PWR_CR レジスタの CFUF ビットをセットするかのいずれかによってクリアされます。
- 0 : ウェイクアップイベントは発生していません。
- 1 : WKUP ピン、RTC アラーム (アラーム A またはアラーム B)、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップのいずれかよりウェイクアップイベントを受信しています。
- 注 : WKUP ピンのレベルがすでにハイであっても、(EWUP ビットをセットすることによって)WKUP ピンが有効化された場合、追加のウェイクアップイベントが検出されます。

5.5 PWR レジスタマップ

次の表に PWR レジスタの一覧を示します。

表 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	予約済み																VOS[1:0]		ADCDC1	予約済み	MRLVDS	LPLVDS	FPDS	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
	リセット値																	1	1	0													
0x004	PWR_CSR	予約済み																VOSRDY	予約済み						BRE	EWUP	予約済み			BRR	PVDO	SBF	WUF
	リセット値																																

レジスタ境界アドレスについては、38 ページの表 1 を参照してください。

6 リセットおよびクロック制御 (STM32F401xB/C および STM32F401xD/E 向け) (RCC)

6.1 リセット

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

6.1.1 システムリセット

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

システムリセットは、次のイベントのいずれかの発生時に生成されます。

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

ソフトウェアリセット

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

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

低電力管理リセット

低電力管理リセットを生成するには、2つの方法があります。

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

ユーザオプションバイトの詳細については、ST のウェブサイトから入手可能な STM32F401xB/C および STM32F401xD/E フラッシュプログラミングマニュアルを参照してください。

6.1.2 電源リセット

電源リセットは、次のいずれかのイベントが発生したときに生成されます。

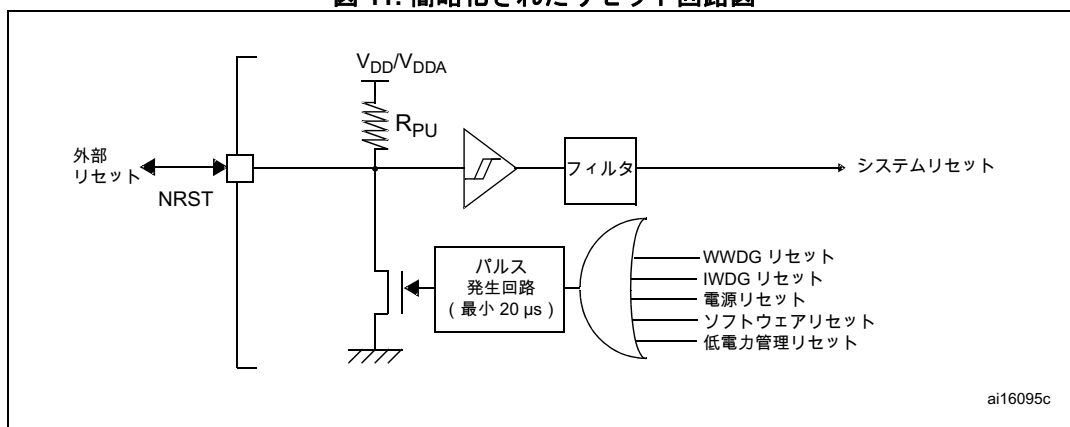
1. パワーオン/パワーダウンリセット (POR/PDR リセット) またはブラウンアウト (BOR) リセット
2. STANDBY モードを終了するとき

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

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

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

図 11. 簡略化されたリセット回路図



6.1.3 バックアップドメインリセット

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

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

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

6.2 クロック

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

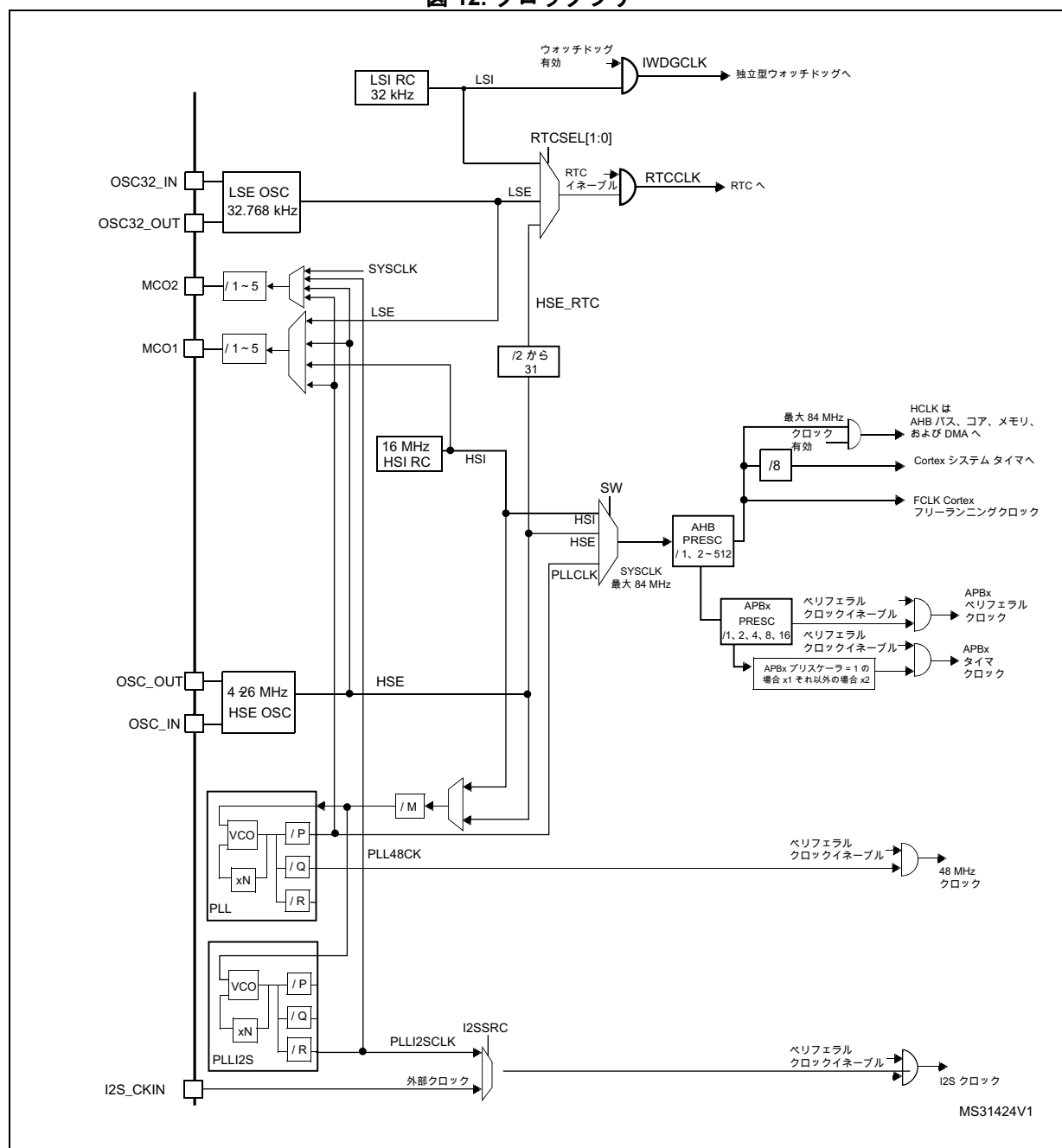
- HSI オシレータクロック
- HSE オシレータクロック
- メイン PLL (PLL) クロック

デバイスには、次の 2 つの 2 次クロックソースがあります。

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

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

図 12. クロックツリー



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

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

複数のプリスケアラを使用して AHB 周波数、ハイスピード APB (APB2) および低速 APB (APB1) ドメインの周波数を設定します。AHB ドメインの最大周波数は 84 MHz です。ハイスピード APB2 ドメインの最大許容周波数は 84 MHz です。ロースピード APB1 ドメインの最大許容周波数は 42 MHz です。

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

- USB OTG FS クロック (48 MHz)、SDIO クロック (≤ 48 MHz)。これらのクロックは、特定の PLL (PLL48CLK) の出力から供給されます。
- I²S クロック。

高品質のオーディオ性能を実現するため、I²S クロックを特定の PLL (PLL12S) または I2S_CKIN ピン上に配置された外部クロックから得ることができます。I²S クロック周波数と精度に関する詳細については、[セクション 20.4.4 : クロックジェネレータ](#)を参照してください。

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

STM32F401xB/C および STM32F401xD/E のタイマクロック周波数は、ハードウェアによって自動的に設定されます。これには、2 つのケースがあります。

1. APB プリスケアラが 1 の場合、タイマクロック周波数はタイマが接続されている APB ドメインの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、タイマが接続されている APB ドメインの周波数の 2 倍に設定されます。

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

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

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

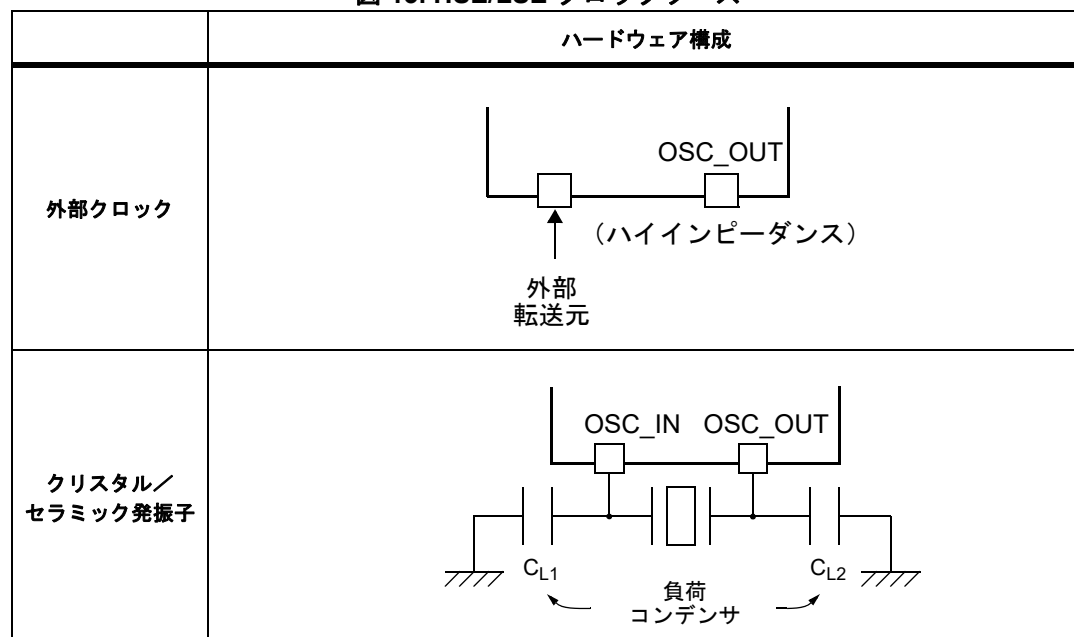
6.2.1 HSE クロック

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

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

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

図 13. HSE/LSE クロックソース



外部ソース (HSE バイパス)

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

外部クリスタル／セラミック発振子 (HSE クリスタル)

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

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

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

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

6.2.2 HSI クロック

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

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

較正

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

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

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

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

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

HSI 信号は、HSE クリスタルオシレータに障害がある場合のバックアップソース（補助クロック）としても使用できます。[セクション 6.2.7 : 99 ページのクロックセキュリティシステム \(CSS\)](#) を参照してください。

6.2.3 PLL の設定

STM32F401xB/C および STM32F401xD/E デバイスには、次の 2 つの PLL が搭載されています。

- HSE または HSI オシレータで駆動されるメイン PLL（PLL）で、2 種類の異なる出力クロックを備えています。
 - 最初の出力は、高速システムクロック（最大 84 MHz）の生成に使用します。
 - 2 つ目の出力は、USB OTG フルスピード（48 MHz）、アナログ乱数発生回路（ ≤ 48 MHz）、SDIO（ ≤ 48 MHz）のクロックの生成に使用します。
- I²S インタフェースで高品質オーディオ性能を実現するための正確なクロックの生成に使用する専用の PLL（PLLI2S）。

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

PLLI2S は、メイン PLL と同じ入力クロックを使用します（PLLM[5:0] と PLLSRC ビットはどちらの PLL にも共通）。しかしながら、PLLI2S には有効／無効および分周比を設定する専用のビットがあります。[セクション 6.3.1 : RCC クロック制御レジスタ \(RCC_CR\)](#)、[セクション 6.3.2 : RCC PLL 設定レジスタ \(RCC_PLLCFGR\)](#)、および [セクション 6.3.20: RCC PLLI2S 設定レジスタ \(RCC_PLLI2SCFGR\)](#) を参照してください。PLLI2S が有効になると、設定パラメータは変更できません。

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

スタ (RCC_CFGR) を使用して、PLL、PLLI2S をそれぞれ設定できます。

6.2.4 LSE クロック

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

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

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

外部ソース (LSE バイパス)

このモードでは、外部クロックソースが必要です。最大 1 MHz までの周波数を使用する必要があります。このモードを選択するには、RCC バックアップドメイン制御レジスタ (RCC_BDCR) の LSEBYP および LSEON ビットをセットします。約50%のデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC32_IN ピンを駆動する必要があり、その間、OSC32_OUT ピンはハイインピーダンスに保つ必要があります (図 13 を参照)。

6.2.5 LSI クロック

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

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

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

6.2.6 システムクロック (SYSCLK) の選択

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

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

6.2.7 クロックセキュリティシステム (CSS)

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

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

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

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

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

6.2.8 RTC/AWU クロック

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

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

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

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

- LSE が RTC クロックとして使用されている場合：
 - V_{DD} 供給がオフになった場合でも、 V_{BAT} の供給が保たれている限り、RTC は動作を続けます。
- LSI が自動ウェイクアップユニット (AWU) クロックとして選択されている場合：
 - V_{DD} 供給がオフになった場合、AWU の状態は保証されません。LSI 較正の詳細については、[セクション 6.2.5 : 98 ページの LSI クロック](#)を参照してください。
- HSE クロックが RTC クロックとして使用されている場合：
 - V_{DD} 供給がオフになった場合や、内蔵されている電圧レギュレータがオフになった場合 (1.2 V 電源ドメインへの電力供給が停止にされた場合)、RTC の状態は保証されません。

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

6.2.9 ウォッチドッグクロック

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

6.2.10 クロック信号出力

2 つのマイクロコントローラクロック出力 (MCO) ピンが利用できます。

- MCO1

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

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

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

- MCO2

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

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

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

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

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

6.2.11 TIM5/TIM11 を使用した内部／外部クロックの測定

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

TIM5 チャンネル 4 を使用した内部／外部クロックの測定

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

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

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

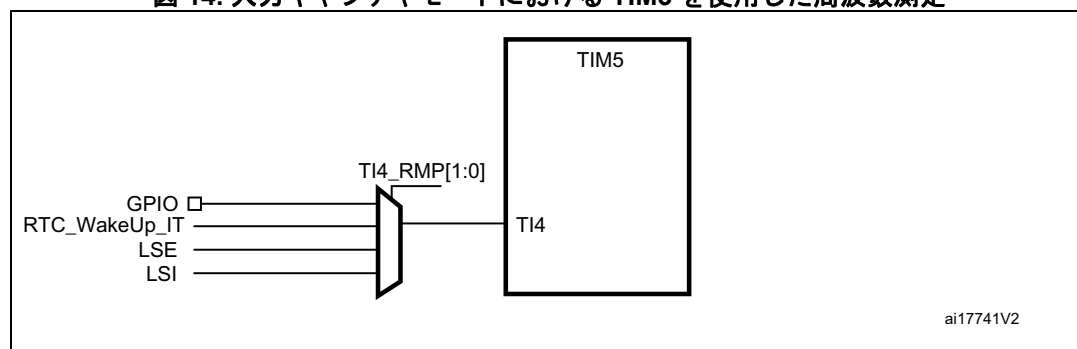
この基本概念は、相対測定 (HSI/LSE 比など) ができることにあり、そのため、精度は 2 つのクロックソース間の比に密接につながっています。この比が大きければ大きいほど、測定精度は向上します。

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

LSI 周波数を測定するには、次の手順に従ってください。

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

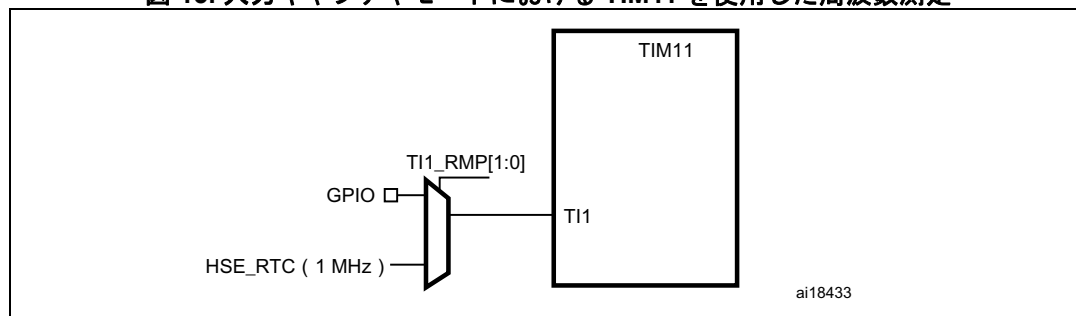
図 14. 入力キャプチャモードにおける TIM5 を使用した周波数測定



TIM11 チャンネル 1 を使用した内部／外部クロックの測定

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

図 15. 入力キャプチャモードにおける TIM11 を使用した周波数測定



6.3 RCC レジスタ

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

6.3.1 RCC クロック制御レジスタ (RCC_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 XX81 (X は未定義)。

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

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

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

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

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

0 : PLLI2S アンロック

1 : PLLI2S ロック

ビット 26 **PLLI2SON** : PLLI2S イネーブル

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

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

0 : PLLI2S オフ

1 : PLLI2S オン

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

ハードウェアによってセットされ、PLL がロック状態であることを示します。

0 : PLL アンロック

1 : PLL ロック

ビット 24 **PLLON** : メイン PLL (PLL) 有効化

PLL を有効にするために、ソフトウェアによってセット/クリアされます。

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

0 : PLL オフ

1 : PLL オン

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CSSON** : クロックセキュリティシステム有効化

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

0 : クロックセキュリティシステムオフ (クロック検出回路オフ)

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

ビット 18 HSEBYP : HSE クロックバイパス

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

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

0 : HSE オシレータはバイパスされません。

1 : HSE オシレータはバイパスされ、外部クロックが使用されます。

ビット 17 HSERDY : HSE クロックレディフラグ

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

0 : HSE オシレータはレディ状態ではありません。

1 : HSE オシレータはレディ状態です。

ビット 16 HSEON : HSE クロック有効化

ソフトウェアでセット／クリアされます。

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

0 : HSE オシレータオフ

1 : HSE オシレータオン

ビット 15:8 HSICAL[7:0] : 内部ハイスピードクロック較正

これらのビットは、スタートアップ時に自動的に初期化されます。

ビット 7:3 HSITRIM[4:0] : 内部ハイスピードクロックトリミング

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

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 HSIRDY : 内部ハイスピードクロックレディフラグ

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

0 : HSI オシレータはレディ状態ではありません。

1 : HSI オシレータはレディ状態です。

ビット 0 HSION : 内部ハイスピードクロック有効化

ソフトウェアでセット／クリアされます。

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

0 : HSI オシレータオフ

1 : HSI オシレータオン

6.3.2 RCC PLL 設定レジスタ (RCC_PLLCFGR)

アドレスオフセット : 0x04

リセット値 : 0x2400 3010

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

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み				PLLQ3	PLLQ2	PLLQ1	PLLQ0	予約済み	PLLSRC	予約済み				PLLP1	PLLP0
				rw	rw	rw	rw		rw					rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	PLLN									PLLM5	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **PLLQ** : USB OTG フルスピード、SDIO、および乱数発生回路クロックのためのメイン PLL (PLL) 分周比
USB OTG フルスピードクロック、乱数発生回路クロック、SDIO クロックの周波数を制御するためにソフトウェアによってセット／クリアされます。これらのビットに書き込めるのは、PLL が無効なときだけです。

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

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

0000 : PLLQ = 0、誤った設定

0001 : PLLQ = 1、誤った設定

0010 : PLLQ = 2

0011 : PLLQ = 3

0100 : PLLQ = 4

...

1111 : PLLQ = 15

ビット 23 予約済みであり、リセット値に保持する必要があります。

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

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

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

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

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

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

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

注意： ソフトウェアは、このドメインで 84MHz を超えないように、これらのビットを正しくセットする必要があります。

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

00 : PLL_P = 2

01 : PLL_P = 4

10 : PLL_P = 6

11 : PLL_P = 8

ビット 14:6 **PLL_N** : メイン PLL (PLL) の VCO に対する通倍数

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

注意： これらのビットは、VCO 出力周波数が 192~432 MHz の間になるように、ソフトウェアで正しく設定する必要があります。(セクション 6.3.20 : [RCC_PLLI2S 設定レジスタ \(RCC_PLLI2SCFGR\)](#) も参照)

VCO 出力周波数 = VCO 入力周波数 × PLL_N、ここで $192 \leq PLL_N \leq 432$

000000000 : PLL_N = 0、誤った設定

000000001 : PLL_N = 1、誤った設定

...

...

110110000 : PLL_N = 432

110110001 : PLL_N = 433、誤った設定

...

111111111 : PLL_N = 511、誤った設定

ビット 5:0 **PLL_M** : メイン PLL (PLL) およびオーディオ PLL (PLLI2S) 入力クロックのための分周比

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

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

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

000000 : PLL_M = 0、誤った設定

000001 : PLL_M = 1、誤った設定

000010 : PLL_M = 2

000011 : PLL_M = 3

000100 : PLL_M = 4

...

111110 : PLL_M = 62

111111 : PLL_M = 63

6.3.3 RCC クロック設定レジスタ (RCC_CFGR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : 0 ≤ 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 個のウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2		MCO2 PRE[2:0]			MCO1 PRE[2:0]			I2SSCR	MCO1		RTCPRE[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
PPRE2[2:0]			PPRE1[2:0]			予約済み		HPRE[3:0]				SWS1	SWS0	SW1	SW0
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	r	r	rw	rw

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

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

00 : システムクロック (SYSCLK) の選択

01 : PLLI2S クロックの選択

10 : HSE オシレータクロックの選択

11 : PLL クロックの選択

ビット 29:27 **MCO2PRE** : MCO2 プリスケーラ

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

0xx : 分周なし

100 : 分周比 2

101 : 分周比 3

110 : 分周比 4

111 : 分周比 5

ビット 26:24 **MCO1PRE** : MCO1 プリスケーラ

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

0xx : 分周なし

100 : 分周比 2

101 : 分周比 3

110 : 分周比 4

111 : 分周比 5

ビット 23 **I2SSRC** : I²S クロック選択

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

0 : I²S クロックソースとして PLLI2S クロックが使用されます。

1 : I²S クロックソースとして I2S_CKIN ピンに割り当てられている外部クロックが使用されます。

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

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

00 : HSI クロックの選択

01 : LSE オシレータの選択

10 : HSE オシレータクロックの選択

11 : PLL クロックの選択

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

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

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

00000 : クロックなし

00001 : クロックなし

00010 : HSE/2

00011 : HSE/3

00100 : HSE/4

...

11110 : HSE/30

11111 : HSE/31

ビット 15:13 **PPRE2** : APB ハイスピードブリスケーラ（APB2）

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

注意： ソフトウェアは、このドメインで 84 MHz を超えないように、これらのビットを正しくセットする必要があります。クロックは、PPRE2 に書き込んでから 1~16 AHB の間の新しい分周比で分周されます。

0xx : AHB クロックは分周されません。

100 : AHB クロックは 2 分周されます。

101 : AHB クロックは 4 分周されます。

110 : AHB クロックは 8 分周されます。

111 : AHB クロックは 16 分周されます。

ビット 12:10 **PPRE1** : APB 低速ブリスケーラ（APB1）

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

注意： ソフトウェアは、このドメインで 42 MHz を超えないように、これらのビットを正しくセットする必要があります。クロックは、PPRE1 に書き込んでから 1~16 AHB の間の新しい分周比で分周されます。

0xx : AHB クロックは分周されません。

100 : AHB クロックは 2 分周されます。

101 : AHB クロックは 4 分周されます。

110 : AHB クロックは 8 分周されます。

111 : AHB クロックは 16 分周されます。

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **HPRE** : AHB プリスケアラ

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

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

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

0xxx : システムクロックは分周されません。

1000 : システムクロックは 2 分周されます。

1001 : システムクロックは 4 分周されます。

1010 : システムクロックは 8 分周されます。

1011 : システムクロックは 16 分周されます。

1100 : システムクロックは 64 分周されます。

1101 : システムクロックは 128 分周されます。

1110 : システムクロックは 256 分周されます。

1111 : システムクロックは 512 分周されます。

ビット 3:2 **SWS** : 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

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

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

00 : HSI オシレータがシステムクロックとして選択されます。

01 : HSE オシレータがシステムクロックとして選択されます。

10 : PLL がシステムクロックとして選択されます。

11 : 設定禁止

6.3.4 RCC クロック割込みレジスタ（RCC_CIR）

アドレスオフセット：0x0C

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

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

0 : 影響なし。

1 : CSSF フラグをクリアします。

ビット 22 予約済みであり、リセット値に保持する必要があります。

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

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

0 : 影響なし。

1 : PLLI2SRDYF がクリアされます。

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

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

0 : 影響なし。

1 : PLLRDYF がクリアされます。

ビット 19 **HSERDYC** : HSE レディ割込みクリア

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

0 : 影響なし。

1 : HSERDYF がクリアされます。

ビット 18 **HSIRDYC** : HSI レディ割込みクリア

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

0 : 影響なし。

1 : HSIRDYF がクリアされます。

ビット 17 **LSERDYC** : LSE レディ割込みクリア

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

0 : 影響なし。

1 : LSERDYF がクリアされます。

ビット 16 **LSIRDYC** : LSI レディ割込みクリア

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

0 : 影響なし。

1 : LSIRDYF がクリアされます。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

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

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

0 : PLLI2S ロック割込み無効

1 : PLLI2S ロック割込み有効

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

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

0 : PLL ロック割込みは無効です。

1 : PLL ロック割込みは有効です。

ビット 11 **HSERDYIE** : HSE レディ割込み有効化

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

0 : HSE レディ割込みは無効です。

1 : HSE レディ割込みは有効です。

ビット 10 **HSIRDYIE** : HSI レディ割込み有効化

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

0 : HSI レディ割込み無効

1 : HSI レディ割込み有効

ビット 9 **LSERDYIE** : LSE レディ割込み有効化

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

0 : LSE レディ割込みは無効です。

1 : LSE レディ割込みは有効です。

ビット 8 **LSIRDYIE** : LSI レディ割込み有効化

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

0 : LSI レディ割込みは無効です。

1 : LSI レディ割込みは有効です。

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。

CSSF 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込みが発生しました。

ビット 6 予約済みであり、リセット値に保持する必要があります。

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

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

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

0 : PLLI2S ロックによるクロックレディ割込みは発生していません。

1 : PLLI2S ロックによるクロックレディ割込みが発生しました。

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

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。

PLLRDYF 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込みが発生しました。

ビット 3 HSERDYF : HSE レディ割込みフラグ

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

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

0 : HSE オシレータによるクロックレディ割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込みが発生しました。

ビット 2 HSIRDYF : HSI レディ割込みフラグ

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

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

0 : HSI オシレータによるクロックレディ割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込みが発生しました。

ビット 1 LSERDYF : LSE レディ割込みフラグ

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

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

0 : LSE オシレータによるクロックレディ割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込みが発生しました。

ビット 0 LSIRDYF : LSI レディ割込みフラグ

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

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

0 : LSI オシレータによるクロックレディ割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込みが発生しました。

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

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

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

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

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 DMA2RST : DMA2 リセット

ソフトウェアでセット／クリアされます。

0 : DMA2 をリセットしません。

1 : DMA2 をリセットします。

ビット 21 DMA1RST : DMA1 リセット

ソフトウェアでセット／クリアされます。

0 : DMA1 をリセットしません。

1 : DMA1 をリセットします。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCRST** : CRC リセット

ソフトウェアでセット／クリアされます。

0 : CRC をリセットしません。

1 : CRC をリセットします。

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GPIOHRST** : IO ポート H リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート H をリセットしません。

1 : IO ポート H をリセットします。

ビット 6:5 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 **GPIOERST** : IO ポート E リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート E をリセットしません。

1 : IO ポート E をリセットします。

ビット 3 **GPIODRST** : IO ポート D リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート D をリセットしません。

1 : IO ポート D をリセットします。

ビット 2 **GPIOCRST** : IO ポート C リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート C をリセットしません。

1 : IO ポート C をリセットします。

ビット 1 **GPIOBRST** : IO ポート B リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート B をリセットしません。

1 : IO ポート B をリセットします。

ビット 0 **GPIOARST** : IO ポート A リセット

ソフトウェアでセット／クリアされます。

0 : IO ポート A をリセットしません。

1 : IO ポート A をリセットします。

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

アドレスオフセット：0x14

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								OTGFS RST	予約済み						
								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 6:0 予約済みであり、リセット値に保持する必要があります。

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

アドレスオフセット：0x20

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み			PWR RST	予約済み				I2C3 RST	I2C2 RST	I2C1 RST	予約済み			USART2 RST	予約 済み
			rw					rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	予約済み		WWDG RST	予約済み							TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw			rw								rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **PWRRST**：電源インタフェースリセット

ソフトウェアでセット／クリアされます。

0：電源インタフェースをリセットしません。

1：電源インタフェースをリセットします。

ビット 27:24 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 **I2C3RST**：I2C3 リセット

ソフトウェアでセット／クリアされます。

0：I2C3 をリセットしません。

1：I2C3 をリセットします。

ビット 22 **I2C2RST**：I2C2 リセット

ソフトウェアでセット／クリアされます。

0：I2C2 をリセットしません。

1：I2C2 をリセットします。

ビット 21 **I2C1RST** : I2C1 リセット

ソフトウェアでセット／クリアされます。

0 : I2C1 をリセットしません。

1 : I2C1 をリセットします。

ビット 20:18 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 **USART2RST** : USART2 リセット

ソフトウェアでセット／クリアされます。

0 : USART2 をリセットしません。

1 : USART2 をリセットします。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **SPI3RST** : SPI3 リセット

ソフトウェアでセット／クリアされます。

0 : SPI3 をリセットしません。

1 : SPI3 をリセットします。

ビット 14 **SPI2RST** : SPI2 リセット

ソフトウェアでセット／クリアされます。

0 : SPI2 をリセットしません。

1 : SPI2 をリセットします。

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **WWDGRST** : ウィンドウ型ウォッチドッグリセット

ソフトウェアでセット／クリアされます。

0 : ウィンドウ型ウォッチドッグをリセットしません。

1 : ウィンドウ型ウォッチドッグをリセットします。

ビット 10:4 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 **TIM5RST** : TIM5 リセット

ソフトウェアでセット／クリアされます。

0 : TIM5 をリセットしません。

1 : TIM5 をリセットします。

ビット 2 **TIM4RST** : TIM4 リセット

ソフトウェアでセット／クリアされます。

0 : TIM4 をリセットしません。

1 : TIM4 をリセットします。

ビット 1 **TIM3RST** : TIM3 リセット

ソフトウェアでセット／クリアされます。

0 : TIM3 をリセットしません。

1 : TIM3 をリセットします。

ビット 0 **TIM2RST** : TIM2 リセット

ソフトウェアでセット／クリアされます。

0 : TIM2 をリセットしません。

1 : TIM2 をリセットします。

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

アドレスオフセット：0x24

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み													TIM11 RST	TIM10 RST	TIM9 RST
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	SYSCFG RST	SPI4 RST	SPI1 RST	SDIO RST	予約済み		ADC1 RST	予約済み		USART6 RST	USART1 RST	予約済み		TIM1 RST	
	rw	rw	rw	rw			rw			rw					

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11RST** : TIM11 リセット

ソフトウェアでセット／クリアされます。

0 : TIM11 をリセットしません。

1 : TIM11 をリセットします。

ビット 17 **TIM10RST** : TIM10 リセット

ソフトウェアでセット／クリアされます。

0 : TIM10 をリセットしません。

1 : TIM10 をリセットします。

ビット 16 **TIM9RST** : TIM9 リセット

ソフトウェアでセット／クリアされます。

0 : TIM9 をリセットしません。

1 : TIM9 をリセットします。

ビット 15 予約済みであり、リセット値に保持する必要があります。

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

ソフトウェアでセット／クリアされます。

0 : システム設定コントローラをリセットしません。

1 : システム設定コントローラをリセットします。

ビット 13 **SPI4RST** : SPI4 リセット

ソフトウェアでセット／クリアされます。

0 : SPI4 をリセットしません。

1 : SPI4 をリセットします。

ビット 12 **SPI1RST** : SPI1 リセット

ソフトウェアでセット／クリアされます。

0 : SPI1 をリセットしません。

1 : SPI1 をリセットします。

ビット 11 **SDIORST** : SDIO リセット

ソフトウェアでセット／クリアされます。

0 : SDIO モジュールをリセットしません。

1 : SDIO モジュールをリセットします。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ADC1RST** : ADC インタフェースリセット

ソフトウェアでセット／クリアされます。

0 : ADCインタフェースをリセットしません。

1 : ADCインタフェースをリセットします。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **USART6RST** : USART6 リセット

ソフトウェアでセット／クリアされます。

0 : USART6 をリセットしません。

1 : USART6 をリセットします。

ビット 4 **USART1RST** : USART1 リセット

ソフトウェアでセット／クリアされます。

0 : USART1 をリセットしません。

1 : USART1 をリセットします。

ビット 3:1 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 **TIM1RST** : TIM1 リセット

ソフトウェアでセット／クリアされます。

0 : TIM1 をリセットしません。

1 : TIM1 をリセットします。

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

アドレスオフセット：0x30

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									DMA2EN	DMA1EN	予約済み				
									rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み			CRCEN	予約済み				GPIOH EN	予約済み		GPIOEEN	GPIOD EN	GPIOC EN	GPIO BEN	GPIOA EN
			rw					rw			rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DMA2EN** : DMA2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : DMA2 クロック無効

1 : DMA2 クロック有効

ビット 21 **DMA1EN** : DMA1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : DMA1 クロック無効

1 : DMA1 クロック有効

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCEN** : CRC クロック有効化

ソフトウェアでセット／クリアされます。

0 : CRC クロック無効

1 : CRC クロック有効

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GPIOHEN** : IO ポート H クロック有効化

ソフトウェアでセット／クリアされます。

0 : IO ポート H クロック無効

1 : IO ポート H クロック有効

ビット 6:5 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 **GPIOEEN** : IO ポート E クロック有効化

ソフトウェアでセット／クリアされます。

0 : IO ポート E クロック無効

1 : IO ポート E クロック有効

ビット 3 **GPIODEN** : IO ポート D クロック有効化

ソフトウェアでセット／クリアされます。

0 : IO ポート D クロック無効

1 : IO ポート D クロック有効

ビット 2 **GPIOCEN** : IO ポート C クロック有効化

ソフトウェアでセット／クリアされます。

0 : IO ポート C クロック無効

1 : IO ポート C クロック有効

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

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

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

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

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

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTGFSEN** : USB OTG FS クロック有効化
ソフトウェアでセット/クリアされます。
0 : USB OTG フルスピード クロック無効
1 : USB OTG フルスピード クロック有効

ビット 6:0 予約済みであり、リセット値に保持する必要があります。

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

アドレスオフセット : 0x40

リセット値 : 0x0000 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
予約済み			PWR EN	予約済み				I2C3 EN	I2C2 EN	I2C1 EN	予約済み			USART2 EN	予約済み
			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	予約済み		WWDG EN	予約済み						TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN	
rw	rw			rw							rw	rw	rw	rw	

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

ビット 28 **PWREN** : 電源インタフェースクロック有効
ソフトウェアでセット／クリアされます。
0 : 電源インタフェースクロック無効
1 : 電源インタフェースクロック有効

ビット 27:24 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 **I2C3EN** : I2C3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C3 クロック無効
1 : I2C3 クロック有効

ビット 22 **I2C2EN** : I2C2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C2 クロック無効
1 : I2C2 クロック有効

ビット 21 **I2C1EN** : I2C1 クロック有効化
ソフトウェアでセット／クリアされます。
0 : I2C1 クロック無効
1 : I2C1 クロック有効

ビット 20:18 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 **USART2EN** : USART2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : USART2 クロック無効
1 : USART2 クロック有効

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **SPI3EN** : SPI3 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI3 クロック無効
1 : SPI3 クロック有効

ビット 14 **SPI2EN** : SPI2 クロック有効化
ソフトウェアでセット／クリアされます。
0 : SPI2 クロック無効
1 : SPI2 クロック有効

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **WWDGEN** : ウィンドウ型ウォッチドッグクロック有効化
ソフトウェアでセット／クリアされます。
0 : ウィンドウ型ウォッチドッグクロック無効
1 : ウィンドウ型ウォッチドッグクロック有効

ビット 10:4 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 **TIM5EN** : TIM5 クロック有効化
ソフトウェアでセット／クリアされます。
0 : TIM5 クロック無効
1 : TIM5 クロック有効

ビット 2 **TIM4EN** : TIM4 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM4 クロック無効

1 : TIM4 クロック有効

ビット 1 **TIM3EN** : TIM3 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM3 クロック無効

1 : TIM3 クロック有効

ビット 0 **TIM2EN** : TIM2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM2 クロック無効

1 : TIM2 クロック有効

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

アドレスオフセット：0x44

リセット値：0x0000 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
予約済み													TIM11 EN	TIM10 EN	TIM9 EN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	SYSCFG EN	SPI4EN	SPI1 EN	SDIO EN	予約済み			予約済み			USART6 EN	USART1 EN	予約済み		TIM1 EN
	rw	rw	rw	rw							rw	rw			rw

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11EN** : TIM11 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM11 クロック無効

1 : TIM11 クロック有効

ビット 17 **TIM10EN** : TIM10 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM10 クロック無効

1 : TIM10 クロック有効

ビット 16 **TIM9EN** : TIM9 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM9 クロック無効

1 : TIM9 クロック有効

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **SYSCFGEN** : システム設定コントローラクロック有効化

ソフトウェアでセット／クリアされます。

0 : システム設定コントローラクロック無効

1 : システム設定コントローラクロック有効

ビット 13 **SPI4EN** : SPI4 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SPI4 クロック無効

1 : SPI4 クロック有効化

ビット 12 **SPI1EN** : SPI1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SPI1 クロック無効

1 : SPI1 クロック有効

ビット 11 **SDIOEN** : SDIO クロックイネーブル

ソフトウェアでセット／クリアされます。

0 : SDIO モジュールクロック無効

1 : SDIO モジュールクロック有効

ビット 8 **ADC1EN** : ADC1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : ADC1 クロック無効

1 : ADC1 クロック無効

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **USART6EN** : USART6 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART6 クロック無効

1 : USART6 クロック有効

ビット 4 **USART1EN** : USART1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART1 クロック無効

1 : USART1 クロック有効

ビット 3:1 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 **TIM1EN** : TIM1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM1 クロック無効

1 : TIM1 クロック有効

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

アドレスオフセット：0x50

リセット値：0x0061 900F

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
予約済み									DMA2 LPEN	DMA1 LPEN	予約済み				SRAM1 LPEN
									rw	rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLITF LPEN	予約済み		CRC LPEN	予約済み				GPIOH LPEN	予約済み		GPIOE LPEN	GPIOD LPEN	GPIOC LPEN	GPIOB LPEN	GPIOA LPEN
rw			rw					rw			rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DMA2LPEN** : SLEEP モード時 DMA2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 DMA2 クロック無効

1 : SLEEP モード時 DMA2 クロック有効

ビット 21 **DMA1LPEN** : SLEEP モード時 DMA1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 DMA1 クロック無効

1 : SLEEP モード時 DMA1 クロック有効

ビット 20:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **SRAM1LPEN** : SLEEP モード時 SRAM1 インタフェースクロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 SRAM1 インタフェースクロック無効

1 : SLEEP モード時 SRAM1 インタフェースクロック有効

ビット 15 **FLITFLPEN** : SLEEP モード時フラッシュインタフェースクロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時フラッシュインタフェースクロック無効

1 : SLEEP モード時フラッシュインタフェースクロック有効

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CRCLPEN** : SLEEP モード時 CRC クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 CRC クロック無効

1 : SLEEP モード時 CRC クロック有効

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **GPIOHLPEN** : SLEEP モード時 IO ポート H クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 IO ポート H クロック無効

1 : SLEEP モード時 IO ポート H クロック有効

ビット 6:5 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 **GPIOLPEN** : SLEEP モード時 IO ポート E クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 IO ポート E クロック無効

1 : SLEEP モード時 IO ポート E クロック有効

ビット 3 **GPIODLPEN** : SLEEP モード時 IO ポート D クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 IO ポート D クロック無効

1 : SLEEP モード時 IO ポート D クロック有効

ビット 2 **GPIOCLPEN** : SLEEP モード時 IO ポート C クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 IO ポート C クロック無効

1 : SLEEP モード時 IO ポート C クロック有効

ビット 1 **GPIOBLPEN** : SLEEP モード時 IO ポート B クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 IO ポート B クロック無効

1 : SLEEP モード時 IO ポート B クロック有効

ビット 0 **GPIOALPEN** : SLEEP モード時 IO ポート A クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 IO ポート A クロック無効

1 : SLEEP モード時 IO ポート A クロック有効

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

アドレスオフセット : 0x54

リセット値 : 0x0000 0080

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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								OTGFS LPEN	予約済み						
								rw							

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **OTGFSLPEN** : SLEEP モード時 USB OTG FS クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 USB OTG フルスピード クロック無効

1 : SLEEP モード時 USB OTG フルスピード クロック有効

ビット 6:0 予約済みであり、リセット値に保持する必要があります。

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

アドレスオフセット：0x60

リセット値：0x10E2 C80F

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
予約済み			PWR LPEN	予約済み				I2C3 LPEN	I2C2 LPEN	I2C1 LPEN	予約済み			USART2 LPEN	予約済み
			rw					rw	rw	rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 LPEN	SPI2 LPEN	予約済み		WWDG LPEN	予約済み							TIM5 LPEN	TIM4 LPEN	TIM3 LPEN	TIM2 LPEN
rw	rw			rw								rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値に保持する必要があります。

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時電源インタフェースクロック無効

1 : SLEEP モード時電源インタフェースクロック有効

ビット 27:24 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 **I2C3LPEN** : SLEEP モード時 I2C3 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 I2C3 クロック無効

1 : SLEEP モード時 I2C3 クロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 I2C2 クロック無効

1 : SLEEP モード時 I2C2 クロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 I2C1 クロック無効

1 : SLEEP モード時 I2C1 クロック有効

ビット 20:18 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 **USART2LPEN** : SLEEP モード時 USART2 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 USART2 クロック無効

1 : SLEEP モード時 USART2 クロック有効

ビット 16 予約済みであり、リセット値に保持する必要があります。

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 SPI3 クロック無効

1 : SLEEP モード時 SPI3 クロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 SPI2 クロック無効

1 : SLEEP モード時 SPI2 クロック有効

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

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

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時ウィンドウ型ウォッチドッグクロック無効

1 : SLEEP モード時ウィンドウ型ウォッチドッグクロック有効

ビット 10:4 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 **TIM5LPEN** : SLEEP モード時 TIM5 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM5 クロック無効

1 : SLEEP モード時 TIM5 クロック有効

ビット 2 **TIM4LPEN** : SLEEP モード時 TIM4 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM4 クロック無効

1 : SLEEP モード時 TIM4 クロック有効

ビット 1 **TIM3LPEN** : SLEEP モード時 TIM3 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM3 クロック無効

1 : SLEEP モード時 TIM3 クロック有効

ビット 0 **TIM2LPEN** : SLEEP モード時 TIM2 クロック有効化

ソフトウェアでセット/クリアされます。

0 : SLEEP モード時 TIM2 クロック無効

1 : SLEEP モード時 TIM2 クロック有効

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

アドレスオフセット：0x64

リセット値：0x0007 7930

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	
予約済み													TIM11 LPEN	TIM10 LPEN	TIM9 LPEN	
													rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
予約済み	SYSC FG LPEN	SPI4LP EN	SPI1 LPEN	SDIO LPEN	予約済み			ADC1 LPEN	予約済み			USART6 LPEN	USART1 LPEN	予約済み		TIM1 LPEN
	rw	rw	rw	rw				rw				rw	rw			

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM11LPEN** : SLEEP モード時 TIM11 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM11 クロック無効

1 : SLEEP モード時 TIM11 クロック有効

ビット 17 **TIM10LPEN** : SLEEP モード時 TIM10 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM10 クロック無効

1 : SLEEP モード時 TIM10 クロック有効

ビット 16 **TIM9LPEN** : SLEEP モード時 TIM9 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM9 クロック無効

1 : SLEEP モード時 TIM9 クロック有効

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **SYSCFGLPEN** : SLEEP モード時システム設定コントローラクロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時システム設定コントローラクロック無効

1 : SLEEP モード時システム設定コントローラクロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 SPI4 クロック無効

1 : SLEEP モード時 SPI4 クロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 SPI1 クロック無効

1 : SLEEP モード時 SPI1 クロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 SDIO モジュールクロック無効

1 : SLEEP モード時 SDIO モジュールクロック有効

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 ADC1 クロック無効

1 : SLEEP モード時 ADC1 クロック有効

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 USART6 クロック無効

1 : SLEEP モード時 USART6 クロック有効

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

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 USART1 クロック無効

1 : SLEEP モード時 USART1 クロック有効

ビット 3:1 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 **TIM1LPEN** : SLEEP モード時 TIM1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SLEEP モード時 TIM1 クロック無効

1 : SLEEP モード時 TIM1 クロック有効

6.3.17 RCC バックアップドメイン制御レジスタ（RCC_BDCR）

アドレスオフセット：0x70

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

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

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

RCC バックアップドメイン制御レジスタ (RCC_BDCR) の LSEON、LSEBYP、RTCSEL、および RTCEN ビットは、バックアップドメインにあります。このため、リセット後、これらのビットは書き込み保護されるので、これらを変更するには、**PWR 電源制御レジスタ (PWR_CR)** の DBP ビットをセットする必要があります。詳細については、[72 ページのセクション 5.1.2](#) を参照してください。これらのビットは、バックアップドメインリセット後にのみリセットされます（[セクション 6.1.3：バックアップドメインリセット](#)を参照）。内部または外部リセットは、これらのビットに影響しません。

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

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **BDRST**：バックアップドメインソフトウェアリセット

ソフトウェアでセット／クリアされます。

0：リセットは有効ではありません。

1：バックアップドメイン全体をリセットします。

ビット 15 **RTCEN**：RTC クロック有効化

ソフトウェアでセット／クリアされます。

0：RTC クロック無効

1：RTC クロック有効

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **RTCSEL[1:0]**：RTC クロックソース選択

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

00：クロックなし。

01：LSE オシレータクロックが RTC クロックとして使用されます。

10：LSI オシレータクロックが RTC クロックとして使用されます。

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

ビット 7:3 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 **LSEBYP**：外部低速オシレータバイパス

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

0：LSE オシレータはバイパスされません。

1：LSE オシレータはバイパスされます。

ビット 1 **LSERDY** : 外部低速オシレータレディ

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

0 : LSE クロックはレディ状態ではありません。

1 : LSE クロックはレディ状態です。

ビット 0 **LSEON** : 外部低速オシレータ有効化

ソフトウェアでセット/クリアされます。

0 : LSE クロックオフ

1 : LSE クロックオン

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

アドレスオフセット : 0x74

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

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

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

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

ビット 31 **LPWRRSTF** : 低電力リセットフラグ

低電力管理リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : 低電力管理リセットは発生していません。

1 : 低電力管理リセットが発生しました。

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

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

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

RMVF ビットに書き込むことによってクリアされます。

0 : ウィンドウ型ウォッチドッグリセットは発生していません。

1 : ウィンドウ型ウォッチドッグリセットが発生しました。

ビット 29 **IWDGRSTF** : 独立型ウォッチドッグリセットフラグ

V_{DD} ドメインから独立型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : ウォッチドッグリセットは発生していません。

1 : ウォッチドッグリセットが発生しました。

ビット 28 **SFTRSTF** : ソフトウェアリセットフラグ

ソフトウェアリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : ソフトウェアリセットは発生していません。

1 : ソフトウェアリセットが発生しました。

ビット 27 PORRSTF : POR/PDR リセットフラグ

POR/PDR リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : POR/PDR リセットは発生していません。

1 : POR/PDR リセットが発生しました。

ビット 26 PINRSTF : PIN リセットフラグ

NRST ピンに外部リセット信号が入力され、リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : NRST ピンからのリセットは発生していません。

1 : NRST ピンからリセットが発生しました。

ビット 25 BORRSTF : BOR リセットフラグ

RMVF ビットに書き込むことによりソフトウェアによってクリアされます。

POR/PDR リセットまたは BOR リセットが発生したときに、ハードウェアによってセットされます。

0 : POR/PDR リセットまたは BOR リセットは発生していません。

1 : POR/PDR リセットまたは BOR リセットが発生しました。

ビット 24 RMVF : リセットフラグ解除

リセットフラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : リセットフラグをクリアします。

ビット 23:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 LSIRDY : 内部低速オシレータレディ

内部 RC 40 kHz オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSION ビットがクリアされた後、LSIRDY は、LSI オシレータクロックの 3 サイクル後にローになります。

0 : LSI RC オシレータはレディ状態ではありません。

1 : LSI RC オシレータはレディ状態です。

ビット 0 LSION : 内部低速オシレータ有効化

ソフトウェアでセット/クリアされます。

0 : LSI RC オシレータオフ

1 : LSI RC オシレータオン

6.3.19 RCC スペクトル拡散クロック生成レジスタ (RCC_SSCGR)

アドレスオフセット : 0x80

リセット値 : 0x0000 0000

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

スペクトル拡散クロック生成は、メイン PLL に対してのみ利用可能です。

RCC_SSCGR レジスタは、メイン PLL が有効になる前またはメイン PLL が無効になった後に書き込む必要があります。

注 : PLL スペクトル拡散クロック生成 (SSCG) 特性の詳細については、デバイスのデータシートの電気的特性のセクションを参照してください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCG EN	SPR EAD SEL	予約済み		INCSTEP											
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
INCSTEP			MODPER												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31 **SSCGEN** : スペクトル拡散変調有効化
ソフトウェアでセット/クリアされます。
0 : スペクトル拡散変調無効 (書込みは クリア後 : CR[24]=PLLON ビット)
1 : スペクトル拡散変調有効 (書込みは セット前 : CR[24] = PLLON ビット)

- ビット 30 **SPREADSEL** : 拡散選択
ソフトウェアでセット/クリアされます。
CR[24]=PLLON ビットをセットする前に書き込みます。
0 : センタースプレッド
1 : ダウンスプレッド

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

- ビット 27:13 **INCSTEP** : インクリメントステップ
ソフトウェアでセット/クリアされます。CR[24]=PLLON ビットをセットする前に書き込みます。
変調プロファイルの振幅の設定入力です。

- ビット 12:0 **MODPER** : 変調周期
ソフトウェアでセット/クリアされます。CR[24]=PLLON ビットをセットする前に書き込みます。
変調プロファイルの周期の設定入力です。

6.3.20 RCC PLLI2S 設定レジスタ (RCC_PLLI2SCFGR)

アドレスオフセット : 0x84

リセット値 : 0x2400 3000

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

このレジスタは、次の式に従って PLLI2S クロック出力を設定するために使用します。

- $f_{\text{(VCO クロック)}} = f_{\text{(PLLI2S クロック入力)}} * (\text{PLLI2SN} / \text{PLLM})$
- $f_{\text{(PLL I2S クロック出力)}} = f_{\text{(VCO クロック)}} / \text{PLLI2SR}$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み	PLLI2S R2	PLLI2S R1	PLLI2S R0	予約済み											
	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	PLLI2SN8	PLLI2SN7	PLLI2SN6	PLLI2SN5	PLLI2SN4	PLLI2SN3	PLLI2SN2	PLLI2SN1	PLLI2S N0	予約済み					
	rw	rw	rw	rw	rw	rw	rw	rw	rw						

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:28 **PLLI2SR** : I²S クロックのための PLLI2S 分周比

I²S クロック周波数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。標準クリスタル使用時の誤差が 0.3 %、オーディオクリスタル使用時の誤差が 0 % となるよう、分周比は I²S ペリフェラル内のプリスケール値に従って選択する必要があります。I²S クロック周波数と精度に関する 詳細については、I²S の章の[セクション 20.4.4 : クロックジェネレータ](#)を参照してください。

注意 : I²S が正しく動作するには、192 MHz 以下の周波数が必要です。

I²S クロック周波数 = VCO 周波数 / PLLR、ここで $2 \leq \text{PLLR} \leq 7$

000 : PLLR = 0、誤った設定

001 : PLLR = 1、誤った設定

010 : PLLR = 2

...

111 : PLLR = 7

ビット 27:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:6 **PLLI2SN** : VCOのための PLLI2S 通倍数

VCO の通倍数を制御するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLLI2S が無効なときだけです。これらのビットの書き込みでは、ハーフワードおよびワード単位でのアクセスのみが可能です。

注意 : これらのビットは、VCO 出力周波数が 192~432 MHz の間になるように、ソフトウェアで正しく設定する必要があります。VCO 入力周波数範囲は 1~2 MHz (図 13 および [RCC PLL 設定レジスタ \(RCC_PLLCFGR\)](#) の分周比 M を参照)

VCO 出力周波数 = VCO 入力周波数 × PLLI2SN、ここで $192 \leq \text{PLLI2SN} \leq 432$

000000000 : PLLI2SN = 0、誤った設定

000000001 : PLLI2SN = 1、誤った設定

...

011000000 : PLLI2SN = 192

011000001 : PLLI2SN = 193

011000010 : PLLI2SN = 194

...

110110000 : PLLI2SN = 432

110110000 : PLLI2SN = 433、誤った設定

...

111111111 : PLLI2SN = 511、誤った設定

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

6.3.21 RCC 専用クロック設定レジスタ（RCC_DCKCFGR）

アドレスオフセット：0x8C

リセット値：0x0000 0000

アクセス：ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

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

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 **TIMPRE**：タイマクロックプリスケアラ選択

APB1 ドメインおよび APB2 ドメインに接続されているすべてのタイマのクロック周波数を制御するために、ソフトウェアによってセット／リセットされます。

0：APB プリスケアラ（RCC_CFGR レジスタの PPRE1、PPRE2）が分周比 1 に設定されている場合、TIMxCLK = HCKL となります。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 2 倍に設定されます：

$TIMxCLK = 2 * PCLKx$

1：APB プリスケアラ（RCC_CFGR レジスタの PPRE1、PPRE2）が分周比 1 または 2 のいずれかに設定されている場合、TIMxCLK = HCKL となります。そうでない場合、タイマクロック周波数は、タイマが接続されている APB ドメインの周波数の 4 倍に設定されます：

$TIMxCLK = 4 * PCLKx$ 。

ビット 23：0 予約済みであり、リセット値に保持する必要があります。

6.3.22 RCC レジスタマップ

表 22 に、レジスタマップとリセット値を示します

表 22. RCC レジスタマップとリセット値 STM32F401xB/C および STM32F401xD/E

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RCC_CR	予約済み				PLL12SRDY	PLL12SON	PLL1RDY	PLL1ON	予約済み				CSSON	HSEBYP	HSERDY	HSEON	HSICAL7	HSICAL6	HSICAL5	HSICAL4	HSICAL3	HSICAL2	HSICAL1	HSICAL0	HSITRIM4	HSITRIM3	HSITRIM2	HSITRIM1	HSITRIM0	予約済み	HSIRDY	HSION	
0x04	RCC_PLLCFGR	予約済み				PLLQ3	PLLQ2	PLLQ1	PLLQ0	予約済み	PLLSRC	予約済み				PLLP1	PLLP0	予約済み	PLLN8	PLLN7	PLLN6	PLLN5	PLLN4	PLLN3	PLLN2	PLLN1	PLLN0	PLLM5	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0	
0x08	RCC_CFGR	MCO21	MCO20	MCO2PRE2	MCO2PRE1	MCO2PRE0	MCO1PRE2	MCO1PRE1	MCO1PRE0	I2SSRC	MCO11	MCO10	RTCPRE4	RTCPRE3	RTCPRE2	RTCPRE1	RTCPRE0	PPRE22	PPRE21	PPRE20	PPRE12	PPRE11	PPRE10	予約済み		HPRE3	HPRE2	HPRE1	HPRE0	SWS1	SWS0	SW1	SW0	
0x0C	RCC_CIR	予約済み								CSSC	予約済み	PLL12SRDYC	PLL1RDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC	予約済み		PLL12SRDYIE	PLL1RDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	予約済み		PLL12SRDYF	PLL1RDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
0x10	RCC_AHB1RSTR	予約済み										DMA2RST	DMA1RST	予約済み								予約済み				GPIOHRST	予約済み	予約済み		GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
0x14	RCC_AHB2RSTR	予約済み																									OTGFSRST	予約済み						
0x18	予約済み	予約済み																																
0x1C	予約済み	予約済み																																
0x20	RCC_APB1RSTR	予約済み	PWRST	予約済み							I2C3RST	I2C2RST	I2C1RST	予約済み		USART2RST	予約済み	SPI3RST	SPI2RST	予約済み		WWDGRST	予約済み								TIM5RST	TIM4RST	TIM3RST	TIM2RST
0x24	RCC_APB2RSTR	予約済み														TIM11RST	TIM10RST	TIM9RST	予約済み	SYSCFGRST	SP45RST	SPI1RST	SDIORST	予約済み	ADC1RST	予約済み	USART6RST	USART1RST	予約済み		TIM1RST			
0x28	予約済み	予約済み																																
0x2C	予約済み	予約済み																																
0x30	RCC_AHB1ENR	予約済み										DMA2EN	DMA1EN	予約済み								CRCEEN	予約済み		GPIOHEN	予約済み	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN			
0x34	RCC_AHB2ENR	予約済み																									OTGFSEN	予約済み						
0x38	予約済み	予約済み																																
0x3C	予約済み	予約済み																																
0x40	RCC_APB1ENR	予約済み	PWREN	予約済み							I2C3EN	I2C2EN	I2C1EN	予約済み		USART2EN	予約済み	SPI3EN	SPI2EN	予約済み		WWDGEN	予約済み								TIM5EN	TIM4EN	TIM3EN	TIM2EN
0x44	RCC_APB2ENR	予約済み														TIM11EN	TIM10EN	TIM9EN	予約済み	SYSCFGEN	SPI4EN	SPI1EN	SDIOEN	予約済み	ADC1EN	予約済み	USART6EN	USART1EN	予約済み		TIM1EN			
0x48	予約済み	予約済み																																
0x4C	予約済み	予約済み																																
0x50	RCC_AHB1LPENR	予約済み										DMA2LPEN	DMA1LPEN	予約済み				SRAM1LPEN	FLITFLPEN	予約済み		CRCLPEN	予約済み				GPIOHLPEN	予約済み	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN	

表 22. RCC レジスタマップとリセット値 STM32F401xB/C および STM32F401xD/E（続き）

アドレス オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
0x54	RCC_AHB2LP ENR	予約済み																									OTGFSLPEN	予約済み																
0x58	予約済み	予約済み																																										
0x5C	予約済み	予約済み																																										
0x60	RCC_APB1LP ENR	予約済み		PWRLPEN		予約済み				I2C3LPEN		I2C2LPEN		I2C1LPEN		予約済み		USART2LPEN		予約済み		SPI3LPEN		SPI2LPEN		予約済み		WWDGLPEN		予約済み				TIM5LPEN		TIM4LPEN		TIM3LPEN		TIM2LPEN				
0x64	RCC_APB2LP ENR	予約済み													TIM11LPEN		TIM10LPEN		TIM9LPEN		予約済み		SYSCFGLPEN		SPI4LPEN		SPI1LPEN		SDIOLPEN		予約済み		ADC1LPEN		予約済み		USART6LPEN		USART1LPEN		予約済み		TIM1LPEN	
0x68	予約済み	予約済み																																										
0x6C	予約済み	予約済み																																										
0x70	RCC_BDCR	予約済み															BDRST		RTCEN		予約済み					RTCSSEL1		RTCSSEL0		予約済み					LSEBYP		LSERDY		LSEON					
0x74	RCC_CSR	LPWRRSTF		WWDGRSTF		WDGRSTF		SFTRSTF		PORRSTF		PADRSTF		BORRSTF		RMVF		予約済み																		LSIRDY		LSION						
0x78	予約済み	予約済み																																										
0x7C	予約済み	予約済み																																										
0x80	RCC_SSCGR	SSCGEN		SPREADSEL		予約済み		INCSTEP										MODPER																										
0x84	RCC_PLLI2SC FGR	予約済み		PLL12SRx		予約済み										PLL12SNx										予約済み																		
0x88		予約済み																																										
0x8C	RCC_DCKCFG R	予約済み								TIMPRE		予約済み																																

レジスタ境界アドレスについては、38 ページの表 1 を参照してください。

7 システム設定コントローラ (SYSCFG)

システム設定コントローラは、主にコード領域内でアクセス可能なメモリの再配置、および GPIO に対する外部割込みライン接続の管理に使用します。

7.1 I/O 補正セル

デフォルトでは I/O 補正セルは使用されません。しかしながら、I/O 出力バッファ速度が 50 MHz または 100 MHz モードに設定されている場合、I/O の転流 $t_{f(I/O)out}/t_{r(I/O)out}$ のスルーレート制御のため、補正セルを使用して電源の I/O ノイズを低減することを推奨します。

補正セルが有効になると、補正セルが使用可能であることを示すため READY フラグがセットされ、使用可能となります。I/O 補正セルは、電源電圧が 2.4 から 3.6 V の範囲にある場合にのみ使用できます。

7.2 SYSCFG レジスタ

7.2.1 SYSCFG メモリ再配置レジスタ (SYSCFG_MEMRMP)

このレジスタは、メモリ再配置時の特定の設定に使用されます。

- アドレス 0x0000 0000 でアクセス可能なメモリのタイプを設定するために、2 つのビットが使用されます。これらのビットを用いて、ソフトウェアで物理的な再配置を選択し、BOOT ピンをバイパスします。
- リセット後、これらのビットは、BOOT ピンで選択された値になります。BOOT0 ピンを 0 に設定してメインフラッシュメモリから起動する場合、このレジスタの値は 0x00 になります。

再割当てモードでは、CPU がシステムバスではなく ICode バスを通じて外部メモリにアクセスできるため、パフォーマンスを高めることができます。

アドレスオフセット : 0x00

リセット値 : 0x0000 000X (X は BOOT ピンで選択されたメモリモード)

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

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **MEM_MODE** : メモリマッピング選択

ソフトウェアでセット/クリアされます。これらのビットによって、アドレス 0x0000 0000 のメモリ内部マッピングを制御します。リセット後、これらのビットは、BOOT ピンで選択された値になります。

00 : メインフラッシュメモリは 0x0000 0000 に配置されます。

01 : システムフラッシュメモリは 0x0000 0000 に配置されます。

11 : 内蔵 SRAM は 0x0000 0000 に配置されます。

注 : アドレス 0x0000 0000 のメモリマッピングの詳細は [セクション 2.3: メモリマップ](#) を参照してください。

7.2.2 SYSCFG ペリフェラルモード設定レジスタ (SYSCFG_PMC)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															ADC1D C2
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み															

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 **ADCxDC2** :

0 : 影響はありません。

1 : このビットの使い方は、AN4073 を参照してください。

注 : これらのビットは、次の条件が満足されたときのみセットできます。

- ADC クロックが 30 MHz 以上
- 複数の ADC 変換が同時に開始せず、サンプリング時間が異なる場合、ADCxDC2 ビットは、ただ 1 つだけ選択する必要があります。
- PWR_CR レジスタで ADCDC1 ビットがセットされている場合、これらのビットをセットすることはできません。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

7.2.3 SYSCFG 外部割込み設定レジスタ 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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 0~3)

これらのビットは、EXTIx 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : 予約済み

0110 : 予約済み

0111 : PH[x] ピン

7.2.4 SYSCFG 外部割込み設定レジスタ 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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 4~7)

これらのビットは、EXTIx 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : 予約済み

0110 : 予約済み

0111 : PH[x] ピン

7.2.5 SYSCFG 外部割込み設定レジスタ 3 (SYSCFG_EXTICR3)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
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]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 8~11)

これらのビットは、EXTIx 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : 予約済み

0110 : 予約済み

0111 : PH[x] ピン

7.2.6 SYSCFG 外部割込み設定レジスタ 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
予約済み															
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]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定 (x= 12~15)

これらのビットは、EXTIx 外部割込みのソース入力を選択するために、ソフトウェアで書き込みます。

0000 : PA[x] ピン

0001 : PB[x] ピン

0010 : PC[x] ピン

0011 : PD[x] ピン

0100 : PE[x] ピン

0101 : 予約済み

0110 : 予約済み

0111 : PH[x] ピン

7.2.7 補正セル制御レジスタ (SYSCFG_CMPCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み							READY	予約済み						CMP_PD	
							r							rw	

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **READY** : 補正セルレディフラグ

- 0 : I/O 補正セルはレディ状態ではありません。
- 1 : I/O 補正セルはレディ状態です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CMP_PD** : 補正セルパワーダウン

- 0 : I/O 補正セルパワーダウンモード
- 1 : I/O 補正セルは有効です。

7.2.8 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 23. SYSCFG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SYSCFG_ MEMRMP	予約済み																														MEM_MODE	
	リセット値																															x	x
0x04	SYSCFG_PMC	予約済み															ADC1DC2	予約済み															
	リセット値																0																
0x08	SYSCFG_ EXTICR1	予約済み															EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]				
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	SYSCFG_ EXTICR2	予約済み															EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]				
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	SYSCFG_ EXTICR3	予約済み															EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]				
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	SYSCFG_ EXTICR4	予約済み															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	
0x20	SYSCFG_ CMPCR	予約済み																						READY	予約済み				CMP_PD				
	リセット値																							0					0				

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

8 汎用 I/O (GPIO)

GPIO F/G/H/I/J/K (GPIOH0 および GPIOH1 を除く) は、STM32F401xB/C および STM32F401xD/E で使用できません。

8.1 GPIO の概要

各汎用 I/O ポートにはそれぞれ 4 つの 32 ビット設定レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR)、2 つの 32 ビットデータレジスタ (GPIOx_IDR、GPIOx_ODR)、32 ビットセット/リセットレジスタ (GPIOx_BSRR)、32 ビットロックレジスタ (GPIOx_LCKR)、2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx_AFRH、GPIOx_AFRL) があります。

8.2 GPIO の主な機能

- 最大 16 個の I/O を制御
- 出力状態: プッシュプルまたはオープンドレイン + プルアップ/プルダウン
- 出力データレジスタ (GPIOx_ODR) またはペリフェラル (オルタネート機能出力) からの出力データ
- 各 I/O のスピード選択
- 入力状態: フローティング、プルアップ/プルダウン、アナログ
- 入力データレジスタ (GPIOx_IDR) またはペリフェラル (オルタネート機能入力) への入力データ
- GPIOx_ODR へのビット単位の書込みアクセス用のビットセット/リセットレジスタ (GPIOx_BSRR)
- I/O 設定を凍結可能にするロック機構 (GPIOx_LCKR)
- アナログ機能
- オルタネート機能入力/出力選択レジスタ (入出力ごとに最大 16 のオルタネート機能)
- 2 クロックサイクルで変化可能な高速トグル
- 柔軟性の高いピンの多重化により、I/O ピンを GPIO またはいくつかのペリフェラル機能の 1 つとして使用可能

8.3 GPIO の機能説明

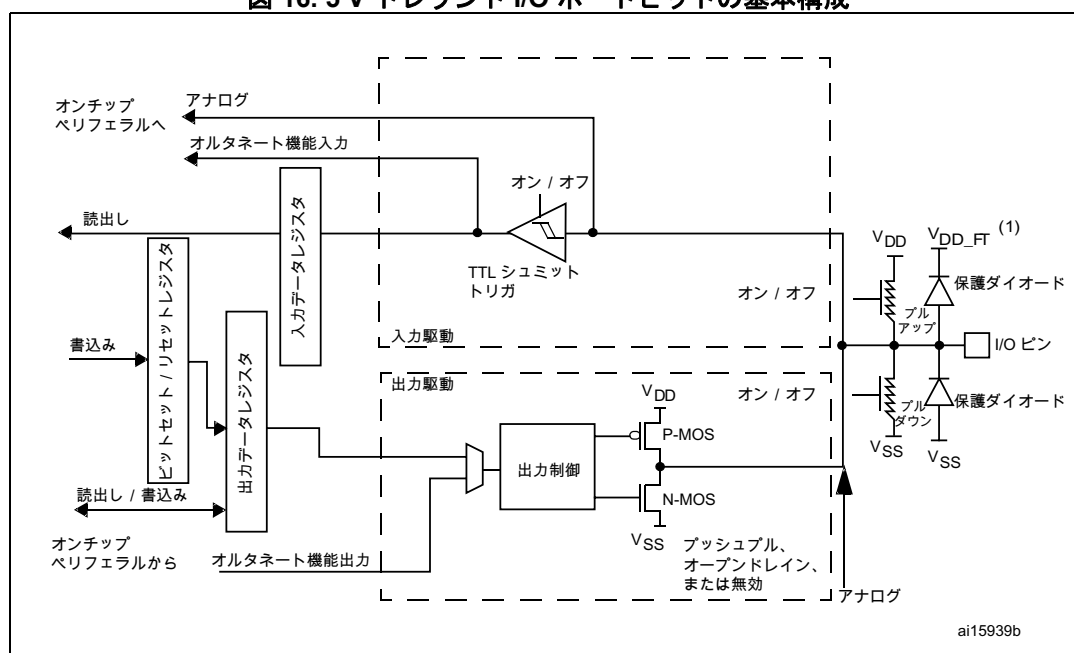
各 I/O ポートの特定のハードウェア特性については、データシートに記載されています。汎用 I/O (GPIO) ポートの各ポートビットは、ソフトウェアによって以下の動作モードを個別に設定できます。

- 入力フローティング
- 入力プルアップ
- 入力プルダウン
- アナログ
- プルアップまたはプルダウン機能を持つ出力オープンドレイン
- プルアップまたはプルダウン機能を持つ出力プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能オープンドレイン

各 I/O ポートビットは自由にプログラム可能ですが、I/O ポートレジスタには 32 ビットワード、ハーフワード、またはバイト単位でアクセスする必要があります。GPIOx_BSRR レジスタを使用すると、任意の GPIO レジスタにビット単位の読出し／変更アクセスを行うことができます。これにより、読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16に 5 V トレラント I/O ポートビットの基本構成を示します。表 27 には、実行可能なポートビットの設定を示します。

図 16. 5 V トレラント I/O ポートビットの基本構成



1. V_{DD_FT} は、5 V トレラント I/O 固有の電位であり、 V_{DD} とは異なります。

表 24. ポートビット設定表⁽¹⁾

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O 設定	
01	0	SPEED [B:A]		0	0	GP 出力	PP
	0			0	1	GP 出力	PP + PU
	0			1	0	GP 出力	PP + PD
	0			1	1	予約済み	
	1			0	0	GP 出力	OD
	1			0	1	GP 出力	OD + PU
	1			1	0	GP 出力	OD + PD
	1			1	1	予約済み (GP 出力 OD)	
10	0	SPEED [B:A]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	予約済み	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	予約済み	
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

8.3.1 汎用 I/O (GPIO)

リセット中とリセット直後は、オルタネート機能は有効ではなく、I/O ポートは入力フローティングモードに設定されています。

リセット後、デバッグピンはオルタネート機能のプルアップ/プルダウンに設定されています。

- PA15 : JTDI プルアップ
- PA14 : JTCK/SWCLK プルダウン
- PA13 : JTMS/SWDAT プルアップ
- PB4 : NJTRST プルアップ
- PB3 : JTDO フローティング状態

ピンが出力として設定されている場合、出力データレジスタ (GPIOx_ODR) に書き込まれた値が I/O ピンに出力されます。出力ドライバをプッシュプルモードまたはオープンドレインモードで 사용할 ことができます (0 を出力すると、N-MOS のみが有効になります)。

入力データレジスタ (GPIOx_IDR) は、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。

8.3.2 I/O ピンマルチプレクサとマッピング

本マイクロコントローラの I/O ピンは、マルチプレクサを通してオンボードのペリフェラル/モジュールに接続され、一度に 1 つのペリフェラルオルタネート機能 (AF) のみが 1 つの I/O ピンに接続可能となっています。この方法により、同じ I/O ピンを共有するペリフェラル間での競合を無くすことができます。

各 I/O ピンは、16 のオルタネート機能入力 (AF0 ~ AF15) を持つマルチプレクサを持ち、これらの入力は、GPIOx_AFRL (ピン 0 ~ 7) と GPIOx_AFRH (ピン 8 から 15) レジスタを通して設定することができます。

- リセット後、すべての I/O はシステムのオルタネート機能 0 (AF0) に接続されます。
- ペリフェラルのオルタネート機能は AF1~AF13 に配置されます
- FPU 搭載 Cortex®-M4 EVENTOUT は AF15 に配置されます。

この仕組みを次の [図 17](#) に示します。

この柔軟性の高い I/O 多重化アーキテクチャに加え、各ペリフェラルではオルタネート機能がそれぞれの I/O ピンに配置されており、さらに小型のパッケージで利用できるペリフェラルの数を最適化します。

I/O を任意の設定で使用するには、次の手順に従います。

- **システム機能**
 - I/O を AF0 に接続し、使用する機能に応じて設定します。
 - JTAG/SWD : 各デバイスリセット後、これらのピンはデバッガーストが直ちに使用できる専用ピンとして割り当てられます (GPIO コントローラでは制御されません)
 - RTC_REFIN : このピンは入力フローティングモードに設定する必要があります。
 - MCO1 および MCO2 : これらのピンはオルタネート機能モードに設定する必要があります。

注 : JTAG/SWD ピンの一部またはすべてを無効にすることができるため、該当するピンを GPIO としても使用できます。

詳細については、[セクション 6.2.10 : クロック信号出力](#)を参照してください。

表 25. 柔軟性の高い SWJ-DP ピンの割り当て

使用可能なデバッグポート	SWJ I/O ピン配置				
	PA13 / JTMS / SWDIO	PA14 / JTCK / SWCLK	PA15 / JTDI	PB3 / JTDO	PB4 / NJTRST
全 SWJ (JTAG-DP + SW-DP) - リセット状態	X	X	X	X	X
全 SWJ (JTAG-DP + SW-DP)、NJTRST なし	X	X	X	X	
JTAG-DP 無効、SW-DP 有効	X	X			
JTAG-DP 無効、SW-DP 無効	解放				

- **GPIO**

必要とする I/O を、GPIOx_MODER レジスタで出力または入力として設定します。

- **ペリフェラルオルタネート機能**

ADC については、必要とする I/O を GPIOx_MODER レジスタでアナログとして設定します。
他のペリフェラルでは次のように設定します。

- 必要とする I/O を、GPIOx_MODER レジスタでオルタネート機能として設定します。
- タイプ、プルアップ／プルダウン、出カ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。
- I/O を GPIOx_AFR1 または GPIOx_AFRH レジスタで、必要とする AFx に接続します。

- **EVENTOUT**

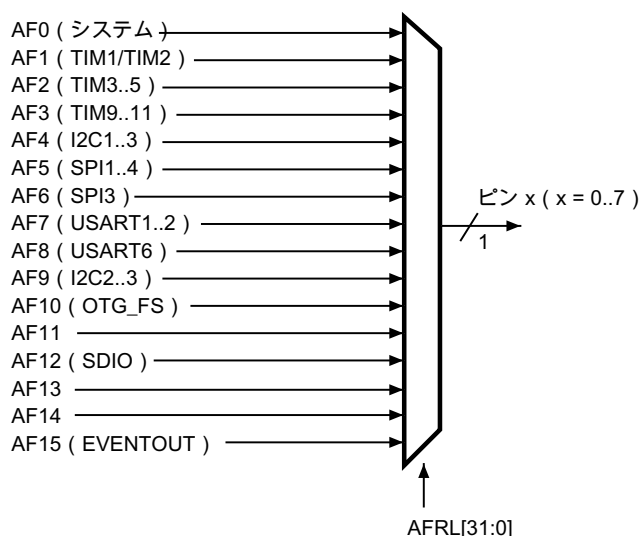
AF15 に接続することで、I/O ピンを FPU 搭載 Cortex®-M4 EVENTOUT 信号を出力する用途に設定します。

注： **EVENTOUT は次の I/O ピンには配置されません：PC13、PC14、PC15、PH0 および PH1。**

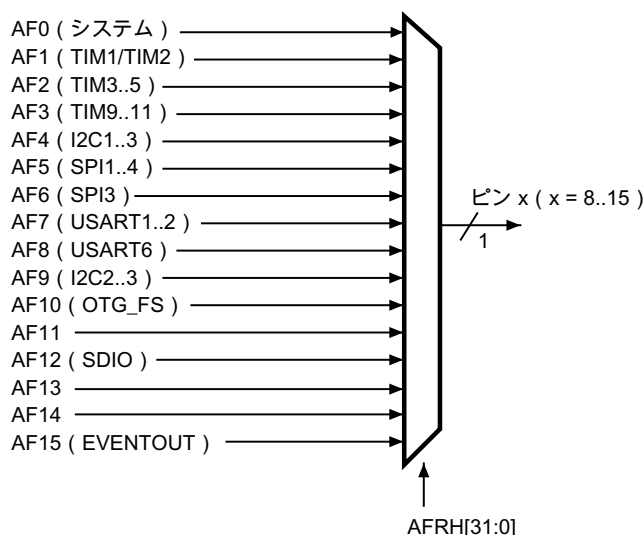
システムおよびペリフェラルのオルタネート機能 I/O ピンの詳細配置については、データシートの「オルタネート機能配置」表を参照してください。

図 17. オルタネート機能の選択 : STM32F401xB/C および STM32F401xD/E

ピン 0~7 に対し、GPIOx_AFRL[31:0] レジスタで専用のオルタネート機能を選択します



ピン 8~15 に対し、GPIOx_AFRH[31:0] レジスタで専用のオルタネート機能を選択します



MS31422V1

1. FS で設定します。

8.3.3 I/O ポート制御レジスタ

各 GPIO には 32 ビットメモリマップド制御レジスタが 4 つ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR) あり、最大 16 個の I/O を設定します。

GPIOx_MODER レジスタは I/O 方向 (入力、出力、AF、アナログ) を選択するために使用されます。GPIOx_OTYPER および GPIOx_OSPEEDR レジスタは、出力タイプ (プッシュプルまたはオープンドレイン) および速度を選択するために使用されます (I/O の方向がどちらであっても、I/O 速度ピンは、対応する GPIOx_OSPEEDR レジスタの各ビットに直接接続されています)。I/O の方向がどちらであっても、GPIOx_PUPDR レジスタは、プルアップ/プルダウンを選択するために使用されます。

8.3.4 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、読出し/書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 8.4.5 : GPIO ポート入力データレジスタ \(GPIOx_IDR\) \(x = A~E, H\)](#) および [セクション 8.4.6 : GPIO ポート出力データレジスタ \(GPIOx_ODR\) \(x = A~E, H\)](#) を参照してください。

8.3.5 I/O データのビット単位の操作

ビットセット/リセットレジスタ (GPIOx_BSRR) は 32 ビットレジスタで、アプリケーションによる出力データレジスタ (GPIOx_ODR) のビット単位のセット/リセットを可能にします。ビットセット/リセットレジスタは GPIOx_ODR の 2 倍のサイズです。

GPIOx_ODR の各ビットには GPIOx_BSRR の 2 つの制御ビット BS(i) と BR(i) が対応します。BSRR(i) および BSRR(i+SIZE) です。ビット BSRR(i) に 1 を書き込むと、対応する ODR(i) ビットがセットされます。ビット BSRR(i+SIZE) に 1 を書き込むと、対応する ODR(i) ビットがリセットされます。

GPIOx_BSRR のいかなるビットに 0 を書き込んでも GPIOx_ODR の対応するビットには影響しません。仮に、GPIOx_BSRR のビットに対してセットおよびリセットの両方を実行しようとした場合、セット動作が優先されます。

GPIOx_BSRR レジスタを使用した GPIOx_ODR 内の個々のビットの変更には、1 回限りの効果しかなく、GPIOx_ODR ビットを固定するものではありません。GPIOx_ODR のビットは常に直接アクセスすることができます。GPIOx_BSRR レジスタによって、ビット単位の不可分操作を行うことができます。

GPIOx_ODR をビットレベルでプログラムする場合、ソフトウェアで割込みを無効にする必要はありません。不可分な AHB1 の書込みアクセスで 1 ビットまたは複数ビットを変更することができます。

8.3.6 GPIO ロック機構

GPIOx_LCKR レジスタへ特定の書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。停止されるレジスタは、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、特定の書込み/読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 (この書込みシーケンス中、LCKR[15:0] の値は同じである必要があります)。あるポートビットに LOCK シーケンスが行われると、次の MCU またはペリフェラルリセットまで、そのポートビットの値を変更できなくなります。GPIOx_LCKR の各ビットによって、制御レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH) の対応するビットがロックされます。

GPIOx_LCKR ビット 16 を [15:0] ビットと同時に設定する必要があるため、この LOCK シーケンス ([セクション 8.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx_LCKR\) \(x = A~E, H\)](#) を参照) は、GPIOx_LCKR レジスタへのワード (32 ビット長) アクセスを使用してのみ実行できます。

詳細については、[セクション 8.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx_LCKR\) \(x = A~E, H\)](#) の LCKR レジスタの説明を参照してください。

8.3.7 I/O オルタネート機能の入力／出力

各 I/O が使用できる 16 個のオルタネート機能入力／出力の 1 つを選択するため、2 つのレジスタが用意されています。これらのレジスタを使用し、必要に応じて、アプリケーションでオルタネート機能を他のピンに接続することができます。

つまり、GPIOx_AFRL および GPIOx_AFRH オルタネート機能レジスタを使用していくつかの使用可能なペリフェラル機能が、各 GPIO に多重化されることになります。こうして各 I/O に使用可能な機能のどれか 1 つをアプリケーションによって選択できます。AF 選択信号はオルタネート機能入力およびオルタネート機能出力に共通なので、1 つの I/O が持つオルタネート機能入力／出力に対し 1 つのチャンネルが選択されます。

どの機能が各 GPIO ピンに多重化されているかについてはデータシートを参照してください。

注： アプリケーションからは、各 I/O に対して使用可能な機能を一度に 1 つだけ選択することができます。

8.3.8 外部割込み／ウェイクアップライン

すべてのポートに外部割込み機能があります。外部割込みラインを使用するには、ポートを入力モードに設定する必要があります。[セクション 10.2：外部割込み／イベントコントローラ \(EXTI\)](#) および [セクション 10.2.3：ウェイクアップイベント管理](#) を参照してください。

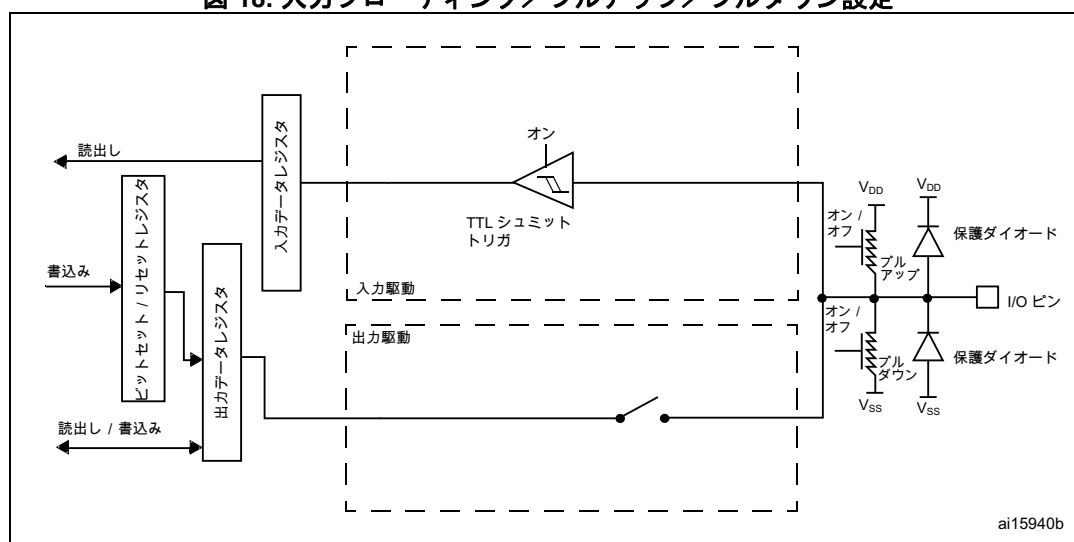
8.3.9 入力設定

I/O ポートが入力としてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力が有効になります。
- GPIOx_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、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出しアクセスによって、I/O 状態が得られます。

図 18 は、I/O ポートビットの入力設定を示しています。

図 18. 入力フローティング／プルアップ／プルダウン設定



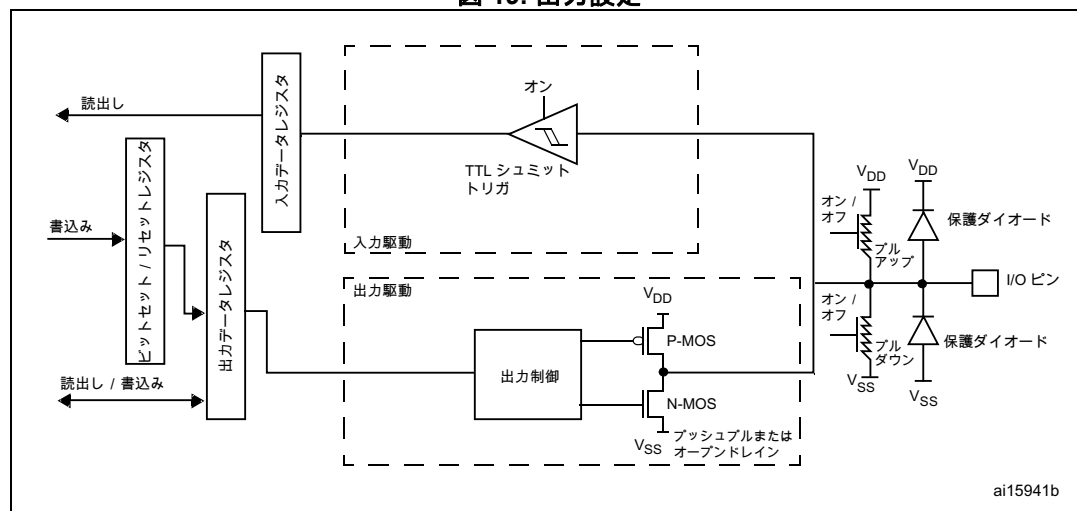
8.3.10 出力設定

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 that 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 19 は、I/O ポートビットの出力設定を示しています。

図 19. 出力設定



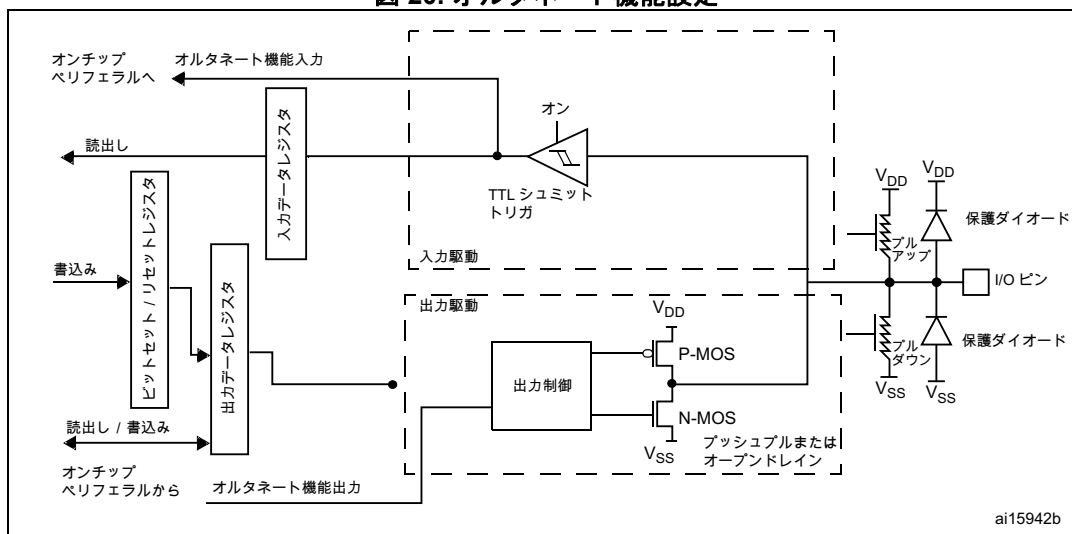
8.3.11 オルタネート機能設定

I/O ポートがオルタネート機能としてプログラムされた場合、

- 出力バッファをオープンドレインまたはプッシュプルとして設定することができます。
- 出力バッファが、ペリフェラル（トランスミッタインペブルおよびデータ）からの信号で駆動されます。
- シュミットトリガ入力 that 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出しアクセスによって、I/O 状態が得られます。

図 20 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 20. オルタネート機能設定



8.3.12 アナログ設定

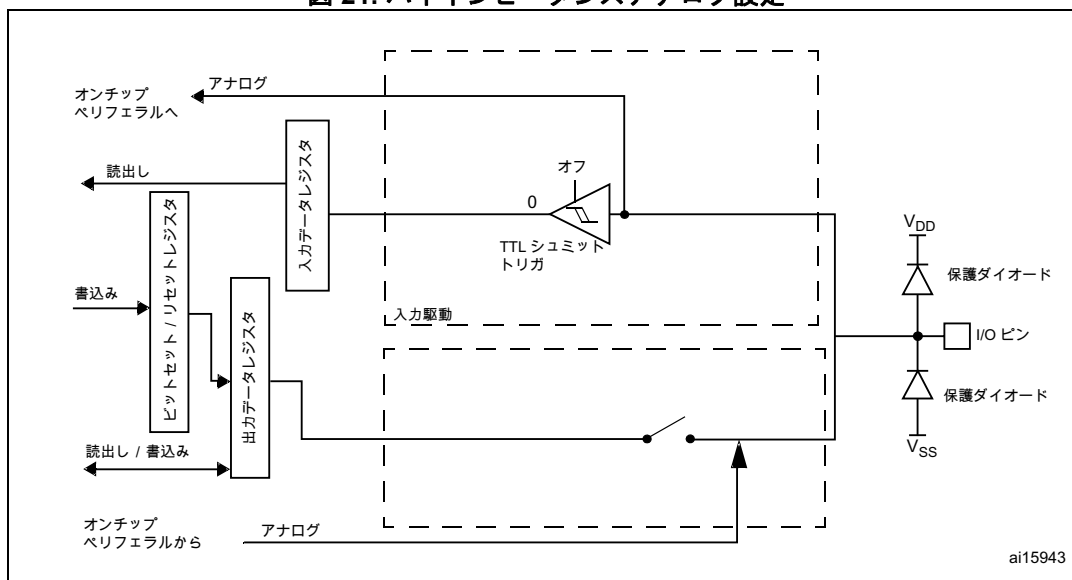
I/O ポートがアナログとしてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力は無効になり、I/O ピンのどのようなアナログ値に対しても消費電力をゼロに抑えます。シュミットトリガ回路の出力は、常に“0”になります。
- ウィークプルアップ抵抗およびプルダウン抵抗は無効になります。
- 入力データレジスタの読出しアクセスを行うと、値“0”が得られます。

注：アナログ設定では、I/O ピンは 5 V 耐性にできません。

図 21 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 21. ハイインピーダンスアナログ設定



8.3.13 OSC32_IN/OSC32_OUT ピンの GPIO PC14/PC15 ポートピンとしての使用

LSE オシレータがオフのとき、LSE オシレータのピン OSC32_IN および OSC32_OUT は、それぞれ汎用 I/O の PC14 および PC15 として使用できます。LSE オシレータがオンのとき、PC14 および PC15 の I/O は、LSE オシレータピン OSC32_IN および OSC32_OUT としてのみ設定できます。この設定は、RCC_BDCR レジスタの LSEON ビットをセットすることによって行います。LSE は、GPIO 機能より優先順位が高くなっています。

注： 1.2 V ドメインの電源がオフになった（デバイスが STANDBY モードに入った）とき、またはバックアップドメインに V_{BAT} から電源供給された（ V_{DD} が供給されない）とき、PC14/PC15 の GPIO が機能しなくなります。この場合、I/O はアナログ入力モードに設定されます。

8.3.14 OSC_IN/OSC_OUT ピンの GPIO PH0/PH1 ポートピンとしての使用

HSE オシレータがオフのとき、HSE オシレータのピン OSC_IN/OSC_OUT は、それぞれ汎用 I/O の PH0/PH1 として使用できます。（リセット後、HSE オシレータはオフになります）。HSE オシレータがオンのとき、PH0/PH1 I/O は、OSC_IN/OSC_OUT HSE オシレータピンとしてのみ設定できます。この設定は、RCC_CR レジスタの HSEON ビットをセットすることによって行います。HSEは、GPIO 機能より優先順位が高くなっています。

8.3.15 RTC 機能の選択

STM32F4xx には、1 つの GPIO ピン RTC_AF1 があり、タンパまたはタイムスタンプイベントの検出に使用でき、RTC_ALARM または RTC_CALIB の RTC 出力としても使用できます。

- RTC_AF1 (PC13) は次の目的に使用できます。

RTC_ALARM 出力：この出力は RTC_CR レジスタの OSEL[1:0] ビットによって RTC アラーム A、RTC アラーム B、RTC ウェイクアップのいずれかとして使用できます

- RTC_CALIB 出力：この機能は RTC_CR レジスタの COE[23] をセットすることで有効になります
- RTC_TAMP1：タンパイベント検出
- RTC_TS：タイムスタンプイベント検出

対応するピンの選択は次のように RTC_TAFCR レジスタを通して行います。

- TAMP1INSEL を使用して、どのピンを RTC_TAMP1 タンパ入力として使用するか選択します。
- TSINSEL を使用して、どのピンを RTC_TS タイムスタンプ入力として使用するか選択します。
- ALARMOUTTYPE を使用して、RTC_ALARM をプッシュプルモード出力とするかオープンドレインモード出力とするか選択します。

出力形式は 表 26 に一覧化された優先順位に従います。

表 26. RTC 追加機能⁽¹⁾

ピンの設定と機能	RTC_ALARM 有効	RTC_CALIB 有効	タンパ 有効	タイム スタンプ 有効	TAMP1INSEL AMPER1 ピン 選択	TSINSELTIME STAMP ピン選 択	ALARMOUTTYPE RTC_ALARM 設定
アラーム出力 OD	1	無視	無視	無視	無視	無視	0
アラーム出力 PP	1	無視	無視	無視	無視	無視	1
較正出力 PP	0	1	無視	無視	無視	無視	無視
TAMPER1 入力 フローティング	0	0	1	0	0	無視	無視
TIMESTAMP および TAMPER1 入力フローティング	0	0	1	1	0	0	無視
TIMESTAMP 入力フローティング	0	0	0	1	無視	0	無視
標準 GPIO	0	0	0	0	無視	無視	無視

1. OD : オープンドレイン、PP : プッシュプル

8.4 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 27 を参照してください。

GPIO レジスタはバイト (8 ビット)、ハーフワード (16 ビット)、ワード (32 ビット) のいずれかの単位でアクセスできます。

8.4.1 GPIO ポートモードレジスタ (GPIOx_MODER) (x = A~E、H)

アドレスオフセット : 0x00

リセット値 :

- ポート A 0x0C00 0000
- ポート B 0x0000 0280
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 2y : 2y+1 **MODERy[1:0]** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O 方向モードを設定するために、ソフトウェアによって書き込まれます。

00 : 入力 (リセット状態)

01 : 汎用出力モード

10 : オルタネート機能モード

11 : アナログモード

8.4.2 GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A~E、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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **OTy** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O ポートの出力タイプを設定するために、ソフトウェアによって書き込まれます。

0 : 出力プッシュプル (リセット状態)

1 : 出力オープンドレイン

8.4.3 GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A~E、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]	
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
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 2y : 2y+1 **OSPEEDRy[1:0]** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O の出力スピードを設定するために、ソフトウェアによって書き込まれます。

00 : ロースピード

01 : ミディアムスピード

10 : 準高速

11 : ハイスピード

注 : **OSPEEDRy** ビット値対 V_{DD} 範囲および外部負荷については製品データシートを参照してください。

8.4.4 GPIO ポートプルアップ／プルダウンレジスタ (GPIOx_PUPDR) (x = A~E、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]	
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
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 2y : 2y+1 **PUPDRy[1:0]** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O のプルアップまたはプルダウンを設定するために、ソフトウェアによって書き込まれます。

00 : プルアップ／プルダウンなし

01 : プルアップ

10 : プルダウン

11 : 予約済み

8.4.5 GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A~E、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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **IDRy** : ポート入力データ (x = 0 .. 15)

これらのビットは、読み出し専用であり、ワードモードでのみアクセスできます。これらのビットには、対応する I/O ポートの入力値が格納されています。

8.4.6 GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A~E、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
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ODRy** : ポート出力データ (y = 0..15)

これらのビットは、ソフトウェアによって読み出し／書き込みができます。

注 : ビット単位のセット／リセットは、GPIOx_BSRR レジスタ (x = A~E、H) への書き込みによって、ODR ビットを個々にセット／リセットできます。

8.4.7 GPIO ポートビットセット／リセットレジスタ (GPIOx_BSRR)
(x = A~E、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、書込み専用であり、ワードモード、ハーフワードモード、バイトモードのいずれかでアクセスできます。これらのビットを読み出すと、値 0x0000 が返されます。
0 : 対応する ODRx ビットの値は変化しません。
1 : 対応する ODRx ビットをリセットします。
注 : **BSx** ビットと **BRx** ビットの両方がセットされた場合、**BSx** ビットが優先されます。

ビット 15:0 **BSy** : ポート x セットビット y (y= 0~15)
これらのビットは、書込み専用であり、ワードモード、ハーフワードモード、バイトモードのいずれかでアクセスできます。これらのビットを読み出すと、値 0x0000 が返されます。
0 : 対応する ODRx ビットの値は変化しません。
1 : 対応する ODRx ビットをセットします。

8.4.8 GPIO ポート設定ロックレジスタ (GPIOx_LCKR)
(x = A~E、H)

このレジスタは、ビット 16 (LCKK) に正しい書込みシーケンスが行われた際に、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。書込みシーケンスの間は、LCKR[15:0] の値を変更することはできません。あるポートビットに LOCK シーケンスが適用されると、次の MCU またはペリフェラルリセットまで、このポートビットの値を変更できなくなります。

注 : **GPIOx_LCKR** レジスタへの書込みには特定の書込みシーケンスが使われます。この書込みシーケンスではワードアクセス (32 ビット長) のみ可能です。

各ロックビットによって、特定の設定レジスタ (制御レジスタおよびオルタネート機能レジスタ) が固定されます。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

アクセス : 32 ビットワードのみ、読出し/書込みレジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															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



8.4.10 GPIO オルタネート機能上位レジスタ (GPIOx_AFRH) (x = A~E、H)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]			
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
AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **AFRHy** : ポート x ビット y (y = 8..15) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRHy 選択 :

0000 : AF0	1000 : AF8
0001 : AF1	1001 : AF9
0010 : AF2	1010 : AF10
0011 : AF3	1011 : AF11
0100 : AF4	1100 : AF12
0101 : AF5	1101 : AF13
0110 : AF6	1110 : AF14
0111 : AF7	1111 : AF15

8.4.11 GPIO レジスタマップ

次の表に、GPIO レジスタマップとリセット値を示します。

表 27. GPIO レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_ MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	リセット値	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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～E、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～E、H)	予約済み																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

表 27. 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
0x08	GPIOx_ OSPEEDER (x = C～E、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
0x08	GPIOA_ OSPEEDER	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_ OSPEEDER	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	1	1	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
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	0	1	0	0	0	0	0	0	0
0x0C	GPIOx_PUPDR (x = C～E、H)	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	0	0	0	0	0	0	0	0	0
0x10	GPIOx_IDR (x = A～E、H)	予約済み															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～E、H)	予約済み															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～E、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 = A～E、H)	予約済み															LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (x = A～E、H)	AFRL7[3:0]			AFRL6[3:0]			AFRL5[3:0]			AFRL4[3:0]			AFRL3[3:0]			AFRL2[3:0]			AFRL1[3:0]			AFRL0[3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (x = A～E、H)	AFRH15[3:0]			AFRH14[3:0]			AFRH13[3:0]			AFRH12[3:0]			AFRH11[3:0]			AFRH10[3:0]			AFRH9[3:0]			AFRH8[3:0]										
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、38 ページの表 1 を参照してください。

9 DMA コントローラ (DMA)

9.1 DMA の概要

DMA (Direct Memory Access) は、ペリフェラルとメモリ間、およびメモリ間でハイスピードデータ転送を行うために使用します。データは、CPU の動作を必要とすることなく、DMA によって高速に移動できます。これにより、CPU リソースを他の操作のためにあけておくことができます。

このDMA コントローラは、高機能なバスマトリックスアーキテクチャに基づいてシステムの帯域幅を最大限に利用するため、強力なデュアル AHB マスタバスアーキテクチャと、独立した FIFO を兼ね備えています。

2 つの DMA コントローラは合計 16 ストリーム (各コントローラごとに 8 つ) を搭載し、それぞれが 1 つ以上のペリフェラルからのメモリアセスクエストを管理します。各ストリームは、合計で最大 8 チャンネル (リクエスト) を処理することができます。また、それぞれ DMA リクエスト間の優先順位を操作するためのアービタを内蔵しています。

9.2 DMA の主な機能

DMA の主な機能は次のとおりです。

- デュアル AHB マスタバスアーキテクチャ (メモリアクセス専用、およびペリフェラルへのアクセス専用)
- 32 ビットアクセスのみをサポートする AHB スレーブプログラミングインタフェース
- DMA コントローラごとに 8 ストリーム、ストリームごとに最大 8 チャンネル (リクエスト)
- ストリームごとに独立した 4 つの 32 ビットファーストインファーストアウトメモリバッファ (FIFO)。 (FIFO モードまたはダイレクトモードで使用可能)
 - FIFO モード: FIFO サイズの 1/4、1/2 または 3/4 からソフトウェアで選択可能な閾値レベルを持つ
 - ダイレクトモード
各 DMA リクエストによって、直ちにメモリとの転送を開始します。DMA がダイレクトダイレクトモード (FIFO が無効) に設定されている場合、“メモリからペリフェラル”モードでデータを転送するため、ペリフェラルからの DMA リクエストが発生した際、即座にかつ確実にデータ転送が行われるよう、DMA はデータをメモリから内部 FIFO へ 1 つだけプリロードします。
- 各ストリームはハードウェアによって次のように設定することができます。
 - “ペリフェラルからメモリ”、“メモリからペリフェラル”および“メモリ間”転送をサポートするレギュラチャネル
 - メモリ側でダブルバッファリングもサポートするダブルバッファチャネル
- 8 つのストリームのそれぞれは、専用のハードウェア DMA チャネル (リクエスト) に接続されます。
- DMA ストリームリクエスト間の優先順位は、ソフトウェアによってプログラミングできます (最高、高、中、低の 4 レベル)。レベルが等しい場合はハードウェアによって決まります (リクエスト 0 はリクエスト 1 よりも優先、など) です。
- 各ストリームは、メモリ間転送のためのソフトウェアトリガもサポートしています (DMA2 コントローラでのみ使用可能)。
- 各ストリームリクエストは、発生しうる最大 8 つのチャネルリクエストから選択することができます。この選択は、ソフトウェアによって設定可能であり、これによっていくつかのペリフェラルが DMA リクエストを発生することができるようになります。

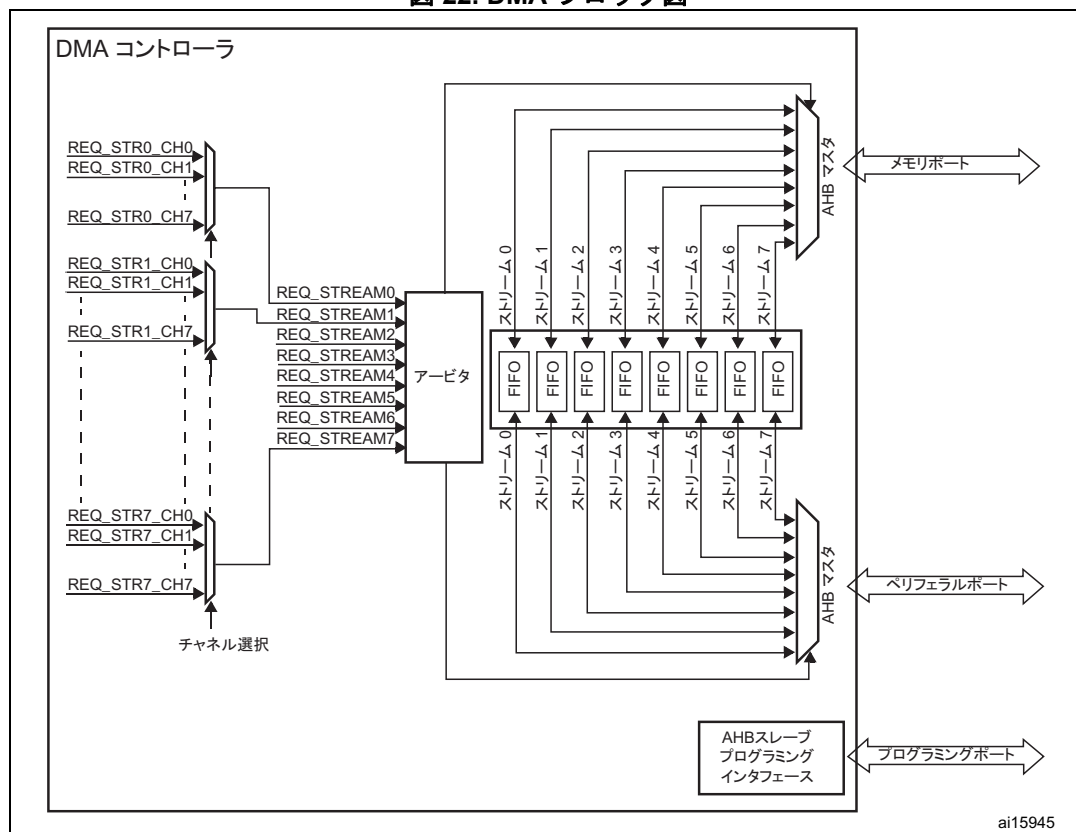
- 転送するデータの数、DMA コントローラまたはペリフェラルのいずれかによって管理することができます。
 - DMA フローコントローラ：転送するデータ数は、1 から 65535 の間でソフトウェアによってプログラム可能です。
 - ペリフェラルフローコントローラ：転送されるデータ項目の数は不明であり、ハードウェアによって転送の終了を通知する転送元または転送先ペリフェラルによって制御されます。
- 転送元および転送先の転送幅（バイト、ハーフワード、ワード）に依存しない：転送元と転送先のデータ幅が異なる場合、DMA によって自動的に必要な転送のパック／アンパックが行われ、バンド幅が最適化されます。この機能は、FIFO モードでのみ使用できます。
- 転送元および転送先に対するインクリメントアドレッシングまたはノンインクリメントアドレッシング
- 4、8、または 16 ビートのインクリメンタルバースト転送をサポート。バーストのサイズはソフトウェアで設定可能であり、通常はペリフェラルの FIFO サイズの半分です。
- サークュラバッファ管理は、各ストリームがサポートします。
- 各ストリームに対する同一の割込みリクエストにおいて、5 つのイベントフラグ（DMA 1/2 転送、DMA 転送完了、DMA 転送エラー、DMA FIFO エラー、ダイレクトモードエラー）の論理和がとられます。

9.3 DMA の機能説明

9.3.1 概要

図 22 に、DMA のブロック図を示します。

図 22. DMA ブロック図



DMA コントローラはダイレクトメモリ転送を行います。AHB マスタとして、AHB トランザクションを開始するための AHB バスマトリックスの制御を行うことができます。

以下のトランザクションを実行することができます。

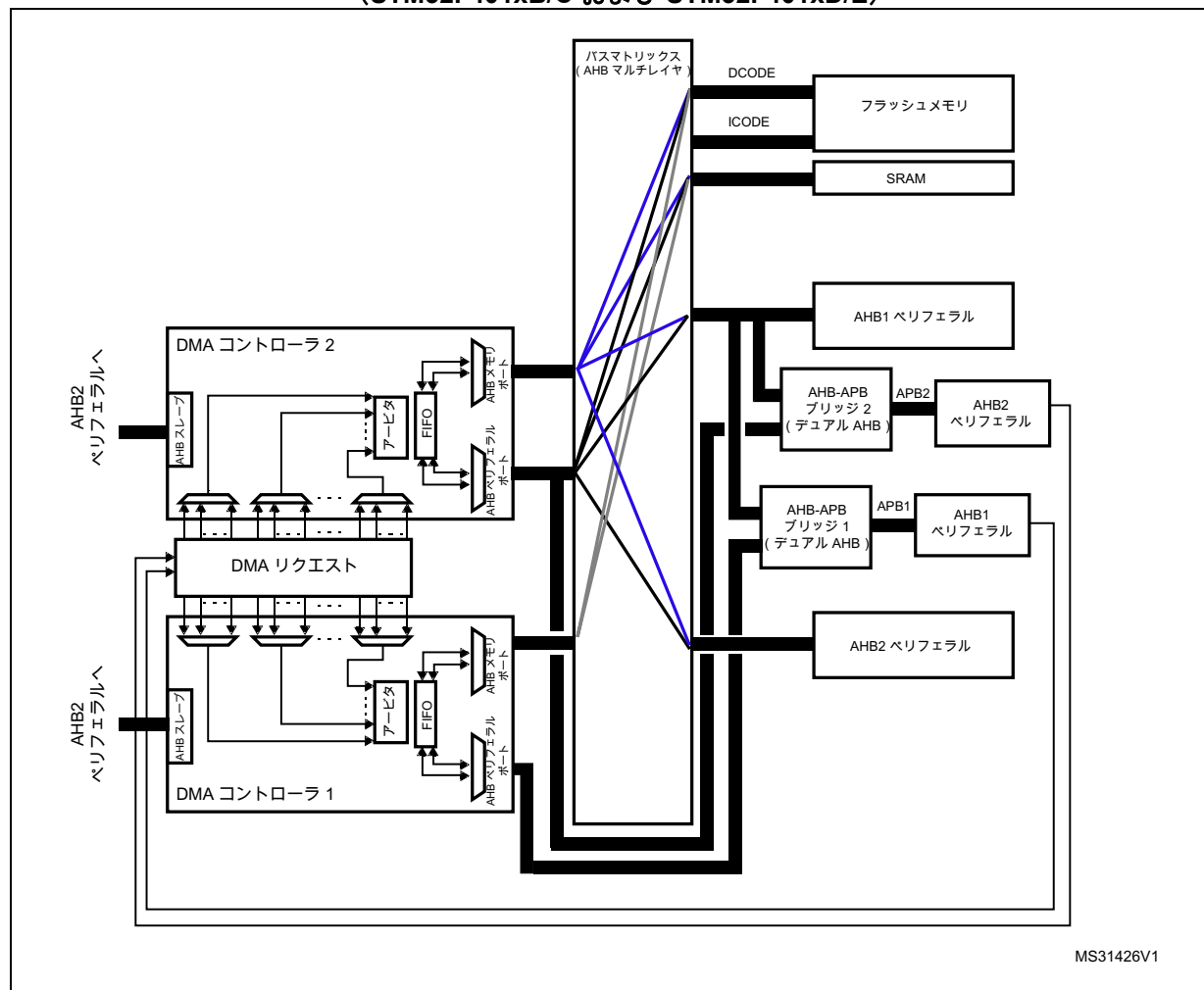
- ペリフェラルからメモリ
- メモリからペリフェラル
- メモリ間

この DMA コントローラは 2 つの AHB マスタポートを搭載しています。メモリに接続するための AHB メモリポート、およびペリフェラルに接続するための AHB ペリフェラルポートです。しかし、メモリ間転送を可能とするため、AHB ペリフェラルポートはメモリにもアクセスする必要があります。

AHB スレーブポートは DMA コントローラをプログラムするために用いられます (32 ビットアクセスのみサポート)。

2 つの DMA コントローラのシステムへの実装については図 23 を参照してください。

図 23. 2 つの DMA コントローラのシステム実装
(STM32F401xB/C および STM32F401xD/E)



1. DMA1 コントローラの AHB ペリフェラルポートは、DMA2 コントローラのようにバスマトリックスに接続されていません。そのため、DMA2 ストリームだけがメモリ間転送を実行できます。

9.3.2 DMA トランザクション

DMA トランザクションは、一定数のデータ転送の列で構成されています。転送されるデータ項目の数とその幅（8 ビット、16 ビット、または 32 ビット）は、ソフトウェアでプログラム可能です。

各 DMA 転送は次の 3 つの操作で構成されます。

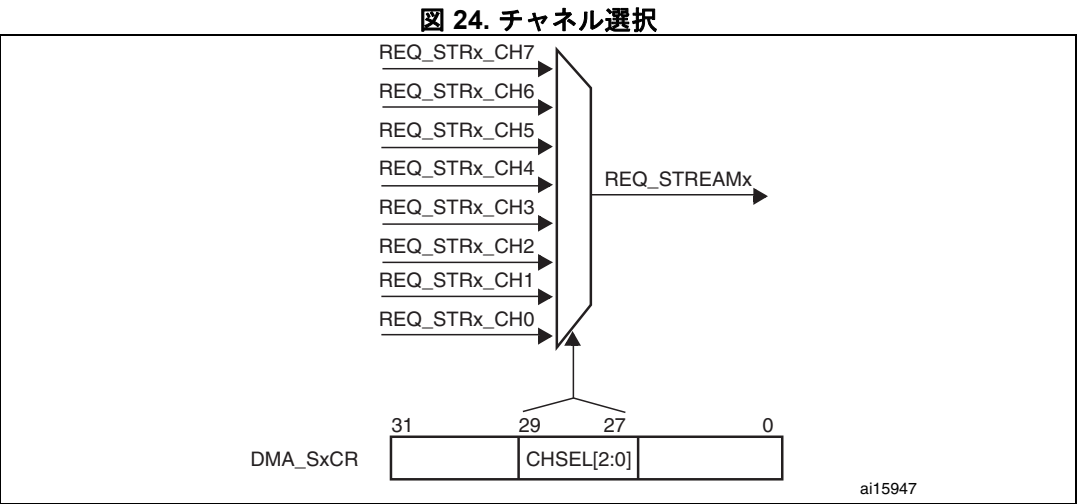
- DMA_SxPAR または DMA_SxM0AR レジスタを通じてアドレス指定されたメモリ上の場所またはペリフェラルデータレジスタからのデータのロード。
- DMA_SxPAR または DMA_SxM0AR レジスタを通じてアドレス指定されたメモリ内の場所またはペリフェラルデータレジスタにロードされたデータの格納。
- このあと予定される DMA トランザクションの数を保持している DMA_SxNDTR レジスタをカウントダウンします。

イベントの終了後、ペリフェラルは DMA コントローラにリクエスト信号を送信します。DMA コントローラは、チャネルの優先順位に応じて、リクエストを処理します。DMA コントローラがペリフェラルにアクセスするとすぐに、DMA コントローラはペリフェラルに確認応答信号を送信します。ペ

リフェラルは、DMA コントローラからの確認応答信号を受け取るとすぐにそのリクエストを解除します。ペリフェラルによってリクエストがネゲートされると、DMA コントローラは確認応答信号を解除します。さらにリクエストがある場合、ペリフェラルは次のトランザクションを開始できます。

9.3.3 チャンネル選択

各ストリームは、8 つのチャンネルリクエストの中から選択することができる DMA リクエストに関連付けられています。チャンネル選択は DMA_SxCR レジスタの CHSEL[2:0] ビットによって制御されます。



ペリフェラルからの 8 つのリクエスト (TIM、ADC、SPI、I²C など) は各チャンネルに個々に接続されており、それらの接続は製品の実装によって異なります。

表 81 および表 83 に DMA リクエストマッピングの例を示します。

表 28. DMA1 リクエストマッピング (STM32F401xB/C および STM32F401xD/E)

ペリフェラル リクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 0	SPI3_RX		SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX		SPI3_TX
チャンネル 1	I2C1_RX	I2C3_RX				I2C1_RX	I2C1_TX	I2C1_TX
チャンネル 2	TIM4_CH1		I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
チャンネル 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
チャンネル 4						USART2_RX	USART2_TX	
チャンネル 5			TIM3_CH4 TIM3_UP		TIM3_CH1 TIM3_TRIG	TIM3_CH2		TIM3_CH3
チャンネル 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2	I2C3_TX	TIM5_UP	
チャンネル 7			I2C2_RX	I2C2_RX				I2C2_TX

表 29. DMA2 リクエストマッピング (STM32F401xB/C および STM32F401xD/E)

ペリフェラル リクエスト	ストリーム 0	ストリーム 1	ストリーム 2	ストリーム 3	ストリーム 4	ストリーム 5	ストリーム 6	ストリーム 7
チャンネル 0	ADC1				ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
チャンネル 1								
チャンネル 2								
チャンネル 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
チャンネル 4	SPI4_RX	SPI4_TX	USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
チャンネル 5		USART6_RX	USART6_RX	SPI4_RX	SPI4_TX		USART6_TX	USART6_TX
チャンネル 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
チャンネル 7								

9.3.4 アービタ

アービタは、2 つの AHB マスタポート（メモリポートおよびペリフェラルポート）のそれぞれについて、その優先順位に基づいて 8 つの DMA ストリームリクエストを管理し、ペリフェラル/メモリアクセスシーケンスを起動します。

優先順位は 2 段階で管理されます。

- ソフトウェア：各ストリームの優先順位は、DMA_SxCR レジスタで設定できます。4 つのレベルがあります。
 - 最優先
 - 高優先
 - 中優先
 - 低優先
- ハードウェア：2 つのリクエストのソフトウェア優先順位レベルが同じ場合、小さな番号のストリームが大きな番号のストリームよりも優先されます。たとえば、ストリーム 2 はストリーム 4 よりも優先されます。

9.3.5 DMA ストリーム

8 つの DMA コントローラストリームのそれぞれに、転送元と転送先の間の単方向転送リンクを備えています。

各ストリームは次のことを行うよう設定することができます。

- レギュラータイプトランザクション：メモリからペリフェラル、ペリフェラルからメモリまたはメモリ間の転送
- ダブルバッファタイプトランザクション：メモリ用に 2 つのメモリポインタを使用したダブルバッファ転送 (DMA がバッファの読み書きを行っている間、アプリケーションは他のバッファの読み書きが可能)

転送するデータの量 (65535 以内) はプログラム可能で、ペリフェラル AHB ポートに接続された DMA 転送を要求するペリフェラルのデータ幅に関連しています。転送されるデータ数を格納しているレジスタは、トランザクションの終了ごとにデクリメントされます。

9.3.6 転送元、転送先、および転送モード

転送元および転送先は両方とも、アドレス 0x0000 0000 と 0xFFFF FFFF の間の全 4 GB メモリ領域内のペリフェラルおよびメモリを転送アドレスとして使用できます。

方向は DMA_SxCR レジスタの DIR[1:0] ビットを使用して設定し、メモリからペリフェラル、ペリフェラルからメモリ、およびメモリ間の転送の 3 つの選択肢があります。表 30 に対応する転送元と転送先アドレスを示します。

表 30. 転送元および転送先アドレス

DMA_SxCR レジスタの DIR [1:0] ビット	方向	転送元アドレス	転送先アドレス
00	ペリフェラルからメモリ	DMA_SxPAR	DMA_SxM0AR
01	メモリからペリフェラル	DMA_SxM0AR	DMA_SxPAR
10	メモリ間転送	DMA_SxPAR	DMA_SxM0AR
11	予約済み	-	-

DMA_SxCR レジスタの PSIZE または MSIZE ビットでプログラムされたデータ幅が、それぞれハーフワードまたはワードである場合、DMA_SxPAR または DMA_SxM0AR/M1AR レジスタに書き込まれたペリフェラルアドレスまたはメモリアドレスは、それぞれワードアドレス境界またはハーフワードアドレス境界で整列する必要があります。

“ペリフェラルからメモリ”モード

図 25 でこのモードを説明しています。

このモードが有効になっている場合 (DMA_SxCR レジスタの EN ビットをセットすることにより)、ペリフェラルリクエストが発生するたびに、ストリームは転送元からの転送を開始し、FIFO を満たします。

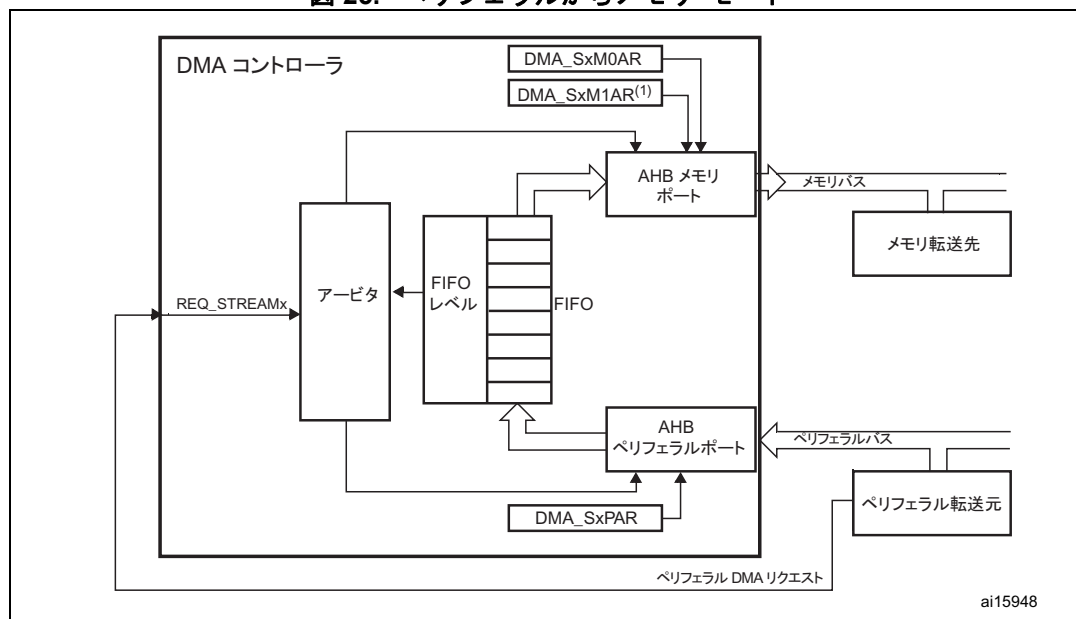
FIFO の閾値レベルに達すると、FIFO の内容が排出され、転送先に格納されます。

DMA_SxNDTR レジスタがゼロに達したか、ペリフェラルが転送の終了を要求したか (ペリフェラルフローコントローラの場合)、またはソフトウェアによって DMA_SxCR レジスタの EN ビットがクリアされたとき、転送は停止します。

ダイレクトモード (DMA_SxFCR レジスタの DMDIS 値が“0”のとき) では、FIFO の閾値レベルは使用されません。ペリフェラルから FIFO への各データの転送後、対応するデータはすぐに排出されて転送先に格納されます。

ストリームは、対応するストリームのアービトレーションでバスの使用権を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

図 25. “ペリフェラルからメモリ”モード



1. ダブルバッファモード用。

“メモリからペリフェラル”モード

図 26 でこのモードを説明しています。

このモードが有効になっている場合 (DMA_SxCR レジスタの EN ビットをセットしている場合)、ストリームはすぐに転送元から転送を開始し、FIFO を完全に満たします。

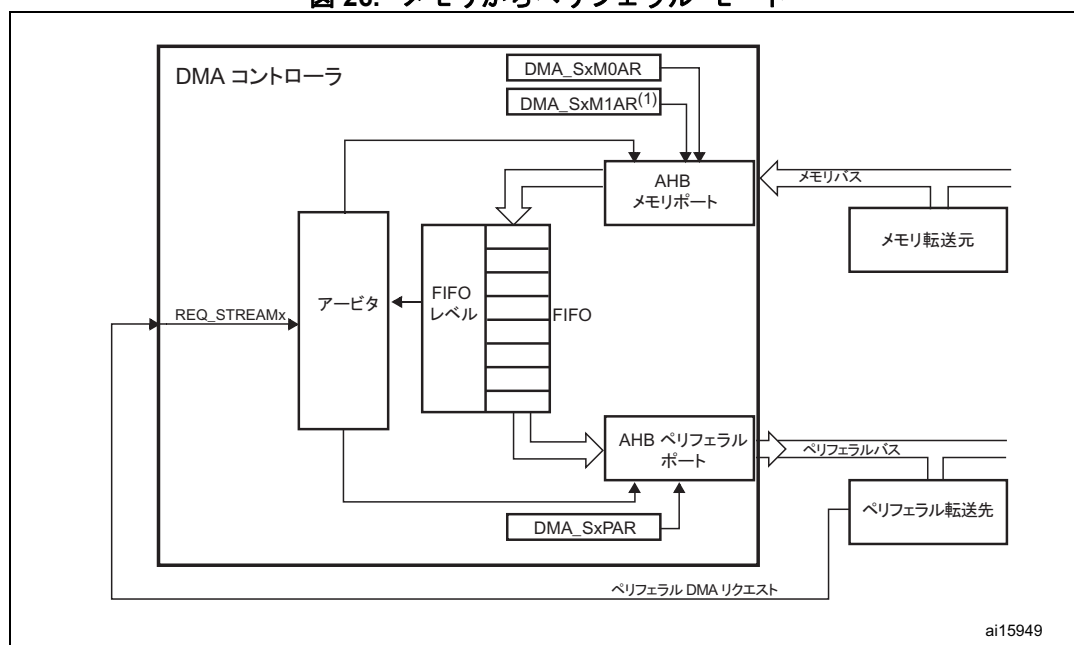
ペリフェラルリクエストが発生するたびに、FIFO のデータが転送されて転送先に格納されます。FIFO のレベルがあらかじめ設定された閾値レベルより低いとき、FIFO はメモリからのデータで完全に再ロードされます。

DMA_SxNDTR レジスタがゼロに達したか、ペリフェラルが転送の終了を要求したか (ペリフェラルフローコントローラの場合)、またはソフトウェアによって DMA_SxCR レジスタの EN ビットがクリアされたとき、転送は停止します。

ダイレクトモード (DMA_SxFCR レジスタの DMDIS 値が“0”のとき) では、FIFO の閾値レベルは使用されません。ストリームが有効になると、DMA は内部 FIFO に転送するための最初のデータをプリロードします。ペリフェラルがデータ転送を要求すると、DMA はプリロードされたデータを設定された転送先へすぐに転送します。そしてまた転送すべき次のデータを空の内部 FIFO に再ロードします。プリロードデータサイズは、DMA_SxCR レジスタの PSIZE ビットフィールドの値に対応します。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

図 26. “メモリからペリフェラル”モード



1. ダブルバッファモード用。

メモリ間モード

DMA チャンネルは、ペリフェラルからのリクエストによってトリガされなくても機能します。これは [図 27](#) で説明しているメモリ間モードです。

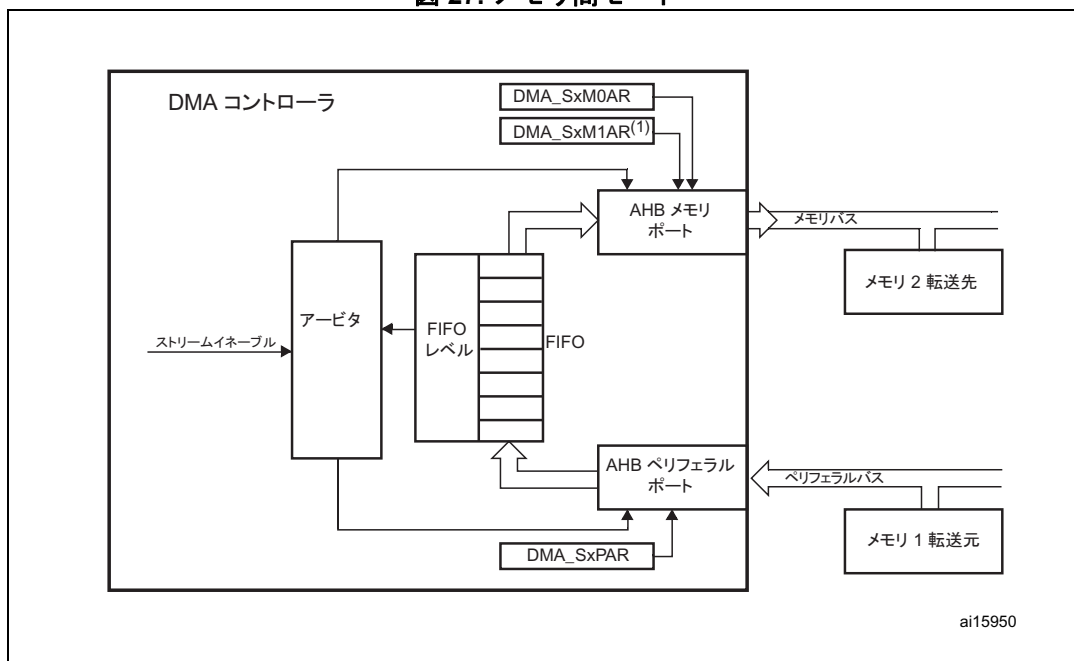
DMA_SxCR レジスタのイネーブルビット (EN) をセットすることによってストリームが有効になっている場合、ストリームはすぐに閾値レベルまで FIFO への書き込みを開始します。FIFO の閾値レベルに達すると、FIFO のデータが転送され、転送先に格納されます。

DMA_SxNDTR レジスタがゼロに達するかソフトウェアによって DMA_SxCR レジスタの EN ビットがクリアされたとき転送を停止します。

ストリームは、対応するストリームのアービトレーションでバスの使用权を得た場合にのみ AHB 転送元または転送先のポートにアクセスできます。このアービトレーションは DMA_SxCR レジスタの PL[1:0] ビットを使用して各ストリームに対して定義された優先度に基づいて行われます。

注： **メモリ間モードを使用する場合、サーキュラモードおよびダイレクトモードは使用できません。**
DMA2 コントローラのみメモリ間転送を行うことができます。

図 27. メモリ間モード



1. ダブルバッファモード用。

9.3.7 ポインタのインクリメント

ペリフェラルやメモリのポインタは、DMA_SxCR レジスタの PINC および MINC ビットに応じて、各転送後に自動的にポストインクリメントまたは一定に保つことができます。

インクリメントモードの無効化は、転送元または転送先ペリフェラルデータを 1 つのレジスタを介してアクセスする場合に便利です。

インクリメントモードが有効な場合、次の転送アドレスは、DMA_SxCR レジスタの PSIZE または MSIZE ビットにプログラムされたデータ幅に応じて、前回のアドレスに 1 (バイトの場合)、2 (ハーフワードの場合)、または 4 (ワードの場合) をインクリメントしたアドレスになります。

パッキング動作を最適化するため、AHB ペリフェラルポート上で転送されるデータのサイズに関わらずペリフェラルアドレスのインクリメントオフセットサイズを固定することができます。DMA_SxCR レジスタの PINCOS ビットは、インクリメントオフセットサイズをペリフェラル AHB ポート上のデータサイズまたは 32 ビットアドレス (この場合アドレスは 4 つインクリメントされます) に調整するために使用されます。PINCOS ビットは AHB ペリフェラルポートにのみ影響します。

PINCOS ビットがセットされている場合、次の転送のアドレスは、PSIZE 値にかかわらず前のアドレスに 4 をインクリメント (自動的に 32 ビットアドレスに調整) したアドレスです。しかしながら、この動作による AHB メモリポートへの影響はありません。

9.3.8 サークュラモード

サーキュラモードを使用すると、サーキュラバッファや連続したデータフロー（ADC スキャンモードなど）を処理できます。この機能は、DMA_SxCR レジスタの CIRC ビットを使用して有効にできます。

サーキュラモードが有効になると、転送されるデータ項目数にはストリーム設定段階でプログラミングされた初期値が自動的に再ロードされ、DMA リクエストの処理が続行されます。

注： サークュラモードでは、メモリ用にバーストモードが設定されている場合、次のルールを守る必要があります。

$DMA_SxNDTR = ((Mburst \text{ ビート}) \times (Msize) / (Psize))$ 、ここで

- $(Mburst \text{ ビート}) = 4, 8 \text{ または } 16$ (DMA_SxCR レジスタの MBURST ビットによって異なる)
- $((Msize) / (Psize)) = 1, 2, 4, 1/2 \text{ または } 1/4$ (Msize と Psize は DMA_SxCR レジスタの MSIZE と PSIZE ビットを表す。これらはバイト依存)
- $DMA_SxNDTR = AHB \text{ ペリフェラルポート上で転送するデータ項目の数}$

例： Mburst ビート=8 (INCR8)、MSIZE=00 (バイト) および PSIZE=01 (ハーフワード)、この場合、DMA_SxNDTR は $(8 \times 1/2 = 4)$ の倍数でなければなりません。

この式を守らないと DMA 動作とデータの整合性は保証されません。

NDTR もペリフェラルバーストサイズの倍数にペリフェラルデータサイズを乗じた値でなければならず、さもないと不正な DMA 動作を引き起こす可能性があります。

9.3.9 ダブルバッファモード

このモードは、すべての DMA1 と DMA2 ストリームに対して使用可能です。

ダブルバッファモードは、DMA_SxCR レジスタの DBM ビットをセットすることで有効になります。

ダブルバッファストリームは通常の（シングルバッファ）ストリームとして動作しますが、2 つのメモリポインタを持っているところが異なります。ダブルバッファモードが有効になっていると、サーキュラモードが自動的に有効になり（DMA_SxCR の CIRC ビットは無視）、トランザクションが終わるごとにメモリポインタをスワップします。

このモードでは、DMA コントローラはトランザクションが終わるごとに、あるメモリターゲットから他へスワップします。これにより、DMA 転送によってある 1 つのメモリ領域がフルまたは使用中である間に、ソフトウェアは別のメモリ領域を処理することができます。ダブルバッファストリームは、[表 31：ダブルバッファモード \(DBM=1\) における転送元および転送先アドレスレジスタ](#)で説明しているように両方向に動作することができます（メモリは転送元または転送先のいずれにも設定可能）。

注： ダブルバッファモードでは、ストリームが有効なとき、以下の条件を守ることにより、いつでも AHB メモリポートのベースアドレス (DMA_SxM0AR または DMA_SxM1AR) を更新することができます。

- DMA_SxCR レジスタで CT ビットが“0”のとき DMA_SxM1AR レジスタに書き込むことができます。CT=1 のときに、このレジスタに書き込みを試みると、エラーフラグ (TEIF) がセットされ、ストリームが自動的に無効になります。
- DMA_SxCR レジスタで CT ビットが“1”のとき DMA_SxM0AR レジスタに書き込むことができます。CT=0 のときに、このレジスタに書き込みを試みると、エラーフラグ (TEIF) がセットされ、ストリームが自動的に無効になります。

エラー条件を回避するためには、TCIF フラグがアサートされたらできるだけ早くベースアドレスを変更することを推奨します。なぜなら、この時点で、上記 2 つの内いずれかの条件にしたがって DMA_SxCR レジスタの CT の値に応じてターゲットメモリがメモリ 0 から 1（または 1 から 0）に変更されているためです。

他のすべてのモード（ダブルバッファモードを除く）に対し、ストリームが有効になるとすぐに、メモリアドレスレジスタは書き込み保護されます。

表 31. ダブルバッファモード（DBM=1）における転送元および転送先アドレスレジスタ

DMA_SxCR レジスタの DIR [1:0] ビット	方向	転送元アドレス	転送先アドレス
00	ペリフェラルからメモリ	DMA_SxPAR	DMA_SxM0AR/DMA_SxM1AR
01	メモリからペリフェラル	DMA_SxM0AR/DMA_SxM1AR	DMA_SxPAR
10	設定禁止 ⁽¹⁾		
11	予約済み	-	-

1. ダブルバッファモードが有効になっている場合、サーキュラモードが自動的に有効になります。メモリ間モードはサーキュラモードと互換性がないため、ダブルバッファモードが有効になっているときは、メモリ間モードを設定することはできません。

9.3.10 プログラム可能なデータ幅、パッキング／アンパッキング、エンディアン形式

転送されるデータ項目の数は、ストリームを有効にする前に DMA_SxNDTR（転送データ項目数ビット、NDT）にプログラムする必要があります（フローコントローラがペリフェラルで、DMA_SxCR の PFCTRL ビットがセットされている場合を除く）。

内部 FIFO を使用する場合、転送元と転送先データのデータ幅は DMA_SxCR レジスタの PSIZE と MSIZE ビットを介してプログラム可能です（8、16、または 32 ビットが可能）。

PSIZE と MSIZE が等しくない場合：

- DMA_SxNDTR レジスタに設定されている転送データ項目数のデータ幅は、ペリフェラルバスの幅（DMA_SxCR レジスタの PSIZE ビットで設定されている）と同じです。たとえば、ペリフェラルからメモリ、メモリからペリフェラル、またはメモリ間転送のケースで PSIZE[1:0] ビットがハーフワード用に設定されている場合、転送されるバイト数は 2×NDT です。
- DMA コントローラは、転送元と転送先の両方でリトルエンディアンアドレッシングのみ取り扱うことができます。これについては、[表 32：パッキング／アンパッキングおよびエンディアンの動作（ビット PINC=MINC=1）](#)で説明されています。

このパッキング／アンパッキング手順には、データが完全にパック／アンパックされる前に操作が中断された場合にデータが破損する危険性があります。したがって、データの一貫性を確保するため、ストリームはバースト転送を生成するよう設定することができます。この場合、バーストに属する各転送グループは分割できません（[セクション 9.3.11：シングル転送とバースト転送](#)を参照）。

ダイレクトモード（DMA_SxFCR レジスタで DMDIS=0 のとき）では、データのパッキング／アンパッキングはできません。この場合には、転送元と転送先の転送データ幅が異なってはなりません。両方が等しく、DMA_SxCR の PSIZE ビットで定義します（MSIZE ビットは無視されます）。

表 32. パッキング／アンパッキングおよびエンディアンの動作（ビット PINC=MINC=1）

AHB メモリ ポート幅	AHB ペリフェラル ポート幅	転送する データ 項目の数 (NDT)	メモリ 転送番号	メモリポート アドレス/ バイトレーン	ペリ フェラル 転送番号	ペリフェラルポートアドレス/バイトレーン	
						PINCOS = 1	PINCOS = 0
8	8	4	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
8	16	2	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
8	32	1	1 2 3 4	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]	1	0x0 / B3[B2:B1[B0:31:0]	0x0 / B3[B2:B1[B0:31:0]
16	8	4	1 2	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
16	16	2	1 2	0x0 / B1[B0:15:0] 0x2 / B1[B0:15:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
16	32	1	1 2	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]	1	0x0 / B3[B2:B1[B0:31:0]	0x0 / B3[B2:B1[B0:31:0]
32	8	4	1	0x0 / B3[B2:B1[B0:31:0]	1 2 3 4	0x0 / B0[7:0] 0x4 / B1[7:0] 0x8 / B2[7:0] 0xC / B3[7:0]	0x0 / B0[7:0] 0x1 / B1[7:0] 0x2 / B2[7:0] 0x3 / B3[7:0]
32	16	2	1	0x0 / B3[B2:B1[B0:31:0]	1 2	0x0 / B1[B0:15:0] 0x4 / B3[B2:15:0]	0x0 / B1[B0:15:0] 0x2 / B3[B2:15:0]
32	32	1	1	0x0 / B3[B2:B1[B0:31:0]	1	0x0 / B3[B2:B1[B0:31:0]	0x0 / B3[B2:B1[B0:31:0]

注： ペリフェラルポートは、転送元または転送先として設定可能です（メモリ間転送の場合はメモリ転送元としての設定も可能です）。

PSIZE、MSIZE、および NDT[15:0] は、最後の転送が不完全にならないよう必ず設定する必要があります。これはペリフェラルポートのデータ幅（PSIZE ビット）がメモリポートのデータ幅（MSIZE ビット）より小さい場合に発生する可能性があります。この制約の一覧を表 33 に示します。

表 33. PSIZE と MSIZE に対する NDT に関する制限事項

PSIZE[1:0] of DMA_SxCR	MSIZE[1:0] of DMA_SxCR	NDT[15:0] of DMA_SxNDTR
00（8 ビット）	01（16 ビット）	2 の倍数でなければならない
00（8 ビット）	10（32 ビット）	4 の倍数でなければならない
01（16 ビット）	10（32 ビット）	2 の倍数でなければならない

9.3.11 シングル転送とバースト転送

DMA コントローラは、シングル転送を生成することもできれば、4、8、または 16 ビートのインクリメンタルバースト転送を生成することもできます。

バーストのサイズは、DMA_SxCR レジスタの MBURST[1:0] と PBURST[1:0] ビットを使用して 2 つの AHB ポートに対して別々にソフトウェアで設定されます。

バーストサイズは、バースト内のビート数を示します。転送するバイト数ではありません。

データの一貫性を確保するため、バーストを形成する転送の各グループは分割できません。AHB 転送はロックされ、AHB バスマトリックスのアービタは、バースト転送シーケンス中 DMA マスタからバスの使用を奪取しません。

シングルまたはバーストの設定によって各 DMA リクエストによって AHB ペリフェラルポート上で開始される転送の数は異なります。

- AHB ペリフェラルポートがシングル転送用に設定されている場合、各 DMA リクエストによって、DMA_SxCR レジスタの PSIZE[1:0] ビットに応じた、バイト、ハーフワード、またはワードのデータ転送が生成されます。
- AHB ペリフェラルポートがバースト転送用に設定されている場合、各 DMA リクエストによって、DMA_SxCR レジスタの PBURST[1:0] ビットと PSIZE[1:0] ビットに応じた、4、8、または 16 ビートのバイト、ハーフワード、またはワードの転送が生成されます。

MBURST と MSIZE ビットに関しても、AHB メモリポートに対して上記と同じことを考慮する必要があります。

ダイレクトモードでは、ストリームはシングル転送のみ生成することができ、MBURST[1:0] と PBURST[1:0] ビットはハードウェアで決まってしまう。

アドレスポインタ (DMA_SxPAR または DMA_SxM0AR レジスタ) は、バーストブロック内のすべての転送が転送サイズに等しいアドレス境界で整列するように選択する必要があります。

バースト設定は、AHB プロトコルを守るよう選択する必要があります。ここで、1 個のスレーブに割り当てることができる最小アドレス空間が 1 KB であるため、バーストが 1 KB アドレス境界を越えることはできません。つまり、バーストブロック転送が 1 KB アドレス境界を超えてはならず、超えた場合には AHB エラーが生成される可能性があります。このエラーは DMA レジスタで報告されません。

9.3.12 FIFO

FIFO の構造

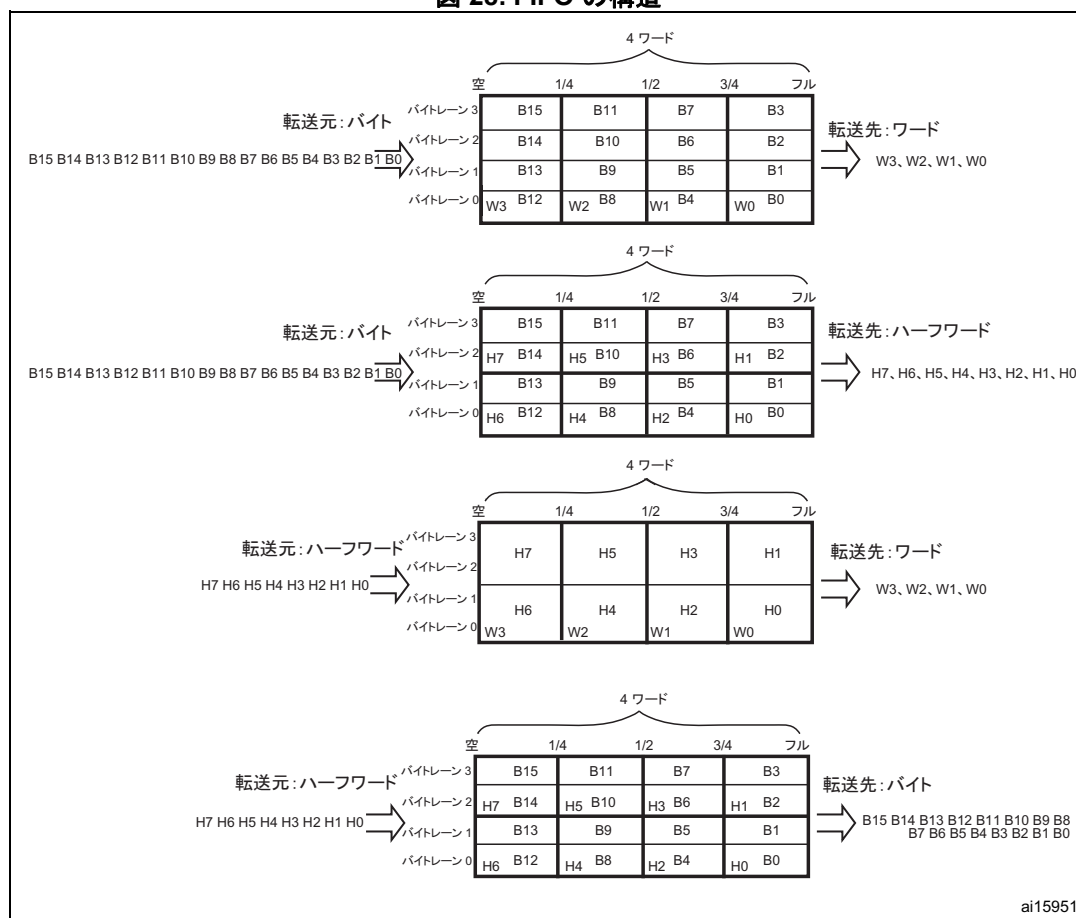
FIFO は転送元からのデータを転送先へ送信する前に一時的に格納するために使用します。

各ストリームには独立した 4 ワードの FIFOがあり、閾値レベルはソフトウェアで 1/4、1/2、3/4、またはフルに設定できます。

FIFO 閾値レベルを使用できるようにするには、DMA_SxFCR レジスタの DMDIS ビットをセットしてダイレクトモードを無効にする必要があります。

FIFO の構造は、転送元および転送先のデータ幅によって異なり、図 28 : FIFO の構造で説明しています。

図 28. FIFO の構造



FIFO 閾値とバーストの設定

FIFO の閾値 (DMA_SxFCR レジスタの FTH[1:0] ビット) とメモリバーストのサイズ (DMA_SxCR レジスタの MBURST[1:0]) を選択する場合は注意が必要です。FIFO 閾値が指す内容はメモリバースト転送の整数倍に一致する必要があります。一致しない場合、ストリームが有効なとき FIFO エラー (DMA_HISR または DMA_LISR レジスタの FEIFx フラグ) が発生し、ストリームが自動的に無効になります。可能な設定と禁止されている設定の説明は表 34 : FIFO 閾値設定を参照してください。

表 34. FIFO 閾値設定

MSIZE	FIFO レベル	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16	
バイト	1/4	4 ビートの 1 バースト	設定禁止	設定禁止	
	1/2	4 ビートの 2 バースト	8 ビートの 1 バースト		
	3/4	4 ビートの 3 バースト	設定禁止		
	フル	4 ビートの 4 バースト	8 ビートの 2 バースト	16 ビートの 1 バースト	
ハーフワード	1/4	設定禁止	設定禁止	設定禁止	
	1/2	4 ビートの 1 バースト			
	3/4	設定禁止			
	フル	4 ビートの 2 バースト	8 ビートの 1 バースト		
ワード	1/4	設定禁止	設定禁止		
	1/2				
	3/4				
	フル	4 ビートの 1 バースト			

いかなる場合でも、バーストサイズにデータサイズを乗じた値が FIFO サイズを超えることはできません (可能なデータサイズは1 (バイト)、2 (ハーフワード) または 4 (ワード))。

次のいずれかの条件が発生した場合、DMA 転送の終了時に、バースト転送が不完全となる可能性があります。

- AHB ペリフェラルポート設定において、データ項目の合計数 (DMA_SxNDTR レジスタに設定) が、バーストサイズにデータサイズを乗じた値の倍数ではない
- AHB メモリポート設定において、メモリに転送する FIFO 内の残りのデータ項目数が、バーストサイズにデータサイズを乗じた値の倍数ではない

このような場合、DMA ストリーム設定時にバーストランザクションが要求されていても、転送すべき残りのデータは、DMA によってシングルモードで管理されます。

注：ペリフェラル AHB ポートでバースト転送が要求されており、FIFO が使用されている場合 (DMA_SxCR レジスタの DMDIS=1)、DMA ストリーム方向に応じて、恒久的なアンダーランまたはオーバーラン状態を回避するため、次のルールを守ることが必要です。

(PBURST × PSIZE) = FIFO_SIZE (4 ワード) の場合、PSIZE = 1、2、または 4 であり、かつ、PBURST = 4、8、または 16 の時、FIFO_Threshold = 3/4 は 禁止。

このルールによって、ペリフェラルからの要求を一度に処理するのに十分な FIFO スペースを確実に確保できます。

FIFO フラッシュ

FIFO は DMA_SxCR レジスタの EN ビットをリセットすることによりストリームが無効化された場合、およびストリームが“ペリフェラルからメモリ”転送またはメモリ間転送を管理するよう設定された場合、一掃することができます。ストリームを無効化したときに FIFO にデータが残っている場合、DMA コントローラは引き続き残りのデータを転送先へ転送します（ストリームが事実上無効化されているとしても）。このフラッシュが完了すると、DMA_LISR または DMA_HISR レジスタの転送完了ステータスビット (TCIFx) がセットされます。

この場合、残りのデータカウンタ DMA_SxNDTR は値を保持し、現在いくつかのデータ項目が転送先メモリに格納できるかを示します。

FIFO フラッシュ動作中に、FIFO 内のメモリに転送する残りのデータ項目数（バイト単位）がメモリデータ幅より少ない場合（たとえば、MSIZE がワードに設定されているとき FIFO 内に 2 バイト）、データは DMA_SxCR レジスタの MSIZE ビットに設定されたデータ幅で送信されます。つまり、メモリには不要な値が書き込まれることになります。ソフトウェアで DMA_SxNDTR レジスタを読み出すことにより、正しいデータが格納されたメモリ領域（開始アドレスと終了アドレス）を判断することができます。

FIFO 内の残りのデータ項目数がバーストサイズよりも小さい場合（AHB メモリポート上のバーストを管理するストリームを設定するため DMA_SxCR レジスタの MBURST ビットがセットされている場合）、FIFO フラッシュを完了するためシングルトランザクションが生成されます。

ダイレクトモード

デフォルトでは、FIFO はダイレクトモードで動作し（DMA_SxFCR の DMDIS ビットがリセットされています）、FIFO 閾値レベルは使用しません。このモードは、システムが各 DMA リクエスト後にメモリとの即時シングル転送を必要とする場合に便利です。

DMA がダイレクトモード（FIFOが無効）に設定されている場合、“メモリからペリフェラル”モードでデータを転送するため、ペリフェラルからの DMA リクエストが発生した際、即座にかつ確実にデータ転送が行われるように、DMA はデータをメモリから内部 FIFO へ 1 つプリロードします。

FIFO の飽和を避けるためには、対応するストリームを高い優先度で設定することを推奨します。

このモードは、次のような転送に制限されています。

- 転送元および転送先の転送幅が等しく、両方とも DMA_SxCR の PSIZE[1:0] ビットによって定義されている（MSIZE[1:0] ビットは無視される）
- バースト転送は不可（DMA_SxCR の PBURST[1:0] と MBURST[1:0] ビットは無視される）

メモリ間転送を実行する際はダイレクトモードを使用することはできません。

9.3.13 DMA 転送の完了

DMA_LISR または DMA_HISR ステータスレジスタの TCIFx ビットを設定することにより、さまざまなイベントによって転送を終了させることができます。

- DMA フローコントローラモード :
 - “メモリからペリフェラル”モードで DMA_SxNDTR カウンタがゼロに達した
 - (DMA_SxCR レジスタの EN ビットをクリアすることにより) ストリームが転送終了する前に無効化され、(転送がペリフェラルからメモリまたはメモリ間の場合) 残りのすべてのデータが FIFO からメモリにフラッシュされた
- ペリフェラルフローコントローラモード :
 - 最後の外部バーストまたはシングルリクエストがペリフェラルから生成され、(DMA が“ペリフェラルからメモリ”モードで動作しているとき) 残りのデータが FIFO からメモリへ転送された
 - ストリームがソフトウェアによって無効化され、(DMA が“ペリフェラルからメモリ”モードで動作しているとき)、残りのデータが FIFO からメモリへ転送された

注 : 転送完了は、“ペリフェラルからメモリ”モードの場合のみ、メモリに転送される FIFO 内の残りのデータによって異なります。この条件は、“メモリからペリフェラル”モードには適用されません。

ストリームがノンサーキュラモードで設定されている場合、転送終了後 (転送すべきデータの数がゼロに達したとき)、DMA は停止し (DMA_SxCR レジスタの EN ビットがハードウェアでクリアされます)、ソフトウェアによってストリームを再プログラムして再有効化 (DMA_SxCR レジスタの EN ビットを設定) しない限り、DMA リクエストは処理されません。

9.3.14 DMA 転送の中断

DMA 転送は、いつでも中断して再開することもできれば、DMA 転送が終了する前に完全に無効化することもできます。

これには、2 つのケースがあります。

- ストリームは、転送を無効にし、停止した場所以降再開しません。ストリームを無効化するために DMA_SxCR レジスタの EN ビットをクリアする以外、特に行うことはありません。ストリームを無効にするのに時間を要する可能性があります (まず、進行中の転送を完了します)。転送の終了を示すために転送完了割込みフラグ (DMA_LISR または DMA_HISR レジスタの TCIF) がセットされます。その時点で、DMA_SxCR の EN ビットの値が“0”となって、ストリームが中断されます。ソフトウェアによって、ストリームが中断される前に転送されたデータ項目の数を判断できるように、DMA_SxNDTR レジスタに、ストリームが停止した時点で残っているデータ項目の数を保持しています。
- ストリームは、DMA_SxNDTR レジスタ内の転送されるべき残りのデータ項目数が 0 に到達する前に転送を中断します。その目的は、ストリームを再度有効にすることで後から転送を再開することにあります。転送が停止した時点から再開するにあたり、ソフトウェアが既に収集したデータ項目数を知るため DMA_SxCR レジスタの EN ビットに書き込んで (その後それが“0”であることを確認します) ストリームを無効にした後で DMA_SxNDTR レジスタを読み出す必要があります。その後、
 - アドレスポインタを調整するために、ペリフェラルやメモリのアドレスを更新する必要があります。
 - SxNDTR レジスタは、転送すべきデータ項目の残数 (ストリームを無効にしたとき読み出される値) で更新する必要があります。
 - ストリームは、その後再度有効化して転送が停止したところから再開することができます。

注 : ストリームの中断による、転送の終了を示すため、転送完了割込みフラグ (DMA_LISR または DMA_HISR の TCIF) がセットされることに注意してください。

9.3.15 フローコントローラ

転送するデータの数を制御するものはフローコントローラとして知られています。このフローコントローラは、DMA_SxCR レジスタの PFCTRL ビットを使用して、各ストリームごとに個別に設定されます。

フローコントローラは次のように使用することができます。

- DMA コントローラ：この場合、DMA ストリームを有効にする前に、転送するデータ項目数を DMA_SxNDTR レジスタにソフトウェアでプログラムします。
- ペリフェラル転送元またはペリフェラル転送先：この場合、転送されるデータ項目数は不明です。最後のデータが転送されるとき、ペリフェラルがハードウェアによって DMA コントローラに知らせます。この機能は、転送の終了を通知できるペリフェラルでのみサポートされています。すなわち、
 - SDIO

特定のストリームでペリフェラルフローコントローラを使用した場合、DMA_SxNDTR に書き込まれる値は DMA 転送に影響しません。実際に、いかなる値が書き込まれても、以下のスキームを守るためストリームが有効になるとすぐに、ハードウェアによって強制的に 0xFFFF になります。

- 想定されるストリームの中断：最終データを示すハードウェア信号（シングルまたはバースト）がペリフェラルによって送信される前に、DMA_SxCR レジスタの EN ビットをソフトウェアによって 0 にリセットし、ストリームを停止します。このようなケースでは、“ペリフェラルからメモリ” DMA 転送の場合、ストリームが停止し、FIFO フラッシュが作動します。対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、DMA の完了を示します。DMA 転送中に転送されたデータ項目数を知るには、DMA_SxNDTR レジスタを読み出し次の式を適用します。
 - 転送されたデータ数 = 0xFFFF – DMA_SxNDTR
- 最終データを示すハードウェア信号の受信による通常ストリームの中断：ペリフェラルが最後の転送（シングルまたはバースト）を要求しこの転送が完了したときストリームは自動的に中断されます。対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、DMA 転送の完了を示します。転送されたデータ項目数を知るには、DMA_SxNDTR レジスタを読み出し上記と同じ式を適用します。
- DMA_SxNDTR レジスタが 0 に到達：対応するストリームの TCIFx フラグがステータスレジスタにてセットされ、強制的な DMA 転送の完了を示します。最終データを示すハードウェア信号（シングルまたはバースト）がまだアサートされていない場合でも、ストリームは自動的に停止します。既に転送されたデータは失われません。つまり、ペリフェラルフローコントローラモードの場合でも、1 つの DMA のトランザクションにより最大 65535 のデータ項目が管理できることになります。

注： メモリ間モードに設定されているとき、DMA は常にフローコントローラとなり PFCTRL ビットはハードウェアによって強制的に 0 になります。

ペリフェラルフロー制御モードでは、サーキュラモードは禁止されています。

9.3.16 実現可能な DMA 設定の概要

表 35 に、実現可能なさまざまな DMA 設定をまとめています。

表 35. 実現可能な DMA 設定

DMA 転送モード	転送元	転送先	フロー コントローラ	サーキュラ モード	転送タイプ	ダイレクト モード	ダブルバッファ モード
ペリフェラル からメモリ	AHB ペリフェラル ポート	AHB メモリポート	DMA	設定可能	シングル	設定可能	設定可能
					バースト	設定禁止	
			ペリフェラル	設定禁止	シングル	設定可能	設定禁止
					バースト	設定禁止	
メモリから ペリフェラル	AHB メモリポート	AHB ペリフェラル ポート	DMA	設定可能	シングル	設定可能	設定可能
					バースト	設定禁止	
			ペリフェラル	設定禁止	シングル	設定可能	設定禁止
					バースト	設定禁止	
メモリ間転送	AHB ペリフェラル ポート	AHB メモリポート	DMA のみ	設定禁止	シングル	設定禁止	設定禁止
					バースト		

9.3.17 ストリーム設定手順

DMA ストリーム x の設定は、次の手順で行います (x はストリーム番号です)。

1. ストリームが有効になっている場合、DMA_SxCR レジスタの EN ビットをリセットすることによって無効にし、その後、進行中のストリーム動作がないことを確認するため、このビットを読み出します。このビットに 0 を書き込んでもすぐに効果はありません。それは現在行われているすべての転送が完了した後、実際に 0 が書き込まれるからです。EN ビットが 0 として読み出される場合、ストリームが設定可能な状態であることを意味します。したがって、ストリームの設定を開始する前に EN ビットがクリアされるのを待つ必要があります。以前のデータブロックの DMA 転送からステータスレジスタ (DMA_LISR と DMA_HISR) に設定されたすべてのストリーム専用ビットは、ストリームを再度有効にする前にクリアする必要があります。
2. DMA_SxPAR レジスタにペリフェラルポートレジスタアドレスを設定します。ペリフェラルイベントの後、データはこのアドレスとペリフェラルポートの間で転送されます。
3. DMA_SxMA0R レジスタ (およびダブルバッファモードの場合は DMA_SxMA1R レジスタ) にメモリアドレスを設定します。ペリフェラルイベント後は、このメモリに対してデータを読み書きします。
4. 転送されるデータ項目の合計数を DMA_SxNDTR レジスタで設定します。各ペリフェラルイベントまたはバーストの各ビットの後、この値はデクリメントされます。
5. DMA_SxCR レジスタの CHSEL[2:0] を使用して DMA チャンネル (リクエスト) を選択します。
6. ペリフェラルをフローコントローラとして使用することを意図し、かつこの機能をサポートしている場合、DMA_SxCR レジスタの PFCTRL ビットをセットします。
7. DMA_SxCR レジスタの PL[1:0] ビットを使用して、ストリームの優先順位を設定します。
8. FIFO の使用方法を設定します (有効または無効にする、送信と受信における閾値)。
9. DMA_SxCR レジスタで以下の項目を設定します。データ転送方向、ペリフェラルとメモリのインクリメントモード/固定モード、シングルトランザクションまたはバーストランザクション、ペリフェラルとメモリのデータ幅、サーキュラモード、ダブルバッファモード、ハーフ転送やフル転送後の割込み、およびエラー。

10. DMA_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

ストリームが有効になると、そのストリームに接続されているペリフェラルからの DMA リクエストを処理できます。

AHB 転送先ポート上でデータの半分が転送されると、1/2 転送フラグ (HTIF) がセットされ、1/2 転送割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、割込みが生成されます。

警告： DMA ストリームリクエストに接続されているペリフェラルを切断するには、まず、ペリフェラルが接続されている DMA ストリームを切断し、次いで EN ビット=0 になるのを待つ必要があります。その後はじめてペリフェラルを安全に無効にすることができます。

9.3.18 エラー管理

DMA コントローラによって、次のエラーを検出することができます。

- **転送エラー：**転送エラー割込みフラグ (TEIFx) は次の場合にセットされます。
 - DMA 読みまたは書き込みアクセス時にバスエラーが発生した場合
 - ダブルバッファモードでは、メモリアドレスレジスタの書き込みアクセスはソフトウェアによって要求されますが、ストリームが有効になり、メモリアドレスレジスタへの書き込みによって影響を受けるのは現在のターゲットメモリです ([セクション 9.3.9: ダブルバッファモード](#)を参照)。
- **FIFO エラー：**FIFO エラー割込みフラグ (FEIFx) は次の場合にセットされます。
 - FIFO アンダーラン状態が検出された場合
 - FIFO オーバーラン状態が検出された場合 (リクエストと転送は DMA が内部で管理するため、このエラーはメモリ間モードでは検出されない)
 - FIFO 閾値レベルがメモリバーストのサイズに適合していないときにストリームが有効化された場合 ([表 34：FIFO 閾値設定](#)を参照)
- **ダイレクトモードエラー：**ダイレクトモードで動作していて DMA_SxCR レジスタの MINC ビットがクリアされたときのみ、“ペリフェラルからメモリ”モードにおいてダイレクトモードエラー割込みフラグ (DMEIFx) がセットされることがあります。このフラグは、メモリバスの使用権が与えられていなかったことにより、前回のデータがまだ完全にメモリに転送されていない状態で DMA リクエストが発生したときにセットされます。このケースでは、このフラグは 2 つのデータ項目が同じ転送先アドレスに連続して転送されたことを示し、転送先がこの状況を管理できない場合に、このことが問題となる可能性があります。

ダイレクトモードでは、FIFO エラーフラグが以下の条件でセットされることもあります。

- “ペリフェラルからメモリ”モードでは、複数のペリフェラルリクエストのためにメモリバスの使用が許可されない場合、FIFO が飽和 (オーバーラン) することがあります。
- “メモリからペリフェラル”モードでは、ペリフェラルリクエストが発生する前に、メモリバスの使用が許可されていない場合、アンダーラン状態が発生する可能性があります。

バーストサイズと FIFO 閾値レベルが適合しないことによって TEIFx または FEIFx フラグがセットされた場合、欠陥のあるストリームは対応するストリーム設定レジスタ (DMA_SxCR) の EN ビットがハードウェアによってクリアされることにより、自動的に無効化されます。

オーバーランまたはアンダーラン状態によって DMEIFx または FEIFx フラグがセットされる場合、欠陥のあるストリームは自動的に無効化されず、DMA_SxCR レジスタの EN ビットをリセットする

ことによってストリームを無効化するかどうかはソフトウェアに任されています。なぜならば、この種のエラーが発生してもデータの損失がないためです。

DMA_LISR または DMA_HISR レジスタにあるストリームのエラー割込みフラグ (TEIF、FEIF、DMEIF) がセットされると、DMA_SxCR または DMA_SxFCR レジスタの対応する割込みイネーブルビット (TEIE、FEIE、DMIE) がセットされていれば、割込みが発生します。

注： FIFO オーバーランまたはアンダーラン状態が発生した場合、オーバーランまたはアンダーラン状態がクリアされるまではペリフェラルリクエストがストリームによって承認されないため、データが失われることはありません。この承認に時間がかかりすぎた場合、ペリフェラル自体がその内部バッファのオーバーランまたはアンダーラン状態を検出し、データが失われる可能性があります。

9.4 DMA 割込み

各 DMA ストリームに対し、次のイベントにて割込みを生成することができます。

- 1/2 転送終了
- 転送完了
- 転送エラー
- FIFO エラー (オーバーラン、アンダーラン、または FIFO レベルのエラー)
- ダイレクトモードエラー

柔軟性を持たせるため、表 36 に示すよう個別の割込みイネーブル制御ビットが使用できます。

表 36. DMA 割込みリクエスト

割込みイベント	イベントフラグ	有効制御ビット
1/2 転送	HTIF	HTIE
転送完了	TCIF	TCIE
転送エラー	TEIF	TEIE
FIFO オーバーラン／アンダーラン	FEIF	FEIE
ダイレクトモードエラー	DMEIF	DMEIE

注： イネーブル制御ビットを '1' に設定する前に、対応するイベントフラグをクリアする必要があります。さもないと直ちに割込みが生成されてしまいます。

9.5 DMA レジスタ

DMA レジスタには、ワード (32 ビット) 単位でアクセスすることができます。

9.5.1 DMA ロー割込みステータスレジスタ (DMA_LISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み				TCIF3	HTIF3	TEIF3	DMEIF3	予約済み	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	予約済み	FEIF2
r	r	r	r	r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				TCIF1	HTIF1	TEIF1	DMEIF1	予約済み	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	予約済み	FEIF0
r	r	r	r	r	r	r	r		r	r	r	r	r		r

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **TCIFx** : ストリーム x 転送完了割込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送完了イベントは発生していません。

1 : ストリーム x に転送完了イベントが発生しました。

ビット 26、20、10、4 **HTIFx** : ストリーム x 1/2 転送割込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に 1/2 転送イベントは発生していません。

1 : ストリーム x に 1/2 転送イベントが発生しました。

ビット 25、19、9、3 **TEIFx** : ストリーム x 転送エラー割込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送エラーは発生していません。

1 : ストリーム x に転送エラーが発生しました。

ビット 24、18、8、2 **DMEIFx** : ストリーム x ダイレクトモードエラー割込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x にダイレクトモードエラーは発生していません。

1 : ストリーム x にダイレクトモードエラーが発生しました。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **FEIFx** : ストリーム x FIFO エラー割込みフラグ (x=3..0)

このビットは、ハードウェアによってセットされます。DMA_LIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に FIFO エラーイベントは発生していません。

1 : ストリーム x に FIFO エラーイベントが発生しました。

9.5.2 DMA ハイ割込みステータスレジスタ (DMA_HISR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み				TCIF7	HTIF7	TEIF7	DMEIF7	予約済み	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	予約済み	FEIF6
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				TCIF5	HTIF5	TEIF5	DMEIF5	予約済み	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	予約済み	FEIF4
				r	r	r	r		r	r	r	r	r		r

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **TCIFx** : ストリーム x 転送完了割込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送完了イベントは発生していません。

1 : ストリーム x に転送完了イベントが発生しました。

ビット 26、20、10、4 **HTIFx** : ストリーム x 1/2 転送割込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に 1/2 転送イベントは発生していません。

1 : ストリーム x に 1/2 転送イベントが発生しました。

ビット 25、19、9、3 **TEIFx** : ストリーム x 転送エラー割込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に転送エラーは発生していません。

1 : ストリーム x に転送エラーが発生しました。

ビット 24、18、8、2 **DMEIFx** : ストリーム x ダイレクトモードエラー割込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x にダイレクトモードエラーは発生していません。

1 : ストリーム x にダイレクトモードエラーが発生しました。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **FEIFx** : ストリーム x FIFO エラー割込みフラグ (x=7..4)

このビットは、ハードウェアによってセットされます。DMA_HIFCR レジスタの対応するビットに 1 を書き込むことで、ソフトウェアによってクリアされます。

0 : ストリーム x に FIFO エラーイベントは発生していません。

1 : ストリーム x に FIFO エラーイベントが発生しました。

9.5.3 DMA ロー割込みフラグクリアレジスタ (DMA_LIFCR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み				CTCIF3	CHTIF3	CTEIF3	CDMEIF3	予約済み	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	予約済み	CFEIF2
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				CTCIF1	CHTIF1	CTEIF1	CDMEIF1	予約済み	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	予約済み	CFEIF0
				w	w	w	w		w	w	w	w	w		w

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **CTCIFx** : ストリーム x クリア転送完了割込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA_LISR レジスタの対応する TCIFx フラグをクリアします。

ビット 26、20、10、4 **CHTIFx** : ストリーム x クリア 1/2 転送割込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA_LISR レジスタの対応する HTIFx フラグをクリアします。

ビット 25、19、9、3 **CTEIFx** : ストリーム x クリア転送エラー割込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA_LISR レジスタの対応する TEIFx フラグをクリアします。

ビット 24、18、8、2 **CDMEIFx** : ストリーム x クリアダイレクトモードエラー割込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA_LISR レジスタの対応する DMEIFx フラグをクリアします。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **CFEIFx** : ストリーム x クリア FIFO エラー割込みフラグ (x=3..0)

このビットに 1 を書き込むと DMA_LISR レジスタの対応する CFEIFx フラグをクリアします。

9.5.4 DMA ハイ割込みフラグクリアレジスタ (DMA_HIFCR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み				CTCIF7	CHTIF7	CTEIF7	CDMEIF7	予約済み	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	予約済み	CFEIF6
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				CTCIF5	CHTIF5	CTEIF5	CDMEIF5	予約済み	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	予約済み	CFEIF4
				w	w	w	w		w	w	w	w	w		w

ビット 31:28、15:12 予約済みであり、リセット値に保持する必要があります。

ビット 27、21、11、5 **CTCIFx** : ストリーム x クリア転送完了割込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する TCIFx フラグをクリアします。

ビット 26、20、10、4 **CHTIFx** : ストリーム x クリア 1/2 転送割込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する HTIFx フラグをクリアします。

ビット 25、19、9、3 **CTEIFx** : ストリーム x クリア転送エラー割込みフラグ (x=7..4)

このビットに 1 を書き込むと DMA_HISR レジスタの対応する TEIFx フラグをクリアします。

ビット 24、18、8、2 **CDMEIFx** : ストリーム x クリアダイレクトモードエラー割込みフラグ (x=7..4)
このビットに 1 を書き込むと DMA_HISR レジスタの対応する DMEIFx フラグをクリアします。

ビット 23、17、7、1 予約済みであり、リセット値に保持する必要があります。

ビット 22、16、6、0 **CFEIFx** : ストリーム x クリア FIFO エラー割込みフラグ (x=7..4)
このビットに 1 を書き込むと DMA_HISR レジスタの対応する CFEIFx フラグをクリアします。

9.5.5 DMA ストリーム x 設定レジスタ (DMA_SxCR) (x=0~7)

このレジスタは当該ストリームの設定に使用されます。

アドレスオフセット : 0x10 + 0x18 × ストリーム番号

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み				CHSEL[3:0]			MBURST [1:0]		PBURST[1:0]		予約済み	CT	DBM または 予約済み	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw または r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:25 **CHSEL[2:0]** : チャンネル選択
これらのビットは、ソフトウェアによってセット／クリアされます。
000 : チャンネル 0 が選択されます。
001 : チャンネル 1 が選択されます。
010 : チャンネル 2 が選択されます。
011 : チャンネル 3 が選択されます。
100 : チャンネル 4 が選択されます。
101 : チャンネル 5 が選択されます。
110 : チャンネル 6 が選択されます。
111 : チャンネル 7 が選択されます。
これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ビット 24:23 **MBURST** : メモリバースト転送設定
これらのビットは、ソフトウェアによってセット／クリアされます。
00 : シングル転送
01 : INCR4 (4 ビートのインクリメンタルバースト)
10 : INCR8 (8 ビートのインクリメンタルバースト)
11 : INCR16 (16 ビートのインクリメンタルバースト)
これらのビットは保護されており EN が“0”の場合のみ書き込みできます。
ダイレクトモードでは、これらのビットは EN ビット=“1”になり次第ハードウェアによって強制的に 0x0 になります。

ビット 22:21 PBURST[1:0] : ペリフェラルバースト転送設定

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : シングル転送

01 : INCR4 (4 ビートのインクリメンタルバースト)

10 : INCR8 (8 ビートのインクリメンタルバースト)

11 : INCR16 (16 ビートのインクリメンタルバースト)

これらのビットは保護されており EN が“0”の場合のみ書き込みできます。

ダイレクトモードでは、これらのビットはハードウェアによって強制的に 0x0 になります。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19 CT : 現在のターゲット (ダブルバッファモード時のみ)

このビットは、ハードウェアによってセット／クリアされます。また、ソフトウェアによって書き込むこともできます。

0 : 現在のターゲットメモリはメモリ 0 (DMA_SxM0AR ポインタによってアドレス指定) です。

1 : 現在のターゲットメモリはメモリ 1 (DMA_SxM1AR ポインタによってアドレス指定) です。

このビットは、EN が '0' のときのみ書き込むことができ、最初の転送の対象メモリ領域を示します。ストリームが有効になると、このビットはステータスフラグとして動作し、どのメモリ領域が現在のターゲットかを示します。

ビット 18 DBM : ダブルバッファモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : 転送終了時にバッファを切り替えません

1 : DMA 転送終了時にメモリターゲットを切り替えます

このビットは保護されており、EN が“0”のときのみ書き込みできます。

ビット 17:16 PL[1:0] : 優先順位レベル

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : 低

01 : 中

10 : 高

11 : 最優先

これらのビットは保護されており EN が“0”のときのみ書き込みできます。

ビット 15 PINCOS : ペリフェラルインクリメントオフセットサイズ

このビットは、ソフトウェアによってセット／クリアされます。

0 : ペリフェラルアドレス計算のためのオフセットサイズは PSIZE と関連付けられています。

1 : ペリフェラルアドレス計算のためのオフセットサイズは 4 に固定されています (32 ビットアライメント)。

ビット PINC=“0”の場合、このビットは意味を持ちません。

このビットは保護されており EN=“0”のときのみ書き込みできます。

ダイレクトモードが選択されているか、PBURSTが“00”ではない場合、ストリームが有効になったとき (EN ビット=“1”) このビットはハードウェアによって強制的にローになります。

ビット 14:13 MSIZE[1:0] : メモリデータサイズ

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : バイト (8 ビット)

01 : ハーフワード (16 ビット)

10 : ワード (32 ビット)

11 : 予約済み

これらのビットは保護されており EN が“0”のときのみ書き込みできます。

ダイレクトモードでは、EN ビット=“1”になり次第、MSIZE はハードウェアによって強制的に PSIZE と同じ値になります。

ビット 12:11 PSIZE[1:0] : ペリフェラルデータサイズ

これらのビットは、ソフトウェアによってセット／クリアされます。

- 00 : バイト (8 ビット)
- 01 : ハーフワード (16 ビット)
- 10 : ワード (32 ビット)
- 11 : 予約済み

これらのビットは保護されており EN が“0”の場合のみ書込みできます。

ビット 10 MINC : メモリインクリメントモード

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : メモリアドレスポインタは固定されています。
- 1 : メモリアドレスポインタは、各データ転送後にインクリメントされます (MSIZE に応じてインクリメント)

このビットは保護されており、EN が“0”のときのみ書込みできます。

ビット 9 PINC : ペリフェラルインクリメントモード

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : ペリフェラルアドレスポインタは固定されています
- 1 : ペリフェラルアドレスポインタは、各データ転送後にインクリメントされます (PSIZE に応じてインクリメント)

このビットは保護されており、EN が“0”のときのみ書込みできます。

ビット 8 CIRC : サーキュラモード

このビットは、ソフトウェアによってセット／クリアされ、かつハードウェアによってクリアできます。

- 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。

ペリフェラルがフローコントローラ (PFCTRL ビット =1) となっていて、ストリームが有効なとき (ビット EN=1)、このビットは自動的にハードウェアによって 0 になります。

DBMビットがセットされている場合、ストリームが有効になり次第 (EN ビット =“1”) ハードウェアにより強制的に 1 になります。

ビット 7:6 DIR[1:0] : データ転送方向

これらのビットは、ソフトウェアによってセット／クリアされます。

- 00 : ペリフェラルからメモリ
- 01 : メモリからペリフェラル
- 10 : メモリ間転送
- 11 : 予約済み

これらのビットは保護されており EN が“0”のときのみ書込みできます。

ビット 5 PFCTRL : ペリフェラルフローコントローラ

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : DMA がフローコントローラ
- 1 : ペリフェラルがフローコントローラ

このビットは保護されており、EN が“0”のときのみ書込みできます。

メモリ間モードが選択されている場合 (DIR[1:0] ビット =10)、このビットは自動的にハードウェアにより強制的に 0 になります。

ビット 4 TCIE : 転送完了割込み有効化

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 : 転送完了割込みは無効です。
- 1 : 転送完了割込みは有効です。

ビット 3 HTIE : 1/2 転送割込みイネーブル

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 : 1/2 転送割込みは無効です。
- 1 : 1/2 転送割込みは有効です。

- ビット 2 **TEIE** : 転送エラー割込み有効化
- 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 : 転送エラー割込みは無効です。
- 1 : 転送エラー割込みは有効です。
- ビット 1 **DMEIE** : ダイレクトモードエラー割込みイネーブル
- 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 : ダイレクトモードエラー割込みは無効です。
- 1 : ダイレクトモードエラー割込みは有効です。
- ビット 0 **EN** : ストリームイネーブル／ロー読み出し時のストリームレディフラグ
- このビットは、ソフトウェアによってセット／クリアされます。
- 0 : ストリームは無効です。
- 1 : ストリームは有効です。
- このビットは、次のときハードウェアによってクリアできます。
- DMA 転送終了時 (ストリームが設定準備完了)
 - AHB マスタバス上で転送エラーが発生した場合
 - メモリ AHB ポート上の FIFO 閾値がバーストの大きさと適合しない場合
- このビットが 0 として読み出された場合、ソフトウェアによって設定レジスタと FIFO ビットレジスタをプログラムすることができます。EN ビットが 1 として読み出された場合、これらのレジスタの書き込みは禁止されています。
- 注 : 新しい転送を開始するために EN ビットを“1”に設定する前に、DMA_LISR または DMA_HISR レジスタのストリームに対応するイベントフラグをクリアする必要があります。

9.5.6 DMA ストリーム x データ数レジスタ (DMA_HIFCR) (x=0~7)

アドレスオフセット : 0x14 + 0x18 × ストリーム番号

リセット値 : 0x0000 0000

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

- ビット 31:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15:0 **NDT[15:0]** : 転送するデータ項目の数
- 転送されるデータ項目数 (0~65535)。このレジスタに書き込めるのは、ストリームが無効のときだけです。ストリームが有効になると、このレジスタは読み出し専用になり、送信されるべき残りのデータ項目を示します。このレジスタは、各 DMA 転送後にデクリメントされます。
- 転送が完了すると、このレジスタは、ゼロのままにすることもできれば (ストリームが通常モードのとき)、次の場合には事前にプログラムされた値を自動的に再ロードすることもできます。
- ストリームがサーキュラモードに設定されている場合
 - EN ビットを“1”に設定することにより、ストリームが再び有効になった場合
- このレジスタの値が 0 の場合、ストリームが有効であっても、トランザクションは処理できません。

9.5.7 DMA ストリーム x ペリフェラルアドレスレジスタ (DMA_SxPAR) (x=0~7)

アドレスオフセット : $0x18 + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

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

ビット 31:0 **PAR[31:0]** : ペリフェラルアドレス

データの読出し／書込みが行われるペリフェラルデータレジスタのベースアドレス。

これらのビットは書き込み保護されており、DMA_SxCR レジスタの EN ビット = "0" のときのみ書き込みできます。

9.5.8 DMA ストリーム x メモリ 0 アドレスレジスタ (DMA_SxM0AR) (x=0~7)

アドレスオフセット : $0x1C + 0x18 \times \text{ストリーム番号}$

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **M0A[31:0]** : メモリ 0 アドレス

データの読出し／書込みが行われるメモリ領域 0 のベースアドレス。

これらのビットは書き込み保護されています。次の場合にのみ書き込みが可能です。

- ストリームが無効 (DMA_SxCR レジスタの EN = "0") の場合、または
- ストリームが有効 (DMA_SxCR レジスタの EN = "1") で DMA_SxCR レジスタのビット CT="1" (ダブルバッファモード) の場合。

9.5.9 DMA ストリーム x メモリ 1 アドレスレジスタ (DMA_SxM1AR) (x=0~7)

アドレスオフセット : 0x20 + 0x18 × ストリーム番号

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M1A[31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1A[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **M1A[31:0]** : メモリ 1 アドレス (ダブルバッファモードの場合に使用)

データの読出し/書込みが行われるメモリ領域 1 のベースアドレス。

このレジスタは、ダブルバッファモードでのみ使用されます。

これらのビットは書き込み保護されています。次の場合にのみ書き込みが可能です。

- ストリームが無効 (DMA_SxCR レジスタの EN =“0”) の場合、または
- ストリームが有効 (DMA_SxCR レジスタの EN =“1”) で DMA_SxCR レジスタのビット CT=“0”の場合。

9.5.10 DMA ストリーム x FIFO 制御レジスタ (DMA_SxFCR) (x=0~7)

アドレスオフセット : 0x24 + 0x24 × ストリーム番号

リセット値 : 0x0000 0021

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **FEIE** : FIFO エラー割込みイネーブル

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 : FIFO エラー割込みは無効です。

1 : FIFO エラー割込みは有効です。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5:3 FS[2:0] : FIFO ステータス

これらのビットは読み出し専用です。

000 : $0 < \text{FIFO レベル} < 1/4$

001 : $1/4 \leq \text{FIFO レベル} < 1/2$

010 : $1/2 \leq \text{FIFO レベル} < 3/4$

011 : $3/4 \leq \text{FIFO レベル} < \text{フル}$

100 : FIFO はエンプティです。

101 : FIFO はフルです。

その他 : 意味を持ちません。

これらのビットは、ダイレクトモードには無関係です (DMDIS ビットがゼロ)。

ビット 2 DMDIS : ダイレクトモードディセーブル

このビットは、ソフトウェアによってセット／クリアされます。ハードウェアによってセットすることができます。

0 : ダイレクトモードは有効です。

1 : ダイレクトモードは無効です。

このビットは保護されており、EN が“0”のときのみ書込みできます。

このビットは、メモリ間モードが選択されており (DMA_SxCR の DIR ビットが“10”)、DMA_SxCR レジスタの EN ビットが“1”の場合、ハードウェアによってセットされます。その理由は、ダイレクトモードではメモリ間転送設定が使用できないからです。

ビット 1:0 FTH[1:0] : FIFO 閾値選択

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : FIFO フルの $1/4$

01 : FIFO フルの $1/2$

10 : FIFO フルの $3/4$

11 : FIFO フル

これらのビットは、DMIS 値がゼロのときダイレクトモードで使用されません。

これらのビットは保護されており EN が“0”のときのみ書込みできます。

9.5.11 DMA レジスタマップ

表 37 に DMA レジスタの要約を示します。

表 37. DMA レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
0x0000	DMA_LISR	予約済み				TCIF3	HTIF3	TEIF3	DMEIF3	予約済み	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	予約済み	FEIF2	予約済み				TCIF1	HTIF1	TEIF1	DMEIF1	予約済み	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	予約済み	FEIF0							
	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						
0x0004	DMA_HISR	予約済み				TCIF7	HTIF7	TEIF7	DMEIF7	予約済み	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	予約済み	FEIF6	予約済み				TCIF5	HTIF5	TEIF5	DMEIF5	予約済み	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	予約済み	FEIF4							
	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						
0x0008	DMA_LIFCR	予約済み				CTCIF3	CHTIF3	CTEIF3	CDMEIF3	予約済み	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	予約済み	CFEIF2	予約済み				CTCIF1	CHTIF1	CTEIF1	CDMEIF1	予約済み	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	予約済み	CFEIF0							
	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						
0x000C	DMA_HIFCR	予約済み				CTCIF7	CHTIF7	CTEIF7	CDMEIF7	予約済み	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	予約済み	CFEIF6	予約済み				CTCIF5	CHTIF5	CTEIF5	CDMEIF5	予約済み	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	予約済み	CFEIF4							
	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						
0x0010	DMA_S0CR	予約済み				CHSEL[2:0]		MBURST[1:0]			PBURST[1:0]		予約済み		CT	DBM	PL[1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN							
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x0014	DMA_S0NDTR	予約済み																NDT[15:]																						
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	DMA_S0PAR	PA[31:0]																																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x001C	DMA_S0M0AR	M0A[31:0]																																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x0020	DMA_S0M1AR	M1A[31:0]																																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x0024	DMA_S0FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS	FTH[1:0]									
	リセット値																								0	1	0	0	0	0	1									
0x0028	DMA_S1CR	予約済み				CHSEL[2:0]		MBURST[1:1]			PBURST[1:0]		ACK	CT	DBM	PL[1:0]	PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN								
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x002C	DMA_S1NDTR	予約済み																NDT[15:]																						
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	DMA_S1PAR	PA[31:0]																																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x0034	DMA_S1M0AR	M0A[31:0]																																						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							



表 37. 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
0x0038	DMA_S1M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x003C	DMA_S1FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS	FTH [1:0]		
	リセット値																									0		1	0	0	0	0	1
0x0040	DMA_S2CR	予約済み	CHSEL [2:0]		MBURST[1:0]		PBURST[1:0]		ACK	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x0044	DMA_S2NDTR	予約済み														NDT[15:]																	
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	DMA_S2PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x004C	DMA_S2M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0050	DMA_S2M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0054	DMA_S2FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS	FTH [1:0]		
	リセット値																									0		1	0	0	0	0	1
0x0058	DMA_S3CR	予約済み	CHSEL[2:0]		MBURST[1:0]		PBURST[1:0]		ACK	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x005C	DMA_S3NDTR	予約済み														NDT[15:]																	
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	DMA_S3PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0064	DMA_S3M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0068	DMA_S3M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x006C	DMA_S3FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS	FTH [1:0]		
	リセット値																									0		1	0	0	0	0	0
0x0070	DMA_S4CR	予約済み	CHSEL[2:0]		MBURST[1:0]		PBURST[1:0]		ACK	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR [1:0]		PFCCTRL	TCIE	HTIE	TEIE	DMEIE	EN				
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x0074	DMA_S4NDTR	予約済み														NDT[15:]																	
	リセット値															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 37. 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
0x0078	DMA_S4PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x007C	DMA_S4M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0080	DMA_S4M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	DMA_S4FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS		FTH [1:0]	
	リセット値																									0		1	0	0	0	0	1
0x0088	DMA_S5CR	予約済み				CHSEL[2:0]		MBURST[1:0]		PBURST[1:0]		ACK	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN	
	リセット値					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008C	DMA_S5NDTR	予約済み															NDT[15:]																
	リセット値																0																
0x0090	DMA_S5PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094	DMA_S5M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	DMA_S5M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x009C	DMA_S5FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS		FTH [1:0]	
	リセット値																									0		1	0	0	0	0	1
0x00A0	DMA_S6CR	予約済み				CHSEL[2:0]		MBURST[1:0]		PBURST[1:0]		ACK	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN	
	リセット値					0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A4	DMA_S6NDTR	予約済み															NDT[15:]																
	リセット値																0																
0x00A8	DMA_S6PAR	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
0x00AC	DMA_S6M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	DMA_S6M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B4	DMA_S6FCR	予約済み																								FEIE	予約済み	FS[2:0]		DMDIS		FTH [1:0]	
	リセット値																									0		1	0	0	0	0	1
0x00B8	DMA_S7CR	予約済み				CHSEL[2:0]		MBURST[1:0]		PBURST[1:0]		ACK	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN	
	リセット値					0				0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x00BC	DMA_S7NDTR	予約済み															NDT[15:]																
	リセット値																0																
0x00C0	DMA_S7PAR	PA[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	DMA_S7M0AR	M0A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



表 37. 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
0x00C8	DMA_S7M1AR	M1A[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00CC	DMA_S7FCR	予約済み																								FEIE	予約済み	FS[2:0]			DMDIS	FTH [1:0]	
	リセット値																									0		1	0	0	0	0	0

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

10 割込みとイベント

10.1 ネスト化されたベクタ割込みコントローラ（NVIC）

10.1.1 NVIC の機能

ネスト化されたベクタ割込みコントローラ NVIC には、次のような機能があります。

- 52 本のマスク可能な割込みチャネル（FPU 搭載 Cortex[®]-M4 の 16 本の割込みラインは数に含めない）
- 16 のプログラム可能な優先レベル（4 ビットの割込み優先順位を使用）
- 遅延時間の少ない例外および割込み処理
- 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、割込み処理の遅延時間を小さくし、遅れて到着した割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込みは、NVIC によって管理されます。例外と NVIC プログラミングの詳細については、プログラミングマニュアル PM0214 を参照してください。

10.1.2 SysTick 較正值レジスタ

SysTick 較正值は 10500 に固定され、SysTick クロックを 10.5 MHz（HCLK を 84 MHz に設定した状態で HCLK/8）に設定した状態で 1 ms という基準タイムベースを与えます。

10.1.3 割込みベクタと例外ベクタ

表 38 は、STM32F401xB/C および STM32F401xD/E デバイスのベクタテーブルです。

10.2 外部割込み／イベントコントローラ（EXTI）

イベント／割込みリクエストを生成するための外部割込み／イベントコントローラは、最大 23 のエッジ検出回路で構成されます。各入力ラインは、個別に設定可能であり、タイプ（割込みまたはイベント）と対応するトリガイメント（立ち上がり、立ち下がり、または両方）を選択できます。また、ラインごとにマスクをかけることもできます。ペンディングレジスタは、割込みリクエストのステータスラインを維持します。

表 38. STM32F401xB/CSTM32F401xD/E のベクタテーブル

位置	優先順位	優先種別	項目（略称）	説明	アドレス
	-	-	-	予約済み	0x0000 0000
	-3	固定	リセット	リセット	0x0000 0004
	-2	固定	NMI	ノンマスカブル割込み、 クロックセキュリティシステム	0x0000 0008
	-1	固定	HardFault	あらゆる種類の異常	0x0000 000C
	0	設定可能	MemManage	メモリ管理	0x0000 0010
	1	設定可能	BusFault	プリフェッチ異常、メモリアクセス異常	0x0000 0014

表 38. STM32F401xB/CSTM32F401xD/Eのベクタテーブル（続き）

位置	優先順位	優先種別	項目（略称）	説明	アドレス
	2	設定可能	UsageFault	未定義命令または無効状態	0x0000 0018
	-	-	-	予約済み	0x0000 001C - 0x0000 002B
	3	設定可能	SVCall	SWI 命令によるシステムサービスコール	0x0000 002C
	4	設定可能	デバッグモニタ	デバッグモニタ	0x0000 0030
		-	-	予約済み	0x0000 0034
	5	設定可能	PendSV	ペンディング可能な システムサービスリクエスト	0x0000 0038
	6	設定可能	SysTick	システムティックタイマ	0x0000 003C
0	7	設定可能	WWDG	ウィンドウ型ウォッチドッグ割込み	0x0000 0040
1	8	設定可能	EXTI16 / PVD	EXTI ライン 16 割込み／PVD 検出割込み (EXTI ライン経由)	0x0000 0044
2	9	設定可能	EXTI21 / TAMP_STAMP	EXTI ライン 21 割込み／ タンパおよびタイムスタンプ割込み (EXTI ライン経由)	0x0000 0048
3	10	設定可能	EXTI22 / RTC_WKUP	EXTI ライン 22 割込み／ RTC ウェイクアップ割込み (EXTI ライン経由)	0x0000 004C
4	11	設定可能	FLASH	フラッシュグローバル割込み	0x0000 0050
5	12	設定可能	RCC	RCC グローバル割込み	0x0000 0054
6	13	設定可能	EXTI0	EXTI ライン 0 割込み	0x0000 0058
7	14	設定可能	EXTI1	EXTI ライン 1 割込み	0x0000 005C
8	15	設定可能	EXTI2	EXTI ライン 2 割込み	0x0000 0060
9	16	設定可能	EXTI3	EXTI ライン 3 割込み	0x0000 0064
10	17	設定可能	EXTI4	EXTI ライン 4 割込み	0x0000 0068
11	18	設定可能	DMA1_Stream0	DMA1 ストリーム 0 グローバル割込み	0x0000 006C
12	19	設定可能	DMA1_Stream1	DMA1 ストリーム 1 グローバル割込み	0x0000 0070
13	20	設定可能	DMA1_Stream2	DMA1 ストリーム 2 グローバル割込み	0x0000 0074
14	21	設定可能	DMA1_Stream3	DMA1 ストリーム 3 グローバル割込み	0x0000 0078
15	22	設定可能	DMA1_Stream4	DMA1 ストリーム 4 グローバル割込み	0x0000 007C
16	23	設定可能	DMA1_Stream5	DMA1 ストリーム 5 グローバル割込み	0x0000 0080
17	24	設定可能	DMA1_Stream6	DMA1 ストリーム 6 グローバル割込み	0x0000 0084
18	25	設定可能	ADC	ADC1 グローバル割込み	0x0000 0088
23	30	設定可能	EXTI9_5	EXTI ライン [9:5] 割込み	0x0000 009C
24	31	設定可能	TIM1_BRK_TIM9	TIM1 ブレーク割込みおよび TIM9 グローバル割込み	0x0000 00A0
25	32	設定可能	TIM1_UP_TIM10	TIM1 更新割込みおよび TIM10 グローバル割込み	0x0000 00A4
26	33	設定可能	TIM1_TRG_COM_TIM11	TIM1 トリガおよび転流割込みおよび TIM11 グローバル割込み	0x0000 00A8

表 38. STM32F401xB/CSTM32F401xD/Eのベクタテーブル（続き）

位置	優先順位	優先種別	項目（略称）	説明	アドレス
27	34	設定可能	TIM1_CC	TIM1 キャプチャ / 比較割込み	0x0000 00AC
28	35	設定可能	TIM2	TIM2 グローバル割込み	0x0000 00B0
29	36	設定可能	TIM3	TIM3 グローバル割込み	0x0000 00B4
30	37	設定可能	TIM4	TIM4 グローバル割込み	0x0000 00B8
31	38	設定可能	I2C1_EV	I ² C1 イベント割込み	0x0000 00BC
32	39	設定可能	I2C1_ER	I ² C1 エラー割込み	0x0000 00C0
33	40	設定可能	I2C2_EV	I ² C2 イベント割込み	0x0000 00C4
34	41	設定可能	I2C2_ER	I ² C2 エラー割込み	0x0000 00C8
35	42	設定可能	SPI1	SPI1 グローバル割込み	0x0000 00CC
36	43	設定可能	SPI2	SPI2 グローバル割込み	0x0000 00D0
37	44	設定可能	USART1	USART1 グローバル割込み	0x0000 00D4
38	45	設定可能	USART2	USART2 グローバル割込み	0x0000 00D8
40	47	設定可能	EXTI15_10	EXTI ライン [15:10] 割込み	0x0000 00E0
41	48	設定可能	EXTI17 / RTC_Alarm	EXTI ライン 17 割込み / RTC アラーム (A および B) 割込み (EXTI ライン経由)	0x0000 00E4
42	49	設定可能	EXTI18 / OTG_FS WKUP	EXTI ライン 18 割込み / USB OTG FS ウェイクアップ割込み (EXTI ライン経由)	0x0000 00E8
47	54	設定可能	DMA1_Stream7	DMA1 ストリーム 7 グローバル割込み	0x0000 00FC
49	56	設定可能	SDIO	SDIO グローバル割込み	0x0000 0104
50	57	設定可能	TIM5	TIM5 グローバル割込み	0x0000 0108
51	58	設定可能	SPI3	SPI3 グローバル割込み	0x0000 010C
56	63	設定可能	DMA2_Stream0	DMA2 ストリーム 0 グローバル割込み	0x0000 0120
57	64	設定可能	DMA2_Stream1	DMA2 ストリーム 1 グローバル割込み	0x0000 0124
58	65	設定可能	DMA2_Stream2	DMA2 ストリーム 2 グローバル割込み	0x0000 0128
59	66	設定可能	DMA2_Stream3	DMA2 ストリーム 3 グローバル割込み	0x0000 012C
60	67	設定可能	DMA2_Stream4	DMA2 ストリーム 4 グローバル割込み	0x0000 0130
67	74	設定可能	OTG_FS	USB OTG FS グローバル割込み	0x0000 014C
68	75	設定可能	DMA2_Stream5	DMA2 ストリーム 5 グローバル割込み	0x0000 0150
69	76	設定可能	DMA2_Stream6	DMA2 ストリーム 6 グローバル割込み	0x0000 0154
70	77	設定可能	DMA2_Stream7	DMA2 ストリーム 7 グローバル割込み	0x0000 0158
71	78	設定可能	USART6	USART6 グローバル割込み	0x0000 015C
72	79	設定可能	I2C3_EV	I ² C3 イベント割込み	0x0000 0160
73	80	設定可能	I2C3_ER	I ² C3 エラー割込み	0x0000 0164
81	88	設定可能	FPU	FPU グローバル割込み	0x0000 0184
84	91	設定可能	SPI4	SPI 4 グローバル割込み	0x0000 0190

- 割込み／イベントラインごとに独立したトリガとマスク
- 割込みラインごとに専用のステータスビット
- 最大 23 本のソフトウェアイベント／割込みリクエストの生成
- APB2 クロック周期よりも短いパルス幅の外部信号の検出。このパラメータの詳細については、STM32F4xx データシートの電気的特性のセクションを参照してください。

図 29 にブロック図を示します。

10.2.3 ウェイクアップイベント管理

STM32F4xx は、外部イベントや内部イベントを処理することによって、コアをウェイクアップできます (WFE)。ウェイクアップイベントは、次のいずれかによって生成できます。

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

外部ラインをウェイクアップイベントとして使用する方法については、[セクション 10.2.4: 機能詳細](#)を参照してください。

10.2.4 機能詳細

割込みを生成するには、割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込みラインで発生すると、割込みリクエストが生成されます。割込みラインに対応するペンディングビットもセットされます。このリクエストは、ペンディングレジスタに "1" を書き込むことによってリセットされます。

イベントを生成するには、イベントラインを設定し、有効にする必要があります。それには、2 つのトリガレジスタに希望するエッジ検出をプログラムし、イベントマスクレジスタの対応するビットに "1" を書き込むことによってイベントリクエストを有効にします。選択されたエッジがイベントラインで発生すると、イベントパルスが生成されます。イベントラインに対応するペンディングビットはセットされません。

割込み/イベントリクエストは、ソフトウェア割込み/イベントレジスタに "1" を書き込むことによって、ソフトウェアによって生成することもできます。

ハードウェア割込みの選択

23 本のラインを割込みソースとして設定するには、次の手順を使用します。

- 23 本の割込みラインのマスクビットを設定します (EXTI_IMR)。
- 割込みラインのトリガ選択ビットを設定します (EXTI_RTSR および EXTI_FTSR)。
- 23 本のラインのいずれかからの割込みを正しく処理できるように、外部割込みコントローラ (EXTI) にマップされた NVIC IRQ チャンネルを制御するイネーブルビットとマスクビットを設定します。

ハードウェアイベントの選択

23 本のラインをイベントソースとして設定するには、次の手順を使用します。

- 23 本のイベントラインのマスクビットを設定します (EXTI_EMR)。
- イベントラインのトリガ選択ビットを設定します (EXTI_RTSR および EXTI_FTSR)。

ソフトウェア割り込み／イベントの選択

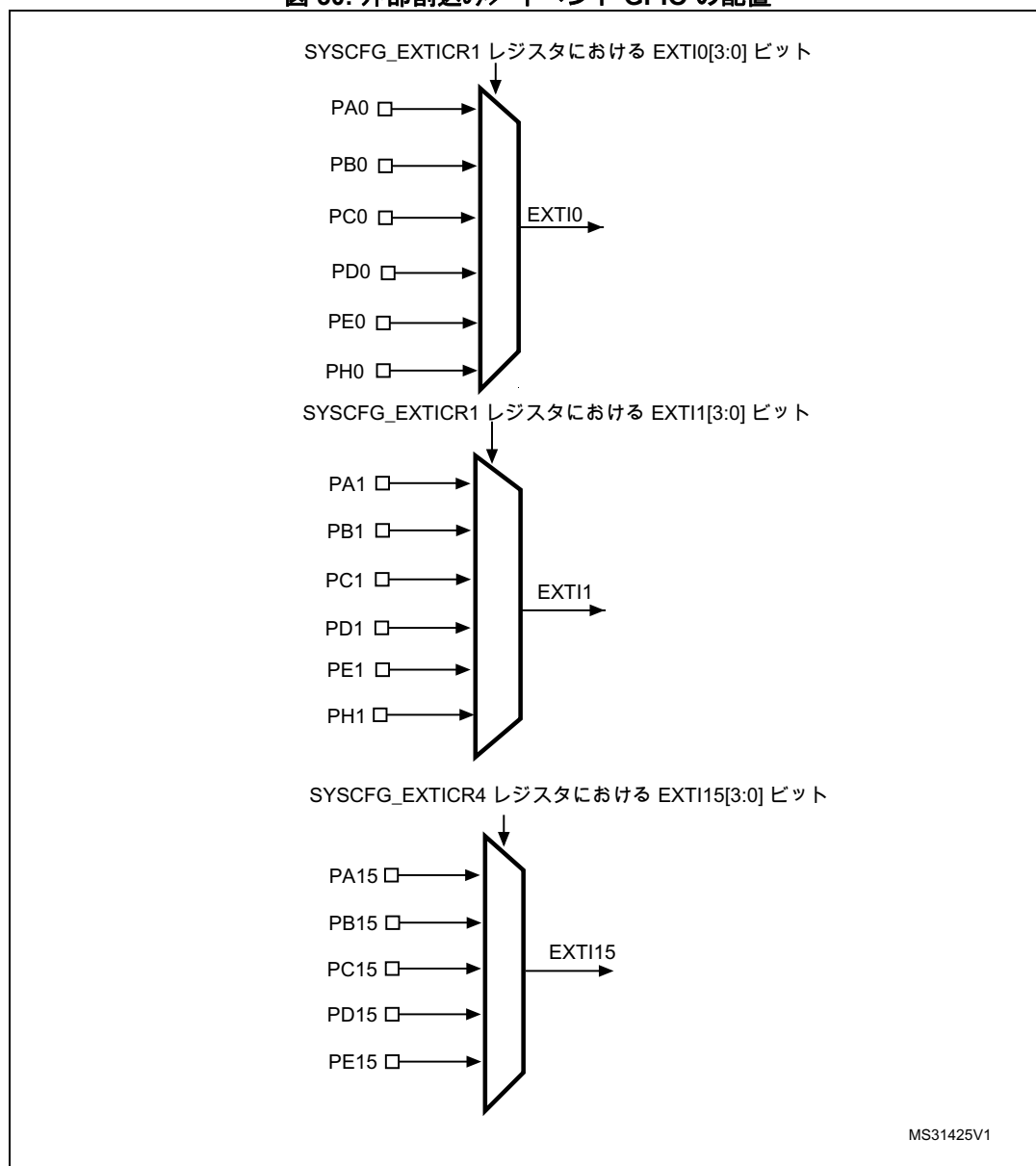
23 本のラインは、ソフトウェア割り込み／イベントラインとして設定できます。ソフトウェア割り込みの生成には、次の手順を使用します。

- 23 本の割り込み／イベントラインのマスクビットを設定します (EXTI_IMR、EXTI_EMR)。
- ソフトウェア割り込みレジスタの必要なビットをセットします (EXTI_SWIER)。

10.2.5 外部割り込み／イベントラインの配置

最大 81 個の GPIO (STM32F401xB/C および STM32F401xD/E) は、次のように、16 本の外部割り込み／イベントラインに接続されます。

図 30. 外部割り込み／イベント GPIO の配置



他の 5 本の EXTI ラインは次のように接続されます。

- EXTI ライン 16 は PVD 出力に接続されます。
- EXTI ライン 17 は RTC アラームイベントに接続されます。
- EXTI ライン 18 は USB OTG FS ウェイクアップイベントに接続されます。
- EXTI ライン 21 は RTC のタンパイイベントおよびタイムスタンプイベントに接続されます。
- EXTI ライン 22 は RTC ウェイクアップイベントに接続されます。

10.3 EXTI レジスタ

レジスタの説明で使用されている略語のリストについては、[34 ページのセクション 1.1](#) を参照してください。

10.3.1 割込みマスクレジスタ (EXTI_IMR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									MR22	MR21	予約済み		MR18	MR17	MR16
									rw	rw			rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **MRx** : ライン x の割込みマスク

- 0 : ライン x からの割込みリクエストはマスクされます。
- 1 : ライン x からの割込みリクエストはマスクされません。

10.3.2 イベントマスクレジスタ (EXTI_EMR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									MR22	MR21	予約済み		MR18	MR17	MR16
									rw	rw			rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **MRx** : ライン x のイベントマスク

- 0 : ライン x からのイベントリクエストはマスクされます。
- 1 : ライン x からのイベントリクエストはマスクされません。

10.3.3 立ち上がりトリガ選択レジスタ (EXTI_RTSTR)

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	
予約済み									TR22	TR21	予約済み			TR18	TR17	TR16
									r/w	r/w				r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **TRx** : ライン x の立ち上がりトリガイベント設定ビット
0 : 入力ラインの (イベントと割込みについて) 立ち上がりトリガは無効です。
1 : 入力ラインの (イベントと割込みについて) 立ち上がりトリガは有効です。

注 : 外部ウェイクアップラインはエッジトリガであるため、グリッチが生成されないようにする必要があります。
EXTI_RTSTR レジスタへの書込み中に外部割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。
同じ割込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この設定では、両方のエッジでトリガ条件が生成されます。

10.3.4 立ち下がりトリガ選択レジスタ (EXTI_FTSR)

アドレスオフセット : 0x0C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
予約済み									TR22	TR21	予約済み			TR18	TR17	TR16
									r/w	r/w				r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **TRx** : ライン x の立ち下がりトリガイベント設定ビット
0 : 入力ラインの (イベントと割込みについて) 立ち下がりトリガは無効です
1 : 入力ラインの (イベントと割込みについて) 立ち下がりトリガは有効です

注 : 外部ウェイクアップラインはエッジトリガであるため、グリッチが生成されないようにする必要があります。
EXTI_FTSR レジスタへの書込み中に外部割込みラインで立ち下がりエッジが発生した場合、ペンディングビットはセットされません。
同じ割込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この設定では、両方のエッジでトリガ条件が生成されます。

10.3.5 ソフトウェア割込みイベントレジスタ (EXTI_SWIER)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000

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

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **SWIERx** : ライン x のソフトウェア割込み

割込みが EXTI_IMR レジスタのライン x で有効になっている場合、SWIERx ビットが“0”のときに“1”を書き込むと EXTI_PR レジスタの対応するペンディングビットがセットされるため、割込みリクエストが生成されます。
EXTI_PR レジスタの対応するビットをクリアする（対応ビットに“1”を書き込む）と、このビットはクリアされます。

10.3.6 ペンディングレジスタ (EXTI_PR)

アドレスオフセット : 0x14
リセット値 : 定義されていません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									PR22	PR21	予約済み		PR18	PR17	PR16
									rc_w1	rc_w1			rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:0 **PRx** : ペンディングビット

0 : トリガリクエストは発生していません。
1 : 選択されたトリガリクエストが発生しました。
このビットは、選択されたエッジイベントが外部割込みラインで発生したときにセットされます。
このビットは、“1”をプログラムすることによってクリアされます。

10.3.7 EXTI レジスタマップ

表 39 に、EXTI レジスタマップとリセット値を示します。

表 39. 外部割込み／イベントコントローラのレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_IMR	予約済み										MR [22:21]	予約 済み	MR[18:0]																			
	リセット値											0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR	予約済み										MR [22:21]	予約 済み	MR[18:0]																			
	リセット値											0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_RTSR	予約済み										TR [22:21]	予約 済み	TR[18:0]																			
	リセット値											0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	EXTI_FTSR	予約済み										TR [22:21]	予約 済み	TR[18:0]																			
	リセット値											0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	EXTI_SWIER	予約済み										SWIER [22:21]	予約 済み	SWIER[18:0]																			
	リセット値											0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	EXTI_PR	予約済み										PR [22:21]	予約 済み	PR[18:0]																			
	リセット値											0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、38 ページの表 1 を参照してください。

11 アナログデジタルコンバータ (ADC)

ADC2 および ADC3 は、STM32F401xB/C および STM32F401xD/E で使用できません。

11.1 ADC の概要

この 12 ビット ADC は、逐次比較型アナログデジタルコンバータです。最大 19 の多重化チャネルを持ち、16 の外部ソース、2 つの内部ソース、および V_{BAT} チャネルの測定を行うことができます。これらのチャネルの A/D 変換は、シングル、連続、スキャン、または不連続モードで行うことができます。ADC の結果は、左詰めまたは右詰めで 16 ビットのデータレジスタに格納されます。

アナログウォッチドッグ機能により、入力電圧が、ユーザ定義の上限値または下限値から逸脱していないかを、アプリケーションで検出することができます。

11.2 ADC の主な機能

- 12 ビット、10 ビット、8 ビット、または 6 ビットの設定可能な分解能
- 変換終了時、インジェクト変換終了時、および、アナログウォッチドッグイベントまたはオーバーランイベント時に割込みを生成
- シングルおよび連続変換モード
- チャネル 0 からチャネル“n”を自動変換するためのスキャンモード
- データコヒーレンシ機能内蔵による変換データ配置調整機能
- チャネル単位でプログラム可能なサンプリング時間
- レギュラ変換とインジェクト変換の両方について極性設定可能な外部トリガオプション
- 不連続モード
- ADC 電源仕様：フルスピード時 2.4 V から 3.6 V、低速時 1.8 V まで対応可能
- ADC 入力電圧範囲： $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- レギュラチャネル変換中の DMA リクエストの生成

[図 31](#) に、ADC のブロック図を示します。

注： V_{REF-} ピンが用意されているパッケージでは、このピンを V_{SSA} に固定する必要があります。

11.3 ADC の機能詳細

図 31 に 1 つの ADC ブロック図を、表 40 に ADC ピンの説明を示します。

図 31.1 つの ADC ブロック図

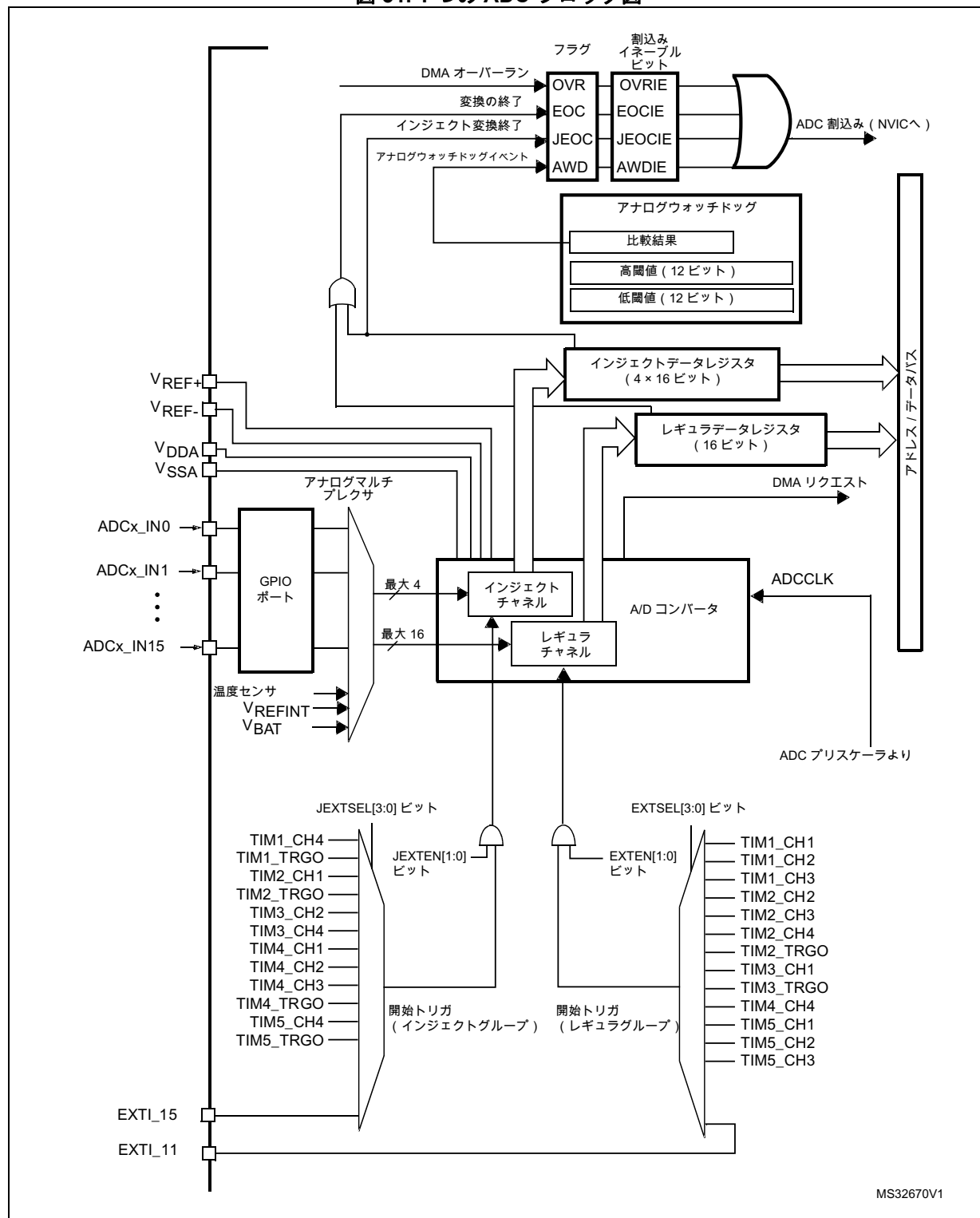


表 40. ADC ピン

名前	信号タイプ	説明
V_{REF+}	入力、アナログ基準電圧正	ADC のハイレベル／正基準電圧、 $1.8\text{ V} \leq V_{REF+} \leq V_{DDA}$
V_{DDA}	入力、アナログ電源供給	アナログ電源供給、 V_{DD} に等しく、 $2.4\text{ V} \leq V_{DDA} \leq V_{DD}$ (3.6 V) フルスピード用 $1.8\text{ V} \leq V_{DDA} \leq V_{DD}$ (3.6 V) 低速用
V_{REF-}	入力、アナログ基準電圧負	ADC のローレベル／負基準電圧、 $V_{REF-} = V_{SSA}$
V_{SSA}	入力、アナログ供給グラウンド	V_{SS} に等しいアナログ電源供給のグラウンド
ADCx_IN[15:0]	アナログ入力信号	16 チャンネルのアナログ入力

11.3.1 ADC のオン／オフ制御

ADC は、ADC_CR2 レジスタの ADON ビットをセットすることで、電源オンの状態になります。ADON ビットが最初にセットされると、ADC はパワーダウンモードからウェイクアップします。

SWSTART ビットまたは JSWSTART ビットのいずれかがセットされると変換が開始します。

ADON ビットをクリアすることによって、変換を停止し、ADC をパワーダウンモードにできます。このモードでは、ADC は電力をほとんど消費しません (数 μA 程度)。

11.3.2 ADC クロック

この ADC には 2 つのクロックスキームがあります。

- アナログ回路のクロック : ADCCLK
このクロックは、プログラム可能なプリスケアラによって分周された APB2 クロックによって生成され、ADC は $f_{PCLK2}/2$ 、 $/4$ 、 $/6$ または $/8$ で動作することができます。ADCCLK の最大値に関してはデータシートを参照してください。
- デジタルインタフェース用のクロック (レジスタの読み書きアクセスに使用)
このクロックは APB2 クロックと同一です。デジタルインタフェースクロックは、RCC APB2 ペリフェラルクロック有効レジスタ (RCC_APB2ENR) を使用して ADC ごとに有効／無効にすることができます。

11.3.3 チャンネル選択

16 の多重化チャンネルがあります。変換は、レギュラとインジェクトの 2 つのグループに構成することができます。各グループは、任意のチャンネルに対して任意の順序で行うことができる一連の変換で構成されます。たとえば、次のような順で変換順を設定することができます : ADC_IN3、ADC_IN8、ADC_IN2、ADC_IN2、ADC_IN0、ADC_IN2、ADC_IN2、ADC_IN15。

- レギュラグループ**は、最大 16 の変換で構成されます。レギュラチャンネルと変換シーケンス内での順序は、ADC_SQRx レジスタで選択する必要があります。レギュラグループの合計変換数は、ADC_SQR1 レジスタの L[3:0] ビットに書き込む必要があります。
- インジェクトグループ**は、最大 4 つの変換で構成されます。インジェクトチャンネルと変換シーケンス内での順序は、ADC_JSQR レジスタで選択する必要があります。インジェクトグループの合計変換数は、ADC_JSQR レジスタの L[1:0] ビットに書き込む必要があります。

変換中に ADC_SQRx または ADC_JSQR レジスタが変更された場合、現在の変換はリセットされ、新しい開始パルスが ADC に送られて、新しく選ばれたグループを変換します。

温度センサ、V_{REFINT}、および V_{BAT} 内部チャネル

- 温度センサは VBAT と共用で ADC1_IN18 チャネルに内部的に接続されています。一度に選択できるのは、温度センサまたは VBAT のうち、どちらか 1 つだけです。温度センサと VBAT 変換を同時に設定すると、VBAT 変換のみ行われます。

内部基準電圧 VREFINT は ADC1_IN17 に接続されています。

V_{BAT} チャネルはチャネル ADC1_IN18 に接続されています。これも、インジェクトまたはレギュラチャネルとして変換を行うことができます。

注： 温度センサ、V_{REFINT} および V_{BAT} チャネルは、マスタ ADC1 ペリフェラルでのみ使用可能です。

11.3.4 シングル変換モード

シングル変換モードでは、ADC は変換を 1 回行います。CONT ビットが 0 の時、次のいずれかによってこのモードが開始します。

- ADC_CR2 レジスタの SWSTART ビットをセットすること（レギュラチャネルのみ）
- JSWSTART ビットをセットすること（インジェクトチャネル）
- 外部トリガ（レギュラまたはインジェクトチャネル）

選択されたチャネルの変換が完了すると、

- レギュラチャネルが変換された場合：
 - 変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
 - EOC（end of conversion）フラグがセットされます。
 - EOCIE ビットがセットされている場合、割込みが生成されます。
- インジェクトチャネルが変換された場合：
 - 変換されたデータは 16 ビットの ADC_JDR1 レジスタに格納されます。
 - JEOC（end of conversion injected）フラグがセットされます。
 - JEOCIE ビットがセットされている場合、割込みが生成されます。

その後、ADC が停止します。

11.3.5 連続変換モード

連続変換モードでは、ADC は 1 回の変換が終了すると次の変換に移ります。CONT ビットが 1 の時、外部トリガまたは ADC_CR2 レジスタの SWSTRT ビットをセットすることによって、このモードが開始されます（レギュラチャネルのみ）。

各変換後、

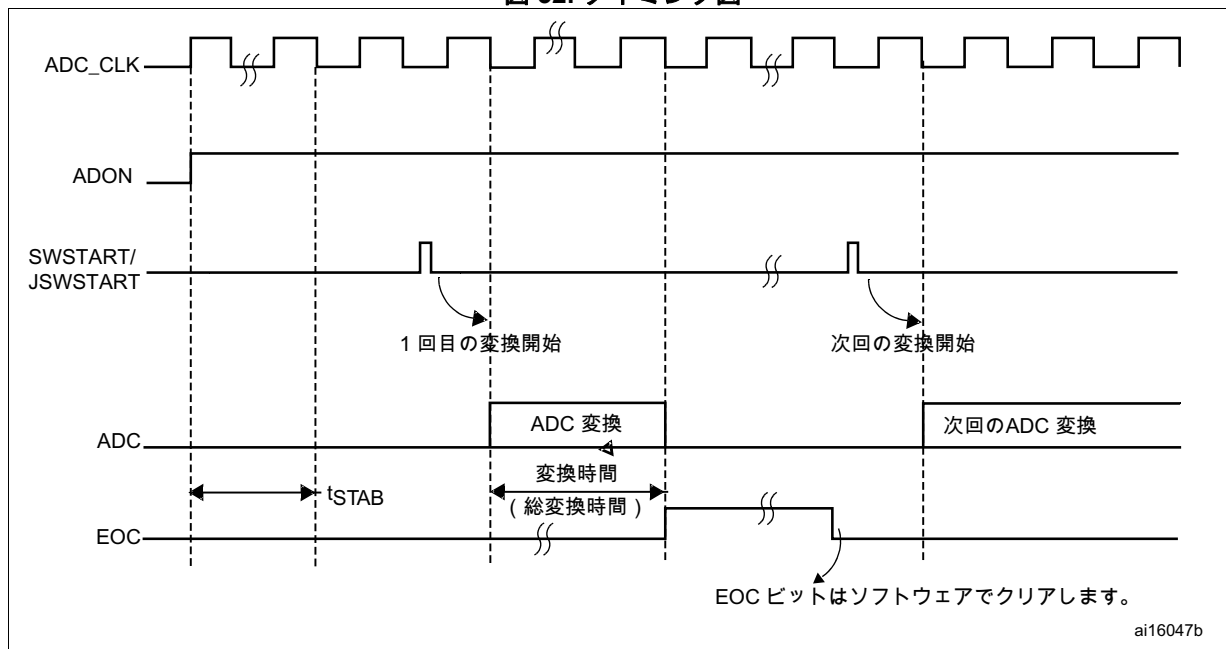
- レギュラチャネルのグループが変換された場合：
 - 最後に変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
 - EOC（end of conversion）フラグがセットされます。
 - EOCIE ビットがセットされている場合、割込みが生成されます。

注： インジェクトチャネルを連続して変換することはできません。唯一の例外は、連続モードに設定されたレギュラチャネルの後にインジェクトチャネルを自動的に変換するよう設定されている（JAUTO ビットを使用）場合です。自動インジェクションのセクションを参照してください。

11.3.6 タイミング図

図 32 に示すように、ADC は正確な変換を開始する前に、安定時間 t_{STAB} を必要とします。ADC 変換の開始後 15 クロックサイクルで EOC フラグがセットされ、変換されたデータが 16 ビットの ADC データレジスタに格納されます。

図 32. タイミング図



11.3.7 アナログウォッチドッグ

アナログウォッチドッグ (AWD) ステータスビットは、ADC によって変換されたアナログ電圧が低閾値を下回るか、高閾値を上回る場合にセットされます。これらの閾値は、ADC_HTR および ADC_LTR 16 ビットレジスタの下位 12 ビットに設定します。ADC_CR1 レジスタの AWDIE ビットを使用して、割込みを有効にできます。

閾値は、ADC_CR2 レジスタの ALIGN ビットによって選択された配置とは無関係です。アナログ電圧は配置される前に低閾値および高閾値と比較されます。

表 41 に、1 つまたは複数のチャネル上でアナログウォッチドッグを有効にするためにどのようにして ADC_CR1 レジスタを設定するかを示します。

図 33. アナログウォッチドッグによって保護される領域

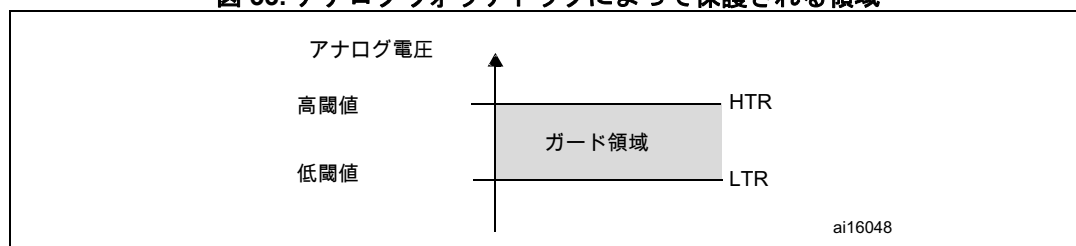


表 41. アナログウォッチドッグチャンネル選択

アナログウォッチドッグによって 保護されるチャンネル	ADC_CR1 レジスタ制御ビット (x = 無視)		
	AWDSGL ビット	AWDEN ビット	JAWDEN ビット
なし	x	0	0
すべてのインジェクトチャンネル	0	0	1
すべてのレギュラチャンネル	0	1	0
すべてのレギュラおよびインジェクトチャンネル	0	1	1
1つの ⁽¹⁾ インジェクトチャンネル	1	0	1
1つの ⁽¹⁾ レギュラチャンネル	1	1	0
1つの ⁽¹⁾ レギュラまたはインジェクトチャンネル	1	1	1

1. AWDC[4:0] ビットによって選択

11.3.8 スキャンモード

このモードは、アナログチャンネルのグループをスキャンするために使用されます。

スキャンモードを選択するには、ADC_CR1 レジスタの SCAN ビットをセットします。このビットがセットされると、ADC は ADC_SQRx レジスタ（レギュラチャンネルの場合）または ADC_JSQR レジスタ（インジェクトチャンネルの場合）で選択されたすべてのチャンネルをスキャンします。グループの各チャンネルに対して 1 回の変換が行われます。各変換が完了すると、グループ内の次のチャンネルが自動的に変換されます。CONT ビットがセットされている場合、グループの最後の選択チャンネルでレギュラチャンネル変換は停止せず、選択チャンネルの最初から続行します。

DMA ビットがセットされている場合、変換されたデータ（ADC_DR レジスタに格納されている）は、各レギュラチャンネル変換後、ダイレクトメモリアクセス（DMA）コントローラを使用して、レギュラチャンネルグループから SRAM に転送されます。

次の場合に ADC_SR レジスタの EOC ビットがセットされます。

- EOCs ビットが 0 にクリアされている場合、各レギュラグループシーケンス終了後
- EOCs ビットが 1 にセットされている場合、各レギュラチャンネル変換終了後

インジェクトチャンネルから変換されたデータは常に ADC_JDRx レジスタに格納されます。

11.3.9 インジェクトチャンネルの管理

トリガインジェクション

トリガインジェクションを使用するには、ADC_CR1 レジスタの JAUTO ビットがクリアされていなければなりません。

1. 外部トリガによって、または、ADC_CR2 レジスタの SWSTART ビットをセットすることによって、レギュラチャンネルのグループの変換を開始します。
2. レギュラチャンネルグループの変換中に外部インジェクトトリガが発生した場合または JSWSTART ビットがセットされた場合、現在の変換はリセットされ、インジェクトチャンネルシーケンスはスキャンワンスモードに切り替わります。
3. その後、レギュラチャンネルグループのレギュラ変換が、最後に中断されたレギュラ変換から再開されます。

インジェクト変換中にレギュライベントが発生した場合、インジェクト変換は中断されず、インジェクトシーケンスの終了時にレギュラシーケンスが実行されます。図 34 に対応するタイミング図を示します。

注：トリガインジェクションを使用する場合は、トリガイベントの間隔がインジェクションシーケンスより長くなるようにしなければなりません。たとえば、シーケンスの長さが 30 ADC クロックサイクルの場合 (3 クロック周期のサンプリング時間を持つ 2 つの変換)、トリガの最小間隔は 31 ADC クロックサイクルでなければなりません。

自動インジェクション

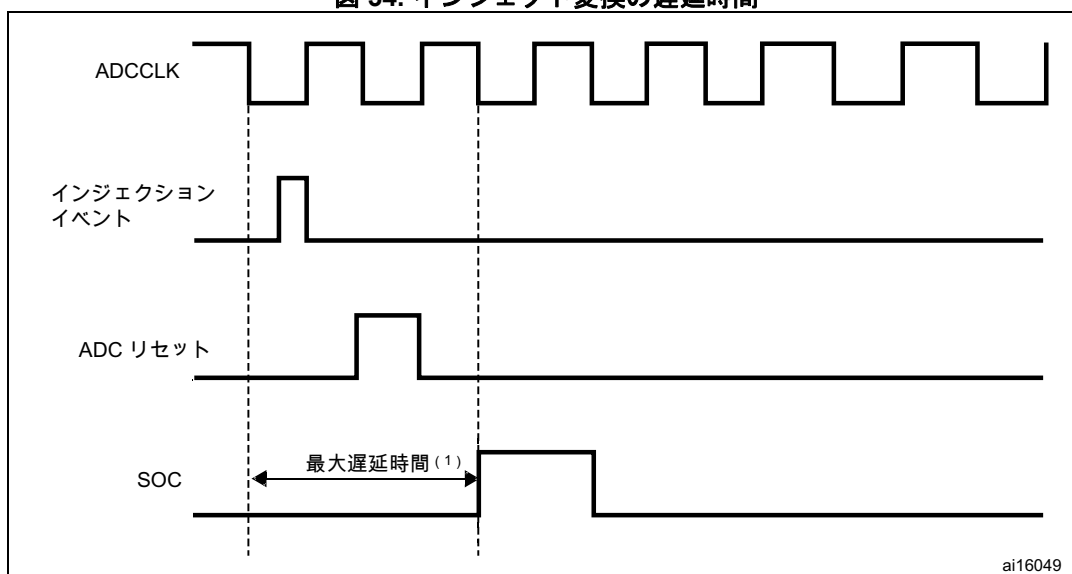
JAUTO ビットがセットされている場合、レギュラグループチャネルの変換後、インジェクトグループチャネルが自動的に変換されます。これを使用して、ADC_SQRx レジスタおよび ADC_JSQR レジスタにプログラミングされた最大 20 回の変換シーケンスを実行することができます。

このモードでは、インジェクトチャネルの外部トリガは無効でなければなりません。

JAUTO ビットに加えて CONT ビットもセットされている場合、レギュラチャネルに続いてインジェクトチャネルが連続して変換されます。

注：自動インジェクトモードと不連続モードを同時に使用することはできません。

図 34. インジェクト変換の遅延時間



1. 最大遅延時間の値は、STM32F401xB/C および STM32F401xD/E のデータシートの電気的特性に記載されています。

11.3.10 不連続モード

レギュラグループ

このモードは、ADC_CR1 レジスタの DISCEN ビットをセットすることによって有効になります。これを使用して、ADC_SQRx レジスタで選択された変換シーケンスの一部である短い n 個の変換シーケンス ($n \leq 8$) を変換できます。 n の値は、ADC_CR1 レジスタの DISCNUM[2:0] ビットに書き込むことによって指定します。

外部トリガが発生すると、ADC_SQRx レジスタで選択された次の n 個の変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC_SQR1 レジスタの L[3:0] ビットによって定義されます。

例：

- $n = 3$ 、変換されるチャネル = 0、1、2、3、6、7、9、10
- 最初のトリガ：0、1、2 が変換されます。各変換で EOC イベントが生成されます。
- 2 番目のトリガ：3、6、7 が変換されます。各変換で EOC イベントが生成されます。
- 3 番目のトリガ：9、10 が変換されます。各変換で EOC イベントが生成されます。
- 4 番目のトリガ：0、1、2 が変換されます。各変換で EOC イベントが生成されます。

注： レギュラグループが不連続モードで変換されるときには、ロールオーバーは発生しません。
すべてのサブグループが変換されると、次のトリガでは、最初のサブグループの変換が開始されます。
上の例では、4 番目のトリガで最初のサブグループのチャネル 0、1、および 2 が再変換されます。

インジェクトグループ

このモードは、ADC_CR1 レジスタの JDISCEN ビットをセットすることで有効になります。これを使用して、ADC_JSQR レジスタで選択されたシーケンスを、外部トリガイイベント後に 1 チャネルずつ変換することができます。

外部トリガが発生すると、ADC_JSQR レジスタで選択された次のチャネル変換が開始され、シーケンス内のすべての変換が行われるまで続きます。合計シーケンス長は、ADC_JSQR レジスタの JL[1:0] ビットによって定義されます。

例：

- $n = 1$ 、変換されるチャネル = 1、2、3
- 最初のトリガ：チャネル 1 が変換されます。
- 2 番目のトリガ：チャネル 2 が変換されます。
- 3 番目のトリガ：チャネル 3 が変換され、JEOC イベントが生成されます。
- 4 番目のトリガ：チャネル 1 が変換されます。

注： すべてのインジェクトチャネルが変換されると、次のトリガでは、最初のインジェクトチャネルの変換が開始されます。上の例では、4 番目のトリガで最初のインジェクトチャネル 1 が再変換されます。
自動インジェクトモードと不連続モードを同時に使用することはできません。
レギュラグループとインジェクトグループの両方で不連続モードを同時に設定することはできません。不連続モードは、1 つのグループ変換に対してのみ有効にする必要があります。

11.4 データの配置

ADC_CR2 レジスタの ALIGN ビットは、変換後に格納されるデータの配置を選択します。データは、[図 35](#) および [図 36](#) に示すように、右詰めまたは左詰めに配置できます。

インジェクトグループチャネルの変換後のデータ値は、ADC_JOFRx レジスタに書き込まれたユーザ定義オフセットだけ減算されるので結果は負の値になることもあります。SEXT ビットは、拡張符号値を表します。

レギュラグループのチャネルの場合、オフセットは引かれないので、12 ビットのみが有効です。

図 35. 12 ビットデータの右詰め

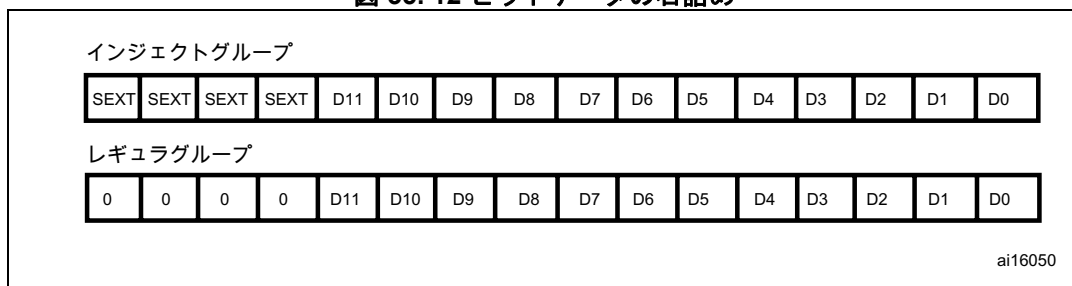
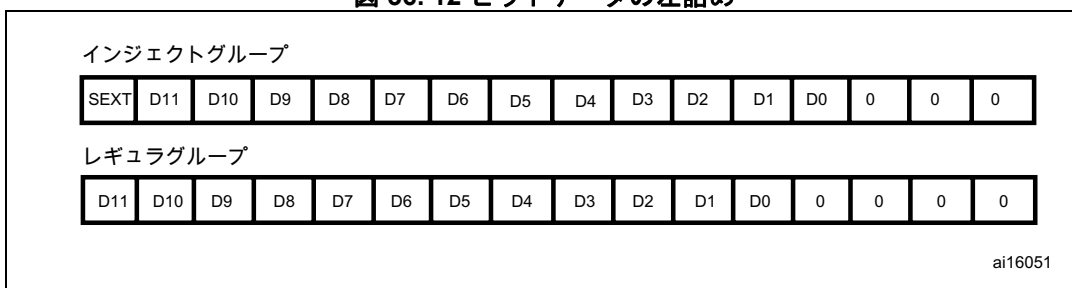


図 36. 12 ビットデータの左詰め



特殊なケース：左詰めの場合、バイト単位に配置される場合で分解能が 6 ビットに設定されている場合 ([図 37](#)) を除きデータはハーフワード単位に配置されます。

図 37. 6 ビットデータの左詰め



11.5 チャンネル単位でプログラム可能なサンプリング時間

ADC は、一定の ADCCLK サイクル数だけ入力電圧をサンプリングしますが、この時間は、ADC_SMPR1 および ADC_SMPR2 レジスタの SMP[2:0] ビットを使用して変更できます。チャンネルごとに異なるサンプリング時間でサンプリングすることができます。

合計変換時間は、次のように計算されます。

$$T_{\text{conv}} = \text{サンプリング時間} + 12 \text{ サイクル}$$

例：

ADCCLK = 30 MHz、サンプリング時間が 3 サイクルのとき：

$$T_{\text{conv}} = 3 + 12 = 15 \text{ サイクル} = 0.5 \mu\text{s} \text{ (APB2 が 60 MHz の場合)}$$

11.6 外部トリガによる変換およびトリガ極性

変換は、外部イベント（タイマキャプチャ、EXTI ラインなど）によってトリガできます。EXTEN[1:0] 制御ビット（レギュラ変換の場合）または JEXTEN[1:0] ビット（インジェクト変換の場合）が“0b00”以外の場合、選択した極性で外部イベントによって変換をトリガすることができます。表 42 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 42. トリガ極性の設定

転送元	EXTEN[1:0] / JEXTEN[1:0]
トリガ検出は無効です。	00
立ち上がりエッジで検出します。	01
立ち下がりエッジで検出します。	10
立ち上がりエッジと立ち下がりエッジの両方で検出します。	11

注：外部トリガの極性は動作中に変更することができます。

EXTSEL[3:0] および JEXTSEL[3:0] 制御ビットを使用してレギュラおよびインジェクトグループの変換をトリガできるイベントを 16 のイベントから選択します。

表 43 に、レギュラ変換に使用できる外部トリガを示します。

表 43. レギュラチャネルの外部トリガ

転送元	タイプ	EXTSEL[3:0]
TIM1_CH1 イベント	オンチップタイマからの内部信号	0000
TIM1_CH2 イベント		0001
TIM1_CH3 イベント		0010
TIM2_CH2 イベント		0011
TIM2_CH3 イベント		0100
TIM2_CH4 イベント		0101
TIM2_TRGO イベント		0110
TIM3_CH1 イベント		0111
TIM3_TRGO イベント		1000
TIM4_CH4 イベント		1001
TIM5_CH1 イベント		1010
TIM5_CH2 イベント		1011
TIM5_CH3 イベント		1100
予約済み		1101
予約済み		1110
EXTI ライン 11	外部ピン	1111

表 44 に、インジェクト変換に使用できる外部トリガを示します。

表 44. インジェクトチャネルの外部トリガ

転送元	接続タイプ	JEXTSEL[3:0]
TIM1_CH4 イベント	オンチップタイマからの内部信号	0000
TIM1_TRGO イベント		0001
TIM2_CH1 イベント		0010
TIM2_TRGO イベント		0011
TIM3_CH2 イベント		0100
TIM3_CH4 イベント		0101
TIM4_CH1 イベント		0110
TIM4_CH2 イベント		0111
TIM4_CH3 イベント		1000
TIM4_TRGO イベント		1001
TIM5_CH4 イベント		1010
TIM5_TRGO イベント		1011
予約済み		1100
予約済み		1101
予約済み		1110
EXTI ライン 15	外部ピン	1111

ソフトウェアソーストリガイイベントは、ADC_CR2 レジスタの SWSTART (レギュラ変換の場合) ビットまたは JSWSTART (インジェクト変換の場合) ビットをセットすることによって生成できます。

レギュラグループ変換は、インジェクトトリガによって中断することが可能です。

注：トリガの選択は動作中に変更することができます。ただし、選択が変更されるとき 1 APB クロックサイクルの期間トリガの検出が無効になります。これは、変更時の誤検出を避けるためです。

11.7 高速変換モード

ADC の分解能を下げることによって、高速変換を行うことができます。RES ビットを使用してデータレジスタで利用できるビット数を選択します。各精度に対する最小変換時間は、次のとおりです。

- 12 ビット : $3 + 12 = 15$ ADCCLK サイクル
- 10 ビット : $3 + 10 = 13$ ADCCLK サイクル
- 8 ビット : $3 + 8 = 11$ ADCCLK サイクル
- 6 ビット : $3 + 6 = 9$ ADCCLK サイクル

11.8 データ管理

11.8.1 DMA の使用

変換されたレギュラチャネルの値は特定のデータレジスタに格納されるので、複数のレギュラチャネルの変換には DMA の使用が便利です。これによって、ADC_DR レジスタにすでに格納されているデータの損失を防ぐことができます。

DMA モードが有効 (ADC_CR2 レジスタの DMA ビットが 1 に設定されている) な場合、レギュラチャネルが変換されるごとに DMA リクエストが生成されます。これにより、変換されたデータを ADC_DR レジスタからソフトウェアで選択した場所へ転送することができます。

これにもかかわらず、データが失われる場合 (オーバーラン)、ADC_SR レジスタの OVR ビットがセットされ割込みが生成されます (OVRIE イネーブルビットがセットされている場合)。続いて DMA 転送が無効になり、その後の DMA リクエストは受け付けられません。この場合、DMA リクエストが行われると、実行中のレギュラ変換は中断され、その後のレギュラトリガは無視されます。そのときは、必要な変換済みチャネルデータを正しいメモリ位置へ転送するために OVR フラグおよび使用している DMA ストリームの DMAEN ビットをクリアして DMA と ADC を再初期化する必要があります。その後初めて、変換を再開することができ、データ転送を再び有効にすることができます。インジェクトチャネル変換はオーバーランエラーの影響を受けません。

DMA モードで OVR = 1 の場合、最後の有効なデータが転送されたあとに DMA リクエストがブロックされます。つまり、RAM へ転送されたデータはすべて有効とみなすことができます。

最後の DMA 転送 (DMA コントローラの DMA_SxNTR レジスタに設定されている転送数) が終了すると、

- ADC_CR2 レジスタの DDS ビットが 0 にクリアされている場合、DMA コントローラへの新規の DMA リクエストは発生しません (これによりオーバーランエラーの発生を防ぎます)。ただし、DMA ビットはハードウェアによってクリアされることはありません。新たな転送を開始するにはそこに 0 を書き込み、そのあと 1 を書き込む必要があります。
- DDS ビットが 1 にセットされている場合、リクエストの生成は継続することができます。これにより DMA をダブルバッファサーキュラモードに設定することができます。

DMA 使用時に ADC を OVR 状態から回復するには次の手順に従います。

1. DMA を再初期化します (転送先アドレスと NDTR カウンタを調整します)
2. ADC_SR レジスタの ADC OVR ビットをクリアします。
3. ADC をトリガして変換を開始します。

11.8.2 DMA を使用しない変換シーケンスの管理

変換が十分遅い場合、ソフトウェアで変換シーケンスを処理することができます。この場合、変換シーケンスの終わりだけではなく各変換が終わるたびに EOC ステータスビットをセットするため、ADC_CR2 レジスタの EOCS ビットをセットする必要があります。EOCS = 1 のとき、オーバーラン検出は自動的に有効になります。したがって、変換が終了するごとに EOC がセットされ ADC_DR レジスタを読み出すことができます。オーバーラン管理は DMA を使用するときと同じです。

EOCS がセットされているとき ADC を OVR 状態から回復するには次の手順に従います。

1. ADC_SR レジスタの ADC OVR ビットをクリアします。
2. ADC をトリガして変換を開始します。

11.8.3 DMA およびオーバーラン検出を使用しない変換

毎回データの読出しをせずに ADC に 1 つまたは複数のチャンネルを変換させると便利な場合があります (たとえば、アナログウォッチドッグがある場合)。そのためには、DMA を無効 (DMA = 0) にし、シーケンスの終了時にのみ EOC ビットをセットする必要があります (EOCS = 0)。この設定では、オーバーラン検出が無効になります。

11.9 温度センサ

温度センサを使用して、デバイスの温度 (T_A) を測定できます。

図 38 に、温度センサのブロック図を示します。

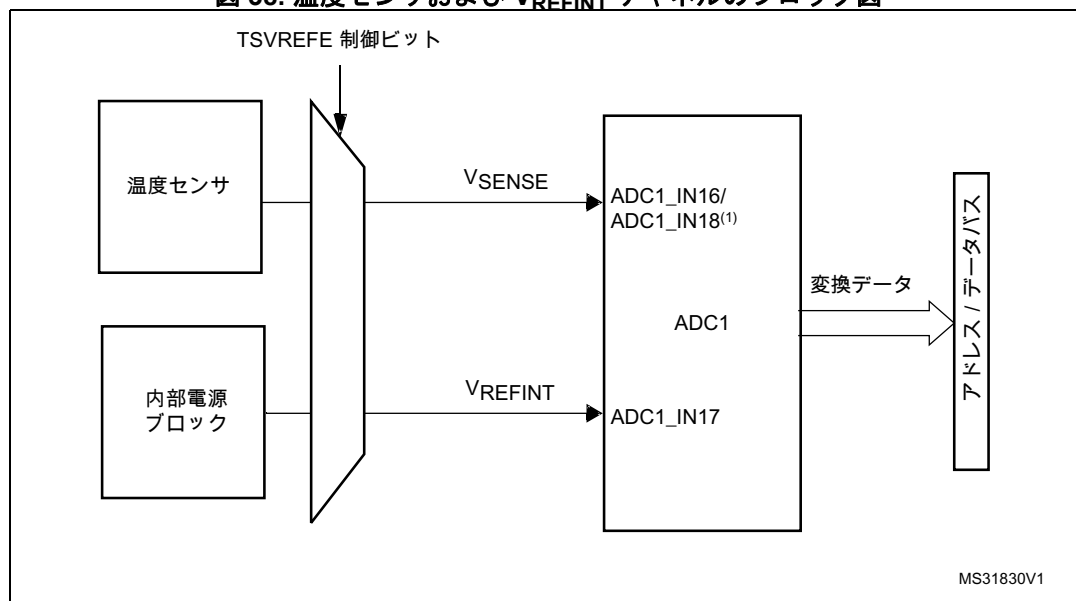
使用しないときには、センサをパワーダウンモードにすることができます。

注： 両方の内部チャンネル、すなわち、ADC1_IN16 または ADC1_IN18 (温度センサ)、および ADC1_IN17 (VREFINT) の変換を有効にするには、TSVREFE ビットをセットする必要があります。

主な特徴

- サポートしている温度範囲：-40 ~ 125 °C
- 精度：±1.5 °C

図 38. 温度センサおよび V_{REFINT} チャンネルのブロック図



1. V_{SENSE} が ADC1_IN18 に入力されます。

温度の読出し

温度センサを使用するには :

3. ADC1_IN16 または ADC1_IN18 入力チャネルを選択します。
4. データシートに指定されている最小サンプリング時間より長いサンプリング時間を選択します。
5. ADC_CCR レジスタの TSVREFE ビットをセットして、温度センサをパワーダウンモードからウェイクアップします。
6. SWSTART ビットをセットすることによって（または外部トリガによって）ADC 変換を開始します。
7. ADC データレジスタの V_{SENSE} データを読み出します。
8. 次の式を使用して、温度を計算します。

$$\text{温度 (摂氏)} = \{(V_{SENSE} - V_{25}) / \text{Avg_Slope}\} + 25$$

ここで、

- V_{25} = 25°C における V_{SENSE} 値
- Avg_Slope = 温度対 V_{SENSE} 曲線の平均傾斜 (mV/°C または $\mu\text{V}/^\circ\text{C}$)

V_{25} および Avg_Slope の実際の値については、データシートの電気的特性のセクションを参照してください。

注 : センサがパワーダウンモードからウェイクアップして、正しいレベルで V_{SENSE} を出力できるようになるまでには時間がかかります (スタートアップ時間)。ADC にも起動後のスタートアップ時間があるので、遅延を最小にするには、ADON ビットと TSVREFE ビットを同時にセットしてください。

温度センサの出力電圧は、温度に比例して変化します。この一次関数のオフセットは、プロセスのばらつきにより、チップごとに異なります (チップ間で最大 45 °C)。

内部温度センサは、絶対温度の代わりに温度変化を検出するアプリケーションに適しています。正確な計測温度が必要な場合は、外部温度センサを使用してください。

11.10 バッテリ残量監視

ADC_CCR レジスタの VBATE ビットはバッテリー電圧 (VBAT チャンネル) に切り替えるために使用します。VBAT 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、VBAT ピンはブリッジ分圧回路に内部接続されています。

VBATE をセットするとブリッジが自動的に有効になり以下を接続します。

- VBAT/4 を ADC1_IN18 入力チャネル

注 : VBAT および温度センサは同じ ADC 内部チャネル (ADC1_IN18) に接続されます。一度に 1 つの変換 (温度センサまたは VBAT のどちらか一方) を選択する必要があります。両方の変換を同時に有効すると、VBAT 変換のみ行われます。

11.11 ADC 割込み

アナログウォッチドッグステータスビットがセットされ、かつオーバーランステータスビットがセットされていれば、レギュラおよびインジェクトグループの変換終了時、割込みを生成することができます。高い柔軟性を実現するため、個別の割込みイネーブルビットを使用できます。

ADC_SR レジスタには他に 2 つのフラグがありますが、これらは割込みには無関係です。

- JSTRT (インジェクトグループチャネルの変換開始)
- STRT (レギュラグループチャネルの変換開始)

表 45. ADC 割込み

割込みイベント	イベントフラグ	有効制御ビット
レギュラグループの変換終了	EOC	EOCIE
インジェクトグループの変換終了	JEOC	JEOCIE
アナログウォッチドッグステータスビットのセット	AWD	AWDIE
オーバーラン	OVR	OVRIE

11.12 ADC レジスタ

レジスタの説明で使用されている略語のリストについては、[34 ページのセクション 1.1](#)を参照してください。

ペリフェラルレジスタには、ワードレベル (32 ビット) で書き込む必要があります。読出しアクセスは、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位で行うことができます。

11.12.1 ADC ステータスレジスタ (ADC_SR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み										OVR	STRT	JSTRT	JEOC	EOC	AWD
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **OVR** : オーバーラン

このビットは、データが失われたとき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。オーバーラン検出は DMA = 1 または EOCS = 1 のときのみ有効です。

0 : オーバーランは発生していません。
1 : オーバーランが発生しました。

ビット 4 **STRT** : レギュラチャネル開始フラグ

このビットは、レギュラチャネルの変換が開始したときにハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : レギュラチャネルの変換は開始されていません。
1 : レギュラチャネルの変換が開始されました。

ビット 3 **JSTRT** : インジェクトチャネル開始フラグ

このビットは、インジェクトグループの変換が開始すると、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : インジェクトグループの変換は開始されていません。
1 : インジェクトグループの変換が開始されました。

ビット 2 **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、グループ内のすべてのインジェクトチャネル変換が終了したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 変換が完了していません。
1 : 変換が完了しました。

ビット 1 **EOC** : レギュラチャネルの変換終了フラグ

このビットは、レギュラチャネルグループの変換が終了したときに、ハードウェアによってセットされます。ソフトウェアによって、または ADC_DR レジスタを読み出すことによってクリアされます。

0 : 変換が終了していない (EOCS=0)、または変換のシーケンスが終了していません (EOCS=1)。
1 : 変換が終了した (EOCS=0)、または変換のシーケンスが終了しました (EOCS=1)。

ビット 0 **AWD** : アナログウォッチドッグフラグ

このビットは、変換された電圧が、ADC_LTR および ADC_HTR レジスタでプログラミングされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : アナログウォッチドッグイベントは発生していません。
1 : アナログウォッチドッグイベントが発生しました。

11.12.2 ADC 制御レジスタ 1 (ADC_CR1)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み					OVR1E	RES		AWDEN	JAWDEN	予約済み					
					rw	rw	rw	rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **OVR1E** : オーバーラン割込みイネーブル

このビットは、オーバーラン割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : オーバーラン割込みは無効です。

1 : オーバーラン割込みは有効です。OVR ビットがセットされると、割込みが生成されます。

ビット 25:24 **RES[1:0]** : 分解能

これらのビットは、変換の分解能を選択するために、ソフトウェアによって書き込まれます。

00 : 12 ビット (15 ADCCLK サイクル)

01 : 10 ビット (13 ADCCLK サイクル)

10 : 8 ビット (11 ADCCLK サイクル)

11 : 6 ビット (9 ADCCLK サイクル)

ビット 23 **AWDEN** : アナログウォッチドッグ有効 (レギュラチャネル)

このビットは、ソフトウェアによってセット／クリアされます。

0 : レギュラチャネルに対するアナログウォッチドッグは無効です。

1 : レギュラチャネルに対するアナログウォッチドッグは有効です。

ビット 22 **JAWDEN** : アナログウォッチドッグ有効 (インジェクトチャネル)

このビットは、ソフトウェアによってセット／クリアされます。

0 : インジェクトチャネルに対するアナログウォッチドッグは無効です。

1 : インジェクトチャネルに対するアナログウォッチドッグは有効です。

ビット 21:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:13 **DISCNUM[2:0]** : 不連続モードチャネルカウント

これらのビットは、外部トリガの受信後に、不連続モードで変換されるレギュラチャネルの数を定義するために、ソフトウェアによって書き込まれます。

000 : 1 チャネル

001 : 2 チャネル

...

111 : 8 チャネル

ビット 12 **JDISCEN** : インジェクトチャネルの不連続モード

このビットは、グループのインジェクトチャネルに対する不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : インジェクトチャネルの不連続モードは無効です。

1 : インジェクトチャネルの不連続モードは有効です。

ビット 11 **DISCEN** : レギュラチャネルの不連続モード

このビットは、レギュラチャネルに対する不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : レギュラチャネルの不連続モードは無効です。

1 : レギュラチャネルの不連続モードは有効です。

ビット 10 JAUTO : 自動インジェクトグループ変換

このビットは、レギュラグループ変換の後、自動インジェクトグループ変換を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 自動インジェクトグループ変換は無効です。

1 : 自動インジェクトグループ変換は有効です。

ビット 9 AWDSGL : スキャンモードでの単一チャネルのウォッチドッグイネーブル

このビットは、AWDCH[4:0] ビットによって指定されたチャネルに対するアナログウォッチドッグを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : すべてのチャネルでアナログウォッチドッグは有効です。

1 : 単一チャネルでアナログウォッチドッグは有効です。

ビット 8 SCAN : スキャンモード

このビットは、スキャンモードを有効／無効にするために、ソフトウェアによってセット／クリアされます。スキャンモードでは、ADC_SQRx および ADC_JSQRx レジスタを通じて選択された入力に変換されます。

0 : スキャンモードは無効です。

1 : スキャンモードは有効です。

注 : EOCIE ビットがセットされている場合、EOC 割込みが生成されます。

— EOCs ビットが 0 にクリアされている場合、各レギュラグループシーケンス終了後

— EOCs ビットが 1 にセットされている場合、各レギュラチャネル変換終了後

注 : JEOCIE ビットがセットされている場合、最後のチャネルの変換が終了したときのみ、JEOC 割込みが生成されます。

ビット 7 JEOCIE : インジェクトチャネルの割込みイネーブル

このビットは、インジェクトチャネルの変換終了割込みを有効／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 割込みは無効です。

1 : JEOC 割込みは有効です。JEOC ビットがセットされると、割込みが生成されます。

ビット 6 AWDIE : アナログウォッチドッグ割込みイネーブル

このビットは、アナログウォッチドッグ割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : アナログウォッチドッグ割込みは無効です。

1 : アナログウォッチドッグ割込みは有効です。

ビット 5 EOCIE : EOC 割込みイネーブル

このビットは、変換終了時の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOC 割込みは無効です。

1 : EOC 割込みは有効です。EOC ビットがセットされると、割込みが生成されます。

ビット 4:0 AWDCH[4:0] : アナログウォッチドッグチャネル選択ビット

これらのビットは、ソフトウェアによってセット／クリアされます。アナログウォッチドッグによって保護される入力チャネルを選択します。

注 : 00000 : ADC アナログ入力チャネル0

00001 : ADC アナログ入力チャネル1

...

01111 : ADC アナログ入力チャネル15

10000 : ADC アナログ入力チャネル16

10001 : ADC アナログ入力チャネル17

10010 : ADC アナログ入力チャネル18

その他の値は予約済みです。

11.12.3 ADC 制御レジスタ 2 (ADC_CR2)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み	SWSTA RT	EXTEN		EXTSEL[3:0]				予約済み	JSWSTA RT	JEXTEN		JEXTSEL[3:0]			
	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				ALIGN	EOCS	DDS	DMA	予約済み						CONT	ADON
				rW	rW	rW	rW							rW	rW

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **SWSTART** : レギュラチャネルの変換開始

このビットは、変換を開始するためにソフトウェアによってセットされ、変換開始直後にハードウェアによってクリアされます。

0 : リセット状態

1 : レギュラチャネルの変換を開始します。

注 : このビットは **ADON = 1** のときのみセットできます。それ以外の場合変換は開始されません。

ビット 29:28 **EXTEN** : レギュラチャネルの外部トリガイネーブル

これらのビットは、外部トリガ極性を選択しレギュラグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00 : トリガ検出は無効です。

01 : 立ち上がりエッジでトリガを検出します。

10 : 立ち下がりエッジでトリガを検出します。

11 : 立ち上がりと下がりエッジの両方でトリガを検出します。

ビット 27:24 **EXTSEL[3:0]** : レギュラグループの外部イベント選択

これらのビットは、レギュラグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : タイマ 1 CC1 イベント

0001 : タイマ 1 CC2 イベント

0010 : タイマ 1 CC3 イベント

0011 : タイマ 2 CC2 イベント

0100 : タイマ 2 CC3 イベント

0101 : タイマ 2 CC4 イベント

0110 : タイマ 2 TRGO イベント

0111 : タイマ 3 CC1 イベント

1000 : タイマ 3 TRGO イベント

1001 : タイマ 4 CC4 イベント

1010 : タイマ 5 CC1 イベント

1011 : タイマ 5 CC2 イベント

1100 : タイマ 5 CC3 イベント

1101 : 予約済み

1110 : 予約済み

1111 : EXTI ライン 11

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 JSWSTART : インジェクトチャネルの変換開始

このビットは、ソフトウェアによってセットされ、変換開始直後にハードウェアによってクリアされます。

0 : リセット状態

1 : インジェクトチャネルの変換を開始します。

注 : このビットは **ADON = 1** のときのみセットできます。それ以外の場合変換は開始されません。

ビット 21:20 JEXTEN : インジェクトチャネルの外部トリガイネーブル

これらのビットは、外部トリガ極性を選択しインジェクトグループのトリガを有効にするために、ソフトウェアによってセット/クリアされます。

00 : トリガ検出は無効です。

01 : 立ち上がりエッジでトリガを検出します。

10 : 立ち下がりエッジでトリガを検出します。

11 : 立ち上がりと下がりエッジの両方でトリガを検出します。

ビット 19:16 JEXTSEL[3:0] : インジェクトグループの外部イベント選択

これらのビットは、インジェクトグループの変換の開始をトリガするために使用される外部イベントを選択します。

0000 : タイマ 1 CC4 イベント

0001 : タイマ 1 TRGO イベント

0010 : タイマ 2 CC1 イベント

0011 : タイマ 2 TRGO イベント

0100 : タイマ 3 CC2 イベント

0101 : タイマ 3 CC4 イベント

0110 : タイマ 4 CC1 イベント

0111 : タイマ 4 CC2 イベント

1000 : タイマ 4 CC3 イベント

1001 : タイマ 4 TRGO イベント

1010 : タイマ 5 CC4 イベント

1011 : タイマ 5 TRGO イベント

1100 : 予約済み

1101 : 予約済み

1110 : 予約済み

1111 : EXTI ライン 15

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 ALIGN : データの配置

このビットは、ソフトウェアによってセット/クリアされます。図 35 および 図 36 を参照してください。

0 : 右詰め

1 : 左詰め

ビット 10 EOCS : 変換選択終了

このビットは、ソフトウェアによってセット/クリアされます。

0 : EOC ビットはレギュラ変換シーケンスが終了するたびにセットされます。オーバーラン検出は DMA = 1 のときのみ有効です。

1 : EOC ビットはレギュラ変換が終了するたびにセットされます。オーバーラン検出は有効です。

ビット 9 DDS : DMA ディセーブル選択 (シングル ADC モード)

このビットは、ソフトウェアによってセット/クリアされます。

0 : 最後の転送後新たな DMA リクエストは発行されません (DMA コントローラの設定どおり)

1 : データが変換され DMA=1 である限り DMA リクエストは発行されます。

ビット 8 DMA : ダイレクトメモリアクセスモード (シングル ADC モード)

このビットは、ソフトウェアによってセット/クリアされます。詳細については、DMA コントローラの章を参照してください。

0 : DMA モードは無効です。

1 : DMA モードは有効です。

ビット 7:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CONT** : 連続変換

このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされた場合、それがクリアされるまで連続的に変換が行われます。

- 0 : シングル変換モード
- 1 : 連続変換モード

ビット 0 **ADON** : A/D コンバータのオン／オフ

このビットは、ソフトウェアによってセット／クリアされます。

- 注：
- 0 : ADC の変換を無効にして、パワーダウンモードに移行します。
 - 1 : ADC を有効にします。

11.12.4 ADC サンプル時間レジスタ 1 (ADC_SMPR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み					SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 : 27 予約済みであり、リセット値に保持する必要があります。

ビット 26:0 **SMPx[2:0]** : チャネル x サンプル時間選択

これらのビットは、各チャネルのサンプリング時間を個別に選択するために、ソフトウェアによって書き込まれます。サンプリングサイクル中は、チャネル選択ビットを変更することはできません。

- 注：
- 000 : 3 サイクル
 - 001 : 15 サイクル
 - 010 : 28 サイクル
 - 011 : 56 サイクル
 - 100 : 84 サイクル
 - 101 : 112 サイクル
 - 110 : 144 サイクル
 - 111 : 480 サイクル

11.12.5 ADC サンプル時間レジスタ 2 (ADC_SMPR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み		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]			SMP0[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 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:0 **SMPx[2:0]** : チャネル x サンプリング時間選択

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。

注 : **000** : 3 サイクル
 001 : 15 サイクル
 010 : 28 サイクル
 011 : 56 サイクル
 100 : 84 サイクル
 101 : 112 サイクル
 110 : 144 サイクル
 111 : 480 サイクル

11.12.6 ADC インジェクトチャネルデータオフセットレジスタ x (ADC_JOFRx) (x=1..4)

アドレスオフセット : 0x14~0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				JOFFSETx[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 **JOFFSETx[11:0]** : インジェクトチャネル x のデータオフセット

これらのビットは、インジェクトチャネルを変換するときに変換データから引かれるオフセットを定義するために、ソフトウェアによって書き込まれます。変換結果は、ADC_JDRx レジスタで読み出すことができます。

11.12.7 ADC ウォッチドッグ高閾値レジスタ (ADC_HTR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				HT[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 **HT[11:0]** : アナログウォッチドッグ高閾値

これらのビットは、アナログウォッチドッグの高閾値を定義するために、ソフトウェアによって書き込まれます。

11.12.8 ADC ウォッチドッグ低閾値レジスタ (ADC_LTR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

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

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **LT[11:0]** : アナログウォッチドッグ低閾値

これらのビットは、アナログウォッチドッグの低閾値を定義するために、ソフトウェアによって書き込まれます。

11.12.9 ADC レギュラシーケンスレジスタ 1 (ADC_SQR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み								L[3:0]				SQ16[4:1]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16_0	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				
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 23:20 **L[3:0]** : レギュラチャネルシーケンス長

これらのビットは、レギュラチャネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

0000 : 1 変換

0001 : 2 変換

...

1111 : 16 変換

ビット 19:15 **SQ16[4:0]** : レギュラシーケンスの 16 番目の変換

これらのビットには、変換シーケンス内の 16 番目として割り当てられたチャネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 14:10 **SQ15[4:0]** : レギュラシーケンスの 15 番目の変換

ビット 9:5 **SQ14[4:0]** : レギュラシーケンスの 14 番目の変換

ビット 4:0 **SQ13[4:0]** : レギュラシーケンスの 13 番目の変換

11.12.10 ADC レギュラシーケンスレジスタ 2 (ADC_SQR2)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10_0	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29:26 **SQ12[4:0]** : レギュラシーケンスの 12 番目の変換

これらのビットには、シーケンス内の 12 番目の変換として割り当てられたチャネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 24:20 **SQ11[4:0]** : レギュラシーケンスの 11 番目の変換

ビット 19:15 **SQ10[4:0]** : レギュラシーケンスの 10 番目の変換

ビット 14:10 **SQ9[4:0]** : レギュラシーケンスの 9 番目の変換

ビット 9:5 **SQ8[4:0]** : レギュラシーケンスの 8 番目の変換

ビット 4:0 **SQ7[4:0]** : レギュラシーケンスの 7 番目の変換

11.12.11 ADC レギュラシーケンスレジスタ 3 (ADC_SQR3)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4_0		SQ3[4:0]					SQ2[4:0]					SQ1[4:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:30 予約済みであり、リセット値に保持する必要があります。
- ビット 29:25 **SQ6[4:0]** : レギュラシーケンスの 6 番目の変換
これらのビットには、変換シーケンスで変換される 6 番目として割り当てられたチャネル番号 (0..18) がソフトウェアによって書き込まれます。
- ビット 24:20 **SQ5[4:0]** : レギュラシーケンスの 5 番目の変換
- ビット 19:15 **SQ4[4:0]** : レギュラシーケンスの 4 番目の変換
- ビット 14:10 **SQ3[4:0]** : レギュラシーケンスの 3 番目の変換
- ビット 9:5 **SQ2[4:0]** : レギュラシーケンスの 2 番目の変換
- ビット 4:0 **SQ1[4:0]** : レギュラシーケンスの 1 番目の変換

11.12.12 ADC インジェクトシーケンスレジスタ (ADC_JSQR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み										JL[1:0]		JSQ4[4:1]			
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]		JSQ3[4:0]				JSQ2[4:0]					JSQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:20 **JL[1:0]** : インジェクトシーケンス長

これらのビットは、インジェクトチャンネル変換シーケンスの合計変換数を定義するために、ソフトウェアによって書き込まれます。

00 : 1 変換
01 : 2 変換
10 : 3 変換
11 : 4 変換

ビット 19:15 **JSQ4[4:0]** : インジェクトシーケンスの 4 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

これらのビットには、変換シーケンス内の 4 番目として割り当てられたチャンネル番号 (0..18) がソフトウェアによって書き込まれます。

ビット 14:10 **JSQ3[4:0]** : インジェクトシーケンスの 3 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

ビット 9:5 **JSQ2[4:0]** : インジェクトシーケンスの 2 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

ビット 4:0 **JSQ1[4:0]** : インジェクトシーケンスの 1 番目の変換 (JL[1:0]=3 のとき。下記の注を参照)

注 : JL[1:0]=3 のとき (シーケンサには 4 つのインジェクト変換)、ADC はチャンネルを次の順に変換します。JSQ1[4:0]、JSQ2[4:0]、JSQ3[4:0]、JSQ4[4:0]。

JL=2 のとき (シーケンサには 3 つのインジェクト変換)、ADC はチャンネルを次の順に変換します。JSQ2[4:0]、JSQ3[4:0]、JSQ4[4:0]。

JL=1 のとき (シーケンサには 2 つのインジェクト変換)、ADC はチャンネルを JSQ3[4:0]、JSQ4[4:0] の順に変換します。

JL=0 のとき (シーケンサには 1 つのインジェクト変換)、ADC は JSQ4[4:0] チャンネルだけ変換します。

11.12.13 ADC インジェクトデータレジスタ x (ADC_JDRx) (x= 1..4)

アドレスオフセット : 0x3C~0x48

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
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 x の変換結果が格納されます。[図 35](#) および [図 36](#) 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。

11.12.14 ADC レギュラデータレジスタ (ADC_DR)

アドレスオフセット : 0x4C

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **DATA[15:0]** : レギュラデータ

これらのビットは読み出し専用です。これらのビットには、レギュラチャンネルの変換結果が格納されます。[図 35](#) および [図 36](#) 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。

11.12.15 ADC 共通制御レジスタ (ADC_CCR)

アドレスオフセット : 0x04 (このオフセットアドレスは ADC1 ベースアドレス + 0x300 に対する相対値です。)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み								TSVREFE	VBATE	予約済み				ADCPRE	
								rw	rw					rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み															

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **TSVREFE** : 温度センサと V_{REFINT} イネーブル

このビットは、温度センサおよび V_{REFINT} チャンネルを有効/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 : 温度センサおよび V_{REFINT} チャンネルは無効です。

1 : 温度センサおよび V_{REFINT} チャンネルは有効です。

注 : **TSVREFE** がセットされているときは **VBATE** を無効にする必要があります。両方のビットがセットされている場合、**VBAT** 変換のみ行われます。

ビット 22 **VBATE** : V_{BAT} イネーブル

このビットは、 V_{BAT} チャンネルを有効/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_{BAT} チャンネルは無効です。

1 : V_{BAT} チャンネルは有効です。

ビット 21:18 予約済みであり、リセット値に保持する必要があります。

ビット 17:16 **ADCPRE** : ADC プリスケール

ADC へのクロックの周波数を選択するために、ソフトウェアによってセット/クリアされます。..

注 : **00** : PCLK2 は 2 分周されます。

01 : PCLK2 は 4 分周されます。

10 : PCLK2 は 6 分周されます。

11 : PCLK2 は 8 分周されます。

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

11.12.16 ADC レジスタマップ

次の表は ADC レジスタの一覧です。

表 46. ADC グローバルレジスタマップ

オフセット	レジスタ
0x000~0x04C	ADC1
0x050~0x0FC	予約済み
0x100~0x14C	予約済み
0x118~0x1FC	予約済み
0x200~0x24C	予約済み
0x250~0x2FC	予約済み
0x300~0x308	共通レジスタ

表 47. ADC レジスタマップと各 ADC のリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
0x00	ADC_SR	予約済み																										OVR	STRT	JSTRT	JEOC	EOC	AWD								
	リセット値																											0	0	0	0	0	0								
0x04	ADC_CR1	予約済み						OVRIE	RES[1:0]		AWDEN	JAWDEN	予約済み				DISC NUM [2:0]		JDISCEN	DISCEN	JAUTO	AWD SGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]														
	リセット値							0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x08	ADC_CR2	予約 済み	SWSTART	EXTEN[1:0]		EXTSEL [3:0]				予約 済み	JSWSTART	JEXTEN[1:0]		JEXTSEL [3:0]			予約済み			ALIGN	EOCS	DDS	DMA	予約済み				CONT	ADON												
	リセット値		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0		0						0	0											
0x0C	ADC_SMPR1	サンプル時間ビット SMPx_x																																							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x10	ADC_SMPR2	サンプル時間ビット SMPx_x																																							
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x14	ADC_JOFR1	予約済み															JOFFSET1[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			
0x18	ADC_JOFR2	予約済み															JOFFSET2[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		
0x1C	ADC_JOFR3	予約済み															JOFFSET3[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	
0x20	ADC_JOFR4	予約済み															JOFFSET4[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
0x24	ADC_HTR	予約済み															HT[11:0]																								
	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
0x28	ADC_LTR	予約済み															LT[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
0x2C	ADC_SQR1	予約済み								L[3:0]			レギュラチャネルシーケンス SQx_x ビット																												
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x30	ADC_SQR2	予約 済み	レギュラチャネルシーケンス SQx_x ビット																																						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x34	ADC_SQR3	予約 済み	レギュラチャネルシーケンス SQx_x ビット																																						
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x38	ADC_JSQR	予約済み									JL[1:0]		インジェクトチャネルシーケンス JSQx_x ビット																												
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					

表 47. ADC レジスタマップと各 ADC のリセット値 (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
0x3C	ADC_JDR1	予約済み																JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	ADC_JDR2	予約済み																JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	ADC_JDR3	予約済み																JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	ADC_JDR4	予約済み																JDATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	ADC_DR	予約済み																レギュラ DATA[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 48. ADC レジスタマップとリセット値 (共通 ADC レジスタ)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x04	ADC_CCR	予約済み								TSVREFE	VBATE	予約済み					ADCPRE[1:0]		予約済み														
	リセット値									0	0						0 0																

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

12 高機能制御タイマ (TIM1)

TIM8 は STM32F401xB/C および STM32F401xD/E では使用できません。

12.1 TIM1 の概要

高機能制御タイマ (TIM1) は、プログラム可能なプリスケアラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

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、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、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。

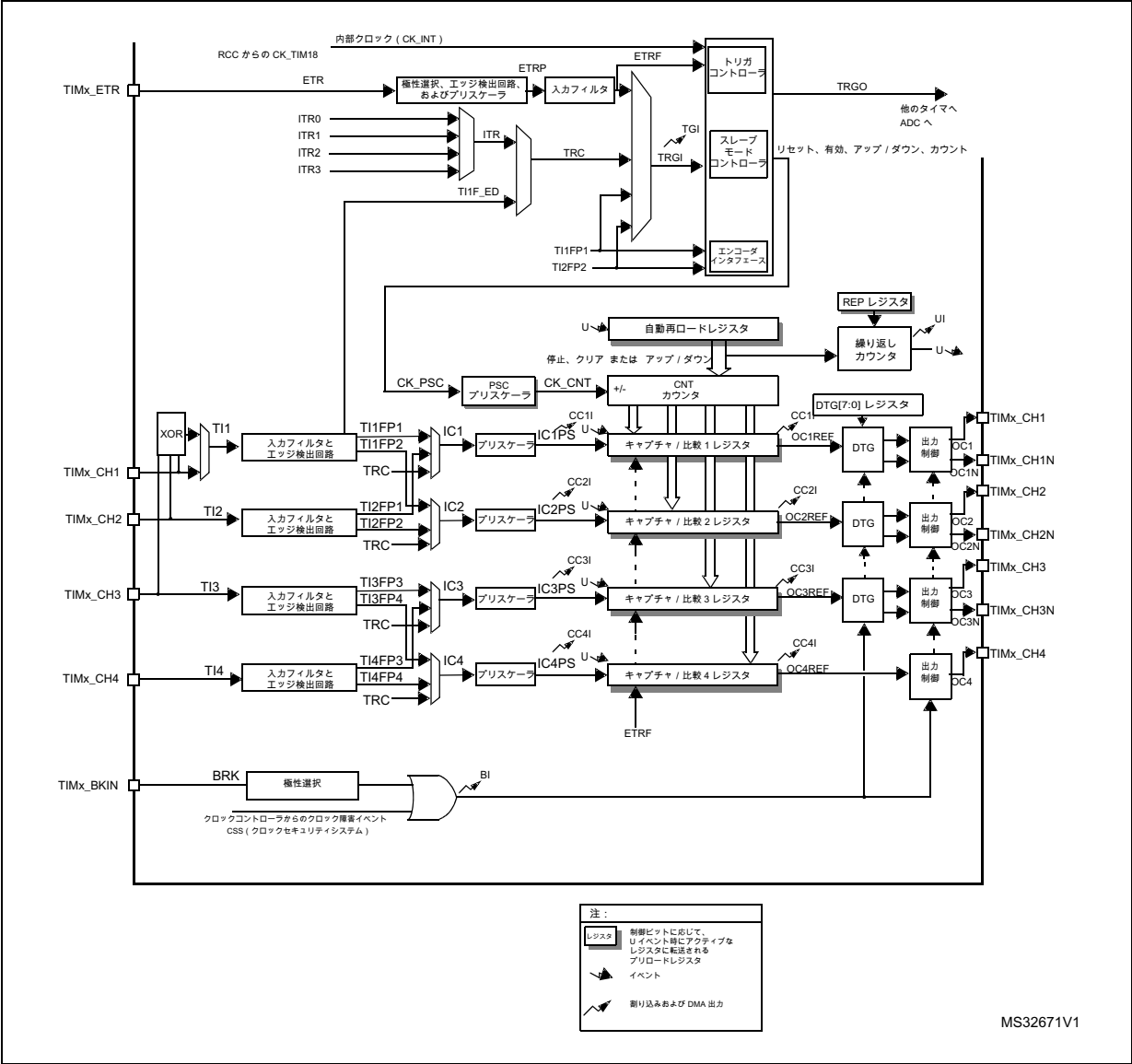
高機能制御タイマ (TIM1) と汎用タイマ (TIMx) は、互いに独立しており、リソースを共有しません。これらのタイマは、[セクション 12.3.20](#) に示すように、相互に同期させることができます。

12.2 TIM1 の主な機能

TIM1 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アラ。カウンタクロック周波数を 1 から 65536 の間で分周でき、分周比の動作中の変更も可能。
- 次の機能を持つ、最大 4 つの独立チャネル。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成 (エッジアラインモードとセンターアラインモード)
 - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレーク入力。
- 以下のイベント時の割込み/DMA 生成。
 - 更新：カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
 - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
 - 入力キャプチャ
 - 出力比較
 - ブレーク入力
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 39. 高機能制御タイマのブロック図



12.3 TIM1 の機能詳細

12.3.1 タイムベースユニット

プログラム可能な高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)
- 繰り返しカウンタレジスタ (TIMx_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

図 40 と 図 41 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 40. 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

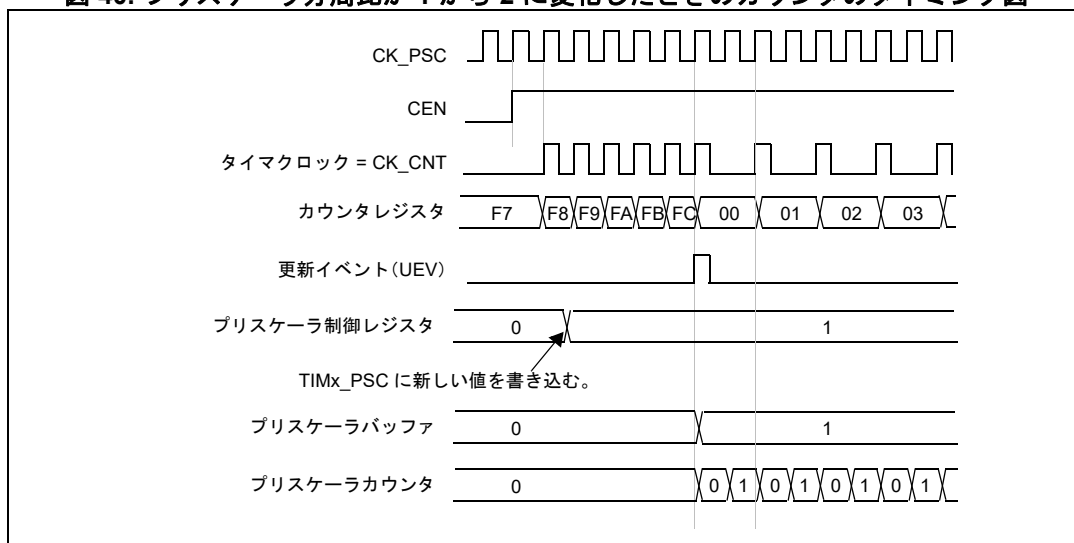
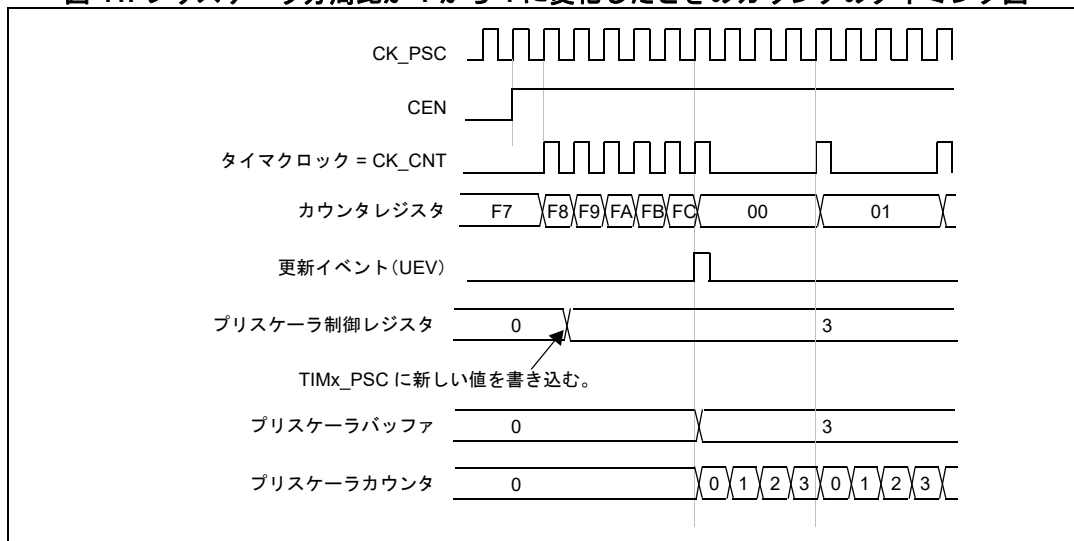


図 41. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



12.3.2 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) までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします（ただし、プリスケアラ比は変化しません）。さらに、TIMx_CR1 レジスタの URS ビット（更新リクエスト選択）がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません（したがって、割り込みや DMA リクエストは送信されません）。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 42. 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

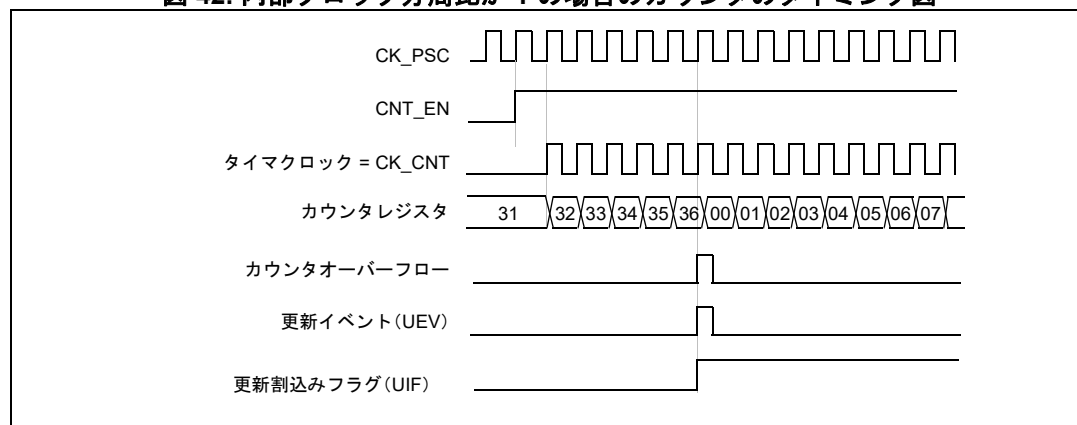


図 43. 内部クロック分周比が 2 の場合のカウンタのタイミング図

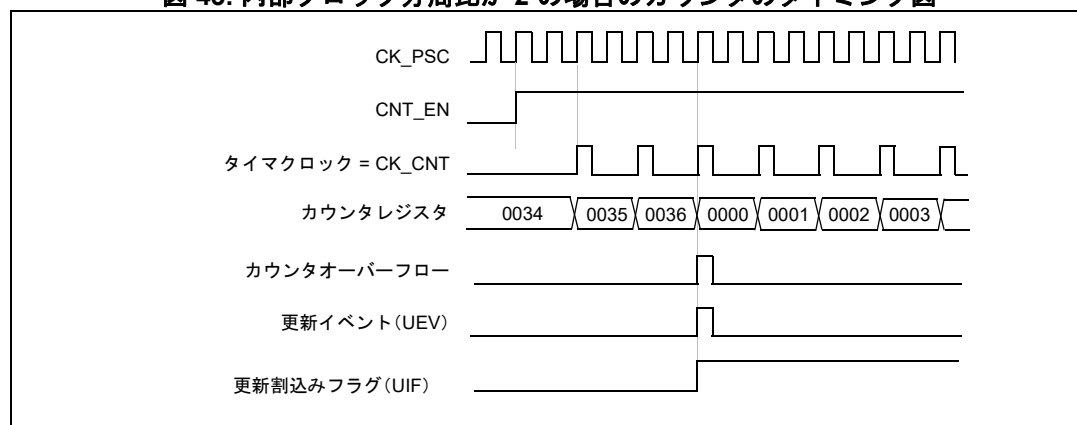


図 44. 内部クロック分周比が 4 の場合のカウンタのタイミング図

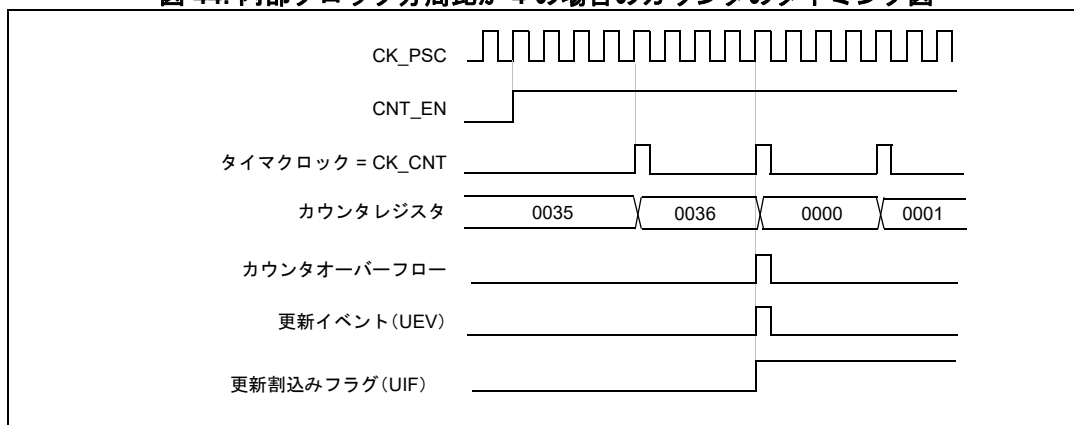


図 45. 内部クロック分周比が N の場合のカウンタのタイミング図

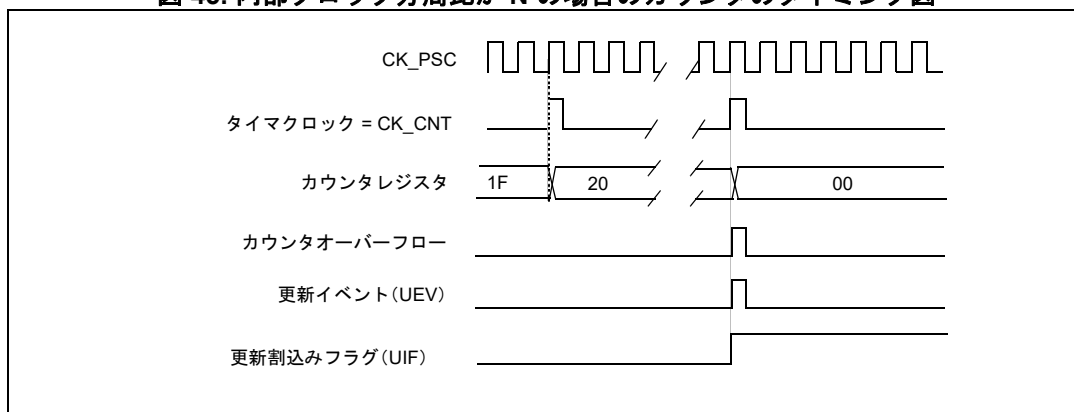


図 46. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

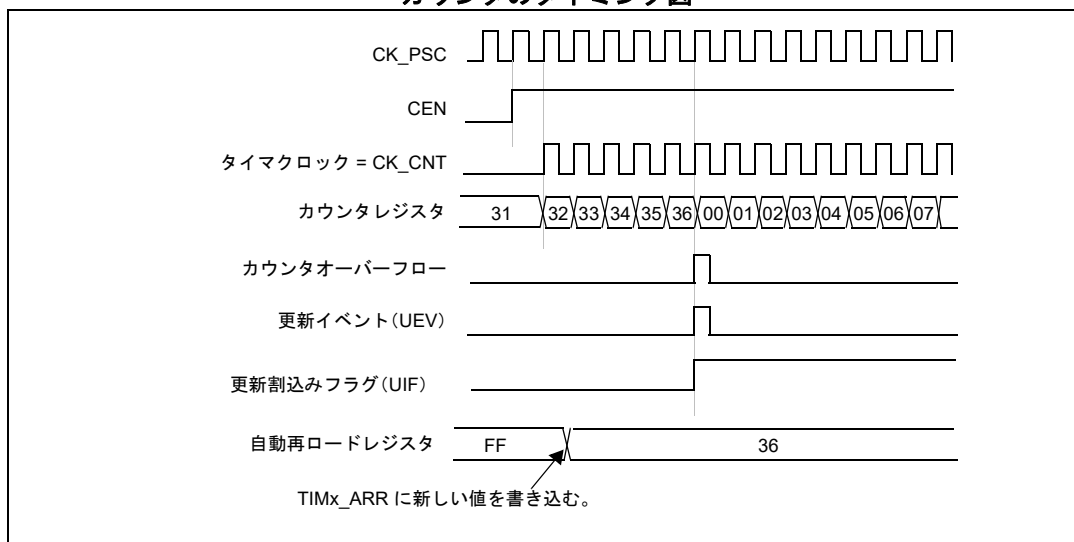
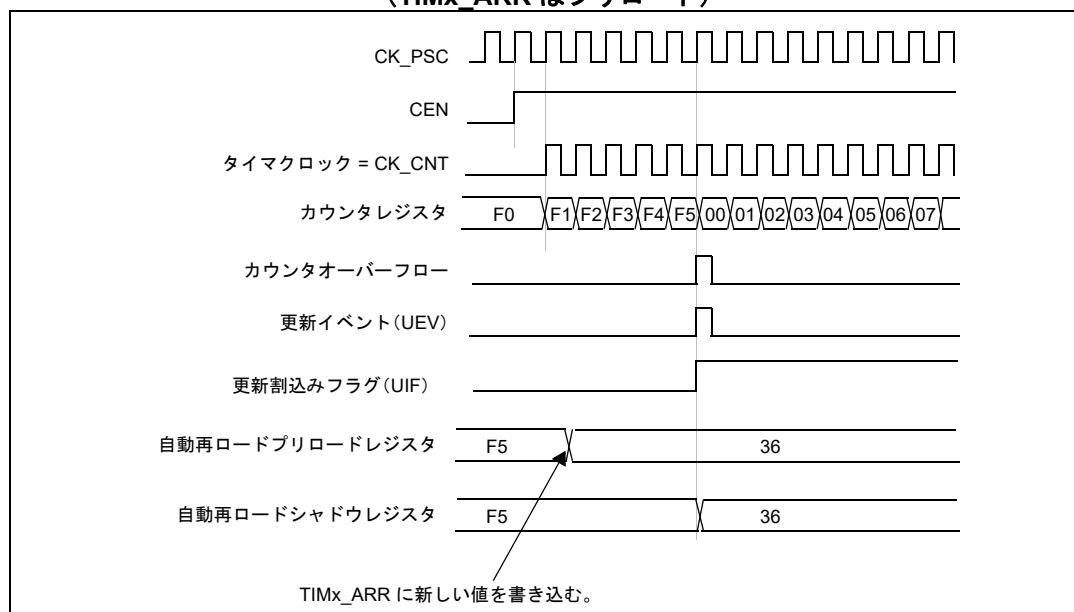


図 47. ARPE=1 の場合の更新イベント時のカウンタのタイミング図
(TIMx_ARR はプリロード)



ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタに設定されている回数プラス 1 (TIMx_RCR+1) までダウンカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのアンダーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや 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込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタが再ロードされる前に自動再ロードが更新されるので、次の周期は予想通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 48. 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

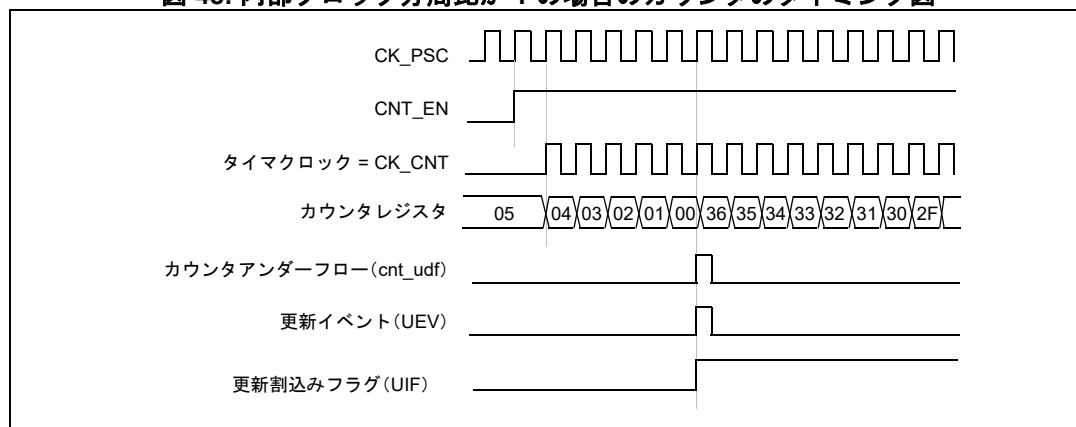


図 49. 内部クロック分周比が 2 の場合のカウンタのタイミング図

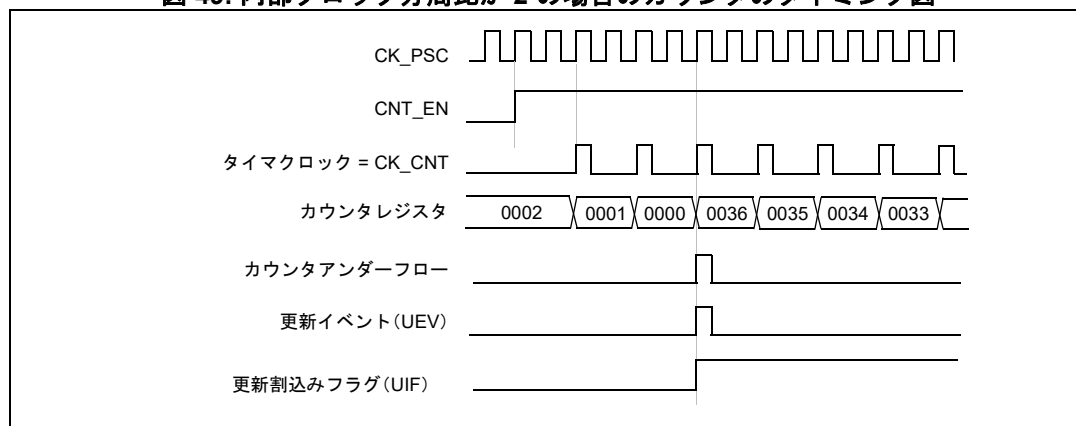


図 50. 内部クロック分周比が 4 の場合のカウンタのタイミング図

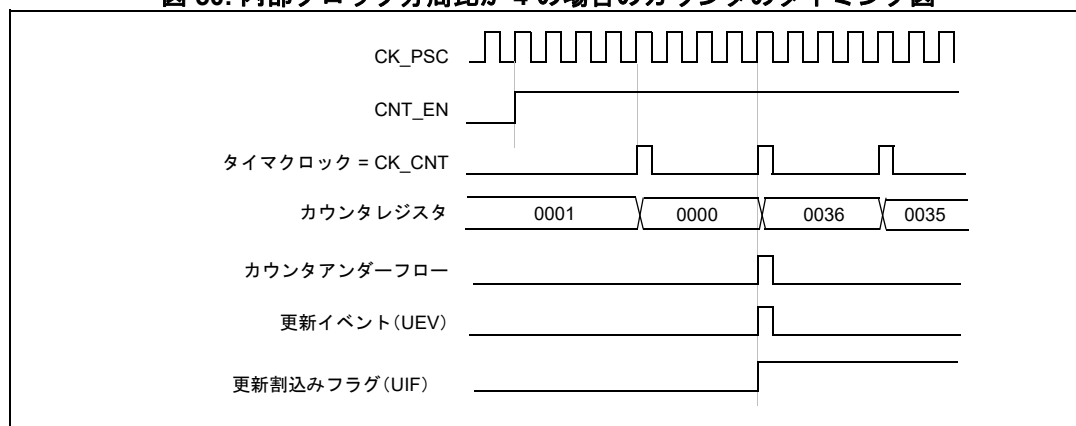


図 51. 内部クロック分周比が N の場合のカウンタのタイミング図

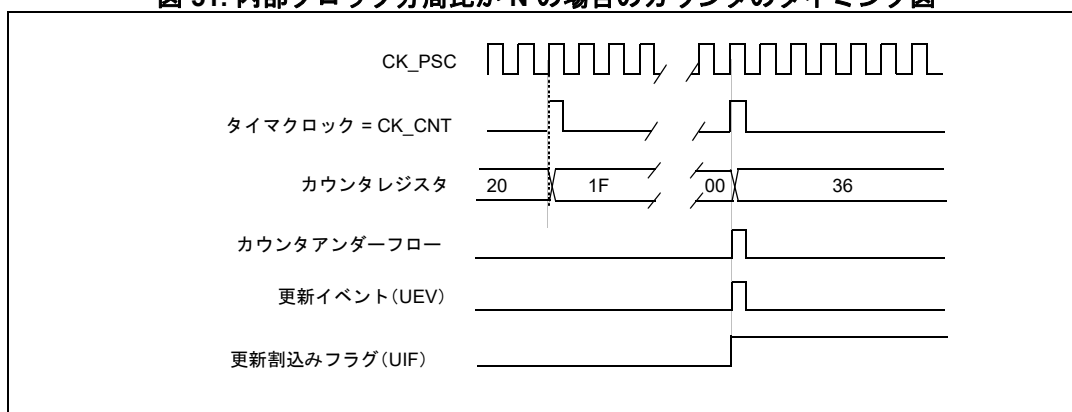
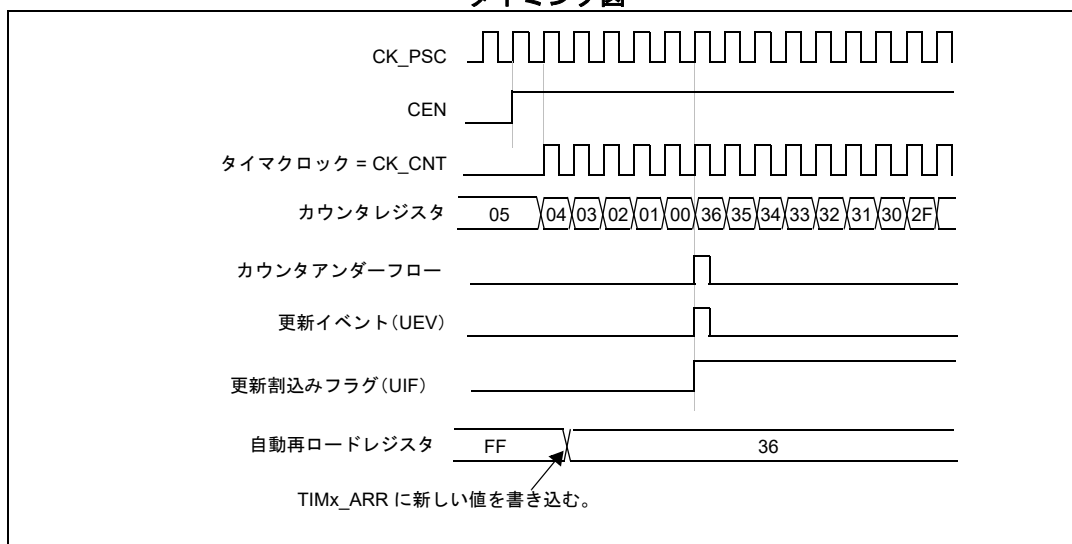


図 52. 繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図



センターアラインモード (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。

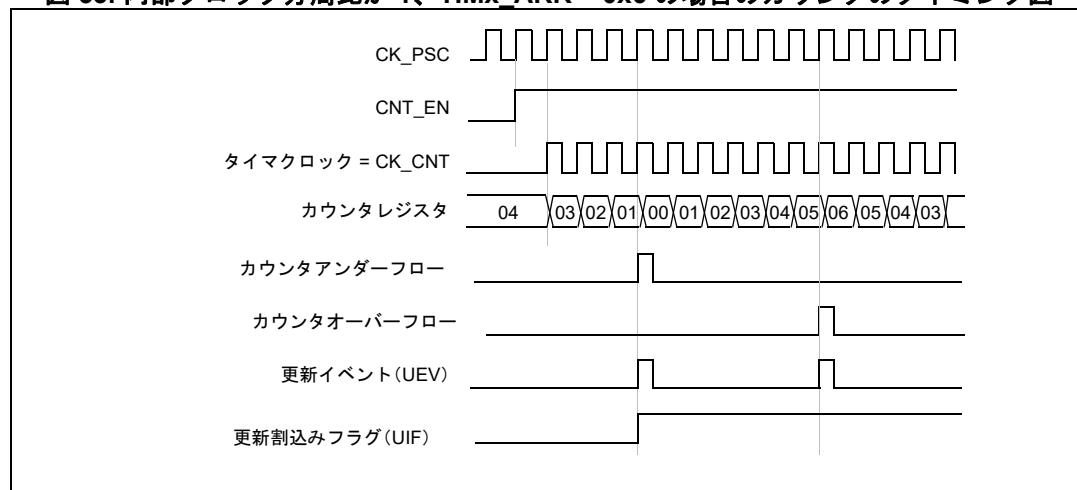
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合、UG ビットをセットすると UEV 更新イベントが生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

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 53. 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 12.4:283 ページのTIM1 レジスタ](#)を参照)。

図 54. 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

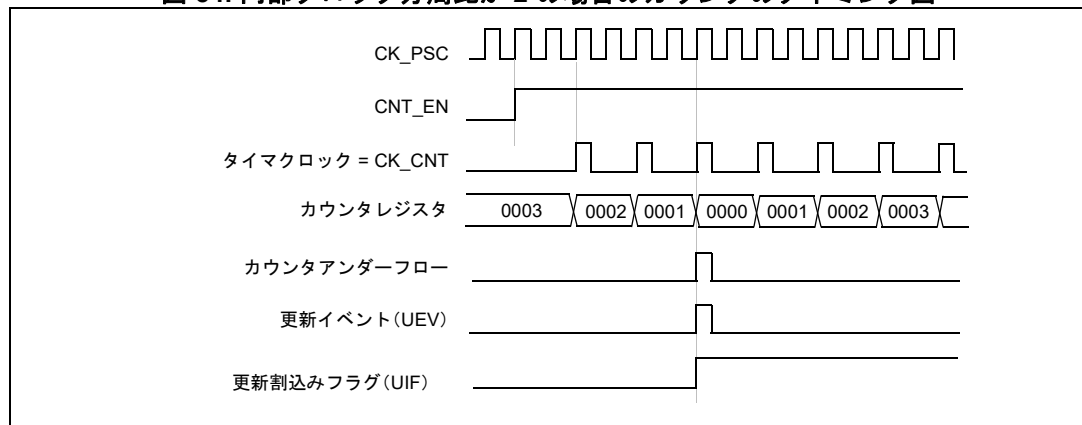
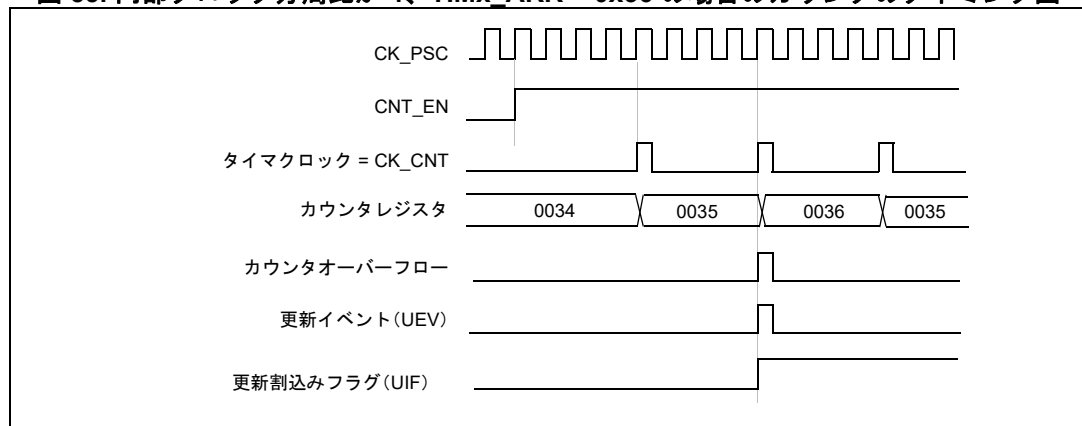


図 55. 内部クロック分周比が 4、TIMx_ARR = 0x36 の場合のカウンタのタイミング図



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 56. 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

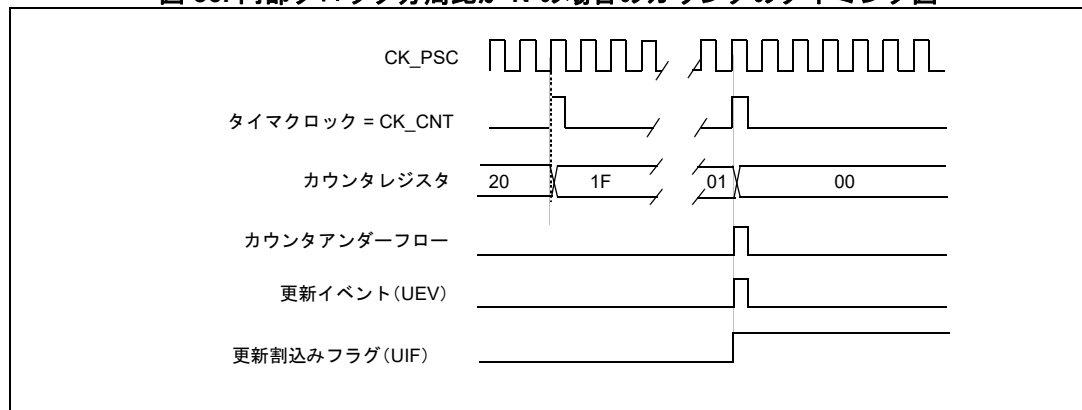


図 57. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図

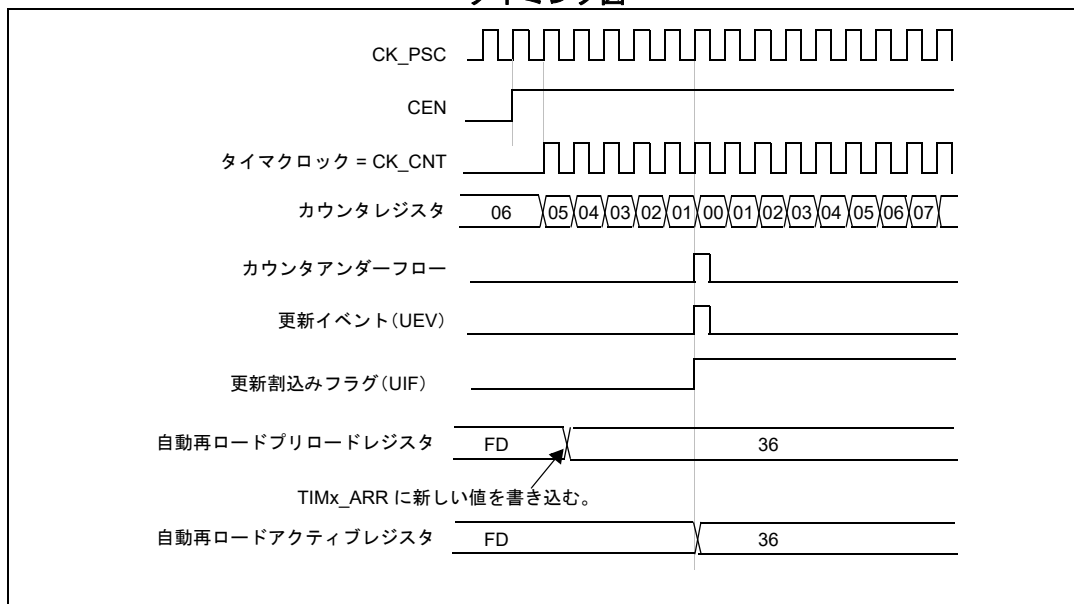
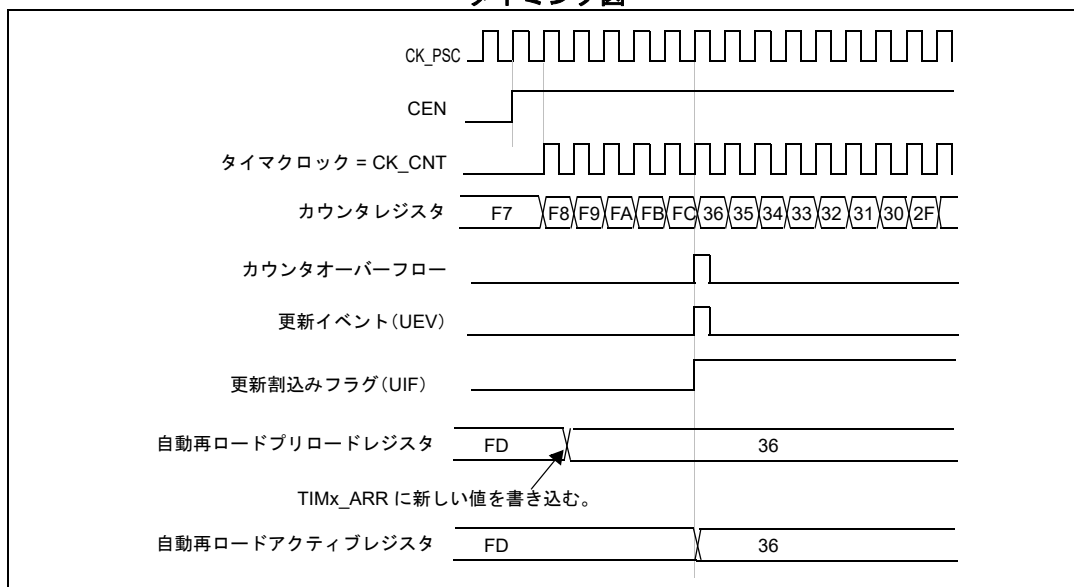


図 58. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



12.3.3 繰り返しカウンタ

セクション 12.3.1: 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。

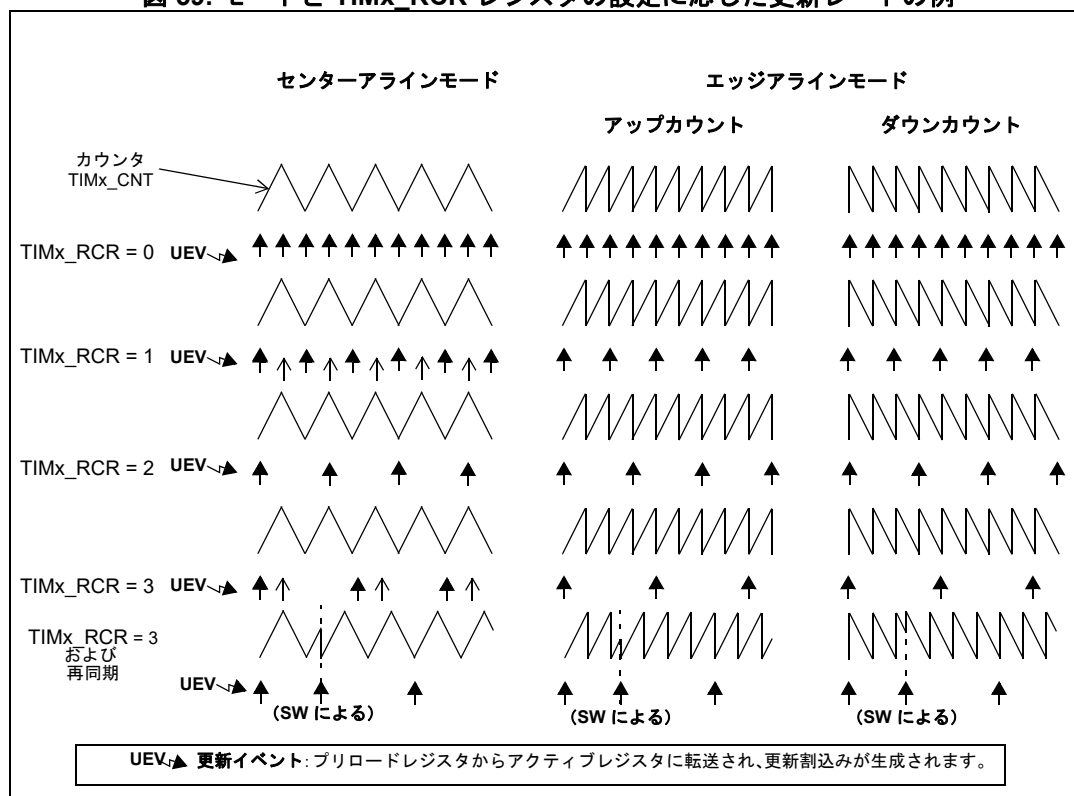
繰り返しカウンタは、次の場合にデクリメントします。

- 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 128 PWM サイクルに限られますが、PWM 周期ごとにデューティサイクルを 2 回更新することが可能になります。センターアラインモードで比較レジスタの値を PWM 周期あたり 1 回のみ更新するときには、パターンが対称なので、最大精度は $2 \times T_{ck}$ 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 59 を参照してください)。ソフトウェアによって (TIMx_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが発生されると、繰り返しカウンタの値にかかわらず直ちにイベントが発生し、繰り返しカウンタに TIMx_RCR レジスタの内容が再ロードされます。

センターアラインモードでは、TIMx_RCR が奇数の場合、TIMx_RCR レジスタが書き込まれたタイミングおよびカウンタが開始されたタイミングに応じてオーバーフローまたはアンダーフロー時に更新イベントが発生します。カウンタの開始前に TIMx_RCR が書き込まれた場合は、オーバーフローで、UEV が発生します。カウンタの開始後に TIMx_RCR が書き込まれた場合は、アンダーフローで UEV が発生します。たとえば、TIMx_RCR = 3 の場合、TIMx_RCR の書き込みタイミングに応じて 4 回目のオーバーフローイベントまたはアンダーフローイベントごとに UEV が発生します。

図 59. モードと TIMx_RCR レジスタの設定に応じた更新レートの例



12.3.4 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

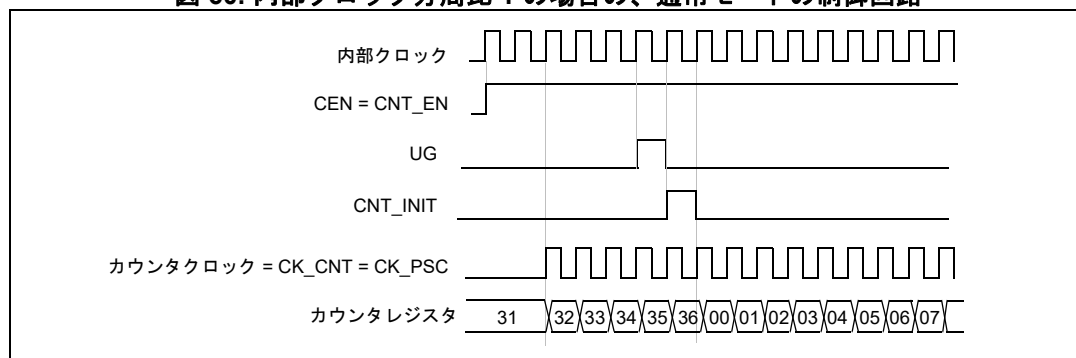
- 内部クロック (CK_INT)
- 外部クロックモード 1: 外部入力ピン
- 外部クロックモード 2: 外部トリガ入力 ETR
- 内部トリガ入力 (ITRx): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 1 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する](#)を参照してください。

内部クロックソース (CK_INT)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx_CR1 レジスタ)、および UG ビット (TIMx_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INTが供給されます。

図 60 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。

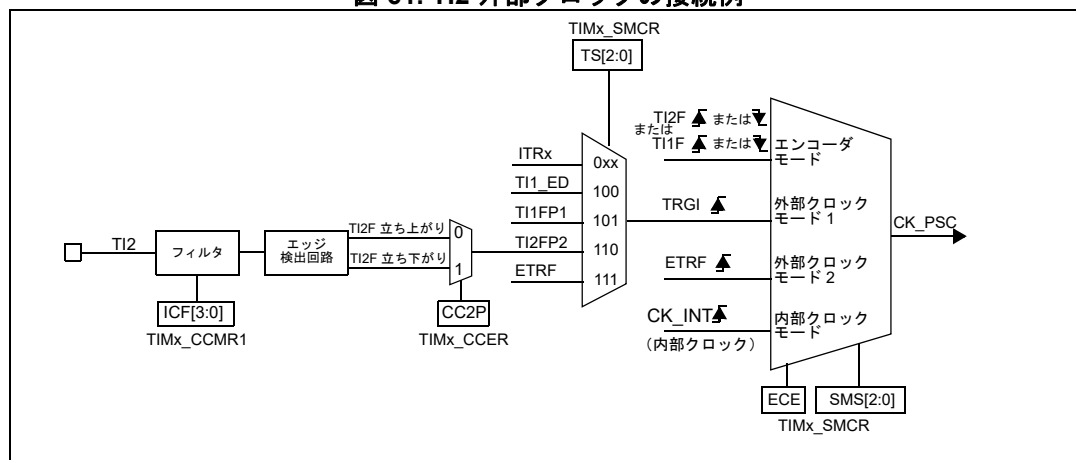
図 60. 内部クロック分周比 1 の場合の、通常モードの制御回路



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 61. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

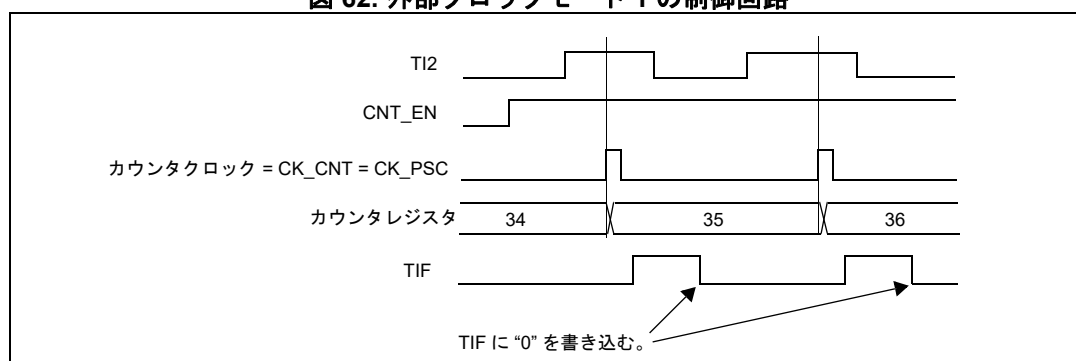
1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。
3. CC2P=0 と CC2NP=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

注： キャプチャプリスケアラはトリガには使用されないで、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 62. 外部クロックモード 1 の制御回路



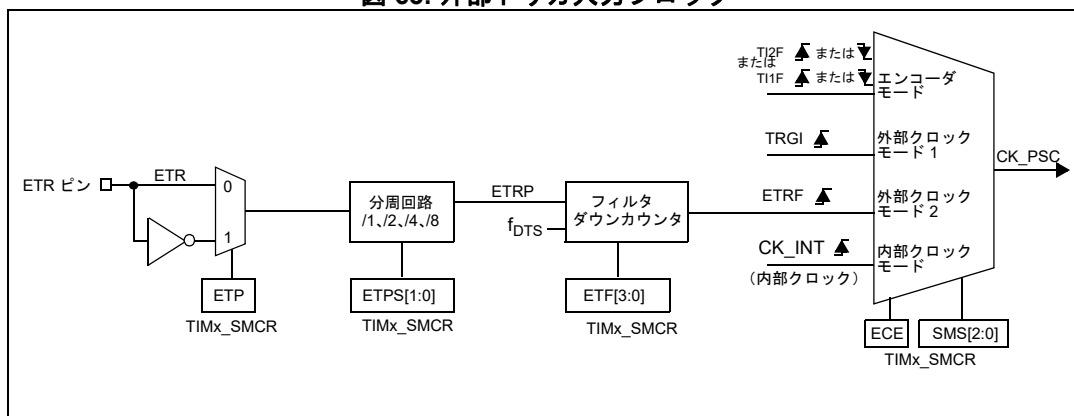
外部クロックソースモード 2

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 63 に、外部トリガ入力ブロックの概要を示します。

図 63. 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 64. 外部クロックモード 2 の制御回路

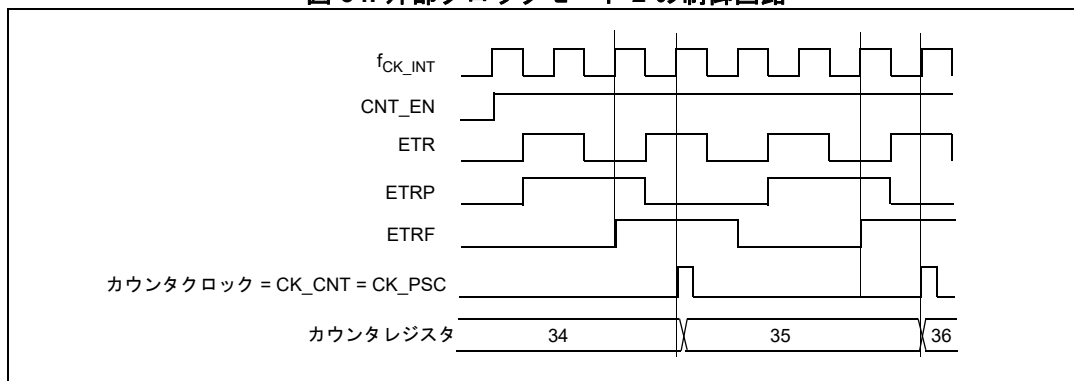


図 67. キャプチャ／比較チャンネル (チャンネル 1 から 3) の出力ステージ

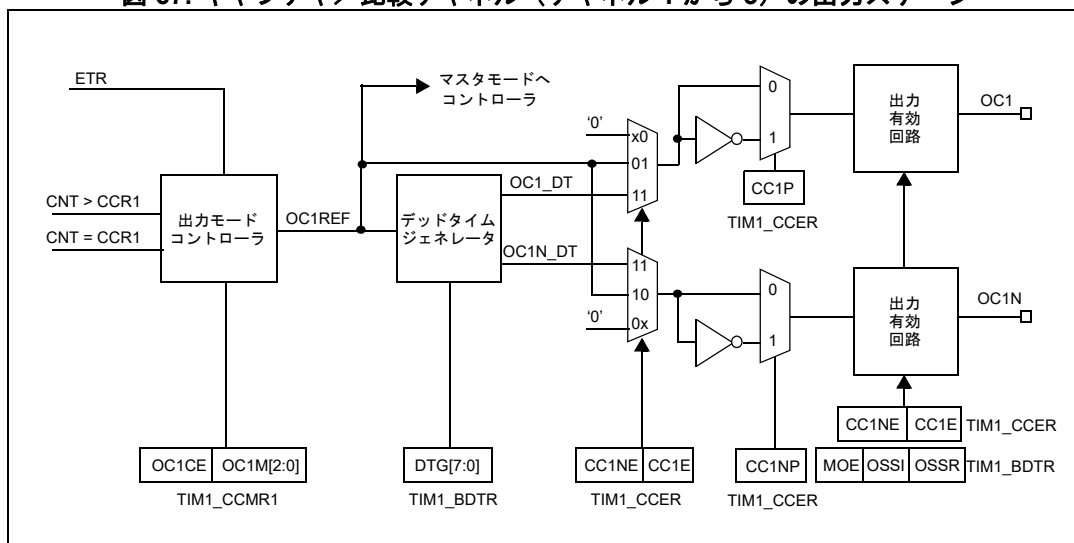
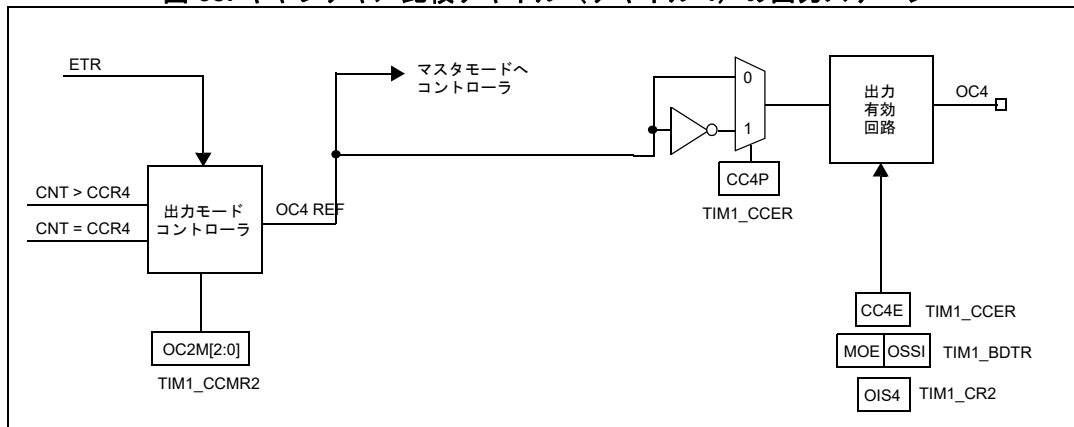


図 68. キャプチャ／比較チャンネル (チャンネル 4) の出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書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。

12.3.6 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバーキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力が入立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならないので、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 レジスタは読出し専用になります。
- タイマに接続する信号に対して必要とする入力フィルタ時間を設定します (入力が TIx 入力の内の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 の遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ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。

オーバーキャプチャを処理するために、オーバー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、フラグ読出し後、データ読出し前に発生するオーバーキャプチャの見落としを避けることができます。

注： IC 割り込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

12.3.7 PWM 入力モード

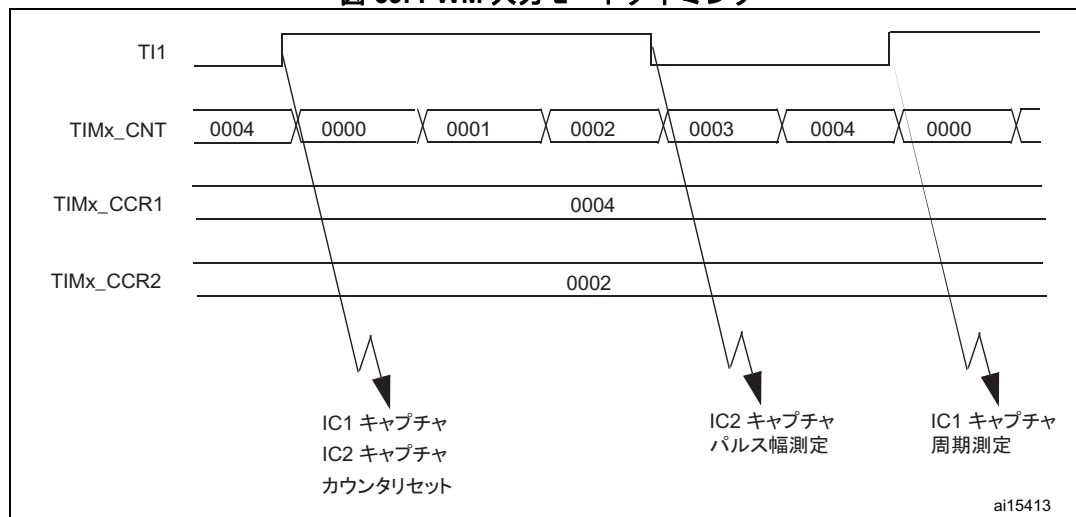
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケール値によって、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“1”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 69. PWM 入力モードタイミング



12.3.8 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット =00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルとするには、対応する TIMx_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。

12.3.9 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割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、割込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

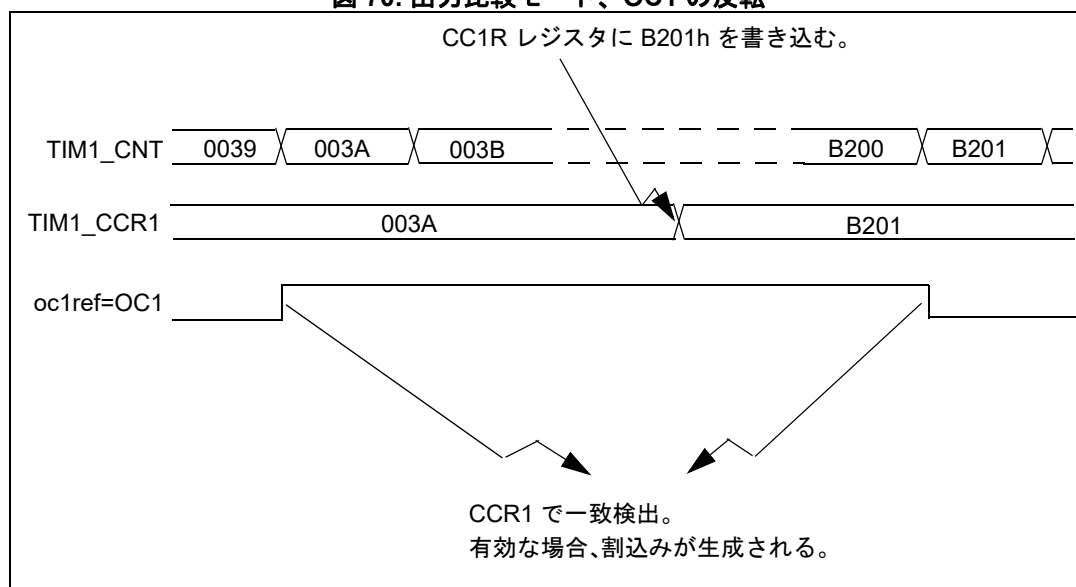
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは、単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順：

1. カウンタクロックを選択します（内部、外部、プリスケアラ）。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割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 70 に示します。

図 70. 出力比較モード、OC1 の反転



12.3.10 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx_CCER および TIMx_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx か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。

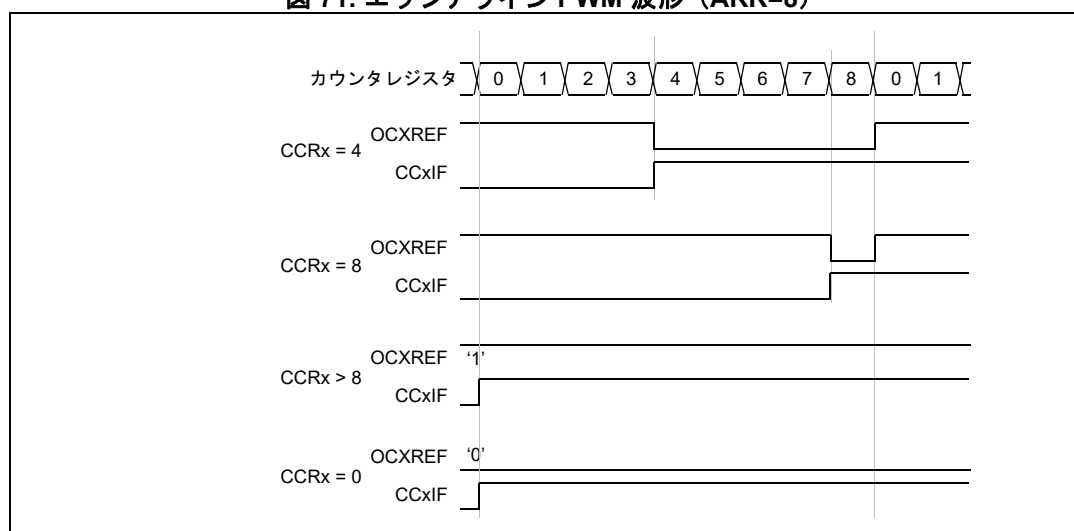
PWM エッジアラインモード

- 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。[セクション : 245 ページのアップカウントモード](#)を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 71](#) 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 71. 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。[セクション : 248 ページのダウンカウントモード](#)を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

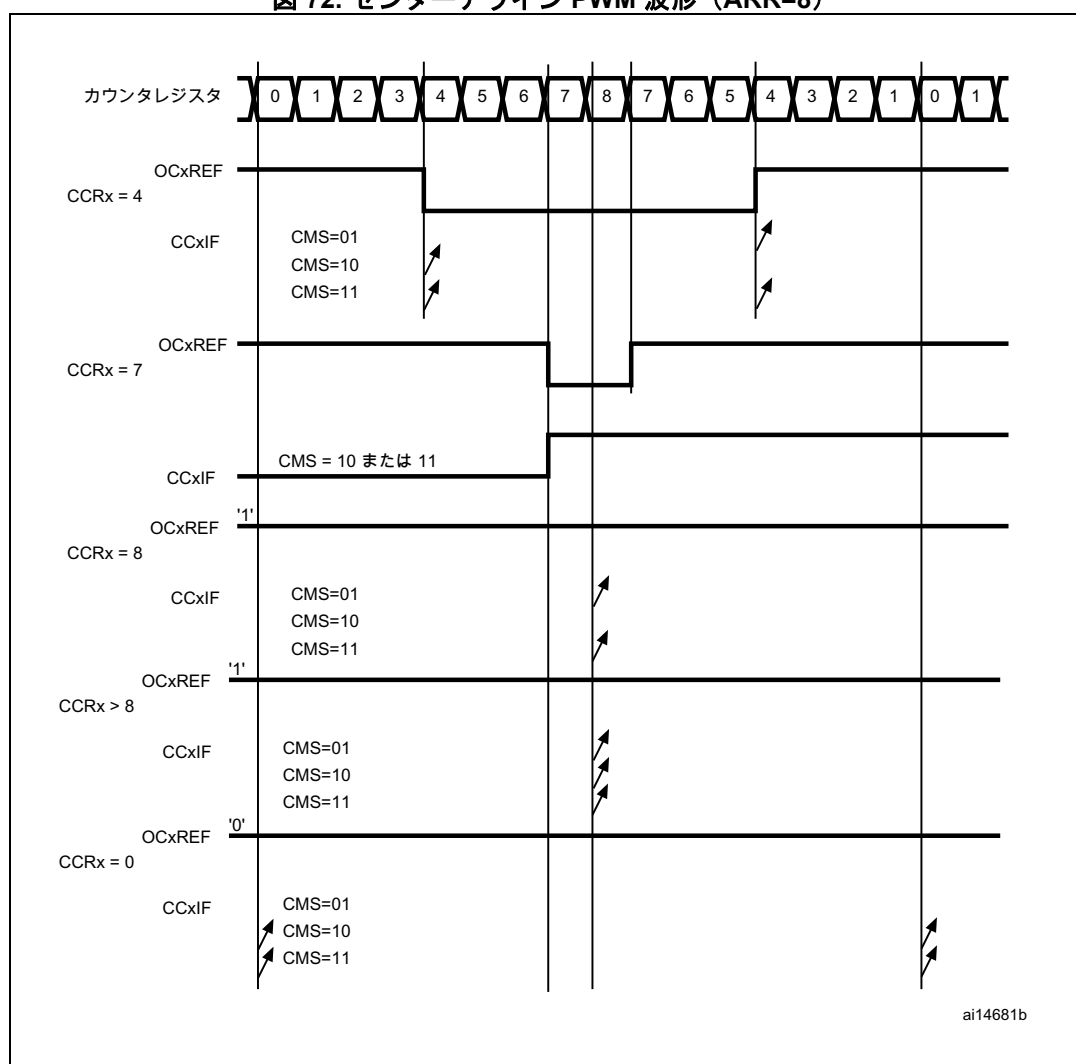
PWM センターアラインモード

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです（その他すべての構成は、OCxRef/OCx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット（DIR）はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[セクション : 250 ページの センターアラインモード（アップ/ダウンカウント）](#)を参照してください。

図 72 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 72. 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。

12.3.11 相補出力とデッドタイム挿入

高機能制御タイマ (TIM1) 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 $TIMxCR2$ レジスタの MOE、OISx、OISxN、OSS1、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 51:300 ページの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](#) を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるときに)、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。 $TIMx_BDTR$ レジスタの DTG[7:0] ビットは、全チャンネルに対するデッドタイムの生成を制御するために使います。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点異なります。
- 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 73. デッドタイム挿入のある相補出力

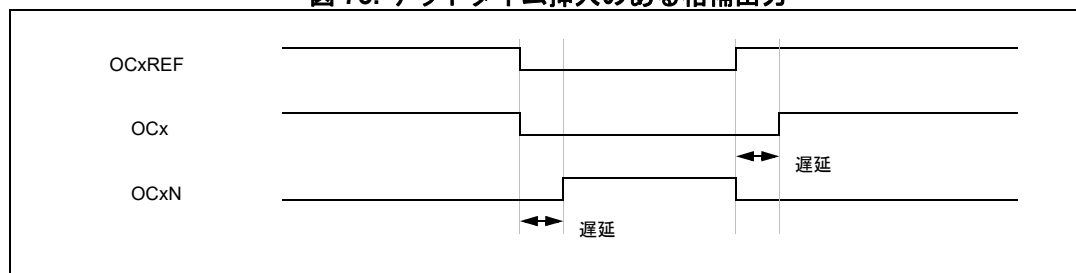


図 74. 負のパルスより長い遅延があるときのデッドタイムの波形

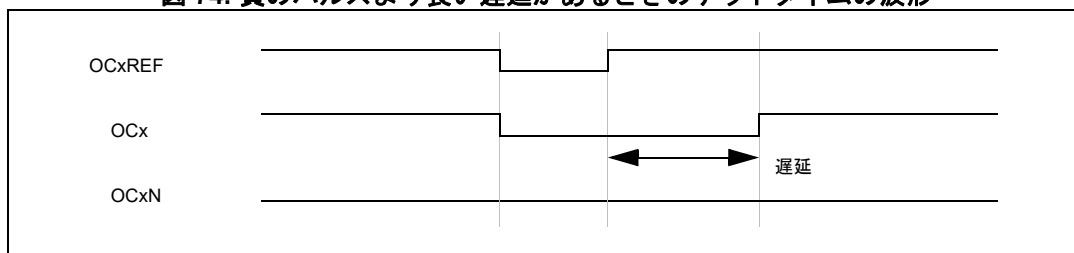
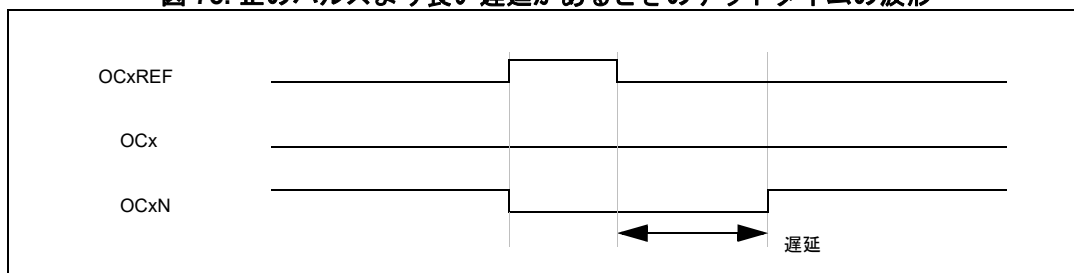


図 75. 正のパルスより長い遅延があるときのデッドタイムの波形



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 ビットでプログラム可能です。遅延計算については、[セクション 12.4.18 : 304 ページのTIM1 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。

12.3.12 ブレーク機能の使用

ブレーク機能を使用しているときには、出力イネーブル信号とインアクティブレベルは追加の制御ビット (TIMx_BDTR レジスタの MOE、OSSI、および OSSR ビットと TIMx_CR2 レジスタの OISx および OISxN ビット) に応じて変更されます。ただし、OCx および OCxN 出力の両方を同時にアクティブレベルに設定することはできません。詳細については、[表 51: 300 ページの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](#) を参照してください。

ブレークソースは、ブレーク入力ピンか、リセットクロックコントローラ (RCC) のクロックセキュリティシステム (CSS) によって生成されたクロック障害イベントです。クロックセキュリティシステムの詳細については、[セクション 6.2.7: クロックセキュリティシステム \(CSS\)](#) を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。TIMx_BDTR レジスタの BKE ビットをセットすることによって、ブレーク機能を有効にできます。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。BKE および BKP ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がリエッジは非同期のことがあるので、実際の信号 (出力に作用する信号) と同期制御ビット (TIMx_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延 (ダミー命令) を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

ブレークが発生すると (ブレーク入力で選択されたレベル)、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態、アイドル状態、またはリセット状態 (OSSI ビットで選択) になります。これは、MCU オシレータがオフの場合も同様です。
- 各出力チャンネルは、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、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには：
 - 出力は、まずリセット状態のインアクティブ状態に置かれます (極性に依存します)。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
 - タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください (約 2 CK_TIM クロックサイクル)。
 - OSSI=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はそのままか、CCxE または CCxNE ビットがハイになったときにハイになります。
- ブレーク状態フラグ (TIMx_SR レジスタの BIF ビット) がセットされます。TIMx_DIER レジスタの BIE ビットがセットされている場合は、割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 をクリアできません。

ブレークは、BRK 入力によって生成でき、BRK はプログラミング可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。

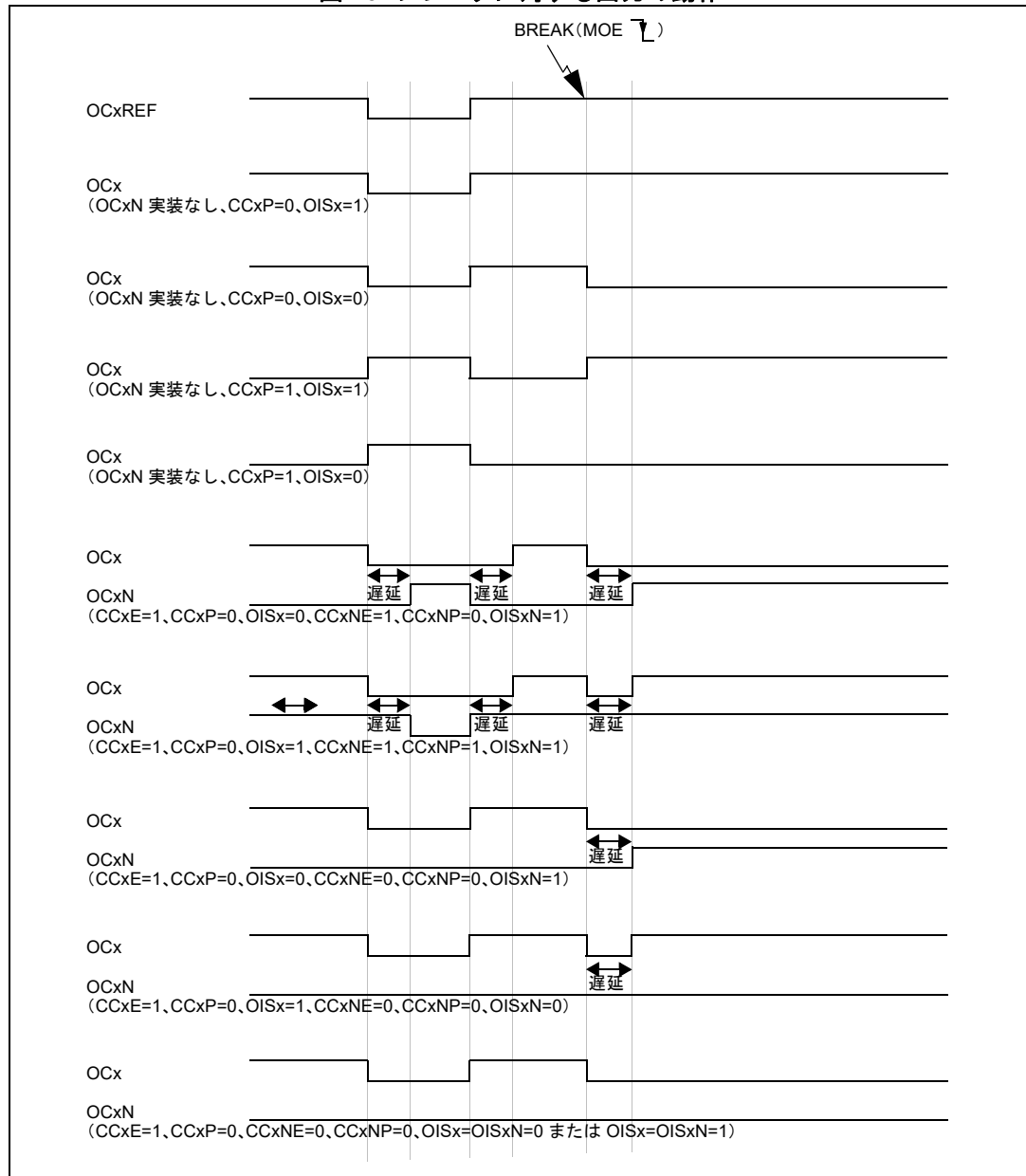
ブレークの生成には次の2つの方法があります。

- BRK 入力を使用します。BRK はプログラム可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。
- ソフトウェアから生成します。TIMx_EGR レジスタで BG ビットをセットします。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ（デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性）を固定することができます。TIMx_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 12.4.18 : 304 ページのTIM1 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。

[図 76](#) に、ブレークに対する出力の動作例を示します。

図 76. ブレークに対する出力の動作



12.3.13 外部イベントによる OCxREF 信号のクリア

特定のチャネルの OCxREF 信号は、ETRF 入力にハイレベルを適用する（対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする）ことによってローに駆動できます。OCxREF 信号は、次の更新イベント UEV が発生するまで、ローレベルを保ちます。

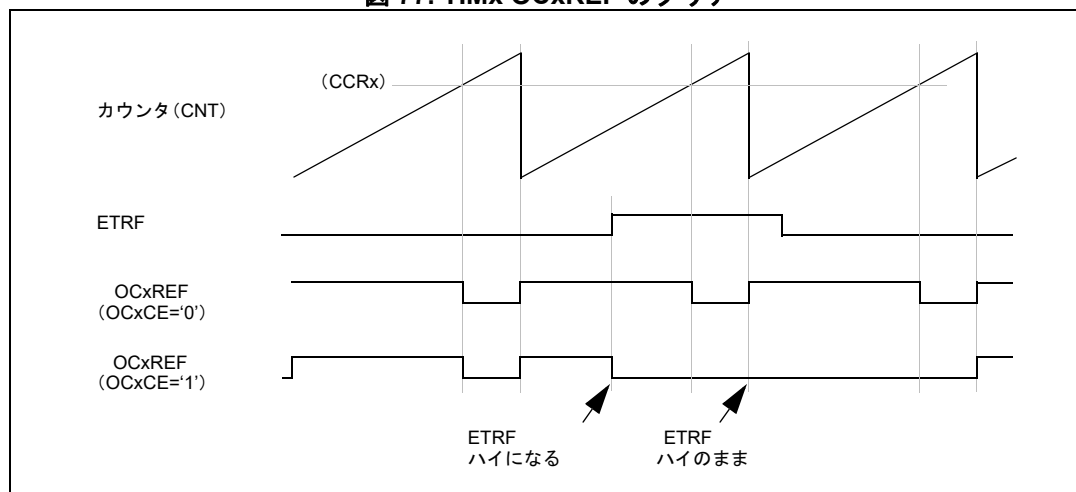
この機能は、出力比較モードと PWM モードでのみ使用でき、強制モードでは機能しません。

たとえば、ETR 信号をコンパレータの出力に接続して、電流処理に使用することができます。この場合、ETR は次のように設定する必要があります。

1. 外部トリガプリスケラはオフ状態に維持します (TIMx_SMCR レジスタの ETPS[1:0] ビットを“00”にセット)。
2. 外部クロックモード 2 を無効にします (TIMx_SMCR レジスタの ECE ビットを“0”にセット)。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、ユーザのニーズに応じて設定できます。

図 77 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 77. TIMx OCxREF のクリア



注： 100% デューティサイクルの PWM の場合 (CCR_x > ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

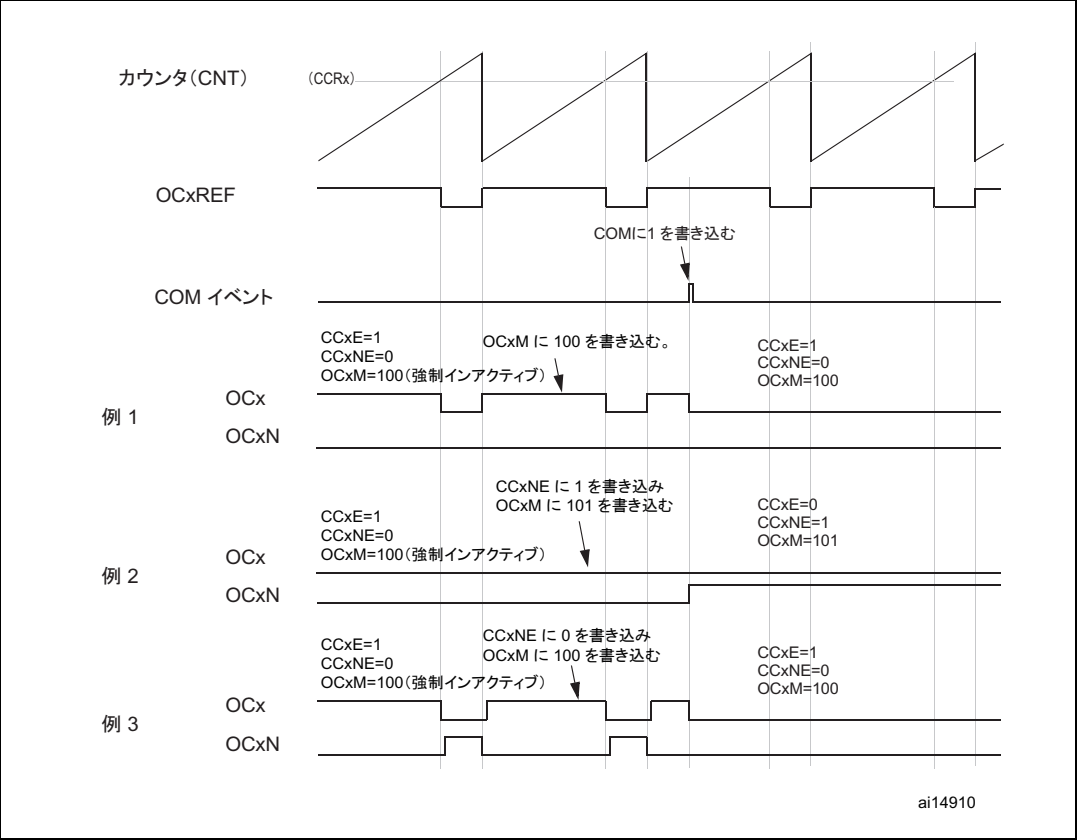
12.3.14 6 ステップ PWM 生成

チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転流イベントでシャドウビットに転送されます。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 78 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 78. 6 ステップ生成 COM の例 (OSSR=1)



12.3.15 ワンパルスモード

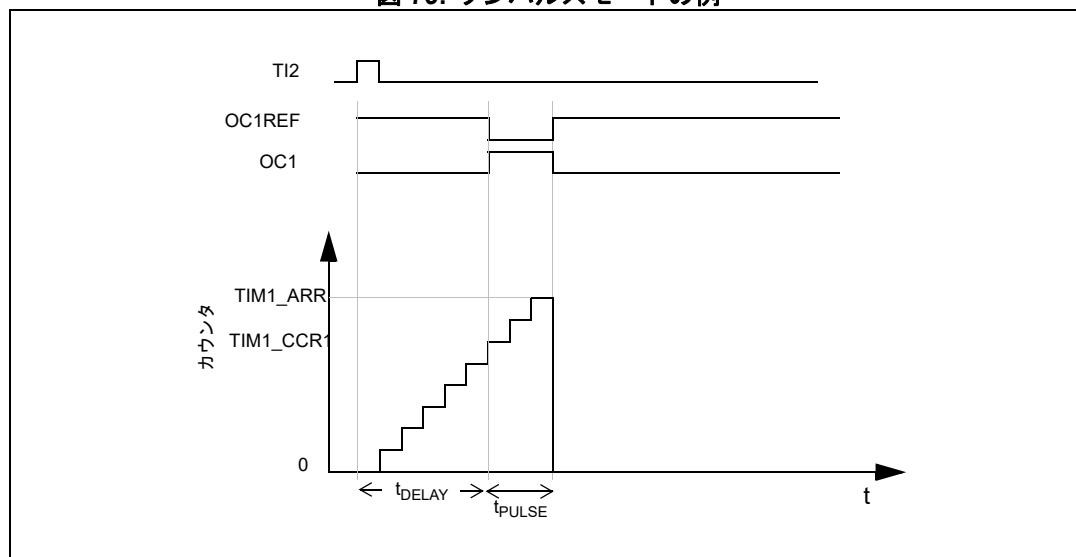
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 79. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで CC2P=“0”と CC2NP=“0”を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx_SMCR レジスタの TS ビットに“110”を書き込みます。
- TI2FP2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます（クロック周波数とカウンタプリスケアラを考慮に入れて）。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx_CCMR1 レジスタの OC1PE ビットに“1”を書込み、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込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので（シングルモード）、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します（カウンタが自動再ロード値に達して、“0”に戻る時点）。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース：OCx 高速イネーブル：

ワンパルスモードでは、TIx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF（および OCx）は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

12.3.16 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS=“001”を、TI1 エッジのみをカウントしている場合は SMS=“010”を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS=“011”を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2 つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。[表 49](#) を参照してください。カウンタのクロック供給は、TI1FP1 または TI2FP2（入力フィルタおよび極性選択の後）は TI1 と TI2、フィルタされず反転されない場合は TI1FP1=TI1、フィルタされず反転されない場合は TI2FP2=TI2）での有効な遷移ごとに行われます。ただし、カウンタは有効である（TIMx_CR1 レジスタの CEN ビットに“1”が書き込まれている）ことが前提です。2 つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力（TI1 または TI2）の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします（方向に応じて、0 から ARR まで、または ARR から 0 まで）。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、繰り返しカウンタ、トリガ出力の機能は通常どおりに機

能を続けます。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタはインクリメンタルエンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。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 49. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	高	ダウン	アップ	カウントなし	カウントなし
	低	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	高	カウントなし	カウントなし	アップ	ダウン
	低	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の 両方をカウント	高	ダウン	アップ	アップ	ダウン
	低	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割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 80 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S="01" (TIMx_CCMR2 レジスタ、TI1FP2 は TI2 に配置)
- CC1P="0"、CC1NP="0"、IC1F="0000" (TIMx_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1)
- CC2P="0"、CC2NP="0"、IC2F="0000" (TIMx_CCER レジスタ、TI1FP2 非反転、TI1FP2=TI2)
- SMS="011" (TIMx_SMCR レジスタ、両方の入力が立ち上がりと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 80. 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

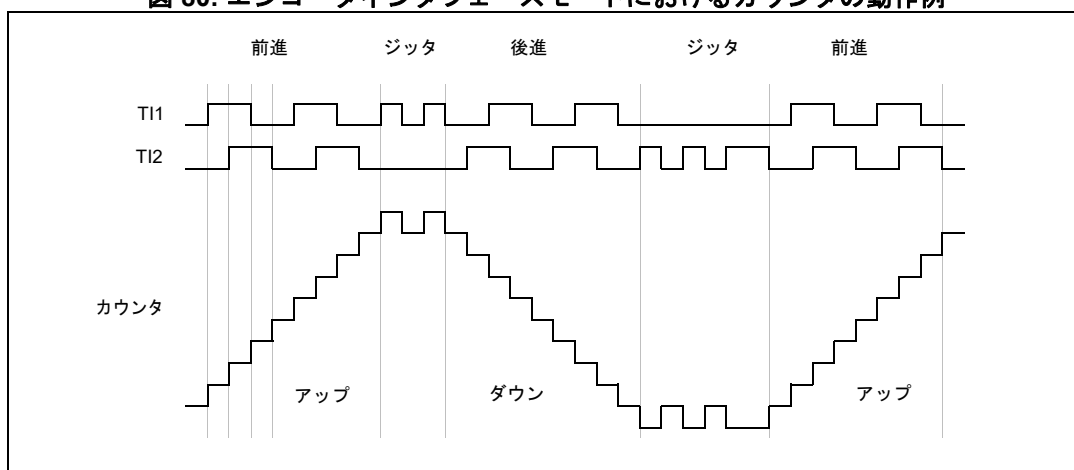
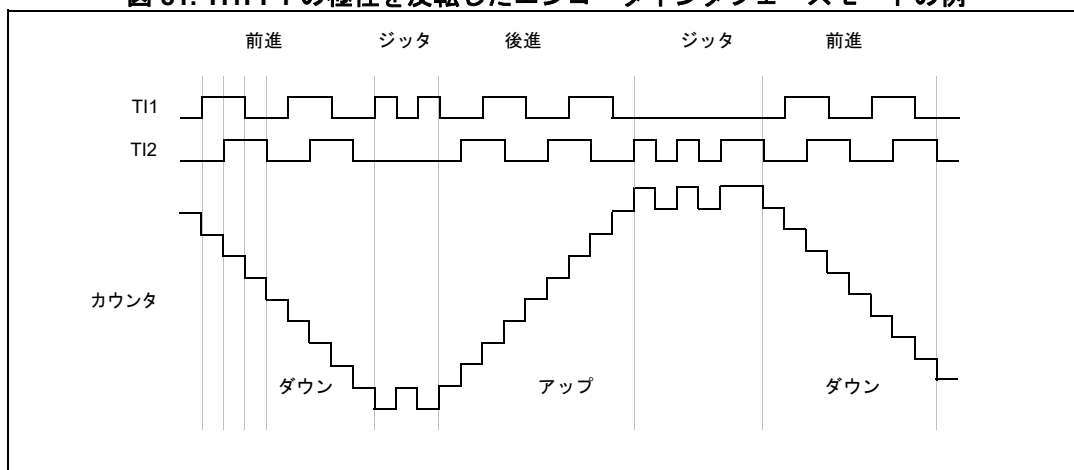


図 81 に、TI1FP1 の極性を反転したときのカウンタの動作を示します（上記と同じ設定ですが、CC1P="1"）。

図 81. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

12.3.17 タイマ入力 XOR 機能

TIMx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 3 つの入力ピン TIMx_CH1、TIMx_CH2、および TIMx_CH3 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。この機能をホールセンサのインタフェースに使用した例を、次の[セクション 12.3.18](#)に示します。

12.3.18 ホールセンサとのインタフェース

これは、高機能制御タイマ (TIM1) を使用して PWM 信号を生成し、モータと [図 82](#) で「インタフェースタイマ」と記されている別のタイマ TIMx (TIM2、TIM3、TIM4 または TIM5) を駆動することによって実現します。「インタフェースタイマ」は、XOR を通じて TI1 入力チャネル (TIMx_CR2 レジスタの TI1S ビットをセットすることで選択できます) に接続された 3 つのタイマ入力ピン (TIMx_CH1、TIMx_CH2、および TIMx_CH3) をキャプチャします。

スレーブモードコントローラはリセットモードに設定され、スレーブ入力は TI1F_ED です。したがって、3 つの入力のいずれかが反転するごとに、カウンタは 0 からカウントをリスタートします。これが、ホール入力の変化によってトリガされるタイムベースとなります。

「インタフェースタイマ」上で、キャプチャ/比較チャネル 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 65 : 258 ページのキャプチャ/比較チャネル \(例: チャネル 1 入力スケーリング\)](#) 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) のチャ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) に送られます。

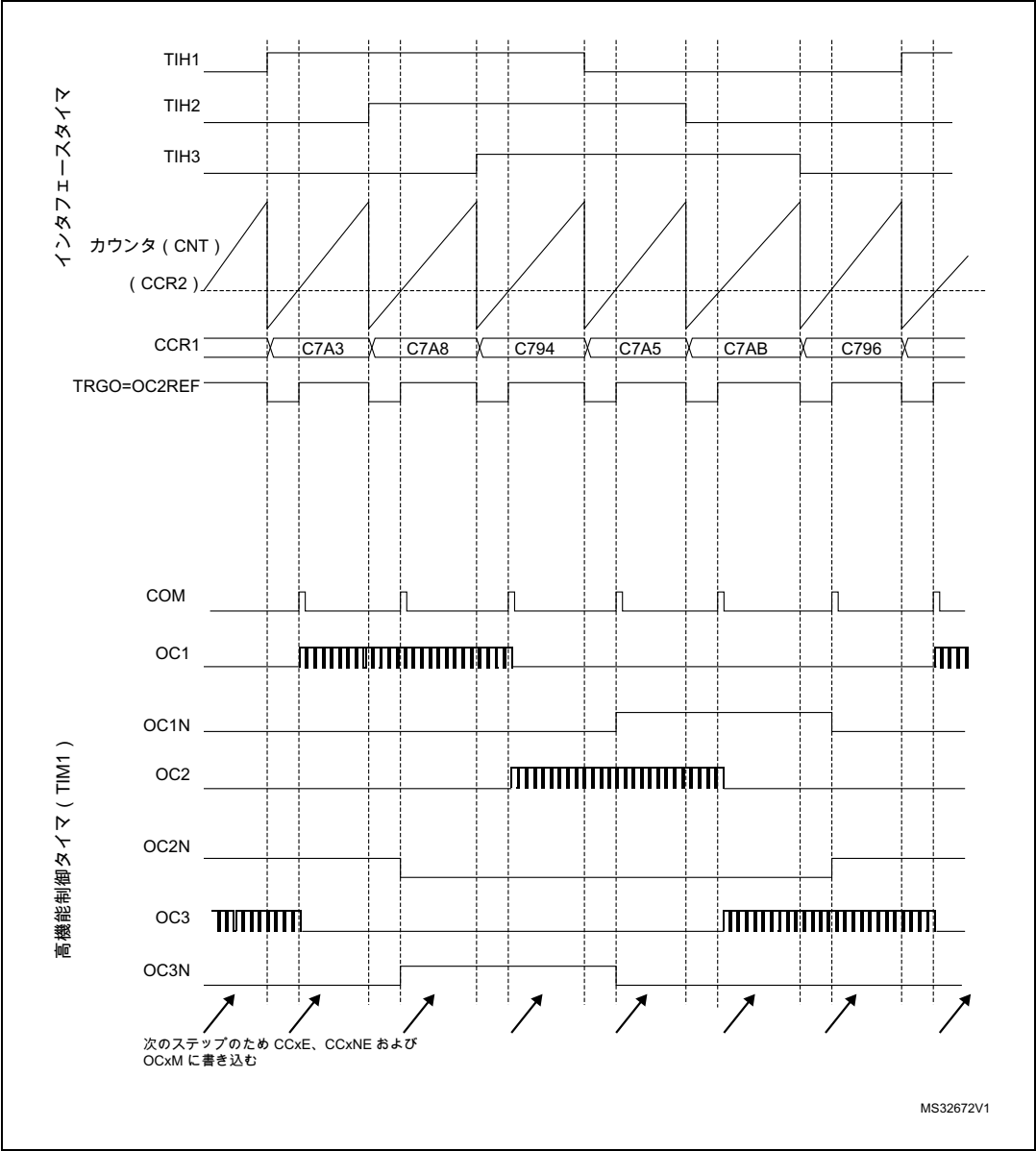
例: TIMx タイマの 1 つに接続されているホール入力に変化するたびに、プログラミングした遅延の後に高機能制御タイマ TIM1 の PWM 設定を変更するとします。

- 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、プリスケールを設定します。
- チャネル 1 をキャプチャモード (TRC 選択) にプログラムします。すなわち、TIMx_CCMR1 レジスタの CC1S ビットに“11”を書き込みます。必要な場合は、デジタルフィルタをプログラムすることもできます。
- チャネル 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込みサブルーチンで行うことができます)。

[図 82](#) に、この例を示します。

図 82. ホールセンサインタフェースの例



12.3.19 TIMx と外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

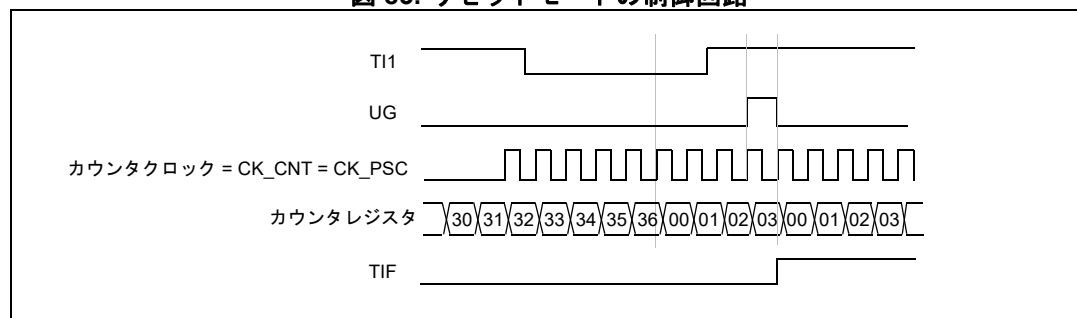
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、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 83. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

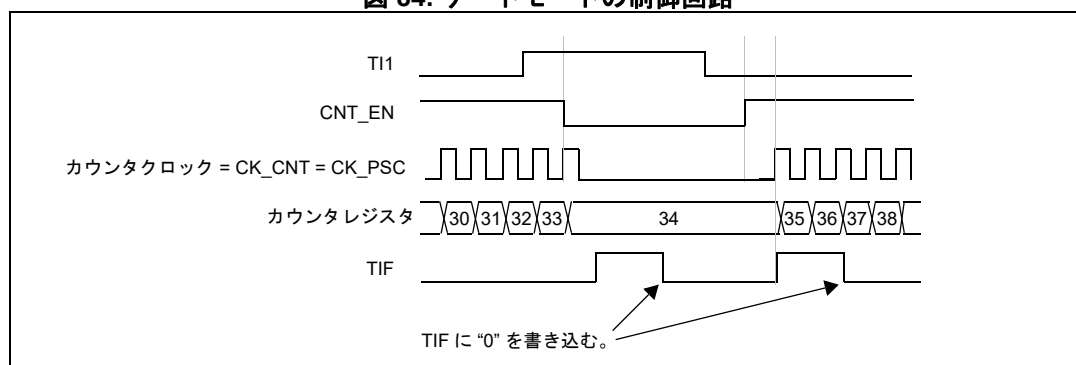
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01 ビット)。TIMx_CCER レジスタで 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 84. 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。

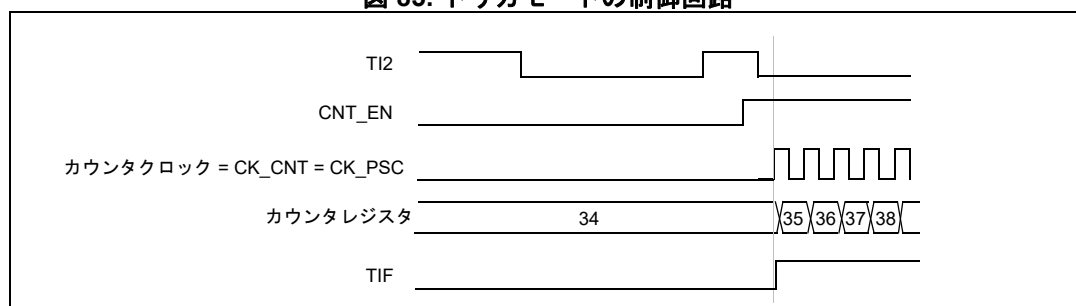
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

- 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なので、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタで 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 85. トリガモードの制御回路



スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます（ただし、外部クロックモード 1 とエンコーダモードは除きます）。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます（リセットモード、ゲートモード、またはトリガモード）。TIMx_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

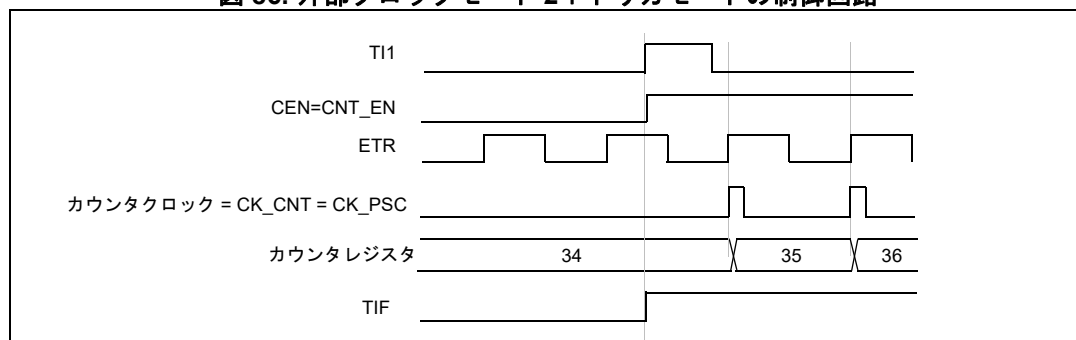
1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000 : フィルタなし
 - ETPS = 00 : プリスケアラは無効
 - ETP = 0 : ETR の立ち上がりエッジを検出。ECE = 1 で外部クロックモード 2 を有効にする。
2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000 : フィルタなし。
 - キャプチャプリスケアラはトリガには使用されないなので、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします（そして、立ち上がりエッジのみを検出します）。

3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 86. 外部クロックモード 2+トリガモードの制御回路



12.3.20 タイマの同期

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 13.3.15 : 341 ページのタイマの同期](#)を参照してください。

12.3.21 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 23.16.2 : タイマ、ウォッチドッグ、および I²C のデバッグサポート](#)を参照してください。

12.4 TIM1 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、[34 ページのセクション 1.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）またはワード（32 ビット）単位で行うことができます。

12.4.1 TIM1 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
予約済み						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

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]**：クロック分周

このビットフィールドは、タイマクロック（CK_INT）周波数と、デッドタイムジェネレータとデジタルフィルタ（ETR、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 6:5 **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ネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割込みフラグは、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
 - 1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCR_x) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケ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。

12.4.2 TIM1 制御レジスタ 2 (TIM_x_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **OIS4** : 出力アイドル状態 4 (OC4 出力)
- OIS1 ビットの説明を参照してください。
- ビット 13 **OIS3N** : 出力アイドル状態 3 (OC3N 出力)
- OIS1N ビットの説明を参照してください。
- ビット 12 **OIS3** : 出力アイドル状態 3 (OC3 出力)
- OIS1 ビットの説明を参照してください。

ビット 11 **OIS2N** : 出力アイドル状態 2 (OC2N 出力)
OIS1N ビットの説明を参照してください。

ビット 10 **OIS2** : 出力アイドル状態 2 (OC2 出力)
OIS1 ビットの説明を参照してください。

ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)
0 : MOE=0 のとき、デッドタイム後に OC1N=0
1 : MOE=1 のとき、デッドタイム後に OC1N=0

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)
0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)
1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 7 **TI1S** : TI1 選択
0 : TIMx_CH1 ピンが TI1 入力に接続されます。
1 : TIMx_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[2:0]** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在しません (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とき (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。 (TRGO)。

100 : 比較 - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : 比較 - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

110 : 比較 - OC3REF 信号がトリガ出力 (TRGO) として使用されます。

111 : 比較 - OC4REF 信号がトリガ出力 (TRGO) として使用されます。

注 : スレーブタイマおよび 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。書込みの後、これらのビットは、遷移イベント (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。

12.4.3 TIM1 のスレーブモード制御レジスタ (TIMx_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]				Res.	SMS[2:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	Res.	rw	rw

ビット 15 **ETP** : 外部トリガ極性

このビットは、ETR と 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 個のイベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 2

0010 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 4

0011 : $f_{SAMPLING} = f_{CK_INT}$ 、N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 8

ビット 7 **MSM** : マスタ/スレーブモード

- 0 : 影響なし。
- 1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS[2:0]** : トリガ選択

- このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。
- 000 : 内部トリガ 0 (ITR0)
 - 001 : 内部トリガ 1 (ITR1)
 - 010 : 内部トリガ 2 (ITR2)
 - 011 : 内部トリガ 3 (ITR3)
 - 100 : TI1 エッジ検出回路 (TI1F_ED)
 - 101 : フィルタタイマ入力 1 (TI1FP1)
 - 110 : フィルタタイマ入力 2 (TI2FP2)
 - 111 : 外部トリガ入力 (ETRF)
- 各タイマでの ITRx の詳細については、[表 50:288 ページのTIMx 内部トリガ接続](#)を参照してください。

注： 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

- 外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。
- 000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。
 - 001 : エンコーダモード 1 - カウンタは、TI1FP2 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。
 - 010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。
 - 011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。
 - 100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
 - 101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。
 - 110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。
 - 111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。
- 注： トリガ入力として TI1F_ED が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。
- 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 50. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM5_TRGO	TIM2_TRGO	TIM3_TRGO	TIM4_TRGO

12.4.4 TIM1 DMA／割込み有効レジスタ (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	
	r/w	r/w	r/w	r/w	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 予約済みであり、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** : ブレーク割込みイネーブル

0 : ブレーク割込みは無効です。

1 : ブレーク割込みは有効です。

ビット 6 **TIE** : トリガ割込みイネーブル

0 : トリガ割込みは無効です。

1 : トリガ割込みは有効です。

ビット 5 **COMIE** : COM 割込みイネーブル

0 : COM 割込みは無効です。

1 : COM 割込みは有効です。

ビット 4 **CC4IE** : キャプチャ／比較 4 割込みイネーブル

0 : CC4 割込みは無効です。

1 : CC4 割込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割込みイネーブル

0 : CC3 割込みは無効です。

1 : CC3 割込みは有効です。

ビット 2 **CC2IE** : キャプチャ／比較 2 割込みイネーブル

0 : CC2 割込みは無効です。

1 : CC2 割込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割込みイネーブル

0 : CC1 割込みは無効です。

1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル

0 : 更新割込みは無効です。

1 : 更新割込みは有効です。

12.4.5 TIM1 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
予約済み	CC4OF		CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF	
	rc_w0		rc_w0	rc_w0	rc_w0	Res.	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ／比較 4 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ／比較 3 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ／比較 2 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BIF** : ブレーク割込みフラグ

このフラグは、ブレーク入力が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 : ブレーク入力で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 : トリガ割込みが保留中です。

ビット 5 COMIF : COM 割込みフラグ

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、トリガイイベントによって CNT が再初期化されたとき（[セクション 12.4.3 : TIM1 のスレープモード制御レジスタ \(TIMx_SMCR\)](#) を参照）。

12.4.6 TIM1 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
予約済み								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4 **CC4G** : キャプチャ / 比較 4 イベント生成

CC1G の説明を参照してください。

ビット 3 **CC3G** : キャプチャ / 比較 3 イベント生成

CC1G の説明を参照してください。

ビット 2 **CC2G** : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx_ARR) をとります。

12.4.7 TIM1 のキャプチャ／比較モードレジスタ 1 (TIMx_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード :

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

OC1CE : 出力比較 1 クリアイネーブル

0 : OC1Ref は ETRF 入力の影響を受けません。

1 : OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注 : 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2: PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときのみ、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。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャネルは出力として設定されます。

01 : CC1 チャネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャネルは出力として設定されます。

01 : CC2 チャネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC2S ビットは、チャネルがオフ (TIMx_CCER レジスタの CC2E=0) のときに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 個のイベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{CK_INT}$, $N = 2$

0010 : $f_{SAMPLING} = f_{CK_INT}$, $N = 4$

0011 : $f_{SAMPLING} = f_{CK_INT}$, $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$, $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$, $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$, $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$, $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$, $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$, $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$, $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$, $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$, $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$, $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$, $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$, $N = 8$

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

12.4.8 TIM1 のキャプチャ／比較モードレジスタ 2 (TIMx_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000

上記の CCMR1 レジスタの説明を参照してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OC4 CE	OC4M[2:0]				OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]				OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]					IC3F[3:0]				IC3PSC[1:0]				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	

出力比較モード

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC3S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケアラ

ビット 9:8 **CC4S** : キャプチャ/比較 4 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケアラ

ビット 1:0 **CC3S** : キャプチャ/比較 3 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC3S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

12.4.9 TIM1 のキャプチャ/比較有効レジスタ (TIMx_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	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 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC4P** : キャプチャ/比較 4 出力極性

CC1P の説明を参照してください。

ビット 12 **CC4E** : キャプチャ/比較 4 出力イネーブル

CC1E の説明を参照してください。

ビット 11 **CC3NP** : キャプチャ/比較 3 相補出力極性

CC1NP の説明を参照してください。

ビット 10 **CC3NE** : キャプチャ/比較 3 相補出力イネーブル

CC1NE の説明を参照してください。

ビット 9 **CC3P** : キャプチャ/比較 3 出力極性

CC1P の説明を参照してください。

ビット 8 **CC3E** : キャプチャ/比較 3 出力イネーブル

CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ / 比較 2 相補出力極性

CC1NP の説明を参照してください。

ビット 6 **CC2NE** : キャプチャ / 比較 2 相補出力イネーブル

CC1NE の説明を参照してください。

ビット 5 **CC2P** : キャプチャ / 比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ / 比較 2 出力イネーブル

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ / 比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1N はアクティブハイです。

1 : OC1N はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

このビットは、TI1FP1 と TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、遷移イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になると書き込みできません。

ビット 2 **CC1NE** : キャプチャ / 比較 1 相補出力イネーブル

0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NE アクティブビットは、遷移イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 1 **CC1P** : キャプチャ / 比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

00 : 非反転/立ち上がりエッジ

この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

01 : 反転/立ち下がりエッジ

この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されます (ゲートモードまたはエンコーダモードでのトリガ動作)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ

この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。この設定をエンコーダモードで使用することはできません。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、遷移イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 0 **CC1E** : キャプチャ / 比較 1 出力イネーブル

CC1 チャンネルが出力として設定されている場合 :

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1E アクティブビットは、遷移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 51. 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 ⁽¹⁾	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	0	0	0	出力無効(タイマによって駆動されない)、OCx=0、OCx_EN=0	出力無効(タイマによって駆動されない)、OCxN=0、OCxN_EN=0
		0	0	1	出力無効(タイマによって駆動されない)、OCx=0、OCx_EN=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP、OCxN_EN = 1
		0	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	出力無効(タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補(OCREF ではなく) + 極性 + デッドタイム OCxN_EN=1
		1	0	0	出力無効(タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効(タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
		1	0	1	オフ状態(インアクティブ状態で出力有効) OCx=CCxP、OCx_EN=1	OCxREF + 極性 OCxN=OCxREF xor CCxNP、 OCxN_EN=1
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	オフ状態(インアクティブ状態で出力有効) OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補(OCREF ではなく) + 極性 + デッドタイム OCxN_EN=1
0	0	X	0	0	出力無効(タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効(タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
	0		0	1	出力無効(タイマによって駆動されない)	
	0		1	0	非同期: OCx=CCxP、OCx_EN=0、OCxN=CCxNP、OCxN_EN=0 クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。 ただし、OISx と OISxN は、アクティブ状態における OCx と OCxN の両方に対応しないことを前提とします。	
	0		1	1	出力無効(タイマによって駆動されない) OCx=CCxP、OCx_EN=0	
	1		0	0	出力無効(タイマによって駆動されない) OCx=CCxP、OCx_EN=0	
	1		0	1	オフ状態(インアクティブ状態で出力有効)	
	1		1	0	非同期: OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。 ただし、OISx と OISxN は、アクティブ状態における OCx と OCxN の両方に対応しないことを前提とします。	
	1		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 (CCxE=CCxNE=0)、OISx、OISxN、CCxP、および CCxNP はクリアされたままでなければなりません。

注: 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、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。

12.4.10 TIM1 のカウンタ (TIMx_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CNT[15:0]** : カウンタ値

12.4.11 TIM1 のプリスケアラ (TIMx_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 (CK_CNT) は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブ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)。

12.4.12 TIM1 の自動再ロードレジスタ (TIMx_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **ARR[15:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 12.3.1 : 244 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

12.4.13 TIM1 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
予約済み								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周期的な転送）と、割込みが有効な場合の更新割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

12.4.14 TIM1 のキャプチャ／比較レジスタ 1 (TIMx_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、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。

12.4.15 TIM1 のキャプチャ／比較レジスタ 2 (TIMx_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR2[15:0]** : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、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。

12.4.16 TIM1 のキャプチャ／比較レジスタ 3 (TIMx_CCR3)

アドレスオフセット : 0x3C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR3[15:0]** : キャプチャ / 比較値

CC3 チャンネルが出力として設定されている場合 :

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR3 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC3 出力に送信される値を含みます。

CC3 チャンネルが入力として設定されている場合 :

CCR3 は、最後の入力キャプチャ 3 イベント (IC3) によって転送されたカウンタ値です。

12.4.17 TIM1 のキャプチャ／比較レジスタ 4 (TIMx_CCR4)

アドレスオフセット : 0x40

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR4[15:0]** : キャプチャ / 比較値

CC4 チャンネルが出力として設定されている場合 :

CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR4 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC4 出力に送信される値を含みます。

CC4 チャンネルが入力として設定されている場合 :

CCR4 は、最後の入力キャプチャ 4 イベント (IC4) によって転送されたカウンタ値です。

12.4.18 TIM1 ブレークおよびデッドタイムレジスタ (TIMx_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注 : ビット AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] は、LOCK 設定に応じて書き込みがロックされるので、TIMx_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 15 **MOE** : メイン出力イネーブル

このビットは、ブレーク入力がアクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャンネルに対してのみ有効です。

0 : OC および OCN 出力は無効か、強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 12.4.9 : 297 ページの TIM1 のキャプチャ／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 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 はアクティブラーです。
- 1 : ブレーク入力 BRK はアクティブハイです。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 BKE : ブレークイネーブル

- 0 : ブレーク入力 (BRK および CSS クロック障害イベント) は無効です。
- 1 : ブレーク入力 (BRK および CSS クロック障害イベント) は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 12.4.9 : 297 ページの TIM1 のキャプチャ/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 (OC/OCN イネーブル出力信号 = 0)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります。この場合、OC/OCN イネーブル出力信号 = 1 です。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 12.4.9 : 297 ページの TIM1 のキャプチャ/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 (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停止されます。

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 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)。

12.4.19 TIM1 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
予約済み			DBL[4:0]					予約済み			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 ビットのベクタは、DMA 転送回数（タイマは、TIMx_DMAR アドレスに対して読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検出します）を指定します。

TIMx_DMAR アドレス)

00000 : 1 回転送

00001 : 2 回転送

00010 : 3 回転送

...

10001 : 18 回転送

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx_DMAR アドレスを通じて読出し/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。

12.4.20 完全転送用の TIM1 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 レジスタへの読出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(\text{TIMx_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL は TIMx_DCR 内で設定)。

DMA バースト機能の使用例

この例では、タイマ DMA バースト機能を使って CCRx レジスタ値を更新します (x = 2、3、4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

- 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします (下の注を参照)。
 - サーキュラモードは無効です。
- DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 転送、DBA = 0xE。
- TIMx 更新 DMA リクエストを有効にします (DIER レジスタの UDE ビットをセット)。
- TIMx を有効化
- DMA チャンネルを有効化注 :

注 : この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

12.4.21 TIM1 レジスタマップ

TIM1 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 52. TIM1 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	予約済み																						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				
0x04	TIMx_CR2	予約済み																	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TIS1		MMS[2:0]		CCDS	CCUS	予約済み	CCPC		
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	TIMx_SMCR	予約済み																	ETP	ECE	ETPS [1:0]		ETF[3:0]			MSM	TS[2:0]			予約済み	SMS[2:0]				
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	TIMx_DIER	予約済み																	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	0	0
0x10	TIMx_SR	予約済み																	CC4OF	CC3OF	CC2OF	CC1OF	予約済み		BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF			
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	TIMx_EGR	予約済み																						BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG				
	リセット値																							0	0	0	0	0	0	0	0				
0x18	TIMx_CCMR1 出力比較モード	予約済み																	OC2CE	OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	TIMx_CCMR1 入力キャプチャ モード	予約済み																	IC2F[3:0]			IC2 PSC [1:0]	CC2S [1:0]		IC1F[3:0]			IC1 PSC [1:0]	CC1S [1:0]						
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x1C	TIMx_CCMR2 出力比較モード	予約済み																	OC4CE	OC4M [2:0]		OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]		OC3PE	OC3FE	CC3S [1:0]				
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	TIMx_CCMR2 入力キャプチャ モード	予約済み																	IC4F[3:0]			IC4 PSC [1:0]	CC4S [1:0]		IC3F[3:0]			IC3 PSC [1:0]	CC3S [1:0]						
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x20	TIMx_CCER	予約済み																	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E			
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x24	TIMx_CNT	予約済み															CNT[15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x28	TIMx_PSC	予約済み															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	予約済み															ARR[15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

表 52. TIM1 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	TIMx_RCR	予約済み																						REP[7:0]									
	リセット値																							0	0	0	0	0	0	0	0		
0x34	TIMx_CCR1	予約済み												CCR1[15:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x38	TIMx_CCR2	予約済み												CCR2[15:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x3C	TIMx_CCR3	予約済み												CCR3[15:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x40	TIMx_CCR4	予約済み												CCR4[15:0]																			
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x44	TIMx_BDTR	予約済み												MOE	AOE	BKP	BKE	OSR	OSI	LOCK [1:0]	DT[7:0]												
	リセット値													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x48	TIMx_DCR	予約済み												DBL[4:0]				予約済み				DBA[4:0]											
	リセット値													0	0	0	0					0	0	0	0	0	0	0	0	0			
0x4C	TIMx_DMAR	予約済み												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			

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

13 汎用タイマ (TIM2 から TIM5)

13.1 TIM2 から TIM5 の概要

この汎用タイマは、プログラム可能なプリスケアラによって駆動される 16 ビットまたは 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、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、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。

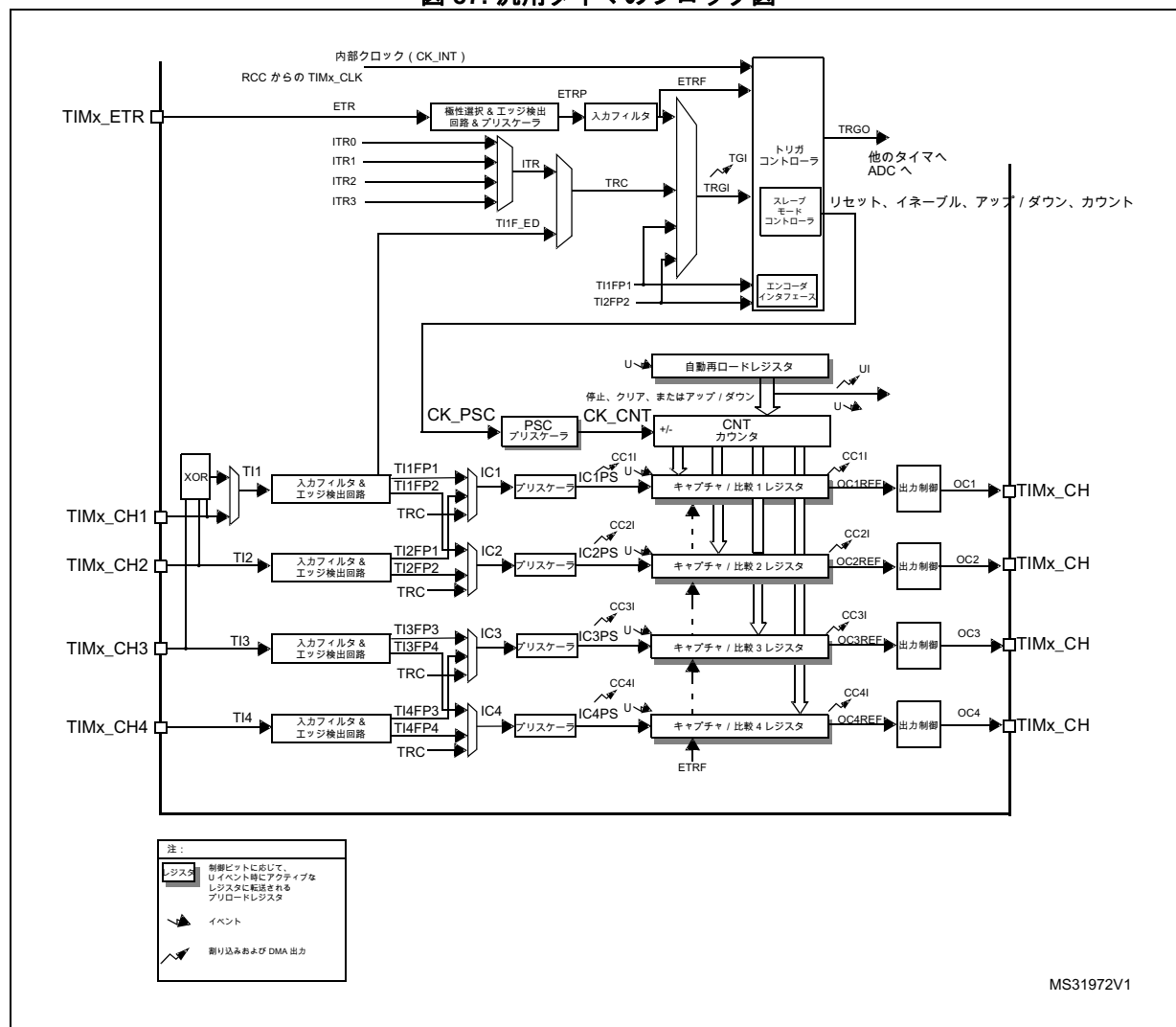
タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 13.3.15](#) に示すように、相互に同期させることができます。

13.2 TIM2 から TIM5 の主な機能

汎用 TIMx タイマの主な機能は、次のとおりです。

- 16 ビット (TIM3 と TIM4) または 32 ビット (TIM2 と TIM5) のアップ、ダウン、アップ/ダウン自動再ロードカウンタ。
- カウンタクロック周波数を、1 から 65536 の間で (動作中でも) 分周する 16 ビットプログラム可能プリスケアラ。
- 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 87. 汎用タイマのブロック図



13.3 TIM2 から TIM5 の機能説明

13.3.1 タイムベースユニット

プログラマブルなタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビット/32 ビットカウンタです。カウンタはカウントアップの動作が可能です。カウンタのクロックは、プリスケラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- ・ カウンタレジスタ (TIMx_CNT)
- ・ プリスケラレジスタ (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 88 と 図 89 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 88. 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

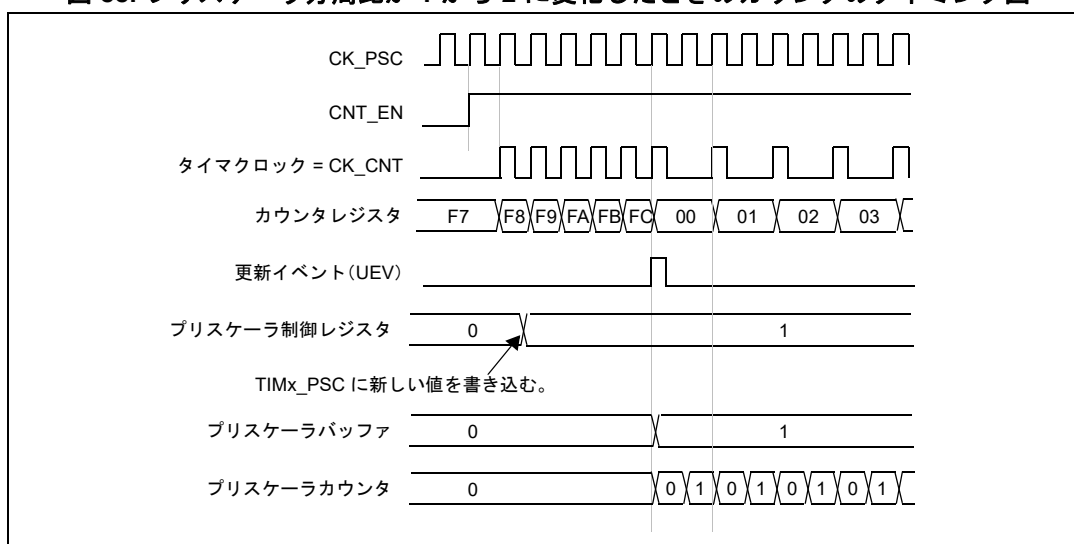
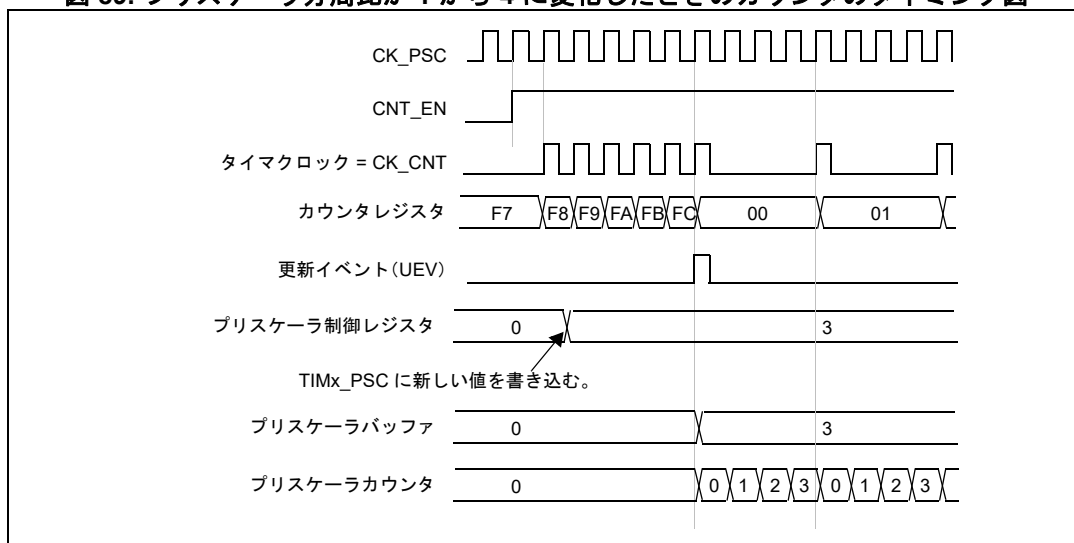


図 89. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



13.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることで生成できます。

UEV イベントは、TIMx_CR1 レジスタの UDIS ビットをセットすることにより、ソフトウェアで無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや 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 90. 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

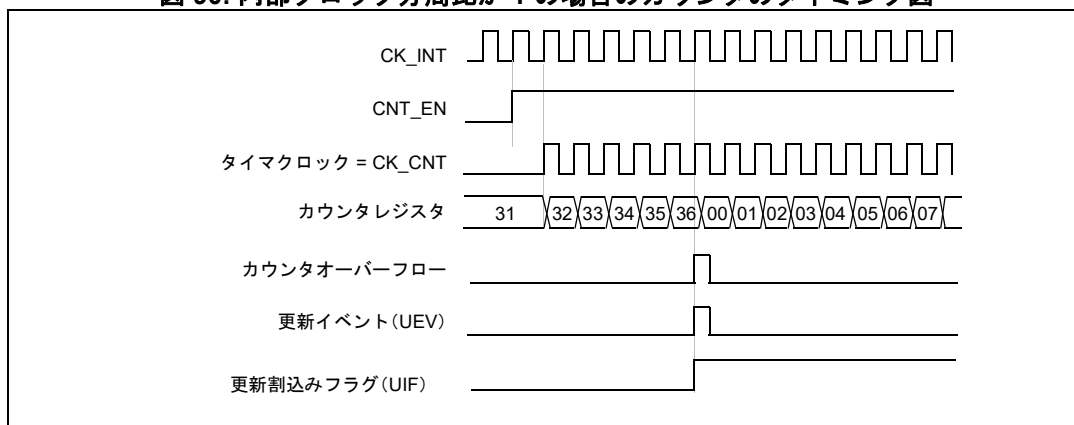


図 91. 内部クロック分周比が 2 の場合のカウンタのタイミング図

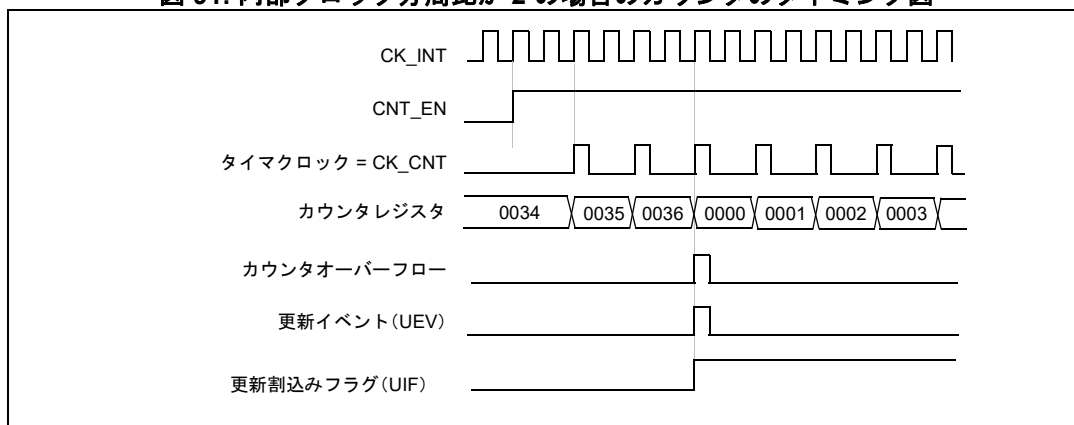


図 92. 内部クロック分周比が 4 の場合のカウンタのタイミング図

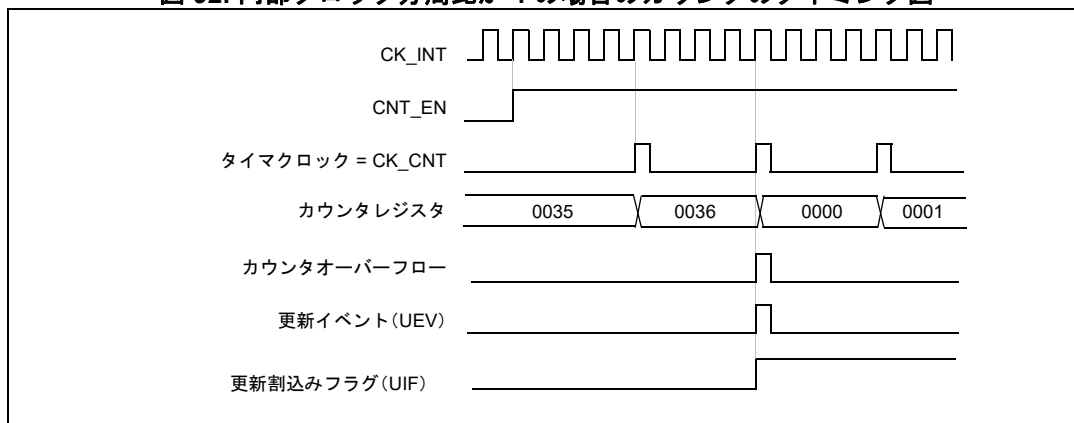


図 93. 内部クロック分周比が N の場合のカウンタのタイミング図

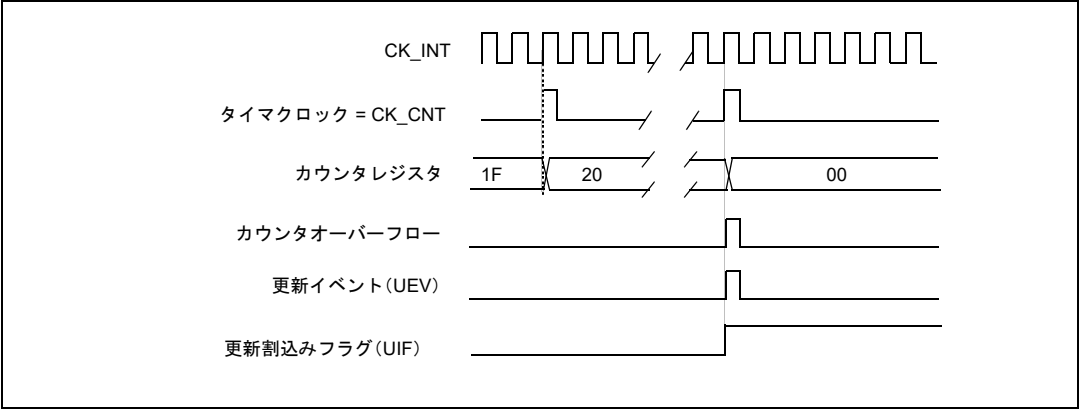


図 94. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

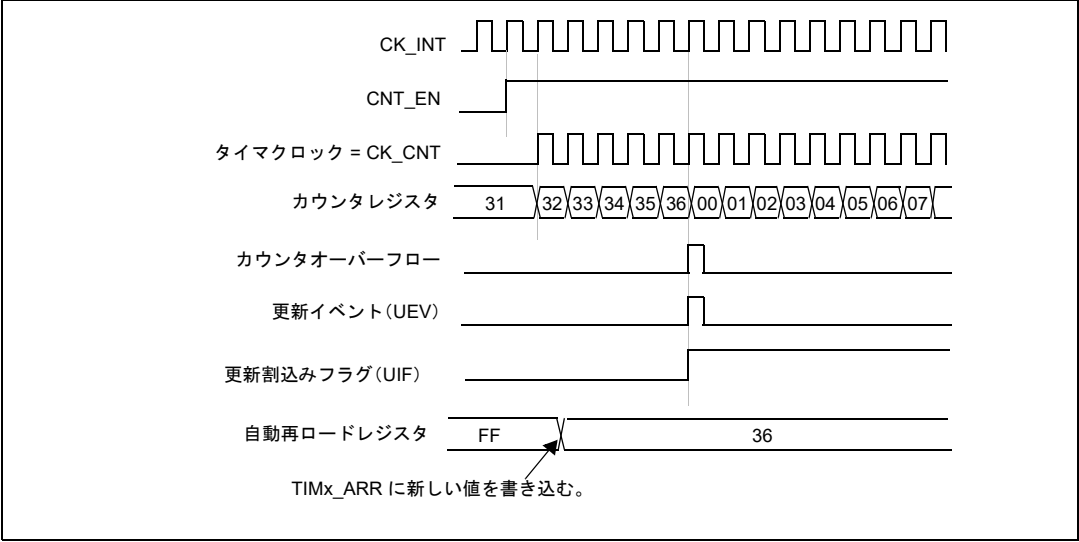
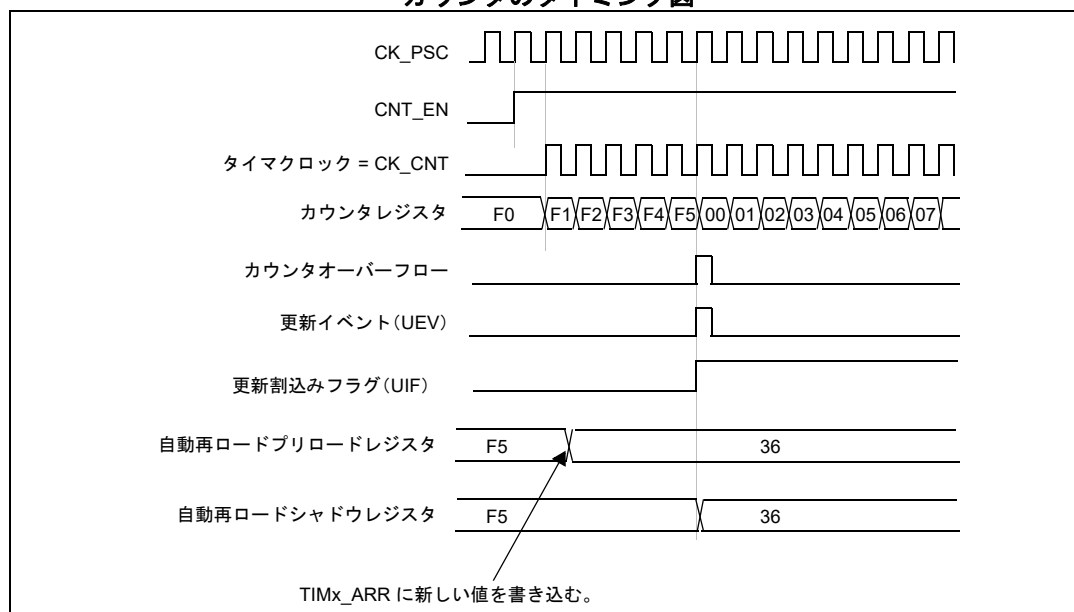


図 95. ARPE=1 (TIMx_ARR はプリロードされる) の場合の更新イベント時のカウンタのタイミング図



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。

更新イベントは、カウンタアンダーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることにより生成できます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや 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。

- プリスケアラのパ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 96. 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

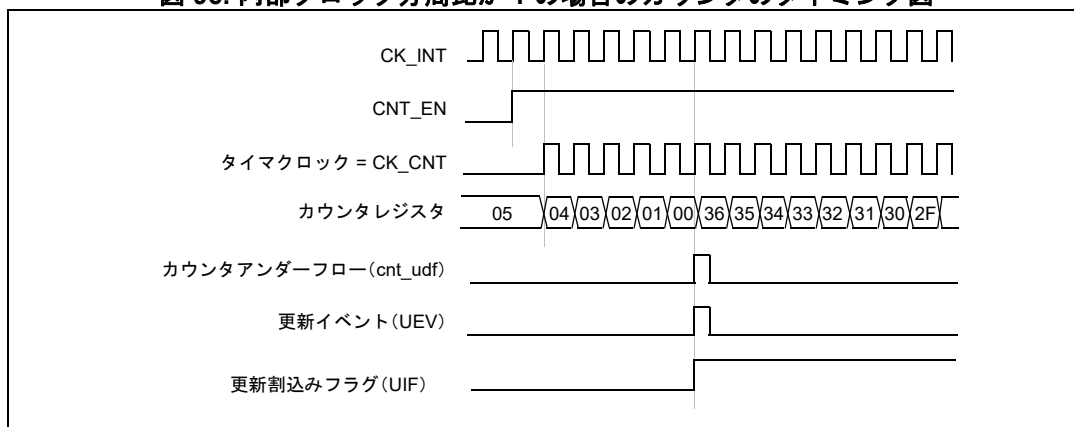


図 97. 内部クロック分周比が 2 の場合のカウンタのタイミング図

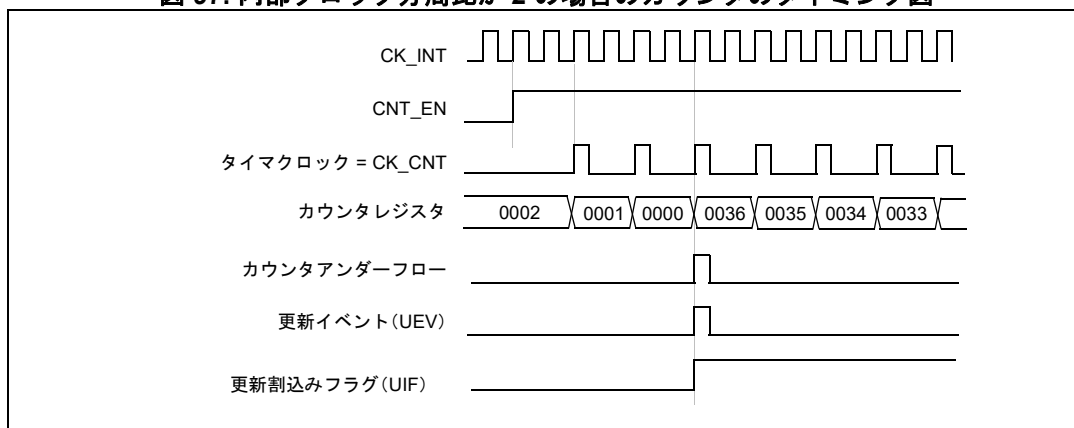


図 98. 内部クロック分周比が 4 の場合のカウンタのタイミング図

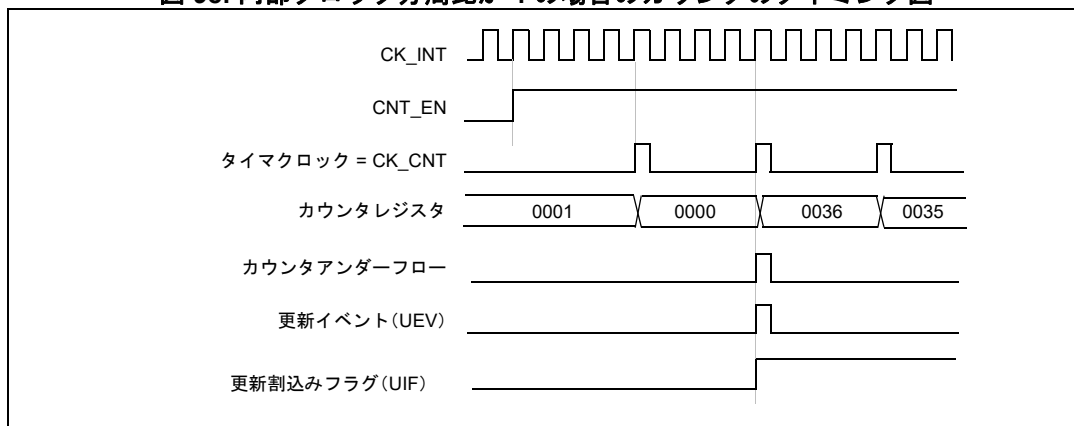


図 99. 内部クロック分周比が N の場合のカウンタのタイミング図

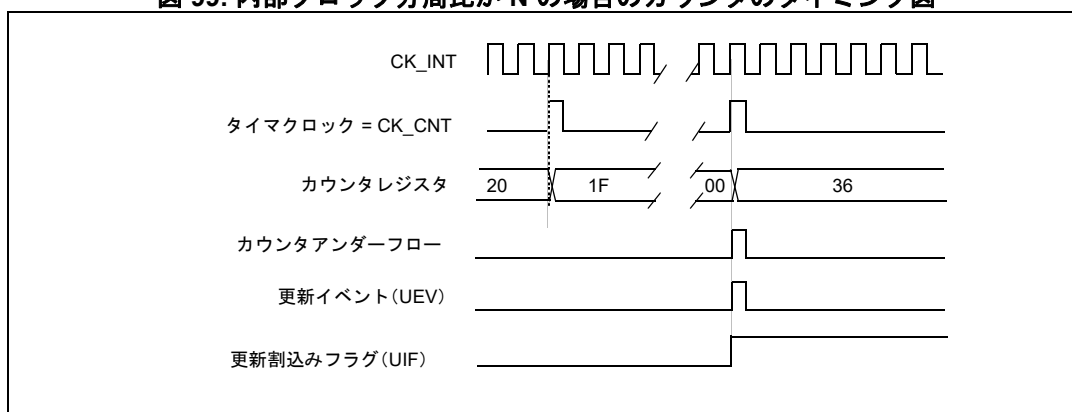
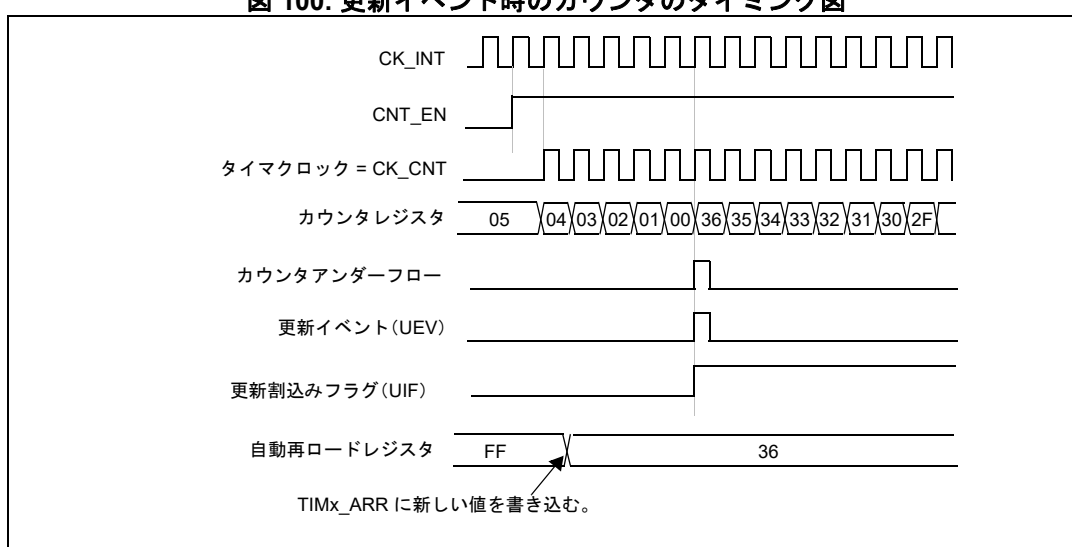


図 100. 更新イベント時のカウンタのタイミング図



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。

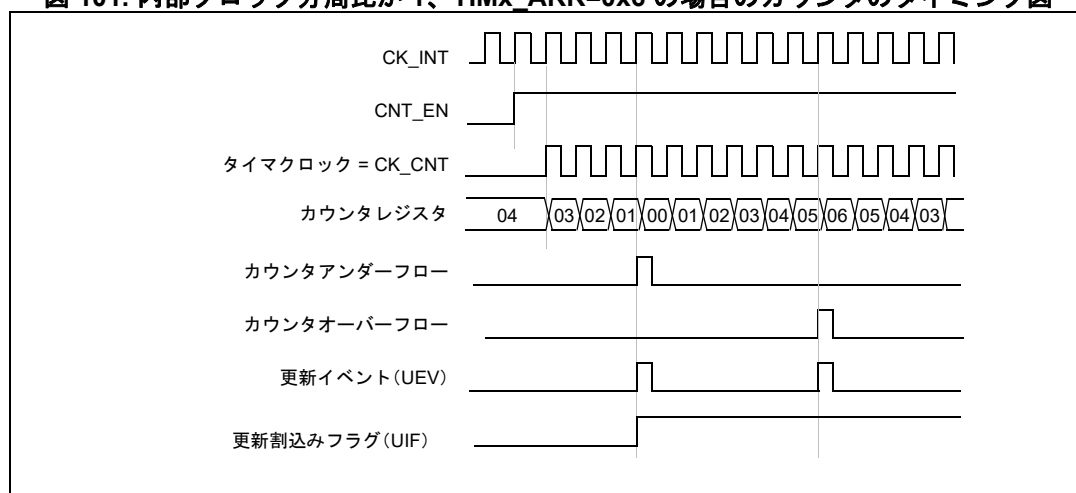
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

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。

- プリスケアラのパ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 101. 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 13.4.1: 346 ページのTIMx 制御レジスタ 1 \(TIMx_CR1\)](#) を参照してください)。

図 102. 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

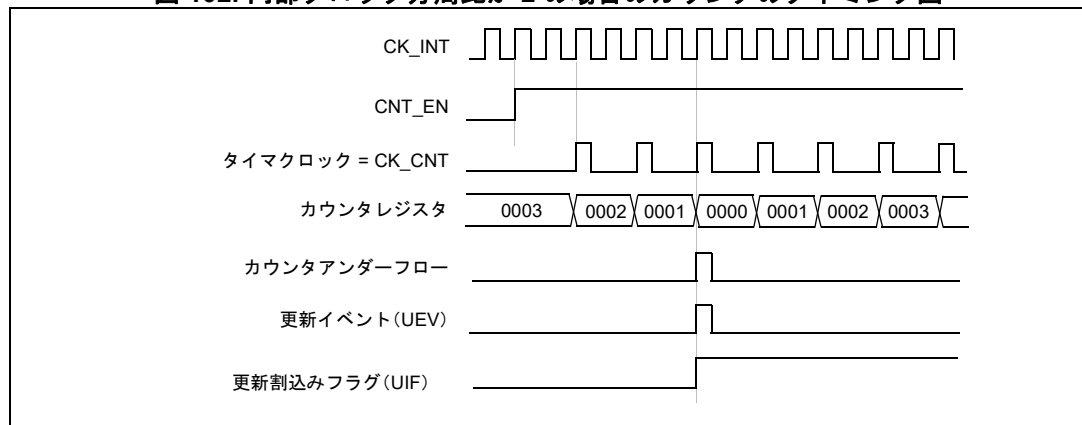
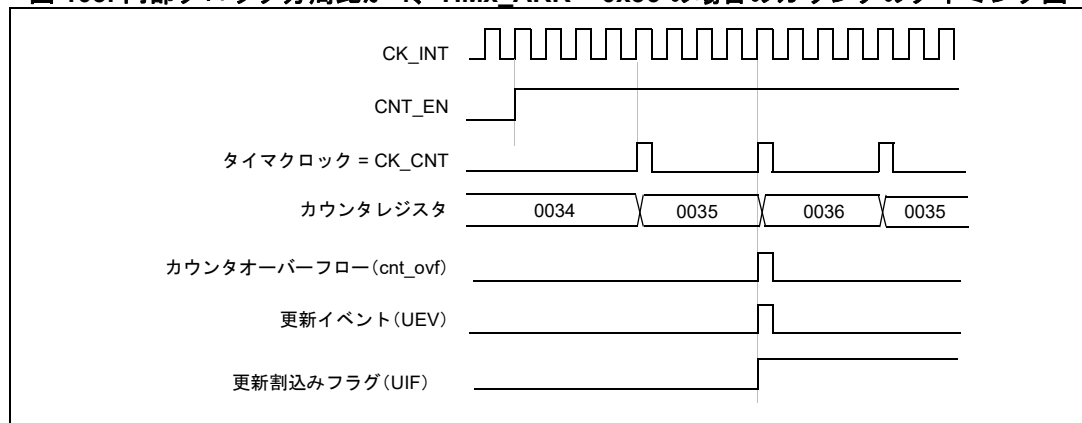


図 103. 内部クロック分周比が 4、TIMx_ARR = 0x36 の場合のカウンタのタイミング図



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 104. 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

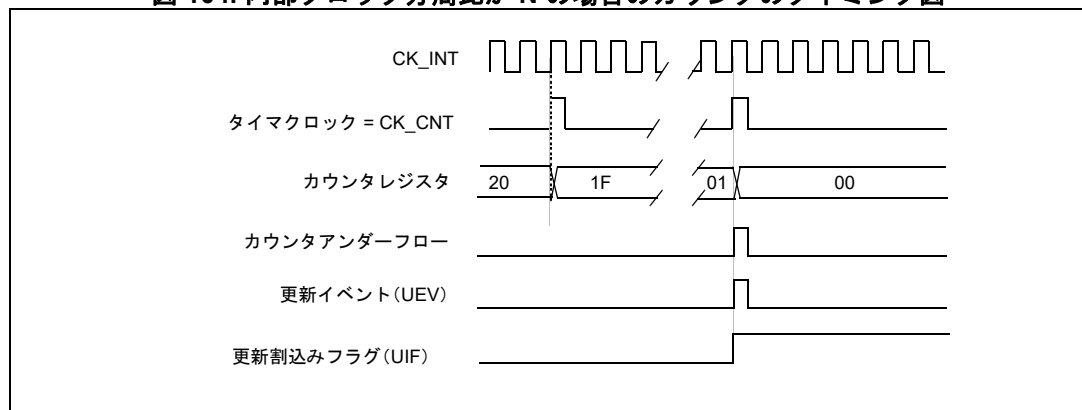


図 105. ARPE=1（カウンタアンダーフロー）の場合の更新イベント時、
カウンタタイミング図

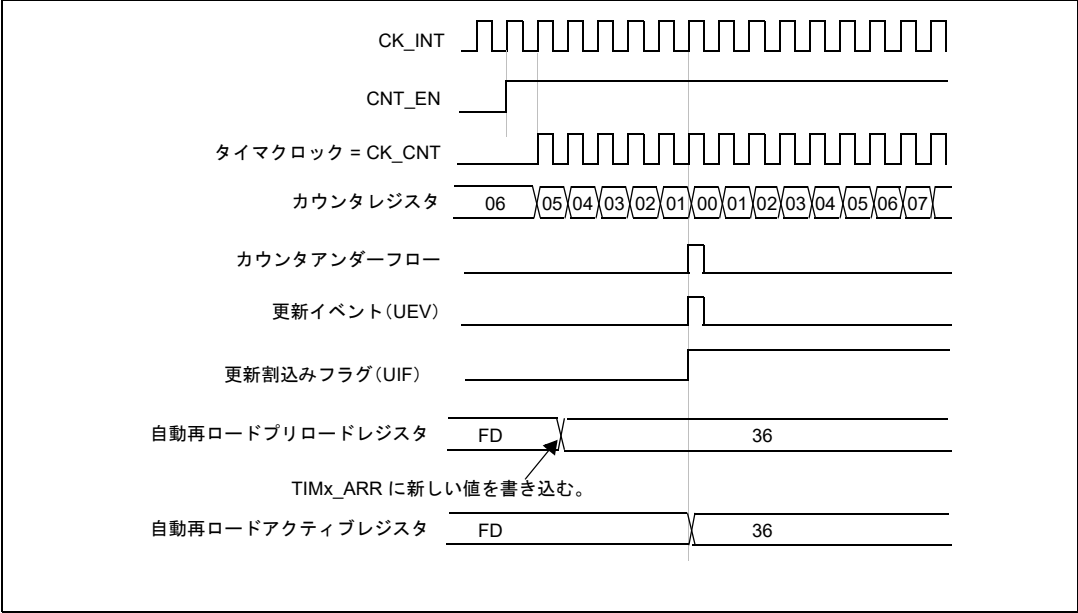
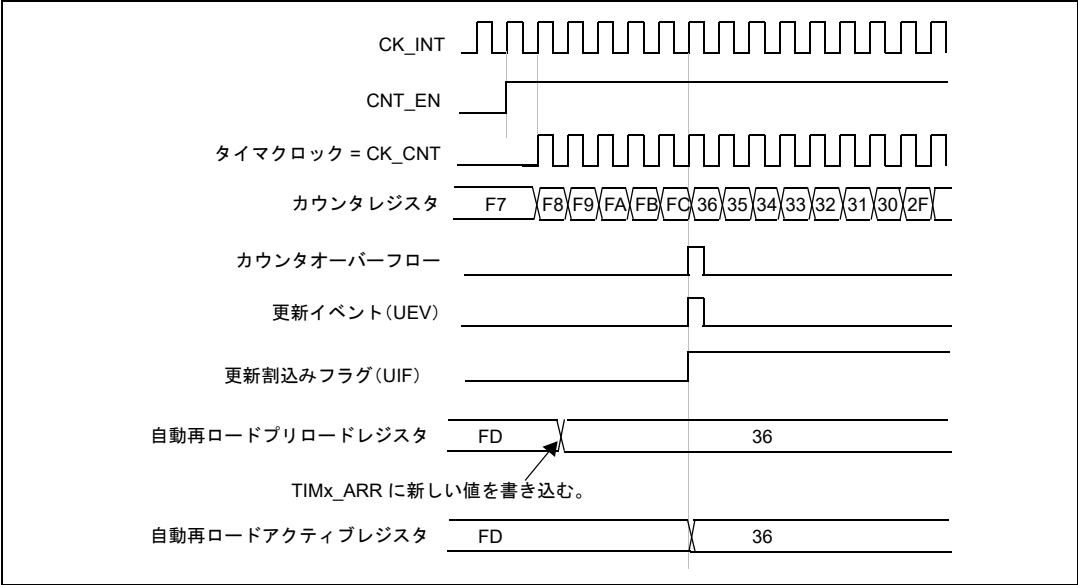


図 106. ARPE=1（カウンタオーバーフロー）の場合の更新イベント時のカウンタの
タイミング図



13.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

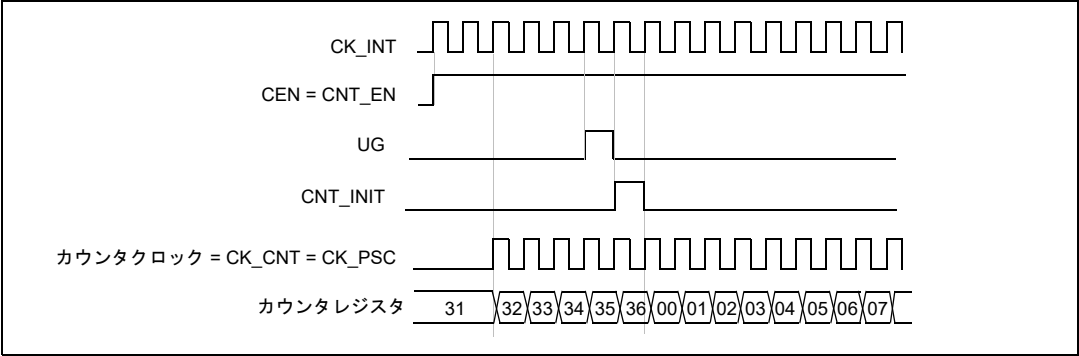
- 内部クロック (CK_INT)
- 外部クロックモード 1: 外部入力ピン (Tix)
- 外部クロックモード 2: 外部トリガ入力 (ETR) は、TIM2、TIM3、TIM4 のみで利用可能。
- 内部トリガ入力 (ITRx): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマがタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、341 ページの[タイマを別のタイマのプリスケアラとして使用する](#)を参照してください。

内部クロックソース (CK_INT)

スレーブモードコントローラが無効の場合 (TIMx_SMCR レジスタの SMS=000)、CEN、DIR ビット (TIMx_CR1 レジスタ) と UG ビット (TIMx_EGR レジスタ) が実際の制御ビットであり、ソフトウェアでのみ変更できます (自動的にクリアされたままの UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

図 107 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。

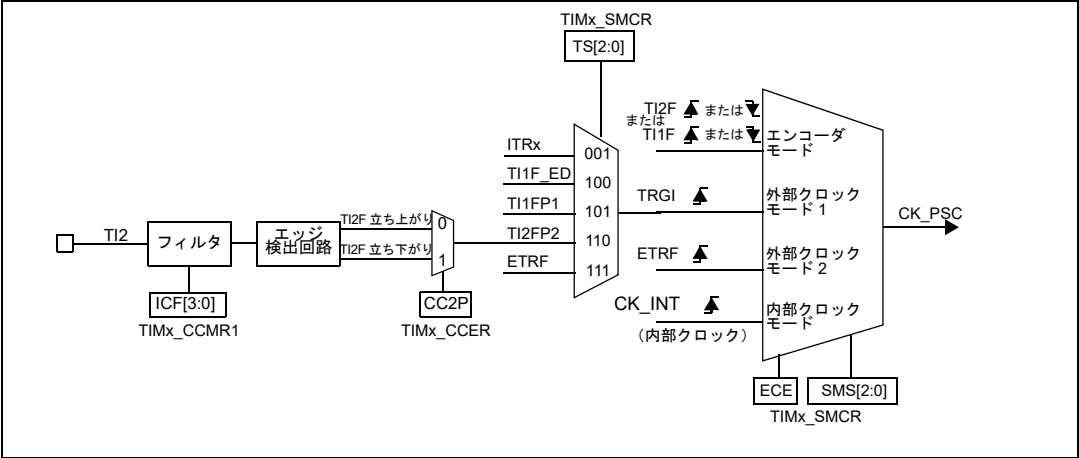
図 107. 内部クロック分周比 1 の場合の、通常モードの制御回路



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 108. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_CCMR1 レジスタに CC2S=01 を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの 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：

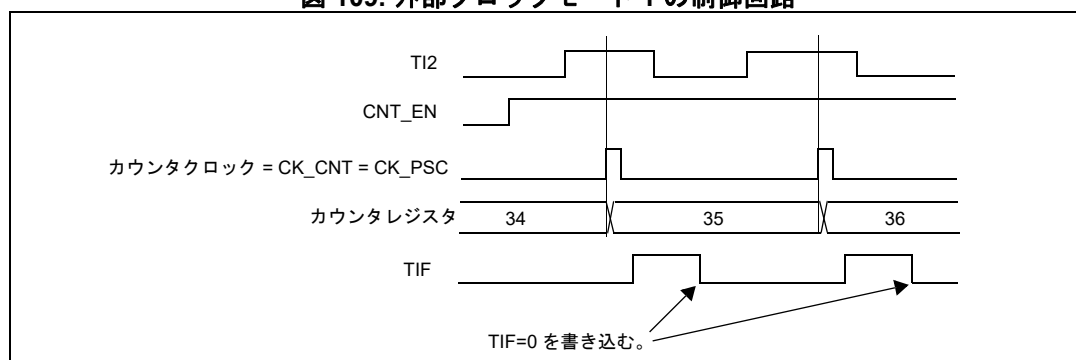
キャプチャプリスケアラはトリガには使用されないので、設定は不要です。

3. CC2P=0 と CC2NP=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウン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 109. 外部クロックモード 1 の制御回路



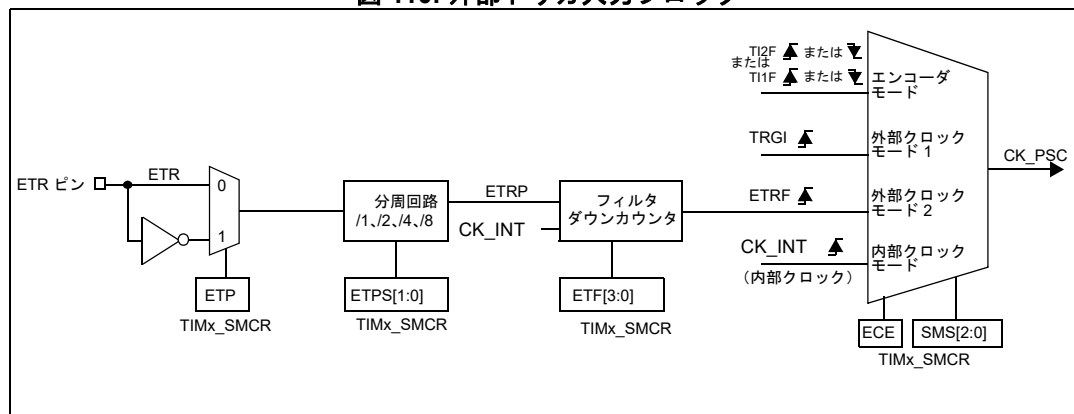
外部クロックソースモード 2

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 110 に、外部トリガ入力ブロックの概要を示します。

図 110. 外部トリガ入力ブロック



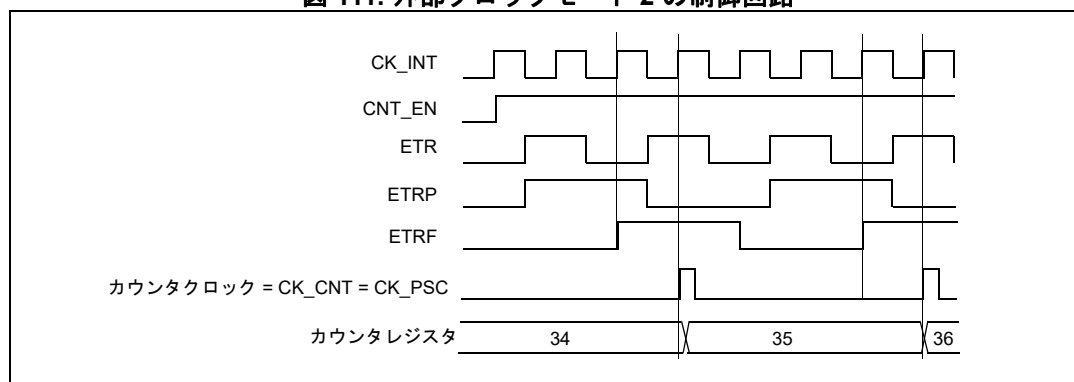
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

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 111. 外部クロックモード 2 の制御回路



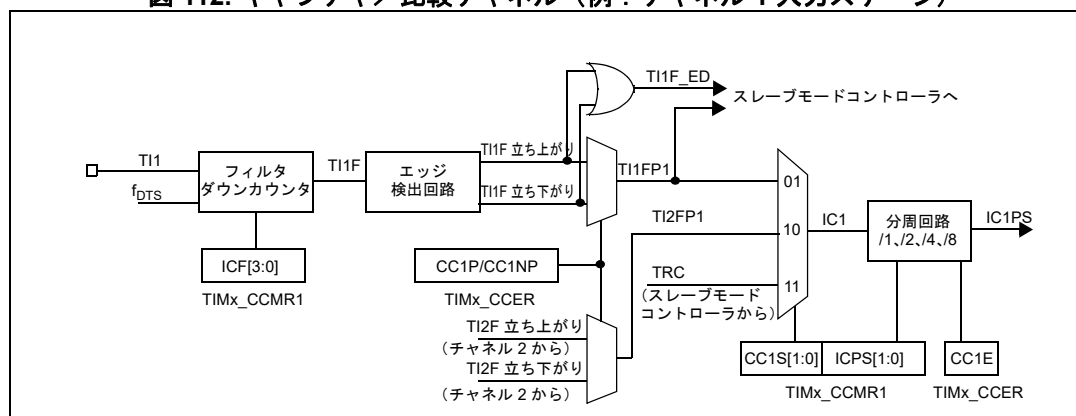
13.3.4 キャプチャ／比較チャンネル

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）、キャプチャの入力ステージ（デジタルフィルタ、マルチプレクス、プリスケアラ）、および出力ステージ（コンパレータと出力制御）から構成されています。

次の図に、キャプチャ／比較チャンネルの概要を示します。

入力ステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った TIxF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIxFPx) を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 112. 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。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 113. キャプチャ／比較チャンネル 1 メイン回路

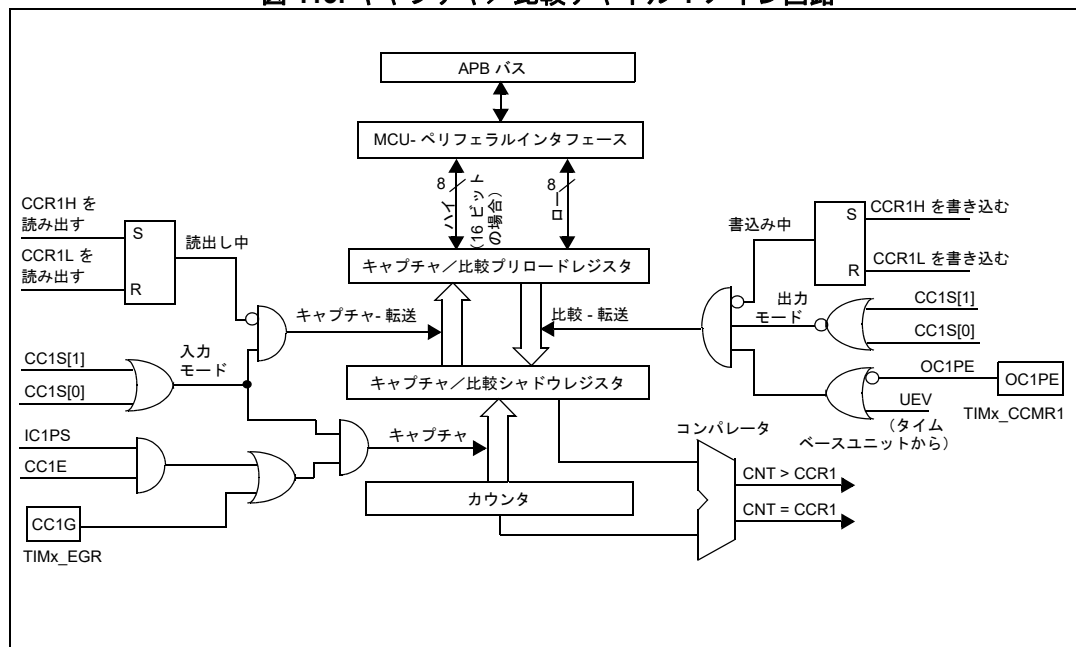
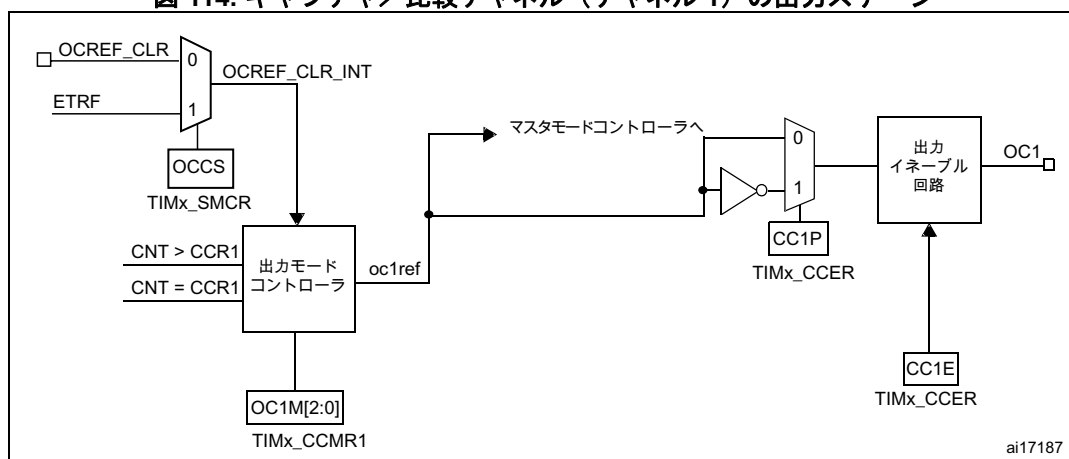


図 114. キャプチャ／比較チャンネル (チャンネル 1) の出力ステージ



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。

13.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ／比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力が入立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならないので、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、すぐにチャンネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に対して必要とする入力フィルタ時間を設定します (入力が TIx 入力の内の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 の遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“00”を書き込みます (この場合、立ち上がりエッジの選択)。
- 入力プリスケアラをプログラムします。今回の例では、有効な信号変化ごとにキャプチャを行いたいので、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 を書き込みます)。
- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。

- 必要な場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割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、フラグ読出し後、データ読出し前に発生するオーバーキャプチャの見落としを避けることができます。

注： IC 割込みと 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。

13.3.6 PWM 入力モード

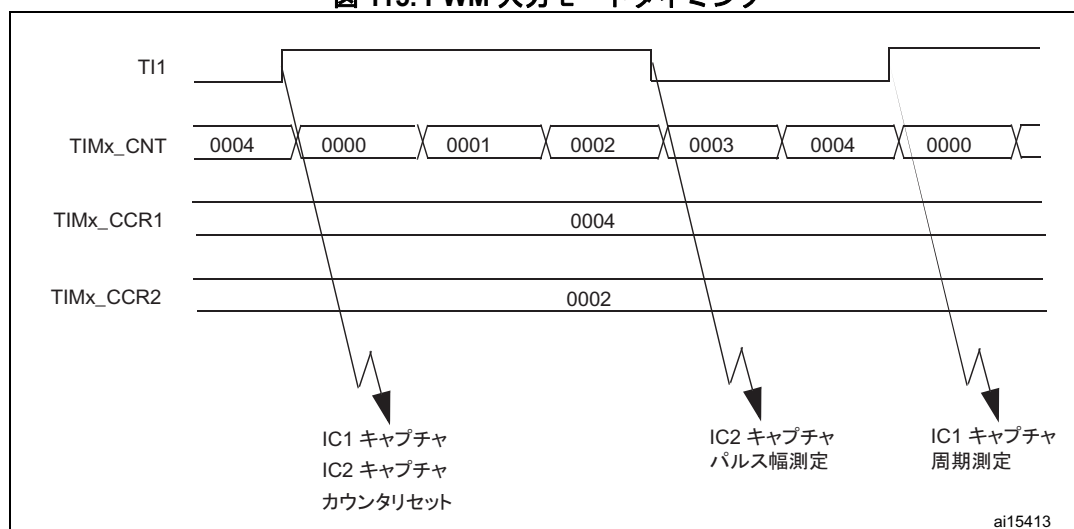
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2 つの ICx 信号が同じ Tlx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TlxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

- TIMx_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブ極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用します)。
- TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
- CC2P ビットに“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)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに 1 を書き込むことによって、キャプチャを有効にします。

図 115. PWM 入力モードタイミング



13.3.7 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。

13.3.8 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マブル値を割り当てます。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、割込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

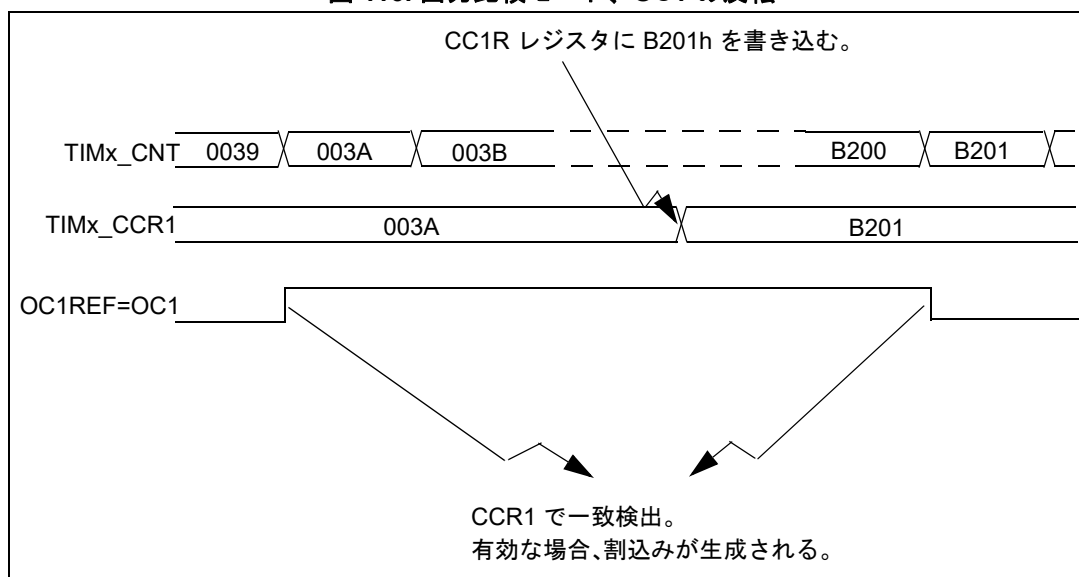
出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

手順 :

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエスト/DMA リクエストを生成する場合は、CCxIE ビット/CCxDE ビットをセットします。
4. 出力モードを選択します。たとえば、CNT が CCRx と一致したときに OCx 出力をトグルし、CCRx プリロードを使用せず、OCx が有効でアクティブハイのときには、OCxM=011、OCxPE=0、CCxP=0、CCxE=1 を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

いつでもソフトウェアで TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を 図 116 に示します。

図 116. 出力比較モード、OC1 の反転



13.3.9 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、個々のチャネル (OCx 出力ごとに PWM 1 波形) で、TIMx_CCMRx レジスタの OCxM ビットに "110" (PWM モード 1) や "111" (PWM モード 2) を書き込むことで、独自に選択できます。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx かどうかが判断されます (カウンタの方向によります)。ただし、ETRF (OCREF は、次の PWM 周期までは ETR 信号を通じて外部イベントによってクリアできる) に従って、OCREF 信号は次の場合にのみアサートされます。

- 比較結果が変化するとき、または
- 出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) が停止構成 (比較なし、OCxM=000) から PWM モードの 1 つ (OCxM=110 または 111) へ切り替えられたとき。

タイマの動作中は、ソフトウェアで強制的に PWM になります。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

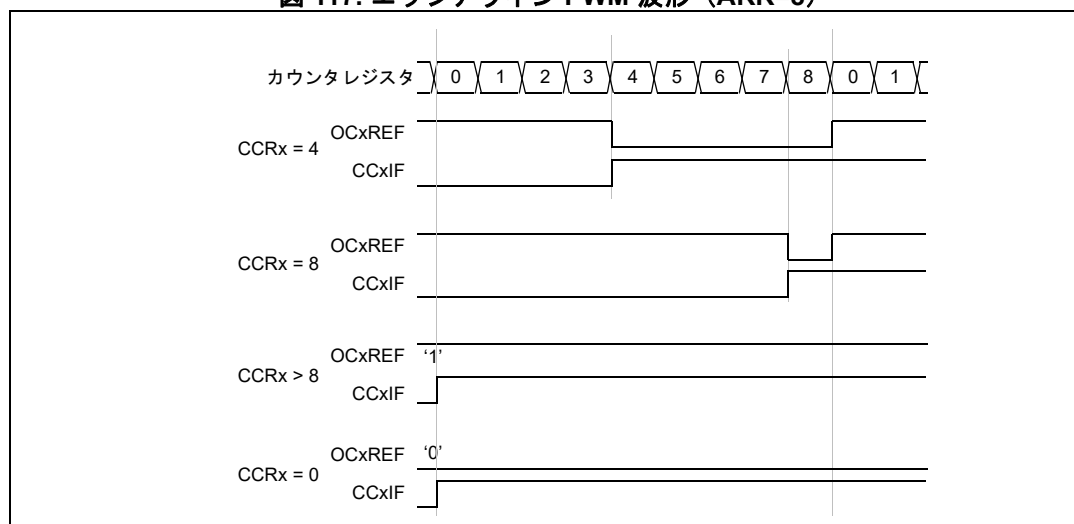
PWM エッジアラインモード

アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[セクション : 313 ページのアップカウントモード](#)を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”です。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 117](#) 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 117. 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。316 ページの[ダウンカウントモード](#)を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

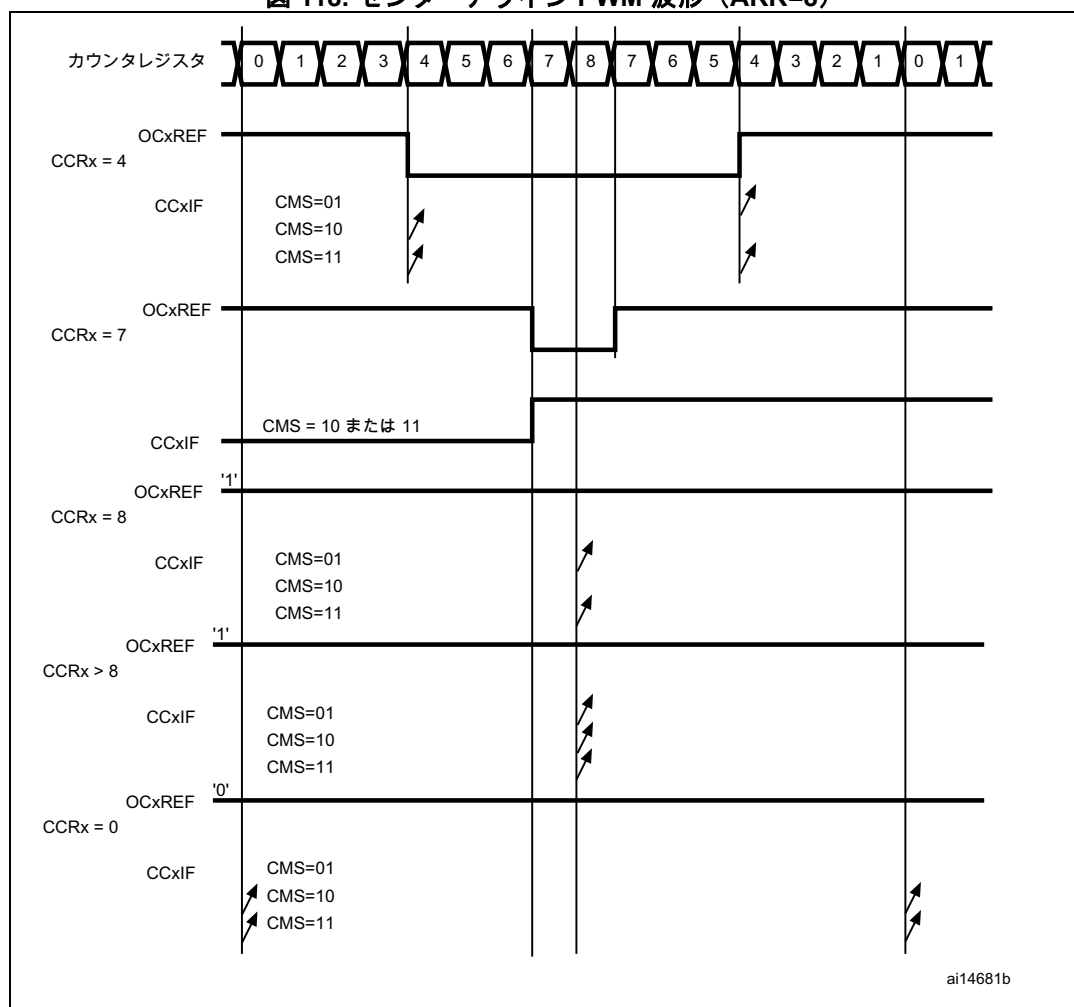
PWM センターアラインモード

センタアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです（その他すべての設定は、OCxRef/OCx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット（DIR）はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。318 ページの[センタアラインモード（アップ／ダウンカウント）](#)を参照してください。

図 118 に、次の条件でのセンタアライン 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 118. 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。

13.3.10 ワンパルスモード

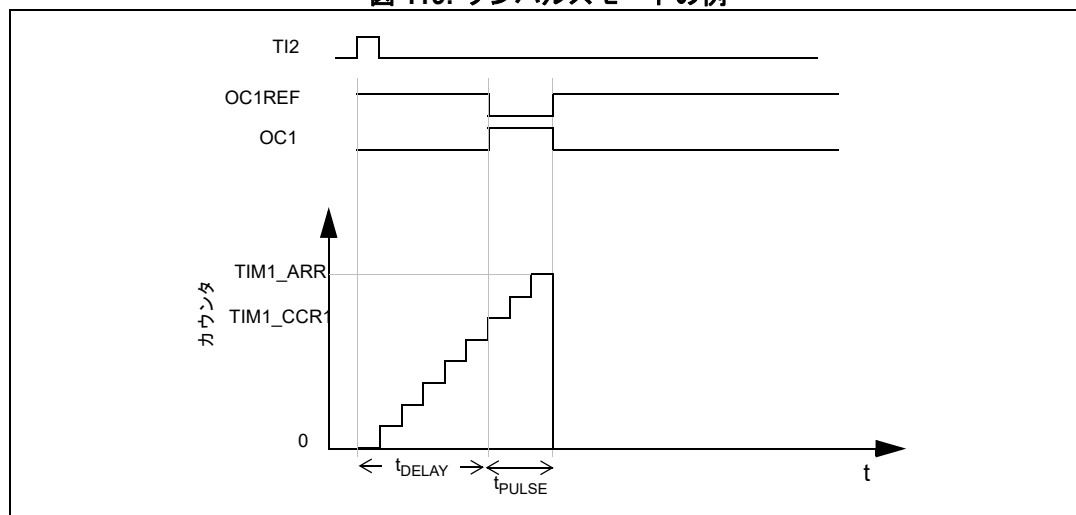
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 119. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタに $CC2S=01$ を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで $CC2P = 0$ と $CC2NP = 0$ を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として構成するために、TIMx_SMCR レジスタの $TS="110"$ を書き込みます。
- TI2FP2 を使用してカウンタを開始するために、TIMx_SMCR レジスタの SMS ビットに "110" (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 ($TIMx_ARR - TIMx_CCR + 1$) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの $OC1M=111$ を書き込むことによって、PWM モード 2 を有効にします。オプションで、TIMx_CCMR1 レジ

スタの OC1PE=1 と TIMx_CR1 レジスタの ARPE=1 を書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書込み、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。

13.3.11 外部イベントによる OCxREF 信号のクリア

指定されたチャンネルの OCxREF 信号は ETRF 入力をハイレベルとする（対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする）ことでローレベルにすることができます。OCxREF信号は、次の更新イベント UEV が発生するまで、ローレベルを保ちます。

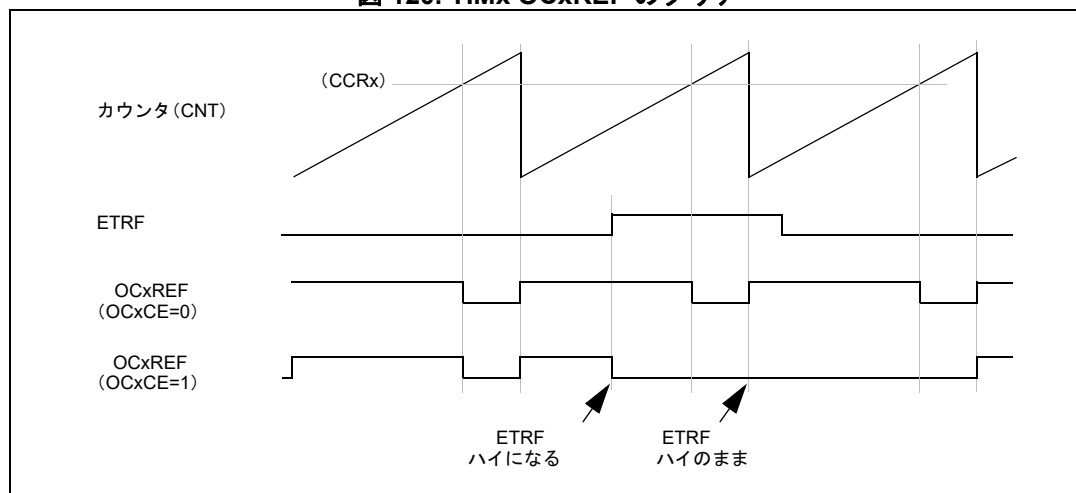
この機能は、出力比較モードと PWM モードでのみ使用でき、強制モードでは機能しません。

たとえば、ETR 信号をコンパレータの出力に接続して、電流処理に使用することができます。この場合、ETR は次のように設定する必要があります。

1. 外部トリガプリスケアラをオフに保つ必要があります。すなわち、TIMx_SMCR レジスタのビット ETPS[1:0] が 00 にクリアされます。
2. 外部クロックモード 2 を無効にする必要があります。すなわち、TIM1_SMCR レジスタのビット ECE が 0 にクリアされます。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて設定できます。

[図 120](#) 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 120. TIMx OCxREF のクリア



1. 100% デューティサイクルの PWM の場合 (CCRx>ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

13.3.12 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS="001"を、TI1 エッジのみをカウントしている場合は SMS="010"を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS="011"を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。

2つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。表 53 を参照してください。カウンタのクロックは、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択後は TI1 および TI2。フィルタされず、反転されない場合は TI1FP1=TI1、フィルタされず、反転されない場合は TI2FP2=TI2) の有効な変化によって駆動されます。ただし、カウンタ有効なことが前提となります (TIMx_CR1 レジスタの CEN ビットが"1")。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIRビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします (方向に応じて、0 から ARR まで、または ARR から 0 まで)。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、およびトリガ出力機能は、通常動作を続けます。

このモードでは、カウンタはインクリメンタルエンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します (TI1 と TI2 は同時に切り替わらないと想定しています)。

表 53. 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 121 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S="01" (TIMx_CCMR2 レジスタ、TI2FP2 は TI2 に配置)
- CC1P="0"、CC1NP="0"、IC1F="0000" (TIMx_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1)
- CC2P="0"、CC2NP="0"、IC2F="0000" (TIMx_CCER レジスタ、TI2FP2 非反転、TI2FP2=TI2)
- SMS="011" (TIMx_SMCR レジスタ、両方の入力が立ち上がりと立ち下りの両エッジでアクティブ)
- CEN=1 (TIMx_CR1 レジスタ、カウンタ有効)

図 121. 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

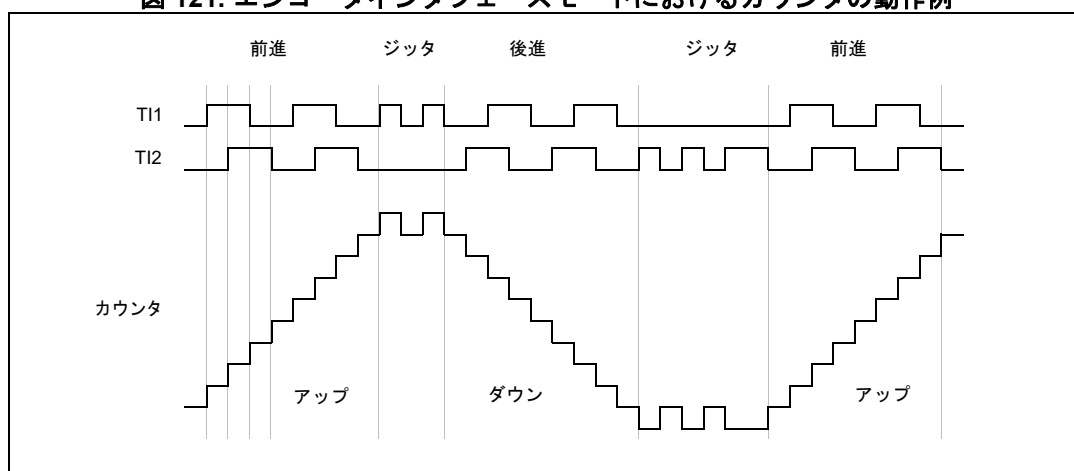
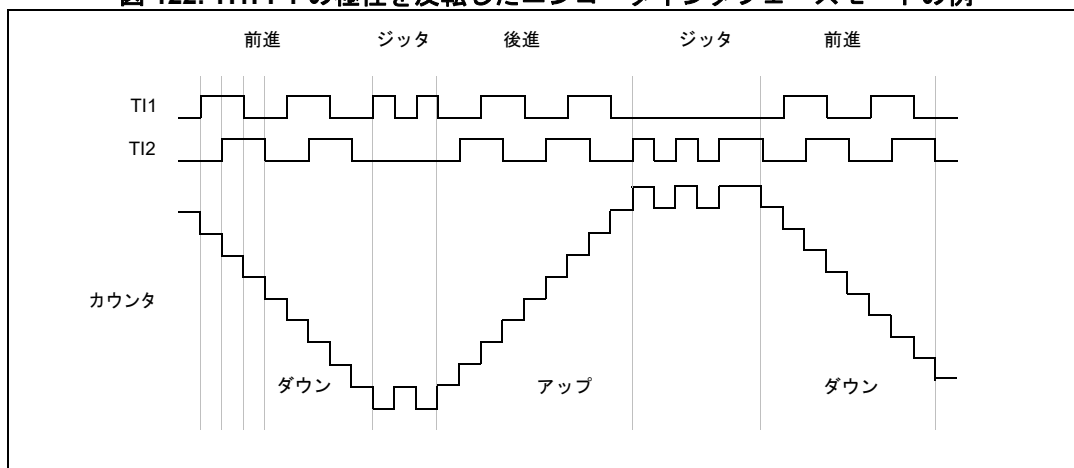


図 122 に、TI1FP1 の極性を反転したときのカウンタの動作を示します (上記と同じ設定ですが、CC1P=1)。

図 122. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

13.3.13 タイマ入力 XOR 機能

TIM_CR2 レジスタの TI1S ビットを使用して、チャンネル 1 の入力フィルタを、TIMx_CH1 から TIMx_CH3 までの 3 つの入力ピンを組み合わせた XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。

13.3.14 タイマと外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレーブモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

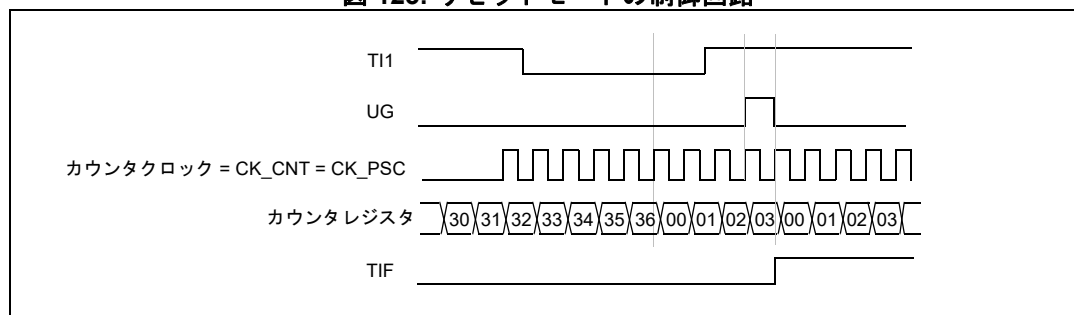
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割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 123. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

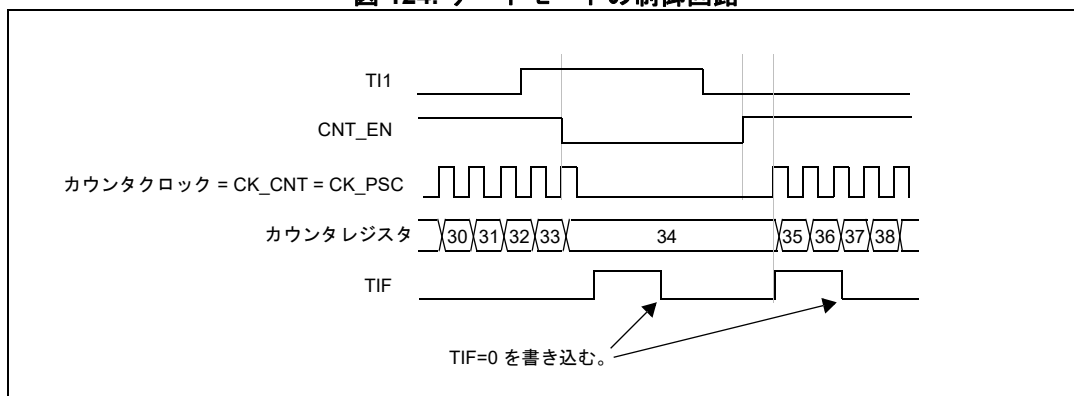
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01 ビット)。TIMx_CCER レジスタに CC1P=1 を書き込んで、極性を有効にします (そして、ローレベルのみを検出します)。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします (ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 124. 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。

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。

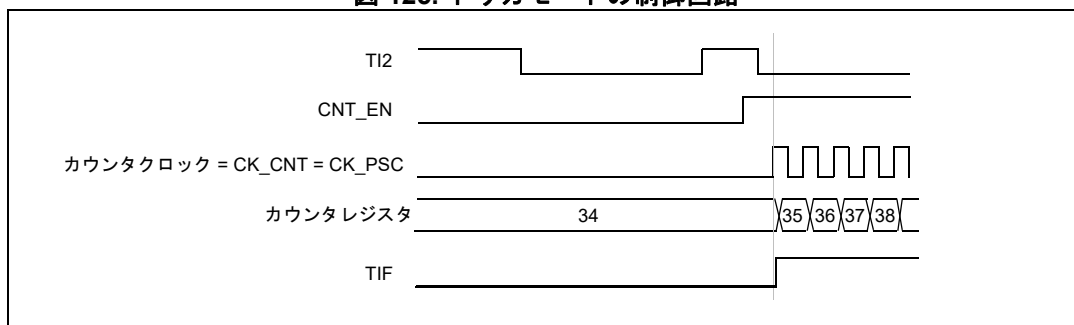
次の例では、アップカウンタは、TIM2 入力の立ち上がりエッジに応じて開始します。

- TIM2 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します（TIMx_CCMR1 レジスタの CC2S=01）。TIMx_CCER レジスタに CC2P=1 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TIM2 を選択します。

TIM2 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。

TIM2 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、TIM2 入力の再同期回路によるものです。

図 125. トリガモードの制御回路



スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます（ただし、外部クロックモード 1 とエンコーダモードは除きます）。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます（リセットモード、ゲートモード、およびトリガモードで動作している場合）。TIMx_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

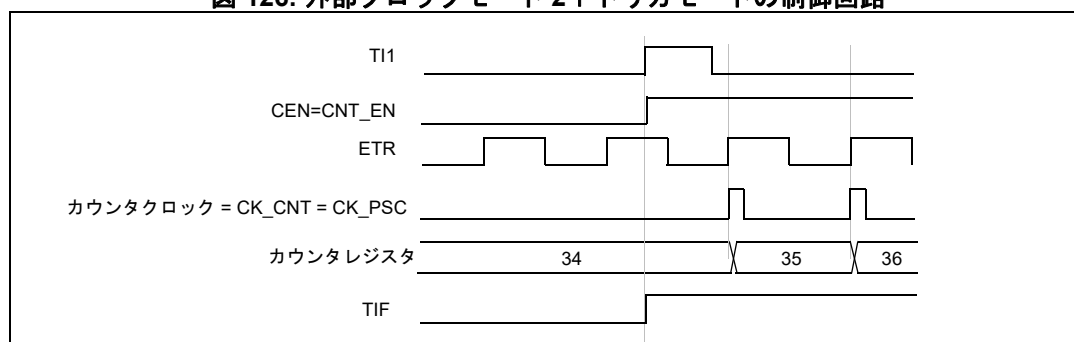
次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000：フィルタなし
 - ETPS = 00：プリスケアラは無効
 - ETP = 0：ETR の立ち上がりエッジを検出。ECE = 1 で外部クロックモード 2 を有効にする。
2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000：フィルタなし。
 - キャプチャプリスケアラはトリガには使用されないので、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタに CC1P=0 を書き込んで、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 126. 外部クロックモード 2+トリガモードの制御回路



13.3.15 タイマの同期

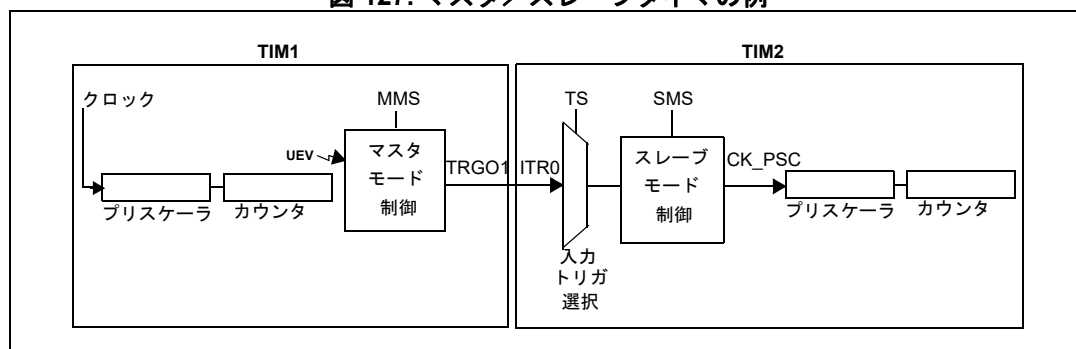
タイマの同期や連携した動作のために、TIMx タイマを内部で相互リンクすることができます。マスタモードに設定されたタイマは、スレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給を行うことができます。

図 127: マスタ/スレーブタイマの例に、トリガ選択およびマスタモード選択ブロックの概要を示します。

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。

タイマを別のタイマのプリスケラとして使用する

図 127. マスタ/スレーブタイマの例



たとえば、タイマ 1 がタイマ 2 のプリスケラとして動作するように設定できます。図 127 を参照してください。このためには、次の操作を行います。

- 更新イベント UEV ごとに定期的なトリガ信号を出力するように、タイマ 1 をマスタモードに設定します。TIM1_CR2 レジスタの MMS=010 を書き込んだ場合、更新イベントが生成されるたびに、TRGO1 で立ち上がりエッジが出力されます。
- タイマ 1 の出力 TRGO1 をタイマ 2 に接続し、タイマ 2 が内部トリガ ITR0 を使用するスレーブモードに設定します。このためには、TIM2_SMCR レジスタの TS ビットに 000 を書き込みます。
- 次に、スレーブモードコントローラを外部クロックモード 1 にします (TIM2_SMCR レジスタの SMS=111 を書き込みます)。これによりタイマ 2 は、タイマ 1 の周期的なトリガ信号の立ち上がりエッジ (タイマ 1 カウンタのオーバーフローに対応しています) をクロックとして動作します。
- 最後に、それぞれの CEN ビット (TIMx_CR1 レジスタ) をセットすることによって、両方のタイマを有効にする必要があります。

注: タイマ 1 のトリガ出力として OCx が選択された場合 (MMS=1xx)、その立ち上がりエッジがタイマ 2 カウンタのクロックとして使用されます。

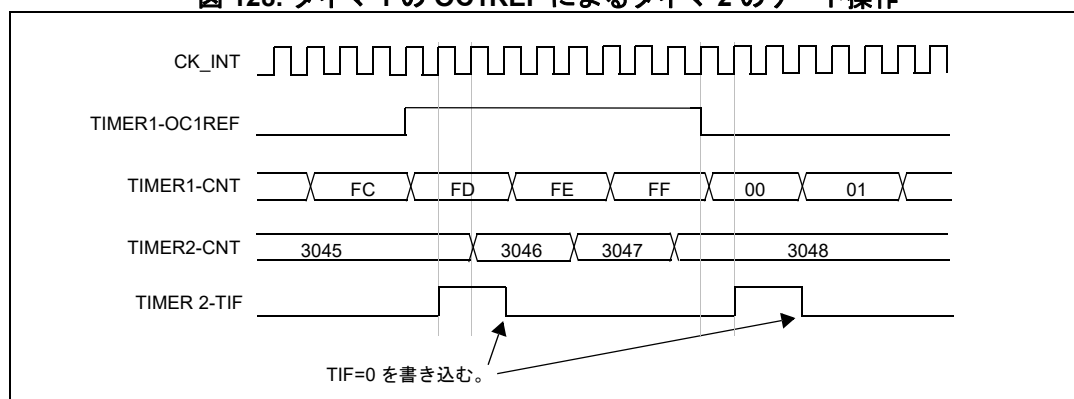
1つのタイマを使用して別のタイマを有効にする

この例では、タイマ 1 の出力比較 1 でタイマ 2 の有効／無効を制御します。接続については、[図 127](#) を参照してください。タイマ 2 は、タイマ 1 の OC1REF がハイのときにのみ、分周された内部クロックをカウントします。両方のカウンタクロック周波数は、CK_INT をプリスケアラで 3 分周したものです ($f_{CK_CNT} = f_{CK_INT}/3$)。

- タイマ 1 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=100)。
- タイマ 1 の OC1REF 波形を設定します (TIM1_CCMR1 レジスタ)。
- タイマ 2 がタイマ 1 から入カトリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をゲートモードに設定します (TIM2_SMCR レジスタの SMS=101)。
- CEN ビット (TIM2_CR1 レジスタ) に“1”を書き込んで、タイマ 2 を有効にします。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。

注： タイマ 2 のクロックはカウンタ 1 と同期しないので、このモードはタイマ 2 カウンタのイネーブル信号にのみ影響を与えます。

図 128. タイマ 1 の OC1REF によるタイマ 2 のゲート操作



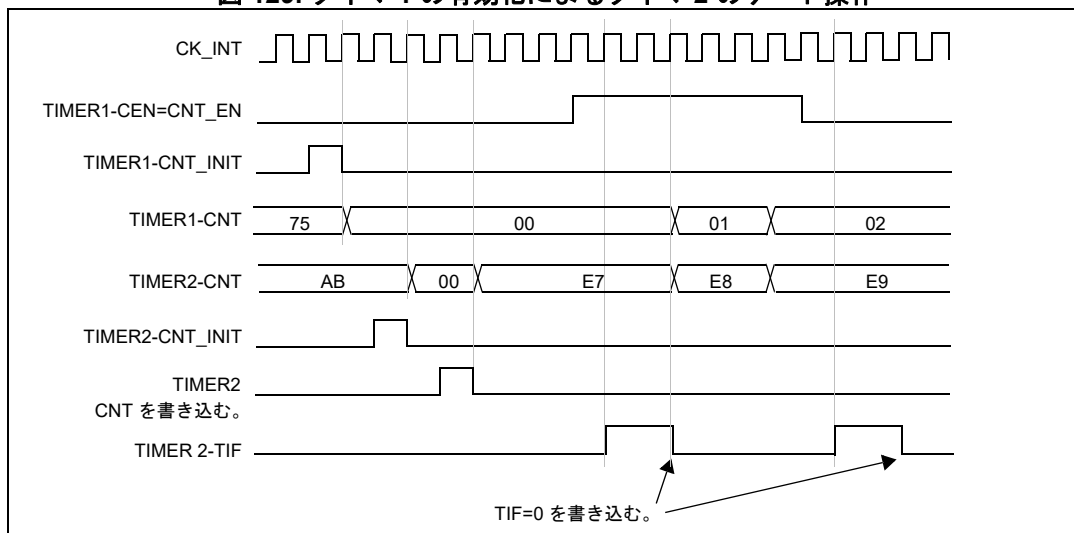
[図 162](#) の例では、タイマ 2 のカウンタとプリスケアラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。タイマ 1 を開始する前に両方のタイマをリセットすることによって、特定の値から開始することが可能です。この場合、タイマカウンタに任意の値を書き込むことができます。TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで容易にタイマをリセットできます。

次の例では、タイマ 1 とタイマ 2 を同期させます。タイマ 1 はマスタであり、0 からカウントを開始します。タイマ 2 はスレーブであり、0xE7 から開始します。プリスケアラの分周比は両方のタイマ

で同じです。TIM1_CR1 レジスタの CEN ビットに“0”を書き込むことによってタイマ 1 を無効にすると、タイマ 2 は停止します。

- タイマ 1 をマスタモードに設定して、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=100)。
- タイマ 1 の OC1REF 波形を設定します (TIM1_CCMR1 レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をゲートモードに設定します (TIM2_SMCR レジスタの SMS=101)。
- UG ビット (TIM1_EGR レジスタ) に“1”を書き込むことによって、タイマ 1 をリセットします。
- UG ビット (TIM2_EGR レジスタ) に“2”を書き込むことによって、タイマ 2 をリセットします。
- タイマ 2 のカウンタ (TIM2_CNT) に 0xE7 を書き込むことによって、タイマ 2 を 0xE7 に初期化します。
- CEN ビット (TIM2_CR1 レジスタ) に“1”を書き込んで、タイマ 2 を有効にします。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。
- CEN ビット (TIM1_CR1 レジスタ) に“0”を書き込むことによって、タイマ 1 を停止します。

図 129. タイマ 1 の有効化によるタイマ 2 のゲート操作



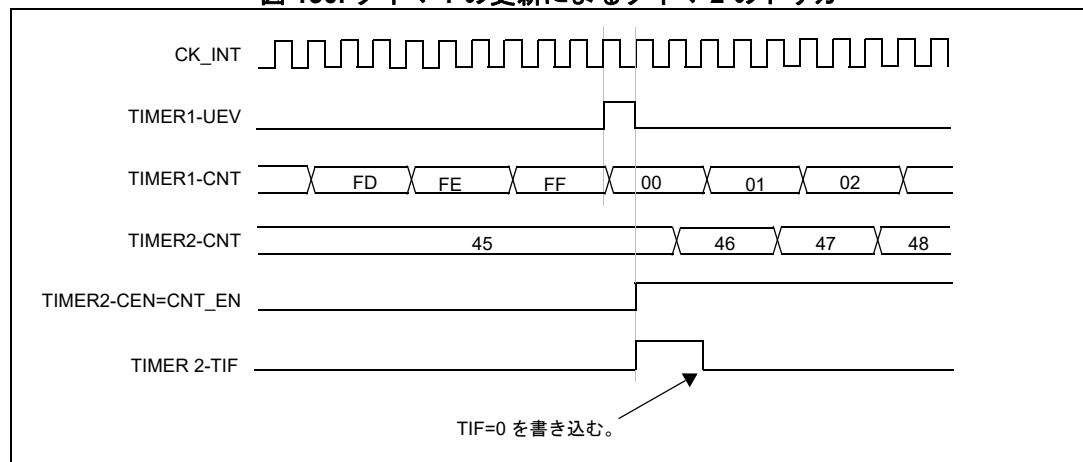
タイマを使用して別のタイマを開始する

この例では、タイマ 1 の更新イベントによってタイマ 2 の有効化を設定します。接続については、[図 127](#) 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 をプリスケアラで 3 分周したものです ($f_{CK_CNT} = f_{CK_INT}/3$)。

- タイマ 1 をマスタモードに設定して、その更新イベント (UEV) をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=010)。
- タイマ 1 の周期を設定します (TIM1_ARR レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をトリガモードに設定します (TIM2_SMCR レジスタの SMS=110)。

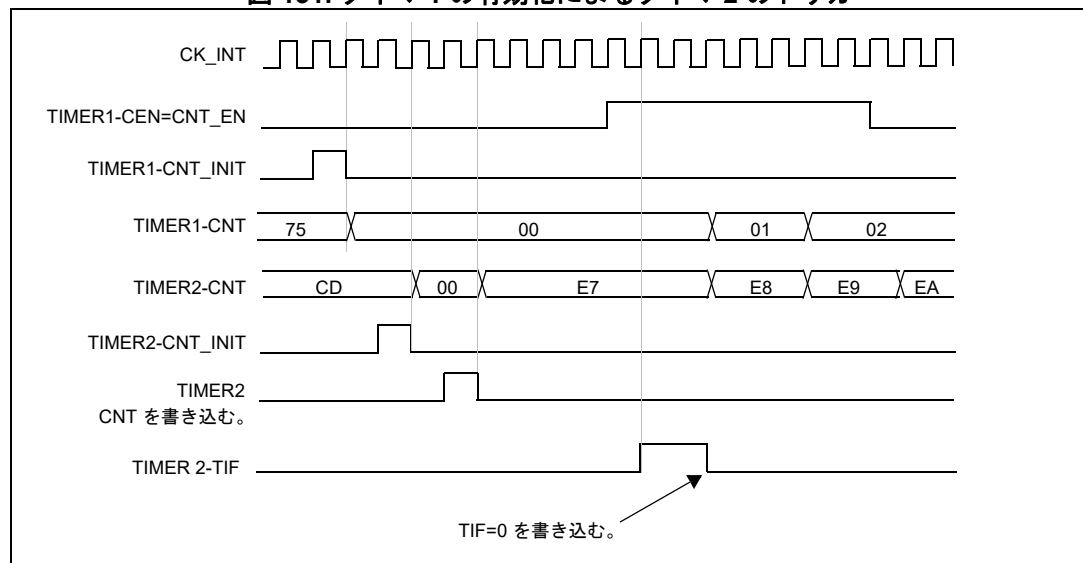
- CEN ビット（TIM1_CR1 レジスタ）に“1”を書き込んで、タイマ 1 を開始します。

図 130. タイマ 1 の更新によるタイマ 2 のトリガ



前の例と同じように、カウントを開始する前に両方のカウンタを初期化することができます。図 131 は 図 162 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 131. タイマ 1 の有効化によるタイマ 2 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。

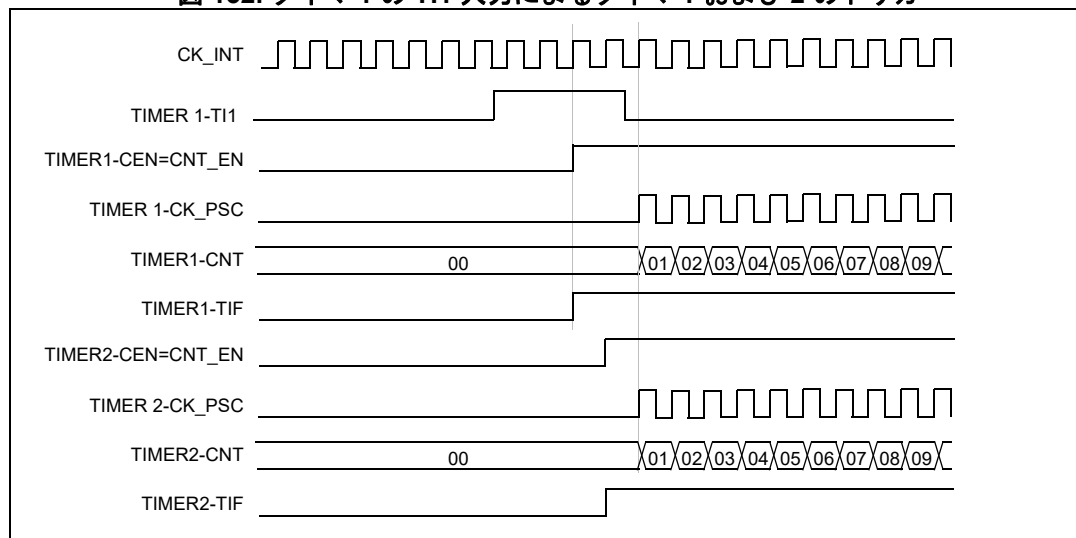
この例では、TI1 入力の立ち上がりエッジでタイマ 1 を有効にし、タイマ 1 が有効になると、タイマ 2 を有効にします。接続については、[図 127](#) を参照してください。カウンタの同時性を確保するため、タイマ 1 はマスタ/スレーブモードに設定する必要があります (TI1 に対してはスレーブ、タイマ 2 に対してはマスタ)。

- タイマ 1 をマスタモードに設定して、その有効化をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=001)。
- タイマ 1 をスレーブモードに設定して、TI1 から入力トリガを受け取るようにします (TIM1_SMCR レジスタの TS=100)。
- タイマ 1 をトリガモードに設定します (TIM1_SMCR レジスタの SMS=110)。
- MSM=1 (TIM1_SMCR レジスタ) を書き込むことによって、タイマ 1 をマスタ/スレーブモードに設定します。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をトリガモードに設定します (TIM2_SMCR レジスタの SMS=110)。

TI1 (タイマ 1) で立ち上がりエッジが発生すると、両方のカウンタが同時に内部クロックによるカウントを開始し、両方の TIF フラグがセットされます。

注： この例では、両方のタイマが開始前に初期化されます (それぞれの UG ビットをセットすることによって)。両方のカウンタは 0 から開始しますが、カウンタレジスタ (TIMx_CNT) に書き込むことによって、容易にオフセットを挿入できます。マスタ/スレーブモードでは、タイマ 1 の CNT_EN と CK_PSC の間に遅延が挿入されます。

図 132. タイマ 1 の TI1 入力によるタイマ 1 および 2 のトリガ



13.3.16 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex®-M4 コアは停止状態)、TIMx カウンタは、DBGMCU モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、停止します。詳細については、[セクション 23.16.2: タイマ、ウォッチドッグ、および I²C のデバッグサポート](#)を参照してください。

13.4 TIM2 から TIM5 のレジスタ

レジスタの説明で使用されている略語のリストについては、[34 ページのセクション 1.1](#) を参照してください。

32 ビットのペリフェラルレジスタには、ワード（32 ビット）単位で書き込む必要があります。他のすべてのペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

13.4.1 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
予約済み						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

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD**：クロック分周

このビットフィールドは、タイマクロック（CK_INT）周波数と、デジタルフィルタ（ETR、Tix）によって使用されるサンプリングクロックとの間の分周比を示します。

00: $t_{DTS} = t_{CK_INT}$
01: $t_{DTS} = 2 \times t_{CK_INT}$
10: $t_{DTS} = 4 \times t_{CK_INT}$
11: 予約済み

ビット 7 **ARPE**：自動再ロードプリロードイネーブル

0: TIMx_ARR レジスタはバッファされません。
1: TIMx_ARR レジスタはバッファされます。

ビット 6:5 **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

パッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケラは再初期化されます。

ビット 0 CEN : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注： 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

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。

13.4.2 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
予約済み								TI1S	MMS[2:0]			CCDS	予約済み		
								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 **TI1S** : TI1 選択

- 0 : TIMx_CH1 ピンが TI1 入力に接続されます。
- 1 : TIMx_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[2:0]** : マスタモード選択

- これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。
- 000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。
 - 001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。
 - カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。
 - 010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。
 - 011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされるとき (すでにハイであった場合も)、トリガ出力は正のパルスを送信します。 (TRGO)
 - 100 : **比較** - OC1REF 信号がトリガ出力 (TRGO) として使用されます。
 - 101 : **比較** - OC2REF 信号がトリガ出力 (TRGO) として使用されます。
 - 110 : **比較** - OC3REF 信号がトリガ出力 (TRGO) として使用されます。
 - 111 : **比較** - OC4REF 信号がトリガ出力 (TRGO) として使用されます。
- 注 :** スレーブタイマおよび 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。

13.4.3 TIMx スレーブモード制御レジスタ（TIMx_SMCR）

アドレスオフセット：0x08

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]				予約済み	SMS[2:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

ビット 15 **ETP**：外部トリガ極性

このビットは、ETR と ETR のいずれがトリガ動作に使用されるかを選択します。
0：ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブになります。
1：ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブになります。

ビット 14 **ECE**：外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。
0：外部クロックモード 2 は無効です。
1：外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。
1：ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります（SMS=111、TS=111）。
2：外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません（TS ビットが 111 でないことが必要）。
3：外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力 は ETRF です。

ビット 13:12 **ETPS**：外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、最大でも CK_INT 周波数の 1/4 でなければなりません。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。
00：プリスケアラオフ
01：ETRP 周波数は 2 分周されます。
10：ETRP 周波数は 4 分周されます。
11：ETRP 周波数は 8 分周されます。

ビット 11:8 **ETF[3:0]**：外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、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 個のイベント発生が必要です。
0000：フィルタなし、 f_{DTS} でサンプリング
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)

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 54:351 ページのTIMx 内部トリガ接続](#) を参照してください。

注： 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

001 : エンコーダモード 1 - カウンタは、TI1FP2 のレベルに応じて、TI2FP1 のエッジでカウントアップ/ダウンします。

010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。

011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。

100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。

110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。

111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

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 54. 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_TRGO	予約済み	TIM3_TRGO	TIM4_TRGO
TIM3	TIM1_TRGO	TIM2_TRGO	TIM5_TRGO	TIM4_TRGO
TIM4	TIM1_TRGO	TIM2_TRGO	TIM3_TRGO	予約済み
TIM5	TIM2_TRGO	TIM3_TRGO	TIM4_TRGO	予約済み

13.4.4 TIMx DMA／割込み有効レジスタ（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
予約済み	TDE	予約済み	CC4DE	CC3DE	CC2DE	CC1DE	UDE	予約済み	TIE	予約済み	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE**：トリガ DMA リクエストイネーブル

0：トリガ DMA リクエストは無効です。

1：トリガ DMA リクエストは有効です。

ビット 13 予約済み。常に 0 として読み出されます。

ビット 12 **CC4DE**：キャプチャ／比較 4 DMA リクエストイネーブル

0：CC4 DMA リクエストは無効です。

1：CC4 DMA リクエストは有効です。

ビット 11 **CC3DE**：キャプチャ／比較 3 DMA リクエストイネーブル

0：CC3 DMA リクエストは無効です。

1：CC3 DMA リクエストは有効です。

ビット 10 **CC2DE**：キャプチャ／比較 2 DMA リクエストイネーブル

0：CC2 DMA リクエストは無効です。

1：CC2 DMA リクエストは有効です。

ビット 9 **CC1DE**：キャプチャ／比較 1 DMA リクエストイネーブル

0：CC1 DMA リクエストは無効です。

1：CC1 DMA リクエストは有効です。

ビット 8 **UDE**：更新 DMA リクエストイネーブル

0：更新 DMA リクエストは無効です。

1：更新 DMA リクエストは有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE**：トリガ割込みイネーブル

0：トリガ割込みは無効です。

1：トリガ割込みは有効です。

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 割込みイネーブル

0：CC4 割込みは無効です。

1：CC4 割込みは有効です。

- ビット 3 **CC3IE** : キャプチャ／比較 3 割込みイネーブル
0 : CC3 割込みは無効です。
1 : CC3 割込みは有効です。
- ビット 2 **CC2IE** : キャプチャ／比較 2 割込みイネーブル
0 : CC2 割込みは無効です。
1 : CC2 割込みは有効です。
- ビット 1 **CC1IE** : キャプチャ／比較 1 割込みイネーブル
0 : CC1 割込みは無効です。
1 : CC1 割込みは有効です。
- ビット 0 **UIE** : 更新割込みイネーブル
0 : 更新割込みは無効です。
1 : 更新割込みは有効です。

13.4.5 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
予約済み	CC4OF				CC3OF	CC2OF	CC1OF	予約済み	TIF	予約済み	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	rc_w0				rc_w0	rc_w0	rc_w0		rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

- ビット 15:13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **CC4OF** : キャプチャ／比較 4 オーバーキャプチャフラグ
CC1OF の説明を参照してください。
- ビット 11 **CC3OF** : キャプチャ／比較 3 オーバーキャプチャフラグ
CC1OF の説明を参照してください。
- ビット 10 **CC2OF** : キャプチャ／比較 2 オーバーキャプチャフラグ
CC1OF の説明を参照してください。
- ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ
このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。ソフトウェアで“0”を書き込むことによってクリアされます。
0 : オーバキャプチャは検出されていません。
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。
- ビット 8:7 予約済みであり、リセット値に保持する必要があります。
- ビット 6 **TIF** : トリガ割込みフラグ
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 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。
- オーバーフローまたはアンダーフロー時（TIM2 から TIM5）、および TIMx_CR1 レジスタの UDIS=0 である場合。
- TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイイベントによって 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）。

13.4.6 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
予約済み									TG	予約済み	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。

13.4.7 TIMx キャプチャ／比較モードレジスタ 1（TIMx_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

出力比較モード

ビット 15 **OC2CE**：出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]**：出力比較 2 モード

ビット 11 **OC2PE**：出力比較 2 プリロードイネーブル

ビット 10 **OC2FE**：出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]**：キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC2 チャンネルは出力として設定されます。

01：CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10：CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11：CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC2S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC2E=0）のときにのみ書き込み可能です。

ビット 7 **OC1CE**：出力比較 1 クリアイネーブル

OC1CE：出力比較 1 クリアイネーブル

0：OC1Ref は ETRF 入力の影響を受けません。

1：OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合は非アクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注 : PWM モード 1 または 2 では、比較結果が変化したとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2: PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から 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) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング

0001 : $f_{SAMPLING} = f_{CK_INT}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{CK_INT}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{CK_INT}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E=0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

13.4.8 TIMx キャプチャ／比較モードレジスタ 2（TIMx_CCMR2）

アドレスオフセット：0x1C

リセット値：0x0000

上記の CCMR1 レジスタの説明を参照してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 15 **OC4CE**：出力比較 4 クリアイネーブル

ビット 14:12 **OC4M**：出力比較 4 モード

ビット 11 **OC4PE**：出力比較 4 プリロードイネーブル

ビット 10 **OC4FE**：出力比較 4 高速イネーブル

ビット 9:8 **CC4S**：キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC4 チャンネルは出力として設定されます。

01：CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10：CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11：CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注：**CC4S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC4E=0）のときにのみ書き込み可能です。

ビット 7 **OC3CE**：出力比較 3 クリアイネーブル

ビット 6:4 **OC3M**：出力比較 3 モード

ビット 3 **OC3PE**：出力比較 3 プリロードイネーブル

ビット 2 **OC3FE**：出力比較 3 高速イネーブル

ビット 1:0 **CC3S**：キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00：CC3 チャンネルは出力として設定されます。

01：CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10：CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11：CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注：**CC3S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC3E=0）のときにのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ

ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケアラ

ビット 9:8 **CC4S** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC4 チャンネルは出力として設定されます。

01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。

10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。

11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ

ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケアラ

ビット 1:0 **CC3S** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC3 チャンネルは出力として設定されます。

01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。

10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。

11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC3S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。

13.4.9 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	予約済み	CC4P	CC4E	CC3NP	予約済み	CC3P	CC3E	CC2NP	予約済み	CC2P	CC2E	CC1NP	予約済み	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

ビット 15 **CC4NP** : キャプチャ／比較 4 出力極性

CC1NP の説明を参照してください。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **CC4P** : キャプチャ／比較 4 出力極性

CC1P の説明を参照してください。

ビット 12 **CC4E** : キャプチャ／比較 4 出力イネーブル。

CC1E の説明を参照してください。

ビット 11 **CC3NP** : キャプチャ／比較 3 出力極性

CC1NP の説明を参照してください。

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC3P** : キャプチャ／比較 3 出力極性

CC1P の説明を参照してください。

ビット 8 **CC3E** : キャプチャ／比較 3 出力イネーブル。

CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性

CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P** : キャプチャ／比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ／比較 2 出力イネーブル。

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

この場合、CC1NP はクリアされたままでなければなりません。

CC1 チャンネルが入力として設定されている場合 :

このビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。

00 : 非反転／立ち上がりエッジ

回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。

01 : 反転／立ち下がりエッジ

回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転／両エッジ

回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル。

CC1 チャンネルが出力として設定されている場合 :

0 : オフ - OC1 はアクティブではありません。

1 : オン - OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ／比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

表 55. 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 レジスタに依存します。

13.4.10 TIMx カウンタ（TIMx_CNT）

アドレスオフセット：0x24

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[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
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 15:0 CNT[15:0]：カウンタ値

13.4.11 TIMx プリスケアラ（TIMx_PSC）

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
r/w	r/w	r/w	r/w	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。

13.4.12 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：0x 0000

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 15:0 **ARR[15:0]**：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 13.3.1：311 ページのタイムベースユニット](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

13.4.13 TIMx キャプチャ／比較モードレジスタ 1（TIMx_CCR1）

アドレスオフセット：0x34

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR1[31:16]**：キャプチャ／比較 1 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値下位ビット

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

13.4.14 TIMx キャプチャ／比較モードレジスタ 2（TIMx_CCR2）

アドレスオフセット：0x38

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR2[31:16]**：キャプチャ／比較 2 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR2[15:0]**：キャプチャ／比較 2 値下位ビット

CC2 チャンネルが出力として設定されている場合：

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合：

CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。

13.4.15 TIMx キャプチャ／比較モードレジスタ 3（TIMx_CCR3）

アドレスオフセット：0x3C

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR3[31:16]**：キャプチャ／比較 3 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR3[15:0]**：キャプチャ／比較値下位ビット

CC3 チャンネルが出力として設定されている場合：

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC3 出力に送信される値を含みます。

CC3 チャンネルが入力として設定されている場合：

CCR3 は、最後の入力キャプチャ 3 イベント（IC3）によって転送されたカウンタ値です。

13.4.16 TIMx キャプチャ／比較モードレジスタ 4（TIMx_CCR4）

アドレスオフセット：0x40

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]（タイマに依存）															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR4[31:16]**：キャプチャ／比較 4 値上位ビット（TIM2 および TIM5）

ビット 15:0 **CCR4[15:0]**：キャプチャ／比較値下位ビット

1. CC4 チャンネルが出力として設定されている場合（CC4S ビット）：
CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。
TIMx_CCMR レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。
アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC4 出力に送信される値を含みます。
2. CC4 チャンネルが入力として設定されている場合（TIMx_CCMR4 レジスタの CC4S ビット）：
CCR4 は、最後の入力キャプチャ 4 イベント（IC4）によって転送されたカウンタ値です。

13.4.17 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
予約済み	DBL[4:0]					予約済み					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 ビットのベクタは、DMA 転送回数（タイマは、TIMx_DMAR アドレスに対して読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）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出し／書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。

13.4.18 完全転送の TIMx DMA アドレス（TIMx_DMAR）

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **DMAB[15:0]** : DMA パーストアクセスレジスタ

DMAR レジスタへの読出しまたは書込み動作は、次のアドレスにあるレジスタへのアクセスとなります：

$(\text{TIMx_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です（DBL は TIMx_DCR 内で設定）。

DMA パースト機能の使用例

この例では、タイマ DMA パースト機能を使って CCRx レジスタ値を更新します（x = 2、3、4）。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

- 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
- DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 転送、DBA = 0xE。
- TIMx 更新 DMA リクエストを有効にします（DIER レジスタのUDE ビットをセット）。
- TIMx を有効化
- DMA チャンネルを有効化注：

注： この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送さ

れます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

13.4.19 TIM2 オプションレジスタ (TIM2_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				ITR1_RMP		予約済み									
				rw	rw										

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:10 **ITR1_RMP** : 内部トリガ 1 再配置

ソフトウェアでセット／クリアされます。

00 : 予約済み

01 : PTP トリガ出力は TIM2_ITR1 に接続されます。

10 : OTG FS SOF は TIM2_ITR1 入力に接続されます。

11 : OTG HS SOF は TIM2_ITR1 入力に接続されます。

ビット 9:0 予約済みであり、リセット値に保持する必要があります。

13.4.20 TIM5 オプションレジスタ (TIM5_OR)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								TI4_RMP		予約済み					
								rw	rw						

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:6 **TI4_RMP** : タイマ 入力 4 再配置

ソフトウェアでセット／クリアされます。

00 : TIM5 チャンネル 4 を GPIO に接続 : データシートの代替機能配置表を参照してください。

01 : LSI の内部クロックは較正のために TIM5_CH4 入力へ接続

10 : LSE の内部クロックは較正のために TIM5_CH4 入力へ接続

11 : RTC ウェイクアップ割込みは較正のために TIM5_CH4 入力へ接続ウェイクアップ割込みを有効化する必要があります。

ビット 5:0 予約済みであり、リセット値に保持する必要があります。

13.4.21 TIMx レジスタマップ

TIMx レジスタは、次の表のようにマップされます。

表 56. TIM2 から TIM5 のレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
0x00	TIMx_CR1	予約済み																						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	0														
0x04	TIMx_CR2	予約済み																						TIS		MMS[2:0]		CCDS		予約済み																			
	リセット値																							0	0	0	0	0																					
0x08	TIMx_SMCR	予約済み																ETP		ECE		ETPS [1:0]		ETF[3:0]		MSM		TS[2:0]		予約済み		SMS[2:0]																	
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0			0	0																
0x0C	TIMx_DIER	予約済み																		TDE		COMDE		CC4DE		CC3DE		CC2DE		CC1DE		UDE		予約済み		TIE		予約済み		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	0			0	0	0	0	0	0	0	0	0	0
0x10	TIMx_SR	予約済み																		CC4OF		CC3OF		CC2OF		CC1OF		予約済み		TIF		予約済み		CC4IF		CC3IF		CC2IF		CC1IF		UIF							
	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	TIMx_EGR	予約済み																						TG		予約済み		CC4G		CC3G		CC2G		CC1G		UG													
	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
0x18	TIMx_CCMR1 出力比較モード	予約済み																OC2CE		OC2M [2:0]		OC2PE		OC2FE		CC2S [1:0]		OC1CE		OC1M [2:0]		OC1PE		OC1FE		CC1S [1:0]													
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	予約済み																IC2F[3:0]		IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]		IC1 PSC [1:0]		CC1S [1:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1C	TIMx_CCMR2 出力比較モード	予約済み																OC24CE		OC4M [2:0]		OC4PE		OC4FE		CC4S [1:0]		OC3CE		OC3M [2:0]		OC3PE		OC3FE		CC3S [1:0]													
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	予約済み																IC4F[3:0]		IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]		IC3 PSC [1:0]		CC3S [1:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	TIMx_CCER	予約済み																CC4NP 予約済み		CC4P		CC4E		CC3NP		予約済み		CC3P		CC3E		CC2NP		予約済み		CC2P		CC2E		CC1NP		予約済み		CC1P		CC1E			
	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
0x24	TIMx_CNT	CNT[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CNT[15:0]																															
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
0x28	TIMx_PSC	予約済み																PSC[15:0]																															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIMx_ARR	ARR[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																ARR[15:0]																															
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										

表 56. TIM2 から TIM5 のレジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x30	予約済み																																		
0x34	TIMx_CCR1	CCR1[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR1[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x38	TIMx_CCR2	CCR2[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR2[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x3C	TIMx_CCR3	CCR3[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR3[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x40	TIMx_CCR4	CCR4[31:16] (TIM2 および TIM5 のみ。他のタイマについては予約済み。)																CCR4[15:0]																	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x44	予約済み																																		
0x48	TIMx_DCR	予約済み																DBL[4:0]				予約済み				DBA[4:0]									
	リセット値																	0	0	0	0					0	0	0	0	0	0	0	0	0	0
0x4C	TIMx_DMAR	予約済み																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	0
0x50	TIM2_OR	予約済み																予約済み		ITR1_RMP		予約済み													
	リセット値																			0	0														
0x50	TIM5_OR	予約済み																予約済み						IT4_RMP		予約済み									
	リセット値																							0	0										

レジスタ境界アドレスについては[38 ページの表 1](#)を参照してください。

14 汎用タイマ (TIM9 から TIM11)

TIM12、TIM13 および TIM14 は、STM32F401xB/C および STM32F401xD/E で使用できません。

14.1 TIM9/10/11 の概要

TIM9/10/11 汎用タイマは、プログラム可能なプリスケアラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

これらのカウンタは、入力信号パルス長の測定（入力キャプチャ）や出力波形の生成（出力比較、PWM）など様々な目的に使用できます。

パルス幅と波形の周期は、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、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。

TIM9/10/11 タイマは完全に独立していて、いかなるリソースも共用しません。これらのタイマは、[セクション 14.3.12](#) に示すように、相互に同期させることができます。

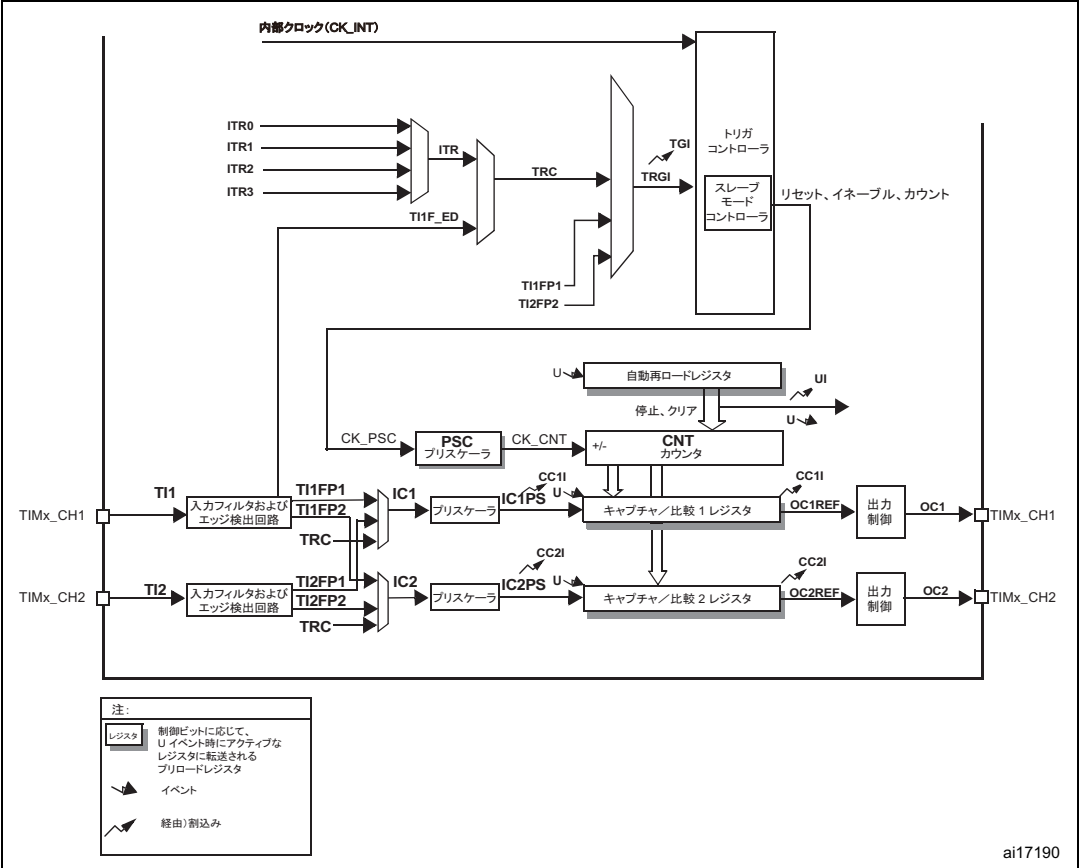
14.2 TIM9/10/11 の主な機能

14.2.1 TIM9 の主な機能

汎用タイマ TIM9 の機能は以下のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケアラ（動作中に変更可能）。
- 次の機能を持つ、最大 2 つの独立チャンネル。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジアラインモード）
 - 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込み生成。
 - 更新：カウンタオーバーフロー、カウンタ初期化（ソフトウェアまたは内部トリガによる）
 - トリガイベント（内部トリガによるカウンタの開始、停止、初期化、またはカウント）
 - 入力キャプチャ
 - 出力比較

図 133. 汎用タイマのブロック図 (TIM9)

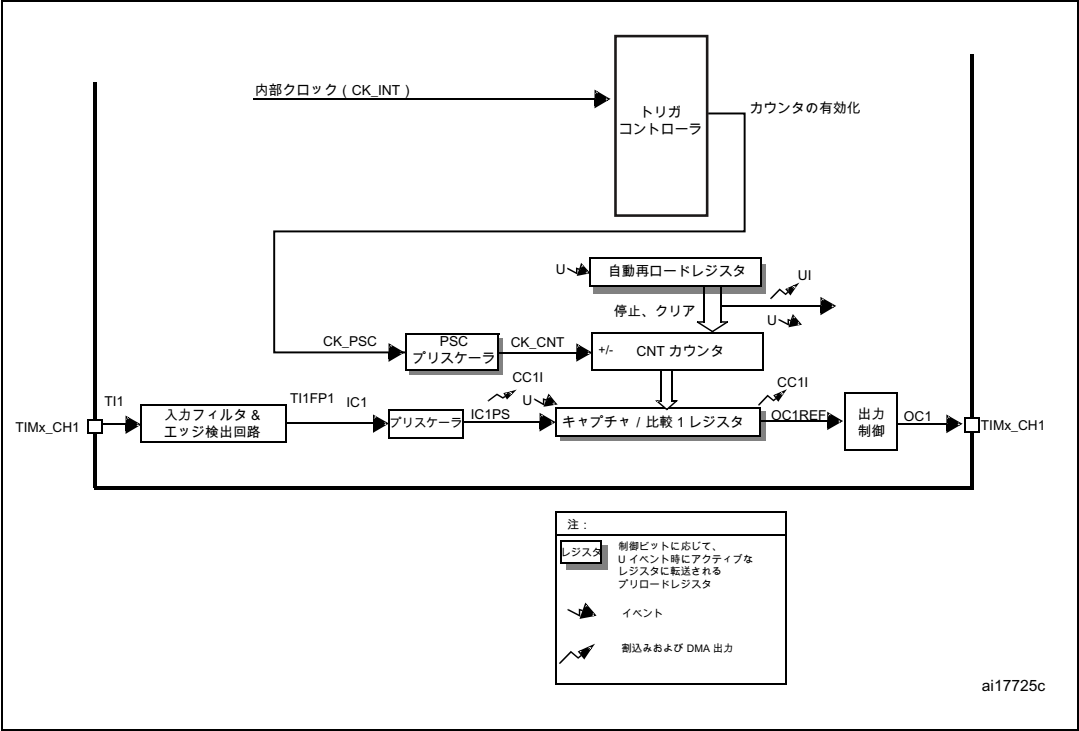


14.2.2 TIM10/TIM11 の主な機能

汎用タイマ TIM10/TIM11 の機能は以下のとおりです。

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65536 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケアラ (動作中に変更可能)。
- 次の機能を持つ独立チャネル：
 - 入力キャプチャ
 - 出力比較
 - PWM 生成 (エッジアラインモード)
 - ワンパルスモード出力
- 以下のイベント時の割込み生成。
 - 更新：カウンタオーバーフロー、カウンタの初期化 (ソフトウェアによる)
 - 入力キャプチャ
 - 出力比較

図 134. 汎用タイマのブロック図 (TIM10/11)



14.3 TIM9 から TIM11 の機能詳細

14.3.1 タイムベースユニット

タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。

カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定ごとに詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 135](#) と [図 136](#) 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 135. プリスケール分周比が 1 から 2 に変化したときのカウンタのタイミング図

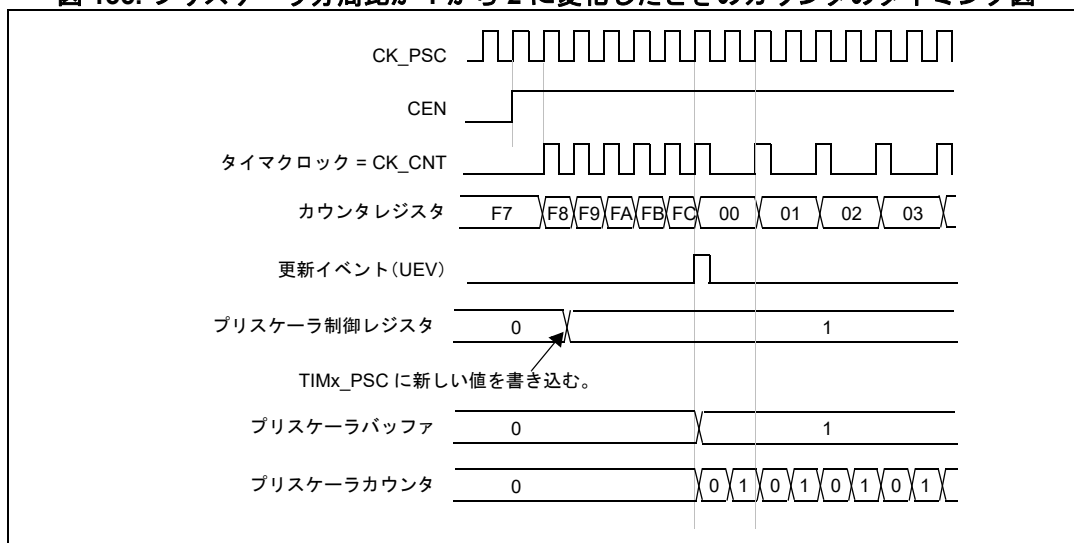
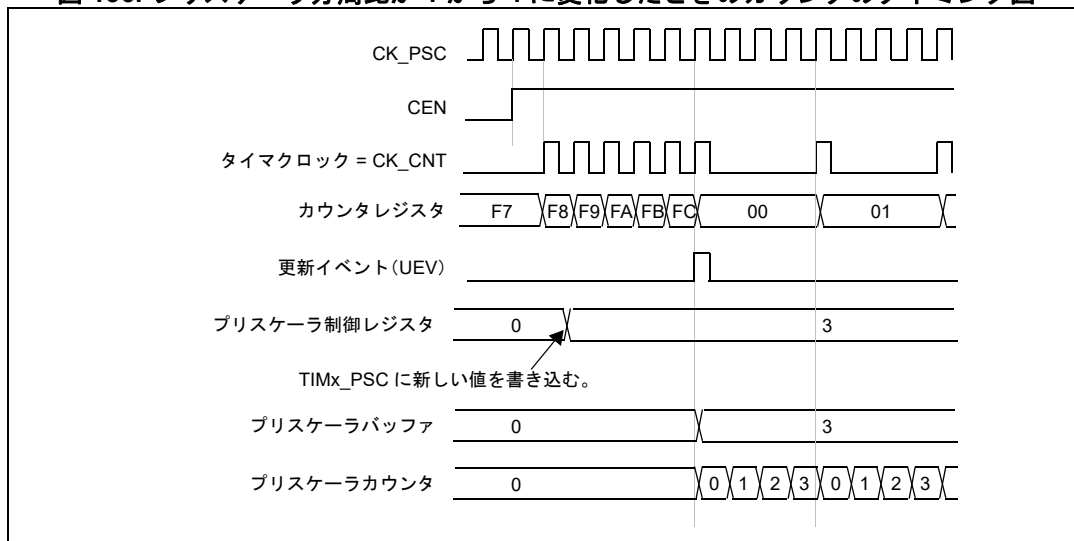


図 136. プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図



14.3.2 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。

TIMx_EGR レジスタの UG ビットを (ソフトウェアによって) セットすると、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、プリスケール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、プリスケール比は変化しません)。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、割込みは送信されません）。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 137. 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

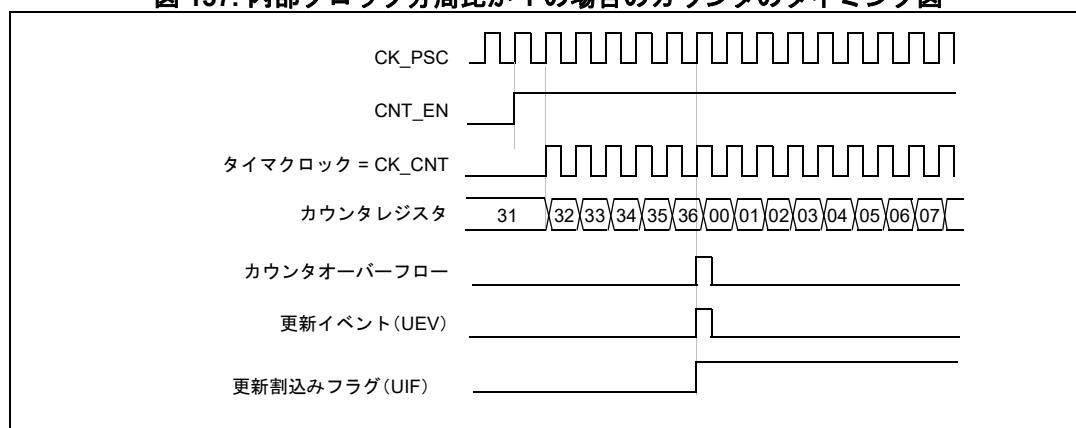


図 138. 内部クロック分周比が 2 の場合のカウンタのタイミング図

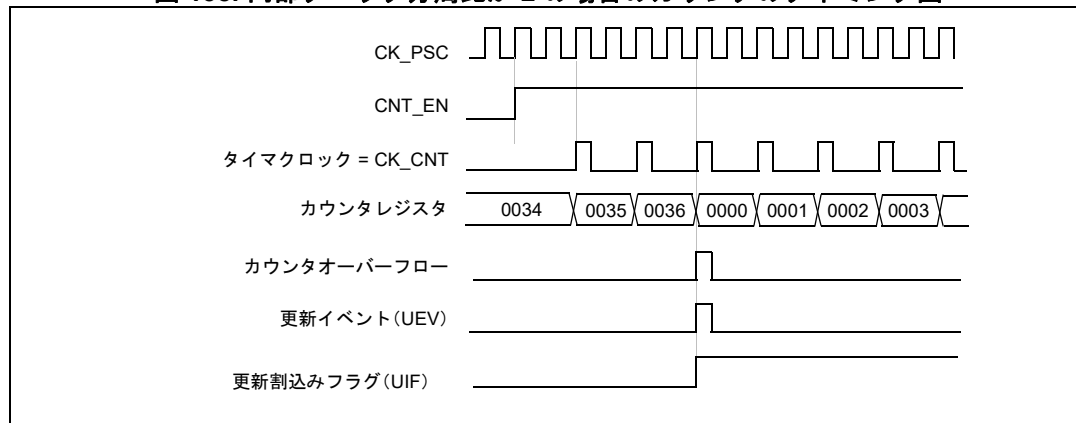


図 139. 内部クロック分周比が 4 の場合のカウンタのタイミング図

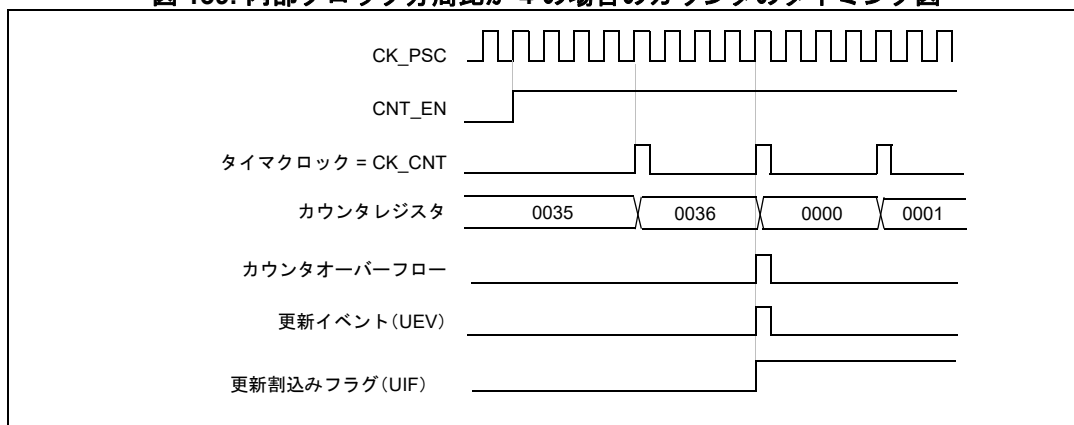


図 140. 内部クロック分周比が N の場合のカウンタのタイミング図

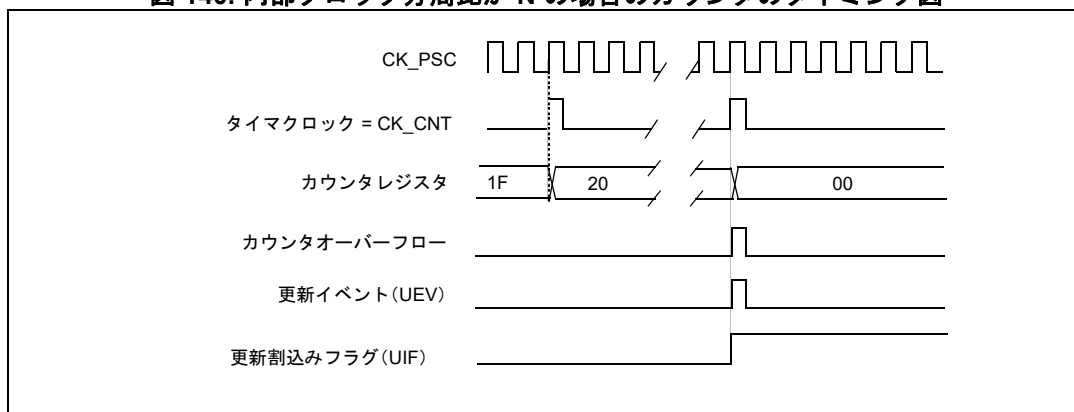


図 141. ARPE=0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

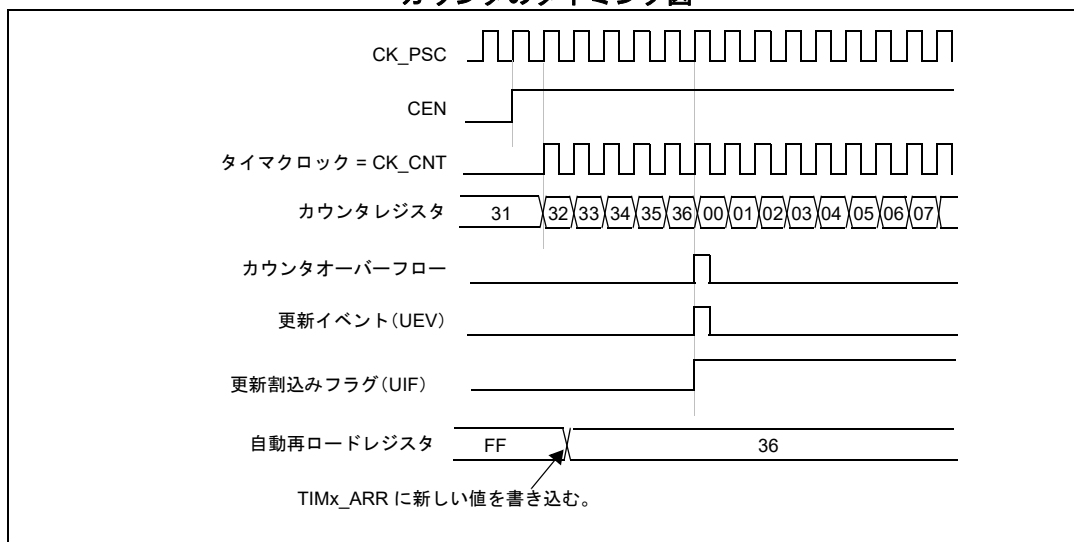
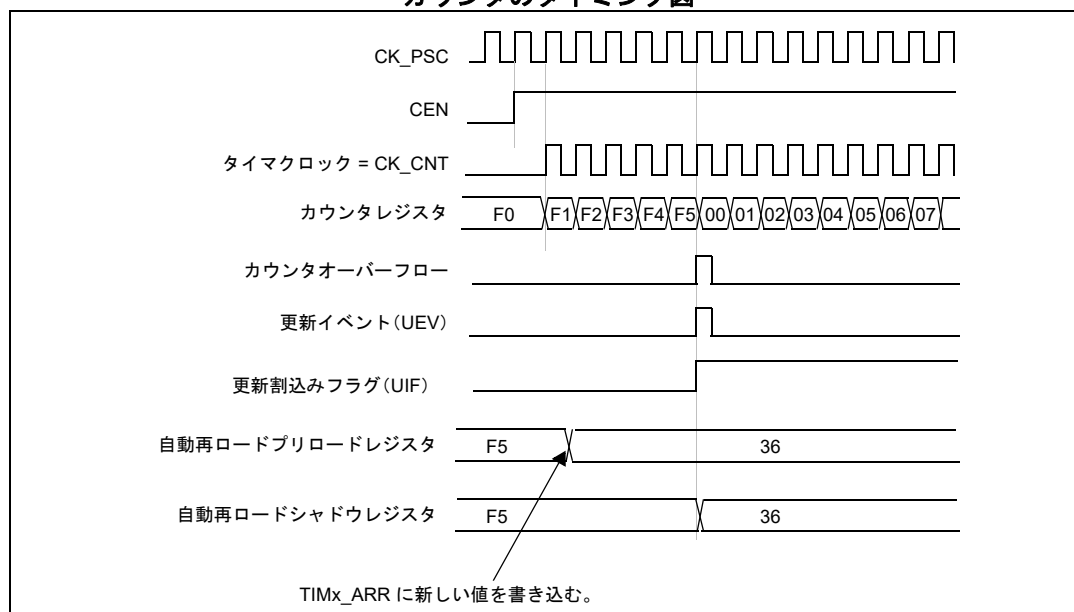


図 142. ARPE=1 (TIMx_ARR がプリロードされる) の場合の更新イベント時のカウンタのタイミング図



14.3.3 クロック選択

カウンタクロックは、次のクロックソースによって供給されます。

- 内部クロック (CK_INT)
- 外部クロックモード 1 (TIM9 用) : 外部入力ピン (Tlx)
- 内部トリガ入力 (ITRx) (TIM9 用) : 他のタイマからトリガ出力を接続。詳細については、[セクション : タイマを別のタイマのプリスケアラとして使用する](#)を参照してください。

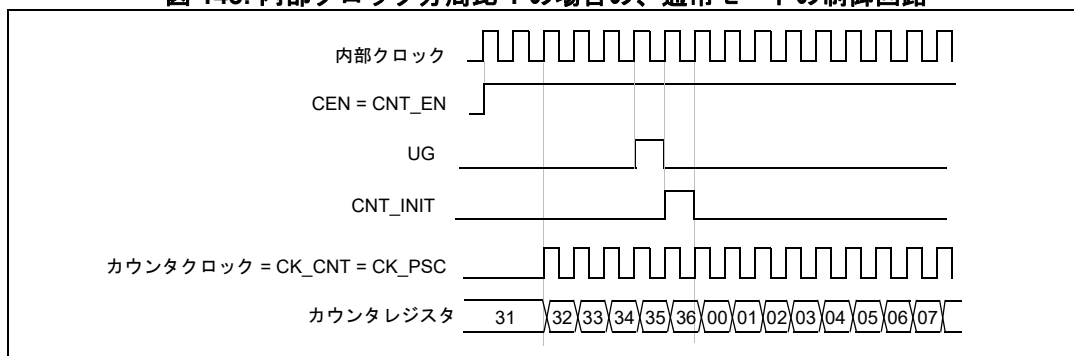
内部クロックソース (CK_INT)

内部クロックソースは、TIM10/TIM11 のデフォルトクロックソースです。

TIM9 では、スレーブモードコントローラが無効の場合 (SMS = 000)、内部クロックソースが選択されます。その際、TIMx_CR1 レジスタの CEN ビットと TIMx_EGR レジスタの UG ビットが制御ビットとして使用され、ソフトウェアからのみ変更できます (クリアされたままの UG ビットは除く)。CEN ビットに 1 が設定されると直ちに、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 143](#) 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。

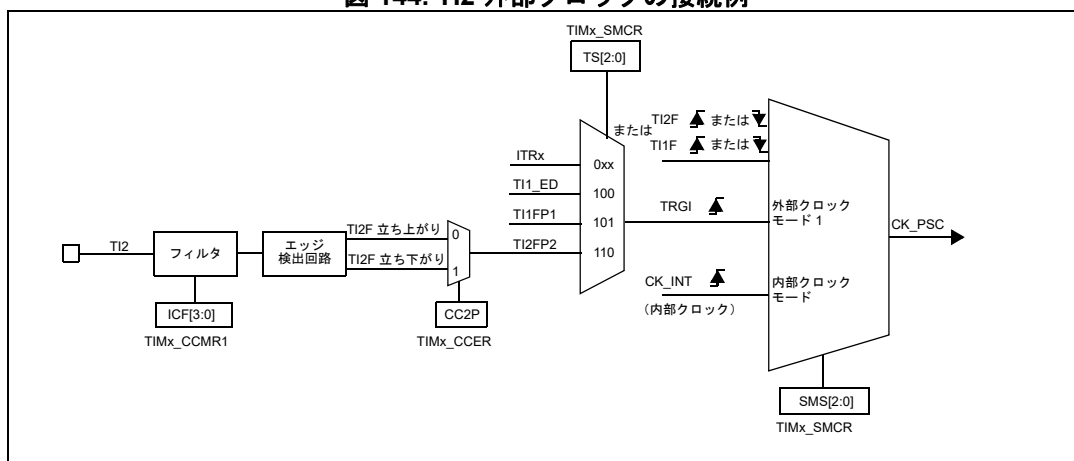
図 143. 内部クロック分周比 1 の場合の、通常モードの制御回路



外部クロックソースモード 1 (TIM9)

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 144. TI2 外部クロックの接続例



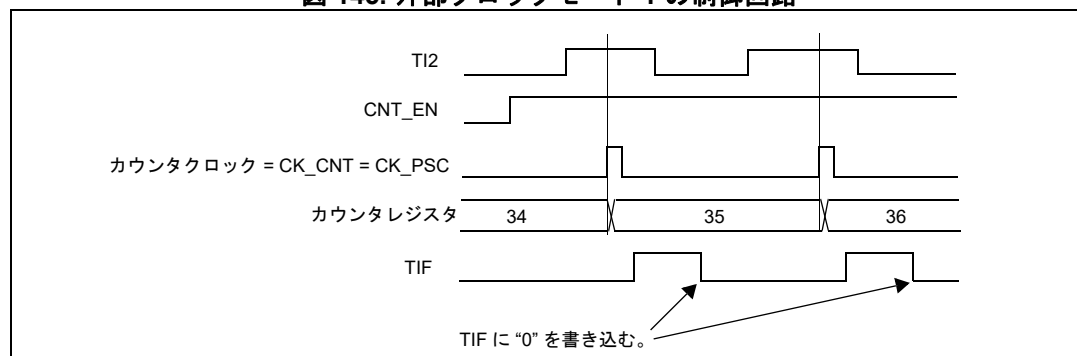
たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

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 ICF[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F = 0000 にしておきます)。
3. TIMx_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込むことによって、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタの SMS ビットに“111”を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタの TS ビットに“110”を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタの CEN ビットに“1”を書き込むことによって、カウンタを有効にします。

注：キャプチャプリスケアラはトリガには使用されないので、設定は不要です。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。
 TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 145. 外部クロックモード 1 の制御回路



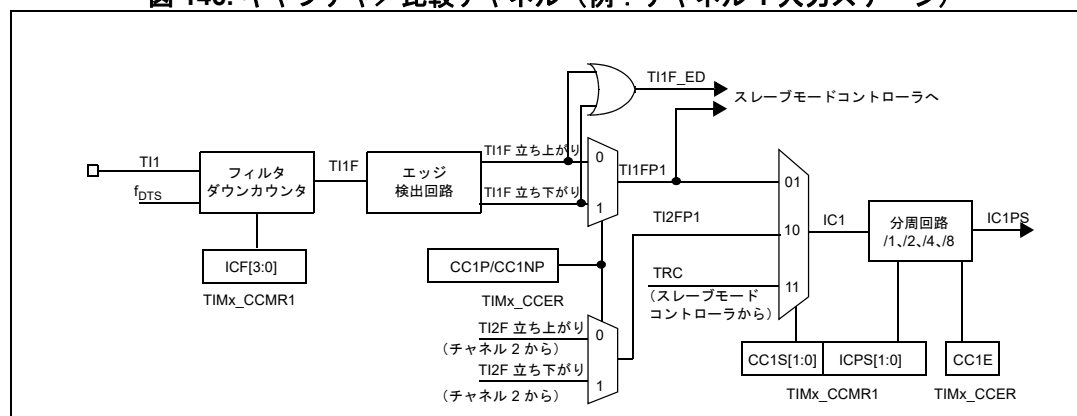
14.3.4 キャプチャ／比較チャネル

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、マルチプレクス、プリスケアラ）、および出力ステージ（コンパレータと出力制御）から構成されています。

図 146 から 図 148 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 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アラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 146. キャプチャ／比較チャネル（例：チャネル 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。信号の極性は最終出力に影響を与えます。

図 147. キャプチャ/比較チャンネル 1 メイン回路

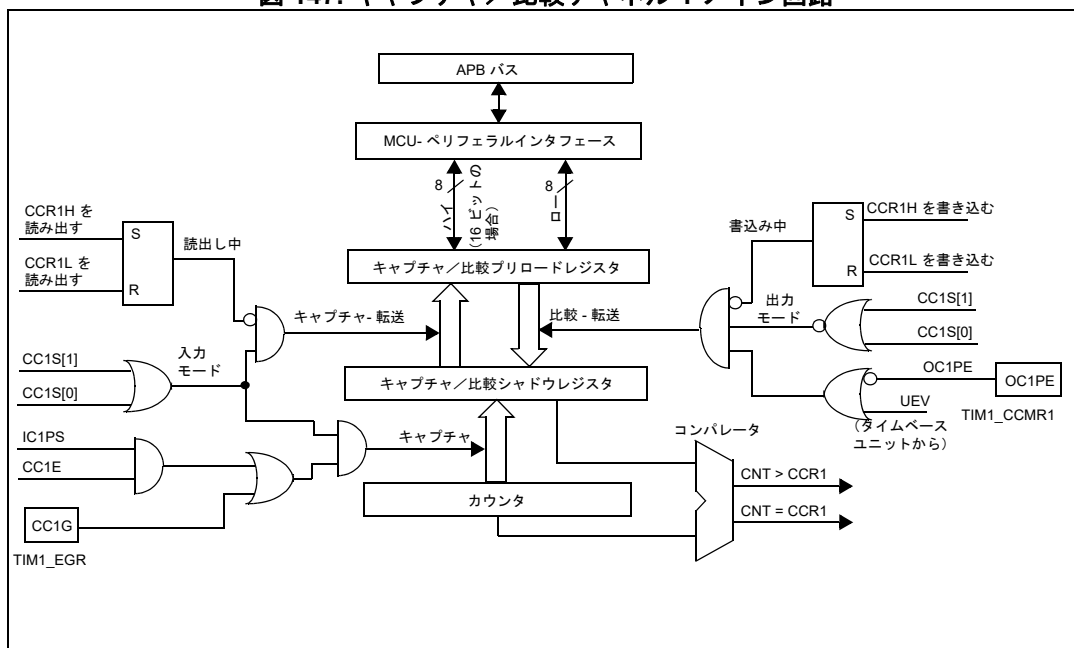
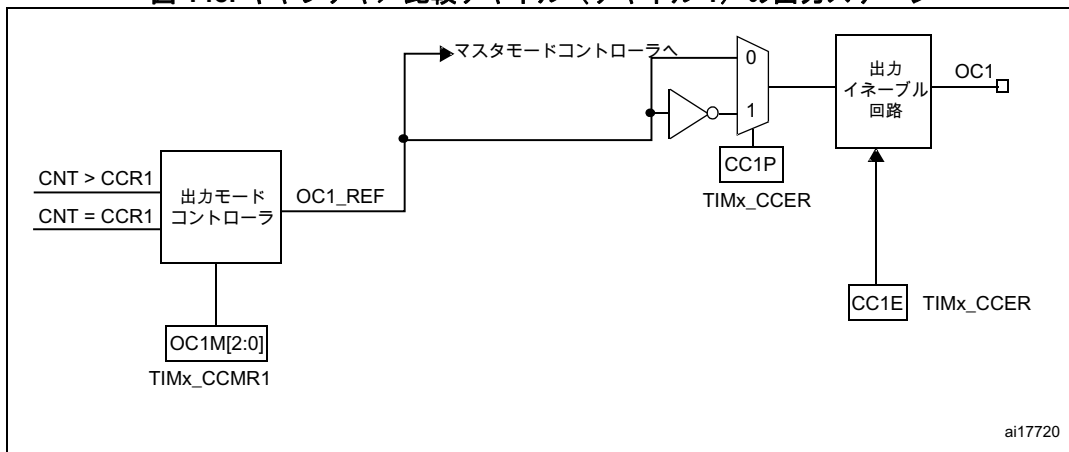


図 148. キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ



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。

14.3.5 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、割込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力が入立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx_CCR1 は TI1 入力にリンクされていなければならないので、TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化するとすぐに、チャンネルは入力モードに設定され、TIMx_CCR1 レジスタは読出し専用になります。
2. タイマに接続する信号に対して必要とする入力フィルタ時間を設定します (入力が TIx 入力の内の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビットを設定して行います)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 の遷移を検証できます (周波数 f_{DTS} でサンプリング)。次に、TIMx_CCMR1 レジスタの IC1F ビットに“0011”を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“00”を設定します (この場合、立ち上がりエッジの選択)。
4. 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
5. TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
6. 必要に応じて、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割込みリクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割込みフラグ)。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、割込みが生成されます。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読出し後、データ読出し前に発生するオーバキャプチャの見落としを避けることができます。

注： TIMx_EGR レジスタの対応する CCxG ビットをセットすることで、IC 割込みリクエストをソフトウェアで発生させることができます。

14.3.6 PWM 入力モード (TIM9 の場合のみ)

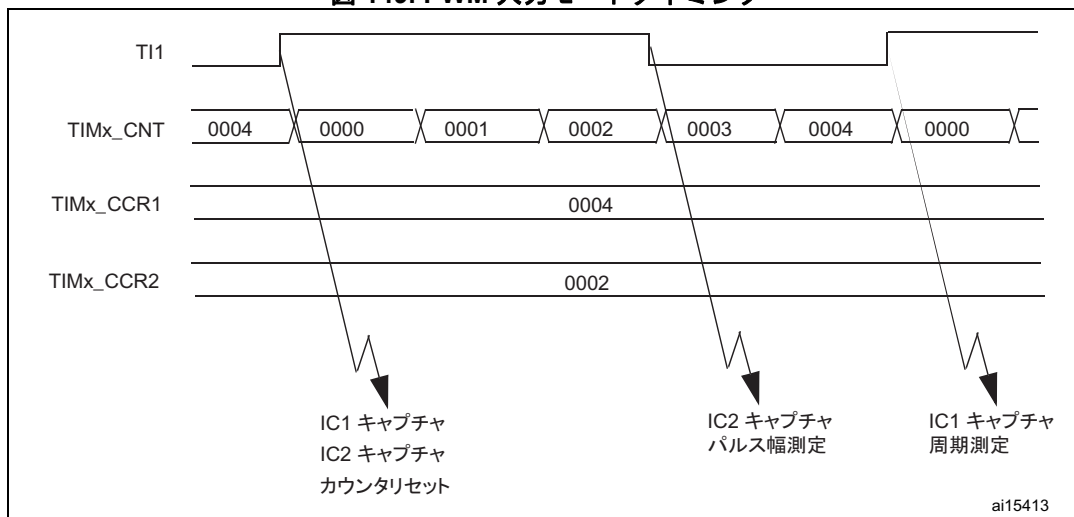
このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

1. TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
2. CC1P ビットと CC1NP ビットに“00”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用)。
3. TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
4. CC2P ビットと CC2NP ビットに“11”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
5. TIMx_SMCR レジスタの TS ビットに“101”を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
6. TIMx_SMCR レジスタの SMS ビットに“100”を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
7. TIMx_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

図 149. PWM 入力モードタイミング



1. TI1FP1 と TI1FP2 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。

14.3.7 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF、次に OCx) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx_OC MRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例 : CCxP = 0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割込みリクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

14.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

1. 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM = 000)、アクティブにセットされるか (OCxM = 001)、インアクティブにセットされるか (OCxM = 010)、または反転されます (OCxM = 011)。
2. 割込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
3. 対応する割込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割込みを生成します。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

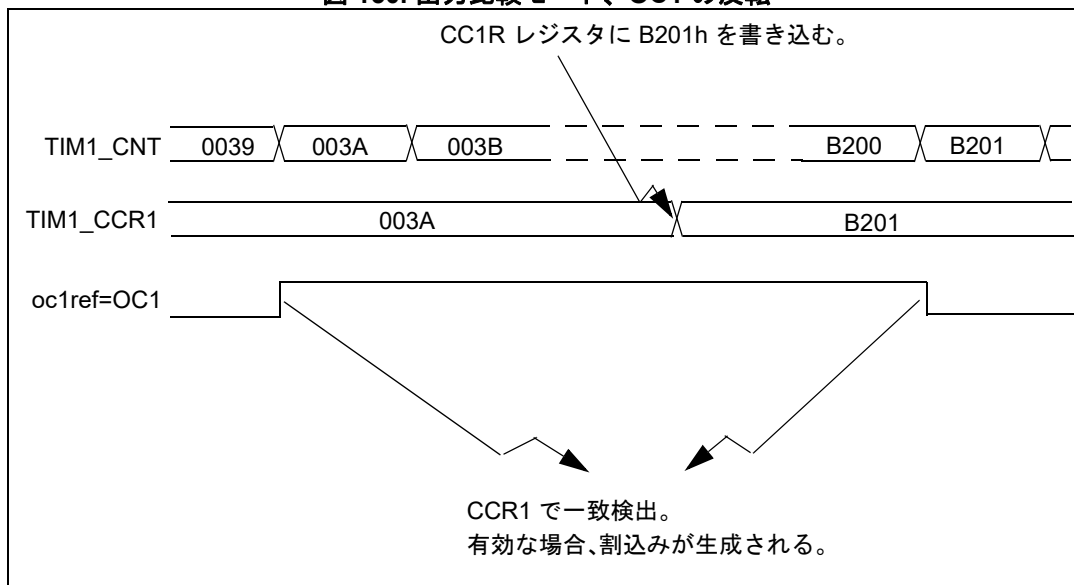
手順 :

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割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、CCxP 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 150 に示します。

図 150. 出力比較モード、OC1 の反転



14.3.9 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx の極性は、TIMx_CCER レジスタの CCxP ビットを使用して、ソフトウェアでプログラムできます。アクティブハイまたはアクティブラーとしてプログラムできます。OCx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効化されます。詳細については、TIMx_CCERx レジスタの説明を参照してください。

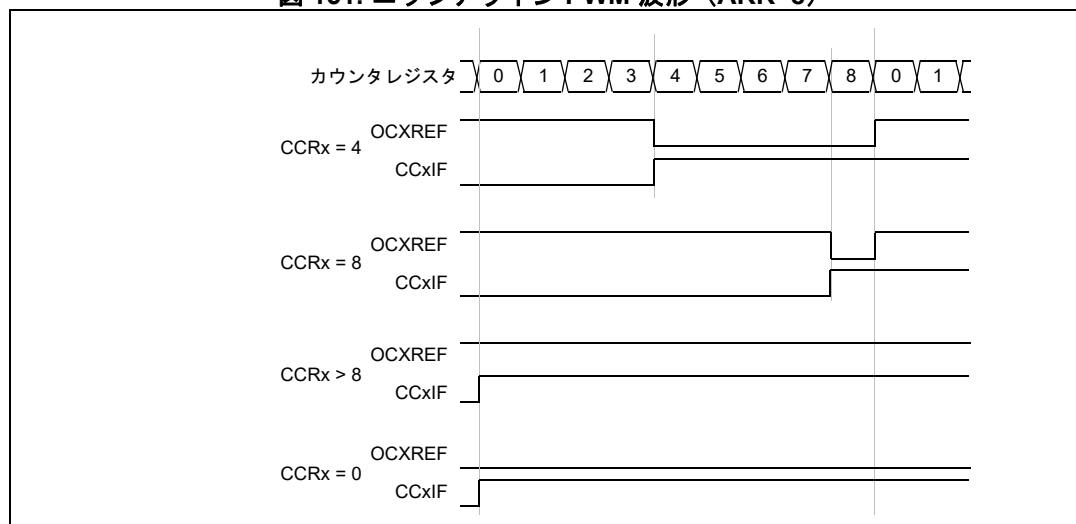
PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CNT ≤ TIMx_CCRx かどうか判断されます。≤

カウンタはカウントアップしているので、タイマはエッジアラインモードでのみ PWM を生成できます。

PWM エッジアラインモード

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値（TIMx_ARR レジスタの）より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。図 151 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 151. エッジアライン PWM 波形（ARR=8）



14.3.10 ワンパルスモード

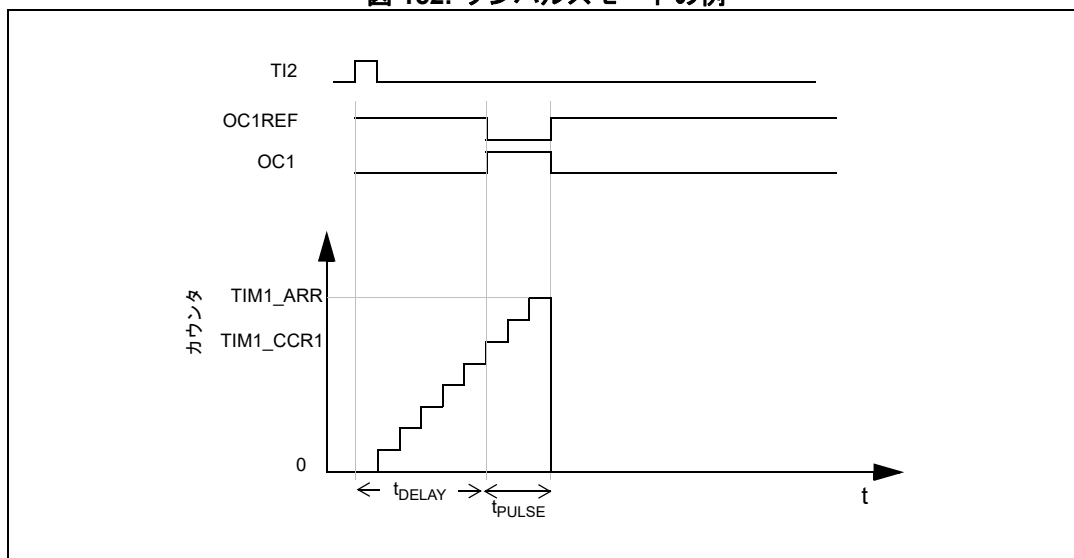
ワンパルスモード（OPM : One Pulse Mode）は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に（タイマがトリガを待っているときに）、設定が次のようであればなりません。

$$\text{CNT} < \text{CCRx} \leq \text{ARR} \quad (\text{特に、} 0 < \text{CCRx})$$

図 152. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
2. TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタの CC2P ビットと CC2NP ビットに“0”を書き込みます。
3. TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx_SMCR レジスタの TS ビットに“110”を書き込みます。
4. TI2FP2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M ビットに“111”を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx_CCMR1 レジスタの OC1PE ビットに“1”を書込み、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 (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。

14.3.11 TIM9 外部トリガ同期

TIM9 タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレーブモード : リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

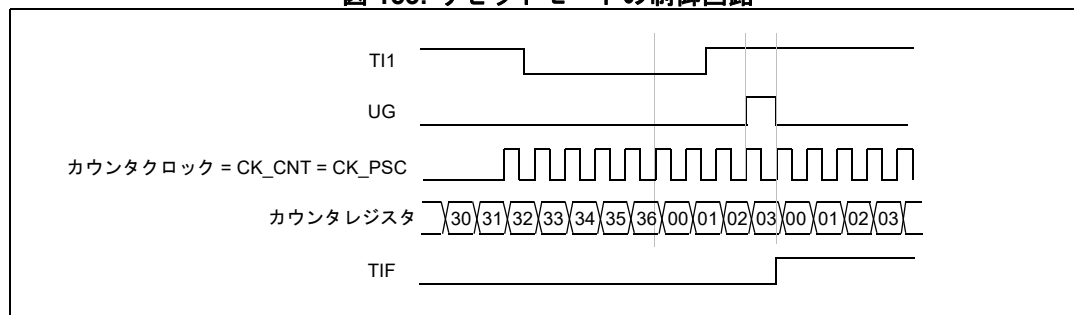
次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

1. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S = 01)。TIMx_CCER レジスタの CC1P と CC1NP に“00”を書き込んで、極性を有効にします (その後、立ち上がりエッジのみを検出)。
2. TIMx_SMCR レジスタの SMS ビットに“100”を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタの TS ビットに“101”を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx_CR1 レジスタの CEN ビットに“1”を書き込むことによって、カウンタを開始します。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、(TIMx_DIER レジスタの TIE ビットが) 有効な場合は割り込みリクエストを送信できます。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 153. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

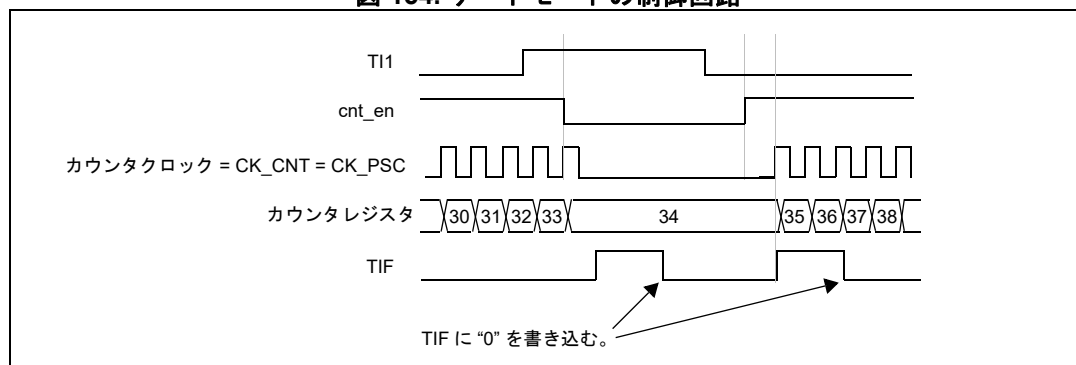
次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

1. TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S = 01)。TIMx_CCER レジスタの CC1P ビットに“1”、CC1NP ビットに“0”を書き込んで、極性を有効にします (その後、立ち上がりエッジのみを検出)。
2. TIMx_SMCR レジスタの SMS ビットに“101”を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタの TS ビットに“101”を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx_CR1 レジスタの CEN ビットに“1”を書き込んで、カウンタを有効にします (ゲートモードでは、CEN = 0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません)。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 154. 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、カウンタを開始できます。

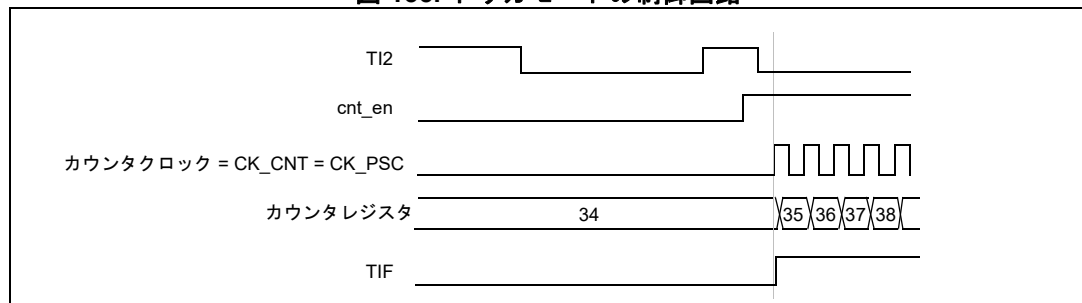
次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

1. TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F = 0000 のままにしておく)。キャプチャプリスケアラはトリガには使用されないため、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 レジスタの CC2P ビットに“1”、CC2NP ビットに“0”を書き込んで、極性を有効にします (その後、ローレベルのみを検出)。
2. TIMx_SMCR レジスタの SMS ビットに“110”を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタの TS ビットに“110”を書き込むことによって、入力ソースとして TI2 を選択します。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 155. トリガモードの制御回路



14.3.12 タイマ同期（TIM9）

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 13.3.15 : 341 ページのタイマの同期](#)を参照してください。

14.3.13 デバッグモード

マイクロコントローラがデバッグモードになると（FPU 搭載 Cortex®-M4 コアは停止状態）、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 23.16.2 : タイマ、ウォッチドッグ、および I²C のデバッグサポート](#)を参照してください。

14.4 TIM9 レジスタ

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

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

14.4.1 TIM9 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	
予約済み						CKD[1:0]		ARPE	予約済み				OPM	URS	UDIS	CEN
						rw	rw	rw					rw	rw	rw	rw

- ビット 15:10 予約済みであり、リセット値に保持する必要があります。
- ビット 9:8 **CKD** : クロック分周
- このビットフィールドは、タイマクロック (CK_INT) 周波数と、デジタルフィルタ (Tix) によって使用されるサンプリングクロックとの間の分周比を示します。
- 00: $t_{DTS} = t_{CK_INT}$
01: $t_{DTS} = 2 \times t_{CK_INT}$
10: $t_{DTS} = 4 \times t_{CK_INT}$
11: 予約済み
- ビット 7 **ARPE** : 自動再ロードプリロードイネーブル
- 0 : TIMx_ARR レジスタはバッファされません。
1 : TIMx_ARR レジスタはバッファされます。
- ビット 6:4 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込みを生成します (有効な場合)。
- カウンタオーバーフロー
 - UG ビットのセット
- 1 : カウンタオーバーフローのみが更新割込みを生成します (有効な場合)。
- ビット 1 **UDIS** : 更新ディセーブル
- このビットは、更新イベント (UEV) 生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。
- 0 : UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。
- カウンタオーバーフロー
 - UG ビットのセット
- バッファを持つレジスタにはプリロード値がロードされます。
- 1 : UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR、PSC、CCRx) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。
- ビット 0 **CEN** : カウンタイネーブル
- 0 : カウンタは無効です。
1 : カウンタは有効です。
- ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

14.4.2 TIM9 のスレーブモード制御レジスタ（TIMx_SMCR）

アドレスオフセット：0x08

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
予約済み								MSM	TS[2:0]				Res.	SMS[2:0]		
								rw	rw	rw	rw	rw		rw	rw	

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **MSM**：マスタ／スレーブモード

0：影響なし。

1：トリガ入力（TRGI）に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期（TRGO を通じて）を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期する場合に役立ちます。

ビット 6:4 **TS**：トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

000：内部トリガ 0（ITR0）

001：内部トリガ 1（ITR1）

010：内部トリガ 2（ITR2）

011：内部トリガ 3（ITR3）

100：TI1 エッジ検出回路（TI1F_ED）

101：フィルタタイマ入力 1（TI1FP1）

110：フィルタタイマ入力 2（TI2FP2）

111：予約済み。

各タイマにおける ITRx の意味の詳細については、表 57：391 ページのTIMx 内部トリガ接続を参照してください。

注： 遷移時の誤ったエッジ検出を避けるために、これらのビットは必ず使用されていないとき（SMS = 000 のときなど）に変更してください。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMS**：スレーブモード選択

外部信号が選択されると、トリガ信号（TRGI）のアクティブエッジが外部入力で選択された極性にリンクされます（入力制御レジスタおよび制御レジスタの説明を参照してください）。

000：スレーブモードは無効です。CEN = 1 の場合、プリスケアラは内部クロックによって直接クロック供給されます。

001：予約済み

010：予約済み

011：予約済み

100：リセットモード - 選択されたトリガ入力（TRGI）の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

101：ゲートモード - カウンタクロックは、トリガ入力（TRGI）がハイのときに有効になります。トリガがローになると、カウンタは停止します（リセットはされません）。カウンタの開始と停止の両方が制御されます。

110：トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します（リセットはされません）。カウンタの開始のみが制御されます。

111：外部クロックモード 1 - 選択されたトリガ（TRGI）の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合（TS = 100）、ゲートモードを使用することはできません。TI1F_ED は TI1F の遷移ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

表 57. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = '000')	ITR1 (TS = '001')	ITR2 (TS = '010')	ITR3 (TS = '011')
TIM9	TIM2_TRGO	TIM3_TRGO	TIM10_OC	TIM11_OC

14.4.3 TIM9 割込み有効レジスタ (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
予約済み									TIE	Res.			CC2IE	CC1IE	UIE
									rw				rw	rw	rw

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE** : トリガ割込みイネーブル
0 : トリガ割込みは無効です。
1 : トリガ割込みは有効です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IE** : キャプチャ／比較 2 割込みイネーブル
0 : CC2 割込みは無効です。
1 : CC2 割込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割込みイネーブル
0 : CC1 割込みは無効です。
1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル
0 : 更新割込みは無効です。
1 : 更新割込みは有効です。

14.4.4 TIM9 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
予約済み					CC2OF	CC1OF	予約済み		TIF	予約済み			CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

ビット 15:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **CC2OF** : キャプチャ／比較 2 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ
このフラグは、対応するチャ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 : トリガ割込みが保留中です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2IF** : キャプチャ/比較 2 割込みフラグ

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 割込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました (選択された極性に一致するエッジが IC1 で検出されました)。

ビット 0 **UIF** : 更新割込みフラグ

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 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、トリガイイベントによって 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)。

14.4.5 TIM9 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
予約済み									TG	予約済み			CC2G	CC1G	UG
									W				W	W	W

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG**：トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込が発生します。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **CC2G**：キャプチャ／比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G**：キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：チャンネル 1 でキャプチャ／比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合：

CC1IF フラグがセットされ、対応する割り込みが送信されます（有効な場合）。

CC1 チャンネルが入力として設定されている場合：

現在のカウンタ値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みが送信されます（有効な場合）。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG**：更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：カウンタを再初期化し、レジスタの更新を生成します。プリスケアラのカウンタもクリアされます（分周比は変化しません）。カウンタはクリアされます。

14.4.6 TIM9 のキャプチャ／比較モードレジスタ 1 (TIMx_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のすべてのビットは、入力モードと出力モードで機能が異なります。特定のビットについて、OCxx は、チャンネルが出力モードに設定されているときの機能を記述し、ICxx は、チャンネルが入力モードに設定されているときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
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 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
00 : CC2 チャンネルは出力として設定されます。
01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。
10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。
11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E = 0) のときにのみ書き込み可能です。

ビット 6:4 **OC1M** : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは、それぞれ CC1P および CC1NP ビットに依存します。
000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません（このモードはタイミングベースを生成するために使用されます）。
001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、TIMx_CNT カウンタがキャプチャ／比較レジスタ 1 (TIMx_CCR1) に一致したとき、強制的にハイになります。
010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、TIMx_CNT カウンタがキャプチャ／比較レジスタ 1 (TIMx_CCR1) に一致したとき、強制的にローになります。
011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。
100 : 強制インアクティブレベル - OC1REF は強制的にローになります。
101 : 強制アクティブレベル - OC1REF は強制的にハイになります。
110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF = 0) に、そうでない場合はアクティブ (OC1REF = 1) になります。
111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

注 : **PWM モード 1 または 2 では、比較結果が変化したとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。**

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われず、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： PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 は、トリガがオンのときでも、カウンタと CCR1 の値に依存して、通常どおりに動作します。トリガ入力にエッジが発生してから CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注： CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E = 0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E = 0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング 1000 : $f_{SAMPLING}=f_{DTS}/8$, N = 6

0001 : $f_{SAMPLING}=f_{CK_INT}$, N=2 1001 : $f_{SAMPLING}=f_{DTS}/8$, N=8

0010 : $f_{SAMPLING}=f_{CK_INT}$, N = 4 1010 : $f_{SAMPLING}=f_{DTS}/16$, N = 5

0011 : $f_{SAMPLING}=f_{CK_INT}$, N=8 1011 : $f_{SAMPLING}=f_{DTS}/16$, N = 6

0100 : $f_{SAMPLING}=f_{DTS}/2$, N=6 1100 : $f_{SAMPLING}=f_{DTS}/16$, N=8

0101 : $f_{SAMPLING}=f_{DTS}/2$, N=8 1101 : $f_{SAMPLING}=f_{DTS}/32$, N=5

0110 : $f_{SAMPLING}=f_{DTS}/4$, N=6 1110 : $f_{SAMPLING}=f_{DTS}/32$, N=6

0111 : $f_{SAMPLING}=f_{DTS}/4$, N=8 1111 : $f_{SAMPLING}=f_{DTS}/32$, N=8

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの比を定義します。

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。

14.4.7 TIM9 のキャプチャ／比較有効レジスタ（TIMx_CCER）

アドレスオフセット：0x20

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
								rw		rw	rw	rw		rw	rw

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **CC2NP**：キャプチャ／比較 2 出力極性

CC1NP の説明を参照してください。

ビット 6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **CC2P**：キャプチャ／比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E**：キャプチャ／比較 2 出力イネーブル

CC1E の説明を参照してください。

ビット 3 **CC1NP**：キャプチャ／比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合：CC1NP はクリア状態に維持する必要があります。

CC1 チャンネルが入力として設定されている場合：CC1NP ビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます（CC1P の説明を参照）。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P**：キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合：

0：OC1 はアクティブハイです。

1：OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合：

CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。

00：非反転／立ち上がりエッジ

回路は TIxFP1 の立ち上がりエッジに反応し（キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード）、TIxFP1 は反転されません（ゲートモードでのトリガ、エンコーダモード）。

01：反転／立ち下がりエッジ

回路は TIxFP1 の立ち下がりエッジに反応し（キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード）、TIxFP1 は反転されます（ゲートモードでのトリガ、エンコーダモード）。

10：予約済み。この設定は使用しないでください。

注： 11：非反転／両エッジ

回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し（キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード）、TIxFP1 は反転されません（ゲートモードでのトリガ）。この設定をエンコーダモードに使用することはできません。

ビット 0 **CC1E**：キャプチャ／比較 1 出力イネーブル

CC1 チャンネルが出力として設定されている場合：

0：オフ - OC1 はアクティブではありません。

1：オン - OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合：

このビットによって、カウンタ値のキャプチャ／比較レジスタ 1（TIMx_CCR1）へのキャプチャが実際に行われるかどうかが決まります。

0：キャプチャは無効です。

1：キャプチャは有効です。

表 58. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx = 0、OCx_EN = 0)
1	OCx = OCxREF + 極性、OCx_EN = 1

注：標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

14.4.8 TIM9 のカウンタ（TIMx_CNT）

アドレスオフセット：0x24

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0]：カウンタ値

14.4.9 TIM9 プリスケアラ（TIMx_PSC）

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値

カウンタクロック周波数 CK_CNT は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブなプリスケアラレジスタにロードされる値を含みます。

14.4.10 TIM9 自動再ロードレジスタ（TIMx_ARR）

アドレスオフセット：0x2C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0]：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 14.3.1：372 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

14.4.11 TIM9 キャプチャ／比較レジスタ 1（TIMx_CCR1）

アドレスオフセット：0x34

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、プリロード値がロードされたままになります。そうでない場合、プリロード値は、更新イベントが発生すると、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブなキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

14.4.12 TIM9 キャプチャ／比較レジスタ 2（TIMx_CCR2）

アドレスオフセット：0x38

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR2[15:0]**：キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合：

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、プリロード値がロードされたままになります。そうでない場合、プリロード値は、更新イベントが発生すると、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブなキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合：

CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。

14.4.13 TIM9 レジスタマップ

TIM9 のレジスタは、次のように 16 ビットのアドレス可能レジスタとしてマップされます。

表 59. TIM9 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 リセット値	予約済み														CKD [1:0]		ARPE	予約済み				OPM	URS	UDIS	CEN							
0x08	TIMx_SMCR リセット値	予約済み														MSM		TS[2:0]				予約済み	SMS[2:0]										
0x0C	TIMx_DIER リセット値	予約済み														TIE		予約済み				CC2IE	CC1IE	UIE									
0x10	TIMx_SR リセット値	予約済み														CC2OF	CC1OF	予約済み	TIF	予約済み				CC2IF	CC1IF	UIF							
0x14	TIMx_EGR リセット値	予約済み														TG		予約済み				CC2G	CC1G	UG									
0x18	TIMx_CCMR1 出力比較モード リセット値	予約済み										OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]		予約済み	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]										
	TIMx_CCMR1 入力キャプチャ モード リセット値	予約済み										IC2F[3:0]		IC2 PSC [1:0]	CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]	CC1S [1:0]											
0x1C	予約済み																																
0x20	TIMx_CCER リセット値	予約済み														CC2NP		予約済み	CC2P	CC2E	CC1NP	予約済み	CC1P	CC1E									
0x24	TIMx_CNT リセット値	予約済み										CNT[15:0]																					
0x28	TIMx_PSC リセット値	予約済み										PSC[15:0]																					
0x2C	TIMx_ARR リセット値	予約済み										ARR[15:0]																					
0x30	予約済み																																
0x34	TIMx_CCR1 リセット値	予約済み										CCR1[15:0]																					
0x38	TIMx_CCR2 リセット値	予約済み										CCR2[15:0]																					
0x3C～ 0x4C	予約済み																																

レジスタ境界アドレスについては、38 ページの表 1 を参照してください。

14.5 TIM10/11 レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位で書き込む必要があります。読出しアクセスは、バイト（8 ビット）、ハーフワード（16 ビット）、またはワード（32 ビット）単位で行うことができます。

14.5.1 TIM10/11 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
予約済み						CKD[1:0]		ARPE	予約済み				URS	UDIS	CEN
						rw	rw	rw					rw	rw	rw

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD** : クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デジタルフィルタ (Tix) によって使用されるサンプリングクロックとの間の分周比を示します。

00: $t_{DTS} = t_{CK_INT}$
01: $t_{DTS} = 2 \times t_{CK_INT}$
10: $t_{DTS} = 4 \times t_{CK_INT}$
11: 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0: TIMx_ARR レジスタはバッファされません。
1: TIMx_ARR レジスタはバッファされます。

ビット 6:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **URS** : 更新リクエストソース

このビットは、更新割込み (UEV) ソースを選択するために、ソフトウェアによってセット/クリアされます。

0: 次のイベントのいずれかが UEV を発生します (有効な場合)。

- カウンタオーバーフロー
- UG ビットのセット

1: カウンタオーバーフローのみが UEV を発生します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、更新割込み (UEV) イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

0: UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー
- UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

1: UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR, PSC, CCRx) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0: カウンタは無効です。
1: カウンタは有効です。

14.5.2 TIM10/11 割込み有効レジスタ (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
予約済み														CC1IE	UIE
														rw	rw

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ/比較 1 割込みイネーブル

0 : CC1 割込みは無効です。

1 : CC1 割込みは有効です。

ビット 0 **UIE** : 更新割込みイネーブル

0 : 更新割込みは無効です。

1 : 更新割込みは有効です。

14.5.3 TIM 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
予約済み						CC1OF	予約済み							CC1IF	UIF
						rc_w0								rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

0 : オーバーキャプチャは検出されていません。

1: CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IF** : キャプチャ／比較 1 割込みフラグ

CC1 チャンネルが出力として設定されている場合 :
このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。
0 : 一致していません。
1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。
CC1 チャンネルが入力として設定されている場合 :
このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。
0 : 入力キャプチャは発生していません。
1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました（選択された極性に一致するエッジが IC1 で検出されました）。

ビット 0 **UIF** : 更新割込みフラグ

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 : 更新割込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

14.5.4 TIM 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
予約済み														CC1G	UG
														w	w

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1G** : キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。
0 : 影響なし。
1 : チャンネル 1 でキャプチャ／比較イベントが生成されます。
CC1 チャンネルが出力として設定されている場合 :
CC1IF フラグがセットされ、対応する割込みが送信されます（有効な場合）。
CC1 チャンネルが入力として設定されている場合 :
カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割込みが送信されます（有効な場合）。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。
0 : 影響なし。
1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。カウンタはクリアされます。

14.5.5 TIM10/11 のキャプチャ／比較モードレジスタ 1 (TIMx_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み									OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
予約済み							IC1F[3:0]				IC1PSC[1:0]				
							rw	rw	rw	rw	rw	rw	rw	rw	

出力比較モード

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **OC1M** : 出力比較 1 モード

これらのビットは、OC1 の元となる出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 のアクティブレベルは CC1P ビットに依存します。

000 : 停止。出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

111 : PWM モード 2 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

注 : PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときに、OCREF のレベルが変化します。

ビット 3 **OC1PE** : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 **OC1FE** : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力のエッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 :

11 :

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、 f_{DTS} でサンプリング1000 : $f_{SAMPLING}=f_{DTS}/8$, $N = 6$

0001 : $f_{SAMPLING}=f_{CK_INT}$, $N=2$ 1001 : $f_{SAMPLING}=f_{DTS}/8$, $N=8$

0010 : $f_{SAMPLING}=f_{CK_INT}$, $N=4$ 1010 : $f_{SAMPLING}=f_{DTS}/16$, $N=5$

0011 : $f_{SAMPLING}=f_{CK_INT}$, $N = 8$ 1011 : $f_{SAMPLING}=f_{DTS}/16$, $N = 6$

0100 : $f_{SAMPLING}=f_{DTS}/2$, $N=6$ 1100 : $f_{SAMPLING}=f_{DTS}/16$, $N=8$

0101 : $f_{SAMPLING}=f_{DTS}/2$, $N=8$ 1101 : $f_{SAMPLING}=f_{DTS}/32$, $N=5$

0110 : $f_{SAMPLING}=f_{DTS}/4$, $N=6$ 1110 : $f_{SAMPLING}=f_{DTS}/32$, $N=6$

0111 : $f_{SAMPLING}=f_{DTS}/4$, $N=8$ 1111 : $f_{SAMPLING}=f_{DTS}/32$, $N=8$

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : 予約済み

11 : 予約済み

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

14.5.6 TIM10/11 のキャプチャ／比較有効レジスタ（TIMx_CCER）

アドレスオフセット：0x20

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP**：キャプチャ／比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合：CC1NP はクリアされたままにする必要があります。
 CC1 チャンネルが入力として設定されている場合：CC1NP ビットは、TI1FP1 の極性を定義するために CC1P と組み合わせて使用されます（CC1P の説明を参照してください）。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P**：キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合：

0：OC1 はアクティブハイです。

1：OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合：

CC1P ビットは、トリガ動作またはキャプチャ動作での TI1FP1 および TI2FP1 の極性を選択します。

00：非反転／立ち上がりエッジ

回路は TI1FP1 の立ち上がりエッジに反応し（キャプチャモード）、TI1FP1 は反転されません。

01：反転／立ち下がりエッジ

回路は TI1FP1 の立ち下がりエッジに反応し（キャプチャモード）、TI1FP1 は反転されます。

10：予約済み。この設定は使用しないでください。

11：非反転／両エッジ

回路は TI1FP1 の立ち上がりエッジに反応し（キャプチャモード）、TI1FP1 は反転されません。

ビット 0 **CC1E**：キャプチャ／比較 1 出力イネーブル。

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 60. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効（OCx=0、OCx_EN=0）
1	OCx = OCxREF + 極性、OCx_EN = 1

注：標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

14.5.7 TIM10/11 のカウンタ (TIMx_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CNT[15:0]** : カウンタ値

14.5.8 TIM10/11 プリスケアラ (TIMx_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **PSC[15:0]** : プリスケアラ値

カウンタクロック周波数 CK_CNT は $f_{CK_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます。

14.5.9 TIM10/11 自動再ロードレジスタ (TIMx_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **ARR[15:0]** : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

APR の更新と動作の詳細については、[セクション 14.3.1 : 372 ページのタイムベースユニット](#)を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

14.5.10 TIM10/11 キャプチャ／比較レジスタ 1（TIMx_CCR1）

アドレスオフセット：0x34

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

14.5.11 TIM11 オプションレジスタ 1（TIM11_OR）

アドレスオフセット：0x50

リセット値：0x0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み														TI1_RMP[1:0]	
														rw	

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TI1_RMP[1:0]**：TIM11 入力 1 再配置機能

ソフトウェアでセット／クリアされます。

00、01、11：TIM11 のチャンネル 1 は、GPIO に接続されます（データシートの代替機能配置表を参照してください）。

10：HSE_RTC クロック（HSE は、プログラム可能なプリスケアラで分周されます）は、測定目的のため、TIM11_CH1 入力に接続されます。

14.5.12 TIM10/11 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットのアドレス可能レジスタとしてマップされます。

表 61. TIM10/11 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	予約済み																						CKD [1:0]		ARPE	予約済み				URS	UDIS	CEN	
	0 0 0																							0	0									0
0x08	TIMx_SMCR	予約済み																																
	リセット値																																	
0x0C	TIMx_DIER	予約済み																														CC1IE	UIE	
	リセット値																																	0
0x10	TIMx_SR	予約済み																						CC1OF	予約済み				CC1IF	UIF				
	リセット値																														0	0	0	
0x14	TIMx_EGR	予約済み																														CC1G	UG	
	リセット値																																	0
0x18	TIMx_CCMR1 出力比較モード	予約済み																						OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]						
	リセット値																							0	0				0	0	0	0	0	
	TIMx_CCMR1 入力キャプチャ モード	予約済み																						IC1F[3:0]		IC1 PSC [1:0]	CC1S [1:0]							
	リセット値																							0	0			0	0	0	0	0		
0x1C	予約済み																																	
0x20	TIMx_CCER	予約済み																												CC1NP	予約済み	CC1P	CC1E	
	リセット値																																	0
0x24	TIMx_CNT	予約済み																CNT[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x28	TIMx_PSC	予約済み																PSC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x2C	TIMx_ARR	予約済み																ARR[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x30	予約済み																																	
0x34	TIMx_CCR1	予約済み																CCR1[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x38～ 0x4C	予約済み																																	

表 61. TIM10/11 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	TIMx_OR	予約済み																														TIM_RMP	
	リセット値																															0	0

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

15 独立型ウォッチドッグ (IWDG)

15.1 IWDG の概要

デバイスは、2つの内蔵ウォッチドッグペリフェラルを搭載しており、使用上、高い安全レベル、タイミングの正確さ、および柔軟性を兼ね備えています。2つのウォッチドッグペリフェラル（独立型とウィンドウ型）は、ソフトウェア障害による誤動作を検出および解決し、カウンタが与えられたタイムアウト値に達すると、システムリセットまたは割込み（ウィンドウ型ウォッチドッグのみ）をトリガします。


独立型ウォッチドッグ (IWDG) は、独自の低速クロック (LSI) によってクロック供給されるので、メインクロックに障害があってもアクティブなままです。ウィンドウ型ウォッチドッグ (WWDG) のクロックは、APB1 クロックから分周され、また設定可能な時間枠 (time-window) があるので、これをプログラムしてアプリケーション動作の異常な進み・遅れを検出できます。

IWDG は、メインアプリケーションの外部で、完全に独立したプロセスとして実行するウォッチドッグが必要な場合に最適ですが、タイミング精度が低いという制約があります。WWDG は、正確な時間枠内で反応するウォッチドッグが必要なアプリケーションに適しています。ウィンドウ型ウォッチドッグの詳細については、[418 ページのセクション 16](#) を参照してください。

15.2 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 オシレータからクロック供給 (STANDBY および STOP モードで動作可能)
- ダウンカウンタ値が 0x000 に達したときにリセット (ウォッチドッグが有効な場合)

15.3 IWDG の機能説明

 [図 156](#) に、独立型ウォッチドッグモジュールの機能ブロックを示します。

キーレジスタ (IWDG_KR) に値 0xCCCC が書き込まれることによって独立型ウォッチドッグが開始すると、カウンタはリセット値 0xFFFF からカウントダウンを開始します。カウント値の終わり (0x000) に達すると、リセット信号が生成されます (IWDG_reset)。

IWDG_KR レジスタにキー値 0xAAAA が書き込まれると、IWDG_RLR の値がカウンタに再ロードされ、ウォッチドッグのリセットが防止されます。

15.3.1 ハードウェアウォッチドッグ

デバイスのオプションビットを通じて「ハードウェアウォッチドッグ」機能が有効な場合、ウォッチドッグは電源投入時に自動的に有効になり、カウンタがカウントの終わりに達する前にソフトウェアによってキーレジスタへ書き込まれない限り、リセットを生成します。

15.3.2 レジスタのアクセス保護

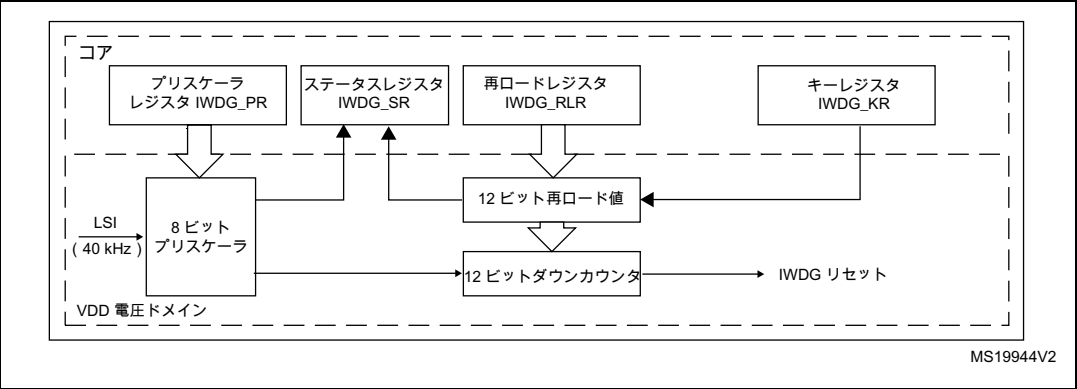
IWDG_PR および IWDG_RLR レジスタへの書込みアクセスは保護されます。これらを変更するには、まず、IWDG_KR レジスタにコード 0x5555 を書き込む必要があります。このレジスタに別の値を書き込むと、シーケンスがブレイクされ、レジスタへのアクセスが再び保護されます。これは、再ロード操作 (0xAAAA の書込み) であることを意味します。

ステータスレジスタは、プリスケアラやダウンカウンタ再ロード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。

15.3.3 デバッグモード

マイクロコントローラがデバッグモードになると（FPU 搭載 Cortex®-M4 コアは停止状態）、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 23.16.2：タイマ、ウォッチドッグ、および I²C のデバッグサポート](#)を参照してください。

図 156. 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 V_{DD} 電圧ドメインに実装されています。

表 62. 最小／最大 IWDG タイムアウト期間（32 kHz）(LSI)⁽¹⁾

プリスケアラ 分周回路	PR[2:0] ビット	タイムアウト最小値 (ms) RL[11:0]= 0x000	タイムアウト最大値 (ms) RL[11:0]= 0xFFFF
/4	0	0.125	512
/8	1	0.25	1024
/16	2	0.5	2048
/32	3	1	4096
/64	4	2	8192
/128	5	4	16384
/256	6		32768

1. これらのタイミングは、32 kHz クロックが供給されたときの値ですが、マイクロコントローラの内蔵 RC 周波数は変化する場合があります。最大値および最小値については、デバイスのデータシートの LSI オシレータ特性の表を参照してください。

15.4 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、[34 ページのセクション 1.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

15.4.1 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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																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（書込み専用、読出しでは 0000h）

これらのビットには、ソフトウェアによって一定間隔でキー値 AAAAh が書き込まれなければなりません。そうしないと、カウンタが 0 に達した時点でウォッチドッグがリセットを生成します。

キー値 5555h を書き込むことによって、IWDG_PR および IWDG_RLR レジスタへのアクセスが可能になります（[セクション 15.3.2](#) を参照）。

キー値 CCCCh を書き込むと、ウォッチドッグが開始します（ハードウェアウォッチドッグオプションが選択されている場合を除く）。

15.4.2 プリスケアラレジスタ (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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
予約済み																														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]** : プリスケアラ分周回路

これらのビットは、書込みアクセス保護されています（[セクション 15.3.2](#) を参照）。カウンタクロックを供給するプリスケア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。プリスケア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アラ値が返されます。このレジスタへの書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。

15.4.3 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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																				RL[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 **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、書込みアクセス保護されています (セクション 15.3.2 を参照)。IWDG_KR レジスタに値 AAAAh が書き込まれるたびにウォッチドッグカウンタにロードされる値を定義するために、ソフトウェアで書き込まれます。ウォッチドッグカウンタは、この値からカウントダウンします。タイムアウトまでの時間は、この値とクロックプリスケアラによって決まります。表 62 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。

15.4.4 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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																											RVU	PVU			
																											r	r			

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **RVU** : ウォッチドッグカウンタ再ロード値の更新

このビットは、再ロード値の更新が進行中であることを示すために、ハードウェアによってセットされます。VDD 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。再ロード値は、RVU ビットがリセットされているときのみ更新できます。

ビット 0 **PVU** : ウォッチドッグプリスケアラ値の更新

このビットは、プリスケアラ値の更新が進行中であることを示すために、ハードウェアによってセットされます。VDD 電圧ドメインでプリスケアラ更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。プリスケアラ値は、PVU ビットがリセットされているときのみ更新できます。

注： 複数の再ロード値またはプリスケアラ値がアプリケーションで使用される場合は、再ロード値を変更する前に RVU ビットがリセットされるまで待つ必要があります、またプリスケアラ値を変更する前に PVU ビットがリセットされるまで待つ必要があります。ただし、プリスケアラ値や再ロード値を更新した後は、RVU または PVU がリセットされるのを待たずに、コード実行を続けることができます (低電力モードに入った場合でも、書込み操作は実行され、完了します)。

15.4.5 IWDG レジスタマップ

次の表に、IWDG レジスタマップとリセット値を示します。

表 63. 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	予約済み																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	予約済み																													PR[2:0]		
	リセット値																														0	0	0
0x08	IWDG_RLR	予約済み																			RL[11:0]												
	リセット値																				1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	IWDG_SR	予約済み																													RVU	PVU	
	リセット値																														0	0	

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

16 ウィンドウ型ウォッチドッグ (WWDG)

16.1 WWDG の概要

ウィンドウ型ウォッチドッグは、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。

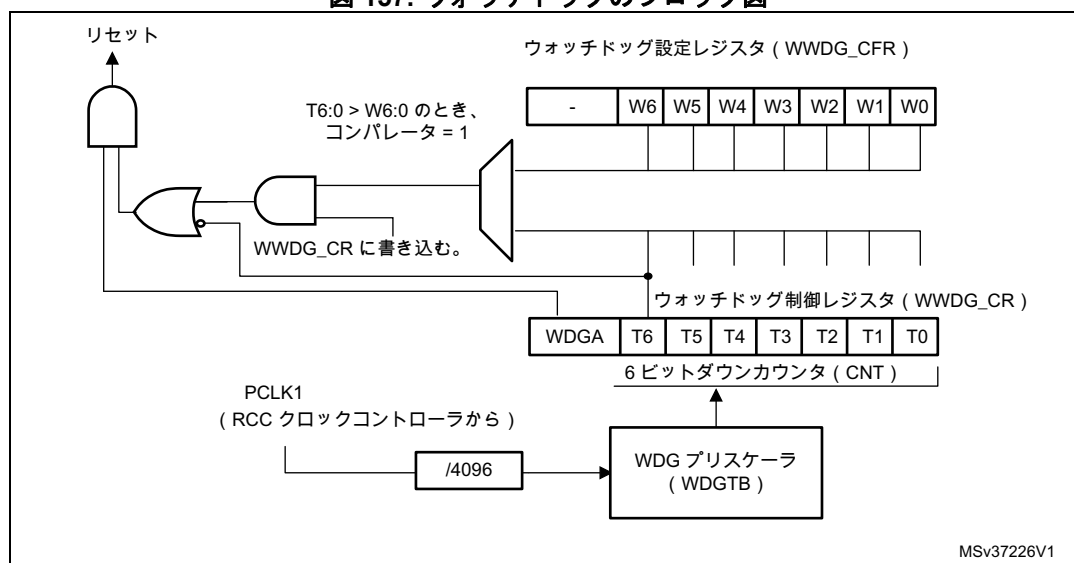
16.2 WWDG の主な機能

- 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 158](#) を参照）。
- 早期ウェイクアップ割込み（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）

16.3 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 に戻った（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 157. 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 レジスタへの書込みを行って、MCU リセットを防ぐ必要があります。この操作は、カウンタの値がウィンドウレジスタの値より小さいときに限られます。WWDG_CR レジスタに格納される値は、0xFF から 0xC0 の間でなければなりません。

ウォッチドッグの有効化

ウォッチドッグはリセット後は常に無効です。これを有効にするには、WWDG_CR レジスタの WDGA ビットをセットします。この後は、リセット以外の方法でウォッチドッグを無効にすることはできません。

ダウンカウンタの制御

このダウンカウンタはフリーランニングであり、ウォッチドッグが無効状態であってもカウントダウンを続けます。ウォッチドッグを有効にするときには、T6 ビットをセットして、ただちにリセットが生成されるのを防ぐ必要があります。

T[5:0] ビットは、ウォッチドッグがリセットを生成するまでの時間遅延を表すインクリメント数を含みます。このタイミングは、WWDG_CR レジスタへの書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 158 参照)。設定レジスタ (WWDG_CFR) は、ウィンドウの上限値を含みます。リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、0x3F よりも大きい間にダウンカウンタを再ロードする必要があります。図 158 に、ウィンドウ型ウォッチドッグのプロセスを示します。

注： T6 ビットを使用して、ソフトウェアリセットを生成することができます (WDGA ビットはセット、T6 ビットはクリアされる)。

高度なウォッチドッグ割込み機能

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込み (EWI) が使用できます。EWI 割込みは、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 割込みが生成され、対応する割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 割込みは、WWDG_SR レジスタの EWIF ビットに“0”を書き込むことによってクリアされます。

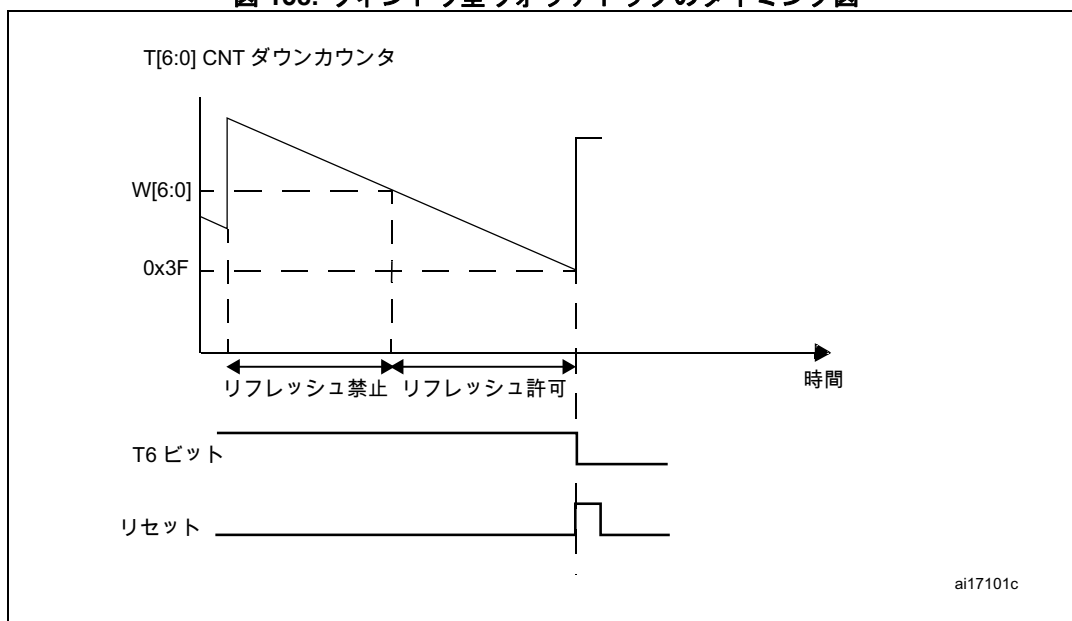
注： 優先順位の高いタスクにおけるシステムロックなどによって EWI 割込みが使用できない場合、最終的には WWDG リセットが生成されます。

16.4 ウォッチドッグタイムアウトをプログラムする方法

図 158 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 158. 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。

$$t_{\text{WWDG}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{WDGTB}[1:0]} \times (T[5:0] + 1) \quad (\text{ms})$$

ここで、

t_{WWDG} : WWDG タイムアウト

t_{PCLK1} : APB1 クロック周期 (ms)

4096 : 内部分周器に対応する値

たとえば、APB1 周波数が 24 MHz と等しい場合、WDGTB[1:0] は 3 にセットされ、T[5:0] は 63 にセットされます。

$$t_{\text{WWDG}} = 1/24000 \times 4096 \times 2^3 \times (63 + 1) = 21.85\text{ms}$$

t_{WWDG} の最小値と最大値については、次の表を参照してください。

表 64. 30 MHz (f_{PCLK1}) でのタイムアウトの最大値および最小値

プリスケアラ	WDGTB	タイムアウト最小値 (μ s) T[5:0] = 0x00	タイムアウト最大値 (ms) T[5:0] = 0x3F
1	0	136.53	8.74
2	1	273.07	17.48
4	2	546.13	34.95
8	3	1092.27	69.91

16.5 デバッグモード

マイクロコントローラがデバッグモードになると (FPU 搭載 Cortex[®]-M4 コアは停止状態)、WWDG カウンタは、DBG モジュールの DBG_WWDG_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 23.16.2: タイマ、ウォッチドッグ、および I²C のデバッグサポート](#)を参照してください。

16.6 WWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[34 ページのセクション 1.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

16.6.1 制御レジスタ（WWDG_CR）

アドレスオフセット：0x00

リセット値：0x0000 007F

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

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDGA** : 有効化ビット

このビットは、ソフトウェアでセットされ、リセット後はハードウェアによってのみクリアされます。
WDGA = 1 のとき、ウォッチドッグはリセットを生成できます。

- 0 : ウォッチドッグは無効です。
- 1 : ウォッチドッグは有効です。

ビット 6:0 **T[6:0]** : 7 ビットカウンタ（MSB から LSB まで）

これらのビットは、ウォッチドッグカウンタの値を含みます。(4096 x 2^{WDGTB[1:0]}) PCLK1 サイクルごとにデクリメントされます。0x40 に達して 0x3F に戻ると（T6 がクリアされると）、リセットが生成されます。

16.6.2 設定レジスタ (WWDG_CFR)

アドレスオフセット : 0x04

リセット値 : 0x0000 007F

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

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **EWI** : 早期ウェイクアップ割込み

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 に達したときに割込みが発生します。
この割込みは、リセット後にハードウェアによってのみクリアされます。

ビット 8:7 **WDGTB[1:0]** : タイマーベース

プリスケアラのタイムベースは、次のように変更できます。

00 : CK カウンタクロック (PCLK1/4096) /1

01 : CK カウンタクロック (PCLK1/4096) /2

10 : CK カウンタクロック (PCLK1/4096) /4

11 : CK カウンタクロック (PCLK1/4096) /8

ビット 6:0 **W[6:0]** : 7 ビットウィンドウ値

これらのビットは、ダウンカウンタと比較されるウィンドウ値を含みます。

16.6.3 ステータスレジスタ (WWDG_SR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み														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、割込みが有効でない場合にもセットされます。

16.6.4 WWDG レジスタマップ

次の表に、WWDG のレジスタマップとリセット値を示します。

表 65. 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	予約済み																							WDGA	T[6:0]							
	リセット値																									0	1	1	1	1	1	1	1
0x04	WWDG_CFR	予約済み																					EWI	WDGTB1	WDGTB0	W[6:0]							
	リセット値																									0	0	0	1	1	1	1	1
0x08	WWDG_SR	予約済み																														EWIF	
	リセット値																																0

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

17 リアルタイムクロック (RTC)

17.1 概要

本リアルタイムクロック (RTC) は、独立した BCD タイマ／カウンタです。この RTC には、時刻クロック／カレンダー、プログラム可能なアラーム割込みを 2 本、割込み機能を備えたプログラム可能な周期的ウェイクアップフラグを搭載しています。RTC には、低電力モードを管理する自動ウェイクアップユニットも搭載されています。

2 つの 32 ビットレジスタには、2 進化 10 進数形式 (BCD) で表現した秒、分、時 (12 時間または 24 時間形式)、曜日、日、月、年が含まれています。サブセカンドの値もバイナリ形式で利用できます。

28 日、29 日 (うるう年)、30 日、31 日の補正は、自動的に行われます。サマータイム補正も行われます。

サブセカンド、秒、分、時、曜日、日付のプログラム可能なアラームを備えた 32 ビットレジスタが追加されています。

クリスタルオシレータ精度の偏差を補正するために、デジタル較正機能が利用可能です。

バックアップドメインリセット後、すべての RTC レジスタは、起こりうる不要な書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。

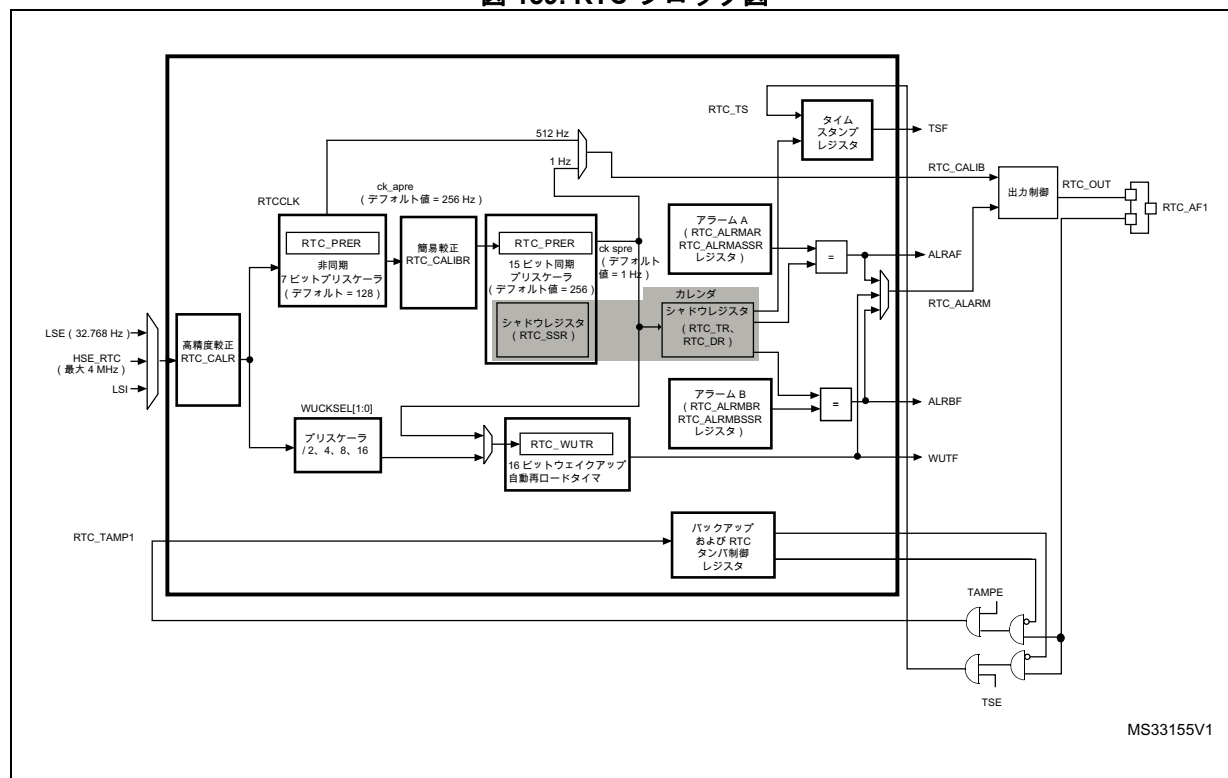
17.2 RTC の主な機能

RTC ユニットの主な機能を以下に示します (図 159 : RTC ブロック図を参照)。

- サブセカンド、秒、分、時 (12 時間または 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。
- 割込み機能を備えた 2 本のプログラム可能なアラーム。アラームは、カレンダー項目のどの組み合わせでもトリガ可能。
- 自動ウェイクアップ割込みをトリガする周期的なフラグを生成する自動ウェイクアップユニット。
- リファレンスクロック検出 : より正確な秒のクロックソース (50 または 60 Hz) の使用で、カレンダーの精度を向上。
- サブセカンドシフト機能を使用する外部クロックとの正確な同期。
- マスク可能な割込み/イベント :
 - アラーム A
 - アラーム B
 - ウェイクアップ割込み
 - タイムスタンプ
 - タンパ検出
- デジタル較正回路 (周期カウンタ修正)
 - 5 ppm の精度
 - 0.95 ppm の精度 (数秒の較正時間範囲で達成)。
- イベントを保存するタイムスタンプ機能 (1 イベント)
- タンパ検出 :
 - タンパイベント (設定可能なフィルタおよび内部プルアップあり)。
- 20 個のバックアップレジスタ (80 バイト)。このバックアップレジスタは、タンパ検出イベントが発生するとリセットされます。
- 次の 2 つの出力のうち 1 つを選択するオルタネート機能出力 (RTC-OUT) :
 - RTC_CALIB : 512 Hz または 1 Hz のクロック出力 (LSE 周波数 32.768 kHz の場合)。
この出力は、RTC_CR レジスタの COE ビットをセットして有効にします。この出力はデバイスの RTC_AF1 機能に送られます。
 - RTC_ALARM (アラーム A、アラーム B またはウェイクアップ)。
この出力は、RTC_CR レジスタの OSEL[1:0] ビットを設定して選択します。この出力はデバイスの RTC_AF1 機能に送られます。
- _RTC 追加機能入力 :
 - RTC_TS : タイムスタンプイベント検出。この出力はデバイスの RTC_AF1 機能に送られます。
 - RTC_TAMP1 : TAMPER1 イベント検出。この出力はデバイスの RTC_AF1 機能に送られます。
 - RTC_REFIN : リファレンスクロック入力 (通常は 50 または 60 Hz の商用電源)。

セクション 8.3.15 : RTC 機能の選択および **STM32F401xx のデータシートのピン定義表**を参照してください。

図 159. RTC ブロック図



1. RTC_AF1 追加機能は PC13 に接続されます。

17.3 RTC の機能説明

17.3.1 クロックとプリスケアラ

RTC クロックソース (RTCCLK) は、LSE クロック、LSI オシレータクロック、HSE クロックのうちから、クロックコントローラを介して選択されます。RTC クロックソースの設定に関する詳細は、[セクション 6: リセットおよびクロック制御 \(STM32F401xB/C および STM32F401xD/E 向け\) \(RCC\)](#) 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 159 : 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_spre} 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 17.3.4：周期的自動ウェイクアップ](#) を参照）。

17.3.2 リアルタイムクロックとカレンダー

RTC カレンダーの時間および日付レジスタには、PCLK1 (APB1 クロック) と同期するシャドウレジスタを通じてアクセスします。同期するため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 17.6.4](#) を参照）。STOP モードおよび STANDBY モードでは、コピーは行われません。これらのモードが終了すると、RTCCLK 2 サイクル以内にシャドウレジスタが更新されます。

アプリケーションが、カレンダーレジスタを読み出す際、実際にはシャドウレジスタの内容にアクセスします。RTC_CR レジスタの BYPSHAD 制御ビットをセットすることにより、カレンダーレジスタに直接アクセスできます。デフォルトでは、このビットはクリアされており、ユーザはシャドウレジスタにアクセスします。

RTC_SSR、RTC_TR または RTC_DR レジスタを BYPSHAD = 0 の状態で読み出す際は、APB クロックの周波数 (f_{APB}) は、RTC クロック (f_{RTCCLK}) の周波数の 7 倍以上でなければなりません。

シャドウレジスタは、システムリセットによってリセットされます。

17.3.3 プログラム可能なアラーム

本 RTC ユニットには、2 本のプログラム可能なアラームである、アラーム A およびアラーム B を搭載しています。

このプログラム可能なアラーム機能は、RTC_CR レジスタの ALRAIE ビットおよび ALRBIE ビットを通じて有効にします。ALRAF および ALRBF フラグは、カレンダーのサブセカンド、秒、分、時、日または曜日がそれぞれアラームレジスタ RTC_ALRMASR/RTC_ALRMAR および RTC_ALRMBSSR/RTC_ALRMBR にプログラムされている値と一致すると 1 にセットされます。各カレンダー項目は、RTC_ALRMAR および RTC_ALRMBR レジスタの MSKx ビット、および RTC_ALRMASR および RTC_ALRMBSSR レジスタの MASKSx ビットで個別に選択できます。アラーム割込みは、RTC_CR レジスタの ALRAIE ビットおよび ALRBIE ビットを通じて有効にします。

アラーム A およびアラーム B (RTC_CR レジスタの OSEL[1:0] ビットで有効になっている場合) は、RTC_ALARM 出力に送ることができます。RTC_ALARM の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

注意 : 秒の項目が選択されている場合 (RTC_ALRMAR または RTC_ALRMBR で MSK0 ビットがリセットされている)、正しい動作を確実にするため、RTC_PRER レジスタでセットされる同期プリスケアラの分周比は 3 以上である必要があります。

17.3.4 周期的自動ウェイクアップ

周期的ウェイクアップフラグは、16 ビットのプログラム可能な自動再ロードダウンカウンタによって生成されます。ウェイクアップタイマの範囲は 17 ビットまで拡張できます。

ウェイクアップ機能は、RTC_CR レジスタの WUTE ビットを通じて有効にします。

ウェイクアップタイマクロック入力には、次のものが使用できます。

- 2、4、8、または 16 分周した RTC クロック (RTCCLK)
RTCCLK が LSE (32.768kHz) である場合、最小分解能 61 μ s で、ウェイクアップ割込み周期を 122 μ s から 32 s の範囲で設定できます。
- ck_spre (通常は 1 Hz の内部クロック)
ck_spre 周波数が 1 Hz の場合、1 秒の分解能でウェイクアップ時間を 1 秒からおよそ 36 時間までの範囲で設定できます。このプログラム可能な広い時間範囲は、2 つの部分に分かれます。
 - WUCKSEL[2:1] = 10 の場合は 1 秒から 18 時間、
 - WUCKSEL[2:1] = 11 の場合は約 18 時間から 36 時間です。後者の場合、16 ビットカウンタの現在値に 216 が加算されます。初期化シーケンスが完了すると (431 ページの [ウェイクアップタイマのプログラミング](#) を参照)、タイマがカウントダウンを開始します。ウェイクアップ機能が有効な場合、低電力モードでもカウントダウンはアクティブのままとなります。さらに、カウンタがゼロに到達すると、RTC_ISR レジスタの WUTF フラグがセットされ、ウェイクアップカウンタが再ロード値 (RTC_WUTR レジスタ値) で自動的に再ロードされます。

その後、WUTF フラグはソフトウェアでクリアする必要があります。

RTC_CR2 レジスタの WUTIE ビットをセットして周期的ウェイクアップ割込みを有効にすると、デバイスは低電力モードを終了できます。

周期的なウェイクアップフラグは、RTC_CR レジスタの OSEL[0:1] ビットを通じて有効になっている場合に限り、RTC_ALARM 出力に送ることができます。RTC_ALARM の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

低電力モード (SLEEP、STOP、STANDBY) と同様に、システムリセットもウェイクアップタイマには影響しません。

17.3.5 RTC の初期化と設定

RTC レジスタアクセス

RTC レジスタは、32 ビットのレジスタです。APB インターフェイスは、RTC レジスタアクセスに 2 ウェイトステートを挿入します。ただし、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 電源制御レジスタ (PWR_CR) の DBP ビットへの不要な書き込みアクセスから保護されます。RTC レジスタ書き込みアクセスを可能にするには、DBP ビットをセットする必要があります。

バックアップドメインリセット後、すべての RTC レジスタは書き込み保護されます。RTC レジスタへの書き込みは、書き込み保護レジスタ RTC_WPR にキーを書き込むことにより有効になります。

RTC_ISR[13:8]、RTC_TAFCR、RTC_BKPxR を除くすべての RTC レジスタの書き込み保護を解除するには、次のステップが必要です。

1. RTC_WPR レジスタに“0xCA”を書き込みます。
2. RTC_WPR レジスタに“0x53”を書き込みます。

誤ったキーを書き込むと、書き込み保護が再度アクティブになります。

保護メカニズムは、システムリセットの影響を受けません。

カレンダーの初期化と設定

時間形式やプリスケアラ設定を含むカレンダー時刻と日付の初期値をプログラムするには、次のシーケンスが必要です。

1. RTC_ISR レジスタで INIT ビットを 1 にセットして、初期化モードに入ります。このモードでは、カレンダーカウンタが停止し、その値を更新することができます。
2. RTC_ISR レジスタの INITF ビットをポーリングします。INITF が 1 にセットされると、初期化フェーズモードに入ります。これには RTCCLK クロック約 2 周期を必要とします (クロック同期のため)。
3. カレンダーカウンタのための 1 Hz クロックを生成するには、まず RTC_PRER レジスタで同期プリスケアラ分周比をプログラムし、その後非同期プリスケアラ分周比をプログラムします。2 つの項目のうち 1 つだけを変更する必要がある場合でも、RTC_PRER レジスタに対し書き込みアクセスを 2 回に分けて行う必要があります。
4. シャドウレジスタ (RTC_TR および RTC_DR) に時刻と日付の初期値をロードし、RTC_CR レジスタの FMT ビットを介して時間形式 (12 時間または 24 時間) を設定します。
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 INITIS フラグを読み出し、カレンダーが初期化されたか否かを確認できるようになります。このフラグが 0 である場合、カレンダーの年の項目がバックアップドメインリセットデフォルト値 (0x00) にセットされているため初期化されていません。

初期化後にカレンダーを読み出すには、まずソフトウェアで RTC_ISR レジスタの RSF フラグがセットされていることを確認する必要があります。

サマータイム

サマータイム管理は、RTC_CR レジスタの SUB1H ビット、ADD1H ビット、BKP ビットを介して行われます。

SUB1H または ADD1H を使用すると、ソフトウェアは初期化手順を踏まずに 1 度の操作で、カレンダーから 1 時間引いたり足したりすることができます。

さらに、ソフトウェアは BKP ビットを使用してこの操作を記憶することができます。

アラームのプログラミング

プログラム可能なアラーム（アラーム A またはアラーム B）をプログラムまたは更新するには、同様の手順を踏む必要があります。

1. RTC_CR の ALRAE または ALRBIE をクリアしてアラーム A またはアラーム B を無効にします。
2. RTC_ISR の ALRAWF または ALRBWF がセットされてアラームレジスタへのアクセスが許可されていることが確認できるまで、これらのビットをポーリングします。これには RTCCLK クロック 1~2 周期を必要とします（クロック同期のため）。
3. アラーム A レジスタまたはアラーム B レジスタ（RTC_ALRMSSR/RTC_ALRMAR または RTC_ALRMBSSR/RTC_ALRMBR）をプログラムします。
4. RTC_CR レジスタで ALRAE または ALRBIE をセットしてアラーム A またはアラーム B を再び有効にします。

注： RTC_CR レジスタの各変更は、クロック同期のため RTCCLK クロック約 2 周期後に有効になります。

ウェイクアップタイマのプログラミング

ウェイクアップタイマ自動再ロード値（RTC_WUTR の WUT[15:0]）の設定または変更には、次の手順が必要です。

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 周期を必要とします（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 の 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。

17.3.6 カレンダーの読出し

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_{PCLK1} ）が RTC クロック周波数（ f_{RTCCLK} ）の 7 倍以上である必要があります。これにより、同期メカニズムの安全な動作が保証されます。

APB1 クロック周波数が RTC クロック周波数の 7 倍未満である場合、ソフトウェアによってカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC_TR の 2 回目の読出しが 1 回目の読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 回目の読出しアクセスを行う必要があります。どの場合も、APB1 クロック周波数は必ず RTC クロック周波数以上である必要があります。

RTC_ISR レジスタの RSF ビットは、カレンダーレジスタが RTC_SSR、RTC_TR、および RTC_DR シャドウレジスタにコピーされるたびにセットされます。コピーは、2 RTCCLK サイクルごとに行われます。3 つの値における一貫性を確実にするため、RTC_SSR または RTC_TR を読み出すと、高次の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出しアクセスを行う場合、最初のカレンダー読出し後に 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つてから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再度読み出す必要があります。

低電力モード (STOP または STANDBY) からのウェイクアップ後は、RSF をソフトウェアでクリアする必要があります。その後、ソフトウェアは、いまだ RSF がセットされるまで待つてから、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。

RSF ビットは、ウェイクアップ後にクリアする必要がありますが、低電力モードに入る前には、その必要はありません。

注： システムリセット後、ソフトウェアは RSF がセットされるまで待つてから、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 (430 ページの [カレンダーの初期化と設定](#) を参照) 後、ソフトウェアは RSF がセットされるまで待つてから、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 17.3.8 : 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つてから、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 レジスタの BYPSHAD 制御ビットがセットされている場合 (バイパスシャドウレジスタ)

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がありません。シャドウレジスタは低電力モード (STOP または STANDBY) で更新されないため、これは低電力モード終了後に特に有用です。

BYPSHAD ビットが 1 にセットされている場合、レジスタへの 2 回の読み出しアクセス間で RTCCLK エッジが発生した場合は、さまざまなレジスタ間で互いに不整合が起きる場合があります。さらに、読み出し操作中に RTCCLK エッジが発生した場合、レジスタの 1 つが不正な値となる場合があります。ソフトウェアはすべてのレジスタを 2 回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要があります。その代わりに、ソフトウェアはカレンダーレジスタの最下位の数値を 2 回比較するだけで構いません。

注： BYPSHAD = 1 の間、カレンダーレジスタの読み出し命令が完了するには 1 APB サイクルだけ余計に必要となります。

17.3.7 RTC のリセット

カレンダーシャドウレジスタ (RTC_SSR、RTC_TR、RTC_DR) および RTC ステータスレジスタ (RTC_ISR) の一部のビットは、利用可能なすべてのシステムリセットソースによってデフォルト値にリセットされます。

逆に、次のレジスタは バックアップドメインリセットによってそれぞれのデフォルト値にリセットされ、システムリセットの影響は受けません。RTC の現在のカレンダーレジスタ、RTC 制御レジスタ (RTC_CR)、プリスケアラレジスタ (RTC_PRER)、RTC 較正レジスタ (RTC_CALIBR または RTC_CALR)、RTC シフトレジスタ (RTC_SHIFTR)、RTC タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、および RTC_TSDR)、RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)、RTC バックアップレジスタ (RTC_BKPxR)、ウェイクアップタイマレジスタ (RTC_WUTR)、およびアラーム A およびアラーム B レジスタ (RTC_ALRMASR/RTC_ALRMAR および RTC_ALRMBSSR/RTC_ALRMBR)。

さらに、RTC はリセットソースがバックアップドメインリセットと異なっていれば、システムリセット中も動作し続けます。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。

17.3.8 RTC の同期

RTC は、高精度でリモートクロックと同期できます。サブセカンド項目 (RTC_SSR または RTC_TSSSR) を読み出すと、リモートクロックによって維持されている時刻と RTC 間の正確なオフセットが計算できます。その後、RTC_SHIFTR を使用してほんの一瞬クロックを「シフト」することによって RTC を調整し、このオフセットを取り除くことができます。

RTC_SSR には、同期プリスケアラのカウンタの値が入っています。これにより、RTC によって維持されている正確な時刻を $1/(\text{PREDIV_S} + 1)$ 秒の分解能まで $1/(\text{PREDIV_S} + 1)$ 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 μ 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。

17.3.9 RTC リファレンスクロック検出

RTC カレンダの更新は、リファレンスクロックである RTC_REFIN に同期させることができます。通常は商用電源 (50 または 60 Hz) です。RTC_REFIN リファレンスクロックには、32.768 kHz LSE クロックよりも高い精度が必要です。RTC_REFIN 検出を有効にした際 (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) の誤差の補正に使用されます。

各 1 Hz クロックエッジは、一番近いリファレンスクロックエッジ (所与の時間枠内に見つかった場合) と比較されます。ほとんどの場合、2 つのクロックエッジは正しく整列しています。LSE クロックが不正確なために 1 Hz のクロックがずれた場合、RTC は 1 Hz のクロックを少しシフトさせ、その後の 1 Hz のクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダはリファレンスクロックと同様に正確になります。

RTC は、32.768 kHz クォーツから生成される 256 Hz クロック (ck_apre) を使用して、リファレンスクロックソースがあるかどうかを検出します。検出は各カレンダ更新 (1 秒ごと) 程度の時間枠で行われます。最初のリファレンスクロックエッジを検出する際、この時間枠は ck_apre 7 周期に等しくなります。その後のカレンダ更新では、ck_apre 3 周期より短い時間枠が使用されます。

リファレンスクロックがこの時間枠内で検出されるたびに、ck_apre クロックを出力する非同期プリスケアラは強制的に再ロードされます。プリスケア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。

リファレンスクロック検出を有効にした場合、PREDIV_A および PREDIV_S をデフォルト値にセットする必要があります。

- PREDIV_A = 0x007F
- PREDIV_S = 0x00FF

注： リファレンスクロック検出は、STANDBY モードでは利用できません。

注意： リファレンスクロック検出機能は、簡易デジタル較正と併用することはできません。REFCKON = 1 の場合、RTC_CALIBR は 0x0000 0000 に保持する必要があります。

17.3.10 RTC の簡易デジタル較正

デジタル較正手法には、簡易較正と高精度較正の 2 種類があります。簡易較正を実行するには、[セクション 17.6.7 : RTC 較正レジスタ \(RTC_CALIBR\)](#) を参照してください。

これら 2 種類の較正方法は同時に使用することは想定されていませんので、アプリケーションは 2 種類の較正方法のうち、いずれかを選択する必要があります。簡易較正は、互換性を確保するために搭載されています。高精度較正を実行するには、[セクション 17.3.11 : RTC の高精度デジタル較正](#)および[セクション 17.6.16 : RTC 較正レジスタ \(RTC_CALR\)](#) を参照してください。

簡易デジタル較正を使用すると、非同期プリスケアラ (ck_apre) の出力でクロックサイクルを追加 (正の較正) またはマスキング (負の較正) して、クリスタルの精度を補償することができます。

正および負の較正は、RTC_CALIBR レジスタの DCS ビットをそれぞれ“0”および“1”にセットして選択します。

正の較正が有効 (DCS = “0”) の場合、2 x DC 分の間、毎分 (ck_apre 約 15360 サイクル) ck_apre 2 サイクルが追加されます。これによりカレンダーの更新更新速度が早くなり、それによって有効な RTC 周波数は少し高くなります。

負の較正が有効 (DCS = “1”) の場合、2 x DC 分の間、毎分 (ck_apre 約 15360 サイクル) ck_apre 1 サイクルが取り除かれます。これによりカレンダーの更新速度が遅くなり、それによって有効な RTC 周波数は少し低くなります。

DC は RTC_CALIBR レジスタの DC[4:0] ビットで設定されます。この数値は 0 から 31 の範囲で設定可能であり、0 から 62 の範囲の時間間隔 (2 x DC) に相当します。

簡易デジタル較正は、初期化モードの期間でのみ設定でき、INIT ビットがクリアされると設定した動作を開始します。較正サイクルは全体で 64 分間続きます。64 分間のサイクルの最初の 2 x DC 分が、上記の設定のように変更されます。

負の較正は、約 2 ppm の分解能で実行できますが、正の較正は約 4ppm の分解能になります。最大較正範囲は、-63 ppm から 126 ppm です。

較正は LSE または HSE クロックのどちらに対してでも実行可能です。

注意： PREDIV_A < 6 の場合、デジタル較正が正しく動作しない場合があります。

RTCCLK=32.768kHz かつ PREDIV_A+1=128 の場合

次の説明は、ck_apre 周波数が、LSE クロックの公称周波数 32.768 kHz をもとに、PREDIV_A を 127 (デフォルト値) に設定して得られる 256Hz であることを前提としています。

ck_spre クロック周波数は、64 分間のサイクルの最初の 2 × DC 分間でのみ変更されます。たとえば、DC が 1 である場合、最初の 2 分のみが変更されます。つまり、各 ck_apre サイクルが 128 RTCCLK サイクル (PREDIV_A+1=128) に相当するならば、各 64 分サイクルの最初の 2 × DC 分に対して、毎分 1 回だけ、1 秒が RTCCLK 256 サイクル分短くなる、あるいは、RTCCLK 128 サイクル分長くなります。

したがって、各校正ステップには、125829120 RTCCLK サイクル (64 分 × 60 秒 / 分 × 32768 サイクル / 秒) ごとに 512 オシレータサイクルを追加または 256 オシレータサイクルを差し引くという効果があります。これは、校正ステップにつき +4.069 ppm または -2.035 ppm と等価です。その結果、校正分解能は毎月 +10.5 秒または -5.27 秒となり、総校正範囲は毎月 +5.45 ~ -2.72 分となります。

クロック偏差を測定するため、校正用に 512Hz クロックが出力されます。[セクション 17.3.14 : 校正クロック出力](#)を参照してください。

17.3.11 RTC の高精度デジタル校正

RTC 周波数の精度は、-487.1 ppm から +488.5 ppm の範囲で、分解能約 0.954 ppm でデジタル的に校正できます。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 が十分に校正されるように、かなり広範に分散して行われます。

この高精度デジタル校正は、入力周波数が 32768 Hz の場合、RTCCLK 約 2^{20} パルスのサイクルまたは 32 秒の間で行われます。[このサイクルは、RTCCLK によって駆動される 20 ビットカウンタ、cal_cnt\[19:0\] によって維持されます。](#)

高精度校正レジスタ (RTC_CALR) によって、32 秒サイクル中にマスクされる RTCCLK クロックサイクル数を指定します。

- CALM[0] ビットを 1 にセットすると、32 秒サイクルの中でちょうど 1 サイクルがマスクされます。
- 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 サイクルがマスクされます。

注 : CALM[8:0] (RTC_CALRx) によって、32 秒サイクル中にマスクされる RTCCLK 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、[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 サイクルがマスクされます。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 サイクルがマスクされま [す \(cal_cnt = 0xFF800\)。](#)

CALM では、細かい分解能で RTC 周波数を最大 487.1 ppm 負の方向に調整することができ、CALP ビットでは周波数を 488.5 ppm 正の方向に調整することができます。CALP を 1 にセットすることにより、実質上は、RTCCLK 2^{11} サイクルごとに、RTCCLK パルスが 1 パルス追加で挿入されます。すなわち、32 秒サイクルごとに 512 クロックが追加されることとなります。

CALM を CALP と合わせて使用すると、32 秒サイクルの間に RTCCLK -511 から +512 サイクルまでのオフセットが追加でき、これは約 0.954 ppm の分解能で校正範囲 -487.1 ~ +488.5 ppm に換算されます。

有効校正周波数 (FCAL) を入力周波数 (FRTCCLK) に対して求める計算式は次のとおりです。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (\text{CALP} \times 512 - \text{CALM}) / (2^{20} + \text{CALM} - \text{CALP} \times 512)]$$

PREDIV_A < 3 の場合の較正

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 未満の値に設定された場合、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 秒サイクルの間に 255~256 クロックパルス (243.3~244.1 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。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (256 - \text{CALM}) / (2^{20} + \text{CALM} - 256)]$$

この場合、RTCCLK が正確に 32768.00 Hz であれば、CALM[7:0] が 0x100 (CALM 設定範囲の中間値) と等しくなるのが正しい設定です。

RTC 較正值の確認

RTC の精度は、RTCCLK の正確な周波数を測定し、正しい CALM 値および CALP 値を計算することにより向上することができます。オプションの 1 Hz 出力が搭載されており、アプリケーションによって RTC 精度の測定と確認を行うことができます。ある時間間隔で RTC の周波数を精密に測定すると、デジタル較正サイクルを測定周期とどのように合わせているかにより、測定期間中に最大 2 RTCCLK クロックサイクルの測定誤差が生じます。

ただし、この測定誤差は、測定周期が較正サイクル周期と同じ長さであれば排除できます。この場合、観測される唯一の誤差はデジタル較正の分解能による誤差となります。

- デフォルトでは、較正サイクル周期は 32 秒です。
このモードを使用して正確に 32 秒で 1 Hz 出力の精度を測定すると、その精度は 0.477 ppm (較正分解能の制限により 32 秒で 0.5 RTCCLK サイクル) 以内となることが保証されます。
- RTC_CALR レジスタの CALW16 ビットを 1 にセットして、較正サイクル周期を強制的に 16 秒にすることができます。
この場合、RTC 精度は最大誤差 0.954 ppm (16 秒で 0.5 RTCCLK サイクル) で 16 秒間で測定できます。ただし、較正分解能が下がるため、長期的な RTC 精度もまた 0.954 ppm に下がります。CALW16 が 1 にセットされると、CALM[0] ビットは 0 のままとなります。
- RTC_CALR レジスタの CALW8 ビットを 1 にセットして、較正サイクル周期を強制的に 8 秒にすることができます。
この場合、RTC 精度は最大誤差 1.907 ppm (8 秒で 0.5 RTCCLK サイクル) で 8 秒で測定できます。長期的な RTC 精度もまた 1.907 ppm に下がります。CALW8 が 1 にセットされると、CALM[1:0] ビットは 00 のままとなります。

動作中の再較正

次の処理を実施することにより、RTC_ISR/INITF = 0 の間でも、較正レジスタ (RTC_CALR) を動作中に更新することができます。

1. RTC_ISR/RECALPF (再較正保留フラグ) をポーリングします。
2. このフラグが 0 にセットされている場合は、必要に応じて新しい値を RTC_CALR に書き込みます。すると、RECALPF が自動的に 1 にセットされます。
3. RTC_CALR への書き込み動作後 ck_apre 3 サイクル以内に、新しい較正設定が有効になります。

17.3.12 タイムスタンプ機能

タイムスタンプは、RTC_CR レジスタの TSE ビットを 1 にセットすることにより有効になります。

TIMESTAMP に追加機能が設定されているピンでタイムスタンプイベントが検出されると、カレンダーは、タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、RTC_TSDR) に保存されます。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込みが生成されます。

タイムスタンプフラグ (TSF) がすでにセットされている間に新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) がセットされ、タイムスタンプレジスタ (RTC_TSTR および RTC_TSDR) は、その前のイベントの結果を維持します。

注： 同期処理のため、TSF はタイムスタンプイベント発生から ck_apre 2 サイクル後にセットされます。一方、TSOVF のセットに遅延はありません。これは、2 つのタイムスタンプイベントの発生したタイミングが近い場合、TSF がまだ“0”であっても TSOVF が“1”と検出される可能性があることを意味します。よって、TSOVF のポーリングは TSF がセットされた後に実施することをお勧めします。

注意： TSF ビットのクリア処理を行った直後にタイムスタンプイベントが発生した場合、TSF ビットおよび TSOVF ビットの両方がセットされます。同時に発生するタイムスタンプイベントのマスキングを回避するため、アプリケーションは TSF がすでに“1”と読み出されていない限り“0”クリアの処理を行ってはなりません。

オプション機能として、タンパイイベントによってタイムスタンプイベントを記録することもできます。TAMPTS 制御ビットの詳細については、[セクション 17.6.17: RTC タンパおよびオルタネート機能設定レジスタ \(RTC_TAFRCR\)](#) を参照してください。タイムスタンプイベントがフィルタモードで設定されているタンパイイベントと同じピン上にある場合 (TAMPFLT がゼロ以外の値にセットされている場合)、RTC_TAFRCR レジスタの TAMPTS を“1”にセットすることにより、タンパ検出イベントモードにおけるタイムスタンプを選択する必要があります。

TIMESTAMP 追加機能

TIMESTAMP 追加機能は RTC_AF1 に配置されます。

17.3.13 タンパ検出

1 つのタンパ検出入力を利用できます。これらの入力は、フィルタを通してエッジ検出またはレベル検出のどちらにも設定できます。

RTC バックアップレジスタ

バックアップレジスタ (RTC_BKPxR) は、20 個の 32 ビットレジスタであり、80 バイトのユーザーアプリケーションデータを保存できます。これらのレジスタは、バックアップドメインに搭載され、V_{DD} 電源が遮断された場合、V_{BAT} によって電源が供給されます。これらは、システムリセットや STANDBY モードからのウェイクアップではリセットされません。バックアップドメインリセットによりリセットされます。

このバックアップレジスタは、タンパ検出イベントが発生するとリセットされます ([セクション 17.6.20 : RTC バックアップレジスタ \(RTC_BKPxR\)](#) および 438 ページの [タンパ検出の初期化](#) を参照)。

タンパ検出の初期化

タンパ検出入力、RTC_ISR レジスタの TAMP1F フラグに関連付けられています。入力は、RTC_TAFCR レジスタの TAMP1E ビットを 1 にセットすることにより有効にできます。

タンパ検出イベントが発生すると、すべてのバックアップレジスタ (RTC_BKPxR) がリセットされます。

RTC_TAFCR レジスタの TAMPIE ビットをセットすることにより、タンパ検出イベント発生時に割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 RTC_ISR の TSF ビットまたは TSOVF ビットがセットされます。タンパフラグレジスタ (TAMP1F) が TSF または TSOVF がセットされるのと同時にセットされます。

タンパ入力でのエッジ検出

TAMPFLT ビットが“00”の場合、TAMPER ピンは、該当する TAMPxTRG ビットに応じて、立ち上りエッジまたは立ち下りエッジが観測されるとタンパ検出イベント (RTC_TAMP[2:1]) を生成します。エッジ検出を選択すると、TAMPER 入力の内部プルアップ抵抗が無効になります。

注意 : タンパ検出イベントを確実に検出するため、TAMPERx ピンが有効になる前にタンパ検出イベントが発生した場合でも検出できるよう、エッジ検出に使用される信号は TAMPxE と AND がとられます。

- TAMPxTRG = 0 の場合 : タンパ検出が有効になる (TAMPxE ビットが 1 にセットされる) 前に TAMPERx 追加機能がすでにハイであれば、TAMPxE がセットされた後に TAMPERx に立ち上がりエッジ入力がなくとも TAMPERx が有効になった直後にタンパイベントが検出されます。
- TAMPxTRG = 1 の場合 : タンパ検出が有効になる前に TAMPERx 追加機能がすでにローであれば、TAMPxE がセットされた後に TAMPERx に立ち下がりエッジ入力がなくとも TAMPERx が有効になった直後にタンパイベントが検出されます。

タンパイベントが検出されクリアされた後に、バックアップレジスタ (RTC_BKPxR) を再プログラムする場合には、事前に、TAMPERx 追加機能を無効にしてから再度有効に (TAMPxE を 1 にセット) する処理を行う必要があります。これによって、TAMPERx の値がタンパ検出を示している期間中に、アプリケーションがバックアップレジスタにデータを書き込むのを防ぎます。これは、TAMPERx 追加機能でのレベル検出と同じことです。

注：タンパ検出は、 V_{DD} 電源がオフのときでも有効です。バックアップレジスタの不要なリセットを避けるには、TAMPER 追加機能が設定されているピンを外部で適切な信号レベルに接続しておく必要があります。

タンパ入力でのフィルタを使ったレベル検出

フィルタを使ったレベル検出は、TAMPFLT を 0 以外の値にセットすることにより行われます。タンパ検出イベントは、2、4 または 8 (TAMPFLT による) 回のいずれかの連続したサンプルが TAMPxTRG ビット (TAMP1TRG) で指定するレベルで観測されると発生されます。

TAMPER 入力は、TAMPPUDIS が 1 にセットされて無効な状態になっていない限り、その状態がサンプリングされる前に I/O の内部プルアップ抵抗でプリチャージされています。プリチャージの継続時間は TAMPPRCH ビットによって決定され、タンパ入力に、より大きな容量を持たせることができます。

タンパ検出の遅延時間と、プルアップによる電力消費との間のトレードオフは、TAMPFREQ を使用してレベル検出のサンプリング周波数を決定することにより、最適化できます。

注：プルアップ抵抗の電気的特性については、データシートを参照してください。

TAMPER 追加機能検出

TAMPER1 追加機能は RTC_AF1 ピンに配置されます。

17.3.14 較正クロック出力

RTC_CR レジスタで COE ビットが 1 にセットされると、RTC_CALIB デバイス出力にリファレンスクロックが供給されます。RTC_CR レジスタの COSEL ビットがリセットされ、かつ PREDIV_A = 0x7F である場合、RTC_CALIB 周波数は $f_{RTCCLK}/64$ です。これは 32.768 kHz の RTCCLK 周波数に対する 512 Hz の較正出力に相当します。

RTC_CALIB 出力は、RTC_CALIBR レジスタでプログラムされる較正值に影響されません。立ち下が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 $f_{RTCCLK}/(256 * (PREDIV_A+1))$ となります。これは、RTCCLK 周波数が 32.768 kHz で、プリスケアラデフォルト値 (PREDIV_A = 0x7F、PREDIV_S = 0xFF) に対する 1 Hz の較正出力に相当します。

較正クロック出力時の RTC_AF1 オルタネート機能

RTC_CR レジスタの COE ビットが 1 にセットされると、較正オルタネート機能 (RTC_CALIB) が、RTC_OUT で有効になります。

注：RTC_CALIB または RTC_ALARM が選択されると、RTC_OUT は自動的に出力のオルタネート機能に設定されます。

17.3.15 アラーム出力

アラーム出力については、3 つの機能が選択できます。ALRAF。これらの機能は、RTC_ISR レジスタの該当するフラグの内容を反映します。

RTC_CR レジスタの OSEL[1:0] 制御ビットを使用して RTC_AF1 でアラームオルタネート機能出力 (RTC_ALARM) を有効にし、RTC_ALARM の出力となる機能を選択します。

RTC_CR の POL 制御ビットは、POL が 1 にセットされているときに選択されたフラグビットの逆が出力されるよう、出力の極性を決定します。

アラームオルタネート機能出力

RTC_ALARM は、RTC_TAFCR レジスタの ALARMOUTTYPE 制御ビットを使用して、出力オープンドレインまたは出力プッシュプルに設定できます。

注： RTC_ALARM が有効になると、この設定は、RTC_CALIB (COE ビットは RTC_AF1 では無視) の設定よりも優先されます。

RTC_CALIB または RTC_ALARM が選択されると、RTC_OUT は自動的に出力のオルタネート機能に設定されます。

17.4 RTC と低電力モード

表 66. 低消費電力モードが 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。

17.5 RTC 割込み

すべての RTC 割込みは、EXTI コントローラに接続されています。

RTC アラーム割込みを有効にするには、次のシーケンスが必要です。

1. 割込みモードで EXTI ライン 17 を設定して有効にし、立ち上がりエッジ検出を選択します。
2. NVIC で RTC_Alarm IRQ チャンネルを設定し、有効にします。
3. RTC アラーム (アラーム A またはアラーム B) を発生させるように RTC を設定します。

RTC ウェイクアップ割込みを有効にするには、次のシーケンスが必要です。

1. EXTI ライン 22 を割込みモードに設定して有効にし、立ち上がりエッジ検出を選択します。
2. NVIC で RTC_WKUP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC ウェイクアップタイマイイベントを発生させるように設定します。

RTC タンパ割込みを有効にするには、次のシーケンスが必要です。

1. EXTI ライン 21 を割込みモードに設定して有効にし、立ち上がりエッジ検出を選択します。
2. NVIC で TAMP_STAMP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC タンパイイベントを検出するように設定します。

RTC タイムスタンプ割込みを有効にするには、次のシーケンスが必要です。

1. EXTI ライン 21 を割込みモードに設定して有効にし、立ち上がりエッジ検出を選択します。
2. NVIC で TAMP_STAMP IRQ チャンネルを設定し、有効にします。
3. RTC が RTC タイムスタンプイベントを検出するように設定します。

表 67. 割込み制御ビット

割込みイベント	イベント フラグ	有効化 制御ビット	SLEEP モードの 終了	STOP モードの 終了	STANDBY モードの 終了
アラーム A	ALRAF	ALRAIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
アラーム B	ALRBF	ALRBIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
ウェイクアップ	WUTF	WUTIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
タイムスタンプ	TSF	TSIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
タンパ 1 検出	TAMP1F	TAMPIE	あり	あり ⁽¹⁾	あり ⁽¹⁾

1. STOP モードおよび STANDBY モードからのウェイクアップは、RTC クロックソースが LSE または LSI のときのみ可能です。

17.6 RTC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの[セクション 1.1](#)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。

17.6.1 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。430 ページの[カレンダーの初期化と設定](#)および 431 ページの[カレンダーの読出し](#)を参照してください。

アドレスオフセット : 0x00

バックアップドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31-24 予約済み

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.2 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。430 ページの[カレンダーの初期化と設定](#)および 431 ページの[カレンダーの読出し](#)を参照してください。

アドレスオフセット : 0x04

バックアップドメインリセット値 : 0x0000 2101

システムリセット : BYPSHAD = 0 の場合、0x0000 2101 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み								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]				予約済み		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

ビット 23:20 **YT[3:0]** : BCD 形式での年の十の位

ビット 19:16 **YU[3:0]** : BCD 形式での年の一の位

ビット 15:13 **WDU[2:0]** : 曜日

000 : 禁止
001 : 月曜日
...
111 : 日曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.3 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
予約済み								COE	OSEL[1:0]		POL		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	DCE	FMT	BYPSHAD	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	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 17.3.14: 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ときにセットすると、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、冬時間変更に使えます。
- 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えます。

ビット 15 **TSIE** : タイムスタンプ割込みイネーブル

- 0 : タイムスタンプ割込みは無効です。
- 1 : タイムスタンプ割込みは有効です。

ビット 14 **WUTIE** : ウェイクアップタイマ割込みイネーブル

- 0 : ウェイクアップタイマ割込みは無効です。
- 1 : ウェイクアップタイマ割込みは有効です。

ビット 13 **ALRBIE** : アラーム B 割込みイネーブル

- 0 : アラーム B 割込みは無効です。
- 1 : アラーム B 割込みは有効です。

ビット 12 **ALRAIE** : アラーム A 割込みイネーブル

- 0 : アラーム A 割込みは無効です。
- 1 : アラーム A 割込みは有効です。

ビット 11 **TSE** : タイムスタンプイネーブル

- 0 : タイムスタンプは無効です。
- 1 : タイムスタンプイネーブル

ビット 10 **WUTE** : ウェイクアップタイマイネーブル

- 0 : ウェイクアップタイマは無効です。
- 1 : ウェイクアップタイマは有効です。

ビット 9 **ALRBE** : アラーム B イネーブル

- 0 : アラーム B は無効です。
- 1 : アラーム B は有効です。

ビット 8 **ALRAE** : アラーム A イネーブル

- 0 : アラーム A は無効です。
- 1 : アラーム A は有効です。

ビット 7 **DCE** : 簡易デジタル較正イネーブル

- 0 : デジタル較正は無効です。
 - 1 : デジタル較正は有効です。
- PREDIV_A は 6 以上である必要があります。

ビット 6 **FMT** : 時間形式

- 0 : 24 時間/日形式
- 1 : AM/PM 時間形式

ビット 5 **BYP SHAD** : シャドウレジスタをバイパスします。

- 0 : カレンダ値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、シャドウレジスタから取得され、これらは 2 RTCCLK サイクルごとに 1 回更新されます。
- 1 : カレンダ値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、カレンダカウンタから直接取得されます。

注 : APB1 クロックの周波数が RTCCLK の 7 倍未満である場合、BYP SHAD は“1”にセットする必要があります。

ビット 4 **REFCKON** : リファレンスクロック検出イネーブル (50 または 60 Hz)

0 : リファレンスクロック検出は無効です。

1 : リファレンスクロック検出は有効です。

注 : **PREDIV_S** は **0x00FF** である必要があります。

ビット 3 **TSEDGE** : タイムスタンプイベントアクティブエッジ

0 : **TIMESTAMP** の立ち上りエッジによってタイムスタンプイベントを生成します。

1 : **TIMESTAMP** の立ち下りエッジによってタイムスタンプイベントを生成します。

不要な **TSF** 設定を回避するため、**TSEDGE** が変化した場合には **TSE** をリセットする必要があります。

ビット 2:0 **WUCKSEL[2:0]** : ウェイクアップクロック選択

000 : **RTC/16** クロックが選択されます。

001 : **RTC/8** クロックが選択されます。

010 : **RTC/4** クロックが選択されます。

011 : **RTC/2** クロックが選択されます。

10x : **ck_spre** (通常は 1 Hz) クロックが選択されます。

11x : **ck_spre** (通常は 1 Hz) クロックが選択され、 2^{16} が **WUT** カウンタ値に加算されます (下記注を参照)。

注 : **WUT** = ウェイクアップユニットカウンタ値 $WUT = (0x0000 \sim 0xFFFF) + 0x10000$ (**WUCKSEL[2:1]** = 11 の場合追加されます。)

初期化モード (**RTC_ISR/INITF** = 1) の場合のみ、このレジスタのビット 7、6、4 が書き込めます。

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、430 ページの [RTC レジスタ書き込み保護](#) を参照してください。

17.6.4 RTC 初期化とステータスレジスタ (RTC_ISR)

アドレスオフセット : 0x0C

バックアップドメインリセット値 : 0x0000 0007

システムリセット値 : 0 にクリアされる INIT、INITF、RSF 以外は影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TAMP 1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUT WF	ALRB WF	ALRA WF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

ビット 31:17 予約済み

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。[セクション : 動作中の再較正](#)を参照してください。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **TAMP1F** : タンパ検出フラグ

このフラグは、タンパ検出イベントが検出されたときに、ハードウェアによってセットされます。ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 12 **TSOVF** : タイムスタンプオーバーフローフラグ

このフラグは、TSF が既にセットされている間にタイムスタンプイベントが発生したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。さもないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 11 **TSF** : タイムスタンプフラグ

このフラグは、タイムスタンプイベントが発生したときに、ハードウェアによってセットされます。このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 10 **WUTF** : ウェイクアップタイマフラグ

このフラグは、ウェイクアップ自動再ロードカウンタが 0 に到達したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。
このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

ビット 9 **ALRBF** : アラーム B フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム B レジスタ (RTC_ALRMBR) と一致したときにハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 8 **ALRAF** : アラーム A フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム A レジスタ (RTC_ALRMAR) と一致したときにハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 7 INIT : 初期化モード

0 : フリーランニングモード

1 : 時刻と日付レジスタ (RTC_TR と RTC_DR)、およびプリスケアラレジスタ (RTC_PRER) のプログラムに使用する初期化モードです。INIT がリセットされると、カウンタは停止し、新しい値からカウントし始めます。

ビット 6 INITF : 初期化フラグ

このビットが 1 にセットされると、RTC は初期化状態となり、時刻、日付およびプリスケアラレジスタが更新できます。

0 : カレンダーレジスタを更新できません。

1 : カレンダーレジスタを更新できます。

ビット 5 RSF : レジスタ同期フラグ

このビットは、カレンダーレジスタがシャドウレジスタ (RTC_SSRx、RTC_TRx および RTC_DRx) にコピーされるたびにハードウェアによってセットされます。このビットは、シフト操作が保留中 (SHPF = 1)、またはバイパスシャドウレジスタモード中 (BYPHAD = 1) に、初期化モードでハードウェアによってクリアされます。このビットは、ソフトウェアでクリアすることもできます。

0 : カレンダーシャドウレジスタはまだ同期していません。

1 : カレンダーシャドウレジスタは同期しています。

ビット 4 INITS : 初期化ステータスフラグ

このビットは、カレンダーの年の項目が 0 ではないとき (バックアップドメインリセット値状態) にハードウェアによってセットされます。

0 : カレンダーは初期化されていません。

1 : カレンダーは初期化されています。

ビット 3 SHPF : シフト操作保留

0 : 保留中のシフト操作はありません。

1 : 保留中のシフト操作があります。

このフラグは、RTC_SHIFTR への書き込みによってシフト操作が開始された直後に、ハードウェアによってセットされます。該当するシフト操作が実行されると、ハードウェアによってクリアされます。SHPF への書き込みは無効です。

ビット 2 WUTF : ウェイクアップタイマ書き込みフラグ

このビットは、RTC_CR で WUTE ビットが 0 にセットされた後、ウェイクアップタイマ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。

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。

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、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.5 RTC プリスケアラレジスタ (RTC_PRER)

アドレスオフセット：0x10

バックアップドメインリセット値：0x007F 00FF

システムリセット：影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									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 31:24 予約済み

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 22:16 **PREDIV_A[6:0]**：非同期プリスケアラ分周比
非同期分周比です。
 $ck_apre \text{ 周波数} = RTCCLK \text{ 周波数} / (PREDIV_A + 1)$

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **PREDIV_S[14:0]**：同期プリスケアラ分周比
同期分周比です。
 $ck_spre \text{ 周波数} = ck_apre \text{ 周波数} / (PREDIV_S + 1)$

注： このレジスタは、必ず初期化モードで書き込む必要があります。初期化は、2 回の書き込みアクセスに分けて行う必要があります。430 ページの[カレンダーの初期化と設定](#)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、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.6 RTC ウェイクアップタイマレジスタ (RTC_WUTR)

アドレスオフセット : 0x14

バックアップドメインリセット値 : 0x0000 FFFF

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
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

ビット 15:0 **WUT[15:0]** : ウェイクアップ自動再ロード値ビット

ウェイクアップタイマが有効 (WUTE が 1 にセット) なとき、ck_wut の (WUT[15:0] + 1) サイクル毎に WUTF フラグがセットされます。ck_wut の周期は、RTC_CR レジスタの WUCKSEL[2:0] ビットで選択します。

WUCKSEL[2] = 1 のとき、ウェイクアップタイマは 17 ビットとなり、WUCKSEL[1] が事実上タイマに再ロードされる最上位ビットである WUT[16] となります。

注 : **WUTF の最初のアサートは、WUTE がセットされてから ck_wut の (WUT+1) サイクル後に発生します。WUCKSEL[2:0] = 011 (RTCCLK/2) のときに WUT[15:0] を 0x0000 にセットすることはできません。**

注 : このレジスタは、RTC_ISR の WUTWF が 1 にセットされているときにのみ書き込めます。
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、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.7 RTC 較正レジスタ (RTC_CALIBR)

アドレスオフセット : 0x18

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

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

ビット 31:8 予約済み

ビット 7 **DCS** : デジタル較正符号

0 : 正の較正 : カレンダ更新頻度が増加します。

1 : 負の較正 : カレンダ更新頻度が減少します。

ビット 6:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DC[4:0]** : デジタル較正

DCS = 0 (正較正)

00000 : + 0 ppm

00001 : + 4 ppm (概数値)

00010 : + 8 ppm (概数値)

..

11111 : + 126 ppm (概数値)

DCS = 1 (負較正)

00000 : - 0 ppm

00001 : - 2 ppm (概数値)

00010 : - 4 ppm (概数値)

..

11111 : - 63 ppm (概数値)

正確なステップ値については、435 ページの[RTCCLK=32.768kHz](#) かつ [PREDIV_A+1=128](#) の場合を参照してください。

注 : このレジスタは、初期化モード (RTC_ISR/INITF =“1”) の場合のみ書き込みます。

 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、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.8 RTC アラーム A レジスタ (RTC_ALRMAR)

アドレスオフセット : 0x1C

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
r/w	r/w	r/w	r/w	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

注 : このレジスタは、RTC_ISR の **ALRAWF** が 1 にセットされた場合、または初期化モードの場合にのみ書き込みます。

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、430 ページの [RTC レジスタ書き込み保護](#) を参照してください。

17.6.9 RTC アラーム B レジスタ (RTC_ALRMBR)

アドレスオフセット : 0x20

バックアップドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
r/w	r/w	r/w	r/w	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

注 : このレジスタは、RTC_ISR の ALRBWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込みめます。

 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、430 ページの[RTC レジスタ書き込み保護](#)を参照してください。

17.6.10 RTC 書き込み保護レジスタ (RTC_WPR)

アドレスオフセット : 0x24

バックアップドメインリセット値 : 0x0000 0000

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

17.6.11 RTC サブセカンドレジスタ（RTC_SSR）

アドレスオフセット：0x28

バックアップドメインリセット値：0x0000 0000

システムリセット：BYP SHAD = 0 の場合、0x0000 0000 です。BYP SHAD = 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
予約済み															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済み

ビット 15:0 **SS** : サブセカンド値

SS[15:0]は、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。



17.6.12 RTC シフト制御レジスタ (RTC_SHIFTR)

アドレスオフセット : 0x2C
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	予約済み														
w	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.	SUBFS[14:0]														
r	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

ビット 14:0 **SUBFS** : 秒の小数部差し引き
このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。
SUBFS に書き込まれた値は、同期プリスケアラのカウンタに加算されます。このカウンタはカウントダウンしていくので、この操作によって、次の式で求める値が効果的にクロックから差し引き(遅延)されます。
$$\text{遅れ (秒)} = \text{SUBFS} / (\text{PREDIV_S} + 1)$$

ADD1S 機能が SUBFS と共に用いられた場合、秒の小数部を効果的にクロックに加算する (クロックを進める) ことができ、実際のクロックの進みは次の式のとおりとなります。
$$\text{進み(秒)} = (1 - (\text{SUBFS} / (\text{PREDIV_S} + 1)))$$

注 : SUBFS に書き込むことにより RSF はクリアされます。その後、ソフトウェアが RSF = 1 まで待つことにより、シャドウレジスタがシフトされた時刻で更新されていることが確実にあります。
[セクション 17.3.8 : RTC の同期](#)を参照してください。

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、[430 ページのRTC レジスタ書き込み保護](#)を参照してください。

17.6.13 RTC タイムスタンプ時刻レジスタ (RTC_TSTR)

アドレスオフセット : 0x30
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み									PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **PM** : AM/PM 表記
0 : AM または 24 時間形式
1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

注 : このレジスタの内容は、RTC_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

17.6.14 RTC タイムスタンプ日付レジスタ (RTC_TSDR)

アドレスオフセット : 0x34
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:13 **WDU[1:0]** : 曜日

ビット 12 **MT** : BCD 形式での月の十の位

- ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位
- ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

注 : このレジスタの内容は、RTC_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

17.6.15 RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)

- アドレスオフセット : 0x38
- バックアップドメインリセット値 : 0x0000 0000
- システムリセット : 影響なし

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

- ビット 31:16 予約済み
- ビット 15:0 **SS** : サブセカンド値
- SS[15:0]は、タイムスタンプイベントが発生したときの同期プリスケアラのカウンタの値です。

注 : このレジスタの内容は、RTC_ISR/TSF がセットされている場合にのみ有効です。また、RTC_ISR/TSF ビットがリセットされるとクリアされます。

17.6.16 RTC 校正レジスタ (RTC_CALR)

アドレスオフセット : 0x3C
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	予約済み				CALM[8:0]								
rW	rW	rW	r	r	r	r	rW	rW	rW	rW	rW	rW	rW	rW	rW

- ビット 31:16 予約済み
- 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 2¹¹ 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 17.3.11 : RTC の高精度デジタル校正](#)を参照してください。
- ビット 14 CALW8 : 8 秒校正サイクル周期の使用
CALW8 が“1”にセットされると、8 秒校正サイクル周期が選択されます。
CALW8 = 1 の場合、CALM[1:0] は“00”に固定されます。
[セクション 17.3.11 : RTC の高精度デジタル校正](#)を参照してください。
- ビット 13 CALW16 : 16 秒校正サイクル周期の使用
CALW16 が“1”にセットされると、16 秒校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 17.3.11 : RTC の高精度デジタル校正](#)を参照してください。
- ビット 12:9 予約済み
- ビット 8:0 CALM[8:0] : 校正マイナス
RTCCLK 2²⁰ 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 17.3.11 : 435 ページのRTC の高精度デジタル校正](#)参照してください。

注 : このレジスタは書き込み保護されています。書き込みアクセスの手順は、[430 ページのRTC レジスタ書き込み保護](#)を参照してください。

17.6.17 RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)

アドレスオフセット : 0x40
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み													ALARMOUT TYPE	TSIN SEL	TAMP1 INSEL
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP- PUDIS	TAMP-PRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPTS	予約済み				TAMPIE	TAMP1T RG	TAMP1 E
rw	rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw

- ビット 31:19 予約済み。常に 0 が読み出されます。
- ビット 18 **ALARMOUTTYPE** : RTC_ALARM 出力形式
0 : RTC_ALARM はオープンドレイン出力になります。
1 : RTC_ALARM はプッシュプル出力になります。
- ビット 17 **TSINSEL** : TIMESTAMP マッピング
0 : RTC_AF1 が TIMESTAMP として使用されます。
1 : 予約済み
- ビット 16 **TAMP1INSEL** : TAMPER1 マッピング
0 : RTC_AF1 が TAMPER1 として使用されます。
1 : 予約済み
- ビット 15 **TAMPPUDIS** : TAMPER ブルアップ無効
このビットにより、毎回のサンプリング前に各タンパピンをプリチャージするかどうか決定します。
0 : サンプリング前にタンパピンをプリチャージします (内部ブルアップを有効化)。
1 : タンパピンのプリチャージを無効化します。
注 :
- ビット 14:13 **TAMPPRCH[1:0]** : タンパプリチャージ持続時間
これらのビットにより、各サンプリングの前にブルアップを有効化している時間を決定します。
TAMPPRCH は、各タンパ入力に対して有効です。
0x0 : 1 RTCCLK サイクル
0x1 : 2 RTCCLK サイクル
0x2 : 4 RTCCLK サイクル
0x3 : 8 RTCCLK サイクル
- ビット 12:11 **TAMPFLT[1:0]** : タンパフィルタカウント
これらのビットにより、タンパイベントをアクティブにするのに必要な指定のレベル (TAMP*TRG) での連続サンプリングの数を決定します。TAMPFLT は、各タンパ入力に対して有効です。
0x0 : タンパ入力 that アクティブレベル (タンパ入力における内部ブルアップ無し) に変化するときのエッジで、タンパがアクティブになります。
0x1 : アクティブレベルでの連続した 2 回のサンプリングの後、タンパがアクティブになります。
0x2 : アクティブレベルでの連続した 4 回のサンプリングの後、タンパがアクティブになります。
0x3 : アクティブレベルでの連続した 8 回のサンプリングの後、タンパがアクティブになります。

ビット 10:8 **TAMPFREQ[2:0]** : タンパサンプリング周波数

これらのビットにより、各タンパ入力がサンプリングされる周波数を決定します。

0x0 : RTCCLK / 32768 (RTCCLK = 32768 Hz の場合 1 Hz)

0x1 : RTCCLK / 16384 (RTCCLK = 32768 Hz の場合 2 Hz)

0x2 : RTCCLK / 8192 (RTCCLK = 32768 Hz の場合 4 Hz)

0x3 : RTCCLK / 4096 (RTCCLK = 32768 Hz の場合 8 Hz)

0x4 : RTCCLK / 2048 (RTCCLK = 32768 Hz の場合 16 Hz)

0x5 : RTCCLK / 1024 (RTCCLK = 32768 Hz の場合 32 Hz)

0x6 : RTCCLK / 512 (RTCCLK = 32768 Hz の場合 64 Hz)

0x7 : RTCCLK / 256 (RTCCLK = 32768 Hz の場合 128 Hz)

ビット 7 **TAMPTS** : タンパ検出イベント時のタイムスタンプの有効化

0 : タンパ検出イベントがあっても、タイムスタンプは保存されません。

1 : タンパ検出イベント時、タイムスタンプが保存されます。

RTC_CR レジスタで TSE = 0 であっても TAMPTS は有効です。

ビット 6:3 予約済み。常に 0 が読み出されます。

ビット 2 **TAMPIE** : タンパ割込みイネーブル

0 : タンパ割込みは無効です。

1 : タンパ割込みは有効です。

ビット 1 **TAMP1TRG** : タンパ 1 のアクティブレベル

TAMPFLT != 00 の場合

0 : TAMPER1 がローのままのとき、タンパ検出イベントがトリガされます。

1 : TAMPER1 がハイのままのとき、タンパ検出イベントがトリガされます。

TAMPFLT = 00 の場合 :

0 : TAMPER1 の立ち上りエッジでタンパ検出イベントがトリガされます。

1 : TAMPER1 の立ち下りエッジでタンパ検出イベントがトリガされます。

注意 : **TAMPFLT = 0 の場合、TAMP1E ビットは、TAMP1TRG が変更されたらリセットし、TAMP1F の不要なセットを回避する必要があります。**

ビット 0 **TAMP1E** : タンパ 1 検出イネーブル

0 : タンパ 1 検出は無効です。

1 : タンパ 1 検出は有効です。

17.6.18 RTC アラーム A サブセカンドレジスタ (RTC_ALRMASSR)

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
予約済み				MASKSS[3:0]				予約済み							
r	r	r	r	rw	rw	rw	rw	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	SS[14:0]														
r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

ビット 31:28 予約済み

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

ビット 14:0 **SS[14:0]** : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。

注 : このレジスタは、RTC_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込めます。

 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、[430 ページの RTC レジスタ書き込み保護](#)を参照してください。

17.6.19 RTC アラーム B サブセカンドレジスタ（RTC_ALRMBSSR）

アドレスオフセット：0x48
バックアップドメインリセット値：0x0000 0000
システムリセット：影響なし

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

ビット 31:28 予約済み

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

ビット 14:0 **SS[14:0]**：サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム B をアクティブ化するかどうかを決定します。0～MASKSS-1 のビットだけが比較されます。

注：このレジスタは、RTC_CR レジスタの ALRBIE がリセットされた場合、または初期化モードの場合にのみ書き込めます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、次を参照してください。[セクション：RTC レジスタ書き込み保護](#)

17.6.20 RTC バックアップレジスタ (RTC_BKPxR)

アドレスオフセット : 0x50 から 0x9C
バックアップドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

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

ビット 31:0 BKP[31:0]
アプリケーションはこれらのレジスタに対してデータの読み書きをすることができます。
これらのレジスタは、V_{DD} がオフになった場合、V_{BAT} によって電源が供給されるため、システムリセットによりリセットされず、デバイスが低電力モードで動作する場合、レジスタの内容は有効なまま保持されます。このレジスタは、TAMPx_F=1 である限り、タンパ検出イベントでリセットされます。

17.6.21 RTC レジスタマップ

表 68. 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	予約済み										PM	HT [1:0]		HU[3:0]			予約済み	MNT[2:0]			MNU[3:0]			予約済み	ST[2:0]		SU[3:0]						
	リセット値											0	0	0	0	0	0		0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0x04	RTC_DR	予約済み								YT[3:0]			YU[3:0]			WDU[2:0]		MT	MU[3:0]			予約済み	DT [1:0]		DU[3:0]									
	リセット値																	0	0	1	0		0	0	0	0	1	0	0	0	0	0	0	0
0x08	RTC_CR	予約済み										COE	OSEL [1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	DCE	FMT	BYPHAD	REFCKON	TSEDGE	WCKSEL [2:0]	
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	RTC_ISR	予約済み																		TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INTF	RSF	INTS	SHPF	WUTWF	ALRBWF	ALRAWF	
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	RTC_PRER	予約済み										PREDIV_A[6:0]				予約済み	PREDIV_S[14:0]																	
	リセット値											1	1	1	1		1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0x14	RTC_WUTR	予約済み																		WUT[15:0]														
	リセット値																			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x18	RTC_CALIBR	予約済み																								DCS	予約済み	DC[4:0]						
	リセット値																											0	0	0	0	0	0	0
0x1C	RTC_ALRMAR	MSK4	WDSEL	DT [1:0]		DU[3:0]			MSK3	PM	HT [1:0]		HU[3:0]			MSK2	MNT[2:0]		MNU[3:0]			MSK1	ST[2:0]		SU[3:0]									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

表 68. 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			
0x20	RTC_ALRMBR	MSK4	WDSEL	DT [1:0]	DU[3:0]				MSK3	PM	HT [1:0]	HU[3:0]				MSK2	MNT[2:0]				MNU[3:0]				MSK2	ST[2:0]				SU[3:0]						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x24	RTC_WPR	予約済み																								KEY[7:0]										
	リセット値																									0	0	0	0	0	0	0	0			
0x28	RTC_SSR	予約済み												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	0				
0x30	RTC_TSTR	予約済み								PM	HT[1:0]	HU[3:0]				予約済み	MNT[2:0]				MNU[3:0]				予約済み	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		
0x38	RTC_TSSSR	予約済み												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	0	0	0	0	0
0x3C	RTC_CALR	予約済み												CALP	CALW8	CALW16	予約済み	CALM[8:0]																		
	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	
0x40	RTC_TAFCR	予約済み										ALARMOUTTYPE	TSINSEL	TAMP1INSEL	TAMPPUDIS	TAMPPRCH[1:0]				TAMPFLT[1:0]				TAMPFREQ[2:0]				TAMPTS	予約済み	TAMPIE	TAMP1ETRQ	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	0		
0x44	RTC_ALRMASSR	予約済み				MASKSS[3:0]				予約済み								SS[14:0]																		
	リセット値					0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	RTC_ALRMBSSR	予約済み				MASKSS[3:0]				予約済み								SS[14:0]																		
	リセット値					0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50 ~0x9C	RTC_BKP0R	BKP[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	~RTC_BKP19R	BKP[31:0]																																		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、38 ページの表 1 を参照してください。

注意 : 表 68 では、リセット値とはバックアップドメインリセット後の値です。レジスタの大部分は、システムリセットによる影響を受けません。詳細については、セクション 17.3.7 : RTC のリセット を参照してください。



18 I²C (Inter-integrated circuit) インタフェース

18.1 I²C の概要

I²C (Inter-integrated circuit) バスインタフェースは、マイクロコントローラとシリアル I²C バス間のインタフェースとして機能します。マルチマスタ機能を備え、すべての I²C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準 (最大 100 kHz) モードと高速 (最大 400 kHz) モードをサポートしています。I²C バス周波数は、最大 1 MHz まで上げることができます。完全なソリューションについての詳細は、ST セールス担当までお問い合わせください。

また、SMBus2.0 と互換性を持っています。このインタフェースは、CRC の生成と検証、SMBus (System Management Bus)、PMBus (PowerManagement Bus) など、さまざまな目的に使用できます。

18.2 I²C の主な特長

- パラレルバス/I²C プロトコル間のコンバータ
- マルチマスタ機能: 同じインタフェースがマスタまたはスレーブとして機能
- I²C マスタの機能:
 - クロック生成
 - START 信号、STOP 信号の生成
- I²C スレーブの機能:
 - プログラム可能な I²C アドレス検出
 - 2 つのスレーブアドレスを確認応答するデュアルアドレス機能
 - STOP ビット検出
- 7 ビット/10 ビットアドレッシングおよび同報 (General Call) の生成と検出
- 複数の通信速度をサポート:
 - 標準モード (最大 100 kHz)
 - 高速モード (最大 400 kHz)
 - I²C バス周波数は、最大 1 MHz まで上げることができます。完全なソリューションについての詳細は、ST セールス担当までお問い合わせください。
- アナログノイズフィルタ
- プログラム可能なデジタルノイズフィルタ
- ステータスフラグ:
 - トランスマッタ/レシーバモードフラグ
 - 最終バイト送信フラグ
 - I²C ビジーフラグ
- エラーフラグ:
 - マスタモードのアービトレーションロスト条件
 - アドレス/データ送信後の確認応答障害
 - START/STOP コンディションの誤配置検出
 - クロックストレッチが無効な場合のオーバーラン/アンダーラン
- 2 つの割り込みベクタ:
 - 正常なアドレス/データ通信用の割り込み

- エラー条件用の割込み
- クロックストレッチオプション
- DMA 機能付きの 1 バイトバッファ
- 設定可能な PEC (Packet Error Checking) の生成/検証 ;
 - Tx モードでは PEC 値を最終バイトとして送信可能
 - 最後の受信バイトの PEC エラーチェック
- SMBus 2.0 互換性 :
 - クロックロータイムアウト遅延 : 25 ms
 - マスタのクロックロー累積延長時間 : 10 ms
 - スレーブのクロックロー累積延長時間 : 25 ms
 - ハードウェアによる PEC 生成/検証 (ACK 制御付き)
 - アドレス解決プロトコル (ARP) サポート
- PMBus 互換性

注 : 製品によっては、上記の機能の一部を使用できないことがあります。搭載されている I²C インタフェースによってサポートされる特定の機能については、製品のデータシートを参照してください。

18.3 I²C の機能説明

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン (SDA) とクロックピン (SCL) によって I²C バスに接続されます。標準（最大 100 kHz）または高速（最大 400 kHz）の I²C バスと接続できます。

18.3.1 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます :

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成した後はスレーブからマスタに、アービトレーションロスまたは STOP 生成が生じた場合はマスタからスレーブに、それぞれ自動的に切り替わることで、マルチマスタ機能を可能にします。

通信の流れ

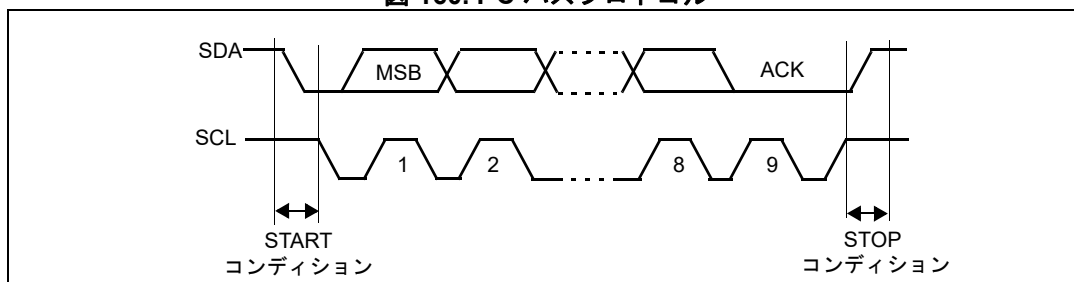
マスタモードでは、I²C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで始まり、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス（7 または 10 ビット）と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効/無効にできます。

データとアドレスは、MSB ファーストの 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト（7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト）にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります。図 160 を参照してください。

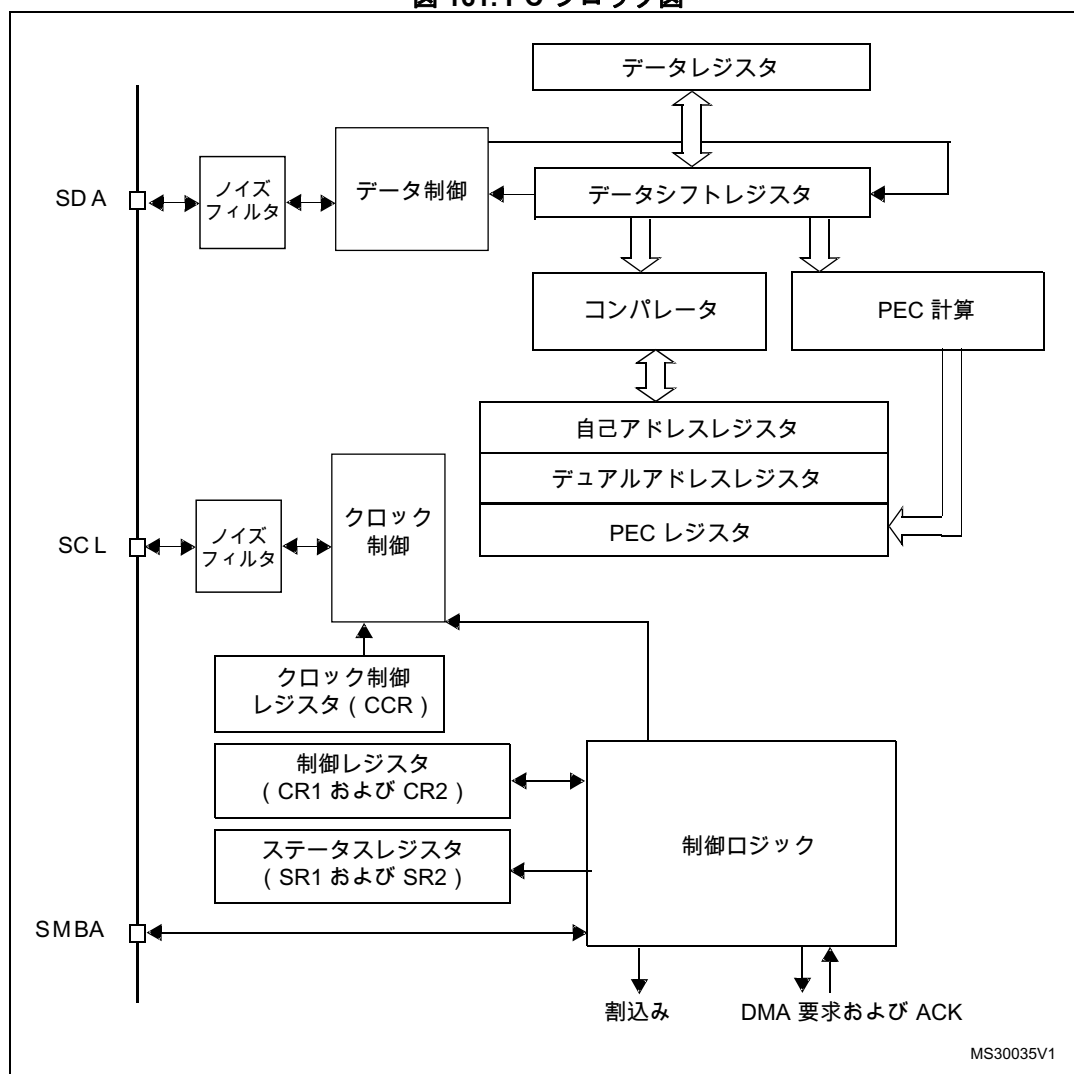
図 160. I²C バスプロトコル



確認応答は、ソフトウェアによって有効／無効にできます。I²C インタフェースアドレス（デュアルアドレッシングの 7 ビット／10 ビットや同報アドレス）は、ソフトウェアによって選択できます。

I²C インタフェースのブロック図を 図 161 に示します。

図 161. I²C ブロック図



1. SMBA は、SMBus モードでのオプション信号です。SMBus が無効な場合、この信号は適用されません。

18.3.2 I²C スレーブモード

デフォルトでは、I²C インタフェースはスレーブモードで動作します。デフォルトのスレーブモードからマスタモードに切り替えるには、START コンディションの生成が必要です。

正しいタイミングを生成するには、ペリフェラル入力クロックを I2C_CR2 レジスタにプログラミングする必要があります。ペリフェラル入力クロックは、少なくとも次の周波数以上でなければなりません。

- 標準モード : 2 MHz
- 高速モード : 4 MHz

START コンディションが検出されると、アドレスは、SDA ラインから受信されて、シフトレジスタに送信されます。その後、このアドレスはインタフェースのアドレス (OAR1) および OAR2 (ENDUAL=1 の場合) または同報アドレス (ENGCG=1 の場合) と比較されます。

注 : 10 ビットアドレスモードでは、この比較にはヘッダシーケンス (11110xx0) が含まれます。ここで、xx はアドレスの上位 2 ビットを示します。

ヘッダまたはアドレスの不一致：インタフェースはこれを見逃し、別の START コンディションを待ちます。

ヘッダー一致 (10 ビットモードのみ)：インタフェースは、ACK ビットがセットされている場合は確認応答パルスを生じ、8 ビットのスレーブアドレスを待ちます。

アドレス一致：インタフェースは以下のイベントを順に生じます。

- ACK ビットがセットされている場合、確認応答パルスを生じます。
- ADDR ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割込みが生じます。
- ENDUAL=1 の場合、ソフトウェアは、DUALF ビットを読み出して、どのスレーブアドレスが確認応答されたかをチェックする必要があります。

10 ビットモードでは、アドレスシーケンスの受信後、スレーブは常にレシーバモードになります。スレーブがトランスマッタモードに入るのは、反復 START コンディションの後、一致するアドレスビットを持ち、最下位ビットがセットされた (11110xx1) ヘッダシーケンスを受信したときです。

TRA ビットは、スレーブがレシーバモードかトランスマッタモードかを示します。

スレーブトランスマッタ

アドレス受信に続いて、ADDR をクリアした後、スレーブは、内部シフトレジスタを介して、DR レジスタから SDA ラインにバイトを送信します。

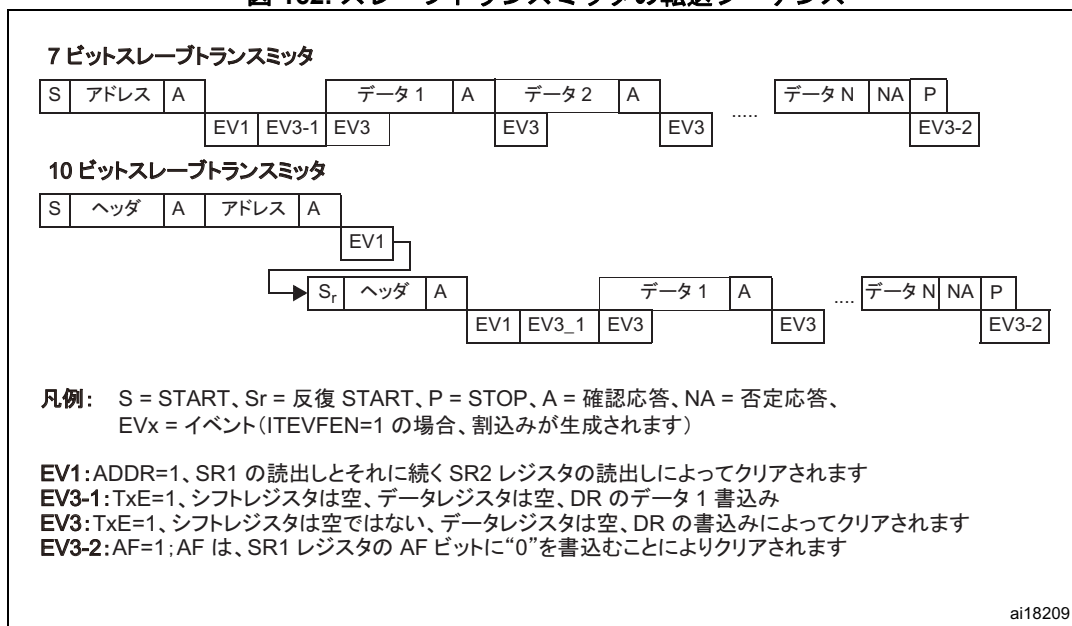
スレーブは、ADDR がクリアされ、DR に送信データが書き込まれるまで、SCL のローレベルを伸長します (図 162 転送シーケンスの EV1 EV3 を参照)。

確認応答パルスが受信されると：

- TxE ビットがハードウェアによってセットされ、ITEVFEN および ITBUFEN ビットがセットされている場合は割込みが生じます。

TxE がセットされ、次のデータ送信の終わりまでに I2C_DR レジスタに一部のデータが書き込まれていない場合、BTF ビットがセットされ、インタフェースは、I2C_SR1 の読出しと、それに続く I2C_DR レジスタへの書き込みによって BTF がクリアされるまで、SCL のローレベルを伸長して待ちます。

図 162. スレーブトランスミッタの転送シーケンス



1. EV1 イベントおよび EV3-1 イベントは、対応するソフトウェアシーケンスの終わりまで SCL のローレベルを伸長します。
2. 次のバイト送信の終わりまでにソフトウェアシーケンスが完了しない場合、EV3 イベントは SCL のローレベルを伸長します。

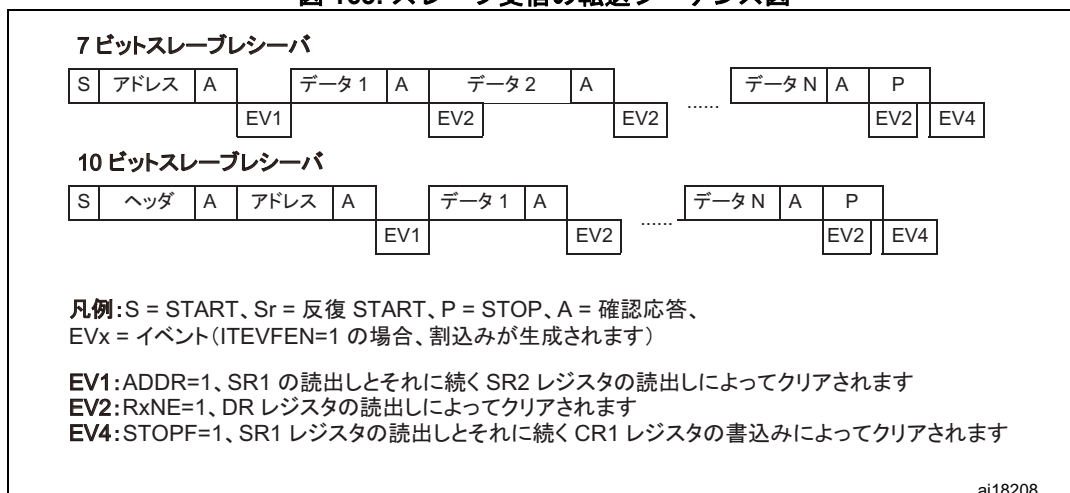
スレーブレシーバ

アドレス受信に続いて、ADDR がクリアされた後、スレーブは、SDA ラインからのバイトを内部シフトレジスタを介して DR レジスタに受信します。各バイトの後、インタフェースは以下のイベントを順に生成します。

- ACK ビットがセットされている場合、確認応答パルスを生成します。
- RxNE ビットがハードウェアによってセットされ、ITEVFEN および ITBUFEN ビットがセットされている場合は、割込みが生成されます。

RxNE がセットされ、次のデータ受信の終わりまでに DR レジスタ内のデータが読み出されなかった場合、BTF ビットがセットされます。さらにインタフェースは、I2C_DR の読出しによって BTF がクリアされるまで、SCL のローレベルを伸長して待ちます (図 163 転送シーケンスを参照)。

図 163. スレーブ受信の転送シーケンス図



- EV1 イベントは、対応するソフトウェアシーケンスの終わりまで SCL のローレベルを伸長します。
- 次のバイト受信の終わりにソフトウェアシーケンスが完了しない場合、EV2 イベントは SCL のローレベルを延長します。
- SR1 レジスタの状態をチェックした後、ユーザは、セットされているフラグごとに、完全なクリアシーケンスを実行してください。
すなわち、ADDR フラグと STOPF フラグに対し、I²C 割込みルーチンの中に次のシーケンスを記述する必要があります。

```

READ SR1
if (ADDR == 1) {READ SR1; READ SR2}
if (STOPF == 1) {READ SR1; WRITE CR1}

```

これは、ADDR フラグと STOPF フラグがセットされていることが分かった場合、両方とも確実にクリアするための処理です。

スレーブ通信の終了

最後のデータバイトが転送された後、マスタは STOP コンディションを生成します。インタフェースは、STOP コンディションを検出して、次の動作を行います。

- STOPF ビットをセットし、ITEVFEN ビットがセットされている場合は割込みを生成します。

STOPF は、SR1 レジスタの読出しと、それに続く CR1 レジスタへの書き込みによってクリアされます (図 163 : スレーブ受信の転送シーケンス図 EV4 を参照)。

18.3.3 I²C マスタモード

マスタモードでは、I²C インタフェースは、データ転送を開始し、クロック信号を生成します。データ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START ビットによってバス上に START コンディションが生成されると、マスタモードが選択されます。

次に、マスタモードに必要なシーケンスを示します。

- 正しいタイミングを生成するため、I2C_CR2 レジスタにペリフェラル入カクロックを設定します。
- クロック制御レジスタを設定します。
- 立ち上がり時間レジスタを設定します。
- I2C_CR1 レジスタを設定して、ペリフェラルを有効にします。
- I2C_CR1 レジスタの START ビットをセットして、START コンディションを生成します。

ペリフェラル入カクロックは、少なくとも次の周波数以上でなければなりません。

- 標準モード : 2 MHz
- 高速モード : 4 MHz

SCL マスタクロック生成

CCR ビットは、立ち上がりエッジおよび立ち下がりエッジの生成（それぞれ）を始め、ハイおよびローレベルの SCL クロックの生成に使用します。スレーブは SCL ラインをストレッチする場合があります。そのため、ペリフェラルは立ち上がりエッジ生成後、TRISE ビットにプログラムされた時間の最後にバスからの SCL 入力をチェックします。

- SCL ラインがローの場合、スレーブがバスをストレッチしており、ハイレベルカウンタが SCL ラインがハイで検出されるまで停止することを示します。これにより、SCL クロックパラメータの最小ハイ周期を保証できます。
- SCL ラインがハイの場合、ハイレベルカウンタはカウントを継続します。

スレーブがクロックをストレッチしていなくても、ペリフェラルによる SCL 立ち上がりエッジ生成からペリフェラルによる SCL 立ち上がりエッジ検出までのフィードバックループには時間がかかります。このループバック時間は、SCL 立ち上がり時間（SCL VIH 入力検出に影響）、SCL 入力パスにあるノイズフィルタによる遅延、および APB クロックとの内部 SCL 入力同期による遅延に関連します。フィードバックループで使用される最大時間は、TRISE ビットでプログラムされているため、SCL 周波数は SCL 立ち上がり時間にかかわらず安定した状態になります。

START コンディション

BUSY ビットがクリアされているときに START ビットをセットすると、インタフェースは、START コンディションを生成し、マスタモードに切り替わります（MSL ビットをセット）。

注： マスタモードで START ビットをセットすると、インタフェースは、現在のバイト転送の最後に RESTART コンディションを生成します。

START コンディションが送信されると、以下のイベントが発生します。

- SB ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割込みが生成されます。

次に、マスタは、SR1 レジスタの読出しと、それに続く DR レジスタへのスレーブアドレスの書き込みを待ちます（図 164 と図 165 の転送シーケンスの EV5 を参照）。

スレーブアドレス送信

次に、スレーブアドレスが内部シフトレジスタを介して SDA ラインに送信されます。

- 10 ビットアドレスモードでは、ヘッダシーケンスの送信によって、次のイベントが発生します。
 - ADD10 ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割込みが生成されます。

次に、マスタは、SR1 レジスタの読出しと、それに続く DR レジスタへの 2 番目のアドレスバイトの書き込みを待ちます（図 164 と図 165 の転送シーケンスを参照）。

- ADDR ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割込みが生成されます。

次にマスタは SR1 レジスタの読出しと、それに続く SR2 レジスタの読出しを待ちます（図 164 と図 165 転送シーケンスを参照）。

- 7 ビットアドレスモードでは、1 個のアドレスバイトが送信されます。
 - ADDR ビットがハードウェアによってセットされ、ITEVFEN ビットがセットされている場合は割込みが生成されます。

次にマスタは SR1 レジスタの読出しと、それに続く SR2 レジスタの読出しを待ちます（図 164 と図 165 転送シーケンスを参照）。

マスタは、送信されたスレーブアドレスの LSB に応じて、トランスミッタモードに入るか、レシーバモードに入るかを決定できます。

- 7 ビットアドレスモードでは、
 - トランスミッタモードに入るには、マスタは、LSB をリセットしたスレーブアドレスを送信します。
 - レシーバモードに入るには、マスタは、LSB をセットしたスレーブアドレスを送信します。
- 10 ビットアドレスモードでは、
 - トランスミッタモードに入るには、マスタは、ヘッダ (11110xx0) を送信してから、スレーブアドレスを送信します (xx はアドレスの上位 2 ビットです)。
 - レシーバモードに入るには、マスタは、ヘッダ (11110xx0) を送信してから、スレーブアドレスを送信します。さらに、反復 START コンディションと、それに続いてヘッダ (11110xx1) を送信します (xx はアドレスの上位 2 ビットです)。

TRA ビットは、マスタがレシーバモードにあるか、トランスミッタモードにあるかを示します。

マスタトランスミッタ

アドレス送信に続いて、ADDR をクリアした後、マスタは、内部シフトレジスタを介して、DR レジスタから SDA ラインにバイトを送信します。

マスタは、最初のデータバイトが I2C_DR に書き込まれるまで待ちます (図 164 転送シーケンスの EV8_1 を参照)。

ACK (アクノリッジパルス) を受信すると、TxE ビットがハードウェアによってセットされ、ITEVFEN ビットおよび ITBUFEN ビットがセットされている場合は割込みが生成されます。

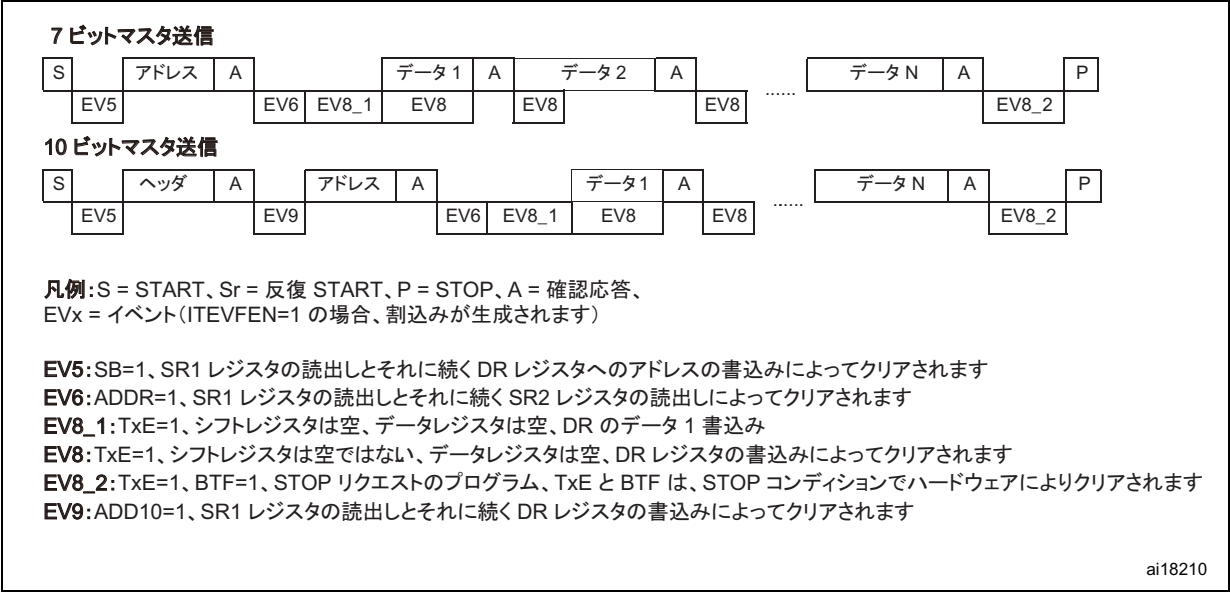
TxE がセットされていて、最後のデータ送信が終わるまでにデータバイトが DR レジスタに書き込まれなかった場合、BTF がセットされ、インタフェースは I2C_DR への書き込みによって BTF がクリアされるまで、SCL のローレベルを延長して待ちます。

通信の終了

最後のバイトが DR レジスタに書き込まれた後、STOP ビットがソフトウェアによってセットされて STOP コンディションを生成します (図 164 の転送シーケンスの EV8_2 を参照)。インタフェースは自動的にスレーブモードに戻ります (MSL ビットがクリアされます)。

注 : STOP コンディションは、TxE または BTF ビットがセットされたとき、EV8_2 イベント時にプログラミングしてください。

図 164. マスタ送信の転送シーケンス図



1.

EV5、EV6、EV9、EV8_1、および EV8_2 の各イベントは、対応するソフトウェアシーケンスの終わりまで、SCL のローレベルを延長します。
2.

次のバイト送信の終わりまでにシーケンスが完了しない場合、EV8 イベントは SCL のローレベルを延長します。

マスタレシーバ

アドレス送信に続いて ADDR がクリアされると、I²C インタフェースはマスタレシーバモードに入ります。このモードでは、インタフェースは、SDA ラインから内部シフトレジスタを介して DR レジスタにバイトを受信します。各バイトの後、インタフェースは以下のイベントを順に生成します。

1. ACK ビットがセットされている場合、確認応答パルスを生成します。
2. RxNE ビットがセットされ、ITEVFEN および ITBUFEN ビットがセットされている場合は割込みが生成されます（図 165 転送シーケンスの EV7 を参照）。

RxNE ビットがセットされ、最後のデータ受信の終わりまでに DR レジスタ内のデータが読み出されなかった場合、BTF ビットがハードウェアによってセットされ、インタフェースは、DR レジスタの読出しによって BTF がクリアされるまで、SCL のローレベルを延長して待ちます。

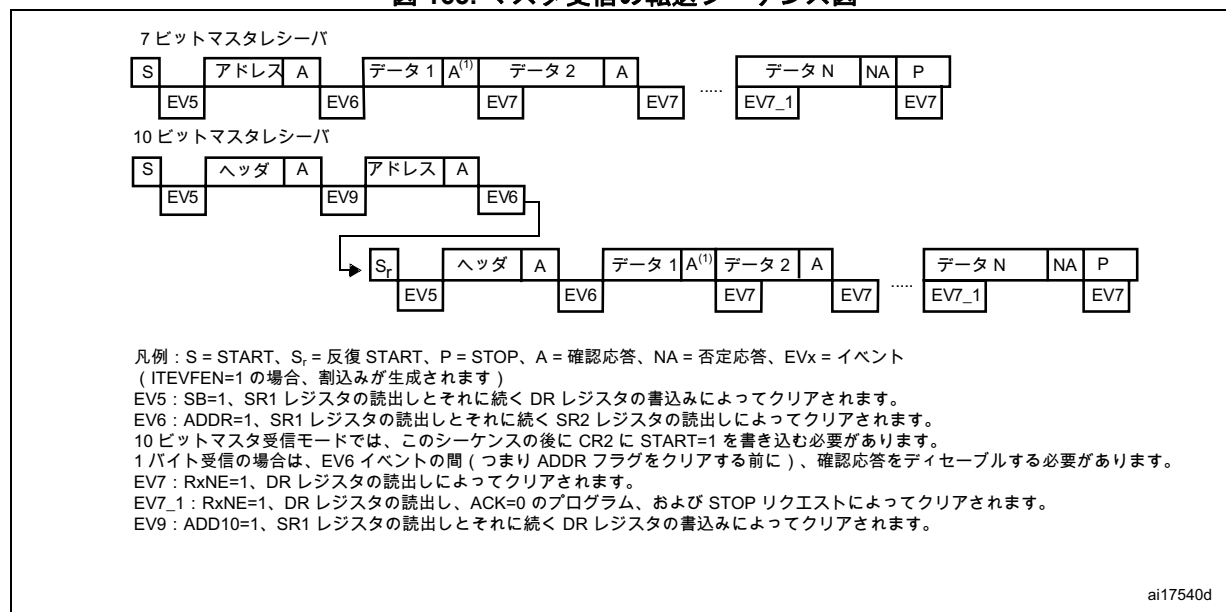
通信の終了

マスタは、スレーブから受信した最終バイトに対して NACK を送信します。この NACK を受信したスレーブは、SCL ラインと SDA ラインの制御を解除します。これによって、マスタは STOP／RESTART コンディションを送信できます。

1. 最後の受信データバイトの後で NACK パルスを生成するには、最後から 2 番目の RxNE イベントの後で最後から 2 番目のデータバイトの読出し直後に、ACK ビットをクリアする必要があります。
2. STOP／RESTART コンディションを生成するには、ソフトウェアは、最後から 2 番目の RxNE イベントの後で最後から 2 番目のデータバイトの読出し後に、STOP／START ビットをセットする必要があります。
3. 1 バイトを受信する必要がある場合は、EV6 の間 (ADDR フラグがクリアされる前) に ACK (アクノリッジ) が無効化され、EV6 の後に STOP コンディションが生成されます。

STOP コンディションの生成後、インタフェースは自動的にスレーブモードに戻ります (MSL ビットがクリアされます)。

図 165. マスタ受信の転送シーケンス図



- 1 バイトが受信された場合、無効となります。
- EV5、EV6、および EV9 イベントは、対応するソフトウェアシーケンスの終わりまで、SCL のローレベルを延長します。
- 次のバイト受信の終わりまでにソフトウェアシーケンスが完了しない場合、EV7 イベントは SCL のローレベルを延長します。
- EV7_1 ソフトウェアシーケンスは、現在のバイト転送の ACK パルスより前に完了する必要があります。

現在のバイト転送の ACK パルスより前に EV7-1 ソフトウェアシーケンスが完了しない場合、以下の手順を推奨します。

以下の手順に従う必要があります。

- 最後のデータ受信が終わる前に、ACK ビットが遅れずにローにセットされる
- 最後のデータ受信の後、補助データの受信なしに STOP ビットがハイにセットされる。

2 バイト受信の場合：

- ADDR = 1 になるまで待ちます (ADDR フラグがクリアされるまで SCL のローレベルを延長します)。
- ACK をローにセットし、POS をハイにセットします。
- ADDR フラグをクリアします。
- BTF = 1 になるまで待ちます (DR にデータ 1、シフトレジスタにデータ 2、データ 1 が読み出されるまで SCL のローレベルを伸長)。
- STOP をハイにセットします。
- データ 1 とデータ 2 を読み出します。

N > 2 バイト受信の場合 (N-2 データ受信から)

- BTF = 1 になるまで待ちます (DR にデータ N-2、シフトレジスタにデータ N-1、データ N-2 が読み出されるまで SCL のローレベルを伸長)。
- ACK をローにセットします。
- データ N-2 を読み出します。
- BTF = 1 になるまで待ちます (DR にデータ N-1、シフトレジスタにデータ N、データ N-1 が読み出されるまで SCL のローレベルを伸長)。
- STOP をハイにセットします。
- データ N-1 とデータ N を読み出します。

18.3.4 エラー条件

以下は、通信エラーを引き起こす可能性のあるエラー条件です。

バスエラー (BERR)

このエラーは、I²C インタフェースがアドレスやデータの転送中に外部の STOP または START コンディションを検出したときに発生します。このとき、以下のイベントが発生します。

- BERR ビットがセットされ、ITERREN ビットがセットされている場合は割込みが生成されます。
- スレーブモード：データは破棄され、バスはハードウェアによって解放されます。
 - START の誤配置である場合、スレーブはこれを RESTART とみなし、アドレスまたは STOP コンディションを待ちます。
 - STOP の誤配置である場合、スレーブは STOP コンディションの場合のように動作し、バスはハードウェアによって解放されます。
- マスタモード：バスは解放されず、現在の送信の状態は影響を受けません。現在の送信を中止するかどうかは、ソフトウェアに依存します。

確認応答障害 (AF)

このエラーは、インタフェースが NACK ビットを検出したときに発生します。このとき、

- AF ビットがセットされ、ITERREN ビットがセットされている場合は割込みが生成されます。
- NACK を受信したトランスミッタは、通信をリセットする必要があります。
 - スレーブの場合：バスはハードウェアによって解放されます。
 - マスタの場合：ソフトウェアが STOP または反復 START コンディションを生成する必要があります。

アービトレーション喪失 (ARLO)

このエラーは、I²C インタフェースがアービトレーションロスト状態を検出したときに発生します。このとき、

- ARLO ビットがハードウェアによってセットされます (ITERREN ビットがセットされている場合、割込みが生成されます)。
- I²C インタフェースは自動的にスレーブモードに戻ります (MSL ビットがクリアされます)。I²C がアービトレーションを失うと、同じ転送ではそのスレーブアドレスを確認応答できませんが、通信を許可されたマスタからの反復 START 後に確認応答できます。
- バスはハードウェアによって解放されます。

オーバーラン／アンダーランエラー (OVR)

スレーブモードでオーバーランエラーが発生するのは、クロックストレッチが無効であり、I²C インタフェースがデータを受信しているときです。インタフェースがすでにバイトを受信し (RxNE=1)、インタフェースが次のバイトを受信するまでに DR 内のデータがまだ読み出されていません。このとき、

- 最後に受信されたバイトは失われます。
- オーバーランエラーが発生した場合、ソフトウェアで RxNE ビットをクリアし、トランスミッターで最後の受信バイトを再送信してください。

スレーブモードでアンダーランエラーが発生するのは、クロックストレッチが無効であり、I²C インタフェースがデータを送信しているときです。次のバイト送信用のクロックが現れるまでに、次に送信すべきデータがまだ DR レジスタに格納されていません (TxNE=1)。このとき、

- DR レジスタ内の同じバイトが再送信されます。
- ユーザは、アンダーランエラー中にレシーバ側で受信されたデータが捨てられ、I²C バス規格で指定されたクロックロー時間内に次のバイトが書き込まれるようにしてください。

送信される最初のバイトに関しては、ADDR がクリアされてから最初の SCL 立ち上がりエッジまでに DR を書き込む必要があります。さもなければ、レシーバは最初のデータを破棄しなければなりません。

18.3.5 プログラム可能なノイズフィルタ

高速 モードでは、I²C 標準により、SDA ラインおよび SCL ライン上のスパイク幅を 50 ns まで短くすることが求められています。

SDA および SCL I/O には、アナログノイズフィルタが内蔵されています。このフィルタはデフォルトで有効化されており、I2C_FLTR レジスタの ANOFF ビットをセットして無効化することができます。

DNF[3:0] ビットをゼロ以外の値に設定することにより、デジタルノイズフィルタを有効化することができます。このフィルタは、SDA 入力および SCL 入力のスパイク幅を DNF[3:0] * T_{PCLK1} まで抑えることができます。

デジタルノイズフィルタを有効化すると、SDA ホールド時間が (DNF[3:0] + 1) * T_{PCLK1} だけ長くなります。

I²C-バス仕様バージョン 2.1 (Thd:dat) の最大ホールド時間を満たすためには、アナログフィルタを無効化し、表 69 に示す制約に従って DNF ビットをプログラムする必要があります。

注： DNF[3:0] は、必ず I²C が無効化されているとき (PE = 0) に設定する必要があります。アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。

表 69. Thd:dat (max) に準拠した最大 DNF[3:0] 値

PCLK1 周波数	最大 DNF 値	
	標準モード	高速 モード
2 ≤ F _{PCLK1} ≤ 5	2	0
5 < F _{PCLK1} ≤ 10	12	0
10 < F _{PCLK1} ≤ 20	15	1
20 < F _{PCLK1} ≤ 30	15	7
30 < F _{PCLK1} ≤ 40	15	13
40 < F _{PCLK1} ≤ 50	15	15

注： 各周波数範囲に対し、この制約は範囲の最小周波数である最悪ケースに基いて決められています。システムが最大ホールド時間を越えた対応が可能な場合は、DNF 値を大きくすることができます。

18.3.6 SDA/SCL ライン制御

- クロックストレッチが有効な場合：
 - トランスマッターモード：TxNE=1 および BTF=1 の場合、インタフェースは、送信の前にクロックラインをローレベルに保ち、マイクロコントローラがバイトをデータレジスタに書き込むのを待ちます（バッファとシフトレジスタは、いずれも空）。
 - レシーバモード：RxNE=1 および BTF=1 の場合、インタフェースは、受信の後にクロックラインをローレベルに保ち、マイクロコントローラがバイトをデータレジスタに読み出すのを待ちます（バッファとシフトレジスタは、いずれも満杯）。
- スレーブモードでクロック伸長が無効な場合：
 - オーバーランエラー：RxNE=1 で、次のバイトが受信されるまでに DR が読み出されていない場合。最後に受信されたバイトは失われます。
 - アンダーランエラー：TxNE=1 で、次のバイトを送信するまでに DR への書込みが行われていない場合。同じバイトが再送信されます。
 - 書込み衝突は管理されません。

18.3.7 SMBus

概要

システム管理バス（SMBus）は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I²C の動作原理に基づきます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。システムが SMBus を使用することで、個々の制御ラインをトグルすることなく、デバイスとの間でメッセージをやり取りできます。

システム管理バス仕様では、3 種類のデバイスを規定しています。スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ/スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

SMBus と I²C の類似点

- 2 線バスプロトコル（1 クロック、1 データ）+ SMBus アラートライン（オプション）
- マスタ/スレーブ通信（マスタがクロックを供給します）
- マルチマスタ機能
- I²C 7 ビットアドレスフォーマットに類似した SMBus データフォーマット（[図 160](#)）

SMBus と I²C の相違点

次の表に、SMBus と I²C の相違点を示します。

表 70. SMBus と I²C

SMBus	I ² C
最大速度 : 100 kHz	最大速度 : 400 kHz
最小クロック速度 : 10 kHz	最小クロック速度の規定なし
クロックロータイムアウト : 35 ms	タイムアウトなし
論理レベルは固定	ロジックレベルはV _{DD} に依存
種々のアドレスタイプ (予約、ダイナミック、など)	7 ビット、10 ビット、および同報のスレーブアドレスタイプ
種々のバスプロトコル (クイックコマンド、プロセスコール、など)	バスプロトコルなし

SMBus アプリケーションの用途

SMBus を使用すれば、デバイスは、製造者情報を提供し、モデル／部品番号をシステムに通知し、サスペンドイベントに備えてその状態を保存し、種々のエラーを報告し、制御パラメータを受け付け、そのステータスを返すことができます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。

デバイス識別

SMBus 上にスレーブとして存在するデバイスには、スレーブアドレスと呼ばれるユニークなアドレスがあります。予約済みのスレーブアドレスの一覧については、SMBus 仕様 ver.2.0 (<http://smbus.org/>) を参照してください。

バスプロトコル

SMBus 仕様では、最大 9 種のプロトコルをサポートします。これらのプロトコルと SMBus アドレスタイプの詳細については、SMBus 仕様 ver.2.0 (<http://smbus.org/>) を参照してください。これらのプロトコルは、ユーザのソフトウェアによって実装してください。

アドレス解決プロトコル (ARP)

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス解決プロトコル (ARP) には、以下の属性があります。

- アドレス割り当てには、標準の SMBus 物理層アービトレーション機構を使用します。
- 割り当てられたアドレスは、デバイスに電源が供給されている間は一定しています。デバイスの電源切断後もアドレスの保持が可能です。
- アドレス割り当ての後、新たな SMBus パケットのオーバーヘッドは生じません。(つまり、割り当てられたスレーブアドレスへのその後のアクセスでは、固定アドレスのデバイスへのアクセスと同じオーバーヘッドになります。)
- いずれの SMBus マスタも、バスの番号付けが可能です。

ユニークデバイス識別子 (UDID)

アドレス割り当てを目的として各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子 (UDID) を実装する必要があります。

128 ビット UDID と ARP の詳細については、SMBus 仕様 ver. 2.0 (<http://smbus.org/>) を参照してください。

SMBus アラートモード

SMBus アラートは、マスタ機能をピンに置き換えたいデバイス用の割込みラインを持つオプション信号です。SMBA は、SCL 信号や SDA 信号と同じく、ワイヤード AND 信号です。SMBA は、SMBus 同報アドレスとともに使用されます。SMBus で呼び出されるメッセージの長さは 2 バイトです。

スレーブ専用デバイスは、I2C_CR1 レジスタの ALERT ビットをセットすることによって、ホストに接続したいことを SMBA を通じて通知できます。ホストは、割込みを処理し、アラート応答アドレス (ARA と呼ばれ、値は 0001 100X) を通じて全 SMBA デバイスに同時にアクセスします。SMBA をローレベルに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。このステータスは、I2C_SR1 レジスタの SMBALERT ステータスフラグを使用して識別されます。ホストは、モディファイド受信バイト動作を行います。スレーブ送信デバイスによって提供された 7 ビットのデバイスアドレスは、バイトの上位 7 ビットに置かれます。8 番目のビットは 0 または 1 にできます。

複数のデバイスが SMBA をローレベルに引き下げている場合、スレーブアドレス転送時の標準的なアービトラーションによって、最高優先順位 (最小アドレス) のデバイスが通信権を獲得します。スレーブアドレスの確認応答後、デバイスは、その SMBA のプルダウンを解放する必要があります。メッセージ転送の完了時に、SMBA が依然としてローレベルであることにホストが気づいた場合、再び ARA の読出しが行われます。

SMBA 信号を実装していないホストは、周期的に ARA にアクセスすることができます。

SMBus アラートモードの詳細については、SMBus 仕様 ver. 2.0 (<http://smbus.org/>) を参照してください。

タイムアウトエラー

I²C と SMBus では、タイミング仕様に違いがあります。

SMBus では、クロックロータイムアウトとして 35 ms の TIMEOUT が定義されています。さらに SMBus では、スレーブデバイスのクロックロー累積延長時間として TLOW: SEXT を規定しています。また、マスタデバイスのクロックロー累積延長時間として TLOW: MEXT も規定しています。これらのタイムアウトの詳細については、SMBus 仕様 ver. 2.0 (<http://smbus.org/>) を参照してください。

I2C_SR1 レジスタのステータスフラグ Timeout や Tlow Error は、この機能のステータスを示します。

SMBus モードでのインタフェースの使い方

I²C モードから SMBus モードに切り替えるには、次のシーケンスを実行してください。

- I2C_CR1 レジスタの SMBus ビットをセットします。
- アプリケーションの必要に応じて、I2C_CR1 レジスタの SMBTYPE ビットと ENARP ビットを設定します。

デバイスをマスタとして設定したい場合、[セクション 18.3.3 : I²C マスタモード](#)に記載されている START コンディションの生成手順に従ってください。あるいは、[セクション 18.3.2 : I²C スレーブモード](#)のシーケンスに従ってください。

アプリケーションは、ソフトウェアによってさまざまな SMBus プロトコルを制御する必要があります。

- ENARP=1 および SMBTYPE=0 の場合：SMB デバイスのデフォルトアドレスが確認応答されます。
- ENARP=1 および SMBTYPE=1 の場合：SMB ホストヘッダが確認応答されます。
- SMBALERT=1 の場合：SMB アラート応答アドレスが確認応答されます。

18.3.8 DMA リクエスト

DMA リクエストは、データ転送用にのみ生成されます（有効な場合）。DMA リクエストが生成されるのは、送信中にデータレジスタが空になったとき、および受信中にデータレジスタが満杯になったときです。I²C データ転送の前に DMA を初期化して有効化する必要があります。ADDR イベントの前に I2C_CR2 レジスタで DMAEN ビットをセットする必要があります。マスタモードまたはスレーブモードにおいてクロックストレッチを有効化した場合、ADDR フラグをクリアする前に ADDR イベント中に DMAEN ビットをセットすることもできます。DMA リクエストは、現在のバイト転送の終わりまでに処理される必要があります。対応する DMA ストリームにプログラミングされたデータ転送回数に達したとき、DMA コントローラは、I²C インタフェースに転送終了（EOT）信号を送信し、転送完了割り込みを生成します（有効な場合）。

- マスタトランスミッタ：EOT 割り込み後の割り込みルーチンでは、DMA リクエストを無効にし、BTF イベントの発生を待ってから、STOP コンディションをプログラミングします。
- マスタレシーバ
 - 受信するバイト数が 2 以上であるとき、DMA コントローラは、最後から 2 番目のデータバイト（number_of_bytes - 1）に対応するハードウェア信号（EOT_1）を送信します。I2C_CR2 レジスタで LAST ビットがセットされた場合、I²C は、EOT_1 に続く次のバイトの後に、自動的に NACK を送信します。ユーザは、DMA 転送完了割り込みルーチンで STOP コンディションを生成できます（有効な場合）。
 - 1 バイトを受信する必要がある場合：EV6 イベント中に NACK をプログラムする必要があります。すなわち、ADDR フラグをクリアする前に ADDR=1 のときに ACK=0 をプログラムします。すると、ADDR フラグのクリア後、または DMA 転送完了割り込みルーチン内で、STOP コンディションをプログラムすることができます。

DMA を使用した送信

DMA モードの送信を有効にするには、I2C_CR2 レジスタの DMAEN ビットをセットします。データは、TxE ビットがセットされるたびに、DMA ペリフェラルを使用して設定されたメモリ領域（DMA 仕様を参照）から I2C_DR レジスタにロードされます。I²C 送信に DMA ストリーム x（ここで x はストリーム番号）を割り当てるには、次のシーケンスを実行します：

1. I2C_DR レジスタのアドレスを DMA_SxPAR レジスタにセットします。データは、毎回の TxE イベント後に、メモリからこのアドレスに移動されます。
2. DMA_SxMA0R レジスタに（およびダブルバッファモードの場合は DMA_SxMA1R レジスタ）メモリアドレスを設定します。データは、毎回の TxE イベント後に、このメモリから I2C_DR にロードされます。
3. 転送されるバイトの総数を DMA_SxNDTR レジスタに設定します。毎回の TxE イベント後に、この値はデクリメントされます。
4. DMA_SxCR レジスタの PL[0:1] ビットを使用して、DMA ストリームの優先順位を設定します。
5. DMA_SxCR レジスタの DIR ビットをセットし、アプリケーション条件に応じて 1/2 転送終了後または転送完了後に、割り込みを設定します。
6. DMA_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

DMA コントローラのレジスタにプログラミングされたデータ転送回数に達したとき、DMA コントローラは、I²C インタフェースに転送終了（EOT/EOT_1）信号を送信し、DMA は DMA チャネル割り込みベクタ上に割り込みを生成します（有効な場合）。

注： DMA が送信に使用される場合、I2C_CR2 レジスタの ITBUFEN ビットを有効にしないでください。

DMA を使用した受信

DMA モードの受信を有効にするには、I2C_CR2 レジスタの DMAEN ビットをセットします。データは、データバイトが受信されるたびに、DMA ペリフェラルを使用して設定されたメモリ領域（DMA 仕様を参照）に I2C_DR レジスタからロードされます。I²C 受信に DMA ストリーム x（ここで x はストリーム番号）を割り当てるには、次のシーケンスを実行します：

1. I2C_DR レジスタのアドレスを DMA_SxPAR レジスタにセットします。データは、毎回の RxNE イベント後に、このアドレスからメモリに移動されます。
2. DMA_SxMA0R レジスタに（およびダブルバッファモードの場合は DMA_SxMA1R レジスタ）メモリアドレスを設定します。データは、毎回の RxNE イベント後に、I2C_DR レジスタからこのメモリ領域にロードされます。
3. 転送されるバイトの総数を DMA_SxNDTR レジスタに設定します。毎回の RxNE イベント後に、この値はデクリメントされます。
4. DMA_SxCR レジスタの PL[0:1] ビットを使用して、DMA ストリームの優先順位を設定します。
5. DMA_SxCR レジスタの DIR ビットをリセットし、アプリケーション条件に応じて 1/2 転送終了後または転送完了後に、割り込みを設定します。
6. DMA_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。

DMA コントローラのレジスタにプログラミングされたデータ転送回数に達したとき、DMA コントローラは、I²C インタフェースに転送終了（EOT/EOT_1）信号を送信し、DMA は DMA チャネル割り込みベクタ上に割り込みを生成します（有効な場合）。

注： DMA が受信に使用される場合、I2C_CR2 レジスタの ITBUFEN ビットを有効にしないでください。

18.3.9 パケットエラーチェック

通信の信頼性を高めるため、PEC 計算回路が実装されています。PEC の計算には、 $C(x) = x^8 + x^2 + x + 1$ という CRC-8 多項式を各ビットに連続的に適用します。

- PEC 計算を有効にするには、I2C_CR1 レジスタの ENPEC ビットをセットします。PEC は、アドレスと R/W ビットを含むすべてのメッセージバイトに基づいて計算される CRC-8 コードです。
 - 送信時、最後のバイトに対応する TxE イベントの後、I2C_CR1 レジスタの PEC 転送ビットをセットします。PEC は、最後に送信されたバイトの後で転送されます。
 - 受信時、最後のバイトに対応する RxNE イベントの後、I2C_CR1 レジスタの PEC ビットをセットすることで、次に受信するバイトが内部的に計算した PEC と等しくない場合にレシーバが NACK を送信するようにします。マスタレシーバの場合、チェックの結果にかかわらず、PEC の後には NACK が続く必要があります。スレーブモードでは、CRC 受信の ACK の前に PEC をセットする必要があります。マスタモードでは、ACK がローにセットされたときにセットする必要があります。
- PECERR エラーフラグ／割り込みは、I2C_SR1 レジスタでも使用できます。
- DMA と PEC 計算機能がともに有効な場合：-
 - 送信時、I²C インタフェースが DMA コントローラから EOT 信号を受信すると、最終バイトの後で自動的に PEC を送信します。
 - 受信時、I²C インタフェースが DMA コントローラから EOT_1 信号を受信すると、次のバイトを自動的に PEC とみなし、その検査を行います。PEC 受信後に DMA リクエストが生成されます。
- 中間での PEC 転送を可能にするため、I2C_CR2 レジスタには制御ビット（LAST ビット）が用意されており、本当に最後の DMA 転送かどうかを判定できます。マスタレシーバの最後の DMA リクエストである場合、最後に受信されたバイトの後で自動的に NACK が送信されます。

- PEC 計算は、アービトレーションロスによって損なわれます。

18.4 I²C 割込み

次の表に、I²C 割込みリクエストの一覧を示します。

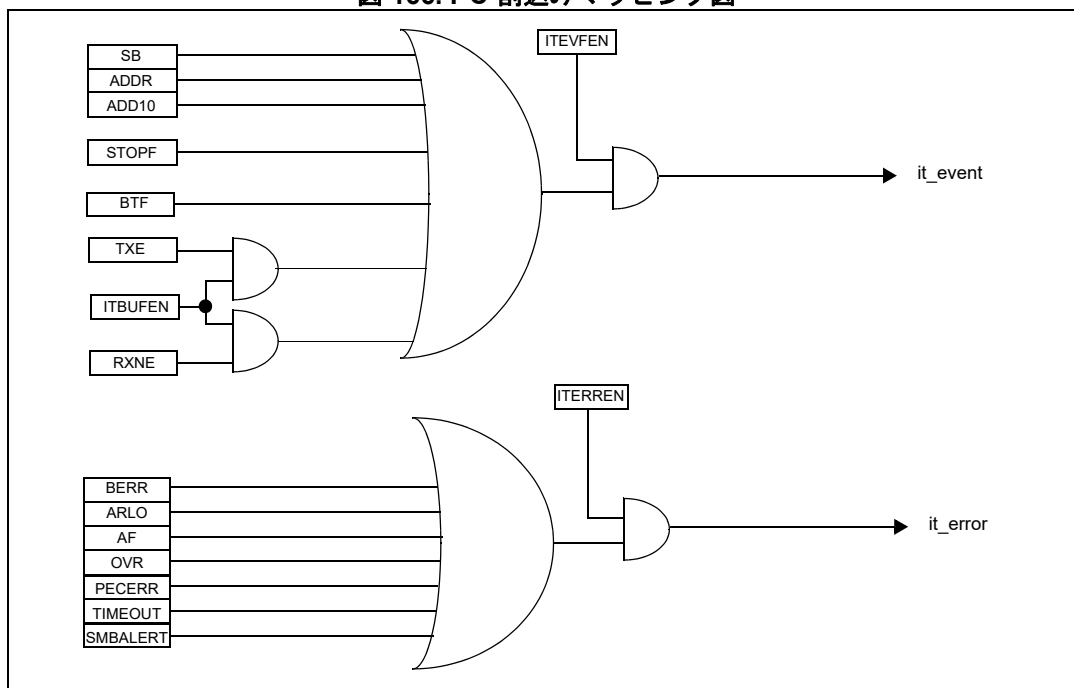
表 71. I²C 割込みリクエスト

割込みイベント	イベントフラグ	有効制御ビット
START ビット送信 (マスタ)	SB	ITEVFEN
アドレス送信 (マスタ) またはアドレス一致 (スレーブ)	ADDR	
10 ビットヘッダ送信 (マスタ)	ADD10	
STOP 受信 (スレーブ)	STOPF	
データバイト転送終了	BTF	
受信バッファノットエンプティ	RXNE	ITEVFEN および ITBUFEN
送信バッファエンプティ	TXE	
バスエラー	BERR	ITERREN
アービトレーションロスト (マスタ)	ARLO	
確認応答障害	AF	
オーバーラン/アンダーラン	OVR	
PEC エラー	PECERR	
タイムアウト/T lowエラー	TIMEOUT	
SMBus アラート	SMBALERT	

注 : SB、ADDR、ADD10、STOPF、BTF、RxNE、および TxE の各ビットは、同じ割込みチャンネル上で論理和がとられます。

BERR、ARLO、AF、OVR、PECERR、TIMEOUT、および SMBALERT の各ビットは、同じ割込みチャンネル上で論理和がとられます。

図 166. I²C 割込みマッピング図



18.5 I²C デバッグモード

マイクロコントローラがデバッグモードに入ると（FPU 搭載 Cortex[®]-M4 コア 停止）、DBG モジュールの DBG_I2Cx_SMBUS_TIMEOUT 設定ビットに応じて、SMBUS タイムアウトは、通常の動作を続行するか、あるいは停止します。詳細については、[セクション 23.16.2 : 816 ページのタイマ、ウォッチドッグ、および I²C のデバッグサポート](#)を参照してください。

18.6 I²C レジスタ

レジスタの説明で使用されている略語のリストについては、[34 ページのセクション 1.1](#)を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

18.6.1 I²C 制御レジスタ 1 (I2C_CR1)

アドレスオフセット : 0x00
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Res.	ALERT	PEC	POS	ACK	STOP	START	NO STRETCH	ENGCG	ENPEC	ENARP	SMB TYPE	Res.	SMBUS	PE
r/w		r/w	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 **SWRST** : ソフトウェアリセット

このビットがセットされると、I²C はリセット状態になります。このビットをリセットする前に、I²C ラインが解放されていて、バスが空いていることを確認してください。

0 : I²C ペリフェラルはリセット状態ではありません。

1 : I²C ペリフェラルはリセット状態です。

注： このビットを使って、エラー状態またはロック状態の後にペリフェラルを再初期化することができます。例えば、BUSY ビットがセットされ、バス上のグリッチのためにロックされたままになった場合、SWRST ビットを使ってこの状態から抜け出すことができます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **ALERT** : SMBus アラート

このビットは、ソフトウェアによってセット／クリアされ、PE=0 のときにハードウェアによってクリアされます。

0 : SMBA ピンをハイレベルに解放します。アラート応答アドレスヘッダの後には NACK が続きます。

1 : SMBA ピンをローレベルに駆動します。アラート応答アドレスヘッダの後には ACK が続きます。

ビット 12 **PEC** : パケットエラーチェック

このビットは、ソフトウェアによってセット／クリアされ、PEC が転送されたとき、START/STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

0 : PEC は転送されません。

1 : PEC が転送されます（送信または受信モード）。

注： PEC 計算は、アービトラションロスによって損なわれます。

ビット 11 POS : 確認応答/PEC 位置 (データ受信)

このビットは、ソフトウェアによってセット/クリアされ、PE=0 のときにハードウェアによってクリアされます。

0 : ACK ビットは、シフトレジスタに受信されている現在のバイトの (N)ACK を制御します。PEC ビットは、シフトレジスタ内の現在のバイトが PEC であることを示します。

1 : ACK ビットは、シフトレジスタに受信される次のバイトの (N)ACK を制御します。PEC ビットは、シフトレジスタ内の次のバイトが PEC であることを示します。

注 : POS ビットは、マスタモードの 2 バイト受信設定でのみ使用する必要があります。また、[セクション : 475 ページのマスタレシーバ](#)で推奨する 2 バイト受信手順に説明されているように、データ受信の開始前に設定する必要があります。

ビット 10 ACK : 確認応答イネーブル

このビットは、ソフトウェアによってセット/クリアされ、PE=0 のときにハードウェアによってクリアされます。

0 : 確認応答は返されません。

1 : バイトの受信後 (アドレスまたはデータの一致)、確認応答が返されます。

ビット 9 STOP : STOP 生成

このビットは、ソフトウェアによってセット/クリアされ、STOP コンディションの検出時にハードウェアによってクリアされ、タイムアウトエラーの検出時にハードウェアによってセットされます。

マスタモード :

0 : STOP が生成されません。

1 : 現在のバイト転送後、または現在の START コンディションの送信後に、STOP が生成されます。

スレーブモード :

0 : STOP が生成されません。

1 : 現在のバイト転送後に SCL ラインと SDA ラインを解放します。

ビット 8 START : START 生成

このビットは、ソフトウェアによってセット/クリアされ、START の送信時または PE=0 のときにハードウェアによってクリアされます。

マスタモード :

0 : START は生成されません。

1 : 反復 START が生成されます。

スレーブモード :

0 : START は生成されません。

1 : バスが空いていれば、START が生成されます。

ビット 7 NOSTRETCH : クロックストレッチディセーブル (スレーブモード)

このビットは、スレーブモードで ADDR または BTF フラグがセットされているとき、ソフトウェアによってリセットされるまで、クロックストレッチを無効にするために使用されます。

0 : クロックストレッチ有効

1 : クロックストレッチ無効

ビット 6 ENGCG : 同報イネーブル

0 : 同報は無効です。アドレス 00h は NACK されます。

1 : 同報は有効です。アドレス 00h は ACK されます。

ビット 5 ENPEC : PEC イネーブル

0 : PEC 計算は無効です。

1 : PEC 計算は有効です。

ビット 4 ENARP : ARP イネーブル

0 : ARP は無効です。

1 : ARP は有効です。

SMBTYPE=0 の場合、SMBus デバイスのデフォルトアドレスが認識されます。

SMBTYPE=1 の場合、SMBus ホストのアドレスが認識されます。

ビット 3 **SMBTYPE** : SMBus タイプ

0 : SMBus デバイス

1 : SMBus ホスト

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SMBUS** : SMBus モード

0 : I²C モード

1 : SMBus モード

ビット 0 **PE** : ペリフェラルは有効です。

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

注 : 通信中にこのビットがリセットされた場合、現在の通信の最後にアイドル状態に戻ったとき、ペリフェラルは無効になります。

 通信の最後には PE=0 になるので、すべてのビットがリセットされます。

 マスタモードでは、通信が終るまで、このビットをリセットしてはなりません。

注 : STOP、START、または PEC ビットがセットされると、このビットがハードウェアによってクリアされるまで、ソフトウェアは I2C_CR1 に書き込みアクセスを実行してはいけません。さもなければ、第 2 の STOP、START、または PEC リクエストを設定する危険性があります。

18.6.2 I²C 制御レジスタ 2 (I2C_CR2)

アドレスオフセット : 0x04															
リセット値 : 0x0000															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み			LAST	DMA EN	ITBUFE N	ITEVTE N	ITERRE N	予約済み			FREQ[5:0]				
			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 **LAST** : DMA 最終転送

0 : 次の DMA EOT は最終転送ではありません。

1 : 次の DMA EOT は最終転送です。

注 : このビットは、最後の受信データでの NACK の生成を可能にするために、マスタレーバモードで使用されます。

ビット 11 **DMAEN** : DMA リクエストイネーブル

0 : DMA リクエストは無効です。

1 : TxE=1 または RxNE =1 のとき、DMA リクエストは有効です。

ビット 10 **ITBUFEN** : バッファ割込みイネーブル

0 : TxE = 1 または RxNE = 1 で、割込みは生成されません。

1 : TxE = 1 または RxNE = 1 で、イベント割込みを生成します (DMAEN ビットの状態には依存しません)。

ビット 9 **ITEVTEN** : イベント割込みイネーブル

0 : イベント割込みは無効です。

1 : イベント割込みは有効です。

この割込みは、以下の場合に生成されます。

- SB = 1 (マスタ)
- ADDR = 1 (マスタ/スレーブ)
- ADD10 = 1 (マスタ)
- STOPF = 1 (スレーブ)
- BTF = 1 (ただし、TxE または RxNE イベントが生成してしないとき)
- ITBUFEN = 1 で TxE イベントが 1 になったとき
- ITBUFEN = 1 で RxNE イベントが 1 になったとき

ITERREN : エラー割込みイネーブル

0 : エラー割込みは無効です。

1 : エラー割込みは有効です。

この割込みは、以下の場合に生成されます。

- BERR = 1
- ARLO = 1
- AF = 1
- OVR = 1
- PECERR = 1
- TIMEOUT = 1
- SMBALERT = 1

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **FREQ[5:0]** : ペリフェラルクロック周波数

ペリフェラルクロック周波数は、入力 APB クロック周波数 (I²C ペリフェラルは APB に接続) を使用して設定する必要があります。FREQ フィールドは、ペリフェラルによって使用され、I²C 仕様に準拠したデータセットアップとホールド時間を生成します。最小許容周波数は 2 MHz であり、最大周波数は最大 APB1 周波数で制限され、50 MHz (ペリフェラルの本質的な上限) を超えることはできません。

0b000000 : 設定禁止

0b000001 : 設定禁止

0b000010 : 2 MHz

...

0b110010 : 50 MHz

0b100100 を超える場合 : 設定禁止

18.6.3 I²C 自己アドレスレジスタ 1 (I2C_OAR1)

アドレスオフセット : 0x08
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD MODE	予約済み					ADD[9:8]		ADD[7:1]							ADD0
rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 15 **ADDMODE** アドレスモード (スレーブモード)
0 : 7 ビットスレーブアドレス (10 ビットアドレスは確認応答されません)
1 : 10 ビットスレーブアドレス (7 ビットアドレスは確認応答されません)
- ビット 14 ソフトウェアによって常に 1 に保持してください。
- ビット 13:10 予約済みであり、リセット値に保持する必要があります。
- ビット 9:8 **ADD[9:8]** : インタフェースアドレス
7 ビットアドレッシングモード : 無視されます。
10 ビットアドレッシングモード : アドレスのビット 9:8。
- ビット 7:1 **ADD[7:1]** : インタフェースアドレス
アドレスのビット 7:1
- ビット 0 **ADD0** : インタフェースアドレス
7 ビットアドレッシングモード : 無視されます。
10 ビットアドレッシングモード : アドレスのビット 0。

18.6.4 I²C 自己アドレスレジスタ 2 (I2C_OAR2)

アドレスオフセット : 0x0C
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								ADD2[7:1]							ENDUAL
								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 7:1 **ADD2[7:1]** : インタフェースアドレス
デュアルアドレスモードでアドレスのビット 7:1
- ビット 0 **ENDUAL** : デュアルアドレスモードイネーブル
0 : 7 ビットアドレスモードで、OAR1 のみが認識されます。
1 : 7 ビットアドレスモードで、OAR1 と OAR2 の両方が認識されます。

18.6.5 I²C データレジスタ (I2C_DR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								DR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DR[7:0]** 8 ビットデータレジスタ

受信されたバイト、またはバスに送信されるバイト。

- トランスマッタモード：データレジスタにバイトが書き込まれると、バイト送信が自動的に始まります。送信が始まり (TxNE=1)、次に送信されるデータがデータレジスタに書き込まれた場合、連続的な送信ストリームを維持できます。
- レシーバモード：受信したバイトはデータレジスタにコピーされます (RxNE=1)。データレジスタが読み出されてから次のデータバイトが受信された場合 (RxNE=1)、連続的な送信ストリームを維持できます。

注： スレーブモードでは、アドレスはデータレジスタにコピーされません。

書き込み衝突は管理されません (TxNE=0 の場合、データレジスタは書き込み可能です)。

ACK パルスで ARLO イベントが発生した場合、受信したバイトはデータレジスタにコピーされないため、読出しできません。

18.6.6 I²C ステータスレジスタ 1 (I2C_SR1)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB ALERT	TIMEO UT	Res.	PEC ERR	OVR	AF	ARLO	BERR	TXE	RXNE	Res.	STOPF	ADD10	BTF	ADDR	SB
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

ビット 15 **SMBALERT** : SMBus アラート

SMBus ホストモードで、

0 : SMBALERT はありません。

1 : SMBALERT イベントがピンに発生しました。

SMBus スレーブモードで、

0 : SMBALERT 応答アドレスヘッダはありません。

1 : SMBALERT LOW への SMBALERT 応答アドレスヘッダを受信しました。

- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ビット 14 **TIMEOUT** : タイムアウトまたはタイマロー期間エラー

0 : タイムアウトエラーは発生していません。

1 : SCL は 25 ms の間ローレベルを維持しました (タイムアウト)。

または

マスタのクロックロー累積延長時間が 10 ms を超えました (Tlow:mext)。

または

スレーブのクロックロー累積伸長時間が 25 ms を超えました (Tlow:sext)。

- スレーブモードに設定した場合 : スレーブは通信をリセットし、ラインはハードウェアによって解放されます。
- マスタモードに設定した場合 : STOP コンディションがハードウェアによって送信されます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

注 : この機能は SMBus モードでのみ使用できます。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **PECERR** : 受信時の PEC エラー

0 : PEC エラーなし。レシーバは PEC 受信後に ACK を返します (ACK=1 の場合)。

1 : PEC エラーあり。レシーバは PEC 受信後に NACK を返します (ACK は任意)。

- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。
- 注 : 受信 CRC が正しくないとき、CRC 受信が終わる前に PEC 制御ビットがセットされない場合には、スレーブモードでの PECERR はセットされません。しかしながら、PEC 値を読み出すと、受信 CRC が正しいか否かが分かります。

ビット 11 **OVR** : オーバーラン/アンダーラン

0 : オーバーラン/アンダーランなし。

1 : オーバーランまたはアンダーランあり。

- スレーブモードで NOSTRETCH=1 の場合、次の条件でハードウェアによってセットされます。
- 受信時、新しいバイト (ACK パルスを含む) が受信され、DR レジスタがまだ読み出されていないとき。新しい受信バイトは失われます。
- 送信時、新しいバイトを送信すべきときに、DR レジスタへの書き込みがまだ行われていないとき。同じバイトが 2 回送信されます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

注 : DR への書き込みが SCL の立ち上がりエッジのごく近くで行われた場合、送信されるデータは不定で、ホールドタイミングエラーが発生します。

ビット 10 **AF** : 確認応答障害

0 : 確認応答障害なし。

1 : 確認応答障害あり。

- 確認応答が返されないとき、ハードウェアによってセットされます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ビット 9 **ARLO** : アービトレーションロスト (マスタモード)

0 : アービトレーションロストは検出されていません。

1 : アービトレーションロストが検出されました。

インタフェースがバスのアービトレーションを他のマスタに開放したとき、ハードウェアによってセットされます。

- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ARLO イベントの後、インタフェースは自動的にスレーブモード (MSL=0) に切り替わります。

注 : SMBUS では、スレーブモードでのデータのアービトレーションは、データフェーズ中または確認応答の送信中にのみ行われます (アドレスの確認応答では行われません)。

ビット 8 BERR : バスエラー

- 0 : START または STOP コンディションの誤配置はありません。
- 1 : START または STOP コンディションが誤配置されました。
- バイト転送中に SCL がハイである間、SDA の立ち上がりまたは立ち下がりがエッジが有効でない位置に発生していることをインタフェースが検出したとき、ハードウェアによってセットされます。
- ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

ビット 7 TxE : データレジスタエンプティ (トランスミッタ)

- 0 : データレジスタは空ではありません。
 - 1 : データレジスタは空です。
 - 送信中に DR が空になったときにセットされます。TxE はアドレスフェーズではセットされません。
 - DR レジスタへのソフトウェア書込みによってクリアされるか、もしくは START/STOP コンディションの後または PE=0 のときにハードウェアによってクリアされます。
 - NACK が受信された場合、または次に送信されるバイトが PEC である場合 (PEC=1)、TxE はセットされません。
- 注：** 送信される最初のデータを書き込んだり、BTF がセットされているときにデータを書き込んだりしても、TxE はクリアされません。いずれの場合も、データレジスタはまだ空だからです。

ビット 6 RxNE : データレジスタノットエンプティ (レシーバ)

- 0 : データレジスタは空です。
 - 1 : データレジスタは空ではありません。
 - レシーバモードで、データレジスタが空でないときにセットされます。RxNE は、アドレスフェーズではセットされません。
 - ソフトウェアで DR レジスタへの読み書きを行ってクリアするか、PE=0 のときにハードウェアによってクリアされます。
 - ARLO イベントの場合、RxNE はセットされません。
- 注：** BTF がセットされているとき、データレジスタはまだ満杯であるため、データを読み出しても RxNE はクリアされません。

ビット 5 予約済みであり、リセット値に保持する必要があります。**ビット 4 STOPF** : STOP 検出 (スレーブモード)

- 0 : STOP コンディションは検出されていません。
 - 1 : STOP コンディションが検出されました。
 - 確認応答後に、スレーブがバス上で STOP コンディションを検出したとき、ハードウェアによってセットされます (ACK=1 の場合)。
 - ソフトウェアで SR1 レジスタを読み出し、続いて CR1 レジスタに書き込むとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。
- 注：** STOPF ビットは、NACK 受信後にはセットされません。
STOPF がセットされた後、完全なクリアシーケンス (READ SR1、続いて WRITE CR1) を実行することを推奨します。図 163 : 471 ページのスレーブ受信の転送シーケンス図を参照してください。

ビット 3 ADD10 : 10 ビットヘッダの送信 (マスタモード)

- 0 : ADD10 イベントは発生していません。
 - 1 : マスタは最初のアドレスバイト (ヘッダ) を送信しました。
 - マスタが 10 ビットアドレスモードで最初のバイトを送信すると、ハードウェアによってセットされます。
 - ソフトウェアで SR1 レジスタを読み出し、続いて DR レジスタに 2 番目のアドレスバイトを書き込むとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。
- 注：** ADD10 ビットは、NACK 受信後にはセットされません。

ビット 2 BTF : バイト転送終了

- 0 : データバイト転送は完了していません。
- 1 : データバイト転送が正常に終了しました。
- NOSTRETCH=0 の場合、次の条件でハードウェアによってセットされます。
- 受信時、新しいバイト (ACK パルスを含む) が受信され、DR レジスタがまだ読み出されていないとき (RxNE=1)。
- 送信時、新しいバイトを送信すべきときに、DR レジスタへの書込みがまだ行われていないとき (TxE=1)。
- DR レジスタへの読み書きによりソフトウェアでクリアされます。または、送信時の START/STOP コンディションの後や PE=0 のときに、ハードウェアによってクリアされます。

注： BTF ビットは、NACK 受信後にはセットされません。

BTF ビットは、次に送信されるバイトが PEC である場合 (I2C_SR2 レジスタの TRA=1 かつ I2C_CR1 レジスタの PEC=1)、セットされません。

ビット 1 ADDR : アドレス送信 (マスタモード) / アドレス一致 (スレーブモード)

このビットは、ソフトウェアで SR1 レジスタを読み出し、続いて SR2 レジスタを読み出すことでクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

アドレス一致 (スレーブ)

- 0 : アドレスは受信されていないか、一致しませんでした。
- 1 : 受信したアドレスが一致しました。
- 受信したスレーブアドレスが OAR レジスタの内容と一致したとき、あるいは同報 / SMBus デバイスデフォルトアドレスまたは SMBus ホスト / SMBus アラートが認識されたとき、ハードウェアによってセットされます。(設定に応じて有効化されている場合)。

注： スレーブモードでは、ADDR がセットされた後、完全なクリアシーケンス (READ SR1、続いて READ SR2) を実行することを推奨します。図 163 : 471 ページのスレーブ受信の転送シーケンス図を参照してください。

アドレス送信 (マスタ)

- 0 : アドレス送信は終了していません。
- 1 : アドレス送信は終了しました。
- 10 ビットアドレッシングでは、2 番目のバイトの ACK 後にビットがセットされます。
- 7 ビットアドレッシングでは、バイトの ACK 後にビットがセットされます。

注： ADDR は、NACK 受信後にはセットされません。

ビット 0 SB : スタートビット (マスタモード)

- 0 : START コンディションなし。
- 1 : START コンディションが生成されました。
- START コンディションが生成されたときにセットされます。
- ソフトウェアで SR1 レジスタを読み出し、続いて DR レジスタへの書込みを行うとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

18.6.7 I²C ステータスレジスタ 2 (I2C_SR2)

アドレスオフセット : 0x18
リセット値 : 0x0000

注 : たとえ I2C_SR1 の読出し後に ADDR フラグがセットされた場合でも、I2C_SR1 の読出し後に I2C_SR2 を読み出すと、ADDR フラグはクリアされます。したがって、I2C_SR1 の ADDR がセットされていると分かったとき、または STOPF ビットがクリアされているときにのみ、I2C_SR2 を読み出す必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	SMB HOST	SMBDEF AULT	GEN CALL	Res.	TRA	BUSY	MSL
r	r	r	r	r	r	r	r	r	r	r	r		r	r	r

- ビット 15:8 **PEC[7:0]** : パケットエラーチェックレジスタ
ENPEC=1 のとき、このレジスタは内部 PEC を保持しています。
- ビット 7 **DUALF** : デュアルフラグ (スレーブモード)
0 : 受信アドレスが OAR1 と一致
1 : 受信アドレスが OAR2 と一致
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。
- ビット 6 **SMBHOST** : SMBus ホストヘッダ (スレーブモード)
0 : SMBus ホストアドレスは受信されていません。
1 : SMBTYPE=1 かつ ENARP=1 のとき、SMBus ホストアドレスが受信されました。
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。
- ビット 5 **SMBDEFAULT** : SMBus デバイスデフォルトアドレス (スレーブモード)
0 : SMBus デバイスデフォルトアドレスは受信されていません。
1 : ENARP=1 のとき、SMBus デバイスデフォルトアドレスが受信されました。
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。
- ビット 4 **GENCALL** : 同報アドレス (スレーブモード)
0 : 同報アドレスは受信されていません。
1 : ENGC=1 のとき、同報アドレスが受信されました。
– STOP コンディションまたは反復 START コンディションの後、あるいは PE=0 のときに、ハードウェアによってクリアされます。
- ビット 3 予約済みであり、リセット値に保持する必要があります。
- ビット 2 **TRA** : トランスミッタ/レシーバ
0 : データバイトが受信されました。
1 : データバイトが送信されました。
このビットは、全アドレスフェーズの最後に、アドレスバイトの R/W ビットに応じてセットされます。
また、STOP コンディション (STOPF=1)、繰り返し START コンディション、バスアービトラーションロスト (ARLO=1) の検出後、または PE=0 のときに、ハードウェアによってクリアされます。

- ビット 1 **BUSY** : バスビジー
- 0 : バス上で通信は行われていません。
 - 1 : バス上で通信が行われています。
- SDA または SCL のローレベル検出時に、ハードウェアによってセットされます。
 - STOP コンディションの検出時に、ハードウェアによってクリアされます。
- バス上で通信が進行中であることを表示します。この情報は、インタフェースが無効である (PE=0) と
きにも更新されます。
- ビット 0 **MSL** : マスタ/スレーブ
- 0 : スレーブモード
 - 1 : マスタモード
- インタフェースがマスタモードになると (SB=1)、ハードウェアによってセットされます。
 - バス上での STOP コンディションまたはアービトレーションロスト (ARLO=1) を検出した後、または
PE=0 のときに、ハードウェアによってクリアされます。

注 : たとえば I2C_SR1 の読出し後に ADDR フラグがセットされた場合でも、I2C_SR1 の読出し後に
I2C_SR2 を読み出すと、ADDR フラグはクリアされます。したがって、I2C_SR1 の ADDR がセット
されていると分かったとき、または STOPF ビットがクリアされているときにのみ、I2C_SR2 を読み
出す必要があります。

18.6.8 I²C クロック制御レジスタ (I2C_CCR)

アドレスオフセット : 0x1C
リセット値 : 0x0000

注 : 標準モード I²C 周波数を実現するためには、f_{PCLK1} は最小 2 MHz 以上である必要があります。高速
モード I²C 周波数を実現するためには、最小 4 MHz 以上である必要があります。400 kHz の最大 I²C
高速 モードクロックに到達するためには、10MHz の整数倍である必要があります。

CCR レジスタは、I²C が無効 (PE=0) の場合にのみ設定する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	予約済み		CCR[11:0]											
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 15 **F/S** : I²C マスタモード選択
- 0 : 標準モード I²C
 - 1 : 高速モード I²C
- ビット 14 **DUTY** : 高速モードデューティサイクル
- 0 : 高速 モード t_{low}/t_{high} = 2
 - 1 : 高速 モード t_{low}/t_{high} = 16/9 (CCR を参照)

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **CCR[11:0]** : 高速／標準モード（マスタモード）でのクロック制御レジスタ

マスタモードにおける SCL クロックを制御します。

標準モードまたは SMBus :

$$T_{\text{high}} = \text{CCR} * T_{\text{PCLK1}}$$

$$T_{\text{low}} = \text{CCR} * T_{\text{PCLK1}}$$

高速モード :

DUTY = 0 の場合 :

$$T_{\text{high}} = \text{CCR} * T_{\text{PCLK1}}$$

$$T_{\text{low}} = 2 * \text{CCR} * T_{\text{PCLK1}}$$

DUTY = 1 の場合 : (400 kHz に到達するには)

$$T_{\text{high}} = 9 * \text{CCR} * T_{\text{PCLK1}}$$

$$T_{\text{low}} = 16 * \text{CCR} * T_{\text{PCLK1}}$$

たとえば、標準モードで 100 kHz の SCL 周波数を生成するには、

FREQR = 08、 $T_{\text{PCLK1}} = 125 \text{ ns}$ の場合、CCR には 0x28 をプログラムする必要があります。

($0x28 \Leftrightarrow 40d \times 125 \text{ ns} = 5000 \text{ ns}$)

注 : 最小許容値は 0x04 です (最小許容値が 0x01 の FAST DUTY モードを除く)

$t_{\text{high}} = t_{\text{r(SCL)}} + t_{\text{w(SCLH)}}$ 。パラメータの定義については、デバイスのデータシートを参照してください。

$t_{\text{low}} = t_{\text{f(SCL)}} + t_{\text{w(SCLL)}}$ 。パラメータの定義については、デバイスのデータシートを参照してください。

I²C 通信速度、 $f_{\text{SCL}} \sim 1/(t_{\text{high}} + t_{\text{low}})$ 。アナログノイズフィルタの入力遅延によって、実際の周波数は異なる場合があります。

CCR レジスタは、I²C が無効 (PE=0) の場合にのみ設定する必要があります。

18.6.9 I²C TRISE レジスタ (I2C_TRISE)

アドレスオフセット : 0x20

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み										TRISE[5:0]					
										rW	rW	rW	rW	rW	rW

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 5:0 **TRISE[5:0]** : 高速／標準モード（マスタモード）における最大立ち上がり時間

これらのビットは、マスタモードで SCL フィードバックループの最大時間を提供します。SCL 立ち上がりエッジ時間にかかわらず、安定した SCL 周波数を保持する目的があります。

これらのビットには、I²C バス仕様で定められている SCL 最大立ち上がり時間より 1 だけ大きな値をプログラミングする必要があります。

たとえば、標準モードで許容される SCL の最大立ち上がり時間は 1000 ns です。

I2C_CR2 レジスタの FREQ[5:0] ビットの値が 0x08 で、 $T_{\text{PCLK1}} = 125 \text{ ns}$ である場合、TRISE[5:0] ビットには 09h をプログラミングする必要があります。

($1000 \text{ ns} / 125 \text{ ns} = 8 + 1$)

フィルタ値も TRISE[5:0] に加算することができます。

結果が整数でない場合、 t_{HIGH} パラメータを考慮して、その整数部を TRISE[5:0] にプログラミングする必要があります。

注 : TRISE[5:0] は、I²C が無効 (PE=0) の場合にのみ設定する必要があります。

18.6.10 I²C FLTR レジスタ (I2C_FLTR)

アドレスオフセット : 0x24
リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み											ANOFF	DNF[3:0]			
											rW	rW	rW	rW	rW

- ビット 15:5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **ANOFF** : アナログノイズフィルタ OFF
- 0 : アナログノイズフィルタ有効
 - 1 : アナログノイズフィルタ無効
- 注 : **ANOFF** は、I²C が無効 (PE=0) の場合にのみ設定する必要があります。
- ビット 3:0 **DNF[3:0]** : デジタルノイズフィルタ
- これらのビットを使って、SDA 入力と SCL 入力のデジタルノイズフィルタを設定します。デジタルフィルタは、最大 DNF[3:0] * TPCLK1 の幅を持つスパイクを除去します。
- 0000 : デジタルノイズフィルタを無効化
 - 0001 : デジタルノイズフィルタを有効化し、最大 1 * TPCLK1 の除去能力を持ちます
 - ...
 - 1111 : デジタルノイズフィルタを有効化し、最大 15 * TPCLK1 の除去能力を持ちます
- 注 : **DNF[3:0]** は、I²C が無効 (PE=0) の場合にのみ設定する必要があります。アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。

18.6.11 I²C のレジスタマップ

次の表に、I²C のレジスタマップとリセット値を示します。

表 72. I²C レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	I2C_CR1	予約済み																SWRST	予約済み	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENG	ENEC	ENARP	SMBTYPE	予約済み	SMBUS	PE				
	リセット値																	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	I2C_CR2	予約済み																			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	予約済み	FREQ[5:0]										
	リセット値																				0	0	0	0	0		0	0	0	0	0	0	0				
0x08	I2C_OAR1	予約済み																ADDMODE	予約済み					ADD[9:8]		ADD[7:1]					ADD0						
	リセット値																	0						0	0	0	0	0	0	0	0						
0x0C	I2C_OAR2	予約済み																							ADD2[7:1]					ENDUAL							
	リセット値																								0	0	0	0	0	0	0	0	0				
0x10	I2C_DR	予約済み																							DR[7:0]												
	リセット値																								0	0	0	0	0	0	0	0					
0x14	I2C_SR1	予約済み																SMBALERT	TIMEOUT	予約済み	PECERR	OVR	AF	ARLO	BERR	TXE	RXNE	予約済み	STOPF	ADD10	BTF	ADDR	SB				
	リセット値																	0	0		0	0	0	0	0	0	0	0		0	0	0	0				
0x18	I2C_SR2	予約済み																PEC[7:0]							DUALF	SMBHOST	SMBDEFAULT	GENCALL	予約済み	TRA	BUSY	MSL					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0				
0x1C	I2C_CCR	予約済み																F/S	DUTY	予約済み	CCR[11:0]																
	リセット値																	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x20	I2C_TRISE	予約済み																							TRISE[5:0]												
	リセット値																								0	0	0	0	1	0							
0x24	I2C_FLTR	予約済み																											DNF[3:0]								
	リセット値																												0	0	0	0					

レジスタ境界アドレス表については、[38 ページの表 1](#) を参照してください。

19 USART (Universal synchronous asynchronous receiver transmitter)

19.1 USART の概要

USART (Universal Synchronous Asynchronous Receiver Transmitter) を使用すると、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟性の高い全二重データ交換を行うことができます。USART は、フラクショナルボーレートジェネレータを使用して、非常に広範囲のボーレートを提供します。

USART は、同期式単方向通信と半二重単線通信をサポートします。また、LIN (Local Interconnection Network)、スマートカードプロトコルおよび IrDA (infrared data association) SIR ENDEC 仕様、およびモデム動作 (CTS/RTS) もサポートします。マルチプロセッサ通信も可能です。

また、マルチバッファ設定用の DMA を使用して、高速データ通信が可能です。

19.2 USART の主な機能

- 全二重、非同期通信
- NRZ 標準フォーマット (マーク/スペース)
- 16 倍または 8 倍に設定可能なオーバーサンプリング方式によって、速度とクロックの柔軟な許容誤差を実現
- フラクショナルボーレートジェネレータシステム
 - 共通のプログラム可能な送信および受信ボーレート (最大 APB 周波数におけるボーレート値については、データシートを参照)。
- プログラム可能なデータワード長 : 8 または 9 ビット
- 設定可能なストップビット : 1 個または 2 個のストップビットをサポート
- LIN マスタの同期ブレイク送信機能と LIN スレーブのブレイク検出機能
 - USART が LIN 用にハードウェア設定されている場合、13 ビットのブレイク生成と 10/11 ビットのブレイク検出
- 同期送信用のトランスミッタクロック出力
- IrDA SIR エンコーダデコーダ
 - 通常モード用に 3/16 ビット持続時間のサポート
- スマートカードエミュレーション機能
 - スマートカードインタフェースは、ISO 7816-3 規格で定義された非同期プロトコルのスマートカードをサポート
 - スマートカード動作用に 0.5 個、1.5 個のストップビット
- 単線半二重通信
- DMA (Direct Memory Access) を使用した設定可能なマルチバッファ通信
 - 集中型 DMA を使用して、送受信バイトを専用 SRAM にバッファリング
- トランスミッタとレシーバ用に個別の有効ビット
- 転送検出フラグ :
 - 受信バッファフル
 - 送信バッファエンプティ
 - 送信終了フラグ

- パリティ制御：
 - パリティビットの送信
 - 受信したデータバイトのパリティ検査
- 4つのエラー検出フラグ：
 - オーバーランエラー
 - ノイズ検出
 - フレームエラー
 - パリティエラー
- 10種の割込みソース（フラグ付き）
 - CTSの変化
 - LINブレーク検出
 - 送信データレジスタエンプティ
 - 送信完了
 - 受信データレジスタフル
 - アイドルラインの受信
 - オーバーランエラー
 - フレーミングエラー
 - ノイズエラー
 - パリティエラー
- マルチプロセッサ通信：アドレス不一致の場合、ミュートモードに遷移
- ミュートモードからのウェイクアップ（アイドルライン検出またはアドレスマーク検出による）
- 2種のレシーバウェイクアップモード：アドレスビット（MSB、9番目のビット）、アイドルライン

19.3 USART の機能詳細

インタフェースは、3個のピンによって他のデバイスに外部接続されます（[図 167](#) を参照）。USART の双方向通信には、少なくとも2本のピンが必要です。すなわち、受信データ入力（RX）と送信データ出力（TX）です。

RX：受信データ入力はシリアルデータ入力です。データ復旧にはオーバーサンプリング技術が使用され、有効な入力データとノイズを区別しています。

TX：送信データ出力です。トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、送信すべきデータがないとき、TXピンはハイレベルになります。単線モードとスマートカードモードでは、この入出力を使用してデータを送受信します（USART レベルでは、データは続いて SW_RX で受信されます）。

通常の USART モードでは、フレームは以下の部分で構成され、これらのピンを通じてシリアルデータが送受信されます。

- 送受信前のアイドルライン
- スタートビット
- データワード (8 または 9 ビット)、LSB ファースト
- フレームの完了を示す 0.5 個、1 個、1.5 個、2 個のストップビット
- このインタフェースはフラクショナルボーレートジェネレータを使用します - 12 ビットの整数部と 4 ビットの小数部
- ステータスレジスタ (USART_SR)
- データレジスタ (USART_DR)
- ボーレートレジスタ (USART_BRR) - 12 ビットの整数部と 4 ビットの小数部
- ガードタイムレジスタ (USART_GTPR) - スマートカードモードの場合

各ビットの定義については、[セクション 19.6 : 540 ページのUSART レジスタ](#)を参照してください。

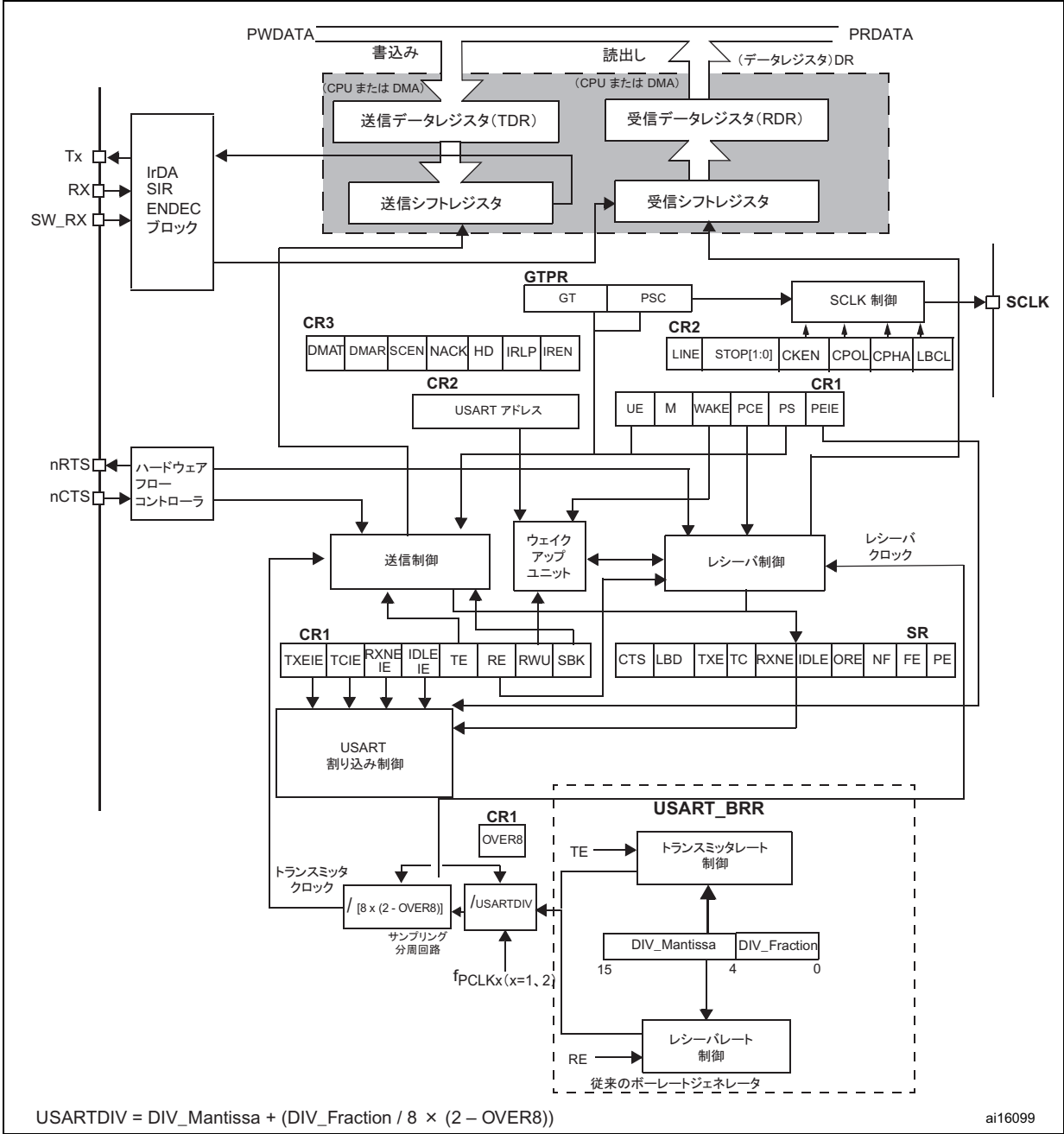
同期モードでのインタフェースには次のピンが必要になります。

- **SCLK** : トランスミッタクロック出力。このピンは、SPI マスタモードに対応する同期送信用のトランスミッタデータクロックを出力します (スタートビットとストップビットのクロックパルスはなく、ソフトウェアオプションで最後のデータビットのクロックパルスを送信します)。これと並行して、RX でデータを同期受信できます。これを使用して、シフトレジスタを持つペリフェラル (例 : LCD ドライバ) を制御できます。クロックの位相と極性は、ソフトウェアでプログラム可能です。スマートカードモードでは、SCLK はスマートカードヘクロックを供給できます。

ハードウェアフロー制御モードでは次のピンが必要になります。

- **nCTS** : Clear To Send は、ハイレベルのとき、現在の転送の終わりにデータ送信をブロックします。
- **nRTS** : Request To Send は、ローレベルのとき、USART がデータを受信できることを示します。

図 167. USART のブロック図



ai16099

19.3.1 USART キャラクタの説明

ワード長は、USART_CR1 レジスタの M ビットをプログラムすることによって、8 または 9 ビットとして選択できます（図 168 を参照）。

TX ピンは、スタートビットの処理中にはロー状態です。また、ストップビットの処理中にはハイ状態です。

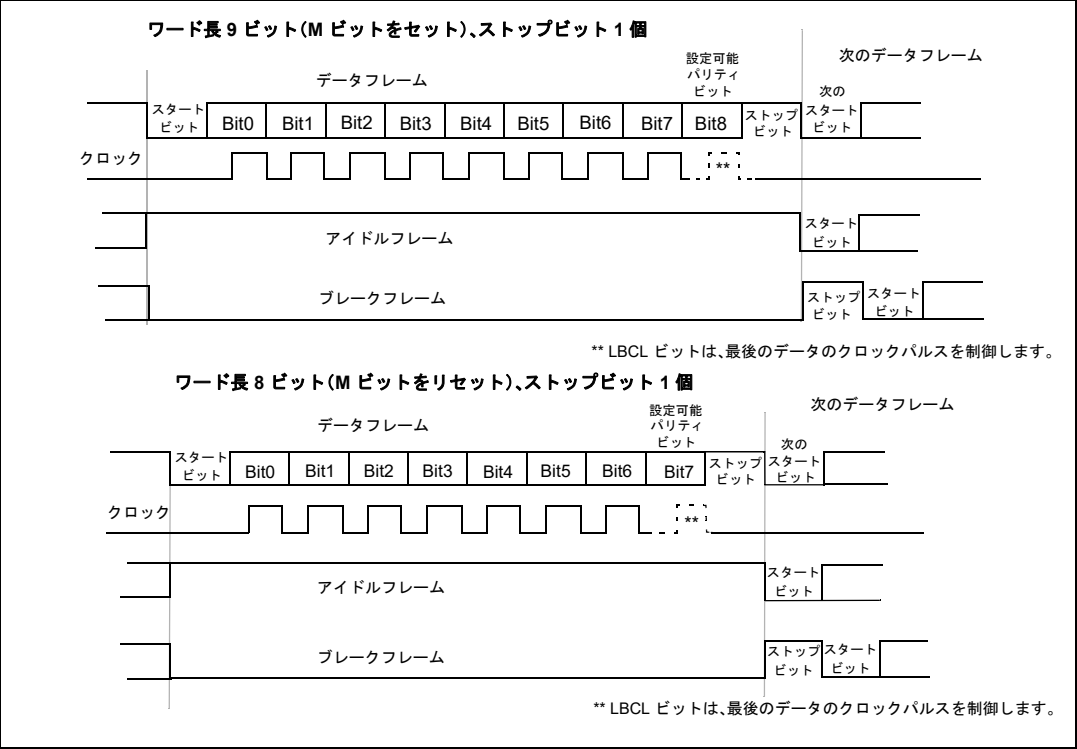
アイドルキャラクタは、1 フレーム分の「1」であると解釈され、この後にデータが格納された次のフレームのスタートビットが続きます（「1」の数にはストップビットの数も含まれます）。

ブレークキャラクタは、フレーム周期中に「0」を受信することと解釈されます。トランスミッタは、ブレークフレームの終わりに 1 個または 2 個のストップビット（論理「1」ビット）を挿入して、スタートビットを承認します。

送信と受信は、共通のボーレートジェネレータによって駆動され、それぞれのクロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。

図 168. ワード長のプログラミング



19.3.2 トランスミッタ

トランスミッタは、M ビットのステータスに応じて、8 または 9 ビットのデータワードを送信できます。送信有効ビット (TE) がセットされているとき、送信シフトレジスタ内のデータは TX ピンに出力され、対応するクロックパルスは SCLK ピンに出力されます。

キャラクタ送信

USART 送信時、データは LSB ファーストで TX ピンにシフトアウトされます。このモードでは、USART_DR レジスタは、内部バスと送信シフトレジスタの間のバッファ (TDR) で構成されます (図 167 を参照)。

各キャラクタの前には、スタートビット (1 ビット周期、論理レベルロー) があります。キャラクタは、設定可能な数のストップビットで終端されます。

USART では、0.5、1、1.5、および 2 個のストップビットがサポートされます。

注： データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、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

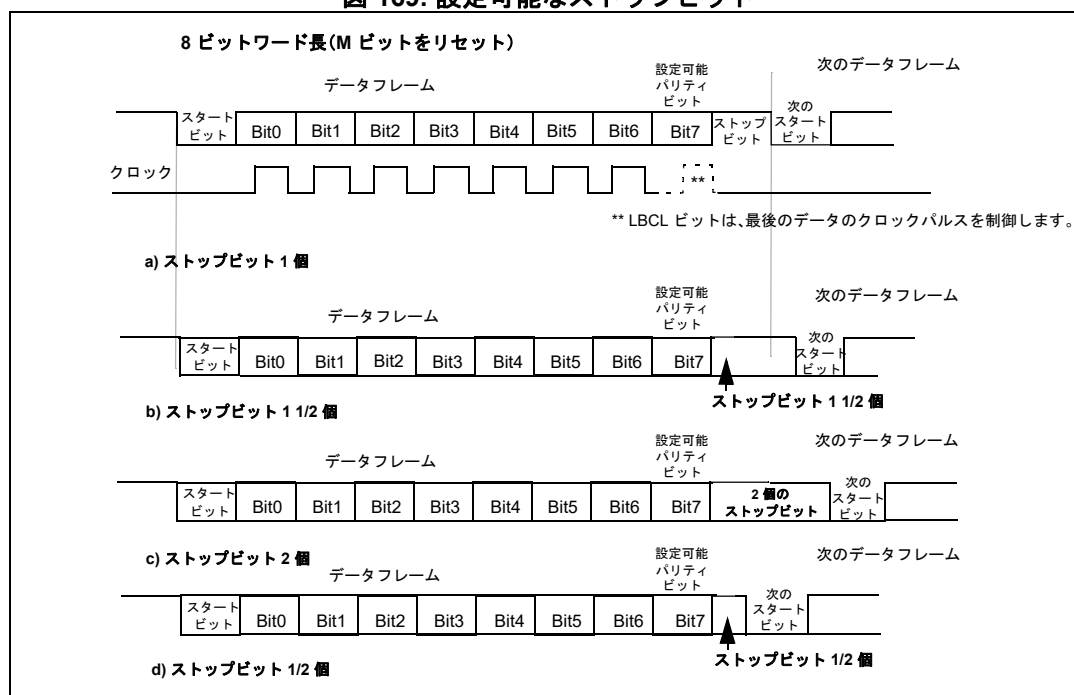
各キャラクタとともに送信されるストップビットの数は、制御レジスタ 2 のビット 13、12 でプログラミングできます。

- **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。
- **0.5 個のストップビット：** スマートカードモードでのデータの受信に使用されます。
- **1.5 個のストップビット：** スマートカードモードでのデータの送受信に使用されます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信では、10 個 ($m = 0$ の場合) または 11 個 ($m = 1$ の場合) のローレベルビットの後に、設定された数のストップビットが続きます。ローレベルビット 10/11 個を超える長いブ레이크を送信することはできません。

図 169. 設定可能なストップビット



手順：

1. USART_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
2. USART_CR1 の M ビットをプログラミングして、ワード長を定義します。
3. USART_CR2 レジスタでストップビットの数をプログラミングします。
4. マルチバッファ通信を行う場合は、USART_CR3 レジスタの DMA 有効 (DMAT) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
5. USART_BRR レジスタを使用して、希望するボーレートを選択します。
6. USART_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。
7. 送信するデータを USART_DR レジスタに書き込みます (これによって TXE ビットがクリアされます)。シングルバッファの場合、送信される各データにこれを繰り返します。
8. USART_DR レジスタに最後のデータを書き込んだ後、TC=1 になるまで待ちます。これは、最後のフレームの送信が完了したことを示します。これは、たとえば最後の送信内容の破壊を避けるために、USART が無効になったり HALT モードに入ったりする場合に必要です。

1 バイト通信

TXE ビットは、常にデータレジスタへの書き込みによってクリアされます。

TXE ビットは、ハードウェアによってセットされ、次を示します。

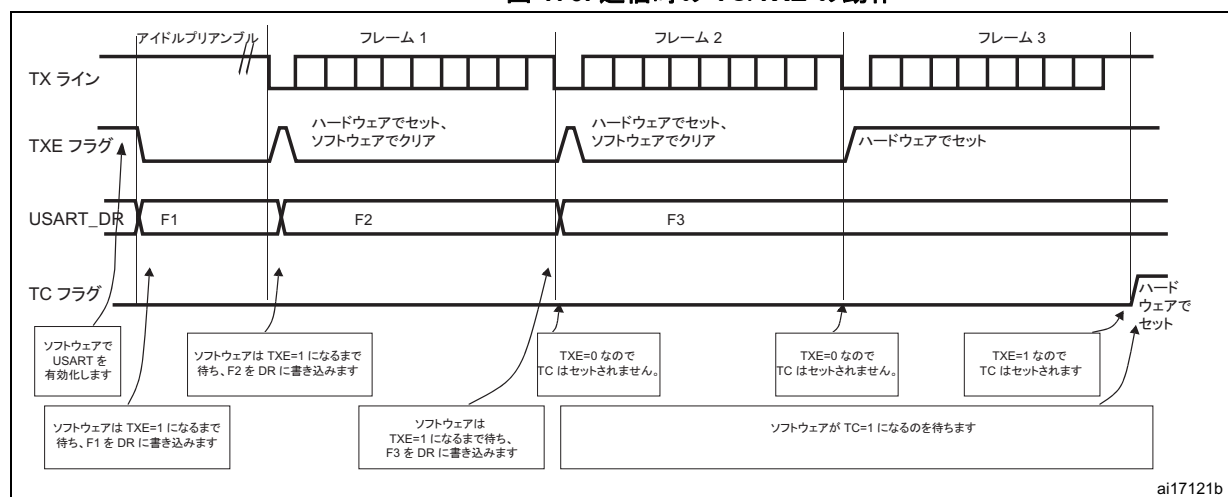
- データが TDR からシフトレジスタに移動し、データ送信が開始しています。
- TDR レジスタは空です。
- 次のデータを、前のデータに上書きすることなく、USART_DR レジスタに書き込みます。

TXEIE ビットがセットされている場合、このフラグは割込みを生成します。

送信が行われている場合、USART_DR レジスタへの書き込み命令によってデータが TDR レジスタに格納され、さらに、現在の送信の最後にシフトレジスタにコピーされます。

1. USART_SR レジスタからの読出し
2. USART_DR レジスタへの書込み

図 170. 送信時の TC/TXE の動作



19.3.3 レシーバ

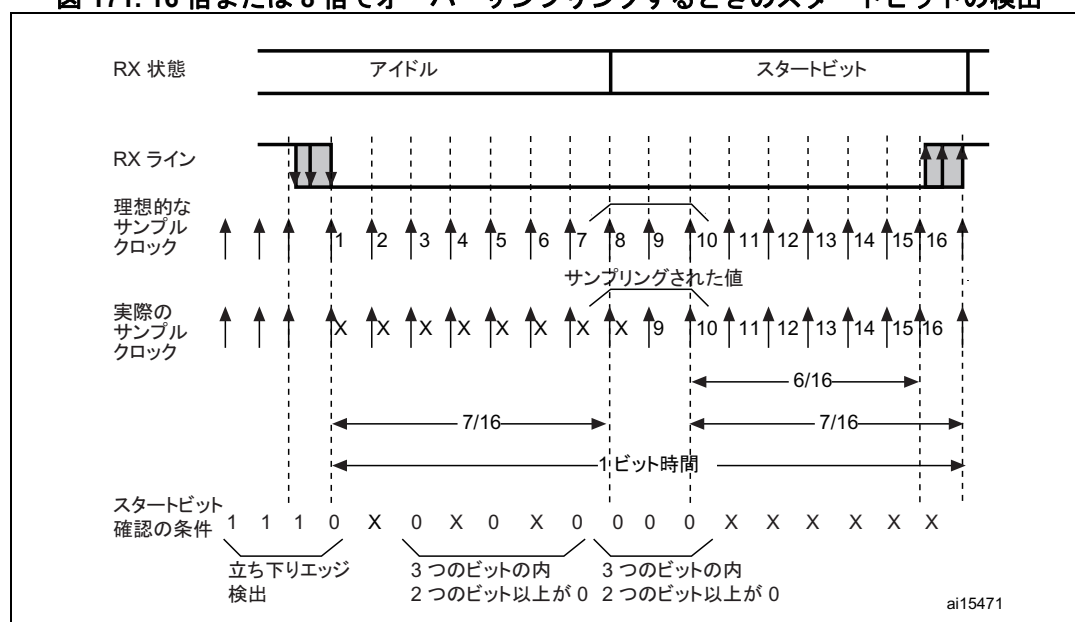
USART は、USART_CR1 レジスタの M ビットに応じて、8 ビットまたは 9 ビットのデータワードを受信できます。

スタートビット検出

スタートビット検出シーケンスは、16 倍または 8 倍でオーバーサンプリングするときと同じです。

USART では、サンプルの特定シーケンスが認識されると、スタートビットが検出されます。このシーケンスは、1110X0X0X0000 です。

図 171. 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 の場合は割込みが生成されます）。

両方のサンプリングでサンプリングされた 3 つのビットのうち、少なくとも 2 つが 0 の場合（第 3、第 5、および第 7 ビットでのサンプリングと、第 8、第 9、および第 10 ビットでのサンプリング）、スタートビットは確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割込みが生成されます）、NE ノイズフラグがセットされます。この条件が満たされない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻ります（フラグはセットされません）。

いずれか一方のサンプリングで（第 3、第 5、および第 7 ビットでのサンプリングまたは第 8、第 9、および第 10 ビットでのサンプリング）、3 つのビットのうち 2 つが 0 の場合、スタートビットは確認されますが、NE ノイズフラグビットがセットされます。

キャラクタの受信

USART の受信時には、データは RX ピンを通じて LSB ファーストでシフトインされます。このモードでは、USART_DR レジスタは、内部バスと受信シフトレジスタの間のバッファ (RDR) で構成されます。

手順 :

1. USART_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
2. USART_CR1 の M ビットをプログラミングして、ワード長を定義します。
3. USART_CR2 レジスタでストップビットの数をプログラミングします。
4. マルチバッファ通信を行う場合には、USART_CR3 レジスタの DMA 有効 (DMAR) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。手順 3
5. ボーレートレジスタ USART_BRR を使用して、希望するボーレートを選択します。
6. USART_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

キャラクタが受信されると、

- RXNE ビットがセットされます。これは、シフトレジスタの内容が RDR レジスタに転送されたことを示します。言い換えると、データは受信され、読出し可能です (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、割込みが生成されます。
- 受信中にフレームエラー、ノイズまたはオーバーランエラーが検出された場合、エラーフラグをセットできます。
- マルチバッファでは、RXNE ビットはバイトを受信するたびにセットされ、データレジスタへの DMA 読出しによってクリアされます。
- シングルバッファモードでは、RXNE ビットは USART_DR レジスタへのソフトウェア読出しによってクリアされます。RXNE フラグは、0 を書き込むとクリアできます。オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE ビットをクリアする必要があります。

注 : データの受信中には、RE ビットをリセットしないでください。受信中に RE ビットが無効にされた場合、現在のバイトの受信が中止されます。

ブレークキャラクタ

ブレークキャラクタを受信すると、USART はブレークキャラクタをフレーミングエラーとして処理します。

アイドルキャラクタ

アイドルフレームが検出された場合、データ受信キャラクタと同じ手順があり、IDLEIE ビットがセットされている場合は割込みが生成されます。

オーバーランエラー

RXNE ビットがリセットされていないときにキャラクタを受信すると、オーバーランエラーが発生します。RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。

RXNE フラグは、バイトを受信するたびにセットされます。次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

- ORE ビットがセットされます。
- RDR の内容は失われません。USART_DR への読み出しが行われると、前回のデータが読み出されます。
- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXNEIE ビットがセットされた場合、または EIE ビットと DMAR ビットの両方がセットされた場合は、割込みが生成されます。
- ORE ビットは、USART_SR レジスタの読み出しと、それに続く USART_DR レジスタの読み出し動作によってリセットされます。

注： ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。この場合、次の 2 つの可能性がありえます。

- RXNE=1 の場合、有効な最後のデータは、受信レジスタ RDR に格納され、読み出しが可能です。
- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR には読み出すべきものがないことを意味します。このケースは、有効な最後のデータが RDR で読み出されると同時に新しい（そして失われた）データが受信されると発生します。また、読み出しシーケンス中（USART_SR レジスタと USART_DR レジスタの読み出しアクセスの間）に新しいデータが受信された場合にも発生します。

適切なオーバーサンプリング方式の選択

レシーバは有効な受信データとノイズを区別して、データを復旧するユーザ設定可能なさまざまなオーバーサンプリング技術を実装しています（同期モードの場合を除く）。

オーバーサンプリング方式は、USART_CR1 レジスタの OVER8 ビットをプログラムすることによって選択でき、ボーレートクロックの 16 倍または 8 倍にすることができます（[図 172](#) および [図 173](#)）。

アプリケーションに応じて、

- 高速（最大 $f_{PCLK}/8$ ）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 19.3.5 : 522 ページのクロック偏差に対する USART レシーバの許容誤差](#)を参照）。
- クロック偏差に対するレシーバの許容誤差を増やすには、16 倍（OVER8=0）のオーバーサンプリングを選択します。この場合、最大速度は $f_{PCLK}/16$ に制限されます。

論理レベルの評価方法を選択するには、USART_CR3 レジスタの ONEBIT ビットをプログラミングします。これには 2 つのオプションがあります。

- 受信されたビットの中央にある 3 つのサンプルの多数決。この場合、多数決に使用された 3 つのサンプルが等しくないとき、NF ビットがセットされます。
- 受信されたビットの中央にある 1 つのサンプル。
アプリケーションに応じて、
 - ノイズの多い環境で操作するときは 3 つのサンプルの多数決方式 (ONEBIT=0) を選択します。そしてノイズが検出された場合 (図 73 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 19.3.5 : 522 ページの [クロック偏差に対する USART レシーバの許容誤差](#) を参照)。この場合、NF ビットはセットされません。

フレーム内でノイズが検出された場合 :

- RXNE ビットの立ち上がりエッジで、NF ビットがセットされます。
- 無効なデータがシフトレジスタから USART_DR レジスタへ転送されます。
- シングルバイト通信の場合、割込みは生成されません。ただし、このビットは、割込みを生成する 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込みが発行されます。

NF ビットは、USART_SR レジスタの読出し動作と、それに続く USART_DR レジスタの読出し動作によってリセットされます。

注 : 8 倍のオーバーサンプリングは、スマートカード、IrDA、および LIN の各モードでは使用できません。これらのモードでは、OVER8 ビットはハードウェアによって“0”に固定されています。

図 172. データサンプリング (16 倍のオーバーサンプリング)



図 173. データサンプリング (8 倍のオーバーサンプリング)

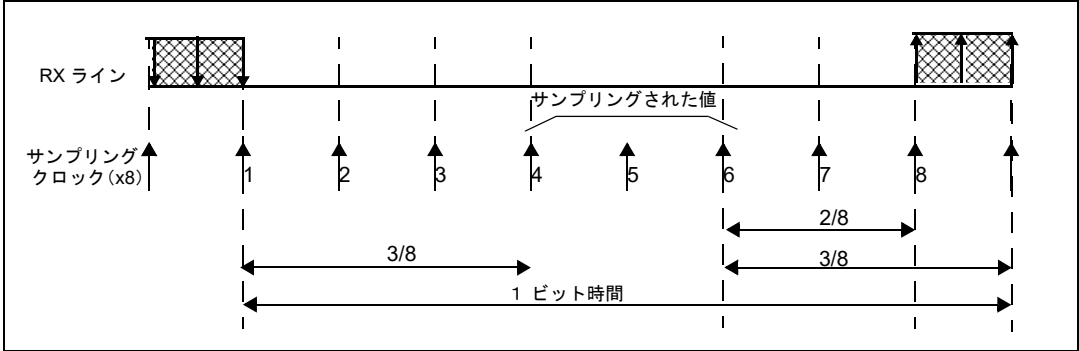


表 73. サンプリングされたデータからのノイズ検出

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

フレーミングエラーは、次の場合に検出されます。

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから USART_DR レジスタへ転送されます。
- シングルバイト通信の場合、割込みは生成されません。ただし、このビットは、割込みを生成する 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込みが発行されます。

FE ビットは、USART_SR レジスタの読出し動作と、それに続く USART_DR レジスタの読出し動作によってリセットされます。

受信時の設定可能なストップビット

受信するストップビット数は、制御レジスタ 2 の制御ビットを通じて設定でき、通常モードでは 1 または 2、スマートカードモードでは 0.5 または 1.5 にできます。

1. **0.5 個のストップビット (スマートカードモードでの受信)** : 0.5 個のストップビットでは、サンプリングは行われません。したがって、0.5 個のストップビットが選択されている場合、フレーミングエラーやブレイクフレームは検出されません。
2. **1 個のストップビット** : ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
3. **1.5 個のストップビット (スマートカードモード)** : スマートカードモードでの送信時は、デバイスは、データが正しく送信されたことをチェックする必要があります。したがって、レシーバブロックを有効にし (USART_CR1 レジスタの RE=1)、ストップビットをチェックして、スマートカードがパリティエラーを検出したかどうかをテストする必要があります。パリティエラーが発生した場合、スマートカードはサンプリング時のデータ信号を強制的にローレベルにします (これは、フレーミングエラーとしてフラグされる NACK 信号です)。その後、1.5 個のストップビットの最後に、RXNE ビットとともに FE フラグがセットされます。ストップビット 1.5 個のサンプリングは、16 番目、17 番目、および 18 番目のサンプルで (ストップビットの開始から 1 ボ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 19.3.11 : 531 ページのスマートカード](#)を参照してください。
4. **2 個のストップビット** : ストップビット 2 個のサンプリングは、最初のストップビットの 8 番目、9 番目、および 10 番目のサンプルで行われます。最初のストップビットでフレーミングエラーが検出された場合、フレーミングエラーフラグがセットされます。2 番目のストップビットでは、フレーミングエラーの検査は行われません。RXNE フラグは、最初のストップビットの最後でセットされます。

19.3.4 フラクショナルボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートには、USARTDIV の整数部と小数部にプログラミングされたものと同じ値が設定されます。

式 1 : 標準 USART のボーレート (SPI モードを含む)

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

式 2 : スマートカード、LIN、および IrDA モードのボーレート

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{16 \times \text{USARTDIV}}$$

USARTDIV は、符号なしの固定小数点数であり、USART_BRR レジスタでコード化されます。

- OVER8=0 のとき、小数部は 4 ビットでコード化され、USART_BRR レジスタの DIV_fraction[3:0] ビットによってプログラミングされます。
- OVER8=1 のとき、小数部は 3 ビットでコード化され、USART_BRR レジスタの DIV_fraction[2:0] ビットによってプログラミングされます。ビット DIV_fraction[3] はクリアされたままで必要があります。

注 : ボーカウンタは、USART_BRR への書込み後、ボーレジスタの新しい値によって更新されます。したがって、通信中はボーレートレジスタの値を変更しないでください。

OVER8=0 のとき、USART_BRR レジスタの値から USARTDIV を得る方法

例 1 :

DIV_Mantissa = 0d27 かつ DIV_Fraction = 0d12 (USART_BRR = 0x1BC) の場合、

整数部 (USARTDIV) = 0d27

小数部 (USARTDIV) = $12/16 = 0d0.75$

したがって、USARTDIV = 0d27.75

例 2 :

USARTDIV = 0d25.62 をプログラミングするには

次のようにします。

$DIV_Fraction = 16 * 0d0.62 = 0d9.92$

最も近い実数は 0d10 = 0xA

DIV_Mantissa = 整数部 (0d25.620) = 0d25 = 0x19

USART_BRR = 0x19A であるため、USARTDIV = 0d25.625

例 3 :

USARTDIV = 0d50.99 をプログラミングするには

次のようにします。

$DIV_Fraction = 16 * 0d0.99 = 0d15.84$

最も近い実数は 0d16 = 0x10 => DIV_frac[3:0] のオーバーフロー => キャリーを整数部に加算する必要があります。

DIV_Mantissa = 整数部 (0d50.990 + キャリー) = 0d51 = 0x33

USART_BRR = 0x330 であるため、USARTDIV = 0d51.000

OVER8=1 のとき、USART_BRR レジスタの値から USARTDIV を得る方法

例 1 :

DIV_Mantissa = 0x27 かつ DIV_Fraction[2:0] = 0d6 (USART_BRR = 0x1B6) の場合、

整数部 (USARTDIV) = 0d27

小数部 (USARTDIV) = $6/8 = 0d0.75$

したがって、USARTDIV = 0d27.75

例 2 :

USARTDIV = 0d25.62 をプログラミングするには

次のようにします。

$DIV_Fraction = 8 * 0d0.62 = 0d4.96$

最も近い実数は 0d5 = 0x5

DIV_Mantissa = 整数部 (0d25.620) = 0d25 = 0x19

これによって、USART_BRR = 0x195 => USARTDIV = 0d25.625

例 3 :

USARTDIV = 0d50.99 をプログラミングするには

次のようにします。

DIV_Fraction = 8*0d0.99 = 0d7.92

最も近い実数は 0d8 = 0x8 => DIV_frac[2:0] のオーバーフロー => キャリーを整数部に加算する必要があります。

DIV_Mantissa = 整数部 (0d50.990 + キャリー) = 0d51 = 0x33

これによって、USART_BRR = 0x0330 => USARTDIV = 0d51.000

表 74. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 12 \text{ MHz}$)、16 倍のオーバーサンプリング⁽¹⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート 7		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 12 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 = (計算値 - 目標値) B レート / 目標の B レート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 KBps	1.2 KBps	416.6875	0	1.2 KBps	625	0
2	2.4 KBps	2.4 KBps	208.3125	0.01	2.4 KBps	312.5	0
3	9.6 KBps	9.604 KBps	52.0625	0.04	9.6 KBps	78.125	0
4	19.2 KBps	19.185 KBps	26.0625	0.08	19.2 KBps	39.0625	0
5	38.4 KBps	38.462 KBps	13	0.16	38.339 KBps	19.5625	0.16
6	57.6 KBps	57.554 KBps	8.6875	0.08	57.692 KBps	13	0.16
7	115.2 KBps	115.942 KBps	4.3125	0.64	115.385 KBps	6.5	0.16
8	230.4 KBps	228.571 KBps	2.1875	0.79	230.769 KBps	3.25	0.16
9	460.8 KBps	470.588 KBps	1.0625	2.12	461.538 KBps	1.625	0.16
10	921.6 KBps	NA	NA	NA	NA	NA	NA
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 75. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 12 \text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 12 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 = (計算値 - 目標値) B レート / 目標のB レート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 KBps	1.2 KBps	833.375	0	1.2 KBps	1250	0
2	2.4 KBps	2.4 KBps	416.625	0.01	2.4 KBps	625	0
3	9.6 KBps	9.604 KBps	104.125	0.04	9.6 KBps	156.25	0
4	19.2 KBps	19.185 KBps	52.125	0.08	19.2 KBps	78.125	0
5	38.4 KBps	38.462 KBps	26	0.16	38.339 KBps	39.125	0.16
6	57.6 KBps	57.554 KBps	17.375	0.08	57.692 KBps	26	0.16
7	115.2 KBps	115.942 KBps	8.625	0.64	115.385 KBps	13	0.16
8	230.4 KBps	228.571 KBps	4.375	0.79	230.769 KBps	6.5	0.16
9	460.8 KBps	470.588 KBps	2.125	2.12	461.538 KBps	3.25	0.16
10	921.6 KBps	888.889 KBps	1.125	3.55	923.077 KBps	1.625	0.16
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 76. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16 \text{ MHz}$ または $f_{PCLK} = 24 \text{ MHz}$)、16 倍のオーバーサンプリング⁽¹⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 16 \text{ MHz}$			$f_{PCLK} = 24 \text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 = (計算値 - 目標値) B レート / 目標のB レート	実際	ボーレート レジスタに プログラミング されている値	誤差
1	1.2 KBps	1.2 KBps	833.3125	0	1.2	1250	0
2	2.4 KBps	2.4 KBps	416.6875	0	2.4	625	0
3	9.6 KBps	9.598 KBps	104.1875	0.02	9.6	156.25	0
4	19.2 KBps	19.208 KBps	52.0625	0.04	19.2	78.125	0
5	38.4 KBps	38.369 KBps	26.0625	0.08	38.4	39.0625	0
6	57.6 KBps	57.554 KBps	17.375	0.08	57.554	26.0625	0.08
7	115.2 KBps	115.108 KBps	8.6875	0.08	115.385	13	0.16
8	230.4 KBps	231.884 KBps	4.3125	0.64	230.769	6.5	0.16
9	460.8 KBps	457.143 KBps	2.1875	0.79	461.538	3.25	0.16

表 76. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16 \text{ MHz}$ または $f_{PCLK} = 24 \text{ MHz}$)、16 倍のオーバーサンプリング⁽¹⁾ (続き)

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 16 \text{ MHz}$			$f_{PCLK} = 24 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) B レート / 目標のB レート	実際	ボーレートレジスタにプログラミングされている値	誤差
10	921.6 Kbps	941.176 Kbps	1.0625	2.12	923.077	1.625	0.16
11	2 Mbps	NA	NA	NA	NA	NA	NA
12	3 Mbps	NA	NA	NA	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 77. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 16 \text{ MHz}$ または $f_{PCLK} = 24 \text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 16 \text{ MHz}$			$f_{PCLK} = 24 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) B レート / 目標のB レート	実際	ボーレートレジスタにプログラミングされている値	誤差
1	1.2 Kbps	1.2 Kbps	1666.625	0	1.2 Kbps	2500	0
2	2.4 Kbps	2.4 Kbps	833.375	0	2.4 Kbps	1250	0
3	9.6 Kbps	9.598 Kbps	208.375	0.02	9.6 Kbps	312.5	0
4	19.2 Kbps	19.208 Kbps	104.125	0.04	19.2 Kbps	156.25	0
5	38.4 Kbps	38.369 Kbps	52.125	0.08	38.4 Kbps	78.125	0
6	57.6 Kbps	57.554 Kbps	34.75	0.08	57.554 Kbps	52.125	0.08
7	115.2 Kbps	115.108 Kbps	17.375	0.08	115.385 Kbps	26	0.16
8	230.4 Kbps	231.884 Kbps	8.625	0.64	230.769 Kbps	13	0.16
9	460.8 Kbps	457.143 Kbps	4.375	0.79	461.538 Kbps	6.5	0.16
10	921.6 Kbps	941.176 Kbps	2.125	2.12	923.077 Kbps	3.25	0.16
11	2 Mbps	2000 Kbps	1	0	2000 Kbps	1.5	0
12	3 Mbps	NA	NA	NA	3000 Kbps	1	0

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 78. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 16 \text{ MHz}$)、16 倍のオーバーサンプリング⁽¹⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 16 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
1.	2.4 KBps	2.400 KBps	208.3125	0.00%	2.400 KBps	416.6875	0.00%
2.	9.6 KBps	9.604 KBps	52.0625	0.04%	9.598 KBps	104.1875	0.02%
3.	19.2 KBps	19.185 KBps	26.0625	0.08%	19.208 KBps	52.0625	0.04%
4.	57.6 KBps	57.554 KBps	8.6875	0.08%	57.554 KBps	17.3750	0.08%
5.	115.2 KBps	115.942 KBps	4.3125	0.64%	115.108 KBps	8.6875	0.08%
6.	230.4 KBps	228.571 KBps	2.1875	0.79%	231.884 KBps	4.3125	0.64%
7.	460.8 KBps	470.588 KBps	1.0625	2.12%	457.143 KBps	2.1875	0.79%
8.	896 KBps	NA	NA	NA	888.889 KBps	1.1250	0.79%
9.	921.6 KBps	NA	NA	NA	941.176 KBps	1.0625	2.12%
10.	1.792 MBps	NA	NA	NA	NA	NA	NA
11.	1.8432 MBps	NA	NA	NA	NA	NA	NA
12.	3.584 MBps	NA	NA	NA	NA	NA	NA
13.	3.6864 MBps	NA	NA	NA	NA	NA	NA
14.	7.168 MBps	NA	NA	NA	NA	NA	NA
15.	7.3728 MBps	NA	NA	NA	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 79. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 16 \text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 16 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
1.	2.4 KBps	2.400 KBps	416.625	0.01%	2.400 KBps	833.375	0.00%
2.	9.6 KBps	9.604 KBps	104.125	0.04%	9.598 KBps	208.375	0.02%
3.	19.2 KBps	19.185 KBps	52.125	0.08%	19.208 KBps	104.125	0.04%
4.	57.6 KBps	57.557 KBps	17.375	0.08%	57.554 KBps	34.750	0.08%
5.	115.2 KBps	115.942 KBps	8.625	0.64%	115.108 KBps	17.375	0.08%
6.	230.4 KBps	228.571 KBps	4.375	0.79%	231.884 KBps	8.625	0.64%
7.	460.8 KBps	470.588 KBps	2.125	2.12%	457.143 KBps	4.375	0.79%

表 79. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 8 \text{ MHz}$ または $f_{PCLK} = 16 \text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾ (続き)

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 8 \text{ MHz}$			$f_{PCLK} = 16 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
8.	896 Kbps	888.889 Kbps	1.125	0.79%	888.889 Kbps	2.250	0.79%
9.	921.6 Kbps	888.889 Kbps	1.125	3.55%	941.176 Kbps	2.125	2.12%
10.	1.792 MBps	NA	NA	NA	1.7777 MBps	1.125	0.79%
11.	1.8432 MBps	NA	NA	NA	1.7777 MBps	1.125	3.55%
12.	3.584 MBps	NA	NA	NA	NA	NA	NA
13.	3.6864 MBps	NA	NA	NA	NA	NA	NA
14.	7.168 MBps	NA	NA	NA	NA	NA	NA
15.	7.3728 MBps	NA	NA	NA	NA	NA	NA

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。

表 80. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30 \text{ MHz}$ または $f_{PCLK} = 60 \text{ MHz}$)、16 倍のオーバーサンプリング⁽¹⁾⁽²⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 30 \text{ MHz}$			$f_{PCLK} = 60 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
1.	2.4 Kbps	2.400 Kbps	781.2500	0.00%	2.400 Kbps	1562.5000	0.00%
2.	9.6 Kbps	9.600 Kbps	195.3125	0.00%	9.600 Kbps	390.6250	0.00%
3.	19.2 Kbps	19.194 Kbps	97.6875	0.03%	19.200 Kbps	195.3125	0.00%
4.	57.6 Kbps	57.582 Kbps	32.5625	0.03%	57.582 Kbps	65.1250	0.03%
5.	115.2 Kbps	115.385 Kbps	16.2500	0.16%	115.163 Kbps	32.5625	0.03%
6.	230.4 Kbps	230.769 Kbps	8.1250	0.16%	230.769 Kbps	16.2500	0.16%
7.	460.8 Kbps	461.538 Kbps	4.0625	0.16%	461.538 Kbps	8.1250	0.16%
8.	896 Kbps	909.091 Kbps	2.0625	1.46%	895.522 Kbps	4.1875	0.05%
9.	921.6 Kbps	909.091 Kbps	2.0625	1.36%	923.077 Kbps	4.0625	0.16%
10.	1.792 MBps	1.1764 MBps	1.0625	1.52%	1.8182 MBps	2.0625	1.36%
11.	1.8432 MBps	1.8750 MBps	1.0000	1.73%	1.8182 MBps	2.0625	1.52%
12.	3.584 MBps	NA	NA	NA	3.2594 MBps	1.0625	1.52%
13.	3.6864 MBps	NA	NA	NA	3.7500 MBps	1.0000	1.73%

表 80. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30\text{ MHz}$ または $f_{PCLK} = 60\text{ MHz}$)、16 倍のオーバーサンプリング⁽¹⁾⁽²⁾ (続き)

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 30\text{ MHz}$			$f_{PCLK} = 60\text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
14.	7.168 MBps	NA	NA	NA	NA	NA	NA
15.	7.3728 MBps	NA	NA	NA	NA	NA	NA

- CPU クロック周波数が低いほど、ボーレートの精度も低下します。ボーレートの達成可能な上限は、これらのデータによって決定できます。
- USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

表 81. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 30\text{ MHz}$ または $f_{PCLK} = 60\text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾⁽²⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 30\text{ MHz}$			$f_{PCLK} = 60\text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
1.	2.4 KBps	2.400 KBps	1562.5000	0.00%	2.400 KBps	3125.0000	0.00%
2.	9.6 KBps	9.600 KBps	390.6250	0.00%	9.600 KBps	781.2500	0.00%
3.	19.2 KBps	19.194 KBps	195.3750	0.03%	19.200 KBps	390.6250	0.00%
4.	57.6 KBps	57.582 KBps	65.1250	0.16%	57.582 KBps	130.2500	0.03%
5.	115.2 KBps	115.385 KBps	32.5000	0.16%	115.163 KBps	65.1250	0.03%
6.	230.4 KBps	230.769 KBps	16.2500	0.16%	230.769 KBps	32.5000	0.16%
7.	460.8 KBps	461.538 KBps	8.1250	0.16%	461.538 KBps	16.2500	0.16%
8.	896 KBps	909.091 KBps	4.1250	1.46%	895.522 KBps	8.3750	0.05%
9.	921.6 KBps	909.091 KBps	4.1250	1.36%	923.077 KBps	8.1250	0.16%
10.	1.792 MBps	1.7647 MBps	2.1250	1.52%	1.8182 MBps	4.1250	1.46%
11.	1.8432 MBps	1.8750 MBps	2.0000	1.73%	1.8182 MBps	4.1250	1.36%
12.	3.584 MBps	3.7500 MBps	1.0000	4.63%	3.5294 MBps	2.1250	1.52%
13.	3.6864 MBps	3.7500 MBps	1.0000	1.73%	3.7500 MBps	2.0000	1.73%
14.	7.168 MBps	NA	NA	NA	7.5000 MBps	1.0000	4.63%
15.	7.3728 MBps	NA	NA	NA	7.5000 MBps	1.0000	1.73%

- CPU クロック周波数が低いほど、ボーレートの精度も低下します。ボーレートの達成可能な上限は、これらのデータによって決定できます。
- USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

表 82. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42\text{MHz}$ または $f_{PCLK} = 84\text{MHz}$)、16 倍のオーバーサンプリング⁽¹⁾⁽²⁾

16 倍のオーバーサンプリング (OVER8=0)							
ボーレート		$f_{PCLK} = 42 \text{ MHz}$			$f_{PCLK} = 84 \text{ MHz}$		
S.No	目標	実際	ボーレートレジスタにプログラミングされている値	誤差 = (計算値 - 目標値) / 目標ボーレート	実際	ボーレートレジスタにプログラミングされている値	誤差
1	1.2 Kbps	1.2 Kbps	2187.5	0	1.2 Kbps	4375	0
2	2.4 Kbps	2.4 Kbps	1093.75	0	2.4 Kbps	2187.5	0
3	9.6 Kbps	9.6 Kbps	273.4375	0	9.6 Kbps	546.875	0
4	19.2 Kbps	19.195 Kbps	136.75	0.02	19.2 Kbps	273.4375	0
5	38.4 Kbps	38.391 Kbps	68.375	0.02	38.391 Kbps	136.75	0.02
6	57.6 Kbps	57.613 Kbps	45.5625	0.02	57.613 Kbps	91.125	0.02
7	115.2 Kbps	115.068 Kbps	22.8125	0.11	115.226 Kbps	45.5625	0.02
8	230.4 Kbps	230.769 Kbps	11.375	0.16	230.137 Kbps	22.8125	0.11
9	460.8 Kbps	461.538 Kbps	5.6875	0.16	461.538 Kbps	11.375	0.16
10	921.6 Kbps	913.043 Kbps	2.875	0.93	923.076 Kbps	5.6875	0.93
11	1.792 MBps	1.826 MBps	1.4375	1.9	1.787 MBps	2.9375	0.27
12	1.8432 MBps	1.826 MBps	1.4375	0.93	1.826 MBps	2.875	0.93
13	3.584 MBps	N/A	N/A	N/A	3.652 MBps	1.4375	1.9
14	3.6864 MBps	N/A	N/A	N/A	3.652 MBps	1.4375	0.93
15	7.168 MBps	N/A	N/A	N/A	N/A	N/A	N/A
16	7.3728 MBps	N/A	N/A	N/A	N/A	N/A	N/A
18	9 MBps	N/A	N/A	N/A	N/A	N/A	N/A
20	10.5 MBps	N/A	N/A	N/A	N/A	N/A	N/A

- CPU クロック周波数が低いほど、ボーレートの精度も低下します。ボーレートの達成可能な上限は、これらのデータによって決定できます。
- USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

表 83. プログラミングされたボーレートの誤差計算 ($f_{PCLK} = 42\text{MHz}$ または $f_{PCLK} = 84\text{ MHz}$)、8 倍のオーバーサンプリング⁽¹⁾⁽²⁾

8 倍のオーバーサンプリング (OVER8=1)							
ボーレート		$f_{PCLK} = 42\text{ MHz}$			$f_{PCLK} = 84\text{ MHz}$		
S.No	目標	実際	ボーレート レジスタに プログラミング されている値	誤差 = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	ボーレート レジスタに プログラミング されている値	誤差
1.	1.2 KBps	1.2 KBps	4375	0	1.2 KBps	8750	0
2.	2.4 KBps	2.4 KBps	2187.5	0	2.4 KBps	4375	0
3.	9.6 KBps	9.6 KBps	546.875	0	9.6 KBps	1093.75	0
4.	19.2 KBps	19.195 KBps	273.5	0.02	19.2 KBps	546.875	0
5.	38.4 KBps	38.391 KBps	136.75	0.02	38.391 KBps	273.5	0.02
6.	57.6 KBps	57.613 KBps	91.125	0.02	57.613 KBps	182.25	0.02
7.	115.2 KBps	115.068 KBps	45.625	0.11	115.226 KBps	91.125	0.02
8.	230.4 KBps	230.769 KBps	22.75	0.11	230.137 KBps	45.625	0.11
9.	460.8 KBps	461.538 KBps	11.375	0.16	461.538 KBps	22.75	0.16
10.	921.6 KBps	913.043 KBps	5.75	0.93	923.076 KBps	11.375	0.93
11.	1.792 MBps	1.826 MBps	2.875	1.9	1.787Mbps	5.875	0.27
12.	1.8432 MBps	1.826 MBps	2.875	0.93	1.826 MBps	5.75	0.93
13.	3.584 MBps	3.5 MBps	1.5	2.34	3.652 MBps	2.875	1.9
14.	3.6864 MBps	3.82 MBps	1.375	3.57	3.652 MBps	2.875	0.93
15.	7.168 MBps	N/A	N/A	N/A	7 MBps	1.5	2.34
16.	7.3728 MBps	N/A	N/A	N/A	7.636 MBps	1.375	3.57
18.	9 MBps	N/A	N/A	N/A	9.333 MBps	1.125	3.7
20.	10.5 MBps	N/A	N/A	N/A	10.5 MBps	1	0

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。
2. USART1 および USART6 だけが PCLK2 でクロック供給されます。他の USART は PCLK1 でクロック供給されます。PCLK1 および PCLK2 の最大値については、デバイスのデータシートを参照してください。

19.3.5 クロック偏差に対する USART レシーバの許容誤差

USART の非同期レシーバが正しく機能するのは、クロックシステムの合計偏差が USART レシーバの許容誤差を下回る場合に限りです。合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタ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)

$$DTRA + DQUANT + DREC + DTCL < \text{USART レシーバの許容誤差}$$

データを正しく受信するための USART レシーバの許容誤差は、許容される最大偏差と等しく、以下の選択に依存します。

- USART_CR1 レジスタの M ビットによって定義される 10 または 11 ビットのキャラクタ長
- USART_CR1 レジスタの OVER8 ビットによって定義された 8 倍または 16 倍のオーバーサンプリング
- フラクショナルボーレートを使用するかどうか
- データのサンプリングに 1 ビットを使用するか 3 ビットを使用するか (USART_CR3 レジスタの ONEBIT ビットの値に依存)。

表 84. USART レシーバの許容誤差 (DIV_Fraction が 0 の場合)

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.75%	4.375%	2.50%	3.75%
1	3.41%	3.97%	2.27%	3.41%

表 85. USART レシーバ許容誤差 (DIV_Fraction が 0 でない場合)

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.33%	3.88%	2%	3%
1	3.03%	3.53%	1.82%	2.73%

注：表 84 と表 85 に示されている数値は、受信したフレームが正確に 10 ビット時間 (M=0 の場合) または 11 ビット時間 (M=1 の場合) のアイドルフレームを含んでいる場合、若干異なることがあります。

19.3.6 マルチプロセッサ通信

複数の USART をネットワークに接続すると、マルチプロセッサ通信を行える可能性があります。たとえば、1 つの USART をマスタとして、その TX 出力を別の USART の RX 入力に接続することができます。別の USART はスレーブであり、それぞれの TX 出力の論理積をとった上でマスタの RX 入力に接続します。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する USART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードにできます。ミュートモードでは、

- 受信ステータスビットはセットできません。
- 受信割込みはすべて禁止されます。
- USART_CR1 レジスタの RWU ビットが 1 にセットされます。RWU ビットは、ハードウェアによって自動的に制御できますが、特定の条件下ではソフトウェアによる書込みが可能です。

USART は、USART_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

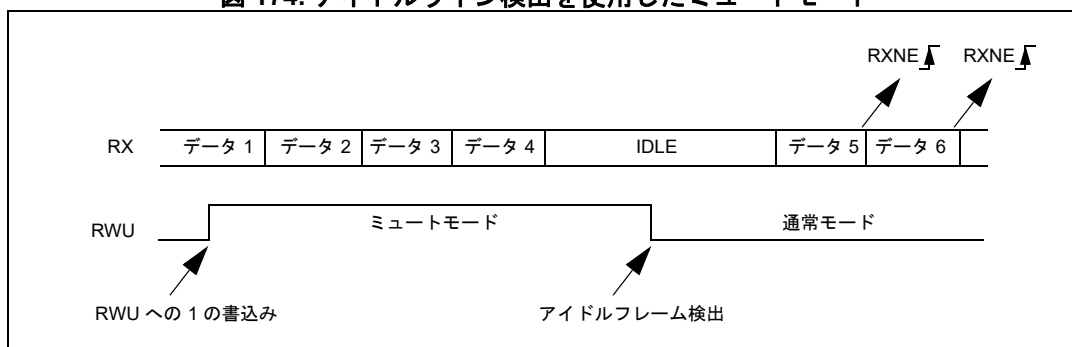
アイドルライン検出 (WAKE=0)

RWU ビットに 1 が書き込まれると、USART はミュートモードに入ります。

USART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、USART_SR レジスタの IDLE ビットはセットされません。RWU ビットには、ソフトウェアによって 0 を書き込むこともできます。

アイドルライン検出を使用したミュートモードの動作例を [図 174](#) に示します。

図 174. アイドルライン検出を使用したミュートモード



アドレスマーク検出 (WAKE=1)

このモードでは、MSB が“1”のバイトはアドレスとして、そうでないバイトはデータとして、それぞれ認識されます。アドレスバイトでは、ターゲットレシーバのアドレスは下位 4 ビットになります。レシーバは、この 4 ビットワードを、USART_CR2 レジスタの ADD ビットにプログラミングされている自分のアドレスと比較します。

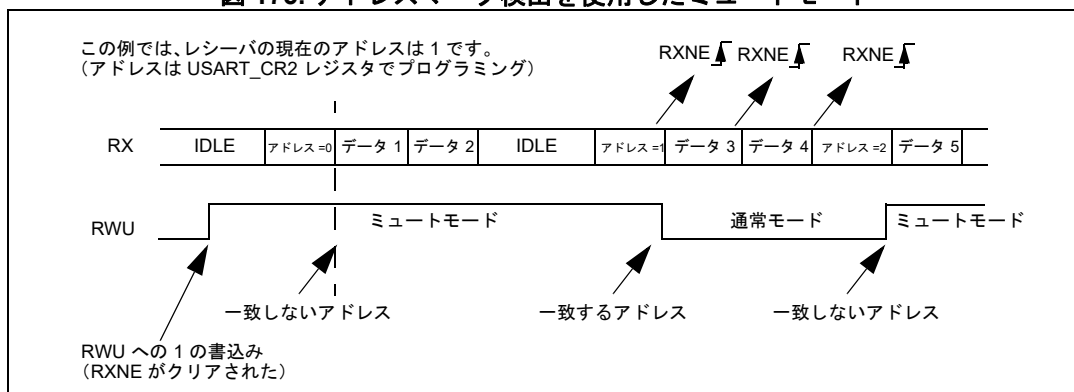
プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、USART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。USART はミュートモードに入っているため、このアドレスバイトに対して RXNE フラグはセットされず、割り込みも DMA リクエストも発行されません。

プログラミングされたアドレスと一致するアドレスキャラクタを受信すると、USART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているため、アドレスキャラクタに対して RXNE ビットがセットされます。

レシーババッファにデータが存在しない場合 (USART_SR レジスタの RXNE=0)、RWU ビットに 0 または 1 を書き込むことができます。そうでない場合、書き込みは無視されます。

アドレスマーク検出を使用したミュートモードの動作例を [図 175](#) に示します。

図 175. アドレスマーク検出を使用したミュートモード



19.3.7 パリティ制御

パリティ制御（送信中のパリティビット生成と受信中のパリティチェック）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 86 に示します。

表 86. フレームフォーマット

M ビット	PCE ビット	USART フレーム ⁽¹⁾
0	0	SB 8 ビットデータ STB
0	1	SB 7 ビットデータ PB STB
1	0	SB 9 ビットデータ STB
1	1	SB 8 ビットデータ PB STB

1. 凡例：SB：スタートビット、STB：ストップビット、PB：パリティビット。

偶数パリティ

下位 7 ビットまたは 8 ビット (M=0 または M=1 に応じて) とパリティビットから構成されるフレーム内で、「1」の数が偶数になるように、パリティビットが計算されます。

例：データ = 00110101、4 ビットがセット => 偶数パリティが選択された場合 (USART_CR1 の PS=0)、パリティビットは 0 になります。

奇数パリティ

下位 7 ビットまたは 8 ビット (M=0 または M=1 に応じて) とパリティビットから構成されるフレーム内で、「1」の数が奇数になるように、パリティビットが計算されます。

例：データ = 00110101、4 ビットがセット => 奇数パリティが選択された場合 (USART_CR1 の PS=1)、パリティビットは 1 になります。

受信中のパリティチェック

パリティチェックに失敗した場合、USART_SR レジスタの PE フラグがセットされ、USART_CR1 レジスタの PEIE ビットがセットされている場合は割込みが生成されます。PE フラグは、ソフトウェアシーケンスによってクリアされます (ステータスレジスタの読出しと、それに続く USART_DR データレジスタへの読出しまたは書き込みアクセス)。

注： アドレスマークによるウェイクアップの場合、アドレス（パリティビットではありません）の特定にはデータの最上位ビットが考慮されます。そしてレシーバは、アドレスデータのパリティをチェックしません（パリティエラーの場合、PE はセットされません）。

送信中のパリティ生成

USART_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます（偶数パリティが選択された場合（PS=0）は偶数個の「1」、奇数パリティが選択された場合（PS=1）は奇数個の「1」）。

注： 送信を管理するソフトウェアルーチンは、PE フラグをクリアするソフトウェアシーケンス（ステータスレジスタの読出しに続くデータレジスタへの読出しまたは書き込みアクセス）を起動できます。半二重モードでは、ソフトウェアによっては PE フラグが不意にクリアされることがあります。

19.3.8 LIN (Local Interconnection Network) モード

LIN モードを選択するには、USART_CR2 レジスタの LINEN ビットをセットします。LIN モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの STOP[1:0] および CLKEN ビット
- USART_CR3 レジスタの SCEN、HDSEL、および IREN ビット

LIN 送信

LIN マスタの送信では、通常の USART 送信の手順ではなく、[セクション 19.3.2](#) で説明したものと同一手順を適用する必要がありますが、次の点が異なります。

- 8 ビットのワード長を設定するには M ビットをクリアします。
- LIN モードに入るには、LINEN ビットをセットします。この場合、SBK ビットをセットすると、13 個の“0”ビットがブレークキャラクタとして送信されます。その後、値“1”のビットが送信され、次の START 検出が可能になります。

LIN 受信

USART インタフェースにはブレーク検出回路が実装されています。この検出は、通常の USART レシーバとは完全に独立しています。アイドル状態時やフレームの処理中には、発生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“0”ビットが検出され、その後にデリミタキャラクタが続く場合、USART_SR レジスタの LBD フラグがセットされます。LBDIE ビットが 1 の場合、割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。

ブレイク検出回路ステートマシンの動作とブレイクフラグを [図 176: 527 ページのLIN モードでのブレイク検出 \(11 ビットブレイク長、LBDL=1\)](#) に示します。

ブレイクフレームの例を [図 177: 528 ページのLIN モードでのブレイク検出とフレーミングエラー検出](#) に示します。

図 176. LIN モードでのブレイク検出 (11 ビットブレイク長、LBDL=1)

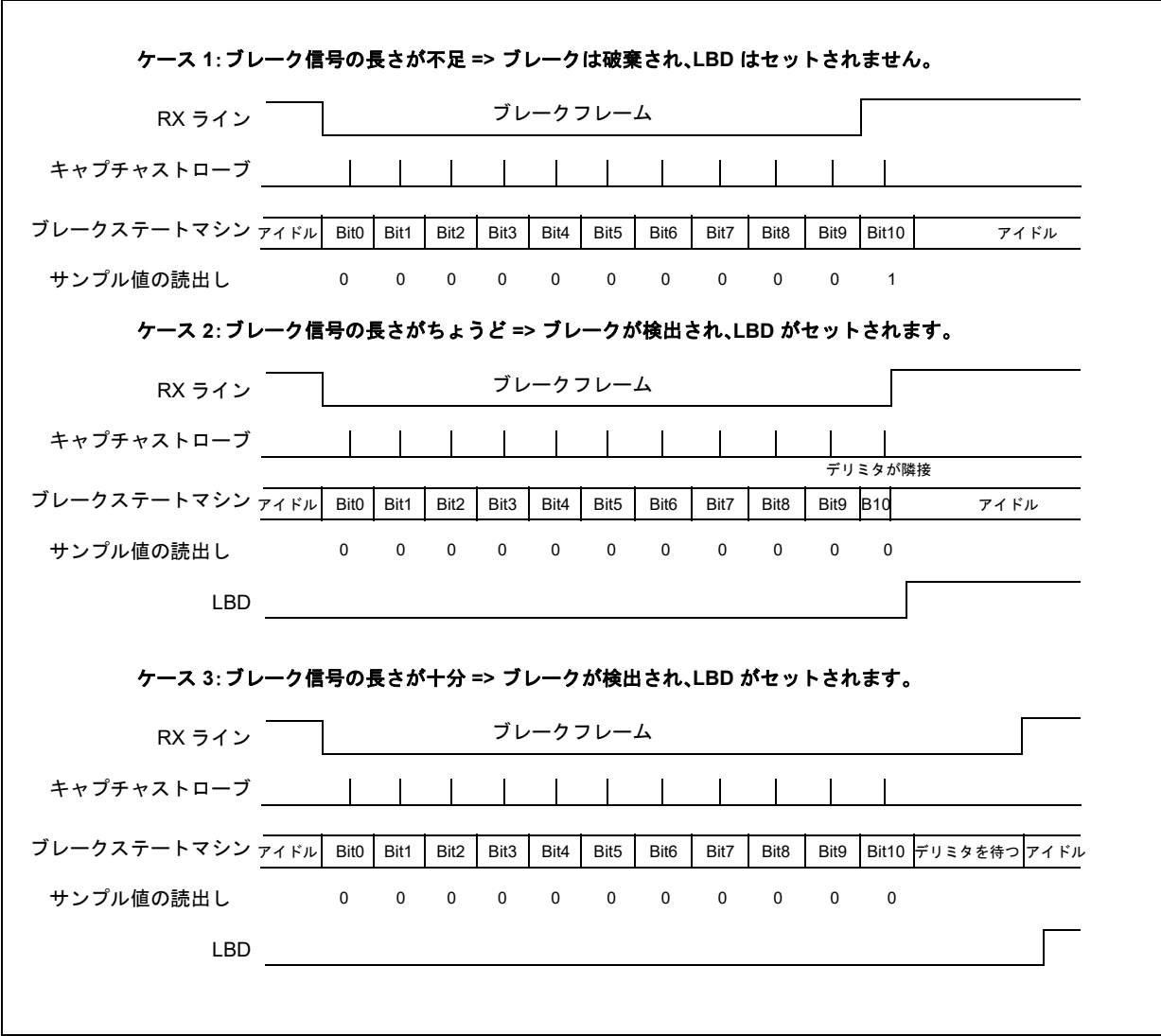
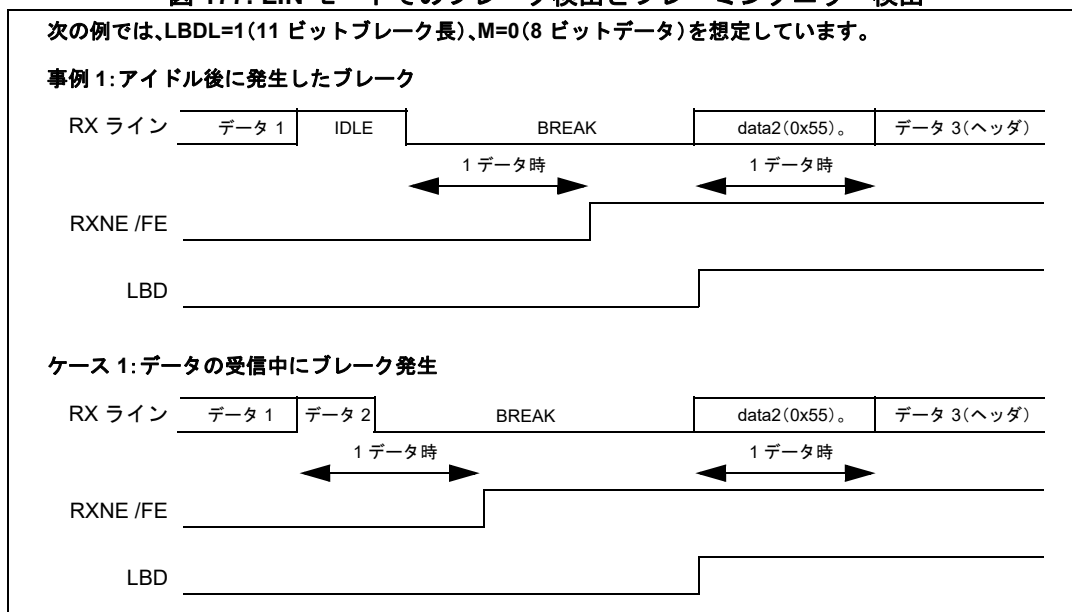


図 177. LIN モードでのブレーク検出とフレーミングエラー検出



19.3.9 USART 同期モード

同期モードを選択するには、USART_CR2 レジスタの CLKEN ビットに 1 を書き込みます。同期モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの SCEN、HDSEL、および IREN ビット

USART では、マスタモードで双方向の同期シリアル通信を制御できます。SCLK ピンは USART トランスミッタクロックの出力です。スタートビットとストップビットの処理中には、SCLK ピンにクロックパルスは送信されません。USART_CR2 レジスタの LBCL ビットの状態によっては、有効な最後のデータビット（アドレスマーク）の処理中にクロックパルスが生成されることもあります。USART_CR2 レジスタの CPOL ビットは、クロック極性を選択するために使用でき、USART_CR2 レジスタの CPHA ビットは、外部クロックの位相を選択するために使用できます（図 178、図 179、および図 180 を参照）。

アイドル時、プリアンプル処理時、およびブレーク送信時には、外部 SCLK クロックは起動されません。

同期モードでは、USART トランスミッタは非同期モードの場合とまったく同じように機能します。しかし、CPOL と CPHA に基づいて SCLK が TX と同期すると、TX 上のデータが同期します。

このモードでは、USART レシーバは非同期モードの場合とは異なる動作をします。RE=1 の場合、データはオーバーサンプリングなしで、SCLK (CPOL と CPHA に応じて立上りまたは立下りエッジ) でサンプリングされます。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： SCLK ピンは TX ピンと連携して動作します。したがって、クロックが供給されるのは、トランスミッタが有効であり (TE=1)、データが送信中 (データレジスタ USART_DR への書き込み) の場合に限られます。つまり、データ送信を行わずに同期データを受信することはできません。

トランスミッタとレシーバが無効にされた場合 (TE=RE=0)、クロックパルスが正常に機能するには、LBCL、CPOL、および CPHA の各ビットを選択する必要があります。これらのビットは、トランスミッタまたはレシーバが有効である間は、変更しないでください。

レシーバのセットアップ時間とホールド時間を最小限に抑えるため、TE ビットと RE ビットを同じ命令でセットすることをお勧めします。

USART は、マスタモードのみをサポートし、入力クロックに関連するデータを送受信することはできません (SCLK は常に出力です)。

図 178. USART の同期送信の例

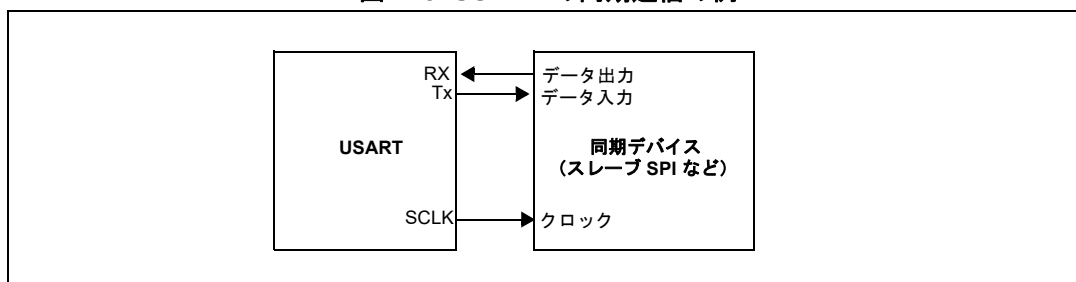


図 179. USART データクロックタイミング図 (M=0)

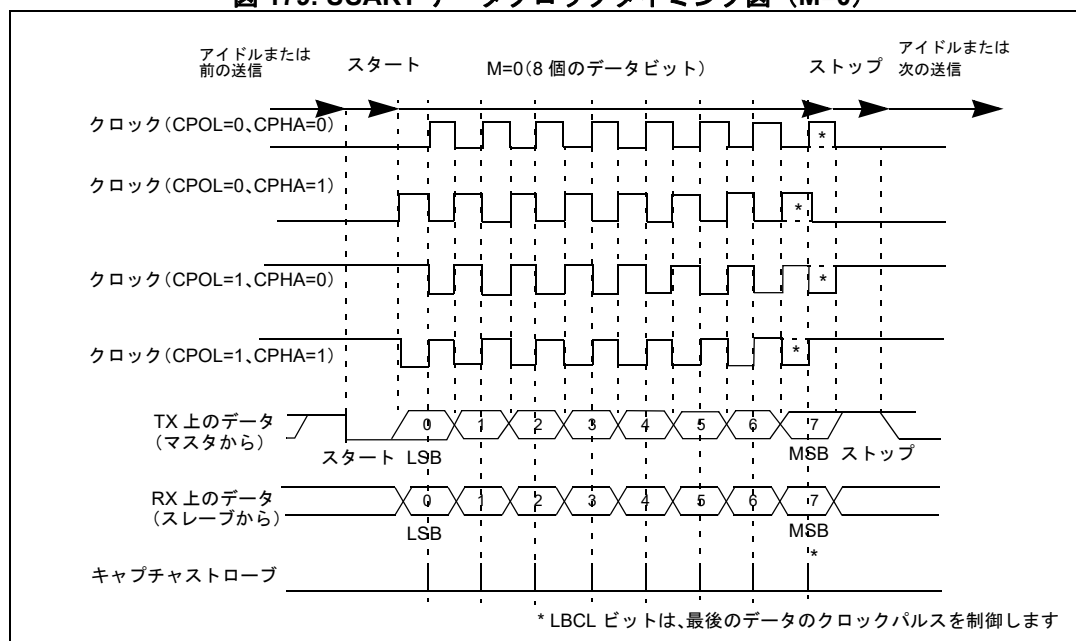
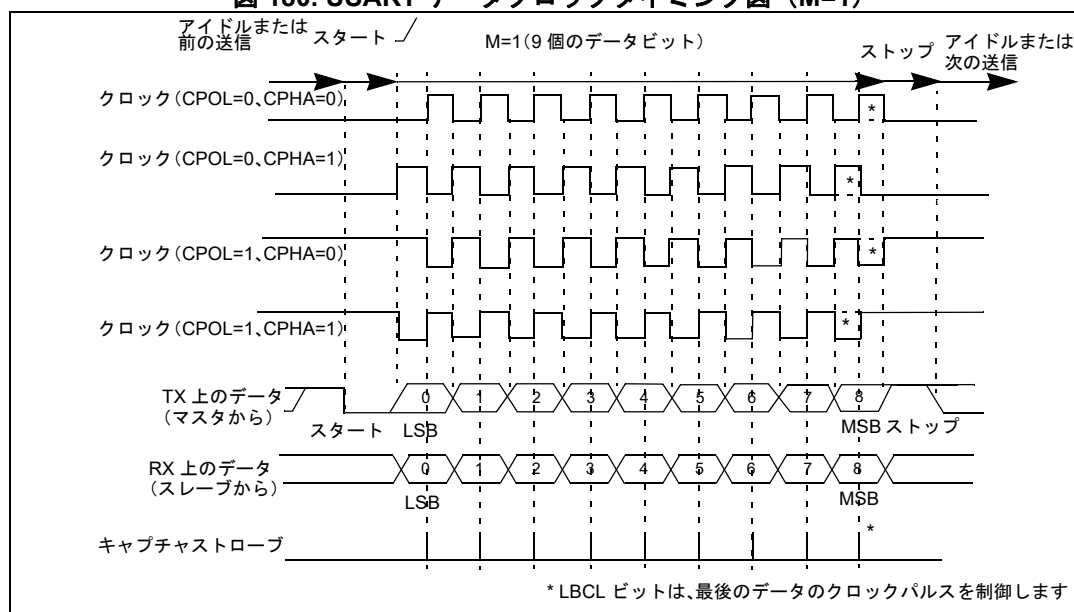
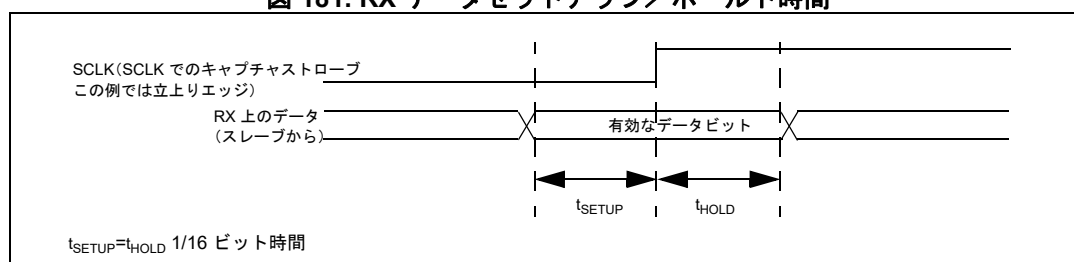


図 180. USART データクロックタイミング図 (M=1)

図 181. RX データセットアップ/ホールド時間


注：スマートカードモードでは SCLK の機能が異なります。詳細については、スマートカードモードの章を参照してください。

19.3.10 単線半二重通信

単線半二重モードを選択するには、USART_CR3 レジスタの HDSEL ビットをセットします。このモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN および CLKEN ビット
- USART_CR3 レジスタの SCEN および IREN ビット

USART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、制御ビット“HALF DUPLEX SEL” (USART_CR3 レジスタの HDSEL) で行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が USART によって駆動されないときはフローティング入力 (または出力ハイのオープンドレイン) として設定されるように、入出力を設定する必要があります。

この点を除くと、通信は通常の USART モードの場合と同様です。ラインの競合はソフトウェアによって管理する必要があります（たとえば、集中型アービタを使用）。特に、TE ビットがセットされている間は、送信はハードウェアによってブロックされることはなく、データがデータレジスタに書き込まれると送信を続行します。

19.3.11 スマートカード

スマートカードモードを選択するには、USART_CR3 レジスタの SCEN ビットをセットします。スマートカードモードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの HDSEL および IREN ビット

さらに、スマートカードにクロックを供給するために CLKEN ビットをセットすることもできます。

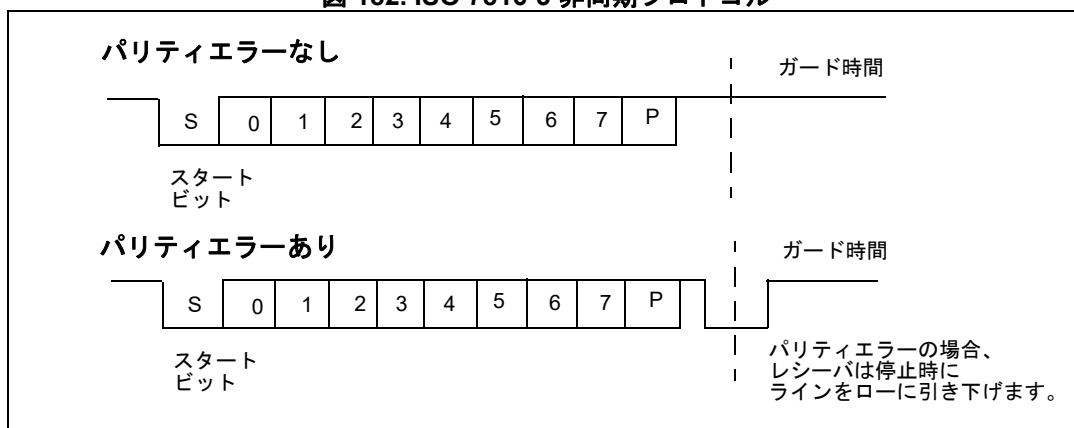
スマートカードインタフェースは、ISO 7816-3 標準で定義された非同期プロトコルスマートカードをサポートするように設計されています。USART は次のように設定してください。

- 8 ビット + パリティ : M ビット = 1 および USART_CR1 レジスタの PCE = 1
- 送受信時の 1.5 個の STOP ビット : USART_CR2 レジスタの STOP = 11

注 : 受信にはストップビット 0.5 個を選択することも可能ですが、2 つの設定間の切り替えを避けるために、送信と受信の両方にストップビット 1.5 個を使用することを推奨します。

図 182 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 182. ISO 7816-3 非同期プロトコル



スマートカードと接続されると、USART の TX 出力は、やはりスマートカードによって駆動される双方向ラインを駆動します。TX ピンは、オープンドレインとして設定される必要があります。

スマートカードは、単線半二重通信プロトコルです。

- 送信シフトレジスタからのデータの送信は、少なくとも 1/2 ボークロックの遅れが保証されます。通常動作では、満杯の送信シフトレジスタは、次のボークロックエッジでシフト動作を開始します。スマートカードモードでは、この送信は、保証された 1/2 ボークロック分だけさらに遅れます。
- ストップビット 0.5 または 1.5 周期でプログラミングされたフレームの受信時にパリティエラーが検出された場合、受信フレームの完了後 1 ボークロック期間、送信ラインがローに引き下げられます。これは、USART に送信されたデータが正しく受信されなかったことをスマートカードに知らせるためです。この NACK 信号（1 ボークロックの間、送信ラインをローに引き下げ）は、1.5 個のストップビットが組み込まれたトランスミッタ側にフレーミングエラーを引き起こします。アプリケーションは、プロトコルに基づいて、データの再送信に対処できます。

NACK 制御ビットがセットされている場合、パリティエラーはレシーバによって“NACK”されます。そうでない場合、NACK は送信されません。

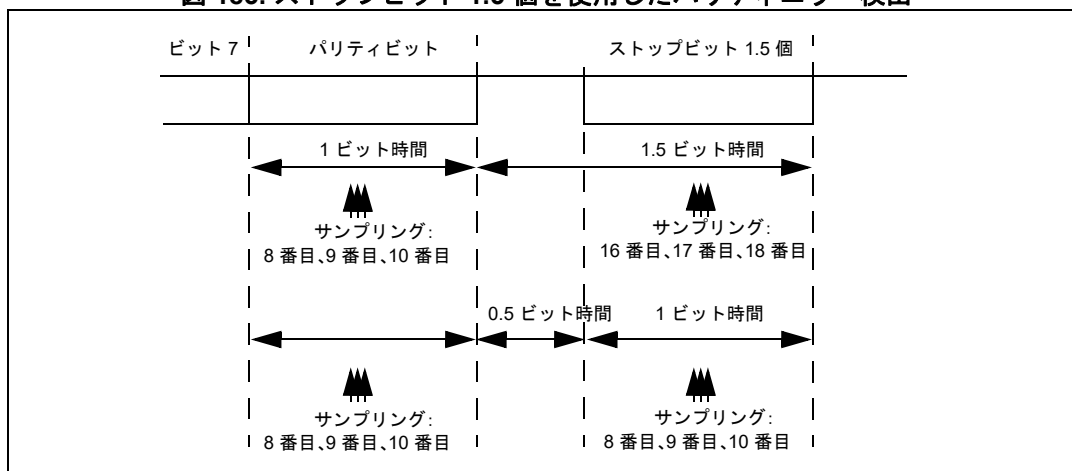
- TC フラグのアサーションは、ガードタイムレジスタをプログラムすることによって遅らせることができます。通常動作では、TC がアサートされるのは、送信シフトレジスタが空であり、他に未処理の送信リクエストがない場合です。スマートカードモードでは、空の送信シフトレジスタは、ガードタイムカウンタをトリガして、ガードタイムレジスタにプログラミングされた値までカウントアップします。この間、TC は強制的にローレベルに保たれます。ガードタイムカウンタがプログラミングされた値に達すると、TC がハイにアサートされます。
- TC フラグのネゲートは、スマートカードモードの影響を受けません。
- レシーバからの NACK によってトランスミッタ側でフレーミングエラーが検出された場合、トランスミッタの受信ブロックは、この NACK をスタートビットとして検出しません。ISO プロトコルによれば、受信される NACK の期間は 1 または 2 ボークロック周期です。
- レシーバ側では、パリティエラーが検出されて NACK が送信された場合、レシーバはこの NACK をスタートビットとして検出しません。

注： スマートカードモードでは、ブレイクキャラクタは意味を持ちません。フレーミングエラー発生時のデータ 0x00 は、ブレイクではなくデータとして処理されます。

TE ビットをトグルするとき、アイドルフレームは送信されません。アイドルフレームは、他の設定では定義されますが、ISO プロトコルでは定義されていません。

図 183 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 183. ストップビット 1.5 個を使用したパリティエラー検出



USART は、SCLK 出力を通じてスマートカードにクロックを供給できます。スマートカードモードでは、SCLK は通信に関係せず、5 ビットのプリスケアラを通じて単に内部のペリフェラル入力クロックから取得されます。この分周比は、プリスケアラレジスタ USART_GTPR で設定されます。SCLK 周波数は、 $f_{CK}/2$ から $f_{CK}/62$ までの範囲でプログラムでき、 f_{CK} はペリフェラル入力クロックです。

19.3.12 Ir SIR ENDEC ブロック

IrDA モードを選択するには、USART_CR3 レジスタの IREN ビットをセットします。IrDA モードでは、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN、STOP、および CLKEN ビット
- USART_CR3 レジスタの SCEN および HDSEL ビット

IrDA SIR 物理層は、ロジック 0 を赤外光パルスとして表現する RZI (Return to Zero, Inverted) 変調方式の使用を指定します (図 184 を参照)。

SIR 送信エンコーダは、USART からの NRZ (Non Return to Zero) 送信ビットストリーム出力を変調します。出力パルスストリームは、外部の出力ドライバと赤外線 LED に送信されます。SIR ENDEC の場合、USART は最大 115.2 Kbps のビットレートしかサポートしません。通常モードでは、送信されるパルス幅は、ビット周期の 3/16 と指定されます。

SIR 受信デコーダは、赤外線検出回路からの RZ (Return to Zero) ビットストリームを復調し、受信した NRZ シリアルビットストリームを USART に出力します。デコーダの入力は、アイドル状態のノーマルハイレベル (マーク状態) です。送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。デコーダ入力が高レベルのとき、スタートビットが検出されます。

- IrDA は半二重通信プロトコルです。トランスミッタがビジーである (つまり、USART が IrDA エンコーダにデータを送信している) 場合、IrDA デコーダは IrDA 受信ライン上にあるすべてのデータを無視します。データの受信中は、送信データの破壊を防ぐために、送信を避けてください。
- 0 はハイパルスとして送信され、1 は 0 として送信されます。通常モードでは、パルスの幅は、選択されたビット周期の 3/16 と規定されます (図 185 を参照)。
- SIR デコーダは、IrDA 準拠の受信信号を USART 用のビットストリームに変換します。
- SIR 受信ロジックは、ハイ状態を論理値 1 とみなし、ローパルスを論理値 0 とみなします。
- 送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。SIR 出力は、アイドル時にロー状態になります。
- IrDA 仕様では、1.41 us より大きなパルスを受け入れる必要があります。受け入れられるパルス幅は、プログラム可能です。レシーバ端のグリッチ検出回路は、PSC 2 周期 (PSC は IrDA 低電力ポーレジスタ USART_GTPR にプログラミングされたプリスケール値) より小さな幅のパルスをフィルタします。PSC 1 周期より小さな幅のパルスは常に拒否されますが、1 周期以上 2 周期未満の幅のパルスは受け入れられることも、拒否されることもあります。2 周期より大きな幅のパルスは、パルスとして受け入れられます。PSC=0 のとき、IrDA エンコーダ/デコーダは機能しません。
- レシーバは、低電力トランスミッタと通信できます。
- IrDA モードでは、USART_CR2 レジスタのストップビットを「1 ストップビット」に設定する必要があります。

IrDA 低電力モード

トランスミッタ :

低電力モードでは、パルス幅はビット周期の 3/16 に維持されません。代わりに、パルス幅は低電力ボーレート（最小で 1.42 MHz）の 3 倍となります。一般に、この値は 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz) です。低電力モードのプログラム可能な分周器は、この値を得るためにシステムクロックを分周します。

レシーバ :

低電力モードでの受信は、通常モードでの受信と同様です。グリッチ検出の場合、USART は 1/PSC よりも短いパルスを破棄する必要があります。有効なローレベルは、その期間が IrDA 低電力ボーレート (USART_GTPR の PSC 値) の 2 周期分を超える場合にのみ受け入れられます。

注 : PSC 2 周期未満 1 周期以上の幅のパルスは、拒否されることも、拒否されないこともあります。
レシーバのセットアップ時間は、ソフトウェアで管理してください。IrDA 物理層仕様では、送信と受信の間に最小 10 ms の遅延を指定しています (IrDA は半二重プロトコルです)。

図 184. IrDA SIR ENDEC- ブロック図

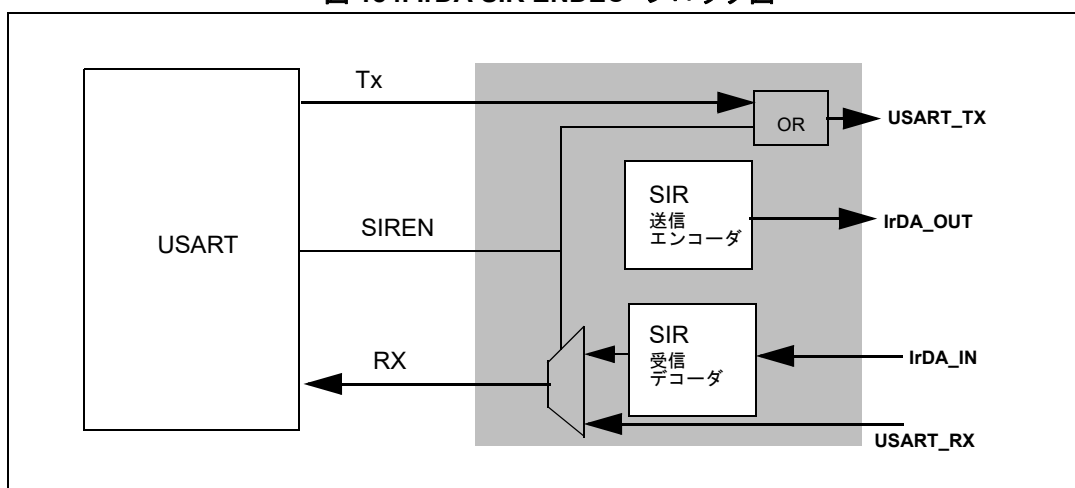
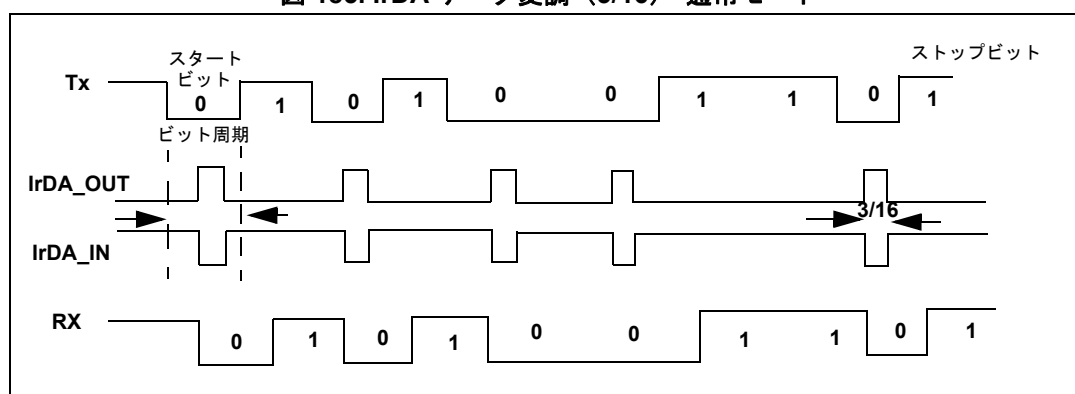


図 185. IrDA データ変調 (3/16) - 通常モード



19.3.13 DMA を使用した連続通信

USART は、DMA を使用して通信を継続することができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

DMA を使用した送信

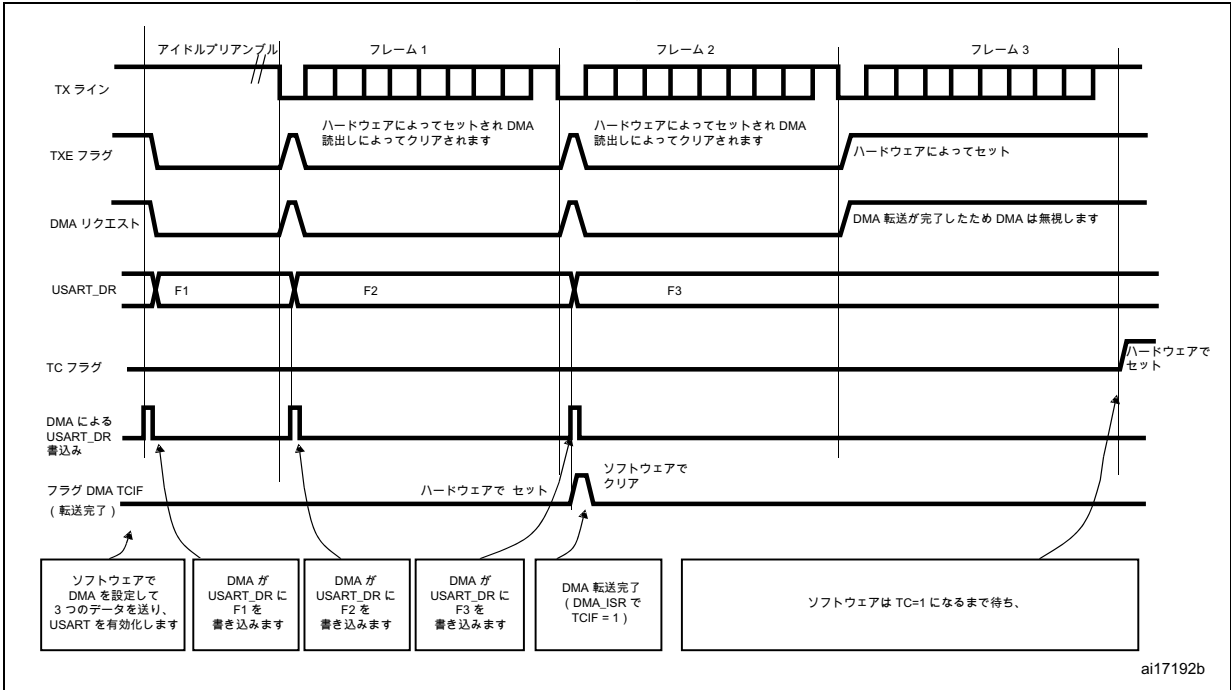
DMA モードでの送信を有効にするには、USART_CR3 レジスタの DMAT ビットをセットします。TXE ビットがセットされるたびに、データは、DMA ペリフェラル（DMA 仕様を参照）を使用して設定された SRAM 領域から USART_DR レジスタにロードされます。DMA チャンネルを USART 送信用に割り付けるには、次の手順を実行します（x はチャンネル番号を示します）

1. DMA 制御レジスタに USART_DR レジスタのアドレスを書込み、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込み、これを転送元として設定します。データは、各 TXE イベント後に、このメモリ領域から USART_DR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. SR レジスタの TC ビットに 0 を書き込んでクリアします。
7. DMA レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと（DMA_ISR レジスタの TCIF フラグがセットされます）、TC フラグを観察して USART 通信の完了を確認することができます。これは、USART を無効にしたり STOP モードに入ったりする前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、データ転送中はクリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

図 186. DMA を使用した送信



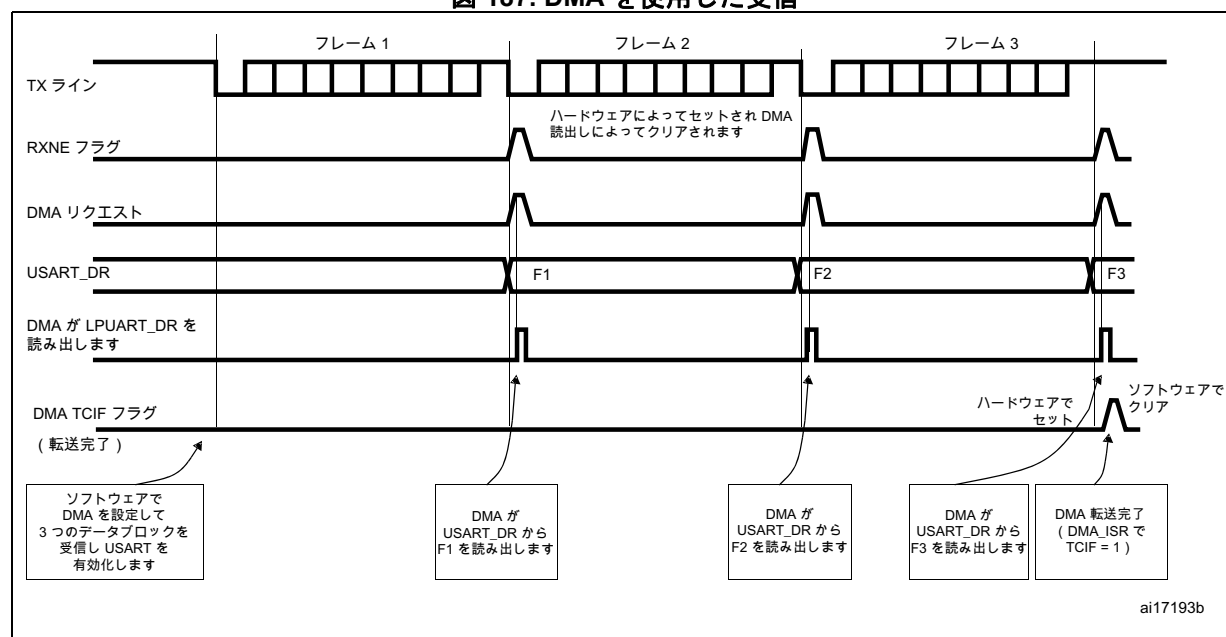
DMA を使用した受信

DMA モードでの受信を有効にするには、USART_CR3 レジスタの DMAR ビットをセットします。データは、データバイトが受信されるたびに、USART_DR レジスタから DMA ペリフェラル（DMA 仕様を参照）を使用して設定された SRAM 領域にロードされます。DMA チャンネルを USART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに USART_DR レジスタのアドレスを書込み、これを転送元として設定します。データは、各 RXNE イベントの後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書込み、これを転送先として設定します。データは、各 RXNE イベントの後に、USART_DR レジスタからこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割り込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。割り込み処理サブルーチンでは、USART_CR3 レジスタの DMAR ビットをソフトウェアでクリアする必要があります。

図 187. DMA を使用した受信



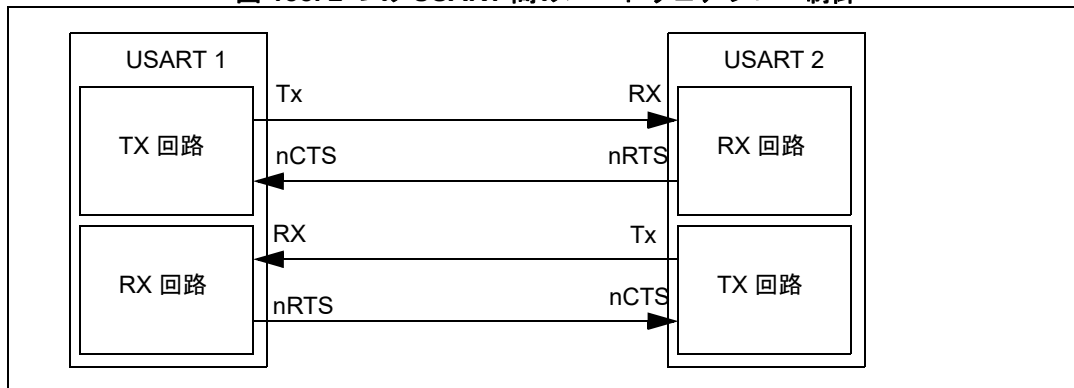
マルチバッファ通信における割り込み生成とエラーフラグ

マルチバッファ通信でトランザクションの処理中にエラーが発生した場合、現在のバイトの後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、別個のエラーフラグ割り込み有効ビット（USART_CR3 レジスタの EIE ビット）が用意されており、これがセットされている場合、いずれかのエラーを持つ現在のバイトの後に割り込みが発行されます。

19.3.14 ハードウェアフロー制御

nCTS 入力と nRTS 出力を使用すると、2 つのデバイス間でシリアルデータフローを制御できます。
[図 188](#) 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 188. 2 つの USART 間のハードウェアフロー制御

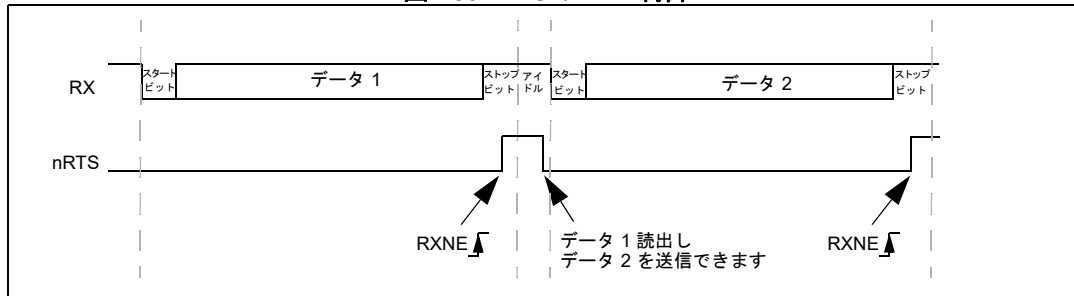


RTS と CTS のフロー制御は、USART_CR3 レジスタの RTSE ビットと CTSE ビットにそれぞれ 1 を書き込むことによって、個別に有効にできます。

RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、USART レシーバが新しいデータを受信可能である限り、nRTS がアサートされます (ローレベル接続)。受信レジスタがフルになると nRTS がネゲートされ、現在のフレームの終わりに送信が停止する予定であることを示します。[図 189](#) に、RTS フロー制御が有効な場合の通信例を示します。

図 189. RTS フロー制御

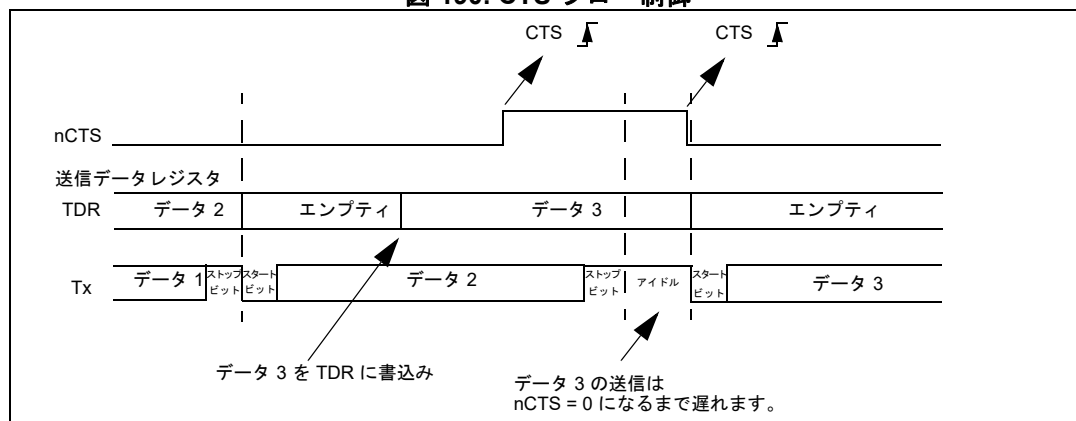


CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、nCTS 入力を検査してから、次のフレームを送信します。nCTS がアサートされていれば (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE=0 の場合)。そうでない場合、送信は行われません。送信中に nCTS がネゲートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、nCTS 入力が入力されると、CTSIF ステータスビットはハードウェアによって自動的にセットされます。このビットは、レシーバの通信準備ができてい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、割込みが生成されます。次の図に、CTS フロー制御が有効な場合の通信例を示します。

図 190. CTS フロー制御



注 : ブレークフレームの特殊な動作 : CTS フローが有効になると、トランスミッタは、ブレークを送信するために nCTS 入力の状態をチェックしません。

19.4 USART 割込み

表 87. USART 割込みリクエスト

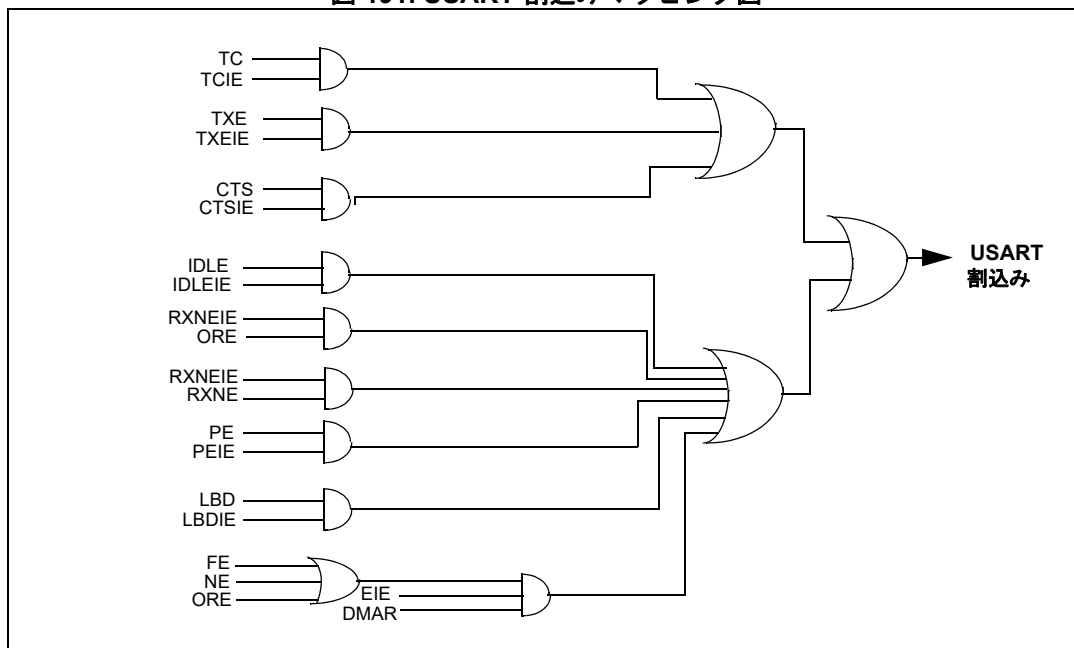
割込みイベント	イベントフラグ	有効制御ビット
送信データレジスタエンプティ	TXE	TXEIE
CTS フラグ	CTS	CTSIE
送信完了	TC	TCIE
受信データ読み出し可能	RXNE	RXNEIE
オーバーランエラー検出	ORE	
アイドルライン検出	IDLE	IDLEIE
パリティエラー	PE	PEIE
ブレークフラグ	LBD	LBDIE
マルチバッファ通信におけるノイズフラグ、オーバーランエラー、およびフレーミングエラー	NF または ORE または FE	EIE

USART 割込みイベントは、同じ割込みベクタに接続されます (図 191 を参照)。

- 送信時：送信完了割込み、CTS 割込み、または送信データレジスタエンプティ割込み
- 受信時：アイドルライン検出、オーバーランエラー、受信データレジスタノットエンプティ、パリティエラー、LIN ブレーク検出、ノイズフラグ (マルチバッファ通信時のみ)、およびフレーミングエラー (マルチバッファ通信時のみ)

これらのイベントは、対応する有効制御ビットがセットされている場合に割込みを生成します。

図 191. USART 割込みマッピング図



19.5 USART モード設定

表 88. USART モード設定⁽¹⁾

USART モード	USART1	USART2	USART6
非同期モード	X	X	X
ハードウェアフロー制御	X	X	X
マルチバッファ通信 (DMA)	X	X	X
マルチプロセッサ通信	X	X	X
同期	X	X	X
スマートカード	X	X	X
半二重 (単線モード)	X	X	X
IrDA	X	X	X
LIN	X	X	X

1. X = サポート、NA = 該当せず

19.6 USART レジスタ

レジスタの説明で使用されている略語のリストについては、[34 ページのセクション 1.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスできます。

19.6.1 ステータスレジスタ (USART_SR)

アドレスオフセット : 0x00

リセット値 : 0x00C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CTS** : CTS フラグ

このビットは、CTSE ビットがセットされていた場合、nCTS 入力がトグルしたときにハードウェアによってセットされます。ソフトウェアで 0 を書き込むことによってクリアされます。また、USART_CR3 レジスタで CTSIE=1 であれば、割込みが生成されます。

注 : 0 : nCTS ステータスラインでの変更はありません。
1 : nCTS ステータスラインで変更がありました。

ビット 8 LBD : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。ソフトウェアで 0 を書き込むことによってクリアされます。USART_CR2 レジスタの LBDIE=1 である場合、割込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

注 : LBDIE=1 の場合、LBD=1 になると割込みが生成されます。

ビット 7 TXE : 送信データレジスタエンプティ

このビットは、TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。USART_CR1 レジスタの TXEIE ビット=1 の場合、割込みが生成されます。このビットは、USART_DR レジスタへの書き込みによってクリアされます。

0 : データはシフトレジスタに転送されていません。

1 : データはシフトレジスタに転送されます。

注 : このビットは、シングルバッファ送信時に使用されます。

ビット 6 TC : 送信完了

データを含むフレームの送信が完了し、TXE がセットされている場合、このビットはハードウェアによってセットされます。USART_CR1 レジスタの TCIE=1 である場合、割込みが生成されます。このビットは USART_SR レジスタを読み出し、その後 USART_DR レジスタへの書き込みを行うソフトウェアシーケンスによってクリアされます。TC ビットは、“0”を書き込むことによってクリアできます。このクリアシーケンスは、マルチバッファ通信の場合にのみ推奨されます。

0 : 送信は完了していません。

1 : 送信は完了しています。

ビット 5 RXNE : 読み出しデータレジスタノットエンプティ

このビットは、RDR シフトレジスタの内容が USART_DR レジスタに転送されると、ハードウェアによってセットされます。USART_CR1 レジスタの RXNEIE=1 の場合、割込みが生成されます。このビットは、USART_DR レジスタへの読み出しによってクリアされます。RXNE フラグは、0 を書き込むとクリアできます。このクリアシーケンスは、マルチバッファ通信の場合にのみ推奨されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

ビット 4 IDLE : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。USART_CR1 レジスタの IDLEIE=1 である場合、割込みが生成されます。このビットは USART_SR レジスタを読み出し、その後 USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

注 : RXNE ビットがセットされる（つまり、新しいアイドルラインが出現する）まで、IDLE ビットが再びセットされることはありません。

ビット 3 ORE : オーバーランエラー

このビットは、RXNE=1 のときに、シフトレジスタで現在受信中のワードを RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USART_CR1 レジスタの RXNEIE=1 の場合、割込みが生成されます。このビットは USART_SR レジスタを読み出し、その後 USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : オーバーランエラーはありません。

1 : オーバーランエラーが検出されました。

注 : このビットがセットされると、RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。マルチバッファ通信で EIE ビットがセットされた場合、ORE フラグに基づいて割込みが生成されます。

ビット 2 NF : ノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。このビットは USART_SR レジスタを読み出し、その後に USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : 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、割込みを生成する RXNE ビットと同時に出現するため、割込みを生成しません。マルチバッファ通信で EIE ビットがセットされている場合は、NF フラグに基づいて割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 19.3.5 : 522 ページのクロック偏差に対する USART レシーバの許容誤差を参照](#))。

ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。このビットは USART_SR レジスタを読み出し、その後に USART_DR レジスタを読み出すソフトウェアシーケンスによってクリアされます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

注 : このビットは、割込みを生成する RXNE ビットと同時に出現するため、割込みを生成しません。現在転送中のワードによってフレームエラーとオーバーランエラーが発生すると、そのワードが転送され、ORE ビットのみがセットされます。

マルチバッファ通信で EIE ビットがセットされている場合は、FE フラグに基づいて割込みが生成されます。

ビット 0 PE : パリティエラー

このビットは、レシーバモードでパリティエラーが発生したときに、ハードウェアによってセットされます。このビットはステータスレジスタを読み出し、その後に USART_DR データレジスタへの読み出しまたは書き込みアクセスを行うソフトウェアシーケンスによってクリアされます。ソフトウェアは、RXNE フラグがセットされるまで待たなければ、PE ビットをクリアできません。

USART_CR1 レジスタの PEIE=1 の場合、割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

19.6.2 データレジスタ (USART_DR)

アドレスオフセット : 0x04

リセット値 : 0xFFFF XXXX

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 **DR[8:0]** : データ値

読出ししか書き込みかに応じて、受信または送信のデータキャラクタが含まれます。
データレジスタは、送信用 (TDR) と受信用 (RDR) の 2 個のレジスタから構成されるため、2 つの機能 (読出しと書き込み) を実行します。
TDR レジスタは、内部バスと出力シフトレジスタとの間に並列インタフェースを提供します (図 1 を参照)。
RDR レジスタは、入力シフトレジスタと内部バスとの間に並列インタフェースを提供します。
パリティを有効にして (USART_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。
パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

19.6.3 ボーレートレジスタ (USART_BRR)

注 : TE または RE ビットがそれぞれ無効になると、ボーカウンタはカウントを停止します。

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DIV_Mantissa[11:0]** : USARTDIV の整数部

これらの 12 ビットは、USART 分周比 (USARTDIV) の整数部を定義します。

ビット 3:0 **DIV_Fraction[3:0]** : USARTDIV の小数部

これらの 4 ビットは、USART 分周比 (USARTDIV) の小数部を定義します。OVER8=1 のとき、DIV_Fraction3 ビットは考慮されないため、クリアされた状態に保つ必要があります。

19.6.4 制御レジスタ 1 (USART_CR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	予約済み	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	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 **OVER8** : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

注 : 8 倍のオーバーサンプリングは、スマートカード、IrDA、および LIN の各モードでは使用できません。SCEN=1、IREN=1、または LINEN=1 であるとき、OVER8 はハードウェアによって“0”に固定されます。

ビット 14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **UE** : USART 有効

このビットがクリアされると、USART プリスケアラと出力は現在のバイト転送の最後に停止し、消費電力を減らします。このビットは、ソフトウェアによってセット/クリアされます。

0 : USART プリスケアラと出力は無効です。

1 : USART は有効です。

ビット 12 **M** : ワード長

このビットは、ワード長を決定します。ソフトウェアによってセット/クリアされます。

0 : スタートビット 1 個、データビット 8 個、ストップビット n 個

1 : スタートビット 1 個、データビット 9 個、ストップビット n 個

注 : M ビットは、送受信のデータ転送中に変更することはできません。

ビット 11 **WAKE** : ウェイクアップ方法

このビットは、USART のウェイクアップ方法を決定し、ソフトウェアによってセット/クリアされます。

0 : アイドルライン

1 : アドレスマーク

ビット 10 **PCE** : パリティ制御有効

このビットは、ハードウェアのパリティ制御（生成と検出）を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合はビット 9、M=0 の場合はビット 8) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

ビット 9 **PS** : パリティ選択

このビットは、パリティの生成/検出が有効である (PCE ビットがセットされている) とき、奇数パリティ/偶数パリティを選択します。ソフトウェアによってセット/クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

ビット 8 **PEIE** : PE 割込み有効

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 : 割込みは禁止されています。

1 : USART_SR レジスタで PE=1 の時に必ず USART 割込みが生成されます。

ビット 7 TXEIE : TXE 割込み有効

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 : 割込みは禁止されています。

1 : USART_SR レジスタで TXE=1 のときに必ず USART 割込みが生成されます。

ビット 6 TCIE : 転送完了割込み有効

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 : 割込みは禁止されています。

1 : USART_SR レジスタで TC=1 のときに必ず USART 割込みが生成されます。

ビット 5 RXNEIE : RXNE 割込み有効

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 : 割込みは禁止されています。

1 : USART_SR レジスタで RXNE=1 または ORE=1 のときに必ず USART 割込みが生成されます。

ビット 4 IDLEIE : IDLE 割込み有効

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 : 割込みは禁止されています。

1 : USART_SR レジスタで IDLE=1 のときに必ず USART 割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： スマートカードモードの場合を除いて、送信中に TE ビットに「0」パルスを与える（「0」に続けて「1」を書き込む）と、現在のワードの後にプリアンプル（アイドルライン）が送信されます。

TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。

ビット 1 RWU : レシーバウェイクアップ

このビットは、USART がミュートモードかどうかを決定します。ソフトウェアによってセット／クリアされ、ウェイクアップシーケンスが認識されるとハードウェアによってクリアできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

注： ミュートモードを選択する（RWU ビットをセットする）前に、USART はまずデータバイトを受信する必要があります。受信しない場合、ミュートモードでアイドルライン検出によるウェイクアップを伴う動作はできません。

アドレスマーク検出のウェイクアップ設定（WAKE=1）では、RXNE ビットがセットされている限り、ソフトウェアで RWU ビットを変更することはできません。

ビット 0 SBK : ブレーク送信

このビットは、ブレークキャラクタを送信するために使用されます。ソフトウェアによってセット／クリアできます。ソフトウェアによってセットされ、ブレークのストップビットの処理中にハードウェアによってリセットされます。

0 : ブレークキャラクタは送信されません。

1 : ブレークキャラクタは送信されます。

19.6.5 制御レジスタ 2 (USART_CR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]			
	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 **LINEN** : LIN モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : LIN モードは無効です。

1 : LIN モードは有効です。

LIN モードでは、USART_CR1 レジスタの SBK ビットを使用して LIN 同期ブレーク（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。

ビット 13:12 **STOP** : STOP ビット

このビットは、ストップビットのプログラミングに使用します。

00 : ストップビットが 1

01 : ストップビットが 0.5

10 : ストップビットが 2

注： 11 : ストップビットが 1.5

ビット 11 **CLKEN** : クロック有効

このビットによって、SCLK ピンを有効にできます。

0 : SCLK ピンは無効です。

1 : SCLK ピンは有効です。

ビット 10 **CPOL** : クロック極性

このビットによって、同期モードにおける SCLK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して動作し、希望するクロック／データ関係になるようにします。

0 : 送信ウィンドウの外で、SCLK ピンはローレベルを維持します。

1 : 送信ウィンドウの外で、SCLK ピンはハイレベルを維持します。

ビット 9 **CPHA** : クロック位相

このビットによって、同期モードにおける SCLK ピンのクロック出力の位相を選択できます。CPOL ビットと連携して動作し、希望するクロック／データ関係になるようにします（図 179～180 を参照）。

注： 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。

ビット 8 **LBCL** : 最終ビットのクロックパルス

このビットによって、同期モードで送信される最終データビット (MSB) に関連するクロックパルスを、SCLK ピンに出力する必要があるかどうかを選択します。

0 : 最終データビットのクロックパルスは、SCLK ピンに出力されません。

1 : 最終データビットのクロックパルスは、SCLK ピンに出力されます。

1 : 最終ビットは、USART_CR1 レジスタの M ビットによって選択された 8 または 9 ビットフォーマットに応じて送信された 8 番目または 9 番目のデータビットです。

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_SR レジスタで LBD=1 になるたび、割り込みが生成されます。

ビット 5 **LBDL** : *lin* ブレーク検出長
このビットでは、10 ビットと 11 ビットのブレーク検出を選択します。
0 : 10 ビットブレーク検出
1 : 11 ビットブレーク検出

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **ADD[3:0]** : USART ノードのアドレス
このビットフィールドは、USART ノードのアドレスを示します。
これは、ミュートモードでのマルチプロセッサ通信において、アドレスマーク検出によるウェイクアップに使用されます。

注 : CPOL、CPHA、LBCL の 3 ビットには、トランスミッタが有効である間は書き込まないでください。

19.6.6 制御レジスタ 3 (USART_CR3)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
				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 **ONEBIT** : 1 サンプルビット方式有効
このビットによって、サンプル方式を選択できます。1 サンプルビット方式が選択されると、ノイズ検出フラグ (NF) が無効になります。
0 : 3 サンプルビット方式
1 : 1 サンプルビット方式

ビット 10 **CTSIE** : CTS 割込み有効
注 : 0 : 割込みは禁止されています。
1 : USART_SR レジスタで CTS=1 になるたびに割込みが生成されます。

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。
注 : 1 : CTS モードが有効です。データは nCTS 入力のアサート (0 に固定) されている場合にのみ転送されます。データ転送中に nCTS 入力にネゲートされると、転送は停止前に完了します。nCTS 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ると、転送は nCTS がアサートされるまで延期されます。

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。
注 : 1 : RTS 割込みが有効です。データは受信バッファに空きがある場合にのみリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。nRTS 出力は、データが受信可能な時にアサートされます (0 に関係)。

ビット 7 **DMAT** : DMA 有効トランスミッタ
このビットは、ソフトウェアでセット / クリアされます。
1 : DMA モードが転送に有効です。
0 : DMA モードが転送に無効です。

ビット 6 DMAR : DMA 有効レシーバ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。

ビット 5 SCEN : スマートカードモード有効

このビットはスマートカードモードを有効にするために使用します。

注 : 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。

ビット 4 NACK : スマートカード NACK 有効

注 : 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。

ビット 3 HDSEL : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

ビット 2 IRLP : IrDA 低電力

このビットは、通常と低電力の IrDA モードの選択に使用されます。

0 : 通常モード

1 : 低電力モード

ビット 1 IREN : IrDA モード有効

このビットは、ソフトウェアによってセット / クリアされます。

0 : IrDA は無効です。

1 : IrDA は有効です。

ビット 0 EIE : エラー割込みイネーブル

エラー割込み有効ビットは、マルチバッファ通信 (USART_CR3 レジスタの DMAR=1) において、フレーミングエラー、オーバーランエラー、またはノイズフラグ (USART_SR レジスタの FE=1、ORE=1、または NF=1) の場合に割込み生成を有効にするために必要です。

0 : 割込みは禁止されています。

1 : USART_CR3 レジスタで DMAR=1 になるたび、および USART_SR レジスタで FE=1、ORE=1、または NF=1 になるたび、割込みが生成されます。

19.6.7 ガード時間およびプリスケアラレジスタ (USART_GTPR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

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

ビット 31:16 予約済みであり、リセット値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、ガード時間値をポーロック数で表示します。

注 : 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。

ビット 7:0 **PSC[7:0]** : プリスケーラ値

– **IrDA 低電力モード :**

PSC[7:0] = IrDA 低電力ポーレート

システムクロックを分周して低電力周波数を得るためのプリスケアラのプログラミングに使用します。クロックソースは、レジスタに与えられた値 (上位 8 ビット) で分周されます。

00000000 : 予約済み - この値はプログラミングしないでください。

00000001 : クロックソースは 1 で分周されます。

00000010 : クロックソースは 2 で分周されます。

...

– **IrDA 通常モード :** PSC には 00000001 を設定する必要があります。

– **スマートカードモード :**

PSC[4:0] : プリスケーラ値

システムクロックを分周してスマートカードのクロックを提供するプリスケアラのプログラミングに使用します。

レジスタで指定された値 (上位 5 ビット) を 2 倍して、クロックソース周波数の分周比を求めます。

00000 : 予約済み - この値はプログラミングしないでください。

00001 : クロックソースは 2 で分周されます。

00010 : クロックソースは 4 で分周されます。

00011 : クロックソースは 6 で分周されます。

...

注 : 1 : スマートカードモードが使用される場合、ビット [7:5] は無効です。

19.6.8 USART レジスタマップ

次の表に、USART のレジスタマップとリセット値を示します。

表 89. USART レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_SR	予約済み																						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	リセット値																							0	0	1	1	0	0	0	0	0	0
0x04	USART_DR	予約済み																						DR[8:0]									
	リセット値																							0	0	0	0	0	0	0	0	0	0
0x08	USART_BRR	予約済み										DIV_Mantissa[15:4]										DIV_Fraction [3:0]											
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x0C	USART_CR1	予約済み										OVER8	予約済み	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK						
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_CR2	予約済み										LINEN	STOP [1:0]	CLKEN	CPOL	CPHA	LBCL	予約済み	LBDE	LBDL	予約済み	ADD[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
0x14	USART_CR3	予約済み										ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE										
	リセット値											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	USART_GTPR	予約済み										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	0	0	0	0	0	0

レジスタ境界アドレスについては、[38 ページの表 1](#) を参照してください。

20 シリアルペリフェラルインタフェース (SPI)

20.1 SPI の概要

SPI インタフェースは、SPI プロトコルまたは I²S オーディオプロトコルをサポートする 2 つの主要機能に対応します。デフォルトで選択されているのは、SPI 機能です。ソフトウェアによって、インタフェースを SPI から I²S に切り替えることができます。

SPI (シリアルペリフェラルインタフェース) では、外部デバイスとの半二重/全二重の同期シリアル通信が可能です。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。

2 本のライン (双方向データラインを使用可能) での単方向同期転送や、CRC チェックを使用した信頼性の高い通信など、さまざまな目的に使用することができます。

I²S もまた、同期シリアル通信インタフェースです。フィリップス I²S 規格、MSB/LSB 詰め規格、PCM 規格など、4 つのオーディオ規格に対応できます。全二重モード (4 ピン使用) または半二重モード (3 ピン使用) でマスタデバイスまたはスレーブデバイスとして動作することができます。I²S が通信マスタとして設定されると、インタフェースは外部スレーブ部品にマスタクロックを供給できます。

警告 : SPI1 ピンと SPI3/I2S3 ピンは、JTAG インタフェース (SPI1_NSS は JTDI に、SPI3_NSS/I2S3_WS は JTDI に、SPI3_SCK/I2S3_CK は JTDO) に使用するいくつかのピンに割り当てることができるため、次のいずれでも可能です :

- SPI/I2S を他のピンに割り当てること
- JTAG を無効にして、SPI I/O として列挙されたピンを設定する前に SWD インタフェースを使用すること (アプリケーションのデバッグ時)、または、
- JTAG/SWD の両インタフェースを無効にすること (スタンドアロンアプリケーションの場合)

JTAG/SWD インタフェースピンの設定の詳細については、[セクション 8.3.2 : I/O ピンマルチプレクサとマッピング](#)を参照してください。

20.2 SPI および I²S の主な機能

20.2.1 SPI の機能

- 3本のラインでの全二重同期転送
- 2本のラインで単方向同期転送（双方向データラインの有無にかかわらず）
- 8または16ビットの転送フレームフォーマット選択
- マスタまたはスレーブ動作
- マルチマスタモード機能
- 8個のマスタモードボーレートプリスケアラ（最大周波数： $f_{PCLK}/2$ ）
- スレーブモード周波数（最大周波数： $f_{PCLK}/2$ ）
- マスタとスレーブの両方で高速通信
- 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（割込み機能付き）
- SPI バスビジステータスフラグ
- SPI TI モード
- 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、および CRC エラーの各フラグ（割込み機能付き）
- 1バイトの送受信バッファ（DMA 機能付き：送受信リクエスト）

20.2.2 I²S の機能

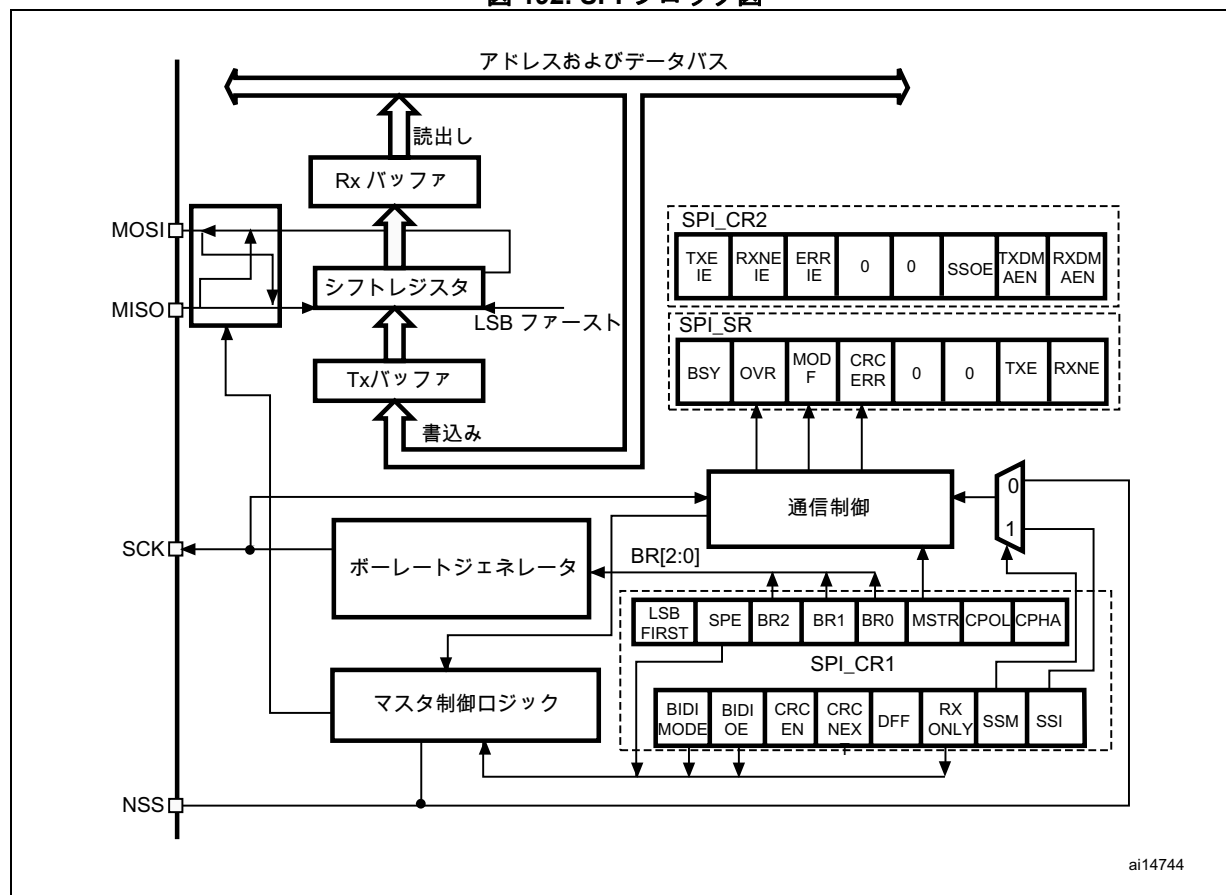
- 全二重通信
- 半二重通信（トランスミッタまたはレシーバのみ）
- マスタまたはスレーブ動作
- 正確なオーディオサンプリング周波数（8～192 kHz）を実現するプログラム可能な 8 ビットのリニアプリスケアラ
- 16、24、または 32 ビットのデータフォーマット
- パケットフレームはオーディオチャンネルによって 16 ビット（16 ビットデータフレーム）または 32 ビット（16、24、32 ビットデータフレーム）に固定。
- プログラム可能なクロック極性（定常状態）
- スレーブ送信モードのアンダーランフラグ、受信モード（マスタおよびスレーブ）のオーバーランフラグ、受信モードと送信モード（スレーブの場合のみ）のフレームエラーフラグ
- 送受信用の 16 ビットレジスタ（両チャンネルサイドに対して 1 個のデータレジスタ）
- 以下の I²S プロトコルをサポート。
 - フィリップス I²S 規格
 - MSB 詰め規格（左詰め）
 - LSB 詰め規格（右詰め）
 - PCM 規格（16 ビットチャンネルフレーム、または 32 ビットチャンネルフレームに拡張された 16 ビットデータフレームでの、ショートおよびロングフレーム同期付き）
- データ方向は常に MSB ファースト。
- 送受信（16 ビット幅）用の DMA 機能
- 外部オーディオ部品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。
- 両 I²S（I2S2 と I2S3）には専用 PLL（PLL I2S）があり、さらにより正確なクロックを生成
- I²S（I2S2 と I2S3）クロックは、I2S_CLKIN ピンに割り当てられた外部クロックから取得可能

20.3 SPI の機能説明

20.3.1 概要

SPI のブロック図を図 192 に示します。

図 192. SPI ブロック図



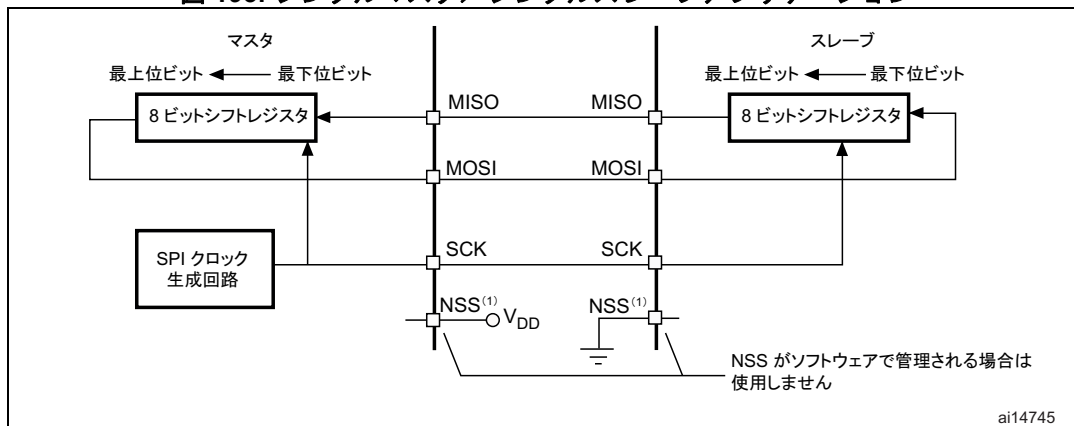
通常、SPI は 4 本のピンで外部デバイスに接続されます。

- MISO : マスターイン/スレーブアウトデータ。このピンは、マスターモードではデータの受信に、スレーブモードではデータの送信に使用できます。
- MOSI : マスターアウト/スレーブインデータ。このピンは、マスター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用できます。
- SCK : SPI マスタでは出力に、SPI スレーブでは入力に使用されます。
- NSS : スレーブ選択。これはスレーブデバイスを選択するためのオプションピンです。このピンが“チップセレクト”として機能することで、SPI マスタは複数のスレーブと個別に通信を行うことができ、データラインの競合を避けることができます。スレーブの NSS 入力は、マスタデバイスの標準入出力ポートによって駆動できます。NSS ピンは、SSOE ビットによって有効にされている場合は出力として使用することもでき、SPI がマスタ設定の場合はローレベルになります。このように、マスタの NSS ピンに接続されているデバイスからのすべての NSS ピンは、NSS ハードウェアモードに設定されている場合、ローレベルを受けてスレーブになります。NSS が入力として設定された状態で SPI がマスタモードに設定され (MSTR=1 および SSOE=0) 、NSS がローレベルに引っぱられた場合、SPI はマスタモードフォールト状態に入ります。MSTR

ビットは自動的にクリアされ、デバイスはスレーブモードに設定されます(セクション 20.3.10: 575 ページのエラーフラグを参照)。

1 個のマスタと 1 個のスレーブの間の基本的な相互接続例を図 193 に示します。

図 193. シングルマスタ/シングルスレーブアプリケーション



1. この例では、NSS ピンは入力として設定されています。

MOSI ピンと MISO ピンは、それぞれに相互接続されています。このようにして、データはマスタとスレーブの間を (MSB ファーストで) シリアルに転送されます。

通信は常にマスタによって開始されます。マスタデバイスが MOSI ピンを介してスレーブデバイスにデータを送信すると、スレーブデバイスは MISO ピンを介して応答します。このことは、データ出力とデータ入力にマスタデバイスから SCK ピンを介して供給される同じクロック信号に同期した全二重通信を意味します。

スレーブ選択 (NSS) ピンの管理

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、SPI_CR1 レジスタの SSM ビットを使用して設定することができます。

- ソフトウェア NSS 管理 (SSM = 1)

スレーブ選択情報は、SPI_CR1 レジスタの SSI ビット値に応じて内部的に反応します。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)

NSS 出力設定 (SPI_CR2 レジスタの SSOE ビット) に応じて 2 つの設定が可能です。

 - NSS 出力有効 (SSM = 0、SSOE = 1)

この設定は、デバイスがマスタモードで動作する場合にのみ使用されます。NSS 信号は、マスタが通信を開始したときにローレベルになり、SPI が無効化されるまでローレベルを保持します。
 - NSS 出力無効 (SSM = 0、SSOE = 0)

この設定により、マスタモードで動作するデバイスでマルチマスタ機能が可能になります。スレーブに設定されたデバイスでは、NSS ピンは従来型の NSS 入力として機能します (NSS がローレベルのときにスレーブが選択され、NSS がハイレベルのときに非選択)。

クロックの位相と極性

SPI_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定常値を制御します。このビットは、マスタモードとスレーブモードの両方に影響を与えます。CPOL がリセットされると、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされると、SCK ピンはハイレベルのアイドル状態になります。

CPHA (クロック位相) ビットがセットされている場合、SCK ピンの2番目のエッジ (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) が最上位ビットのキャプチャストロブです。データは、2番目のクロック遷移でラッチされます。CPHA ビットがリセットされている場合、SCK ピンでの最初のエッジ (CPOL ビットがリセットされていれば立ち上がりエッジ、セットされていれば立ち下がりエッジ) が最上位ビットのキャプチャストロブです。データは、最初のクロック遷移でラッチされます。

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 194 は、CPHA ビットと CPOL ビットの4つの組み合わせによる SPI 転送を示しています。この図は、SCK ピン、MISO ピン、MOSI ピンがマスタ/スレーブデバイス間で直接接続されている場合のマスタまたはスレーブのタイミング図とみなすことができます。

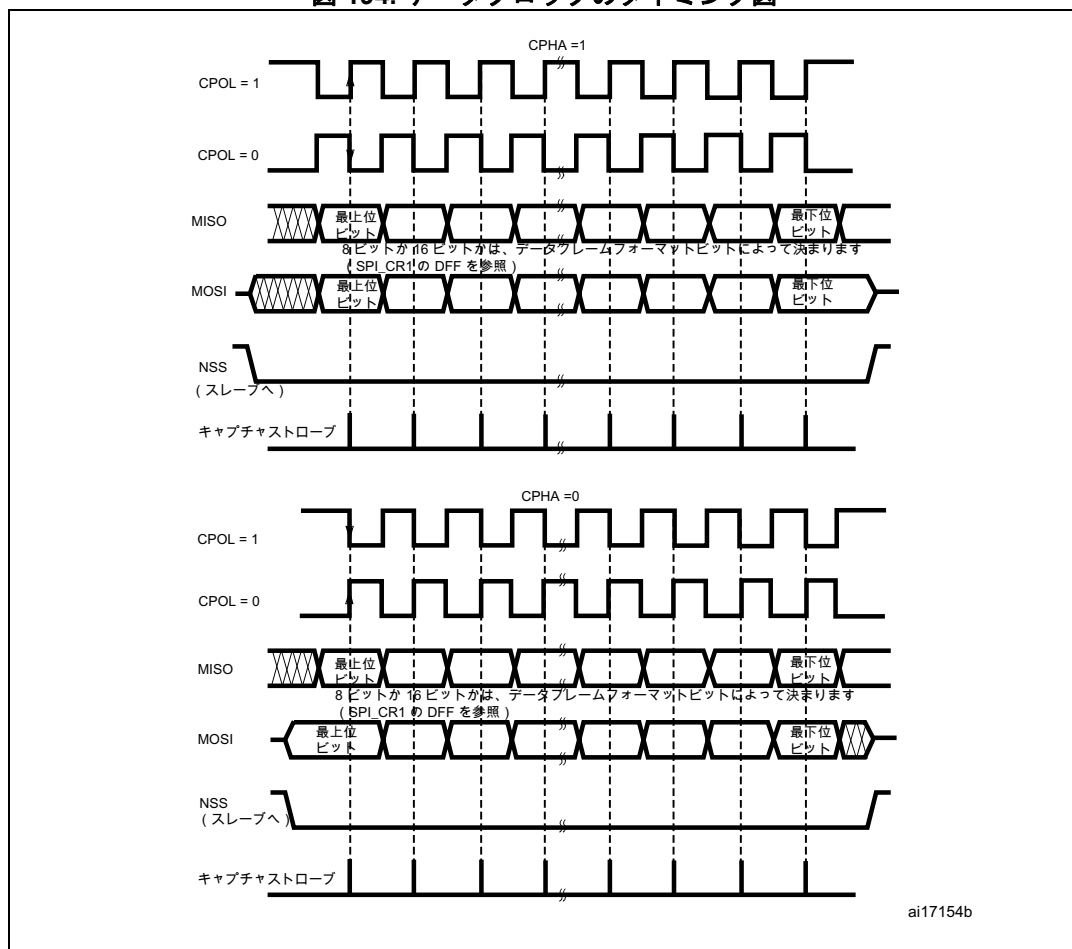
注： CPOL または CPHA ビットを変更する前に、SPE ビットをリセットすることによって、SPI を無効にする必要があります。

マスタとスレーブは、同じタイミングモードでプログラミングする必要があります。

SCK のアイドル状態は、SPI_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。

データフレームフォーマット (8 ビットまたは 16 ビット) は、SPI_CR1 レジスタの DFF ビットによって選択され、送受信時のデータ長を決定します。

図 194. データクロックのタイミング図



1. これらのタイミングは、SPI_CR1 レジスタの LSBFIRST ビットをリセットした状態で示されています。

データフレームフォーマット

データは、SPI_CR1 レジスタの LSBFIRST ビットの値に応じて、MSB ファーストまたは LSB ファーストでシフトアウトできます。

各データフレームの長さは、SPI_CR1 レジスタの DFF ビットを使用してプログラミングされたデータのサイズに応じて、8 または 16 ビットです。選択されたデータフレームフォーマットは、送受信に適用できます。

20.3.2 SPI のスレーブモード設定

スレーブ設定では、マスタデバイスからのシリアルクロックは SCK ピンで受信されます。SPI_CR1 レジスタの BR[2:0] ビットに設定された値は、データ転送速度に影響を与えません。

注： 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、通信クロックの最初のエッジまたは現在の通信の終了より前に、準備ができてい必要があります。スレーブとマスタを有効にする前に、通信クロックの極性を定常値に設定することが必要です。

SPI をスレーブモードに設定するには、次の手順に従ってください。

手順

1. DFF ビットをセットして、8 または 16 ビットのデータフレームフォーマットを定義します。
2. CPOL ビットと CPHA ビットを選択して、データ転送とシリアルクロックの間の 4 つの関係のうちの 1 つを定義します (図 194 を参照)。データを正しく転送するには、スレーブデバイスとマスタデバイスで CPOL ビットと CPHA ビットを同じに設定する必要があります。SPI_CR2 レジスタの FRF ビットにより TI モードを選択している場合、このステップは不要です。
3. フレームフォーマット (SPI_CR1 レジスタの LSBFIRST ビットの値に応じて、MSB ファーストまたは LSB ファースト) は、マスタデバイスと同じでなければなりません。TI モードを選択している場合、このステップは不要です。
4. ハードウェアモード (555 ページの [スレーブ選択 \(NSS\) ピンの管理](#) を参照) では、バイト送信シーケンス期間中は、NSS ピンをローレベル信号に接続しておく必要があります。NSS ソフトウェアモードでは、SPI_CR1 レジスタの SSM ビットをセットし、SSI ビットをクリアします。TI モードを選択している場合、このステップは不要です。
5. SPI_CR2 レジスタの FRF ビットをセットして、シリアル通信の TI モードプロトコルを選択します。
6. ピンをオルタネート機能に割り当てるには、SPI_CR1 レジスタの MSTR ビットをクリアし、SPE ビットをセットします。

この設定では、MOSI ピンはデータ入力に、MISO ピンはデータ出力になります。

送信シーケンス

データバイトは、書込みサイクル中に Tx バッファにパラレルロードされます。

スレーブデバイスが MOSI ピンでクロック信号と、データの最上位ビットを受信すると、送信シーケンスが開始されます。残りのビット (8 ビットデータフレームフォーマットでは 7 ビット、16 ビットデータフレームフォーマットでは 15 ビット) はシフトレジスタにロードされます。Tx バッファからシフトレジスタにデータが送信されると、SPI_SR レジスタの TXE フラグがセットされ、SPI_CR2 レジスタの TXEIE ビットがセットされている場合は、割り込みが生成されます。

受信シーケンス

レシーバの場合、データ転送が完了すると、

- シフトレジスタ内のデータが Rx バッファに転送され、SPI_SR レジスタの RXNE フラグがセットされます。
- SPI_CR2 レジスタの RXNEIE 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。

最後のサンプリングクロックエッジの後、RXNE ビットがセットされ、シフトレジスタに受信されたデータバイトのコピーが Rx バッファに移されます。SPI_DR レジスタが読み出されると、SPI ペリフェラルはこのバッファリングされた値を返します。

RXNE ビットのクリアは、SPI_DR レジスタを読み出すことによって行われます。

スレーブモードでの SPI TI プロトコル

スレーブモードでは、SPI インタフェースは TI プロトコルと互換性があります。SPI_CR2 レジスタの FRF ビットを使って、スレーブ SPI シリアル通信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。

SPI_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、SPI_CR1 レジスタと SPI_CR2 レジスタによる NSS 管理の設定 (SSM、SSI、SSOE など) がユーザーに対して分かりやすくなります。

スレーブモード (図 195 : TI モード - スレーブモード、シングル転送 および 図 196 : TI モード - スレーブモード、連続転送) では、SPI ボーレートプリスケラを使用して、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。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になるために要する時間 (t_{release}) は、内部再同期と SPI_CR1 レジスタの BR[2:0] 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：

$$\frac{t_{\text{baud_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud_rate}}}{2} + 6 \times t_{\text{pclk}}$$

注： 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)。

エラー割込み (ERRIE = 1) を使ってスレーブトランスミッタ専用モードで TI フレームエラーを検出するには、SPI_CR1 レジスタで BIDIMODE と BIDIOE を 1 に設定して SPI を 2 線単方向モードに設定する必要があります。BIDIMODE を 0 に設定すると、OVR が 1 にセットされます。これは、データレジスタが読み出されることがなく、エラー割込みが常に生成されるためです。これに対して BIDIMODE を 1 に設定すると、データが受信されないため、OVR はセットされません。

図 195. TI モード - スレーブモード、シングル転送

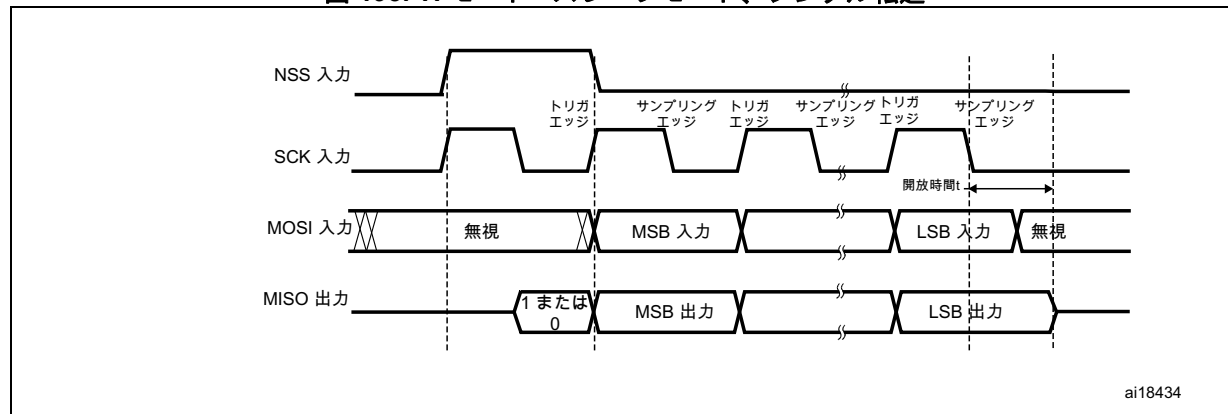
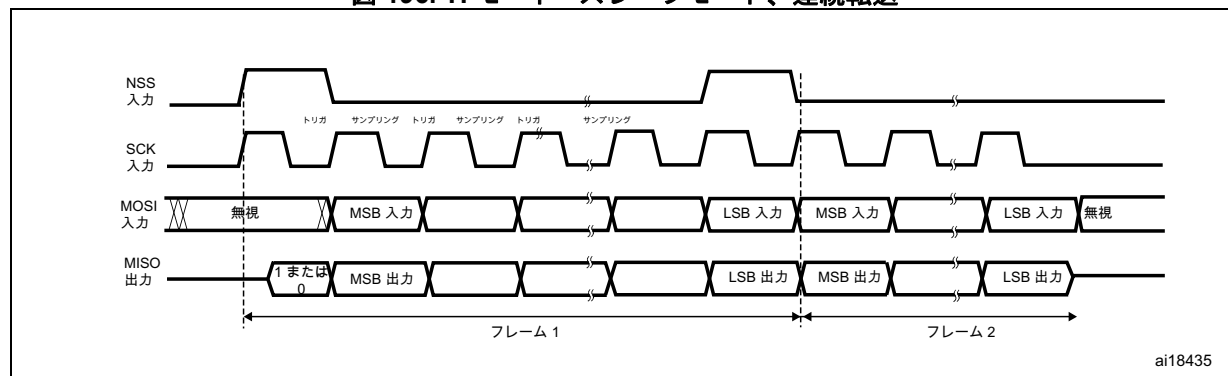


図 196. TI モード - スレーブモード、連続転送



20.3.3 SPI のマスタモード設定

マスタ設定では、シリアルクロックは SCK ピンで生成されます。

手順

1. BR[2:0] ビットを選択して、シリアルクロックのボーレートを定義します (SPI_CR1 レジスタを参照)。
2. CPOL ビットと CPHA ビットを選択して、データ転送とシリアルクロックの間の 4 つの関係のうちの 1 つを定義します (図 194 を参照)。TI モードを選択している場合、このステップは不要です。
3. DFF ビットをセットして、8 または 16 ビットのデータフレームフォーマットを定義します。
4. SPI_CR1 レジスタの 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。TI モードを選択している場合、このステップは不要です。
5. 入力モードの NSS ピンが必要な場合、ハードウェアモードで、完全なバイト送信シーケンス中に NSS ピンをハイレベル信号に接続します。NSS ソフトウェアモードで、SPI_CR1 レジスタの SSM ビットと SSI ビットをセットします。出力モードの NSS ピンが必要な場合、SSOE ビットのみをセットします。TI モードを選択している場合、このステップは不要です。
6. SPI_CR2 の FRF ビットをセットして、シリアル通信の TI プロトコルを選択します。
7. MSTR ビットと SPE ビットをセットする必要があります (これらのビットは、NSS ピンがハイレベル信号に接続されている場合にのみ、セット状態を維持します)。

この設定では、MOSI ピンはデータ出力、MISO ピンはデータ入力になります。

送信シーケンス

送信シーケンスは、Tx バッファにバイトが書き込まれたときに開始されます。

データバイトは、最初のビット送信中に内部バスからシフトレジスタに同時にロードされ、SPI_CR1 レジスタの LSBFIRST ビットに応じて MSB ファーストまたは LSB ファーストで、MOSI ピンにシリアルにシフトアウトされます。データが Tx バッファからシフトレジスタに転送されると TXE フラグがセットされ、SPI_CR2 レジスタの TXEIE 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。

受信シーケンス

レシーバの場合、データ転送が完了すると、

- シフトレジスタのデータが Rx バッファに転送され、RXNE フラグがセットされます。
- SPI_CR2 レジスタの RXNEIE 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。

最後のサンプリングクロックエッジで RXNE ビットがセットされ、シフトレジスタに受信されたデータバイトのコピーが Rx バッファに移されます。SPI_DR レジスタが読み出されると、SPI ペリフェラルはこのバッファリングされた値を返します。

RXNE ビットは、SPI_DR レジスタの読出しによってクリアされます。

送信が開始されると、次に送信されるデータが Tx バッファに格納されていれば、連続した送信ストリームを維持できます。なお、Tx バッファにデータを書き込むには、TXE フラグが 1 であることが必要です。

注： マスタが、送信と受信の間に選択解除する必要のある SPI スレーブと通信している場合、NSS ピンは GPIO として設定する必要があります。さもないと、別の GPIO を使ってソフトウェアでオン・オフする必要があります。

マスタモードでの SPI TI プロトコル

マスタモードでは、SPI インタフェースは TI プロトコルと互換性があります。SPI_CR2 レジスタの FRF ビットを使って、マスタ SPI シリアル 通信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。

SPI_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、SPI_CR1 レジスタと SPI_CR2 レジスタによる NSS 管理の設定 (SSM、SSI、SSOE など) がユーザーに対して分かりやすくなります。

図 197 : TI モード - マスタモード、シングル転送 および 図 198 : TI モード - マスタモード、連続転送) に、マスタモードで TI モードを選択したときの SPI マスタ 通信波形を示します。

図 197. TI モード - マスタモード、シングル転送

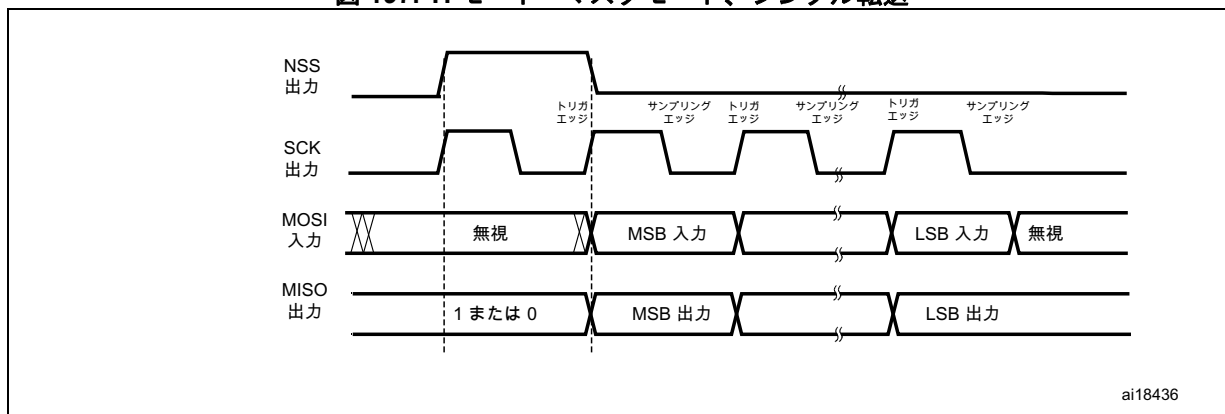
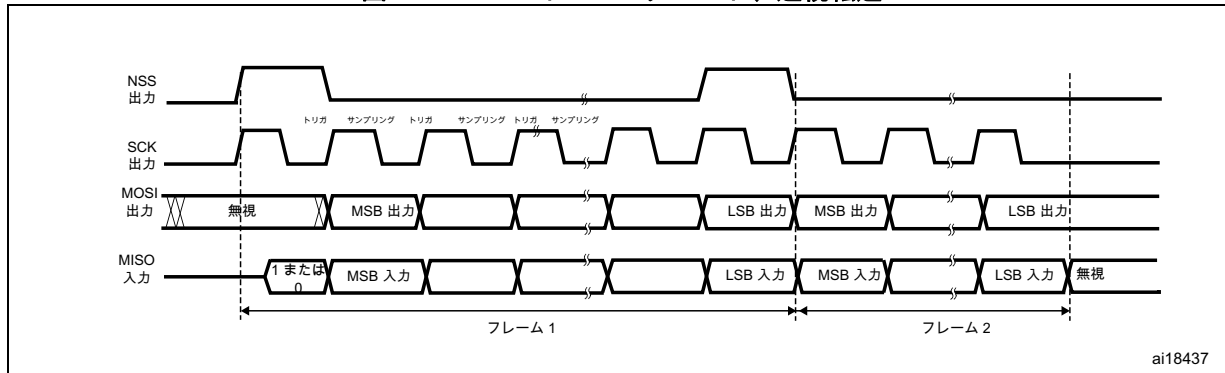


図 198. TI モード - マスタモード、連続転送



20.3.4 SPI の半二重通信設定

SPI は、次の 2 つの設定において半二重モードで動作できます。

- 1 本のクロック線と 1 本の双方向データ線
- 1 本のクロック線と 1 本のデータ線（受信専用または送信専用）

1 本のクロック線と 1 本の双方向データ線 (BIDIMODE=1)

このモードを有効にするには、SPI_CR1 レジスタの BIDIMODE ビットをセットします。このモードでは、SCK ピンがクロックに使用され、マスタモードの MOSI ピンまたはスレーブモードの MISO ピンがデータ通信に使用されます。転送方向（入力／出力）は、SPI_CR1 レジスタの BIDIOE ビットによって選択されます。このビットが 1 の場合、データラインは出力です。そうでない場合、データラインは入力です。

1 本のクロック線と 1 本の単方向データ線 (BIDIMODE=0)

このモードでは、アプリケーションは、SPI を送信専用モードまたは受信専用モードで使用できます。

- 送信専用モードは全二重モード (BIDIMODE=0、RXONLY=0) と似ています。データは送信ピン（マスタモードの MOSI ピンまたはスレーブモードの MISO ピン）で送信され、受信ピン（マスタモードの MISO ピンまたはスレーブモードの MOSI ピン）は汎用入出力として使用できます。この場合、アプリケーションは Rx バッファを無視する必要があります（データレジスタに受信したデータ値は含まれていません）。
- 受信専用モードでは、アプリケーションは、SPI_CR2 レジスタの 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。この場合、送信 I/O ピン（マスタモードの MOSI ピンまたはスレーブモードの MISO ピン）が解放されるので、他の目的に使用できます。

受信専用モードで通信を開始するには、SPI を設定して有効にします。

- マスタモードでは、通信はただちに開始され、SPE ビットがクリアされて現在の受信が停止すると、通信は停止します。このモードでは、BSY フラグを読み出す必要はありません。SPI 通信が行われているときは、常にセットされています。
- スレーブモードでは、NSS がプルダウンされていて（または NSS ソフトウェアモードで SSI ビットがクリアされていて）、SCK が動作している限り、SPI は受信を続行します。

20.3.5 データの送受信手順

RxバッファとTxバッファ

受信の場合、受信されたデータは内部の Rx バッファに格納されます。一方、送信の場合、データはまず内部の Tx バッファに格納されてから、送信されます。

SPI_DR レジスタの読み出しアクセスでは、Rx バッファに格納された値が返されます。一方、SPI_DR への書き込みアクセスでは、書き込まれたデータが Tx バッファに格納されます。

マスタモードでの開始シーケンス

- 全二重の場合 (BIDIMODE=0 および RXONLY=0)
 - データが SPI_DR レジスタ (Tx バッファ) に書き込まれると、シーケンスが開始されます。
 - その後、データは、最初のビット送信時に Tx バッファから 8 ビットシフトレジスタに同時にロードされてから、MOSI ピンに連続的にシフトアウトされます。
 - 同時に、MISO ピンで受信されたデータは、8 ビットシフトレジスタに連続的にシフトインされてから、SPI_DR レジスタ (Rx バッファ) に同時にロードされます。
- 単方向受信専用モードの場合 (BIDIMODE=0 および RXONLY=1)
 - SPE=1 になると、このシーケンスはすぐに開始されます。
 - レシーバのみが有効になり、MISO ピンで受信されたデータは 8 ビットシフトレジスタに連続的にシフトインされてから、SPI_DR レジスタ (Rx バッファ) に同時にロードされます。
- 双方向モードでの送信時 (BIDIMODE=1 および BIDIOE=1)
 - データが SPI_DR レジスタ (Tx バッファ) に書き込まれると、シーケンスが開始されます。
 - その後、データは、最初のビット送信時に Tx バッファから 8 ビットシフトレジスタに同時にロードされてから、MOSI ピンに連続的にシフトアウトされます。
 - データは受信されません。
- 双方向モードでの受信時 (BIDIMODE=1 および BIDIOE=0)
 - SPE=1 および BIDIOE=0 になると、このシーケンスはすぐに開始されます。
 - MOSI ピンで受信されたデータは、8 ビットシフトレジスタに連続的にシフトインされてから、SPI_DR レジスタ (Rx バッファ) に同時にロードされます。
 - トランスミッタは有効にされず、MOSI ピンに連続的にシフトアウトされるデータはありません。

スレーブモードでの開始シーケンス

- 全二重モードの場合 (BIDIMODE=0 および RXONLY=0)
 - スレーブデバイスがクロック信号と、MOSI ピンでデータの最初のビットを受信すると、このシーケンスが開始されます。残りの 7 ビットは、シフトレジスタにロードされます。
 - 同時に、最初のビット送信時にデータは Tx バッファから 8 ビットシフトレジスタに同時にロードされてから、MISO ピンに連続的にシフトアウトされます。SPI マスタデバイスが転送を開始する前に、ソフトウェアは送信すべきデータを書き込んでおく必要があります。
- 単方向受信専用モードの場合 (BIDIMODE=0 および RXONLY=1)
 - スレーブデバイスがクロック信号と、MOSI ピンでデータの最初のビットを受信すると、このシーケンスが開始されます。残りの 7 ビットは、シフトレジスタにロードされます。
 - トランスミッタは有効にされず、MISO ピンに連続的にシフトアウトされるデータはありません。
- 双方向モードでの送信時 (BIDIMODE=1 および BIDIOE=1)
 - スレーブデバイスがクロック信号を受信し、Tx バッファの最初のビットが MISO ピンで送信されると、このシーケンスが開始されます。
 - その後、データは、最初のビット送信時に Tx バッファから 8 ビットシフトレジスタに同時にロードされてから、MISO ピンに連続的にシフトアウトされます。SPI マスタデバイスが転送を開始する前に、ソフトウェアは送信すべきデータを書き込んでおく必要があります。
 - データは受信されません。

- 双方向モードでの受信時 (BIDIMODE=1 および BIDIOE=0)
 - スレーブデバイスがクロック信号と、MISO ピンでデータの最初のビットを受信すると、このシーケンスが開始されます。
 - MISO ピンで受信されたデータは、8 ビットシフトレジスタに連続的にシフトインされてから、SPI_DR レジスタ (Rx バッファ) に同時にロードされます。
 - トランスマッタは有効にされず、MISO ピンに連続的にシフトアウトされるデータはありません。

データの送受信処理

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 フラグ (Tx バッファエンプティ) がセットされます。このフラグは、内部のTx バッファに次のデータをロードする準備ができていますを示します。SPI_CR2 レジスタの TXEIE ビットがセットされている場合は、割込みを生成できます。TXE ビットは、SPI_DR レジスタへの書き込みによってクリアされます。

注： ソフトウェアは、Tx バッファへの書き込みを行う前に、TXE フラグが 1 にセットされているのを確認する必要があります。さもなければ、以前にTx バッファに書き込まれたデータが上書きされます。

データがシフトレジスタからRx バッファに転送されると、最後のサンプリングクロックエッジでRXNE フラグ (Rx バッファノートエンプティ) がセットされます。このフラグは、SPI_DR レジスタからデータを読み出す準備ができていますを示します。SPI_CR2 レジスタの RXNEIE ビットがセットされている場合は、割込みを生成できます。RXNE ビットは、SPI_DR レジスタの読出しによってクリアされます。

設定によっては、最後のデータ転送時に BSY フラグを使用して、転送の完了まで待つことができます。

マスタまたはスレーブモードでの全二重送受信手順 (BIDIMODE=0 および RXONLY=0)

データを送受信するには、ソフトウェアは次の手順に従う必要があります (図 199 および図 200 を参照)。

1. SPE ビットを 1 にセットして、SPI を有効にします。
2. 送信する最初のデータ項目を SPI_DR レジスタに書き込みます (これによって TXE フラグがクリアされます)。
3. TXE=1 になるまで待ち、送信する 2 番目のデータ項目を書き込みます。その後、RXNE=1 になるまで待ち、SPI_DR を読み出して、最初の受信データ項目を取得します (これによって RXNE ビットがクリアされます)。n-1 個のデータを受信するまで、送受信するデータ項目ごとにこの操作を繰り返します。
4. RXNE=1 になるまで待ち、最後の受信データを読み出します。
5. TXE = 1 になるまで待ち、さらに BSY = 0 になるまで待つてから、SPI を無効にします。

この手順は、RXNE または TXE フラグの立ち上がりエッジのたびに起動される専用の割込みサブルーチンを使用しても実行できます。

図 199. マスタ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合

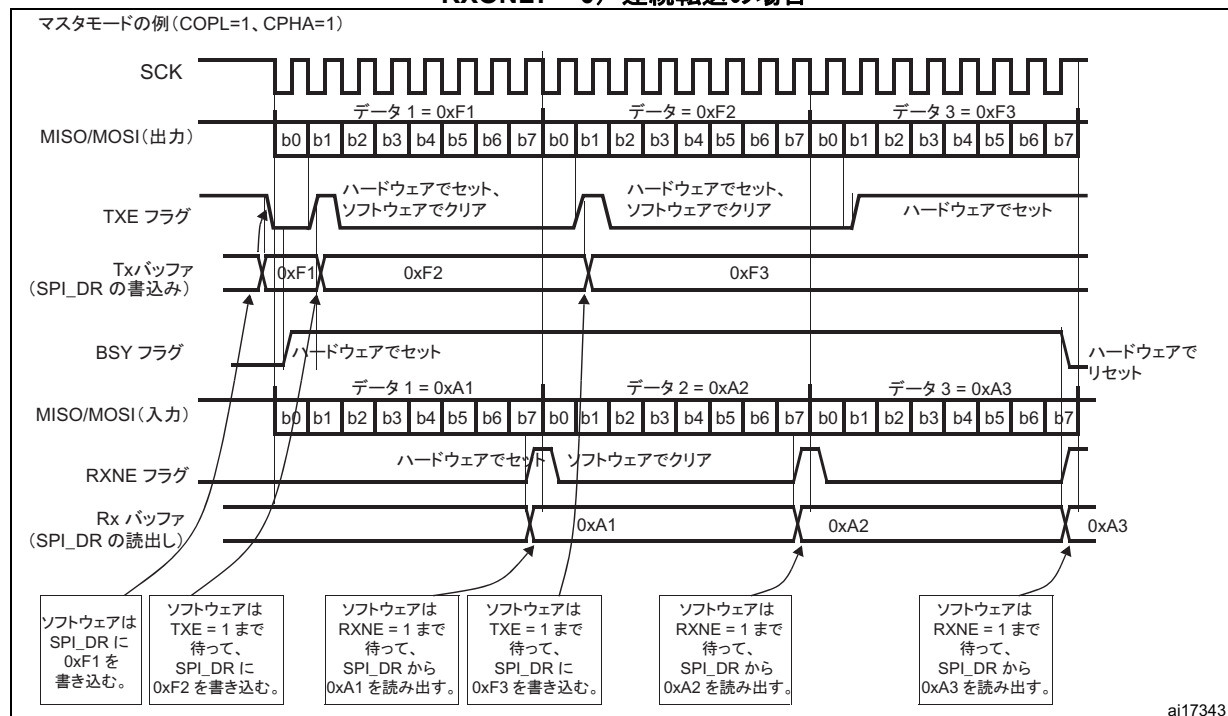
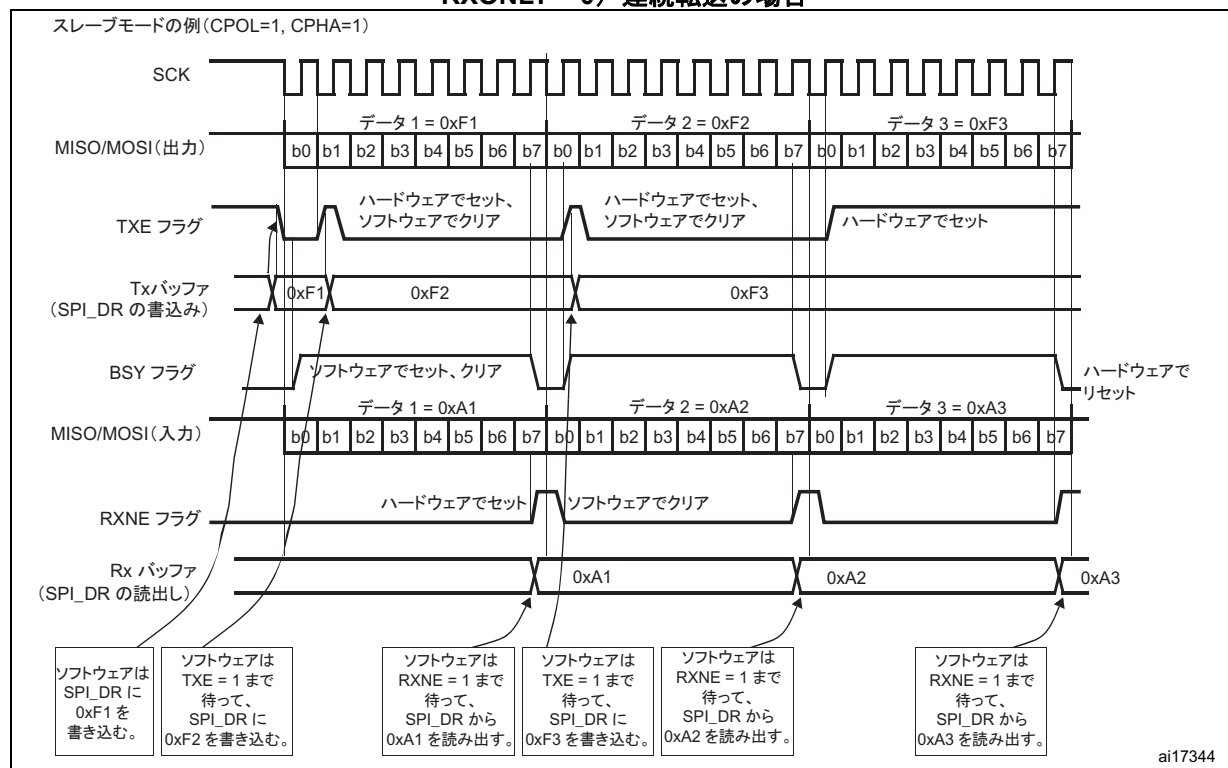


図 200. スレーブ/全二重モードでの TXE/RXNE/BSY 動作 (BIDIMODE = 0、RXONLY = 0) 連続転送の場合



送信専用手順 (BIDIMODE=0、RXONLY=0)

このモードでは、手順は下記のように短縮でき、BSY ビットを使用して送信の完了まで待つことができます (図 201 と図 202 を参照)。

1. SPE ビットを 1 にセットして、SPI を有効にします。
2. 送信する最初のデータ項目を SPI_DR レジスタに書き込みます (これによって TXE ビットがクリアされます)。
3. TXE=1 になるまで待ち、送信する次のデータ項目を書き込みます。送信するデータ項目ごとに、このステップを繰り返します。
4. SPI_DR レジスタに最後のデータ項目を書き込んだ後、TXE=1 になるまで待ち、さらに BSY=0 になるまで待ちます。これは最後のデータの送信が完了したことを示します。

この手順は、TXE フラグの立ち上がりエッジのたびに起動される専用の割り込みサブルーチンを使用しても実行できます。

注： 不連続通信時には、SPI_DR への書き込み動作と BSY ビットの設定の間に、APB の 2 クロック周期分の遅延があります。その結果、送信専用モードでは、最後のデータを書き込んだ後、まず TXE がセットされるまで待ち、さらに BSY がクリアされるまで待つ必要があります。

2 つのデータ項目を送信専用モードで送信した後、受信データが読み出されることはないので、SPI_SR レジスタの OVR フラグがセットされます。

図 201. マスタ送信専用モードでの TXE/BSY 動作 (BIDIMODE=0 および RXONLY=0)
連続転送の場合

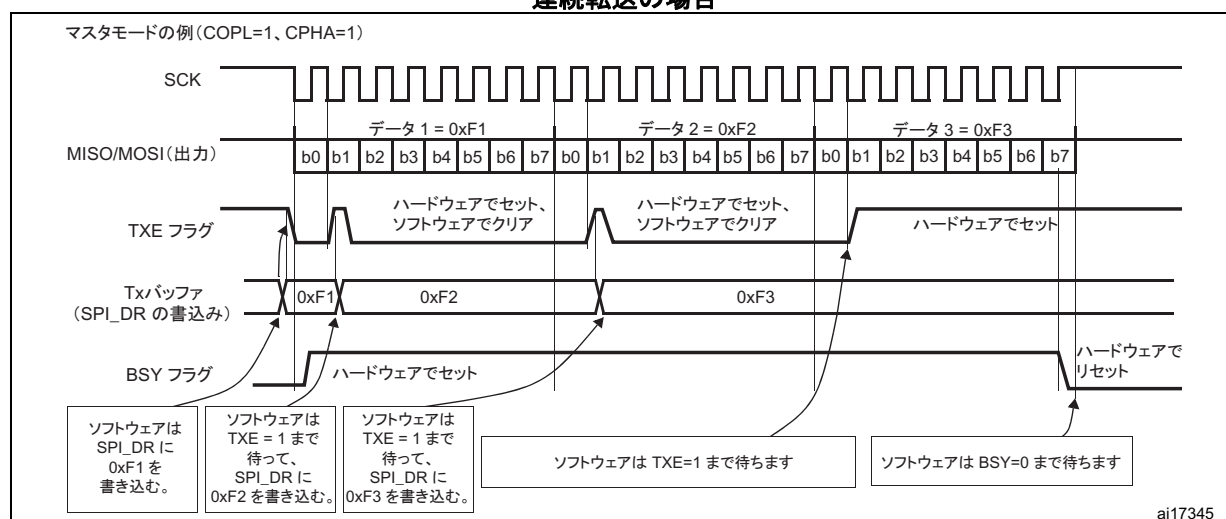
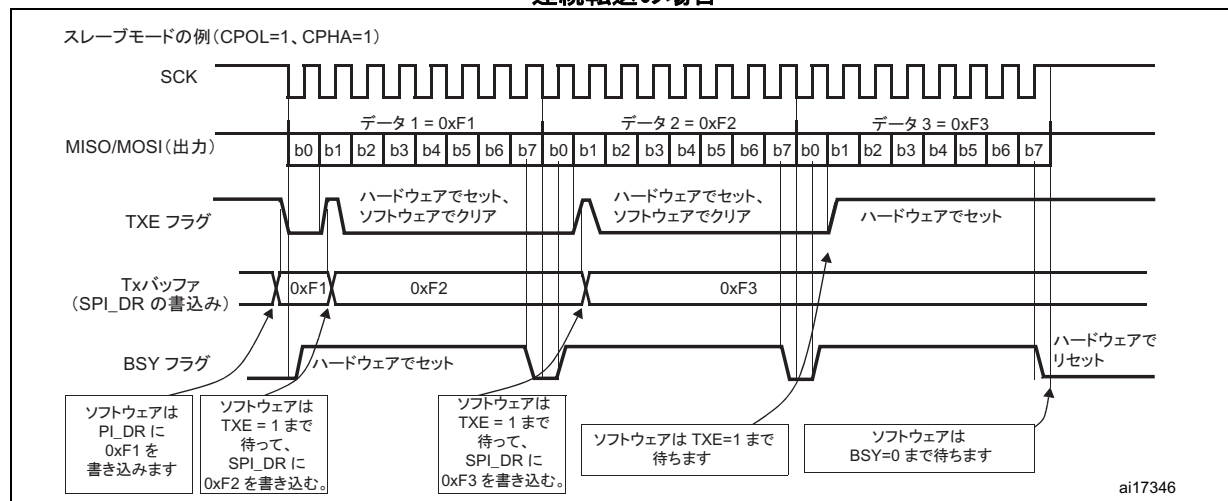


図 202. スレーブ送信専用モードでの TXE/BSY 動作 (BIDIMODE=0 および RXONLY=0)
連続転送の場合



双方向送信手順 (BIDIMODE=1 および BIDIOE=1)

このモードでは、手順は送信専用モードの手順と似ていますが、SPI を有効にする前に、SPI_CR2 レジスタの BIDIMODE ビットと BIDIOE ビットをセットする必要があります。

単方向受信専用手順 (BIDIMODE=0 および RXONLY=1)

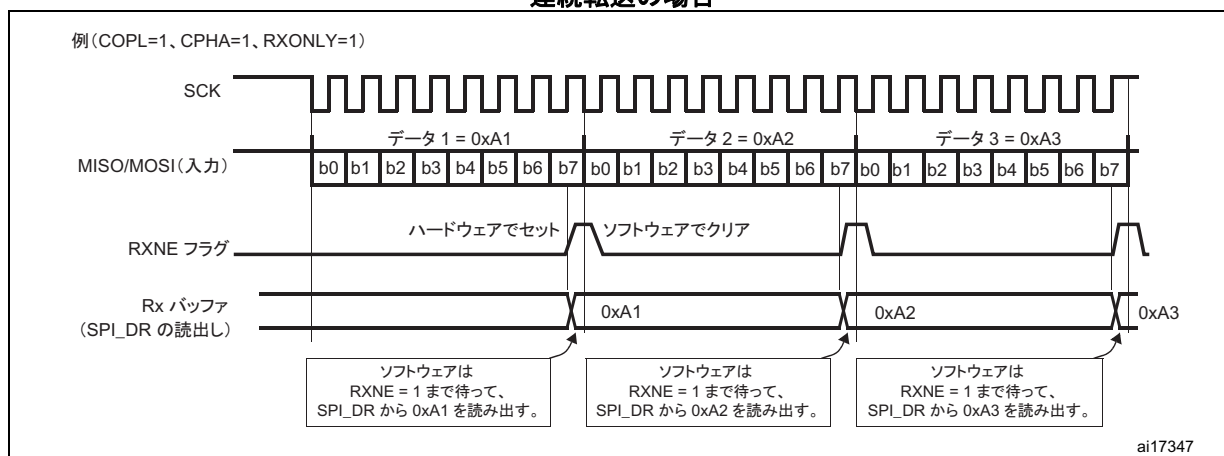
このモードでは、手順は下記のように短縮できます (図 203 を参照) :

- SPI_CR2 レジスタの RXONLY ビットをセットします。
- SPE ビットを 1 にセットすることによって、SPI を有効にします :
 - マスタモードでは、これによって SCK クロックの生成がすぐに有効になり、データは、SPI が無効にされる (SPE=0) まで連続的に受信されます。
 - スレーブモードでは、SPI マスタデバイスが NSS をローレベルに駆動して SCK クロックを生成すると、データが受信されます。
- RXNE=1 になるまで待ち、SPI_DR レジスタから受信データを読み出します (これによって RXNE ビットがクリアされます)。受信されるデータ項目ごとにこの操作を繰り返します。

この手順は、RXNE フラグの立ち上がりエッジのたびに起動される専用の割り込みサブルーチンを使用しても実行できます。

注 : 最後の転送の後に SPI を無効にする必要がある場合は、[セクション 20.3.8 : 572 ページの SPI の無効化](#)に記載されている推奨事項に従ってください。

図 203. 受信専用モードでの RXNE 動作 (BIDIRMODE=0 および RXONLY=1)
連続転送の場合



双方向受信手順 (BIDIMODE=1 および BIDIOE=0)

このモードでは、手順は受信専用モードの手順と似ていますが、SPI を有効にする前に、SPI_CR2 レジスタの BIDIMODE ビットをセットし、BIDIOE ビットをクリアする必要があります。

連続転送と不連続転送

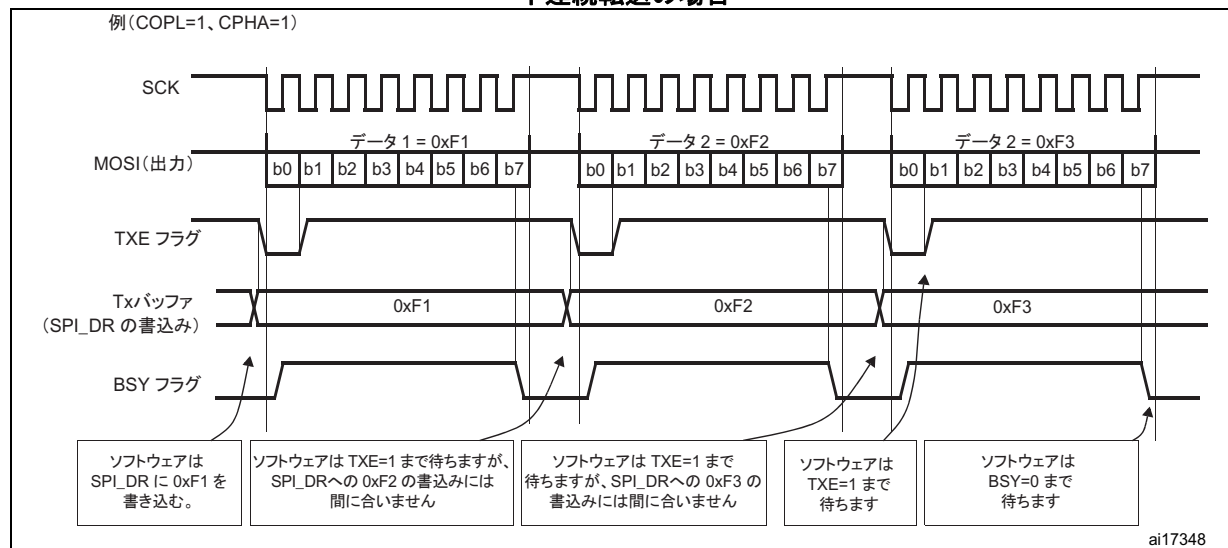
マスタモードでデータを送信するとき、ソフトウェアが十分に高速であり、TXE (または TXE 割込み) の各立ち上がりエッジを検出し、現在のデータ転送が完了する前に SPI_DR レジスタにすぐに書込みできる場合、通信は連続的であると言われます。この場合、各データ項目間で SPI クロックを生成する際に不連続性はなく、各データ転送間で BSY ビットがクリアされることはありません。

逆に、ソフトウェアが十分に高速でない場合、通信に若干の不連続性が生じることがあります。この場合、各データ送信間で BSY ビットがクリアされます (図 204 を参照)。

マスタ受信専用モード (RXONLY=1) では、通信は常に連続的であり、BSY フラグは常に 1 で読み出されます。

スレーブモードでは、通信の連続性は SPI マスタデバイスによって決定されます。いずれにしても、通信が連続的であっても、BSY フラグは、各転送間で SPI の 1 クロックサイクルという最小期間だけローレベルになります (図 202 を参照)。

図 204. 送信時の TXE/BSY 動作 (BIDIRMODE=0 および RXONLY=0)
不連続転送の場合



20.3.6 CRC 計算

CRC 計算機は、信頼性の高い通信を実現するために実装されました。送信データと受信データに対して、別々の CRC 計算機が実装されています。CRC は、各ビットに対してプログラミング可能な多項式を連続的に使用して計算されます。CRC は、SPI_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計算されます。

注： SPI が提供する 2 種類の CRC 計算基準は、送受信用に選択されたデータフレームフォーマットに直接依存します (8 ビットデータでは CR8、16 ビットデータでは CRC16)。

CRC 計算は、SPI_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。これによって CRC レジスタ (SPI_RXCRCR および SPI_TXCRCR) がリセットされます。全二重モードまたは送信専用モードでは、転送がソフトウェアで管理される場合 (CPU モード)、送信される最後のデータが SPI_DR に書き込まれた直後に、CRCNEXT ビットを書き込む必要があります。この最終データ転送の終了時、SPI_TXCRCR の値が送信されます。

受信専用モードで転送がソフトウェアで管理される場合 (CPU モード)、最後から 2 番目のデータが受信された後に CRCNEXT ビットを書き込む必要があります。最終データの受信後に CRC が受信された後、CRC チェックが実行されます。

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 SPI_SR レジスタの CRCERR フラグがセットされます。

Tx バッファにデータが存在する場合、CRC 値はデータバイトの送信後にのみ送信されます。CRC の送信時に、CRC 計算機はスイッチオフされ、レジスタの値は変化しません。

CRC を使用した SPI 通信は、次の手順によって可能になります。

1. CPOL、CPHA、LSBFirst、BR、SSM、SSI、および MSTR の値をプログラミングします。
2. SPI_CR1 レジスタに多項式をプログラミングします。
3. SPI_CR1 レジスタの CRCEN ビットをセットして、CRC 計算を有効にします。これによって、SPI_RXCRCR レジスタと SPI_TXCRCR レジスタがクリアされます。
4. SPI_CR1 レジスタの SPE ビットをセットして、SPI を有効にします。
5. 通信を開始し、1 つを除いてすべてのバイトまたはハーフワードが送受信されるまで、通信を維持します。
 - 全二重モードまたは送信専用モードで、転送がソフトウェアで管理される場合、最後のバイトまたはハーフワードを Tx バッファに書き込んだときに、SPI_CR1 レジスタの CRCNEXT ビットをセットし、最終バイトの送信後に CRC が送信されることを示します。
 - 受信専用モードでは、最後から 2 番目のデータを受信した直後に CRCNEXT ビットをセットして、最終データの受信終了時に、SPI が CRC フェーズに入る準備を行います。CRC の転送中、CRC 計算は停止します。
6. 最終バイトまたはハーフワードの転送後、SPI は CRC 転送およびチェックのフェーズに入ります。全二重モードまたは受信専用モードでは、受信した CRC は SPI_RXCRCR 値と比較されます。値が一致しない場合、SPI_SR レジスタの CRCERR フラグがセットされ、SPI_CR2 レジスタの ERRIE ビットがセットされていれば、割込みを生成できます。

注： SPI がスレーブモードのとき、CRC 計算を有効にするのは、クロックが安定している（つまり、クロックが定常状態にある）ときのみにしてください。そうしないと、CRC 計算を間違えることがあります。実際、SPE ビットの値に関係なく、CRCEN がセットされると、CRC はすぐに SCK スレーブ入力クロックの影響を受けます。

高ビットレート周波数では、CRC の送信時に注意が必要です。CRC 転送フェーズで使用する CPU サイクル数は、できるだけ少なくする必要があります。したがって、最後のデータと CRC 受信におけるエラーを避けるために、CRC 送信シーケンスでソフトウェア関数を呼び出すことは禁止されています。つまり、CRCNEXT ビットには、最後のデータの送受信が終わるまでに書き込む必要があります。

ビットレートの高い周波数では、CPU アクセスが SPI 帯域幅に影響を与えることによる SPI 速度性能の低下を避けるため、DMA モードの使用をお勧めします。

デバイスがスレーブとして設定され、NSS ハードウェアモードが使用される場合、データフェーズと CRC フェーズの間で NSS ピンをローレベルに保持する必要があります。

CRC 機能を有効にした状態で SPI がスレーブモードに設定されると、NSS ピンにハイレベルが入力された場合でも、CRC 計算が行われます。これは、たとえば、通信マスタが複数のスレーブに交互に対処するマルチスレーブ環境の場合に起こることがあります。

スレーブの選択解除（NSS のハイレベル）と新しいスレーブの選択（NSS のローレベル）の間では、マスタとスレーブをそれぞれの CRC 計算用に再同期するために、マスタ側とスレーブ側の両方で CRC 値をクリアしてください。

CRC をクリアするには、次の手順に従います。

1. SPI を無効にします（SPE = 0）。
2. CRCEN ビットをクリアします。
3. CRCEN ビットをセットします。
4. SPI を有効にします（SPE = 1）。

20.3.7 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、4 つのステータスフラグが用意されています。

Tx バッファエンプティフラグ (TXE)

このフラグがセットされると、Tx バッファが空であり、次に送信するデータをバッファにロードできることを示します。TXE フラグは、SPI_DR レジスタへの書込み時にクリアされます。

Rx バッファノートエンプティ (RXNE)

このフラグがセットされると、Rx バッファに有効な受信データがあることを示します。このフラグは SPI_DR の読出し時にクリアされます。

ビジーフラグ (BSY)

この 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（このフラグへの書込みは無効です）。BSY フラグは、SPI の通信の状態を示します。

BSY がセットされると、SPI が通信中でビジーであることを示します。マスタモード／双方向受信モード（MSTR=1、BDM=1、および BDOE=0）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 SPI を無効にして停止モードに入りたい（またはペリフェラルクロックを無効にしたい）場合、BSY フラグは転送の終わりを検出するのに役立ちます。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、マルチマスタシステムでの書込み衝突の回避にも役立ちます。

BSY フラグは転送が開始されるとセットされます。ただし、マスタモード／双方向受信モード（MSTR=1、BDM=1、および BDOE=0）の場合を除きます。

BSY フラグは次の場合にクリアされます。

- 転送が終わったとき（通信が連続的である場合はマスタモードを除きます）
- SPI が無効にされたとき
- マスタモード障害が発生したとき（MODF=1）

通信が連続的でないとき、BSY フラグは各通信間でローレベルになります。

通信が連続的な場合

- マスタモードでは、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 SPI 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。

20.3.8 SPI の無効化

転送が終了すると、アプリケーションは SPI ペリフェラルを無効にすることによって、通信を停止することができます。これには SPE ビットをクリアします。

設定によっては、転送が行われているときに SPI を無効にして停止モードに入ると、現在の転送内容が破壊されたり、BSY フラグが信頼できなくなることがあります。

このような影響を回避するには、SPI を無効にするとき、以下の手順を順守することをお勧めします。

マスタまたはスレーブの全二重モード (BIDIMODE=0、RXONLY=0)

1. RXNE=1 になるまで待ってから、最後のデータを受信します。
2. TXE=1 になるまで待ちます。
3. 次に、BSY=0 になるまで待ちます。
4. SPI を無効にし (SPE=0)、結果的に停止モードに入ります (あるいは、ペリフェラルクロックを無効にします)。

マスタまたはスレーブの単方向送信専用モード (BIDIMODE=0、RXONLY=0) または双方向送信モード (BIDIMODE=1、BIDIOE=1)

SPI_DR レジスタに最後のデータが書き込まれた後：

1. TXE=1 になるまで待ちます。
2. 次に、BSY=0 になるまで待ちます。
3. SPI を無効にし (SPE=0)、結果的に停止モードに入ります (あるいは、ペリフェラルクロックを無効にします)。

マスタの単方向受信専用モード (MSTR=1、BIDIMODE=0、RXONLY=1) または双方向受信モード (MSTR=1、BIDIMODE=1、BIDIOE=0)

このケースは、SPI が新しい転送を開始しないように、特別の方法で管理する必要があります。次のシーケンスは SPI モトローラ設定に対してのみ有効です (FRF ビットに 0 を設定)：

1. RXNE=1 (n-1) の最後から 2 番目の出現を待ちます。
2. ソフトウェアループを使用して SPI の 1 クロックサイクルの間待ってから、SPI を無効にします (SPE=0)。
3. 次に、最後の RXNE=1 になるまで待ってから、停止モードに入ります (あるいは、ペリフェラルクロックを無効にします)。

SPI が TI モードに設定されている場合 (FRF ビットを 1 に設定)、次の手順を守り、SPI を無効化する際に NSS 上に不要なパルスが発生するのを防止する必要があります。

1. RXNE=1 (n-1) の最後から 2 番目の出現を待ちます。
2. ソフトウェアループを使って、次のウインドウフレームで SPI を無効化します (SPE = 0)：
 - 少なくとも 1 SPI クロックサイクル経過後で、
 - LSB データ転送の開始前。

注： マスタの双方向受信モード (MSTR=1、BDM=1、および BDOE=0) では、BSY フラグは転送時にローレベルに保持されます。

スレーブの受信専用モード (MSTR=0、BIDIMODE=0、RXONLY=1) または双方向受信モード (MSTR=0、BIDIMODE=1、BIDOE=0)

1. SPI はいつでも無効にできます (SPE=0 の書込み)。現在の転送が完了してから、SPI は実質的に無効になります。
2. 次に、停止モードに入りたい場合は、まず BSY = 0 になるまで待ってから、停止モードに入ります (あるいは、ペリフェラルクロックを無効にします)。

20.3.9 DMA (ダイレクトメモリアクセス) を使用する SPI 通信

SPI が最大速度で動作するには、SPI に送信用のデータを供給し、Rx バッファに受信したデータを読み出してオーバーランを回避する必要があります。転送を容易にするため、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。

SPI_CR2 レジスタのイネーブルビットを有効にすると、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 (図 205 および図 206 を参照)。

- 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 SPI_DR レジスタに書き込みます (これによって TXE フラグがクリアされます)。
- 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 SPI_DR レジスタを読み出します (これによって RXNE フラグがクリアされます)。

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、受信され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 が送信されるすべてのデータを書き込んだとき (DMA_ISR レジスタのフラグ TCIF がセットされます)、BSY フラグを監視することで SPI 通信の完了を確認できます。最後の送信内容の破壊を避けるために、SPI を無効にする前、または STOP モードに入る前にこの操作を行う必要があります。ソフトウェアは、まず TXE=1 になるまで待ってから、BSY=0 になるまで待つ必要があります。

注： 不連続通信時には、SPI_DR への書込み動作と BSY ビットの設定の間に、APB の 2 クロック周期分の遅延があります。結果として、最後のデータを書き込んだ後、まず TXE=1 になるまで待ち、さらに BSY=0 になるまで待つ必要があります。

図 205. DMA を使用した送信

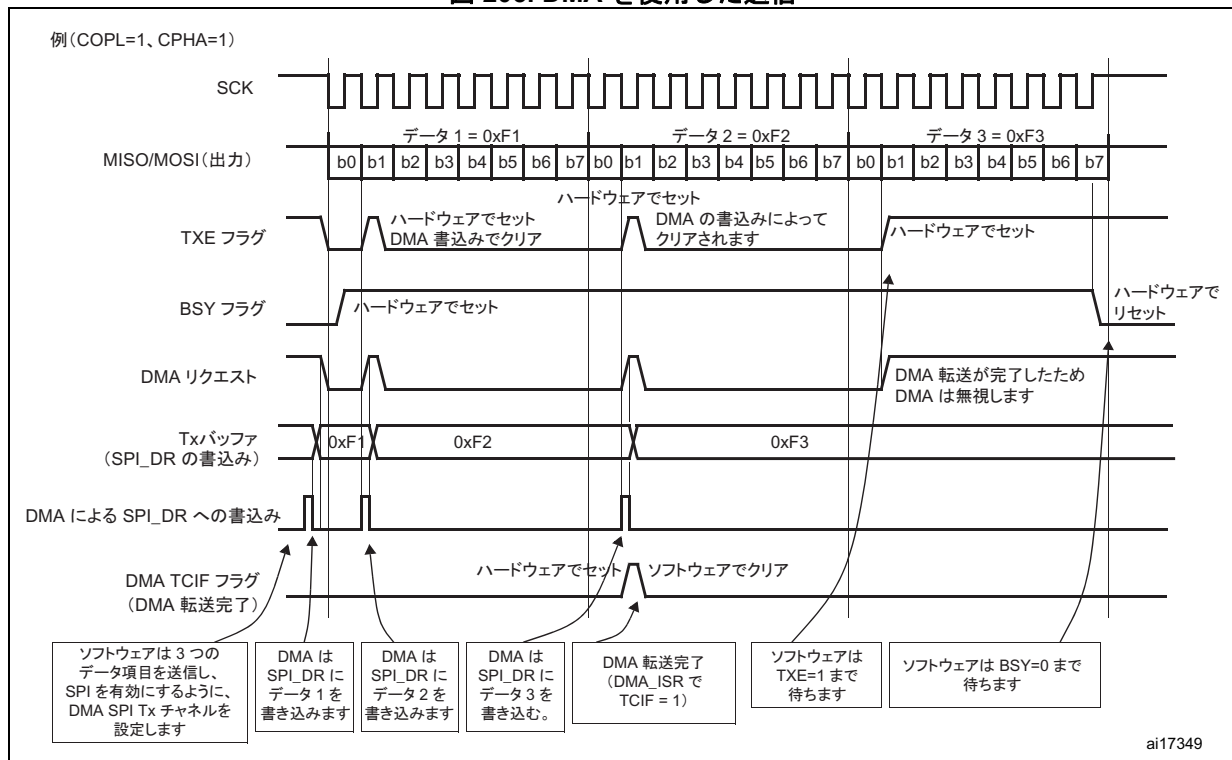
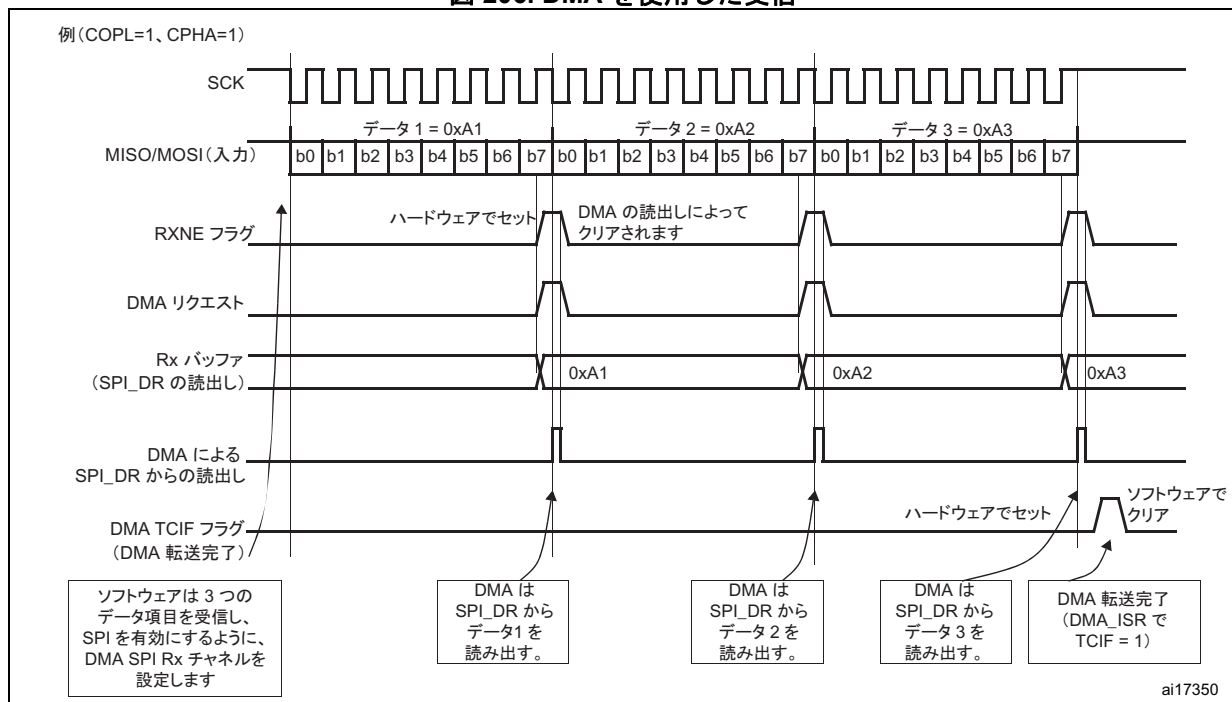


図 206. DMA を使用した受信



CRC 付きの DMA 機能

CRC 通信付きの DMA モードで SPI 通信が有効にされると、CRCNEXT ビットを使用することなく、通信の最後で CRC の送受信が自動的に行われます。CRC 受信後、SPI_DR レジスタから CRC を読み出し、RXNE フラグをクリアする必要があります。

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 SPI_SR レジスタの CRCERR フラグがセットされます。

20.3.10 エラーフラグ

マスタモードフォールト (MODF)

マスタモードフォールトは、マスタデバイスが NSS ピンをローにしたとき (NSS ハードウェアモード)、または SSI ビットをローにしたとき (NSS ソフトウェアモード) に発生し、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 ペリフェラル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に、SPI_SR レジスタに読み出し/書き込みアクセスを行います。
2. 次に、SPI_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 ビットをセットできません。ただし、マルチマスタ設定では、デバイスはこの MODF ビットをセットした状態でスレーブモードになることができます。この場合、MODF ビットは、システム制御に関してマルチマスタ競合が生じた可能性を示します。割込みルーチンを使用して、リセットを行ったりデフォルト状態に復帰することによって、この状態からクリーンに回復できます。

オーバーラン条件

オーバーラン条件が発生するのは、マスタデバイスがデータバイトを送信し、スレーブデバイスが以前に送信されたデータバイトに起因する RXNE ビットをクリアしなかった場合です。オーバーラン条件が発生すると、

- OVR ビットがセットされ、ERRIE ビットがセットされている場合は割込みが生成されます。

この場合、レシーババッファは、マスタデバイスから新しく受信したデータによって更新されません。このバイトは、SPI_DR レジスタの読み出しによって返されます。その後に送信されたすべてのバイトは失われます。

OVF ビットのクリアは、SPI_DR レジスタを読み出した後に SPI_SR レジスタを読み出すことによって行われます。

CRC エラー

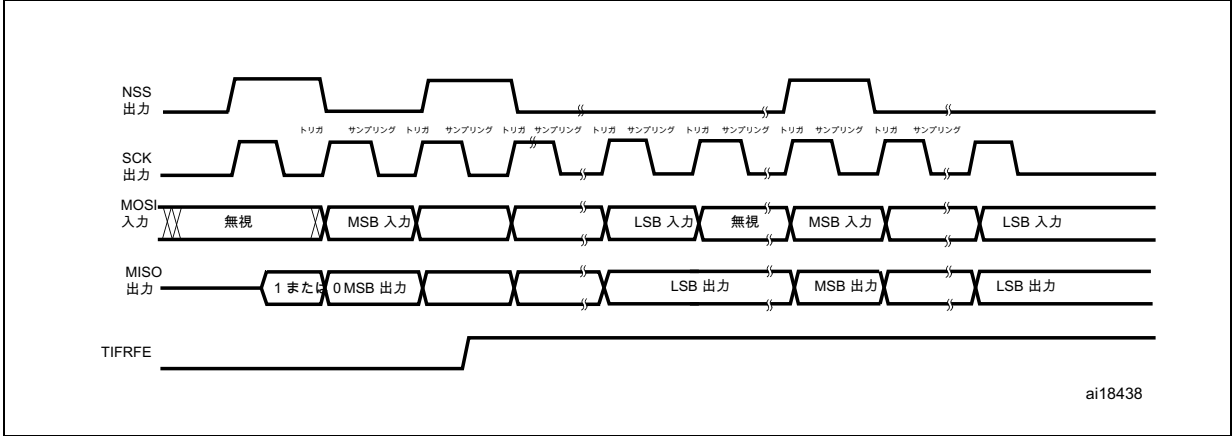
このフラグを使用して、SPI_CR1 レジスタの CRCEN ビットがセットされているときに受信された値の有効性を検証します。シフトレジスタに受信された値が、レシーバである SPI_RXCRCR の値と一致しなかった場合、SPI_SR レジスタの CRCERR フラグがセットされます。

TI モードフレームフォーマットエラー

SPI がスレーブモードとして機能し、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、FRE フラグが SPI_SR レジスタでセットされます。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、データは破損した可能性があります。

FRE フラグは、SPI_SR レジスタを読み出すとクリアされます。ERRIE ビットがセットされていると、NSS エラー検出時に割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。

図 207. TI モードフレームフォーマットエラー検出



20.3.11 SPI 割込み

表 90. SPI 割込みリクエスト

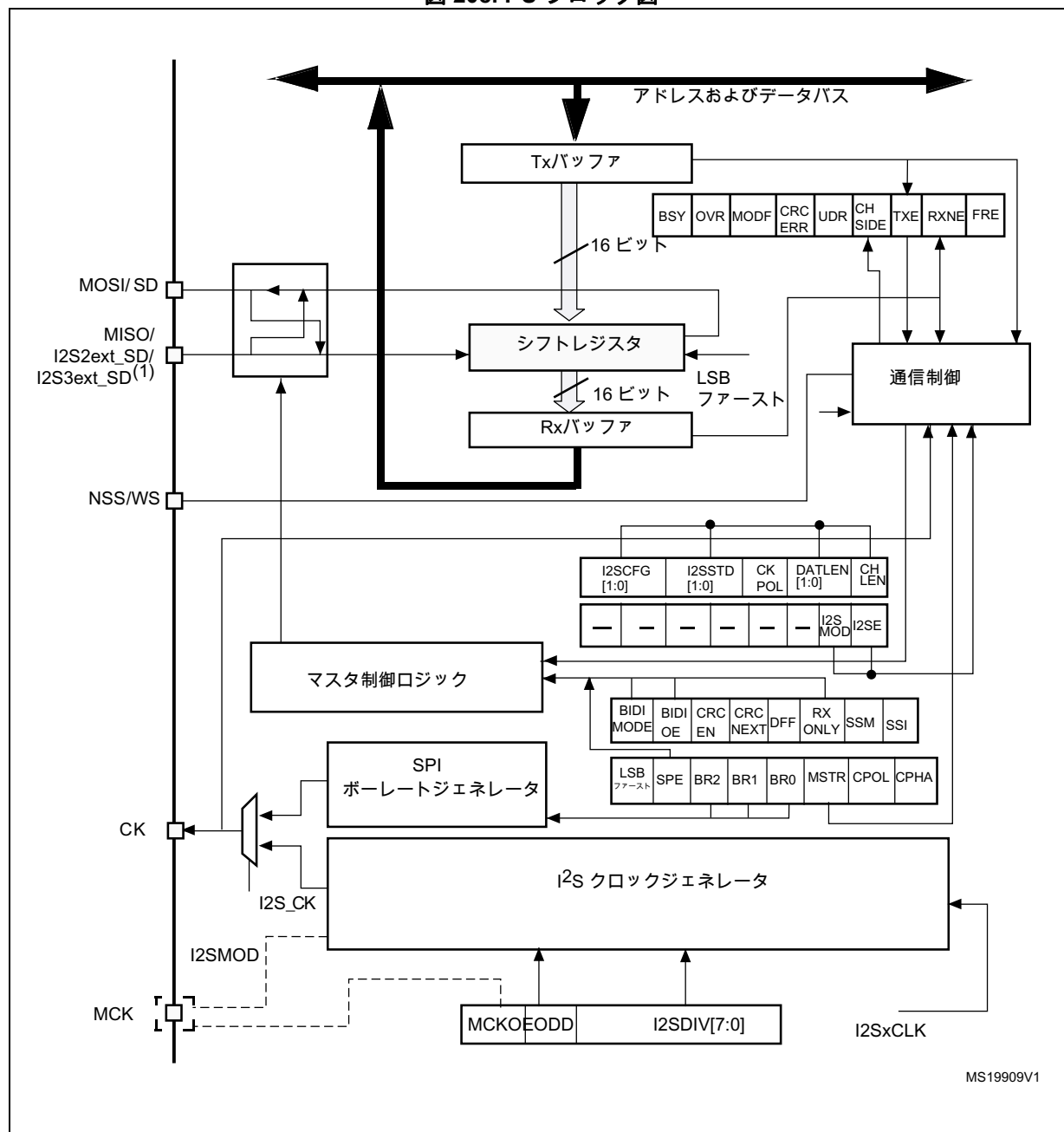
割込みイベント	イベントフラグ	イネーブル制御ビット
送信バッファエンプティフラグ	TXE	TXEIE
受信バッファノットエンプティフラグ	RXNE	RXNEIE
マスタモードフォールトイベント	MODF	ERRIE
オーバーランエラー	OVR	
CRC エラーフラグ	CRCERR	
TI フレームフォーマットエラー	FRE	ERRIE

20.4 I²S の機能説明

20.4.1 I²S の概要

I²S のブロック図を図 208 に示します。

図 208. I²S ブロック図



1. I2S2ext_SD と I2S3ext_SD は、I²S 全二重モードを制御する拡張 SD ピンです。

SPI はオーディオ I²S インタフェースとして機能することができます。それには、SPI_I2SCFGR レジスタの I2SMOD ビットをセットして I²S 機能を有効にしてください。このインタフェースは、SPI とほぼ同じピン、フラグ、および割り込みを使用します。

I²S と SPI は、以下の3 つのピンを共用します。

- SD : MOSI ピンに配置され、2 つの時間多重化データチャネルを送受信します（半二重モードのみ）。
- WS : NSS ピンに配置され、マスタモードではデータ制御信号の出力、スレーブモードでは入力です。
- CK : SCK ピンに配置され、マスタモードではシリアルクロック出力、スレーブモードではシリアルクロック入力です。
- 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 : I²S がマスタモードに設定されている（かつ、SPI_I2SPR レジスタの MCKOE ビットがセットされている）とき、周波数 $256 \times F_S$ （ F_S はオーディオサンプリング周波数）で生成されたこの追加クロックを出力するために別々に配置されたマスタクロックが使用されます。

I²S は、マスタモードに設定されているとき、専用のクロックジェネレータを使用して通信クロックを生成します。このクロックジェネレータは、マスタクロック出力のソースでもあります。I²S モードでは、2 つの追加レジスタを使用できます。1 つはクロックジェネレータ設定 SPI_I2SPR にリンクされ、もう 1 つは汎用 I²S 設定レジスタ SPI_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。

SPI_CR1 レジスタとすべての CRC レジスタは、I²S モードでは使用されません。同様に、SPI_CR2 レジスタの SSOE ビットと SPI_SR の MODF および CRCERR ビットも使用されません。

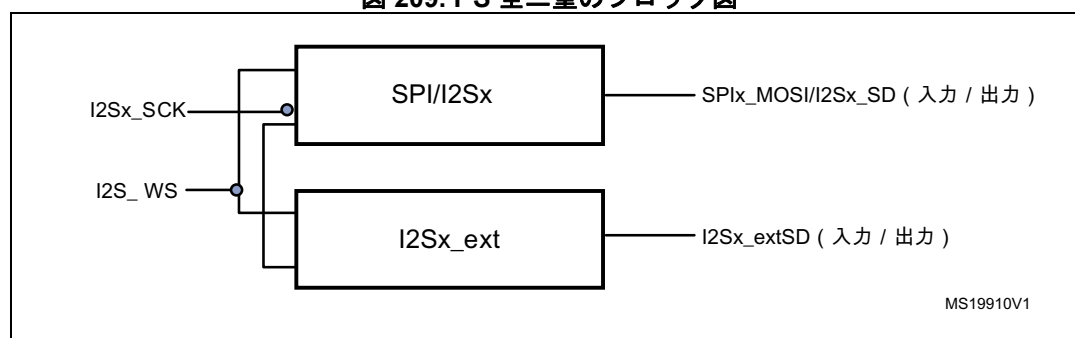
I²S は、16 ビット幅モードでのデータ転送に同じ SPI レジスタ（SPI_DR）を使用します。

20.4.2 I²S 全二重

I²S 全二重 モードをサポートするため、I2S2 と I2S3 の他に拡張 I²S（I2S2_ext、I2S3_ext）と呼ぶ 2 つの追加 I²S インスタンスが利用可能です（図 209 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 209. 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。

20.4.3 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、注目する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 SPI_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
- 32 ビットフレームにパックされた 16 ビットデータ
- 32 ビットフレームにパックされた 24 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、SPI_DR との間で 2 つの CPU 読み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 つの DMA 動作）を必要とします。24 ビットのデータフレームの場合、8 つの下位ビットはハードウェアによって 0 のビットで 32 ビットに拡張されます。

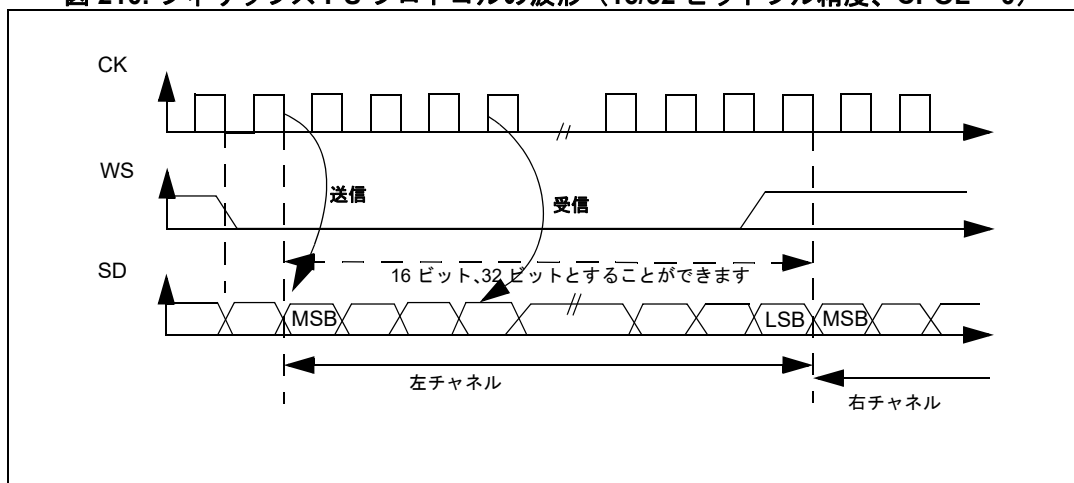
すべてのデータフォーマットと通信規格に対して、最上位ビットは常に最初に送信されます（MSB ファースト）。

I²S インタフェースは、SPI_I2SCFGR レジスタの I2SSTD[1:0] と PCMSYNC ビットを使用して設定可能な 4 つのオーディオ規格をサポートします。

フィリップス I²S 規格

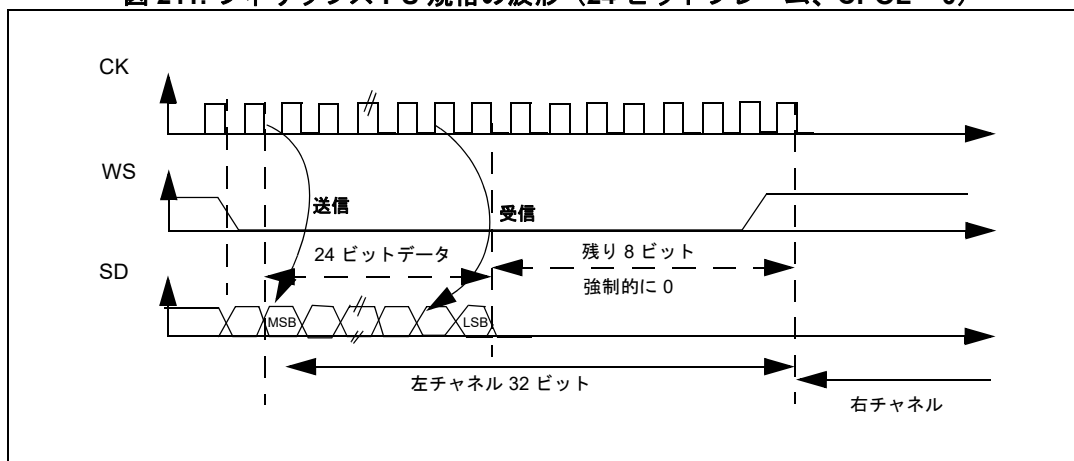
この規格では、どのチャンネルが送信されているかを示すために WS 信号を使用します。この信号が有効になってから 1 CK クロックサイクル後に最初のビット（MSB）が使用可能になります。

図 210. フィリップス I²S プロトコルの波形 (16/32 ビットフル精度、CPOL = 0)



データは、CK の立ち下がりエッジでラッチされ (トランスミッタの場合)、立ち上がりエッジで読み出されます (レシーバの場合)。WS 信号も CK の立ち下がりエッジでラッチされます。

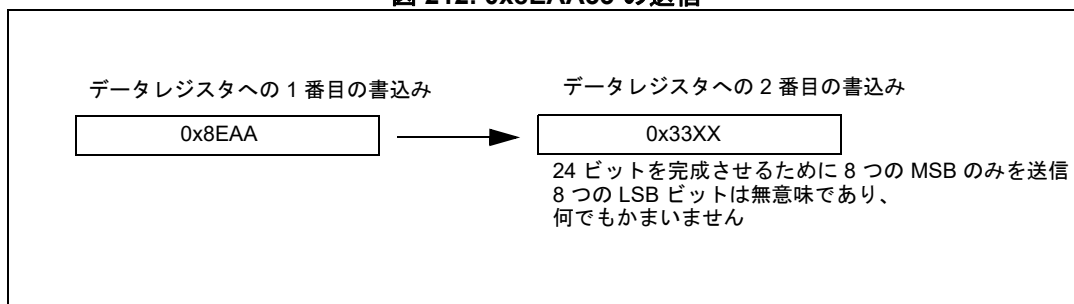
図 211. フィリップス I²S 規格の波形 (24 ビットフレーム、CPOL = 0)



このモードでは、SPI_DR に対して 2 つ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 212. 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 213. 0x8EAA33 の受信

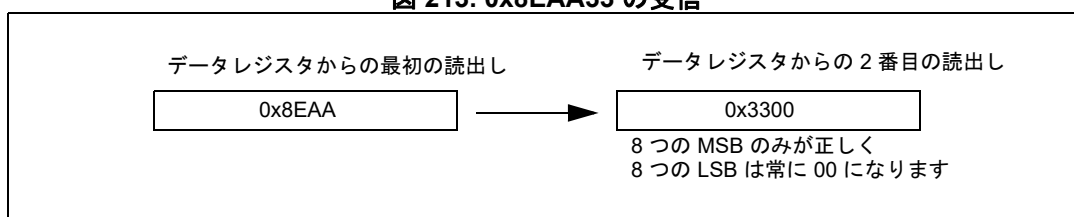
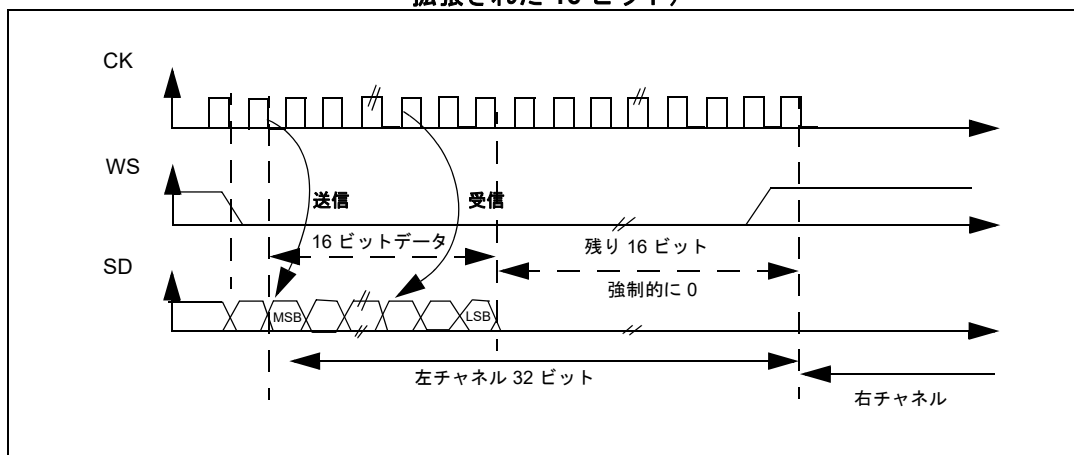


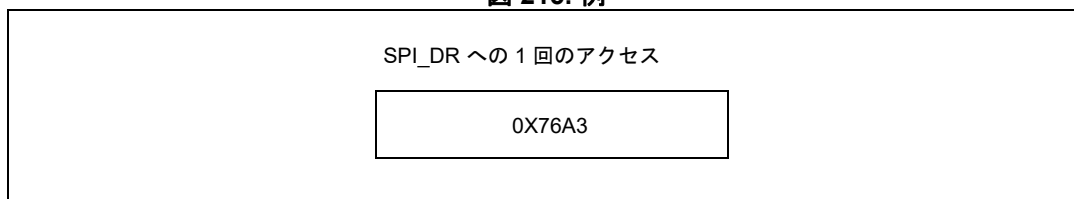
図 214. フィリップス I²S 規格 (CPOL=0で32 ビットパケットフレームに拡張された 16 ビット)



I²S 設定フェーズで、32 ビットチャネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPI_DR 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 215](#) に示す動作が要求されます。

図 215. 例



送信では、SPI_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、SPI_DR に新しい送信値をロードするために割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いとしても行われます。

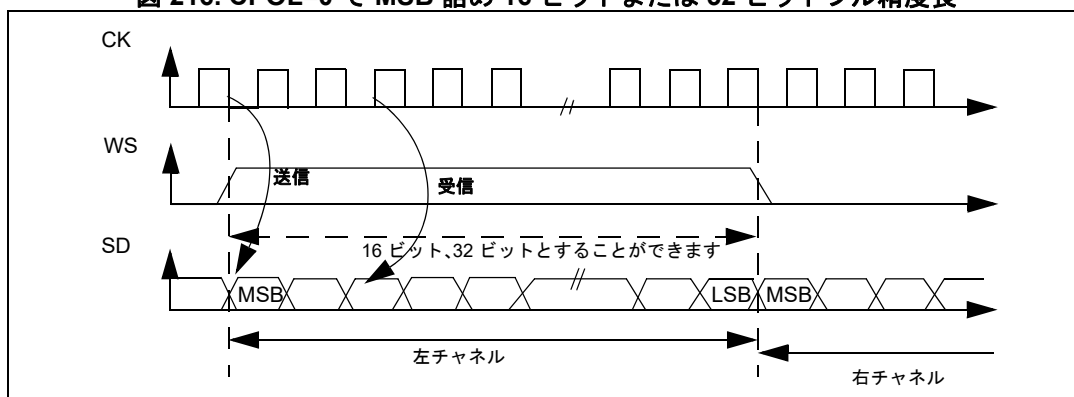
受信では、RXNE フラグがセットされ、可能ならば、最初の 16 MSB ハーフワードの受信時にその割込みが生成されます。

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 216. CPOL=0 で MSB 詰め 16 ビットまたは 32 ビットフル精度長



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 217. CPOL=0 で MSB 詰め 24 ビットフレーム長

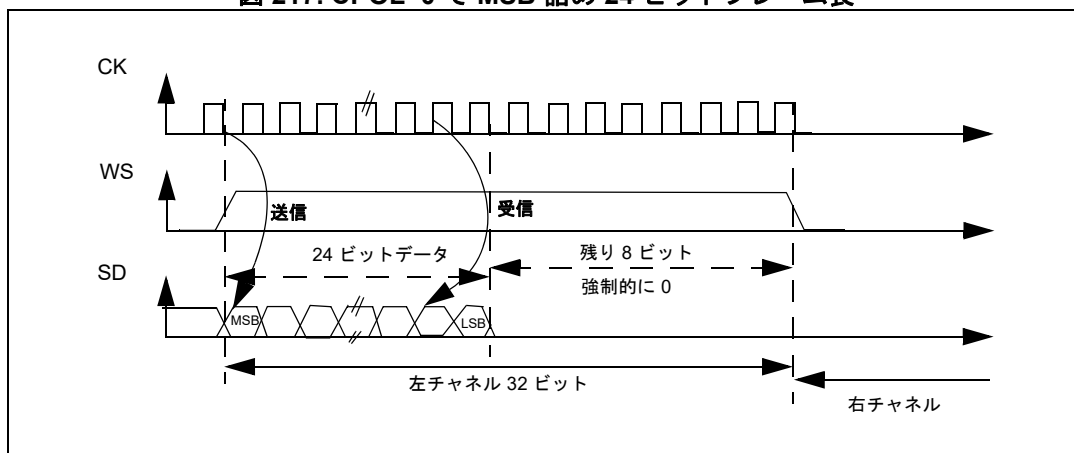
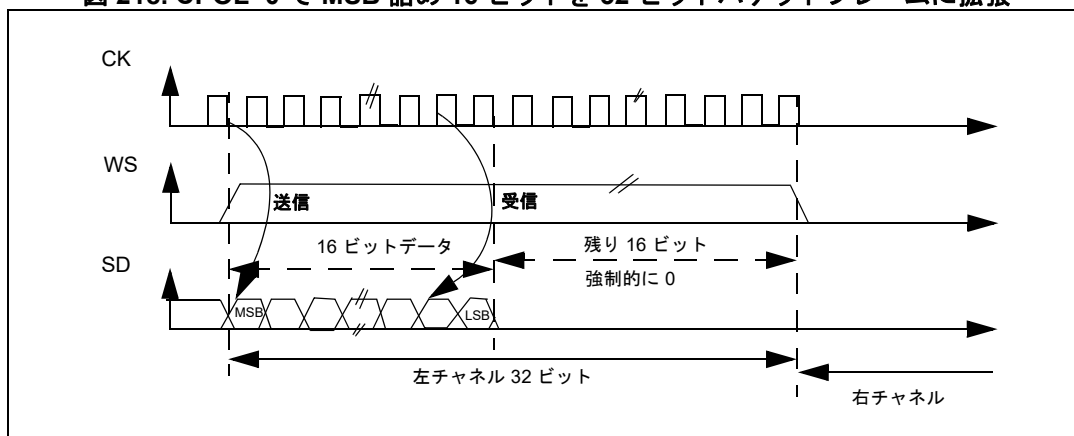


図 218. CPOL=0 で MSB 詰め 16 ビットを 32 ビットパケットフレームに拡張



LSB 詰め規格

この規格は、MSB 詰め規格と似ています (16 ビットと 32 ビットのフル精度フレームフォーマットに違いはありません)。

図 219. CPOL=0 で LSB 詰め 16 ビットまたは 32 ビットフル精度

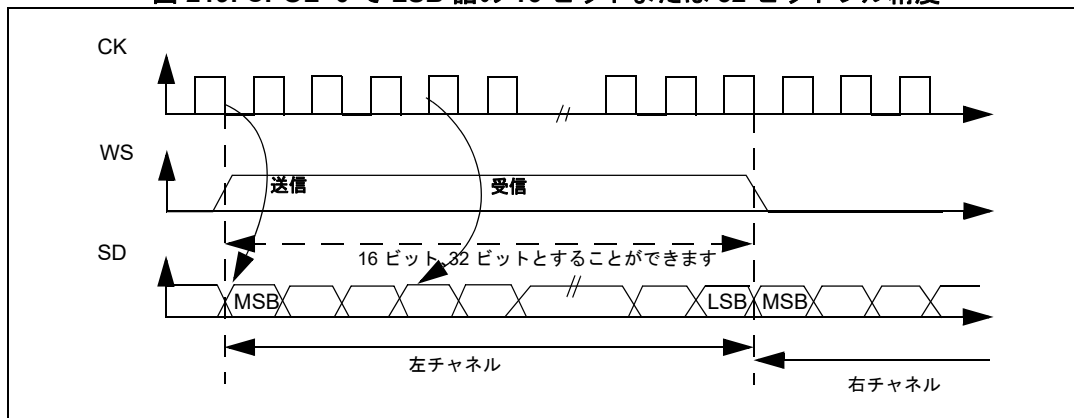
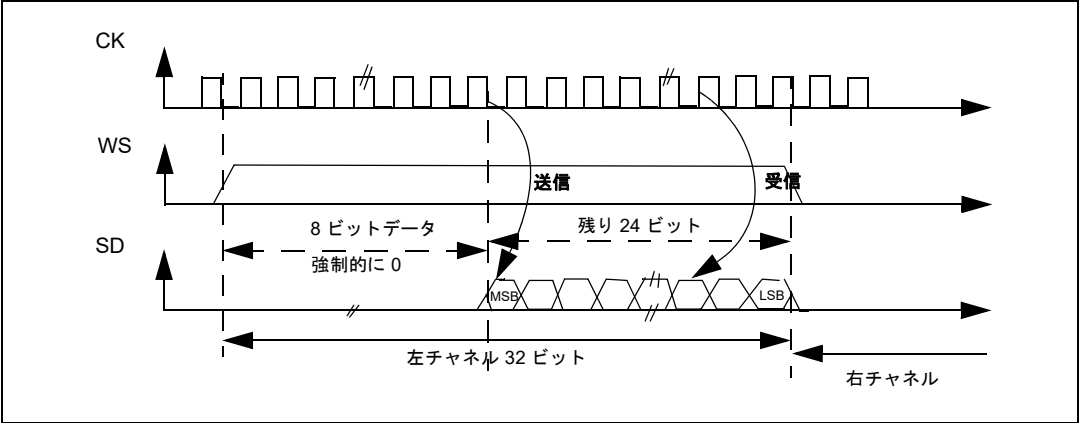
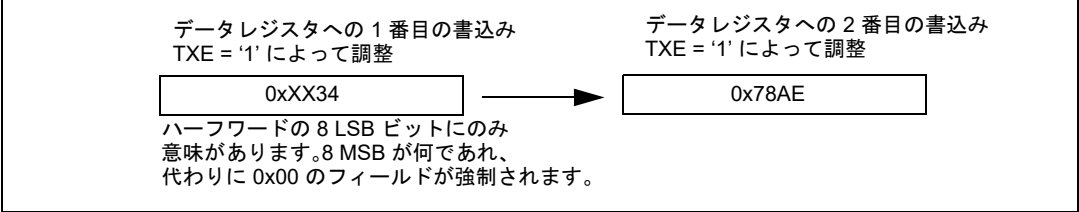


図 220. CPOL=0 で LSB 詰め 24 ビットフレーム長



- 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 SPI_DR レジスタへの 2 つの書き込み動作が必要です。この動作を次に示します。

図 221. 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 SPI_DR から連続する 2 つの読み出し動作が必要です。

図 222. 0x3478AE の受信に必要な動作

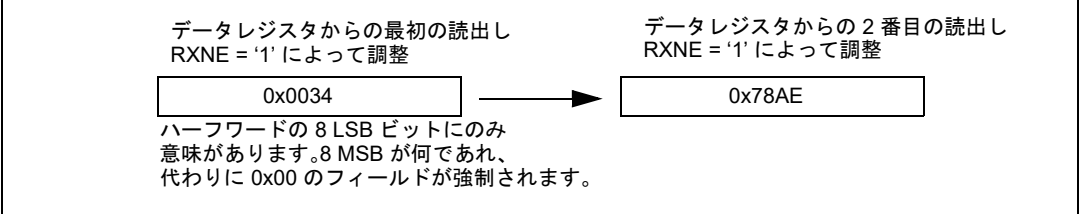
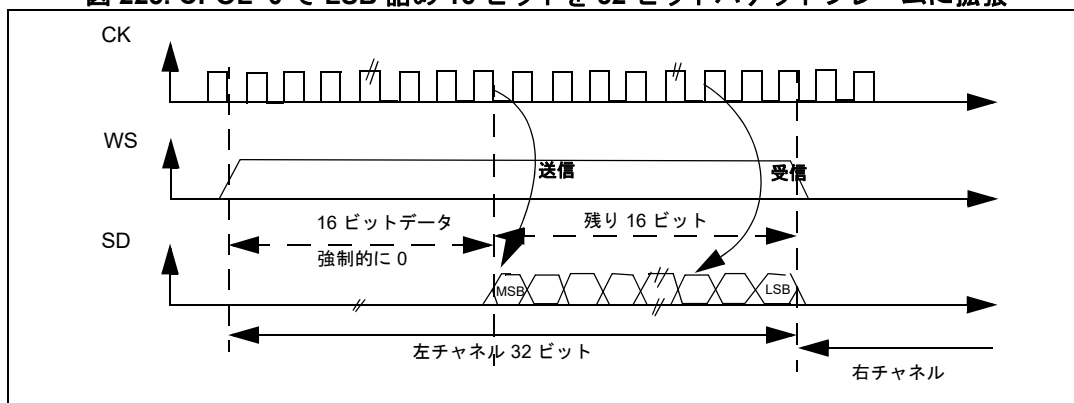


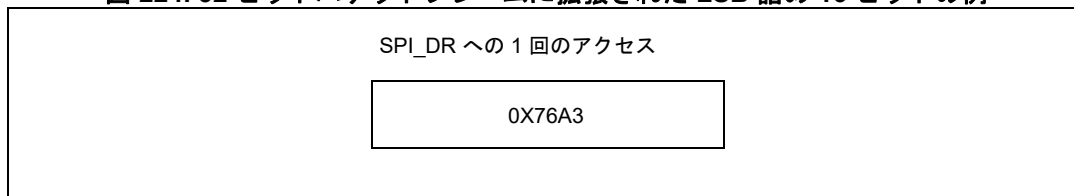
図 223. CPOL=0 で LSB 詰め 16 ビットを 32 ビットパケットフレームに拡張



I²S 設定フェーズで、32 ビットチャネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPI_DR 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 224](#) に示す動作が要求されます。

図 224. 32 ビットパケットフレームに拡張された LSB 詰め 16 ビットの例



送信モードでは、TXE がアサートされると、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 が再びアサートされます。

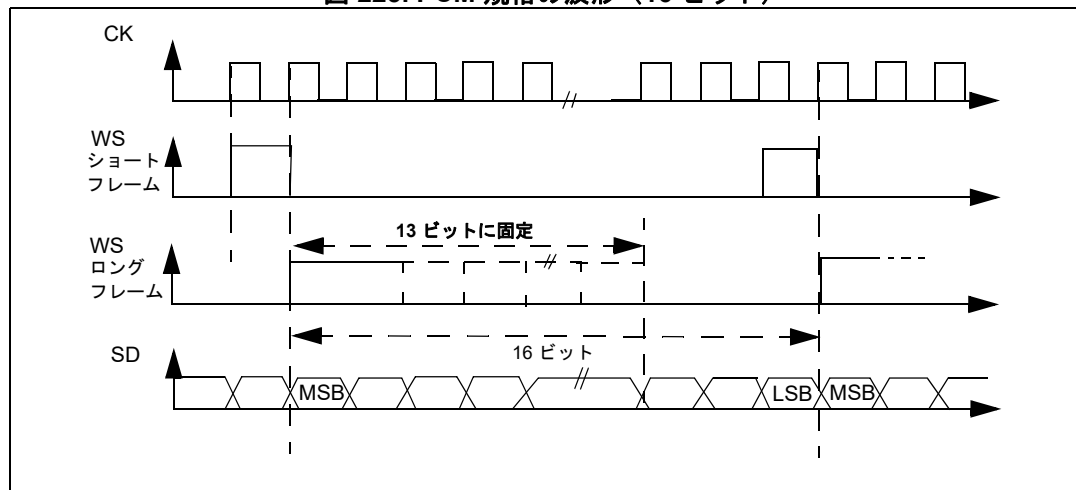
受信モードでは、0x0000 フィールドではない有効なハーフワードが受信されると、すぐに RXNE がアサートされます。

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。SPI_I2SCFGR 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）を使用し、設定可能です。

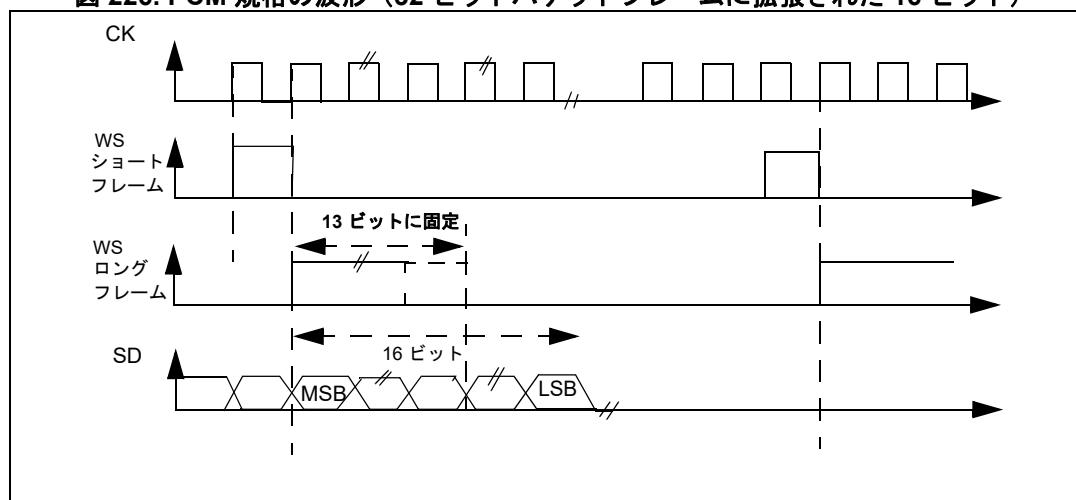
図 225. PCM 規格の波形（16 ビット）



ロングフレーム同期では、WS 信号のアサーション時間はマスタモードで 13 ビットに固定されています。

ショートフレーム同期では、WS 同期信号の長さは、わずか 1 サイクルです。

図 226. 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（SPI_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。

20.4.4 クロックジェネレータ

I²S ビットレートにより、I²S データライン上のデータフローと I²S クロック信号周波数が決まります。

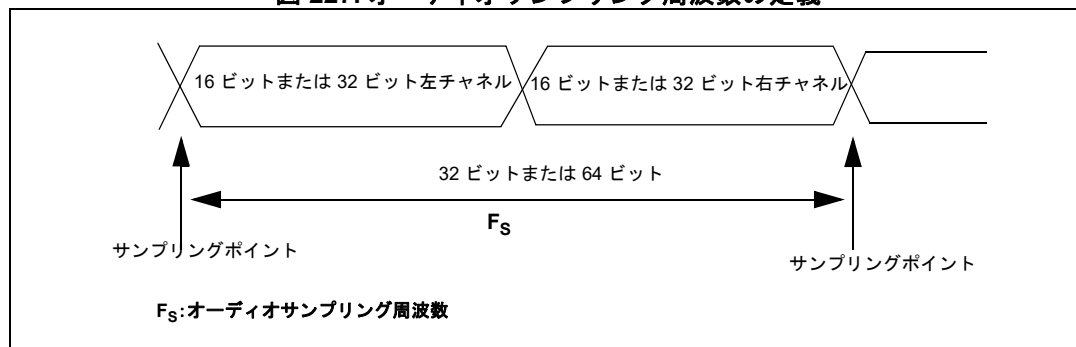
I²S ビットレート = チャンネルあたりのビット数 × チャンネル数 × オーディオサンプリング周波数

16 ビットオーディオ、左チャンネルおよび右チャンネルの場合、I²S ビットレートは次のように算出されます。

$$I^2S \text{ ビットレート} = 16 \times 2 \times F_S$$

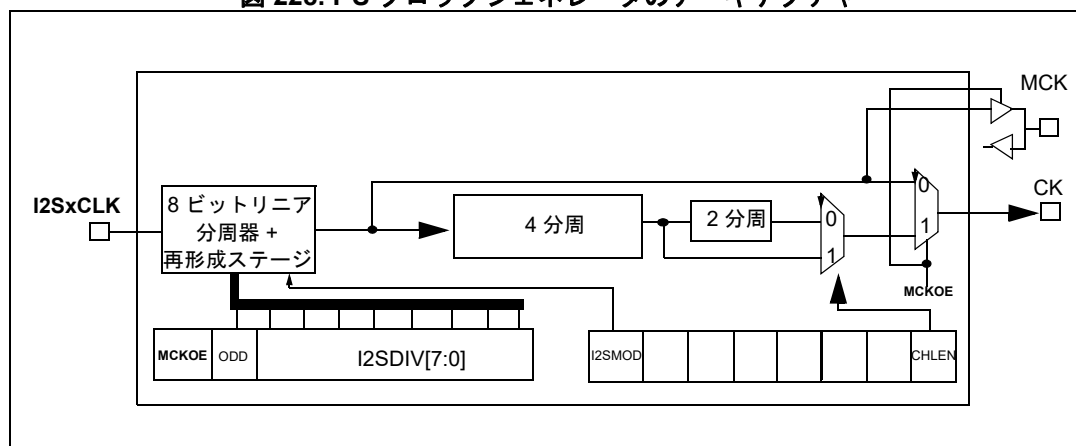
パケット長が 32 ビットの場合、I²S ビットレート = 32 × 2 × F_S となります。

図 227. 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 228. I²S 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 227 に、通信クロックのアーキテクチャを示します。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、PLL I²S 出力（分周比 R を使用）または外部クロック（I2S_CKIN ピンに割り付け）を使用することができます。

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（SPI_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)

$$F_S = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャネルフレームが 16 ビット幅の場合}$$

$$F_S = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャネルフレームが 32 ビット幅の場合}$$

表 91 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 91. オーディオ周波数精度 (PLLM VCO = 1 MHz または 2 MHz の場合)⁽¹⁾

マスタ クロック	ターゲット f _s (Hz)	データ フォーマット	PLLI2SN	PLLI2SR	I2SDIV	I2SODD	実際の f _s (Hz)	誤差
無効	8000	16 ビット	192	2	187	1	8000	0.0000%
		32 ビット	192	3	62	1	8000	0.0000%
	16000	16 ビット	192	3	62	1	16000	0.0000%
		32 ビット	256	2	62	1	16000	0.0000%
	32000	16 ビット	256	2	62	1	32000	0.0000%
		32 ビット	256	5	12	1	32000	0.0000%
	48000	16 ビット	192	5	12	1	48000	0.0000%
		32 ビット	384	5	12	1	48000	0.0000%
	96000	16 ビット	384	5	12	1	96000	0.0000%
		32 ビット	424	3	11	1	96014.49219	0.0151%
	22050	16 ビット	290	3	68	1	22049.87695	0.0006%
		32 ビット	302	2	53	1	22050.23438	0.0011%
	44100	16 ビット	302	2	53	1	44100.46875	0.0011%
		32 ビット	429	4	19	0	44099.50781	0.0011%
有効	8000	16 ビット	424	3	11	1	192028.9844	0.0151%
		32 ビット	258	3	3	1	191964.2813	0.0186%
	8000	無視	256	5	12	1	8000	0.0000%
	16000	無視	213	2	13	0	16000.60059	0.0038%
	32000	無視	213	2	6	1	32001.20117	0.0038%
	48000	無視	258	3	3	1	47991.07031	0.0186%
	96000	無視	344	2	3	1	95982.14063	0.0186%
	22050	無視	429	4	9	1	22049.75391	0.0011%
	44100	無視	271	2	6	0	44108.07422	0.0183%

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。

20.4.5 I²S マスタモード

I²S は次のように設定することができます：

- 送信用または受信用のマスタモード (I2Sx を使用した半二重モード)
- 送受信用のマスタモード (I2Sx および I2Sx_ext を使用した全二重モード)。

つまり、シリアルクロックは、ワードセレクト信号 WS だけでなく、CK ピン上でも生成されます。マスタクロック (MCK) は、SPI_I2SPR レジスタの MCKOE ビットで、出力にすることも、出力にしないこともできます。

手順

1. 適切なオーディオサンプリング周波数に到達するシリアルクロックボーレートを定義するため、SPI_I2SPR レジスタの I2SDIV[7:0] ビットを選択します。SPI_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、SPI_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 20.4.4：クロックジェネレータ](#)を参照)。
3. SPI_I2SCFGR の I2SMOD ビットをセットして I²S 機能を有効にし、I2SSTD[1:0] と PCMSYNC ビットにより I²S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してチャンネルあたりのビット数を、それぞれ選択します。SPI_I2SCFGR レジスタの I2SCFG[1:0] ビットにより I²S マスタモードと方向 (トランスミッタまたはレシーバ) も選択します。
4. 必要ならば、SPI_CR2 レジスタへの書込みによって、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. SPI_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。SPI_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。

想定上、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、SPI_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

選択された I²S 規格モードに応じた書込み動作の詳細については、[セクション 20.4.3：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 SPI_DR に書き込むことが必要です。

I2SE をクリアすることによって I²S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

受信シーケンス

動作モードは、ポイント 3 を除いて、送信モードの場合と同じです（[セクション 20.4.5 : I²S マスタモード](#)に示す手順を参照）。ポイント 3 では、I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットの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 RXNE フラグがセットされ、SPI_CR2 レジスタの RXNEIE ビットがセットされている場合は割込みが生成されます。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、SPI_DR レジスタの読出しによってクリアされます。

CHSIDE は毎回の受信後に更新されます。CHSIDE は、I²S セルによって生成される WS 信号に反応します。

選択された I²S 規格モードに応じた読出し動作の詳細については、[セクション 20.4.3 : 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。SPI_CR2 レジスタの ERRIE ビットがセットされている場合、割込みが生成されてエラーを知らせます。

I²S をスイッチオフするには、I²S が新しいデータ転送を開始することなく転送サイクルを適切に完了できるように、特定の動作が要求されます。そのシーケンスは、データ長とチャンネル長の設定、および選択したオーディオプロトコルモードに依存します。ケース別の説明

- 32 ビットチャンネル長に拡張された 16 ビットデータ長 (DATLEN = 00、CHLEN = 1)、LSB 詰めモードを使用 (I2SSTD = 10)
 - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
 - b) 次に I²S の 17 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。
- 32 ビットチャンネル長に拡張された 16 ビットデータ長 (DATLEN = 00 および CHLEN = 1)、MSB 詰め、I²S または PCM モード (それぞれ、I2SSTD = 00、I2SSTD = 01、または I2SSTD = 11)
 - a) 最後の RXNE を待ちます。
 - b) 次に I²S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。
- DATLEN と CHLEN のその他すべての組み合わせについては、I2SSTD ビットを通じて選択したオーディオモードが何であれ、次のシーケンスを実行して I²S をスイッチオフします。
 - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
 - b) 次に I²S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。

注 : 転送時、BSY フラグはローレベルに保持されます。

20.4.6 I²S スレーブモード

I²S は次のように設定することができます：

- 送信用または受信用のスレーブモード (I2Sx を使用した半二重モード)
- 送受信用のスレーブモード (I2Sx および I2Sx_ext を使用した全二重モード)

動作モードは、主として I²S マスタ設定に関して述べたものと同じ規則に従います。スレーブモードでは、I²S インタフェースによって生成されるクロックはありません。クロックと WS 信号は、I²S インタフェースに接続された外部マスタから入力されます。したがって、ユーザがクロックを設定する必要はありません。

設定ステップを次に示します。

1. SPI_I2SCFGR レジスタの I2SMOD ビットをセットして I²S 機能に到達し、I2SSTD[1:0] ビットにより I²S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してフレームのチャンネルあたりのビット数を、それぞれ選択します。SPI_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、SPI_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込みソースと DMA 機能を選択します。
3. SPI_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

送信シーケンス

送信シーケンスは、外部マスタデバイスがクロックを送信したときと、NSS_WS 信号がデータの転送を要求したときに開始されます。スレーブを有効にしなければ、外部マスタは通信を開始しません。I²S データレジスタは、マスタが通信を開始する前に、ロードする必要があります。

I²S、MSB 詰め、および LSB 詰めモードの場合、データレジスタに書き込まれる最初のデータは、左チャンネルのデータに対応します。通信が開始されると、データはTxバッファからシフトレジスタに転送されます。次に、右チャンネルのデータを I²S データレジスタに書き込むように要求するために、TXE フラグがセットされます。

CHSIDE フラグは、どのチャンネルが送信されるかを示します。スレーブモードでは、マスタ送信モードに比べて、CHSIDE は外部マスタからの WS 信号に反応します。つまり、スレーブが最初のデータの送信準備をしなければ、マスタはクロックを生成できません。WS アサーションは、最初に送信される左チャンネルに対応します。

注： I2SE は、マスタの最初のクロックが CK ラインに到達するよりも、少なくとも 2 PCLK サイクル前に書き込まれる必要があります。

データハーフワードは、最初のビット送信時に内部バスから 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPI_CR2 レジスタの TXEIE ビットがセットされている場合は割込みが生成されます。

なお、Txバッファへの書き込みの前に、TXE フラグが 1 であることを確認する必要があります。

選択された I²S 規格モードに応じた書き込み動作の詳細については、[セクション 20.4.3：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 SPI_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 SPI_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。SPI_CR2 レジスタの ERRIE ビットがセットされた場合、SPI_SR レジスタの UDR フラグがハイレベルになると割込みが生成されます。この場合、I²S をスイッチオフし、左チャ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 ビットをクリアすることによって I²S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

受信シーケンス

動作モードは、ポイント 1 を除いて、送信モードの場合と同じです（[セクション 20.4.6 : I²S スレーブモード](#)に示す手順を参照）。ポイント 1 では、SPI_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 packets によって受信されます。つまり、Rx バッファがフルになるたびに SPI_SR レジスタの RXNE フラグがセットされ、SPI_CR2 レジスタの RXNEIE ビットがセットされている場合は割込みが生成されます。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、SPI_DR 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、SPI_DR レジスタの読み出しによってクリアされます。

選択された I²S 規格モードに応じた読み出し動作の詳細については、[セクション 20.4.3 : 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。SPI_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込みが生成されます。

受信モードで I²S をスイッチオフするには、最後の RXNE = 1 を受信した直後に I2SE をクリアする必要があります。

注： 外部マスタコンポーネントには、オーディオチャンネルを介して 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。

20.4.7 ステータスフラグ

アプリケーションが I²S バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット／クリアされます（このフラグへの書き込みは無効）。このフラグは I²S の通信層の状態を示します。

BSY がセットされると、I²S が通信中でビジーであることを示します。マスタ受信モード (I2SCFG=11) には 1 つの例外があり、BSY フラグは受信時にローレベルに保持されます。

ソフトウェアが I²S を無効にする必要がある場合、BSY フラグは転送の終わりを検出するために役立ちます。これによって、最後の転送データの破壊を回避します。このため、下記の手順を厳守する必要があります。

BSY フラグは転送が開始されるとセットされます。ただし、I²S がマスタ受信モードにある場合を除きます。

BSY フラグをクリアするタイミング

- 転送が完了したとき（ただし、マスタ送信モードの場合は通信は連続的であると考えられます）
- I²S が無効化されたとき

通信が連続的な場合

- マスタ送信モードでは、BSY フラグはすべての転送期間を通じてハイレベルに保持されます。
- スレーブモードでは、BSY フラグは、各転送間で I²S の 1 クロックサイクルの間ローレベルになります。

注： 各データの送受信の処理には BSY フラグを使用しないでください。代わりに、TXE フラグと RXNE フラグを使用することをお勧めします。

Tx バッファエンプティフラグ (TXE)

このフラグがセットされると、Tx バッファはエンプティ（空）であり、次に送信するデータをバッファにロードできることを示します。送信されるデータがすでに Tx バッファに格納されているとき、TXE フラグはリセットされます。TXE フラグは、I²S が無効にされている (I2SE ビットがリセット) ときにもリセットされます。

Rx バッファノートエンプティ (RXNE)

このフラグがセットされると、Rx バッファに有効な受信データがあることを示します。このフラグは、SPI_DR レジスタが読み出されるとリセットされます。

チャネルサイドフラグ (CHSIDE)

送信モードでは、このフラグは TXE がハイレベルになるとリフレッシュされます。このフラグは、SD 上の転送データが属するチャネルサイドを示します。スレーブ送信モードでアンダーランエラーイベントが発生した場合、このフラグは信頼できないため、通信を再開する前に、I²S をスイッチオフし、さらにスイッチオンする必要があります。

受信モードでは、このフラグは SPI_DR にデータが受信されるとリフレッシュされます。このフラグは、どちらのチャネルサイドからデータが受信されたかを示します。なお、エラー (OVR など) が発生した場合、このフラグは無意味になるため、I²S を無効にし、さらに有効にすることによってリセットする必要があります（変更が必要な場合は設定する）。

このフラグは、PCM 規格では意味を持ちません（ショートとロングフレームの両モード）。

SPI_SR の OVR または UDR フラグがセットされ、SPI_CR2 の ERRIE ビットもセットされると、割込みが生成されます。この割込みをクリアするには、割込みソースをいったんクリアした後で、SPI_SR ステータスレジスタを読み出します。

20.4.8 エラーフラグ

I²S セルには 3 つのエラーフラグがあります。

アンダーランフラグ (UDR)

スレーブ送信モードでは、ソフトウェアが SPI_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われると、このフラグがセットされます。このフラグは、SPI_I2SCFGR の I2SMOD ビットがセットされると使用できます。SPI_CR2 の ERRIE ビットがセットされた場合、割込みが生成されることがあります。

UDR ビットは、SPI_SR レジスタの読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 SPI_DR 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。SPI_CR2 の ERRIE ビットがセットされた場合、割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。SPI_DR レジスタへの読み出し動作によって、以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、SPI_DR レジスタを読み出し、続けて SPI_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、I²S がスレーブモードに設定された場合にのみハードウェアによってセットすることができます。このフラグは、スレーブとして WS ラインの変化が想定されないときに外部マスタが WS ラインを変化させようとしている場合にセットされます。同期が失われた場合、この状態から回復し、外部マスタデバイスと I²S スレーブデバイスを再同期させるするためには、次のステップに従います。

1. I²S を無効化します。
2. 正しいレベルが WS ラインで検出されたとき (WS ラインは I²S モードではハイレベル、MSB 詰めまたは LSB 詰めモードあるいは PCM モードではローレベル)、再度有効化します。

マスタデバイスとスレーブデバイスの間の同期外れは、SCK 通信クロック上または WS フレーム同期ライン上のノイズの多い環境に起因する可能性があります。ERRIE ビットをセットすれば、エラー割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。

20.4.9 I²S 割込み

表 92 に I²S 割込みのリストを示します。

表 92. I²S 割込みリクエスト

割込みイベント	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	ERRIE

20.4.10 DMA の機能

DMA は SPI モードとまったく同じように機能します。I²S の場合も違いはありません。I²S モードではデータ転送保護システムがないため、CRC 機能を使用できません。

20.5 SPI および I²S 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、[34 ページのセクション 1.1](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスできます。

20.5.1 SPI 制御レジスタ 1 (SPI_CR1) (I²S モードでは使用しません)

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 15 **BIDIMODE** : 双方向データモードイネーブル
0 : 2 線単方向データモードを選択します。
1 : 1 線双方向データモードを選択します。
注 : このビットは I²S モードでは使用しません。
- ビット 14 **BIDIOE** : 双方向モードでの出カイネーブル
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 : このビットは I²S モードでは使用しません。
MISO ピンがスレーブモードで使用されている間、マスタモードでは、**MOSI** ピンが使用されます。
- ビット 13 **CRCEN** : ハードウェア CRC 計算イネーブル
0 : CRC 計算は無効です。
1 : CRC 計算は有効です。
注 : 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。
このフラグは I²S モードでは使用しません。
- ビット 12 **CRCNEXT** : CRC 転送指示
0 : データフェーズです (CRC フェーズなし)。
1 : 次の転送は CRC です (CRC フェーズ)。
注 : SPI が全二重モードまたは送信専用モードに設定されている場合、最後のデータが SPI_DR に書き込まれた直後に、CRCNEXT ビットを書き込む必要があります。
SPI が受信専用モードに設定されている場合、最後から 2 番目のデータが受信された後に CRCNEXT ビットをセットする必要があります。
転送が DMA で管理される場合、このビットはクリアされた状態に保持する必要があります。
このフラグは I²S モードでは使用しません。
- ビット 11 **DFF** : データフレームフォーマット
0 : 送受信に 8 ビットデータフレームフォーマットが選択されます。
1 : 送受信に 16 ビットデータフレームフォーマットが選択されます。
注 : 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。
このフラグは I²S モードでは使用しません。

ビット 10 RXONLY : 受信専用

2 線単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。このビットはマルチスレーブシステムでも役立ちます。そのシステムでは、この特定のスレーブはアクセスされず、アクセスされたスレーブからの出力は破壊されません。

0 : 全二重 (送受信)

1 : 出力は無効です (受信専用モード)。

注： このビットは I²S モードでは使用しません。

ビット 9 SSM : ソフトウェアスレーブ管理

SSM ビットがセットされているとき、NSS ピンの入力値は 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 I²S モードおよび SPI TI モードでは使用しません。

ビット 8 SSI : 内部スレーブ選択

このビットは、SSM ビットがセットされているときにのみ有効です。このビットの値は強制的に NSS ピンに設定され、NSS ピンの IO 値は無視されます。

注： このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 7 LSBFIRST : フレームフォーマット

0 : MSB が最初に送信されます。

1 : LSB が最初に送信されます。

注： このビットは、通信中には変更しないでください。

このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 6 SPE : SPI イネーブル

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

注： このビットは I²S モードでは使用しません。

SPI を無効にするときは、[セクション 20.3.8 : SPI の無効化](#)に記載されている手順に従ってください。

ビット 5:3 BR[2:0] : ボーレート制御

000 : $f_{PCLK}/2$

001 : $f_{PCLK}/4$

010 : $f_{PCLK}/8$

011 : $f_{PCLK}/16$

100 : $f_{PCLK}/32$

101 : $f_{PCLK}/64$

110 : $f_{PCLK}/128$

111 : $f_{PCLK}/256$

注： これらのビットは、通信中には変更しないでください。

これらのビットは I²S モードでは使用されません。

- ビット 2 **MSTR** : マスタ選択
- 0 : スレーブ設定
 - 1 : マスタ設定
- 注: このビットは、通信中には変更しないでください。
このフラグは I²S モードでは使用しません。
- ビット 1 **CPOL** : クロック極性
- 0 : アイドル時に CK は 0 となります。
 - 1 : アイドル時に CK は 1 となります。
- 注: このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。
- ビット 0 **CPHA** : クロック位相
- 0 : 最初のクロック遷移が最初のデータキャプチャエッジです。
 - 1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。
- 注: このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。

20.5.2 SPI 制御レジスタ 2 (SPI_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								TXEIE	RXNEIE	ERRIE	FRF	Res.	SSOE	TXDMAEN	RXDMAEN
								rw	rw	rw	rw		rw	rw	rw

- ビット 15:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **TXEIE** : Txバッファエンプティ割込みイネーブル
- 0 : TXE 割込みはマスクされます。
 - 1 : TXE 割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込みイネーブル
- 0 : RXNE 割込みはマスクされます。
 - 1 : RXNE 割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 ; I²S モードでは UDR、OVR、FRE)、割込みの生成を制御します。
- 0 : エラー割込みはマスクされます。
 - 1 : エラー割込みは有効です。
- ビット 4 **FRF** : フレームフォーマット
- 0 : SPI モトローラモード
 - 1 : SPI TI モード
- 注: このビットは I²S モードでは使用しません。
- ビット 3 予約済み。ハードウェアによって 0 に固定されています。

ビット 2 **SSOE** : SS 出力有効

- 0 : マスタモードで SS 出力は無効にされ、セルはマルチマスタ設定で機能できます。
- 1 : セルが有効であるとき、マスタモードで SS 出力は有効です。セルはマルチマスタ環境では機能できません。

注： このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 1 **TXDMAEN** : Txバッファ DMA イネーブル

- このビットがセットされると、TXE フラグがセットされるたびに DMA リクエストが行われます。
- 0 : Txバッファ DMA は無効です。
- 1 : Txバッファ DMA は有効です。

ビット 0 **RXDMAEN** : Rxバッファ DMA イネーブル

- このビットがセットされると、RXNE フラグがセットされるたびに DMA リクエストが行われます。
- 0 : Rxバッファ DMA は無効です。
- 1 : Rxバッファ DMA は有効です。

20.5.3 SPI ステータスレジスタ (SPI_SR)

アドレスオフセット : 0x08

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み							FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
							r	r	r	r	rc_w0	r	r	r	r

ビット 15:9 予約済み。ハードウェアによって 0 に固定されています。

ビット 8 **FRE** : フレームフォーマットエラー

- 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。
- このフラグは、ハードウェアによってセットされ、SPIx_SR レジスタを読み出すとソフトウェアでクリアされます。

注： このフラグは、SPI が TI スレープモードまたは I²S スレープモードで動作する場合に使用されず (セクション 20.3.10 を参照)。

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 20.3.7 : ステータスフラグ および セクション 20.3.8 : SPI の無効化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、593 ページのセクション 20.4.8 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、575 ページのセクション 20.3.10 を参照してください。

注： このビットは I²S モードでは使用しません。

- ビット 4 **CRCERR** : CRC エラーフラグ
- 0 : 受信した CRC 値が SPI_RXCRCR 値と一致します。
 - 1 : 受信した CRC 値が SPI_RXCRCR 値と一致しません。
- このフラグは、ハードウェアによってセットされ、ソフトウェアによって 0 を書き込むことでクリアされます。
- 注 :** このビットは I²S モードでは使用しません。
- ビット 3 **UDR** : アンダーランフラグ
- 0 : アンダーランは発生していません。
 - 1 : アンダーランが発生しました。
- このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[593 ページのセクション 20.4.8](#) を参照してください。
- 注 :** このビットは SPI モードでは使用しません。
- ビット 2 **CHSIDE** : チャネルサイド
- 0 : 左チャネルを送信する必要があるか、または受信が行われました。
 - 1 : 右チャネルを送信する必要があるか、または受信が行われました。
- 注 :** このビットは、SPI モードに対しては使用しません。また、PCM モードでは意味を持ちません。
- ビット 1 **TXE** : 送信バッファエンプティ
- 0 : Tx バッファは空ではありません。
 - 1 : Tx バッファは空です。
- ビット 0 **RXNE** : 受信バッファノットエンプティ
- 0 : Rx バッファは空です。
 - 1 : Rx バッファは空ではありません。

20.5.4 SPI データレジスタ (SPI_DR)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 15:0 **DR[15:0]** : データレジスタ
- 受信したデータまたは送信されるデータ
- データレジスタは 2 つのバッファに分割されます。一方は書き込み用 (送信バッファ) であり、他方は読出し用 (受信バッファ) です。データレジスタへの書き込みは、Tx バッファへの書き込みになります。データレジスタからの読出しでは、Rx バッファに保持されている値が返されます。
- 注 :** 以下の注記は SPI モードに適用されます。
- 送受信されるデータは、データフレームフォーマットの選択ビット (SPI_CR1 レジスタの DFF ビット) に応じて、8 ビットまたは 16 ビットになります。正しい動作のためには、この選択を行ってから、SPI を有効にする必要があります。
- 8 ビットデータフレームの場合、バッファは 8 ビットであり、レジスタの LSB (SPI_DR[7:0]) のみが送受信に使用されます。受信モードのとき、レジスタの MSB (SPI_DR[15:8]) は強制的に 0 にされます。
- 16 ビットデータフレームの場合、バッファは 16 ビットであり、レジスタ全体 (SPI_DR[15:0]) が送受信に使用されます。

20.5.5 SPI CRC 多項式レジスタ (SPI_CRCPR) (I²S モードでは使用しません)

アドレスオフセット : 0x10

リセット値 : 0x0007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
r/w	r/w	r/w	r/w	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 : これらのビットは I²S モードでは使用しません。

20.5.6 SPI RX CRC レジスタ (SPI_RXCRCR) (I²S モードでは使用しません)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **RXCRC[15:0]** : Rx CRC レジスタ

CRC 計算が有効なとき、RxCRC[15:0] ビットには、その後に受信したバイトから算出された CRC 値が格納されています。このレジスタは、SPI_CR1 レジスタの CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPI_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPI_CR1 の DFF ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPI_CR1 レジスタの DFF ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

注 : BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。

これらのビットは I²S モードでは使用しません。

20.5.7 SPI TX CRC レジスタ (SPI_TXCRCR) (I²S モードでは使用しません)

アドレスオフセット : 0x18

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **TXCRC[15:0]** : Tx CRC レジスタ

CRC 計算が有効なとき、TxCRC[7:0] ビットには、その後に送信されたバイトから算出された CRC 値が格納されます。このレジスタは、SPI_CR1 の CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPI_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。データフレームフォーマットが 8 ビットデータに設定された場合 (SPI_CR1 の DFF ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPI_CR1 レジスタの DFF ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

注 : BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。
これらのビットは I²S モードでは使用しません。

20.5.8 SPI_I²S 設定レジスタ (SPI_I2SCFGR)

アドレスオフセット : 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				I2SMOD	I2SE	I2SCFG		PCMSYN C	Res.	I2SSTD		CKPOL	DATLEN		CHLEN
				rw	rw	rw	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 **I2SMOD** : I²S モード選択

- 0 : SPI モードが選択されます。
- 1 : I²S モードが選択されます。

注 : このビットは、SPI または I²S が無効なときに設定してください。

ビット 10 **I2SE** : I²S イネーブル

- 0 : I²S ペリフェラルは無効です。
- 1 : I²S ペリフェラルは有効です。

注 : このビットは SPI モードでは使用しません。

ビット 9:8 **I2SCFG** : I²S 設定モード

- 00 : スレーブ - 送信
- 01 : スレーブ - 受信
- 10 : マスタ - 送信
- 11 : マスタ - 受信

注 : このビットは、I²S が無効なときに設定してください。
SPI モードでは使用しません。

ビット 7 **PCMSYNC** : PCM フレーム同期

0 : ショートフレーム同期

1 : ロングフレーム同期

注 : このビットは、**I2SSTD = 11 (PCM 規格使用)** の場合にのみ意味を持ちます。

SPI モードでは使用しません。

ビット 6 予約済みであり、ハードウェアによって 0 に固定されています。

ビット 5:4 **I2SSTD** : I²S 規格選択

00 : フィリップス I²S 規格

01 : MSB 詰め規格 (左詰め)

10 : LSB 詰め規格 (右詰め)

11 : PCM 規格

I²S 規格に関する詳細は、[579 ページのセクション 20.4.3](#) を参照してください。SPI モードでは使用しません。

注 : 正しい動作のためには、これらのビットは、I²S が無効のときに設定してください。

ビット 3 **CKPOL** : 定常状態クロック極性

0 : I²S クロックの定常状態はローレベルです。

1 : I²S クロックの定常状態はハイレベルです。

注 : 正しい動作のためには、このビットは、I²S が無効のときに設定してください。

このビットは SPI モードでは使用しません。

ビット 2:1 **DATLEN** : 転送されるデータ長

00 : 16 ビットデータ長

01 : 24 ビットデータ長

10 : 32 ビットデータ長

11 : 設定禁止

注 : 正しい動作のためには、これらのビットは、I²S が無効のときに設定してください。

このビットは SPI モードでは使用しません。

ビット 0 **CHLEN** : チャネル長 (オーディオチャネルごとのビット数)

0 : 16 ビット幅

1 : 32 ビット幅

ビット書込み動作は、**DATLEN = 00** のときにのみ意味を持ちます。そうでない場合、書き込まれた値とは無関係に、チャネル長はハードウェアによって 32 ビットに固定されます。SPI モードでは使用しません。

注 : 正しい動作のためには、このビットは、I²S が無効のときに設定してください。

20.5.9 SPI_I²S プリスケアラレジスタ (SPI_I2SPR)

アドレスオフセット : 0x20

リセット値 : 0000 0010 (0x0002)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み						MCKOE	ODD	I2SDIV							
						rw	rw	rw							

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **MCKOE** : マスタクロック出力有効

0 : マスタクロック出力は無効です。

1 : マスタクロック出力は有効です。

注 : このビットは、I²S が無効なときに設定してください。これらのビットは、I²S がマスタモードのときにのみ使用します。

このビットは SPI モードでは使用しません。

ビット 8 **ODD** : プリスケアラの奇数分周比

0 : 真のディバイダ値 = I2SDIV * 2

1 : 真のディバイダ値 = (I2SDIV * 2) + 1

[587 ページのセクション 20.4.4](#) を参照してください。SPI モードでは使用しません。

注 : このビットは、I²S が無効なときに設定してください。これらのビットは、I²S がマスタモードのときにのみ使用します。

ビット 7:0 **I2SDIV** : I²S リニアプリスケアラ

I2SDIV [7:0] = 0 または I2SDIV [7:0] = 1 は禁止されている値です。

[587 ページのセクション 20.4.4](#) を参照してください。SPI モードでは使用しません。

注 : これらのビットは、I²S が無効なときに設定してください。これらのビットは、I²S がマスタモードのときにのみ使用します。

20.5.10 SPI レジスタマップ

次の表に、SPI レジスタマップとリセット値を示します。

表 93. SPI レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	SPI_CR1	予約済み																BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA		
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	SPI_SR	予約済み																							FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE		
	リセット値	0																							0	0	0	0	0	0	0	0	1	0	
0x0C	SPI_DR	予約済み																DR[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	SPI_CRCPR	予約済み																CRCPOLY[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	1	1	1			
0x14	SPI_RXCRCR	予約済み																RxCRC[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	SPI_TXCRCR	予約済み																TxCRC[15:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1C	SPI_I2SCFGR	予約済み																				I2SMOD	I2SE	I2SCFG	PCMSSYNC	予約済み			I2SSTD	CKPOL	DATLEN	CHLEN			
	リセット値	0																				0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	SPI_I2SPR	予約済み																						MCKOE	ODD	I2SDIV									
	リセット値	0																						0	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、[38 ページの表 1](#) を参照してください。

21 セキュアデジタル入力／出力インタフェース (SDIO)

21.1 SDIO の主な機能

SD/SDIO MMC カードホストインタフェース (SDIO) は、APB2 ペリフェラルバスと MultiMediaCard (MMC)、SD メモリカード、SDIO カード、および CE-ATA デバイス間のインタフェースを提供します。

MMCA 技術委員会から発行された MultiMediaCard システム仕様書は、MultiMediaCard Association のウェブサイト www.mmca.org から入手できます。

SD メモリカードと SD I/O カードシステム仕様書は、SD card Association のウェブサイト <http://www.sdcard.org> から入手できます。

CE-ATA システム仕様書は、CE-ATA ワークグループのウェブサイト www.ce-ata.org から入手できます。

SDIO には以下の機能があります。

- MultiMediaCard System Specification Version 4.2 に完全準拠。3 種類の異なるデータバスモードのカードに対応：1 ビット（デフォルト）、4 ビットおよび 8 ビット
- 以前のバージョンの MultiMediaCard との完全互換（上位互換性）
- SD Memory Card Specifications Version 2.0 に完全準拠
- SD I/O Card Specification Version 2.0 完全準拠：2 種類の異なるデータバスモードのカードに対応：1 ビット（デフォルト）および 4 ビット
- CE-ATA 機能の完全サポート（CE-ATA digital protocol Rev1.1 に完全準拠）
- 8 ビットモードで最大 48 MHz のデータ転送
- 外部双方向ドライバを制御するデータおよびコマンド出力有効信号

注：SDIO は SPI 互換の通信モードを備えていません。

SD メモリカードプロトコルは、MultiMediaCard system specification V2.11 で定義されている MultiMediaCard プロトコルのスーパーセットです。SD メモリデバイスに必要なコマンドの中には、SD I/O 専用カードやコンポカードの I/O 部分ではサポートされないものもあります。消去コマンドなど、SD I/O デバイスで使用されないコマンドは、SDIO でサポートされません。また、コマンドの中には SD メモリカードと SD I/O カードで異なるものがあり、このようなコマンドは SDIO でサポートされません。詳細については、SD I/O card Specification Version 1.0 を参照してください。CEATA は、既存の MMC アクセスプリミティブを利用するプロトコルを使用して、MMC 電気接続によりサポートされます。インタフェースの電気信号定義は、MMC 基準で定義されているとおりです。

MultiMediaCard/SD バスは、カードをコントローラに接続します。

現行バージョンの SDIO は、1 度に 1 枚の SD/SDIO/MMC 4.2 カードと、MMC4.1 以前のスタックだけをサポートします。

21.2 SDIO バストロロジ

バス上の通信は、コマンドとデータ転送に基づいて行われます。

MultiMediaCard/SD/SD I/O バス上の基本的なトランザクションは、コマンド/レスポンストランザクションです。これらの種類のバストランザクションでは、コマンドやレスポンスの決められた通信フォーマットでカードの情報を直接転送します。また、データトークンを持つ操作もあります。

SD/SDIO メモリカードに対するデータ転送は、データブロック単位で行われます。MMC に対するデータ転送は、データブロックまたはストリーム単位で行われます。CE-ATA デバイスに対するデータ転送は、データブロック単位で行われます。

図 229. SDIO 「no response」と「no data」操作

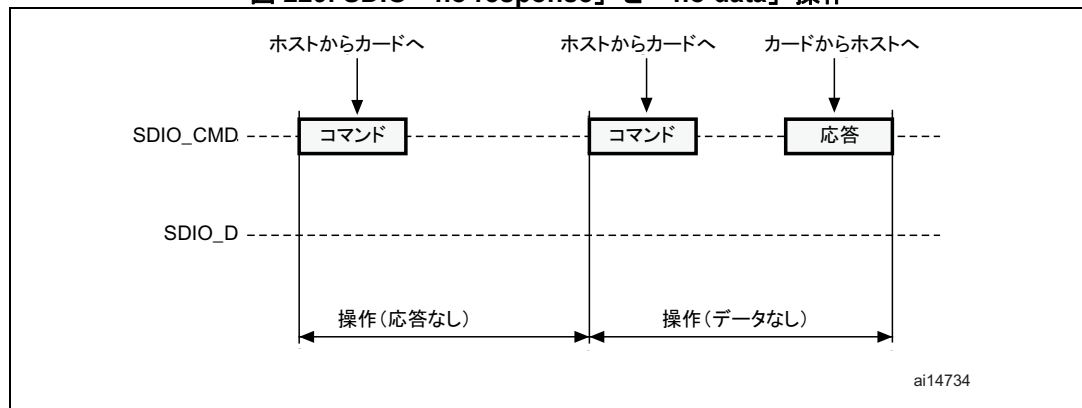


図 230. SDIO (複数) ブロック読み出し操作

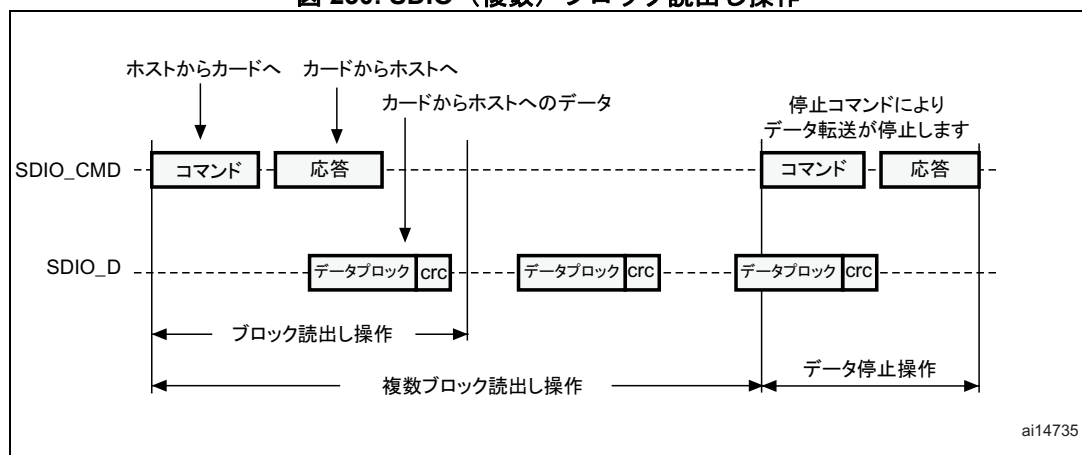
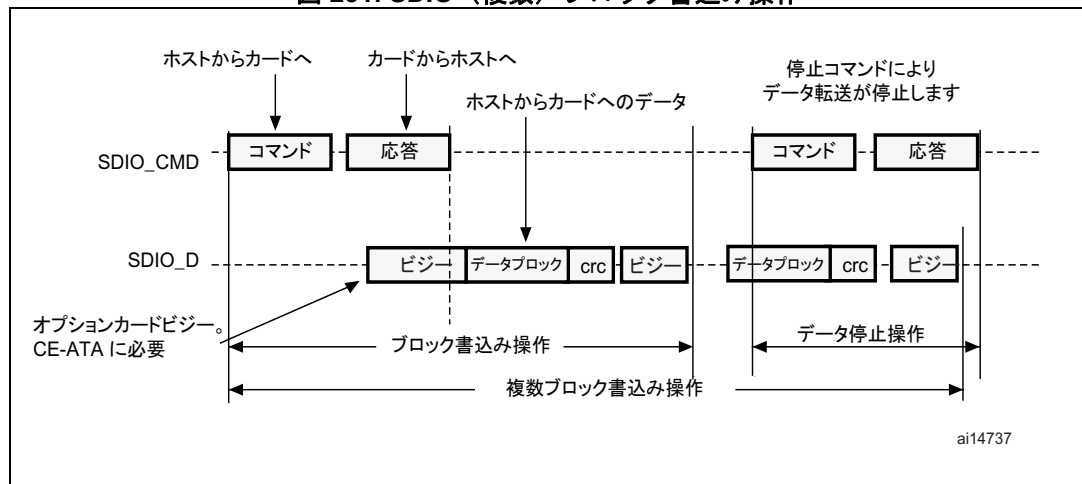


図 231. SDIO (複数) ブロック書き込み操作



注： SDIO は、ビジー信号がアサートされている限り (SDIO_D0 をローヘプル)、データを一切送信しません。

図 232. SDIO 順次読出し操作

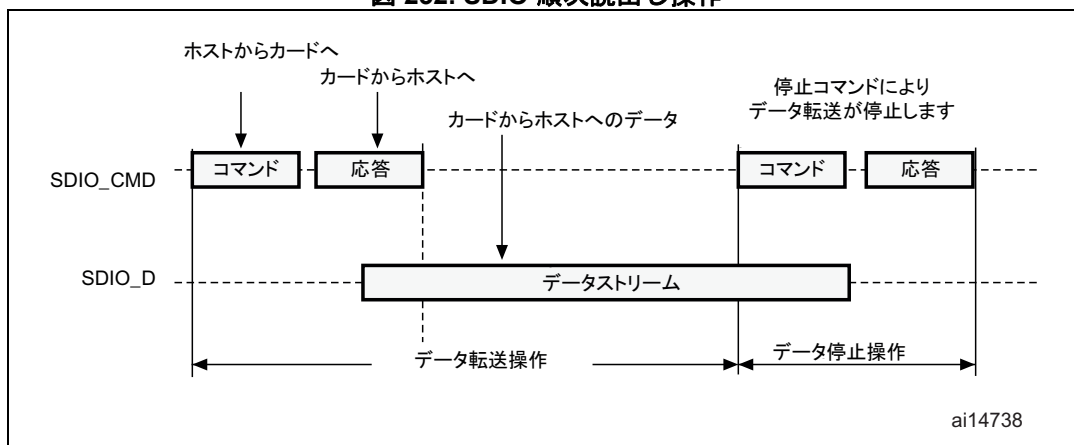
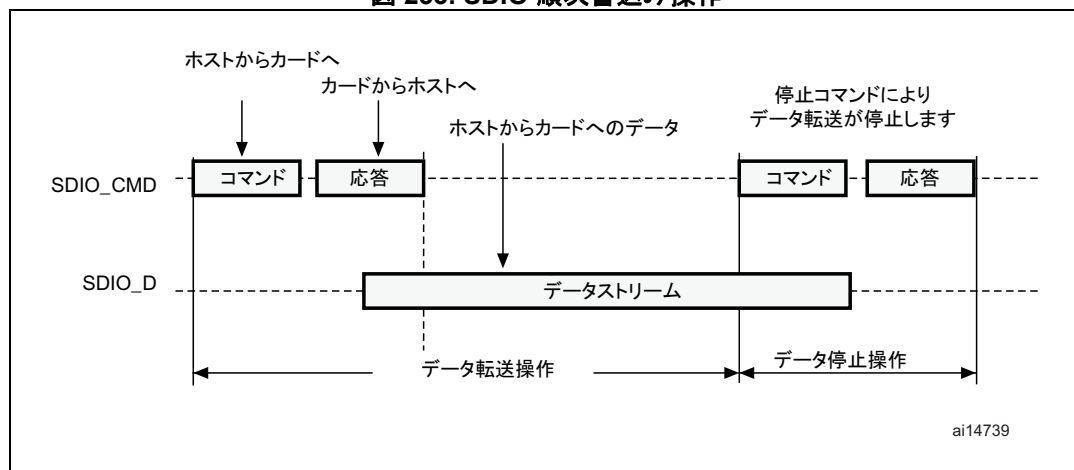


図 233. SDIO 順次書き込み操作

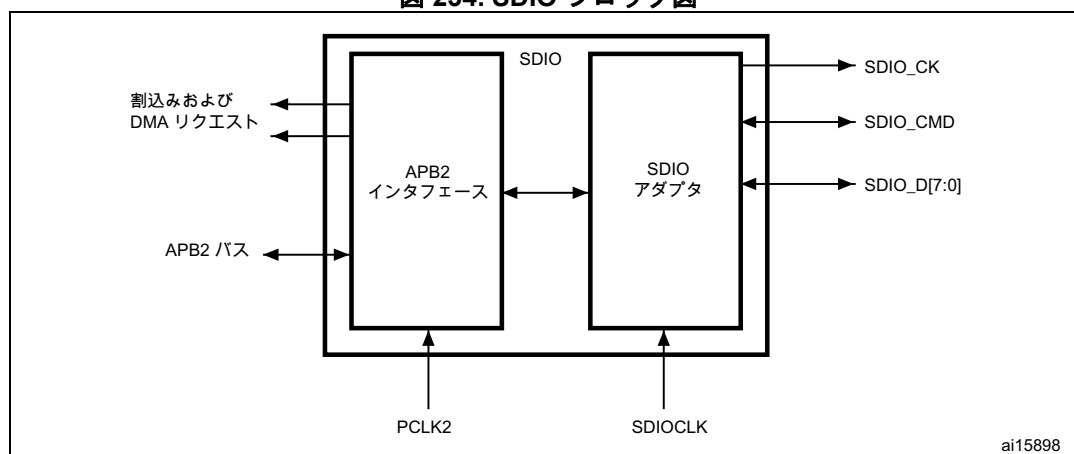


21.3 SDIO の機能詳細

SDIO は、2 つの部分で構成されています。

- SDIO アダプタブロックは、クロック生成ユニット、コマンドおよびデータ転送など、MMC/SD/SD I/O カードに固有のすべての機能を提供します。
- APB2 インタフェースは、SDIO アダプタレジスタにアクセスして、割り込みおよび DMA リクエスト信号を生成します。

図 234. SDIO ブロック図



デフォルトでは、SDIO_D0 はデータ転送に使用されます。初期化後、ホストはデータバス幅を変更することができます。

MultiMediaCard がバスに接続されている場合、SDIO_D0、SDIO_D[3:0]、または SDIO_D[7:0] はデータ転送に使用できます。MMC V3.31 以前では、1 ビットのデータのみがサポートされるため、使用できるのは SDIO_D0 だけです。

SD または SD I/O カードがバスに接続されている場合、ホストはデータ転送に SDIO_D0 または SDIO_D[3:0] を使用するように設定できます。すべてのデータラインはプッシュプルモードで動作します。

SDIO_CMD には、2 つの動作モードがあります。

- 初期化の場合は、オープンドレイン（MMCV3.31 以前）
- コマンド転送の場合は、プッシュプル（SD/SD I/O カード MMC4.2 は、初期化にもプッシュプルドライバを使用）

SDIO_CK は、カードに供給されるクロックです。各クロックサイクルで、コマンドラインとデータラインの両方に 1 ビットが転送されます。クロック周波数は、MultiMediaCard V3.31 の場合は 0 MHz から 20 MHz、MultiMediaCard V4.0/4.2 の場合は 0 から 48 MHz、SD/SD I/O カードの場合は 0 から 25 MHz の間で変化します。

SDIO は 2 つのクロック信号を使用します。

- SDIO アダプタクロック（SDIOCLK = 48 MHz）
- APB2 バスクロック（PCLK2）

PCLK2 および SDIO_CK クロック周波数は、次の条件を満足している必要があります。

$$\text{Frequency(PCLK2)} \geq 3/8 \times \text{Frequency(SDIO_CK)}$$

表 94 に示す信号は、MultiMediaCard/SD/SD I/O カードバス上で使用されます。

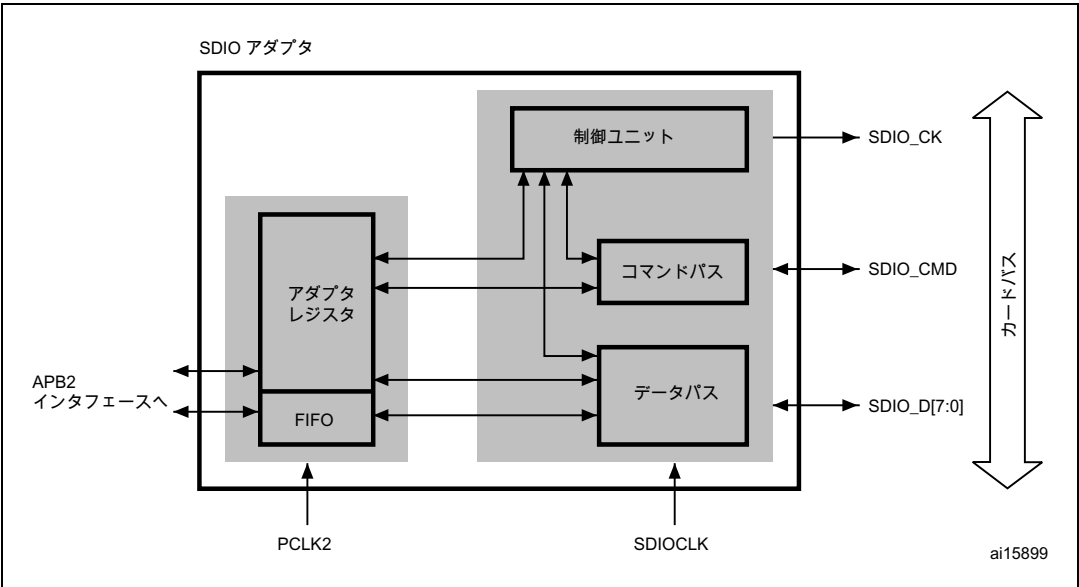
表 94. SDIO I/O 定義

ピン	方向	説明
SDIO_CK	出力	MultiMediaCard/SD/SDIO カードクロック。このピンは、ホストからカードへのクロックです。
SDIO_CMD	双方向	MultiMediaCard/SD/SDIO カードコマンド。このピンは、双方向のコマンド/レスポンス信号です。
SDIO_D[7:0]	双方向	MultiMediaCard/SD/SDIO カードデータ。これらのピンは、双方向データバスです。

21.3.1 SDIO アダプタ

図 235 は、SDIO アダプタの簡略化されたブロック図です。

図 235. SDIO アダプタ



SDIO アダプタは、マルチメディアカードスタックやセキュアデジタルメモリカードとのインタフェースを提供するマルチメディア/セキュアデジタルメモリカードバスマスターです。5つのサブユニットで構成されています。

- アダプタレジスタブロック
- 制御ユニット
- コマンドバス
- データバス
- データ FIFO

注：アダプタレジスタと FIFO は、APB2 バスクロックドメイン (PCLK2) を使用します。制御ユニット、コマンドバス、およびデータバスは、SDIO アダプタクロックドメイン (SDIOCLK) を使用します。

アダプタレジスタブロック

アダプタレジスタブロックは、すべてのシステムレジスタを含んでいます。このブロックは、マルチメディアカードのスタティックフラグをクリアする信号も生成します。クリア信号は、SDIO クリアレジスタの対応するビット位置に 1 が書き込まれると生成されます。

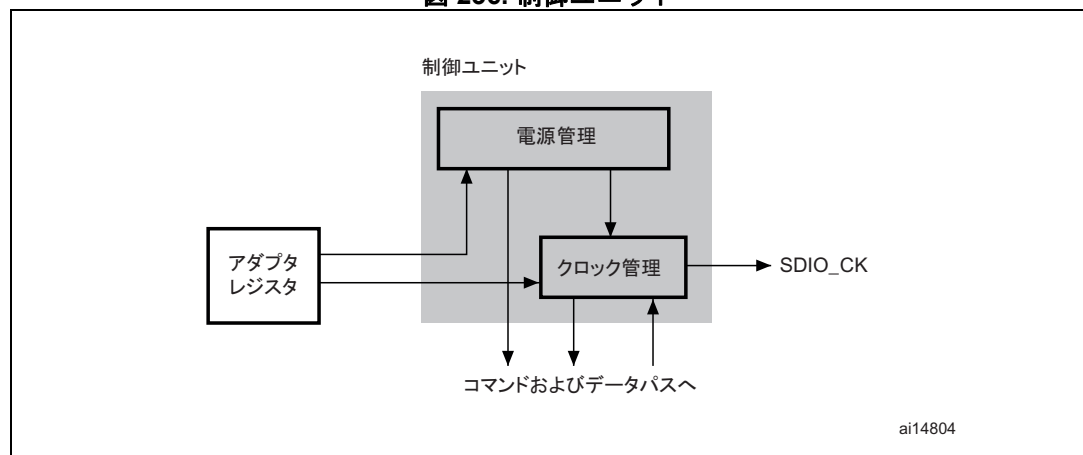
制御ユニット

制御ユニットには、電源管理機能とメモ리카ードクロックのクロック分周器が含まれています。

3つの電源フェーズがあります。

- パワーオフ
- パワーアップ
- パワーオン

図 236. 制御ユニット



制御ユニットは、図 236 に示されます。電源管理サブユニットとクロック管理サブユニットで構成されています。

電源管理サブユニットは、パワーオフおよびパワーアップフェーズの間、カードバス出力信号を無効にします。

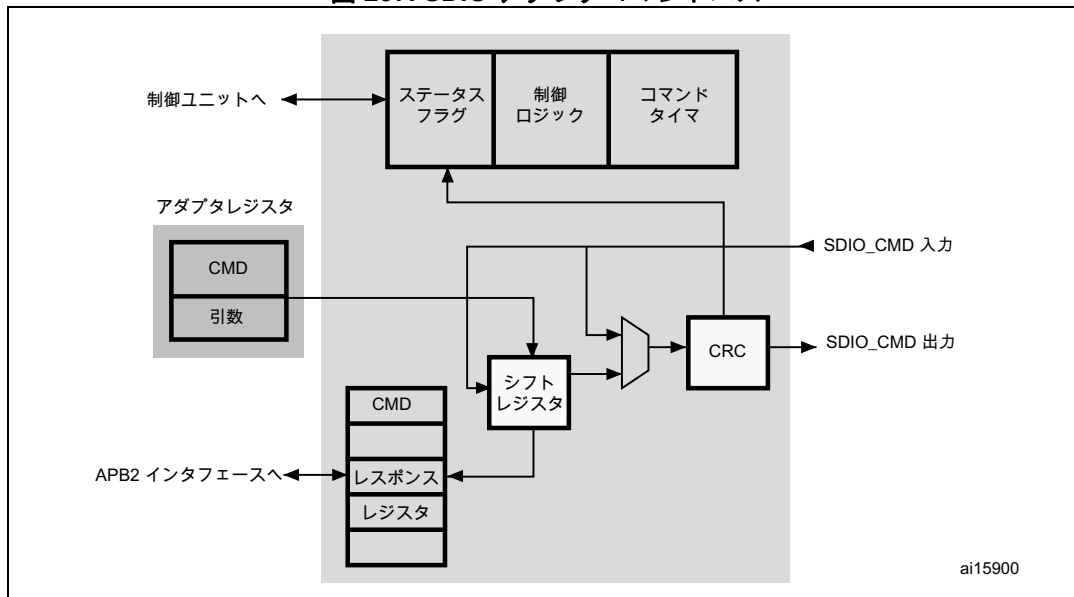
クロック管理サブユニットは、SDIO_OK 信号を生成し、制御します。SDIO_CK 出力は、クロック分周またはクロックバイパスモードを使用することができます。クロック出力は、次の場合にインアクティブです。

- リセット後
- パワーオフまたはパワーアップフェーズ中
- 節電モードが有効であり、カードバスがアイドル状態の場合（コマンドおよびデータバスサブユニットがアイドルフェーズに入ってから 8 クロック周期後）

コマンドパス

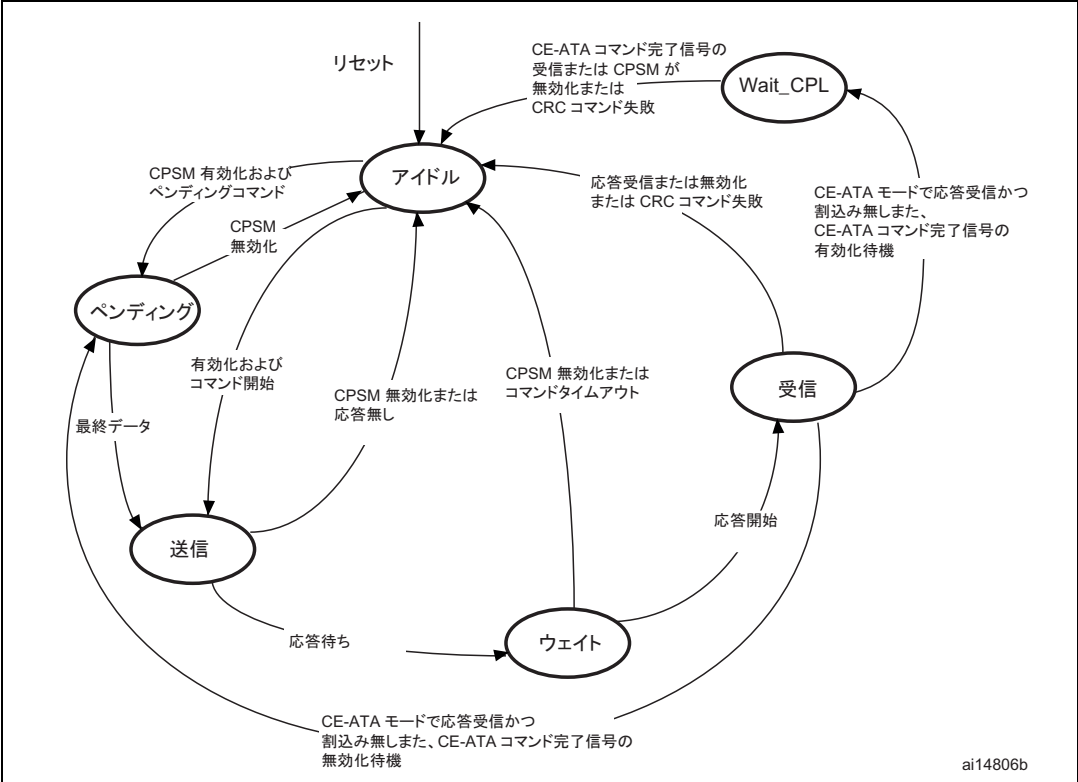
コマンドパスユニットは、カードにコマンドを送信し、カードからレスポンスを受信します。

図 237. SDIO アダプタコマンドパス



- コマンドパスステートマシン (CPSM)
 - コマンドレジスタが書き込まれて、有効ビットがセットされると、コマンド転送が開始します。コマンドが送信されると、コマンドパスステートマシン (CPSM) は、ステータスフラグをセットし、レスポンスが不要な場合はアイドル状態になります。レスポンスが必要な場合は、レスポンスを待ちます (612 ページの図 238 を参照)。レスポンスが受信されると、受信した CRC コードと内部生成されたコードを比較して適切なステータスフラグをセットします。

図 238. コマンドパスステートマシン (CPSM)



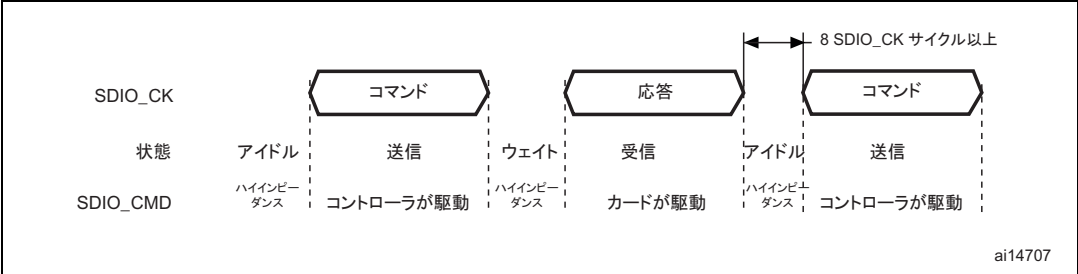
ウェイト状態に入ると、コマンドタイマが作動します。CPSM が受信状態になる前にタイムアウトに達した場合には、タイムアウトフラグがセットされ、アイドル状態に入ります。

注： コマンドタイムアウトは、64 SDIO_CK クロック周期の固定値です。

コマンドレジスタの割込みビットがセットされると、タイマは無効になり、CPSM はカードの 1 枚からの割込みリクエストを待ちます。コマンドレジスタのペンディングビットがセットされると、CPSM はペンディング状態に入り、データパスサブユニットからの CmdPend 信号を待ちます。CmdPend が検出されると、CPSM は送信状態に移行します。これによって、停止コマンドの送信をトリガするデータカウンタが有効になります。

注： CPSM は、 N_{CC} および N_{RC} タイミング制約を満たすために、少なくとも 8 SDIO_CK 周期の間、アイドル状態を保ちます。 N_{CC} は、2 つのホストコマンド間の最小遅延であり、 N_{RC} は、ホストコマンドとカードレスポンスの間の最小遅延です。

図 239. SDIO コマンド転送



- コマンドフォーマット
 - コマンド: コマンドは、動作を開始するトークンです。コマンドは、ホストから 1 枚のカード（アドレス付きコマンド）または接続されたすべてのカードに送信されます（ブロードキャストコマンドは MMCV3.31 以前で使用可能）。コマンドは、CMD ライン上で連続的に転送されます。すべてのコマンドは、固定長 48 ビットです。MultiMediaCard、SD メモリカード、および SDIO カードのコマンドトークンの一般的なフォーマットを表 95 に示します。CE-ATA コマンドは、MMC コマンド V4.2 の拡張であり、同じフォーマットです。
コマンドパスは、コマンドとレスポンスを送受信できるように半二重モードで動作します。CPSM が送信状態でない場合、SDIO_CMD 出力は 612 ページの図 239 に示すようにハイインピーダンス状態になります。SDIO_CMD のデータは、SDIO_CK の立上りエッジと同期します。表 95 に、コマンドフォーマットを示します。

表 95. コマンドフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	1	送信ビット
[45:40]	6	-	コマンドインデックス
[39:8]	32	-	引数
[7:1]	7	-	CRC7
0	1	1	終了ビット

- レスポンス: レスポンスは、以前に受信したコマンドへの応答として、アドレス指定されたカードから（MMC V3.31 以前の場合はすべてのカードから同期して）ホストに送信されるトークンです。レスポンスは、CMD ライン上で連続的に転送されます。
- SDIO は、2 種類のレスポンスをサポートします。どちらも CRC エラーチェックを使用します。
- 48 ビットショートレスポンス
 - 136 ビットロングレスポンス

注: レスポンスが CRC を含まない場合（CMD1 レスポンス）、デバイスドライバは CRC 障害ステータスを無視する必要があります。

表 96. ショートレスポンスフォーマット

ビット位置	幅	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	-	コマンドインデックス
[39:8]	32	-	引数
[7:1]	7	-	CRC7（または 1111111）
0	1	1	終了ビット

表 97. ロングレスポンスフォーマット

ビット位置	幅	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	111111	予約済み
[127:1]	127	-	CID または CSD（内部 CRC7 を含む）
0	1	1	終了ビット

コマンドレジスタは、コマンドインデックス（カードに送信される 6 ビット）とコマンドタイプを含みます。これらによって、コマンドがレスポンスを必要とするかどうかと、レスポンスが 48 ビット長か 136 ビット長かが決まります（648 ページのセクション 21.9.4 を参照）。コマンドパスは、表 98 に示すようにステータスフラグを実装します。

表 98. コマンドパスステータスフラグ

フラグ	説明
CMDREND	レスポンス CRC が OK の場合にセットされます。
CCRCFAIL	レスポンス CRC が失敗した場合にセットされます。
CMDSENT	コマンド（レスポンスを必要としない）が送信されたときにセットされます。
CTIMEOUT	レスポンスタイムアウトです。
CMDACT	コマンド転送中です。

CRC 生成回路は、CRC コードの前のすべてのビットについて CRC チェックサムを計算します。これには、スタートビット、送信ビット、コマンドインデックス、およびコマンド引数（またはカードステータス）が含まれます。CRC チェックサムは、ロングレスポンスフォーマットの場合、CID または CSD の最初の 120 ビットについて計算されます。スタートビット、送信ビット、および 6 つの予約ビットは、CRC 計算では使用されませんので注意してください。

CRC チェックサムは 7 ビットの値です。

$$\text{CRC}[6:0] = \text{剰余} [(M(x) * x^7) / G(x)]$$

$$G(x) = x^7 + x^3 + 1$$

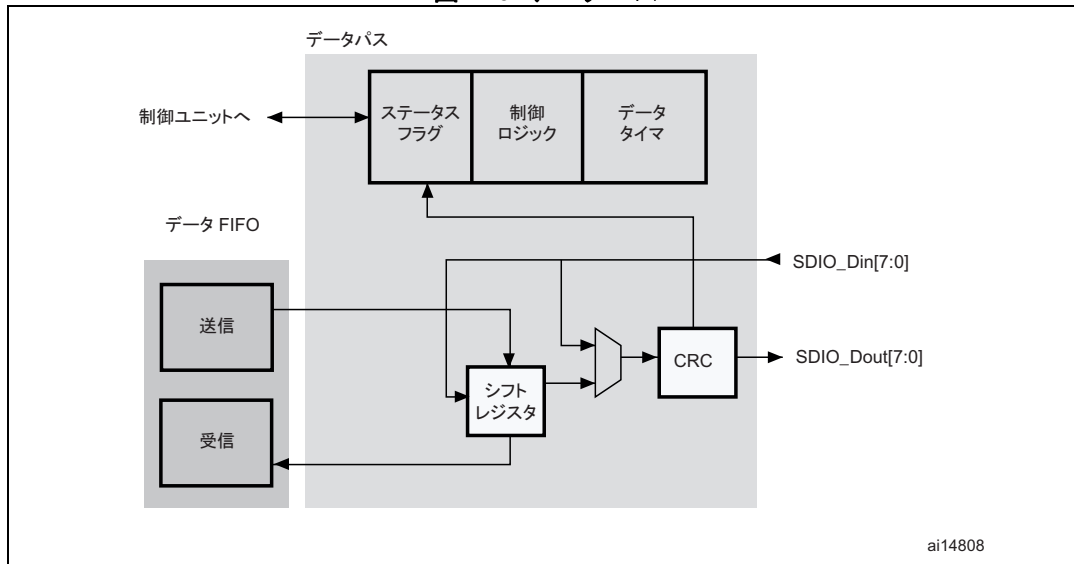
$$M(x) = (\text{スタートビット}) * x^{39} + \dots + (\text{CRC 前の最後のビット}) * x^0, \text{ または}$$

$$M(x) = (\text{スタートビット}) * x^{119} + \dots + (\text{CRC 前の最後のビット}) * x^0$$

データパス

データパスサブユニットは、カードとの間でデータ転送を行います。図 240 に、データパスのブロック図を示します。

図 240. データパス



カードデータバス幅は、クロック制御レジスタを使用してプログラムできます。4 ビット幅バスモードが有効な場合、データは 4 つのデータ信号すべて (SDIO_D[3:0]) で、クロックサイクルごとに 4 ビットで転送されます。8 ビット幅バスモードが有効な場合、データは 8 つのデータ信号すべて (SDIO_D[7:0]) で、クロックサイクルごとに 8 ビットで転送されます。ワイドバスモードが有効でない場合は、SDIO_D0 上でクロックサイクルごとに 1 ビットだけが転送されます。

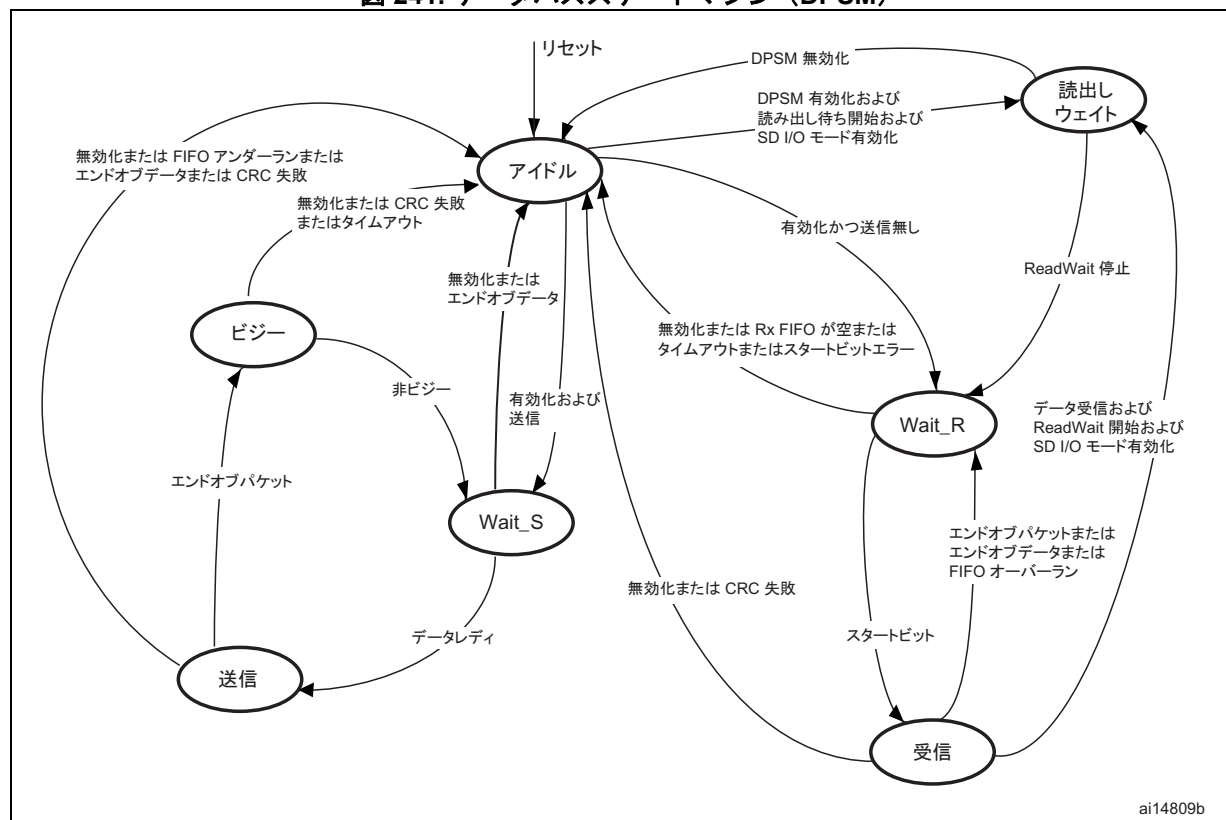
転送の方向 (送信または受信) によって、データパスステートマシン (DPSM) は、有効になったときに Wait_S または Wait_R 状態へ移行します。

- 送信 : DPSM は Wait_S 状態へ移行します。送信 FIFO にデータがある場合、DPSM は送信状態へ移行し、データパスサブユニットはカードへのデータ転送を開始します。
- 受信 : DPSM は Wait_R 状態へ移行して、スタートビットを待ちます。スタートビットを受信すると、DPSM は受信状態へ移行し、データパスサブユニットはカードからのデータ受信を開始します。

データパスステートマシン (DPSM)

DPSM は、SDIO_CK 周波数で動作します。カードバス信号上のデータは、SDIO_CK の立上りエッジと同期しています。図 241 : データパスステートマシン (DPSM) に示すように、DPSM には、6 つの異なる状態があります。

図 241. データパスステートマシン (DPSM)



- ・ アイドル：データパスはインアクティブであり、SDIO_D[7:0] 出力はハイインピーダンスです。データ制御レジスタが書き込まれ、有効ビットがセットされると、DPSM はデータカウンタに新しい値をロードし、データ方向ビットに応じて Wait_S または Wait_R 状態へ移行します。
- ・ Wait_R：データカウンタがゼロに等しい場合、DPSM は受信 FIFO が空になると、アイドル状態へ移行します。データカウンタがゼロでない場合、DPSM は SDIO_D 上のスタートビットを待ちます。タイムアウト前にスタートビットを受信した場合、DPSM は受信状態に移行し、データブロックカウンタをロードします。スタートビットを検出する前にタイムアウトに達した場合、またはスタートビットエラーが発生した場合には、アイドル状態へ移行して、タイムアウトステータスフラグをセットします。
- ・ 受信：カードから受信したシリアルデータは数バイトにまとめられ、データ FIFO に書き込まれます。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
 - － ブロックモードでは、データブロックカウンタがゼロに達すると、DPSM は CRC コードを受信するまで待ちます。受信したコードが内部生成された CRC コードと一致する場合、DPSM は Wait_R 状態へ移行します。そうでない場合、CRC 失敗ステータスフラグをセットして、DPSM はアイドル状態へ移行します。
 - － ストリームモードでは、DPSM はデータカウンタがゼロでない間、データを受信します。カウンタがゼロになると、シフトレジスタ内の残りのデータがデータ FIFO に書き込まれ、DPSM は Wait_R 状態へ移行します。

FIFO オーバーランエラーが発生した場合、DPSM は FIFO エラーフラグをセットして、アイドル状態へ移行します。

- ・ Wait_S：データカウンタがゼロの場合、DPSM はアイドル状態へ移行します。そうでない場合は、データ FIFO エンプティフラグがネゲートされるまで待ってから、送信状態へ移行します。

- 注 : DPSM は、 N_{WR} タイミングの要件を満たすため、少なくとも 2 クロック周期の間、Wait_S 状態を保ちます。ここで、 N_{WR} は、カードレスポンスの受信とホストからのデータ転送の開始との間のクロックサイクル数です。
- 送信 : DPSM はカードへのデータの送信を開始します。データ制御レジスタの転送モードビットに応じて、データ転送はブロックモードまたはストリームモードのいずれかとなります。
 - ブロックモードでは、データブロックカウンタがゼロに達すると、DPSM は内部生成された CRC コードと終了ビットを送信して、ビジー状態へ移行します。
 - ストリームモードでは、DPSM は、有効ビットがハイであり、データカウンタがゼロでない間、カードへデータを送信します。その後、アイドル状態へ移行します。FIFO アンダーランエラーが発生した場合、DPSM は FIFO エラーフラグをセットして、アイドル状態へ移行します。
 - ビジー : DPSM は、CRC ステータスフラグを待ちます。
 - 良好な CRC ステータスを受信しなかった場合、DPSM はアイドル状態へ移行し、CRC 失敗ステータスフラグをセットします。
 - 良好な CRC ステータスを受信した場合、SDIO_D0 がローでない (カードがビジーでない) 場合は Wait_S 状態へ移行します。DPSM がビジー状態の間にタイムアウトエラーが発生した場合には、DPSM はデータタイムアウトフラグをセットして、アイドル状態へ移行します。

データタイマは、DPSM が Wait_R またはビジー状態のときに有効になり、以下の場合にデータタイムアウトを生成します。

 - データ送信中、DPSM がプログラムされたタイムアウト時間より長く、ビジー状態にとどまっていた場合。
 - データ受信時、データの終わりが真でなく、DPSM がプログラムされたタイムアウト時間より長く Wait_R 状態にとどまっていた場合、タイムアウトが発生します。 - データ : データは、カードからホストへ、またはホストからカードへ転送できます。データは、データラインを介して転送されます。32 ワードの FIFO に格納され、各ワードは 32 ビット幅です。

表 99. データトークンフォーマット

説明	スタートビット	データ	CRC16	終了ビット
ブロックデータ	0	-	あり	1
ストリームデータ	0	-	なし	1

データ FIFO

データ FIFO（ファーストインファーストアウト）サブユニットは、送受信ユニットを持つデータバッファです。

FIFO には、32 ビット幅、32 ワードの深さのデータバッファと、送受信ロジックがあります。データ FIFO は APB2 クロックドメイン（PCLK2）で動作するため、SDIO クロックドメイン（SDIOCLK）のサブユニットからの信号はすべて再同期されます。

TXACT および RXACT フラグに応じて、FIFO は無効、送信有効、または受信有効にできます。TXACT と RXACT は、データパスサブユニットによって駆動され、相互排他的となっています。

- 送信 FIFO は、TXACT がアサートされると、送信ロジックとデータバッファを参照します。
- 受信 FIFO は、RXACT がアサートされると、受信ロジックとデータバッファを参照します。

• 送信 FIFO :

データは、SDIO が送信有効なときに、APB2 インタフェースを通じて送信 FIFO に書き込むことができます。

送信 FIFO は、32 個の連続アドレスを介してアクセス可能です。送信 FIFO には、読出しポインタが示すデータワードを保持するデータ出力レジスタが含まれています。データパスサブユニットがシフトレジスタへロードすると、読出しポインタがインクリメントされ、新しいデータが出力されます。

送信 FIFO が無効な場合、すべてのステータスフラグがネゲートされます。データパスサブユニットは、データを送信するときに TXACT をアサートします。

表 100. 送信 FIFO ステータスフラグ

フラグ	説明
TXFIFOOF	32 個すべての送信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
TXFIFOE	送信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
TXFIFOHE	8 個以上の送信 FIFO ワードが空のとき、ハイにセットされます。このフラグは DMA リクエストとして使用できます。
TXDAVL	送信 FIFO が有効なデータを含んでいるとき、ハイにセットされます。このフラグは、TXFIFIOE フラグの反転です。
TXUNDERR	アンダーランエラーが発生すると、ハイにセットされます。このフラグは、SDIO クリアレジスタへの書き込みによってクリアされます。

• 受信 FIFO

データパスサブユニットがワードデータを受信すると、書込みデータバス上にデータが出力されます。書込み操作が完了すると、書込みポインタがインクリメントされます。読出し側では、読出しポインタの現在値が示す FIFO ワードの内容が、読出しバス上に出力されます。受信 FIFO が無効な場合、すべてのステータスフラグがネゲートされ、読出しおよび書込みポインタがリセットされます。データパスサブユニットは、データを受信すると、RXACT をアサートします。[表 101](#) に、受信 FIFO ステータスフラグをリストします。受信 FIFO は、32 個の連続アドレスを介してアクセス可能です。

表 101. 受信 FIFO ステータスフラグ

フラグ	説明
RXFIFO	32 個すべての受信 FIFO ワードが有効なデータを含んでいるとき、ハイにセットされます。
RXFIFOE	受信 FIFO が有効なデータを含んでいないとき、ハイにセットされます。
RXFIFOHF	8 個以上の受信 FIFO ワードが空のとき、ハイにセットされます。このフラグは DMA リクエストとして使用できます。
RXDAVL	受信 FIFO が空でないとき、ハイにセットされます。このフラグは RXFIFOE フラグの反転です。
RXOVERR	オーバーランエラーが発生すると、ハイにセットされます。このフラグは、SDIO クリアレジスタへの書き込みによってクリアされます。

21.3.2 SDIO APB2 インタフェース

APB2 インタフェースは、割込みおよび DMA リクエストを生成し、SDIO アダプタレジスタとデータ FIFO にアクセスします。このインタフェースは、データバス、レジスタデコーダ、および割込み／DMA ロジックで構成されています。

SDIO 割込み

割込みロジックは、選択されたステータスフラグの少なくとも 1 つがハイのときにアサートされる割込みリクエスト信号を生成します。割込み生成条件が選択できるように、マスクレジスタが提供されています。ステータスフラグは、対応するマスクフラグがセットされている場合に割込みリクエストを生成します。

SDIO/DMA インタフェース - SDIO とメモリ間のデータ転送手順

次に示す例では、SDIO ホストコントローラから MMC に対して転送が行われます (CMD24 (WRITE_BLOCK) を使用して 512 バイトを転送)。SDIO FIFO には、DMA コントローラを使用してメモリに格納されたデータが入力されます。

1. カード識別プロセスを行います。
2. SDIO_CK 周波数を上げます。
3. CMD7 を送信してカードを選択します。
4. 次のように DMA2 を構成します。
 - a) DMA2 コントローラを有効にして、ペンディング中の割込みをすべてクリアします。
 - b) DMA2_Stream3 または DMA2_Stream6 Channel4 ソースアドレスレジスタをメモリ位置のベースアドレスでプログラムし、DMA2_Stream3 または DMA2_Stream6 Channel4 デスティネーションアドレスレジスタを SDIO FIFO レジスタアドレスでプログラムします。
 - c) DMA2_Stream3 または DMA2_Stream6 Channel4 制御レジスタをプログラムします (ペリフェラルインクリメントではなくメモリインクリメント、またペリフェラル幅とソース幅はワードサイズ)。
 - d) DMA2_Stream3 または DMA2_Stream6 Channel4 をプログラムしてペリフェラルをフローコントローラとして選択します (DMA_S3CR または DMA_S6CR 設定レジスタの PFCTRL ビットをセットします)。
 - e) DMA2_Stream3 または DMA2_Stream6 Channel4 で、インクリメンタルバースト転送を 4 ビート (最低でもペリフェラル側から) に設定します。
 - f) DMA2_Stream3 または DMA2_Stream6 Channel4 を有効にします。

5. 次のように、CMD24 (WRITE_BLOCK) を送信します。
 - a) SDIO データ長レジスタをプログラムします (SDIO データタイマレジスタは、カード識別プロセスの前にすでにプログラムされている必要があります)。
 - b) SDIO 引数レジスタに、データが転送されるカードのアドレス位置をプログラムします。
 - c) SDIO コマンドレジスタを次のようにプログラムします: CmdIndex は 24 (WRITE_BLOCK) に、WaitResp は "1" (SDIO カードホストはレスポンスを待つ)、CPSMEN は "1" (SDIO カードホストはコマンドの送信が可能) にプログラムします。他のフィールドはリセット値です。
 - d) SDIO_STA[6] = CMDREND 割込みを待ち、SDIO データ制御レジスタを次のようにプログラムします: DTEN は "1" (SDIO カードホストはデータの送信が可能)、DTDIR は "0" (コントローラからカードへ)、DTMODE は "0" (ブロックデータ転送)、DMAEN は "1" (DMA 有効)、DBLOCKSIZE は 0x9 (512 バイト) にプログラムします。他のフィールドは無視します。
 - e) SDIO_STA[10] = DBCKEND を待ちます。
6. DMA 有効チャネルステータスレジスタをポーリングして、有効になっているチャネルがまだないことを確認します。

21.4 カード機能詳細

21.4.1 カード識別モード

カード識別モードでは、ホストはすべてのカードをリセットし、動作電圧範囲を検証し、カードを識別し、バス上の各カードの相対カードアドレス (RCA) をセットします。カード識別モードでは、すべてのデータ通信にコマンドライン (CMD) のみを使用します。

21.4.2 カードリセット

GO_IDLE_STATE コマンド (CMD0) は、ソフトウェアリセットコマンドであり、MultiMediaCard と SD メモリをアイドル状態にします。IO_RW_DIRECT コマンド (CMD52) は、SD I/O カードをリセットします。パワーアップまたは CMD0 の後、すべてのカード出力バスドライバはハイインピーダンス状態となり、カードはデフォルトの相対カードアドレス (RCA=0x0001) とデフォルトのドライバステージレジスタ設定 (最低速度、最高駆動電流) で初期化されます。

21.4.3 動作電圧範囲の検証

すべてのカードは、仕様範囲内の任意の動作電圧で SDIO カードホストと通信できます。サポートされる最小および最大 V_{DD} 値は、カード上の動作条件レジスタ (OCR) で定義されています。

ペイロードメモリにカード識別番号 (CID) とカード固有データ (CSD) を格納するカードは、この情報をデータ転送 V_{DD} 条件下でのみ通信することができます。SDIO カードホストモジュールとカードの V_{DD} 範囲に互換性がないときには、カードは識別サイクルを完了できず、CSD データを送信できません。このため、SDIO カードホストに必要な V_{DD} 範囲に一致しないカードを識別して拒否するメカニズムとして、特殊なコマンド SEND_OP_COND (CMD1)、SD_APP_OP_COND (SD メモリ用 ACMD41)、および IO_SEND_OP_COND (SD I/O 用 CMD5) が用意されています。SDIO カードホストは、これらのコマンドのオペランドとして、必要な V_{DD} 電圧範囲を送信します。指定された範囲でデータ転送を行えないカードは、バスから切断され、インアクティブ状態になります。

電圧範囲をオペランドとして含めずにこれらのコマンドを使用すると、SDIO カードホストは各カードへの問い合わせを行い、共通の電圧範囲を判断した後に、範囲外のカードをインアクティブ状態にできます。この問い合わせは、SDIO カードホストが共通の電圧範囲を選択できるときや、カードが使用不可能であることをユーザに通知する必要があるときに使用されます。

21.4.4 カード識別プロセス

カード識別プロセスは、MultiMediaCard と SD カードで異なります。MultiMediaCard では、識別プロセスは F_{od} クロックレートで開始します。SDIO_CMD ライン出力ドライバはオープンドレインであり、このプロセス中は並行カード操作が可能です。登録プロセスは、次のように行われます。

1. バスがアクティベートされます。
2. SDIO カードホストは、SEND_OP_COND (CMD1) をブロードキャストして、動作条件を受信します。
3. レスポンスは、すべてのカードからの動作条件レジスタのワイヤード AND 操作です。
4. 互換性のないカードはインアクティブ状態となります。
5. SDIO カードホストは、すべてのアクティブカードに ALL_SEND_CID (CMD2) をブロードキャストします。
6. アクティブカードは、それぞれの CID 番号をシリアルに同時送信します。発信 CID ビットがコマンドラインのビットと一致しないカードは、転送を停止して、次の識別サイクルを待たなければなりません。1 枚のカードが CID 全体を正常に SDIO カードホストに送信すると、識別状態へ移行します。
7. SDIO カードホストは、そのカードに SET_RELATIVE_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードはスタンバイ状態に遷移し、その後の識別サイクルには反応せず、出力はオープンドレインからプッシュプルへ切り替わります。
8. SDIO カードホストは、タイムアウト条件を受信するまで、ステップ 5~7 を繰り返します。

SD カードの場合、識別プロセスは F_{od} クロックレートで開始し、SDIO_CMD ライン出力ドライバはオープンドレインでなくプッシュプルドライバです。登録プロセスは、次のように行われます。

1. バスがアクティベートされます。
2. SDIO カードホストは SD_APP_OP_COND (ACMD41) をブロードキャストします。
3. カードは、それぞれの動作条件レジスタの内容で応答します。
4. 互換性のないカードは、インアクティブ状態へ移行されます。
5. SDIO カードホストは、すべてのアクティブカードに ALL_SEND_CID (CMD2) をブロードキャストします。
6. カードは、一意のカード識別番号 (CID) を送り返し、識別状態に入ります。
7. SDIO カードホストは、アドレスのあるアクティブなカードに SET_RELATIVE_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードは、スタンバイ状態に遷移します。SDIO カードホストは、このコマンドを再発行して RCA を変更することができます。カードの RCA は、最新の値が割り当てられます。
8. SDIO カードホストは、すべてのアクティブなカードについて、ステップ 5~7 を繰り返します。

SD I/O カードの場合、登録プロセスは次のように行われます。

1. バスがアクティベートされます。
2. SDIO カードホストは IO_SEND_OP_COND (CMD5) を送信します。
3. カードは、それぞれの動作条件レジスタの内容で応答します。
4. 互換性のないカードは、インアクティブ状態に設定されます。
5. SDIO カードホストは、アドレスのあるアクティブなカードに SET_RELATIVE_ADDR (CMD3) を発行します。この新しいアドレスは相対カードアドレス (RCA) と呼ばれ、CID より短く、カードをアドレス指定します。これを割り当てられたカードは、スタンバイ状態に遷移します。SDIO カードホストは、このコマンドを再発行して RCA を変更することができます。カードの RCA は、最新の値が割り当てられます。

21.4.5 ブロック書込み

ブロック書込み (CMD24-27) では、1 つ以上のブロックデータがホストからカードに転送され、ホストによって各ブロックの最後に CRC が付加されます。ブロック書込みをサポートするカードは、WRITE_BL_LEN によって定義されたデータブロックを常に受け入れることができます。CRC が失敗した場合、カードは SDIO_D ライン上で失敗を示し、転送されたデータは破棄されて書き込まれず、それ以降に転送されたブロック (複数ブロック書込みモード時) はすべて無視されます。

ホストが、累積長がブロック長に合わない部分的なブロックを使用し、ブロックのミスアラインメントが許可されない場合 (CSD パラメータ WRITE_BLK_MISALIGN がセットされていない)、カードはミスアラインされた最初のブロック開始前にブロックミスアラインメントエラーを検出します。(ステータスレジスタの ADDRESS_ERROR ビットがセットされます)。ホストが書込み保護されたエリアに書き込もうとした場合も、書込み操作がアボートされます。ただし、この場合、カードは WP_VIOLATION ビットをセットします。

CID および CSD レジスタをプログラムする場合、以前のブロック長設定は不要です。転送されるデータも、CRC 保護されます。CSD または CID レジスタの一部が ROM に格納される場合、この変更不能部分が受信バッファの対応する部分と一致しなければなりません。これが一致しない場合、カードはエラーを報告し、レジスタの内容を変更しません。一部のカードでは、ブロックデータの書込みに時間がかかり、その時間が予測不能ことがあります。ブロックデータの受信と CRC チェックの完了後、カードは書込みを開始し、書込みバッファがフルで新しい WRITE_BLOCK コマンドから新しいデータを受け入れられない場合には、SDIO_D ラインをローに保ちます。ホストは、SEND_STATUS コマンド (CMD13) でいつでもカードのステータスをポーリングでき、カードはそれぞれのステータスで応答します。READY_FOR_DATA ステータスビットは、カードが新しいデータを受け入れられるかどうか、または書込み動作がまだ進行中かどうかを示します。ホストは、カードを切断状態にする CMD7 を発行してカードの選択を解除し (別のカードを選択するため)、書込み操作を中断せずに SDIO_D ラインを解放することができます。カードを再び選択したときにプログラミングがまだ進行中であり、書込みバッファが使用できない場合には、SDIO_D をローにプルすることによって、ビジー表示を再びアクティブにします。

21.4.6 ブロック読出し

ブロック読出しモードでは、データ転送の基本単位はブロックであり、その最大サイズは CSD (READ_BL_LEN) で定義されます。READ_BL_PARTIAL がセットされると、開始アドレスと終了アドレスが 1 つの物理ブロック (READ_BL_LEN で定義) に完全に含まれる小さなブロックを送信することもできます。各ブロックの最後には CRC を付加して、データの整合性を確保します。CMD17 (READ_SINGLE_BLOCK) は、ブロック読出しを開始し、転送完了後、カードは転送状態に戻ります。

CMD18 (READ_MULTIPLE_BLOCK) は、複数の連続したブロックの転送を開始します。

ホストは、複数ブロック操作内においてはタイプに関係なく、いつでも読出しをアボートできます。トランザクションは、送信停止コマンドを送信してアボートします。

カードが複数ブロック読出し操作中にエラー (範囲外、アドレスのミスアラインメント、内部エラーなど) を検出した場合は、データ転送を停止して、データ状態に留まります。その場合、ホストは、送信停止コマンドを送信して操作をアボートする必要があります。読出しエラーは、送信停止コマンドに対するレスポンスで報告されます。

カードが事前定義されたブロック数による複数ブロック操作で、最後のブロックを送信した後でホストが送信停止コマンドを送信した場合には、カードがすでにデータ状態ではないため、不正なコマンドとして応答されます。ホストが、累積長がブロック長に合わない部分的なブロックを使用し、ブロックのミスアラインメントが許可されない場合、カードはミスアラインされた最初のブロック開始前にブロックミスアラインメントエラーを検出します (ステータスレジスタの ADDRESS_ERROR ビットがセットされます)。

21.4.7 ストリームアクセス、ストリーム書込み、およびストリーム読出し (MultiMediaCard のみ)

ストリームモードでは、データはバイト単位で転送され、各ブロックの最後に CRC は付加されません。

ストリーム書込み (MultiMediaCard のみ)

WRITE_DAT_UNTIL_STOP (CMD20) は、SDIO カードホストからカードへのデータ転送を指定されたアドレスで開始し、SDIO カードホストが停止コマンドを発行するまで続けます。部分的なブロックが許可される場合 (CSD パラメータ WRITE_BL_PARTIAL がセットされている)、データストリームはカードアドレス空間内の任意のアドレスで開始および停止できます。そうでない場合は、ブロック境界でのみ開始と停止が可能です。転送データ量を事前に決めることができないため、CRC は使用できません。データ送信中にメモリ範囲の終わりに達し、SD カードホストから停止コマンドが送信されなかった場合、その後に転送されるデータは破棄されます。

ストリーム書込み操作の最大クロック周波数は、カード固有データレジスタの次の式で与えられます。

$$\text{Maximumspeed} = \min(\text{TRANSPEED}, \frac{(8 \times 2^{\text{writeblen}})(-\text{NSAC})}{\text{TAAC} \times \text{R2WFACTOR}})$$

- Maximumspeed = 書込み最大周波数
- TRANSPEED = データ転送最大速度
- writeblen = 書込みデータブロック最大長
- NSAC = データ読出しアクセス時間 2 (CLK サイクル数)
- TAAC = データ読出しアクセス時間 1
- R2WFACTOR = 書込み速度係数

ホストがより高い周波数を使用しようとする、カードはデータを処理できずにプログラミングを停止し、ステータスレジスタの OVERRUN エラービットをセットすることがあります。この場合、それ以降のデータ転送は無視され、停止コマンドを待ちます (データ受信状態で)。ホストが書込み保護領域に書き込もうとした場合も、書込み操作がアボートされます。ただし、この場合、カードは WP_VIOLATION ビットをセットします。

ストリーム読出し (MultiMediaCard のみ)

READ_DAT_UNTIL_STOP (CMD11) は、ストリーム指向のデータ転送を制御します。

このコマンドは、SDIO カードホストが STOP_TRANSMISSION (CMD12) を送信するまで、指定されたアドレスからデータを送信するようにカードに指示します。停止コマンドには、シリアルコマンド送信による実行の遅延があり、データ転送は停止コマンドの終了ビット後に停止します。データ送信中にメモリ範囲の終わりに達し、SDIO カードホストから停止コマンドが送信されなかった場合、その後に転送されるデータは未定義と考慮されます。

ストリーム読出し操作の最大クロック周波数は次の式によって与えられ、カード固有データレジスタのフィールドを使用します。

$$\text{Maximumspeed} = \text{MIN}(\text{TRANSPEED}, \frac{(8 \times 2^{\text{readblen}})(-\text{NSAC})}{\text{TAAC} \times \text{R2WFACTOR}})$$

- Maximumspeed = 読出し最大周波数
- TRANSPEED = データ転送最大速度
- readblen = 読出しデータブロック最大長
- writeblen = 書込みデータブロック最大長
- NSAC = データ読出しアクセス時間 2 (CLK サイクル数)
- TAAC = データ読出しアクセス時間 1
- R2WFACTOR = 書込み速度係数

ホストがより高い周波数を使用しようとする、カードはデータ転送を維持できません。これが発生した場合、カードはステータスレジスタの UNDERRUN エラービットをセットし、送信をアボートし、データ状態で停止コマンドを待ちます。

21.4.8 消去：グループ消去とセクタ消去

MultiMediaCard の消去可能単位は、消去グループです。消去グループは、カードの基本的な書込み可能単位である書込みブロックで分けます。消去グループのサイズはカード固有のパラメータであり、CSD で定義されます。

ホストは連続した範囲の消去グループを消去することができます。消去プロセスは、3 ステップのシーケンスで開始します。

はじめにホストは、ERASE_GROUP_START (CMD35) コマンドを使用して範囲の開始アドレスを定義し、次に、ERASE_GROUP_END (CMD36) コマンドを使用して範囲の最後のアドレスを定義し、最後に、ERASE (CMD38) コマンドを発行して消去プロセスを開始します。消去コマンドのアドレスフィールドは、バイト単位の消去グループアドレスです。カードは、消去グループサイズ未満のすべての LSB を無視して、アドレスを効果的に消去グループの境界まで切り捨てます。

消去コマンドが誤った順序で受信されると、カードはステータスレジスタの ERASE_SEQ_ERROR ビットをセットし、シーケンス全体をリセットします。

シーケンス外 (SEND_STATUS を除き、どちらの消去コマンドでもない) のコマンドを受信した場合、カードはステータスレジスタの ERASE_RESET ステータスビットをセットして消去シーケンスをリセットし、最後のコマンドを実行します。

消去範囲に書込み保護ブロックが含まれていると、それらのブロックは消去されず、保護されていないブロックだけが消去されます。ステータスレジスタの WP_ERASE_SKIP ステータスビットがセットされます。

カードは、SDIO_D をローに保持することによって、消去が進行中であることを示します。実際の消去時間はかなり長くなることがあり、ホストは CMD7 を発行して、カードを選択解除できます。

21.4.9 ワイドバス選択または選択解除

ワイドバス (4 ビットバス幅) 操作モードは、SET_BUS_WIDTH (ACMD6) を使用して選択または選択解除されます。パワーアップ後または GO_IDLE_STATE (CMD0) 後のデフォルトのバス幅は、1 ビットです。SET_BUS_WIDTH (ACMD6) は、転送状態でのみ有効です。すなわち、SELECT/DESELECT_CARD (CMD7) によって選択された後でなければバス幅を変更することはできません。

21.4.10 保護管理

SDIO カードホストモジュールでは、カードの書き込み保護方法が 3 種類サポートされています。

1. 内部カード書き込み保護（カード側の責任）
2. 機械的書き込み保護スイッチ（SDIO カードホストモジュール側のみ）の責任）
3. パスワードで保護されたカードロック操作

内部カード書き込み保護

カードデータは、書き込みや消去から保護できます。CSD の永久的または一時的書き込み保護ビットをセットすると、製造業者やコンテンツプロバイダによってカード全体を永久的に保護することができます。セクタのグループの書き込み保護をサポートするカードの場合、CSD の WP_GRP_ENABLE ビットをセットするとデータの一部を保護することができ、書き込み保護はアプリケーションによって変更できます。書き込み保護は、CSD で指定された WP_GRP_SIZE セクタ単位です。SET_WRITE_PROT および CLR_WRITE_PROT コマンドは、アドレス指定されたグループの保護を制御します。SEND_WRITE_PROT コマンドは、1 ブロックの読出しコマンドと似ています。カードは、32 個の書き込み保護ビット（指定されたアドレスから始まる 32 個の書き込み保護グループを表す）に続いて、16 個の CRC ビットを含むデータブロックを送信します。書き込み保護コマンドのアドレスフィールドは、バイト単位のグループアドレスです。

カードは、グループサイズ未満のすべての LSB を無視します。

機械的書き込み保護スイッチ

カードの横にある機械的なスライドタブによって、カードの書き込み保護をセット／クリアできます。スライドタブが開位置にあるときにはカードは書き込み保護され、閉位置にあるときには、カードの内容を変更できます。ソケット側の対応するスイッチは、カードが書き込み保護されていることを SDIO カードホストモジュールに示します。カードを保護するのは、SDIO カードホストモジュールです。書き込み保護スイッチの位置は、カードの内部回路にはわかりません。

パスワード保護

パスワード保護機能によって、SDIO カードホストモジュールはカードをパスワードでロック／アンロックできます。パスワードは 128 ビットの PWD レジスタに格納され、そのサイズは 8 ビットの PWD_LEN レジスタで設定されます。これらのレジスタは不揮発性のため、パワーサイクルでは消去されません。ロックされたカードは、特定のコマンドに応答し、それを実行します。すなわち、SDIO カードホストモジュールは、リセット、初期化、選択、およびステータスの問い合わせを行うことはできますが、カード上のデータにアクセスすることはできません。パスワードが設定されると（PWD_LEN の値がゼロ以外）、カードはパワーアップ後に自動的にロックされます。CSD および CID レジスタの書き込みコマンド同様、ロック／アンロックコマンドは転送状態でのみ使用できます。この状態では、コマンドはアドレス引数を含まず、カードは使用前に選択しなければなりません。カードロック／アンロックコマンドは、通常の 1 ブロック書き込みコマンドの構造とバストランザクシオンタイプとなっています。転送されるデータブロックは、コマンドに必要なすべての情報（パスワード設定モード、PWD 自体、およびカードのロック／アンロック）を含みます。コマンドデータブロックサイズは、カードロック／アンロックコマンド送信前に、SDIO カードホストモジュールによって定義され、表 115 に示す構造となっています。

ビット設定は、次のとおりです。

- ERASE : セットすると、強制的に消去操作が行われます。他のすべてのビットはゼロでなければならず、コマンドバイトだけが送信されます。
- LOCK_UNLOCK : セットすると、カードをロックします。LOCK_UNLOCK は SET_PWD と同時にセットできますが、CLR_PWD と同時にセットすることはできません。
- CLR_PWD : セットすると、パスワードデータをクリアします。
- SET_PWD : セットすると、パスワードデータをメモリに保存します。
- PWD_LEN : パスワードの長さ (バイト数) を定義します。
- PWD : パスワード (コマンドに応じて、新しいパスワードまたは現在使用中のパスワード)。

以下のセクションでは、パスワードのセット／リセット、カードのロック／アンロック、および強制消去のコマンドシーケンスを説明します。

パスワードの設定

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード、8 ビットの PWD_LEN、および新しいパスワードのバイト数によって定義されます。パスワードの書き換えが完了すると、ブロックサイズでは古いパスワードと新しいパスワードの両方がコマンドとともに送信されることを考慮する必要があります。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (SET_PWD=1)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。パスワードの書き換えが完了すると、長さ値 (PWD_LEN) には古いパスワードと新しいパスワードの両方の長さが含まれ、PWD フィールドには、古いパスワード (現在使用中) に続いて新しいパスワードが含まれます。
4. パスワードが一致すると、新しいパスワードとそのサイズがそれぞれ PWD と PWD_LEN フィールドに保存されます。送信された古いパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、パスワードは変更されません。

パスワード長フィールド (PWD_LEN) は、パスワードが現在設定されているかどうかを示します。このフィールドがゼロ以外のときには、パスワードが設定されていて、カードはパワーアップ後に自身をロックします。LOCK_UNLOCK ビットをセットするか (パスワードの設定時に)、カードをロックする追加のコマンドを送信すると、現在のパワーセッションでただちにカードをロックすることが可能です。

パスワードのリセット

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード、8 ビットの PWD_LEN、および現在使用中のパスワードのバイト数によって定義されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (CLR_PWD=1)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。LOCK_UNLOCK ビットは無視されます。

4. パスワードが一致すると、PWD フィールドはクリアされ、PWD_LEN は 0 にセットされます。送信されたパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、パスワードは変更されません。

カードのロック

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード (表 115 のバイト 0)、8 ビットの PWD_LEN、および現在のパスワードのバイト数で定義されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (LOCK_UNLOCK = 1)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。
4. パスワードが一致すると、カードはロックされ、カードステータスレジスタの CARD_IS_LOCKED ステータスビットがセットされます。送信されたパスワードが予期されたパスワードとサイズや内容が一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、ロックは失敗します。

パスワードの設定とカードのロックは、同じシーケンスで行うことが可能です。この場合、SDIO カードホストモジュールは、パスワードの設定に必要なすべてのステップを実行します (626 ページの [パスワードの設定](#) を参照)。ただし、新しいパスワードコマンドを送信するときには、ステップ 3 で LOCK_UNLOCK ビットをセットする必要があります。

パスワードが以前に設定されているときには (PWD_LEN が 0 でない)、カードはパワーオンリセット後に自動的にロックされます。ロックされたカードをロックしようとしたり、パスワードが設定されていないカードをロックしようすると、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされます。

カードのアンロック

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. 送信するブロック長 (SET_BLOCKLEN、CMD16) を定義します。8 ビットのカードロック／アンロックモード (表 115 のバイト 0)、8 ビットの PWD_LEN、および現在のパスワードのバイト数で定義されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータブロックサイズとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (LOCK_UNLOCK = 0)、長さ (PWD_LEN)、およびパスワード (PWD) 自体を示します。
4. パスワードが一致すると、カードはアンロックされ、カードステータスレジスタの CARD_IS_LOCKED ステータスビットがクリアされます。送信されたパスワードのサイズまたは内容が予期されたパスワードと一致しないときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、カードはロックされたままとなります。

アンロック機能は、現在のパワーセッションでのみ有効です。PWD フィールドがクリアされていないときには、カードは次のパワーアップ時に自動的にロックされます。

アンロックされたカードをアンロックしようすると、アンロックは失敗し、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされます。

強制消去

ユーザがパスワード (PWD の内容) を忘れた場合は、カード上のすべてのデータをクリアした後で、カードにアクセスすることができます。この強制消去操作では、すべてのカードデータとすべてのパスワードデータが消去されます。

1. まだカードが選択されていない場合は、カードを選択します (SELECT/DESELECT_CARD、CMD7)。
2. ブロック長 (SET_BLOCKLEN、CMD16) を 1 バイトに設定します。8 ビットのカードロック／アンロックバイト (表 115 のバイト 0) だけが送信されます。
3. 16 ビットの CRC を含めた、データライン上に適切なデータバイトとともに LOCK/UNLOCK (CMD42) を送信します。データブロックは、モード (ERASE = 1) を示します。他のすべてのビットはゼロでなければなりません。
4. ERASE ビットがデータフィールドでセットされた唯一のビットであるときには、PWD および PWD_LEN フィールドを含め、すべてのカード内容が消去され、カードはロックされなくなります。他にもセットされているビットがあるときには、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされ、カードはすべてのデータを保持し、ロックされたままとなります。

アンロックされたカードを強制消去しようすると、カードステータスレジスタの LOCK_UNLOCK_FAILED エラービットがセットされます。

21.4.11 カードステータスレジスタ

レスポンスフォーマット R1 には、カードステータスと呼ばれる 32 ビットのフィールドがあります。このフィールドは、カードステータス情報 (ローカルステータスレジスタに格納可能) をホストに送信するためのものです。特に記載がない限り、ステータスエントリは、常に、以前に発行されたコマンドに関連するものとします。

表 102 に、さまざまなステータスエントリを示します。表中のタイプおよびクリア条件フィールドは、次のように略記されます。

タイプ :

- E : エラービット。
- S : ステータスビット。
- R : 実際のコマンドレスポンスに対して検出され、セットされます。
- X : コマンド実行中に検出され、セットされます。SDIO カードホストは、ステータスコマンドを発行してカードにポーリングし、これらのビットを読み出します。

クリア条件 :

- A : カードの現在の状態によります。
- B : 常に前のコマンドに関連しています。有効なコマンドを受信すると、クリアされます (1 コマンドの遅延)。
- C : 読み出すとクリアされます。

表 102. カードステータス

ビット	識別子	タイプ	値	説明	クリア条件
31	ADDRESS_OUT_OF_RANGE	E R X	'0' = エラーなし '1' = エラー	コマンドアドレス引数が、このカードの許容範囲を超えています。 複数ブロックまたはストリーム読み出し／書き込み操作は、有効なアドレスから開始していますが、カード容量を超える読み込みまたは書き込みをしようとしています。	C
30	ADDRESS_MISALIGN	-	'0' = エラーなし '1' = エラー	現在設定されているブロック長にしたがってコマンドアドレス引数が指している最初のデータブロックは、カードの物理ブロックに合っていない。 複数ブロックの読み出し／書き込み操作は、(有効なアドレス／ブロック長の組み合わせから開始している場合でも) カードの物理ブロックに合っていないデータブロックの読み込みまたは書き込みをしようとしています。	C
29	BLOCK_LEN_ERROR	-	'0' = エラーなし '1' = エラー	SET_BLOCKLEN コマンドの引数がカードの許容する最大値を超えているか、以前に定義されたブロック長が現在のコマンドに対して無効です (ホストが書き込みコマンドを発行し、現在のブロック長がカードの許容する最大値より小さく、部分的なブロックの書き込みが許可されない場合など)。	C
28	ERASE_SEQ_ERROR	-	'0' = エラーなし '1' = エラー	消去コマンドのシーケンス中にエラーが発生しました。	C
27	ERASE_PARAM	E X	'0' = エラーなし '1' = エラー	消去に対して無効な消去グループが選択されました。	C
26	WP_VIOLATION	E X	'0' = エラーなし '1' = エラー	書き込み保護されたブロックをプログラムしようとした。	C
25	CARD_IS_LOCKED	S R	0 = カードがアンロックされています。 1 = カードがロックされています。	セットされると、カードがホストによってロックされていることを示します。	A
24	LOCK_UNLOCK_FAILED	E X	'0' = エラーなし '1' = エラー	ロック／アンロックコマンドで、シーケンスエラーまたはパスワードエラーが発生したときにセットされます。	C
23	COM_CRC_ERROR	E R	'0' = エラーなし '1' = エラー	以前のコマンドの CRC チェックが失敗しました。	B
22	ILLEGAL_COMMAND	E R	'0' = エラーなし '1' = エラー	カードの状態に対して不正なコマンドです。	B
21	CARD_ECC_FAILED	E X	'0' = 成功 '1' = 失敗	カードの内部 ECC が適用されましたが、データの訂正に失敗しました。	C
20	CC_ERROR	E R	'0' = エラーなし '1' = エラー	(標準では定義されていません) ホストコマンドに関係ないカードエラーが発生しました。	C

表 102. カードステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
19	ERROR	E X	'0'= エラーなし 1= エラー	(標準では定義されていません) 最後のホストコマンドの実行に関係した一般カードエラーが検出されました (読出しまたは書き込みの失敗など)。	C
18	予約済み				
17	予約済み				
16	CID/CSD_OVERWRITE	E X	'0'= エラーなし '1'= エラー	次のいずれかのエラーです。 – CID レジスタはすでに書き込まれていて、上書きできません。 – CSD の読出し専用セクションが、カードの内容と一致しません。 – 不正なコピー (オリジナルとしてセット) または永久的な WP ビット (保護されない) を元に戻そうとしました。	C
15	WP_ERASE_SKIP	E X	0= 保護されていません 1= 保護されています	既存の書き込みが理由で、部分的なアドレス空間だけが消去されたときにセットされます。	C
14	CARD_ECC_DISABLED	S X	0= 有効 1= 無効	内部 ECC を使用せずに、コマンドが実行されました。	A
13	ERASE_RESET	-	0= クリア 1= セット	シーケンス外の消去コマンド (CMD35、CMD36、CMD38、または CMD13 以外) が受信されたため、実行前に消去シーケンスがクリアされました。	C
12:9	CURRENT_STATE	S R	0 = アイドル 1 = レディ状態 2 = 識別 3 = スタンバイ 4 = 転送 5 = データ 6 = 受信 7 = プログラム 8 = Dis 9 = Btst 10-15 = 予約済み	コマンドを受信したときのカードの状態。コマンドの実行によってカードの状態が変化する場合、次のコマンドのレスポンスで、ホストに状態が知らされます。4 ビットは、0 から 15 までの2進数として解釈されます。	B
8	READY_FOR_DATA	S R	'0'= 非レディ状態 '1'=レディ状態	バス上のバッファ空信号に対応します。	-
7	SWITCH_ERROR	E X	'0'= エラーなし 1= スイッチエラー	セットされた場合に、カードは SWICHTH コマンドで要求された予期されたモードに切り替わりませんでした。	B
6	予約済み				
5	APP_CMD	S R	0= 無効 1= 有効	カードは ACMD を期待しているか、コマンドが ACMD として解釈されたという指示を期待します。	C
4	SD I/O カード用に予約済み				
3	AKE_SEQ_ERROR	E R	'0'= エラーなし 1= エラー	認証プロセスのシーケンスにエラーがあります。	C

表 102. カードステータス (続き)

ビット	識別子	タイプ	値	説明	クリア条件
2	アプリケーション固有のコマンド用に予約済み。				
1	製造業者のテストモード用に予約済み。				
0					

21.4.12 SD ステータスレジスタ

SD ステータスは、SD メモリカード独自の機能に関連するステータスビットを含み、将来のアプリケーション固有の用途で使用できます。SD ステータスのサイズは、512 ビットのデータブロックです。このレジスタの内容は、ACMD13 が送信された場合に (CMD55 に続いて CMD13)、SDIO カードホストに送信されます。ACMD13 は、転送状態のカード (カードが選択されている) にのみ送信できます。

表 103 に、SD ステータスレジスタのさまざまなエントリを示します。表中のタイプおよびクリア条件フィールドは、次のように略記されます。

タイプ :

- E : エラービット。
- S : ステータスビット。
- R : 実際のコマンドレスポンスに対して検出され、セットされます。
- X : コマンド実行中に検出され、セットされます。SDIO カードホストは、ステータスコマンドを発行してカードにポーリングし、これらのビットを読み出します。

クリア条件 :

- A : カードの現在の状態によります。
- B : 常に前のコマンドに関連しています。有効なコマンドを受信すると、クリアされます (1 コマンドの遅延)。
- C : 読み出すとクリアされます。

表 103. SD ステータス

ビット	識別子	タイプ	値	説明	クリア条件
511 : 510	DAT_BUS_WIDTH	S R	00= 1 (デフォルト) 01= 予約済み 10= 4 ビット幅 11= 予約済み	SET_BUS_WIDTH コマンドによって定義された、現在定義されているバス幅を示します。	A
509	SECURED_MODE	S R	0= セキュアモードではありません。 1= セキュアモードです。	カードはセキュア操作モードになっています (「SD セキュリティ仕様」を参照)。	A
508 : 496	予約済み				
495 : 480	SD_CARD_TYPE	S R	00xxh= 物理仕様 Ver1.01-2.00 で定義された SD メモリカード ("x"= 無視) 現在定義されているカードは、次のとおりです。 0000= 通常の SD RD/WR カード 0001= SD ROM カード	将来は、8 つの LSB を使用して、さまざまな種類の SD メモリカードを定義する予定です (各ビットは異なる SD タイプを定義します)。8 つの MSB は、現在の SD 物理レイヤ仕様に準拠しない SD カードを定義するために使用します。	A

表 103. SD ステータス（続き）

ビット	識別子	タイプ	値	説明	クリア条件
479 : 448	SIZE_OF_PROTECTED_AREA	S R	保護領域のサイズ（下記を参照）	（下記を参照）	A
447 : 440	SPEED_CLASS	S R	カードの速度クラス（下記を参照）	（下記を参照）	A
439 : 432	PERFORMANCE_MOVE	S R	1 [MB/s] ステップが表す移動性能（下記を参照）	（下記を参照）	A
431:428	AU_SIZE	S R	AU のサイズ（下記を参照）	（下記を参照）	A
427:424	予約済み				
423:408	ERASE_SIZE	S R	一度に消去される AU の数	（下記を参照）	A
407:402	ERASE_TIMEOUT	S R	UNIT_OF_ERASE_AU で指定された領域を消去する際のタイムアウト値	（下記を参照）	A
401:400	ERASE_OFFSET	S R	消去時間に追加される固定オフセット値	（下記を参照）	A
399:312	予約済み				
311:0	製造業者のために予約済み。				

SIZE_OF_PROTECTED_AREA

このフィールドの設定は、標準カードと大容量カードで異なります。標準容量のカードでは、保護領域の容量は次のように計算されます。

保護領域 = SIZE_OF_PROTECTED_AREA * MULT * BLOCK_LEN

SIZE_OF_PROTECTED_AREA は、MULT*BLOCK_LEN 単位で指定されます。

大容量のカードでは、保護領域の容量は、このフィールドで指定されます。

保護領域 = SIZE_OF_PROTECTED_AREA

SIZE_OF_PROTECTED_AREA は、バイト単位で指定されます。

SPEED_CLASS

この 8 ビットフィールドは速度クラスを示し、値は $P_W/2$ で求めることができます (P_W は書込み性能)。

表 104. 速度クラスコードフィールド

SPEED_CLASS	値の定義
00h	クラス 0
01h	クラス 2
02h	クラス 4
03h	クラス 6
04h – FFh	予約済み

PERFORMANCE_MOVE

この 8 ビットフィールドは、Pm（移動パフォーマンス）を示し、値は、1 [MB/sec] ステップで設定できます。カードが使用済みの RU（記録ユニット）を移動しない場合は、Pm を無限大であるとみなします。このフィールドを FFh にセットすると、無限を意味します。

表 105. 移動パフォーマンスフィールド

PERFORMANCE_MOVE	値の定義
00h	定義されません。
01h	1 [MB/sec]
02h	02h 2 [MB/sec]
-----	-----
FEh	254 [MB/sec]
FFh	無限

AU_SIZE

この 4 ビットフィールドは、AU サイズを示し、値は、16 KB から 2 の累乗で選択できます。

表 106. AU_SIZE フィールド

AU_SIZE	値の定義
00h	定義されません。
01h	16 KB
02h	32 KB
03h	64 KB
04h	128 KB
05h	256 KB
06h	512 KB
07h	1 MB
08h	2 MB
09h	4 MB
Ah – Fh	予約済み

最大 AU サイズはカード容量に依存し、表 107 に示されています。カードは、RU サイズと最大 AU サイズの間の任意の AU サイズに設定できます。

表 107. 最大 AU サイズ

容量	16 MB-64 MB	128 MB-256 MB	512 MB	1 GB-32 GB
最大 AU サイズ	512 KB	1 MB	2 MB	4 MB

ERASE_SIZE

この 16 ビットフィールドは、NERASE を示します。AU の NERASE 数が消去されるときの、タイムアウト値は ERASE_TIMEOUT によって指定されます ([ERASE_TIMEOUT](#) を参照)。ホストは、1 回の操作で消去される正しい AU の数を決定する必要があります。それにより、ホストは消去操作の進捗状況を表示することができます。このフィールドが 0 にセットされた場合、消去タイムアウト計算はサポートされません。

表 108. 消去サイズフィールド

ERASE_SIZE	値の定義
0000h	消去タイムアウトの計算はサポートされません。
0001h	1 AU
0002h	2 AU
0003h	3 AU
-----	-----
FFFFh	65535 AU

ERASE_TIMEOUT

この 6 ビットフィールドは、TERASE を示し、値は、ERASE_SIZE によって指定された複数の AU が消去されるときのオフセットからの消去タイムアウトを示します。ERASE_TIMEOUT の範囲は、63秒までの間で定義でき、カード製造業者は実装に応じて、ERASE_SIZE と ERASE_TIMEOUT の任意の組み合わせを選ぶことができます。ERASE_TIMEOUT を決めると、ERASE_SIZE が決まります。

表 109. 消去タイムアウトフィールド

ERASE_TIMEOUT	値の定義
00	消去タイムアウトの計算はサポートされません。
01	1 [sec]
02	2 [sec]
03	3 [sec]
-----	-----
63	63 [sec]

ERASE_OFFSET

この 2 ビットフィールドは、TOFFSET を示し、4 つの値から 1 つを選択できます。このフィールドは、ERASE_SIZE および ERASE_TIMEOUT フィールドが 0 にセットされている場合は意味を持ちません。

表 110. 消去オフセットフィールド

ERASE_OFFSET	値の定義
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

21.4.13 SD I/O モード

SD I/O 割込み

SD インタフェースのピンには、SD I/O カードから MultiMediaCard/SD モジュールへの割込みを可能にする割込み機能が用意されています。ピン 8 は、4 ビット SD モードで動作するときに SDIO_D1 として使用され、MultiMediaCard/SD モジュールへのカード割込みを示します。割込みの使用は、各カードまたはカード内の各機能でオプションとなっています。SD I/O 割込みはレベル対応であるため、割込みラインは MultiMediaCard/SD モジュールによって認識され、割込みが処理されるか、割込み期間の終了によってネゲートされるまで、アクティブ（ロー）に保たれる必要があります。MultiMediaCard/SD モジュールが割込み処理を終えると、SD I/O カードの内部レジスタの該当するビットへの I/O 書き込みによって割込みステータスビットがクリアされます。すべての SD I/O カードの割込み出力はアクティブローであり、MultiMediaCard/SD モジュールはすべてのデータライン (SDIO_D[3:0]) 上に、外部プルアップ抵抗を提供する必要があります。MultiMediaCard/SD モジュールは、割込み期間に限って、割込み検出器にピン 8 (SDIO_D/IRQ) のレベルをサンプリングします。その他の状況では、MultiMediaCard/SD モジュールはこの値を無視します。

割込み期間は、メモリと I/O の両方の操作に適用されます。単一ブロックの操作の割込み期間の定義は、複数ブロックのデータ転送の定義とは異なります。

SD I/O のサスペンドとレジューム

マルチファンクション SD I/O または I/O とメモリの両方の機能を持つカードでは、MMC/SD バスへのアクセスを共有する複数のデバイス (I/O とメモリ) があります。複数デバイス間で MMC/SD モジュールへのアクセスを共有するため、SD I/O およびコンボカードはオプションでサスペンド／レジュームの概念を実装します。カードがサスペンド／レジュームをサポートするときには、MMC/SD モジュールは 1 つの機能またはメモリへのデータ転送を一時的に中止（サスペンド）して、別の機能またはメモリへの、より優先順位の高い転送用にバスを解放できます。この優先順位の高い転送が完了すると、本来の転送がアポートした所から再開（リスタート）されます。サスペンド／レジュームのサポートは、カードごとのオプションです。MMC/SD バス上でサスペンド／レジューム操作を行うには、MMC/SD モジュールは以下の手順を実行します。

1. SDIO_D [3:0] ラインを現在使用している機能を特定します。
2. 優先順位が低いトランザクションや時間のかかるトランザクションをサスペンドするように要求します。
3. トランザクションのサスペンドの完了を待ちます。
4. 優先順位の高いトランザクションを開始します。
5. 優先順位の高いトランザクションの完了を待ちます。
6. サスペンドされたトランザクションをリストアします。

SD I/O ReadWait

オプションの ReadWait (RW) 操作は、SD の 1 ビットおよび 4 ビットモードについてのみ定義されます。ReadWait 操作によって、MMC/SD モジュールは複数のレジスタを読み出していることをカードに知らせ (IO_RW_EXTENDED、CMD53)、データ転送を一時的にストールし、その間に MMC/SD モジュールは SD I/O デバイス内の機能にコマンドを送信できます。カードが ReadWait プロトコルをサポートしているかどうかを判断するには、MMC/SD モジュールが内部カードレジスタの機能ビットをテストする必要があります。ReadWait のタイミングは、割込み期間に基づいています。

21.4.14 コマンドおよびレスポンス

アプリケーション固有コマンドと汎用コマンド

SD カードホストモジュールシステムは、さまざまな種類のアプリケーションに標準インタフェースを提供するように設計されています。この環境では、特定の顧客やアプリケーション機能へのニーズがあります。これらの機能を実装するために、この標準ではアプリケーション固有コマンド (ACMD) と汎用コマンド (GEN_CMD) の 2 種類のコマンドが定義されています。

APP_CMD (CMD55) コマンドを受信すると、カードは次のコマンドがアプリケーション固有コマンドであると予想します。ACMD は、通常の MultiMediaCard コマンドと同じ構造を持っており、同じ CMD 番号を持つことができます。カードは、APP_CMD (CMD55) の後に現れるコマンドを ACMD として認識します。APP_CMD (CMD55) のすぐ後に続くコマンドが定義済みのアプリケーション固有コマンドでないときには、標準コマンドが使用されます。たとえば、カードで SD_STATUS (ACMD13) が定義されているときに APP_CMD (CMD55) の直後に CMD13 を受信すると、これは SD_STATUS (ACMD13) であると解釈されます。ただし、カードが APP_CMD (CMD55) の直後に CMD7 を受信し、カードで ACMD7 が定義されていないときには、これは標準 (SELECT/DESELECT_CARD) CMD7 であると解釈されます。

製造業者固有の ACMD の 1 つを使用するには、SD カードホストが以下の手順を実行する必要があります。

1. APP_CMD (CMD55) を送信します。
カードは MultiMediaCard/SD モジュールに응答して、APP_CMD ビットがセットされていることと、ACMD が予期されていることを示します。
2. 必要な ACMD を送信します。
カードは MultiMediaCard/SD モジュールに응答して、APP_CMD ビットがセットされていることと、受け取ったコマンドが ACMD として解釈されたことを示します。ACMD でないコマンドが送信されたときには、カードはそのコマンドを通常の MultiMediaCard コマンドとして扱い、カードステータスレジスタの APP_CMD ビットはクリアされたままとなります。

無効なコマンド (ACMD でも CMD でもない) が送信されたときには、標準の MultiMediaCard 不正コマンドエラーとして処理されます。

GEN_CMD のバストランザクションは、単一ブロックの読みまたは書き込みコマンド (WRITE_BLOCK、CMD24、または READ_SINGLE_BLOCK、CMD17) と同じです。この場合、引数はアドレスではなくデータ転送の方向を示し、データブロックはベンダ固有のフォーマットと意味を持ちます。

カードは、GEN_CMD (CMD56) を送信する前に選択されなければなりません (転送状態でなければなりません)。データブロックサイズは、SET_BLOCKLEN (CMD16) によって定義されます。GEN_CMD (CMD56) へのレスポンスは、R1b フォーマットです。

コマンドタイプ

アプリケーション固有コマンドと汎用コマンドは、どちらも、次の 4 つのタイプに分類されます。

- **ブロードキャストコマンド (BC)** : すべてのカードに送信されます。レスポンスは返されません。
- **レスポンス付きブロードキャストコマンド (BCR)** : すべてのカードに送信されます。レスポンスは、すべてのカードから同時に受信します。
- **アドレス指定 (ポイントツーポイント) コマンド (AC)** : 選択されたカードに送信されます。SDIO_D ライン上のデータ転送は含みません。
- **アドレス指定 (ポイントツーポイント) データ転送コマンド (ADTC)** : 選択されたカードに送信されます。SDIO_D ライン上のデータ転送は含みます。

コマンドフォーマット

コマンドフォーマットについては、[613 ページの表 95](#)を参照してください。

MultiMediaCard/SD モジュール用コマンド

表 111. ブロック指向書き込みコマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD23	ac	[31:16] 0 に セット [15:0] ブロック数	R1	SET_BLOCK_COUNT	複数ブロック読みまたは書き込みコマンドで転送されるブロック数を定義します。
CMD24	adtc	[31:0] データ アドレス	R1	WRITE_BLOCK	SET_BLOCKLEN コマンドによって選択されたサイズのブロックを書き込みます。
CMD25	adtc	[31:0] データ アドレス	R1	WRITE_MULTIPLE_BLOCK	STOP_TRANSMISSION まで、または、要求されたブロック数を受信するまで、データのブロックを書き込み続けます。
CMD26	adtc	[31:0] スタッ フビット	R1	PROGRAM_CID	カード識別レジスタのプログラミングです。このコマンドは、1 枚のカードにつき 1 度だけ発行する必要があります。カードには、最初のプログラミングの後にこの操作を防ぐためのハードウェアが含まれています。通常、このコマンドは製造業者用に予約されています。
CMD27	adtc	[31:0] スタッ フビット	R1	PROGRAM_CSD	CSD のプログラマブルビットのプログラミング用です。

表 112. ブロック指向書き込み保護コマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD28	ac	[31:0] データ アドレス	R1b	SET_WRITE_PROT	カードに書き込み保護機能がある場合、このコマンドはアドレス指定されたグループの書き込み保護ビットをセットします。書き込み保護のプロパティは、カード固有データ (WP_GRP_SZIE) でコード化されます。
CMD29	ac	[31:0] データ アドレス	R1b	CLR_WRITE_PROT	カードが書き込み保護機能を備えている場合、このコマンドはアドレス指定されたグループの書き込み保護ビットをクリアします。
CMD30	adtc	[31:0] 書き込み 保護データ アクセス	R1	SEND_WRITE_PROT	カードが書き込み保護機能を備えている場合、このコマンドはカードに書き込み保護ビットのステータスを送るように要求します。
CMD31	予約済み				

表 113. 消去コマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD32 ... CMD34		予約済み。古いバージョンの MultiMediaCard との下位互換性を保つため、これらのコマンドインデックスは使用できません。			
CMD35	ac	[31:0] データ アドレス	R1	ERASE_GROUP_START	消去対象として選択される範囲の最初の消去グループのアドレスを設定します。
CMD36	ac	[31:0] データ アドレス	R1	ERASE_GROUP_END	消去対象として選択される連続した範囲内の最後の消去グループのアドレスを設定します。
CMD37		予約済み。古いバージョンの MultiMediaCard との下位互換性を保つため、これらのコマンドインデックスは使用できません。			
CMD38	ac	[31:0] スタッ フ ビット	R1	ERASE	以前に選択されたすべての書き込みブロックを消去します。

表 114. I/O モードコマンド

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD39	ac	[31:16] RCA [15:15] レジスタ書き込みフラグ [14:8] レジスタ アドレス [7:0] レジスタデータ	R4	FAST_IO	8 ビット (レジスタ) データフィールドの読出しおよび書き込みに使用されます。このコマンドは、カードとレジスタをアドレス指定し、書き込みフラグがセットされている場合は書き込み用データを提供します。R4 レスポンスは、アドレス指定されたレジスタから読み出されたデータを含みます。このコマンドは、MultiMediaCard 標準で定義されていないアプリケーション依存レジスタにアクセスします。
CMD40	bcr	[31:0] スタッ フ ビット	R5	GO_IRQ_STATE	システムを割込みモードにします。
CMD41		予約済み			

表 115. カードのロック

CMD インデックス	タイプ	引数	レスポンス フォーマット	略語	説明
CMD42	adtc	[31:0] スタッ フ ビット	R1b	LOCK_UNLOCK	パスワードのセット／リセット、またはカードのロック／アンロックを行います。データブロックのサイズは、SET_BLOCK_LEN コマンドによって設定されます。
CMD43 ...CMD54		予約済み			

表 116. アプリケーション固有コマンド

CMD インデックス	タイプ	引数	レスポンスフォーマット	略語	説明
CMD55	ac	[31:16] RCA [15:0] スタッ ビット	R1	APP_CMD	次のコマンドビットが標準コマンドでなく、アプリケーション固有コマンドであることをカードに示します。
CMD56	adtc	[31:1] スタッ ビット [0] : RD/WR			汎用／アプリケーション固有コマンドに対し、データブロックをカードに転送するため、またはカードからデータを取得するために使用されます。データブロックのサイズは、SET_BLOCK_LEN コマンドによって設定されます。
CMD57 ...CMD59	予約済み。				
CMD60 ...CMD63	製造業者用に予約されています。				

21.5 レスポンスフォーマット

すべてのレスポンスは MCCMD コマンドライン SDIO_CMD 経由で送信されます。レスポンスの送信は、常に、レスポンスコードワードに対応するビット文字列の左のビットから始まります。コード長は、レスポンスのタイプに依存します。

レスポンスは、常にスタートビット（常に 0）から始まり、送信方向を示すビットが続きます（card=0）。下の表で x で示されている値は、可変エントリを示します。R3 レスポンスを除くすべてのレスポンスは、CRC によって保護されます。すべてのコマンドコードワードは、終了ビット（常に 1）で終わります。

レスポンスには 5 つのタイプがあります。フォーマットは、次のとおりです。

21.5.1 R1（ノーマルレスポンスコマンド）

コード長 = 48 ビット。45:40 ビットはレスポンスされるコマンドのインデックスを示します。この値は 2 進コード番号（0～63）として解釈されます。カードのステータスは 32 ビットにコード化されます。

表 117. R1 レスポンス

ビット位置	幅（ビット）	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	X	コマンドインデックス
[39:8]	32	X	カードステータス
[7:1]	7	X	CRC7
0	1	1	終了ビット

21.5.2 R1b

R1 と同じですが、オプションとしてデータライン上にビジー信号を出力します。カードは、コマンドを受信する前の状態に基づいて、これらのコマンドを受信した後、ビジーになることがあります。

21.5.3 R2 (CID、CSD レジスタ)

コード長 = 136 ビット。CID レジスタの内容は、CMD2 および CMD10 コマンドのレスポンスとして送信されます。CSD レジスタの内容は、CMD9 のレスポンスとして送信されます。CID および CSD のビット [127... 1] だけが送信され、これらのレジスタの予約ビット [0] は、レスポンスの終了ビットで置き換えられます。カードは、MCDAT をローに維持することによって、消去が進行中であることを示します。実際の消去時間はかなり長くなることがあり、ホストは CMD7 を発行して、カードを選択解除できます。

表 118. R2 レスポンス

ビット位置	幅 (ビット)	値	説明
135	1	0	スタートビット
134	1	0	送信ビット
[133:128]	6	'111111'	コマンドインデックス
[127:1]	127	X	カードステータス
0	1	1	終了ビット

21.5.4 R3 (OCR レジスタ)

コード長 : 48 ビット。OCR レジスタの内容は、CMD1 へのレスポンスとして送信されます。レベルのコーディングは、次のようになります。制限された電圧ウィンドウ=ロー、カードビジー=ロー。

表 119. R3 レスポンス

ビット位置	幅 (ビット)	値	説明
47	1	0	スタートビット
46	1	0	送信ビット
[45:40]	6	'111111'	予約済み
[39:8]	32	X	OCR レジスタ
[7:1]	7	'1111111'	予約済み
0	1	1	終了ビット

21.5.5 R4（高速 I/O）

コード長：48 ビット。引数フィールドは、アドレス指定されたカードの RCA、読出しまたは書込みレジスタアドレス、およびその内容を含みます。

表 120. R4 レスpons

ビット位置		幅（ビット）	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	'100111'	CMD39
[39:8] 引数フィールド	[31:16]	16	X	RCA
	[15:8]	8	X	レジスタアドレス
	[7:0]	8	X	読出しレジスタの内容
[7:1]		7	X	CRC7
0		1	1	終了ビット

21.5.6 R4b

SD I/O のみ：CMD5 を受信した SDIO カードは、一意の SDIO レスpons R4 で応答します。フォーマットは、次のとおりです。

表 121. R4b レスpons

ビット位置		幅（ビット）	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	x	予約済み
[39:8] 引数フィールド	39	16	X	カードはレディ状態
	[38:36]	3	X	I/O 機能数
	35	1	X	現在のメモリ
	[34:32]	3	X	スタッフビット
	[31:8]	24	X	I/O ORC
[7:1]		7	X	予約済み
0		1	1	終了ビット

SD I/O カードが CMD5 を受信すると、カードの I/O 部分は、以降のすべてのコマンドに対して通常どおりに応答できるようになります。I/O カード内の機能のこの I/O 有効は、カードがリセット、パワーサイクル、または I/O リセットに書き込む CMD52 を受信するまで、セットされたままです。SD メモリ専用カードは、CMD5 に応答することがありますので注意してください。メモリ専用カードの正しいレスponsは、現在のメモリ = 1 と I/O 機能数 = 0 です。SD メモリカード仕様 version 1.0 に準拠するように作られたメモリ専用カードは、CMD5 を不正なコマンドとして検出して、応答しません。I/O を認識できるホストは、CMD5 を送信します。カードがレスpons R4 で応答した場合、ホストはレスpons R4 に含まれるデータから、カードの構成を判断します。

21.5.7 R5（割込みリクエスト）

MultiMediaCard 専用です。コード長：48 ビット。ホストによってレスポンスが生成された場合、引数の RCA フィールドは 0x0 になります。

表 122. R5 レスpons

ビット位置		幅 (ビット)	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	'101000'	CMD40
[39:8] 引数フィールド	[31:16]	16	X	獲得したカードまたはホストの RCA [31:16]
	[15:0]	16	X	定義されていません。IRQ データに使われる場合があります。
[7:1]		7	X	CRC7
0		1	1	終了ビット

21.5.8 R6

SD I/O 専用です。メモリデバイスによる CMD3 への通常のレスポンスです。表 123 に示されています。

表 123. R6 レスpons

ビット位置		幅 (ビット)	値	説明
47		1	0	スタートビット
46		1	0	送信ビット
[45:40]		6	'101000'	CMD40
[39:8] 引数フィールド	[31:16]	16	X	獲得したカードまたはホストの RCA [31:16]
	[15:0]	16	X	定義されていません。IRQ データに使われる場合があります。
[7:1]		7	X	CRC7
0		1	1	終了ビット

カード [23:8] ステータスビットは、CMD3 が I/O 専用カードに送信されると変更されます。この場合、レスポンスの 16 ビットは、SD I/O 専用の値です。

- ビット [15] COM_CRC_ERROR
- ビット [14] ILLEGAL_COMMAND
- ビット [13] ERROR
- ビット [12:0] 予約済み

21.6 SDIO I/O カード固有の操作

以下の機能は、SD I/O 固有の操作です。

- SDIO_D2 信号による SDIO 読出しウェイト操作
- クロックの停止による SDIO 読出しウェイト操作
- SDIO サスペンド／レジューム操作（書込みおよび読出しサスペンド）
- SDIO 割込み

SDIO は、SDIO_DCTRL[11] ビットがセットされている場合のみ、これらの機能をサポートします。ただし、特定のハードウェア実装を必要としない読出しサスペンドを除きます。

21.6.1 SDIO_D2 信号による SDIO I/O 読出しウェイト操作

最初のブロックが受信される前に読出しウェイトインターバルを開始することができます。データパスが有効（SDIO_DCTRL[0] ビットがセット）、SDIO 固有操作が有効（SDIO_DCTRL[11] ビットがセット）、読出しウェイトが開始（SDIO_DCTRL[10] =0 および SDIO_DCTRL[8] =1）、データ方向がカードからSDIO（SDIO_DCTRL[1] = 1）のとき、DPSM はアイドルから ReadWait へ直接移行します。ReadWait では、DPSM は 2 SDIO_CK クロックサイクル後に、SDIO_D2 を 0 に駆動します。この状態では、RWSTOP ビット（SDIO_DCTRL[9]）をセットすると、DPSM は、さらに 2 SDIO_CK クロックサイクルの間ウェイト状態にとどまり、1 クロックサイクルの間 SDIO_D2 を 1 に駆動します（SDIO 仕様に従う）。その後、DPSM はカードからデータを受信するまで再びウェイトを開始します。DPSM は、読出しウェイト開始がセットされている場合でも、ブロックを受信中は読出しウェイトインターバルを開始しません。読出しウェイトインターバルは、CRC の受信後に開始されます。新しい読出しウェイト操作を開始するには、RWSTOP ビットをクリアする必要があります。読出しウェイトインターバル中は、SDIO は SDIO_D1 上で SDIO 割込みを検出することができます。

21.6.2 SDIO_CK の停止による SDIO 読出しウェイト操作

SDIO カードが先の読出しウェイト方法をサポートしない場合、SDIO は SDIO_CK を停止することで（SDIO_DCTRL は [セクション 21.6.1](#) に記載される方法と同じようにセットされますが、SDIO_DCTRL[10] =1 です）読出しウェイトを実行することができます。DPSM は、現在受信しているブロックの終了ビット後 2 SDIO_CK サイクル後にクロックを停止し、読出しウェイトスタートビットがセットされた後に再びクロックを開始します。

SDIO_CK が停止されると、任意のコマンドをカードに発行できます。読出しウェイトインターバル中、SDIO は SDIO_D1 上の SDIO 割込みを検出できます。

21.6.3 SDIO サスペンド／レジューム操作

カードにデータを送信している間、SDIO は書込み操作をサスペンドできます。SDIO_CMD[11] ビットがセットされ、現在のコマンドがサスペンドコマンドであることを CPSM に示します。CPSM はレスポンスを解析し、カードから ACK を受信すると（サスペンドの受入れ）、現在のブロックの CRC トークンの受信後にアイドル状態となることを DPSM に知らせます。

ハードウェアは、サスペンド操作（レジューム）を完了させるために送信しなければならない残りの送信ブロック数を保存しません。

書込み操作は、カードからサスペンドコマンドの ACK を受信したときに、DPSM を無効にするだけで（SDIO_DCTRL[0] =0）、ソフトウェアによってサスペンドできます。その場合、DPSM はアイドル状態に入ります。

読出しをサスペンドするには、DPSM は Wait_r 状態でサスペンドされる機能が完全なパケットを送信するのを待ってから、データ処理を停止します。アプリケーションは、FIFO が空になって、DPSM が自動的にアイドル状態になるまで、RxFIFO の読出しを続けます。

21.6.4 SDIO 割込み

SDIO_DCTRL[11] ビットがセットされると、SDIO_D1 ライン上で SDIO 割込みが検出されます。

21.7 CE-ATA 固有の操作

以下の機能は、CE-ATA 固有の操作です。

- CE-ATA デバイスを無効にするコマンド完了信号の送信。
- CE-ATA デバイスからのコマンド完了信号の受信。
- ステータスビットや割込みを使用して、CE-ATA コマンドの完了を CPU に通知。

SDIO は、SDIO_CMD[14] がセットされた場合、CE-ATA CMD61 コマンドに対してのみ、これらの操作をサポートします。

21.7.1 コマンド完了信号無効

コマンド完了信号無効は、"CMD 完了有効" ビット (SDIO_CMD[12]) がセットされておらず、"非割込み有効" ビット (SDIO_CMD[13]) がセットされている場合に、**short** レスポンスの受信から 8 ビットサイクル後に送信されます。

CPSM は Pend 状態に入り、コマンドシフトレジスタに無効シーケンス「00001」をロードし、コマンドカウンタに 43 をロードします。8 サイクル後、トリガは CPSM を送信状態へ移行します。コマンドカウンタが 48 に達すると、CPSM は、待ち受けるレスポンスがないのでアイドル状態になります。

21.7.2 コマンド完了信号有効

"CMD 完了有効" ビット (SDIO_CMD[12]) がセットされ、"非割込み有効" ビット (SDIO_CMD[13]) がセットされた場合、CPSM は Waitcpl 状態でコマンド完了信号を待ちます。

CMD ライン上で "0" を受信すると、CPSM はアイドル状態に入ります。7 ビットサイクルの間、新しいコマンドを送信することはできません。7 サイクルのうちの最後の 5 サイクルの間、CMD ラインはプッシュプルモードで "1" に駆動されます。

21.7.3 CE-ATA 割込み

コマンド完了は、ステータスビット SDIO_STA[23] によって CPU に通知されます。この静的ビットは、クリアビット SDIO_ICR[23] によってクリアできます。

SDIO_STA[23] ステータスビットは、マスクビット SDIO_MASKx[23] に応じて、各割込みライン上に割込みを生成することができます。

21.7.4 CMD61 のアボート

コマンド完了無効信号が送信されておらず、CMD61 をアボートする必要がある場合には、コマンドステートマシンを無効にする必要があります。コマンドステートマシンはアイドル状態になり、CMD12 コマンドを送信できるようになります。操作中は、コマンド完了無効信号は送信されません。

21.8 HW フロー制御

HW フロー制御機能は、FIFO アンダーラン (TX モード) およびオーバーラン (RX モード) エラーを回避するために使用します。

この機能は、SDIO_CK を停止し、SDIO ステートマシンをフリーズします。FIFO を使用してデータを送受信できない間、データ転送がストールされます。SDIOCLK によってクロック供給されるステートマシンだけが停止し、APB2 インタフェースは動作を続けます。したがって、フロー制御がアクティブであっても、FIFO がいっぱいになったり、空になることがあります。

HW フロー制御を有効にするには、SDIO_CLKCR[14] レジスタビットを 1 にセットする必要があります。リセット後、フロー制御は無効になります。

21.9 SDIO レジスタ

デバイスは、APB2 経由でアクセス可能な 32 ビット幅の制御レジスタによってシステムと通信します。

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。

21.9.1 SDIO 電源制御レジスタ (SDIO_POWER)

アドレスオフセット : 0x00
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																PWRCT		RL													
																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 **PWRCTRL** : 電源制御ビット

これらのビットは、カードクロックの現在の機能状態を定義するために使用されます。

00 : パワーオフ : カードへのクロック供給は停止されます。

01 : 予約済み

10 : 予約済み、パワーアップ

11 : パワーオン : カードにクロックが供給されます。

注 : このレジスタへの各書き込みアクセス間には、7 HCLK クロック周期以上が必要です。

データ書き込み後、3 SDIOCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込めません。

21.9.2 SDIO クロック制御レジスタ (SDIO_CLKCR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

SDIO_CLKCR レジスタは、SDIO_CK 出力クロックを制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
予約済み																	HWFC_EN	NEGEDGE	WID BUS		BYPASS	PWRSV	CLKEN	CLKDIV									
																	rW																

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **HWFC_EN** : HW フロー制御有効

0b : HW フロー制御は無効です。

1b : HW フロー制御は有効です。

HW フロー制御が有効なときの TXFIFOE および RXFIFOE 割込み信号の意味については、[セクション 21.9.11](#) の SDIO ステータスレジスタの定義を参照してください。

ビット 13 **NEGEDGE** : SDIO_CK 反位相選択ビット

0b : SDIO_CK は、マスタクロック SDIOCLK の立上りエッジで生成されます。

1b : SDIO_CK は、マスタクロック SDIOCLK の立下りエッジで生成されます。

ビット 12:11 **WIDBUS** : ワイドバスモード有効ビット

00 : デフォルトバスモード : SDIO_D0 を使用します。

01 : 4 ワイドバスモード : SDIO_D[3:0] を使用します。

10 : 8 ワイドバスモード : SDIO_D[7:0] を使用します。

ビット 10 **BYPASS** : クロック分周器バイパス有効ビット

0 : バイパス無効 : SDIOCLK は、SDIO_CK 出力信号を駆動する前に、CLKDIV の値に従って分周されます。

1 : バイパス有効 : SDIOCLK は、SDIO_CK 出力信号を直接駆動します。

ビット 9 **PWRSV** : 節電構成ビット

節電のため、バスがアイドル状態のときには、PWRSV をセットして SDIO_CK クロック出力を無効にできます。

0 : SDIO_CK クロックは常に有効です。

1 : SDIO_CK はバスがアクティブなときに限り有効です。

ビット 8 **CLKEN** : クロック有効ビット

0 : SDIO_CK は無効です。

1 : SDIO_CK は有効です。

ビット 7:0 **CLKDIV** : クロック分周ファクタ

このフィールドは、入力クロック (SDIOCLK) と出力クロック (SDIO_CK) の間の分周ファクタを定義します。SDIO_CK 周波数 = SDIOCLK / [CLKDIV + 2]

注 : SD/SDIO カードまたは MultiMediaCard が識別モードのときには、SDIO_CK 周波数は 400 kHz 未満である必要があります。

クロック周波数は、相対カードアドレスがすべてのカードに割り当てられているときには、最大カードバス周波数に変更できます。

データ書き込み後、3 SDIOCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込めません。SDIO_CK は、SD I/O カードの読出しウェイトインターバルの間、停止することができます。この場合、SDIO_CLKCR レジスタは SDIO_CK を制御しません。

21.9.3 SDIO 引数レジスタ (SDIO_ARG)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

SDIO_ARG レジスタは、コマンドメッセージの一部としてカードに送信される 32 ビットのコマンド引数を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDARG																															
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 31:0 **CMDARG** : コマンド引数

コマンドメッセージの一部としてカードに送られるコマンド引数です。コマンドが引数を含む場合、コマンドをコマンドレジスタに書き込む前に、このレジスタにロードする必要があります。

21.9.4 SDIO コマンドレジスタ（SDIO_CMD）

アドレスオフセット：0x0C

リセット値：0x0000 0000

SDIO_CMD レジスタは、コマンドインデックスおよびコマンドタイプビットを含みます。コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。コマンドタイプビットは、コマンドパスステートマシン（CPSM）を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																	CE-ATACMD	nIEN	ENCMDcompl	SDIOSUSPEND	CPSMEN	WAITPEND	WAITINT	WAITRESP	CMDINDEX						
																	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 31:15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **ATACMD** : CE-ATA コマンド
ATACMD がセットされると、CPSM は CMD61 を転送します。
- ビット 13 **nIEN** : 非割込み有効
このビットが 0 の場合、CE-ATA の割込みが有効になります。
- ビット 12 **ENCMDcompl** : CMD 完了有効
このビットがセットされると、コマンド完了信号が有効になります。
- ビット 11 **SDIOSUSPEND** : SD I/O サスペンドコマンド
このビットがセットされると、送信されるコマンドはサスペンドコマンドです（SDIO カードでのみ使用されます）。
- ビット 10 **CPSMEN** : コマンドパスステートマシン（CPSM）有効ビット
このビットがセットされると、CPSM が有効になります。
- ビット 9 **WAITPEND** : CPSM のデータ転送終了待ち（CmdPend 内部信号）。
このビットがセットされると、CPSM は、データ転送の終了を待ってから、コマンドの送信を開始します。
- ビット 8 **WAITINT** : CPSM の割込みリクエスト待ち
このビットがセットされると、CPSM は、コマンドタイムアウトを無効にして、割込みリクエストを待ちます。
- ビット 7:6 **WAITRESP** : レスポンスビット待ち
これらは、CPSM がレスポンスを待つかどうか、および、待つ場合にはレスポンスの種類を設定するために使用されます。
00 : レスポンスなし。CMDSENT フラグを待ちます。
01 : ショートレスポンス。CMDREND または CCRCFAIL フラグを待ちます。
10 : レスポンスなし。CMDSENT フラグを待ちます。
11 : ロングレスポンス。CMDREND または CCRCFAIL フラグを待ちます。
- ビット 5:0 **CMDINDEX** : コマンドインデックス
コマンドインデックスは、コマンドメッセージの一部としてカードに送信されます。

注： データ書き込み後、3 SDIOCLK（48 MHz）クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込めません。

MultiMediaCards はショートレスポンス（48 ビット長）またはロングレスポンス（136 ビット長）の 2 種類のレスポンスを送信できます。SD カードと SD I/O カードはショートレスポンスのみ送信でき、引数はレスポンスのタイプによって異なることがあります。ソフトウェアは、送信されたコマン

ドに応じてレスポンスのタイプを区別します。CE-ATA デバイスは、ショートレスポンスのみを送信します。

21.9.5 SDIO コマンドレスポンスレジスタ (SDIO_RESPCMD)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

SDIO_RESPCMD レジスタは、最後に受信したコマンドレスポンスのコマンドインデックスフィールドを含みます。コマンドレスポンス送信にコマンドインデックスフィールドが含まれていない場合（ロングまたは OCR レスポンス）、RESPCMD フィールドは未知ですが、11111b（レスポンスの予約フィールドの値）が含まれている必要があります。

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

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:0 **RESPCMD** : レスポンスコマンドインデックス

読出し専用ビットフィールドです。最後に受信したコマンドレスポンスのコマンドインデックスを含みます。

21.9.6 SDIO レスポンス 1~4 レジスタ (SDIO_RESPx)

アドレスオフセット : (0x10 + (4 × x)); x = 1..4

リセット値 : 0x0000 0000

SDIO_RESP0/1/2/3/4 レジスタは、受信したレスポンスの一部であるカードのステータスを含みます。

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

ビット 31:0 **CARDSTATUSx** : 表 124 を参照してください。

カードステータスサイズは、レスポンスのタイプによって 32 または 127 ビットです。

表 124. レスポンスタイプと SDIO_RESPx レジスタ

レジスタ	ショートレスポンス	ロングレスポンス
SDIO_RESP1	カードステータス [31:0]	カードステータス [127:96]
SDIO_RESP2	使用されない	カードステータス [95:64]
SDIO_RESP3	使用されない	カードステータス [63:32]
SDIO_RESP4	使用されない	カードステータス [31:1]0b

カードステータスの最上位ビットから先に受信します。SDIO_RESP3 レジスタ LSB は常に 0b です。

21.9.7 SDIO データタイマレジスタ（SDIO_DTIMER）

アドレスオフセット：0x24

リセット値：0x0000 0000

SDIO_DTIMER レジスタは、データタイムアウト期間（カードバスクロック周期単位）を含みます。
カウンタは SDIO_DTIMER レジスタから値をロードして、データパスステートマシン（DPSM）が Wait_R またはビジー状態に入ったときに、デクリメントを開始します。DPSM がこのどちらかの状態のときにタイマが 0 に達した場合、タイムアウトステータスフラグがセットされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATETIME																															
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 31:0 **DATETIME**：データタイムアウト期間
カードバスクロック周期で表されたデータタイムアウト期間です。

注： データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。

21.9.8 SDIO データ長レジスタ（SDIO_DLEN）

アドレスオフセット：0x28

リセット値：0x0000 0000

SDIO_DLEN レジスタは、転送されるデータバイト数を含みます。値は、データ転送が開始されたときに、データカウンタにロードされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み							DATALENGTH																								
							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 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:0 **DATALENGTH**：データ長の値
転送されるデータバイト数です。

注： ブロックデータ転送では、データ長レジスタの値はブロックサイズの倍数である必要があります（SDIO_CTRL を参照してください）。データ転送は、データ制御レジスタに書き込まれる前に、データタイマレジスタとデータ長レジスタに書き込まれなければなりません。
SDIO マルチバイト転送の場合、データ長レジスタ内の値は 1 から 512 まででなければなりません。

21.9.9 SDIO データ制御レジスタ (SDIO_DCTRL)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

SDIO_DCTRL レジスタは、データパスステートマシン (DPSM) を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																				SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCKSIZE				DMAEN	DTMODE	DTDIR	DTEN
																				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 **SDIOEN** : SD I/O 有効機能

このビットがセットされると、DPSM は SD I/O カード固有の操作を実行します。

ビット 10 **RWMOD** : 読出しウェイトモード

0 : SDIO_D2 を停止する読出しウェイト制御です。

1 : SDIO_CK を使用した読出しウェイト制御です。

ビット 9 **RWSTOP** : 読出しウェイト停止

0 : RWSTART ビットがセットされると、読出しウェイトが進行中となります。

1 : RWSTART ビットがセットされると、読出しウェイト停止が有効になります。

ビット 8 **RWSTART** : 読出しウェイト開始

このビットがセットされると、読出しウェイト操作が開始されます。

ビット 7:4 **DBLOCKSIZE** : データブロックサイズ

ブロックデータ転送モードが選択されているときのデータブロック長を定義します。

0000 : (0 10 進) ロック長 = 2^0 = 1 バイト

0001 : (1 10 進) ロック長 = 2^1 = 2 バイト

0010 : (2 10 進) ロック長 = 2^2 = 4 バイト

0011 : (3 10 進) ロック長 = 2^3 = 8 バイト

0100 : (4 10 進) ロック長 = 2^4 = 16 バイト

0101 : (5 10 進) ロック長 = 2^5 = 32 バイト

0110 : (6 10 進) ロック長 = 2^6 = 64 バイト

0111 : (7 10 進) ロック長 = 2^7 = 128 バイト

1000 : (8 10 進) ロック長 = 2^8 = 256 バイト

1001 : (9 10 進) ロック長 = 2^9 = 512 バイト

1010 : (10 10 進) ロック長 = 2^{10} = 1024 バイト

1011 : (11 10 進) ロック長 = 2^{11} = 2048 バイト

1100 : (12 10 進) ロック長 = 2^{12} = 4096 バイト

1101 : (13 10 進) ロック長 = 2^{13} = 8192 バイト

1110 : (14 10 進) ロック長 = 2^{14} = 16384 バイト

1111 : (15 10 進) 予約済み

ビット 3 **DMAEN** : DMA 有効

0 : DMA は無効です。

1 : DMA は有効です。

ビット 2 **DTMODE** : データ転送モード選択 1 : ストリームまたは SDIO マルチバイトデータ転送
0 : ブロックデータ転送です。
1 : ストリームまたは SDIO マルチバイトデータ転送です。

ビット 1 **DTDIR** : データ転送方向選択
0 : コントローラからカードへ
1 : カードからコントローラへ

ビット 0 **DTEN** : データ転送有効ビット
データ転送は、1b が DTEN ビットに書き込まれると開始します。方向ビット DTDIR に応じて、DPSM は Wait_S、Wait_R、または転送の始めにすぐに RW Start がセットされた場合には ReadWait へ移行します。データ転送の終わりに有効ビットをクリアする必要はありませんが、新しいデータ転送を有効にするには、SDIO_DCTRL を更新する必要があります。

注 : データ書き込み後、3 SDIOCLK (48 MHz) クロック周期プラス 2 PCLK2 クロック周期間はデータをこのレジスタに書き込めません。
DTMODE ビットの意味は、SDIOEN ビットの値によって変化します。SDIOEN=0 および DTMODE=1 の場合、MultiMediaCard ストリームモードが有効になり、SDIOEN=1 および DTMODE=1 の場合、ペリフェラルが SDIO マルチバイト転送を有効にします。

21.9.10 SDIO データカウンタレジスタ（SDIO_DCOUNT）

アドレスオフセット : 0x30
リセット値 : 0x0000 0000

SDIO_DCOUNT レジスタは、DPSM がアイドル状態から Wait_R または Wait_S 状態へ移行すると、データ長レジスタから値をロードします（SDIO_DLEN を参照してください）。データが転送されると、カウンタは 0 に達するまで値をデクリメントします。その後、DPSM はアイドル状態へ移行し、データステータス終了フラグ DATAEND がセットされます。

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

ビット 31:25 予約済みであり、リセット値に保持する必要があります。
ビット 24:0 **DATACOUNT** : データカウント値
このビットを読み出すと、残りの転送バイト数が返されます。書き込み操作は何の影響も与えません。

注 : このレジスタは、データ転送が完了したときだけ読み出してください。

21.9.11 SDIO ステータスレジスタ (SDIO_STA)

アドレスオフセット : 0x34

リセット値 : 0x0000 0000

SDIO_STA レジスタは、読み出し専用レジスタです。2 種類のフラグを含みます。

- **スタティックフラグ** (ビット [23:22,10:0]) : これらのビットは、SDIO 割込みクリアレジスタ (SDIO_ICR を参照) に書き込んでクリアされるまで、アサートされたままです。
- **ダイナミックフラグ** (ビット [21:11]) : これらのビットは、基礎回路の状態に応じて状態が変わります (たとえば、FIFO フルおよびエンプティフラグは、データが FIFO に書き込まれるとアサートおよびネゲートされます)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								CEATAEND	SDIOIT	RXDAVL	TXDAVL	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF	RXFIFOHF	TXFIFOHE	RXACT	TXACT	CMDACT	DBCKEND	STBITERR	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
								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 予約済みであり、リセット値に保持する必要があります。

ビット 23 **CEATAEND** : CMD61 に対して CE-ATA コマンド完了信号が受信されました。

ビット 22 **SDIOIT** : SDIO 割込みが受信されました。

ビット 21 **RXDAVL** : 受信 FIFO にデータがあります。

ビット 20 **TXDAVL** : 送信 FIFO にデータがあります。

ビット 19 **RXFIFOE** : 受信 FIFO エンプティ

ビット 18 **TXFIFOE** : 送信 FIFO エンプティ

HW フロー制御が有効なとき、FIFO に 2 ワードが含まれると、TXFIFOE 信号がアクティブになります。

ビット 17 **RXFIFOF** : 受信 FIFO フル

HW フロー制御が有効なとき、FIFO がフルになる 2 ワード前に RXFIFOF 信号がアクティブになります。

ビット 16 **TXFIFOF** : 送信 FIFO フル

ビット 15 **RXFIFOHF** : 受信 FIFO がハーフフル状態です。FIFO には少なくとも 8 ワード入っています。

ビット 14 **TXFIFOHE** : 送信 FIFO がハーフエンプティ状態です。FIFO には少なくとも 8 ワード書き込み可能です。

ビット 13 **RXACT** : データ受信中です。

ビット 12 **TXACT** : データ送信中です。

ビット 11 **CMDACT** : コマンド転送中です。

ビット 10 **DBCKEND** : データブロック送受信 (CRC 確認成功) しました。

ビット 9 **STBITERR** : ワイドバスモードで、すべてのデータ信号ではスタートビットが検出されませんでした。

ビット 8 **DATAEND** : データの終わり (データカウンタ SDIDCOUNT がゼロ) です。

ビット 7 **CMDSENT** : コマンドが送信されました (レスポンス不要)。

ビット 6 **CMDREND** : コマンドレスポンスが受信されました (CRC 確認成功)。

ビット 5 **RXOVERR** : FIFO オーバーランエラーを受信しました。

ビット 4 **TXUNDERR** : 送信 FIFO アンダーランエラーです。

ビット 3 **DTIMEOUT** : データタイムアウトです。

- ビット 2 **CTIMEOUT** : コマンドレスポンスタイムアウトです。
コマンドタイムアウト期間は、固定値の 64 SDIO_CK クロック周期です。
- ビット 1 **DCRCFAIL** : データブロック送受信 (CRC 確認失敗) しました。
- ビット 0 **CCRCFAIL** : コマンドレスポンス受信 (CRC 確認失敗) しました。

21.9.12 SDIO 割込みクリアレジスタ (SDIO_ICR)

アドレスオフセット : 0x38

リセット値 : 0x0000 0000

SDIO_ICR レジスタは、書き込み専用レジスタです。ビットに 1b を書き込むと、SDIO_STA ステータスレジスタの対応するビットがクリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
予約済み								CEATAENDC	SDIOITC	予約済み											DBCKENDC	STBITERRC	DATAENDC	CMDSENTC	CMDREND	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC									
								r/w	r/w												r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	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 **CEATAENDC** : CEATAEND フラグクリアビット
CEATAEND 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 : CEATAEND はクリアされません。
1 : CEATAEND はクリアされます。
- ビット 22 **SDIOITC** : SDIOIT フラグクリアビット
SDIOIT フラグをクリアするために、ソフトウェアによってセットされます。
0 : SDIOIT はクリアされません。
1 : SDIOIT はクリアされます。
- ビット 21:11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **DBCKENDC** : DBCKEND フラグクリアビット
DBCKEND フラグをクリアするために、ソフトウェアによってセットされます。
0 : DBCKEND はクリアされません。
1 : DBCKEND はクリアされます。
- ビット 9 **STBITERRC** : STBITERR フラグクリアビット
STBITERR 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 : STBITERR はクリアされません。
1 : STBITERR はクリアされます。
- ビット 8 **DATAENDC** : DATAEND フラグクリアビット
DATAEND フラグをクリアするために、ソフトウェアによってセットされます。
0 : DATAEND はクリアされません。
1 : DATAEND はクリアされます。
- ビット 7 **CMDSENTC** : CMDSENT フラグクリアビット
CMDSENT フラグをクリアするために、ソフトウェアによってセットされます。
0 : CMDSENT はクリアされません。
1 : CMDSENT はクリアされます。

ビット 6 CMDREND : CMDREND フラグクリアビット

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

0 : CMDREND はクリアされません。

1 : CMDREND はクリアされます。

ビット 5 RXOVERR : RXOVERR フラグクリアビット

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

0 : RXOVERR はクリアされません。

1 : RXOVERR はクリアされます。

ビット 4 TXUNDERR : TXUNDERR フラグクリアビット

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

0 : TXUNDERR はクリアされません。

1 : TXUNDERR はクリアされます。

ビット 3 DTIMEOUT : DTIMEOUT フラグクリアビット

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

0 : DTIMEOUT はクリアされません。

1 : DTIMEOUT はクリアされます。

ビット 2 CTIMEOUT : CTIMEOUT フラグクリアビット

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

0 : CTIMEOUT はクリアされません。

1 : CTIMEOUT はクリアされます。

ビット 1 DCRCFAIL : DCRCFAIL フラグクリアビット

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

0 : DCRCFAIL はクリアされません。

1 : DCRCFAIL はクリアされます。

ビット 0 CCRCFAIL : CCRCFAIL フラグクリアビット

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

0 : CCRCFAIL はクリアされません。

1 : CCRCFAIL はクリアされます。

21.9.13 SDIO マスクレジスタ (SDIO_MASK)

アドレスオフセット : 0x3C

リセット値 : 0x0000 0000

割込みマスクレジスタは、対応するビットを 1b にセットすることによって、割込みリクエストを生成するステータスフラグを決めます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								CEATAENDIE	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIFOEIE	TXFIFOEIE	RXFIFOEIE	TXFIFOEIE	RXFIFOEIE	TXFIFOEIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	STBITERRIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
								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 予約済みであり、リセット値に保持する必要があります。

ビット 23 **CEATAENDIE** : CE-ATA コマンド完了信号受信割込み有効

CE-ATA コマンド完了信号を受信したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : CE-ATA コマンド完了信号受信割込みは無効です。

1 : CE-ATA コマンド完了信号受信割込みは有効です。

ビット 22 **SDIOITIE** : SDIO モード割込み受信割込み有効

SDIO モード割込みを受信したときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : SDIO モード割込み受信割込みは無効です。

1 : SDIO モード割込み受信割込みは有効です。

ビット 21 **RXDAVLIE** : Rx FIFO データ有無割込み有効

Rx FIFO のデータの有無によって生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO のデータの有無による割込みは無効です。

1 : Rx FIFO のデータの有無による割込みは有効です。

ビット 20 **TXDAVLIE** : Tx FIFO データ有無割込み有効

Tx FIFO のデータの有無によって生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO のデータの有無による割込みは無効です。

1 : Tx FIFO のデータの有無による割込みは有効です。

ビット 19 **RXFIFOEIE** : Rx FIFO エンプティ割込み有効

Rx FIFO エンプティによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO エンプティ割込みは無効です。

1 : Rx FIFO エンプティ割込みは有効です。

ビット 18 **TXFIFOEIE** : Tx FIFO エンプティ割込み有効

Tx FIFO エンプティによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO エンプティ割込みは無効です。

1 : Tx FIFO エンプティ割込みは有効です。

ビット 17 RXFIFOFIE : Rx FIFO フル割込み有効

Rx FIFO がフルになったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO フル割込みは無効です。

1 : Rx FIFO フル割込みは有効です。

ビット 16 TXFIFOFIE : Tx FIFO フル割込み有効

Tx FIFO がフルになったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO フル割込みは無効です。

1 : Tx FIFO フル割込みは有効です。

ビット 15 RXFIFOHFIE : Rx FIFO ハーフフル割込み有効

Rx FIFO がハーフフル状態になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO ハーフフル割込みは無効です。

1 : Rx FIFO ハーフフル割込みは有効です。

ビット 14 TXFIFOHEIE : Tx FIFO ハーフエンプティ割込み有効

Tx FIFO がハーフエンプティ状態になったときに生成される割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO ハーフエンプティ割込みは無効です。

1 : Tx FIFO ハーフエンプティ割込みは有効です。

ビット 13 RXACTIE : データ受信割込み有効

データ受信中の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ受信割込みは無効です。

1 : データ受信割込みは有効です。

ビット 12 TXACTIE : データ送信割込み有効

データ送信中の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ送信割込みは無効です。

1 : データ送信割込みは有効です。

ビット 11 CMDACTIE : コマンド動作中割込み有効

コマンド転送中の割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド動作中割込みは無効です。

1 : コマンド動作中割込みは有効です。

ビット 10 DBCKENDIE : データブロックエンド割込み有効

データブロックエンドによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データブロックエンド割込みは無効です。

1 : データブロックエンド割込みは有効です。

ビット 9 STBITERRIE : スタートビットエラー割込み有効

スタートビットエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : スタートビットエラー割込みは無効です。

1 : スタートビットエラー割込みは有効です。

ビット 8 DATAENDIE : データエンド割込み有効

データエンドによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データエンド割込みは無効です。

1 : データエンド割込みは有効です。

ビット 7 CMDSENTIE : コマンド送信割込み有効

コマンド送信による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド送信割込みは無効です。

1 : コマンド送信割込みは有効です。

ビット 6 CMDRENDIE : コマンドレスポンス受信割込み有効

コマンドレスポンスの受信による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンドレスポンス受信割込みは無効です。

1 : コマンドレスポンス受信割込みは有効です。

ビット 5 RXOVERRIE : Rx FIFO オーバーランエラー割込み有効

Rx FIFO オーバーランエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Rx FIFO オーバーランエラー割込みは無効です。

1 : Rx FIFO オーバーランエラー割込みは有効です。

ビット 4 TXUNDERIE : Tx FIFO アンダーランエラー割込み有効

Tx FIFO アンダーランエラーによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Tx FIFO アンダーランエラー割込みは無効です。

1 : Tx FIFO アンダーランエラー割込みは有効です。

ビット 3 DTIMEOUTIE : データタイムアウト割込み有効

データタイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データタイムアウト割込みは無効です。

1 : データタイムアウト割込みは有効です。

ビット 2 CTIMEOUTIE : コマンドタイムアウト割込み有効

コマンドタイムアウトによる割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンドタイムアウト割込みは無効です。

1 : コマンドタイムアウト割込みは有効です。

ビット 1 DCRCFAILIE : データ CRC 失敗割込み有効

データ CRC 失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : データ CRC 失敗割込みは無効です。

1 : データ CRC 失敗割込みは有効です。

ビット 0 CCRCFAILIE : コマンド CRC 失敗割込み有効

コマンド CRC 失敗による割込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : コマンド CRC 失敗割込みは無効です。

1 : コマンド CRC 失敗割込みは有効です。

21.9.14 SDIO FIFO カウンタレジスタ (SDIO_FIFOCNT)

アドレスオフセット : 0x48

リセット値 : 0x0000 0000

SDIO_FIFOCNT レジスタは、FIFO に書き込まれるか、FIFO から読み出される残りのワード数を含みます。FIFO カウンタは、データ制御レジスタ (SDIO_DCTRL レジスタ) のデータ転送有効ビット DTEN がセットされていて、DPSM がアイドル状態のときに、データ長レジスタ (SDIO_DLEN を参照) から値をロードします。データ長がワード境界 (4 の倍数) に合っていない場合、残りの 1~3 バイトは 1 ワードとみなされます。

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

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:0 **FIFOCOUNT** : FIFO に書き込まれるか、FIFO から読み出される残りのワード数

21.9.15 SDIO データ FIFO レジスタ (SDIO_FIFO)

アドレスオフセット : 0x80

リセット値 : 0x0000 0000

受信および送信 FIFO は、32 ビット幅のレジスタとして読み書きできます。FIFO は、32 の連続したアドレスに 32 個のエントリを含みます。このため、CPUはソフトウェアによるFIFOへの読み書きを行うことができます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFOData																															
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 31:0 **FIFOData** : 受信および送信 FIFO データ

FIFO データは、次のアドレスから 32 ビットワードの 32 個のエントリを占有します。
SDIO ベース+ 0x080~SDIO ベース+ 0xFC

21.9.16 SDIO レジスタマップ

次の表は SDIO レジスタのまとめです。

表 125. SDIO 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	SDIO_POWER	予約済み																														PWRCTRL	
0x04	SDIO_CLKCR	予約済み														HWFC_EN	NEGEDGE	WIDBUS	BYPASS	PWRSV	CLKEN	CLKDIV											
0x08	SDIO_ARG	CMDARG																															
0x0C	SDIO_CMD	予約済み														CE-ATACMD	nIEN	ENCMDcompl	SDIOSUSPEND	CPSMEN	WAITPEND	WAITINT	WAITRESP	CMDINDEX									
0x10	SDIO_RESPCMD	予約済み																							RESPCMD								
0x14	SDIO_RESP1	CARDSTATUS1																															
0x18	SDIO_RESP2	CARDSTATUS2																															
0x1C	SDIO_RESP3	CARDSTATUS3																															
0x20	SDIO_RESP4	CARDSTATUS4																															
0x24	SDIO_DTIMER	DATATIME																															
0x28	SDIO_DLEN	予約済み					DATALENGTH																										
0x2C	SDIO_DCTRL	予約済み														SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCKSIZE				DMAEN	DTMODE	DTDIR	DTEN						
0x30	SDIO_DCOUNT	予約済み					DATACOUNT																										
0x34	SDIO_STA	予約済み					CEATAEND	SDIOIT	RXDAVL	TXDAVL	RXFIOE	TXFIOE	RXFIOF	TXFIOF	RXFIOHF	TXFIOHE	RXACT	TXACT	CMDACT	DBCKEND	STBITERR	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL			
0x38	SDIO_ICR	予約済み					CEATAENDC	SDIOITC	予約済み											DBCKENDC	STBITERRC	DATAENDC	CMDSENTC	CMDREND	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC			
0x3C	SDIO_MASK	予約済み					CEATAENDIE	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIOEIE	TXFIOEIE	RXFIOFIE	TXFIOFIE	RXFIOHFIE	TXFIOHEIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	STBITERRIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILE	CCRCFAILE			
0x48	SDIO_FIFOCNT	予約済み					FIFOCOUNT																										
0x80	SDIO_FIFO	FIFOData																															

22 USB On-The-Go フルスピード (OTG_FS)

22.1 OTG_FS の概要

Portions Copyright (c) 2004, 2005 Synopsys, Inc. All rights reserved. 掲載許可取得済み。

このセクションでは、OTG_FS コントローラのアーキテクチャとプログラミングモデルについて説明します。

このセクションでは、次の略記が使用されます。

FS	フルスピード
LS	ロースピード
MAC	メディアアクセスコントローラ
OTG	On-The-Go
PFC	パケット FIFO コントローラ
PHY	物理層
USB	ユニバーサルシリアルバス
UTMI	USB 2.0 トランシーバマクロセルインタフェース (UTMI)

このセクションでは、参照文献として以下の文書が使用されています。

- USB On-The-Go Supplement, Revision 1.3
- Universal Serial Bus Revision 2.0 Specification

OTG_FS はデバイスとホストの両方の機能をサポートするデュアルロールデバイス (DRD) コントローラであり、On-The-Go Supplement to the USB 2.0 Specification に完全に準拠しています。USB OTG は、USB 2.0 Specification に完全に準拠したホスト専用またはデバイス専用のコントローラとして設定することも可能です。ホストモードの場合、OTG_FS はフルスピード (FS、12 Mbits/s) とロースピード (LS、1.5 Mbits/s) の両方の転送をサポートしますが、デバイスモードの場合は、フルスピード転送 (FS、12 Mbits/s) のみをサポートします。OTG_FS は HNP と SRP の両方をサポートします。必要な外部デバイスは、ホストモードで使用する V_{BUS} 用のチャージポンプのみです。

22.2 OTG_FS の主な機能

主な機能は、標準機能、ホストモード機能、およびデバイスモード機能の 3 つのカテゴリに分けて考えることができます。

22.2.1 標準機能

OTG_FS インタフェースの機能概要を次に示します:

- Universal Serial Bus Specification Rev 2.0 に準拠した USB-IF です。
- On-The-Go Supplement Rev 1.3 Specification に詳細が規定されているオプションの On-The-Go (OTG) プロトコルの完全サポート (PHY) を含みます。
 - A/B デバイス識別 (ID ライン) を統合的にサポートしています。
 - ホストネゴシェーションプロトコル (HNP) およびセッションリクエストプロトコル (SRP) を統合的にサポートしています。
 - OTG アプリケーションでバッテリーを節約するために、ホストで V_{BUS} をオフすることができます。
 - 内部コンパレータによって OTG に V_{BUS} のレベルを監視させることができます。
 - ホスト-ペリフェラルのダイナミックな役割切り替えをサポートします。
- 次の動作を行うようにソフトウェアでの設定が可能です。
 - SRP 対応の USB FS ペリフェラル (B デバイス)
 - SRP 対応の USB FS/LS ホスト (A デバイス)
 - USB On-The-Go フルスピードデュアルロールデバイス
- 以下の要素により FS SOF および LS Keep-alive をサポートします。
 - SOF パルス PAD 接続性
 - タイマ 2 (TIM2) との SOF パルス内部接続
 - 設定可能なフレーム周期
 - 設定可能なフレームの終わりの割込み
- USB サスペンド中のシステム停止、デジタルコア、PHY、および DFIFO 電源管理回路内部のクロックドメインの電源オフなど、節電機能を内蔵しています。
- 以下のような高度な FIFO 制御を備えた専用の 1.25 KB RAM を備えています。
 - RAM 空間を複数の FIFO に分割して、RAM を柔軟かつ効率的に使用できるように設定できます。
 - 各 FIFO は複数のパケットを保持できます。
 - 動的なメモリ割り当てが可能です。
 - FIFO を 2 のべき乗でないサイズに設定して、連続的なメモリ位置の使用が可能です。
- システムの介入なしで、最大 1 フレーム (1ms) の間、最大 USB バンド幅を保証します。

22.2.2 ホストモード機能

ホストモードでの OTG_FS インタフェースの主な機能と条件は次の通りです:

- V_{BUS} 電圧発生用の外部チャージポンプを持っています。
- 最大 8 個のホストチャネル (パイプ) を持っています。各チャネルはどんなタイプの USB 転送も割り当てられるように動的に設定可能です。
- 次を保持するハードウェアスケジューラを内蔵しています。
 - 最大 8 個のインタラプトに加えて、周期的ハードウェアキュー内でのアイソクロナス転送リクエスト
 - 最大 8 つのコントロールに加えて、非周期的ハードウェアキュー内でのバルク転送リクエスト
- USB データ RAM の効率的な使用のために、共有 RX FIFO、周期的 TX FIFO、および非周期的 TX FIFO を管理します。

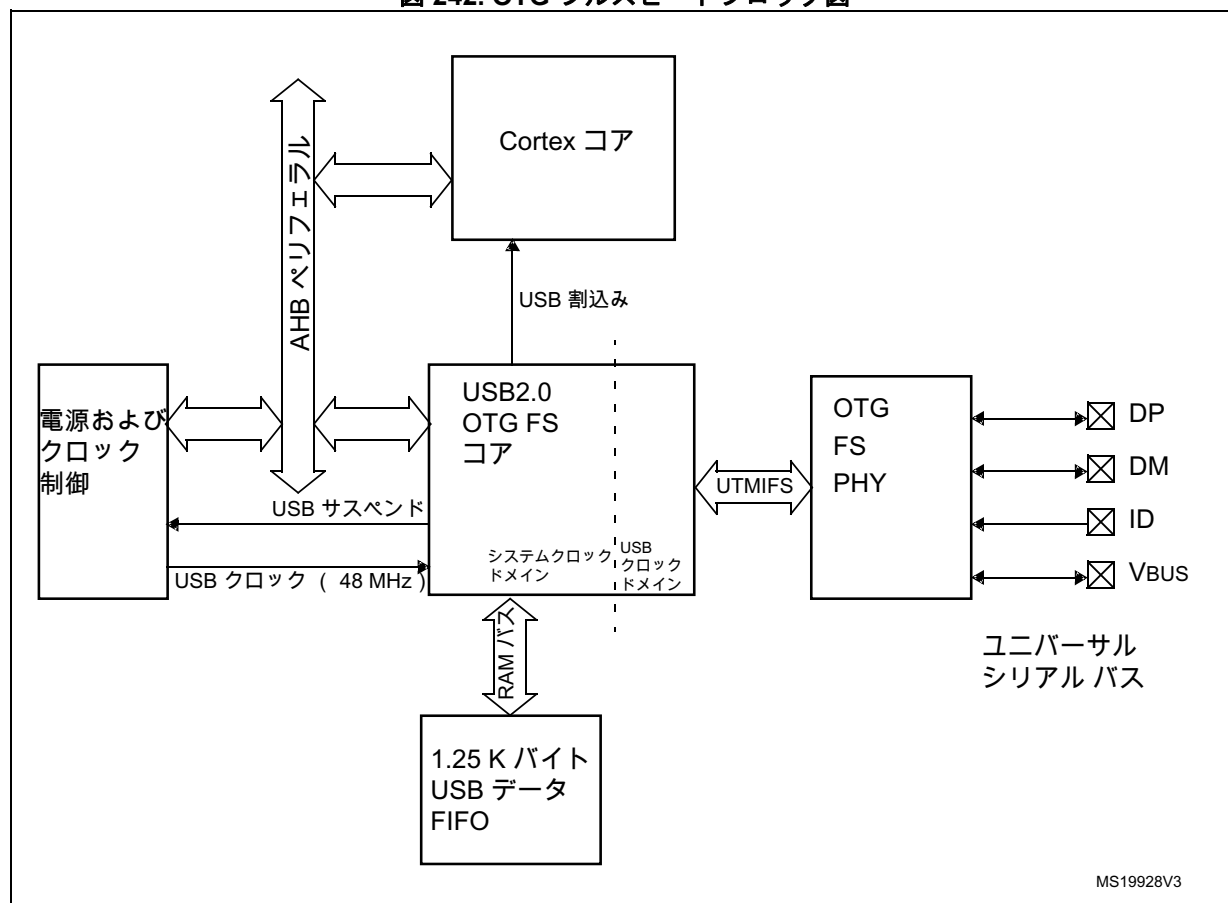
22.2.3 ペリフェラルモード機能

ペリフェラルモードでのOTG_FSインタフェースの主要機能を次に示します:

- 1つの双方向コントロールエンドポイント0を持っています。
- バルク、インタラプト、またはアイソクロナス転送をサポートするために、設定可能な3つのINエンドポイント(EP)を持っています。
- バルク、インタラプト、またはアイソクロナス転送をサポートするために、設定可能な3つのOUTエンドポイント(EP)を持っています。
- USBデータRAMの効率的な使用のために、共有RXFIFOとTx-OUTFIFOを管理します。
- アプリケーションの負荷を軽減するために、最大4つの専用Tx-INFIFO(有効なINEPに対して1つずつ)を管理します。
- ソフト切断機能をサポートしています。

22.3 OTG_FSの機能説明

図 242. OTG フルスピードブロック図



22.3.1 OTG フルスピードコア

USB OTG FS は、外部水晶発振器を通じて、リセットおよびクロックコントローラ (RCC) から $48\text{MHz} \pm 0.25\%$ のクロックを受け取ります。USB クロックは、フルスピード (12 M ビット/秒) で 48 MHz ドメインを駆動するために使用され、OTG FS コアを設定する前に有効にされなければなりません。

CPU は、AHB ペリフェラルバスを使用して OTG FS コアレジスタに対する読み書きを実行します。USB イベントは 1 本の USB OTG 割込みラインによって通知されます ([セクション 22.15: OTG_FS 割込み](#)を参照)。

CPU は、専用の OTG_FS 位置 (プッシュレジスタ) に 32 ビットのワードを書き込むことによって、USB を介してデータを提供します。データは USB データ RAM 内に設定された Tx データ FIFO に自動的に格納されます。各 IN エンドポイント (ペリフェラルモードの場合) または OUT チャネル (ホストモードの場合) に対して 1 つの Tx FIFO プッシュレジスタがあります。

CPU は、専用の OTG_FS アドレス (ポップレジスタ) から 32 ビットのワードを読み出すことによって、USB からデータを受け取ります。データは、1.25 KB の USB データ RAM 内に設定された共有 Rx FIFO から自動的に取得されます。各 OUT エンドポイントまたは IN チャネルに対して 1 つの Rx FIFO ポップレジスタがあります。

USB プロトコル層は、シリアルインタフェースエンジン (SIE) によって駆動され、オンチップ物理層 (PHY) 内のフルスピード/ロースピードトランシーバモジュールによって USB 上で直列化 (シリアルライズ) されます。

22.3.2 フルスピード OTG PHY

組み込みのフルスピード OTG PHY は、OTG フルスピード コアによって制御され、UTMI+ バス (UTMIFS) のフルスピードサブセットを通じて USB 制御信号およびデータ信号を運びます。USB 接続の物理サポートを提供します。

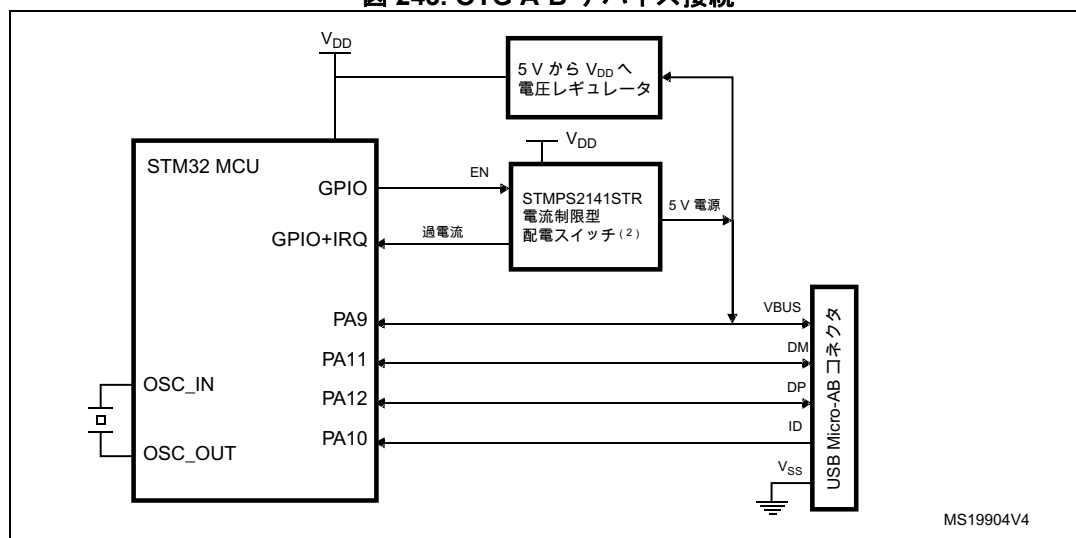
フルスピード OTG PHY には、以下のコンポーネントが含まれます。

- ホストとデバイスの両方で使用される FS/LS トランシーバモジュール。シングルエンド USB ライン上で送受信を直接駆動します。
- A/B デバイス識別のための ID ラインをサンプリングするために使用される組み込みの ID プルアップ抵抗。
- デバイスの現在の役割に依存して OTG_FS コアによって制御される 組み込みの DP/DM プルアップおよびプルダウン抵抗。ペリフェラルとして、DP プルアップ抵抗を有効にし、フルスピードペリフェラル接続を知らせます。信号を送るタイミングは、 V_{BUS} が有効なレベルであると判断されてすぐです (B セッションが有効な場合)。ホストモードでは DP と DM の両方でプルダウン抵抗が有効にされます。プルアップ抵抗とプルダウン抵抗は、デバイスの役割がホストネゴシエーションプロトコル (HNP) を通じて変更されたとき、動的に切り替えられます。
- プルアップ/プルダウン抵抗 ECN 回路。DP プルアップ回路は、USB Rev2.0 に適用される抵抗の技術変更通知 (Engineering Change Notice) に従って、OTG_FS から個別に制御される 2 つの抵抗で構成されます。DP プルアップの動的なトリミングは、ノイズの低減と Tx/Rx 信号品質の向上を可能にします。
- V_{BUS} レベル検出用コンパレータ。 V_{BUS} の有効性、A-B セッションの有効性、およびセッション終了電圧閾値を検出するために使用されるヒステリシスを有します。これらは、セッションリクエストプロトコル (SRP) の駆動、有効な起動およびセッション終了条件の検出、および USB 操作中の V_{BUS} 電源の定常的な監視のために使用されます。
- V_{BUS} パルス駆動回路。SRP 中に抵抗により V_{BUS} を充電/放電するために使用します (弱駆動)。

注意 : USB OTG FSペリフェラルの正常動作を保証するためには、AHB 周波数を14.2 MHzより高くする必要があります。

22.4 OTG デュアルロールデバイス (DRD)

図 243. OTG A-B デバイス接続



1. V_{BUS} で電源が供給されるデバイスを設定するときだけ必要な外部電圧レギュレータ
2. アプリケーションが V_{BUS} で電源が供給されるデバイスをサポートしなければならない場合のみ STMPS2141STR が必要です。アプリケーションボード上で 5 V が使用可能な場合、基本的なパワースイッチを使用することができます。

22.4.1 ID ラインの検出

ホストかペリフェラル（デフォルト）かの役割は、ID 入力ピンに応じて割り当てられます。ID ラインのステータスは、USB 端子への挿入時に、USB ケーブルのどちら側がマイクロ AB レセプタクルに接続されたかによって決まります。

- USB ケーブルの B 側がフローティング ID ワイヤに接続された場合、組み込みのプルアップ抵抗が ID ラインのハイレベルを検出して、デフォルトのペリフェラルの役割が確立されます。この設定では、OTG_FS はセクション 6.8.2 で説明する標準 FSM に準拠しています（On-The-Go Specification Rev1.3 supplement to the USB2.0 の On-The-Go B デバイス）。
- USB ケーブルの A 側がアース付き ID ラインに接続された場合、OTG_FS は、ホストのソフトウェア初期化のために ID ラインステータス変化割込み（OTG_FS_GINTSTS の CIDSCHG ビット）を発行して、自動的にホストの役割に切り替わります。この設定では、OTG_FS はセクション 6.8.1 で説明する標準 FSM に準拠しています（On-The-Go Specification Rev1.3 supplement to the USB2.0 の On-The-Go A デバイス）。

22.4.2 HNP デュアルロールデバイス

グローバル USB 設定レジスタの HNP 対応ビット（OTG_FS_GUSBCFG の HNPCAP ビット）は、OTG_FS コアがホストネゴシエーションプロトコル（HNP）に従って、役割を A ホストから A ペリフェラルに（またその逆）、または B ペリフェラルから B ホスト（またその逆）に動的に変更できるようにします。現在のデバイスのステータスは、グローバル OTG 制御およびステータスレジスタのコネクタ ID ステータスビット（OTG_FS_GOTGCTL の CIDSTS ビット）とグローバル割込みおよびステータスレジスタの現在の動作モードビット（OTG_FS_GINTSTS の CMOD ビット）の値の組み合わせによって読み出すことができます。

HNP プログラミングモデルの詳細は、[セクション 22.17: OTG_FS プログラミングモデル](#)に記載されています。

22.4.3 SRP デュアルロールデバイス

グローバル USB 設定レジスタの SRP 対応ビット (OTG_FS_GUSBCFG の SRPCAP ビット) は、OTG_FS コアが A デバイスの節電のために V_{BUS} の生成を停止できるようにします。A デバイスは OTG_FS の役割がホストかペリフェラルかに関係なく、常に、 V_{BUS} を駆動する役割を持っていることに注意してください。

SRP A/B デバイスのプログラミングモデルの詳細は、[セクション 22.17: OTG_FS プログラミングモデル](#)に記載されています。

22.5 USB ペリフェラル

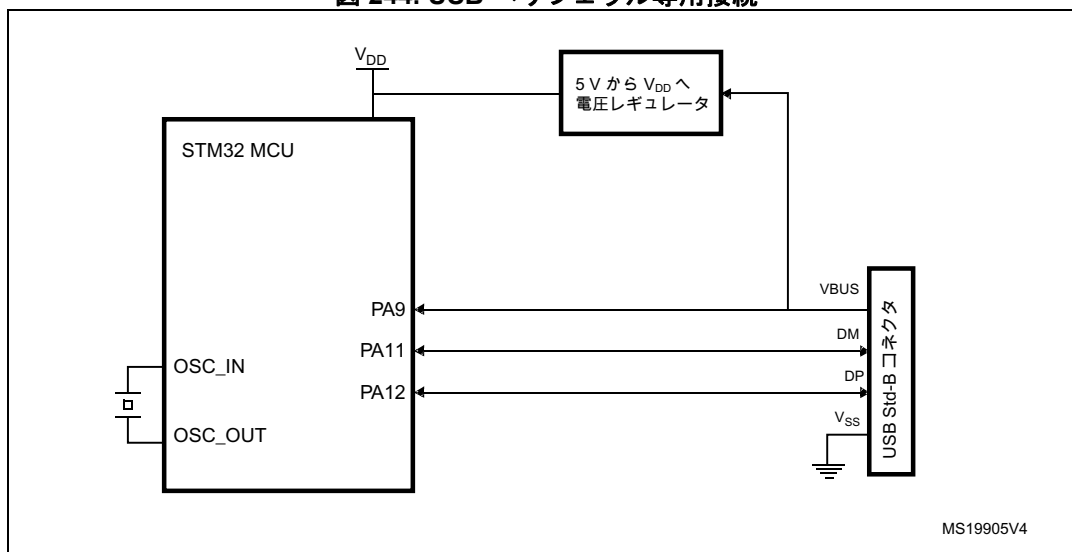
このセクションでは、USB ペリフェラルモードでの OTG_FS の機能について説明します。OTG_FS は、以下の環境で USB ペリフェラルとして機能します。

- OTG B ペリフェラル
 - USB ケーブルの B 側が挿入された場合の OTG B デバイスのデフォルトの状態
- OTG A ペリフェラル
 - HNP が OTG_FS の役割をペリフェラルに切り替えた後の OTG A デバイスの状態
- B デバイス
 - ID ラインが存在し、機能していて、USB ケーブルの B 側に接続され、かつ、グローバル USB 設定レジスタの HNP 対応ビット (OTG_FS_GUSBCFG の HNPCAP ビット) がクリアされている場合 (On-The-Go Rev1.3 の 6.8.3 項を参照)。
- ペリフェラルのみ ([図 244 : USB ペリフェラル専用接続](#)を参照)
 - グローバル USB 設定レジスタの強制デバイスモードビット (OTG_FS_GUSBCFG の FDMOD) が 1 にセットされると、OTG_FS コアは強制的に USB ペリフェラル専用として機能します (On-The-Go Rev1.3 の 6.8.3 項を参照)。この場合、ID ラインは、USB コネクタ上に存在していても無視されます。

注： B デバイスまたはペリフェラル専用設定の場合に電源がバスから供給されるデバイスを実装するには、 V_{DD} チップ供給 (V_{BUS} からの) を生成する外部レギュレータを追加する必要があります。

V_{BUS} センシングオプションを無効化することにより、 V_{BUS} ピンを解放することができます。このためには、OTG_FS_GCCFG レジスタの NOVBUSSSENS ビットをセットします。この場合、 V_{BUS} は内部で常に V_{BUS} の有効レベル (5 V) にあると見なされます。

図 244. USB ペリフェラル専用接続



1. バスから電源が供給されるデバイスを構築するには、レギュレータを使用します。

22.5.1 SRP 対応ペリフェラル

グローバル USB 設定レジスタの SRP 対応ビット (OTG_FS_GUSBCFG の SRPCAP ビット) は、OTG_FS コアがセッションリクエストプロトコル (SRP) をサポートできるようにします。この方法により、USB セッションがサスペンド状態にある間、 V_{BUS} の供給を停止して、リモート A デバイスの消費電力を節約できます。

SRP ペリフェラルモードのプログラミングモデルの詳細は、[B デバイスセッションリクエストプロトコル](#) のセクションに記載されています。

22.5.2 ペリフェラルの状態

電源が供給された状態

V_{BUS} 入力は、USB ペリフェラルが電源供給状態に入ることが許される B セッション有効電圧を検出します (USB 2.0 の 9.1 項を参照)。その場合、OTG_フルスピード は、DP プルアップ抵抗を自動的に接続して、フルスピードデバイス接続をホストに知らせ、セッションリクエスト割込みを生成して (OTG_FS_GINTSTS の SRQINT ビット)、電源供給状態を通知します。

V_{BUS} 入力は、また、USB 動作中にホストによって適切な V_{BUS} レベルが供給されることを保証します。 V_{BUS} 電源が B セッションの有効値以下に低下したことが検出された場合 (たとえば、電源の変動があった場合や、ホストポートの電源がオフとなった場合など)、OTG_FS は自動的に切断され、セッション終了検出割込み (OTG_FS_GOTGINT の SEDET ビット) が生成されて、OTG_FS が電源供給状態でなくなったことを通知します。

電源供給状態では、OTG_FS は、ホストからリセットシグナリングを受け取ることを期待しています。他の USB 操作を実行することはできません。リセットシグナリングを受け取ると、リセット検出割込み (OTG_FS_GINTSTS の USBRST ビット) が生成されます。リセットシグナリングが完了すると、エミュレーション終了割込み (OTG_FS_GINTSTS の ENUMDNE ビット) が生成され、OTG_FS はデフォルト状態に入ります。

ソフト切断

電源供給状態は、ソフト切断機能によってソフトウェアで終了させることができます。DP プルアップ抵抗は、デバイス制御レジスタのソフト切断ビット (OTG_FS_DCTL の SDIS ビット) をセットすることによって取り除かれ、USB ケーブルが実際にはホストポートから取り除かれていなくても、ホスト側にデバイス切断検出割込みを生成します。

デフォルト状態

デフォルト状態では、OTG_FS はホストから SET_ADDRESS コマンドを受け取ることを期待しています。他の USB 操作を実行することはできません。有効な SET_ADDRESS コマンドが USB 上でデコードされると、アプリケーションは対応する番号をデバイス設定レジスタのデバイスアドレスフィールドに書き込みます (OTG_FS_DCFG の DAD ビット)。その場合、OTG_FS アドレス状態に入り、設定された USB アドレスでホストトランザクションに応答する準備が整います。

サスペンド状態

OTG_FS ペリフェラルは、USB の活動状態を定期的に監視します。USB のアイドル状態が 3 ms 続くと、アーリーサスペンド割込み (OTG_FS_GINTSTS の ESUSP ビット) が発行され、適切な場合は 3 ms 後にサスペンド割込み (OTG_FS_GINTSTS の USBSUSP ビット) によって確認されます。その場合、デバイスステータスレジスタのデバイスサスペンドビット (OTG_FS_DSTS の SUSPSTS ビット) が自動的にセットされ、OTG_FS はサスペンド状態に入ります。

サスペンド状態は、デバイス自身によって終了することもできます。この場合、アプリケーションはデバイス制御レジスタのリモートウェイクアップシグナリングビット (OTG_FS_DCTL の RWUSIG ビット) をセットし、1~15 ms 後にクリアします。

ホストからレジュームシグナリングが検出されると、レジューム割込み (OTG_FS_GINTSTS の WKUPINT ビット) が発生され、デバイスサスペンドビットは自動的にクリアされます。

22.5.3 ペリフェラルエンドポイント

OTG_FS コアは、以下の USB エンドポイントをインスタンス化します。

- コントロールエンドポイント 0 :
 - 双方向であり、制御メッセージのみを取り扱います。
 - 入力および出力トランザクションを処理する個別のレジスタセットを備えています。
 - 固有の制御レジスタ (OTG_FS_DIEPCTL0/OTG_FS_DOEPCTL0)、転送設定レジスタ (OTG_FS_DIEPTSIZ0/OTG_FS_DOEPSIZ0)、およびステータス割込みレジスタ (OTG_FS_DIEPINTx/OTG_FS_DOEPINT0) を備えています。制御および転送サイズレジスタ内で使用可能なビットのセットは、他のエンドポイントとは少し違います。
- 3 個の IN エンドポイント
 - それぞれ、アイソクロナス転送、バルク転送、またはインタラプト転送タイプをサポートするように設定できます。
 - また、それぞれに固有の制御レジスタ (OTG_FS_DIEPCTLx)、転送設定レジスタ (OTG_FS_DIEPTSIZx)、およびステータス割込みレジスタ (OTG_FS_DIEPINTx) を備えています。
 - デバイス IN エンドポイント共通割込みマスクレジスタ (OTG_FS_DIEPMSK) を使用して、すべての IN エンドポイント (EP0 も含む) 上で 1 種類のエンドポイント割込みソースを有効/無効にできます。
 - 現在のフレーム内で転送が完了していないアイソクロナス IN エンドポイントが少なくとも 1 つあるときにアサートされる、不完全アイソクロナス IN 転送割込み (OTG_FS_GINTSTS の IISOIXFR ビット) をサポートします。この割込みは、周期的フレーム割込み (OTG_FS_GINTSTS/EOPF) の終わりとともにアサートされます。

- 3つのOUTエンドポイント
 - それぞれ、アイソクロナス転送、バルク転送、またはインタラプト転送タイプをサポートするように設定できます。
 - また、それぞれに固有の制御レジスタ (OTG_FS_DOEPCTLx)、転送設定レジスタ (OTG_FS_DOEPSIZx)、およびステータス割込みレジスタ (OTG_FS_DOEPINTx) を備えています。
 - デバイスOUTエンドポイント共通割込みマスクレジスタ (OTG_FS_DOEPMASK) を使用して、すべてのOUTエンドポイント (EP0も含む) 上で1種類のエンドポイント割込みソースを有効/無効にすることができます。
 - 現在のフレーム内で転送が完了していないアイソクロナスOUTエンドポイントが少なくとも1つあるときにアサートされる、不完全アイソクロナスOUT転送割込み (OTG_FS_GINTSTSのINCOMPIISOOUTビット) をサポートします。この割込みは、周期的フレーム割込み (OTG_FS_GINTSTS/EOPF) の終わりとともにアサートされます。

エンドポイント制御

- 以下のエンドポイント制御は、デバイスエンドポイント-x IN/OUT 制御レジスタ (DIEPCTLx/DOEPCTLx) を通じて、アプリケーションから使用できます。
 - エンドポイント有効/無効
 - 現在の設定でのエンドポイントのアクティブ化
 - USB転送タイプ (アイソクロナス、バルク、インタラプト) のプログラミング
 - サポートされるパケットサイズのプログラミング
 - INエンドポイントに関連するTx-FIFO番号のプログラミング
 - 期待される、または送信されるdata0/data1PID (バルク/インタラプトのみ) のプログラミング
 - トランザクションが送受信される偶数/奇数フレームのプログラミング (アイソクロナス転送のみ)
 - オプションで、FIFOのステータスに関係なく、常にホストに否定応答するNAKビットのプログラミング
 - オプションで、エンドポイントに対するホストトークンを常に停止するSTALLビットのプログラミング
 - オプションで、OUTエンドポイントが受信データのCRCフィールドをチェックしないようにするSNOOPモードのプログラミング

エンドポイント転送

アプリケーションは、デバイスエンドポイント-x転送サイズレジスタ (DIEPTSIZx/DOEPSIZx) を使用して、転送サイズパラメータをプログラムし、転送ステータスを読み出すことができます。エンドポイント制御レジスタのエンドポイントイネーブルビットをセットする前にプログラミングを行う必要があります。エンドポイントが有効になると、これらのフィールドは、OTG_FSコアが現在の転送ステータスで更新すると、読み出し専用になります。

以下の転送パラメータがプログラム可能です。

- 転送サイズ (バイト単位)
- 全体の転送サイズを構成するパケットの数

エンドポイントのステータス／割込み

デバイスエンドポイント-x 割込みレジスタ (DIEPINTx/DOEPINTx) は、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。アプリケーションは、コア割込みレジスタの OUT エンドポイント割込みビットまたは IN エンドポイント割込みビット (OTG_FS_GINTSTS の OEPINT ビット、または OTG_FS_GINTSTS の IEPINT ビット) がセットされたとき、これらのレジスタを読み出さなければなりません。アプリケーションがこれらのレジスタを読み出すためには、まず、デバイスオールエンドポイント割込み (OTG_FS_DAIN) レジスタを読み出して、デバイスエンドポイント-x 割込みレジスタの正確なエンドポイント番号を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、DAINT および GINTSTS レジスタの対応するビットをクリアする必要があります。

ペリフェラルコアは、以下のステータスチェックおよび割込み生成を行います。

- データ転送がアプリケーション側 (AHB) と USB 側の両方で完了したことを示す転送完了割込み
- セットアップステージが終了しているかどうか (コントロールOUT のみ)
- 関連する送信 FIFO が半エンプティか完全エンプティか (IN エンドポイント)
- NAK 応答がホストに送信されたかどうか (アイソクロナス IN のみ)
- Tx-FIFO がエンプティのとき IN トークンが受け取られたかどうか (バルク IN/インタラプト IN のみ)
- エンドポイントがまだ有効になっていないときに OUT トークンを受け取ったかどうか
- バブルエラー条件が検出されたかどうか
- アプリケーションによるエンドポイントの無効化が可能かどうか
- アプリケーションによるエンドポイント NAK が可能かどうか (アイソクロナス IN のみ)
- 3つを超える連続したセットアップパケットが受信されたかどうか (コントロール OUT のみ)
- タイムアウト条件が検出されたかどうか (コントロール IN のみ)
- アイソクロナス OUT パケットが割込みを生成せずにドロップされたかどうか

22.6 USB ホスト

このセクションでは、USB ホストモードでの OTG_FS の機能について説明します。OTG_FS は以下の環境で USB ホストとして機能します。

- OTG A ホスト
 - USB ケーブルの A 側が挿入されたときの OTG A デバイスのデフォルト状態
- OTG B ホスト
 - HNP がホストの役割に切り替わった後の OTG B デバイス
- A デバイス
 - ID ラインが存在し、機能していて、USB ケーブルの A 側に接続され、かつ、グローバル USB 設定レジスタの HNP 対応ビット (OTG_FS_GUSBCFG の HNPCAP ビット) がクリアされている場合。組み込みのプルダウン抵抗が自動的に DP/DM ライン上にセットされます。
- ホストのみ (図 245 : USB ホスト専用接続を参照)。
 - グローバル USB 設定レジスタの強制ホストモードビット (OTG_FS_GUSBCFG の FHMOD ビット) は、OTG_FS コアを強制的に USB ホスト専用として機能させます。この場合、ID ラインは、USB コネクタ上に存在していても無視されます。組み込みのプルダウン抵抗が自動的に DP/DM ライン上にセットされます。

V_{BUS} 入力は、USB 操作中に、チャージポンプによって有効な V_{BUS} レベルが供給されることを保証します。チャージポンプの過電流出力は、ポート割込みを生成するように設定された任意の GPIO ピンに入力できます。過電流 ISR は速やかに V_{BUS} の生成を無効にする必要があります。

The diagram illustrates the electrical connections for the STM32 MCU system. The STM32 MCU is connected to the STMP2141STR current-limited power switch (EN, EN, 過電流) and the USB Std-A connector (P 9, PA11, PA12, DM, DP, VSS). The power supply is provided by a 5V source (5V 電源) connected to the VDD pin of the STMP2141STR and the VBUS pin of the USB connector. The ground connection is labeled VSS.

1. アプリケーションが V_{BUS} で電源が供給されるデバイスをサポートしなければならない場合のみSTMPS2141STRが必要です。アプリケーションボード上で 5 V が使用可能な場合、基本的なパワースイッチを使用することができます。
2. V_{DD} 範囲は 2.0~3.6 V です。

SRP のサポートはグローバル USB 設定レジスタの SRP 対応ビット (OTG_FS_GUSBCFG の SRPCAP ビット) を通じて使用可能です。SRP 機能が有効なときには、ホストは、USB セッションがサスペンド状態にある間、V_{BUS} の電力供給を停止することによって、消費電力を節約できます。

HNP または SRP が有効化された場合、VBUS センシングピン（PA9）は V_{BUS} に接続する必要があります。V_{BUS} 入力は、USB 動作中にチャージポンプによって適切な V_{BUS} レベルが供給されることを保証します。予期せぬ V_{BUS} 電圧の低下（V_{BUS} の有効閾値 4.25 V 以下へ）が発生すると、セッション終了検出ビット（OTG FS GOTGINT の SEDET ビット）によって OTG 割込みがトリガされ

ます。その場合、アプリケーションは V_{BUS} 電源供給を停止し、ポートパワービットをクリアする必要があります。

HNP と SRP が共に無効化された場合 V_{BUS} センシングピン (PA9) は V_{BUS} に接続してはいけません。このピンは GPIO として使用することができます。

チャージポンプの過電流フラグは、電氣的ダメージを防止するためにも使用できます。チャージポンプからの過電流フラグ出力を任意の GPIO 入力に接続して、有効レベルになったときにポート割込みを生成するように設定します。過電流 ISR は迅速に V_{BUS} の発生を無効にし、ポートパワービットをクリアする必要があります。

ペリフェラル接続のホスト検出

SRP または HNP を有効化し、かつ USB ペリフェラルまたは B デバイスをいつでも接続できる場合でも、OTG_FS はバス接続を検出しません。 V_{BUS} が有効なレベルにあり、リモート B デバイスが接続されると、OTG_FS コアは、ホストポート制御およびステータスレジスタのデバイス接続ビット (OTG_FS_HPRT の PCDET ビット) によってトリガされるホストポート割込みを発行します。

HNP と SRP が共に切り離されたとき、USB ペリフェラルまたは B デバイスが接続されると直ちに検出されます。OTG_FS コアは、ホストポート制御およびステータス内のデバイス接続ビット (OTG_FS_HPRT の PCDET ビット) によってトリガされるホストポート割込みを発行します。

ペリフェラル切断のホスト検出

ペリフェラル切断イベントは、切断検出割込み (OTG_FS_GINTSTS の DISCINT ビット) をトリガします。

ホストのエニュメレーション

ペリフェラル接続を検出した後、ホストは、USB リセットおよび設定コマンドを新しいペリフェラルに送信することによって、エニュメレーションプロセスを開始しなければなりません。

USB リセットの駆動を開始する前に、アプリケーションは、デバウンス終了ビット (OTG_FS_GOTGINT の DBCDNE ビット) によってトリガされる OTG 割込みを待ちます。これは、DP (FS) または DM (LS) 上のプルアップ抵抗の接続によって生じた電氣的デバウンスの後、バスが再び安定状態に戻ったことを示します。

アプリケーションは、ホストポート制御およびステータスレジスタのポートリセットビット (OTG_FS_HPRT の PRST ビット) を最短 10 ms、最長 20 ms の間セット状態に保つことによって、USB を介した USB リセットシグナリング (シングルエンド、ゼロ) を駆動します。アプリケーションは、タイミングカウントをチェックした後、ポートリセットビットがクリアされているかどうかを確認します。

USB リセットシーケンスが完了すると、ポートイネーブル/ディセーブル変更ビット (OTG_FS_HPRT の PENCHNG ビット) によってホストポート割込みがトリガされます。これは、エニュメレーションされたペリフェラルの速度をホストポート制御およびステータスレジスタ (OTG_FS_HPRT の PSPD ビット) のポート速度フィールドから読み出せることと、ホストが SOF (FS) またはキープアライブ (LS) の駆動を開始しようとしていることをアプリケーションに知らせます。これで、ホストは、ペリフェラル設定コマンドを送信することによってペリフェラルエニュメレーションを完了する準備ができたことになります。

ホストのサスペンド

アプリケーションは、ホストポート制御およびステータスレジスタのポートサスペンドビット (OTG_FS_HPRT の OSUSP ビット) をセットすることによって、USB の活動をサスペンド (一時停止) します。OTG_FS コアは、SOF の送信を停止し、サスペンド状態に入ります。

サスペンド状態は、リモートデバイスが主導して（リモートウェイクアップ）、終了することもできます。この場合、ホストポート制御およびステータスレジスタのポートレジャームビット (OTG_FS_HPRT の PRES ビット) セルフセットが検出されると、リモートウェイクアップ割込み (OTG_FS_GINTSTS の WKUPINT ビット) が生成され、レジャーム信号が USB 上で自動的に駆動されます。アプリケーションは、レジャームウィンドウの時間を決めてから、ポートレジャームビットをクリアしてサスペンド状態を終了し、SOF をリスタートする必要があります。

サスペンド状態をホスト主導で終了した場合、アプリケーションは、ポートレジャームビットをセットして、ホストポートでのシグナリングのレジャームを開始し、レジャームウィンドウの時間を決め、最後に、ポートレジャームビットをクリアする必要があります。

22.6.3 ホストチャネル

OTG_FS コアは、8 つのホストチャネルをインスタンス化します。各ホストチャネルは USB ホスト転送をサポートします (USB パイプ)。ホストは、同時に 8 つを超える転送リクエストをサポートすることはできません。アプリケーションからの 8 つを超える転送リクエストがベンディング中の場合、ホストコントローラドライバ (HCD) は、チャネルが以前のデューティから使用可能になったとき、すなわち、転送完了およびチャネルホールド割込みの受信後、チャネルの再割当てを行う必要があります。

各ホストチャネルは、入力／出力および任意のタイプの周期的/非周期的トランザクションをサポートするように設定できます。各ホストチャネルは、適切な制御レジスタ (HCCHARx)、転送設定レジスタ (HCTSIZx)、およびステータス／割込みレジスタ (HCINTx) と関連するマスクレジスタ (HCINTMSKx) を使用します。

ホストチャネル制御

- アプリケーションは、ホストチャネル x 特性レジスタ (HCCHARx) を通じて、以下のホストチャネル制御を使用できます。
 - チャネルの有効化／無効化
 - ターゲット USB ペリフェラルの FS/LS 速度のプログラミング
 - ターゲット USB ペリフェラルのアドレスのプログラミング
 - ターゲット USB ペリフェラルのエンドポイント数のプログラミング
 - IN/OUT 転送方向のプログラミング
 - USB 転送タイプ (コントロール、バルク、インタラプト、アイソクロナス) のプログラミング
 - 最大パケットサイズ (MPS) のプログラミング
 - 奇数／偶数フレームで実行される周期的転送のプログラミング

ホストチャネル転送

アプリケーションは、ホストチャネル転送サイズレジスタ (HCTSIZx) を使用して、転送サイズパラメータをプログラムし、転送ステータスを読み出すことができます。プログラミングは、ホストチャネル特性レジスタのチャネルイネーブルビットをセットする前に行う必要があります。エンドポイントが有効になると、OTG_FS コアが現在の転送ステータスに従って更新するので、パケットカウントフィールドは読み出し専用になります。

- 以下の転送パラメータがプログラム可能です。
 - 転送サイズ (バイト単位)
 - 全体の転送サイズを構成するパケットの数
 - 初期データ PID

ホストチャネルのステータス／割込み

ホストチャネル x 割込みレジスタ (HCINTx) は、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。アプリケーションは、コア割込みレジスタのホストチャネル割込みビット (OTG_FS_GINTSTS の OEPINT ビット) がセットされたとき、これらのレジスタを読み出さなければなりません。アプリケーションがこれらのレジスタを読み出すためには、まず、ホストオールチャネル割込み (HCAINT) レジスタを読み出して、ホストチャネル x 割込みレジスタの正確なチャネル数を取得する必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、HAINT および GINTSTS レジスタの対応するビットをクリアする必要があります。各チャネルの各割込みソースのマスクビットも、OTG_FS_HCINTMSK-x レジスタで使用可能です。

- ホストコアは、以下のステータスチェックおよび割込み生成を行います。
 - データ転送がアプリケーション側 (AHB) と USB 側の両方で完了したことを示す転送完了割込み
 - 転送完了、USB トランザクションエラー、またはアプリケーションからの無効化コマンドのためにチャネルが停止したかどうか
 - 関連する送信 FIFO が半エンプティか完全エンプティか (IN エンドポイント)
 - ACK 応答を受け取った時
 - NAK 応答を受け取った時
 - STALL 応答を受け取った時
 - CRC エラー、タイムアウト、ビットスタップエラー、偽 EOP による USB トランザクションエラー
 - バブルエラー
 - フレームオーバーラン
 - データトグルエラー

22.6.4 ホストスケジューラ

ホストコアは、アプリケーションによってポストされた USB トランザクションリクエストを自律的に並べ替えて、管理する組み込みハードウェアスケジューラを備えています。各フレームの開始時に、ホストは、周期的 (アイソクロナスおよびインタラプト) トランザクションを最初に実行し、次に、非周期的 (コントロールおよびバULK) トランザクションを行うことによって USB の仕様に従って、アイソクロナスおよびインタラプト転送タイプを優先します。

ホストは、リクエストキュー (1 つは周期的転送のキュー、もう 1 つは非周期的転送のキュー) を通じて USB トランザクションを処理します。各リクエストキューは、最大 8 個のエントリを保持できます。各エントリはアプリケーションからのペンディング中のトランザクションリクエストを表し、IN または OUT チャネル番号と USB 上でトランザクションを実行するためのその他の情報を保持しています。リクエストがキューに書き込まれた順序によって USB インタフェース上のトランザクションの順序が決まります。

各フレームの先頭で、ホストは、まず、周期的リクエストキューを処理し、次に、非周期的リクエストキューを処理します。現在のフレームに対してスケジュールされたアイソクロナスまたはインタラプトトランザクションが現在のフレームの終了時点でまだ処理されていない場合、ホストは、不完全周期転送割込み (OTG_FS_GINTSTS の IPXFER ビット) を発行します。OTG_HS コアは周期的と非周期的リクエストキューの管理を行います。周期的送信 FIFO およびキューステータスレジスタ (HPTXSTS) と非周期的送信 FIFO およびキューステータスレジスタ (HNPTXSTS) は読み出し専用レジスタであり、アプリケーションから各リクエストキューのステータスを読み出すときに使うことができます。これらのレジスタには次が含まれます。

- 周期 (非周期) リクエストキューで現在使用可能なフリーエントリの数 (最大 8 エントリ)。

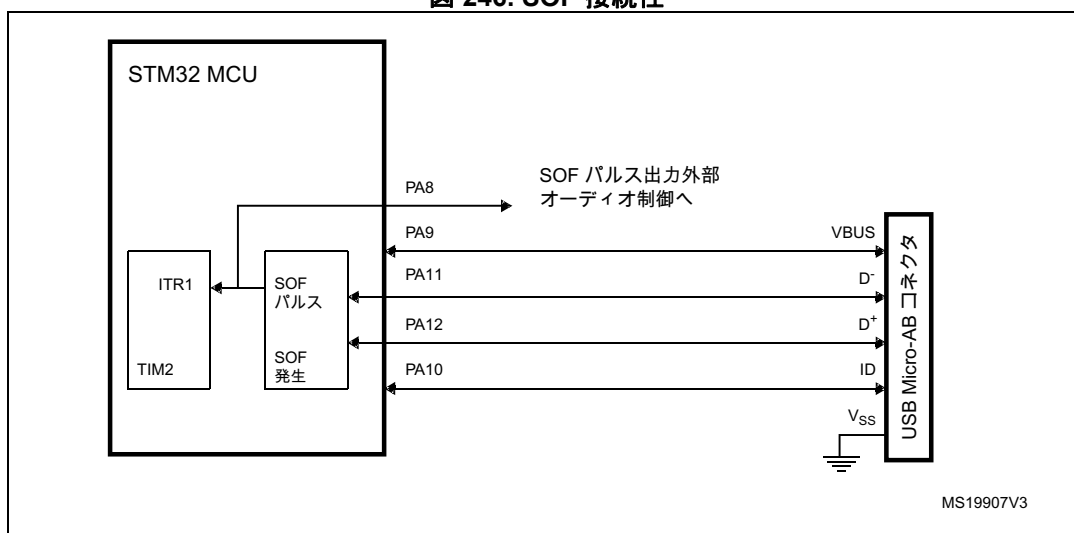
- 周期（非周期）Tx FIFO（OUT トランザクション）で現在使用可能な空きスペース
- IN/OUT トークン、ホストチャンネル番号、およびその他のステータス情報

リクエストキューはそれぞれ最大 8 つのエントリを保持できるので、アプリケーションは、最大 8 つのペンディング中の周期トランザクションに加えて、8 つのペンディング中の非周期トランザクションについて、SB に物理的に到達する前に、ホストトランザクションを事前にスケジュールできます。

ホストスケジューラ（キュー）へトランザクションリクエストをポストするときは、アプリケーションは OTG_FS_HNPTXSTS レジスタの PTXQSAV ビット、または OTG_FS_HNPTXSTS レジスタの NPTQSAV ビットを読み出して、周期的（非周期的）リクエストキュー内に少なくとも 1 個のエントリが存在することを確認する必要があります。

22.7 SOF トリガ

図 246. SOF 接続性



OTG FS コアは、ホスト、ペリフェラル、SOF パルス出力接続機能内で SOF フレーミングを監視、追跡、設定する手段を提供します。

これらのユーティリティは、オーディオペリフェラルが PC によって提供されるアイソクロナスストリームに同期する必要がある場合や、ホストがオーディオペリフェラルの要件に従ってフレーミング速度をトリム（微調整）する必要がある場合など、適応型オーディオクロック生成技術にとって特に役立ちます。

22.7.1 ホストの SOF

ホストモードでは、2 つの連続した SOF (FS) またはキープアライブ (LS) トークンの生成の間に発生する PHY クロックの数は、ホストフレームインターバルレジスタ (HFIR) でプログラム可能であり、SOF フレーミング周期に対する制御をアプリケーションに与えます。割込みは、どのフレームの開始時点でも生成されます (OTH_FS_GINTSTS の SOF ビット)。現在のフレーム番号と次の SOF までの残り時間が、ホストフレーム番号レジスタ (HFNUM) で追跡されます。

SOF パルス信号は、任意の SOF 開始トークンで生成され、幅は 12 システムクロックサイクルであり、グローバル制御および設定レジスタの SOFOUTEN ビットを使用して、SOF ピン上で外部的に使用可能になります。SOF パルスは、タイマ 2 (TIM2) の入力トリガにも内部接続されるので、入力キャプチャ機能、出力比較機能、およびタイマを SOF パルスによってトリガできます。TIM2 接続は レジスタを使って有効化します。

22.7.2 ペリフェラルの SOF

デバイスモードでは、フレーム割込みの開始は、SOF トークンが USB 上で受信されるたびに生成されます (OTH_FS_GINTSTS の SOFT ビット)。対応するフレーム番号は、デバイスステータスレジスタ (OTG_FS_DSTS の FNSOF ビット) から読み出すことができます。12 システムクロックサイクルの幅を持つ SOF パルス信号も生成され、グローバル制御および設定レジスタの SOF 出力イネーブルビット (OTG_FS_GCCFG の SOFOUTEN ビット) を使用して、SOF ピン上で外部的に使用可能になります。SOF パルス信号は、タイマ 2 (TIM2) 入力トリガにも内部接続されるので、入力キャプチャ機能、出力比較機能、およびタイマを SOF パルスによってトリガできます。TIM2 接続は、有効化されます。

周期的フレーム割込みの終わり (GINTSTS/EOPF) は、デバイス設定レジスタの周期的フレームインターバルフィールド (OTG_FS_DCFG の PFIVL ビット) に応じて、タイムフレームインターバルの 80%、85%、90%、または 95% が経過した時点をアプリケーションに通知するために使用されます。この機能を使用して、そのフレームのアイソクロナストラフィックのすべてが完了したかどうかを判断できます。

22.8 電力オプシオン

OTG PHY の消費電力は、一般コア設定レジスタの 3つのビットによって制御されます。

- PHY パワーダウン (GCCFG/PWRDWN)
PHY のフルスピードトランシーバモジュールのオン/オフを切り替えます。事前にセットして、すべての USB 動作を可能にする必要があります。
- A- V_{BUS} センシイネーブル (GCCFG/VBUSASEN)
A デバイスの動作に関連する V_{BUS} コンパレータのオン/オフを切り替えます。A デバイス (USB ホスト) モードおよび HNP 中に、これをセットする必要があります。
- B- V_{BUS} センシイネーブル (GCCFG/VBUSASEN)
B デバイスの動作に関連する V_{BUS} コンパレータのオン/オフを切り替えます。B デバイス (USB ペリフェラル) モードおよび HNP 中に、これをセットする必要があります。

USB セッションがまだ有効でないか、デバイスが切断されているときには、USB サスペンド状態で節電テクニックを使用できます。

- ストップ PHY クロック (OTG_FS_PCGCCTL の STPPCLK ビット)。
クロックゲーティング制御レジスタのストップ PHY クロックビットをセットすると、OTG 全速コア内部の 48 MHz クロックドメインのほとんどがクロックゲーティングによってスイッチオフされます。USB クロック切り替え機能による動的な電力消費は、48 MHz クロック入力がアプリケーションによって動作状態を維持された場合でも節減されます。
ほとんどのトランシーバも無効になり、非同期レジュームやリモートウェイクアップイベントの検出を担当する部分だけが動作状態を維持します。
- ゲート HCLK (OTG_FS_PCGCCTL の GATEHCLK ビット)。
クロックゲーティング制御レジスタのゲート HCLK ビットをセットすると、OTG フルスピードコア内部のシステムクロックドメインのほとんどがクロックゲーティングによってスイッチオフされます。レジスタ読出しおよび書込みインタフェースのみが動作状態を維持します。USB クロック切り替え機能による動的な電力消費は、システムクロックが他の目的のためにアプリケーションによって動作状態を維持されている場合でも節減されます。
- USB システムの停止
OTG_FS が USB サスペンド状態にある場合、アプリケーションはシステム内のすべてのクロックソースを完全にシャットダウンすることによって、電力消費全体を大幅に削減する方法を取ることができます。USB システムの停止は、まず、ストップ PHY クロックビットをセットし、次

に、パワー制御システムモジュール (PWR) でシステムディープスリープモードを設定することによって有効になります。

OTG_FS コアは、USB 上でリモートウェイクアップ (ホストとして) またはリモートレジューム (デバイスとして) 信号の非同期検出によって、システムと USB クロックの両方を自動的に再起動します。

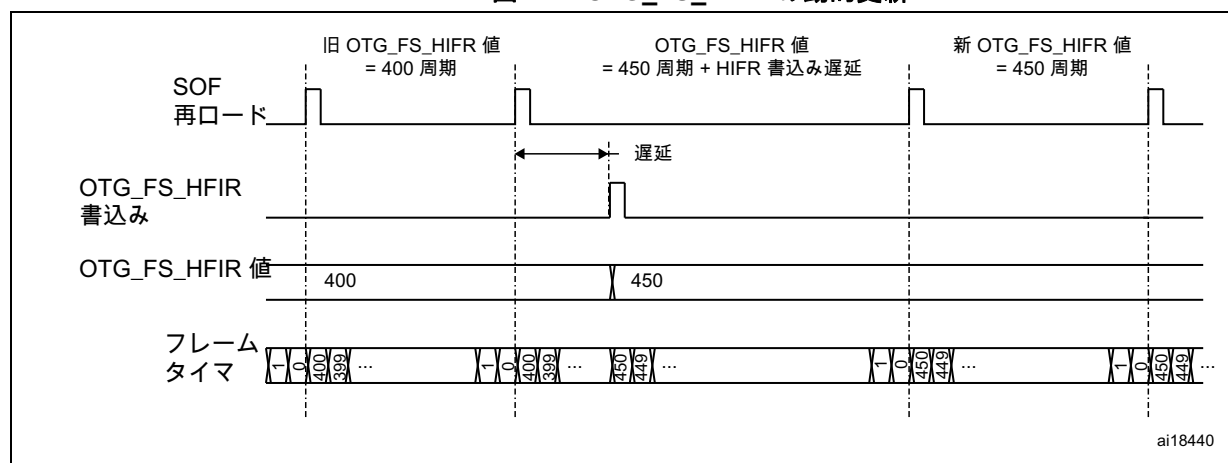
動的消費電力を節減するために、USB データ FIFO は、OTG_FS コアによってアクセスされたときだけクロック駆動されます。

22.9 OTG_FS_HFIR レジスタの動的更新

USB コアはホストモードでマイクロ SOF フレーミング周期を動的にトリミングする機能を内蔵しています。この機能を使うと、外部デバイスをマイクロ SOF フレームと同期させることができます。

現在のマイクロ SOF フレーム内で OTG_HS_HFIR レジスタを変更すると、[図 247](#) で説明するように次のフレームで SOF 周期補正が適用されます。

図 247. OTG_FS_HFIR の動的更新

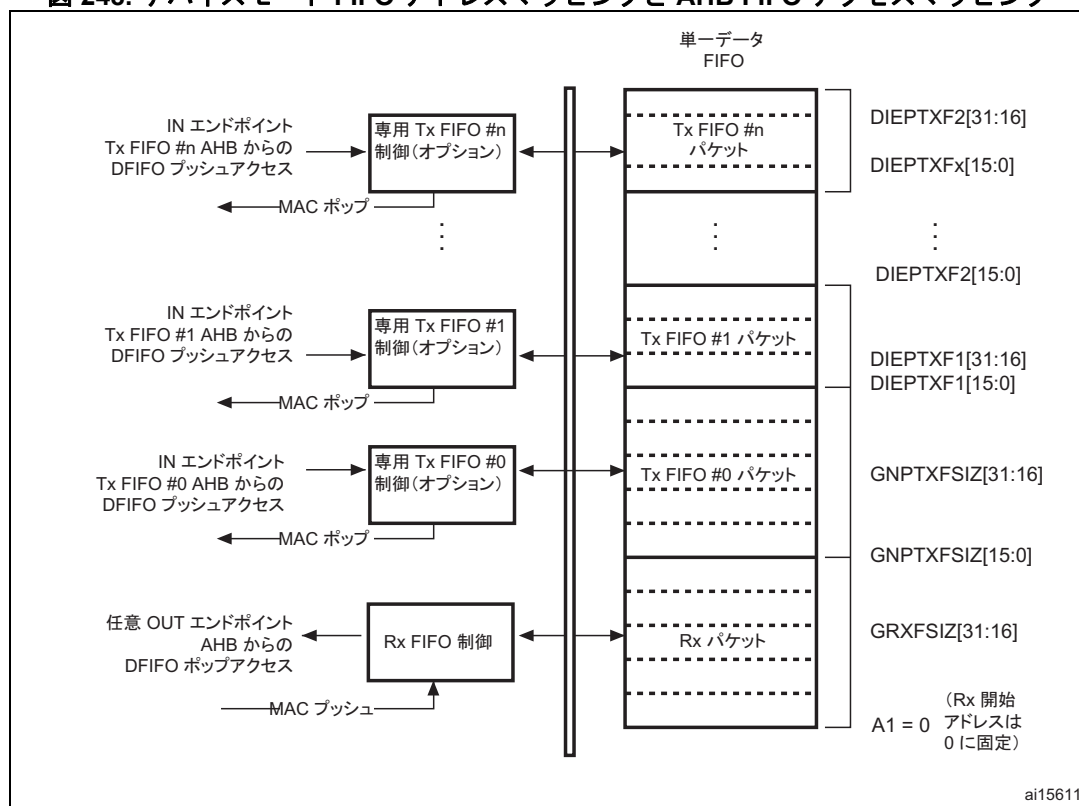


22.10 USB データ FIFO

USB システムは、高度な FIFO 制御メカニズムを持つ 1.25 KB の専用 RAM を備えています。OTG_FS コアのパケット FIFO コントローラモジュールは、RAM 空間を、アプリケーションが USB 転送の前に一時的にデータを格納する (プッシュする) Tx-FIFO と、USB から受信したデータをアプリケーションが取得する (ポップする) 前に一時的に格納される Rx FIFO に編成します。割り当てられる FIFO の数と、これらが RAM の内部でどのように構成されるかは、デバイスの役割に依存します。ペリフェラルモードでは、追加の Tx-FIFO が有効な IN エンドポイントのそれぞれに対して割り当てられます。FIFO のサイズは、アプリケーションの要件に合うように、ソフトウェアで設定できます。

22.11 ペリフェラル FIFO アーキテクチャ

図 248. デバイスモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング



22.11.1 ペリフェラル Rx FIFO

OTG ペリフェラルは、すべての OUT エンドポイントに向けたデータを受信する単一の受信 FIFO を使用します。受信パケットは、Rx-FIFO の空きスペースが使用可能になるまで、連続的にスタックされます。受信パケットのステータス (OUT エンドポイントの宛先番号、バイト数、データ PID、および受信データの有効性を含みます) も、コアによってデータペイロードの最上部に格納されます。使用可能なスペースがなくなると、ホストランザクションは NACK され、アドレス指定されたエンドポイントに割込みが受信されます。受信 FIFO のサイズは、受信 FIFO サイズレジスタ (GRXFSIZ) で設定されます。

単一受信 FIFO アーキテクチャは、USB ペリフェラルによる受信 RAM バッファへの書き込みをより効率的にします。

- すべての OUT エンドポイントは同じ RAM バッファを共有します (共有 FIFO)。
- OTG FS コアは、OUT トークンのいかなるホストシーケンスについても、限度まで受信 FIFO に書き込むことができます。

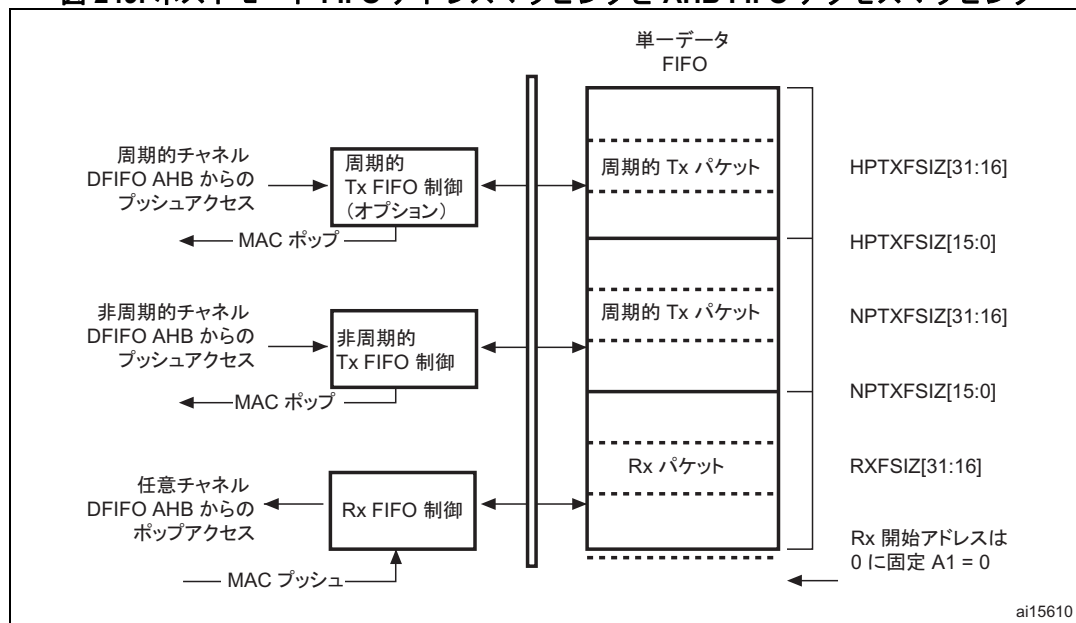
アプリケーションは Rx-FIFO 非エンプティ割込み (OTG_FS_GINTSTS の RXFLVL ビット) の受信を、少なくとも 1 パケット以上がダウンロード可能である限り継続します。アプリケーションは、受信ステータス読出しおよびポップレジスタ (GRXSTSP) からパケット情報を読み出し、最後に、エンドポイントに関するポップアドレスから読み取ることによって受信 FIFO からデータを吸い上げます。

22.11.2 ペリフェラル Tx FIFO

コアは、各 IN エンドポイント用の専用 FIFO を備えています。アプリケーションは、IN エンドポイント 0 については非周期的送信 FIFO サイズレジスタ (OTG_FS_TX0FSIZ) に書き込み、IN エンドポイント x についてはデバイス IN エンドポイント送信 FIFOx レジスタ (DIEPTXFx) に書き込むことによって、FIFO のサイズを設定します。

22.12 ホスト FIFO アーキテクチャ

図 249. ホストモード FIFO アドレスマッピングと AHB FIFO アクセスマッピング



22.12.1 ホスト Rx FIFO

ホストは、すべての周期的および非周期的トランザクションについて 1 つの受信 FIFO を使用します。この FIFO は、USB から受信したデータ (受信パケットのペイロード) をシステムメモリに転送されるまで保持する受信バッファとして使用されます。任意のリモート IN エンドポイントから受信されたパケットは、空きスペースがある限り連続的にスタックされます。各受信パケット (ホストチャネルデスティネーション、バイト数、データ PID、および受信データの有効性を含みます) のステータスも、FIFO に格納されます。受信 FIFO のサイズは、受信 FIFO サイズレジスタ (GRXFSIZ) で設定されます。

単一受信 FIFO アーキテクチャは、USB ホストによる受信データバッファへの書き込みをより効率的にします。

- IN に設定されたすべてのホストチャネルは、同じ RAM バッファを共有します (共有 FIFO)。
- OTG FS コアは、ホストのソフトウェアによって駆動された IN トークンのいかなるシーケンスについても、限度まで受信 FIFO に書き込むことができます。

アプリケーションは、少なくとも 1 つのパケットがダウンロード可能である限り、Rx FIFO ノットエンプティ割り込みを受け取ります。アプリケーションは、受信ステータス読出しおよびポップレジスタからパケット情報を読み出し、最後に、受信 FIFO からデータを吸い上げます。

22.12.2 ホスト Tx FIFO

ホストは、すべての非周期的（コントロールおよびバルク）OUT トランザクションについて 1 つの送信 FIFO を、すべての周期的（アイソクロナスおよびインタラプト）OUT トランザクションについて 1 つの送信 FIFO を使用します。FIFO は、USB 経由で送信されるデータ（送信パケットのペイロード）を保持する送信バッファとして使用されます。周期的（非周期的）Tx FIFO のサイズは、ホストの周期的（非周期的）送信 FIFO サイズレジスタ（HPTXFSIZ/HNPTXFSIZ）で設定されます。

Tx FIFO が 2 つ実装されているのは、USB フレームよりも周期的トラフィックタイプに対して高い優先度が与えられていることによります。各フレームの先頭で、組み込みのホストスケジューラは、まず、周期的リクエストキューを処理し、次に、非周期的リクエストキューを処理します。

2 つの送信 FIFO アーキテクチャにより、USB ホストは周期的送信データバッファと非周期的送信データバッファを個別に最適化できます。

- OUT 方向の周期的（非周期的）トランザクションをサポートするように設定されたすべてのホストチャネルは、同じ RAM バッファを共有します（共有 FIFO）。
- OTG FS コアは、ホストのソフトウェアによって駆動された OUT トークンのいかなるシーケンスについても、周期的送信 FIFO に書き込むことができます。

OTG_FS コアは、AHB 設定レジスタの周期的 Tx FIFO エンプティレベルビット（OTG_FS_GAHBCFG の PTXFELVL ビット）の値に応じて、周期的 Tx FIFO が半分または完全にエンプティである限り、周期的 Tx FIFO エンプティ割込み（OTG_FS_GINTSTS の PTXFE ビット）を発行します。アプリケーションは、周期的 Tx FIFO と周期的リクエストキューの両方に空きスペースが存在する限り、前もって送信データをプッシュすることができます。両方のレジスタにどれだけのスペースがあるかは、ホストの周期的送信 FIFO およびキューステータスレジスタ（HPTXSTS）を読み出すことで知ることができます。

OTG_FS コアは、AHB 設定レジスタの非周期的 Tx FIFO エンプティレベルビット（OTG_FS_GAHBCFG の TXFELVL ビット）の値に応じて、非周期的 Tx FIFO が半分または完全にエンプティである限り、非周期的 Tx FIFO エンプティ割込み（OTG_FS_GINTSTS の NPTXFE ビット）を発行します。アプリケーションは、非周期的 Tx FIFO と非周期的リクエストキューの両方に空きスペースが存在する限り、送信データをプッシュすることができます。両方のレジスタにどれだけのスペースがあるかは、ホストの非周期的送信 FIFO およびキューステータスレジスタ（HNPTXSTS）を読み出すことで知ることができます。

22.13 FIFO RAM の割り当て

22.13.1 デバイスモード

受信 FIFO RAM 割り当て：アプリケーションは SETUP パケット用に RAM を割り当てる必要があります。コントロールエンドポイントで SETUP パケットを受信するため受信 FIFO に 10 ロケーションを確保する必要があります。コアは、SETUP パケット用に予約されている位置に他のデータを書き込むことはありません。1 つの位置は、グローバル OUT NAK のために割り当てられます。各受信パケットとともに、ステータス情報が FIFO に書き込まれます。したがって、パケットを受信するには、少なくとも、（最大パケットサイズ / 4）+ 1 のスペースを割り当てる必要があります。複数のアイソクロナスエンドポイントが有効な場合、連続パケットを受信するには、少なくとも、（最大パケットサイズ / 4）+ 1 の 2 倍のスペースが割り当てられなければなりません。一般に、（最大パケットサイズ / 4）+ 1 の 2 倍のスペースが推奨されます。前のパケットが CPU に転送されている間に、USB は次のパケットを受信できるからです。

各エンドポイントの最後のパケットとともに、送信完了ステータス情報が FIFO にプッシュされます。一般に、各 OUT エンドポイントに対して 1 つのロケーションを予約することが推奨されます。

送信 FIFO RAM の割り当て： 各 IN エンドポイント送信 FIFO に必要な最小 RAM スペースは、その特定の IN エンドポイントに対する最大パケットサイズを収容できるスペースとなります。

注： 送信 IN エンドポイント FIFO に割り当てるスペースが大きいほど、USB のパフォーマンスは高くなります。

22.13.2 ホストモード

受信 FIFO RAM 割り当て

各受信パケットとともに、ステータス情報が FIFO に書き込まれます。したがって、パケットを受信するには、少なくとも、 $(\text{最大パケットサイズ} / 4) + 1$ のスペースを割り当てる必要があります。複数のアイソクロナスチャネルが有効な場合、連続パケットを受信するには、少なくとも、 $(\text{最大パケットサイズ} / 4) + 1$ の 2 倍のスペースが割り当てられなければなりません。一般に、 $(\text{最大パケットサイズ} / 4) + 1$ の 2 倍のスペースが推奨されます。前のパケットが CPU に転送されている間に、USB は次のパケットを受信できるからです。

ホストチャネルの最後のパケットとともに、転送完了ステータス情報が FIFO にプッシュされます。したがって、このために 1 つの位置が割り当てられなければなりません。

送信 FIFO RAM 割り当て

ホストの非周期的送信 FIFO に必要な最小 RAM スペースは、サポートされるすべての非周期的 OUT チャネルの中で最大のパケットサイズを収容できるスペースとなります。

一般に、最大パケットサイズの 2 倍のスペースが推奨されます。現在のパケットが USB に転送されているときに、CPU が次のパケットを取得できるからです。

ホストの周期的送信 FIFO に必要な最小 RAM スペースは、サポートされるすべての周期的 OUT チャネルの中で最大のパケットサイズを収容できるスペースとなります。少なくとも 1 つのアイソクロナス OUT エンドポイントがある場合には、スペースはそのチャネルの最大パケットサイズの少なくとも 2 倍でなければなりません。

注： 非周期的送信 FIFO に割り当てるスペースが大きいほど、USB のパフォーマンスは高くなります。

22.14 USB システムのパフォーマンス

USB とシステムの最高のパフォーマンスを達成するには、大きな RAM バッファ、柔軟に設定可能な FIFO サイズ、AHB プッシュ/ポップレジスタを通した迅速な 32 ビット FIFO アクセス、そして、特に、高度な FIFO 制御メカニズムが必要です。このメカニズムにより、OTG_FS は、現在の USB シーケンスに関係なく、使用可能な RAM スペースに効率的に書き込むことができます。これらの機能によって、以下のことが可能になります。

- アプリケーションは、CPU のバンド幅使用率を最適化するための介入を調整する十分な余裕ができます。
 - データを効率的に USB に送信できるときまで、大量の送信データを事前に蓄積することができます。
 - 単一の受信 FIFO からデータをダウンロードするための時間の余裕が増えるという利点があります。
- USB コアは、最高動作速度を維持できるので、アプリケーションの介入がある場合に比べて、自律的に動作できる幅が広がり、最大のフルスピードバンド幅を実現できます。
 - 大量の送信データを自由に蓄積して、USB 経由のデータ送信を自律的に管理できます。
 - 受信バッファの使用可能な空きスペースが大量にあるので、USB から入ってくるデータを自律的に書き込むことができます。

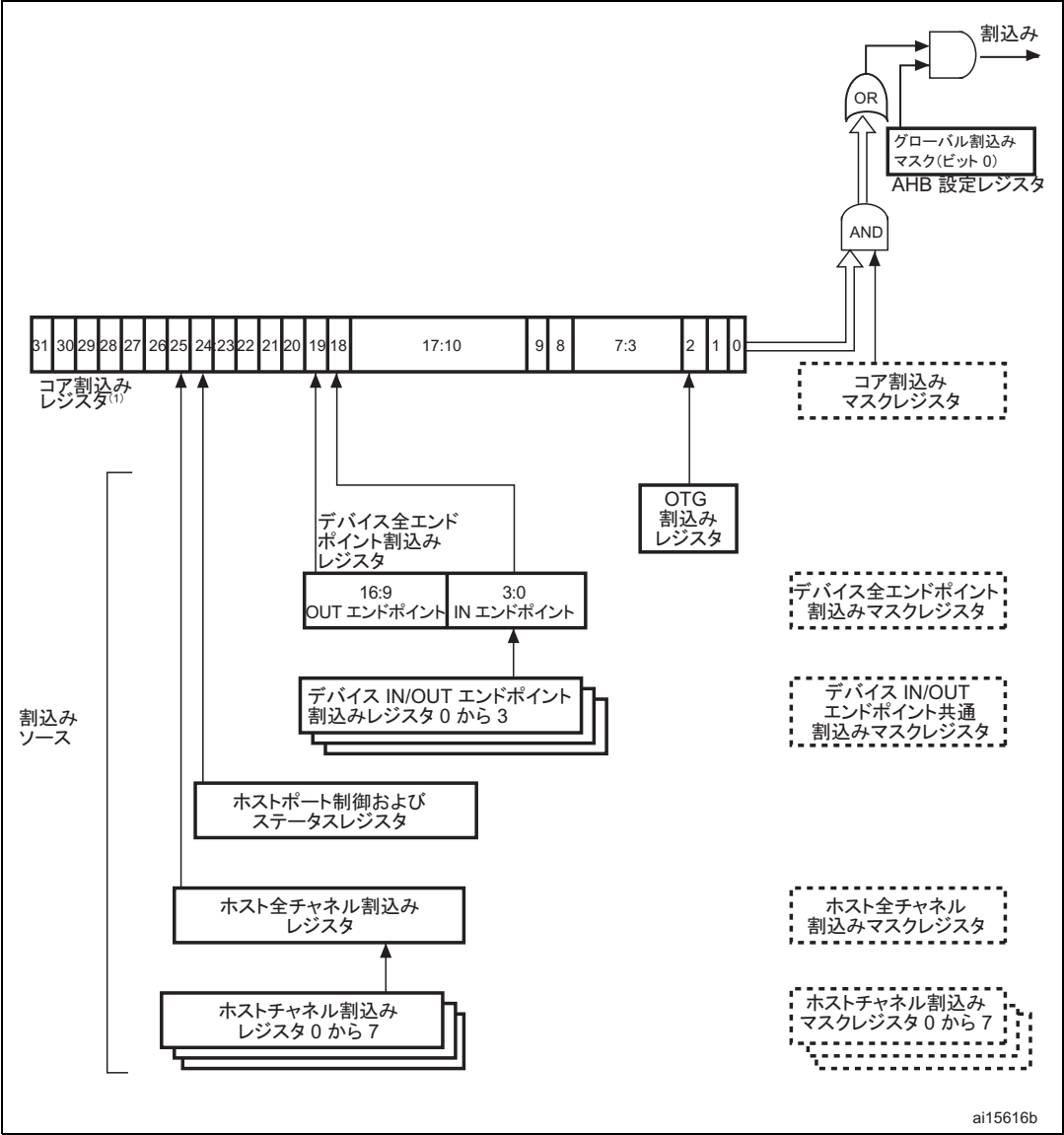
OTG_FS コアは 1.25 KB の RAM バッファを非常に効率的に使用でき、また、1.25 KB の送信／受信データはフルスピードフレームをカバーするには十分以上なので、USB システムは CPU の介入がなくても、最大 1 USB フレーム (1 ms) に対応する最高のフルスピードデータ速度に耐えることができます。

22.15 OTG_FS 割込み

OTG_FS コントローラがデバイスモードまたはホストモードのいずれかで動作しているときには、アプリケーションは、別のモードからレジスタにアクセスしてはなりません。不正なアクセスが発生した場合、モード不一致割込みが生成され、コア割込みレジスタ (OTG_FS_GINTSTS レジスタの MMIS ビット) に反映されます。コアがあるモードから別のモードに切り替わったとき、新しい動作モードにおけるレジスタは、パワーオンリセット後と同様に再プログラムされなければなりません。

[図 250](#) に、割込みの階層を示します。

図 250. 割り込みの階層



1. コア割り込みレジスタビットは、698 ページのOTG_FS コア割り込みレジスタ (OTG_FS_GINTSTS) に示されています。

22.16 OTG_FS 制御／ステータスレジスタ

アプリケーションは、AHB スレーブインタフェースを通じて制御／ステータスレジスタ (CSR) を読み書きすることによって、OTG_FS コントローラを制御します。これらのレジスタは 32 ビット幅であり、アドレスは 32 ビットブロックに配置されています。OTG_FS レジスタは、ワード (32 ビット) 単位でアクセスする必要があります。

CSR は、次のように分類されます。

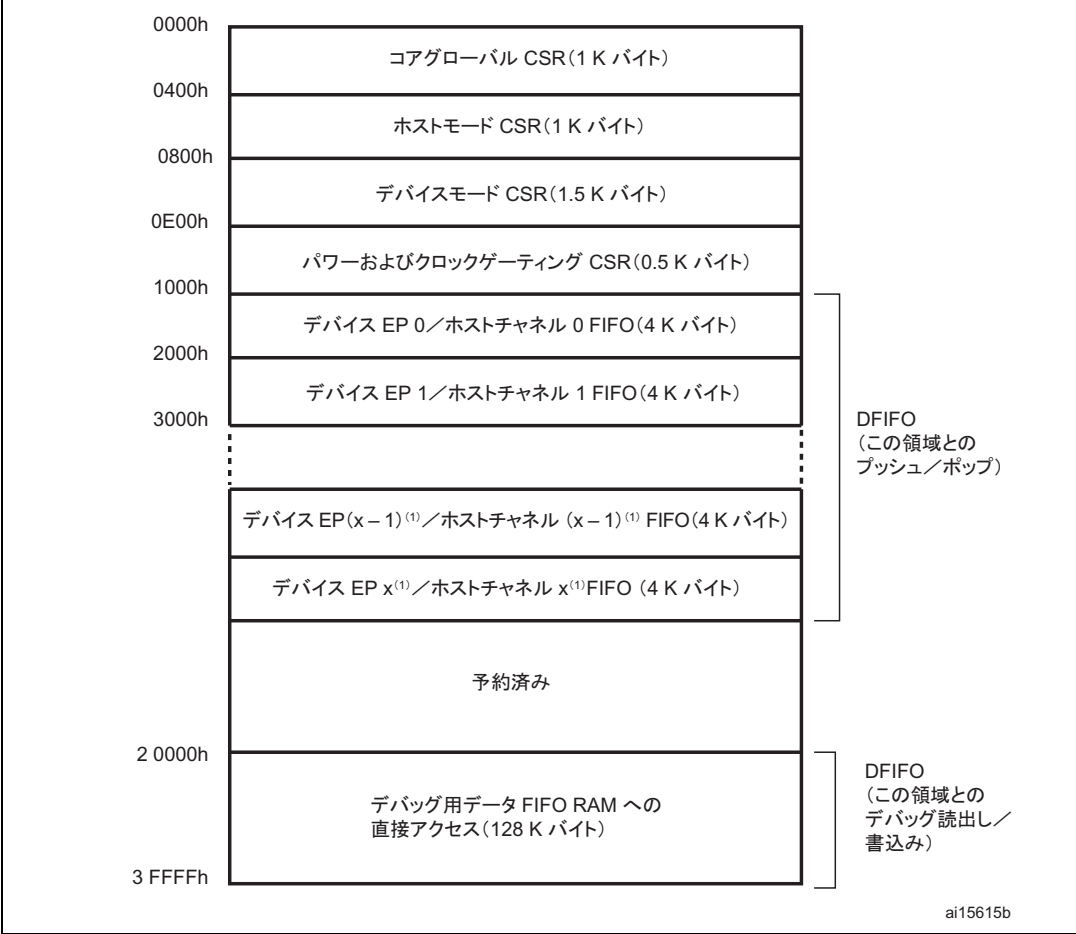
- コアグローバルレジスタ
- ホストモードレジスタ
- ホストグローバルレジスタ
- ホストポート CSR
- ホストチャネル固有レジスタ
- デバイスモードレジスタ
- デバイスグローバルレジスタ
- デバイスエンドポイント固有レジスタ
- パワーおよびクロックゲーティングレジスタ
- データ FIFO (DFIFO) アクセスレジスタ

コアグローバル、パワーおよびクロックゲーティング、データ FIFO (DFIFO) アクセス、およびホストポート CSR レジスタのみが、ホストモードおよびデバイスモードの両方でのアクセスが可能です。OTG_FS コントローラがデバイスモードまたはホストモードのいずれかで動作しているときには、アプリケーションは、別のモードからレジスタにアクセスしてはなりません。不正なアクセスが発生した場合、モード不一致割込みが生成され、コア割込みレジスタ (OTG_FS_GINTSTS レジスタの MMIS ビット) に反映されます。コアがあるモードから別のモードに切り替わったとき、新しい動作モードにおけるレジスタは、パワーオンリセット後と同様に再プログラムされなければなりません。

22.16.1 CSR メモリマップ

ホストモードおよびデバイスモードレジスタは、それぞれ異なるアドレスを占めます。すべてのレジスタは AHB クロックドメインで実装されます。

図 251. CSR メモリマップ



1. デバイスモードでは $x = 3$ 、ホストモードでは $x = 7$ です。

グローバル CSR マップ

これらのレジスタは、ホストモードとデバイスモードの両方で使用可能です。

表 126. コアグローバル制御およびステータスレジスタ (CSR)

項目 (略称)	アドレス オフセット	レジスタ名
OTG_FS_GOTGCTL	0x000	690 ページの OTG_FS 制御およびステータスレジスタ (OTG_FS_GOTGCTL)
OTG_FS_GOTGINT	0x004	692 ページの OTG_FS 割込みレジスタ (OTG_FS_GOTGINT)
OTG_FS_GAHBCCFG	0x008	693 ページの OTG_FS AHB 設定レジスタ (OTG_FS_GAHBCCFG)
OTG_FS_GUSBCCFG	0x00C	694 ページの OTG_FS USB 設定レジスタ (OTG_FS_GUSBCCFG)
OTG_FS_GRSTCTL	0x010	696 ページの OTG_FS リセットレジスタ (OTG_FS_GRSTCTL)

表 126. コアグローバル制御およびステータスレジスタ (CSR) (続き)

項目 (略称)	アドレス オフセット	レジスタ名
OTG_FS_GINTSTS	0x014	698 ページのOTG_FS コア割込みレジスタ (OTG_FS_GINTSTS)
OTG_FS_GINTMSK	0x018	702 ページのOTG_FS 割込みマスクレジスタ (OTG_FS_GINTMSK)
OTG_FS_GRXSTSR	0x01C	705 ページのOTG_FS 受信ステータスデバッグ読出し/OTG ステータス読出しおよびポップレジスタ (OTG_FS_GRXSTSR/OTG_FS_GRXSTSP)
OTG_FS_GRXSTSP	0x020	
OTG_FS_GRXFSIZ	0x024	706 ページのOTG_FS 受信 FIFO サイズレジスタ (OTG_FS_GRXFSIZ)
OTG_FS_HNPTXFSIZ/ OTG_FS_DIEPTXF0 ⁽¹⁾	0x028	OTG_FS ホスト非周期的送信 FIFO サイズレジスタ (OTG_FS_HNPTXFSIZ) / エンドポイント 0 送信 FIFO サイズ (OTG_FS_DIEPTXF0)
OTG_FS_HNPTXSTS	0x02C	708 ページのOTG_FS 非周期的送信 FIFO/キューステータスレジスタ (OTG_FS_HNPTXSTS)
OTG_FS_GCCFG	0x038	709 ページのOTG_FS 一般コア設定レジスタ (OTG_FS_GCCFG)
OTG_FS_CID	0x03C	710 ページのOTG_FS コア ID レジスタ (OTG_FS_CID)
OTG_FS_HPTXFSIZ	0x100	710 ページのOTG_FS ホスト周期的送信 FIFO サイズレジスタ (OTG_FS_HPTXFSIZ)
OTG_FS_DIEPTFXx	0x104	710 ページのOTG_FS デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_FS_DIEPTFXx) (x = 1..3, x は FIFO_number)
	0x124	
	...	
	0x138	

1. 一般的ルールとして、OTG_FS_HNPTXFSIZ はホストモードで、OTG_FS_DIEPTXF0 はデバイスモードで使用されます。

ホストモード CSR マップ

これらのレジスタは、コアがホストモードに変わるたびにプログラムされなければなりません。

表 127. ホストモード制御およびステータスレジスタ (CSR)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_FS_HCFG	0x400	711 ページのOTG_FS ホスト設定レジスタ (OTG_FS_HCFG)
OTG_FS_HFIR	0x404	712 ページのOTG_FS ホストフレームインターバルレジスタ (OTG_FS_HFIR)
OTG_FS_HFNUM	0x408	712 ページのOTG_FS ホストフレーム番号/残りフレーム時間レジスタ (OTG_FS_HFNUM)
OTG_FS_HPTXSTS	0x410	713 ページのOTG_FS ホスト周期的送信 FIFO/キューステータスレジスタ (OTG_FS_HPTXSTS)
OTG_FS_HAINT	0x414	714 ページのOTG_FS ホスト全チャネル割込みレジスタ (OTG_FS_HAINT)
OTG_FS_HAINTMSK	0x418	714 ページのOTG_FS ホスト全チャネル割込みマスクレジスタ (OTG_FS_HAINTMSK)
OTG_FS_HPRT	0x440	715 ページのOTG_FS ホストポート制御およびステータスレジスタ (OTG_FS_HPRT)

表 127. ホストモード制御およびステータスレジスタ (CSR) (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_FS_HCCHARx	0x500 0x520 ... 0x6E0h	717 ページのOTG_FS ホストチャネル x 特性レジスタ (OTG_FS_HCCHARx) (x = 0..7, x = Channel_number)
OTG_FS_HCINTx	508h	718 ページのOTG_FS ホストチャネル x 割込みレジスタ (OTG_FS_HCINTx) (x = 0..7, x = Channel_number)
OTG_FS_HCINTMSKx	50Ch	719 ページのOTG_FS ホストチャネル x 割込みマスクレジスタ (OTG_FS_HCINTMSKx) (x = 0..7, x = Channel_number)
OTG_FS_HCTSIZx	510h	720 ページのOTG_FS ホストチャネル x 転送サイズレジスタ (OTG_FS_HCTSIZx) (x = 0..7, x = Channel_number)

デバイスモード CSR マップ

これらのレジスタは、コアがデバイスモードに変わるたびにプログラムする必要があります。

表 128. デバイスモード制御およびステータスレジスタ

項目 (略称)	オフセット アドレス	レジスタ名
OTG_FS_DCFG	0x800	721 ページのOTG_FS デバイス設定レジスタ (OTG_FS_DCFG)
OTG_FS_DCTL	0x804	722 ページのOTG_FS デバイス制御レジスタ (OTG_FS_DCTL)
OTG_FS_DSTS	0x808	723 ページのOTG_FS デバイスステータスレジスタ (OTG_FS_DSTS)
OTG_FS_DIEPMSK	0x810	724 ページのOTG_FS デバイス IN エンドポイント共通割込みマスクレジスタ (OTG_FS_DIEPMSK)
OTG_FS_DOEPMSK	0x814	725 ページのOTG_FS デバイス OUT エンドポイント共通割込みマスクレジスタ (OTG_FS_DOEPMSK)
OTG_FS_DAIN	0x818	726 ページのOTG_FS デバイス全エンドポイント割込みレジスタ (OTG_FS_DAIN)
OTG_FS_DAINMSK	0x81C	726 ページのOTG_FS 全エンドポイント割込みマスクレジスタ (OTG_FS_DAINMSK)
OTG_FS_DVBUSDIS	0x828	727 ページのOTG_FS デバイス V _{BUS} 放電時間レジスタ (OTG_FS_DVBUSDIS)
OTG_FS_DVBUSPULSE	0x82C	727 ページのOTG_FS デバイス V _{BUS} パルシング時間レジスタ (OTG_FS_DVBUSPULSE)
OTG_FS_DIEPEMPMSK	0x834	728 ページのOTG_FS デバイス IN エンドポイント FIFO エンプティ割込みマスクレジスタ : (OTG_FS_DIEPEMPMSK)
OTG_FS_DIEPCTL0	0x900	728 ページのOTG_FS デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_FS_DIEPCTL0)

表 128. デバイスモード制御およびステータスレジスタ (続き)

項目 (略称)	オフセット アドレス	レジスタ名
OTG_FS_DIEPCTLx	0x920 0x940 ... 0xAE0	730 ページのOTG デバイスエンドポイント x 制御レジスタ (OTG_FS_DIEPCTLx) (x = 1..3, x = Endpoint_number)
OTG_FS_DIEPINTx	0x908	736 ページのOTG_FS デバイスエンドポイント x 割込みレジスタ (OTG_FS_DIEPINTx) (x = 0..3, x = Endpoint_number)
OTG_FS_DIEPTSIZ0	0x910	738 ページのOTG_FS デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_FS_DIEPTSIZ0)
OTG_FS_DTXFSTSx	0x918	741 ページのOTG_FS デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_FS_DTXFSTSx) (x = 0..3, x = Endpoint_number)
OTG_FS_DIEPTSIZx	0x930 0x950 ... 0xAF0	741 ページのOTG_FS デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_FS_DIEPTSIZx) (x = 1..3, x = Endpoint_number)
OTG_FS_DOEPCTL0	0xB00	732 ページのOTG_FS デバイスコントロール OUT エンドポイント 0 制御レジスタ (OTG_FS_DOEPCTL0)
OTG_FS_DOEPCTLx	0xB20 0xB40 ... 0xCC0 0xCE0 0xCFD	730 ページのOTG デバイスエンドポイント x 制御レジスタ (OTG_FS_DIEPCTLx) (x = 1..3, x = Endpoint_number)
OTG_FS_DOEPINTx	0xB08	736 ページのOTG_FS デバイスエンドポイント x 割込みレジスタ (OTG_FS_DIEPINTx) (x = 0..3, x = Endpoint_number)
OTG_FS_DOEPTSIZx	0xB10	741 ページのOTG_FS デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_FS_DIEPTSIZx) (x = 1..3, x = Endpoint_number)

データ FIFO (DFIFO) アクセスレジスタマップ

これらのレジスタは、ホストモードとデバイスモードの両方で使用可能であり、特定のエンドポイントまたはチャネルの FIFO スペースを指定された方向で読み書きするために使用されます。ホストチャネルが IN タイプの場合 FIFO はそのチャネル上では読出しのみが可能です。同様に、ホストチャネルが OUT タイプの場合 FIFO はそのチャネル上では書込みのみが可能です。

表 129. データ FIFO (DFIFO) アクセスレジスタマップ

FIFO アクセスレジスタセクション	アドレス範囲	アクセス
デバイス IN エンドポイント 0 / ホスト OUT チャンネル 0 : DFIFO 書込みアクセス デバイス OUT エンドポイント 0 / ホスト IN チャンネル 0 : DFIFO 読出しアクセス	0x1000~0x1FFC	w r
デバイス IN エンドポイント 1 / ホスト OUT チャンネル 1 : DFIFO 書込みアクセス デバイス OUT エンドポイント 1 / ホスト IN チャンネル 1 : DFIFO 読出しアクセス	0x2000~0x2FFC	w r
...
デバイス IN エンドポイント x ⁽¹⁾ / ホスト OUT チャンネル x ⁽¹⁾ : DFIFO 書込みアクセス デバイス OUT エンドポイント x ⁽¹⁾ / ホスト IN チャンネル x ⁽¹⁾ : DFIFO 読出しアクセス	0xX000~0xXFFC	w r

1. デバイスモードでは x=3、ホストモードでは x=7 です。

パワーおよびクロックゲーティング CSR マップ

パワーおよびクロックゲーティングの場合、シングルレジスタです。このレジスタは、ホストモードとデバイスモードの両方で使用できます。

表 130. パワーおよびクロックゲーティング制御およびステータスレジスタ

レジスタ名	項目 (略称)	オフセットアドレス : 0xE00~0xFFFF
パワーおよびクロックゲーティング制御レジスタ	PCGCR	0xE00~0xE04
予約済み		0xE05~0xFFFF

22.16.2 OTG_FS グローバルレジスタ

これらのレジスタは、ホストモードとデバイスモードの両方で使用でき、これらのモード間で切り替わっても再プログラムの必要はありません。

レジスタの説明で示されているビット値は、特に記載がない限りバイナリで表されています。

OTG_FS 制御およびステータスレジスタ (OTG_FS_GOTGCTL)

アドレスオフセット : 0x000

リセット値 : 0x0000 0800

OTG_FS_GOTGCTL レジスタは、コアの OTG 機能の動作を制御し、そのステータスを反映します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み												BSVLD	ASVLD	DBCT	CIDSTS	予約済み				DHNPEN	HSNPEN	HNPRQ	HNGSCS	予約済み				SRQ	SRQSCS		
												r	r	r	r					rW	rW	rW	r					rW	r		

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 **BSVLD** : B セッション有効

デバイスモードでのトランシーバのステータスを示します。

0 : Bセッションは無効。

1 : Bセッションは有効。

OTG モードでは、このビットを使用して、デバイスが接続されているか、切断されているかを判断できます。

注 : デバイスモードでのみアクセス可能です。

ビット 18 **ASVLD** : A セッション有効

ホストモードでのトランシーバのステータスを示します。

0 : Aセッションは無効。

1 : Aセッションは有効。

注 : ホストモードでのみアクセス可能です。

ビット 17 **DBCT** : ロング/ショートデバウンス時間

検出された接続のデバウンス時間を示します。

0 : ロングデバウンス時間 (100 ms + 2.5 μs)。物理的接続に使用されます。

1 : ショートデバウンス時間 (2.5 μs)。ソフト接続に使用されます。

注 : ホストモードでのみアクセス可能です。

ビット 16 **CIDSTS** : コネクタ ID ステータス

接続イベント時のコネクタ ID のステータスを示します。

0 : OTG_FS コントローラは A デバイスモードです。

1 : OTG_FS コントローラは B デバイスモードです。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 15:12 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 **DHNPEN** : デバイス HNP 有効化

アプリケーションは、接続された USB ホストから SetFeature.SetHNPEnable コマンドを正常に受信したときに、このビットをセットします。

0 : HNP は、アプリケーションでは無効です。

1 : HNP は、アプリケーションで有効です。

注 : デバイスモードでのみアクセス可能です。

ビット 10 HSHNPEN : ホストセット HNP 有効化

アプリケーションは、接続されたデバイスで HNP を正常に有効できたとき (SetFeature.SethNPEnable コマンドを使用して)、このビットをセットします。

0 : ホストセット HNP は無効です。

1 : ホストセット HNP は有効です。

注 : ホストモードでのみアクセス可能です。

ビット 9 HNPRQ : HNP リクエストあり

アプリケーションは、接続された USB ホストへの HNP リクエストを開始するために、このビットをセットします。アプリケーションは、OTG_FS_GOTGINT レジスタのホストネゴシエーション正常ステータス変更ビット (OTG_FS_GOTGINT の HNSSCHG ビット) がセットされたとき、0 を書き込むことによって、このビットをクリアできます。コアは HNSSCHG ビットがクリアされたとき、このビットをクリアします。

0 : HNP リクエストなし

1 : HNP リクエストあり

注 : デバイスモードでのみアクセス可能です。

ビット 8 HNGSCS : ホストネゴシエーション成功

コアは、ホストネゴシエーションに成功したとき、このビットをセットします。コアは、このレジスタの HNP リクエストビット (HNPRQ) がセットされたとき、このビットをクリアします。

0 : ホストネゴシエーション失敗

1 : ホストネゴシエーション成功

注 : デバイスモードでのみアクセス可能です。

ビット 7:2 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 SRQ : セッションリクエスト

アプリケーションは USB 上でセッションリクエストを開始するために、このビットをセットします。アプリケーションは、OTG_FS_GOTGINT レジスタのホストネゴシエーション正常ステータス変更ビット (OTG_FS_GOTGINT の HNSSCHG ビット) がセットされたとき、0 を書き込むことによって、このビットをクリアできます。コアは HNSSCHG ビットがクリアされたとき、このビットをクリアします。USB 1.1 フルスピードシリアルトランシーバインタフェースを使用してセッションリクエストを開始する場合、アプリケーションは、このレジスタの B セッション有効ビット (OTG_FS_GOTGCTL の BSVLD ビット) がクリアされた後 V_{BUS} が 0.2 V へ放電するまで待つ必要があります。この放電時間は PHY ごとに異なり PHY ベンダーから得ることができます。

0 : セッションリクエストなし

1 : セッションリクエスト

注 : デバイスモードでのみアクセス可能です。

ビット 0 SRQSCS : セッションリクエスト成功

コアは、セッションリクエストの開始に成功したとき、このビットをセットします。

0 : セッションリクエスト失敗

1 : セッションリクエスト成功

注 : デバイスモードでのみアクセス可能です。

OTG_FS 割込みレジスタ (OTG_FS_GOTGINT)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アプリケーションは OTG 割込みがあると、このレジスタを読み出し、このレジスタのビットをクリアすることによって OTG 割込みをクリアします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み												DBCONE	ADTOCHG	HNGDET	予約済み						HNSSCHG	SRSSCHG	予約済み						SEDET	Res.	
												rc_w1	rc_w1	rc_w1							rc_w1	rc_w1							rc_w1		

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 **DBCONE** : デバウンス終了

コアは、デバイスが接続された後、デバウンスが完了したときに、このビットをセットします。アプリケーションは、この割込みを確認した後に USB リセットの駆動を開始できます。このビットは、OTG_FS_GUSBCFG レジスタの HNP 対応ビットまたは SRP 対応ビット (OTG_FS_GUSBCFG の HNPCAP ビットまたは SRPCAP ビット) がセットされたときのみ有効です。

注 : ホストモードでのみアクセス可能です。

ビット 18 **ADTOCHG** : A デバイスタイムアウト変更

コアは B デバイスの接続を待っている間に A デバイスがタイムアウトしたことを示すために、このビットをセットします。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 17 **HNGDET** : ホストネゴシエーションの検出

コアは USB 上でホストネゴシエーションリクエストを検出したときに、このビットをセットします。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 16:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **HNSSCHG** : ホストネゴシエーション成功ステータス変化

コアは USB ホストネゴシエーションリクエストの成功または失敗時に、このビットをセットします。アプリケーションは、OTG_FS_GOTGCTL レジスタのホストネゴシエーション成功ビット (OTG_FS_GOTGCTL の HNGSCS ビット) を読み出して、成功か失敗かをチェックする必要があります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 8 **SRSSCHG** : セッションリクエスト成功ステータス変化

コアは、セッションリクエストの成功または失敗のときに、このビットをセットします。アプリケーションは、OTG_FS_GOTGCTL レジスタのセッションリクエスト成功ビット (OTG_FS_GOTGCTL の SRQSCS ビット) を読み出して、成功か失敗かをチェックする必要があります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 2 **SEDET** : セッション終了検出

コアは、このビットをセットして、V_{BUS} の電圧レベルが B ペリフェラルセッションにとって有効でなくなったことを示します (V_{BUS} < 0.8 V のとき)。

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。

OTG_FS AHB 設定レジスタ (OTG_FS_GAHBCFG)

アドレスオフセット : 0x008

リセット値 : 0x0000 0000

このレジスタを使用して、電源投入後、またはモード変化後のコアを設定できます。このレジスタは、主に AHB システム関連の設定パラメータを含んでいます。初期プログラミングの後は、このレジスタを変更しないでください。アプリケーションは AHB または USB のいずれかでトランザクションを開始する前に、このレジスタをプログラムする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																							PTXFELVL	TXFELVL	予約済み				GINTMSK		
																							RW	RW					RW		

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 8 **PTXFELVL** : 周期的 TxFIFO エンプティレベル

OTG_FS_GINTSTS レジスタの周期的 TxFIFO エンプティ割込みビット (OTG_FS_GINTSTS の PTXFE ビット) がトリガされるときを示します。

0 : PTXFE (OTG_FS_GINTSTS の) 割込みビットは、周期的 TxFIFO が半分エンプティであることを示します。

1 : PTXFE (OTG_FS_GINTSTS の) 割込みビットは、周期的 TxFIFO が完全エンプティであることを示します。

注 : ホストモードでのみアクセス可能です。

ビット 7 **TXFELVL** : TxFIFO エンプティレベル

デバイスモードでは、このビットは、IN エンドポイント送信 FIFO エンプティ割込み (OTG_FS_DIEPINTx の TXFE) がトリガされるときを示します。

0 : TXFE (OTG_FS_DIEPINTx の) 割込みは、IN エンドポイントの Tx FIFO が半エンプティであることを示します。

1 : TXFE (OTG_FS_DIEPINTx の) 割込みは、IN エンドポイントの Tx FIFO が完全エンプティであることを示します。

ホストモードでは、このビットは、非周期的 Tx FIFO エンプティ割込み (OTG_FS_GINTSTS の NPTXFE ビット) がトリガされるタイミングを示します。

0 : NPTXFE (OTG_FS_GINTSTS の) 割込みは、非周期的 Tx FIFO が半エンプティであることを示します。

1 : NPTXFE (OTG_FS_GINTSTS の) 割込みは、非周期的 Tx FIFO が完全エンプティであることを示します。

ビット 6:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **GINTMSK** : グローバル割込みマスク

アプリケーションは、このビットを使用して、アプリケーション自身に対する割込みラインのアサーションをマスクまたはマスク解除します。このビットの設定に関係なく、割込みステータスレジスタはコアによって更新されます。

0 : アプリケーションへの割込みアサーションをマスクします。

1 : アプリケーションへの割込みアサーションをマスク解除します。

注 : デバイスモードとホストモードの両方でアクセス可能です。

OTG_FS USB 設定レジスタ (OTG_FS_GUSBCFG)

アドレスオフセット : 0x00C

リセット値 : 0x0000 0A00

このレジスタを使用して、電源投入後、またはホストモードあるいはデバイスモードへの変化後に、コアを設定することができます。このレジスタは USB および USB-PHY 関連の設定パラメータを含んでいます。アプリケーションは AHB または USB のいずれかでトランザクションを開始する前に、このレジスタをプログラムする必要があります。初期プログラミングの後には、このレジスタを変更しないでください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTXPKT	FDMOD	FHMOD	予約済み															TRDT		HNPCAP	SRPCAP	Res.	PHYSEL	予約済み	TOTAL						
rw	rw	rw																rw		rw	rw		wo		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 31 **CTXPKT** : 破損 Tx パケット

このビットは、デバッグ目的でのみ使用されます。このビットを 1 にセットしないでください。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 30 **FDMOD** : 強制デバイスモード

このビットに 1 を書き込むと、コアは OTG_FS_ID 入力ピンに関係なく、強制的にデバイスモードになります。

0 : 通常モード

1 : 強制デバイスモード

強制ビットをセットした後、アプリケーションは、変更が有効になるまで少なくとも 25 ms 待つ必要があります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 29 **FHMOD** : 強制ホストモード

このビットに 1 を書き込むと、コアは、OTG_FS_ID 入力ピンに関係なく、強制的にホストモードになります。

0 : 通常モード

1 : 強制ホストモード

強制ビットをセットした後、アプリケーションは、変更が有効になるまで少なくとも 25 ms 待つ必要があります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 28:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:10 **TRDT** : USB ターンアラウンド時間

ターンアラウンド時間を PHY クロックで設定します。これらは、表 131: TRDT 値によってアプリケーション AHB 周波数により設定する必要があります。TRDT 値が高ければ、IN トークンへの USB 応答時間を伸ばして、データ FIFO へのより長い AHB 読出しアクセス遅延を補正できます。

注 : デバイスモードでのみアクセス可能です。

ビット 9 **HNPCAP** : HNP 機能

アプリケーションは、このビットを使用して、OTG_FS コントローラの HNP 機能を制御します。

0 : HNP 機能は無効です。

1 : HNP 機能は有効です。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 8 **SRPCAP** : SRP 機能

アプリケーションは、このビットを使用して、OTG_FS コントローラの SRP 機能を制御します。コアが非 SRP 対応 B デバイスとして動作する場合、デバイスは、接続された A デバイス (ホスト) に V_{BUS} を起動し、セッションを開始させるように要求することはできません。

0 : SRP 機能は無効です。

1 : SRP 機能は有効です。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **PHYSEL** : フルスピードシリアルトランシーバの選択

このビットは常に 1 であり、アクセスは書き込み専用です。

ビット 5:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **TOTAL** : FS タイムアウトの較正

アプリケーションがこのフィールドでプログラムする PHY クロックの数は、PHY によって導入される追加遅延を考慮して、コアのフルスピードパケット間タイムアウト時間に追加されます。ライン状態条件の生成時に PHY によって導入される遅延は PHY ごとに異なる場合があるので、較正が必要なことがあります。

フルスピード動作での USB の標準タイムアウト値は 16~18 ビット時間です。アプリケーションは、エニューメレーションのスピードに基づいて、このフィールドをプログラムする必要があります。PHY クロックごとに追加されるビット時間の数は 0.25 ビット時間です。

表 131. TRDT 値

AHB 周波数範囲 (MHz)		TRDT 最小値
最小値	最大値	
14.2	15	0xF
15	16	0xE
16	17.2	0xD
17.2	18.5	0xC
18.5	20	0xB
20	21.8	0xA
21.8	24	0x9
24	27.5	0x8
27.5	32	0x7
32	-	0x6

OTG_FS リセットレジスタ (OTG_FS_GRSTCTL)

アドレスオフセット : 0x10

リセット値 : 0x2000 0000

アプリケーションは、このレジスタを使用して、コア内のさまざまなハードウェア機能をリセットします。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHBIDL	予約済み																				TXFNUM		TXFFLSH	RXFFLSH	予約済み	FCRST	HSRST	CSRST			
r																					rw		rs	rs		rs	rs	rs			

ビット 31 **AHBIDL** : AHB マスタアイドル

AHB マスタステートマシンがアイドル状態にあることを示します。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 30:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:6 **TXFNUM** : TxFIFO 番号

これは TxFIFO フラッシュビットを使用してフラッシュしなければならない FIFO 番号です。コアが TxFIFO フラッシュビットをクリアするまでは、このフィールドを変更しないでください。

00000 :

- ホストモードでの非周期的 TxFIFO フラッシュ
- デバイスモードでの Tx FIFO 0 の一掃

00001 :

- ホストモードでの周期的 TxFIFO フラッシュ
- デバイスモードでのTXFIFO 1フラッシュ

00010 : デバイスモードでのTXFIFO 2フラッシュ

...

00101 : デバイスモードでのTXFIFO 15フラッシュ

10000 : デバイスモードまたはホストモードのすべての送信 FIFO をフラッシュします。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 5 **TXFFLSH** : TxFIFO フラッシュ

このビットは、単一またはすべての送信 FIFO を選択的にフラッシュしますが、コアがトランザクション中の場合はフラッシュできません。

アプリケーションは、コアが TxFIFO への書き込み中か、TxFIFO からの読出し中でないことを確認した後でなければ、このビットに書き込むことはできません。以下のレジスタを使用して確認してください。

読出し — NAK 有効割込みを使用すると、コアが FIFO から読出しを行っていないことを確認できます。

書き込み — OTG_FS_GRSTCTL の AHBIDL ビットで、コアが FIFO に書き込みを行っていないことを確認できます。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 4 **RXFFLSH** : RxFIFO フラッシュ

アプリケーションは、このビットを使用して RxFIFO 全体をフラッシュできますが、まず、コアがトランザクション中でないことを確認する必要があります。

アプリケーションは、コアが RxFIFO からの読出し中、または RxFIFO への書き込み中でないことを確認した後でなければ、このビットに書き込むことはできません。

アプリケーションは、他の操作を実行する前に、このビットがクリアされるまで待つ必要があります。このビットがクリアされるまで、8 クロック (PHY または AHB クロックの中で最も低速なクロック) かかります。

注 : デバイスモードとホストモードの両方でアクセス可能です。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **FCRST** : ホストフレームカウンタリセット

アプリケーションは、このビットに書き込むことによって、コア内のフレーム数カウンタをリセットします。フレームカウンタがリセットされると、コアによって次に送出される SOF のフレーム番号は 0 になります。

注： ホストモードでのみアクセス可能です。

ビット 1 **HSRST** : HCLK ソフトリセット

アプリケーションは、このビットを使用して AHB クロックドメイン内の制御ロジックをフラッシュします。AHB クロックドメインパイプラインだけがリセットされます。

このビットでは FIFO はフラッシュされません。

AHB クロックドメイン内のすべてのステートマシンは、プロトコルに従って AHB 上のトランザクションを終了した後、アイドル状態にリセットされます。

AHB クロックドメインステートマシンによって使用される CSR 制御ビットがクリアされます。

この割込みをクリアするために、割込みステータスを制御して、AHB クロックドメインステートマシンによって生成されるステータスマスクビットがクリアされます。

割込みステータスビットはクリアされないで、アプリケーションは、このビットをセットした後に発生したコアイベントのステータスを取得できます。

これはセルフクリアビットであり、コア内のすべての必要なロジックがリセットされた後、コアがクリアします。これには、コアの現在の状態によって、数クロックサイクルかかります。

注： デバイスモードとホストモードの両方でアクセス可能です。

ビット 0 **CSRST** : コアソフトリセット

以下のように HCLK および PCLK ドメインをリセットします。

割込みと、以下のビットを除くすべての CSR レジスタビットをクリアします。

- OTG_FS_PCGCCTL の RSTPDMODL ビット
- OTG_FS_PCGCCTL の GAYEHCLK ビット
- OTG_FS_PCGCCTL の PWRCLMP ビット
- OTG_FS_PCGCCTL の STPPCLK ビット
- OTG_FS_HCFG の FSLSPCS ビット
- OTG_FS_DCFG の DSPD ビット

すべてのモジュールのステートマシン (AHB スレーブユニットを除く) がアイドル状態にリセットされ、すべての送信 FIFO および受信 FIFO がフラッシュされます。

AHB 転送の最後のデータフェーズの完了後 AHB マスタ上のすべてのトランザクションは速やかに終了されます。USB 上のすべてのトランザクションをただちに終了します。

アプリケーションは、いつでもこのビットに書き込むことによって、コアをリセットできます。これはセルフクリアビットであり、コア内のすべての必要なロジックがリセットされた後、コアがこのビットをクリアします。コアの現在の状態によって、クリアには数クロックかかります。このビットがクリアされると、ソフトウェアは PHY ドメインにアクセスできるようになるまで少なくとも 3 PHY クロック待つ必要があります (同期遅延)。また、ソフトウェアは、操作を開始する前に、このレジスタのビット 31 が 1 にセットされていること (AHB マスタがアイドル状態) を確認する必要があります。

一般に、ソフトウェアリセットは、ソフトウェア開発時に使用され、上記の USB 設定レジスタの PHY 選択ビットを動的に変更するときにも使用されます。PHY を変更するときには PHY の対応クロックが選択され PHY ドメインで使用されます。新しいクロックが選択されたら、適切な動作のためには PHY ドメインをリセットする必要があります。

注： デバイスモードとホストモードの両方でアクセス可能です。

OTG_FS コア割込みレジスタ (OTG_FS_GINTSTS)

アドレスオフセット : 0x014

リセット値 : 0x0400 0020

このレジスタは、現在のモード（デバイスモードまたはホストモード）において、システムレベルのイベントのためにアプリケーションに割込みをかけます。

このレジスタには、ホストモードでのみ有効なビットと、デバイスモードでのみ有効なビットがあります。このレジスタは、現在のモードも示します。rc_w1 タイプの割込みステータスビットをクリアするには、アプリケーションは、そのビットに 1 を書き込む必要があります。

FIFO ステータス割込みは読み出し専用です。これらの割込みの処理中に、ソフトウェアが FIFO への書き込みまたは FIFO からの読み出しを行うと、FIFO 割込み条件は自動的にクリアされます。

初期化の前に割込みが生成されるのを避けるためには、アプリケーションは、割込みビットのマスクを解除する前に、OTG_FS_GINTSTS レジスタをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
WKUINT	SRQINT	DISCINT	CIDSCHG	予約済み	PTXFE	HCINT	HPRINT	予約済み	IPXFER/INCOMPISOOUT	IISOXFR	OEPINT	IEPINT	予約済み	EOPF	ISOODRP	ENUMDNE	USBRST	USBSUSP	ESUSP	予約済み	GOUTNAKEFF	GINAKEFF	NPTXFE	RXFLVL	SOF	OTGINT	MMIS	CMOD					
rc_w1					r	r	r		Res.	rc_w1	r	r		rc_w1				r	r		r	r	rc_w1	r	rc_w1	r							

- ビット 31 **WKUPINT** : レジューム／リモートウェイクアップ検出割込み
デバイスモードでは、この割込みは、USB 上でレジュームが検出されたときにアサートされます。ホストモードでは、この割込みは USB 上でリモートウェイクアップが検出されたときにアサートされます。
注： デバイスモードとホストモードの両方でアクセス可能です。
- ビット 30 **SRQINT** : セッションリクエスト／新規セッション検出割込み
ホストモードでは、この割込みは、デバイスからのセッションリクエストが検出されたときにアサートされます。デバイスモードでは、この割込みは、V_{BUS} が B ペリフェラルデバイスに対して有効な範囲にあるときにアサートされます。デバイスモードとホストモードの両方でアクセス可能です。
- ビット 29 **DISCINT** : 切断検出割込み
デバイスの切断が検出されたときにアサートされます。
注： ホストモードでのみアクセス可能です。
- ビット 28 **CIDSCHG** : コネクタ ID ステータス変化
コアは コネクタ ID ステータスに変化があったとき、このビットをセットします。
注： デバイスモードとホストモードの両方でアクセス可能です。
- ビット 27 予約済みであり、リセット値に保持する必要があります。
- ビット 26 **PTXFE** : 周期的 Tx FIFO エンプティ
周期的送信 FIFO が半エンプティか完全エンプティであり、周期的リクエストキューに少なくとも 1 つのエントリを書き込むためのスペースがあるときにアサートされます。半エンプティステータスか完全エンプティステータスかは、OTG_FS_GAHBCFG レジスタの Tx FIFO エンプティレベルビット (OTG_FS_GAHBCFG の PTXFELVL ビット) によって判断できます。
注： ホストモードでのみアクセス可能です。

ビット 25 HCINT : ホストチャネル割込み

コアは、コアのチャネルの 1 つで割込みがペンディング中であることを示すために、このビットをセットします (ホストモード)。アプリケーションは、OTG_FS_HAINT レジスタを読み出して、割込みが発生しているチャネルの正確な数を確認した後、対応する OTG_FS_HCINTx レジスタを読み出して、割込みの正確な原因を確認する必要があります。アプリケーションは、OTG_FS_HCINTx レジスタの適切なステータスビットをクリアすることによって、このビットをクリアする必要があります。

注： ホストモードでのみアクセス可能です。

ビット 24 HPRTINT : ホストポート割込み

コアは、ホストモードの OTG_FS コントローラポートの 1 つのポートステータスに変化があったことを示すために、このビットをセットします。アプリケーションは、OTG_FS_HPRT レジスタを読み出して、この割込みの原因となった正確なイベントを確認する必要があります。アプリケーションは、OTG_FS_HPRT レジスタの適切なステータスビットをクリアすることによって、このビットをクリアする必要があります。

注： ホストモードでのみアクセス可能です。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 IPXFR : 不完全な周期的転送

ホストモードでは、コアは、現在のフレームでスケジュールされていて、まだペンディング中の不完全な周期的トランザクションが存在するとき、この割込みビットをセットします。

INCOMPIISOOUT : 不完全アイソクロナス OUT 転送

デバイスモードでは、コアは、現在のフレームで転送が完了していないアイソクロナス OUT エンドポイントが少なくとも 1 つあることを示すために、この割込みをセットします。この割込みは、このレジスタの周期的フレーム割込み終了 (EOPF) ビットとともにアサートされます。

ビット 20 IISOXFR : 不完全アイソクロナス IN 転送

コアは、現在のフレームで転送が完了していないアイソクロナス IN エンドポイントが少なくとも 1 つあることを示すために、この割込みをセットします。この割込みは、このレジスタの周期的フレーム割込み終了 (EOPF) ビットとともにアサートされます。

注： デバイスモードでのみアクセス可能です。

ビット 19 OEPINT : OUT エンドポイント割込み

コアは、コアの OUT エンドポイントの 1 つで割込みがペンディングされていることを示すために、このビットをセットします (デバイスモード)。アプリケーションは、OTG_FS_DAINTE レジスタを読み出して、割込みが発生した OUT エンドポイントの正確な数を確認した後、対応する OTG_FS_DOEPINTx レジスタを読み出して、割込みの正確な原因を確認する必要があります。アプリケーションは、対応する OTG_FS_DOEPINTx レジスタの適切なステータスビットをクリアすることによって、このビットをクリアする必要があります。

注： デバイスモードでのみアクセス可能です。

ビット 18 IEPINT : IN エンドポイント割込み

コアは、コアの IN エンドポイントの 1 つで割込みが保留されていることを示すために、このビットをセットします (デバイスモード)。アプリケーションは、OTG_FS_DAINTE レジスタを読み出して、割込みが発生した IN エンドポイントの正確な数を確認した後、対応する OTG_FS_DIEPINTx レジスタを読み出して、割込みの正確な原因を確認する必要があります。アプリケーションは、対応する OTG_FS_DIEPINTx レジスタの適切なステータスビットをクリアすることによって、このビットをクリアする必要があります。

注： デバイスモードでのみアクセス可能です。

ビット 17:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 EOPF : 周期的フレーム終了割込み

現在のフレームで、OTG_FS_DCFG レジスタの周期的フレームインターバルフィールド (OTG_FS_DCFG の PFIVL ビット) で指定された時間に達したことを示します。

注： デバイスモードでのみアクセス可能です。

ビット 14 ISOODRP : アイソクロナス OUT パケット欠落割込み

コアは、アイソクロナス OUT エンドポイントの最大サイズのパケットに対応できるだけの十分なスペースが RxFIFO になかったために RxFIFO にアイソクロナス OUT パケットを書き込めなかったとき、このビットをセットします。

注： デバイスモードでのみアクセス可能です。

ビット 13 ENUMDNE : エニユメレーション終了

コアは、スピードのエニユメレーションが完了したことを示すために、このビットをセットします。アプリケーションは、OTG_FS_DSTS レジスタを読み出して、エニユメレーションされた速度を取得する必要があります。

注： デバイスモードでのみアクセス可能です。

ビット 12 USBRST : USB リセット

コアは USB 上でリセットが検出されたことを示すために、このビットをセットします。

注： デバイスモードでのみアクセス可能です。

ビット 11 USBSUSP : USB サスペンド

コアは USB 上でサスペンドが検出されたことを示すために、このビットをセットします。コアは、データライン上で 3 ms の間、何のアクティビティもなかった場合、サスペンド状態に入ります。

注： デバイスモードでのみアクセス可能です。

ビット 10 ESUSP : アーリーサスペンド

コアは USB 上で 3 ms 間、アイドル状態が検出されたことを示すために、このビットをセットします。

注： デバイスモードでのみアクセス可能です。

ビット 9:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 GONAKEFF : グローバル OUT NAK 有効

アプリケーションによってセットされた OTG_FS_DCTL レジスタのセットグローバル OUT NAK ビット (OTG_FS_DCTL の SGONAK ビット) がコアで有効になったことを示します。このビットは、OTG_FS_DCTL レジスタのクリアグローバル OUT NAK ビット (OTG_FS_DCTL の CGONAK ビット) に書き込むことによってクリアできます。

注： デバイスモードでのみアクセス可能です。

ビット 6 GINAKEFF : グローバル IN 非周期的 NAK 有効

アプリケーションによってセットされた OTG_FS_DCTL レジスタのセットグローバル 非周期的 INNAK ビット (OTG_FS_DCTL の SGINAK ビット) がコアで有効になったことを示します。すなわち、コアは、アプリケーションによってセットされたグローバル IN NAK ビットをサンプリングしました。このビットは、OTG_FS_DCTL レジスタのクリアグローバル非周期的 IN NAK ビット (OTG_FS_DCTL の CGONAK ビット) をクリアすることによってクリアできます。

この割込みは、必ずしも NAK ハンドシェイクが USB に送出されることを意味しません。STALL ビットが NAK ビットに優先します。

注： デバイスモードでのみアクセス可能です。

ビット 5 NPTXFE : 非周期的 Tx FIFO エンプティ

この割込みは、非周期的 Tx FIFO が半エンプティか完全エンプティであり、非周期的送信リクエストキューに、少なくとも 1 つのエントリを書き込むスペースがあるときにアサートされます。半エンプティ状態か完全エンプティ状態かは、OTG_FS_GAHBCFG レジスタの 非周期的 Tx FIFO エンプティレベルビット (OTG_FS_GAHBCFG の TXFELVL ビット) によって確認できます。

注： ホストモードでのみアクセス可能です。

ビット 4 RXFLVL : Rx FIFO 非エンプティ

RxFIFO から読み出されるべきペンディング中のパケットが少なくとも 1 つあることを示します。

注： ホストモードとデバイスモードの両方でアクセス可能です。

ビット 3 SOF : フレーム開始

ホストモードでは、コアは SOF (FS) またはキープアライブ (LS) が USB で送信されることを示すために、このビットをセットします。アプリケーションは、このビットに 1 を書き込むことによって、割込みをクリアする必要があります。

デバイスモードでは、コアは、USB 上で SOF トークンが受信されたことを示すために、このビットをセットします。アプリケーションは、デバイスステータスレジスタを読み出すことによって、現在のフレーム番号を知ることができます。この割込みは、コアが FS で動作しているときのみ確認できます。

注： ホストモードとデバイスモードの両方でアクセス可能です。

ビット 2 OTGINT : OTG 割込み

コアは、OTG プロトコルイベントが完了したことを示すために、このビットをセットします。アプリケーションは、OTG 割込みステータスレジスタ (OTG_FS_GOTGINT) を読み出して、この割込みの原因となった正確なイベントを確認する必要があります。アプリケーションは、OTG_FS_GOTGINT レジスタの適切なステータスビットをクリアすることによって、このビットをクリアする必要があります。

注： ホストモードとデバイスモードの両方でアクセス可能です。

ビット 1 MMIS : モード不一致割込み

コアは、アプリケーションが以下へのアクセスを試みているときに、このビットをセットします。

- ホストモードレジスタ。コアがデバイスモードで動作しているとき。
- デバイスモードレジスタ。コアがホストモードで動作しているとき。

レジスタへのアクセスは、OKAY 応答によって AHB 上で完了しますが、内部的にはコアによって無視され、コアの動作には影響を与えません。

注： ホストモードとデバイスモードの両方でアクセス可能です。

ビット 0 CMOD : 現在の動作モード

現在の動作モードを示します。

- 0 : デバイスモード
- 1 : ホストモード

注： ホストモードとデバイスモードの両方でアクセス可能です。

OTG_FS 割込みマスクレジスタ (OTG_FS_GINTMSK)

アドレスオフセット : 0x018

リセット値 : 0x0000 0000

このレジスタは、コア割込みレジスタとともに働き、アプリケーションに割込みをかけます。割込みビットがマスクされると、そのビットに関連する割込みは生成されません。ただし、その割込みに対応するコア割込みレジスタ (OTG_FS_GINTSTS) のビットは、セットされたままです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
rw	rw	rw	rw		rw	rw	r			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	
WUIM	SRQIM	DISCINT	CIDSCHGM		予約済み	PTXFEM	HCIM			PRTIM	予約済み	IPXFRM/ISOXFRM	ISOXFRM	OEPIINT		IEPIINT	EPIMISM	予約済み	EOPFM	ISOODRPM	ENUMDNEM			USBRST	USBSUSPM	ESUSPM	予約済み	GONAKEFFM	GINAKEFFM	NPTXFEM	RXLVLIM	

- ビット 31 **WUIM** : レジューム／リモートウェイクアップ検出割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 30 **SRQIM** : セッションリクエスト／新規セッション検出割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 29 **DISCINT** : 切断検出割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **デバイスモードでのみアクセス可能です。**
- ビット 28 **CIDSCHGM** : コネクタ ID ステータス変化マスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **ホストモードとデバイスモードの両方でアクセス可能です。**
- ビット 27 予約済みであり、リセット値に保持する必要があります。
- ビット 26 **PTXFEM** : 周期的 TxFIFO エンプティマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **ホストモードでのみアクセス可能です。**
- ビット 25 **HCIM** : ホストチャネル割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **ホストモードでのみアクセス可能です。**
- ビット 24 **PRTIM** : ホストポート割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。
注 : **ホストモードでのみアクセス可能です。**
- ビット 23:22 予約済みであり、リセット値に保持する必要があります。

- ビット 21 **IPXFRM** : 不完全周期的転送マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **ホストモードでのみアクセス可能です。**
- IISOXFRM** : 不完全アイソクロナス OUT 転送マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 20 **IISOXFRM** : 不完全アイソクロナス IN 転送マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 19 **OEPINT** : OUT エンドポイント割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 18 **IEPINT** : IN エンドポイント割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 17 **EPMISM** : エンドポイント不一致割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **EOPFM** : 周期的フレーム終了割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 14 **ISOODRPM** : アイソクロナス OUT パケット欠落割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 13 **ENUMDNEM** : エニユメレーション終了マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 12 **USBRST** : USB リセットマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**
- ビット 11 **USBSUSPM** : USB サスペンドマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 :** **デバイスモードでのみアクセス可能です。**

- ビット 10 **ESUSPM** : アーリーサスペンドマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードでのみアクセス可能です。**
- ビット 9:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **GONAKEFFM** : グローバル OUT NAK 有効マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードでのみアクセス可能です。**
- ビット 6 **GINAKEFFM** : グローバル非周期的IN NAK 有効マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードでのみアクセス可能です。**
- ビット 5 **NPTXFEM** : 非周期的TxFIFOエンプティマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : ホストモードでのみアクセス可能です。**
- ビット 4 **RXFLVLM** : 受信 FIFO ノットエンプティマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードとホストモードの両方でアクセス可能です。**
- ビット 3 **SOFM** : フレーム開始マスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードとホストモードの両方でアクセス可能です。**
- ビット 2 **OTGINT** : OTG 割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードとホストモードの両方でアクセス可能です。**
- ビット 1 **MMISM** : モード不一致割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- 注 : デバイスモードとホストモードの両方でアクセス可能です。**
- ビット 0 予約済みであり、リセット値に保持する必要があります。

OTG_FS 受信ステータスデバッグ読出し／OTG ステータス読出しおよびポップレジスタ (OTG_FS_GRXSTSR/OTG_FS_GRXSTSP)

読出しアドレスオフセット : 0x01C

ポップアドレスオフセット : 0x020

リセット値 : 0x0000 0000

受信ステータスデバッグ読出しレジスタを読み出すと、受信 FIFO の最上部の内容が返されます。受信ステータス読出しおよびポップレジスタを読み出すと RxFIFO の最上部のデータエントリもポップアウトされます。

受信ステータスの内容は、ホストモードとデバイスモードで異なった解釈をする必要があります。コアは、受信 FIFO がエンプティのときには、受信ステータスのポップ／読出しを無視して、0x0000 0000 の値を返します。アプリケーションは、コア割込みレジスタの受信 FIFO 非エンプティビット (OTG_FS_GINTSTS の RXFLVL ビット) がアサートされたときにのみ、受信ステータス FIFO をポップしなければなりません。

ホストモード

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み											PKTSTS		DPID	BCNT								CHNUM									
											r		r	r								r									

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20:17 **PKTSTS** : パケットステータス

受信パケットのステータスを示します。

0010 : IN データパケットが受信されました。

0011 : IN 転送が完了しました (割込みをトリガします)。

0101 : データトグルエラー (割込みをトリガします)。

0111 : チャネル停止 (割込みをトリガします)。

その他 : 予約済み

ビット 16:15 **DPID** : データ PID

受信パケットのデータ PID を示します。

00 : DATA0

10 : DATA1

01 : データ 2

11 : MDATA

ビット 14:4 **BCNT** : バイトカウント

受信した IN データパケットのバイト数を示します。

ビット 3:0 **CHNUM** : チャネル番号

現在の受信パケットが属するチャネル番号を示します。

デバイスモード

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み							FRMNUM	PKTSTS	DPID	BCNT											EPNUM										
							r	r	r	r											r										

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:21 **FRMNUM** : フレーム番号

これは USB 上でパケットが受信されるフレーム番号の下位 4 ビットです。このフィールドは、アイソクロナス OUT エンドポイントがサポートされる場合にのみサポートされます。

ビット 20:17 **PKTSTS** : パケットステータス

受信パケットのステータスを示します。

0001 : グローバル OUT NAK (割込みをトリガします)。

0010 : OUT データパケットが受信されました。

0011 : OUT 転送が完了しました (割込みをトリガします)。

0100 : SETUP トランザクションが完了しました (割込みをトリガします)。

0110 : SETUP データパケットが受信されました。

その他 : 予約済み

ビット 16:15 **DPID** : データ PID

受信した OUT データパケットのデータ PID を示します。

00 : DATA0

10 : DATA1

01 : データ 2

11 : MDATA

ビット 14:4 **BCNT** : バイトカウント

受信したデータパケットのバイト数を示します。

ビット 3:0 **EPNUM** : エンドポイント番号

現在の受信パケットが属するエンドポイント番号を示します。

OTG_FS 受信 FIFO サイズレジスタ (OTG_FS_GRXFSIZ)

アドレスオフセット : 0x024

リセット値 : 0x0000 0200

アプリケーションは RxFIFO に割り当てられなければならない RAM サイズをプログラムできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																RXFD															
																r/rw															

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **RXFD** : RxFIFO の深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

最大値は 256 です。

このレジスタのパワーオンリセット値は Rx データ FIFO の最大の深さで指定されます。

OTG_FS ホスト非周期的送信 FIFO サイズレジスタ (OTG_FS_HNPTXFSIZ) /
エンドポイント 0 送信 FIFO サイズ (OTG_FS_DIEPTXF0)

アドレスオフセット : 0x028
リセット値 : 0x0000 0200

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
NPTXFD/TX0FD																NPTXFSA/TX0FSA															
r/rw																r/rw															

ホストモード

ビット 31:16 **NPTXFD** : 非周期的TxFIFO 深さ
この値は 32 ビットワード単位です。
最小値は 16 です。
最大値は 256 です。

ビット 15:0 **NPTXFSA** : 非周期的送信 RAM 開始アドレス
このフィールドは、非周期的送信 FIFO RAM のメモリ開始アドレスを含みます。

デバイスモード

ビット 31:16 **TX0FD** : エンドポイント 0 TxFIFO の深さ
この値は 32 ビットワード単位です。
最小値は 16 です。
最大値は 256 です。

ビット 15:0 **TX0FSA** : エンドポイント 0 送信 RAM 開始アドレス
このフィールドは、エンドポイント 0 送信 FIFO RAM のメモリ開始アドレスを含みます。

OTG_FS 非周期的送信 FIFO／キューステータスレジスタ
(OTG_FS_HNPTXSTS)

アドレスオフセット : 0x02C
リセット値 : 0x0008 0200

注： デバイスモードでは、このレジスタは無効です。

この読出し専用レジスタは、非周期的 TxFIFO および非周期的送信リクエストキューの空きスペース情報を含みます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	NPTXQTOP								NPTQXSAV								NPTXFSAV														
	r								r								r														

ビット 31 予約済みであり、リセット値に保持する必要があります。

- ビット 30:24 **NPTXQTOP** : 非周期的送信リクエストキューの先頭
- MAC によって処理中の非周期的 Tx リクエストキューのエントリ。
- ビット 30:27チャンネル／エンドポイント番号
- ビット 26:25
- 00 : IN/OUT トークン
 - 01 : ゼロ長送信パケット (デバイス IN／ホスト OUT)
 - 11 : チャンネル停止コマンド
- ビット 24 : 終了 (選択されたチャンネル／エンドポイントの最後のエントリ)
- ビット 23:16 **NPTQXSAV** : 非周期的送信リクエストキューの使用可能スペース
- 非周期的送信リクエストキューの使用可能な空きスペースの量を示します。このキューは、ホストモードでの IN および OUT リクエストを保持します。デバイスモードは IN リクエストのみです。
- 00 : 非周期的送信リクエストキューはフルです。
- 01 : 1 位置が使用可能です。
- 10 : 2 位置が使用可能です。
- bxn : n ロケーションが使用可能です ($0 \leq n \leq 8$)。
- その他 : 予約済み
- ビット 15:0 **NPTXFSAV** : 使用可能な非周期的TxFIFO スペース
- 非周期的 TxFIFO 内で使用できる空きスペースの量を示します。
- 値は 32 ビットワード単位です。
- 00 : 非周期的 Tx FIFO はフルです。
- 01 : 1 ワードが使用可能です。
- 10 : 2 ワードが使用可能です。
- $0xn$: n ワードが使用可能です ($0 \leq n \leq 256$)
- その他 : 予約済み

OTG_FS 一般コア設定レジスタ (OTG_FS_GCCFG)

アドレスオフセット : 0x038

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み										NOVBUSSENS	SOFOUTEN	VBUSBSEN	VBUSASEN	予約済み	PWRDWN	予約済み															
										rw	rw	rw	rw		rw																

ビット 31:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **NOVBUSSENS** : V_{BUS} 検出無効オプション

このビットがセットされると V_{BUS} は内部で常に V_{BUS} の有効レベル (5 V) にあると見なされます。このオプションは専用 V_{BUS} パッドを不要にし、このパッドを共用機能のような他の用途に開放します。 V_{BUS} 接続は別の汎用入力パッドに再割り当てされ、ソフトウェアでモニタされます。
このオプションは、ホスト専用またはデバイス専用アプリケーションにのみ適しています。
0 : ハードウェアによる V_{BUS} 検出が使用できます。
1 : ハードウェアによる V_{BUS} 検出が使用できます。

ビット 20 **SOFOUTEN** : SOF 出カインーブル

0 : SOF パルスは PAD 上で使用できません。
1 : SOF パルスは PAD 上で使用できます。

ビット 19 **VBUSBSEN** : V_{BUS} による「B」デバイスの検出を有効にします。

0 : V_{BUS} による「B」の検出は無効です。
1 : V_{BUS} による「B」の検出は有効です。

ビット 18 **VBUSASEN** : V_{BUS} による「A」デバイスの検出を有効にします。

0 : V_{BUS} による「A」の検出は無効です。
1 : V_{BUS} による「A」の検出は有効です。

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 **PWRDWN** : パワーダウン

送信／受信のためにトランシーバを起動するために使用されます。
0 : パワーダウンアクティブ
1 : パワーダウン非アクティブ (「トランシーバアクティブ」)

ビット 15:0 予約済みであり、リセット値に保持する必要があります。

OTG_FS コア ID レジスタ (OTG_FS_CID)

アドレスオフセット : 0x03C

リセット値 : 0x0000 1100

これは、製品 ID を含む読出し専用レジスタです。

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

ビット 31:0 **PRODUCT_ID** : 製品 ID フィールド

アプリケーションによってプログラム可能な ID フィールド。

OTG_FS ホスト周期的送信 FIFO サイズレジスタ (OTG_FS_HPTXFSIZ)

アドレスオフセット : 0x100

リセット値 : 0x0200 0600

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
PTXFSIZ																PTXSA															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **PTXFD** : ホスト周期的 TxFIFO の深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

ビット 15:0 **PTXSA** : ホスト周期的 TxFIFO の開始アドレス

このレジスタのパワーオンリセット値は、Rx データ FIFO の最大深さと Tx データ FIFO の最大深さの合計です。

OTG_FS デバイス IN エンドポイント送信 FIFO サイズレジスタ (OTG_FS_DIEPTXF_x) (x = 1..3、x は FIFO_{number})

アドレスオフセット : 0x104 + (FIFO_{number} - 1) * 0x04

リセット値 : 0x02000400

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
INEPTXFD																INEPTXSA															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 **INEPTXFD** : IN エンドポイント TxFIFO の深さ

この値は 32 ビットワード単位です。

最小値は 16 です。

このレジスタのパワーオンリセット値は、IN エンドポイント FIFO 番号の最大深さとして指定されます。

ビット 15:0 **INEPTXSA** : IN エンドポイント FIFO_x 送信 RAM 開始アドレス

このフィールドは IN エンドポイント送信 FIFO_x のメモリ開始アドレスを含みます。アドレスは 32 ビットメモリロケーションに整列されている必要があります。

22.16.3 ホストモードレジスタ

レジスタの説明で示されているビット値は、特に記載がない限りバイナリで表されています。

ホストモードレジスタは、ホストモードでのコアの動作に影響を与えます。ホストモードレジスタは、結果が定義されていないため、デバイスモードではアクセスしないでください。ホストモードレジスタは、次のように分類することができます。

OTG_FS ホスト設定レジスタ (OTG_FS_HCFG)

アドレスオフセット : 0x400

リセット値 : 0x0000 0000

このレジスタは、電源投入後のコアを設定します。ホストを初期化した後は、このレジスタを変更しないでください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
予約済み																											FSLSS		FSLSPCS								
																											r	rw			rw						

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **FSLSS** : FS および LS のみのサポート

アプリケーションは、このビットを使用して、コアのエnumレーション速度を制御します。このビットを使用して、アプリケーションは、接続されたデバイスが HS トラフィックをサポートする場合でも、FS ホストとしてコアにエnumレーションさせることができます。初期プログラミングの後は、このフィールドを変更しないでください。

1 : 接続されたデバイスが HS トラフィックをサポートしている場合でも、FS/LS のみ (読み出し専用)。

ビット 1:0 **FSLSPCS** : FS/LS PHY クロック選択

コアが FS ホストモードのとき、
01 : PHY クロックは 48 MHz で動作しています。
その他 : 予約済み
コアが LS ホストモードのとき、
00 : 予約済み
01 : PHY クロック周波数として 48 MHz を選択
10 : PHY クロック周波数として 6 MHz を選択
11 : 予約済み

注 : **FSLSPCS** は、接続されたデバイスの速度に応じて、接続イベント時にセットする必要があります (このビットを変更した後、ソフトウェアリセットを実行する必要があります)。

OTG_FS ホストフレームインターバルレジスタ (OTG_FS_HFIR)

アドレスオフセット : 0x404

リセット値 : 0x0000 EA60

このレジスタは、OTG_FS コントローラがエニュメレーションした現在の速度でのフレームインターバル情報を格納します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																FRIVL															
																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 **FRIVL** : フレームインターバル

アプリケーションがこのフィールドにプログラムする値は、2 つの連続した SOF (FS) またはキープアライブトークン (LS) の間のインターバルを指定します。このフィールドは、必要なフレームインターバルを構成する PHY クロック数を含みます。アプリケーションは、ホストポート制御およびステータスレジスタのポートイネーブルビット (OTG_FS_HPRT の PENA ビット) がセットされた後のみ、このレジスタに値を書き込むことができます。値がプログラムされていない場合、コアは、ホスト設定レジスタの FS/LS PHY クロック選択フィールド (OTG_FS_HCFG の FSLSPCS) で指定された PHY クロックに基づいて値を計算します。初期プログラミングの後には、このフィールドの値を変更しないでください。

$1 \text{ ms} \times (\text{PHY クロック周波数})$

OTG_FS ホストフレーム番号/残りフレーム時間レジスタ (OTG_FS_HFNUM)

アドレスオフセット : 0x408

リセット値 : 0x0000 3FFF

このレジスタは、現在のフレーム番号を示します。また、現在のフレームの残り時間 (PHY クロック数) も示します。

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

ビット 31:16 **FTREM** : 残りフレーム時間

現在のフレームの残り時間を PHY クロック数で示します。このフィールドは PHY クロックごとにデクリメントされます。値がゼロになると、このフィールドにはフレームインターバルレジスタの値が再ロードされ、新しい SOF が USB に送信されます。

ビット 15:0 **FRNUM** : フレーム番号

このフィールドは USB 上で新しい SOF が送信されるたびにインクリメントされ、0x3FFF に達すると 0 にクリアされます。

OTG_FS_ホスト周期的送信 FIFO／キューステータスレジスタ (OTG_FS_HPTXSTS)

アドレスオフセット : 0x410

リセット値 : 0x0008 0100

この読み出し専用レジスタは、周期的 TxFIFO および周期的送信リクエストキューの空きスペース情報を含みます。

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

ビット 31:24 **PTXQTOP** : 周期的送信リクエストキューの先頭

これは MAC によって処理中の周期的 Tx リクエストキューのエントリを示します。

このレジスタは、デバッグに使用されます。

ビット 31 : 奇数／偶数フレーム

- 0 : 偶数フレームで送信
- 1 : 奇数フレームで送信

ビット 30:27 チャンネル／エンドポイント番号

ビット 26:25 タイプ

- 00 : IN/OUT
- 01 : 長さゼロの packets
- 11 : チャンネルコマンド無効化

ビット 24 : 終了 (選択されたチャンネル／エンドポイントの最後のエントリ)

ビット 23:16 **PTXQSAV** : 周期的送信リクエストキューの使用可能スペース

周期的送信リクエストキューで書き込みに使用できる空きロケーションの数を示します。このキューは IN および OUT リクエストの両方を保持します。

00 : 周期的送信リクエストキューはフルです。

01 : 1 位置が使用可能です。

10 : 2 位置が使用可能です。

bxn : n 位置が使用可能です (0 ≤ n ≤ 8)

その他 : 予約済み

ビット 15:0 **PTXFSAVL** : 周期的送信データ FIFO の使用可能スペース

周期的 TxFIFO で書き込みに使用できる空きロケーションの数を示します。

値は 32 ビットワード単位です。

0000 : 周期的 Tx FIFO はフルです。

0001 : 1 ワードが使用可能です。

0010 : 2 ワードが使用可能です。

bxn : n ワードが使用可能です (0 ≤ n ≤ PTXFD)

その他 : 予約済み

OTG_FS ホスト全チャンネル割込みレジスタ (OTG_FS_HAINT)

アドレスオフセット : 0x414

リセット値 : 0x0000 000

チャンネル上で重大なイベントが発生すると、ホスト全チャンネル割込みレジスタは、コア割込みレジスタのホストチャンネル割込みビット (OTG_FS_GINTSTS の HCINT ビット) を使用して、アプリケーションに割込みをかけます。これを [図 250](#) に示します。チャンネルごとに 1 つの割込みビットがあります (最大 16 ビット)。このレジスタのビットは、アプリケーションが対応するホストチャンネル x 割込みレジスタのビットをセット/クリアしたときにセット/クリアされます。

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

- ビット 31:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15:0 **HAINT** : チャンネル割込み
- チャンネルあたり 1 ビット: チャンネル 0 に対してビット 0、チャンネル 15 に対してビット 15

OTG_FS ホスト全チャンネル割込みマスクレジスタ (OTG_FS_HAINTMSK)

アドレスオフセット : 0x418

リセット値 : 0x0000 0000

ホスト全チャンネル割込みマスクレジスタは、ホスト全チャンネル割込みレジスタとともに働き、チャンネル上でイベントが発生したときに、アプリケーションに割込みをかけます。チャンネルごとに 1 つの割込みマスクビットがあります (最大 16 ビット)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																HAINTM															
																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 **HAINTM** : チャンネル割込みマスク
- 0 : 割込みはマスクされます。
- 1 : 割込みはマスクされません。
- チャンネルあたり 1 ビット: チャンネル 0 に対してビット 0、チャンネル 15 に対してビット 15

OTG_FS ホストポート制御およびステータスレジスタ (OTG_FS_HPRT)

アドレスオフセット : 0x440

リセット値 : 0x0000 0000

このレジスタは、ホストモードでのみ使用可能です。現時点では OTG ホストは 1 つのポートのみをサポートしています。

1 つのレジスタは、ポートごとの USB リセット、イネーブル、サスペンド、レジューム、接続ステータス、テストモードなどの USB ポート関連情報を保持します。図 250 に示されています。このレジスタの rc_w1 ビットは、コア割込みレジスタのホストポート割込みビット (OTG_FS_GINTSTS の HPRTINT ビット) を通じて、アプリケーションへの割込みをトリガできます。ポート割込みの場合、アプリケーションは、このレジスタを読み出して、割込みの原因となったビットをクリアする必要があります。rc_w1 ビットの場合、アプリケーションは、このビットに 1 を書き込んで、割込みをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み													PSPD		PTCTL				PPWR	PLSTS		予約済み	PRST	PSUSP	PRES	POCHNG	POCA	PENCHNG	PENA	PCDET	PCSTS
													r	r	rw	rw	rw	rw	rw	r	r		rw	rs	rw	rc_w1	r	rc_w1	rc_w0	rc_w1	r

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:17 **PSPD** : ポートスピード

このポートに接続されたデバイスのスピードを示します。

01 : フルスピード

10 : ローススピード

11 : 予約済み

ビット 16:13 **PTCTL** : ポートテスト制御

アプリケーションがこのフィールドにゼロ以外の値を書き込んで、ポートをテストモードにすると、対応するパターンがポートにシグナリングされます。

0000 : テストモードは無効です

0001 : Test_J モード

0010 : Test_K モード

0011 : Test_SE0_NAK モード

0100 : Test_Packet モード

0101 : Test_Force_Enable

その他 : 予約済み

ビット 12 **PPWR** : ポートパワー

アプリケーションは、このフィールドを使用して、このポートへの電力供給を制御し、コアは、過電流条件が発生すると、このビットをクリアします。

0 : パワーオフ

1 : パワーオン

ビット 11:10 **PLSTS** : ポートラインステータス

USB データラインの現在のロジックレベルを示します。

ビット 10 : OTG_FS_FS_DP のロジックレベル

ビット 11 : OTG_FS_FS_DM のロジックレベル

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8 PRST : ポートリセット

アプリケーションがこのビットをセットすると、このポートでリセットシーケンスが開始されます。アプリケーションは、リセットシーケンスの完了後、リセット時間を決めて、このビットをクリアする必要があります。

0 : ポートは非リセット中

1 : ポートはリセット中

アプリケーションは、ポートのリセットが開始されるまで、少なくとも 10 ms の間、このビットをセット状態に保つ必要があります。USB 規格では最大限度が規定されていませんが、アプリケーションは、このビットをクリアする前に、必要最小時間に加えて、さらに 10 ms の間、このビットをセット状態に保つことができます。

ビット 7 PSUSP : ポートサスペンド

アプリケーションは、このビットをセットして、このポートをサスペンドモードにします。コアは、このビットがセットされると SOF の送信のみを停止します。PHY クロックを停止するには、アプリケーションは PHY のサスペンド入力ピンをアサートするポートクロック停止ビットをセットする必要があります。

このビットの読出し値は、ポートの現在のサスペンドステータスを反映しています。このビットは、リモートウェイクアップ信号の検出後、またはアプリケーションがこのレジスタのポートリセットビットまたはポートレジュームビット、またはコア割込みレジスタのレジューム／リモートウェイクアップ検出割込みビットまたは切断検出割込みビット（それぞれ、OTG_FS_GINTSTS の WKUINT または DISCINT ビット）をセットしたときに、コアによってクリアされます。

0 : ポートは非サスペンドモード

1 : ポートはサスペンドモード

ビット 6 PRES : ポートレジューム

アプリケーションは、このビットをセットして、ポート上でレジューム信号を駆動します。コアは、アプリケーションがこのビットをクリアするまで、レジューム信号の駆動を続行します。

コアが、コア割込みレジスタのポートレジューム／リモートウェイクアップ検出割込みビット (OTG_FS_GINTSTS の WKUINT ビット) によって示される USB リモートウェイクアップシーケンスを検出した場合、コアは、切断条件を検出すると、アプリケーションの介入なしでレジューム信号の駆動を開始、このビットをクリアします。このビットの読出し値は、コアが現在レジューム信号を駆動中であるかどうかを示します。

0 : レジューム信号を駆動していません。

1 : レジューム信号を駆動しています。

ビット 5 POCCHNG : ポート過電流の変化

コアは、このレジスタのポート過電流アクティブビット (ビット 4) に変化があったとき、このビットをセットします。

ビット 4 POCA : ポート過電流アクティブ

ポートの過電流条件を示します。

0 : 非過電流状態

1 : 過電流状態

ビット 3 PENCHNG : ポートイネーブル／ディセーブルの変化

コアは、このレジスタのポートイネーブルビット 2 のステータスに変化があったとき、このビットをセットします。

ビット 2 PENA : ポートイネーブル

ポートは、リセットシーケンス後、コアによってのみ有効にされ、過電流条件、切断条件、またはアプリケーションがこのビットをクリアすることによって無効にされます。アプリケーションは、レジスタへの書込みによってこのビットをクリアすることはできません。クリアして、ポートを無効にすることだけができます。このビットは、アプリケーションへの割込みをトリガしません。

0 : ポートを無効化

1 : ポートを有効化

ビット 1 PCDET : ポート接続の検出

コアは、デバイス接続が検出されると、このビットをセットして、コア割込みレジスタのホストポート割込みビット (OTG_FS_GINTSTS の HPRTINT ビット) を使用して、アプリケーションへの割込みをトリガします。アプリケーションは、このビットに 1 を書き込むことによって、割込みをクリアする必要があります。

ビット 0 PCSTS : ポート接続ステータス

0 : ポートにはデバイスが接続されていません。
1 : ポートにデバイスが接続されています。

OTG_FS ホストチャネル x 特性レジスタ (OTG_FS_HCCHARx) (x = 0..7、x = Channel_number)

アドレスオフセット : 0x500 + (Channel_number * 0x20)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ											
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 CHENA : チャネルイネーブル

このフィールドは、アプリケーションによってセットされ OTG ホストによってクリアされます。
0 : チャネルは無効です。
1 : チャネルは有効です。

ビット 30 CHDIS : チャネル無効化

アプリケーションは、チャネルへの転送が完了する前でも、このビットをセットして、そのチャネル上での送信/受信を停止することができます。アプリケーションは、チャネルを無効として扱う前に、チャネル無効割込みを待つ必要があります。

ビット 29 ODDFRM : 奇数フレーム

このフィールドは OTG ホストが奇数フレームで転送を実行しなければならないことを示すために、アプリケーションによってセット (リセット) されます。このフィールドは、周期的 (アイソクロナスおよび割込み) トランザクションに対してのみ適用されます。
0 : 偶数フレーム
1 : 奇数フレーム

ビット 28:22 DAD : デバイスアドレス

このフィールドは、データソースまたはデータシンクとして機能する特定のデバイスを選択します。

ビット 21:20 MCNT : マルチカウント

このフィールドは、この周期的エンドポイントについてフレームあたりで実行されなければならないトランザクションの数をホストに示します。非周期的転送の場合、このフィールドは使用されません。
00 : 予約済み。このフィールドは定義されていない結果をもたらします。
01 : 1 トランザクション
10 : このエンドポイントに対して発行されるべきトランザクションは、フレームあたり 2 つです。
11 : このエンドポイントに対して発行されるべきトランザクションは、フレームあたり 3 つです。

注 : このフィールドは、01 以上にセットする必要があります。

- ビット 19:18 **EPTYP** : エンドポイントタイプ
選択された転送タイプを示します。
00 : コントロール
01 : アイソクロナス
10 : バルク
11 : インタラプト
- ビット 17 **LSDEV** : ロースピードデバイス
このフィールドは、このチャンネルがロースピードデバイスと通信中であることを示すために、アプリケーションによってセットされます。
- ビット 16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **EPDIR** : エンドポイントの方向
トランザクションが IN か OUT かを示します。
0 : OUT
1 : IN
- ビット 14:11 **EPNUM** : エンドポイント番号
データソースまたはデータシンクとして機能しているデバイスのエンドポイント番号を示します。
- ビット 10:0 **MPSIZ** : 最大パケットサイズ
関連するエンドポイントの最大パケットサイズを示します。

OTG_FS ホストチャンネル x 割込みレジスタ (OTG_FS_HCINTx) (x = 0..7、x = Channel_number)

アドレスオフセット : 0x508 + (Channel_number * 0x20)

リセット値 : 0x0000 0000

このレジスタは USB および AHB 関連イベントに関するチャンネルのステータスを示します。[図 250](#) に示されています。アプリケーションは、コア割込みレジスタのホストチャンネル割込みビット (OTG_FS_GINTSTS の OEPINT ビット) がセットされたとき、このレジスタを読み出す必要があります。アプリケーションがこのレジスタを読み出すためには、まず、ホスト全チャンネル割込み (OTG_FS_HAINT) レジスタを読み出して、ホストチャンネル x 割込みレジスタにの正確なチャンネル番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_FS_HAINT および OTG_FS_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
予約済み																					DTERR	FRMOR	BBERR	TXERR	予約済み		ACK	NAK	STALL	予約済み		CHH	XFRQ
																					rc_w1	rc_w1	rc_w1	rc_w1	予約済み		rc_w1	rc_w1	rc_w1	予約済み		rc_w1	rc_w1

- ビット 31:11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **DTERR** : データトグルエラー
- ビット 9 **FRMOR** : フレームオーバーラン
- ビット 8 **BBERR** : バブルエラー

- ビット 7 **TXERR** : トランザクションエラー
- USB 上で以下のエラーの 1 つが発生したことを示します。
- CRC チェック失敗
 - タイムアウト
 - ビットスタッフエラー
 - 偽の EOP
- ビット 6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **ACK** : ACK レスポンス受信／送信割込み
- ビット 4 **NAK** : NAK レスポンス受信割込み
- ビット 3 **STALL** : STALL レスポンス受信割込み
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **CHH** : チャネル停止
- USB トランザクションエラーまたはアプリケーションによるディセーブルリクエストへの応答のいずれかの理由により、転送が異常終了したことを示します。
- ビット 0 **XFRC** : 転送完了
- 転送がエラーなく正常に完了しました。

OTG_FS ホストチャネル x 割込みマスクレジスタ (OTG_FS_HCINTMSKx) (x = 0..7、x = Channel_number)

アドレスオフセット : 0x50C + (Channel_number * 0x20)

リセット値 : 0x0000 0000

このレジスタは、前のセクションで説明した各チャネルステータスのマスクを反映します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																					DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRCM
																					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 **DTERRM** : データトグルエラーマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- ビット 9 **FRMORM** : フレームオーバーランマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- ビット 8 **BBERRM** : バブルエラーマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- ビット 7 **TXERRM** : トランザクションエラーマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。
- ビット 6 **NYET** : 応答受信割込みマスク
- 0 : 割込みはマスクされます。
 - 1 : 割込みはマスクされません。

ビット 5 **ACKM** : ACK レスポンス受信／送信割込みマスク

- 0 : 割込みはマスクされます。
- 1 : 割込みはマスクされません。

ビット 4 **NAKM** : NAK レスポンス受信割込みマスク

- 0 : 割込みはマスクされます。
- 1 : 割込みはマスクされません。

ビット 3 **STALLM** : STALL 応答受信割込みマスク

- 0 : 割込みはマスクされます。
- 1 : 割込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CHHM** : チャネル停止マスク

- 0 : 割込みはマスクされます。
- 1 : 割込みはマスクされません。

ビット 0 **XFRM** : 転送完了マスク

- 0 : 割込みはマスクされます。
- 1 : 割込みはマスクされません。

OTG_FS ホストチャネル x 転送サイズレジスタ (OTG_FS_HCTSIZx) (x = 0..7、x = Channel_number)

アドレスオフセット : 0x510 + (Channel_number * 0x20)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	DPID		PKTCNT										XFRSIZ																		
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	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 30:29 **DPID** : データ PID

- アプリケーションは、このフィールドを最初のトランザクションで使用する PID のタイプでプログラムします。ホストは、残りの転送の間、このフィールドを維持します。
- 00 : DATA0
 - 01 : データ 2
 - 10 : DATA1
 - 11 : MDATA (非制御) /SETUP (制御)

ビット 28:19 **PKTCNT** : パケットカウント

- このフィールドは、アプリケーションによって、送信 (OUT) または受信 (IN) されるパケットの予想数でプログラムされます。
- ホストは OUT/IN パケットの送信または受信が成功するたびに、このカウントをデクリメントします。このカウントがゼロに達すると、アプリケーションに割り込んで、正常完了を示します。

ビット 18:0 **XFRSIZ** : 転送サイズ

- OUT の場合、このフィールドは、ホストが転送時に送信するデータバイト数です。
- IN の場合、このフィールドは、アプリケーションが転送のために確保したバッファサイズです。アプリケーションは、IN トランザクション (周期的および非周期的) の場合、このフィールドを最大パケットサイズの整数倍でプログラムすることが期待されます。



22.16.4 デバイスモードレジスタ

OTG_FS デバイス設定レジスタ (OTG_FS_DCFG)

アドレスオフセット : 0x800

リセット値 : 0x0220 0000

このレジスタは、電源投入後、または特定の制御コマンドあるいはエニユメレーションの後、コアをデバイスモードに設定します。初期プログラミングの後には、このレジスタを変更しないでください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																			PFIVL	DAD							予約済み	NZLSOHSK		DSPD	
																												rw	rw	rw	rw
																			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:13 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 **PFIVL** : 周期的フレームインターバル

周期的フレーム終了割込みを使用してアプリケーションに通知しなければならないフレーム内の時間を示します。これを使用して、そのフレームのアイソクロナストラフィックのすべてが完了したかどうかを判断できます。

- 00 : フレームインターバルの 80%
- 01 : フレームインターバルの 85%
- 10 : フレームインターバルの 90%
- 11 : フレームインターバルの 95%

ビット 10:4 **DAD** : デバイスアドレス

アプリケーションは、各 SetAddress 制御コマンドの後、このフィールドをプログラムする必要があります。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **NZLSOHSK** : 非ゼロ長ステータスOUTハンドシェイク

アプリケーションは、このフィールドを使用して、制御転送のステータスステージの OUT トランザクション中に、コアが非ゼロ長のデータパケットを受信したときに送信するハンドシェイクを選択できます。

- 1 : 非ゼロ長ステータスの OUT トランザクション時には STALL ハンドシェイクを送信し、受信した OUT パケットをアプリケーションに送信しません。
- 0 : 受信した OUT パケット（ゼロ長または非ゼロ長）をアプリケーションに送信し、デバイスエンドポイント制御レジスタの NAK および STALL ビットに基づいてハンドシェイクを送信します。

ビット 1:0 **DSPD** : デバイススピード

アプリケーションがコアにエニユメレーションを要求するスピード、またはアプリケーションがサポートできる最高スピードを示します。ただし、実際のバススピードは、チャープシーケンスの完了後に決まり、コアが接続される USB ホストのスピードに基づきます。

- 00 : 予約済み
- 01 : 予約済み
- 10 : 予約済み
- 11 : フルスピード (USB 1.1 トランシーバのクロックは 48 MHz です)

OTG_FS デバイス制御レジスタ (OTG_FS_DCTL)

アドレスオフセット : 0x804

リセット値 : 0x0000 0000

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
予約済み																				POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL			GONSTS	GINSTS	SDIS	RWUSIG
																				RW	W	W	W	W	RW	RW	RW	R	R	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 **POPRGDNE** : パワーオンプログラミング終了

アプリケーションは、このビットを使用して、パワーダウンモードからのウェイクアップ後に、レジスタのプログラミングが終了したことを示します。

ビット 10 **CGONAK** : グローバル OUT NAK のクリア

このフィールドに書き込むと、グローバル OUT NAK がクリアされます。

ビット 9 **SGONAK** : グローバル OUT NAK のセット

このフィールドに書き込むと、グローバル OUT NAK がセットされます。

アプリケーションは、このビットを使用して、すべての OUT エンドポイントに NAK ハンドシェイクを送信します。

アプリケーションは、コア割込みレジスタのグローバル OUT NAK 有効ビット (OTG_FS_GINTSTS の GONAKEFF ビット) がクリアされていることを確認した後でのみ、このビットをセットしなければなりません。

ビット 8 **CGINAK** : グローバル IN NAK のクリア

このフィールドに書き込むと、グローバル IN NAK がクリアされます。

ビット 7 **SGINAK** : グローバル IN NAK のセット

このフィールドに書き込むと、グローバル非周期的 IN NAK がセットされます。アプリケーションは、このビットを使用して、すべての非周期的 IN エンドポイントに NAK ハンドシェイクを送信します。

アプリケーションは、コア割込みレジスタのグローバル IN NAK 有効ビット (OTG_FS_GINTSTS の GINAKEFF ビット) がクリアされていることを確認した後でのみ、このビットをセットしなければなりません。

ビット 6:4 **TCTL** : テスト制御

000 : テストモードは無効です

001 : Test_J モード

010 : Test_K モード

011 : Test_SE0_NAK モード

100 : Test_Packet モード

101 : Test_Force_Enable

その他 : 予約済み

ビット 3 **GONSTS** : グローバル OUT NAK のステータス

0 : ハンドシェイクは、FIFO ステータスと NAK および STALL ビットの設定に基づいて送信されます。

1 : 使用可能なスペースに関係なく RxFIFO にはデータが書き込まれません。SETUP トランザクションを除き、すべてのパケットに NAK ハンドシェイクを送信します。すべてのアイソクロナス OUT パケットがドロップされます。

ビット 2 **GINSTS** : グローバル IN NAK ステータス

- 0 : ハンドシェイクは、送信 FIFO 内の使用可能なデータに基づいて送信されます。
- 1 : NAK ハンドシェイクは、送信 FIFO 内の使用可能なデータに関係なく、非周期的 IN エンドポイントに送出されます。

ビット 1 **SDIS** : ソフト切断

- アプリケーションは、このビットを使用して USB OTG コアにソフト切断を実行することを伝えます。このビットがセットされている限り、ホストはデバイスが接続されているとは解釈せず、デバイスは USB 上で信号を受信しません。コアは、アプリケーションがこのビットをクリアするまで、切断状態にとどまります。
- 0 : 通常動作。ソフト切断後にこのビットがクリアされると、コアは USB ホストに対してデバイス接続イベントを生成します。デバイスが再接続されると、USB ホストはデバイスのエnumレーションをリスタートします。
 - 1 : コアは、USB ホストに対して、デバイス切断イベントを生成します。

ビット 0 **RWUSIG** : リモートウェイクアップ信号

アプリケーションがこのビットをセットすると、コアはリモートシグナリングを実行して USB ホストをウェイクアップします。コアをサスペンド状態から抜け出させるには、アプリケーションは、このビットをセットする必要があります。USB 2.0 仕様で規定されているように、アプリケーションは、このビットをセットしてから 1 ms から 15 ms 後にクリアする必要があります。

表 132 は USB ホストがデバイスの切断を検出するためにソフト切断 (SDIS) ビットがセットされなければならない最短時間 (デバイスの状態による) を示します。クロックジッタに対応するには、アプリケーションは指定された最短時間まで遅延を追加することが推奨されます。

表 132. ソフト切断のための最短時間

動作速度	デバイスの状態	最短時間
フルスピード	サスペンド状態	1 ms + 2.5 μs
フルスピード	アイドル	2.5 μs
フルスピード	アイドル状態でもサスペンド状態でもない (トランザクション実行中)	2.5 μs

OTG_FS デバイスステータスレジスタ (OTG_FS_DSTS)

アドレスオフセット : 0x808

リセット値 : 0x0000 0010

このレジスタは USB 関連イベントに関するコアのステータスを示します。デバイス全割込みレジスタ (OTG_FS_DAIN) からの割込み時に読み出されなければなりません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み										FNSOF										予約済み			EERR	ENUMSPD		SUSPSTS					
										r	r	r	r	r	r	r	r	r	r								r	r	r	r	r

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 21:8 **FNSOF** : 受信 SOF のフレーム番号

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **EERR** : 不規則エラー

コアは、このビットをセットして、不規則エラーを報告します。
不規則エラーが発生すると、OTG_FS コントローラはサスペンド状態に移行し、OTG_FS_GINTSTS レジスタのアーリーサスペンドビット (OTG_FS_GINTSTS の ESUP ビット) でアプリケーションへの割込みが生成されます。不規則エラーによるアーリーサスペンドがアサートされた場合、アプリケーションは、ソフト切断回復を実行するしかありません。

ビット 2:1 **ENUMSPD** : エニユメレーションされた速度

チャープシーケンスによる速度検出後の OTG_FS コントローラの速度を示します。
01 : 予約済み
10 : 予約済み
11 : フルスピード (PHY クロックは 48 MHz で動作しています)
その他 : 予約済み

ビット 0 **SUSPSTS** : サスペンドステータス

デバイスモードでは、このビットは、USB 上でサスペンド状態が検出されている限り、セットされます。USB データライン上で 3 ms の間アクティビティが 1 つもなかった場合、コアはサスペンド状態に入ります。コアは、以下の場合にサスペンド状態から抜け出します。
– USB データライン上で何らかのアクティビティがあったとき。
– アプリケーションが OTG_FS_DCTL レジスタのリモートウェイクアップシグナリングビット (OTG_FS_DCTL の RWUSIG ビット) に書き込みを行ったとき。

OTG_FS デバイス IN エンドポイント共通割込みマスクレジスタ (OTG_FS_DIEPMSK)

アドレスオフセット : 0x810

リセット値 : 0x0000 0000

このレジスタは、すべてのエンドポイントの各 OTG_FS_DIEPINTx レジスタとともに機能して、IN エンドポイントごとに割込みを生成します。このレジスタの対応するビットに書き込むことによって、OTG_FS_DIEPINTx レジスタの特定のステータスに対する IN エンドポイント割込みをマスクできます。ステータスビットはデフォルトでマスクされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																								INENEM	INENMM	ITTXFEMSK	TOM	予約済み	EPDM	XFRDM	
																								r/w	r/w	r/w	r/w		r/w	r/w	

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 **INENEM** : IN エンドポイント NAK 有効マスク

0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 5 **INENMM** : EP 不一致付きで受信された IN トークンマスク

0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 4 **ITTXFEMSK** : TxFIFO エンプティ時に受信された IN トークンマスク

0 : 割込みはマスクされます。
1 : 割込みはマスクされません。



ビット 3 **TOM** : タイムアウト状態マスク (非アイソクロナスエンドポイント)
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDM** : エンドポイントディセーブル割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 0 **XFRM** : 転送完了割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

OTG_FS デバイス OUT エンドポイント共通割込みマスクレジスタ (OTG_FS_DOEPMASK)

アドレスオフセット : 0x814
リセット値 : 0x0000 0000

このレジスタは、すべてのエンドポイントの各 OTG_FS_DOEPINTx レジスタとともに機能して、OUT エンドポイントごとに割込みを生成します。このレジスタの対応するビットに書き込むことによって、OTG_FS_DOEPINTx レジスタの特定のステータスに対する OUT エンドポイント割込みをマスクできます。ステータスビットはデフォルトでマスクされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																										OTEPDM	STUPM	予約済み	EPDM	XFRM	
																										rw	rw		rw	rw	

ビット 31:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OTEPDM** : エンドポイント無効時 OUT トークン受信マスク
制御 OUT エンドポイントにのみ適用されます。
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 3 **STUPM** : SETUP フェーズ終了マスク
コントロールエンドポイントにのみ適用されます。
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **EPDM** : エンドポイントディセーブル割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

ビット 0 **XFRM** : 転送完了割込みマスク
0 : 割込みはマスクされます。
1 : 割込みはマスクされません。

OTG_FS デバイス全エンドポイント割込みレジスタ (OTG_FS_DAINTEP)

アドレスオフセット : 0x818

リセット値 : 0x0000 0000

エンドポイント上で重要なイベントが発生すると、OTG_FS_DAINTEP レジスタは、OTG_FS_GINTSTS レジスタのデバイス OUT エンドポイント割込みビットまたはデバイス IN エンドポイント割込みビット（それぞれ、OTG_FS_GINTSTS の OEPINT または IEPINT ビット）を使用して、アプリケーションに割込みをかけます。エンドポイントごとに 1 つの割込みビットがあります（OUT エンドポイントおよび IN エンドポイントに対してそれぞれ最大 16 ビット）。双方向エンドポイントの場合、対応する IN および OUT 割込みビットが使用されます。このレジスタのビットは、アプリケーションが対応するデバイスエンドポイント x 割込みレジスタのビット (OTG_FS_DIEPINTx/OTG_FS_DOEPINTx) をセット/クリアしたときに、セット/クリアされます。

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

ビット 31:16 **OEPINT** : OUT エンドポイント割込みビット

OUT エンドポイントチャンネルあたり 1 ビット :

OUT エンドポイント 0 に対してビット 16、OUT エンドポイント 3 に対してビット 18

ビット 15:0 **IEPINT** : IN エンドポイント割込みビット

IN エンドポイントあたり 1 ビット :

IN エンドポイント 0 に対してビット 0、IN エンドポイント 3 に対してビット 3

OTG_FS 全エンドポイント割込みマスクレジスタ (OTG_FS_DAINTEPM)

アドレスオフセット : 0x81C

リセット値 : 0x0000 0000

OTG_FS_DAINTEPM レジスタは、デバイスエンドポイント割込みレジスタとともに機能して、デバイスエンドポイント上にイベントが発生すると、アプリケーションに割り込みます。ただし、その割込みに対応する OTG_FS_DAINTEP レジスタビットはセットされたままです。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OEPM																IEPM															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	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 **OEPM** : OUT EP 割込みマスクビット

OUT エンドポイント (EP) あたり 1 ビット :

OUT EP 0 に対してビット 16、OUT EP 3 に対してビット 18

0 : 割込みはマスクされます。

1 : 割込みはマスクされません。

ビット 15:0 **IEPM** : IN EP 割込みマスクビット

IN エンドポイントあたり 1 ビット :

IN EP 0 に対してビット 0、IN EP 3 に対してビット 3

0 : 割込みはマスクされます。

1 : 割込みはマスクされません。

OTG_FS デバイス V_{BUS} 放電時間レジスタ (OTG_FS_DVBUSDIS)

アドレスオフセット : 0x0828

リセット値 : 0x0000 17D7

このレジスタは SRP 中の V_{BUS} 放電時間 (V_{BUS} パルシング後) を指定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																VBUSDT															
																rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **VBUSDT** : デバイス V_{BUS} 放電時間
SRP 中の V_{BUS} 放電時間 (V_{BUS} パルシング後) を指定します。この値は以下のようになります。
 V_{BUS} 放電時間 (PHY クロック / 1024)
この値は V_{BUS} の負荷によって調整の必要があるかもしれません。

OTG_FS デバイス V_{BUS} パルシング時間レジスタ (OTG_FS_DVBUSPULSE)

アドレスオフセット : 0x082C

リセット値 : 0x0000 05B8

このレジスタは SRP 期間中の V_{BUS} パルシング時間を指定します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																DVBUSP															
																rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DVBUSP** : デバイス V_{BUS} パルシング時間
SRP 中の V_{BUS} パルシング時間を指定します。この値は以下のようになります。
 V_{BUS} パルシング時間 (PHY クロック / 1024)

OTG_FS デバイス IN エンドポイント FIFO エンプティ割込みマスクレジスタ : (OTG_FS_DIEPEMPMSK)

アドレスオフセット : 0x834

リセット値 : 0x0000 0000

このレジスタは、IN エンドポイント FIFO エンプティ割込みの生成 (TXFE_OTG_FS_DIEPINTx) を制御するために使用されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																INEPTXFEM															
																rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **INEPTXFEM** : IN EP Tx FIFO エンプティ割込みマスクビット

これらのビットは OTG_FS_DIEPINTx に対するマスクビットとして機能します。

IN EP あたり TXFE 割込み 1 ビット :

IN EP 0 に対してビット 0、IN EP 3 に対してビット 3

0 : 割込みはマスクされます。

1 : 割込みはマスクされません。

OTG_FS デバイス制御 IN エンドポイント 0 制御レジスタ (OTG_FS_DIEPCTL0)

アドレスオフセット : 0x900

リセット値 : 0x0000 0000

このセクションでは、OTG_FS_DIEPCTL0 レジスタについて説明します。非ゼロコントロールエンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPENA	EPDIS	予約済み				SNAK	CNAK	TXFNUM				STALL	予約済み	EPTYP		NAKSTS	予約済み	USBAEP	予約済み										MPSIZ		
r	r					w	w	rW	rW	rW	rW	rs		r	r	r		r													

ビット 31 **EPENA** : エンドポイントイネーブル

アプリケーションは、このビットをセットして、エンドポイント 0 でデータ送信を開始します。

コアは、このエンドポイント上で以下の割込みをセットする前に、このビットをクリアします。

- エンドポイント無効
- 転送完了

ビット 30 **EPDIS** : エンドポイントディセーブル

アプリケーションは、このビットをセットして、エンドポイントへの転送が完了する前でも、そのエンドポイントでのデータ送信を停止することができます。アプリケーションは、エンドポイントが無効として扱う前に、エンドポイント無効割込みを待つ必要があります。コアは、エンドポイント無効割込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイントイネーブルがすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 SNAK : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、エンドポイント上で SETUP パケットが受信された後、そのエンドポイントに対してこのビットをセットすることもできます。

ビット 26 CNAK : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 TXFNUM : TxFIFO 番号

この値は、IN エンドポイント 0 に割り当てられる FIFO 番号に設定されます。

ビット 21 STALL : STALL のハンドシェイク

アプリケーションは、このビットのセットのみが可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル INNAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。

ビット 20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 EPTYP : エンドポイントタイプ

コントロールのために"00"にハードコードされています。

ビット 17 NAKSTS : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

このビットがアプリケーションまたはコアによってセットされると、コアは、TxFIFO に使用可能なデータがある場合でも、データの送信を停止します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 USBAEP : USB アクティブエンドポイント

このビットは常に 1 にセットされ、コントロールエンドポイント 0 がすべての設定およびインタフェースにおいて常に有効であることを示します。

ビット 14:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 MPSIZ : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。

00 : 64 バイト

01 : 32 バイト

10 : 16 バイト

11 : 8 バイト

OTG デバイスエンドポイント x 制御レジスタ (OTG_FS_DIEPCTLx) (x = 1..3、 x = Endpoint_number)

アドレスオフセット : 0x900 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0000

アプリケーションは、このレジスタを使用して、エンドポイント 0 以外の各論理エンドポイントの動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
EPENA		EPDIS		SODDFRM		SD0PID/SEVNFRM		SNAK		CNAK		TXFNUM				STALL		予約済み		EPTYP		NAKSTS		EONUM/DPID		USBAEP		予約済み				MPSIZ															
rs	rs	w	w	w	w	rw	rw	rw	rw	rw/rs	rw	rw	r	r	rw	rw	rw																														

ビット 31 EPENA : エンドポイントイネーブル

アプリケーションは、このビットをセットして、エンドポイントでのデータ送信を開始します。

コアは、このエンドポイント上で以下の割込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効
- 転送完了

ビット 30 EPDIS : エンドポイントディセーブル

アプリケーションは、エンドポイントへの転送が完了する前でも、このビットをセットして、そのエンドポイントでのデータ送信／受信を停止することができます。アプリケーションは、エンドポイントが無効として扱う前に、エンドポイント無効割込みを待つ必要があります。コアは、エンドポイント無効割込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイントイネーブルがすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29 SODDFRM : 奇数フレームの設定

アイソクロナス IN および OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数／奇数フレーム (EONUM) フィールドが奇数フレームに設定されます。

ビット 28 SD0PID : DATA0 PID の設定

インタラプト／パルク IN エンドポイントにのみ適用されます。

このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA0 に設定されます。

SEVNFRM : 偶数フレームの設定

アイソクロナス IN エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数／奇数フレーム (EONUM) フィールドが偶数フレームに設定されます。

ビット 27 SNAK : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、送信完了割込み時、またはエンドポイントで SETUP パケットが受信された後、OUT エンドポイントに対してこのビットをセットすることもできます。

ビット 26 CNAK : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 TXFNUM : TxFIFO 番号

これらのビットは、このエンドポイントに関連する FIFO 番号を指定します。各有効 IN エンドポイントは、別々の FIFO 番号にプログラムされなければなりません。
このフィールドは、IN エンドポイントについてのみ有効です。

ビット 21 STALL : STALL のハンドシェイク

非制御の非アイソクロナス IN エンドポイントにのみ適用されます (アクセスタイプは rw です)。
アプリケーションは、このビットをセットして、USB ホストからこのエンドポイントへのすべてのトークンを停止します。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。アプリケーションでのみこのビットをクリアできます。コアはクリアできません。

コントロールエンドポイントにのみ適用されます (アクセスタイプは rs です)。

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット (グローバル IN NAK またはグローバル OUT NAK) がセットされた場合は、STALL ビットが優先されます。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 20 予約済みであり、リセット値に保持する必要があります。**ビット 19:18 EPTYP** : エンドポイントタイプ

これは、この論理エンドポイントがサポートする転送タイプです。

- 00 : コントロール
- 01 : アイソクロナス
- 10 : バルク
- 11 : インタラプト

ビット 17 NAKSTS : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、

非アイソクロナス IN エンドポイントの場合 : コアは、TxFIFO に使用可能なデータがある場合でも、データの送信を停止します。

アイソクロナス IN エンドポイントの場合 : コアは、TxFIFO に使用可能なデータがある場合でも、ゼロ長のデータパケットを送出します。

このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 EONUM : 偶数/奇数フレーム

アイソクロナス IN エンドポイントにのみ適用されます。

コアがこのエンドポイントにアイソクロナスデータを送受信するフレーム番号を示します。アプリケーションは、このレジスタの SEVNFRM および SODDFRM フィールドを使用して、このエンドポイントに対してアイソクロナスデータを送信/受信する偶数/奇数番号をプログラムする必要があります。

0 : 偶数フレーム

1 : 奇数フレーム

DPID : エンドポイントデータPID

インタラプト/バルク IN エンドポイントにのみ適用されます。

このエンドポイント上で受信または送信されるパケットの PID を含みます。アプリケーションは、エンドポイントがアクティブ化された後、このエンドポイントで受信または送信される最初のパケットの PID をプログラムする必要があります。アプリケーションは SC0PID レジスタを使用して、DATA0 または DATA1 PID のいずれかをプログラムします。

0 : DATA0

1 : DATA1

ビット 15 **USBAEP** : USB アクティブエンドポイント

このエンドポイントが現在の設定およびインタフェースでアクティブかどうかを示します。コアは、USB リセットを検出した後、すべてのエンドポイント (EP 0 を除き) について、このビットをクリアします。SetConfiguration および SetInterface コマンドを受信した後、アプリケーションはそれに応じてエンドポイントをプログラムし、このビットをセットする必要があります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。この値はバイト単位です。

OTG_FS デバイスコントロール OUT エンドポイント 0 制御レジスタ (OTG_FS_DOEPTCTL0)

アドレスオフセット : 0xB00

リセット値 : 0x0000 8000

このセクションでは、OTG_FS_DOEPTCTL0 レジスタについて説明します。非ゼロコントロールエンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
EPENA	EPDIS	予約済み				SNAK	CNAK	予約済み				STALL	SNPM	EPTYP		NAKSTS	予約済み		USBAEP	予約済み								MPSIZ				
w	r					w	w					rs	rw	r	r	r		r													r	r

ビット 31 **EPENA** : エンドポイントイネーブル

アプリケーションは、このビットをセットして、エンドポイント 0 上でデータ送信を開始します。コアは、このエンドポイント上で以下の割込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効
- 転送完了

ビット 30 **EPDIS** : エンドポイントディセーブル

アプリケーションは、コントロール OUT エンドポイント 0 を無効にできません。

ビット 29:28 予約済みであり、リセット値に保持する必要があります。

ビット 27 **SNAK** : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、送信完了割込み時、または SETUP パケットを受信した後に、このビットをセットすることもできます。

ビット 26 **CNAK** : NAK のクリア

このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。

ビット 25:22 予約済みであり、リセット値に保持する必要があります。

ビット 21 **STALL** : STALL のハンドシェイク

アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。NAK ビット、すなわち、グローバル OUT NAK がこのビットとともにセットされた場合は、STALL ビットが優先します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 20 **SNPM** : スヌープモード

このビットは、エンドポイントをスヌープモードに設定します。スヌープモードでは、コアは OUT パケットをアプリケーションメモリに転送する前に、それらの正確さをチェックしません。

ビット 19:18 **EPTYP** : エンドポイントタイプ

制御のために 2'b00 にハードコード化されています。

ビット 17 **NAKSTS** : NAK ステータス

以下を示します。

0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。

1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。

アプリケーションまたはコアがこのビットをセットすると、RxFIFO に受信パケットを収容できるスペースがある場合でも、コアはデータの受信を停止します。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **USBAEP** : USB アクティブエンドポイント

このビットは常に 1 にセットされ、コントロールエンドポイント 0 がすべての設定およびインタフェースにおいて常にアクティブであることを示します。

ビット 14:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **MPSIZ** : 最大パケットサイズ

制御 OUT エンドポイント 0 の最大パケットサイズは、制御 IN エンドポイント 0 でプログラムされるサイズと同じです。

00 : 64 バイト

01 : 32 バイト

10 : 16 バイト

11 : 8 バイト

OTG_FS デバイスエンドポイント x 制御レジスタ (OTG_FS_DOEPCTLx) (x = 1..3, x = Endpoint_number)

OUT エンドポイントのアドレスオフセット : $0xB00 + (\text{Endpoint_number} * 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、このレジスタを使用して、エンドポイント 0 以外の各論理エンドポイントの動作を制御します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	予約済み					STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み					MPSIZ									
rs	rs	w	w	w	w						rw/rs	rw	rw	rw	r	r	rw						rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **EPENA** : エンドポイントイネーブル

IN および OUT エンドポイントに適用されます。

アプリケーションは、このビットをセットして、エンドポイントでのデータ送信を開始します。

コアは、このエンドポイント上で以下の割込みをセットする前に、このビットをクリアします。

- SETUP フェーズ終了
- エンドポイント無効
- 転送完了

ビット 30 **EPDIS** : エンドポイントディセーブル

アプリケーションは、エンドポイントへの転送が完了する前でも、このビットをセットして、そのエンドポイントでのデータ送信/受信を停止することができます。アプリケーションは、エンドポイントが無効として扱う前に、エンドポイント無効割込みを待つ必要があります。コアは、エンドポイント無効割込みを設定する前に、このビットをクリアします。アプリケーションは、このエンドポイントのエンドポイントイネーブルがすでにセットされている場合のみ、このビットをセットしなければなりません。

ビット 29 **SD1PID** : DATA1 PID の設定

インタラプト/バルク IN および OUT エンドポイントにのみ適用されます。このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA1 に設定されます。

SODDFRM : 奇数フレームの設定

アイソクロナス IN および OUT エンドポイントにのみ適用されます。このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが奇数フレームに設定されます。

ビット 28 **SD0PID** : DATA0 PID の設定

インタラプト/バルク OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、このレジスタのエンドポイントデータ PID (DPID) フィールドが DATA0 に設定されます。

SEVNFRM : 偶数フレームの設定

アイソクロナス OUT エンドポイントにのみ適用されます。

このフィールドに書き込むと、偶数/奇数フレーム (EONUM) フィールドが偶数フレームに設定されます。

ビット 27 **SNAK** : NAK のセット

このビットに書き込むと、エンドポイントの NAK ビットがセットされます。

アプリケーションは、このビットを使用して、エンドポイント上での NAK ハンドシェイクの送信を制御できます。コアは、転送完了割込み時、または SETUP パケットを受信した後に、OUT エンドポイントに対してこのビットをセットすることもできます。

- ビット 26 **CNAK** : NAK のクリア
このビットに書き込むと、エンドポイントの NAK ビットがクリアされます。
- ビット 25:22 予約済みであり、リセット値に保持する必要があります。
- ビット 21 **STALL** : STALL のハンドシェイク
非コントロールの非アイソクロナス OUT エンドポイントにのみ適用されます（アクセスタイプは rw です）。
アプリケーションは、このビットをセットして、USB ホストからこのエンドポイントへのすべてのトークンを停止します。このビットとともに NAK ビット（グローバル IN NAK またはグローバル OUT NAK）がセットされた場合は、STALL ビットが優先されます。アプリケーションでのみこのビットをクリアできます。コアはクリアできません。
コントロールエンドポイントにのみ適用されます（アクセスタイプは rs です）。
アプリケーションは、このビットのセットのみ可能です。このエンドポイントの SETUP トークンが受信されると、コアがこのビットをクリアします。このビットとともに NAK ビット（グローバル IN NAK またはグローバル OUT NAK）がセットされた場合は、STALL ビットが優先されます。このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。
- ビット 20 **SNPM** : スヌープモード
このビットは、エンドポイントをスヌープモードに設定します。スヌープモードでは、コアは OUT パケットをアプリケーションメモリに転送する前に、それらの正確さをチェックしません。
- ビット 19:18 **EPTYP** : エンドポイントタイプ
これは、この論理エンドポイントがサポートする転送タイプです。
00 : コントロール
01 : アイソクロナス
10 : バルク
11 : インタラプト
- ビット 17 **NAKSTS** : NAK ステータス
以下を示します。
0 : コアは、FIFO のステータスに基づいて非 NAK ハンドシェイクを送信中です。
1 : コアは、このエンドポイント上で NAK ハンドシェイクを送信中です。
アプリケーションまたはコアがこのビットをセットすると、
コアは、RxFIFO に受信パケットを収容するためのスペースがある場合でも、OUT エンドポイントでのデータの受信を停止します。
このビットの設定に関係なく、コアは、常に SETUP データパケットに ACK ハンドシェイクで応答します。
- ビット 16 **EONUM** : 偶数／奇数フレーム
アイソクロナス IN および OUT エンドポイントにのみ適用されます。
コアがこのエンドポイントにアイソクロナスデータを送受信するフレーム番号を示します。アプリケーションは、このレジスタの SEVNFRM および SODDFRM フィールドを使用して、このエンドポイントに対してアイソクロナスデータを送信／受信する偶数／奇数番号をプログラムする必要があります。
0 : 偶数フレーム
1 : 奇数フレーム
DPID : エンドポイントデータPID
インタラプト／バルク OUT エンドポイントにのみ適用されます。
このエンドポイント上で受信または送信されるパケットの PID を含みます。アプリケーションは、エンドポイントがアクティブ化された後、このエンドポイントで受信または送信される最初のパケットの PID をプログラムする必要があります。アプリケーションは SC0PID レジスタを使用して、DATA0 または DATA1 PID のいずれかをプログラムします。
0 : DATA0
1 : DATA1

ビット 15 **USBAEP** : USB アクティブエンドポイント

このエンドポイントが現在の設定およびインタフェースでアクティブかどうかを示します。コアは、USB リセットを検出した後、すべてのエンドポイント (EP 0 を除き) について、このビットをクリアします。SetConfiguration および SetInterface コマンドを受信した後、アプリケーションはそれに応じてエンドポイントをプログラムし、このビットをセットする必要があります。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:0 **MPSIZ** : 最大パケットサイズ

アプリケーションは、このフィールドを現在の論理エンドポイントの最大パケットサイズでプログラムする必要があります。この値はバイト単位です。

OTG_FS デバイスエンドポイント x 割込みレジスタ (OTG_FS_DIEPINTx) (x = 0..3、x = Endpoint_number)

アドレスオフセット : 0x908 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0080

このレジスタは、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。[図 250](#) に示されています。アプリケーションは、コア割込みレジスタの IN エンドポイント割込みビット (OTG_FS_GINTSTS の IEPINT ビット) がセットされたとき、このレジスタを読み出さなければなりません。アプリケーションがこのレジスタを読み出すためには、まず、デバイス全エンドポイント割込み (OTG_FS_DAINTE) レジスタを読み出して、デバイスのエンドポイント x 割込みレジスタの正確なエンドポイント番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_FS_DAINTE および OTG_FS_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																								TXFE	INEPNE	予約済み	ITTXFE	TOC	予約済み	EPDSD	XFRC
																								r	rc_w1/rw		rc_w1	rc_w1		rc_w1	rc_w1

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TXFE** : 送信 FIFO エンプティ

この割込みは、このエンドポイントの TxFIFO が半エンプティまたは完全エンプティのいずれかであるとき、アサートされます。半エンプティステータスか完全エンプティステータスかは、OTG_FS_GAHBCFG レジスタの TxFIFO エンプティレベルビット (OTG_FS_GAHBCFG の TXFELVL ビット) によって判断できます。

ビット 6 **INEPNE** : IN エンドポイント NAK 有効

このビットは、アプリケーションが OTG_FS_DIEPCTLx レジスタの CNAK ビットに書き込むことによって IN エンドポイント NAK をクリアするとクリアされます。
この割込みは、コアが (アプリケーションまたはコアによって) 設定された NAK ビットをサンプリングしたことを示します。この割込みは、アプリケーションによってセットされた IN エンドポイント NAK ビットがコアで有効になったことを示します。
この割込みは、必ずしも NAK ハンドシェイクが USB に送出されることを保証しません。STALL ビットが NAK ビットに優先します。

ビット 5 予約済みであり、リセット値に保持する必要があります。



- ビット 4 **ITTXFE** : TxFIFO がエンプティ時に受信された IN トークン
非アイソクロナス IN エンドポイントにのみ適用されます。
関連する TxFIFO (周期的/非周期的) が空のときに IN トークンが受信されたことを示します。この割り込みは、IN トークンが受信されたエンドポイント上でアサートされます。
- ビット 3 **TOC** : タイムアウト条件
コントロール IN エンドポイントにのみ適用されます。
このエンドポイント上の最後の IN トークンに対して、コアが USB 上でタイムアウト条件を検出したことを示します。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **EPDISD** : エンドポイントディセーブル割り込み
このビットは、エンドポイントがアプリケーションのリクエストによって無効にされたことを示します。
- ビット 0 **XFRC** : 転送完了割り込み
このフィールドは、このエンドポイントに対してプログラムされた転送が AHB に加えて USB 上でも完了したことを示します。

OTG_FS デバイスエンドポイント x 割り込みレジスタ (OTG_FS_DOEPINTx) (x = 0..3、x = Endpoint_number)

アドレスオフセット : 0xB08 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0080

このレジスタは、USB および AHB 関連イベントに関するエンドポイントのステータスを示します。
[図 250](#) に示されています。アプリケーションは、OTG_FS_GINTSTS レジスタの OUT エンドポイント割り込みビット (OTG_FS_GINTSTS の OEPINT ビット) がセットされたとき、このレジスタを読み出さなければなりません。アプリケーションがこのレジスタを読み出すためには、まず、OTG_FS_DAINTE レジスタを読み出して、OTG_FS_DOEPINTx レジスタの正確なエンドポイント番号を知る必要があります。アプリケーションは、このレジスタの適切なビットをクリアすることによって、OTG_FS_DAINTE および OTG_FS_GINTSTS レジスタの対応するビットをクリアする必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
予約済み																								予約済み	B2BSTUP		予約済み	OTEPDIS		予約済み	EPDISD		予約済み	XFRC	
																									rc_w1/rw			rc_w1	rc_w1			rc_w1		rc_w1	

- ビット 31:7 予約済みであり、リセット値に保持する必要があります。
- ビット 6 **B2BSTUP** : 連続 SETUP パケットの受信
制御 OUT エンドポイントにのみ適用されます。
このビットは、コアがこの特定のエンドポイントで 3 つを超える連続 SETUP パケットを受信したことを示します。
- ビット 5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **OTEPDIS** : エンドポイント無効時 OUT トークン受信
コントロール OUT エンドポイントにのみ適用されます。
エンドポイントがまだ有効になっていないときに OUT トークンを受け取ったことを示します。この割り込みは、OUT トークンが受信されたエンドポイントでアサートされます。

- ビット 3 **STUP** : SETUP フェーズ終了
- コントロール OUT エンドポイントにのみ適用されます。
- コントロールエンドポイントの SETUP フェーズが完了し、現在の制御転送のために、これ以上、連続 SETUP パケットは受信されなかったことを示します。この割込みにより、アプリケーションは、受信したSETUP データパケットをデコードできます。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **EPDISD** : エンドポイントディセーブル割込み
- このビットは、エンドポイントがアプリケーションのリクエストによって無効にされたことを示します。
- ビット 0 **XFRC** : 転送完了割込み
- このフィールドは、このエンドポイントに対してプログラムされた転送が AHB に加えて USB 上でも完了したことを示します。

OTG_FS デバイス IN エンドポイント 0 転送サイズレジスタ (OTG_FS_DIEPTSIZ0)

- アドレスオフセット : 0x910
- リセット値 : 0x0000 0000
- アプリケーションは、エンドポイント 0 を有効にする前に、このレジスタを変更する必要があります。デバイスコントロールエンドポイント 0 制御レジスタのエンドポイントイネーブルビット (OTG_FS_DIEPCTL0 の EPENA) を使用して、エンドポイント 0 が有効にされると、コアは、このレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことしかできません。
- 非ゼロエンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み												PKTCNT		予約済み										XFRSIZ							
												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 20:19 **PKTCNT** : パケットカウント
- エンドポイント 0 のデータ転送サイズを構成する USB パケットの総数を示します。
- このフィールドは、パケットが TxFIFO から読み出されるたびにデクリメントされます (最大サイズまたはショートパケット)。
- ビット 18:7 予約済みであり、リセット値に保持する必要があります。
- ビット 6:0 **XFRSIZ** : 転送サイズ
- エンドポイント 0 の転送サイズをバイト単位で示します。コアは、データの転送サイズを使いきった後でのみ、アプリケーションに割込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割込みを生成できます。
- コアは、外部メモリからのパケットが TxFIFO に書き込まれるたびに、このフィールドをデクリメントします。

OTG_FS デバイス OUT エンドポイント 0 転送サイズレジスタ (OTG_FS_DOEPTSIZ0)

アドレスオフセット : 0xB10

リセット値 : 0x0000 0000

アプリケーションは、エンドポイント 0 を有効にする前に、このレジスタを変更する必要があります。OTG_FS_DOEPCTL0 レジスタのエンドポイントイネーブルビット (OTG_FS_DOEPCTL0 の EPENA ビット) を使用してエンドポイント 0 が有効にされると、コアは、このレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことしかできません。

非ゼロエンドポイントは、エンドポイント 1~3 のレジスタを使用します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	STUPCNT		予約済み										PKTCNT	予約済み										XFRSIZ							
	rw	rw																							rw	rw	rw	rw	rw	rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **STUPCNT** : SETUP パケットカウント

このフィールドは、エンドポイントが受信できる連続 SETUP データパケット数を指定します。

01 : 1 パケット

10 : 2 パケット

11 : 3 パケット

ビット 28:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **PKTCNT** : パケットカウント

このフィールドは、パケットが RxFIFO に書き込まれた後、ゼロまでデクリメントされます。

ビット 18:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:0 **XFRSIZ** : 転送サイズ

エンドポイント 0 の転送サイズをバイト単位で示します。コアは、データの転送サイズを使い切った後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。

コアは、RxFIFO からパケットが読み出されて、外部メモリに書き込まれるたびに、このフィールドをデクリメントします。

OTG_FS デバイスエンドポイント x 転送サイズレジスタ (OTG_FS_DIEPTSIZx)
(x = 1..3、x = Endpoint_number)

アドレスオフセット : 0x910 + (Endpoint_number * 0x20)

リセット値 : 0x0000 0000

アプリケーションは、エンドポイントを有効にする前に、このレジスタを変更する必要があります。
OTG_FS_DIEPCTLx レジスタのエンドポイントイネーブルビット (OTG_FS_DIEPCTLx の EPENA ビット) を使用してエンドポイントが有効にされると、コアは、このレジスタを変更します。コアが
エンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すこと
しかできません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	MCNT		PKTCNT										XFRSIZ																		
	rw/r /rw	rw/r /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 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **MCNT** : マルチカウント

周期的 IN エンドポイントの場合、このフィールドは、USB でフレームごとに送信されなければならない
パケット数を示します。コアは、このフィールドを使用して、アイソクロナス IN エンドポイントの
データ PID を計算します。

- 01 : 1 パケット
- 10 : 2 パケット
- 11 : 3 パケット

ビット 28:19 **PKTCNT** : パケットカウント

このエンドポイントのデータの転送サイズを構成する USB パケットの総数を示します。
このフィールドは、パケットが TxFIFO から読み出されるたびにデクリメントされます (最大サイズま
たはショートパケット)。

ビット 18:0 **XFRSIZ** : 転送サイズ

このフィールドは、現在のエンドポイントの転送サイズをバイト単位で示します。コアは、データの転
送サイズを使い終わった後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイント
の最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。
コアは、外部メモリからのパケットが TxFIFO に書き込まれるたびに、このフィールドをデクリメント
します。



OTG_FS デバイス IN エンドポイント送信 FIFO ステータスレジスタ (OTG_FS_DTXFSTSx) (x = 0..3, x = Endpoint_number)

IN エンドポイントのアドレスオフセット : $0x918 + (\text{Endpoint_number} \times 0x20)$ この読み出し専用レジスタは、デバイス IN エンドポイント Tx FIFO の空きスペース情報を含みます。

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

31:16 予約済みであり、リセット値に保持する必要があります。

15:0 **INEPTFSAV** : IN エンドポイント Tx FIFO の使用可能スペース

エンドポイントの Tx FIFO で使用できる空きスペースの量を示します。

値は 32 ビットワード単位です。

0x0 : エンドポイント Tx FIFO はフルです。

0x1 : 1 ワードが使用可能です。

0x2 : 2 ワードが使用可能です。

0xn : n ワードが使用可能です。

その他 : 予約済み

OTG_FS デバイス OUT エンドポイント x 転送サイズレジスタ (OTG_FS_DOEPTSIZx) (x = 1..3, x = Endpoint_number)

アドレスオフセット : $0xB10 + (\text{Endpoint_number} \times 0x20)$

リセット値 : 0x0000 0000

アプリケーションは、エンドポイントを有効にする前に、このレジスタを変更する必要があります。OTG_FS_DOEPCTLx レジスタのエンドポイントイネーブルビット (OTG_FS_DOEPCTLx の EPENA ビット) を使用してエンドポイントが有効にされると、コアは、このレジスタを変更します。コアがエンドポイントイネーブルビットをクリアすると、アプリケーションはこのレジスタを読み出すことができず。

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
予約済み	RXDPID/ST UPCNT		PKTCNT													XFRSIZ															
	rw/r	rw/r	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 予約済みであり、リセット値に保持する必要があります。

ビット 30:29 **RXDPID** : 受信データ PID

アイソクロナス OUT エンドポイントにのみ適用されます。

これは、このエンドポイントの最後のパケットで受信されたデータの PID です。

00 : DATA0

01 : データ 2

10 : DATA1

11 : MDATA

STUPCNT : SETUP パケットカウント

コントロール OUT エンドポイントにのみ適用されます。
このフィールドは、エンドポイントが受信できる連続 SETUP データパケット数を指定します。
01 : 1 パケット
10 : 2 パケット
11 : 3 パケット

ビット 28:19 **PKTCNT** : パケットカウント

このエンドポイントのデータの転送サイズを構成する USB パケットの総数を示します。
このフィールドは、パケット（最大サイズまたはショートパケット）が RxFIFO に書き込まれるたびにデクリメントされます。

ビット 18:0 **XFRSIZ** : 転送サイズ

このフィールドは、現在のエンドポイントの転送サイズをバイト単位で示します。コアは、データの転送サイズを使い終わった後でのみ、アプリケーションに割り込みをかけます。転送サイズをエンドポイントの最大パケットサイズにセットすると、各パケットの最後で割り込みを生成できます。
コアは、RxFIFO からパケットが読み出されて、外部メモリに書き込まれるたびに、このフィールドをデクリメントします。

22.16.5 OTG_FS パワーおよびクロックゲーティング制御レジスタ (OTG_FS_PCGCTL)

アドレスオフセット : 0xE00
リセット値 : 0x0000 0000
このレジスタは、ホストモードとデバイスモードで使用できます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み																												PHYSUSP	予約済み	GATEHCLK	STPPCLK
																												RW		RW	RW

- ビット 31:5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **PHYSUSP** : PHYをサスペンド
PHY がサスペンドされたことを示します。このビットは、アプリケーションが STPPCLK ビット（ビット 0）をセットした後、PHY がサスペンドされると更新されます。
- ビット 3:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **GATEHCLK** : HCLK をゲート
USB がサスペンドされるか、セッションが有効でないとき、アプリケーションは、このビットをセットして、AHB スレーブとマスタおよびウェイクアップロジック以外のモジュールへの HCLK をゲーティングします。アプリケーションは、USB がレジュームされるか、新しいセッションが開始されると、このビットをクリアします。
- ビット 0 **STPPCLK** : PHY クロックを停止
USB がサスペンドされるか、セッションが有効でないか、デバイスが切断されると、アプリケーションは、このビットをセットして、PHY クロックを停止します。アプリケーションは、USB がレジュームされるか、新しいセッションが開始されると、このビットをクリアします。

22.16.6 OTG_FS レジスタマップ

次の表に、USB OTG のレジスタマップとリセット値を示します。

表 133. OTG_FS レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x000	OTG_FS_GOT GCTL	予約済み												BSVLD	ASVLD	DBCT	CIDSTS	予約済み				DHNPN	HSHNPN	HNPRQ	HNGSCS	予約済み				SRQ	SROSCS				
	リセット値													0	0	0	1					0	0	0	0					0	0				
0x004	OTG_FS_GOT GINT	予約済み												DBCNE	ADTOCHG	HNGDET	予約済み				HNSSCHG		SRSSCHG	予約済み				SEDET	Res						
	リセット値													0	0	0					0		0					0							
0x008	OTG_FS_GAH BCFG	予約済み																						PTXFELVL	TXFELVL	予約済み				GINTMSK					
	リセット値																							0	0					0					
0x00C	OTG_FS_GUS BCFG	CTXPKT	FDMOD	FHMOD	予約済み												TRDT				HNPCAP	SRPCAP	予約済み	PHYSEL	予約済み		TOTAL								
	リセット値																0	1	0	0	1	0	0	1			0	0	0						
0x010	OTG_FS_GRS TCTL	AHBIDL	予約済み																		TXFNUM				RXFFLSH	予約済み	FORST	HSRST	CSRST						
	リセット値	1																			0	0	0		0	0	0	0	0	0					
0x014	OTG_FS_GIN TSTS	WKUINT	SRQINT	DISCINT	CIDSCHG	予約済み		PTXFE	HCINT	HPRTINT	予約済み		IPXFR/INCOMPISOOUT	IIISOIXFR	OEPI	IEPI	予約済み		EOPF	ISOODRP	ENUMDNE	USBRST	USBSUSP	ESUSP	予約済み		GOUTNAKEFF	GINAKEFF	NPTXFE	RXFLVL	SOF	OTGINT	MMIS	CMOD	
	リセット値	0	0	0	0			1	0	0			0	0	0	0			0	0	0	0	0			0	0	1	0	0	0	0	0		
0x018	OTG_FS_GIN TMSK	WUIM	SRQIM	DISCINT	CIDSCHGM	予約済み		PTXFEM	HCIM	PRTIM	予約済み		IPXFRM/ISOXXFRM	IIISOIXFRM	OEPI	IEPI	EPMISM	予約済み		EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	予約済み		GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	予約済み
	リセット値	0	0	0	0			0	0	0			0	0	0	0	0			0	0	0	0	0			0	0	0	0	0	0	0	0	
0x01C	OTG_FS_GRX STSR (ホスト モード)	予約済み												PKTSTS				DPID		BCNT								CHNUM							
	リセット値													0 0 0 0				0		0 0 0 0 0 0 0 0 0 0 0 0								0 0 0 0							
	OTG_FS_GRX STSR (デバイ スモード)	予約済み								FRMNUM				PKTSTS				DPID		BCNT								EPNUM							
	リセット値									0 0 0 0				0 0 0 0				0 0		0 0 0 0 0 0 0 0 0 0 0 0								0 0 0 0							

表 133. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
0x020	OTG_FS_GRX STSR（ホスト モード）	予約済み											PKTSTS				DPID		BCNT										CHNUM													
	リセット値												0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
	OTG_FS_GRX STSPR（デバ イスモード）	予約済み							FRMNUM				PKTSTS				DPID		BCNT										EPNUM													
	リセット値								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x024	OTG_FS_GRX FSIZ	予約済み															RXFD																									
	リセット値																0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x028	OTG_FS_HNPTX FSIZ/ OTG_FS_DIEPT XF0	NPTXFD/TX0FD															NPTXFSA/TX0FSA																									
	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	0	0	0	0	0	0	0	0									
0x02C	OTG_FS_HNPTX STS	Res.	NPTXQTOP							NPTQXSAV							NPTXFSAV																									
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0									
0x038	OTG_FS GCCFG	予約済み										NOVBUSSENS	SOFOUTEN	VBUSSEN	VBUSASEN	予約済み	PWRDWN	予約済み																								
	リセット値											0	0	0	0		0																									
0x03C	OTG_FS_CID	PRODUCT_ID																																								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0									
0x100	OTG_FS_HPTXF SIZ	PTXFSIZ															PTXSA																									
	リセット値	0	0	0	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0									
0x104	OTG_FS_DIEPT XF1	INEPTXFD															INEPTXSA																									
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0									
0x108	OTG_FS_DIEPT XF2	INEPTXFD															INEPTXSA																									
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0									
0x10C	OTG_FS_DIEPT XF3	INEPTXFD															INEPTXSA																									
	リセット値	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0									
0x400	OTG_FS_HCFG	予約済み																												FSLSS		FSLSPCS										
	リセット値																													0	0	0										
0x404	OTG_FS_HFIR	予約済み															FRIVL																									
	リセット値																1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0			
0x408	OTG_FS_HFNU M	FTREM															FRNUM																									
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1									
0x410	OTG_FS_HPTXS TS	PTXQTOP							PTXQSAV							PTXFSAVL																										
	リセット値	0	0	0	0	0	0	0	0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y									
0x414	OTG_FS_HAINT	予約済み															HAINT																									
	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
0x418	OTG_FS_HAINT MSK	予約済み															HAINTM																									
	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

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											
0x440	OTG_FS_HPRT	予約済み													PSPD		PTCTL				PPWR		PLSTS		予約済み		PRST		PSUSP		PRES		POCHNG		POCA		PENCHNG		PENA		PODET		PCSTS	
	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			
0x500	OTG_FS_HCCH AR0	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x520	OTG_FS_HCCH AR1	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x540	OTG_FS_HCCH AR2	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x560	OTG_FS_HCCH AR3	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x580	OTG_FS_HCCH AR4	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x5A0	OTG_FS_HCCH AR5	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x5C0	OTG_FS_HCCH AR6	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x5E0	OTG_FS_HCCH AR7	CHENA	CHDIS	ODDFRM	DAD								MCNT		EPTYP		LSDEV	予約済み	EPDIR	EPNUM				MPSIZ																				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
0x508	OTG_FS_HCINT 0	予約済み																				DTERR		FRMOR		BBERR		TXERR		予約済み	ACK	NAK		STALL		予約済み	CHH	XFRC						
	リセット値																					0		0		0		0		0	0	0		0		0	0	0						
0x528	OTG_FS_HCINT 1	予約済み																				DTERR		FRMOR		BBERR		TXERR		予約済み	ACK	NAK		STALL		予約済み	CHH	XFRC						
	リセット値																					0		0		0		0		0	0	0		0		0	0	0						
0x548	OTG_FS_HCINT 2	予約済み																				DTERR		FRMOR		BBERR		TXERR		予約済み	ACK	NAK		STALL		予約済み	CHH	XFRC						
	リセット値																					0		0		0		0		0	0	0		0		0	0	0						
0x568	OTG_FS_HCINT 3	予約済み																				DTERR		FRMOR		BBERR		TXERR																

表 133. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x588	OTG_FS_HCINT 4	予約済み																				DTERR	FRMOR	BBERR	TXERR	予約済み	ACK	NAK	STALL	予約済み	CHH	XFRC	
	リセット値																					0	0	0	0		0	0					
0x5A8	OTG_FS_HCINT 5	予約済み																				DTERR	FRMOR	BBERR	TXERR	予約済み	ACK	NAK	STALL	予約済み	CHH	XFRC	
	リセット値																					0	0	0	0		0	0					
0x5C8	OTG_FS_HCINT 6	予約済み																				DTERR	FRMOR	BBERR	TXERR	予約済み	ACK	NAK	STALL	予約済み	CHH	XFRC	
	リセット値																					0	0	0	0		0	0					
0x5E8	OTG_FS_HCINT 7	予約済み																				DTERR	FRMOR	BBERR	TXERR	予約済み	ACK	NAK	STALL	予約済み	CHH	XFRC	
	リセット値																					0	0	0	0		0	0					
0x50C	OTG_FS_HCINT MSK0	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x52C	OTG_FS_HCINT MSK1	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x54C	OTG_FS_HCINT MSK2	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x56C	OTG_FS_HCINT MSK3	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x58C	OTG_FS_HCINT MSK4	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x5AC	OTG_FS_HCINT MSK5	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x5CC	OTG_FS_HCINT MSK6	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x5EC	OTG_FS_HCINT MSK7	予約済み																				DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	予約済み	CHHM	XFRM	
	リセット値																					0	0	0	0	0	0	0					
0x510	OTG_FS_HCTSIZ0	予約済み	DPID	PKTCNT										XFRSIZ																			
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x530	OTG_FS_HCTSIZ1	予約済み	DPID	PKTCNT										XFRSIZ																			
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

表 133. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x550	OTG_FS_HCTSI Z2	予約済み	DPID		PKTCNT										XFRSIZ																		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x570	OTG_FS_HCTSI Z3	予約済み	DPID		PKTCNT										XFRSIZ																		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x590	OTG_FS_HCTSI Z4	予約済み	DPID		PKTCNT										XFRSIZ																		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5B0	OTG_FS_HCTSI Z5	予約済み	DPID		PKTCNT										XFRSIZ																		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5D0	OTG_FS_HCTSI Z6	予約済み	DPID		PKTCNT										XFRSIZ																		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5F0	OTG_FS_HCTSI Z7	予約済み	DPID		PKTCNT										XFRSIZ																		
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x800	OTG_FS_DCFG	予約済み														PFIVL		DAD				予約済み	NZLSOHSK		DSPD								
	リセット値																										0	0	0	0	0	0	0
0x804	OTG_FS_DCTL	予約済み														POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL		GONSTS	GINSTS	SDIS	RWUSIG							
	リセット値																										0	0	0	0	0	0	0
0x808	OTG_FS_DSTS	予約済み										FNSOF										予約済み		EERR	ENUMSPD	SUSPSTS							
	リセット値																										0	0	0	0	0	0	0
0x810	OTG_FS_DIEPM SK	予約済み														INENEM	INENMM	ITTXFEMSK	TOM	予約済み	EPDM	XFERM											
	リセット値																						0	0	0	0	0	0	0	0			
0x814	OTG_FS_DOEP MSK	予約済み														OTEPDM	STUPM	予約済み	EPDM	XFERM													
	リセット値																				0	0	0	0	0	0	0	0					
0x818	OTG_FS_DAIINT	OEPINT										IEPINT																					
	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	
0x81C	OTG_FS_DAIINT MSK	OEPM										IEPM																					
	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	
0x828	OTG_FS_DVBUS SDIS	予約済み										VBUSDT																					
	リセット値																													0	0	0	1
0x82C	OTG_FS_DVBUS SPULSE	予約済み										DVBUSP																					
	リセット値																													0	1	0	0
0x834	OTG_FS_DIEPE MPMSK	予約済み										INEPTXFEM																					
	リセット値																													0	0	0	0

表 133. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
0x900	OTG_FS_DIEPC TL0	EPENA	EPDIS	予約済み	SNAK	CNAK	TXFNUM					STALL	予約済み	EPTYP	NAKSTS	予約済み	USBAEP	予約済み										MPSI Z													
	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
0x918	TG_FS_DTXFST S0	予約済み																INEPTFSAV																							
	リセット値																	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x920	OTG_FS_DIEPC TL1	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM					STALL	予約済み	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み	MPSIZ																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x938	TG_FS_DTXFST S1	予約済み																INEPTFSAV																							
	リセット値																	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x940	OTG_FS_DIEPC TL2	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM					STALL	予約済み	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み	MPSIZ																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x958	TG_FS_DTXFST S2	予約済み																INEPTFSAV																							
	リセット値																	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x960	OTG_FS_DIEPC TL3	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM					STALL	予約済み	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み	MPSIZ																					
	リセット値	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x978	TG_FS_DTXFST S3	予約済み																INEPTFSAV																							
	リセット値																	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB00	OTG_FS_DOEP CTL0	EPENA	EPDIS	予約済み	SNAK	CNAK	予約済み					STALL	SNPM	EPTYP	NAKSTS	予約済み	USBAEP	予約済み										MPSI Z													
	リセット値	0	0				0	0	0	0	0							0	0	0	0	0	0	1	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0xB20	OTG_FS_DOEP CTL1	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	予約済み					STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み	MPSIZ																					
	リセット値	0	0	0	0	0	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0xB40	OTG_FS_DOEP CTL2	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	予約済み					STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み	MPSIZ																					
	リセット値	0	0	0	0	0	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

表 133. OTG_FS レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0xB60	OTG_FS_DOEP CTL3	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFIRM	SNAK	CNAK	予約済み				STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	予約済み				MPSIZ													
	リセット値	0	0	0	0	0	0					0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	
0x908	OTG_FS_DIEPIN T0	予約済み																								TXFE	INEPNE	予約済み	ITTXFE	TOC	予約済み	EPDISD	XFRC		
	リセット値																									1	0	0	0	0	0				
0x928	OTG_FS_DIEPIN T1	予約済み																								TXFE	INEPNE	予約済み	ITTXFE	TOC	予約済み	EPDISD	XFRC		
	リセット値																									1	0	0	0	0	0				
0x948	OTG_FS_DIEPIN T2	予約済み																								TXFE	INEPNE	予約済み	ITTXFE	TOC	予約済み	EPDISD	XFRC		
	リセット値																									1	0	0	0	0	0				
0x968	OTG_FS_DIEPIN T3	予約済み																								TXFE	INEPNE	予約済み	ITTXFE	TOC	予約済み	EPDISD	XFRC		
	リセット値																									1	0	0	0	0	0				
0xB08	OTG_FS_DOEPI NT0	予約済み																								予約済み	B2BSTUP	予約済み	OTEPDIS	STUP	予約済み	EPDISD	XFRC		
	リセット値																									0	0	0	0	0	0				
0xB28	OTG_FS_DOEPI NT1	予約済み																								予約済み	B2BSTUP	予約済み	OTEPDIS	STUP	予約済み	EPDISD	XFRC		
	リセット値																									0	0	0	0	0	0				
0xB48	OTG_FS_DOEPI NT2	予約済み																								予約済み	B2BSTUP	予約済み	OTEPDIS	STUP	予約済み	EPDISD	XFRC		
	リセット値																									0	0	0	0	0	0				
0xB68	OTG_FS_DOEPI NT3	予約済み																								予約済み	B2BSTUP	予約済み	OTEPDIS	STUP	予約済み	EPDISD	XFRC		
	リセット値																									0	0	0	0	0	0				
0x910	OTG_FS_DIEPT SIZ0	予約済み										PKTC NT		予約済み										XFRSIZ											
	リセット値											0 0												0 0 0 0 0 0 0 0											
0x930	OTG_FS_DIEPT SIZ1	予約済み	MCNT		PKTCNT										XFRSIZ																				
	0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0																															
0x950	OTG_FS_DIEPT SIZ2	予約済み	MCNT		PKTCNT										XFRSIZ																				
	0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0																															
0x970	OTG_FS_DIEPT SIZ3	予約済み	MCNT		PKTCNT										XFRSIZ																				
	0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0																															
0xB10	OTG_FS_DOEP TSIZ0	予約済み	STUP CNT		予約済み										PKTCNT		予約済み										XFRSIZ								
	0 0												0											0 0 0 0 0 0 0 0											

22.17 OTG_FS プログラミングモデル

22.17.1 コアの初期化

アプリケーションはコア初期化シーケンスを実行する必要があります。パワーアップ中にケーブルが接続された場合、OTG_FS_GINTSTS レジスタの現在の動作モードビット (OTG_FS_GINTSTS の CMOD ビット) が、モードを反映します。OTG_FS コントローラは、A プラグが接続されたときにはホストモードに、B プラグが接続されたときにはデバイスモードに入ります。

このセクションでは、電源投入後の OTG_FS コントローラの初期化について説明します。アプリケーションは、ホストモードかデバイスモードかに関係なく、初期化シーケンスに従う必要があります。コアのすべてのグローバルレジスタは、コアの設定に従って以下の手順で初期化されます。

1. OTG_FS_GAHBCFG レジスタの以下のフィールドをプログラムします。
 - グローバル割込みマスクビット GINTMSK = 1
 - RxFIFO 非エンプティ (OTG_FS_GINTSTS の RXFLVL ビット)
 - 周期的 TxFIFO エンプティレベル
2. OTG_FS_GUSBCFG レジスタの以下のフィールドをプログラムします。
 - HNP 対応ビット
 - SRP 対応ビット
 - FS タイムアウト較正フィールド
 - USB ターンアラウンド時間フィールド
3. ソフトウェアは、OTG_FS_GINTMSK レジスタの以下のビットのマスクを解除する必要があります。
OTG 割込みマスク
モード不一致割込みマスク
4. ソフトウェアは、OTG_FS_GINTSTS レジスタの CMOD ビットを読み出すことによって、OTG_FS コントローラがホストモードで動作中か、デバイスモードで動作中かを判別することができます。

22.17.2 ホストの初期化

コアをホストとして初期化するには、アプリケーションは、以下の手順を実行しなければなりません。

1. OTG_FS_GINTMSK レジスタの HPRTTINT ビットをプログラムして、マスクを解除します。
2. OTG_FS_HCFG レジスタをプログラムして、フルスピードホストを選択します。
3. OTG_FS_HCFG レジスタの PPWR ビットを 1 にプログラムします。これによって USB 上で V_{BUS} が駆動されます。
4. OTG_FS_HPRT0 の PCDET 割込みを待ちます。これは、デバイスがポートに接続されていることを示します。
5. OTG_FS_HPRT レジスタの PRST ビットを 1 にプログラムします。これによってリセットプロセスが開始されます。
6. リセットプロセスが完了するまで、少なくとも 10 ms 待ちます。
7. OTG_FS_HPRT レジスタの PRST ビットを 0 にプログラムします。
8. OTG_FS_HPRT レジスタの PENCHNG 割込みを待ちます。
9. OTG_FS_HPRT レジスタの PDPD ビットを読み出して、エニュメレーションされた速度を取得します。
10. HFIR レジスタを、選択された PHY クロック 1 に対応する値でプログラムします。
11. OTG_FS_HCFG レジスタの FSLSPCS フィールドを、手順 9 で検出した速度に従ってプログラムします。FSLSPCS が変更されている場合は、ポートのリセットを行う必要があります。
12. OTG_FS_GRXFSIZ レジスタをプログラムして、受信 FIFO のサイズを選択します。
13. OTG_FS_HNPTXFSIZ レジスタをプログラムして、非周期的トランザクションの非周期的送信 FIFO のサイズと開始アドレスを選択します。
14. OTG_FS_HPTXFSIZ レジスタをプログラムして、周期的トランザクションの周期的送信 FIFO のサイズと開始アドレスを選択します。

デバイスと通信するには、システムソフトウェアは、少なくとも 1 つのチャンネルを初期化し、有効にする必要があります。

22.17.3 デバイスの初期化

アプリケーションは、パワーアップ時、またはホストモードからデバイスモードへの変更後、以下の手順を実行して、コアをデバイスとして初期化する必要があります。

1. OTG_FS_DCFG レジスタの以下のフィールドをプログラムします。
 - デバイススピード
 - 非ゼロ長ステータスOUTハンドシェイク
2. OTG_FS_GINTMSK レジスタをプログラムして、以下の割込みのマスクを解除します。
 - USB リセット
 - エニュメレーション終了
 - アーリーサスペンド
 - USB サスペンド
 - SOF
3. OTG_FS_GCCFG レジスタの VBUSSEN ビットをプログラムして、B デバイスモードでの V_{BUS} 検出を有効にし、DP ライン上でプルアップ抵抗に 5 V を供給します。
4. OTG_FS_GINTSTS の USBRST 割込みを待ちます。これは、この割込みの受信時、約 10ms 続くリセットが USB で検出されたことを示します。

OTG_FS_GINTSTS レジスタの ENUMDNE 割込みを待ちます。この割込みは、USB 上でリセットが終了したことを示します。この割込みを受信したとき、アプリケーションは、OTG_FS_DSTS レジスタを読み出して、エニユメレーション速度を判別し、769 ページの[エニユメレーション完了時のエンドポイント初期化](#)にリストされている手順を実行する必要があります。

この時点で、デバイスは、SOF パケットを受け入れて、コントロールエンドポイント 0 で制御転送を実行する準備ができたことになります。

22.17.4 ホストプログラミングモデル

チャンネルの初期化

アプリケーションが接続されたデバイスと通信するためには、1 つまたは複数のチャンネルを初期化する必要があります。チャンネルを初期化して、有効にするには、アプリケーションは、以下の手順を実行する必要があります。

1. OTG_FS_GINTMSK レジスタをプログラムして、以下の割込みのマスクを解除します。
2. チャンネル割込み
 - OUT トランザクションの非周期的送信 FIFO エンプティ割込み（パケットカウンタフィールドが複数でプログラムされたパイプライントランザクションレベルで動作する場合に適用可能）
 - OUT トランザクションの非周期的送信 FIFO 半エンプティ割込み（パケットカウンタフィールドが複数でプログラムされたパイプライントランザクションレベルで動作する場合に適用可能）
3. OTG_FS_HAINTMSK レジスタをプログラムして、選択されたチャンネルの割込みのマスクを解除します。
4. OTG_FS_HCINTMSK レジスタをプログラムして、ホストチャンネル割込みレジスタで指定された、関心あるトランザクション関連割込みのマスクを解除します。
5. 選択されたチャンネルの OTG_FS_HCTSIZx レジスタを、合計転送サイズ（バイト数）およびショートパケットを含む期待されるパケット数でプログラムします。アプリケーションは、PID フィールドを初期データ PID（最初の OUT トランザクションで使用される PID、または最初の IN トランザクションから期待される PID）でプログラムする必要があります。
6. 選択されたチャンネルの OTG_FS_HCCHARx レジスタを、タイプ、速度、方向など、デバイスのエンドポイントの特性でプログラムします。（チャンネルは、アプリケーションがパケットを送信または受信する準備ができていない場合にのみ、チャンネルイネーブルビットを 1 にセットすることでは有効にできません）。

チャンネルの停止

アプリケーションは、OTG_FS_HCCHARx レジスタの CHDIS および CHENA ビットを 1 にセットすることによって、任意のチャンネルを無効にできます。これにより、OTG_FS ホストは、ポストされたリクエスト（ある場合）をフラッシュして、チャンネル停止割込みを生成できます。アプリケーションは、チャンネルを他のトランザクションに割り当てる前に、OTG_FS_HCINTx の CHH 割込みを待つ必要があります。OTG_FS ホストは、USB 上ですでに転送が開始しているトランザクションに対しては割込みを行いません。

チャンネルを無効にする前に、アプリケーションは、非周期的リクエストキュー（非周期的チャンネルを無効にするとき）、または周期的リクエストキュー（周期的チャンネルを無効にするとき）に使用可能な空きスペースが少なくとも 1 つ以上存在することを確認する必要があります。アプリケーションは、リクエストキューが満杯状態のときには、OTG_FS_HCCHARx レジスタの CHDIS ビットを 1 にセットし、CHENA ビットを 0 にクリアするだけで、ポストされたリクエストをフラッシュできます。

アプリケーションは、以下のいずれかの条件のとき、チャンネルを無効にすることが期待されます。

1. IN または OUT チャンネルで、OTG_FS_HCINTx レジスタの STALL、TXERR、BBERR、または DTERR 割り込みが受信されたとき。アプリケーションは、停止割り込みを受信する前に、同じチャンネルについての他の割り込み (DTERR、Nak、Data、TXERR) を受信できなければなりません。
2. OTG_FS_GINTSTS レジスタの DISCINT (デバイス切断) 割り込みを受信したとき。(アプリケーションは、すべての有効なチャンネルを無効にすることが期待されます)。
3. アプリケーションが正常完了の前に転送を中断したとき。

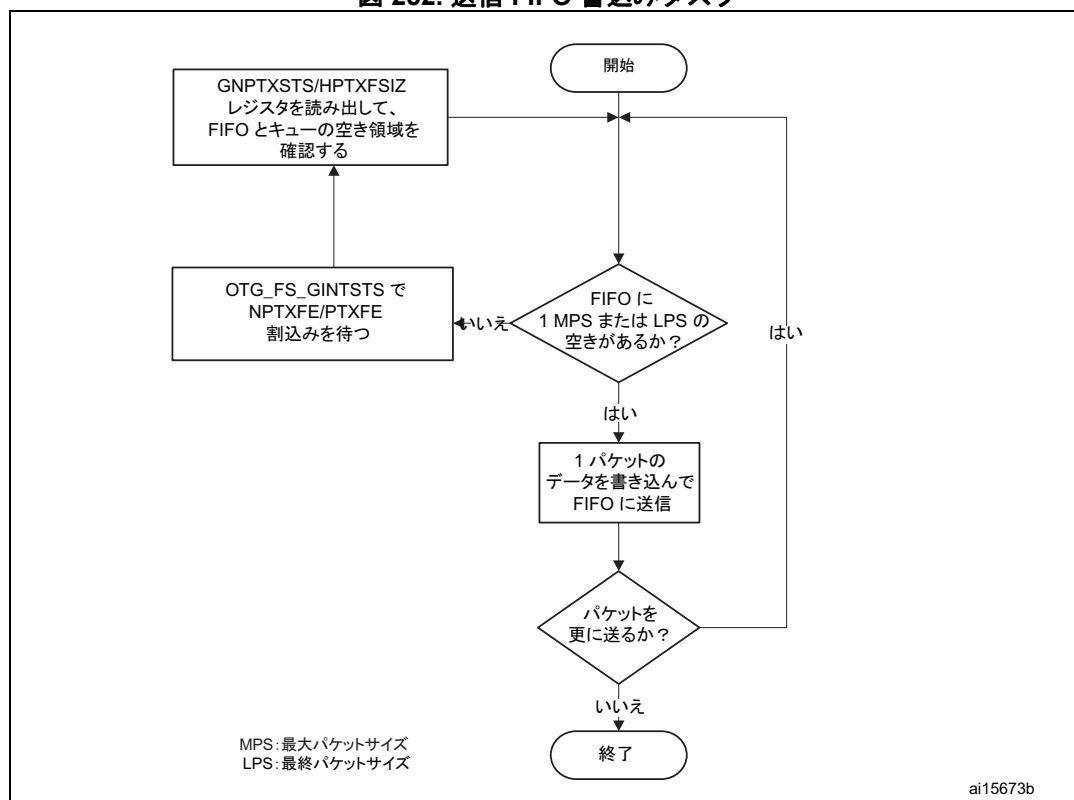
動作モデル

アプリケーションが接続されたデバイスと通信するためには、チャンネルを初期化する必要があります。このセクションでは、さまざまなタイプの USB トランザクションについて、実行しなければならない動作のシーケンスについて説明します。

送信 FIFO への書込み

OTG_FS ホストは、パケットの最後のダブルワードの書込みとともに、周期的/非周期的リクエストキューにエントリ (OUT リクエスト) を自動的に書き込みます。アプリケーションは、送信 FIFO への書込みを開始する前に、周期的/非周期的リクエストキューに少なくとも 1 つ以上の空きスペースがあることを確認しなければなりません。アプリケーションは送信 FIFO に必ずダブルワード単位で書込みを行う必要があります。パケットサイズがダブルワード単位で揃えられていない場合、アプリケーションはパディングを使用する必要があります。OTG_FS ホストは、プログラムされた最大パケットサイズと転送サイズに基づいて、実際のパケットサイズを決定します。

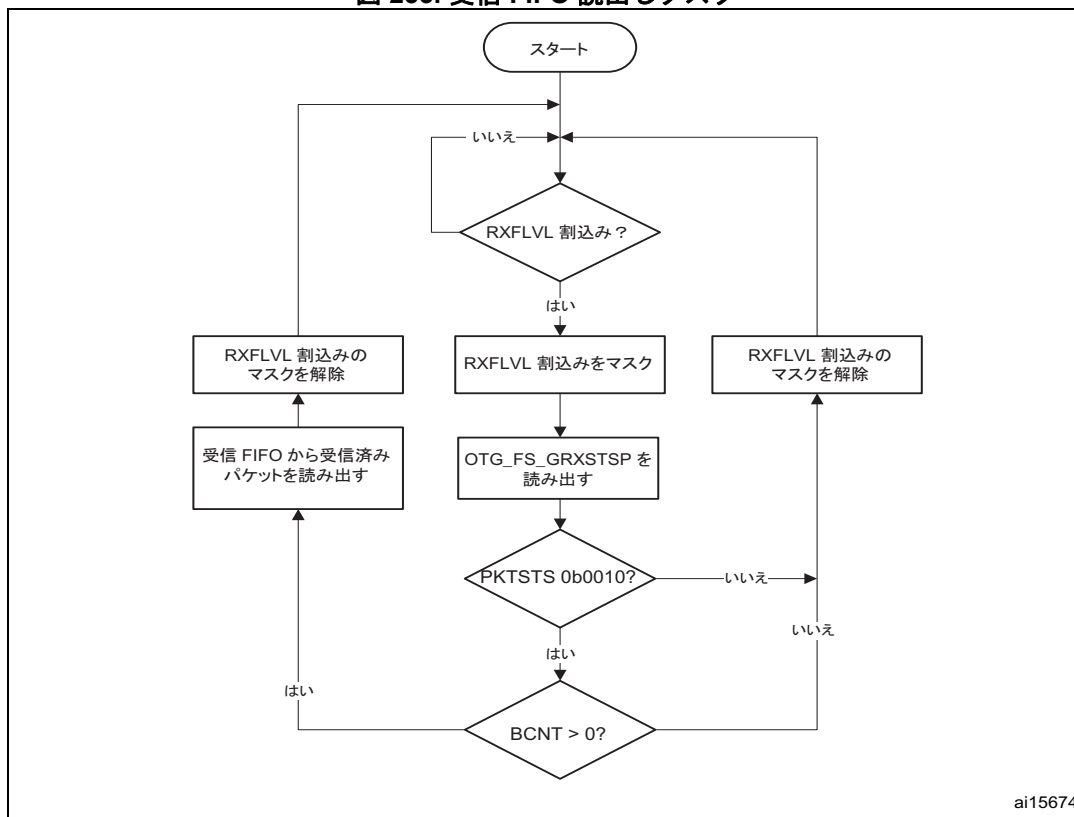
図 252. 送信 FIFO 書込みタスク



- 受信 FIFO の読出し

アプリケーションは、IN データパケット (bx0010) 以外のすべてのパケットステータスを無視しなければなりません。

図 253. 受信 FIFO 読出しタスク



- バルクおよびコントロール OUT/SETUP トランザクション

典型的なバルクまたはコントロール OUT/SETUP パイプライントランザクションレベルの動作を 図 254 に示します。チャンネル 1 (ch_1) に注目してください。2 つのバルク OUT パケットが転送されます。コントロール SETUP トランザクションは同じように動作しますが、1 パケットのみです。ここでは、以下のように仮定されています。

- アプリケーションは、最大パケットサイズの 2 つのパケットを送信しようとしています (転送サイズ = 1,024 バイト)。
- 非周期的送信 FIFO は 2 つのパケットを保持することができます (FS の場合 128 バイト)。
- 非周期的リクエストキューの深さ = 4。

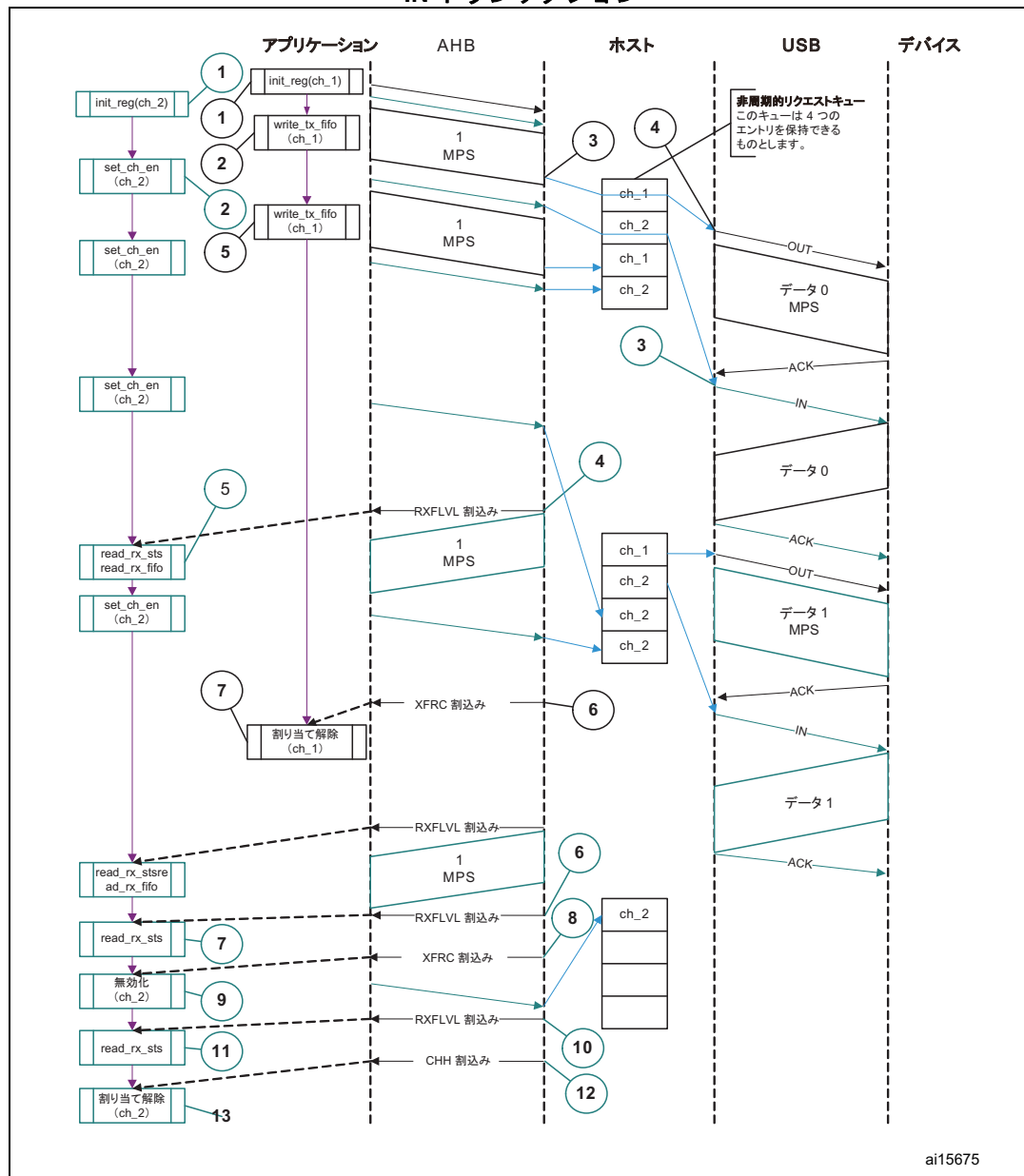
- 通常のバルクおよびコントロール OUT/SETUP 動作

動作シーケンス (チャンネル 1) は、次のとおりです。

- チャンネル 1 を初期化します。
- チャンネル 1 の最初のパケットを書き込みます。
- 最後のワードの書き込みと同時に、コアは、非周期的リクエストキューにエントリを書き込みます。
- 非周期的キューがエンプティでなくなるとすぐに、コアは、現在のフレームで OUT トークンの送信を試みます。
- チャンネル 1 の 2 番目 (最後) のパケットを書き込みます。

- f) コアは、最後のトランザクションが正常に完了すると、XFRC 割込みを生成します。
- g) XFRC 割込みに応答して、他の転送のためにチャネル割り当てを解除します。
- h) 非 ACK 応答の取り扱い

図 254. 通常のバルク/コントロール OUT/SETUP およびバルク/コントロール IN トランザクション



バルクおよびコントロール OUT/SETUP トランザクションのためのチャネル固有の割込みサービスルーチンを、以下のコードサンプルで示します。

- バルク／コントロール OUT/SETUP とバルクまたはコントロール IN トランザクションの割り込みサービスルーチン

a) バルク／コントロール OUT/SETUP

```
Unmask (NAK/TXERR/STALL/XFRC)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    De-allocate Channel
}
else if (STALL)
{
    Transfer Done = 1
    Unmask CHH
    Disable Channel
}
else if (NAK or TXERR )
{
    Rewind Buffer Pointers
    Unmask CHH
    Disable Channel
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
    }
    else
    {
        Reset Error Count
    }
}
else if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel
    }
}
else if (ACK)
{
    Reset Error Count
    Mask ACK
}
```

アプリケーションは、送信 FIFO およびリクエストキューの中に使用可能なスペースがある場合、データパケットを送信 FIFO に書き込むことが期待されます。アプリケーションは、OTG_FS_GINTSTS レジスタの NPTXFE 割込みを使用して、送信 FIFO のスペースを見つけることができます。

b) バルク／コントロール IN

```
Unmask (TXERR/XFRC/BBERR/STALL/DTERR)
if (XFRC)
{
    Reset Error Count
    Unmask CHH
    Disable Channel
    Reset Error Count
    Mask ACK
}
else if (TXERR or BBERR or STALL)
{
    Unmask CHH
    Disable Channel
    if (TXERR)
    {
        Increment Error Count
        Unmask ACK
    }
}
else if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel
    }
}
else if (ACK)
{
    Reset Error Count
    Mask ACK
}
else if (DTERR)
{
    Reset Error Count
}
```

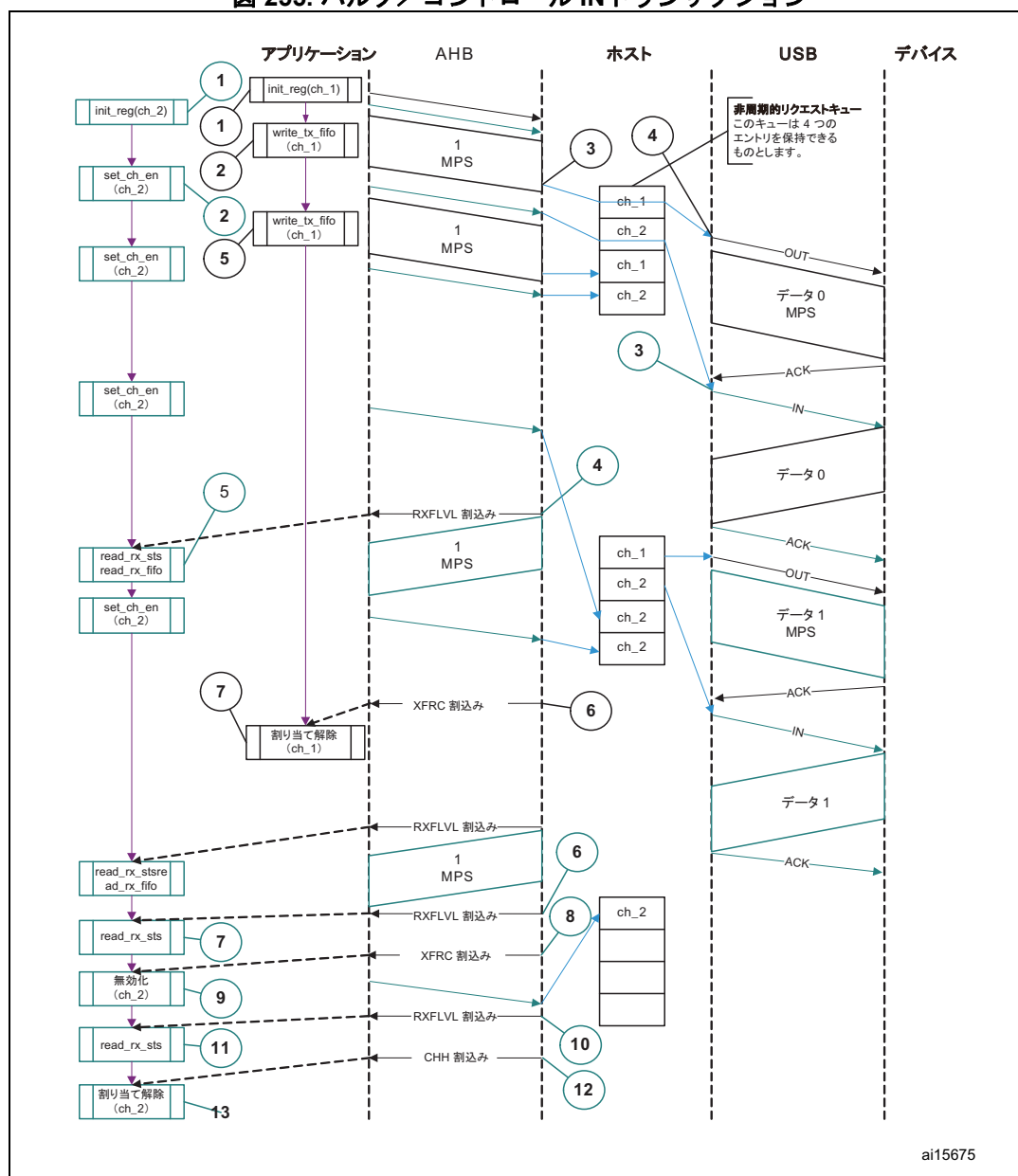
アプリケーションは、リクエストキューのスペースが使用可能なときには、XFRC 割込みが受信されるまで、リクエストを書き込むことが期待されます。

- **バルクおよびコントロール IN トランザクション**

典型的なバルクまたはコントロール IN パイプライントランザクションレベルの動作を [図 255](#) に示します。チャンネル 2 (ch_2) に注目してください。ここでは、以下のように仮定されています。

- アプリケーションは、最大パケットサイズ (転送サイズ = 1,024 バイト) の 2 つのパケットを受信しようとしています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットと、パケットあたり 2 つのステータスワード (FS の場合 72 バイト) を収容することができます。
- 非周期的リクエストキューの深さ = 4。

図 255. バルク/コントロール IN トランザクション



動作シーケンスは、次のとおりです。

- チャンネル 2 を初期化します。
- HCCHAR2 の CHENA ビットをセットして、IN リクエストを非周期的リクエストキューに書き込みます。
- コアは、現在の OUT トランザクションの完了後、IN トークンの送信を試みます。
- コアは、受信したパケットが受信 FIFO に書き込まれると、RXFLVL 割込みを生成します。
- RXFLVL 割込みに応答して、RXFLVL 割込みをマスクし、受信パケットステータスを読み出して受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。続いて、RXFLVL 割込みのマスクを解除します。
- コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割込みを生成します。

- g) アプリケーションは、受信パケットステータスを読み出して、受信パケットステータスが IN データパケットでないとき (GRXSTSR ≠ 0b0010 の PKTSTS) には無視する必要があります。
 - h) コアは、受信パケットステータスを読み出されると、XFRC 割込みを生成します。
 - i) XFRC 割込みに応答して、チャンネルを無効にし、それ以降のリクエストについて、OTG_FS_HCCHAR2 レジスタへの書き込みを停止します。コアは、OTG_FS_HCCHAR2 に書き込まれると、非周期的リクエストキューにチャンネルディセーブルリクエストを書き込みます。
 - j) コアは、停止ステータスが受信 FIFO に書き込まれると、RXFLVL 割込みを生成します。
 - k) 受信パケットステータスを読み出して、それを無視します。
 - l) コアは、停止ステータスが受信 FIFO からポップされるとすぐに、CHH 割込みを生成します。
 - m) CHH 割込みに応答して、他の転送のためにチャンネル割り当てを解除します。
 - n) 非 ACK 応答の取り扱い
- **コントロールトランザクション**

コントロール転送のセットアップ、データ、およびステータスステージは 3 つの個別の転送として実行されなければなりません。セットアップ、データ、およびステータスステージの OUT トランザクションは、すでに説明したバルク OUT トランザクションと同様に実行されます。データまたはステータスステージの IN トランザクションは、すでに説明したバルク IN トランザクションと同様に実行されます。3 つのステージすべてについて、アプリケーションは OTG_FS_HCCHAR1 レジスタの EPTYP フィールドを Control に設定することが期待されます。セットアップステージでは、アプリケーションは OTG_FS_HCTSIZ1 レジスタの PID フィールドを SETUP に設定することが期待されます。
 - **インタラプトOUT トランザクション**

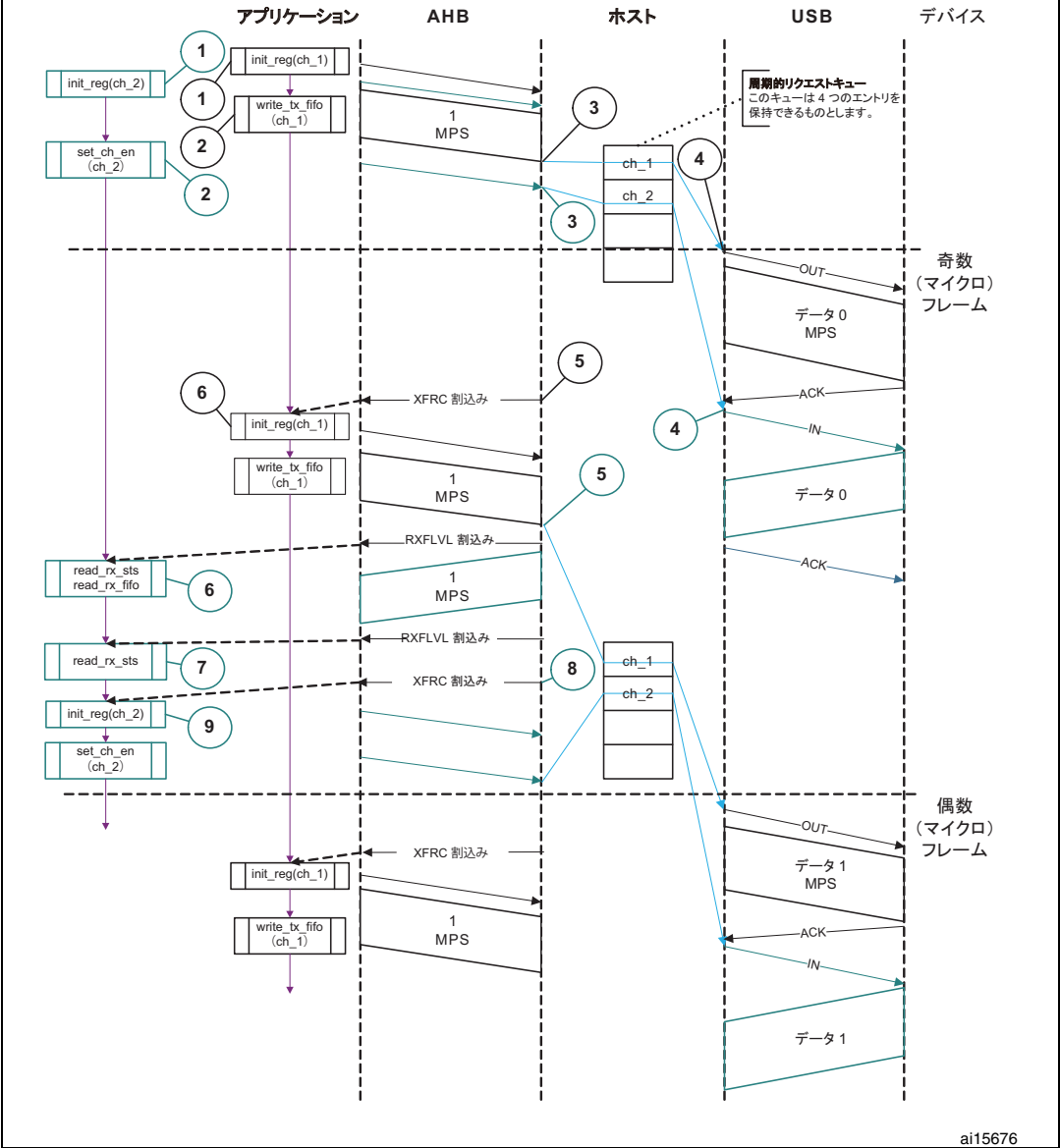
典型的なインタラプト OUT 動作を [図 256](#) に示します。ここでは、以下のように仮定されています。

 - アプリケーションは、奇数フレーム (転送サイズ = 1 024 バイト) から始めて、フレームごとに 1 パケット (最大 1 パケットサイズ) の送信を試みています。
 - 周期的送信 FIFO は 1 パケット (1 KB) を保持することができます。
 - 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

 - a) チャンネル 1 を初期化し、有効にします。アプリケーションは OTG_FS_HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
 - b) チャンネル 1 の最初のパケットを書き込みます。
 - c) 各パケットの最後のワードの書き込みとともに、OTG_FS ホストは、周期的リクエストキューにエントリを書き込みます。
 - d) OTG_FS ホストは、次の (奇数) フレームで OUT トークンの送信を試みます。
 - e) OTG_FS ホストは、最後のパケットが正常に送信されると、XFRC 割込みを生成します。
 - f) XFRC 割込みに応答して、次の転送のためにチャンネルを再初期化します。

図 256. 通常インタラプト OUT/IN トランザクション



• インタラプト OUT/IN トランザクションのための割込みサービスルーチン

a) インタラプト OUT

```
Unmask (NAK/TXERR/STALL/XFRM/FRMOR)
if (XFRM)
{
    Reset Error Count
    Mask ACK
    De-allocate Channel
}
else
{
    if (STALL or FRMOR)
    {
        Mask ACK
        Unmask CHH
    }
}
```

```

Disable Channel
if (STALL)
{
    Transfer Done = 1
}
}
else
if (NAK or TXERR)
{
    Rewind Buffer Pointers
    Reset Error Count
    Mask ACK
    Unmask CHH
    Disable Channel
}
else
if (CHH)
{
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
        De-allocate Channel
    }
    else
    {
        Re-initialize Channel (in next b_interval - 1 Frame)
    }
}
else
if (ACK)
{
    Reset Error Count
    Mask ACK
}

```

アプリケーションは、OTG_FS_GINTSTS レジスタの NPTXFE 割込みを使用して、送信 FIFO のスペースを見つけます。

b) インタラプト IN

```

Unmask (NAK/TXERR/XFRC/BBERR/STALL/FRMOR/DTERR)
if (XFRC)
{
    Reset Error Count
    Mask ACK
    if (OTG_FS_HCTSIZx.PKTCNT == 0)
    {
        De-allocate Channel
    }
}
else
{
    Transfer Done = 1
    Unmask CHH
    Disable Channel
}

```

```
    }
  }
else
  if (STALL or FRMOR or NAK or DTERR or BBERR)
  {
    Mask ACK
    Unmask CHH
    Disable Channel
    if (STALL or BBERR)
    {
      Reset Error Count
      Transfer Done = 1
    }
    else
      if (!FRMOR)
      {
        Reset Error Count
      }
  }
else
  if (TXERR)
  {
    Increment Error Count
    Unmask ACK
    Unmask CHH
    Disable Channel
  }
else
  if (CHH)
  {
    Mask CHH
    if (Transfer Done or (Error_count == 3))
    {
      De-allocate Channel
    }
    else
      Re-initialize Channel (in next b_interval - 1 /Frame)
  }
}
else
  if (ACK)
  {
    Reset Error Count
    Mask ACK
  }
```


- **インタラプト IN トランザクション**

ここでは、以下のように仮定されています。

- アプリケーションは、奇数フレーム（転送サイズ = 1024 バイト）から始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の受信を試みています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットとパケットあたり 2 つのステータスワード（1,031 バイト）を保持することができます。
- 周期的リクエストキューの深さ = 4。

- **通常のインタラプト IN 動作**

動作シーケンスは、次のとおりです。

- チャンネル 2 を初期化します。アプリケーションは OTG_FS_HCCHAR2 レジスタの ODDFRM ビットをセットする必要があります。
- OTG_FS_HCCHAR2 の CHENA ビットをセットして IN リクエストを周期的リクエストキューを書き込みます。
- OTG_FS ホストは、CHENA ビットがセットされた OTG_FS_HCCHAR2 レジスタの書き込みごとに、周期的リクエストキューに IN リクエストを書き込みます。
- OTG_FS ホストは、次の（奇数）フレームで IN トークンの送信を試みます。
- IN パケットが受信され、受信 FIFO に書き込まれると、OTG_FS ホストは RXFLVL 割込みを生成します。
- RXFLVL 割込みに応答して、受信パケットステータスを読み出して、受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。アプリケーションは、受信 FIFO を読み出す前に RXFLVL 割込みをマスクし、すべてのパケットを読み出した後にマスクを解除しなければなりません。
- コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割込みを生成します。アプリケーションは、受信パケットステータスを読み出して、受信パケットステータスが IN データパケットでないとき（GRXSTSR ≠ 0b0010 の PKTSTS）には無視する必要があります。
- コアは、受信パケットステータスが読み出されると XFRC 割込みを生成します。
- XFRC 割込みに応答して OTG_FS_HCTSIZ2 の PKTCNT フィールドを読み出します。OTG_FS_HCTSIZ2 の PKTCNT ビットが 0 でない場合は、次の転送（ある場合）のためにチャンネルを再初期化する前に、チャンネルを無効にします。OTG_FS_HCTSIZ2 の PKTCNT ビットが 0 の場合、次の転送のためにチャンネルを再初期化します。今度は、アプリケーションは OTG_FS_HCCHAR2 レジスタの ODDFRM ビットをリセットする必要があります。

- **アイソクロナス OUT トランザクション**

典型的なアイソクロナス OUT 動作を [図 257](#) に示します。ここでは、以下のように仮定されています。

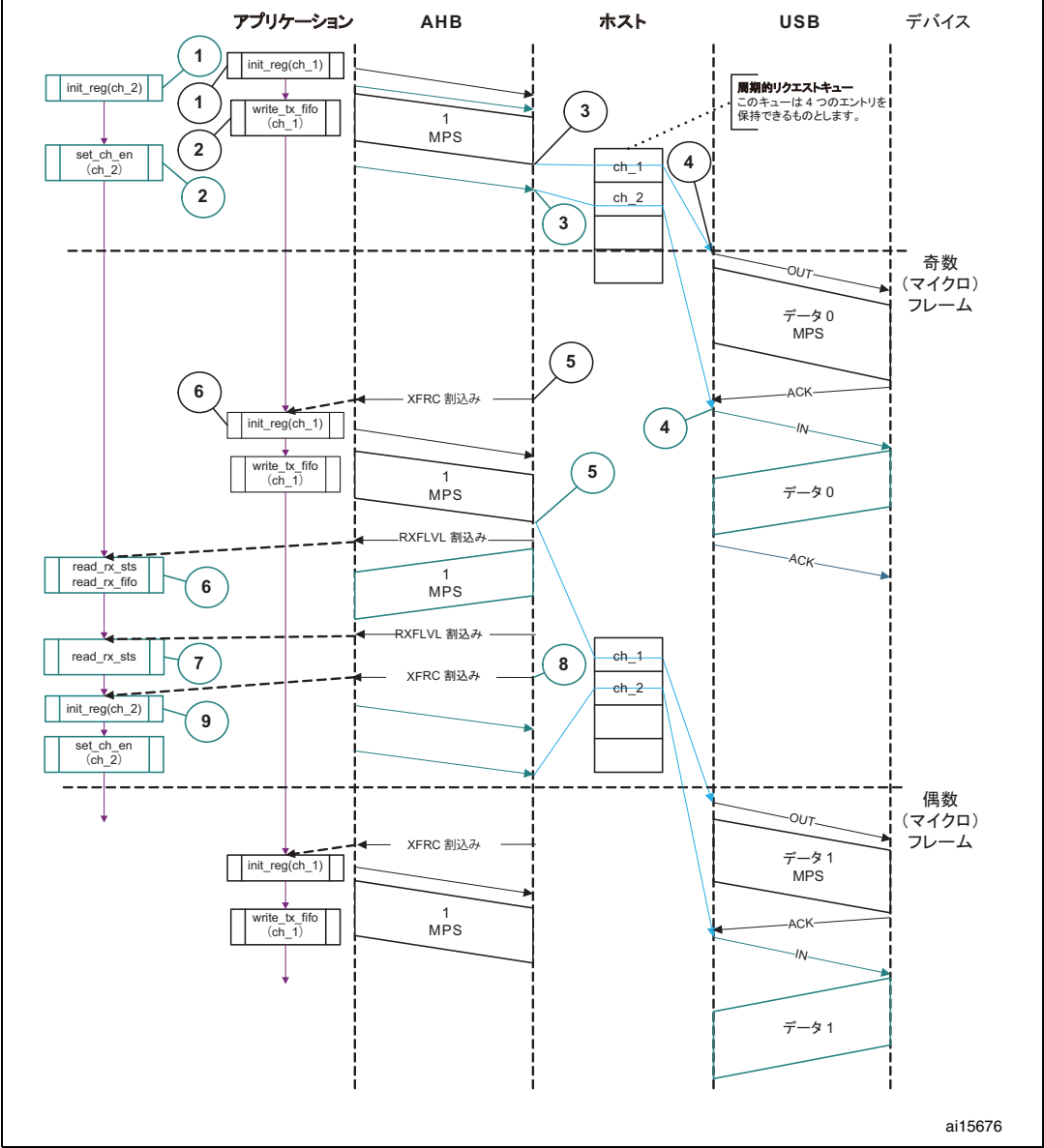
- アプリケーションは、奇数フレームから始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の送信を試みています。（転送サイズ = 1,024 バイト）。
- 周期的送信 FIFO は 1 パケット（1 KB）を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

- チャンネル 1 を初期化し、有効にします。アプリケーションは OTG_FS_HCCHAR1 レジスタの ODDFRM ビットをセットする必要があります。
- チャンネル 1 の最初のパケットを書き込みます。
- 各パケットの最後のワードの書き込みとともに、OTG_FS ホストは、周期的リクエストキューにエントリを書き込みます。
- OTG_FS ホストは、次のフレーム（奇数）で OUT トークンの送信を試みます。

- e) OTG_FS ホストは、最後のパケットが正常に送信されると、XFRC 割り込みを生成します。
- f) XFRC 割り込みに応答して、次の転送のためにチャンネルを再初期化します。
- g) 非 ACK 応答の取り扱い

図 257. 通常のアイソクロナス OUT/IN トランザクション



- アイソクロナス OUT/IN トランザクションのための割り込みサービスルーチン
コードサンプル : アイソクロナス OUT

```
Unmask (FRMOR/XFRC)
if (XFRC)
{
    De-allocate Channel
}
else
    if (FRMOR)
```

```
    {
        Unmask CHH
        Disable Channel
    }
else
    if (CHH)
    {
        Mask CHH
        De-allocate Channel
    }

コードサンプル : アイソクロナスIN
Unmask (TXERR/XFRC/FRMOR/BBERR)
if (XFRC or FRMOR)
{
    if (XFRC and (OTG_FS_HCTSIZx.PKTCNT == 0))
    {
        Reset Error Count
        De-allocate Channel
    }
else
    {
        Unmask CHH
        Disable Channel
    }
}
else
    if (TXERR or BBERR)
    {
        Increment Error Count
        Unmask CHH
        Disable Channel
    }
else
    if (CHH)
    {
        Mask CHH
        if (Transfer Done or (Error_count == 3))
        {
            De-allocate Channel
        }
        else
        {
            Re-initialize Channel
        }
    }
}
```

- アイソクロナス IN トランザクション

ここでは、以下のように仮定されています。

- アプリケーションは、次の奇数フレーム（転送サイズ = 1,024 バイト）から始めて、フレームごとに 1 パケット（最大 1 パケットサイズ）の受信を試みています。
- 受信 FIFO は、少なくとも 1 つの最大パケットサイズのパケットとパケットあたり 2 つのステータスワード（1,031 バイト）を保持することができます。
- 周期的リクエストキューの深さ = 4。

動作シーケンスは、次のとおりです。

- チャンネル 2 を初期化します。アプリケーションは OTG_FS_HCCHAR2 レジスタの ODDFRM ビットをセットする必要があります。
- OTG_FS_HCCHAR2 の CHENA ビットをセットして IN リクエストを周期的リクエストキューを書き込みます。
- OTG_FS ホストは、CHENA ビットがセットされた OTG_FS_HCCHAR2 レジスタの書き込みごとに、周期的リクエストキューに IN リクエストを書き込みます。
- OTG_FS ホストは、次の奇数フレームで IN トークンの送信を試みます。
- IN パケットが受信され、受信 FIFO に書き込まれると、OTG_FS ホストは RXFLVL 割込みを生成します。
- RXFLVL 割込みに応答して、受信パケットステータスを読み出して、受信バイト数を調べた後、それに応じて受信 FIFO を読み出します。アプリケーションは、受信 FIFO を読み出す前に RXFLVL 割込みをマスクし、すべてのパケットを読み出した後でマスクを解除しなければなりません。
- コアは、受信 FIFO 内の転送完了ステータスエントリに対して RXFLVL 割込みを生成します。今度は、アプリケーションは受信パケットステータスを読み出して、それが IN データパケット（OTG_FS_GRXSTSR の PKTSTS ビット ≠ 0b0010）でなかった場合、無視する必要があります。
- コアは、受信パケットステータスが読み出されると XFRC 割込みを生成します。
- XFRC 割込みに応答して OTG_FS_HCTSIZ2 の PKTCNT フィールドを読み出します。OTG_FS_HCTSIZ2 の PKTCNT ≠ 0 の場合、次の転送（ある場合）のためにチャンネルを再初期化する前に、チャンネルを無効にします。OTG_FS_HCTSIZ2 の PKTCNT = 0 の場合、次の転送のためにチャンネルを再初期化します。今度は、アプリケーションは OTG_FS_HCCHAR2 レジスタの ODDFRM ビットをリセットする必要があります。

- キューの深さの選択

周期的および非周期的リクエストキューの深さは、アクセスされる周期的/非周期的エンドポイントの数に合うように注意して選択してください。

非周期的リクエストキューの深さは、非周期的転送の性能に影響を与えます。キューが深いほど（FIFO のサイズが十分であれば）、コアは非周期的転送をより多くパイプライン化できます。キューのサイズが小さいと、コアはキューのスペースが空いたときしか新しいリクエストを入れることができません。

コアの周期的リクエストキューの深さは、周期的転送をスケジュールどおりに実行するために不可欠です。周期的キューの深さは、マイクロフレーム内でスケジュールされた周期的転送の数に基づいて選択してください。周期的リクエストキューの深さがマイクロフレーム内でスケジュールされた周期的転送の数より小さい場合、フレームオーバーラン条件が発生します。

- バブル条件の取り扱い

OTG_FS コントローラは、パケットバブルとポートバブルの 2 つのバブルを処理します。パケットバブルは、デバイスがチャンネルの最大パケットサイズよりも多くのデータを送信した場合に発生します。ポートバブルは、コアが EOF2（フレーム 2 の終わりの SOF にきわめて近い）でデバイスからデータを受信し続けると発生します。

OTG_FS コントローラがパケットバブルを検出すると、Rx バッファへのデータの書き込みを停止して、パケットの終わり (EOP) を待ちます。EOP を検出すると コントローラは Rx バッファにすでに書き込まれたデータを一掃して、アプリケーションに対するバブル割込みを生成します。

OTG_FS コントローラがポートバブルを検出すると、RxFIFO をフラッシュして、ポートを無効にします。コアは、ポート無効割込み (OTG_FS_GINTSTS の HPRTINT、OTG_FS_HPRT の PENCHNG) を生成します。この割込みを受信すると、アプリケーションは、OTG_FS_HPRT の POCA ビットをチェックして、これが過電流条件 (ポート無効割込みのもう 1 つの原因) によるものではないことを確認してから、ソフトリセットを行う必要があります。コアは、ポートバブル条件を検出した後は、それ以上トークンを送信しません。

22.17.5 デバイスプログラミングモデル

USB リセット時のエンドポイントの初期化

1. すべての OUT エンドポイントの NAK ビットをセットします。
 - OTG_FS_DOEPCCTLx の SNAK = 1 (すべての OUT エンドポイントについて)
2. 以下の割込みビットのマスクを解除します。
 - OTG_FS_DAINMSK の INEP0 = 1 (制御 0 IN エンドポイント)
 - OTG_FS_DAINMSK の OUTEP0 = 1 (制御 0 OUT エンドポイント)
 - DOEPMASK の STUP = 1
 - DOEPMASK の XFRC = 1
 - DIEPMASK の XFRC = 1
 - DIEPMASK の TOC = 1
3. 各 FIFO の DATA FIFO RAM をセットアップします。
 - OTG_FS_GRXFSIZ レジスタをプログラムして、コントロール OUT データとセットアップデータを受信できるようにします。閾値の設定が無効の場合、これは、少なくとも、コントロールエンドポイント 0 の最大パケットサイズ 1 個分 + 2 ワード (制御 OUT データパケットのステータス用) + 10 ワード (セットアップパケット用) でなければなりません。
 - OTG_FS_TX0FSIZ レジスタを、コントロール IN データを送信できるようにプログラムします (選択された FIFO 番号に応じて)。これは、少なくとも、コントロールエンドポイント 0 の最大パケットサイズ 1 個分以上でなければなりません。
4. SETUP パケットを受信するために、コントロール OUT エンドポイント 0 のエンドポイント固有レジスタの以下のフィールドをプログラムします。
 - OTG_FS_DOEPTSIZ0 の STUPCNT = 3 (最大 3 つの連続 SETUP パケットを受信するため)

この時点で SETUP パケットを受信するために必要なすべての初期化が終了したことになります。

エニューメレーション完了時のエンドポイント初期化

1. エニューメレーション終了割込み (OTG_FS_GINTSTS の ENUMDNE ビット) 時には、OTG_FS_DSTS レジスタを読み出して、エニューメレーション速度を決めます。
2. OTG_FS_DIEPCTL0 の MPSIZ フィールドをプログラムして、最大パケットサイズを設定します。このステップでは、コントロールエンドポイント 0 を設定します。コントロールエンドポイントの最大パケットサイズは、エニューメレーションスピードに依存します。

この時点で、デバイスは SOF パケットを受信する準備ができ、制御エンドポイント 0 で制御転送を行うように設定されたことになります。

SetAddress コマンド受信時のエンドポイントの初期化

このセクションでは SETUP パケットで SetAddress コマンドを受信したときにアプリケーションが行わなければならないことについて説明します。

1. SetAddress コマンドで受信したデバイスアドレスで OTG_FS_DCFG レジスタをプログラムします。
1. ステータス IN パケットを送信するように、コアをプログラムします。

SetConfiguration/SetInterface コマンド受信時のエンドポイントの初期化

このセクションでは SETUP パケットで SetConfiguration または SetInterface コマンドを受信したときにアプリケーションが行わなければならないことについて説明します。

1. SetConfiguration コマンドを受信したとき、アプリケーションは、新しい設定で有効なエンドポイントの特性で、エンドポイントのレジスタをプログラムする必要があります。
2. SetInterface コマンドを受信したとき、アプリケーションは、このコマンドの影響を受けるエンドポイントのレジスタをプログラムする必要があります。
3. 前の設定または代替設定ではアクティブであったエンドポイントが、新しい設定または代替設定では無効なことがあります。これらの無効なエンドポイントは、機能を停止させる必要があります。
4. 有効な各エンドポイントの割込みのマスクを解除して、OTG_FS_DAINMSK レジスタですべての無効なエンドポイントに対する割込みをマスクします。
5. 各 FIFO のデータ FIFO RAM をセットアップします。
6. 必要なすべてのエンドポイントを設定した後、アプリケーションは、ステータス IN パケットを送信するようにコアをプログラムする必要があります。

この時点で、デバイスのコアは、任意のタイプのデータパケットを送信/受信できるように設定されたことになります。

エンドポイントのアクティブ化

このセクションでは、デバイスエンドポイントをアクティブ化するか、既存のデバイスエンドポイントを新しいタイプに設定するために必要な手順について説明します。

1. 必要なエンドポイントの特性を、OTG_FS_DIEPCTLx レジスタ (IN または双方向エンドポイントの場合)、または OTG_FS_DOEPCTLx レジスタ (OUT または双方向エンドポイントの場合) の以下のフィールドにプログラムします。
 - 最大パケットサイズ
 - USB アクティブエンドポイント = 1
 - エンドポイント開始データトグル (インタラプトおよびバルクエンドポイントの場合)
 - エンドポイントタイプ
 - TxFIFO 番号
2. エンドポイントがアクティブ化されると、コアは、そのエンドポイントに宛てたトークンのデコードを開始し、そのエンドポイントで受信された有効な各トークンについて有効なハンドシェイクを送出します。

エンドポイントの機能停止

このセクションでは、既存のエンドポイントの機能を停止させるために必要な手順について説明します。

1. 機能を停止するエンドポイントで、OTG_FS_DIEPCTLx レジスタ (IN または双方向エンドポイントの場合)、または OTG_FS_DOEPCTLx レジスタ (OUT または双方向エンドポイントの場合) の USB 有効エンドポイントビットをクリアします。
2. エンドポイントが非アクティブな状態になると、コアは、そのエンドポイントにアドレス指定されたトークンを無視するので USB 上でタイムアウトが発生します。

注： アプリケーションは、以下の条件を満たすように、デバイスコアがトラフィックを処理できるようにセットアップする必要があります。
OTG_FS_GINTMSK レジスタの NPTXFEM および RXFLVLM をクリアする必要があります。

22.17.6 動作モデル

SETUP および OUT データ転送

このセクションでは、データ OUT 転送および SETUP トランザクション時の内部データフローとアプリケーションレベルの動作について説明します。

• パケットの読出し

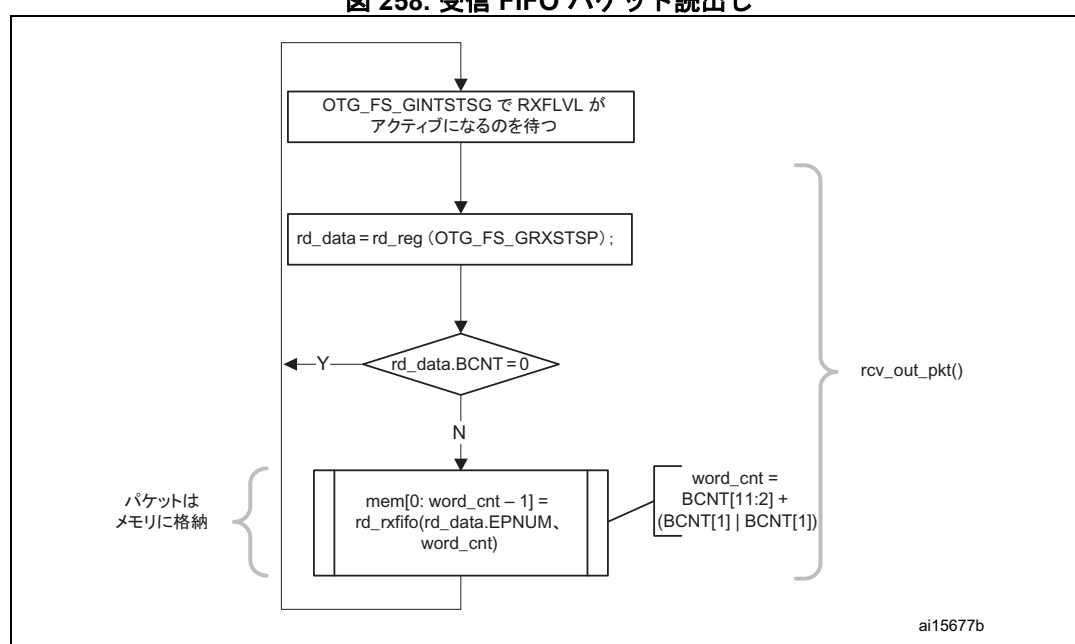
このセクションでは、受信 FIFO からパケット (OUT データおよび SETUP パケット) を読み出す方法について説明します。

1. RXFLVL 割込み (OTG_FS_GINTSTS レジスタ) を受信すると、アプリケーションは、受信ステータスポップレジスタ (OTG_FS_GRXSTSP) を読み出す必要があります。
2. アプリケーションは、RXFLVL = 0 (OTG_FS_GINTMSK レジスタ) を書き込むことによって、受信 FIFO からのパケット読出しが終了するまで、RXFLVL 割込み (OTG_FS_GINTSTS レジスタ) をマスクすることができます。
3. 受信パケットのバイトカウントが 0 でない場合、データのバイトカウントは受信データ FIFO からポップされ、メモリに格納されます。受信パケットのバイトカウントが 0 の場合、受信データ FIFO からデータはポップされません。
4. 受信 FIFO のパケットステータスの読出し結果は、以下のいずれかを示します。
 - a) グローバル OUT NAK のパターン
PKTSTS = グローバル OUT NAK、BCNT = 0x000、EPNUM = 無視 (0x0)、DPID = 無視 (0b00)。
これらのデータは、グローバル OUT NAK ビットが有効になっていることを示します。
 - b) SETUP パケットのパターン
PKTSTS = SETUP、BCNT = 0x008、EPNUM = 制御 EP 番号、DPID = D0。これらのデータは、指定されたエンドポイントの SETUP パケットを受信 FIFO から読み出せることを示します。
 - c) セットアップステージ終了パターン
PKTSTS = セットアップステージ終了、BCNT = 0x0、EPNUM = コントロール EP 番号、DPID = 無視 (0b00)。
これらのデータは、指定されたエンドポイントのセットアップステージが完了し、データステージが開始されたことを示します。このエントリが受信 FIFO からポップされた後、コアは、指定されたコントロール OUT エンドポイントでセットアップ割込みをアサートします。
 - d) データ OUT パケットのパターン
PKTSTS = データ OUT、BCNT = 受信データ OUT パケットのサイズ ($0 \leq BCNT \leq 1024$)、EPNUM = パケットを受信した EPNUM、DPID = 実際のデータ PID。

- e) データ転送完了パターン
 PKTSTS = データ OUT 転送終了、BCNT = 0x0、EPNUM = データ転送が完了した OUT EP 番号、DPID = 無視 (0b00)。
 これらのデータは、指定された OUT エンドポイントの OUT データ転送が完了したことを示します。このエントリが受信 FIFO からポップされた後、コアは、指定された OUT エンドポイントで転送完了割込みをアサートします。
5. データペイロードが受信 FIFO からポップされた後、RXFLVL 割込み (OTG_FS_GINTSTS) がマスク解除されなければなりません。
6. 手順 1~5 は、アプリケーションが OTG_FS_GINTSTS レジスタの RXFLVL による割込みラインのアサーションを検出するたびに繰り返されます。空の受信 FIFO を読み出すと、定義されていないコア動作を引き起こすことがあります。

図 258 上記の手順のフローチャートです。

図 258. 受信 FIFO パケット読出し



SETUP トランザクション

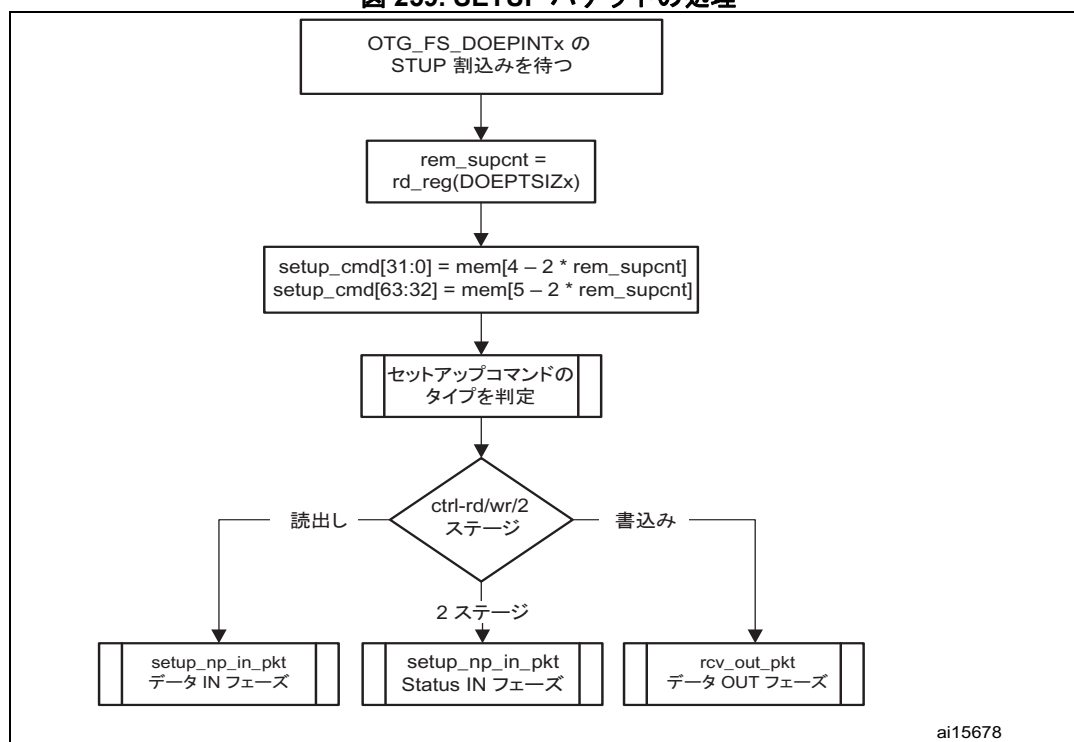
このセクションでは、コアが SETUP パケットを処理する方法と、SETUP トランザクションを処理するアプリケーションのシーケンスについて説明します。

アプリケーションの要件

- SETUP パケットを受信するには、制御 OUT エンドポイントの STUPCNT フィールド (OTG_FS_DOEPTSIz レジスタ) が、ゼロでない値にプログラムされなければなりません。アプリケーションが STUPCNT フィールドをゼロでない値にプログラムすると、NAK のステータスと OTG_FS_DOEPCTLx レジスタの EOENA ビットの設定に関係なく、コアは、SETUP パケットを受信して、受信 FIFO に書き込みます。STUPCNT フィールドは、コントロールエンドポイントが SETUP パケットを受信するたびにデクリメントされます。SETUP パケットを受信する前に、STUPCNT フィールドが適切な値にプログラムされていなかった場合、コアは SETUP パケットを受信し、STUPCNT フィールドをデクリメントしますが、アプリケーションはコントロール転送のセットアップステージで受信した SETUP パケットの正しい数を判定できないことがあります。
- OTG_FS_DOEPTSIz レジスタの STUPCNT = 3

2. コントロールエンドポイントで 3 つまでの SETUP パケットを受信するためには、アプリケーションは、常に受信データ FIFO に余分のスペースを割り当てておく必要があります。
 - 確保すべきスペースは 10 ワード分です。最初の SETUP パケット用に 3 ワードが必要であり、セットアップステージ終了ワードのために 1 ワード、すべてのコントロールエンドポイントにわたって 2 つの余分な SETUP パケットを格納するために 6 ワードが必要です。
 - 8 バイトの SETUP データと 4 バイトの SETUP ステータス(セットアップパケットパターン) を格納するには、SETUP パケットあたり 3 ワードが必要です。コアは、このスペースを受信データの中に確保します。
 - FIFO は SETUP データ書き込み専用であり、このスペースをデータパケットのために使うことはありません。
3. アプリケーションは、受信 FIFO から SETUP パケットの 2 ワードを読み出す必要があります。
4. アプリケーションは受信 FIFO からセットアップステージ終了ワードを読み出して、破棄する必要があります。
- **内部データフロー**
5. SETUP パケットが受信されると、コアは受信 FIFO 内の空きスペースをチェックせずに、また、エンドポイントの NAK および STALL ビットの設定に関係なく、受信したデータを受信 FIFO に書き込みます。
 - コアは、SETUP パケットが受信されたコントロール IN/OUT エンドポイントの IN NAK および OUT NAK ビットを内部でセットします。
6. USB 上で受信された各 SETUP パケットについて、3 ワードのデータが受信 FIFO に書き込まれ、STUPCNT フィールドが 1 ずつデクリメントされます。
 - 最初のワードは、コアが内部的に使用する制御情報を含みます。
 - 2 番目のワードは、SETUP コマンドの最初の 4 バイトを含みます。
 - 3 番目のワードは、SETUP コマンドの最後の 4 バイトを含みます。
7. セットアップステージがデータ IN/OUT ステージに変わると、コアは、エントリ(セットアップステージ終了ワード)を受信 FIFO に書き込んで、セットアップステージの完了を示します。
8. AHB 側では、SETUP パケットはアプリケーションによってエンプティにされます。
9. アプリケーションが受信 FIFO からセットアップステージ終了ワードをポップすると、コアは、STUP 割込み (OTG_FS_DOEPINTx) でアプリケーションに割り込んで、受信した SETUP パケットを処理できることを示します。
 - コアは、コントロール OUT エンドポイントのエンドポイントイネーブルビットをクリアします。
- **アプリケーションのプログラミングシーケンス :**
1. OTG_FS_DOEPTSIZEx レジスタをプログラムします。
 - STUPCNT = 3
2. RXFLVL 割込み (OTG_FS_GINTSTS) を待ち、受信 FIFO からデータパケットをエンプティにします。
3. STUP 割込み (OTG_FS_DOEPINTx) のアサートは、SETUP データ転送が正常に完了したことを示します。
 - この割込み時、アプリケーションは、OTG_FS_DOEPTSIZEx レジスタを読み出して、受信した SETUP パケットの数を確認し、最後に受信した SETUP パケットを処理する必要があります。

図 259. SETUP パケットの処理



• 3 つを超える連続 SETUP パケットの処理

USB 2.0 仕様に従い、通常 SETUP パケットエラー時にはホストは同じエンドポイントに対して 3 つを超える連続 SETUP パケットを送信しません。ただし USB 2.0 仕様では、ホストが同じエンドポイントに送信できる連続 SETUP パケットの数を制限してはいません。この条件が発生すると、OTG_FS コントローラは、割込み (OTG_FS_DOEPINTx の B2BSTUP) を生成します。

• グローバル OUT NAK の設定

内部データフロー

1. アプリケーションがグローバル OUT NAK (OTG_FS_DCTL の SGONAK ビット) をセットすると、コアは、SETUP パケットを除き、受信 FIFO へのデータの書き込みを停止します。受信 FIFO の使用可能なスペースの有無に関係なく、非アイソクロナス OUT トークンは NAK ハンドシェイクレスポンスを受信し、コアはアイソクロナス OUT データパケットを無視します。
2. コアは、グローバル OUT NAK パターンを受信 FIFO に書き込みます。アプリケーションがこのデータパターンを書き込むには、受信 FIFO に十分なスペースを確保する必要があります。
3. アプリケーションがグローバル OUT NAK パターンワードを受信 FIFO からポップすると、コアは、GONAKEFF 割込み (OTG_FS_GINTSTS) を生成します。
4. この割込みを検出したアプリケーションは、コアがグローバル OUT NAK モードにあるとみなすことができます。アプリケーションは、OTG_FS_DCTL の SGONAK ビットをクリアすることによって、この割込みをクリアできます。

アプリケーションのプログラミングシーケンス：

1. 受信 FIFO でのデータの受信も停止するには、アプリケーションは、以下のフィールドをプログラムすることによって、グローバル OUT NAK ビットをセットする必要があります。
 - OTG_FS_DCTL の SGONAK = 1
2. OTG_FS_GINTSTS の GONAKEFF 割込みのアサートを待ちます。アサートされた場合、この割込みは、コアが SETUP パケット以外のいかなるタイプのデータの受信も停止したことを示します。
3. アプリケーションは、OTG_FS_DCTL の SGONAK ビットをセットした後、コアが GONAKEFF 割込み (OTG_FS_GINTSTS) をアサートする前に、有効な OUT パケットを受信することができます。
4. アプリケーションは、OTG_FS_GINTMSK レジスタの GINAKEFFM ビットに書き込むことによって、この割込みを一時的にマスクできます。
 - OTG_FS_GINTMSK レジスタの GINAKEFFM = 0
5. アプリケーションがグローバル OUT NAK モードを終了する準備ができたときには、OTG_FS_DCTL レジスタの SGONAK ビットをクリアする必要があります。これによって、GONAKEFF 割込み (OTG_FS_GINTSTS) もクリアされます。
 - CGONAK の OTG_FS_DCTL = 1
6. アプリケーションがこの割込みをマスクするのが早すぎた場合は、以下のようにしてマスク解除を行う必要があります。
 - GINTMSK の GINAKEFFM = 1

• OUT エンドポイントの無効化

アプリケーションが有効にした OUT エンドポイントを無効にするには、このシーケンスを使用する必要があります。

アプリケーションのプログラミングシーケンス：

1. OUT エンドポイントを無効にする前に、アプリケーションは、コアのグローバル OUT NAK モードを有効にする必要があります。
 - OTG_FS_DCTL の SGONAK = 1
2. OTG_FS_GINTSTS の GONAKEFF 割込みを待ちます。
3. 以下のフィールドをプログラムすることによって、必要な OUT エンドポイントを無効にします。
 - OTG_FS_DOEPCTLx の EPDIS = 1
 - OTG_FS_DOEPCTLx の SNAK = 1
4. EPDISD 割込み (OTG_FS_DOEPINTx) を待ちます。これは、OUT エンドポイントが完全に無効にされたことを示します。EPDISD 割込みがアサートされると、コアは、以下のビットもクリアします。
 - OTG_FS_DOEPCTLx の EPDIS = 0
 - OTG_FS_DOEPCTLx の EPENA = 0
5. アプリケーションは、無効にされていない他のエンドポイントからのデータ受信を開始するために、グローバル OUT NAK ビットをクリアする必要があります。
 - OTG_FS_DCTL の SGONAK = 0

• 一般の非アイソクロナス OUT データ転送

このセクションでは、通常の非アイソクロナス OUT データ転送（コントロール、バルク、またはインタラプト）について説明します。

アプリケーションの要件

1. OUT 転送をセットアップする前に、アプリケーションは OUT 転送の一部として受信されるすべてのデータを収容できるバッファをメモリ内で割り当てる必要があります。
2. OUT 転送の場合、エンドポイントの転送サイズレジスタの転送サイズフィールドは、エンドポイントの最大パケットサイズの倍数でなければならず、ワードの境界に揃えられていなければなりません。
 - 転送サイズ [EPNUM] = $n * (\text{MPSIZ}[\text{EPNUM}] + 4 - (\text{MPSIZ}[\text{EPNUM}] \bmod 4))$
 - パケットカウント [EPNUM] = n
 - $n > 0$
3. OUT エンドポイント割込み時には、アプリケーションは、エンドポイントの転送サイズレジスタを読み出して、メモリ内のペイロードのサイズを計算しなければなりません。受信したペイロードのサイズが、プログラムされた転送サイズより小さいこともあります。
 - メモリ内のペイロードのサイズ = アプリケーションがプログラムした初期転送サイズ – コアが更新した最終転送サイズ
 - このペイロードが受信された USB パケットの数 = アプリケーションがプログラムした初期パケット数 – コアが更新した最終パケット数

内部データフロー

1. アプリケーションがデータを受信するためには、エンドポイント固有レジスタの転送サイズおよびパケットカウントのフィールドを設定し、NAK ビットをクリアし、エンドポイントを有効にする必要があります。
2. NAK ビットがクリアされると、コアは、データの受信を開始し、受信 FIFO にスペースがある限り、データを受信 FIFO に書き込みます。USB で受信された各データパケットについて、データパケットとそのステータスが受信 FIFO に書き込まれます。受信 FIFO にパケット（最大パケットサイズまたはショートパケット）が書き込まれるたびに、そのエンドポイントのパケットカウントフィールドが 1 ずつデクリメントされます。
 - 受信された OUT データパケットのデータ CRC が不良な場合、受信 FIFO から自動的に一掃されます。
 - USB 上のパケットに対して ACK を送信した後、コアは ACK を検出できないホストが再送信する非アイソクロナス OUT データパケットを破棄します。アプリケーションは、同じエンドポイント上では、同じデータ PID を持つ複数の連続データ OUT パケットを検出しません。この場合、パケットカウントはデクリメントされません。
 - 受信 FIFO にスペースがない場合、アイソクロナスまたは非アイソクロナスデータパケットは無視され、受信 FIFO には書き込まれません。さらに、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信します。
 - 上記の 3 つのケースのすべてにおいて、データは受信 FIFO に書き込まれないので、パケットカウントはデクリメントされません。
3. パケットカウントが 0 になるか、エンドポイント上でショートパケットが受信されると、そのエンドポイントの NAK ビットがセットされます。NAK ビットがセットされると、アイソクロナスまたは非アイソクロナスデータパケットは無視され、受信 FIFO には書き込まれず、非アイソクロナス OUT トークンは NAK ハンドシェイク応答を受信します。
4. データが受信 FIFO に書き込まれた後、アプリケーションは受信 FIFO からデータを読み出して、エンドポイントあたり一度に 1 パケットずつ外部メモリに書き込みます。
5. AHB 上で外部メモリへのパケットの書き込みが終わるたびに、書き込まれたパケットのサイズだけエンドポイントの転送サイズがデクリメントされます。

6. 以下の条件の 1 つで OUT エンドポイントの OUT データ転送完了パターンが受信 FIFO に書き込まれます。
 - 転送サイズが 0、およびパケットカウントが 0。
 - 受信 FIFO に書き込まれた最後の OUT データパケットがショートパケット。
($0 \leq \text{パケットサイズ} < \text{最大パケットサイズ}$)
7. アプリケーションがこのエントリ (OUT データ転送完了) をポップすると、エンドポイントの転送完了割込みが生成され、エンドポイントイネーブルビットがクリアされます。

アプリケーションのプログラミングシーケンス :

1. OTG_FS_DOEPTSIZE レジスタで転送サイズおよび対応するパケットカウントをプログラムします。
2. OTG_FS_DOEPCTLx レジスタをエンドポイントの特性でプログラムし、EPENA および CNAK ビットをセットします。
 - OTG_FS_DOEPCTLx の EPENA = 1
 - OTG_FS_DOEPCTLx の CNAK = 1
3. RXFLVL 割込み (OTG_FS_GINTSTS) を待ち、受信 FIFO からデータパケットをエンプティにします。
 - この手順は、転送サイズに応じて何度でも繰り返すことができます。
4. XFRC 割込み (OTG_FS_DOEPINTx) のアサートは、非アイソクロナス OUT データ転送が正常に完了したことを示します。
5. OTG_FS_DOEPTSIZE レジスタを読み出して、受信したデータペイロードのサイズを確認します。

• 一般のアイソクロナス OUT データ転送

このセクションでは、通常のアイソクロナス OUT データ転送について説明します。

アプリケーションの要件

1. 非アイソクロナス OUT データ転送のアプリケーションの要件はすべて、アイソクロナス OUT データ転送にも適用されます。
2. アイソクロナス OUT データ転送の場合、転送サイズおよびパケットカウントフィールドは、常に、単一フレームで受信できる最大パケットサイズのパケット数を設定しなければなりません。アイソクロナス OUT データ転送は、複数のフレームにまたがってはいけません。
3. アプリケーションは、周期的フレームの終わり (OTG_FS_GINTSTS の EOPF 割込み) の前に、受信 FIFO からすべてのアイソクロナス OUT データパケット (データとステータス) を読み出す必要があります。
4. 次のフレームでデータを受信するには、EOPF (OTG_FS_GINTSTS) の後と SOF (OTG_FS_GINTSTS) の前に、アイソクロナス OUT エンドポイントを有効にしなければなりません。

内部データフロー

1. アイソクロナス OUT エンドポイントの内部データフローは、非アイソクロナス OUT エンドポイントの内部データフローと基本的に同じですが、少し異なっているところがあります。
2. エンドポイントイネーブルビットをセットし、NAK ビットをクリアすることによって、アイソクロナス OUT エンドポイントが有効にされたときには、偶数/奇数フレームビットも適切にセットされなければなりません。コアは、以下の条件が満たされた場合に限り、アイソクロナス OUT エンドポイント上で特定のフレームのデータを受信します。
 - EONUM (OTG_FS_DOEPCTLx レジスタ) = SOFFN[0] (OTG_FS_DSTS レジスタ)
3. アプリケーションが受信 FIFO からアイソクロナス OUT データパケット (データとステータス) を完全に読み出すと、コアは、OTG_FS_DOEPTSIZE レジスタの RXDPID フィールドを、受信 FIFO から読み出された最後のアイソクロナス OUT データパケットのデータ PID で更新します。

アプリケーションのプログラミングシーケンス：

1. OTG_FS_DOEPTSIZE レジスタで転送サイズおよび対応するパケットカウントをプログラムします。
2. OTG_FS_DOEPCTLx レジスタをエンドポイントの特性でプログラムし、エンドポイントイネーブル、ClearNAK、および偶数／奇数フレームの各ビットをセットします。
 - EPENA = 1
 - CNAK = 1
 - EONUM = (0: 偶数 / 1: 奇数)
3. RXFLVL 割込み (OTG_FS_GINTSTS) を待ち、受信 FIFO からのデータパケットをエンプティにします。
 - この手順は、転送サイズに応じて何度でも繰り返すことができます。
4. XFRC 割込み (OTG_FS_DOEPINTx) のアサートは、アイソクロナス OUT データ転送が正常に完了したことを示します。この割込みは、必ずしもメモリ内のデータが良好であることを意味しません。
5. この割込みは、アイソクロナス OUT 転送で必ず検出されるとは限りません。その代わり、アプリケーションは、OTG_FS_GINTSTS の IISOXFRM 割込みを検出することができます。
6. OTG_FS_DOEPTSIZE レジスタを読み出して、受信した転送データのサイズを確認し、フレームで受信したデータの有効性を確認します。アプリケーションは、以下の条件の 1 つが満たされた場合のみ、メモリに受信されたデータを有効として扱う必要があります。
 - RXDPID = D0 (OTG_FS_DOEPTSIZE)、およびこのペイロードが受信された USB パケットの数 = 1
 - RXDPID = D1 (OTG_FS_DOEPTSIZE)、およびこのペイロードが受信された USB パケットの数 = 2
 - RXDPID = D2 (OTG_FS_DOEPTSIZE)、およびこのペイロードが受信された USB パケットの数 = 3

このペイロードが受信された USB パケットの数 =
アプリケーションがプログラムした初期パケット数 – コアが更新した最終パケット数

アプリケーションは、無効なデータパケットを破棄できます。

• 不完全アイソクロナス OUT データ転送

このセクションでは、アイソクロナス OUT データパケットがコアの内部でドロップされたときのアプリケーションのプログラミングシーケンスについて説明します。

内部データフロー

1. アイソクロナス OUT エンドポイントの場合、XFRC 割込み (OTG_FS_DOEPINTx) は必ずアサートされるわけではありません。コアがアイソクロナス OUT データパケットをドロップした場合、アプリケーションは以下の状況で XFRC 割込み (OTG_FS_DOEPINTx) の検出に失敗することがあります。
 - 受信 FIFO が完全な ISO OUT データパケットを収容できない場合、コアは、受信した ISO OUT データをドロップします。
 - アイソクロナス OUT データパケットが CRC エラー付きで受信されたとき。
 - コアが受信したアイソクロナス OUT トークンが破損しているとき。
 - アプリケーションが受信 FIFO からデータを読み出すのに長時間かかっているとき。
2. コアが、すべてのアイソクロナスエンドポイントへの転送を完了する前に周期的フレームの終わりを検出すると、不完全アイソクロナス OUT データ転送割込み (OTG_FS_GINTSTS レジスタの IISOXFRM) をアサートして、アイソクロナスエンドポイントのうち少なくとも 1 つで XFRC 割込み (OTG_FS_DOEPINTx) がアサートされていないことを示します。この時点で、

不完全転送のエンドポイントは有効なままですが USB 上のこのエンドポイントでは、アクティブな転送は進行していません。

アプリケーションのプログラミングシーケンス：

1. IISOXFRM 割込み (OTG_FS_GINTSTS) のアサートは、現在のフレームで、少なくとも 1 つのアイソクロナス OUT エンドポイントが転送を完了していないことを示します。
2. アイソクロナス OUT データがエンドポイントから完全に出されていないためにこの割込みが発生した場合、アプリケーションは処理を進める前に、受信 FIFO からすべてのアイソクロナス OUT データ (データとステータス) を出力する必要があります。
 - すべてのデータが受信 FIFO から出されると、アプリケーションは XFRC 割込み (OTG_FS_DOEPINTx) を検出することができます。この場合、アプリケーションは、次のフレームでアイソクロナス OUT データを受信するには、エンドポイントを再び有効にする必要があります。
3. IISOXFRM 割込み (OTG_FS_GINTSTS) を受信すると、アプリケーションは、すべてのアイソクロナス OUT エンドポイントの制御レジスタ (OTG_FS_DOEPCTLx) を読み出して、現在のマイクロフレームにおいて転送を完了しなかったエンドポイントを確認する必要があります。次の両方の条件が満たされた場合、エンドポイント転送は完了しません。
 - EONUM ビット (OTG_FS_DOEPCTLx) = SOFFN[0] (OTG_FS_DSTS)
 - EPENA = 1 (OTG_FS_DOEPCTLx)
4. 現在のフレーム番号が変更されないように、上記の手順は、SOF 割込み (OTG_FS_GINTSTS) が検出される前に実行されなければなりません。
5. 不完全転送のアイソクロナス OUT エンドポイントの場合、アプリケーションは、メモリ内のデータを破棄し、OTG_FS_DOEPCTLx レジスタの EDPIS ビットをセットすることによってエンドポイントを無効にする必要があります。
6. EDPIS 割込み (OTG_FS_DOEPINTx) を待ち、次のフレームで新しいデータを受信するために、エンドポイントを有効にします。
 - コアがエンドポイントを無効にするには若干の時間がかかるので、アプリケーションは不良なアイソクロナスデータを受信した後、次のフレームのデータを受信できないことがあります。

• 非アイソクロナス OUT エンドポイントの停止

このセクションでは、アプリケーションが非アイソクロナスエンドポイントを停止する方法について説明します。

1. コアをグローバル OUT NAK モードにします。
2. 必要なエンドポイントを無効にします。
 - エンドポイントを無効にするときには、OTG_FS_DOEPCTL の SNAK ビットをセットする代わりに、STALL = 1 にセットしてください (OTG_FS_DOEPCTL)。

STALL ビットは常に NAK ビットより優先されます。
3. アプリケーションがエンドポイントの STALL ハンドシェイクを終了する準備ができたときには、STALL ビット (OTG_FS_DOEPCTLx) をクリアしなければなりません。
4. アプリケーションが SetFeature.Endpoint Halt または ClearFeature.Endpoint Halt コマンドのためにエンドポイントの STALL ビットをセットまたはクリアする場合、STALL ビットは、アプリケーションがコントロールエンドポイントでステータスステージ転送をセットアップする前にセットまたはクリアされなければなりません。

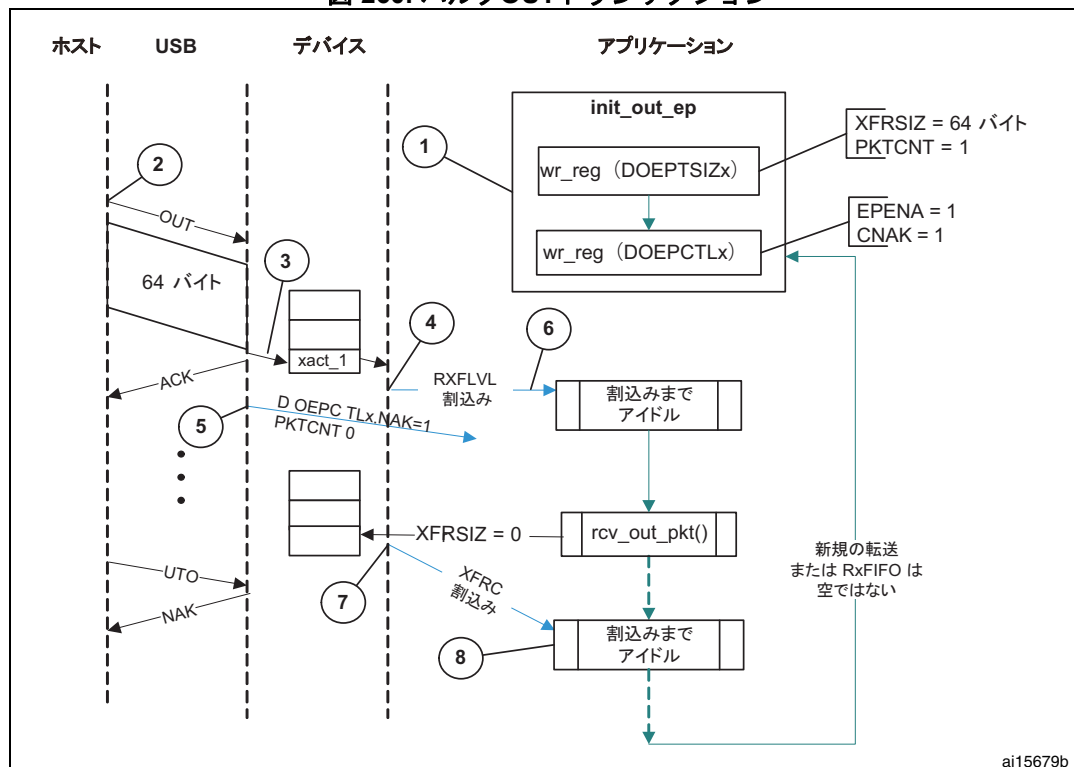
例

このセクションでは、いくつかの基本的な転送タイプとシナリオについて説明します。

- バルクOUTトランザクション

図 260 に USB から AHB への単一バルク OUT データパケットの受信と、このプロセスに伴うイベントを示します。

図 260. バルクOUTトランザクション



SetConfiguration/SetInterface コマンドの後、アプリケーションは、CNAK = 1 および EPENA = 1 (OTG_FS_DOEPCTLx) をセットし、OTG_FS_DOEPTSIZx レジスタの適切な XFRSIZ および PKTCNT ビットをセットすることによって、すべての OUT エンドポイントを初期化します。

1. ホストは、データ (OUT トークン) のエンドポイントへの送信を試みます。
2. コアは、USB 上で OUT トークンを受信すると、スペースが使用可能なので、RxFIFO にパケットを格納します。
3. RxFIFO に完全なパケットを書き込んだ後、コアは RXFLVL 割込み (OTG_FS_GINTSTS) をアサートします。
4. USB パケットの PKTCNT 番号を受信すると、コアは、それ以上パケットが受信されないように、このエンドポイントの NAK ビットを内部でセットします。
5. アプリケーションは割込みを処理して、RxFIFO からデータを読み出します。
6. アプリケーションがすべてのデータ (XFRSIZ に相当) を読み出すと、コアは、XFRC 割込み (OTG_FS_DOEPINTx) を生成します。
7. アプリケーションは割込みを処理して、XFRC 割込みビット (OTG_FS_DOEPINTx) の設定を使用して、意図した転送が完了したかどうかを確認します。

IN データ転送

• パケットの書き込み

このセクションでは、専用の送信 FIFO が有効なときに、アプリケーションがデータパケットをエンドポイント FIFO に書き込む方法について説明します。

1. アプリケーションは、ポーリングまたは割込みモードのいずれかを選択できます。
 - ポーリングモードでは、アプリケーションは、OTG_FS_DTXFSTSx レジスタを読み出すことによってエンドポイント送信データ FIFO のステータスを監視して、データ FIFO に十分なスペースがあるかどうかを確認します。
 - 割込みモードでは、アプリケーションは、TXFE 割込み (OTG_FS_DIEPINTx) を待ってから、OTG_FS_DTXFSTSx レジスタを読み出して、データ FIFO 内に十分なスペースがあるかどうかを確認します。
 - 単一の非ゼロ長データパケットを書き込むには、パケット全体を書き込むためのスペースがデータ FIFO になければなりません。
 - ゼロ長パケットを書き込むには、アプリケーションは FIFO のスペースを考慮してはなりません。
2. 上記の方法の 1 つを使用して、アプリケーションが送信パケットを書き込むのに十分なスペースがあることを確認するときには、アプリケーションは、データをデータ FIFO に書き込む前に、まず、エンドポイント制御レジスタに書き込む必要があります。通常、アプリケーションは、エンドポイントイネーブルビットをセットする場合を除き、レジスタの内容を変更しないように、OTG_FS_DIEPCTLx レジスタにリードモディファイライトを行う必要があります。

アプリケーションは、使用可能なスペースがあれば、同じエンドポイントに対する複数のパケットを送信 FIFO に書き込むことができます。周期的 IN エンドポイントの場合、アプリケーションは 1 つのマイクロフレームのパケットのみを書き込む必要があります。アプリケーションは、前のトランザクションの転送完了割込みを受信した後でのみ、次の周期的トランザクションのパケットを書き込むことができます。

• IN エンドポイント NAK の設定

内部データフロー

1. アプリケーションが特定のエンドポイントの IN NAK をセットすると、コアは、エンドポイントの送信 FIFO にデータがあるかどうかに関係なく、そのエンドポイントでのデータ送信を停止します。
2. 非アイソクロナス IN トークンは NAK ハンドシェイク応答を受信します。
 - アイソクロナス IN トークンはゼロデータ長のパケット応答を受信します。
3. コアは、OTG_FS_DIEPCTLx の SNAK ビットに応答して、OTG_FS_DIEPINTx の INEPNE 割込み (IN エンドポイント NAK 有効) をアサートします。
4. この割込みがアプリケーションによって検出されると、アプリケーションは、エンドポイントが IN NAK モードにあるとみなすことができます。この割込みは、アプリケーションが OTG_FS_DIEPCTLx の CNAK ビットをセットすることによってクリアできます。

アプリケーションのプログラミングシーケンス :

1. 特定の IN エンドポイントでのデータ送信を停止するには、アプリケーションは、IN NAK ビットをセットする必要があります。このビットをセットするには、以下のフィールドをプログラムする必要があります。
 - OTG_FS_DIEPCTLx の SNAK = 1
2. OTG_FS_DIEPINTx の INEPNE 割込みのアサートを待ちます。この割込みは、コアがエンドポイントでのデータ送信を停止したことを示します。
3. コアは、アプリケーションが NAK ビットをセットした後の NAK 有効割込みがアサートされる前に、エンドポイントで有効な IN データを送信できます。

4. アプリケーションは、DIEPMSK の INEPNEM ビットに書き込むことによって、この割込みを一時的にマスクできます。
 - DIEPMSK の INEPNEM = 0
5. エンドポイント NAK モードを終了するには、アプリケーションは、OTG_FS_DIEPCTLx の NAK ステータスビット (NAKSTS) をクリアする必要があります。これによって、INEPNE 割込み (OTG_FS_FS_DIEPINTx) もクリアされます。
 - OTG_FS_DIEPCTLx の CNAK = 1
6. アプリケーションがこの割込みをマスクするのが早すぎた場合、以下のようにしてマスク解除を行う必要があります。
 - DIEPMSK の INEPNEM = 1

• IN エンドポイントディセーブル

以前に有効にされた特定の IN エンドポイントを無効にするには、以下のシーケンスを使用します。

アプリケーションのプログラミングシーケンス：

1. アプリケーションは、AHB でのデータ書き込みを停止して、IN エンドポイントを無効にする必要があります。
2. アプリケーションは、エンドポイントを NAK モードに設定する必要があります。
 - OTG_FS_DIEPCTLx の SNAK = 1
3. OTG_FS_DIEPINTx の INEPNE 割込みを待ちます。
4. 無効にしなければならないエンドポイントについて、OTG_FS_DIEPCTLx レジスタの以下のビットをセットします。
 - OTG_FS_DIEPCTLx の EPDIS = 1
 - OTG_FS_DIEPCTLx の SNAK = 1
5. OTG_FS_DIEPINTx レジスタの EPDISD 割込みのアサートは、コアが指定されたエンドポイントを完全に無効にしたことを示します。割込みのアサートとともに、コアは、以下のビットもクリアします。
 - OTG_FS_DIEPCTLx の EPENA = 0
 - OTG_FS_DIEPCTLx の EPDIS = 0
6. アプリケーションは、周期的 IN EP について OTG_FS_DIEPTSIZx レジスタを読み出して、エンドポイント上のどれだけのデータが USB で送信されたかを計算する必要があります。
7. アプリケーションは、OTG_FS_GRSTCTL レジスタの以下のフィールドを設定することによって、エンドポイント送信 FIFO 内のデータをフラッシュする必要があります。
 - TXFNUM (OTG_FS_GRSTCTL) = エンドポイント送信 FIFO 番号
 - TXFFLSH (OTG_FS_GRSTCTL) = 1

アプリケーションは、TXFFLSH ビットがコアによってクリアされるまで（フラッシュ動作が終了したことを示す）、OTG_FS_GRSTCTL レジスタにポーリングする必要があります。このエンドポイントで新しいデータを送信するために、アプリケーションは後で、エンドポイントを再び有効にできます。

• 一般の非周期的 IN データ転送

アプリケーションの要件

1. IN 転送をセットアップする前に、アプリケーションは、IN 転送の一部として送信されるすべてのデータが単一バッファの一部であることを確認する必要があります。
2. IN 転送の場合、エンドポイント転送サイズレジスタの転送サイズフィールドは、最大パケットサイズの複数のパケットと単一のショートパケットから成るペイロードを表します。このショートパケットは、転送の最後に送信されます。
 - 転送の終わりに少数の最大パケットサイズのパケットとショートパケットを送信するには :

$$\text{転送サイズ [EPNUM]} = x * \text{MPSIZ[EPNUM]} + \text{sp}$$
 (sp > 0) の場合、パケットカウント [EPNUM] = x + 1。
 そうでない場合、パケットカウント [EPNUM] = x
 - 単一のゼロ長データパケットを送信するには :

$$\text{転送サイズ [EPNUM]} = 0$$

$$\text{パケットカウント [EPNUM]} = 1$$
 - 転送の終わりに少数の最大パケットサイズのパケットとゼロ長データパケットを送信するには、アプリケーションは転送を 2 つの部分に分ける必要があります。最初の部分では最大パケットサイズのデータパケットを送信し、2 番目の部分ではゼロ長データパケットのみを送信します。

$$\text{最初の転送 : 転送サイズ [EPNUM]} = x * \text{MPSIZ[epnum]} ; \text{パケットカウント} = n ;$$

$$\text{2 番目の転送 : 転送サイズ [EPNUM]} = 0 ; \text{パケットカウント} = 1 ;$$
3. エンドポイントがデータ転送のために有効にされると、コアは、転送サイズレジスタを更新します。IN 転送の終了時に、アプリケーションは、転送サイズレジスタを読み出して、送信 FIFO にポストされたデータのうち、どれだけが USB で送信されたかを確認する必要があります。
4. 送信 FIFO にフェッチされたデータ = アプリケーションがプログラムした初期転送サイズ – コアが更新した最終転送サイズ
 - USB で送信されたデータ = (アプリケーションがプログラムした初期パケットカウント – コアが更新した最終パケットカウント) * MPSIZ[EPNUM]
 - USB 上でこれから送信されるデータ = (アプリケーションがプログラムした初期転送サイズ – USB で送信されたデータ)

内部データフロー

1. アプリケーションは、エンドポイント固有レジスタの転送サイズおよびパケットカウントフィールドを設定して、データを送信するためにエンドポイントを有効にする必要があります。
2. アプリケーションは、要求されたデータをエンドポイントの送信 FIFO に書き込む必要もあります。
3. アプリケーションによってパケットが送信 FIFO に書き込まれるたびに、そのエンドポイントの転送サイズがパケットのサイズだけデクリメントされます。エンドポイントの転送サイズが 0 になるまで、データはアプリケーションによってメモリからフェッチされます。FIFO にデータを書き込んだ後、「FIFO 内のパケット数」はインクリメントされます (これは 3 ビットのカウンタであり、各 IN エンドポイントの送信 FIFO について、コアによって内部で維持されます。IN エンドポイント FIFO 内にコアによって一度に維持されるパケットの最大数は 8 です)。ゼロ長のパケットの場合、各 FIFO について、FIFO 内にデータがないことを示す個別のフラグがセットされます。
4. データが送信 FIFO に書き込まれると、コアは、IN トークンの受信時に読み出します。非アイソクロナス IN データパケットが ACK ハンドシェイクとともに送信されるたびに、エンドポイントのパケットカウントは、0 になるまで 1 ずつデクリメントされます。パケットカウントは、タイムアウト時にはデクリメントされません。

5. ゼロ長パケットの場合（内部ゼロ長フラグで示されます）、コアは IN トークンのゼロ長パケットを送出し、パケットカウンフィールドをデクリメントします。
6. 受信された IN トークンのデータが FIFO になく、そのエンドポイントのパケットカウンフィールドがゼロの場合、コアは、そのエンドポイントについて、「Tx FIFO がエンプティのとき IN トークンが受信された」という割込み (ITTXFE 割込み) を生成します。ただし、エンドポイントの NAK ビットがセットされていない場合に限りです。コアは、USB 上の非アイソクロナスエンドポイントに対して NAK ハンドシェイクで応答します。
7. コアは、内部で FIFO ポインタを巻き戻し、タイムアウト割込みは生成されません。
8. 転送サイズが 0 であり、パケットカウンが 0 のとき、エンドポイントの転送完了割込み (XFRC) が生成され、エンドポイントイネーブルビットがクリアされます。

アプリケーションのプログラミングシーケンス：

1. OTG_FS_DIEPTISZx レジスタで転送サイズと対応するパケットカウンをプログラムします。
2. OTG_FS_DIEPCTLx レジスタをエンドポイントの特性でプログラムし、CNAK および EPENA (エンドポイントイネーブル) ビットをセットします。
3. 非ゼロ長データパケットを送信するときには、アプリケーションは、OTG_FS_DTXFSTSx レジスタ (x はそのエンドポイントに関連する FIFO 番号) にポーリングして、データ FIFO に十分なスペースがあるかどうかを確認する必要があります。アプリケーションは、データを書き込む前に、オプションで TXFE (OTG_FS_DIEPINTx) を使用することができます。

一般の周期的 IN データ転送

このセクションでは、典型的な周期的 IN データ転送について説明します。

アプリケーションの要件

1. 783 ページの一般の非周期的 IN データ転送のアプリケーションの要件 1、2、3、および 4 は、要件 2 が少し変更されていることを除いて、周期的 IN データ転送にも適用されます。
 - アプリケーションは複数の最大パケットサイズのデータパケットの送信、または複数の最大パケットサイズのパケットに最後にショートパケットを加えた送信のみを行うことができます。転送の終わりに少数の最大パケットサイズのパケットとショートパケットを送信するには、以下の条件が満たされなければなりません。

$$\text{転送サイズ [EPNUM]} = x * \text{MPSIZ[EPNUM]} + sp$$
 (x は整数 ≥ 0 、および $0 \leq sp < \text{MPSIZ[EPNUM]}$)

$$(sp > 0) \text{ の場合、パケットカウン [EPNUM]} = x + 1$$
 そうでない場合、パケットカウン [EPNUM] = x ;

$$\text{MCNT[EPNUM]} = \text{パケットカウン [EPNUM]}$$
 - アプリケーションは転送の最後にゼロ長データパケットを送信することはできません。自分で単一のゼロ長パケットを送信することは可能です。単一のゼロ長データパケットを送信するには：
 - 転送サイズ [EPNUM] = 0
 - パケットカウン [EPNUM] = 1
 - $\text{MCNT[EPNUM]} = \text{パケットカウン [EPNUM]}$
2. アプリケーションは一度に 1 フレームのデータ転送のみをスケジュールすることができます。
 - $(\text{MCNT} - 1) * \text{MPSIZ} \leq \text{XFERSIZ} \leq \text{MCNT} * \text{MPSIZ}$
 - $\text{PKTCNT} = \text{MCNT}$ (OTG_FS_DIEPTISZx レジスタ)
 - $\text{XFERSIZ} < \text{MCNT} * \text{MPSIZ}$ の場合、転送の最後のデータパケットはショートパケットです。
 - 以下のことに注意してください。MCNT は OTG_FS_DIEPTISZx レジスタに、MPSIZ は OTG_FS_DIEPCTLx レジスタに、PKTCNT は OTG_FS_DIEPTISZx レジスタに、また XFERSIZ は OTG_FS_DIEPTISZx レジスタにあります。

3. フレームで送信される完全なデータは、IN トークンが受信される前に、アプリケーションによって送信 FIFO に書き込まれなければなりません。IN トークンが受信されたときに、フレームあたりで送信されるデータの 1 ワードが送信 FIFO 内で欠落していても、コアは FIFO がエンプティである場合のように動作します。送信 FIFO がエンプティのとき：
 - アイソクロナス IN エンドポイントに対して USB 上でゼロデータ長のパケットが送信されます。
 - IN エンドポイントに割込みをかけるため、USB 上で NAK ハンドシェイクが送信されます。

内部データフロー

1. アプリケーションは、エンドポイント固有レジスタの転送サイズおよびパケットカウントフィールドを設定して、データを送信するためにエンドポイントを有効にする必要があります。
2. また、アプリケーションは、要求されたデータをエンドポイントの関連する送信 FIFO に書き込む必要があります。
3. アプリケーションがパケットを送信 FIFO に書き込むたびに、そのエンドポイントの転送サイズがパケットのサイズだけデクリメントされます。エンドポイントの転送サイズが 0 になるまで、アプリケーションメモリからデータがフェッチされます。
4. 周期的エンドポイントの IN トークンが受信されると、コアは、FIFO のデータ（ある場合）を送信します。フレームの完全なデータペイロード（専用 FIFO モードでは完全なパケット）が FIFO に存在しない場合、コアは、エンドポイントに対する TxFIFO エンプティ割込みが受信されたときに IN トークンを生成します。
 - アイソクロナス IN エンドポイントに対して、USB 上でゼロ長のデータパケットが送信されます。
 - IN エンドポイントに割込みをかけるため、USB 上で NAK ハンドシェイクが送信されます。
5. エンドポイントのパケットカウントは、以下の条件のもとで 1 ずつデクリメントされます。
 - アイソクロナスエンドポイントに対して、ゼロ長または非ゼロ長データパケットが送信されたとき。
 - インタラプトエンドポイントに対して、ACK ハンドシェイクが送信されたとき。
 - 転送サイズとパケットカウントの両方が 0 のとき、エンドポイントに対して転送完了割込みが生成され、エンドポイントイネーブルビットがクリアされます。
6. 「周期的フレームインターバル」（OTG_FS_DCFG の PFIVL によって制御されます）で、コアが現在のフレームについてスケジュールされたアイソクロナス IN エンドポイント FIFO のいずれかが非エンプティであることを検出すると、コアは OTG_FS_GINTSTS の IISOIXFR 割込みを生成します。

アプリケーションのプログラミングシーケンス：

1. OTG_FS_DIEPCTLx レジスタをエンドポイントの特性でプログラムして、CNAK および EPENA ビットをセットします。
2. 次のフレームで送信するデータを送信 FIFO に書き込みます。
3. OTG_FS_DIEPINTx レジスタの ITTXFE 割込みのアサートは、アプリケーションが送信するすべてのデータを送信 FIFO にまだ書き込んでいないことを示します。
4. この割込みが検出されたときに、インタラプトエンドポイントがすでに有効だった場合、割込みは無視されます。まだ有効になっていなかった場合は、次の IN トークン試行の際にデータを送信できるように、エンドポイントを有効にします。
5. OTG_FS_DIEPINTx レジスタの ITTXFE 割込みがない状態での XFRC 割込み (OTG_FS_DIEPINTx) のアサートは、アイソクロナス IN 転送が正常に完了したことを示します。OTG_FS_DIEPTSIZx レジスタの読出し結果は、転送サイズ = 0、かつパケットカウント = 0、すなわち、すべてのデータが USB で送信されたことを示さなければなりません。

6. ITTXFE 割込み (OTG_FS_DIEPINTx) の有無にかかわらず、XFRC 割込み (OTG_FS_DIEPINTx) のアサートは、アイソクロナス IN 転送が正常に完了したことを示します。OTG_FS_DIEPTSIZx レジスタの読出し結果は、転送サイズ = 0、かつパケットカウンタ = 0、すなわち、すべてのデータが USB で送信されたことを示さなければなりません。
 7. 前述のどの割込みもない状態で OTG_FS_GINTSTS の不完全アイソクロナス IN 転送 (IISOIXFR) 割込みのアサートは、コアが現在のフレームで少なくとも 1 つの周期的 IN トークンを受信しなかったことを示します。
- **不完全アイソクロナス IN データ転送**

このセクションでは、アプリケーションが不完全なアイソクロナス IN データ転送で行う必要があることについて説明します。

内部データフロー

1. アイソクロナス IN 転送は、以下の条件のいずれかがあてはまる場合に不完全として扱われます。
 - a) コアが、少なくとも 1 つのアイソクロナス IN エンドポイントで破損したアイソクロナス IN トークンを受信した場合。この場合、アプリケーションは、不完全アイソクロナス IN 転送割込み (OTG_FS_GINTSTS の IISOIXFR ビット) を検出します。
 - b) アプリケーションが完全なデータペイロードを送信 FIFO に書き込むのに時間がかかり、完全なデータペイロードが FIFO に書き込まれる前に IN トークンを受信された場合。この場合、アプリケーションは、OTG_FS_DIEPINTx の TxFIFO エンプティ割込み時に IN トークンの受信を検出します。最終的には、周期的フレームの終わりで不完全アイソクロナス IN 転送割込み (OTG_FS_GINTSTS の IISOIXFR) になるので、アプリケーションはこの割込みを無視することができます。
コアは、受信した IN トークンにตอบสนองして、USB 上にゼロ長データパケットを送信します。
2. アプリケーションは、できるだけ速やかに、送信 FIFO へのデータペイロードの書き込みを停止する必要があります。
3. アプリケーションは、エンドポイントの NAK ビットとディセーブルビットをセットする必要があります。
4. コアは、エンドポイントを無効にし、ディセーブルビットをクリアし、エンドポイントのエンドポイントディセーブル割込みをアサートします。

アプリケーションのプログラミングシーケンス：

1. 最終的には不完全アイソクロナス IN 転送割込み (OTG_FS_GINTSTS) になるので、アプリケーションは、アイソクロナス IN エンドポイントで OTG_FS_DIEPINTx の TxFIFO エンプティ割込み時に受信された IN トークンを無視することができます。
2. 不完全アイソクロナス IN 転送割込み (OTG_FS_GINTSTS) のアサートは、少なくとも 1 つのアイソクロナス IN エンドポイントで不完全アイソクロナス IN 転送があったことを示します。
3. アプリケーションは、すべてのアイソクロナス IN エンドポイントのエンドポイント制御レジスタを読み出して、不完全 IN データ転送があるエンドポイントを検出する必要があります。
4. アプリケーションは、AHB 上でこれらのエンドポイントに関連する周期的送信 FIFO へのデータの書き込みを停止する必要があります。
5. OTG_FS_DIEPCTLx レジスタの以下のフィールドをプログラムして、エンドポイントを無効にします。
 - OTG_FS_DIEPCTLx の SNAK = 1
 - OTG_FS_DIEPCTLx の EPDIS = 1

6. OTG_FS_DIEPINTx のエンドポイントディセーブル割込みのアサートは、コアがエンドポイントを無効にしたことを示します。
 - この時点で、アプリケーションは、次のマイクロフレームで新しい転送を行うために、関連する送信 FIFO 内のデータを一扫するか、エンドポイントを有効にすることによって、FIFO 内の既存のデータに上書きする必要があります。データをフラッシュするには、アプリケーションは OTG_FS_GRSTCTL レジスタを使用する必要があります。

• 非アイソクロナス IN エンドポイントの停止

このセクションでは、アプリケーションが非アイソクロナスエンドポイントを停止する方法について説明します。

アプリケーションのプログラミングシーケンス：

1. 停止する IN エンドポイントを無効にします。STALL ビットもセットします。
2. エンドポイントがすでに有効になっているときには、OTG_FS_DIEPCTLx の EPDIS = 1 にします。
 - OTG_FS_DIEPCTLx の STALL = 1
 - STALL ビットは常に NAK ビットより優先されます。
3. エンドポイント無効割込み (OTG_FS_DIEPINTx) のアサートは、コアが指定されたエンドポイントを無効にしたことをアプリケーションに示します。
4. アプリケーションは、エンドポイントのタイプに応じて、非周期的または周期的 FIFO をフラッシュする必要があります。非周期的エンドポイントの場合、アプリケーションは、データを送信するために、停止する必要がある他の非周期的エンドポイントを再び有効にする必要があります。
5. アプリケーションがエンドポイントの STALL ハンドシェイクを終了する準備ができたときには、OTG_FS_DIEPCTLx の STALL ビットがクリアされなければなりません。
6. アプリケーションが SetFeature.Endpoint Halt コマンドまたは ClearFeature.Endpoint Halt コマンドのためにエンドポイントの STALL ビットをセットまたはクリアする場合、STALL ビットは、アプリケーションがコントロールエンドポイントでステータスステージ転送をセットアップする前にセットまたはクリアされなければなりません。

特殊なケース：コントロール OUT エンドポイントの停止

コントロール転送のデータステージで、ホストが SETUP パケットで指定されたより多くの IN/OUT トークンを送信した場合、コアは、IN/OUT トークンを停止する必要があります。この場合、アプリケーションは、コアが SETUP パケットで指定されたデータ量を転送した後、制御転送のデータステージで、OTG_FS_DIEPINTx の ITTXFE 割込みと OTG_FS_DOEPINTx の OTEPDIS 割込みを有効にしなければなりません。その後、アプリケーションがこの割込みを受信したとき、アプリケーションは、対応するエンドポイント制御レジスタの STALL ビットをセットし、この割込みをクリアする必要があります。

22.17.7 最悪ケースの応答時間

OTG_FS コントローラがデバイスとして機能するとき、アイソクロナス OUT に続く任意のトークンで、最悪ケースの応答時間があります。この最悪ケース応答時間は、AHB クロック周波数に依存します。

コアのレジスタは AHB ドメインにあり、コアはこれらのレジスタを更新するまでは、別のトークンを受け入れません。アイソクロナストランザクションの場合、ハンドシェイクはなく、次のトークンがすぐに入ってくる可能性があるため、アイソクロナス OUT に続くトークンで最悪ケースが生じます。この最悪ケース値は、AHB クロックが PHY クロックと同じときには 7 PHY クロックです。AHB クロックが高速なほど、この値は小さくなります。

この最悪ケース条件が発生した場合、コアは、バルク/インタラプトトークンに NAK で応答し、アイソクロナスおよび SETUP トークンをドロップします。ホストは、これを SETUP のタイムアウト条件と解釈して、SETUP パケットを再試行します。アイソクロナス転送の場合、不完全アイソクロ


ナス IN 転送割込み (IISOIXFR) と不完全アイソクロナス OUT 転送割込み (IISOOXFR) は、アイソクロナス IN/OUT パケットがドロップされたことをアプリケーションに知らせます。

OTG_FS_GUSBCFG の TRDT の値の選択

TRDT (OTG_FS_GUSBCFG) の値は、MAC が IN トークンを受信した後、FIFO ステータスと PFC ブロックから最初のデータを取得するまでの時間 (PHY クロック数) です。この時間には、PHY クロックと AHB クロック間の同期遅延も含まれます。この場合の最悪ケース遅延は、AHB クロックが PHY クロックと同じときに発生します。この場合、遅延は 5 クロックです。

MAC が IN トークンを受信すると、この情報 (トークンの受信) は PFC (PFC は AHB クロックで動作します) によって AHB に同期されます。次に、PFC は、SPRAM からデータを読み出し、それらをデュアルクロックソースバッファに書き込みます。MAC は、データをソースバッファ (深さ 4) から読み出します。

AHB が PHY より高い周波数で動作している場合、アプリケーションは TRDT (OTG_FS_GUSBCFG) の値として、より小さな値を使用することができます。

 261 に以下の信号を示します。

- tkn_rcvd : MAC から PFC への、トークンが受信されたという情報
- dynced_tkn_rcvd : PCLK から HCLK ドメインへの二重同期 tkn_rcvd
- spr_read : SPRAM への読み出し
- spr_addr : SPRAM へのアドレス
- spr_rdata : SPRAM からのデータの読み出し
- srcbuf_push : ソースバッファへのプッシュ
- srcbuf_rdata : ソースバッファからのデータの読み出し MAC によるデータの検出

TRDT の値対 AHB クロック周波数については、[表 131 : TRDT 値](#)を参照してください。

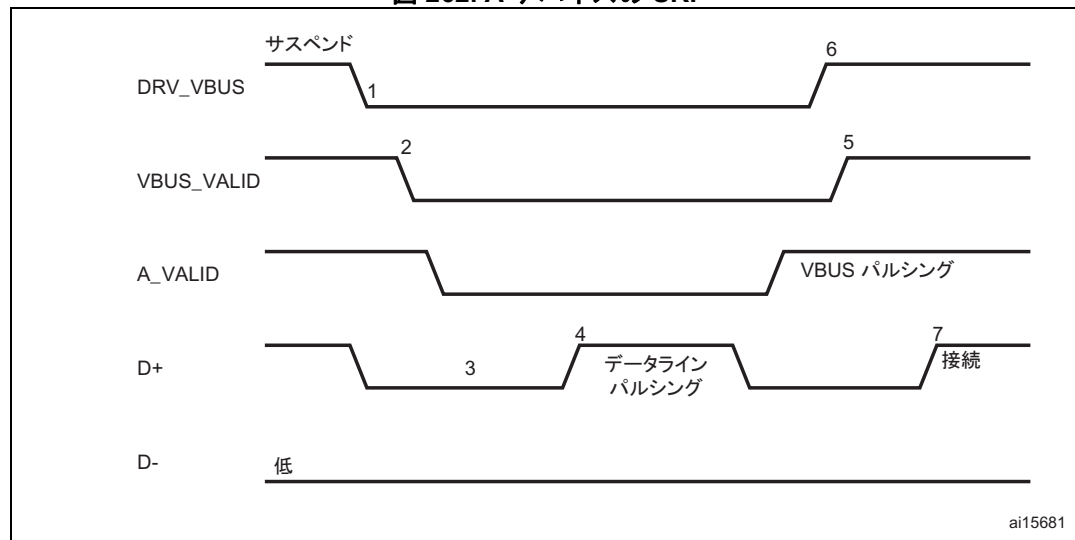
Timing diagram showing clock signals (HCLK, PCLK) and data signals (tkn_rcvd, dsynced_tkn_rcvd, spr_read, spr_addr, spr_rdata, srcbuf_push, srcbuf_rdata) over 200ns. The diagram is divided into 8 clock cycles (1-8). A 5-cycle period is indicated at the bottom.

OTG_FS コントローラは、HNP と SRP をサポートする OTG デバイスです。コアが A プラグに接続されると、A デバイスと呼ばれます。コアが B プラグに接続されると、B デバイスと呼ばれます。ホストモードでは、OTG_FS コントローラは、電力を節約するために V_{BUS} をオフにします。SRP は、B デバイスが A デバイスに V_{BUS} 電源をオンにするように信号を送る手段です。デバイスは、データラインパルシングと V_{BUS} パルシングの両方を実行する必要がありますが、ホストは SRP のデータラインパルシングか V_{BUS} パルシングのどちらかを検出することができます。HNP は、B デバイスがネゴシエイトし、役割をホストに切り替える手段です。HNP 後のネゴシエイトモードでは、B デバイスはバスをサスペンドし、その役割をデバイスに戻します。

A デバイスセッションリクエストプロトコル

アプリケーションは、コア USB 設定レジスタの SRP 対応ビットをセットする必要があります。これによって、OTG_FS コントローラは SRP を A デバイスとして検出できるようになります。

図 262. A デバイスの SRP



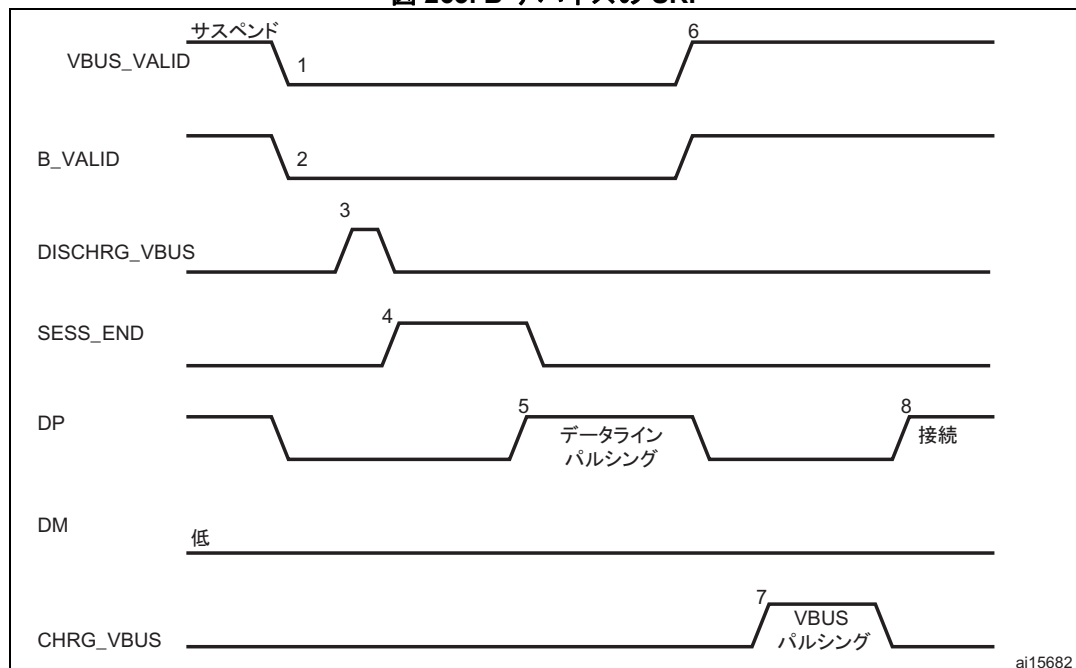
1. DRV_VBUS = PHY への V_{BUS} 駆動信号
VBUS_VALID = PHY からの V_{BUS} 有効信号
A_VALID = PHY への A ペリフェラル V_{BUS} レベル信号
D+ = データ正ライン
D- = データ負ライン

1. 節電のためには、アプリケーションは、バスがアイドル状態のときに、ホストポート制御およびステータスレジスタのポートサスペンドビットおよびポートパワービットに書き込むことによって、ポート電源をサスペンドおよびオフにします。
2. PHY は、VBUS_VALID 信号をネゲートすることによって、ポート電源のオフを示します。
3. デバイスは、 V_{BUS} 電源がオフになると、SRP を開始するために少なくとも 2 ms 以上 SE0 を検出しなければなりません。
4. SRP を開始するには、デバイスはデータラインプルアップ抵抗を 5~10 ms の間オンにします。OTG_FS コントローラはデータラインのパルシングを検出します。
5. デバイスは、 V_{BUS} を A デバイスセッション有効レベル以上（最低 2.0 V）に駆動します（ V_{BUS} パルシングに対して）。
OTG_FS コントローラは、SRP を検出すると、アプリケーションに割込みをかけます。グローバル割込みステータスレジスタのセッションリクエスト検出ビットがセットされます (OTG_FS_GINTSTS の SRQINT)。
6. アプリケーションは、セッションリクエスト検出割込みを処理して、ホストポート制御およびステータスレジスタのポートパワービットに書き込むことによって、ポート電源をオンにする必要があります。PHY は、VBUS_VALID 信号をアサートすることによって、ポート電源がオンになったことを示します。
7. USB に電源が投入されると、デバイスが接続され、SRP プロセスが完了します。

B デバイスセッションリクエストプロトコル

アプリケーションは、コア USB 設定レジスタの SRP 対応ビットをセットする必要があります。これによって、OTG_FS コントローラは SRP を B デバイスとして扱えるようになります。SRP は、OTG_FS コントローラがホストからの新しいセッションをリクエストする手段です。

図 263. B デバイスの SRP



1. VBUS_VALID = PHY からの V_{BUS} 有効信号
B_VALID = PHY への B ペリフェラル有効セッション信号
DISCHRG_VBUS = PHY への放電信号
SESS_END = PHY へのセッション終了信号
CHRG_VBUS = PHY への V_{BUS} 充電信号
DP = データ正ライン
DM = データ負ライン

1. 節電のために、ホストはバスがアイドル状態のときはポート電源をサスペンドおよびオフにします。OTG_FS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割込みレジスタのアーリーサスペンドビットをセットします。これに続いて、OTG_FS コントローラは、コア割込みレジスタの USB サスペンドビットをセットします。OTG_FS コントローラは、PHY に V_{BUS} を放電するように通知します。
2. PHY はデバイスにセッションの終了を知らせます。これが SRP の初期条件です。OTG_FS コントローラは、SRP を初期化する前に、2 ms の SE0 を必要とします。USB 1.1 フルスピードシリアルトランシーバの場合、アプリケーションは、BSVLD (OTG_FS_GOTGCTL) がネゲートされた後、 V_{BUS} が 0.2 V まで放電するまで待つ必要があります。この放電時間の値は、トランシーバのベンダから入手でき、ベンダごとに異なります。
3. USB OTG コアは PHY に V_{BUS} の放電を加速させることを通知します。
4. アプリケーションは、OTG 制御およびステータスレジスタのセッションリクエストビットに書き込むことによって、SRP を開始します。OTG_FS コントローラは、データラインパルシングに続いて V_{BUS} パルシングを実行します。
5. ホストは、データラインパルシングまたは V_{BUS} パルシングから SRP を検出して、 V_{BUS} をオンにします。PHY は、デバイスに V_{BUS} 電源オンを知らせます。
6. OTG_FS コントローラは、 V_{BUS} パルシングを実行します。

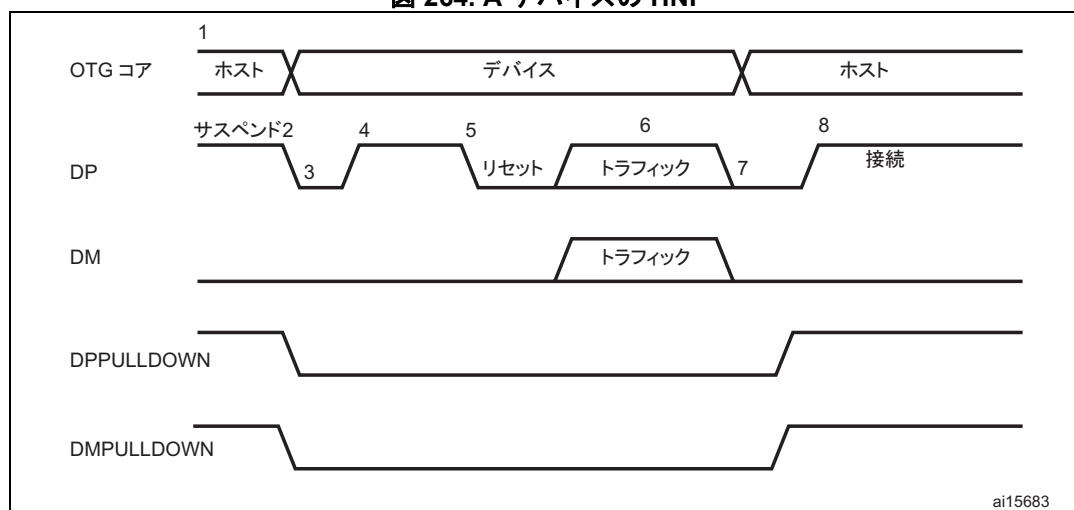
ホストは、 V_{BUS} をオンにして、SRP が成功したことを示すことによって、新しいセッションを開始します。OTG_FS コントローラは、OTG 割込みステータスレジスタのセッションリクエスト成功ステータス変化ビットをセットすることによって、アプリケーションに割込みをかけます。アプリケーションは、OTG 制御およびステータスレジスタのセッションリクエスト成功ビットを読み出します。

7. USB に電源が投入されると、OTG_FS コントローラが接続され、SRP プロセスが完了します。

A デバイスホストネゴシエーションプロトコル

HNP は、USB ホストの役割を A デバイスから B デバイスに切り替えます。アプリケーションは、OTG_FS コントローラが HNP を A デバイスとして実行できるように、コア USB 設定レジスタの HNP 対応ビットをセットする必要があります。

図 264. A デバイスの HNP



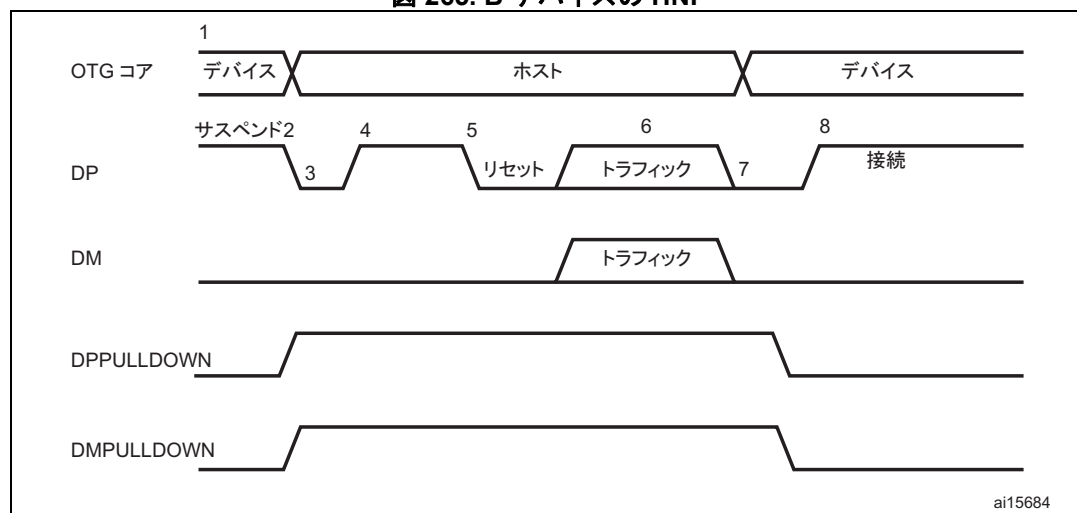
1. DPPULLDOWN = PHY 内部の DP ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
DMPULLDOWN = PHY 内部の DM ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
1. OTG_FS コントローラは、SetFeature b_hnp_enable 記述子を B デバイスに送信して、HNP サポートを有効にします。B デバイスの ACK 応答は、B デバイスが HNP をサポートしていることを示します。アプリケーションは、OTG 制御およびステータスレジスタのセット HNP イネーブルビットをセットして、B デバイスが HNP をサポートしていることを OTG_FS コントローラに示す必要があります。
2. アプリケーションがバスの使用を終了したときには、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによって、サスペンドします。
3. B デバイスが USB サスペンドを検出すると、切断して、HNP の初期条件を示します。B デバイスは、ホストの役割に切り替える必要があるときにのみ、HNP を開始します。そうでないときには、バスはサスペンド状態を継続します。
OTG_FS コントローラは、OTG 割込みステータスレジスタのホストネゴシエーション検出割込みをセットして、HNP の開始を示します。
OTG_FS コントローラは、PHY の DP プルダウンと DM プルダウンをネゲートして、デバイスの役割を示します。PHY は、OTG_FS_DP プルアップ抵抗を有効にして、B デバイスの接続を示します。
アプリケーションは、OTG 制御およびステータスレジスタの現在モードビットを読み出して、デバイスモードの動作を確認する必要があります。
4. B デバイスは接続を検出し、USB リセットを発行し、データトラフィックのために OTG_FS コントローラをエニューメレーションします。

5. B デバイスはホストの役割を継続し、トラフィックを開始し、終了時にはバスをサスペンドします。OTG_FS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割込みレジスタのアーリーサスペンドビットをセットします。これに続いて、OTG_FS コントローラは、コア割込みレジスタの USB サスペンドビットをセットします。
6. ネゴシエートモードでは、OTG_FS コントローラは、サスペンドを検出して切断し、ホストの役割に戻ります。OTG_FS コントローラは、ホストの役割の前提を示すために、PHY の DP プルダウンと DM プルダウンをアサートします。
7. OTG_FS コントローラは、OTG 割込みステータスレジスタのコネクタ ID ステータスレジスタ変更割込みをセットします。アプリケーションは、OTG 制御およびステータスレジスタのコネクタ ID ステータスを読み出して、OTG_FS コントローラの動作が A デバイスであることを確認する必要があります。これは、HNP の完了をアプリケーションに示します。アプリケーションは、OTG 制御およびステータスレジスタの現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
8. B デバイスが接続され、HNP プロセスが完了します。

B デバイスホストネゴシエーションプロトコル

HNP は、USB ホストの役割を B デバイスから A デバイスに切り替えます。アプリケーションは、OTG_FS コントローラが HNP を B デバイスとして実行できるように、コア USB 設定レジスタの HNP 対応ビットをセットする必要があります。

図 265. B デバイスの HNP



1. DPPULLDOWN = PHY 内部の DP ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
DMPULLDOWN = PHY 内部の DM ライン上でプルダウンを有効/無効にするためのコアから PHY への信号
1. A デバイスは、SetFeature b_hnp_enable 記述子を送信して、HNP サポートを有効にします。OTG_FS コントローラの ACK 応答は、それが HNP をサポートしていることを示します。アプリケーションは、OTG 制御およびステータスレジスタのデバイス HNP イネーブルビットをセットして、HNP をサポートしていることを示す必要があります。
アプリケーションは、OTG 制御およびステータスレジスタの HNP リクエストビットをセットして、HNP を開始することを OTG_FS コントローラに示す必要があります。
2. A デバイスがバスの使用を終了したときには、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによってサスペンドします。
OTG_FS コントローラは、バスがアイドル状態になってから 3 ms 後に、コア割込みレジスタのアーリーサスペンドビットをセットします。これに続いて、OTG_FS コントローラは、コア割込みレジスタの USB サスペンドビットをセットします。

OTG_FS コントローラは切断され、A デバイスはバス上で SE0 を検出して HNP を示します。OTG_FS コントローラは、ホストの役割の前提を示すために、PHY の DP プルダウンと DM プルダウンをアサートします。

A デバイスは、SE0 を検出してから 3 ms 以内に OTG_FS_DP プルアップ抵抗を有効化することによって応答します。OTG_FS コントローラはこれを接続として検出します。

OTG_FS コントローラは、OTG 割込みステータスレジスタのホストネゴシエーション成功ステータス変更割込みをセットして、HNP ステータスを示します。アプリケーションは、OTG 制御およびステータスレジスタのホストネゴシエーション成功ビットを読み出して、ホストネゴシエーションの成功を確認する必要があります。アプリケーションは、コア割込みレジスタ (OTG_FS_GINTSTS) の現在モードビットを読み出して、ホストモードの動作を確認する必要があります。

3. アプリケーションはリセットビット (OTG_FS_HPRT の PRST) をセットし、OTG_FS コントローラは USB リセットを発行し、データトラフィックのために A デバイスをエニュメレーションします。
4. OTG_FS コントローラは、トラフィックを開始するというホストの役割を継続し、これが終了すると、ホストポート制御およびステータスレジスタのポートサスペンドビットに書き込むことによってバスをサスペンドします。
5. ネゴシエートモードでは、A デバイスは、サスペンドを検出すると、接続を切断し、ホストの役割に戻ります。OTG_FS コントローラは、PHY の DP プルダウンと DM プルダウンをネゲートして、デバイスの役割の前提を示します。
6. アプリケーションは、コア割込みレジスタ (OTG_FS_GINTSTS) の現在モードビットを読み出して、ホストモードの動作を確認する必要があります。
7. OTG_FS コントローラが接続され、HNP プロセスが完了します。

23 デバッグサポート (DBG)

23.1 概要

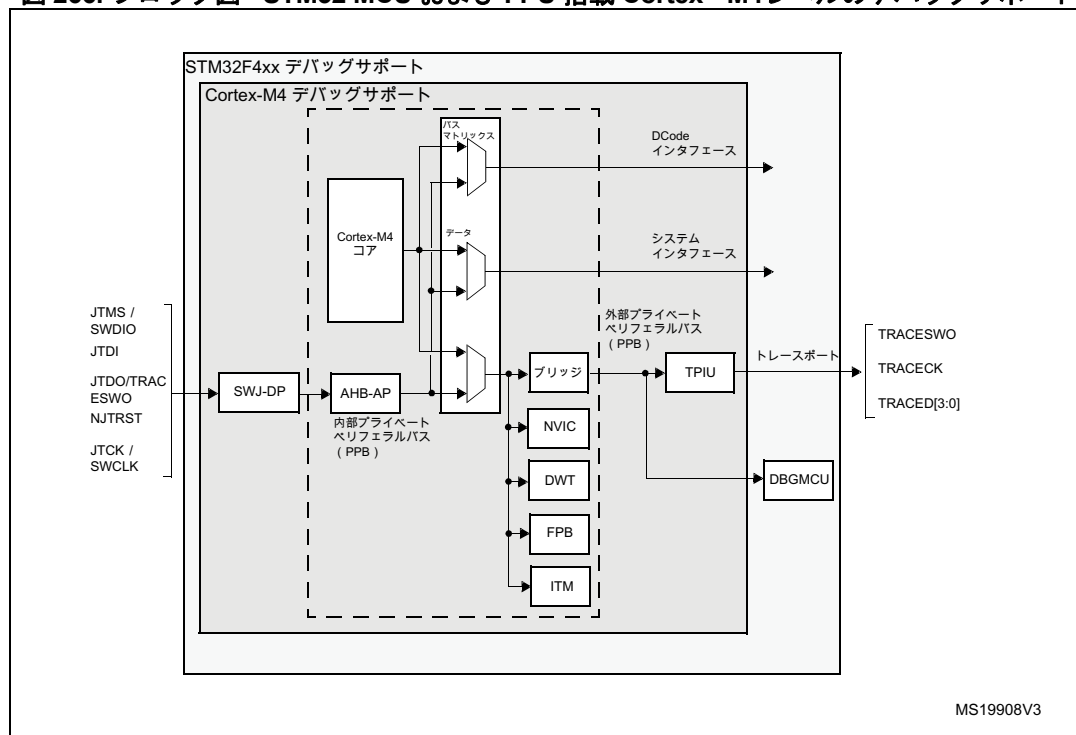
STM32F401xB/C および STM32F401xD/E は、FPU 搭載 Cortex®-M4 コアを中心にして構築されており、高度なデバッグ機能のためのハードウェア拡張機能を含んでいます。デバッグ拡張機能によって、コアは、特定の命令フェッチ時（ブレークポイント）またはデータアクセス時（ウォッチポイント）に停止することができます。停止したとき、コアの内部状態とシステムの外部状態を調査することができます。調査が終わったら、コアとシステムを復元して、プログラム実行を再開することができます。

デバッグ機能は、STM32F401xB/C および STM32F401xD/E の MCU への接続時とデバッグ時にデバッガホスト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 266. ブロック図 - STM32 MCU および FPU 搭載 Cortex®-M4 レベルのデバッグサポート



注： FPU 搭載 Cortex®-M4 コアに内蔵されているデバッグ機能は、ARM® CoreSight Design Kit のサブセットです。

ARM® FPU 搭載 Cortex®-M4 コアは、統合的なオンチップデバッグサポート機能を提供します。この機能は以下の要素で構成されます。

- SWJ-DP : シリアルワイヤ/JTAG デバッグポート
- AHP-AP : AHB アクセスポート
- ITM : 計測トレースマクロセル (Instrumentation Trace Macrocell)
- FPB : フラッシュパッチブレークポイント (Flash Patch Breakpoint)
- DWT : データウォッチポイントトリガ
- 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大型パッケージで使用)

また、STM32F401xB/C および STM32F401xD/E 専用の以下のデバッグ機能も内蔵されています。

- 柔軟性の高いデバッグピンの割り当て
- MCU デバッグボックス (低電力モードのサポート、ペリフェラルクロックの制御など)

注 : ARM® FPU 搭載 Cortex®-M4 コアでサポートされているデバッグ機能の詳細は、FPU 搭載 Cortex®-M4-r0p1 Technical Reference Manual および CoreSight Design Kit-r0p1 TRM を参照してください ([セクション 23.2 : ARM® リファレンス資料](#)を参照)。

23.2 ARM® リファレンス資料

- FPU 搭載 Cortex®-M4 r0p1 Technical Reference Manual (TRM)
(1 ページの関連資料を参照)
- ARM® Debug Interface V5
- ARM® CoreSight Design Kit revision r0p1 Technical Reference Manual

23.3 SWJ デバッグポート (シリアルワイヤと JTAG)

STM32F401xB/C および STM32F401xD/E 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。

Figure 1-10 illustrates the internal pin connections for JTAG and SWD interfaces. The diagram shows the following components and connections:

- External Pins (Left):** JTDO, JTDI, NJTRST, JTMS/SWDIO, JTCK/SWCLK.
- Internal Components:**
 - SWJ-DP:** Contains TDO, TDI, nTRST, TCK, TMS, and nPOTRST.
 - JTAG-DP:** Contains TDO, TDI, nTRST, TCK, TMS, and nPOTRST.
 - SWD/JTAG 選択 (Selection Block):** Receives SWDITMS and TMS signals.
 - SW-DP:** Contains DBGDI, DBGDO, DBGDOEN, and DBGCLK.
- Connections:**
 - JTDO is connected to TDO via a buffer.
 - JTDI is connected to TDI.
 - NJTRST is connected to nTRST.
 - JTMS/SWDIO is connected to SWDITMS and SWDO via a bidirectional buffer.
 - JTCK/SWCLK is connected to SWCLKTCK.
 - Internal logic gates (AND/OR) connect TMS, TCK, and SWDITMS to the selection block and SW-DP.
 - nPOTRST is connected to the selection block and SW-DP.
- Note:** nPOTRST is active from power-on reset (パワーオンリセットから).

797/836

23.4.1 SWJ デバッグポートピン

5 個のピンが SWJ-DP 用の STM32F401xB/C および STM32F401xD/E 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 134. 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

23.4.2 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、STM32F401xB/C および STM32F401xD/E の MCU は SWJ-DP ポートの一部またはすべてを無効にして、汎用 IO (GPIO) に使用する関連ピンを解放することができます。SWJ-DP ポートピンを無効にする方法の詳細については、[セクション 8.3.2: I/O ピンマルチプレクサとマッピング](#)を参照してください。

表 135. 柔軟性の高い SWJ-DP ピンの割り当て

使用可能なデバッグポート	SWJ IO ピンの割り当て				
	PA13 / JTMS / SWDIO	PA14 / JTCK / SWCLK	PA15 / JTDI	PB3 / JTDO	PB4 / NJTRST
全 SWJ (JTAG-DP + SW-DP)、リセット状態	X	X	X	X	X
全 SWJ (JTAG-DP + SW-DP)、NJTRST なし	X	X	X	X	
JTAG-DP 無効、SW-DP 有効	X	X			
JTAG-DP 無効、SW-DP 無効					解放

注： APB ブリッジの書き込みバッファがフルのとき、GPIO_AFR レジスタへの書き込みには APB 1 サイクル分が追加で必要になります。これは、コアの nTRST および TCK 入力信号でのクリーンレベルを保証するために、JTAGSW ピンの無効化が 2 サイクルで行われるからです。

- サイクル 1：コアへの JTAGSW 入力信号は 1 または 0 に（nTRST、TDI、および TMS では 1 に、TCK では 0 に）接続されます。
- サイクル 2：GPIO コントローラが SWJTAG 入出力ピンの制御（方向、プルアップ/ダウン、シュミットトリガの有効化などの制御）信号を受け取ります。

23.4.3 JTAG ピンでの内部プルアップ／プルダウン

JTAG 入力ピンは、デバッグモード機能を制御するためにフリップフロップに直結されます。したがって、JTAG 入力ピンをフロート状態にしないことが必要です。これらのフリップフロップの一部のクロックに直結される SWCLK/TCK ピンについては、特に注意が必要です。

入出力レベルを正しく制御するため、デバイスには内部プルアップ／プルダウンが JTAG 入力ピンに内蔵されています。

NJTRST : 内部プルアップ JTDI : 内部プルアップ JTMS/SWDIO : 内部プルアップ TCK/SWCLK : 内部プルダウン JTAG 入出力がユーザソフトウェアによって解放されると、GPIO コントローラが再び制御権を獲得します。リセット状態では、GPIO 制御レジスタは入出力をこれと同等の状態に設定します。

- NJTRST : AF 入力プルアップ
- JTDI : AF 入力プルアップ
- JTMS/SWDIO : AF 入力プルアップ
- JTCK/SWCLK : AF 入力プルダウン
- JTDO : AF 出力フローティング

ソフトウェアはこれらの入出力を標準の GPIO 信号として使用することができます。

注 : JTAG IEEE 規格では、TDI、TMS、および nTRST にプルアップを追加することを推奨していますが、TCK に関しては特別な推奨はありません。ただし、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。

23.4.4 シリアルワイヤの使用と、未使用のデバッグピンを GPIO として解放する方法

シリアルワイヤ DP を使って GPIO を解放するには、ユーザソフトウェアが GPIO_MODER レジスタで GPIO (PA15、PB3、および PB4) コンフィギュレーションモードを変更する必要があります。これによって PA15、PB3、および PB4 が解放され、GPIO として使用できるようになります。

デバッグ時には、ホストは次の動作を行います。

- システムリセット中に、すべての SWJ ピンが割り当てられます (JTAG-DP + SW-DP)。
- システムリセット中に、デバッグホストは、JTAG-DP から SW-DP に切り替える JTAG シーケンスを送信します。
- システムリセット中に、さらに、デバッグはベクタリセットにブレークポイントを設定します。
- システムリセットは解除され、コアは停止します。
- これ以降のすべてのデバッグ通信は、SW-DP を使用して行われます。残りの JTAG ピンは、ユーザソフトウェアによって GPIO として再割当てできます。

注： ユーザソフトウェアの設計に関しては、次の点に注意してください。

デバッグピンを解放するには、リセット後にユーザソフトウェアがピンを解放するまでの一定の期間、デバッグピンは、まず、入力プルアップ (nTRST、TMS、TDI)、プルダウン (TCK)、または出力トライステート (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。

23.5 STM32F401xB/C および STM32F401xD/E の JTAG TAP 接続

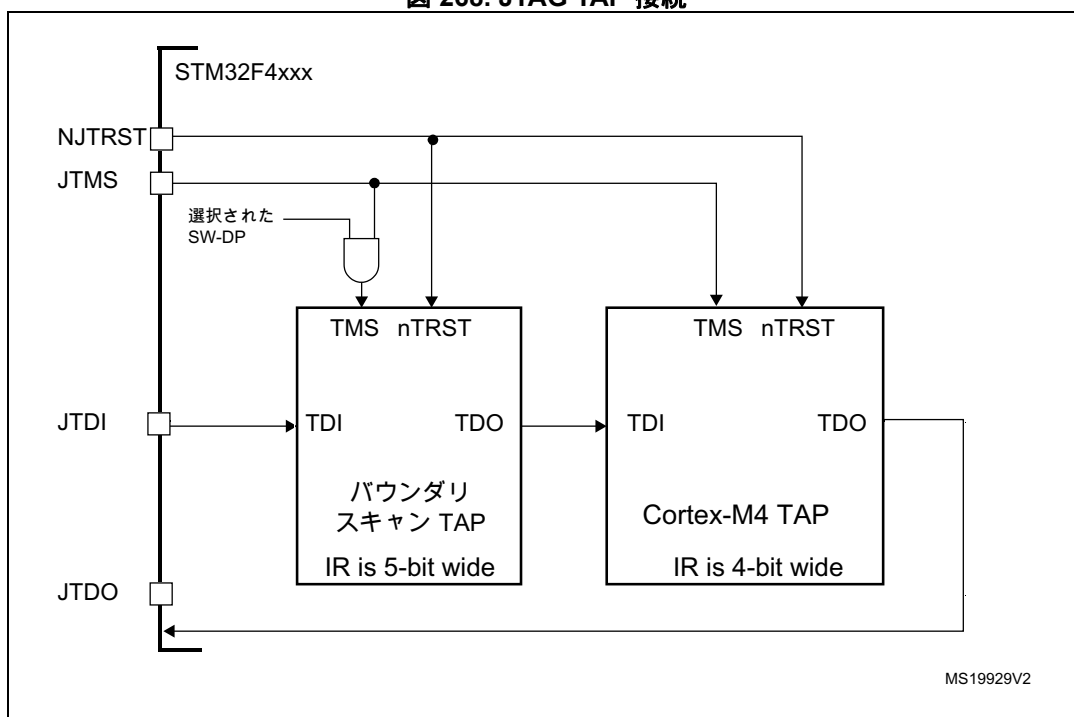
STM32F401xB/C および STM32F401xD/E の MCU は、バウンダリスキャン TAP (IR は 5 ビット幅) と FPU 搭載 Cortex®-M4 TAP (IR は 4 ビット幅) という、シリアル接続された 2 個の JTAG TAP を内蔵しています。

デバッグ目的で FPU 搭載 Cortex®-M4 の TAP にアクセスするには、

1. まず、バウンダリスキャン TAP の BYPASS 命令をシフトする必要があります。
2. 次に、IR シフトごとに、スキャンチェーンには 9 (=5+4) ビットが格納され、未使用の TAP 命令を BYPASS 命令によってシフトインする必要があります。
3. データシフトごとに、BYPASS モードにある未使用の TAP は、データスキャンチェーンに 1 個のデータビットを追加します。

注： 重要：専用の ARM® JTAG シーケンスを使用してシリアルワイヤが選択されると、バウンダリスキャン TAP は自動的に無効になります (JTMS は強制的にハイレベルになります)。

図 268. JTAG TAP 接続



23.6 ID コードとロック機構

STM32F401xB/C および STM32F401xD/E の 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。

23.6.1 MCU デバイス ID コード

STM32F401xB/C および STM32F401xD/E の MCU には MCU ID コードが内蔵されています。この ID は、ST 社製 MCU の部品番号とダイのリビジョンを識別します。これは DBG_MCU 部品の一部であり、外部 PPB バスに配置されます (815 ページのセクション 23.16 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
予約済み				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。

STM32F401xB/C デバイス

0x1000 = リビジョン Z

0x1001 = リビジョン A

STM32F401xD/E デバイス

0x1000 = リビジョン A

0x1001 = リビジョン Z

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DEV_ID(11:0)** : デバイス識別子

デバイス ID は 0x423 (STM32F401xB/C の場合) です。

デバイス ID は 0x433 (STM32F401xD/E の場合) です。

23.6.2 バウンダリスキャン TAP

JTAG ID コード

STM32F401xB/C および STM32F401xD/E BSC (バウンダリスキャン) の TAP は、次の JTAG ID コードを内蔵しています :

- 0x06423041 (STM32F401xB/C の場合)
- 0x06433041 (STM32F401xD/E の場合)

23.6.3 FPU 搭載 Cortex®-M4 TAP

ARM® FPU 搭載 Cortex®-M4 の TAP は JTAG ID コードを内蔵しています。この ID コードは ARM® のデフォルトであり、変更されていません。このコードは、JTAG デバッグポートからのみアクセスできます。

このコードは です (FPU 搭載 Cortex®-M4 r0p1 に対応、[セクション 23.2 : ARM® 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。

23.6.4 FPU 搭載 Cortex®-M4 JEDEC-106 ID コード

ARM® FPU 搭載 Cortex®-M4 は JEDEC-106 ID コードを内蔵しています。これは内部 PPB バスのアドレス 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。

23.7 JTAG デバッグポート

標準的な JTAG ステートマシンは、4 ビット命令レジスタ (IR) と 5 個のデータレジスタを搭載しています (全詳細については、FPU 搭載 Cortex®-M4 r0p1 Technical Reference Manual (TRM) を参照し、参考として[セクション 23.2 : ARM® リファレンス資料](#)を参照)。

表 136. 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 0x4BA00477 (ARM® FPU 搭載 Cortex®-M4r0p1 の ID コード)

表 136. 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
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"> 入力データ転送時 : <ul style="list-style-type: none"> ビット 34:3 = DATA[31:0] = 書き込みリクエスト用に転送する 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 = 読出しリクエスト (1) または書き込みリクエスト (0) 出力データ転送時 : <ul style="list-style-type: none"> ビット 34:3 = DATA[31:0] = 読出しリクエストに続いて読み出される 32 ビットデータ ビット 2:0 = ACK[2:0] = 3 ビット確認応答 : <ul style="list-style-type: none"> 010 = OK/FAULT 001 = WAIT その他 = 予約済み <p>A[3:2] ビットの説明については、表 137 を参照してください。</p>
1011	APACC [35 ビット]	<p>アクセスポートアクセスレジスタ</p> <p>アクセスポートを初期化し、アクセスポートレジスタへのアクセスを可能にします。</p> <ul style="list-style-type: none"> 入力データ転送時 : <ul style="list-style-type: none"> ビット 34:3 = DATA[31:0] = 書き込みリクエスト用にシフトインする 32 ビットデータ ビット 2:1 = A[3:2] = 2 ビットアドレス (サブアドレス AP レジスタ) ビット 0 = RnW = 読出しリクエスト (1) または書き込みリクエスト (0) 出力データ転送時 : <ul style="list-style-type: none"> ビット 34:3 = DATA[31:0] = 読出しリクエストに続いて読み出される 32 ビットデータ ビット 2:0 = ACK[2:0] = 3 ビット確認応答 : <ul style="list-style-type: none"> 010 = OK/FAULT 001 = WAIT その他 = 予約済み <p>次の項目の組み合わせとして、多くの AP レジスタ (AHB-AP を参照) をアドレス指定します。</p> <ul style="list-style-type: none"> シフトされた値 A[3:2] DP SELECT レジスタの現在値
1000	ABORT [35 ビット]	<p>アボートレジスタ</p> <ul style="list-style-type: none"> ビット 31:1 = 予約済み ビット 0 = DAPABORT : DAP アボートを生成するには 1 を書き込みます

表 137. 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	DP CTRL/STAT レジスタ：次の目的で使用されます。 – 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）の読出し
0x8	10	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。 – ビット 31:24 (APSEL)：現在の AP を選択します。 – ビット 23:8：予約済み – ビット 7:4 (APBANKSEL)：現在の AP でアクティブな 4 ワードレジスタウィンドウを選択します。 – ビット 3:0：予約済み
0xC	11	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。

23.8 SW デバッグポート

23.8.1 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）の読出し／書込みが可能です。

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（100 K Ω が ARM® の推奨値）。

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。

23.8.2 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 138. 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 : 書込みリクエスト 1 : 読出しリクエスト
4:3	A[3:2]	DP/AP レジスタのアドレスフィールド (表 137 を参照)
5	Parity	先行するビットの 1 ビットパリティ
6	STOP	0
7	Park	ホストによって駆動されません。プルアップの効果で、ターゲットには「1」と読み出される必要があります。

DPACC レジスタと APACC レジスタの詳細については、FPU 搭載 Cortex®-M4r0p1 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 139. ACK 応答 (3 ビット)

ビット	名前	説明
0..2	ACK	001 : FAULT 010 : WAIT 100 : OK

読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 140. データ転送 (33 ビット)

ビット	名前	説明
0..31	WDATA または RDATA	書込み／読出しデータ
32	Parity	32 データビットの 1 ビットパリティ

読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。

23.8.3 SW-DP ステートマシン (リセット、アイドル状態、ID コード)

SW-DP のステートマシンには、SW-DP を識別する内部 ID コードがあります。これは JEP-106 規格に準じています。この ID コードは、ARM[®] のデフォルトコードであり、**0x2BA01477** (FPU 搭載 Cortex[®]-M4 r0p1 に対応) がセットされています。

注： SW-DP ステートマシンは、ターゲットがこの ID コードを読み出すまで非アクティブであることに注意してください。

- パワーオンリセット後、または DP が JTAG から SWD に切り替えられた後、またはラインが 50 サイクルを超えてハイレベルにあった後では、SW-DP ステートマシンは RESET 状態になります。
- リセット状態のあと、ラインが 2 サイクル以上の間ローレベルであれば、SW-DP ステートマシンはアイドル状態になります。
- リセット状態のあとは、まずアイドル状態に入り、次に DP-SW ID CODE レジスタの読出しアクセスを行う **必要があります**。そうしないと、ターゲットは、他のトランザクションに対して ACK 応答の“FAULT”を発行します。

SW-DP ステートマシンの詳細については、FPU 搭載 Cortex[®]-M4r0p1 TRM および CoreSight Design Kit r0p1 TRM を参照してください。

23.8.4 DP と AP の読出し／書込みアクセス

- DP への読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 への読出しアクセスはポストされます。つまり、アクセスの結果は次の転送時に返されます。次のアクセスが AP アクセスでない場合、結果を得るには DP-RDBUFF レジスタを読み出す必要があります。
AP 読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 読出しアクセスまたは RDBUFF 読出しリクエストのたびに更新されます。
- SW-DP は、DP と AP の両方の書込みに使用できる書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、書込み動作を受け付けることができます。書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 読出し、CTRL/STAT 読出し、または ABORT 書込みは、書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、書込みを内部的に有効にするには、書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。

23.8.5 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 141. SW-DP レジスタ

A[3:2]	読出し／書込み	SELECT レジスタの CTRLSEL ビット	レジスタ	注
00	読出し	-	IDCODE	製造者コードは ST 社のコードではありません。0x2BA01477 (SW-DP を識別)。
00	書込み	-	ABORT	-
01	読出し／書込み	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）の読出し
01	読出し／書込み	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出しデータの復旧を可能にします。
10	書込み		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	読出し／書込み		読出しバッファ	AP アクセスはポストされるため、この読出しバッファは効果的です（AP 読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。

23.8.6 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

23.9 AHB-AP (AHB アクセスポート) - JTAG-DP と SW-DP の両方に有効

機能 :

- システムアクセスはプロセッサステータスから独立しています。
- SW-DP または JTAG-DP が AHB-AP にアクセスします。
- AHB-AP はバスマトリックスの AHB マスタです。したがって、AHB-AP は ICode バスを除くすべてのデータバス (Dcode バス、システムバス、内部/外部の PPB バス) にアクセスできます。
- ビットバンドトランザクションがサポートされます。
- AHB-AP トランザクションは FPB を迂回します。

32 ビット AHB-AP レジスタは 6 ビット幅 (最大 64 ワードまたは 256 バイト) であり、以下の構成となっています。

- c) ビット [7:4] = DP_SELECT レジスタのビット [7:4] APBANKSEL
- d) ビット [3:2] = SW-DP 用の 35 ビットパケットリクエストの 2 ビットアドレス A[3:2]

FPU 搭載 Cortex®-M4 の AHB-AP は、9 個の 32 ビットレジスタを内蔵しています。

表 142. FPU 搭載 Cortex®-M4 AHB-AP レジスタ

アドレス オフセット	レジスタ名	注
0x00	AHB-AP 制御およびステータス ワード	AHB インタフェースを通じて転送を設定および制御します (サイズ、hprot、現在の転送のステータス、アドレスインクリ メントタイプ)。
0x04	AHB-AP 転送アドレス	-
0x0C	AHB-AP データ読み出し/書き込み	-
0x10	AHB-AP バンクデータ 0	転送アドレスレジスタを書き換えずに、4 個のアラインドデー タワードを直接配置します。
0x14	AHB-AP バンクデータ 1	
0x18	AHB-AP バンクデータ 2	
0x1C	AHB-AP バンクデータ 3	
0xF8	AHB-AP デバッグ ROM アドレス	デバッグインタフェースのベースアドレス
0xFC	AHB-AP ID レジスタ	-

詳細については、FPU 搭載 Cortex®-M4r0p1 TRM を参照してください。

23.10 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 143. 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出しおよび書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 TRCENA というビットがあります。

注： **重要**：これらのレジスタは、システムリセットによってはリセットされません。パワーオンリセットによってのみリセットされます。

詳細については、FPU 搭載 Cortex®-M4r0p1 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。

23.11 システムリセット中のデバッグホスト接続機能

STM32F401xB/C および STM32F401xD/E の MCU のリセットシステムは、次のリセットソースから構成されます。

- POR (パワーオンリセット) : パワーアップのたびに RESET をアサートします。
- 内部ウォッチドッグリセット
- ソフトウェアリセット
- 外部リセット

FPU 搭載 Cortex®-M4 では、デバッグ部のリセット (一般に PORRESETn) とそれ以外のリセット (SYSRESETn) が区別されます。

こうすれば、リセットベクタをフェッチするときにコアを停止させるようコアデバッグレジスタをプログラミングすることで、デバッグはシステムリセット中に接続が可能になります。これによってホストはシステムリセットを解除でき、コアは命令を実行しなくても、ただちに停止します。さらに、システムリセット中にデバッグ機能をプログラミングできます。

注 : システムリセット中にデバッグホストを接続する (リセットベクタにブレークポイントを設定する) ことを強く推奨します。

23.12 FPB (フラッシュパッチブレークポイント)

FPB ユニットは、

- ハードウェアブレークポイントを実装します。
- コード空間からシステム空間に、コードとデータをパッチします。この機能によって、コードメモリ空間にあるソフトウェアバグの訂正が可能になることがあります。

ソフトウェアパッチとハードウェアブレークポイントは同時に使用できません。

FPB は以下から構成されます。

- コード空間からのリテラルロードと照合し、システム空間内の対応する領域に再配置する 2 個のリテラルコンパレータ。
- コード空間からの命令フェッチと照合する 6 個の命令コンパレータ。命令コンパレータを使用すると、システム空間内の対応する領域に再配置したり、コアへのブレークポイント命令を生成したりできます。

23.13 DWT (データウォッチポイントトリガ)

DWT ユニットは 4 個のコンパレータから構成されており、次の用途に設定できます。

- ハードウェアウォッチポイント
- ETM へのトリガ
- PC サンプラ
- データアドレスサンプラ

DWT は、プロファイル情報を提供することもできます。このため、次の値を得るためのカウンタにアクセスできます。

- クロックサイクル数
- フォールドされた命令数
- ロードストアユニット (LSU) の動作数
- スリープサイクル数
- CPI (命令あたりのクロック数)
- 割込みオーバーヘッドの回数

23.14 ITM (計測トレースマクロセル)

23.14.1 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込みを行って、パケットを発行できます。
- **ハードウェアトレース**: DWT がパケットを生成し、ITM がそれを発行します。
- **タイムスタンプ**: タイムスタンプはパケットを基準にして発行されます。ITM は、タイムスタンプを生成するための 21 ビットカウンタを内蔵しています。カウンタは、FPU 搭載 Cortex®-M4 クロックまたはシリアルワイヤビューア (SWV) 出力のビットクロックレートによって駆動されます。

ITM によって発行されたパケットは、TPIU (トレースポートインタフェースユニット) に出力されます。TPIU のフォーマットは、いくつかのパケットを追加してから (TPIU を参照)、完全なパケットシーケンスをデバッガホストに出力します。

ITM をプログラミングまたは使用するには、その前に、デバッグ例外/モニタ制御レジスタの TRCEN ビットを有効にする必要があります。

23.14.2 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 144. 主な ITM レジスタ

アドレス	レジスタ	詳細
@E0000FB0	ITM ロックアクセス	他の ITM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
@E0000E80	ITM トレース制御	ビット 31-24 = 常に 0
		ビット 23 = ビジー
		ビット 22-16 = トレースデータのソースを識別する 7 ビットの ATB ID
		ビット 15-10 = 常に 0
		ビット 9:8 = TSPrescale = タイムスタンププリスケラ
		ビット 7-5 = 予約済み
		ビット 4 = SWOENA = SWV 動作 (タイムスタンプカウンタを SWV クロックによって駆動) を有効にします。
		ビット 3 = DWTENA : DWT スティムラスを有効にします。
		ビット 2 = SYNCENA : DWT が同期トリガを生成して TPIU が同期パケットを発行できるようにするには、このビットを 1 にする必要があります。
		ビット 1 = TSENA (タイムスタンプ有効)
		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の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 23.17.2 : TRACE ピンの割当て](#)および [セクション 23.16.3 : 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)。

23.15 ETM (組み込みトレースマクロセル)

23.15.1 概要

ETM はプログラム実行の再構築を有効にします。データはデータウォッチポイントおよびトレース (DWT) コンポーネントまたは計測トレースマクロセル (ITM) を使用してトレースされます。一方、命令は組み込みトレースマクロセル (ETM) を使用してトレースされます。

ETMは、情報をパケットとして送信し、埋め込みリソースによってトリガされます。これらのリソースは個別にプログラミングする必要があります。トリガソースの選択にはトリガイベントレジスタ (0xE0041008) を使用します。イベントは、単一イベント (アドレスコンパレータからのアドレスマッチ) または 2 つのイベント間の論理式とすることができます。トリガソースは、DWT モジュールの 4 番目のコンパレータの 1 つで、次のイベントを監視できます。

- クロックサイクルマッチング
- データアドレスマッチング

トリガリソースの詳細については、[セクション 23.13 : 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 23.17 : 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。

23.15.2 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。

23.15.3 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 145. 主な ETM レジスタ

アドレス	レジスタ	詳細
0xE0041FB0	ETM ロックアクセス	他の ETM レジスタへの書き込みアクセスをロック解除するには、0xC5ACCE55 を書き込みます。
0xE0041000	ETM 制御	このレジスタは、ETM の一般的な動作（たとえば、トレースを有効にする方法）を制御します。
0xE0041010	ETM ステータス	このレジスタは、トレースおよびトリガロジックの現在のステータスに関する情報を提供します。
0xE0041008	ETM トリガイイベント	このレジスタは、トリガを制御するイベントを定義します。
0xE004101C	ETM トレース有効制御	このレジスタは、どのコンパレータが選択されるかを定義します。
0xE0041020	ETM トレース有効イベント	このレジスタは、トレース有効化イベントを定義します。
0xE0041024	ETM トレース開始/停止	このレジスタは、トレースをそれぞれ開始および停止するためにトリガソースによって使用されるトレースを定義します。

23.15.4 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 STM32F401xB/C および STM32F401xD/E の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、ITM 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）。
- トリガイイベントレジスタに 0000406F を書き込みます（トリガイイベントを定義します）。
- 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）。

23.16 MCU デバッグコンポーネント (DBGMCU)

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

- 低電力モード
- ブレークポイントにおける、タイマ、ウォッチドッグ、および I²C のクロック制御
- TRACE ピンの割当て制御

23.16.1 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。

23.16.2 タイマ、ウォッチドッグ、および I²C のデバッグサポート

ブレークポイントにおいては、以下に示すタイマのカウンタやウォッチドッグの動作方法を選択する必要があります。

- ブレークポイントの中でもカウントを継続できます。この動作は、たとえば、PWM がモータを制御しているときに一般的に必要です。
- ブレークポイントの中でカウントを停止できます。この動作はウォッチドッグ用に必要です。

I²C の場合、ブレークポイントにおいては SMBUS タイムアウトをブロックするように選択できます。

23.16.3 デバッグ MCU 設定レジスタ

このレジスタを使用して、デバッグ中に MCU を設定できます。次の設定が可能です。

- 低電力モードのサポート
- タイマおよびウォッチドッグカウンタのサポート
- TRACE ピンの割当て

この DBGMCU_CR は、アドレス 0xE0042004 にある外部 PPB バスに配置されます。

このレジスタは PORESET によって非同期的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込みが可能です。

DBGMCU_CR レジスタ

アドレス : 0xE004 2004

32 ビットアクセスのみサポートされます。

POR リセット : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								TRACE_ MODE [1:0]		TRACE_ I OEN	予約済み	DBG_ STANDB Y	DBG_ STOP	DBG_ SLEEP	
								rw	rw	rw		rw	rw	rw	

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:5 **TRACE_MODE[1:0]** および **TRACE_IOEN** : TRACE ピン割り当て制御

- TRACE_IOEN=0 の場合 :
TRACE_MODE=xx : TRACE ピンは割り当てられません (デフォルト)。
- TRACE_IOEN=1 の場合 :
 - TRACE_MODE=00 : 非同期モードの TRACE ピン割り当てです。
 - TRACE_MODE=01 : TRACEDATA サイズが 1 の同期モードの TRACE ピン割り当てです。
 - TRACE_MODE=10 : TRACEDATA サイズが 2 の同期モードの TRACE ピン割り当てです。
 - TRACE_MODE=11 : TRACEDATA サイズが 4 の同期モードの TRACE ピン割り当てです。

ビット 4:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG_STANDBY** : デバッグ STANDBY モード

- 0 : (FCLK=オフ、HCLK=オフ) デジタル部全体が電源オフになります。
ソフトウェアから見て、STANDBY モードから抜けることは、リセットベクタのフェッチと同じになります (ただし、いくつかのステータスビットは MCU が STANDBY モードから再開していることを示す)。
- 1 : (FCLK=オン、HCLK=オン) この場合、デジタル部は電源オフ状態ではなく、FCLK と HCLK は引き続きアクティブ状態の内部 RC オシレータから供給されます。さらに、MCU は STANDBY モード中にシステムリセットを生成するため、STANDBY モードから抜けることはリセットからのフェッチと同じになります。

ビット 1 **DBG_STOP** : デバッグ STOP モード

- 0 : (FCLK=オフ、HCLK=オフ) STOP モードでは、クロックコントローラがすべてのクロック (HCLK と FCLK を含む) を無効にします。STOP モードから抜けると、クロック設定はリセット後の場合と同じになります (CPU は 8 MHz の内部 RC オシレータ (HSI) から供給されます)。したがって、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります。
- 1 : (FCLK=オン、HCLK=オン) この場合、STOP モードに入ると、FCLK と HCLK は STOP モードでもアクティブ状態の内部 RC オシレータから供給されます。STOP モードから抜けるとき、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります (DBG_STOP=0 の場合と同様です)。

ビット 0 **DBG_SLEEP** : デバッグ SLEEP モード

- 0 : (FCLK=オン、HCLK=オフ) SLEEP モードでは、FCLK はソフトウェアによって事前に設定されたシステムクロックによって駆動され、HCLK は無効にされます。
- SLEEP モードでは、クロックコントローラの設定はリセットされず、事前にプログラミングされた状態のままです。したがって、SLEEP モードから抜けるときに、ソフトウェアでクロックコントローラを再設定する必要はありません。
- 1 : (FCLK=オン、HCLK=オン) この場合、SLEEP モードに入ると、HCLK には FCLK と同じクロック (ソフトウェアによって事前に設定されたシステムクロック) が供給されます。

23.16.4 デバッグ MCU APB1 凍結レジスタ (DBGMCU_APB1_FZ)

DBGMCU_APB1_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB1 ペリフェラルと関係のあるレジスタで、アドレス 0xE004 2008 にある外部 PPB バスに配置されています。

このレジスタは PORによって非同期にリセットされます（システムリセットではリセットされません）。システムリセット中に、デバッグによって書込みが可能です。

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
予約済み								DBG_I2C3_SMBUS_TIMEOUT	DBG_I2C2_SMBUS_TIMEOUT	DBG_I2C1_SMBUS_TIMEOUT	予約済み				
								RW	RW	RW					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み				DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	予約済み					DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP

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 **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 タイムアウトは停止されます。

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
0 : 通常モードと同じ動作です。
1 : SMBUS タイムアウトは停止されます。

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 タイムアウトは停止されます。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DBG_IWDG_STOP** : コア停止時にデバッグ独立型ウォッチドッグは停止
0 : コアが停止しても独立型ウォッチドッグカウンタのクロックは継続されます。
1 : コア停止時に独立型ウォッチドッグカウンタのクロックは停止します。

ビット 11 **DBG_WWDG_STOP** : コア停止時にデバッグウィンドウ型ウォッチドッグは停止
0 : コアが停止してもウィンドウ型ウォッチドッグカウンタのクロックは継続されます。
1 : コア停止時にウィンドウ型ウォッチドッグカウンタのクロックは停止します。

- ビット 10 **DBG_RTC_STOP** : コア停止時は RTC 停止
- 0 : コアが停止した場合も RTC カウンタのクロックは継続されます。
 - 1 : コア停止時に RTC カウンタのクロックは停止します。
- ビット 9:4 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 **DBG_TIMx_STOP** : コア停止時に TIMx カウンタは停止 (x=2..5)
- 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。

23.16.5 デバッグ MCU APB2 フリーズレジスタ (DBGMCU_APB2_FZ)

DBGMCU_APB2_FZ レジスタは、デバッグにおいて MCU の設定に使用します。APB2 ペリフェラルと関係のあるレジスタで、

アドレス 0xE004 200C にある外部 PPB バスに配置されています。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書込みが可能です。

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
予約済み													DBG_TIM11_STOP	DBG_TIM10_STOP	DBG_TIM9_STOP
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み														DBG_TIM1_STOP	
														rw	

- ビット 31:19 予約済みであり、リセット値に保持する必要があります。
- ビット 18:16 **DBG_TIMx_STOP** : コア停止時に TIMx カウンタは停止 (x=9..11)
- 0 : コアが停止した場合も当該タイマカウンタのクロックは供給されます。
 - 1 : コア停止時に当該タイマカウンタのクロックは停止します。
- ビット 15:1 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 **DBG_TIM1_STOP** : コア停止時に TIM1 カウンタは停止
- 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。

23.17 TPIU (トレースポートインタフェースユニット)

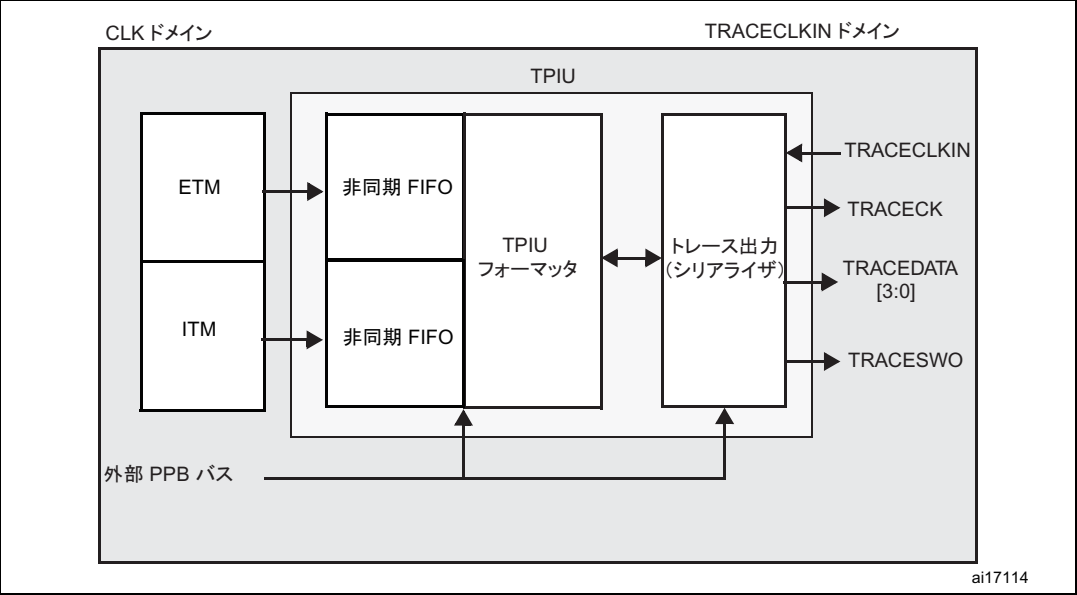
23.17.1 概要

TPIU は、ITM と ETM からオンチップトレースデータ間のブリッジとしての役割を果たします。

出力データストリームによってカプセル化されたトレースソース ID は、トレースポートアナライザ (TPA) によってキャプチャされます。

コア には、特に低コストのデバッグ向けの単純な TPIU (特別バージョンの CoreSight TPIU から構成) が内蔵されています。

図 269. TPIU ブロック図



23.17.2 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 146. 非同期 TRACE ピンの割当て

TPIU ピン名	TRACE 同期モード		ピン割り当て
	タイプ	説明	
TRACESWO	O	TRACE 非同期データ出力	PB3

- 同期モード
同期モードは、データトレースサイズに応じて 2~6 本のピンを追加で必要とし、大型パッケージでのみ使用できます。さらに、JTAG モードとシリアルワイヤモードでも使用でき、非同期トレースよりも優れた帯域幅出力機能を提供します。

表 147. 同期 TRACE ピンの割当て

TPIU ピン名	TRACE 同期モード		STM32F401xB/C および STM32F401xB/C の ピンの割り当て
	タイプ	説明	
TRACECK	O	TRACE クロック	PE2
TRACED[3:0]	O	TRACE 同期データ出力 1、2、または 4 とすることが出来ます。	PE[6:3]

TPIU TRACE ピンの割り当て

デフォルトでは、これらのピンは割り当てられません。これらのピンを割り当てるには、TRACE_IOEN および TRACE_MODE ビット (**MCU デバッグコンポーネント設定レジスタ**内) をセットします。この設定はデバッグホストで行う必要があります。

さらに、割り当てるピン数は、トレースの設定（非同期／同期）によって異なります。

- **非同期モード**：1 本のピンが追加で必要です。
- **同期モード**：データトレースポートレジスタのサイズ（1、2、または 4）に応じて、2～5 本のピンが追加で必要です。
 - TRACECK
 - TRACED(0)、ポートサイズが 1、2、または 4 に設定された場合
 - TRACED(1)、ポートサイズが 2 または 4 に設定された場合
 - TRACED(2)、ポートサイズが 4 に設定された場合
 - TRACED(3)、ポートサイズが 4 に設定された場合

TRACE ピンを割り当てるには、デバッグホストは、デバッグ MCU 設定レジスタ (DBGMCU_CR) の TRACE_IOEN ビットと TRACE_MODE[1:0] ビットをプログラミングする必要があります。デフォルトでは、TRACE ピンは割り当てられません。

このレジスタは、外部 PPB に配置され、PORESET によってリセットされます（システムリセットではありません）。このレジスタは、システムリセット中にデバッグによる書込みができます。

表 148. 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 ⁽¹⁾	-				
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 ⁽¹⁾	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 入力クロックは GND に接続されています。このクロックは、TRACE_IOEN ビットがセットされてから 2 クロックサイクル後に HCLK に割り当てられます。

デバッグは、TPIU の SPP_R (選択ピンプロトコル) レジスタの PROTOCOL[1:0] ビットに書き込むことによって、トレースモードをプログラミングする必要があります。

- PROTOCOL=00 : トレースポートモード (同期モード)。
- PROTOCOL = 01 または 10 : シリアルワイヤ (マンチェスタまたは NRZ) モード (非同期モード)。デフォルト状態は 01 です。

次に、TPIU の CPSPS_R (カレント同期化ポートサイズレジスタ) のビット [3:0] に書き込むことによって TRACE ポートサイズも設定します。

- 0x1 : 1 ピンの場合 (デフォルト状態)
- 0x2 : 2 ピンの場合
- 0x8 : 4 ピンの場合

23.17.3 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。

23.17.4 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。

23.17.5 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)。

23.17.6 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。

23.17.7 非同期モード

これは 1 ピン (非同期出力ピン TRACESWO) のみを使用してトレースを出力する低コストな方法ですが、明らかにその帯域幅は制限されます。

SW-DP ピンを使用する場合、TRACESWO ピンは JTDO ピンと多重化されます。このように、この機能はすべての STM32F401xB/C および STM32F401xD/E 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。

23.17.8 TRACECLKIN 接続 (STM32F401xB/C および STM32F401xD/E 内)

STM32F401xB/C および STM32F401xD/E 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。

STM32F401xB/C および STM32F401xD/E の 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。

23.17.9 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 149. 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</p> <p>ビット 1 : ポートサイズ = 2</p> <p>ビット 2 : ポートサイズ = 3、サポートされません</p> <p>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</p> <p>01 : シリアルワイヤ出力 - マンチェスタ (デフォルト値)</p> <p>10 : シリアルワイヤ出力 - NRZ</p> <p>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"</p> <p>ビット 8 = TrgIn = 常に "1" で、トリガが指定されたことを示します</p> <p>ビット 7-4 = 常に 0</p> <p>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 <> 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	FPU 搭載 Cortex®-M4 では使用されず、常に 0x00000008 が読み出されます。

23.17.10 設定例

- デバッグ例外およびモニタ制御レジスタ (DEMCR) の TRCENA ビットをセットします。
- TPIU カレントポートサイズレジスタに希望する値を書き込みます (デフォルトは 0x1 で 1 ビットのポートサイズ)。
- TPIU フォーマッタおよびフラッシュ制御レジスタに 0x102 (デフォルト値) を書き込みます。
- TPIU 選択ピンプロトコルレジスタに書き込んで同期／非同期モードを選択します。例: 非同期 NRZ モード (UART に類似) の場合は 0x2 を書き込みます。
- DBGMCU 制御レジスタに 0x20 (IO_TRACEN ビット) を書き込み、TRACE I/O を非同期モードとします。このとき、TPIU 同期パケット (FF_FF_FF_7F) が発行されます。
- ITM を設定し、ITM スティムラスレジスタに書き込んで値を出力します。

23.18 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 150. DBG レジスタマップとリセット値

アドレス	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
0xE004 2000	DBGMCU_IDCODE	REV_ID																予約済み		DEV_ID																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
	リセット値 ⁽¹⁾	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0xE004 2004	DBGMCU_CR	予約済み														DBG_TIM5_STOP		予約済み		DBG_I2C2_SMBUS_TIMEOUT		予約済み												TRACE_MODE[1:0]		TRACE_IOEN		予約済み		DBG_STANDBY		DBG_STOP		DBG_SLEEP																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
	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																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
0xE004 2008	DBGMCU_APB1_FZ	予約済み									DBG_I2C3_SMBUS_TIMEOUT			予約済み			DBG_I2C2_SMBUS_TIMEOUT			予約済み			DBG_I2C1_SMBUS_TIMEOUT			予約済み			DBG_IWDG_STOP			DBG_WWDG_STOP			予約済み		DBG_RTC_STOP			予約済み																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
	リセット値										0	0	0				0	0	0				0	0	0				0	0	0	0	0	0			0	0	0													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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. 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 23.6.1 : MCU デバイス ID コード](#)を参照してください。

24 デバイス電子署名

電子署名は、フラッシュメモリ領域に格納され、JTAG/SWD または CPU を使用して読み出すことができます。電子署名には、出荷時にプログラミングされた識別データが含まれています。このデータを使用すれば、ユーザファームウェアやその他の外部デバイスは、そのインタフェースを STM32F4xx マイクロコントローラの特性に自動的に整合させることができます。

24.1 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として使用
- セキュアなブートプロセスなどの起動に使用

96 ビットのユニークデバイス識別子は、状況やデバイスの違いとは無関係にユニークなリファレンス番号を提供します。ユーザは、これらのビットを変更できません。

96 ビットのユニークデバイス識別子は、さまざまな方法で 1 バイト/ハーフワード/ワード単位で読出し、カスタムアルゴリズムを使用して連結することもできます。

ベースアドレス : 0x1FFF 7A10

アドレスオフセット : 0x00

読出し専用 = 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	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(31:0)																															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **U_ID(31:0)** : 31:0 ユニーク ID ビット

アドレスオフセット : 0x04

読出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID(63:48)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(47:32)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **U_ID(63:32)** : 63:32 ユニーク ID ビット

アドレスオフセット : 0x08

読出し専用 = 0xXXXX XXXX (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID(95:80)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(79:64)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **U_ID(95:64)** : 95:64 ユニーク ID ビット

24.2 フラッシュサイズ

ベースアドレス : 0x1FFF 7A22

アドレスオフセット : 0x00

読出し専用 = 0xXXXX (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **F_ID(15:0)** : フラッシュメモリサイズ

このビットフィールドは、デバイスのフラッシュメモリサイズを KB 単位で示します。

たとえば、0x0400 は 1024 KB に対応します。

索引

A

ADC_CCR	238
ADC_CR1	227
ADC_CR2	229
ADC_DR	237
ADC_HTR	233
ADC_JDRx	237
ADC_JOFRx	232
ADC_JSQR	236
ADC_LTR	233
ADC_SMPR1	231
ADC_SMPR2	232
ADC_SQR1	234
ADC_SQR2	234
ADC_SQR3	235
ADC_SR	226

C

CRC_DR	69
CRC_IDR	69

D

DBGMCU_APB1_FZ	818
DBGMCU_APB2_FZ	819
DBGMCU_CR	816
DBGMCU_IDCODE	802
DMA_HIFCR	188
DMA_HISR	187
DMA_LIFCR	188
DMA_LISR	186
DMA_SxCR	189
DMA_SxFCR	194
DMA_SxM0AR	193
DMA_SxM1AR	194
DMA_SxNDTR	192
DMA_SxPAR	193

E

EXTI_EMR	206
EXTI_FTSR	207
EXTI_IMR	206
EXTI_PR	208
EXTI_RTSR	207
EXTI_SWIER	208

F

FLITF_FCR	63
FLITF_FKEYR	61
FLITF_FOPTCR	65
FLITF_FOPTKEYR	61
FLITF_FSR	62

G

GPIOx_AFRH	162
GPIOx_AFRL	161
GPIOx_BSRR	160
GPIOx_IDR	159
GPIOx_LCKR	160
GPIOx_MODER	157
GPIOx_ODR	159
GPIOx_OSPEEDR	158
GPIOx_OTYPER	157
GPIOx_PUPDR	158

I

I2C_CCR	496
I2C_CR1	486
I2C_CR2	488
I2C_DR	491
I2C_OAR1	490
I2C_OAR2	490
I2C_SR1	491
I2C_SR2	495
I2C_TRISE	497
IWDG_KR	415
IWDG_PR	415
IWDG_RLR	416
IWDG_SR	416

O

OTG_FS_CID	710
OTG_FS_DAIN	726
OTG_FS_DAINMSK	726
OTG_FS_DCFG	721
OTG_FS_DCTL	722
OTG_FS_DIEPCTL0	728
OTG_FS_DIEPEMPMSK	728
OTG_FS_DIEPINTx	736
OTG_FS_DIEPMSK	724
OTG_FS_DIEPTSIZ0	738
OTG_FS_DIEPTSIZx	740
OTG_FS_DIEPTXFx	710
OTG_FS_DOEPCTL0	732
OTG_FS_DOEPCTLx	734

OTG_FS_DOEPINTx	737
OTG_FS_DOEPMSK	725
OTG_FS_DOEPSIZ0	739
OTG_FS_DOEPSIZx	741
OTG_FS_DSTS	723
OTG_FS_DTXFSTSx	741
OTG_FS_DVBUSDIS	727
OTG_FS_DVBUSPULSE	727
OTG_FS_GAHBCFG	693
OTG_FS_GCCFG	709
OTG_FS_GINTMSK	702
OTG_FS_GINTSTS	698
OTG_FS_GNPTXFSIZ	707
OTG_FS_GNPTXSTS	708
OTG_FS_GOTGCTL	690
OTG_FS_GOTGINT	692
OTG_FS_GRSTCTL	696
OTG_FS_GRXFSIZ	706
OTG_FS_GRXSTSP	705
OTG_FS_GRXSTSR	705
OTG_FS_GUSBCFG	694
OTG_FS_HAINT	714
OTG_FS_HAINTMSK	714
OTG_FS_HCCHARx	717
OTG_FS_HCFG	711
OTG_FS_HCINTMSKx	719
OTG_FS_HCINTx	718
OTG_FS_HCTSIZx	720
OTG_FS_HFIR	712
OTG_FS_HFNUM	712
OTG_FS_HPRT	715
OTG_FS_HPTXFSIZ	710
OTG_FS_HPTXSTS	713
OTG_FS_PCGCCTL	742

P

PWR_CR	87
PWR_CSR	89

R

RCC_AHB1ENR	118
RCC_AHB1LPENR	124
RCC_AHB1RSTR	112
RCC_AHB2ENR	119
RCC_AHB2LPENR	125
RCC_AHB2RSTR	114
RCC_APB1ENR	119
RCC_APB1LPENR	126
RCC_APB2ENR	122
RCC_APB2LPENR	128
RCC_BDCR	130

RCC_CFGR	107
RCC_CIR	110
RCC_CR	103
RCC_CSR	131
RCC_PLLCFGR	105, 134
RCC_SSCGR	133
RTC_ALRMAR	451
RTC_ALRMBR	452
RTC_ALRMBSSR	462
RTC_BKxR	463
RTC_CALIBR	450
RTC_CALR	458
RTC_CR	444
RTC_DR	443
RTC_ISR	447
RTC_PRER	449
RTC_SHIFTR	455
RTC_SSR	454
RTC_TR	442
RTC_TSDR	456
RTC_TSSSR	457
RTC_TSTR	456
RTC_WPR	453
RTC_WUTR	450

S

SDIO_CLKCR	646
SDIO_DCOUNT	652
SDIO_DCTRL	651
SDIO_DLEN	650
SDIO_DTIMER	650
SDIO_FIFO	659
SDIO_FIFOCNT	659
SDIO_ICR	654
SDIO_MASK	656
SDIO_POWER	645
SDIO_RESPCMD	649
SDIO_RESPx	649
SDIO_STA	653
SPI_CR1	595
SPI_CR2	597
SPI_CRCPR	600
SPI_DR	599
SPI_I2SCFGR	601
SPI_I2SPR	603
SPI_RXCR	600
SPI_SR	598
SPI_TXCR	601
SYSCFG_EXTICR1	141
SYSCFG_EXTICR2	141
SYSCFG_EXTICR3	142

SYSCFG_EXTICR4 142
SYSCFG_MEMRMP 139

T

TIM2_OR 366
TIM5_OR 366
TIMx_ARR 362, 399, 409
TIMx_BDTR 304
TIMx_CCER 297, 359, 398, 408
TIMx_CCMR1 293, 355, 395, 405
TIMx_CCMR2 296, 358
TIMx_CCR1 302, 362, 400, 410
TIMx_CCR2 303, 363, 400
TIMx_CCR3 303, 363
TIMx_CCR4 304, 364
TIMx_CNT 301, 361, 399, 409
TIMx_CR1 283, 346, 389, 402
TIMx_CR2 284, 348
TIMx_DCR 306, 364
TIMx_DIER 289, 351, 391, 403
TIMx_DMAR 307, 365
TIMx_EGR 292, 354, 394, 404
TIMx_PSC 301, 361, 399, 409
TIMx_RCR 302
TIMx_SMCR 287, 349, 390
TIMx_SR 290, 352, 392, 403

U

USART_BRR 543
USART_CR1 544
USART_CR2 546
USART_CR3 547
USART_DR 543
USART_GTPR 549
USART_SR 540

W

WWDG_CFR 423
WWDG_CR 422
WWDG_SR 423

改版履歴

表 151. 文書改版履歴

日付	バージョン	変更内容
2013 年 10 月 7 日	1	初版発行。
2014 年 1 月 30 日	2	<p>マニュアル全体に STM32F401xD/E デバイスに関する情報を追加。</p> <p>表 4 : メモリ割当てとブートモード／物理的な再割当て STM32F401xD/E を追加</p> <p>表 5 : フラッシュモジュールの構成 (STM32F401xB/C および STM32F401xD/E) を更新。</p> <p>セクション 3.8.5 : フラッシュ制御レジスタ (FLASH_CR) を更新。</p> <p>セクション 3.8.6 : フラッシュオプション制御レジスタ (FLASH_OPTCR) を更新。</p> <p>TIM1 :</p> <p>表 50 : TIMx 内部トリガ接続を更新。</p> <p>TIM9 から TIM11 :</p> <p>表 57 : 391 ページのTIMx 内部トリガ接続を追加</p> <p>DMA :</p> <p>セクション 9.2 : DMA の主な機能の内容を修正</p> <p>セクション 9.3.7 : ポインタのインクリメントの文章を削除</p> <p>セクション 9.3.11 : シングル転送とバースト転送の注を削除</p> <p>割込み :</p> <p>表 38 : 200 ページのSTM32F401xB/CSTM32F401xD/Eのベクタテーブルの MemManage 例外の説明を更新</p> <p>RTC :</p> <p>セクション 17.6.3 : RTC 制御レジスタ (RTC_CR) のビット 9 の名前を訂正</p> <p>セクション 17.3.11 : RTC の高精度デジタル較正の内容を修正</p> <p>I2C :</p> <p>セクション 18 : I²C (Inter-integrated circuit) インタフェースの標準モードと高速モードの略記を Sm と Fm に変更</p> <p>DEBUG :</p> <p>REV_ID を更新 (セクション : DBGMCU_IDCODE)。</p>
2014 年 6 月 3 日	3	<p>セクション : 全体消去の条件付きテキストの内容を修正。注を追加</p> <p>セクション 3.6 : オプションバイトを更新。</p>

表 151. 文書改版履歴（続き）

日付	バージョン	変更内容
2015 年 4 月 9 日	4	<p>PWR : 表 15 : 低電力モードの概要を更新して ISR 終了 を移行条件として追加 セクション : 低電力モードへの移行およびセクション : 低電力モードの終了 を追加。 セクション : SLEEP モードへの移行、セクション : SLEEP モードの終了、 表 16: Sleep-now への移行と終了 および 表 17: Sleep-on-exit への移行と終 了を更新。 セクション : STOP モードへの移行、セクション : STOP モードの終了およ び 表 19 : STOP モードへの移行と終了を更新。 セクション : STANDBY モードへの移行、セクション : STANDBY モードの 終了および表 20 : STANDBY モードへの移行と終了を更新。 表 20 : STANDBY モードへの移行と終了のSTANDBYモード移行シーケンス を更新して WUF ビット (PWR_CSR) を CWUF (PWR_CR) に変更</p> <p>RCC : セクション 6.3.18 : RCC クロック制御およびステータスレジスタ (RCC_CSR) のビット 24 および 25:31 のアクセスタイプを変更</p> <p>DMA : セクション 9.5.10 : DMA ストリーム x FIFO 制御レジスタ (DMA_SxFCR) (x=0~7) の FTH[1:0] ビットの説明を更新</p> <p>TIM1 : 表 50 : TIMx 内部トリガ接続を更新。 セクション 12.4.2 : TIM1 制御レジスタ 2 (TIMx_CR2) の MMS ビットのス レーブクロックに関する注を追加 セクション 12.4.3 : TIM1 のスレーブモード制御レジスタ (TIMx_SMCR) の SMS ビットの説明を更新して、スレーブクロックに関する注を追加</p> <p>TIM2/5 : 表 54 : TIMx 内部トリガ接続を更新。 セクション 13.3.15 : タイマの同期のスレーブタイマクロックに関する注を 追加 セクション 13.4.2 : TIMx 制御レジスタ 2 (TIMx_CR2) の MMS ビットのス レーブクロックに関する注を追加 セクション 13.4.3 : TIMx スレーブモード制御レジスタ (TIMx_SMCR) の SMS ビットの説明を更新して、スレーブクロックに関する注を追加 セクション 13.4.10: TIMx カウンタ (TIMx_CNT) およびセクション 13.4.12: TIMx 自動再ロードレジスタ (TIMx_ARR) のレジスタのフォーマットを 16 ビットではなく 32 ビットに変更</p> <p>TIM10/11 : 表 57 : TIMx 内部トリガ接続を更新。 セクション 14.5.2 : TIM10/11 割込み有効レジスタ (TIMx_DIER) を追加。</p>

表 151. 文書改版履歴（続き）

日付	バージョン	変更内容
2015 年 4 月 30 日	4	<p>WWDG 図 157: ウォッチドッグのブロック図 および セクション 16.4: ウォッチドッグタイムアウトをプログラムする方法を更新。</p> <p>WDGLS : 表 62 : 最小／最大 IWDG タイムアウト期間 (32 kHz) (LSI)にある 1. の注を更新</p> <p>I2C : セクション 18.6.2 : I²C 制御レジスタ 2 (I2C_CR2) の FREQ[5:0] ビットフィールドの説明を更新</p> <p>USART : セクション : DMA を使用した受信の RXNEIE に関する注を削除</p> <p>USB_OTG : セクション : OTG_FS USB 設定レジスタ (OTG_FS_GUSBCFG) の TRDT ビットの説明を更新して 表 131 : TRDT 値を追加</p> <p>DEBUG : セクション : DBGMCU_IDCODEの REV_ID ビットを更新 セクション 23.16.5 : デバッグ MCU APB2 フリーズレジスタ (DBGMCU_APB2_FZ) から DB G_TIM8_STOP ビットを削除</p>

表 152. 日本語版文書改版履歴

日付	バージョン	変更内容
2018 年 10 月	1	日本語版 初版発行

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。

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。

© 2018 STMicroelectronics - All rights reserved