



#### 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。STM8TL5xxx マイクロコントローラファミリのメモリおよびペリフェラルの使用方法について詳細に説明しています。

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

STM8 SWIM 通信プロトコルとデバッグモジュールについては、ユーザマニュアル (UM0470) を参照してください。

フラッシュメモリ関連の詳細については、STM8TL5xxx フラッシュメモリプログラミングマニュアル (PM0212) を参照してください。

STM8 コアについては、STM8 CPU プログラミングマニュアル (PM0044) を参照してください。

## 目次

<b>1</b>	<b>メモリおよびレジスタマップ</b> .....	<b>18</b>
1.1	レジスタの説明の略語 .....	18
<b>2</b>	<b>中央演算処理装置（CPU）</b> .....	<b>19</b>
2.1	概要 .....	19
2.2	CPU レジスタ .....	19
2.2.1	CPU レジスタの説明 .....	19
2.2.2	STM8 CPU レジスタマップ .....	23
2.3	グローバル設定レジスタ（CFG_GCR） .....	24
2.3.1	起動レベル .....	24
2.3.2	SWIM ディセーブル .....	24
2.3.3	グローバル設定レジスタの説明（CFG_GCR） .....	24
2.3.4	グローバル設定レジスタマップとリセット値 .....	24
<b>3</b>	<b>シングルワイヤインタフェースモジュール（SWIM）および デバッグモジュール（DM）</b> .....	<b>25</b>
3.1	概要 .....	25
3.2	主な特長 .....	25
3.3	SWIM モード .....	25
<b>4</b>	<b>フラッシュプログラムメモリとデータ EEPROM</b> .....	<b>26</b>
4.1	概要 .....	26
4.2	用語 .....	26
4.3	メインフラッシュメモリの機能 .....	26
4.4	メモリ構成 .....	27
4.4.1	独自仕様コード領域（PCODE） .....	28
4.4.2	ユーザブート領域（UBC） .....	28
4.4.3	データ EEPROM（DATA） .....	29
4.4.4	メインプログラム領域 .....	29
4.4.5	オプションバイト .....	29
4.5	メモリ保護 .....	30
4.5.1	読み出し保護 .....	30
4.5.2	メモリアクセスセキュリティシステム（MASS） .....	30

4.6	メモリのプログラミング	32
4.6.1	バイトのプログラミング	32
4.6.2	ワードプログラミング	32
4.6.3	ブロックプログラミング	33
4.7	ICP と IAP	34
4.8	フラッシュレジスタ	37
4.8.1	フラッシュ制御レジスタ 1 (FLASH_CR1)	37
4.8.2	フラッシュ制御レジスタ 2 (FLASH_CR2)	38
4.8.3	フラッシュプログラムメモリ保護解除キーレジスタ (FLASH_PUKR)	38
4.8.4	データ EEPROM 保護解除キーレジスタ (FLASH_DUKR)	39
4.8.5	フラッシュステータスレジスタ (FLASH_IAPSR)	39
4.8.6	フラッシュレジスタマップとリセット値	40
<b>5</b>	<b>割り込みコントローラ (ITC)</b>	<b>41</b>
5.1	ITC の概要	41
5.2	割り込みマスクと処理の流れ	41
5.2.1	ペンディング割り込みの処理	42
5.2.2	割り込みソース	43
5.3	割り込みと低電力モード	44
5.4	起動レベル／低電力モードの制御	45
5.5	同時およびネスト割り込み管理	45
5.5.1	同時割り込み管理モード	45
5.5.2	ネスト割り込み管理モード	46
5.6	外部割り込み	47
5.7	割り込み命令	48
5.8	割り込みの配置	48
5.9	ITC および EXTI レジスタ	49
5.9.1	コンディションコードレジスタ割り込みビット (CCR)	49
5.9.2	ソフトウェア優先順位レジスタ x (ITC_SPRx)	50
5.9.3	外部割り込み制御レジスタ 1 (EXTI_CR1)	51
5.9.4	外部割り込み制御レジスタ 2 (EXTI_CR2)	52
5.9.5	外部割り込み制御レジスタ 3 (EXTI_CR3)	53
5.9.6	外部割り込みステータスレジスタ 1 (EXTI_SR1)	53
5.9.7	外部割り込みステータスレジスタ 2 (EXTI_SR2)	54
5.9.8	外部割り込みポート選択レジスタ (EXTI_CONF)	54
5.9.9	ITC および EXTI レジスタマップとリセット値	55

<b>6</b>	<b>電源</b>	<b>56</b>
<b>7</b>	<b>リセット (RST) と電圧検出</b>	<b>57</b>
7.1	「リセット状態」と「リセット中」の定義	57
7.2	外部リセット (NRST ピン)	57
7.2.1	非同期外部リセットの説明	57
7.2.2	NRST/PA5 ピンの汎用出力としての設定	58
7.3	内部リセット	58
7.3.1	パワーオンリセット (POR)	58
7.3.2	独立型ウォッチドッグリセット	58
7.3.3	SWIM リセット	58
7.3.4	不正オペコードリセット	58
7.4	RST レジスタ	59
7.4.1	リセットピン設定レジスタ (RST_CR)	59
7.4.2	リセットステータスレジスタ (RST_SR)	59
7.5	RST レジスタマップとリセット値	60
<b>8</b>	<b>クロック制御 (CLK)</b>	<b>61</b>
8.1	マスタクロック (HSI クロック)	62
8.1.1	ペリフェラルクロックゲーティング (PCG)	62
8.2	LSI クロック	63
8.3	設定可能クロック出力機能 (CCO)	63
8.4	CLK レジスタ	63
8.4.1	クロック分周器レジスタ (CLK_CKDIVR)	63
8.4.2	ペリフェラルクロックゲーティングレジスタ 1 (CLK_PCKENR1)	64
8.4.3	ペリフェラルクロックゲーティングレジスタ 2 (CLK_PCKENR2)	64
8.4.4	設定可能クロック出力レジスタ (CLK_CCOR)	65
8.4.5	CLK レジスタマップとリセット値	65
<b>9</b>	<b>電源管理</b>	<b>66</b>
9.1	一般的な考慮事項	66
9.2	低消費電力のためのクロック管理	67
9.2.1	システムクロックの低速化	67
9.2.2	ペリフェラルクロックゲーティング	67

9.3	ペリフェラルのスイッチオフ	67
9.4	低電力モード	68
9.4.1	WAIT モード	69
9.4.2	HALT モード	70
9.4.3	アクティブ HALT モード	70
9.5	WFE レジスタ	71
9.5.1	WFE 制御レジスタ 1 (WFE_CR1)	71
9.5.2	WFE 制御レジスタ 2 (WFE_CR2)	72
9.6	WFE レジスタマップとリセット値	73
<b>10</b>	<b>汎用入出力ポート (GPIO)</b>	<b>74</b>
10.1	概要	74
10.2	GPIO の主な機能	74
10.3	ポートの設定と使い方	75
10.3.1	入力モード	76
10.3.2	出力モード	76
10.4	リセット設定	77
10.5	未使用の入出力ピン	77
10.6	低電力モード	77
10.7	入力モードの詳細	77
10.7.1	オルタネート機能入力	77
10.7.2	割り込み機能	78
10.8	出力モードの詳細	78
10.8.1	オルタネート機能出力	78
10.8.2	スロープ制御	78
10.9	GPIO レジスタ	79
10.9.1	ポート x 出力データレジスタ (Px_ODR)	79
10.9.2	ポート x ピン入力レジスタ (Px_IDR)	79
10.9.3	ポート x データ方向レジスタ (Px_DDR)	80
10.9.4	ポート x 制御レジスタ 1 (Px_CR1)	80
10.9.5	ポート x 制御レジスタ 2 (Px_CR2)	81
10.9.6	GPIO レジスタマップとリセット値	81

<b>11</b>	<b>自動ウェイクアップ (AWU) .....</b>	<b>82</b>
11.1	概要 .....	82
11.2	LSI クロックの測定 .....	82
11.3	AWU 機能の説明 .....	83
11.3.1	AWU 動作 .....	83
11.3.2	タイムベースの選択 .....	84
11.3.3	LSI クロック周波数の測定 .....	85
11.4	AWU レジスタ .....	86
11.4.1	制御／ステータスレジスタ (AWU_CSR) .....	86
11.4.2	非同期プリスケアラレジスタ (AWU_APR) .....	86
11.4.3	タイムベース選択レジスタ (AWU_TBR) .....	87
11.4.4	AWU レジスタマップとリセット値 .....	87
<b>12</b>	<b>ビーパ (BEEP) .....</b>	<b>88</b>
12.1	概要 .....	88
12.2	ビーパ機能の説明 .....	88
12.2.1	ビーパ動作 .....	88
12.2.2	ビーパ較正 .....	89
12.3	ビーパレジスタ .....	89
12.3.1	ビーパ制御／ステータスレジスタ (BEEP_CSR) .....	89
12.3.2	ビーパレジスタマップとリセット値 .....	89
<b>13</b>	<b>ウィンドウ型ウォッチドッグ (WWDG) .....</b>	<b>90</b>
13.1	概要 .....	90
13.2	WWDG の主な機能 .....	90
13.3	WWDG 機能の説明 .....	90
13.4	ウォッチドッグタイムアウトをプログラムするには .....	91
13.5	WWDG 低電力モード .....	92
13.6	ハードウェアウォッチドッグオプション .....	92
13.7	WWDG 割り込み .....	93
13.8	WWDG レジスタ .....	93
13.8.1	Control register (WWDG_CR) .....	93
13.8.2	Window register (WWDG_WR) .....	93

13.9	ウィンドウ型ウォッチドッグレジスタマップとリセット値	93
<b>14</b>	<b>独立型ウォッチドッグ (IWDG)</b>	<b>94</b>
14.1	概要	94
14.2	IWDG 機能の説明	94
14.3	IWDG レジスタ	96
14.3.1	キーレジスタ (IWDG_KR)	96
14.3.2	プリスケアラレジスタ (IWDG_PR)	96
14.3.3	再ロードレジスタ (IWDG_RLR)	97
14.3.4	IWDG register map and reset values	97
<b>15</b>	<b>タイマの概要</b>	<b>98</b>
15.1	タイマ機能の比較	98
15.2	タイマ信号名の用語	99
<b>16</b>	<b>16 ビット汎用タイマ (TIM2/TIM3)</b>	<b>100</b>
16.1	概要	100
16.2	TIMx の主な機能	100
16.3	TIMx タイムベースユニット	102
16.3.1	16 ビットカウンタの読み出しと書き込み	103
16.3.2	16 ビット TIMx_ARR レジスタの書き込みシーケンス	103
16.3.3	プリスケアラ	103
16.3.4	アップカウントモード	104
16.3.5	ダウンカウントモード	106
16.3.6	センターアラインモード (アップ/ダウンカウント)	108
16.4	TIMx クロック/トリガコントローラ	110
16.4.1	プリスケアラクロック (CK_PSC)	110
16.4.2	内部クロックソース	110
16.4.3	外部クロックソースモード 1	111
16.4.4	外部クロックソースモード 2	112
16.4.5	トリガの同期	113
16.4.6	他のタイマからの同期	117
16.5	TIMx キャプチャ/比較チャネル	123
16.5.1	16 ビット TIMx_CCRi レジスタの書き込みシーケンス	124
16.5.2	入力ステージ	124
16.5.3	入力キャプチャモード	125

16.5.4	出力ステージ	127
16.5.5	強制出力モード	128
16.5.6	出力比較モード	128
16.5.7	PWM モード	129
16.5.8	ブレーク機能の使用	133
16.5.9	外部イベント時の OCiREF 信号のクリア	134
16.5.10	エンコーダインタフェースモード	135
16.6	TIMx 割り込み	138
16.6.1	TIMx イベント待ち機能	138
16.7	TIMx レジスタ	139
16.7.1	制御レジスタ 1 (TIMx_CR1)	139
16.7.2	制御レジスタ 2 (TIMx_CR2)	140
16.7.3	スレーブモード制御レジスタ (TIMx_SMCR)	141
16.7.4	外部トリガレジスタ (TIMx_ETR)	142
16.7.5	割り込みイネーブルレジスタ (TIMx_IER)	143
16.7.6	ステータスレジスタ 1 (TIMx_SR1)	143
16.7.7	ステータスレジスタ 2 (TIMx_SR2)	144
16.7.8	イベント生成レジスタ (TIMx_EGR)	145
16.7.9	キャプチャ/比較モードレジスタ 1 (TIMx_CCMR1)	146
16.7.10	キャプチャ/比較モードレジスタ 2 (TIMx_CCMR2)	148
16.7.11	キャプチャ/比較イネーブルレジスタ 1 (TIMx_CCER1)	149
16.7.12	カウンタハイ (TIMx_CNTRH)	150
16.7.13	カウンタロー (TIMx_CNTRL)	150
16.7.14	プリスケアラレジスタ (TIMx_PSCr)	150
16.7.15	自動再ロードレジスタハイ (TIMx_ARRH)	151
16.7.16	自動再ロードレジスタロー (TIMx_ARRL)	151
16.7.17	キャプチャ/比較レジスタ 1 ハイ (TIMx_CCR1H)	151
16.7.18	キャプチャ/比較レジスタ 1 ロー (TIMx_CCR1L)	152
16.7.19	キャプチャ/比較レジスタ 2 ハイ (TIMx_CCR2H)	152
16.7.20	キャプチャ/比較レジスタ 2 ロー (TIMx_CCR2L)	152
16.7.21	ブレークレジスタ (TIMx_BKR)	153
16.7.22	出力アイドル状態レジスタ (TIMx_OISR)	155
16.7.23	TIMx レジスタマップとリセット値	155



<b>17</b>	<b>8 ビット基本タイマ (TIM4) .....</b>	<b>157</b>
17.1	概要 .....	157
17.2	TIM4 の主な機能 .....	157
17.3	TIM4 割り込み .....	157
17.4	TIM4 クロックの選択 .....	158
17.5	TIM4 レジスタ .....	159
17.5.1	制御レジスタ 1 (TIM4_CR1) .....	159
17.5.2	制御レジスタ 2 (TIM4_CR2) .....	160
17.5.3	スレーブモード制御レジスタ (TIM4_SMCR) .....	161
17.5.4	割り込みイネーブルレジスタ (TIM4_IER) .....	162
17.5.5	ステータスレジスタ 1 (TIM4_SR1) .....	162
17.5.6	イベント生成レジスタ (TIM4_EGR) .....	163
17.5.7	カウンタ (TIM4_CNTR) .....	163
17.5.8	プリスケアラレジスタ (TIM4_PSCR) .....	163
17.5.9	自動再ロードレジスタ (TIM4_ARR) .....	164
17.5.10	TIM4 レジスタマップとリセット値 .....	164
<b>18</b>	<b>I<sup>2</sup>C (Inter-Integrated Circuit) インタフェース .....</b>	<b>165</b>
18.1	概要 .....	165
18.2	I <sup>2</sup> C の主な機能 .....	165
18.3	I <sup>2</sup> C の概要 .....	166
18.4	I <sup>2</sup> C 機能の説明 .....	168
18.4.1	I <sup>2</sup> C スレーブモード .....	168
18.4.2	I <sup>2</sup> C マスタモード .....	170
18.4.3	エラー条件 .....	175
18.4.4	SDA/SCL ライン制御 .....	176
18.5	I <sup>2</sup> C 低電力モード .....	176
18.6	I <sup>2</sup> C 割り込み .....	177
18.7	I <sup>2</sup> C レジスタ .....	178
18.7.1	制御レジスタ 1 (I2C_CR1) .....	178
18.7.2	制御レジスタ 2 (I2C_CR2) .....	179
18.7.3	周波数レジスタ (I2C_FREQR) .....	180
18.7.4	自己アドレスレジスタ 1 LSB (I2C_OAR1L) .....	181
18.7.5	自己アドレスレジスタ 1 MSB (I2C_OAR1H) .....	181
18.7.6	自己アドレスレジスタ 2 (I2C_OAR2) .....	181
18.7.7	データレジスタ (I2C_DR) .....	182

18.7.8	ステータスレジスタ 1 (I2C_SR1)	182
18.7.9	ステータスレジスタ 2 (I2C_SR2)	184
18.7.10	ステータスレジスタ 3 (I2C_SR3)	185
18.7.11	割り込みレジスタ (I2C_ITR)	186
18.7.12	クロック制御レジスタロー (I2C_CCRL)	187
18.7.13	クロック制御レジスタハイ (I2C_CCRH)	188
18.7.14	TRISE レジスタ (I2C_TRISER)	189
18.7.15	I <sup>2</sup> C レジスタマップとリセット値	190
<b>19</b>	<b>シリアルペリフェラルインタフェース (SPI)</b>	<b>191</b>
19.1	概要	191
19.2	SPI の主な機能	191
19.3	SPI 機能の説明	192
19.3.1	概要	192
19.3.2	SPI のスレーブモード設定	196
19.3.3	SPI のマスタモード設定	196
19.3.4	SPI の単方向通信設定	197
19.3.5	データの送受信手順	197
19.3.6	ステータスフラグ	204
19.3.7	SPI の無効化	205
19.3.8	エラーフラグ	206
19.3.9	SPI 低電力モード	207
19.3.10	SPI 割り込み	207
19.4	SPI レジスタ	208
19.4.1	SPI 制御レジスタ 1 (SPI_CR1)	208
19.4.2	SPI 制御レジスタ 2 (SPI_CR2)	209
19.4.3	SPI 割り込み制御レジスタ (SPI_ICR)	210
19.4.4	SPI ステータスレジスタ (SPI_SR)	211
19.4.5	SPI データレジスタ (SPI_DR)	212
19.5	SPI レジスタマップとリセット値	212
<b>20</b>	<b>Universal synchronous/asynchronous receiver transmitter (USART)</b>	<b>213</b>
20.1	USART の概要	213
20.2	USART の主な機能	213
20.3	USART 機能の説明	214
20.3.1	USART キャラクタの説明	216

20.3.2	トランスミッタ	217
20.3.3	レシーバ	219
20.3.4	高精度ボーレートジェネレータ	223
20.3.5	クロック偏差に対する USART レシーバの許容誤差	224
20.3.6	パリティ制御	224
20.3.7	マルチプロセッサ通信	226
20.3.8	USART 同期通信	227
20.4	USART 低電力モード	230
20.5	USART 割り込み	230
20.6	USART レジスタ	231
20.6.1	ステータスレジスタ (USART_SR)	231
20.6.2	データレジスタ (USART_DR)	232
20.6.3	ボーレートレジスタ 1 (USART_BRR1)	233
20.6.4	ボーレートレジスタ 2 (USART_BRR2)	233
20.6.5	制御レジスタ 1 (USART_CR1)	234
20.6.6	制御レジスタ 2 (USART_CR2)	235
20.6.7	制御レジスタ 3 (USART_CR3)	236
20.6.8	制御レジスタ 4 (USART_CR4)	237
20.6.9	USART レジスタマップとリセット値	237
<b>21</b>	<b>ProxSense™ (PXS)</b>	<b>238</b>
21.1	概要	238
21.2	ProxSense ペリフェラルの主な機能	238
21.3	PXS ピン	239
21.4	機能の説明	239
21.4.1	概要	239
21.4.2	ProxSense 電荷移動シーケンス	240
21.5	ProxSense の操作	241
21.5.1	PXS のオン/オフ制御	241
21.5.2	PXS の初期化	241
21.5.3	PXS 取得の起動	242
21.5.4	PXS 取得の結果	243
21.5.5	PXS 取得の停止	243
21.6	特殊機能	243
21.6.1	サンプリングコンデンサの選択	243
21.6.2	EPCC の選択	243

21.6.3	RF の検出	243
21.7	低電力モード	244
21.7.1	低電力モードでの動作	244
21.7.2	消費電力の低減	244
21.8	PXS 割り込み	245
21.9	ProxSense レジスタ	246
21.9.1	ProxSense 制御レジスタ 1 (PXS_CR1)	246
21.9.2	ProxSense 制御レジスタ 2 (PXS_CR2)	247
21.9.3	ProxSense 制御レジスタ 3 (PXS_CR3)	248
21.9.4	ProxSense 割り込みおよびステータスレジスタ (PXS_ISR)	249
21.9.5	ProxSense クロック制御レジスタ 1 (PXS_CKCR1)	250
21.9.6	ProxSense クロック制御レジスタ 2 (PXS_CKCR2)	251
21.9.7	Rx[9:8] のレシーバイネーブルレジスタ (PXS_RXENRH)	251
21.9.8	Rx[7:0] のレシーバイネーブルレジスタ (PXS_RXENRL)	251
21.9.9	Rx[9:8] のレシーバ制御レジスタ 1 (PXS_RXCR1H)	252
21.9.10	Rx[7:0] のレシーバ制御レジスタ 1 (PXS_RXCR1L)	252
21.9.11	Rx[9:8] のレシーバ制御レジスタ 2 (PXS_RXCR2H)	252
21.9.12	Rx[7:0] のレシーバ制御レジスタ 2 (PXS_RXCR2L)	252
21.9.13	Rx[9:8] のレシーバ制御レジスタ 3 (PXS_RXCR3H)	252
21.9.14	Rx[7:0] のレシーバ制御レジスタ 3 (PXS_RXCR3L)	253
21.9.15	Rx[9:8] のレシーバインアクティブ状態レジスタ (PXS_RXINSRH)	253
21.9.16	Rx[7:0] のレシーバインアクティブ状態レジスタ (PXS_RXINSRL)	253
21.9.17	Tx[15:8] の送信イネーブルレジスタ (PXS_TXENRH)	254
21.9.18	Tx[7:0] の送信イネーブルレジスタ (PXS_TXENRL)	254
21.9.19	Rx[9:8] の最大カウンタ値レジスタ (PXS_MAXRH)	254
21.9.20	Rx[7:0] の最大カウンタ値レジスタ (PXS_MAXRL)	254
21.9.21	Rx[9:8] の最大カウンタイネーブルレジスタ (PXS_MAXENRH)	255
21.9.22	Rx[7:0] の最大カウンタイネーブルレジスタ (PXS_MAXENRL)	255
21.9.23	Rx[9:8] のレシーバステータスレジスタ (PXS_RXSRH)	255
21.9.24	Rx[7:0] のレシーバステータスレジスタ (PXS_RXSRL)	255
21.9.25	レシーバチャネル n のビット [15:8] のカウンタレジスタ (PXS_RXnCNTRH)	256
21.9.26	レシーバチャネル n のビット [7:0] のカウンタレジスタ (PXS_RXnCNTRL)	256
21.9.27	レシーバ 0 ~ 9 のサンプリングコンデンサ (CS) サイズ選択レジスタ 0-9 (PXS_RXnCSSELR)	256
21.9.28	レシーバ 0 ~ 9 の電極寄生補償コンデンサ (EPCC) サイズ選択レジスタ 0-9 (PXS_RXnEPCCSELR)	257

21.9.29	ProxSense レジスタマップとリセット値 .....	258
22	改版履歴 .....	260

## 表の一覧

表 1.	略語一覧	18
表 2.	割り込みレベル	22
表 3.	CPU レジスタマップ	23
表 4.	CFG_GCR レジスタマップ	24
表 5.	ブロックサイズ	34
表 6.	メモリアクセスとプログラミング法	35
表 7.	フラッシュレジスタマップ	40
表 8.	ソフトウェア優先順位レベル	41
表 9.	ISR の内部で割り込みを有効／無効にする方法	42
表 10.	ベクタアドレスマップとソフトウェア優先順位ビット	47
表 11.	外部割り込みトリガ	48
表 12.	専用割り込み命令セット	48
表 13.	ITC および EXTI レジスタマップ	55
表 14.	RST レジスタマップとリセット値	60
表 15.	ペリフェラルクロックゲーティングビット	64
表 16.	CLK レジスタマップとリセット値	65
表 17.	低電力モードの管理	68
表 18.	WFE レジスタマップ	73
表 19.	入出力ポート設定の要約	76
表 20.	低電力モードが GPIO ポートに与える影響	77
表 21.	GPIO レジスタマップ	81
表 22.	タイムベースの計算テーブル	84
表 23.	AWU レジスタマップ	87
表 24.	ビーパレレジスタマップ	89
表 25.	ウィンドウ型ウォッチドッグのタイミング例	92
表 26.	低電力モードが WWDG に与える影響	92
表 27.	WWDG レジスタマップとリセット値	93
表 28.	IWDG タイムアウト時間の最小値／最大値 (LSI クロック周波数 = 38 kHz)	95
表 29.	IWDG register map	97
表 30.	タイマの特性	98
表 31.	タイマ機能の比較	98
表 32.	内部タイマ信号の用語	99
表 33.	カウント方向とエンコーダ信号	136
表 34.	ブレーク機能を持つ OCx チャネルの出力制御ビット	154
表 35.	TIMx レジスタマップ	155
表 36.	TIM4 レジスタマップ	164
表 37.	低電力モードでの I <sup>2</sup> C インタフェースの動作	176
表 38.	I <sup>2</sup> C 割り込みリクエスト	177
表 39.	SCL 周波数に対する I2C_CCR 値 (fMASTER = 16 MHz)	189
表 40.	I <sup>2</sup> C レジスタマップ	190
表 41.	低電力モードでの SPI 動作	207
表 42.	SPI 割り込みリクエスト	207
表 43.	SPI レジスタマップとリセット値	212
表 44.	サンプリングされたデータからのノイズ検出	222
表 45.	ボーレートのプログラミングと誤差計算	223
表 46.	USART レシーバの許容誤差 (USART_DIV が 0 の場合)	224
表 47.	USART レシーバの許容誤差 (USART_DIV が 0 でない場合)	224
表 48.	フレームフォーマット	224
表 49.	低電力モードでの USART インタフェースの動作	230

表 50.	USART 割り込みリクエスト .....	230
表 51.	USART レジスタマップ .....	237
表 52.	PXS ピン .....	239
表 53.	UP および PASS フェーズ中の動作 .....	240
表 54.	低電力モードでの動作 .....	244
表 55.	ProxSense レジスタマップ .....	258
表 56.	改版履歴 .....	260

## 図の一覧

図 1.	プログラミングモデル	20
図 2.	スタッキングの順序	21
図 3.	SWIM ピンの接続	25
図 4.	STM8TL5xxx フラッシュプログラムおよびデータ EEPROM の構成	27
図 5.	タッチセンシング STM8TL5xxx デバイスの UBC 領域サイズの定義	29
図 6.	割り込み処理のフローチャート	42
図 7.	優先順位決定プロセス	43
図 8.	同時割り込み管理	46
図 9.	ネスト割り込み管理	47
図 10.	電源の概要	56
図 11.	リセット回路	57
図 12.	クロック構造	61
図 13.	GPIO ブロック図	75
図 14.	AWU のブロック図	82
図 15.	ビーブのブロック図	88
図 16.	ウォッチドッグのブロック図	90
図 17.	ウィンドウ型ウォッチドッグのタイミング図	92
図 18.	独立型ウォッチドッグのブロック図	94
図 19.	TIMx の一般的ブロック図	101
図 20.	タイムベースユニット	102
図 21.	カウンタの 16 ビット読み出しシーケンス (TIMx_CNTR)	103
図 22.	アップカウントモードのカウンタ	104
図 23.	プリスケアラ = 2 で ARPE = 0 (ARR はプリロードされない) のときのカウンタの更新	105
図 24.	ARPE = 1 (TIMx_ARR はプリロードされる) のときのカウンタ更新イベント	105
図 25.	ダウンカウントモードのカウンタ	106
図 26.	プリスケアラ = 2 で ARPE = 0 (ARR はプリロードされない) のときのカウンタの更新	107
図 27.	プリスケアラ = 1 で ARPE = 1 (ARR はプリロードされる) のときのカウンタの更新	107
図 28.	センターアラインモードのカウンタ	108
図 29.	カウンタのタイミング図、CK_PSC を 1 分周、TIMx_ARR = 06h、ARPE = 1	109
図 30.	クロック／トリガコントローラのブロック図	110
図 31.	通常モードの制御回路、fMASTER を 1 分周	111
図 32.	TI2 外部クロックの接続例	111
図 33.	外部クロックモード 1 の制御回路	112
図 34.	外部トリガ入力ブロック	112
図 35.	外部クロックモード 2 の制御回路	113
図 36.	トリガモードの制御回路	114
図 37.	トリガリセットモードの制御回路	115
図 38.	トリガゲートモードの制御回路	115
図 39.	外部クロックモード 2 + トリガモードの制御回路	116
図 40.	タイマ連結システムの実装例	117
図 41.	トリガ／マスタモード選択ブロック	117
図 42.	マスタ／スレーブタイマの例	118
図 43.	タイマ A の OC1REF によるタイマ B のゲート操作	119
図 44.	タイマ A のカウンタイネーブル信号 (CNT_EN) によるタイマ B のゲート操作	120
図 45.	タイマ A の更新イベント (TIMERA-UEV) によるタイマ B のトリガ	121
図 46.	タイマ A のカウンタイネーブル信号 (CNT_EN) によるタイマ B のトリガ	121
図 47.	タイマ A の TI1 入力によるタイマ A とタイマ B のトリガ	122
図 48.	キャプチャ／比較チャネル 1 メイン回路	123
図 49.	キャプチャモードでの TIMx_CCRi レジスタの 16 ビット読み出しシーケンス	124



図 50.	チャンネル入カステージのブロック図	124
図 51.	TIM 1 チャンネル 1 の入カステージ	125
図 52.	PWM 入力信号の測定	126
図 53.	PWM 入力信号測定の例	127
図 54.	チャンネル出カステージのブロック図	127
図 55.	チャンネル 1 の出カステージ	128
図 56.	出力比較モード、OC1 のトグル	129
図 57.	エッジアラインカウントモード PWM モード 1 の波形 (ARR=8)	130
図 58.	センターアライン PWM 波形 (ARR=8)	131
図 59.	ワンパルスモードの例	132
図 60.	ブレークに対する出力の動作	134
図 61.	ETR のアクティブ化	135
図 62.	エンコーダイインタフェースモードにおけるカウンタの動作例	137
図 63.	IC1 の極性を反転したエンコーダイインタフェースモードの例	137
図 64.	TIM4 ブロック図	157
図 65.	I <sup>2</sup> C バスプロトコル	166
図 66.	I <sup>2</sup> C ブロック図	167
図 67.	スレーブトランスミッタの転送シーケンス図	169
図 68.	スレーブレシーバの転送シーケンス図	170
図 69.	マスタトランスミッタの転送シーケンス図	172
図 70.	マスタレシーバの転送シーケンス図	173
図 71.	I2C 割り込みマッピング図	177
図 72.	SPI のブロック図	192
図 73.	シングルマスタ/シングルスレーブアプリケーション	193
図 74.	データクロックのタイミング図	195
図 75.	全二重モードでの TXE/RXNE/BSY 動作 (RXONLY = 0) 連続転送の場合	200
図 76.	スレーブ/全二重モードでの TXE/RXNE/BSY 動作 (BDM = 0、RXONLY = 0) 連続転送の場合	200
図 77.	マスタ送信専用モードでの TXE/BSY 動作 (BDM = 0 および RXONLY = 0) 連続転送の場合	201
図 78.	スレーブ送信専用モードでの TXE/BSY 動作 (BDM = 0 および RXONLY = 0) 連続転送の場合	202
図 79.	受信専用モードでの RXNE 動作 (BDM = 0 および RXONLY = 1) 連続転送の場合	203
図 80.	送信時の TXE/BSY 動作 (BDM = 0 および RXONLY = 0) 不連続転送の場合	204
図 81.	USART のブロック図	215
図 82.	ワード長のプログラミング	216
図 83.	設定可能な STOP ビット	217
図 84.	送信時の TC/TXE の動作	219
図 85.	START ビットの検出	220
図 86.	ノイズ検出のためのデータサンプリング	221
図 87.	BRR レジスタで USART_DIV をコード化する方法	223
図 88.	アイドルライン検出を使用したミュートモード	226
図 89.	アドレスマーク検出を使用したミュートモード	227
図 90.	USART の同期送信の例	228
図 91.	USART データクロックタイミング図 (M=0)	229
図 92.	USART データクロックタイミング図 (M=1)	229
図 93.	RX データのセットアップ/ホールド時間	229
図 94.	USART 割り込みマッピング図	230
図 95.	ProxSense のブロック図	240

1      メモリおよびレジスタマップ

メモリマップ、I/O ポートハードウェアレジスタマップ、および CPU/SWIM/デバッグモジュール/割り込み制御レジスタの詳細については、製品データシートを参照してください。

1.1      レジスタの説明の略語

このリファレンスマニュアルの各章のレジスタの説明では、以下の略語が使用されています。

表 1.      略語一覧

略語	説明
読み出し／書き込み (rw)	ソフトウェアは、これらのビットの読み出し／書き込みができます。
読み出し専用 (r)	ソフトウェアは、これらのビットの読み出しのみが可能です。
書き込み専用 (w)	ソフトウェアは、このビットへの書き込みのみが可能です。ビットを読み出した場合、意味のない値が返されます。
読み出し／1 回書き込み (rwo)	ソフトウェアは、このビットに 1 回だけ書き込むことができますが、読み出しはいつでも可能です。このビットをリセット値に戻すには、リセットする必要があります。
読み出し／クリア (rc_w1)	ソフトウェアは、このビットを読み出すことができ、1 を書き込むことによってクリアできます。“0”を書き込んでも、ビットの値は変化しません。
読み出し／クリア (rc_w0)	ソフトウェアは、このビットを読み出すことができ、0 を書き込むことによってクリアできます。“1”を書き込んでも、ビットの値は変化しません。
読み出し／セット (rs)	ソフトウェアは、このビットの読み出しとセットを行うことができます。“0”を書き込んでも、ビットの値は変化しません。
読み出し／読み出しによるクリア (rc_r)	ソフトウェアは、このビットを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。
予約済み (Res.)	予約済みのビットであり、リセット値のままにしておかなければなりません。



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。

## 2 中央演算処理装置 (CPU)

### 2.1 概要

この CPU は 8 ビットアーキテクチャを採用しています。6 つの内部レジスタにより効率的なデータ処理が可能です。この CPU は 80 個の基本命令を実行することができます。20 通りのアドレスモードをサポートし、6 つの内部レジスタを指定することができます。命令セットの詳しい説明については、STM8 マイクロコントローラファミリプログラミングマニュアル (PM0044) を参照してください。

### 2.2 CPU レジスタ

6 つの CPU レジスタが図 1 のプログラミングモデルに示されています。割り込みの後、レジスタは図 2 に示されている順序でスタックにプッシュされます。逆の順序でスタックからポップされます。したがって、割り込みルーチンが必要に応じて、POP および PUSH 命令を使って処理する必要があります。

#### 2.2.1 CPU レジスタの説明

##### アキュムレータ (A)

アキュムレータは、オペランドと算術演算および論理演算の結果に加えて、データ操作の結果を保持するために使用される 8 ビットの汎用レジスタです。

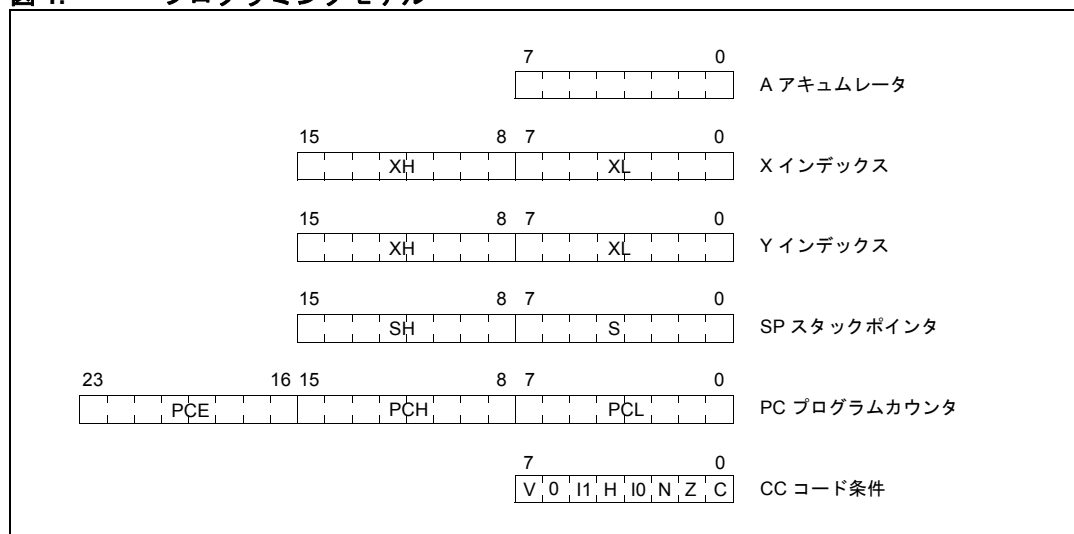
##### インデックスレジスタ (X および Y)

これらは、有効なアドレスを作成するために使用される 16 ビットのレジスタです。また、データ操作のための一時的なストレージ領域としても使用され、特定の命令 (乗算/除算) のために本来使用されるメモリとしての役割も持っています。ほとんどの場合、後続の命令が Y レジスタを参照することを示すために、クロスアセンブラは PRECODE 命令 (PRE) を生成します。

##### プログラムカウンタ (PC)

プログラムカウンタは、CPU によって実行される次の命令のアドレスを格納するために使用される 24 ビットのレジスタです。各命令が処理された後、自動的にリフレッシュされます。その結果、STM8 コアは最大 16 MB のメモリにアクセスできます。

図 1. プログラミングモデル



## スタックポインタ (SP)

スタックポインタは、16 ビットのレジスタです。スタックの次の空き位置のアドレスを含みます。製品によって異なりますが、最上位ビットを強制的にプリセット値に設定できます。

スタックは、サブルーチンコールまたは割り込み時に CPU コンテキストを保存するために使用されます。ユーザは、POP および PUSH 命令を使って直接使用することもできます。

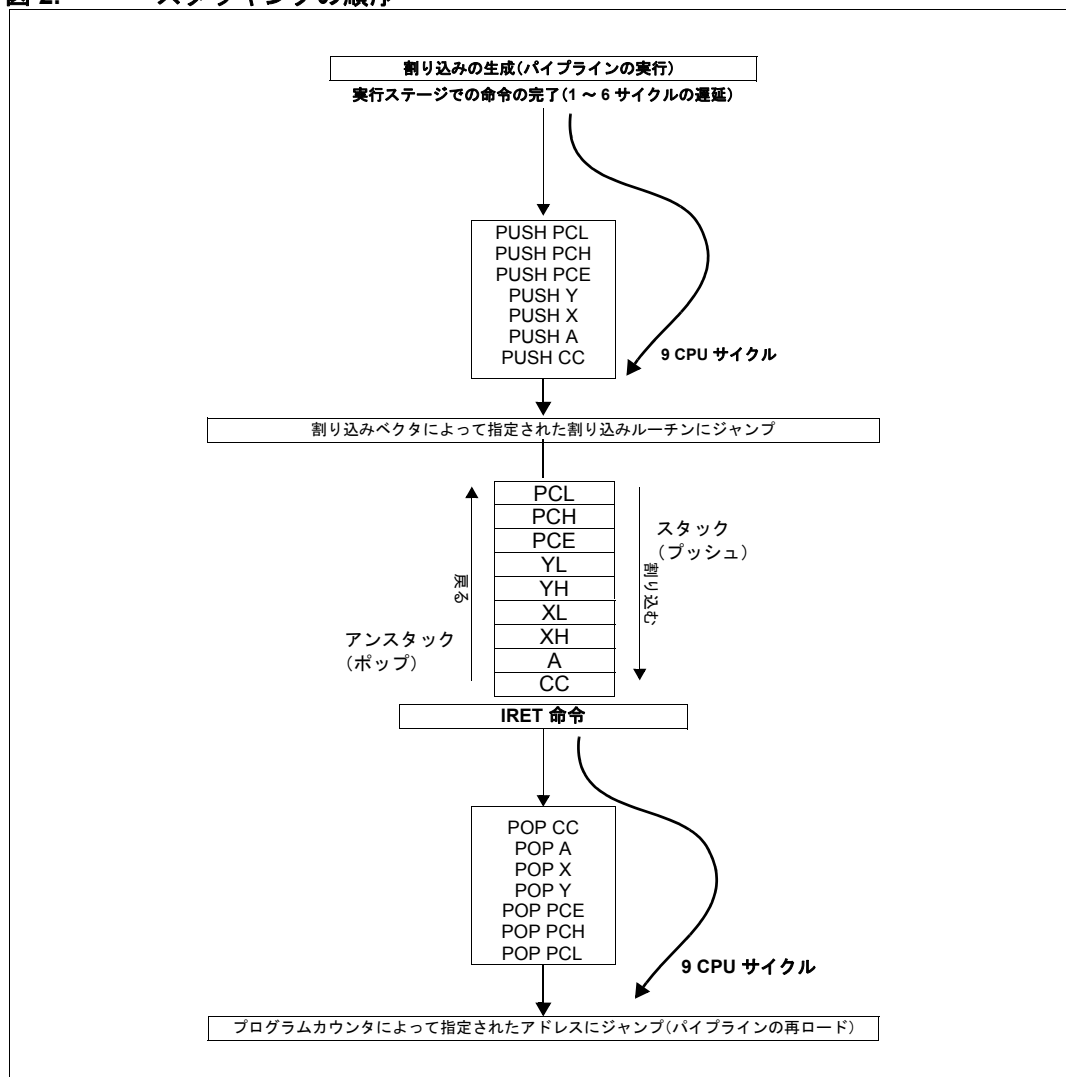
スタックポインタは、C コンパイラで提供されるスタートアップ関数によって初期化できます。C 言語で書かれたアプリケーションの場合、初期化は、C ユーザ用リンクファイルで指定されたアドレスに従って実行されます。ユーザ固有のリンクファイルまたはスタートアップファイルを使用する場合は、スタックポインタが正しく（データシートに示されているアドレスで）初期化されていることを確認してください。アセンブラで書かれたアプリケーションの場合、ST によって提供されるスタートアップ機能を使用するか、正しいアドレスでスタックポインタを初期化することによってユーザ独自のスタートアップ機能をプログラムすることができます。

スタックポインタは、データがスタックにプッシュされるとデクリメントされ、スタックからポップされるとインクリメントされます。アプリケーションでは、下限値を超えないようにする必要があります。

サブルーチンコールは、2 つまたは 3 つのスタックを占有します。割り込みは、すべての内部レジスタ（SP を除く）を保存するために、9 つのスタックを占有します。詳細については、図 2 を参照してください。

**注：** **WFI/HALT 命令は、コンテキストを事前に保存します。CPU がこれらのモードにあるときに割り込みが発生した場合、遅延は小さくなります。**

図 2. スタッキングの順序



## 条件コードレジスタ (CC)

条件コードレジスタは、実行直後の命令の結果とプロセッサの状態を示す 8 ビットのレジスタです。このレジスタの第 6 ビット (MSB) は予約済みです。これらのビットは、プログラムによって個別にテストでき、その状態の結果としてとられるアクションを指定できます。各ビットについて、以下に説明します。

- V : オーバーフロー

セットされたとき、V は、最後の符号付き算術演算中に MSB 結果ビットのオーバーフローが発生したことを示します。INC、INCW、DEC、DECW、NEG、NEGW、ADD、ADDW、ADC、SUB、SUBW、SBC、CP、および CPW 命令を参照してください。

- I1 : 割り込みマスクレベル 1

I1 フラグは、表 2 に示されているように、I0 フラグとともに現在の割り込み可能レベルを定義します。これらのフラグは、RIM、SIM、HALT、WFI、IRET、TRAP、および POP 命令を使って、ソフトウェアによってセット/クリアでき、割り込みサービスルーチンに入ったときにハードウェアによって自動的にセットされます。

表 2. 割り込みレベル

割り込み可能	優先度	I1	I0
割り込み可能メイン	最低	1	0
割り込み可能レベル 1		0	1
割り込み可能レベル 2		0	0
割り込み不可	最高	1	1

● H : ハーフキャリービット

H ビットは、ADD または ADC 命令の実行中に ALU のビット 3 とビット 4 の間で桁上げが発生したときに 1 にセットされます。H ビットは、BCD 算術演算サブルーチンで役立ちます。

● I0 : 割り込みマスクレベル 0

フラグ I1 を参照してください。

● N : ネガティブ

1 にセットされた場合、このビットは、最後の算術演算、論理演算、またはデータ操作の結果が負である（すなわち、最上位ビットが論理 1 である）ことを示します。

● Z : ゼロ

1 にセットされた場合、このビットは、最後の算術演算、論理演算、またはデータ操作の結果がゼロであることを示します。

● C : 桁上げ

セットされた場合、C は、最後の算術演算中に MSB 演算結果ビットで ALU の桁上げまたは桁下げが発生したことを示します。このビットは、ビットテスト、ブランチ、シフト、ローテート、およびロード命令中にも影響されます。ADD、ADC、SUB、および SBC 命令を参照してください。

除算演算では、C は、実行中に問題（商のオーバーフロー、ゼロでの除算）が発生したかどうかを示します。DIV 命令を参照してください。

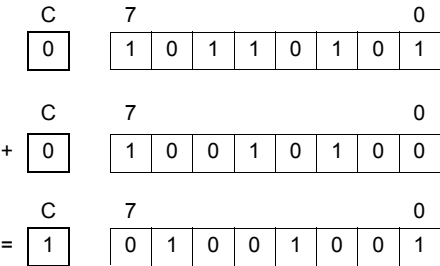
ビットテスト演算では、C は、テストされたビットのコピーです。BTJF および BTJT 命令を参照してください。

シフトおよびローテート演算では、桁上げは更新されます。RRC、RLC、SRL、SLL、および SRA 命令を参照してください。

このビットは、SCF、RCF、および CCF 命令を使用して、ソフトウェアによってセット、リセット、または補数化できます。

例：加算

$\$B5 + \$94 = "C" + \$49 = \$149$



## 2.2.2 STM8 CPU レジスタマップ

CPU レジスタは、表 3 に示されているように、STM8 アドレス空間に配置されます。これらのレジスタは、デバッグモジュールからのみアクセスでき、コアで実行されるメモリアクセス命令によってアクセスすることはできません。

表 3. CPUレジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	A	MSB	-	-	-	-	-	-	LSB
0x01	PCE	MSB	-	-	-	-	-	-	LSB
0x02	PCH	MSB	-	-	-	-	-	-	LSB
0x03	PCL	MSB	-	-	-	-	-	-	LSB
0x04	XH	MSB	-	-	-	-	-	-	LSB
0x05	XL	MSB	-	-	-	-	-	-	LSB
0x06	YH	MSB	-	-	-	-	-	-	LSB
0x07	YL	MSB	-	-	-	-	-	-	LSB
0x08	SPH	MSB	-	-	-	-	-	-	LSB
0x09	SPL	MSB	-	-	-	-	-	-	LSB
0x0A	CC	V	0	I1	H	I0	N	Z	C



2.3 グローバル設定レジスタ（CFG\_GCR）

2.3.1 起動レベル

MCU 起動レベルは、CFG\_GCR レジスタの AL ビットをプログラムすることによって設定されます。  
このビットの詳細については、[セクション 5.4：起動レベル／低電力モードの制御（45 ページ）](#)を参照してください。

2.3.2 SWIM ディセーブル

デフォルトでは、MCU リセット後、SWIM ピンは、デバッグ用またはフラッシュ／EEPROM プログラミング用の外部ツールとの通信ができるように設定されます。このピンをアプリケーションによって汎用 I/O として使用できるように設定できます。このためには、CFG\_GCR レジスタの SWD ビットをセットします。

2.3.3 グローバル設定レジスタ（CFG\_GCR）の説明

アドレスオフセット：0x00  
リセット値：0x00

7	6	5	4	3	2	1	0
予約済み						AL	SWD
						rw	rw

ビット 7:2 予約済み

ビット 1 **AL**：起動レベル

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。メインまたは割り込み専用起動レベルを設定します。  
0：メイン起動レベル。IRET 命令があると、スタックからコンテキストが取り出され、メインプログラムは WFI 命令後から続行します。  
1：割り込み専用起動レベル。IRET 命令があると、CPU はコンテキストを復元せずに、WFI/HALT モードに戻ります。

ビット 0 **SWD**：SWIM ディセーブル

0：SWIM モードは有効です。  
1：SWIM モードは無効です。

SWIM モードが有効なときには、SWIM ピンを汎用 I/O として使用することはできません。

2.3.4 グローバル設定レジスタマップとリセット値

CFG\_GCR は、STM8 アドレス空間に配置されます。ベースアドレスについては、対応するデータシートを参照してください。

表 4. CFG\_GCR レジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	CFG_GCR	-	-	-	-	-	-	AL	SWD
	リセット値	0	0	0	0	0	0	0	0





## 3 シングルワイヤインタフェースモジュール (SWIM) およびデバッグモジュール (DM)

### 3.1 概要

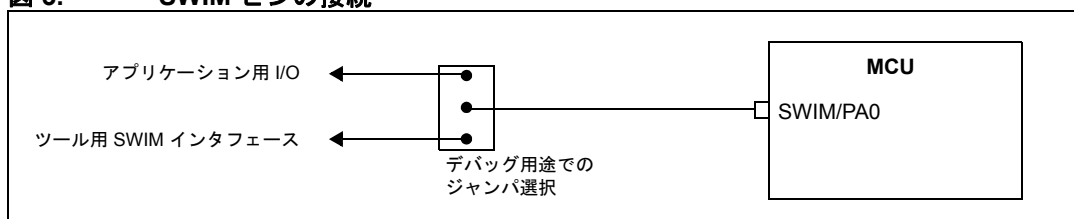
インサーキットデバッグモードまたはインサーキットプログラミングモードは、超高速のメモリプログラミング機能を備えたシングルワイヤハードウェアインタフェースを通じて管理されます。インサーキットデバッグモジュールとの組み合わせによって非侵入型エミュレーションモードも提供し、インサーキットデバッグをフル装備のエミュレータに近い性能の、非常に強力なツールにします。

### 3.2 主な特長

- 非同期の大電流シンク (8 mA)、オープンドレイン、双方向通信。
- メモリ空間のどの部分でも読み出しまたは書き込みが可能です。
- CPU レジスタ (A、X、Y、CC、SP) へのアクセスが可能です。読み出しまたは書き込みアクセスについてメモリマッピングされます。
- RAM およびペリフェラルレジスタへの非侵入型の迅速な読み出し／書き込みが可能です。
- [リセットステータスレジスタ \(RST\\_SR\)](#) のステータスフラグによるデバイスリセット機能。

SWIM ピンは、標準 I/O として使用でき、デバッグ用途で使いたい場合は、いくつかの制約があります。最も安全な方法は、PCB でストラップオプションを提供することです。

図 3. SWIM ピンの接続



### 3.3 SWIM モード

パワーオンリセット後、SWIM はリセットされ、OFF モードに入ります。

1. **OFF** : パワーオンリセット後のデフォルト状態。SWIM ピンは、アプリケーションが I/O として使用することはできません。
2. **I/O** : [グローバル設定レジスタ \(CFG\\_GCR\)](#) の SWD ビットに書き込むことによって、ソフトウェアによってこの状態になります。この状態では、アプリケーションは SWIM を標準 I/O ピンとして使用できます。リセットの場合、SWIM は OFF モードに戻ります。
3. **SWIM** : SWIM ピンに対して特定のシーケンスが実行されると、この状態になります。この状態では、SWIM ピンは、ホストツールによって STM8 を 3 つのコマンド (SRST システムリセット、ROTF 実行時読み出し、WOTF 実行時書き込み) で制御するために使用されます。

**注 :** [SWIM およびデバッグモジュール \(DM\) については、STM8 SWIM 通信プロトコルおよびデバッグモジュールユーザマニュアルを参照してください。](#) [グローバル設定レジスタ \(CFG\\_GCR\) の説明 \(24 ページ\)](#) も参照してください。

## 4 フラッシュプログラムメモリとデータ EEPROM

### 4.1 概要

内蔵のフラッシュプログラムメモリとデータ EEPROM は、共通のレジスタセットによって制御されます。これらのレジスタを使用して、アプリケーションは、メモリ内容をプログラムまたは消去したり、書き込み保護を設定することができます。アプリケーションは、デバイスのオプションバイトをプログラムすることもできます。

### 4.2 用語

- **ブロック**  
ブロックとは、1 回のプログラミング操作で一括プログラムまたは一括消去できるバイトのセットです。ブロック単位で実行される操作は、標準のプログラミングおよび消去より高速に実行されます。ブロックサイズの詳細については、[表 5](#) を参照してください。
- **ページ**  
ページとは、ブロックのセットです。  
専用オプションバイトを使用して、ユーザブートコード、データ EEPROM、および独自仕様コードのサイズを 1 ページ単位で設定できます。

### 4.3 メインフラッシュメモリの機能

- **STM8TL5xxx タッチセンシングデバイス用の EEPROM は、2 つのメモリ配列に分けられます**（メモリマッピングの詳細については、[セクション 4.4 : メモリ構成](#) を参照）。
  - 最大 16 KB の組み込みフラッシュプログラムメモリ（最大 2 KB のデータ EEPROM を含む）。データ EEPROM およびフラッシュプログラムメモリ領域は、メモリアクセスセキュリティシステム（MASS）を使用して、個別に書き込み保護することができます。
  - 64 バイトのオプションバイト（1 ブロック）：そのうち 5 バイトはすでにデバイス用に使用されています。
- **プログラミングモード**
  - バイトプログラミングと自動高速バイトプログラミング（消去操作なし）
  - ワードプログラミング
  - ブロックプログラミングおよび高速ブロックプログラミングモード（消去操作なし）
  - プログラム／消去操作の終了時と不正プログラム操作時の割り込み生成。
- **アプリケーション内プログラミング（IAP）およびインサーキットプログラミング（ICP）機能**
- **保護機能**
  - メモリ読み出し保護（ROP）
  - メモリアクセスセキュリティシステムによるプログラムメモリの書き込み保護（MASS キー）
  - メモリアクセスセキュリティシステムによるデータメモリの書き込み保護（MASS キー）
  - プログラマブルな、書き込み保護されたユーザブートコード領域（UBC）

## 4.4 メモリ構成

STM8TL5xxx タッチセンシングフラッシュプログラムメモリは、ページ単位（1 ページあたり 64 バイト）で分割すると 256 ページになり、32 ビットワード（1 ワードあたり 4 バイト）で構成されます。

メモリ配列は、4 つの領域に分けられます。

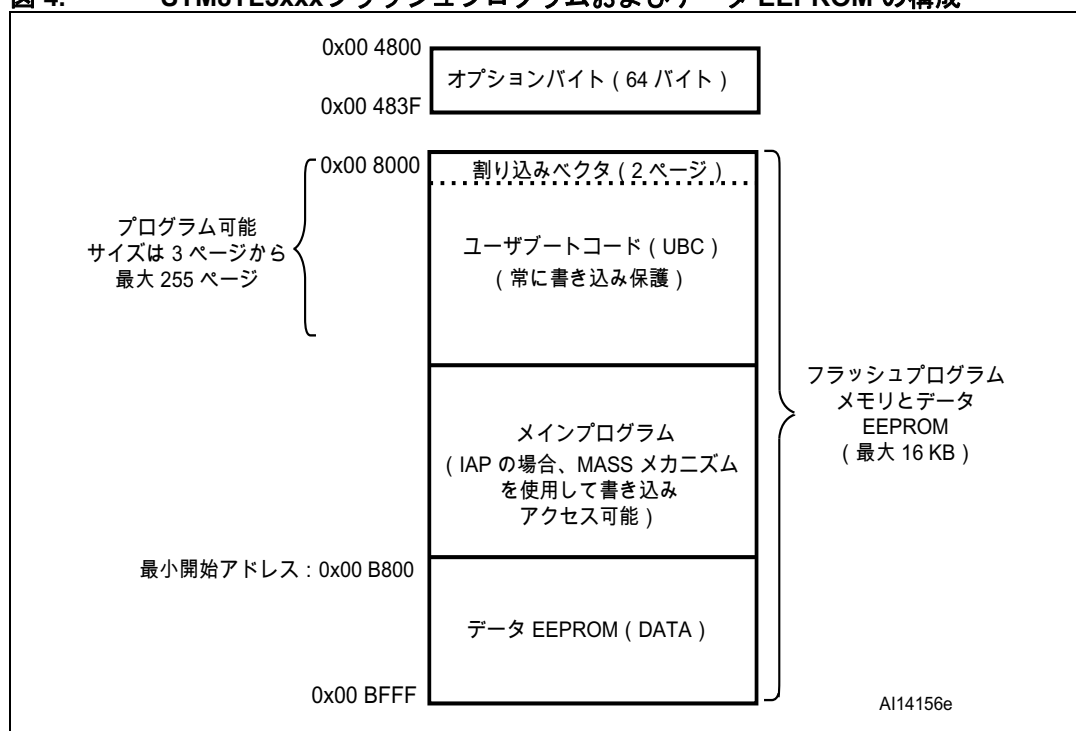
- 独自仕様コード領域（PCODE）
- ユーザブートコード領域（UBC）
- メインプログラム領域
- データ EEPROM（DATA）

フラッシュプログラムメモリ（アドレス 0x00 8000 から開始）の最初の 2 ページは、割り込みベクタを含みます。

デバイスは、別のメモリ配列に 1 ブロックのオプションバイト（64 バイト）も備えています。

メモリ構成については、[図 4](#) を参照してください。

**図 4. STM8TL5xxxフラッシュプログラムおよびデータ EEPROM の構成**



## 4.4.1 独自仕様コード領域（PCODE）

独自仕様コード領域（PCODE）は、ペリフェラルを駆動する独自仕様ソフトウェアライブラリを保護するために使用できます。

PCODE 領域のサイズは、ICP モードで（SWIM インタフェースを使用して）PCODE オプションバイト（PCODESIZE）によって設定できます。このオプションバイトは、アドレス 0x00 8000 から始まる PCODE 領域に割り当てられるページ数（64 バイト単位）を指定します。PCODF オプションバイトは一度プログラムされると、消去できなくなり、PCODE 領域のサイズは固定されます。

PCODESIZE に値 3 が設定されているとき、PCODE 領域の最小実効サイズは、アドレス 0x8080 から 0x80BF までの 1 ページです。PCODE 領域の最大サイズは、アドレス 0x00 8080 から 0x00 BFBF までの 253 ページです（PCODESIZE が 0xFF のとき）。この場合、メインプログラム領域用またはデータ EEPROM 用に使えるのは 1 ページのみです。PCODE が有効である間、TRAP 割り込みベクタは書き込み保護されます。

PCODE 領域は、TRAP 割り込みベクタを除き、自動的に読み出し保護されます（[セクション 4.5.1：読み出し保護](#)を参照）。読み出し保護は、この領域内では無効にできません。これは、PCODE の内容を読み出したり、変更できないことを意味します。

PCODE 領域は、TRAP ベクタを通してのみアクセスできます。

## 4.4.2 ユーザブート領域（UBC）

ユーザブート領域（UBC）は、リセットベクタおよび割り込みベクタ群を含みます。IAP および通信ルーチンを格納するために使用できます。UBC 領域は、IAP プログラミング中の意図しない消去や変更を防ぐために、第 2 レベルの保護対策が施されています。これは、常に書き込み保護され、MASS キーを使用して書き込み保護をロック解除できないことを意味します。

UBC 領域のサイズは、UBC オプションバイトを読み出すことによって確認できます。

UBC 領域のサイズは、ICP モードで（SWIM インタフェースを使用して）UBC オプションバイトによって設定できます。UBC オプションバイトは、アドレス 0x00 8000 から始まる UBC 領域に割り当てられるページ数を指定します。

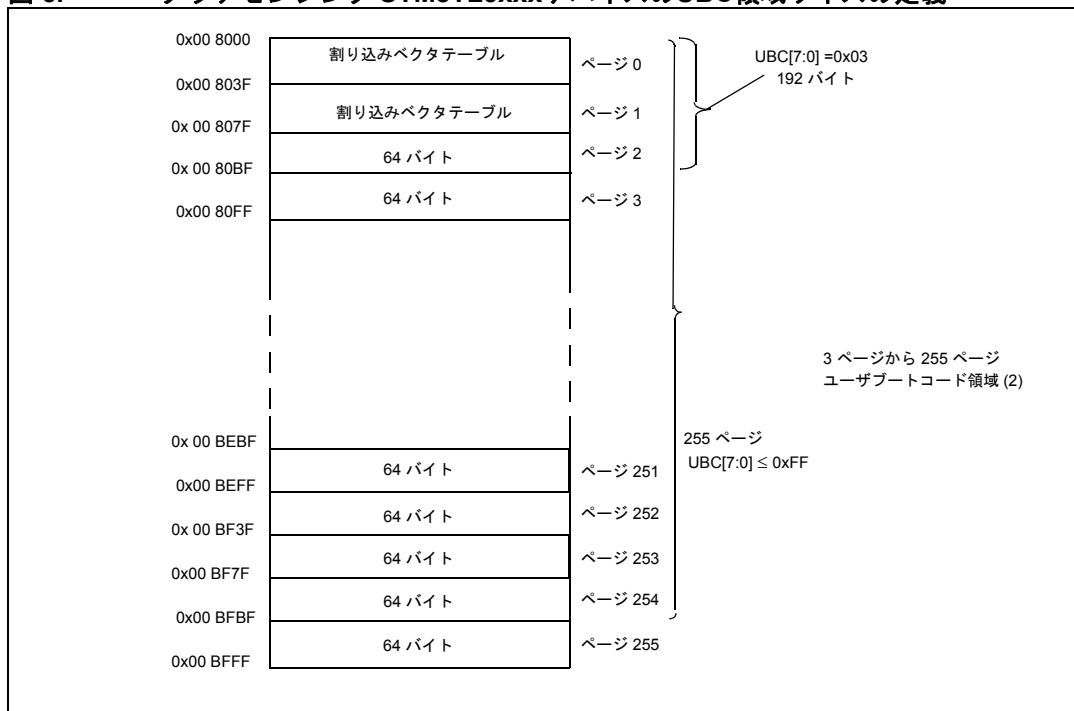
UBC 領域の最小実効サイズは 3 ページであり、そのうち 2 ページは割り込みベクタを格納するために使用されます。

**注：** *PCODE 領域が定義されているとき、UBC の最小サイズは PCODESIZE+1 となります。UBC 領域内でブートコードの格納に使用できる部分は、PCODE 領域の終端と定義済み UBC 領域の終端との間に位置する領域です。*

ブート領域の最大サイズは 255 ページであり、アドレス範囲は 0x00 8000 から 0x00 BFBF です（割り込みベクタを含む）。この場合、メインプログラム領域用またはデータ EEPROM 用に使えるのは 1 ページのみです。

UBC 領域のメモリマッピングの説明については [図 5](#) を、UBC オプションバイトの詳細については、データシートのオプションバイトのセクションを参照してください。

図 5. タッチセンシング STM8TL5xxxデバイスのUBC領域サイズの定義



1. UBC[7:0] = 0x00 は、UBC 領域にメモリ空間が割り当てられないことを意味します。
2. ページ 255 は、メインプログラム領域またはデータ EEPROM 用に予約済みです。

## 4.4.3 データ EEPROM (DATA)

データ EEPROM 領域は、アプリケーションデータを格納するために使用できます。初期状態で、データ領域は、IAP モードでメインプログラムを更新する際の意図しない変更を防ぐため、書き込み保護されています。書き込み保護は、特定の MASS キーシーケンスを使用することによってのみロック解除できます（[データ領域への書き込みアクセスの有効化](#)を参照）。

データ領域のサイズは、ICP モードで DATA オプションバイト (DATASIZE) によって設定できます。このオプションバイトは、アドレス 0x00 BFFF から下位アドレスに向かって連なるページ数 (1 ページあたり 64 バイト) を指定します。

データ領域の最大サイズは 2 KB であり、開始アドレス 0x00 B800 に対応します（[図 4](#)を参照）。つまり、最大 32 ページの 64 バイトをデータ EEPROM として定義できます。

## 4.4.4 メインプログラム領域

メインプログラム領域は、UBC または PCODE の終わりから始まり、アドレス 0x00 BFFF で終わります。アプリケーションコードを格納するために使用できます（[図 4](#)を参照）。

## 4.4.5 オプションバイト

オプションバイトは、デバイスハードウェア機能とメモリ保護を設定するために使用されます。1 ブロックの専用メモリ配列にあります。

オプションバイトは、FLASH\_CR2 レジスタの OPT ビットに 1 をセットして、ICP/SWIM モードでのみ変更できます（[セクション 4.8.2 : フラッシュ制御レジスタ 2 \(FLASH\\_CR2\)](#)を参照）。

オプションバイトの詳細については、データシートのオプションバイトのセクションを、プログラミングの詳細については、STM8 SWIM プロトコルおよびデバッグモジュールユーザマニュアル (UM0470) を参照してください。

## 4.5 メモリ保護

### 4.5.1 読み出し保護

読み出し保護は、ROP オプションバイトを 0xAA にプログラミングすることによって選択されます。読み出し保護が有効なときには、ICP モード (SWIM インタフェースを使用) でのフラッシュプログラムメモリおよびデータ領域の読み出しと変更は、書き込み保護の設定とは関係なく禁止されます。

絶対に打ち破れない保護とは言えないまでも、読み出し保護機能は、汎用マイクロコントローラに対して非常に高いレベルの保護を提供します。

読み出し保護機能は、ICP モードで ROP オプションバイトをプログラミングすることによって、プログラムメモリ、UBC、PCODE、およびデータ領域について無効にできます。この場合、フラッシュプログラムメモリ、データ領域、およびオプションバイトは自動的に消去され、デバイスの再プログラムを行うことができます。

初期状態で、PCODE 領域は、割り込みベクタ TRAP を除いて、常に読み出し保護されます ([セクション 4.4.1 : 独自仕様コード領域 \(PCODE\)](#) を参照)。

読み出し保護が有効または無効なときのメモリアクセスの詳細については、[表 6 : メモリアクセスとプログラミング方法](#) を参照してください。

### 4.5.2 メモリアクセスセキュリティシステム (MASS)

リセット後、メインプログラムおよびデータ領域 (存在する場合) は、意図しない書き込み操作に対して保護されます。内容の変更を試みる前に、ロック解除する必要があります。このロック解除メカニズムは、メモリアクセスセキュリティシステム (MASS) によって管理されます。

UBC オプションバイトで指定された UBC 領域は、常に書き込み保護されます ([セクション 4.4.2 : ユーザブート領域 \(UBC\)](#) を参照)。

メインプログラムおよびデータ領域の内容が破壊されないように、これらの内蔵不揮性メモリの内容を変更した後すぐに書き込み保護を再度有効にすることを推奨します。

#### メインプログラムメモリへの書き込みアクセスの有効化

デバイスのリセット後、MASS キーと呼ばれる 2 つの値を FLASH\_PUKR レジスタに連続して書き込むことによって、メインプログラムメモリの書き込み保護を無効にできます ([セクション 4.8.3 : フラッシュプログラムメモリ保護解除キーレジスタ \(FLASH\\_PUKR\)](#) を参照)。これらのプログラムキーは、次に 2 つのハードウェアキーと比較されます。

- 最初のハードウェアキー : 0b0101 0110 (0x56)
- 2 番目のハードウェアキー : 0b1010 1110 (0xAE)



メインプログラム領域の書き込み保護を無効にするには、以下の手順が必要です。

1. 最初の 8 ビットキーを FLASH\_PUKR レジスタに書き込みます。このレジスタがリセット後に初めて書き込まれるときには、データバスの内容はレジスタにラッチされず、最初のハードウェアキーの値 (0x56) と比較されます。
2. データバス上で使用可能なキーが正しくなかった場合、FLASH\_PUKR レジスタは、次のリセットまでロックされたままです。このアドレスに送信された新しい書き込みコマンドは破棄されます。
3. FLASH\_PUKR レジスタが 2 回目に書き込まれたとき、最初のハードウェアキーが正しい場合、データバスの内容はレジスタにラッチされず、2 番目のハードウェアキーの値 (0xAE) と比較されます。
4. データバス上で使用可能なキーが正しくなかった場合、プログラムメモリの書き込み保護は、次のリセットまでロックされたままです。このアドレスに送信された新しい書き込みコマンドは破棄されます。
5. 2 番目のハードウェアキーが正しい場合、メインプログラムメモリの書き込み保護が解除され、FLASH\_IAPSR レジスタの PUL ビットに 1 がセットされます ([セクション 4.8.5 : フラッシュステータスレジスタ \(FLASH\\_IAPSR\)](#) レジスタを参照)。

プログラミングを開始する前に、アプリケーションは、PUL ビットに 1 が確実にセットされていることを確認する必要があります。アプリケーションはいつでも、PUL ビットをクリアすることによって、フラッシュプログラムメモリへの書き込みアクセスを再び無効にできます。

## データ領域への書き込みアクセスの有効化

デバイスのリセット後、MASS キーと呼ばれる 2 つの値を連続して FLASH\_DUKR レジスタに書き込むことによって、データ領域書き込み保護を無効にできます ([セクション 4.8.6 : フラッシュレジスタマップとリセット値](#)を参照)。これらのプログラムキーは、次に 2 つのハードウェアキーと比較されます。

- 最初のハードウェアキー : 0b1010 1110 (0xAE)
- 2 番目のハードウェアキー : 0b0101 0110 (0x56)

データ領域の書き込み保護を無効にするには、以下の手順が必要です。

1. 最初の 8 ビットキーを FLASH\_DUKR レジスタに書き込みます。このレジスタがリセット後に初めて書き込まれるときには、データバスの内容はレジスタにラッチされず、最初のハードウェアキーの値 (0xAE) と比較されます。
2. データバス上に存在するキーが正しくなかった場合、アプリケーションは、2 つの MASS キーを再入力することによって、データ領域の保護解除を試みることができます。
3. 最初のハードウェアキーが正しい場合に、2 番目のキーを FLASH\_DUKR レジスタに書き込むと、データバスの内容は、やはりレジスタにラッチされず、2 番目のハードウェアキーの値 (0x56) と比較されます。
4. データバス上で使用可能なキーが正しくなかった場合、データ EEPROM 領域は、次のリセットまで書き込み保護されたままです。このアドレスに送信された新しい書き込みコマンドは無視されます。
5. 2 番目のハードウェアキーが正しい場合、データ領域の書き込み保護が解除され、FLASH\_IAPSR レジスタの DUL ビットに 1 がセットされます ([セクション 4.8.5 : フラッシュステータスレジスタ \(FLASH\\_IAPSR\)](#) を参照)。

プログラミングを開始する前に、アプリケーションは、DUL ビットに 1 が確実にセットされているかどうかを確認して、データ領域が書き込み保護されていないことを確かめる必要があります。アプリケーションはいつでも、DUL ビットをクリアすることによって、データ領域への書き込みアクセスを再び無効にできます。

## 4.6 メモリのプログラミング

メインプログラムメモリとデータ領域は、プログラム操作の実行を試みる前にロック解除する必要があります。ロック解除のメカニズムは、[セクション 4.5.2: メモリアクセスセキュリティシステム \(MASS\)](#) で説明されているように、プログラムされるメモリ領域によって異なります。

**注:** *PCODE 領域は、常に読み出し保護されています。PCODE 領域を再プログラムする唯一の方法は、ROP オプションバイトをリセットする方法ですが、フラッシュプログラムメモリ、データ領域、およびオプションバイトは消去されます。*

### 4.6.1 バイトのプログラミング

メインプログラムメモリとデータ領域は、バイトレベルでプログラムできます。1 バイトをプログラムするには、アプリケーションは、目的のアドレスに直接書き込みます。バイトプログラム操作の間、アプリケーションは停止します。

バイトを消去するには、対応するアドレスに 0x00 を書き込むだけです。

アプリケーションは、FLASH\_IAPSR レジスタを読み出すことによって、プログラミングまたは消去操作が正しく実行されたかどうかを確認することができます。

- プログラミング操作が成功すると、EOP フラグがセットされます。
- ソフトウェアが保護されたページへの書き込みを試みると、WR\_PG\_DIS がセットされます。この場合、書き込み処理は実行されません。

FLASH\_CR1 レジスタの IE ビットをセットすることによって、前もってフラッシュ割り込みが有効にされていた場合、これらのフラグの 1 つがセットされると、フラッシュ割り込みが生成されます。

#### 自動高速バイトプログラミング

プログラミング時間は、目的アドレスの初期内容によって変わることがあります。プログラムされるバイトを含むワード (4 バイト) が空でない場合、プログラム操作の前にワード全体が自動的に消去されます。逆に、ワードが空の場合、消去操作は行われず、プログラミング時間は短くなります ([データシートの表「フラッシュプログラムメモリ」](#)の t<sub>PROG</sub> を参照)。

ただし、FLASH\_CR1 レジスタの FIX ビットをセットすると、バイトの内容に関係なく、プログラム操作によってバイトの内容が強制的に消去されるので、プログラミング時間を固定できます ([セクション 4.8.1: フラッシュ制御レジスタ 1 \(FLASH\\_CR1\)](#) を参照)。結果として、プログラミング時間は固定され、消去時間と書き込み時間の合計に等しくなります ([データシートの表「フラッシュプログラムメモリ」](#)の t<sub>PROG</sub> を参照)。

**注:** *バイト書き込みを高速で (消去なしで) 行うには、書き込まれるワード全体の内容が、あらかじめ消去されていない必要があります。このため、同じワードに 2 回の高速書き込みを行う (2 番目の書き込みの前に消去を行わない) ことはできません。最初の書き込みは高速ですが、他のバイトへの 2 番目の書き込みでは消去が必要です。*

### 4.6.2 ワードプログラミング

ワード書き込み操作では、4 バイトのワード全体を 1 回でプログラムできるので、プログラミング時間をより速くできます。

バイトプログラミングと同様、ワード操作は、メインプログラムメモリとデータ EEPROM の両方について可能です。

ワードをプログラムするには、FLASH\_CR2 レジスタの WPRG ビットがあらかじめセットされていて、ワードプログラミングモードが有効になっている必要があります ([セクション 4.8.2: フラッシュ制御レジスタ 2 \(FLASH\\_CR2\)](#) を参照)。次に、プログラムされるワードの 4 バイトが、最初のアド



レスからロードされなければなりません。プログラミングサイクルは、4 バイトが書き込まれた時点で自動的に開始します。

バイト操作と同様、FLASH\_IAPSR の EOP および WR\_PG\_DIS 制御フラグとフラッシュ割り込みを使用して、操作が正常に完了したかどうかを確認できます。

## 4.6.3 ブロックプログラミング

ブロックプログラム操作は、バイトやワードのプログラム操作よりもはるかに高速です。ブロックプログラム操作では、1 つのプログラミングサイクルでブロック全体がプログラムまたは消去されます。デバイスに応じたブロックサイズの詳細については、表 5 を参照してください。

ブロック操作は、メインプログラムメモリとデータ領域の両方に対して実行可能です。ブロック操作は、完全に RAM から実行されます。

次の 3 つのブロック操作が可能です。

- ブロックプログラミング（標準ブロックプログラミングとも呼ばれます）：ブロックはプログラムされる前に自動的に消去されます。
- 高速ブロックプログラミング：事前の消去操作は行われません。
- ブロック消去

ブロックプログラミング中は、割り込みはハードウェアによってマスクされます。

### 標準ブロックプログラミング

標準ブロックプログラム操作では、1 回でブロック全体を書き込むことができます。ブロックは、プログラムの前に自動的に消去されます。

標準モードでブロック全体をプログラムするには、FLASH\_CR2 レジスタの PRG ビットがあらかじめセットされていて、標準ブロックプログラミングが有効になっている必要があります（[セクション 4.8.2：フラッシュ制御レジスタ 2 \(FLASH\\_CR2\)](#) を参照）。次に、プログラムされるデータのブロックがメインプログラムメモリまたはデータ領域のデスティネーションアドレスに順にロードされなければなりません。これによって、データのすべてのバイトがラッチされます。ブロック全体のプログラミングを開始するには、データのすべてのバイトが書き込まれなければなりません。プログラミングシーケンスで書き込まれるすべてのバイトは、同じブロック内に存在していなければなりません。これは、同じ上位アドレスを持っていないなければならないことを意味します。アドレスの最下位 6 ビットのみが変更可能です。目的ブロックの最後のバイトがロードされると、プログラミングは自動的に開始します。その前にブロック全体が自動的に消去されます。

FLASH\_IAPSR の EOP および WR\_PG\_DIS 制御フラグとフラッシュ割り込みを使用して、操作が正常に完了したかどうかを確認できます。

### 高速ブロックプログラミング

高速ブロックプログラミングでは、あらかじめメモリの内容を消去せずにプログラミングできます。したがって、高速ブロックプログラミングは標準プログラミングの 2 倍程度高速です。

このモードは、すでに消去された部分のプログラミングのみを目的としています。時間が節約できるので、空白部分を完全なアプリケーションコードでプログラムするときに役立ちます。

高速ブロックプログラミングは、標準ブロックプログラミングと同じシーケンスを使用して実行されます。高速ブロックプログラミングを有効にするには、FLASH\_CR2 レジスタの FPRG ビットが事前にセットされていなければなりません。

FLASH\_IAPSR レジスタの EOP および WR\_PG\_DIS ビットをチェックすることで、高速ブロックプログラミング操作が正常に完了したかどうかを確認できます。

**注：** 高速ブロックプログラミング操作の前にブロックが空白になっていない場合、そのブロックでプログラムされたデータは保証されません。

ブロック消去

ブロック消去操作では、ブロック全体を消去できます。

ブロック全体を消去するには、FLASH\_CR2 レジスタの ERASE ビットがあらかじめセットされていて、ブロック消去が有効になっている必要があります（[セクション 4.8.2：フラッシュ制御レジスタ 2 \(FLASH\\_CR2\)](#) を参照）。次に、ブロック内の任意のワードに '0x00 00 00 00' を書き込むことによってブロックが消去されます。ワードの開始アドレスは、'0'、'4'、'8'、または 'C' で終わらなければなりません。

FLASH\_IAPSR の EOP および WR\_PG\_DIS 制御フラグとフラッシュ割り込みを使用して、操作が正常に完了したかどうかを確認できます。

表 5.           ブロックサイズ

STM8 マイクロコントローラファミリ	ブロックサイズ
STM8TL5xxx タッチセンシング	64 バイト

4.7           ICP と IAP

インサーキットプログラミング（ICP）は、SWIM インタフェースを使用してユーザアプリケーションをマイクロコントローラにロードして、メモリの内容全体を更新するために使用されます。ICP は、迅速かつ効率的なデザインの繰り返しを提供し、不必要なパッケージの取り扱いやデバイスへの脱着をなくします。SWIM インタフェース（シングルワイヤインタフェースモジュール）は、プログラミングツールへの接続に SWIM ピンを使用します。

ICP とは対照的に、アプリケーション内プログラミング（IAP）は、マイクロコントローラによってサポートされる任意の通信インタフェース（I/O、I<sup>2</sup>C、SPI、USART など）を使用して、プログラムされるデータをメモリにダウンロードできます。IAP では、フラッシュプログラムメモリの内容を、アプリケーションの実行中に再プログラムできます。ただし、アプリケーションの一部は、ICP を使用して、フラッシュプログラムメモリに事前にプログラムされていなければなりません。

プログラミング手順の詳細については、STM8TL5xxx フラッシュプログラミングマニュアル（PM0212）および STM8 SWIM プロトコルおよびデバッグマニュアル（UM0470）を参照してください。



表 6. メモリアクセスとプログラミング方法<sup>(1)</sup>

モード	ROP (読み出し保護機能)	メモリ領域	コアからの アクセス
ユーザモード、IAP、 およびブートローダ	読み出し保護有効	TRAP を除く 割り込みベクタ	R/W <sup>(2)</sup> /E
		TRAP	R/W <sup>(3)</sup> /E
		独自仕様コード領域 (PCODE)	R/E <sup>(6)</sup>
		ユーザブートコード領域 (UBC)	R/E
		メインプログラム	R/W/E <sup>(4)</sup>
		データ EEPROM 領域 (DATA)	R/W <sup>(5)</sup>
		オプションバイト	R
	読み出し保護無効	TRAP を除く 割り込みベクタ	R/W <sup>(2)</sup> /E
		TRAP	R/W <sup>(3)</sup> /E
		独自仕様コード領域 (PCODE)	R/E <sup>(6)</sup>
		ユーザブートコード領域 (UBC)	R/E <sup>(7)</sup>
		メインプログラム	R/W/E <sup>(4)</sup>
		データ EEPROM 領域 (DATA)	R/W <sup>(5)</sup>
		オプションバイト	R/W <sup>(8)</sup>
SWIM アクティブ (ICP モード)	読み出し保護有効	TRAP を除く 割り込みベクタ	P
		TRAP	P
		独自仕様コード領域 (PCODE)	P <sup>(6)</sup>
		ユーザブートコード領域 (UBC)	P
		メインプログラム	P
		データ EEPROM 領域 (DATA)	P
		オプションバイト	P/W <sub>ROP</sub> <sup>(9)</sup>
	読み出し保護無効	TRAP を除く 割り込みベクタ	R/W <sup>(2)</sup> /E
		TRAP	R/W/E
		独自仕様コード領域 (PCODE)	R/E <sup>(6)</sup>
		ユーザブートコード領域 (UBC)	R/E <sup>(7)</sup>
		メインプログラム	R/W/E <sup>(4)</sup>
		データ EEPROM 領域 (DATA)	R/W <sup>(5)</sup>
		オプションバイト	R/W <sup>(7)</sup>

- R/W/E = Read (読み出し)、Write (書き込み)、および Execute (実行)  
R/E = 読み出しと実行 (書き込み操作は禁止)  
R = 読み出し (書き込みおよび実行操作は禁止)  
P = アクセスできない領域 (読み出し、実行、書き込み操作がすべて禁止)  
P/W<sub>ROP</sub> = 保護されていて、ROP オプションバイト以外の書き込みは禁止。
- UBC 領域が定義されていない場合、TRAP を除く 割り込みベクタは、ユーザ/IAP モードで変更できます。
- PCODE 領域が定義されている場合、TRAP ベクタは、ユーザおよび IAP モードでは変更できません。そうでない場合、TRAP は他の割り込みベクタと同じルールに従います。
- フラッシュプログラムメモリは、FLASH\_PUKR レジスタに正しい MASS キーが書き込まれるまで、書き込み保護されます (ロックされます)。FLASH\_PUKR レジスタの PUL ビットをリセットすることによって、メモリを再びロックできます。ロック解除は、2 回のリセットの間に 1 回だけ行うことができます。正しくないキーを指定した場合は、デバイスをリセットして、新しいキーをプログラムしなければなりません。

5. データメモリは、FLASH\_DUKR レジスタに正しい MASS キーが書き込まれるまで、書き込み保護されます（ロックされます）。FLASH\_IAPSR レジスタの DUL ビットをリセットすることによって、メモリを再びロックできます。正しくないキーを指定した場合は、デバイスをリセットしなくても、別のキープログラムシーケンスを実行できます。
6. PCODE 領域は、TRAP ベクタを通じて、特権モードでのみ読み出しおよび実行が可能です。PCODE は、SWIM を通じて直接アクセスすることはできません。
7. UBC 領域をプログラムするには、アプリケーションは、まず、UBC オプションバイトをクリアする必要があります。
8. オプションバイトは、FLASH\_DUKR レジスタ（OPT が 1 にセットされた）に正しい MASS キーが書き込まれるまで、書き込み保護されます（ロックされます）。FLASH\_IAPSR レジスタの DUL ビットをリセットすることによって、メモリを再びロックできます。正しくないキーを指定した場合は、デバイスをリセットしなくても、別のキープログラムシーケンスを実行できます。
9. ROP が解除されると、オプションバイトを含むメモリ全体が消去されます。

4.8 フラッシュレジスタ

4.8.1 フラッシュ制御レジスタ 1 (FLASH\_CR1)

アドレスオフセット : 0x00  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						IE	FIX
						rw	rw

ビット 7:2 予約済み

ビット 1 **IE** : フラッシュ割り込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。  
0 : 割り込みは無効です。  
1 : 割り込みは有効です。FLASH\_IAPSR レジスタの EOP または WR\_PG\_DIS 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。

ビット 0 **FIX** : 固定のバイトプログラミング時間

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 : メモリがすでに消去されている場合は標準のプログラミング時間 (1/2  $t_{prog}$ )、そうでない場合は  $t_{prog}$ 。  
1 : プログラミング時間は  $t_{prog}$  に固定されます。

## 4.8.2 フラッシュ制御レジスタ 2 (FLASH\_CR2)

アドレスオフセット : 0x01  
リセット値 : 0x00

7	6	5	4	3	2	1	0
OPT	WPRG	ERASE	FPRG	予約済み			PRG
rw	rw	rw	rw				rw

- ビット 7 **OPT** : オプションバイトの書き込み  
このビットは、ソフトウェアによってセット／クリアされます。  
0 : オプションバイトへの書き込みアクセスは無効です。  
1 : オプションバイトへの書き込みアクセスは有効です。
- ビット 6 **WPRG** : ワードプログラミング  
このビットはソフトウェアによってセットされ、操作が完了したときにハードウェアによってクリアされます。  
0 : ワードプログラム操作は無効です。  
1 : ワードプログラム操作は有効です。
- ビット 5 **ERASE**<sup>(1)</sup> : ブロック消去  
このビットはソフトウェアによってセットされ、操作が完了したときにハードウェアによってクリアされます。  
0 : ブロック消去操作は無効です。  
1 : ブロック消去操作は有効です。
- ビット 4 **FPRG**<sup>(1)</sup> : 高速ブロックプログラミング  
このビットはソフトウェアによってセットされ、操作が完了したときにハードウェアによってクリアされます。  
0 : 高速ブロックプログラム操作は無効です。  
1 : 高速ブロックプログラム操作は有効です。
- ビット 3:1 予約済み
- ビット 0 **PRG** : 標準ブロックプログラミング  
このビットはソフトウェアによってセットされ、操作が完了したときにハードウェアによってクリアされます。  
0 : 標準ブロックプログラミング操作は無効です。  
1 : 標準ブロックプログラミング操作は有効です (自動的に最初に消去)。

1. メモリがビジーのとき、ERASE および FPRG ビットはロックされます。

## 4.8.3 フラッシュプログラムメモリ保護解除キーレジスタ (FLASH\_PUKR)

アドレスオフセット : 0x02  
リセット値 : 0x00

7	6	5	4	3	2	1	0
MASS_PRG KEYS							
rw							

- ビット 7:0 **PUK [7:0]** : メインプログラムメモリロック解除キー  
このビットはソフトウェアによって書き込まれます (すべてのモード)。読み出し時、0x00 を返します。  
メインプログラム領域の書き込み保護解除メカニズムの説明については、[メインプログラムメモリへの書き込みアクセスの有効化 \(30 ページ\)](#) を参照してください。



## 4.8.4 データ EEPROM 保護解除キーレジスタ (FLASH\_DUKR)

アドレスオフセット : 0x03  
リセット値 : 0x00

7	6	5	4	3	2	1	0
MASS_DATA KEYS							
rw							

ビット 7:0 **DUK[7:0]** : データ EEPROM 書き込みロック解除キー  
このビットはソフトウェアによって書き込まれます (すべてのモード)。読み出し時、0x00 を返します。  
メインプログラム領域の書き込み保護解除メカニズムの説明については、[データ領域への書き込みアクセスの有効化 \(31 ページ\)](#) を参照してください。

## 4.8.5 フラッシュステータスレジスタ (FLASH\_IAPSR)

アドレスオフセット : 0x04  
リセット値 : 0xX0 (X は未定義)

7	6	5	4	3	2	1	0
予約済み				DUL	EOP	PUL	WR_PG_DIS
				rc_w0	rc_r	rc_w0	rc_r

ビット 7:4 予約済み

ビット 3 **DUL** : データ EEPROM 領域ロック解除フラグ  
このビットは、ハードウェアによってセットされ、0 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 : データ EEPROM 領域書き込み保護は有効です。  
1 : データ EEPROM 領域の書き込み保護は、正しい MASS キーを書き込むことによって無効にされました。

ビット 2 **EOP** : プログラミング (書き込みまたは消去操作) の終了フラグ  
このビットは、ハードウェアによってセットされます。レジスタを読み出すことによってソフトウェアによって、または新しい書き込み/消去操作が開始したときにクリアされます。  
0 : EOP イベントは発生していません。  
1 : EOP 操作が発生しました。FLASH\_CR1 レジスタの IE 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。

ビット 1 **PUL** : フラッシュプログラムメモリロック解除フラグ  
このビットは、ハードウェアによってセットされ、0 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 : メインプログラム領域の書き込み保護は有効です。  
1 : メインプログラム領域の書き込み保護は、正しい MASS キーを書き込むことによって無効にされました。

ビット 0 **WR\_PG\_DIS** : 保護ページへの書き込み試行フラグ  
このビットは、ハードウェアによってセットされ、レジスタを読み出すことによってソフトウェアによってクリアされます。  
0 : WR\_PG\_DIS イベントは発生していません。  
1 : 書き込み保護されているページへの書き込みの試みが行われました。FLASH\_CR1 レジスタの IE 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。



## 4.8.6 フラッシュレジスタマップとリセット値

フラッシュレジスタ境界アドレスの詳細については、データシートの一般ハードウェアレジスタマップを参照してください。

表 7. フラッシュレジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	FLASH_CR1	- 0	- 0	- 0	- 0	- 0	- 0	IE 0	FIX 0
0x01	FLASH_CR2	OPT 0	WPRG 0	ERASE 0	FPRG 0	- 0	- 0	- 0	PRG 0
0x02	FLASH_PUKR	PUK7 0	PUK6 0	PUK5 0	PUK4 0	PUK3 0	PUK2 0	PUK1 0	PUK0 0
0x03	FLASH_DUKR	DUK7 0	DUK6 0	DUK5 0	DUK4 0	DUK3 0	DUK2 0	DUK1 0	DUK0 0
0x04	FLASH_IAPSR	- 0	- 0	- 0	- 0	DUL 0	EOP 0	PUL 0	WR_PG_DIS 0



## 5 割り込みコントローラ (ITC)

### 5.1 ITC の概要

- ハードウェア割り込みの管理
  - － ペリフェラル割り込み機能
- ソフトウェア割り込みの管理 (TRAP)
- 柔軟な割り込み優先順位とレベル管理による、ネスト型／同時割り込み管理：
  - － ソフトウェアでプログラムできる最大 4 つのネストレベル
  - － ハードウェアによって固定された 22 の割り込みベクタ
  - － 2 つのノンマスカブルイベント：RESET、TRAP

この割り込み管理は、以下に基づきます。

- CPU 条件コードレジスタ (CCR) のビット I1 と I0
- ソフトウェア優先順位レジスタ (ITC\_SPRx)
- プログラムメモリの先頭の 0x00 8000 にあるリセットベクタ。リセット初期化ルーチンは、STMicroelectronics によって ROM にプログラミングされます。
- メモリマップの上位アドレス (0x00 8004～0x00 807C) にあり、ハードウェアの優先順位によって分類される固定の割り込みベクタアドレス

### 5.2 割り込みマスクと処理の流れ

割り込みマスキングの管理は、CCR レジスタのビット I1 と I0 に加えて、各割り込みベクタのソフトウェア優先順位レベル (表 8 を参照) を設定する ITC\_SPRx レジスタによって行われます。処理フローを 図 6 に示します。

割り込みリクエストを処理する必要がある場合、

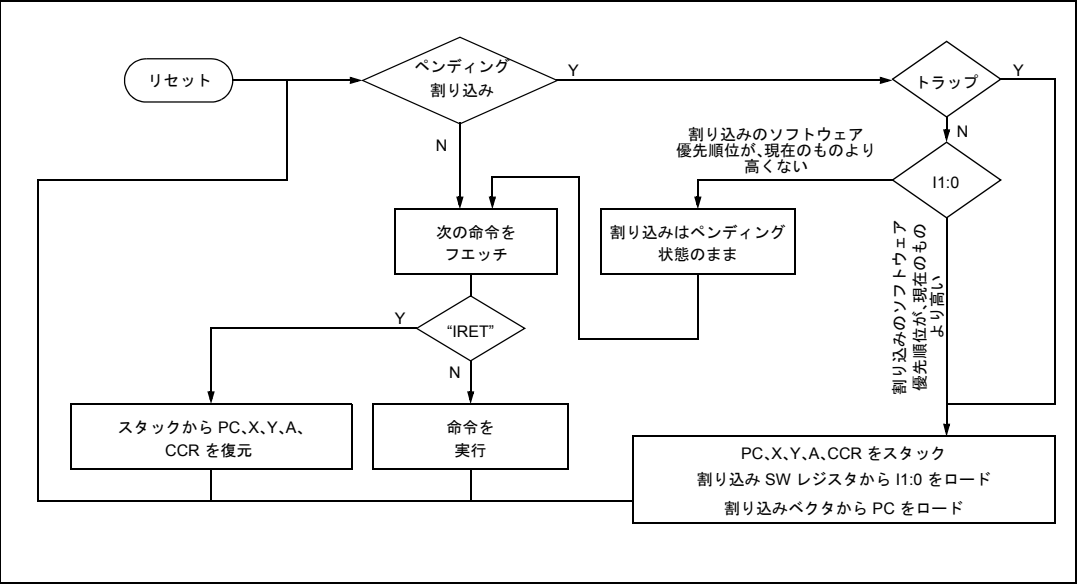
1. 通常の処理は、現在の命令実行の終わりに中断されます。
2. PC、X、Y、A、および CCR レジスタは、スタックに保存されます。
3. CCR レジスタのビット I1 と I0 は、処理された割り込みベクタに対応する ITC\_SPRx レジスタの値に基づいてセットされます。
4. 続いて、PC には処理する割り込みの割り込みベクタがロードされ、割り込みサービスルーチンの最初の命令がフェッチされます (ベクタアドレスの詳細については、データシートの割り込みマッピングテーブルを参照)。

割り込みサービスルーチンは IRET 命令で終わるようにします。これによって、保存されたレジスタの内容がスタックから復旧されます。IRET 命令の結果として、ビット I1 と I0 はスタックから復元され、プログラム実行が再開されます。

表 8. ソフトウェア優先順位レベル

ソフトウェア優先順位	レベル	I1	I0
レベル 0 (メイン)	低 ↓ 高	1	0
レベル 1		0	1
レベル 2		0	0
レベル 3 (= ソフトウェア優先順位無効)		1	1

図 6. 割り込み処理のフローチャート



注： 割り込みサービスルーチン (ISR) の内部で SIM 命令によって割り込みマスクビット I0 と I1 がセットされた場合、RIM 命令による割り込みマスクの解除によって、ソフトウェア優先順位はレベル 0 に設定されます。

ISR の内部で割り込みを有効／無効にするとき、正しい優先順位を復元するには、表 9 に示す手順に従ってください。

表 9. ISR の内部で割り込みを有効／無効にする方法

割り込みを無効にするとき	割り込みを有効にするとき
#asm PUSH CC POP ISR_CC <sup>(1)</sup> SIM #endasm	#asm PUSH ISR_CC <sup>(1)</sup> POP CC #endasm

1. IRS\_CC は、CC レジスタの現在値を格納する変数です。

5.2.1 ペンディング割り込みの処理

いくつかの割り込みは同時にペンディング状態になることがあります。ペンディングされる割り込みは、次の 2 ステップのプロセスによって決まります。

1. ソフトウェア優先順位が最も高い割り込みが処理されます。
2. いくつかの割り込みのソフトウェア優先順位が同じ場合、ハードウェア優先順位が最も高い割り込みから処理されます。

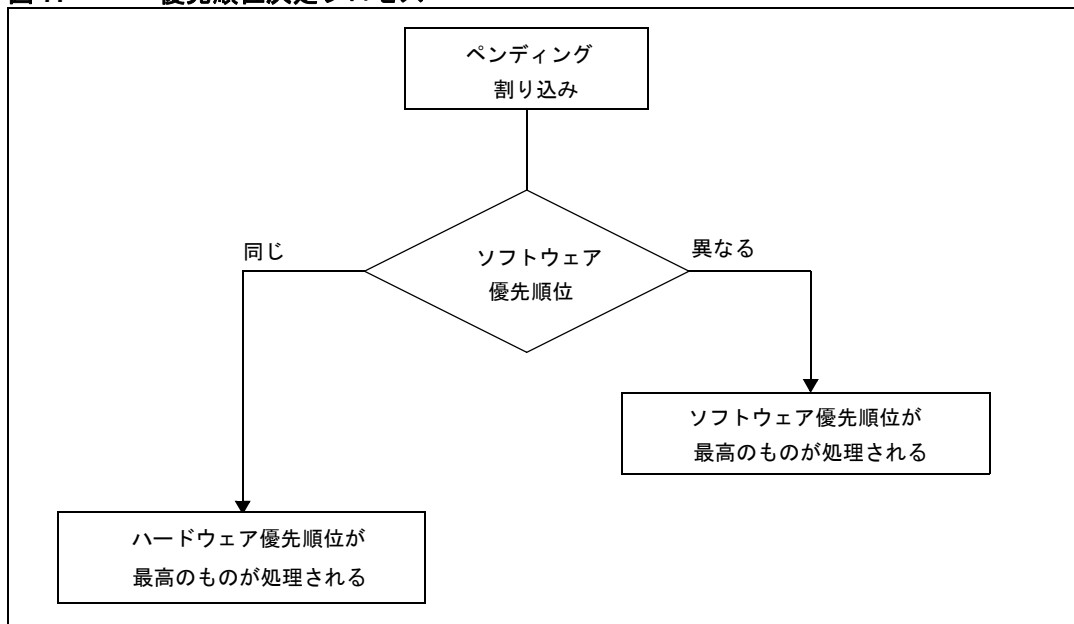
割り込みリクエストがすぐに処理されないときにはラッチされ、そのソフトウェア優先順位とハードウェア優先順位を組み合わせた値が最も高くなったときに処理されます。

- 注： 1 ハードウェア優先順位は排他的ですが、ソフトウェア優先順位はそうではありません。このため、先行するプロセスは、ただ 1 つの割り込みを処理できます。
- 2 リセットとトラップは、決定プロセスにおいて最高のソフトウェア優先順位を持つとみなされます。

ペンディング割り込み処理プロセスについては、図 7 を参照してください。



図 7. 優先順位決定プロセス



## 5.2.2 割り込みソース

STM8 割り込みコントローラは、2 種類の割り込みソースを管理します。

- ノンマスカブル割り込み：リセットとトラップ
- マスク可能割り込み：外部割り込みまたは内部ペリフェラルによって生成された割り込み

### ノンマスカブル割り込みソース

ノンマスカブル割り込みソースは、CCR レジスタのビット I1 と I0 の状態に関係なく処理されます (図 6 を参照)。PC、X、Y、A、および CCR レジスタは、トラップ割り込みが発生したときにのみスタックされます。対応するベクタが PC レジスタにロードされ、CCR レジスタのビット I1 と I0 がセットされて、割り込みを無効にします (レベル 3)。

- **トラップ (ノンマスカブルソフトウェア割り込み)**  
このソフトウェア割り込みソースは、トラップ命令が実行されるときに処理されます。この処理は、図 6 に示されるフローチャートに従って行われます。  
トラップ割り込みでは、プロセッサは HALT モードから抜けることができません。
- **リセット**  
リセット割り込みソースは、最も高い STM8 ソフトウェアおよびハードウェア優先順位です。これは、リセットルーチンの始めに、すべての割り込みが無効になることを意味します。RIM 命令によって再び有効にする必要があります (表 12：専用割り込み命令セットを参照)。  
リセット割り込みでは、プロセッサは HALT モードから抜けることができます。

リセット割り込み管理の詳細については、リセットの章を参照してください。

## マスク可能割り込みソース

マスク可能な割り込みベクタソースが処理されるのは、対応する割り込みが有効であり、ITC\_SPRx レジスタの割り込みソフトウェア優先順位が現在処理中のもの (CCR レジスタの I1 と I0) より高い場合です。2 つの条件のいずれかが満たされない場合、割り込みはラッチされ、ペンディング状態のままです。

- 外部割り込み

外部割り込みを使用して、MCU を HALT モードからウェイクアップできます。外部割り込みに対するトリガは、外部割り込み制御レジスタ (EXTI\_CRx) を介して、ソフトウェアによって選択できます。

同じ割り込みラインに接続された複数の入力ピンが同時に選択されたときには、論理和 (OR) がとられます。

レベルトリガされた外部割り込みがラッチされたとき、割り込みルーチンの終わりにそのレベルがまだ存在した場合には、その割り込みは有効なままです (ルーチンで無効にされた場合を除きます)。

- ペリフェラル割り込み

いくつかのペリフェラル割り込みによって、MCU は HALT モードからウェイクアップします。データシートの割り込みベクタテーブルを参照してください。

ペリフェラル割り込みが発生するのは、ペリフェラルステータスレジスタの特定のフラグがセットされ、ペリフェラル制御レジスタの対応するイネーブルビットがセットされたときです。

ペリフェラル割り込みをクリアするための標準的なシーケンスでは、ステータスレジスタへのアクセスを行い、関連するレジスタへの読み出しまたは書き込みを行います。このクリアシーケンスによって、内部ラッチがリセットされます。したがって、クリアシーケンスが実行されると、ペンディング割り込み (すなわち、処理を待っている割り込み) は失われます。

## 5.3 割り込みと低電力モード

すべての割り込みで、プロセッサは WAIT モードから抜けることができます。

プロセッサが低電力 WAIT モードから抜けられるのは、リセットまたはイベントに限られます。このモードに入るには、低電力 RUN モードで WFE 命令を実行します。イベントによるウェイクアップでは、システムは低電力 RUN モードに戻ります (詳細については [セクション 9: 電源管理](#) を参照)。

プロセッサが HALT/アクティブ HALT モードから抜けられるのは、外部割り込みやその他の特定の割り込みに限られます (データシートの割り込みベクタテーブルにある HALT からのウェイクアップおよびアクティブ HALT からのウェイクアップを参照)。

HALT モードからのウェイクアップ中に複数のペンディング割り込みが存在する場合、最初に処理される割り込みは、HALT モードから抜ける機能を備えた割り込みに限られます。これは、[図 7](#) に示す決定プロセスを通じて選択されます。優先順位が最も高いペンディング割り込みがデバイスを HALT モードからウェイクアップできない場合、その処理は次に回されます。

HALT 命令の実行中に内部割り込みやタイマなどからの外部割り込みが発生した場合、HALT 命令は実行されますが、HALT 命令が実行を終了した直後に、割り込みはウェイクアッププロセスを呼び出します。この場合、MCU は実際には HALT モードから RUN モードへのウェイクアップ中であり、データシートで指定された対応する遅延  $t_{WUH}$  が伴います。

## 5.4 起動レベル／低電力モードの制御

MCU 起動レベルは、CFG\_GCR レジスタの AL ビットをプログラミングすることによって設定されます（グローバル設定レジスタ (CFG\_GCR) を参照）。

このビットは、MCU の低電力モードを制御するために使用されます。きわめて低電力のアプリケーションでは、MCU は大部分の時間を WFI/HALT モードで費やし、特定のタスクを実行するために、特定の時点で割り込みによってウェイクアップされます。これらの反復的なタスクのいくつかは十分に短いため、メインプログラムに戻らずに、ISR（割り込みサービスルーチン）で直接処理されます。このようなケースに対応するため、低電力モードに入る前に WFI/HALT 命令を実行して AL ビットをセットすることができます。これによって、割り込みルーチンは低電力モードに直接戻ります。レジスタのコンテキストは最初の割り込み時にのみ保存されるので、実行時間／ISR 実行は削減されます。

結果として、きわめて簡単なアプリケーションでは、すべての操作を ISR で実行できます。より複雑なアプリケーションでは、割り込みルーチンは、AL ビットをリセットするだけでメインプログラムを再起動できます。

たとえば、ピン／センサ／押しボタンのステータスをチェックするために、自動ウェイクアップユニット (AWU) によってアプリケーションを 50 ms ごとにウェイクアップすることが必要な場合もあります。ほとんどの場合、これらのピンは無効になっているので、MCU は、メインプログラムを実行しなくても低電力モードに戻ることができます。1 本でも有効なピンがある場合、ISR は、AL ビットをリセットすることによってメインプログラムを起動します。

## 5.5 同時およびネスト割り込み管理

STM8 デバイスは、2 つの割り込み管理モードを備えています。

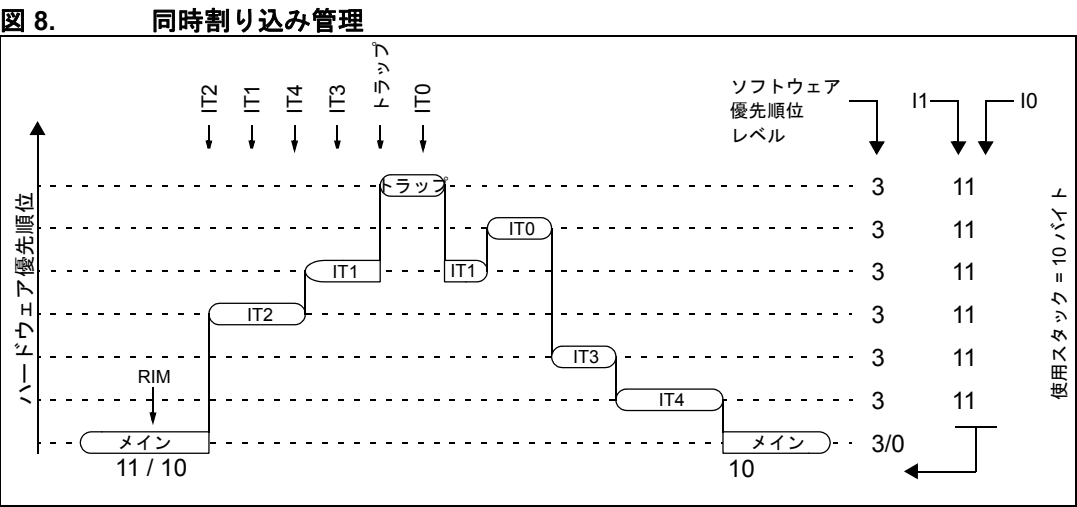
- 同時モード
- ネストモード

### 5.5.1 同時割り込み管理モード

このモードでは、すべての割り込みの割り込み優先順位レベルが 3 なので、リセットやトラップによる場合を除いて、割り込みを受けることはありません。

ハードウェア優先順位は、最低から最高の優先順位まで、メイン、IT4、IT3、IT2、IT1、IT0、トラップ（同じ優先順位）、およびリセットの順です。

図 8 に同時割り込み管理モードの例を示します。



5.5.2 ネスト割り込み管理モード

このモードでは、割り込みルーチンの処理中の割り込みが可能です。このモードは、レベル 3 を下回る割り込み優先順位レベルが設定されると有効になります。

ハードウェア優先順位は、最低から最高の優先順位まで、メイン、IT4、IT3、IT2、IT1、IT0、およびトラップの順です。

ソフトウェア優先順位は、ITC\_SPRx レジスタの対応する I1\_x ビットと I0\_x ビットをセットすることによって、割り込みベクタごとに設定されます。I1\_x および I0\_x ビットは、CCR レジスタの I1 および I0 ビットと同じ意味を持ちます (表 10 を参照)。

レベル 0 はプログラミングできません (I1\_x=1、I0\_x=0)。この場合、以前に格納された値が保持されます。たとえば、以前の値が 0xCF であり、プログラミングされた値が 64h に等しい場合、結果は 44h です。

リセットおよびトラップベクタには、ソフトウェア優先順位はありません。これらのベクタが処理されると、CCR レジスタのビット I1 と I0 の両方がセットされます。

注：割り込み x の実行中に I1\_x および I0\_x ビットが変更された場合、デバイスは次のように動作します。割り込み x がまだペンディング中であり (新しい割り込みまたはフラグがクリアされていない)、新しいソフトウェア優先順位が前のものよりも高い場合、再び割り込み x に入ります。そうでない場合、割り込み x の IRET 後の次の割り込みリクエストまで、ソフトウェア優先順位は変化しません。

割り込みルーチンの実行時には、HALT、POPCC、RIM、SIM、および WFI 命令は、次の IRET 命令または前述のいずれかの命令が発行されるまで、現在のソフトウェア優先順位を変更します。専用の割り込み命令のリストについては、セクション 5.7 を参照してください。

図 9 にネスト割り込み管理モードの例を示します。

警告：スタックオーバーフローは、ソフトウェアに障害を通知することなく発生することがあります。

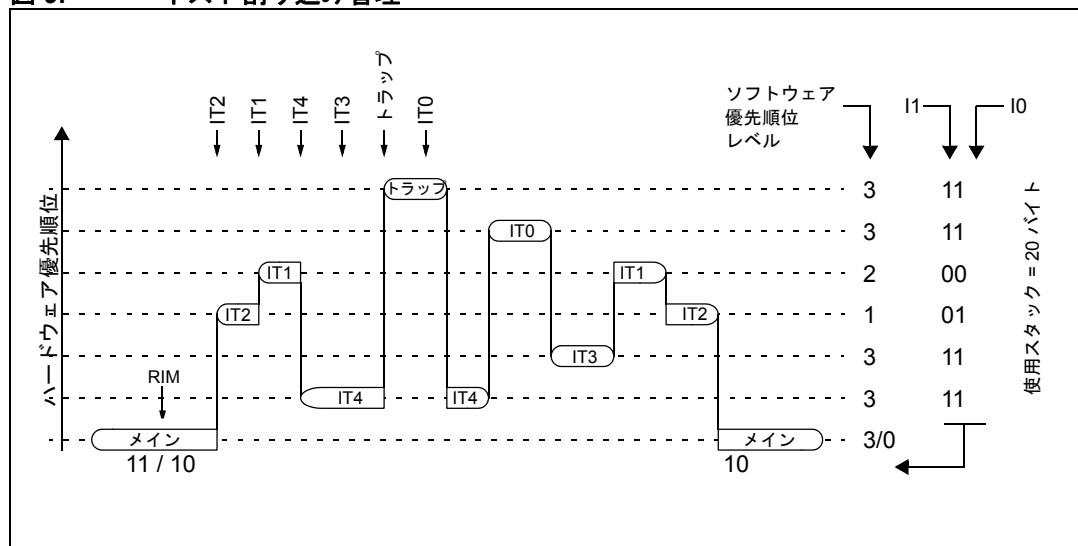




表 10. ベクタアドレスマップとソフトウェア優先順位ビット

ベクタアドレス	ITC_SPRx ビット
0x00 8008h	I1_0 および I0_0 ビット
0x00 800Ch	I1_1 および I0_1 ビット
...	...
0x00 807Ch	I1_29 および I0_29 ビット

図 9. ネスト割り込み管理



## 5.6 外部割り込み

10 個の割り込みベクタが外部割り込みイベント専用割り当てられています。

- EXTIB - ポート B 上の 8 ライン : PB[7:0]
- EXTID - ポート D 上の 8 ライン : PD[7:0]
- EXTI0 - ポート A/B/C/D 上の 4 ライン、ビット 0 : Px[0]
- EXTI1 - ポート A/B/D 上の 3 ライン
- EXTI2 - ポート A/B/D 上の 3 ライン
- EXTI3 - ポート A/B/D 上の 3 ライン
- EXTI4 - ポート A/B/D 上の 3 ライン
- EXTI5 - ポート B/D 上の 2 ライン
- EXTI6 - ポート A/B/D 上の 3 ライン
- EXTI7 - ポート A/B/D 上の 3 ライン

割り込みを生成するには、割り込みを有効にした状態で、対応する GPIO ポートを入力モードに設定する必要があります。詳細については、GPIO の章のレジスタ説明を参照してください。

外部割り込みが発生すると、EXTI\_SRx ステータスレジスタの対応するビットがセットされます。これはペンディング割り込みを示します。このビットに 1 を書き込んでクリアすると、対応する外部ペンディング割り込みがクリアされます。



割り込みトリガは、外部割り込み制御レジスタ 1 (EXTI\_CR1)、外部割り込み制御レジスタ 2 (EXTI\_CR2)、および外部割り込み制御レジスタ 3 (EXTI\_CR3) で設定する必要があります ([セクション 5.9.5](#)、[セクション 5.9.6](#)、および [セクション 5.9.7](#) を参照)。

表 11. 外部割り込みトリガ

GPIO ポート	割り込みトリガ	設定レジスタ
ポート A、B、C、および D 上の EXTI0 ~EXTI3	立ち下がりエッジとローレベル 立ち上がりエッジのみ 立ち下がりエッジのみ 立ち上がりおよび立ち下がりエッジ	EXTI_CR1
ポート A、B、C、および D 上の EXTI4 ~EXTI7		EXTI_CR2
EXTIB および EXTID		EXTI_CR3

5.7 割り込み命令

[表 12](#) に割り込み命令を示します。

表 12. 専用割り込み命令セット

命令	新しい説明	機能／例	I1	H	I0	N	Z	C
HALT	HALT モードへの移行		1		0			
IRET	割り込みルーチン復帰	CCR、A、X、Y、PC をポップ	I1	H	I0	N	Z	C
JRM	I1:0=11 (レベル 3) の場合はジャンプ	I1:0=11 ?						
JRNM	I1:0<>11 の場合はジャンプ	I1:0<>11 ?						
POP CC	スタックから CCR をポップ	メモリ => CCR	I1	H	I0	N	Z	C
PUSH CC	CC をスタックにプッシュ	CC => メモリ						
RIM	割り込みを有効にする (レベル 0 をセット)	CCR の I1:0 に 10 をロード	1		0			
SIM	割り込みを無効にする (レベル 3 をセット)	CCR の I1:0 に 11 をロード	1		1			
TRAP	ソフトウェアトラップ	ソフトウェア NMI	1		1			
WFI	割り込みを待機		1		0			
WFE	イベントを待機		1		0			

5.8 割り込みの配置

割り込みベクタアドレスのテーブルについては、対応するデバイスのデータシートを参照してください。



5.9 ITC およびEXTI レジスタ

5.9.1 CPU コンディションコードレジスタ割り込みビット (CCR)

アドレス：データシートの「一般ハードウェアレジスタマップ」の表を参照してください。  
リセット値：0x28

7	6	5	4	3	2	1	0
V	–	I1	H	I0	N	Z	C
r	r	rw	r	rw	r	r	r

ビット 5、3<sup>(1)</sup> **I[1:0]**：ソフトウェア割り込み優先順位ビット<sup>(2)</sup>

この 2 つのビットは、現在の割り込みリクエストのソフトウェア優先順位を示します。割り込みリクエストが発生すると、対応するベクタのソフトウェア優先順位がソフトウェア優先順位レジスタ (ITC\_SPRx) から自動的にロードされます。  
I[1:0] ビットは、RIM、SIM、HALT、WFI、IRET、または PUSH/POP 命令を使用して、ソフトウェアによってセット/クリアすることもできます (図 9：ネスト割り込み管理を参照)。

I1	I0	優先順位	レベル
1	0	レベル 0 (メイン)	低 ↓ 高
0	1	レベル 1	
0	0	レベル 2	
1	1	レベル 3 (= ソフトウェア優先順位無効*)	

- 1. 他の CCR ビットの詳細については、「中央演算処理」のセクションを参照してください。
- 2. TRAP および RESET イベントは、レベル 3 のプログラムで割り込むことができます。

5.9.2      ソフトウェア優先順位レジスタ x (ITC\_SPRx)

アドレスオフセット : 0x00~0x07  
リセット値 : 0xFF

	7	6	5	4	3	2	1	0
ITC_SPR1	予約済み		VECT2SPR[1:0]		VECT1SPR[1:0]		予約済み	
ITC_SPR2	VECT7SPR[1:0]		VECT6SPR[1:0]		予約済み		VECT4SPR[1:0]	
ITC_SPR3	VECT11SPR[1:0]		VECT10SPR[1:0]		VECT9SPR[1:0]		VECT8SPR[1:0]	
ITC_SPR4	VECT15SPR[1:0]		VECT14SPR[1:0]		VECT13SPR[1:0]		VECT12SPR[1:0]	
ITC_SPR5	VECT19SPR[1:0]		予約済み		予約済み		予約済み	
ITC_SPR6	予約済み		VECT22SPR[1:0]		VECT21SPR[1:0]		VECT20SPR[1:0]	
ITC_SPR7	VECT27SPR[1:0]		VECT26SPR[1:0]		VECT25SPR[1:0]		予約済み	
ITC_SPR8	予約済み				VECT29SPR[1:0]		VECT28SPR[1:0]	
	rw				rw	rw	rw	rw

ビット 7:0 **VECTxSPR[1:0]** : ベクタ x ソフトウェア優先順位ビット

これら 8 つの読み出し／書き込みレジスタ (ITC\_SPR1~ITC\_SPR8) は、各割り込みベクタのソフトウェア優先順位を定義するために、ソフトウェアによって書き込まれます。

ベクタのリストを [表 10 : ベクタアドレスマップとソフトウェア優先順位ビット](#) に示します。

優先順位ごとにプログラミングする値については、[セクション 5.9.1 : CPU コンディションコードレジスタ割り込みビット \(CCR\)](#) を参照してください。

予約済み

ITC\_SPR8 のビット 7:4 は、ハードウェアによって 1 に固定されています。

**注：**    **10 (優先順位レベル 0) を書き込むことは禁止されています。10 が書き込まれた場合、以前の値が保持され、割り込み優先順位は変更されません。**

## 5.9.3 外部割り込み制御レジスタ 1 (EXTI\_CR1)

アドレスオフセット : 0x00  
リセット値 : 0x00

7	6	5	4	3	2	1	0
P3IS[1:0]		P2IS[1:0]		P1IS[1:0]		P0IS[1:0]	
rw		rw		rw		rw	

- ビット 7:6 **P3IS[1:0]** : ポート x ビット 3 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1 (レベル 3) にセットされた場合にのみ書き込みできます。ポート A、B、D 外部割り込みのビット 3 のトリガを定義します。
- 00 : 立ち下がりエッジとローレベル
  - 01 : 立ち上がりエッジのみ
  - 10 : 立ち下がりエッジのみ
  - 11 : 立ち上がりおよび立ち下がりエッジ
- ビット 5:4 **P2IS[1:0]** : ポート x ビット 2 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1 (レベル 3) にセットされた場合にのみ書き込みできます。ポート A、B、D 外部割り込みのビット 2 のトリガを定義します。
- 00 : 立ち下がりエッジとローレベル
  - 01 : 立ち上がりエッジのみ
  - 10 : 立ち下がりエッジのみ
  - 11 : 立ち上がりおよび立ち下がりエッジ
- ビット 3:2 **P1IS[1:0]** : ポート x ビット 1 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1 (レベル 3) にセットされた場合にのみ書き込みできます。ポート A、B、D 外部割り込みのビット 1 のトリガを定義します。
- 00 : 立ち下がりエッジとローレベル
  - 01 : 立ち上がりエッジのみ
  - 10 : 立ち下がりエッジのみ
  - 11 : 立ち上がりおよび立ち下がりエッジ
- ビット 1:0 **P0IS[1:0]** : ポート x ビット 0 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1 (レベル 3) にセットされた場合にのみ書き込みできます。ポート A、B、C、D 外部割り込みのビット 0 のトリガを定義します。
- 00 : 立ち下がりエッジとローレベル
  - 01 : 立ち上がりエッジのみ
  - 10 : 立ち下がりエッジのみ
  - 11 : 立ち上がりおよび立ち下がりエッジ

5.9.4 外部割り込み制御レジスタ 2（EXTI\_CR2）

アドレスオフセット：0x01  
リセット値：0x00

7	6	5	4	3	2	1	0
P7IS[1:0]		P6IS[1:0]		P5IS[1:0]		P4IS[1:0]	
rw		rw		rw		rw	

- ビット 7:6 **P7IS[1:0]**：ポート x ビット 7 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1（レベル 3）にセットされた場合にのみ書き込みできます。ポート A、B、D 外部割り込みのビット 7 のトリガを定義します。
- 00：立ち下がリエッジとローレベル  
01：立ち上がりエッジのみ  
10：立ち下がリエッジのみ  
11：立ち上がりおよび立ち下がリエッジ
- ビット 5:4 **P6IS[1:0]**：ポート x ビット 6 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1（レベル 3）にセットされた場合にのみ書き込みできます。ポート A、B、D 外部割り込みのビット 6 のトリガを定義します。
- 00：立ち下がリエッジとローレベル  
01：立ち上がりエッジのみ  
10：立ち下がリエッジのみ  
11：立ち上がりおよび立ち下がリエッジ
- ビット 3:2 **P5IS[1:0]**：ポート x ビット 5 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1（レベル 3）にセットされた場合にのみ書き込みできます。ポート B、D 外部割り込みのビット 5 のトリガを定義します。
- 00：立ち下がリエッジとローレベル  
01：立ち上がりエッジのみ  
10：立ち下がリエッジのみ  
11：立ち上がりおよび立ち下がリエッジ
- ビット 1:0 **P4IS[1:0]**：ポート x ビット 4 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1（レベル 3）にセットされた場合にのみ書き込みできます。ポート A、B、D 外部割り込みのビット 4 のトリガを定義します。
- 00：立ち下がリエッジとローレベル  
01：立ち上がりエッジのみ  
10：立ち下がリエッジのみ  
11：立ち上がりおよび立ち下がリエッジ

## 5.9.5 外部割り込み制御レジスタ 3 (EXTI\_CR3)

アドレスオフセット : 0x02  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み				PDIS[1:0]		PBIS[1:0]	
				rw		rw	

- ビット 7:4 予約済み
- ビット 3:2 **PDIS[1:0]** : ポート D 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1 (レベル 3) にセットされた場合にのみ書き込みできます。ポート D[3:0] またはポート D[7:4] (あるいはその両方) の EXTID が有効なときのポート D 外部割り込みのトリガを定義します。
- 00 : 立ち下がりエッジとローレベル  
01 : 立ち上がりエッジのみ  
10 : 立ち下がりエッジのみ  
11 : 立ち上がりおよび立ち下がりエッジ
- ビット 1:0 **PBIS[1:0]** : ポート B 外部割り込みトリガビット
- これらのビットは、CCR レジスタの I1 と I0 の両方が 1 (レベル 3) にセットされた場合にのみ書き込みできます。ポート B[3:0] またはポート B[7:4] (あるいはその両方) の EXTIB が有効なときのポート B 外部割り込みのトリガを定義します。
- 00 : 立ち下がりエッジとローレベル  
01 : 立ち上がりエッジのみ  
10 : 立ち下がりエッジのみ  
11 : 立ち上がりおよび立ち下がりエッジ

## 5.9.6 外部割り込みステータスレジスタ 1 (EXTI\_SR1)

アドレスオフセット : 0x03  
リセット値 : 0x00

7	6	5	4	3	2	1	0
P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

- ビット 7:0 **PxF** : ポート A/B/C/D ビット x 外部割り込みフラグ
- これらのビットは、対応するピンで割り込みイベントが発生したとき、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"1"を書き込むことによってクリアされます。
- 0 : 割り込みは発生していません。  
1 : 外部割り込みがペンディング中です。

## 5.9.7 外部割り込みステータスレジスタ 2 (EXTI\_SR2)

アドレスオフセット : 0x04  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						PDF	PBF
						rc_w1	rc_w1

ビット 7:2 予約済み

ビット 1 **PDF** : ポート D 外部割り込みフラグ

このビットは、対応するピンで割り込みイベントが発生したとき、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"1"を書き込むことによってクリアされます。

0 : 割り込みは発生していません。  
1 : 外部割り込みがペンディング中です。

ビット 0 **PBF** : ポート B 外部割り込みフラグ

このビットは、対応するピンで割り込みイベントが発生したとき、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"1"を書き込むことによってクリアされます。

0 : 割り込みは発生していません。  
1 : 外部割り込みがペンディング中です。

## 5.9.8 外部割り込みポート選択レジスタ (EXTI\_CONF)

アドレスオフセット : 0x05  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み				PDHIS	PDLIS	PBHIS	PBLIS
				rw	rw	rw	rw

ビット 7:4 予約済み

ビット 3 **PDHIS** : ポート D[7:4] 外部割り込み選択

EXTID 割り込み用のピン PD[7:4] を選択します。

0 : PD[7:4] が EXTI7-EXTI4 割り込み生成に使用されます。  
1 : PD[7:4] が EXTID 割り込み生成に使用されます。

ビット 2 **PDLIS** : ポート D[3:0] 外部割り込み選択

EXTID 割り込み用のピン PD[3:0] を選択します。

0 : PD[3:0] が EXTI3-EXTI0 割り込み生成に使用されます。  
1 : PD[3:0] が EXTID 割り込み生成に使用されます。

ビット 1 **PBHIS** : ポート B[7:4] 外部割り込み選択

EXTIB 割り込み用のピン PB[7:4] を選択します。

0 : PB[7:4] が EXTI7-EXTI4 割り込み生成に使用されます。  
1 : PB[7:4] が EXTIB 割り込み生成に使用されます。

ビット 0 **PBLIS** : ポート B[3:0] 外部割り込み選択

EXTIB 割り込み用のピン PB[3:0] を選択します。

0 : PB[3:0] が EXTI3-EXTI0 割り込み生成に使用されます。  
1 : PB[3:0] が EXTIB 割り込み生成に使用されます。



## 5.9.9 ITC および EXTI レジスタマップとリセット値

表 13. ITC および EXTI レジスタマップ

アドレス オフセット	レジスタ 名称	7	6	5	4	3	2	1	0
ITC-SPR ブロック <sup>(1)</sup>									
0x00	ITC_SPR1 リセット値	- 1	- 1	VECT2SPR1 1	VECT2SPR0 1	VECT1SPR1 1	VECT1SPR0 1	- 1	- 1
0x01	ITC_SPR2 リセット値	VECT7SPR1 1	VECT7SPR0 1	VECT6SPR1 1	VECT6SPR0 1	- 1	- 1	VECT4SPR1 1	VECT4SPR0 1
0x02	ITC_SPR3 リセット値	VECT11SPR1 1	VECT11SPR0 1	VECT10SPR1 1	VECT10SPR0 1	VECT9SPR1 1	VECT9SPR0 1	VECT8SPR1 1	VECT8SPR0 1
0x03	ITC_SPR4 リセット値	VECT15SPR1 1	VECT15SPR0 1	VECT14SPR1 1	VECT14SPR0 1	VECT13SPR1 1	VECT13SPR0 1	VECT12SPR1 1	VECT12SPR0 1
0x04	ITC_SPR5 リセット値	VECT19SPR1 1	VECT19SPR0 1	- 1	- 1	- 1	- 1	- 1	- 1
0x05	ITC_SPR6 リセット値	- 1	- 1	VECT22SPR1 1	VECT22SPR0 1	VECT21SPR1 1	VECT21SPR0 1	VECT20SPR1 1	VECT20SPR0 1
0x06	ITC_SPR7 リセット値	VECT27SPR1 1	VECT27SPR0 1	VECT26SPR1 1	VECT26SPR0 1	VECT25SPR1 1	VECT25SPR0 1	- 1	- 1
0x07	ITC_SPR8 リセット値	- 1	- 1	- 1	- 1	VECT29SPR1 1	VECT29SPR0 1	VECT28SPR1 1	VECT28SPR0 1
ITC-EXTI ブロック <sup>(2)</sup>									
0x00	EXTI_CR1	P3IS1 0	P3IS0 0	P2IS1 0	P2IS0 0	P1IS1 0	P1IS0 0	P0IS1 0	P0IS0 0
0x01	EXTI_CR2	P7IS1 0	P7IS0 0	P6IS1 0	P6IS0 0	P5IS1 0	P5IS0 0	P4IS1 0	P4IS0 0
0x02	EXTI_CR3	- 0	- 0	- 0	- 0	PDIS1 0	PDIS0 0	PBIS1 0	PBIS0 0
0x03	EXTI_SR1	P7F 0	P6F 0	P5F 0	P4F 0	P3F 0	P2F 0	P1F 0	P0F 0
0x04	EXTI_SR2	- 0	- 0	- 0	- 0	- 0	- 0	PDF 0	PBF 0
0x05	EXTI_CONF 0	- 0	- 0	- 0	- 0	PDHIS 0	PDLIS 0	PBHIS 0	PBLIS 0

1. アドレスオフセットは、ITC-SPR ブロックのベースアドレスに対して示されています（データシートの CPU/SWIM/デバッグモジュール / 割り込みコントローラレジスタの表を参照）。
2. アドレスオフセットは、ITC-EXTI ブロックのベースアドレスに対して示されています（データシートの「一般ハードウェアレジスタマップ」の表を参照）。

## 6 電源

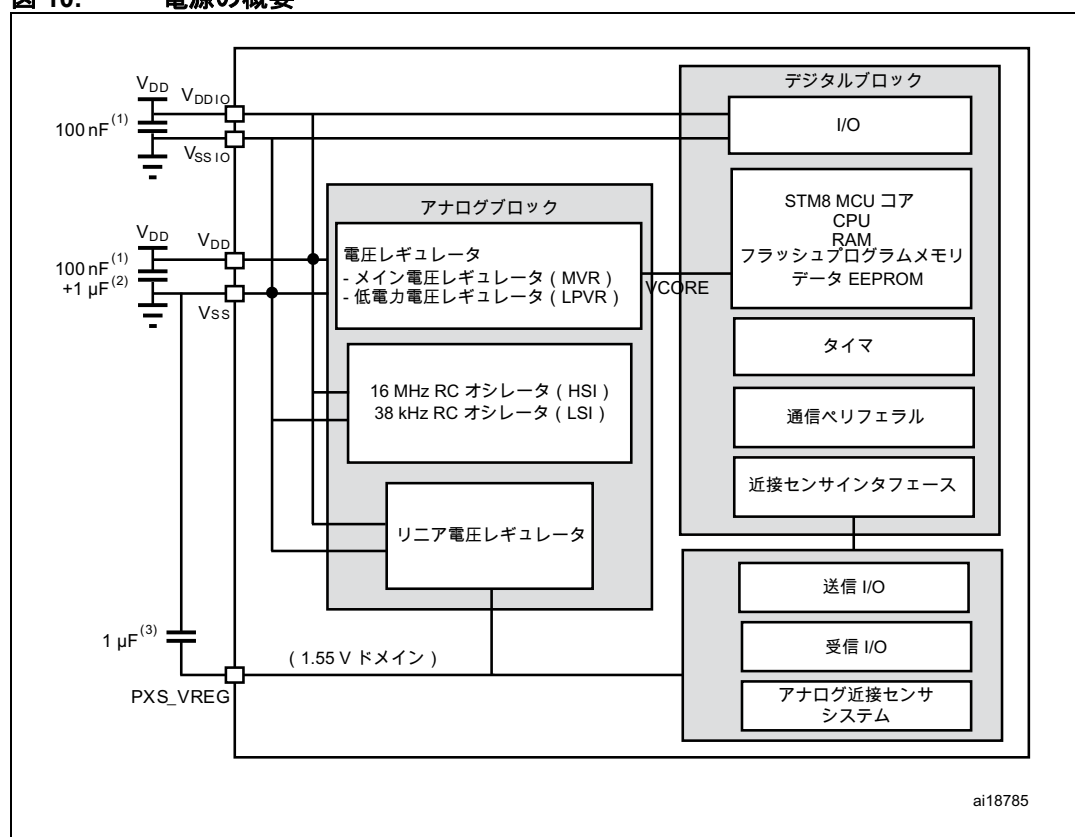
本 MCU は、入出力ポートと内部電圧レギュレータの両方に電源を供給するための 1 つの電源  $V_{DD}/V_{SS}$  (1.65~3.6 V) を備えています。電圧レギュレータには、メイン電圧レギュレータモード (MVR) と低電力電圧レギュレータモード (LPVR) という、2 つのモードがあります。

HALT またはアクティブ HALT モード ([セクション 9.4.2: HALT モード](#) と [セクション 9.4.3: アクティブ HALT モード](#) で説明) に入ると、消費電流を減らすために、システムは自動的に MVR から LPVR に切り替わります。

パワーオンリセット (POR) は、MCU の電源オン時に安全なリセットを保証します。

動作範囲の詳細については、データシートの電気特性のセクションを参照してください。

図 10. 電源の概要



1. 上記のように、各電源ペアは、フィルタリングセラミックコンデンサを使用して安定化する必要があります。デバイスを正しく機能させるため、これらのコンデンサは、該当するピンのできるだけ近くに (または下方に) 配置する必要があります。
2. この 1 μF コンデンサは、 $V_{DD}$  ピンに接続する必要があります。
3. PXS\_VREG に接続されるこの 1 μF セラミックコンデンサは、低 ESR ( $ESR \leq 1 \Omega$ ) である必要があります。

**注:** [さまざまな電力モードについては、セクション 9: 電源管理 \(66 ページ\) を参照してください。](#)

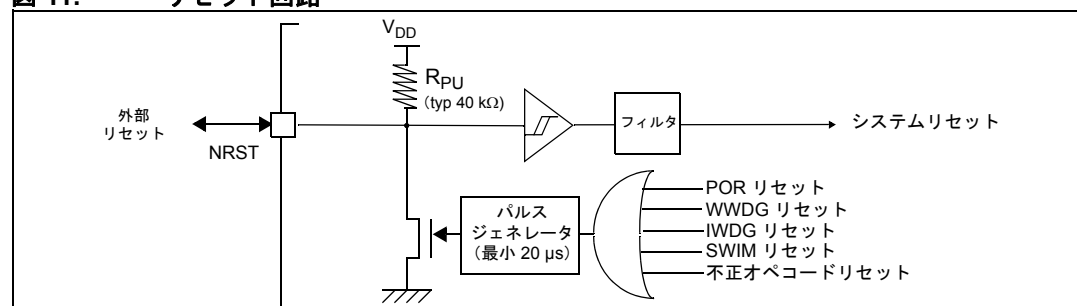
## 7 リセット (RST) と電圧検出

6 つのリセットリソースがあります。

- NRST ピンを通じた外部リセット (このピンは、汎用出力ピンとして設定することもできます)。
- パワーオンリセット (POR)
- 独立型ウォッチドッグリセット (IWDG)
- ウィンドウ型ウォッチドッグリセット (WWDG)
- SWIM リセット
- 不正オペコードリセット

これらのソースは、NRST ピンに作用します。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 0x8000 に固定されています。

図 11. リセット回路



### 7.1 「リセット状態」と「リセット中」の定義

リセットが発生すると、外部ピンのプルダウンから内部リセット信号のリリースまで、リセットフェーズが存在します。このフェーズ中に、マイクロコントローラは、リセットベクタに行く前に、いくつかのハードウェア設定を行います。

このフェーズの終わりに、ほとんどのレジスタが「リセット状態」の値に設定されます。リセットフェーズ中、すなわち「リセット中」では、いくつかのピンの設定が「リセット状態」の設定とは異なることがあります。

### 7.2 外部リセット (NRST ピン)

#### 7.2.1 非同期外部リセットの説明

NRST ピンは、入力ピンと、内蔵の  $R_{PU}$  弱プルアップ抵抗を備えたオープンドレイン出力ピンの両方として使用できます。

外部リセットを生成するには、NRST ピンに最短で 300 ns のローパルスが必要です。リセットの検出は非同期なので、MCU は HALT モードでもリセット状態に移行できます。

NRST ピンは、外部デバイスをリセットするためのオープンドレイン出力としても働きます。

内部の過度状態時間によって、内部リセットソースごとに少なくとも 20  $\mu$ s のパルスが維持されます。追加の内部弱プルアップは、リセットが強制されなかったとき、リセットピンをハイレベル状態に保ちます。

詳細については、[図 11](#) とデータシートの電気パラメータのセクションを参照してください。

## 7.2.2 NRST/PA5 ピンの汎用出力としての設定

使用可能なピン数を最適化するために、NRST ピン（外部リセットピン）を汎用のプッシュプル出力（PA5）として設定することができます。

セキュリティ上、この設定は、[リセットピン設定レジスタ \(RST\\_CR\)](#) に指定されたキー（D0h）を書き込むことによって、リセット後に 1 回だけ行うことができます。

PA5 ピンが汎用出力ピンとして設定されているときには、パワーオンリセット（POR）またはその他の内部リセットソースによってのみ、MCU をリセットできます。

## 7.3 内部リセット

内部リセットソースの場合、NRST ピンはパルスジェネレータによって生成された遅延フェーズの間、ロー状態に保たれます。

各内部リセットソースは、[リセットステータスレジスタ \(RST\\_SR\)](#) の特定のフラグビットにリンクされます。これらのフラグは、与えられたリセットソースに応じて、それぞれリセット時にセットされます。したがって、最後のリセットソースを特定するために使用されます。論理値“1”を書き込むことによって、ソフトウェアによってクリアされます。

**注：** *POR フラグを含むすべてのフラグは POR 時にリセットされます。*

### 7.3.1 パワーオンリセット（POR）

パワーオン時、POR は、電源電圧（ $V_{DD}$ ）が指定の電圧に達するまで、デバイスをリセット状態に保ちます。その後、 $V_{DD}$  が最低動作電圧に達したことを保証するために、指定の時間にわたってさらに長くリセットをアクティブに保持してから、NRST ピンを解放します。詳細については、データシートの電気パラメータのセクションを参照してください。

### 7.3.2 独立型ウォッチドッグリセット

詳細については、独立型ウォッチドッグのセクションを参照してください。

リセットは、アプリケーションソフトウェアによって、独立型ウォッチドッグを使用してトリガできます。

### 7.3.3 SWIM リセット

SWIM インタフェースに接続された外部デバイスは、SWIM ブロックに MCU リセットの生成を要求できます。

### 7.3.4 不正オペコードリセット

予期しない動作に対するデバイスの安定性を強化するため、不正オペコード検出のシステムが実装されています。実行されるコードがどのオペコード値やプリバイト値にも対応していない場合、リセットが生成されます。これをウォッチドッグと組み合わせることによって、予期しない故障や干渉からの復帰が可能になります。

## 7.4 RST レジスタ

### 7.4.1 リセットピン設定レジスタ（RST\_CR）

アドレスオフセット：0x00  
リセット値：0x00

7	6	5	4	3	2	1	0
RSTPIN_KEY							
rwo							

ビット 7:0 **RSTPIN\_KEY[7:0]**：リセットピン設定キー

0x00：NRST/PA5 はリセットピンとして設定されます（デフォルトのリセット値）。

0xD0：NRST/PA5 は汎用出力として設定されます。

これらのビットは 1 回のみ書き込み可能です。いつでも読み出すことができます。

**注：** *D0h 以外の値を書き込むと、00h を書き込んだものとして扱われます。*

### 7.4.2 リセットステータスレジスタ（RST\_SR）

アドレスオフセット：0x01  
パワーオンリセット後のリセット値：0x01

7	6	5	4	3	2	1	0
予約済み			WWDGF	SWIMF	ILLOPF	IWDGF	PORF
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 7:5 予約済み

ビット 4 **WWDGF**：ウィンドウ型ウォッチドッグリセットフラグ

0：WWDG リセットは発生していません。

1：WWDG リセットが発生しました。

このビットは、ハードウェアによってセットされ、ソフトウェアによって“1”を書き込むことでクリアされます。

ビット 3 **SWIMF**：SWIM リセットフラグ

0：SWIM リセットは発生していません。

1：SWIM リセットが発生しました。

このビットは、ハードウェアによってセットされ、ソフトウェアによって“1”を書き込むことでクリアされます。

ビット 2 **ILLOPF**：不正オペコードリセットフラグ

0：ILLOP リセットは発生していません。

1：ILLOP リセットが発生しました。

このビットは、ハードウェアによってセットされ、ソフトウェアによって“1”を書き込むことでクリアされます。

ビット 1 **IWDGF**：独立型ウォッチドッグリセットフラグ

0：IWDG リセットは発生していません。

1：IWDG リセットが発生しました。

このビットは、ハードウェアによってセットされ、ソフトウェアによって“1”を書き込むことでクリアされます。

ビット 0 **PORF**：パワーオンリセット（POR）フラグ

0：POR は発生していません。

1：POR が発生しました。

このビットは、ハードウェアによってセットされ、ソフトウェアによって“1”を書き込むことでクリアされます。

**注：** *RST\_SR リセット値は、以前のリセットソースによって異なります。*

7.5 RST レジスタマップとリセット値

表 14. RST レジスタマップとリセット値

アドレス オフセット <sup>(1)</sup>	レジスタ名	7	6	5	4	3	2	1	0
0x00	RST_CR リセット値	RSTPIN _KEY7 0	RSTPIN _KEY6 0	RSTPIN _KEY5 0	RSTPIN _KEY4 0	RSTPIN _KEY3 0	RSTPIN _KEY2 0	RSTPIN _KEY1 0	RSTPIN _KEY0 0
0x01	RST_SR リセット値	- 0	- 0	- 0	WWDGF 0	SWIMF 0	ILLOPF 0	IWDGF 0	PORF 1

1. レジスタアドレスの詳細については、データシートの「一般ハードウェアレジスタマップ」の表を参照してください。



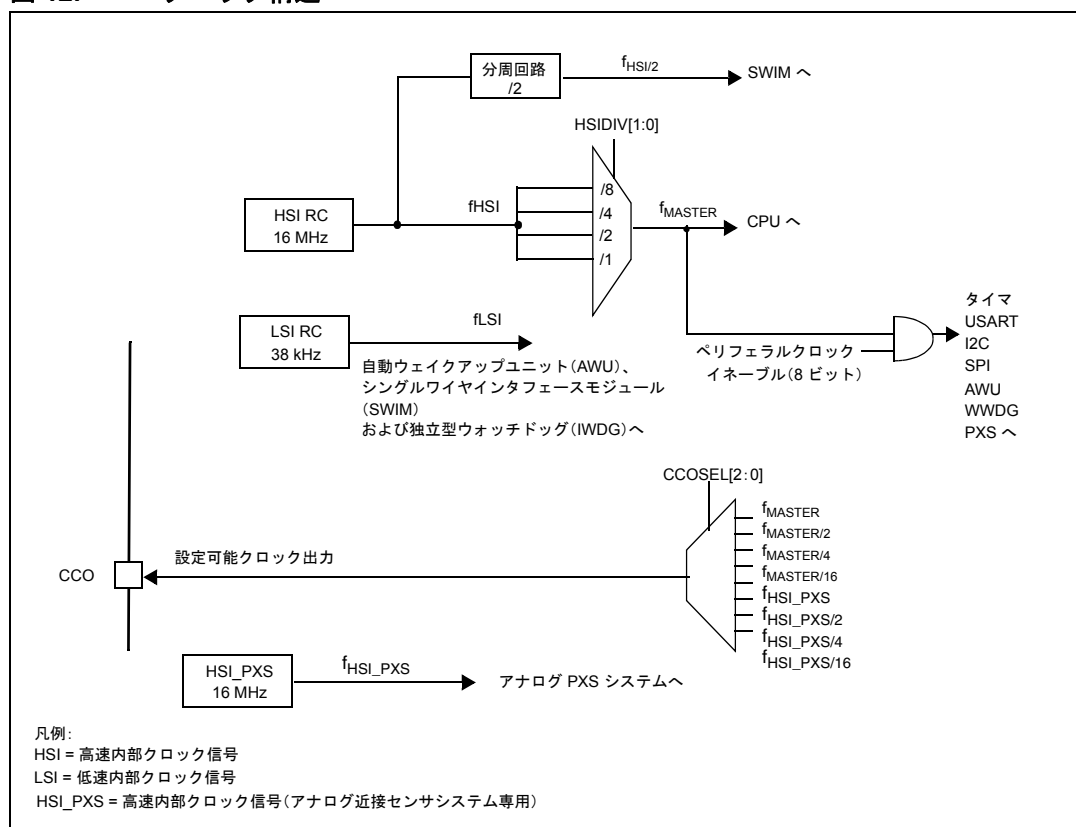
## 8 クロック制御 (CLK)

クロックコントローラは、非常に耐ノイズ性に優れていると同時に使いやすく設計されています。その目的は、アプリケーションから最高の性能を引き出すと同時に、すべてのマイクロコントローラの節電機能のメリットをフルに活用することにあります。

CPU とさまざまなペリフェラルへのクロック分配を管理して、電力消費を最適化できます。

安全でグリッチのないスイッチ機構により、クロックプリスケータを使用して、マスタクロックの分周比を動作中に切り替えることができます。

図 12. クロック構造





## 8.1 マスタクロック (HSI クロック)

システムのマスタクロックは、16 MHz の高速内部 RC オシレータ (HSI RC) と、その後続くプログラマブルプリスケアラです。

内部 16 MHz RC オシレータから生成された  $f_{\text{HSI}}$  クロック信号を、プログラマブル分周回路 (係数: 1~8) によって分周して、 $f_{\text{MASTER}}$  を得ることができます。これは [クロック分周器レジスタ \(CLK\\_CKDIVR\)](#) でプログラムされます。

**注:** **起動時、マスタクロックソースは、8 分周された HSI RC クロック出力 ( $f_{\text{HSI}}/8$ ) として自動的に選択されず。**

HSI RC オシレータには、起動時間が短い上に低コスト (外付部品不要) で 50% のデューティサイクルを持つ、16 MHz のマスタクロックソースを供給できるという利点があります。

クロックコントローラは、マスタクロックソースを、8 分周された HSI RC クロック出力 ( $f_{\text{HSI}}/8$ ) として設定します。したがって、 $V_{\text{DD}}$  条件が悪い場合に安全な起動を保証するのに役立ちます。

### 8.1.1 ペリフェラルクロックゲーティング (PCG)

使用しないペリフェラルへのクロックをゲーティングすると、消費電力の低減に役立ちます。ペリフェラルクロックゲーティング (PCG) モードは、RUN モードまたは低速モード中にいつでも、以下のペリフェラルへの  $f_{\text{MASTER}}$  クロック接続を選択的に有効または無効にします。

- TIM2
- TIM3
- TIM4
- I2C
- SPI
- USART
- AWU/BEEP (ただし、[制御/ステータスレジスタ \(AWU\\_CSR\)](#) の AWUEN ビットと [ビーパ制御/ステータスレジスタ \(BEEP\\_CSR\)](#) の BEEPEN ビットによって制御される LSI クロックを除く)
- PXS (近接センサ)
- WWDG

ウィンドウ型ウォッチドッグ (WWDG) ペリフェラルクロックは、対応するオプションビットがセットされている場合にハードウェアによって開始できるため、デフォルトで有効にされる唯一のクロックです。

デバイスリセット後、すべてのペリフェラルクロックは無効です (ただし、WWDG クロックを除く)。[ペリフェラルクロックゲーティングレジスタ 1 \(CLK\\_PCKENR1\)](#) または [ペリフェラルクロックゲーティングレジスタ 2 \(CLK\\_PCKENR2\)](#) の対応する PCKEN ビットをセットすることによって、任意のペリフェラルへのクロックを有効にできます。ペリフェラルを有効にするには、まず、CLK\_PCKENR レジスタの対応する PCKEN ビットを有効にし、次に、ペリフェラル制御レジスタのペリフェラルイネーブルビットをセットします。

ペリフェラルを適切に無効にするには、まず、ペリフェラル制御レジスタの適切なビットを無効にし、次に、対応するクロックを停止します。

AWU カウンタには  $f_{\text{MASTER}}$  とは異なる低速の内部固有クロック ( $f_{\text{LSI}}$ ) が供給されるので、たとえこのペリフェラルレジスタへのクロックゲーティングがアサートされた場合でも実行を続けます。同じことが、ビーパにも当てはまります。

## 8.2 LSI クロック

LSI RC オシレータは、 $f_{LSI}$  クロックを生成します。このクロックは、38 kHz の低速内部 RC オシレータ (LSI RC) であり、独立型ウォッチドッグ (IWDG)、自動ウェイクアップユニット (AWU)、およびビーパ (BEEP) によって使用されます。LSI は、アクティブ HALT モードで有効であり、RUN モードで保持することができます。

## 8.3 設定可能クロック出力機能 (CCO)

設定可能クロック出力機能 (CCO) は、外部 CLK\_CCO ピンにクロックを出力します。マスタクロックは、クロック周波数を1、2、4、または 16 分周できるプログラマブルプリスケラによるクロック出力の駆動に使用されます。

この選択は、[設定可能クロック出力レジスタ \(CLK\\_CCOR\)](#) の CCOSEL[2:0] ビットによって制御されます。

ターゲットクロックを出力するシーケンスは、[設定可能クロック出力レジスタ \(CLK\\_CCOR\)](#) にて CCOEN=1 を設定することから始まります。CCOEN=1 のとき、CCOSEL[2:0] ビットは書き込み保護されます。

クロック出力を無効にするには、CCOEN ビットをクリアする必要があります。

設定可能なクロック出力には設定可能なスロープがあり、出力周波数に応じて適合させる必要があります。ターゲット出力周波数の 2 倍以上のパッファスロープを選択することを推奨します。

## 8.4 CLK レジスタ

### 8.4.1 クロック分周器レジスタ (CLK\_CKDIVR)

アドレスオフセット : 0x00  
リセット値 : 0x03

7	6	5	4	3	2	1	0
予約済み						HSIDIV[1:0]	
						rw	

ビット 7:2 予約済み

ビット 1:0 **HSIDIV[1:0]** : 高速内部クロックプリスケラ

- 00 :  $f_{MASTER} = f_{HSI\ RC\ output}$
- 01 :  $f_{MASTER} = f_{HSI\ RC\ output}/2$
- 10 :  $f_{MASTER} = f_{HSI\ RC\ output}/4$
- 11 :  $f_{MASTER} = f_{HSI\ RC\ output}/8$

これらのビットは、HSIDIV プリスケール比を定義するために、ソフトウェアによって書き込まれます。

## 8.4.2 ペリフェラルクロックゲーティングレジスタ 1 (CLK\_PCKENR1)

アドレスオフセット : 0x03  
リセット値 : 0x00

7	6	5	4	3	2	1	0
PCKENR1[7:0]							
rw							

ビット 7:0 **PCKENR1[7:0]** : ペリフェラルクロックイネーブル

- 0 : ペリフェラルへの  $f_{\text{MASTER}}$  は無効です。
- 1 : ペリフェラルへの  $f_{\text{MASTER}}$  は有効です。

これらのビットは、対応するペリフェラルへの  $f_{\text{MASTER}}$  クロックを有効／無効にするために、ソフトウェアによって書き込まれます。表 15 を参照してください。

表 15. ペリフェラルクロックゲーティングビット

制御ビット	ペリフェラル
PCKENR17	ProxSense
PCKENR16	AWU (ただし、 <a href="#">制御／ステータスレジスタ (AWU_CSR)</a> の AWUEN ビットによって制御される LSI クロックを除く)。
PCKENR15	USART
PCKENR14	SPI
PCKENR13	I <sup>2</sup> C
PCKENR12	TIM4
PCKENR11	TIM3
PCKENR10	TIM2

## 8.4.3 ペリフェラルクロックゲーティングレジスタ 2 (CLK\_PCKENR2)

アドレスオフセット : 0x04  
リセット値 : 0x01

7	6	5	4	3	2	1	0
予約済み							PCKENR20
							rw

ビット 7:1 予約済み

ビット 0 **PCKENR20** : WWDG ペリフェラルクロックイネーブル

- 0 : ペリフェラルへの  $f_{\text{MASTER}}$  は無効です。
- 1 : ペリフェラルへの  $f_{\text{MASTER}}$  は有効です (リセット値)。

### 8.4.4 設定可能クロック出力レジスタ (CLK\_CCOR)

アドレスオフセット : 0x05  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	CCOSLP[1:0]			CCOSEL[2:0]			CCOEN
	rw			rw			rw

ビット 7:6 予約済み

ビット 5:4 **CCOSLP[1:0]** : 設定可能クロック出力バッファスロープ

00 : 400 kHz      10 : 10 MHz  
01 : 20 MHz      11 : 50 MHz

ビット 3:1 **CCOSEL[2:0]** : 設定可能クロック出力選択

これらのビットは、CLK\_CCO ピンで使用可能な出力クロックのソースを選択するために、ソフトウェアによって書き込まれます。

000 : f<sub>MASTER</sub>      100 : f<sub>PXS</sub>  
001 : f<sub>MASTER</sub>/2      101 : f<sub>PXS</sub>/2  
010 : f<sub>MASTER</sub>/4      110 : f<sub>PXS</sub>/4  
011 : f<sub>MASTER</sub>/16      111 : f<sub>PXS</sub>/16

**注 :** **ProxSense RC 信号を正しく出力するには、以下のシーケンスを守る必要があります。**

- CLK\_PCKENR1 レジスタのビット PCKENR17 をセットすることによって、PXS クロックペリフェラルを有効にします。
- CCOSEL[2:0] を設定し、CLK\_CCOR レジスタのビット CCO をセットします。
- PXS\_CR1 レジスタのビット PXS\_EN をセットすることによって、ProxSense ペリフェラルを有効にします。

ビット 0 **CCOEN** : 設定可能クロック出力イネーブル

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 : CCO クロック出力は無効です。  
1 : CCO クロック出力は有効です。

### 8.4.5 CLK レジスタマップとリセット値

表 16. CLK レジスタマップとリセット値

アドレス オフ セット <sup>(1)</sup>	レジスタ名	7	6	5	4	3	2	1	0
0x00	CLK_CKDIVR リセット値	- 0	- 0	- 0	- 0	- 0	- 0	HSIDIV1 1	HSIDIV0 1
0x01~ 0x02	予約済み領域 (2 バイト)								
0x03	CLK_PCKENR1 リセット値	PCKENR17 0	PCKENR16 0	PCKENR15 0	PCKENR14 0	PCKENR13 0	PCKENR12 0	PCKENR11 0	PCKENR10 0
0x04	CLK_PCKENR2 リセット値	- 0							PCKENR20 1
0x05	CLK_CCOR リセット値	- 0	- 0	CCOSLP1 0	CCOSLP0 0	CCOSEL2 0	CCOSEL1 0	CCOSEL0 0	CCOEN 0

1. レジスタアドレスの詳細については、データシートの「一般ハードウェアレジスタマップ」の表を参照してください。

## 9 電源管理

デフォルトでは、システムリセットまたは電源リセット後、マイクロコントローラは RUN モードです。このモードでは、CPU は  $f_{\text{MASTER}}$  によってクロック駆動されてプログラムコードを実行します。一方、すべてのペリフェラルへのシステムクロックは、ペリフェラルクロックゲーティングシステムによってゲートオフされます。

RUN モードでは、CPU を動作させてコードを実行しながら、アプリケーションは、以下のような方法で消費電力を低減します。

- システムクロックの低速化
- 使用されない個々のペリフェラルへのクロックのゲーティング
- 使用されないアナログ機能のスイッチオフ

しかし、CPU 動作を維持する必要があるときは、3 つの専用低電力モードを使用することができます。

- WAIT (割り込み待ちとイベント待ち)
- アクティブ HALT
- HALT

これらの 3 つのモードのいずれかを選択および設定することで、最低の消費電力、最速の起動時間、および使用可能なウェイクアップソースの間で、最善の妥協点を得ることができます。

### 9.1 一般的な考慮事項

低消費電力機能は、あらゆるタイプのアプリケーションの省エネにとって、一般的にきわめて重要です。超低電力機能は、長いバッテリー寿命を確保するために、モバイルアプリケーションにとって特に重要です。これは、環境保護のためにも重要です。

シリコンチップには、2 種類の消費電力があります。

- **静的消費電力**：アナログ偏波やリークに起因します。リークに起因する消費電力はきわめて小さく、HALT モードとアクティブ HALT モードでのみ意味を持ちます ([セクション 9.4: 低電力モード](#))。
- **動的消費電力**：チップのデジタル部の動作に由来するものであり、 $V_{\text{DD}}$ 、クロック周波数、および負荷コンデンサによって異なります。

マイクロコントローラデバイスでは、消費電力は以下の条件によって異なります。

- $V_{\text{DD}}$  電源電圧
- MCU のサイズまたはデジタルゲートの数 (リークと負荷コンデンサ)
- クロック周波数
- 有効ペリフェラルの数
- 使用可能な低電力モードと低電力レベル

デバイスの処理性能もきわめて重要であり、これによってアプリケーションは、RUN モードでの所要時間を最小限に抑えて、低電力モードでの時間を最大化できます。

MCU の柔軟な電源管理機能を使用すれば、システム動作時に大幅な節電を実現したり、操作を短時間で再開することができます。

## 9.2 低消費電力のためのクロック管理

### 9.2.1 システムクロックの低速化

実行モードでは、パフォーマンスと消費電力の最善の妥協点を得るには、クロック周波数の選択が非常に重要です。選択は、CLK\_CKDIVR レジスタのプログラミングによって行われます。[8.4.1: クロック分周器レジスタ \(CLK\\_CKDIVR\) \(63 ページ\)](#)。

**注:** MCU をある期間停止できるアプリケーションでは、アクティブ期間と HALT 期間 (消費電力ゼロ) の比を小さくするために、アクティブ期間のみ高速クロック (高性能実行) を維持することによって、消費電力を最小化できます。

### 9.2.2 ペリフェラルクロックゲーティング

さらに消費電力を節約するために、ペリフェラルクロックゲーティング (PCG) を使用できます。これは、個々のペリフェラルへの  $f_{\text{MASTER}}$  クロック接続を選択して有効/無効にすることによって、いつでも行うことができます。[8.4.2: ペリフェラルクロックゲーティングレジスタ 1 \(CLK\\_PCKENR1\) \(64 ページ\)](#) および [8.4.3: ペリフェラルクロックゲーティングレジスタ 2 \(CLK\\_PCKENR2\) \(64 ページ\)](#) を参照してください。

これらの設定は、RUN モードと WAIT モードの両方で有効です。[8.1.1: ペリフェラルクロックゲーティング \(PCG\) \(62 ページ\)](#) を参照してください。

各 PCG 状態は、特定の電力レベルまたは低電力レベルを表します。

## 9.3 ペリフェラルのスイッチオフ

消費電力を最小限に抑えるため、どの MCU ペリフェラルでも、使用されていないときは無効にすることができます。これは、アナログペリフェラルとデジタルペリフェラルの両方に当てはまります。

オン/オフの各組み合わせは、特定の電力レベルまたは低電力レベルを表します。

各ペリフェラルは、専用の制御ビットによってスイッチオフすることができます。

- シングルワイヤインタフェースモジュール (SWIM) : SWIM\_CSR レジスタの SWD ビット
- 自動ウェイクアップユニット (AWU) : AWU\_CSR レジスタの AWUEN ビット
- ビープ: AWU\_CSR レジスタの BEEPEN ビット
- タイマ: TIMx\_CR1 レジスタの CEN ビット
- I2C: I2C\_CR1 レジスタの PE ビット
- SPI: SPI\_CR1 レジスタの SPE ビット
- USART: USART\_CR2 レジスタ
- WWDG: WWDG\_CR レジスタ
- ProxSense: PXS\_CR1 レジスタ

**注:** これらの制御ビットは、対応するペリフェラルクロックが有効なときにのみ書き込んでください。IWDG が有効になると、リセット以外の方法では無効にできません。



## 9.4 低電力モード

デフォルトでは、マイクロコントローラは、システムリセットまたは電源リセット後は RUN モードです。ただし、このデバイスは、低消費電力、起動時間の短縮、および使用可能なウェイクアップソースの間で最善の妥協点を得るために、3つの低電力モードをサポートしています。

- **WAIT モード** : CPU クロックは停止しますが、選択されたペリフェラルは実行を続けます。内部または外部割り込み、またはリセットを使用して、マイクロコントローラを WAIT モードから抜けさせることができます。 [セクション 9.4.1 : WAIT モード \(69 ページ\)](#) を参照してください。
- **アクティブ HALT モード** : CPU およびペリフェラルクロックは、LSI を除いて停止します。ウェイクアップは、AWU または PXS 割り込み、外部割り込み、またはリセットによってトリガされます。
- **HALT モード** : CPU およびペリフェラルクロックは停止し、デバイスは電源オンの状態に維持されます。ウェイクアップは、外部割り込みまたはリセットによってトリガされます。少数のペリフェラルは、HALT からのウェイクアップ機能も備えています。

3つの低電力モードの主な特性を [表 17](#) に示します。

表 17. 低電力モードの管理

モード (消費電力レベル)		メイン電圧 レギュレータ (MVR)	オシレータ	CPU	ペリフェラル	ウェイクアップ トリガイメント
WAIT (-)	WFI	オン	オン	オフ	オン <sup>(1)</sup>	すべての内部または外部割り込み、リセット
	WFE	オン	オン	オフ	オン <sup>(1)</sup>	すべての内部または外部割り込み <sup>(2)</sup> 、WFE イベント、リセット
アクティブ HALT (--)	AWU	オフ (LPVR オン)	HSI オフ、 LSI オン	オフ	AWU、BEEP また IWDG が有効であり、「HALT 状態でウォッチドッグなし」オプションが無効である場合はのみ	AWU または外部 <sup>(3)</sup> 割り込み、リセット
	PXS	オン	HSI オフ、 LSI オフ、 HSI_PXS オン	オフ	PXS	PXS、外部 <sup>(3)</sup> 割り込みまたはリセット
HALT (---)		オフ (LPVR オン)	IWDG が有効であり、「HALT 状態でウォッチドッグなし」オプションが無効である場合は、LSI オン	オフ	BEEP と IWDG が有効であり、「HALT 状態でウォッチドッグなし」オプションが無効である場合	外部割り込み <sup>(3)</sup> 、リセット

1. ペリフェラルクロックがペリフェラルクロックゲーティング機能によって無効にされていない場合。
2. [イベント待ち \(WFE\) モード \(69 ページ\)](#) を参照してください。
3. 通信ペリフェラル割り込みを含む (割り込みベクタテーブルを参照)。

さらに、次のような手段で 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
- 使用されないペリフェラルクロックのゲーティング

[セクション 9.2 : 低消費電力のためのクロック管理 \(67 ページ\)](#) を参照してください。



## 9.4.1 WAIT モード

WFI（割り込み待ち）または WFE（イベント待ち）命令を実行することによって、RUN モードから WAIT モードに移行します。CPU は停止しますが、他のペリフェラルと割り込みコントローラは実行を続けます。その結果として消費電力が低下します。WAIT モードと PCG を組み合わせることによって、デバイスの消費電力をさらに削減できます。[セクション 8.1.1：ペリフェラルクロックゲーティング \(PCG\) \(62 ページ\)](#)。

WAIT モードでは、すべてのレジスタと RAM の内容が保持され、[クロック分周レジスタ \(CLK\\_CKDIVR\)](#) を通じて選択されたクロック設定がそのまま保持されます。

### 割り込み待ち (WFI) モード

内部または外部割り込みリクエストが発生すると、CPU は WFI モードからウェイクアップし、割り込みに対応し、処理を再開します。

**注：** *処理のほとんどが割り込みルーチンを通じて行われる割り込みベースのアプリケーションでは、CFG\_GCR レジスタのアクティベーションレベルビット (AL) をセットすることによって、メインプログラムの実行コンテキストを中断できます。このビットをセットすると、CPU は、メインプログラムの実行コンテキストを復元せずに、WFI モードに戻ります。このため、コンテキストの保存／復元作業の両方と、(WFI モードに戻るための) 電源管理のためのメインソフトウェアループが実行されなくなるので、消費電力が低下します。*

### イベント待ち (WFE) モード

CPU を WFE モードからウェイクアップさせるには、2 つの方法があります（[表 17：低電力モードの管理 \(68 ページ\)](#) を参照）。

- 割り込みが発生すると、
  - － CPU は WFE モードからウェイクアップして、割り込みに対応します。割り込みを処理した後、プロセッサは WFE モードに戻ります。
- WFE イベントが発生すると、
  - － CPU はウェイクアップし、処理を再開します。処理は WFE 命令の直後から再開されるので、コンテキストの保存／復元作業は行われません（これによって時間と消費電力が節減されます）。

**注：** *WFE モードでは、WFE\_CR1 または WFE\_CR2 レジスタの対応するステータスフラグがクリアされている場合にのみ、割り込みソースは外部割り込みとして設定されます。そうでない場合は、WFE イベントが生成されます（割り込みは処理されず、ユーザは EXTI\_SR1 または EXTI\_SR2 レジスタの対応するステータスフラグを適切にクリアする必要があります）。[セクション 9.5：WFE レジスタ \(71 ページ\)](#)、[外部割り込みステータスレジスタ 1 \(EXTI\\_SR1\) \(53 ページ\)](#)、および[外部割り込みステータスレジスタ 2 \(EXTI\\_SR2\) \(54 ページ\)](#) を参照してください。*

このモードと RAM からの実行を組み合わせることにより、消費電力をさらに節減できます。一部の超低電力アプリケーションでは、メインのソフトウェアルーチンが短く、実行時間が短いときには、このルーチンを RAM に移動して、RAM から実行できる場合もあります。フラッシュプログラムメモリはウェイクアップ時は使用されないため、実行時間中の消費電力が減ります。

このルーチンをコール／ジャンプするだけで、いつでも、（フラッシュプログラムメモリに格納されている）別のルーチンをソフトウェアによって実行できます。

## 9.4.2 HALT モード

このモードでは、マスタクロックは停止します。これは、CPU と、 $f_{\text{MASTER}}$  またはそこから生成されたクロックによってクロック供給されているすべてのペリフェラルが無効になることを意味します。その結果、どのペリフェラルもクロック供給されず、MCU のデジタル部分はほとんど電力を消費しません。

MCU が HALT モードに入ると、メインボルテージレギュレータ (MVR) は、自動的にスイッチオフされます。MCU コアは低消費ボルテージレギュレータ (LPVR) によってのみ電源供給され、すべてのレジスタと RAM の内容が保持されます。

HALT 命令が実行されると、MCU は HALT モードに入ります。HALT モードからのウェイクアップは、割り込み入力に設定されている汎用 I/O ポート、またはペリフェラル割り込みをトリガできる代替機能ピンによる外部割り込みによってトリガされます。

処理のほとんどが割り込みルーチンを通じて行われる割り込みベースのアプリケーションでは、CPU 設定レジスタのアクティベーションレベルビット (AL) をセットすることによって、メインプログラムを中断できます。このビットをセットすると、CPU は割り込みから戻るときに、メインプログラムの実行コンテキストを復元せずに、HALT モードに戻ります。

コンテキストの保存/復元作業と、(WFI モードに戻るための) 電源管理のためのメインレベルソフトウェアループが実行されなくなるので、消費電力が削減されます。

**注：** HALT モードに入る前に、ProxSense ペリフェラルは、無効にするか低電力モードにする必要があります (PXS\_CR1 レジスタの PXS\_EN=0 または LOW\_POWER=1)。

## 9.4.3 アクティブ HALT モード

アクティブ HALT モードは HALT モードと似ていますが、ウェイクアップのために外部割り込みは必要ありません。AWU を使用してプログラマブルな遅延の後でウェイクアップイベントを内部的に生成するか、PXS を使用して PXS 割り込みの発生後にウェイクアップイベントを生成します。

### AWU によるアクティブ HALT モード

AWU によるアクティブ HALT モードに入るには、まず、AWU のセクションで説明されているように AWU を有効にします。次に、HALT 命令を実行します。

AWU によるアクティブ HALT モードでは、メインオシレータ、CPU、およびほとんどすべてのペリフェラルが停止します。AWU カウンタ、BEEP、および IWDG カウンタが有効な場合、LSI RC オシレータのみが動作して、これらを駆動します。

MCU がアクティブ HALT モードに入ると、メインボルテージレギュレータ (MVR) は、自動的にスイッチオフされます。MCU コアは低消費ボルテージレギュレータ (LPVR) によってのみ電源供給され、すべてのレジスタと RAM の内容が保持されます。

AWU によるアクティブ HALT モードの使用時に消費電力を低減するため、ProxSense を無効にする (PXS\_CR1 レジスタの PXS\_EN=0) か低電力モードにする (PXS\_CR1 レジスタの LOW\_POWER=1) 必要があります。

### ProxSense によるアクティブ HALT モード

ProxSense によるアクティブ HALT モードに入るには、アクイジションが現在行われている必要があります (PXS\_ISR レジスタの CIPF=1)。同期信号に基づいてアクイジションがペンディング中である場合、HALT 命令の実行前に同期トリガが発生しない限り、デバイスは、ProxSense によるアクティブ HALT モードに入りません。これをチェックするには、PXS\_ISR レジスタの CIPF ビットをポーリングします。

ProxSense によるアクティブ HALT モードでは、HSI および LSI オシレータ、CPU、およびほとんどすべてのペリフェラルが停止します。HSI\_PXS オシレータのみが動作して、アナログ ProxSense システムを駆動します。

メインボルテージレギュレータ（MVR）は常にオンに保持され、MCU コアクロックはスイッチオフされ、すべてのレジスタと RAM の内容が保持されます。

## 9.5 WFE レジスタ

これらのレジスタは、さまざまな割り込みソースを外部割り込みや WFE イベントとして設定するために使用されます。[セクション：イベント待ち（WFE）モード（69 ページ）](#)を参照してください。

### 9.5.1 WFE 制御レジスタ 1（WFE\_CR1）

アドレスオフセット：0x00

リセット値：0x00

7	6	5	4	3	2	1	0
EXTI_EV3	EXTI_EV2	EXTI_EV1	EXTI_EV0	PXS_EV	予約済み	TIM2_EV1	TIM2_EV0
rw	rw	rw	rw	rw		rw	rw

- ビット 7 **EXTI\_EV3**：外部割り込みイベント 3
- 0：すべてのポートのピン 3 からの割り込みソースは、外部割り込みを生成します。
  - 1：すべてのポートのピン 3 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 6 **EXTI\_EV2**：外部割り込みイベント 2
- 0：すべてのポートのピン 2 からの割り込みソースは、外部割り込みを生成します。
  - 1：すべてのポートのピン 2 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 5 **EXTI\_EV1**：外部割り込みイベント 1
- 0：すべてのポートのピン 1 からの割り込みソースは、外部割り込みを生成します。
  - 1：すべてのポートのピン 1 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 4 **EXTI\_EV0**：外部割り込みイベント 0
- 0：すべてのポートのピン 0 からの割り込みソースは、外部割り込みを生成します。
  - 1：すべてのポートのピン 0 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 3 **PXS\_EV**：ProxSense 割り込み
- 0：PXS 割り込みソースは、外部割り込みを生成します。
  - 1：PXS 割り込みソースは、論理和がとられ、WFE イベントを生成するように設定されます（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 2 予約済み

- ビット 1 **TIM2\_EV1** : TIM2 イベント 1
- 0 : TIM2 キャプチャおよび比較割り込みソースは、外部割り込みを生成します。
  - 1 : TIM2 キャプチャおよび比較割り込みソースは、論理和がとられ、WFE イベントを生成するように設定されます（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 0 **TIM2\_EV0** : TIM2 イベント 0
- 0 : TIM2 更新、トリガ、およびブレーク割り込みソースは、外部割り込みを生成します。
  - 1 : TIM2 更新、トリガ、およびブレーク割り込みソースは、論理和がとられ、WFE イベントを生成するように設定されます（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。

9.5.2 WFE 制御レジスタ 2 (WFE\_CR2)

アドレスオフセット : 0x01

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	EXTI_EVD	EXTI_EVB	EXTI_EV7	EXTI_EV6	EXTI_EV5	EXTI_EV4	
	rw	rw	rw	rw	rw	rw	rw

- ビット 7:6 予約済み
- ビット 5 **EXTI\_EVD** : ポート D 上の外部割り込みイベント
- 0 : ポート D からの割り込みソースは、外部割り込みを生成します。
  - 1 : ポート D からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 4 **EXTI\_EVB** : ポート B 上の外部割り込みイベント
- 0 : ポート B からの割り込みソースは、外部割り込みを生成します。
  - 1 : ポート B からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 3 **EXTI\_EV7** : 外部割り込みイベント 7
- 0 : すべてのポートのピン 7 からの割り込みソースは、外部割り込みを生成します。
  - 1 : すべてのポートのピン 7 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 2 **EXTI\_EV6** : 外部割り込みイベント 6
- 0 : すべてのポートのピン 6 からの割り込みソースは、外部割り込みを生成します。
  - 1 : すべてのポートのピン 6 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。
- ビット 1 **EXTI\_EV5** : 外部割り込みイベント 5
- 0 : すべてのポートのピン 5 からの割り込みソースは、外部割り込みを生成します。
  - 1 : すべてのポートのピン 5 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。
- このビットはソフトウェアによって書き込まれます。



ビット 0 **EXTI\_EV4** : 外部割り込みイベント 4

0 : すべてのポートのピン 4 からの割り込みソースは、外部割り込みを生成します。

1 : すべてのポートのピン 4 からの割り込みソースは、WFE イベントを生成します（割り込みは処理されません）。

このビットはソフトウェアによって書き込まれます。

## 9.6 WFE レジスタマップとリセット値

表 18. WFE レジスタマップ

アドレス オフ セット <sup>(1)</sup>	レジスタ名	7	6	5	4	3	2	1	0
0x00	WFE_CR1 リセット値	EXTI_EV3 0	EXTI_EV2 0	EXTI_EV1 0	EXTI_EV0 0	PXS_EV 0	- 0	TIM2_EV1 1	TIM2_EV0 1
0x01	WFE_CR2 リセット値	- 0	- 0	EXTI_EVD 0	EXTI_EVB 0	EXTI_EV7 0	EXTI_EV6 0	EXTI_EV5 0	EXTI_EV4 0

1. レジスタアドレスの詳細については、データシートの「一般ハードウェアレジスタマップ」の表を参照してください。

## 10 汎用入出力ポート (GPIO)

### 10.1 概要

汎用入出力ポートは、チップと周辺回路との間のデータ転送に使用されます。入出力ポートは、最大 8 本のピンを内蔵できます。各ピンは、デジタル入力またはデジタル出力として個別にプログラミングすることができます。さらに、アナログ入力、外部割り込み、オンチップペリフェラル用の入出力などのオルタネート機能を備えたポートもあります。ピンには、一度に 1 つのオルタネート機能しか配置できません。

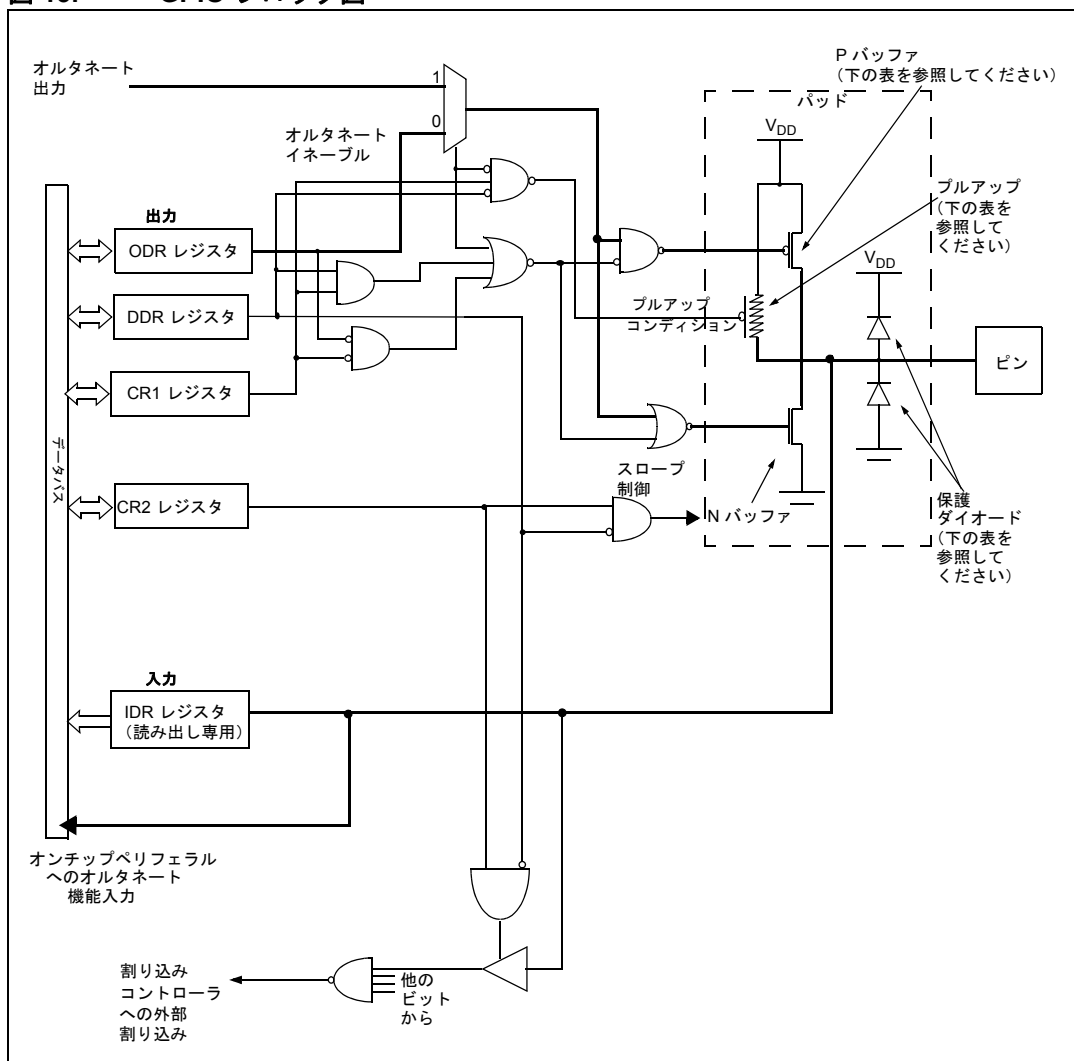
出力データレジスタ、入力データレジスタ、データ方向レジスタ、および 2 つの設定レジスタは、各ポートに関連付けられます。特定のポートは、ポートのデータ方向レジスタのステータスに応じて、入力または出力として動作します。

### 10.2 GPIO の主な機能

- ポートビットは個別に設定できます。
- 選択可能な入力モード：フローティング入力またはプルアップ付き入力。
- 選択可能な出力モード：プッシュプル出力または疑似オープンドレイン。
- データの入出力に別個のレジスタ。
- 外部割り込みを個別に有効／無効にすることが可能。
- 出力スロープ制御によって EMC ノイズを低減。
- オンチップペリフェラル用のオルタネート機能入出力。
- データ出力のラッチ時に読み出し／変更／書き込みが可能。
- 1.6 V ~  $V_{DDIOmax}$  の電圧範囲で入出力状態を保証。



図 13. GPIO ブロック図



## 10.3 ポートの設定と使い方

出力データレジスタ (ODR)、ピン入力レジスタ (IDR)、データ方向レジスタ (DDR) は、常に各ポートに関連付けられています。

制御レジスタ 1 (CR1) と制御レジスタ 2 (CR2) によって、入出力オプションが可能になります。入出力ピンは、DDR、ODR、CR1、および CR2 レジスタの対応するビットを使用してプログラミングされます。

レジスタのビット  $n$  は、ポートのピン  $n$  に対応します。

表 19 に、さまざまな設定の要約を示します。



表 19. 入出力ポート設定の要約

モード	DDR ビット	CR1 ビット	CR2 ビット	機能	プルアップ	P バッファ	ダイオード	
							V <sub>DD</sub> へ	V <sub>SS</sub> へ
入力	0	0	0	割り込みのない フローティング	オフ	オフ	オン	オン
	0	1	0	割り込みのないプルアップ	オン			
	0	0	1	割り込みのある フローティング	オフ			
	0	1	1	割り込みのあるプルアップ	オン			
出力	1	0	0	オープンドレイン出力	オフ	オフ	オン	オン
	1	1	0	プッシュプル出力		オン		
	1	0	1	オープンドレイン出力、 高速モード		オフ		
	1	1	1	プッシュプル、高速モード	オフ	オン		
	1	x	x	真のオープンドレイン (特定のピンで)	実装なし		実装 なし <sup>(1)</sup>	

1. V<sub>DD</sub> に接続されるダイオードは、真のオープンドレインパッドでは実装されません。デバイスを正ストレスから保護するために、パッドと V<sub>OL</sub> の間にローカル保護が実装されています。

**警告：**                    パッケージによっては、いくつかのポートは、パッケージに存在しない場合でも、アクティブとみなす必要があります。スプリアス効果を回避するには、起動時に割り込みのないプルアップ入力として設定し、ポート設定の変更時にはこの状態に保持します。詳細については、データシートを参照してください。

10.3.1    入力モード

DDR<sub>x</sub> ビットをクリアすると、入力モードが選択されます。このモードでは、IDR ビットを読み出すと、対応する入出力ピンのデジタル値が返されます。

アナログ入力、外部割り込み、およびシュミットトリガの有効化／無効化については、[セクション 10.7：入力モードの詳細 \(77 ページ\)](#) を参照してください。

**表 19** に示すように、理論的には、ソフトウェアによって 4 つの異なる入力モード（割り込みのないフローティング、割り込みのあるフローティング、割り込みのないプルアップ、割り込みのあるプルアップ）を設定できます。ただし、実際には、すべてのポートが外部割り込み機能やプルアップを備えているわけではありません。各ポートの実際のハードウェア機能については、データシートのピン配置説明を参照してください。

10.3.2    出力モード

DDR<sub>x</sub> ビットをセットすると、出力モードが選択されます。このモードでは、ODR ビットへの書き込みによって、デジタル値がラッチを介して入出力に適用されます。IDR ビットを読み出すと、対応する入出力ピンからデジタル値が返されます。CR1 および CR2 レジスタを使用して、異なる出力モード（プッシュプル出力、オープンドレイン出力）をソフトウェアによって設定できます。



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

詳細については、[セクション 10.8 : 出力モードの詳細 \(78 ページ\)](#) を参照してください。

## 10.4 リセット設定

すべての入出力ピンは、リセット中（すなわち、リセットフェーズ中）とリセット状態時（すなわち、リセットリリース後）には、一般に入力フローティングになっています。ただし、いくつかのピンでは動作が異なる場合もあります。詳細については、データシートのピン配置説明を参照してください。

## 10.5 未使用の入出力ピン

未使用の入出力ピンは、固定の電圧レベルに接続し、入力フローティングとして設定する必要があります。未使用の入出力ピンにはプルアップまたはプルダウンを接続するか、あるいは内部ウィークプルアップを使用してください（それらのピンで利用できる場合）。

## 10.6 低電力モード

表 20. 低電力モードが GPIO ポートに与える影響

モード	説明
WAIT	入出力ポートに影響を与えません。外部割り込みによって、デバイスは WAIT モードから抜けます。
HALT	入出力ポートに影響を与えません。外部割り込みによって、デバイスは HALT モードからウェイクアップします。

## 10.7 入力モードの詳細

### 10.7.1 オルタネート機能入力

いくつかの入出力は、オルタネート機能入力として使用できます。たとえば、ポートはタイマへの入力キャプチャ入力として使用できます。オルタネート機能入力は自動的に選択されません。選択するには、対応するペリフェラルのレジスタの制御ビットに書き込みを行います。オルタネート機能入力の場合、DDR および CR1 レジスタでフローティングまたはプルアップの入力設定を選択してください。

## 10.7.2 割り込み機能

各入出力を割り込み機能のある入力として設定するには、入出力が入力モードの時 CR2x ビットをセットします。この設定では、入出力でのレベル入力や信号エッジで割り込みリクエストが生成されます。

立ち下がり／立ち上がりエッジの極性は、EXTI\_CR[2:1] レジスタで割り込みベクタごとに個別にプログラムされます。

外部割り込み機能は、ポートが入力モードに設定された場合にのみ使用できます。

### 割り込みのマスク

割り込みを個別に有効／無効にするには、設定レジスタ (Px\_CR2) の対応するビットをプログラムします。リセット状態では、割り込みは無効です。

ピンのオルタネート機能が TLI である場合、Px\_CR2 ビットを使用して TLI 割り込みを有効／無効にします。TLI 割り込みは専用の割り込みベクタに関連付けられます。

## 10.8 出力モードの詳細

### 10.8.1 代替機能出力

オルタネート機能出力は、ペリフェラルから出力または入出力パッドへのダイレクトパスを提供し、データ出力ラッチレジスタ (Px\_ODR) のポートビットよりも優先され、Px\_DDR の対応するビットを強制的に 1 にします。

オルタネート機能出力は、ペリフェラルと制御レジスタ 1 (Px\_CR1) に応じて、プッシュプルまたは疑似オープンドレインとすることができます。スロープは、制御レジスタ 2 (Px\_CR2) の値に応じて制御できます。

**例：**

SPI 出力は、プッシュプルとしてセットアップする必要があります。SPI 出力のスロープは、ハードウェアによって制御され、最適な動作を可能にするために高速モードで設定されます。したがって、スプリアス割り込みを回避するため、CR2 スロープ制御ビットをクリアした状態にしておく必要があります。

### 10.8.2 スロープ制御

入出力に印加できる最大周波数は、CR2 ビットを使用して、ソフトウェアによって制御できます。リセット時には、EMC 動作が改善された低周波数動作が選択されます。必要ならば、より高い周波数 (最大 10 MHz) を選択できます。この機能は、出力タイプ O3 または O4 の入出力ポートで、オープンドレインとプッシュプルの両方の出力モードに適用できます。各ピンの特定の出力タイプ情報については、データシートのピン詳細テーブルを参照してください。

## 10.9 GPIO レジスタ

各ポートレジスタのビットは、ポートの対応するピンを駆動します。

### 10.9.1 ポート x 出力データレジスタ (Px\_ODR)

アドレスオフセット : 0x00  
リセット値 : 0x00

7	6	5	4	3	2	1	0
ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 7:0 ODR[7:0] : 出力データレジスタのビット

出力モードで ODR レジスタに書き込むと、デジタル値がラッチを介して入出力に適用されます。ODR レジスタを読み出すと、レジスタ内の以前にラッチされた値が返されます。  
入力モードでは、ODR レジスタに書き込むと、値がレジスタにラッチされますが、ピン状態は変化しません。リセット後、ODR レジスタは常にクリアされます。DR レジスタでは、ビット読み出し／変更／書き込み命令 (BSET、BRST) を使用して、他のピンに影響を与えずに、個々のピンを駆動できます。

### 10.9.2 ポート x ピン入力レジスタ (Px\_IDR)

アドレスオフセット : 0x01  
リセット値 : 0xXX

7	6	5	4	3	2	1	0
IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r

ビット 7:0 IDR[7:0] : ピン入力値

ピンレジスタを使用して、ポートが入力モードか出力モードかに関係なく、ピン値を読み出すことができます。このレジスタは読み出し専用です。  
0 : Low レベル入力  
1 : High レベル入力

**注 :** Px\_IDR のリセット値は、外部回路によって異なります。

## 10.9.3 ポート x データ方向レジスタ (Px\_DDR)

アドレスオフセット : 0x02  
リセット値 : 0x00

7	6	5	4	3	2	1	0
DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
rw	rw	rw	rw	rw	rw	rw	rw

ビット 7:0 **DDR[7:0]** : データ方向ビット  
これらのビットは、ポートの特定のピンの入力または出力モードを選択するために、ソフトウェアによってセット/クリアされます。  
0 : 入力モード  
1 : 出力モード

## 10.9.4 ポート x 制御レジスタ 1 (Px\_CR1)

アドレスオフセット : 0x03  
リセット値 : 0x00 (ただし、PA\_CR1 のみ 0x01)

7	6	5	4	3	2	1	0
C17	C16	C15	C14	C13	C12	C11	C10
rw	rw	rw	rw	rw	rw	rw	rw

ビット 7:0 **C1[7:0]** : 制御ビット  
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。入力モードと出力モードで異なる機能を選択します (表 19 (76 ページ) を参照してください)。  
- **入力モードの場合 (DDR = 0) :**  
0 : フローティング入力  
1 : プルアップ付き入力  
- **出力モードの場合 (DDR = 1) :**  
0 : 疑似オープンドレイン  
1 : 出力のプッシュプル、スロープ制御は対応する CR2 ビットに依存する  
**注 :** このビットは、真のオープンドレインポートには影響がありません (データシートのピン記述テーブルで "T" とマークされたピンを参照)。

## 10.9.5 ポート x 制御レジスタ 2 (Px\_CR2)

アドレスオフセット : 0x04  
リセット値 : 0x00

7	6	5	4	3	2	1	0
C27	C26	C25	C24	C23	C22	C21	C20
rw	rw	rw	rw	rw	rw	rw	rw

ビット 7:0 **C2[7:0]** : 制御ビット

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。入力モードと出力モードで異なる機能を選択します。入力モードでは、CR2 ビットは割り込み機能を有効にします（使用可能な場合）。入出力に割り込み機能がない場合、CR2 ビットをセットしても影響はありません。出力モードでは、このビットをセットすると入出力の速度が向上します。これは、O3 と O4 の出力タイプのポートに適用されます（ピン記述テーブルを参照）。

- **入力モードの場合 (DDR = 0) :**
  - 0 : 外部割り込み無効
  - 1 : 外部割り込み有効
- **出力モードの場合 (DDR = 1) :**
  - 0 : 最大 2 MHz の出力速度
  - 1 : 最大 10 MHz の出力速度

## 10.9.6 GPIO レジスタマップとリセット値

各 GPIO ポートには 5 つのレジスタがあり、表 21 に示すように配置されます。各ポートのベースアドレスについては、対応するデータシートのレジスタマップを参照してください。

**注 :** リセット状態では、すべてのポートは入力フローティングです。例外については、対応するデータシートのピン記述テーブルに示されています。

表 21. GPIO レジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	Px_ODR	ODR7 0	ODR6 0	ODR5 0	ODR4 0	ODR3 0	ODR2 0	ODR1 0	ODR0 0
0x01	Px_IDR	IDR7 x	IDR6 x	IDR5 x	IDR4 x	IDR3 x	IDR2 x	IDR1 x	IDR0 x
0x02	Px_DDR	DDR7 0	DDR6 0	DDR5 0	DDR4 0	DDR3 0	DDR2 0	DDR1 0	DDR0 0
0x03	Px_CR1	C17 0	C16 0	C15 0	C14 0	C13 0	C12 0	C11 0	C10 0
0x04	Px_CR2	C27 0	C26 0	C25 0	C24 0	C23 0	C22 0	C21 0	C20 0

## 11 自動ウェイクアップ (AWU)

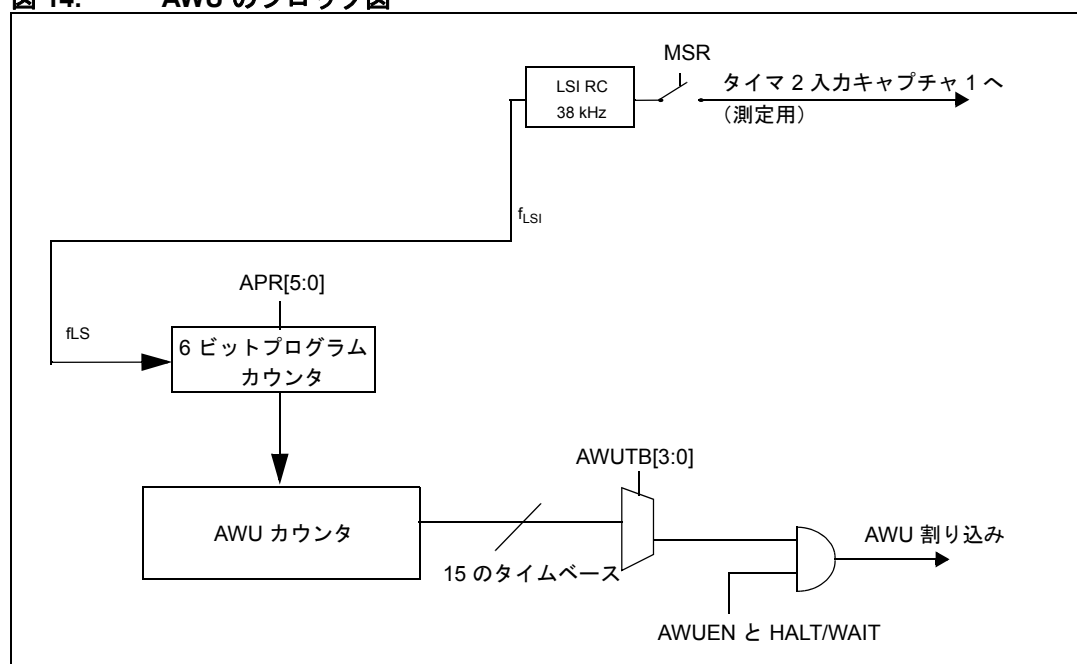
### 11.1 概要

AWU は、MCU がアクティブ HALT 低消費電力モードに入るときに使用される、内部ウェイクアップのタイムベースを提供するために使用されます。このタイムベースは、低速内部 (LSI) RC オシレータクロックによってクロック供給されます。

### 11.2 LSI クロックの測定

LSI クロックを使用するときに最良の精度を確保するため、その周波数は、TIM2 のタイマ入力キャプチャ 1 によって測定することができます。

図 14. AWU のブロック図





## 11.3 AWU 機能の説明

### 11.3.1 AWU 動作

AWU を使用するには、以下の手順を順番に実行してください。

1. LS クロック周波数の測定 : AWU\_CSR レジスタの MSR ビットと TIM 入力キャプチャ 1 を使用します。
2. 適切なプリスケアラ値の定義 : [非同期プリスケアラレジスタ \(AWU\\_APR\)](#) の APR [5:0] ビットに書き込みます。
3. 希望する自動ウェイクアップ遅延の選択 : [タイムベース選択レジスタ \(AWU\\_TBR\)](#) の AWUTB[3:0] ビットに書き込みます。
4. [制御/ステータスレジスタ \(AWU\\_CSR\)](#) の AWUEN ビットをセットします。
5. HALT 命令を実行します。AWU カウンタが再ロードされ、新しい AWU タイムインターバルのカウンタが開始されます。

**注 :** カウンタが開始されるのは、HALT 命令の後で MCU がアクティブ HALT モードに入ったときのみです (電源管理の章のアクティブ HALT モードのセクションを参照してください)。これによって、同時に AWU 割り込みが有効になります。

プリスケアラカウンタがカウントを開始するのは、APR[5:0] の値がそのリセット値 (0x3F) とは異なる場合のみです。

#### アイドルモード

AWU が使用中でない場合は、消費電力を減らすために、[タイムベース選択レジスタ \(AWU\\_TBR\)](#) の AWUTB[3:0] ビットに 0b0000 をロードします。

## 11.3.2 タイムベースの選択

非同期プリスケアラレジスタ (AWU\_APR) と タイムベース選択レジスタ (AWU\_TBR) の説明を参照してください。

AWU タイムインターバルは、以下の値に依存します：

- AWUTB[3:0] ビット。これはカウンタの出力ランクを示します。
- APR[5:0] ビット。これはプリスケアラの分周比 ( $APR_{DIV}$ ) を示します。

タイムインターバルの 15 個の非重複範囲は、次のように定義できます。

AWUTB[3:0] と  $APR_{DIV}$  に適切な値を得るには、次の操作が必要です。

表 22. タイムベースの計算テーブル

インターバル範囲		AWUTB[3:0]	タイムインターバル 計算用の $APR_{DIV}$ 式	$APR_{DIV}$ 範囲
$f_{LS} = f$	$f_{LS} = 38 \text{ kHz}$			
$2/f - 64/f$	0.53 ms - 1.68 ms	0001	$APR_{DIV}/f_{LS}$	2~64
$2 \times 32/f - 2 \times 2 \times 32/f$	1.68 ms - 3.37 ms	0010	$2 \times APR_{DIV}/f_{LS}$	32~64
$2 \times 64/f - 2 \times 2 \times 64/f$	3.37 ms - 6.74 ms	0011	$2^2 \times APR_{DIV}/f_{LS}$	32~64
$2^2 \times 64/f - 2^2 \times 128/f$	6.74 ms - 13.47 ms	0100	$2^3 \times APR_{DIV}/f_{LS}$	32~64
$2^3 \times 64/f - 2^3 \times 128/f$	13.47 ms - 26.95 ms	0101	$2^4 \times APR_{DIV}/f_{LS}$	32~64
$2^4 \times 64/f - 2^4 \times 128/f$	26.95 ms - 53.89 ms	0110	$2^5 \times APR_{DIV}/f_{LS}$	32~64
$2^5 \times 64/f - 2^5 \times 128/f$	53.89 ms - 107.8 ms	0111	$2^6 \times APR_{DIV}/f_{LS}$	32~64
$2^6 \times 64/f - 2^6 \times 128/f$	107.8 ms - 215.6 ms	1000	$2^7 \times APR_{DIV}/f_{LS}$	32~64
$2^7 \times 64/f - 2^7 \times 128/f$	215.6 ms - 431.2 ms	1001	$2^8 \times APR_{DIV}/f_{LS}$	32~64
$2^8 \times 64/f - 2^8 \times 128/f$	431.2 ms - 862.3 ms	1010	$2^9 \times APR_{DIV}/f_{LS}$	32~64
$2^9 \times 64/f - 2^9 \times 128/f$	862.3 ms - 1.72 s	1011	$2^{10} \times APR_{DIV}/f_{LS}$	32~64
$2^{10} \times 64/f - 2^{10} \times 128/f$	1.72 s - 3.45 s	1100	$2^{11} \times APR_{DIV}/f_{LS}$	32~64
$2^{11} \times 64/f - 2^{11} \times 128/f$	3.45 s - 6.90 s	1101	$2^{12} \times APR_{DIV}/f_{LS}$	32~64
$2^{11} \times 130/f - 2^{11} \times 320/f$	7.01 s - 17.25 s	1110	$5 \times 2^{11} \times APR_{DIV}/f_{LS}$	26~64
$2^{11} \times 330/f - 2^{12} \times 960/f$	17.79 s - 103.48 s	1111	$30 \times 2^{11} \times APR_{DIV}/f_{LS}$	11~64

- 最初に、希望するタイムインターバルに対応するインターバル範囲を検索します。これで AWUTB[3:0] の値が得られます。
- 続いて、希望する値にできるだけ近いタイムインターバル値が得られるよう、 $APR_{DIV}$  を選択することができます。それには、上の表に記載された式を使用します。

注： 目標値が  $2^{11} \times 128/f_{LS} \sim 2^{11} \times 130/f_{LS}$  または  $2^{11} \times 320/f_{LS} \sim 2^{11} \times 330/f_{LS}$  にある場合は、より目標値に近い値を選択する必要があります。

**例 1**

- $f_{LS} = 38 \text{ kHz}$
- 目標タイムインターバル = 16 ms

適切なインターバル範囲 : 13.47 ms - 26.95 ms

したがって、AWUTB[3:0] 値は 0x5 です。

APR<sub>DIV</sub> 値 :

$$16 \text{ ms} = 2^4 \times \text{APR}_{\text{DIV}} / f_{LS} \Rightarrow \text{APR}_{\text{DIV}} = (16 \times 10^{-3} \times f_{LS}) / 2^4 = 38$$

したがって、APR[5:0] 値は 38 (0x26) です。

**例 2**

- $f_{LS} = 38 \text{ kHz}$
- 目標タイムインターバル = 20 ms

適切なインターバル範囲 : 13.47 ms - 26.95 ms

したがって、AWUTB[3:0] 値は 0x5 です。

APR<sub>DIV</sub> 値 :

$$20 \text{ ms} = 2^4 \times \text{APR}_{\text{DIV}} / f_{LS} \Rightarrow \text{APR}_{\text{DIV}} = (20 \times 10^{-3} \times f_{LS}) / 2^4 = 47.5$$

したがって、AWUTB[3:0] は 47 または 48 とすることができます (それぞれ、19.79 ms または 20.21 ms のタイムベースを与えます)。これは、厳密には 20 ms ではありません。

### 11.3.3 LSI クロック周波数の測定

オシレータ周波数のばらつきによって、正確な AWU タイムインターバルまたはビーパ出力を得るには、正確な LSI 周波数を測定する必要があります。

以下の手順に従ってください。

1. [制御/ステータスレジスタ \(AWU\\_CSR\)](#) の MSR ビットをセットして、LSI クロックをタイマ入力キャプチャに内部接続します。
2. タイマ入力キャプチャ割り込みを使用して、LSI クロックの周波数を測定します。
3. [非同期プリスケアラレジスタ \(AWU\\_APR\)](#) の APR [5:0] ビットに適切な値を書き込み、AWU タイムインターバルを希望する長さに調整します。AWUTB[3:0] ビットを変更して、さまざまなタイムインターバルを選択できます。

LSI クロック周波数の測定は、ビーパ周波数の較正にも使用できます ([セクション 12.2.2](#) を参照)。

## 11.4 AWU レジスタ

### 11.4.1 制御／ステータスレジスタ (AWU\_CSR)

アドレスオフセット：0x00

リセット値：0x00

7	6	5	4	3	2	1	0
予約済み	AWUF		AWUEN	予約済み			MSR
	rc_r		rw				rw

ビット 7:6 予約済み

ビット 5 **AWUF**：自動ウェイクアップフラグ

このビットは、AWU モジュールが割り込みを生成したときにハードウェアによってセットされ、AWU\_CSR レジスタの読み出しによってクリアされます。このビットに書き込んでも、その値は変化しません。

- 0：AWU 割り込みは発生しませんでした。
- 1：AWU 割り込みが発生しました。

ビット 4 **AWUEN**：自動ウェイクアップイネーブル

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。これは自動ウェイクアップ機能を有効にします。マイクロコントローラがアクティブ HALT または WAIT モードに入った場合、AWU 機能は、プログラマブルな時間遅延の後でマイクロコントローラをウェイクアップします。

- 0：AWU（自動ウェイクアップ）は無効です。
- 1：AWU（自動ウェイクアップ）は有効です。

ビット 3:1 予約済み

ビット 0 **MSR**：測定イネーブル

このビットは、 $f_{LS}$  クロックをタイマ入力キャプチャに接続します。これによって、タイマを LS 周波数 ( $f_{LS}$ ) の測定に使用できます。

- 0：測定は無効です。
- 1：測定は有効です。

**注：** 特定の製品で LSI クロックに接続できるタイマ入力キャプチャについては、データシートを参照してください。

### 11.4.2 非同期プリスケアラレジスタ (AWU\_APR)

アドレスオフセット：0x01

リセット値：0x3F

7	6	5	4	3	2	1	0
予約済み	APR[5:0]						
	rw						

ビット 7:6 予約済み

ビット 5:0 **APR[5:0]**：非同期プリスケアラ分周回路

これらのビットは、カウンタクロックを供給するプリスケアラ分周回路 ( $APR_{DIV}$ ) 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。

- |                       |                        |
|-----------------------|------------------------|
| 0x00： $APR_{DIV} = 2$ | 0x0E： $APR_{DIV} = 16$ |
| 0x01： $APR_{DIV} = 3$ | 0x0F： $APR_{DIV} = 17$ |
| ...                   | ...                    |
| 0x06： $APR_{DIV} = 8$ | 0x3E： $APR_{DIV} = 64$ |

**注：** このレジスタは、そのリセット値 (0x3F) を保持してはいけません。

## 11.4.3 タイムベース選択レジスタ (AWU\_TBR)

アドレスオフセット : 0x02

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み				AWUTB[3:0]			
				rw			

ビット 7:4 予約済み

ビット 3:0 **AWUTB[3:0]** : 自動ウェイクアップのタイムベース選択

これらのビットは、AWU 割り込み間のタイムインターバルを定義するために、ソフトウェアによって書き込まれます。AWUEN = 1 のとき、AWU 割り込みは有効です。

**0000** : 割り込みは発生していません。

<b>0001</b> : $APR_{DIV}/f_{LS}$	<b>0010</b> : $2 \times APR_{DIV}/f_{LS}$	<b>0011</b> : $2^2 APR_{DIV}/f_{LS}$
<b>0100</b> : $2^3 APR_{DIV}/f_{LS}$	<b>0101</b> : $2^4 APR_{DIV}/f_{LS}$	<b>0110</b> : $2^5 APR_{DIV}/f_{LS}$
<b>0111</b> : $2^6 APR_{DIV}/f_{LS}$	<b>1000</b> : $2^7 APR_{DIV}/f_{LS}$	<b>1001</b> : $2^8 APR_{DIV}/f_{LS}$
<b>1010</b> : $2^9 APR_{DIV}/f_{LS}$	<b>1011</b> : $2^{10} APR_{DIV}/f_{LS}$	<b>1100</b> : $2^{11} APR_{DIV}/f_{LS}$
<b>1101</b> : $2^{12} APR_{DIV}/f_{LS}$	<b>1110</b> : $5 \times 2^{11} APR_{DIV}/f_{LS}$	<b>1111</b> : $30 \times 2^{11} APR_{DIV}/f_{LS}$

## 11.4.4 AWU レジスタマップとリセット値

表 23. AWUレジスタマップ

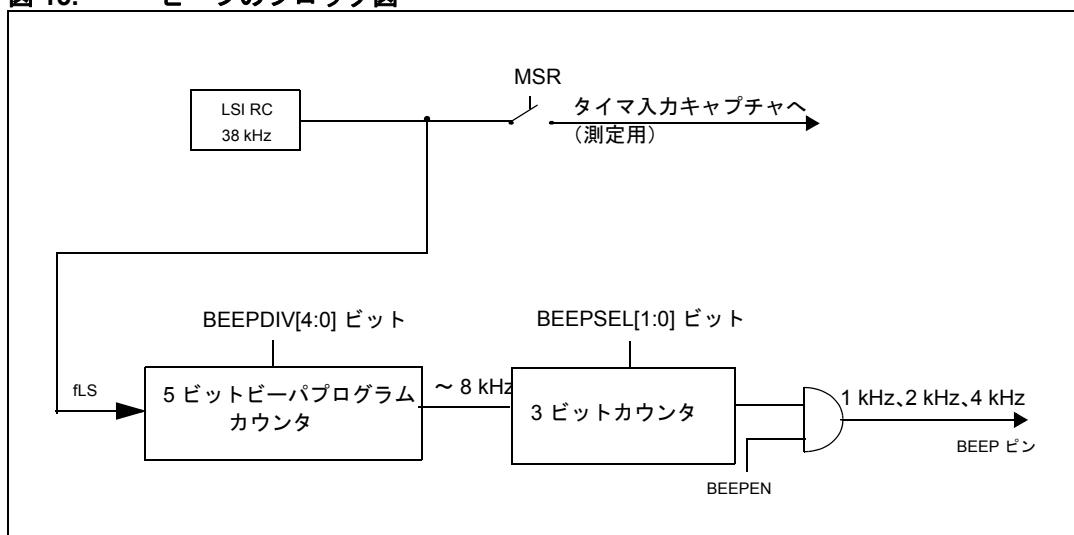
アドレス オフ セット	レジスタ名	7	6	5	4	3	2	1	0
0x00	AWU_CSR	- 0	- 0	AWUF 0	AWUEN 0	- 0	- 0	- 0	MSR 0
0x01	AWU_APR	- 0	- 0	APR5 1	APR4 1	APR3 1	APR2 1	APR1 1	APR0 1
0x02	AWU_TBR	- 0	- 0	- 0	- 0	AWUTB3 0	AWUTB2 0	AWUTB1 0	AWUTB0 0

## 12 ビーパ (BEEP)

### 12.1 概要

この機能は、LS クロックが 38 kHz の周波数で動作しているとき、1、2、または 4 kHz の範囲でビーパ信号を生成します。

図 15. ビープのブロック図



### 12.2 ビーパ機能の説明

#### 12.2.1 ビーパ動作

ビーパ機能を使用するには、以下の手順を順番に実行してください。

1. [セクション 12.2.2 : ビーパ校正](#)で説明されているように LS クロック周波数を校正して BEEPDIV[4:0] 値を定義します。
2. [ビーパ制御/ステータスレジスタ \(BEEP\\_CSR\)](#)の BEEPSEL[1:0] ビットに書き込むことによって、1 kHz、2 kHz、または 4 kHz の出力周波数を選択します。
3. [ビーパ制御/ステータスレジスタ \(BEEP\\_CSR\)](#)の BEEPEN ビットをセットして、LS クロックソースを有効にします。

**注：** プリスケアラカウンタがカウントを開始するのは、BEEPDIV[4:0] の値がそのリセット値 (0x1F) とは異なる場合のみです。

1. LSI クロック周波数を測定します（上の [セクション 11.3.3: LSI クロック周波数の測定](#) を参照）。
2. BEEP<sub>DIV</sub> 値を次のように計算します。ここで、A と x は f<sub>LS</sub>/8 (kHz 単位) の整数部と小数部です。  
x が A/(1+2\*A) 以下の場合は BEEP<sub>DIV</sub> = A-2、そうでない場合は  
BEEP<sub>DIV</sub> = A-1。
3. 結果として得られた BEEP<sub>DIV</sub> 値を [ブルーパ制御／ステータスレジスタ \(BEEP\\_CSR\)](#) の BEEPDIV[4:0] ビットに書き込みます。



## 13 ウィンドウ型ウォッチドッグ (WWDG)

### 13.1 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更新しない限り、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）が更新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）の間にカウンタが更新されなければならないことを意味します。

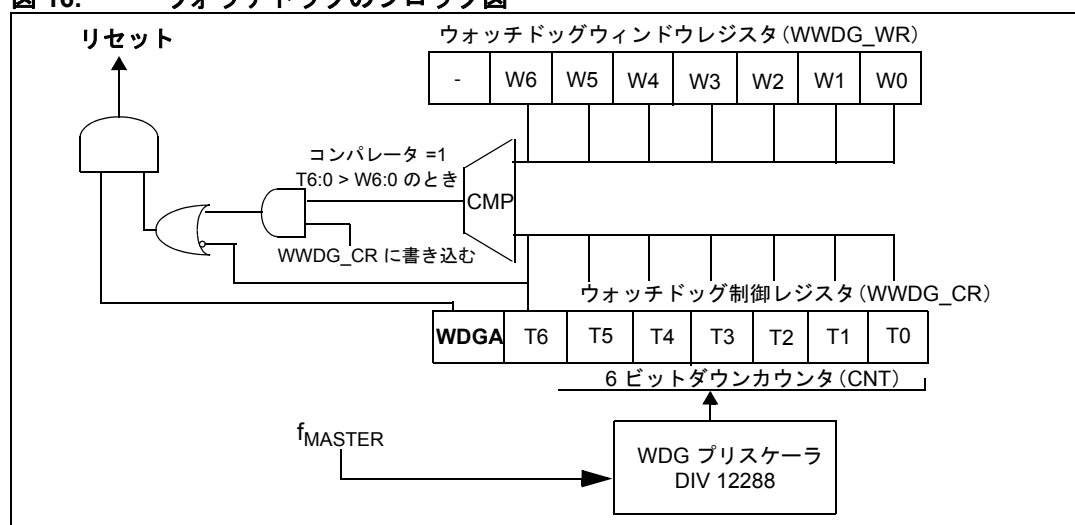
### 13.2 WWDG の主な機能

- 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 17 を参照）。
- ハードウェア/ソフトウェアウォッチドッグの起動（オプションバイトで選択可能）。
- HALT 命令によるオプションのリセット（オプションバイトで設定可能）。

### 13.3 WWDG 機能の説明

ウォッチドッグが有効であり（WDGA ビットがセット）、7 ビットのダウンカウンタ（T[6:0] ビット）が 0x40 から 0x3F にロールオーバーすると（T6 がクリア）、リセットサイクルを開始して、リセットピンをローに引き下げます。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リフレッシュした場合にも、リセットが生成されます。

図 16. ウォッチドッグのブロック図



アプリケーションプログラムは、通常動作時には定期的に WWDG\_CR レジスタへの書き込みを行って、MCU リセットを防ぐ必要があります。この操作は、カウンタの値がウィンドウレジスタの値より小さいときに限られます。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。

- ウォッチドッグを有効にするには：

ソフトウェアウォッチドッグが選択されると（オプションバイトによって）、ウォッチドッグはリセット後に無効になります。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。

ハードウェアウォッチドッグが選択されると（オプションバイトによって）、ウォッチドッグは常にアクティブであり、WDGA ビットは使用されません。

- ダウンカウンタを制御するには：

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、リセットの生成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。WWDG\_CR レジスタに書き込むときのプリスケアラの状態は不明なので、タイミングは最小値から最大値までさまざまです（図 17 を参照）。

ウィンドウレジスタ（WWDG\_WR）はウィンドウの上限値を含みます。リセットを防ぐには、ダウンカウンタの再ロードは、カウンタの値がウィンドウレジスタの値より小さく、0x3F より大きいときに行わなければなりません。図 17 に、ウィンドウ型ウォッチドッグ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ます）。

- ホールトオプションによるウォッチドッグのリセット  
ウォッチドッグが起動され、ホールトオプションによるウォッチドッグのリセットが選択された場合、HALT 命令によってリセットが生成されます。

## 13.4 ウォッチドッグタイムアウトをプログラムするには

次の式を使用して、WWDG タイムアウト  $t_{\text{WWDG}}$ （ms 単位）を計算できます。

$$t_{\text{WWDG}} = T_{\text{fMASTER}} 12288 \times (T[5:0] + 1)$$

ここで、 $T_{\text{fMASTER}}$  は、ペリフェラルのクロック周期です（単位：ms）。

**警告：** ただちにリセットされるのを防ぐために、WWDG\_CR レジスタに書き込むときには、常に T6 ビットに 1 を書き込んでください。

図 17. ウィンドウ型ウォッチドッグのタイミング図

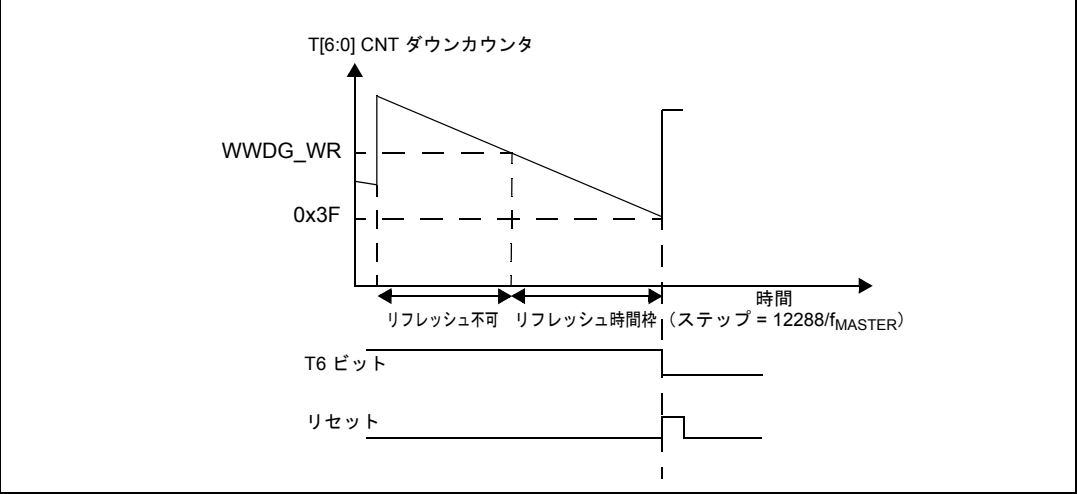


表 25. ウィンドウ型ウォッチドッグのタイミング例

T[6:0]	f <sub>MASTER</sub> (MHz)	
	2	16
40h	6.144	0.768
7Fh	393.216	49.152

## 13.5 WWDG 低電力モード

表 26. 低電力モードが WWDG に与える影響

モード	説明
WAIT	ウォッチドッグに影響なし：ダウンカウンタはデクリメントを続行します。
HALT	オプションバイトの WWDG_HALT
	0 ウォッチドッグリセットは生成されません。MCU は HALT モードに入ります。ウォッチドッグカウンタは、デクリメントされた後、カウントを停止し、MCU が外部割り込みまたはリセットを受け取るまで、ウォッチドッグリセットを生成できなくなります。 割り込みを受信した場合（HALT モードで発生する可能性のある割り込みについては、割り込みテーブルマッピングを参照）、ウォッチドッグは安定化遅延後にカウントを再スタートします。リセットが生成された場合、オプションバイトによってハードウェアウォッチドッグが選択されていない限り、ウォッチドッグは無効になります（リセット状態）。アプリケーションの推奨事項については、下記の <a href="#">セクション 13.7</a> を参照してください。
	1 HALT モードに入る代わりに、リセットが生成されます。
アクティブ HALT	x リセットは生成されません。MCU はアクティブ HALT モードに入ります。ウォッチドッグカウンタは、デクリメントされません。カウントを停止します。MCU がオシレータ割り込みまたは外部割り込みを受け取ると、ウォッチドッグはただちにカウントを再スタートします。MCU がリセットを受け取ると、ウォッチドッグは安定化遅延後にカウントを再スタートします。

## 13.6 ハードウェアウォッチドッグオプション

ハードウェアウォッチドッグがオプションバイトによって選択された場合、ウォッチドッグは常にアクティブであり、WWDG\_CR レジスタの WDGA ビットは使用されません。データシートのオプションバイトの説明を参照してください。

## 13.7 WWDG 割り込み

なし

## 13.8 WWDG レジスタ

### 13.8.1 制御レジスタ (WWDG\_CR)

アドレスオフセット : 0x00  
リセット値 : 0x7F

7	6	5	4	3	2	1	0
WDGA	T6	T5	T4	T3	T2	T1	T0
rw	rw	rw	rw	rw	rw	rw	rw

ビット 7 **WDGA** : 有効化ビット<sup>(1)</sup>

このビットは、ソフトウェアによってセットされ、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。12288 f<sub>MASTER</sub> サイクル (およそ) ごとにデクリメントされます。0x40 から 0x3F にロールオーバーすると (T6 がクリアされると)、リセットが生成されます。

1. このビットは、ハードウェアウォッチドッグオプションがオプションバイトによって有効にされた場合には使用されません。

### 13.8.2 ウィンドウレジスタ (WWDG\_WR)

アドレスオフセット : 0x01  
リセット値 : 0x7F

7	6	5	4	3	2	1	0
予約済み	W6	W5	W4	W3	W2	W1	W0
	rw	rw	rw	rw	rw	rw	rw

ビット 7 予約済み

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。

## 13.9 ウィンドウ型ウォッチドッグレジスタマップとリセット値

表 27. WWDG レジスタマップとリセット値

アドレス オフ セット	レジスタ名	7	6	5	4	3	2	1	0
0x00	WWDG_CR リセット値	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1
0x01	WWDG_WR リセット値	- 0	W6 1	W5 1	W4 1	W3 1	W2 1	W1 1	W0 1

## 14 独立型ウォッチドッグ (IWDG)

### 14.1 概要

独立型ウォッチドッグペリフェラルは、ハードウェアまたはソフトウェア障害によるプロセッサの誤動作を解決するために使用できます。38 kHz の LSI 内部 RC クロックソース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発生しても動作を継続します。

### 14.2 IWDG 機能の説明

図 18 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。

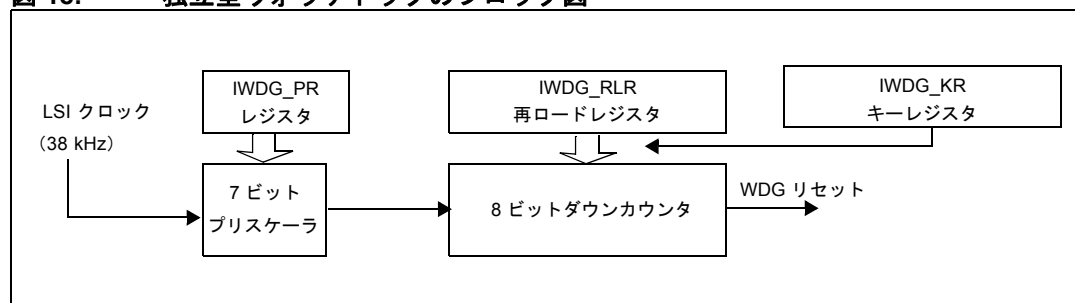
キーレジスタ (IWDG\_KR) に値 0xCC が書き込まれることによって独立型ウォッチドッグが開始すると、カウンタはリセット値 0xFF からカウントダウンを開始します。カウント値の終わり (0x00) に達すると、リセット信号が生成されます (IWDG RESET)。

有効にした後、独立型ウォッチドッグは、IWDG\_PR および IWDG\_RLR レジスタを通じて設定できます。IWDG\_PR レジスタは、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ために使用されます。IWDG\_KR レジスタに KEY\_REFRESH 値 (0xAA) が書き込まれると、IWDG\_RLR の値をカウンタに再ロードすることによって IWDG がリフレッシュされ、ウォッチドッグのリセットが防止されます。

IWDG\_PR および IWDG\_RLR レジスタは書き込み保護されます。これらを変更するには、まず、IWDG\_KR レジスタに KEY\_ACCESS コード (0x55) を書き込みます。このシーケンスは、IWDG\_KR レジスタに 0xAA を書き込んで更新することによって中止できます。

IWDG レジスタの詳細については、[セクション 14.3 : IWDG レジスタ](#) を参照してください。

図 18. 独立型ウォッチドッグのブロック図



#### ハードウェアウォッチドッグ機能

IWDG\_HW オプションバイトを通じてハードウェアウォッチドッグ機能が有効になった場合、ウォッチドッグは電源投入時に自動的に有効になり、カウンタが最終値に達する前にソフトウェアによってキーレジスタへの書き込みが行われな限り、リセットを生成します。データシートのオプションバイトの説明を参照してください。

## タイムアウト時間

最大タイムアウト時間は、IWDG\_PR および IWDG\_RLR レジスタを通じて設定でき、次の式で与えられます。

$$T = T_{LSI} \times P \times R$$

ここで、

$T$  = 最大タイムアウト時間

$T_{LSI} = 1/f_{LSI}$

$P = 2^{(PR[2:0] + 2)}$

$R = RLR[7:0] + 1$

IWDG カウンタは、このタイムアウト時間が満了する前に、ソフトウェアによってリフレッシュされる必要があります。そうしないと、最後のリフレッシュ操作から次の遅延が経過した後、IWDG リセットが生成されます。

$$D = T + 3 \times T_{LSI}$$

ここで、 $D$  = 最後のリフレッシュ操作と IWDG リセットの間の遅延。

表 28. IWDG タイムアウト時間の最小値／最大値 (LSI クロック周波数 = 38 kHz)

プリスケアラ分周回路	PR[2:0] ビット	タイムアウト (ms)	
		RL[7:0]= 0x00	RL[7:0]= 0xFF
/4	0	0.11	26.95
/8	1	0.21	53.89
/16	2	0.42	107.79
/32	3	0.84	215.58
/64	4	1.68	431.16
/128	5	3.37	862.32
/256	6	6.74	1724.63

## HALT／アクティブ HALT モードでの IWDG の使用

IWDG は、IWDG\_HALT オプションバイトの設定に応じて、HALT またはアクティブ HALT モードでも動作を続行できます。この場合、これらのモードのいずれかからデバイスをウェイクアップできます。詳細については、データシートのオプションバイトの説明を参照してください。

**注：** 予期しない IWDG のリセットを避けるために、アプリケーションは、HALT 命令を実行する前に、IWDG のタイムアウトを正しく設定し、IWDG カウンタを更新しておく必要があります。

## 14.3 IWDG レジスタ

### 14.3.1 キーレジスタ (IWDG\_KR)

アドレスオフセット : 0x00

リセット値 : 0xxx

7	6	5	4	3	2	1	0
KEY[7:0]							
w							

ビット 7:0 **KEY[7:0]** : キー値

KEY\_REFRESH の値は、ソフトウェアによって一定間隔で書き込まなければなりません。そうしないと、カウンタが 0 に達したときにウォッチドッグが MCU リセットを生成します。

IWDG がオプションバイトによって有効にされていない場合（データシートのオプションバイトの説明を参照）、KEY\_ENABLE の値がこのレジスタに書き込まれる最初の値になります。

**KEY\_ENABLE** 値 = 0xCC

KEY\_ENABLE の値が書き込まれると、IWDG が開始します。

**KEY\_REFRESH** 値 = 0xAA

KEY\_ENABLE の値が書き込まれると、IWDG がリフレッシュされます。

**KEY\_ACCESS** 値 = 0x55

KEY\_ENABLE の値が書き込まれると、保護されている 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 14.2](#) を参照）。

### 14.3.2 プリスケアラレジスタ (IWDG\_PR)

アドレスオフセット : 0x01

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						PR[2:0]	
						rw	

ビット 7:3 予約済み

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、書き込みアクセス保護がなされています（[セクション 14.2](#) 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ために、ソフトウェアによって書き込まれます。

- 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 : 予約済み



### 14.3.3 再ロードレジスタ (IWDG\_RLR)

アドレスオフセット : 0x02

リセット値 : 0xFF

7	6	5	4	3	2	1	0
RL[7:0]							
rw							

ビット 7:0 **RL[7:0]** : ウォッチドッグカウンタ再ロード値

これらのビットは、書き込みアクセス保護がなされています ([セクション 14.2](#) を参照)。IWDG\_KR レジスタに値 0xAA 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、ソフトウェアによって書き込まれます。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 28 を参照してください。

### 14.3.4 IWDG レジスタマップとリセット値

表 29. IWDG レジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	IWDG_KR リセット値	KEY7 x	KEY6 x	KEY5 x	KEY4 x	KEY3 x	KEY2 x	KEY1 x	KEY0 x
0x01	IWDG_PR リセット値	- 0	- 0	- 0	- 0	- 0	PR2 0	PR1 0	PR0 0
0x02	IWDG_RLR リセット値	RL7 1	RL6 1	RL5 1	RL4 1	RL3 1	RL2 1	RL1 1	RL0 1

15 タイマの概要

本マイクロコントローラには 2 種類のタイマがあります。すなわち、2 つの汎用タイマ (TIM2/TIM3) と、1 つの基本タイマ (TIM4) です。これらは、異なる機能を備えていますが、共通のアーキテクチャに基づいています。このため、さまざまなタイマを使用するアプリケーションを簡単に設計できます (同じレジスタ配置、共通の基本機能)。

これらのタイマはリソースを共有しませんが、互いにリンクして同期することができます。

このセクションでは、さまざまなタイマの機能の比較と、タイマ内部の信号名の用語を示します。

[セクション 16 : 16 ビット汎用タイマ \(TIM2/TIM3\)](#) では、タイマのさまざまなモードについて、詳細に説明します。

表 30. タイマの特性

記号	パラメータ	最小値	代表値	最大値	単位
$t_{w(ICAP)in}$	入力キャプチャパルス時間	2			$t_{MASTER}$
$t_{res(TIM)}$	タイマの時間分解能	1			$t_{MASTER}$
$Res_{TIM}$	16 ビットカウンタを使用したタイマの時間分解能		16		ビット
	8 ビットカウンタを使用したタイマの時間分解能		8		ビット
$t_{COUNTER}$	内部クロックが選択されたときのカウンタクロック周期		1		$t_{MASTER}$
$t_{MAX\_COUNT}$	16 ビットカウンタを使用したタイマの最大カウント数			65,536	$t_{MASTER}$
	8 ビットカウンタを使用したタイマの最大カウント数			256	$t_{MASTER}$

15.1 タイマ機能の比較

表 31. タイマ機能の比較

タイマ	カウンタ分解能	カウンタタイプ	プリスケアラ係数	キャプチャ/比較チャンネル	相補出力	繰り返しカウンタ	外部トリガ入力	外部ブレーク入力	タイマの同期/連結
TIM2 と TIM3 (汎用タイマ)	16 ビット	アップ/ダウン	1 から 128 までの 2 のべき乗	2	なし	なし	1	1	あり
TIM4 (基本タイマ)	8 ビット	アップ	1 から 32768 までの 2 のべき乗	0			0	0	



## 15.2 タイマ信号名の用語

表 32. 内部タイマ信号の用語

内部信号名	説明	関連図
BI	ブレーク割り込み	図 19 : TIMx の一般的ブロック図 (101 ページ)
CC1I、CC1I、CC2I、CC3I、CC4I	キャプチャ／比較割り込み	
CK_CNT	カウンタクロック	図 23 : プリスケアラ = 2 で ARPE = 0 (ARR はプリロードされない) のときのカウンタの更新 (105 ページ)
CK_PSC	プリスケアラクロック	
CNT_EN	カウンタイネーブル	図 32 : TI2 外部クロックの接続例 (111 ページ)
CNT_INIT	カウンタ初期化	
ETR	TIMx_ETR ピンからの外部トリガ	図 34 : 外部トリガ入力ブロック (112 ページ)
ETRF	フィルタされた外部トリガ	
ETRP	プリスケールされた外部トリガ	図 12 : クロック構造 (61 ページ)
f <sub>MASTER</sub>	クロックコントローラ (CLK) からのタイマペリフェラルクロック	
IC <sub>i</sub> 、IC1、IC2	入力キャプチャ	図 51 : TIM 1 チャンネル 1 の入カステージ (125 ページ)
IC <sub>1</sub> PS、IC1PS、IC2PS	プリスケールされた入力キャプチャ	
ITRx : ITR1、ITR2、ITR3	他のタイマの TRGO に固定された内部トリガ入力	図 19 : TIMx の一般的ブロック図 (101 ページ)
MATCH1	比較一致	図 41 : トリガ／マスタモード選択ブロック (117 ページ) および セクション 16.7.2 : 制御レジスタ 2 (TIMx_CR2) (140 ページ)
OC <sub>i</sub> : OC1、OC2	タイマ出力チャネル	図 16.5.5 : 強制出力モード (128 ページ)
OC/REF : OC1REF、OC2REF	出力比較基準信号	
TGI	トリガ割り込み	図 30 : クロック／トリガコントローラのブロック図 (110 ページ)
TI <sub>i</sub> : TI1、TI2	タイマ入力	図 51 : TIM 1 チャンネル 1 の入カステージ (125 ページ) 図 51 : TIM 1 チャンネル 1 の入カステージ (125 ページ)
TI <sub>1</sub> F : TI1F、TI2F	フィルタされたタイマ入力	
TI1_ED	タイマ入力エッジ検出器	
TI <sub>1</sub> FPx : TI1FP1、TI1FP2、TI2FP1、TI2FP2	フィルタされ、プリスケールされたタイマ入力	
TRC	トリガキャプチャ	図 31 : 通常モードの制御回路、f <sub>MASTER</sub> を 1 分周 (111 ページ)
TRGI	クロック／トリガ／スレーブモードコントローラへのトリガ入力	
TRGO	他のタイマのトリガ入力 INTx に固定されたトリガ出力	図 19 : TIMx の一般的ブロック図 (101 ページ)
UEV	更新イベント	図 23 : プリスケアラ = 2 で ARPE = 0 (ARR はプリロードされない) のときのカウンタの更新 (105 ページ)
UIF	更新割り込み	

## 16 16ビット汎用タイマ (TIM2/TIM3)

### 16.1 概要

この章では、同じタイマである TIM2 と TIM3 (TIMx と呼ばれます) について説明します。

各汎用タイマ (TIMx) は、プログラマブルなプリスケアラによって駆動される 16 ビットのアップダウン自動再ロードカウンタを備えています。

ここでは、インデックス  $i$  は 1 または 2 であり、2 つのキャプチャ/比較チャンネルを表します。

これらのタイマは、以下のようなさまざまな目的で使用されます。

- タイムベースの生成
- 入力信号 (入力キャプチャ) のパルス長の測定
- 出力波形の形成 (出力比較、PWM、およびワンパルスモード)
- さまざまなイベント (キャプチャ、比較、オーバーフロー、ブレイク、トリガ) 時の割り込み機能
- 他のタイマまたは外部信号 (外部クロック、リセット、トリガ、およびイネーブル) との同期

これらのタイマは、センターアライン PWM 機能を必要とするものなど、広範囲な制御アプリケーションに最適です。

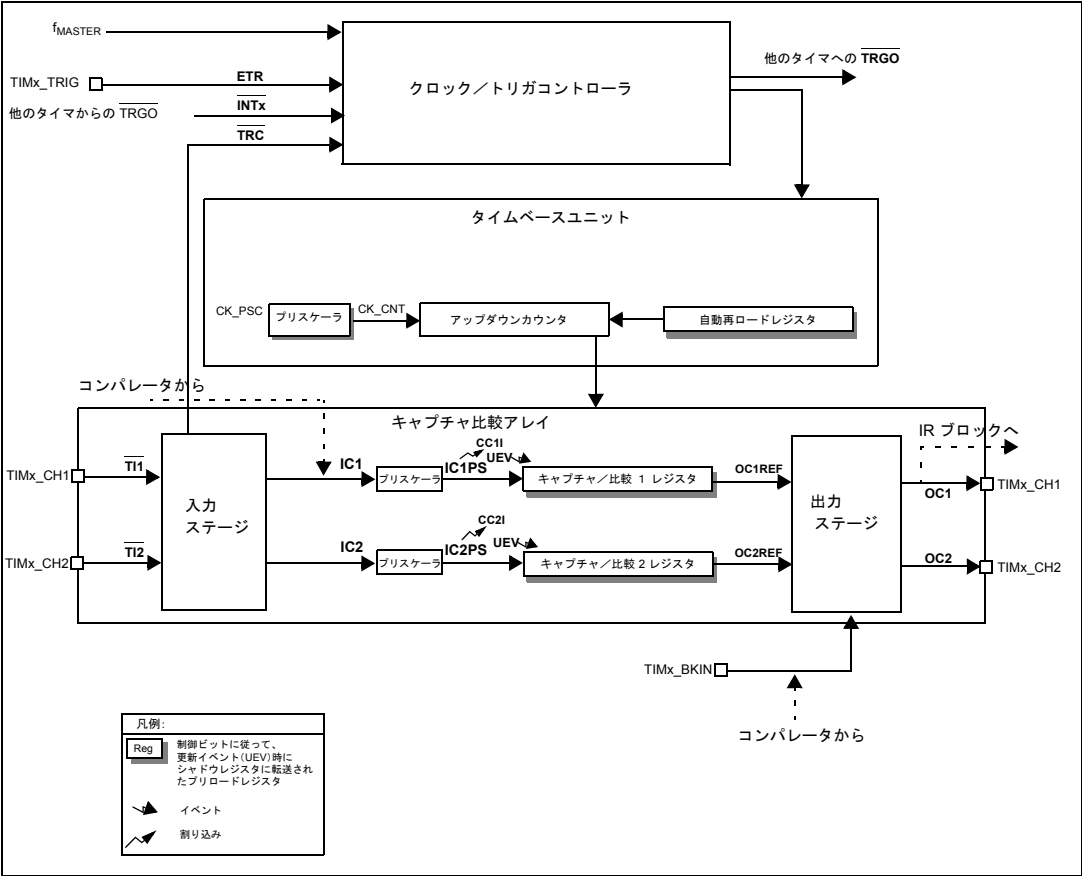
タイマクロックは、内部クロック または外部クロックから供給でき、設定レジスタで選択できます。

### 16.2 TIMx の主な機能

TIMx 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/ダウンカウンタ、自動再ロードカウンタ。
- 3 ビットのプログラマブルプリスケアラによって、カウンタクロック周波数を動作中に 1 から 128 までの任意の 2 のべき乗で分周できます。
- 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。
- 以下に対応する 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とセンターアラインモード)
  - ワンパルスモード出力
- 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イク入力。
- 以下のイベント時の割り込み生成：
  - 更新：カウンタオーバーフロー/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 (内部/外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
  - ブレイク入力

図 19. TIMx の一般的ブロック図

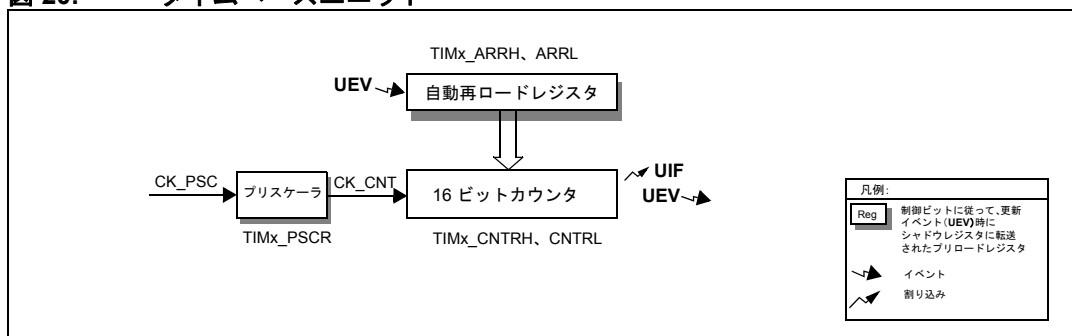


## 16.3 TIMx タイムベースユニット

このタイマは、以下を含むタイムベースユニットを備えています。

- 16ビットアップ/ダウンカウンタ
- 16ビット自動再ロードレジスタ
- プリスケアラ

図 20. タイムベースユニット



16ビットカウンタ、プリスケアラ、および自動再ロードレジスタは、ソフトウェアによる読み書きが可能です。

自動再ロードレジスタは、プリロードレジスタとシャドウレジスタで構成されています。

自動再ロードレジスタへの書き込みは、次の 2 つのモードで行うことができます。

- **自動再ロードのプリロード有効** (TIMx\_CR1 レジスタの ARPE ビットがセット) このモードでは、データが自動再ロードレジスタに書き込まれると、データはプリロードレジスタに保持され、次の更新イベント (UEV) 時にシャドウレジスタに転送されます。
- **自動再ロードのプリロード無効** (TIMx\_CR1 レジスタの ARPE ビットがクリア) このモードでは、データが自動再ロードレジスタに書き込まれると、データはただちにシャドウレジスタに転送されます。

更新イベントは、以下の場合に生成されます。

- カウンタオーバーフローまたはアンダーフロー時
- 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
- クロック/トリガコントローラからのトリガイイベントによって

プリロード有効で (ARPE=1)、更新イベントが発生した場合：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、プリスケアラのパッファにプリロード値 (TIMx\_PSCR レジスタの内容) が再ロードされます。

更新イベント (UEV) は、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。

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。

**注：** 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。

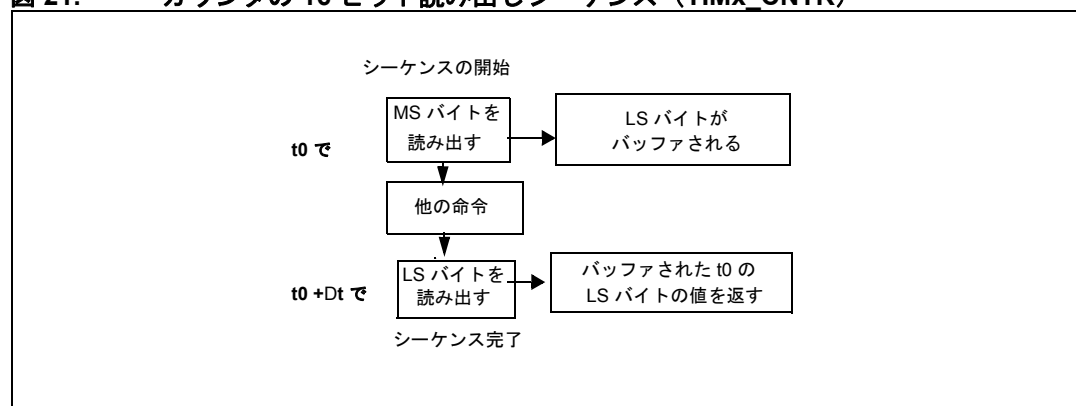
## 16.3.1 16 ビットカウンタの読み出しと書き込み

カウンタに書き込む場合、バッファリングは行われません。TIMx\_CNTRH と TIMx\_CNTRL はいつでも書き込みが可能なので、正しくない中間的内容のロードを避けるために、このカウンタの動作中には新しい値を書き込まないでください。

読み出しのために、8 ビットのバッファが実装されています。ユーザは、最初に MS バイトを読み出す必要があります。その後、LS バイトの値が自動的にバッファされます (図 21 を参照)。このバッファされた値は、16 ビット読み出しシーケンスが完了するまで、そのまま保持されます。

**注：** LDW 命令を使用して 16 ビットカウンタを読み出さないでください。この命令は、LS バイトを最初に読み出して、間違った結果を返します。

図 21. カウンタの 16 ビット読み出しシーケンス (TIMx\_CNTR)



## 16.3.2 16 ビット TIMx\_ARR レジスタの書き込みシーケンス

TIMx\_ARR レジスタには、プリロードレジスタを介して 16 ビットの値がロードされます。これは、2 回の書き込み命令 (各バイトについて 1 回ずつ) で行う必要があります。MS バイトが最初に書き込まなければなりません。

シャドウレジスタの更新は、MS バイトが書き込まれるとただちにブロックされ、LS バイトが書き込まれるまでブロックされたままです。LDW 命令は最初に LS バイトを書き込むため、この場合は正しくない結果になるので、使用しないでください。

## 16.3.3 プリスケアラ

プリスケアラは、3 ビットレジスタ (TIMx\_PSCr レジスタ) を通じて制御される 7 ビットのカウンタに基づきます。この制御レジスタはバッファされるので、動作中に変更できます。カウンタクロック周波数を 1、2、4、8、16、32、64、または 128 分周することができます。

カウンタクロック周波数は、次の式で計算されます。

$$f_{CK\_CNT} = f_{CK\_PSC} / 2^{(PSCr[2:0])}$$

新しいプリスケアラ値は、次の期間 (次のカウンタ更新イベントの後) で、有効になります。

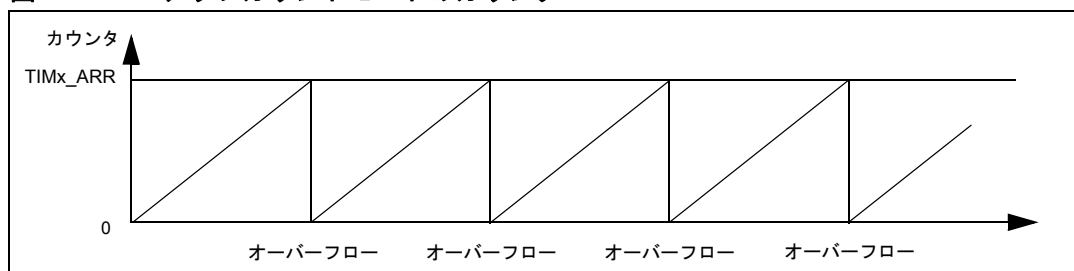


## 16.3.4 アップカウントモード

アップカウントモードでは、カウンタは、0 からユーザによって定義された比較値 (TIMx\_ARR レジスタの内容) までカウントします。その後、0 から再スタートして、カウンタオーバーフローイベントを生成し、TIMx\_CR1 レジスタの UDIS ビットが 0 の場合は更新イベント (UEV) を生成します。

図 22 に、このカウントモードの例を示します。

図 22. アップカウントモードのカウンタ



更新イベント (UEV) は、TIMx\_EGR レジスタの UG ビットをセットすることによって (ソフトウェアによって、または、トリガコントローラを使用して) 生成することもできます。

UEV イベントは、ソフトウェアで TIMx\_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。UDIS ビットに 0 が書き込まれるまで、UEV 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 からカウントを再スタートします (しかし、プリスケアラの分周比は変化しません)。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\_SR1 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。

プリスケアラのパッファにはプリロード値 (TIMx\_PSCR レジスタの内容) が再びロードされます。

以下の図は、TIMx\_ARR=36h 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 2 つ示します。

図 23 では、プリスケアラ分周回路は 2 にセットされているので、カウンタクロック (CK\_CNT) 周波数はプリスケアラクロックソース (CK\_PSC) の半分です。

図 23 では、自動再ロードのプリロードは無効 (ARPE=0) なので、シャドウレジスタはただちに変更され、アップカウントが 36h に達すると、カウンタオーバーフローが発生します。これによって、更新イベントが生成されます。

図 23. プリスケアラ = 2 で ARPE = 0 (ARR はプリロードされない) のときのカウンタの更新

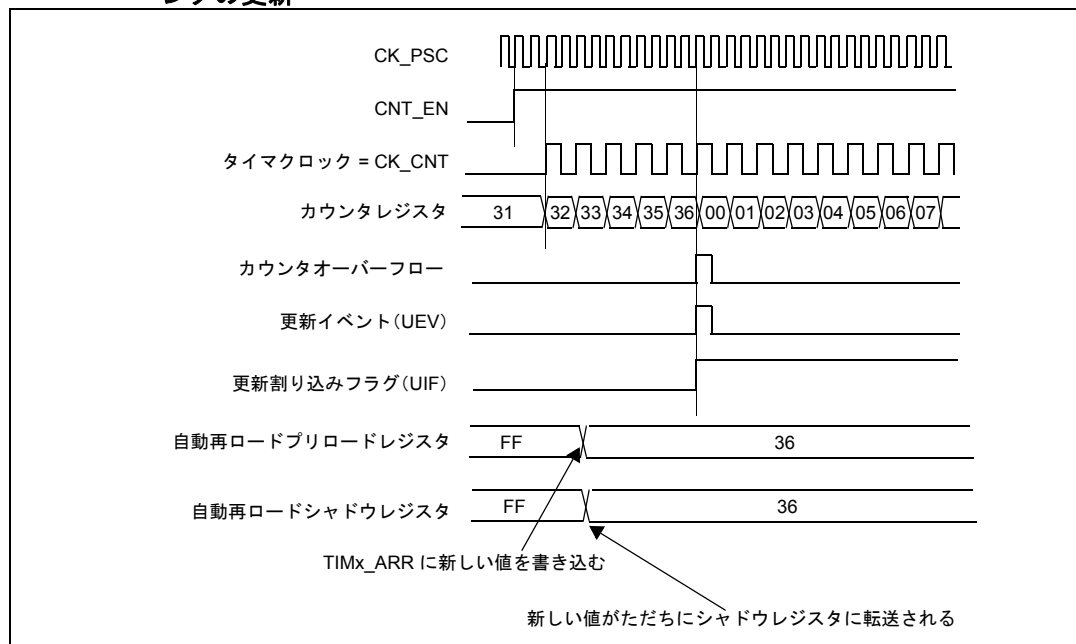
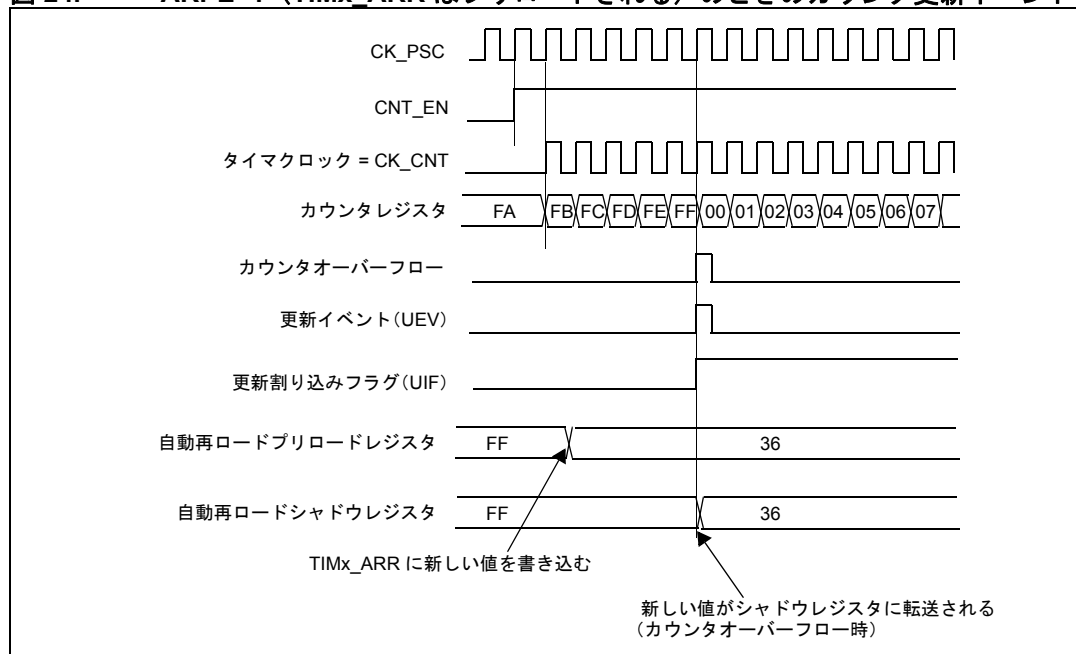


図 24 では、プリスケアラ分周回路は 1 にセットされているので、CK\_CNT は CK\_PSC と同じ周波数です。

図 24 では、自動再ロードのプリロードは有効 (ARPE=1) なので、次のカウンタオーバーフローは FFh で発生します。新しい自動再ロードレジスタの値 36h は、更新イベントを生成するオーバーフローの後、有効になります。

図 24. ARPE=1 (TIMx\_ARR はプリロードされる) のときのカウンタ更新イベント

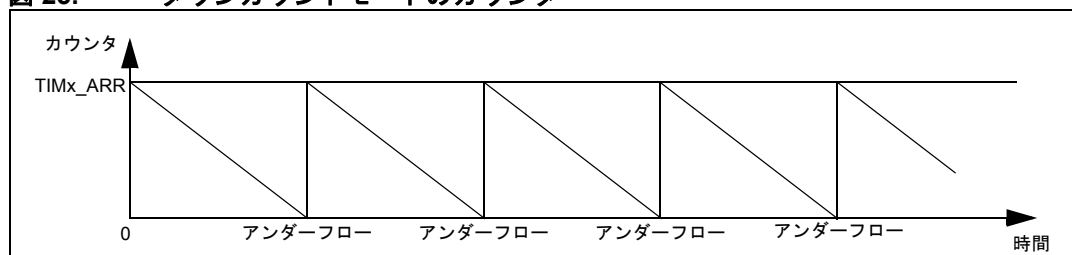


## 16.3.5 ダウンカウントモード

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 までカウントします。次にカウンタは自動再ロード値から再スタートして、TIMx\_CR1 レジスタの UDIS ビットが 0 の場合は、カウンタアンダーフローと更新イベント (UEV) を生成します。

図 25 に、このカウントモードの例を示します。

図 25. ダウンカウントモードのカウンタ



更新イベントは、TIMx\_EGR レジスタの UG ビットをセットすることによって (ソフトウェアによって、または、クロック/トリガモードコントローラを使用して) 生成することもできます。

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。UDIS ビットに 0 が書き込まれるまで、UEV 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再スタートしますが、プリスケアラのカウンタは 0 から再スタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx\_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込み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\_SR1 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\_PSCR 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 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。

次の図は、TIMx\_ARR=36h のときの、幾つかの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。

ダウンカウントモードでは、新しい値が次の周期で有効になるように、プリロードは一般的に使用されません (図 26 を参照)。

図 26. プリスケアラ = 2 で ARPE = 0 (ARR はプリロードされない) のときのカウンタの更新

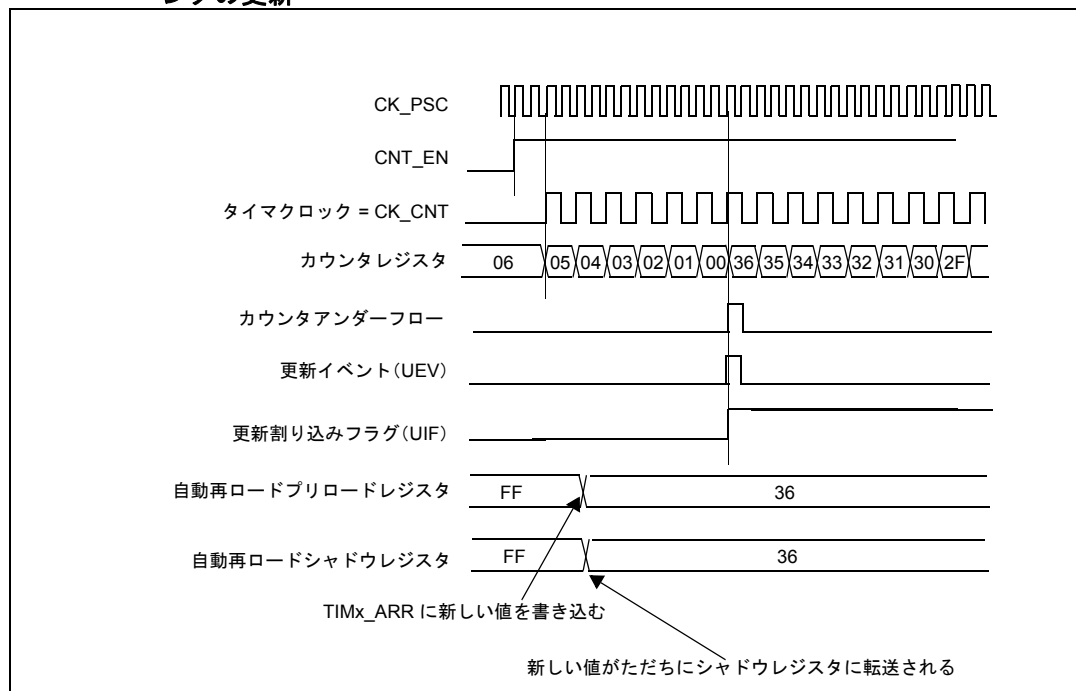
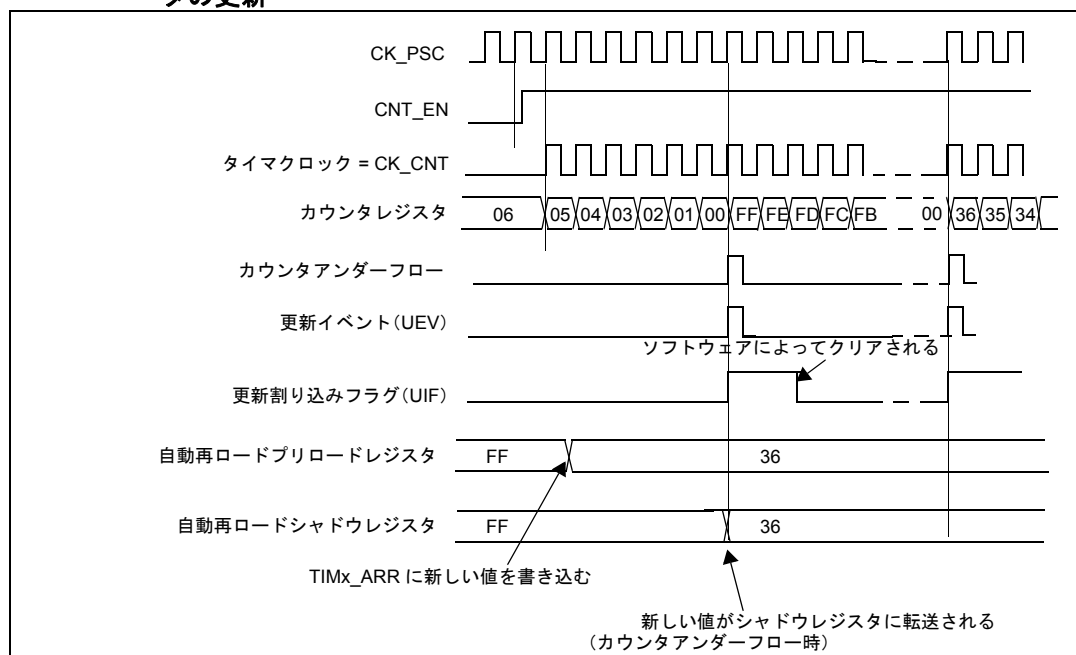


図 27. プリスケアラ = 1 で ARPE = 1 (ARR はプリロードされる) のときのカウンタの更新



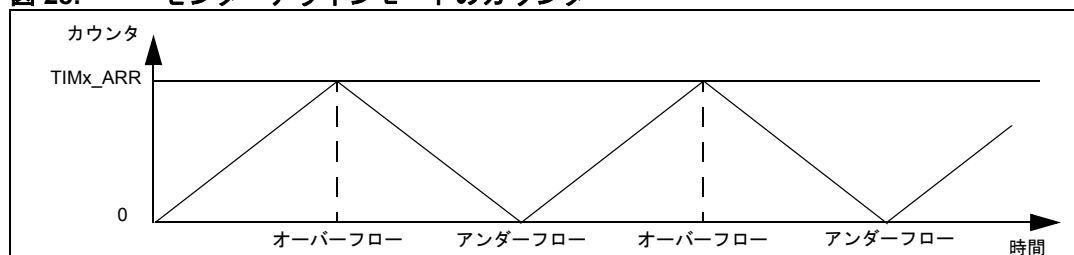
## 16.3.6 センターアラインモード (アップ/ダウンカウント)

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 までカウントを行い、カウンタオーバーフローイベントを生成します。次に 0 までのカウントダウンを行い、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 からカウントを再スタートします。

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。

図 28 に、このカウントモードの例を示します。

図 28. センターアラインモードのカウンタ



更新イベントは、カウンタオーバーフローごと、およびカウンタアンダーフローごとに生成されます。

TIMx\_EGR レジスタの UG ビットをセットすることでも (ソフトウェアによって、またはクロック/トリガモードコントローラを使用して)、更新イベントが生成されます。この場合、カウンタとプリスケアラのカウンタは 0 からカウントを再スタートします。

更新イベント (UEV) 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。UDIS ビットに 0 が書き込まれるまで、UEV 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。

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。

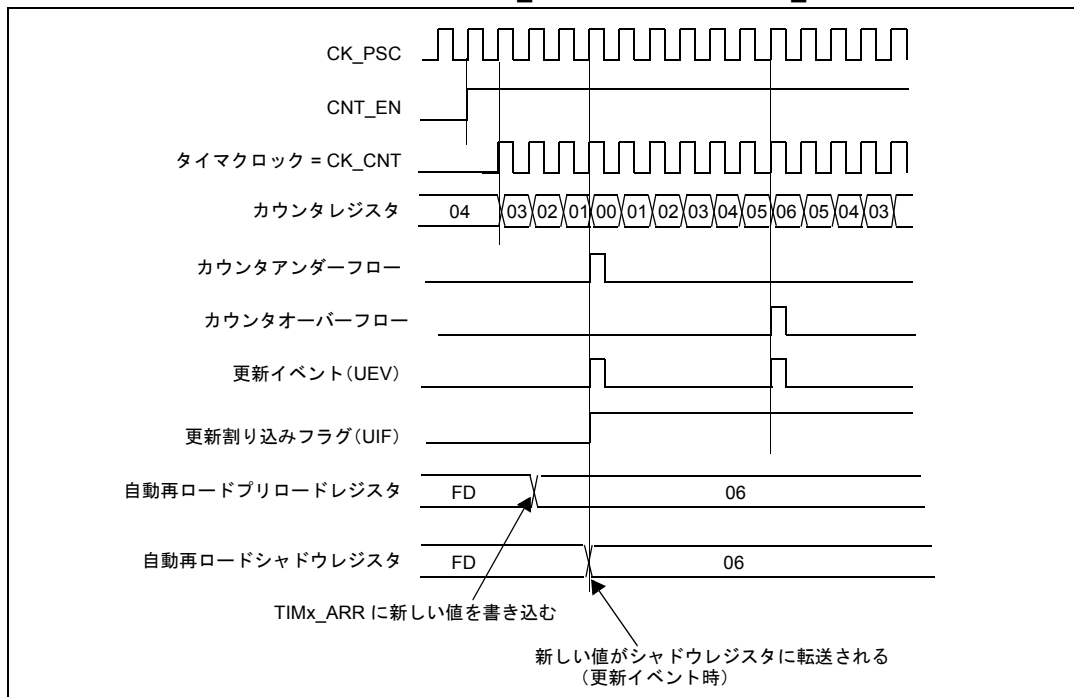
更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx\_SR1 レジスタの UIF ビット) がセットされます。

プリスケアラのバッファにはプリロード値 (TIMx\_PSCR 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 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、次の周期は予想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)。

次に、ある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 29. カウンタのタイミング図、CK\_PSC を 1 分周、TIMx\_ARR=06h、ARPE=1



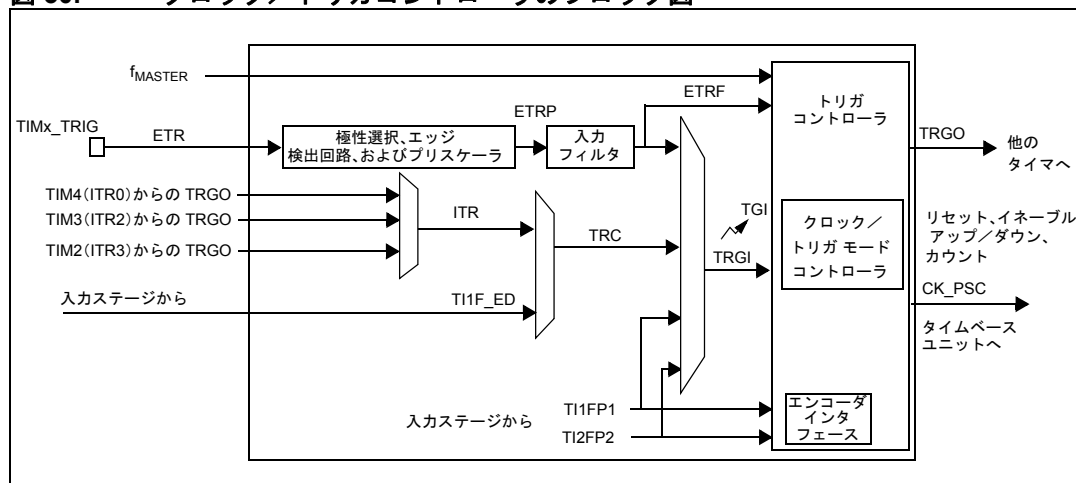
センターアラインモードの使用に関するヒント：

- センターアラインモードで開始するときには、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を開始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は、カウントアップを続けます。
  - － カウンタに 0 または TIMx\_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx\_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

## 16.4 TIMx クロック／トリガコントローラ

クロック／トリガコントローラによって、タイマクロックソース、入力トリガ、および出力トリガを設定できます。図 30. にブロック図を示します。

図 30. クロック／トリガコントローラのブロック図



### 16.4.1 プリスケーラクロック (CK\_PSC)

タイムベースユニットプリスケラクロック (CK\_PSC) は、次のクロックソースのいずれかから供給できます。

- 内部クロック ( $f_{MASTER}$ )
- 外部クロックモード 1 : 外部タイマ入力 (Tix)
- 外部クロックモード 2 : 外部トリガ入力 (ETR)
- 内部トリガ入力 (ITRx) : タイマを別のタイマのプリスケラとして使用。詳細については、[タイマを別のタイマのプリスケラとして使用する \(118 ページ\)](#) を参照してください。

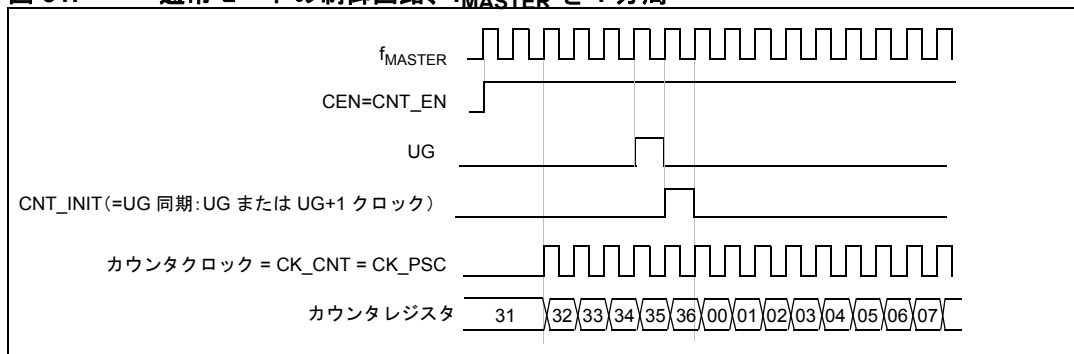
### 16.4.2 内部クロックソース

クロック／トリガモードコントローラと外部トリガ入力の両方とも無効の場合 (TIMx\_SMCR レジスタの SMS=0b000、TIMx\_ETR レジスタの ECE=0)、CEN、DIR、および UG ビットが実際の制御ビットであり、ソフトウェアによってのみ変更できます (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、プリスケラには内部クロックが供給されます。



図 31 は、プリスケアラを使用しない場合の通常モードの制御回路とアップカウンタの動作を示します。

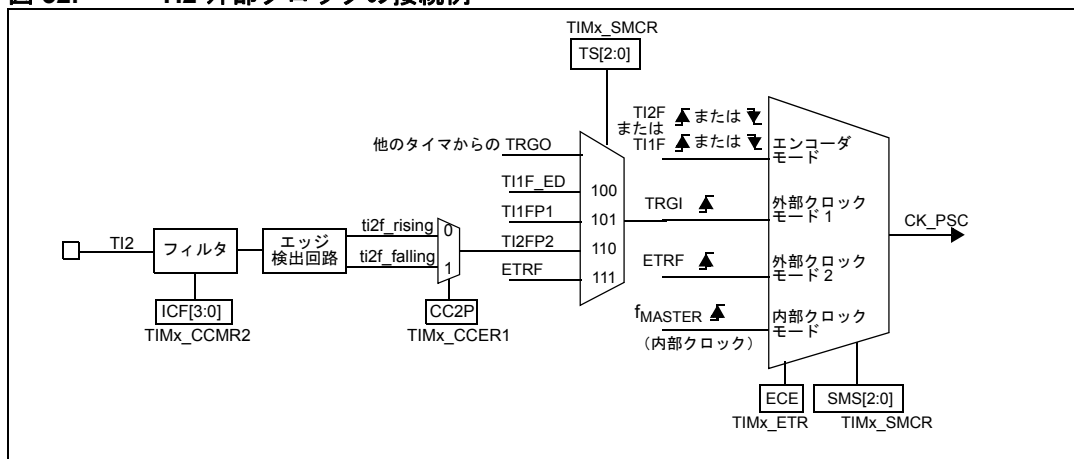
図 31. 通常モードの制御回路、 $f_{\text{MASTER}}$  を 1 分周



## 16.4.3 外部クロックソースモード 1

カウンタは、選択されたタイマ入力の立ち上がりまたは立ち下がりエッジでカウントできます。このモードは TIMx\_SMCR レジスタの SMS=0b111 のときに選択されます。

図 32. TI2 外部クロックの接続例



アップカウンタを設定して、たとえば、TI2 入力の立ち上がりエッジでカウントするように設定するには、以下の手順に従ってください。

1. TIMx\_CCMR2 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\_CCMR2 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)。

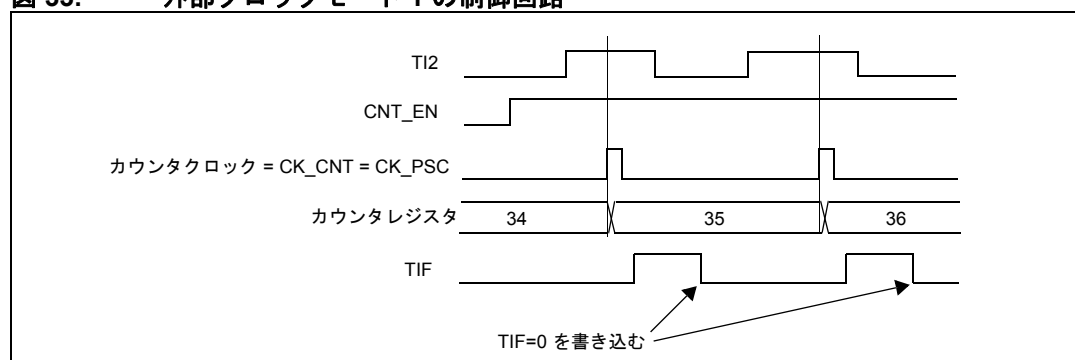
注： キャプチャプリスケラはトリガには使用されないで、設定は不要です。CC2S ビットは、入力キャプチャソースのみを選択するため、設定する必要はありません。

3. TIMx\_CCER1 レジスタに CC2P=0 を書き込むことによって、立ち上がりエッジの極性を選択します。
4. TIMx\_SMCR レジスタに SMS=0b111 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 に立ち上がりエッジが発生すると、カウンタはカウントを開始し、トリガフラグがセットされ (TIMx\_SR1 レジスタの TIF ビット)、有効な場合は割り込みリクエストが送信されます (TIMx\_IER レジスタの TIE ビットに応じて)。

TI2 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、TI2 入力の再同期回路によるものです。

図 33. 外部クロックモード 1 の制御回路

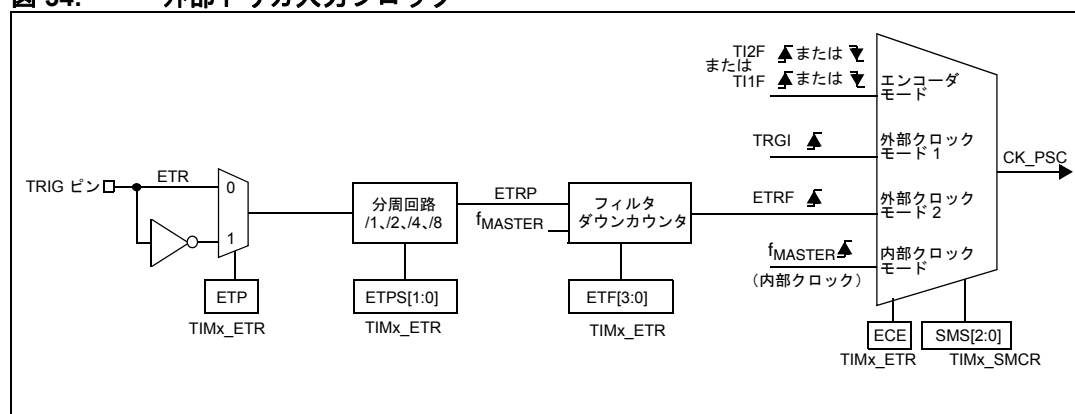


### 16.4.4 外部クロックソースモード 2

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でカウントできます。このモードは、TIMx\_ETR レジスタに ECE=1 を書き込むことによって選択されます。

図 34 に、外部トリガ入力ブロックの概要を示します。

図 34. 外部トリガ入力ブロック



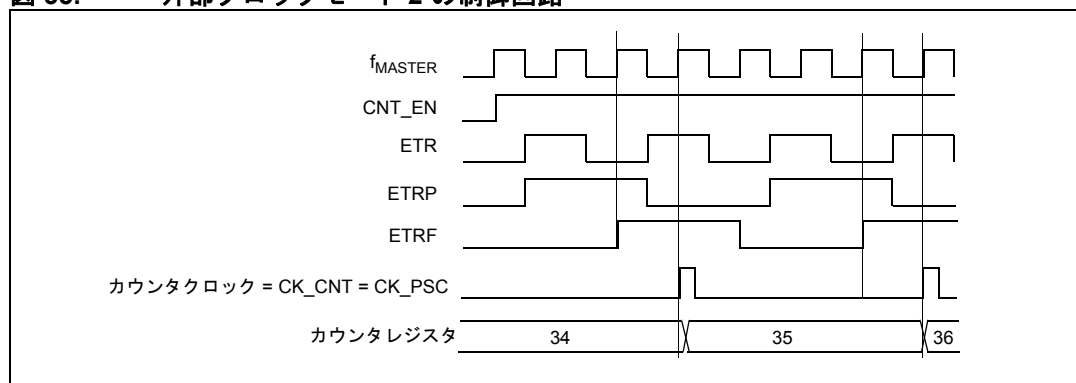
アップカウンタを設定して、たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするように設定するには、以下の手順に従ってください。

1. この例ではフィルタは不要なので、TIMx\_ETR レジスタに ETRF[3:0]=0b0000 を書き込みます。
2. TIMx\_ETR レジスタに ETPS[1:0]=0b01 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\_ETR レジスタに ETP=0 を書き込むことによって、TRIG ピンの立ち上がりエッジ検出を選択します。
4. TIM1\_ETR 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のリセットまでの遅延は、ETRP 信号の再同期回路によるものです。

図 35. 外部クロックモード 2 の制御回路



## 16.4.5 トリガの同期

トリガ入力には、次の 4 つのタイプがあります (表 32: 内部タイマ信号の用語 (99 ページ) を参照)。

- ETR
- TI1
- TI2
- 他のタイマからの TRGO

タイマは、外部トリガと 3 つのモード (トリガ標準モード、トリガリセットモード、およびトリガゲートモード) で同期できます。

### トリガ標準モード

選択された入力のイベントに対応して、カウンタが開始されます。

次の例では、TI2 入力の立ち上がりエッジに応じてアップカウンタが開始されます。

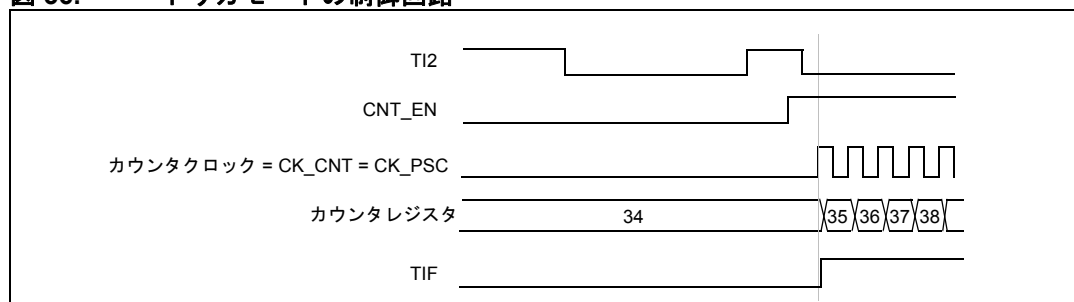
- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC2F=0b0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するので、設定も不要です。TIMx\_CCER1 レジスタに CC2P=0 を書き込んで、立ち上がりエッジの極性を選択します。

- TIMx\_SMCR レジスタに SMS=0b110 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=0b110 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リセットされるまでの遅延は、TI2 入力の再同期回路によるものです。

**図 36. トリガモードの制御回路**



### トリガリセットモード

カウンタとそのプリスケアラを、トリガ入力イベントに対応し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。

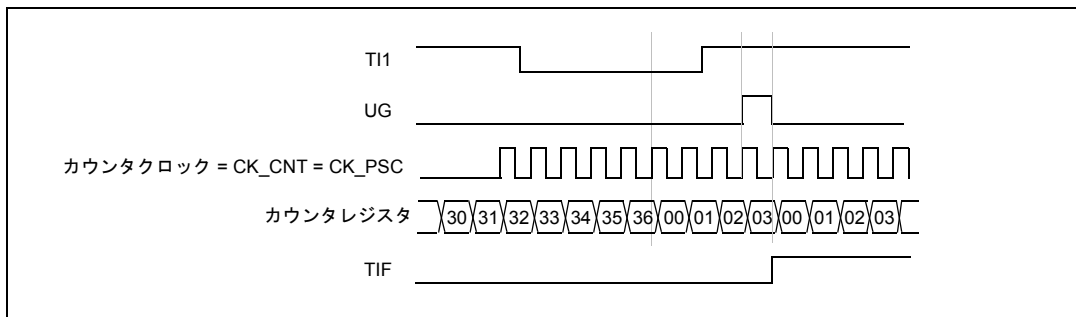
次の例では、TI1 入力の立ち上がりエッジに対応して、アップカウンタがクリアされます。

- 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=0b0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC1S ビットを入力キャプチャソースのみを選択するので、設定は不要です。TIMx\_CCER1 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)。
- 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=0b101 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 から再スタートします。同時に、トリガフラグがセットされ (TIMx\_SR1 レジスタの TIF ビット)、有効な場合は割り込みリクエストを送信できます (TIMx\_IER レジスタの TIE ビットに応じて)。

次の図は、自動再ロードレジスタ TIMx\_ARR=36h 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 37. トリガリセットモードの制御回路



## トリガゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

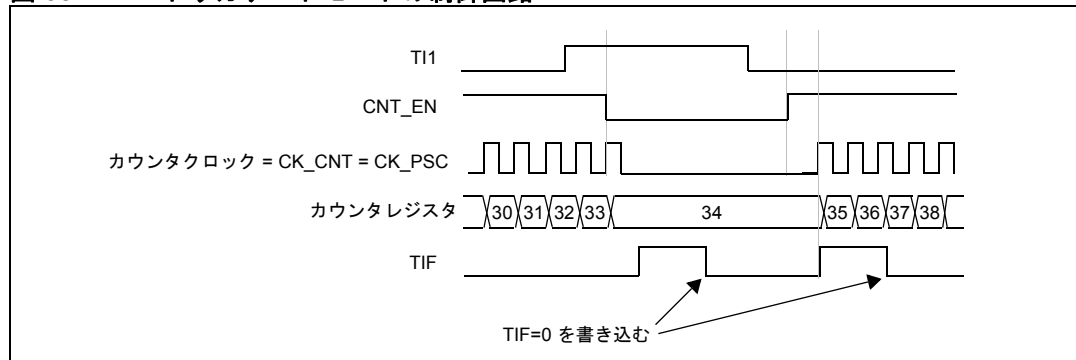
次の例では、アップカウンタは TI1 入力があるときにのみカウントします。

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=0b0000 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 ビットも入力キャプチャソースのみを選択するので、設定は不要です。TIMx\_CCER1 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)。
2. TIMx\_SMCR レジスタに SMS=0b101 を書き込むことによって、タイマをトリガゲートモードに設定します。TIMx\_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
3. TIMx\_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします (トリガゲートモードでは、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 がローである限り内部クロックでカウントを開始し、TI1 がハイになると停止します。カウンタの開始時と停止時に、TIF フラグがセットされます。

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 38. トリガゲートモードの制御回路



## トリガモードと外部クロックモード 2 の組み合わせ

外部クロックモード 2 は、別のトリガモードとともに使用できます。この場合、ETR を外部クロック入力として使用し、別の入力をトリガ入力として選択できます (トリガ標準モード、トリガリセットモード、またはトリガゲートモード)。ETR を TRGI として選択しないように注意してください (TIMx\_SMCR レジスタの TS ビットを通じて)。

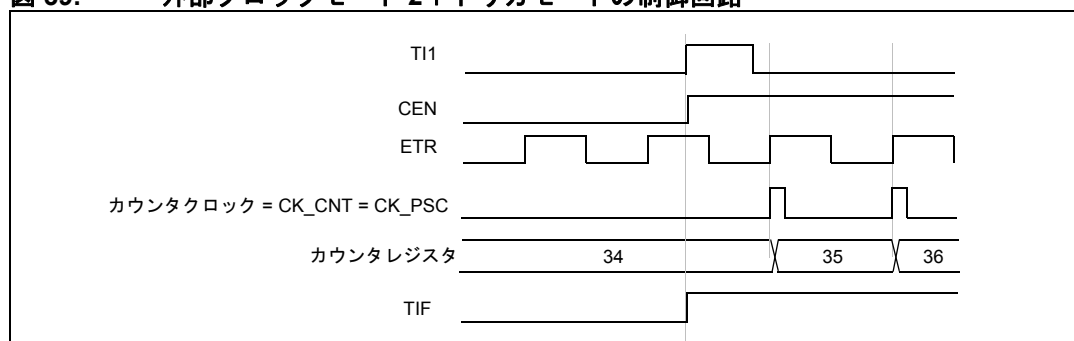
TI1 の立ち上がりエッジが発生すると同時に ETR の各立ち上がりエッジでアップカウンタを有効にするには、以下の手順に従ってください (外部 ETR クロックでの標準トリガモード)。

- TIMx\_ETR レジスタに書き込むことによって、外部トリガ入力回路を設定します。この例ではフィルタは不要なので、ETF=0b0000 を書き込みます。ETPS=00 を書き込んでプリスケアラを無効にし、ETP=0 を書き込んで ETR の立ち上がりエッジを検出し、ECE=1 を書き込んで外部クロックモード 2 を有効にします。
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。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=0b0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC1S ビットは入力キャプチャソースのみを選択するので、設定は不要です。TIMx\_CCER1 レジスタに CC1P=0 を書き込んで、立ち上がりエッジの極性を選択します。
- TIMx\_SMCR レジスタに SMS=0b110 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=0b101 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。結果として、カウンタは ETR の立ち上がりエッジでカウントします。

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。ETR の立ち上がりエッジから実際のカウンタのリセットまでの遅延は、ETRP 信号の再同期回路によるものです。

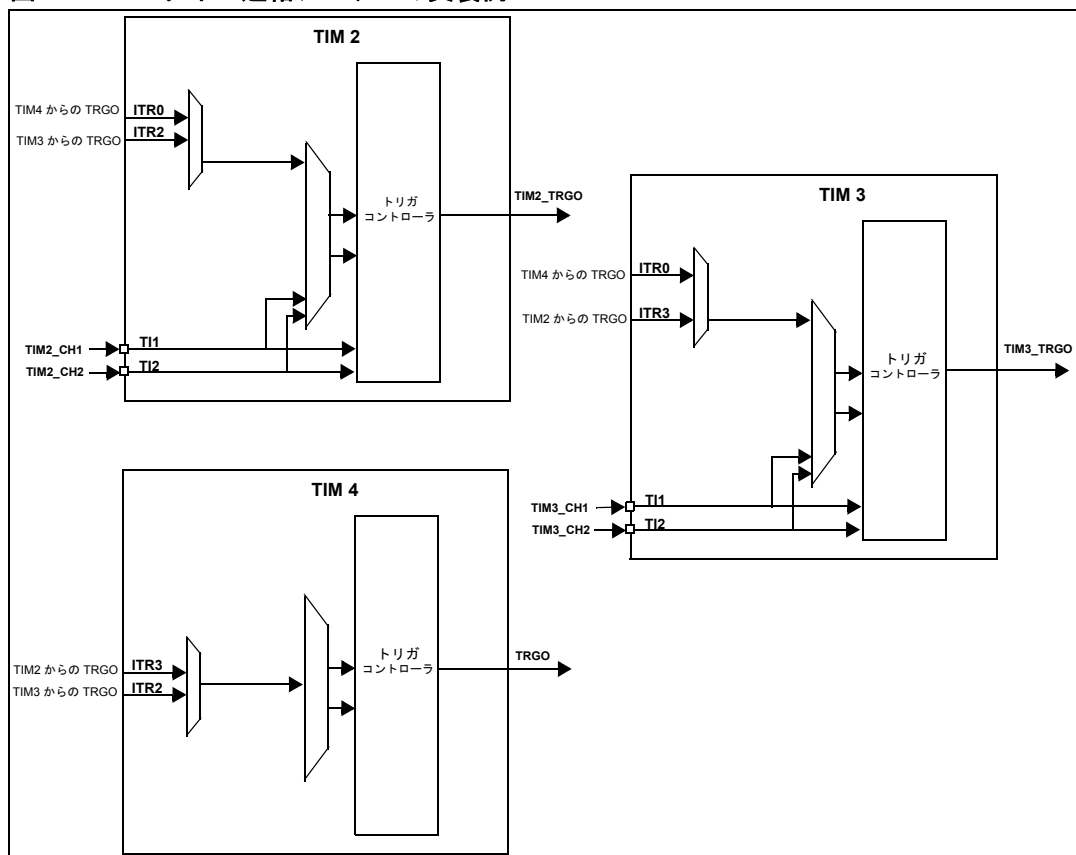
**図 39. 外部クロックモード 2+トリガモードの制御回路**



## 16.4.6 タイマ間の同期

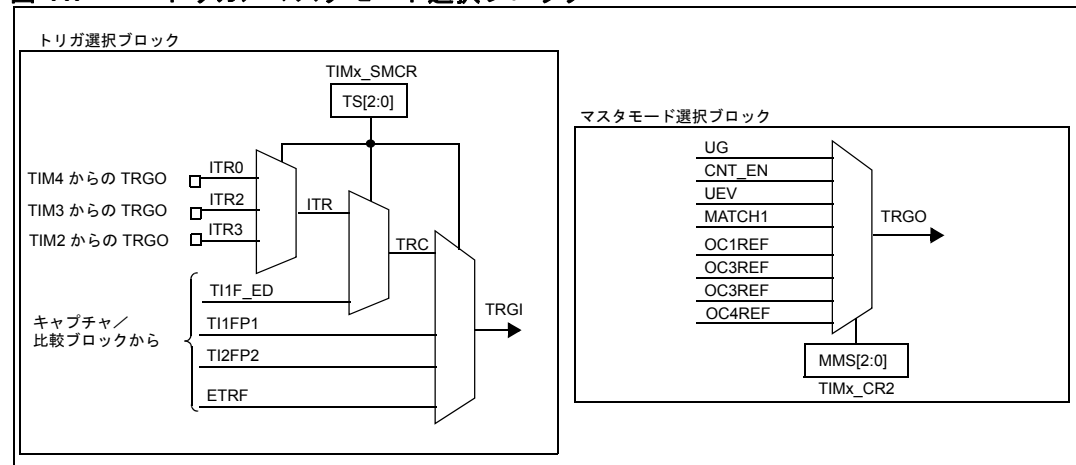
タイマは、タイマの同期や連結のために内部で相互リンクされます。あるタイマがマスタモードに設定されると、そのタイマはスレーブモードに設定された別のタイマのカウンタのリセット、開始、停止、またはクロック供給のためにトリガ (TRGO) を出力することができます。

図 40. タイマ連結システムの実装例



次の図に、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。

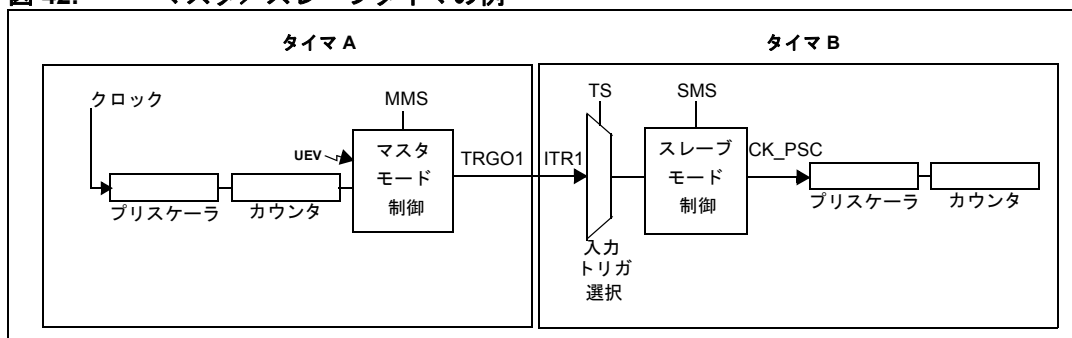
図 41. トリガ/マスタモード選択ブロック





## タイマを別のタイマのプリスケアラとして使用する

図 42. マスタ/スレーブタイマの例



たとえば、タイマ A がタイマ B のプリスケアラとして動作するように設定できます。[図 43](#) を参照してください。このためには、次の操作を行います。

1. 更新イベント UEV ごとに定期的なトリガ信号を出力するように、タイマ A をマスタモードに設定します。更新イベントが生成されるたびに TRGO1 に立ち上がりエッジが出力されるように設定するために、TIMx\_CR2 レジスタに MMS=010 を書き込みます。
2. タイマ A の TRGO1 出力をタイマ B に接続します。タイマ B は、内部トリガとして ITR1 を使用するように、スレーブモードに設定する必要があります。TIMx\_SMCR レジスタの TS ビットを通じて (TS=001 を書き込んで) 選択します。
3. TIMx\_SMCR レジスタに SMS=111 を書き込むことによって、クロック/トリガコントローラを外部クロックモード 1 に設定します。これによりタイマ B は、タイマ A の定期的なトリガ信号の立ち上がりエッジ (タイマ A カウンタのオーバーフローに対応) をクロックとして動作します。
4. 最後に、それぞれの CEN ビット (TIMx\_CR1 レジスタ) をセットすることによって、両方のタイマを有効にします。

**注：** タイマ A のトリガ出力として OC1 が選択された場合 (MMS=1xx)、その立ち上がりエッジがタイマ B カウンタのクロックとして使用されます。

## タイマを使用して別のタイマを有効にする

この例では、タイマ A の出力比較 1 でタイマ B の有効/無効を制御します。接続については、[図 43](#) を参照してください。タイマ B は、タイマ A の OC1REF がハイのときにのみ、分周された内部クロックでカウントします。両方のカウンタクロック周波数は、 $f_{\text{MASTER}} (f_{\text{CK\_CNT}} = f_{\text{MASTER}}/4)$  に対して、プリスケアラによって 4 分周されます。

1. タイマ A 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 (TIMx\_CR2 レジスタの MMS=100)。
2. タイマ A の OC1REF 波形を設定します (TIMx\_CCMR1 レジスタ)。
3. タイマ B がタイマ A から入力トリガを受け取るように設定します (TIMx\_SMCR レジスタの TS=001)。
4. タイマ B をトリガゲートモードに設定します (TIMx\_SMCR レジスタの SMS=101)。
5. CEN ビット (TIMx\_CR1 レジスタ) に 1 を書き込むことによって、タイマ B を有効にします。
6. CEN ビット (TIMx\_CR1 レジスタ) に 1 を書き込むことによって、タイマ A を開始します。

**注：** カウンタ 2 のクロックは、カウンタ 1 と同期されません。このモードは、タイマ B カウンタイネーブル信号にのみ影響します。

図 43. タイマ A の OC1REF によるタイマ B のゲート操作

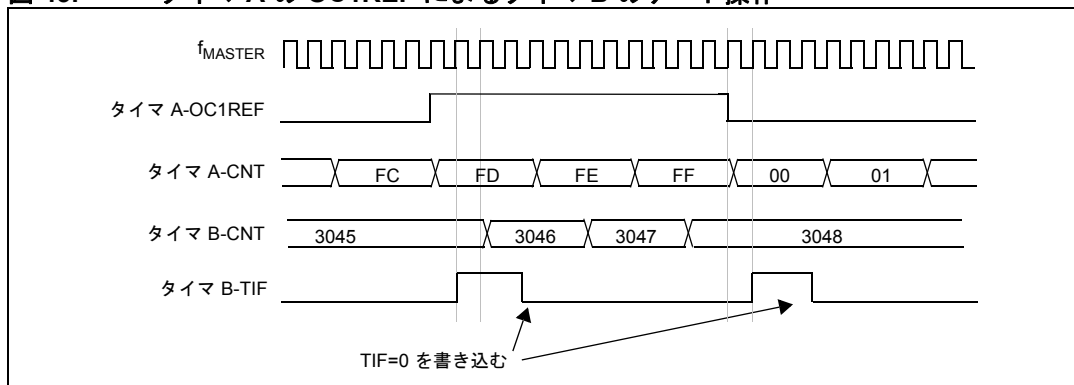
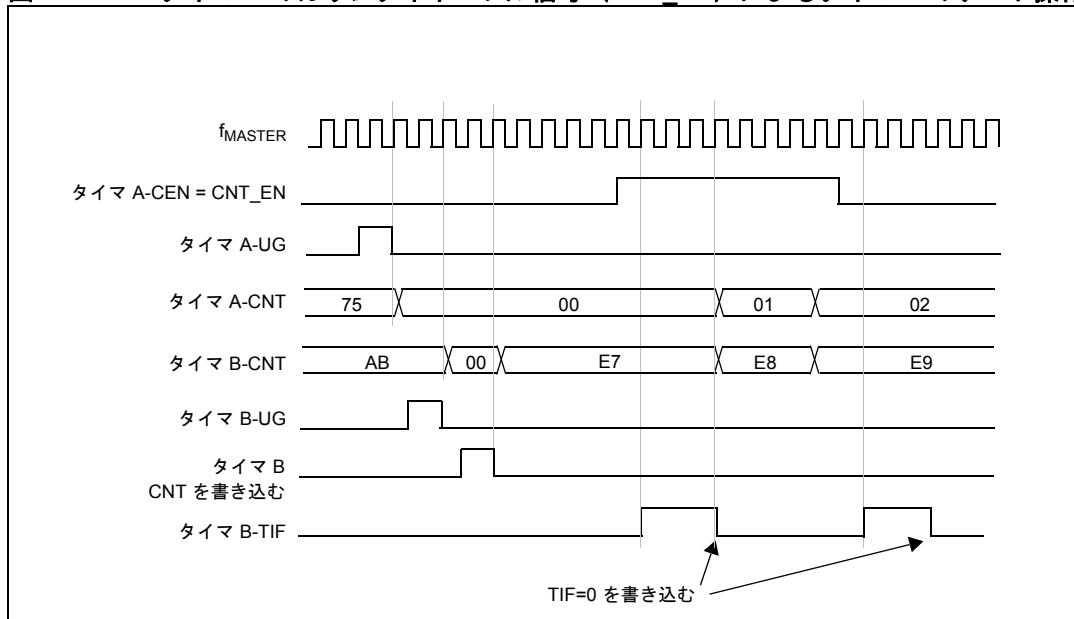


図 43 の例では、タイマ B のカウンタとプリスケアラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。タイマ A を開始する前に、両方のタイマをリセットすることによって、任意の値からカウントを開始することが可能です。この場合、タイマカウンタに任意の値を書き込むことができます。TIMx\_EGR レジスタの UG ビットを使用して、ソフトウェアによって容易にタイマをリセットできます。

次の例では、タイマ A とタイマ B を同期させます。タイマ A はマスタであり、0 から開始します。タイマ B はスレーブであり、E7h 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。TIMx\_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 A を無効にすると、タイマ B は停止します。

1. タイマ A 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 (TIMx\_CR2 レジスタの MMS=100)。
2. タイマ A の OC1REF 波形を設定します (TIMx\_CCMR1 レジスタ)。
3. タイマ B がタイマ A から入力トリガを受け取るように設定します (TIMx\_SMCR レジスタの TS=001)。
4. タイマ B をトリガゲートモードに設定します (TIMx\_SMCR レジスタの SMS=101)。
5. UG ビット (TIMx\_EGR レジスタ) に 1 を書き込むことによって、タイマ A をリセットします。
6. UG ビット (TIMx\_EGR レジスタ) に 1 を書き込むことによって、タイマ B をリセットします。
7. タイマ B のカウンタ (TIMx\_CNTRL) に E7h を書き込むことによって、タイマ B を 0xE7 に初期化します。
8. CEN ビット (TIMx\_CR1 レジスタ) に 1 を書き込むことによって、タイマ B を有効にします。
9. CEN ビット (TIMx\_CR1 レジスタ) に 1 を書き込むことによって、タイマ A を開始します。
10. CEN ビット (TIMx\_CR1 レジスタ) に 0 を書き込むことによって、タイマ A を停止します。

図 44. タイマ A のカウンタインエーブル信号 (CNT\_EN) によるタイマ B のゲート操作

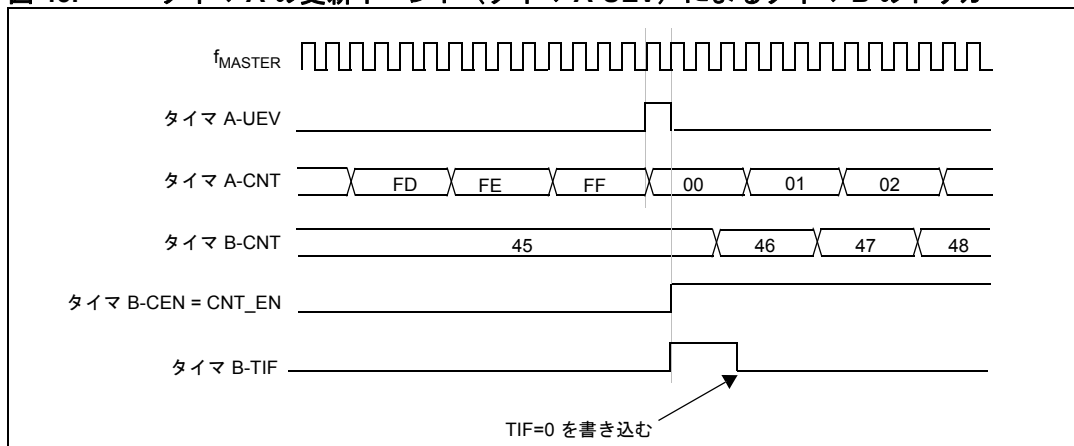


## タイマを使用して別のタイマを開始する

この例では、タイマ A の更新イベントによってタイマ B の有効／無効を制御します。接続については、[図 42](#) を参照してください。タイマ B は、タイマ A によって更新イベントが生成されると同時に、分周された内部クロックで現在値（非ゼロでも可）からカウントを開始します。タイマ B 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 TIMx\_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、 $f_{MASTER}$  ( $f_{CK\_CNT} = f_{MASTER}/4$ ) に比較して、プリスケールによって 4 分周されます。

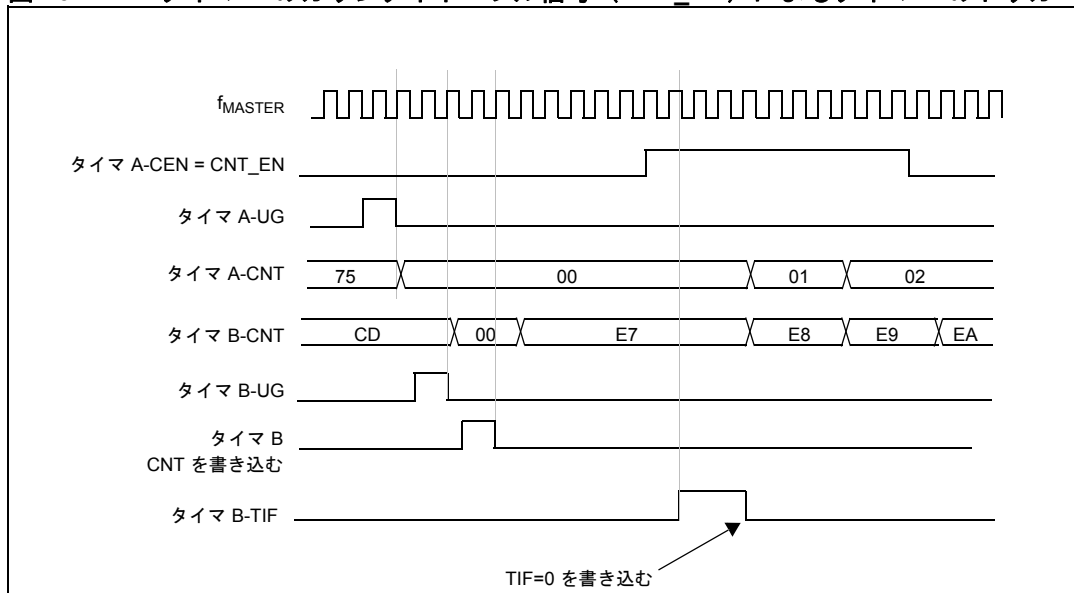
1. 更新イベント (UEV) をトリガ出力として送信するように、タイマ A をマスタモードに設定します (TIMx\_CR2 レジスタの MMS=010)。
2. タイマ A の周期を設定します (TIMx\_ARR レジスタ)。
3. タイマ B がタイマ A から入力トリガを受け取るように設定します (TIMx\_SMCR レジスタの TS=001)。
4. タイマ B をトリガモードに設定します (TIMx\_SMCR レジスタの SMS=110)。
5. CEN ビット (TIMx\_CR1 レジスタ) に 1 を書き込むことによって、タイマ A を開始します。

図 45. タイマ A の更新イベント (タイマ A-UEV) によるタイマ B 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 46 は、図 44 と同じ設定ですが、トリガゲートモードではなくトリガ標準モードでの動作を示しています (TIMx\_SMCR レジスタの SMS=110)。

図 46. タイマ A のカウンタインエーブル信号 (CNT\_EN) によるタイマ B 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する

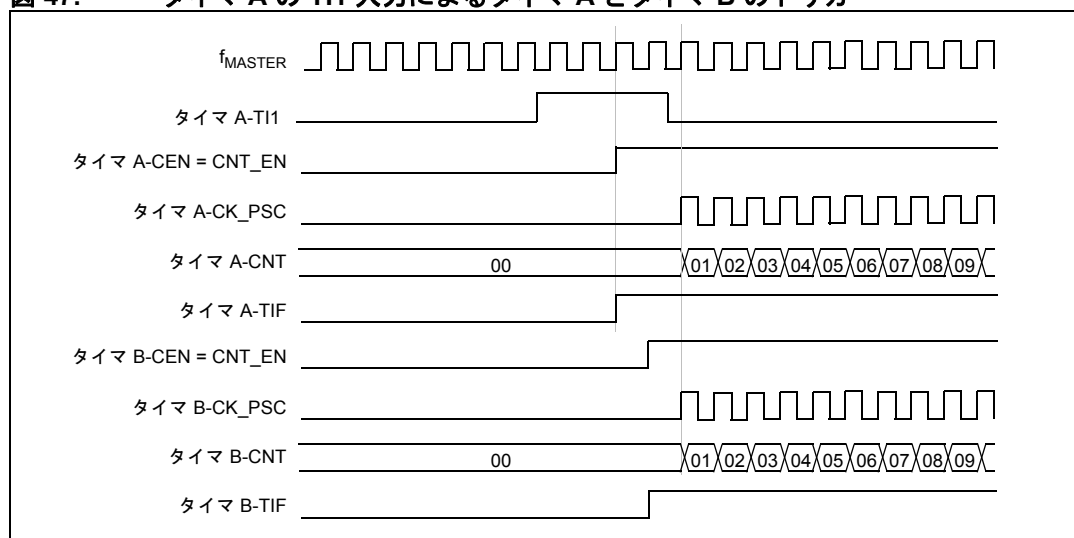
この例では、タイマ A の TI1 入力が入立ち上がるとタイマ A が有効になり、タイマ A が有効になるとタイマ B が有効になります。接続については [図 42](#) を参照してください。カウンタの同時性を確保するには、タイマ A をマスタ/スレーブモードに設定する必要があります（TI1 に対してはスレーブ、タイマ B に対してはマスタ）。

1. イネーブル信号をトリガ出力として送信するように、タイマ A をマスタモードに設定します（TIMx\_CR2 レジスタの MMS=001）。
2. TI1 から入力トリガを受け取るように、タイマ A をスレーブモードに設定します（TIMx\_SMCR レジスタの TS=100）。
3. タイマ A をトリガモードに設定します（TIMx\_SMCR レジスタの SMS=110）。
4. MSM=1（TIMx\_SMCR レジスタ）を書き込むことによって、タイマ A をマスタ/スレーブモードに設定します。
5. タイマ B がタイマ A から入力トリガを受け取るように設定します（TIMx\_SMCR レジスタの TS=001）。
6. タイマ B をトリガモードに設定します（TIMx\_SMCR レジスタの SMS=110）。

TI1（タイマ A）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 A の CNT\_EN と CK\_PSC の間に遅延が挿入されます。

図 47. タイマ A の TI1 入力によるタイマ A とタイマ B のトリガ

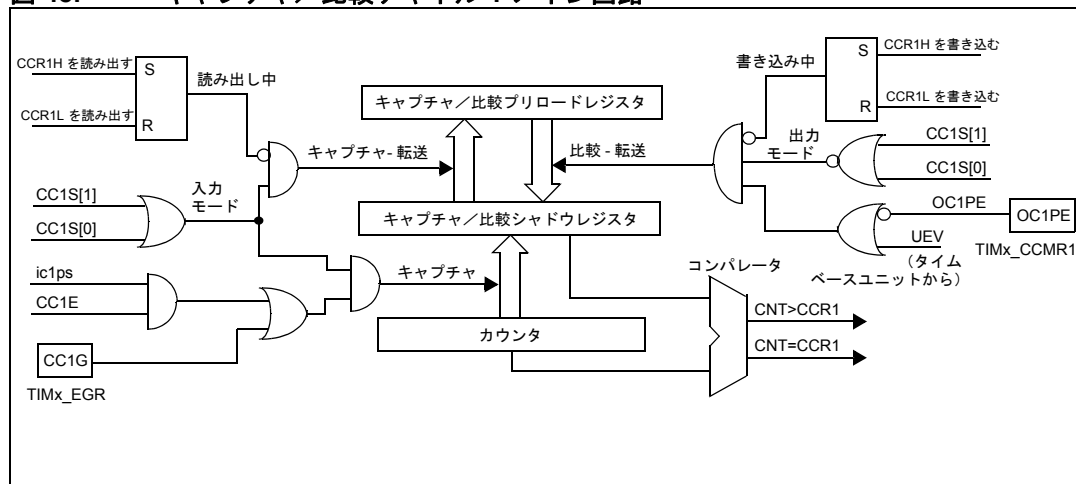


## 16.5 TIMx キャプチャ／比較チャネル

タイマ I/O ピン (TIMx\_CHi) は、入力キャプチャまたは出力比較機能に設定することができます。その選択は、キャプチャ／比較チャネルモードレジスタ (TIMx\_CCMRi) の CCiS チャネル選択ビットを設定することによって行います (i はチャネル番号)。

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ (シャドウレジスタを含む)、キャプチャの入力ステージ (デジタルフィルタ、マルチプレクサ、ブリスケーラ)、および出力ステージ (比較回路と出力制御) から構成されています。

図 48. キャプチャ／比較チャネル 1 メイン回路



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。読み書きのアクセスは、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。

チャネルが出力モード (TIMx\_CCMRi レジスタの CCiS=0b00) に設定されている場合、TIMx\_CCRi レジスタには制約なしにアクセスできます。

チャネルが入力モードに設定されている場合、TIMx\_CCRi レジスタの読み出しシーケンスは、カウンタの読み出しシーケンスと同じです。図 49 を参照してください。キャプチャが発生すると、カウンタの内容が TIMx\_CCRi シャドウレジスタにキャプチャされます。次に、この値がプリロードレジスタにロードされますが、プリロードレジスタがフリーズされている読み出しシーケンス中は除きます。

図 49. キャプチャモードでの TIMx\_CCRi レジスタの 16 ビット読み出しシーケンス

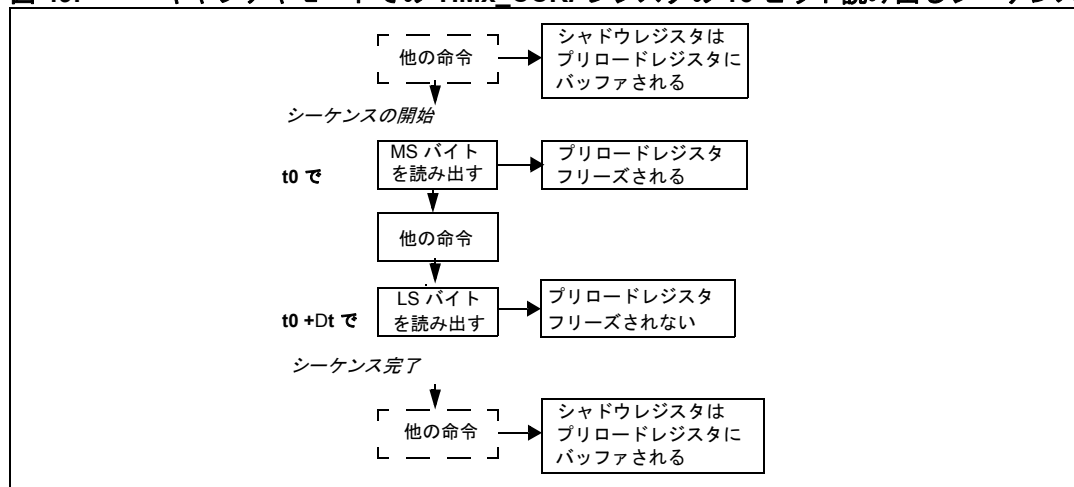


図 49 に、CCR*i* レジスタの 16 ビットタイマ読み出しシーケンスを示します。このバッファされた値は、16 ビット読み出しシーケンスが完了するまで、そのまま保持されます。

読み出しシーケンスの完了後、TIMx\_CCR*L* レジスタのみが読み出された場合、読み出した時点でのカウント値の LS バイトを返します。

LS バイトの後に MS バイトが読み出された場合は、LS バイトと同じキャプチャされた値には対応していません。

## 16.5.1 16 ビット TIMx\_CCRi レジスタの書き込みシーケンス

TIMx\_CCR*i* レジスタには、プリロードレジスタを介して 16 ビットの値がロードされます。これは、2 つの書き込み命令（各バイトについて 1 つずつ）で行う必要があります。MS バイトが最初に書き込まれなければなりません。

シャドウレジスタの更新は、MS バイトが書き込まれると同時にブロックされ、LS バイトが書き込まれるまでブロックされたままです。LDW 命令は最初に LS バイトを書き込むため、この場合は正しくない結果になるので、使用しないでください。

## 16.5.2 入力ステージ

図 50. チャネル入力ステージのブロック図

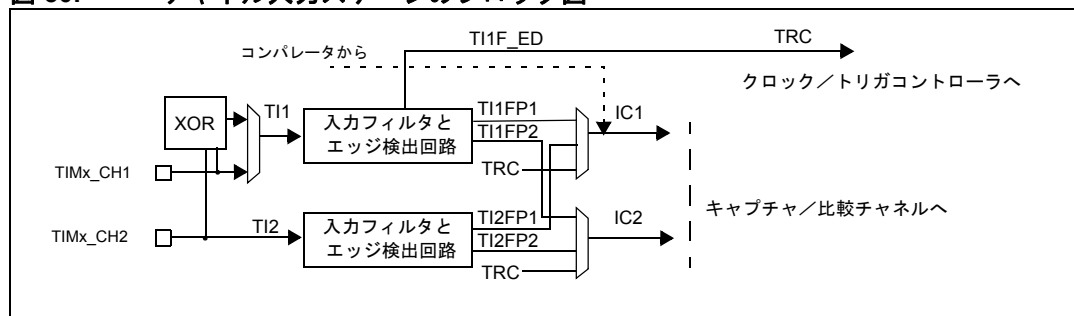
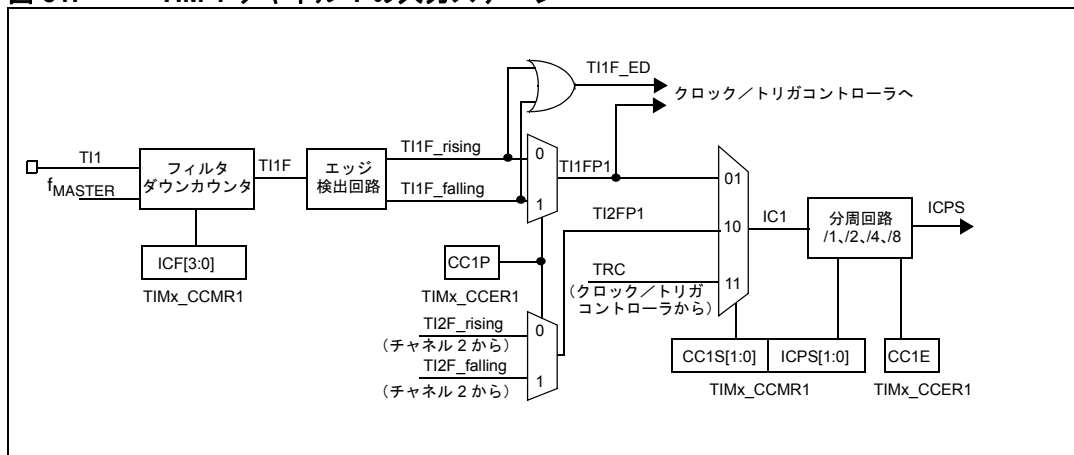


図 51 に示すように、入力ステージでは、対応する TI*i* 入力をサンプリングして、フィルタされた信号 TI*i*F を生成します。次に、極性選択付きのエッジ検出回路が信号（TI*i*FP*x*）を生成し、それがクロック/トリガコントローラによってトリガ入力として、またはキャプチャコマンドとして使用されます。信号は、キャプチャレジスタ（IC*i*PS）よりも前でプリスケールされます。



図 51. TIM 1 チャンネル 1 の入力ステージ



## 16.5.3 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 IC<sub>i</sub> 信号でモード変更が検出された後、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\_CCR<sub>i</sub>) が使用されます。キャプチャが発生すると、対応する CC<sub>i</sub>F フラグ (TIMx\_SR1 レジスタ) がセットされます。

割り込みが有効な場合、TIMx\_IER レジスタの CC<sub>i</sub>IE ビットをセットすることによって、割り込みを送信できます。CC<sub>i</sub>F 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 CC<sub>i</sub>OF (TIMx\_SR2 レジスタ) がセットされます。CC<sub>i</sub>F フラグは、ソフトウェアによって 0 を書き込むことによって、または、TIMx\_CCR<sub>i</sub>L 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。CC<sub>i</sub>OF は、0 を書き込むことによってクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx\_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。たとえば、TIMx\_CCR1 レジスタを TI1 入力にリンクするには、TIMx\_CCMR1 レジスタの CC1S ビットに 0b01 を書き込みます。これによりチャンネルは入力モードに設定され、TIMx\_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号のタイプに応じて必要な入力フィルタ時間をプログラムします。これは、各 TI<sub>i</sub> 入力について、TIMx\_CCMR<sub>i</sub> レジスタの IC<sub>i</sub>F ビットを使用して行われます。たとえば、入力信号がトグルするとき、最大 5 f<sub>MASTER</sub> サイクルまで不安定な場合、フィルタ時間は 5 クロックサイクルより長い必要があります。TIMx\_CCMR1 レジスタのフィルタビットに 0b0011 を書き込むことによって、最大 8 サイクルまでの接続時間を選択できます。このフィルタ設定では、新しいレベルを持つ連続した 8 個のサンプルが検出されたときのみ (f<sub>MASTER</sub> の周波数でサンプリング)、TI1 の遷移が有効です。
3. TI1 チャンネルのアクティブ遷移のエッジを選択するには、TIMx\_CCER1 レジスタの CC1P ビットに 0 を書き込みます (この場合、立ち上がりエッジを選択します)。
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する必要があるため、プリスケアラを無効にします (TIMx\_CCMR1 レジスタの IC1PS ビットに 0b00 を書き込みます)。
5. TIMx\_CCER1 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. 必要の場合は、TIMx\_IER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にします。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx\_CCR1 レジスタに格納されます。
- 入力キャプチャフラグ (CC1IF) がセットされます (割り込みフラグ)。このフラグがクリアされていない状態で少なくとも 2 つの連続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 (CC1OF) もセットされます。
- CC1IE ビットに応じて、割り込みが生成されます。

オーバーキャプチャイベント (CC1OF フラグ) を処理するには、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、フラグ読み出し後、データ読み出し前に発生するオーバーキャプチャの見落としを避けることができます。

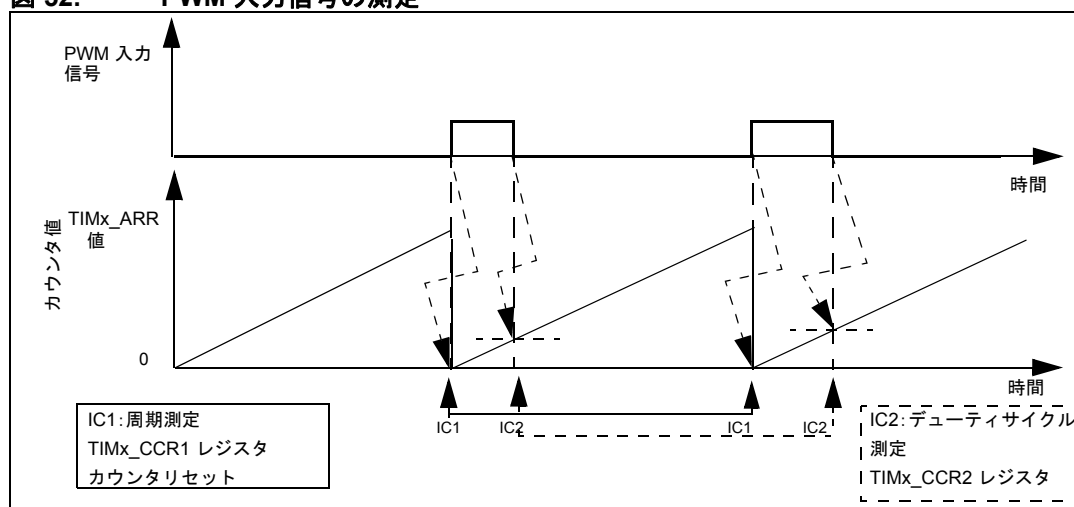
**注：** カウンタ値を実際に TIMx\_CCRx レジスタにキャプチャしなければ、IC 割り込みを送信することはできません。ただし、ソフトウェアで TIMx\_EGR レジスタの対応する CCiG ビットをセットすることによって、キャプチャイベントを生成することはできます。

## PWM 入力信号の測定

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 ICi が、同じ TIi 入力にマッピングされます。
- これら 2 つの ICi は、反対の極性を持つエッジ上でアクティブです。
- 2 つの TIIFP の 1 つがトリガ入力として選択され、クロック／トリガコントローラはトリガリセットモードに設定されます。

図 52. PWM 入力信号の測定

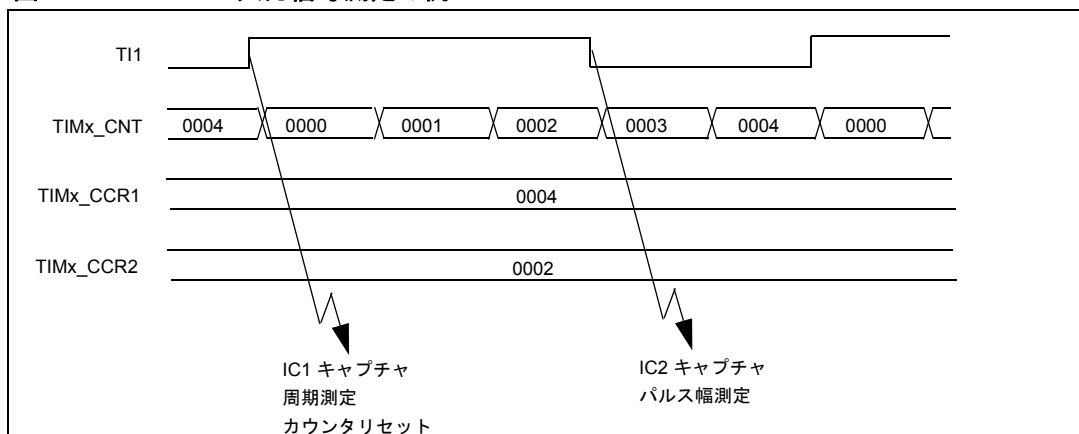


たとえば、 $f_{\text{MASTER}}$  周波数とプリスケアラ値に応じて次の手順を使用すれば、TI1 に印加された PWM の周期 (TIMx\_CCR1 レジスタ) とデューティサイクル (TIMx\_CCR2 レジスタ) を測定できます。

1. TIMx\_CCMR1 レジスタの CC1S ビットに 0b01 を書き込むことによって (TI1FP1 を選択)、TIMx\_CCR1 のアクティブ入力キャプチャまたはトリガ入力を選択します。
2. CC1P 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されます)。
3. TIMx\_CCMR2 レジスタの CC2S ビットに 0b10 を書き込むことによって (TI1FP2 を選択)、TIMx\_CCR2 のアクティブ入力を選択します。
4. CC2P ビットに "1" を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 0b101 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、クロック／トリガコントローラをリセットモードに設定します。
7. TIMx\_CCER1 レジスタの CC1E および 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 53. PWM 入力信号測定の実例



## 16.5.4 出力ステージ

出力ステージは、OCiREF (アクティブハイ) と呼ばれ、基準電圧として使用される中間波形を生成します。ブレイク機能と極性は、最終出力に影響を与えます。

図 54. チャネル出力ステージのブロック図

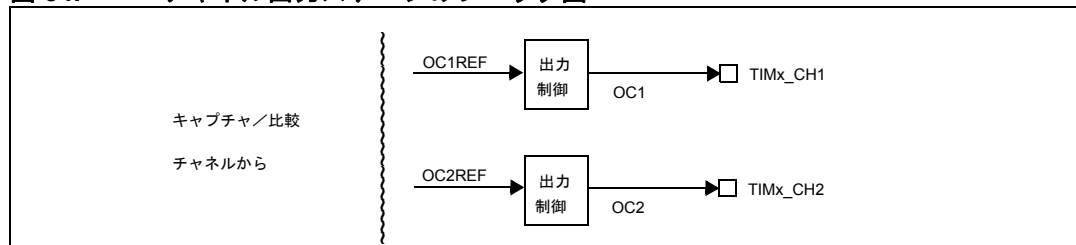
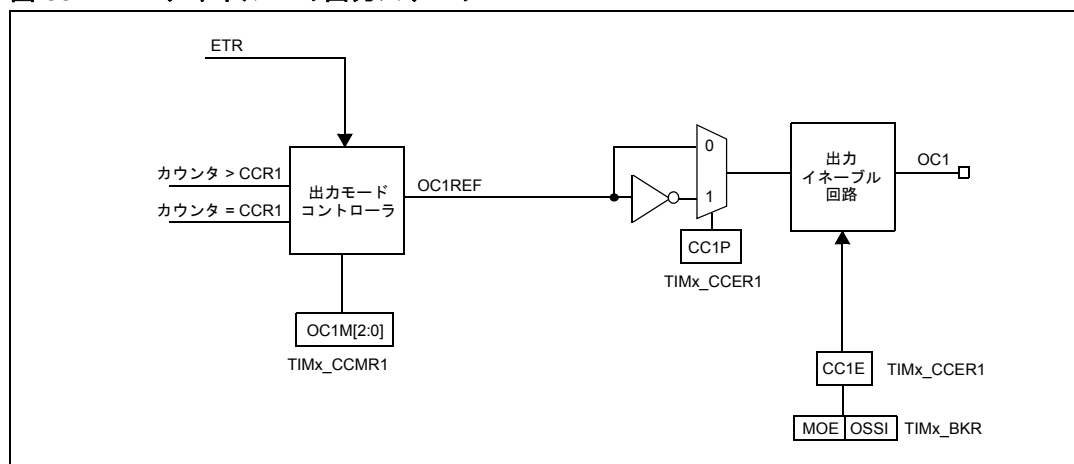


図 55. チャンネル 1 の出力ステージ



## 16.5.5 強制出力モード

出力モード (TIMx\_CCMR*i* レジスタの CC*i*S ビット = 0b00、*i* はチャンネル番号) では、出力比較レジスタとカウンタの比較とは関係なく、ソフトウェアによって直接、各出力比較信号を強制的にハイまたはローレベルにできます。

出力比較信号を強制的にアクティブレベルにするには、対応する TIMx\_CCMR*i* レジスタの OC*i*M ビットに“101”を書き込むだけです。したがって、OC*i*REF は強制的にハイになり (OC*i*REF は常にアクティブハイ)、OC*i* 出力は CC*i*P の極性ビットに応じて強制的にハイまたはローレベルになります。

たとえば、CC*i*P = 0 (OC*i* アクティブハイ) の場合、OC*i* は強制的にハイレベルになります。

OC*i*REF 信号は、TIMx\_CCMR*x* レジスタの OC*i*M ビットに 0b100 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\_CCR*i* シャドウレジスタとカウンタの比較は行われるので、フラグをセットできます。それに応じて、割り込みリクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

## 16.5.6 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、

- 出力比較モードに応じて、対応する OC*i* 出力ピンは、
  - レベルを維持します (OC*i*M = 0b000)。
  - アクティブにセットされます (OC*i*M = 0b001)。
  - インアクティブにセットされます (OC*i*M = 0b010)。
  - トグルします (OC*i*M = 0b011)。
- 割り込みステータスレジスタのフラグがセットされます (TIMx\_SR1 レジスタの CC*i*IF ビット)。
- 対応する割り込みマスク (TIMx\_IER レジスタの CC*i*IE 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。

出力比較モードは、TIMx\_CCMR*i* レジスタの OC*i*M ビットによって定義されます。アクティブまたはインアクティブレベルの極性は、TIMx\_CCER*i* レジスタの CC*i*P ビットによって定義されます。

TIMx\_CCR*i* レジスタは、TIMx\_CCMR*i* レジスタの OC*i*PE ビットを使用して、プリロードレジスタを使用して、またはプリロードレジスタを使用せずにプログラミングできます。

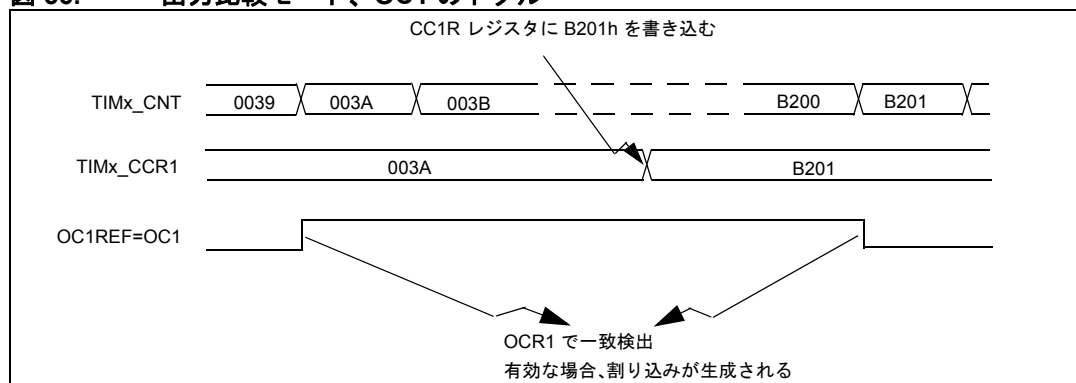
出力比較モードでは、更新イベント UEV は OC*i*REF および OC*i* 出力に影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは、単一パルスを出力するためにも使用できます。

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\_CCRi レジスタに希望のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCiE ビットをセットします。
4. 出力モードを以下のように選択します。
  - － CNT が CCRi に一致したときに OCi 出力ピンを反転するには、OCiM = 0b011 を書き込みます。
  - － プリロードレジスタを無効にするには、OCiPE = 0 を書き込みます。
  - － アクティブハイ極性を選択するには、CCiP = 0 を書き込みます。
  - － 出力を有効にするには、CCiE = 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。

TIMx\_CCRi レジスタは、ソフトウェアによっていつでも更新して、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 (OCiPE=0) に限ります。そうでない場合は、TIMx\_CCRi 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 56](#) に示します。

**図 56. 出力比較モード、OC1 のトグル**



## 16.5.7 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\_CCRi レジスタの値によって決められたデューティサイクルを持つ信号を生成できます。

PWM モードは、TIMx\_CCMRi レジスタの OCiM ビットに 0b110 (PWM モード 1) または 0b111 (PWM モード 2) を書き込むことによって、チャンネルごとに個別に選択できます (OCi 出力ごとに 1 つの PWM)。TIMx\_CCMRi レジスタの OCiPE 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。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 (アップカウントまたはセンターアラインモード) を、オポジションで有効にすることができます。

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。

OCi 極性は、TIMx\_CCERi レジスタの CCiP 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ローとしてプログラムできます。OCi 出力は、CCiE、MOE、OISi、OSSR、および OSSI ビット (TIMx\_CCERi および TIMx\_BKR レジスタ) の組み合わせによって有効にされます。詳細については、TIMx\_CCERi 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\_CCRi が常に比較されて、TIMx\_CCRi ≤ TIMx\_CNT または TIMx\_CNT ≤ TIMx\_CCRi かどうか判断されます (カウンタの方向によります)。

タイマは、TIMx\_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

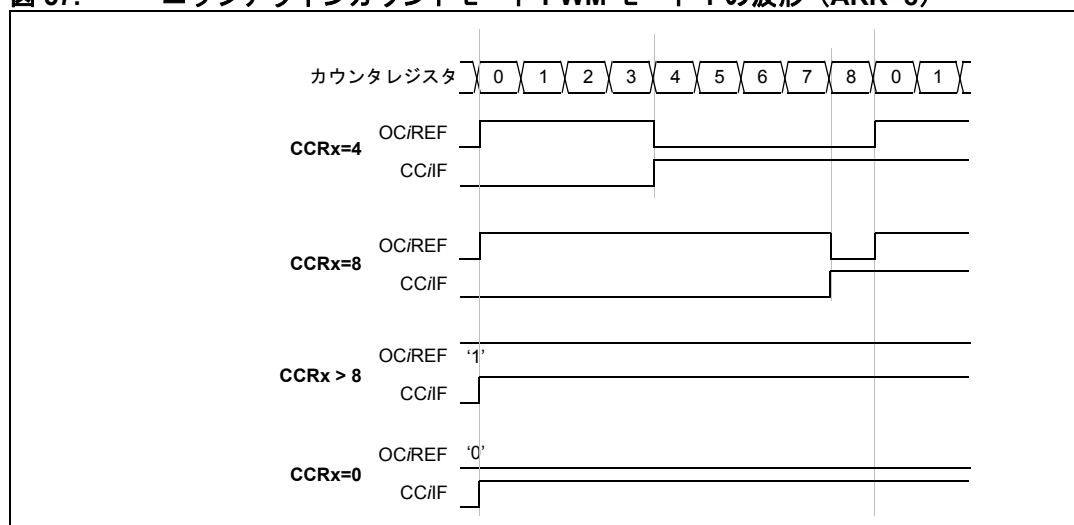
### PWM エッジアラインモード

#### アップカウント設定

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。

次の例では、PWM モード 1 を使用しています。基準の PWM 信号、OC/REF は TIMx\_CNT < TIMx\_CCRi である限り、ハイです。そうでない場合は、ローになります。TIMx\_CCRi の比較値が自動再ロード値 (TIMx\_ARR) より大きい場合、OC/REF は 1 に保持されます。比較値が 0 の場合、OC/REF は 0 に保持されます。図 57 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 57. エッジアラインカウントモード PWM モード 1 の波形 (ARR=8)



#### ダウンカウント設定

TIMx\_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。[ダウンカウントモード \(106 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OC/REF は、TIMx\_CNT > TIMx\_CCRi である限り、ローです。そうでない場合は、ハイになります。TIMx\_CCRi レジスタの比較値が自動再ロード値 (TIMx\_ARR レジスタ) より大きい場合、OC/REF は 1 に保持されます。このモードでは、0 % の PWM 信号を生成することはできません。

### PWM センターアラインモード

センターアラインモードは、TIMx\_CR1 レジスタの CMS ビットが“00”でないときにアクティブです (その他すべての設定が、OC/REF/OCi 信号に対して同じ効果を持ちます)。

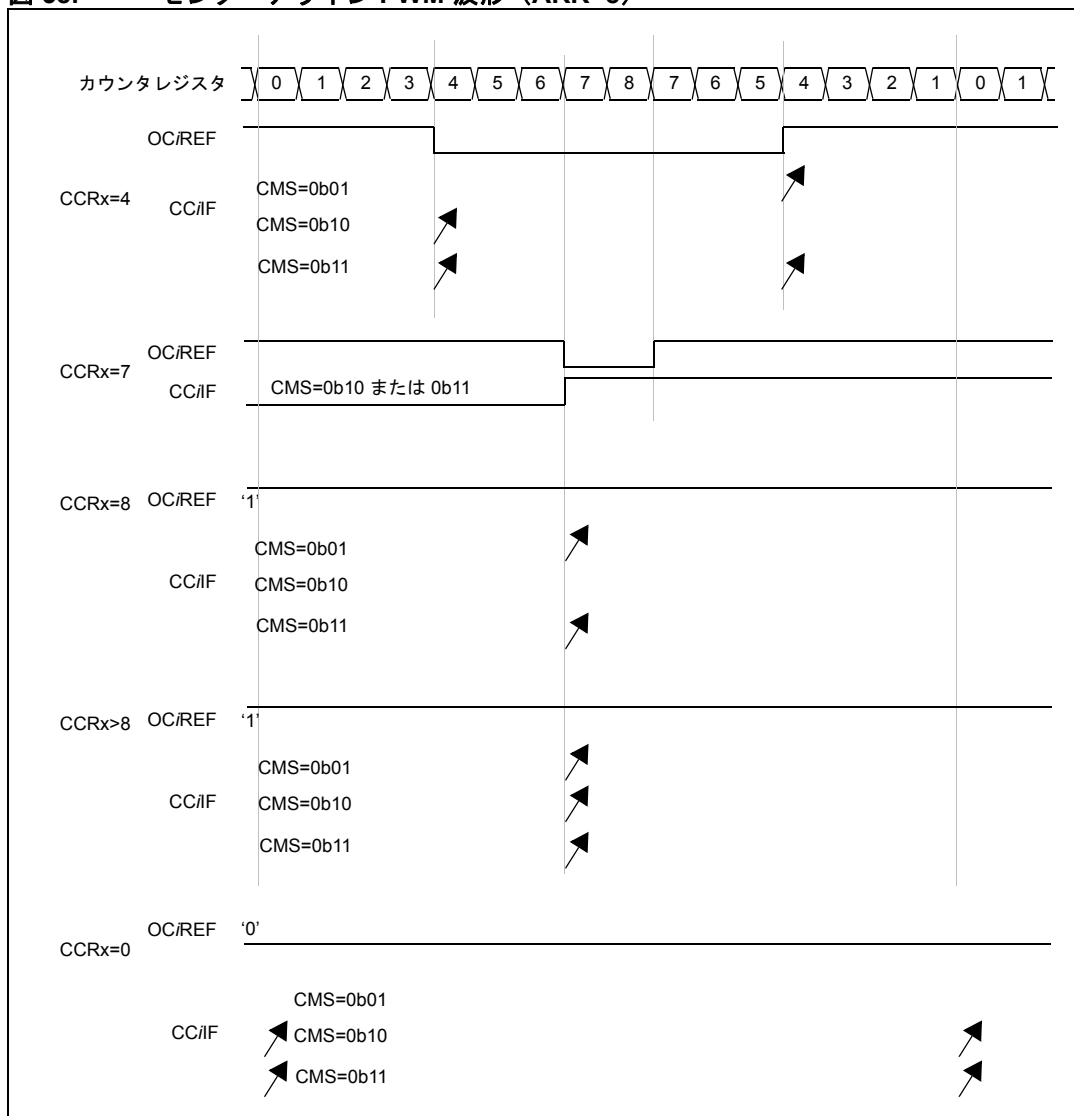
比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx\_CR1 レジスタの方向ビット (DIR) は、ハードウェアによって更新され、このモードでは読み出し専用です。[センターアラインモード \(アップ/ダウンカウント\) \(108 ページ\)](#) を参照してください。



図 58 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、次の 3 つの場合にセットされます (図 58 の矢印)。
  - カウンタがカウントダウンしているときのみ (CMS = 0b01)
  - カウンタがカウントアップしているときのみ (CMS = 0b10)
  - 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 (CMS = 0b11)

図 58. センターアライン PWM 波形 (ARR=8)





## 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。

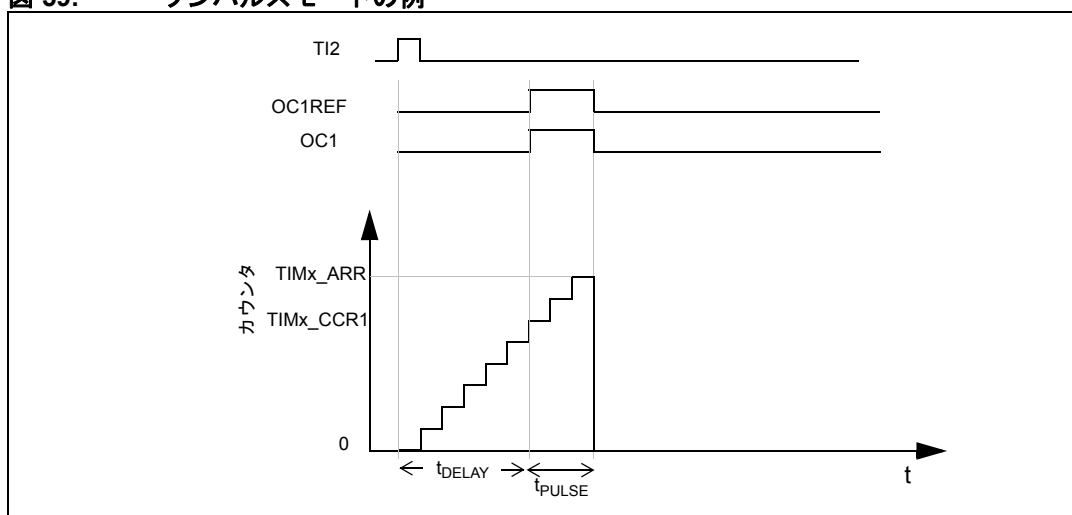
カウンタの開始は、クロック／トリガコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx\_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

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 :  $CNT < CCR1 \leq ARR$  (特に、 $0 < CCR1$ )

ダウンカウント時 :  $CNT > CCR1$

図 59. ワンパルスモードの例



たとえば、TIM2 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。

IC2 をトリガ 1 として使用します。

- TIMx\_CCMR2 レジスタに  $CC2S=0b01$  を書き込むことによって、IC2 を TIM2 に配置します。
- IC2 は立ち上がりエッジを検出しなければならないので、TIMx\_CCER1 レジスタに  $CC2P=0$  を書き込みます。
- TIMx\_SMCR レジスタに  $TS=0b110$  を書き込むことによって、IC2 をクロック／トリガコントローラ (TRGI) のトリガとして設定します。
- IC2 を使用してカウンタを開始するには、TIMx\_SMCR レジスタの SMS ビットに  $0b110$  (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。そのためには、TIMx\_CCMR1 レジスタに  $OC1M=0b111$  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 レジスタに OC1PE="1" を書き込み、TIMx\_CR1 レジスタに ARPE=0 を書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx\_CCR1 レジスタに比較値を書き込み、TIMx\_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイベントを待ちます。この例では、CC1P に"0"を書き込みます。

上の例では、TIMx\_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので、次の更新イベント（カウンタが自動再ロード値から 0 に戻る時点）でカウンタが停止するように、TIMx\_CR1 レジスタの OPM ビットに 1 を書き込みます。

特殊な場合：OCi 高速イネーブル：

ワンパルスモードでは、TIi 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 ( $t_{\text{DELAY min}}$ ) を制限します。

最小遅延の波形を出力するには、TIMx\_CCMRi レジスタの OCiFE ビットをセットします。OCiREF（および OCi）は、比較結果に関係なく、トリガに応じてレベルが強制的に決まります。その新しいレベルは、比較が一致した場合と同じです。OCiFE 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にのみ機能します。

## 16.5.8 ブレーク機能の使用

ブレーク機能は、しばしばモータ制御に使用されます。ブレーク機能を使用すると、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\_BKR レジスタの MOE、OSSR、および OSSI ビット）に応じて変更されます。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。ブレーク機能は、TIMx\_BKR レジスタの BKE ビットをセットすることによって有効になります。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。

MOE の立ち下がりエッジは非同期のことがあるので、実際の信号（出力に作用する）と同期制御ビット（TIMx\_BKR レジスタからアクセス）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。

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 になると、TIMx\_OISR レジスタの OISi ビットでプログラムされたレベルに駆動されます。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。
- ブレーク状態フラグ（TIMx\_SR1 レジスタの BIF ビット）がセットされます。TIMx\_IER レジスタの BIE ビットがセットされている場合は、割り込みを生成できます。
- TIMx\_BKR 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。

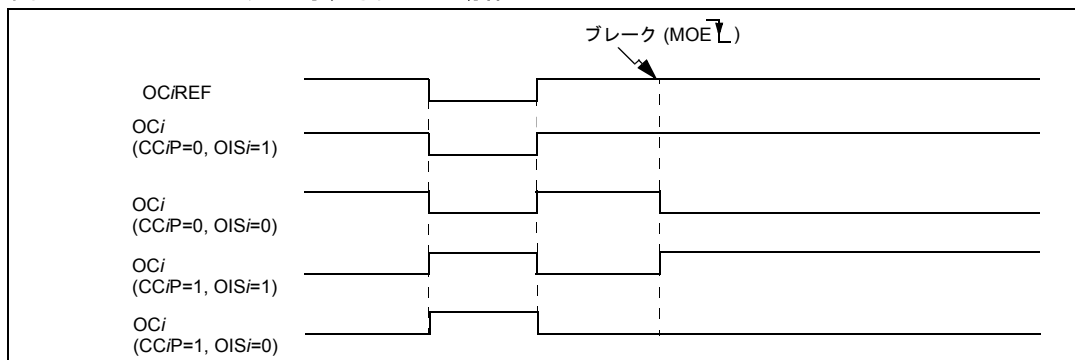
**注：** **ブレーク入力は、信号レベルに対して作用します。このため、ブレーク入力がアクティブな間は、MOE ビットをセットできません（自動的に、ソフトウェアによっても）。この間、ステータスフラグ BIF をクリアできません。**

ブレークは、プログラム可能な極性を持つブレーク入力 (BKIN) によって生成でき、TIMx\_BKR レジスタの BKE ビットをセットまたはリセットすることによって、有効または無効にできます。

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が施されています。いくつかのパラメータの設定 (無効なときの OC<sub>i</sub> の極性と状態、OC<sub>i</sub>M の設定、ブレークイネーブル、および極性) を凍結することができます。TIMx\_BKR レジスタの LOCK ビットを使用して、3 つの保護レベルを選択できます。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 60 に、ブレークに対する出力の動作例を示します。

図 60. ブレークに対する出力の動作



### 16.5.9 外部イベント時の OC/REF 信号のクリア

ETRF でハイレベルが検出されたときに、特定のチャネルの OC/REF 信号をクリアできます (TIMx\_CCMR<sub>i</sub> レジスタの OC<sub>i</sub>CE = 1 の場合、チャネルあたり 1 つのイネーブルビット)。OC/REF 信号は、次の UEV 更新イベントが発生するまで、ローのままです。この機能は、出力比較モードと PWM モードでのみ使用でき、強制モードでは使用できません。

たとえば、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。

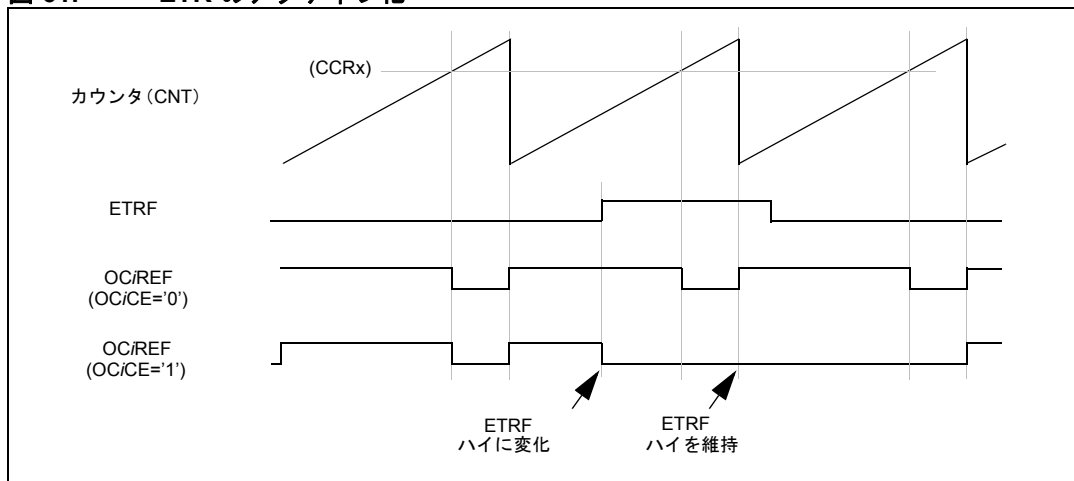
たとえば、OC/REF 信号をコンパレータの出力に接続して、電流処理に使用することができます。この場合、外部トリガを以下のように設定する必要があります。

1. TIMx\_ETR レジスタの ETPS[1:0] ビットに“00”を設定することによって、外部トリガプリスケアラをオフに保持します。
2. TIMx\_ETR レジスタの ECE ビットに“0”を設定することによって、外部クロックモード 2 を無効にします。
3. 外部トリガの極性 (ETP) と外部トリガフィルタ (ETF) を必要に応じて設定します。

外部トリガ入力のブロック図 (図 34 (112 ページ)) を参照してください。

図 61 に、イネーブルビット OC<sub>i</sub>CE の両方の値について、ETRF 入力が高になったときの OC/REF 信号の動作を示します。この例では、タイマは PWM モードにプログラミングされています。

図 61. ETR のアクティブ化



## 16.5.10 エンコーダインタフェースモード

このモードは、通常、モータ制御に使用されます。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=0b001 を、TI1 エッジのみをカウントしている場合は SMS=0b010 を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS=0b011 を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx\_CCER1 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。必要に応じて、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 33 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、TIMx\_ARR は、開始前に設定されなければなりません。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。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 33. 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	カウントしない	カウントしない
	ロー	アップ	ダウン	カウントしない	カウントしない
TI2 のみカウント	ハイ	カウントしない	カウントしない	アップ	ダウン
	ロー	カウントしない	カウントしない	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、外部にインタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

図 62 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 = 0b01 (TIMx\_CCMR1 レジスタ、IC1 は TI1 に配置)
- CC2S = 0b01 (TIMx\_CCMR2 レジスタ、IC2 は TI2 に配置)
- CC1P = 0 (TIMx\_CCER1 レジスタ、IC1 非反転、IC1=TI1)
- CC2P = 0 (TIMx\_CCER2 レジスタ、IC2 非反転、IC2=TI2)
- SMS = 0b011 (TIMx\_SMCR レジスタ、両方の入力立ち上がり立ち下がり両エッジでアクティブ)
- CEN = 1 (TIMx\_CR1 レジスタ、カウンタ有効)



図 62. 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

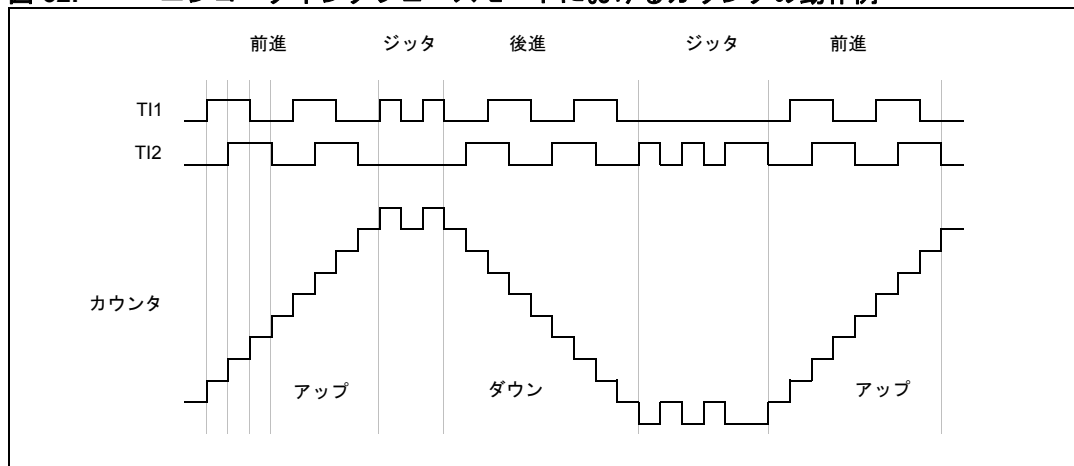
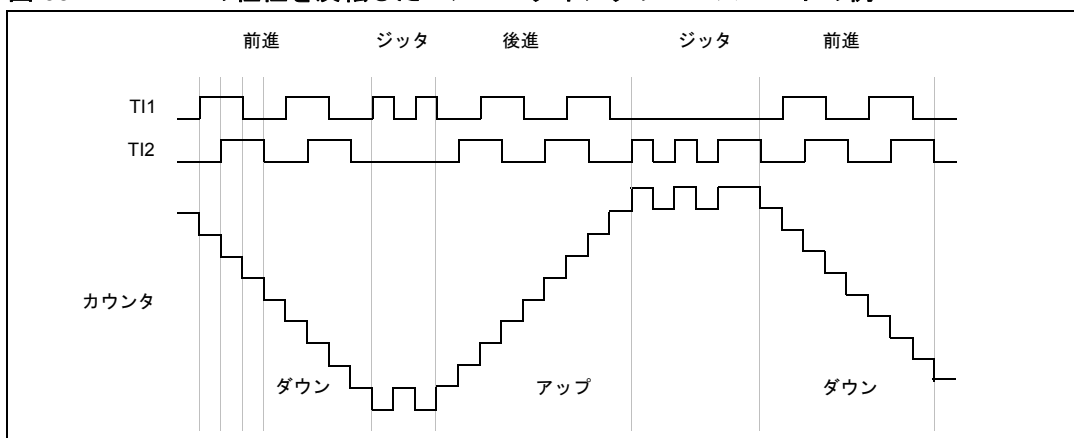


図 63 は、IC1 の極性を反転したときのカウンタの動作例を示しています（上と同じ設定ですが、CC1P=1 である点が異なります）。

図 63. IC1 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設定された 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によって、速度、加速度、減速度などの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。これは、3 番目の入力キャプチャレジスタが使用可能な場合、これにカウンタ値をラッチすることによって行うことができます。この場合、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。



## 16.6 TIMx 割り込み

TIMx には 6 つの割り込みリクエストソースがあり、2 つの割り込みベクタに配置されています。

- ブレーク割り込み
- トリガ割り込み
- 転流割り込み
- キャプチャ／比較 2 割り込み
- キャプチャ／比較 1 割り込み
- 更新割り込み（例：オーバーフロー、アンダーフロー、カウンタ初期化）

使用されている各割り込みチャネルに対して割り込み機能を使用するには、TIMx\_IER レジスタの必要な割り込みイネーブルビット（BIE、TIE、COMIE、CCIE、および UIE）をセットして、割り込みリクエストを有効にします。

TIMx\_EGR レジスタの対応するビットを使用して、ソフトウェアによってさまざまな割り込みソースを生成することもできます。

### 16.6.1 TIMx イベント待ち機能

イベント待ちモード（WFE）では、TIMx キャプチャ／比較、ブレーク、トリガ、および更新割り込みを使用して、デバイスをウェイクアップできます。そのためには、WFE\_CR1 レジスタの TIMx\_EV0 および TIMx\_EV1 を通じて、割り込みイベントをあらかじめ設定しておく必要があります（[セクション 9.5 : WFE レジスタ](#)を参照）。



## 16.7 TIMx レジスタ

### 16.7.1 制御レジスタ 1 (TIMx\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x00

7	6	5	4	3	2	1	0
ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
rw	rw		rw	rw	rw	rw	rw

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、プリロードレジスタを通じてバッファされません。直接書き込むことができます。
- 1 : TIMx\_ARR レジスタは、プリロードレジスタを通じてバッファされます。

ビット 6:5 **CMS** : センターアラインモード選択

- 00 : エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。
- 01 : センターアラインモード 1。カウンタは、カウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。
- 10 : センターアラインモード 2。カウンタは、カウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。
- 11 : センターアラインモード 3。カウンタは、カウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx\_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

**注 :** カウンタが有効 (CEN=1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。  
センターアラインモードでは、エンコーダモード (GPT\_SMCR レジスタの SMS = 001、010、または 011) を無効にする必要があります。

ビット 4 **DIR** : 方向

- 0 : カウンタはアップカウンタとして使用されます。
- 1 : カウンタはダウンカウンタとして使用されます。

**注 :** このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリアします)。

ビット 2 **URS** : 更新リクエストソース

- 0 : 有効にされている場合、レジスタが更新されると (カウンタオーバーフロー)、更新割り込みリクエストが送信されます。
- 1 : 有効にされている場合、カウンタがオーバーフローしたときにのみ、更新割り込みリクエストが送信されます。

ビット 1 **UDIS** : 更新ディセーブル

- 0 : カウンタオーバーフローが発生するか、ソフトウェア更新が生成されるか、またはクロックノトリガモードコントローラによってハードウェアリセットが生成されると、更新イベントが生成されます。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 : 更新イベントは生成されず、シャドウレジスタは値を維持します (ARR、PSC、CCRx)。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。

16.7.2 制御レジスタ 2（TIMx\_CR2）

アドレスオフセット : 0x01

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	MMS[2:0]			予約済み			
	rw						

ビット 7 予約済み

ビット 6:4 **MMS** : マスタモード選択

これらのビットは、同期をとるためにマスタモードで他のタイマに送信される情報を選択します（TRGO）。組み合わせは、次のとおりです。

- 000 : **リセット** - TIM3\_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た場合（クロック／トリガモードコントローラがトリガリセットモードに設定されているとき）、TRGO 信号は実際のリセットから遅れて発生します。
- 001 : **イネーブル** - カウンタイネーブル信号がトリガ出力（TRGO）として使用されます。複数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ために使用されます。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には、マスタ／スレーブモードが選択されている場合を除き、TRGO に遅延が存在します（TIM3\_SMCR レジスタの MSM ビットの説明を参照してください）。
- 010 : **更新** - 更新イベントがトリガ出力（TRGO）として選択されます。
- 011 : 予約済み
- 100 : 予約済み
- 101 : 予約済み
- 111 : 予約済み

ビット 3:0 予約済み



## 16.7.3 スレーブモード制御レジスタ（TIMx\_SMCR）

アドレスオフセット：0x02

リセット値：0x00

7	6	5	4	3	2	1	0
MSM	TS[2:0]			予約済み	SMS[2:0]		
rw	rw				rw		

ビット 7 **MSM**：マスタ／スレーブモード

- 0：影響なし。
- 1：イベントがトリガ入力（TRGI）に与える効果は、タイマ間の完全な同期（TRGO を通じて）を可能にするために遅延されます。

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 ITR0 は TIM4 TRGO に接続されます。
  - 001：予約済み
  - 010：内部トリガ ITR2 は TIM3 TRGO に接続されます。
  - 011：内部トリガ ITR3 は TIM2 TRGO に接続されます。
  - 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）

**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、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

ビット 2:0 **SMS[2:0]**：クロック／トリガ／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）。
- 000：クロック／トリガコントローラは無効です。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：トリガリセットモード - 選択されたトリガ信号（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、トリガ信号（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で、カウンタにクロックが供給されます。

## 16.7.4 外部トリガレジスタ (TIMx\_ETR)

アドレスオフセット : 0x03

リセット値 : 0x00

7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]			
rw	rw	rw		rw			

ビット 7 **ETP** : 外部トリガ極性

このビットは、ETR と  $\overline{\text{ETR}}$  のいずれがトリガ動作に使用されるかを選択します。

0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。

1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブです。

ビット 6 **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 :** *ECE* ビットをセットすると、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 (TIMx\_SMCR レジスタの SMS=111、TS=111)。

*トリガ標準モード、トリガリセットモード、およびトリガゲートモードでは、外部クロックモード 2 を同時に使用することが可能です。ただし、この場合、TRGI を ETRF に接続することはできません (TIMx\_SMCR レジスタの TS ビットを 111 にしないでください)。*

*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 5:4 **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  $f_{\text{MASTER}}$  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 3:0 **ETF** : 外部トリガフィルタ

このビットフィールドは、ETRP 信号のサンプリングに使用される周波数と、それ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、サンプリングは  $f_{\text{MASTER}}$  で行われます 1000 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/8$ 、N=6

0001 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}$ 、N=2 1001 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/8$ 、N=8

0010 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}$ 、N=4 1010 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/16$ 、N=5

0011 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}$ 、N=8 1011 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/16$ 、N=6

0100 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/2$ 、N=6 1100 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/16$ 、N=8

0101 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/2$ 、N=8 1101 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/32$ 、N=5

0110 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/4$ 、N=6 1110 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/32$ 、N=6

0111 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/4$ 、N=8 1111 :  $f_{\text{SAMPLING}}=f_{\text{MASTER}}/32$ 、N=8

## 16.7.5 割り込みイネーブルレジスタ（TIMx\_IER）

アドレスオフセット：0x04

リセット値：0x00

7	6	5	4	3	2	1	0
BIE	TIE	予約済み			CC2IE	CC1IE	UIE
rw	rw				rw	rw	rw

ビット 7 **BIE**：ブレーク割り込みイネーブル  
 0：ブレーク割り込みは無効です。  
 1：ブレーク割り込みは有効です。

ビット 6 **TIE**：トリガ割り込みイネーブル  
 0：トリガ割り込みは無効です。  
 1：トリガ割り込みは有効です。

ビット 5:3 予約済み

ビット 2 **CC2IE**：キャプチャ／比較 2 割り込みイネーブル  
 0：CC2 割り込みは無効です。  
 1：CC2 割り込みは有効です。

ビット 1 **CC1IE**：キャプチャ／比較 1 割り込みイネーブル  
 0：CC1 割り込みは無効です。  
 1：CC1 割り込みは有効です。

ビット 0 **UIE**：更新割り込みイネーブル  
 0：更新割り込みは無効です。  
 1：更新割り込みは有効です。

## 16.7.6 ステータスレジスタ 1（TIMx\_SR1）

アドレスオフセット：0x05

リセット値：0x00

7	6	5	4	3	2	1	0
BIF	TIF	予約済み			CC2IF	CC1IF	UIF
rc_w0	rc_w0				rc_w0	rc_w0	rc_w0

ビット 7 **BIF**：ブレーク割り込みフラグ  
 このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。  
 0：ブレークイベントは発生していません。  
 1：ブレーク入力でアクティブレベルが検出されました。

ビット 6 **TIF**：トリガ割り込みフラグ  
 このフラグは、トリガイイベント時に（TRGI 信号でアクティブエッジが検出されたとき、またはゲートモードが選択されている場合は両方のエッジが検出されたとき）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：トリガイイベントは発生していません。  
 1：トリガ割り込みがペンディング中です。

ビット 5:3 予約済み

ビット 3 **CC3IF**：キャプチャ／比較 3 割り込みフラグ  
 CC1IF の説明を参照してください。

- ビット 2 **CC2IF** : キャプチャ／比較 2 割り込みフラグ  
CC1IF の説明を参照してください。
- ビット 1 **CC1IF** : キャプチャ／比較 1 割り込みフラグ
- チャンネル CC1 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 : 一致なし。  
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。
  - チャンネル CC1 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\_CCR1L レジスタを読み出すことによってクリアされます。  
0 : 入力キャプチャは発生していません。  
1 : カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。
- ビット 0 **UIF** : 更新割り込みフラグ
- このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされず。
- 0 : 更新は発生していません。  
1 : 更新割り込みがペンディング中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。
- TIMx\_CR1 レジスタの UDIS = 0 の場合、オーバーフローが発生したとき。
  - 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。

## 16.7.7 ステータスレジスタ 2 (TIMx\_SR2)

アドレスオフセット : 0x06

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み					CC2OF	CC1OF	予約済み
					rc_w0	rc_w0	

- ビット 7:3 予約済み
- ビット 2 **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 1 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ
- このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。
- 0 : オーバキャプチャは検出されていません。  
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx\_CCR1 レジスタにキャプチャされました。
- ビット 0 予約済み

## 16.7.8 イベント生成レジスタ（TIMx\_EGR）

アドレスオフセット：0x07

リセット値：0x00

7	6	5	4	3	2	1	0
BG	TG	予約済み			CC2G	CC1G	UG
W	W				W	W	W

### 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。BIE ビットによって有効にされている場合、割り込みが生成されます。

### 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\_SR1 レジスタの TIF フラグがセットされます。TIE ビットによって有効にされている場合、割り込みが生成されます。

### ビット 5:3 予約済み

### 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 チャンネルが出力モードに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 チャンネルが入力モードに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ることに注意してください。



16.7.9 キャプチャ／比較モードレジスタ 1（TIMx\_CCMR1）

このチャンネルは、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、CC1S ビットを設定することによって定義されます。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示します。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。

アドレスオフセット：0x08  
リセット値：0x00

出力に設定されたチャンネル

7	6	5	4	3	2	1	0
予約済み	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
	rw			rw	rw	rw	

ビット 7 予約済み

ビット 6:4 **OC1M**：出力比較 1 モード

これらのビットは、OC1 から派生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 のアクティブレベルは CC1P ビットに依存します。

000：凍結 - 出力比較レジスタ TIMx\_CCR1 とカウンタ TIMx\_CNT と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 をイン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。

**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凍結モードから PWM モードに変更されたときのみ、OCiREF のレベルが変化します。詳細については、[セクション 16.5.7 \(129 ページ\)](#) を参照してください。

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シャドウレジスタにロードされます。

**注：** 正しい動作のためには、プリロードレジスタは、タイマが PWM モードのときに有効でなければなりません。これはワンパルスモードの場合（TIMx\_CR1 レジスタの OPM ビットがセットされている場合）は、必須ではありません。

ビット 2 **OC1FE**：出力比較 1 高速イネーブル

このビットは、トリガ入力のイベントから CC 出力への影響の伝播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 TI1FP1 にマップされます。
  - 10 : CC1 チャンネルは入力として設定され、IC1 は TI2FP1 にマップされます。
  - 11 : 予約済み

**注：** **CC1S** ビットは、チャンネルがオフのとき（TIMx\_CCER 1 レジスタの CC1E=0 であり、更新されたとき）のみ、書き込みができます。

入力に設定されたチャンネル

7	6	5	4	3	2	1	0
IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw				rw		rw	

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ために使用される周波数と、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、サンプリングは f <sub>MASTER</sub> で行われます         | 1000 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /8、N=6  |
| 0001 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> 、N=2   | 1001 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /8、N=8  |
| 0010 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> 、N=4   | 1010 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /16、N=5 |
| 0011 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> 、N=8   | 1011 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /16、N=6 |
| 0100 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /2、N=6 | 1100 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /16、N=8 |
| 0101 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /2、N=8 | 1101 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /32、N=5 |
| 0110 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /4、N=6 | 1110 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /32、N=6 |
| 0111 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /4、N=8 | 1111 : f <sub>SAMPLING</sub> =f <sub>MASTER</sub> /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。キャプチャ入力でエッジが検出されるたびにキャプチャが行われます。
  - 01 : イベント 2 回ごとにキャプチャが行われます。
  - 10 : イベント 4 回ごとにキャプチャが行われます。
  - 11 : イベント 8 回ごとにキャプチャが行われます。

**注：** **動作中に IC1PSC が変更された場合、内部イベントカウンタはリセットされません。この場合、次のキャプチャが発生するまで、古い値が使用されます。新しい値を強制的にすぐに有効にするには、CC1E ビットをクリアして、再びセットします。**

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 TI1FP1 にマップされます。
  - 10 : CC1 チャンネルは入力として設定され、IC1 は TI2FP1 にマップされます。
  - 11 : 予約済み

**注：** **CC1S** ビットは、チャンネルがオフのとき（TIMx\_CCER 1 レジスタの CC1E=0 であり、更新されたとき）のみ、書き込みができます。

16.7.10 キャプチャ／比較モードレジスタ 2（TIMx\_CCMR2）

注： これらのビットの使用方法の詳細については、[キャプチャ／比較モードレジスタ 1（TIMx\\_CCMR1）](#)を参照してください。

アドレスオフセット：0x09  
リセット値：0x00

出力に設定されたチャネル

7	6	5	4	3	2	1	0
予約済み	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]	
	rw			rw	rw	rw	

ビット 7 予約済み

ビット 6:4 **OC2M**：出力比較 2 モード

ビット 3 **OC2PE**：出力比較 2 プリロードイネーブル

ビット 2 **OC2FE**：出力比較 2 高速イネーブル

ビット 1:0 **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 TI2FP2 にマップされます。

10：CC2 チャネルは入力として設定され、IC2 は TI1FP2 にマップされます。

11：予約済み

注： **CC2S** ビットは、チャネルがオフ（TIMx\_CCER1 レジスタの CC2E=0）のときにのみ書き込みができます。



入力に設定されたチャネル

7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]	
rw				rw		rw	

- ビット 7:4 **IC2F** : 入力キャプチャ 2 フィルタ
- ビット 3:2 **IC2PSC** : 入力キャプチャ 2 プリスケアラ
- ビット 1:0 **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 TI2FP2 にマップされます。
- 10 : CC2 チャネルは入力として設定され、IC2 は TI1FP2 にマップされます。
- 11 : 予約済み
- 注：** **CC2S** ビットは、チャネルがオフ (TIMx\_CCER1 レジスタの CC2E=0) のときにのみ書き込みができます。

16.7.11 キャプチャ／比較イネーブルレジスタ 1（TIMx\_CCER1）

アドレスオフセット : 0x0A

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	CC2P		CC2E	予約済み	CC1P		CC1E
	rw		rw		rw		rw

- ビット 6:7 予約済み
- 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 2:3 予約済み
- ビット 1 **CC1P** : キャプチャ／比較 1 出力極性
- CC1 チャ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 チャネルがキャプチャ機能のための入力として設定されている場合** (図 51 を参照) :
- 0 : キャプチャは TI1F または TI2F の立ち上がりエッジで行われます。
- 1 : キャプチャは TI1F または TI2F の立ち下がりエッジで行われます。
- ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル
- CC1 チャ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 チャネルが入力として設定されている場合 :**
- この場合、このビットは、入力キャプチャ／比較レジスタ 1 (TIMx\_CCR1) でカウンタ値のキャプチャが可能かどうかを決めます。
- 0 : キャプチャは無効です。
- 1 : キャプチャは有効です。

## 16.7.12 カウンタハイ (TIMx\_CNTRH)

アドレスオフセット : 0x0B

リセット値 : 0x00

7	6	5	4	3	2	1	0
CNT[15:8]							
rw							

ビット 7:0 CNT[15:8] : カウンタ値 (MSB)

## 16.7.13 カウンタロー (TIMx\_CNTRL)

アドレスオフセット : 0x0C

リセット値 : 0x00

7	6	5	4	3	2	1	0
CNT[7:0]							
rw							

ビット 7:0 CNT[7:0] : カウンタ値 (LSB)

## 16.7.14 プリスケアラレジスタ (TIMx\_PSCR)

アドレスオフセット : 0x0D

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						PSC[2:0]	
						rw	

ビット 7:3 予約済み

ビット 2:0 PSC[2:0] : プリスケアラ値

プリスケアラ値は、CK\_PSC クロック周波数を分周します。  
カウンタのクロック周波数  $f_{CK\_CNT}$  は、 $f_{CK\_PSC} / 2^{(PSC[2:0])}$  に等しくなります。PSC[7:3] は、ハードウェアによって 0 に固定されています。  
PSCR 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です。(TIMx\_EGR レジスタの UG ビットを通じてカウンタがクリアされる場合を含みます)。  
つまり新しいプリスケアラ値が取り入れられるためには、更新イベントが生成されなければなりません。

## 16.7.15 自動再ロードレジスタハイ (TIMx\_ARRH)

アドレスオフセット : 0x0E

リセット値 : 0xFF

7	6	5	4	3	2	1	0
ARR[15:8]							
rw							

ビット 7:0 **ARR[15:8]** : 自動再ロード値 (MSB)

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 16.3 : TIMx タイムベースユニット \(102 ページ\)](#) を参照してください。

自動再ロード値が 0 のときには、カウンタはブロックされます。

## 16.7.16 自動再ロードレジスタロー (TIMx\_ARRL)

アドレスオフセット : 0x0F

リセット値 : 0xFF

7	6	5	4	3	2	1	0
ARR[7:0]							
rw							

ビット 7:0 **ARR[7:0]** : 自動再ロード値 (LSB)

## 16.7.17 キャプチャ／比較レジスタ 1 ハイ (TIMx\_CCR1H)

アドレスオフセット : 0x10

リセット値 : 0x00

7	6	5	4	3	2	1	0
CCR1[15:8]							
rw							

ビット 7:0 **CCR1[15:8]** : キャプチャ／比較 1 の値 (MSB)

**CC1 チャンネルが出力として設定されている場合 (TIMx\_CCMR1 レジスタの CC1S ビット) :**

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 (TIMx\_CCMR1 レジスタの CC1S ビット) :**

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。この場合は、読み出し専用です。

### 16.7.18 キャプチャ／比較レジスタ 1 ロー（TIMx\_CCR1L）

アドレスオフセット：0x11

リセット値：0x00

7	6	5	4	3	2	1	0
CCR1[7:0]							
rw							

ビット 7:0 **CCR1[7:0]**：キャプチャ／比較 1 の値（LSB）

### 16.7.19 キャプチャ／比較レジスタ 2 ハイ（TIMx\_CCR2H）

アドレスオフセット：0x12

リセット値：0x00

7	6	5	4	3	2	1	0
CCR2[15:8]							
rw							

ビット 7:0 **CCR2[15:8]**：キャプチャ／比較 2 の値（MSB）

**CC2 チャンネルが出力として設定されている場合（TIMx\_CCMR2 レジスタの CC2S ビット）：**

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（TIMx\_CCMR2 レジスタの CC2S ビット）：**

CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。

### 16.7.20 キャプチャ／比較レジスタ 2 ロー（TIMx\_CCR2L）

アドレスオフセット：0x13

リセット値：0x00

7	6	5	4	3	2	1	0
CCR2[7:0]							
rw							

ビット 7:0 **CCR2[7:0]**：キャプチャ／比較値（LSB）



## 16.7.21 ブレークレジスタ（TIMx\_BKR）

アドレスオフセット：0x14

リセット値：0x00

7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	予約済み	OSSI	LOCK	
rw	rw	rw	rw		rw	rw	

### ビット7 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 出力は無効か、強制的にアイドル状態になります。

1：OC 出力は、それぞれのイネーブルビット（TIMx\_CCERx レジスタの CCxE ビット）がセットされている場合に有効です。

詳細については、OC イネーブルの説明を参照してください（表 34：ブレーク機能を持つ OCx チャンネルの出力制御ビット（154 ページ））。

### ビット6 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 がプログラムされると（TIMx\_BKR レジスタの LOCK ビット）、変更できなくなります。

### ビット5 BKP：ブレーク極性

0：ブレーク入力 BKIN はアクティブラーです。

1：ブレーク入力 BKIN はアクティブハイです。

**注：** このビットは、LOCK レベル1 がプログラムされると（TIMx\_BKR レジスタの LOCK ビット）、変更できなくなります。

### ビット4 BKE：ブレークイネーブル

0：ブレーク入力（BKIN）は無効です。

1：ブレーク入力（BKIN）は有効です。

**注：** このビットは、LOCK レベル1 がプログラムされると（TIMx\_BKR レジスタの LOCK ビット）、変更できなくなります。

### ビット3 予約済み

### ビット2 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 イネーブルの説明を参照してください（表 34：ブレーク機能を持つ OCx チャンネルの出力制御ビット（154 ページ））。

0：インアクティブのとき、OCx 出力は無効です（OCx イネーブル出力信号 = 0）。

1：インアクティブのとき、CCxE=1 になると、OCx 出力は、まず、強制的にアイドルレベルになります（OC イネーブル出力信号 = 1）。

**注：** このビットは、LOCK レベル2 がプログラムされると（TIMx\_BKR レジスタの LOCK ビット）、変更できなくなります。

ビット 1:0 LOCK：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\_OISR レジスタの OISx ビットと TIMx\_BKR レジスタの BKE/BKP/AOE ビットは書き込めなくなります。

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\_CCERx レジスタの CCxP 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\_CCMRx 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。TIMx\_BKR レジスタに書き込みが行われると、それらの内容は次のリセットまで凍結されます。

注： AOE、BKP、BKE、および OSSI は LOCK 設定に応じて書き込みがロックされるので、TIMx\_BKR レジスタへの最初の書き込みアクセスで、これらのすべてを設定する必要があります。

表 34. ブレーク機能を持つ OCx チャネルの出力制御ビット

制御ビット			OCx/OCx_EN 出力の状態
MOE ビット	OSSI ビット	CCxE ビット	
1	X	0	出力無効（タイマによって駆動されない） OCx=CCxP、OCx_EN=0
		1	OCx= OCxREF + 極性（OCxREF xor CCxP） OCx_EN=1
0	0	0	出力無効（タイマによって駆動されない） OCx=OISx、OCx_EN=0
	0	1	
	1	0	オフ状態（インアクティブ状態で出力有効） OCx=OISx、OCx_EN=1
	1	1	

注： OCx チャネルに接続されている外部入出力ピンの状態は、OCx チャネルの状態と GPIO レジスタに依存します。



## 16.7.22 出力アイドル状態レジスタ（TIMx\_OISR）

アドレスオフセット：0x15

リセット値：0x00

7	6	5	4	3	2	1	0
予約済み					OIS2	予約済み	OIS1
					rw		rw

ビット 7:3 予約済み

ビット 2 **OIS2**：出力アイドル状態 2（OC2 出力）

OIS1 ビットを参照

ビット 1 予約済み

ビット 0 **OIS1**：出力アイドル状態 1（OC1 出力）

0：MOE=0 のとき、OC1=0

1：MOE=0 のとき、OC1=1

**注：** このビットは、**LOCK レベル 1、2、または 3 がプログラムされると（TIMx\_BKR レジスタの LOCK ビット）、変更できなくなります。**

## 16.7.23 TIMx レジスタマップとリセット値

TIM2 と TIM3 のベースアドレスについては、データシートを参照してください。

表 35. TIMx レジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	TIMx_CR1 リセット値	ARPE 0	CMS1 0	CMS0 0	DIR 0	OPM 0	URS 0	UDIS 0	CEN 0
0x01	TIMx_CR2 リセット値	TI1S 0	MMS2 0	MMS1 0	MMS0 0	- 0	- 0	- 0	- 0
0x02	TIMx_SMCR リセット値	MSM 0	TS2 0	TS1 0	TS0 0	- 0	SMS2 0	SMS1 0	SMS0 0
0x03	TIMx_ETR リセット値	ETP 0	ECE 0	ETPS1 0	ETPS0 0	EFT3 0	EFT2 0	EFT1 0	EFT0 0
0x04	TIMx_IER リセット値	BIE 0	TIE 0	- 0	- 0	- 0	CC2IE 0	CC1IE 0	UIE 0
0x05	TIMx_SR1 リセット値	BIF 0	TIF 0	- 0	- 0	- 0	CC2IF 0	CC1IF 0	UIF 0
0x06	TIMx_SR2 リセット値	- 0	- 0	- 0	- 0	- 0	CC2OF 0	CC1OF 0	- 0
0x07	TIMx_EGR リセット値	BG 0	TG 0	- 0	- 0	- 0	CC2G 0	CC1G 0	UG 0
0x08	TIMx_CCMR1 (出力モード) リセット値	- 0	OC1M2 0	OC1M1 0	OC1M0 0	OC1PE 0	OC1FE 0	CC1S1 0	CC1S0 0
	TIMx_CCMR1 (入力モード) リセット値	IC1F3 0	IC1F2 0	IC1F1 0	IC1F0 0	IC1PSC1 0	IC1PSC0 0	CC1S1 0	CC1S0 0

表 35. TIMx レジスタマップ（続き）

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x09	TIMx_CCMR2 (出力モード)	- 0	OC2M2 0	OC2M1 0	OC2M0 0	OC2PE 0	OC2FE 0	CC2S1 0	CC2S0 0
	TIMx_CCMR2 (入力モード)	IC2F3 0	IC2F2 0	IC2F1 0	IC2F0 0	IC2PSC1 0	IC2PSC0 0	CC2S1 0	CC2S0 0
0x0A	TIMx_CCER1	- 0	- 0	CC2P 0	CC2E 0	- 0	- 0	CC1P 0	CC1E 0
0x0B	TIMx_CNTRH	CNT15 0	CNT14 0	CNT13 0	CNT12 0	CNT11 0	CNT10 0	CNT9 0	CNT8 0
0x0C	TIMx_CNTRL	CNT7 0	CNT6 0	CNT5 0	CNT4 0	CNT3 0	CNT2 0	CNT1 0	CNT0 0
0x0D	TIMx_PSCR	- 0	- 0	- 0	- 0	- 0	PSC2 0	PSC1 0	PSC0 0
0x0E	TIMx_ARRH	ARR15 1	ARR14 1	ARR13 1	ARR12 1	ARR11 1	ARR10 1	ARR9 1	ARR8 1
0x0F	TIMx_ARRL	ARR7 1	ARR6 1	ARR5 1	ARR4 1	ARR3 1	ARR2 1	ARR1 1	ARR0 1
0x10	TIMx_CCR1H	CCR115 0	CCR114 0	CCR113 0	CCR112 0	CCR111 0	CCR110 0	CCR19 0	CCR18 0
0x11	TIMx_CCR1L	CCR17 0	CCR16 0	CCR15 0	CCR14 0	CCR13 0	CCR12 0	CCR11 0	CCR10 0
0x12	TIMx_CCR2H	CCR215 0	CCR214 0	CCR213 0	CCR212 0	CCR211 0	CCR210 0	CCR29 0	CCR28 0
0x13	TIMx_CCR2L	CCR27 0	CCR26 0	CCR25 0	CCR24 0	CCR23 0	CCR22 0	CCR21 0	CCR20 0
0x14	TIMx_BKR	MOE 0	AOE 0	BKP 0	BKE 0	OSSR 0	OSSI 0	LOCK 0	LOCK 0
0x15	TIMx_OISR	- 0	0	0	0	0	OIS2 0	0	OIS1 0

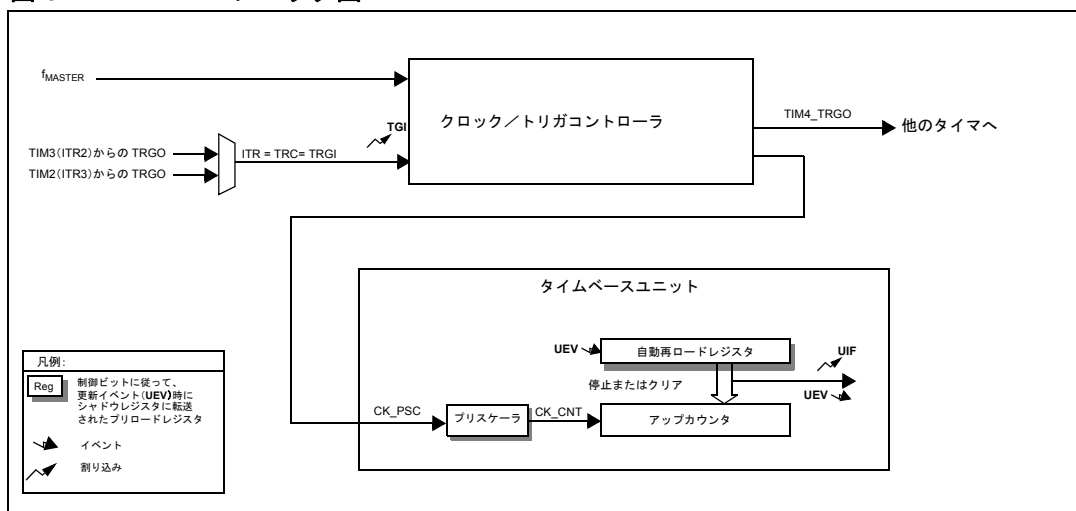
## 17 8 ビット基本タイマ (TIM4)

### 17.1 概要

このタイマは、プログラマブルプリスケアラによって駆動される 8 ビット自動再ロードアップカウンタで構成されています。タイマオーバーフロー時の割り込み生成機能を備え、タイムベース生成に使用できます。

タイマ機能の一般的な説明については、[セクション 16.3 \(102 ページ\)](#) を参照してください。

図 64. TIM4 ブロック図



### 17.2 TIM4 の主な機能

主な機能は、次のとおりです。

- 8 ビットのアップカウント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 32768 までの任意の 2 のべき乗で（動作中に）分周できる 4 ビットプログラマブルプリスケアラ
- 割り込みの生成
  - カウンタ更新時：カウンタオーバーフロー
  - トリガ入力時

### 17.3 TIM4 割り込み

タイマには 2 つの割り込みリクエストソースがあります。

- 更新割り込み（オーバーフロー、カウンタ初期化）
- トリガ入力

## 17.4 TIM4 クロックの選択

このタイマのクロックソースは、内部クロック ( $f_{\text{MASTER}}$ ) です。カウンタクロック CK\_CNT を駆動するプリスケアラに供給される CK\_PSC クロックに直接接続されます。

### プリスケアラ

プリスケアラは、次のように実装されています。

- TIM4 プリスケアラは、4 ビットレジスタ (TIM4\_PSCR レジスタ) を介して制御される 16 ビットのカウンタに基づきます。この制御レジスタはバッファされるので、動作中に変更できます。カウンタクロック周波数を 1 から 32768 までの任意の 2 のべき乗で分周することができます。カウンタクロック周波数は、次の式で計算されます。

$$f_{\text{CK\_CNT}} = f_{\text{CK\_PSC}} / 2^{(\text{PSCR}[3:0])}$$

プリスケアラの値は、プリロードレジスタを介してロードされます。シャドウレジスタは、使用される現在値を含み、LS バイトが書き込まれるとロードされます。

TIM4\_PSCR レジスタの読み出し動作はプリロードレジスタにアクセスするので、読み出しに特別な注意は必要ありません。

## 17.5 TIM4 レジスタ

### 17.5.1 制御レジスタ 1 (TIM4\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x00

7	6	5	4	3	2	1	0
ARPE	予約済み			OPM	URS	UDIS	CEN
rw				rw	rw	rw	rw

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0 : TIM4\_ARR レジスタは、プリロードレジスタを通じてバッファされません。直接書き込むことができます。
- 1 : TIM4\_ARR レジスタは、プリロードレジスタを通じてバッファされます。

ビット 6:4 予約済み

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 ビットをクリアします)。

ビット 2 **URS** : 更新リクエストソース

- 0 : 有効なときには、レジスタが更新されると (カウンタオーバーフロー)、更新割り込みリクエストが送信されます。
- 1 : 有効なときには、カウンタがオーバーフロー/アンダーフローしたときのみ、更新割り込みリクエストが送信されます。

ビット 1 **UDIS** : 更新ディセーブル

- 0 : カウンタオーバーフローが発生したとき、またはソフトウェア更新が生成されたとき、更新イベントが生成されます。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 : 更新イベントは生成されず、シャドウレジスタは値を維持します (ARR、PSC)。UG ビットがセットされている場合、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。



17.5.2 制御レジスタ 2（TIM4\_CR2）

アドレスオフセット：0x01

リセット値：0x00

7	6	5	4	3	2	1	0
予約済み	MMS[2:0]			予約済み			
	rw						

ビット 7 予約済み

ビット 6:4 **MMS**：マスタモード選択

これらのビットは、他のタイマと同期するためにマスタモードで送信する情報を選択します（TRGO）。組み合わせは、次のとおりです。

000：リセット - TIM4\_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た場合（クロック／トリガモードコントローラがトリガリセットモードに設定されているとき）、TRGO 信号は実際のリセットから遅れて発生します。

001：イネーブル - カウンタイネーブル信号がトリガ出力（TRGO）として使用されます。複数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ために使用されます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和（OR）によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ／スレーブモードが選択されている場合を除いて、TRGO 出力には遅延が存在します（TIM4\_SMCR レジスタの MSM ビットの説明を参照）。

010：更新 - 更新イベントがトリガ出力（TRGO）として選択されます。

011：予約済み

100：予約済み

101：予約済み

111：予約済み

ビット 3:0 予約済み



## リセット値：0x00

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。

## 17.5.4 割り込みイネーブルレジスタ (TIM4\_IER)

アドレスオフセット : 0x03

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	TIE	予約済み					UIE
	rw						rw

ビット 7 予約済み

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5:1 予約済み

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

## 17.5.5 ステータスレジスタ 1 (TIM4\_SR1)

アドレスオフセット : 0x04

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	TIF	予約済み					UIF
	rc_w0						rc_w0

ビット 7 予約済み

ビット 6 **TIF** : トリガ割り込みフラグ

このフラグは、トリガイイベント時に (TRGI 信号でアクティブエッジが検出されたとき、またはゲートモードが選択されている場合は両方のエッジが検出されたとき) 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 : トリガイイベントは発生していません。

1 : トリガ割り込みがペンディング中です。

ビット 5:1 予約済み

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みがペンディング中です。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。

- TIM4\_CR1レジスタの UDIS = 0 の場合、オーバーフローが発生したとき。
- TIM4\_CR1 レジスタの URS=0 かつ UDIS=0 の場合、TIM4\_EGR レジスタの UG ビットを使用して、ソフトウェアによって CNT が再初期化されたとき。

## 17.5.6 イベント生成レジスタ (TIM4\_EGR)

アドレスオフセット : 0x05

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	TG	予約済み				UG	
	W					W	

ビット 7 予約済み

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 : TIM4\_SR1 レジスタの TIF フラグがセットされます。TIE ビットによって有効にされている場合、割り込みが生成されます。

ビット 5:1 予約済み

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。プリスケラカウンタもクリアされることに注意してください。

## 17.5.7 カウンタ (TIM4\_CNTR)

アドレスオフセット : 0x06

リセット値 : 0x00

7	6	5	4	3	2	1	0
CNT[7:0]							
rw							

ビット 7:0 CNT[7:0] : カウンタ値

## 17.5.8 プリスケラレジスタ (TIM4\_PSCR)

アドレスオフセット : 0x07

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み				PSC[3:0]			
				rw			

ビット 7:4 予約済み

ビット 3:0 **PSC[3:0]** : プリスケラ値

プリスケラ値は、CK\_PSC クロック周波数を分周します。

カウンタのクロック周波数  $f_{CK\_CNT}$  は、 $f_{CK\_PSC} / 2^{(PSC[3:0])}$  に等しくなります。

PSC は、更新イベントごとにアクティブプリスケラレジスタにロードされる値を含みます (TIM4\_EGR レジスタの UG ビットを通じてカウンタがクリアされる場合を含みます)。

新しいプリスケラ値が取り入れられるように、更新イベントが生成されなければなりません。

## 17.5.9 自動再ロードレジスタ (TIM4\_ARR)

アドレスオフセット : 0x08

リセット値 : 0xFF

7	6	5	4	3	2	1	0
ARR[7:0]							
rw							

ビット 7:0 ARR[7:0] : 自動再ロード値

## 17.5.10 TIM4 レジスタマップとリセット値

ベースアドレスについては、データシートを参照してください。

表 36. TIM4 レジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	TIM4_CR1 リセット値	ARPE 0	- 0	- 0	- 0	OPM 0	URS 0	UDIS 0	CEN 0
0x01	TIM4_CR2 リセット値	- 0	MMS2 0	MMS1 0	MMS0 0	- 0	- 0	- 0	- 0
0x02	TIM4_SMCR リセット値	MSM 0	TS2 0	TS1 0	TS0 0	- 0	SMS2 0	SMS1 0	SMS0 0
0x03	TIM4_IER リセット値	- 0	TIE 0	- 0	- 0	- 0	- 0	- 0	UIE 0
0x04	TIM4_SR1 リセット値	- 0	TIF 0	- 0	- 0	- 0	- 0	- 0	UIF 0
0x05	TIM4_EGR リセット値	- 0	TG 0	- 0	- 0	- 0	- 0	- 0	UG 0
0x06	TIM4_CNTR リセット値	CNT7 0	CNT6 0	CNT5 0	CNT4 0	CNT3 0	CNT2 0	CNT1 0	CNT0 0
0x07	TIM4_PSCR リセット値	- 0	- 0	- 0	- 0	PSC3 0	PSC2 0	PSC1 0	PSC0 0
0x08	TIM4_ARR リセット値	ARR7 1	ARR6 1	ARR5 1	ARR4 1	ARR3 1	ARR2 1	ARR1 1	ARR0 1

## 18 I<sup>2</sup>C (Inter-Integrated Circuit) インタフェース

### 18.1 概要

I<sup>2</sup>C (Inter-Integrated Circuit) バスインタフェースは、マイクロコントローラとシリアル I<sup>2</sup>C バスとの間のインタフェースとして機能します。このインタフェースはマルチマスタ機能を提供し、I<sup>2</sup>C バス固有のすべてのシーケンシング、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。また、標準モードと高速モードをサポートします。

### 18.2 I<sup>2</sup>C の主な機能

- パラレルバス/I<sup>2</sup>C 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
- I<sup>2</sup>C マスタの機能：
  - － クロック生成
  - － START 信号、STOP 信号の生成
- I<sup>2</sup>C スレーブの機能：
  - － プログラマブルな I<sup>2</sup>C アドレス検出
  - － ストップビット検出
  - － 2 つのスレーブアドレスを確認応答する I<sup>2</sup>C デュアルアドレス機能
- 7 ビット/10 ビットアドレスリングおよび同報 (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)
- 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
  - － バイト終わり送信フラグ
  - － I<sup>2</sup>C 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
  - － クロック伸長が無効な場合のオーバーラン/アンダーラン
- 3 種類の割り込み：
  - － 通信割り込み
  - － エラー条件割り込み
  - － HALT からのウェイクアップ割り込み
- ウェイクアップ機能：
  - － MCU は、スレーブモードでのアドレス検出時に低電力モードからウェイクアップします。
- オプションのクロック伸長

### 18.3 I<sup>2</sup>C の概要

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割り込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン（SDA）とクロックピン（SCL）によって I<sup>2</sup>C 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 I<sup>2</sup>C バスと接続できます。

#### 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トまたは STOP 生成が発生し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

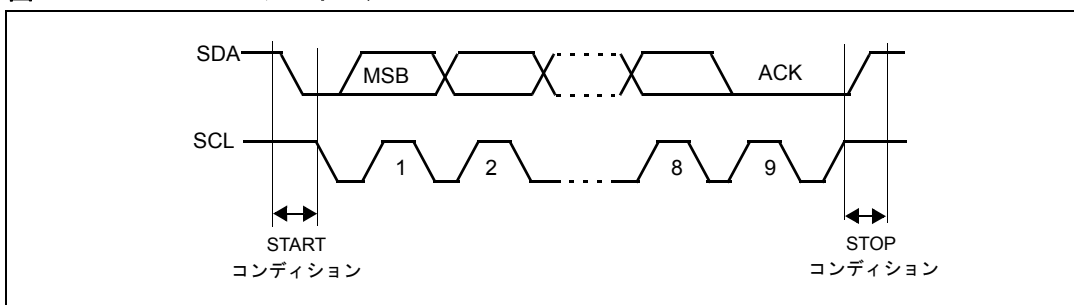
マスタモードでは、I<sup>2</sup>C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで始まり、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

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。次の図を参照してください。

図 65. I<sup>2</sup>C バスプロトコル

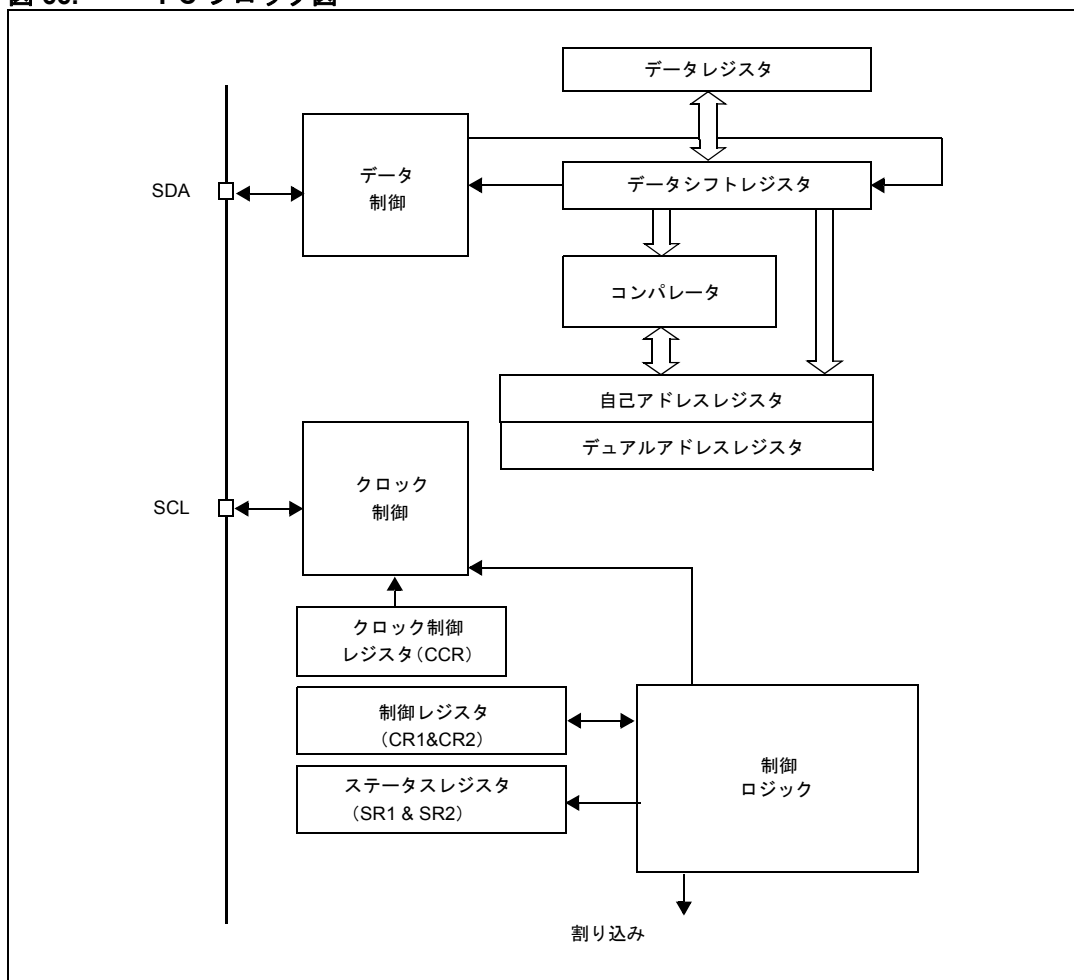


確認応答は、ソフトウェアによって有効／無効にできます。I<sup>2</sup>C インタフェースアドレス（デュアルアドレッシングの 7 ビット／10 ビットや同報アドレス）は、ソフトウェアによって選択できます。

I<sup>2</sup>C インタフェースのブロック図を次に示します。



図 66. I<sup>2</sup>C ブロック図



### 18.4 I<sup>2</sup>C 機能の説明

デフォルトでは、I<sup>2</sup>C 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。

#### 18.4.1 I<sup>2</sup>C スレーブモード

正しいタイミングを生成するには、ペリフェラル入カクロックを I2C\_FREQR レジスタにプログラミングする必要があります。ペリフェラル入カクロックは、少なくとも次の周波数以上でなければなりません。

- 標準モード : 1 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 (OAR1L 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無視し、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成し、8 ビットのスレーブアドレスを待ちます。

**アドレス一致 :** インタフェースは次の順序で動作します。

- ACK ビットがセットされている場合、確認応答パルスを生成します。
- ADDR ビットがハードウェアによってセットされ、ITEVTEN ビットがセットされている場合は割り込みが生成されます。ENDUAL=1 の場合、ソフトウェアは、DUALF ビットを読み出して、どのスレーブアドレスが確認応答されたかをチェックする必要があります。

10 ビットモードでは、アドレスシーケンスの受信後、スレーブは常にレシーバモードになります。スレーブがトランスミッタモードに入るのは、反復 START コンディションの後、一致するアドレスビットを持ち、最下位ビットがセットされた (11110xx1) ヘッダシーケンスを受信したときです。

TRA ビットは、スレーブがレシーバモードかトランスミッタモードかを示します。

## スレーブトランスミッタ

アドレス受信に続いて、ADDR をクリアした後、スレーブは、内部シフトレジスタを介して、DR レジスタから SDA ラインにバイトを送信します。

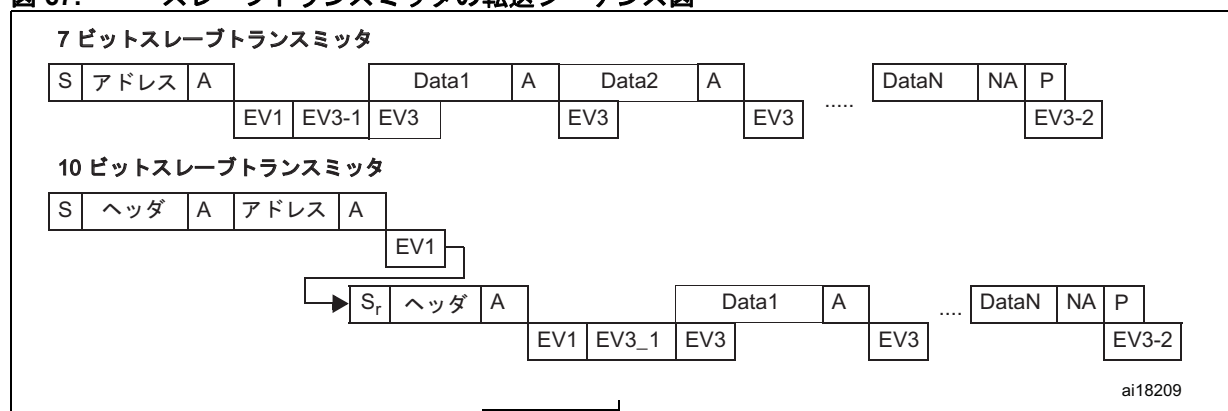
スレーブは、ADDR がクリアされ、DR が送信データで満杯になるまで、SCL のローレベルを伸長します（次の図の転送シーケンスの EV1 EV3 を参照）。

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、ITEVTEN および ITBUFEN ビットがセットされている場合は割り込みが生成されます。

TXE がセットされ、次のデータ送信の終わりまでに DR レジスタにデータが書き込まれなかった場合は、BTF ビットがセットされ、インタフェースは、SR1 レジスタの読み出しと、それに続く 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 67. スレーブトランスミッタの転送シーケンス図



1. 凡例：  
**S** = START, **S<sub>r</sub>** = 反復 START, **P** = STOP, **A** = 確認応答, **NA** = 否定応答, **EVx** = イベント (ITEVTEN=1 の場合は割り込み付き)  
**EV1** : ADDR=1, SR1 の読み出しとそれに続く SR3 の読み出しによってクリア  
**EV3-1** : TXE=1, シフトレジスタエンプティ、データレジスタエンプティ、Data1 を DR に書き込み  
**EV3** : TXE=1, シフトレジスタエンプティ、データレジスタエンプティ、DR への書き込みによってクリア  
**EV3-2** : AF=1, AF は、SR2 レジスタの AF ビットに"0"を書き込むことによってクリア
2. EV1 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。
3. 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。

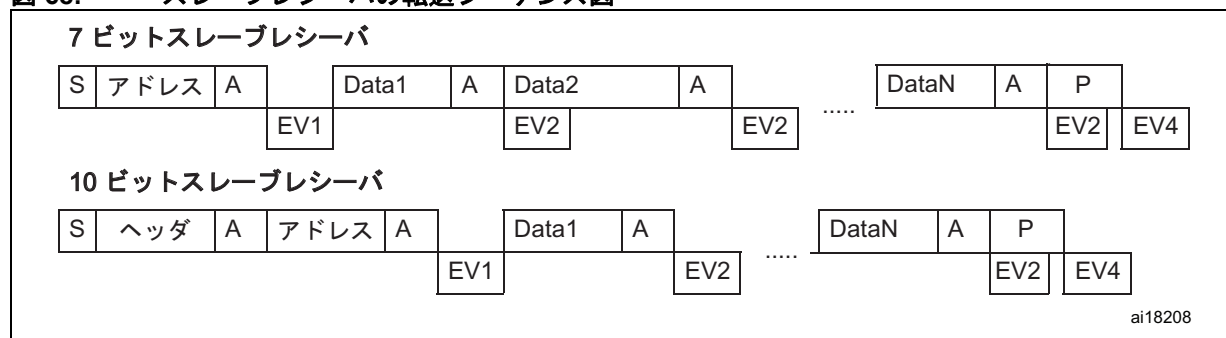
## スレーブレシーバ

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ものを順に生成します。

- 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、ITEVTEN および ITBUFEN ビットがセットされている場合は割り込みが生成されます。

RXNE がセットされ、次のデータ受信の終わりまでに 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。インタフェースは、SR1 レジスタの読み出しと、それに続く DR レジスタの読み出しによって BTF がクリアされるまで、SCL のローレベルを伸長して待ちます（図 68 を参照）。

図 68. スレーブレシーバの転送シーケンス図



- 凡例：  
**S** = START、**S<sub>r</sub>** = 反復 START、**P** = STOP、**A** = 確認応答、**NA** = 否定応答、**EVx** = イベント (ITEVTEN=1 の場合は割り込み付き)  
**EV1** : ADDR=1、SR1 の読み出しとそれに続く SR3 の読み出しによってクリア  
**EV2** : RXNE=1、DR レジスタの読み出しによってクリア  
**EV4** : STOPF=1、SR1 レジスタの読み出しとそれに続く CR2 レジスタへの書き込みによってクリア
- EV1 イベントは、対応するソフトウェアシーケンスの終わりまで SCL のローレベルを伸長します。
- 次のバイト受信の終わりまでにソフトウェアシーケンスが完了しない場合、EV2 イベントは SCL のローレベルを伸長します。
- SR1 レジスタの内容をチェックした後、ユーザは、セットされていることが分かったフラグごとに、完全なクリアシーケンスを実行してください。したがって、ADDR フラグと STOPF フラグの場合、I2C 割り込みルーチンの中に次のシーケンスを記述することを推奨します。

```

READ SR1
if (ADDR == 1) {READ SR1; READ SR3}
if (STOPF == 1) {READ SR1; WRITE CR2}

```

この目的は、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ことです。

- 注 8 (ページ) 184 も参照してください。

### 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、このコンディションを検出して、STOPF ビットをセットし、ITEVTEN ビットがセットされている場合は割り込みを生成します。

STOPF は、SR1 レジスタの読み出しと、それに続く CR2 レジスタへの書き込みによってクリアされます (図 68 : スレーブレシーバの転送シーケンス図の EV4 を参照)。

## 18.4.2 I<sup>2</sup>C マスタモード

マスタモードでは、I<sup>2</sup>C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START 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に、I2C\_FREQR レジスタにペリフェラル入力クロックをプログラミングします。
- 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 レジスタをプログラミングして、ペリフェラルを有効にします。
- I2C\_CR2 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 : 1 MHz
- 高速モード : 4 MHz

## 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、

- SB ビットがハードウェアによってセットされ、ITEVTEN ビットがセットされている場合は割り込みが生成されます。

次に、マスタは、SR1 レジスタの読み出しと、それに続く DR レジスタへのスレーブアドレスの書き込みを待ちます (図 69 と 図 70)。

## 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、下記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、ITEVTEN ビットがセットされている場合は割り込みが生成されます。

次に、マスタは、SR1 レジスタの読み出しと、それに続く DR レジスタへの 2 番目のアドレスバイトの書き込みを待ちます (図 69 と 図 70 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、ITEVTEN ビットがセットされている場合は割り込みが生成されます。次に、マスタは、SR1 レジスタの読み出しと、それに続く SR3 レジスタの読み出しを待ちます (図 69 と 図 70 の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、ITEVTEN ビットがセットされている場合は割り込みが生成されます。

次に、マスタは、SR1 レジスタの読み出しと、それに続く SR3 レジスタの読み出しを待ちます (図 69 と 図 70 の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 がクリアされた後、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。

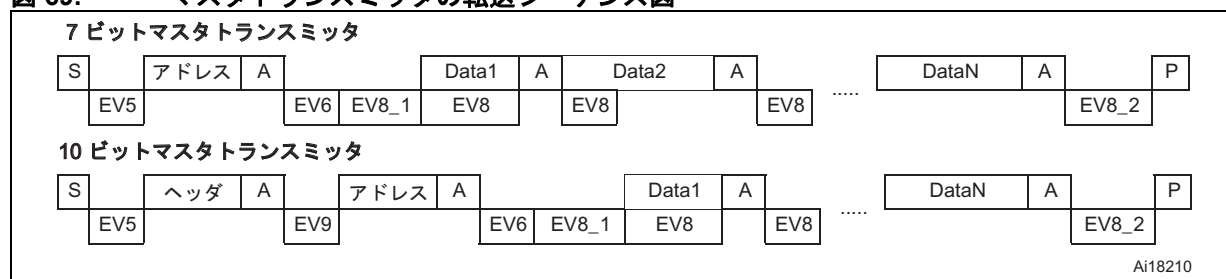
マスタは、最初のデータバイトが DR レジスタに書き込まれるまで待ちます（図 69 転送シーケンスの EV8\_1 を参照）。

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、ITEVTEN および ITBUFEN ビットがセットされている場合は割り込みが生成されます。

TXE がセットされていて、次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なかった場合、BTF がセットされ、インタフェースは、SR1 レジスタからの読み出しと、それに続く 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 69. マスタトランスミッタの転送シーケンス図



- 凡例：  
**S** = START、**S<sub>r</sub>** = 反復 START、**P** = STOP、**A** = 確認応答、**NA** = 否定応答、  
**EV<sub>x</sub>** = イベント（ITEVTEN=1 の場合は割り込み付き）  
**EV5** : SB=1、SR1 レジスタの読み出しと、それに続く DR レジスタへのアドレスの書き込みによってクリア。  
**EV6** : ADDR=1、SR1 レジスタの読み出しと、それに続く SR3 の読み出しによってクリア。  
**EV8\_1** : TXE=1、シフトレジスタエンプティ、データレジスタエンプティ、DR レジスタへの書き込み。  
**EV8** : TXE=1、シフトレジスタノットエンプティ、データレジスタエンプティ、DR レジスタへの書き込みによってクリア。  
**EV8\_2** : TXE=1、BTF = 1、STOP リクエストのプログラミング。TXE と BTF は、STOP コンディションでハードウェアによってクリア。  
**EV9** : ADD10=1、SR1 レジスタの読み出しと、それに続く DR レジスタへの書き込みによってクリア。[注 8 \(ページ 184\)](#) も参照してください。
- 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 のローレベルを伸長します。
- 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 のローレベルを伸長します。

## 通信の終了

最後のバイトが DR レジスタに書き込まれた後、STOP ビットがソフトウェアによってセットされて STOP コンディションを生成します（図 69 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。



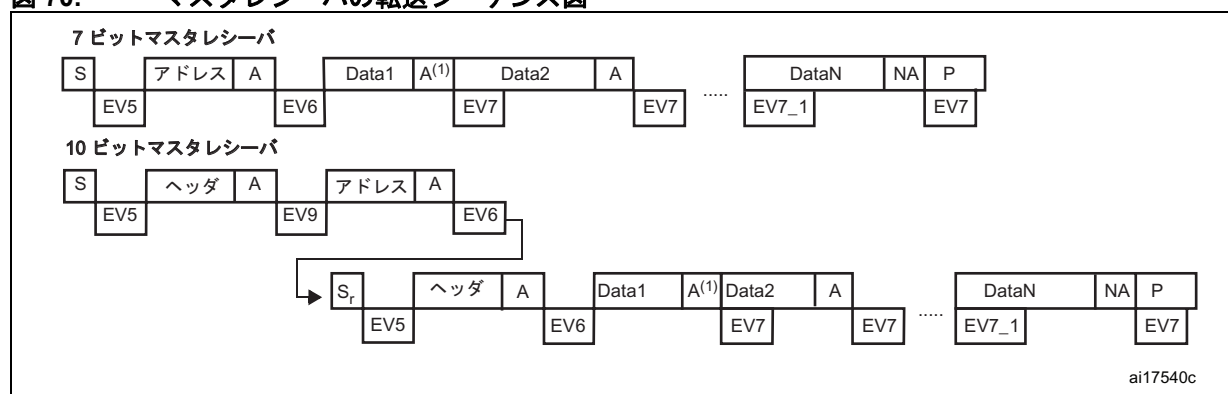
## マスタレシーバ

アドレス送信に続いて ADDR がクリアされると、I<sup>2</sup>C 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、インタフェースは下記のことを順に生成します。

- 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 ビットがセットされ、ITEVTEN および ITBUFEN ビットがセットされている場合は割り込みが生成されます (図 69 と 図 70)。

RXNE ビットがセットされ、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 ビットがハードウェアによってセットされ、インタフェースは、I2C\_SR1 レジスタの読み出しと、それに続く I2C\_DR レジスタの読み出しによって BTF ビット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 70. マスタレシーバの転送シーケンス図



- 凡例:  
**S** = START、**S<sub>r</sub>** = 反復 START、**P** = STOP、**A** = 確認応答、**NA** = 否定応答、  
**EVx** = イベント (ITEVTEN=1 の場合は割り込み付き)  
**EV5**: SB=1、SR1 レジスタの読み出しと、それに続く DR レジスタへの書き込みによってクリア。  
**EV6**: ADDR=1、SR1 レジスタの読み出しと、それに続く SR3 の読み出しによってクリア。10 ビットマスタレシーバモードでは、このシーケンスの後に続けて、CR2 に START = 1 を書き込んでください。  
**EV6\_1**: 関連するフラグイベントはありません。1 バイト受信にのみ使用されます。ADDR をクリアした後、ACK=0 と STOP=1 をプログラミングします。  
**EV7**: RXNE=1、DR レジスタの読み出しによってクリアされます。  
**EV7\_1**: RXNE=1、DR レジスタの読み出しによってクリア、ACK=0 と STOP リクエストのプログラミング。  
**EV9**: ADD10=1、SR1 レジスタの読み出しと、それに続く DR レジスタへの書き込みによってクリア。
- 単一バイトが受信された場合、それは NA です。
- 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 のローレベルを伸長します。
- 次のバイト受信の終わりまでにソフトウェアシーケンスが完了しない場合、EV7 イベントは SCL のローレベルを伸長します。
- 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。



### 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。

- 最後の受信データバイトの後に否定応答パルス（NACK）を生成するには、最後から 2 番目の RXNE イベントの後で最後から 2 番目のデータバイトの読み出し直後に、ACK ビットをクリアする必要があります。
- STOP/RESTART コンディションを生成するには、ソフトウェアは、最後から 2 番目の RXNE イベントの後で最後から 2 番目のデータバイトの読み出し直後に、STOP/START ビットをセットする必要があります。
- 単一バイトが受信された場合、確認応答の無効化と STOP コンディションの生成は、EV6 の直後（EV6-1 では、ADDR がクリアされた直後）に行われます。

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）。

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。

次の条件を実現するには、以下の手順に従う必要があります。

- 最後のデータ受信が終わる前に、ACK ビットが遅れずにローにセットされる。
- 最後のデータ受信の後、補助データの受信なしに STOP ビットがハイにセットされる。

#### 2 バイト受信の場合：

- ADDR = 1 になるまで待ちます（ADDR フラグがクリアされるまで SCL のローレベルを伸長します）。
- ACK をローにセットし、POS をハイにセットします。
- ADDR フラグをクリアします。
- BTF = 1 になるまで待ちます（DR にデータ、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 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 N を読み出します。

## 18.4.3 エラー条件

以下は、通信障害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、I<sup>2</sup>C 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、

- BERR ビットがセットされ、ITERREN ビットがセットされている場合は割り込みが生成されません。
- スレーブモードの場合：データは破棄され、ライン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、ラインはハードウェアによって解放されます。
- マスタモードの場合：ラインは解放されず、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、ソフトウェア次第です。

### 確認応答障害 (AF)

このエラーは、インタフェースが否定応答ビットを検出したときに発生します。このとき、

- AF ビットがセットされ、ITERREN ビットがセットされている場合は割り込みが生成されます。
- NACK を受信したトランスミッタは、通信をリセットする必要があります。
  - スレーブの場合：ライン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ロスト (ARLO)

このエラーは、I<sup>2</sup>C インタフェースがアービトレーションロスト条件を検出したときに発生します。このとき、

- ARLO ビットがハードウェアによってセットされます (ITERREN ビットがセットされている場合、割り込みが生成されます)。
- I<sup>2</sup>C 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)。
- I<sup>2</sup>C 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、マスタからの反復 START 後に確認応答できます。
- ラインはハードウェアによって解放されます。

### オーバーラン／アンダーランエラー (OVR)

スレーブモードでオーバーランエラーが発生するのは、クロック伸長が無効であり、I<sup>2</sup>C 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、トランスミッタで最後の受信バイトを再送信してください。

スレープモードでアンダーランエラーが発生するのは、クロック伸長が無効であり、I<sup>2</sup>C インタフェースがデータを送信しているときです。次のバイト用のクロックが到着するまでに、まだ DR は次のバイトで更新されていません (TXE=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。
- 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、I<sup>2</sup>C 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捨てなければなりません。

## 18.4.4 SDA/SCL ライン制御

- クロック伸長が有効な場合：
  - － トランスミッタモード：TXE=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 SR1 を読み出し、その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 SR1 を読み出し、データレジスタ内のバイトを読み出すか CR2 に書き込むのを待ちます (バッファとシフトレジスタは、いずれも満杯)。
- スレープ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：TXE=1 で、次のバイトを送信するまでに DR への書き込みが行われていない場合。同じバイトが再送信されます。
  - － 書き込み衝突は管理されません。

## 18.5 I<sup>2</sup>C 低電力モード

表 37. 低電力モードでの I<sup>2</sup>C インタフェースの動作

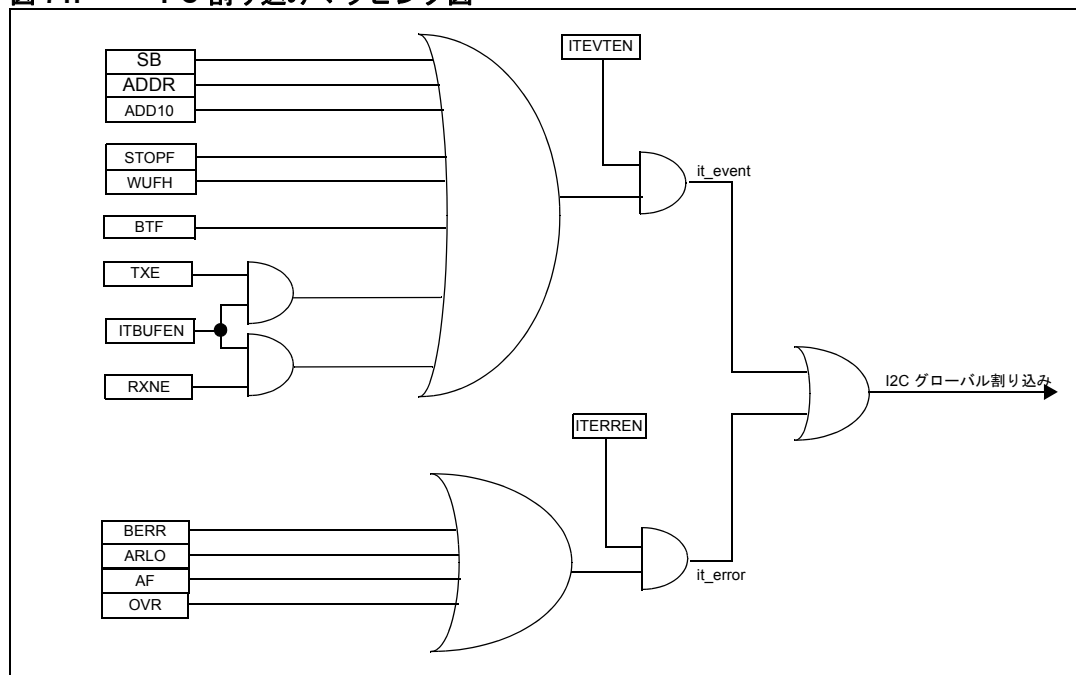
モード	説明
WAIT	I <sup>2</sup> C インタフェースには影響ありません。 I <sup>2</sup> C 割り込みによって、デバイスは WAIT モードから抜けます。
HALT	<b>スレープモードの場合：</b> 設定レジスタを除いて、通信はリセットされます。デバイスはスレープモードです。 ITEVTEN = 1 であり、許容ヘッダも含めてアドレスが一致した場合、HALT からのウェイクアップ割り込みが生成されます。 一致したアドレスは HALT モードでは確認応答されないため、マスタは、CPU がウェイクアップしたときにこれを再送信して確認応答を受信する必要があります。 NOSTRETCH = 0 の場合、SCLH は、WUFH がソフトウェアによってクリアされるまで、HALT モードでの確認応答パルス後に伸長されます。 どのフラグも、CPU をウェイクアップするアドレスによってはセットされません。 <b>マスタモードの場合：</b> 通信は、CPU がウェイクアップされるまで凍結されます。ITEVTEN=1 であり、HALT 命令がある場合、HALT からのウェイクアップフラグと割り込みが生成されます。 <b>注：</b> <b>通信中に HALT モードに入ることは禁止されています。</b>

## 18.6 I<sup>2</sup>C 割り込み

表 38. I<sup>2</sup>C 割り込みリクエスト

割り込みイベント	イベント フラグ	イネーブル 制御 ビット	WAIT からの 復帰	HALT からの 復帰
START ビット送信（マスタ）	SB	ITEVTEN	あり	なし
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			
HALT からのウェイクアップ	WUFH	ITEVTEN		あり
受信バッファノットエンプティ	RXNE	ITEVTEN および ITBUFEN		なし
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			

図 71. I<sup>2</sup>C 割り込みマッピング図



18.7 I<sup>2</sup>C レジスタ

18.7.1 制御レジスタ 1 (I2C\_CR1)

アドレスオフセット : 0x00  
リセット値 : 0x00

7	6	5	4	3	2	1	0
NOSTRETCH	ENGCG	予約済み					PE
rw	rw						rw

ビット 7 **NOSTRETCH** : クロック伸長ディセーブル (スレーブモード)  
このビットは、スレーブモードで ADDR または BTF フラグがセットされているとき、ソフトウェアによってリセットされるまで、クロック伸長を無効にするために使用されます。  
0 : クロック伸長は有効です。  
1 : クロック伸長は無効です。

ビット 6 **ENGCG** : 同報イネーブル  
0 : 同報は無効です。アドレス 0x00 は NACK されます。  
1 : 同報は有効です。アドレス 0x00 は ACK されます。

ビット 5:1 予約済み

ビット 0 **PE** : ペリフェラルイネーブル  
0 : ペリフェラルは無効です。  
1 : ペリフェラルは有効です。対応する入出力は代替機能として選択されます。

**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。

## 18.7.2 制御レジスタ 2 (I2C\_CR2)

アドレスオフセット : 0x01

リセット値 : 0x00

7	6	5	4	3	2	1	0
SWRST	予約済み			POS	ACK	STOP	START
rw				rw	rw	rw	rw

ビット 7 **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 :** このビットは、バス上で STOP コンディションが検出されていないときに BUSY ビットが "1" に設定された場合に備えて使用できます。

ビット 6:4 予約済み

ビット 3 **POS** : 確認応答位置 (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。

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。

**注 :** POS ビットは、マスタモードの 2 バイト受信設定でのみ使用する必要があります。また、マスタスレーブ (173 ページ) 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 2 **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 1 **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 は生成されません。

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 は生成されません。

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 0 **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。

**注 :** STOP または START がセットされると、制御ビットがハードウェアによってクリアされるまで、ユーザは I2C\_CR2 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、2 番目の STOP または START リクエストが発生することがあります。

18.7.3 周波数レジスタ (I2C\_FREQR)

アドレスオフセット : 0x02  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み		FREQ[5:0]					
		rw					

ビット 7:6 予約済み

ビット 5:0 **FREQ[5:0]** : ペリフェラルクロック周波数 <sup>(1)</sup>

入力クロック周波数は正しいタイミングを生成するようにプログラミングする必要があります。  
許容範囲は 1 MHz~24 MHz です。  
000000 : 設定禁止  
000001 : 1 MHz  
000010 : 2 MHz  
...  
011000 : 24 MHz  
さらに高い値 : 設定禁止

1. I<sup>2</sup>C バスタイミングを守るための最小ペリフェラルクロック周波数は、標準モードでは 1MHz、高速モードでは 4MHz 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。



## 18.7.4 自己アドレスレジスタ 1 LSB (I2C\_OAR1L)

アドレスオフセット : 0x03  
リセット値 : 0x00

7	6	5	4	3	2	1	0
ADD1[7:1]							ADD1[0]
rw							rw

- ビット 7:1 **ADD1[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 **ADD1[0]** : インタフェースアドレス  
7 ビットアドレスモード : 無視  
10 ビットアドレスモード : アドレスのビット 0

## 18.7.5 自己アドレスレジスタ 1 MSB (I2C\_OAR1H)

アドレスオフセット : 0x04  
リセット値 : 0x00

7	6	5	4	3	2	1	0
ADDMODE	ADDCONF	予約済み			ADD1[9:8]		予約済み
rw	rw				rw		

- ビット 7 **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 6 **ADDCONF** : アドレスモード設定  
このビットは、ソフトウェアによってセットされる必要があります (常に"1"として書き込まれます)。
- ビット 5:3 予約済み
- ビット 2:1 **ADD1[9:8]** : インタフェースアドレス  
10 ビットアドレスモード : アドレスのビット 9:8
- ビット 0 予約済み

## 18.7.6 自己アドレスレジスタ 2 (I2C\_OAR2)

アドレスオフセット : 0x05  
リセット値 : 0x00

7	6	5	4	3	2	1	0
ADD2[7:1]							ENDUAL
rw							rw

- 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。

## 18.7.7 データレジスタ (I2C\_DR)

アドレスオフセット : 0x06  
リセット値 : 0x00

7	6	5	4	3	2	1	0
DR[7:0]							
rw							

ビット 7:0 **DR[7:0]** : データレジスタ <sup>(1)(2)(3)</sup>

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 : DR 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。送信が開始されて (TXE=1) からすぐに次の送信データを DR に書き込むことで、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 DR にコピーされます (RXNE=1)。次のデータが受信される (RXNE=1) 前に DR を読み出すことで、連続的な送信ストリームを維持できます。

1. スレーブモードでは、アドレスは DR にコピーされません。
2. 書き込み衝突は管理されません (TXE=0 の場合、DR は書き込み可能です)。
3. 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 DR にコピーされないため、読み出しできません。

## 18.7.8 ステータスレジスタ 1 (I2C\_SR1)

アドレスオフセット : 0x07  
リセット値 : 0x00

7	6	5	4	3	2	1	0
TXE	RXNE	予約済み	STOPF	ADD10	BTF	ADDR	SB
r	r		r	r	r	r	r

ビット 7 **TXE** : データレジスタエンプティ (トランスマッタ) <sup>(1)</sup>

- 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。
- ソフトウェアで DR レジスタに書き込みを行ってクリアするか、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。

**注 :** 送信中に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 はクリアできません。いずれの場合も、DR 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) <sup>(2)(3)</sup>

- 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。

**注 :** BTF ビットがセットされているときにデータを読み出しても、DR レジスタはまだ満杯なので、RXNE はクリアできません。

ビット 5 予約済み

**ビット 4 STOPF** : STOP 検出 (スレーブモード) <sup>(4)(5)</sup>

- 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 CR2 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。

**ビット 3 ADD10** : 10 ビットヘッダの送信 (マスタモード) <sup>(6)</sup>

- 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。

**ビット 2 BTF** : バイト転送終了 <sup>(7)(8)</sup>

- 0 : データバイト転送は完了していません。
- 1 : データバイト転送が正常に終了しました。

- NOSTRETCH=0 の場合、次の条件でハードウェアによってセットされます。
  - 受信時、新しいバイト (ACK パルスを含む) が受信され、DR レジスタがまだ読み出されていないとき (RXNE=1)。
  - 送信時、新しいバイトを送信すべきときに、DR レジスタへの書き込みがまだ行われていないとき (TXE=1)。
- ソフトウェアで SR1 を読み出し、続いて DR レジスタへの読み書きを行ったときにクリアされます。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。

**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) <sup>(8)(9)</sup>

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 SR3 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、あるいは同報または SMBus 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、それぞれ有効である場合)。

**– 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) <sup>(8)</sup>

- 0 : START コンディションなし。
- 1 : START コンディションが生成されました。

- START コンディションが生成されたときにセットされます。
- ソフトウェアで SR1 レジスタを読み出し、続いて DR レジスタへの書き込みを行うとクリアされます。または、PE=0 のときにハードウェアによってクリアされます。

1. ACK パルスの後で DR がシフトレジスタにコピーされると、割り込みが生成されます。NACK が受信された場合、コピーは行われず、TXE はセットされません。
2. ACK パルスの後でシフトレジスタが DR にコピーされると、割り込みが生成されます。
3. ARLO イベントの場合、RXNE はセットされません。
4. STOPF ビットは、NACK 受信後にはセットされません。
5. STOPF がセットされた後、完全なクリアシーケンス (SR1 を読み出し、続いて CR2 への書き込み) を実行することを推奨します。図 68 : スレーブシーバの転送シーケンス図 (ページ) 170 を参照してください。

6. ADD10 ビットは、NACK 受信後にはセットされません。
7. BTF ビットは、NACK 受信後または ARLO イベントの場合にはセットされません。
8. タイミング制約があるため、標準モードにおいて CCR が 9 未満（つまり、ペリフェラルクロックが 2 MHz 未満）、 $f_{\text{MASTER}} = f_{\text{CPU}}$ 、およびイベント割り込みが無効である場合、以下の手順に従う必要があります。  
フラグのクリアシーケンスにおける各動作間に少なくとも 5 サイクルを挿入するため、リセットシーケンスを変更します。たとえば、 $f_{\text{MASTER}} = f_{\text{CPU}} = 1 \text{ MHz}$  のとき、以下のシーケンスを使用して SB ビットをポーリングします。  
\_label\_wait: BTJF I2C\_SR1,SB,\_label\_wait  
NOP;  
NOP;  
NOP;  
NOP;  
NOP  
LD I2C\_DR, A ;実行後、SB ビットはクリアされます。
9. スレープモードでは、ADDR がセットされた後、完全なクリアシーケンス（SR1 を読み出し、続いて SR3 を読み出す）を実行することを推奨します。[図 68 : スレープレシーバの転送シーケンス](#) [図 \(ページ 170\)](#) を参照してください。

18.7.9 ステータスレジスタ 2 (I2C\_SR2)

アドレスオフセット : 0x08  
リセット値 : 0x00

7		6		5		4		3		2		1		0	
予約済み		WUFH		予約済み		OVR		AF		ARLO		BERR			
		rc_w0				rc_w0		rc_w0		rc_w0		rc_w0			

- ビット 7:6 予約済み
- ビット 5 **WUFH** : HALT からのウェイクアップ
- 0 : HALT モードからのウェイクアップなし。
- 1 : HALT モード（スレープモード）で 7 ビットアドレスまたはヘッダが一致、あるいはマスタモードのときに HALT になりました。
- 注：** このビットは、スレープモードで（HALT モード中に）非同期にセットされます。ITEVTEN = 1 の場合にのみセットされます。
- 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 4 予約済み
- ビット 3 **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。
- スレープモードで 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 2 **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 1 **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。

- ビット 0 **BERR** : バスエラー
- 0 : START または STOP コンディションの誤配置はありません。
  - 1 : START または STOP コンディションが誤配置されました。
- バイト転送中に SCL がハイである間、SDA の立ち上がりまたは立ち下がりエッジが有効でない位置に発生していることをインタフェースが検出したとき、ハードウェアによってセットされます。
  - ソフトウェアによって 0 を書き込んでクリアするか、PE=0 のときにハードウェアによってクリアされます。

18.7.10 ステータスレジスタ 3 (I2C\_SR3)

アドレスオフセット : 0x09  
リセット値 : 0x00

7	6	5	4	3	2	1	0
DUALF	予約済み		GENCALL	予約済み	TRA	BUSY	MSL
r			r		r	r	r

- 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 6:5 予約済み

- ビット 4 **GENCALL** : 同報ヘッダ (スレーブモード)
- 0 : 同報ヘッダは受信されていません。
  - 1 : ENGC=1 のとき、同報ヘッダが受信されました。
- 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

- ビット 2 **TRA** : トランスマッタ/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終わりで、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)、反復 START コンディション、バスアービトレーションロスト (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。
- このビットは、バス上で通信が進行中であることを示します。この情報は、インタフェースが無効である (PE=0) のときにも更新されます。



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

- 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 の読み出し後に ADDR フラグがセットされた場合でも、I2C\_SR1 の読み出し後に I2C\_SR3 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\_SR3 を読み出す必要があります。

18.7.11   割り込みレジスタ (I2C\_ITR)

アドレスオフセット : 0x0A  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み					ITBUFEN	ITEVTEN	ITERREN
					rw	rw	rw

- ビット 7:3 予約済み
- ビット 2 **ITBUFEN** : バッファ割り込みイネーブル
- 0 : TXE = 1 または RXNE = 1 で、割り込みは生成されません。
  - 1 : TXE = 1 または RXNE = 1 で、イベント割り込みを生成します。
- ビット 1 **ITEVTEN** : イベント割り込みイネーブル
- 0 : イベント割り込みは無効です。
  - 1 : イベント割り込みは有効です。
- この割り込みは、以下の場合に生成されます。
- SB = 1 (マスタ)
  - ADDR = 1 (マスタ/スレーブ)
  - ADD10= 1 (マスタ)
  - STOPF = 1 (スレーブ)
  - BTF = 1 (ただし、TXE または RXNE イベントが発生してしないとき)
  - 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
  - WUFH = 1 (HALT からウェイクアップするための非同期割り込み)
- ビット 0 **ITERREN** : エラー割り込みイネーブル
- 0 : エラー割り込みは無効です。
  - 1 : エラー割り込みは有効です。
- この割り込みは、以下の場合に生成されます。
- BERR = 1
  - ARLO = 1
  - AF = 1
  - OVR = 1

## 18.7.12 クロック制御レジスタロー (I2C\_CCRL)

アドレスオフセット : 0x02

リセット値 : 0x0B

7	6	5	4	3	2	1	0
CCR[7:0]							
rw							

ビット 7:0 **CCR[7:0]** : クロック制御レジスタ (マスタモード)

マスタモードにおける SCLH クロックを制御します。

– 標準モード :

$$\text{Period(I2C)} = 2 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{high}} = \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{low}} = \text{CCR} * t_{\text{MASTER}}$$

– 高速モード :

DUTY = 0 の場合 :

$$\text{Period(I2C)} = 3 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{high}} = \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{low}} = 2 * \text{CCR} * t_{\text{MASTER}}$$

DUTY = 1 の場合 : (400 kHz に到達するには)

$$\text{Period(I2C)} = 25 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{high}} = 9 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{low}} = 16 * \text{CCR} * t_{\text{MASTER}}$$

注 :  $t_{\text{CK}} = 1 / f_{\text{MASTER}}$ 。  $f_{\text{MASTER}}$  は、クロック制御レジスタを使用して設定されたペリフェラルへの入力クロックです。

最小許容値は 04h です。ただし、FAST DUTY モードでは最小許容値は 0x01 になります。

$t_{\text{high}} = t_{\text{r(SCL)}} + t_{\text{w(SCLH)}}$ 。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。

$t_{\text{low}} = t_{\text{r(SCL)}} + t_{\text{w(SCLL)}}$ 。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、 $f_{\text{SCL}} = 1 / (t_{\text{high}} + t_{\text{low}})$

これらのタイミングは、フィルタを通さないものです。



### 18.7.13 クロック制御レジスタハイ (I2C\_CCRH)

アドレスオフセット : 0x0C

リセット値 : 0x00

7	6	5	4	3	2	1	0
F/S	DUTY	予約済み		CCR[11:8]			
rw	rw			rw			

ビット 7 **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 6 **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 5:4 予約済み

ビット 3:0 **CCR[11:8]** : 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<sup>(1)</sup>

マスタモードにおける SCLH クロックを制御します。

– 標準モード :

$$\text{Period(I2C)} = 2 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{high}} = \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{low}} = \text{CCR} * t_{\text{MASTER}}$$

– 高速モード :

DUTY = 0 の場合 :

$$\text{Period(I2C)} = 3 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{high}} = \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{low}} = 2 * \text{CCR} * t_{\text{MASTER}}$$

DUTY = 1 の場合 : (400 kHz に到達するには)

$$\text{Period(I2C)} = 25 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{high}} = 9 * \text{CCR} * t_{\text{MASTER}}$$

$$t_{\text{low}} = 16 * \text{CCR} * t_{\text{MASTER}}$$

例 : 標準モードで 100 kHz の SCL 周波数を生成するには、

FREQR = 08、 $t_{\text{MASTER}} = 125 \text{ ns}$  の場合、CCR には 0x28 をプログラミングする必要があります。

( $0x28 \Leftrightarrow 40 \times 125 \text{ ns} = 5000 \text{ ns}$ )

**注 :**  $t_{\text{high}} = t_{\text{r(SCL)}} + t_{\text{w(SCLH)}}$ 。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。

これらのタイミングは、フィルタを通さないものです。

1. 表 39 : SCL 周波数に対する I2C\_CCR 値 ( $t_{\text{MASTER}} = 16 \text{ MHz}$ ) (189 ページ) を参照してください。

**注 :** CCR レジスタは、I<sup>2</sup>C が無効 (PE=0) の場合にのみ設定する必要があります。

400 kHz で高速クロックを生成するには、 $f_{\text{MASTER}} = 10 \text{ MHz}$  の倍数にする必要があります。

100 kHz で標準クロックを生成するには、 $f_{\text{MASTER}} \geq 1 \text{ MHz}$  にする必要があります。

表 39. SCL 周波数に対する I2C\_CCR 値 (f<sub>MASTER</sub> = 16 MHz<sup>(1)</sup>)

I <sup>2</sup> C 速度	I <sup>2</sup> C 周波数 (f <sub>SCL</sub> ) Hz	f <sub>MASTER</sub> = 16 MHz			
		実際 (Hz)	誤差 (%)	I2C_CCR (h)	デューティ サイクルビット
高速 モード	400000	410256.41	2.56	D	0
	370000	380952.38	2.96	E	0
	350000	355555.56	1.59	F	0
	320000	320000	0	2	1
	300000	313725.49	4.57	11	0
	270000	280701.75	3.96	13	0
	250000	253968.25	1.59	15	0
	220000	222222.22	1.01	18	0
	200000	205128.20	2.56	1A	0
	170000	172043.01	1.20	1F	0
	150000	152380.95	1.59	23	0
	120000	121212.12	1.01	2C	0
標準モード	100000	100000	0	50	影響なし
	50000	50000	0	A0	
	30000	30075.19	0.25	10A	
	20000	20000	0	190	

1. 次の表は、必要な I<sup>2</sup>C SCL ライン周波数を得るために I2C\_CCR レジスタに書き込む値を示します。

18.7.14 TRISE レジスタ (I2C\_TRISER)

アドレスオフセット : 0x0D

リセット値 : 0x02

7	6	5	4	3	2	1	0
予約済み		TRISE[5:0]					
		rw					

ビット 7:6 予約済み

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。

例：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\_FREQR レジスタの値が 08h で、t<sub>MASTER</sub> = 125 ns である場合、TRISE[5:0] ビットには 0x09 をプログラミングする必要があります。

(1000 ns / 125 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<sub>HIGH</sub> 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] の設定は、I2C が無効なとき (PE=0) にのみ行う必要があります。**

18.7.15 I<sup>2</sup>C レジスタマップとリセット値

表 40. I<sup>2</sup>C レジスタマップ

アドレス オフ セット	レジスタ名	7	6	5	4	3	2	1	0
0x00	I2C_CR1 リセット値	NO STRETCH 0	ENGCG 0	- 0	- 0	- 0	- 0	- 0	PE 0
0x01	I2C_CR2 リセット値	SWRST 0	- 0	- 0	- 0	POS 0	ACK 0	STOP 0	START 0
0x02	I2C_FREQR リセット値	- 0	- 0	FREQ5 0	FREQ4 0	FREQ3 0	FREQ2 0	FREQ1 0	FREQ0 0
0x03	I2C_OAR1L リセット値	ADD1[7:1] 0							ADD1[0] 0
0x04	I2C_OAR1H リセット値	ADDMODE 0	ADDCONF 0	- 0	- 0	- 0	ADD1[9:8] 0		- 0
0x05	I2C_OAR2 リセット値	ADD2[7:1] 0							ENDUAL 0
0x06	I2C_DR リセット値	DR[7:0] 0							
0x07	I2C_SR1 リセット値	TXE 0	RXNE 0	- 0	STOPF 0	ADD10 0	BTF 0	ADDR 0	SB 0
0x08	I2C_SR2 リセット値	- 0	- 0	WUFH 0	- 0	OVR 0	AF 0	ARLO 0	BERR 0
0x09	I2C_SR3 リセット値	DUALF 0	- 0	- 0	GENCALL 0	- 0	TRA 0	BUSY 0	MSL 0
0x0A	I2C_ITR リセット値	- 0	- 0	- 0	- 0	- 0	ITBUFEN 0	ITEVTEN 0	ITERREN 0
0x0B	I2C_CCRL リセット値	CCR[7:0] 0							
0x0C	I2C_CCRH リセット値	FS 0	DUTY 0	- 0	- 0	CCR[11:8] 0			
0x0D	I2C_TRISER リセット値	- 0	- 0	TRISE[5:0] 0					

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

## 19 シリアルペリフェラルインタフェース (SPI)

### 19.1 概要

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。

### 19.2 SPI の主な機能

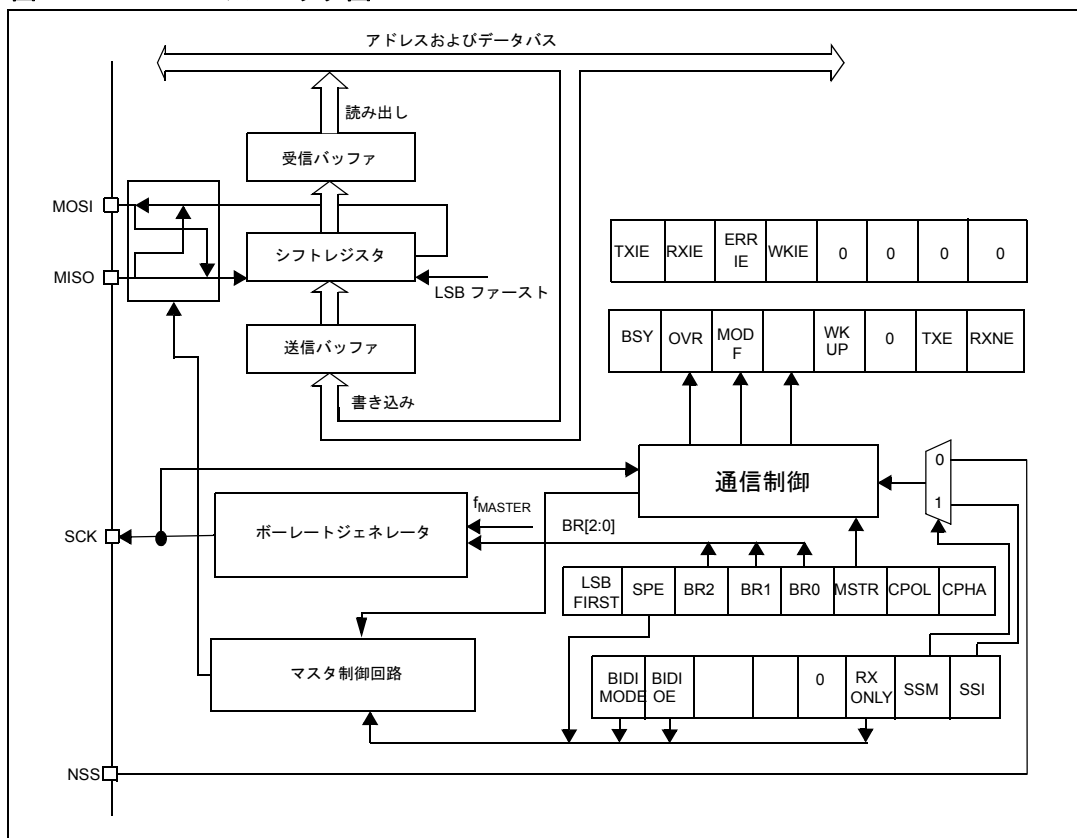
- 全二重同期転送 (3 線式)
- 2 本のラインで単方向同期転送 (双方向データラインの有無にかかわらず)
- マスタまたはスレーブ動作
- 8 つのマスタモード周波数 (最大:  $f_{\text{MASTER}}/2$ )
- スレーブモード周波数 (最大:  $f_{\text{MASTER}}/2$ )
- 高速通信 - 最大 SPI 速度: 8 MHz
- マスタとスレーブの両方に対するハードウェア／ソフトウェアによる NSS 管理
- クロックの極性と位相をプログラミング可能
- データ順序をプログラミング可能 (MSB ファースト／LSB ファーストのシフト)
- 割り込み機能付きの専用の送受信フラグ
- SPI バスビジステータスフラグ
- 割り込み機能付きのマスタモード障害およびオーバーランフラグ
- ウェイクアップ機能:  
MCU は、全二重／半二重の送信専用モードにおいて、低電力モードからウェイクアップします

## 19.3 SPI 機能の説明

### 19.3.1 概要

SPI のブロック図を [図 72](#) に示します。

図 72. SPI のブロック図



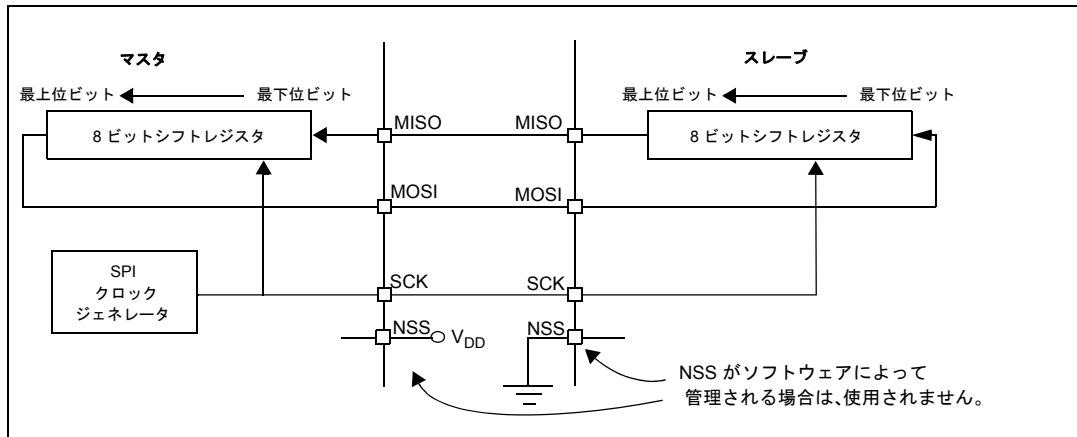
SPI は 4 本のピンで外部デバイスに接続されます。

- MISO : Master In / Slave Out データ (ポート C7)。このピンは、マスタ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 : Master Out / Slave In データ (ポート C6)。このピンは、マスタモードではデータの送信に、スレーブモードではデータの受信に使用できます。
- SCK : SPI マスタではシリアルクロック出力 (ポート C5) に、SPI スレーブではシリアルクロック入力に使用されます。
- NSS : スレーブ選択 (ポート E5)。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出力ポートによって駆動できます。SPI がマスタモードに設定され (MSTR ビット=1)、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ルト状態に入ります。MSTR ビットは自動的にリセットされ、デバイスはスレーブモードに設定されます ([セクション 19.3.8 : エラーフラグ \(206 ページ\)](#) 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 73 に示します。

**注：** SPI を高速モードで使用する場合、想定されるバス速度に到達できるように、SPI 出力が接続される入出力は、高速スロープ出力としてプログラムしてください。

図 73. シングルマスタ/シングルスレーブアプリケーション



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の設定は、SPI\_CR2 レジスタのソフトウェアスレーブ管理 (SSM) ビットを使用して行うことができます。

- **ソフトウェア NSS 管理 (SSM = 1) :** この設定では、スレーブ選択情報は、SPI\_CR2 レジスタの内部スレーブ選択 (SSI) ビット値によって内部的に駆動されます。外部 NSS ピンは、依然として他のアプリケーション用途に解放されたままです。
- **ハードウェア NSS 管理 (SSM = 0) :** マスタとしてセットされたデバイスの場合、この設定はマルチマスタ機能を可能にします。スレーブとしてセットされたデバイスの場合、NSS ピンは古典的な NSS 入力として機能します。スレーブは、NSS ラインがローレベルのときに選択され、NSS ラインがハイレベルの場合は選択されません。

**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。

## クロックの位相と極性

CPOL ビットと CPHA ビットを使用することによって、4 つのタイミング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 がリセットされた場合、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされた場合、SCK ピンはハイレベルのアイドル状態になります。

**注：** **SPI セルを有効または無効にするときに SPI クロックピンでのエッジ生成を避けるには、SPI ピンを SPI のアイドル状態レベルで設定するようにします。**

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、最初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、2 番目のクロック遷移でラッチされます。

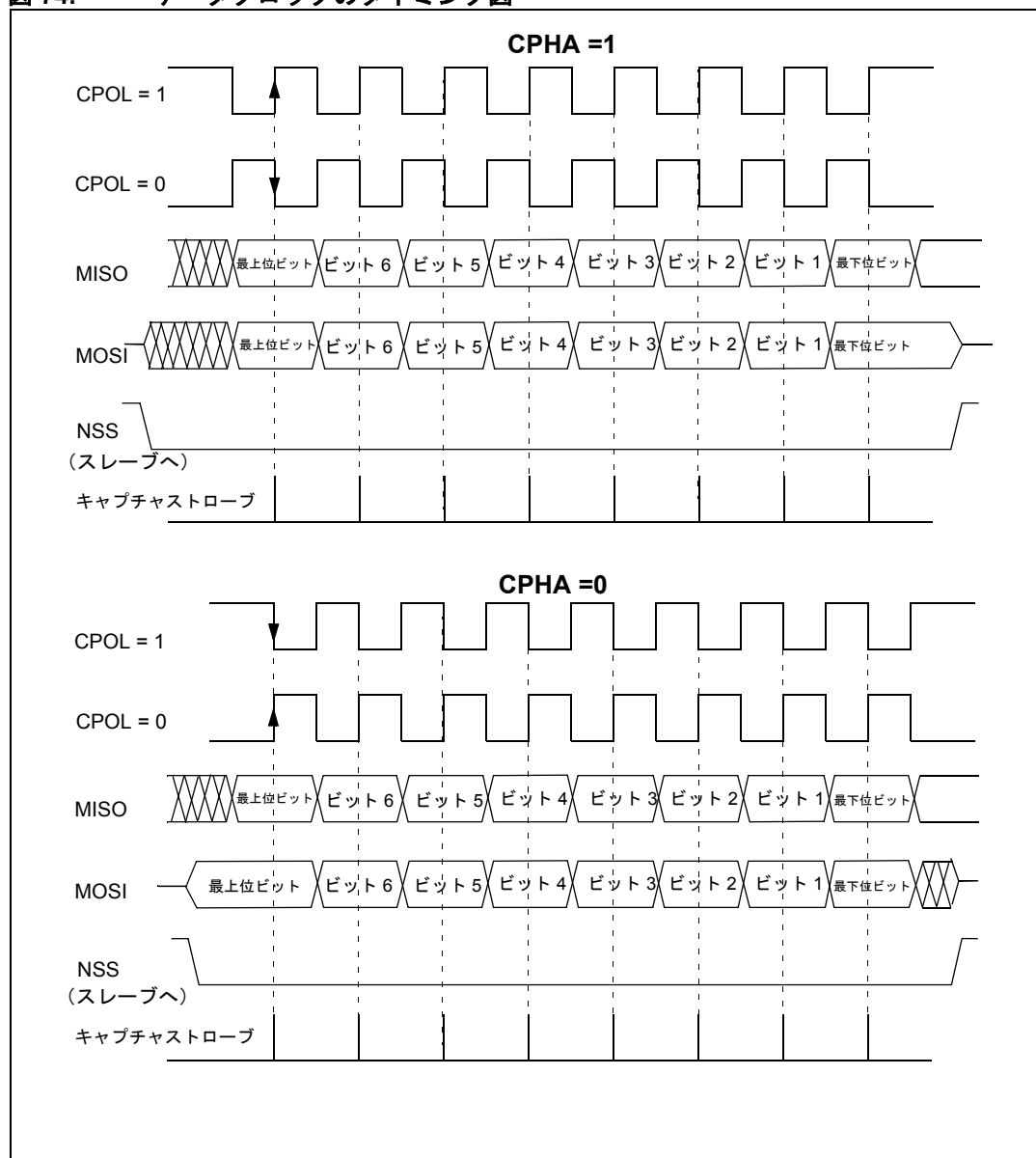
CPOL (クロック極性) ビットと CPHA (クロック位相) ビットの組み合わせによって、データキャプチャのクロックエッジを選択できます。

[図 74](#) 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：**
- 1 **CPOL/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。**
  - 2 **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。**
  - 3 **SCK のアイドル状態は、SPI\_CR1 レジスタで (CPOL=1 の場合は SCK のプルアップ、CPOL=0 の場合は SCK のプルダウンによって) 選択された極性に一致する必要があります。**



図 74. 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。

## 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。

## 19.3.2 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。

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。

1. 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 74 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。
2. 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。
3. ハードウェアモード (スレーブ選択 (NSS) ピンの管理 (193 ページ) を参照) では、バイト送信シーケンス全体を通じて、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\_CR2 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。
4. ピンを代替機能に割り当てるには、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。

**注：** *パラレルマルチスレーブ構造のアプリケーションでは、別個の NSS 信号とスレーブ MISO 出力を接続することによって、対応する GPIO レジスタを正しく設定する必要があります。SPI\_MISO ピンが SPI ペリフェラルによって制御されるのは、NSS 信号が有効であり、デバイスがスレーブとして選択された場合に限られます。NSS 信号がリリースされると、ピンは、GPIO レジスタの設定によってのみ駆動されます。正しく機能させるには、GPIO を割り込みなしの入力プルアップモードに設定する必要があります。この設定には、GPIO\_DDR、GPIO\_CR1、および GPIO\_CR2 レジスタを使用します (セクション 10.8.1 : 代替機能出力 を参照)。*

## 19.3.3 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 ピンに生成されます。

SPI をマスタモードに設定するには、次の手順に従ってください。

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 74 を参照)。
3. 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。
4. ハードウェアモードでは、完全なデータ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。ソフトウェアモードでは、SPI\_CR2 レジスタの SSM ビットと SSI ビットをセットします。
5. MSTR ビットと SPE ビットをセットします (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。

## 19.3.4 SPI の単方向通信設定

SPI は、次の 2 つの設定において単方向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

このモードを有効にするには、SPI\_CR2 レジスタの BDM 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、マスタモードでは MOSI ピン、スレーブモードでは 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\_CR2 レジスタの BDOE ビットによって選択されます。このビットが 1 にセットされた場合、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（BDM=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（BDM=0、RXONLY=0）と似ています。データは送信ピン（マスタモードでは MOSI ピン、スレーブモードでは MISO ピン）で送信され、受信ピン（マスタモードでは MISO ピン、スレーブモードでは 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受信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（マスタモードでは MOSI ピン、スレーブモードでは 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 ビットがリセット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 ビットがリセットされていて）、SCK が動作している限り、SPI は受信を続行します。

## 19.3.5 データの送受信手順

### 受信バッファと送信バッファ

受信の場合、受信されたデータは内部の受信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送信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、受信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送信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（BDM=0 および RXONLY=0）
  - － データが SPI\_DR レジスタ（送信バッファ）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送信バッファから 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（受信バッファ）に同時にロードされます。

- 単方向受信専用モードの場合 (BDM=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 (受信バッファ) に同時にロードされます。
- 双方向モードでの送信時 (BDM=1 および BDOE=1)
  - データが SPI\_DR レジスタ (送信バッファ) 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送信バッファから 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 (BDM=1 および BDOE=0)
  - SPE=1 および BDOE=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 (受信バッファ) 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 (BDM=0 および RXONLY=0)
  - 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送信バッファから 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 (BDM=0 および RXONLY=1)
  - 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 (BDM=1 および BDOE=1)
  - スレーブデバイスがクロック信号を受信し、送信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送信バッファから 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 (BDM=1 および BDOE=0)
  - スレーブデバイスが MISO ピンでクロック信号と、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 (受信バッファ) 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、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

データが送信バッファからシフトレジスタに転送されると、TXE フラグ（送信バッファエンプティ）がセットされます。これは、内部の送信バッファに次のデータをロードする準備ができていることを示します。SPI\_ICR レジスタの TXIE ビットがセットされている場合は、割り込みを生成できます。

**注：** ソフトウェアは、送信バッファへの書き込みを行う前に、TXE 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。そうしないと、以前に送信バッファに書き込まれたデータに上書きされます。

データがシフトレジスタから受信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（受信バッファノットエンプティ）がセットされます。これは、SPI\_DR レジスタからデータを読み出す準備ができていることを示します。SPI\_ICR レジスタの RXIE ビットがセットされている場合は、割り込みを生成できます。RXNE ビットのクリアは、SPI\_DR レジスタを読み出すことで行われます。

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 (BDM=0 および RXONLY = 0)

1. SPE ビットをセットすることによって、SPI を有効にします。
2. 送信する最初のデータを 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、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り込みサブルーチンを使用しても実行できます。

図 75. 全二重モードでの TXE/RXNE/BSY 動作 (RXONLY = 0)  
連続転送の場合

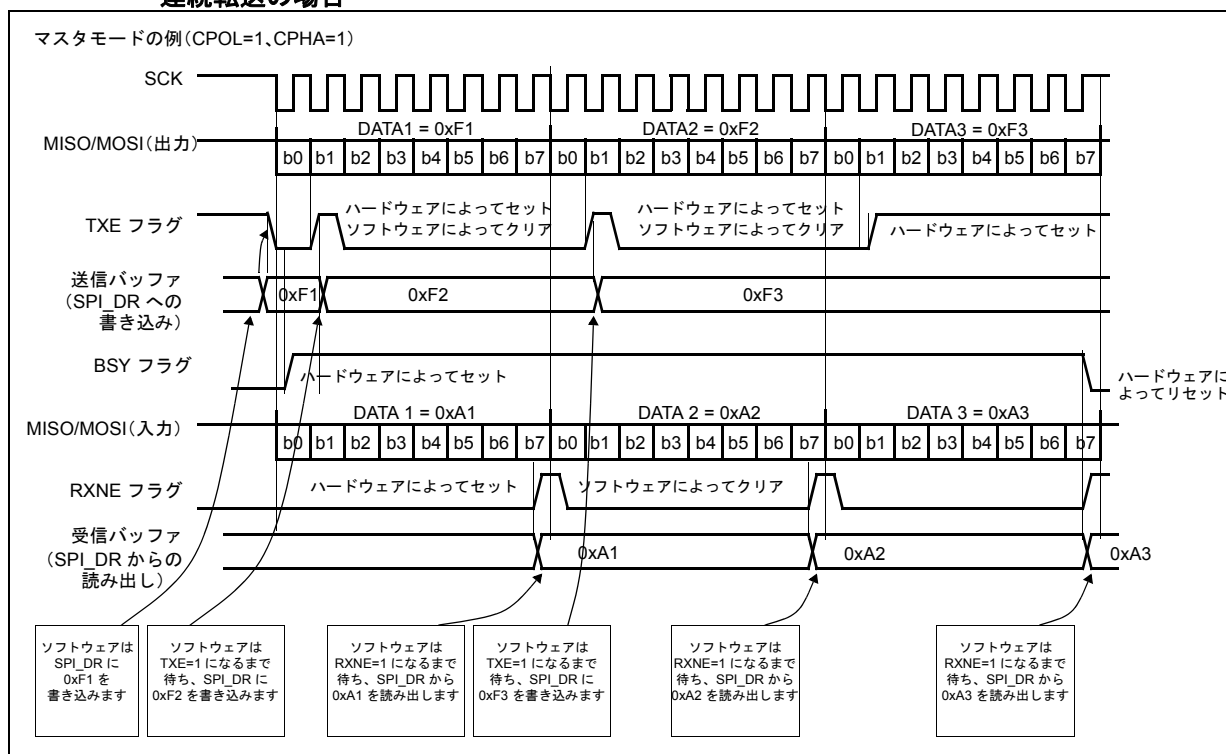
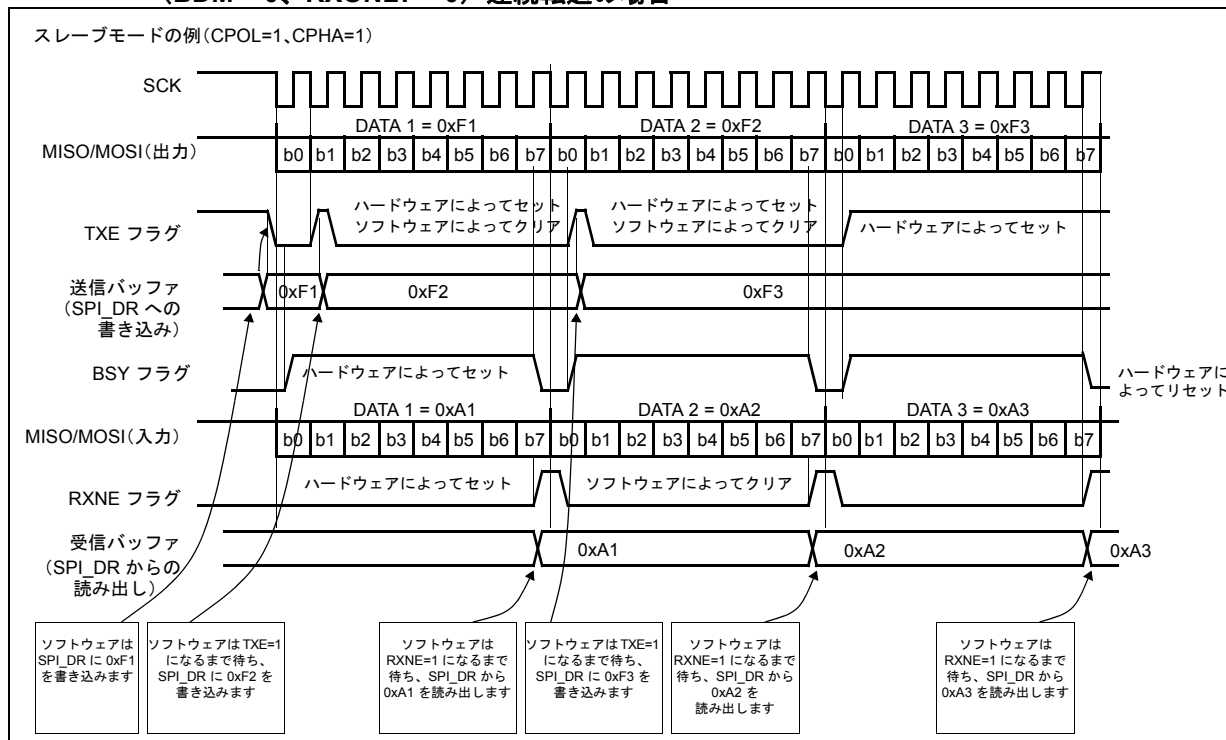


図 76. スレーブ/全二重モードでの TXE/RXNE/BSY 動作  
(BDM = 0, RXONLY = 0) 連続転送の場合





## 送信専用手順 (BDM = 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 75 と図 76 を参照)。

1. SPE ビットをセットすることによって、SPI を有効にします。
2. 送信する最初のデータを 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を書き込みます。送信するデータごとに、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:
1. マスタモードでの不連続通信時には、SPI\_DR への書き込み動作と BSY ビットの設定の間に、CPU 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 がリセットされるまで待つ必要があります。
  2. 2 つのデータ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 77. マスタ送信専用モードでの TXE/BSY 動作  
(BDM = 0 および RXONLY = 0) 連続転送の場合

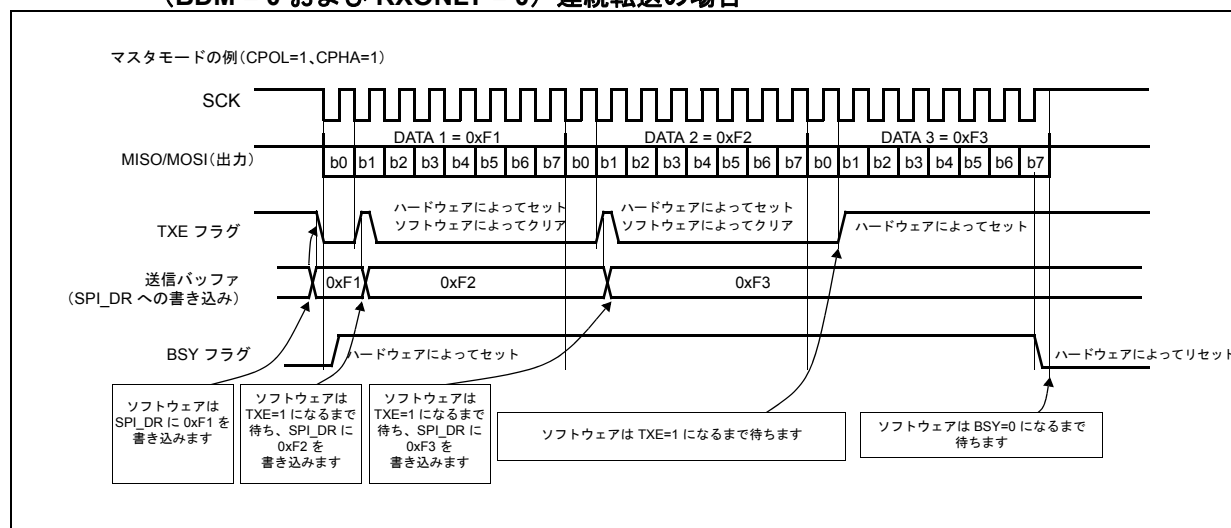
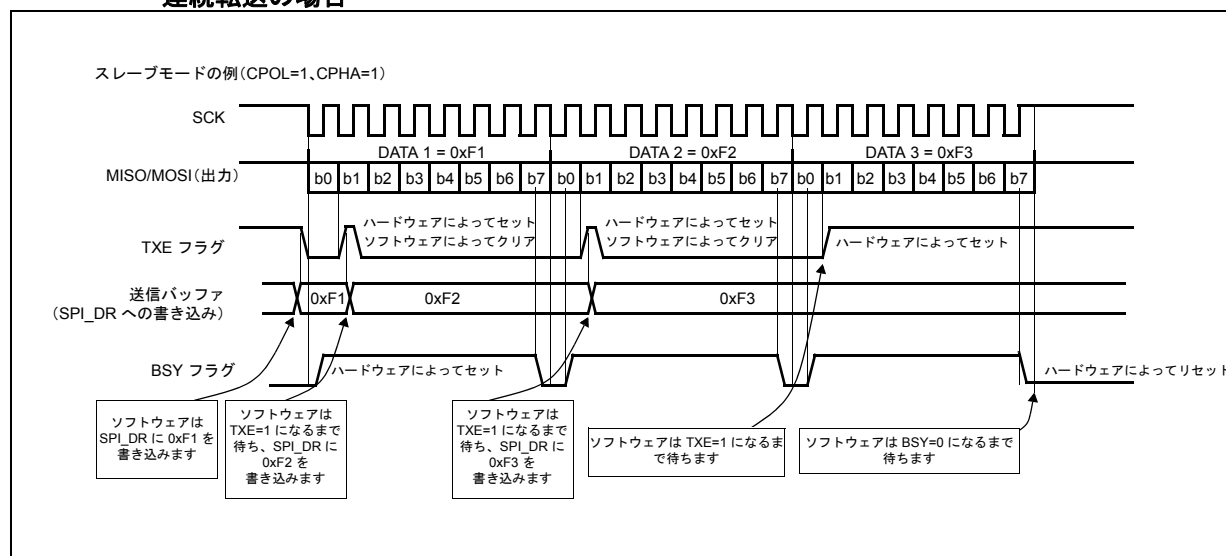




図 78. スレーブ送信専用モードでの TXE/BSY 動作 (BDM = 0 および RXONLY = 0)  
連続転送の場合



## 双方向送信手順 (BDM = 1 および BDOE = 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 BDM ビットと BDOE 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 (BDM = 0 および RXONLY = 1)

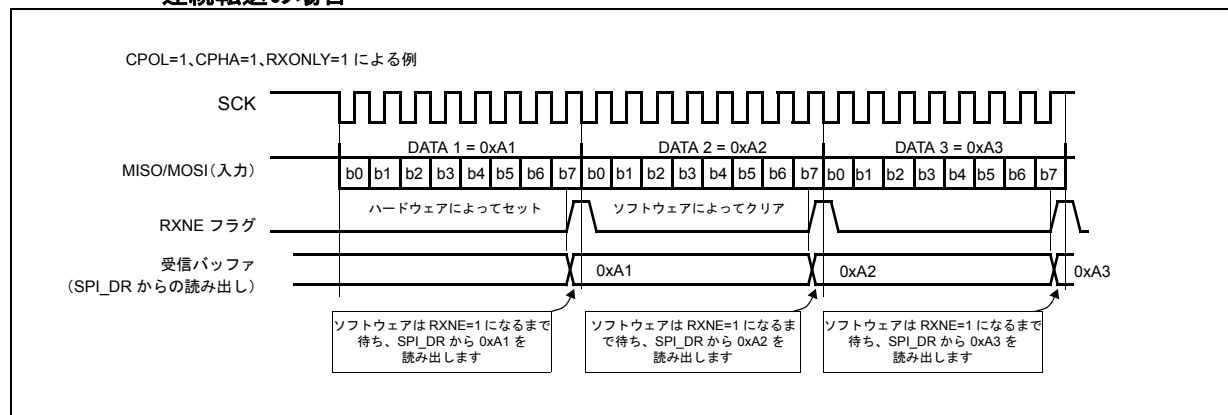
このモードでは、手順は下記のように短縮できます (図 79 を参照) :

- 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 19.3.7 : SPI の無効化 \(205 ページ\)](#) に述べる推奨事項に従ってください。

図 79. 受信専用モードでの RXNE 動作 (BDM = 0 および RXONLY = 1)  
連続転送の場合



## 双方向受信手順 (BDM = 1 および BDOE = 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 BDM ビットをセットし、BDOE ビットをリセットする必要があります。

## 連続転送と不連続転送

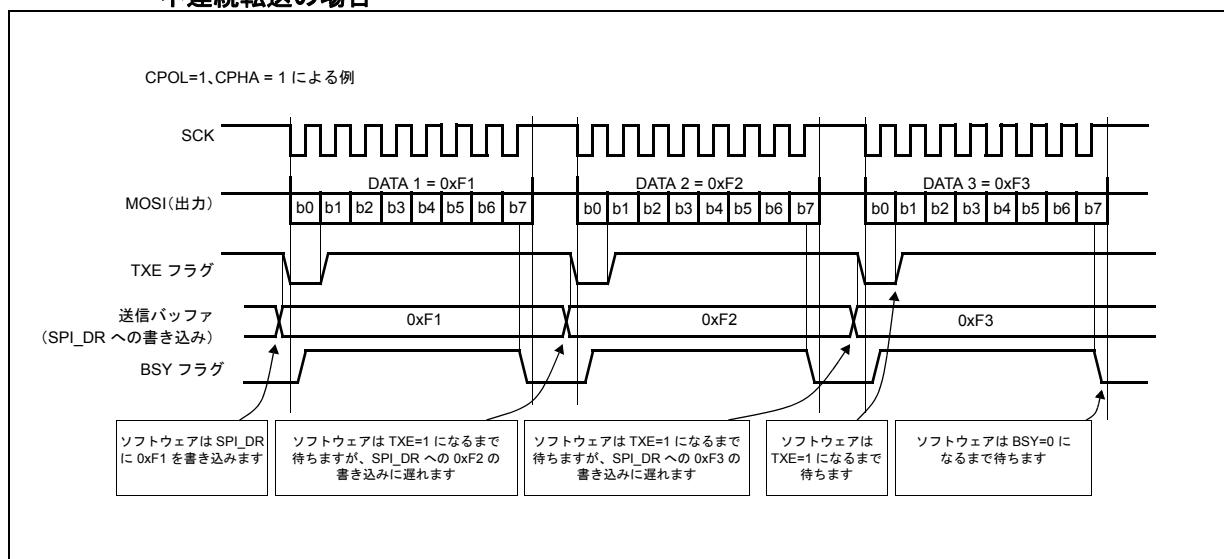
マスタモードでデータを送信するとき、ソフトウェアが十分に高速であり、各 TXE 立ち上がりエッジ (または TXE 割り込み) を検出し、現在のデータ転送が完了する前に SPI\_DR レジスタにすぐに書き込みできる場合、通信は連続的であると言われます。この場合、各データ間で SPI クロックを生成する際に不連続性はなく、各データ転送間で BSY ビットがリセットされることはありません。

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 ビットがリセットされます (図 80 を参照)。

マスタ受信専用モード (BDM = 0 および RXONLY = 1) または双方向受信モード (BDM = 1 および BDOE = 0) 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。しかし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 76 を参照)。

図 80. 送信時の TXE/BSY 動作 (BDM = 0 および RXLONRY = 0)  
不連続転送の場合



## 19.3.6 ステータスフラグ

アプリケーションが SPI バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

### 送信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、送信バッファが空であり、次に送信するデータをバッファにロードできることを示します。TXE フラグは、SPI\_DR レジスタへの書き込み時にリセットされます。

### 受信バッファノットエンプティ (RXNE)

このフラグがセットされると、受信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 の読み出し時にリセットされます。

### 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/リセットされます (このフラグへの書き込みは無効です)。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 HALT 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、下記の手順を厳しく順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き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リセットされます。

- 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。

通信が連続的なとき、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。

通信が連続的なとき、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戻ります。

**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。

## 19.3.7 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。これは SPE ビットをリセットすることによって行われます。

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 HALT 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 (BDM = 0、RXONLY = 0) の場合：**

1. RXNE=1 になるまで待つから、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 HALT 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 (BDM=0、RXONLY=0) または双方向送信モード (BDM=1、BDOE=1) の場合：**

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)、必要なら、HALT 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、BDM=0、RXONLY=1) または双方向受信モード (MSTR=1、BDM=1、BDOE=0) の場合：**

このケースは、SPI が新しい転送を開始しないように、特別な方法で管理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用して SPI の 1 クロックサイクルの間待つから、SPI を無効にします (SPE=0)。
3. 次に、最後の RXNE=1 になるまで待つから、HALT 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）。

**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、BDM=0、RXONLY=1) または双方向受信モード (MSTR=0、BDM=1、BDOE=0) の場合 :**

1. SPI はいつでも無効にできます (SPE=1 の書き込み)。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、HALT 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、HALT 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)。

## 19.3.8 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 割り込みが生成されます。
- SPE ビットがクリアされます。これによって、デバイスからのすべての出力がブロックされ、SPI インタフェースが無効になります。
- MSTR ビットがリセットされ、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。割り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り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 レジスタを読み出すこと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。

OVR 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。

## 19.3.9 SPI 低電力モード

表 41. 低電力モードでの SPI 動作

モード	説明
WAIT	SPI には影響ありません。 SPI 割り込みイベントによって、デバイスは WAIT モードから抜けます。
HALT	SPI レジスタは凍結されます。 HALT モードでは、SPI は無効です。SPI がマスタモードにある場合、「HALT モードからのウェイクアップ」機能を備えた割り込みによってデバイスがウェイクアップすると、通信は再開します。 SPI がスレーブモードにある場合、データの最初のサンプリングエッジを検出した後、MCU を HALT モードからウェイクアップできます。

### SPI を使用してデバイスを HALT モードからウェイクアップする方法

マイクロコントローラが HALT モードにあるときでも、SPI はスレーブとして応答することができます。ただし、NSS ピンがローに接続されているか、HALT モードに入る前に SSI ビットがリセットされている必要があります。

データの最初のサンプリングエッジ (CPHA ビットによって定義) が検出されると、

- SPI\_SR レジスタの WKUP ビットがセットされます。
- SPI\_ICR レジスタの WKIE 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。
- この割り込みは、デバイスを HALT モードからウェイクアップします。
- システムクロックの復元には時間が必要なため、SPI スレーブは、正しく通信できるようになる前に、いくつかのデータを送受信します。したがって、次のプロトコルを使用することが不可欠です。
  - － HALT モードに入る前に、SPI\_DR に特定の値が書き込まれます。この値は、SPI が HALT モードにあることを外部マスタに知らせます。
  - － 外部マスタは、SPI が HALT モードにあることを示すユニークな値以外の新しい値を SPI スレーブデバイスから受信するまで、同じバイトを連続的に送信します。この新しい値は、SPI スレーブがウェイクアップし、正しく通信できることを示します。

### 受信専用モードでの制約

受信専用モード (BDM = 0 と RXONLY = 1 または BDM = 1 と BDOE = 0) では、システムクロックの復元に必要な時間がデータ受信時間を上回ることがあるため、ウェイクアップ機能は保証されません。これによって受信時にデータの消失が発生するため、スレーブデバイスは、どのデータが正しく受信されたかをマスタに知らせることができません。

## 19.3.10 SPI 割り込み

表 42. SPI 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット	WAIT 終了	HALT 終了
送信バッファエンプティフラグ	TXE	TXIE	あり	なし
受信バッファノットエンプティフラグ	RXNE	RXIE	あり	なし
ウェイクアップイベントフラグ	WKUP	WKIE	あり	あり
マスタモードフォールトイベント	MODF	ERRIE	あり	なし
オーバーランエラー	OVR		あり	なし



## 19.4 SPI レジスタ

### 19.4.1 SPI 制御レジスタ 1 (SPI\_CR1)

アドレスオフセット : 0x00

リセット値 : 0x00

7	6	5	4	3	2	1	0
LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
rw	rw	rw			rw	rw	rw

ビット 7 **LSBFIRST** : フレームフォーマット <sup>(1)</sup>

0 : MSB が最初に送信されます

1 : LSB が最初に送信されます

ビット 6 **SPE** : SPI イネーブル <sup>(2)</sup>

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

ビット 5:3 **BR[2:0]** : ボーレート制御

000 :  $f_{\text{MASTER}}/2$

100 :  $f_{\text{MASTER}}/32$

001 :  $f_{\text{MASTER}}/4$

101 :  $f_{\text{MASTER}}/64$

010 :  $f_{\text{MASTER}}/8$

110 :  $f_{\text{MASTER}}/128$

011 :  $f_{\text{MASTER}}/16$

111 :  $f_{\text{MASTER}}/256$

**注 :** これらのビットは、通信中には変更しないでください。

ビット 2 **MSTR** : マスタ選択 <sup>(1)</sup>

0 : スレーブ設定

1 : マスタ設定

ビット 1 **CPOL** : クロック極性 <sup>(1)</sup>

0 : アイドル時に 0 に SCK

1 : アイドル時に 1 に SCK

ビット 0 **CPHA** : クロック位相 <sup>(1)</sup>

0 : 最初のクロック遷移が最初のデータキャプチャエッジです。

1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

- このビットは、通信中には変更しないでください。
- SPI を無効にするときには、[セクション 19.3.7 : SPI の無効化 \(205 ページ\)](#) で説明されている手順に従ってください。



## 19.4.2 SPI 制御レジスタ 2 (SPI\_CR2)

アドレスオフセット : 0x01  
リセット値 : 0x00

7	6	5	4	3	2	1	0
BDM	BDOE	予約済み			RXOnly	SSM	SSI
rw	rw				rw	rw	rw

ビット 7 **BDM** : 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  
1 : 1 線双方向データモードを選択

ビット 6 **BDOE** : 双方向モードの入力／出力イネーブル  
このビットは、BDM に 1 がセットされた場合の双方向モードでの転送方向を選択します。  
0 : 入力有効 (受信専用モード)  
1 : 出力有効 (送信専用モード)  
マスタモードでは MOSI ピンが使用され、スレーブモードでは MISO ピンが使用されます。

ビット 5:3 予約済み

ビット 2 **RXONLY** : 受信専用  
0 : 全二重 (送受信)  
1 : 出力無効 (受信専用モード)  
2 線単方向モードでの転送方向は、このビットと BDM 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。つまり、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。

ビット 1 **SSM** : ソフトウェアスレーブ管理  
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。  
SSM ビットがセットされているとき、NSS ピンの入力値は SSI ビットからの値によって置き換えられます。

ビット 0 **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 NSS ピンに設定され、NSS ピンの I/O 値は無視されます。  
0 : スレーブモード  
1 : マスタモード

19.4.3 SPI 割り込み制御レジスタ (SPI\_ICR)

アドレスオフセット : 0x02  
リセット値 : 0x00

7	6	5	4	3	2	1	0
TXIE	RXIE	ERRIE	WKIE	予約済み			
rw	rw	rw	rw				

- ビット 7 **TXIE** : 送信バッファエンプティ割り込みイネーブル
- 0 : TXE 割り込みはマスクされます。
  - 1 : TXE 割り込みはマスクされません。TXE フラグがセットされたとき、割り込みリクエストの生成に使用されます。
- ビット 6 **RXIE** : 受信バッファノットエンプティ割り込みイネーブル
- 0 : RXNE 割り込みはマスクされます。
  - 1 : RXNE 割り込みはマスクされません。RXNE フラグがセットされたとき、割り込みリクエストの生成に使用されます。
- ビット 5 **ERRIE** : エラー割り込みイネーブル
- 0 : エラー割り込みはマスクされます。
  - 1 : エラー割り込みは有効です。エラー条件が発生したとき (OVR、MODF)、割り込みリクエストの生成に使用されます。
- ビット 4 **WKIE** : ウェイクアップ割り込みイネーブル
- 0 : ウェイクアップ割り込みはマスクされます。
  - 1 : ウェイクアップ割り込みは有効です。WKUP フラグがセットされたとき、割り込みリクエストの生成に使用されます。
- ビット 3:0 予約済み



## 19.4.4 SPI ステータスレジスタ (SPI\_SR)

アドレスオフセット : 0x03  
リセット値 : 0x02

7	6	5	4	3	2	1	0
BSY	OVR	MODF	予約済み	WKUP	予約済み	TXE	RXNE
r	rc_w0	rc_w0		rc_w0		r	r

- ビット 7 **BSY** : ビジーフラグ
- 0 : SPI はビジー状態ではありません。
  - 1 : SPI は通信ビジー状態です。
- このフラグは、ハードウェアによってセット／リセットされます。
- 注 :** **BSY フラグの使用には注意が必要です (セクション 19.3.6 : ステータスフラグ (204 ページ) およびセクション 19.3.7 : SPI の無効化 (205 ページ) を参照)。**
- 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。
- 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。
- ビット 4 予約済み
- ビット 3 **WKUP** : ウェイクアップフラグ
- 0 : ウェイクアップイベントは発生していません。
  - 1 : ウェイクアップイベントが発生しました。
- このフラグは、STM8 が HALT モードであり、SPI がスレーブとして設定されているとき、SCK の最初のサンプリングエッジでセットされます。
- 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リセットされます。
- ビット 2 予約済み
- ビット 1 **TXE** : 送信バッファエンプティ
- 0 : 送信バッファは空ではありません。
  - 1 : 送信バッファは空です。
- ビット 0 **RXNE** : 受信バッファノットエンプティ
- 0 : 受信バッファは空です。
  - 1 : 受信バッファは空ではありません。

19.4.5 SPI データレジスタ (SPI\_DR)

アドレスオフセット : 0x04  
リセット値 : 0x00

7	6	5	4	3	2	1	0
DR[7:0]							
rw							

ビット 7:0 **DR[7:0]** : データレジスタ  
受信されたバイト、または送信されるバイト。  
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、もう一方は読み出し用 (受信バッファ) です。データレジスタへの書き込みは、送信バッファへの書き込みになります。データレジスタからの読み出しでは、受信バッファに保持されている値が返されます。

19.5 SPI レジスタマップとリセット値

表 43. SPI レジスタマップとリセット値

アドレス オフ セット	レジスタ名	7	6	5	4	3	2	1	0
0x00	SPI_CR1 リセット値	LSB FIRST 0	SPE 0	BR2 0	BR1 0	BR1 0	MSTR 0	CPOL 0	CPHA 0
0x01	SPI_CR2 リセット値	BDM 0	BDOE 0	- 0	- 0	- 0	RXONLY 0	SSM 0	SSI 0
0x02	SPI_ICR リセット値	TXIE 0	RXIE 0	ERRIE 0	WKIE 0	- 0	- 0	- 0	- 0
0x03	SPI_SR リセット値	BSY 0	OVR 0	MODF 0	- 0	WKUP 0	- 0	TXE 1	RXNE 0
0x04	SPI_DR リセット値	MSB 0	- 0	- 0	- 0	- 0	- 0	- 0	LSB 0



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

## 20 Universal synchronous/asynchronous receiver transmitter (USART)

### 20.1 USART の概要

USART (Universal Synchronous Asynchronous Receiver Transmitter) を使用すれば、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟に全二重データ交換を行うことができます。USART は広範囲なボーレートを提供します。

USART は同期式単方向通信をサポートします。USART は、また、マルチプロセッサ通信にも使用することができます。

### 20.2 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）
- 高精度ボーレートジェネレータシステム
  - － プログラミング可能な送受信共通のボーレート：最大  $f_{MASTER}/16$
- プログラミング可能なデータワード長：8 または 9 ビット
- 設定可能な STOP ビット：1 個または 2 個の STOP ビットをサポート
- 同期通信用のトランスミッタクロック出力
- トランスミッタとレシーバ用に別個のイネーブルビット
- 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
  - － ノイズエラー
  - － フレームエラー
  - － パリティエラー
- 5 つの割り込みソースとフラグ：
  - － 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
  - － パリティエラー
- 2 つの割り込みベクタ：
  - － トランスミッタ割り込み
  - － レシーバ割り込み

- 低電力モード
- 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つのレシーバウェイクアップモード：
  - － アドレスビット（MSB）
  - － アイドルライン

## 20.3 USART 機能の説明

インタフェースは、3本のピンによって他のデバイスに外部接続されます（[図 81](#) を参照）。USART 双方向通信では、少なくとも USART 受信データ入力（USART\_RX）と送信データ出力（USART\_TX）の2本のピンが必要です。

**USART\_RX** 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ます。

**USART\_TX** はシリアルデータ出力です。トランスミッタが無効なときには、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あり、送信すべきデータがないときには、ピンはハイレベルになります。

通常の USART モードでは、これらのピンを通じてシリアルデータが以下を含むフレームとして送受信されます。

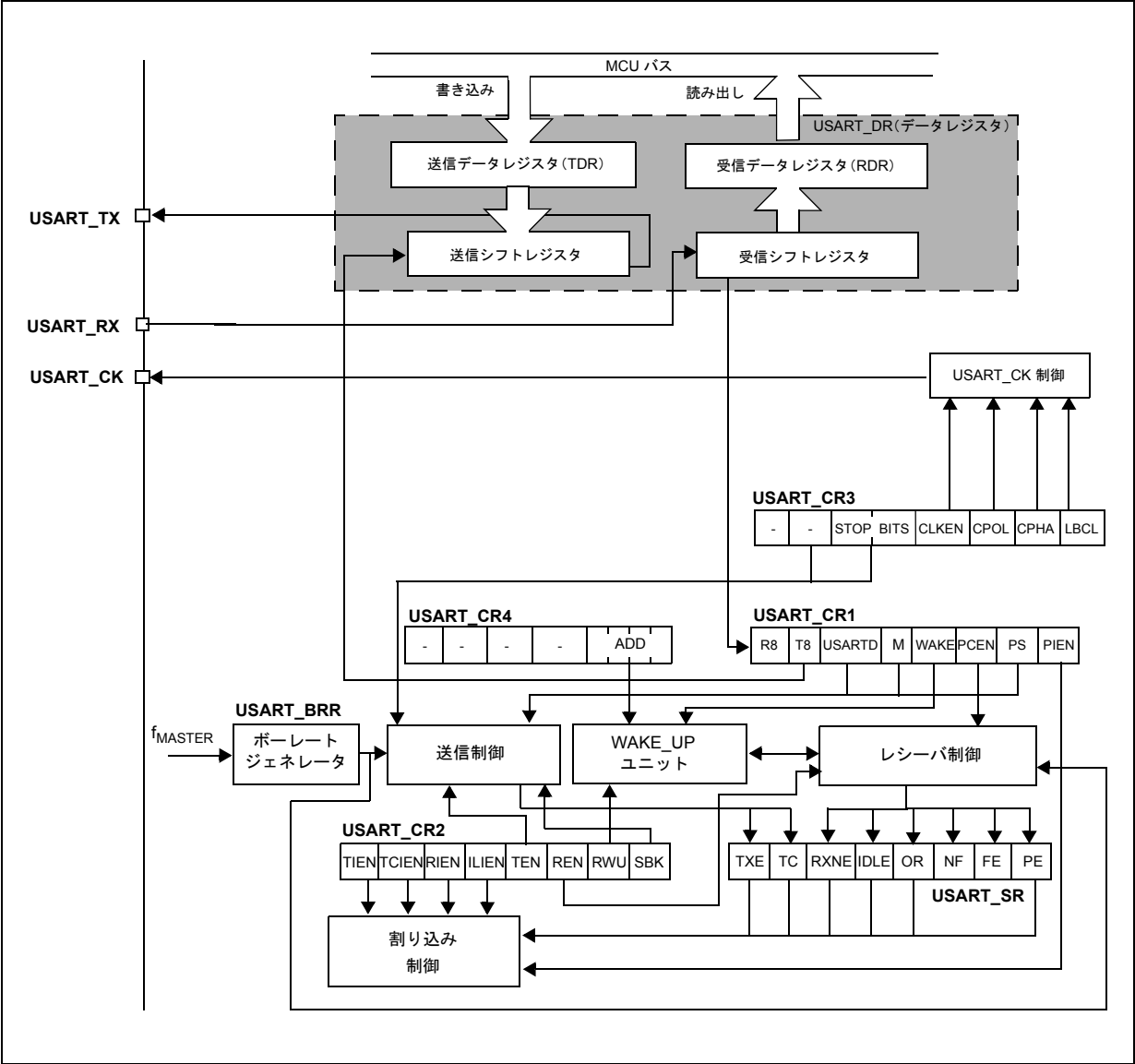
- 送受信前のアイドルライン
- START ビット
- データワード（8 または 9 ビット）、LSB ファースト
- フレームの完了を示す 1 および 2 個の STOP ビット
- ステータスレジスタ（USART\_SR）
- データレジスタ（USART\_DR）
- 16 ビットのボーレートプリスケアラ（USART\_BRR）

各ビットの定義については、レジスタの説明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、以下のピンが必要です。

**USART\_CK**：トランスミッタクロック出力。このピンは、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（START ビットと STOP 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ミングできます。

図 81. USART のブロック図





## 20.3.1 USART キャラクタの説明

ワード長は、USART\_CR1 レジスタの M ビットをプログラミングすることによって、8 ビットまたは 9 ビットとして選択できます（図 82 を参照）。

USART\_TX ピンは、START ビットの処理中にはロー状態です。また、STOP ビットの処理中にはハイ状態です。

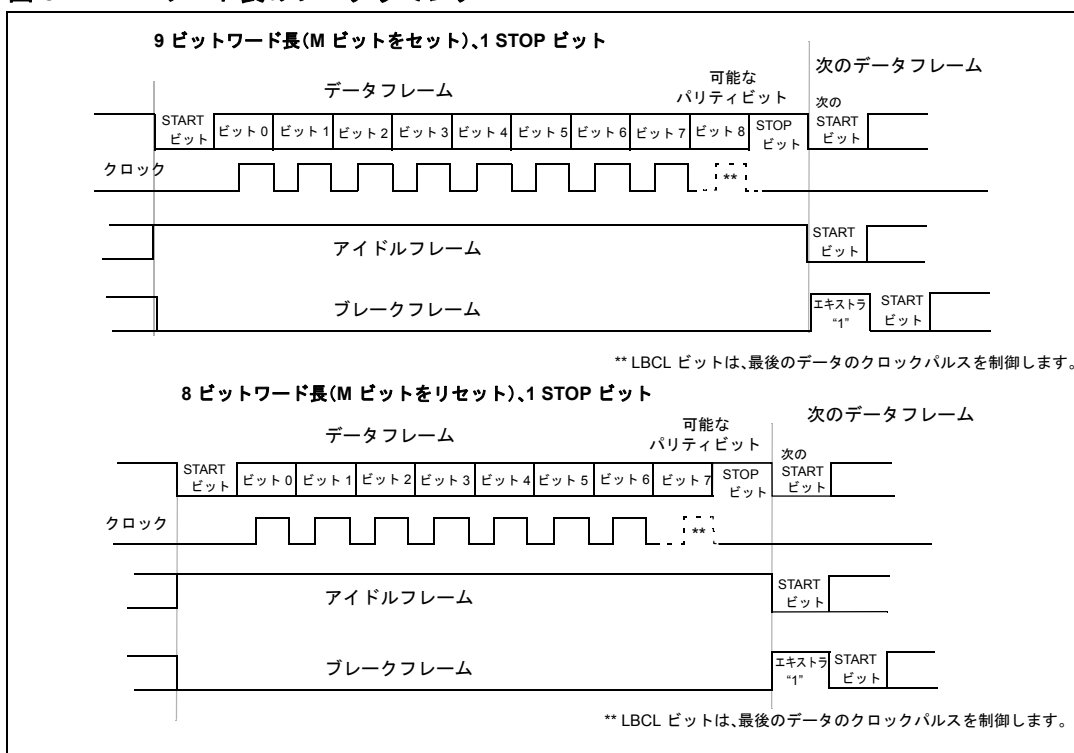
**アイドルキャラクタ**は、1 のフレーム全体として解釈されます（1 の数には、START ビット、データビットの数、および STOP ビットの数が含まれます）。

**ブレイクキャラクタ**は、フレーム周期中に“0”を受信することであると解釈されます。ブレイクフレームの終わりに、トランスミッタは、START ビットに確認応答するために、1 個または 2 個の STOP ビット（論理“1”ビット）を挿入します。

送信と受信は、共通のボーレートジェネレータによって駆動され、それぞれのクロックは、トランスミッタとレシーバのイネーブル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 82. ワード長のプログラミング



## 20.3.2 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。M ビットがセットされると、ワード長は 9 ビットとなり、9 番目のビット (MSB) が USART\_CR1 レジスタの T8 ビットに格納されなければなりません。

送信イネーブルビット (TEN) がセットされると、送信シフトレジスタ内のデータが USART\_TX ピンに出力され、対応するクロックパルスが USART\_CK ピンに出力されます。

### キャラクタの送信

USART 送信時、データは LSB ファーストで USART\_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 81 を参照)。

各キャラクタの前に START ビット (1 ビット周期の間、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 STOP ビットによって終端されます。

以下の STOP ビットが USART によってサポートされます。

- 注:
- 1 **TEN ビットは、データ送信中はリセットしないでください。送信中に TEN ビットをリセットすると、ポーレートカウンタが凍結されるので、USART\_TX ピンのデータが破壊されます。現在送信中のデータは失われます。**
  - 2 **TEN ビットが有効にされた後、アイドルフレームが送信されます。**

### 設定可能な送信時の STOP ビット

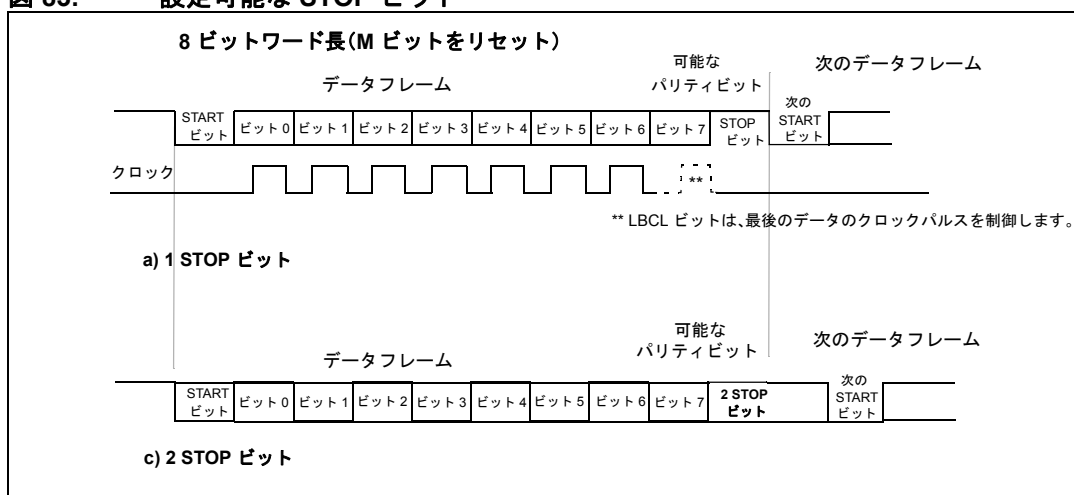
キャラクタごとに送信される STOP ビットの数、制御レジスタ 3 のビット 5、4 でプログラミングできます。

- **1 STOP ビット**: これは STOP ビット数のデフォルト値です。
- **2 STOP ビット**: 通常モードの USART でサポートされます。

アイドルフレームの送信には STOP ビットが含まれます。

ブレーク送信では、10 個 (m = 0 の場合) または 11 個 (m = 1 の場合) のロービットの後に、設定された数の STOP ビットが続きます。ローレベルビットの数が 10/11 個を超える長いブレークを送信することはできません。

図 83. 設定可能な STOP ビット



## 手順

1. 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。
2. USART\_CR3 で、STOP ビットの数プログラミングします。
3. 以下の順序でボーレートレジスタをプログラミングすることによって、望ましいボーレートを選択します。
  - a) USART\_BRR2
  - b) USART\_BRR1
4. USART\_CR2 の TEN ビットをセットして、トランスミッタモードを有効にします。
5. 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、送信されるデータごとにこれを繰り返します。
6. USART\_DR レジスタに最後のデータが書き込まれたら、TC が 1 にセットされるまで、すなわち最後のデータ送信が完了するまで待ちます。この最後の手順は、たとえば、USART を無効にするときや HALT モードに入るときに、最後のデータ送信の破損を避けるために必要です。

## 単一バイト通信

TXE ビットのクリアは、常にデータレジスタへの書き込みによって行われます。

TXE ビットは、ハードウェアによってセットされ、次のことを示します。

- データが TDR からシフトレジスタに移動され、データ送信が開始されました。
- TDR レジスタは空です。
- 次のデータを、前のデータを上書きすることなく、USART\_DR レジスタに書き込みます。

TIEEN ビットがセットされている場合、このフラグは割り込みを生成します。

送信が行われているときは、データは USART\_DR レジスタへの書き込み命令によって TDR レジスタに格納されます。データは現在の送信が終了した時点でシフトレジスタにコピーされます。

送信が行われていないときには、データは、USART\_DR レジスタへの書き込み命令によってシフトレジスタに直接格納され、データ送信が開始され、TXE ビットがただちにセットされます。

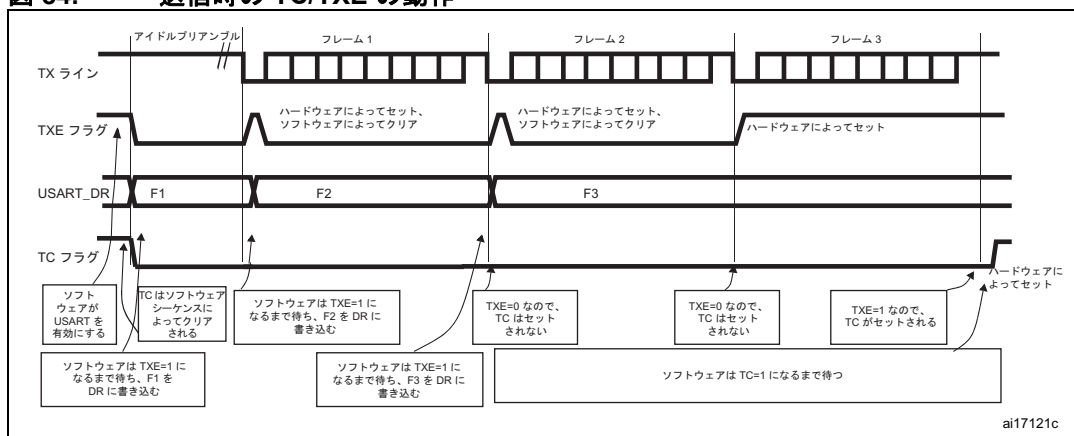
フレームの送信が完了し（STOP ビット後）、TXE ビットがセットされると、TC ビットがセットされます。USART\_CR2 レジスタの TCIEEN ビットがセットされている場合、割り込みが生成されます。USART\_DR レジスタにデータを書き込んだ後、HALT モードに入るか USART を無効にする前に、TC が“1”にセットされるまで待つ必要があります（図 84：送信時の TC/TXE の動作を参照）。

TC ビットのクリアは、以下のソフトウェアシーケンスによって行われます。

1. USART\_SR レジスタの読み出し
2. USART\_DR レジスタへの書き込み

**注：** TC ビットは、“0”を書き込むことによってもクリアされます。このクリアシーケンスは、マルチバッファ通信の場合にのみ推奨されます。

図 84. 送信時の TC/TXE の動作



1. この例では、TE がセットされた後、複数の別の送信が発生したと仮定しています。そうでない場合は、USART\_DR に初めて書き込みを行ったときに IDLE プリアンブルが最初に送信されます。

## ブレイクキャラクタ

SBK ビットをセットすると、ブレイクキャラクタが送信されます。ブレイクフレーム長は、M ビットに依存します (図 82 を参照)。

SBK ビットが“1”にセットされると、現在のキャラクタ送信が完了してから、USART\_TX ラインにブレイクキャラクタが送信されます。このビットは、ブレイクキャラクタが完了したとき (ブレイクキャラクタの STOP ビット)、ハードウェアによってリセットされます。USART は、次のフレームの START ビットが正しく認識されるように、最後のブレイクフレームの終わりに論理 1 ビットを挿入します。

**注：** ブレイクキャラクタは、STOP ビットの数に関係なく送信されます。USART が 2 つの STOP ビットでプログラムされた場合、TX ラインは、最初の STOP ビットの終了までローに引き下げられます。次に、2 つの論理 1 ビットが次のキャラクタの前に挿入されます。

**注：** ソフトウェアがブレイク送信の開始前に SBK ビットをリセットした場合、ブレイクキャラクタは送信されません。2 つの連続したブレイクの場合、前のブレイクの STOP ビットの後で SBK ビットをセットしてください。

## アイドルキャラクタ

TEN ビットをセットすると、USART は、最初のデータフレームの前にアイドルフレームを送信します。

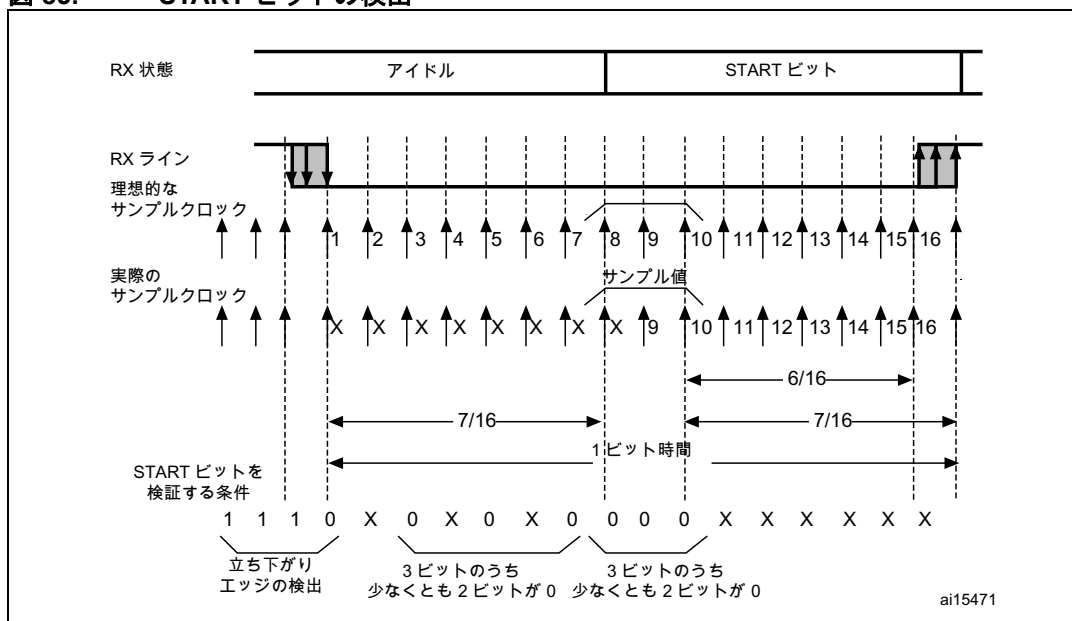
## 20.3.3 レシーバ

USART は、8 ビットまたは 9 ビットのデータワードを受け取ることができます。M ビットがセットされると、ワード長は 9 ビットとなり、MSB が USART\_CR1 レジスタの R8 ビットに格納されます。

## START ビットの検出

USART では、特定のサンプルシーケンスが認識されると、START ビットが検出されます。このシーケンスは、1 1 1 0 X 0 X 0 X 0 X 0 X 0 X 0 です。START ビットの検出シーケンスを図 85 に示します。

図 85. START ビットの検出



**注：**シーケンスが完了していない場合、START 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。

3 ビットのうち 2 ビットのみが 0 の場合（3 番目、5 番目、および 7 番目のビットでのサンプリング、または 8 番目、9 番目、および 10 番目のビットでのサンプリング）、START ビットは確認されますが、NF ノイズフラグビットがセットされます。

START ビットは、最後の 3 つのサンプルが 0 である場合に確認されます（8 番目、9 番目、および 10 番目のビットでのサンプリング）。

## キャラクタの受信

USART の受信時には、データは USART\_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（図 2 を参照）。

手順：

1. 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。
2. USART\_CR3 で、STOP ビットの数プログラミングします。
3. 以下の順序でボーレートレジスタをプログラミングすることによって、望ましいボーレートを選択します。
  - a) USART\_BRR2
  - b) USART\_BRR1
4. USART\_CR2 の REN 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、START 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。
- RIEN 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。

- 受信時にフレームエラー、ノイズ、またはオーバーランエラーが検出された場合、エラーフラグをセットできます。
- RXNE ビットのクリアは、USART\_DR レジスタへのソフトウェア読み出しによって行われます。RXNE フラグは、0 を書き込むことによってもクリアできます。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、REN ビットをリセットしないでください。受信中に REN ビットが無効にされた場合、現在のバイトの受信は中止されます。

## ブレークキャラクタ

ブレークキャラクタが受信されると、USART は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ときには、データキャラクタの受信と同じ手順があり、ILIEEN ビットがセットされている場合は割り込みが生成されます。

## オーバーランエラー

RXNE ビットがリセットされていないときにキャラクタが受信されると、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。

オーバーランエラーが発生すると、

- OR 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。オーバーラン中に受信された 2 番目のデータは失われます。
- RIEN ビットがセットされていた場合、割り込みが生成されます。
- OR 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。

## ノイズエラー

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ます。

**図 86. ノイズ検出のためのデータサンプリング**



**注：** サンプルクロック周波数はボーレートの 16 倍です。



表 44. 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	NF の状態	受信ビットの値	データの有効性
000	0	0	有効
001	1	0	無効
010	1	0	無効
011	1	1	無効
100	1	0	無効
101	1	1	無効
110	1	1	無効
111	0	1	有効

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。

ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。NF ビットは、USART\_SR レジスタの読み出し動作と、それに続く USART\_DR レジスタの読み出し動作によってリセットされます。

## 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 STOP 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：

- ハードウェアによって FE 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。
- 1 バイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。

FE ビットは、USART\_SR レジスタの読み出し動作と、それに続く USART\_DR レジスタの読み出し動作によってリセットされます。

## 設定可能な受信時の STOP ビット

受信する STOP ビット数は、制御レジスタ 3 の制御ビットを通じて設定でき、1 または 2 とすることができます。

1. 1 STOP ビット：1 STOP ビットのサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
2. 2 STOP ビット：2 STOP ビットのサンプリングは、最初の STOP 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 STOP 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 STOP 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 STOP ビットの最後でセットされます。



## 20.3.4 高精度ボーレートジェネレータ

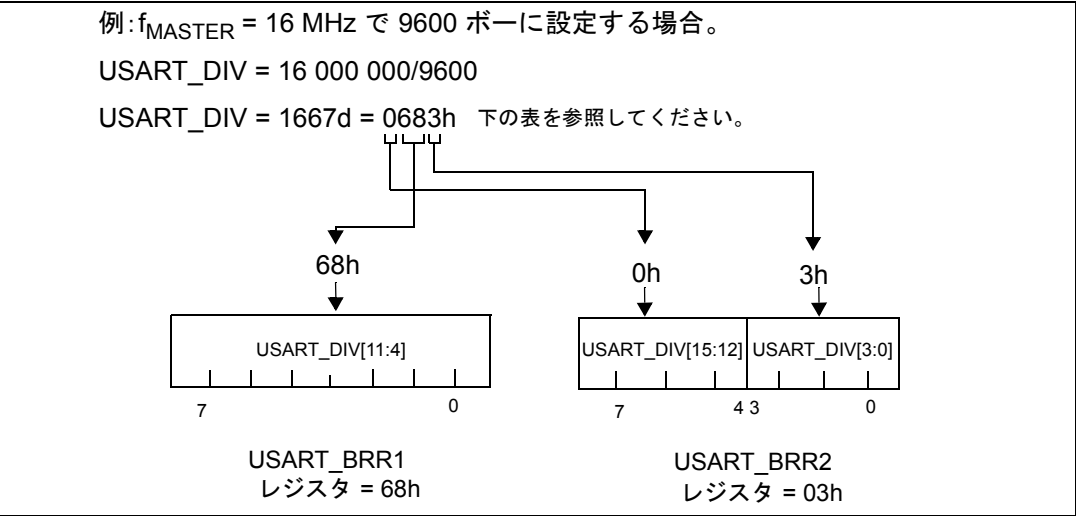
レシーバとトランスミッタ (Rx と Tx) は、次の式に従って、16 ビットのディバイダ USART\_DIV によってプログラミングされた同じボーレートにセットされます。

$$\text{Tx/Rx ボーレート} = \frac{f_{\text{MASTER}}}{\text{USART\_DIV}}$$

USART\_DIV ボーレートディバイダは、符号なし整数であり、図 87 に示されているように、BRR1 および BRR2 レジスタでコード化されます。

典型的なボーレートのプログラム例を表 45 に示します。

図 87. BRR レジスタで USART\_DIV をコード化する方法



注: BRR1 に書き込みが行われた後、ボーレートカウンタはボーレートレジスタの新しい値で更新されます。したがって、トランザクション中はボーレートレジスタの値を変更しないでください。BRR2 は BRR1 の前にプログラムされなければなりません。

注: USART\_DIV は 16d 以上でなければなりません。

表 45. ボーレートのプログラミングと誤差計算

ボーレート	$f_{\text{MASTER}} = 16 \text{ MHz}$				
単位: bps	実際	誤差 (%) <sup>(1)</sup>	USART_DIV	BRR1	BRR2
2.4	2.4	0.0	2710h	71h	20h
9.6	9.6	0.0	09C4h	9Ch	04h
19.2	19.2	0.0	04E2	4Eh	02h
57.6	57.554	-0.08	01A1h	1Ah	01h
115.2	115.385	0.16	00D0h	0Dh	00h
230.4	230.769	0.16	0068h	06h	08h
460.8	461.538	0.16	0034h	03h	04h
921.6	923.077	0.16	001Ah	01h	0Ah

1. 誤差 (%) = (計算値 - 期待値) ボーレート / ボーレート期待値

注:  $f_{\text{MASTER}}$  周波数が低いほど、特定のボーレートに対する精度は低くなります。上のデータで到達可能なボーレートの上限を固定できます。

## 20.3.5 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 10/11 ビットのキャラクタ長
- 小数点適用のボーレートを使用するかどうか

表 46. USART レシーバの許容誤差 (USART\_DIV が 0 の場合)

M ビット	NF はエラー	NF は無視
0	3.75%	4.375%
1	3.41%	3.97%

表 47. USART レシーバの許容誤差 (USART\_DIV が 0 でない場合)

M ビット	NF はエラー	NF は無視
0	3.33%	3.88%
1	3.03%	3.53%

注 : 表 46 と表 47 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、若干異なる場合があります。

## 20.3.6 パリティ制御

パリティ制御 (送信中のパリティビット生成と受信中のパリティ検査) を有効にするには、USART\_CR1 レジスタの PCEN ビットをセットします。M ビットによって定義されるフレーム長に応じて、予想される USART フレームフォーマットを表 48 に示します。

表 48. フレームフォーマット

M ビット	PCEN ビット	USART フレーム
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

凡例 : SB : START ビット、STB : STOP ビット、PB : パリティビット

注 : アドレスマークによるウェイクアップの場合、データの最上位ビットが考慮されます (パリティビットではありません)。

**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”のビット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 ビット =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”のビット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 ビット =1)、パリティビットは 1 になります。

**送信**：USART\_CR1 の PCEN ビットがセットされていると、データレジスタに書き込まれたデータの MSB は送信されませんが、パリティビットによって変更され、偶数パリティが選択されている (PS=0) 場合には 1 を偶数個付与し、奇数パリティが選択されている場合には 1 を奇数個付与します。

**受信**：パリティ検査に失敗した場合、USART\_SR レジスタの PE フラグがセットされ、USART\_CR1 レジスタの PIEN 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。

## 20.3.7 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ことによって、マルチプロセッサ通信を行うことができます。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、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い場合が多く、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置かれます。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。
- すべての受信割り込みは禁止されます。
- USART\_CR2 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されますが、特定の条件下ではソフトウェアによる書き込みが可能です。

USART は、USART\_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードの出入りを制御できます。

- 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

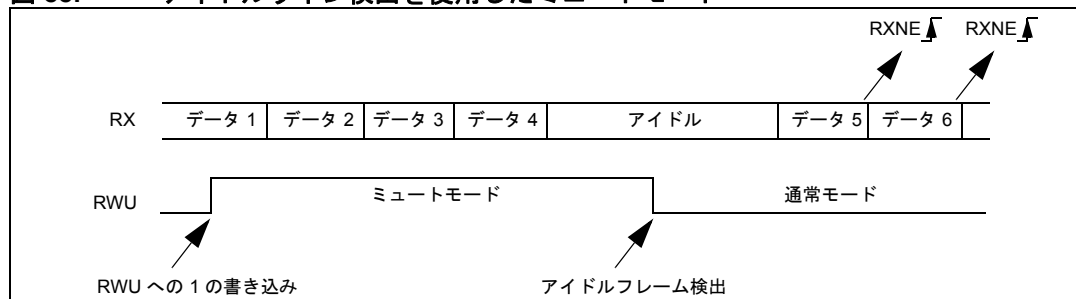
### アイドルライン検出 (WAKE=0)

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、アイドルフレームの検出によって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 88](#) に示します。

図 88. 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 ビットに置かれます。レシーバは、この 4 ビットワードを、USART\_CR4 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。

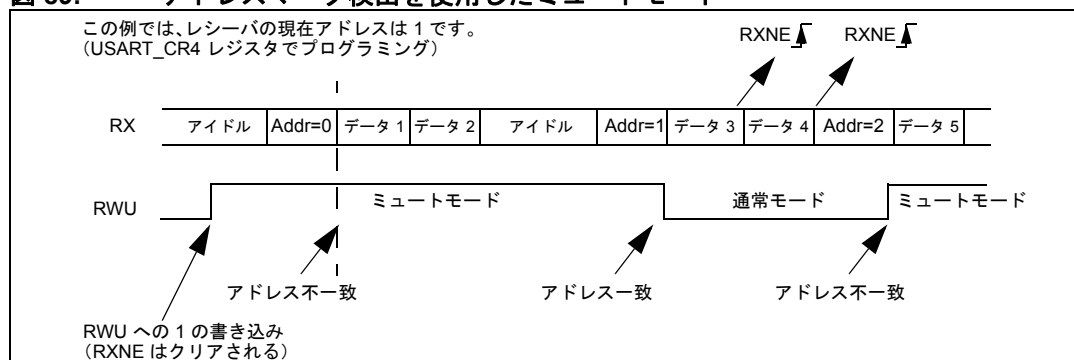
プログラミングされたアドレスと一致しないアドレスキャラクタが受信されると、USART はミュートモードに入ります。USART はミュートモードに入っているため、このアドレスバイトに対して RXNE フラグはセットされず、割り込みリクエストは発行されません。

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が受信されると、USART はミュートモードから抜けます。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE ビットがセットされます。

レシーババッファにデータが存在しない場合 (USART\_SR レジスタの RXNE=0)、RWU ビットに 0 または 1 を書き込むことができます。そうでない場合、書き込みは無視されます。

アドレスマーク検出を使用したミュートモードの動作例を [図 89](#) に示します。

**図 89. アドレスマーク検出を使用したミュートモード**



**注：** パリティ制御が有効な場合、パリティビットは MSB に残っており、アドレスビットは MSB - 1 ビットに入られます。

たとえば、7 ビットデータの場合、アドレスモードとパリティ制御は次のようになります。

SB | 7 ビットデータ | ADD | PB | STB

ここで、

SB = START ビット

STB = STOP ビット

ADD = アドレスビット

PB = パリティビット

## 20.3.8 USART 同期通信

USART トランスミッタでは、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。

**注：** この機能は USART\_CK ピンを備えたデバイスでのみ使用できます。使用できるかどうかは、デバイスのピン配置を確認してください。

USART\_CK ピンは USART トランスミッタクロックの出力です。START ビットと STOP ビットの処理中には、USART\_CK ピンにクロックパルスは送信されません。USART\_CR3 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\_CR3 レジスタの CPOL ビットによってクロック極性を選択でき、同じレジスタの CPHA ビットによって外部クロックの位相を選択できます ([図 90](#)、[図 91](#)、および [図 92](#)を参照)。アイドルフレームとブレイクフレームの処理中、外部 CK クロックは起動されません。

同期モードでは、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。REN=1 の場合、データは、オーバーサンプリングなしで、USART\_CK (CPOL と CPHA に応じて立ち上がりまたは立ち下がりエッジ) でサンプリングされます。たとえ SPI プロトコルによりホールド時間が無関係であっても、ボーレート (整数ボーレートでは 1/16 ビット時間) 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。

アイドル状態時、プリアンブル処理時、およびブレイク送信時には、外部 USART\_CK クロックは起動されません。同期モードでは、USART トランスミッタは非同期モードの場合とまったく同じように機能します。しかし、USART\_CK が USART\_TX と同期すると（CPOL と CPHA に依存して）、USART\_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とは若干異なる動作をします。REN=1 の場合、データは依然として内部オーバーサンプリングクロックを使用してサンプリングされ、ボーレートクロックは USART\_CK ピンに出力されます（CPOL および CPHA に応じて、立ち上がりまたは立ち下がりエッジがデータサンプリングイベントに整列されます）。しかし、非同期モードとは異なり、データは 3 つのサンプルの多数決ではなく 1 つのサンプルで評価されます。これは NF ビットがセットされないことを意味します。

このとき、ボーレート（整数ボーレートに対して 1/16 ビット時間）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：
- 1 USART\_CK ピンは USART\_TX ピンとともに動作します。USART トランスミッタが無効にされると（TEN および REN = 0）、USART\_CK ピンと USART\_TX ピンはハイインピーダンス状態になります。
  - 2 トランスミッタとレシーバが無効にされた場合（TEN=REN=0）、クロックパルスが正常に機能するには、USART\_CR3 の 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。
  - 3 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には、TEN ビットと REN ビットを同じ命令でセットすることをお勧めします。
  - 4 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（USART\_CK は常に出力です）。
  - 5 このセクションに記載されているデータは、USART\_BRR2 レジスタの USART\_DIV[3:0] ビットが 0 に保持されているときのみ有効です。そうでない場合、セットアップ時間とホールド時間はビット時間の 1/16 ではなく、ビット時間の 4/16 となります。

このオプションによって、非同期通信の機能を失わずに（他の非同期トランスミッタおよびレシーバと通信可能）、シフトレジスタから構成されるペリフェラルをシリアルに制御できます。

図 90. USART の同期送信の例

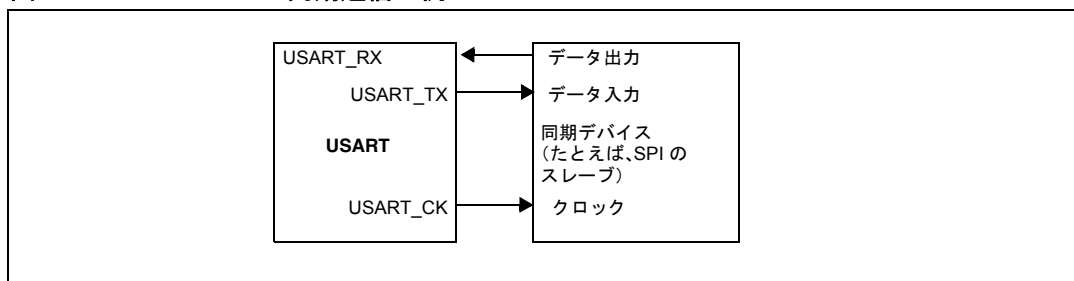




図 91. USART データクロックタイミング図 (M=0)

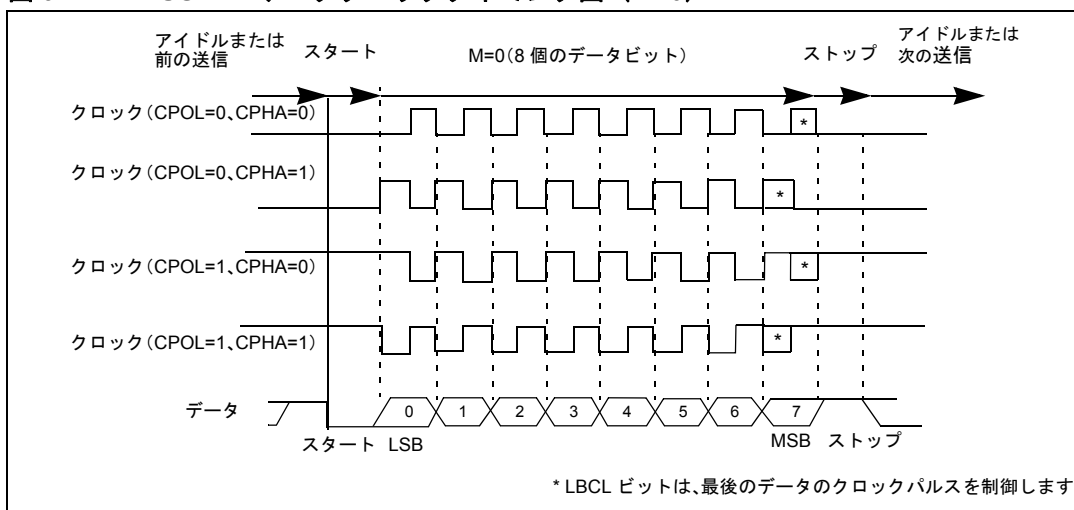


図 92. USART データクロックタイミング図 (M=1)

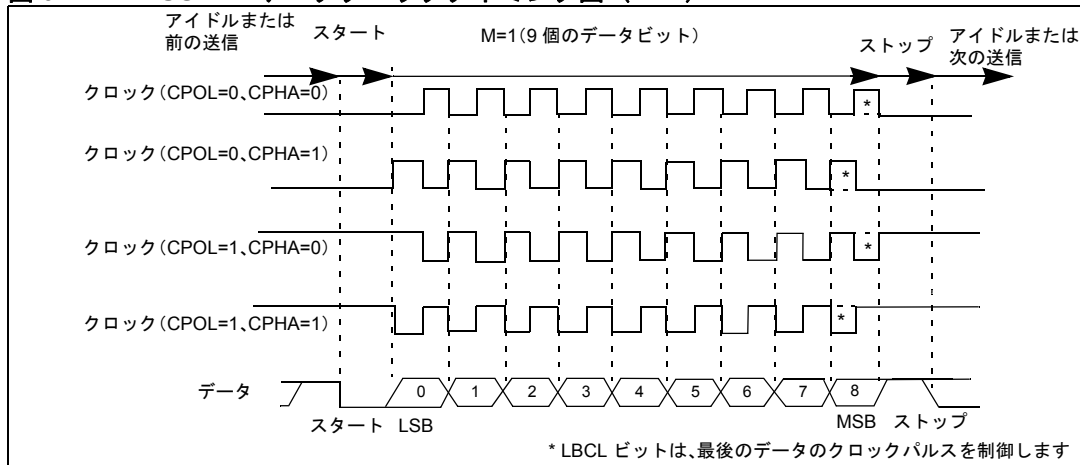
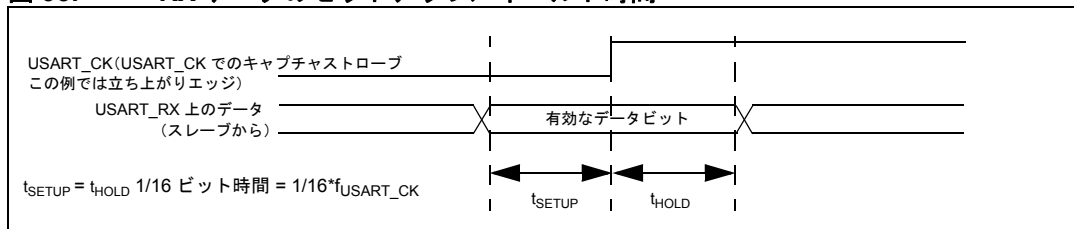


図 93. RX データのセットアップ/ホールド時間





20.4 USART 低電力モード

表 49. 低電力モードでの USART インタフェースの動作

モード	説明
WAIT	USART には影響ありません。 USART 割り込みによって、デバイスは WAIT モードから抜けます。
HALT	USART レジスタは凍結されます。 HALT モードでは、USART は HALT モードが終了するまで送信／受信を停止します。

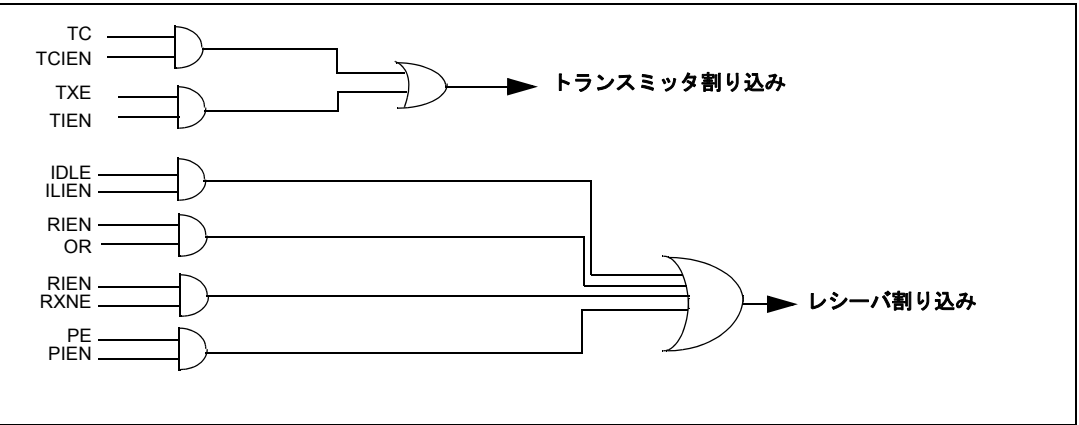
20.5 USART 割り込み

表 50. USART 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット	WAIT 終了	HALT 終了
送信データレジスタエンプティ	TXE	TIEN	あり	なし
送信完了	TC	TCIEN	あり	なし
受信データ読み出し可能	RXNE	RIEN	あり	なし
オーバーランエラー検出	OR		あり	なし
アイドルライン検出	IDLE	ILIEN	あり	なし
パリティエラー	PE	PIEN	あり	なし

- 注： 1 USART 割り込みイベントは、2 つの割り込みベクタに接続されます (図 94 を参照)。
- a) 送信完了または送信データレジスタエンプティ割り込み。
  - b) 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フル、パリティエラー割り込み、およびノイズフラグ。
- 2 これらのイベント時、対応するイネーブル制御ビットがセットされ、CCR レジスタの割り込みマスクがリセットされていた場合、割り込みが生成されます (RIM 命令)。

図 94. USART 割り込みマッピング図



## 20.6 USART レジスタ

### 20.6.1 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 : 0xC0

7	6	5	4	3	2	1	0
TXE	TC	RXNE	IDLE	OR	NF	FE	PE
r	rc_w0	r	r	r	r	r	r

#### 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たとき、ハードウェアによってセットされます。USART\_CR2 レジスタの TIEN ビット=1 の場合、割り込みが生成されます。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した。

#### ビット 6 TC : 送信完了

TC ビットは、データを含んでいるフレームの送信が完了し、TXE ビットがセットされている場合に、ハードウェアによってセットされます。USART\_CR2 レジスタの TCIE=1 である場合、割り込みが生成されます。

TC ビットは、ソフトウェアシーケンス (USART\_SR レジスタの読み出しに続く USART\_DR レジスタへの書き込み)、またはこのビットを 0 にプログラミングすることによってクリアされます。このクリアシーケンスは、マルチパ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たときに、ハードウェアによってセットされます。USART\_CR2 レジスタの RIEN=1 である場合、割り込みが生成されます。USART\_DR レジスタの読み出しによってクリアされます。

- 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<sup>(1)</sup>

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\_CR2 レジスタの ILIE=1 である場合、割り込みが生成されます。ソフトウェアシーケンスによってクリアされます (USART\_SR レジスタの読み出しに続く USART\_DR レジスタの読み出し)。

- 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。

#### ビット 3 OR : オーバーランエラー<sup>(2)</sup>

このビットは、RXNE=1 のときに、シフトレジスタに受信中のワードを 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\_CR2 レジスタの RIEN=1 である場合、割り込みが生成されます。ソフトウェアシーケンスによってクリアされます (USART\_SR レジスタの読み出しに続く USART\_DR レジスタの読み出し)。

- 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。

#### ビット 2 NF : ノイズフラグ<sup>(3)</sup>

このビットは、受信フレームでノイズ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。ソフトウェアシーケンスによってクリアされます (USART\_SR レジスタの読み出しに続く USART\_DR レジスタの読み出し)。

- 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。

- ビット 1 **FE** : フレーミングエラー <sup>(4)</sup>

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。ソフトウェアシーケンスによってクリアされます (USART\_SR レジスタの読み出しに続く USART\_DR レジスタの読み出し)。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。
- ビット 0 **PE** : パリティエラー

このビットは、レシーバモードでパリティエラーが発生したときに、ハードウェアによってセットされます。ソフトウェアシーケンスによってクリアされます (ステータスレジスタの読み出しと、それに続く USART\_DR データレジスタの読み出し)。クリアする前に、RXNE フラグがセットされるまで待つ必要があります。USART\_CR1 レジスタの PIEN=1 の場合、割り込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー。
1. 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発生する) まで、IDLE ビットが再びセットされることはありません。

2. 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。

3. このビットは、割り込みを生成する RXNE ビットと同時に現れるので、割り込みを生成しません。

4. このビットは、割り込みを生成する RXNE ビットと同時に現れるので、割り込みを生成しません。現在転送中のワードによってフレームエラーとオーバーランエラーの両方が発生した場合、転送は行われ、OR ビットのみがセットされます。

20.6.2     データレジスタ (USART\_DR)

アドレスオフセット : 0x01

リセット値 : 0xXX

7	6	5	4	3	2	1	0
DR[7:0]							
rw							

- ビット 7:0 **DR[7:0]** : データ値

読み出しか書き込みかに応じて、受信または送信されたデータキャラクタを含みます。

データレジスタは、送信用 (TDR) と受信用 (RDR) の 2 個のレジスタから構成されるため、2 つの機能 (読み出しと書き込み) を実行します。

TDR レジスタは、内部バスと出力シフトレジスタとの間にパラレルインタフェースを提供します。

RDR レジスタは、入力シフトレジスタと内部バスとの間にパラレルインタフェースを提供します。

## 20.6.3 ボーレートレジスタ 1 (USART\_BRR1)

ボーレートレジスタは、トランスミッタとレシーバの両方に共通です。ボーレートは、BRR1 と BRR2 の 2 つのレジスタを使用してプログラムされます。BRR1 への書き込みによってボーカウンタが更新されるので、BRR2 への書き込みを先に行ってください（必要な場合）。

[図 87 : BRR レジスタで USART\\_DIV をコード化する方法 \(223 ページ\)](#) および [表 45 : ボーレートのプログラミングと誤差計算 \(223 ページ\)](#) を参照してください。

**注 :** **TEN または REN ビットがそれぞれ無効にされている場合、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 : 0x02

リセット値 : 0x00

7	6	5	4	3	2	1	0
USART_DIV[11:4]							
rw	rw	rw	rw	-	rw	rw	rw

ビット 7:0 **USART\_DIV[11:4]** : USART\_DIV ビット <sup>(1)</sup>

これらの 8 ビットは、16 ビット USART ディバイダ (USART\_DIV) の 2 番目と 3 番目のニブルを定義します。

1. BRR1 = 0x00 は、USART クロックが無効であることを意味します。

## 20.6.4 ボーレートレジスタ 2 (USART\_BRR2)

アドレスオフセット : 0x03

リセット値 : 0x00

7	6	5	4	3	2	1	0
USART_DIV[15:12]				USART_DIV[3:0]			
rw				rw			

ビット 7:4 **USART\_DIV[15:12]** : USART\_DIV の MSB

これらの 4 ビットは、USART ディバイダ (USART\_DIV) の MSB を定義します。

ビット 3:0 **USART\_DIV[3:0]** : USART\_DIV の LSB

これらの 4 ビットは、USART ディバイダ (USART\_DIV) の LSB を定義します。

## 20.6.5 制御レジスタ 1 (USART\_CR1)

アドレスオフセット : 0x04  
リセット値 : 0x00

7	6	5	4	3	2	1	0
R8	T8	USARTD	M	WAKE	PCEN	PS	PIEN
rw	rw	rw	rw	rw	rw	rw	rw

- ビット 7 **R8** : 受信データのビット 8  
このビットは、M=1 のときの受信ワードの 9 番目のビットを格納するために使用されます。
- ビット 6 **T8** : 送信データのビット 8  
このビットは、M=1 のときの送信ワードの 9 番目のビットを格納するために使用されます。
- ビット 5 **USARTD** : USART ディセーブル (低電力用)  
このビットがセットされると、消費電力を低減するため、USART プリスケアラと信号出力は進行中の転送動作が終わった時点で停止します。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 は有効です。  
1 : USART プリスケアラと出力は無効です。
- ビット 4 **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 : 1 START ビット、8 データビット、n STOP ビット (n は USART\_CR3 レジスタの STOP[1:0] ビットで指定)。  
1 : 1 START ビット、9 データビット、1 STOP ビット。  
**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 3 **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 2 **PCEN** : パリティ制御イネーブル  
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、送受信での現在のバイトの後で PCEN 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 1 **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 (PCEN ビットがセットされた) 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 0 **PIEN** : パリティ割り込みイネーブル  
このビットは、ソフトウェアによってセット/クリアされます。  
0 : パリティ割り込みは無効です。  
1 : USART\_SR レジスタの PE=1 のときには、パリティ割り込みが生成されます。

## 20.6.6 制御レジスタ 2 (USART\_CR2)

アドレスオフセット : 0x05  
リセット値 : 0x00

7	6	5	4	3	2	1	0
TIEN	TCIEN	RIEN	ILIEN	TEN	REN	RWU	SBK
rw	rw	rw	rw	rw	rw	rw	rw

- ビット 7 **TIEN** : トランスミッタ割り込みイネーブル  
このビットは、ソフトウェアによってセット／クリアされます。  
0 : 割り込みは禁止です。  
1 : USART\_SR レジスタの TXE=1 のときには、USART 割り込みが生成されます。
- ビット 6 **TCIEN** : 送信完了割り込みイネーブル  
このビットは、ソフトウェアによってセット／クリアされます。  
0 : 割り込みは禁止です。  
1 : USART\_SR レジスタの TC=1 のときには、USART 割り込みが生成されます。
- ビット 5 **RIEN** : レシーバ割り込みイネーブル  
このビットは、ソフトウェアによってセット／クリアされます。  
0 : 割り込みは禁止です。  
1 : USART\_SR レジスタの OR=1 または RXNE=1 のときには、USART 割り込みが生成されます。
- ビット 4 **ILIEN** : アイドルライン割り込みイネーブル  
このビットは、ソフトウェアによってセット／クリアされます。  
0 : 割り込みは禁止です。  
1 : USART\_SR レジスタの IDLE=1 のときには、USART 割り込みが生成されます。
- ビット 3 **TEN** : トランスミッタイネーブル <sup>(1)</sup> <sup>(2)</sup>  
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。
- ビット 2 **REN** : レシーバイネーブル  
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、START 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 <sup>(3)</sup> <sup>(4)</sup>  
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たときに、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。
- 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 STOP ビットの処理時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。
1. 送信中に TEN ビットに 0 パルスを与える (0 を書き込み、続いて 1 を書き込む) と、現在のワードの後にプリアンブル (アイドルライン) が送信されます。
  2. TEN がセットされると、送信が開始されるまでに 1 ビット時間の遅延が生じます。
  3. ミュートモードを選択する (RWU ビットをセットする) 前に、まず、USART はデータバイトを受信しなければなりません。そうしないと、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。
  4. アドレスマーク検出による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ときには、ソフトウェアによって RWU ビットを変更することはできません。

## 20.6.7 制御レジスタ 3 (USART\_CR3)

アドレスオフセット : 0x06

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み	STOP[1:0]			CLKEN	CPOL	CPHA	LBCL
	rw			rw	rw	rw	rw

ビット 7:6 予約済み

ビット 5:4 **STOP** : STOP ビット

これらのビットは、STOP ビットのプログラミングに使用されます。

- 00 : 1 STOP ビット
- 01 : 予約済み
- 10 : 2 STOP ビット
- 11 : 予約済み

ビット 3 **CLKEN** : クロックイネーブル

このビットによって、USART\_CK ピンを有効にできます。

- 0 : USART\_CK ピンは無効です。
- 1 : USART\_CK ピンは有効です。

ビット 2 **CPOL** : クロック極性<sup>(1)</sup>

このビットによって、USART\_CK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して機能し、希望するクロック／データ関係をもたらします。

- 0 : アイドル時には、USART\_CK は 0 になります。
- 1 : アイドル時には、USART\_CK は 1 になります。

ビット 1 **CPHA** : クロック位相<sup>(1)</sup>

このビットによって、USART\_CK ピンのクロック出力の位相を選択できます。CPOL ビットと連携して機能し、希望するクロック／データ関係をもたらします。

- 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 0 **LBCL** : 最終ビットのクロックパルス<sup>(1)(2)</sup>

このビットによって、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 USART\_CK ピンに出力する必要があるかどうかを選択できます。

- 0 : 最終データビットのクロックパルスは、USART\_CK ピンに出力されません。
- 1 : 最終データビットのクロックパルスは、USART\_CK ピンに出力されます。

- 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。
- 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。



## 20.6.8 制御レジスタ 4 (USART\_CR4)

アドレスオフセット : 0x07

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み				ADD[3:0]			
				rw			

ビット 7:4 予約済み

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。

## 20.6.9 USART レジスタマップとリセット値

表 51. USART レジスタマップ

アドレス オフセット	レジスタ名	7	6	5	4	3	2	1	0
0x00	USART_SR リセット値	TXE 1	TC 1	RXNE 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
0x01	USART_DR リセット値	DR[7:0] xxxxxxxx							
0x02	USART_BRR1 リセット値	USART_DIV[11:4] 00000000							
0x03	USART_BRR2 リセット値	USART_DIV[15:12] 0000				USART_DIV[3:0] 0000			
0x04	USART_CR1 リセット値	R8 0	T8 0	USARTD 0	M 0	WAKE 0	PCEN 0	PS 0	PIEN 0
0x05	USART_CR2 リセット値	TIEN 0	TCIEN 0	RIEN 0	ILIEN 0	TEN 0	REN 0	RWU 0	SBK 0
0x06	USART_CR3 リセット値	- 0	- 0	STOP 00		CKEN 0	CPOL 0	CPHA 0	LBCL 0
0x07	USART_CR4 リセット値	- 0	- 0	- 0	- 0	ADD[3:0] 0000			

## 21 ProxSense™ (PXS)

### 21.1 概要

ProxSense ペリフェラルは静電容量の変動を測定することによって、電極が PXS ピンに接続されたときに、アプリケーションは 1 つまたは複数の導電性の物体の近接またはタッチを検出できます。

このペリフェラルは**投影型静電容量方式**で動作します。このモードは、駆動された電極によって別の電極に転送される電荷の測定に使用されます。

**注：** ProxSense™ は、Azoteq 社の商標です。

### 21.2 ProxSense ペリフェラルの主な機能

- 独立した 10 チャンネルのレシーバによって、10 件の検出を並行して実行できます。
- レシーバの各チャンネルを 2 本の異なるピンに関連付けることができるため、アプリケーションは、実質的に最大 20 チャンネルのレシーバを持つことができます。
- 各レシーバチャンネルは、投影型静電容量の測定を行うよう、個別に設定できます。
- 各 C<sub>S</sub> コンデンサのサイズは、5 ビットの分解能で個別に設定できます。
- 各電極寄生容量補償 (EPCC) コンデンサのサイズは、8 ビットの分解能で個別に設定できます。
- 各変換の起動は外部入力と同期できるため、50 Hz/60 Hz の電源供給回路などの周期的障害が存在しても、きわめて正確な測定が可能です。
- 変換終了割り込みは、関連する割り込みフラグを持ち、CPU をアクティブ HALT モードからウェイクさせることができます。
- 最初のチャンネル完了割り込みは、関連する割り込みフラグを持ち、最初のチャンネルの検出終了後に、CPU をアクティブ HALT モードからウェイクさせることができます。
- HSI\_PXS に基づく電荷移動プロトコルを制御するクロックの周波数は、125 kHz~16 MHz の範囲でスケールリングできます。
- UP フェーズと PASS フェーズの時間は、データ取得周波数を最適化するように、個別に設定できます。
- ノイズ検出機能を有効にすると、前回の変換中にノイズがあったかどうかを判別します。
- 応答時間は、最大変換測定値を指定することによって拡張できます。これは、主にタッチキーアプリケーションに役立ちます。

## 21.3 PXS ピン


表 52. PXS ピン

名称	信号種別	摘要
PXS_RXnA/ PXS_RXnB	アナログ入出力	20 個の各レシーバは、電極電荷が転送される 10 個の内部 C <sub>S</sub> コンデンサの 1 つと関連付けられます (RXAn と RXBn は 1 つの Cs を共有します)。
PXS_TXm	デジタル出力	15 個の各送信出力は、Rx ピンに接続された別の電極に (投影型 静電容量方式で) 電荷を転送する電極を駆動できます。
PXS_RFIN	アナログ入力	無線周波数 (RF) ノイズを検出するアンテナ入力。
PXS_TRIG	デジタル入力	検出用の外部トリガ。
PXS_VREG	外部デカップリング コンデンサ	コンデンサを接続して、専用の ProxSense レギュレータを安定させる必要があります。

注 : m = 0~14 (送信出力)、n = 0~9 (レシーバ)。

## 21.4 機能の説明

### 21.4.1 概要

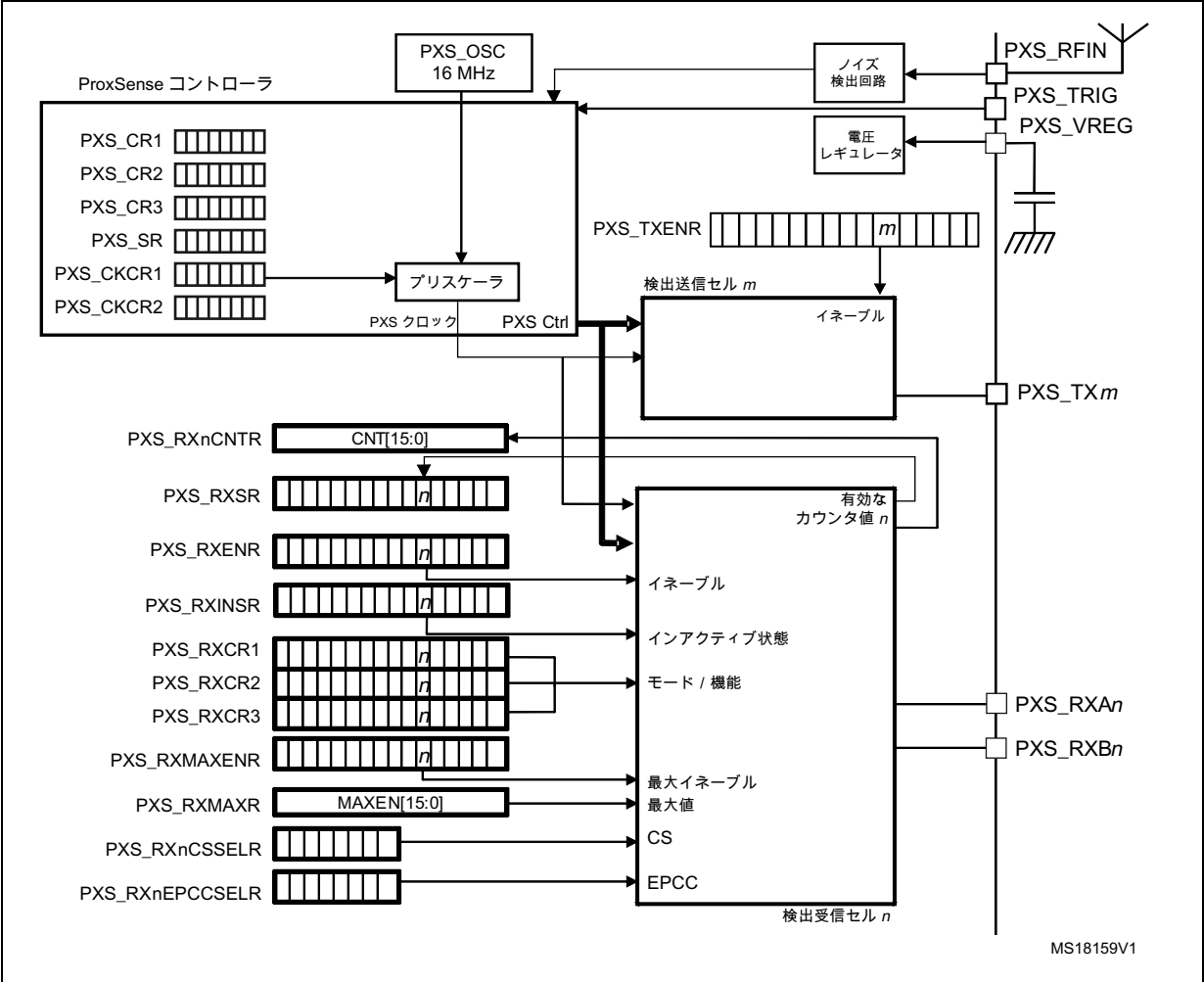
 95 に、ProxSense ペリフェラルのブロック図を示します。

10 チャネルのレシーバにより、10 個の電極の容量を並行して測定できます。各レシーバチャネルには 2 本のピン (A グループと B グループから 1 本ずつ) を関連付けられるため、20 個の電極の測定が可能になります。

各レシーバチャネルは、電極で測定した容量を 16 ビット値 (PXS\_RXnCNTR) に変換します。また、この結果を取得するために、電極から内部コンデンサ (Cs) に電荷を繰り返し転送します。結果として得られる値は、Cs の電圧が基準電圧を上回るまでに必要な電荷移動の回数です。したがって、容量が大きくなれば、値は小さくなります。

投影型静電容量モード (PXS\_RXCRn を使用して選択) では、Tx 電極と対応する Rx 電極との間の容量が測定されます。この場合、導電性の物体がこれら 2 個の電極の交差点に近づくにつれて、より大きなデータ値が取得されます。その理由は、Tx 電極と導電性物体との容量結合が増え、Rx 電極との結合が減るからです。

図 95. ProxSense のブロック図



15 個の送信出力 (m = 0~14) と最大 10 個のレシーバ (n = 0~9) があり、各セルには 2 本の PXS\_RX ピン (A と B) があります。

21.4.2 ProxSense 電荷移動シーケンス

ProxSense による検出は一連の電荷移動により構成され、それぞれの電荷移動は UP フェーズと PASS フェーズに分けられます。検出結果は、毎回の電荷移動を受け取る内部コンデンサ (Cs) が閾値電圧に達するまでに必要な電荷移動の回数です。

UP および PASS フェーズ中に発生するイベントを表 53 に規定します。

表 53. UP および PASS フェーズ中の動作

モード	UP フェーズ	PASS フェーズ
投影型負荷容量	<ul style="list-style-type: none"><li>- Tx ラインは強制的にローになります</li><li>- Rx ラインはサンプリングされた電圧に保持されます (前回の PASS フェーズの最後にサンプリング)。</li></ul>	<ul style="list-style-type: none"><li>- Tx ラインは強制的にハイになります</li><li>- Cs は、投影型負荷容量検出により Rx ラインに転送された電荷を吸収します。</li></ul>



## 電荷移動シーケンスの設定

PXS\_CKCR1 レジスタの PRESC[2:0] を使用して ProxSense クロックをプリスケールし、UP および PASS フェーズを生成するための基準とすることができます。

2 つのフェーズの長さは、UPLEN[2:0] と PASSLEN[2:0] (PXS\_CKCR2) を使用して個別に設定できます。さらに柔軟性を高めるには、INCPHASE ビット (PXS\_CKCR1) をセットすることで、UP および PASS フェーズの長さを、それぞれ ProxSense クロックサイクルの半分増やすことができます。

非重複期間 (UP フェーズと PASS フェーズ間のデッドタイム) の長さは、ANADEAD (PXS\_CKCR1) によって決まります。ANADEAD = 0 の場合、UP フェーズと PASS フェーズの間に 1/2 サイクルのデッドタイムがあります。ANADEAD = 1 の場合、デッドタイムは基本的に 0 サイクルであり、重複しないことを保証するため、UP サイクルと PASS サイクルは、それぞれおよそ 10 ns だけ短縮されます。ANADEAD に“1”を設定すると、UP フェーズは 1 サイクルだけ増加します。

ANADEAD ビットと INCPHASE ビットに“0”を設定し、UPLEN ビットと PASSLEN ビットに“1”を設定したりリセット設定では、電荷移動サイクルは  $4 t_{\text{PXS}}$  だけ持続します。たとえば、PRESC[2:0] = “111” のとき、 $f_{\text{PXS}} = 16 \text{ MHz}$  であり、各電荷移動サイクルは 250 ns 持続します。

## 21.5 ProxSense の操作

### 21.5.1 PXS のオン／オフ制御

ProxSense を有効／無効にするには、PXS\_CR1 レジスタの PXSEN ビットをそれぞれセット／リセットします。ProxSense を有効にする前に、PXSEN = 0 のときにのみ書き込み可能な特定の PXS レジスタを初期化する必要があります。

### 21.5.2 PXS の初期化

PXS レジスタへの書き込みを行う前に、CLK\_PCKENR1 レジスタで対応するペリフェラルクロックを有効にする必要があります。

#### ProxSense を有効にする前の初期化

対応するペリフェラルクロックを有効にした後、PXS\_CKCR1 レジスタのクロック周波数と、PXS\_CKCR2 の UP および PASS フェーズの長さを設定する必要があります。またこの段階で、PXS\_CKCR1 の ANADEAD と INCPHASE を選択することを推奨します。

安定化時間、変換完了閾値、およびサンプル&ホールド強度は、PXS\_CR3 で選択する必要があります。ノイズ検出を使用する場合は、この段階で、PXS\_CR2 レジスタで有効にする必要があります。

## タッチ/近接検出の初期化

以下の初期化は、PXS\_CR1 で PXS が有効になっているときに行うことができ、変換の進行中には行うことができません。

- オプションで、割り込み、同期機能、および Rx 結合を PXS\_CR2 で有効にします。
- PXS\_RXCR1/2/3 でレシーバのモードを選択します。標準のアプリケーションでは、この操作は、すべての検出に対して 1 回のみ行われます。
- PXS\_RXINSR レジスタで Rx が無効（インアクティブ）にされたときのピンの状態を選択します。無効にされた PXS\_TXm ピンの状態は、対応する GPIO によって設定されます。たとえば、PXS\_TX0 が無効なときにグラウンドと接続するには、PD0 を出力ローレベルとして設定する必要があります（PD\_DR = 0bxxxxxx0）。TX0 をハイインピーダンスに設定するには、PD0 を出力オープンドレインハイレベルとして設定する必要があります（PD\_DR = 0bxxxxxxx1）。
- PXS\_MAXR で最大カウント限度を設定し、PXS\_MAXENR で選択されたレシーバに対して有効にします。
- PXS\_RXnCSSELR レジスタでサンプリングコンデンサ（Cs）のサイズを選択します。特にいくつかのレシーバ（たとえば、スライダ、ホイール、スクリーン）の測定結果から検出位置が補間される場合は、同じオブジェクトに属するすべてのレシーバに同じ Cs サイズを選択することを推奨します。
- PXS\_CR2 レジスタでレシーバグループチャネル A または B を選択します。
- PXS\_RXENR レジスタと PXS\_TXENR レジスタの対応するビットをセットすることによって、それぞれレシーバとトランスミッタを選択します。
- PXS\_RXnEPCSELR レジスタでの検出設定に基づいて、電極の寄生容量と補償するサイズを選択します。この補償容量は、カップル（Rx、Tx）ごとにアレンジする必要があります。

最後の 3 つのアクションは、異なるトランスミッタによる検出の前に、毎回繰り返してください。

## 21.5.3 PXS 検出の起動

### 手動起動

検出は、PXS\_CR1 の START ビットをセットすることによって、有効なすべてのレシーバで並行して起動されます。

### 同期起動

PXS\_TRIG ピンに接続された外部信号と検出を同期させるには、PXS\_CR2 で SYNC 機能を有効にして信号エッジを選択する必要があります。PXS\_CR1 の START ビットをセットすることによって検出が有効になり、選択したエッジが PXS\_TRIG ピンで次に発生した時点で検出が起動されます。同期検出が完了すると、新しい検出を起動するには、START ビットを再びセットする必要があります。

アクティブ HALT モード中には PXS\_TRIG 入力での遷移が認識されないため、アクティブ HALT モード中に起動を同期させることはできません。



## 21.5.4 PXS 検出結果

アプリケーションは、PXS\_ISR の EOCF ビットをポーリングすることによって、検出が完了したことを確認できます。あるいは、アプリケーションは、PXS\_ISR の FCCF ビットをポーリングして、少なくとも 1 つのチャンネルがその取得を終了したことを確認できます。それ以外では、EOCF や FCCF をポーリングする代わりに、対応する割り込みを有効にすることができます。PXS\_RXSR レジスタと PXS\_RXENR レジスタが同じである場合、選択されたすべての Rx は正しく取得されています。各チャンネル結果は、PXS\_RXnCNTR で確認できます。

PXS\_ISR のステータスフラグ EOCF と FCCF は、再び PXS\_CR1 の START ビットをセットすることによってクリアされます。

ノイズ検出が有効である場合、データを処理する前にそのフラグをチェックする必要があります。

## 21.5.5 PXS 検出の停止

実行中の変換を停止する唯一の方法は、PXSEN に“0”を書き込むことです。この場合、途中で停止された変換の結果は保証されません。

注： CIPF は、PXSEN がクリアされた後 8  $\mu$ s の間は“1”のままです。

## 21.6 特殊機能

### 21.6.1 サンプリングコンデンサの選択

サンプリングコンデンサが内蔵されていて、その容量は、接続される電極に合わせてアレンジできます。32 種類のサンプリング容量 ( $C_S$ ) があり、PXS\_RXnCSSELR レジスタを使用してレシーバごとに選択できます。

### 21.6.2 EPCC の選択

電極寄生容量補償 (EPCC) は、まず  $C_S$  容量値を微調整することによって、すべてのレシーバ変換結果が同じ目標値に近似するようにします。次に、これを使用して、環境条件の変化を補償できます。この補償によってシステムは変化への感度が高くなるため、電極の小さな容量変化を検出可能にします。

EPCC のサイズは、PXS\_RXnEPCCSELR レジスタで調整されます。

### 21.6.3 RF の検出

RF からの影響を防止するオンチップ保護に加えて、本製品は、RF 干渉を検出する機能も提供します。

PXS\_CR1 の PXS\_EN ビットをセットすることによって ProxSense を有効にする前に、PXS\_RFIN ピンにアンテナを接続し、PXS\_CR2 レジスタの NOISEDETEN ビットをセットする必要があります。これによって、毎回の検出後に、アプリケーションは、PXS\_ISR レジスタの NOISEDETF ビットをチェックして、結果を破棄すべきかどうかを検証できます。NOISEDETF ビットは、アプリケーションによってリセットする必要があります。



21.7 低電力モード

21.7.1 低電力モードでの動作

表 54. 低電力モードでの動作

モード	説明
WAIT	PXS インタフェースはアクティブです。
ACTIVE-HALT	割り込みイベントによって、デバイスは WAIT および ACTIVE-HALT から抜けま す。
HALT	PXS インタフェースはアクティブではありません。

21.7.2 消費電力の低減

他のペリフェラルに関しては、ペリフェラルが使用されないときに ProxSense を無効にし、CLK\_PCKENiR レジスタでそのクロックを切り替えることによって、消費電力を低減できます。

LOW\_POWER ビット

PXS\_CR1 の LOW\_POWER ビットをセットすると、変換がアクティブな間だけ、ProxSense のアナログ部分がパワーオンされます。これによって、START ビットがセットされた直後に安定化までの時間が発生するため、時間制約のある連続変換の場合は、検出間に生じるこの安定化時間を避けるため、START ビットをセットする直前に LOW\_POWER ビットをクリアすることを推奨します。

安定化時間

ProxSense ペリフェラルは、その安定化を自動的に管理します。言い換えると、ソフトウェアや SYNC 機能が安定化の前に変換の起動を要請した場合、その変換は延期され、安定化が完了するとすぐに自動的に開始されます。

ProxSense をパワーオンしてからアナログ回路が安定するまでに必要な時間は、変換中に ProxSense 回路によってシンクされる電流の量に反比例します。

例：

- 1つのチャンネルのみがアクティブで、4 MHz の電荷移動周波数、BIAS = “10”、Tx ラインに 10 pF という条件では、さらに 420  $\mu$ s の安定化時間が必要になります。
- 設定されたいくつかのチャンネルがアクティブである場合、さらに 120  $\mu$ s の安定化時間が必要です。

安定化の時間は、PXS\_CR3 レジスタの STAB[1:0] ビットを使用して減らすことができます。これによって、アプリケーションの全体的な消費電流を減らすことができます。デフォルトでは、1.7 ms の時間が選択されます。

## 消費電力を減らすその他のパラメータ

ProxSense モジュールにおける消費電力の重要な要素は変換時間です。最適な消費電力を実現するには、感度に悪影響を与えない最小の検出時間を選択する必要があります。

消費電力をさらに減らすには、PXS\_RXINSR レジスタをリセットすることによって 無効にされた Rx ピンを接地し、Tx チャンネルに接続された GPIO ピンを、ProxSense によって駆動されていないときには出力ローレベルとして設定することを推奨します。

Rx ラインが相互に情報をやり取りしていない場合、PXS\_CR2 の RXCOUPLING ビットをリセットすることによって Rx 結合を無効にすると、対応する  $C_S$  が閾値電圧 ( $V_{THR}$ ) に到達するとすぐに Rx ライン起動が停止するため、消費電力も減ります。

PXS\_CR3 の BIAS[1:0] ビットを使用してサンプル&ホールド強度を設定することで、送信ライン (Tx) がローに遷移する間、最終的な放電電圧を維持できる最小値を設定します。

## 21.8 PXS 割り込み

ProxSense には 2 つの割り込み要求があります。

- 変換終了 (EOC) 割り込み: 有効にされたすべてのチャンネルが検出を完了すると有効になります。
- 最初のチャンネル変換完了 (FCC) 割り込み: 1 つのチャンネルが検出を完了するとすぐに有効になります。

割り込みソースごとの割り込みリクエストを有効にするには、PXS\_CR2 で必要な EOCITEN または FCCITEN 割り込みイネーブルビットをセットします。

割り込みを有効にする前に、それまでに完了した検出による割り込みのトリガを避けるため、PXS\_ISR のステータスフラグ EOCF と FCCF をクリアすることを推奨します。

割り込みルーチンを抜ける前に、PXS\_ISR の対応するビットに“0”を書き込むことによって、ステータスフラグ EOCF と FCCF をクリアする必要があります。

**注:** *これらの割り込みによって、デバイスが WAIT モードと ACTIVE-HALT モードから抜けることはできませんが、HALT モードから抜けることはできません。*

21.9 ProxSense レジスタ

21.9.1 ProxSense 制御レジスタ 1 (PXS\_CR1)

アドレスオフセット : 0x00  
リセット値 : 0x00

7	6	5	4	3	2	1	0
PXSEN	START	LOW_POWER	予約済み				
rw	r0_w	rw					

- ビット 7 **PXSEN** : ProxSense イネーブル
- 0 : ProxSense 機能は無効になり、ProxSense アナログ回路はパワーダウンされます。
  - 1 : ProxSense 機能は有効になります。
- このビットは、ソフトウェアによってのみ 0 にリセットされます。
- 注 :** 変換中にソフトウェアによって PXSEN が 0 に設定された場合、変換は停止され、変換結果は保証されません。
- ビット 6 **START** : 変換開始
- 0 : “0”を書き込んでも効果はありません。
  - 1 : 変換開始
- ProxSense ペリフェラルが有効であり (PXSEN = 1)、変換がまだ実行中でない (CIPF = 0) 場合、このビットに 1 を書き込むと、新しい変換を開始します。START ビットは常に “0”として読み出されます。
- 注 :** 変換がすでに実行中である (CIPF = 1) とき、1 を書き込んでも効果はありません。
- 注 :** ProxSense ペリフェラルが無効である (PXSEN = 0) とき、1 を書き込んでも効果はありません (同じ書き込み命令で PXSEN がセットされている場合を除きます)。
- ビット 5 **LOW\_POWER** : 低電力モード
- 0 : PXSEN = 1 の時は、ProxSense アナログはパワーオンされます。
  - 1 : 変換中 (CIPF = 1 および SYNC PF = 0) にのみ、ProxSense アナログはパワーオンされます。
- ビット 4:0 予約済み

## 21.9.2 ProxSense 制御レジスタ 2 (PXS\_CR2)

アドレスオフセット : 0x01  
リセット値 : 0x00

7	6	5	4	3	2	1	0
EOCITEN	FCCITEN	NOISEDETEN	予約済み	RXGROUP	RXCOUPLING	SYNCEN	SYNCEDGE
rw	rw	rw		rw	rw	rw	rw

ビット 7 **EOCITEN** : 変換終了割り込みイネーブル

- 0 : 変換終了割り込みは無効です。
- 1 : 変換終了割り込みは有効です。

有効にされると、EOCF = 1 のとき (すなわち、レシーバですべての変換が完了したとき)、割り込みが発生します。EOCF は、割り込みサービスルーチンの実行中にクリアする必要があります。

ビット 6 **FCCITEN** : 最初の変換完了割り込みイネーブル

- 0 : 最初の変換完了割り込みは無効です。
- 1 : 最初の変換完了割り込みは有効です。

有効にされると、FCCF = 1 のとき (すなわち、有効にされたレシーバの 1 つで最初の変換が完了したとき)、割り込みが発生します。たとえ FCCF が事前にクリアされた場合でも、それ以降の変換が他のチャンネルで完了しても割り込みはトリガされません。ソフトウェアは、PXS\_RXSR レジスタを読み出すことによって、どのチャンネルが変換完了したかを判定できます。

ビット 5 **NOISEDETEN** : ノイズ検出イネーブル

- 0 : RF ノイズ検出は無効です。
- 1 : RF ノイズ検出は有効です。

NOISEDETEN = 1 のとき、RF ノイズ検出回路は有効です。RF ノイズを正しく検出するため、PXS\_RFIN ピンにアンテナを接続してください。PXS\_ISR の NOISEDETF は、変換中にノイズが検出されたかどうかを示します。

**注 :** このビットは、**PXSEN = 0** のときにのみ変更できます。

ビット 4 予約済み

ビット 3 **RXGROUP** : Rx グループ選択

- 0 : Rx "A" グループ (RX0A~RX9A) がアクティブです。
- 1 : Rx "B" グループ (RX0B~RX9B) がアクティブです。

**注 :** **RXGROUP = 0** のとき、Rx "B" グループは (**PXS\_RXINSR** によって指定されるように) インアクティブ状態であり、逆の場合も同様です。

**注 :** このビットは、**CIPF = 0** のときにのみ変更できます。

ビット 2 **RXCOUPLING** : レシーバライン間のカップリング低減

- 0 : Rx ライン上の電荷移動サイクルは、その対応する Cs が閾値電圧 ( $V_{THR}$ ) に到達すると停止します。
- 1 : 有効にされた (**RXEN $n$**  = 1) 各 Rx は、有効にされたすべての Rx チャンネルが閾値電圧 ( $V_{THR}$ ) に到達するかタイムアウトが発生するまで、トグルし続けます。

Rx ラインが相互に大きなカップリングを持つアプリケーションの場合、このビットをセットして、隣接する Rx チャンネルが閾値電圧に到達した後、特定の Rx ラインでの寄生容量が増加しないようにすると有益場合があります。

このビットを 0 にリセットすると、いくつかの Rx ラインが短時間だけトグルすることで消費電力を減らせる可能性があります。

**注 :** このビットは、**CIPF = 0** のときにのみ変更できます。

ビット 1 **SYNCEN** : 同期 (SYNC) 機能イネーブル

- 0 : SYNC 機能を無効にします。
- 1 : SYNC 機能を有効にします。

ACTIVE-HALT モード中には PXS\_TRIG 入力での遷移が認識されないため、この機能は ACTIVE-HALT モードでは使用できません。

- ビット 0 **SYNCEGE** : 同期エッジ選択
- 0 : PXS\_TRIG 入力の立ち下がりエッジで変換を起動 (SYNCEN = 1 のとき)
  - 1 : PXS\_TRIG 入力の立ち上がりエッジで変換を起動 (SYNCEN = 1 のとき)

## 21.9.3 ProxSense 制御レジスタ 3 (PXS\_CR3)

アドレスオフセット : 0x02  
リセット値 : 0x04

7	6	5	4	3	2	1	0
STAB[1:0]		BIAS[1:0]		VTHR[3:0]			
rw		rw		rw			

- ビット 7:6 **STAB[1:0]** : ProxSense パワーオン後の安定化時間の選択
- 00 : 1.7 ms。すべての設定に使用できます。
  - 01 : 500  $\mu$ s。少なくとも 1 つのチャンネルが有効にされたときに使用でき、中程度の消費電力 (>170  $\mu$ A) を持つ。
  - 10 : 120  $\mu$ s。いくつかのチャンネルが投影型モードで有効にされたときに使用できます。
  - 11 : 予約済み。この選択は使用しないでください。
- この安定化時間は、毎回の ProxSense パワーオン後に (LOW\_POWER = 1 のときすべての変換で)、最初の変換の開始時に適用されます。
- ビット 5:4 **BIAS[1:0]** : サンプル&ホールド強度の選択
- 00 : 2.5  $\mu$ A
  - 01 : 5  $\mu$ A
  - 10 : 7.5  $\mu$ A
  - 11 : 10  $\mu$ A
- これは、送信ラインがローに遷移する (戻る) 間、PXS\_RXn ピンを最終 (放電) 電圧に維持する、オペアンプの公称バイアス電流です。この機能は、投影型の測定にのみ有効です。検出容量 (各 Rx ラインと各 Tx ラインの間の容量) が大きい場合、さらに高いバイアスが必要です。
- 注 : このレジスタは、PXSSEN = 0 のときにのみ変更できます。
- ビット 3:0 **VTHR[3:0]** : 閾値電圧 ( $V_{THR}$ ) の選択
- 0000: 0.5 x  $V_{REG}$
  - ...
  - 0100: 0.61 x  $V_{REG}$  (リセットの後のデフォルト)
  - ...
  - 1111: 0.9 x  $V_{REG}$
- 閾値電圧は、VTHR[3:0] を使用して 0.5 \*  $V_{REG}$  ~ 0.9 \*  $V_{REG}$  の値に設定できます。
- $$V_{REG} = 0.02667 \times V_{THR} \times V_{REG} + V_{REG}/2$$
- 所定のサンプリングコンデンサ ( $C_S$ ) の電圧が  $V_{THR}$  に到達すると、対応する受信チャンネルの変換は完了します。
- 注 : このレジスタは、PXSSEN = 0 のときにのみ変更できます。

## 21.9.4 ProxSense 割り込みおよびステータスレジスタ (PXS\_ISR)

アドレスオフセット : 0x04  
リセット値 : 0x00

7	6	5	4	3	2	1	0
EOCF	FCCF	NOISEDETf	CIPF	SYNCPF	SYNC_OVRF	予約済み	
rc	rc	rc	r	r	r		

### ビット 7 EOCF : 変換終了フラグ

- 0 : 変換終了は発生していません。
- 1 : 変換終了が発生しました。

有効にされたそれぞれの Rx チャンルが変換を終了し、結果の準備ができると、このステータスビットはハードウェアによってセットされます。

変換が開始された (CIPF が“1”になった) とき、または PXSEN = 0 のとき、EOCF はハードウェアによってクリアされます。

EOCITEN = 1 の場合、EOCF が 1 に遷移すると、ProxSense 割り込みが発生します。EOCF は、割り込みサービスルーチンを抜ける前に、ソフトウェアによってクリアされる必要があります。

**注 :** このフラグをクリアするとき、他のフラグビットを誤って消去することを避けるため、このレジスタに 0x60 を書き込むことを推奨します。BRES 命令を使用すると、読み出しと書き込みの間にイベントが発生することがあるため、好ましくありません。

### ビット 6 FCCF : 最初の変換完了フラグ

- 0 : 最初の変換完了割り込みは発生していません。
- 1 : 最初の変換完了割り込みが発生しました。

変換が開始された後、初めてチャンネル変換が完了すると、このステータスビットは、ハードウェアによってセットされます。他のチャンネルでそれ以降の変換が完了しても、このビットはセットされません。

変換が開始された (CIPF が“1”になった) とき、または PXSEN = 0 のとき、FCCF はハードウェアによってクリアされます。

FCCITEN = 1 の場合、FCCF が立てられると、ProxSense 割り込みが発生します。FCCF は、割り込みサービスルーチンを抜ける前に、ソフトウェアによってクリアされる必要があります。

**注 :** このフラグをクリアするとき、他のフラグビットを誤って消去することを避けるため、このレジスタに 0xA0 を書き込むことを推奨します。BRES 命令を使用すると、読み出しと書き込みの間にイベントが発生することがあるため、好ましくありません。

### ビット 5 NOISEDETf : ノイズ検出フラグ

- 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。

変換中に RF ノイズが検出された場合、このステータスビットはハードウェアによってセットされます。NOISEDETf はソフトウェアによってクリアされます。

**注 :** このフラグをクリアするとき、他のフラグビットを誤って消去することを避けるため、このレジスタに 0xC0 を書き込むことを推奨します。BRES 命令を使用すると、読み出しと書き込みの間にイベントが発生することがあるため、好ましくありません。

### ビット 4 CIPF : 変換実行中フラグ

- 0 : 実行中の変換はありません。
- 1 : 変換を実行中です。

このステータスビットは、START ビットがセットされたときにハードウェアによってセットされ、変換が完了するとハードウェアによってクリアされます。CIPF は読み出し専用です。

**注 :** CIPF = 1 のとき、多くの制御レジスタはロックされます (読み出し専用)。

**注 :** 変換中にソフトウェアによって PXSEN が 0 に設定された場合、変換は停止します。この場合、CIPF は 8  $\mu$ s の間 “1” のままで、その後 “0” に遷移します。



ビット 3 **SYNCPF** : 同期 (SYNC) ペンディングフラグ

- 0 : ProxSense ペリフェラルは、PXS\_TRIG 入力のアクティブエッジを待ちません。
- 1 : ProxSense ペリフェラルは、PXS\_TRIG 入力のアクティブエッジを待ちます。

START ビットがセットされ、SYNC 機能が有効になると (SYNCEN = 1 および PXSEN = 1)、このステータスビットはハードウェアによってセットされます。選択されたエッジ (SYNCEDGE によって指定) が発生して変換が開始されると、このビットはハードウェアによってクリアされます。

ビット 2 **SYNC\_OVRF** : 同期 (SYNC) オーバーフローフラグ

- 0 : SYNC オーバーフローは発生していません。
- 1 : SYNC オーバーフローが発生しています。

SYNCEN = 1、SYNCPF = 0、および PXSEN = 1 の間に、選択されたエッジが PXS\_TRIG ピンで発生した場合、「SYNC オーバーフロー」が発生します。このようなエッジは無視され、ステータスビット SYNC\_OVRF がセットされます。SYNC\_OVRF = 1 の場合、PXS\_TRIG のアクティブエッジで変換が開始されなかったことを示すだけであり、必ずしも問題があるわけではありません。

ビット 1:0 予約済み

## 21.9.5 ProxSense クロック制御レジスタ 1 (PXS\_CKCR1)

アドレスオフセット : 0x06  
リセット値 : 0x30

7	6	5	4	3	2	1	0
予約済み	PRESC[2:0]			予約済み		ANADEAD	INCPHASE
	rw					rw	rw

ビット 7 予約済み

ビット 6:4 **PRESC[2:0]** : ProxSense クロックの周波数選択

- 000 : 125 kHz
- 001 : 250 kHz
- 010 : 500 kHz
- 011 : 1 MHz (リセットの後のデフォルト)
- 100 : 2 MHz
- 101 : 4 MHz
- 110 : 8 MHz
- 111 : 16 MHz

**注 :** これらのビットは、PXSEN = 0 のときにのみ変更できます。

ビット 3:2 予約済み

ビット 1 **ANADEAD** : ハーフサイクルのデッドタイムではなく短いアナログ遅延を使用して、UP フェーズと PASS フェーズの間のデッドタイム (非重複インターバル) を確保します。

- 0 : UP フェーズと PASS フェーズの間のデッドタイムは、ProxSense クロックサイクルの半分です。
- 1 : UP フェーズと PASS フェーズ間のデッドタイムは約 10 ns であり、UP フェーズの長さは 1 サイクルだけ増えます。

**注 :** このビットは、CIPF = 0 のときにのみ変更できます。

ビット 0 **INCPHASE** : UP フェーズと PASS フェーズのそれぞれの長さを、クロックサイクルの半分だけ増やします。

- 0 : 電荷移動フェーズの長さは、次のとおりです。
  - UP フェーズ : ProxSense クロックの 0.5 + UPLEN + ANADEAD サイクル
  - PASS フェーズ : ProxSense クロックの 0.5 + PASSLEN サイクル
- 1 : 電荷移動フェーズの長さは、2 分の 1 サイクルだけ長くなります。
  - UP フェーズ : ProxSense クロックの 1 + UPLEN + ANADEAD サイクル
  - PASS フェーズ : ProxSense クロックの 1 + PASSLEN サイクル

**注 :** このビットは、CIPF = 0 のときにのみ変更できます。



## 21.9.6 ProxSense クロック制御レジスタ 2 (PXS\_CKCR2)

アドレスオフセット : 0x07

リセット値 : 0x11

7	6	5	4	3	2	1	0
予約済み	UPLEN[2:0]			予約済み	PASSLEN[2:0]		
	rw				rw		

ビット 7 予約済み

ビット 6:4 **UPLEN[2:0]** : UP フェーズの長さ<sup>(1)</sup>

UP フェーズのサイクル数 :  $0.5 + \text{UPLEN} + (0.5 * \text{INCPhase}) + \text{ANADEAD}$ 。

**注 :** これらのビットは、**PXSEN = 0** のときにのみ変更できます。

ビット 3 予約済み

ビット 2:0 **PASSLEN[2:0]** : PASS フェーズの長さ<sup>(1)</sup>

PASS フェーズのサイクル数 :  $0.5 + \text{PASSLEN} + (0.5 * \text{INCPhase})$ 。

**注 :** これらのビットは、**PXSEN = 0** のときにのみ変更できます。

- UPLEN と PASSLEN に関しては、値が 4 以下の場合、UP フェーズ =  $0.5 + \text{UPLEN} + (0.5 * \text{INCPhase}) + \text{ANADEAD}$  であり、値が 4 を超える場合、UP フェーズ =  $0.5 + 2 * (\text{UPLEN} - 2) + (0.5 * \text{INCPhase}) + \text{ANADEAD}$  です。したがって、UPLEN = 5 の場合、UP フェーズ =  $0.5 + 8 + (0.5 * \text{INCPhase}) + \text{ANADEAD}$  です。

## 21.9.7 レシーバイネーブルレジスタ Rx[9:8] (PXS\_RXENRH)

アドレスオフセット : 0x08

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						RXEN9	RXEN8
						rw	rw

## 21.9.8 レシーバイネーブルレジスタ Rx[7:0] (PXS\_RXENRL)

アドレスオフセット : 0x09

リセット値 : 0x00

7	6	5	4	3	2	1	0
RXEN7	RXEN6	RXEN5	RXEN4	RXEN3	RXEN2	RXEN1	RXEN0
rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:10 予約済み

ビット 9:0 **RXENn** : レシーバチャネル *n* イネーブル

0 : Rx チャネル *n* は無効であり、そのインアクティブ状態は RXINSn によって決定されます。

1 : Rx チャネル *n* は有効であり、その機能モードは RXCR1/2/3\_n によって決定されます。

**注 :** これらのビットは、**CIPF = 0** のときにのみ変更できます。

## 21.9.9 レシーバ制御レジスタ 1 Rx[9:8] (PXS\_RXCR1H)

アドレスオフセット : 0x0A

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						RXCR1_9	RXCR1_8
						rw	rw

## 21.9.10 レシーバ制御レジスタ 1 Rx[7:0] (PXS\_RXCR1L)

アドレスオフセット : 0x0B

リセット値 : 0x00

7	6	5	4	3	2	1	0
RXCR1_7	RXCR1_6	RXCR1_5	RXCR1_4	RXCR1_3	RXCR1_2	RXCR1_1	RXCR1_0
rw	rw	rw	rw	rw	rw	rw	rw

## 21.9.11 レシーバ制御レジスタ 2 Rx[9:8] (PXS\_RXCR2H)

アドレスオフセット : 0x0C

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						RXCR2_9	RXCR2_8
						rw	rw

## 21.9.12 レシーバ制御レジスタ 2 Rx[7:0] (PXS\_RXCR2L)

アドレスオフセット : 0x0D

リセット値 : 0x00

7	6	5	4	3	2	1	0
RXCR2_7	RXCR2_6	RXCR2_5	RXCR2_4	RXCR2_3	RXCR2_2	RXCR2_1	RXCR2_0
rw	rw	rw	rw	rw	rw	rw	rw

## 21.9.13 レシーバ制御レジスタ 3 Rx[9:8] (PXS\_RXCR3H)

アドレスオフセット : 0x0E

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						RXCR3_9	RXCR3_8
						rw	rw

## 21.9.14 レシーバ制御レジスタ 3 Rx[7:0] (PXS\_RXCR3L)

アドレスオフセット : 0x0F

リセット値 : 0x00

7	6	5	4	3	2	1	0
RXCR3_7	RXCR3_6	RXCR3_5	RXCR3_4	RXCR3_3	RXCR3_2	RXCR3_1	RXCR3_0
rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:10 予約済み

ビット 9:0 **RXCR1/2/3\_n** : レシーバチャネル *n* 変換モード

以下のモードは  $RXENn = 1$  および  $CIPF = 1$  のときに有効になります。

011 : レシーバチャネル *n* は、投影型送信モード (Tx) です。

111 : レシーバチャネル *n* は、投影型受信モード (Rx) です。

これらのビットは、 $CIPF = 0$  のときにのみ変更できます。

## 21.9.15 レシーバ・インアクティブ状態レジスタ Rx[9:8] (PXS\_RXINSRH)

アドレスオフセット : 0x12

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						RXINS9	RXINS8
						rw	rw

## 21.9.16 レシーバ・インアクティブ状態レジスタ Rx[7:0] (PXS\_RXINSRL)

アドレスオフセット : 0x13

リセット値 : 0x00

7	6	5	4	3	2	1	0
RXINS7	RXINS6	RXINS5	RXINS4	RXINS3	RXINS2	RXINS1	RXINS0
rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:10 予約済み

ビット 9:0 **RXINSn** : レシーバチャネル *n* インアクティブ状態選択

このビットは、対応するレシーバチャネルが無効である ( $RXENn = 0$ ) のときに有効です。これはまた、所属するグループ (A または B) が選択されていないチャネルにも適用されます (たとえば、 $RXGROUP = 0$  のとき、すべての Rx “B”チャネルに適用されます)。

0 : 無効なとき、レシーバチャネル *n* は VSS に駆動されます。

1 : 無効なとき、レシーバチャネル *n* はハイインピーダンスです。

## 21.9.17 送信イネーブルレジスタ Tx[15:8] (PXS\_TXENRH)

アドレスオフセット : 0x16

リセット値 : 0x00

7	6	5	4	3	2	1	0
TXEN15	TXEN14	TXEN13	TXEN12	TXEN11	TXEN10	TXEN9	TXEN8
rw	rw	rw	rw	rw	rw	rw	rw

## 21.9.18 送信イネーブルレジスタ Tx[7:0] (PXS\_TXENRL)

アドレスオフセット : 0x17

リセット値 : 0x00

7	6	5	4	3	2	1	0
TXEN7	TXEN6	TXEN5	TXEN4	TXEN3	TXEN2	TXEN1	TXEN0
rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **TXEN $m$**  : 送信出力  $m$  機能イネーブル

0 : 送信出力  $m$  用のオルタネート機能は無効であり、対応するピンは、汎用入出力 (GPIO) になるか、他のオルタネート機能によって使用できます。

1 : 送信出力  $m$  用のオルタネート機能は有効です。

**注 :**  $PXSEN = 0$  のとき、すべての送信オルタネート機能は無効です。

**注 :** これらのビットは、 $CIPF = 0$  のときにのみ変更できます。

**注 :** すべてのトランスミッタは、個別に有効にして連続して動作させることができます。

## 21.9.19 最大カウンタ値レジスタ Rx[9:8] (PXS\_MAXRH)

アドレスオフセット : 0x1A

リセット値 : 0xFF

7	6	5	4	3	2	1	0
MAX[15:8]							
rw							

## 21.9.20 最大カウンタ値レジスタ Rx[7:0] (PXS\_MAXRL)

アドレスオフセット : 0x1B

リセット値 : 0xFF

7	6	5	4	3	2	1	0
MAX[7:0]							
rw							

ビット 15:10 予約済み

ビット 7:0 **MAX[15:0]** : 変換データに許容される最大値を示す 16 ビット値。レシーバチャネルの変換データがこの値に到達すると、変換は停止します。最大カウンタに到達するまでに閾値電圧 ( $V_{THR}$ ) に到達しなかったチャネルの場合、変換データ (PXS\_RX $n$ CNTR) は 0x0000 であり、その対応するデータ有効ビット (VALID $n$ ) は "0" です (PXS\_RXSR)。

**注 :** これらのレジスタは、 $CIPF = 0$  のときにのみ変更できます。

## 21.9.21 最大カウンタイネーブルレジスタ Rx[9:8] (PXS\_MAXENRH)

アドレスオフセット : 0x1C

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						MAXEN9	MAXEN8
						r/w	r/w

## 21.9.22 最大カウンタイネーブルレジスタ Rx[7:0] (PXS\_MAXENRL)

アドレスオフセット : 0x1D

リセット値 : 0x00

7	6	5	4	3	2	1	0
MAXEN7	MAXEN6	MAXEN5	MAXEN4	MAXEN3	MAXEN2	MAXEN1	MAXEN0
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

ビット 9:0 **MAXEN $n$**  : 受信チャネル  $n$  に対して有効にされた最大カウンタ限度

0 : RX $n$ CNTR に限度はなく、閾値電圧 ( $V_{THR}$ ) に到達するとカウンタ値が設定され、 $V_{THR}$  に到達しない場合は“0”が設定されます。

1 : RX $n$ CNTR は MAX[15:0] に制限されます。レーバチャネル  $n$  に関して、そのカウンタが MAX に到達するまでに閾値電圧 ( $V_{THR}$ ) に到達しない場合、変換は停止され、RX $n$ CNTR は 0x0000 として読み出されます。

**注 :** これらのビットは、CIPF = 0 のときにのみ変更できます。

## 21.9.23 レシーバステータスレジスタ Rx[9:8] (PXS\_RXSRH)

アドレスオフセット : 0x1E

リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み						VALID9	VALID8
						r	r

## 21.9.24 レシーバステータスレジスタ Rx[7:0] (PXS\_RXSRL)

アドレスオフセット : 0x1F

リセット値 : 0x00

7	6	5	4	3	2	1	0
VALID7	VALID6	VALID5	VALID4	VALID3	VALID2	VALID1	VALID0
r	r	r	r	r	r	r	r

ビット 15:10 予約済み

ビット 9:0 **VALID $n$**  : レシーバチャネル  $n$  の変換データの有効ビットは有効です

0 : RX $n$ CNTR は“0”であり、有効ではありません。

1 : RX $n$ CNTR は有効です。

Cs コンデンサの電圧が閾値電圧 ( $V_{THR}$ ) に到達すると、すぐにそのデータレジスタが更新され、対応する VALID $n$  ビットがセットされます。

PXS\_RXSR は、各変換の開始時 (START が“0”から“1”に遷移) にクリアされます。PXS\_RXSR は、PXSEN = 0 のときにもクリアされます。

VALID $n$  は、チャネルが無効である (RXEN $n$  = 0) か投影型送信モード (RXCR1/2/3\_ $n$  が“011”) であるとき、クリアされたままです。また、対応するチャネルのデータが 65535 (0xFFFF) に到達して MAXEN $n$  がクリアされているか、MAX[15:0] を上回って MAXEN $n$  が“1”である場合にも、クリアされたままです。

## 21.9.25 レシーバチャネル $n$ のビット [15:8] カウンタレジスタ (PXS\_RX $n$ CNTRH)

アドレスオフセット : 0x20 - 0x32 (偶数アドレス)  
リセット値 : 0x00

7	6	5	4	3	2	1	0
RX $n$ CNT[15:8]							
r							

## 21.9.26 レシーバチャネル $n$ のビット [7:0] カウンタレジスタ (PXS\_RX $n$ CNTRL)

アドレスオフセット : 0x21 - 0x33 (奇数アドレス)  
リセット値 : 0x00

7	6	5	4	3	2	1	0
RX $n$ CNT[7:0]							
r							

ビット 15:0 **RX $n$ CNT[15:0]** : レシーバチャネル  $n$  での測定から得られる 16 ビットカウンタ値。

注 : この値は、PXS\_RXSR の VALID = 1 のときに有効であり、それ以外の場合 0x0000 として読み出されず。

注 : この値は、PXSEN = 0 のときにクリアされます。

## 21.9.27 レシーバ 0~9 サンプリングコンデンサ (C<sub>S</sub>) サイズ選択レジスタ 0-9 (PXS\_RX $n$ CSSELR)

アドレスオフセット : 0x40 - 0x49  
リセット値 : 0x00

7	6	5	4	3	2	1	0
予約済み				RX $n$ CSSEL[4:0]			
				rw			

ビット 7:5 予約済み

ビット 4:0 **RX $n$ CSSEL[4:0]** : レシーバチャネル  $n$  の内部サンプリングコンデンサ (C<sub>S</sub>) のサイズ選択。

0x00 - 0x1F : TBD

このレジスタは、CIPF = 0 のときにのみ変更できます。

## 21.9.28 レシーバ 0~9 電極寄生容量補償コンデンサ (EPCC) サイズ選択レジスタ 0-9 (PXS\_RXnEPCCSELR)

アドレスオフセット : 0x50 - 0x59  
リセット値 : 0x00

7	6	5	4	3	2	1	0
RXnEPCC[7:0]							
rw							

ビット 7:0 **RXnEPCC[7:0]** : レシーバチャネル *n* の内部電極寄生容量補償コンデンサ (EPCC) のサイズ選択。

- 0x00 : EPCCは無効です。
- 0x01 - 0xFF : レシーバチャネル *n* の EPCC コンデンサのサイズは次のようになります。  
 $RXnEPCC * 0.0625 \text{ pF}$

**注 :** このレジスタは、CIPF = 0 のときにのみ変更できます。



## 21.9.29 ProxSense レジスタマップとリセット値

表 55. ProxSense レジスタマップ

アドレス	レジスタ名	7	6	5	4	3	2	1	0
0x00	PXS_CR1 リセット値	PXSEN 0	START 0	LOW_POWER 0	- 00000				
0x01	PXS_CR2 リセット値	EOCITEN 0	FCCITEN 0	NOISE- DETEN 0	- 0	RXGROUP 0	RX COUPLING 0	SYNCEN 0	SYNCEGE 0
0x02	PXS_CR3 リセット値	STAB[1:0] 00		BIAS[1:0] 00		VTHR[3:0] 0100			
0x04	PXS_ISR リセット値	EOCF 0	FCCF 0	NOISE- DETF 0	CIPF 0	SYNCPF 0	SYNC OVRF 0	- 00	
0x06	PXS_CKCR1 リセット値	- 0	PRESC[2:0] 011			- 00		ANADEAD 0	INCPHASE 0
0x07	PXS_CKCR2 リセット値	- 0	UPLEN[2:0] 001			- 0	PASSLEN[2:0] 001		
0x08	PXS_RXENRH リセット値	- 000000						RXEN9 0	RXEN8 0
0x09	PXS_RXENRL リセット値	RXEN7 0	RXEN6 0	RXEN5 0	RXEN4 0	RXEN3 0	RXEN2 0	RXEN1 0	RXEN0 0
0x0A	PXS_RXCR1H リセット値	- 000000						RXCR1_9 0	RXCR1_8 0
0x0B	PXS_RXCR1L リセット値	RXCR1_7 0	RXCR1_6 0	RXCR1_5 0	RXCR1_4 0	RXCR1_3 0	RXCR1_2 0	RXCR1_1 0	RXCR1_0 0
0x0C	PXS_RXCR2H リセット値	- 000000						RXCR2_9 0	RXCR2_8 0
0x0D	PXS_RXCR2L リセット値	RXCR2_7 0	RXCR2_6 0	RXCR2_5 0	RXCR2_4 0	RXCR2_3 0	RXCR2_2 0	RXCR2_1 0	RXCR2_0 0
0x0E	PXS_RXCR3H リセット値	- 000000						RXCR3_9 0	RXCR3_8 0
0x0F	PXS_RXCR3L リセット値	RXCR3_7 0	RXCR3_6 0	RXCR3_5 0	RXCR3_4 0	RXCR3_3 0	RXCR3_2 0	RXCR3_1 0	RXCR3_0 0
0x12	PXS_RXINSRH リセット値	- 000000						RXINS9 0	RXINS8 0
0x13	PXS_RXINSRL リセット値	RXINS7 0	RXINS6 0	RXINS5 0	RXINS4 0	RXINS3 0	RXINS2 0	RXINS1 0	RXINS0 0
0x16	PXS_TXENRH リセット値	TXEN15 0	TXEN14 0	TXEN13 0	TXEN12 0	TXEN11 0	TXEN10 0	TXEN9 0	TXEN8 0
0x17	PXS_TXENRL リセット値	TXEN7 0	TXEN6 0	TXEN5 0	TXEN4 0	TXEN3 0	TXEN2 0	TXEN1 0	TXEN0 0
0x1A	PXS_MAXRH リセット値	MAX[15:8] 11111111							
0x1B	PXS_MAXRL リセット値	MAX[7:0] 11111111							
0x1C	PXS_MAXENRH リセット値	- 000000						MAXEN9 0	MAXEN8 0
0x1D	PXS_MAXENRL リセット値	MAXEN7 0	MAXEN6 0	MAXEN5 0	MAXEN4 0	MAXEN3 0	MAXEN2 0	MAXEN1 0	MAXEN0 0
0x1E	PXS_RXSRH リセット値	- 000000						VALID9 0	VALID8 0
0x1F	PXS_RXSRL リセット値	VALID7 0	VALID6 0	VALID5 0	VALID4 0	VALID3 0	VALID2 0	VALID1 0	VALID0 0

表 55. ProxSense レジスタマップ (続き)

アドレス	レジスタ名	7	6	5	4	3	2	1	0
0x20	PXS_RX0CNTRH リセット値	RX0CNT[15:8] 00000000							
0x21	PXS_RX0CNTRL リセット値	RX0CNT[7:0] 00000000							
0x22	PXS_RXnCNTRH リセット値	RXnCNT[15:8] 00000000							
0x23	PXS_RXnCNTRL リセット値	RXnCNT[7:0] 00000000							
...	...	...							
0x32	PXS_RX9CNTRH リセット値	RX9CNT[15:8] 00000000							
0x33	PXS_RX9CNTRL リセット値	RX9CNT[7:0] 00000000							
0x40	PXS_RX0CSSELR リセット値	- 000			RX0CSSEL[4:0] 00000				
0x41	PXS_RXnCSSELR リセット値	- 000			RXnCSSEL[4:0] 00000				
...	...	...							
0x49	PXS_RX9CSSELR リセット値	- 000			RX9CSSEL[4:0] 00000				
0x50	PXS_RX0EPCCSELR リセット値	RX0EPCC[7:0] 00000000							
0x51	PXS_RXnEPCCSELR リセット値	RXnEPCC[7:0] 00000000							
...	...	...							
0x59	PXS_RX9EPCCSELR リセット値	RX9EPCC[7:0] 00000000							

22 改版履歴

表 56. 改版履歴

日付	版	変更内容
2011年10月28日	1	初版リリース。
2012年3月20日	2	STM8T52x および STM8TL53x デバイスに対応するため、タイトルを「STM8TL5xxx マイクロコントローラファミリ」に変更 <a href="#">セクション 4.4 : メモリ構成 (27 ページ)</a> に PCODE の説明を追加

## よくお読み下さい：

本書記載の内容はST 製品に関してのみ適用されるものです。STMicroelectronics NV およびその子会社（以下、ST）は、本書、本製品ならびに本書に記載されたサービスの内容を予告なく変更、修正、改定もしくは改良する権利を留保します。

すべてのST 製品はST の販売条件に従って販売されます。

本書記載のST製品およびサービスの選択並びに使用については購入者が全ての責任を負うものとし、本書記載のST製品およびサービスの選択並びに使用に関してST は一切の責任を負いません。

本書は、明示されているか否かに関わらず、また禁反言によるとやらずに関わらず、いかなる知的財産権の実施権を許諾するためのものではありません。本書で第三者の製品もしくはサービスに言及する場合、そのような言及は、STが当該第三者の製品、サービスまたはそれらに含まれる知的財産権を利用する実施権を許諾したとみなすものではなく、またいかなる用法であれ当該第三者の製品、サービスまたはそれらに含まれる知的財産権の使用を保証するものではありません。

ST の販売条件に規定される場合を除き、STは、商品性、特定目的への適合性(その他管轄の如何を問わず法律で認められる同等のもの)、若しくは特許権、著作権その他の知的財産権の侵害に関する黙示の保証を含め、ST 製品の使用または販売に関する明示または黙示の保証をすべて放棄します。

権限のあるST の代表者二人による書面での明示の許可がある場合を除き、ST 製品を軍사용、航空技術、宇宙用、救命用、若しくは生命維持用に用いること、または不具合若しくは誤動作により負傷または死亡事故、深刻な財産上または環境上の損害を招くおそれのある製品またはシステムへ使用することは、推奨、認可、保証されておりません。「自動車向け」として指定されていないST 製品を自動車用に用いることは、使用者自身の責任においてなされるものとします。

本書に説明されている記述または技術的特徴とは異なる条件でST 製品が再販された場合、本書に記載されるST 製品またはサービスについてのST による保証は直ちに失われるものとし、いかなる形であれ、ST の責任を生ぜしめること、若しくはその責任が延長されることはないものとします。

ST およびST ロゴは各国におけるSTMicroelectronics の商標または登録商標です。

本書の情報はそれ以前に提供された全ての情報に優先します。

ST ロゴはSTMicroelectronics の登録商標です。その他の名称は、それぞれの所有者に帰属します。

© 2013 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)