

**TOSHIBA**

32

RISC  
TX04

TMPM475FDFG/FZFG/FYFG

株式会社 **東芝**

セミコンダクター & ストレージ社





\*\*\*\*\*  
ARM, Cortex および Thumb はARM Limited(またはその子会社)のEUまたはその他の国における  
登録商標です。All rights reserved.  
\*\*\*\*\*



## 製品ご使用上の注意点について

本資料に掲載されている製品について、使用上の注意点を説明します。

なお、本項目と本資料での記述について、異なる場合は、本資料の記述が優先されます。

### 1. 電源投入時の動作について

電源投入時、本資料に掲載されている製品の内部は不定状態となります。

このため、リセットが有効となるまで、端子の状態は不定となります。

外部リセット端子を使用してリセットする製品の場合、電源投入後外部から入力するリセットが有効になるまでの間、端子の状態は不定となります。

また、内蔵パワーオンリセットを使用してリセットする製品の場合、電源電圧が電源投入から内蔵パワーオンリセットが有効となる電圧に上昇するまでの間、端子の状態は不定となります。

### 2. 未使用端子の処置について

本資料に掲載されている製品では、未使用の入出力ポートは、入出力禁止となり、端子はハイインピーダンスです。一般にハイインピーダンスの端子を開放状態で製品を動作させると、外部からのノイズを受け誘起電圧が発生して LSI 内部で静電破壊やラッチアップが発生することがあります。

未使用端子については、1 本ずつ、抵抗を通して電源端子または GND 端子に固定することを推奨します。

### 3. クロック発振の安定について

リセットはクロック発振が安定してから解除してください。プログラム動作中にクロックを切り替える場合、切り替える先のクロック発振が安定している状態で切り替えてください。



b. 各 SFR(レジスタ)の説明

- 各レジスタは、基本的にすべて 32bit のレジスタで構成されています(一部例外有)。
- 各レジスタの説明では、対象ビット、ビットシンボル、タイプ、リセット後の初期値、機能説明が表現されています。

1.2.2 SAMCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	MODE	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MODE		TDATA					
	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	
31-10		R	"0"
9-7	MODE[2:0]	R/W	000 : 0 001 : 1 010 : 2 011 : 3 : Reserved
6-0	TDATA[6:0]	W	

) Type 3

R/W: READ WRITE /  
 R: READ  
 W: WRITE

c. データ表記について

SFR の説明において使用しているシンボルには以下のようなものがあります。

- x:チャンネル番号/ポート
- n,m:ビット番号

d. レジスタの表現

説明文においてレジスタを以下のように表現しています。

- レジスタ名<Bit Symbol>  
 例: SAMCR<MODE>="000"または SAMCR<MODE[2:0]>="000"  
 <MODE[2:0]>はビットシンボル MODE(3 ビット幅)の 2~0 ビット目を意味します。
- レジスタ名[Bit]  
 例: SAMCR[9:7]="000"  
 レジスタ SAMCR(32 ビット幅)の 9~7 ビット目を意味します。

日付	版	改訂理由
2014/12/04	Tentative 1	First Release
2015/04/24	Tentative 2	Contents Revised
2015/05/27	Tentative 3	Contents Revised
2015/09/15	1	First Release
2015/12/24	2.1	Contents Revised
2016/02/18	2.2	Contents Revised
2016/03/10	2.3	Contents Revised
2021/10/15	2.4	Contents Revised
2022/03/31	2.5	Contents Revised
2022/06/01	2.6	Contents Revised
2023/07/14	2.7	Contents Revised
2023/07/31	2.8	Contents Revised



---

---

## 製品ご使用上の注意点について

---

---

### TMPM475FDFG/FZFG/FYFG

---

---

1.1	機能概要	1
1.2	ブロック図	5
1.3	ピン配置図(Top view)	6
1.4	ピン名称と機能	7
1.4.1	機能端子名称と機能	7
1.4.1.1	周辺機能端子名称	
1.4.1.2	デバッグ端子名称	
1.4.1.3	制御端子名称	
1.4.1.4	電源端子名称	
1.4.1.5	電源間コンデンサ	
1.4.2	ピン名称と機能	11
1.4.2.1	表の見方	
1.4.2.2	PORT/デバッグ端子	
1.4.2.3	制御端子	
1.4.2.4	電源端子	

---

---

## 第2章 製品情報

---

---

2.1	各周辺機能の情報	18
2.1.1	DMA コントローラ(DMAC)	18
2.1.2	16ビットタイマ/イベントカウンタ(TMRB)	18
2.1.3	シリアルチャネル(SIO/UART)	19
2.1.4	シリアルバスインタフェース(I2C)	20
2.1.5	CAN コントローラ(CAN)	20
2.1.6	ベクトルエンジン (A-VE)	21
2.1.7	モータ制御回路 (PMD)	21
2.1.8	エンコーダ (A-ENC)	22
2.1.9	アナログ/デジタルコンバータ(ADC)	22
2.1.10	ウォッチドッグタイマ(WDT)	23
2.1.11	デバッグインタフェース	23

---

---

## 第3章 プロセッサコア

---

---

3.1	コアに関する情報	25
3.2	構成可能なオプション	25
3.3	例外/割り込み	26
3.3.1	割り込み本数	26
3.3.2	割り込み優先度ビット数	26
3.3.3	SysTick	26
3.3.4	SYSRESETREQ	26
3.3.5	LOCKUP	26
3.3.6	補助フォールトステータスレジスタ	26
3.4	イベント	27
3.5	電力管理	27

3.6	排他アクセス	27
3.7	浮動小数点演算装置(FPU)	27

## 第4章 メモリマップ

4.1	メモリマップ	29
4.2	バスマトリクス	33
4.2.1	構成	34
4.2.1.1	シングルチップモード	
4.2.1.2	シングルブートモード	
4.2.2	接続表	36
4.2.2.1	Code 領域/ SRAM 領域	
4.2.2.2	Peripheral 領域	
4.2.3	周辺機能ベースアドレス一覧	38

## 第5章 リセット動作

5.1	コールドリセット時	42
5.1.1	POR によるリセット(RESET 端子を使用しない場合)	42
5.1.2	VLTD によるリセット(RESET 端子を使用しない場合)	43
5.1.3	RESET 端子によるリセット(POR によるリセットが有効な場合)	44
5.1.4	RESET 端子によるリセット(VLTD によるリセットが有効な場合)	45
5.2	ウォームリセット時	46
5.2.1	リセット期間	46
5.3	リセット解除後	46

## 第6章 クロック/モード制御

6.1	特長	47
6.2	レジスタ説明	48
6.2.1	レジスタ一覧	48
6.2.2	CGSYSCR(システムコントロールレジスタ)	49
6.2.3	CGOSCCR(発振コントロールレジスタ)	50
6.2.4	CGSTBYCR(スタンバイコントロールレジスタ)	52
6.2.5	CGPLLSEL(PLL セレクトレジスタ)	53
6.2.6	CGCKSTP(パリアフェラル用クロックストップレジスタ)	54
6.3	クロック制御	55
6.3.1	クロックの種類	55
6.3.2	リセット動作による初期値	55
6.3.3	クロック系統図	56
6.3.4	クロック逡倍回路(PLL)	57
6.3.4.1	安定時間	
6.3.4.2	PLL 設定シーケンス	
6.3.5	ウォーミングアップ機能	59
6.3.6	システムクロック	61
6.3.7	プリスケラクロック	62
6.4	動作モードとモード遷移	63
6.4.1	動作モード状態遷移	63
6.5	動作モード	63
6.5.1	NORMAL モード	63
6.6	低消費電力モード	64
6.6.1	IDLE モード	64
6.6.2	STOP モード	65
6.6.3	低消費電力モードの選択	66
6.6.4	各モードにおける動作状態	66
6.6.5	低消費電力モードの解除	67

6.6.6	ウォーミングアップ.....	69
6.6.7	モード遷移によるクロック動作.....	70
6.6.7.1	NORMAL → STOP → NORMAL 動作モード遷移	

## 第7章 内蔵高速発振調整機能

7.1	構成.....	71
7.2	レジスタ説明.....	72
7.2.1	レジスタ一覧.....	72
7.2.2	TRMOSCxPRO (プロテクトレジスタ).....	72
7.2.3	TRMOSCxEN (イネーブルレジスタ).....	73
7.2.4	TRMOSCxINIT (初期トリミング値モニタレジスタ).....	74
7.2.5	TRMOSCxSET (トリミング値設定レジスタ).....	75
7.3	動作説明.....	76
7.3.1	概要.....	76
7.3.2	調整範囲.....	76

## 第8章 例外

8.1	概要.....	77
8.1.1	種類.....	77
8.1.2	処理の流れ.....	78
8.1.2.1	例外要求と検出	
8.1.2.2	例外の処理と割り込み処理ルーチンへの分岐(横取り)	
8.1.2.3	割り込み処理ルーチンの発行	
8.1.2.4	例外からの復帰	
8.2	リセット例外.....	84
8.3	マスク不能割り込み(NMI).....	85
8.4	SysTick.....	85
8.5	割り込み.....	86
8.5.1	要因.....	86
8.5.1.1	経路	
8.5.1.2	割り込み要求の発生	
8.5.1.3	割り込み要因の伝達	
8.5.1.4	外部割り込み端子を使用する際の注意	
8.5.1.5	要因一覧	
8.5.1.6	アクティブレベル	
8.5.2	処理詳細.....	91
8.5.2.1	処理の流れ	
8.5.2.2	準備	
8.5.2.3	検出(クロックジェネレータ)	
8.5.2.4	検出(CPU)	
8.5.2.5	CPUの処理	
8.5.2.6	割り込み処理ルーチンでの処理(要因の取り下げ)	
8.6	例外/割り込み関連レジスタ.....	97
8.6.1	レジスタ一覧.....	97
8.6.2	NVIC レジスタ.....	98
8.6.2.1	SysTick 制御およびステータスレジスタ	
8.6.2.2	SysTick リロード値レジスタ	
8.6.2.3	SysTick 現在値レジスタ	
8.6.2.4	SysTick 較正值レジスタ	
8.6.2.5	割り込み制御用レジスタ	
8.6.2.6	割り込み優先度レジスタ	
8.6.2.7	ベクタテーブルオフセットレジスタ	
8.6.2.8	アプリケーション割り込みおよびリセット制御レジスタ	
8.6.2.9	システムハンドラ優先度レジスタ	
8.6.2.10	システムハンドラ制御および状態レジスタ	
8.6.3	クロックジェネレータレジスタ.....	116
8.6.3.1	CG 割り込みモードコントロールレジスタ	
8.6.3.2	CGICRCG(CG 割り込み要求クリアレジスタ)	
8.6.3.3	CGRSTFLG(リセットフラグレジスタ)	

---

---

## 第9章 デジタルノイズフィルタ回路(DNF)

---

9.1 構成	123
9.2 レジスタ説明	124
9.2.1 レジスタ一覧	124
9.2.1.1 NFCKCR(ノイズフィルタ制御レジスタ)	
9.2.1.2 NFENCR(ノイズフィルタ許可レジスタ)	
9.3 動作説明	127
9.3.1 構成	127
9.3.2 動作	127
9.3.3 使用可能な動作モード	127
9.3.4 STOPモードを使用する場合の注意点	127
9.3.5 最小ノイズ除去時間	127

---

---

## 第10章 $\mu$ DMA コントローラ ( $\mu$ DMAC)

---

10.1 概要	129
10.1.1 機能一覧	129
10.2 ブロック図	130
10.3 レジスタ説明	131
10.3.1 レジスタ一覧	131
10.3.2 DMAxStatus(DMAC Status Register)	132
10.3.3 DMAxCfg(DMAC Configuration Register)	133
10.3.4 DMAxCtrlBasePtr(Channel control data base pointer Register)	134
10.3.5 DMAxAltCtrlBasePtr(Channel alternate control data base pointer Register)	134
10.3.6 DMAxChnlSwRequest(Channel software request Register)	135
10.3.7 DMAxChnlUseburstSet(Channel useburst set Register)	136
10.3.8 DMAxChnlUseburstClr(Channel useburst clear Register)	137
10.3.9 DMAxChnlReqMaskSet(Channel request mask set Register)	138
10.3.10 DMAxChnlReqMaskClr(Channel request mask clear Register)	139
10.3.11 DMAxChnlEnableSet(Channel enable set Register)	140
10.3.12 DMAxChnlEnableClr(Channel enable clear Register)	141
10.3.13 DMAxChnlPriAltSet(Channel primary-alternate set Register)	142
10.3.14 DMAxChnlPriAltClr(Channel primary-alternate clear Register)	143
10.3.15 DMAxChnlPrioritySet(Channel priority set Register)	144
10.3.16 DMAxChnlPriorityClr(Channel priority clear Register)	145
10.3.17 DMAxErrClr(Bus error clear Register)	146
10.3.18 DMAIFFLGx(DMA Flag Register)	147
10.4 動作説明	148
10.4.1 チャンネル制御データメモリマップ	148
10.4.2 チャンネル制御データの構造	149
10.4.2.1 転送データ最終アドレス	
10.4.2.2 転送先の最終アドレス	
10.4.2.3 制御データ設定	
10.4.3 動作モード	151
10.4.3.1 無効	
10.4.3.2 基本モード	
10.4.3.3 自動要求モード	
10.4.3.4 ビンボンモード	
10.4.3.5 メモリスキャッターギャザーモード	
10.4.3.6 周辺スキャッターギャザーモード	
10.5 使用上の注意	158
10.5.1 SIO/UART、TMRB、ADCを使用する場合	158

---

---

## 第11章 入出力ポート

---

11.1 レジスタ説明	159
-------------	-----

11.1.1	レジスタ一覧	159
11.1.2	ポート機能と設定一覧	161
11.1.2.1	PORT A	
11.1.2.2	PORT B	
11.1.2.3	PORT C	
11.1.2.4	PORT D	
11.1.2.5	PORT E	
11.1.2.6	PORT F	
11.1.2.7	PORT G	
11.1.2.8	PORT H	
11.1.2.9	PORT J	
11.1.2.10	PORT K	
11.1.2.11	PORT L	
11.1.2.12	PORT N	
11.1.2.13	PORT P	
<b>11.2</b>	<b>ポート回路図</b>	<b>173</b>
11.2.1	タイプ FT1	173
11.2.2	タイプ FT2	174
11.2.3	タイプ FT3	175
11.2.4	タイプ FT4	176
11.2.5	タイプ FT5	177
11.2.6	タイプ FT6	178
11.2.7	タイプ FT7	179
11.2.8	タイプ FT8	180

---

## 第12章 16ビットタイマ/イベントカウンタ(TMRB)

---

<b>12.1</b>	<b>概要</b>	<b>181</b>
<b>12.2</b>	<b>構成</b>	<b>182</b>
<b>12.3</b>	<b>レジスタ説明</b>	<b>183</b>
12.3.1	レジスタ一覧	183
12.3.2	TBxEN(イネーブルレジスタ)	184
12.3.3	TBxRUN(RUNレジスタ)	185
12.3.4	TBxCR(コントロールレジスタ)	186
12.3.5	TBxMOD(モードレジスタ)	187
12.3.6	TBxFFCR(フリップフロップコントロールレジスタ)	188
12.3.7	TBxST(ステータスレジスタ)	189
12.3.8	TBxIM(割り込みマスクレジスタ)	190
12.3.9	TBxUC(アップカウンタキャプチャレジスタ)	191
12.3.10	TBxRG0(タイマレジスタ 0)	192
12.3.11	TBxRG1(タイマレジスタ 1)	192
12.3.12	TBxCP0(キャプチャレジスタ 0)	193
12.3.13	TBxCP1(キャプチャレジスタ 1)	193
<b>12.4</b>	<b>回路別の動作説明</b>	<b>194</b>
12.4.1	プリスケータ	194
12.4.2	アップカウンタ(UC)	194
12.4.2.1	ソースクロック	
12.4.2.2	動作の開始と停止	
12.4.2.3	カウンタクリアのタイミング	
12.4.2.4	カウンタのオーバフロー	
12.4.3	タイマレジスタ(TBxRG0, TBxRG1)	195
12.4.4	キャプチャ制御	195
12.4.5	キャプチャレジスタ(TBxCP0, TBxCP1)	196
12.4.6	アップカウンタキャプチャレジスタ(TBxUC)	196
12.4.7	コンパレータ(CP0, CP1)	196
12.4.8	タイマフリップフロップ(TBxFF0)	196
12.4.9	キャプチャ割り込み(INTCAPx0, INTCAPx1)	196
<b>12.5</b>	<b>モード別動作説明</b>	<b>197</b>
12.5.1	16ビットインタバルタイマモード	197
12.5.2	16ビットイベントカウンタモード	197
12.5.3	16ビットPPG(プログラマブル矩形波)出力モード	198
12.5.4	外部トリガPPG(プログラマブル矩形波)出力モード	200
<b>12.6</b>	<b>キャプチャ機能を利用した応用例</b>	<b>201</b>
12.6.1	外部トリガパルスからのワンショットパルス出力	201

12.6.2	パルス幅測定.....	203
--------	-------------	-----

---

---

## 第13章 4バイト FIFO 付きシリアルチャネル(SIO/UART)

---

<b>13.1</b>	<b>概要.....</b>	<b>205</b>
<b>13.2</b>	<b>構成.....</b>	<b>206</b>
<b>13.3</b>	<b>レジスタ説明.....</b>	<b>207</b>
13.3.1	レジスタ一覧.....	207
13.3.2	SCxEN (イネーブルレジスタ).....	208
13.3.3	SCxBUF (バッファレジスタ).....	209
13.3.4	SCxCR (コントロールレジスタ).....	210
13.3.5	SCxMOD0 (モードコントロールレジスタ 0).....	212
13.3.6	SCxMOD1 (モードコントロールレジスタ 1).....	213
13.3.7	SCxMOD2 (モードコントロールレジスタ 2).....	214
13.3.8	SCxBRCR (ボーレートジェネレータコントロールレジスタ).....	216
13.3.9	SCxBRADD (ボーレートジェネレータコントロールレジスタ 2).....	217
13.3.10	SCxFCNF (FIFO コンフィグレジスタ).....	218
13.3.11	SCxRFC (受信 FIFO コンフィグレジスタ).....	220
13.3.12	SCxTFC (送信 FIFO コンフィグレジスタ).....	221
13.3.13	SCxRST (受信 FIFO ステータスレジスタ).....	222
13.3.14	SCxTST (送信 FIFO ステータスレジスタ).....	223
<b>13.4</b>	<b>動作モード.....</b>	<b>224</b>
<b>13.5</b>	<b>データフォーマット.....</b>	<b>225</b>
13.5.1	データフォーマット一覧.....	225
13.5.2	パリティ制御.....	226
13.5.2.1	送信.....	
13.5.2.2	受信.....	
13.5.3	STOP ビット長.....	226
<b>13.6</b>	<b>クロック制御.....</b>	<b>227</b>
13.6.1	プリスケアラ.....	227
13.6.2	シリアルクロック生成回路.....	227
13.6.2.1	ボーレートジェネレータ.....	
13.6.2.2	クロック選択回路.....	
<b>13.7</b>	<b>送信/受信バッファと FIFO.....</b>	<b>231</b>
13.7.1	構成.....	231
13.7.2	送信/受信バッファ.....	231
13.7.3	送信バッファの初期化.....	232
13.7.4	FIFO.....	232
<b>13.8</b>	<b>ステータスフラグ.....</b>	<b>233</b>
<b>13.9</b>	<b>エラーフラグ.....</b>	<b>233</b>
13.9.1	OERR フラグ.....	233
13.9.2	PERR フラグ.....	234
13.9.3	FERR フラグ.....	234
<b>13.10</b>	<b>受信.....</b>	<b>235</b>
13.10.1	受信カウンタ.....	235
13.10.2	受信制御部.....	235
13.10.2.1	I/O インタフェースモードの場合.....	
13.10.2.2	UART モードの場合.....	
13.10.3	受信動作.....	235
13.10.3.1	受信バッファの動作.....	
13.10.3.2	受信 FIFO の動作.....	
13.10.3.3	I/O インタフェースモード、クロック出力モードでの受信.....	
13.10.3.4	受信データの読み出し.....	
13.10.3.5	ウエイクアップ機能.....	
13.10.3.6	オーバーランエラー.....	
<b>13.11</b>	<b>送信.....</b>	<b>239</b>
13.11.1	送信カウンタ.....	239
13.11.2	送信制御部.....	239
13.11.2.1	I/O インタフェースモードの場合.....	
13.11.2.2	UART モードの場合.....	
13.11.3	送信動作.....	240
13.11.3.1	送信バッファの動作.....	

13.11.3.2	送信 FIFO の動作	
13.11.3.3	I/O インタフェースモード、クロック出力モードでの送信	
13.11.3.4	I/O インタフェースモード時の最終ビット出力後の SCxTXD 端子の状態	
13.11.3.5	アンダーランエラー	
13.11.3.6	I/O インタフェースモード、クロック入力モードでのデータのホールド時間	
<b>13.12</b>	<b>ハンドシェイク機能</b>	<b>244</b>
<b>13.13</b>	<b>割り込み/エラー発生タイミング</b>	<b>245</b>
13.13.1	受信割り込み	245
13.13.1.1	シングルバッファ/ダブルバッファ構成の場合	
13.13.1.2	FIFO 使用の場合	
13.13.2	送信割り込み	246
13.13.2.1	シングルバッファ/ダブルバッファ構成の場合	
13.13.2.2	FIFO 使用の場合	
13.13.3	エラー発生	247
13.13.3.1	UART モード	
13.13.3.2	I/O インタフェースモード	
<b>13.14</b>	<b>DMA 要求</b>	<b>248</b>
<b>13.15</b>	<b>ソフトウェアリセット</b>	<b>249</b>
<b>13.16</b>	<b>モード別動作説明</b>	<b>250</b>
13.16.1	モード 0 (I/O インタフェースモード)	250
13.16.1.1	送信	
13.16.1.2	受信	
13.16.1.3	送受信(全二重)	
13.16.2	モード 1 (7 ビット UART モード)	261
13.16.3	モード 2 (8 ビット UART モード)	261
13.16.4	モード 3 (9 ビット UART モード)	262
13.16.4.1	ウェイクアップ機能	
13.16.4.2	プロトコル	

---

## 第 14 章 シリアルバスインタフェース(I2C/SIO)

---

<b>14.1</b>	<b>構成</b>	<b>265</b>
<b>14.2</b>	<b>レジスタ説明</b>	<b>266</b>
14.2.1	レジスタ一覧	266
<b>14.3</b>	<b>I2C バスモード</b>	<b>267</b>
14.3.1	I2C バスモード時のコントロールレジスタ	267
14.3.1.1	SB1xCR0(コントロールレジスタ 0)	
14.3.1.2	SB1xCR1(コントロールレジスタ 1)	
14.3.1.3	SB1xCR2(コントロールレジスタ 2)	
14.3.1.4	SB1xSR(ステータスレジスタ)	
14.3.1.5	SB1xBR0(ボーレートレジスタ 0)	
14.3.1.6	SB1xDBR(データバッファレジスタ)	
14.3.1.7	SB1xI2CAR(I2C バスアドレスレジスタ)	
14.3.2	制御	275
14.3.2.1	動作モードの設定	
14.3.2.2	シリアルクロック	
14.3.2.3	アクノリッジメントモードの指定	
14.3.2.4	転送ビット数の選択	
14.3.2.5	スレーブアドレスとアドレス認識モードの設定	
14.3.2.6	マスタ/スレーブの選択	
14.3.2.7	トランスミッタ/レシーバの選択	
14.3.2.8	バスビジーモニタ	
14.3.2.9	割り込みサービス要求と解除	
14.3.2.10	アービトラージョンロスト検出モニタ	
14.3.2.11	スレーブアドレス一致検出モニタ	
14.3.2.12	ゼネラルコール検出モニタ	
14.3.2.13	最終受信ビットモニタ	
14.3.2.14	データバッファレジスタ(SB1xDBR)	
14.3.2.15	ボーレートレジスタ(SB1xBR0)	
14.3.2.16	ソフトウェアリセット	
14.3.3	データ転送手順	281
14.3.3.1	デバイスの初期化	
14.3.3.2	スタートコンディション、スレーブアドレスの発生	
14.3.3.3	1 ワードのデータ転送	
14.3.3.4	ストップコンディションの発生	
14.3.3.5	再スタートの手順	
14.3.4	データフォーマット	289

14.3.5	マルチマスタで使用する際の注意点	289
<b>14.4</b>	<b>SIO モード</b>	<b>290</b>
14.4.1	SIO モード時のコントロールレジスタ	290
14.4.1.1	SBIxCR0(コントロールレジスタ 0)	
14.4.1.2	SBIxCR1(コントロールレジスタ 1)	
14.4.1.3	SBIxDBR(データバッファレジスタ)	
14.4.1.4	SBIxCR2(コントロールレジスタ 2)	
14.4.1.5	SBIxSR(ステータスレジスタ)	
14.4.1.6	SBIxBR0(ボーレートレジスタ 0)	
14.4.2	制御	296
14.4.2.1	シリアルクロック	
14.4.2.2	転送モード	

## 第15章 CAN コントローラ

<b>15.1</b>	<b>概要</b>	<b>305</b>
<b>15.2</b>	<b>ブロック図</b>	<b>306</b>
<b>15.3</b>	<b>CAN インタフェース</b>	<b>306</b>
<b>15.4</b>	<b>レジスタ説明</b>	<b>307</b>
15.4.1	レジスタ一覧	307
15.4.2	メッセージ ID フィールドレジスタ (CANMBxID)	308
15.4.3	タイムスタンプおよびメッセージ制御フィールドレジスタ (CANMBxTSMVCF)	309
15.4.4	データフィールドレジスタ (CANMBxDL/CANMBxDH)	311
15.4.5	CANMC(メールボックスコンフィグレーションレジスタ)	313
15.4.6	CANMD(メールボックスディレクションレジスタ)	314
15.4.7	CANTRS(送信要求セットレジスタ)	315
15.4.8	CANTRR(送信要求リセットレジスタ)	316
15.4.9	CANTA(送信アクノリッジレジスタ)	317
15.4.10	CANAA(アポートアクノリッジレジスタ)	318
15.4.11	CANRMP(受信メッセージペンディングレジスタ)	319
15.4.12	CANRML(受信メッセージロストレジスタ)	320
15.4.13	CANLAM(ローカル受信マスクレジスタ)	321
15.4.14	CANGAM(グローバル受信マスクレジスタ)	322
15.4.15	CANMCR(マスタ制御レジスタ)	323
15.4.16	CANGSR(グローバルステータスレジスタ)	325
15.4.17	CANBCR1(ビットコンフィグレーションレジスタ 1)	327
15.4.18	CANBCR2(ビットコンフィグレーションレジスタ 2)	328
15.4.19	CANGIF(グローバル割り込みフラグレジスタ)	329
15.4.20	CANGIM(グローバル割り込みマスクレジスタ)	330
15.4.21	CANMBTIF(メールボックス送信割り込みフラグレジスタ)	331
15.4.22	CANMBRIF(メールボックス受信割り込みフラグレジスタ)	332
15.4.23	CANMBIM(メールボックス割り込みマスクレジスタ)	333
15.4.24	CANCDR(チェンジデータ要求レジスタ)	334
15.4.25	CANRFP(リモートフレームペンディングレジスタ)	335
15.4.26	CANCEC(エラーカウンタレジスタ)	336
15.4.27	CANTSP(タイムスタンプカウンタプリスケーラレジスタ)	337
15.4.28	CANTSC(タイムスタンプカウンタレジスタ)	338
<b>15.5</b>	<b>回路別の動作説明</b>	<b>339</b>
15.5.1	メールボックス	339
15.5.2	送信制御レジスタ	340
15.5.3	受信制御レジスタ	341
15.5.4	リモートフレーム制御レジスタ	342
15.5.5	受信フィルタリング	343
15.5.6	タイムスタンプ機能	344
15.5.7	割り込み制御	345
<b>15.6</b>	<b>動作モード</b>	<b>347</b>
15.6.1	コンフィグレーションモード	347
15.6.2	スリープモード	349
15.6.3	サスペンドモード	349
15.6.4	テストループバックモード	350
15.6.5	テストエラーモード	350
<b>15.7</b>	<b>動作説明</b>	<b>351</b>

15.7.1	メッセージ受信	351
15.7.2	メッセージ送信	352
15.7.3	リモートフレームの処理	353
<b>15.8</b>	<b>ビットコンフィグレーション</b>	<b>354</b>

## 第16章 12ビットアナログ/デジタルコンバータ

<b>16.1</b>	<b>機能と特徴</b>	<b>357</b>
<b>16.2</b>	<b>ブロック図</b>	<b>358</b>
<b>16.3</b>	<b>レジスタ一覧</b>	<b>359</b>
<b>16.4</b>	<b>レジスタ詳細</b>	<b>361</b>
16.4.1	ADxCLK (変換クロック設定レジスタ)	361
16.4.2	ADxMOD0 (モード設定レジスタ 0)	363
16.4.3	ADxMOD1 (モード設定レジスタ 1)	364
16.4.4	ADxMOD2 (モード設定レジスタ 2)	365
16.4.5	ADxMOD3 (モード設定レジスタ 3)	366
16.4.6	ADxMOD4 (モード設定レジスタ 4)	367
16.4.7	ADxMOD5 (モード設定レジスタ 5)	368
16.4.8	ADxCMPCR0(監視割り込み設定レジスタ 0)	369
16.4.9	ADxCMPCR1(監視割り込み設定レジスタ 1)	370
16.4.10	ADxCMP0(変換結果比較レジスタ 0)	371
16.4.11	ADxCMP1(変換結果比較レジスタ 1)	371
16.4.12	ADxREG0(変換結果格納レジスタ 0)	372
16.4.13	ADxREG1(変換結果格納レジスタ 1)	373
16.4.14	ADxREG2(変換結果格納レジスタ 2)	374
16.4.15	ADxREG3(変換結果格納レジスタ 3)	375
16.4.16	ADxREG4(変換結果格納レジスタ 4)	376
16.4.17	ADxREG5(変換結果格納レジスタ 5)	377
16.4.18	ADxREG6(変換結果格納レジスタ 6)	378
16.4.19	ADxREG7(変換結果格納レジスタ 7)	379
16.4.20	ADxREG8(変換結果格納レジスタ 8)	380
16.4.21	ADxREG9(変換結果格納レジスタ 9)	381
16.4.22	ADxREG10(変換結果格納レジスタ 10)	382
16.4.23	ADxREG11(変換結果格納レジスタ 11)	383
16.4.24	PMD トリガ用プログラムレジスタ	384
16.4.24.1	ADxPSEL0 ~ ADxPSEL11(PMD トリガ用プログラム番号選択レジスタ 0 ~ 11)	
16.4.24.2	ADxPINTS0 ~ 5(PMD トリガ用割り込み選択レジスタ 0 ~ 5)	
16.4.24.3	ADxPSET0 ~ 5(PMD トリガ用プログラム選択レジスタ 0 ~ 5)	
16.4.25	ADxTSET03 / ADxTSET47 / ADxTSET811(タイマトリガ用プログラムレジスタ)	403
16.4.26	ADxSSET03 / ADxSSET47 / ADxSSET811(ソフトウエアトリガ用プログラムレジスタ)	407
16.4.27	ADxASET03 / ADxASET47 / ADxASET811(常時変換用プログラムレジスタ)	411
<b>16.5</b>	<b>動作説明</b>	<b>415</b>
16.5.1	アナログ基準電圧	415
16.5.2	AD 変換開始	415
16.5.3	AD 監視機能	416
<b>16.6</b>	<b>AD 変換タイミングチャート</b>	<b>417</b>
16.6.1	ソフトウエア AD 変換	417
16.6.2	常時 AD 変換	418
16.6.3	トリガによる AD 変換開始	419
<b>16.7</b>	<b>使用方法の例</b>	<b>421</b>
16.7.1	PMD (3 シャント)、AD コンバータ × 1、順次変換方式	421
16.7.2	PMD A(3 シャント)、AD コンバータ × 2、同時変換方式	422
16.7.3	PMD 0(3 シャント)、PMD 1(1 シャント)、AD コンバータ × 2、順次変換方式	423
16.7.4	PMD (1 シャント)、AD コンバータ × 1、順次変換方式	424
<b>16.8</b>	<b>AD コンバータ使用時の注意</b>	<b>425</b>

## 第17章 モータ制御回路(PMD : Programmable Motor Driver)

<b>17.1</b>	<b>PMD 回路構成</b>	<b>429</b>
-------------	-----------------	------------

<b>17.2</b>	<b>PMD レジスタ一覧</b> .....	<b>430</b>
17.2.1	PMDxMDEN(PMD イネーブルレジスタ).....	431
17.2.2	PMDxPORTMD(ポート出力モードレジスタ).....	432
17.2.3	PMDxMODESEL(モード選択レジスタ).....	433
17.2.4	パルス幅変調回路.....	434
17.2.4.1	PMDxMDCR(PMD コントロールレジスタ)	
17.2.4.2	PMDxCNTSTA(PWM カウンタステータスレジスタ)	
17.2.4.3	PMDxMDCNT(PWM カウンタレジスタ)	
17.2.4.4	PMDxMDPRD(PWM 周期レジスタ)	
17.2.4.5	PMDxCMPU(U 相用 PWM コンペアレジスタ)	
17.2.4.6	PMDxCMPV(V 相用 PWM コンペアレジスタ)	
17.2.4.7	PMDxCMPW(W 相用 PWM コンペアレジスタ)	
17.2.5	通電制御回路.....	446
17.2.5.1	PMDxMDPOT(PMD 出力設定レジスタ)	
17.2.5.2	PMDxMDOUT(PMD 通電制御レジスタ)	
17.2.6	保護制御回路.....	451
17.2.6.1	保護制御回路(EMG 入力部)	
17.2.6.2	PMDxEMGREL(EMG 解除レジスタ)	
17.2.6.3	PMDxEMGCR(EMG コントロールレジスタ)	
17.2.6.4	PMDxEMGSTA(EMG ステータスレジスタ)	
17.2.6.5	保護制御回路(OVV 入力部)	
17.2.6.6	PMDxOVVCR(OVV コントロールレジスタ)	
17.2.6.7	PMDxOVVSTA(OVV ステータスレジスタ)	
17.2.7	デッドタイム制御回路.....	460
17.2.7.1	PMDxDTR(デッドタイムレジスタ)	
17.2.8	同期トリガ生成回路.....	463
17.2.8.1	PMDxTRGCMP0(トリガコンペアレジスタ 0)	
17.2.8.2	PMDxTRGCMP1(トリガコンペアレジスタ 1)	
17.2.8.3	PMDxTRGCMP2(トリガコンペアレジスタ 2)	
17.2.8.4	PMDxTRGCMP3(トリガコンペアレジスタ 3)	
17.2.8.5	PMDxTRGCR(トリガコントロールレジスタ)	
17.2.8.6	PMDxTRGSYNCR(トリガ更新タイミング設定レジスタ)	
17.2.8.7	PMDxTRGMD(トリガ出力モード設定レジスタ)	
17.2.8.8	PMDxTRGSEL(トリガ出力選択レジスタ)	

---

## 第 18 章 バクトルエンジン(A-VE)

---

<b>18.1</b>	<b>概要</b> .....	<b>475</b>
18.1.1	特徴.....	475
18.1.2	主な機能.....	476
<b>18.2</b>	<b>構成</b> .....	<b>476</b>
18.2.1	バクトルエンジンとモータ制御回路および A/D 変換器の関連.....	477
<b>18.3</b>	<b>レジスタ説明</b> .....	<b>479</b>
18.3.1	レジスタ一覧.....	479
18.3.2	VE 制御レジスタ.....	482
18.3.2.1	VExEN(バクトルエンジン動作許可/禁止レジスタ)	
18.3.2.2	VExCPURUNTRG(CPU 起動トリガ選択レジスタ)	
18.3.2.3	VExTASKAPP(タスク指定レジスタ)	
18.3.2.4	VExACTSCH(動作スケジュール選択レジスタ)	
18.3.2.5	VExREPTIME(動作スケジュール繰り返し回数指定レジスタ)	
18.3.2.6	VExTRGMODE(起動トリガモード設定レジスタ)	
18.3.2.7	VExERRINTEN(エラー割り込み許可/禁止設定レジスタ)	
18.3.2.8	VExCOMPEND(VE 強制終了レジスタ)	
18.3.2.9	VExERRDET(エラー検出レジスタ)	
18.3.2.10	VExSCHTASKRUN(スケジュール動作状態/実行中タスク番号レジスタ)	
18.3.2.11	VExTMPREG0~5(テンポラリレジスタ)	
18.3.3	専用レジスタ.....	495
18.3.3.1	VExMODE(タスク制御モードレジスタ)	
18.3.3.2	VExFMODE(フロー制御レジスタ)	
18.3.3.3	VExTPWM(PWM 周期レート設定レジスタ)	
18.3.3.4	VExOMEGA(回転速度設定レジスタ)	
18.3.3.5	VExTHETA(モータ位相設定レジスタ)	
18.3.3.6	VExCOS/VExSIN/VExCOSM/VExSINM(SIN/COS レジスタ)	
18.3.3.7	VExIDREF/VExIQREF(d 軸/q 軸基準電流値設定レジスタ)	
18.3.3.8	VExVD/VExVQ(d 軸/q 軸電圧設定レジスタ)	
18.3.3.9	VExCIDKI/VExCIDKP/VExVICQKI/VExVICQKP(PI 制御係数レジスタ)	
18.3.3.10	VExVDIH/VExVDILH/VExVQIH/VExVQILH(PI 制御積分項保持レジスタ)	
18.3.3.11	VExMCTLF(異常/判定結果保持レジスタ)	
18.3.3.12	VExFPWMCHG(PWM 切り替え速度設定レジスタ)	

18.3.3.13	VExMDPRD(PWM 周期設定レジスタ)	
18.3.3.14	VExMINPLS(最小パルス幅差設定レジスタ)	
18.3.3.15	VExSECTOR/VExSECTORM(セクタ情報レジスタ)	
18.3.3.16	VExIAO/VExIBO/VExICO(ゼロ電流レジスタ)	
18.3.3.17	VExIAADC/VExIBADC/VExICADC(電流 ADC 結果レジスタ)	
18.3.3.18	VExVDC/VExVDCL(電源電圧レジスタ)	
18.3.3.19	VExID/VExIQ(d 軸/q 軸電流レジスタ)	
18.3.3.20	VExTADC(ADC 変換時間設定レジスタ)	
18.3.3.21	VExCMPU/VExCMPV/VExCMPW(PWM DUTY レジスタ)	
18.3.3.22	VExOUTCR(6 相出力制御レジスタ)	
18.3.3.23	VExTRGCRC(同期トリガ補正量設定レジスタ)	
18.3.3.24	VExTRGCMP0/VExTRGCMP1(トリガタイミング設定レジスタ)	
18.3.3.25	VExTRGSEL(同期トリガ指定レジスタ)	
18.3.3.26	VExEMGRS(EMG 復帰設定レジスタ)	
18.3.3.27	VExPIOLIM(PI 制御出力制限レジスタ)	
18.3.3.28	VExCIDKG/VExCIQKG(PI 制御 d 軸/q 軸係数レンジ設定レジスタ)	
18.3.3.29	VExVSLIM(電圧スカラー制限レジスタ)	
18.3.3.30	VExVDQ(電圧スカラーレジスタ)	
18.3.3.31	VExDELTA(偏角レジスタ)	
18.3.3.32	VExCPhi/VExCLD/VExCLQ/VExCR/VExCPhiG/VExCLG/VExCRG(モータ定数レジスタ)	
18.3.3.33	VExVDE/VExVQE(非干渉制御 d 軸/q 軸電圧レジスタ)	
18.3.3.34	VExDTC(デッドタイム補償レジスタ)	
18.3.3.35	VExHYS(電流判定ヒステリシスレジスタ)	
18.3.3.36	VExDTCs(デッドタイム補償制御/ステータスレジスタ)	
18.3.3.37	VExPwMMAX/VExPwMMIN(PWM 出力制限レジスタ)	
18.3.3.38	VExTHTCLP(位相クリップレジスタ)	
<b>18.4</b>	<b>動作説明</b>	<b>545</b>
18.4.1	スケジュール管理	545
18.4.1.1	スケジュール制御	
18.4.1.2	起動制御	
18.4.1.3	割り込み制御	
18.4.2	タスク概要	551
18.4.2.1	電流制御(タスク 5)	
18.4.2.2	SIN/COS 演算(タスク 6)	
18.4.2.3	出力電圧変換(座標軸変換/相変換)	
18.4.2.4	出力制御	
18.4.2.5	トリガ生成(タスク 1)	
18.4.2.6	入力処理	
18.4.2.7	入力電流変換(相変換/座標軸変換)	
18.4.2.8	その他タスク	
<b>18.5</b>	<b>VE チャネルと PMD および ADC の組み合わせ</b>	<b>583</b>

---

## 第 19 章 エンコーダ入力回路 (A-ENC)

---

<b>19.1</b>	<b>概要</b>	<b>585</b>
<b>19.2</b>	<b>ブロック図</b>	<b>585</b>
<b>19.3</b>	<b>レジスタ</b>	<b>586</b>
19.3.1	レジスタ一覧	586
19.3.2	ENxTNCr(ENC 制御レジスタ)	587
19.3.3	ENxRELOAD(RELOAD コンペアレジスタ)	590
19.3.4	ENxINT(INT コンペアレジスタ)	591
19.3.5	ENxCNT(カウンタレジスタ)	592
19.3.6	ENxMCMP(MCMP カウンタレジスタ)	593
19.3.7	ENxRATE(位相カウントレートレジスタ)	594
19.3.8	ENxSTS(ステータスレジスタ)	595
19.3.9	ENxINPCR(入力処理制御レジスタ)	596
19.3.10	ENxSMPDLY(サンプルディレイレジスタ)	597
19.3.11	ENxINPMON(入力モニタレジスタ)	598
19.3.12	ENxCLKCR(サンプルクロック制御レジスタ)	599
19.3.13	ENxINTCR(割り込み制御レジスタ)	600
19.3.14	ENxINTF(割り込みフラグレジスタ)	601
<b>19.4</b>	<b>動作説明</b>	<b>602</b>
19.4.1	エンコーダモード	602
19.4.2	センサモード	603
19.4.2.1	イベントカウント	
19.4.2.2	タイマカウント	
19.4.2.3	位相カウント	

19.4.3	タイマモード.....	608
19.4.4	位相カウンタモード.....	609
19.4.4.1	位相差カウンタモード	
<b>19.5</b>	<b>回路別の機能概要.....</b>	<b>612</b>
19.5.1	入力回路.....	612
19.5.1.1	サンプルクロック	
19.5.1.2	サンプリングモード	
19.5.1.3	ノイズキャンセル	
19.5.2	デコーダ.....	615
19.5.2.1	回転エッジ検出と方向信号生成	
19.5.2.2	Z判定回路	
19.5.2.3	スキップ判定と入力異常判定	
19.5.2.4	エッジ検出エラー判定	
19.5.2.5	バッファ更新制御	
19.5.2.6	BEMF 検出制御	
19.5.3	カウンタ.....	618
19.5.3.1	エンコーダモード、センサモード(イベントカウント)	
19.5.3.2	センサモード(タイマカウント)、タイマモード	
19.5.3.3	センサモード(位相カウント)、位相カウンタモード	
19.5.4	割り込み要求制御.....	621
<b>19.6</b>	<b>ブラシレス DC モータ制御例.....</b>	<b>622</b>

---

## 第 20 章 パワーオンリセット回路(POR)

---

<b>20.1</b>	<b>構成.....</b>	<b>623</b>
<b>20.2</b>	<b>機能.....</b>	<b>624</b>

---

## 第 21 章 電圧検出回路(VLTD)

---

<b>21.1</b>	<b>構成.....</b>	<b>625</b>
<b>21.2</b>	<b>レジスタ説明.....</b>	<b>626</b>
21.2.1	レジスタ一覧.....	626
21.2.2	VDCR (電圧検出制御レジスタ).....	626
<b>21.3</b>	<b>動作説明.....</b>	<b>627</b>
21.3.1	制御.....	627
21.3.2	機能.....	627
21.3.2.1	電圧検出動作の許可/禁止	
21.3.2.2	検出電圧レベル選択	

---

## 第 22 章 周波数検知回路(OFD)

---

<b>22.1</b>	<b>構成.....</b>	<b>629</b>
<b>22.2</b>	<b>レジスタ説明.....</b>	<b>630</b>
22.2.1	レジスタ一覧.....	630
22.2.1.1	OFDCR1(周波数検知回路制御レジスタ 1)	
22.2.1.2	OFDCR2(周波数検知回路制御レジスタ 2)	
22.2.1.3	OFDMNPLLOFF (検知周波数下限値設定レジスタ(PLL OFF 時))	
22.2.1.4	OFDMNPLLON (検知周波数下限値設定レジスタ(PLL ON 時))	
22.2.1.5	OFDMXPLOFF (検知周波数上限値設定レジスタ(PLL OFF 時))	
22.2.1.6	OFDMXPLLON (検知周波数上限値設定レジスタ(PLL ON 時))	
<b>22.3</b>	<b>動作説明.....</b>	<b>637</b>
22.3.1	設定.....	637
22.3.2	使用可能な動作モード.....	637
22.3.3	動作.....	638
22.3.4	検知周波数.....	639
22.3.5	動作手順例.....	640

---

---

## 第23章 ウォッチドッグタイマ(WDT)

---

23.1	構成	641
23.2	レジスタ	642
23.2.1	レジスタ一覧	642
23.2.1.1	WDMOD(ウォッチドッグタイマモードレジスタ)	
23.2.1.2	WDCR(ウォッチドッグタイマコントロールレジスタ)	
23.3	動作説明	644
23.3.1	基本動作	644
23.3.2	動作モードと動作状態	644
23.3.3	暴走検出時の動作	644
23.3.3.1	INTWDT 割り込み発生の場合	
23.3.3.2	内部リセット発生の場合	
23.4	ウォッチドッグタイマの制御	645
23.4.1	ディセーブル制御	645
23.4.2	イネーブル制御	645
23.4.3	ウォッチドッグタイマのクリア制御	645
23.4.4	ウォッチドッグタイマ検出時間の設定	645

---

---

## 第24章 フラッシュメモリ(FLASH)

---

24.1	フラッシュメモリの特長	647
24.1.1	メモリ容量と構成	647
24.1.2	機能	648
24.1.3	動作モード	648
24.1.3.1	モードの説明	
24.1.3.2	モードの決定	
24.1.4	メモリマップ	651
24.1.5	プロテクト/セキュリティ機能	653
24.1.5.1	プロテクト機能	
24.1.5.2	セキュリティ機能	
24.1.6	メモリスワップ機能	654
24.1.6.1	概要	
24.1.6.2	動作説明	
24.1.6.3	操作方法	
24.1.7	レジスタ	656
24.1.7.1	レジスタ一覧	
24.1.7.2	FCSECBIT(セキュリティビットレジスタ)	
24.1.7.3	FCPSR0(プロテクトステータスレジスタ 0)	
24.1.7.4	FCSR(ステータスレジスタ)	
24.1.7.5	FCSWPSR(スワップステータスレジスタ)	
24.1.7.6	FCAREASEL(エリア選択レジスタ)	
24.1.7.7	FCCR(コントロールレジスタ)	
24.1.7.8	FCSTSCLR(ステータスクリアレジスタ)	
24.1.7.9	FCWCLKCR(WCLK 設定レジスタ)	
24.1.7.10	FCPROGCR(Program 用カウンタ設定レジスタ)	
24.1.7.11	FCERASECR(Erase 用カウンタ設定レジスタ)	
24.2	フラッシュメモリ詳細	668
24.2.1	機能	668
24.2.2	フラッシュメモリの動作モード	668
24.2.3	コマンド実行方法	668
24.2.4	自動動作の中止	669
24.2.5	自動動作の完了検知	670
24.2.5.1	手順	
24.2.6	コマンド説明	670
24.2.6.1	自動プログラム	
24.2.6.2	自動チップ消去	
24.2.6.3	自動エリア消去	
24.2.6.4	自動ブロック消去	
24.2.6.5	自動ページ消去	
24.2.6.6	自動プロテクトビットプログラム	
24.2.6.7	自動プロテクトビット消去	
24.2.6.8	ID-Read	

24.2.6.9	Read/リセットコマンド(ソフトウェアリセット)	
24.2.6.10	自動メモリスワップ	
24.2.7	コマンドシーケンス	675
24.2.7.1	コマンドシーケンス一覧	
24.2.7.2	バスライトサイクル時のアドレスビット構成	
24.2.7.3	エリアアドレス(AA)、ブロックアドレス(BA)	
24.2.7.4	プロテクトビットの指定(PBA)	
24.2.7.5	ID-Readのコード(IA, ID)	
24.2.7.6	メモリスワップビットの指定(MSA)	
24.2.7.7	コマンドシーケンス例	
24.2.8	フローチャート	682
24.2.8.1	自動プログラム	
24.2.8.2	自動消去	
<b>24.3</b>	<b>シングルブートモードによる書き替え方法</b>	<b>684</b>
24.3.1	モード設定	684
24.3.2	インタフェース仕様	684
24.3.3	メモリの制約について	685
24.3.4	動作コマンド	685
24.3.4.1	RAM 転送	
24.3.4.2	フラッシュメモリチップ消去およびプロテクトビット消去	
24.3.5	コマンドによらず共通の動作	686
24.3.5.1	シリアル動作モード判定	
24.3.5.2	ACK 応答データ	
24.3.5.3	パスワード判定	
24.3.5.4	CHECK SUM の計算方法	
24.3.6	シリアル動作モード判定の通信ルール	691
24.3.7	RAM 転送コマンドの通信ルール	692
24.3.8	フラッシュメモリチップ消去およびプロテクトビット消去の通信ルール	694
24.3.9	内蔵ブートプログラム全体フローチャート	695
24.3.10	内蔵 BOOT ROM の書き替えアルゴリズムを利用した書き替え手順	696
24.3.10.1	Step-1	
24.3.10.2	Step-2	
24.3.10.3	Step-3	
24.3.10.4	Step-4	
24.3.10.5	Step-5	
24.3.10.6	Step-6	
<b>24.4</b>	<b>ユーザブートモードによる書き替え方法</b>	<b>699</b>
24.4.1	(1-A)書き替えルーチンをフラッシュメモリに内蔵する場合の手順例	699
24.4.1.1	Step-1	
24.4.1.2	Step-2	
24.4.1.3	Step-3	
24.4.1.4	Step-4	
24.4.1.5	Step-5	
24.4.1.6	Step-6	
24.4.2	(1-B)書き替えルーチンを外部から転送する手順例	703
24.4.2.1	Step-1	
24.4.2.2	Step-2	
24.4.2.3	Step-3	
24.4.2.4	Step-4	
24.4.2.5	Step-5	
24.4.2.6	Step-6	
<b>24.5</b>	<b>ユーザブートプログラムの書き替え方法</b>	<b>707</b>
24.5.1	フラッシュ書き替えの手順例	707
24.5.1.1	Step-1	
24.5.1.2	Step-2	
24.5.1.3	Step-3	
24.5.1.4	Step-4	
24.5.1.5	Step-5	
24.5.1.6	Step-6	
24.5.1.7	Step-7	
24.5.1.8	Step-8	
24.5.1.9	Step-9	

---

## 第 25 章 デバッグインタフェース

---

<b>25.1</b>	<b>仕様概要</b>	<b>711</b>
<b>25.2</b>	<b>SWJ-DP</b>	<b>711</b>
<b>25.3</b>	<b>ETM</b>	<b>711</b>

25.4	ホールドモード中の周辺機能.....	711
25.5	デバッグツールとの接続.....	712
25.5.1	接続方法.....	712
25.5.2	デバッグインタフェース端子を汎用ポートとして使用する際の注意.....	712

---

## 第 26 章 ポート等価回路図

---

26.1	ポート.....	713
26.2	アナログ端子.....	714
26.3	制御端子.....	714
26.4	クロック端子.....	715

---

## 第 27 章 電気的特性

---

27.1	絶対最大定格.....	717
27.2	DC 電気的特性 (1/2).....	718
27.3	DC 電気的特性 (2/2).....	719
27.4	12 ビット A/D コンバータ変換特性.....	720
27.5	AC 電気的特性.....	721
27.5.1	AC 測定条件.....	721
27.5.2	シリアルチャネル (SIO/UART).....	721
27.5.2.1	I/O インタフェースモード (VDD = 4.5 ~ 5.5V)	
27.5.3	シリアルバスインタフェース (I2C).....	723
27.5.3.1	I2C モード	
27.5.4	16 ビットタイマ/カウンタ (TMRB).....	725
27.5.4.1	イベントカウンタ	
27.5.4.2	キャプチャ	
27.5.5	外部割り込み.....	725
27.5.6	デバッグ通信.....	726
27.5.6.1	AC 測定条件	
27.5.6.2	SWD インタフェース	
27.5.6.3	JTAG インタフェース	
27.5.7	ETM トレース.....	727
27.5.8	Flash 特性.....	728
27.5.9	内蔵発振回路特性.....	728
27.5.10	外部発振子.....	728
27.6	発振回路.....	729
27.6.1	セラミック発振子.....	729

---

## 第 28 章 パッケージ寸法図

---



## CMOS 32

## TMPM475FDFG/FZFG/FYFG

TMPM475FDFG/FZFG/FYFG は、ARM®社 Cortex®-M4F コアを内蔵した 32 ビット RISC マイクロプロセッサです。

機能概要と特長は次のとおりです。

## 1.1

## 1. ARM®社製 Cortex®-M4F コア使用

## a. Thumb®-2 命令で、コード効率の向上を実現

- ・プログラムフロー改善のための新しい 16 ビット命令
- ・性能とコードサイズ向上のための新しい 32 ビット命令
- ・32 ビット/16 ビット混在の命令セットでコード効率を向上

## b. 高性能化と低消費電力化を同時に実現

## 【高性能化】

- ・32 ビット乗算( $32 \times 32 = 32$  ビット)、積和演算( $32 + 32 \times 32 = 32$  ビット)を 1 クロックで実行
- ・SIMD(Single Instruction Multiple Data)演算を 1 クロックで実行
- ・除算を 2~12 クロックで実行

## 【低消費電力化】

- ・低消費電力ライブラリを使用した最適化設計
  - ・プロセッサコアの動作を停止させるスタンバイ機能
- c. リアルタイム制御に向けた高速割り込み応答
- ・実行時間の長い命令は割り込みで中断可能
  - ・スタックへの PUSH をハードウェアで自動的に実行

## 2. 単精度浮動小数点演算(FPU)

- ・IEEE754 標準に準拠
- ・加算/減算/乗算は 1 クロックで実行、積和は 3 クロックで実行
- ・CPU とは別に専用データレジスタによる並列処理が可能

## 3. 内蔵プログラムメモリ/データメモリ

- ・内蔵 RAM :
  - TMPM475FDFG : 34Kbyte
  - TMPM475FZFG : 34Kbyte
  - TMPM475FYFG : 18Kbyte
- ・内蔵 FlashROM :
  - TMPM475FDFG : 512Kbyte
  - TMPM475FZFG : 384Kbyte
  - TMPM475FYFG : 256Kbyte

- 
4.  $\mu$ DMA コントローラ( $\mu$ DMAC) : 32 チャンネル/1 ユニット  
転送対象 : 内蔵メモリ、周辺機能
  5. クロック制御(CG)
    - ・ PLL 内蔵(8, 10, 12 通倍)
    - ・ クロックギア機能 : 高速クロックを 1/1, 1/2, 1/4, 1/8, 1/16 に分周可能
  6. 低消費電力機能  
IDLE, STOP
  7. 入出力ポート(PORT) : 入出力端子 : 79 本
  8. 16 ビットタイマ(TMRB) : 10 チャンネル
    - ・ 16 ビットインタバルタイマモード
    - ・ 16 ビットイベントカウンタモード
    - ・ インพุットキャプチャ機能
    - ・ 16 ビット PPG 出力
    - ・ 外部トリガ PPG 出力
  9. ウォッチドッグタイマ(WDT) : 1 チャンネル  
リセットまたはマスク不能割り込み(NMI)発生
  10. シリアルチャンネル(SIO/UART) : 4 チャンネル
    - ・ UART/クロック同期式モード選択可能
    - ・ 送信 FIFO : 8 ビット幅 4 段、受信 FIFO : 8 ビット幅 4 段
  11. シリアルバスインタフェース(I2C/SIO) : 1 チャンネル  
通信速度 100kbps / 400kbps
  12. CAN コントローラ 2.0 (CAN) : 1 ユニット
    - ・ Version2.0B Active 対応
    - ・ 32 メールボックス
    - ・ 最大転送レート : 1Mbps
  13. パワーオンリセット回路(POR) : 1 ユニット
  14. 電圧検出回路(VLTD) : 1 ユニット
  15. 周波数検知回路(OFD) : 1 ユニット
  16. ベクトルエンジン(A-VE) : 2 チャンネル
    - ・ モータ制御用演算機能
    - ・ 2 モータ対応
  17. モータ制御回路(PMD) : 2 チャンネル
-

- ・ 3相相補 PWM 出力
- ・ AD コンバータを連動させる同期トリガ生成
- ・ 緊急停止保護機能(EMG 端子)

18. エンコーダ入力回路(A-ENC) : 2チャンネル

- ・ 位置検出機能
  - インクリメンタル形エンコーダ対応(AB,ABZ 入力)
  - ホール IC 対応(1~3 相入力)
  - 120°通電の 3 相 BLDC モータの誘起電圧ゼロクロス検出対応
- ・ デジタルノイズフィルタ内蔵
- ・ モータ制御回路(PMD)の PWM 信号に同期した入力サンプリング可能
- ・ タイマカウンタ機能
  - エンコーダ,ホール IC での位置検出による 16 ビットアップダウンカウンタ
  - 任意周期でカウント可能な 16 ビットカウンタ/キャプチャ
  - 32 ビットタイマカウンタ/キャプチャ

19. 12 ビット AD コンバータ(ADC) : 2 ユニット

- ユニット A アナログ入力 : 12 チャンネル
- ユニット B アナログ入力 : 11 チャンネル
- ユニット A 外部アナログ入力端子 : 9 本
- A/B 両ユニットへの外部アナログ入力端子 : 3 本
- ユニット B 外部アナログ入力端子 : 8 本
- ・ トリガスタート機能 : TMRB 割り込み/PMD トリガによるスタート可能
- ・ 常時変換可能
- ・ AD 監視機能 2ch
- ・ 変換時間 1.0  $\mu$ sec (ADC 変換クロック 120 MHz 時)

20. 割り込み機能 : 優先順位を 7 レベルで設定可能

- ・ 内部 80 本 (ウォッチドッグタイマ割り込みを除く)
- ・ 外部 16 本

21. デジタルノイズフィルタ(DNF) : 外部割り込み 16 本

22. エンディアン

リトルエンディアン

23. 内蔵高速発振回路:10MHz

24. 内蔵高速発振調整機能(TRMOSC):1 ユニット

25. 最大動作周波数 : 120 MHz

26. 動作電圧範囲

- ・ DVDD5 = 4.5 V~5.5 V @ fsys = 120 MHz
- 全機能動作

- ・ DVDD5 = 3.9 V~4.5 V @ fsys = 120 MHz

除く、12 ビット AD コンバータと AC 電气的特性, Flash 書き込み

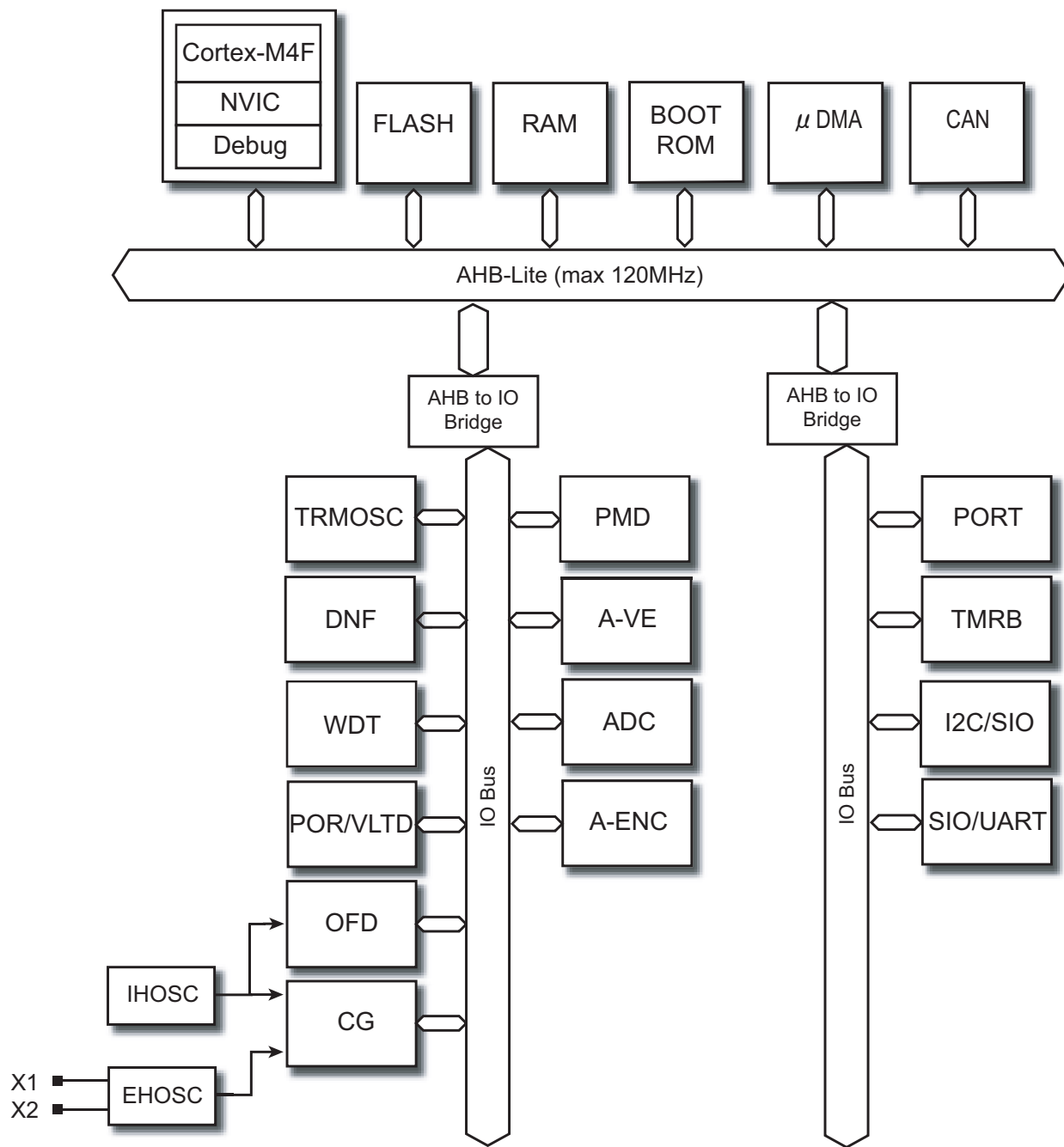
#### 27. 温度範囲

-40°C ~ 85°C

#### 28. パッケージ

LQFP100 (14mm×14mm, 0.5mm ピッチ)

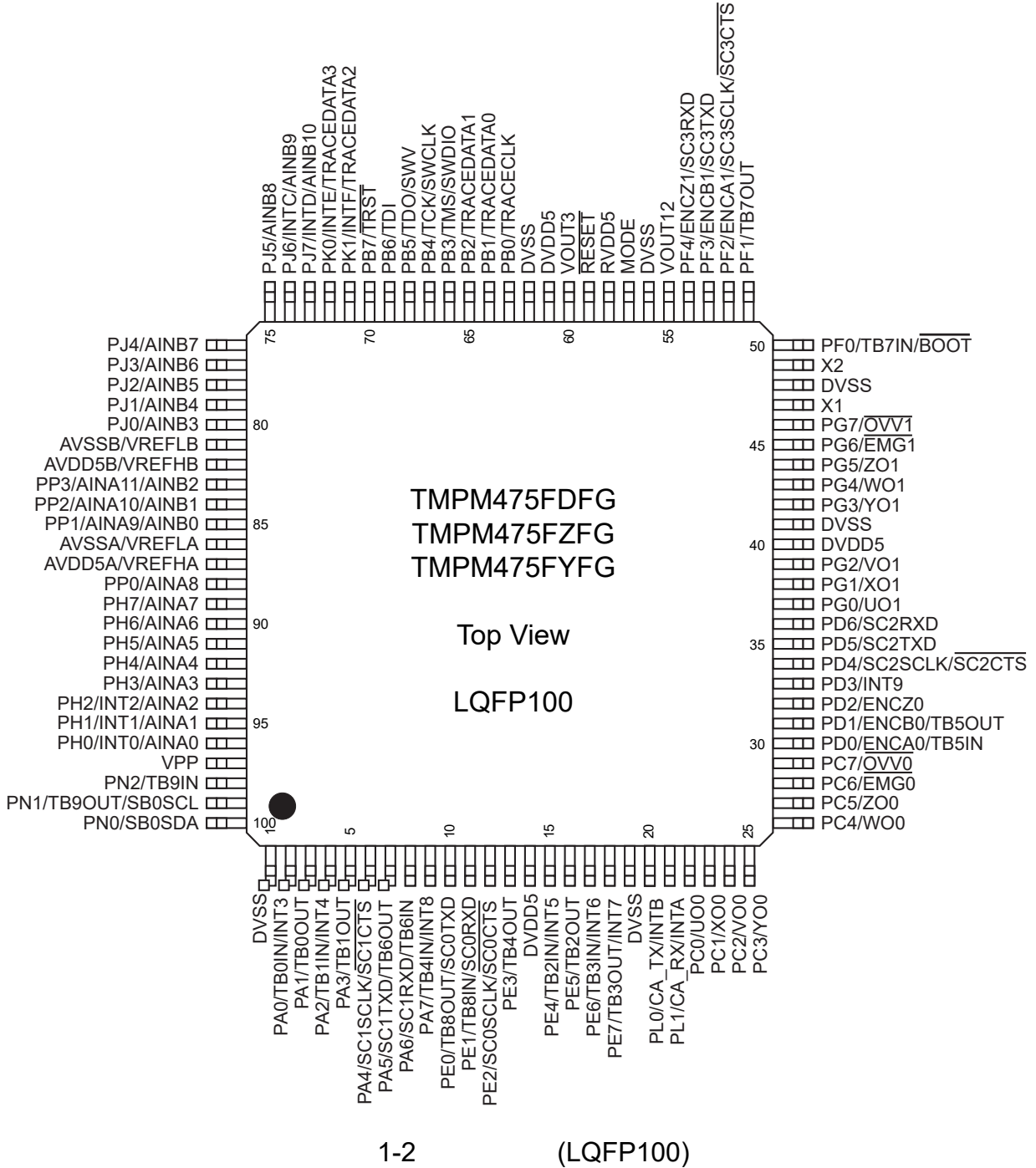
1.2



1-1 TMPM475FDFG/FZFG/FYFG

### 1.3 (Top view)

TMPM475FDFG/FZFG/FYFG のピン配置図は、下図のとおりです。



1.4

1.4.1

1.4.1.1

1-1

		Input or Output	
	INTx	Input	x x (typ. 30ns)
16 /	TBxIN	Input	
	TBxOUT	Output	
SIO/UART	SCxTXD	Output	
	SCxRXD	Input	
	SCxSCLK	I/O	
	SCxCTS	Input	
I2C	SBxSDA	I/O	
	SBxSCL	I/O	
CAN	CA_TX	Output	
	CA_RX	Input	
	AINAx	Input	
	AINBx	Input	
	ENCAx	Input	
	ENCBx	Input	
	ENCZx	Input	
PMD	XOx	Output	X
	YOx	Output	Y
	ZOx	Output	Z
	UOx	Output	U
	VOx	Output	V
	WOx	Output	W
	EMGx	Input	
	OVVx	Input	

## 1.4.1.2

1-2

	Input or Output	
TMS	Input	JTAG
TCK	Input	JTAG
TDO	Output	JTAG
TDI	Input	JTAG
$\overline{\text{TRST}}$	Input	JTAG
SWDIO	I/O	
SWCLK	Input	
SWV	Output	
TRACECLK	Output	
TRACEDATA0	Output	0
TRACEDATA1	Output	1
TRACEDATA2	Output	2
TRACEDATA3	Output	3

## 1.4.1.3

1-3

	Input or Output	
X1	Input	
X2	Output	
MODE	Input	"Low"
$\overline{\text{RESET}}$	Input	
$\overline{\text{BOOT}}$	Input	BOOT BOOT "High" "
		BOOT "Low" "

1.4.1.4

1-4

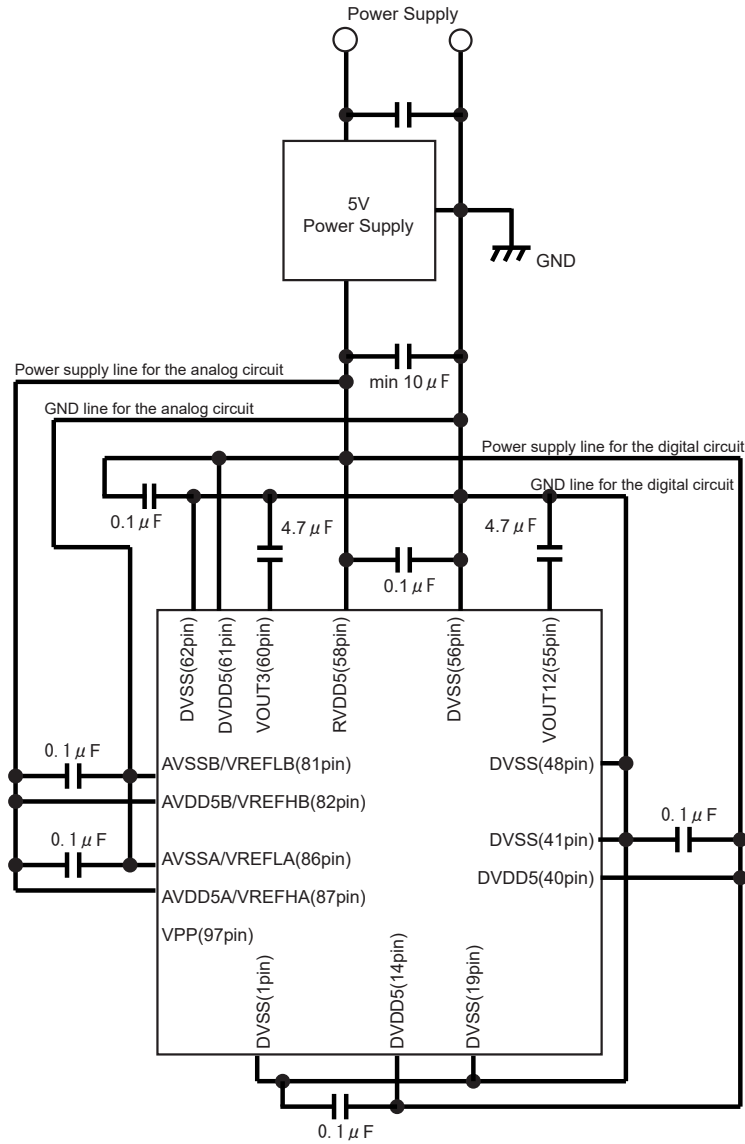
VOUT12	(3.3 to 4.7 $\mu$ F)
VOUT3	(3.3 to 4.7 $\mu$ F)
VPP	( )
RVDD5	
DVDD5	DVDD5 PA, PB, PC, PD, PE, PF, PG, PL, PK, PN, X1, X2, MODE, $\overline{\text{RESET}}$
DVSS	GND
AVDD5A AVDD5B	ADC ( 1) AVDD5A PH, PP0 AVDD5B PJ, PP1 to 3
AVSSA AVSSB	ADC GND ( 2)
VREFHA VREFHB	ADC
VREFLA VREFLB	ADC

1) AD

2) AD

GND

1.4.1.5



1-3

- 1) 58 RVDD5, 56 DVSS (10 μF )
- 2) 58 RVDD5, 56 DVSS (10 μF )  
5V
- 3) GND GND
- 4) GND VOUT3 VOUT12 56 DVSS
- 5) 14 19 40 41 58 56 61 62
- 6) 82 81 87 86

## 1.4.2

### 1.4.2.1

表中の記号の意味は下記の通りです。

#### 1. 機能 A

ファンクションレジスタ設定なしにポートに割り当てられる兼用機能が記載されています。

#### 2. 機能 B

ファンクションレジスタ設定によりポートに割り当てられる兼用機能が記載されています。機能 B の番号はファンクションレジスタの番号と対応しています。

#### 3. 端子仕様

記号の意味は下記のとおりです。

- ・ SMT/CMOS : 入力ゲート
  - SMT : Schmitt 入力
  - CMOS : CMOS 入力
- ・ 5V\_T : 5V トレラント対応
  - Yes : 対応
  - N/A : 非対応
- ・ OD : プログラマブル Open Drain 出力対応
  - Yes : 対応
  - N/A : 非対応
- ・ PU/PD: プログラマブル Pull-Up/Pull-Down 対応
  - PU : プログラマブル Pull-Up 選択可能
  - PD : プログラマブル Pull-Down 選択可能

## 1.4.2.2 PORT /

1-5

&lt;PORT &gt;

Pin No.	PORT	A	B					PU/PD	OD	SMT/ CMOS
			1	2	3	4	5			
PORT A										
2	PA0	INT3	TB0IN					PU/PD	Yes	SMT
3	PA1		TB0OUT					PU/PD	Yes	SMT
4	PA2	INT4	TB1IN					PU/PD	Yes	SMT
5	PA3		TB1OUT					PU/PD	Yes	SMT
6	PA4		SC1SCLK	SC1CTST				PU/PD	Yes	SMT
7	PA5		SC1TXD	TB6OUT				PU/PD	Yes	SMT
8	PA6		SC1RXD	TB6IN				PU/PD	Yes	SMT
9	PA7	INT8	TB4IN					PU/PD	Yes	SMT
PORTB										
63	PB0		TRACECLK					PU/PD	Yes	SMT
64	PB1		TRACEDATA0					PU/PD	Yes	SMT
65	PB2		TRACEDATA1					PU/PD	Yes	SMT
66	PB3		TMS/SWDIO					PU/PD	Yes	SMT
67	PB4		TCKSWCLK					PU/PD	Yes	SMT
68	PB5		TDO/SWV					PU/PD	Yes	SMT
69	PB6		TDI					PU/PD	Yes	SMT
70	PB7		TRST					PU/PD	Yes	SMT
PORTC										
22	PC0		U00					PU/PD	Yes	SMT
23	PC1		X00					PU/PD	Yes	SMT
24	PC2		VO0					PU/PD	Yes	SMT
25	PC3		YO0					PU/PD	Yes	SMT
26	PC4		WO0					PU/PD	Yes	SMT
27	PC5		ZO0					PU/PD	Yes	SMT
28	PC6		EMG0					PU/PD	Yes	SMT
29	PC7		OVV0					PU/PD	Yes	SMT
PORTD										
30	PD0		ENCA0	TB5IN				PU/PD	Yes	SMT
31	PD1		ENCB0	TB5OUT				PU/PD	Yes	SMT
32	PD2		ENCZ0					PU/PD	Yes	SMT
33	PD3	INT9						PU/PD	Yes	SMT
34	PD4		SC2SCLK	SC2CTS				PU/PD	Yes	SMT
35	PD5		SC2TXD					PU/PD	Yes	SMT
36	PD6		SC2RXD					PU/PD	Yes	SMT
PORTE										
10	PE0		SC0TXD	TB8OUT				PU/PD	Yes	SMT
11	PE1		SC0RXD	TB8IN				PU/PD	Yes	SMT
12	PE2		SC0SCLK	SC0CTS				PU/PD	Yes	SMT
13	PE3		TB4OUT					PU/PD	Yes	SMT
15	PE4	INT5	TB2IN					PU/PD	Yes	SMT
16	PE5		TB2OUT					PU/PD	Yes	SMT
17	PE6	INT6	TB3IN					PU/PD	Yes	SMT

1-5

&lt;PORT &gt;

Pin No.	PORT	A	B					PU/PD	OD	SMT/ CMOS
			1	2	3	4	5			
18	PE7	INT7	TB3OUT					PU/PD	Yes	SMT
PORTF										
50	PF0	$\overline{\text{BOOT}}$	TB7IN					PU/PD	Yes	SMT
51	PF1		TB7OUT					PU/PD	Yes	SMT
52	PF2		ENCA1	SC3SCLK	$\overline{\text{SC3CTS}}$			PU/PD	Yes	SMT
53	PF3		ENCB1	SC3TXD				PU/PD	Yes	SMT
54	PF4		ENCZ1	SC3RXD				PU/PD	Yes	SMT
PORTG										
37	PG0		UO1					PU/PD	Yes	SMT
38	PG1		XO1					PU/PD	Yes	SMT
39	PG2		VO1					PU/PD	Yes	SMT
42	PG3		YO1					PU/PD	Yes	SMT
43	PG4		WO1					PU/PD	Yes	SMT
44	PG5		ZO1					PU/PD	Yes	SMT
45	PG6		$\overline{\text{EMG1}}$					PU/PD	Yes	SMT
46	PG7		$\overline{\text{OVV1}}$					PU/PD	Yes	SMT
PORTH										
96	PH0	AINA0 INT0						PU/PD	Yes	SMT
95	PH1	AINA1 INT1						PU/PD	Yes	SMT
94	PH2	AINA2 INT2						PU/PD	Yes	SMT
93	PH3	AINA3						PU/PD	Yes	SMT
92	PH4	AINA4						PU/PD	Yes	SMT
91	PH5	AINA5						PU/PD	Yes	SMT
90	PH6	AINA6						PU/PD	Yes	SMT
89	PH7	AINA7						PU/PD	Yes	SMT
PORTJ										
80	PJ0	AINB3						PU/PD	Yes	SMT
79	PJ1	AINB4						PU/PD	Yes	SMT
78	PJ2	AINB5						PU/PD	Yes	SMT
77	PJ3	AINB6						PU/PD	Yes	SMT
76	PJ4	AINB7						PU/PD	Yes	SMT
75	PJ5	AINB8						PU/PD	Yes	SMT
74	PJ6	AINB9 INTC						PU/PD	Yes	SMT
73	PJ7	AINB10 INTD						PU/PD	Yes	SMT
PORTK										
72	PK0	INTE	TRACEDATA3					PU/PD	Yes	SMT
71	PK1	INTF	TRACEDATA2					PU/PD	Yes	SMT
PORTL										
20	PL0	INTB	CA_TX					PU/PD	Yes	SMT
21	PL1	INTA	CA_RX					PU/PD	Yes	SMT
PORTN										
100	PN0			SB0SDA				PU/PD	Yes	SMT

1-5

&lt;PORT &gt;

Pin No.	PORT	A	B					PU/PD	OD	SMT/ CMOS
			1	2	3	4	5			
99	PN1		TB9OUT	SB0SCL				PU/PD	Yes	SMT
98	PN2		TB9IN					PU/PD	Yes	SMT
PORTP										
88	PP0	AINA8						PU/PD	Yes	SMT
85	PP1	AINA9 AINB0						PU/PD	Yes	SMT
84	PP2	AINA10 AINB1						PU/PD	Yes	SMT
83	PP3	AINA11 AINB2						PU/PD	Yes	SMT

1.4.2.3

1-6

Pin No.	
47	X1
49	X2
57	MODE
59	RESET
50	BOOT

1.4.2.4

1-7

Pin No.	
55	VOUT12
60	VOUT3
58	RVDD5
97	VPP
14, 40, 61	DVDD5
1, 19, 41, 48, 56, 62	DVSS
87	AVDD5A VREFHA
82	AVDD5B VREFHB
86	VREFLA AVSSA
81	AVREFLB AVSSB



## 2

本章では、周辺機能に関し、チャンネルまたはユニット数、端子情報、その他の製品固有機能の情報についてまとめます。周辺機能の章と合わせてご使用ください。

- 「2.1.1 DMA コントローラ(DMAC)」
- 「2.1.2 16 ビットタイマ/イベントカウンタ(TMRB)」
- 「2.1.3 シリアルチャンネル(SIO/UART)」
- 「2.1.4 シリアルバスインタフェース(I2C)」
- 「2.1.5 CAN コントローラ(CAN)」
- 「2.1.6 ベクトルエンジン (A-VE)」
- 「2.1.7 モータ制御回路 (PMD)」
- 「2.1.8 エンコーダ (A-ENC)」
- 「2.1.9 アナログ/デジタルコンバータ(ADC)」
- 「2.1.10 ウォッチドッグタイマ(WDT)」
- 「2.1.11 デバッグインタフェース」

## 2.1

### 2.1.1 DMA (DMAC)

TMPM475FDFG/FZFG/FYFG では DMA コントローラを 1 ユニット内蔵しています。

#### 2-1 DMA

0	SIO/UART0	
1	SIO/UART0	
2	SIO/UART1	
3	SIO/UART1	
4	SIO/UART2	
5	SIO/UART2	
6	SIO/UART3	
7	SIO/UART3	
8	ADCA	
9	ADCB	
10	ADCA	
11	ADCB	
12	PMD0 ADCA	
13	PMD0 ADCB	
14	PMD1 ADCA	
15	PMD1 ADCB	
16	PMD0 PWM	
17	PMD1 PWM	
18		
19		
20	TMRB0	0
21	TMRB1	0
22	TMRB2	0
23	TMRB3	0
24	VE	0
25	VE	1
26	TMRB0	( )
27	TMRB1	( )
28	TMRB2	( )
29	TMRB3	( )
30	VE	0
31	VE	1

) TMRB DMA TMRB TMRB  
0/1  
TBxIM

### 2.1.2 16 / (TMRB)

TMPM475FDFG/FZFG/FYFG では 10 チャンネルの TMRB を内蔵しています。

## 2-2

	TBxOUT	TBxIN
TMRB0	PA1	PA0
TMRB1	PA3	PA2
TMRB2	PE5	PE4
TMRB3	PE7	PE6
TMRB4	PE3	PA7
TMRB5	PD1	PD0
TMRB6	PA5	PA6
TMRB7	PF1	PF0
TMRB8	PE0	PE1
TMRB9	PN1	PN2

## 2-3

		TMRB
TMRB0	INTCAP00, INTCAP01	INTTB00, INTTB01
TMRB1	INTCAP10, INTCAP11	INTTB10, INTTB11
TMRB2	INTCAP20, INTCAP21	INTTB20, INTTB21
TMRB3	INTCAP30, INTCAP31	INTTB30, INTTB31
TMRB4	INTCAP40, INTCAP41	INTTB40, INTTB41
TMRB5	INTCAP50, INTCAP51	INTTB50, INTTB51
TMRB6	INTCAP60, INTCAP61	INTTB60, INTTB61
TMRB7	INTCAP70, INTCAP71	INTTB70, INTTB71
TMRB8	INTCAP80, INTCAP81	INTTB80, INTTB81
TMRB9	INTCAP90, INTCAP91	INTTB90, INTTB91

## 2-4

TMRB0	TMRB1, TMRB2, TMRB3
TMRB4	TMRB5, TMRB6
TMRB7	TMRB8, TMRB9

## 2-5

TMRB0	ENC0
TMRB1	ENC1

### 2.1.3 (SIO/UART)

TPM475FDFG/FZFG/FYFG では 4 チャンネルの SIO/UART を内蔵しています。

## 2-6

	SCxTXD	SCxRXD	SCxSCLK	$\overline{\text{SCxCTS}}$
SC0	PE0	PE1	PE2	PE2
SC1	PA5	PA6	PA4	PA4
SC2	PD5	PD6	PD4	PD4
SC3	PF3	PF4	PF2	PF2

## 2-7

SC0	INTRX0	INTTX0
SC1	INTRX1	INTTX1
SC2	INTRX2	INTTX2
SC3	INTRX3	INTTX3

## 2-8

SC0	TMRB4
SC1	TMRB4
SC2	TMRB7
SC3	TMRB7

## 2.1.4 (I2C)

TMPM475FDFG/FZFG/FYFG では 1 チャンネルの I2C を内蔵し、SIO をサポートしていません。

## 2-9

	SBxSDA	SBxSCL
I2C0	PN0	PN1

## 2-10

I2C0	INTSBI0

## 2.1.5 CAN (CAN)

TMPM475FDFG/FZFG/FYFG では 1 ユニットの CAN を内蔵しています。

2-11

	CA_TX	CA_RX
CAN	PL0	PL1

2-12

CAN	INTCANRX	INTCANTX	INTCANGB

### 2.1.6 (A-VE)

TMPM475FDFG/FZFG/FYFG では2チャンネルの A-VE を内蔵しています。

2-13

VE0	INTVCN0
VE1	INTVCN1

2-14

	PWM	ADC	ADC
VE0	INTPMD0	INTADAPDA INTADBPDA	ADAREG0 to 3
VE1	INTPMD1	INTADAPDB INTADBPDB	ADBREG0 to 3

### 2.1.7 (PMD)

TMPM475FDFG/FZFG/FYFG では2チャンネルの PMD を内蔵しています。

2-15

	OVV <sub>x</sub>	EMG <sub>x</sub>	ZO <sub>x</sub> Z	WO <sub>x</sub> W	YO <sub>x</sub> Y	VO <sub>x</sub> V	XO <sub>x</sub> X	UO <sub>x</sub> U
PMD0	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PMD1	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0

2-16

	OVV	EMG	PWM
PMD0	INTOVV0	INTEMG0	INTPMD0
PMD1	INTOVV1	INTEMG1	INTPMD1

2-17 (1/2)

	EMG (VE)	PWM (VE)	(VE)	(VE)	(VE)	OVV (ADC)	
PMD0	VEEMGRS0	VECMPU0, V0, W0	VEOUTCR0	VETRGCMP00, 01	VETRGSEL0	ADCA	0, 1
PMD1	VEEMGRS1	VECMPU1, V1, W1	VEOUTCR1	VETRGCMP10, 11	VETRGSEL1	ADCB	0, 1

2-18 (2/2)

	MDOU
PMD0	INTENC0, INTTB00, CTRGO(ENC0)
PMD1	INTENC1, INTTB10, CTRGO(ENC1)

2.1.8 (A-ENC)

TMPM475FDFG/FZFG/FYFG では 2 チャンネルの A-ENC を内蔵しています。

2-19

	ENCAx	ENCBx	ENCZx
ENC0	PD0	PD1	PD2
ENC1	PF2	PF3	PF4

2-20

ENC0	INTENC0
ENC1	INTENC1

2-21

		PWM
ENC0	TB0OUT	PWMON0
ENC1	TB1OUT	PWMON1

2.1.9 / (ADC)

TMPM475FDFG/FZFG/FYFG では 2 ユニットの 12 ビット逐次変換方式アナログ/デジタルコンバータ(ADC)を内蔵しています。モータ制御用ベクトルエンジン、および PMD 回路と連携してモータのベクトル制御を支援します。

2-22

	AINA0 to 7	AINA8	AINA9 to 11 AINB0 to 2	AINB3 to 10
ADCA	PH0 to 7	PP0	PP1 to 3	-
ADCB	-	-	PP1 to 3	PJ0 to 7

## 2-23

	PMD			
ADCA	INTADAPDB	INTADATMR	INTADASFT	INTADACPA INTADACPB
ADCB	INTADBPDB	INTADBTMR	INTADBSFT	INTADBCPA INTADBCPB

## 2-24

	PMD	TMRTRG
ADCA	PMD0TRG0 to 5	INTTB50
ADCB	PMD1TRG0 to 5	INTTB60

## 2.1.10 (WDT)

TMPM475FDFG/FZFG/FYFG では WDMOD レジスタの I2WDT ビットをサポートしていませんので、必ず、"0" をライトしてください。

## 2.1.11

TMPM475FDFG/FZFG/FYFG はシリアルワイヤデバッグポート、JTAG デバッグポートおよびトレース出力をサポートしています。

## 2-25

	TMS SWDIO	TCK SWCLK	TDO SWV	TDI	$\overline{\text{TRST}}$
JTAG	PB3	PB4	PB5	PB6	PB7

	TRACECLK	TRACEDATA0	TRACEDATA1	TRACEDATA2	TRACEDATA3
	PB0	PB1	PB2	PK1	PK0



## 3

TMPM475 シリーズには、高性能 32 ビットプロセッサコア (ARM 社 Cortex-M4F コア) が内蔵されています。プロセッサコアの動作については、ARM 社からリリースされるドキュメンテーションセットを参照してください。ここでは、製品固有の情報について説明します。

## 3.1

TMPM475FDFG/FZFG/FYFG で使用している Cortex-M4F コアのリビジョンは以下のとおりです。

CPU コア部、アーキテクチャなどの詳細は、下記 URL より ARM 社の "Cortex-M4 プロセッサ用ドキュメンテーションセット" を参照してください。

<http://infocenter.arm.com/help/index.jsp>

TMPM475FDFG/FZFG/ FYFG	r0p1

## 3.2

Cortex-M4F コアは、一部のブロックについて実装するかどうかを選択することができます。

TMPM475FDFG/FZFG/FYFG での構成は以下のとおりです。

Configurable options	Implementation
MPU (Memory Protection Unit)	Absent
FPB (Flash Patch and Breakpoint)	Two literal comparators Six instruction comparators
DWT (Data Watchpoint and Trace)	Four comparators
ITM (Instrumentation Trace Macrocell)	Present
ETM (Embedded Trace Macrocell)	Present
AHB-AP (AHB Access Port)	Present
HTM Interface (AHB Trace Macrocell Interface)	Absent
TPIU (Trace Port Interface Unit)	Present
WIC (Wake-up Interrupt Controller)	Absent
Debug Port (Serial-Wire or JTAG Debug Port)	Present
FPU (Floating Point Unit)	Present
Bit banding	Present
Constant AHB control	Disable

## 3.3 /

例外/割り込みに関連する製品固有の情報をまとめます。

### 3.3.1

Cortex-M4F コアは割り込み本数を 1 ~ 240 本の間で任意に構成することができます。

TMPM475FDFG/FZFG/FYFG の割り込み本数は 96 本です。割り込み本数は NVIC レジスタの割り込みコントローラタイプレジスタの<INTLINESNUM[3:0]>ビットに反映され、本製品では"0x02"が読み出されます。

### 3.3.2

Cortex-M4F コアは割り込み優先度ビット数を 3 ~ 8 ビットの間で任意に構成することができます。

TMPM475FDFG/FZFG/FYFG の割り込み優先度は 3 ビットです。このビット数は割り込み優先度レジスタとシステムハンドラ優先度レジスタのビット構成に反映されます。

### 3.3.3 SysTick

Cortex-M4F コアには SysTick と呼ばれるシステムタイマがあり、SysTick 例外を発生させることができます。

SysTick 例外の詳細については、例外の「SysTick」の章および、「NVIC レジスタ」の SysTick 関連レジスタの章を参照してください。

### 3.3.4 SYSRESETREQ

Cortex-M4F コアは、アプリケーション割り込みおよびリセット制御レジスタの<SYSRESETREQ>ビットがセットされると SYSRESETREQ 信号を出力します。

TMPM475FDFG/FZFG/FYFG では SYSRESETREQ 信号が出力されるとウォームリセットと同様の動作になります。

### 3.3.5 LOCKUP

回復不能な例外が発生すると Cortex-M4F コアは LOCKUP 信号を出力し、ソフトウェアに重大な誤りのあることを示します。

TMPM475FDFG/FZFG/FYFG ではこの信号は未使用です。LOCKUP 状態から復帰する際にはマスク不能割り込み(NMI)またはリセットを使用する必要があります。

### 3.3.6

Cortex-M4F コアにはソフトウェアに対して追加のシステムフォールト情報を提供するための補助フォールトステータスレジスタが準備されています。

TMPM475FDFG/FZFG/FYFG ではこのレジスタに対して機能を定義していません。リードすると常に"0x0000\_0000"が読み出されます。

### 3.4

Cortex-M4F コアにはイベント出力信号とイベント入力信号があります。イベント出力信号は、SEV 命令実行により出力されます。また、イベントが入力されると WFE 命令による低電力状態から復帰します。

TMPM475FDFG/FZFG/FYFG では、イベント出力信号、イベント入力信号とも未使用です。SEV 命令、WFE 命令は使用しないでください。

### 3.5

Cortex-M4F コアには電力管理のための信号として SLEEPING および SLEEPDEEP があります。SLEEPDEEP は、システム制御レジスタの<SLEEPDEEP>ビットがセットされている場合に出力されます。

これらの信号は、割り込み待ち(WFI)命令の実行、イベント待ち(WFE)命令の実行または、システム制御レジスタの<SLEEPONEXIT>ビットがセットされている場合の割り込みサービスルーチン(ISR)からの退出時に出力されます。

TMPM475FDFG/FZFG/FYFG では、SLEEPDEEP 信号は使用していません。<SLEEPDEEP>ビットはセットしないでください。また、イベント信号も未使用のため、WFE 命令は使用しないでください。

電力管理については、「クロック/モード制御」の章を参照してください。

### 3.6

Cortex-M4F コアの DCode バスおよびシステムバスは排他アクセスをサポートしていますが、TMPM475FDFG/FZFG/FYFG ではこの機能を使用していません。

### 3.7 (FPU)

本製品は、ARMv7M 浮動小数点拡張(FPv4-SP)の派生仕様の演算装置(FPU)を実装しており、IEEE 標準(ANSI/IEEE Std 754-2008)に準拠した単精度浮動小数点演算が可能です。

この FPU は、アドレスバスとデータバスを Cortex-M4F コアと共有し、協調して動作します。加算/減算/乗算を 1 クロックで、積和を 3 クロックで実行します。また、CPU とは別に専用データレジスタによる並列処理が可能です。

なお、この FPU は、ARM アーキテクチャリファレンスマニュアルに掲載された、全ての単精度データ演算命令とデータタイプをサポートします。



## 4

### 4.1

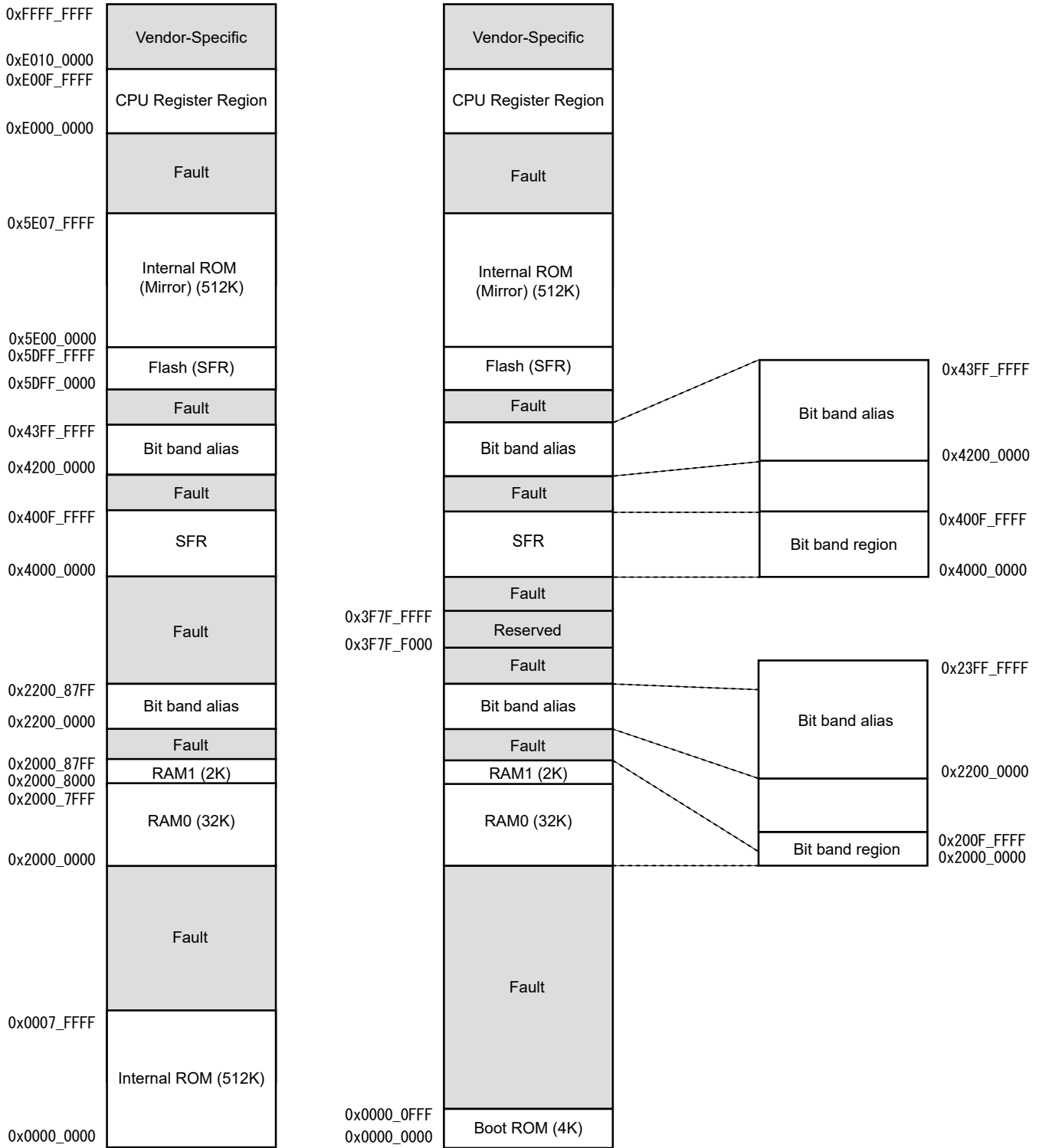
TMPM475FDFG/FZFG/FYFG のメモリマップは、ARM Cortex-M4F コアのメモリマップに沿って作られており、内蔵 ROM 領域は Cortex-M4F コアメモリマップの Code 領域、内蔵 RAM 領域は SRAM 領域、特殊機能レジスタ(SFR)領域は Peripheral 領域に割り付けられています。特殊機能レジスタ(SFR : Special function register)とは、入出力ポートおよび周辺機能のコントロールレジスタを示します。

CPU 内レジスタ領域はコア内部のレジスタ領域です。

各領域の詳細については、"ARM ドキュメンテーションセット Cortex-M4F 編"を参照してください。

"Fault"と記載された領域では、アクセスするとメモリフォールトが有効な場合にはメモリフォールト、無効な場合にはハードフォールトが発生します。また、ベンダ固有領域にはアクセスしないでください。

TMPM475FDFG/FZFG/FYFG のメモリマップを以下に示します。

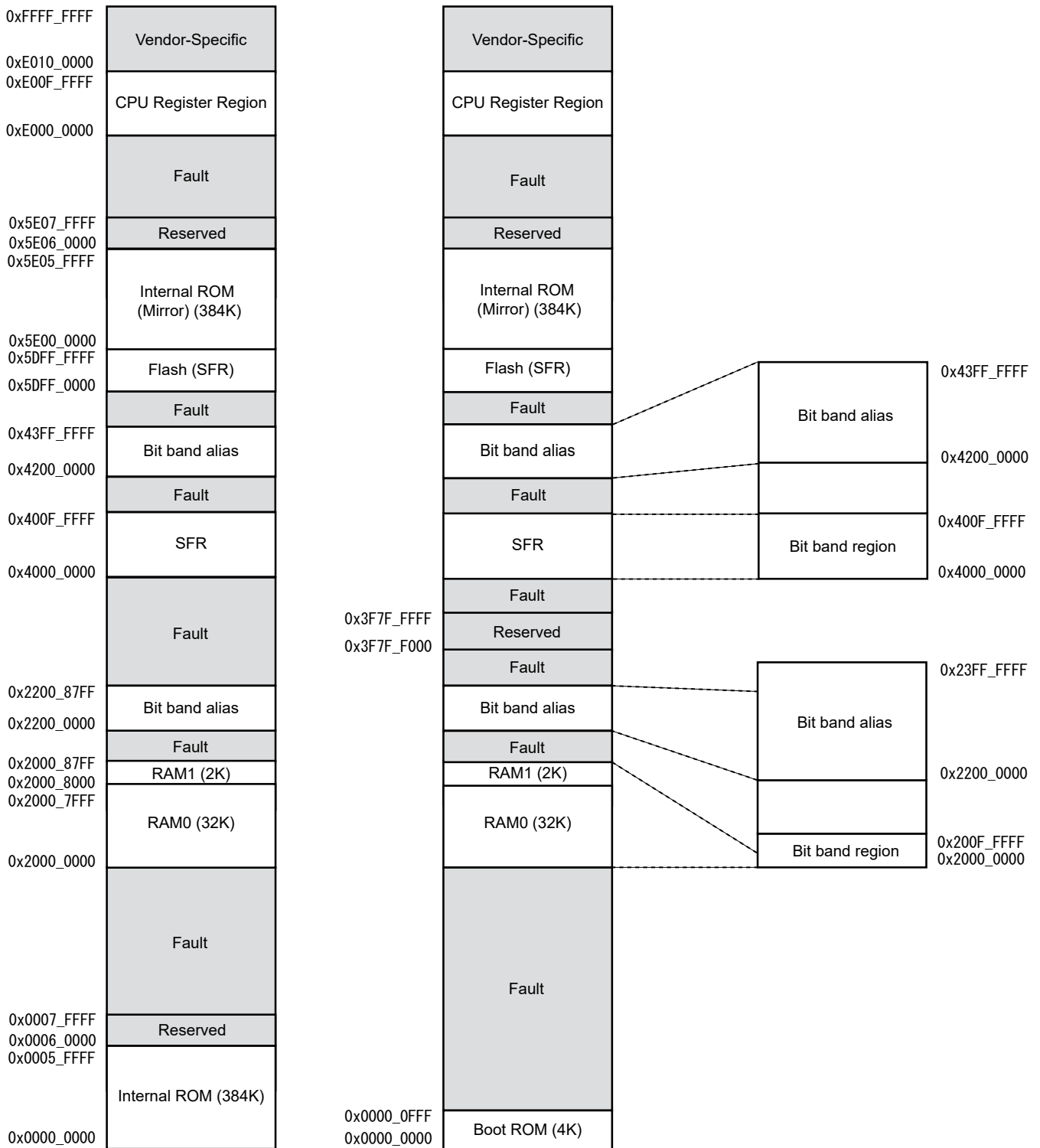


Single chip mode

Single boot mode

4-1

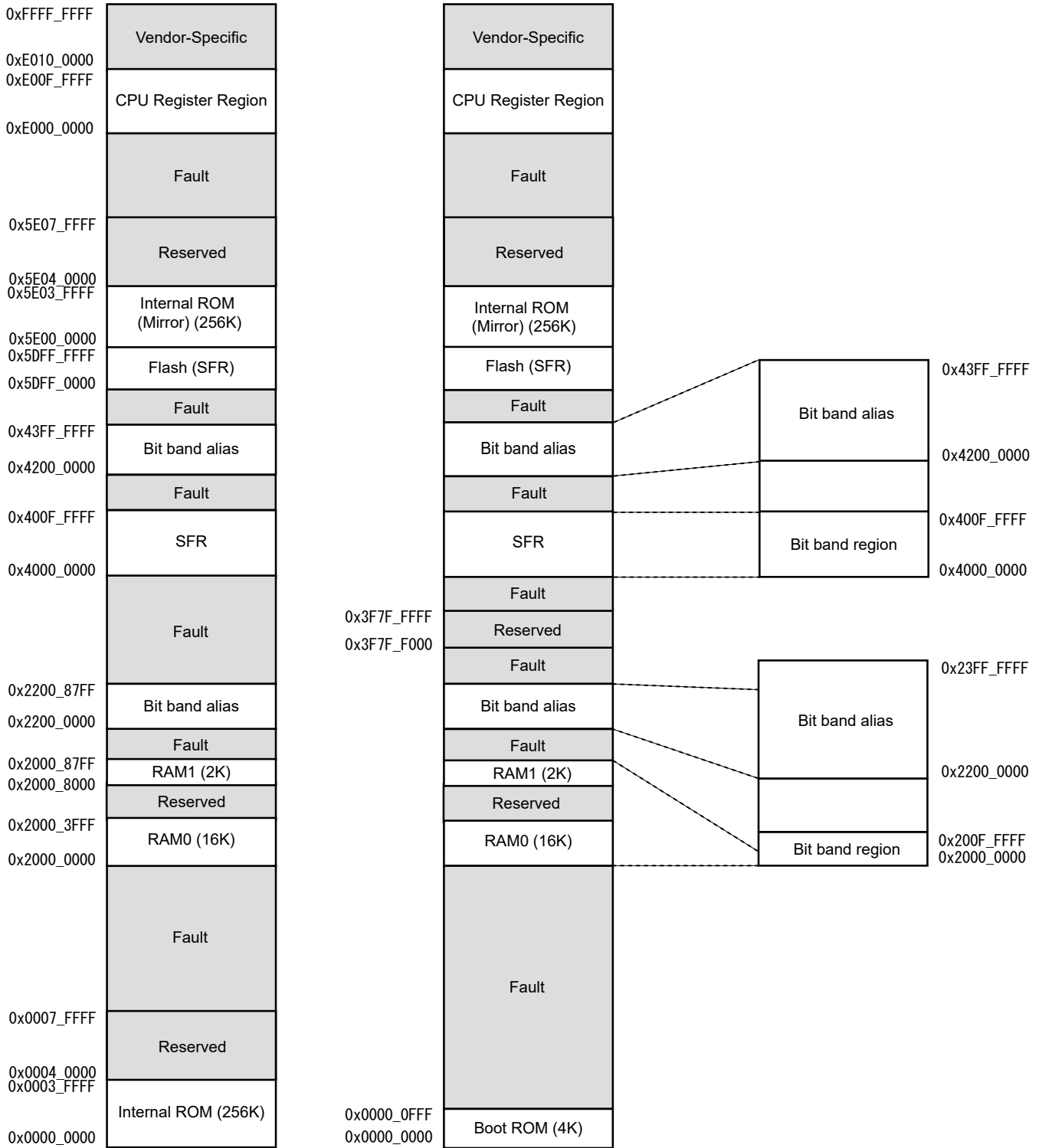
(512KB



Single chip mode

Single boot mode

4-2 (384KB )



Single chip mode

Single boot mode

4-3 (256KB )

## 4.2

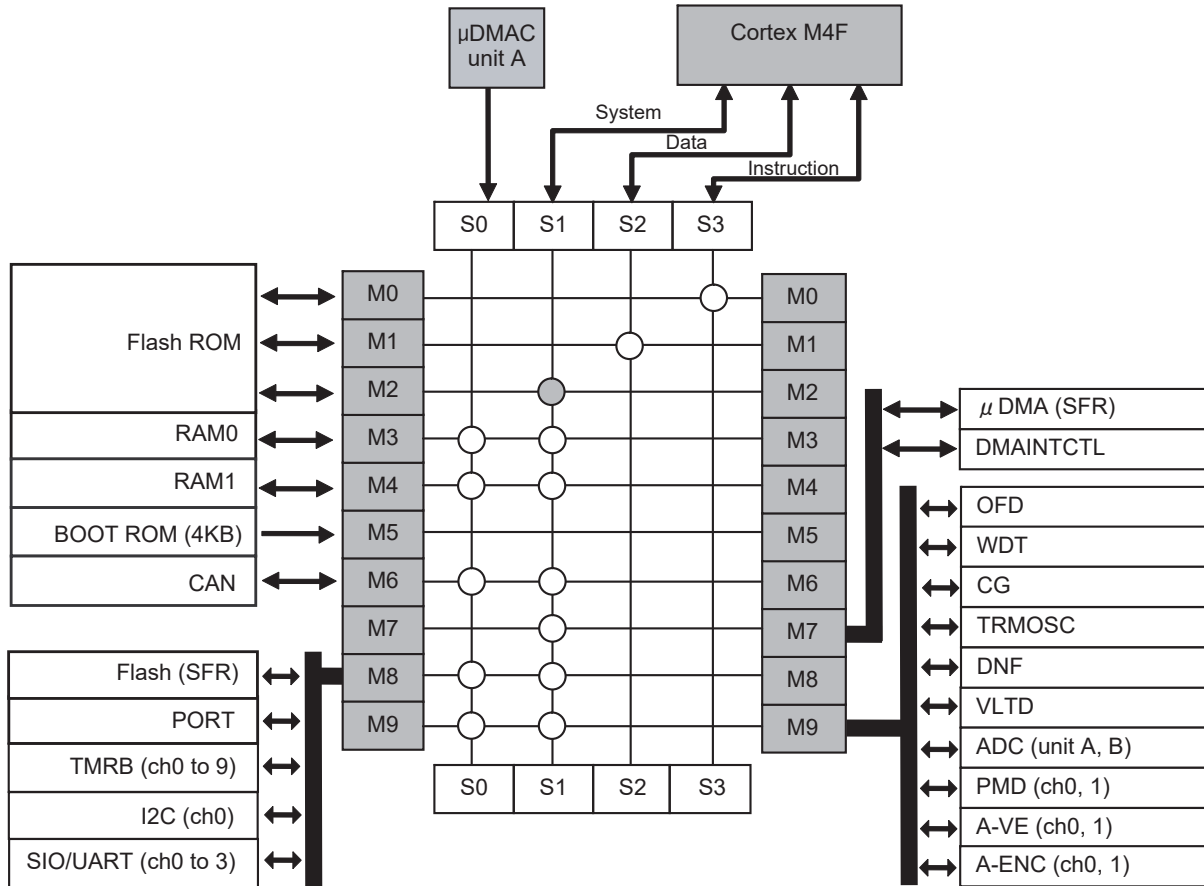
本マイコンでは、CPU コア、 $\mu$ DMA コントローラ 2 種類のバスマスタが搭載されています。

バスマスタは、バスマトリクスのスレーブポート(S0~S5)に接続され、バスマトリクス内で、接続を示す記号(○,●)を経由して、マスタポート(M0~M15)から、周辺機能に接続されます。●は、ミラー領域への接続を示します。

バスマトリクス内の同一マスタライン上に、複数のスレーブが接続されている場合で、同一タイミングで複数のスレーブにアクセスが発生した場合は、スレーブ番号の小さいマスタのアクセスが優先されます。

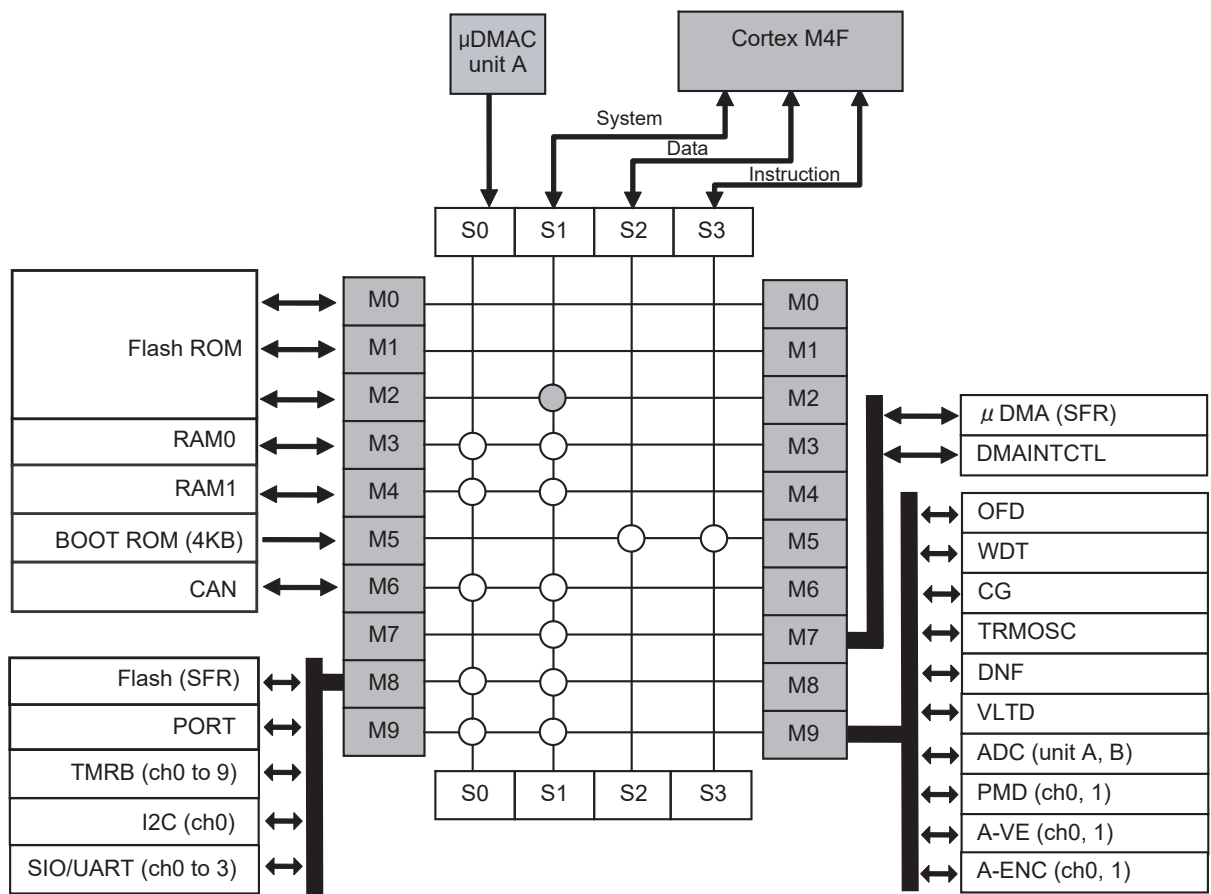
4.2.1

4.2.1.1



4-4 TMPM475FDFG/FZFG/FYFG

4.2.1.2



4-5 TMPM475FDFG/FZFG/FYFG

## 4.2.2

## 4.2.2.1 Code / SRAM

(1)

Start Address	Master		μDMAC unitA	Core S-Bus	Core D-Bus	Core I-Bus
	Slave					
0x0000_0000	Flash ROM	M0 M1	Fault	Fault	o	o
0x0010_0000	Fault	-	Fault	Fault	Fault	Fault
0x2000_0000	RAM0	M3	o	o	Fault	Fault
0x2000_8000	RAM1	M4	o	o	Fault	Fault
0x2000_8800	Fault	-	Fault	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
0x2400_0000	Fault	-	Fault	Fault	Fault	Fault

(2)

Start Address	Master		μDMAC unitA	Core S-Bus	Core D-Bus	Core I-Bus
	Slave					
0x0000_0000	Boot ROM	M5	Fault	Fault	o	o
0x0000_1000	Fault	-	Fault	Fault	Fault	Fault
0x2000_0000	RAM0	M3	o	o	Fault	Fault
0x2000_8000	RAM1	M4	o	o	Fault	Fault
0x2003_0400	Fault	-	Fault	Fault	Fault	Fault
0x2200_0000	Bit band alias	-	Fault	o	Fault	Fault
0x2400_0000	Fault	-	Fault	Fault	Fault	Fault
0x3F7F_F000	Reserved	-	Fault	Reserved	Fault	Fault
0x3F80_0000	Fault	-	Fault	Fault	Fault	Fault

) Reserved

4.2.2.2 Peripheral

Start Address	Master		μDMAC unitA	Core S-Bus	Core D-Bus	Core I-Bus
	Slave		S0	S1	S2	S3
0x4000_0000	Fault	-	Fault	Fault	Fault	Fault
0x4000_5000	CAN	M6	o	o	Fault	Fault
0x4004_C000	μDMAC unitA (SFR)	M7	-	o	Fault	Fault
0x4005_F000	DMAIF	M7	-	o	Fault	Fault
0x400C_0000	PORT	M8	o	o	Fault	Fault
0x400C_4000	TMRB (ch0 to -9)	M8	o	o	Fault	Fault
0x400E_0000	I2C (ch0)	M8	o	o	Fault	Fault
0x400E_1000	SIO/UART(ch0 to 3)	M8	o	o	Fault	Fault
0x400F_1000	OFD	M9	o	o	Fault	Fault
0x400F_2000	WDT	M9	o	o	Fault	Fault
0x400F_3000	CG	M9	o	o	Fault	Fault
0x400F_3200	TRMOSC	M9	o	o	Fault	Fault
0x400F_3400	DNF	M9	o	o	Fault	Fault
0x400F_4000	LVD	M9	o	o	Fault	Fault
0x400F_6000	PMD (ch0,1)	M9	o	o	Fault	Fault
0x400F_7000	A-ENC (ch0,1)	M9	o	o	Fault	Fault
0x400F_8000	A-VE (ch0, 1)	M9	o	o	Fault	Fault
0x400F_D000	ADC (unit A, B)	M9	o	o	Fault	Fault
0x4010_0000	Fault	-	Fault	Fault	Fault	Fault
0x4200_0000	Bit band alias	-	Fault	o	Fault	Fault
0x4400_0000	Fault	-	Fault	Fault	Fault	Fault
0x5DFF_0000	Flash (SFR)	M8	-	o	Fault	Fault
0x5E00_0000	Flash(Mirror)	M2	o	o	Fault	Fault
0x5E08_0000	Fault	-	Fault	Fault	Fault	Fault

## 4.2.3

周辺機能のベースアドレスを示します。周辺機能はすべて SFR0 領域です。

SFR0 領域のうち、制御レジスタ以外のアドレスにはアクセスしないでください。制御レジスタの詳細は、各周辺機能の章を参照してください。

CAN	ch0	0x4000_5000	SFR0
	ch1	0x4000_5020	SFR0
	ch2	0x4000_5040	SFR0
	ch3	0x4000_5060	SFR0
	ch4	0x4000_5080	SFR0
	ch5	0x4000_50A0	SFR0
	ch6	0x4000_50C0	SFR0
	ch7	0x4000_50E0	SFR0
	ch8	0x4000_5100	SFR0
	ch9	0x4000_5120	SFR0
	ch10	0x4000_5140	SFR0
	ch11	0x4000_5160	SFR0
	ch12	0x4000_5180	SFR0
	ch13	0x4000_51A0	SFR0
	ch14	0x4000_51C0	SFR0
	ch15	0x4000_51E0	SFR0
	ch16	0x4000_5200	SFR0
	ch17	0x4000_5220	SFR0
	ch18	0x4000_5240	SFR0
	ch19	0x4000_5260	SFR0
	ch20	0x4000_5280	SFR0
	ch21	0x4000_52A0	SFR0
	ch22	0x4000_52C0	SFR0
	ch23	0x4000_52E0	SFR0
	ch24	0x4000_5300	SFR0
	ch25	0x4000_5320	SFR0
	ch26	0x4000_5340	SFR0
	ch27	0x4000_5360	SFR0
	ch28	0x4000_5380	SFR0
	ch29	0x4000_53A0	SFR0
	ch30	0x4000_53C0	SFR0
	ch31	0x4000_53E0	SFR0
CAN (CAN)	unit 0	0x4000_5400	SFR0
μDMA (μDMAC)	unit A	0x4004_C000	SFR0
	DMAIF	0x4005_F000	SFR0

	PORTA	0x400C_0000	SFR0
	PORTB	0x400C_0100	SFR0
	PORTC	0x400C_0200	SFR0
	PORTD	0x400C_0300	SFR0
	PORTE	0x400C_0400	SFR0
	PORTF	0x400C_0500	SFR0
	PORTG	0x400C_0600	SFR0
	PORTH	0x400C_0700	SFR0
	PORTJ	0x400C_0800	SFR0
	PORTK	0x400C_0900	SFR0
	PORTL	0x400C_0A00	SFR0
	PORTN	0x400C_0C00	SFR0
	PORTP	0x400C_0D00	SFR0
16	/ (TMRB)	ch0	0x400C_4000
		ch1	0x400C_4100
		ch2	0x400C_4200
		ch3	0x400C_4300
		ch4	0x400C_4400
		ch5	0x400C_4500
		ch6	0x400C_4600
		ch7	0x400C_4700
		ch8	0x400C_4800
		ch9	0x400C_4900
	(I2C)	ch0	0x400E_0000
	(SIO/UART)	ch0	0x400E_1000
		ch1	0x400E_1100
		ch2	0x400E_1200
		ch3	0x400E_1300
	(OFD)	unit 0	0x400F_1000
	(WDT)	ch0	0x400F_2000
	/ (CG)		0x400F_3000
	(TRMOSC)	unit 0	0x400F_3200
	(DNF)	unit 0	0x400F_3400
	(VLTD)	unit 0	0x400F_4000
	(PMD)	ch0	0x400F_6000
		ch1	0x400F_6100
	(A-ENC)	ch0	0x400F_7000
		ch1	0x400F_7100
	(A-VE)	ch0	0x400F_8000
		ch1	0x400F_8400
	/ (ADC)	unit A	0x400F_D000
		unit B	0x400F_D100
	/ (FC)	ch0	0x5DFF_0000



## 5

リセットの種類として以下のものがあります。

- ・ パワーオンリセット回路(POR)
- ・ 電圧検出回路(VLTD)
- ・ リセット端子(RESET)
- ・ ウォッチドッグタイマ(WDT)
- ・ 周波数検知回路(OFD)
- ・ CPU のアプリケーション割り込みおよびリセットレジスタ<SYSRESETREQ>ビット

リセットの要因を確認するためには、クロックジェネレータレジスタの CGRSTFLG を参照してください。CGRSTFLG については、例外の章を参照してください。

パワーオンリセット回路、電圧検出回路、ウォッチドッグタイマ、周波数検知回路についてはそれぞれの章を参照してください。

<SYSRESETREQ>によるリセットについては、"ARM ドキュメンテーションセット"を参照してください。

) RAM

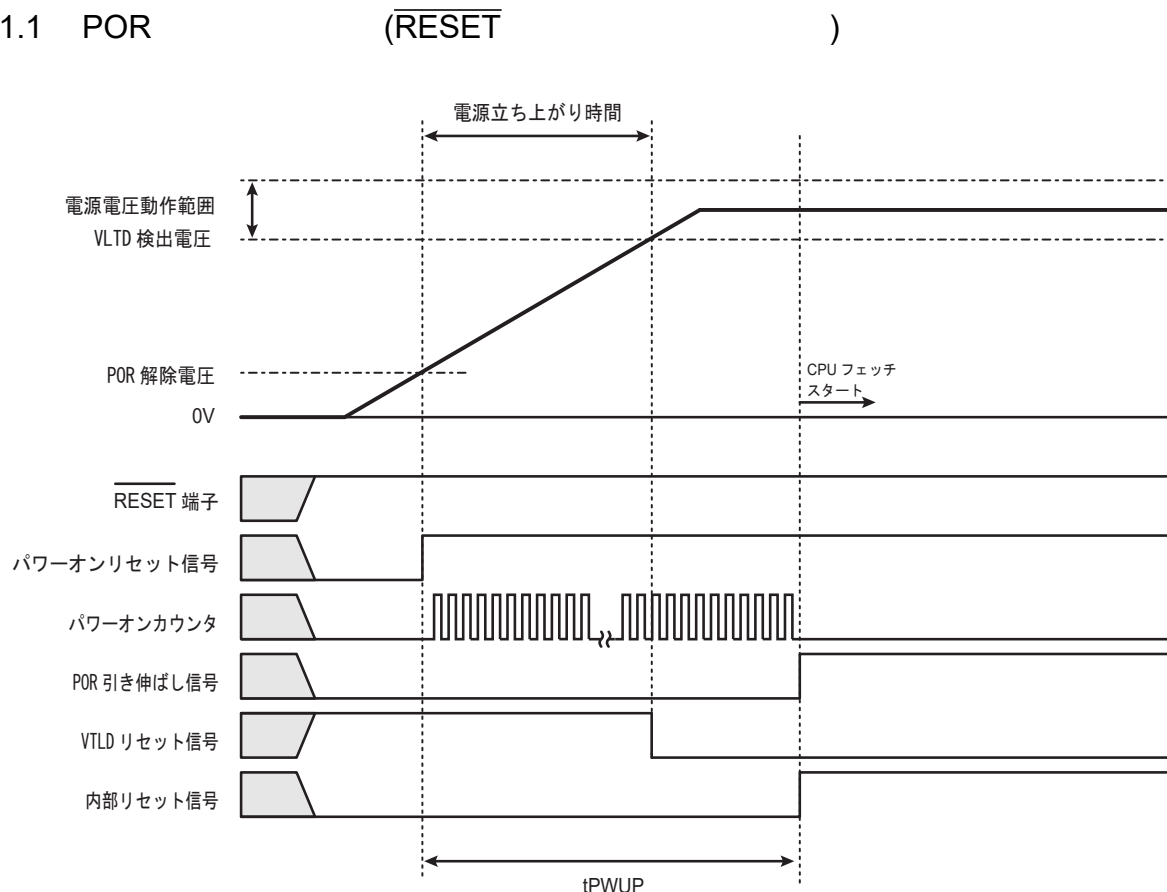
## 5.1

電源投入の際には、内蔵レギュレータ、内蔵フラッシュメモリおよび内蔵高速発振の安定時間を考慮する必要があります。本製品では、これらの機能の安定のための時間を内部回路が自動的に挿入します。この回路は、電源電圧がパワーオンリセット解除電圧を超えると、パワーオンリセット回路(POR)の出力によってパワーオンカウンタが動作を開始し、 $t_{PWUP}$ ( $f_{osc}$ が10MHzの場合に約0.9ms)の間、POR引き伸ばし信号を有効にします。

また、本製品では、電源の電圧検出回路(VLTD)の動作が許可されています。

パワーオンリセット回路の動作については、「パワーオンリセット回路(POR)」の章を参照してください。電源検知回路の動作については、「電源検知回路(VLTD)」の章を参照してください。

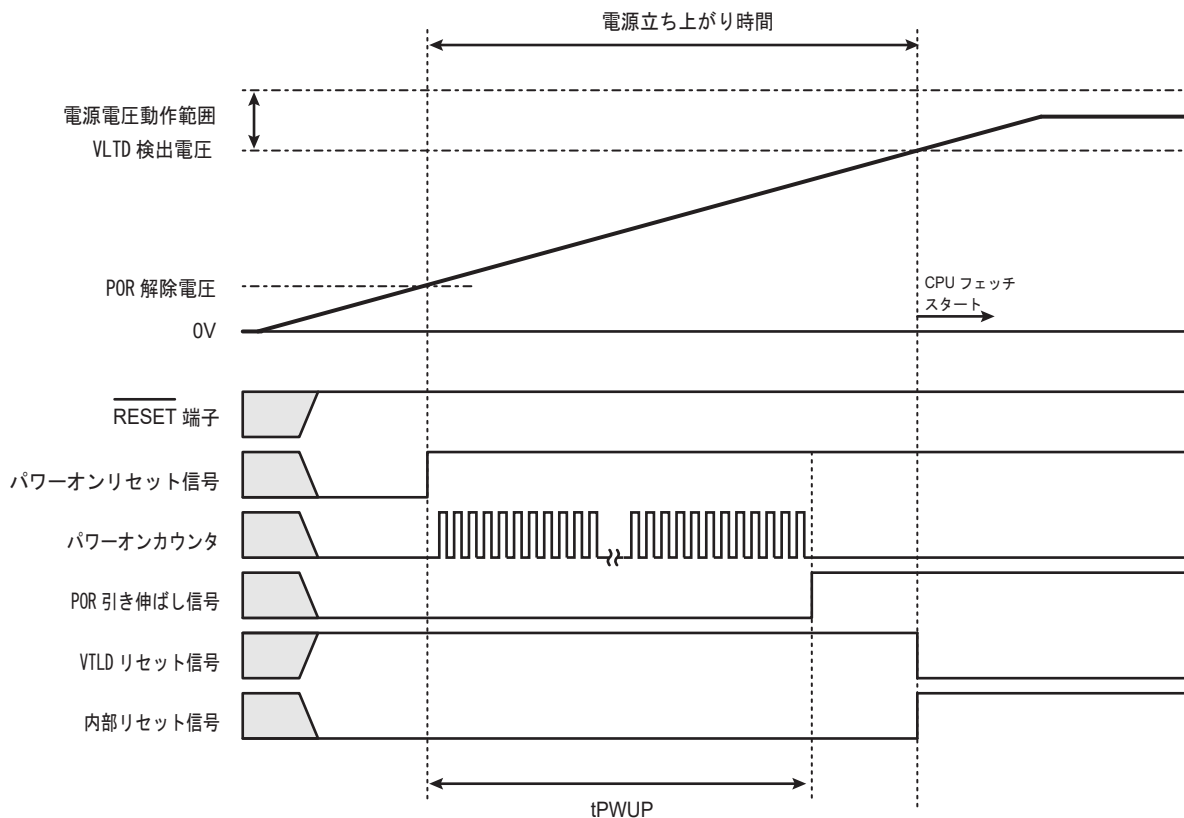
### 5.1.1 POR



POR引き伸ばし信号が"Low"のときに電源電圧をVLTD検出電圧以上に立ち上げると、POR引き伸ばし信号が"High"となるとときに内部リセット信号が"High"となって、リセットが解除されます。

5.1.2 VLTD

( $\overline{\text{RESET}}$ )

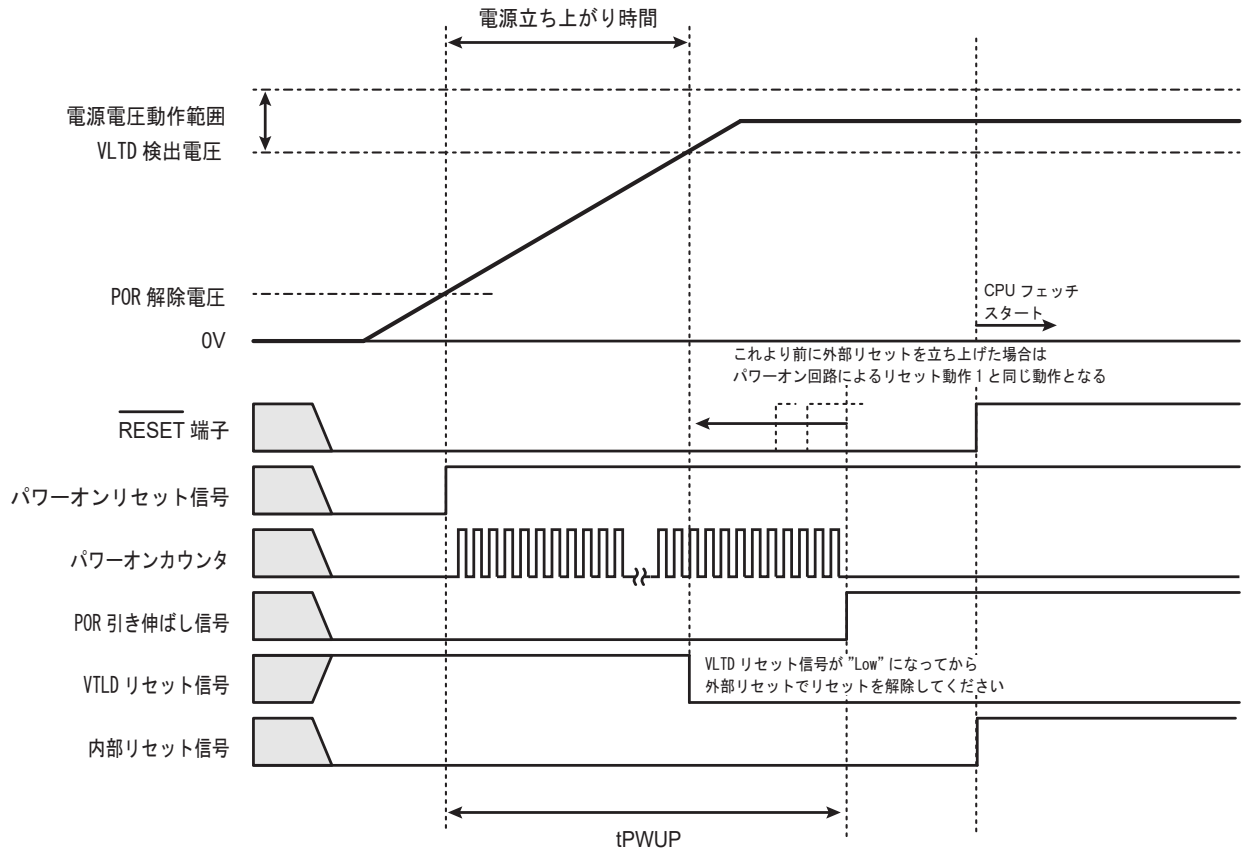


5-2 VLTD

( $\overline{\text{RESET}}$ )

POR 引き伸ばし信号が"High"のときに電源電圧を VLTD 検出電圧以上に立ち上げると、VLTD リセット信号が"Low"となるとときに内部リセット信号が"High"となって、リセットが解除されます。

### 5.1.3 $\overline{\text{RESET}}$ (POR )

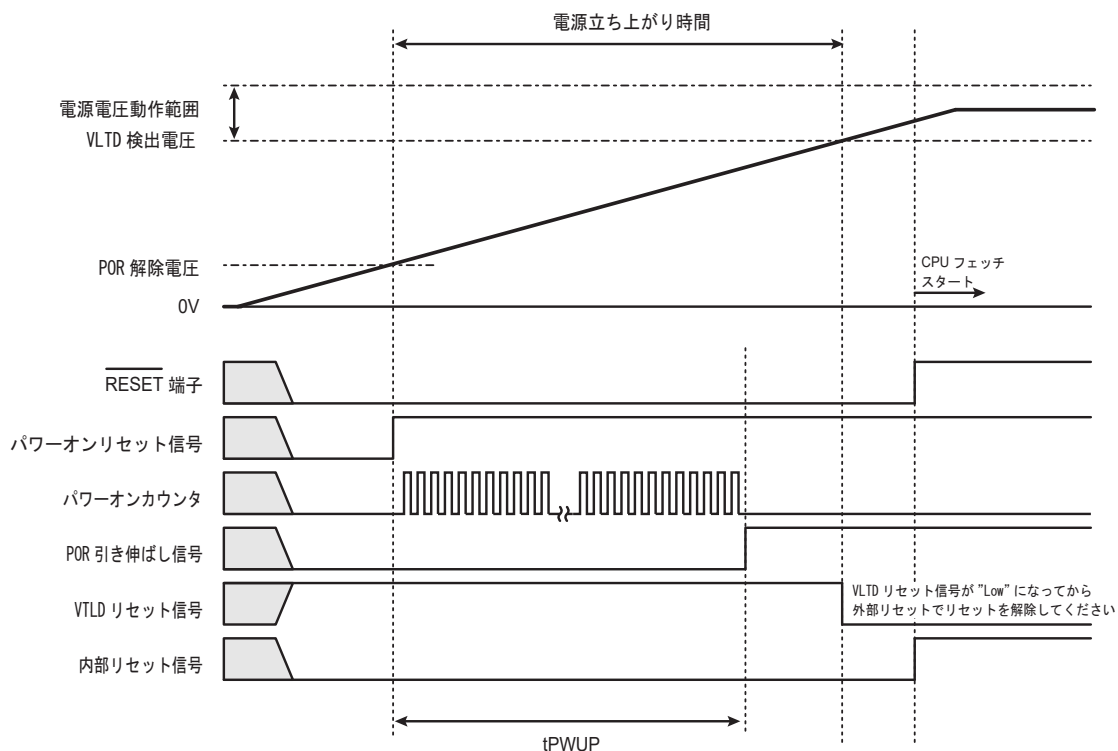


### 5-3 $\overline{\text{RESET}}$ (POR )

POR によるリセットが有効な場合、POR 引き伸ばし信号が "High" になってから、 $\overline{\text{RESET}}$  端子を "High" とするとき内部リセット信号が "High" となって、リセットが解除されます。したがって、VLTD リセット信号が "Low" となる直後から POR 引き伸ばし信号が "High" となる直前まで間に  $\overline{\text{RESET}}$  端子を "High" にする場合は、1.1.1 の POR によるリセット動作と同じ動作になります。

なお、VLTD リセット信号が "High" のとき、すなわち、電源電圧が動作電圧未満のときには  $\overline{\text{RESET}}$  端子を "High" にしないでください。

5.1.4  $\overline{\text{RESET}}$  (VLTD )



5-4  $\overline{\text{RESET}}$  (VLTD )

VLTD によるリセットが有効な場合、VLTD リセット信号が "Low" となつてから、 $\overline{\text{RESET}}$  端子を "High" とするとき内部リセット信号が "High" となつて、リセットが解除されます。

なお、VLTD リセット信号が "High" のとき、すなわち、電源電圧が動作電圧未満のときには  $\overline{\text{RESET}}$  端子を "High" にしないでください。

## 5.2

### 5.2.1

本製品にリセットをかけるには、電源電圧が動作範囲内であり、高速発振器の発振が安定した状態で、RESET 端子を少なくとも 12 システムクロック (fosc が 10MHz の場合に最小 1.2  $\mu$ s) 間"Low"にしてください。RESET 端子が"High"になってから内部リセットが解除されます。

## 5.3

リセット解除後は、ほとんどのコアの制御レジスタや周辺機能の制御レジスタ(SFR)は初期化されません。コア内部のシステムデバッグコンポーネント(FPB、DWT、ITM)レジスタ、クロックジェネレータレジスタの CGRSTFLG および FLASH 関連レジスタの FCSECBIT はコールドリセットでのみ初期化されません。

リセット解除後、内蔵高速発振器のクロックで動作を開始します。必要に応じて外部クロック、PLL 逡倍回路の設定を行ってください。

## 6

## /

## 6.1

クロック/モード制御は、内部／外部発振、クロックギア、プリスケールクロックの選択、PLL(通倍回路)や発振器のウォーミングアップ等の制御を行います。

また、低消費電力モードがあり、モード遷移を行うことで電力の消費を抑えることが可能です。

本章では、クロックの制御および動作モードとモード遷移について説明します。

## 6.2

### 6.2.1

クロック/モード制御のレジスタとアドレスを以下に示します。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照してください。

		Address(Base+)
	CGSYSCR	0x0000
	CGOSCCR	0x0004
	CGSTBYCR	0x0008
PLL	CGPLLSEL	0x000C
	CGCKSTP	0x0040

6.2.2 CGSYSCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	FPSEL	-	PRCK		
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	GEAR		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-18		R	"0"
17-16		R/W	"01"
15-13		R	"0"
12	FPSEL	R/W	fperiph 0: fgear 1: fc fperiph fc fperiph
11		R	"0"
10-8	PRCK[2:0]	R/W	( T0) 000: fperiph 100: fperiph/16 001: fperiph/2 101: fperiph/32 010: fperiph/4 110: Reserved 011: fperiph/8 111: Reserved ( T0) T0 fsys
7-3		R	"0"
2-0	GEAR[2:0]	R/W	(fgear) 000: fc 100: fc/2 001: Reserved 101: fc/4 010: Reserved 110: fc/8 011: Reserved 111: fc/16 T0 fsys

6.2.3 CGOSCCR( )

	31	30	29	28	27	26	25	24
bit symbol	WUODR							
	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	WUODR				WUPSEL2	-	OSCSEL	XEN2
	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	XEN1
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	WUPSEL1	PLLON	WUEF	WUEON
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-20	WUODR[11:0]	R/W	16 12
19	WUPSEL2	R/W	( 1) 0: (fosc) 1: (feosc)
18		R/W	Reserved ("0" "1" )
17	OSCSEL	R/W	(fosc) 0: (fosc) 1: (feosc)  <OSCSEL>
16	XEN2	R/W	(IOSC) 0: 1: (fosc)  6.3.5
15-12		R/W	"0"
11-10		R	"0"
9		R/W	"0"
8	XEN1	R/W	(EOSC) 0: 1:  6.3.5
7-4		R	"0"
3	WUPSEL1	R/W	(WUP) "0"
2	PLLON	R/W	(PLL) 0: 1: (PLL)  6.3.5

Bit	Bit Symbol	Type	
1	WUEF	R	0: 1:
0	WUEON	W	1:  ("1" ) <WUEF> "0" <WUON> "0" "0"

) STOP

STOP

<OSCSEL>

6.2.4 CGSTBYCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	RXEN
	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	STBY		
	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	
31-18		R	"0"
17		R/W	"0"
16-10		R	"0"
9		R/W	"0"
8	RXEN	R/W	STOP "1"
7-3		R	"0"
2-0	STBY[2:0]	R/W	000: Reserved 001: STOP 010: Reserved 011: IDLE 100: Reserved 101: Reserved 110: Reserved 111: Reserved  STOP (IOSC) (EOSC)

6.2.5 CGPLLSEL(PLL )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PLLSET							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PLLSET							PLLSEL
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-1	PLLSET	R/W	PLL ( ) 0x591E: 10MHz 80MHz 0x5926: 10MHz 100MHz 0x59AE: 10MHz 120MHz
0	PLLSEL	R/W	(fc) 0: fosc 1: f <sub>PLL</sub> (fc)

6.2.6 CGCKSTP( )

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

Bit	Bit Symbol	Type	
31-3		R	"0"
2	CANSTP	R/W	CAN 0: 1:  CAN
0-1		R	"0"

) CGCKLSTP<CANSTP> "0" "1" CAN

## 6.3

### 6.3.1

クロックの一覧を以下に示します。

feosc	:	(EOSC)	
fiosc	:	(IOSC)	
fosc	:		
f <sub>PLL</sub>	:	(4 )	
fc	:	fosc f <sub>PLL</sub>	( )
fgear	:		
f <sub>sys</sub>	:	fgear	( )
f <sub>periph</sub>	:	CGSYSCR<FPSEL>	
T0	:	CGSYSCR<PRCK[2:0]>	( )

高速クロック **fc** と、プリスケールクロック  $\phi$  T0 は以下のように分周することが可能です。

: fc, fc/2, fc/4, fc/8, fc/16

: f<sub>periph</sub>, f<sub>periph</sub>/2, f<sub>periph</sub>/4, f<sub>periph</sub>/8, f<sub>periph</sub>/16, f<sub>periph</sub>/32

### 6.3.2

リセット動作により、クロックの設定は下記のような状態に初期化されます。

(EOSC)	:	
(IOSC)	:	
(PLL)	:	
	:	fc ( )

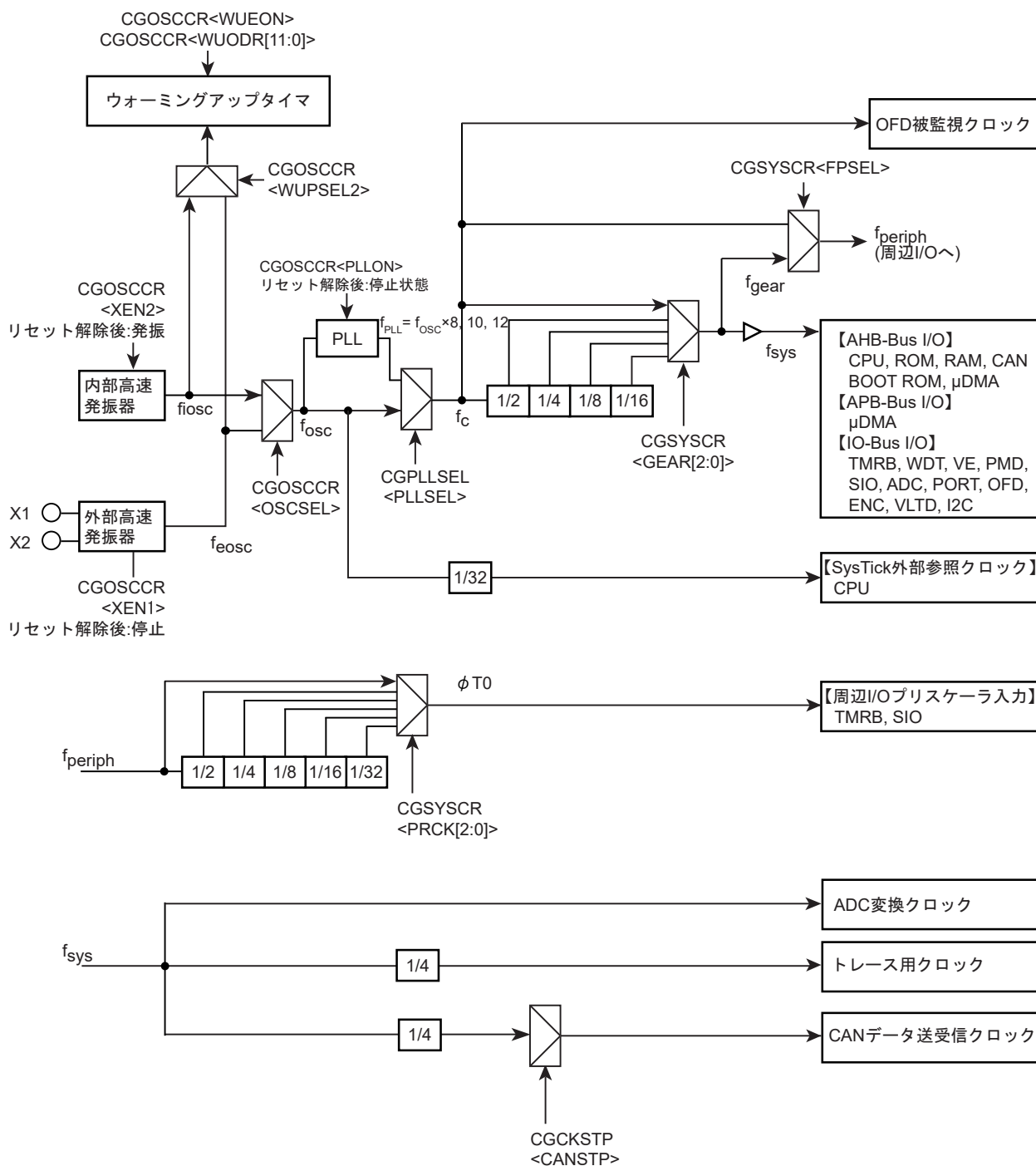
リセット動作により、すべてのクロックの設定が **f<sub>OSC</sub>** と同じになります。

fc	=	fosc
f <sub>sys</sub>	=	fc (= fosc)
f <sub>periph</sub>	=	fc (= fosc)
T0	=	f <sub>periph</sub> (= fosc)

## 6.3.3

クロック系統図を図 6-1 に示します。

セレクタに入力されるクロックのうち、矢印付きのものがリセット後の初期状態として選択されます。



6-1

### 6.3.4 (PLL)

高速発振器の出力クロック  $f_{osc}$  を 8, 10, 12 逡倍したクロック ( $f_{PLL}$ ) を出力する回路です。これにより、発振器への入力周波数は低く内部クロックは高速にすることが可能です。

PLL はリセット解除後、ディセーブル状態です。PLL を有効にするためには、CGOSCCR<PLLON> を "1" に設定し、CGPLLSEL<PLLSEL> を "1" に設定することで  $f_{osc}$  を 8, 10, 12 逡倍したクロック ( $f_{PLL}$ ) を出力することができます。なお、PLL の動作が安定するまでの時間をウォーミングアップ機能等を用いて確保する必要があります。

#### 6.3.4.1

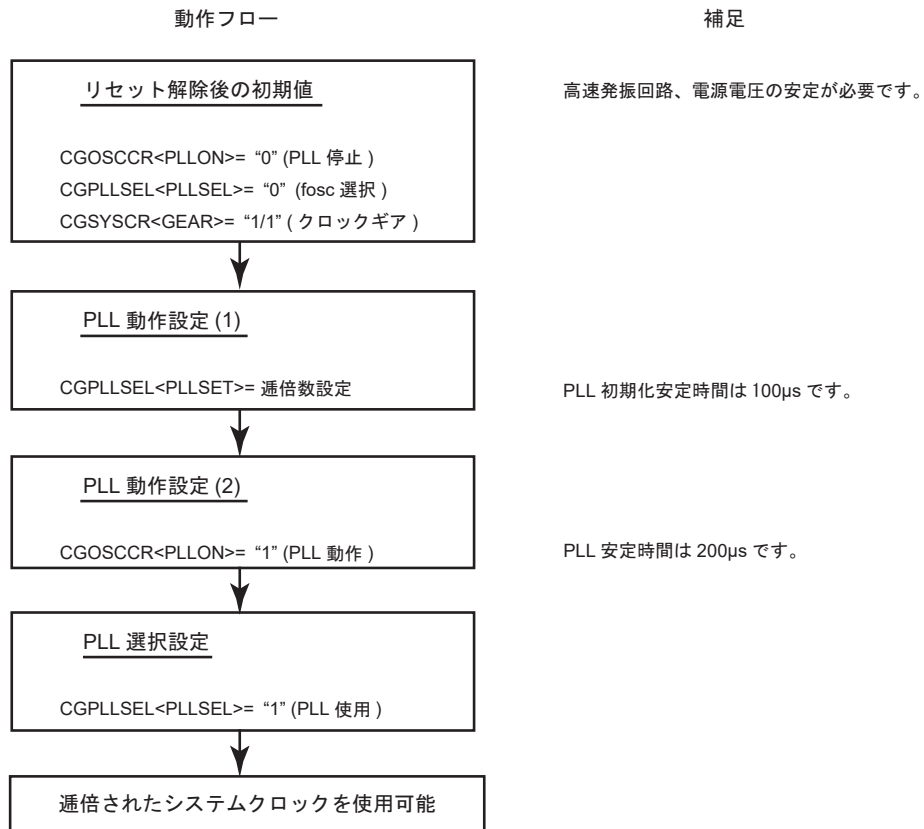
PLL 動作開始および、逡倍値の変更の際にはウォーミングアップ機能等を用いて安定時間を確保する必要があります。

動作を開始するときはロックアップ時間として約 200 $\mu$ s 必要です。

逡倍数の変更を行う場合、まず CGPLLSEL<PLLSEL> = "0" として逡倍クロックを使用しない設定に切り替えた上で <PLLON> を "0" として PLL を停止します。<PLLSET> の逡倍値を変更し、PLL の初期化時間として約 100 $\mu$ s 経過後に <PLLON> を "1" として PLL の動作を開始します。その後、ロックアップ時間(PLL 安定時間)を確保してください。

## 6.3.4.2 PLL

以下にリセット解除後の PLL 設定シーケンスを示します。



## 6.3.5

ウォーミングアップ機能は、ウォーミングアップタイマを用いて発振子の発振安定時間や、PLLの安定時間を確保するための機能です。発振が安定している外部クロックなどを使用する場合にはウォーミングアップを行う必要はありません。

詳細機能については「6.6.6 ウォーミングアップ」にて説明します。

)

ウォーミングアップ機能は、STOPモードからの復帰の際にも使用されます。この場合には、低消費電力モードからの復帰割り込みが発生すると自動的にタイマがカウントを開始し、設定された時間のカウント終了後、システムクロックが出力されCPUが動作を開始します。

ウォーミングアップ機能の設定方法を説明します。

## 1. クロックの選択

ウォーミングアップカウンタのカウントアップクロックをCGOSCCR<WUPSEL1>および<WUPSEL2>で選択します。( <WUPSEL1>は"0"を、<WUPSEL2>は"0"または"1"を設定してください。)

## 2. ウォーミングアップカウンタ設定値の算出

ウォーミングアップ時間はCGOSCCR<WUODR[11:0]>により任意の値が設定可能です。CGOSCCR<WUODR[11:0]>の設定値は、以下の計算式から算出し、下位4ビットを切り捨てて上位12ビットを設定します。

以下にウォーミングアップ時間の計算式と設定例を示します。

$$\text{ウォーミングアップサイクル数} = \frac{\text{ウォーミングアップ時間}}{\text{ウォームアップクロック周期}}$$

<例>高速発振子 8MHz 使用時、ウォーミングアップ時間 5ms を設定する場合

$$\frac{\text{ウォーミングアップ時間}}{\text{ウォームアップクロック周期}} = \frac{5\text{ms}}{1/8\text{MHz}} = 40,000\text{サイクル} = 0x9C40$$

下位4ビットを切り捨て、0x9C4をCGOSCCR<WUODR[11:0]>に設定します。

## 3. ウォーミングアップの開始および終了確認

ソフトウェア(命令)によりウォーミングアップタイマの開始および終了確認を行う場合、CGOSCCR<WUEON>に"1"を設定することでウォーミングアップを開始します。また、終了の確認は<WUEF>で行います。<WUEF>が"1"でウォーミングアップ中、"0"で終了を示します。

)

以下に、ウォーミングアップ機能の設定例を示します。

```
CGOSCCR<WUPSEL1> = "0"           (           )  
CGOSCCR<WUPSEL2> = "1"           :           (1: (feosc))  
CGOSCCR<WUODR[11:0]> = "0x9C4"   :  
6.3.6  
CGOSCCR<WUEON>="1"               :           (WUP)  
CGOSCCR<WUEF>                    : "0"(WUP )
```

6.3.6

システムクロックは高速クロックのみですが、内部発振器と外部発振器のどちらかを選択して使用します。リセット解除時は内部発振器が有効（発振）となっており、外部発振器は停止しています。高速クロックは分周することができます。

- ・ X1, X2 入力周波数: 10MHz
- ・ 内部発振入力周波数:10MHz
- ・ クロックギア: 1/1, 1/2, 1/4, 1/8, 1/16 (リセット後は 1/1 分周)

6-1 ( MHz)

feosc fiosc				(PLL = OFF, CG = 1/1)	(CG) PLL = ON					(CG) PLL = OFF				
					1/1	1/2	1/4	1/8	1/16	1/1	1/2	1/4	1/8	1/16
10	8	1.25	80	10	80	40	20	10	5	10	5	2.5	1.25	-
10	10	1.25	100	10	100	50	25	12.5	6.25	10	5	2.5	1.25	-
10	12	1.25	120	10	120	60	30	15	7.5	10	5	2.5	1.25	-

- 1) PLL=ON / OFF CGOSCCR<PLLON>
- 2) CGSYSCR<GEAR[2:0]>
- 3) PLL=OFF 1/16
- 4) SysTick 1/16
- 5) 85°C ~ 105°C 32MHz PLL

内部発振器から外部発振器への切り替え手順を以下に示します。

1. CGOSCCR<WUODR[11:0]> = " " :
2. CGOSCCR<XEN1> = "1" :
3. CGOSCCR<WUPSEL2> = "1" : (1: (feosc))
4. CGOSCCR<WUEON>="1" : (WUP)  
CGOSCCR<WUEF> : "0" ( WUP )
5. CGOSCCR<OSCSEL> = "1" : (feosc)
6. CGOSCCR<OSCSEL> : (<OSCSEL> = "1")
7. CGOSCCR<XEN2> = "0" :

### 6.3.7

周辺機能には、それぞれにクロックを分周するプリスケータがあります。これらのプリスケータへ入力するクロック  $\phi T0$  は、CGSYSCR<FPSEL>から選択されたクロック  $f_{periph}$  をさらにCGSYSCR<PRCK[2:0]>にて分周することが可能です。リセット後の  $\phi T0$  は  $f_{periph}/1$  が選択されます。

$$\left( Tn \quad f_{sys} \right) \quad Tn \quad Tn < f_{sys}$$

## 6.4

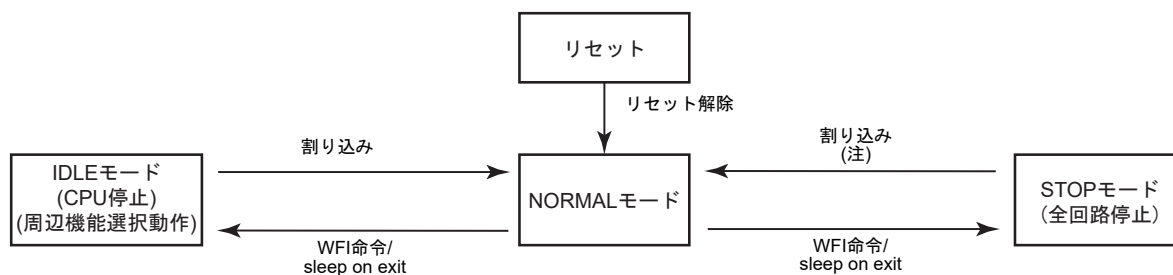
### 6.4.1

動作モードとしてシステムクロックに高速クロックを使用する NORMAL モードがあります。

また、プロセッサコアの動作を停止して電力の消費を抑える低消費電力モードとして、IDLE モード、STOP モードがあります。

図 6-2 にモード状態遷移図を示します。

WFI 命令、Sleep-on-exit については、"ARM ドキュメンテーションセット"を参照してください。



6-2

)

(NORMAL)

STOP

6.6.6

## 6.5

### 6.5.1 NORMAL

CPU コアおよび周辺回路を高速クロックで動作させるモードです。リセット解除後は、NORMAL モードになります。

## 6.6

低消費電力モードには、IDLE, STOP モードがあります。低消費電力モードに移行するには、システムコントロールレジスタ CGSTBYCR<STBY[2:0]>にてモードを選択し、WFI(Wait For Interrupt)命令を実行します。WFI 命令によって低消費電力モードへ移行した場合、低消費電力モードからの復帰はリセットまたは割り込み発生により行われます。割り込みで復帰する場合には、設定を行っておく必要があります。詳細は「例外」の章の「割り込み」を参照してください。

- 1) WFE (Wait For Event)
- 2) Cortex-M0 SLEEPDEEP  
<SLEEPDEEP>
- 3)

IDLE, STOP モードの特長は次のとおりです。

### 6.6.1 IDLE

CPU のみが停止するモードです。周辺機能は、各モジュールの中のレジスタに IDLE モード時の動作/停止設定レジスタを 1 ビット持ち、IDLE モードでの動作設定が可能です。IDLE モード時に動作停止に設定された周辺機能は、IDLE モードへ遷移した時の状態で停止します。

以下に IDLE モードでの動作を設定できる周辺機能の一覧を示します。設定方法は、各機能の章を参照してください。

- 16 ビットタイマ/イベントカウンタ(TMRB)
- シリアルチャネル(SIO/UART)
- シリアルバスインタフェース(I2C)
- ベクトルエンジン(A-VE)

### 6.6.2 STOP

一部の回路を除き、内部発振器も含めてすべての内部回路が停止します。STOP モードが解除されると内蔵発振器が発振を開始し、NORMAL モードへ復帰します。

STOP モード中は、ポートレジスタの設定により端子の状態を保持することができます。STOP モード時の端子状態を表 6-2 に示します。

6-2 STOP

		STOP ( )
	RESET, MODE	
	X1	
	X2	"High"
	PAx ~ PPx	PxIE[m]
		PxCR[m]
	SWCLK, SWDIO TRST, TCK, TMS, TDI	PxIE[m]
	SWDIO, SWV TDO, TRACECLK TRACEDATA0/1/2/3	PxCR[m]
	INT0 ~ INTF	PxIE[m]
PMD	UO0, VO0, WO0, XO0, YO0, ZO0 UO1, VO1, WO1, XO1, YO1, ZO1	PxCR[m]
		PxIE[m]
		PxCR[m]

) x m

### 6.6.3

低消費電力モード選択は、CGSTBYCR<STBY[2:0]>の設定で選択されます。

表 6-3 に<STBY[2:0]>の設定より選択されるモードを示します。

6-3

	CGSTBYCR <STBY[2:0]>
STOP	001
IDLE	011

)

### 6.6.4

各モードにおける動作状態を表 6-4 に示します。

6-4

Block	NORMAL	IDLE	STOP
Processor core	o	-	-
μDMAC	o	o	-
I/O port	o	o	-( 1)
PMD	o	o	-
A-ENC	o	o	-
OFD	o	o	Δ
ADC	o	o	-
CAN	o	o	-
A-VE	o		-
SIO/UART	o		-
I2C	o		-
TMRB	o		-
WDT	o	-	Δ
VLTD	o	o	o( 2)
POR	o	o	o( 2)
DNF	o	o	-
CG	o	o	-
PLL	o	o	Δ
(fc)	o	o	-

o :  
 : ( / )  
 - : Block  
 Δ : Block

- 1)
- 2)

## 6.6.5

低消費電力モードからの解除は、割り込み、リセットによって行うことができます。使用できる低消費電力モード解除ソースは、低消費電力モードにより決まります。詳細を表 6-5 に示します。

6-5

		IDLE	STOP
	INT0 ~ INTF ( 1 )	o	o
	INTENC00, INTENC01 INTENC10, INTENC11	o	3
	INTVCN0, INTVCT0 INTVCN1, INTVCT1	o	3
	INTPWM0, INTEMG0, INTOVV0 INTPWM1, INTEMG1, INTOVV1	o	3
	INTRX0 ~ 3, INTTX0 ~ 3	o	3
	INTADACPA, INTADBCPA INTADACPB, INTADBCPB INTADATMR, INTADBTRM INTADASFT, INTADBSFT INTADAPDA, INTADBPDA INTADAPDB, INTADBPDB	o	3
	INTTB00 ~ INTTB90 INTCAP00 ~ INTCA90 INTTB01 ~ INTTB91 INTCAP01 ~ INTCAP91	o	3
	INTDMACATC, INTDMACAERR	o	3
	INTCANRX, INTCANTYX, INTCANGB	o	3
	SysTick	o	3
	(OFD)	o	3
( $\overline{\text{RESET}}$ , POR, VLTD)	o	o	

o: (RESET )  
x:

1) IDLE STOP

2) CPU

3) 6.6.6

- 割り込み要求による解除

割り込みによって低消費電力モードを解除する場合、CPU で割り込みが検出されるよう準備しておく必要があります。STOP モードの解除に使用する割り込みは、CPU の設定のほかにクロックジェネレータで割り込み検出の設定を行う必要があります、またデジタルノイズフィルタ回路を禁止にする必要があります。

- リセットによる解除

$\overline{\text{RESET}}$  端子、POR、VLTD によるリセットですべての低消費電力モードからの解除を行うことができます。

OFD によるリセットでは、IDLE モードからの解除を行うことができます。

リセットで解除した場合には通常のリセット動作と同様になり、解除後は NORMAL モードですべてのレジスタが初期化された状態になります。

- SysTick 割り込みによる解除

SysTick 割り込みは IDLE モードでのみ使用可能です。

割り込みの詳細に関しては、「例外」の章の「割り込み」をご参照ください。

6.6.6

モード遷移の際には内部発振器の安定のためウォーミングアップが必要な場合があります。

STOP モードから NORMAL モードへの遷移では、自動的にウォーミングアップ用カウンタが起動され、設定されているウォーミングアップ時間経過後にシステムクロックの出力が開始されます。このため、STOP モードに移行する命令を実行する前に CGOSCCR<WUPSEL1><WUPSEL2>でウォーミングアップする発振器の選択(注 1)および、<WUODR>でウォーミングアップ時間の設定(注 2)を行ってください。

- 1) TMPM475FDFG/FZFG/FYFG                      CGOSCCR<WUPSEL1>=0
- 2) STOP                      PLL                      STOP                      PLL                      200 μs
- 3)    CGOSCCR<WUEON>

各動作モード遷移時におけるウォーミングアップ有無を表 6-6 に示します。

6-6

NORMAL → IDLE	
NORMAL → STOP	
IDLE → NORMAL	
STOP → NORMAL	

)

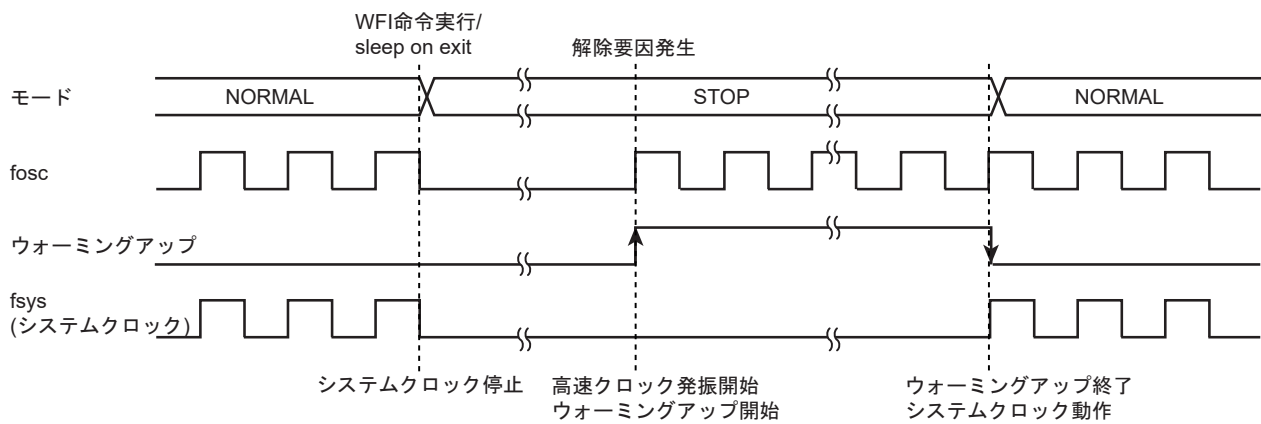
## 6.6.7

モード遷移の際の、クロック動作について示します。

### 6.6.7.1 NORMAL → STOP → NORMAL

STOP モードから NORMAL モードへ復帰する場合、ウォーミングアップは自動的に起動します。STOP モードへ遷移する前にウォーミングアップ時間の設定を行ってください。

リセットで NORMAL モードへ復帰する場合にはウォーミングアップは行われません。コールドリセット時と同じリセットを入力してください。



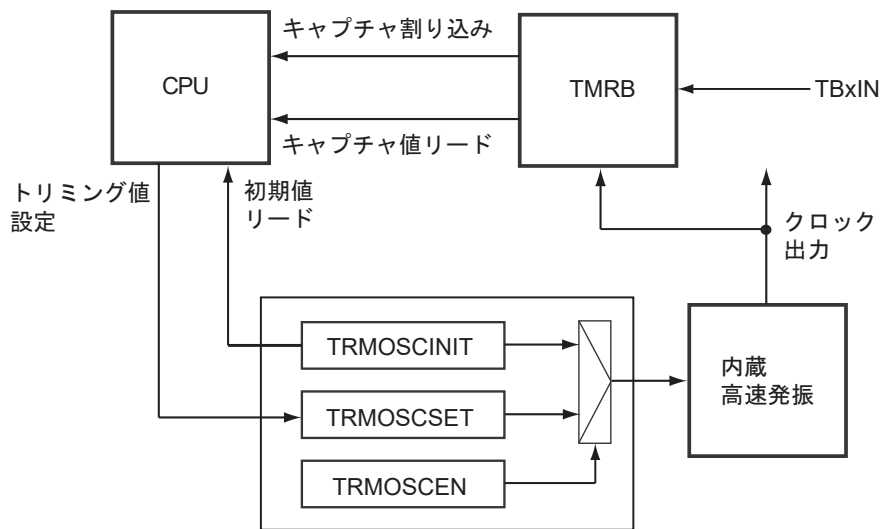
## 7

TMPM475FDFG/FZFG/FYFG には、内蔵高速発振の周波数を調整する機能があります。

## 7.1

内蔵高速発振調整機能は、16 ビットタイマ/ イベントカウンタ(TMRB) のパルス幅測定機能を使用して周波数の調整を行います。

図 7-1 に機能ブロック図を示します。



7-1

## 7.2

### 7.2.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

(x=0~7)		Address(Base+)
	TRMOSCxPRO	0x0000
	TRMOSCxEN	0x0004
	TRMOSCxINIT	0x0008
	TRMOSCxSET	0x000C

### 7.2.2 TRMOSCxPRO ( )

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	PROTECT[7:0]	R/w	0xC1 : 0xC1 : "0xC1" TRMOSCxEN, TRMOSCxINIT, TRMOSCxSET

7.2.3 TRMOSCxEN ( )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	TRIMEN	R/W	0 : 1 : "1" TRIMOSCxINIT TRMOSCxSET

7.2.4 TRMOSCxINIT ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	TRIMINITC					
	0	0						
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	TRIMINITF			
	0	0	0	0				

Bit	Bit Symbol	Type	
31-14		R	"0"
13-8	TRIMINITC [5:0]	R	
7-4		R	"0"
3-0	TRIMINITF[3:0]	R	

粗トリミング、微トリミングの具体的な設定と調整値については、「7.3.2 調整範囲」を参照してください。

7.2.5 TRMOSCxSET ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	TRIMSETC					
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	TRIMSETF			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-14		R	"0"
13-8	TRIMSETC [5:0]	R/W	
7-4		R	"0"
3-0	TRIMSETF[3:0]	R/W	

粗トリミング、微トリミングの具体的な設定と調整値については、「7.3.2 調整範囲」を参照してください。

## 7.3

### 7.3.1

周波数の調整は、粗トリミング値と微トリミング値で行います。

出荷時の設定値は、TRMOSCINIT<TRIMINITC> および<TRIMINITF> で確認できます。変更する設定値は、TRMOSCSET<TRIMSETC> および<TRIMSETF> に設定します。TRMOSCEN<TRIMEN> に"1"を設定することで内蔵高速発振器の設定値が切り替わります。

```
)          TRMOSCSET TRMOSCEN
          TRMOSCPRO<PROTECT>  "0xC1"
```

### 7.3.2

粗トリミングは平均 0.8% ステップで、およそ-19%~+32% の調整が可能です。

微トリミングは 0.1%ステップで、-0.8%~+0.7%の調整が可能です。

表 7-1 に調整範囲を示します。

1) 1                  typ.                                  † 0.6%                                  † 0.1%

2)    /

7-1

<TRIMSETC[5:0]>	(typ.)
011111	+32.0%
011110	+30.6%
011101	+29.2%
011100	+27.8%
011011	+26.5%
000011	+2.3%
000010	+1.5%
000001	+0.8%
000000	±0%
111111	0.7%
111110	1.5%
100100	16.9%
100011	17.4%
100010	17.9%
100001	18.3%
100000	18.8%

<TRIMSETF[3:0]>	(typ.)
0111	+0.7%
0001	+0.1%
0000	±0%
1111	-0.1%
1110	-0.2%
1000	-0.8%

## 8

この章では、例外の特長, 種類, 処理について概略を説明します。

例外は CPU のアーキテクチャと深くかかわる部分ですので、必要に応じて"ARM ドキュメンテーションセット Cortex-M4F 用"もご覧ください。

### 8.1

例外は CPU に対し現在実行中の処理を中断して別の処理に移ることを要求するものです。

例外には、何らかの異常な状態が起こったときや例外を発生する命令を実行したときに発生するものと、外部端子や周辺機能からの割り込み要求信号といった、ハードウェアによる要因で発生する割り込みがあります。

すべての例外は優先度にしたがって CPU 内にあるネスト型ベクタ割り込みコントローラ(NVIC)によって処理されます。例外が発生すると、CPU はそのときの状態をスタックに退避し、割り込み処理ルーチンへ分岐します。割り込み処理ルーチンの実行後、スタックに退避した情報は自動的に復帰されます。

#### 8.1.1

例外には以下のようなものがあります。

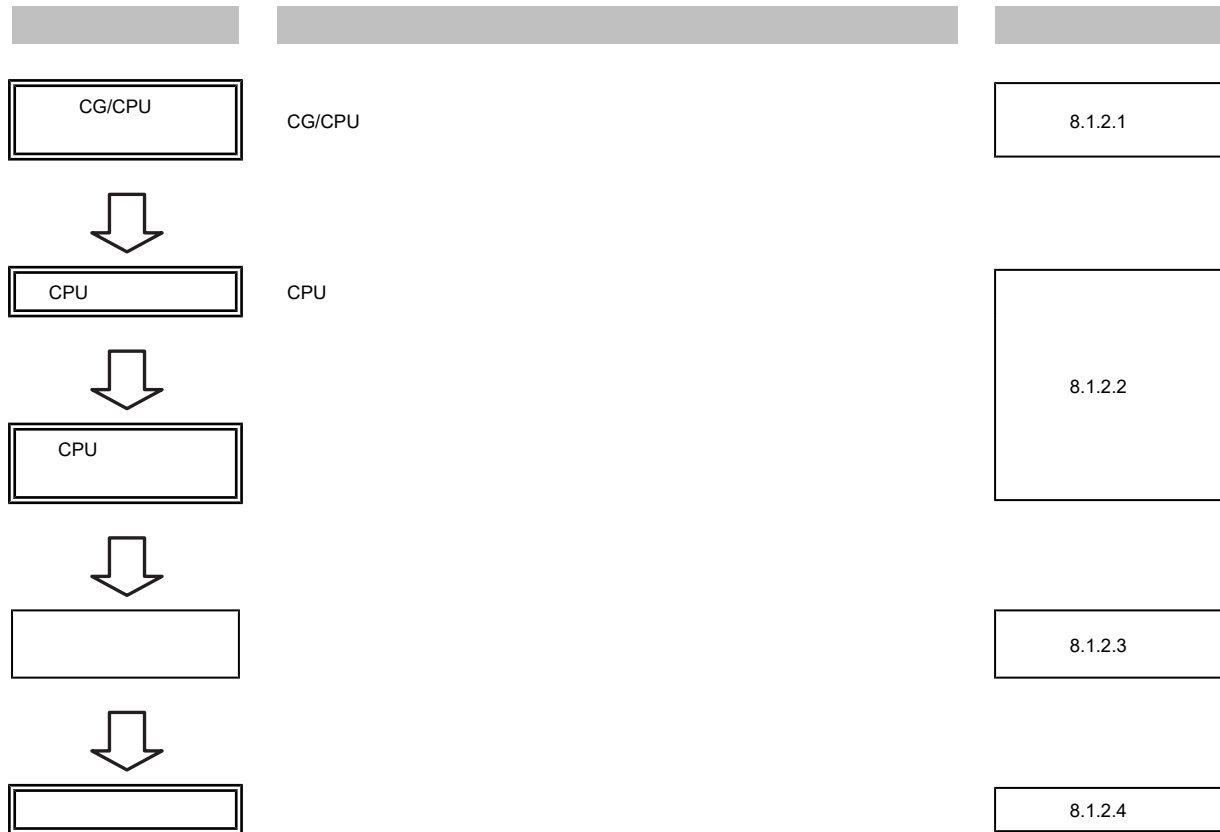
それぞれの例外の詳細な内容は、"ARM ドキュメンテーションセット Cortex-M4F 用"をご覧ください。

- ・ リセット
- ・ マスク不能割り込み(NMI)
- ・ ハードフォールト
- ・ メモリ管理
- ・ バスフォールト
- ・ 用法フォールト
- ・ SVCcall (スーパーバイザコール)
- ・ デバッグモニタ
- ・ PendSV
- ・ SysTick
- ・ 外部割り込み

### 8.1.2

例外／割り込みの処理の流れの概略を以下に示します。以下の説明で、 はハードウェアによる処理を、 はソフトウェアによる処理を示しています。

それぞれの処理の内容について、後続の節で説明します。



8.1.2.1

(1)

例外は、CPU の命令実行、メモリアクセス、外部割り込み端子や周辺機能からの割り込み要求などにより発生します。

CPU の命令実行による例外の要求は、例外が発生する命令の実行や、命令実行中の異常が要因で発生します。

メモリアクセスによる例外の要求は、実行不可領域からの命令フェッチや、フォールト領域へのアクセスにより発生します。

割り込みの要求は、外部割り込み端子からの信号入力や周辺機能から発生します。低消費電力モード解除に使用する割り込みの場合、クロックジェネレータの設定も必要になります。詳細は「8.5 割り込み」の節で説明します。

(2)

複数の例外が同時に検出された場合には、CPU は優先度にしたがって最も優先度の高い例外を選択します。

各例外の優先度は以下のとおりです。"構成可能"と記載された例外は、優先度を設定することができます。また、メモリ管理、バスフォールト、用法フォールトは許可/禁止を選択することができます。

禁止された例外が発生した場合にはハードフォールトとして扱われます。

8-1

1		-3 ( )	, WDT, POR, VLTD, OFD, SYSRESETREQ
2		-2	WDT
3		-1	
4			MPU ( ) ( 1) (XN) (Execute Never)
5			
6			
7~10			
11	SVCcall		SVC
12			CPU
13			
14	PendSV		
15	SysTick		
16~			( 2)

1) MPU

2)

8.5.1.5

(3)

- 優先度レベル

外部割り込みの優先度は、割り込み優先度レジスタで、それ以外の例外はシステムハンドラ優先度レジスタで該当する<PRI\_n>に設定します。

<PRI\_n>は、構成を変更することが可能になっており、優先度設定のためのビット数は製品により3ビット~8ビットのいずれかになります。このため、設定できる優先度の範囲も製品により異なります。

TMPM475FDFG/FZFG/FYFGでは、<PRI\_n>は3ビット構成になっています。

8ビットの構成の場合、優先度は0~255のレベルを設定できます。最も高い優先度は"0"です。複数の要因を同じ優先度に設定した場合、番号の小さい例外の優先度が高くなります。

- 優先度のグループ化

優先度をグループ化することもできます。アプリケーション割り込みおよびリセット制御レジスタの<PRIGROUP>を設定することで、<PRI\_n>を横取り優先度とサブ優先度に分割することができます。

優先度はまず横取り優先度で判定され、横取り優先度が同じ場合サブ優先度で判定されます。サブ優先度も同じ場合は例外番号の小さいほうが優先度が高くなります。

表 8-2 に優先度のグループ化の設定についてまとめます。表中の横取り優先度数、サブ優先度数は、<PRI\_n>が8ビット構成の場合の数です。

8-2

<PRIGROUP[2:0]>	<PRI_n[7:0]>			
000	[7:1]	[0]	128	2
001	[7:2]	[1:0]	64	4
010	[7:3]	[2:0]	32	8
011	[7:4]	[3:0]	16	16
100	[7:5]	[4:0]	8	32
101	[7:6]	[5:0]	4	64
110	[7]	[6:0]	2	128
111		[7:0]	1	256

) <PRI\_n>                    8                    "0"  
                                   3                    <PRI\_n[7:5]>                    <PRI\_n[4:0]> "00000"

## 8.1.2.2

( )

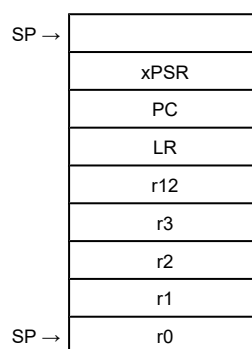
例外により、実行中の処理を中断して割り込み処理ルーチンへ分岐する動作を"横取り"と呼びます。

## (1)

例外を検出すると、CPUは8つのレジスタの内容を退避します。退避するレジスタと退避の順序は以下のとおりです。

1. プログラムカウンタ(PC)
2. プログラムステータスレジスタ(xPSR)
3. r0~r3
4. r12
5. リンクレジスタ(LR)

レジスタの退避が終了すると、SPは8ワード分減らされます。レジスタ退避終了後のスタックの状態は以下のようになっています。



## (2)

レジスタの退避と同時にCPUは割り込み処理ルーチンの命令フェッチを行います。

各例外の割り込み処理ルーチンの先頭番地をベクタテーブルに準備しておきます。ベクタテーブルはリセット後、コード領域の0x0000\_0000番地に置かれます。ベクタテーブルは、ベクタテーブルオフセットレジスタを設定することでコード空間またはSRAM空間の任意のアドレスに置くことができます。

また、ベクタテーブルにはメインスタックの初期値を設定します。

## (3)

割り込み処理ルーチンの実行前に、検出した例外よりも優先度の高い例外を検出した場合、CPUは優先度の高い例外の処理に移行します。これを後着と呼びます。

後着の場合、CPUは新たに検出された例外の割り込み処理ルーチンのフェッチを改めて行い、分岐しますが、再度レジスタの内容を退避することはありません。

(4)

ベクタテーブルの構成は以下のとおりです。

最初の4ワード(スタックの先頭アドレス、リセット、マスク不能割り込み、ハードフォールトの割り込み処理ルーチンアドレス)は必ず設定する必要があります。

その他の例外については、割り込み処理ルーチンのアドレスを必要に応じて準備します。

0x00			
0x04			
0x08			
0x0C			
0x10			
0x14			
0x18			
0x1C ~ 0x28			
0x2C	SVCall		
0x30			
0x34			
0x38	PendSV		
0x3C	SysTick		
0x40			

### 8.1.2.3

割り込み処理ルーチンでは、発生した例外に応じて必要な処理を行います。割り込み処理ルーチンはユーザが準備します。

割り込み処理ルーチンでは、通常の処理プログラムに戻ったときに再度同じ割り込みが発生しないよう、割り込み要求の取り下げなどの処理が必要になる場合があります。

割り込みについての詳細は「8.5 割り込み」の節で説明します。

割り込み処理ルーチンの実行中に現在処理中の例外よりも優先度の高い例外を検出した場合、CPUは現在実行中の割り込み処理ルーチンを中断し新たに検出された例外の処理を行います。

## 8.1.2.4

(1)

割り込み処理ルーチン終了時の状態により復帰先が決まります。

- ・ テールチェーン

保留中の例外が存在し、中断されている例外処理がないかまたは中断されているどの例外よりも優先度が高い場合、保留中の例外の割り込み処理ルーチンへ復帰します。

このとき、スタックの退避と復帰は省略されます。この動作をテールチェーンと呼びます。

- ・ 処理が中断されている割り込み処理ルーチンへ復帰

保留中の例外がない場合、または存在しても処理が中断されている例外の優先度が高い場合、中断されている例外の割り込み処理ルーチンへ復帰します。

- ・ 元のプログラムへ復帰

保留中の例外も処理が中断されている例外もない場合、元のプログラムへ復帰します。

(2)

CPU は、割り込み処理ルーチンから復帰する際に以下の処理を行います。

- ・ レジスタの復帰

退避していた 8 つのレジスタ(xPSR, PC, LR, r12, r3 ~ r0)を復帰し SP を調整します。

- ・ 割り込み番号のロード

退避していた xPSR から現在有効な割り込み番号をロードします。この割り込み番号によって CPU はどの割り込みに復帰するかを制御します。

- ・ SP の選択

例外(ハンドラモード)へ復帰する場合、SP は SP\_main です。スレッドモードへ復帰する場合、SP は SP\_main または SP\_process です。

## 8.2

リセット例外には、以下の要因があります。

リセットの要因を確認するためには、クロックジェネレータレジスタの **CGRSTFLG** を参照してください。

- ・ 外部リセット端子によるリセット例外  
外部リセット端子を"Low"にしたのち、"High"にすることによりリセット例外が発生します。
- ・ POR によるリセット例外  
POR にリセット例外が発生する機能があります。詳細は「パワーオンリセット回路」の章をご覧ください。
- ・ VLTD によるリセット例外  
VLTD にリセットが発生する機能があります。詳細は「電源検出回路」の章をご覧ください。
- ・ OFD によるリセット例外  
OFD にリセットが発生する機能があります。詳細は「周波数検知回路」の章をご覧ください。
- ・ WDT によるリセット例外  
WDT にリセット例外が発生する機能があります。詳細は「ウォッチドッグタイマ」の章をご覧ください。
- ・ <SYSRESETREQ>によるリセット例外  
NVIC レジスタの、アプリケーション割り込みおよびリセット制御レジスタの<SYSRESETREQ>をセットすることで、リセット例外が発生させることができます。

### 8.3 (NMI)

WDT にマスク不能割り込みを発生する機能があります。詳細は「ウォッチドッグタイマ」の章をご覧ください。

### 8.4 SysTick

SysTick は、CPU の持つシステムタイマを使用した割り込み機能です。

SysTick リロード値レジスタに値を設定し、SysTick 制御およびステータスレジスタで機能をイネーブルにすると、リロード値レジスタに設定された値がカウンタへリロードされカウントダウンを開始します。カウンタが"0"になると SysTick 例外を発生します。また、例外を保留しフラグでタイマが"0"になったことを確認することもできます。

) fosc(CGOSCCR<OSCSEL> ) 32

## 8.5

この節では、割り込み要求の伝わる経路、要因、必要な設定について説明します。

割り込みは、割り込み要因ごとに割り込み要求信号により CPU へ通知されます。CPU は、優先順位付けを行い、最も優先度の高い割り込みを発生します。

スタンバイ解除に使用する割り込み要求は、クロックジェネレータを経由して CPU に要因が伝わるため、クロックジェネレータの設定も必要です。

### 8.5.1

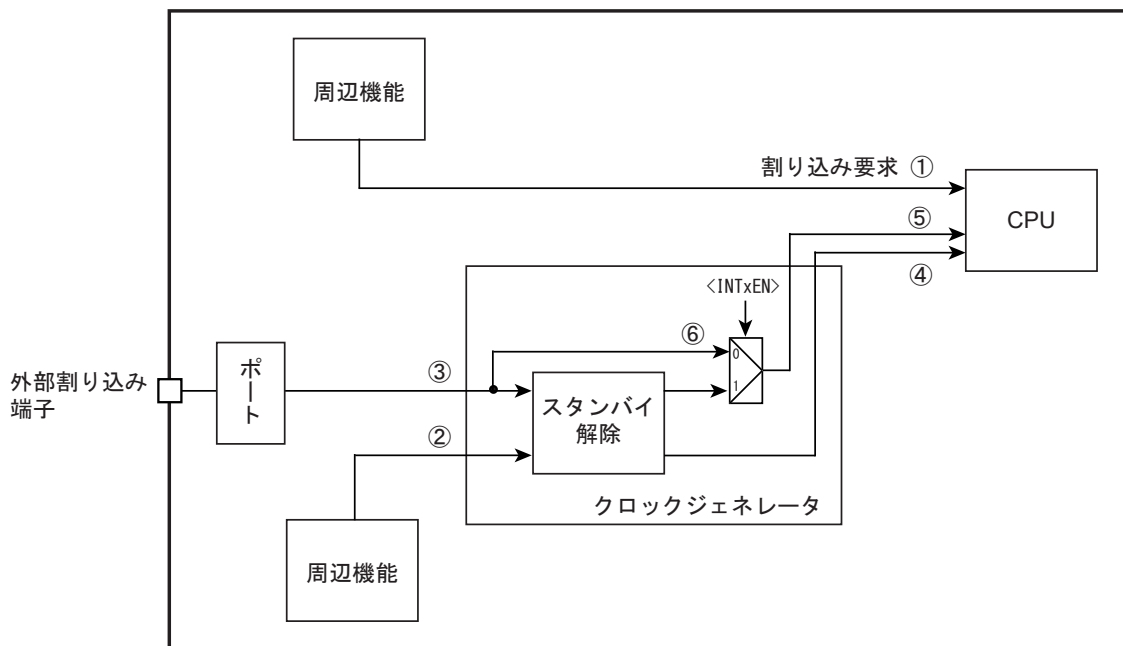
#### 8.5.1.1

割り込み要求の経路を図 8-1 に示します。

周辺機能からの割り込み要求のうち、低消費電力モード解除に使用されないものは直接 CPU に入力されます。(経路 ①)

スタンバイ解除に使用される周辺機能割り込み要求(経路 ②)および、外部割り込み端子からの割り込み(経路 ③)はクロックジェネレータに入力され、スタンバイ解除のロジックを経由して CPU に入力されます。(経路 ④⑤)

外部割り込み端子からの割り込みは、スタンバイ解除に使用しない場合スタンバイ解除ロジックを経由せずに CPU に入力されます。(経路 ⑥)



8-1

### 8.5.1.2

割り込み要求は、割り込み要因に割り当てられた外部割り込み端子、周辺機能、NVIC レジスタの割り込み保留セットレジスタの設定により発生します。

- ・ 外部割り込み端子からの割り込み  
外部割り込み端子を使用する場合、ポートの制御レジスタで端子を割り込み機能に設定します。
- ・ 周辺機能の割り込み  
周辺機能の割り込みを使用する場合、使用する周辺機能で割り込み要求が発生されるよう設定する必要があります。  
設定の詳細については各章を参照ください。
- ・ 割り込み要求の強制的な発生  
NVIC の割り込み保留セットレジスタの該当するビットをセットすることで、割り込み要求を強制的に発生させることができます。

CPU は、割り込み要求の"High"レベルを割り込みとして認識します。

### 8.5.1.3

外部端子/周辺機能から発生した割り込み要求のうち、スタンバイ解除要因にならないものは直接 CPU に接続されます。

スタンバイ解除要因として使用できる割り込みは、クロックジェネレータを経由して CPU に接続されるため、クロックジェネレータの設定が必要です。ただし、外部割り込みについてはスタンバイ解除要因として使用しない場合はクロックジェネレータの設定なしで使用することもできます。この場合、次項の注意事項に留意して使用してください。

### 8.5.1.4

外部割り込みを使用際には、予期しない割り込みが発生しないよう以下の点に留意してください。

外部割り込み端子から入力信号は、入力ディセーブル(PxIE<PxIE> = "0")の場合"High"となります。また、外部割り込みをスタンバイ解除要因として使用しない場合(「図 8-1 割り込みの経路」の ⑥ の経路)、外部割り込み端子からの入力信号がそのまま CPU に伝わります。CPU は"High"入力を割り込みとして認識しますので、入力ディセーブルのまま CPU で該当する割り込みを有効にすると割り込みが発生します。

外部割り込みをスタンバイ解除要因とせずに使用する際には、割り込み端子入力を"Low"レベルとして入力イネーブルにし、その後 CPU で割り込み許可設定を行ってください。

## 8.5.1.5

割り込みの要因一覧を表 8-3 に示します。

8-3

			( )	CG
0	INT0	0	[H]/[L] [↑]/[↓]	CGIMCGA
1	INT1	1		
2	INT2	2		
3	INT3	3	[H]/[L] [↑]/[↓]	CGIMCGB
4	INT4	4		
5	INT5	5		
6	INTRX0	0		
7	INTTX0	0		
8	INTRX1	1		
9	INTTX1	1		
10	INTVCN0	0		
11	INTVCN1	1		
12	INTEMG0	PMD0 EMG		
13	INTEMG1	PMD1 EMG		
14	INTOVV0	PMD0 OVV		
15	INTOVV1	PMD1 OVV		
16	INTADAPDA	ADCA PMD ADCA INTADxPDA		
17	INTADBPDA	ADCB PMD ADCB INTADxPDA		
18	INTADAPDB	ADCA PMD ADCA INTADxPDB		
19	INTADBPDB	ADCB PMD ADCB INTADxPDB		
20	INTTB00	16bit TMRB0 0/		
21	INTTB01	16bit TMRB0 1		
22	INTTB10	16bit TMRB1 0/		
23	INTTB11	16bit TMRB1 1		
24	INTTB40	16bit TMRB4 0/		
25	INTTB41	16bit TMRB4 1		
26	INTTB50	16bit TMRB5 0/		
27	INTTB51	16bit TMRB5 1		
28	INTPMD0	PMD0		
29	INTPMD1	PMD1		
30	INTCAP00	16bit TMRB0 0		
31	INTCAP01	16bit TMRB0 1		
32	INTCAP10	16bit TMRB1 0		
33	INTCAP11	16bit TMRB1 1		
34	INTCAP40	16bit TMRB4 0		
35	INTCAP41	16bit TMRB4 1		
36	INTCAP50	16bit TMRB5 0		
37	INTCAP51	16bit TMRB5 1		
38	INT6	6	[H]/[L] [↑]/[↓]	CGIMCGB
39	INT7	7		
40	INTRX2	2		

8-3

			( )	CG
41	INTTX2	2		
42	INTADACPA	ADCA A		
43	INTADBCPA	ADCB A		
44	INTADACPB	ADCA B		
45	INTADBCPB	ADCB B		
46	INTTB20	16bit TMRB2 0/		
47	INTTB21	16bit TMRB2 1		
48	INTTB30	16bit TMRB3 0/		
49	INTTB31	16bit TMRB3 1		
50	INTCAP20	16bit TMRB2 0		
51	INTCAP21	16bit TMRB2 1		
52	INTCAP30	16bit TMRB3 0		
53	INTCAP31	16bit TMRB3 1		
54	INTADASFT	ADCA		
55	INTADBSFT	ADCB		
56	INTADATMR	ADCA		
57	INTADBTMR	ADCB		
58	INT8	8	[H]/[L] [↑]/[↓]	CGIMCGC
59	INT9	9		
60	INTA	A		
61	INTB	B		
62	INTENC00	0 0		
63	INTENC01	0 1		
64	INTRX3	3		
65	INTTX3	3		
66	INTTB60	16bit TMRB6 0/		
67	INTTB61	16bit TMRB6 1		
68	INTTB70	16bit TMRB7 0/		
69	INTTB71	16bit TMRB7 1		
70	INTCAP60	16bit TMRB6 0		
71	INTCAP61	16bit TMRB6 1		
72	INTCAP70	16bit TMRB7 0		
73	INTCAP71	16bit TMRB7 1		
74	INTC	C	[H]/[L] [↑]/[↓]	CGIMCGD
75	INTD	D	[H]/[L] [↑]/[↓]	CGIMCGD
76	INTE	E		
77	INTF	F		
78	INTVCT0	0		
79	INTVCT1	1		
80	INTSB10	I2C		
81	INTCANRX	CAN		
82	INTCANTX	CAN		
83	INTCANGB	CAN		
84	INTTB80	16bit TMRB8 0/		
85	INTTB81	16bit TMRB8 1		
86	INTTB90	16bit TMRB9 0/		

## 8-3

			( )	CG
87	INTTB91	16bit TMRB9	1	
88	INTCAP80	16bit TMRB8	0	
89	INTCAP81	16bit TMRB8	1	
90	INTCAP90	16bit TMRB9	0	
91	INTCAP91	16bit TMRB9	1	
92	INTDMACATC	DMA		
93	INTDMACAERR	DMA		
94	INTENC10	1	0	
95	INTENC11	1	1	

## 8.5.1.6

アクティブレベルはどのような信号変化を割り込み要因とみなすかを示しています。CPUは割り込み信号の"High"を割り込み要因とみなします。各種周辺機能からCPUへ直接割り込み信号が伝わるものは、割り込み要求として"High"パルスを出力するようになっています。

スタンバイ解除要因となる割り込みについては、クロックジェネレータに設定するアクティブレベルは、周辺機能からの割り込み要求は立ち上がり(「↑」)エッジまたは立ち下がり(「↓」)エッジとなり、割り込み端子からの割り込み要求は"High"レベル、"Low"レベル、立ち上がり(「↑」)エッジ、立ち下がり(「↓」)エッジから選ぶことができます。

スタンバイ解除要因となる割り込みを使用するときにはクロックジェネレータレジスタのCGIMCGx<INT x EN>を有効にし、CGIMCGx<EMCGx>にアクティブレベルを設定します。周辺機能からの割り込み要求のアクティブレベルは表 8-3 で指定されているとおりに設定してください。

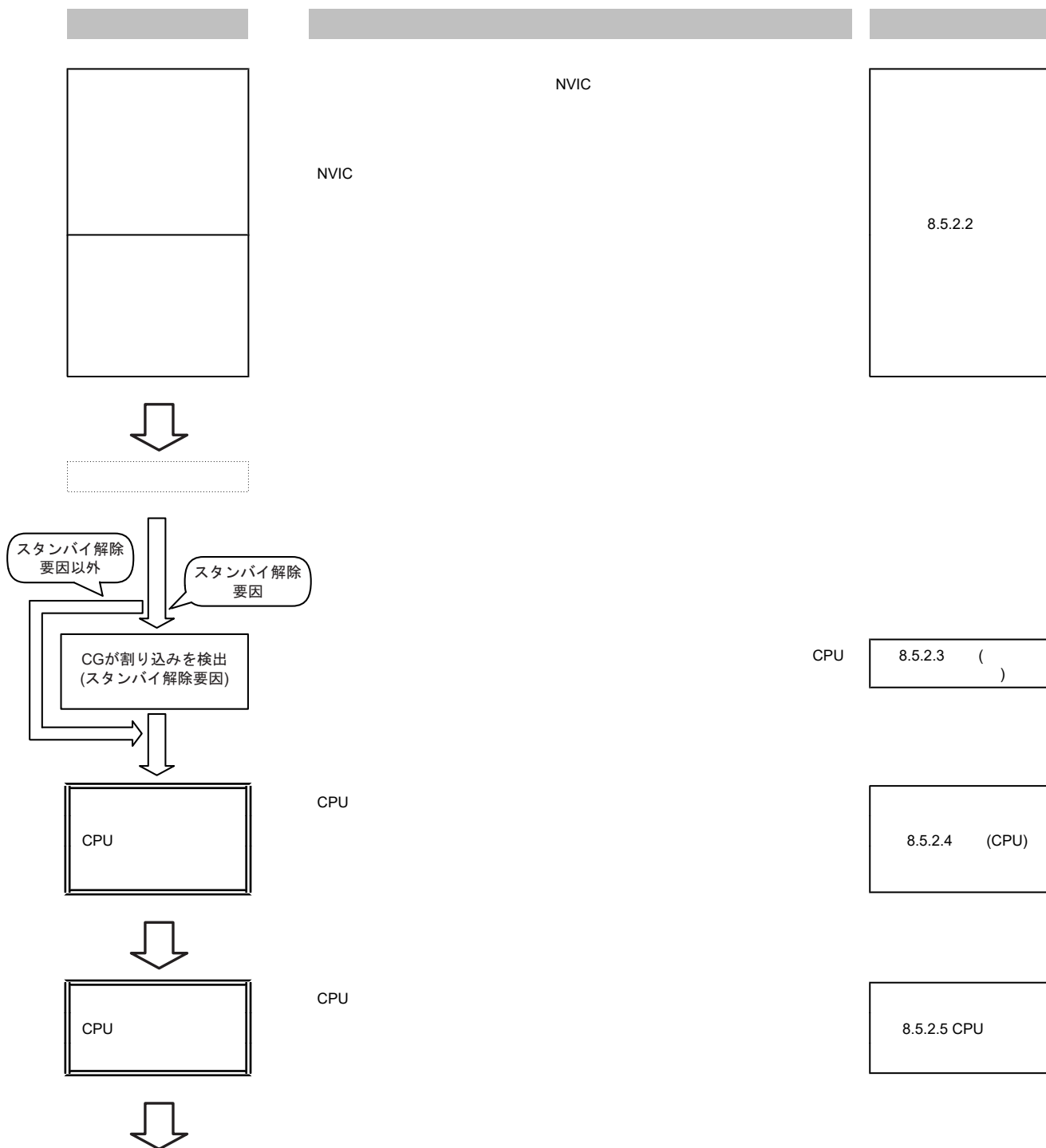
クロックジェネレータで検出された割り込みは、"High"レベル信号でCPUに通知されます。

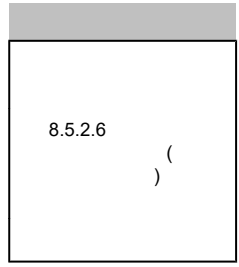
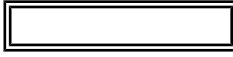
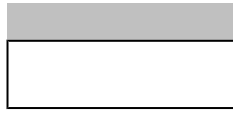
8.5.2

8.5.2.1

割り込みの処理の流れを以下に示します。

以下の説明で、はハードウェアによる処理を、はソフトウェアによる処理を示しています。





## 8.5.2.2

割り込みの準備を行うときには、設定途中で不要な割り込みの発生を防ぐために設定の順番に注意が必要です。

割り込みの使用開始、または設定変更のときの基本的な順序は、まず、CPU で割り込みを禁止し、次に割り込み経路で CPU から遠いところから設定を行い、最後に CPU で割り込みを許可します。

クロックジェネレータの設定を行うときには、条件の設定を行った後、不要な割り込みが発生しないよう、クロックジェネレータ内部の割り込み情報をクリアしてから割り込み許可の設定を行います。

以下に設定の手順と、手順ごとの具体的な設定方法を示します。

1. CPU 割り込み禁止
2. CPU 割り込み設定
3. 要因の準備(1) (外部割り込み端子)
4. 要因の準備(2) (周辺機能からの割り込み)
5. 要因の準備(3) (割り込み保留セットレジスタ)
6. クロックジェネレータの設定
7. CPU 割り込み許可

## (1) CPU

CPU を割り込み禁止状態にするには、PRIMASK レジスタに"1"をセットします。これにより、マスク不能割り込みとハードフォールト例外以外のすべての割り込みと例外がマスクされます。

このレジスタをセットするためには"MSR"命令を使用します。

PRIMASK	"1"(            )

- 1) PRIMASK
- 2) PRIMASK            "1"

## (2) CPU

NVIC レジスタの割り込み優先度 レジスタで<PRI\_n>に優先度の設定を行います。

このレジスタは、8 ビットごとに各割り込み要因に割り当てられていますが、製品ごとに構成するビット数が異なります。8 ビットの構成の場合「0」から「255」までの優先度を設定することができます。最も高い優先度は「0」です。複数の要因を同じ優先度に設定した場合、番号の小さい割り込みの優先度が高くなります。

グループ優先度を設定する場合にはアプリケーション割り込みおよびリセット制御レジスタの<PRIGROUP>も設定します。

NVIC		
<PRL_n>		
<PRIGROUP>		( )

) n / 3

(3) (1) ( )

外部割り込み端子を使用する場合、該当する端子のポートの設定を行います。ポートを入力として使用するために PxIE[m]を"1"に設定します。

PxIE<PxmlE>		"1"
-------------	--	-----

) x m  
PxIE

8.5.1.4

(4) (2) ( )

周辺機能からの割り込みを使用する場合、設定方法は周辺機能によって異なります。各周辺機能の章をご覧ください。

(5) (3) ( )

割り込み保留セットレジスタで割り込みを発生する場合、該当するビットに"1"をセットします。

NVIC	
<SETPEND[m]>	"1"

) m

(6)

スタンバイ解除要因となる割り込みは、クロックジェネレータの CGIMCG レジスタでアクティブレベルと割り込み許可の設定を行います。CGIMCG レジスタは要因ごとの設定レジスタです。

割り込み許可の前に、不要な割り込み発生を防止するため割り込み要求のクリアを CGICRCG レジスタで行います。CGICRCG レジスタは、要因に対応した値を書き込むことで保持されていた割り込み要求をクリアすることができます。具体的な値は、「8.6.3.2 CGICRCG(CG 割り込み要求クリアレジスタ)」を参照してください。

割り込み端子からの割り込み要求をスタンバイ解除要因として使用しない場合、クロックジェネレータの設定を行わずに使用することもできます。ただし、CPU が割り込み要因として検出するためには、"High"パルスまたは"High"レベルの信号を入力する必要があります。また、「8.5.1.4 外部割り込み端子を使用する際の注意」の記載事項に注意してください。

CGIMCGn<EMCGm>		
CGICRCG<ICRCG>		
CGIMCGn<INTmEN>		"1"( )

) n m

## (7) CPU

CPU の割り込み許可の設定をします。

割り込み保留クリアレジスタで保留状態の割り込みをクリアし、割り込みイネーブルセットレジスタで割り込みを許可します。これらのレジスタは1ビットずつ各割り込み要因に割り当てられています。

割り込み保留クリアレジスタの該当する割り込みのビットに"1"を書くことで保留されている要因をクリアすることができ、割り込みイネーブルセットレジスタの該当する割り込みのビットに"1"を書くことで割り込みを許可することができます。

ただし、割り込み保留セットレジスタの設定で割り込みが発生する場合、割り込み保留クリアを行うと割り込み要因そのものが失われるため、この操作は不要です。

最後に、PRIMASK レジスタを"0"にクリアします。

NVIC		
<CLRPEND[m]>		"1"
<SETENA[m]>		"1"
PRIMASK		
		"0"

- 1) m
- 2) PRIMASK

### 8.5.2.3 ( )

スタンバイ解除要因となる割り込みは、クロックジェネレータに設定されたアクティブレベルにしたがって検出され CPU に伝えられます。

アクティブレベルが立ち上がりまたは立ち下がりエッジの割り込み要因は、検出された後、クロックジェネレータで要因が保持されますが、"High"レベルまたは"Low"レベル設定の割り込み要因は、アクティブレベルから変化すると割り込み要因がなくなったとみなされるため、割り込み検出までレベルを保つ必要があります。

クロックジェネレータは割り込みを検出すると CG 割り込み要求クリアレジスタ(CGICRCG)で解除されるまで"High"レベルの割り込み信号を CPU に出力します。解除を行わずに復帰すると再度同じ割り込みが検出されますので、割り込み処理ルーチン内で割り込みの解除を行ってください。

### 8.5.2.4 (CPU)

CPU は優先順位に従って最も優先度の高い割り込み要因を検出します。

### 8.5.2.5 CPU

割り込みが検出されると、CPU はスタックへ xPSR、PC、LR、R12、r3 ~ r0 を退避し、検出した割り込みの割り込み処理ルーチンへ分岐します。

### 8.5.2.6 ( )

割り込み処理ルーチンではアプリケーションにより必要な内容をプログラミングしますが、ここでは推奨する処理と要因の取り下げについて説明します。

#### (1)

通常、割り込み処理ルーチンでは必要なレジスタの退避と割り込み処理を行います。Cortex-M4F コアは自動的に xPSR、PC、LR、R12、r3 ~ r0 をスタックへ退避するため、これらのレジスタをユーザプログラムで退避する必要はありません。

その他のレジスタについては必要に応じて退避します。

割り込み処理ルーチン実行中でも、より高い優先度の割り込みや NMI などの例外は受け付けられます。そのため書き換わる可能性のある、汎用レジスタを退避することを推奨します。

#### (2)

スタンバイ解除要因となる割り込みについては、CGICRCG レジスタで割り込み要求を解除する必要があります。

アクティブレベルがレベル検出の割り込みの場合、要因そのものを取り下げない限り割り込み要求は存在し続けるため、まず要因を取り下げる必要があります。レベル検出の場合は、要因が取り下げられるとクロックジェネレータからの割り込み要求信号は自動的に取り下げられます。

エッジ検出の場合は CGICRCG レジスタに該当する割り込みの値を設定することで要因は取り下げられ、再度有効なエッジが発生したときに改めて要因として認識されます。

8.6 /

以下に、本章で説明した CPU の NVIC レジスタとクロックジェネレータレジスタを示します。

8.6.1

NVIC Base Address = 0xE000\_E000

		Address
SysTick		0x0010
SysTick		0x0014
SysTick		0x0018
SysTick		0x001C
	1	0x0100
	2	0x0104
	3	0x0108
	1	0x0180
	2	0x0184
	3	0x0188
	1	0x0200
	2	0x0204
	3	0x0208
	1	0x0280
	2	0x0284
	3	0x0288
		0x0400 ~ 0x047F
		0x0D08
		0x0D0C
		0x0D18, 0x0D1C, 0x0D20
		0x0D24

CG

		Address
CG	CGICRCG	0x0014
	CGRSTFLG	0x001C
CG	A CGIMCGA	0x0020
CG	B CGIMCGB	0x0024
CG	C CGIMCGC	0x0028
CG	D CGIMCGD	0x002C

## 8.6.2 NVIC

### 8.6.2.1 SysTick

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	COUNTFLAG
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	CLKSOURCE	TICKINT	ENABLE
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-17		R	"0"
16	COUNTFLAG	R/W	0: 0 1: 0 "1" "0"
15-3		R	"0"
2	CLKSOURCE	R/W	0: (fosc/32) ( ) 1: CPU (fsys)
1	TICKINT	R/W	0: SysTick 1: SysTick
0	ENABLE	R/W	0: 1: "1"

)

fosc(CGOSCCR&lt;OSCSEL&gt;

) 32

8.6.2.2 SysTick

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

Bit	Bit Symbol	Type	
31-24		R	"0"
23-0	RELOAD	R/W	"0" SysTick

8.6.2.3 SysTick

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

Bit	Bit Symbol	Type	
31-24		R	"0"
23-0	CURRENT	R/W	[ ] SysTick [ ] SysTick <COUNTFLAG>

8.6.2.4 SysTick

	31	30	29	28	27	26	25	24
bit symbol	NOREF	SKEW	-	-	-	-	-	-
	0	1	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TENMS							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TENMS							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TENMS							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	NOREF	R	0: 1:
30	SKEW	R	0: 10 ms 1: 10 ms
29-24		R	"0"
23-0	TENMS	R	( )

)

8.6.2.5

それぞれの割り込み要因について、割り込みイネーブルセットレジスタ、割り込みイネーブルクリアレジスタ、割り込み保留セットレジスタ、割り込みクリアレジスタがあります。

各ビットが指定された割り込みに対応しています。

(1)

割り込みを許可したり、割り込みの許可/禁止状態が確認できます。

"1"をライトすることで該当する割り込みを許可します。

"0"の書き込みは意味を持ちません。

リードすると該当する割り込みの許可/禁止状態が確認できます。

このレジスタのビットをクリアするには、割り込みイネーブルクリアレジスタの対応するビットに"1"をセットします。

Bit symbol	Type	
SETENA	R/W	[ 95:0 ]
		1: [ ]
		0: [ ]
		1: [ ]

(a)

1

	31	30	29	28	27	26	25	24
bit symbol	SETENA ( 31 )	SETENA ( 30 )	SETENA ( 29 )	SETENA ( 28 )	SETENA ( 27 )	SETENA ( 26 )	SETENA ( 25 )	SETENA ( 24 )
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA ( 23 )	SETENA ( 22 )	SETENA ( 21 )	SETENA ( 20 )	SETENA ( 19 )	SETENA ( 18 )	SETENA ( 17 )	SETENA ( 16 )
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA ( 15 )	SETENA ( 14 )	SETENA ( 13 )	SETENA ( 12 )	SETENA ( 11 )	SETENA ( 10 )	SETENA ( 9 )	SETENA ( 8 )
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA ( 7 )	SETENA ( 6 )	SETENA ( 5 )	SETENA ( 4 )	SETENA ( 3 )	SETENA ( 2 )	SETENA ( 1 )	SETENA ( 0 )
	0	0	0	0	0	0	0	0

)

8.5.1.5

(b)

2

	31	30	29	28	27	26	25	24
bit symbol	SETENA ( 63)	SETENA ( 62)	SETENA ( 61)	SETENA ( 60)	SETENA ( 59)	SETENA ( 58)	SETENA ( 57)	SETENA ( 56)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA ( 55)	SETENA ( 54)	SETENA ( 53)	SETENA ( 52)	SETENA ( 51)	SETENA ( 50)	SETENA ( 49)	SETENA ( 48)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA ( 47)	SETENA ( 46)	SETENA ( 45)	SETENA ( 44)	SETENA ( 43)	SETENA ( 42)	SETENA ( 41)	SETENA ( 40)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA ( 39)	SETENA ( 38)	SETENA ( 37)	SETENA ( 36)	SETENA ( 35)	SETENA ( 34)	SETENA ( 33)	SETENA ( 32)
	0	0	0	0	0	0	0	0

)

8.5.1.5

(c)

3

	31	30	29	28	27	26	25	24
bit symbol	SETENA ( 95)	SETENA ( 94)	SETENA ( 93)	SETENA ( 92)	SETENA ( 91)	SETENA ( 90)	SETENA ( 89)	SETENA ( 88)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	SETENA ( 87)	SETENA ( 86)	SETENA ( 85)	SETENA ( 84)	SETENA ( 83)	SETENA ( 82)	SETENA ( 81)	SETENA ( 80)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SETENA ( 79)	SETENA ( 78)	SETENA ( 77)	SETENA ( 76)	SETENA ( 75)	SETENA ( 74)	SETENA ( 73)	SETENA ( 72)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SETENA ( 71)	SETENA ( 70)	SETENA ( 69)	SETENA ( 68)	SETENA ( 67)	SETENA ( 66)	SETENA ( 65)	SETENA ( 64)
	0	0	0	0	0	0	0	0

)

8.5.1.5

(2)

割り込みを禁止したり、割り込みの許可/禁止状態が確認できます。

"1"をライトすることで該当する割り込みを禁止します。

"0"の書き込みは意味を持ちません。

リードすると該当する割り込みの許可/禁止状態が確認できます。

Bit symbol	Type	
CLRENA	R/W	[95:0]
		[ ]
		1: [ ]
		0: [ ]
		1:

(a)

1

	31	30	29	28	27	26	25	24
bit symbol	CLRENA ( 31)	CLRENA ( 30)	CLRENA ( 29)	CLRENA ( 28)	CLRENA ( 27)	CLRENA ( 26)	CLRENA ( 25)	CLRENA ( 24)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA ( 23)	CLRENA ( 22)	CLRENA ( 21)	CLRENA ( 20)	CLRENA ( 19)	CLRENA ( 18)	CLRENA ( 17)	CLRENA ( 16)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA ( 15)	CLRENA ( 14)	CLRENA ( 13)	CLRENA ( 12)	CLRENA ( 11)	CLRENA ( 10)	CLRENA ( 9)	CLRENA ( 8)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA ( 7)	CLRENA ( 6)	CLRENA ( 5)	CLRENA ( 4)	CLRENA ( 3)	CLRENA ( 2)	CLRENA ( 1)	CLRENA ( 0)
	0	0	0	0	0	0	0	0

)

8.5.1.5

(b)

2

	31	30	29	28	27	26	25	24
bit symbol	CLRENA ( 63)	CLRENA ( 62)	CLRENA ( 61)	CLRENA ( 60)	CLRENA ( 59)	CLRENA ( 58)	CLRENA ( 57)	CLRENA ( 56)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA ( 55)	CLRENA ( 54)	CLRENA ( 53)	CLRENA ( 52)	CLRENA ( 51)	CLRENA ( 50)	CLRENA ( 49)	CLRENA ( 48)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA ( 47)	CLRENA ( 46)	CLRENA ( 45)	CLRENA ( 44)	CLRENA ( 43)	CLRENA ( 42)	CLRENA ( 41)	CLRENA ( 40)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA ( 39)	CLRENA ( 38)	CLRENA ( 37)	CLRENA ( 36)	CLRENA ( 35)	CLRENA ( 34)	CLRENA ( 33)	CLRENA ( 32)
	0	0	0	0	0	0	0	0

)

8.5.1.5

(c)

3

	31	30	29	28	27	26	25	24
bit symbol	CLRENA ( 95)	CLRENA ( 94)	CLRENA ( 93)	CLRENA ( 92)	CLRENA ( 91)	CLRENA ( 90)	CLRENA ( 89)	CLRENA ( 88)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CLRENA ( 87)	CLRENA ( 86)	CLRENA ( 85)	CLRENA ( 84)	CLRENA ( 83)	CLRENA ( 82)	CLRENA ( 81)	CLRENA ( 80)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CLRENA ( 79)	CLRENA ( 78)	CLRENA ( 77)	CLRENA ( 76)	CLRENA ( 75)	CLRENA ( 74)	CLRENA ( 73)	CLRENA ( 72)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLRENA ( 71)	CLRENA ( 70)	CLRENA ( 69)	CLRENA ( 68)	CLRENA ( 67)	CLRENA ( 66)	CLRENA ( 65)	CLRENA ( 64)
	0	0	0	0	0	0	0	0

)

8.5.1.5

(3)

割り込みを強制的に保留したり、保留されているかどうかを確認できます。

"1"をライトすることで該当する割り込みを保留します。ただし、すでに保留されている割り込みおよび禁止されている割り込みに対しては無効です。

"0"の書き込みは意味を持ちません。

リードの場合、該当する割り込みが保留されているかどうかを示します。

このレジスタのビットをクリアするには、割り込み保留クリアレジスタの対応するビットに"1"をセットします。

Bit symbol	Type	
SETPEND	R/W	[95:0]
		[     ]
		1:
		[     ]
		0:
		1:

(a)

1

	31	30	29	28	27	26	25	24
bit symbol	SETPEND (     31)	SETPEND (     30)	SETPEND (     29)	SETPEND (     28)	SETPEND (     27)	SETPEND (     26)	SETPEND (     25)	SETPEND (     24)
	23	22	21	20	19	18	17	16
bit symbol	SETPEND (     23)	SETPEND (     22)	SETPEND (     21)	SETPEND (     20)	SETPEND (     19)	SETPEND (     18)	SETPEND (     17)	SETPEND (     16)
	15	14	13	12	11	10	9	8
bit symbol	SETPEND (     15)	SETPEND (     14)	SETPEND (     13)	SETPEND (     12)	SETPEND (     11)	SETPEND (     10)	SETPEND (     9)	SETPEND (     8)
	7	6	5	4	3	2	1	0
bit symbol	SETPEND (     7)	SETPEND (     6)	SETPEND (     5)	SETPEND (     4)	SETPEND (     3)	SETPEND (     2)	SETPEND (     1)	SETPEND (     0)

)

8.5.1.5

(b)

2

	31	30	29	28	27	26	25	24
bit symbol	SETPEND ( 63)	SETPEND ( 62)	SETPEND ( 61)	SETPEND ( 60)	SETPEND ( 59)	SETPEND ( 58)	SETPEND ( 57)	SETPEND ( 56)
	23	22	21	20	19	18	17	16
bit symbol	SETPEND ( 55)	SETPEND ( 54)	SETPEND ( 53)	SETPEND ( 52)	SETPEND ( 51)	SETPEND ( 50)	SETPEND ( 49)	SETPEND ( 48)
	15	14	13	12	11	10	9	8
bit symbol	SETPEND ( 47)	SETPEND ( 46)	SETPEND ( 45)	SETPEND ( 44)	SETPEND ( 43)	SETPEND ( 42)	SETPEND ( 41)	SETPEND ( 40)
	7	6	5	4	3	2	1	0
bit symbol	SETPEND ( 39)	SETPEND ( 38)	SETPEND ( 37)	SETPEND ( 36)	SETPEND ( 35)	SETPEND ( 34)	SETPEND ( 33)	SETPEND ( 32)

)

8.5.1.5

(c)

3

	31	30	29	28	27	26	25	24
bit symbol	SETPEND ( 95)	SETPEND ( 94)	SETPEND ( 93)	SETPEND ( 92)	SETPEND ( 91)	SETPEND ( 90)	SETPEND ( 89)	SETPEND ( 88)
	23	22	21	20	19	18	17	16
bit symbol	SETPEND ( 87)	SETPEND ( 86)	SETPEND ( 85)	SETPEND ( 84)	SETPEND ( 83)	SETPEND ( 82)	SETPEND ( 81)	SETPEND ( 80)
	15	14	13	12	11	10	9	8
bit symbol	SETPEND ( 79)	SETPEND ( 78)	SETPEND ( 77)	SETPEND ( 76)	SETPEND ( 75)	SETPEND ( 74)	SETPEND ( 73)	SETPEND ( 72)
	7	6	5	4	3	2	1	0
bit symbol	SETPEND ( 71)	SETPEND ( 70)	SETPEND ( 69)	SETPEND ( 68)	SETPEND ( 67)	SETPEND ( 66)	SETPEND ( 65)	SETPEND ( 64)

)

8.5.1.5

(4)

保留された割り込みをクリアしたり、保留されているかどうかを確認できます。

"1"をライトすることで該当する保留された割り込みをクリアします。ただし、すでに処理が開始されている割り込みに対しては無効です。

"0"の書き込みは意味を持ちません。

リードの場合、該当する割り込みが保留されているかどうかを示します。

Bit symbol	Type	
CLRPEND	R/W	[95:0]
		[     ]
		1: [     ]
		0: [     ]
		1: [     ]

(a)

1

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND ( 31)	CLRPEND ( 30)	CLRPEND ( 29)	CLRPEND ( 28)	CLRPEND ( 27)	CLRPEND ( 26)	CLRPEND ( 25)	CLRPEND ( 24)
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND ( 23)	CLRPEND ( 22)	CLRPEND ( 21)	CLRPEND ( 20)	CLRPEND ( 19)	CLRPEND ( 18)	CLRPEND ( 17)	CLRPEND ( 16)
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND ( 15)	CLRPEND ( 14)	CLRPEND ( 13)	CLRPEND ( 12)	CLRPEND ( 11)	CLRPEND ( 10)	CLRPEND ( 9)	CLRPEND ( 8)
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND ( 7)	CLRPEND ( 6)	CLRPEND ( 5)	CLRPEND ( 4)	CLRPEND ( 3)	CLRPEND ( 2)	CLRPEND ( 1)	CLRPEND ( 0)

)

8.5.1.5

(b)

2

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND ( 63)	CLRPEND ( 62)	CLRPEND ( 61)	CLRPEND ( 60)	CLRPEND ( 59)	CLRPEND ( 58)	CLRPEND ( 57)	CLRPEND ( 56)
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND ( 55)	CLRPEND ( 54)	CLRPEND ( 53)	CLRPEND ( 52)	CLRPEND ( 51)	CLRPEND ( 50)	CLRPEND ( 49)	CLRPEND ( 48)
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND ( 47)	CLRPEND ( 46)	CLRPEND ( 45)	CLRPEND ( 44)	CLRPEND ( 43)	CLRPEND ( 42)	CLRPEND ( 41)	CLRPEND ( 40)
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND ( 39)	CLRPEND ( 38)	-CLRPEND ( 37)	CLRPEND ( 36)	CLRPEND ( 35)	CLRPEND ( 34)	CLRPEND ( 33)	CLRPEND ( 32)

)

8.5.1.5

(c)

3

	31	30	29	28	27	26	25	24
bit symbol	CLRPEND ( 95)	CLRPEND ( 94)	CLRPEND ( 93)	CLRPEND ( 92)	CLRPEND ( 91)	CLRPEND ( 90)	CLRPEND ( 89)	CLRPEND ( 88)
	23	22	21	20	19	18	17	16
bit symbol	CLRPEND ( 87)	CLRPEND ( 86)	CLRPEND ( 85)	CLRPEND ( 84)	CLRPEND ( 83)	CLRPEND ( 82)	CLRPEND ( 81)	CLRPEND ( 80)
	15	14	13	12	11	10	9	8
bit symbol	CLRPEND ( 79)	CLRPEND ( 78)	CLRPEND ( 77)	CLRPEND ( 76)	CLRPEND ( 75)	CLRPEND ( 74)	CLRPEND ( 73)	CLRPEND ( 72)
	7	6	5	4	3	2	1	0
bit symbol	CLRPEND ( 71)	CLRPEND ( 70)	CLRPEND ( 69)	CLRPEND ( 68)	CLRPEND ( 67)	CLRPEND ( 66)	CLRPEND ( 65)	CLRPEND ( 64)

)

8.5.1.5

## 8.6.2.6

割り込み優先度レジスタは、各割り込みに対し 8 ビットごとの構成になっています。

割り込み番号と対応する割り込み優先度レジスタのアドレスは以下のとおりです。

	31	24 23	16 15	8 7	0
0xE000_E400	PRI_3	PRI_2	PRI_1	PRI_0	
0xE000_E404	PRI_7	PRI_6	PRI_5	PRI_4	
0xE000_E408	PRI_11	PRI_10	PRI_9	PRI_8	
0xE000_E40C	PRI_15	PRI_14	PRI_13	PRI_12	
0xE000_E410	PRI_19	PRI_18	PRI_17	PRI_16	
0xE000_E414	PRI_23	PRI_22	PRI_21	PRI_20	
0xE000_E418	PRI_27	PRI_26	PRI_25	PRI_24	
0xE000_E41C	PRI_31	PRI_30	PRI_29	PRI_28	
0xE000_E420	PRI_35	PRI_34	PRI_33	PRI_32	
0xE000_E424	PRI_39	PRI_38	PRI_37	PRI_36	
0xE000_E428	PRI_43	PRI_42	PRI_41	PRI_40	
0xE000_E42C	PRI_47	PRI_46	PRI_45	PRI_44	
0xE000_E430	PRI_51	PRI_50	PRI_49	PRI_48	
0xE000_E434	PRI_55	PRI_54	PRI_53	PRI_52	
0xE000_E438	PRI_59	PRI_58	PRI_57	PRI_56	
0xE000_E43C	PRI_63	PRI_62	PRI_61	PRI_60	
0xE000_E440	PRI_67	PRI_66	PRI_65	PRI_64	
0xE000_E444	PRI_71	PRI_70	PRI_69	PRI_68	
0xE000_E448	PRI_75	PRI_74	PRI_73	PRI_72	
0xE000_E44C	PRI_79	PRI_78	PRI_77	PRI_76	
0xE000_E450	PRI_83	PRI_82	PRI_81	PRI_80	
0xE000_E454	PRI_87	PRI_86	PRI_85	PRI_84	
0xE000_E458	PRI_91	PRI_90	PRI_89	PRI_88	
0xE000_E45C	PRI_95	PRI_94	PRI_93	PRI_92	

各割り込みに割り当てられている 8 ビットのうち何ビットを優先度の設定に使用できるかは製品により異なります。本製品では、3 ビットで優先度を設定することができます。

以下に、代表として割り込み番号 0~3 の割り込み優先度レジスタの構成を示します。未使用のビットはリードすると"0"が読め、ライトは無視されます。

	31	30	29	28	27	26	25	24
bit symbol	PRI_3							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PRI_2							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PRI_1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PRI_0							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-29	PRI_3	R/W	3
28-24		R	"0"
23-21	PRI_2	R/W	2
20-16		R	"0"
15-13	PRI_1	R/W	1
12-8		R	"0"
7-5	PRI_0	R/W	0
4-0		R	"0"

8.6.2.7

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

Bit	Bit Symbol	Type	
31-7	TBLOFF	R/W	0x0000_0000 32 16 2
6-0		R	"0"

## 8.6.2.8

	31	30	29	28	27	26	25	24
bit symbol	VECTKEY/VECTKEYSTAT							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	VECTKEY/VECTKEYSTAT							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENDIANESS	-	-	-	-	PRIGROUP		
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SYSRESET REQ	VECTCLR ACTIVE	VECTRESET
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	VECTKEY ( )/ VECTKEYSTAT ( )	R/W	[ ] [ ]      <VECTKEY> "0x05FA" "0xFA05"
15	ENDIANESS	R/W	( 1) 1: 0:
14-11		R	"0"
10-8	PRIGROUP	R/W	000:          7bit          1bit 001:          6bit          2bit 010:          5bit          3bit 011:          4bit          4bit 100:          3bit          5bit 101:          2bit          6bit 110:          1bit          7bit 111:          0bit          8bit <PRI_n>
7-3		R	"0"
2	SYSRESET REQ	R/W	"1"          CPU    SYSRESETREQ          ( 2)
1	VECTCLR ACTIVE	R/W	1:          NMI 0:
0	VECTRESET	R/W	1: 0: "1"          (FPB,DWT,ITM)    CPU

- 1)  
2)                      SYSRESETREQ  
                         <SYSRESETREQ>

8.6.2.9

システムハンドラ優先度レジスタは、各例外に対し 8 ビットごとの構成になっています。  
 例外と対応する割り込み優先度レジスタのアドレスは以下のとおりです。

	31	24 23	16 15	8 7	0
0xE000_ED18	PRI_7		PRI_6 (            )	PRI_5 (            )	PRI_4 (            )
0xE000_ED1C	PRI_11 (SVCall)		PRI_10	PRI_9	PRI_8
0xE000_ED20	PRI_15 (SysTick)		PRI_14 (PendSV)	PRI_13	PRI_12 (            )

各割り込みに割り当てられている 8 ビットのうち何ビットを優先度の設定に使用できるかは製品により異なります。本製品では、3 ビットで優先度を設定することができます。

以下に、代表として割り込み番号 4~7 の割り込み優先度レジスタの構成を示します。未使用のビットはリードすると"0"が読め、ライトは無視されます。

	31	30	29	28	27	26	25	24
bit symbol	PRI_7			-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	PRI_6			-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PRI_5			-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PRI_4			-	-	-	-	-
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-29	PRI_7	R/W	
28-24		R	"0"
23-21	PRI_6	R/W	
20-16		R	"0"
15-13	PRI_5	R/W	
12-8		R	"0"
7-5	PRI_4	R/W	
4-0		R	"0"

## 8.6.2.10

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	USGFAULT ENA	BUSFAULT ENA	MEMFAULT ENA
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SVCALL PENDEDED	BUSFAULT PENDEDED	MEMFAULT PENDEDED	USGFAULT PENDEDED	SYSTICKACT	PENDSVACT	-	MONITOR ACT
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SVCALLACT	-	-	-	USGFAULT ACT	-	BUSFAULT ACT	MEMFAULT ACT
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-19		R	"0"
18	USGFAULT ENA	R/W	0: 1:
17	BUSFAUL TENA	R/W	0: 1:
16	MEMFAULT ENA	R/W	0: 1:
15	SVCALL PENDEDED	R/W	SVCall 0: 1:
14	BUSFAULT PENDEDED	R/W	0: 1:
13	MEMFAULT PENDEDED	R/W	0: 1:
12	USGFAULT PENDEDED	R/W	0: 1:
11	SYSTICKACT	R/W	SysTick 0: 1:
10	PENDSVACT	R/W	PendSV 0: 1:
9		R	"0"
8	MONITORACT	R/W	0: 1:

Bit	Bit Symbol	Type	
7	SVCALLACT	R/W	SVCall 0: 1:
6-4		R	"0"
3	USGFAULT ACT	R/W	0: 1:
2		R	"0"
1	BUSFAULT ACT	R/W	0: 1:
0	MEMFAULT ACT	R/W	0: 1:

)

### 8.6.3

#### 8.6.3.1 CG

CG 割り込みモードコントロールレジスタは低消費電力モード解除に使用する割り込み要因のアクティブレベル設定、検出されたアクティブレベル、低消費電力モード解除の許可/禁止をおこないます。

Bit symbol	Type	
EMCGx[2:0]	R/W	( 8-4 000: "Low" 001: "High" 010: 011: 100: : )
EMSTx[1:0]	R	(EMCGx[2:0]="100" ) 00: - 01: 10: 11:
INTxEN	R/W	0: 1:

8-4

			"Low"	"HIGH"			
INT0	0	CGIMCGA <EMCG00[2:0]>	o	o	o	o	o
INT1	1	CGIMCGA <EMCG01[2:0]>	o	o	o	o	o
INT2	2	CGIMCGA <EMCG02[2:0]>	o	o	o	o	o
INT3	3	CGIMCGA <EMCG03[2:0]>	o	o	o	o	o
INT4	4	CGIMCGB <EMCG04[2:0]>	o	o	o	o	o
INT5	5	CGIMCGB <EMCG05[2:0]>	o	o	o	o	o
INT6	6	CGIMCGB <EMCG06[2:0]>	o	o	o	o	o
INT7	7	CGIMCGB <EMCG07[2:0]>	o	o	o	o	o
INT8	8	CGIMCGC <EMCG08[2:0]>	o	o	o	o	o
INT9	9	CGIMCGC <EMCG09[2:0]>	o	o	o	o	o
INTA	A	CGIMCGC <EMCG0A[2:0]>	o	o	o	o	o
INTB	B	CGIMCGC <EMCG0B[2:0]>	o	o	o	o	o
INTC	C	CGIMCGD <EMCG0C[2:0]>	o	o	o	o	o
INTD	D	CGIMCGD <EMCG0D[2:0]>	o	o	o	o	o
INTE	E	CGIMCGD <EMCG0E[2:0]>	o	o	o	o	o
INTF	F	CGIMCGD <EMCG0F[2:0]>	o	o	o	o	o

)

"o"

"x"

(1) CGIMCGA(CG

A)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG03			EMST03		-	INT03EN
	0	0	1	0	0	0		0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG02			EMST02		-	INT02EN
	0	0	1	0	0	0		0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG01			EMST01		-	INT01EN
	0	0	1	0	0	0		0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG00			EMST00		-	INT00EN
	0	0	1	0	0	0		0

- 1) <EMCGx[2:0]> 8-4
- 2) <EMSTx> <EMCGx[2:0]> "100"  
                  <EMSTx>  
                                  CGICRCG <EMSTx>
- 3) <INTxEN> <INTxEN>
- 4) 31 23 15 7 "0"
- 5) 25 17 9 1

(2) CGIMCGB(CG

B)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG07			EMST07		-	INT07EN
	0	0	1	0	0	0		0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG06			EMST06		-	INT06EN
	0	0	1	0	0	0		0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG05			EMST05		-	INT05EN
	0	0	1	0	0	0		0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG04			EMST04		-	INT04EN
	0	0	1	0	0	0		0

- 1) <EMCGx[2:0]> 8-4
- 2) <EMSTx> <EMCGx[2:0]> "100"  
                  <EMSTx>  
                                  CGICRCG <EMSTx>
- 3) <INTxEN> <INTxEN>
- 4) 31 23 15 7 "0"
- 5) 25 17 9 1

(3) CGIMCGC(CG

C)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG0B			EMST0B		-	INT0BEN
	0	0	1	0	0	0		0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG0A			EMST0A		-	INT0AEN
	0	0	1	0	0	0		0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG09			EMST09		-	INT09EN
	0	0	1	0	0	0		0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG08			EMST08		-	INT08EN
	0	0	1	0	0	0		0

- 1) <EMCGx[2:0]> 8-4
- 2) <EMSTx> <EMCGx[2:0]> "100"  
     <EMSTx> CGICRCG <EMSTx>  
     CGICRCG <EMSTx>
- 3) <INTxEN> <INTxEN>
- 4) 31 23 15 7 "0"
- 5) 25 17 9 1

(4) CGIMCGD(CG

D)

	31	30	29	28	27	26	25	24
bit symbol	-	EMCG0F			EMST0F		-	INT0FEN
	0	0	1	0	0	0		0
	23	22	21	20	19	18	17	16
bit symbol	-	EMCG0E			EMST0E		-	INT0EEN
	0	0	1	0	0	0		0
	15	14	13	12	11	10	9	8
bit symbol	-	EMCG0D			EMST0D		-	INT0DEN
	0	0	1	0	0	0		0
	7	6	5	4	3	2	1	0
bit symbol	-	EMCG0C			EMST0C		-	INT0CEN
	0	0	1	0	0	0		0

- 1) <EMCGx[2:0]> 8-4
- 2) <EMSTx> <EMCGx[2:0]> "100"  
     <EMSTx> CGICRCG <EMSTx>  
     CGICRCG <EMSTx>
- 3) <INTxEN> <INTxEN>
- 4) 31 23 15 7 "0"
- 5) 25 17 9 1

8.6.3.2 CGICRCG(CG )

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

Bit	Bit Symbol	Type	
31-5		R	"0"
4-0	ICRCG[4:0]	W	0_0000: INT0      0_1000: INT8 0_0001: INT1      0_1001: INT9 0_0010: INT2      0_1010: INTA 0_0011: INT3      0_1011: INTB 0_0100: INT4      0_1100 : INTC 0_0101: INT5      0_1101 : INTD 0_0110: INT6      0_1110 : INTE 0_0111: INT7      0_1111 : INTF 1_0000 to 1_1111: "0"

8.6.3.3 CGRSTFLG( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	OFDRSTF	DBGRSTF	VLDRSTF	WDRSTF	PINRSTF	PINRSTF
	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	
31-6		R	"0"
5	OFDRSTF	R/W	OFD 0: 0 1: OFD
4	DBGRSTF	R/W	( 1) 0: 0 1: <SYSRESETREQ>
3	VLDRSTF	R/W	VLTD 0: 0 1: VLTD
2	WDRSTF	R/W	WDT 0: 0 1: WDT
1	PINRSTF	R/W	$\overline{\text{RESET}}$ 0: 0 1: $\overline{\text{RESET}}$
0	PORRSTF	R/W	( 2) 0: 0 1:

1) CPU NVIC <SYSRESETREQ>

2) <PONRSTF> 2 "0"  
<PONRSTF>

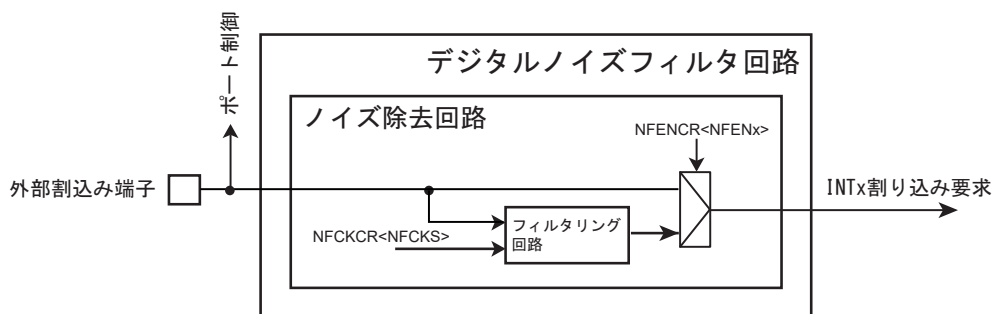


9

(DNF)

デジタル式のノイズキャンセラ回路により、外部割り込み端子に入力される信号を所定の幅でノイズを除去することができます。

9.1



9-1

## 9.2

## 9.2.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)
	NFCKCR	0x0000
	NFENCR	0x0004

## 9.2.1.1 NFCKCR( )

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

Bit	Bit Symbol	Type	
31-3		R	"0"
2-0	NFCKS[2:0]	R/W	000: 001: fsys/2 010: fsys/4 011: fsys/8 100: fsys/16 101: fsys/32 110: fsys/64 111: fsys/128

- 1) NFCKCR<NFCKS>                      NFENCR<NFEN[15:0]>                      "0"
- 2) STOP    NFENCR    NFCKCR

9.2.1.2 NFENCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	NFEN15	NFEN14	NFEN13	NFEN12	NFEN11	NFEN10	NFEN9	NFEN8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	NFEN7	NFEN6	NFEN5	NFEN4	NFEN3	NFEN2	NFEN1	NFEN0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15	NFEN15	R/W	INTF 0: STOP 1:
14	NFEN14	R/W	INTE 0: STOP 1:
13	NFEN13	R/W	INTD 0: STOP 1:
12	NFEN12	R/W	INTC 0: STOP 1:
11	NFEN11	R/W	INTB 0: STOP 1:
10	NFEN10	R/W	INTA 0: STOP 1:
9	NFEN9	R/W	INT9 0: STOP 1:
8	NFEN8	R/W	INT8 0: STOP 1:
7	NFEN7	R/W	INT7 0: STOP 1:
6	NFEN6	R/W	INT6 0: STOP 1:
5	NFEN5	R/W	INT5 0: STOP 1:

Bit	Bit Symbol	Type	
4	NFEN4	R/W	INT4 0: 1: STOP
3	NFEN3	R/W	INT3 0: 1: STOP
2	NFEN2	R/W	INT2 0: 1: STOP
1	NFEN1	R/W	INT1 0: 1: STOP
0	NFEN0	R/W	INT0 0: 1: STOP

1) fsys

fsys

2)

NFENCR

3) STOP  
NFENCR

NFCKCR

## 9.3

### 9.3.1

ノイズフィルタ回路は、ノイズ除去回路、割り込み要求信号発生回路から構成されています。

外部から入力された信号はノイズ除去回路によって High レベルまたは Low レベルのノイズを除去した後、各外部割り込みごとに CG で立ち上がり/立ち下がり/レベル検出を行いません。

### 9.3.2

外部割り込み信号 INTx の High レベル/Low レベル入力ともにノイズを除去します。

ノイズ除去時間は、レベル入力継続時間が NFCKCR<NFCKS>で設定したクロック周期の7クロック以下をノイズと判定し、継続時間がクロック周期の8クロック以上で有効な信号と判定します。

ただし、7、8クロックの間に入力信号はエッジタイミングによって判定が異なる場合があります。

### 9.3.3

ノイズフィルタ回路は NORMAL モード、IDLE モードのときのみ使用可能です。

### 9.3.4 STOP

STOP モードを使用する場合、fsys クロックが停止するためノイズフィルタ回路は使用できません。STOP モード解除に外部割り込みを使用する場合は割り込み許可ビットを禁止した後に NFENCR レジスタのノイズフィルタ許可/禁止ビットを禁止に設定し、NFCKCR レジスタのノイズフィルタクロックを停止させてください。

### 9.3.5

ノイズ除去回路は NFCKCR レジスタで設定したクロック周期の8クロック以上の期間、High レベル入力または Low レベル入力が入力されていた場合、レベル入力を判定し外部割り込み信号を発生します。

9-1

NFCKCR<NFCKS>	fsys [MHz]				Unit
	20	40	80	100	
001	0.7	0.35	0.175	0.14	μs
010	1.4	0.7	0.35	0.28	
011	2.8	1.4	0.7	0.56	
100	5.6	2.8	1.4	1.12	
101	11.2	5.6	2.8	2.24	
110	22.4	11.2	5.6	4.48	
111	44.8	22.4	11.2	8.96	

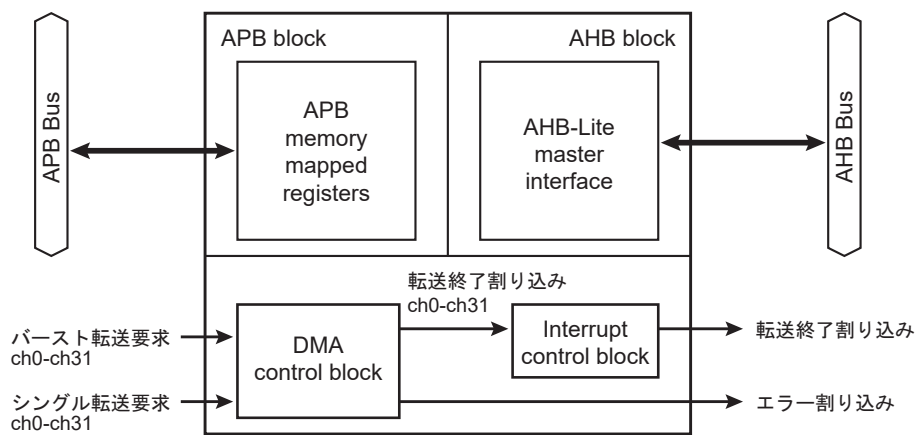




## 10.2

μDMA コントローラは以下の機能ブロックを内蔵しています。

- ・ APB ブロック  
制御レジスタへのアクセスを制御します。
- ・ AHB ブロック  
DMA 転送のバスサイクルを制御します。
- ・ DMA 制御ブロック  
DMA 動作全体の制御を行います。
- ・ 割り込み制御ブロック  
割り込み信号をまとめ、フラグレジスタを設定します。



10-1 μDMA ( )

## 10.3

## 10.3.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照下さい。

:DMA

		Address(Base+)
DMA status Register	DMAxStatus	0x0000
DMA configuration Register	DMAxCfg	0x0004
channel control data base pointer Register	DMAxCtrlBasePtr	0x0008
channel alternate control data base pointer Register	DMAxAltCtlBasePtr	0x000C
channel software request status Register	DMAxChnlSwRequest	0x0014
channel useburst set Register	DMAxChnlUseburstSet	0x0018
channel useburst clear Register	DMAxChnlUseburstClr	0x001C
channel request mask set Register	DMAxChnlReqMaskSet	0x0020
channel request mask clear Register	DMAxChnlReqMaskClr	0x0024
channel enable set Register	DMAxChnlEnableSet	0x0028
channel enable clear Register	DMAxChnlEnableClr	0x002C
channel primary-alternate set Register	DMAxChnlPriAltSet	0x0030
channel primary-alternate clear Register	DMAxChnlPriAltClr	0x0034
channel priority set Register	DMAxChnlPrioritySet	0x0038
channel priority clear Register	DMAxChnlPriorityClr	0x003C
Bus error clear Register	DMAxErrClr	0x004C

:DMAIF

		Address(Base+)
Flag Register A	DMAIFFLGA	0x0000

)

32bit

### 10.3.2 DMAxStatus (DMAC Status Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	1	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	master_ enable
					0	0	0	0

Bit	Bit Symbol	Type	
31-29	-	R	"0"
28	-	R	"1"
27-21	-	R	"0"
20-16	-	R	"1"
15-8	-	R	"0"
7-4	-	R	
3-1	-	R	"0"
0	master_enable	R	DMA 0: 1:

10.3.3 DMAxCfg (DMAC Configuration Register)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	master_ enable

Bit	Bit Symbol	Type	
31-1	-	W	"0"
0	master_enable	W	DMA 0 1

) DMAxCfg = 0x00000001, DMAxChnlReqMaskSet = 0xFFFFFFFF, DMAxChnlEnableSet = 0xFFFFFFFF  
(DMAxChnlReqMaskClr "1"

### 10.3.4 DMAxCtrlBasePtr (Channel control data base pointer Register)

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

Bit	Bit Symbol	Type	
31-10	ctrl_base_ptr	R/W	
9-0	-	R	"0"

### 10.3.5 DMAxAltCtrlBasePtr (Channel alternate control data base pointer Register)

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

Bit	Bit Symbol	Type	
31-0	alt_ctrl_base_pt	R	

10.3.6 DMAxChnlSwRequest(Channel software request Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_sw_re quest (ch31)	chnl_sw_re quest (ch30)	chnl_sw_re quest (ch29)	chnl_sw_re quest (ch28)	chnl_sw_re quest (ch27)	chnl_sw_re quest (ch26)	chnl_sw_re quest (ch25)	chnl_sw_re quest (ch24)
	23	22	21	20	19	18	17	16
bit symbol	chnl_sw_re quest (ch23)	chnl_sw_re quest (ch22)	chnl_sw_re quest (ch21)	chnl_sw_re quest (ch20)	chnl_sw_re quest (ch19)	chnl_sw_re quest (ch18)	chnl_sw_re quest (ch17)	chnl_sw_re quest (ch16)
	15	14	13	12	11	10	9	8
bit symbol	chnl_sw_re quest (ch15)	chnl_sw_re quest (ch14)	chnl_sw_re quest (ch13)	chnl_sw_re quest (ch12q)	chnl_sw_re quest (ch11)	chnl_sw_re quest (ch10)	chnl_sw_re quest (ch9)	chnl_sw_re quest (ch8)
	7	6	5	4	3	2	1	0
bit symbol	chnl_sw_re quest (ch7)	chnl_sw_re quest (ch6)	chnl_sw_re quest (ch5)	chnl_sw_re quest (ch4)	chnl_sw_re quest (ch3)	chnl_sw_re quest (ch2)	chnl_sw_re quest (ch1)	chnl_sw_re quest (ch0)

Bit	Bit Symbol	Type	
31-0	chnl_sw_request	W	DMA 0: 1:

10.3.7 DMAxChnlUseburstSet(Channel useburst set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_useburst_set (ch31)	chnl_useburst_set (ch30)	chnl_useburst_set (ch29)	chnl_useburst_set (ch28)	chnl_useburst_set (ch27)	chnl_useburst_set (ch26)	chnl_useburst_set (ch25)	chnl_useburst_set (ch24)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_useburst_set (ch23)	chnl_useburst_set (ch22)	chnl_useburst_set (ch21)	chnl_useburst_set (ch20)	chnl_useburst_set (ch19)	chnl_useburst_set (ch18)	chnl_useburst_set (ch17)	chnl_useburst_set (ch16)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_useburst_set (ch15)	chnl_useburst_set (ch14)	chnl_useburst_set (ch13)	chnl_useburst_set (ch12)	chnl_useburst_set (ch11)	chnl_useburst_set (ch10)	chnl_useburst_set (ch9)	chnl_useburst_set (ch8)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_useburst_set (ch7)	chnl_useburst_set (ch6)	chnl_useburst_set (ch5)	chnl_useburst_set (ch4)	chnl_useburst_set (ch3)	chnl_useburst_set (ch2)	chnl_useburst_set (ch1)	chnl_useburst_set (ch0)
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	chnl_useburst_set	R/W	<p>[     ]</p> <p>1:</p> <p>[     ]</p> <p>0:</p> <p>1:</p> <p>"1"</p> <p>"0"</p> <p style="text-align: right;">DMAxChnlUseburstClr</p> <p style="text-align: center;">/</p> <p style="text-align: center;">2<sup>R</sup>     2<sup>R</sup>     ("R"     channel_cfg&lt;R_power&gt;     )</p> <p style="text-align: center;">2<sup>R</sup>     "0"</p> <p style="text-align: center;">DMA     channel_cfg&lt;next_useburst&gt;     "1"</p> <p style="text-align: center;">"1"</p>

)

2<sup>R</sup>

"1"

10.3.8 DMAxChnlUseburstClr(Channel useburst clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_useburst_clr (ch31)	chnl_useburst_clr (ch30)	chnl_useburst_clr (ch29)	chnl_useburst_clr (ch28)	chnl_useburst_clr (ch27)	chnl_useburst_clr (ch26)	chnl_useburst_clr (ch25)	chnl_useburst_clr (ch24)
	23	22	21	20	19	18	17	16
bit symbol	chnl_useburst_clr (ch23)	chnl_useburst_clr (ch22)	chnl_useburst_clr (ch21)	chnl_useburst_clr (ch20)	chnl_useburst_clr (ch19)	chnl_useburst_clr (ch18)	chnl_useburst_clr (ch17)	chnl_useburst_clr (ch16)
	15	14	13	12	11	10	9	8
bit symbol	chnl_useburst_clr (ch15)	chnl_useburst_clr (ch14)	chnl_useburst_clr (ch13)	chnl_useburst_clr (ch12)	chnl_useburst_clr (ch11)	chnl_useburst_clr (ch10)	chnl_useburst_clr (ch9)	chnl_useburst_clr (ch8)
	7	6	5	4	3	2	1	0
bit symbol	chnl_useburst_clr (ch7)	chnl_useburst_clr (ch6)	chnl_useburst_clr (ch5)	chnl_useburst_clr (ch4)	chnl_useburst_clr (ch3)	chnl_useburst_clr (ch2)	chnl_useburst_clr (ch1)	chnl_useburst_clr (ch0)

Bit	Bit Symbol	Type	
31-0	chnl_useburst_clr	W	1: "1" "0" DMAxChnlUseburstSet



10.3.10 DMAxChnlReqMaskClr(Channel request mask clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_req_mas k_clr (ch31)	chnl_req_mas k_clr (ch30)	chnl_req_mas k_clr (ch29)	chnl_req_mas k_clr (ch28)	chnl_req_mas k_clr (ch27)	chnl_req_mas k_clr (ch26)	chnl_req_mas k_clr (ch25)	chnl_req_mas k_clr (ch24)
	23	22	21	20	19	18	17	16
bit symbol	chnl_req_mas k_clr (ch23)	chnl_req_mas k_clr (ch22)	chnl_req_mas k_clr (ch21)	chnl_req_mas k_clr (ch20)	chnl_req_mas k_clr (ch19)	chnl_req_mas k_clr (ch18)	chnl_req_mas k_clr (ch17)	chnl_req_mas k_clr (ch16)
	15	14	13	12	11	10	9	8
bit symbol	chnl_req_mas k_clr (ch15)	chnl_req_mas k_clr (ch14)	chnl_req_mas k_clr (ch13)	chnl_req_mas k_clr (ch12)	chnl_req_mas k_clr (ch11)	chnl_req_mas k_clr (ch10)	chnl_req_mas k_clr (ch9)	chnl_req_mas k_clr (ch8)
	7	6	5	4	3	2	1	0
bit symbol	chnl_req_mas k_clr (ch7)	chnl_req_mas k_clr (ch6)	chnl_req_mas k_clr (ch5)	chnl_req_mas k_clr (ch4)	chnl_req_mas k_clr (ch3)	chnl_req_mas k_clr (ch2)	chnl_req_mas k_clr (ch1)	chnl_req_mas k_clr (ch0)

Bit	Bit Symbol	Type	
31-0	chnl_req_mask_clr	W	DMA 1: DMA "1" DMA "0" DMAxChnlReqMaskSet



10.3.12 DMAxChnlEnableClr(Channel enable clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_enable_clr (ch31)	chnl_enable_clr (ch30)	chnl_enable_clr (ch29)	chnl_enable_clr (ch28)	chnl_enable_clr (ch27)	chnl_enable_clr (ch26)	chnl_enable_clr (ch25)	chnl_enable_clr (ch24)
	23	22	21	20	19	18	17	16
bit symbol	chnl_enable_clr (ch23)	chnl_enable_clr (ch22)	chnl_enable_clr (ch21)	chnl_enable_clr (ch20)	chnl_enable_clr (ch19)	chnl_enable_clr (ch18)	chnl_enable_clr (ch17)	chnl_enable_clr (ch16)
	15	14	13	12	11	10	9	8
bit symbol	chnl_enable_clr (ch15)	chnl_enable_clr (ch14)	chnl_enable_clr (ch13)	chnl_enable_clr (ch12)	chnl_enable_clr (ch11)	chnl_enable_clr (ch10)	chnl_enable_clr (ch9)	chnl_enable_clr (ch8)
	7	6	5	4	3	2	1	0
bit symbol	chnl_enable_clr (ch7)	chnl_enable_clr (ch6)	chnl_enable_clr (ch5)	chnl_enable_clr (ch4)	chnl_enable_clr (ch3)	chnl_enable_clr (ch2)	chnl_enable_clr (ch1)	chnl_enable_clr (ch0)

Bit	Bit Symbol	Type	
31-0	chnl_enable_clr	W	DMA 1: "1" <span style="float: right;">"0"</span> <div style="text-align: center;">DMAxChnlEnableSet</div> DMA channel_cfg<cycle_ctrl> "000"

10.3.13 DMAxChnlPriAltSet(Channel primary-alternate set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_pri_alt_set (ch31)	chnl_pri_alt_set (ch30)	chnl_pri_alt_set (ch29)	chnl_pri_alt_set (ch28)	chnl_pri_alt_set (ch27)	chnl_pri_alt_set (ch26)	chnl_pri_alt_set (ch25)	chnl_pri_alt_set (ch24)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_pri_alt_set (ch23)	chnl_pri_alt_set (ch22)	chnl_pri_alt_set (ch21)	chnl_pri_alt_set (ch20)	chnl_pri_alt_set (ch19)	chnl_pri_alt_set (ch18)	chnl_pri_alt_set (ch17)	chnl_pri_alt_set (ch16)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_pri_alt_set (ch15)	chnl_pri_alt_set (ch14)	chnl_pri_alt_set (ch13)	chnl_pri_alt_set (ch12)	chnl_pri_alt_set (ch11)	chnl_pri_alt_set (ch10)	chnl_pri_alt_set (ch9)	chnl_pri_alt_set (ch8)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_pri_alt_set (ch7)	chnl_pri_alt_set (ch6)	chnl_pri_alt_set (ch5)	chnl_pri_alt_set (ch4)	chnl_pri_alt_set (ch3)	chnl_pri_alt_set (ch2)	chnl_pri_alt_set (ch1)	chnl_pri_alt_set (ch0)
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	chnl_pri_alt_set	R/W	<div style="display: flex; justify-content: space-between;"> <span>[ ]</span> <span>/</span> </div> <div style="display: flex; justify-content: space-between;"> <span>1: [ ]</span> <span></span> </div> <div style="display: flex; justify-content: space-between;"> <span>0: [ ]</span> <span></span> </div> <div style="display: flex; justify-content: space-between;"> <span>1: [ ]</span> <span></span> </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <span>"1"</span> <span>DMAxChnlEnableClr</span> <span>"0"</span> </div>

10.3.14 DMAxChnIPriAltClr(Channel primary-alternate clear Register)

	31	30	29	28	27	26	25	24
bit symbol	chn_pri_alt_clr (ch31)	chn_pri_alt_clr (ch30)	chn_pri_alt_clr (ch29)	chn_pri_alt_clr (ch28)	chn_pri_alt_clr (ch27)	chn_pri_alt_clr (ch26)	chn_pri_alt_clr (ch25)	chn_pri_alt_clr (ch24)
	23	22	21	20	19	18	17	16
bit symbol	chn_pri_alt_clr (ch23)	chn_pri_alt_clr (ch22)	chn_pri_alt_clr (ch21)	chn_pri_alt_clr (ch20)	chn_pri_alt_clr (ch19)	chn_pri_alt_clr (ch18)	chn_pri_alt_clr (ch17)	chn_pri_alt_clr (ch16)
	15	14	13	12	11	10	9	8
bit symbol	chn_pri_alt_clr (ch15)	chn_pri_alt_clr (ch14)	chn_pri_alt_clr (ch13)	chn_pri_alt_clr (ch12)	chn_pri_alt_clr (ch11)	chn_pri_alt_clr (ch10)	chn_pri_alt_clr (ch9)	chn_pri_alt_clr (ch8)
	7	6	5	4	3	2	1	0
bit symbol	chn_pri_alt_clr (ch7)	chn_pri_alt_clr (ch6)	chn_pri_alt_clr (ch5)	chn_pri_alt_clr (ch4)	chn_pri_alt_clr (ch3)	chn_pri_alt_clr (ch2)	chn_pri_alt_clr (ch1)	chn_pri_alt_clr (ch0)

Bit	Bit Symbol	Type	
31-0	chnl_pri_alt_clr	W	1: "1" <span style="margin-left: 150px;">DMAxChnIPriAltSet</span> <span style="margin-left: 150px;">"0"</span>

### 10.3.15 DMAxChnlPrioritySet(Channel priority set Register)

	31	30	29	28	27	26	25	24
bit symbol	chnl_priority_set (ch31)	chnl_priority_set (ch30)	chnl_priority_set (ch29)	chnl_priority_set (ch28)	chnl_priority_set (ch27)	chnl_priority_set (ch26)	chnl_priority_set (ch25)	chnl_priority_set (ch24)
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	chnl_priority_set (ch23)	chnl_priority_set (ch22)	chnl_priority_set (ch21)	chnl_priority_set (ch20)	chnl_priority_set (ch19)	chnl_priority_set (ch18)	chnl_priority_set (ch17)	chnl_priority_set (ch16)
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	chnl_priority_set (ch15)	chnl_priority_set (ch14)	chnl_priority_set (ch13)	chnl_priority_set (ch12)	chnl_priority_set (ch11)	chnl_priority_set (ch10)	chnl_priority_set (ch9)	chnl_priority_set (ch8)
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	chnl_priority_set (ch7)	chnl_priority_set (ch6)	chnl_priority_set (ch5)	chnl_priority_set (ch4)	chnl_priority_set (ch3)	chnl_priority_set (ch2)	chnl_priority_set (ch1)	chnl_priority_set (ch0)
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	chnl_priority_set	R/W	<p>[     ]                      1:                      [     ]                      0:                      1:                      "1"                      DMAxChnlPriorityClr                      "0"</p>



10.3.17 DMAxErrClr(Bus error clear Register)

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

Bit	Bit Symbol	Type	
31-1	-	R	"0"
0	err_clr	R/W	[     ] 1: [     ] 0: 1: "1"
			"0"

10.3.18 DMAIFFLGx(DMA Flag Register)

	31	30	29	28	27	26	25	24
bit symbol	FLG31	FLG30	FLG29	FLG28	FLG27	FLG26	FLG25	FLG24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	FLG23	FLG22	FLG21	FLG20	FLG19	FLG18	FLG17	FLG16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	FLG15	FLG14	FLG13	FLG12	FLG11	FLG10	FLG9	FLG8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	FLG7	FLG6	FLG5	FLG4	FLG3	FLG2	FLG1	FLG0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-1	FLG31 - FLG0	R	DMA 0: 1:  DMA "1"  "0"

) DMA

## 10.4

本 DMA は、チャンネル制御データによって制御されます。チャンネル制御データはメモリ上に置かれたデータで、1チャンネルにつき4ワードのデータをチャンネル数分連続した空間に配置します。

チャンネル制御データには、一次データと代替データがあります。動作モードによってどちらかを設定レジスタで選択して使用する場合と、両方を使用する場合があります。

### 10.4.1

図 10-2 にチャンネル制御データのメモリマップ例を示します。

一次データ用のスタートアドレスを DMAxCtrlBasePtr に、代替データ用のスタートアドレスを DMAxAltCtrlBasePtr に設定します。

Alternate Ch31	0x3F0	Primary Ch31	0x1F0
Alternate Ch30	0x3E0	Primary Ch30	0x1E0
Alternate Ch29	0x3D0	Primary Ch29	0x1D0
Alternate Ch28	0x3C0	Primary Ch28	0x1C0
Alternate Ch27	0x3B0	Primary Ch27	0x1B0
Alternate Ch26	0x3A0	Primary Ch26	0x1A0
Alternate Ch25	0x390	Primary Ch25	0x190
Alternate Ch24	0x380	Primary Ch24	0x180
Alternate Ch23	0x370	Primary Ch23	0x170
Alternate Ch22	0x360	Primary Ch22	0x160
Alternate Ch21	0x350	Primary Ch21	0x150
Alternate Ch20	0x340	Primary Ch20	0x140
Alternate Ch19	0x330	Primary Ch19	0x130
Alternate Ch18	0x320	Primary Ch18	0x120
Alternate Ch17	0x310	Primary Ch17	0x110
Alternate Ch16	0x300	Primary Ch16	0x100
Alternate Ch15	0x2F0	Primary Ch15	0x0F0
Alternate Ch14	0x2E0	Primary Ch14	0x0E0
Alternate Ch13	0x2D0	Primary Ch13	0x0D0
Alternate Ch12	0x2C0	Primary Ch12	0x0C0
Alternate Ch11	0x2B0	Primary Ch11	0x0B0
Alternate Ch10	0x2A0	Primary Ch10	0x0A0
Alternate Ch9	0x290	Primary Ch9	0x090
Alternate Ch8	0x280	Primary Ch8	0x080
Alternate Ch7	0x270	Primary Ch7	0x070
Alternate Ch6	0x260	Primary Ch6	0x060
Alternate Ch5	0x250	Primary Ch5	0x050
Alternate Ch4	0x240	Primary Ch4	0x040
Alternate Ch3	0x230	Primary Ch3	0x030
Alternate Ch2	0x220	Primary Ch2	0x020
Alternate Ch1	0x210	Primary Ch1	0x010
Alternate Ch0	0x200	Primary Ch0	0x000

Reserved	0x00C
Control	0x008
Destination End Pointer	0x004
Source End Pointer	0x000

10-2

図 10-2 は、32チャンネルすべてが使用できる場合のメモリマップです。使用できるチャンネル数により必要となる領域は異なります。チャンネル数とアドレスの関係を表 10-2 に示します。

10-2

	[9]	[8]	[7]	[6]	[5]	[4]	[3:0]	
0	-	-	-	-	-	A		0xFFFF_X00, 0xFFFF_X20, 0xFFFF_X40, 0xFFFF_X60, 0xFFFF_X80, 0xFFFF_XA0, 0xFFFF_XC0, 0xFFFF_XE0
0~1	-	-	-	-	A	C[0]		0xFFFF_X00, 0xFFFF_X40, 0xFFFF_X80, 0xFFFF_XC0
0~3	-	-	-	A	C[1:0]			0xFFFF_X00, 0xFFFF_X80
0~7	-	-	A	C[2:0]				0xFFFF_X00, 0xFFFF_X100, 0xFFFF_X200, 0xFFFF_X300, 0xFFFF_X400, 0xFFFF_X500, 0xFFFF_X600, 0xFFFF_X700, 0xFFFF_X800, 0xFFFF_X900, 0xFFFF_XA00, 0xFFFF_XB00, 0xFFFF_XC00, 0xFFFF_XD00, 0xFFFF_XE00, 0xFFFF_XF00
0~15	-	A	C[3:0]					0xFFFF_X00, 0xFFFF_X200, 0xFFFF_X400, 0xFFFF_X600, 0xFFFF_X800, 0xFFFF_XA00, 0xFFFF_XC00, 0xFFFF_XE00
0~31	A	C[4:0]						0xFFFF_X00, 0xFFFF_X400, 0xFFFF_X800, 0xFFFF_XC00

A: / (0: 1: )  
C[x:0]:

10.4.2

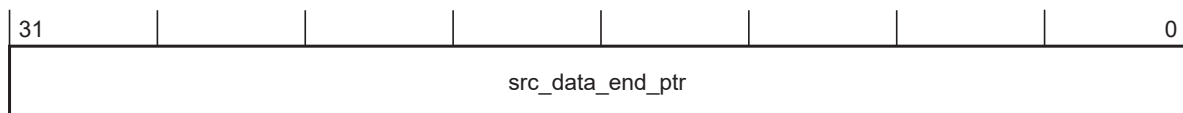
チャンネル制御データは以下の3つのデータを含みます。

- ・ 転送元データの最終アドレス
- ・ 転送先の最終アドレス
- ・ 制御データ

それぞれの内容について以下に説明します。

10.4.2.1

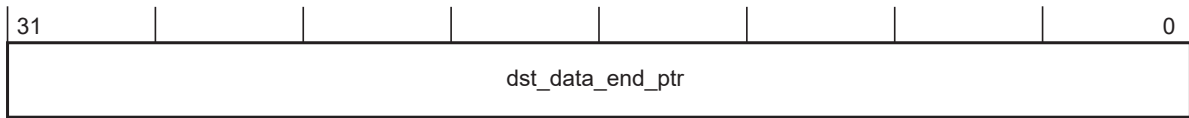
転送するデータの最終アドレスを設定します。アドレスのアライメントは、転送データサイズに合わせてください。このアドレスを元にDMAが転送元のスタートアドレスを計算します。



bit	bitsymbol	
[31:0]	src_data_end_ptr	

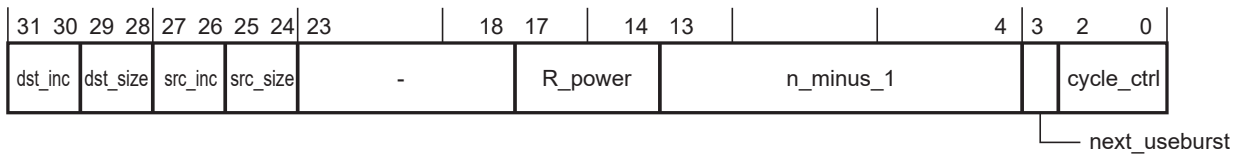
10.4.2.2

転送先の最終アドレスを設定します。アドレスのアライメントは、転送データサイズに合わせてください。このアドレスを元に DMA が転送先のスタートアドレスを計算します。



bit	bitsymbol
[31:0]	dst_data_end_ptr

10.4.2.3



bit	bit symbol	
[31:30]	dst_inc	2) 00: 1byte 01: 2byte 10: 4byte 11:
[29:28]	dst_size	( 1) 00: 1byte 01: 2byte 10: 4byte 11: Reserved
[27:26]	src_inc	2) 00: 1byte 01: 2byte 10: 4byte 11:
[25:24]	src_size	( 1) 00: 1byte 01: 2byte 10: 4byte 11: Reserved
[23:18]	-	"000000"

bit	bit symbol	
[17:14]	R_power	0000: 1 0001: 2 0010: 4 0011: 8 0100: 16 0101: 32 0110: 64 0111: 128 1000: 256 1001: 512 1010 - 1111:
[13:4]	n_minus_1	0x000: 1 0x001: 2 0x002: 3 : 0x3FF: 1024
[3]	next_useburst	0: <chnl_useburst_set> 1: <chnl_useburst_set> "1"  DMA <chnl_useburst_set> "1" ) 2 2 <sup>R</sup> ("R" <R_power> ) 2 <sup>R</sup> "1" <chnl_useburst_set> "0" <chnl_useburst_set> "1"
[2:0]	cycle_ctrl	000: DMA 001: 010: 011: 100: ( ) 101: ( ) 110: ( ) 111: ( )

- 1) <dst\_size> <src\_size>
- 2) <dst\_size> <src\_size> <dst\_inc> <src\_inc>

<src_inc>/<dst_inc>	<src_size>/<dst_size>		
	00 (1byte)	01 (2byte)	10 (4byte)
00(1byte)	o	-	-
01(2byte)	o	o	-
10(4byte)	o	o	o
	o	o	o

### 10.4.3

チャンネル制御データの channel\_cfg<cycle\_ctrl>で設定する動作モードについて説明します。

#### 10.4.3.1

転送終了後に DMA は動作モードを無効に設定します。これにより、再度同じ転送が行われることを防ぎます。また、ピンポンモード、メモリスキャッターギャザーモード、周辺スキャッターギャザーモードの際に、無効設定のデータを読み込むと処理を終了します。

#### 10.4.3.2

基本モードでは、一次または代替のどちらのデータ構造を使用するか設定が可能です。

転送要求により転送を開始します。

<R\_power>設定の転送ごとにアービトレーションを行い、より高い優先度の要求があればチャンネルを切り替えます。動作中のチャンネルの転送要求があると、転送を継続します。

<n\_minus\_1>に設定された回数の転送を行った後、転送終了割り込みを発生します。

#### 10.4.3.3

このモードでは 1 回の転送要求で転送を終了させることができます。一次または代替のどちらのデータ構造を使用するか設定が可能です。

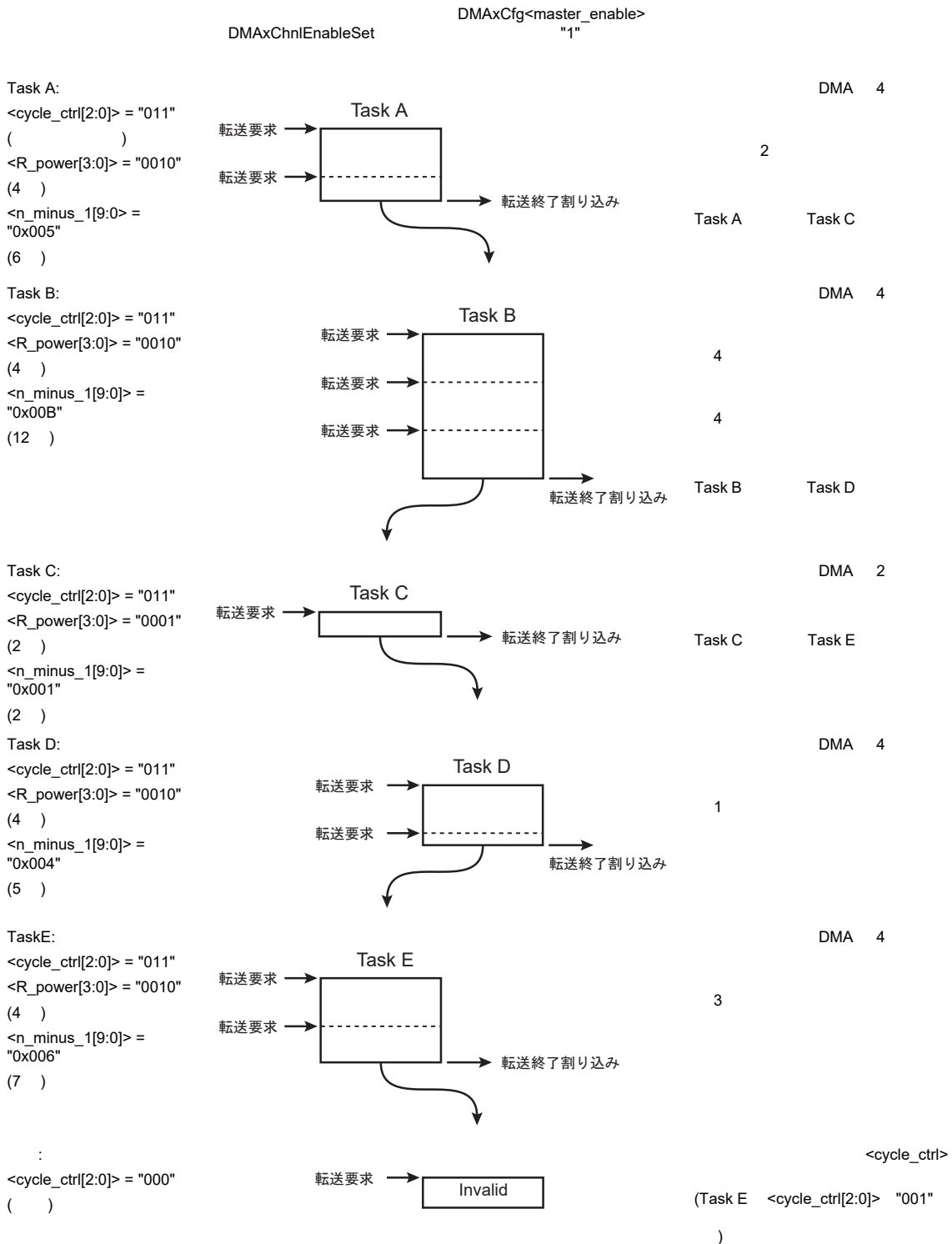
転送要求により転送を開始します。

<R\_power>設定の転送ごとに、より高い優先度の要求があればチャンネルを切り替えます。なければ転送を継続します。

<n\_minus\_1>に設定された回数の転送を行った後、転送終了割り込みを発生します。

#### 10.4.3.4

ピンポンモードでは、一次データと代替データを交互に使用しながら連続した DMA 転送を行います。<cycle\_ctrl>に無効("000")が設定されたデータを読み込むか、チャンネルが無効に設定されると転送を終了します。一次データおよび代替データによる転送が終了するごとに転送終了割り込みを発生します。



10.4.3.5

メモリスキャッターギャザーモードでは、一次データは代替データ用のデータを転送するために使用します。

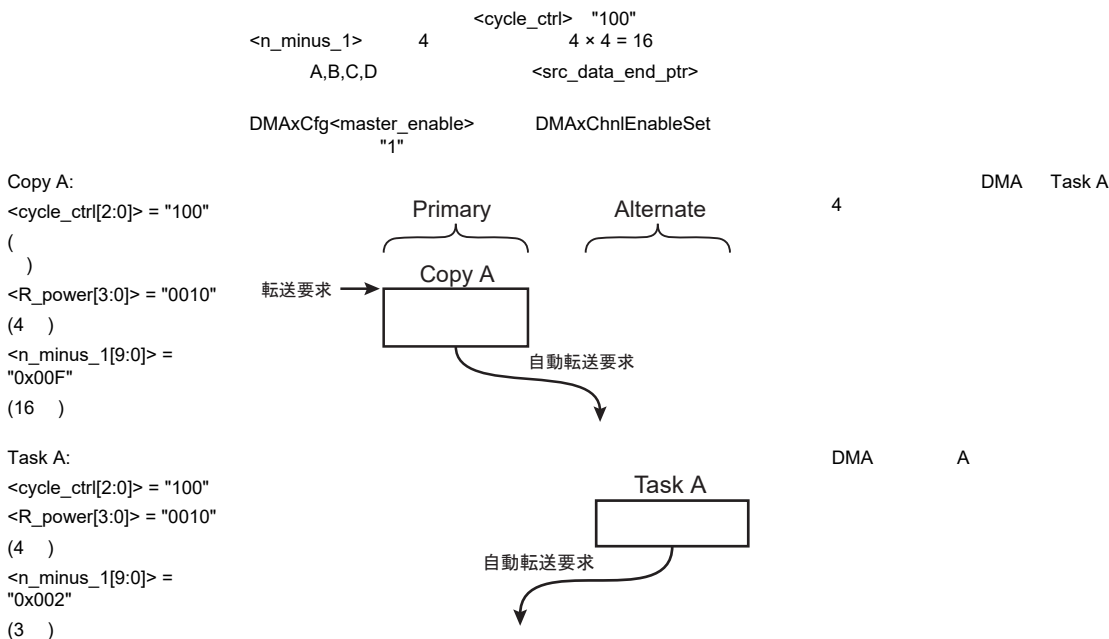
転送要求を受けると、一次データを用いて代替データの4つのデータを転送し、新たな転送要求なしに続けて代替データによるデータ転送を行います。その後、一次データによる代替データの転送と代替データによる転送を、<cycle\_ctrl[2:0]>に無効("000")または基本モード("001")が設定されたデータを読み込むまで行います。この間、新たな転送要求必要ありません。転送終了後、割り込みを発生します。

このモードでは、一次データの channel\_cfg の設定を以下のように設定する必要があります。

10-3 ( )

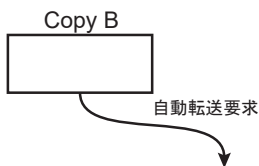
bit	bit symbol		
[31:30]	dst_inc	10	4byte
[29:28]	dst_size	10	4byte
[27:26]	src_inc	10	4byte
[25:24]	src_size	10	4byte
[17:14]	R_power	0010	4
[13:4]	n_minus_1	N	<sup>3</sup> 4
[3]	next_useburst	0	"0"
[2:0]	cycle_ctrl	100	( )

) <n\_minus\_1> "000"



Copy B:

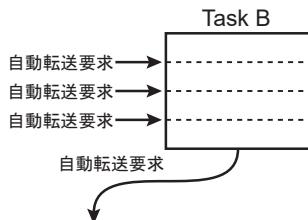
DMA Task B 4



Task B:

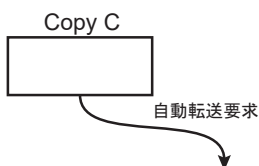
DMA B

<cycle\_ctrl[2:0]> = "100"  
 <R\_power[3:0]> = "0001"  
 (2 )  
 <n\_minus\_1[9:0]> =  
 "0x007"  
 (8 )



Copy C:

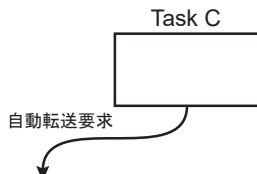
DMA Task C 4



Task C:

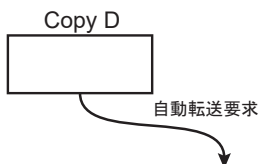
DMA C

<cycle\_ctrl[2:0]> = "100"  
 <R\_power[3:0]> = "0011"  
 (8 )  
 <n\_minus\_1[9:0]> =  
 "0x004"  
 (5 )



Copy D:

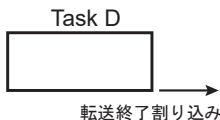
DMA Task D 4  
 "000" <cycle\_ctrl[2:0]>



Task D:

DMA D  
 <cycle\_ctrl[2:0]> "001"

<cycle\_ctrl[2:0]> = "001"  
 <R\_power[3:0]> = "0010"  
 (4 )  
 <n\_minus\_1[9:0]> =  
 "0x003"  
 (4 )



### 10.4.3.6

周辺スキャッターギャザーモードでは、一次データは代替データ用のデータを転送するために使用します。

転送要求を受けると、一次データを用いて代替データの4つのデータを転送し、続けて代替データによるデータ転送を行います。

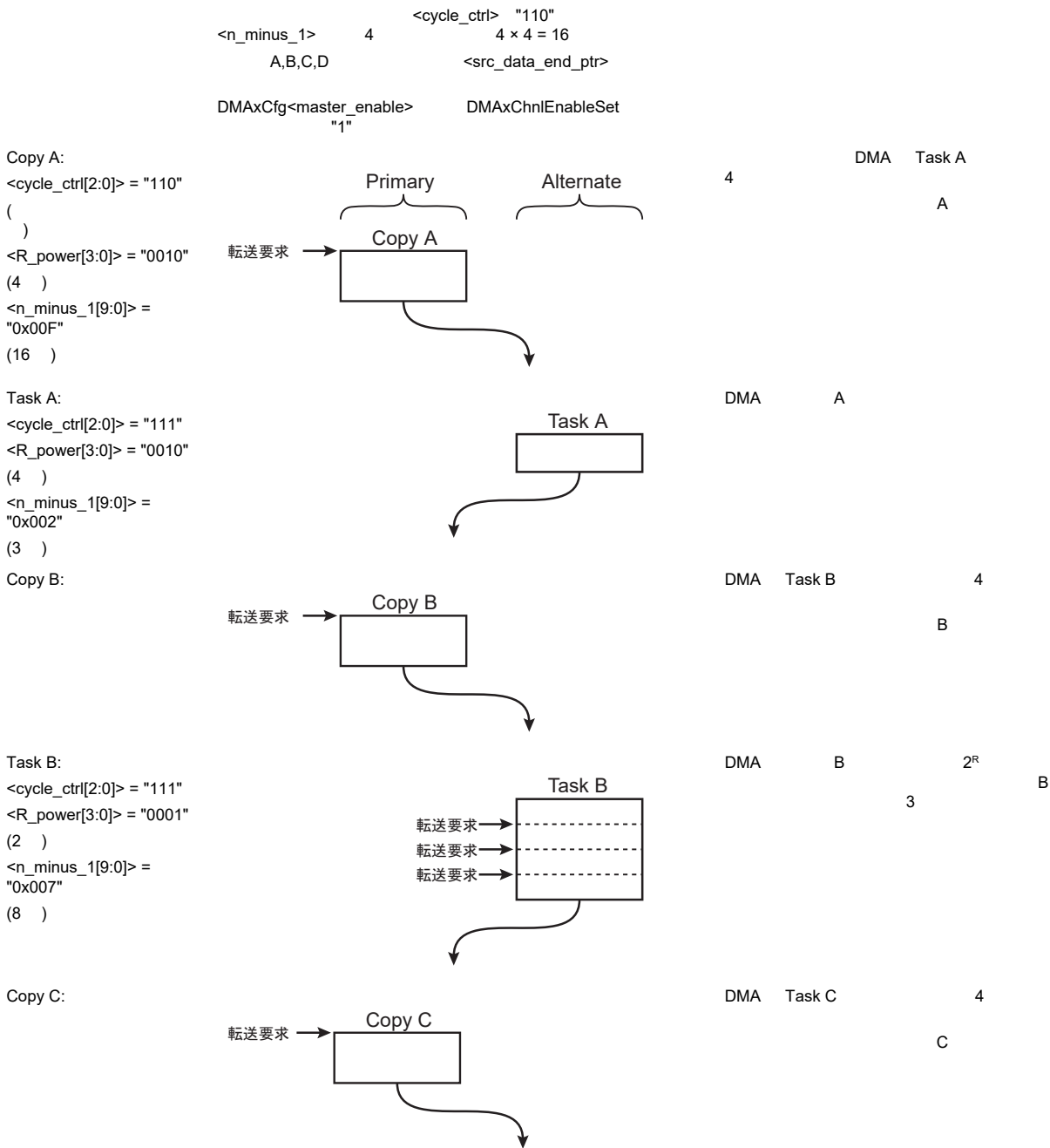
その後、転送要求が発生すると一次データによる代替データの転送と代替データによる転送を、<cycle\_ctrl[2:0]>に無効("000")または基本モード("001")が設定されたデータを読み込むまで行います。この間、新たな転送要求必要ありません。転送終了後、割り込みを発生します。

このモードでは、channel\_cfg の設定を以下のようにする必要があります。

10-4 ( )

bit	bit symbol		
[31:30]	dst_inc	10	4byte
[29:28]	dst_size	10	4byte
[27:26]	src_inc	10	4byte
[25:24]	src_size	10	4byte
[17:14]	R_power	0010	4
[13:4]	n_minus_1	N	<sup>3</sup> 4
[2:0]	cycle_ctrl	110	( )

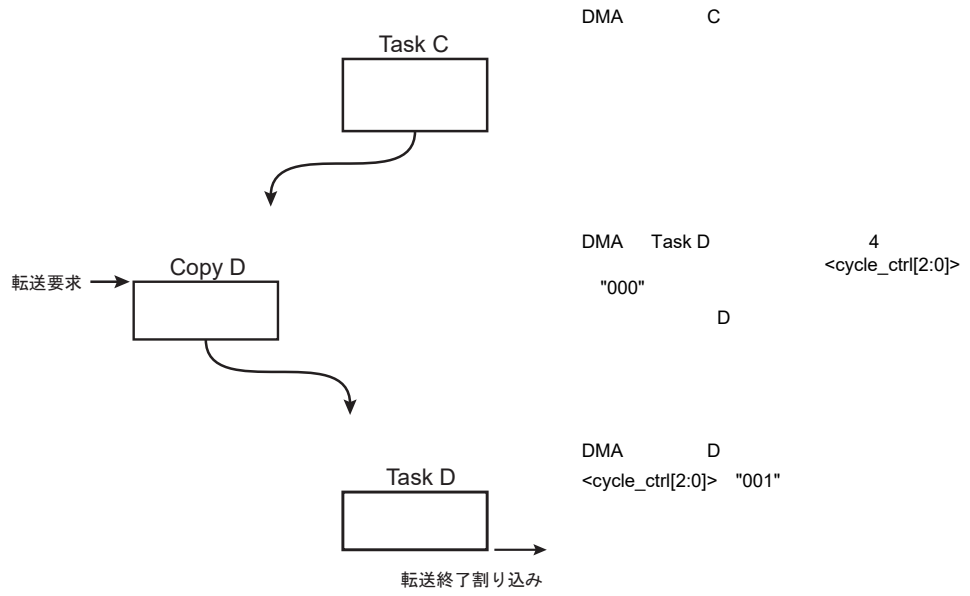
) <n\_minus\_1> "000"



Task C:  
 <cycle\_ctrl[2:0]> = "111"  
 <R\_power[3:0]> = "0011"  
 (8 )  
 <n\_minus\_1[9:0]> =  
 "0x004"  
 (5 )

Copy D:

Task D:  
 <cycle\_ctrl[2:0]> = "001"  
 <R\_power[3:0]> = "0010"  
 (4 )  
 <n\_minus\_1[9:0]> =  
 "0x003"  
 (4 )



## 10.5

下記の周辺機能の DMA 転送要求を使用して転送を行う場合、使い方に注意が必要です。

- ・ 4 バイト FIFO 付きシリアルチャネル(SIO/UART)
- ・ 16 ビットタイマ/イベントカウンタ(TMRB)
- ・ アナログ/デジタルコンバータ(ADC)

### 10.5.1 SIO/UART TMRB ADC

以下の点に注意して使用してください。

- ・ 転送モードは基本モードを推奨します。
- ・ アービトレーションは"1 回転送後"にしてください。  
制御データのアービトレーション<R\_power>設定を"0000"としてください。
- ・ SIO/UART の FIFO は使用しないでください。

SIO/UART は、シングルバッファまたはダブルバッファの設定で使用してください。

転送開始が待たされて新たな要求が同一チャネルで発生した場合、転送は 1 回しか行われません。確実に転送が行われるよう、プログラム設計の際に考慮してください。

転送開始が待たされる状況として、以下のような場合が考えられます。

- ・ 同一ユニット内の優先度の高い転送要求が発生した場合
- ・ 他の優先度の高いバスマスタとアクセス対象が同じ場合

本 μDMA コントローラは、前処理/後処理に約 11 クロック、周辺機能と内蔵 RAM 間の 1 データの転送に約 5 クロックかかります。転送の待ち時間の目安としてください。

# 11

本章では、ポート関連のレジスタとその設定および回路について説明します。

## 11.1

ポートを使用する際には以下のレジスタを設定する必要があります。

レジスタは全て 32 ビットですが、ポートのビット数、機能の割り当てにより構成が異なります。

以下の説明では"x"はポート名、"n"はファンクション番号を示します。

PxDATA		0	1	
PxCR		0:	1:	
PxFRn	n	0: PORT	1:	"1"  1
PxOD		0: CMOS	1:	"1" PxOD
PxPUP		0:	1:	
PxPDN		0:	1:	
PxIE		0:	1:	PxIE PxDATA

### 11.1.1

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

機能の存在しないビットをリードすると"0"が読めます。ライトは意味を持ちません。

	Address (Base+)	A	B	C	D	E
	0x0000	PADATA	PBDATA	PCDATA	PDDATA	PEDATA
	0x0004	PACR	PBCR	PCCR	PDCR	PECR
1	0x0008	PAFR1	PBFR1	PCFR1	PDFR1	PEFR1
2	0x000C	PAFR2			PDFR2	PEFR2
	0x0028	PAOD	PBOD	PCOD	PDOD	PEOD
	0x002C	PAPUP	PBPUP	PCPUP	PDPUP	PEPUP
	0x0030	PAPDN	PBPDN	PCPDN	PDPDN	PEPDN
	0x0038	PAIE	PBIE	PCIE	PDIE	PEIE

	Address (Base+)	F	G	H	J	K
	0x0000	PFDATA	PGDATA	PHDATA	PJDATA	PKDATA
	0x0004	PFCR	PGCR	PHCR	PJCR	PKCR
1	0x0008	PFFR1	PGFR1			PKFR1
2	0x000C	PFFR2				
3	0x0010	PFFR3				
	0x0028	PFOD	PGOD	PHOD	PJOD	PKOD
	0x002C	PFPUP	PGPUP	PHPUP	PJPUP	PKPUP
	0x0030	PFPDN	PGPDN	PHPDN	PJPDN	PKPDN
	0x0038	PFIE	PGIE	PHIE	PJIE	PKIE

	Address (Base+)	L	N	P
	0x0000	PLDATA	PNDATA	PPDATA
	0x0004	PLCR	PNCR	PPCR
1	0x0008	PLFR1	PNFR1	
2	0x000C		PNFR2	
	0x0028	PLOD	PNOD	PPOD
	0x002C	PLPUP	PNPUP	PPPUP
	0x0030	PLPDN	PNPDN	PPPDN
	0x0038	PLIE	PNIE	PPIE

) "-"

## 11.1.2

以下に各ポートの機能とレジスタ設定についての一覧を示します。

- 「表 11-1 ポート A レジスタ設定」
- 「表 11-2 ポート B レジスタ設定」
- 「表 11-3 ポート C レジスタ設定」
- 「表 11-4 ポート D レジスタ設定」
- 「表 11-5 ポート E レジスタ設定」
- 「表 11-6 ポート F レジスタ設定」
- 「表 11-7 ポート G レジスタ設定」
- 「表 11-8 ポート H レジスタ設定」
- 「表 11-9 ポート J レジスタ設定」
- 「表 11-10 ポート K レジスタ設定」
- 「表 11-11 ポート L レジスタ設定」
- 「表 11-12 ポート N レジスタ設定」
- 「表 11-13 ポート P レジスタ設定」

PxFRn の欄は、設定の必要なファンクションレジスタを示します。このレジスタを"1"に設定するとその機能が有効となります。(x はポート名、n はファンクション番号)

表中の網掛けのビットはリードすると"0"が読め、ライトは意味を持ちません。

表中の"0"、"1"は設定値を示し、"0/1"は任意に設定可能であることを示します。

TMPM475FDFG/FZFG/FYFG では異なるポートに同一機能が割り振られている機能端子があります。同一機能の兼用機能端子はそれぞれのポートのファンクションレジスタの設定により、いずれか1つの端子で使用するよう排他的に切り替えを行って下さい。

Pxm は、ポート名"x"、そのレジスタのビット"m"を表します。例えば、PA0 とは、ポート名が A、ポート A のレジスタのビットが 0 を示します。

11.1.2.1 PORT A

11-1 A

PORT	Input/Output	PORT Type								
			PADATA	PACR	PAFRn	PAOD	PAPUP	PAPDN	PAIE	
PA0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INT3	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB0IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
PA1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TB0OUT	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
PA2			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INT4	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB1IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
PA3			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TB1OUT	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
PA4			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	SC1SCLK	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
		Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	SC1CTS	Input	FT2	0/1	0	PAFR2	0/1	0/1	0/1	1
PA5			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	SC1TXD	Output	FT1	0/1	1	PAFR1	0/1	0/1	0/1	0
	TB6OUT	Output	FT1	0/1	1	PAFR2	0/1	0/1	0/1	0
PA6			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	SC1RXD	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1
	TB6IN	Input	FT1	0/1	0	PAFR2	0/1	0/1	0/1	1
PA7			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INT8	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TB4IN	Input	FT1	0/1	0	PAFR1	0/1	0/1	0/1	1

11.1.2.2 PORT B

11-2 B

PO RT	Input/Output	PORT Type								
			PBDATA	PBCR	PBFRn	PBOD	PBPUP	PBPDN	PBIE	
PB0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TRACECLK0	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TRACEDATA0	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB2			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TRACEDATA1	Output	FT1	0/1	1	PBFR1	0/1	0/1	0/1	0
PB3	(TSM/SWDIO)		0	1	PBFR1	0	1	0	0	1
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TMS	I/O	FT3	0/1	1	PBFR1	0/1	0/1	0/1	1
	SWDIO	I/O	FT3	0/1	1	PBFR1	0/1	0/1	0/1	1
PB4	(TCK/SWCLK)		0	0	PBFR1	0	0	1	1	1
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TCK	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
	SWCLK	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
PB5	(TDO/SWV)		0	1	PBFR1	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TDO	Output	FT3	0/1	1	PBFR1	0/1	0/1	0/1	0
	SWV	Output	FT3	0/1	1	PBFR1	0/1	0/1	0/1	0
PB6	(TDI)		0	0	PBFR1	0	1	0	0	1
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TDI	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1
PB7	( $\overline{\text{TRST}}$ )		0	0	PBFR1	0	1	0	0	1
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	$\overline{\text{TRST}}$	Input	FT3	0/1	0	PBFR1	0/1	0/1	0/1	1

11.1.2.3 PORT C

11-3 C

PO RT	Input/Output	PORT Type								
			PCDATA	PCCR	PCFRn	PCOD	PCPUP	PCPDN	PCIE	
PC0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	UO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	XO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC2			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	VO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC3			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	YO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC4			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	WO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC5			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ZO0	Output	FT2	0/1	1	PCFR1	0/1	0/1	0/1	0
PC6			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	EMG0	Input	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1
PC7			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	OVV0	Input	FT1	0/1	0	PCFR1	0/1	0/1	0/1	1

11.1.2.4 PORT D

11-4 D

PORT	Input/Output	PORT Type								
			PDDATA	PDCR	PDFRn	PDOD	PDPUP	PDPDN	PDIE	
PD0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ENCA0	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5IN	Input	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
PD1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ENCB0	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
	TB5OUT	Output	FT1	0/1	1	PDFR2	0/1	0/1	0/1	0
PD2			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ENCZ0	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
PD3			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INT9	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
PD4			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	SC2SCLK	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1
		Output	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
	SC2CTS	Input	FT1	0/1	0	PDFR2	0/1	0/1	0/1	1
PD5			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	SC2TXD	Output	FT1	0/1	1	PDFR1	0/1	0/1	0/1	0
PD6			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	SC2RXD	Input	FT1	0/1	0	PDFR1	0/1	0/1	0/1	1

11.1.2.5 PORT E

11-5 E

PO RT	Input/Output	PORT Type							
			PEDATA	PECR	PEFRn	PEOD	PEPUP	PEPDN	PEIE
PE0			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	SC0TXD	Output	0/1	1	PEFR1	0/1	0/1	0/1	0
	TB8OUT	Output	0/1	1	PEFR2	0/1	0/1	0/1	0
PE1			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	SC0RXD	Input	0/1	0	PEFR1	0/1	0/1	0/1	1
	TB8IN	Input	0/1	0	PEFR2	0/1	0/1	0/1	1
PE2			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	SC0SCLK	Input	0/1	0	PEFR1	0/1	0/1	0/1	1
		Output	0/1	1	PEFR1	0/1	0/1	0/1	0
	SC0CTS	Input	0/1	0	PEFR2	0/1	0/1	0/1	1
PE3			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	TB4OUT	Output	0/1	1	PEFR1	0/1	0/1	0/1	0
PE4			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	INT5	Input	0/1	0	0	0/1	0/1	0/1	1
	TB2IN	Input	0/1	0	PEFR1	0/1	0/1	0/1	1
PE5			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	TB2OUT	Output	0/1	1	PEFR1	0/1	0/1	0/1	0
PE6			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	INT6	Input	0/1	0	0	0/1	0/1	0/1	1
	TB3IN	Input	0/1	0	PEFR1	0/1	0/1	0/1	1
PE7			0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0
	INT7	Input	0/1	0	0	0/1	0/1	0/1	1
	TB3OUT	Output	0/1	1	PEFR1	0/1	0/1	0/1	0

11.1.2.6 PORT F

11-6 F

PORT	Input/Output	PORT Type								
			PFDATA	PFCR	PFFRn	PFOD	PFPUP	PFPDN	PFIE	
PF0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TB7IN	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
PF1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	TB7OUT	Output	FT1	0/1	1	PFFR1	0/1	0/1	0/1	0
PF2			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ENCA1	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3SCLK	Input	FT1	0/1	0	PFFR2	0/1	0/1	0/1	1
		Output	FT1	0/1	1	PFFR2	0/1	0/1	0/1	0
	SC3CTS	Input	FT1	0/1	0	PFFR3	0/1	0/1	0/1	1
PF3			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ENCB1	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3TXD	Output	FT1	0/1	1	PFFR2	0/1	0/1	0/1	0
PF4			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ENCZ1	Input	FT1	0/1	0	PFFR1	0/1	0/1	0/1	1
	SC3RXD	Input	FT1	0/1	0	PFFR2	0/1	0/1	0/1	1

) PF0 "Low" Pull-up BOOT  
 ) "0" PF0 "1" BOOT ROM (

11.1.2.7 PORT G

11-7 G

PORT	Input/Output	PORT Type								
			PGDATA	PGCR	PGFRn	PGOD	PGPUP	PGPDN	PGIE	
PG0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	UO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	XO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG2			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	VO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG3			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	YO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG4			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	WO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG5			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	ZO1	Output	FT2	0/1	1	PGFR1	0/1	0/1	0/1	0
PG6			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	EMG1	Input	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1
PG7			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	OVV1	Input	FT1	0/1	0	PGFR1	0/1	0/1	0/1	1

11.1.2.8 PORT H

11-8 H

PORT	Input/Output	PORT Type								
			PHDATA	PHCR	PHFRn	PHOD	PHPUP	PHPDN	PHIE	
PH0			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA0	Input	FT5	0/1	0	0/1	0	0	0	
	INT0	Input	FT4	0/1	0	0/1	0	0/1	1	
PH1			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA1	Input	FT5	0/1	0	0/1	0	0	0	
	INT1	Input	FT4	0/1	0	0/1	0	0/1	1	
PH2			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA2	Input	FT5	0/1	0	0/1	0	0	0	
	INT2	Input	FT4	0/1	0	0/1	0	0/1	1	
PH3			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA3	Input	FT5	0/1	0	0/1	0	0	0	
PH4			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA4	Input	FT5	0/1	0	0/1	0	0	0	
PH5			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA5	Input	FT5	0/1	0	0/1	0	0	0	
PH6			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA6	Input	FT5	0/1	0	0/1	0	0	0	
PH7			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINA7	Input	FT5	0/1	0	0/1	0	0	0	

11.1.2.9 PORT J

11-9 J

PORT	Input/Output	PORT Type								
			PJDATA	PJCR	PJFRn	PJOD	PJPUP	PJPDN	PJIE	
PJ0			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB3	Input	FT5	0/1	0		0/1	0	0	
PJ1			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB4	Input	FT5	0/1	0		0/1	0	0	
PJ2			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB5	Input	FT5	0/1	0		0/1	0	0	
PJ3			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB6	Input	FT5	0/1	0		0/1	0	0	
PJ4			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB7	Input	FT5	0/1	0		0/1	0	0	
PJ5			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB8	Input	FT5	0/1	0		0/1	0	0	
PJ6			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB9	Input	FT5	0/1	0		0/1	0	0	
	INTC	Input	FT4	0/1	0		0/1	0/1	1	
PJ7			0	0		0	0	0	0	
	Input Port	Input	0	0		0/1	0/1	0/1	1	
	Output Port	Output	0	1		0/1	0/1	0/1	0	
	AINB10	Input	FT5	0/1	0		0/1	0	0	
	INTD	Input	FT4	0/1	0		0/1	0/1	1	

11.1.2.10 PORT K

11-10 K

PORT	Input/Output	PORT Type								
			PKDATA	PKCR	PKFRn	PKOD	PKPUP	PKPDN	PKIE	
PK0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INTE	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TRACEDATA3	Output	FT1	0/1	1	PKFR1	0/1	0/1	0/1	0
PK1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INTF	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	TRACEDATA2	Output	FT1	0/1	1	PKFR1	0/1	0/1	0/1	0

11.1.2.11 PORT L

11-11 L

PORT	Input/Output	PORT Type								
			PLDATA	PLCR	PLFRn	PLOD	PLPUP	PLPDN	PLIE	
PL0			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INTB	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	CA_TX	Output	FT1	0/1	1	PLFR1	0/1	0/1	0/1	0
PL1			0	0	0	0	0	0	0	0
	Input Port	Input	0/1	0	0	0/1	0/1	0/1	0/1	1
	Output Port	Output	0/1	1	0	0/1	0/1	0/1	0/1	0
	INTA	Input	FT4	0/1	0	0	0/1	0/1	0/1	1
	CA_RX	Input	FT1	0/1	0	PLFR1	0/1	0/1	0/1	1

11.1.2.12 PORT N

11-12 N

PORT	Input/Output	PORT Type								
			PNDATA	PNCR	PNFRn	PNOD	PNPUP	PNPDN	PNIE	
PN0			0	0	0	0	0	0	0	0
Input Port	Input		0/1	0	0	0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1	0	0/1	0/1	0/1	0/1	0
SB0SDA	I/O	FT1	0/1	1	PNFR2	1	0/1	0/1	0/1	1
PN1			0	0	0	0	0	0	0	0
Input Port	Input		0/1	0	0	0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1	0	0/1	0/1	0/1	0/1	0
TB9OUT	Output		0/1	1	PNFR1	0/1	0/1	0/1	0/1	0
SB0SCL	I/O	FT1	0/1	1	PNFR2	1	0/1	0/1	0/1	1
PN2			0	0	0	0	0	0	0	0
Input Port	Input		0/1	0	0	0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1	0	0/1	0/1	0/1	0/1	0
TB9IN	Input	FT1	0/1	0	PNFR1	0/1	0/1	0/1	0/1	1

11.1.2.13 PORT P

11-13 P

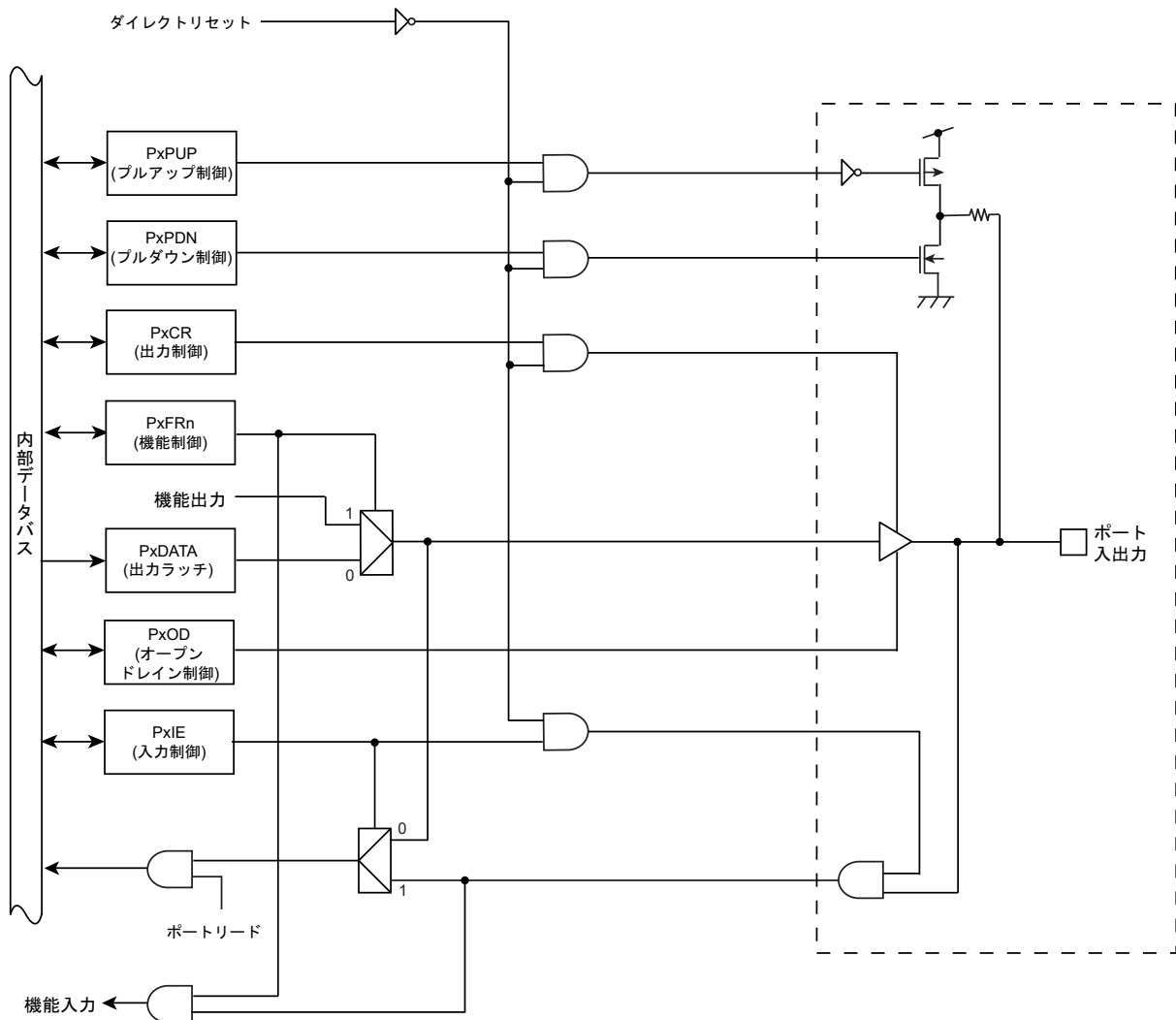
PORT	Input/Output	PORT Type								
			PPDATA	PPCR	PPFRn	PPOD	PPPUP	PPPDN	PPIE	
PP0			0	0		0	0	0	0	0
Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1		0/1	0/1	0/1	0/1	0
AINA8	Input	FT5	0/1	0		0/1	0	0	0	0
PP1			0	0		0	0	0	0	0
Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1		0/1	0/1	0/1	0/1	0
AINA9	Input	FT5	0/1	0		0/1	0	0	0	0
AINB0	Input	FT5	0/1	0		0/1	0	0	0	0
PP2			0	0		0	0	0	0	0
Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1		0/1	0/1	0/1	0/1	0
AINA10	Input	FT5	0/1	0		0/1	0	0	0	0
AINB1	Input	FT5	0/1	0		0/1	0	0	0	0
PP3			0	0		0	0	0	0	0
Input Port	Input		0/1	0		0/1	0/1	0/1	0/1	1
Output Port	Output		0/1	1		0/1	0/1	0/1	0/1	0
AINA11	Input	FT5	0/1	0		0/1	0	0	0	0
AINB2	Input	FT5	0/1	0		0/1	0	0	0	0

## 11.2

ポートには、以下のタイプがあります。それぞれの回路図を次ページから示します。図中の点線は「ポート部等価回路図」で記されている等価回路の範囲を示します。

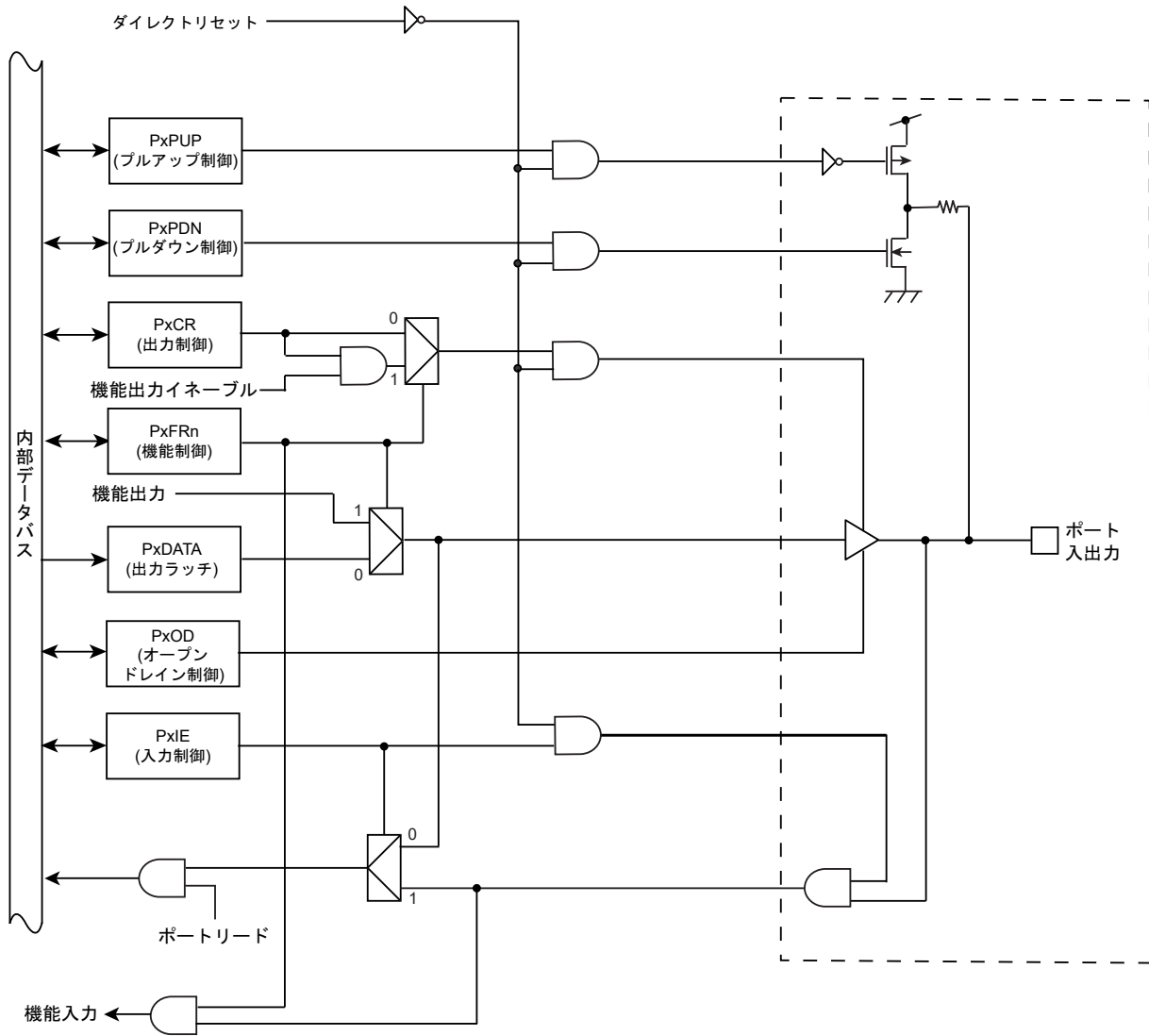
回路図内の"ダイレクトリセット"は、コールドリセット時に有効になります。

### 11.2.1 FT1



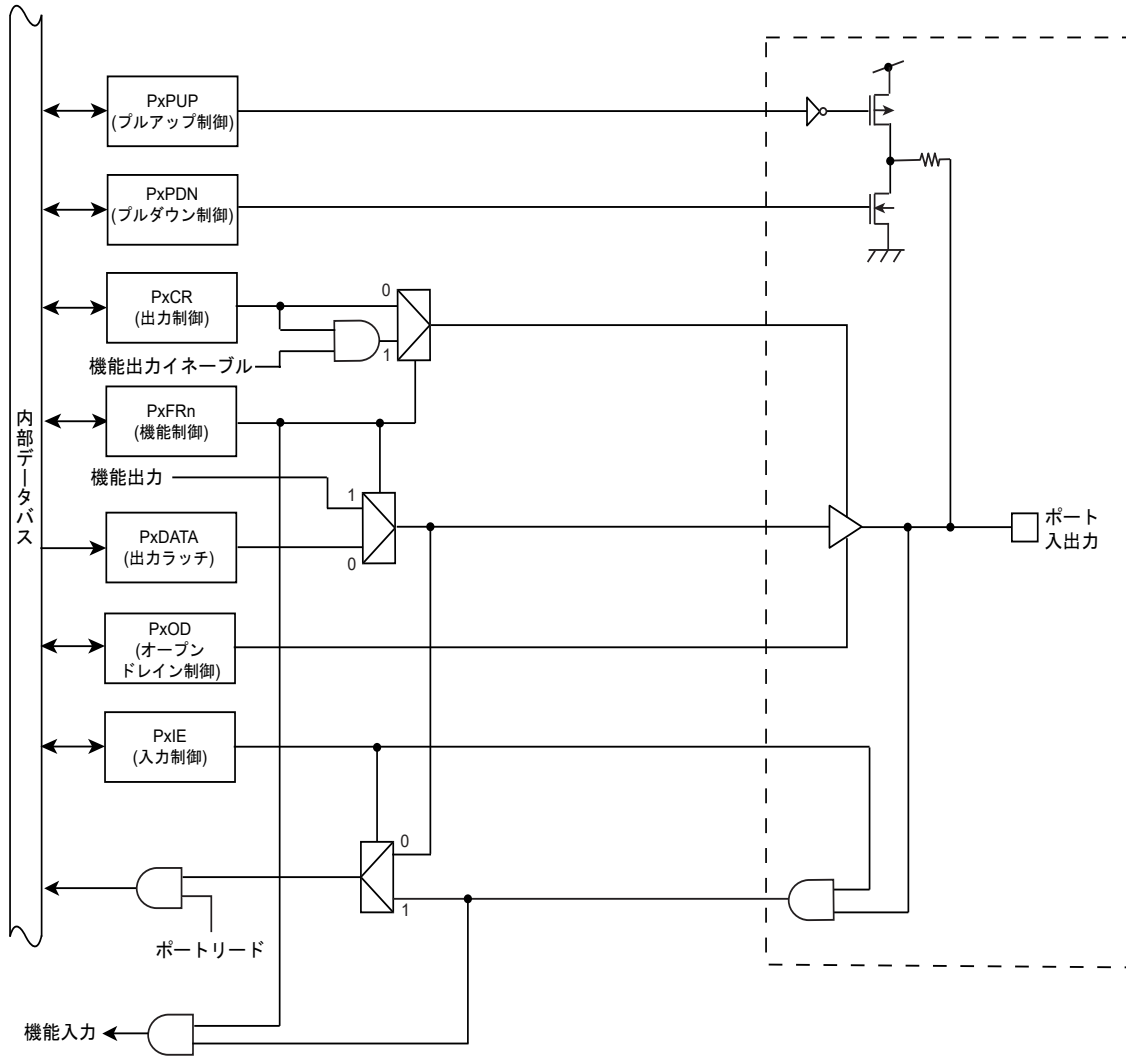
11-1 Port Type FT1

### 11.2.2 FT2



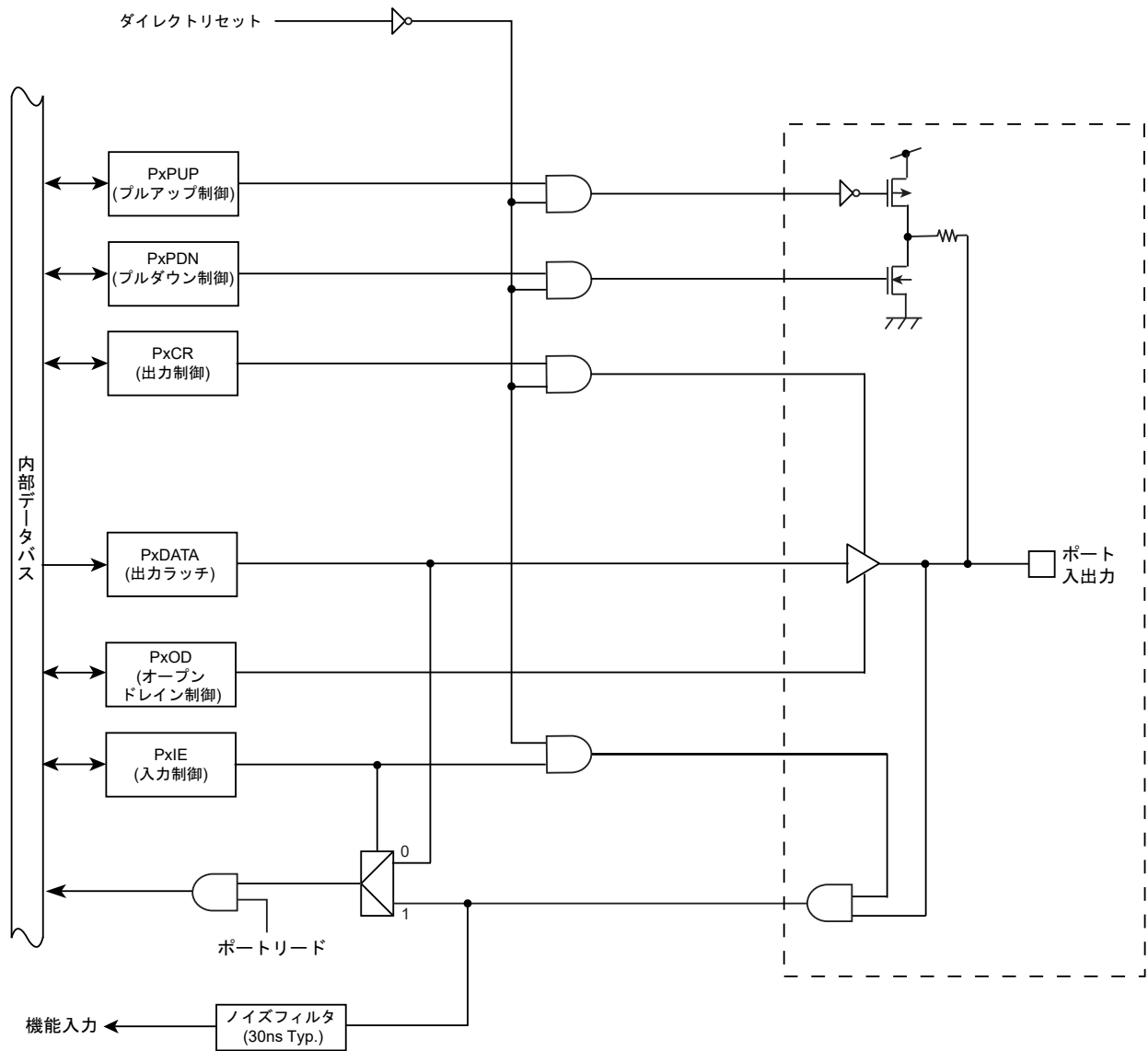
11-2 Port Type FT2

11.2.3 FT3



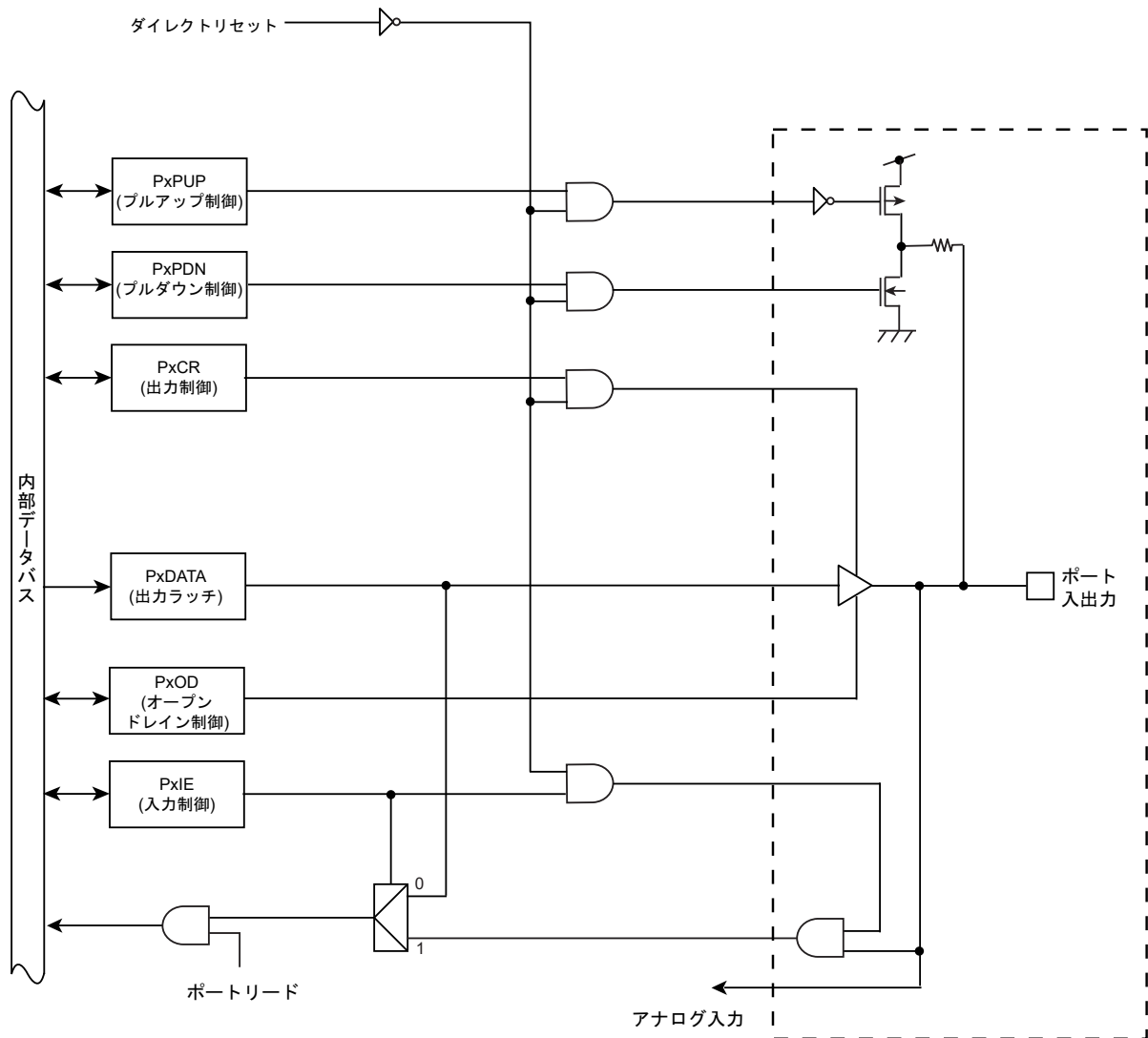
11-3 Port Type FT3

11.2.4 FT4



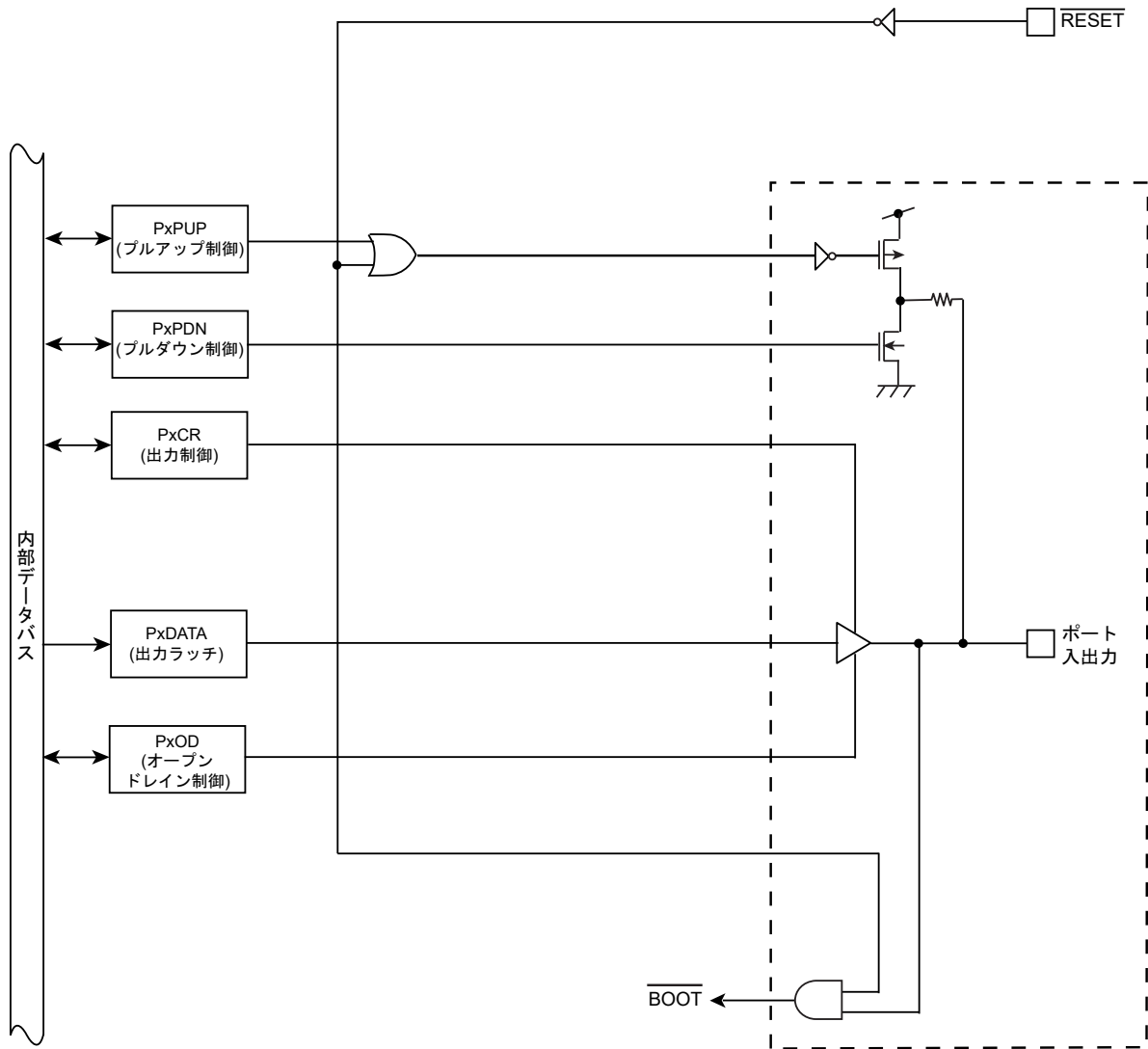
11-4 Port Type FT4

11.2.5 FT5



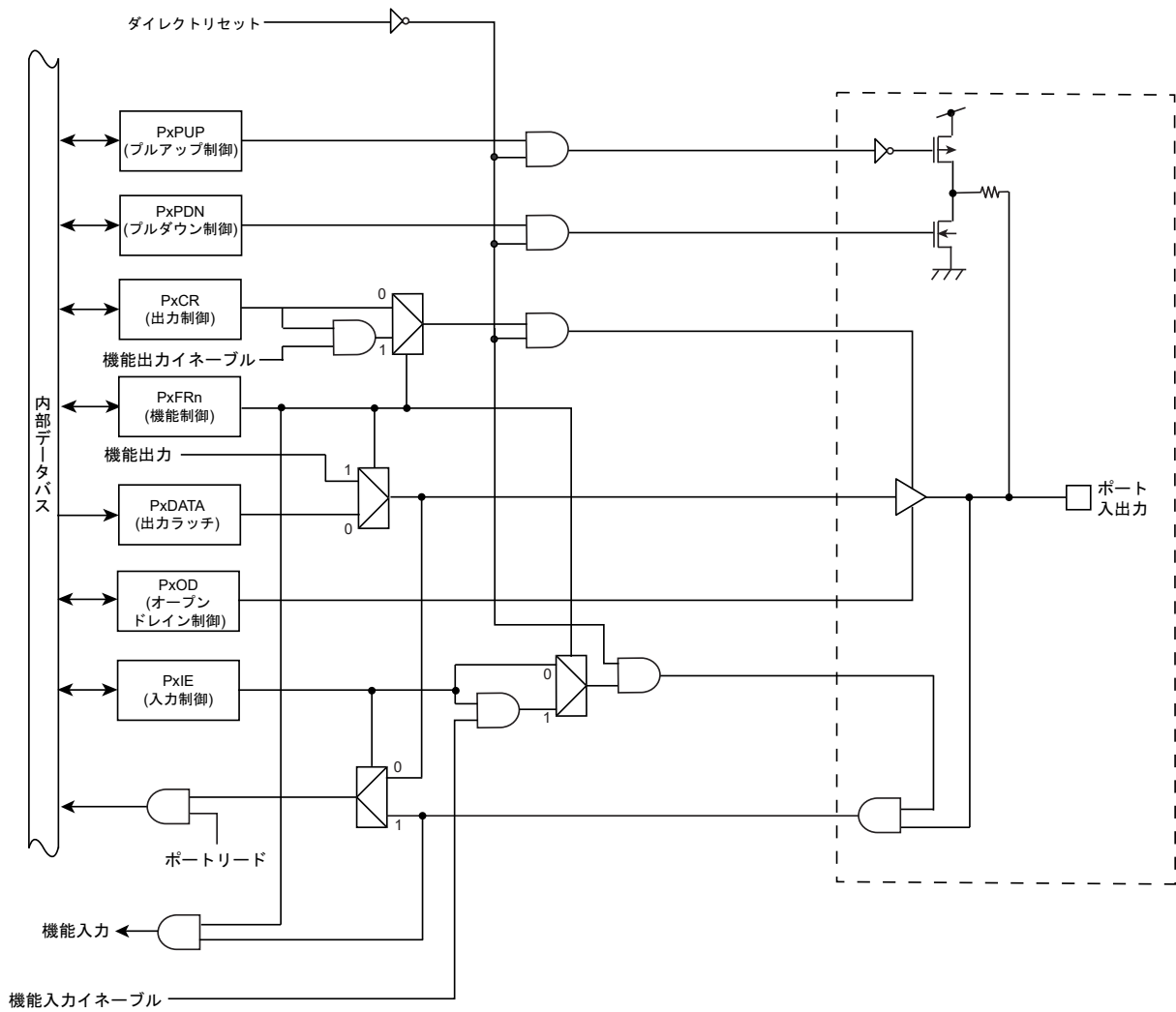
11-5 Port Type FT5

11.2.6 FT6



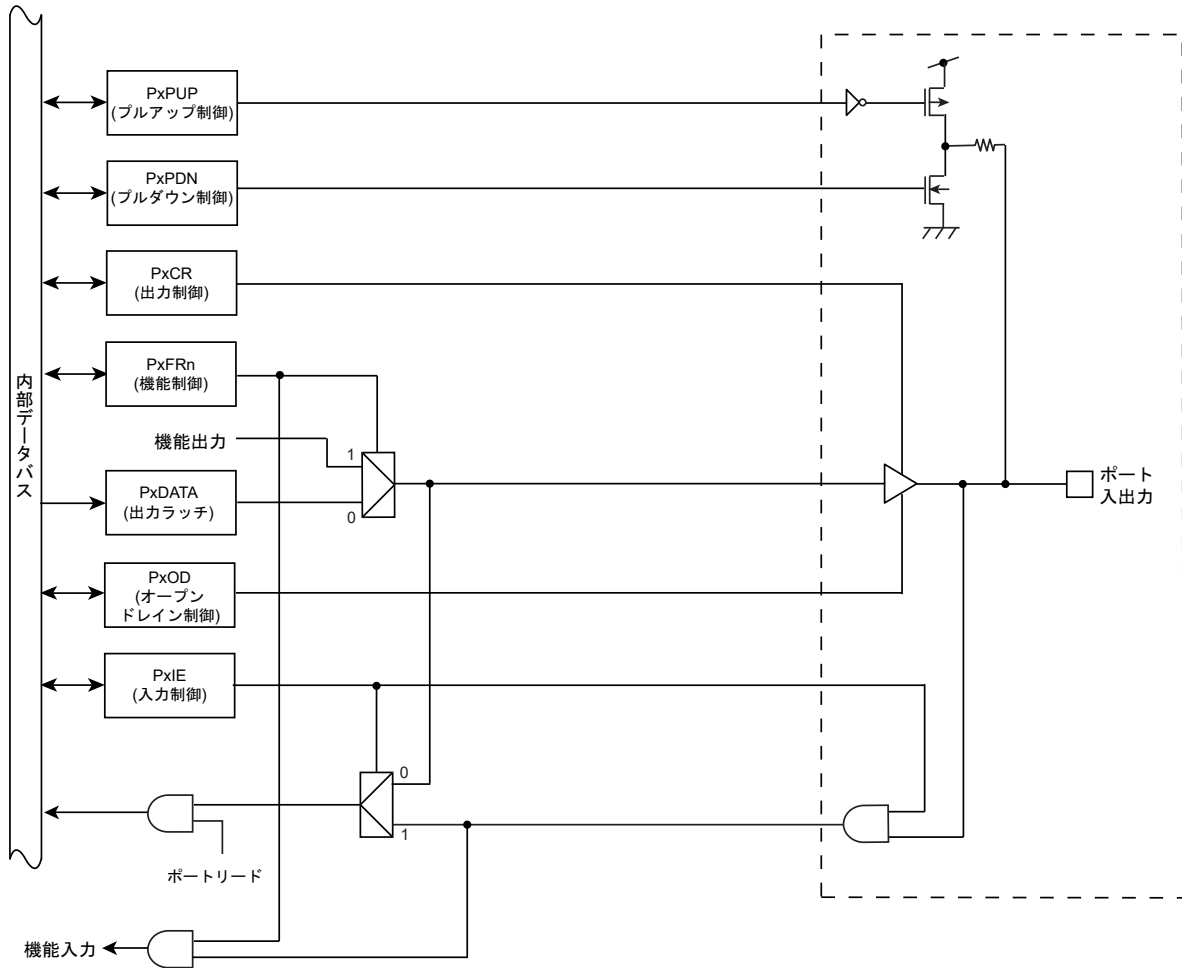
11-6 Port Type FT6

11.2.7 FT7



11-7 Port Type FT7

11.2.8 FT8



11-8 Port Type FT8

## 12      16      /      (TMRB)

## 12.1

TMRB は、次の機能をもっています。

- ・ 16 ビットインタバルタイマモード
- ・ 16 ビットイベントカウンタモード
- ・ 16 ビットプログラマブル矩形波出力 (PPG) モード
- ・ 外部トリガプログラマブル矩形波出力 (PPG) モード

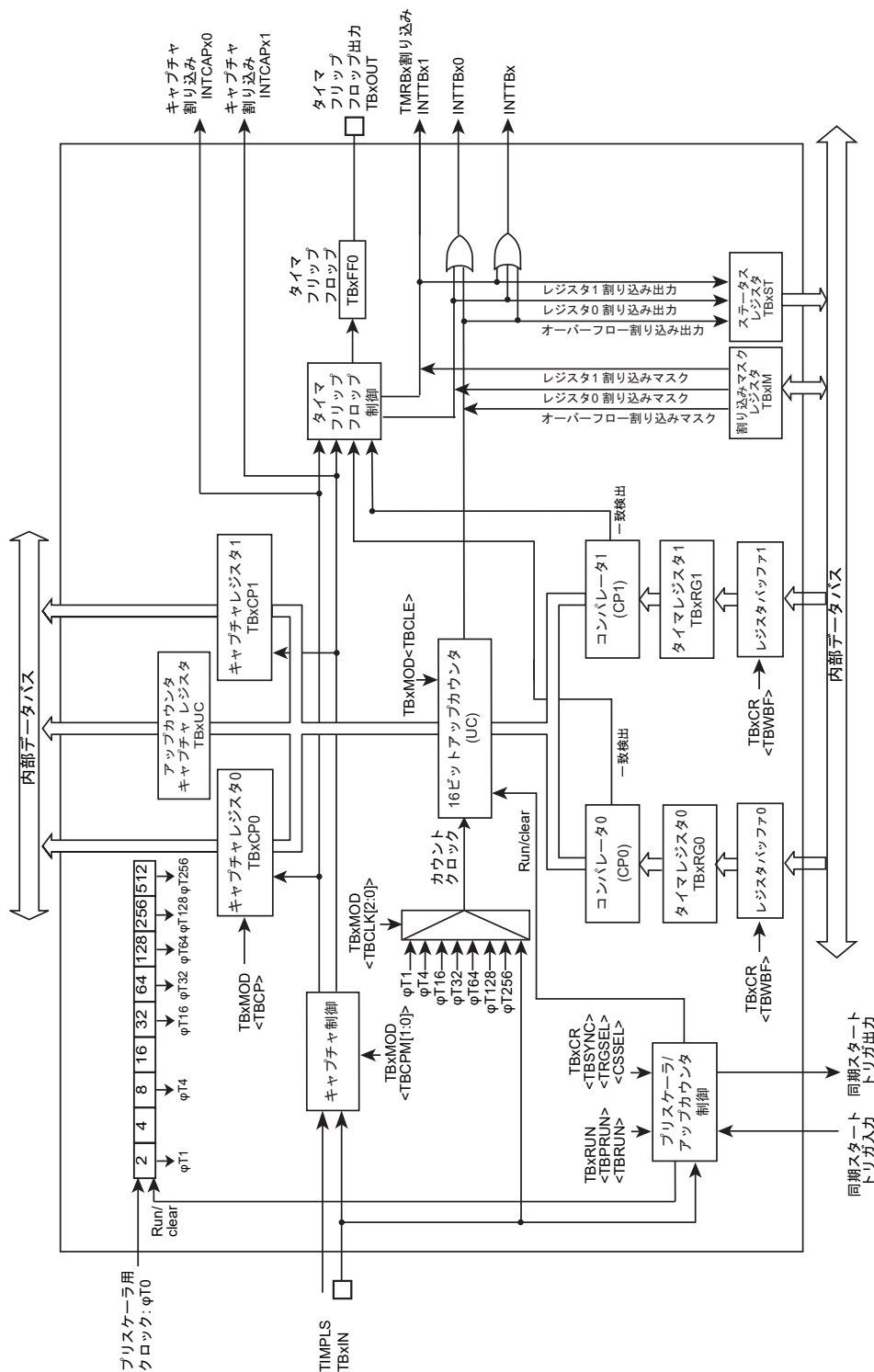
また、キャプチャ機能を利用することで、次のような用途に使用することができます。

- ・ 外部トリガからのワンショットパルス出力
- ・ パルス幅測定

以下の説明中、"x"はチャンネル番号を表します。

## 12.2

各チャネルは、主に 16 ビットアップカウンタ、16 ビットタイマレジスタ 2 本(ダブルバッファ構造)、16 ビットのキャプチャレジスタ 2 本、コンパレータ 2 個、および、キャプチャ入力制御、タイマフリップフロップとその制御回路で構成されています。タイマの動作モードやタイマフリップフロップはレジスタで制御されます。



12-1 TMRBx

## 12.3

## 12.3.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)
	TBxEN	0x0000
RUN	TBxRUN	0x0004
	TBxCR	0x0008
	TBxMOD	0x000C
	TBxFFCR	0x0010
	TBxST	0x0014
	TBxIM	0x0018
	TBxUC	0x001C
0	TBxRG0	0x0020
1	TBxRG1	0x0024
0	TBxCP0	0x0028
1	TBxCP1	0x002C

12.3.2 TBxEN( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBEN	TBHALT	-	-	-	-	-	-
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	TBEN	R/W	TMRBx 0: 1: TMRBx ( TMRBx TBxEN TMRBx TMRBx TMRBx ("1") TMRBx
6	TBHALT	R/W	0: 1: "0"
5-0		R	"0"

12.3.3 TBxRUN(RUN )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	TBPRUN	-	TBRUN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-3		R	"0"
2	TBPRUN	R/W	0: & 1:
1		R	"0"
0	TBRUN	R/W	0: & 1:

- 1) <TBRUN>=1
- 2) (<TBRUN>="0") <TBRUN>=1 <TBUC[15:0]>

### 12.3.4 TBxCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBWBF	-	TBSYNC	-	I2TB	-	TRGSEL	CSSEL
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	TBWBF	R/W	0: 1:
6		R/W	"0"
5	TBSYNC	R/W	0: ( ) 1:
4		R	"0"
3	I2TB	R/W	IDLE 0: 1:
2		R/W	"0"
1	TRGSEL	R/W	0: 1:  TBxIN
0	CSSEL	R/W	0: 1:

1) TBxCR

2)

<TBRUN>=<TBPRUN>=1

<CSSEL> <TRGSEL>

12.3.5 TBxMOD( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	TBCP	TBCPM		TBCLE	TBCLK		
	0	1	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	TBRSWR	R/W	0,1 0: 0 1 1: 0
6	TBCP	W	0: Don't care 1: 0 (TBxCP0) "1"
5-4	TBCPM[1:0]	R/W	00: 0 (TBxCP0) 01: TBxIN↑ TBxIN 0 (TBxCP0) 10: TBxIN↑ TBxIN↓ TBxIN 1 (TBxCP1) 11: TIMPLS↑ TIMPLS↓ TIMPLS 0(TBxCP0) TIMPLS 1(TBxCP1)
3	TBCLE	R/W	0: "0" "1" (TBxRG1) 1:
2-0	TBCLK[2:0]	R/W	TMRBx 000: TBxIN 001: T1 010: T4 011: T16 100: T32 101: T64 110: T128 111: T256

) TBxMOD

12.3.6 TBxFFCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	TBC1T1	TBC0T1	TBE1T1	TBE0T1	TBFF0C	
	1	1	0	0	0	0	1	1

Bit	Bit Symbol	Type	
31-8		R	"0"
7-6		R	"1"
5	TBC1T1	R/W	TBxCP1 TBxFF0 0: 1: "1" 1 (TBxCP1)
4	TBC0T1	R/W	TBxCP0 TBxFF0 0: 1: "1" 0 (TBxCP0)
3	TBE1T1	R/W	TBxRG1 TBxFF0 0: 1: "1" 1 (TBxRG1)
2	TBE0T1	R/W	TBxRG0 TBxFF0 0: 1: "1" 0 (TBxRG0)
1-0	TBFF0C[1:0]	R/W	TBxFF0 00: Invert TBxFF0 ( ) 01: Set TBxFF0 "1" 10: Clear TBxFF0 "0" 11: Don't care * "11"

) TBxFFCR

12.3.7 TBxST( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	INTTBOF	INTTB1	INTTB0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-3		R	"0"
2	INTTBOF	R	0: 1: 1
1	INTTB1	R	(TBxRG1) 0: 1: TBxRG1 1 (TBxRG1) 1
0	INTTB0	R	(TBxRG0) 0: 1: TBxRG0 0 (TBxRG0) 1

1) TBxIM

CPU

2)

12.3.8 TBxIM( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	TBIMOF	TBIM1	TBIM0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-3		R	"0"
2	TBIMOF	R/W	0: 1:
1	TBIM1	R/W	(TBxRG1) 0: 1: 1 (TBxRG1)
0	TBIM0	R/W	(TBxRG0) 0: 1: 0 (TBxRG0)

) TBxIM

TBxST

12.3.9 TBxUC( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TBUC[15:0]	R	TBxUC

) TBxUC

12.3.10 TBxRG0( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBRG0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBRG0							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TBRG0[15:0]	R/W	

12.3.11 TBxRG1( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBRG1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBRG1							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TBRG1[15:0]	R/W	

12.3.12 TBxCP0( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBCP0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBCP0							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TBCP0[15:0]	R	

12.3.13 TBxCP1( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TBCP1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBCP1							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TBCP1[15:0]	R	

## 12.4

### 12.4.1

アップカウンタ UC のソースクロックを生成する 4 ビットのプリスケータです。

プリスケータへの入力クロック  $\phi T0$  は CG 部の CGSYSCR<PRCK[2:0]>で選択された fperiph/1, fperiph/2, fperiph/4, fperiph/8, fperiph/16, fperiph/32 のいずれかのクロックです。このペリフェラルクロック fperiph は CG 部の CGSYSCR<FPSEL>で選択したクロック fgear またはクロックギア分周前のクロック fc のいずれかのクロックです。

プリスケータは TBxRUN<TBPRUN>により動作/停止の設定をします。"1"をライトするとカウント開始し"0"をライトするとクリアされ停止します。

### 12.4.2 (UC)

16 ビットのバイナリカウンタです。

#### 12.4.2.1

ソースクロックは TBxMOD<TBCLK[2:0]>で設定することができます。

プリスケータ出力クロック  $\phi T1$ ,  $\phi T4$ ,  $\phi T16$ ,  $\phi T32$ ,  $\phi T64$ ,  $\phi T128$ ,  $\phi T256$ 、または、TBxIN 入力のいずれかを選択できます。

#### 12.4.2.2

カウンタのスタート方法には、ソフトスタート、外部トリガスタート、同期スタートがあります。

##### 1. ソフトスタート

TBxRUN<TBRUN>に"1"を設定することでカウントを開始します。"0"でカウント停止と同時にアップカウンタのクリアを行います。

##### 2. 外部トリガスタート

外部トリガカウントスタートモードでは、外部信号でタイマのカウントスタートが可能となります。

TBxCR<CSSEL>に"1"を設定することで外部トリガスタートモードとなります。この状態で、<TBRUN>に"1"を設定するとトリガ待ち状態となり、TBxIN0TBxINの立ち上がりまたは立ち下がりによってカウントを開始します。

TBxCR<TRGSEL>ビットの設定により、外部トリガのエッジ切り替えを行います。

- ・ <TRGSEL>="0" : TBxIN の立ち上がりエッジが選択されます。
- ・ <TRGSEL>="1" : TBxIN の立ち下がりエッジが選択されます。

TBxRUN<TBRUN>に"0"を設定することでカウント停止と同時にアップカウンタのクリアを行います。

##### 3. 同期スタート

タイマ同期モードでは、タイマ間のスタートの同期を取ることが可能となります。PPG 出力モードにてタイマ同期モードを使用することによりモータ等の駆動に応用が可能です。

製品によってマスタとなるチャンネルとスレーブとなるチャンネルの組み合わせは決まっています。本製品でのマスタとスレーブの組み合わせは「製品情報」の章を参照ください。

TBxCR<TBSYNC> ビットの設定により、同期モードの切り替えを行います。スレーブチャンネルの<TBSYNC>ビットに"1"を設定するとマスタチャンネルのソフトウェアまたは外部トリガによるスタートに同期してカウント開始および停止します。スレーブチャンネルの TBxRUN <TBPRUN, TBRUN> ビットの設定は不要です。マスタチャンネルの<TBSYNC> ビットは"0"を設定してください。

なお、外部トリガカウントモードとタイマ同期モードが同時に設定されている場合は、タイマ同期モードが優先されます。

### 12.4.2.3

#### 1. コンペア一致時

TBxMOD<TBCLE>="1"に設定することで、TBxRG1 とのコンペア一致とともにカウンタのクリアをすることができます。TBxMOD<TBCLE>="0"に設定するとカウンタはフリーランニングカウンタとして動作します。

#### 2. カウンタ停止時

TBxRUN<TBRUN>="0"に設定すると、カウンタが停止するとともにクリアされず。

### 12.4.2.4

アップカウンタ UC がオーバーフローすると、オーバーフロー割り込み INTTBx0 が発生します。

## 12.4.3 (TBxRG0, TBxRG1)

アップカウンタ UC と比較する値を設定するレジスタで、2 本内蔵されています。タイマレジスタに設定された値とアップカウンタの値をコンパレータで比較し、一致するとコンパレータが一致検出信号を出力します。

TBxRG0/1 はダブルバッファ構成になっており、レジスタバッファとペアになっています。初期状態では、ダブルバッファはディセーブルです。

ダブルバッファのイネーブル/ディセーブル制御は TBxCR<TBWBF> によって行います。<TBWBF>="0" のときディセーブル、<TBWBF>="1" のときイネーブルとなります。ダブルバッファイネーブル時、UC と TBxRG1 との一致時にレジスタバッファ 0/1 からタイマレジスタ TBxRG0/1 へデータ転送が行われます。また、ダブルバッファがイネーブルでもカウンタが停止しているときはシングルバッファ動作となり、TBxRG0/1 に直接値を書き込むことができます。

## 12.4.4

アップカウンタ UC の値をキャプチャレジスタ TBxCP0, TBxCP1 にラッチするタイミングを制御する回路です。キャプチャレジスタのラッチタイミングは、TBxMOD<TBxCPM[1:0]>で設定します。

また、ソフトウェアによってもアップカウンタ UC の値をキャプチャレジスタへ取り込むことができ、TBxMOD<TBxCP>に"0"を書き込むたびに、その時点の UC の値をキャプチャレジスタ TBxCP0 へキャプチャします。

#### 12.4.5 (TBxCP0, TBxCP1)

アップカウンタ UC の値をキャプチャするレジスタです。

#### 12.4.6 (TBxUC)

キャプチャ制御回路によるキャプチャ機能のほかに、TBxUC レジスタを読み出すことにより、アップカウンタの現在のカウント値をキャプチャすることができます。

#### 12.4.7 (CP0, CP1)

アップカウンタ UC と、タイマレジスタ TBxRG0, TBxRG1 への設定値とを比較し、一致を検出します。一致すると、INTTBx0 と INTTBx1 を発生します。

#### 12.4.8 (TBxFF0)

タイマフリップフロップ (TBxFF0) は、コンパレータからの一致信号、キャプチャレジスタへのラッチ信号によって反転するフリップフロップです。反転のディセーブル/イネーブルは、TBxFFCR<TBC1T1, TBC0T1, TBE1T1, TBE0T1>によって設定できます。

リセット後、TBxFF0 の値は不定となります。TBxFFCR<TBFF0C[1:0]>に "00" を書き込むことで反転、"01" を書き込むことで "1" にセット、"10" を書き込むことで "0" にクリアすることが可能です。

TBxFF0 の値は、タイマ出力端子 TBxOUT 端子へ出力することができます。タイマ出力を行う場合、あらかじめ該当するポートの設定を行う必要があります。

#### 12.4.9 (INTCAPx0, INTCAPx1)

キャプチャレジスタ TBxCP0, TBxCP1 にラッチするタイミングで割り込み INTCAPx0, INTCAPx1 をそれぞれ発生します。割り込みの設定は CPU で行います。

## 12.5

### 12.5.1 16

一定周期の割り込みを発生させる場合、タイマレジスタ TBxRG1 にインタバル時間を設定し、INTTBx1 割り込みを発生します。

	7	6	5	4	3	2	1	0	
TBxEN	1	X	X	X	X	X	X	X	TMRBx
TBxRUN	X	X	X	X	X	0	X	0	TMRBx
	*	*	*	*	*	*	*	*	INTTBx1 "1"
TBxFFCR	X	X	0	0	0	0	1	1	TBxFF0
TBxMOD	0	1	0	0	1	*	*	*	
	(*** = 001, 010, 011, 100, 101, 110, 111)								
TBxRG1	*	*	*	*	*	*	*	*	(16 )
	*	*	*	*	*	*	*	*	
TBxRUN	*	*	*	*	*	1	X	1	TMRBx

) X; Don't care -; No change

### 12.5.2 16

入力クロックを外部クロック(TBxIN 端子入力)にすることでイベントカウンタにすることができます。

アップカウンタは TBxIN 端子入力の立ち上がりエッジでカウントアップします。ソフトウェアキャプチャを行い、キャプチャ値をリードすることでカウント値を読むことができます。

	7	6	5	4	3	2	1	0	
TBxEN	1	X	X	X	X	X	X	X	TMRBx
TBxRUN	X	X	X	X	X	0	X	0	TMRBx
PORT									TBxIN
TBxFFCR	X	X	0	0	0	0	1	1	TBxFF0
TBxMOD	0	1	0	0	0	0	0	0	TBxIN
TBxRUN	*	*	*	*	*	1	X	1	TMRBx
TBxMOD	0	0	0	0	0	0	0	0	

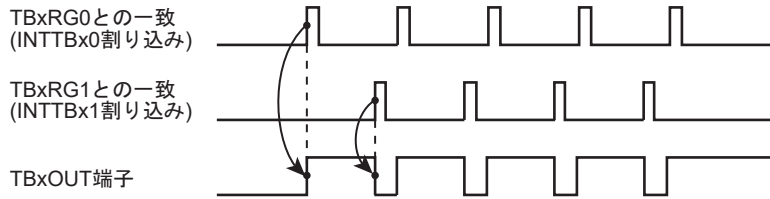
) X; Don't care -; No change

### 12.5.3 16 PPG ( )

任意周波数, 任意デューティの矩形波 (プログラマブル矩形波) を出力することができます。出力パルスは、ローアクティブ, ハイアクティブどちらでも可能です。

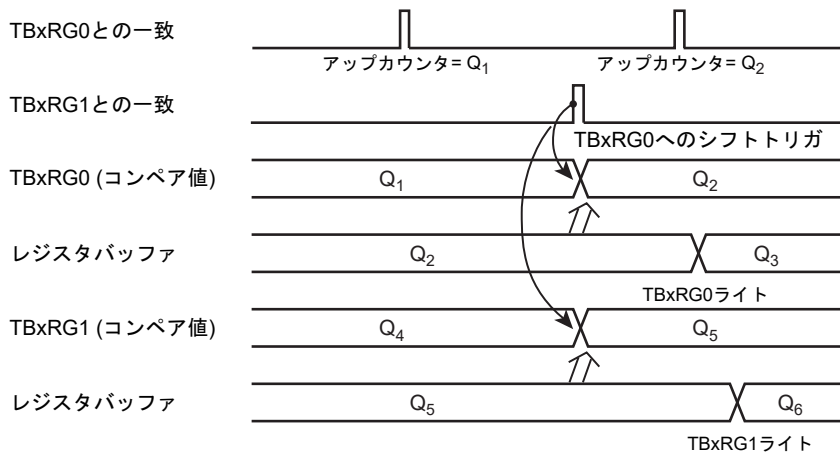
アップカウンタ (UC) とタイマレジスタ (TBxRG0, TBxRG1) への設定値との一致によりタイマフリップフロップ (TBxFF0) の反転トリガをかけることで、プログラマブル矩形波を TBxOUT 端子より出力することができます。ただし、TBxRG0 と TBxRG1 の設定値は次の条件を満たす必要があります。

TBxRG0 設定値 < TBxRG1 設定値



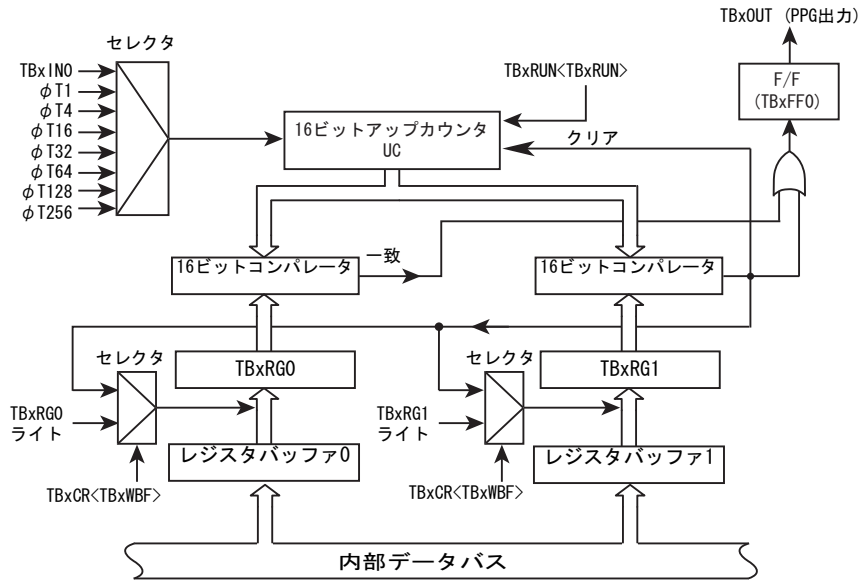
12-2 (PPG)

このモードでは、TBxRG0 のダブルバッファをイネーブルにすることにより、TBxRG1 との一致で、レジスタバッファ 0 の値が TBxRG0 へシフトインされます。これにより、小さいデューティへの対応が容易に行えます。



12-3

このモードのブロック図を示します。



12-4 16 PPG

16 ビット PPG 出力モード時の各レジスタは、次のように設定します。

	7	6	5	4	3	2	1	0	
TBxEN	1	X	X	X	X	X	X	X	TMRBx
TBxRUN	X	X	X	X	X	0	X	0	TMRBx
TBxCR	0	0	0	X		0	0	0	
TBxRG0	*	*	*	*	*	*	*	*	(16 )
TBxRG1	*	*	*	*	*	*	*	*	(16 )
TBxCR	1	0	0	X		0	0	0	TBxRG0 (INTTBx0 / )
TBxFFCR	X	X	0	0	1	1	1	0	TBxFF0 TBxRG0, TBxRG1 TBxFF0 "0"
TBxMOD	0	1	0	0	1	*	*	*	
	(***) = 001, 010, 011, 100, 101, 110, 111)								UC TBxRG1
PORT									TBxOUT
TBxRUN	*	*	*	*	*	1	X	1	TMRBx

) X; Don't care  
-; No change

### 12.5.4 PPG( )

外部トリガカウントスタートを使用すると、小さいディレイでのワンショットパルス出力が可能です。

外部トリガカウントスタートを使用したワンショットパルス出力(ディレイあり) の例を以下に示します。

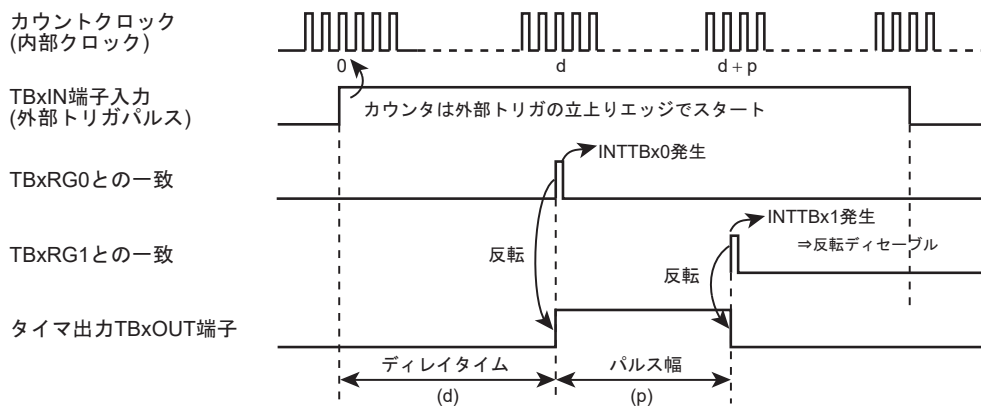
16 ビットアップカウンタ(UC)が停止状態(TBxRUN<TBRUN>=0)で、TBxIN 端子の立ち上がりでカウントアップするように設定しておきます(TBxCR[1:0] = "01")。タイマレジスタ(TBxRG0) には、ディレイタイム(d)を設定します。タイマレジスタ(TBxRG1)には TBxRG0 の値とワンショットパルスの幅(p)を加算した値(d+p)を設定します。

TBxFFCR<TBE1T1, TBE0T1>に"11"を設定し、UC と TBxRG0 との一致、および、TBxRG1 との一致より、タイマフリップフロップ(TBxFF0)が反転するようにトリガをイネーブルにします。

TBxRUN<TBRUN>を"1"にセットし、外部トリガパルスの立ち上がりで、カウントアップスタートを有効にします。

TBxIN 端子の立ち上がりでワンショットパルスが出力された後、INTTBx1 の割り込み処理でタイマフリップフロップ(TBxFF0)の反転をディセーブルにするか、TBxRUN<TBRUN>を"0"にクリアし、16 ビットアップカウンタの動作を停止します。

なお、文中の(d)、(p)は図 12-5 の d、p と対応しています。



12-5

( )

## 12.6

キャプチャ機能を利用することにより、次に示す例をはじめ、多くの応用が可能です。

1. 外部トリガパルスからのワンショットパルス出力
2. パルス幅測定

## 12.6.1

外部トリガパルスからのワンショットパルス出力は、次のように行います。

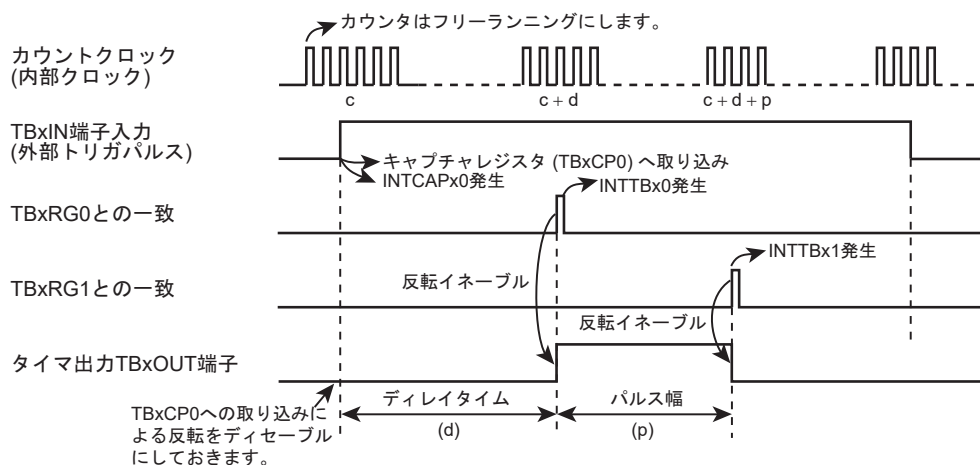
16ビットアップカウンタ UC をプリスケアラ出力クロックにてフリーランニングでカウントアップさせておきます。TBxIN 端子より外部トリガパルスを入力し、キャプチャ機能を用いて、外部トリガパルスの立ち上がりで、アップカウンタ値をキャプチャレジスタ(TBxCP0)に取り込みます。

外部トリガパルスの立ち上がり時、割り込み INTCAPx0 が発生するように CPU で設定します。この割り込みで、タイマレジスタ(TBxRG0)には、TBxCP0 の値(c)とディレイタイム(d)を加算した値(c+d)を設定します。

タイマレジスタ(TBxRG1)には、TBxRG0 の値とワンショットパルスのパルス幅(p)を加算した値(c+d+p)を設定します。(TBxRG1 の変更は次の一致までに完了してください)

さらに、タイマフリップフロップコントロールレジスタ(TBxFFCR<TBE1T1, TBE0T1>)に "11" を設定し、UC と TBxRG0 との一致、および、TBxRG1 との一致により、タイマフリップフロップ(TBxFF0)が反転するように、トリガイネーブルにします。ワンショットパルス出力後、INTTBx0/INTTBx1 の割り込み処理により、これをディセーブルに戻します。

なお、文中の(c), (d), (p)は、図 12-6 の c, d, p と対応しています。



12-6

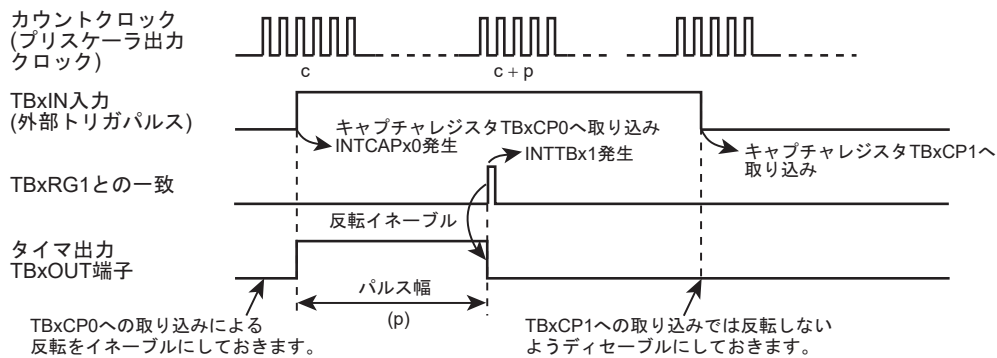
( )

TBxIN 入力の立ち上がりをトリガとして、3ms 後に 2ms 幅のワンショットパルスを出力する場合の設定例を以下に示します。ここではソースクロックに  $\Phi T1$  を使用しています。

	7	6	5	4	3	2	1	0	
[ ] TBxIN									
PORT									TBxIN
TBxEN	1	X	X	X	X	X	X	X	TMRBx
TBxRUN	X	X	X	X	X	0	X	0	TMRBx
TBxMOD	0	1	0	1	0	0	0	1	$\Phi T1$ TBxIN TBxCP0
TBxFFCR	X	X	0	0	0	0	1	0	TBxFF0
PORT									TBxOUT
	*	*	*	*	*	*	*	*	INTCAPx0 "1"
TBxRUN	*	*	*	*	*	1	X	1	TMRBx
[INTCAPx0 ]									
TBxRG0	*	*	*	*	*	*	*	*	$(TBxCP0 + 3\mu\sigma/\Phi T1)$
TBxRG1	*	*	*	*	*	*	*	*	$(TBxCP0 + (3+2)ms/\Phi T1)$
TBxFFCR	X	X			1	1			TBxRG0, TBxRG1 TBxFF0
TBxIM	X	X	X	X	X	1	0	1	TBxRG1
	*	*	*	*	*	*	*	*	INTTBx1 "1"
[INTTBx1 ]									
TBxFFCR	X	X			0	0			TBxFF0
	*	*	*	*	*	*	*	*	INTTBx "1"

) X; Don't care  
 -; No change

ディレイが不要な場合、TBxCP0 への取り込みによって TBxFF0 を反転させ、割り込み INTCAPx0 で TBxCP0 の値(c) にワンショットパルスの幅(p)を加算した値(c + p)を TBxRG1 に設定します。(TBxRG1 の変更は次の一致までに完了してください。) TBxFF0 は、TBxRG1 と UC の一致によって反転するように、反転イネーブルを選択します。また、INTTBx1 割り込みでこれをディセーブルに戻します。



12-7

( )

## 12.6.2

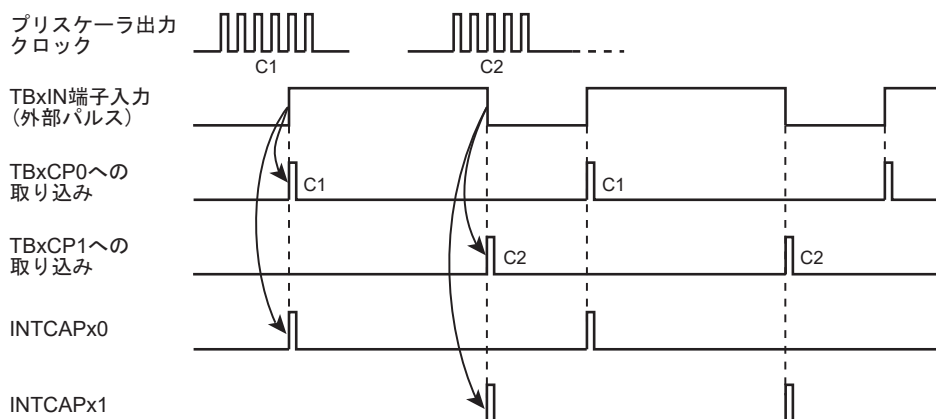
キャプチャ機能を用いて、外部パルスの"High"レベル幅を測定することができます。TBxIN 端子より外部パルスを入力し、アップカウンタ (UC) をプリスケータ出力クロックにてフリーランニングでカウントアップさせておきます。キャプチャ機能を用いて、外部パルスの立ち上がり/立ち下がり、それぞれのエッジでトリガをかけ、このときのアップカウンタ値をキャプチャレジスタ (TBxCP0, TBxCP1) に取り込みます。TBxIN 端子の立ち下がりにより、INTCAPx1 が発生するように CPU で設定します。

"High"レベルパルス幅は、TBxCP0 と TBxCP1 の差を求め、その値に内部クロックの周期をかけることにより、求めることができます。

例えば TBxCP0 と TBxCP1 の差が 100 で、プリスケータ出力クロックの周期が  $0.5 \mu\text{s}$  であれば、パルス幅は、 $100 \times 0.5 \mu\text{s} = 50 \mu\text{s}$  となります。

なお、クロックソースにより定まる UC の最大カウント時間を越えるパルス幅の測定を行う場合は、注意が必要です。この場合、ソフトウェアによる処理を行ってください。

また、外部パルスの"Low"レベル幅を測定することもできます。この場合、図 12-8 における、2 回目の INTCAPx0 割り込み処理により、1 回目の C2 と 2 回目の C1 の差に、プリスケータ出力クロックの周期をかけることにより、求めることができます。



12-8



## 13 4 FIFO (SIO/UART)

## 13.1

シリアルチャネル(SIO/UART)は次の動作モードを持っています。

- ・ 同期通信モード(I/O インタフェースモード)
- ・ 非同期通信モード(UART モード)

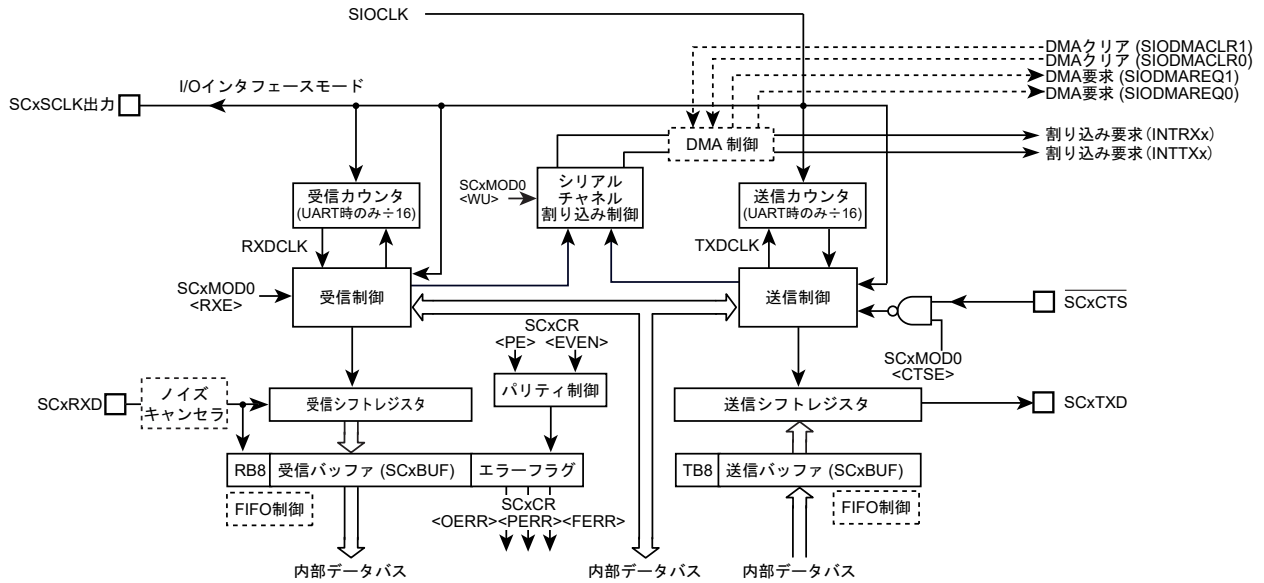
特長は以下のとおりです。

- ・ 転送クロック
  - プリスケーラでペリフェラルクロック( $\phi T0$ )を 1/1、1/2、1/4、1/8、1/16、1/32、1/64、1/128 分周
  - プリスケーラ出力クロックに対し、1~16 分周が可能
  - プリスケーラ出力クロックに対し、 $N + m/16$  ( $N = 2\sim 15$ ,  $m = 1\sim 15$ )分周が可能  
(UART モードのみ)
  - システムクロック(fsys)を使用可能(UART モードのみ)
- ・ バッファ
  - ダブルバッファ構成で使用可能
  - 送信バッファのクリアが可能
- ・ FIFO
  - 送受信合わせて 4 バイトの FIFO を使用可能
- ・ I/O インタフェースモード
  - 転送モード：半二重(受信/送信)、全二重
  - クロック：立ち上がり/立ち下がりエッジ選択
  - 連続転送時のインタバル時間設定が可能
  - 最終ビット出力後の SCxTXD 端子の状態を下記から選択可能  
"High"保持/"Low"保持/最終ビット保持
  - クロック入力モード時、アンダーランエラーが発生したときの SCxTXD 端子の状態を下記から選択可能  
"High"保持/"Low"保持
  - クロック入力モード時、SCxTXD 端子の最終ビットホールド時間を設定可能
- ・ UART モード
  - データ長：7, 8, 9 ビット
  - パリティ付加(9 ビット長では不可)
  - シリアルリンクでのウエイクアップ機能
  - $\overline{SCxCTS}$  端子を用いたハンドシェイク機能
  - SCxRXD 端子へのノイズキャンセラ付加

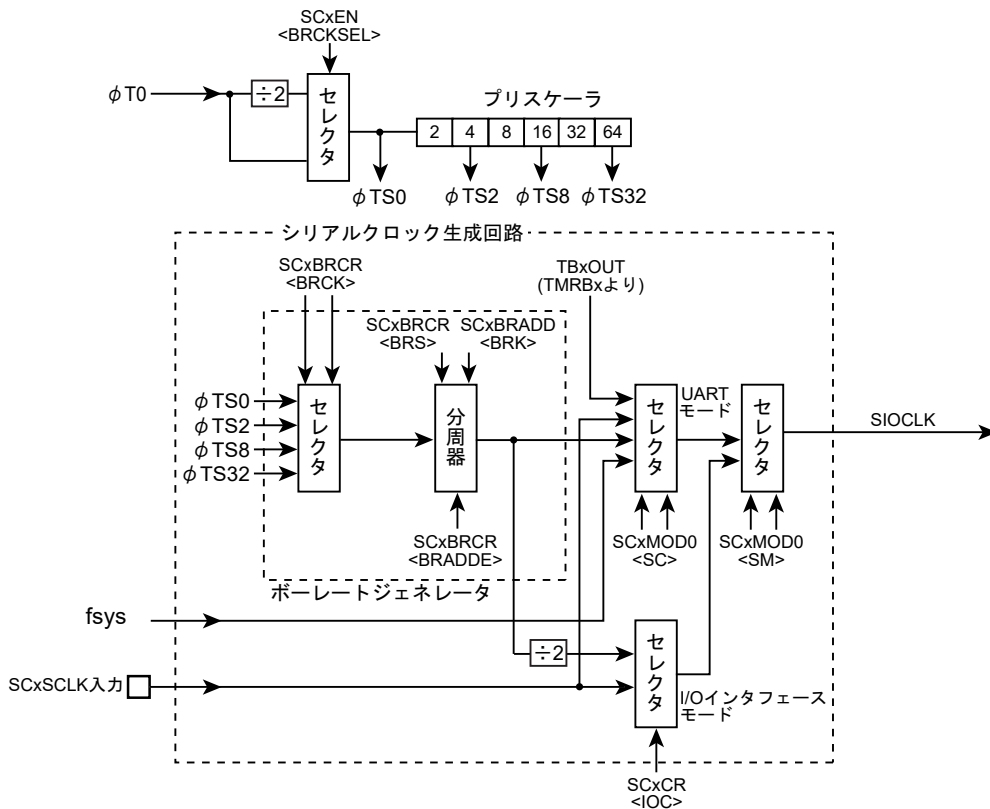
以下の説明中、"x"はチャネル番号をあらわします。

### 13.2

下記にシリアルチャネルとシリアルクロック生成回路のブロック図を示します。



13-1



13-2

## 13.3

## 13.3.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address (Base+)
	SCxEN	0x0000
	SCxBUF	0x0004
	SCxCR	0x0008
0	SCxMOD0	0x000C
	SCxBRCR	0x0010
2	SCxBRADD	0x0014
1	SCxMOD1	0x0018
2	SCxMOD2	0x001C
FIFO	SCxRFC	0x0020
FIFO	SCxTFC	0x0024
FIFO	SCxRST	0x0028
FIFO	SCxTST	0x002C
FIFO	SCxFCNF	0x0030

)



### 13.3.3 SCxBUF ( )

SCxBUF は、書き込み時は送信バッファまたは FIFO、読み出し時は受信バッファまたは FIFO として機能します。

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TB / RB							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	TB[7:0] / RB [7:0]	R/W	[ ] TB FIFO [ ] RB FIFO

13.3.4 SCxCR ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	EHOLD			-	TXDEMP	TIDLE	
	0	0	0	0	0	1	1	0
	7	6	5	4	3	2	1	0
bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-15		R	"0"
14-12	EHOLD[2:0]	R/W	<p>SCxTXD (I/O SCLK )</p> <p><math>\leq \text{SCLK} / 2</math></p> <p>000: 2/fsys                      100: 32/fsys                      001: 4/fsys                      101: 64/fsys                      010: 8/fsys                      110: 128/fsys                      011: 16/fsys                      111: Reserved</p>
11		R	"0"
10	TXDEMP	R/W	<p>SCxTXD (I/O )</p> <p>0: "Low"                      1: "High"</p>
9-8	TIDLE[1:0]	R/W	<p>SCxTXD (I/O )</p> <p>&lt;TIDLE[1:0]&gt;="10"                      &lt;EHOLD[2:0]&gt;="000"</p> <p>00: "Low"                      01: "High"                      10:                      11: Reserved</p>
7	RB8	R	<p>8 (UART )</p> <p>9                      9</p>
6	EVEN	R/W	<p>(UART )</p> <p>7                      8                      8                      8                      8</p> <p>0: Odd                      1: Even</p>
5	PE	R/W	<p>(UART )</p> <p>0:                      1:                      /                      7                      8                      8                      8</p>
4	OERR	R	<p>( )</p> <p>0:                      1:</p>
3	PERR	R	<p>/ ( )</p> <p>0:                      1:</p>

Bit	Bit Symbol	Type	
2	FERR	R	( ) 0: 1:
1	SCLKS	R/W	(I/O ) 0: SCxSCLK 1bit SCxTXD SCxSCLK SCxRXD 1bit SCxSCLK High ( ) 1: SCxSCLK 1bit SCxTXD SCxSCLK SCxRXD 1bit SCxSCLK Low ( )
0	IOC	R/W	(I/O ) 0: ( SCxSCLK ) 1: ( SCxSCLK )

) <OERR>, <PERR>, <FERR>

13.3.5 SCxMOD0 ( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TB8	CTSE	RXE	WU	SM		SC	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	TB8	R/W	8 (UART ) 9 UART 9
6	CTSE	R/W	(UART ) 0: CTS 1: CTS "1" <u>SCxCTS</u>
5	RXE	R/W	( 1)( 2) 0: 1:
4	WU	R/W	(UART ) 0: 1: 9 UART 9 "1"
3-2	SM[1:0]	R/W	00: I/O 01: 7 UART 10: 8 UART 11: 9 UART
1-0	SC[1:0]	R/W	(UART ) 00: TMRB 01: 10: (fsys) 11: (SCxSCLK ) (I/O SCxCR<IOC> )

- 1) <RXE>
- 2) (SCxMOD0<RXE> "0" )

13.3.6 SCxMOD1 ( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	I2SC	FDPX		TXE	SINT			-
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	I2SC	R/W	IDLE 0: 1: IDLE
6-5	FDPX[1:0]	R/W	00: 01: ( ) 10: ( ) 11: I/O FIFO FIFO UART FIFO
4	TXE	R/W	( 1)( 2) 0: 1:
3-1	SINT[2:0]	R/W	(I/O ) 000: 001: 1 x SCLK 010: 2 x SCLK 011: 4 x SCLK 100: 8 x SCLK 101: 16 x SCLK 110: 32 x SCLK 111: 64 x SCLK I/O I/O FIFO
0		R/W	"0"

1) <TXE>

2) (SCxMOD1<TXE> "0" )

13.3.7 SCxMOD2 ( 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TBEMP	RBFULL	TXRUN	SBLEN	DRCHG	WBUF	SWRST	
	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type												
31-8		R	"0"											
7	TBEMP	R	0: Full 1: Empty  empty empty "1" "0"											
6	RBFULL	R	full 0: Empty 1: Full  full "1" "0"											
5	TXRUN	R	0: 1:  <TXRUN> <TBEMP> <table border="1" style="margin-left: 20px;"> <tr> <td>&lt;TXRUN&gt;</td> <td>&lt;TBEMP&gt;</td> <td></td> </tr> <tr> <td>1</td> <td></td> <td></td> </tr> <tr> <td rowspan="2">0</td> <td>1</td> <td></td> </tr> <tr> <td>0</td> <td></td> </tr> </table>	<TXRUN>	<TBEMP>		1			0	1		0	
<TXRUN>	<TBEMP>													
1														
0	1													
	0													
4	SBLEN	R/W	STOP (UART ) 0: 1 1: 2 UART STOP STOP 1											
3	DRCHG	R/W	0: LSB first 1: MSB first  UART LSB first											
2	WBUF	R/W	0: 1: I/O ( / ), ( ), UART I/O ( / ), UART											

Bit	Bit Symbol	Type									
1-0	SWRST[1:0]	R/W	<p>"10"→"01"</p> <p>( 1)( 2)</p> <table border="1"> <tr> <td>SCxMOD0</td> <td>&lt;RXE&gt;</td> </tr> <tr> <td>SCxMOD1</td> <td>&lt;TXE&gt;</td> </tr> <tr> <td>SCxMOD2</td> <td>&lt;TBEMP&gt;, &lt;RBFLL&gt;, &lt;TXRUN&gt;</td> </tr> <tr> <td>SCxCR</td> <td>&lt;OERR&gt;, &lt;PERR&gt;, &lt;FERR&gt;</td> </tr> </table> <p>FIFO</p>	SCxMOD0	<RXE>	SCxMOD1	<TXE>	SCxMOD2	<TBEMP>, <RBFLL>, <TXRUN>	SCxCR	<OERR>, <PERR>, <FERR>
SCxMOD0	<RXE>										
SCxMOD1	<TXE>										
SCxMOD2	<TBEMP>, <RBFLL>, <TXRUN>										
SCxCR	<OERR>, <PERR>, <FERR>										

1)

2

2)

2

13.3.8 SCxBRCR ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	BRADDE	BRCK		BRS			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7		R/W	"0"
6	BRADDE	R/W	$N + (16 - K)/16$ (UART ) 0: 1:
5-4	BRCK[1:0]	R/W	00: TS0 01: TS2 10: TS8 11: TS32
3-0	BRS[3:0]	R/W	"N" 0000 : N = 16 0001 : N = 1 0010 : N = 2 : 1111 : N = 15

1) UART  $N + (16 - K)/16$  "N" 1 ("0001") 16 ("0000")

2) I/O "N" 1 ("0001")



13.3.10 SCxFCNF (FIFO )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	RFST	TFIE	RFIE	RXTXCNT	CNFG
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type							
31-8	-	R	"0"						
7-5	-	R/W	"000"						
4	RFST	R/W	FIFO 0: FIFO FILL 1: FIFO ( 1 ) "0" FIFO (<CNFG> ) "1" SCxRFC<RIL[1:0]> FILL						
3	TFIE	R/W	FIFO 0: FIFO / 1: FIFO /						
2	RFIE	R/W	FIFO 0: FIFO / 1: FIFO /						
1	RXTXCNT	R/W	RXE/TXE 0: / 1: / "1" <table border="1" style="margin-left: 20px;"> <tr> <td></td> <td>FIFO (SCxMOD0 &lt;RXE&gt; "0"</td> </tr> <tr> <td></td> <td>FIFO (&lt;TXE&gt; "0"</td> </tr> <tr> <td></td> <td>"0"</td> </tr> </table>		FIFO (SCxMOD0 <RXE> "0"		FIFO (<TXE> "0"		"0"
	FIFO (SCxMOD0 <RXE> "0"								
	FIFO (<TXE> "0"								
	"0"								
0	CNFG	R/W	FIFO 0: FIFO ( 2 ) 1: FIFO "1" FIFO FIFO <table border="1" style="margin-left: 20px;"> <tr> <td></td> <td>FIFO 4</td> </tr> <tr> <td></td> <td>FIFO 4</td> </tr> <tr> <td></td> <td>FIFO 2 + FIFO 2</td> </tr> </table>		FIFO 4		FIFO 4		FIFO 2 + FIFO 2
	FIFO 4								
	FIFO 4								
	FIFO 2 + FIFO 2								

1) FIFO FIFO (<CNFG> )  
 2) 9 UART FIFO

13.3.11 SCxRFC ( FIFO )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFCS	RFIS	-	-	-	-	RIL	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type																
31-8		R	"0"															
7	RFCS	W	FIFO ( ) 1: "1" FIFO SCxRST<RLVL>="000" "0"															
6	RFIS	R/W	0: FIFO fill (SCxRST<RLVL[2:0]> = fill (<RIL[1:0]>) 1: FIFO fill (SCxRST<RLVL[2:0]> ≥ fill (<RIL[1:0]>) 13.13.1.2															
5-2		R	"0"															
1-0	RIL[1:0]	R/W	FIFO fill <table border="1" style="margin-left: 20px;"> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td>00</td> <td>4</td> <td>2</td> </tr> <tr> <td>01</td> <td>1</td> <td>1</td> </tr> <tr> <td>10</td> <td>2</td> <td>2</td> </tr> <tr> <td>11</td> <td>3</td> <td>1</td> </tr> </table>				00	4	2	01	1	1	10	2	2	11	3	1
00	4	2																
01	1	1																
10	2	2																
11	3	1																

) / FIFO SIO ( / ) FIFO (SCxFCNF<CNFG>="1")  
/ FIFO

13.3.12 SCxTFC ( FIFO )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	TBCLR
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TFCS	TFIS	-	-	-	-	TIL	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type																
31-9		R	"0"															
8	TBCLR	W	0: Don't care 1: "1" "0"															
7	TFCS	W	FIFO ( 1) 0: Don't care 1: "1" FIFO SCxTST<TLVL>="000" "0"															
6	TFIS	R/W	0: FIFO fill (SCxTST<TLVL[2:0]> = fill (<TIL[1:0]>) 1: FIFO fill (SCxTST<TLVL[2:0]> ≤ fill (<TIL[1:0]>) 13.13.2.2															
5-2		R	"0"															
1-0	TIL[1:0]	R/W	FIFO fill <table border="1" style="margin-left: 20px;"> <tr> <td></td> <td></td> <td></td> </tr> <tr> <td>00</td> <td>Empty</td> <td>Empty</td> </tr> <tr> <td>01</td> <td>1</td> <td>1</td> </tr> <tr> <td>10</td> <td>2</td> <td>Empty</td> </tr> <tr> <td>11</td> <td>3</td> <td>1</td> </tr> </table>				00	Empty	Empty	01	1	1	10	2	Empty	11	3	1
00	Empty	Empty																
01	1	1																
10	2	Empty																
11	3	1																

- 1) / FIFO SIO ( / ) FIFO (SCxFCNF<CNFG>="1")  
/ FIFO
- 2) SCxEN<SIOE>=0(SIO/UART ) SCxMOD1<I2SC>=0 IDLE (IDLE )  
SCxTFC

13.3.13 SCxRST ( FIFO )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ROR	-	-	-	-	RLVL		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	ROR	R	FIFO ( ) 0: 1:
6-3		R	"0"
2-0	RLVL[2:0]	R	FIFO fill 000: Empty 001: 1 010: 2 011: 3 100: 4

) <ROR> (SCxBUF) "0"

13.3.14 SCxTST ( FIFO )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TUR	-	-	-	-	TLVL		
	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	TUR	R	FIFO ( ) 0: 1:
6-3		R	"0"
2-0	TLVL[2:0]	R	FIFO fill 000: Empty 001: 1 010: 2 011: 3 100: 4

) <TUR> (SCxBUF) "0"

## 13.4

表 13-2 にモードをまとめます。

13-2

					STOP ( )
0	(I/O )	8	LSB first/MSB first	-	-
1	(UART )	7	LSB first		1 2
2		8			
3		9		3	

モード 0 は同期通信モードで、I/O を拡張するために使用できます。SCLK クロックに同期してデータの送受信を行います。SCLK クロックはクロック入力/出力モードのいずれでも使用できます。転送方向は、LSB first と MSB first から選択可能です。パリティ付加機能はなく、STOP ビットも使用しません。

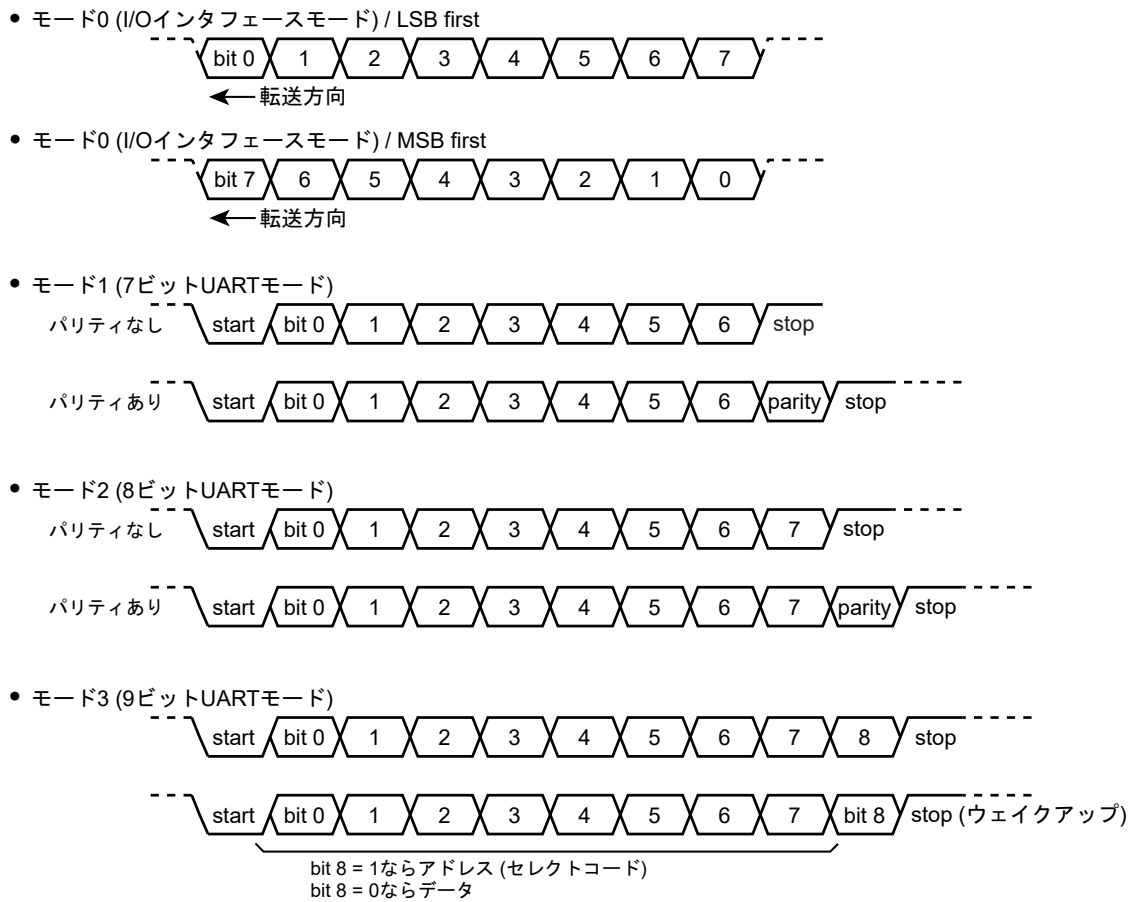
モード 1 からモード 3 は非同期通信モードです。転送方向は LSB first のみ選択できます。

モード 1 とモード 2 はパリティビットの付加が可能です。モード 3 は、マスタコントローラが、シリアルリンク(マルチコントローラシステム)でスレーブコントローラを起動させるためのウエイクアップ機能を有しています。送信時の STOP ビットを 1 ビットまたは 2 ビットから選択できます。受信時の STOP ビット長は 1 ビット固定です。

# 13.5

## 13.5.1

図 13-3 にデータフォーマットを示します。



13-3

## 13.5.2

7ビット UART モードまたは 8 ビット UART モードでは送信データにパリティビットを付加することができます。また、受信データに付加されているパリティビットを生成されたパリティビットと比較することができます。

SCxCR<PE>に"1"を設定するとパリティが有効になります。SCxCR<EVEN>で偶数/奇数パリティを選択することができます。

### 13.5.2.1

送信時、パリティ制御回路は送信バッファのデータに対して自動的にパリティを発生し、7ビット UART モードのときは SCxBUF<TB7>に、8 ビット UART モードのときは SCxMOD0<TB8>にパリティが格納されます。

なお<PE>と<EVEN>の設定は、送信データをバッファレジスタに書き込む前に行ってください。

### 13.5.2.2

受信データが受信シフトレジスタから受信バッファに移されるとパリティを自動発生し、7ビット UART モードのときは SCxBUF<RB7>と、8 ビット UART モードのときは SCxCR<RB8>と生成したパリティを比較します。異なっているとパリティエラーが発生し、SCxCR<PERR>がセットされます。

FIFO を使用する場合、<PERR>は受信したいずれかのデータでパリティエラーが発生したことを示します。

## 13.5.3 STOP

SCxMOD2<SBLLEN>で、UART 送信モードの STOP ビット長を 1 ビットまたは 2 ビットに設定できます。受信の場合にはこのビットの設定にかかわらず 1 ビットの STOP ビット長として認識します。

## 13.6

## 13.6.1

7ビットのプリスケアラを実装しており、 $\phi T0$ の1/2/4/8/16/32/64/128分周のクロックを生成します。

プリスケアラの入力クロックは、クロック/モード制御部のCGSYSCRとSCxEN<BRCKSEL>で選択します。

プリスケアラは、SCxMOD0<SC[1:0]>="01"でポーレートジェネレータを転送クロックとして選択した場合に動作します。

## 13.6.2

送受信クロック(SIOCLK)を生成するブロックで、ポーレートジェネレータとモードによりクロックを選択する回路で構成されています。

## 13.6.2.1

ポーレートジェネレータは、シリアルチャネルの転送速度を定める送受信クロックを生成するための回路です。

## (1)

ポーレートジェネレータの入力クロックは、プリスケアラ出力の1/4/16/64分周から選択します。入力クロックの選択はSCxEN<BRCKSEL>とSCxBRCR<BRCK>で行います。

SCxEN<BRCKSEL>	SCxBRCR<BRCK>	Tx
0	00	T0/2
0	01	T0/8
0	10	T0/32
0	11	T0/128
1	00	T0
1	01	T0/4
1	10	T0/16
1	11	T0/64

## (2)

ポーレートジェネレータの出力クロック分周値は、SCxBRCR、SCxBRADDで設定します。

I/OインタフェースモードではN分周、UARTモードではN分周またはN+(16-K)/16分周が使用できます。

以下に設定可能な分周値をまとめます。

	SCxBRCR<BRADDE>	N SCxBRCR<BRS[3:0]>	K SCxBRADD<BRK[3:0]>
I/O	N	1 ~ 16 ( )	-
UART	N	1 ~ 16	-
	$N + (16-K)/16$	2 ~ 15	1 ~ 15

) 1

ボーレートジェネレータ分周機へのクロック入力を  $\phi Tx$  とした時、N 分周の場合と  $N + (16-K)/16$  分周の場合のボーレートジェネレータ出力クロックは以下の計算式であらわされます。

- ・ N 分周

$$\text{ボーレートジェネレータ出力クロック} = \frac{\phi Tx}{N}$$

- ・  $N + (16-K)/16$  分周

$$\text{ボーレートジェネレータ出力クロック} = \frac{\phi Tx}{N + \frac{(16-K)}{16}}$$

13.6.2.2

モードとレジスタ設定により、クロックが選択されます。

モードは SCxMOD0<SM[1:0]>で指定します。

I/O インタフェースモード時のクロックは、SCxCR<IOC><SCLKS>で設定します。

UART モード時のクロックは、SCxMOD0<SC[1:0]>で設定します。

(1) I/O

表 13-3 に I/O インタフェースモードで可能なクロックを示します。

13-3 I/O

SCxMOD0<SM[1:0]>	SCxCR<IOC>	SCxCR<SCLKS>	
"00" (I/O )	"0" ( )	"0" ( : ) :	2
		"1" ( : ) :	2
	"1" ( )	"0" ( : ) :	SCxSCLK
		"1" ( : ) :	SCxSCLK

SCxSCLK 端子入力を使用する場合、以下の条件を満足する必要があります。

- ・ ダブルバッファ使用の場合  
SCLK 周期 > 6/fsys
- ・ ダブルバッファ未使用の場合  
SCLK 周期 > 8/fsys

(2) UART

表 13-4 に UART モードの場合のクロック選択を示します。UART モードでは、選択されたクロックを受信/送信カウンタでさらに 16 分周して使用します。

13-4 UART

SCxMOD0<SM[1:0]>	SCxMOD0<SC[1:0]>
UART ("01", "10", "11")	"00" :
	"01" :
	"10" : fsys
	"11" : SCxSCLK

SCxSCLK 端子入力を使用する場合、以下の条件を満足する必要があります。

- SCLK 周期 > 2/fsys

タイマの出力を使用する場合、カウンタと TBxRG1 の一致でタイマフリップフロップ出力を反転させる設定とします。SIOCLK クロック周期は「TBxRG1 設定値 × 2」となります。

ボーレートは以下の計算式で求められます。

ボーレートの算出方法

$$\text{転送レート} = \frac{\text{CGSYSCR<PRCK[1:0]>で選択されたクロック周波数}}{(\text{TBxRG1} \times 2) \times 2 \times 16}$$

↑ (タイマフリップフロップ反転2回で1クロック周期となる)

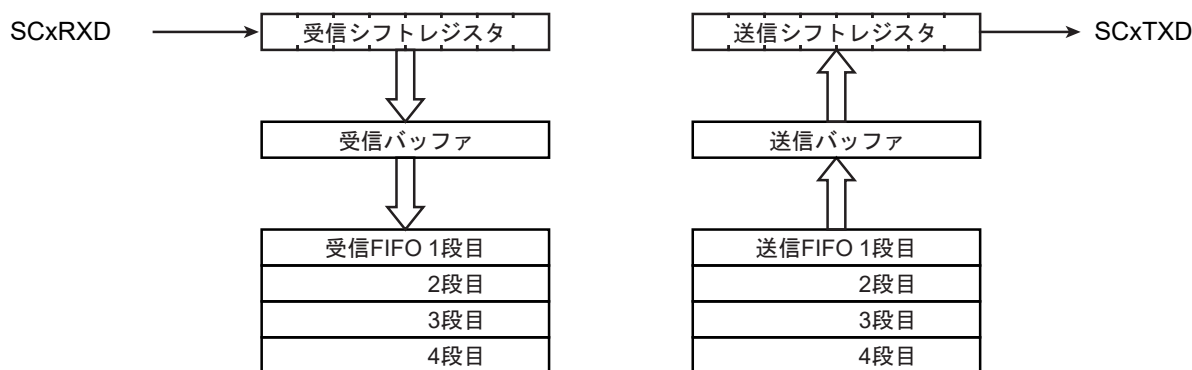
↑ (タイマプリスケーラクロックφT1(2分周)を選択した場合)

13.7 / FIFO

13.7.1

送信/受信バッファと FIFO の構成を図 13-4 に示します。

バッファと FIFO を使用するには設定が必要です。また、モードによっては構成が決まっている場合があります。



13-4 FIFO

13.7.2 /

送信および受信バッファはダブルバッファ構造となっています。バッファ構成の設定は、SCxMOD2<WBUF>で行います。

受信の場合、I/O インタフェースモードでクロック入力モードの場合と UART モードでは、<WBUF>の設定によらずダブルバッファ構成になります。

その他のモードでは<WBUF>の設定に従います。

表 13-5 にモードとバッファ構成の関係をまとめます。

13-5

		SCxMOD2<WBUF>	
		"0"	"1"
UART			
I/O	( )		
I/O	)		

### 13.7.3

送信バッファにデータが残っている状態で送信を中断した場合、送信バッファに新たにデータを書き込む前に送信バッファの初期化を行う必要があります。

送信バッファの初期化は、送信動作が停止している状態で行ってください。SCxMOD2<TXRUN>を読み出すことで、送信動作が停止しているかを判断できます。送信動作が停止していることが確認できたら、SCxTFC<TBCLR>に"1"を設定し、送信バッファの初期化を行います。

送信 FIFO が有効な場合、送信 FIFO の状態で動作が変わります。送信 FIFO にデータがある場合、送信 FIFO から送信バッファにデータが転送されます。送信 FIFO が空なら、SCxMOD2<TBEMP>が"1"にセットされます。

) I/O

### 13.7.4 FIFO

ダブルバッファに加えて、4byte の FIFO を使用することができます。

FIFO を有効にするには SCxMOD2<WBUF>を"1"としてダブルバッファをイネーブルにし、SCxFCNF<CNFG>に"1"をセットします。FIFO バッファの構成は SCxMOD1<FDPX>で設定します。

) / FIFO SIO ( / ) FIFO (SCxFCNF<CNFG>="1")  
 / FIFO

表 13-6 にモードと FIFO 構成の関係をまとめます。

13-6 FIFO

	SCxMOD1<FDPX[1:0]>	FIFO	FIFO
	"01"	4byte	-
	"10"	-	4byte
	"11"	2byte	2byte

### 13.8

SCxMOD2 に 2 種類のステータスフラグが準備されています。これらのフラグはダブルバッファ許可の時のみ意味を持ちます。

<RBFL>は、受信バッファ full を示すフラグです。1 データの受信が終了し、データが受信シフトレジスタから受信バッファに移されると"1"にセットされます。受信バッファを読み出すと"0"にクリアされず。

<TBEMP>は、送信バッファ empty を示すフラグです。送信バッファから送信シフトレジスタへデータが移されると、"1"がセットされます。送信バッファにデータをセットすると"0"にクリアされます。

### 13.9

SCxCR に 3 種類のエラーフラグが準備されています。フラグによってはモードにより意味が変わります。以下にモードごとのフラグの意味をまとめます。

これらのフラグは、SCxCR を読み出すと"0"にクリアされます。

	<OERR>	<PERR>	<FERR>
UART			
I/O ( )		( ) FIFO "0" ( ) FIFO	"0"
I/O ( )			"0"

#### 13.9.1 OERR

UART モード、I/O インタフェースモード共に、受信バッファのデータを読み出す前に次のフレームの受信が終了すると"1"にセットされます。

受信 FIFO を有効にしている場合は、受信 FIFO ヘデータが自動的に移されるので、受信 FIFO が Full(使用バイト数)になるまではフラグはセットされません。

I/O インタフェースモードのクロック出力モードの設定では、フラグのセットとともにクロック出力が停止します。

) I/O

SCxCR

### 13.9.2 PERR

UART モードではパリティエラーを、I/O インタフェースモードではアンダーランエラーまたは送信終了を示します。

パリティエラーは UART モードで受信したデータから生成されたパリティと受信したパリティビットが異なる場合に"1"にセットされます。

アンダーランエラーは、I/O インタフェースモードでダブルバッファが有効な場合に以下の条件で"1"にセットされます。

クロック入力モードの場合、送信シフトレジスタのデータを送信終了後、送信バッファにデータがない状態で次の転送クロックが入力されるとセットされます。

クロック出力モードの場合、すべてのデータ送信が終了するとセットされクロック出力を停止します。

) I/O

SCxCR

### 13.9.3 FERR

フレーミングエラーは UART モードの受信データのストップビットを中央付近でサンプリングし、結果が"0"の場合に発生します。SCxMOD2<SBLEN>でのストップビット長設定に関わらず、判定は第1ストップビットで行われます。

I/O インタフェースモードではこのビットは"0"固定です。

## 13.10

## 13.10.1

受信カウンタは4ビットのバイナリカウンタで、SIOCLKでカウントアップされます。

UARTモードでは、データ1ビットの受信にSIOCLKが16クロック用いられ、8クロック目でデータをサンプリングします。

## 13.10.2

## 13.10.2.1 I/O

SCxCR<IOC>="0"のクロック出力モードのときは、SCxCR<SCLKS>の設定に従って、SCxSCLK端子より出力されるクロックの立ち上がり/立ち下がりによってSCxRXD端子をサンプリングします。

SCxCR<IOC>="1"のクロック入力モードのときは、SCxCR<SCLKS>の設定に従って、SCxSCLK端子の立ち上がり/立ち下がりエッジでシリアル受信データSCxRXD端子をサンプリングします。

## 13.10.2.2 UART

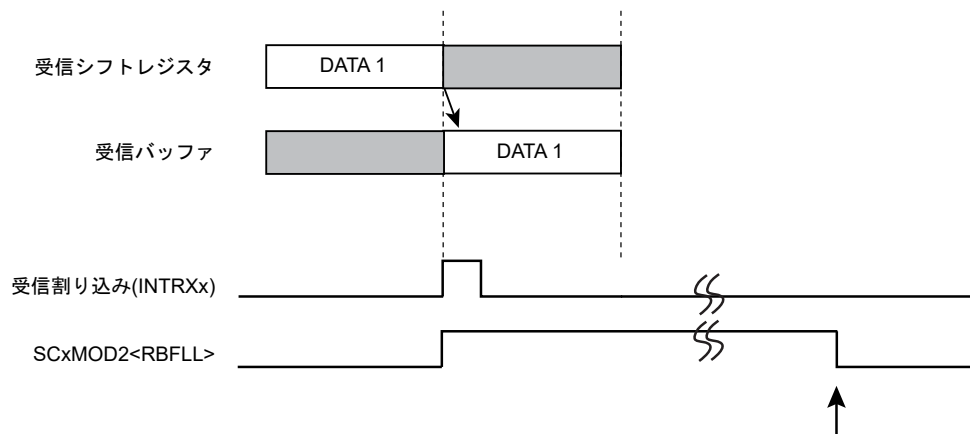
受信制御部はスタートビット検出回路を持ち、正常なスタートビットを判断して受信動作を開始します。

## 13.10.3

## 13.10.3.1

受信シフトレジスタに受信データが1ビットずつ格納され、データがそろそろと割り込みINTRXxが発生します

ダブルバッファ設定の場合は、データは受信バッファ(SCxBUF)へ移され受信バッファのfullフラグ(SCxMOD2<RBFL>)が"1"にセットされます。受信バッファfullフラグは、受信バッファを読み出すと"0"にクリアされます。シングルバッファの場合、受信バッファfullフラグは意味を持ちません。



13.10.3.2 FIFO

FIFO が許可されている場合、受信データは受信バッファから受信 FIFO に移され、受信バッファ full フラグはただちにクリアされます。割り込みは SCxRFC<RIL[1:0]>の設定に従って発生します。

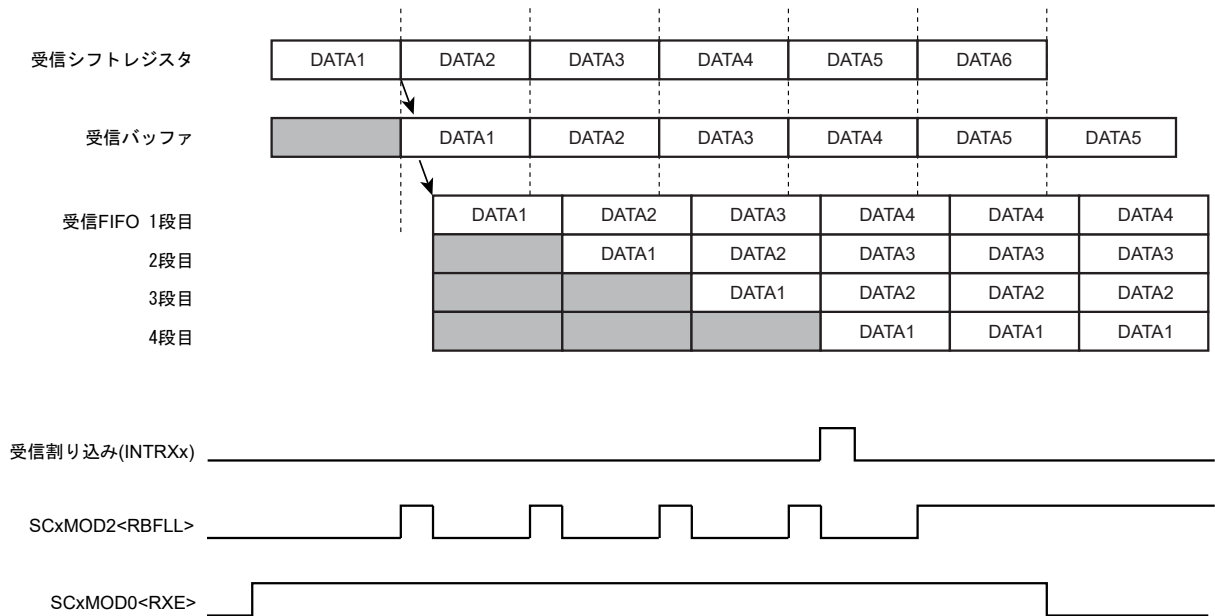
) UART FIFO

以下に、半二重受信の設定と動作を示します。

```
SCxMOD1<FDPX[1:0]> = "01"
SCxFCNF<RFST><TFIE><RFIE> fill
<RXCNT><CNFG> = "10111" FIFO fill
SCxRFC<RIL[1:0]> = "00" FIFO fill 4
SCxRFC<RFCS><RFIS> = "11" FIFO
```

上記の FIFO 構成の設定後、SCxMOD0<RXE> に 1 を書き込むとデータ受信を開始します。受信シフトレジスタ、受信バッファ、受信 FIFO すべてにデータが格納されると<RXE>を自動クリアして受信を終了します。

上記の設定で、fill レベル到達後の継続受信を許可にしておくと、FIFO のデータを読み出すことにより継続して受信動作を行うことができます。



13-6 FIFO

### 13.10.3.3 I/O

I/O インタフェースモードでクロック出力モード設定の場合、使用可能な受信バッファ/FIFO にすべてデータが格納されるとクロック出力が停止されます。このため、このモードではオーバーランエラーフラグは意味を持ちません。

クロック出力の停止/再開のタイミングはバッファ/FIFO の使用状況によって変わります。

#### (1)

1 データ受信後にクロック出力を停止します。このため、通信相手と 1 データごとのハンドシェイクが可能です。バッファからデータが読み出されるとクロック出力を再開します。

#### (2)

受信シフトレジスタ、受信バッファともにデータが格納されると、クロック出力を停止します。1 データが読み出されるとクロック出力を再開します。

#### (3) FIFO

受信シフトレジスタ、受信バッファ、FIFO すべてにデータが格納されるとクロック出力を停止します。

1 データが読み出されると受信バッファから FIFO へ、受信シフトレジスタから受信バッファへデータが転送され、クロック出力を再開します。

また、SCxFNCF<RXTXCNT>がセットされているとクロック出力停止とともに SCxMOD0 <RXE>がクリアされ受信動作を停止します。

### 13.10.3.4

FIFO の有効/無効にかかわらず、受信バッファ (SCxBUF) からデータを読み出します。

受信 FIFO が有効にされていない場合は、この読み出しにより受信バッファの full フラグ SCxMOD2<RBFL>は"0"にクリアされます。受信バッファを読み出す前でも、次の受信データは受信シフトレジスタに格納することができます。8 ビット UART モードでパリティ付加の場合と 9 ビット UART モードの場合、最上位ビットは SCxCR <RB8> に格納されます。

受信 FIFO が有効な場合、FIFO に格納できるデータは最大 8 ビットですので、9 ビット UART モードは使用できません。8 ビット UART モードでパリティ付加の場合、パリティビットは失われますがエラー判定は行われ、結果が SCxCR<PERR>に格納されます。

### 13.10.3.5

9 ビット UART モードの場合、ウェイクアップ機能 SCxMOD0<WU> を"1"にすることによって、スレーブコントローラのウェイクアップ動作が可能で、SCxCR<RB8> = "1"のときのみ、割り込み INTRXx を発生させることができます。

### 13.10.3.6

受信 FIFO が有効にされていない場合は、受信シフトレジスタに次のデータが全ビット受信される前に受信バッファ (SCxBUF) を読み出さなければオーバーランエラーとなります。オーバーランエラーが発生した場合、受信バッファおよび SCxCR <RB8> の内容は保存されていますが、受信シフトレジスタの内容は失われます。

受信 FIFO が有効にされている場合は、受信 FIFO が full になり、受信バッファに次のデータが移される前に受信 FIFO を読み出さないと、受信 FIFO のオーバーランが発生してオーバーランフラグがセットされます。この場合でも、受信 FIFO のデータは保存されます。

I/O インタフェースモードのクロック出力モードの設定では、クロック出力が自動的に停止するためこのフラグは意味を持ちません。

) I/O

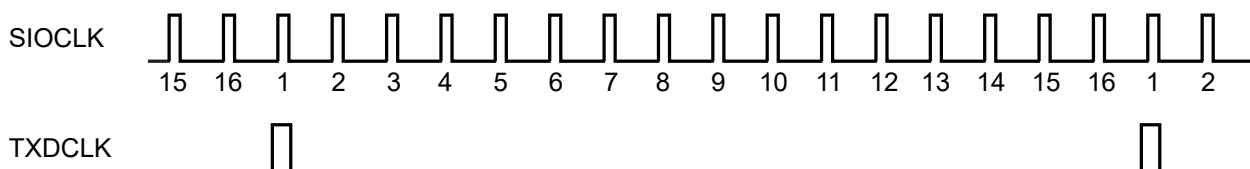
SCxCR

## 13.11

### 13.11.1

送信カウンタは4ビットのバイナリカウンタで、受信カウンタ同様 SIOCLK でカウントされます。

UART モードでは、16 クロックごとに送信クロック (TXDCLK) を生成します。



13-7 UART

### 13.11.2

#### 13.11.2.1 I/O

SCxCR<IOC>="0"のクロック出力モードのときは、SCxCR<SCLKS>の設定に従って、SCxSCLK 端子より出力されるクロックの立ち上がり/立ち下がりで送信バッファのデータを1ビットずつ SCxTXD 端子へ出力します。

SCxCR<IOC>="1"のクロック入力モードのときは、SCxCR<SCLKS>の設定に従って、SCxSCLK 端子の立ち上がり/立ち下がりエッジで送信バッファのデータを1ビットずつシリアル送信データ SCxTXD 端子へ出力します。

#### 13.11.2.2 UART

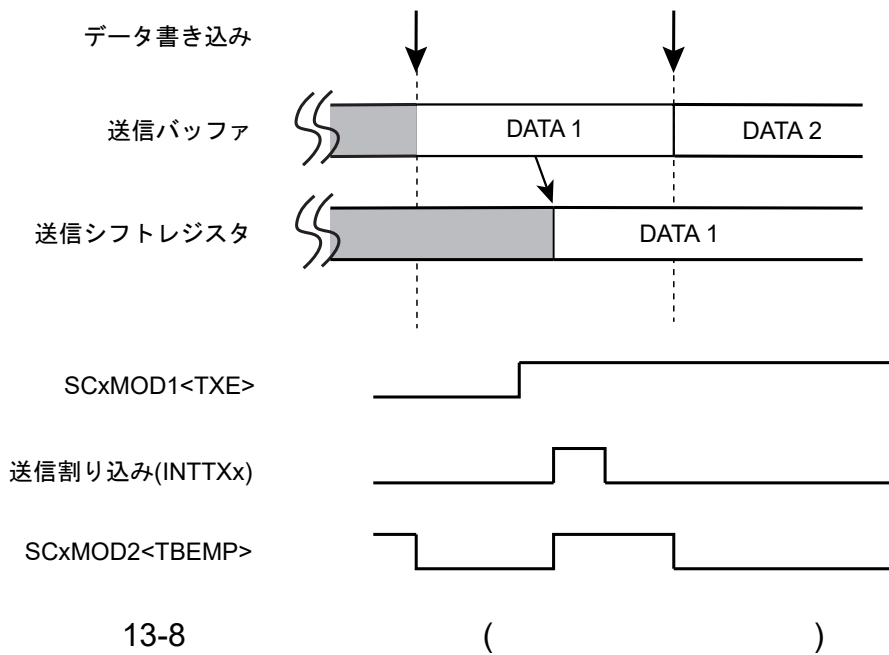
送信バッファに送信データが書き込まれると、次の TXDCLK の立ち上がりエッジから送信を開始し、送信シフトクロック を生成します。

### 13.11.3

#### 13.11.3.1

ダブルバッファ無効の場合、送信データの書き込みは送信シフトレジスタに対して行われ、送信が終了すると送信割り込み INTTXx が発生します。

ダブルバッファ有効の場合(送信 FIFO が有効な場合も含む)、送信バッファへ書き込まれたデータは SC x MOD1<TXE>に"1"がセットされると送信シフトレジスタに転送されます。同時に送信割り込み INTTXx が発生し、送信バッファエンプティフラグ(SCxMOD2<TBEMP>)がセットされ、次のデータが書き込み可能であることを示します。次のデータを送信バッファに書き込むと<TBEMP>は"0"にクリアされます。



13.11.3.2 FIFO

FIFO が許可されている場合、送信バッファと FIFO で最大 5 バイトのデータを格納することができます。送信を許可すると、送信バッファから送信シフトレジスタにデータが転送されて送信を開始するとともに、FIFO にデータが存在する場合はただちに送信バッファへ移され、<TBEMP> フラグは"0"にクリアされます。

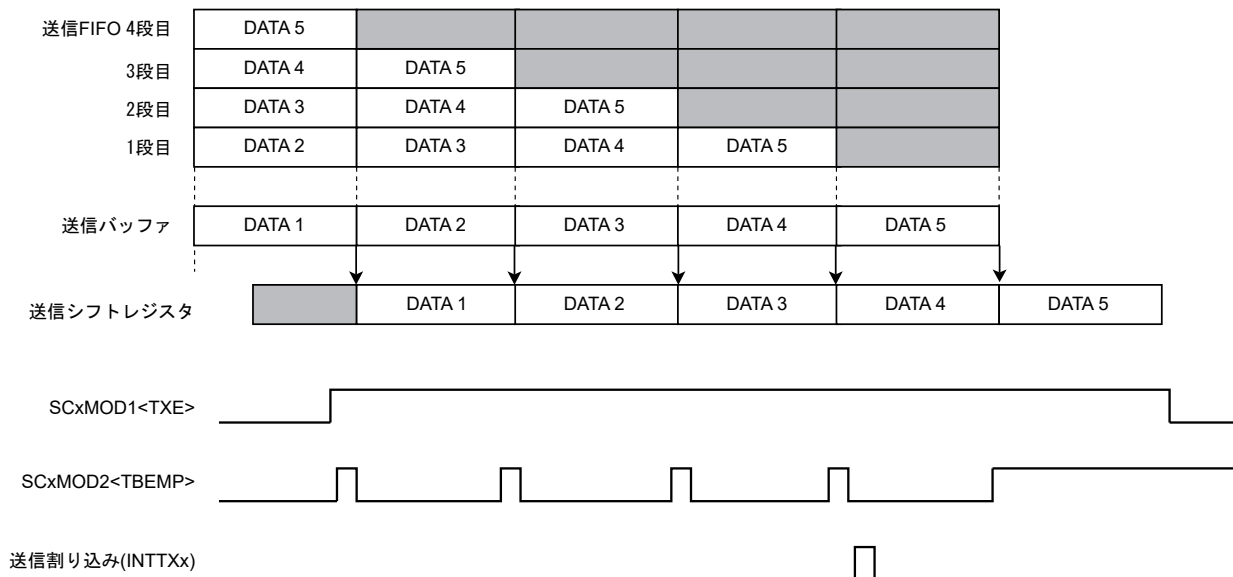
) FIFO SIO ( / ) FIFO (SCxFCNF  
<CNFG>="1") FIFO

以下に、5 バイトのデータを半二重送信する場合の設定と動作を示します。

```
SCxMOD1<FDPX[1:0]> = "10"
SCxFCNF<RFST><TFIE><RFIE> FIFO
<RXTXCNT><CNFG> = "11011" FIFO fill
SCxTFC<TIL[1:0]> = "00" fill 0
SCxTFC<TFCS><TFIS> = "11" FIFO
SCxFCNF<CNFG> = "1" FIFO
```

上記の設定後、送信バッファ/FIFO に 5 バイト分の送信データを書き込み、SCxMOD1<TXE> ビットを 1 に設定することによりデータ送信を開始します。最後の送信データが送信バッファに移されると送信 FIFO 割り込みが発生して最後のデータの送信が終了すると送信を終了します。

上記の設定で、送信の自動禁止としなければ送信データを書き込むことにより継続して送信を行うことができます。



### 13.11.3.3 I/O

I/O インタフェースモードでクロック出力モードの場合、設定されたデータがすべて送信されるとクロック出力は自動的に停止します。このため、このモードではアンダーランエラーは発生しません。

バッファ/FIFO 使用状況によってクロック出力の停止/再開のタイミングが変わります。

(1)

1 データ送信後にクロック出力を停止します。このため、通信相手と 1 データごとのハンドシェイクが可能です。バッファに次のデータが書き込まれるとクロック出力を再開します。

(2)

送信シフトレジスタと送信バッファのデータがすべて送信されるとクロック出力を停止します。バッファに次のデータが書き込まれるとクロック出力を再開します。

(3) FIFO

送信シフトレジスタ、送信バッファ、FIFO すべてのデータ送信が終了するとクロック出力を停止します。次のデータが書き込まれるとクロック出力を再開します。

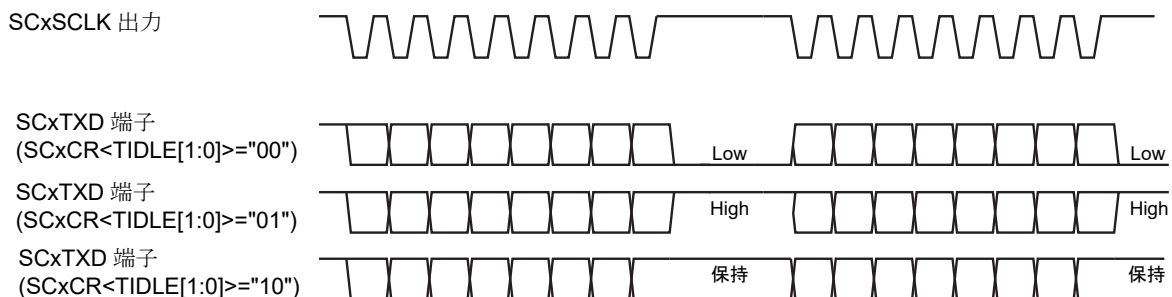
また、SCxFCNF<RXTXCNT>がセットされていると、クロック出力停止とともに SCxMOD0<TXE>がクリアされ、送信動作を停止します。

### 13.11.3.4 I/O

### SCxTXD

最終ビットを出力し、データのホールド時間が終了した後の SCxTXD 端子の状態を、SCxCR<TIDLE>で指定することができます。

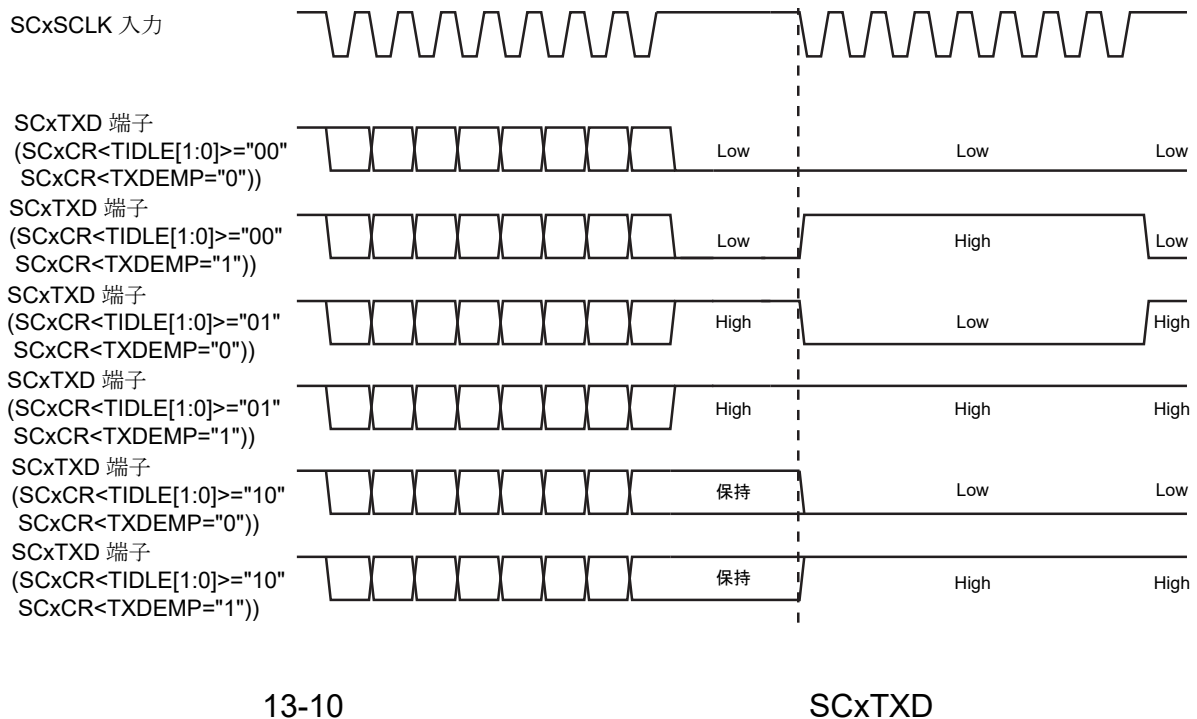
SCxCR<TIDLE>="00"の場合、SCxTXD 端子は"Low"出力を保持します。SCxCR<TIDLE>="01"の場合、SCxTXD 端子は"High"出力を保持します。SCxCR<TIDLE>="10"の場合、SCxTXD 端子は最終ビットを保持します。



13.11.3.5

I/O インタフェースモードのクロック入力モード時、送信 FIFO が空で、送信シフトレジスタのデータの送信が終了し、次の転送クロックが入力される前に送信バッファヘータがセットされないときはアンダーランエラーになり、SCxCR<PERR>に"1"がセットされます。

アンダーランエラーが発生したときの SCxTXD 端子の状態を SCxCR<TXDEMP>で指定することができます。SCxCR<TXDEMP>が"0"の場合、SCxTXD 端子はデータ出力期間の間、"Low"出力を保持します。SCxCR<TXDEMP>が"1"の場合、SCxTXD 端子はデータ出力期間の間、"High"出力を保持します。



I/O インタフェースモードのクロック出力モードの設定では、クロック出力が自動的に停止するためこのフラグは意味を持ちません。

) I/O

SCxCR

13.11.3.6 I/O

I/O インタフェースモードでクロック入力モードの場合、SCxCR<EHOLD[2:0]>で最終ビットのホールド時間を設定できます。

最終ビットのホールド時間 ≤ SCLK 周期/2 を満たすように、最終ビットのホールド時間、SCLK 周期を設定してください。

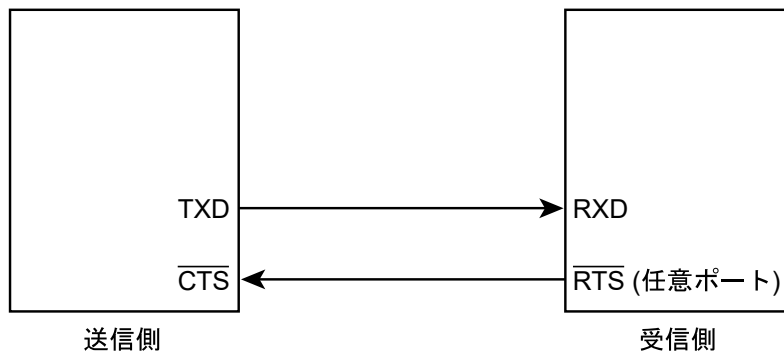
### 13.12

ハンドシェイク機能は  $\overline{\text{SCxCTS}}$  (Clear to send) 端子を用いて 1 データ単位での送信を行う機能で、この機能を使うことでオーバーランエラーの発生を防ぐことができます。ハンドシェイク機能は  $\text{SCxMOD0} <\text{CTSE}>$  によってイネーブル/ディセーブルを設定できます。

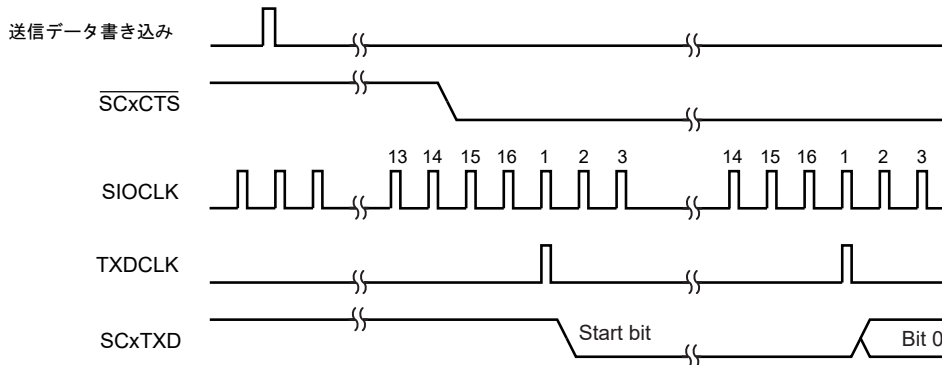
$\overline{\text{SCxCTS}}$  端子が "High" レベルになると、現在送信中のデータを送信完了後、 $\overline{\text{SCxCTS}}$  端子が "Low" レベルに戻るまで送信を停止します。ただし、 $\text{INTTIXx}$  割り込みは通常のタイミングで発生しますので、次の送信データを送信バッファに書き込み、送信待機状態にすることができます。

- 1)  $\overline{\text{SCxCTS}}$
- 2)  $\overline{\text{SCxCTS}}$  TXDCLK

なお、 $\overline{\text{RTS}}$  端子はありませんが、任意のポートの 1 ビットを  $\overline{\text{RTS}}$  機能に割り当て、受信終了時に (受信割り込みルーチン内で) このポートを "High" レベルにし、送信側に送信の一時停止を要求することで容易にハンドシェイク機能を構築できます。



13-11

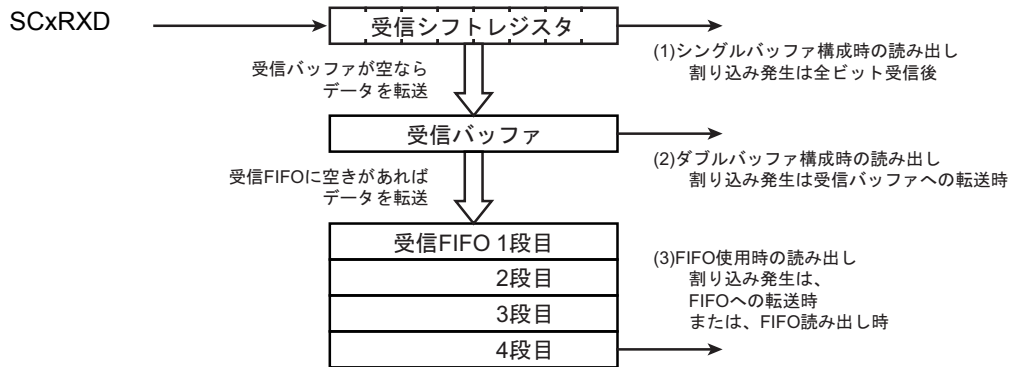


13-12  $\overline{\text{SCxCTS}}$

13.13 /

13.13.1

受信動作のデータの流れと読み出しの経路を図 13-13 に示します。



13-13 /FIFO

13.13.1.1 /

受信割り込みは、転送モードとバッファ構成により以下のタイミングで発生します。

13-7 /

	UART	I/O
		SCxSCLK / ( / SCxCR<SCLKS> )
		SCxSCLK ( / SCxCR<SCLKS> )

)

13.13.1.2 FIFO

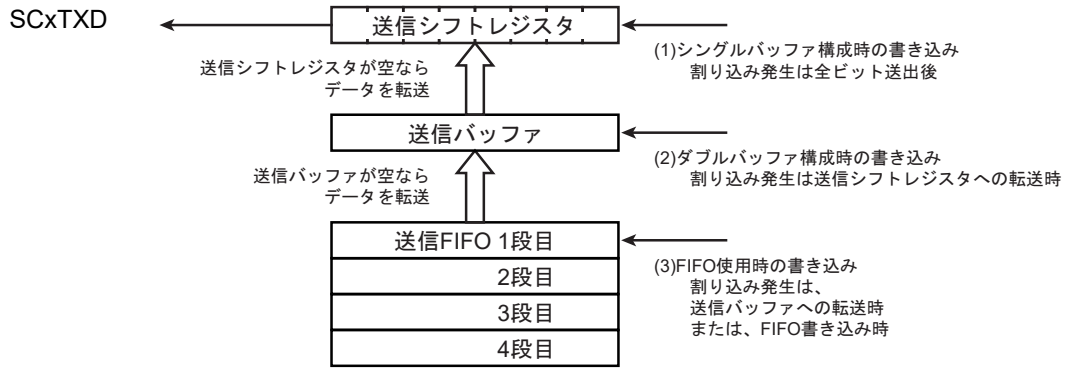
FIFO 使用の場合の受信割り込みは、表 13-8 の割り込み発生タイミングに記載の動作が発生したときに、SCxRFC<RFIS>の設定で決まる条件を満たしていると発生します。

13-8 FIFO

SCxRFC<RFIS>		
"0"	FIFO fill = fill	(SCxRST<RLVL[2:0]>) (<RIL[1:0]>) FIFO
"1"	FIFO fill ≥ fill	(SCxRST<RLVL[2:0]>) (<RIL[1:0]>) FIFO FIFO

13.13.2

送信動作のデータの流れと書き込みの経路を図 13-14 に示します。



13-14 /FIFO

13.13.2.1

送信割り込みは、転送モードとバッファ構成により以下のタイミングで発生します。

13-9

	UART	I/O
		SCxSCLK / ( / SCxCR<SCLKS> )
	SCxMOD1<TXE> = "1"	

13.13.2.2 FIFO

FIFO 使用の場合の送信割り込みは、表 13-10 の割り込み発生タイミングに記載の動作が発生したときに、SCxTFC<TFIS>の設定で決まる条件を満たした場合に発生します。

13-10 FIFO

SCxTFC<TFIS>		
"0"	FIFO fill (SCxTST<TLVL[2:0]>) = fill (<TIL[1:0]>)	FIFO
"1"	FIFO fill (SCxTST<TLVL[2:0]>) ≤ fill (<TIL[1:0]>)	FIFO FIFO

13.13.3

13.13.3.1 UART

	9	7 8 7 + 8 +

13.13.3.2 I/O

	SCxSCLK / ( / SCxCR<SCLKS> )
	SCxSCLK / ( / SCxCR<SCLKS> )

)

## 13.14 DMA

UART/SIO 割り込み(INTRX<sub>x</sub>,INTTX<sub>x</sub>)発生のタイミングでDMAC に対して DMA 要求を発行します。

本製品で DMA 要求に使用できるチャンネルについては、「製品情報」の章を参照してください。

- 1) SIO/UART / DMA SCxMOD<SWRST>  
DMAC /
- 2) DMA FIFO

## 13.15

SCxMOD2<SWRST[1:0]>に"10" → "01"の順でライトすることによりソフトウェアリセットが発生します。

これにより、SCxMOD0<RXE>、SCxMOD1<TXE>、SCxMOD2<TBEMP><RBFL><TXRUN>、SCxCR<OERR> <PERR> <FERR>が初期化され、送受信回路は初期状態になります。

その他の状態は保持されます。

## 13.16

### 13.16.1 0 (I/O )

SCxMOD0 <SM[1: 0]>を"00"にセットすると I/O インタフェースモードになります。

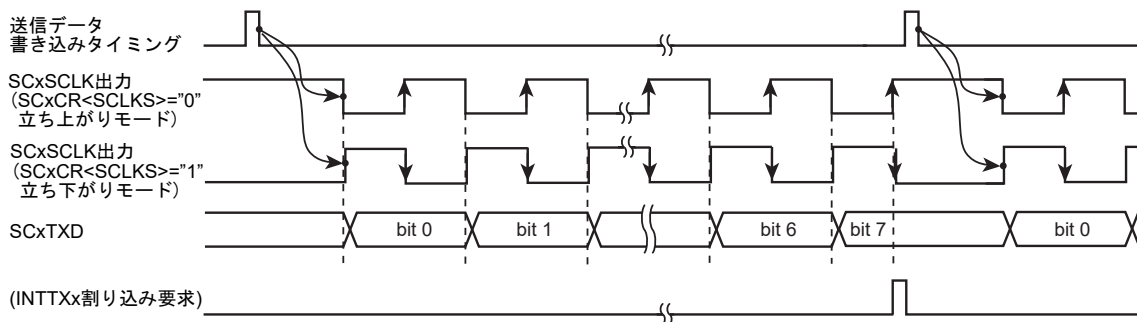
このモードには、同期クロック SCLK を出力するクロック出力モードと、外部より同期クロック SCLK を入力するクロック入力モードがあります。

以下に FIFO の使用が禁止されている状態での各動作の説明を行います。FIFO の動作については、前述の受信 FIFO 動作および送信 FIFO 動作の項を参照してください。

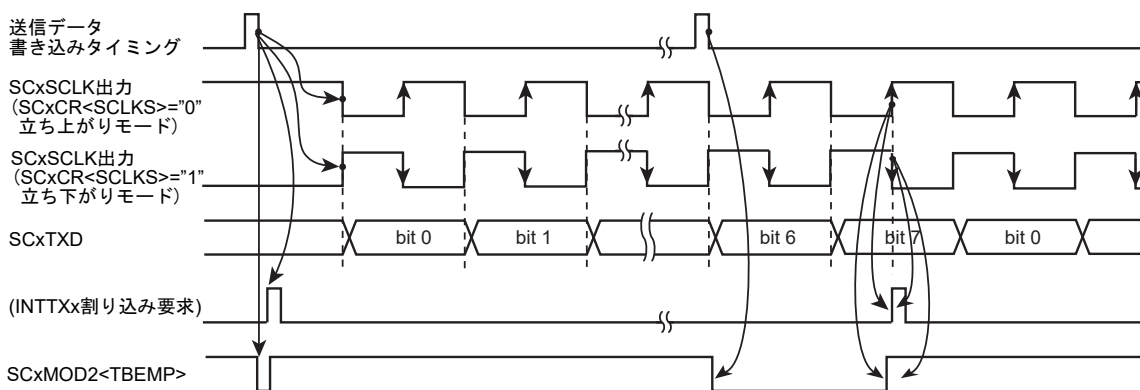
#### 13.16.1.1

##### (1)

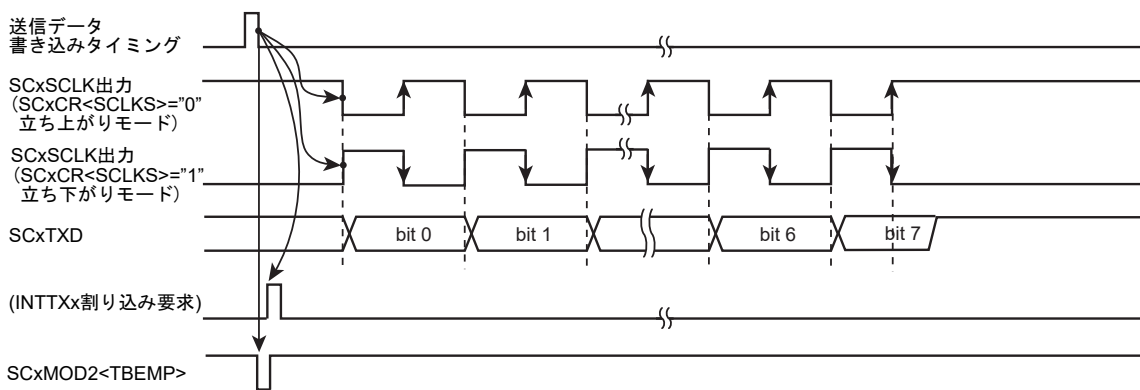
- ダブルバッファ不許可(SCxMOD2<WBUF> = "0")の場合  
送信バッファにデータを書き込むたびに、データが SCxTXD 端子から、クロックが SCxSCLK 端子より出力されます。データがすべて出力されると割り込み(INTTXx)が発生します。
- ダブルバッファ許可(SCxMOD2<WBUF> = "1")の場合  
シフトレジスタが空の状態ですべて送信バッファにデータを書き込んだとき、またはシフトレジスタのデータ送出が終了したときに送信バッファよりシフトレジスタにデータが移されます。これと同時に送信バッファ empty フラグ SCxMOD2 <TBEMP> が"1"にセットされ割り込み(INTTXx)が発生します。  
シフトレジスタのデータ送出終了時に送信バッファにデータが存在しない場合は、割り込み(INTTXx)が発生せず、クロック出力も停止します。



<WBUF> = "0" (ダブルバッファ不許可)の場合 (SCxCR<TIDLE>="10")



<WBUF> = "1" (ダブルバッファ許可)の場合(バッファデータがある場合)



<WBUF> = "1" (ダブルバッファ許可)の場合(バッファにデータがない場合) (SCxCR<TIDLE>="01")

13-15 I/O

( )

(2)

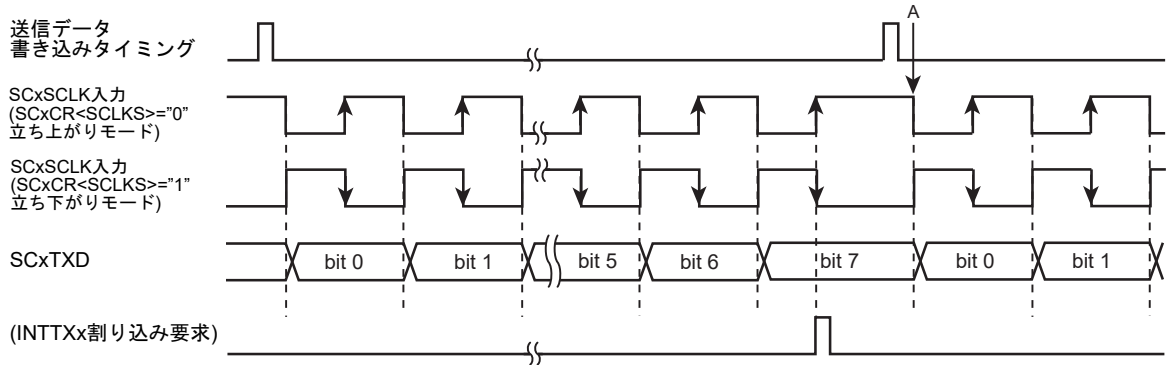
- ダブルバッファ不許可(SCxMOD2<WBUF> = "0")の場合

送信バッファにデータが書き込まれている状態でクロックが入力されると、8ビットのデータが SCxTXD 端子より出力されます。データがすべて出力されると割り込み INTTx が発生します。次の送信データは図 13-16 に示す A 点までに書き込んでください。

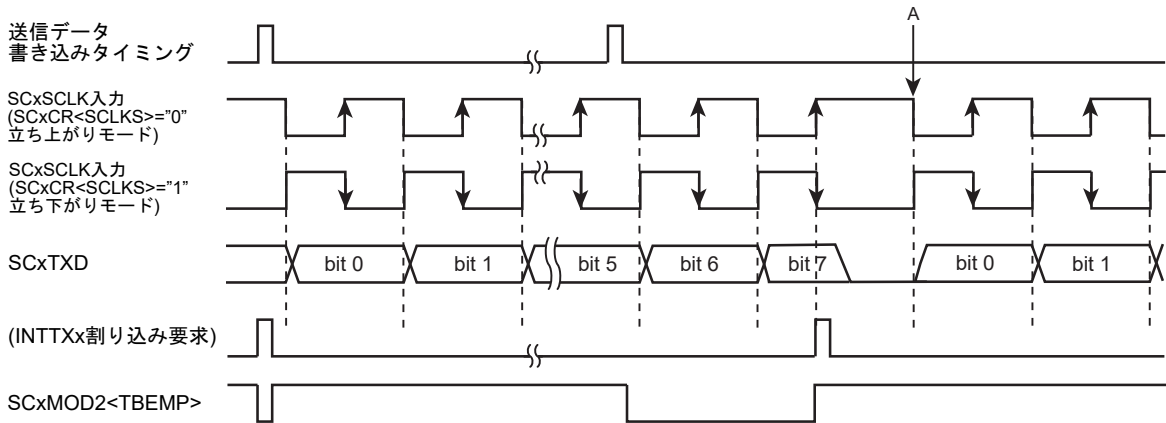
- ダブルバッファ許可(SCxMOD2<WBUF> = "1")の場合

クロックが入力される前に送信バッファにデータを書き込んだとき、または送信シフトレジスタのデータ送達が終了したときに送信バッファのデータがシフトレジスタへ移されます。これと同時に送信バッファ empty フラグ SCxMOD2<TBEMP> が "1" にセットされ、割り込み(INTTx)が発生します。

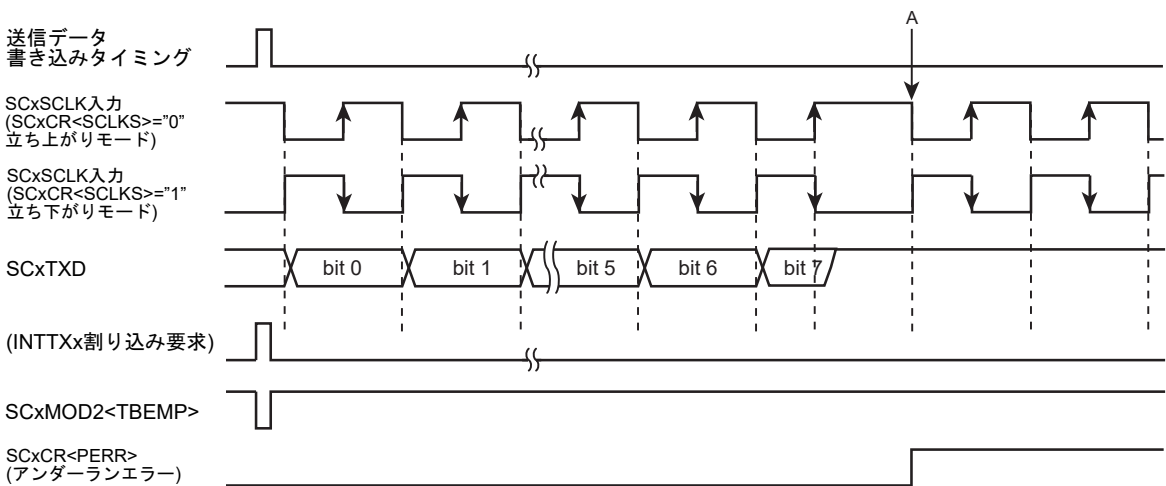
送信バッファにデータが書き込まれていない状態で、クロックが入力された場合、内部ビット数カウンタはカウントを開始しますが、アンダーランエラーがセットされ、SCxCR<TXDEMP>で設定したレベルが SCxTXD 端子に出力されます。



<WBUF> = "0" (ダブルバッファ不許可)の場合 (SCxCR<TILDE>="10")



<WBUF> = "1" (ダブルバッファ許可)の場合(バッファ2にデータがある場合) (SCxCR<TILDE>="00")



<WBUF> = "1" (ダブルバッファ許可)の場合(バッファ2にデータがない場合) (SCxCR<TXDEMP><TILDE>="100")

13-16 I/O

( )

## 13.16.1.2

## (1)

受信許可ビット SCxMOD0<RXE>を"1"にセットすることでクロック出力が開始されます。

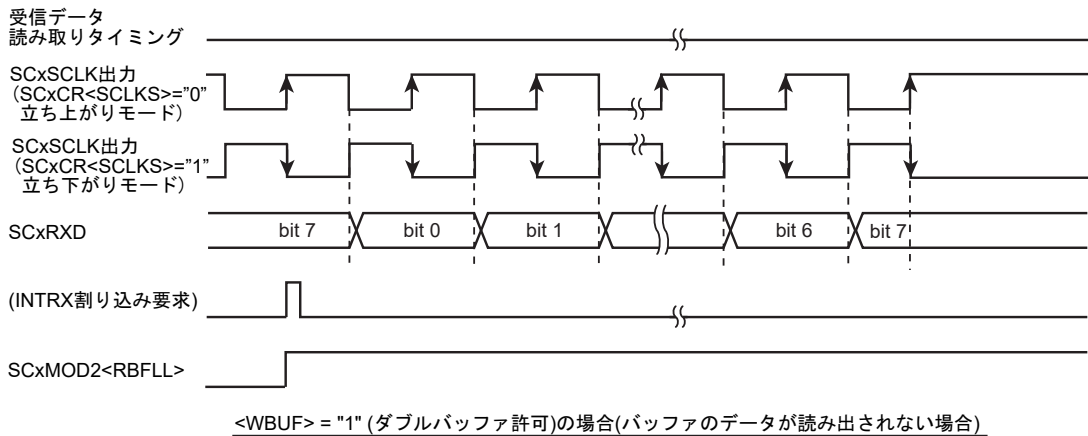
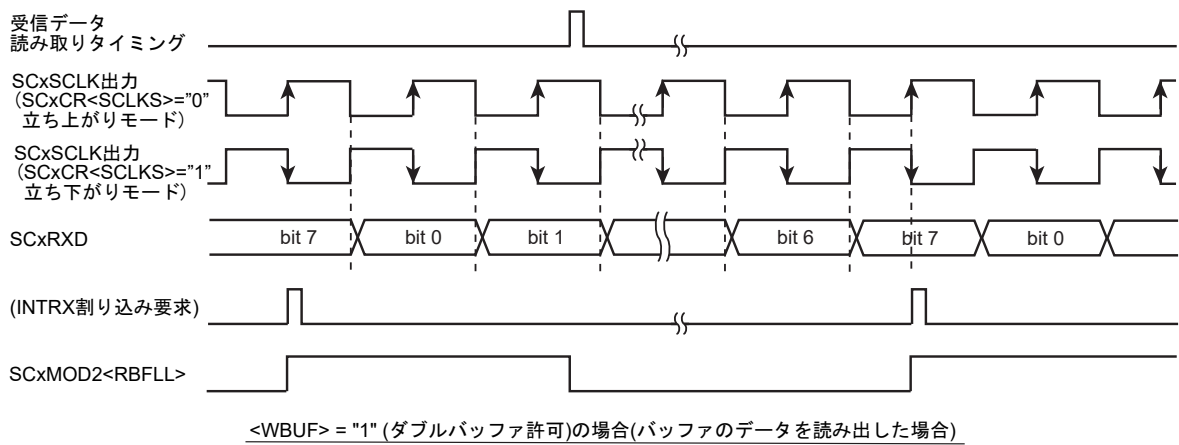
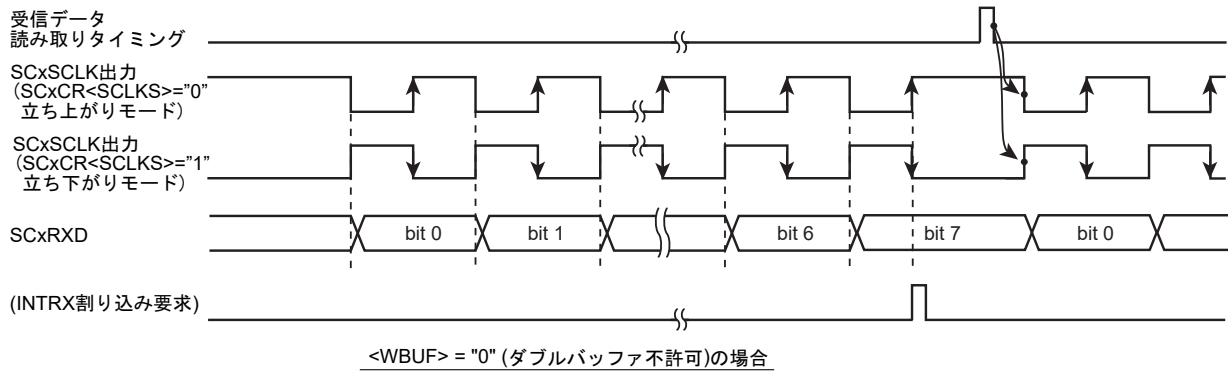
- ・ ダブルバッファ不許可(SCxMOD2<WBUF> = "0")の場合

受信データが読み出されるごとに、SCxSCLK 端子よりクロックが出力され次のデータがシフトレジスタに格納されます。8 ビットデータが受信されると、割り込み INTRX<sub>x</sub> が発生します。

- ・ ダブルバッファ許可(SCxMOD2<WBUF> = "1")の場合

シフトレジスタに格納されたデータは受信バッファに移され、続けて次のフレームを受信することができます。シフトレジスタから受信バッファにデータが移されると、受信バッファ full フラグ SCxMOD2<RBFL>が"1"にセットされ、割り込み INTRX<sub>x</sub> が発生します。

受信バッファにデータが存在する状態で、次の 8 ビット分のデータを受信完了する前に受信バッファのデータが読み出されない場合、割り込み INTRX<sub>x</sub> は発生せず、クロック出力は停止します。この状態で受信バッファのデータを読み出すと、シフトレジスタのデータを受信バッファに移し、割り込み INTRX<sub>x</sub> を発生して受信を再開します。



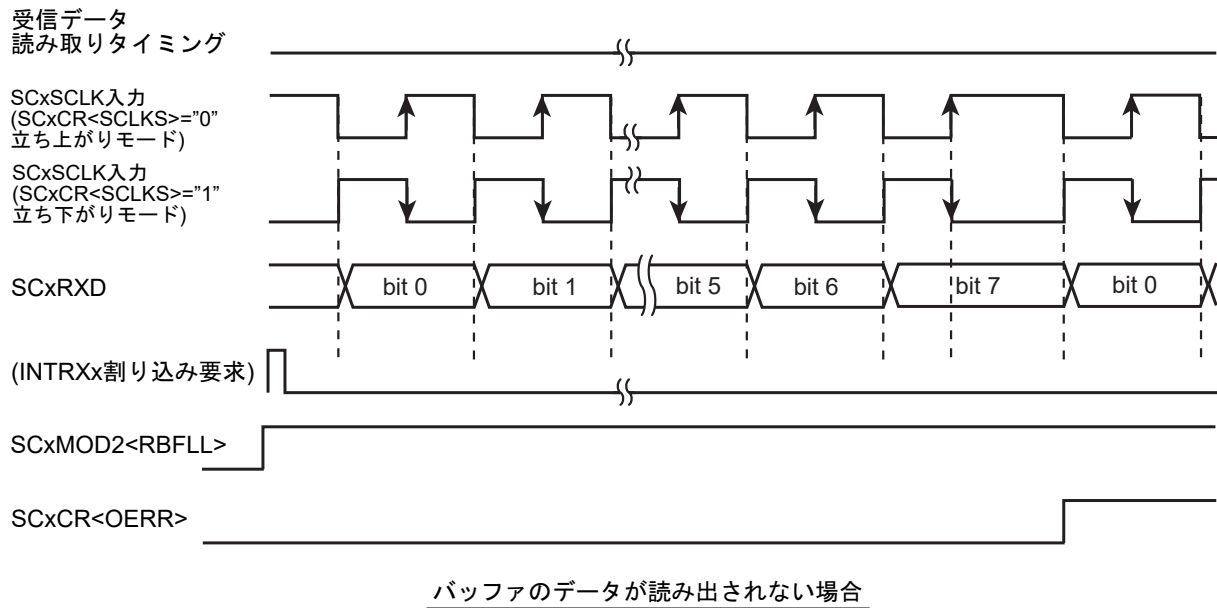
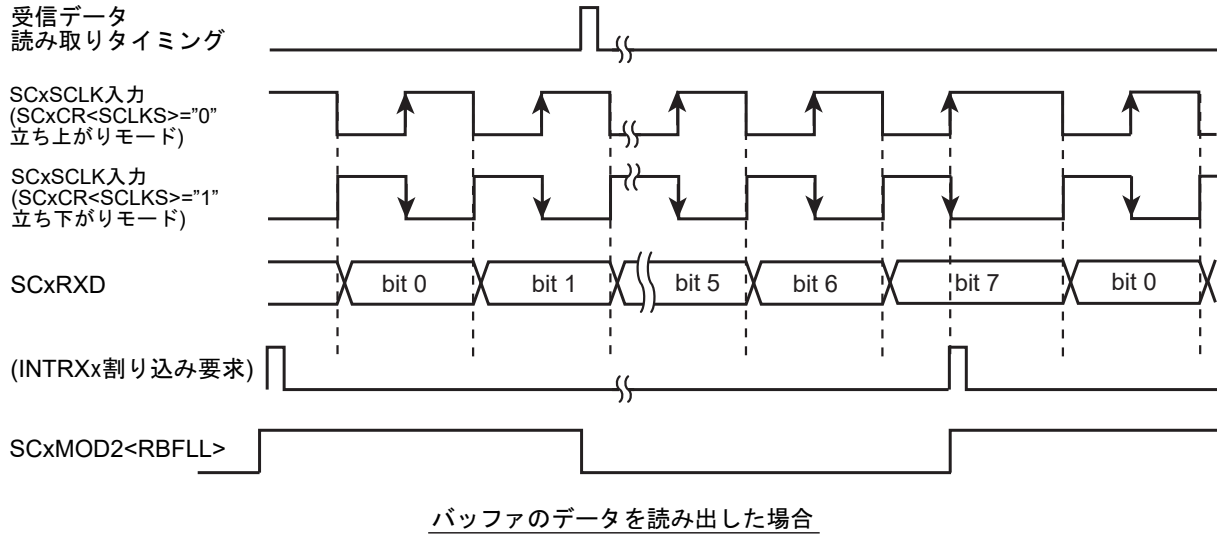
13-17 I/O

( )

(2)

クロック入力モードでは常に受信ダブルバッファが許可されており、受信したデータはシフトレジスタから受信バッファに移され、連続して次のデータを受信することができます。

受信データが受信バッファへ移されるごとに受信割り込み INTRX<sub>x</sub> が発生します。



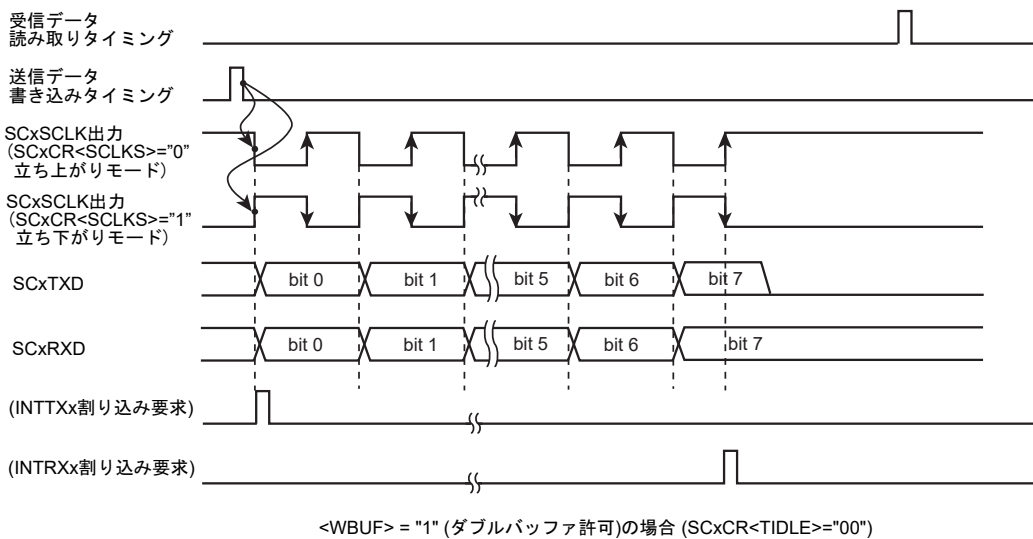
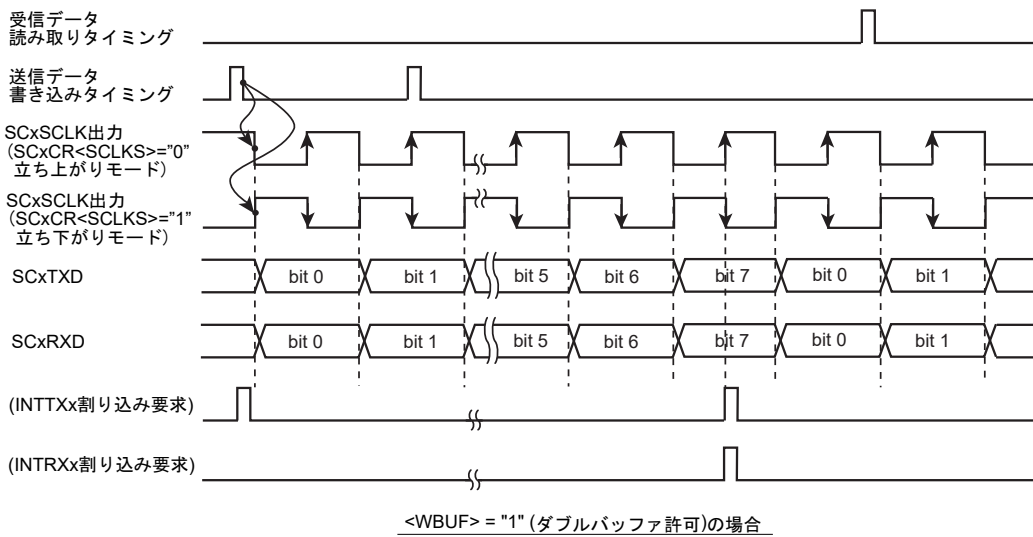
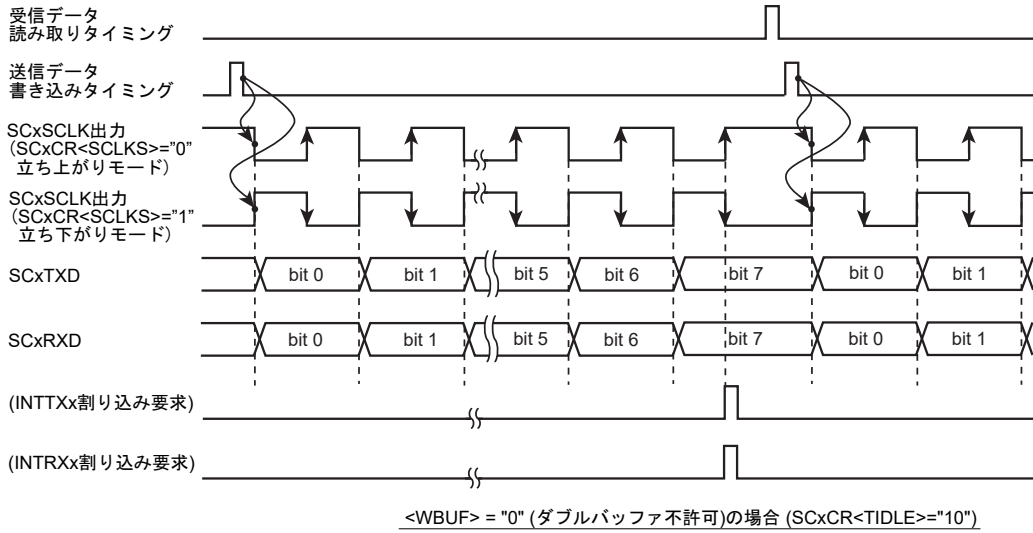
13-18 I/O

( )

## 13.16.1.3 ( )

## (1)

- ・ ダブルバッファ不許可(SCxMOD2<WBUF> = "0")の場合  
送信バッファにデータを書き込むとクロック出力を開始します。  
クロック出力によりデータが受信シフトレジスタにシフトインされ、受信割り込み(INTRXx)が発生します。それと並行して送信バッファに書き込まれたデータが、SCxTXD 端子より出力され、すべてのデータが送信されると送信割り込み(INTTXx)が発生します。この状態でクロック出力は停止します。  
受信バッファの読み出しと送信バッファへのデータ書き込みを行うと次の送受信が開始されます。受信バッファの読み出しと送信バッファの書き込み順番は任意です。両方の条件が成立した場合に再開されます。
- ・ ダブルバッファ許可(SCxMOD2<WBUF> = "1")の場合  
送信バッファにデータを書き込むとクロック出力を開始します。  
データが受信シフトレジスタにシフトインされるとデータは受信バッファに移され、割り込み(INTRXx)が発生します。受信と並行してデータが SCxTXD 端子より出力され、データがすべて出力されると割り込み(INTTXx)が発生して次のデータが送信バッファから送信シフトレジスタに移されます。  
この時、送信バッファに移すデータが存在しない(SCxMOD2 <TBEMP> = "1")または受信バッファにデータが存在している(SCxMOD2 <RBFL> = "1") 場合はクロック出力が停止します。その後は受信データの読み出しと送信データの書き込みの両方の条件が成立すると、クロック出力が再開されて次の送受信が始まります。



(2)

- ・ 送信ダブルバッファ不許可(SCxMOD2<WBUF> = "0")の場合

受信は SCxMOD2<WBUF>の設定に関わらずダブルバッファが有効になります。

送信バッファにデータが書き込まれている状態でクロックが入力されると、8ビットのデータが SCxTXD 端子より出力されると同時に8ビットのデータが受信バッファへシフトインされます。送信が終了すると割り込み(INTTXx)が発生し、受信が終了すると受信シフトレジスタから受信バッファへデータが移されると同時に割り込み(INTRXx)が発生します。

次のフレームのクロックが入力される前にデータを送信バッファへ書き込むようにしてください。(図 13-20 に示す A 点までに書き込んでください)。受信データは、次のデータの受信が終了する前に読み出してください。

- ・ ダブルバッファ許可(SCxMOD2<WBUF> = "1")の場合

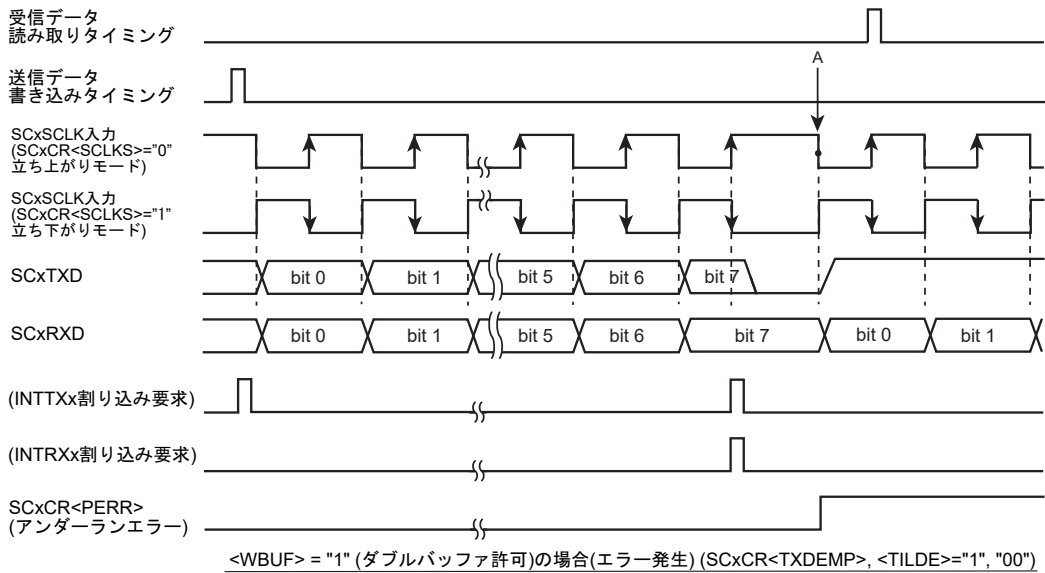
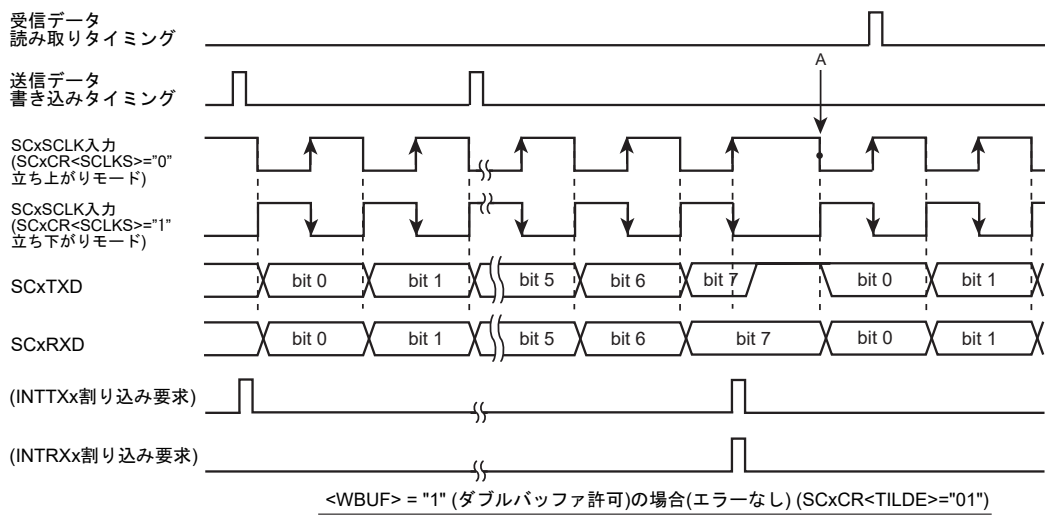
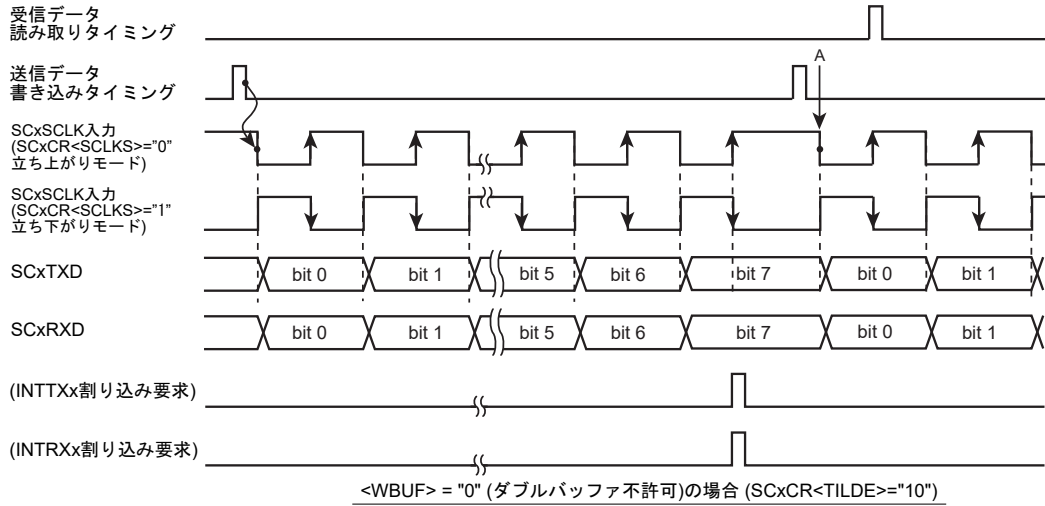
送信シフトレジスタのデータの送信が終了すると、送信バッファのデータが送信シフトレジスタへ移されると同時に割り込み(INTTXx)が発生します。平行して受信が行われ、シフトレジスタにデータが揃うと受信バッファへ移され、割り込み(INTRXx)が発生します。

次のデータのためのクロックが入力される前に送信データを送信バッファへ書き込むようにしてください。(図 13-20 に示す A 点までに書き込んでください)。受信データは、次のデータの受信が終了する前に読み出してください。

続けて次のデータのためのクロックが入力されると、送信バッファから送信シフトレジスタにデータを移して送信が始まり、並行して受信シフトレジスタでのデータ受信が行なわれます。

データの最終ビットの受信までに受信バッファのデータが読み出されていない場合はオーバランエラーが発生します。

また、次のデータのためのクロック入力までに送信バッファへ転送データが書き込まれていない場合は、アンダーランエラーがセットされ、SCxCR<TXDEMP>で設定したレベルが SCxTXD 端子に出力されます。



13-20 I/O

( )

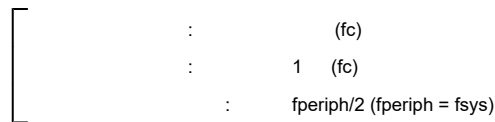
13.16.2 1 (7 UART )

SCxMOD0 <SM[1: 0]>を"01"にセットすると7ビットUARTモードになります。

このモードではパリティビットの付加が可能で、シリアルモードコントロールレジスタ (SCxCR <PE>) でパリティビット付加のイネーブル/ディセーブルを制御しています。

<PE> = "1" (イネーブル)のときは、SCxCR<EVEN>で偶数パリティ/奇数パリティを選択できます。STOP ビットの長さは SCxMOD2<SBLEN>で指定することができます。

下記フォーマットのデータを送信する場合の各コントロールレジスタの設定を示します。



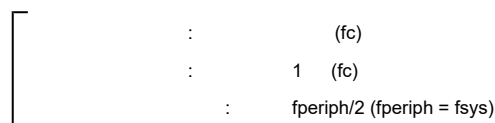
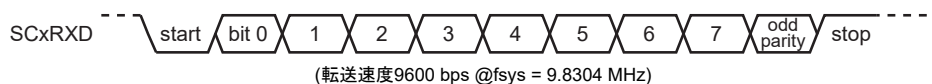
	7	6	5	4	3	2	1	0		
SCxMOD0	x	0	-	0	0	1	0	1	7	UART
SCxCR	x	1	1	x	x	x	0	0		
SCxBRCR	0	0	1	0	0	1	0	0	2400bps	
SCxBUF	*	*	*	*	*	*	*	*		

x : don't care - : no change

13.16.3 2 (8 UART )

SCxMOD0 <SM[1:0]>を"10"にセットすると8ビットUARTモードになります。このモードでは、パリティビットの付加が可能で SCxCR <PE> でパリティビット付加のイネーブル/ディセーブルを制御できます。<PE> = "1" (イネーブル)のとき、SCxCR <EVEN> で偶数パリティ/奇数パリティの選択も可能です。

下記のフォーマットのデータを受信する場合の各コントロールレジスタの設定を示します。



	7	6	5	4	3	2	1	0		
SCxMOD0	x	0	0	0	1	0	0	1	8	UART
SCxCR	x	0	1	x	x	x	0	0		
SCxBRCR	0	0	0	1	0	1	0	0	9600bps	
SCxMOD0	-	-	1	-	-	-	-	-		

x : don't care - : no change

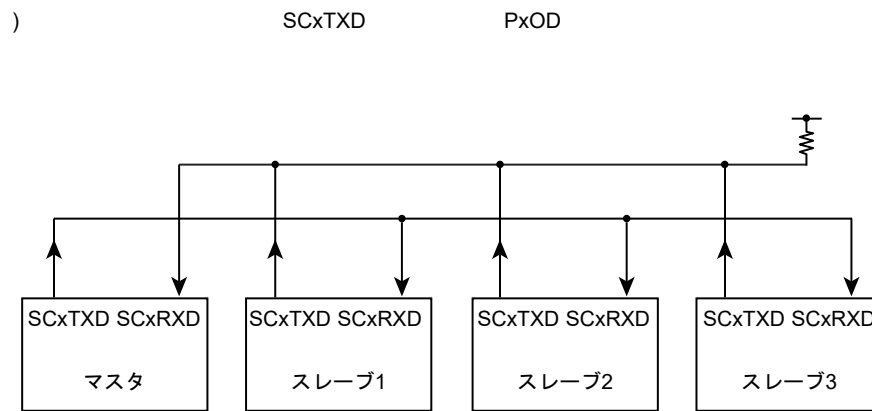
### 13.16.4 3 (9 UART )

SCxMOD0 <SM[1:0]> を "11" にセットすると 9 ビット UART モードになります。このモードでは、パリティビットの付加を禁止(SCxCR<PE> = "0")してください。

最上位ビット(9 ビット目)は、送信の場合 SCxMOD0 <TB8> に書き込みます。受信の場合 SCxCR<RB8> に格納されます。また、バッファに対する書き込み、読み出しは必ず最上位ビットの方を先に行い、SCxBUF の方を後にします。STOP ビットの長さは SCxMOD2<SBLEN>で指定することができます。

#### 13.16.4.1

9 ビット UART モードでは、ウェイクアップ機能制御ビット SCxMOD0 <WU>を"1"にすることによって、スレーブコントローラのウェイクアップ動作が可能で、SCxCR<RB8> = "1"のときのみ割り込み (INTRXx) が発生します。



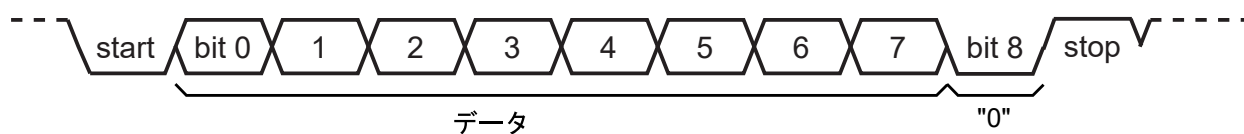
13-21

## 13.16.4.2

1. マスタおよびスレーブコントローラは9ビットUARTモードにします。
2. 各スレーブコントローラはSCxMOD0<WU>を"1"にセットし、受信可能状態とします。
3. マスタコントローラは、スレーブコントローラのセレクトコード(8ビット)を含む1フレームを送信します。このとき最上位ビット(ビット8)<TB8>は"1"にします。



4. 各スレーブコントローラは、上記フレームを受信し、自分のセレクトコードと一致すれば、<WU>を"0"にクリアします。
5. マスタコントローラは指定したスレーブコントローラ(<WU>="0"にクリアされたコントローラ)に対しデータを送信します。このとき、最上位ビット(ビット8)<TB8>は"0"にします。



6. <WU>="1"のままのスレーブコントローラは、受信データの最上位ビット(ビット8)の<RB8>が"0"であるため、割り込み(INTRXx)が発生せず、受信データを無視します。また、<WU>="0"になったスレーブコントローラがマスタコントローラにデータを送信し、この送信データで受信終了をマスタコントローラに知らせることもできます。



14

(I2C/SIO)

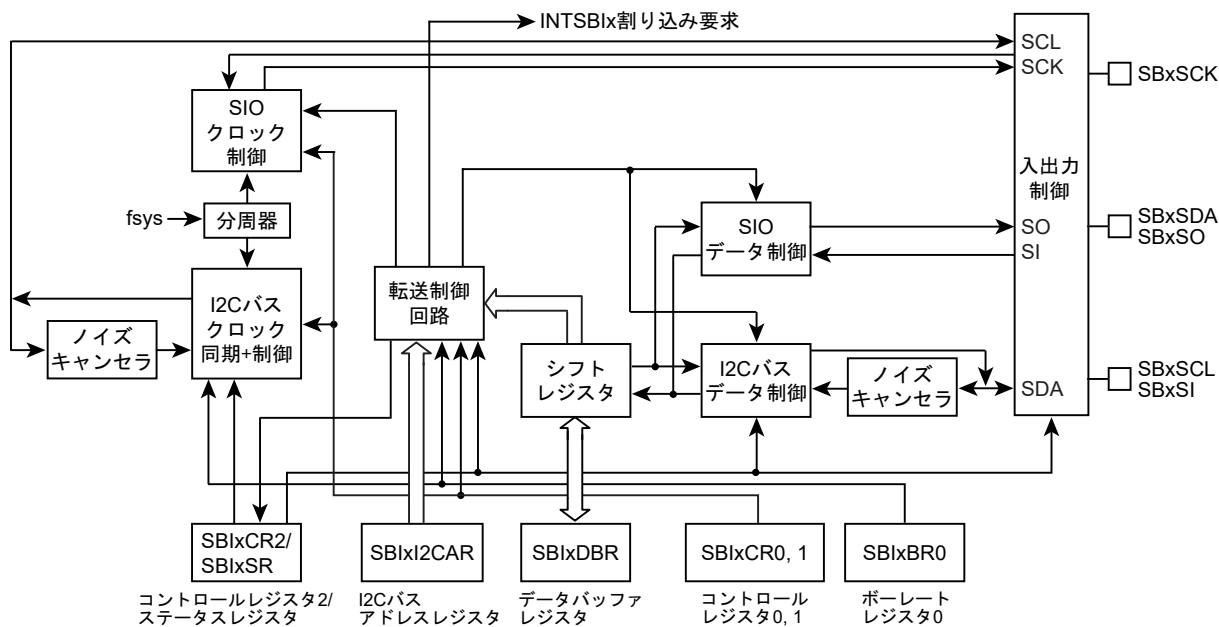
シリアルバスインタフェースは、下記の2つの動作モードを持っています。

- ・ I2C バスモード(マルチマスタ)
- ・ クロック同期式8ビット SIO モード

以下の説明中、「x」はチャンネル番号を表します。

14.1

図 14-1 にシリアルバスインタフェースのブロック図を示します。



14-1

## 14.2

シリアルバスインタフェースの制御および動作状態のモニタを行うレジスタとアドレスは以下のとおりです。

下記レジスタは使用するモードによって、機能が異なります。詳細は「14.3.1 I2C バスモード時のコントロールレジスタ」および「14.4.1 SIO モード時のコントロールレジスタ」を参照してください。

### 14.2.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)
0	SBIXCR0	0x0000
1	SBIXCR1	0x0004
	SBIXDBR	0x0008
I2C	SBIXI2CAR	0x000C
2	SBIXCR2( )	0x0010
	SBIXSR( )	
0	SBIXBR0	0x0014

## 14.3 I2C

### 14.3.1 I2C

シリアルバスインタフェースをI2Cバスモードで使用するときの制御、および動作状態のモニタは以下のレジスタで行います。

#### 14.3.1.1 SBIXCR0( 0)

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

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	SBIEN	R/W	0: 1: <SBIEN> "1" SBI SBIXCR0 <SBIEN> "1"
6-0	-	R	"0"

14.3.1.2 SBxCR1( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	BC			ACK	-	SCK2	SCK1	SCK0 / SWRMON
	0	0	0	0	1	0	0	1( 3)

Bit	Bit Symbol	Type																																																		
31-8	-	R	"0"																																																	
7-5	BC[2:0]	R/W	( 1)																																																	
			<table border="1"> <thead> <tr> <th rowspan="2">&lt;BC&gt;</th> <th colspan="2">&lt;ACK&gt; = 0</th> <th colspan="2">&lt;ACK&gt; = 1</th> </tr> <tr> <th></th> <th></th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>000</td> <td>8</td> <td>8</td> <td>9</td> <td>8</td> </tr> <tr> <td>001</td> <td>1</td> <td>1</td> <td>2</td> <td>1</td> </tr> <tr> <td>010</td> <td>2</td> <td>2</td> <td>3</td> <td>2</td> </tr> <tr> <td>011</td> <td>3</td> <td>3</td> <td>4</td> <td>3</td> </tr> <tr> <td>100</td> <td>4</td> <td>4</td> <td>5</td> <td>4</td> </tr> <tr> <td>101</td> <td>5</td> <td>5</td> <td>6</td> <td>5</td> </tr> <tr> <td>110</td> <td>6</td> <td>6</td> <td>7</td> <td>6</td> </tr> <tr> <td>111</td> <td>7</td> <td>7</td> <td>8</td> <td>7</td> </tr> </tbody> </table>	<BC>	<ACK> = 0		<ACK> = 1						000	8	8	9	8	001	1	1	2	1	010	2	2	3	2	011	3	3	4	3	100	4	4	5	4	101	5	5	6	5	110	6	6	7	6	111	7	7	8	7
<BC>	<ACK> = 0		<ACK> = 1																																																	
000	8	8	9	8																																																
001	1	1	2	1																																																
010	2	2	3	2																																																
011	3	3	4	3																																																
100	4	4	5	4																																																
101	5	5	6	5																																																
110	6	6	7	6																																																
111	7	7	8	7																																																
4	ACK	R/W	0: 1:  0: 1:																																																	
3	-	R	"1"																																																	
2-1	SCK[2:1]	R/W	SBxSCL <SCK[2:0]>@ ( 2)																																																	
0	SCK[0]	W	<table border="1"> <tbody> <tr> <td>000</td> <td>n = 5</td> <td rowspan="8">                     システムクロック: fsys                      { クロックギア: fc/1 }                      周波数 = <math>\frac{fsys}{2^n + 72}</math> [Hz]                 </td> </tr> <tr> <td>001</td> <td>n = 6</td> </tr> <tr> <td>010</td> <td>n = 7</td> </tr> <tr> <td>011</td> <td>n = 8</td> </tr> <tr> <td>100</td> <td>n = 9</td> </tr> <tr> <td>101</td> <td>n = 10</td> </tr> <tr> <td>110</td> <td>n = 11</td> </tr> <tr> <td>111</td> <td>reserved</td> </tr> </tbody> </table>	000	n = 5	システムクロック: fsys { クロックギア: fc/1 } 周波数 = $\frac{fsys}{2^n + 72}$ [Hz]	001	n = 6	010	n = 7	011	n = 8	100	n = 9	101	n = 10	110	n = 11	111	reserved																																
000	n = 5	システムクロック: fsys { クロックギア: fc/1 } 周波数 = $\frac{fsys}{2^n + 72}$ [Hz]																																																		
001	n = 6																																																			
010	n = 7																																																			
011	n = 8																																																			
100	n = 9																																																			
101	n = 10																																																			
110	n = 11																																																			
111	reserved																																																			
	SWRMON	R	0: 1:																																																	

- 1) SIO <BC[2:0]> "000"
- 2) SCL 14.3.2.2
- 3) <SCK[0]/SWRMON> "1" SBxCR2 SIO  
<SCK[0]> "0"
- 4) <SCK[2:0]>=000
- 5) <BC[2:0]>="001" <ACK>="0" SBxSCL  
SBxSCL "L" "2"

14.3.1.3 SBIXCR2( 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MST	TRX	BB	PIN	SBIM		SWRST	
	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	MST	W	/
6	TRX	W	/
5	BB	W	/
4	PIN	W	INTSBIX 0: - 1:
3-2	SBIM[1:0]	W	( 1 ) 00: ( ) ( 2 ) 01: SIO 10: I2C ( 3 ) 11: Reserved
1-0	SWRST[1:0]	W	"10" "01" 14.3.2.16

- 1)
- 2)
- 3) I2C SBxSDA /SBxSCL "High"
- 4) SBIXCR2 SBIXSR

14.3.1.4 SBIXSR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
	0	0	0	1	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	MST	R	/
			0: 1:
6	TRX	R	/
			0: 1:
5	BB	R	I2C
			0: 1:
4	PIN	R	INTSBIX
			0: 1:
3	AL	R	
			0: - 1:
2	AAS	R	
			0: - 1: ( )
1	AD0	R	
			0: - 1:
0	LRB	R	
			0: "0" 1: "1"

14.3.1.5 SBIXBR0( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	-	R	"1"
6	I2SBI	R/W	IDLE 0: 1:
5-1	-	R	"1"
0	-	R/W	"0"



14.3.1.7 SBiI2CAR(I2C )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SA							ALS
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7-1	SA[6:0]	R/W	
0	ALS	R/W	0: 1: ( )

- 1) <ALS> "0" "1"
- 2) SBiI2CAR "0x00" ("0x00" ) I2C  
START ("0x01")

14.3.2

14.3.2.1

SBIxCR2<SBIM[1:0]>で動作モードを設定します。I2C バスモードで使用するとき、<SBIM[1:0]>を"10"に設定してください。

- 1) SBxSDA /SBxSCL "High"
- 2) I2C SBxSDA /SBxSCL "High"

14.3.2.2

(1)

SBIxCR1 <SCK[2:0]>で、マスタモード時に SBxSCL 端子から出力されるシリアルクロックの最大転送周波数を選択します。



$$t_{LOW} = 2^{n-1}/f_{sys} + 58/f_{sys}$$

$$t_{HIGH} = 2^{n-1}/f_{sys} + 14/f_{sys}$$

$$f_{SCL} = 1/(t_{LOW} + t_{HIGH})$$

$$= \frac{f_{sys}}{2^n + 72}$$

SBIxCR1<SCK[2:0]>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

14-2

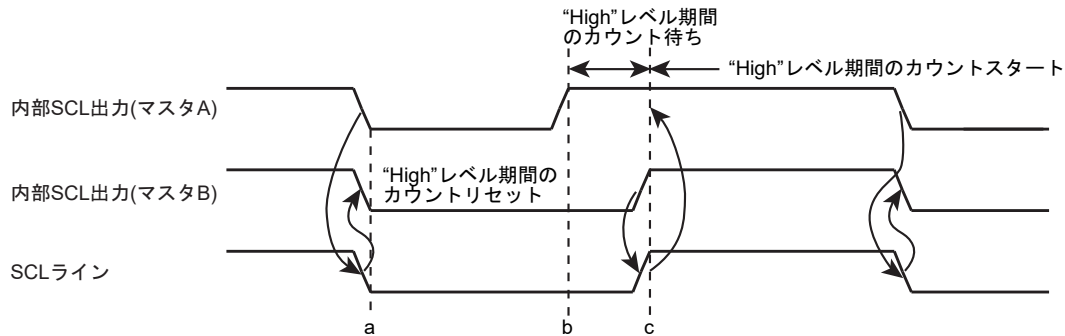
) / f<sub>sys</sub> 100 kHz/400 kHz SCL

(2)

I2C バスでは、端子の構造上バスをワイヤードアンドで駆動させるため、クロックラインを最初に"Low"レベルに引いたマスタが、"High"レベルを出力しているマスタのクロックを無効にします。このため、"High"レベルを出力しているマスタは、これを検出し対応する必要があります。

シリアルバスインタフェース回路はクロック同期化機能をもっており、バス上に複数のマスタが存在する場合でも、正常に転送が行われます。

クロック同期の手順を、バス上に2つのマスタが同時に存在した場合を例に挙げて以下に示します。



14-3

a 点でマスタ A が内部 SCL 出力を "Low" レベルに引くことで、バスの SCL ラインは "Low" レベルになります。マスタ B はこれを検出し、マスタ B の "High" レベル期間のカウントをリセットし、内部 SCL 出力を "Low" レベルに引きます。

b 点でマスタ A は "Low" レベル期間のカウントを終わり、内部 SCL 出力を "High" レベルにします。しかし、マスタ B がバスの SCL ラインを "Low" レベルに保持し続けているので、マスタ A は "High" レベル期間のカウントを止めます。マスタ A は、c 点でマスタ B が内部 SCL 出力を "High" レベルにし、バスの SCL ラインが "High" レベルになったことを検出後、"High" レベル期間のカウントを始めます。その後、"High" レベル期間のカウントを終了したマスタ A が内部 SCL 出力を "Low" に引くことでバスの SCL ラインは "Low" レベルになります。

以上のようにバス上のクロックは、バスに接続されているマスタの中で最も短い "High" レベル期間をもつマスタと最も長い "Low" レベル期間をもつマスタによって決定されます。

### 14.3.2.3

SBIxCR1<ACK>を "1" に設定するとアクノリッジメントモードとして動作します。

マスタモードのときには、アクノリッジ信号のためのクロックを 1 クロック付加します。

スレーブモードのときはアクノリッジ信号のためのクロックをカウントします。

トランスマッタモードのときには、アクノリッジのためのクロック期間中 SBxSDA 端子を開放し、レシーバからのアクノリッジ信号を受信できる状態にします。

レシーバモードのときはクロック期間中 SBxSDA 端子を "Low" レベルに引き、更に、スレーブモードのときにゼネラルコールアドレスを受信した場合にもアクノリッジのためのクロック期間中、SBxSDA 端子を "Low" レベルに引き、アクノリッジ信号を発生します。ゼネラルコールのセカンドバイトは内容によりアクノリッジ信号の発生有無を制御する必要がありますので、ソフトウェアによりアクノリッジ信号の制御を行なってください。

<ACK>を "0" に設定すると、非アクノリッジメントモードとして動作し、マスタモードのときにアクノリッジ信号のためのクロックを発生しません。スレーブモードのときはアクノリッジ信号のためのクロックをカウントしません。

### 14.3.2.4

SBIxCR1<BC[2:0]>により、次に送受信するデータのビット数を選択します。

<BC[2:0]>はスタートコンディションにより "000" にクリアされるため、スレーブアドレス、方向ビットの転送は必ず 8 ビットで行われます。それ以外のときは <BC[2:0]> は一度設定された値を保持します。

## 14.3.2.5

スレーブアドレスを認識するアドレッシングフォーマットで動作させるときは、SBIxI2CAR<ALS>に"0"を設定し、SBIxI2CAR <SA[6:0]>にスレーブアドレスを設定します。

スレーブアドレスを認識しないフリーデータフォーマットで使用する際には<ALS>を"1"に設定します。なお、フリーデータフォーマットで使用了した場合、スレーブアドレスと方向ビットの認識は行われず、スタートコンディション直後からデータとして扱われます。

## 14.3.2.6 /

SBIxCR2<MST>を"1"に設定すると、マスタデバイスとして動作します。

<MST>を"0"に設定すると、スレーブデバイスとして動作します。

なお、<MST>はバス上のストップコンディションの検出、またはアービトレーションロストの検出で、ハードウェアにより"0"にクリアされます。

## 14.3.2.7 /

SBIxCR2<TRX>を"1"に設定すると、トランスミッタとして動作します。

SBIxCR2<TRX>を"0"に設定すると、レシーバとして動作します。

フリーデータフォーマットで使用する場合、<TRX>はハードウェアによって変化することはありません。

アドレッシングフォーマットで使用する際には、<TRX>は下記のように設定されます。

## (1)

マスタモード時は、スレーブアドレスと方向ビットを送信後、スレーブデバイスからアクノリッジが返ってくると、ハードウェアにより、下記のように<TRX>が設定されます。アクノリッジが返ってこないときは、以前の状態を保ちます。

- ・ 送信した方向ビットが"1"の場合、<TRX>は"0"に設定されます。
- ・ 送信した方向ビットが"0"の場合、<TRX>は"1"に設定されます。

## (2)

スレーブモード時は、アドレッシングフォーマットでデータ転送を行う場合に、下記の条件が成立するとマスタデバイスの送ってくる方向ビットにあわせ、<TRX>が設定されます。

- ・ 受信したスレーブアドレスが SBIxI2CAR にセットした値と同じとき
- ・ ゼネラルコールを受信したとき

<TRX>は下記のように設定されます。

- ・ 受信した方向ビットが"1"の場合、<TRX>は"0"に設定されます。
- ・ 受信した方向ビットが"0"の場合、<TRX>は"1"に設定されます。

### 14.3.2.8

SBIxSR<BB>を読み出すことで、バスの状態を知ることができます。

<BB>は、バス上のスタートコンディションを検出すると"1"にセットされ、ストップコンディションを検出すると"0"にクリアされます。

<BB>が"1"のときをバスビジー状態、<BB>が"0"のときをバスフリー状態と呼びます。

マスタデバイスは、バスフリーの状態でのみスタートコンディションを発生することができます。スタートコンディションを発生する前には、必ず<BB>が"0"であることを確認してください。

<BB>が"1"の状態ですらスタートコンディションを発生すると、スタートコンディションは発生せず、アービトレーションロストが発生します。

### 14.3.2.9

割り込み要求(INTSBIx)が発生すると、SBIxCR2<PIN>が"0"にクリアされ、割り込みサービス要求状態になります。<PIN>が"0"の間、SBxSCL 端子を"Low"レベルにします。

<PIN>は SBIxDBR にデータを書き込むか、SBIxDBR からデータを読み出すと"1"にセットされます。プログラムで<PIN>に"1"を書き込むと"1"にセットされますが、"0"を書き込んでも"0"にクリアされません。

<PIN>が"1"にセットされると SBxSCL 端子が開放されます。<PIN>が"1"にセットされてから SBxSCL 端子が開放されるまで t<sub>LOW</sub> の時間がかかります。

) <PIN> "0" (INTSBIx )

### 14.3.2.10

I2C バスではマルチマスタ(1 つのバス上で同時に 2 つ以上のマスタが存在する)が可能なため、転送されるデータの内容を保証するためにバスのアービトレーション手順が必要となります。

I2C バスではバスのアービトレーションに SDA ラインのデータを使用します。

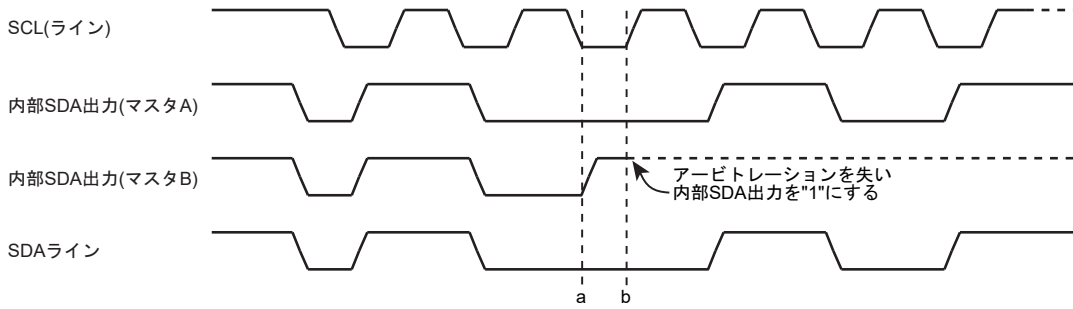
アービトレーションの手順を、バス上に 2 つのマスタが同時に存在した場合を例に挙げて以下に示します。

a 点のビットまでマスタ A、マスタ B とも同じデータを出力し、a 点でマスタ A が"Low"レベルを出力、マスタ B が"High"レベルを出力すると、バスの SDA ラインはワイヤードアンドで駆動されるためにマスタ A によって"Low"レベルに引かれます。

b 点でバスの SCL ラインが立ち上がると、スレーブデバイスは SDA ラインデータ、すなわち、マスタ A のデータを取り込みます。このとき、マスタ B の出力したデータは無効になります。

マスタ B のこの状態を"アービトレーションロスト"と呼びます。アービトレーションロストが発生したマスタ B は、SDA 端子を開放し、他のマスタの出力するデータに影響を及ぼさないようにします。

もし、複数のマスタが 1 ワード目でまったく等しいデータを送信した場合、アービトレーションの手順は 2 ワード目以降も継続されます。



14-4

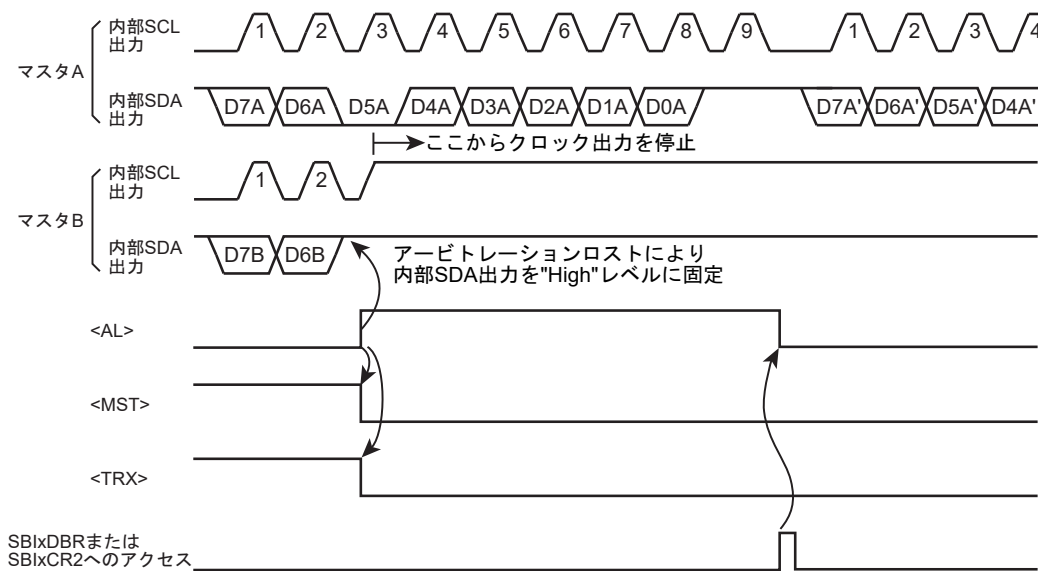
バスの SDA ラインのレベルと内部 SDA 出力のレベルの比較は、SCL ラインの立ち上がりで行います。この比較結果が不一致の場合アービトレーションロストになり、SBIxSR<AL> が"1"にセットされます。

アービトレーションロストが発生すると、SBIxSR<MST>と<TRX>は"0"にクリアされ、スレーブレシーバモードになり、SCL クロックの出力を停止します。

スレーブアドレス転送中にアービトレーションロストが発生したデバイスは、通常のスレーブデバイスのように、他のマスタが送信するスレーブアドレスを受信します。

受信したスレーブアドレスが SBIxI2CAR<SA>と一致した場合、<PIN>が"0"にクリアされ、INTSBIx が発生します。一致しなかった場合、<PIN>は"1"のままですが、INTSBIx が発生します。

<AL>は、SBIxDBR にデータを書き込むか、SBIxDBR からデータを読み込む、または SBIxCR2 にデータを書き込むと"0"にクリアされます。



14-5 B (D7A=D7B, D6A=D6B)

14.3.2.11

SBIxSR<AAS>は、スレーブモード時、アドレス認識モード(SBIxI2CAR<ALS>="0")のとき、ゼネラルコールアドレスまたは SBIxI2CAR<SA>にセットした値と同じスレーブアドレスを受信すると"1"にセットされます。

フリーデータフォーマット(<ALS>="1")のときは、最初の1ワードが受信されると"1"にセットされます。

<AAS> は SBIxDBR にデータを書き込むか、SBIxDBR からデータを読み出すと"0"にクリアされます。

#### 14.3.2.12

SBIxSR<AD0>は、スレーブモード時、ゼネラルコール(スタートコンディション後の8ビットのデータがすべて"0")を受信したとき"1"にセットされます。

バス上のスタートコンディション、またはストップコンディションが検出されると"0"にクリアされます。

#### 14.3.2.13

SBIxSR<LRB>には、SBxSCL 端子の立ち上がりで取り込まれた SBxSDA 端子の値がセットされます。アクノリッジメントモードのとき、INTSBIx 割り込み要求発生直後に SBIxSR<LRB> を読み出すと、ACK 信号が読み出されます。

#### 14.3.2.14

##### (SBIxDBR)

SBIxDBR をリード/ライトすることで、受信データの読み出し/送信データの書き込みを行います。

また、マスタモード時に、SBIxDBR にスレーブアドレスと方向ビットを設定後、スタートコンディションを発生することで、スレーブデバイスに対し、スレーブアドレスと方向ビットを送信することができます。

#### 14.3.2.15

##### (SBIxBR0)

SBIxBR0<I2SBI>は IDLE モードに遷移した際にシリアルバスインタフェース回路の動作の許可/禁止を設定するレジスタです。スタンバイモードに移る命令を実行する前に、あらかじめ設定してください。

#### 14.3.2.16

シリアルバスインタフェース回路が、外部からのノイズによりロックした場合、ソフトウェアリセット機能を使い、シリアルバスインタフェース回路を初期化することができます。

I2C モードで SBIxCR2<SWRST[1:0]>へ、最初に"10"、次に"01"をライトすると、ソフトウェアリセットが発生します。ライトの際、SBIxCR2<MST><TRX><BB><PIN> は "0000"、SBIxCR2<SBIM[1:0]> は "10" (I2C バスモード)を指定してください。ソフトウェアリセットが発生すると、シリアルバスインタフェース回路にリセット信号が入力され、回路が初期化されます。このとき、すべてのコントロールレジスタとステータスフラグはリセット直後の値となります。また、<SWRST>は、シリアルバスインタフェースを初期化すると、自動的に"00"にクリアされます。

)

I2C

PORT

## 14.3.3

## 14.3.3.1

最初に SBIxCR1<ACK>と<SCK[2:0]>を設定します。SBIxCR1[7:5]には"0"を書き込んでください。

次に SBIxI2CAR にスレーブアドレス<SA[6:0]>と<ALS> (アドレッシングフォーマット時には、<ALS> = "0")を設定します。

それから、シリアルバスインタフェース端子の状態が"High"になっていることを確認し、SBIxCR2<MST><TRX><BB>に"0", <PIN>に"1", <SBIM[1:0]>に"10", <SWRST[1:0]>に"00"を書き込み、初期状態をスレーブレシーバモードにします。

)

	7	6	5	4	3	2	1	0		
SBIxCR1	0	0	0	X	0	X	X	X	ACK	SCL
SBIxI2CAR	X	X	X	X	X	X	X	X		
SBIxCR2	0	0	0	1	1	0	0	0		

) X; Don't care

## 14.3.3.2

スタートコンディションとスレーブアドレスを、次の手順で発生します。

はじめに、バスフリー状態(<BB> = "0")を確認します。次に、SBIxCR1<ACK>に"1"を書き込んで、アクノリッジメントモードに設定します。また、SBIxDBR に、送信するスレーブアドレスと方向ビットのデータを書き込みます。

<BB> = "0" の状態で、SBIxCR2<MST><TRX><BB><PIN>にそれぞれ"1"を書き込むと、バス上にスタートコンディションが発生します。

スタートコンディションの発生に次いで、SBxSCL 端子から 9 クロックを出力します。

最初の 8 クロックで、SBIxDBR に設定したスレーブアドレスと方向ビットを出力します。

9 クロック目で、SBxSDA 端子を解放し、スレーブデバイスからのアクノリッジ信号を受信します。

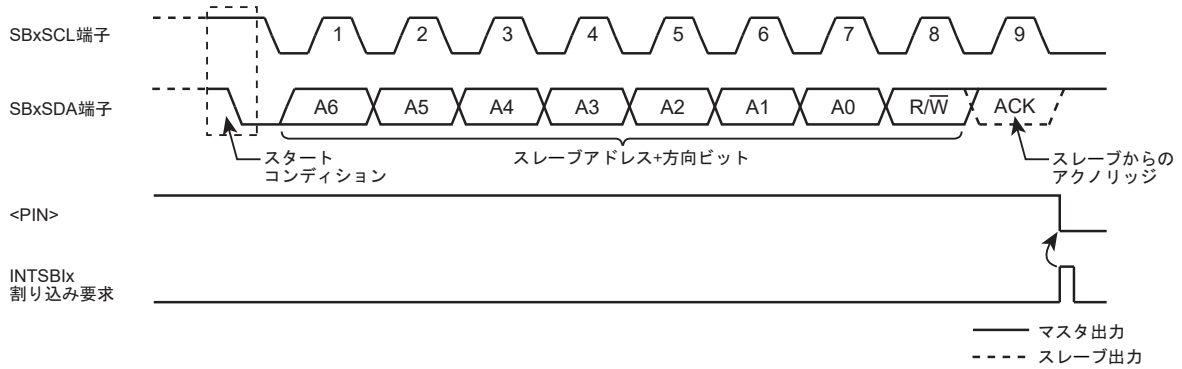
9 クロック目の立ち下がり、INTSBIx 割り込み要求が発生し、<PIN> = "0"にされます。

<PIN> = "0" の間、SBxSCL 端子を"Low"レベルにひきます。また、スレーブデバイスからのアクノリッジ信号が返ってきたときのみ、INTSBIx 割り込み要求の発生により、送信した方向ビットに合わせて<TRX>は変化します。

)

SBIxDBR

	7	6	5	4	3	2	1	0
Reg.	SBIxSR							
Reg.	Reg.AND 0x20							
if Reg.	0x00							
Then								
SBIxCR1	X	X	X	1	0	X	X	X
SBIDBR	X	X	X	X	X	X	X	X
SBICR2	1	1	1	1	1	0	0	0
	INTSBix							



14-6

14.3.3.3 1

1 ワード転送終了の INTSBix 割り込みの処理で<MST>をテストし、マスタモード/スレーブモードの判断をします。

(1) (<MST> = "1")

<TRX>をテストし、トランスミッタ/レシーバの判断をします。

(a) (<TRX> = "1")

<LRB>をテストします。<LRB> = "1"のとき、レシーバはデータを要求していないのでストップコンディションを発生する処理(後記参照)を行ってデータ転送を終了します。

<LRB> = "0"のときレシーバが次のデータを要求しています。

次に転送するデータのビット数が8ビットのとき SBIDBR に転送データを書き込みます。8ビット以外のときは<BC[2:0]>, <ACK>を設定し、転送データを SBIDBR に書き込みます。

データを書き込むと<PIN>が"1"になり SBxSCL 端子から次の 1 ワードデータ転送用のシリアルクロックが発生され、SBxSDA 端子から 1 ワードのデータが転送されます。

転送終了後 INTSBIx 割り込み要求が発生し、<PIN>が"0"になり SBxSCL 端子を"Low"レベルに引きます。

複数ワードの転送が必要な場合は上記<LRB>のテストから繰り返します。

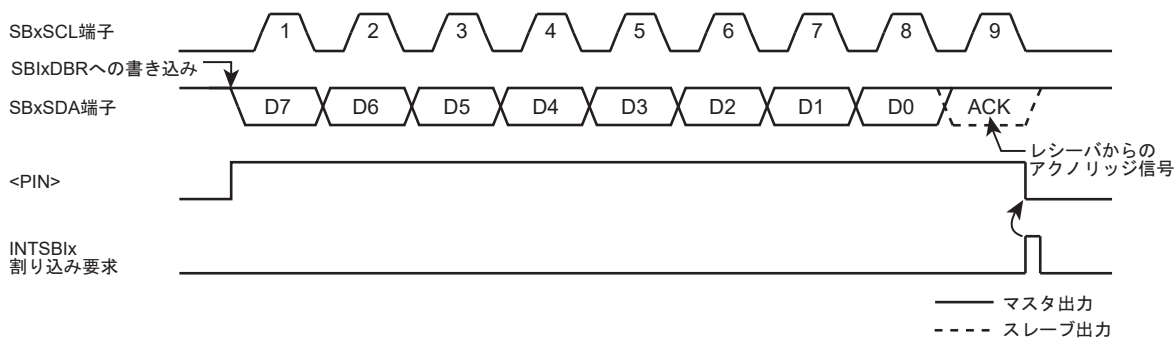
INTSBIx

```

if MST = 0
Then
if TRX = 0
Then
if LRB = 0
Then
SBIXCR1      X  X  X  X  0  X  X  X
SBIXDBR      X  X  X  X  X  X  X  X
    
```

ACK

) X; Don't care



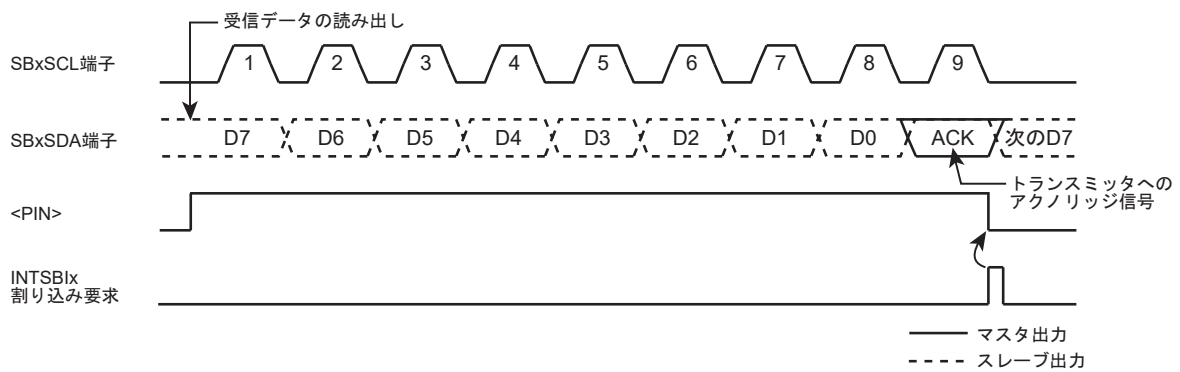
14-7 <BC[2:0]>="000",<ACK>="1"

(b) (<TRX> = "0")

次に転送するデータのビット数が 8 ビットの場合は SBxIDBR に転送データを書き込みます。8 ビット以外のときは<BC[2:0]>を設定し、SBxSCL 端子を解放するために SBxIDBR から受信データを読み出します(スレーブアドレス送信直後のリードデータは不定です)。

データを読み出すと<PIN>は"1"になり、次の 1 ワードデータ転送用の SCL クロックを SBxSCL 端子に出力します。最後のビットでアクノリッジ信号の"Low"レベルのタイミングで"0"を SBxSDA 端子に出力します。

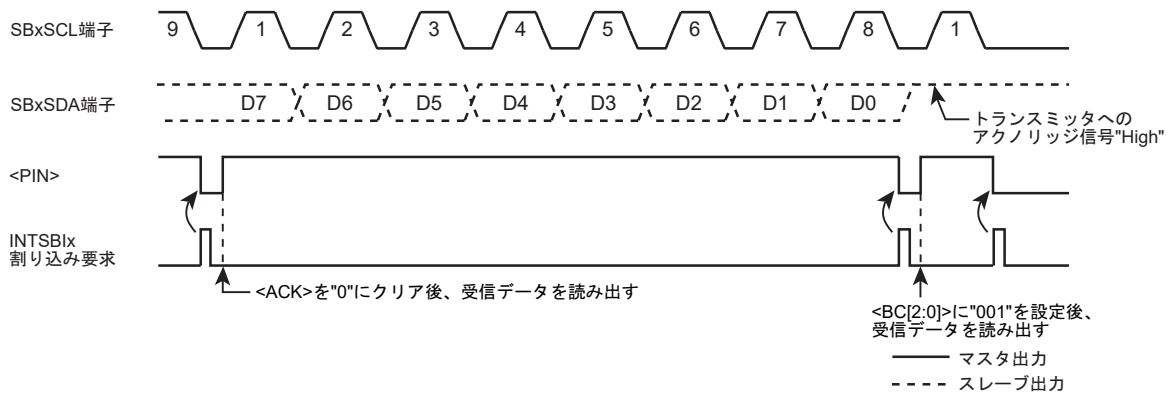
その後、INTSBx 割り込み要求が発生し、<PIN>が"0"になり SBxSCL 端子を"Low"レベルに引きます。SBxIDBR から受信データを読み出すたびに 1 ワードの転送クロックとアクノリッジを出力します。



14-8 <BC[2:0]>="000", <ACK>="1"

トランスマッタに対してデータの送信を終了させるときは、最後に受信したいデータの 1 ワード手前のデータを読み出す前に<ACK>を"0"にします。これにより、最終データのアクノリッジのためのクロックは発生されなくなります。転送終了の割り込み要求発生後の処理で、<BC[2:0]> = "001"に設定し、データを読み出すと、1 ビット転送のためのクロックを発生します。このときマスタはレシーバなのでバスの SDA ラインは"High"レベルを保ちます。トランスマッタは ACK 信号としてこの"High"レベルを受信するので、レシーバはトランスマッタへ送信終了を知らせることができます。

この 1 ビット転送の受信終了割り込み要求後の処理で、ストップコンディションを発生させ、データ転送を終了させます。



14-9

例:データを N 回受信する場合

INTSB <sub>Ix</sub>	(		)							
		7	6	5	4	3	2	1	0	
SBI <sub>x</sub> CR1		X	X	X	X	0	X	X	X	ACK
Reg.		SBI <sub>x</sub> DBR								

INTSB <sub>Ix</sub>	(								1~(N-2)	)
		7	6	5	4	3	2	1	0	
Reg.		SBI <sub>x</sub> DBR								1~(N-2)

INTSB <sub>Ix</sub>	(								(N-1)	)
		7	6	5	4	3	2	1	0	
SBI <sub>x</sub> CR1		X	X	X	0	0	X	X	X	
Reg.		SBI <sub>x</sub> DBR								(N-1)

INTSB <sub>Ix</sub>	(								N	)
		7	6	5	4	3	2	1	0	
SBI <sub>x</sub> CR1		0	0	1	0	0	X	X	X	1
Reg.		SBI <sub>x</sub> DBR								N

INTSB<sub>Ix</sub> ( )

) X; Don't care

## (2) (<MST> = "0")

スレーブモードのとき、マスタが送ったスレーブアドレス、またはゼネラルコールアドレスを受信したとき、もしくは、受信したスレーブアドレスが一致した後、またはゼネラルコールアドレスを受信した後のデータ転送終了時に INTSB<sub>Ix</sub> 割り込み要求が発生します。また、マスタモードのとき、アービトレーションロストを検出するとスレーブモードとして動作し、アービトレーションロストを検出したワード転送の終了時に INTSB<sub>Ix</sub> 割り込み要求が発生します。INTSB<sub>Ix</sub> 割り込み要求が発生すると <PIN> が "0" にされ、SB<sub>x</sub>SCL 端子を "Low" レベルに引きます。SBI<sub>x</sub>DBR にデータを書き込む、SBI<sub>x</sub>DBR からデータを読み出す、または <PIN> に "1" を設定すると SB<sub>x</sub>SCL 端子が t<sub>LOW</sub> 後に開放されます。

なお、ゼネラルコールのセカンドバイトは内容によりアクノリッジ信号の発生有無を制御する必要がありますので、ソフトウェアによりアクノリッジ信号の制御を行なってください。

スレーブモード時は、通常のスレーブモードとしての処理、またはアービトレーションロストを検出し、スレーブモードになったときの処理を行います。

SBIxSR<AL>, <TRX>, <AAS>, <AD0>をテストし、場合分けを行います。「表 14-1 スレーブモード時の処理」にスレーブモード時の状態と必要な処理を示します。

例: スレーブレシーバモード時スレーブアドレスが一致し、方向ビットが"1"の場合

INTSBIx

```

if TRX = 0
Then
if AL = 0
Then
if AAS = 0
Then
SBIxCR1      X  X  X  1  0  X  X  X
SBIxDBR      X  X  X  X  X  X  X  X
    
```

) X; Don't care

14-1

<TRX>	<AL>	<AAS>	<AD0>		
1	1	1	0	"1"	1 <BC[2:0]> SBIxDBR
	0	1	0	"1"	
0		0	0	1	<LRB> "1" <PIN> "1" <TRX> "0" <PIN> <LRB> "0" <LRB> "0" <PIN> <BC[2:0]> 1 SBIxDBR
0	1	1	1/0	"0"	
		0	0		<PIN> "1" SBIxDBR ( ) <PIN> "1"
	0	1	1/0	"0"	
		0	1/0	1	1 <BC[2:0]> SBIxDBR

14.3.3.4

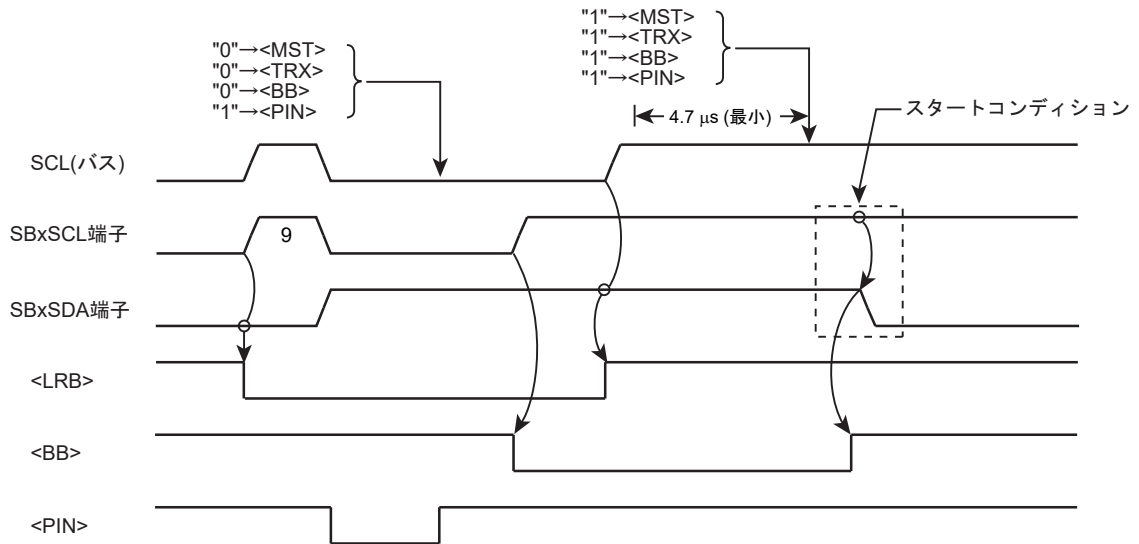
SBIxSR<BB> = "1"のときに、SBIxCR2<MST>、<TRX>、<PIN>に"1"、<BB>に"0"を書き込むとバス上にストップコンディション出力のシーケンスを開始します。バス上にストップコンディションが発生するまで、<MST>、<TRX>、<BB>、<PIN>の内容を書き替えないでください。

なお、バスのSCLラインが他のデバイスにより引かれていた場合、SCLラインが開放されてから、SBxSDA端子が立ち上がり、ストップコンディションが発生します。



SBixDBR	X	X	X	X	X	X	X	X
SBixCR2	1	1	1	1	1	0	0	0

) X; Don't care

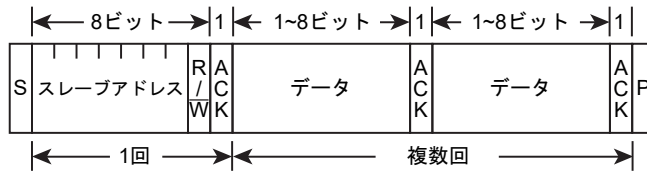


14-11

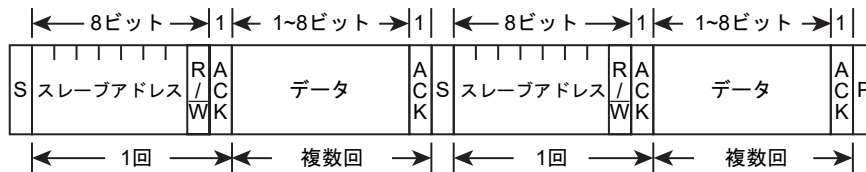
14.3.4

I2C バスモード時のデータフォーマットを図 14-12 に示します。

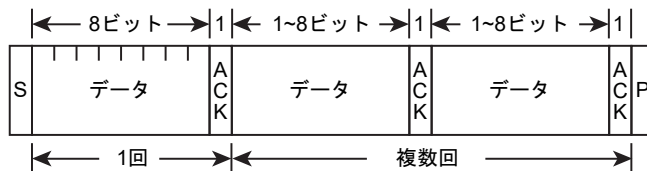
(a) アドレッシングフォーマット



(b) アドレッシングフォーマット (再スタートあり)



(c) フリーデータフォーマット (マスタデバイスからスレーブデバイスへデータを転送する転送フォーマット)



注) S: スタートコンディション  
 R/W: 方向ビット  
 ACK: アクノリッジビット  
 P: ストップコンディション

14-12 I2C

14.3.5

マルチマスタで使用している際に通信がロックした場合に備え、ソフトウェアによるリカバリ対策を用意願います。

リカバリ処理例

1. 送信開始と共にタイムアウト検知用のタイマカウントを開始
  2. 一定時間内にシリアルインタフェース割り込み (INTSBIx) が発生せず、タイムアウトとなった場合、通信がロックしたと判断
  3. シリアルバスインタフェースのソフトウェアリセットを実行し通信ロックを解除
  4. 送信タイミングの調整処理 (注)
  5. 送信データを再送信
- )

## 14.4 SIO

### 14.4.1 SIO

シリアルバスインタフェースをクロック同期式 8 ビット SIO モードで使用するときの制御、および動作状態のモニタは、以下のレジスタで行います。

#### 14.4.1.1 SBIXCR0( 0)

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

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	SBIEN	R/W	0: 1:  SBIXCR0
6-0	-	R	"0"

14.4.1.2 SBIXCR1( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SIOS	SIOINH	SIOM		-	SCK		
	0	0	0	0	1	0	0	0( 1)

Bit	Bit Symbol	Type																	
31-8	-	R	"0"																
7	SIOS	R/W	/																
6	SIOINH	R/W	0: 1:																
5-4	SIOM[1:0]	R/W	00: 01: Reserved 10: 11:																
3	-	R	"1"																
2-0	SCK[2:0]	R/W	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <table border="1"> <tr><td>000</td><td>n = 3</td></tr> <tr><td>001</td><td>n = 4</td></tr> <tr><td>010</td><td>n = 5</td></tr> <tr><td>011</td><td>n = 6</td></tr> <tr><td>100</td><td>n = 7</td></tr> <tr><td>101</td><td>n = 8</td></tr> <tr><td>110</td><td>n = 9</td></tr> <tr><td>111</td><td></td></tr> </table> </div> <div> <math display="block">\left. \begin{array}{l} \text{システムクロック: } f_{\text{sys}} \\ \text{クロックギア: } f_c/1 \\ \text{周波数} = \frac{f_{\text{sys}}/2}{2^n} \text{ [Hz]} \end{array} \right\}</math> </div> </div>	000	n = 3	001	n = 4	010	n = 5	011	n = 6	100	n = 7	101	n = 8	110	n = 9	111	
000	n = 3																		
001	n = 4																		
010	n = 5																		
011	n = 6																		
100	n = 7																		
101	n = 8																		
110	n = 9																		
111																			

- 1) <SCK[0]> "1" SBIXCR2 SIO "0"
- 2) SBIXCR2 SBIXSR <SIOS> = "0" <SIOINH> = "1"
- 3) <BC[2:0]>="001" <ACK>="0" SCL "L" SCL "2"



14.4.1.4 SBIXCR2( 2)

このレジスタをリードすると、SBIXSR レジスタとして機能します。

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	SBIM		-	-
	1( 1)	1( 1)	1( 1)	1( 1)	0	0	1( 1)	1( 1)

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7-4	-	R	"1" ( 1)
3-2	SBIM[1:0]	W	( 2) 00: 01: SIO 10: I2C 11: Reserved
1-0	-	R	"1" ( 1)

1) SIO

2)

## 14.4.1.5 SBIXSR( )

このレジスタをライトすると、SBIXCR2として機能します。

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	SIOF	SEF	-	-
	1( 1)	1( 1)	1( 1)	1( 1)	0	0	1( 1)	1( 1)

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7-4	-	R	"1" ( 1)
3	SIOF	R	0: 1:
2	SEF	R	0: 1:
1-0	-	R	"1" ( 1)

) SIO

14.4.1.6 SBIXBR0( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	I2SBI	-	-	-	-	-	-
	1	0	1	1	1	1	1	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	-	R	"1"
6	I2SBI	R/W	IDLE 0: 1:
5-1	-	R	"1"
0	-	R/W	"0"

## 14.4.2

## 14.4.2.1

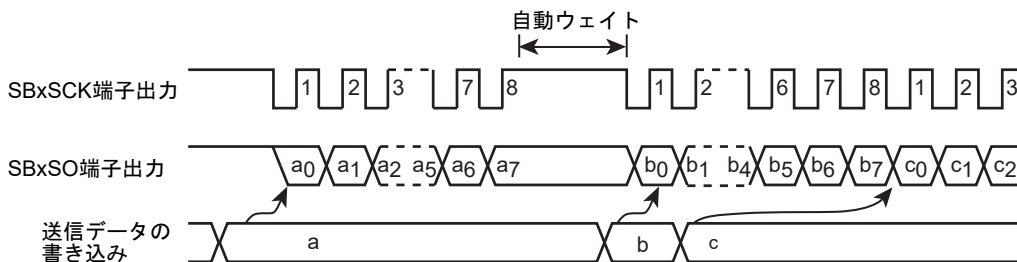
(1)

SBxCR1 <SCK[2:0]> により、次の選択ができます。

(a)

内部クロックモードでは7種類の周波数が選択できます。シリアルクロックはSBxSCK端子より外部に出力されます。なお、転送開始時SBxSCK端子出力は"High"レベルになります。

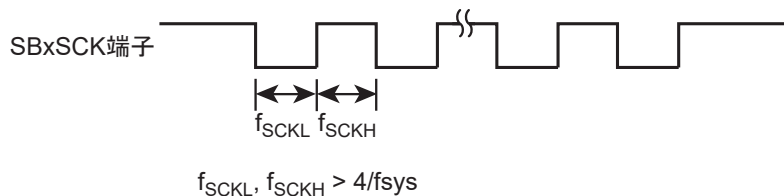
プログラムでデータの書き込み(送信時)またはデータの読み出し(受信時)がこのシリアルクロックレートに追従できないときには、自動的にシリアルクロックを停止し、それらの処理が終了するまで次のシフト動作を待機させる自動ウェイト機能を有しています。



14-13

(b) (&lt;SCK[2:0]&gt; = "111")

外部からSBxSCK端子に供給されるクロックをシリアルクロックとして用います。なお、シフト動作を確実にを行うためには、シリアルクロックの"High"レベル、"Low"レベル幅は下記に示すパルス幅が必要です。

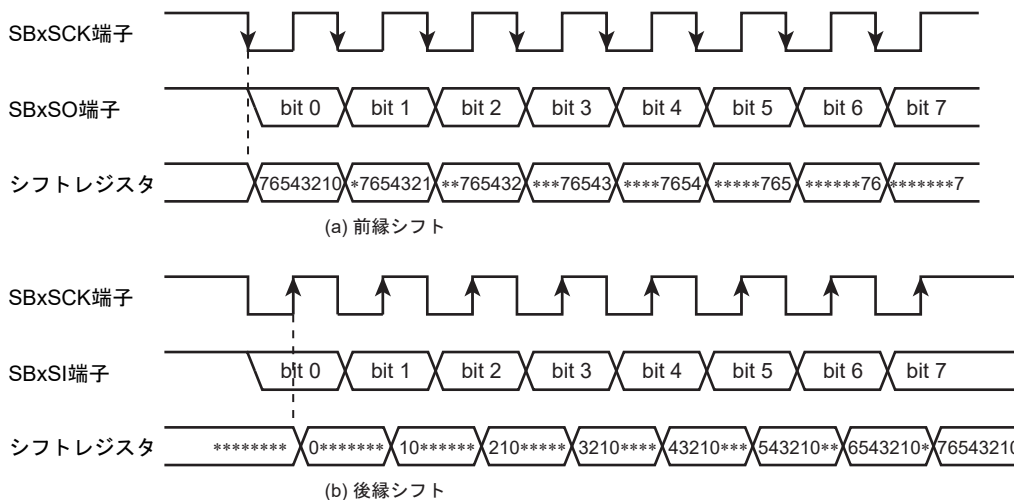


14-14

(2)

送信は前縁シフト、受信は後縁シフトになります。

- 前縁シフト  
シリアルクロックの前縁(SBxSCK 端子入出力の立ち下がりエッジ)でデータをシフトします。
- 後縁シフト  
シリアルクロックの後縁(SBxSCK 端子入出力の立ち上がりエッジ)でデータをシフトします。



## 14.4.2.2

SBIxCR1<SIOM[1:0]>で、送信/受信/送受信モードを選択します。

## (1) 8

コントロールレジスタに送信モードをセットした後、送信データを SBIxDBR に書き込みます。

送信データの書き込み後、SBIxCR1<SIOS>="1"を書き込むことにより送信が開始されます。送信データは、SBIxDBR からシフトレジスタに移され、シリアルクロックに同期して最下位ビット(LSB)側から SBxSO 端子に出力されます。送信データがシフトレジスタに移されると、SBIxDBR が空になりますので、次の送信データを要求する INTSBIx(バッファエンプティ)割り込み要求が発生します。

内部クロック動作の場合、8 ビットのデータをすべて送信したあと、次のデータがセットされていないとシリアルクロックを停止して自動ウェイト動作を行います。次の送信データを書き込むと自動ウェイト動作は解除されます。

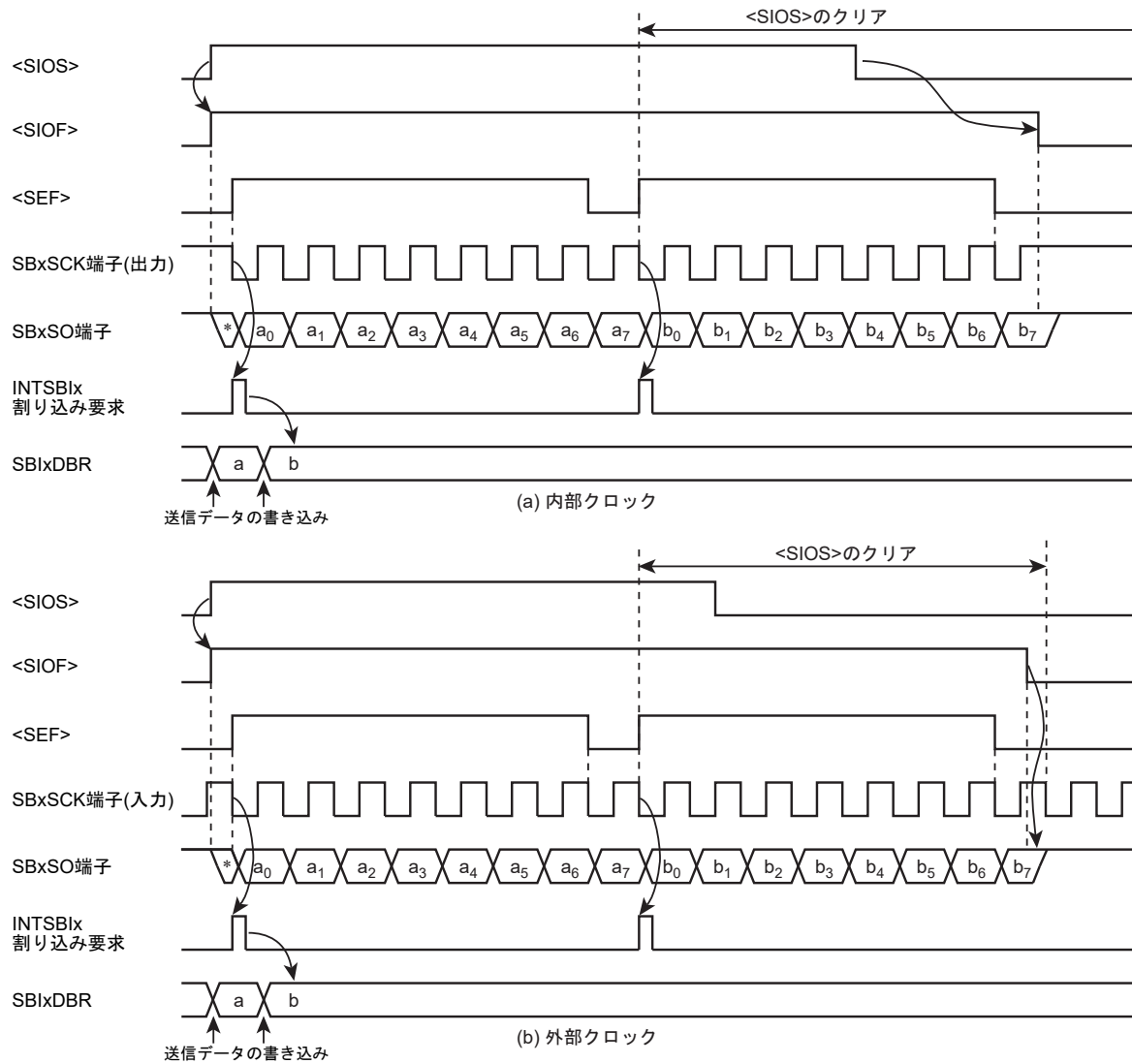
外部クロック動作の場合、次のデータのシフト動作に入る前に、SBIxDBR にデータが書き込まれている必要があります。したがって、転送速度は割り込み要求の発生から割り込みサービスプログラムにて、SBIxDBR にデータを書き込むまでの最大遅れ時間により決まります。

送信開始時、SBIxSR<SIOF>が"1"となってから SBxSCK 端子の立ち下がりエッジまでの間、前回送信したデータの最後のビットと同じ値が出力されます。

送信を終了させるには、INTSBIx 割り込みサービスプログラムで<SIOS>="0"を書き込むか<SIOINH>="1"を書き込みます。<SIOS>がクリアされると、データがすべて出力された時点で送信終了します。プログラムで送信の終了の確認は SBIxSR<SIOF>で行います。<SIOF>は送信の終了で"0"になります。<SIOINH>="1"を書き込んだ場合はただちに送信を打ち切り、<SIOF>は"0"になります。

外部クロック動作では、次の送信データのシフト動作に入る前に<SIOS>を"0"にクリアする必要があります。もしシフトアウトする前に<SIOS>が"0"にクリアされなかった場合は、ダミーのデータの送信後、停止します。

	7	6	5	4	3	2	1	0
SBIxCR1	0	1	0	0	0	X	X	X
SBIxDBR	X	X	X	X	X	X	X	X
SBIxCR1	1	0	0	0	0	X	X	X
INTSBIx								
SBIxDBR	X	X	X	X	X	X	X	X



14-16

例: <SIO> の送信終了指示(外部クロックの場合)のプログラム例

```

    7 6 5 4 3 2 1 0
    if SBxSR<SIOF> ≠ 0
    Then
    if SCK ≠ 1
    Then
    SBxSCK "1"
    SBxCR1 0 0 0 0 0 1 1 1 <SIO> = 0
  
```

## (2) 8

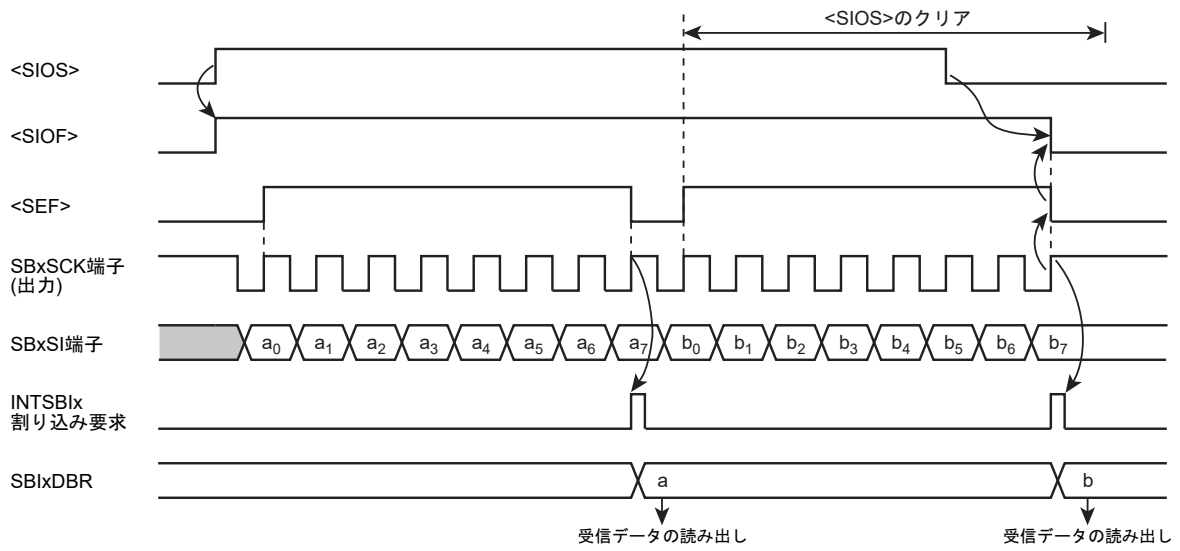
コントロールレジスタに受信モードをセットした後、SBIxCR1<SIOS>="1"を書き込むことにより受信可能となります。シリアルクロックに同期して、SBxSI 端子より最下位ビット側からシフトレジスタへデータを取り込みます。8ビットのデータが取り込まれるとシフトレジスタから SBIxDBR に受信データが書き込まれ、受信データの読み出しを要求する INTSBIx (バッファフル)割り込み要求が発生します。受信データは、割り込みサービスプログラムにて SBIxDBR から読み出します。

内部クロック動作の場合、受信データが SBIxDBR から読み出されるまでシリアルクロックを停止する自動ウェイト動作を行います。

外部クロック動作の場合は、シフト動作が外部から供給されるクロックに同期します。外部クロック動作での最大転送速度は、割り込み要求の発生から受信データの読み出しまでの最大遅れ時間により決まります。

受信を終了させるには INTSBIx 割り込みサービスプログラムで<SIOS>="0"を書き込むか、<SIOINH>="1"を書き込みます。<SIOS>がクリアされると、受信データが全ビット揃い、SBIxDBR への書き込みが完了した時点で受信が終了します。プログラムで受信の終了の確認は、SBIxSR<SIOF>で行います。<SIOF>は受信の終了で"0"にされます。受信終了の確認のあと最終受信データを読み出します。<SIOINH>="1"を書き込んだ場合は、ただちに受信を打ち切り、<SIOF>は"0"になります(受信データは無効になりますので読み出す必要はありません)。

	SBIxDBR (<SIOS> = "0" )							
	7	6	5	4	3	2	1	0
SBIxCR1	0	1	1	1	0	X	X	X
SBIxCR1	1	0	1	1	0	X	X	X
<b>INTSBIx</b>								
Reg.	SBIxDBR							



14-17 ( : )

## (3) 8

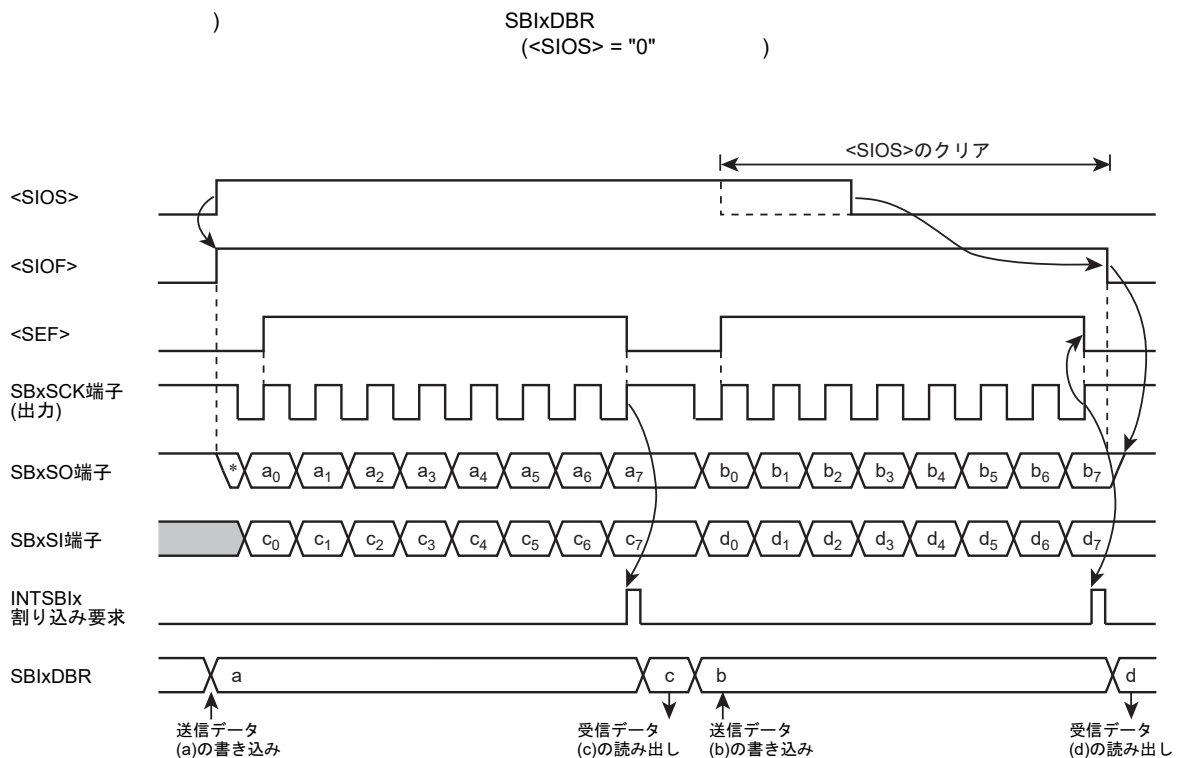
コントロールレジスタに送受信モードをセットした後、送信データを SBxIDBR に書き込みます。その後、SBxCR1<SIOS>に"1"をセットすることにより送受信可能となります。最下位ビットから、シリアルクロックの立ち下がりまで送信データが SBxSO 端子から出力され、立ち上がりで受信データが SBxSI 端子から取り込まれます。8 ビットのデータが取り込まれると、シフトレジスタから SBxIDBR へ受信データが転送され、INTSBx 割り込み要求が発生します。割り込みサービスプログラムにて受信データをデータバッファレジスタから読み出し、そのあと送信データを書き込みます。SBxIDBR は、送信/受信モードで兼用していますので、送信データは、必ず受信データを読み出してから書き込むようにしてください。

内部クロック動作の場合、受信データを読み出し、次の送信データを書き込むまで自動ウェイト動作を行います。

外部クロック動作の場合は、シフト動作が外部から供給されるシリアルクロックに同期するので、次のシフト動作に入る前に受信データを読み出し、次の送信データを書き込む必要があります。外部クロック動作での最大転送速度は、割り込み要求の発生から受信データを読み出し、送信データを書き込むまでの最大遅れ時間により決まります。

送信開始時、<SIOF>が"1"となってから SBxSCK 端子の立ち上がりエッジまでの間、前回送信したデータの最後のビットと同じ値が出力されます。

送受信を終了させるには、INTSBx 割り込みサービスプログラムで<SIOS>="0"を書き込むか SBxCR1 <SIOINH>="1"を書き込みます。<SIOS>が"0"にクリアされると、受信データが揃い、SBxIDBR への転送が完了した時点で送受信が終了します。プログラムで送受信の終了の確認は、SBxSR<SIOF>で行います。<SIOF>は送受信の終了で"0"にクリアされます。<SIOINH>を"1"にセットした場合は、ただちに送受信を打ち切り、<SIOF>は"0"にされます。



14-18

( : )

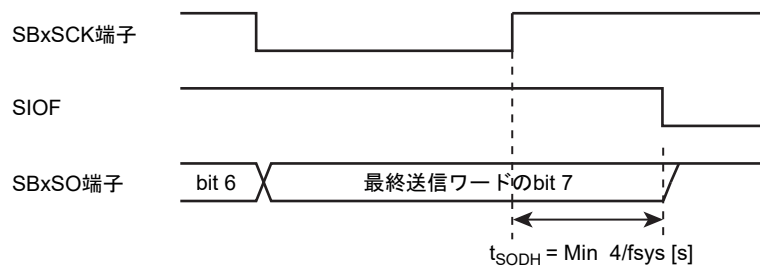
	7	6	5	4	3	2	1	0
SBlxCR1	0	1	1	0	0	X	X	X
SBlxDBR	X	X	X	X	X	X	X	X
SBlxCR1	1	0	1	0	0	X	X	X

INTSBlx

Reg.	SBlxDBR
SBlxDBR	X X X X X X X X

(4)

SBlxCR1<SIOS>="0"の状態では、送信データの最終ビットの SBxSCK 端子の立ち上がりに対するデータ保持時間は以下ようになります。送信モード、送受信モードとも同様です。



14-19



# 15 CAN

本製品は CAN コントローラを 1 ユニット 内蔵しています。

## 15.1

- ・ CAN バージョン 2.0 B アクティブ準拠
- ・ 標準および拡張フォーマット対応
- ・ 各フォーマットにおけるデータフレームおよびリモートフレーム対応
- ・ メールボックス 32 個 (送信/受信兼用 31 個と受信専用 1 個)
- ・ CAN バスボーレート : 最大 1 Mbps (最低 48 MHz のシステムクロックのとき)
- ・ Intel 82527 <sup>TM</sup>同等のビットタイミングパラメータ
- ・ ボーレートプリスケアラ内蔵
- ・ 送信メッセージ送信順番は次の 2 種類の内部アービトレーションからが選択可能
  - メールボックス番号の小さい順
  - メールボックス ID の優先度の高い順
- ・ メッセージ送信/受信時のタイムスタンプ機能
- ・ 動作モード

	CAN (CANMCR<WUBA =1) CANMCR
	CAN

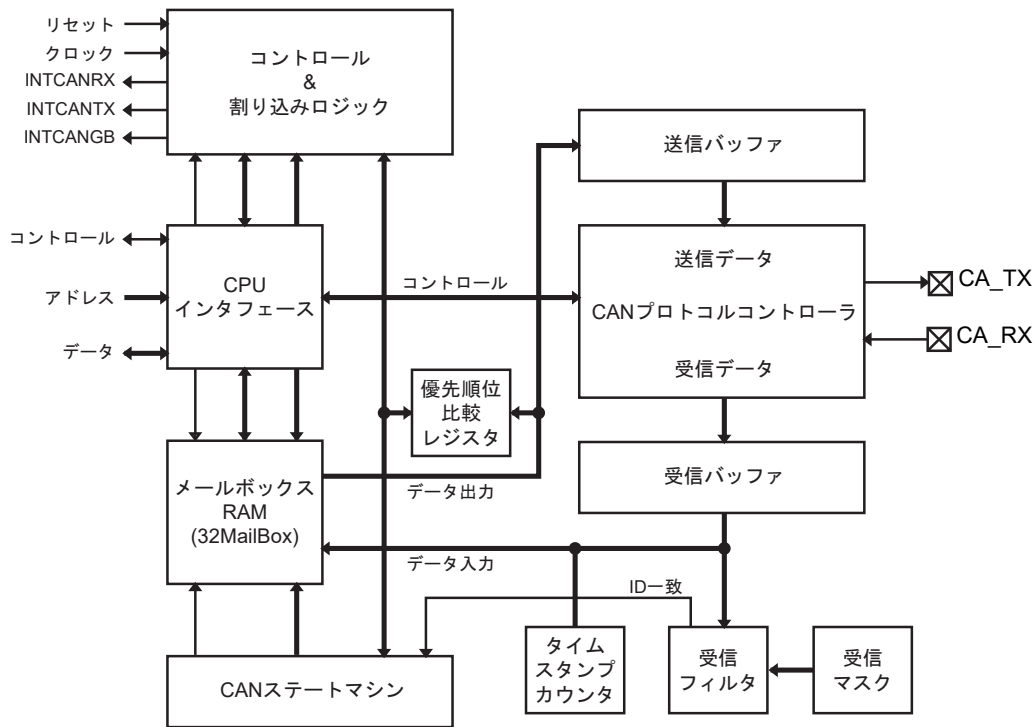
- ・ 2 系統のメッセージ受信マスク機能
  - プログラマブルグローバル受信マスク (メールボックス 0~30 共用)
  - プログラマブルローカル受信マスク (メールボックス 31 専用)
- ・ ID 拡張ビット用受信マスクビット
- ・ 割り込み信号

INTCANRX	: CAN
INTCANTX	: CAN
INTCANGB	: CAN

8

## 15.2

図 15-1 に CAN コントローラのブロック図を示します。



15-1 CAN

## 15.3 CAN

CAN バスへのインタフェースは、入力ピン CA\_RX および出力ピン CA\_TX です。これらのピンは CAN バス トランシーバ (ISO/DIS 11898 準拠) を介して接続してください。

高速トランシーバと低速トランシーバは区別されます。ピンのチップレベルでの電気的特性 (例えば、3.3 V または 5 V) が、トランシーバの要件を満たすよう注意してください。

## 15.4

本製品では、リセット後 CAN 用クロックは停止(CGCKSTP<CANSTP>="0")となっています。CAN 使用時にはクロックを動作(CGCKSTP<CANSTP>="1")としてください。

クロックの動作については、「クロック/モード制御」の章を参照してください。

## 15.4.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

(x=0~31)		Address(Base+)
ID	CANMBxID	0x0000
	CANMBxTSVMCF	0x0008
	CANMBxDL	0x0010
	CANMBxDH	0x0018

		Address(Base+)
	CANMC	0x0000
	CANMD	0x0008
	CANTRS	0x0010
	CANTRR	0x0018
	CANTA	0x0020
	CANAA	0x0028
	CANRMP	0x0030
	CANRML	0x0038
	CANLAM	0x0040
	CANGAM	0x0048
	CANMCR	0x0050
	CANGSR	0x0058
1	CANBCR1	0x0060
2	CANBCR2	0x0068
	CANGIF	0x0070
	CANGIM	0x0078
	CANMBTIF	0x0080
	CANMBRIF	0x0088
	CANMBIM	0x0090
	CANCDR	0x0098
	CANRFP	0x00A0
	CANCEC	0x00A8
	CANTSP	0x00B0
	CANTSC	0x00B8

### 15.4.2 ID (CANMBxID)

	31	30	29	28	27	26	25	24
bit symbol	IDE	GAME/LAME	RFH	ID				
	23	22	21	20	19	18	17	16
bit symbol	ID							
	15	14	13	12	11	10	9	8
bit symbol	ID							
	7	6	5	4	3	2	1	0
bit symbol	ID							

Bit	Bit Symbol	Type	
31	IDE	R/W	ID 0: (11 ID)<ID28> <ID18> 1: (29 ID)<ID28> <ID0> (<IDE>=1) (<IDE>=0)
30	GAME/LAME	R/W	(GAME) (LAME) 0: 1: <GAME> 31 0 ~ 30 LAM GAM <LAME> <GAME>=0 <LAME>=0 ID ID <GAME> "0"
29	RFH	R/W	( ) 0: 1: (<TRS> ) <RFH> ID <RFH>=1 <GAME> = 1 ID ID ID ID ID ID (<RMP> <RFP> )
28-0	ID[28:0]	R/W	ID (11 ID)<ID28> <ID18> (29 ID)<ID28> <ID0> ID ID (<ID28> ) "0"

メールボックスの ID は初期設定時に登録してください。メールボックスを許可した後に、メールボックスのメッセージ ID フィールドを変更する場合は、メールボックスに対応する CANMC レジスタの<MCx>ビットを"0"にクリアして、CAN コントローラに対してメールボックスを禁止してから新しい ID をライトしてください。

15.4.3

(CANMBxTSVMCF)

	31	30	29	28	27	26	25	24
bit symbol	TSV							
	23	22	21	20	19	18	17	16
bit symbol	TSV							
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	RTR	DLC			

Bit	Bit Symbol	Type																															
31-16	TSV[15:0]	R/W	16 15.5.6																														
15-5		R	:" "																														
			:"0"																														
4	RTR	R/W	0: 1:																														
3-0	DLC[3:0]	R/W	:																														
			<table border="1"> <thead> <tr> <th>&lt;DLC[3:0]&gt;</th> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>0</td> <td></td> </tr> <tr> <td>0001</td> <td>1</td> <td>D0</td> </tr> <tr> <td>0010</td> <td>2</td> <td>D0,D1</td> </tr> <tr> <td>0011</td> <td>3</td> <td>D0,D1,D2</td> </tr> <tr> <td>0100</td> <td>4</td> <td>D0,D1,D2,D3</td> </tr> <tr> <td>0101</td> <td>5</td> <td>D0,D1,D2,D3,D4</td> </tr> <tr> <td>0110</td> <td>6</td> <td>D0,D1,D2,D3,D4,D5</td> </tr> <tr> <td>0111</td> <td>7</td> <td>D0,D1,D2,D3,D4,D5,D6</td> </tr> <tr> <td>1000</td> <td>8</td> <td>D0,D1,D2,D3,D4,D5,D6,D7</td> </tr> </tbody> </table>	<DLC[3:0]>			0000	0		0001	1	D0	0010	2	D0,D1	0011	3	D0,D1,D2	0100	4	D0,D1,D2,D3	0101	5	D0,D1,D2,D3,D4	0110	6	D0,D1,D2,D3,D4,D5	0111	7	D0,D1,D2,D3,D4,D5,D6	1000	8	D0,D1,D2,D3,D4,D5,D6,D7
<DLC[3:0]>																																	
0000	0																																
0001	1	D0																															
0010	2	D0,D1																															
0011	3	D0,D1,D2																															
0100	4	D0,D1,D2,D3																															
0101	5	D0,D1,D2,D3,D4																															
0110	6	D0,D1,D2,D3,D4,D5																															
0111	7	D0,D1,D2,D3,D4,D5,D6																															
1000	8	D0,D1,D2,D3,D4,D5,D6,D7																															
			<DLC3:0>=1001 8																														

タイムスタンプ値は初期設定の必要はありません。

メッセージ制御フィールドは、受信メールボックスの場合、初期設定の必要はありません。受信メッセージをメールボックスに格納するときに、<RTR>も、<DLC[3:0]>も一緒にメッセージ制御フィールドに格納されます。送信メールボックスの場合は、初期設定が必要です。

メールボックスを許可した後に、<RFH>=1の送信メールボックスのメッセージ制御フィールドを変更する場合は、CANMC<MCx>ビットを"0"にクリアして、CANコントローラに対してメールボックスを禁止してから新しい<RTR>と<DLC[3:0]>をライトしてください。<RFH>=0の送信メールボッ

クスのメッセージ制御フィールドは、CANMC<MCx>ビットの設定にかかわらず変更することができますが、TRS<TRSx>ビットが "0" であることを確認してから新しい<RTR>と<DLC[3:0]>をライトする必要があります。

### 15.4.4 (CANMBxDL/CANMBxDH)

送信の場合は、メールボックスの<DLC[3:0]>で設定したデータバイト数だけ送信します。

受信の場合は、受信メッセージ中のデータ長コードがメールボックスの<DLC[3:0]>にコピーされ、コピーされた<DLC[3:0]>によるデータバイト数だけが有効になります。

メールボックスはリード/ライト可能ですが、受信メールボックスの場合、データフィールドへのライトはしないでください。ライトした場合、受信データに不整合が生じるおそれがあります。

<RFH>=1 の送信メールボックスのデータフィールドを更新する場合は、CANCDR<CDRx>に "1" を設定し、送信要求を一時保留してから新しいデータをライトしてください。

<RFH>=0 の送信メールボックスのデータフィールドを更新する場合は、CANTRS<TRSx>ビットが "0" であることを確認してから新しいデータをライトしてください。

CANMBxDL

	31	30	29	28	27	26	25	24
bit symbol	D3							
	23	22	21	20	19	18	17	16
bit symbol	D2							
	15	14	13	12	11	10	9	8
bit symbol	D1							
	7	6	5	4	3	2	1	0
bit symbol	D0							

Bit	Bit Symbol	Type	
31-24	D3[7:0]	R/W	/
23-16	D2[7:0]	R/W	/
15-8	D1[7:0]	R/W	/
7-0	D0[7:0]	R/W	/

CANMBxDH

	31	30	29	28	27	26	25	24
bit symbol	D7							
	23	22	21	20	19	18	17	16
bit symbol	D6							
	15	14	13	12	11	10	9	8
bit symbol	D5							
	7	6	5	4	3	2	1	0
bit symbol	D4							

Bit	Bit Symbol	Type	
31-24	D7[7:0]	R/W	/
23-16	D6[7:0]	R/W	/
15-8	D5[7:0]	R/W	/
7-0	D4[7:0]	R/W	/

15.4.5 CANMC( )

	31	30	29	28	27	26	25	24
bit symbol	MC31	MC30	MC29	MC28	MC27	MC26	MC25	MC24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MC23	MC22	MC21	MC20	MC19	MC18	MC17	MC16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MC15	MC14	MC13	MC12	MC11	MC10	MC9	MC8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type																
31-0	MC31- MC0	R/W	<p>( No.31 ~ No.0 )</p> <p>0: MBx CAN</p> <p>1: MBx CAN</p> <p>CPU</p> <table border="1"> <tr> <td></td> <td>ID</td> <td>&lt;RFH&gt;=1</td> <td></td> <td></td> </tr> <tr> <td>&lt;MCx&gt;=0</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>&lt;MCx&gt;=1</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>		ID	<RFH>=1			<MCx>=0					<MCx>=1				
	ID	<RFH>=1																
<MCx>=0																		
<MCx>=1																		

) CANMC

:

:

(CANTRS<TRSx>=1)

(CANTRS<TRSx>=0)

<MCx>=0

### 15.4.6 CANMD( )

	31	30	29	28	27	26	25	24
bit symbol	MD31	MD30	MD29	MD28	MD27	MD26	MD25	MD24
	1	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MD23	MD22	MD21	MD20	MD19	MD18	MD17	MD16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MD15	MD14	MD13	MD12	MD11	MD10	MD9	MD8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	MD31	R	( No.31 ) 31 1
30-0	MD30- MD0	R/W	( No.30 ~ No.0 ) 0: 1:

CANMD レジスタの設定は、初期設定時に行ってください。メールボックスの方向は、動作中に変更できません。CANMD レジスタの設定を変更する場合は、対応する CANMC<MCx>ビットを"0"にしてから実行してください。

15.4.7 CANTRS( )

	31	30	29	28	27	26	25	24
bit symbol	-	TRS30	TRS29	TRS28	TRS27	TRS26	TRS25	TRS24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TRS23	TRS22	TRS21	TRS20	TRS19	TRS18	TRS17	TRS16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRS15	TRS14	TRS13	TRS12	TRS11	TRS10	TRS9	TRS8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRS7	TRS6	TRS5	TRS4	TRS3	TRS2	TRS1	TRS0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31		R	: "0" : "0"
30-0	TRS30-TRS0	R/W	( No.30 ~ No.0 ) <TRSx> x CANMCR<MTOS> CPU "0" CPU "1"

) No.31

送信に設定されているメールボックスの CANTRS<TRSx>ビットにのみ CPU からの "1" ライトでセットできます。受信に設定されているメールボックスの CANTRS<TRSx>ビットはセットできません。

CANTRS<TRSx>ビットは、送信が成功したとき、または、CANTRR<TRRx>ビットを "1" にセットして送信要求をリセットしたとき、"0" にクリアされます。

送信を失敗したときは、送信が成功するまで、または、CANTRR<TRRx>ビットを "1" にセットして送信要求をリセットするまで、送信を繰り返します。

CANTRS<TRSx>ビットが "1" のときは、メールボックス x にライトしないでください。

### 15.4.8 CANTRR( )

	31	30	29	28	27	26	25	24
bit symbol	-	TRR30	TRR29	TRR28	TRR27	TRR26	TRR25	TRR24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TRR23	TRR22	TRR21	TRR20	TRR19	TRR18	TRR17	TRR16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRR15	TRR14	TRR13	TRR12	TRR11	TRR10	TRR9	TRR8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRR7	TRR6	TRR5	TRR4	TRR3	TRR2	TRR1	TRR0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31		R	: "0" : "0"
30-0	TRR30-TRR0	R/W	( No.30 ~ No.0 ) <TRRx> x x CPU "0" CPU "1"

) No.31

送信に設定されているメールボックスの CANTRR<TRRx>にのみ CPU からの "1" ライトでセットできます。受信に設定されているメールボックスの CANTRR<TRRx>はセットできません。

CANTRR<TRRx>は、送信が成功したとき、または、送信アボートしたとき内部ロジックで "0" にクリアされます。CPU からの "0" ライトは無効です。

CANTRR<TRRx>= "1" のときは、メールボックス x にライトしないでください。

CANTRR<TRRx>をセットすると、CANTRS<TRRx>のセットによるメールボックス x のメッセージ送信がキャンセルされますが、以下の 3 通りの場合にわかれます。

- a. 現在まだ送信されていない場合  
メッセージの送信要求は、直ちにクリアされます。  
(CANTRS<TRRx> = 0, CANTRR<TRRx> = 0, CANAA<AAx> = 1)
- b. 現在送信中で、アービトレーションロストまたは CAN バス上のエラー検出した場合  
メッセージの送信要求はクリアされ、送信を中止します。  
(CANTRS<TRRx> = 0, CANTRR<TRRx> = 0, CANAA<AAx> = 1)
- c. 現在送信中で、アービトレーションロストや CAN バス上のエラー検出がない場合  
メッセージの送信要求はクリアされずに、送信を完了します。  
(CANTRS<TRRx> = 0, CANTRR<TRRx> = 0, CANTA<TAx> = 1)

15.4.9 CANTA( )

	31	30	29	28	27	26	25	24
bit symbol	-	TA30	TA29	TA28	TA27	TA26	TA25	TA24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TA23	TA22	TA21	TA20	TA19	TA18	TA17	TA16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TA15	TA14	TA13	TA12	TA11	TA10	TA9	TA8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31		R	: "0" : "0"
30-0	TA30-TA0	R/W	( No.30 ~ No.0 ) x <TAx> "1" <TAx> CPU <TAx> CANTRS<TRSx> "1"

) No.31

メールボックス x のメッセージ送信が正常終了した場合、CANTA<TAx>ビットが "1" にセットされます。また、メールボックス割り込みマスクレジスタ CANMBIM の対応する<MBIMx>ビットを "1" にセットしてメールボックス割り込みを許可している場合、メールボックス送信割り込みフラグレジスタ CANMBTIF<MBTIFx>ビットが "1" にセットされ、CAN 送信完了割り込み INTCANTX が発生します。

CPU から<TAx>ビットまたは CANTRS<TRSx>ビットに "1" をライトすると<TAx>ビットがクリアできません。CPU から<TAx>ビットまたは CANTRS<TRSx>ビットへの "0" ライトは無効です。

15.4.10 CANAA( )

	31	30	29	28	27	26	25	24
bit symbol	-	AA30	AA29	AA28	AA27	AA26	AA25	AA24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	AA23	AA22	AA21	AA20	AA19	AA18	AA17	AA16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AA15	AA14	AA13	AA12	AA11	AA10	AA9	AA8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AA7	AA6	AA5	AA4	AA3	AA2	AA1	AA0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31		R	: "0" : "0"
30-0	AA30-AA0	R/W	( No.30 ~ No.0 ) x <AAx> "1" <AAx> CPU <AAx> CANTRS<TRSx> "1"

) No.31

メールボックス x のメッセージ送信が異常終了した場合、CANAA<AAx>が "1" にセットされます。また、グローバル割り込みフラグレジスタ CANGIF<TRMABF>も "1" にセットされ、グローバル割り込みマスクレジスタ CANGIM<TRMABM>を "1" にセットして送信アボート割り込みを許可している場合、CAN グローバル割り込み INTCANGB が発生します。

CPU から<AAx>または CANTRS<TRSx>に "1" をライトすると<AAx>がクリアできます。CPU から<AAx>または CANTRS<TRSx>への "0" ライトは無効です。

15.4.11 CANRMP( )

	31	30	29	28	27	26	25	24
bit symbol	RMP31	RMP30	RMP29	RMP28	RMP27	RMP26	RMP25	RMP24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RMP23	RMP22	RMP21	RMP20	RMP19	RMP18	RMP17	RMP16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RMP15	RMP14	RMP13	RMP12	RMP11	RMP10	RMP9	RMP8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RMP7	RMP6	RMP5	RMP4	RMP3	RMP2	RMP1	RMP0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	RMP31- RMP0	R/W	( No.31 ~ No.0 ) x <RMPx> "1" <RMPx> "1" <RMPx>

)

メールボックス x のメッセージ受信が正常終了した場合、CANRMP<RMPx>が "1" にセットされま  
す。また、メールボックス割り込みマスクレジスタ CANMBIM の対応する<MBIMx>ビットを "1" に  
セットしてメールボックス割り込みを許可している場合、メールボックス受信割り込みフラグレジス  
タ CANMBRIF の<MBRIFx>が "1" にセットされ、CAN 受信完了割り込み INTCANRX が発生します。

<RMPx>は、CPU から<RMPx>に "1" をライトするとクリアできます。CPU から<RMPx>への "0"  
ライトは無効です。

15.4.12 CANRML( )

	31	30	29	28	27	26	25	24
bit symbol	RML31	RML30	RML29	RML28	RML27	RML26	RML25	RML24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RML23	RML22	RML21	RML20	RML19	RML18	RML17	RML16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RML15	RML14	RML13	RML12	RML11	RML10	RML9	RML8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RML7	RML6	RML5	RML4	RML3	RML2	RML1	RML0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	RML31- RML0	R/W	( No.31 ~ No.0 ) <RMPx> "1" <sup>x</sup> <RMLx> "1" x <RMPx> "1" <RMLx>

CANRML<RMLx>は、内部ロジックがセットし、CPU から CANRMP<RMPx>ビットへの "1" ライトでクリアできます。<RMPx>ビットも同時にクリアされます。CPU から<RMLx>ビットへの "1" または "0" ライトは無効です。

CANRMP<RMPx>ビットが "1" にセットされた状態で、メールボックス x が次のメッセージを受信すると、受信メッセージロストアдресタ CANRML の対応する<RMLx>ビットが "1" にセットされます。この場合、メールボックス x は、新しく受信したデータでオーバライトされます。

また、グローバル割り込みフラグレジスタ CANGIF の<TRMABF>ビットも "1" にセットされ、グローバル割り込みマスクレジスタ CANGIM の<TRMABM>ビットを "1" にセットして送信アポート割り込みを許可している場合、CAN グローバル割り込み INTCANGB が発生します。

また、グローバル割り込みマスクレジスタ CANGIM の<RMLIM>ビットを "1" にセットして受信メッセージロストアдресタ割り込みを許可している場合は、CAN グローバル割り込み INTCANGB が発生します。

メッセージ受信前/後の CANRMP、CANRML レジスタ変化を、表 15-1 に示します。

15-1 / CANRMP CANRML

ID					
	<RMPx>	<RMLx>	<RMPx>	<RMLx>	
	Don't care	Don't care	Don't care	Don't care	
	0	0	1	0	ID x
	1	0	1	1	ID x
	1	1	1	1	

15.4.13 CANLAM( )

ローカル受信マスクレジスタ CANLAM は、メールボックス 31 への受信メッセージ ID のフィルタリングにのみ使用されます。この機能によって、メールボックス 31 への受信メッセージの任意の ID ビットをローカルにマスクできます。

	31	30	29	28	27	26	25	24
bit symbol	LAMI	-	-	LAM				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	LAM							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	LAM							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	LAM							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	LAMI	R/W	ID <IDE> ( No.31) 0: 1:  <LAMI> = 0                    31 ID <IDE>  <LAMI> = 1                    31 ID <IDE>
30-29		R	: "0" : "0"
28-0	LAM[28:0]	R/W	ID 0:                    ID 1:                    ID                    ID

拡張フォーマットの場合、<ID[28:0]>と<LAM[28:0]>がフィルタリングに使用されます。

標準フォーマットの場合、<ID[28:18]>と<LAM[28:18]>がフィルタリングに使用されます。

標準フォーマットのメッセージを受信した場合、ID の拡張部分(<ID[17:0]>)は不定値になります。従って、標準および拡張フォーマットを交互に同じメールボックスで受信することは推奨できません。

CANLAM の設定は、初期設定時（コンフィグレーションモード時）に行い、動作中には設定を変更しないでください。受信中に設定を変更した場合、設定変更途中の CANLAM 値で、受信メッセージ ID のフィルタリングを実行します。

### 15.4.14 CANGAM( )

グローバル受信マスクレジスタ CANGAM は、メールボックス 0~30 への受信メッセージ ID のフィルタリングに使用されます。この機能によって、メールボックス 0~30 への受信メッセージの任意の ID ビットをグローバルにマスクできます。

	31	30	29	28	27	26	25	24
bit symbol	GAMI	-	-	GAM				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	GAM							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	GAM							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	GAM							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	GAMI	R/W	ID <IDE> ( No.0 ~ No.31) 0: 1: <GAMI> = 0                    0 ~ 30 ID                    <IDE> <GAMI> = 1                    0 ~ 30 ID                    <IDE>
30-29		R	: "0" : "0"
28-0	GAM[28:0]	R/W	ID 0:                    ID                    ID 1:                    ID

拡張フォーマットの場合、<ID[28:0]>と<GAM[28:0]>がフィルタリングに使用されます。

標準フォーマットの場合、<ID[28:18]>と<GAM[28:18]>がフィルタリングに使用されます。

標準フォーマットのメッセージを受信した場合、ID の拡張部分(<ID17:0>)は不定値になります。従って、標準および拡張フォーマットを交互に同じメールボックスで受信することは推奨できません。

CANGAM の設定は、初期設定時（コンフィグレーションモード時）に行い、動作中には設定を変更しないでください。受信中に設定を変更した場合、設定変更途中の CANGAM 値で、受信メッセージ ID のフィルタリングを実行します。

15.4.15 CANMCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	SUR	-	TSTLB	TSTERR
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CCR	SMR	-	WUBA	MTOS	-	TSCC	SRES
	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-12		R	: "0" : "0"
11	SUR	R/W	0: ( ) 1:
10		R	: "0" : "0"
9	TSTLB	R/W	0: ( ) 1: ( )
8	TSTERR	R/W	0: ( ) 1: ( CANCEC )
7	CCR	R/W	0: ( ) 1: ( CANBCR1/CANBCR2 )
6	SMR	R/W	0: ( 3 ) 1: ( CAN )
5		R	: "0" : "0"
4	WUBA	R/W	0: CANMCR 1: CANMCR
3	MTOS	R/W	0: 1: ID
2		R	: "0" : "0"
1	TSCC	R/W	0: 1: "0" ( 1 ) "0"

Bit	Bit Symbol	Type	
0	SRES	R/W	2 0: 1:CAN  "0"

- 1) CANTSP CANTSC "0"
- 2) CAN 16 CPU Clock  
CAN 88 CPU Clock
- 3) CANGSR<SMA>="1" CANMCR<SMR>="0"

15.4.16 CANGSR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	MIS
	0	0	0	0	0	0	0	1
	15	14	13	12	11	10	9	8
bit symbol	MIS				RM	TM	-	SUA
	1	1	1	1	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CCE	SMA	-	-	TSO	BO	EP	EW
	1	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-17		R	: "0" : "0"
16-12	MIS[4:0]	R	00000 : 0 01011 : 11 10110 : 22 00001 : 1 01100 : 12 10111 : 23 00010 : 2 01101 : 13 11000 : 24 00011 : 3 01110 : 14 11001 : 25 00100 : 4 01111 : 15 11010 : 26 00101 : 5 10000 : 16 11011 : 27 00110 : 6 10001 : 17 11100 : 28 00111 : 7 10010 : 18 11101 : 29 01000 : 8 10011 : 19 11110 : 30 01001 : 9 10100 : 20 11111 : 01010 : 10 10101 : 21
11	RM	R	0:CAN 1:CAN
10	TM	R	0:CAN 1:CAN
9		R	: "0" : "0"
8	SUA	R	0:CAN 1:CAN
7	CCE	R	0:CAN 1:CAN  CANBCR1 CANBCR2
6	SMA	R	0:CAN 1:CAN  CAN
5-4		R	: "0" : "0"

Bit	Bit Symbol	Type	
3	TSO	R	0: 1: "0" CANGIF <TSOIF> "0" 1
2	BO	R	0: ( ) 1: CAN <TEC> 256 CAN
1	EP	R	0:CAN 1:CAN
0	EW	R	0:<TEC> <REC> 96 1:<TEC> <REC> 1 96

15.4.17 CANBCR1( 1)

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

Bit	Bit Symbol	Type	
31-10		R	: "0" : "0"
9-0	BRP[9:0]	R/W	0 ~ 1023

15.4.18 CANBCR2( 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	SJW	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	SAM	TSEG2			TSEG1			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-10		R	: "0" : "0"
9-8	SJW[1:0]	R/W	00 : 1 × TQ 01 : 2 × TQ 10 : 3 × TQ 11 : 4 × TQ
7	SAM	R/W	0: 1:
6-4	TSEG2[2:0]	R/W	000 :            100 : 5 × TQ 001 : 2 × TQ    101 : 6 × TQ 010 : 3 × TQ    110 : 7 × TQ 011 : 4 × TQ    111 : 8 × TQ
3-0	TSEG1[3:0]	R/W	(SYNCSEG ) 0000 :            1000 : 9 × TQ 0001 : 2 × TQ    1001 : 10 × TQ 0010 : 3 × TQ    1010 : 11 × TQ 0011 : 4 × TQ    1011 : 12 × TQ 0100 : 5 × TQ    1100 : 13 × TQ 0101 : 6 × TQ    1101 : 14 × TQ 0110 : 7 × TQ    1110 : 15 × TQ 0111 : 8 × TQ    1111 : 16 × TQ

15.4.19 CANGIF( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFPF	WUIF	RMLIF	TRMABF	TSOIF	BOIF	EPIF	WLIF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	: "0" : "0"
7	RFPF	R/W	0: 1: ( ) <RFH> "1"
6	WUIF	R/W	0: 1:
5	RMLIF	R/W	0: 1: 1
4	TRMABF	R/W	0: 1: (CANAA 1 )
3	TSOIF	R/W	0: 1: 1
2	BOIF	R/W	0: CAN 1: CAN
1	EPIF	R/W	0: CAN 1: CAN
0	WLIF	R/W	0: 1: 1

グローバル割り込みフラグレジスタ CANGIF の各割り込みフラグは、対応するグローバル割り込み条件が成立すると "1" にセットされます。あるグローバル割り込みフラグが "1" にセットされたとき、グローバル割り込みマスクレジスタ CANGIM の対応するビットが "1" (割り込み許可) のとき、CAN グローバル割り込み INTCANGB が "High" になります。

CANGIF レジスタは、CANGIF レジスタの対応するビットに "1" をライトするとクリアできません。"0" のライトは無効です。

15.4.20 CANGIM( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFPM	WUIM	RMLIM	TRMABF	TSOIM	BOIM	EPIM	WLIM
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	: "0" : "0"
7	RFPM	R/W	0: 1:
6	WUIM	R/W	0: 1:
5	RMLIM	R/W	0: 1:
4	TRMABF	R/W	0: 1:
3	TSOIM	R/W	0: 1:
2	BOIM	R/W	0: 1:
1	EPIM	R/W	0: 1:
0	WLIM	R/W	0: 1:

グローバル割り込みマスクレジスタ CANGIM は、CANGIF レジスタの各割り込み条件に対応して、グローバル割り込みを許可するか禁止するかを制御します。CANGIF レジスタのビットが "0" のとき、対応する割り込み発生条件によるグローバル割り込みを禁止し、"1" のとき許可します。

リセットで CANGIM レジスタは全ビット "0" にクリアされ、グローバル割り込みを禁止します。

15.4.21 CANMBTIF( )

	31	30	29	28	27	26	25	24
bit symbol	-	MBTIF30	MBTIF29	MBTIF28	MBTIF27	MBTIF26	MBTIF25	MBTIF24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MBTIF23	MBTIF22	MBTIF21	MBTIF20	MBTIF19	MBTIF18	MBTIF17	MBTIF16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MBTIF15	MBTIF14	MBTIF13	MBTIF12	MBTIF11	MBTIF10	MBTIF9	MBTIF8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MBTIF7	MBTIF6	MBTIF5	MBTIF4	MBTIF3	MBTIF2	MBTIF1	MBTIF0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31		R	: "0" : "0"
30-0	MBTIF30- MBTIF0	R/W	( No.30 ~ No.0 ) CANMBIM <MBIMx>= "1" <MBTIFx> "1" INTCANTX "High" CANMBIM<MBIMx> "0" <MBTIFx> INTCANTX "Low" CANTA CANMBTIF 1 "1" INTCANTX "High" <MBTIFx> CPU <MBTIFx> "1" "0"

メールボックスが受信に設定されている場合、CANMBTIF レジスタの対応ビットは "0" がリードされます。メールボックスが送信に設定されている場合、CANMBRIF レジスタの対応ビットは "0" がリードされます。

15.4.22 CANMBRIF( )

	31	30	29	28	27	26	25	24
bit symbol	MBRIF31	MBRIF30	MBRIF29	MBRIF28	MBRIF27	MBRIF26	MBRIF25	MBRIF24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MBRIF23	MBRIF22	MBRIF21	MBRIF20	MBRIF19	MBRIF18	MBRIF17	MBRIF16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MBRIF15	MBRIF14	MBRIF13	MBRIF12	MBRIF11	MBRIF10	MBRIF9	MBRIF8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MBRIF7	MBRIF6	MBRIF5	MBRIF4	MBRIF3	MBRIF2	MBRIF1	MBRIF0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	MBRIF31- MBRIF0	R/W	<p>( No.31 ~ No.0 )</p> <p>x CANMBIM &lt;MBIMx&gt;=</p> <p>"1" &lt;MBRIFx&gt; "1" INTCANRX "High"</p> <p>CANMBIM&lt;MBIMx&gt; "0" &lt;MBRIFx&gt; INTCANRX "Low"</p> <p>CANRMP</p> <p>CANMBRIF 1 "1" INTCANRX "High"</p> <p>&lt;MBRIFx&gt; CPU &lt;MBRIFx&gt; "1"</p> <p>"0"</p>

15.4.23 CANMBIM( )

	31	30	29	28	27	26	25	24
bit symbol	MBIM31	MBIM30	MBIM29	MBIM28	MBIM27	MBIM26	MBIM25	MBIM24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	MBIM23	MBIM22	MBIM21	MBIM20	MBIM19	MBIM18	MBIM17	MBIM16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	MBIM15	MBIM14	MBIM13	MBIM12	MBIM11	MBIM10	MBIM9	MBIM8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	MBIM7	MBIM6	MBIM5	MBIM4	MBIM3	MBIM2	MBIM1	MBIM0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	MBIM31- MBIM0	R/W	0: 1:

割り込み発生をどのメールボックスに許可するか、または禁止するかは、CANMBIM 内の設定で決まります。CANMBIM 内のあるビットが "0" なら、対応するメールボックスの割り込み発生が禁止され、"1" なら、許可されます。CANMBIM のリセット値は 0 です。

15.4.24 CANCDR( )

	31	30	29	28	27	26	25	24
bit symbol	-	CDR30	CDR29	CDR28	CDR27	CDR26	CDR25	CDR24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	CDR23	CDR22	CDR21	CDR20	CDR19	CDR18	CDR17	CDR16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	CDR15	CDR14	CDR13	CDR12	CDR11	CDR10	CDR9	CDR8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CDR7	CDR6	CDR5	CDR4	CDR3	CDR2	CDR1	CDR0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31		R	: "0" : "0"
30-0	CDR30-CDR0	R/W	( No.30 ~ No.0 ) x <CDRx> "1" x CANTRS<TRSx> <CDRx> x x <CDRx> "0"

) No.31

CANCDR レジスタは、リモートフレームの自動応答を許可(CANMBxID<RFH>=1)している送信メールボックス x のデータフィールドを更新する場合に有効です。自動応答を許可しているメールボックス x は、リモートフレーム受信で自動的にメッセージの送信を開始するため、送信中にデータフィールドを更新してしまう可能性があります。(この場合、送信の途中から更新後のデータが出力されます。) <CDRx>を "1" に設定し、データ送信を一時保留することで、送信中のデータフィールド更新を避けることが可能です。

15.4.25 CANRFP( )

	31	30	29	28	27	26	25	24
bit symbol	RFP31	RFP30	RFP29	RFP28	RFP27	RFP26	RFP25	RFP24
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	RFP23	RFP22	RFP21	RFP20	RFP19	RFP18	RFP17	RFP16
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	RFP31- RFP0	R/W	( No.31 ~ No.0 ) x <RFPx> CANRMP<RMPx> "1"  <RFPx> CANRMP<RMPx> "1"

CANRFP<RFPx>は、内部ロジックがセットし、CPU から CANRMP<RMPx>への "1" ライトでクリアできます。<RMPx>も同時にクリアされます。CPU から<RMPx>への "0" ライトおよび<RFPx>への "1" または "0" ライトは無効です。

<RFPx>=1 のメールボックス x が、データフレーム受信によってオーバーライトされた場合でも、<RFPx>はクリアされます。

また、グローバル割り込みマスクレジスタ CANGIM の<RFPM>を "1" にセットしてリモートフレームペンディング割り込みを許可している場合は、CAN グローバル割り込み INTCANGB が発生します。

15.4.26 CANCEC( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TEC							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	REC							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	: "0" : "0"
15-8	TEC[7:0]	R	8 (
		R/W	8 (CANMCR<TSTERR>="1" )
7-0	REC[7:0]	R	8 (
		R/W	8 (CANMCR<TSTERR>="1" )

CAN コントローラには、受信エラーカウンタ REC と送信エラーカウンタ TEC の 2 つのエラーカウンタがあります。両カウンタの値は、CPU からリードできます。エラーカウンタへのアクセスは、テストエラーモード (CANMCR レジスタの<TSTERR>ビットが "1") のときだけライト可能です。CANCEC レジスタにライトする場合、下位 8 ビット(REC)へのライトデータが、上位 8 ビット(TEC)にもライトされます。

CAN エラーカウンタは、CAN Specification バージョン 2.0B に従ってカウントアップまたはカウントダウンされます。

<REC>は、エラーパッシブ制限 (128) を超えると、カウントアップしなくなります。<REC>=128 のとき、メッセージの正常受信後、<REC>は 119 ~ 127 の間の値にセットされます。バスオフ状態に達した後は、エラーカウンタは不定です。

バスオフに達すると、受信エラーカウンタは、バス上の連続 11 個の劣勢ビット後にカウントアップされます。カウント値が 128 になると、モジュールは自動的にエラーアクティブの状態になります。すべての内部フラグはリセットされ、エラーカウンタは "0" にクリアされます。コンフィグレーションレジスタはプログラミングされた値を保持します。エラーカウンタの値は、バスオフ状態では不定です。

CAN がコンフィグレーションモードになると、エラーカウンタはクリアされます。

15.4.27 CANTSP( )

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

Bit	Bit Symbol	Type	
31-4		R	: "0" : "0"
3-0	TSP[3:0]	R/W	4 TSC

メールボックスへの書き込みサイクル時に、CANTSC レジスタの値が変化しないように、ホールドレジスタが用意されています。メッセージが正常に受信または送信されると、CANTSC の値はホールドレジスタにコピーされ、そこからメールボックスにライトされます。エンドオブフレームの最後の 1 ビットを除いてエラーがない場合、レシーバでの受信が成功します。エンドオブフレームの最終ビットまでエラーがない場合、トランスミッタでの送信が成功します。(CAN Specification 2.0B を参照してください)

15.4.28 CANTSC( )

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

Bit	Bit Symbol	Type	
31-16		R	: "0" : "0"
15-0	TSC[15:0]	R/W	16

CANTSC のオーバフローは、グローバル割り込みフラグレジスタ CANGIF のタイムスタンプカウンタオーバフロー割り込みフラグ<TSOIF>およびグローバルステータスレジスタ CANGSR のタイムスタンプカウンタオーバフローフラグ<TSO>によって検出できます。両フラグは、CANGIF レジスタの<TSOIF>に "1" を書き込むことによってクリアできます。

CANTSC 用に 4 ビットのプリスケアラがあります。電源投入後、タイムスタンプカウンタはビットクロックから直接ドライブされます (<TSP[3:0]> = 0)。タイムスタンプカウンタ用の周期 T<sub>TSC</sub> は以下の式で計算できます。

$$T_{TSC} = T_{BIT} \times (TSP + 1)$$

## 15.5

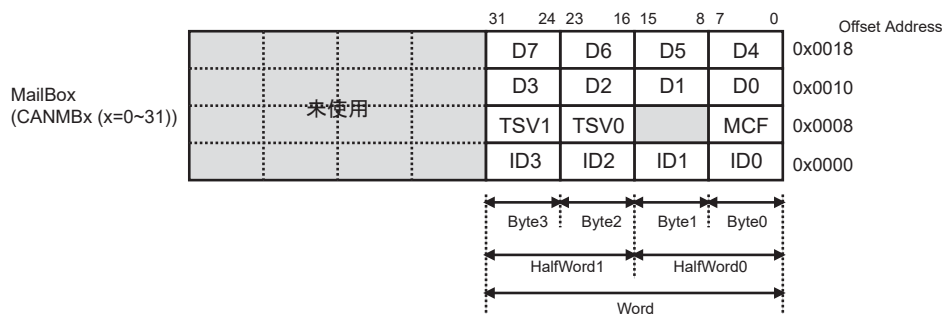
## 15.5.1

メールボックスは、(内部 CAN コアおよび CPU からアクセス可能な) シングルポート RAM で構成されます。CPU はメールボックスや制御レジスタを変更することによって CAN コントローラを制御します。メールボックスや制御レジスタの内容は、受信フィルタリング、メッセージ送信、割り込み処理などを実行するために使われます。

送信を開始するには、送信するメールボックスに対応する送信要求ビットをセットします。その後、CPU が関知することなく、送信のすべての手続きとエラー処理 (エラー発生の場合) を実行します。メールボックスが受信に設定されている場合、CPU はリード命令を使ってメールボックスのデータをリードします。メッセージの送信または受信が成功するたびに CPU に割り込みをかけるように設定することもできます。

メールボックスは 32 個用意され、各メールボックスは、8 バイトのデータ、29 ビットの ID、およびいくつかの制御ビットから構成されます。メールボックス 31 以外の各メールボックスは、送信または受信のどちらかに設定することができます。メールボックス 31 は受信専用のメールボックスです。メールボックス 31 には、メールボックス 0~30 とは別の受信マスクを使って異なるメッセージ ID のグループを受信できるように設計されています。

メールボックスの構成を図 15-2 に示します。



15-2

1. メッセージ ID フィールド (ID3 ~ ID0)
  - ・ ID 拡張ビット <IDE>
  - ・ グローバル/ローカル受信マスク許可ビット <GAME/LAME>
  - ・ リモートフレーム処理ビット <RFH>
  - ・ 29 ビットのメッセージ ID <ID[28:0]>
2. メッセージ制御フィールド (MCF)
  - ・ リモートフレーム送信要求ビット <RTR>
  - ・ 4 ビットのデータ長 <DLC[3:0]>
3. タイムスタンプ値 (TSV1, TSV0)
  - ・ メッセージ送信/受信時のタイムスタンプカウンタの値を格納 <TSV[15:0]>
4. データフィールド (D7 ~ D0)
  - ・ 8 バイトのデータ <D7[7:0]> ~ <D0[7:0]>

## 15.5.2

送信制御部は2個のレジスタで構成されています。1つは送信要求セットレジスタ CANTRS、もう1つは送信要求リセットレジスタ CANTRR です。これによって、ステートマシン内で送信メールボックスを取り扱う際に矛盾を起こすことなく送信要求をクリアできます。この仕組みは、送信が既に進行中のメールボックスがクリアされることも防止します。

送信メールボックス (CANMD<MDx>=0) として設定されたメールボックス x に、データと ID をライトし、メールボックス x へのアクセスが許可 (CANMC<MCx>=1) されているときに、CANTRS<TRSx>ビットを "1" にセットすると、メールボックス x のメッセージを送信します。

送信メールボックスとして構成されたメールボックスが2つ以上あり、対応する TRS ビットが2つ以上セットされている場合、メッセージは選択された順序で送信されます。送信順序は、マスタ制御レジスタ CANMCR の<MTOS>ビットに依存します。

CANMCR<MTOS>ビットが "0" の場合、メールボックス番号が小さいほど優先度が高くなります。たとえば、CANMB0、CANMB2 および CANMB5 が送信メールボックスに設定され、対応する CANTRS<TRSx>ビットが "1" にセットされているとき、メッセージは CANMB0、CANMB2、CANMB5 の順で送信されます。CANMB2 のメッセージを処理中に、新しい送信要求が CANMB0 に対してセットされた場合、次の内部アービトレーションは次の送信メッセージに CANMB0 を選択し、CANMB2 の送信終了後、CANMB0 の送信を開始します。これは、CANMB2 のメッセージ送信中にアービトレーションロストが発生した場合も同様です。アービトレーションロストした CANMB2 に代わり、CANMB0 のメッセージを送信します。

CANMCR<MTOS>ビットが "1" の場合、送信要求のあるメールボックスの中で、最も優先度の高い ID をもったメールボックスのメッセージが送信されます。アービトレーションロスト発生後の送信の場合も、その時点で送信要求のあるメールボックスの中でもっとも優先度の高い ID をもったメールボックスのメッセージを送信します。

15.5.3

受信メッセージの ID は、受信として設定されたメールボックスの ID と比較されます。ID の比較は、メールボックス中のグローバル/ローカル受信マスク有効ビット CANMBxID<GAME>/<LAME> の値およびグローバル/ローカル受信マスクレジスタ CANGAM/CANLAM に保持されるデータに依存します。

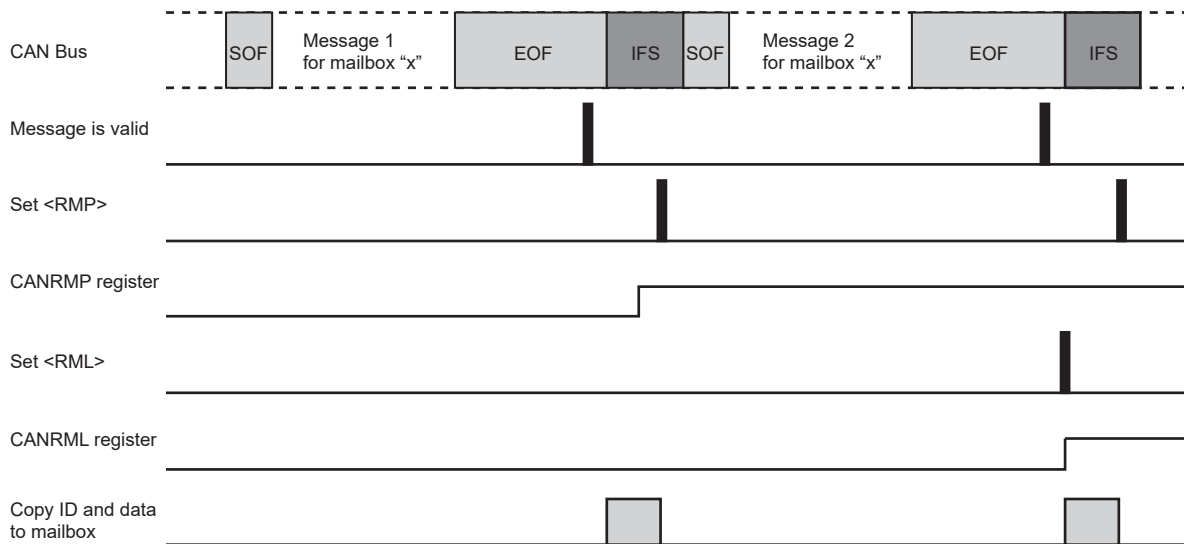
一致検出すると、受信メッセージの ID、制御ビットおよびデータバイトは、一致したメールボックスにライトされます。同時に、対応する受信メッセージペンディングビット CANRMP<RMPx> が "1" にセットされ、メールボックス割り込みを許可(CANMBIM<MBIMx>=1) している場合は、CAN 受信完了割り込み INTCANRX が発生します。一致検出すると、それ以降の ID 比較は行いません。

受信メッセージの ID がメールボックス 0 から 30 のいずれとも一致しない場合、受信専用メールボックス 31 の ID と比較されます。一致検出すると、受信メッセージの内容は、受信専用メールボックス 31 にライトされます。

一致検出しなかった場合、受信メッセージはメールボックスに格納されず、メールボックスに変化はありません。

<RMPx>ビットは、データの読み出し後に CPU がクリアしなければなりません。<RMPx>ビットが "1" にセットされた状態で、このメールボックス x への次のメッセージを受信した場合、対応する受信メッセージロストビット<RMLx>が "1" にセットされます。この場合、メールボックス x は新しいメッセージでオーバライトされます。

受信メッセージロストの発生タイミングを、図 15-3 に示します。



15-3

#### 15.5.4

リモートフレームを受信すると、リモートフレームの ID とメールボックスの ID を比較します。ID の比較は、メールボックス中のグローバル/ローカル受信マスク有効ビット CANMBxID<GAME>/<LAME> の値およびグローバル/ローカル受信マスクレジスタ CANGAM/CANLAM に保持されるデータに依存します。

ID の一致検出があると、それ以降の比較は行いません。

リモートフレーム処理ビット CANMBxID<RFH>を "1" にセットしている、送信メールボックス x の ID と一致すると、リモートフレームにตอบสนองしてメッセージを送信するために、CANTRS<TRSx>ビットが "1" にセットされます。ID が一致しても、CANMBxID<RFH>ビットが "0" の送信メールボックスの場合は、リモートフレームにตอบสนองしません。

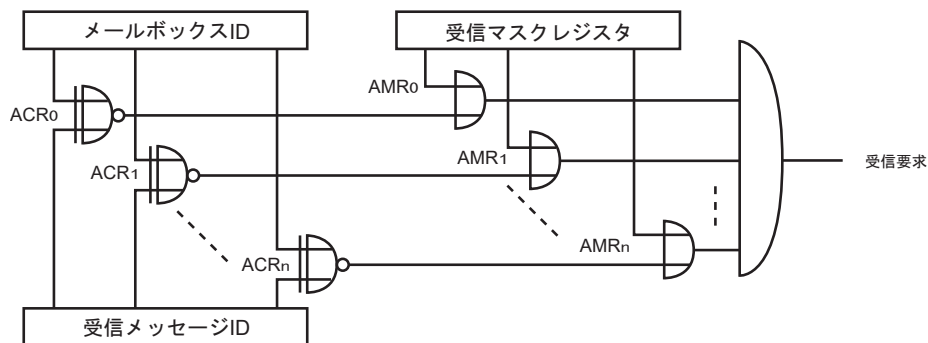
受信メールボックス x で ID が一致した場合は、この受信メッセージはデータフレームと同様に扱われ、CANRMP<RMPx>ビットと、CANRFP<RFPx>ビットが "1" にセットされます。

リモートフレームの ID と、CANMBxID<RFH>ビットと<GAME>ビットがともに "1" のメールボックス x の ID が一致した場合、メールボックス x の ID はリモートフレームの ID がオーバーライトされ、その ID を付けて自動応答 (<TRSx>ビットをセットしデータフレームを送信) します。したがって、グローバル受信マスクレジスタ CANGAM を使用した場合、マスクの値によっては、1つのメールボックス x が複数のリモートフレーム ID に対して応答する可能性があります。

## 15.5.5

メールボックス 0~30 の場合、グローバル受信マスクレジスタ CANGAM は、メールボックス内の <GAME> がセットされているときに使用されます。受信メッセージは、ID が一致する最初のメールボックスに格納されます。メールボックス 0~30 に一致する ID がない場合にだけ、受信メッセージは受信専用メールボックス (メールボックス 31) と比較されます。メールボックス 31 内の <LAME> ビットがセットされている場合、ローカル受信マスクレジスタ CANLAM が使用されます。

受信フィルタリングを、図 15-4 に示します。



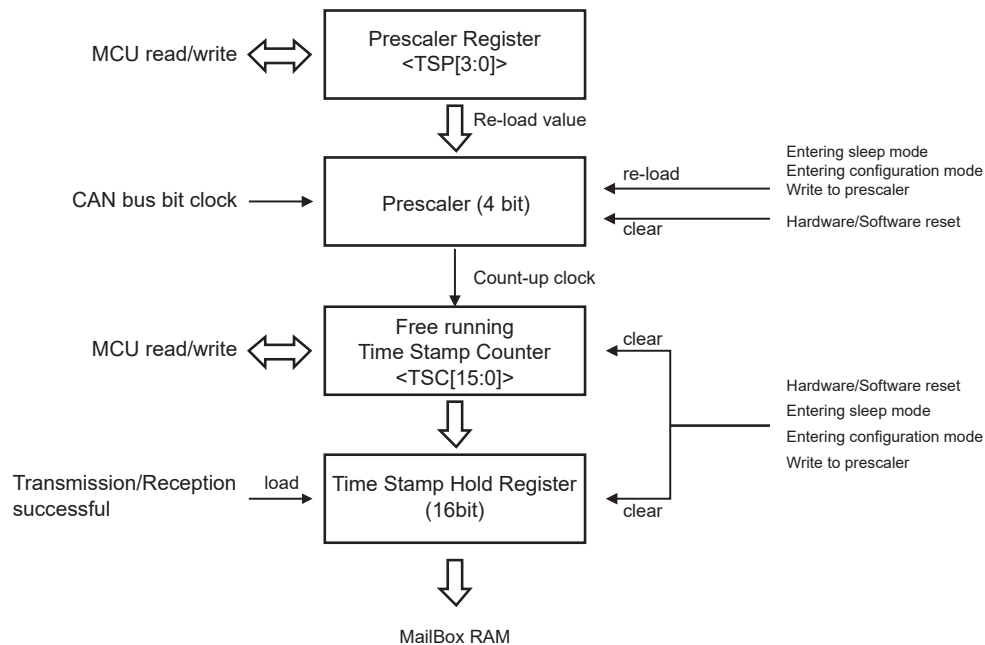
15-4

### 15.5.6

メッセージの送受信時を表示するために、CAN コントローラにはフリーランニング方式の 16 ビットタイムスタンプカウンタ CANTSC を実装しています。受信メッセージが格納されたとき、またはメッセージが送信されたときに、対応するメールボックスのタイムスタンプ値 TSV に CANTSC の内容が書き込まれます。

CANTSC には、CAN バスラインのビットクロックからドライブされます。CAN がコンフィグレーションモード、またはスリープモードのとき、CANTSC は停止します。電源投入リセット後、タイムスタンプカウンタプリスケアラレジスタ CANTSP へのライト動作により、CANTSC が "0" にクリアされます。CANTSC は、コンフィグレーションモード時およびノーマルオペレーションモード時も CPU からリード/ライトが可能です。

タイムスタンプカウンタの構成を、図 15-5 に示します。



15-5

フリーランニング方式のタイムスタンプカウンタおよびタイムスタンプ保持レジスタは、以下のケースでクリアされます。

- ・ リセット (電源投入リセットまたはソフトウェアリセット) 後
- ・ コンフィグレーションモードになったとき
- ・ スリープモードになったとき
- ・ CANTSP レジスタにライトアクセスしたとき

15.5.7

CAN コントローラには以下の割り込み要因があります。また、これらの割り込み要因は 3 グループに分けられ、各割り込みグループに 1 本ずつ割り込み出力信号があります。

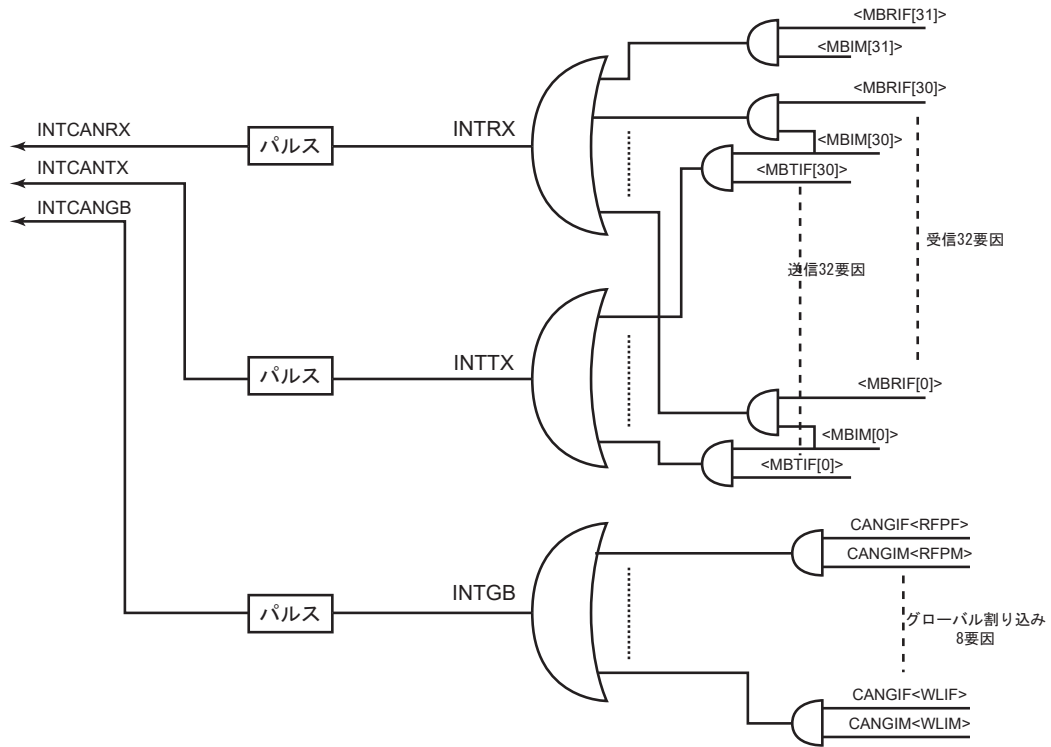
- ・ CAN 送信完了割り込み(INTCANTX)  
送信完了で発生
- ・ CAN 受信完了割り込み(INTCANRX)  
受信完了で発生
- ・ CAN グローバル割り込み(INTCANGB)  
以外の 8 要因で発生

	:	INTCANTX
	:	INTCANRX
	:	1 97
	:CAN	INTCANGB
	:CAN	
	:	

メールボックス割り込みのために、グローバル割り込みとは別の割り込みを 2 本出力します。これは、メールボックスの設定に依存する、メールボックス受信完了割り込み INTCANRX とメールボックス送信完了割り込み INTCANTX です。

割り込みフラグレジスタが 2 つ、割り込みマスクレジスタが 1 つ用意されています。割り込みフラグレジスタの 1 つはメールボックス受信割り込みフラグレジスタ CANMBRIF、もう 1 つはメールボックス送信割り込みフラグレジスタ CANMBTIF です。また、各メールボックス割り込みを許可するか禁止するかを設定する、メールボックス割り込みマスクレジスタ CANMBIM があります。CANMBIM レジスタは送信メールボックスと受信メールボックスの両方に使用されます。

CAN 割り込み信号ブロック図を、図 15-6 に示します。



### 15-6 CAN

メールボックス受信割り込みフラグレジスタ CANMBRIF の 32 要因と、メールボックス割り込みマスクレジスタ CANMBIM の各ビットを AND した信号の OR をとった信号が、CAN 受信完了割り込み信号 INTRX です。

メールボックス送信割り込みフラグレジスタ CANMBTIF の 31 要因と、メールボックス割り込みマスクレジスタ CANMBIM の各ビットを AND した信号の OR をとった信号が、CAN 送信完了割り込み信号 INTTX です。

グローバル割り込みフラグレジスタ CANGIF の 8 要因と、グローバル割り込みマスクレジスタ CANGIM の各ビットを AND した信号の OR をとった信号が、CAN グローバル割り込み信号 INTGB です。

## 15.6

### 15.6.1

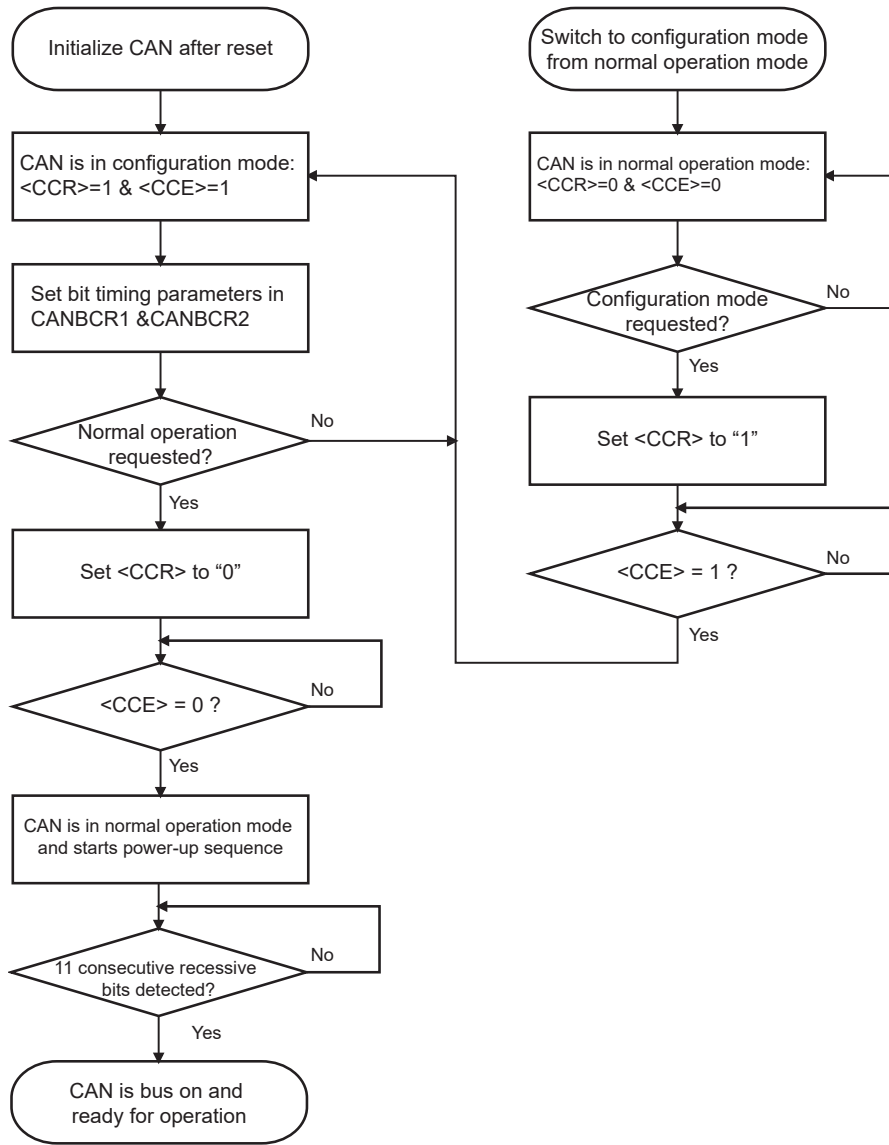
CAN コントローラは動作開始前に初期設定（ビットコンフィグレーションレジスタ CANBCR1, CANBCR2 の設定）が必要です。CANBCR1, CANBCR2 レジスタへのライトは、CAN コントローラのモードがコンフィグレーションモードのときのみ可能です。

リセット後、CANMCR<CCR>および CANGSR<CCE>は "1" にセットされ、コンフィグレーションモードとなります。CAN コントローラは、<CCR>に "0" をライトすることによってノーマルオペレーションモードになります。コンフィグレーションモードを抜けると、<CCE>は "0" にクリアされ、パワーアップシーケンスを開始します。パワーアップシーケンスは、CAN バスライン上で 11 個の連続する劣勢ビットを検出します。検出後、CAN コントローラはバスオン状態となり、動作の準備が完了します。

<CCR>に "1" をライトすると、CAN コントローラはノーマルオペレーションモードからコンフィグレーションモードに移行します。コンフィグレーションモードになると、<CCE>が "1" にセットされます。

CAN コントローラ初期設定のフローチャートを、図 15-7 に示します。

コンフィグレーションモードのときは、エラーカウンタ CANCEC、タイムスタンプカウンタ CANTSC およびタイムスタンプ保持レジスタがクリアされます。



15-7 CAN

## 15.6.2

CANMCR<SMR>に "1" をライトすることで、スリープモードを要求します。スリープモードになると、CANGSR<SMA>が "1" にセットされます。

CANGSR レジスタのリード値は 0xF040 です。これは、送信バッファにメッセージがない状態で <SMA>ビットが "1" で、スリープモードがアクティブであることを表しています。これ以外のすべてのレジスタに対するリード値は 0x0000 です。CANMCR レジスタ以外のすべてのレジスタに対するライトアクセスは拒否されます。

CANMCR レジスタへのライトアクセスか、CANMCR<WUBA>が "1" のときに CAN バス上のバスアクティブ状態を検出すると、CAN コントローラはスリープモードを解除（ウェイクアップ）し、パワーアップシーケンスを開始します。RX 入力端子上で、連続する 11 ビットの劣勢ビットを検出するまで待つ、バスアクティブ状態に移行します。ウェイクアップメッセージは無効です。

スリープモード中は、CAN エラーカウンタとすべての送信要求セット CANTRS<TRSx>と送信要求リセット CANTRR<TRRx>はクリアされます。<SMR>および<SMA>は、スリープモードを抜けた後にクリアされます。

CAN コントローラがメッセージを送信中にスリープモードが要求された（CANMCR<SMR>=1）場合、下記のいずれかの後にスリープモードに移行します。

- ・ 送信の正常終了
- ・ アービトレーションロスト後、送信の正常終了
- ・ アービトレーションロスト後、受信の正常終了

## 15.6.3

サスペンドモードは、CANMCR<SUR>に "1" をライトすると要求されます。CAN バスラインがアイドルでない場合、メッセージの現在の送受信が完了してから、サスペンドモードがアクティブになります。CAN コントローラがサスペンドモードになると、CANGSR<SUA>が "1" にセットされます。

サスペンドモード中は、CAN コントローラは CAN バスライン上ではアクティブではありません。つまり、エラーフレームもアクノリッジも送信しません。エラーカウンタや CANGSR<EP>もクリアされません。

バスオフリカバリシーケンス実行中にサスペンドモードが要求された場合、CAN コントローラは、バスオフリカバリシーケンス終了後にサスペンドモードに移行します。

CAN コントローラをリスタートさせるには、<SUR>をプログラムで "0" にしなければなりません。バスオフ状態またはインアクティブ状態を抜けた後、CAN コントローラはバスオフリカバリシーケンスをリスタートします。

CAN コントローラは、<SUR>に "0" をライトすることによってサスペンドモードを解除します。

### 15.6.4

テストループバックモードでは、CAN コントローラが自身からの送信メッセージを受信でき、アクノリッジビットを発生します。この動作には他の CAN ノードは不要です。

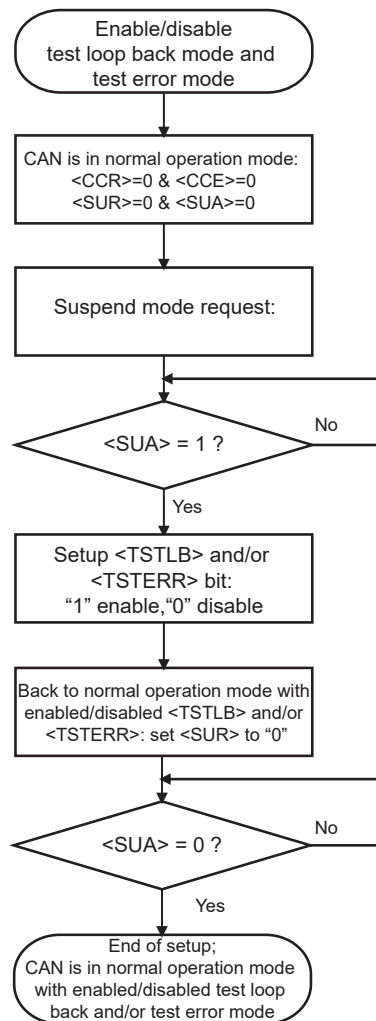
テストループバックモードは、CAN コントローラがサスペンドモードのときだけ、有効または無効の切り替えが可能です。テストループバックモードでは、CAN コントローラはメッセージをあるメールボックスから送信して、別のメールボックスで受信できます。メールボックスの設定は、ノーマルオペレーションモードと同じです。

### 15.6.5

テストエラーモードでは、CAN エラーカウンタレジスタ CANCEC へのライトが可能になります。下位 8 ビットの値が、送信エラーカウンタ TEC と受信エラーカウンタ REC の両方に同時にライトされます。ライトできる最大値は 255 です。CAN コントローラがバスオフになるカウント値 256 はライトできません。

テストエラーモードは、CAN コントローラがサスペンドモードのときだけ、有効または無効の切り替えが可能です。

テストループバックモード、テストエラーモード設定のフローチャートを、図 15-8 に示します。

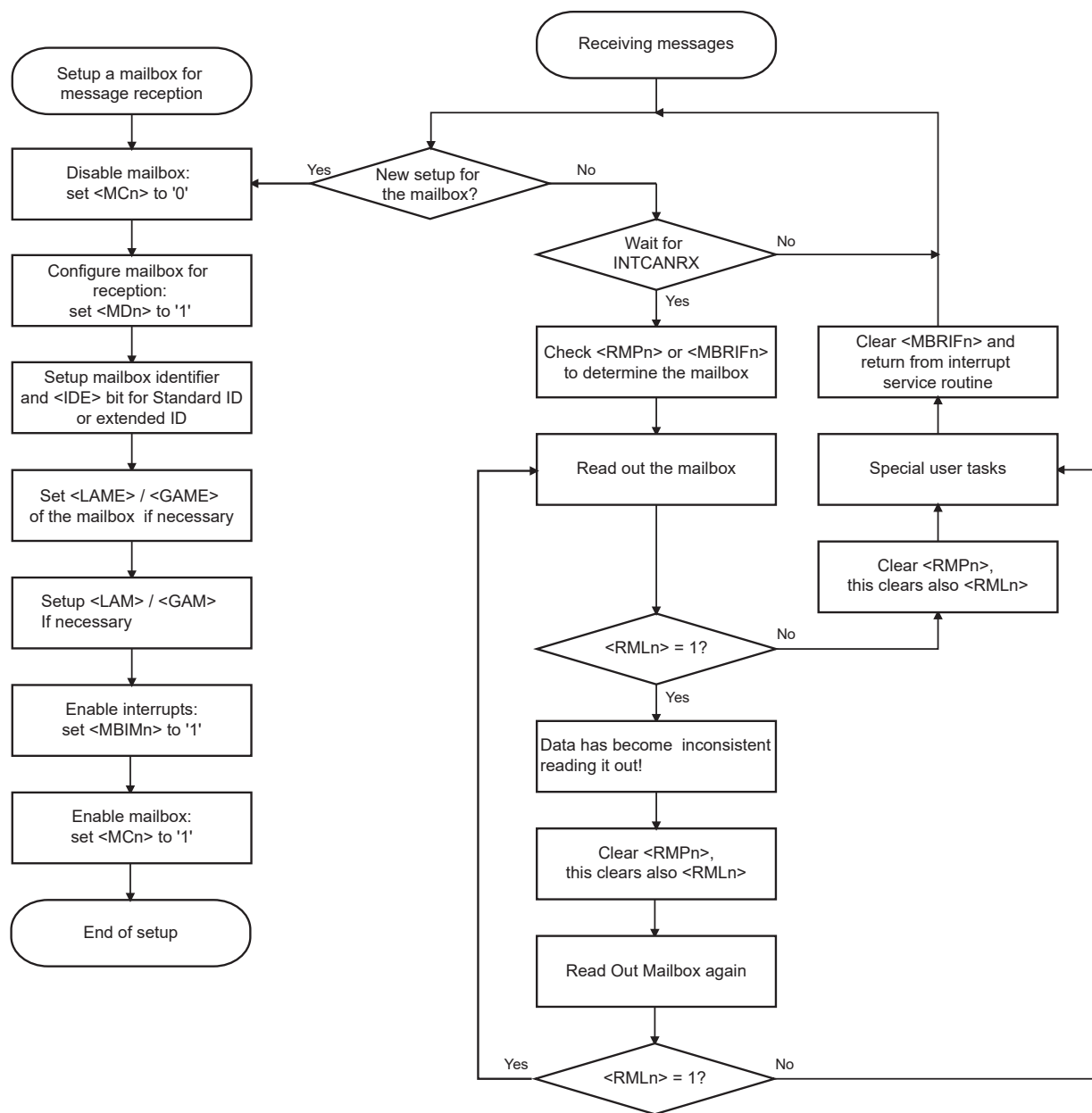


15-8

15.7

15.7.1

CAN 受信完了割り込み INTCANRX を使用したメッセージ受信のフローチャートの一例を、図 15-9 に示します。

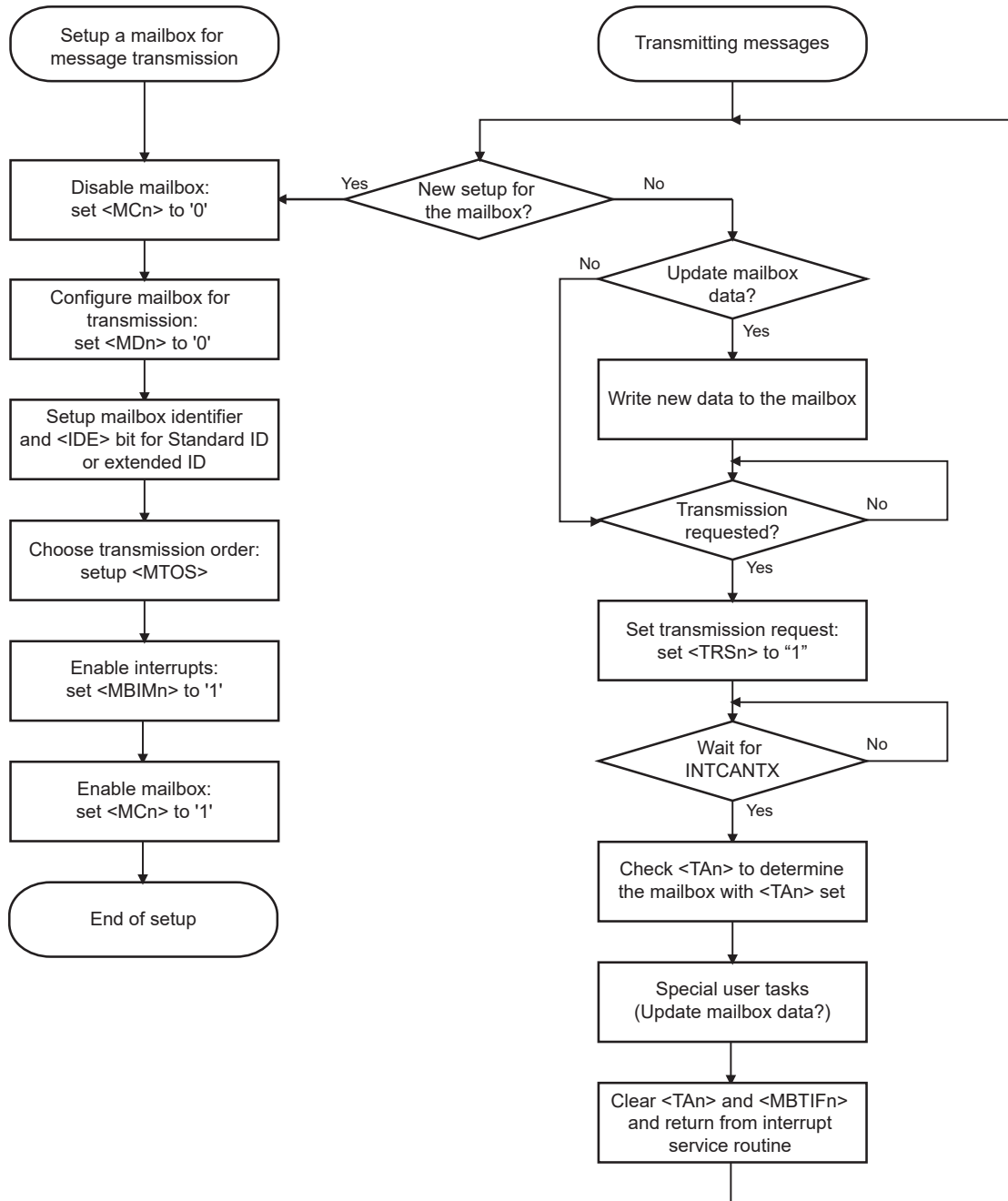


15-9

受信割り込みの代わりにポーリングを使用することも可能です。この場合、上のフローチャートにて"waiting for INTCANRX"を RMP のポーリングで置き換える必要があります。さらに、割り込みの許可と CANMBRIF のクリアをフローから削除する必要があります。

15.7.2

CAN 送信完了割り込み INTCANTX を使用したメッセージ送信のフローチャートの一例を、図 15-10 に示します。

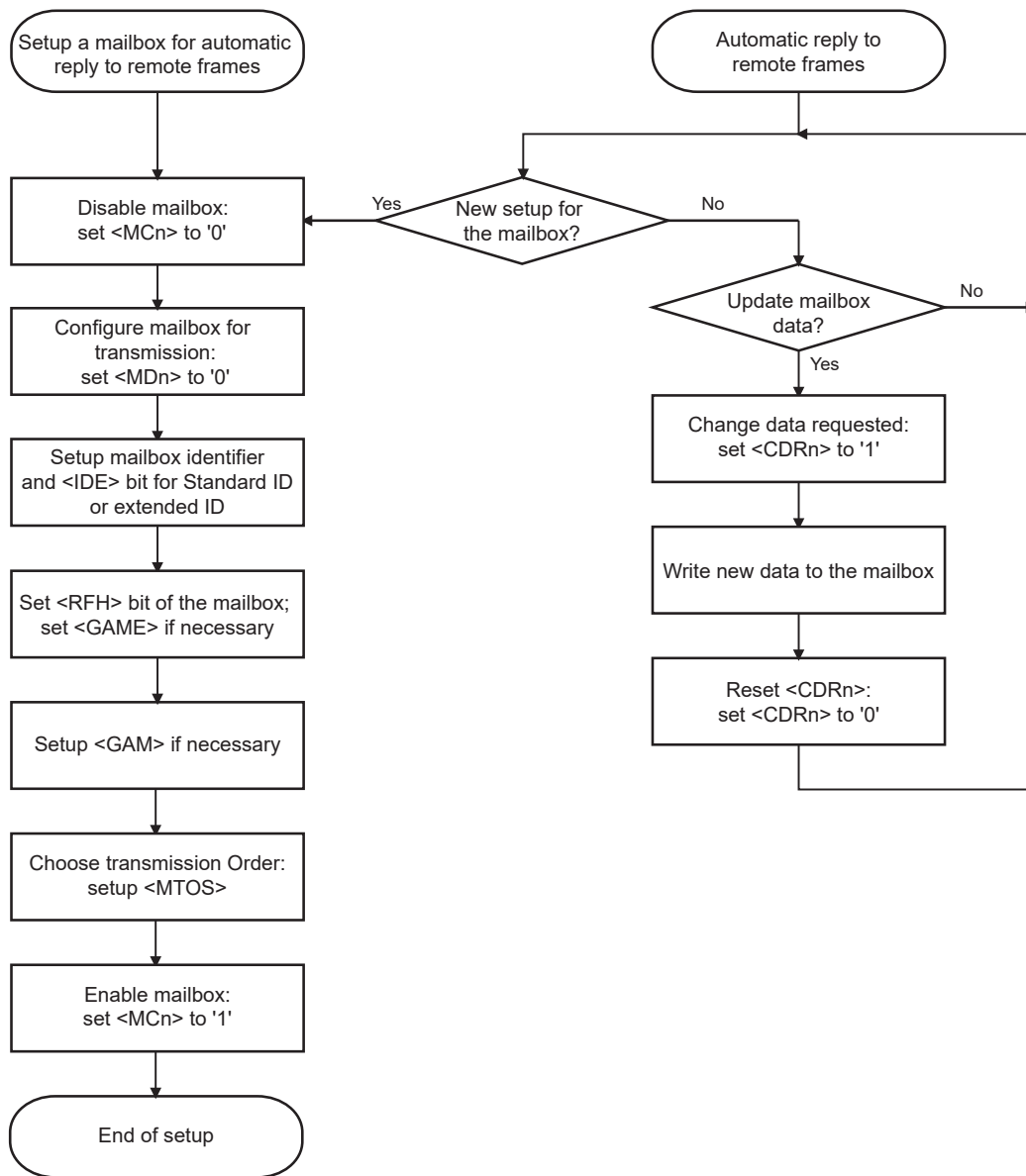


15-10

送信割り込みの代わりにポーリングを使用することも可能です。この場合、上のフローチャートにて"waiting for INTCANTX"を TA のポーリングで置き換える必要があります。さらに、割り込みの許可と CANMBTIF のクリアをフローから削除する必要があります。

15.7.3

自動応答機能を使ったリモートフレーム処理のフローチャートの一例を、図 15-11 に示します。この機能は送信メールボックスの<RFH>ビットが"1"にセットされているときに有効です。メールボックスのデータ更新時のデータ不一致をさけるために、CANCDR レジスタによる送信制御を行い、メールボックスのデータを更新します。

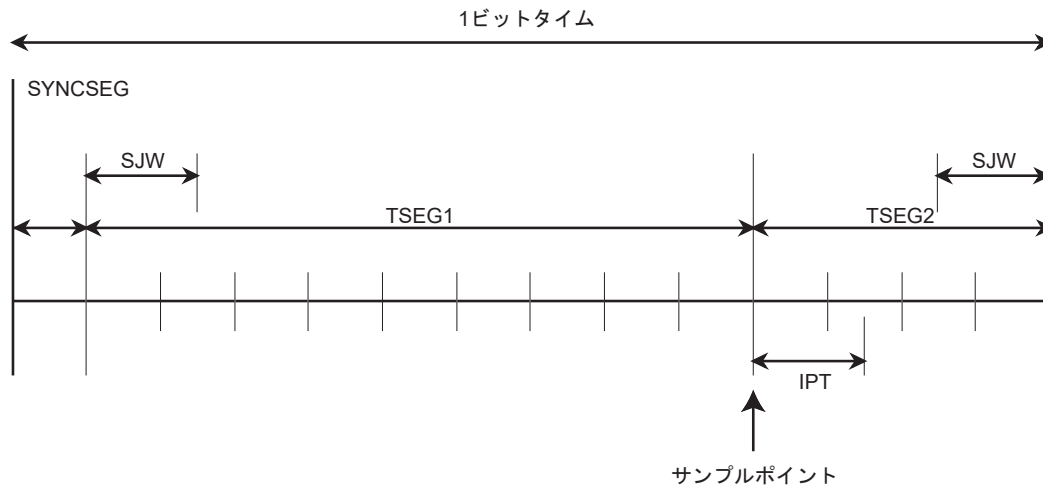


15-11

## 15.8

ビット長は、パラメータ TSEG1、TSEG2 および BRP によって決まります。CAN バス上のすべてのコントローラのボーレートおよびビット長は同一でなければなりません。個々のコントローラのクロック周波数が異なる場合は、ボーレートを上記のパラメータで調整しなければなりません。ビットタイミングロジックでは、要求されるビットタイミングへのパラメータ変換が実施されます。コンフィグレーションレジスタ CANBCR1、CANBCR2 にはビットタイミングに関するデータがあります。その定義は CAN Specification 2 (Intel 82527 同等) に対応します。

CAN ビットタイミングを、図 15-12 に示します。



15-12 CAN

$T_{SCL}$  (CAN システムクロック) は以下のように定義されます。

$$T_{SCL} = \frac{\langle BRP[9:0] \rangle + 1}{f_{CANOSC}}$$

$1 \times T_{SCL} = 1 \times T_Q$  ( $T_Q$ : タイムクアンタム)

$f_{CANOSC}$  は CAN ボーレート生成用クロックです。システムクロック  $f_{SYS}$  を 4 分周したクロックが、CAN ボーレート生成用クロックとして供給されます。 $f_{SYS} = 48\text{MHz}$  のとき、 $f_{CANOSC} = 12\text{MHz}$  です

同期セグメント SYNCSEG の長さは常にタイムクアンタム ( $T_Q$ ) 1 個分です。

ボーレートは以下のように定義されます。

$$BR = \frac{1}{((\langle TSEG1[3:0] \rangle + 1) + (\langle TSEG2[2:0] \rangle + 1) + 1) \times T_{SCL}}$$

)  $\langle TSEG1[3:0] \rangle, \langle TSEG2[2:0] \rangle$  CANBCR2  $T_Q$

IPT (情報処理時間) とは、サンプルされたビットレベルの処理用に予約されたサンプルポイントで始まる時間の一区切りです。この情報処理時間は、CAN システムクロックの 3 サイクル分です。

<SJW[1:0]> は、再同期時にビット長をどれだけタイムクアンタム ( $T_Q$ ) 値、延長または短縮を許可するかを示します。この値の調整可能範囲は "1" (<SJW[1:0]> = 00) から "4" (<SJW[1:0]> = 11) です。1 ビットグリッド以内のバス信号の立ち下がりエッジごとにバスラインがサンプルされ、同期が取られます。<SJW[1:0]> は <TSEG2[2:0]> 以下の値を設定します。

<SAM> ビットをセットすると、バスラインのマルチサンプリングが可能になります。3 回のサンプリング値の多数決でレベルを決定します。サンプルポイントと、その前 2 回の CAN システムクロックのポイントでサンプリングします。<BRP[9:0]> < 4 のときは、<SAM> ビットの設定値にかかわらず、常に 1 回のサンプリングになります。

ボーレート設定時の制限事項を、表 15-2 に示します。

15-2

<BRP[9:0]>	$T_Q$ (CAN )	IPT (CAN )	TSEG2 ( $T_Q$ )
0	1	3	3
1	2	3	2
> 1	<BRP[9:0]>+1	3	2

- ・ TSEG1 の制限事項

TSEG1  $\geq$  TSEG2 : TSEG1 長は TSEG2 長以上でなければなりません。

- ・ SJW の制限事項

SJW  $\leq$  TSEG2 : 同期ジャンプ幅は TSEG2 以下の値を設定します。

- ・ SAM の制限事項

3 回サンプリングは <BRP[9:0]> < 4 には許可されません。<BRP[9:0]> < 4 に対しては、SAM の値にかかわらず、1 回サンプリングとなります。

例:500 Kbit/s の場合

1 ビットは  $2s$  の長さです。クロック周波数  $f_{CANOSC}$  が 12MHz の場合、ボーレートプリスケアラを "1" にセットします。すなわち、この送信速度での 1 ビットは  $12T_Q$  の長さでプログラミングしなければなりません。上述の数式によって、プログラミングすべき値は計算値より常に 1 だけ少なくなります。

<BRP[9:0]> = 00\_0000\_0001

<TSEG1[3:0]> = 0110 (7  $T_Q$ )

<TSEG2[2:0]> = 011 (4  $T_Q$ )

この設定では、サンプルポイントは  $8/12=66\%$  です。

他の TSEG1/TSEG2 の組み合わせも可能です。TSEG2=3 は SJW の最大値です。

SJW は常にできるだけ高い値にセットしてください。SJW を TSEG2 より大きくすることは許可されません。

バスの 3 回サンプリングは <BRP[9:0]> < 4 のため設定できません。従って SAM = 0 と設定します。

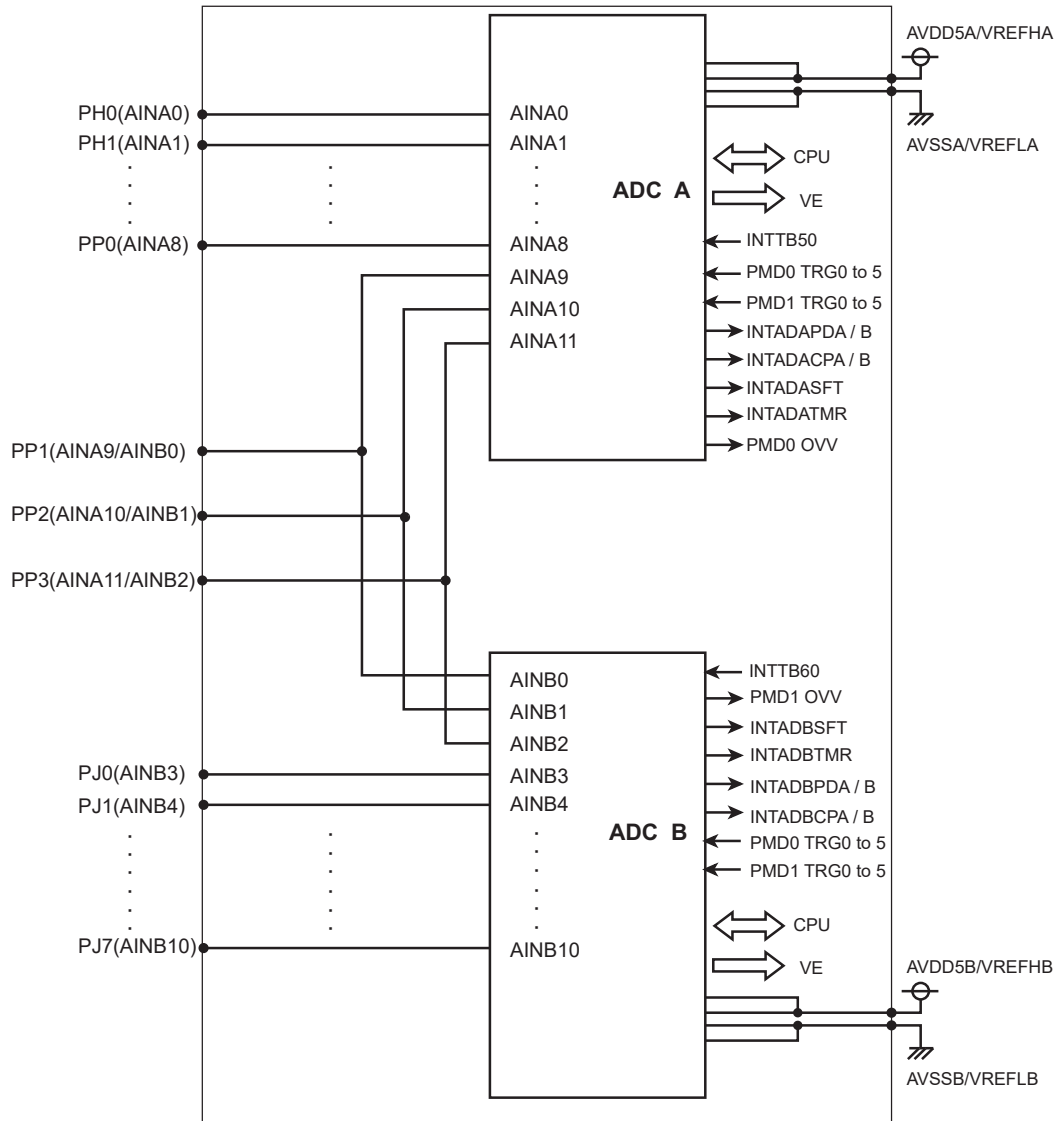


16            12            /

## 16.1

1. PMD やタイマからのトリガ信号に同期して任意のアナログ入力を変換することができます。
2. ソフトウェア起動、常時起動において任意のアナログ入力を変換することができます。
3. AD 変換値レジスタが 12 個あります。
4. トリガ起動によるプログラム終了時に割り込みを発生できます。
5. ソフトウェア起動によるプログラム終了時に割り込みを発生できます。
6. AD 監視機能があります。有効時に比較条件と一致した場合は割り込みを発生します。

16.2



16-1 AD

## 16.3

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

(x=A,B)		Address(Base+)
		ADxCLK
		0x0000
0		ADxMOD0
		0x0004
1		ADxMOD1
		0x0008
2		ADxMOD2
		0x000C
0		ADxCMPCR0
		0x0010
1		ADxCMPCR1
		0x0014
0		ADxCMP0
		0x0018
1		ADxCMP1
		0x001C
0		ADxREG0
		0x0020
1		ADxREG1
		0x0024
2		ADxREG2
		0x0028
3		ADxREG3
		0x002C
4		ADxREG4
		0x0030
5		ADxREG5
		0x0034
6		ADxREG6
		0x0038
7		ADxREG7
		0x003C
8		ADxREG8
		0x0040
9		ADxREG9
		0x0044
10		ADxREG10
		0x0048
11		ADxREG11
		0x004C
PMD	0	ADxPSEL0
		0x0050
PMD	1	ADxPSEL1
		0x0054
PMD	2	ADxPSEL2
		0x0058
PMD	3	ADxPSEL3
		0x005C
PMD	4	ADxPSEL4
		0x0060
PMD	5	ADxPSEL5
		0x0064
PMD	6	ADxPSEL6
		0x0068
PMD	7	ADxPSEL7
		0x006C
PMD	8	ADxPSEL8
		0x0070
PMD	9	ADxPSEL9
		0x0074
PMD	10	ADxPSEL10
		0x0078
PMD	11	ADxPSEL11
		0x007C
PMD	0	ADxPINTS0
		0x0080
PMD	1	ADxPINTS1
		0x0084
PMD	2	ADxPINTS2
		0x0088
PMD	3	ADxPINTS3
		0x008C
PMD	4	ADxPINTS4
		0x0090
PMD	5	ADxPINTS5
		0x0094
PMD	0	ADxPSET0
		0x0098
PMD	1	ADxPSET1
		0x009C
PMD	2	ADxPSET2
		0x00A0
PMD	3	ADxPSET3
		0x00A4

(x=A,B)		Address(Base+)	
PMD	4	ADxPSET4	0x00A8
PMD	5	ADxPSET5	0x00AC
	0 ~ 3	ADxTSET03	0x00B0
	4 ~ 7	ADxTSET47	0x00B4
	8 ~ 11	ADxTSET811	0x00B8
	0 ~ 3	ADxSSET03	0x00BC
	4 ~ 7	ADxSSET47	0x00C0
	8 ~ 11	ADxSSET811	0x00C4
	0 ~ 3	ADxASET03	0x00C8
	4 ~ 7	ADxASET47	0x00CC
	8 ~ 11	ADxASET811	0x00D0
	3	ADxMOD3	0x00D4
	4	ADxMOD4	0x00D8
	5	ADxMOD5	0x00DC

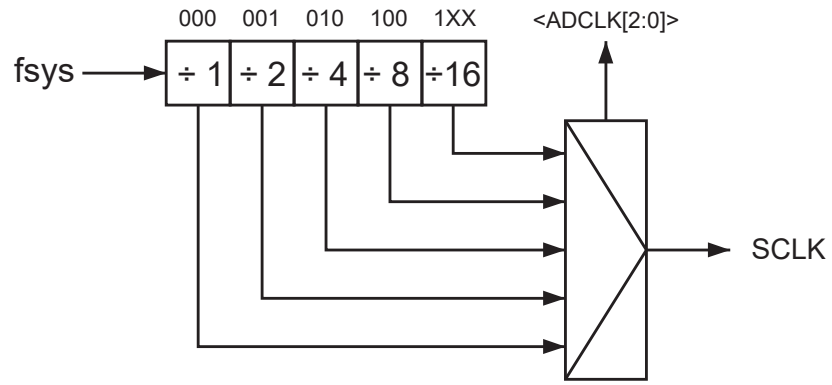
16.4

AD 変換は AD コンバータクロック設定レジスタによって選択されたクロックで実行されます。

16.4.1 ADxCLK ( )

	31	30	29	28	27	26	25	24	
bit symbol	-	-	-	-	-	-	-	-	
	0	0	0	0	0	0	0	0	
	23	22	21	20	19	18	17	16	
bit symbol	-	-	-	-	-	-	-	-	
	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	
bit symbol	-	-	-	-	-	-	-	-	
	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
bit symbol	-	EXAZ				VADCLK			
	0	0	0	0	0	0	0	0	

Bit	Bit Symbol	Type	
31-7		R	"0"
6-3	EXAZ[3:0]	R/W	AIN  0000: (SCLK ×34)×1 0001: (SCLK ×34)×2 0010: (SCLK ×34)×3 0011: (SCLK ×34)×4 0101: (SCLK ×34)×16 0111: (SCLK ×34)×64 1000: (SCLK ×34)×128 1001: (SCLK ×34)×256 1010: (SCLK ×34)×512 1011: (SCLK ×34)×1024  : Reserved
2-0	VADCLK[2:0]	R/W	AD (SCLK) 000: fsys ( 1) 001: fsys/2 010: fsys/4 011: fsys/8 1xx: fsys/16



- 1) SCLK      120MHz
- 2) AD
- 3) AD      AD

16.4.2 ADxMOD0 ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	DACON	ADSS
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2		R	"0"
1	DACON	R/W	DAC 0: OFF 1: ON AD <DACON> "1"
0	ADSS	W	0: Don't care 1: ADxMOD1 <ADEN> "1" PMD AD <ADSS> "1" PMD AD PMD, AD <ADSS> "1" PMD AD

) AD ) ADxMOD0<DACON> "1" ADxMOD1<ADEN> "1"(AD  
ADxMOD0<DACON> "1" 3μs AD

16.4.3 ADxMOD1 ( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADEN	-	-	-	-	-	-	ADAS
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	ADEN	R/W	AD / 0: 1: <ADEN> AD
6-1		R	"0"
0	ADAS	R/W	AD 0: 1: AD (<ADEN>=1) <ADAS> AD

16.4.4 ADxMOD2 ( 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	ADSFN	ADBFN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2		R	"0"
1	ADSFN	R	0: 1: <ADSFN> AD Busy <ADSS> "1" AD "1" "0"
0	ADBFN	R	AD BUSY 0: 1: <ADBFN> AD Busy (PMD, , , ) "1" "0"

16.4.5 ADxMOD3 ( 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	PBSEL	-	BINMOD	BITS		-	RCUT
	0	0	0	0	0	0	0	1
	7	6	5	4	3	2	1	0
bit symbol	BIAS			-	-	-	-	-
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15		R/W	"0"
14	PBSEL	R/W	"0"
13		R	"0"
12	BINMOD	R/W	"0"
11-10	BITS[1:0]	R/W	"00"
9		R/W	"0"
8	RCUT	R/W	0: 1: Iref cut ADC ADC "1"
7-5	BIAS[2:0]	R/W	"000"
4-0		R	"0"

16.4.6 ADxMOD4 ( 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	PHASEC			
	0	0	0	0	0	0	1	0
	7	6	5	4	3	2	1	0
bit symbol	PHASEB				-	PHASEA		
	0	0	0	1	0	0	1	0

Bit	Bit Symbol	Type	
31-12		R	"0"
11-8	PHASEC[3:0]	R/W	"0001"
7-4	PHASEB[3:0]	R/W	"0001"
3		R	"0"
2-0	PHASEA[2:0]	R/W	"001"

16.4.7 ADxMOD5 ( 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	PHASEF				PHASEE			
	0	1	1	0	0	1	1	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	PHASED				
	0	0	0	0	1	0	0	1

Bit	Bit Symbol	Type	
31-16		R	"0"
15-13	PHASEF[2:0]	R/W	"011"
12-8	PHASEE[4:0]	R/W	"00110"
7-5		R	"0"
4-0	PHASED[4:0]	R/W	"01101"

16.4.8 ADxCMPCR0( 0)

判定が確定すると割り込み(INTADxCPn)を発生します。(n=A,B、A:監視 0、B:監視 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT0			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP0EN	-	-	ADBIG0	REGS0			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type													
31-12	-	R	"0"												
11-8	CMPCNT0[3:0]	R/W	0: 1: 2  15: 16 AD ADCMP												
7	CMP0EN	R/W	AD 0: 1:												
6-5	-	R	"0"												
4	ADBIG0	R/W	0: REG 1: REG												
3-0	REGS0[3:0]	R/W	AD <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>0000: ADxREG0</td> <td>0100: ADxREG4</td> <td>1000: ADxREG8</td> </tr> <tr> <td>0001: ADxREG1</td> <td>0101: ADxREG5</td> <td>1001: ADxREG9</td> </tr> <tr> <td>0010: ADxREG2</td> <td>0110: ADxREG6</td> <td>1010: ADxREG10</td> </tr> <tr> <td>0011: ADxREG3</td> <td>0111: ADxREG7</td> <td>1011: ADxREG11</td> </tr> </table>	0000: ADxREG0	0100: ADxREG4	1000: ADxREG8	0001: ADxREG1	0101: ADxREG5	1001: ADxREG9	0010: ADxREG2	0110: ADxREG6	1010: ADxREG10	0011: ADxREG3	0111: ADxREG7	1011: ADxREG11
0000: ADxREG0	0100: ADxREG4	1000: ADxREG8													
0001: ADxREG1	0101: ADxREG5	1001: ADxREG9													
0010: ADxREG2	0110: ADxREG6	1010: ADxREG10													
0011: ADxREG3	0111: ADxREG7	1011: ADxREG11													

### 16.4.9 ADxCMPnCR1( 1)

判定が確定すると割り込み(INTADxCPn)を発生します。(n=A,B、A:監視 0、B:監視 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	CMPCNT1			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CMP1EN	-	-	ADBIG1	REGS1			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type													
31-12	-	R	"0"												
11-8	CMPCNT1[3:0]	R/W	0: 1: 2  15: 16 AD ADCMP												
7	CMP1EN	R/W	A/D 0: 1:												
6-5	-	R	"0"												
4	ADBIG1	R/W	0: REG 1: REG												
3-0	REGS1[3:0]	R/W	AD <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>0000: ADxREG0</td> <td>0100: ADxREG4</td> <td>1000: ADxREG8</td> </tr> <tr> <td>0001: ADxREG1</td> <td>0101: ADxREG5</td> <td>1001: ADxREG9</td> </tr> <tr> <td>0010: ADxREG2</td> <td>0110: ADxREG6</td> <td>1010: ADxREG10</td> </tr> <tr> <td>0011: ADxREG3</td> <td>0111: ADxREG7</td> <td>1011: ADxREG11</td> </tr> </table>	0000: ADxREG0	0100: ADxREG4	1000: ADxREG8	0001: ADxREG1	0101: ADxREG5	1001: ADxREG9	0010: ADxREG2	0110: ADxREG6	1010: ADxREG10	0011: ADxREG3	0111: ADxREG7	1011: ADxREG11
0000: ADxREG0	0100: ADxREG4	1000: ADxREG8													
0001: ADxREG1	0101: ADxREG5	1001: ADxREG9													
0010: ADxREG2	0110: ADxREG6	1010: ADxREG10													
0011: ADxREG3	0111: ADxREG7	1011: ADxREG11													

16.4.10 ADxCMP0( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP0				-	-	-	-
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	AD0CMP0[11:0]	R/W	AD AD
3-0	-	R	"0"

16.4.11 ADxCMP1( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	AD0CMP1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	AD0CMP1				-	-	-	-
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	AD0CMP1[11:0]	R/W	AD AD
3-0	-	R	"0"

16.4.12 ADxREG0( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR0				-	-	OVR0	ADR0RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR0[11:0]	R	AD
3-2	-	R	"0"
1	OVR0	R	Over Run 0 1  ADxREG0 AD "1" ADxREG0 "0"
0	ADR0RF	R	AD 0 1  AD "1" ADxREG0 "0"

16.4.13 ADxREG1( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR1				-	-	OVR1	ADR1RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR1[11:0]	R	AD
3-2	-	R	"0"
1	OVR1	R	Over Run 0 1  ADxREG1 AD "1" ADxREG1 "0"
0	ADR1RF	R	AD 0 1  AD "1" ADxREG1 "0"

16.4.14 ADxREG2( 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR2							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR2				-	-	OVR2	ADR2RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR2[11:0]	R	AD
3-2	-	R	"0"
1	OVR2	R	Over Run 0 1  ADxREG2 AD "1" ADxREG2 "0"
0	ADR2RF	R	AD 0 1  AD "1" ADxREG2 "0"

16.4.15 ADxREG3( 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR3							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR3				-	-	OVR3	ADR3RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR3[11:0]	R	AD
3-2	-	R	"0"
1	OVR3	R	Over Run 0 1  ADxREG3 AD "1" ADxREG3 "0"
0	ADR3RF	R	AD 0 1  AD "1" ADxREG3 "0"

16.4.16 ADxREG4( 4)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR4							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR4				-	-	OVR4	ADR4RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR4[11:0]	R	AD
3-2	-	R	"0"
1	OVR4	R	Over Run 0 1  ADxREG4 AD "1" ADxREG4 "0"
0	ADR4RF	R	AD 0 1  AD "1" ADxREG4 "0"

16.4.17 ADxREG5( 5)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR5							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR5				-	-	OVR5	ADR5RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR5[11:0]	R	AD
3-2	-	R	"0"
1	OVR5	R	Over Run 0 1  ADxREG5 AD "1" ADxREG5 "0"
0	ADR5RF	R	AD 0 1  AD "1" ADxREG5 "0"

16.4.18 ADxREG6( 6)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR6							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR6				-	-	OVR6	ADR6RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR6[11:0]	R	AD
3-2	-	R	"0"
1	OVR6	R	Over Run 0 1  ADxREG6 AD "1" ADxREG6 "0"
0	ADR6RF	R	AD 0 1  AD "1" ADxREG6 "0"

16.4.19 ADxREG7( 7)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR7							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR7				-	-	OVR7	ADR7RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR7[11:0]	R	AD
3-2	-	R	"0"
1	OVR7	R	Over Run 0 1  ADxREG7 AD "1" ADxREG7 "0"
0	ADR7RF	R	AD 0 1  AD "1" ADxREG7 "0"

16.4.20 ADxREG8( 8)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR8							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR8				-	-	OVR8	ADR8RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR8[11:0]	R	AD
3-2	-	R	"0"
1	OVR8	R	Over Run 0 1  ADxREG8 AD "1" ADxREG8 "0"
0	ADR8RF	R	AD 0 1  AD "1" ADxREG8 "0"

16.4.21 ADxREG9( 9)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR9							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR9				-	-	OVR9	ADR9RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR9[11:0]	R	AD
3-2	-	R	"0"
1	OVR9	R	Over Run 0 1  ADxREG9 AD "1" ADxREG9 "0"
0	ADR9RF	R	AD 0 1  AD "1" ADxREG9 "0"

16.4.22 ADxREG10( 10)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR10							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR10				-	-	OVR10	ADR10RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR10[11:0]	R	AD
3-2	-	R	"0"
1	OVR10	R	Over Run 0 1  ADxREG10 AD "1" ADxREG10 "0"
0	ADR10RF	R	AD 0 1  AD "1" ADxREG10 "0"

16.4.23 ADxREG11( 11)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ADR11							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ADR11				-	-	OVR11	ADR11RF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16	-	R	"0"
15-4	ADR11[11:0]	R	AD
3-2	-	R	"0"
1	OVR11	R	Over Run 0 1  ADxREG11 AD "1" ADxREG11 "0"
0	ADR11RF	R	AD 0 1  AD "1" ADxREG11 "0"

## 16.4.24 PMD

本 AD コンバータは PMD 回路が発生するトリガ信号によって AD 変換を開始する事ができます。

PMD トリガ用プログラムレジスタは PMD が発生する 12 種類のトリガによって起動するプログラム番号の設定、プログラム終了時に発生する割り込みの制御、AD 変換の AIN の設定などを行います。

PMD トリガ用プログラムレジスタは 3 種類のレジスタから構成されます。

(x=A,B : AD コンバータユニット)

- PMD トリガ用プログラム番号選択レジスタ(ADxPSEL0 ~ ADxPSEL11)

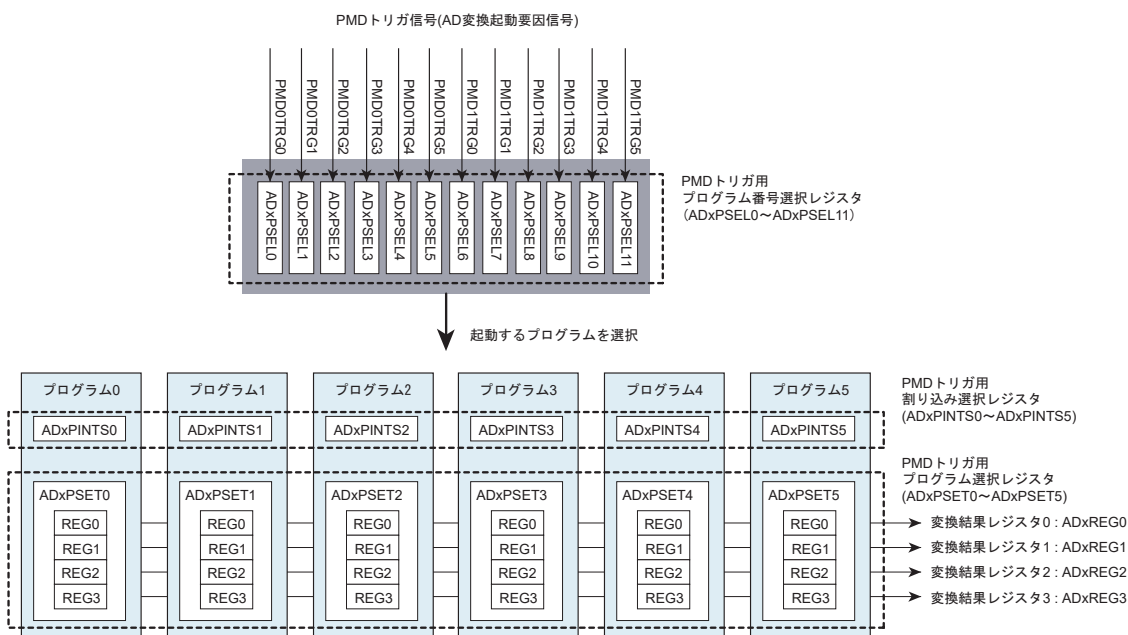
PMD からの 12 本のトリガ信号(PMD0TRG0 ~ 5, PMD1TRG0 ~ 5)に対して、それぞれ起動するプログラム番号(0 ~ 5)を選択するレジスタです。

ADxPSEL0 ~ ADxPSEL5 が PMD0TRG0 ~ 5, ADxPSEL6 ~ ADxPSEL11 が PMD1TRG0 ~ 5 に対応しています。
- PMD トリガ用割り込み選択レジスタ(ADxPINTS0 ~ ADxPINTS5)

それぞれのプログラム番号ごとにプログラム終了時に発生させる割り込みの有無や種類(INTADxPDA,INTADxPDB)を選択するレジスタです。

ADxPINTS0 がプログラム 0 に対応しており、ADxPINTS5(プログラム 5)まであります。
- PMD トリガ用プログラム選択レジスタ(ADxPSET0 ~ ADxPSET5)

プログラム番号(0 ~ 5)に対して、AD 変換する AIN 番号とベクトルエンジンへの通知用に U/V/W の相を設定します。プログラム番号 1 つに対して PMD トリガ用プログラム選択レジスタは 4 組あり、変換結果は、それぞれ変換結果格納レジスタ 0 ~ 3(ADxREG0 ~ 3)に格納されます。



16-2 PMD

16.4.24.1 ADxPSEL0 ~ ADxPSEL11(PMD

0 ~ 11)

ADxPSEL0 PMD

0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS0	-	-	-	-	PMDS0		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS0	R/W	PMD0TRG0 0: 1:
6-3	-	R	"0"
2-0	PMDS0[2:0]	R/W	( 16-1 )

ADxPSEL1 PMD

1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS1	-	-	-	-	PMDS1		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS1	R/W	PMD0TRG1 0: 1:
6-3	-	R	"0"
2-0	PMDS1[2:0]	R/W	( 16-1 )

ADxPSEL2 PMD 2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS2	-	-	-	-	PMDS2		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS2	R/W	PMD0TRG2 0: 1:
6-3	-	R	"0"
2-0	PMDS2[2:0]	R/W	( 16-1 )

ADxPSEL3 PMD 3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS3	-	-	-	-	PMDS3		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS3	R/W	PMD0TRG3 0: 1:
6-3	-	R	"0"
2-0	PMDS3[2:0]	R/W	( 16-1 )

ADxPSEL4 PMD 4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS4	-	-	-	-	PMDS4		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS4	R/W	PMD0TRG4 0: 1:
6-3	-	R	"0"
2-0	PMDS4[2:0]	R/W	( 16-1 )

ADxPSEL5 PMD 5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS5	-	-	-	-	PMDS5		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS5	R/W	PMD0TRG5 0: 1:
6-3	-	R	"0"
2-0	PMDS5[2:0]	R/W	( 16-1 )

ADxPSEL6 PMD 6

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS6	-	-	-	-	PMDS6		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS6	R/W	PMD1TRG0 0: 1:
6-3	-	R	"0"
2-0	PMDS6[2:0]	R/W	( 16-1 )

ADxPSEL7 PMD 7

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS7	-	-	-	-	PMDS7		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS7	R/W	PMD1TRG1 0: 1:
6-3	-	R	"0"
2-0	PMDS7[2:0]	R/W	( 16-1 )

ADxPSEL8 PMD 8

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS8	-	-	-	-	PMDS8		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS8	R/W	PMD1TRG2 0: 1:
6-3	-	R	"0"
2-0	PMDS8[2:0]	R/W	( 16-1 )

ADxPSEL9 PMD 9

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS9	-	-	-	-	PMDS9		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS9	R/W	PMD1TRG3 0: 1:
6-3	-	R	"0"
2-0	PMDS9[2:0]	R/W	( 16-1 )

ADxPSEL10 PMD 10

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS10	-	-	-	-	PMDS10		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS10	R/W	PMD1TRG4 0: 1:
6-3	-	R	"0"
2-0	PMDS10[2:0]	R/W	( 16-1 )

ADxPSEL11 PMD 11

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PENS11	-	-	-	-	PMDS11		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8	-	R	"0"
7	PENS11	R/W	PMD1TRG5 0: 1:
6-3	-	R	"0"
2-0	PMDS11[2:0]	R/W	( 16-1 )

## 16-1

<PMDS0[2:0]>~ <PMDS11[2:0]>	
000	0
001	1
010	2
011	3
100	4
101	5
110	Reserved
111	Reserved

16.4.24.2 ADxPINTS0 ~ 5(PMD 0 ~ 5)

ADxPINTS0 PMD

0

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

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	INTSELO[1:0]	R/W	00: 01:INTADxPDA 10:INTADxPDB 11:  0

ADxPINTS1 PMD

1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL1	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	INTSEL1[1:0]	R/W	00: 01:INTADxPDA 10:INTADxPDB 11:  1

ADxPINTS2 PMD

2

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL2	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	INTSEL2[1:0]	R/W	00: 01:INTADxPDA 10:INTADxPDB 11:  2

ADxPINTS3 PMD

3

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL3	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	INTSEL3[1:0]	R/W	00: 01:INTADxPDA 10:INTADxPDB 11:  3

ADxPINTS4 PMD

4

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL4	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	INTSEL4[1:0]	R/W	00: 01:INTADxPDA 10:INTADxPDB 11:  4

ADxPINTS5 PMD

5

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	INTSEL5	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	INTSEL5[1:0]	R/W	00: 01:INTADxPDA 10:INTADxPDB 11:  5

16.4.24.3 ADxPSET0 ~ 5(PMD) 0 ~ 5)

各 ADxPSETn (n=0~5 : プログラム番号) は、AD 変換入力端子の選択をする<AINSPnm [4:0]>とベクトルエンジンの相選択をする<UVWISnm[1:0]>および<ENSPnm>を 1 組とした 4 つのセットで構成されます。(m=0~3) (x = A,B : AD コンバータユニット)

ADxREGm	m=0	m=1	m=2	m=3
ADxPSETn				
n=0	<ENSP00> <UVWIS00> <AINSP00>	<ENSP01> <UVWIS01> <AINSP01>	<ENSP02> <UVWIS02> <AINSP02>	<ENSP03> <UVWIS03> <AINSP03>
n=1	<ENSP10> <UVWIS10> <AINSP10>	<ENSP11> <UVWIS11> <AINSP11>	<ENSP12> <UVWIS12> <AINSP12>	<ENSP13> <UVWIS13> <AINSP13>
n=2	<ENSP20> <UVWIS20> <AINSP20>	<ENSP21> <UVWIS21> <AINSP21>	<ENSP22> <UVWIS22> <AINSP22>	<ENSP23> <UVWIS23> <AINSP23>
n=3	<ENSP30> <UVWIS30> <AINSP30>	<ENSP31> <UVWIS31> <AINSP31>	<ENSP32> <UVWIS32> <AINSP32>	<ENSP33> <UVWIS33> <AINSP33>
n=4	<ENSP40> <UVWIS40> <AINSP40>	<ENSP41> <UVWIS41> <AINSP41>	<ENSP42> <UVWIS42> <AINSP42>	<ENSP43> <UVWIS43> <AINSP43>
n=5	<ENSP50> <UVWIS50> <AINSP50>	<ENSP51> <UVWIS51> <AINSP51>	<ENSP52> <UVWIS52> <AINSP52>	<ENSP53> <UVWIS53> <AINSP53>

16-2 AD

<AINSP00 [4:0]> ~ <AINSP53 [4:0]>	AD A	AD B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 ~ 1_1111	:Reserved	

ADxPSET0 PMD 0

	31	30	29	28	27	26	25	24
bit symbol	ENSP03	UVWIS03			AINSP03			
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP02	UVWIS02			AINSP02			
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP01	UVWIS01			AINSP01			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP00	UVWIS00			AINSP00			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSP03	R/W	ADxREG3 0: 1:
30-29	UVWIS03[1:0]	R/W	(
28-24	AINSP03[4:0]	R/W	AIN 16-2 AD
23	ENSP02	R/W	ADxREG2 0: 1:
22-21	UVWIS02[1:0]	R/W	(
20-16	AINSP02[4:0]	R/W	AIN 16-2 AD
15	ENSP01	R/W	ADxREG1 0: 1:
14-13	UVWIS01[1:0]	R/W	(
12-8	AINSP01[4:0]	R/W	AIN 16-2 AD
7	ENSP00	R/W	ADxREG0 0: 1:
6-5	UVWIS00[1:0]	R/W	(
4-0	AINSP00[4:0]	R/W	AIN 16-2 AD

00	
01	U
10	V
11	W

ADxPSET1 PMD 1

	31	30	29	28	27	26	25	24
bit symbol	ENSP13	UVWIS13		AINSP13				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP12	UVWIS12		AINSP12				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP11	UVWIS11		AINSP11				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP10	UVWIS10		AINSP10				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSP13	R/W	ADxREG3 0: 1:
30-29	UVWIS13[1:0]	R/W	(
28-24	AINSP13[4:0]	R/W	AIN 16-2 AD
23	ENSP12	R/W	ADxREG2 0: 1:
22-21	UVWIS12[1:0]	R/W	(
20-16	AINSP12[4:0]	R/W	AIN 16-2 AD
15	ENSP11	R/W	ADxREG1 0: 1:
14-13	UVWIS11[1:0]	R/W	(
12-8	AINSP11[4:0]	R/W	AIN 16-2 AD
7	ENSP10	R/W	ADxREG0 0: 1:
6-5	UVWIS10[1:0]	R/W	(
4-0	AINSP10[4:0]	R/W	AIN 16-2 AD

00	
01	U
10	V
11	W

ADxPSET2 PMD

2

	31	30	29	28	27	26	25	24
bit symbol	ENSP23	UVWIS23			AINSP23			
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP22	UVWIS22			AINSP22			
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP21	UVWIS21			AINSP21			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP20	UVWIS20			AINSP20			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSP23	R/W	ADxREG3 0: 1:
30-29	UVWIS23[1:0]	R/W	(
28-24	AINSP23[4:0]	R/W	AIN 16-2 AD
23	ENSP22	R/W	ADxREG2 0: 1:
22-21	UVWIS22[1:0]	R/W	(
20-16	AINSP22[4:0]	R/W	AIN 16-2 AD
15	ENSP21	R/W	ADxREG1 0: 1:
14-13	UVWIS21[1:0]	R/W	(
12-8	AINSP21[4:0]	R/W	AIN 16-2 AD
7	ENSP20	R/W	ADxREG0 0: 1:
6-5	UVWIS20[1:0]	R/W	(
4-0	AINSP20[4:0]	R/W	AIN 16-2 AD

00	
01	U
10	V
11	W

ADxPSET3 PMD 3

	31	30	29	28	27	26	25	24
bit symbol	ENSP33	UVWIS33		AINSP33				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP32	UVWIS32		AINSP32				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP31	UVWIS31		AINSP31				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP30	UVWIS30		AINSP30				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSP33	R/W	ADxREG3 0: 1:
30-29	UVWIS33[1:0]	R/W	(
28-24	AINSP33[4:0]	R/W	AIN 16-2 AD
23	ENSP32	R/W	ADxREG2 0: 1:
22-21	UVWIS32[1:0]	R/W	(
20-16	AINSP32[4:0]	R/W	AIN 16-2 AD
15	ENSP31	R/W	ADxREG1 0: 1:
14-13	UVWIS31[1:0]	R/W	(
12-8	AINSP31[4:0]	R/W	AIN 16-2 AD
7	ENSP30	R/W	ADxREG0 0: 1:
6-5	UVWIS30[1:0]	R/W	(
4-0	AINSP30[4:0]	R/W	AIN 16-2 AD

00	
01	U
10	V
11	W

ADxPSET4 PMD

4

	31	30	29	28	27	26	25	24
bit symbol	ENSP43	UVWIS43			AINSP43			
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP42	UVWIS42			AINSP42			
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP41	UVWIS41			AINSP41			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP40	UVWIS40			AINSP40			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSP43	R/W	ADxEG3 0: 1:
30-29	UVWIS43[1:0]	R/W	(
28-24	AINSP43[4:0]	R/W	AIN 16-2 AD
23	ENSP42	R/W	ADxREG2 0: 1:
22-21	UVWIS42[1:0]	R/W	(
20-16	AINSP42[4:0]	R/W	AIN 16-2 AD
15	ENSP41	R/W	ADxREG1 0: 1:
14-13	UVWIS41[1:0]	R/W	(
12-8	AINSP41[4:0]	R/W	AIN 16-2 AD
7	ENSP40	R/W	ADxREG0 0: 1:
6-5	UVWIS40[1:0]	R/W	(
4-0	AINSP40[4:0]	R/W	AIN 16-2 AD

00	
01	U
10	V
11	W

ADxPSET5 PMD

5

	31	30	29	28	27	26	25	24
bit symbol	ENSP53	UVWIS53		AINSP53				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSP52	UVWIS52		AINSP52				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSP51	UVWIS51		AINSP51				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSP50	UVWIS50		AINSP50				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSP53	R/W	ADxREG3 0: 1:
30-29	UVWIS53[1:0]	R/W	(
28-24	AINSP53[4:0]	R/W	AIN 16-2 AD
23	ENSP52	R/W	ADxREG2 0: 1:
22-21	UVWIS52[1:0]	R/W	(
20-16	AINSP52[4:0]	R/W	AIN 16-2 AD
15	ENSP51	R/W	ADxREG1 0: 1:
14-13	UVWIS51[1:0]	R/W	(
12-8	AINSP51[4:0]	R/W	AIN 16-2 AD
7	ENSP50	R/W	ADxREG0 0: 1:
6-5	UVWIS50[1:0]	R/W	(
4-0	AINSP50[4:0]	R/W	AIN 16-2 AD

00	
01	U
10	V
11	W

### 16.4.25 ADxTSET03 / ADxTSET47 / ADxTSET811( )

本 AD コンバータは、タイマ x のコンペア割り込みをトリガ信号にして、AD 変換を開始する事ができます。

タイマトリガ用プログラムレジスタは 12 個の設定レジスタから構成されます。<ENSTm>を 1 にセットすると ADxTSETm をイネーブルにします。<AINSTm [4:0]>は AIN を選択します。設定レジスタの番号は変換結果レジスタ番号(m=0~11)にそれぞれ対応しています。ADxTSET0 は変換結果レジスタ 0 に対応し、ADxTSET11(変換値レジスタ 11)まで全部で 12 個あります。

タイマトリガによる AD 変換が終了すると割り込み(INTADxTMR)を発生します。(x=A,B : AD コンバータユニット)

16-3 AD

<AINST0 [4:0]> ~ <AINST11 [4:0]>	AD	AD
	A	B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 ~ 1_1111	:Reserved	

ADxTSET03

03

	31	30	29	28	27	26	25	24
bit symbol	ENST3	-	-	AINST3				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST2	-	-	AINST2				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST1	-	-	AINST1				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST0	-	-	AINST0				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENST3	R/W	ADxREG3 0:disable 1:enable
30-29	-	R	"0"
28-24	AINST3[4:0]	R/W	AIN 16-3 AD
23	ENST2	R/W	ADxREG2 0:disable 1:enable
22-21	-	R	"0"
20-16	AINST2[4:0]	R/W	AIN 16-3 AD
15	ENST1	R/W	ADxREG1 0:disable 1:enable
14-13	-	R	"0"
12-8	AINST1[4:0]	R/W	AIN 16-3 AD
7	ENST0	R/W	ADxREG0 0:disable 1:enable
6-5	-	R	"0"
4-0	AINST0[4:0]	R/W	AIN 16-3 AD

ADxTSET47

47

	31	30	29	28	27	26	25	24
bit symbol	ENST7	-	-	AINST7				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST6	-	-	AINST6				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST5	-	-	AINST5				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST4	-	-	AINST4				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENST7	R/W	ADxREG7 0:disable 1:enable
30-29	-	R	"0"
28-24	AINST7[4:0]	R/W	AIN 16-3 AD
23	ENST6	R/W	ADxREG6 0:disable 1:enable
22-21	-	R	"0"
20-16	AINST6[4:0]	R/W	AIN 16-3 AD
15	ENST5	R/W	ADxREG5 0:disable 1:enable
14-13	-	R	"0"
12-8	AINST5[4:0]	R/W	AIN 16-3 AD
7	ENST4	R/W	ADxREG4 0:disable 1:enable
6-5	-	R	"0"
4-0	AINST4[4:0]	R/W	AIN 16-3 AD

ADxTSET811 811

	31	30	29	28	27	26	25	24
bit symbol	ENST11	-	-	AINST11				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENST10	-	-	AINST10				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENST9	-	-	AINST9				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENST8	-	-	AINST8				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENST11	R/W	ADxREG11 0:disable 1:enable
30-29	-	R	"0"
28-24	AINST11[4:0]	R/W	AIN 16-3 AD
23	ENST10	R/W	ADxREG10 0:disable 1:enable
22-21	-	R	"0"
20-16	AINST10[4:0]	R/W	AIN 16-3 AD
15	ENST9	R/W	ADxREG9 0:disable 1:enable
14-13	-	R	"0"
12-8	AINST9[4:0]	R/W	AIN 16-3 AD
7	ENST8	R/W	ADxREG8 0:disable 1:enable
6-5	-	R	"0"
4-0	AINST8[4:0]	R/W	AIN 16-3 AD

## 16.4.26 ADxSSET03 / ADxSSET47 / ADxSSET811( )

本 AD 変換はソフトウェアによってAD 変換を開始する事ができます。ソフトウェアトリガ用のプログラムレジスタは 12 個の設定レジスタから構成されます。

<ENSSm>を 1 にセットすると ADxSSETm をイネーブルにします。<AINSSm 4:0>は AIN を選択します。プログラム設定レジスタの番号(m=0 ~ 11)は変換結果レジスタ番号にそれぞれ対応しており、レジスタは全部で 12 個あります。

ソフトウェアトリガによる AD 変換が終了すると割り込み(INTADxSFT)を発生します。(x = A,B : AD コンバータユニット)

16-4 AD

<AINSS0 [4:0]> ~ <AINSS11 [4:0]>	AD A	AD B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 ~ 1_1111	:Reserved	

ADxSSET03

03

	31	30	29	28	27	26	25	24
bit symbol	ENSS3	-	-	AINSS3				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS2	-	-	AINSS2				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS1	-	-	AINSS1				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS0	-	-	AINSS0				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSS3	R/W	ADxREG3 0:disable 1:enable
30-29	-	R	"0"
28-24	AINSS3[4:0]	R/W	AIN 16-4 AD
23	ENSS2	R/W	ADxREG2 0:disable 1:enable
22-21	-	R	"0"
20-16	AINSS2[4:0]	R/W	AIN 16-4 AD
15	ENSS1	R/W	ADxREG1 0:disable 1:enable
14-13	-	R	"0"
12-8	AINSS1[4:0]	R/W	AIN 16-4 AD
7	ENSS0	R/W	ADxREG0 0:disable 1:enable
6-5	-	R	"0"
4-0	AINSS0[4:0]	R/W	AIN 16-4 AD

ADxSSET47

47

	31	30	29	28	27	26	25	24
bit symbol	ENSS7	-	-	AINSS7				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS6	-	-	AINSS6				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS5	-	-	AINSS5				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS4	-	-	AINSS4				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSS7	R/W	ADxREG7 0:disable 1:enable
30-29	-	R	"0"
28-24	AINSS7[4:0]	R/W	AIN 16-4 AD
23	ENSS6	R/W	ADxREG6 0:disable 1:enable
22-21	-	R	"0"
20-16	AINSS6[4:0]	R/W	AIN 16-4 AD
15	ENSS5	R/W	ADxREG5 0:disable 1:enable
14-13	-	R	"0"
12-8	AINSS5[4:0]	R/W	AIN 16-4 AD
7	ENSS4	R/W	ADxREG4 0:disable 1:enable
6-5	-	R	"0"
4-0	AINSS4[4:0]	R/W	AIN 16-4 AD

ADxSSET811

811

	31	30	29	28	27	26	25	24
bit symbol	ENSS11	-	-	AINSS11				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSS10	-	-	AINSS10				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSS9	-	-	AINSS9				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSS8	-	-	AINSS8				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSS11	R/W	ADxREG11 0:disable 1:enable
30-29	-	R	"0"
28-24	AINSS11[4:0]	R/W	AIN 16-4 AD
23	ENSS10	R/W	ADxREG10 0:disable 1:enable
22-21	-	R	"0"
20-16	AINSS10[4:0]	R/W	AIN 16-4 AD
15	ENSS9	R/W	ADxREG9 0:disable 1:enable
14-13	-	R	"0"
12-8	AINSS9[4:0]	R/W	AIN 16-4 AD
7	ENSS8	R/W	ADxREG8 0:disable 1:enable
6-5	-	R	"0"
4-0	AINSS8[4:0]	R/W	AIN 16-4 AD

## 16.4.27 ADxASET03 / ADxASET47 / ADxASET811( )

本 AD コンバータは常時変換状態に設定する事が出来ます。常時変換用のプログラムレジスタは 12 個の設定レジスタから構成されます。

設定レジスタの番号は変換結果レジスタ番号(m=0~11)にそれぞれ対応しています。常時変換用プログラムレジスタの<ENSA<sub>m</sub>>を 1 にセットすると ADxPSET<sub>m</sub> をイネーブルにします。<AINSA<sub>m</sub>[4:0]>は AIN を選択します。

レジスタは全部で 12 個あります。(x = A,B : AD コンバータユニット)

16-5 AD

<AINSA0[4:0]> ~ <AINSA11[4:0]>	AD	AD
	A	B
0_0000	:AINA0	:AINB0
0_0001	:AINA1	:AINB1
0_0010	:AINA2	:AINB2
0_0011	:AINA3	:AINB3
0_0100	:AINA4	:AINB4
0_0101	:AINA5	:AINB5
0_0110	:AINA6	:AINB6
0_0111	:AINA7	:AINB7
0_1000	:AINA8	:AINB8
0_1001	:AINA9	:AINB9
0_1010	:AINA10	:AINB10
0_1011	:AINA11	:Reserved
0_1100 ~ 1_1111	:Reserved	

ADxASET03 03

	31	30	29	28	27	26	25	24
bit symbol	ENSA3	-	-	AINSA3				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA2	-	-	AINSA2				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA1	-	-	AINSA1				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA0	-	-	AINSA0				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSA3	R/W	ADxREG3 0:disable 1:enable
30-29	-	R	"0"
28-24	AINSA3[4:0]	R/W	AIN 16-5 AD
23	ENSA2	R/W	ADxREG2 0:disable 1:enable
22-21	-	R	"0"
20-16	AINSA2[4:0]	R/W	AIN 16-5 AD
15	ENSA1	R/W	ADxREG1 0:disable 1:enable
14-13	-	R	"0"
12-8	AINSA1[4:0]	R/W	AIN 16-5 AD
7	ENSA0	R/W	ADxREG0 0:disable 1:enable
6-5	-	R	"0"
4-0	AINSA0[4:0]	R/W	AIN 16-5 AD

ADxASET47

47

	31	30	29	28	27	26	25	24
bit symbol	ENSA7	-	-	AINSA7				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA6	-	-	AINSA6				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA5	-	-	AINSA5				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA4	-	-	AINSA4				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSA7	R/W	ADxREG7 0:disable 1:enable
30-29	-	R	"0"
28-24	AINSA7[4:0]	R/W	AIN 16-5 AD
23	ENSA6	R/W	ADxREG6 0:disable 1:enable
22-21	-	R	"0"
20-16	AINSA6[4:0]	R/W	AIN 16-5 AD
15	ENSA5	R/W	ADxREG5 0:disable 1:enable
14-13	-	R	"0"
12-8	AINSA5[4:0]	R/W	AIN 16-5 AD
7	ENSA4	R/W	ADxREG4 0:disable 1:enable
6-5	-	R	"0"
4-0	AINSA4[4:0]	R/W	AIN 16-5 AD

ADxASET811

811

	31	30	29	28	27	26	25	24
bit symbol	ENSA11	-	-	AINSA11				
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	ENSA10	-	-	AINSA10				
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	ENSA9	-	-	AINSA9				
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ENSA8	-	-	AINSA8				
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31	ENSA11	R/W	ADxREG11 0:disable 1:enable
30-29	-	R	"0"
28-24	AINSA11[4:0]	R/W	AIN 16-5 AD
23	ENSA10	R/W	ADxREG10 0:disable 1:enable
22-21	-	R	"0"
20-16	AINSA10[4:0]	R/W	AIN 16-5 AD
15	ENSA9	R/W	ADxREG9 0:disable 1:enable
14-13	-	R	"0"
12-8	AINSA9[4:0]	R/W	AIN 16-5 AD
7	ENSA8	R/W	ADxREG8 0:disable 1:enable
6-5	-	R	"0"
4-0	AINSA8[4:0]	R/W	AIN 16-5 AD

## 16.5

### 16.5.1

アナログ基準電圧は AD コンバータ ユニット A の VREFHA, VREFLA 端子、AD コンバータ ユニット B の VREFHB, VREFLB 端子にそれぞれ High、Low のレベルを入力します。ADxMOD3<RCUT> に"1"をライトすることによって、VREFHx と VREFLx 間のスイッチオン状態をオフ状態に切り換えることができます。

### 16.5.2 AD

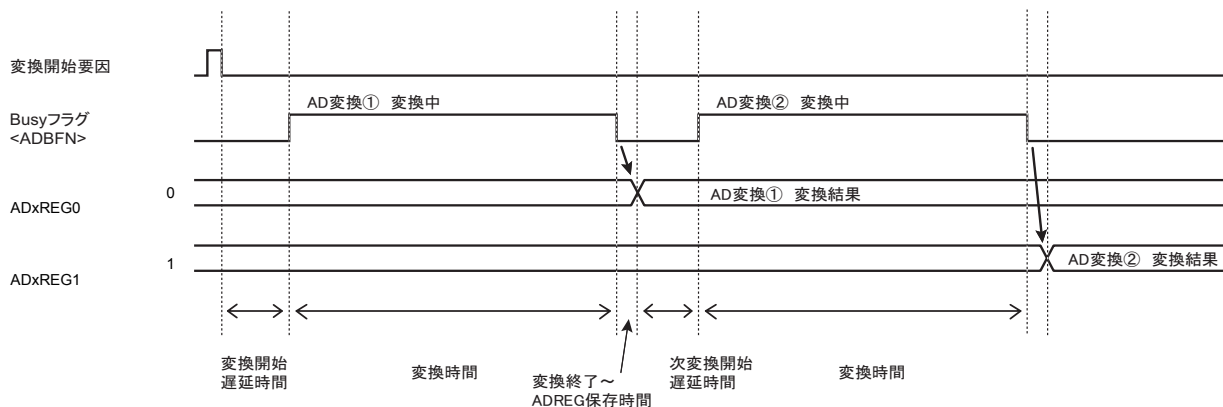
AD コンバータはソフトウェアによる起動（ソフトウェアトリガ）または PMD/タイマからのトリガ信号により任意の AD が変換開始します。

これらの起動要因には優先順位があり、

PMD 0 > > PMD 5 > >

上位の起動要因(トリガ)が発生した時は、実行中のAD 変換を中止して直ちに上位のプログラムを実行します。PMDトリガによるAD変換実行中に上位要因が発生時は、実行中のAD 変換終了後に上位のプログラムに移行します。

起動要因発生から実際の AD 変換実行までには遅れがあります。トリガ要因による AD 変換タイミングと遅延時間を以下に示します。



16-3 AD

16-6 AD

	( 1)			~ADREG	( 2)	
	Min	Max			Min	Max
PMD [1/SCLK]	5	8	120	2	3	5
TMRB [1/SCLK]		16				13
[1/SCLK]	6	17				
[1/SCLK]						

1)

2)

2

16.5.3 AD

本 AD コンバータには AD 監視機能があり、監視機能有効時に比較条件と一致した場合、割り込みが発生します。

AD<sub>x</sub>CMPCR0<CMP0EN>または AD<sub>x</sub>CMPCR1<CMP1EN>を"1"に設定すると AD 監視機能が有効となり、<REGS0>/<REGS1>で指定された変換結果格納レジスタの内容が比較レジスタの値以上または以下になると AD 監視機能割り込みが発生します (<ADBIG0>/<ADBIG1>で以上、以下を指定)

比較動作は監視機能が設定された変換結果格納レジスタに対し、変換結果が格納されるごとにおこなわれ、条件が成立すると割り込みが発生します。

1) AD

AD

<ADR0RF> ~ <ADR11RF>

2) AD

Over Run

<OVR0> ~ <OVR11> "1"

AD

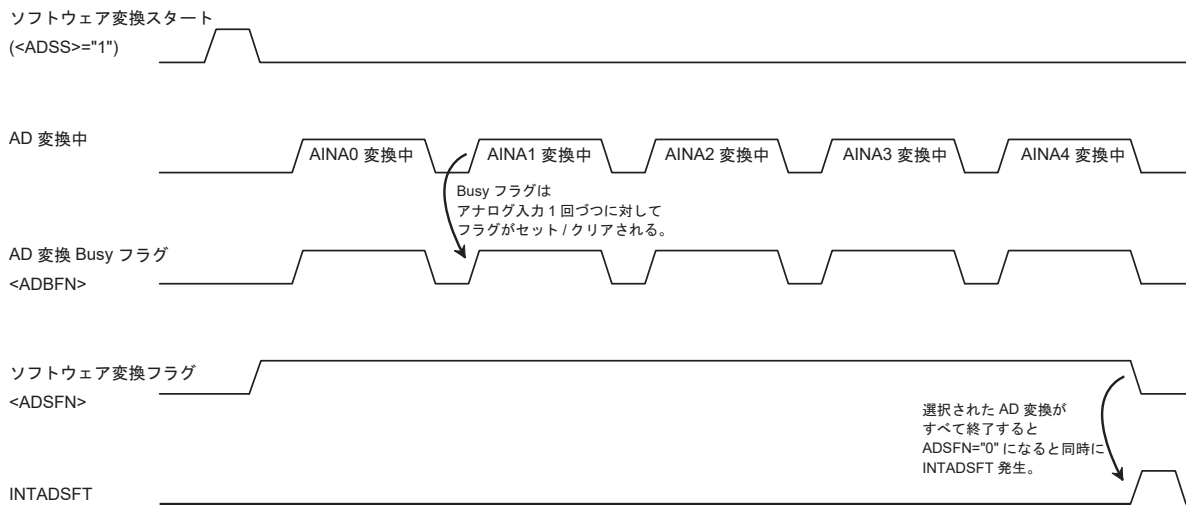
## 16.6 AD

ソフトウェア AD 変換、常時 AD 変換、PMD トリガ/タイマトリガ受け付け時のタイミングチャートを以下に示します。

### 16.6.1 AD

ソフトウェア AD 変換では、ADxSSET03, ADxSSET47, ADxSSET811 でプログラムされた AD 変換が終了時に割り込みが発生します(図 16-4)。

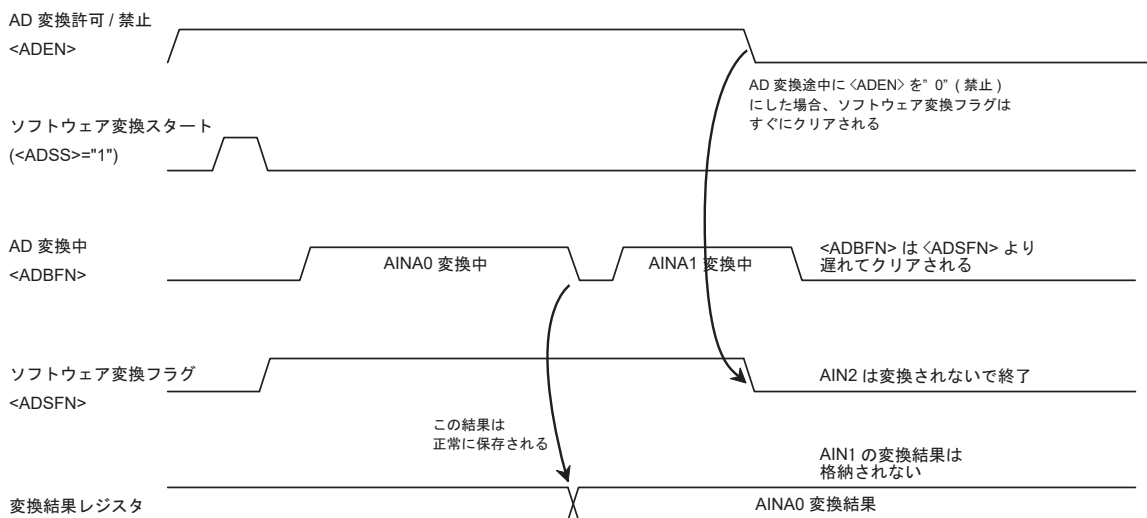
ソフトウェア AD 変換中に ADxMOD1<ADEN>="0"とした場合、実行中の AD 変換を終了します。このとき、変換途中の結果はレジスタには格納されません(図 16-5)。



16-4 AD

[ 設定条件 ]

ソフトウェアトリガ設定 : AINA0, AINA1, AINA2



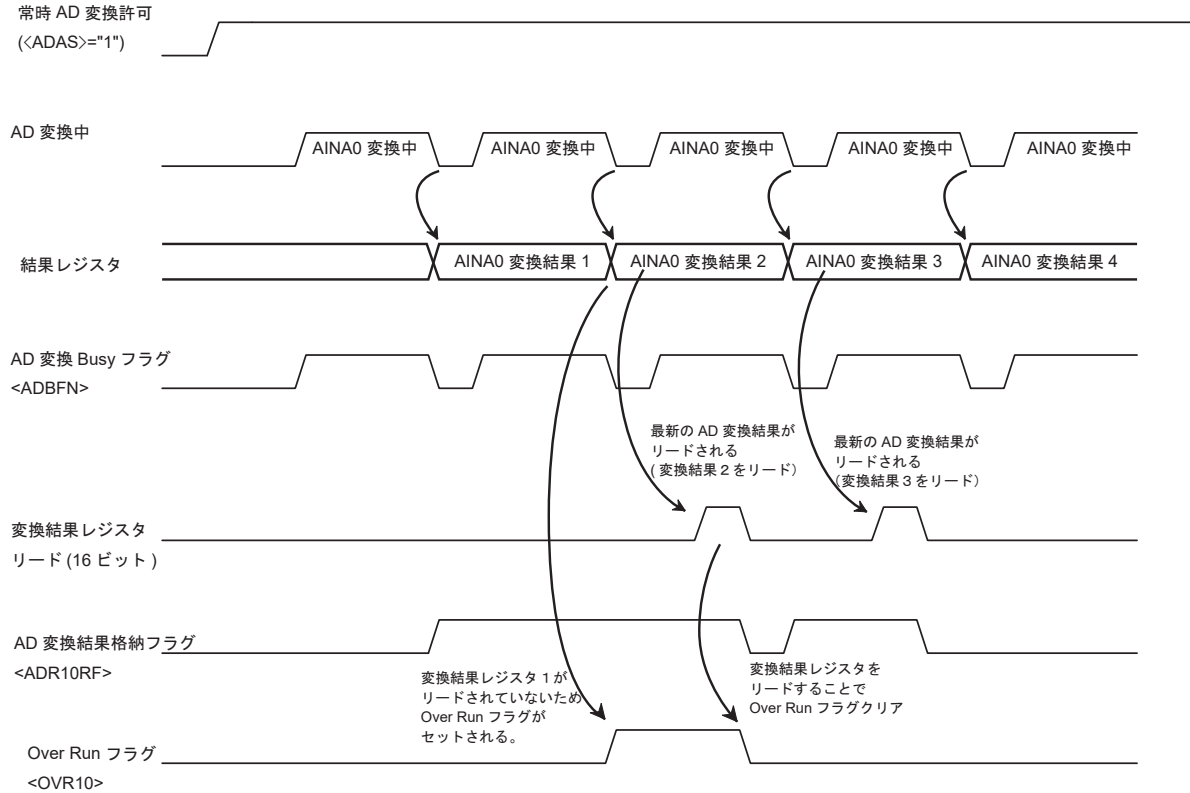
16-5 AD <ADEN>="0"

## 16.6.2 AD

常時 AD 変換では、変換結果を読み出す前に次の AD 変換が終了した場合、Over Run フラグが"1"にセットされます。このとき、変換結果レジスタは後続のデータによって上書きされます。OverRun フラグは変換結果を読み出すことにより"0"にクリアされます (図 16-6)。

[設定条件]

常時変換設定: AINA0

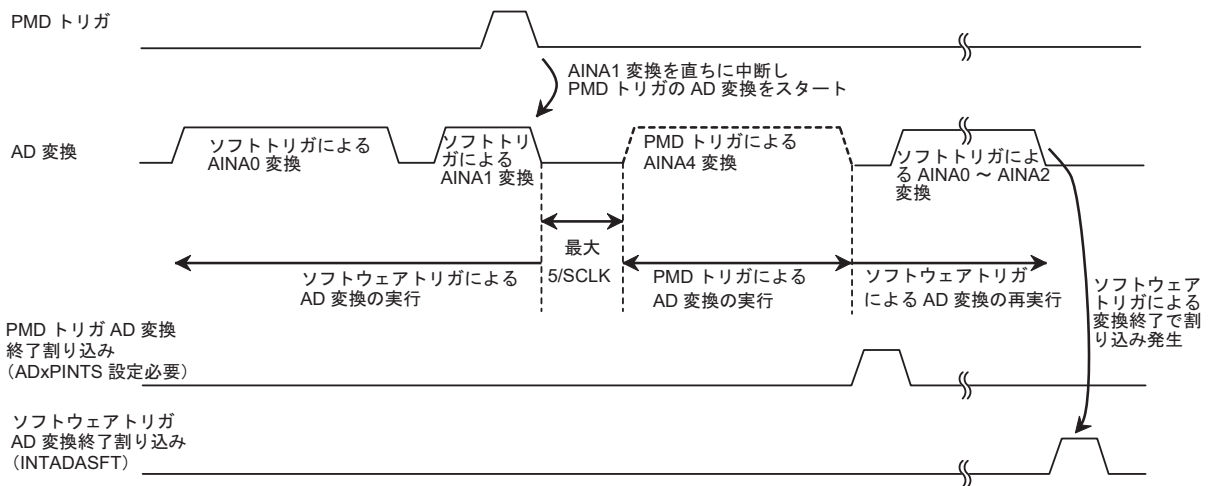


16-6 AD

16.6.3 AD

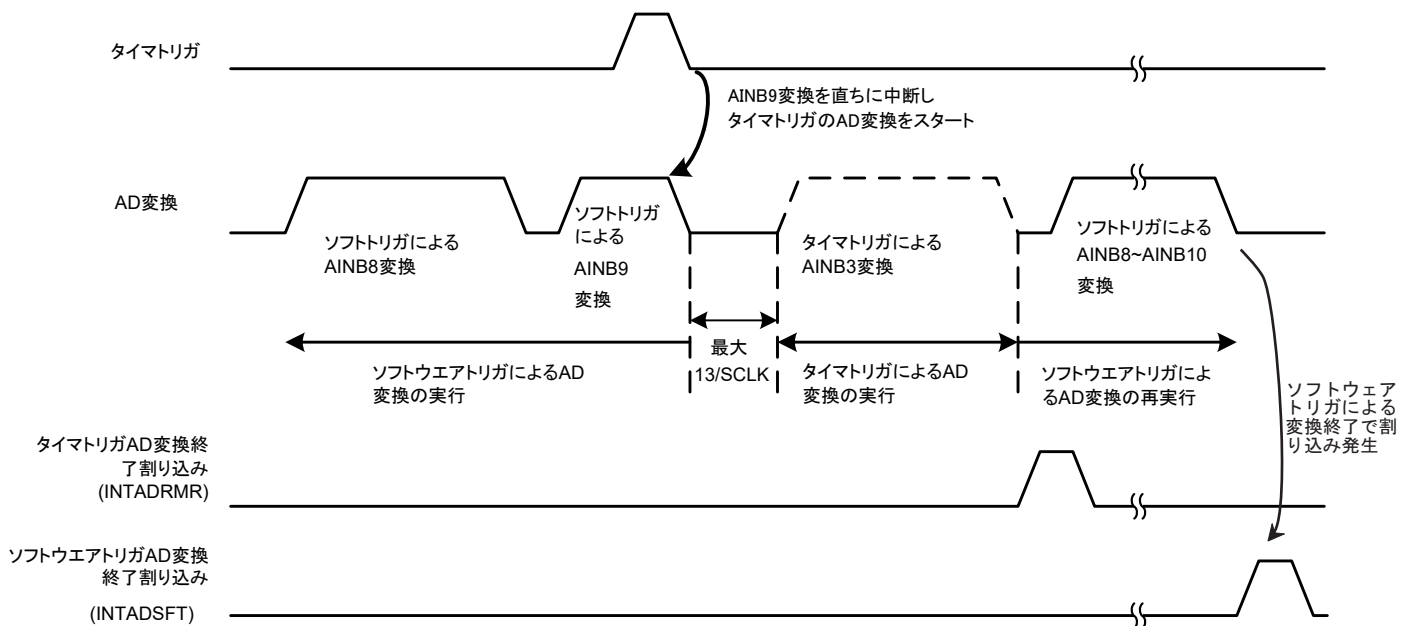
ソフトウェアAD変換実行中にPMDトリガが発生した場合、実行中のソフトウェアAD変換は直ちに中断してPMDトリガによるAD変換を開始します(図16-7)。PMDトリガによるAD変換が終了後、ソフトウェアAD変換は設定されたプログラムの最初からAD変換を開始します。タイマトリガが発生した場合も同様です(図16-8)。

【設定条件】  
ソフトウェアトリガ設定: AINA0, AINA1, AINA2  
PMDトリガ設定: AINA4



16-7 AD PMD

【条件設定】  
ソフトウェアトリガ設定: AINB8, AINB9, AINB10  
タイマトリガ設定: AINB3



16-8 AD

ヒ

INTTB51

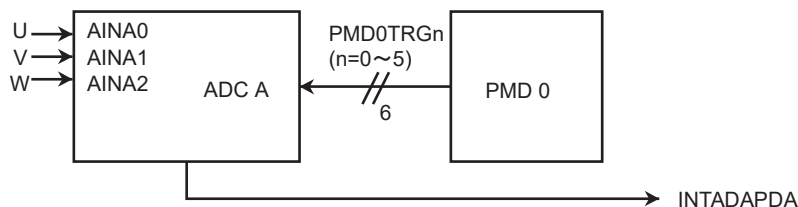
TB5IM<TBIM1>=1



16.7

16.7.1 PMD (3 ) AD × 1

PMD0 を 3 シャントで、AD コンバータ（ユニット A）を 1 つ使用した場合の回路図を以下に示します。



この場合の AD コンバータの設定例を以下に示します。

AD	A					
Program	0	1	2	3	4	5
reg0	U	V	W	V	W	U
reg1	V	W	U	U	V	W
INT	A	A	A	A	A	A

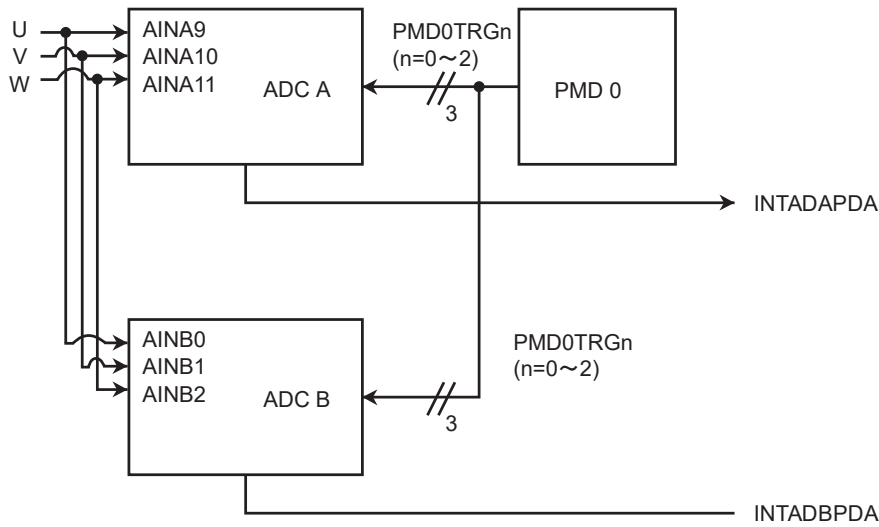
6 本のトリガ入力 PMD0TRG0 ~ 5 に対し、ADAPSEL0 ~ 5 でそれぞれ 0 ~ 5 のプログラム番号を割り付けます。

表中の reg0,1 は ADAPSETn[7:0]および ADAPSETn[15:8]を表します(n : プログラム番号)。表中の「U」、「V」、「W」はモータの位相を表しており、それぞれの位相が得られる AIN 入力を選択し封。

トリガが入力されると reg0、reg1 の順に AD 変換が開始され、それぞれの変換結果が変換結果格納レジスタへ保存され、INTADAPDA 割り込みが出力されます。

### 16.7.2 PMD A(3 ) AD × 2

PMD を 3 シャントで 1 つ、AD コンバータを 2 つ使用した場合の回路図を以下に示します。



この場合の AD コンバータの設定例を以下に示します。

AD	A		
Program	0	1	2
reg0	U	V	W
INT	A	A	A

AD	B		
Program	0	1	2
reg0	V	W	U
INT	A	A	A

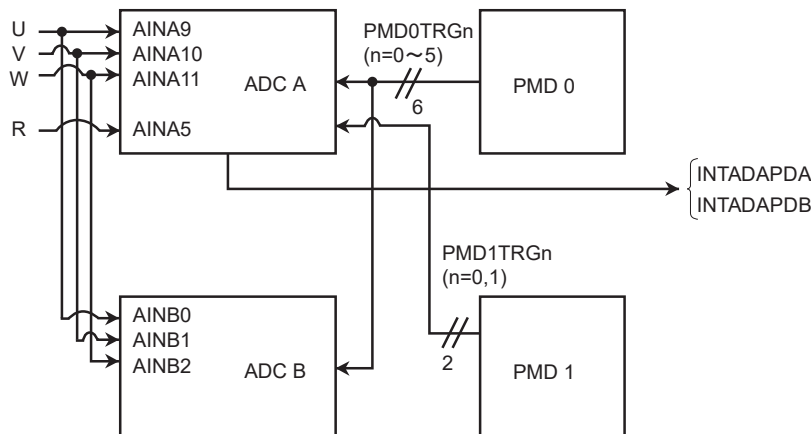
AD コンバータ ユニット A、AD コンバータ ユニット B に入力される 3 本のトリガ入力に、ADAPSEL0~2 および ADBPSEL0~2 でそれぞれ 0~2 のプログラム番号を割り付けます。

表中の reg0 は ADxPSETn[7:0] を表します(x: ADC ユニット, n: プログラム番号)。表中の「U」、 「V」、 「W」はモータの位相を表しており、それぞれの位相が得られる AIN 入力を選択します。

トリガが入力されると AD コンバータ ユニット A、AD コンバータ ユニット B が同時に AD 変換を開始し、それぞれの変換結果が ADxREG0 レジスタ 0 に保存されます。この時、AD コンバータ ユニット A、AD コンバータ ユニット B からそれぞれ割り込み要求 INTADAPDA と INTADBPDA が出力されます。

16.7.3 PMD 0(3 ) PMD 1(1 ) AD × 2

PMD を 3 シャント、1 シャントで 1 つずつ、AD コンバータを 2 つ使用した場合の回路図を以下に示します。



この場合の AD コンバータの設定例を以下に示します。

AD A

	PMD0	PMD0	PMD0	PMD1	PMD1
	0,3	1,4	2,5	6	7
Program	0	1	2	3	4
reg0	U	V	W	-	-
reg1	-	-	-	R	-
reg2	-	-	-	-	R
INT	A	A	A	-	B

AD B

	PMD0	PMD0	PMD0
	0,3	1,4	2,5
Program	0	1	2
reg0	V	W	U
INT	-	-	-

AD コンバータ ユニット A の設定は、PMD0、PMD1 から出力される合計 8 本のトリガ信号に対し、PMD0 からの 6 本を 0~2 のプログラム番号に、PMD1 からの 2 本を 3、4 のプログラム番号に割り付けます。

AD コンバータ ユニット B の設定は、PMD0 から出力される合計 6 本のトリガ信号に対し、0~2 のプログラム番号に割り付けます。

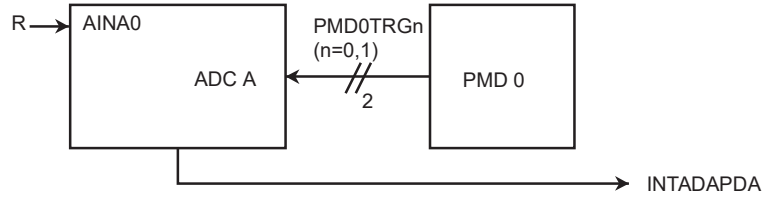
表中の reg0、1、2 は ADxPSETn[7:0]、ADxPSETn[15:8]、ADxPSETn[23:16]を表します。(x:ADC ユニット、n:プログラム番号)表中の「U」、「V」、「W」はモータの位相を表しており、それぞれの位相が得られる AIN 入力を選択します。「R」は抵抗を表しており、接続されている AIN を設定します。

トリガが入力されると AD コンバータ ユニット A、または AD コンバータ ユニット B の AD 変換が開始し、それぞれの変換結果が変換結果格納レジスタ 0 に保存されます。

AD コンバータ ユニット A に関しては PMD 0 からのトリガは INTADAPDA 割り込みが、PMD 1 からのトリガは INTADAPDB 割り込みが出力されます。AD コンバータ ユニット B はここでは割り込みを出力しない設定にしています。

### 16.7.4 PMD (1 ) AD × 1

PMD を 1 シヤントで 1 つ、AD コンバータ を 1 つ使用した場合の回路図を以下に示します。



この場合の AD コンバータの設定例を以下に示します。

AD	A	
	PMD0	PMD0
	0	1
Program	0	1
reg0	R	-
reg1	-	R
INT	-	A

PMD0 から出力される 2 本のトリガ信号に対し、それぞれプログラム番号を割り付けています。

表中の reg0,1 は ADAPSETn[7:0]、ADAPSETn[15:8]を表します(n : プログラム番号)。表中の「R」は抵抗を表しており、接続されている AIN を設定します。

トリガが入力されると AD コンバータ ユニット A の AD 変換が開始し、変換結果が変換結果格納レジスタ 0 と 1 に保存されます。プログラム 0、1 の順で変換が実行され、終了すると INTADAPDA 割り込みを発生します。

## 16.8 AD

AD 変換精度への影響を避けるために AD 動作(変換)中は出力ポートとして使用しているポート H/J/K の出力データを書き換えないようにしてください。

電源電圧の変動や周囲のノイズの影響によって AD 変換結果がばらつくことがあります。

また、AD 変換中に AD 入力が兼用となっている端子への入力及び出力の変化、出力ポートに設定している他の端子の出力電流が変動すると AD 変換精度が低下することがあります。

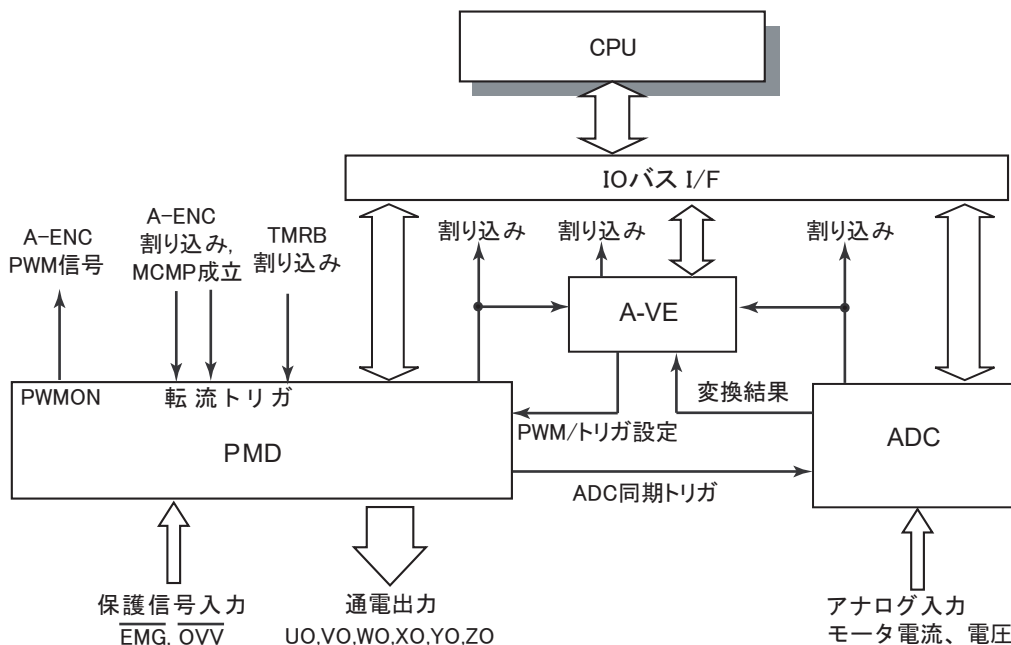
プログラムで複数回の変換結果の平均値をとるなどの対策をして下さい。



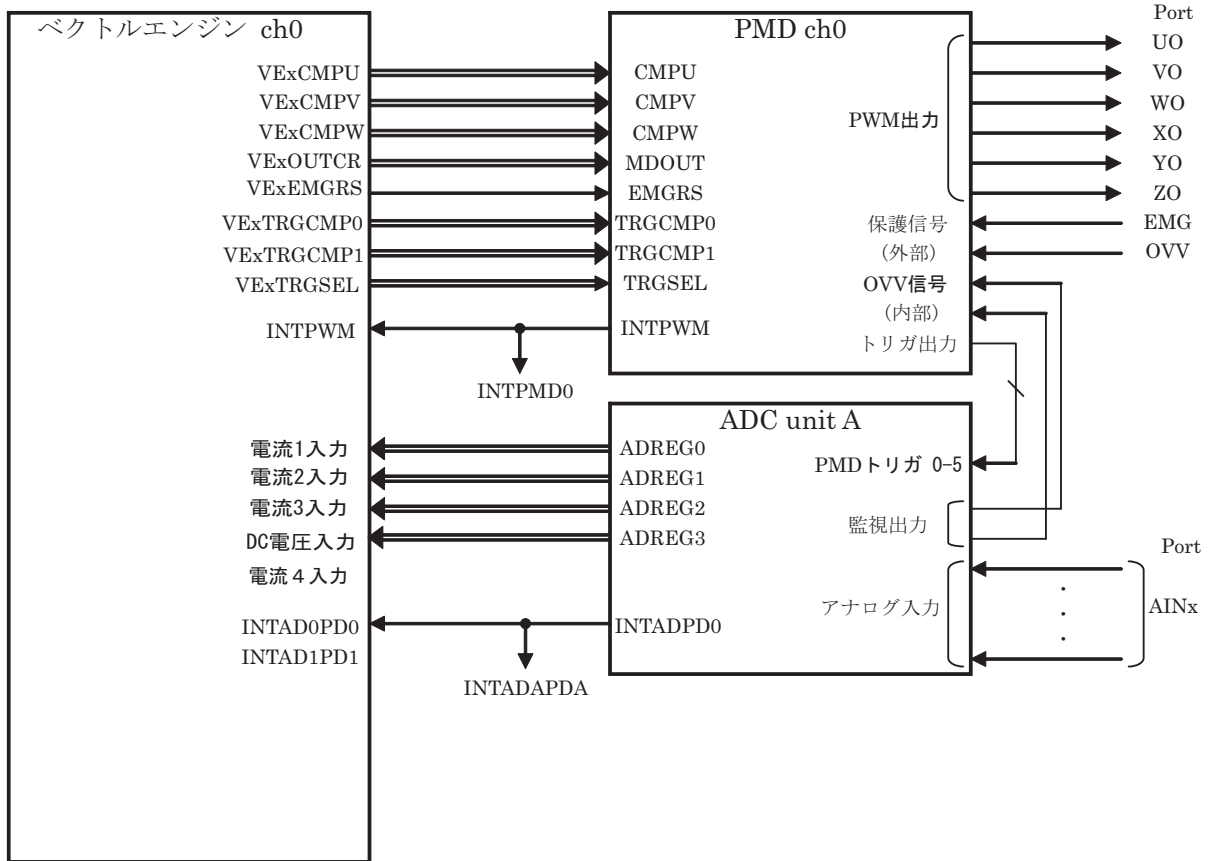
17

(PMD : Programmable Motor Driver)

本製品の PMD はベクトルエンジン(A-VE)やアナログ/デジタルコンバータ(ADC)と連携動作してベクトル制御などの 3 相モータ制御を実現します。パルス幅変調回路、通電制御および同期トリガ生成回路は VE から指令で動作可能で、同期トリガ生成回路は ADC に変換開始指令ができます。



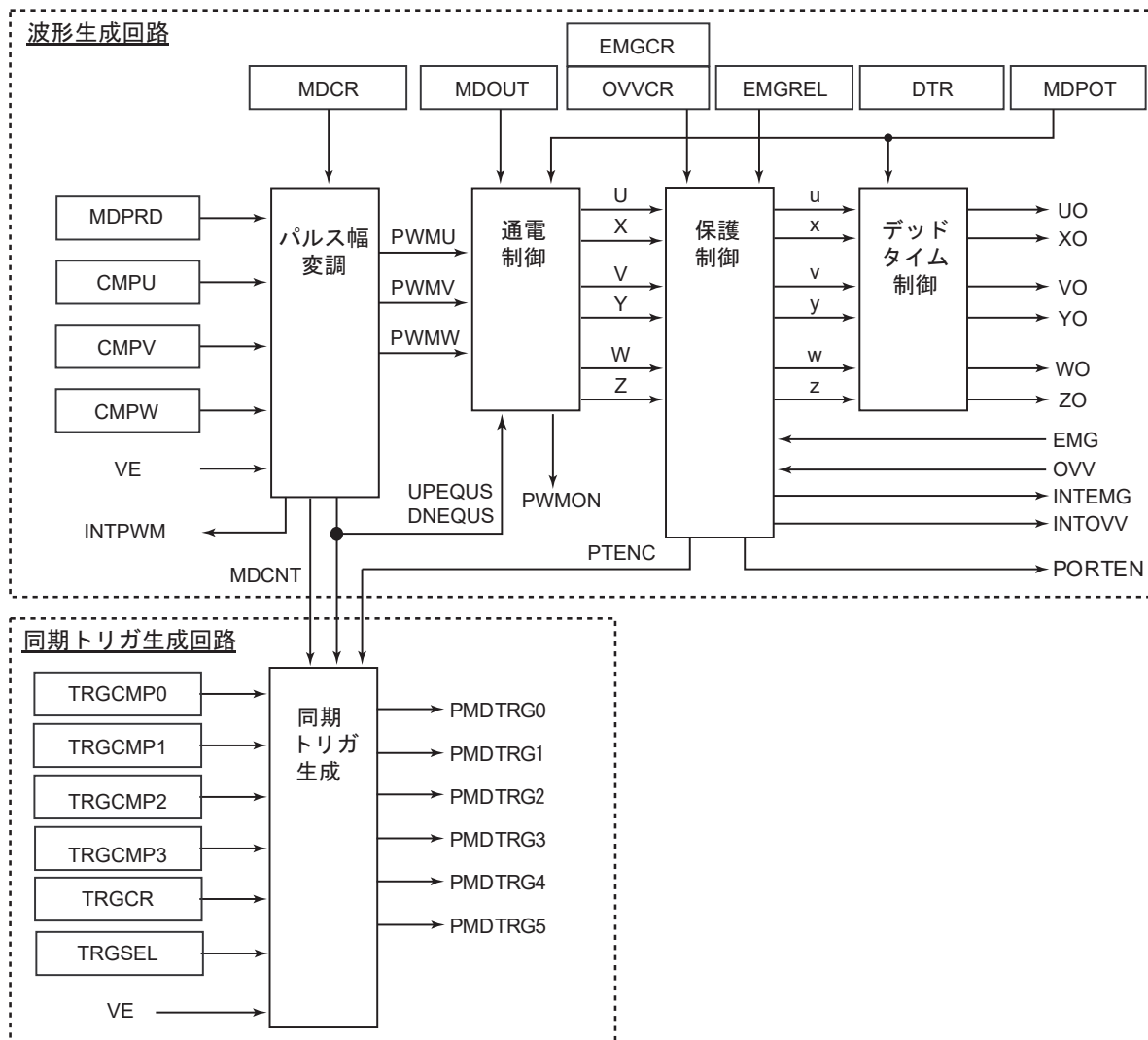
17-1



17-2

A/D

17.1 PMD



17-3 PMD

PMD(プログラマブルモータドライバ)回路は波形生成回路と同期トリガ生成回路の2ブロックから成り、波形生成回路はパルス幅変調回路、通電制御回路、保護制御回路、デッドタイム制御回路で構成されています。

- ・ パルス幅変調回路は、PWM キャリアが共通で3相の独立した PWM 波形を生成します。
- ・ 通電制御回路は U、V、W 相の各上下相の出力パターンを決定します。
- ・ 保護回路では EMG 入力、OVV 入力による緊急出力停止を行ないます。
- ・ デッドタイム制御回路では上下相の切り替え時の短絡を防止します。
- ・ 同期トリガ生成回路では ADC への同期トリガ信号を生成します。

## 17.2 PMD

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)	
PMD	PMDxMDEN	0x0000	
	PMDxPORTMD	0x0004	
PMD	PMDxMDCR	0x0008	
PWM	PMDxCNTSTA	0x000C	
PWM	PMDxMDCNT	0x0010	
PWM	PMDxMDPRD	0x0014	
PWM	U	PMDxCMPU	0x0018
PWM	V	PMDxCMPV	0x001C
PWM	W	PMDxCMPW	0x0020
		PMDxMODESEL	0x0024
PMD		PMDxMDOUT	0x0028
PMD		PMDxMDPOT	0x002C
EMG		PMDxEMGREL	0x0030
EMG		PMDxEMGCR	0x0034
EMG		PMDxEMGSTA	0x0038
OVV		PMDxOVVCR	0x003C
OVV		PMDxOVVSTA	0x0040
		PMDxDTR	0x0044
	0	PMDxTRGCMP0	0x0048
	1	PMDxTRGCMP1	0x004C
	2	PMDxTRGCMP2	0x0050
	3	PMDxTRGCMP3	0x0054
		PMDxTRGCR	0x0058
		PMDxTRGMD	0x005C
		PMDxTRGSEL	0x0060
		PMDxTRGSYNCR	0x0064

17.2.1 PMD<sub>x</sub>MDEN(PMD )

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

Bit	Bit Symbol	Type	
31-1	-	R	"0"
0	PWMEN	R/W	/ 0: 1: ) (PWM ) <PWMEN> = "0" ( ) <PWMEN> <PWMEN> = "1"

17.2.2 PMDxPORTMD( )

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

Bit	Bit Symbol	Type	
31-2		R	"0"
1-0	PORTMD[1:0]	R/W	00 : High-z / High-z 01 : High-z / PMD 10 : PMD / High-z 11 : PMD / PMD  (XO/YO/ZO) (PMW ) (UO/VO/WO) "High-z" ( ) PMD

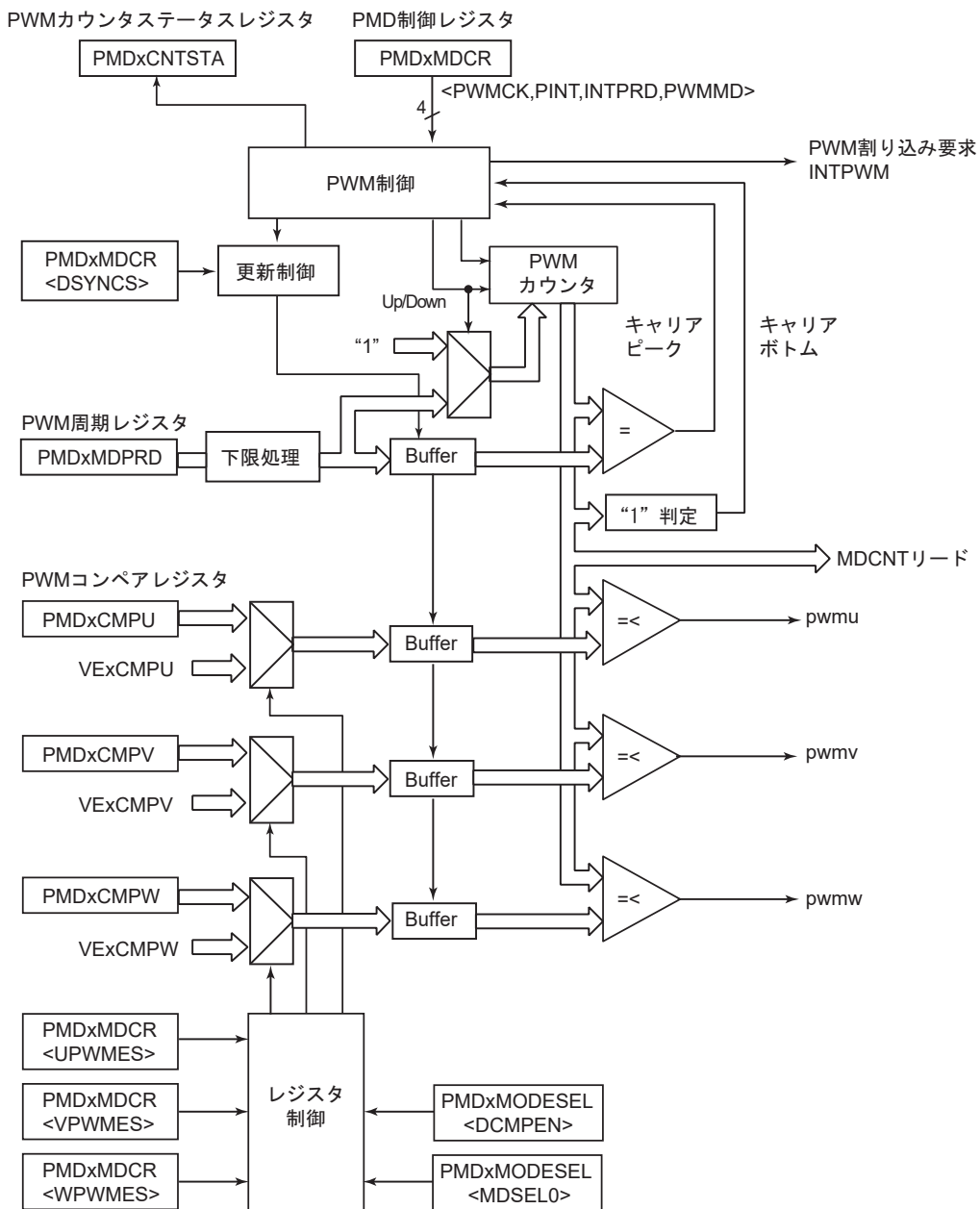
- 1) <PWMEN> = "0" PORTMD ( )
- 2) PMDxEMGCR<EMGMD[1:0]> EMG

17.2.3 PMDxMODESEL ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	DCMPEN	-	-	-	MDSEL3	MDSEL2	MDSEL1	MDSEL0
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	DCMPEN	R/W	VE PMD 0: 2 (<MDSEL0> ) 1: 2 (PWM ) ( ) <MDSEL0> = "1" ( ) (PMDxMDCR<PWMMD> = "1")
6-4		R	"0"
3	MDSEL3	R/W	3 0: (PMD : PMDxTRGSEL) 1: VE (VE : VExTRGSEL)
2	MDSEL2	R/W	2 0: (PMD : PMDxTRGCMP0 PMDxTRGCMP1) 1: VE (VE : VExTRGCMP0 VExTRGCMP1)
1	MDSEL1	R/W	1 0: (PMD : PMDxMDOUT) 1: VE (VE : VExOUTCR)
0	MDSEL0	R/W	0 0: (PMD : PMDxCMPU,PMDxCMPV PMDxCMPW) 1: VE (VE : VExCMPU,VExCMPV,VExCMPW VExEMGRS)

## 17.2.4



17-4

パルス幅変調回路は、16bitのアップ/ダウンカウンタであるPWMカウンタを持ち、 $1/f_{sys}$  (8.33[ns]@120[MHz])の分解能でPWMキャリアを生成します。PWM周期延長モード(<PWMCK>="1")にすることにより、PWMカウンタは $4/f_{sys}$  (33.3[ns]@120[MHz])の分解能でPWMキャリアを生成します。

PWMキャリアの波形モードはPWMモード0としてエッジPWM(のこぎり波変調)、モード1としてセンタPWM(三角波変調)を選択可能です(「図17-5 PWM波形」参照)。また、三角波変調ではセンタPWMの他に立ち下がりエッジ固定と立ち上がりエッジ固定のPWMを相別々に選択することができます。(「図17-6 三角波キャリアPWMのエッジ固定波形」参照)。

1. PWM 周期設定

PMDxMDPRDによりPWM周期を決定します。PMDxMDPRDはダブルバッファ構成であり、後段バッファはPWM周期で更新されます。PWM半周期毎の更新も選択できます(「表17-1 PMDxMDPRD, PMDxCMPU/V/W および VExCMPU/V/W バッファ更新制御」参照)。

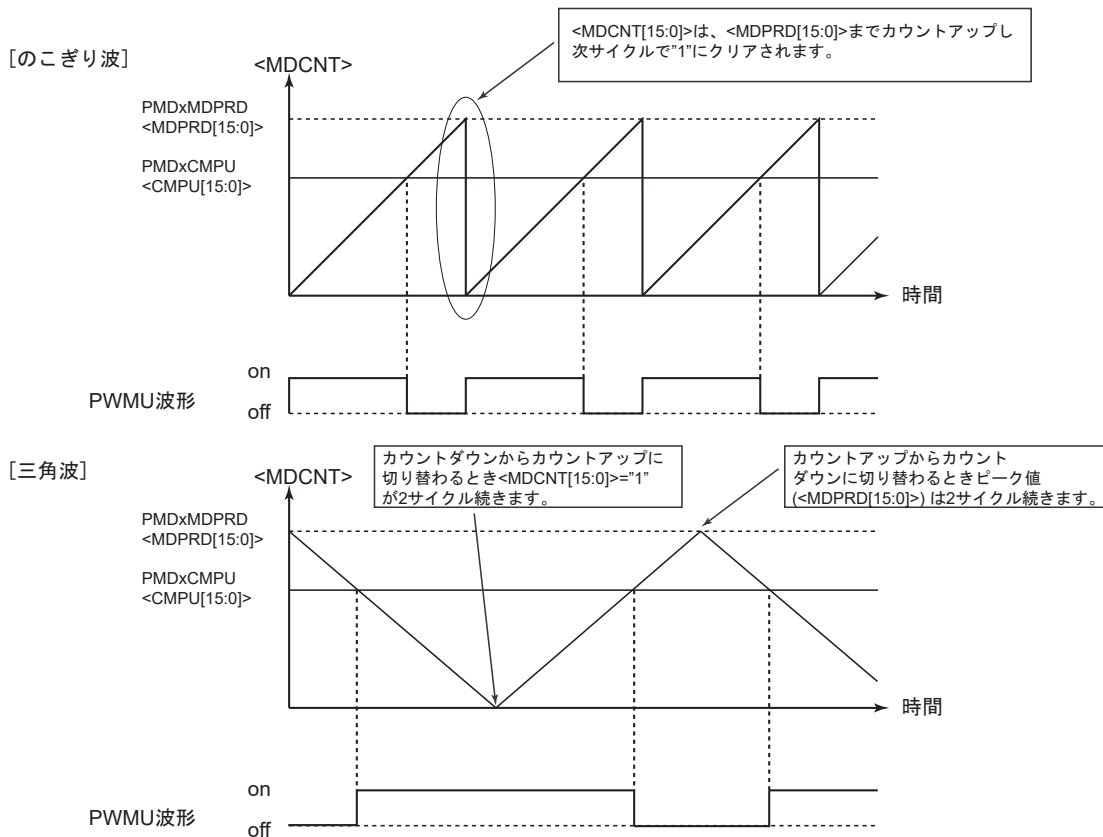
$$\text{のこぎり波変調PWM: PMDxMDPRDレジスタ設定} = \frac{\text{システムクロック } f_{\text{sys}}[\text{Hz}]}{\text{PWM周波数}[\text{Hz}]}$$

$$\text{三角波変調PWM: PMDxMDPRDレジスタ設定値} = \frac{\text{システムクロック } f_{\text{sys}}[\text{Hz}]}{\text{PWM周波数}[\text{Hz}] \times 2}$$

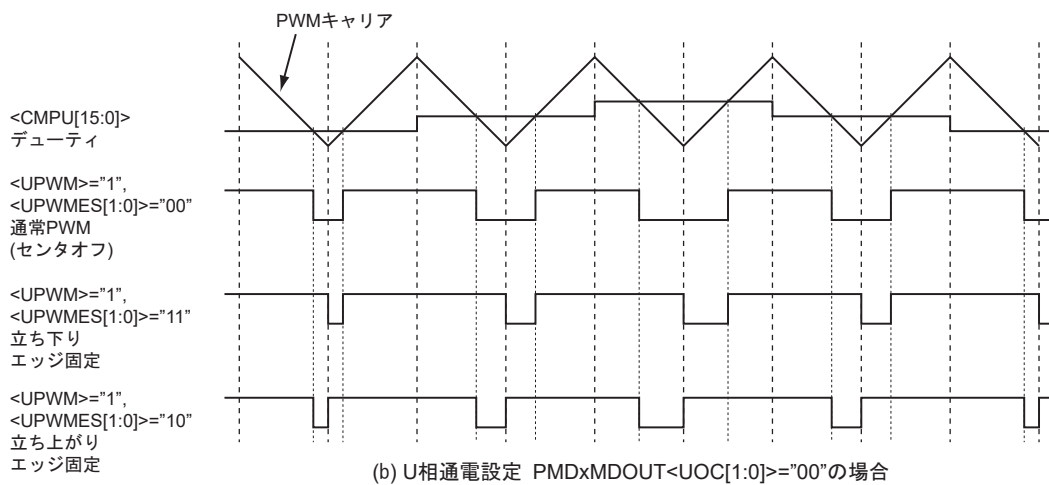
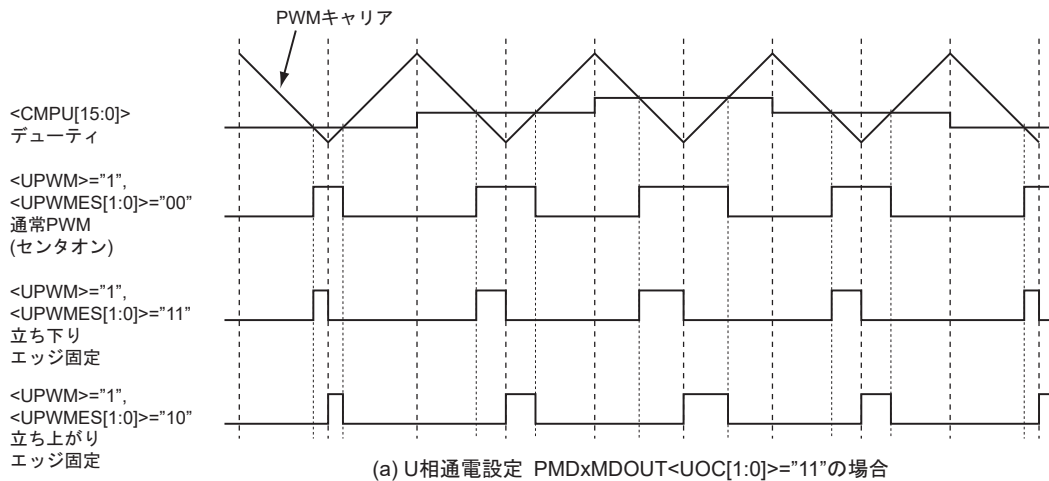
2. コンペア機能

3相のPWMコンペアレジスタ(PMDxCMPU/V/W)の値と、PWMカウンタPMDxMDCNT<MDCNT[15:0]>により生成されるPWMキャリアをコンパレータで大小比較して所望のデューティのPWM波形を生成します。

各相のPWMコンペアレジスタはダブルバッファ構成となります。PWMコンペアレジスタの値はPWM周期に同期して後段バッファにロードされます。PWM半周期での更新(半周期毎ロード)も選択できます(「表17-1 PMDxMDPRD, PMDxCMPU/V/W および VExCMPU/V/W バッファ更新制御」参照)。



17-5 PWM



## 17-6 PWM

### 3. 波形モード

2種類の3相PWMの生成方法を選択できます。

- 3相独立 Duty モード: 3相のPWMコンペアレジスタにそれぞれ独立した値を設定して、3相の独立したPWM波形を生成します。これは、正弦波等の任意の駆動波形生成に使用します。
- 3相共通 Duty モード: U相のPWMコンペアレジスタだけに設定し、U相の設定値で3相同一のPWM波形を生成します。これは、DCモータの矩形波駆動に使用します。

### 4. 割り込み処理

パルス幅変調回路ではPWM波形に同期してPWM割り込み要求を発生します。割り込み要求の発生タイミングはPWMキャリアピークとPWMキャリアボトムを選択できます。

PWM割り込みの頻度はPWM周期半周期に1回、1周期に1回、2周期に1回、4周期に1回を選択することができます。

17.2.4.1 PMDxMDCR (PMD )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	WPWMES		VPWMES		UPWMES		DSYNCS	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	DTCREN	PWMCK	SYNTMD	DTYMD	PINT	INTPRD		PWMMD
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-14	WPWMES[1:0]	R/W	W PWM 00: ( PWM) 01: Resevred 10: PWM ( ) 11: PWM ( ) ) (<PWMMD> = "1")
13-12	VPWMES[1:0]	R/W	V PWM 00: ( PWM) 01: Reserved 10: PWM ( ) 11: PWM ( ) ) (<PWMMD> = "1")
11-10	UPWMES[1:0]	R/W	U PWM 00: ( PWM) 01: Reserved 10: PWM ( ) 11: PWM ( ) ) (<PWMMD> = "1")
9-8	DSYNCS[1:0]	R/W	Duty PWM 00: (INTPRD) ( 17-1 ) 0.5 (<INTPRD> = "00")  01: PWM 10: PWM 11: PWM 1) (<PWMMD> = "0") 2) PMDxMDEN<PWMEN> = "0"
7	DTCREN	R/W	0: 1:
6	PWMCK	R/W	PWM 0: 1: 4 PWM : 1/fsys ( 8.33[ns]@120[MHz] ) / 2/fsys ( 16.7[ns]@120[MHz] ) 4 : 4/fsys ( 33.3[ns]@120[MHz] ) / 8/fsys ( 66.7[ns]@120[MHz] )
5	SYNTMD	R/W	<nOC>,<nPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4)

Bit	Bit Symbol	Type	
4	DTYMD	R/W	DUTY 0: 3 1: 3  PMDxCMPU/V/W 3                      PMDxCMPU                      3
3	PINT	R/W	PWM 0: PWM                      (PMDxMDCNT<MDCNT[15:0]> = 0x0001) 1: PWM                      (PMDxMDCNT<MDCNT[15:0]> = PMDxMDPRD<MDPRD[15:0]>)  1) PWM                      (<PWMMD> = "0") 2)                      0.5                      (<INTPRD> = "00")
2-1	INTPRD[1:0]	R/W	PWM 00: PWM 0.5 1) PWM                      (<PWMMD> = "1") 2)                      (PMDxCMPU/V/W)                      PMDxMDPRD                      PWM  01: PWM 1 10: PWM 2 11: PWM 4 PWM                      PWM                      0.5                      /1                      /2                      /4                      1
0	PWMMD	R/W	PWM 0: PWM                      0(                      PWM                      ) 1: PWM                      1(                      PWM                      )

17-1 PMDxMDPRD, PMDxCMPU/V/W                      VExCMPU/V/W

<DSYNCS[1:0]>	<INTPRD[1:0]>	
00	1x	PWM
	x1	PWM
	00	PWM
01	xx	PWM
10	xx	PWM
11	xx	PWM

x : Don't care

17-2 PMDxCMPU/V/W      VExCMPU/V/W

<DSYNCS[1:0]>	<INTPRD[1:0]>		
		<UPWMES[1]> <VPWMES[1]> <WPWMES[1]>	
01	xx	x	VE
10	xx	x	VE
11	xx	0	:PMD :VE
		1	VE
00	00	0	:PMD :VE
		1	VE

) <MDSEL0> = "1", <DCMEN> = "1", <PWMMD> = "1"

x : Don't care

17.2.4.2 PMDxCNTSTA (PWM )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	UPDWN	R	PWM 0: 1: PWM ) (PMDxMDCR<PWMMD> = "0") "0"

17.2.4.3 PMDxMDCNT(PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	MDCNT[15:0]	R	PWM PWM $1/f_{sys}$ ( 8.33[ns]@120[MHz] ) 1) 4 (PMDxMDCR<PWMCK> = "1") $4/f_{sys}$ ( 33.3[ns]@120[MHz] ) 2) PMD (PMDxMDEN<PWMEN> = "0") PWM PMDxMDCR<PWMMD> ( ) <PWMMD> = "0" : 0x0001 <PWMMD> = "1" : PMDxMDPRD<MDPRD[15:0]>



17.2.4.5 PMDxCMPU ( U PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CMPU[15:0]	R/W	U PWM 0x0000 ~ 0xFFFF ) <CMPU> > <MDPRD> 100%  U PWM PWM PWM PMDxMDCR<PWMCK> PWM <PWMMD>  <PWMCK> = "0" <PWMMD> = "0" : <CMPU> × 1/fsys <PWMMD> = "1" : <CMPU> × 2/fsys <PWMCK> = "1" <PWMMD> = "0" : <CMPU> × 4/fsys <PWMMD> = "1" : <CMPU> × 8/fsys

- 1) "0" ( ) PMDxMODESEL<MDSEL0>
- 2) ( 8bit([15:8]), 8bit([7:0]) )
- 3) PMDxCMPU PWM
- 4) 17-1 PMDxMDPRD, PMDxCMPU/V/W VExCMPU/V/W
- 5) ( )

17.2.4.6 PMDxCMPV ( V PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CMPV[15:0]	R/W	V PWM 0x0000 ~ 0xFFFF ) <CMPV> > <MDPRD> 100% V PWM PWM PWM PMDxMDCR<PWMCK> PWM <PWMMD> <PWMCK> = "0" <PWMMD> = "0" : <CMPV> × 1/fsys <PWMMD> = "1" : <CMPV> × 2/fsys <PWMCK> = "1" <PWMMD> = "0" : <CMPV> × 4/fsys <PWMMD> = "1" : <CMPV> × 8/fsys

- 1) "0" ( ) PMDxMODESEL<MDSEL0>
- 2) ( 8bit([15:8]), 8bit([7:0]) )
- 3) PMDxCMPV PWM
- 4) 17-1 PMDxMDPRD, PMDxCMPU/V/W VExCMPU/V/W
- 5) ( )

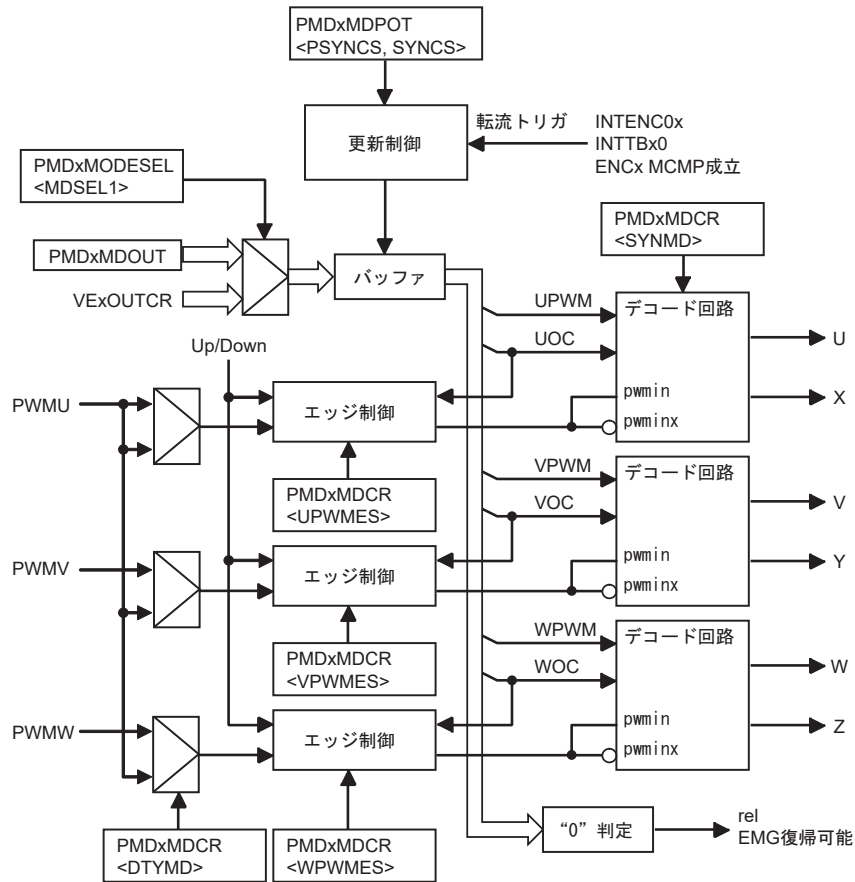
17.2.4.7 PMDxCMPW (W PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CMPW[15:0]	R/W	W PWM 0x0000 ~ 0xFFFF ) <CMPW> > <MDPRD> 100% W PWM PWM PWM PMDxMDCR<PWMCK> PWM <PWMMMD> <PWMCK> = "0" <PWMMMD> = "0" : <CMPW> × 1/fsys <PWMMMD> = "1" : <CMPW> × 2/fsys <PWMCK> = "1" <PWMMMD> = "0" : <CMPW> × 4/fsys <PWMMMD> = "1" : <CMPW> × 8/fsys

- 1) "0" ( ) PMDxMODESEL<MDSEL0>
- 2) ( 8bit([15:8]), 8bit([7:0]) )
- 3) PMDxCMPW PWM
- 4) 17-1 PMDxMDPRD, PMDxCMPU/V/W VExCMPU/V/W
- 5) ( )

## 17.2.5



17-7

通電制御レジスタ **PMDxMDOUT**(**VExOUTCR**)と出力設定レジスタ **PMDxMDPOT** の設定により、出力ポートの制御を行います。**PMDxMDOUT**(**VExOUTCR**)レジスタはダブルバッファ構成であり、更新タイミングは **PWM** への同期更新と非同期更新を選択できます。また、トリガ入力に同期した更新を選択することも可能です。(更新タイミングは「表 17-3 **PMDxMDOUT**(**VExOUTCR**)バッファの更新タイミング」を参照してください。)

6本のポートの出力設定は、上相出力(**UO, VO, WO**)と下相出力(**XO, YO, ZO**)のそれぞれにローアクティブ/ハイアクティブ設定を **PMDxMDPOT** の**<POLH>**,**<POLL>**により行うことができます。さらに、**U, V, W** それぞれに、**PWM** 出力と **High/Low** 出力との選択を **PMDxMDOUT**(**VExOUTCR**)の**<WPWM>**,**<VPWM>**,**<UPWM>**により設定します。**PWM** 出力を選択すると **PWM** 波形が、**High/Low** 出力を選択すると **High** 固定または **Low** 固定の出力が得られます。それぞれの出力を **High** にするか **Low** にするかは **PMDxMDOUT**(**VExOUTCR**)の**<WOC>**,**<VOC>**,**<UOC>**で選択します。**PMDxMDOUT**(**VExOUTCR**)と **PMDxMDPOT** によるポート出力設定と **PMD** 制御レジスタ(**PMDxMDCR**)のポート出力モード設定によって得られる端子出力の関係については「表 17-4 **<UOC>**,**<VOC>**,**<WOC>**および**<UPWM>**,**<VPWM>**,**<WPWM>**の各ビット設定によるポート出力」を参照してください。

通電制御回路はエンコーダ入力回路(**A-ENC**)で **PWM** 同期サンプリングするための **PWM** 信号(**PWMON**)を出力します。

17-3 PMD<sub>x</sub>MDOUT(VE<sub>x</sub>OUTCR)

		PSYNCS			
		00	01	10	11
SYNCS	00		PWM	PWM	PWM PWM
	01	INTENC0x	INTENC0x PWM	INTENC0x PWM	INTENC0x PWM PWM
	10	INTTBx0	INTTBx0 PWM	INTTBx0 PWM	INTTBx0 PWM PWM
	11	ENCx CTRGO (MCMP )	ENCx CTRGO PWM	ENCx CTRGO PWM	ENCx CTRGO PWM PWM

) PMD (PMDxMDCR<PMWEN> = "0")

17.2.5.1 PMDxMDPOT (PMD )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	SYNCS	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	POLH	POLL	PSYNCS	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-10		R	"0"
9-8	SYNCS[1:0]	R/W	PMDxMDOUT(VExOUTCR) ( ) 00 01 INTENC0x (ENCx ) 10 INTTBx0 (TMRBx ) 11 CTRGO(ENCx MCMP )  1) <PSYNC> <SYNCS> 17-3 PMDxMDOUT(VExOUTCR) 2) PMD (PMDxMDEN<PWMEN> = "0")
7-4		R	"0"
3	POLH	R/W	(UO,VO,WO) 0: 1:
2	POLL	R/W	(XO,YO,ZO) 0: 1:
1-0	PSYNCS[1:0]	R/W	PMDxMDOUT(VExOUTCR) (PWM ) 00: PWM PMDxMDOUT/VExOUTCR 01: (<MDCNT[15:0]> = "1" ) 10: (<MDCNT[15:0]> = <MDPRD[15:0]> ) 11:  1) ( PMDxMDCR<PWMMD> = "0" ) "00"  2) <PSYNC> <SYNCS> ( 17-3 PMDxMDOUT(VExOUTCR) ) 3) PMD (PMDxMDEN<PWMEN> = "0") PWM

) PMDxMDEN<PWMEN> = "0"

17.2.5.2 PMDxMDOUT(PMD )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	WPWM	VPWM	UPWM
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	WOC		VOC		UOC	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-11		R	"0"
10	WPWM	R/W	W PWM 0: H/L 1: PWM <WOC>,<WPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4 )
9	VPWM	R/W	V PWM 0: H/L 1: PWM <VOC>,<VPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4 )
8	UPWM	R/W	U PWM 0: H/L 1: PWM <UOC>,<UPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4 )
7-6		R	"0"
5-4	WOC[1:0]	R/W	W <WOC>,<WPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4 )
3-2	VOC[1:0]	R/W	V <VOC>,<VPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4 )
1-0	UOC[1:0]	R/W	U <UOC>,<UPWM>,<POLH>,<POLL>,<SYNTMD> ( 17-4 )

- 1) PMDxMDOUT(VExOUTCR)  
PMDxMODESEL<MDESEL> "0" ( )
- 2) ( 8bit[[15:8]], 8bit[[7:0]] )
- 3) PWM
- 4) 17-3 PMDxMDOUT(VExOUTCR)
- 5) ( )

17-4 <UOC>,<VOC>,<WOC> <UPWM>,<VPWM>,<WPWM>

PMDxMDCR<SYNTMD> = "0"

(PMDxMDPOT<POLH><POLL> = "11")

PMDxMDOUT		<WPWM><VPWM><UPWM>			
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	0:H/L		1: PWM	
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	PWM
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

PMDxMDCR<SYNTMD> = "0"

(PMDxMDPOT<POLH><POLL> = "00")

PMDxMDOUT		<WPWM><VPWM><UPWM>			
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	0:H/L		1: PWM	
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	$\overline{\text{PWM}}$
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

PMDxMDCR<SYNTMD> = "1"

(PMDxMDPOT<POLH><POLL> = "11")

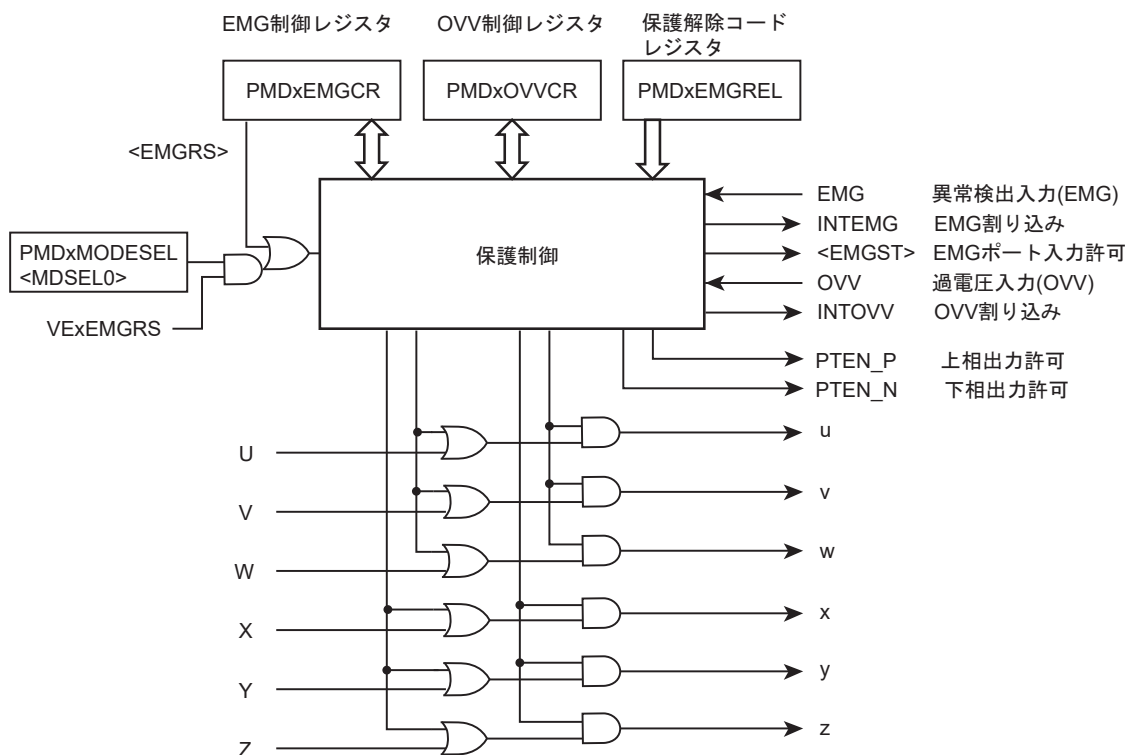
PMDxMDOUT		<WPWM><VPWM><UPWM>			
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	0:H/L		1: PWM	
0	0	L	L	$\overline{\text{PWM}}$	PWM
0	1	L	H	L	$\overline{\text{PWM}}$
1	0	H	L	PWM	L
1	1	H	H	PWM	$\overline{\text{PWM}}$

PMDxMDCR<SYNTMD> = "1"

(PMDxMDPOT<POLH><POLL> = "00")

PMDxMDOUT		<WPWM><VPWM><UPWM>			
<WOC[1]> <VOC[1]> <UOC[1]>	<WOC[0]> <VOC[0]> ><UOC[0]>	0:H/L		1: PWM	
0	0	H	H	PWM	$\overline{\text{PWM}}$
0	1	H	L	H	PWM
1	0	L	H	$\overline{\text{PWM}}$	H
1	1	L	L	$\overline{\text{PWM}}$	PWM

17.2.6



17-8

保護制御回路は、EMG 保護制御部と OVV 保護制御部から構成されます。

### 17.2.6.1 (EMG )

保護制御回路は EMG 保護制御とポート出力禁止部から構成され、EMG 入力が高レベルになると動作します。

EMG 保護回路は緊急停止用の保護回路であり、EMG 入力があった場合(High → Low)、直ちに 6 本の PWM 出力を禁止し(PMDxEMGCR<EMGMD>の設定によります)、EMG 割り込み(INTEMG)を発生します。また<EMGMD>の設定により、外部出力ポートを"Hi-z"に設定する制御信号を出力します。

ツールブレークによる PMD 停止制御時も同様に 6 本の PWM 出力を禁止しますが、但し PMDxPORTMD<PORTMD>の設定によります。ツールブレーク時は PMDxPORTMD <PORTMD>の設定により、外部出力ポートの"Hi-z"制御を選択することができます。また、PMDxEMGSTA<EMGST>をリードした時、"1"の場合は EMG 保護状態であることを示します。

EMG 保護は EMG 制御レジスタ(PMDxEMGCR)で設定します。

EMG 保護状態の時は、ポート出力を全てインアクティブに設定(PMDxMDOUT(VExOUTCR)の<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC>を全て"0")後(注1)、PMDxEMGCR(VExEMGRS)の<EMGRS>に"1"を設定することにより EMG 保護状態から復帰することができます。また、EMG 機能を禁止するには EMG 解除レジスタ(PMDxEMGREL)に 0x5A と 0xA5 を順番に設定後、PMDxEMGCR<EMGEN>に"0"を設定します(3 命令連続して行います)。ただし、EMG 保護入力が Low の期間は、復帰処理を行っても無視されます。PMDxEMGSTA<EMGI>をリードしポート入力が High になったことを確認してから復帰処理をしてください。

EMG 保護回路を禁止するには EMG 解除レジスタに所定のキーコード(0x5A、0xA5)を設定することではじめて可能になり、誤って EMG 保護回路を禁止することを防止します。

1)PMDxMDOUT(VExOUTCR)

2) EMG

EMG	( )	EMG	EMG
	EMG		
1:		(PxFR)	EMG
2:	PMDxEMGSTA<EMGI>	"1"	
3:	PMDxMDOUT(VExOUTCR)	<UPWM>,<VPWM>,<WPWM>,<UOC>,<VOC>,<WOC>	"0"
4:	PMDxEMGCR(VExEMGRS)	<EMGRS>	"1" EMG
	EMG		
5:	PMDxEMGREL	(0x5A → 0xA5 )	
6:	PMDxEMGCR<EMGEN>	"0"	EMG

17.2.6.2 PMDxEMGREL (EMG )

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	EMGREL[7:0]	W	EMG/OVV "0x5A"→"0xA5" EMG OVV PMDxEMGCR<EMGEN> = "0" PMDxOVVCR<OVVEN> = "0"

) EMG OVV

17.2.6.3 PMDxEMGCR (EMG )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	EMGCNT			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	INHEN	EMGMD		-	EMGRS	EMGEN
	0	0	1	1	1	0	0	1

Bit	Bit Symbol	Type	
31-12		R	"0"
11-8	EMGCNT[3:0]	R/W	EMG 0x0 ~ 0xF("0" )  <EMGCNT[3:0]> × 16/fsys ( 133[ns] @120[MHz] )
7-6		R	"0"
5	INHEN	R/W	/ 0: 1: PMD PMD )
4-3	EMGMD[1:0]	R/W	EMG 00: High-z 01: / High-z 10: High-z/ 11: High-z ) : PWM EMG (UO,VO,WO) (XO,YO,ZO)
2		R/W	"0"
1	EMGRS	W	EMG 0: - 1: PMDxMDOUT(VExOUTCR) "0x000" <EMGRS> "1" EMG ) "0" ) PMDxMDOUT(VExOUTCR) "0x000" ) PMDxEMGSTA<EMGI> "1"
0	EMGEN	R/W	EMG / 0: 1: EMG <EMGEN> "0" PMDxEMGREL<EMGREL[7:0]> "0x5A"→"0xA5" (3 )

17.2.6.4 PMDxEMGSTA (EMG )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	EMGI	EMGST
	0	0	0	0	0	0		0

Bit	Bit Symbol	Type	
31-2		R	"0"
1	EMGI	R	EMG EMG EMG
0	EMGST	R	EMG 0: 1: EMG

#### 17.2.6.5 (OVV )

OVV 保護制御とポート出力禁止部から構成されます。OVV 入力ポートがアクティブになることで動作します。

OVV 保護回路は、一定期間(OVV カウントで設定)過電圧信号入力があった場合(High→Low)、通電制御部で6本のポート出力を High または Low に固定します。またその時、OVV 割り込み(INTOVV)が発生します。設定により、下相オフ、上相オフ、全相オフを選択することができます。

OVV 保護は OVV 制御レジスタ PMDxOVVCR で設定します。また、PMDxOVVSTA<OVVST>をリードした時、"1"の場合は OVV 保護回路が動作中であることを示します。

PMDxOVVCR<OVVRS>に"1"を設定することにより OVV 保護からの復帰可能状態となり、OVV 保護入力解除後、保護状態からの復帰タイミングで自動的に OVV 保護から復帰します。(OVV 保護入力が Low の期間は、保護状態からの復帰は行われません。ポート入力の状態は PMDxOVVSTA<OVVI>をリードすることで確認することができます。)

OVV から復帰するタイミングは、PWM 周期(PWM カウント PMDxMDCNT と PMDxMDPRD との一致時。ただし PWM0.5 周期割り込み設定時は PWM カウントが、"1"または PMDxMDPRD との一致時)となります。また、OVV 機能を禁止するには EMG 解除レジスタ PMDxEMGREL に 0x5A と 0xA5 を順番に設定後、PMDxOVVCR<OVVEN>に"0"を設定します。(3 命令連続して行ってください)

OVV 保護回路を禁止するには EMG 解除レジスタに所定のキーコード(0x5A、0xA5)を設定することではじめて可能になり、誤って OVV 保護回路を禁止することを防止します。

17.2.6.6 PMDxOVVCR (OVV )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	OVVCNT			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	ADIN1EN	ADIN0EN	OVVMD		OVSISEL	OVSRS	OVSSEN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-12		R	"0"
11-8	OVVCNT[3:0]	R/W	OVV 1 ~ 15 ("0" "1" ) OVV <OVVCNT[3:0]> × 16/fsys ( 133[ns] @120[MHz] ) >OVVCNT[3:0]> (<OVSISEL> = "0")
7		R	"0"
6	ADIN1EN	R/W	ADCx 1 0: 1: ADCx 1 / ADCx 1 OVV ADC (OVV (<OVSISEL> = "1") ) )ADCx 12 / "AD "
5	ADIN0EN	R/W	ADCx 0 0: 1: ADCx 0 / ADCx 0 OVV ADC (OVV (<OVSISEL> = "1") ) )ADCx 12 / "AD "
4-3	OVVMD	R/W	OVV 00: 01: 10: 11: OVV (UO,VO,WO) (XO,YO,ZO) / ) <POLL>,<POLH> )OVV EMG <EMGMD[1:0]>
2	OVSISEL	R/W	OVV 0: 1: ADC OVV ADC )ADC OVV <OVVCNT[3:0]> ( )

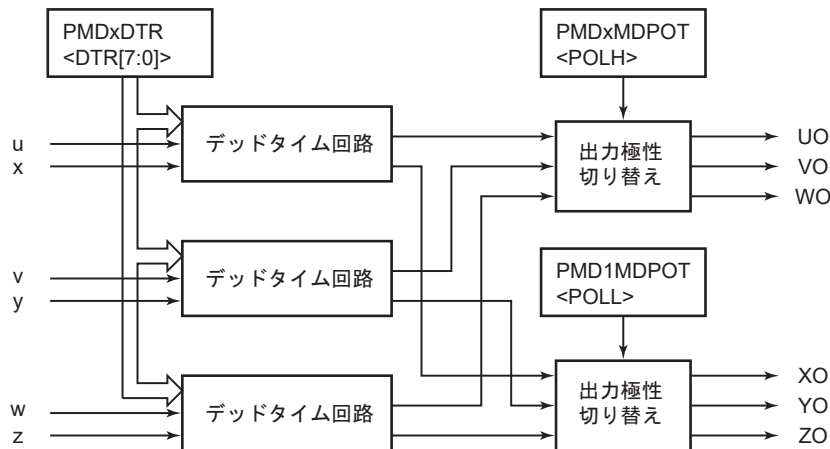
Bit	Bit Symbol	Type	
1	OVRS	R/W	OVV 0: 1:  OVV H → L OVV PWM (PMDxMDPRD) PMDxMDPRD, PMDxCMPU/V/W VExCMPU/V/W OVV High 17-1
0	OVVEN	R/W	OVV / 0: 1:  ) EMG (PMDxEMGREL) "0x5A" → "0xA5" <OVVEN> = "0" 3

17.2.6.7 PMDxOVVSTA (OVV )

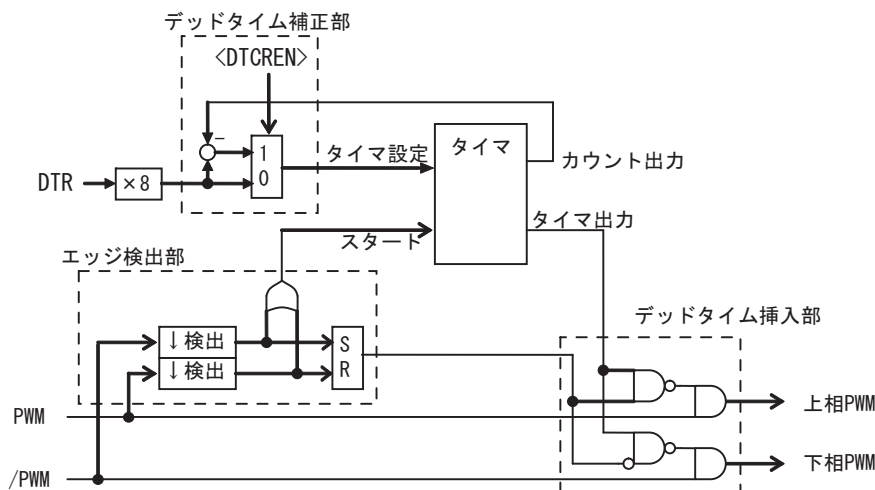
	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	OVVI	OVVST
	0	0	0	0	0	0		0

Bit	Bit Symbol	Type	
31-2		R	"0"
1	OVVI	R	OVVI OVVI OVV (PMDxOVVCR<OVVISEL> )
0	OVVST	R	OVV 0: 1: OVV

## 17.2.7



17-9



17-10

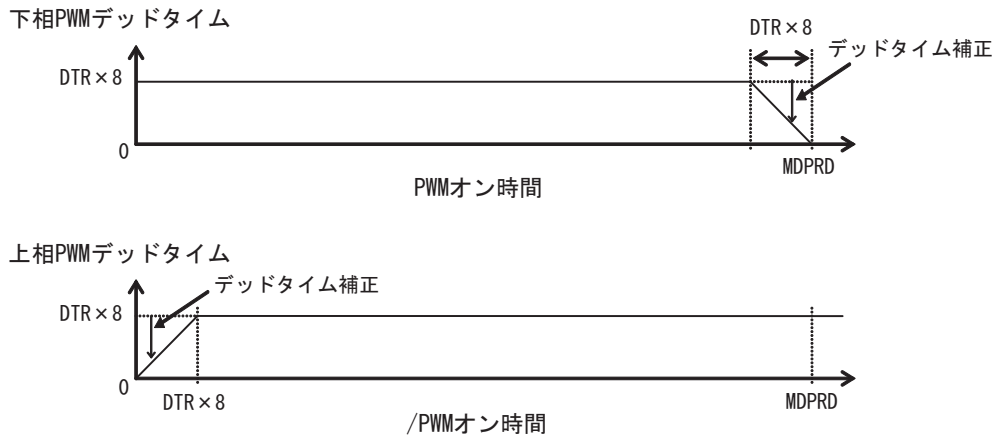
デッドタイム制御回路はデッドタイム部と出力極性切替部から構成されます。デッドタイム回路は、エッジ検出部とタイマ部とデッドタイム挿入部およびデッドタイム補正部から構成されます。(「図 17-10 デッドタイム回路構成」参照)

U,V,W のそれぞれの相において、上下相が逆転する場合に上下相が短絡するのを回避するため、デッドタイム回路によりオン時間を遅延させます。遅延時間は、8bit 値により  $8/fs_{sys}$  (66.7[ns]@120 [MHz]) の分解能で設定が可能です。遅延時間はデッドタイムレジスタ(PMDxDTR<DTR[7:0]>)に設定します。

出力極性切替回路は、上相出力(UO,VO,WO)と下相出力(XO,YO,ZO)のそれぞれに対してハイアクティブ/ローアクティブを PMD 出力設定レジスタ PMDxDTPOT の<POLH>,<POLL>により設定します。

デッドタイム補正部は、PMDxMDCR<DTCREN>を"1"に設定することで、上相 PWM または下相 PWM の一方のオン期間が 0 の場合にもう一方の PWM の遅延時間を短く補正します。補正方法は、デッドタイム期間の途中で PWM がオフに変わった場合に、デッドタイム期間の残り時間 (デッドタイムレジスタ設定時間-オン時間) で反対相の遅延時間を短くします。上相 PWM がデッドタイム期間中にオフに変わった場合には下相の遅延時間を短く補正し、下相 PWM がデッドタイム期間中にオ

フに変わった場合には上相 PWM の遅延時間を短く補正します。遅延時間は「図 17-11 デッドタイム補正」に示すように上相 PWM のデューティ 100%付近と下相 PWM のデューティ 0%付近で補正されます。



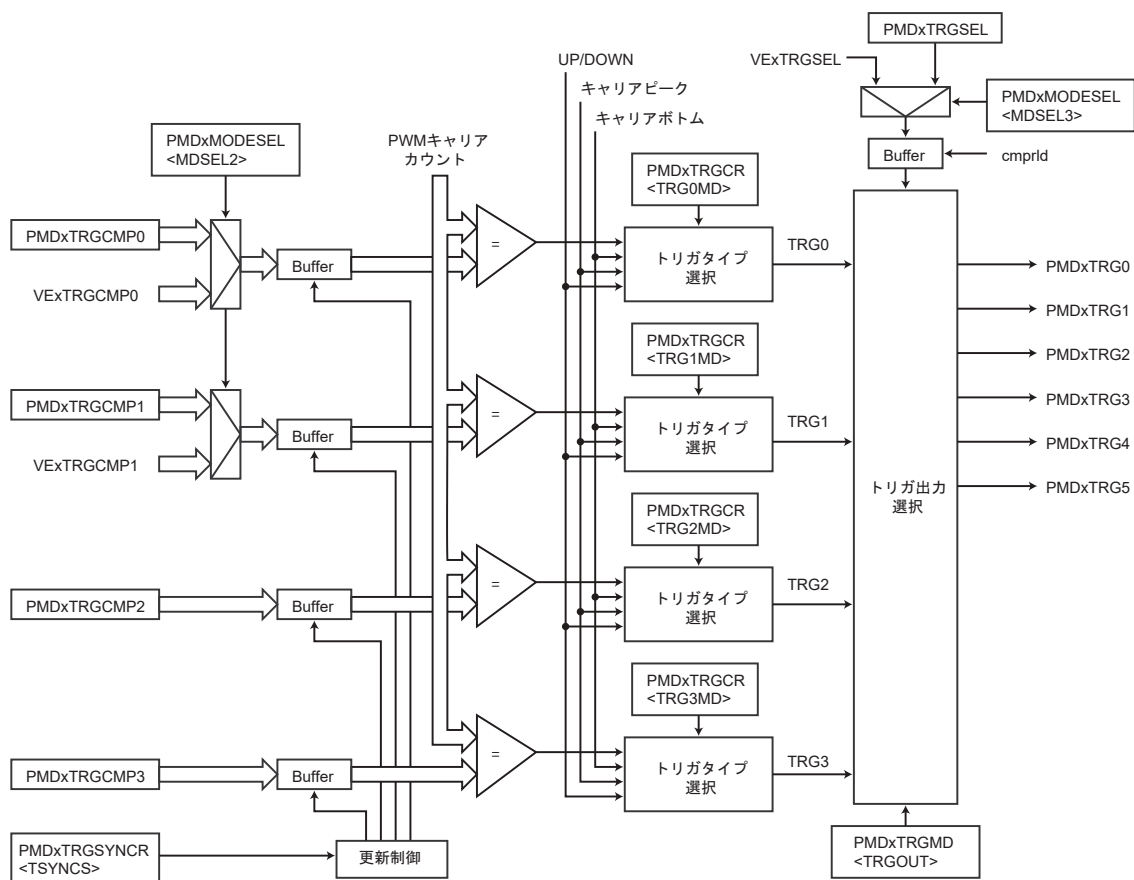
17-11

17.2.7.1 PMDxDTR ( )

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	DTR[7:0]	R/W	0x00 ~ 0xFF DTR × 8/fsys ( 17[μs]@120[MHz])

17.2.8



17-12

同期トリガ生成回路は PWM に同期して ADC のサンプリングを行うための 4 種類(TRG0~TRG3)のトリガ信号を生成します。PMDxMODESEL<MDESEL3>で VE モード選択すると、PMDxTRGCMP0 と PMDxTRGCMP1 は VE レジスタの VExTRGCMP0, VExTRGCMP1 に切り替わります。

トリガタイミングは以下の 6 種類が選択可能です。

1. アップカウント動作時のコンペアー一致(注)
2. ダウンカウント動作時のコンペアー一致(注)
3. アップ/ダウンカウント両動作でのコンペアー一致(注)
4. PWM キャリアピーク
5. PWM キャリアボトム
6. PWM キャリアピークと PWM キャリアボトム

) PWM      PMDxMDCNT<MDCNT[15:0]>    PMDxTRGCMPn<TRGCMPn[15:0]>

トリガ選択出力モード  $\text{PMDxTRGMD}\langle\text{TRGOUT}\rangle = "1"$  では、TRG0 信号 ( $\text{PMDxTRGCMP0}$  ( $\text{VExTRGCMP0}$ ),  $\text{PMDxTRGCR}\langle\text{TRG0MD}\rangle$ ) がトリガ出力選択レジスタ  $\text{PMDxTRGSEL}$  ( $\text{VExTRGSEL}$ ) で選択したトリガ ( $\text{PMDxTRG0}\sim 5$ ) 信号として出力されます。

なお、エッジモード(ノコギリ波キャリアモード)選択時はコンペア一致はアップカウント時となります。また、EMG 保護動作時もトリガ出力許可する場合は  $\text{PMDxTRGMD}\langle\text{EMGTGE}\rangle$  に "1" を設定します。

17.2.8.1 PMDxTRGCMP0 ( 0)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRGCMP0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP0							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TRGCMP0 [15:0]	R/W	<TRGCMP0[15:0]> : 1 ~ [<MDPRD[15:0]> -1] PWM <MDCNT[15:0]> (TRG0) ) <TRGCMP0> = "0" <TRGCMP0> ≥ [<MDPRD[15:0]> ]

- 1) "0" ( ) PMDxMODESEL<MDSEL2>
- 2) ( 8bit([15:8]), 8bit([7:0]) )
- 3) PWM
- 4) 17-5
- 5) ( )

17-5

<TSYNCS>	<TRGnMD>	TRGCMPn
00	000	
	001	PWM
	010	PWM
	011	PWM ( 1)
	1xx	
01	xxx	PWM
10	xxx	PWM
11	xxx	PWM ( 1)

- ) x : Don't care
- ) PMDxMDEN<PWMEN> = "0"
- 1) PWM (PMDxMDCR<PMMMD> = "0")

17.2.8.2 PMDxTRGCMP1 ( 1)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRGCMP1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP1							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TRGCMP1 [15:0]	R/W	: 1 ~ [<MDPRD[15:0]> -1 PWM <MDCNT[15:0]> (TRG1) ) <TRGCMP1> = "0" <TRGCMP1> ≥ [<MDPRD[15:0]> ]

- 1) "0" ( ) PMDxMODESEL<MDSEL2>
- 2) ( 8bit([15:8]), 8bit([7:0]) )
- 3) PWM
- 4) 17-5
- 5) ( )

17.2.8.3 PMDxTRGCMP2 ( 2)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRGCMP2							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP2							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TRGCMP2 [15:0]	R/W	PWM : 1 ~ [<MDPRD[15:0]> -1 (<MDCNT[15:0]> (TRG2) ) <TRGCMP2> = "0" <TRGCMP2> ≥ [<MDPRD[15:0]> ]

- 1) ( 8bit([15:8]), 8bit([7:0]) )
- 2) PWM
- 3) 17-5
- 4) ( )

17.2.8.4 PMDxTRGCMP3 ( 3)

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRGCMP3							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRGCMP3							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TRGCMP3 [15:0]	R/W	: 1 ~ [<MDPRD[15:0]> -1 PWM <MDCNT[15:0]> (TRG3) ) <TRGCMP3> = "0" <TRGCMP3> ≥ [<MDPRD[15:0]> ]

- 1) ( 8bit([15:8]), 8bit([7:0]) )
- 2) PWM
- 3) 17-5
- 4) ( )

17.2.8.5 PMDxTRGCR ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TRG3BE	TRG3MD			TRG2BE	TRG2MD		
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TRG1BE	TRG1MD			TRG0BE	TRG0MD		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15	TRG3BE	R/W	PMDxTRGCMP3<TRGCMP3[15:0]> 0 : 1 : ( ) ) 17-5 ) PMDxMDEN<PWMEN> = "0"
14-12	TRG3MD[2:0]	R/W	PMDxTRGCMP3<TRGCMP3[15:0]> <TRGCMP3[15:0]> 000 : 001 : 010 : 011 : / 100 : PWM 101 : PWM 110 : PWM / 111 : ) PMDxMDCR<PWMMD> = "0"( ) "001" "101" ) "011" PMDxTRGCMP3 = "0x0001" PMDxMDCR<PWMMD> = "1"( ) 1 1
11	TRG2BE	R/W	PMDxTRGCMP2<TRGCMP2[15:0]> 0 : 1 : ( ) ) 17-5 ) PMDxMDEN<PWMEN> = "0"

Bit	Bit Symbol	Type	
10-8	TRG2MD[2:0]	R/W	PMDxTRGCMP2<TRGCMP2[15:0]> <TRGCMP2[15:0]> 000 : 001 : 010 : 011 : / 100 : PWM 101 : PWM 110 : PWM / 111 :  ) PMDxMDCR<PWMMMD> = "0"( ) "001" "101"  ) "011" PMDxTRGCMP2 = "0x0001" PMDxMDCR<PWMMMD> = "1"( ) 1 1
7	TRG1BE	R/W	PMDxTRGCMP1<TRGCMP1[15:0]>  0 : 1 : ( )  ) 17-5 ) PMDxMDEN<PWMEN> = "0"
6-4	TRG1MD[2:0]	R/W	PMDxTRGCMP1<TRGCMP1[15:0]> <TRGCMP1[15:0]> 000 : 001 : 010 : 011 : / 100 : PWM 101 : PWM 110 : PWM / 111 :  ) PMDxMDCR<PWMMMD> = "0"( ) "001" "101"  ) "011" PMDxTRGCMP1 = "0x0001" PMDxMDCR<PWMMMD> = "1"( ) 1 1
3	TRG0BE	R/W	PMDxTRGCMP0<TRGCMP0[15:0]>  0 : 1 : ( )  ) 17-5 ) PMDxMDEN<PWMEN> = "0"
2-0	TRG0MD[2:0]	R/W	PMDxTRGCMP0<TRGCMP0[15:0]> <TRGCMP0[15:0]> 000 : 001 : 010 : 011 : / 100 : PWM 101 : PWM 110 : PWM / 111 :  ) PMDxMDCR<PWMMMD> = "0"( ) "001" "101"  ) "011" PMDxTRGCMP0 = "0x0001" PMDxMDCR<PWMMMD> = "1"( ) 1 1

17.2.8.6 PMDxTRGSYNCR ( )

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

Bit	Bit Symbol	Type	
31-2		R	"0"
1-0	TSYNCS	R/W	00 : PMDxTRGCR<TRGxMD> PWM PWM 01 : PWM 10 : PWM 11 : PWM ) 17-5 ) PMDxMDEN<PWMEN> = "0"

17.2.8.7 PMDxTRGMD ( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	TRGOUT	EMGTGE
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2		R	"0"
1	TRGOUT	R/W	0 : 1 : [15:0]> ) ( <TRGOUT> = "1" )
			PMDxTRG0 ~ PMDxTRG3 <TRGCMP0[15:0]> ~ <PMDCMP3 PMDxTRG4 PMDxTRG5 PMDxTRG0 ~ PMDxTRG5 17-6
0	EMGTGE	R/W	EMG 0 : 1 : EMG

17-6

<TRGOUT>		<TRGSEL[2:0]>	
<TRGOUT> = "0"	PMDxTRGCMP0	3	PMDxTRG0
	PMDxTRGCMP1		PMDxTRG1
	PMDxTRGCMP2		PMDxTRG2
	PMDxTRGCMP3		PMDxTRG3
<TRGOUT> = "1"	PMDxTRGCMP0	0	PMDxTRG0
		1	PMDxTRG1
		2	PMDxTRG2
		3	PMDxTRG3
		4	PMDxTRG4
	5	PMDxTRG5	
	PMDxTRGCMP1	3	
	PMDxTRGCMP2	3	
PMDxTRGCMP3	3		

17.2.8.8 PMDxTRGSEL ( )

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

Bit	Bit Symbol	Type	
31-3		R	"0"
2-0	TRGSEL[2:0]	R/W	000 : PMDxTRG0 001 : PMDxTRG1 010 : PMDxTRG2 011 : PMDxTRG3 100 : PMDxTRG4 101 : PMDxTRG5 110 : 111 : (<TRGOUT> = "1") PMDxTRGCMP0 ( 17-6 )

- 1) "0" ( ) PMDxMODESEL<MDESEL3>
- 2) PWM
- 3) (PMDxCMPU/V/W)
- 4) PMD (PMDxMDCR<PWMEN> = "0")



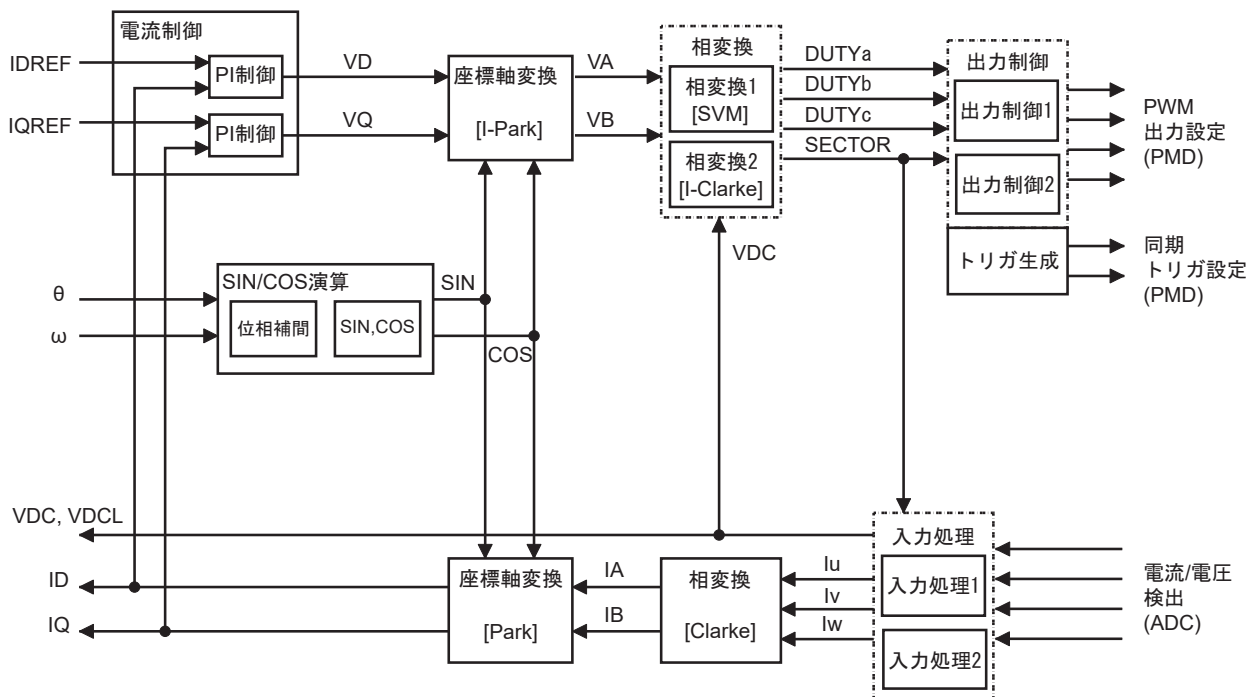
18

(A-VE)

18.1

18.1.1

1. ベクトル制御で実行される基本的な処理(座標軸変換,相変換,SIN/COS 演算)の組み込み  
演算処理は固定小数点形式データで行われます。  
→ソフトウェア処理では面倒な小数点位置管理不要
2. モータ制御回路(PMD)と A/D 変換器(ADC)を制御する I/F 処理(出力制御,トリガ生成,入力処理)の組み込み
  - ・ 固定小数点形式の演算結果をモータ制御回路設定のデータ形式に変換
  - ・ 連携動作タイミングデータの生成
  - ・ A/D 変換結果を固定小数点形式のデータに変換
3. 電流,電圧,回転速度をそれぞれの最大値を基準に正規化した値で演算  
小数点数は固定小数点形式
4. 電流制御に PI 制御の組み込み
5. 回転速度を積分する位相補間の組み込み



18-1

### 18.1.2

- 2相から3相への相変換には、空間ベクトル変換と逆クランク変換に対応します。空間ベクトル変換は、2相変調と3相変調に対応します。
- トリガ生成では、センサレス電流検出方式に対応するA/D変換器(ADC)のサンプリングタイミングを生成します。1シャント方式の電流検出に対応するタイミングを生成できます。
- 電流制御には、d軸とq軸に独立したPI制御を採用。電流制御を使用せずに直接電圧指令することも可能です。

PI制御結果を制限機能

d軸とq軸の非干渉制御可能

d軸,q軸を合わせた電圧スカラ値による出力制限可能

- SIN/COS演算には、級数展開を使った近似式を採用。位相指定は直接設定と回転速度からPWM周期で積分する位相補間が可能です。また、位相補間時に指定位相でのクリッピング可能です。

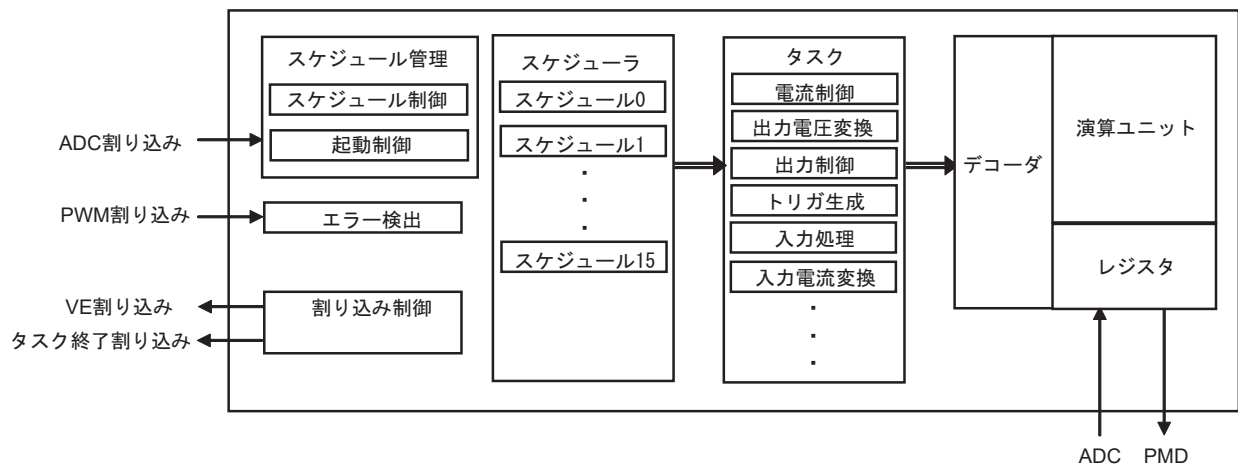
出力制御はデッドタイム補償およびPWM出力制限が可能です。

注)ベクトルエンジンを使用する場合はモータ制御回路およびA/D変換器の設定が必要です。

- モータ制御回路(PMD)はモード選択レジスタ(PMDxMODESEL)でVEモードを選択します。
- A/D変換器(ADC)はモータ制御回路(PMD)からの同期トリガ毎のプログラム(トリガ許可,AIN選択,結果レジスタ選択)を設定します。

## 18.2

ベクトルエンジンの構成を図18-2に示します。



18-2

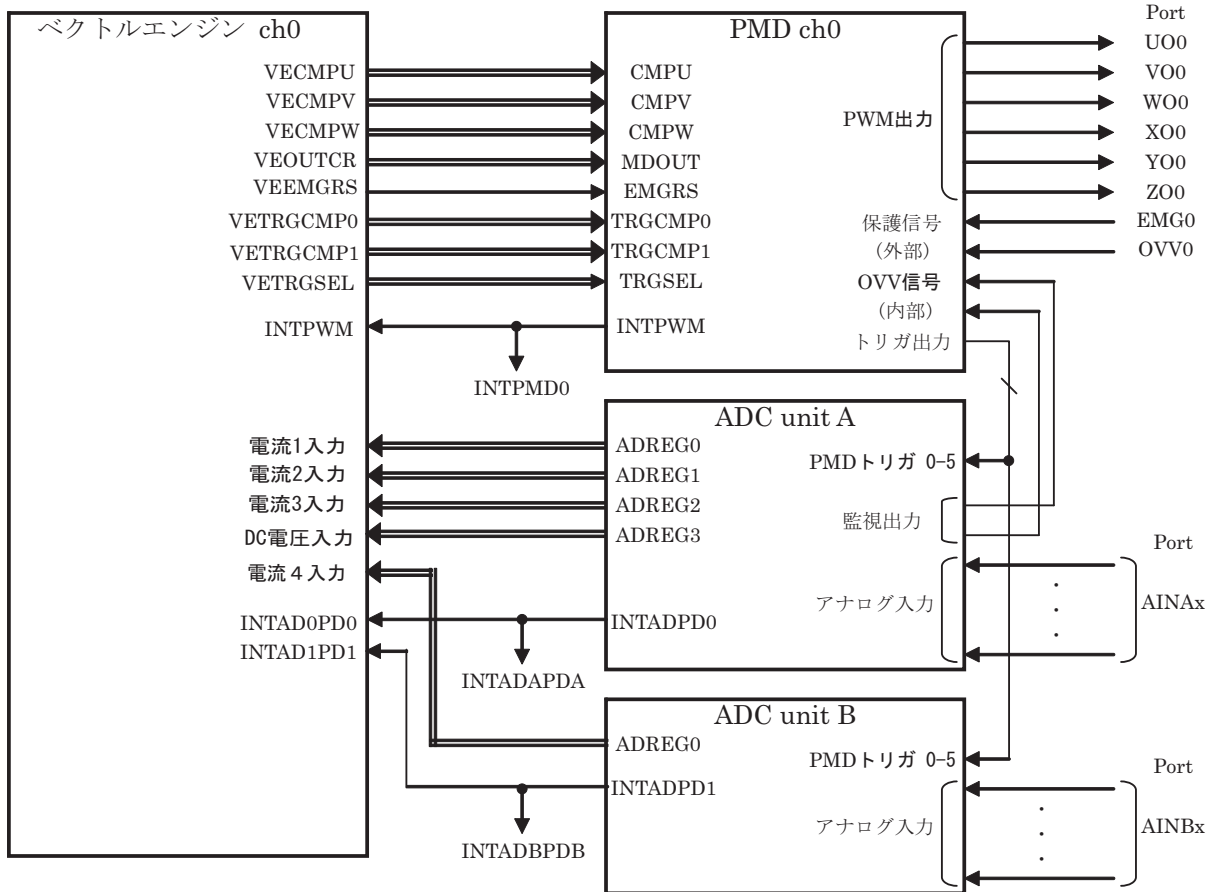
18.2.1

A/D

ベクトルエンジンは図 18-3、図 18-4 に示すようにモータ制御回路(PMD)および A/D 変換器(ADC)と直接データの受け渡しができます。

モータ制御回路(PMD)レジスタの PMDxCMPU, PMDxCMPV, PMDxCMPW, PMDxMDOOUT, PMDxTRGCMP0, PMDxTRGCMP1, PMDxTRGSEL は PMDxMODESEL レジスタで VE モードに設定するとベクトルエンジンレジスタの VExCMPU, VExCMPV, VExCMPW, VExOUTCR, VExTRGCMP0, VExTRGCMP1, VExTRGSEL に切り替わります。この場合、CPU からモータ制御回路(PMD)の該当レジスタの書き込みによる制御はできず、ベクトルエンジンからの書き込みで制御します。その他の PMD のレジスタについては読み書き制限はありません。

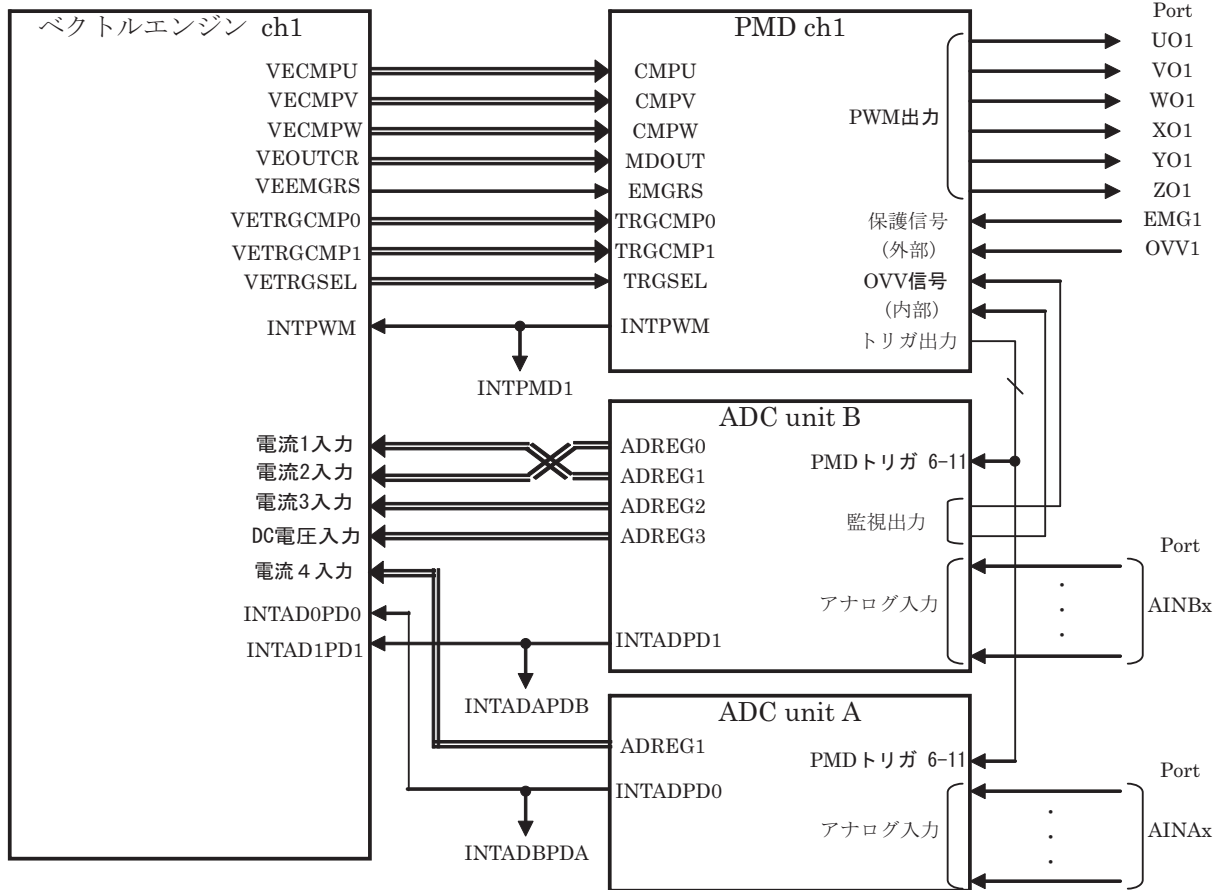
ベクトルエンジンは A/D 変換器(ADC)の変換結果レジスタ (ADxREG0, ADxREG1, ADxREG2, ADxREG3) の値を入力処理タスクで読み込むことができます。また、2 ユニットの ADC から変換結果を読み込むことができ、2 相電流の同時サンプリングに対応します。変換結果を読み込む際には、PMD からの同期トリガ毎にプログラム設定されている相情報も ADC から読み込みます。



18-3

A/D

( 0 )



18-4

A/D

( 1)

### 18.3

ベクトルエンジンはVE制御レジスタと専用レジスタが存在します。

- ・ VE制御レジスタ
  - ベクトルエンジン制御用レジスタおよびテンポラリレジスタ
- ・ 専用レジスタ
  - 演算データおよび演算制御レジスタ

#### 18.3.1

VE

			Address
/	VExEN	R/W	0x0000
CPU	VExCPURUNTRG	W	0x0004
	VExTASKAPP	R/W	0x0008
	VExACTSCH	R/W	0x000C
	VExREPTIME	R/W	0x0010
	VExTRGMODE	R/W	0x0014
	VExERRINTEN	R/W	0x0018
	VExCOMPEND	W	0x001C
	VExERRDET	R	0x0020
	VExSCHTASKRUN	R	0x0024
Reserved	-	R	0x0028
0	VExTMPREG0	R/W	0x002C
1	VExTMPREG1	R/W	0x0030
2	VExTMPREG2	R/W	0x0034
3	VExTMPREG3	R/W	0x0038
4	VExTMPREG4	R/W	0x003C
5	VExTMPREG5	R/W	0x0040
Reserved	-	R	0x0214

) "Reserved"

			Address
/	VExMCTLF	R/W	0x0044
	VExMODE	R/W	0x0048
	VExFMODE	R/W	0x004C
PWM	$(\text{PWM [s]} \times (1) \times 2^{16})$	R/W	0x0050
	$(\text{[Hz]} \div (1) \times 2^{15})$	R/W	0x0054
	$(\text{[deg]/360} \times 2^{16})$	R/W	0x0058
d	$(\text{[A]} \div (2) \times 2^{15})$	R/W	0x005C
q	$(\text{[A]} \div (2) \times 2^{15})$	R/W	0x0060
d	$(\text{[V]} \div (3) \times 2^{31})$	R/W	0x0064
q	$(\text{[V]} \div (3) \times 2^{31})$	R/W	0x0068
d	PI	R/W	0x006C
d	PI	R/W	0x0070

				Address	
q	PI		VExCIQKI	R/W	0x0074
q	PI		VExCIQKP	R/W	0x0078
d	(VDI 32bit)		VExVDIH	R/W	0x007C
d	(VDI 32bit)		VExVDILH	R/W	0x0080
q	(VQI 32bit)		VExVQIH	R/W	0x0084
q	(VQI 32bit)		VExVQILH	R/W	0x0088
PWM			VExFPWMCHG	R/W	0x008C
PWM	(PMD PWM )		VExMDPRD	R/W	0x0090
			VExMINPLS	R/W	0x0094
			VExTRGCRC	R/W	0x0098
DC	( [V] + ( 3) ×2 <sup>15</sup> )		VExVDCL	R/W	0x009C
THETA	(Q15 )		VExCOS	R/W	0x00A0
THETA	(Q15 )		VExSIN	R/W	0x00A4
	(Q15 )		VExCOSM	R/W	0x00A8
	(Q15 )		VExSINM	R/W	0x00AC
			VExSECTOR	R/W	0x00B0
			VExSECTORM	R/W	0x00B4
a	( 4)		VExIAO	R/W	0x00B8
b	( 4)		VExIBO	R/W	0x00BC
c	( 4)		VExICO	R/W	0x00C0
a	ADC ( 4)		VExIAADC	R/W	0x00C4
b	ADC ( 4)		VExIBADC	R/W	0x00C8
c	ADC ( 4)		VExICADC	R/W	0x00CC
	( [V] + ( 3) ×2 <sup>15</sup> )		VExVDC	R/W	0x00D0
d	( [A] + ( 2) ×2 <sup>31</sup> )		VExID	R/W	0x00D4
q	( [A] + ( 2) ×2 <sup>31</sup> )		VExIQ	R/W	0x00D8
Reserved			-	R/W	0x0174
AD			VExTADC	R/W	0x0178
PMD : U PWM			VExCMPU	R/W	0x017C
PMD : V PWM			VExCMPV	R/W	0x0180
PMD : W PWM			VExCMP	R/W	0x0184
PMD : 6			VExOUTCR	R/W	0x0188
PMD :	(TRGCMP0)		VExTRGCMP0	R/W	0x018C
PMD :	(TRGCMP1)		VExTRGCMP1	R/W	0x0190
PMD :			VExTRGSEL	R/W	0x0194
PMD : EMG			VExEMGRS	W	0x0198
PI			VExPIOLIM	R/W	0x01BC
PI d			VExCIDKG	R/W	0x01C0
PI q			VExCIQKG	R/W	0x01C4
			VExVSLIM	R/W	0x01C8
			VExVDQ	R/W	0x01CC
			VExDELTA	R/W	0x01D0
			VExCPHI	R/W	0x01D4
q			VExCLD	R/W	0x01D8
d			VExCLQ	R/W	0x01DC
			VExCR	R/W	0x01E0
			VExCPHIG	R/W	0x01E4
			VExCLG	R/W	0x01E8

			Address
	VExCRG	R/W	0x01EC
d	VExVDE	R/W	0x01F0
q	VExVQE	R/W	0x01F4
	VExDTC	R/W	0x01F8
	VExHYS	R/W	0x01FC
	VExDTCS	R/W	0x0200
PWM	VExPWMMAX	R/W	0x0204
PWM	VExPWMMIN	R/W	0x0208
	VExTHTCLP	R/W	0x020C

- 1) "Reserved"
- 2) : [Hz]
- 3) : A/D 1LSB [A] × 2<sup>11</sup>
- 4) : A/D 1LSB (VDC) [V] × 2<sup>12</sup>
- 5) A/D 16bit 12bit

18.3.2 VE

18.3.2.1 VExEN( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	VEIDLEN	VEEN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-2		R	"0"
1	VEIDLEN	R/W	IDLE 0: 1:
0	VEEN	R/W	0: 1:

) (VExEN = "0")

18.3.2.2 VExCPURUNTRG(CPU )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	VCPURT	W	0: - 1: VExTASKAPP<VTASKA> VExTASKAPP, VExACTSCH VExREPTIME

- 1) "1"
- 2)

0  
VExCOMPEND

VExCPURUNTRG

18.3.2.3 VExTASKAPP( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	VITASK			
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	VTASK			
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-12		R	"0"
11-8	VITASK[3:0]	R/W	0x0 : 1 ( 0) 0x1 : ( 1) 0x2 : 1 ( 2) 0x3 : ( 3) 0x4 : ( 4) 0x5 : ( 5) 0x6 : SIN/COS ( 6) 0x7 : ( 7) 0x8 : 1[SVM] ( 8) 0x9 : 2 ( 9) 0xA : 2 ( 10) 0xB : 2[I-Clarke] ( 11) 0xC : ATAN ( 12) 0xD : SQRT ( 13) 0xE,0xF : Reserved
7-4		R	"0"
3-0	VTASK[3:0]	R/W	0x0 : 1 ( 0) 0x1 : ( 1) 0x2 : 1 ( 2) 0x3 : ( 3) 0x4 : ( 4) 0x5 : ( 5) 0x6 : SIN/COS ( 6) 0x7 : ( 7) 0x8 : 1[SVM] ( 8) 0x9 : 2 ( 9) 0xA : 2 ( 10) 0xB : 2[I-Clarke] ( 11) 0xC : ATAN ( 12) 0xD : SQRT ( 13) 0xE,0xF : Reserved

)

18.3.2.4 VExACTSCH( )

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

Bit	Bit Symbol	Type	
31-4		R	"0"
3-0	VACT[3:0]	R/W	0x0 : 0x1 : 1 0x2 : 2 0x3 : 3 0x4 : 4 0x5 : 5 0x6 : 6 0x7 : 7 0x8 : 8 0x9 : 9 0xA : 10 0xB : 11 0xC : 12 0xD : 13 0xE : 14 0xF : 15
			18-4

18.3.2.5 VExREPTIME( )

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

Bit	Bit Symbol	Type	
31-4		R	"0"
3-0	VREP[3:0]	R/W	0: 1-15:

) "0"

18.3.2.6 VExTRGMODE( )

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

Bit	Bit Symbol	Type	
31-2		R	"0"
1-0	VTRG[1:0]	R/W	AD 00: 01: INTADAPDA ( ADCA PMD0 ) 10: INTADBPDB ( ADCB PMD1 ) 11: Reserved

18.3.2.7 VExERRINTEN( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	INTTEN	-	VERREN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-3		R	"0"
2	INTTEN	R/W	0: 1:
1		R	"0"
0	VERREN	R/W	0: 1: "1" ( ) PWM

18.3.2.8 VExCOMPEND(VE )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	VCEND	W	0: - 1: "1" "0"

18.3.2.9 VExERRDET( )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	VERRD	R	0: 1:  ( ) PWM "1"

18.3.2.10 VExSCHTASKRUN( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	VRTASK				VRSCH
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-5		R	"0"
4-1	VRTASK[3:0]	R	0x0: 1 ( 0) 0x1: ( 1) 0x2: 1 ( 2) 0x3: ( 3) 0x4: ( 4) 0x5: ( 5) 0x6: SIN/COS ( 6) 0x7: ( 7) 0x8: 1 [SVM] ( 8) 0x9: 2 ( 9) 0xA: 2 ( 10) 0xB: 2[I-Clarke] ( 11) 0xC: ATAN ( 12) 0xD: SQRT ( 13) 0xE: Reserved 0xF: Reserved
0	VRSCH	R	0: 1:

18.3.2.11 VExTMPREG0 ~ 5( )

VExTMPREG0

	31	30	29	28	27	26	25	24
bit symbol	TMPREG0							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG0							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG0							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	TMPREG0[31:0]	R/W	0

VExTMPREG1

	31	30	29	28	27	26	25	24
bit symbol	TMPREG1							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG1							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG1							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	TMPREG1[31:0]	R/W	1

VExTMPREG2

	31	30	29	28	27	26	25	24
bit symbol	TMPREG2							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG2							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG2							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG2							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	TMPREG2[31:0]	R/W	2

VExTMPREG3

	31	30	29	28	27	26	25	24
bit symbol	TMPREG3							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG3							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG3							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG3							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	TMPREG3[31:0]	R/W	3

VExTMPREG4

	31	30	29	28	27	26	25	24
bit symbol	TMPREG4							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG4							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG4							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG4							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	TMPREG4[31:0]	R/W	4

VExTMPREG5

	31	30	29	28	27	26	25	24
bit symbol	TMPREG5							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	TMPREG5							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	TMPREG5							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	TMPREG5							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	TMPREG5[31:0]	R/W	5

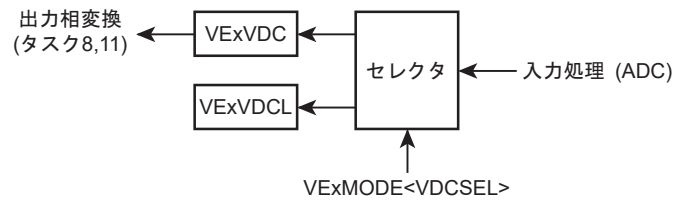
18.3.3

18.3.3.1 VExMODE( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	IPDEN	PMDDTCCEN	PWMFLEN	PWMBLEN	NICEN	T5ECEN	AWUMD	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	CLPEN	ATANMD		VDCSEL	OCRMD		ZIEN	PVIEN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15	IPDEN	R/W	2, 10 0: 1: 18.4.2.6
14	PMDDTCCEN	R/W	0, 9 PMD 0: PMD 1: PMD 18.4.2.4
13	PWMFLEN	R/W	0, 9 PWM 100% 0: 100% 1: 100% 18.4.2.4
12	PWMBLEN	R/W	0, 9 PWM 0% 0: 0% 1: 0% 18.4.2.4
11	NICEN	R/W	5 0: 1: 18.4.2.1 ( 5)
10	T5ECEN	R/W	5 ( , ) 0: 1: 18.4.2.1 ( 5)
9-8	AWUMD[1:0]	R/W	5 PI (AWU) 00: AWU 01: +4 10: +2 11: 18.4.2.1 ( 5)
7	CLPEN	R/W	6 0: 1: 18.4.2.2 SIN/COS ( 6)

Bit	Bit Symbol	Type	
6-5	ATANMD[1:0]	R/W	4 ATAN 0x : 10 : d-q 11 : d-q 18.4.2.7 ( / )
4	VDCSEL	R/W	2 10 0 : VExVDC 1 : VExVDCL 18-5 VExVDC/VExVDCL
3-2	OCRMD[1:0]	R/W	0 9 00 : 01 : 10 : ( ) 11 : EMG ( ) 18.4.2.4
1	ZIEN	R/W	2 0 : 1 : 18.4.2.6
0	PVIEN	R/W	6 0 : 1 : 18.4.2.2 SIN/COS ( 6)



注) VExVDCレジスタで制御される電源電圧に補正した値を使用する場合、保存先にVExVDCLを選択し、補正値をVExVDCレジスタにセットしてください。

### 18-5 VExVDC/VExVDCL

18.3.3.2 VExFMODE( )

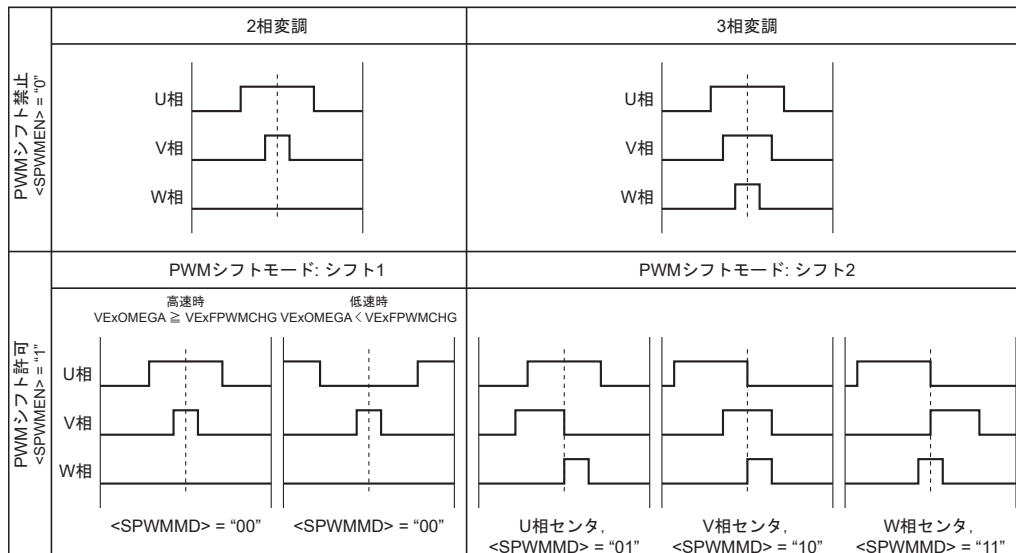
	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	SPWMMMD		SADCEN	PHCVDIS	VSLIMMD		MREGDIS	CRCEN
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ICPLMD	IBPLMD	IAPLMD	-	IDMODE		SPWMEN	C2PEN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-14	SPWMMMD[1:0]	R/W	PWM 00: 1 01: 2 (U ) 10: 2 (V ) 11: 2 (W ) 18-6 PWM ) 1( 0) 2 ( )
13	SADCEN	R/W	2 ADC 0: 1:
12	PHCVDIS	R/W	0: 2-3 (3 ) 1: 2-3 (2 ) ) ( 8) ( )
11-10	VSLIMMD [1:0]	R/W	5 00: ( ) 01: d 10: q 11: dq
9	MREGDIS	R/W	SIN/COS/SECTOR 0: 1: VExSINM = VExSIN VExCOSM = VExCOS VExSECTORM = VExSECTOR
8	CRCEN	R/W	0: 1: ) ( 1) 1 1
7	ICPLMD	R/W	10 Ic 0: ( Ic = VExICO - VExICADC ) 1: ( Ic = VExICADC - VExICO )
6	IBPLMD	R/W	10 Ib 0: ( Ib = VExIBO - VExIBADC ) 1: ( Ib = VExIBADC - VExIBO )
5	IAPLMD	R/W	10 Ia 0: ( Ia = VExIAO - VExIAADC ) 1: ( Ia = VExIAADC - VExIAO )
4		R/W	"0"

Bit	Bit Symbol	Type	
3-2	IDMODE[1:0]	R/W	00: 3 ( 1) 01: 2 ( 2) 10: 1 (PMD TRG ( 3 4)) 11: 1 (PMD TRG ( 3 4)) 1) 2( 10) 3 2) 2( 10) 2 4) 2( 9) 2( 10) PWM 2
1	SPWMEN	R/W	PWM 0: 1: ) 1( 0) 2 1( 2) 1 2( 9) 2( 10) 2
0	C2PEN	R/W	0: 3 1: 2

3) 1 PMDTRG

VExFMODE <IDMODE[1:0]>	PMDxTRGCR <TRG0MD[2:0]>	PMDxTRGCR <TRG1MD[2:0]>
10	010 (up-count)	010 (up-count)
10	101 (carrier bottom)	010 (up-count)
11	001 (down-count)	001 (down-count)
11	001 (down-count)	101 (carrier bottom)



注1) シフト1は2相変調時のみ選択可能です。  
 注2) シフト2は3相変調時のみ選択可能です。  
 注3) シフト2はPMDの設定が必要です。

18.3.3.3 VExTPWM(PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TPWM[15:0]	R/W	PWM $0x0000 \sim 0xFFFF$ ( PWM [s] $\times$ Max_Hz $\times 2^{16}$ ) PWM (Max_Hz : ) SIN/COS ( 6)

18.3.3.4 VExOMEGA( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	OMEGA[15:0]	R/W	$0x8000 \sim 0x7FFF \left( \frac{\text{[Hz]} \div \text{Max\_Hz} \times 2^{15}}{\text{Max\_Hz}} \right)$ <p>SIN/COS ( 6 )                      1( 0 ) 1 PWM 1                      ( 5), ( 4)</p>

18.3.3.5 VExTHETA( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	THETA[15:0]	R/W	$16$ $: [deg] \div 360 \times 2^{15}$ SIN/COS ( 6) SIN/COS ( 6)

18.3.3.6 VExCOS/VExSIN/VExCOSM/VExSINM(SIN/COS )

VExCOS

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	COS[15:0]	R/W	THETA            16bit            -1.0 ~ 1.0 0x8000 ~ 0x7FFF  SIN/COS    (    6) (    7)

VExSIN

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	SIN[15:0]	R/W	THETA            16bit            -1.0 ~ 1.0 0x8000 ~ 0x7FFF  SIN/COS    (    6) (    7)

VExCOSM

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	COSM[15:0]	R/W	VExCOS 0x8000 ~ 0x7FFF  SIN/COS ( 6) ( 4)

VExSINM

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	SINM[15:0]	R/W	VExSIN 0x8000 ~ 0x7FFF  SIN/COS ( 6) ( 4)

18.3.3.7 VExIDREF/VExIQREF(d /q )

VExIDREF

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	IDREF[15:0]	R/W	$d$ 16bit $-1.0 \sim 1.0$ $0x8000 \sim 0x7FFF$ ( $d$ $[A] \div \text{Max}_I \times 2^{15}$ ) $\text{Max}_I$ : (A/D 1LSB $[A] \times 2^{11}$ ) ( 5)

VExIQREF

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	IQREF[15:0]	R/W	$q$ 16bit $-1.0 \sim 1.0$ $0x8000 \sim 0x7FFF$ ( $q$ $[A] \div \text{Max}_I \times 2^{15}$ ) $\text{Max}_I$ : (A/D 1LSB $[A] \times 2^{11}$ ) ( 5)

18.3.3.8 VExVD/VExVQ(d /q )

VExVD

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

Bit	Bit Symbol	Type	
31-0	VD[31:0]	R/W	d 32bit -1.0 ~ 1.0 $0x8000\_0000 \sim 0x7FFF\_FFFF : d [V] \div \text{Max\_V} \times 2^{31}$ $\text{Max\_V: (A/D 1LSB [V])} \times 2^{12}$ ( 5) ( 7) ( 4)

VExVQ

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

Bit	Bit Symbol	Type	
31-0	VQ[31:0]	R/W	q 32bit -1.0 ~ 1.0 $0x8000\_0000 \sim 0x7FFF\_FFFF : q [V] \div \text{Max\_V} \times 2^{31}$ $\text{Max\_V: (A/D 1LSB [V])} \times 2^{12}$ ( 5) ( 7) ( 4)

18.3.3.9 VExCIDKI/VExCIDKP/VExVCIQKI/VExCIQKP(PI )

VExCIDKI

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CIDKI[15:0]	R/W	d PI 0x8000 ~ 0x7FFF

VExCIDKP

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CIDKP[15:0]	R/W	d PI 0x8000 ~ 0x7FFF

VExCIQKI

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CIQKI[15:0]	R/W	q PI 0x8000 ~ 0x7FFF

VExCIQKP

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CIQKP[15:0]	R/W	q PI 0x8000 ~ 0x7FFF

18.3.3.10 VExVDIH/VExVDILH/VExVQIH/VExVQILH(PI )

VExVDIH

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

Bit	Bit Symbol	Type						
31-0	VDIH[31:0]	R/W	d	PI	(VDI)	32bit		

VExVDILH

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

Bit	Bit Symbol	Type						
31-16	VDILH[15:0]	R/W	d	PI	(VDI)	16bit		
15-0		R	"0"					

- 1) VDI 64bit ( 63bit -1.0 ~ 1.0)
- 2) VDI 48bit

VExVQIH

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

Bit	Bit Symbol	Type						
31-0	VQIH[31:0]	R/W	q	PI	(VQI)	32bit		

VExVQILH

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

Bit	Bit Symbol	Type						
31-16	VQILH[15:0]	R/W	q	PI	(VQI)	16bit		
15-0		R	"0"					

- 1) VQI 64bit ( 63bit -1.0 ~ 1.0)
- 2) VQI 48bit

18.3.3.11 VExMCTLF( / )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	PWMOVF	VSOVF	PIQOVF	PIDOVF
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	PLSLFM	PLSLF	-	LVTF	LAVFM	LAVF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-12		R/W	"0"
11	PWMOVF	R/W	PWM 0:3 PWM PWMMIN PWMMAX 1:3 PWM PWMMIN PWMMAX ( 0,9)
10	VSOVF	R/W	0: ≤ VSLIM 1: > VSLIM ( 5)
9	PIQOVF	R/W	q PI 0:q PI ≤ PIOLIM 1:q PI > PIOLIM ( 5)
8	PIDOVF	R/W	d PI 0:d PI ≤ PIOLIM 1:d PI > PIOLIM ( 5)
7-6		R/W	"0"
5	PLSLFM	R/W	<PLSLF> ( 0,9)
4	PLSLF	R/W	1( 0) 1 < VExMINPLS<MINPLS> "1" 2( 9) < VExMINPLS<MINPLS> "1"
3		R/W	"0"
2	LVTF	R/W	VExVDC<VDC> 1/128 "0" VExVDC<VDC> < 1/128 "1" ( 8, 11)
1	LAVFM	R/W	<LAVF> 1( 0) 1 PWM

Bit	Bit Symbol	Type	
0	LAVF	R/W	<p>1( 0) 1 PWM</p> <p>0:</p> <p>1:</p> <p>VExOMEGA&lt;OMEGA&gt; VExFPWMCHG&lt;FPWMCHG&gt; "0"</p> <p>VExOMEGA&lt;OMEGA&gt; &lt; VExFPWMCHG&lt;FPWMCHG&gt; "1"</p>

18.3.3.12 VExFPWMCHG(PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	FPWMCHG[15:0]	R/W	PWM PWM 0x0000 ~ 0x7FFF ( [Hz] ÷ Max_Hz × 2 <sup>15</sup> ) (Max_Hz : [Hz]) 0 1 PWM 1

18.3.3.13 VExMDPRD(PWM )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VMDPRD[15:0]	R/W	PWM PMD PMDxMDPRD ( 0 9 ) ( 1 )

18.3.3.14 VExMINPLS( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	MINPLS[15:0]	R/W	$\frac{1}{(V_{ExCMPU}, V_{ExCMPV}, V_{ExCMPW}) \text{ Duty}} \times \frac{1}{PWM} \times (3 \times PWM)$ $[s] \div PWM \quad [s]$ $\frac{2}{(9)} \times (3 \times PWM(V_{ExCMPU}, V_{ExCMPV}, V_{ExCMPW}) \text{ Duty})$ $[s] \div PWM \quad [s]$

18.3.3.15 VExSECTOR/VExSECTORM( )

VExSECTOR

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

Bit	Bit Symbol	Type	
31-4		R	"0"
3-0	SECTOR[3:0]	R/W	: 0x0 ~ 0xB 30 12 ( 8 11) 1( 0)

VExSECTOR

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

Bit	Bit Symbol	Type	
31-4		R	"0"
3-0	SECTORM[3:0]	R/W	: 0x0 ~ 0xB ( 8 11) 1( 2)

18.3.3.16 VExIAO/VExIBO/VExICO( )

VExIAO

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	IAO[15:0]	R/W	a ADC ( a ADC ) 1( 2) ADC <IAO[15:4]> <IAO[3:0]> "0"

VExIBO

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	IBO[15:0]	R/W	b ADC ( b ADC ) 1( 2) ADC <IBO[15:4]> <IBO[3:0]> "0"

VExICO

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	ICO[15:0]	R/W	c          ADC          (          c          ADC          )  1(          2) ADC          <ICO[15:4]>          <ICO[3:0]>          "0"

18.3.3.17 VExIAADC/VExIBADC/VExICADC( ADC )

VExIAADC

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	IAADC[15:0]	R/W	a ADC 0x0000 ~ 0xFFFF ( 2 10) ADC <IAADC[15:4]> <IAADC[3:0]> "0"

VExIBADC

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	IBADC[15:0]	R/W	b ADC 0x0000 ~ 0xFFFF ( 2 10) ADC <IBADC[15:4]> <IBADC[3:0]> "0"

VExICADC

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	ICADC[15:0]	R/W	c ADC 0x0000 ~ 0xFFFF  ( 2 10) ADC <ICADC[15:4]> <ICADC[3:0]> "0"

18.3.3.18 VExVDC/VExVDCL( )

VEVDCx

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VDC[15:0]	R/W	,16bit : 0 ~ 1.0 : 0x0000 ~ 0x7FFF $\text{VDC} \times \text{Max\_V} \div 2^{15}$ (Max_V : (A/D 1LSB [V]) × 2 <sup>12</sup> ) VExMODE<VDCSEL> = "0" ( 2 10) ( 8 11)

VExVDCL

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VDCL[15:0]	R/W	,16bit : 0 ~ 1.0 : 0x0000 ~ 0x7FFF $\text{VDCL} \times \text{Max\_V} \div 2^{15}$ (Max_V : (A/D 1LSB [V]) × 2 <sup>12</sup> ) VExMODE<VDCSEL> = "1" ( 2 10)

18.3.3.19 VExID/VExIQ(d /q )

VExID

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

Bit	Bit Symbol	Type	
31-0	ID[31:0]	R/W	d 32bit : -1.0 ~ 1.0 : 0x8000_0000 ~ 0x7FFF_FFFF $ID \times \text{Max}_I \div 2^{31}$ (Max_I : (A/D 1LSB [A] × 2 <sup>11</sup> ) ( 4 ) ( 5 )

VExIQ

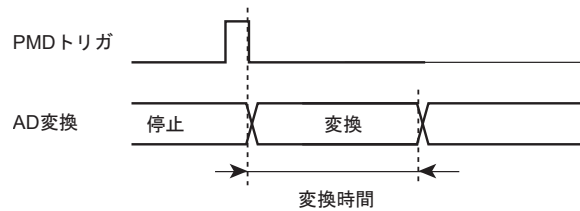
	31	30	29	28	27	26	25	24
bit symbol	IQ							
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	IQ							
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	IQ							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	IQ							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-0	IQ[31:0]	R/W	q 32bit : -1.0 ~ 1.0 : 0x8000_0000 ~ 0x7FFF_FFFF $IQ \times \text{Max}_I \div 2^{31}$ (Max_I : (A/D 1LSB [A] × 2 <sup>11</sup> ) ( 4 ) ( 5 )

18.3.3.20 VExTADC(ADC )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TADC[15:0]	R/W	ADC ( 18-7 ) 0x0000~0xFFFF : (ADC [s]+PWM [s] ) 1 PWM 1 ( 1 " 1" 0



18-7 ADC

18.3.3.21 VExCMPU/ VExCMPV/ VExCMPW(PWM DUTY )

VExCMPU

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VCMPU[15:0]	R/W	U PWM : 0x0000 ~ 0xFFFF  ( 0 9) ( 1)

VExCMPV

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VCMPV[15:0]	R/W	V PWM : 0x0000 ~ 0xFFFF  ( 0 9) ( 1)

VExCMPW

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VCMPW[15:0]	R/W	W PWM : 0x0000 ~ 0xFFFF  ( 0 9) ( 1)

18.3.3.22 VExOUTCR(6 )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	WPWM
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	VPWM	UPWM	WOC		VOC		UOC	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-9		R	"0"
8	WPWM	R/W	W PWM 0: / 1: PWM
7	VPWM	R/W	V PWM 0: / 1: PWM
6	UPWM	R/W	U PWM 0: / 1: PWM
5-4	WOC[1:0]	R/W	W 00: WO ,ZO ( ) 01: WO ,ZO 10: WO ,ZO 11: WO ,ZO ( )<WPWM> = 1
3-2	VOC[1:0]	R/W	V 00: VO ,YO ( ) 01: VO ,YO 10: VO ,YO 11: VO ,YO ( )<VPWM> = 1
1-0	UOC[1:0]	R/W	U 00: UO ,XO ( ) 01: UO ,XO 10: UO ,XO 11: UO ,XO ( )<UPWM> = 1

) ( 0 9)

PMD の U,V,W 相の出力制御を以下に示します。(VE で使用する組み合わせのみ表示)

18-1 <UPWM>,<UOC> PMD : U (UO,XO)

<UPWM>	<UOC>	UO	XO
0	00		
1	00	PWMU	PWMU
1	11	PWMU	PWMU

18-2 <VPWM>,<VOC> PMD : V (VO,YO)

<VPWM>	<VOC>	VO	YO
0	00		
1	00	PWMV	PWMV
1	11	PWMV	PWMV

18-3 <WPWM>,<WOC> PMD : W (WO,ZO)

<WPWM>	<WOC>	WO	ZO
0	00		
1	00	PWMW	PWMW
1	11	PWMW	PWMW

18.3.3.23 VExTRGCRC( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	TRGCRC[15:0]	R/W	$1 \quad [s] \div \text{PWM} \quad [s]$ $\text{PWM} \quad 1$ ( 1)

18.3.3.24 VExTRGCMP0/VExTRGCMP1( )

VExTRGCMP0

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	VTRGCMP0							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	VTRGCMP0							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VTRGCMP0[15:0]	R/W	ADC PMD (PMD ) 0x0000 : 0x0001 ~ (VExMDPRD - 1): VExMDPRD ~ 0xFFFF : PMD ( 1) 1 PWM 1

VExTRGCMP1

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	VTRGCMP1							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	VTRGCMP1							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VTRGCMP1[15:0]	R/W	ADC PMD (PMD ) 0x0000 : 0x0001 ~ (VExMDPRD - 1): VExMDPRD ~ 0xFFFF :  PMD  PMD (PMDxTRGMD<TRGOUT> = "1") ( 1) 1 PWM 1

18.3.3.25 VExTRGSEL( )

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

Bit	Bit Symbol	Type	
31-3		R	"0"
2-0	VTRGSEL[2:0]	R/W	VExTRGCMP0 (PMD ) 0 ~ 5 : 6 ~ 7 : PMD (PMDxTRGMD<TRGOUT> = "1") ( 1) VEx SECTOR + 2

18.3.3.26 VExEMGRS(EMG )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	EMGRS	W	EMG (PMD ) 0: - 1: EMG "1" EMG ( 0 9) "1" "0"

18.3.3.27 VExPIOLIM(PI )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	PIOLIM[15:0]	R/W	PI : 0x0000 ~ 0x7FFF ( )

18.3.3.28 VExCIDKG/VExCIQKG(PI d /q )

VExCIDKG

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	CIDKG[7:0]	R/W	PI d 0x00 : 1/1 0x01 : 1/2 <sup>4</sup> 0x02 : 1/2 <sup>8</sup> 0x03 : 1/2 <sup>12</sup> 0x04 : 1/2 <sup>16</sup> 0x05 ~ 0xFF : Reserved  ( )

VExCIQKG

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	CIQKG[7:0]	R/W	PI q 0x00 : 1/1 0x01 : 1/2 <sup>4</sup> 0x02 : 1/2 <sup>8</sup> 0x03 : 1/2 <sup>12</sup> 0x04 : 1/2 <sup>16</sup> 0x05 ~ 0xFF : Reserved  ( )

18.3.3.29 VExVSLIM( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VSLIM[15:0]	R/W	d (VExVD) q (VExVQ) 0x0000 ~ 0x7FFF : [V] + Max_V × 2 <sup>15</sup> Max_V: (A/D 1LSB [V]) × 2 <sup>12</sup>  0x0000 ( )

18.3.3.30 VExVDQ( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VDQ[15:0]	R/W	$d$ (VExVD) $q$ (VExVQ) 0x0000 ~ 0x7FFF : $[V] \div \text{Max\_V} \times 2^{15}$ Max_V: (A/D 1LSB $[V] \times 2^{12}$ ) ( )

18.3.3.31 VExDELTA( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	DELTA[15:0]	R/W	d-q : 0x8000 ~ 0x7FFF(-180° ~ 180°) [deg] ÷ 360 × 2 <sup>16</sup>  ATAN ( )

18.3.3.32 VExCPHI/VExCLD/VExCLQ/VExCR/VExCPHIG/VExCLG/VExCRG( )

VExCPHI

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CPHI[15:0]	R/W	$[V/rps] ( [Wb/s] )$ $: 0x0000 \sim 0x7FFF$ $[V/rps] \div Max\_V \times Max\_Hz \times 2^{11} \div [CPHIG ]$ $( )$

VExCLD

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CLD[15:0]	R/W	$d$ $: 0x0000 \sim 0x7FFF$ $[H] \times \text{Max}_I \div \text{Max}_V \times \text{Max}_{Hz} \times 2\pi \times 2^{11} \div [\text{CLG} \quad ]$ $(\quad)$

VExCLQ

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CLQ[15:0]	R/W	$q$ $: 0x0000 \sim 0x7FFF$ $[H] \times \text{Max}_I \div \text{Max}_V \times \text{Max}_{Hz} \times 2\pi \times 2^{11} \div [\text{CLG} \quad ]$ $(\quad)$

VExCR

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	CR[15:0]	R/W	: 0x0000 ~ 0x7FFF $[\Omega] \times \text{Max\_I} + \text{Max\_V} \times 2^{11} + [\text{CRG} \quad ]$ (      )

VExCPHIG

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	CPHIG[7:0]	R/W	0x00 : 1/1 0x01 : 1/2 <sup>4</sup> 0x02 : 1/2 <sup>8</sup> 0x03 : 1/2 <sup>12</sup> 0x04 : 1/2 <sup>16</sup> 0x05 ~ 0xFF : Reserved (      )

VExCLG

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	CLG[7:0]	R/W	0x00 : 1/1 0x01 : 1/2 <sup>4</sup> 0x02 : 1/2 <sup>8</sup> 0x03 : 1/2 <sup>12</sup> 0x04 : 1/2 <sup>16</sup> 0x05 ~ 0xFF : Reserved  ( )

VExCRG

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	CRG[7:0]	R/W	0x00 : 1/1 ( CR                      Q27                      [31:16] ) 0x01 : 1/2 <sup>4</sup> 0x02 : 1/2 <sup>8</sup> 0x03 : 1/2 <sup>12</sup> 0x04 : 1/2 <sup>16</sup> 0x05 ~ 0xFF : Reserved  (                      )

18.3.3.33 VExVDE/VExVQE( d /q )

VExVDE

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VDE[15:0]	R/W	$d$ 0x8000 ~ 0x7FFF : $[V] \div \text{Max\_V} \times 2^{15}$ (Max_V : (A/D 1LSB $[V]) \times 2^{12}$ ) -1.0 ~ 1.0 ( )

VExVQE

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	VQE[15:0]	R/W	$q$ 0x8000 ~ 0x7FFF : $[V] \div \text{Max\_V} \times 2^{15}$ (Max_V : (A/D 1LSB $[V]) \times 2^{12}$ ) -1.0 ~ 1.0 ( )

18.3.3.34 VExDTC( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	DTC	R/W	0x0000 ~ 0xFFFF : [s] + PWM [s] × VExMDPRD ( )

18.3.3.35 VExHYS( )

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	HYS[15:0]	R/W	0x8000 ~ 0x7FFF : [A] ÷ Max_I × 2 <sup>15</sup> -1.0 ~ 1.0 (Max_I : (A/D 1LSB [A] × 2 <sup>11</sup> )) ( )

18.3.3.36 VExDTCS( / )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	ICSTS		
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	IBSTS			-	IASTS		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-11		R/W	"0"
10-8	ICSTS[1:0]	R/W	lc CMPW xx0 : x01 : DTC x11 : DTC
7		R/W	"0"
6-4	IBSTS[1:0]	R/W	lb CMPV xx0 : x01 : DTC x11 : DTC
3		R/W	"0"
2-0	IASTS[1:0]	R/W	la CMPU xx0 : x01 : DTC x11 : DTC  ( ) ( )

18.3.3.37 VExPWMMAX/VExPWMMIN(PWM )

VExPWMMAX

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	PWMMAX[15:0]	R/W	PWM : 0x0000 ~ 0xFFFF ( )

VExPWMMIN

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

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	PWMMIN[15:0]	R/W	PWM : 0x0000 ~ 0xFFFF ( )

18.3.3.38 VExTHTCLP( )

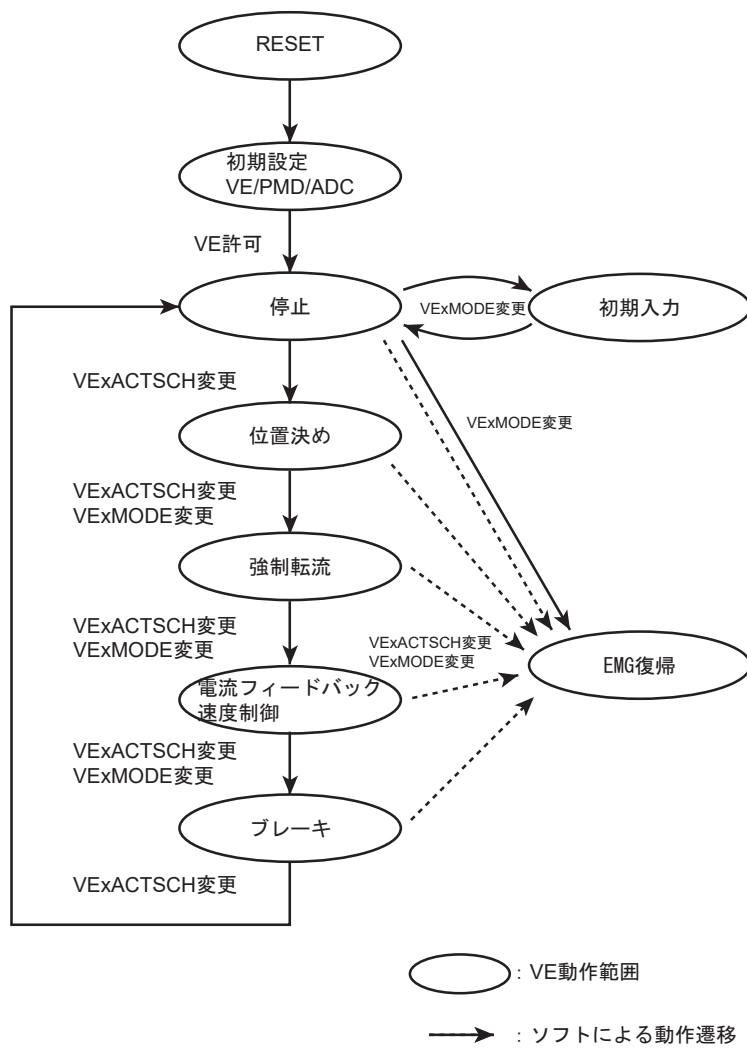
	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	THTCLP							
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	THTCLP							
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-16		R	"0"
15-0	THTCLP[15:0]	R/W	THETA : [deg] ÷ 360 × 2 <sup>15</sup> SIN/COS ( )

18.4

18.4.1

モータ制御は図 18-8 のようなフローで実行されます。ベクトルエンジンは各動作状態をスケジュール設定(VExACTSCH)とモード設定(VExMODE)を切り替えることで遷移させます。



18-8

RESET	:
	:
	:
	:
	:
	:
	:
	:
	:
EMG	: EMG

18.4.1.1

動作スケジュールは VExACTSCH レジスタで選択します。

スケジュールは出力処理を行う出力スケジュールと入力処理を行う入力スケジュールで構成されます。出力スケジュールは出力関連のタスク、入力スケジュールは入力関連のタスクで構成されます。スケジュールと動作するタスクの関連を表 18-4 に示します。

モータ制御方法に応じて、専用レジスタでタスク動作を設定してください。

18-4

VExACTSCH <VACT[3:0]>													ATAN	
		SIN/ COS		1	2	1	2		1	2				
	5	6	7	8	11	0	9	1	2	10	3	4	12	13
0:	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)
1: 1	o	o	o	o		o		o	o		o	o		
2: 2	o	o	o	o			o	o		o	o	o		
3: 3	o	o	o		o		o	o		o	o	o		
4: 4		o	o	o		o		o	o		o	o		
5: 5		o	o	o			o	o		o	o	o		
6: 6		o	o		o		o	o		o	o	o		
7: 7		o	o		o	o		o	o		o	o		
8: 8	o	o	o		o	o		o	o		o	o		
9: 9						o		o	o					
10: 10	o	o	o	o		o								
11: 11	o	o	o	o			o							
12: 12	o	o	o		o		o							
13: 13	o	o	o		o	o								
14: 14								o	o		o	o		
15: 15								o		o	o	o		

- 1) VExTASKAPP
- 2) o:                    -:

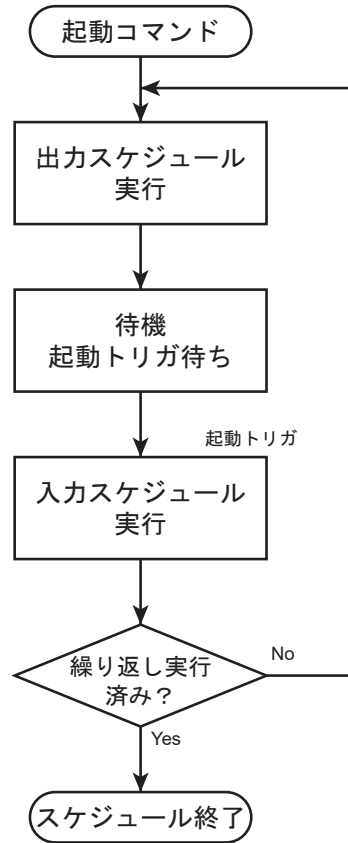
18-5

	VExACTSCH <VACT[3:0]>	VExTASKAPP <VTASK[3:0]>	VExMODE <PVIEN>	VExMODE <OCRMD[1:0]>	VExMODE <ZIEN>
	9	0	x	00	0
	9	0	x	00	1
	1	5	0	01	0
	1	5	1	01	0
	1	5	1	01	0
	4	6	0	01	0
EMG	9	0	x	11	0
	4	6	x	10	0

出力スケジュールはコマンド(VExCPURUNTRG)で動作開始し、出力関連タスクが全て終了するとベクトルエンジンは待機状態に移行して起動トリガ(VExTRGMODE 設定)を待ちます。

入力スケジュールは起動トリガで動作開始し、入力関連タスクが全て終了すると CPU に割り込みを発生してベクトルエンジンは休止状態に移行します。ただし、スケジュールの繰り返し回数 (VExREPTIME) を 2 回以上に設定している場合は、設定回数に達するまでは割り込みを発生せずに出カスケジュールを起動します。

) 10 ~ 15 (VExREPTIME ≥ 2 1 )



18-9

18.4.1.2

初めに、ベクトルエンジン許可(<VEEN> = "1")して、動作スケジュール選択レジスタ (VExACTSCH)、タスク指定レジスタ (VExTASKAPP)、および動作スケジュール繰り返し回数指定レジスタ (VExREPTIME)を設定後、下記の手順でスケジュールを実行してください。

ベクトルエンジンの動作スケジュールは出力スケジュールと入力スケジュールに分かれており、基本的に出力スケジュール実行後の待機状態で起動トリガが発生すると入力スケジュールが実行されます。

出力スケジュールと入力スケジュールはそれぞれ次の条件で起動します。

- ・ 出力スケジュールの起動
  1. コマンドスタート(VExCPURUNTRG)で指定タスク (VExTASKAPP)の開始
  2. 入力スケジュール終了後の繰り返しスタート(VExREPTIME ≥ 2)
- ・ 入力スケジュールの起動
  1. 出力スケジュール終了後の待機状態から起動トリガ(VExTRGMODE で選択したトリガ入力)による入力処理タスクの開始
  2. コマンドスタート(VExCPURUNTRG)で指定タスク (VExTASKAPP)の開始

18-6

VExACTSCH		0x0: VExTASKAPP 0x1: 1 0x2: 2 0x3: 3 0x4: 4 0x5: 5 0x6: 6 0x7: 7	0x8: 8 0x9: 9 0xA: 10 0xB: 11 0xC: 12 0xD: 13 0xE: 14 0xF: 15
VExTASKAPP			
VExREPTIME		1 ~ 15 ) 1	"1" "0"
VExTRGMD		ADC A ADC	B

### 18.4.1.3

ベクトルエンジンはスケジュール終了時に発生するベクトルエンジン割り込み (INTVCN<sub>x</sub>) と指定タスク終了時に発生するタスク終了割り込み (INTVCT<sub>x</sub>) があります。

- ・ ベクトルエンジン割り込み
  1. 動作スケジュール選択(VE<sub>x</sub>ACTSCH)でスケジュール指定しコマンドスタート(VE<sub>x</sub>CPURUNTRG = "1")します。
  2. 繰り返し回数指定(VE<sub>x</sub>REPTIME)した回数だけの選択スケジュールの実行終了時にINTVCN<sub>x</sub> 割り込み発生します。
  3. エラー検出割り込み制御を許可 (VE<sub>x</sub>ERRINTEN<VERREN> = "1") していると出力スケジュール実行中にPMD回路のPWM割り込みが発生するとINTVCN<sub>x</sub> 割り込みが発生し、エラーフラグ(VE<sub>x</sub>ERRDET<VERRD>)に"1"を設定します。
- ・ タスク終了割り込み
  1. タスク終了割り込み発生タスクを指定 (VE<sub>x</sub>TASKAPP<VITASK>) してタスク終了割り込み制御を許可 (VE<sub>x</sub>ERRINTEN<INTTEN> = "1") します。
  2. コマンドスタート(VE<sub>x</sub>CPURUNTRG = "1")でスケジュール開始し、<VITASK>設定のタスク終了時にINTVCT<sub>x</sub> 割り込み発生します。

18.4.2

スケジュールで動作する各タスクの概要を示します。

個別実行タスクや起動タスクの指定には表 18-7 のタスク番号を使用します。

18-7

		d /q PI (PI ) d /q ,	5
	SIN/COS	/ ( )	6
			7
	1	2 3 [SVM]	8
	2	2 3 [ ]	11
	1	PMD PWM 1 PWM	0
	2	PMD PWM 2 PWM	9
			1
	1	,3 PWM /PWM 1 1 ADC	2
	2	,3 PWM 2 1 ADC	10
		3 2	3
		d /q	4
	ATAN		12
	SQRT		13

18.4.2.1 ( 5)

電流制御タスクは、d 軸電流、q 軸電流それぞれに PI 制御演算を行います。PI 制御出力を制限することができます。

また、p 軸と q 軸を合わせて制御する、非干渉制御および電圧スカラ制限が可能です。

1. d 軸電流 PI 制御

<演算式>

```

[PI ]
CIDKP = VExCIDKP × [VExCIDKG ] :
CIDKI = VExCIDKI × [VExCIDKG ] :
Δ ID = VExIDREF - VExID :
VDI0 = CIDKI × Δ ID + VExVDI :
VD0 = CIDKP × Δ ID + VExVDI0 :
[PI ]
if ( VD0 > VExPIOLIM ) :
    VExVD = VExPIOLIM
    VExMCTLF<PIDOVF> = 1
else if ( VD0 < -VExPIOLIM ) :
    VExVD = -VExPIOLIM
    VExMCTLF<PIDOVF> = 1
else VExVD = VD0
[ (AWU) ]
Δ VD = VExVD - VD0 :
VExVDI = VDI0 + Δ VD × [VExMODE<AWUMD> ] :
    
```

	VExID	d	32bit ( 31bit)
	VExIDREF	d	16bit ( 15bit)
	VExCIDKP		16bit
	VExCIDKI		16bit
	VExCIDKG	d PI	000: 1 001: 1/2 <sup>4</sup> 010: 1/2 <sup>8</sup> 011: 1/2 <sup>12</sup> 100: 1/2 <sup>16</sup> 101 ~ 111: Reserved
	VExPIOLIM	PI	16bit ( 15bit ) 0x0 ~ 0x7FFF ) VExPIOLIM = 0
	VExMODE[9:8]		<AWUMD> 00: 01: 1/4 10: 1/2 11: 1
	VExVD	d	32bit ( 31bit)
	VExMCTLF[8]	d	<PIDOVF> 0: 1:
	VExVDI	d	64bit ( 63bit)

) VExVDI 64

VExVDIH

VExVDILH

2. q 軸電流 PI 制御

<演算式>

```

[PI ]
CIQKP = VExCIQKP × [VExCIQKG ] :
CIQKI = VExCIQKI × [VExCIQKG ] :
Δ IQ = VExIQREF - VExIQ :
VQI0 = CIQKI × Δ IQ + VExVQI :
VQ0 = CIQKP × Δ IQ + VQI0 :
[PI ]
if ( VQ0 > VExPIOLIM ) :
    VExVQ = VExPIOLIM
    VExMCTLF<PIQOVF> = 1
else if ( VQ0 < -VExPIOLIM ) :
    VExVQ = -VExPIOLIM
    VExMCTLF<PIQOVF> = 1
else VExVQ = VQ0
[ (AWU) ]
Δ VQ = VExVQ - VQ0 :
VExVQI = VQI0 + Δ VQ × [VExMODE<AWUMD> ] :
    
```

	VExIQ	q	32bit ( 31bit)
	VExIQREF	q	16bit ( 15bit)
	VExCIQKP		16bit
	VExCIQKI		16bit
	VExCIQKG	q PI	000: 1 001: 1/2 <sup>4</sup> 010: 1/2 <sup>8</sup> 011: 1/2 <sup>12</sup> 100: 1/2 <sup>16</sup> 101 ~ 111: Reserved
	VExPIOLIM	PI	16bit ( 15bit ) 0x0 ~ 0x7FFF ) VExPIOLIM = 0
	VExMODE[9:8]		<AWUMD> 00: 01: 1/4 10: 1/2 11: 1
	VExVQ	q	32bit ( 31bit)
	VExMCTLF[9]	q	<PIQOVF> 0: 1:
	VExVQI	q	64bit ( 63bit)

) VExVQI 64

VExVQIH

VExVQILH

3. 非干渉制御

モータの電圧方程式を使って d 軸,q 軸の干渉分を算出して PI 制御結果を補正します。

<演算式>

```

if (VExMODE[10] = 1) :
    LD = VExCLD × [VExCLG    ] : d
    LQ = VExCLQ × [VExCLG    ] : q
    PHI = VExCPHI × [VExCPHIG  ] :
    VExVDE = -VExOMEGA × VExIQ × LQ : d
    VExVQE = VExOMEGA × VExID × LD + VExOMEGA × PHI : q
if (VExMODE[11] = 1) :
    VExVD = VExVD + VExVDE
    VExVQ = VExVQ + VExVQE
    
```

	VExVD	d	32bit ( 31bit)
	VExVQ	q	
	VExID	d	
	VExIQ	q	
	VExCLD	d	16bit ( 11bit)
	VExCLQ	q	
	VExCPHI		
	VExCLG		000: 1 001: 1/2 <sup>4</sup> 010: 1/2 <sup>8</sup> 011: 1/2 <sup>12</sup>
	VExCPHIG		100: 1/2 <sup>16</sup> 101 ~ 111: Reserved
	VExOMEGA		16bit ( 15bit)
	VExMODE[11]		<NICEN> 0: 1:
	VExMODE[10]		<T5ECEN> 0: ( ) 1:
	VExVDE	d	16bit ( 15bit)
	VExVQE	q	
	VExVD	d	32bit ( 31bit)
	VExVQ	q	

## 4. 電圧スカラ制限

d 軸,q 軸電圧の合成値( $\sqrt{VD^2 + VQ^2}$  の平方根)が制限値を超えないように、d 軸,q 軸電圧を制限します。

<演算式>

```

if (VExMODE[10] = 1) :
  [VDQ ] :
  if (VExVD2 + VExVQ2 > VExVSLIM2) :
    if (VExFMODE[11:10] = 00) VExVDQ = SQRT(VD2 + VQ2) :
    else if (VExFMODE[11:10] = 01) VExVDQ = SQRT(VExVSLIM2 - VQ2) :d
    else if (VExFMODE[11:10] = 10) VExVDQ = SQRT(VExVSLIM2 - VD2) :q
    else if (VExFMODE[11:10] = 11) VExVDQ = SQRT(VD2 + VQ2) :dq
  ) SQRT :
  [ ] :
  X = | VExVQ |
  Y = | VExVD |
  VExDELTA = ATAN2(X, Y)
  ) ATAN2 :
  [ ] :
  if (VExFMODE[11:10] = 00) :
    VDLIM = VExVSLIM
    VQLIM = VExVSLIM
  else if (VExFMODE[11:10] = 01) :d
    VDLIM = VExVDQ
    VQLIM = VExVSLIM
  else if (VExFMODE[11:10] = 10) :q
    VDLIM = VExVSLIM
    VQLIM = VExVDQ
  else if (VExFMODE[11:10] = 11) :dq
    VDLIM = VExVSLIM × SIN(VExDELTA)
    VQLIM = VExVSLIM × COS(VExDELTA)
  [ ] :
  if (VExVD > VDLIM) VExVD = VDLIM :d
  VExMCTLF[10] = 1
  else if (VExVD < -VDLIM) VExVD = -VDLIM :d
  VExMCTLF[10] = 1
  if (VExVQ > VQLIM) VExVQ = VQLIM :q
  VExMCTLF[10] = 1
  else if (VExVQ < -VQLIM) VExVQ = -VQLIM :q

```

VExMCTLF[10] = 1

	VExVD	d	32bit ( 31bit)
	VExVQ	q	
	VExVSLIM		16bit ( 15bit ) 0x0 ~ 0x7FFF VExVSLIM = 0x0
	VExMODE[10]		<T5ECEN> 0: ( ) 1:
	VExFMODE[11:10]		<VSLIMMD> 00: 01: d 10: q 11: dq
	VExVDQ		16bit ( 15bit)
	VExDELTA		16bit 0x0000 ~ 0x4000 ( 0 ~ 90° )
	VExVD	d	32bit ( 31bit)
	VExVQ	q	
	VExMCTLF[10]		<VSOVF> 0: 1:

18.4.2.2 SIN/COS ( 6)

SIN/COS 演算タスクは、位相補間演算と SIN/COS 演算を実行します。

位相補間は回転速度を PWM 周期で積分して計算し、位相補間許可時( VExMODE<PVIEN>="1")のみ実行します。位相補間時は指定位相でのクリッピングが可能です。

1. 位相補間

<演算式>

```

THETA0 = VExOMEGA × VExTPWM + VExTHETA
THETA0 = THETA0 & 0x0000FFFF
if ( VExMODE[7] = 1)
    if ( VExOMEGA ≥ 0)
        if ( VExTHETA ≤ VExTHTCLP ≤ THETA0 ) THETA0 = THTCLP
        else if ( THETA0 ≤ VExTHETA ≤ VExTHTCLP ) THETA0 = THTCLP
        else if ( VExTHTCLP ≤ THETA0 ≤ VExTHETA ) THETA0 = THTCLP
    else if ( VExOMEGA < 0)
        if ( THETA0 ≤ VExTHTCLP ≤ VExTHETA ) THETA0 = THTCLP
        else if ( VExTHTCLP ≤ VExTHETA ≤ THETA0 ) THETA0 = THTCLP
        else if ( VExTHETA ≤ THETA0 ≤ VExTHTCLP ) THETA0 = THTCLP
if ( VExMODE[0] = 1 ) VExTHETA = THETA0
    
```

	VExTHETA	θ	16bit	(0.0 ~ 1.0	16bit)
	VExOMEGA		16bit	(-1.0 ~ 1.0	15bit)
	VExTPWM	PWM	16bit		
	VExTHTCLP		16bit	(0.0 ~ 1.0	16bit)
	VExMODE[0]		<PVIEN>	0:	1:
	VExMODE[7]		<CLPEN>	0:	1:
	VExTHETA	θ	16bit	(0.0 ~ 1.0	16bit)

2. SIN/COS 演算

<演算式>

```

VExSINM = VExSIN          : ( )
VExCOSM = VExCOS
VExSIN = SIN ( VExTHETA ) : SIN,COS
VExCOS = SIN ( VExTHETA + 1/4 )
if ( VExFMODE[9] = 1 ) :
    VExSINM = VExSIN
    VExCOSM = VExCOS
) SIN :
    
```

	VExTHETA	$\theta$	16bit (0.0 ~ 1.0 16bit)
	VExFMODE[9]	SIN,COS	<MREGDIS> 0: , 1:
	VExSIN	$\theta$	16bit (-1.0 ~ 1.0 15bit)
	VExCOS	$\theta$	
	VExSINM		
	VExCOSM		

18.4.2.3 ( / )

出力電圧変換は座標軸変換と相変換の2段階で変換します。

相変換タスクには空間ベクトル変換と逆クランクの2種類あります。

1. 出力座標軸変換(タスク 7)

出力座標軸タスクは d 軸電圧、q 軸電圧、sinθ、cosθ から α 軸電圧、β 軸電圧を算出します。

<演算式>

$$\begin{aligned} V_{\text{ExTMPREG3}} &= V_{\text{ExCOS}} \times V_{\text{ExVD}} - V_{\text{ExSIN}} \times V_{\text{ExVQ}} && :V_{\alpha} \\ V_{\text{ExTMPREG4}} &= V_{\text{ExSIN}} \times V_{\text{ExVD}} + V_{\text{ExCOS}} \times V_{\text{ExVQ}} && :V_{\beta} \end{aligned}$$

	VExVD	d	32bit	(-1.0 ~ 1.0	31bit)
	VExVQ	q	32bit	(-1.0 ~ 1.0	31bit)
	VExSIN	THETA	16bit	(-1.0 ~ 1.0	15bit)
	VExCOS	THETA	16bit	(-1.0 ~ 1.0	15bit)
	VExTMPREG3	α	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG4	β	32bit	(-1.0 ~ 1.0	31bit)

2. 出力相変換 1(空間ベクトル変換)(タスク 8)

出力相変換 1 は α 軸電圧、β 軸電圧からセクタ判定し、セクタ別に空間ベクトル変換で a 相電圧、b 相電圧、c 相電圧の Duty を算出します。このタスクは変換方式に 2 相変調か 3 相変調かを選択できます。

a. セクタ判定

<演算式>

```

VExSECTORM = VExSECTOR
Vα = VExTMPREG3
Vβ = VExTMPREG4
if ( Vα ≥ 0 & Vβ ≥ 0 )
    if ( |Vα| ≥ |Vβ| + sqrt(3) )
        if ( |Vα| + sqrt(3) ≥ |Vβ| )      VExSECTOR = 0
        else                             VExSECTOR = 1
    else                                   VExSECTOR = 2
else if ( Vα < 0 & Vβ ≥ 0 )
    if ( |Vα| < |Vβ| + sqrt(3) )          VExSECTOR = 3
    if ( |Vα| + sqrt(3) < |Vβ| )          VExSECTOR = 4
    else                                    VExSECTOR = 5
else if ( Vα < 0 & Vβ < 0 )
    if ( |Vα| ≥ |Vβ| + sqrt(3) )
        if ( |Vα| + sqrt(3) ≥ |Vβ| )      VExSECTOR = 6
        else                             VExSECTOR = 7
    else                                   VExSECTOR = 8
else if ( Vα ≥ 0 & Vβ < 0 )
    if ( |Vα| < |Vβ| + sqrt(3) )          VExSECTOR = 9
    else if ( |Vα| + sqrt(3) < |Vβ| )      VExSECTOR = 10
    else                                    VExSECTOR = 11
if ( VExFMODE[9] = 1 ) VExSECTORM = VExSECTOR
    
```

	VExTMPREG3	α	32bit (-1.0 ~ 1.0 31bit)
	VExTMPREG4	β	32bit (-1.0 ~ 1.0 31bit)
	VExFMODE[9]	SECTOR	<MREGDIS> 0: , 1:
	VExSECTOR		4bit
	VExSECTORM		4bit

b. 空間ベクトル変換(3相変調で<SECTOR[3:0]> = "0", "1"の場合のみ記載)

<演算式>

if (SECTOR = 0,1)

$$t1 = \sqrt{3} \div V_{ExVDC} \times (\sqrt{3} \div 2 \times V_{\alpha} - 1 \div 2 \times V_{\beta}) \quad : t1$$

$$t2 = \sqrt{3} \div V_{ExVDC} \times V_{\beta} \quad : t2$$

$$t3 = 1 - t1 - t2 \quad :$$

if ( VExFMODE[0] = 0 ) : 3

$$DUTYA = t1 + t2 + t3 \div 2$$

$$DUTYB = t2 + t3 \div 2$$

$$DUTYC = t3 \div 2$$

else : 2

$$DUTYA = t1 + t2$$

$$DUTYB = t2$$

$$DUTYC = 0$$

VExTMPREG0 = DUTYA

VExTMPREG1 = DUTYB

VExTMPREG2 = DUTYC

VExTMPREG5 = t3

	VExTMPREG3	$\alpha$	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG4	$\beta$			
	VExVDC		16bit	(0.0 ~ 1.0	15bit)
	VExSECTOR		4bit		
	VExFMODE[0]		<C2PEN> 0: 3 , 1: 2		
	VExTMPREG0	a Duty	32bit	(0..0 ~ 1.0	31bit)
	VExTMPREG1	b Duty			
	VExTMPREG2	c Duty			
	VExTMPREG5	Duty			

3. 出力相変換 2(逆クランク変換)(タスク 11)

出力相変換 2 は  $\alpha$  軸電圧、 $\beta$  軸電圧からセクタ判定し、逆クランク変換で a 相電圧、b 相電圧、c 相電圧のデューティを算出するタスクです。このタスクは変換方式として 3 相変調だけに対応します。

また、このタスクは VExFMODE<PHCVDIS> を"1"に設定することで 2 相電圧の Duty を算出します。

a. セクタ判定

<演算式>

```

VExSECTORM = VExSECTOR :
Va = VExTMPREG3
Vβ = VExTMPREG4
if ( Va ≥ 0 & Vβ ≥ 0 )
    if ( |Va| ≥ |Vβ| + sqrt(3) )
        if ( |Va| + sqrt(3) ≥ |Vβ| ) VExSECTOR = 0
        else VExSECTOR = 1
    else VExSECTOR = 2
else if ( Va < 0 & Vβ ≥ 0 )
    if ( |Va| < |Vβ| + sqrt(3) ) VExSECTOR = 3
    if ( |Va| + sqrt(3) < |Vβ| ) VExSECTOR = 4
    else VExSECTOR = 5
else if ( Va < 0 & Vβ < 0 )
    if ( |Va| ≥ |Vβ| + sqrt(3) )
        if ( |Va| + sqrt(3) ≥ |Vβ| ) VExSECTOR = 6
        else VExSECTOR = 7
    else VExSECTOR = 8
else if ( Va ≥ 0 & Vβ < 0 )
    if ( |Va| < |Vβ| + sqrt(3) ) VExSECTOR = 9
    else if ( |Va| + sqrt(3) < |Vβ| ) VExSECTOR = 10
    else VExSECTOR = 11
if ( VExFMODE[9] = 1 ) VExSECTORM = VExSECTOR :
    
```

	VExTMPREG3	$\alpha$	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG4	$\beta$	32bit	(-1.0 ~ 1.0	31bit)
	VExFMODE[9]	SECTOR	<MREGDIS> 0: , 1:		
	VExSECTOR		4bit		
	VExSECTORM		4bit		

b. 逆クラーク変換

<演算式>

```

if ( VExFMODE[12] = 0 )
    VExTMPREG0 = 1 + VExVDC × Vα + 1/2
    VExTMPREG1 = 1 + VExVDC × ( -1 + 2 × Vα + sqrt(3) ÷ 2 × Vβ ) + 1/2
    VExTMPREG2 = 1 + VExVDC × ( -1 + 2 × Vα - sqrt(3) ÷ 2 × Vβ ) + 1/2
else
    VExTMPREG0 = 1 + VExVDC × Vα + 1/2
    VExTMPREG1 = 1 + VExVDC × Vβ + 1/2
    
```

	VExTMPREG3	α	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG4	β	32bit	(-1.0 ~ 1.0	31bit)
	VExVDC		16bit	(0.0 ~ 1.0	15bit)
	VExFMODE[12]		<PHCVDIS> 0: , 1:		
	VExTMPREG0	a Duty	32bit	(0..0 ~ 1.0	31bit)
	VExTMPREG1	b Duty	32bit	(0..0 ~ 1.0	31bit)
	VExTMPREG2	c Duty	32bit	(0..0 ~ 1.0	31bit)

18.4.2.4

出力制御部は、3相電圧デューティをPMD設定形式に変換します。変換した結果をVExCMPU、VExCMPV、VExCMPWに設定し、出力制御動作設定に応じてVExOUTCRを設定します。また、デッドタイム補償制御およびPWM出力制限が可能です。

出力制御には出力制御1タスクと出力制御2タスクの2種類があり、それぞれ対応できるPWM出力が違います。

1. 出力制御1(タスク0)

出力制御1タスクは通常PWM出力とPWMシフト1モードのPWM出力に対応しています。

PWMシフト許可時に、回転速度(VExOMEGA)がPWMシフト切り替え基準(VExFPWMCHG)より小さい場合にPWM出力がPWMシフト1に切り替わります。

注)PWMシフト1は1シャント電流検出モード時のみ選択できます。

- 出力変換

<演算式>

```

VExMCTLF[1]= VExMCTLF[0]           :
VExMCTLF[0]= 0                       :
if ((FMODE[3]= 1) & (FMODE[0]= 1) & (FMODE[1]= 1)) : 1      ,2      1
    if (|VExOMEGA| < VExFPWMCHG) VExMCTLF[0]= 1   :
DUTYA = VExTMPREG0
DUTYB = VExTMPREG1
DUTYC = VExTMPREG2
if (VExMCTLF[0]= 1)                   : PWM    1
    if (VExSECTOR = 0,3,4,7,8,11)      :
        DUTYA = DUTYA + VExTMPREG5     :          V7
        DUTYB = DUTYB + VExTMPREG5
        DUTYC = DUTYC + VExTMPREG5
PWMA= VExTMPREG0 × VExMDPRD           : PMD
PWMB= VExTMPREG1 × VExMDPRD
PWMC= VExTMPREG2 × VExMDPRD
    
```

VExTMPREG0	a	32bit	(0.0 ~ 1.0	31bit)
VExTMPREG1	b	32bit	(0.0 ~ 1.0	31bit)
VExTMPREG2	c	32bit	(0.0 ~ 1.0	31bit)
VExMDPRD	PWM	16bit	(PMD PWM	)
VExSECTOR		4bit		
VExOMEGA		16bit	(-1.0 ~ 1.0	15bit)
VExFPWMCHG	PWM	16bit	(0.0 ~ 1.0	15bit)
VExFMODE[0]		<C2PEN> 0: 3 , 1: 2		
VExFMODE[1]	PWM	<SPWMEN> 0: , 1:		
VExFMODE[3:2]		<IDMODE[1:0]> 00:3 . 01: 2 , 1x:1		
VExMCTLF[1:0]		<LAVFM>, <LAVF>		

- PWM出力制限

<演算式>

```

if (VExPWMMAX = 0)                MAX = VExMDPRD
else                                MAX = VExPWMMAX
if ((PWMA > MAX) & (VExMDPRD > MAX)) : U  PWM
    if ((VExMODE[13] = 0) | (PWMA < VExMDPRD)) : 100%
        PWMA = MAX
        VExMCTLF[11] = 1
MIN = VExPWMMIN
if ((PWMA < MIN) & (MIN > 0))      : U  PWM
    if ((VExMODE[12] = 0) | (PWMA > 0))      : 0%
        PWMA = MIN
        VExMCTLF[11] = 1
(      2      )
    
```

	VExMDPRD	PWM	16bit (PMD PWM )
	VExPWMMAX	PWM	16bit (0 ~ VExMDPRD )
	VExPWMMIN	PWM	16bit (0 ~ VExMDPRD )
	VExMODE[12]	PWM 0%	<PWMBLEN> 0: , 1:
	VExMODE[13]	PWM 100%	<PWFLEN> 0: , 1:
	VExMCTLF[11]	PWM	<PWMOV>

- デッドタイム補償

<演算式>

```

if (0 < PWMA < VExMDPRD)          DT = VExDTC
else                                DT = 0
if (VExDTCS<IASTS> = 01)           :
  if (VExMODE[14] = 1)             : PMD
    if (PWMA > (VExMDPRD - 2 × DT)) PWMA = (VExMDPRD + PWMA) ÷2
  else                              PWMA = PWMA + DT
  if ((VExMODE[13] = 1) & (PWMMAX < VExMDPRD)) : 100%
    if (PWMA > (VExMDPRD - 1))     PWMA = VExMDPRD - 1 :
  else
    if (PWMA > VExMDPRD)           PWMA = VExMDPRD :
else if (VExDTCS<IASTS> = 11)     :
  if (VExMODE[14] = 1)           : PMD
    if (PWMA < (2 × DT))           PWMA = PWMA ÷2
  else                              PWMA = PWMA - DT
  if ((VExMODE[12] = 1) & (PWMMIN > 0)) : 0%
    if (PWMA < 1)                  PWMA = 1 :
  else
    if (PWMA < 0)                  PWMA = 0 :
( 2 )

```

	VExMDPRD	PWM	16bit (PMD PWM )
	VExDTC		16bit (0 ~ VExMDPRD )
	VExMODE[12]	0%	<PWMBLEN>
	VExMODE[13]	100%	<PWMFLEN>
	VExMODE[14]	PMD	<PMDDTCEN> 0: PMD 1: PMD
	VExDCTS	/	<ICSTS>, <IBSTS>, <IASTS> x0: , 01: , 11:

2. 出力制御/PWM シフト 1 変換

<演算式>

```

OUTCR = 0x1FF                                     :
if (VExMCTLF[0] = 1)                             : PWM      1
  if (VExSECTOR = 0,1,2,11)                       PWMB = MDPRD - PWMB   : V
  OUTCR = 0x1F3
  else if (VExSECTOR = 3,4,5,6)                   PWMC = MDPRD - PWMC   : W
  OUTCR = 0x1CF
  else if (VExSECTOR = 7,8,9,10)                  PWMA = MDPRD - PWMA   : U
  OUTCR = 0x1FC
if (VExMODE[3:2] = 00,11)                         :
else if (VExMODE[3:2] = 10)                       :
VExCMPU = PWMA
VExCMPV = PWMB
VExCMPW = PWMC
VExOUTCR = OUTCR
    
```

	VExSECTOR		4bit
	VExMODE[3:2]		<OCRMD[1:0]>
	VExMINPLS		16bit
	VExCMPU	PMD U PWM	16bit (0 ~ MDPRD )
	VExCMPV	PMD V PWM	16bit (0 ~ MDPRD )
	VExCMPW	PMD PWM	16bit (0 ~ MDPRD )
	VExOUTCR	PMD	9bit
	VExEMGRS	PMD EMG	1bit
	VExMCTLF		<PLSLF>

3. 出力制御 2(タスク 9)

出力制御 2 タスクは通常 PWM 出力と PWM シフト 2 モードの PWM 出力に対応しています。

PWM シフト許可 (VExFMODE<SPWMEN> = "1")かつ PWM シフトモード選択 (VExFMODE<SPWMMD>)を "00"以外に設定することで PWM シフト 2 の PWM 出力になります。

注)PWM シフトは 1 ショット電流検出モード時のみ選択できます。

- 出力変換

<演算式>

VExMCTLF[1] = VExMCTLF[0] :  
 VExMCTLF[0] = 0 :  
 PWMA = VExTMPREG0 × VExMDPRD : PMD  
 PWMB = VExTMPREG1 × VExMDPRD  
 PWMC = VExTMPREG2 × VExMDPRD

	VExTMPREG0	a	32bit	(0.0 ~ 1.0	31bit)
	VExTMPREG1	b	32bit	(0.0 ~ 1.0	31bit)
	VExTMPREG2	c	32bit	(0.0 ~ 1.0	31bit)
	VExMDPRD	PWM	16bit	(PMD PWM	)
	VExMCTLF[1:0]		<LAVFM>, <LAVF>		

- PWM 出力制限

<演算式>

if (VExPWMMAX = 0) MAX = VExMDPRD  
 else MAX = VExPWMMAX  
 if ((PWMA > MAX) & (VExMDPRD > MAX)) : U PWM  
 if ((VExMODE[13] = 1) | (PWMA < VExMDPRD)) : 100%  
 PWMA = MAX  
 VExMCTLF[11] = 1  
 MIN = VExPWMMIN  
 if ((PWMA < MIN) & (MIN > 0)) : U PWM  
 if ((VExMODE[12] = 1) | (PWMA > 0)) : 0%  
 PWMA = MIN  
 VExMCTLF[11] = 1  
 ( 2 )

	VExMDPRD	PWM	16bit	(PMD PWM	)
	VExPWMMAX	PWM	16bit	(0 ~ VExMDPRD	)
	VExPWMMIN	PWM	16bit	(0 ~ VExMDPRD	)
	VExMODE		<OCRMD[1:0]>		
	VExFMODE[12]	PWM 0%	<PWMBLEN> 0: , 1:		
	VExFMODE[13]	PWM 100%	<PWFLEN> 0: , 1:		
	VExMCTLF[11]	PWM	<PWMOVF>		

- デッドタイム補償

<演算式>

```

if (0 < PWMA < VExMDPRD)          DT = VExDTC
else                                DT = 0
if (VExDTCS<IASTS> = 01)           :
  if (VExMODE[14] = 1)             : PMD
    if (PWMA > (VExMDPRD - 2 × DT)) PWMA = (VExMDPRD + PWMA) ÷2
  else                               PWMA = PWMA + DT
if ((VExMODE[13] = 1) & (PWMMAX < VExMDPRD)) : 100%
  if (PWMA > (VExMDPRD - 1))      PWMA = VExMDPRD - 1
  else
    if (PWMA > VExMDPRD)          PWMA = VExMDPRD
else if (VExDTCS<IASTS> = 11)      :
  if (VExMODE[14] = 1)           : PMD
    if (PWMA < (2 × DT))          PWMA = PWMA ÷2
  else                               PWMA = PWMA - DT
if ((VExMODE[12] = 1) & (PWMMIN > 0)) : 0%
  if (PWMA < 1)                   PWMA = 1
  else
    if (PWMA < 0)                 PWMA = 0
( 2 )
    
```

	VExMDPRD	PWM	16bit (PMD PWM )
	VExDTC		16bit (0 ~ VExMDPRD )
	VExMODE[12]	0%	<PWMBLEN>
	VExMODE[13]	100%	<PWFLEN>
	VExMODE[14]	PMD	<PMDDTCEN> 0: PMD 1: PMD
	VExDCTS	/	<ICSTS>, <IBSTS>, <IASTS> x0: , 01: , 11:

- 出力制御/PWM シフト 2 変換

<演算式>

```

OUTCR = 0x1FF
if ((VExFMODE[3] = 1) & (VExFMODE[1] = 1))
    if (VExFMODE[15:14] = 01)
        if (PWMB > VExMDPRD +2)
            PWMB = VExMDPRD - PWMB
            OUTCR = OUTCR & 0x1F3
        if (PWMC > VExMDPRD +2)
            PWMC = VExMDPRD - PWMC
            OUTCR = OUTCR & 0x1CF
    else if (VExFMODE[15:14] = 10)
        if (PWMA > VExMDPRD +2)
            PWMA = VExMDPRD - PWMA
            OUTCR = OUTCR & 0x1FC
        if (PWMC > VExMDPRD +2)
            PWMC = VExMDPRD - PWMC
            OUTCR = OUTCR & 0x1CF
    else if (VExFMODE[15:14] = 11)
        if (PWMA > VExMDPRD +2)
            PWMA = VExMDPRD - PWMA
            OUTCR = OUTCR & 0x1FC
        if (PWMB > VExMDPRD +2)
            PWMB = VExMDPRD - PWMB
            OUTCR = OUTCR & 0x1F3
if (VExMODE[3:2] = 00,11)
    OUTCR = 0x000
else if (VExMODE[3:2] = 10)
    OUTCR = 0x015
VExCMPU = PWMA
VExCMPV = PWMB
VExCMPW = PWMC
VExOUTCR = OUTCR
    
```

	VExMDPRD	PWM	16bit (PMD PWM )
	VExFPWMCHG	PWM	16bit (0.0 ~ 1.0 15bit)
	VExMODE[3:2]		<OCRMD>
	VExMINPLS		16bit
	VExFMODE[1]	PWM	<SPWMEN>
	VExFMODE[3:2]		<IDMODE>
	VExFMODE[15:14]	PWM	<SPWMMD>
	VExCMPU	PMD U PWM	16bit (0 ~ MDPRD )
	VExCMPV	PMD V PWM	16bit (0 ~ MDPRD )
	VExCMPW	PMD PWM	16bit (0 ~ MDPRD )
	VExOUTCR	PMD	9bit
	VExEMGRS	PMD EMG	1bit
	VExMCTLF		<PLSLF>

18.4.2.5 ( 1)

トリガ生成部は、1 シャント電流検出時に PWM 設定値 VExCMPU、VExCMPV、VExCMPW から電流検出方式に応じたトリガタイミングを算出して VExTRGCMP0、VExTRGCMP1 に設定します。

- ) 1 VExTRGCMP0 VExTRGCMP1
- ) PWM 2 VExTRGCMP0 VExTRGCMP1

	VExCMPU	PMD U PWM	16bit (0 ~ MDPRD )
	VExCMPV	PMD V PWM	16bit (0 ~ MDPRD )
	VExCMPW	PMD W PWM	16bit (0 ~ MDPRD )
	VExMDPRD	PWM	16bit (PMD PWM )
	VExTADC	AD	16bit (0 ~ MDPRD )
	VExTRGCRC		16bit (0 ~ MDPRD )
	VExSECTOR		4bit
	VExMODE[0]		<ZIEN>
	VExMODE[3:2]		<OCRMD>
	VExFMODE[0]		<C2PEN> 0: 3 , 1: 2
	VExFMODE[1]	PWM	<SPWMEN>
	VExFMODE[3:2]		<IDMODE>
	VExFMODE[8]		<CRCEN>
	VExFMODE[15:14]	PWM	<SPWMMD>
	VExMCTLF[0]		<LAVF>
	VExTRGCMP0	PMD 0	16bit (0 ~ MDPRD )
	VExTRGCMP1	PMD 1	16bit (0 ~ MDPRD )
	VExTRGSEL	PMD	3bit

18.4.2.6

入力処理では A/D 変換器から変換結果と相情報を読み込みます。電流検出方式や PWM シフトモードなどの設定に応じて 3 相電流、電圧の変換結果を固定小数点データ変換して保存します。また、ゼロ電流検出モード時は電流検出結果をゼロ電流レジスタに保存します。

デッドタイム補償制御のために、ヒステリシス幅を指定して電流極性を判定できます。

入力処理には入力処理 1 タスクと入力処理 2 タスクの 2 種類があり、それぞれ対応できる電流検出方式が違います。

1. 入力処理 1(タスク 2)

入力処理 1 タスクは 3 シャント(2 相検出のみ(注 1))および 1 シャントの電流検出に対応します。ただし、PWM シフト 2 モードの 1 シャントには対応しません(注 2)。

(注 1)電流検出結果は 2 相のみ使用します。残りの 1 相は計算で求めます。

(注 2)PWM シフトは 1 シャント電流検出モード時のみ選択できます。

- 入力変換

<演算式>

```

[VDC          /   ]
if (VExMODE[4] = 0           VExVDC = [DC   ] >>1
else                          VExVDCL = [DC   ] >>1
[   1t   ]
if (VExFMODE[3:2] = 10,11    : 1
    if (VExMCTLF[1] = 0      :   PWM
        if (VExSECTORM = 4,5,6,7)   VExIAADC = [   ] 1
        else if (VExSECTORM = 8,9,10,11) VExIBADC = [   ] 1
        else f (VExSECTORM = 0,1,2,3) VExICADC = [   ] 1
    else if (VExMCTLF[1] = 1      : PWM   1
        if (VExSECTORM = 1,2,7,8)   VExIAADC = [   ] 1
        else if (VExSECTORM = 0,5,6,11) VExIBADC = [   ] 1
        else f (VExSECTORM = 3,4,9,10) VExICADC = [   ] 1
else if (VExFMODE[3:2] = 00,01  : 3   2
    if ([   1   ] = 1)           VExIAADC = [   ] 1
    else if ([   1   ] = 2)       VExIBADC = [   ] 1
    else if ([   1   ] = 3)       VExICADC = [   ] 1
[   2   ]
if (VExFMODE[3:2] = 10,11    : 1
    if (VExMCTLF[1] = 0      :   PWM
        if (VExSECTORM = 0,1,10,11)   VExIAADC = [   ] 2
        else if (VExSECTORM = 2,3,4,5) VExIBADC = [   ] 2
        else f (VExSECTORM = 6,7,8,9) VExICADC = [   ] 2
    else if (VExMCTLF[1] = 1      : PWM   1
        if (VExSECTORM = 3,4,9,10)   VExIAADC = [   ] 2
        else if (VExSECTORM = 1,2,7,8) VExIBADC = [   ] 2
        else f (VExSECTORM = 0,5,6,11) VExICADC = [   ] 2
else if (VExFMODE[3:2] = 00,01  : 3   2

```

```

N = [ 2 ] : 1ADC
X = [ 2]
if (VExFMODE[13] =
    if ( N = 1)
    else if ( N = 2)
    else if ( N = 3)
[ 3 ]
if (VExFMODE[3] ≠ 1) : 1
    if ([ 3 ] = 1) VExIAADC = [ 3]
    else if ([ 3 ] = 2) VExIBADC = [ 3]
    else if ([ 3 ] = 3) VExICADC = [ 3]
[ ]
IA = VExIAO - VExIAADC
IB = VExIBO - VExIBADC
IC = VExICO - VExICADC
if (VExFMODE[3:2] = 10,11 : 1
    if (VExMCTLF[1] = 0 : PWM
        if (VExSECTORM = 0,1,10,11) IA = -IA
        else if (VExSECTORM = 6,7,8,9) IB = -IB
        else f (VExSECTORM = 2,3,4,5) IC = -IC
    else if (VExMCTLF[1] = 1 : PWM 1
        if (VExSECTORM = 1,2,5,6,9,10) IA = -IA
        IB = -IB
        IC = -IC
[ 3 ]
N = 6 - N - [ 1 ] : 3
if ( N = 1) IA = - IB - IC
else if ( N = 2) IB = - IC - IA
else if ( N = 3) IC = - IA - IB]
[ ]
VExTMPREG0 = IA
VExTMPREG1 = IB
VExTMPREG2 = IC
    
```

	(VDC1)		
	( 1)		
	( 2)	ADC	16bit ( 12bit )
	( 3)		
	( 4)		
	VExSECTORM		
	VExMODE[1]		<ZIEN>
	VExMODE[4]	VDC	<VDCSEL>
	VExFMODE[3:2]		<IDMODE>
	VExFMODE[13]		<SADCEN> 0: , 1:
	VExMCTLF[0]		<LAVFM>

	VExVDC	DC	16bit	(0.0 ~ 1.0	15bit)
	VExVDCL	DC	16bit	(0.0 ~ 1.0	15bit)
	VExIAADC	a	16bit	(	12bit
	VExIBADC	b			
	VExICADC	c			
	VExTMPREG0	a	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG1	b			
VExTMPREG2	c				
	VExIAO	a	16bit	(	12bit
	VExIBO	b			
	VExICO	c			

- 電流極性判定

<演算式>

```

if ( VExMODE[15] = 1) :
    IA = VExTMPREG0
    if ( IASTS = xx0) :
        if ( IA ≥ |HYS| ) IASTS = 001 :
        else if ( IA ≤ -|HYS| ) IASTS = 111 :
    else if ( IASTS = 001) :
        if ( IA ≤ -|HYS| ) <IASTS> = 111 :
        else if ( IA < -HYS ) <IASTS> = 011 : ( )
    else if ( IASTS = 101) :
        if ( IA ≥ |HYS| ) <IASTS> = 001 : ( )
    else if ( IASTS = 111) :
        if ( IA ≥ |HYS| ) <IASTS> = 001 :
        else if ( IA > HYS ) <IASTS> = 101 : ( )
    else if ( IASTS = 011) :
        if ( IA ≤ -|HYS| ) <IASTS> = 111 : ( )
( 2 )
    
```

	VExHYS		16bit (0.0 ~ 1.0 15bit)
	VExDCTS	/	<ICSTS>, <IBSTS>, <IASTS> xx0: , x01: , x11:
	VExMODE[15]		<IPDEN> 0: , 1:
	VExTMPREG0	a	32bit (-1.0 ~ 1.0 31bit)
	VExTMPREG1	b	
	VExTMPREG2	c	
	VExDCTS	/	<ICSTS>, <IBSTS>, <IASTS> x0: , 01: , 11:

## 2. 入力処理 2(タスク 10)

入力処理 2 タスクは 3 シャント(3 相検出、2 相検出)、2 センサの電流検出に対応します。また、PWM シフト 2 モードの PWM 出力時の 1 シャント電流検出に対応します。電流検出方向を相別に選択できます。

(注)入力処理 2 タスクはゼロ電流検出モードには対応していません。

(注)PWM シフト 2 は 1 シャント電流検出モード時のみ選択できます。

### - 入力変換

<演算式>

```

[VDC          / ]
if (VExMODE[4] = 0          VExVDC = [DC          ] >>1
else          VExVDCL = [DC          ] >>1
[          1t          ]
if ([          1          ] = 1)          VExIAADC = [          1]
else if ([          1          ] = 2)          VExIBADC = [          1]
else if ([          1          ] = 3)          VExICADC = [          1]
[          2          ]
N = [          2          ]          : 1ADC
X = [          2]
if (VExFMODE[13] =          N = [          4          ]          :          4
          X = [          4]
if ( N = 1)          VExIAADC = X
else if ( N = 2)          VExIBADC = X
else if ( N = 3)          VExICADC = X
[          3          ]
if (VExFMODE[3:2] = 00)          : 3
    if ([          3          ] = 1)          VExIAADC = [          3]
    else if ([          3          ] = 2)          VExIBADC = [          3]
    else if ([          3          ] = 3)          VExICADC = [          3]
else          N = 6 - N - [          1          ]          : 3          3
[          / ]
IA = VExIAO - VExIAADC
if (VExFMODE[5] = 1)          IA = -IA          : Ia
IB = VExIBO - VExIBADC
if (VExFMODE[6] = 1)          IB = -IB          : Ib
IC = VExICO - VExICADC
if (VExFMODE[7] = 1)          IC = -IC          : Ic
if (VExFMODE[3:2] ≠ 00)          : 3          3
    if ( N = 1)          IA = - IB - IC
    else if ( N = 2)          IB = - IC - IA
    else if ( N = 3)          IC = - IA - IB

```

VExTMPREG0 = IA  
 VExTMPREG1 = IB  
 VExTMPREG2 = IC

	(DC )		
	( 1)		
	( 2)	ADC	16bit ( 12bit )
	( 3)		
	( 4)		
	VExMODE[4]	VDC	<VDCSEL>
	VExFMODE[3:2]		<IDMODE>
	VExFMODE[7:5]		<ICPLMD>,<IBPLMD>,<IAPLMD> 0: ( In = VExInO - VExInADC ) 1: ( In = VExInADC - VExInO ) ) n = A,B,C
	VExFMODE[13]		<SADCEN> 0: , 1:
	VExVDC	DC	16bit (0.0 ~ 1.0 15bit)
	VExVDCL	DC	16bit (0.0 ~ 1.0 15bit)
	VExIAADC	a	
	VExIBADC	b	16bit ( 12bit )
	VExICADC	c	
	VExTMPREG0	a	
	VExTMPREG1	b	32bit (-1.0 ~ 1.0 31bit)
	VExTMPREG2	c	
	VExIAO	a	
	VExIBO	b	16bit ( 12bit )
	VExICO	c	

- 電流極性判定

<演算式>

```

if ( VExMODE[15] = 1 ) :
    IA = VExTMPREG0
    if ( IASTS = xx0 ) :
        if ( IA ≥ |HYS| ) IASTS = 001 :
        else if ( IA ≤ -|HYS| ) IASTS = 111 :
    else if ( IASTS = 001 ) :
        if ( IA ≤ -|HYS| ) <IASTS> = 111 :
        else if ( IA < -HYS ) <IASTS> = 011 : ( )
    else if ( IASTS = 101 ) :
        if ( IA ≥ |HYS| ) <IASTS> = 001 : ( )
    else if ( IASTS = 111 ) :
        if ( IA ≥ |HYS| ) <IASTS> = 001 :
        else if ( IA > HYS ) <IASTS> = 101 : ( )
    else if ( IASTS = 011 ) :
        if ( IA ≤ -|HYS| ) <IASTS> = 111 : ( )
( 2 )
    
```

	VExHYS		16bit (0.0 ~ 1.0 15bit)
	VExDCTS	/	<ICSTS>, <IBSTS>, <IASTS> x0: , 01: , 11:
	VExMODE[15]		<IPDEN> 0: , 1:
	VExTMPREG0	a	32bit (-1.0 ~ 1.0 31bit)
	VExTMPREG1	b	
	VExTMPREG2	c	
	VExDCTS	/	<ICSTS>, <IBSTS>, <IASTS> xx0: , x01: , x11:

18.4.2.7 ( / )

入力電流変換は、相変換と座標軸変換の2つのタスクに分かれています。

1. 入力相変換(タスク 3)

入力相変換タスクは、Ia, Ib, Ic から I $\alpha$ , I $\beta$  を算出します。

<演算式>

```

if ( VExFMODE[12] = 0 )
    VExTMPREG3 = VExTMPREG0
    VExTMPREG4 = 1 + sqrt(3) * VExTMPREG1 - 1 + sqrt(3) * VExTMPREG2
else if ( VExFMODE[12] = 1 )
    VExTMPREG3 = VExTMPREG0
    VExTMPREG4 = VExTMPREG1
    
```

	VExTMPREG0	a	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG1	b			
	VExTMPREG2	c			
	VExFMODE[12]	<PHCVDIS>			
	VExTMPREG3	$\alpha$	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG4	$\beta$			

2. 入力座標軸変換(タスク 4)

入力座標軸変換タスクは I $\alpha$ , I $\beta$ , VExSINM, VExCOSM から Id, Iq を算出します。

a. 座標軸変換

<演算式>

```

if ( VExMCTLFM[5] = 0 )
    VExID = VExCOSM * VExTMPREG3 + VExSINM * VExTMPREG4
    VExIQ = - VExSINM * VExTMPREG3 + VExCOSM * VExTMPREG4
    
```

	VExTMPREG3	$\alpha$	32bit	(-1.0 ~ 1.0	31bit)
	VExTMPREG4	$\beta$			
	VExSINM	$\theta$	16bit	(-1.0 ~ 1.0	15bit)
	VExCOSM	$\theta$			
	VExMCTLF[5]	<PLSLFM>			
	VExID	d	32bit	(-1.0 ~ 1.0	31bit)
	VExIQ	q			

b. ATAN 演算

<ATANMD> = "10" の場合、d 軸,q 軸電流の偏角を算出します。

<ATANMD> = "11"の場合、モータの電圧方程式を使って d 軸,q 軸誘起電圧を算出して、誘起電圧の偏角を算出します。

<演算式>

```

if ( VExMODE[6] = 1 )
    LD = VExCLD × [VExCLG    ]           : d
    LQ = VExCLQ × [VExCLG    ]           : q
    R = VExCR × [VExCRG     ]           :
    VDIV = VExVD - (VExID × R - VExOMEGA × VExIQ × LQ)           : d
    VQIV = VExVQ - (VExIQ × R + VExOMEGA × VExID × LD)           : q
if ( VExMODE[5] = 1 )
    VExDELTA = ATAN2(VQIV, VDIV)         :
else
    VExDELTA = ATAN2(VExIQ, VExID)       :
    
```

VExCLD	d	16bit ( 11bit)	
VExCLQ	q		
VExCR			
VExCLG		000: 1	001: 1/2 <sup>4</sup> 010: 1/2 <sup>8</sup> 011: 1/2 <sup>12</sup>
VExCRG		100: 1/2 <sup>16</sup>	101 ~ 111: Reserved
VExOMEGA		16bit	( 15bit)
VExVD	d	32bit	(-1.0 ~ 1.0 31bit)
VExVQ	q		
VExMODE[6:5]	ATAN	<ATANMD> 0x: 10: Id,Iq 11: d ,q	
VExDELTA		16bit	(-180 ~ 180, 0x8000 ~ 0x7FFF)

18.4.2.8

1. ATNA2(逆正接関数 2)(タスク 12)

ATAN2 タスクは、XY 平面上の原点と(X,Y)を結ぶ直線の X 軸から原点周りの角度を算出します。

<演算式>

```

X = VExTMPREG4
Y = VExTMPREG5
Z = ATAN ( |Y| ÷ |X|)           :           , 0 ~ 90°
if ( X < 0 & Y ≥ 0 )  Z = 0x00008000 - Z       : 2       (90 ~ 180°)
if ( X < 0 & Y < 0 )  Z = 0xFFFF8000 + Z       : 3       (-90 ~ -180°)
if ( X ≥ 0 & Y < 0 )  Z = -Z                    : 4       (0 ~ -90°)
if ( X = Y = 0 )      Z = 0x00000000           :         (0°)
VExTMPREG5 = Z
    
```

	VExTMPREG4	X	32bit
	VExTMPREG5	Y	
	VExTMPREG5		32bit (0xFFFF8000 ~ 0x00008000 (-180~ 180°))

2. SQRT(平方根関数)(タスク 13)

SQRT タスクは 0.0 ~ 4.0 の入力値から平方根を計算して 0.0 ~ 2.0 を出力します。

<演算式>

```

VExTMPREG5 = SQRT(VExTMPREG5)           :
X = VExTMPREG5                           :   ( 0 ~ 4.0 )
N = 0
if ( X < 0x2000 ) N = 1
if ( X < 0x0800 ) N = 2
if ( X < 0x0200 ) N = 3
if ( X < 0x0080 ) N = 4
if ( X < 0x0020 ) N = 5
if ( X < 0x0008 ) N = 6
if ( X < 0x0002 ) N = 7
if ( X ≥ 0x8000 ) N = -1
X = X × 22N                             :   ( 0.25 ~ 1.0 )
if ( X > 0x7FFF ) X = 0x7FFF
Z = SQRT( X )                             :   ,   0.5 ~ 1.0
if ( X = 0 ) Z = 0
Z = Z ÷ 2N                             :   ( 0 ~ 2.0 )
VExTMPREG5 = Z
    
```

	VExTMPREG5		32bit (0.0 ~ 4.0) 15bit 0x0000_0000 ~ 0x0001_FFFF
	VExTMPREG5		32bit (0.0 ~ 2.0) 15bit 0x0000_0000 ~ 0x0000_FFFF

18.5 VE PMD ADC

ベクトルエンジンの使用チャネルにより、使用できる PMD および ADC の組み合わせに制限があります。

また、電流検出選択と使用 ADC ユニット選択によっても使用する組み合わせは変わります。

3 シャントおよび 2 センサの電流検出では、2つの ADC を使って電流を同時サンプリングすることができます。この場合に使用する電流入力は電流 1 と電流 4 になります。

18-8 PMD

	PMD	0	PMD	1
0		o		
1			o	

18-9 ADC

		ADC A				ADC B			
	VExMODE <IDMODE[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	00	1	2	3	DC				
	01	1	2		DC				
	1x	1	2		DC				
1	00					2	1	3	DC
	01					2	1		DC
	1x					2	1		DC

18-10 ADC

		ADC A				ADC B			
	VExMODE <IDMODE[1:0]>	ADAREG0	ADAREG1	ADAREG2	ADAREG3	ADBREG0	ADBREG1	ADBREG2	ADBREG3
0	00	1		3	DC	4			
	01	1			DC	4			
	1x								
1	00		4				1	3	DC
	01		4				1		DC
	1x								



19

(A-ENC)

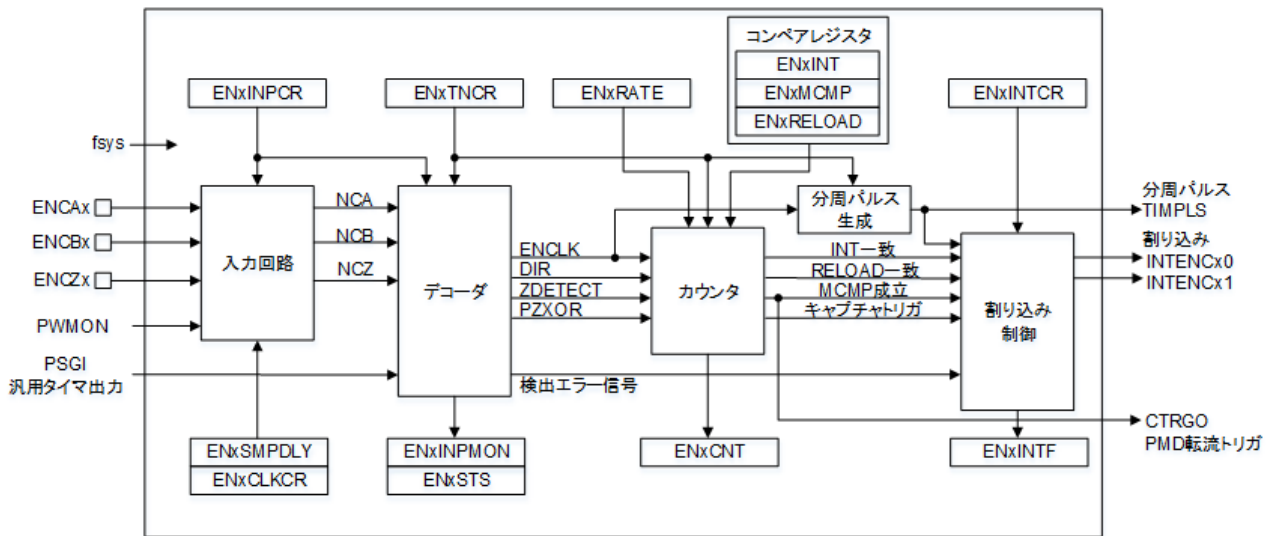
19.1

エンコーダ入力回路は、エンコーダモード、センサモード (3 種類)、タイマモードおよび位相カウンタモードの 6 つの動作モードに対応しています。

以下の機能を有しています。

- ・ インクリメンタル形エンコーダ(Z 信号対応)およびホール IC センサ対応
- ・ 回転エッジ検出(2 相入力 : 4 通倍、3 相入力 : 6 通倍)
- ・ 回転方向判定
- ・ カウンタ機能(イベントカウント、タイマカウント、位相カウント)
- ・ 2 種類の割り込み出力
- ・ 割り込み要求出力許可/禁止および要因フラグ
- ・ 入力回路にデジタルノイズフィルタ内蔵
- ・ PMD 回路の PWM 信号に同期したサンプリング可能
- ・ 矩形波駆動の BLDC モータの BEMF(誘起電圧)ゼロクロス検出対応(PMD 回路使用時)
- ・ 汎用 32 ビットタイマ/キャプチャとして使用可能

19.2



19-1

## 19.3

### 19.3.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		address(Base+)
ENC	ENxTNCR	0x0000
RELOAD	ENxRELOAD	0x0004
INT	ENxINT	0x0008
/	ENxCNT	0x000C
MCMP	ENxMCMP	0x0010
	ENxRATE	0x0014
	ENxSTS	0x0018
	ENxINPCR	0x001C
	ENxSMPDLY	0x0020
	ENxINPMON	0x0024
	ENxCLKCR	0x0028
	ENxINTCR	0x002C
	ENxINTF	0x0030

19.3.2 ENxTNCR(ENC )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	CMPSEL	UDMD		TOVMD	MCMPMD
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	DECMD		SDTEN	-	MODE			P3EN
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	TRGCAPMD	SFTCAP	ENCLR	ZESEL	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	ZEN	ENRUN	-	-	-	ENDEV		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-29		R	"0"
28	CMPSEL	R/W	0 ENxINT 1 ENxRELOAD
27-26	UDMD	R/W	( ) / 00 01 1x ENxRATE "1x" ENxRATE<RATE> < 0 <RATE> ≥ 0
25	TOVMD	R/W	RELOAD ( ) [ ( ) ] 0 1 [ ( ) ] 0 1 [ ] TOVMD CW CCW [ ( ) ] RELOAD
24	MCMPMD	R/W	[ ( ) ] ENxMCMP 0 (ENxMCMP<MCMP> = ) 1 (ENxMCMP<MCMP> ≤ ) ( ) <MCMPMD> = 0

Bit	Bit Symbol	Type	
23-22	DECMD	R/W	[ 00 CW CCW (ENCAx ENCBx ENCZx) 01 CW 10 CCW 11 CW CCW  <DECMD> = 00
21	SDTEN	R/W	3 0 1  (ENxSTS<SKPDT>)
20		R	"0"
19-17	MODE[2:0]	R/W	000 001 ( ) 010 ( ) 011 100 Reserved 101 Reserved 110 ( ) 111
16	P3EN	R/W	[ ( 1) ( 2 / 3 ) 0 2 1 3
15-13		R	"0"
12	TRGCAPMD	R/W	[ ( ) ] 0 1  Z ( )
11	SFTCAP	W	[ ( ) ] 1 "1" ENxCNT "0" "0"
10	ENCLR	W	1 "1" "0" "0" "0"
9-8	ZESEL	R/W	[ ( ) ] Z ( Z /PSGI ) 00 Reserved 01 10 11

Bit	Bit Symbol	Type	
7	ZEN	R/W	[ ( 2 ) Z 0 ENCZ 1 ENCZ
6	ENRUN	R/W	0 1  <ENRUN> = "1" <ZDET> "0" <ENRUN> = "0"
5-3		R	"0"
2-0	ENDEV[2:0]	R/W	[ ( )] (TIMPLS)  000 1 100 16 001 2 101 32 010 4 110 64 011 8 111 128

1) <P3EN> = 0

2) <ZEN> = 0

動作モードは <MODE[2:0]>、<P3EN>、<ZEN>により決定し、全部で 12 種類の設定があります。

動作モードの設定表を以下に示します。

<MODE[2:0]>	<ZEN>	<P3EN>		
000	0	0	A, B	
	1		A, B, Z	( Z )
001	0	0	A, B	( )
		1	A, B, Z	( , Z )
010	0	0	A, B	( , 2 )
		1	A, B, Z	( , 3 )
011	0	0	-	
	1		Z	( Z )
110	0	0	A, B	( , 2 )
		1	A, B, Z	( , 3 )
111	0	0	-	
	1		Z	( Z )



19.3.4 ENxINT(INT )

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

Bit	Bit Symbol	Type	
31-16	INTH	R/W	[ ( ) ] 16 <INTH> INT [ ( ) ] 32 <INTH> 32 16 INT PWM [ ( ) ] 16 <INTH> INT
15-0	INTL	R/W	[ ( ) ] 32 16 [ ]

19.3.5 ENxCNT( )

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

Bit	Bit Symbol	Type	
31-16	CNTH	R	[ ( ) ] [ ( ) ] /ENCZx [ ENCZ ] [ ( ) ] 32 16
15-0	CNTL	R	[ ( ) ] [ ( ) ] 32 16

19.3.6 ENxMCMP(MCMP )

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

Bit	Bit Symbol	Type	
31-16	MCMPH	R/W	<p>[ ( ) ]</p> <p>32 &lt;MCMPH&gt; 32 16</p> <p>( &lt;MCMPLD&gt; = 1 )</p> <p>ENxMCMP ≤ MCMP 1</p> <p>( &lt;MCMPLD&gt; = 0 )</p> <p>ENxMCMP = MCMP</p> <p>[ ( ) ]</p> <p>16 ENxMCMP&lt;MCMPLD&gt; = MCMP</p> <p>ENxINTCR&lt;MCMPLD&gt; PMD</p>
15-0	MCMPL	R/W	<p>[ ( ) ]</p> <p>32 16</p> <p>[ ]</p>

19.3.7 ENxRATE( )

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

Bit	Bit Symbol	Type	
31-15		R	"0"
15-0	RATE	R/W	$[ \quad ( \quad ) \quad ]$ $f_{sys} \times \langle \text{RATE} \rangle / 2^{16}$ $\text{ENxTNCr} \langle \text{UDMD} \rangle \quad \langle \text{RATE} \rangle$ <p>                     &lt;UDMD&gt; = 0x      0    /1.0    (0x0000 ~ 0xFFFF)                      &lt;UDMD&gt; = 1x      -0.5 /0.5    (0x8000 ~ 0x7FFF)                      &lt;UDMD&gt; = 1x      &lt;RATE&gt; 2                 </p>

19.3.8 ENxSTS( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	REVERR	UD	ZDET	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	SKPDT	PDERR	INERR
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-15		R	"0"
14	REVERR	R	[ ( ) ] <UD> ( 1)( 2) 0 - 1 <UD> ENxTNCr<ENRUN> = "0" "0"
13	UD	R	[ ( ) ] 0 CCW ( ) 1 CW ( ) CW ENxTNCr<ENRUN> = "0" "1" CCW "0"
12	ZDET	R	ENCZ 0 Z 1 Z ENxTNCr<ENRUN> = "0"
11-3		R	"0"
2	SKPDT	R	( 1) 0 1
1	PDERR	R	[ ( ) ] ( 1) 0 1
0	INERR	R	[ ( ) ] ( 1) 0 1 3 3 Low High "1"

1)

2)

"0"

19.3.9 ENxINPCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	NCT						
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	PDSTP	PDSTT	-	-	-	SYNCNCZEN	SYNCSPLMD	SYNCSPLEN
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-15		R	"0"
14-8	NCT	R/W	0~127(0x00~0x7F) x (<SPLCKS> ) "0" PWM PWM
7	PDSTP	W	[ ( ) ] PWM (BFMF ) 1 "1" "0"
6	PDSTT	W	( ) PWM (BEMF ) 1 "1" "0"
5-3		R	"0"
2	SYNCNCZEN	R/W	PWM 0 PWM 1 PWM PWM (<SYNCSPLEN> = 1) PWM (<SYNCSPLMD> =
1	SYNCSPLMD	R/W	PWM 0 PWM 1 PWM PWM (<SYNCSPLEN> = 1)
0	SYNCSPLEN	R/W	PWM 0 1 PWM PMD PWM ( ) <SYNCSPLEN> = 1 BEMF

19.3.10 ENxSMPDLY( )

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	SMPDLY	R/W	0~255(0x00~0xFF) <SMPDLY> <sup>3</sup> (SPLCKS ) PWM (ENxINPCR<SYNCSPLEN> = 1 ENxINPCR<SYNCSPLMD> = 0) PWM

19.3.11 ENxINPMON( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	DETMONZ	DETMONB	DETMONA	-	SPLMONZ	SPLMONB	SPLMONA
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-7		R	"0"
6	DETMONZ	R	NCZ NCZ
5	DETMONB	R	NCB NCB
4	DETMONA	R	NCA NCA
3		R	"0"
2	SPLMONZ	R	ENCZx ENCZx (NCZ)
1	SPLMONB	R	ENCBx ENCBx (NCB)
0	SPLMONA	R	ENCAx ENCAx (NCA)

19.3.12 ENxCLKCR( )

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

Bit	Bit Symbol	Type	
31-2		R	"0"
1-0	SPLCKS	R/W	00 fsys 01 fsys/2 10 fsys/4 11 fsys/8  ENCAx ENCBx ENCZx PWM (ENxINPCR<SYNCSPLEN> = 1 ENxINPCR <SYNCSPLMD> = 1)

19.3.13 ENxINTCR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	MCMPIE	RLDIE	CMPIE	ERRIE	CAPIE	TPLSIE
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-6		R	"0"
5	MCMPIE	R/W	MCMCP 0 1 "1" MCMCP INTENCx1
4	RLDIE	R/W	RELOAD 0 1 "1" RELOAD INTENCx1 ( )
3	CMPIE	R/W	INT 0 1 "1" INT INTENCx1
2	ERRIE	R/W	0 1 "1" (PDERR) (SKPDT) INTENCx0
1	CAPIE	R/W	0 1 "1" (ENCZx ) (ENCLK) INTENCx0 ( )
0	TPLSIE	R/W	0 1 "1" INTENCx0 ( )

19.3.14 ENxINTF( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	MCMPF	RLDCPF	INTCPF	ERRF	CAPF	TPLSF
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-6		R	"0"
5	MCMPF	R	MCMP 0 1
4	RLDCPF	R	RELOAD 0 1  ( )
3	INTCPF	R	INT 0 1
2	ERRF	R	0 1
1	CAPF	R	0 1  ( )
0	TPLSF	R	0 1  ( )

) ENxINTF

## 19.4

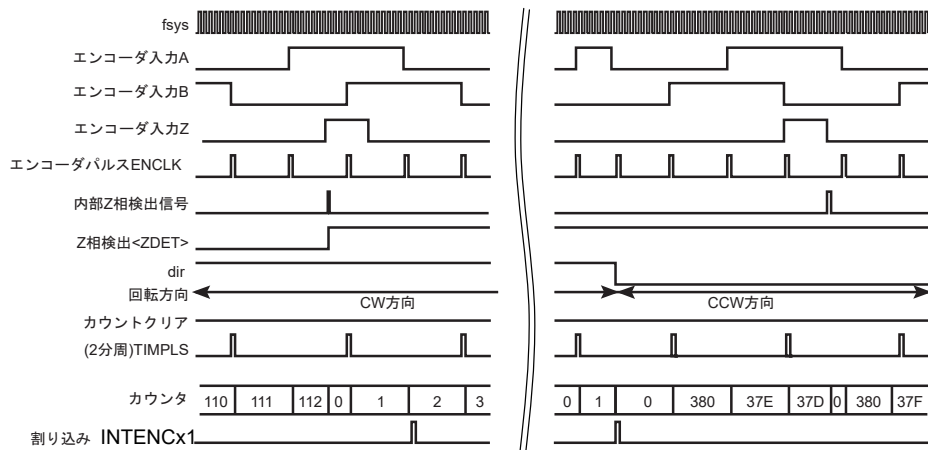
### 19.4.1

高速位置センサ対応(位相判定)で、インクリメンタル形エンコーダ入力(AB および ABZ)に対応しています。

- ・ 回転エッジ検出を行い、分周パルスの出力および割り込み要求が可能
- ・ 回転エッジパルスカウント、任意カウント値で割り込み要求発生が可能
- ・ 回転方向判定
- ・ アップダウンカウント(回転方向判定により制御)
- ・ カウント数設定可能
- ・ 検出回転方向の指定可能
- ・ 異常検出フラグ

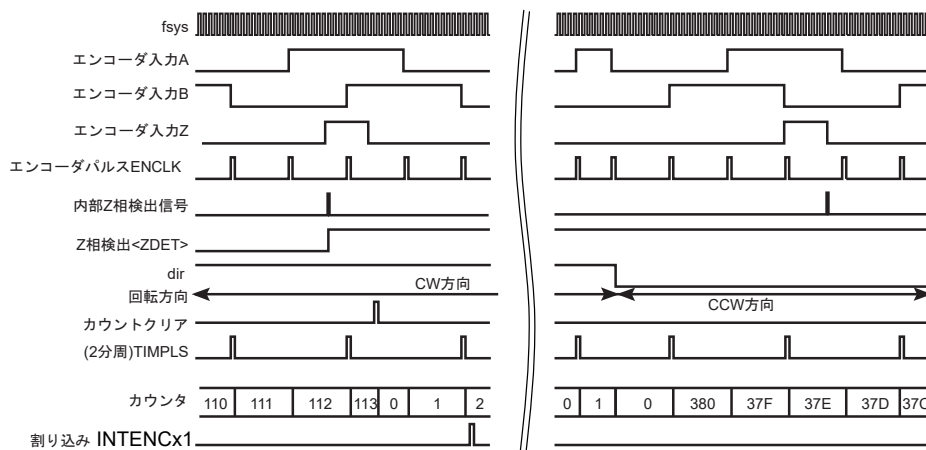
#### 1. ENCZ 入力有効(ENxTNCR<ZEN> = 1)

ENxRELOAD<RELOADH[15:0]> = 0x0380、ENxINT<INTH[15:0]> = 0x0002 の場合



#### 2. ENCZ 入力無効(ENxTNCR<ZEN> = 0)

ENxRELOAD<RELOADH[15:0]> = 0x0380、ENxINT<INTH[15:0]> = 0x0002 の場合



エンコーダモードは、インクリメンタル形エンコーダ信号を ENCA、ENCB、ENCZ 端子に接続します。ENCA、ENCB 信号を 4 通倍して、エンコーダパルスをカウントします。

CW 方向 (A 相が B 相に対して 90 度進んでいる状態) に回転しているときはアップカウントを行い、カウンタ値が <RELOADH> と等しくなったとき、次の ENCLK でカウンタが "0" クリアされます。

CCW 方向 (A 相が B 相に対して 90 度遅れている状態) に回転しているときはダウンカウントを行い、カウンタ値が "0x0000" と等しくなったとき、次の ENCLK でカウンタに <RELOADH> の値がセットされます。

さらに、<ZEN> = "1" の場合は、CW 方向に回転しているときは Z 相の立ち上がりエッジでカウンタが "0" にクリアされます。CCW 方向に回転しているときは Z 相の立ち下がりエッジでカウンタが "0" にクリアされます。ENCLK と Z 相検出のタイミングが同時になったときは、カウント動作を行わず、"0" にクリアされます。

ENxTNCR<ENCLR> に "1" が書き込まれると、カウンタは "0" にクリアされます。

ENxSTS<UD> は CW 方向に回転していることを検出しているときは "1"、CCW 方向のときは "0" がセットされます。

ENxTNCR<DECMD[1:0]> 設定で検出方向を CW 方向のみ、CCW 方向のみ指定することができます。また、<DECMD> = "00" 以外の場合、前回のエッジ検出時に保存した入力状態 ENxINPMON <DETMONA><DETMONB><DETMONZ> と現在の入力値を比較し、回転エッジの検出をします。

ENCLK を分周した信号 (TIMPLS) が出力されます。

ENxINTCR<CMPIE> = "1" のとき、<ENINTH> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<MCMPIE> = "1" のとき、ENxMCMC<MCMPIE[15:0]> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ただし、<ZEN> = "1" の場合、ENxSTS<ZDET> = "0" の期間の一致では割り込みを発生しません。<ZDET> はエンコーダ入力許可後、最初の Z 信号を検出すると "1" になります。

<ZDET>、ENxSTS<UD> は、ENxTNCR<ENRUN> = "0" のときは "0" にクリアされます。

## 19.4.2

低速位置センサ対応 (ゼロクロス判定) で、2 相ホールセンサ入力および 3 相ホールセンサ入力に対応しています。イベントカウントモードとタイマカウントモードおよび位相カウントモードの 3 種類があります。

タイマカウントモードおよび位相カウントモードは、PMD回路からブラシレスDCモータの矩形波駆動している場合、PWM同期サンプリングにすることで誘起電圧のゼロクロス検出に対応できます。

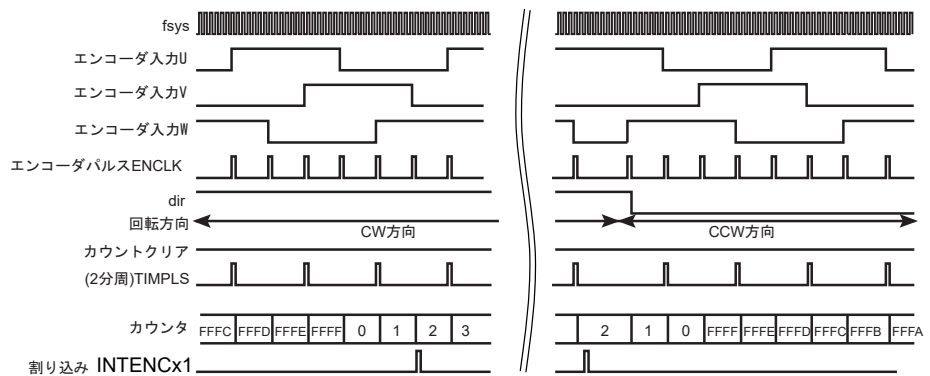
19.4.2.1

回転エッジ検出によりカウントします。

- ・ 回転エッジ検出を行い、分周パルスの出力および割り込み要求が可能
- ・ 回転エッジパルスカウント、任意カウント値で割り込み要求発生が可能
- ・ 回転方向判定
- ・ アップダウンカウント(回転方向判定により制御)
- ・ 検出回転方向の指定可能
- ・ 異常検出フラグ

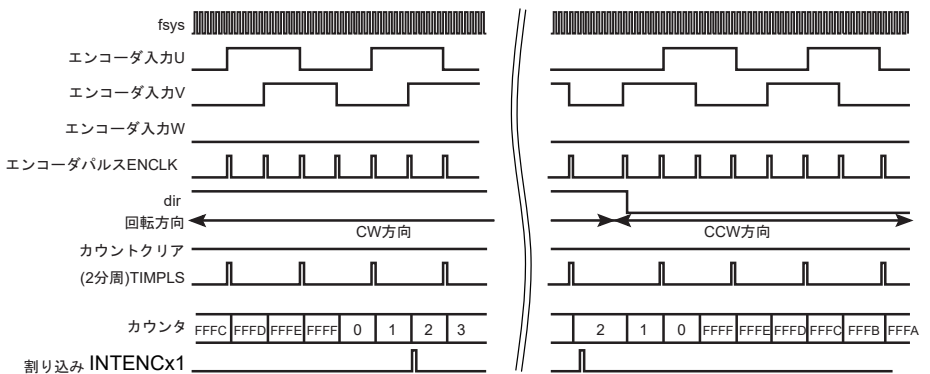
1. 3相デコード(ENxTNCR<P3EN> = 1)

ENxINT<INTH[15:0]> = 0x0002)の場合



2. 2相デコード(ENxTNCR<P3EN> = 0)

ENxINT<INTH[15:0]> = 0x0002)



ホールセンサ入力(U,V,W)を ENCAx、ENCBx、ENCZx に接続します。<P3EN> = "0"の場合は、2 相入力(ENCAx、ENCBx)を 4 通倍、<P3EN> = "1"の場合は、3 相入力(ENCAx、ENCBx、ENCZx)を 6 通倍して、ホールセンサパルスをカウントします。

CW 方向 (ENCA が ENCB に対して 90 度進んでいる状態) に回転しているときはアップカウントを行い、カウント値が"0xFFFF"と等しくなったとき、次の ENCLK でカウンタが"0"にクリアされます。

CCW 方向 (ENCA が ENCB に対して 90 度遅れている状態) に回転しているときはダウンカウントを行い、カウント値が"0x0000"と等しくなったとき、次の ENCLK でカウンタに"0xFFFF"がセットされます。

ENxTNCR<ENCLR> に "1" が書き込まれると、カウンタは"0"にクリアされます。

ENxSTS<UD> は CW 方向に回転していることを検出しているときは "1"、CCW 方向のときは "0" がセットされます。<UD> は ENxTNCR<ENRUN> = "0" のときは "0" にクリアされます。

ENxTNCR<DECMD[1:0]>設定で検出方向を CW 方向のみ、CCW 方向のみ指定することができます。また、<DECMD> = "00"以外の場合、前回のエッジ検出時に保存した入力状態 ENxINPMON<DETMONA><DETMONB><DETMONZ>と現在の入力値を比較し、回転エッジの検出をします。

ENCLK を分周した信号 (TIMPLS) が出力されます。

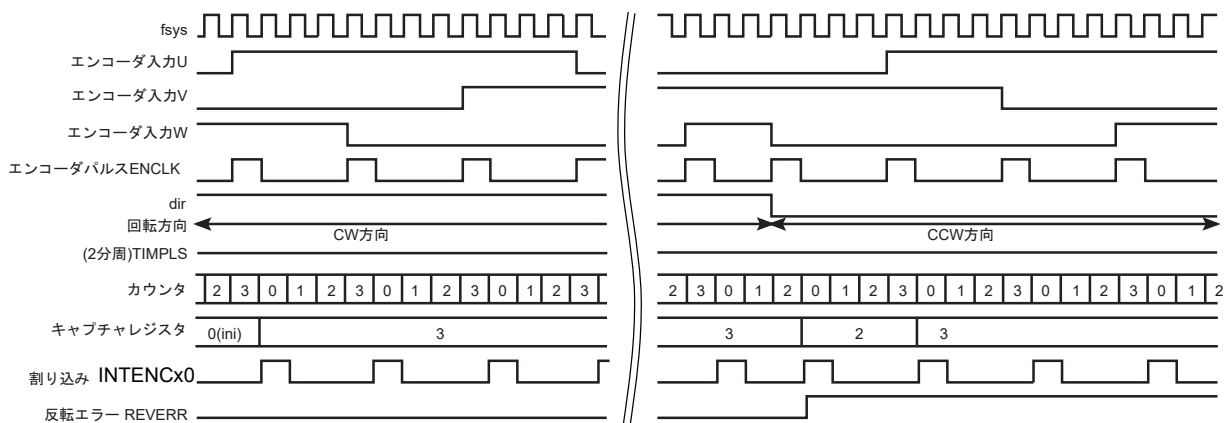
ENxINTCR<CMPIE> = "1" のとき、<ENINTH> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<MCMPIE> = "1" のとき、<ENMCMPH> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

19.4.2.2

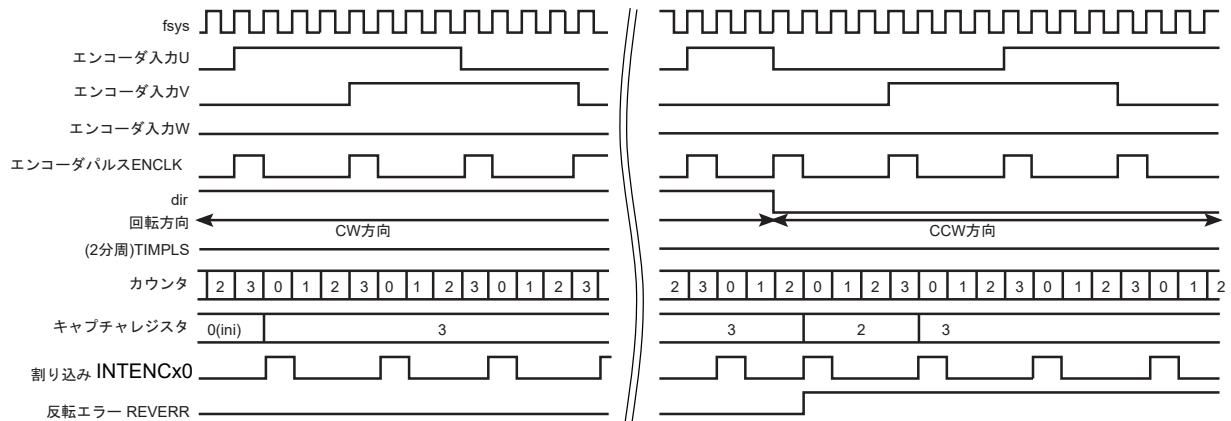
1. 3 相デコード(ENxTNCR<P3EN> = 1)

ENxINT<INTH[15:0]><INTL[15:0]> = 0x00000002 の場合



2. 2 相デコード(ENxTNCR<P3EN> = 0)

ENxINT<INTH[15:0]><INTL[15:0]> = 0x00000002 の場合



ホールセンサ入力(U,V,W)を ENCA<sub>x</sub>、ENCB<sub>x</sub>、ENCZ<sub>x</sub> に接続します。<P3EN> = "0" の場合は、2 相入力(ENCA、ENCB)から 4 通倍、<P3EN> = "1" の場合は、3 相入力(ENCA、ENCB、ENCZ)から 6 通倍の回転エッジパルス(ENCLK)を生成します。

カウンタは常にアップカウントを行い、ENCLK でカウンタが"0x00000000"にクリアされます。

ENxTNCR<ENCLR> に "1" が書き込まれると、カウンタは"0x00000000"にクリアされます。

ENCLK により、カウンタ値がキャプチャされます。キャプチャした値は ENxCNT レジスタから読み出すことができます。

ENxTNCR<SFTCAP> に "1" が書き込まれると、カウンタ値がキャプチャされます。キャプチャは任意のタイミングで行うことができます。キャプチャした値は ENxCNT レジスタから読み出すことができます。

ENxCNT レジスタの値 (キャプチャ値) は、ENxTNCR<ENRUN> の値にかかわらず保持されます。

ENxSTS<UD> は CW 方向に回転していることを検出しているときは "1"、CCW 方向のときは "0" がセットされます。<UD> は <ENRUN> = "0" のときは "0" にクリアされます。回転方向が変化した場合は ENxSTS<REVERR> = "1" にセットされます。フラグは読み出すことでクリアされず。

ENxTNCR<DECMD[1:0]>設定で検出方向を CW 方向のみ、CCW 方向のみ指定することができます。また、<DECMD> = "00"以外の場合、前回のエッジ検出時に保存した入力状態 ENxINPMON<DETMONA><DETMONB><DETMONZ>と現在の入力値を比較し、回転エッジを検出します。

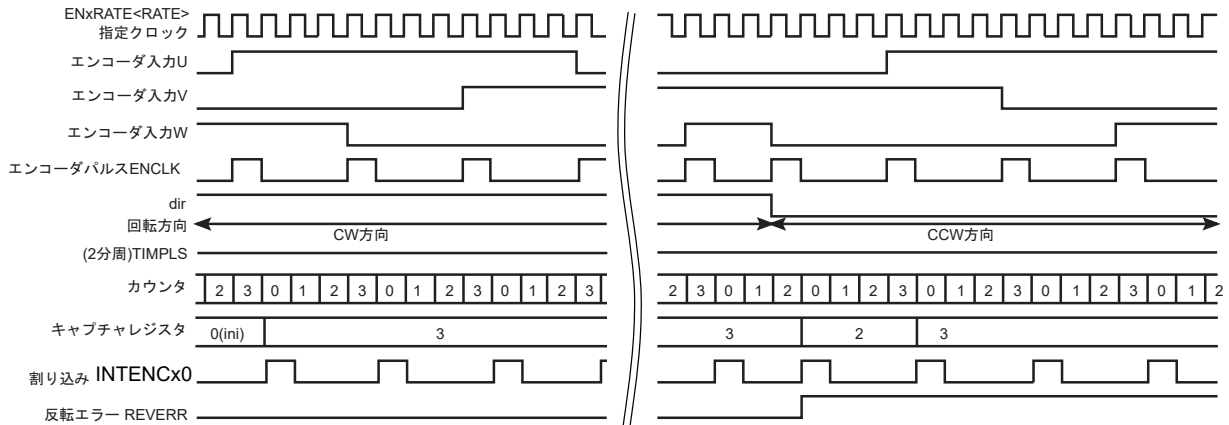
ENxINTCR<RLDIE> = "1"のとき、ENxRELOAD<RELOADH[15:0]><RELOADL[15:0]>の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<CMPIE> = "1" のとき、ENxINT<INTH[15:0]><INTL[15:0]> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

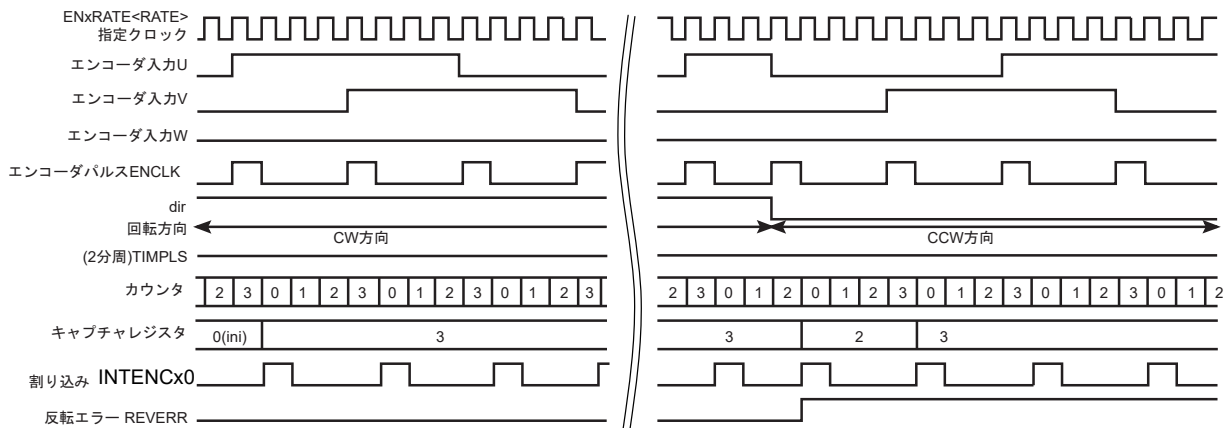
ENxINTCR<MCMPIE> = "1" のとき、ENxMCMP<MCMPL[15:0]><MCMPL[15:0]> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。また、ENxINTCR<MCMPLD> = "1" に設定すると、カウンタ値が <MCMPL> 値以上になったときに INTENCx1 割り込みを発生させることができます。

### 19.4.2.3

#### 1. 3 相デコード(ENxTNCR<P3EN> = 1)



2. 2相デコード(ENxTNCR<P3EN> = 0)



ホールセンサ入力(U,V,W)を ENCAx、ENCBx、ENCZx に接続します。<P3EN> = "0" の場合は、2相入力(ENCAx、ENCBx)から4通倍、<P3EN> = "1" の場合は、3相入力(ENCAx、ENCBx、ENCZx)から6通倍の回転エッジパルス(ENCLK)を生成します。

カウンタは<UDMD>設定およびRATEレジスタ設定で任意周波数でアップダウンカウントを制御できます。アップカウント時にカウンタ値が<RELOADH>と等しくなるとカウンタが"0"クリアされます。ダウンカウント時にカウンタ値が"0x0000"と等しくなるとカウンタに<RELOADH>値がセットされます。

<ENCLR>に"1"が書き込まれると、カウンタは"0"にクリアされます。

<TOVMD> = "1"の場合、カウンタは<RELOADH>値で停止します。

ENCLKにより、カウンタ値がキャプチャされます。キャプチャした値はENxCNTレジスタから読み出すことができます。

<SFTCAP>に"1"が書き込まれると、カウンタ値がキャプチャされます。キャプチャは任意のタイミングで行うことができます。キャプチャした値はENxCNTレジスタから読み出すことができます。

ENxCNTレジスタの値(キャプチャ値)は、<ENRUN>の値にかかわらず保持されます。

<UD>はCW方向に回転していることを検出しているときは"1"、CCW方向のときは"0"がセットされます。<UD>は<ENRUN> = "0"のときは"0"にクリアされます。回転方向が変化した場合は<REVERR> = "1"にセットされます。フラグは読み出すことでクリアされます。

ENxTNCR<DECMD[1:0]>設定で検出方向を CW 方向のみ、CCW 方向のみ指定することができます。また、<DECMD> = "00" 以外の場合、前回のエッジ検出時に保存した入力状態 ENxINPMON <DETMONA> <DETMONB> <DETMONZ>と現在の入力値を比較し、回転エッジの検出をします。

ENxINTCR<CMPIE> = "1" のとき、<ENxINT> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<MCMPIE> = "1" のとき、<ENxMCMP> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

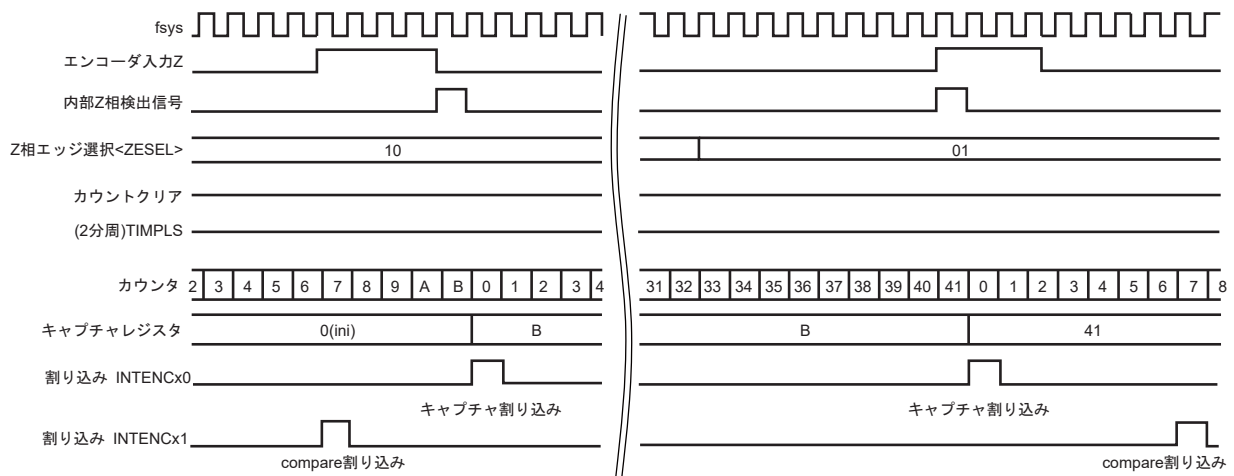
### 19.4.3

汎用 32 ビットタイマとして使用できます。

- ・ 32 ビットアップカウンタ (fsys クロックでカウント)
- ・ カウンタクリア制御 (ソフトクリア、コンペア一致クリア、外部トリガ)
- ・ コンペア機能により一致割り込み発生
- ・ キャプチャ機能：外部トリガキャプチャ (割り込み発生可能)、ソフトキャプチャ

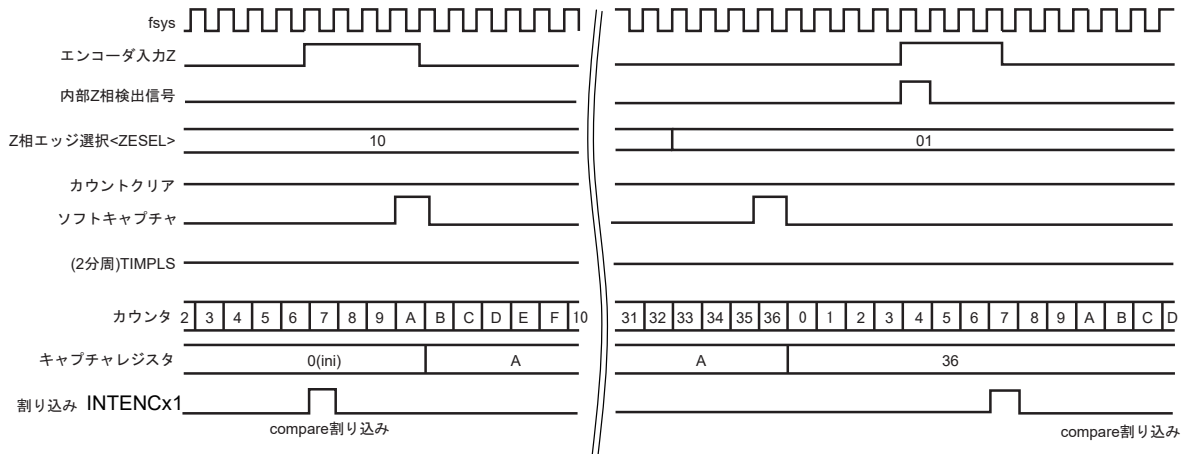
#### 1. ENCZ 入力有効 (ENxTNCR<ZEN> = 1)

$$\text{ENxINT<INTH[15:0]><INTL[15:0]> = 0x0000\_0006}$$



#### 2. ENCZ 入力無効 (ENxTNCR<ZEN> = 0)

$$\text{ENxINT<INTH[15:0]><INTL[15:0]> = 0x0000\_0006}$$



<ZEN> = "1" のとき、Z 入力を外部トリガとして使います。<ZEN> = "0" のとき、外部トリガは使用しません。

カウンタは常にアップカウントを行います。

ENxTNCR<ENCLR> に "1" が書き込まれると、カウンタは"0"にクリアされます。

<ZEN> = "1" の場合、ENxTNCR<ZESEL> = "01" のときは Z 相の立ち上がりエッジでカウンタが"0"にクリアされ、<ZESEL> = "10" のときは Z 相の立ち下がりエッジでカウンタが"0"にクリアされ、<ZESEL> = "11"のときは Z 相の両エッジでカウンタが"0"にクリアされます。

Z 相のエッジ検出により、カウンタ値がキャプチャされます。キャプチャした値は ENxCNT レジスタから読み出すことができます。

ENxTNCR<SFTCAP> に "1" が書き込まれると、カウンタ値がキャプチャされます。キャプチャは任意のタイミングで行うことができます。キャプチャした値は ENxCNT レジスタから読み出すことができます。

ENxCNT レジスタの値 (キャプチャ値)は、ENxTNCR<ENRUN>の値にかかわらず保持されます。キャプチャ値のクリア要因はリセットのみです。

ENxINTCR<RLDIE> = "1"のとき、ENxRELOAD<RELOADH[15:0]><RELOADL[15:0]>の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<CMPIE> = "1" のとき、ENxINT<INTH[15:0]><INTL[15:0]> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<MCMPIE> = "1" のとき、ENxMCMP<MCMPL[15:0]><MCMPL[15:0]> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

また、ENxTNCR<MCMPLD> = "1" に設定すると、カウンタ値が <MCMPL[15:0]><MCMPL[15:0]> > 値以上になったときに INTENCx1 割り込みを発生させることができます。

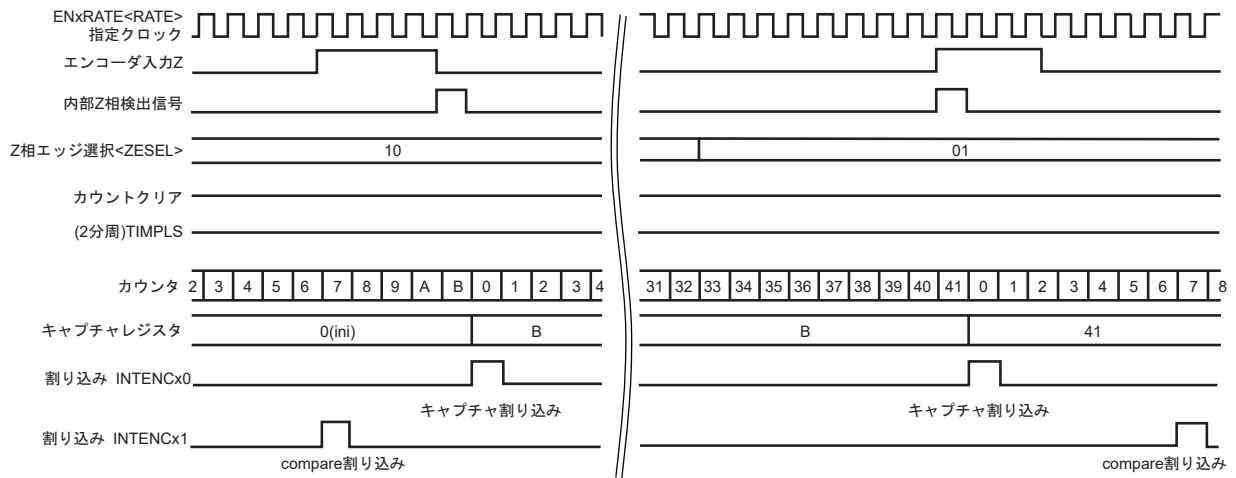
#### 19.4.4

任意周波数で制御できる 16 ビットカウンタです。

- ・ アップダウン制御可能
- ・ コンペア機能、一致割り込み要求発生可能
- ・ ENCZ 入力でキャプチャおよびカウンタクリア可能、割り込み要求発生可能

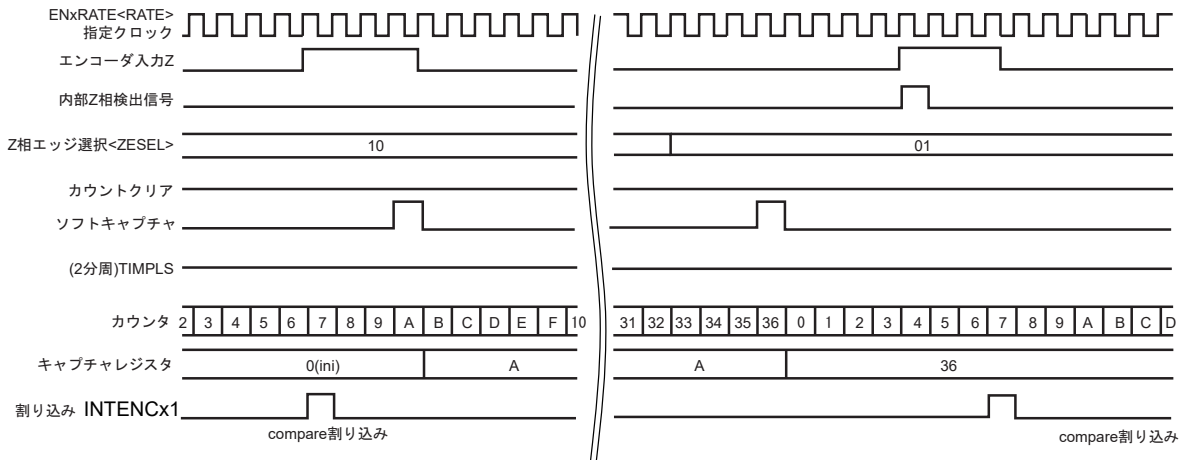
1. ENCZ 入力有効(ENxTNCR<ZEN> = 1)

ENxINT<INTH[15:0]> = 0x0006



2. ENCZ 入力無効(ENxTNCR<ZEN> = 0)

ENxINT<INTH[15:0]> = 0x0006



<ZEN> = "1" のとき、Z 入力端子を外部トリガとして使います。<ZEN> = "0" のとき、外部トリガは使用しません。

カウンタは ENxTNCR<UDMD>設定および ENxRATE レジスタ設定で任意周波数でアップダウンカウンタを制御できます。

アップカウント時に カウンタ値 が ENxRELOAD<RELOADH[15:0]> と等しくなるとカウンタが"0"クリアされます。

ダウンカウント時に カウンタ値 が"0x0000"と等しくなるとカウンタに <RELOADH> 値がセットされます。

ENxTNCR<TOVMD> = "1" の場合、カウンタは <RELOADH> 値で停止します。

ENxTNCR<ENCLR> に "1" が書き込まれると、カウンタは"0"にクリアされます。

<ZEN> = "1" の場合、ENxTNCR<ZESEL> = "01" のときは Z 相の立ち上がりエッジでカウンタが"0"にクリアされ、<ZESEL> = "10" のときは Z 相の立ち下がりエッジでカウンタが"0"にクリアされ、<ZESEL> = "11"のときは Z 相の両エッジでカウンタが"0"にクリアされます。

Z 相のエッジ検出により、カウンタ値がキャプチャされます。キャプチャした値は ENxCNT レジスタから読み出すことができます。

ENxTNCR<SFTCAP> に "1" が書き込まれると、カウンタ値がキャプチャされます。キャプチャは任意のタイミングで行うことができます。キャプチャした値は ENxCNT レジスタから読み出すことができます。

ENxCNT レジスタの値 (キャプチャ値) は、ENxTNCR<ENRUN>の値にかかわらず保持されます。キャプチャ値のクリア要因はリセットのみです。

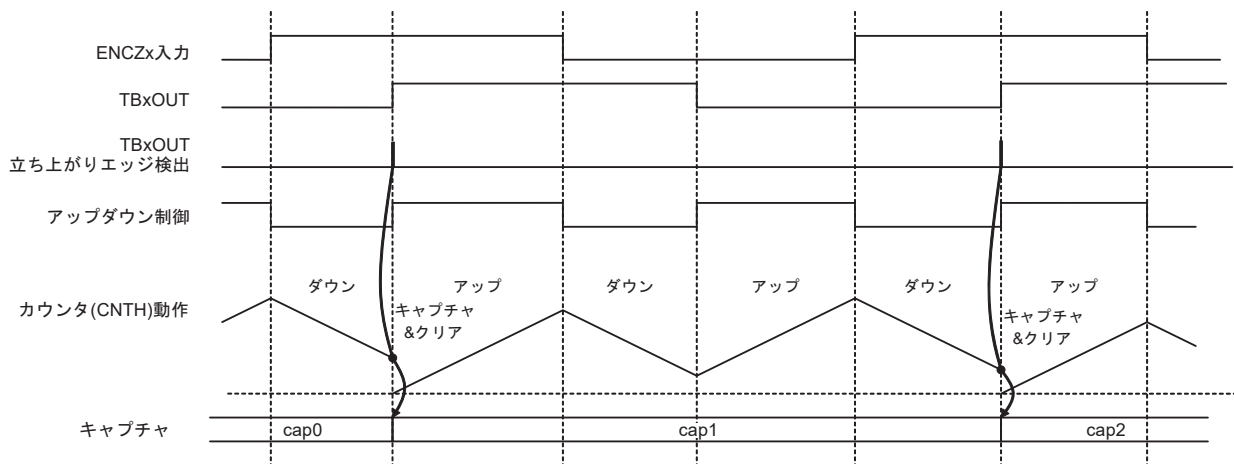
ENxINTCR<CMPIE> = "1" のとき、ENxINT<INTH> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

ENxINTCR<MCMPIE> = "1" のとき、ENxMCMP<MCMPH> の値とカウンタ値が等しくなったときに INTENCx1 割り込みを発生させることができます。

#### 19.4.4.1

位相カウンタモードで <P3EN> = <ZEN> = "1" に設定すると位相差カウンタモードになります。アップダウンカウンタを TMRB 出力と ENCZ 入力で制御します。

- ・ TMRB 出力と ENCZ 入力と同じ場合はアップカウントし、異なる場合はダウンカウントします。
- ・ TMRB 出力エッジでキャプチャおよびカウントクリア可能、割り込み要求発生可能



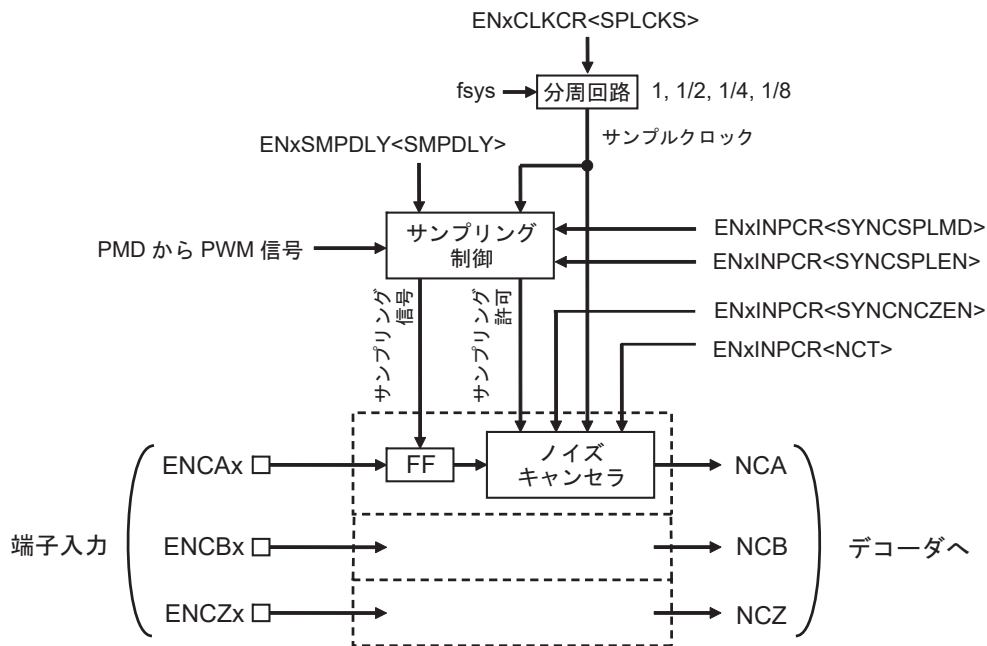
19-2 ( )

TBxOUT 信号のエッジ検出し、カウンタキャプチャおよびカウンタクリアします。この際の検出エッジは ENxTNCR<ZESEL> で選択します。

キャプチャ値は ENCZ 入力と TBxOUT 信号との位相差を表します。ENCZ 入力と TBxOUT 信号との位相差が 1/4 周期の場合が基準 (キャプチャ値が 0) になります。

## 19.5

### 19.5.1



19-3

入力部回路は、端子入力 (ENA,ENB,ENZ) を所定のサンプリング信号でサンプリングし、デジタルノイズフィルタによるノイズ除去を行います。

#### 19.5.1.1

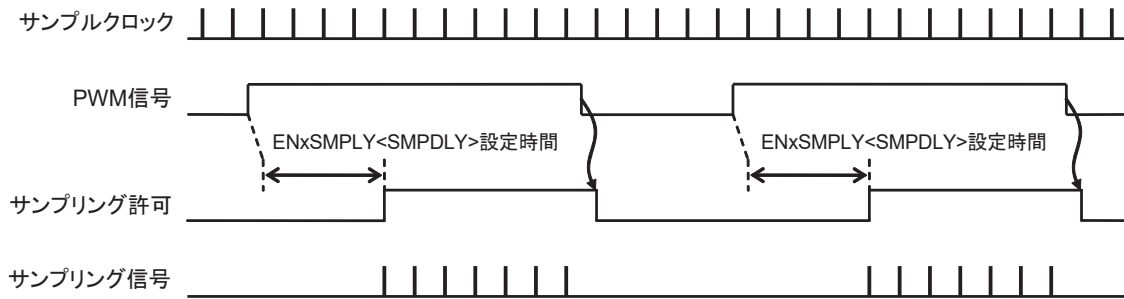
<SPLCKS>により  $f_{sys}$ 、 $f_{sys}/2$ 、 $f_{sys}/4$ 、 $f_{sys}/8$  を選択できます。

#### 19.5.1.2

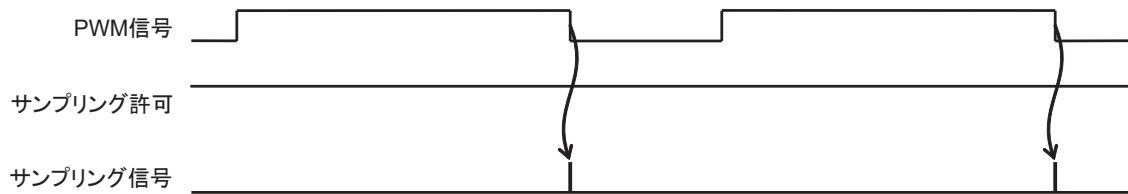
1. 連続サンプリング( $ENxINPCR<SYNCSPLEN> = "0"$ )  
サンプルクロックで入力信号をサンプリングします。
2. PWM 同期サンプリング( $ENxINPCR<SYNCSPLEN> = "1"$ )  
PMD 回路からの PWM 信号に同期したサンプリングを行います。
  - ・ PWM オン期間サンプリング( $ENxINPCR<SYNSPLMD> = "0"$ )。  
PWM 信号がオン期間中だけ  $ENxCLKCR<SPLCKS>$  で選択されたサンプルクロックでサンプリングします。
  - ・ PWM オフエッジサンプリング( $ENxINPCR<SYNSPLMD> = "1"$ )  
PWM 信号がサンプリング信号になり、PWM 信号のオフエッジでサンプリングします。
  - ・ PWM オンディレイ設定

ENxSMPLDSY<SMPDLY> で PWM オン期間サンプリング時にオンディレイを設定できます。

ディレイ時間 : <SMPDLY> × サンプルクロック周期



(a) PWMオン期間サンプリング



(b) PWMオフエッジサンプリング

### 19-4 PWN

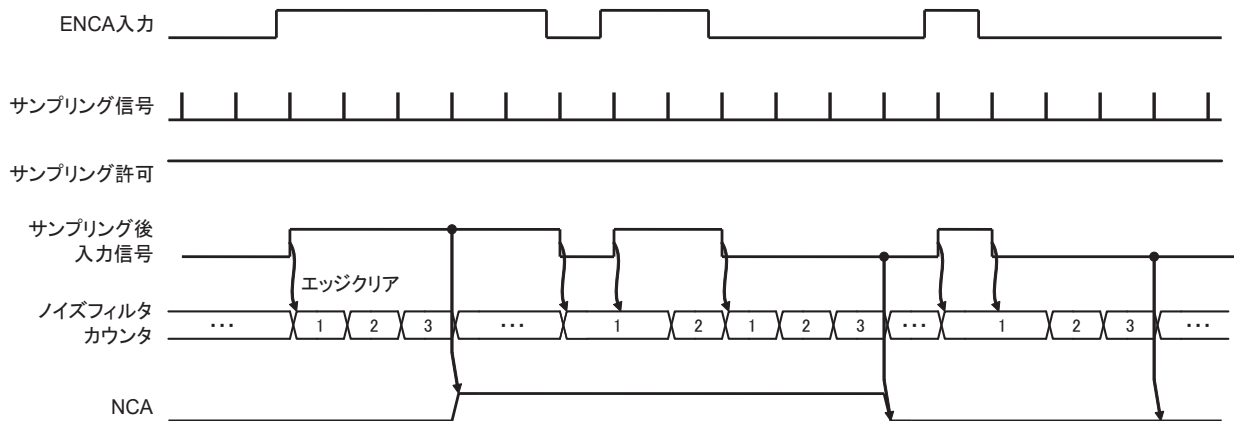
#### 19.5.1.3

##### 1. ノイズキャンセル時間

ENxINPCR<NCT[6:0]>でノイズキャンセル時間を設定します。ノイズキャンセル時間は以下の計算式で求められます。

ノイズキャンセル時間 : <NCT> × サンプルクロック周期

) <NCT> "0"

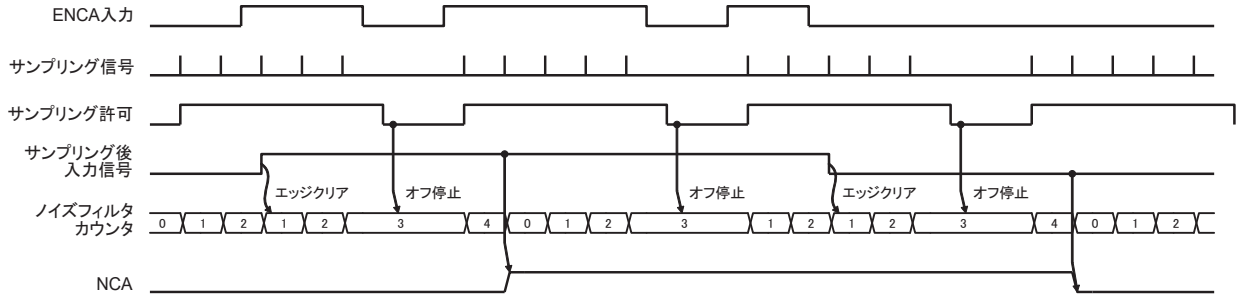


19-5

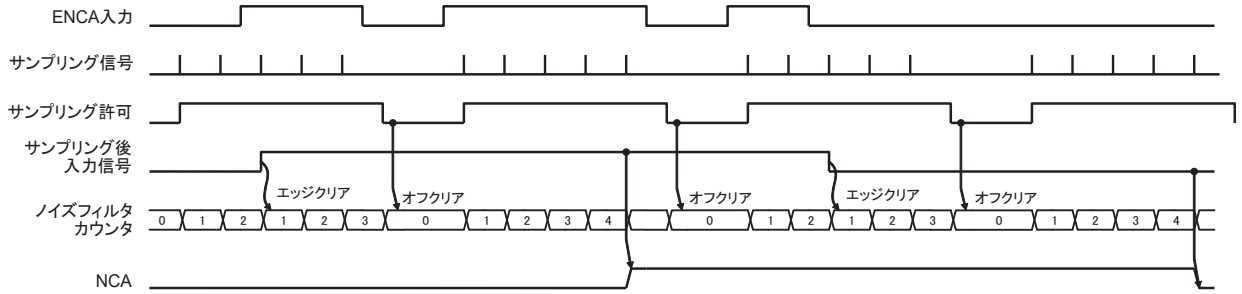
( <NCT> = 3 )

##### 2. PWM オン期間サンプリング時のノイズキャンセル

- ・ PWM オフ期間にノイズ除去タイマを停止 (ENxINPCR<SYNCCNCZEN> = "0")
- ・ PWM オフ期間にノイズ除去タイマをクリア (ENxINPCR<SYNCCNCZEN> = "1")

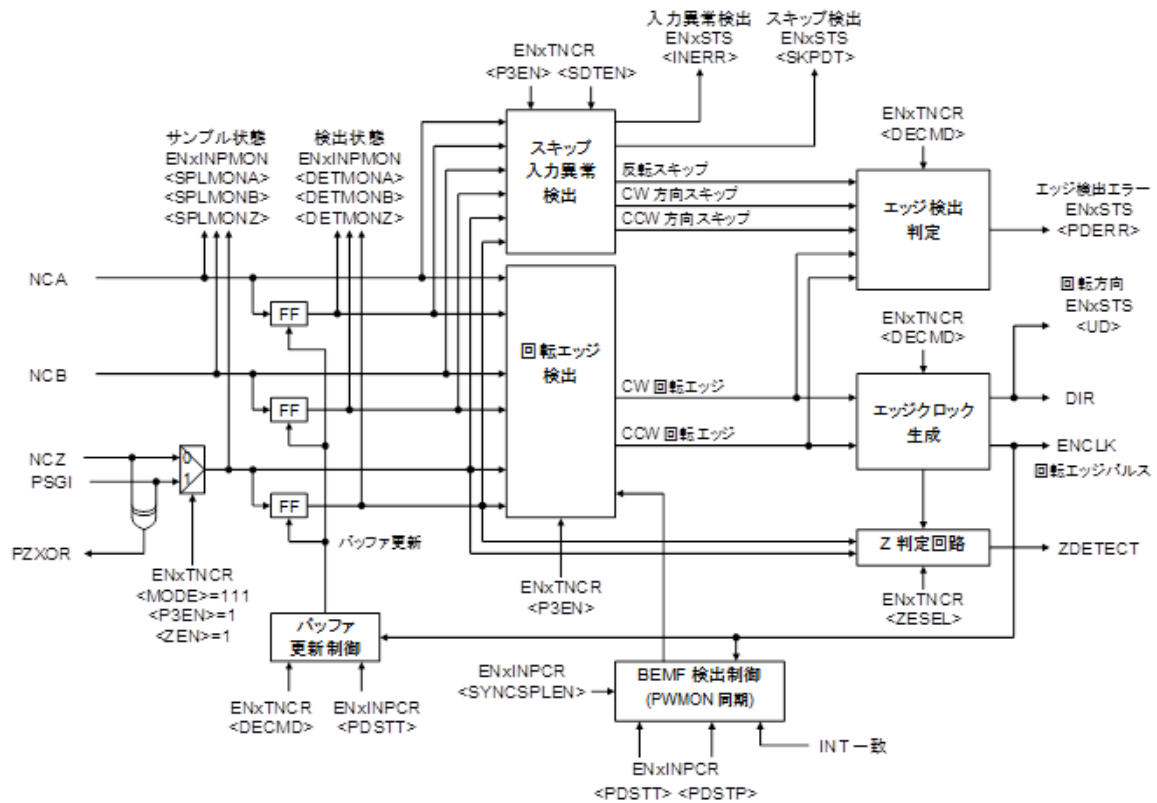


19-6 (PWM PWM <NCT> = 4)



19-7 (PWM PWM <NCT> = 4)

19.5.2



19-8

デコーダは、ノイズ除去後の2相または3相入力信号から、回転エッジ検出および回転方向判定を行います。また、インクリメンタル形エンコーダ接続時のZ相検出、単相入力時のZ信号のエッジ検出を行います。

19.5.2.1

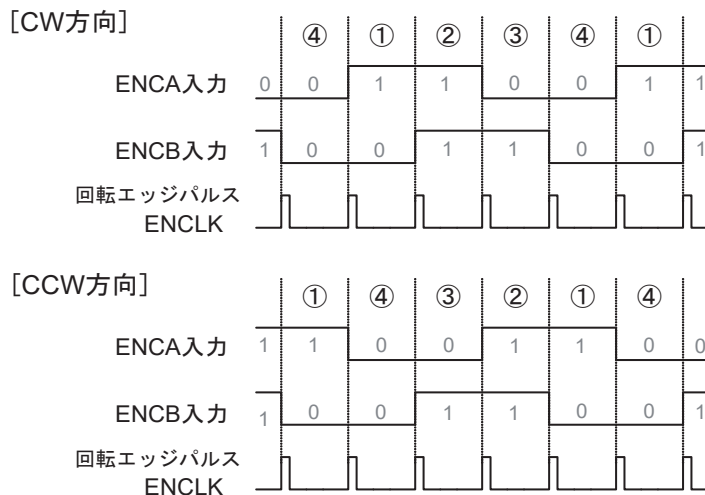
1. 2相デコード (ENxTNCR<P3EN> = "0")

エンコーダモードとセンサモード (<P3EN> = "0"設定) で対応します。

2相デコードでは4つの入力パターンの変化(回転エッジ)を検出します。

CW方向入力の場合、①→②、②→③、③→④、④→①の回転エッジを検出し、これらの回転エッジ検出でステータス ENxSTS<UD> は "1" になります。

CCW方向入力の場合、④→③、③→②、②→①、①→④の回転エッジを検出し、これらの回転エッジ検出でステータス ENxSTS<UD> は "0" になります。



19-9 2

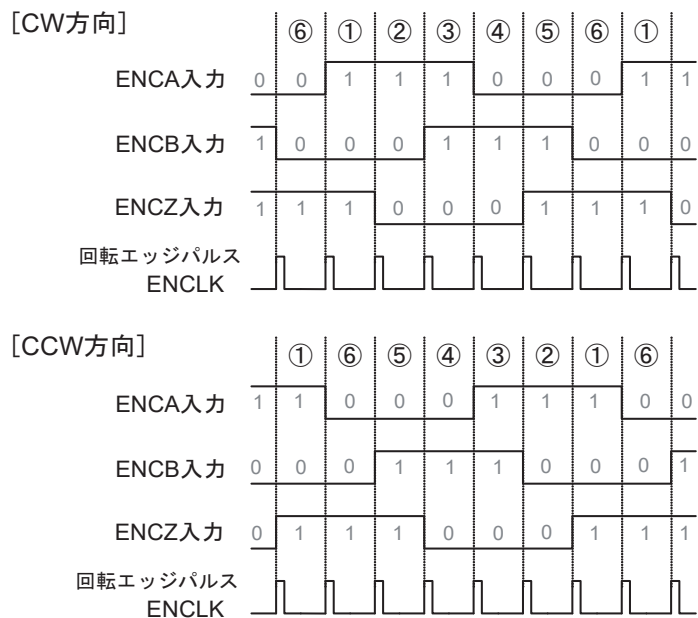
2. 3相デコード (ENxTNCR<P3EN> = "1")

センサモードのみ対応します。

3相デコードでは6つの入力パターンの変化(回転エッジ)を検出します。

CW方向の場合、①→②、②→③、③→④、④→⑤、⑤→⑥、⑥→①の回転エッジを検出し、これらの回転エッジ検出でステータス ENxSTS<UD>に"1"を設定します。

CCW方向の場合、⑥→⑤、⑤→④、④→③、③→②、②→①、①→⑥の回転エッジを検出し、これらの回転エッジ検出でステータス ENxSTS<UD>に"0"を設定します。



19-10 3

19.5.2.2 Z

Z入力のエッジを検出します。

## 1. エンコーダモード

CW 方向では立ち上がりエッジを検出し、CCW 方向では立ち下がりエッジを検出します。

## 2. タイマモード,位相カウンタモード

<ZESEL> で立ち上がりエッジ検出、立ち下がりエッジ検出および両エッジ検出を選択できます。

## 19.5.2.3

## 1. スキップ判定

ENxTNCR<SDTEN> = "1" で有効になります。

スキップ検出すると <SKPDT> が "1" にセットされます。

- ・ 2 相デコード (ENxSTS<P3EN> = "0") のスキップ検出の組み合わせ

反転スキップ検出 : ①→③、②→④、③→①、④→②

- ・ 3 相デコード (ENxSTS<P3EN> = "1") のスキップ検出の組み合わせ

CW 方向スキップ検出 : ①→③、②→④、③→⑤、④→⑥、⑤→①、⑥→②

CCW 方向スキップ検出 : ①→⑤、②→⑥、③→①、④→②、⑤→③、⑥→④

反転スキップ検出 : ①→④、④→①、②→⑤、⑤→②、③→⑥、⑥→③

## 2. 入力異常判定

センサモード(イベントカウント,タイマカウント,位相カウント)で 3 相デコード時に 3 入力全てが "0" または 全て "1" に変わるエッジ検出で入力異常と判定します。入力異常と判定すると ENxSTS<INERR> が "1" にセットされます。

## 19.5.2.4

ENxTNCR<DECMD> 設定で方向指定時に、指定外の方向検出でエラーと判定します。エラー判定は割り込み要因となります。

- ・ スキップ検出禁止 (ENxTNCR<SDTEN> = "0")

CW 回転検出時(ENxTNCR<DECMD> = "01") : CCW 回転エッジでエラー発生

CCW 回転検出時(ENxTNCR<DECMD> = "10") : CW 回転エッジでエラー発生

- ・ スキップ検出許可 (ENxTNCR<SDTEN> = "1")

CW 回転検出時(ENxTNCR<DECMD> = "01") : CCW 方向スキップ,反転スキップおよび CCW 回転エッジでエラー発生

CCW 回転検出時(ENxTNCR<DECMD> = "10") : CW 方向スキップ,反転スキップおよび CW 回転エッジでエラー発生

## 19.5.2.5

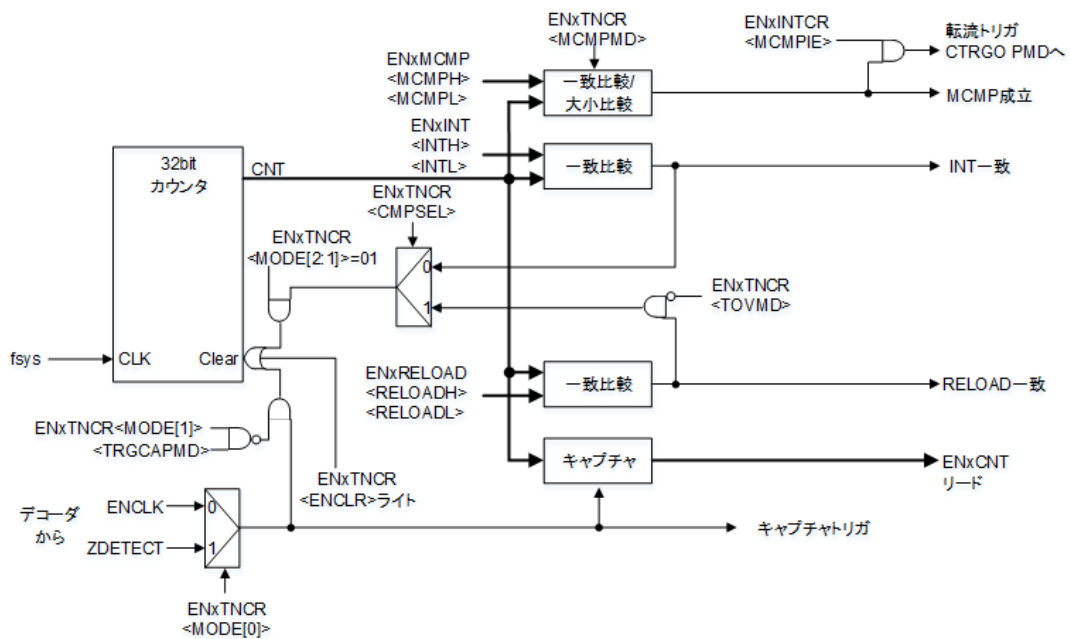
ENxTNCR<DECMD> = "00" に設定すると、バッファは常に有効となります。この場合、回転エッジ判定とスキップ判定は入力信号の変化で判定します。



エンコーダモードでZ検出許可 (ENxTNCR<ZEN> = "1" ) の場合、エンコーダ入力許可 (ENxTNCR<ENRUN> = "1" ) してからZエッジ検出するまでのINTコンペアとMCMPコンペアの一致信号は無視されます。

カウンタレジスタ (ENxCNT) をリードするとアップダウンカウンタの値を読み出せます。ENxINTCR<MCMPIE> = "1" に設定するとMCMPコンペア一致信号をPMD回路の転流トリガに使用できます。

19.5.3.2 ( )



19-12 ( ) ( ) ( )

システムクロック (fsys) で動作する 32 ビットカウンタ、3 種類 (RELOAD,INT,MCMP) のコンペア機能、キャプチャ機能で構成されます。

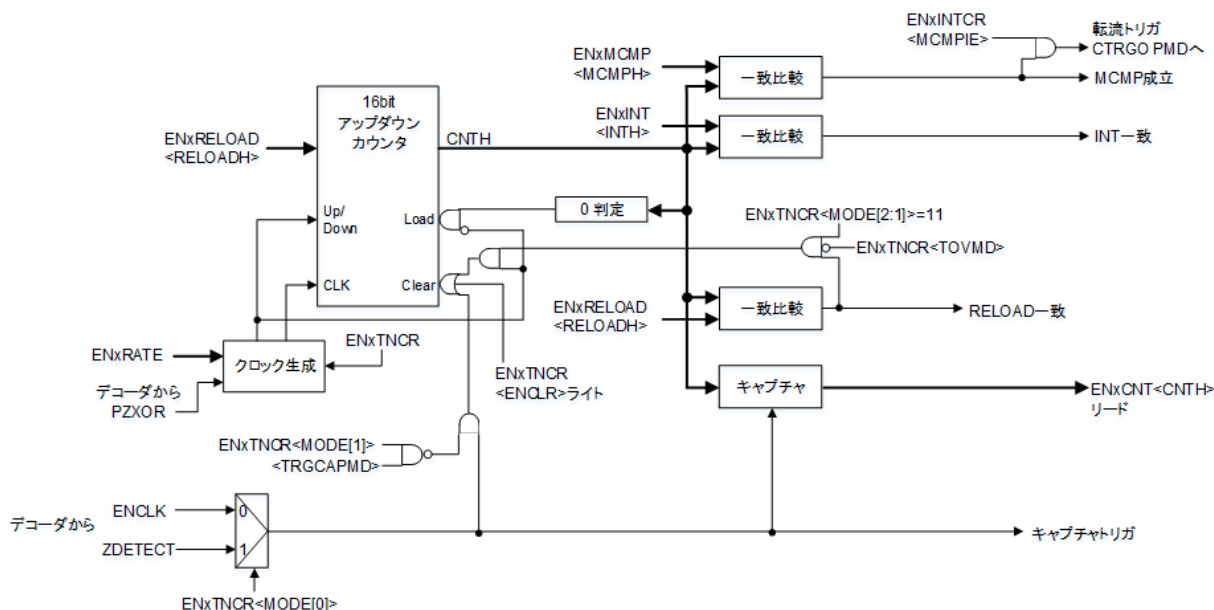
MCMP コンペア機能は一致比較と大小比較を選択できます。大小比較 (ENxTNCR <MCMPL> = "1" ) の場合は、ENxMCMP レジスタ設定することで比較を開始し、条件成立により MCMP 成立信号を出力して比較を終了します。

タイマモードでは、INT コンペアまたは RELOAD コンペアでカウンタクリアできます。

センサモード (タイマカウント) の場合は回転エッジ検出 (ENCLK) でカウンタキャプチャおよびカウンタクリアし、タイマモードの場合はZエッジ検出 (ZDETECT) でカウンタキャプチャおよびカウンタクリアできます。カウンタレジスタ (ENxCNT) をリードするとキャプチャ値を読み出せます。

ENxINTCR<MCMPIE> = "1" に設定すると MCMP コンペア一致信号を PMD 回路の転流トリガに使用できます。

## 19.5.3.3 ( )



## 19-13 ( ) ( ) ( )

ENxRATE<RATE>の設定によりカウンタクロックを生成するクロック生成回路、クロック生成回路からのクロック信号と方向信号で動作する16ビットアップダウンカウンタ、<RELOADH>、<INTH>、<MCMPPH>との3種類の一致比較器、キャプチャ機能で構成されます。

カウンタクロックはENxRATEレジスタで任意に設定します。

アップダウンカウントはENxTNCR<UDMD>で設定します。

アップカウント設定時はRELOADコンペアー一致でカウンタクリアし、ダウンカウント設定時は"0"一致で<RELOADH>の値をカウンタにロードします。

センサモード(位相カウント)の場合は回転エッジ検出(ENCLK)でカウンタキャプチャおよびカウンタクリアし、位相カウンタモードの場合はZエッジ検出(ZDETECT)でカウンタキャプチャおよびカウンタクリアできます。カウンタレジスタ(ENxCNT)をリードするとキャプチャ値を読み出せます。

ENxINTCR<MCMPPEN>="1"に設定するとMCMPコンペアー一致信号をPMD回路の転流トリガに使用できます。

19.5.4

6 種類の割り込み要因から 2 種類の割り込み要求を出力します。割り込み要求は割り込み制御レジスタ(ENxINTCR)で要因別に出力許可し、発生要因は割り込みフラグ(ENxINTF)で確認できます。

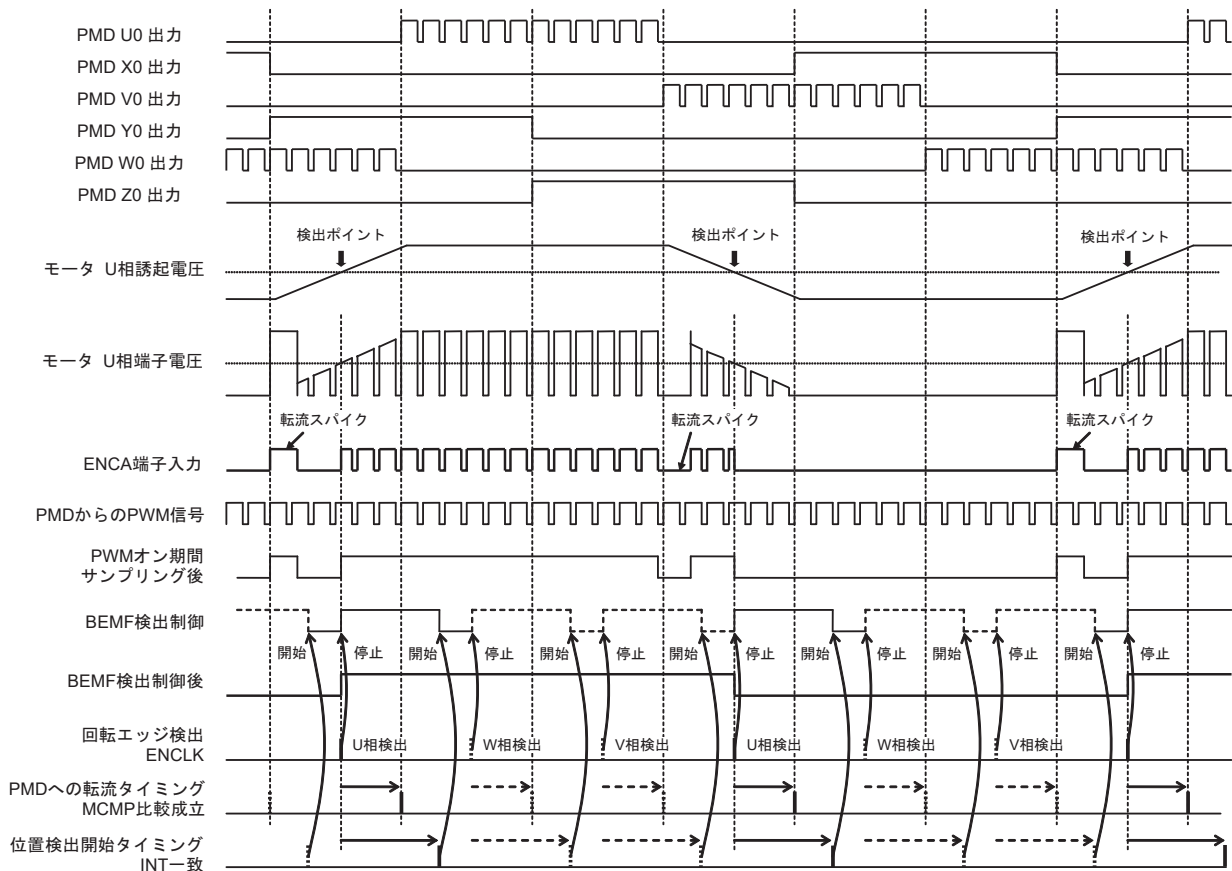
割り込みフラグレジスタ(ENxINTF)は割り込み要因の発生でセットされ、レジスタリードでクリアされます。

19-1

			ENxINTCR	ENxINTF	
	ENxTNCR <ENDEV> 1 ~ 128	( )	<TPLSIE>	<TPLSF>	INTENCx0
	(ENCZ )	( )	<CAPIE>	<CAPF>	INTENCx0
	(ENCLK)	( )			
	(SKPDT) (PDERR)		<ERRIE>	<ERRF>	INTENCx0
INT	ENxINT		<CMPIE>	<INTCPF>	INTENCx1
RELOAD	ENxRELOAD	) (	<RLDIE>	<RLDCPF>	INTENCx1
MCMP	ENxTNCR<MCMPMD> = "0" ENxMCMP	( )	<MCMPIE>	<MCMPF>	INTENCx1
	<MCMPMD> = "1" ENxMCMP	) (			

## 19.6 DC

PMD からの矩形波にて駆動しているブラシレス DC モータの、BEMF ゼロクロス検出する場合の波形を示します。



19-14 120

センサモード(タイマカウント,位相カウント)で PWM 同期サンプリングを許可( ENxINPCR <SYNCSPLN>="1")すると PMD で矩形波駆動しているブラシレス DC モータの誘起電圧(BEMF)のゼロクロス検出に対応できます。この場合、入力回路は PWM 同期サンプリングで入力信号から PWM 成分を除去します。また、デコーダ回路は BEMF 検出制御で転流スパイクを回避できます。これらにより、入力信号から BEMF のゼロクロスを回転エッジパルス(ENCLK)として検出できます。

BEMF 検出制御は、回転エッジパルスでエッジ検出を停止(中断)し、INT コンペア成立でエッジ検出を開始(再開)します。また、コマンドストップ(ENxINPCR<PDSTP> = "1")、コマンドスタート(ENxINPCR<PDSTT> = "1")も可能です。

BEMF ゼロクロス検出は、カウンタのキャプチャ機能および3種類のコンペア機能を次のように使用して PMD 回路と連携動作します。PMD と連携動作するために、転流タイミング出力を有効(ENxINTCR<MCMPIE>="1")にしてください。

- ・ キャプチャ：回転エッジパルス(ENCLK)でキャプチャ(ゼロクロス検出間隔の測定)
- ・ MCMP コンペア：PMD の転流タイミング(設定例：キャプチャ値×0.5)
- ・ INT コンペア：位置検出開始タイミング(設定例：キャプチャ値×0.75)
- ・ RELOAD コンペア：位置検出タイムアウト(設定例：キャプチャ値×2.0)

## 20

## (POR)

パワーオンリセット回路(POR)は、電源投入時にパワーオンリセット信号を発生します。

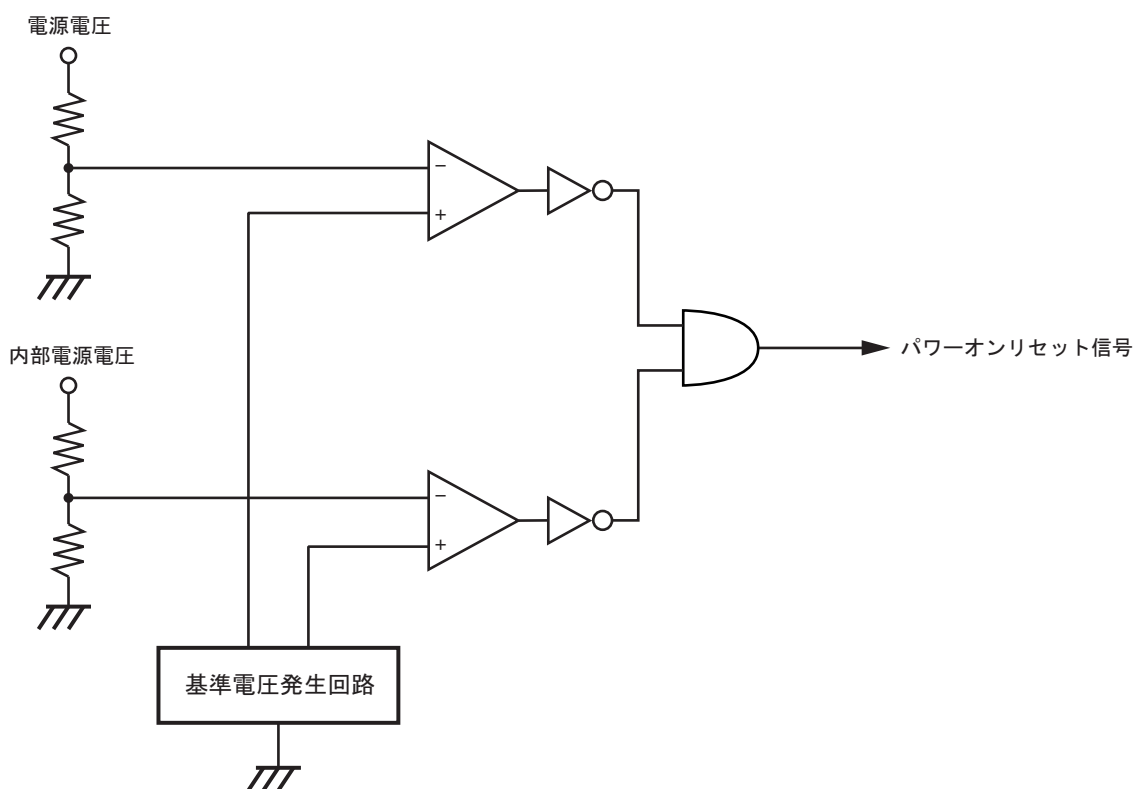
電源電圧とは DVDD5 を、内部電源電圧とは VOUT12 を指しています。

)

## 20.1

パワーオンリセット回路は、基準電圧発生回路、コンパレータで構成されます。

電源電圧および内部電源電圧をラダー抵抗によって分圧した電圧それぞれを、基準電圧発生回路が発生した基準電圧とコンパレータで比較します。



20-1

## 20.2

電源投入時、電源電圧が POR 解除電圧以下の間、パワーオンリセット信号は有効です。パワーオンリセット信号が無効となるのは、電源電圧が  $3.0 \pm 0.2 \text{ V}$  を超えるタイミングです。

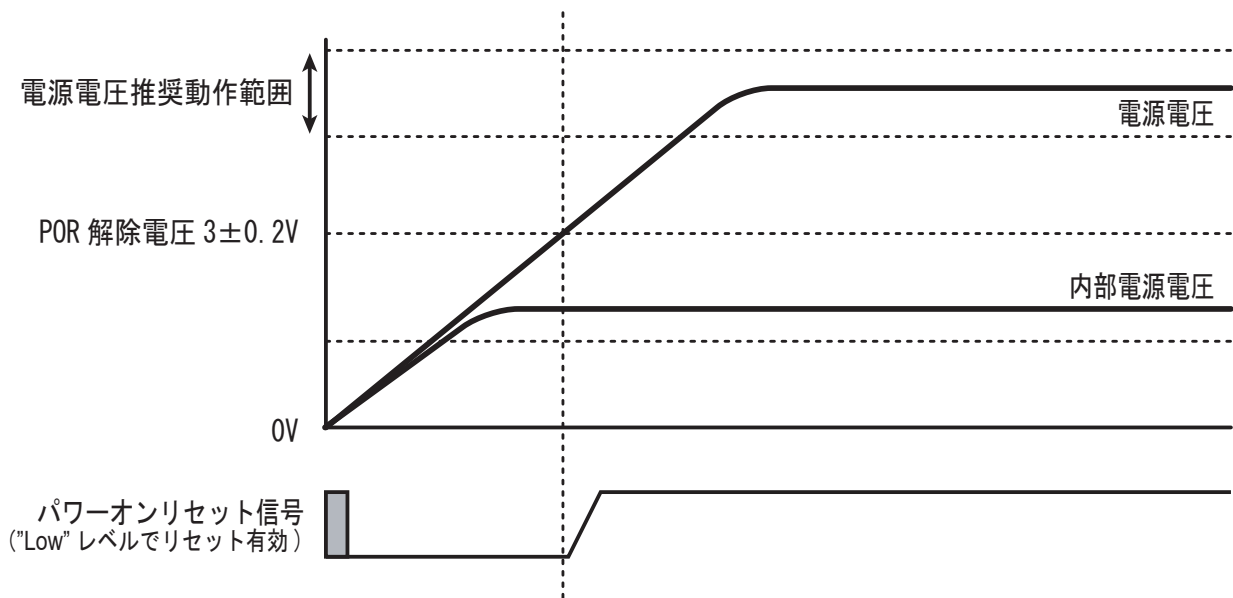
電源下降時、電源電圧が POR 検出電圧以上の間、パワーオンリセット信号は無効です。パワーオンリセット信号が有効となるのは、電源電圧が  $2.8 \pm 0.2 \text{ V}$  を下回るタイミングです。

パワーオンリセット信号が有効の間、CPU および周辺機能はリセットされます。

リセットについては、リセット動作の章を参照してください。

また、動作タイミングの詳細については電源検出回路(VLTD)も参照してください。

) POR            POR



20-2

## 21

## (VLTD)

電圧検出回路(VLTD)は、電源電圧の低下を検出し、電圧検出リセット信号を発生します。

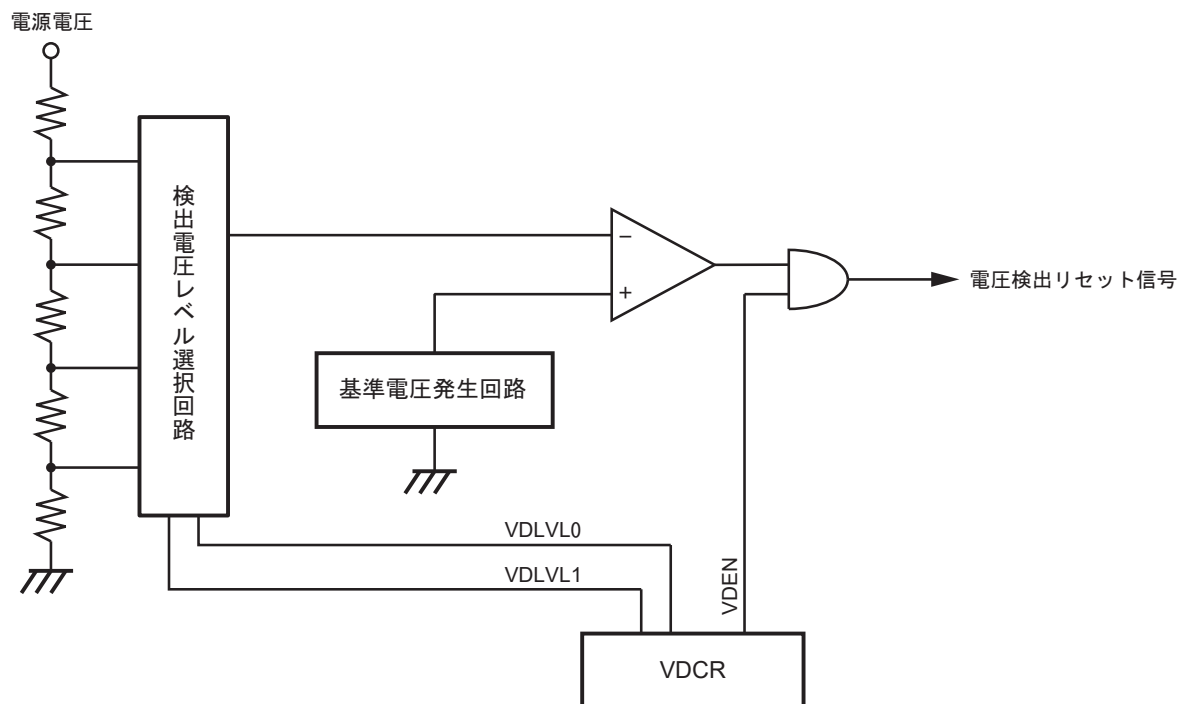
電源電圧とは、DVDD5を指しています。

)

## 21.1

電圧検出回路は、基準電圧発生回路、検出電圧レベル選択回路、コンパレータ、制御レジスタから構成されます。

電源電圧はラダー抵抗で分圧され、検出電圧レベル選択回路に入力されます。検出電圧レベル選択回路で検出電圧(VDLVL)に応じた電圧が選択され、コンパレータで基準電圧と比較されます。



21-1

## 21.2

### 21.2.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)
	VDCR	0x0000

### 21.2.2 VDCR ( )

	31	30	29	28	27	26	25	24
bit symbol								
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol								
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol							-	
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol						VDLVL		VDEN
	0	0	0	0	0	0	1	1

Bit	Bit Symbol	Type	
31-3		R	"0"
2-1	VDLVL[1:0]	R/W	00: Reserved 01: 4.1±0.2V 10: 4.4±0.2V 11: 4.6±0.2V
0	VDEN	R/W	0: 1:

) VDCR

## 21.3

### 21.3.1

電圧検出回路は、電圧検出制御レジスタで制御されます。

### 21.3.2

電圧検出回路は、検出電圧レベル選択ビット  $VDCR\langle VDLVL[1:0]\rangle$ 、電圧検出の許可/禁止  $VDCR\langle VDEN\rangle$  で設定します。電圧検出の許可/禁止を設定し、電源電圧が検出電圧 ( $VDLVL[1:0]$ ) を下回ったとき、電圧検出リセット信号が有効になります。

リセットについては、リセット動作の章を参照してください。

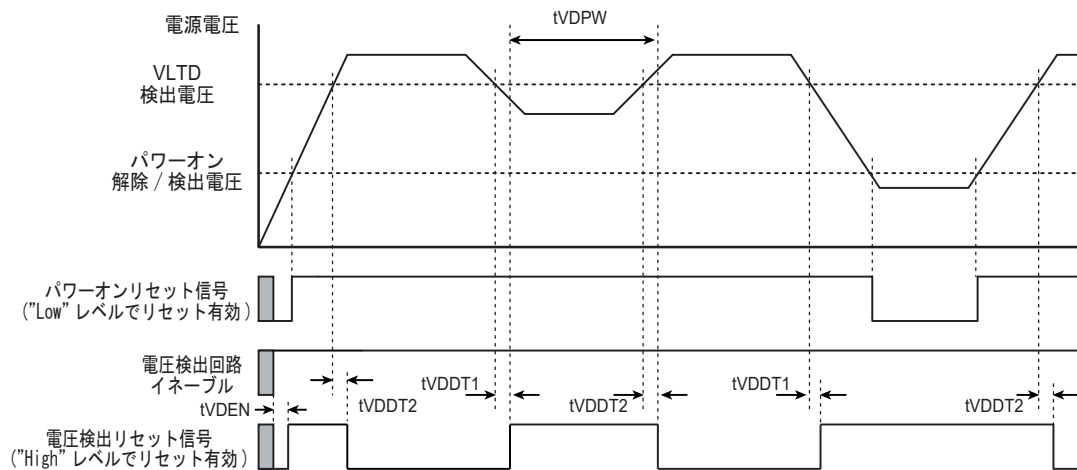
#### 21.3.2.1

$VDCR\langle VDEN\rangle$  はパワーオンリセット、外部リセット解除後、“1” にセットされて電圧検出動作が許可されます。

)  $VDCR\langle VDEN\rangle = "0"$  ( ) "1" ( )

#### 21.3.2.2

$VDCR\langle VDLVL[1:0]\rangle$  で検出電圧を選択します。



21-2

) VLTD VLTD

	Min	Typ.	Max	
tVDEN	-	40	-	μs
tVDDT1	-	40	-	
tVDDT2	-	40	-	
tVDPW	45	-	-	



22

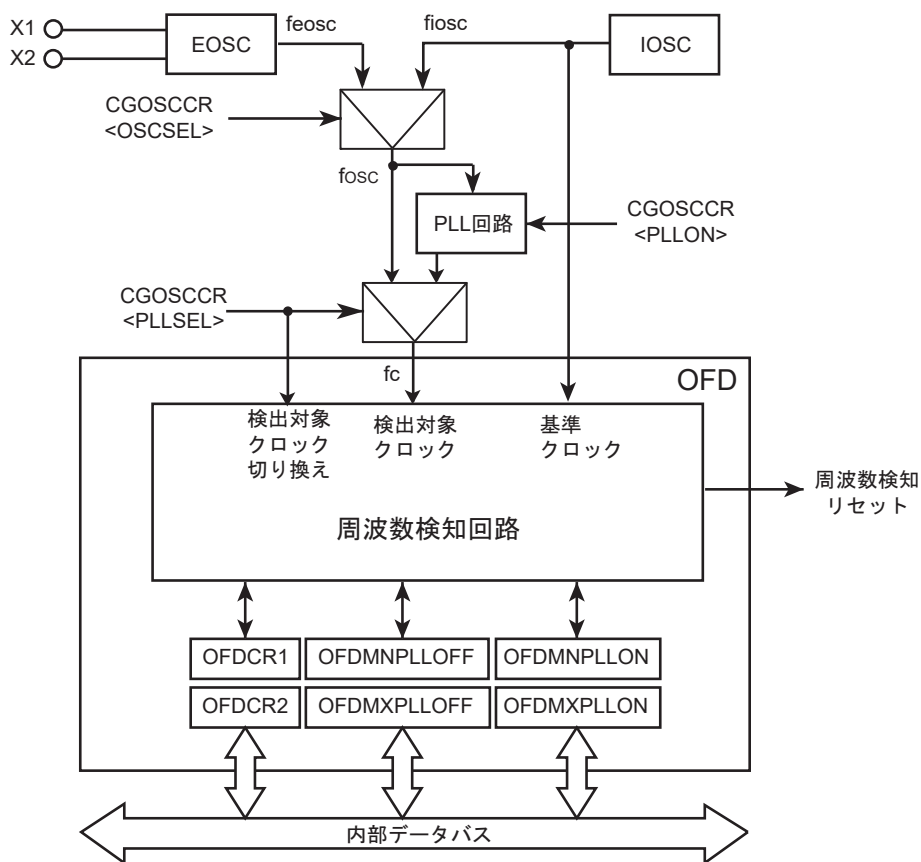
(OFD)

周波数検知回路(OFD ; Oscillation Frequency Detector)はクロック周波数の異常を検出する回路です。OFDを使用することで、高調波、低調波、停止といったクロックの異常を検出することができます。

OFDは、基準となるクロックを用いて対象のクロックを観測し、異常を検出するとリセットを発生します。本製品の基準クロックは内蔵高速発振クロック(fiosc)、検出対象クロックは外部高速発振器クロック(feosc)です。

)

22.1



22-1

## 22.2

### 22.2.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)
1	OFDCR1	0x0000
2	OFDCR2	0x0004
PLL OFF	OFDMNPLLOFF	0x0008
PLL ON	OFDMNPLLON	0x000C
PLL OFF	OFDMXPLLOFF	0x0010
PLL ON	OFDMXPLLON	0x0014

22.2.1.1 OFDCR1( 1)

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

Bit	Bit Symbol	Type							
31-8		R	"0"						
7-0	OFDWEN[7:0]	R/W	0x06: 0xF9: <table style="margin-left: 40px;"> <tr> <td>0xF9</td> <td>OFDCR1</td> <td>0x06</td> </tr> <tr> <td>0x06</td> <td>0xF9</td> <td>0x06</td> </tr> </table>	0xF9	OFDCR1	0x06	0x06	0xF9	0x06
0xF9	OFDCR1	0x06							
0x06	0xF9	0x06							

) OFDCR1 (RESET "L" ) VLTD

22.2.1.2 OFDCR2( 2)

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	OFDEN[7:0]	R/W	0x00: 0xE4: 0x00 0xE4

) OFDCR2 (RESET "L" ) VLTD

22.2.1.3 OFDMNPLLOFF ( (PLL OFF ) )

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

Bit	Bit Symbol	Type	
31-9		R	"0"
8-0	OFDMNPLLOFF[8:0]	R/W	

- 1)
- 2) OFDMNPLLOFF (RESET "L" ) VLTD

22.2.1.4 OFDMNPLLON ( (PLL ON ) )

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

Bit	Bit Symbol	Type	
31-9		R	"0"
8-0	OFDMNPLLON [8:0]	R/W	

- 1)
- 2) OFDMNPLLON (RESET "L" ) VLTD

22.2.1.5 OFDMXPLOFF ( (PLL OFF ) )

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

Bit	Bit Symbol	Type	
31-9		R	"0"
8-0	OFDMXPLOFF[8:0]	R/W	

- 1)
- 2) OFDMXPLOFF (RESET "L" ) VLTD

22.2.1.6 OFDMXPLLON ( (PLL ON ) )

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

Bit	Bit Symbol	Type	
31-9		R	"0"
8-0	OFDMXPLLON [8:0]	R/W	

- 1)
- 2) OFDMXPLLON (RESET "L" ) VLTD

## 22.3

### 22.3.1

外部リセット入力( $\overline{\text{RESET}}$  端子への"L"入力)、パワーオンリセットまたは VLTD リセットにより周波数検知回路(OFD)の制御レジスタは初期化され、OFDCR1 以外のレジスタには書き込みができない状態になります。OFDCR1 に"0xF9"を書き込むことにより、OFDCR1 以外のレジスタに書き込みができるようになります。

検知する周波数の範囲は、対象クロックごとに OFDMNPLLON と OFDMXPLLON 並びに OFDMNPLLOFF と OFDMXPLLOFF で設定します。これらは CGPLLSEL<PLLSEL> 設定により自動的に切り換わります。なお、PLL オンの状態 CGPLLSEL<PLLSEL> = "1"で周波数検知リセットが発生した場合、PLL がオフ状態 CGPLLSEL<PLLSEL> = "0"となり、検知周波数設定レジスタは OFDMNPLLON と OFDMXPLLON から OFDMNPLLOFF と OFDMXPLLOFF に自動的に切り換わります。

OFDCR1 に書き込み許可コード "0xF9" が設定された状態で OFDCR2 に "0xE4" を書き込むと OFD はイネーブルとなり動作します。

誤書き込み防止のため、すべてのレジスタの設定後、OFDCR1 に"0x06"を設定し書き込み禁止にしてください。また、設定の変更は動作が停止した状態で行ってください。

周波数検知動作がイネーブル状態(OFDCR2 = "0xE4")で STOP モードを起動した場合、OFD は自動的にディセーブルとなります。この状態で STOP モードが解除されると OFD は STOP モード解除後のウォーミングアップ期間が終了してからイネーブルとなります。

) CGPLLSEL	PLL	CGOSCCR<OSCSSEL>	fiosc
feosc		OFD	

### 22.3.2

周波数検知回路(OFD)は NORMAL モードと IDLE モードのときのみ使用可能です。詳細は表 22-1 を参照してください。

内部高周波発振に遷移する際は、OFD を停止させてください。

22-1

	(OFDCR2 = "0xE4" )	( $\overline{\text{RESET}}$ MODE X1,X2 )
NORMAL		
IDLE		
STOP		
( )		
SYSRESETREQ		
( $\overline{\text{RESET}}$ "L" )		
VLTD		-

### 22.3.3

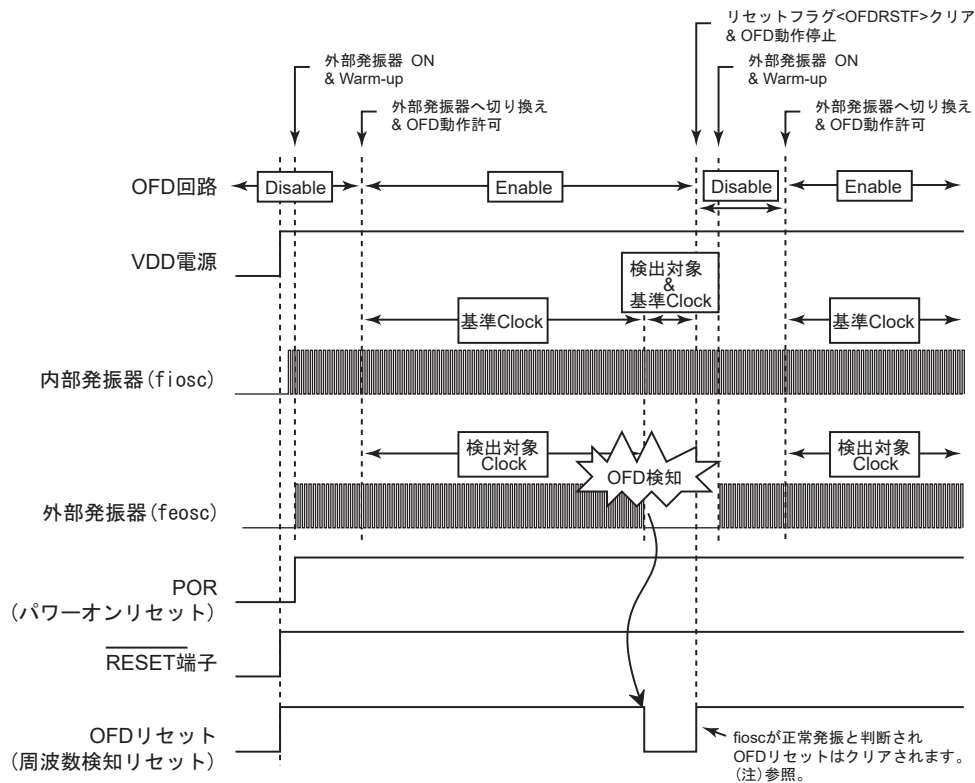
周波数検知回路(OFD)では、動作開始から検知開始まで検知周期 2 周期分の時間が必要で、検知周期は 128/基準クロック周波数です。

検出対象クロックが OFDMNPLLON、OFDMNPLLOFF と OFDMXPLLON、OFDMXPLLOFF で設定した周波数範囲を超えると OFD はリセットを発生します。OFD が異常を検知してからリセットを発生するまで検知周期 1 周期分の時間が必要です。OFD が発生するリセットでは OFD 自身はリセットされず検知動作を継続します。

OFD が発生するリセットにより fosc は内部高速発振クロック fosc に初期化され、検出対象クロック fc は PLL OFF 時の fosc に切り換わるため、検出対象クロックと基準クロックが同じになって周波数範囲が設定値以内に納まるので、周波数検知リセットは解除されます。

なお、システムクロック fc として内部高速発振クロック fosc を選択した場合は、OFD は機能しません。

1)	CGRSTFLG			CGRSTFLG	
2)	OFD	OFDMNPLLOFF	OFDMXPLLOFF	10MHz	8MHz



(注) 検出対象クロックの設定値 (OFDMNPLLOFF、OFDMXPLLOFF) を10MHz 以外、例えば8MHz の設定値にして検知回路を動作させ周波数検知リセットが掛かった場合、対象クロックが異常と判断し、周波数検知リセットを発生し続ける場合があります。

## 22.3.4

検知周波数には、基準クロックの発振精度によって、検出範囲と非検出範囲が存在します。検出範囲と非検出範囲の間の周波数では、検出されるかどうかは未確定です。

検知周波数上限値、下限値は、検出対象クロックと基準クロックの最大誤差より計算します。

OFDMNPLLON、OFDMNPLLOFF と OFDMXPLLON、OFDMXPLLOFF を決める時の計算結果の丸め方により、検出範囲と非検出範囲の上限値、下限値が下記のように変わります。検出対象となるクロックのばらつきに応じて選択してください。

- OFDMXPLLON/OFDMXPLLOFF を切上げ、OFDMNPLLON/OFDMNPLLOFF を切捨てた場合  
検出対象クロックに対して、非検出範囲の上限が高くなり、非検出範囲の下限が低くなります。
- OFDMXPLLON/OFDMXPLLOFF を切捨て、OFDMNPLLON/OFDMNPLLOFF を切上げた場合  
検出対象クロックに対して、非検出範囲の上限が低くなり、非検出範囲の下限が高くなります。

基準クロック誤差を±5%、検出対象クロック誤差±5%(非検出範囲)を許容する場合の OFDMXPLLOFF/OFDMNPLLOFF の設定値の算出方法を以下に示します。この例では、OFDMXPLLOFF を切上げ、OFDMNPLLOFF を切り捨てます。(①～⑧は、「図 22-3 検出周波数範囲例(10MHz の場合)」に対応)

	10MHz ± 5%	Max 10.5MHz Min 9.5MHz	----- -----
	9.7MHz ± 5%	Max 10.185MHz Min 9.215MHz	----- -----

$$\text{OFDMXPLLOFF} = \text{③} \div \text{⑤} \times 32 = 36.46\dots = 37 \text{ (小数点以下切上げ)} = 0x25$$

$$\text{OFDMNPLLOFF} = \text{②} \div \text{⑥} \times 32 = 29.85\dots = 29 \text{ (小数点以下切捨て)} = 0x1D$$

このときの検出範囲は以下のように求められます。

$$\text{①} = \text{⑤} \times \text{OFDMNPLLOFF} \div 32 = 8.35$$

$$\text{④} = \text{⑥} \times \text{OFDMXPLLOFF} \div 32 = 11.78$$

また、このときの非検出範囲は以下のようになります。

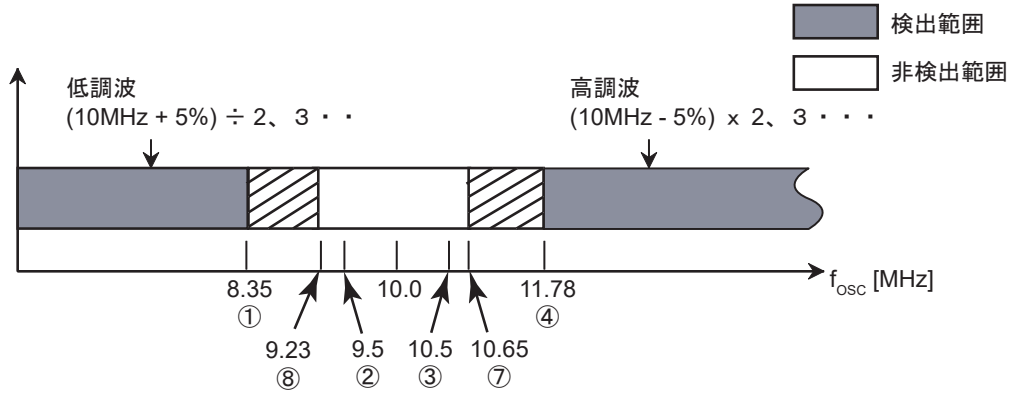
$$\text{⑦} = \text{⑤} \times \text{OFDMXPLLOFF} \div 32 = 10.65$$

$$\text{⑧} = \text{⑥} \times \text{OFDMNPLLOFF} \div 32 = 9.23$$

よって、レジスタ OFDMXPLLOFF に 0x25、OFDMNPLLOFF に 0x1D を設定すると、11.78MHz 以上と 8.35MHz 以下の発振周波数を検知した場合、周波数検知リセットを発生します。

また、9.23MHz から 10.65MHz の発振周波数を検知した場合、周波数検知リセットは発生しません。

図 22-3 に、このときの検出範囲/非検出範囲を示します。

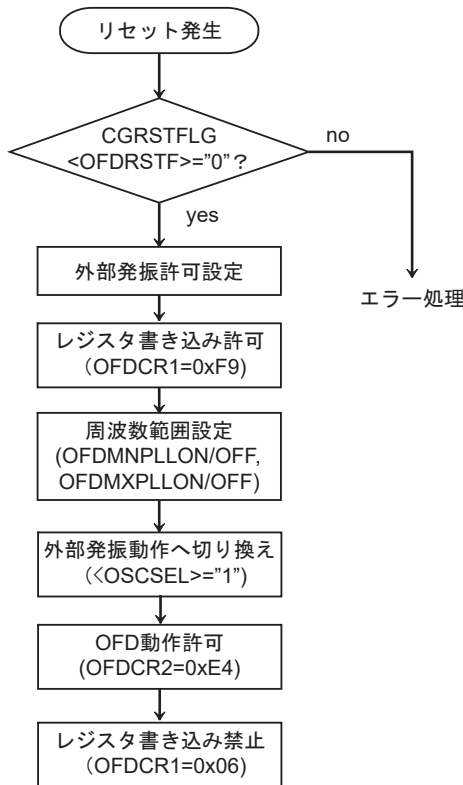


22-3 (10MHz )

22.3.5

周波数検知回路(OFD)を使用する場合の動作手順例を示します。

リセット発生後、CGRSTFLG でリセットの要因を確認します。要因が周波数検知リセットでなければ、外部発振を有効にし、OFD を使用するためのレジスタ設定を行い動作を許可します。



22-4

23

(WDT)

ウォッチドッグタイマは、ノイズなどの原因により CPU が誤動作(暴走)を始めた場合、これを検出し正常な状態に戻すことを目的としています。

暴走を検出した場合、INTWDT 割り込みを発生またはマイコンをリセットします。

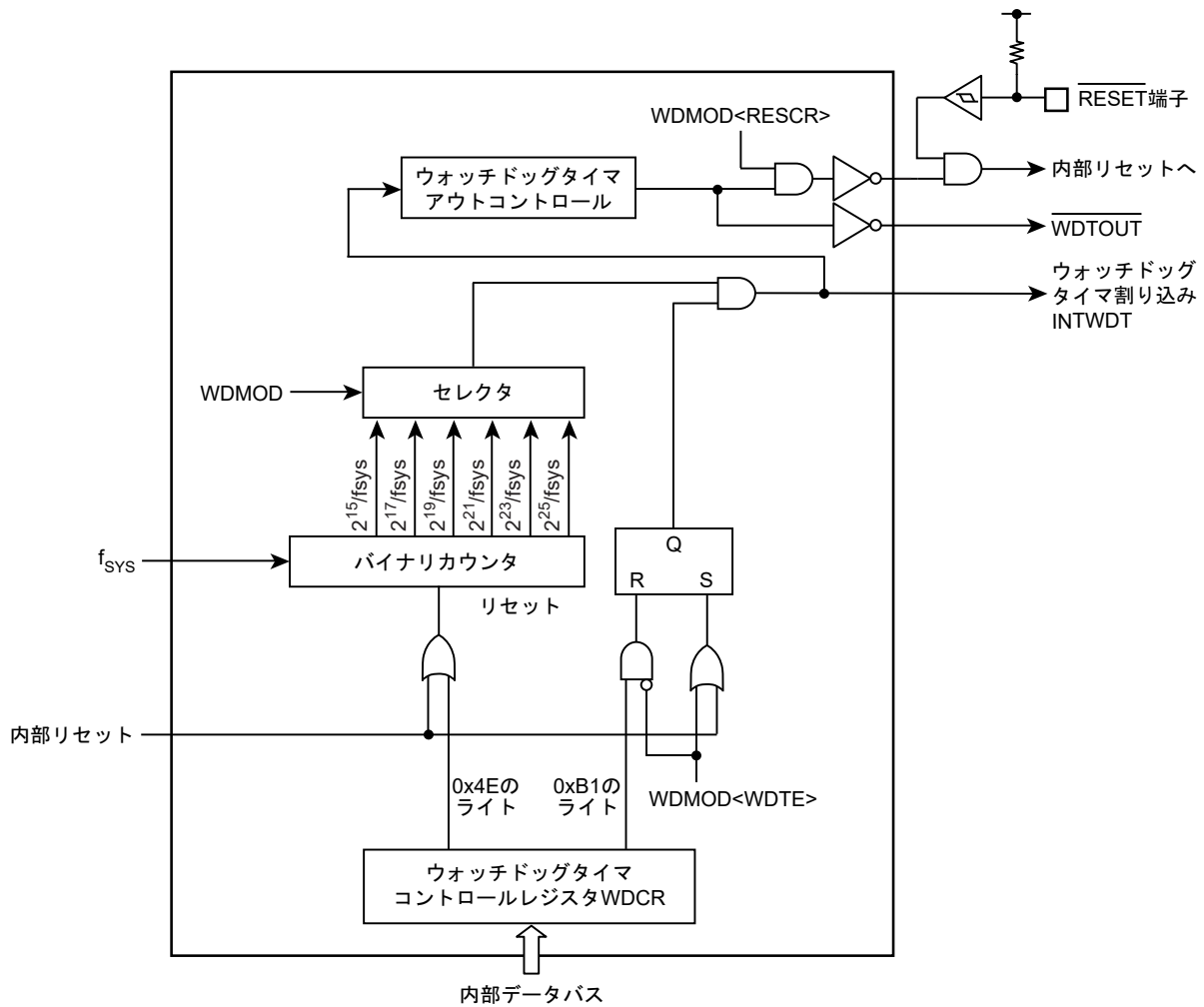
) INTWDT (NMI)

また、外部周辺装置に対しては、ウォッチドッグタイマアウト端子(WDTOUT)より"Low"を出力して暴走の検出を知らせます。

) TMPM475FDFG/FZFG/FYFG (WDTOUT)

23.1

図 23-1 にウォッチドッグタイマのブロック図を示します。



23-1

## 23.2

### 23.2.1

制御レジスタとアドレスは以下のとおりです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照ください。

		Address(Base+)
	WDMOD	0x0000
	WDCR	0x0004

#### 23.2.1.1 WDMOD( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	WDTE	WDTP			-	I2WDT	RESCR	-
	1	0	0	0	0	0	1	0

Bit	Bit Symbol	Type	
31-8		R	"0"
7	WDTE	R/W	/ 0: 1: (0xB1)
6-4	WDTP[2:0]	R/W	000: 2 <sup>15</sup> /f <sub>SYS</sub> 100: 2 <sup>23</sup> /f <sub>SYS</sub> 001: 2 <sup>17</sup> /f <sub>SYS</sub> 101: 2 <sup>25</sup> /f <sub>SYS</sub> 010: 2 <sup>19</sup> /f <sub>SYS</sub> 110: 011: 2 <sup>21</sup> /f <sub>SYS</sub> 111:
3		R	"0"
2	I2WDT	R/W	IDLE 0: 1:
1	RESCR	R/W	0: INTWDT                      ( ) 1:
0		R/W	"0"

) INTWDT

(NMI)

23.2.1.2 WDCR( )

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

Bit	Bit Symbol	Type	
31-8		R	"0"
7-0	WDCR	W	/
			0xB1:
			0x4E:
			:Reserved

## 23.3

### 23.3.1

ウォッチドッグタイマは、システムクロック  $f_{sys}$  を入力クロックとするバイナリカウンタで構成されています。検出時間は  $WDMOD<WDTP[2:0]>$  によって  $2^{15}, 2^{17}, 2^{19}, 2^{21}, 2^{23}$  および  $2^{25}$  から選択します。検出時間経過後にウォッチドッグタイマ割り込み(INTWDT)が発生し、ウォッチドッグタイマアウト端子(WDTOUT)より"Low"が出力されます。

ノイズなどの原因による CPU の暴走を検出するために、ソフトウェア(命令)でウォッチドッグタイマ用のバイナリカウンタを INTWDT 割り込みが発生する前にクリアするように設定しておきます。クリアが行われなかった場合、INTWDT によってマスク不能割り込みが発生しますので、CPU は暴走を検知し、暴走対策プログラムにより正常な状態に戻すことができます。

また、ウォッチドッグタイマアウト端子を外部周辺装置のリセットなどへ接続することにより、CPU の誤動作(暴走)に対処することができます。

) TMPM475FDFG/FZFG/FYFG (WDTOUT)

### 23.3.2

ウォッチドッグタイマは、リセット解除後ただちに動作を開始します。使用しない場合はディセーブルの処理を行ってください。

ウォッチドッグタイマは高速クロックが停止するモードでは使用できません。以下に示すモードに遷移する前にディセーブルしてください。IDLE モード中は  $WDMOD<I2WDT>$  の設定に従います。

- STOP mode

また、デバッグモード中は自動的にバイナリカウンタが停止します。

### 23.3.3

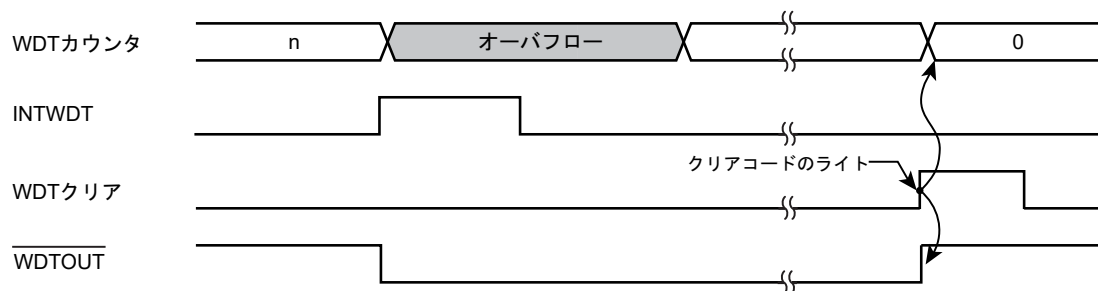
#### 23.3.3.1 INTWDT

図 23-2 に INTWDT 割り込み発生( $WDMOD<RESCR>="0"$ )の場合の動作を示します。

バイナリカウンタのオーバフローにより INTWDT 割り込みが発生します。INTWDT 割り込みはマスク不能割り込み(NMI)の要因であるため、CPU はマスク不能割り込みを検出し処理を行います。

INTWDT 割り込み発生と同時にウォッチドッグタイマアウト(WDTOUT)より"Low"を出力します。WDTOUT は、ウォッチドッグタイマのクリア(WDCR レジスタにクリアコード 0x4E をライト)により"High"に戻ります。

) TMPM475FDFG/FZFG/FYFG (WDTOUT)

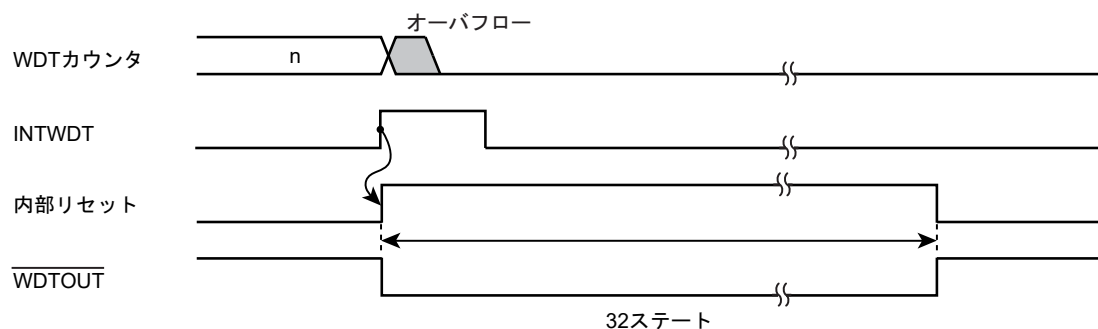


23-2 INTWDT

## 23.3.3.2

図 23-3 に内部リセット発生(WDMOD<RESCR>="1")の場合の動作を示します。

バイナリカウンタのオーバーフローによりマイコンをリセットします。この場合、32 ステートの期間リセットを行います。



23-3

## 23.4

## 23.4.1

WDMOD<WDTE>に"0"を設定したあと、WDCR レジスタにディセーブルコード(0xB1)を書き込むとウォッチドッグタイマはディセーブルになり、バイナリカウンタはクリアされます。

## 23.4.2

WDMOD<WDTE>に"1"を設定します。

## 23.4.3

WDCR レジスタにクリアコード(0x4E)を書き込むと、バイナリカウンタはクリアされ、再カウントします。

## 23.4.4

検出時間を WDMOD<WDTP[2:0]>に設定します。

例えば、検出時間を  $2^{21}/f_{\text{SYS}}$  に設定する場合、WDMOD<WDTP[2:0]>に"011"を設定します。

# 24 (FLASH)

フラッシュメモリについて、構成およびその動作を説明します。本文中の「1ワード」は、32ビットをあらわします。

## 24.1

### 24.1.1

本製品の内蔵するフラッシュメモリの容量と構成は、表 24-1 と表 24-2 の通りです。

24-1

	(KB)					(s) ( )	(ms) ( )					
		(KB)		(KB)			(Byte)		1	1	1	
TMPM475FDFG	512	512	1	32	16	4096	128	8.0	115	920	115	115
TMPM475FZFZFG	384	384	1		12		96	6.0				
TMPM475FYFG	256	256	1		8		64	4.0				

)

24-2

	Block	( )	( ( ) )	(KByte)				
						TMPM475FDFG	TMPM475FZFZFG	TMPM475FYFG
0	0	0x0000_0000 ~ 0x0000_7FFF	0x5E00_0000 ~ 0x5E00_7FFF	32	8	*	*	*
	1	0x0000_8000 ~ 0x0000_FFFF	0x5E00_8000 ~ 0x5E00_FFFF	32	8	*	*	*
	2	0x0001_0000 ~ 0x0001_7FFF	0x5E01_0000 ~ 0x5E01_7FFF	32	8	*	*	*
	3	0x0001_8000 ~ 0x0001_FFFF	0x5E01_8000 ~ 0x5E01_FFFF	32	8	*	*	*
	4	0x0002_0000 ~ 0x0002_7FFF	0x5E02_0000 ~ 0x5E02_7FFF	32	8	*	*	*
	5	0x0002_8000 ~ 0x0002_FFFF	0x5E02_8000 ~ 0x5E02_FFFF	32	8	*	*	*
	6	0x0003_0000 ~ 0x0003_7FFF	0x5E03_0000 ~ 0x5E03_7FFF	32	8	*	*	*
	7	0x0003_8000 ~ 0x0003_FFFF	0x5E03_8000 ~ 0x5E03_FFFF	32	8	*	*	*
	8	0x0004_0000 ~ 0x0004_7FFF	0x5E04_0000 ~ 0x5E04_7FFF	32	8	*	*	
	9	0x0004_8000 ~ 0x0004_FFFF	0x5E04_8000 ~ 0x5E04_FFFF	32	8	*	*	
	10	0x0005_0000 ~ 0x0005_7FFF	0x5E05_0000 ~ 0x5E05_7FFF	32	8	*	*	
	11	0x0005_8000 ~ 0x0005_FFFF	0x5E05_8000 ~ 0x5E05_FFFF	32	8	*	*	
	12	0x0006_0000 ~ 0x0006_7FFF	0x5E06_0000 ~ 0x5E06_7FFF	32	8	*		
	13	0x0006_8000 ~ 0x0006_FFFF	0x5E06_8000 ~ 0x5E06_FFFF	32	8	*		
	14	0x0007_0000 ~ 0x0007_7FFF	0x5E07_0000 ~ 0x5E07_7FFF	32	8	*		
	15	0x0007_8000 ~ 0x0007_FFFF	0x5E07_8000 ~ 0x5E07_FFFF	32	8	*		

フラッシュメモリ構成の単位として、「エリア」、「ブロック」、「ページ」があります。

- ・ ページ
  - 消去機能、プロテクト機能で使⽤します。
  - 1 ページは 4096 バイト固定です。
- ・ ブロック
  - 消去機能、プロテクト機能で使⽤します。
  - 1 ブロックは 32K バイト固定です。
- ・ エリア
  - 消去機能で使⽤します。
  - 1 エリアは 512K バイトです。

書き込みは 16 バイト単位(4 バイト x4 回)で行います。16 バイトあたりの書き込み時間は 163 $\mu$ s (Typ.)です。

消去はページ単位、ブロック単位、エリア単位またはフラッシュメモリ全体で行います。消去時間は使⽤するコマンドによって異なります。自動ブロック消去コマンドを⽤した場合は 1 ブロックあたり 920 ms (Typ.)、それ以外のコマンドを⽤した場合は 115 ms (Typ.)です。

プロテクトの設定は、ページ 0~7 はページ単位で行い、残りのブロックはブロック単位で行います。プロテクト設定の消去は、一括消去で行います。プロテクト機能については「24.1.5 プロテクト/セキュリティ機能」を参照してください。

### 24.1.2

本製品内蔵のフラッシュメモリは、一部の機能を除き JEDEC 標準機能に準拠しています。このため、外部メモリとしてフラッシュメモリをご⽤になられている場合でも、本製品への移行が容易です。また、フラッシュメモリ内に書き込み、チップ消去など自動で行う回路を内蔵していますので、書き込み、消去動作を容易に実現できます。

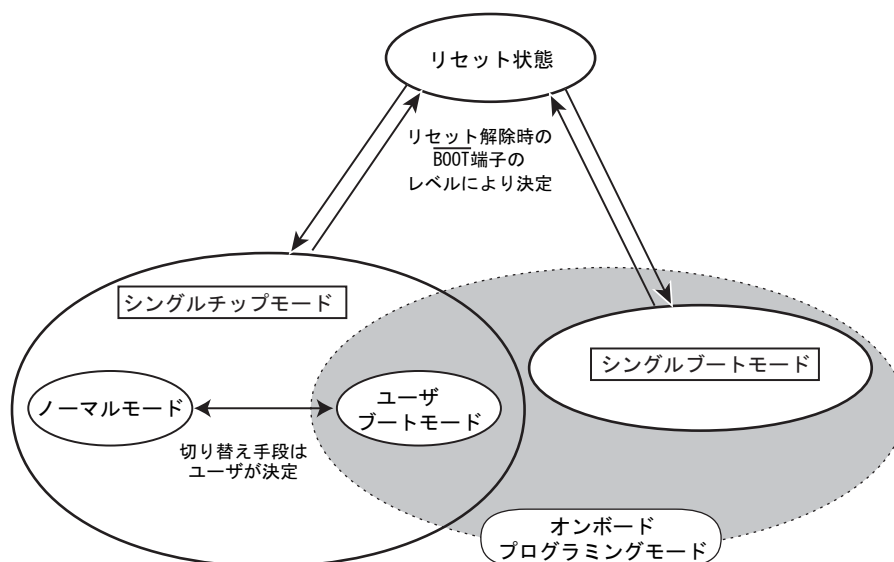
JEDEC	
	< >
/	< > / ( )
	< > /

### 24.1.3

フラッシュメモリの書き込み中と消去中は、電源オフやリセットを禁止します。

#### 24.1.3.1

本製品には、シングルチップモードとシングルブートモードがあり、シングルチップモードにはノーマルモード、ユーザブートモードがあります。モード遷移図を図 24-1 に示します。



## 24-1

(1)

リセット解除後、フラッシュメモリから起動するモードで、以下の3つのモードがあります。

- ・ ノーマルモード  
ユーザのアプリケーションプログラムを実行するモードです。
- ・ ユーザブートモード  
ユーザのセット上でフラッシュメモリ外に置くフラッシュメモリの書き替えプログラムにてフラッシュメモリの書き替えを実行するモードです。  
フラッシュメモリの書き替え方法は「24.4 ユーザブートモードによる書き替え方法」を参照してください。

各モードの切り替えはユーザが独自に設定できます。例えばポートAのPA0が"1"のときノーマルモード、"0"のときにユーザブートモードというように自由に設計することが可能です。ユーザはアプリケーションプログラムの一部に切り替えを判定するためのルーチンを準備してください。

(2)

リセット解除後、内蔵するBOOT ROM (Mask ROM) から起動するモードです。

内蔵メモリなど、フラッシュメモリ外に置くフラッシュメモリの書き替えプログラムにてフラッシュメモリの書き替えを実行することができます。フラッシュメモリの書き替え方法は「24.3 シングルブートモードによる書き替え方法」を参照してください。

BOOT ROMには、本デバイスのシリアルポートを経由してユーザのセット上で書き換えを行うアルゴリズムがプログラムされています。

シリアルポートにより外部ホストと接続し、規定されたプロトコルでデータの転送を行うことでフラッシュメモリの書き替えが実行できます。

(3)

ユーザのセット上でフラッシュメモリの書き替えが可能なモードは、ユーザブートモードとシングルブートモードです。これらをオンボードプログラミングモードと定義します。

24.1.3.2

シングルチップ、シングルブートの各動作モードは、リセットを解除するときの  $\overline{\text{BOOT}}$  端子の状態により決定されます。

リセットの動作については、「リセットの動作」の章を参照してください。

(1)

24-3

	RESET	BOOT
		0 → 1
	0 → 1	0

(2)

24-4

( $\overline{\text{RESET}}$ )

	POR	BOOT
	0 → 1	0

- ・ パワーオンリセット回路によるモード決定( $\overline{\text{RESET}}$  端子を使用しない場合)  
 パワーオンカウンタによる POR 引き伸ばし信号が"0" から"1" になるまで、 $\overline{\text{BOOT}}$  端子の設定を継続してください。
- ・  $\overline{\text{RESET}}$  端子によるモード決定  
 POR 引き伸ばし信号が"1" の時は  $\overline{\text{RESET}}$  端子によるリセットが可能となるので、この場合、表 24-3 と同じ設定を実施してください。  
 また、POR 引き伸ばし信号が"0" の時は表 24-4 と同じ設定を実施してください。この場合、 $\overline{\text{RESET}}$  端子よりも POR 引き伸ばし信号によるリセットが優先となります。

24.1.4

図 24-2 および図 24-3 にシングルチップモードとシングルブートモードのメモリマップの比較を示します。図のように、シングルブートモードでは、フラッシュメモリは 0x5E00\_0000 番地からマッピングされます。また、シングルブートモードでは 0x0000\_0000 番地から 0x0000\_0FFF 番地には BOOT ROM がマッピングされます。

フラッシュメモリと RAM のマッピングは以下のとおりです。

	Flash	RAM	Flash	RAM
TMPM475FDFG	512KB	34KB	0x0000_0000 ~ 0x0007_FFFF( ) 0x5E00_0000 ~ 0x5E07_FFFF( ( )) 0x5E00_0000 ~ 0x5E07_FFFF( )	0x2000_0000 ~ 0x2000_87FF
TMPM475FZFG	384KB	34KB	0x0000_0000 ~ 0x0005_FFFF( ) 0x5E00_0000 ~ 0x5E05_FFFF( ( )) 0x5E00_0000 ~ 0x5E05_FFFF( )	0x2000_0000 ~ 0x2000_87FF
TMPM475FYFG	256KB	18KB	0x0000_0000 ~ 0x0003_FFFF( ) 0x5E00_0000 ~ 0x5E03_FFFF( ( )) 0x5E00_0000 ~ 0x5E03_FFFF( )	0x2000_0000 ~ 0x2000_3FFF, 0x2000_8000 ~ 0x2000_87FF

) TMPM475FZFG TMPM475FYFG ID (0x5E07\_FFF0 ~ 0x5E07\_FFFF)



24-2

(TMPM475FDFG)

シングルチップモード

	0xFFFF_FFFF
内蔵Flash ROM (Mirror)	0x5E05_FFFF 0x5E00_0000
内蔵I/O	0x5DFF_0000
内蔵I/O	0x400F_FFFF 0x4000_0000
内蔵RAM (34KB)	0x2000_87FF 0x2000_0000
内蔵Flash ROM (512 KB)	0x0005_FFFF 0x0000_0000

シングルブートモード

	0xFFFF_FFFF
内蔵Flash ROM (Mirror)	0x5E05_FFFF 0x5E00_0000
内蔵I/O	0x5DFF_0000
内蔵I/O	0x400F_FFFF 0x4000_0000
内蔵RAM (34KB)	0x2000_87FF 0x2000_0000
内蔵BOOT ROM (4 KB)	0x0000_0FFF 0x0000_0000

24-3

(TMPM475FZFG)

シングルチップモード

	0xFFFF_FFFF
内蔵Flash ROM (Mirror)	0x5E03_FFFF 0x5E00_0000
内蔵I/O	0x5DFF_0000
内蔵I/O	0x400F_FFFF 0x4000_0000
内蔵RAM (2KB)	0x2000_87FF 0x2000_8000
内蔵RAM (16KB)	0x2000_3FFF 0x2000_0000
内蔵Flash ROM (512 KB)	0x0003_FFFF 0x0000_0000

シングルブートモード

	0xFFFF_FFFF
内蔵Flash ROM (Mirror)	0x5E03_FFFF 0x5E00_0000
内蔵I/O	0x5DFF_0000
内蔵I/O	0x400F_FFFF 0x4000_0000
内蔵RAM (2KB)	0x2000_87FF 0x2000_8000
内蔵RAM (16KB)	0x2000_3FFF 0x2000_0000
内蔵BOOT ROM (4 KB)	0x0000_0FFF 0x0000_0000

24-4

(TMPM475FYFG)

24.1.5 /

本製品は、フラッシュメモリに対して、プロテクトとセキュリティの2つの機能を持っています。

1. プロテクト機能  
フラッシュメモリへの書き込み、消去を禁止
2. セキュリティ機能  
フラッシュライタによるフラッシュメモリの読み出しの禁止  
デバッグ機能の使用制限

24.1.5.1

Block0 はページ単位とブロック単位で、Block1 から最終ブロックまではブロック単位でプロテクト機能を使用することができます。

プロテクト機能を有効にするためには、自動プロテクトビットプログラムコマンドを用いて対応するプロテクトビットを"1"にします。プロテクトビット消去コマンドによりすべてのプロテクトビットを"0"にし、すべてのプロテクトは解除されます。プロテクトビットは、各 FCPSR レジスタでそれぞれモニタすることができます。

プロテクトビットのプログラムと消去の方法については、「24.2.6 コマンド説明」の章を参照してください。

24.1.5.2

セキュリティ機能が有効な場合の動作を、表 24-5 に示します。

24-5

	CPU

セキュリティ機能が有効になる条件は以下のとおりです。

1. FCSECBIT<SECBIT>が"1"にセットされている。
2. すべてのプロテクトビット(各 FCPSR レジスタのすべてのビット)が"1"にセットされている。

FCSECBIT<SECBIT>はコールドリセットで"1"にセットされます。FCSECBIT<SECBIT>の書き換えは以下の手順で行います。

- ) 1., 2. 32bit
  1. FCSECBIT に対して特定のコード(0xa74a9d23)を書き込む。
  2. 1.の書き込みから 16 クロック以内にデータを書き込む。
- ) FCSECBIT<SECBIT>="0" "0"  
(

24-6 SECBIT

	<SECBIT>=1	<SECBIT>=0
<All>=1		TOOL /Read/Write
<any>=1	TOOL	TOOL /Read/Write
<All>=0	TOOL /Read/Write	TOOL /Read/Write

## 24.1.6

### 24.1.6.1

フラッシュメモリの書き替え操作の途中で電源が OFF になった場合、例えばプログラム消去後の電源が OFF になり、書き込みができなくなるケースが考えられます。このようなケースを回避するために、本機能を利用して書き込みプログラムを残すことができます。

### 24.1.6.2

スワップ領域は 0 番地で始まる領域と次の領域で、スワップサイズは FCSWPSR<SIZE>で決まります。このサイズを変更するには、自動メモリスワップコマンドにて FCSWPSR<SIZE>のビットを"1"にセットします。

メモリスワップを行うには、自動メモリスワップコマンドにて FCSWPSR[0]に"1"を設定します。スワップ状態を解除するには、自動メモリスワップコマンドにて FCSWPSR[1]に"1"を設定します。スワップ状態は FCSWPSR<SWP>にて確認することができます。

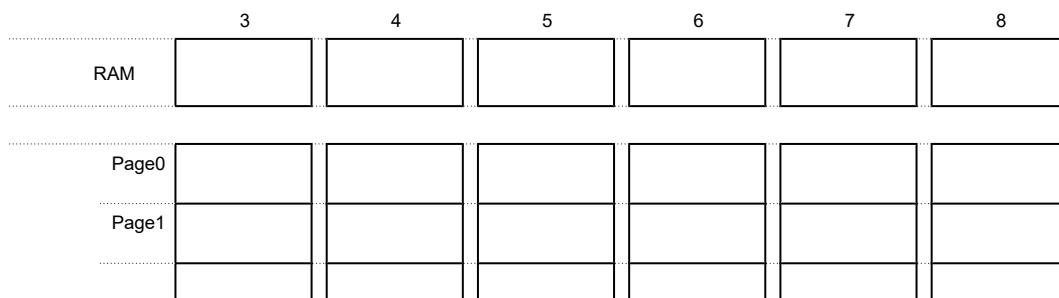
自動メモリスワップコマンドの詳細は「24.2.6 コマンド説明」の章を参照してください。

### 24.1.6.3

メモリスワップ操作の基本的な流れを以下に示します。メモリスワップ操作の具体例は「24.5 ユーザブートプログラムの書き替え方法」を参照してください。

1. セキュリティ機能が有効の場合は、セキュリティを解除してください。  
セキュリティの解除方法は「24.1.5.2 セキュリティ機能」を参照してください。  
セキュリティを解除しない場合、手順におけるコマンド実行にてフラッシュメモリが消去されます。
2. プロテクト機能が有効の場合は、プロテクトビットを消去してください。  
プロテクトビットの消去方法は「24.1.5.1 プロテクト機能」を参照してください。  
プロテクトビットを消去しない場合、手順におけるコマンド実行が行われません。
3. 0 番地で始まる領域の次の領域がブランク状態であることを確認します。(以後、0 番地で始まる領域を Page0、次の領域を Page1 として説明します。)ブランク状態でなければ消去してください。  
Page0 : 旧オリジナルデータ  
Page1 : ブランク
4. 0 番地で始まる領域のオリジナルデータを次の領域にも書き込みます。(両方の領域のデータを同じにします)  
Page0 : 旧オリジナルデータ  
Page1 : コピーデータ(旧オリジナルデータ)
5. メモリスワップを行います。  
Page0 : コピーデータ(旧オリジナルデータ)  
Page1 : 旧オリジナルデータ

6. 旧オリジナルデータを消去して、ブランク状態にします。  
     Page0 : コピーデータ(旧オリジナルデータ)  
     Page1 : ブランク
7. ブランク領域に新しいデータを書き込みます。  
     Page0 : コピーデータ(旧オリジナルデータ)  
     Page1 : 新オリジナルデータ
8. スワップ状態を解除します。  
     Page0 : 新オリジナルデータ  
     Page1 : コピーデータ(旧オリジナルデータ)
9. 自動プロテクトビット消去コマンドを実行します。
10. 必要により以下を行ってください。
  - ・ コピーデータ(旧オリジナルデータ)消去。
  - ・ スワップ領域以外のフラッシュメモリのデータ書き換え。
  - ・ プロテクト機能の有効化
  - ・ セキュリティ機能の有効化



:

:

:

## 24.1.7

### 24.1.7.1

制御レジスタとアドレスは以下の通りです。

ベースアドレスは、「メモリマップ」章の「周辺機能ベースアドレス一覧」を参照下さい。

:FC

		Address(Base+)
	FCSECBIT	0x0010
0	FCPSR0	0x0020
	FCSR	0x0100
	FCSWPSR	0x0104
	FCAREASEL	0x0140
	FCCR	0x0148
	FCSTSCLR	0x014C
WCLK	FCWCLKCR	0x0150
Program	FCPROGCR	0x0154
Erase	FCERASECR	0x0158

24.1.7.2 FCSECBIT( )

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

Bit	Bit Symbol	Type	
31-1		R	"0"
0	SECBIT	R/W	0: 1:

)

## 24.1.7.3 FCPSR0( 0)

	31	30	29	28	27	26	25	24
bit symbol	BLK15	BLK14	BLK13	BLK12	BLK11	BLK10	BLK9	BLK8
	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)
	23	22	21	20	19	18	17	16
bit symbol	BLK7	BLK6	BLK5	BLK4	BLK3	BLK2	BLK1	-
	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	0
	15	14	13	12	11	10	9	8
bit symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)	( 1)
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	-	-	RDY_BSY
	0	0	0	0	0	0	0	1

Bit	Bit Symbol	Type	
31-17	BLK15 ~ BLK1	R	Block1 ~ 15 0: 1:  "1"
16	-	R	"0"
15-8	PG7 ~ PG0	R	Page0 ~ 7 1: 0:  "1"
7-1	-	R	"0"
0	RDY_BSY	R	Ready/Busy ( 2) 0: 1:  "0"  "1"

1)

2)

24.1.7.4 FCSR( )

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

Bit	Bit Symbol	Type	
31-25		R	"0"
24	WEABORT	R	"1" 24.2.4
23-0		R	"0x000001"

24.1.7.5 FCSWPSR( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	SIZE		
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	FLG						SWP	
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-11		R	"0"
10-8	SIZE[2:0]	R	000: 4K ( 0 1) 001: 8K ( 0-1 2-3) 010: 16K ( 0-3 4-7) 011: 32K ( 0 1) :
7-2	FLG	R	( 24.5
1-0	SWP[1:0]	R	11: 10: 01: 00: ( )

)

24.1.7.6 FCAREASEL( )

	31	30	29	28	27	26	25	24
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8
bit symbol	-	-	-	-	-	-	-	-
	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
bit symbol	-	-	-	-	-	AREA0		
	0	0	0	0	0	0	0	0

Bit	Bit Symbol	Type	
31-3		R	"0"
2-0	AREA0	R/W	111: 0 : 0

- 1)
- 2) "111(0x7)"
- 3) "111(0x7)"

24.1.7.7 FCCR( )

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

Bit	Bit Symbol	Type	
31-3		R	"0"
2-0	WEABORT	R/W	111: 24.2.4

24.1.7.8 FCSTSCLR( )

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

Bit	Bit Symbol	Type	
31-3		R	"0"
2-0	WEABORT	R/W	FCSR<WEABORT> "0" 111: 24.2.4

24.1.7.9 FCWCLKCR(WCLK )

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

Bit	Bit Symbol	Type	
31-5		R	"0"
4-0	DIV	R/W	(WCLK : fsys/(DIV+1)) 8 ~ 12MHz 00000: 1 00001: 2 : 11110: 31 11111: 32

1) <DIV>

2) Flash WCLK 8 ~ 12MHz (fsys) <DIV>

24-6

24-6 (fsys) <DIV>

	fsys	10MHz	20MHz	25MHz	30MHz	40MHz	80MHz	100MHz	120MHz
<DIV>		(MHz)							
00000	1	10	-	-	-	-	-	-	-
00001	2	-	10	-	-	-	-	-	-
00010	3	-	-	8.3	10	-	-	-	-
00011	4	-	-	-	-	10	-	-	-
00111	8	-	-	-	-	-	10	-	-
01001	10	-	-	-	-	-	8	10	12
01011	12	-	-	-	-	-	-	8.3	10
01110	15	-	-	-	-	-	-	-	8

-:

24.1.7.10 FCPRGCR(Program )

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

Bit	Bit Symbol	Type	
31-2	-	R	"0"
1-0	CNT	R/W	(CNT/WCLK) 20 ~ 40μs 00: 250 01: 300 : 350

) WCLKCR<DIV> 24-7 WCLK <CNT> 20 ~ 40μs

24-7 WCLK <CNT>

	WCLK	8.33MHz	10MHz	12MHz
<CNT>		(μs)		
00	250	30.0	25.0	20.8
01	300	36.0	30.0	25.0
	350	-	35.0	29.2

-:

24.1.7.11 FCERASECR(Erase )

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

Bit	Bit Symbol	Type	
31- 4	-	R	"0"
3-0	CNT	R/W	(CNT/WCLK) 100 ~ 130ms
			0000: 850000
			0001: 900000
			0010: 950000
			0011: 1000000
			0100: 1050000
			0101: 1100000
			0110: 1150000
			0111: 1200000
			1000: 1250000
			1001: 1300000
			1010: 1350000
			: 1400000

) WCLKCR<DIV>

24-8 WCLK 100 ~ 130ms  
<CNT>

24-8 WCLK <CNT>

	WCLK	8.33MHz	10MHz	12MHz
<CNT>		(ms)		
0000	850000	102.0	-	-
0001	900000	108.0	-	-
0010	950000	114.0	-	-
0011	1000000	120.0	-	-
0100	1050000	126.1	105.0	-
0101	1100000	-	110.0	-
0110	1150000	-	115.0	-
0111	1200000	-	120.0	-
1000	1250000	-	125.0	104.2
1001	1300000	-	-	108.3
1010	1350000	-	-	112.5
	1400000	-	-	116.7

:-

## 24.2

オンボードプログラミングモードでは、制御プログラム中でコマンドを実行することで、フラッシュメモリの書き込み/消去を行います。この書き込み/消去の制御プログラムはユーザがあらかじめ用意しておきます。

またエリア 0 のフラッシュメモリ上でプログラムを実行中に、命令実行を行っていない他の"エリア"(例えばエリア 1)のフラッシュメモリを書き込み/消去できます(逆も可能です)。

### 24.2.1

一部の機能を除き、フラッシュメモリの書き込みおよび消去などは JEDEC 標準コマンドに準拠していますが、動作コマンドのアドレス指定が標準コマンドとは異なります。

書き込み、消去を行う場合、32 ビット(1 ワード)のストア命令を用いてフラッシュメモリへコマンドを入力します。コマンド入力後、書き込みおよび消去は内部で自動的に行われます。

24-9

	4 (16 )
/	
	/ /

### 24.2.2

フラッシュメモリには、大きく分けて以下の 2 種類の動作モードがあります。

- ・ メモリデータを読み出すモード(リードモード)
- ・ メモリデータを自動的に消去/書き替えるモード(自動動作)

電源投入後、リセット解除後、および自動動作の終了時にフラッシュメモリはリードモードになります。フラッシュメモリに書かれた命令の実行、およびデータの読み出しはリードモードで行います。

リードモード中にコマンドを入力すると自動動作へ移り、コマンドの処理が終了するとリードモードに戻ります(ID-Read コマンドを除く)。自動動作中は、フラッシュメモリデータの読み出しとフラッシュメモリ上の命令の実行ができません。コマンドの実行方法については「24.2.3 コマンド実行方法」を参照してください。

### 24.2.3

コマンド実行は、ストア命令を用いてフラッシュメモリに対してコマンドシーケンスを書き込むことで行います。フラッシュメモリは、入力されたアドレスとデータの組み合わせによって各自動動作コマンドを実行します。コマンド実行の詳細は、「24.2.6 コマンド説明」を参照してください。

フラッシュメモリに対するストア命令の実行を"バスライトサイクル"と呼びます。各コマンドはいくつかのバスライトサイクルで構成されています。フラッシュメモリは、バスライトサイクルのアドレスとデータが規定の順番で実行された時はコマンドの自動動作を実施します。規定の順番で実行されなかった場合、フラッシュメモリはコマンドの実行を中止してリードモードになります。

コマンドシーケンスの途中でキャンセルしたい場合や、間違っただコマンドシーケンスを入力した場合は、Read/リセットコマンドを実行します。フラッシュメモリはコマンド実行を中止してリードモードになります。Read/リセットコマンドをソフトウェアリセットと呼びます。

コマンドシーケンスの書き込みが終了すると自動動作を開始し、FCPSR0<RDY\_BSY>="0"になります。自動動作が終了するとFCPSR0<RDY\_BSY>="1"となり、リードモードに復帰します。

自動動作中は、新たなコマンドシーケンスを受け付けません。動作を中止する場合は「24.2.4 自動動作の中止」を参照してください。

コマンドを実行する際には以下の事項に留意してください。

1. 自動動作中は以下の操作を行わないでください。
  - ・ すべての例外発生
2. コマンドシーケンサがコマンドを認識するために、コマンド開始前の状態がリードモードである必要があります。各コマンドシーケンスの第1バスライトサイクル前にFCPSR0<RDY\_BSY>="1"であることを確認してください。続いてRead/リセットコマンドを実行することを推奨します。
3. コマンドシーケンスは、内蔵RAM上で実行してください。
4. 各コマンドを実行する前にFCAREASELレジスタのエリア選択ビットを設定(<AREAn>に"111"(0x07)をライト)してください。
5. 各バスライトサイクルは連続して、1ワード(32ビット)のデータ転送命令で行います。
6. 各コマンドシーケンスの実行中に、実行対象となるフラッシュメモリへのアクセスを行うとバスフォールトが発生します。
7. コマンド発行時、誤ったアドレスやデータをライトした場合は、必ずソフトウェアリセットを発行して、リードモードに戻してください。
8. 各コマンド実行の終了確認手順は以下の通りです。
  - 1)最終バスライトサイクルを実行します。
  - 2)FCPSR0<RDY\_BSY>="0"(Busy)となるまでポーリングします。
  - 3)FCPSR0<RDY\_BSY>="1"(Ready)となるまでポーリングします。
9. フラッシュメモリからデータをリードする場合は、FCAREASELレジスタのエリア選択ビットをクリア(<AREAn>に"000"(0x0))してください。

#### 24.2.4

自動動作を中止する手順を以下に示します。

1. FCPSR0<RDY\_BSY>をリードします。
2. 手順1のリード結果が"1"(Ready)の場合は、手順9で終了してください。"0"(Busy)の場合は、手順3へ進みます。
3. FCCR<WEABORT>に"0x7"をライトします。

4. FCCR<WEABORT>に"0x0"をライトします。
  5. FCPSR0<RDY\_BSY>="1"(Ready)となるまでポーリングします。
  6. FCSR<WEABORT>をリードします。
  7. Read/リセットコマンドを実行します。
  8. 手順6のリード結果が"0"の場合(注)は、手順9で終了してください。"1"の場合は、以下の操作を行い、このフラグをクリアします。
    - (1) FCSTSCLR<WEABORT>に"0x7"をライトします。
    - (2) FCSTSCLR<WEABORT>に"0x0"をライトします。
    - (3) FCSR<WEABORT>="0"となるまでポーリングします。
  9. 終了
- )        3        FCPSR0<RDY\_BSY>="1"

## 24.2.5

Flash の書込み/消去動作等の完了を検知する割り込み機能があります。

### 24.2.5.1

自動動作の完了検知割り込みを使用する手順は以下の通りです。

割り込み処理の詳細については、例外の章の”割り込み”を参照して下さい。

1. Flash に対し書込み/消去コマンドを発行後、FCPSR0<RDY\_BSY>で、自動動作中(BUSY 状態)を確認します。自動動作中を確認したら、CPU 割り込みの許可の設定をします。
2. Flash の自動動作終了後、INTFLRDY 割り込みが発生
3. INTFLRDY 割り込み処理ルーチンの中で、CPU 割り込みの禁止をしてください。

## 24.2.6

各コマンドの内容について説明します。具体的なコマンドシーケンスは「24.2.7 コマンドシーケンス」を参照してください。

### 24.2.6.1

#### (1)

自動プログラムコマンドシーケンスにより、4ワード(16バイト)単位で書き込みができます。16バイトを跨ってデータを書き込むことはできません。

フラッシュメモリへの書き込みは、“1”データセルを“0”データにすることです。“0”データセルを“1”データにすることはできません。“0”データセルを“1”データにするには消去動作を行う必要があります。

自動プログラムは消去後のページに対して1回のみ可能で、"1"データセルであっても"0"データセルであっても2回以上の実行はできません。一度書き込み動作を行ったページに対して再度書き込みを行う場合は、自動ページ消去、自動ブロック消去または自動チップ消去コマンドを行った後に自動プログラムを実行しなおす必要があります。

自動プログラム中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、該当となるデータの書き込みは正常に行われていない可能性があるため、消去動作後に改めて自動プログラムを実行する必要があります。

- 1) 2
- 2)

(2)

第1～第3 バスライトサイクルが自動プログラムのコマンドシーケンスです。

第4 バスライトサイクルでページの先頭アドレスとデータを書き込みます。第5 バスライトサイクル以降は4ワードの内の残りのデータを書き込みます。データは1ワード(32ビット)単位で書き込んでください。

16 バイトの一部に書き込みを行う場合、書き込みが不要なアドレスのデータを"0xFFFFFFFF"として16 バイト分の書き込みを行ってください。

本デバイス内部で自動的なベリファイ動作は行いませんので、正常に書き込みができたか、実行後に読み出しをして確認してください。

#### 24.2.6.2

(1)

自動チップ消去は、全アドレスのメモリセルに対して消去動作を行います。プロテクトされているページまたはブロックがある場合は自動チップ消去を実行せず、コマンドシーケンスの入力後にリードモードに戻ります。

自動消去中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、データの消去は正常に行われていない可能性があるため、改めて自動消去を実行する必要があります。

(2)

第1～第6 バスライトサイクルが自動チップ消去のコマンドシーケンスです。コマンドシーケンス入力後、自動チップ消去動作を行います。

本デバイス内部で自動的なベリファイ動作は行いませんので、正常に消去ができたか、実行後に読み出しをして確認してください。

#### 24.2.6.3

(1)

自動エリア消去コマンドは、指定された"エリア"に対して消去動作を行います。プロテクトされているページまたはブロックがある場合は自動エリア消去を実行せず、コマンドシーケンスの入力後にリードモードに戻ります。

自動消去中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、データの消去は正常に行われていない可能性があるため、改めて自動消去を実行する必要があります。

(2)

第1～第5バスライトサイクルが自動エリア消去のコマンドシーケンスです。第6バスライトサイクルで消去する"エリア"を指定します。コマンドシーケンス入力後、自動エリア消去動作を行います。

本デバイス内部で自動的なベリファイ動作は行いませんので、正常に消去ができたか、実行後に読み出しをして確認してください。

#### 24.2.6.4

(1)

自動ブロック消去コマンドは、指定されたブロックに対する消去動作を行います。プロテクトされているページや指定されたブロックがプロテクトされている場合は自動ブロック消去を実行せず、コマンドシーケンスの入力後にリードモードに戻ります。

自動消去中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、データの消去は正常に行われていない可能性があるため、改めて自動消去を実行する必要があります。

(2)

第1～第5バスライトサイクルが自動ブロック消去のコマンドシーケンスです。第6バスライトサイクルで消去するブロックを指定します。コマンドシーケンス入力後、自動ブロック消去動作を行います。

本デバイス内部で自動的なベリファイ動作は行いませんので、正常に消去ができたか、実行後に読み出しをして確認してください。

#### 24.2.6.5

(1)

自動ページ消去コマンドは、指定されたページに対する消去動作を行います。指定されたページがプロテクトされている場合は消去を実行せず、コマンドシーケンスの入力後にリードモードに戻ります。

自動消去中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、データの消去は正常に行われていない可能性があるため、改めて自動消去を実行する必要があります。

(2)

第1～第5バスライトサイクルが自動ページ消去のコマンドシーケンスです。第6バスライトサイクルで消去するページを指定します。コマンドシーケンス入力後、自動ページ消去動作を行います。

本デバイス内部で自動的なベリファイ動作は行いませんので、正常に消去ができたか、実行後に読み出しをして確認してください。

#### 24.2.6.6

##### (1)

自動プロテクトビットプログラムは、プロテクトビットにビット単位で"1"を書き込みます。プロテクトビットを"0"にするためには自動プロテクトビット消去コマンドを使用します。

プロテクトの機能については「24.1.5 プロテクト/セキュリティ機能」を参照してください。

自動プロテクトビットプログラム中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、プロテクトの設定は正常に行われていない可能性があるため、改めて自動プロテクトビットプログラムを実行する必要があります。

##### (2)

第1～第3バスライトサイクルが自動プロテクトビットプログラムのコマンドシーケンスになります。第4バスライトサイクルで書き込むプロテクトビットを指定します。コマンドシーケンス入力後、自動プロテクトビットプログラム動作を行います。正しく書き込みができたかどうか、各 FCPSR レジスタの各ビットを確認してください。

#### 24.2.6.7

##### (1)

自動プロテクトビット消去コマンドは、実行する際のセキュリティの状態によって動作内容が異なります。

- ・ セキュリティ状態でない場合

すべてのプロテクトビットを"0"にクリアします。

- ・ セキュリティ状態の場合

フラッシュメモリの全アドレスのデータを消去した後、すべてのプロテクトビットを消去します。

プロテクトの機能については、「24.1.5 プロテクト/セキュリティ機能」を参照してください。

自動プロテクトビット消去中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、プロテクトの消去は正常に行われていない可能性があるため、改めて自動プロテクトビット消去を実行する必要があります。

##### (2)

第1～第3バスライトサイクルが自動プロテクトビット消去のコマンドシーケンスになります。第4バスライトサイクルで 0x5E000000 を指定します。コマンドシーケンス入力後、自動動作を行います。

セキュリティ状態でない場合、すべてのプロテクトビットが消去されます。正常に消去が行われたか、各 FCPSR レジスタの各ビットを確認してください。

セキュリティ状態の場合、フラッシュメモリの全アドレスのデータ消去およびすべてのプロテクトビットが消去されます。データの消去およびプロテクトビットの消去が正常に行われたか読み出して確認し、必要に応じて再度自動プロテクトビット消去、自動チップ消去あるいは自動ブロック消去を実行してください。

#### 24.2.6.8 ID-Read

(1)

ID-Read コマンドは、フラッシュメモリのタイプ等の情報を読み出すことができます。読み出せる内容は、メーカーコード、デバイスコード、マクロコードの3種類です。

(2)

第1～第3バスライトサイクルが ID-Read のコマンドシーケンスになります。第4バスサイクルで読み出すコードを指定します。第4バスライトサイクル以降、任意のフラッシュ領域からのリード動作でコードが得られます。

ID-Read コマンドは連続実行が可能です。第4バスライトサイクルと ID の値の読み出しは繰り返し実行できます。

ID-Read コマンドは自動的にリードモードに戻りません。リードモードへの復帰は Read/リセットコマンドで行います。

#### 24.2.6.9 Read/ ( )

(1)

フラッシュメモリをリードモードに戻すコマンドです。

ID-Read コマンドを実行した場合、Flash メモリは自動的にリードモードに復帰せず、その状態で停止します。このような状態からリードモードに復帰させるために、Read/リセットコマンドを使用します。また、途中まで入力したコマンドをキャンセルする場合にも使用します。

(2)

Read/リセットコマンドでは第1バスライトサイクルがコマンドシーケンスになります。コマンドシーケンス実行後、フラッシュメモリはリードモードになります。

#### 24.2.6.10

(1)

自動メモリスワップは、FCSWPSR[10:0]の各ビットにビット単位で"1"を書き込むコマンドです。各ビットを"0"にすることはできず、自動プロテクトビット消去コマンドを使用してすべてのビットを"0"クリアします。

自動メモリスワップ動作中は、新たにコマンドシーケンスを受け付けません。この動作を中止するには「24.2.4 自動動作の中止」を参考に中止してください。この場合、この動作は正常に行われていない可能性があるため、改めて自動メモリスワップを実行する必要があります。

(2)

第1～第4バスライトサイクルが自動メモリスワップのコマンドシーケンスです。コマンドシーケンス入力後、FCSWPSR レジスタの指定ビットに"1"が書き込まれます。

## 24.2.7

### 24.2.7.1

表 24-10 に各コマンドのバスライトサイクルのアドレスとデータを示します。

ID-Read コマンドの第5バスサイクル以外はすべて「バスライトサイクル」です。バスライトサイクルは32ビット(1ワード)のデータ転送命令で実施します(表では、データの下位8ビットのデータのみ示しています)。

アドレスの詳細は、表 24-11 を参照してください。表 24-11 で「コマンド」と記載された、Addr [15:9]に下記値を使用します。

- |    |         |         |
|----|---------|---------|
| 1) | Flash   | (Mirror |
| 2) | [20:19] |         |
|    | 0       | "00"    |

24-10 CPU

	1	2	3	4	5	6	7
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read/	0xFFFFFFFF						
	0xF0						
ID-Read	0XX54XX	0XXAAXX	0XX54XX	IA	0XX		
	0xAA	0x55	0x90	0x00	ID		
	0XX54XX	0XXAAXX	0XX54XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	0XX54XX	
	0xAA	0x55	0x80	0xAA	0x55	0x10	
	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	AA	
	0xAA	0x55	0x80	0xAA	0x55	0x20	
	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	BA	
	0xAA	0x55	0x80	0xAA	0x55	0x30	
	0XX54XX	0XXAAXX	0XX54XX	0XX54XX	0XXAAXX	PGA	
	0xAA	0x55	0x80	0xAA	0x55	0x40	
	0XX54XX	0XXAAXX	0XX54XX	PBA			
	0xAA	0x55	0x9A	0x9A			
	0XX54XX	0XXAAXX	0XX54XX	0x0000XX			
	0xAA	0x55	0x6A	0x6A			
	0XX54XX	0XXAAXX	0XX54XX	MSA			
	0xAA	0x55	0x9A	0x9A			

補足説明

- ・ IA: ID アドレス
- ・ ID: ID データ
- ・ PA: プログラムアドレス
- ・ PD: プログラムデータ(32 ビットデータ)  
第4バスサイクル以降16バイト分をアドレス順にデータ入力
- ・ AA: エリアアドレス(表 24-2 参照)
- ・ BA: ブロックアドレス(表 24-2 参照)
- ・ PGA: ページアドレス
- ・ PBA: プロテクトビットアドレス(表 24-12 参照)
- ・ MSA: メモリスワップアドレス(表 24-14 参照)

24.2.7.2

表 24-11 は「表 24-10 内部 CPU によるフラッシュメモリアクセス」と併せてご使用願います。

第 1 バスサイクルから「通常のバスライトサイクルアドレス設定」に従い、アドレス設定を行ってください。

24-11

[            ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:16]	Addr [15:9]	Addr [8:0]
	0x5E	"000"		"0"		"0"

[Read/            ID-READ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:16]	Addr [15:14]	Addr [13:0]
Read /	Read/            1					
	0x5E	"000"	"00"	"0"		
ID-READ	IA: ID            (ID-READ            4            )					
	0x5E	"000"	"00"	"0"	ID (            24-13)	"0"

[            ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:0]
	AA:            (            6            )			
	0x5E	"000"		"0"

[            ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:13]	Addr [12:0]
	BA:            (            6            )				
	0x5E	"000"		(            24-2)	"0"

[            ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:12]	Addr [11:0]
	PGA:            (            6            )				
	0x5E	"000"			"0"

[ ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:3]	Addr [2:0]
	PA: ( 4 )				
	0x5E	"000"			"0"

[ ]

	Addr [31:24]	Addr [23:21]	Addr [20:19]	Addr [18:12]	Addr [11:8]	Addr [7:4]	Addr [3:0]
	4						
	0x5E	"000"	"0"			"0"	
	PBA: ( 4 )						
	0x5E	"000"	"00"	"0"	( 24-12)		"0"
	MSA: ( 4 )						
	0x5E	"000"	"00"	"0"	( 24-14)		"0"

) [20:19]  
0 "00"

24.2.7.3 (AA) (BA)

表 24-2 にエリアアドレスとブロックアドレスを示します。自動エリア消去コマンドと自動ブロック消去コマンドの第 6 バスライトサイクルで、消去する"エリア"またはブロックに含まれる任意のアドレスを指定します。シングルチップモードでは、ミラー領域のアドレスを指定してください。

24.2.7.4 (PBA)

プロテクトビットは、1 ビット単位の操作になります。

自動プロテクトビットプログラムのプロテクトビット選択表を表 24-12 に示します。

24-12

Block	Page			PBA[11:4]							[31:0]	
				Addr [11:10]	Addr [9]	Addr [8]	Addr [7]	Addr [6]	Addr [5]	Addr [4]		
0	0	FCPSR0	<PG0>	0	0	0	0	0	0	0	0	0x5E00_0000
	1		<PG1>	0	0	0	0	0	0	0	1	0x5E00_0010
	2		<PG2>	0	0	0	0	0	0	1	0	0x5E00_0020
	3		<PG3>	0	0	0	0	0	0	1	1	0x5E00_0030
	4		<PG4>	0	0	0	0	0	1	0	0	0x5E00_0040
	5		<PG5>	0	0	0	0	0	1	0	1	0x5E00_0050
	6		<PG6>	0	0	0	0	0	1	1	0	0x5E00_0060
	7		<PG7>	0	0	0	0	0	1	1	1	0x5E00_0070
1	8 ~ 15		<BLK1>	0	0	0	1	0	0	0	0	0x5E00_0080
2	16 ~ 23		<BLK2>	0	0	0	1	0	0	0	1	0x5E00_0090
3	24 ~ 31		<BLK3>	0	0	0	1	0	1	0	0	0x5E00_00A0
4	32 ~ 39		<BLK4>	0	0	0	1	0	1	1	1	0x5E00_00B0
5	40 ~ 47		<BLK5>	0	0	0	1	1	1	0	0	0x5E00_00C0
6	48 ~ 55		<BLK6>	0	0	0	1	1	0	0	1	0x5E00_00D0
7	56 ~ 63		<BLK7>	0	0	0	1	1	1	1	0	0x5E00_00E0
8	64 ~ 71	<BLK8>	0	0	0	1	1	1	1	1	0x5E00_00F0	
9	72 ~ 79	<BLK9>	0	0	1	0	0	0	0	0	0x5E00_0100	
10	80 ~ 87	<BLK10>	0	0	1	0	0	0	0	1	0x5E00_0110	
11	88 ~ 95	<BLK11>	0	0	1	0	0	0	1	0	0x5E00_0120	
12	96 ~ 103	<BLK12>	0	0	1	0	0	0	1	1	0x5E00_0130	
13	104 ~ 111	<BLK13>	0	0	1	0	0	1	0	0	0x5E00_0140	
14	112 ~ 119	<BLK14>	0	0	1	0	0	1	0	1	0x5E00_0150	
15	120 ~ 127	<BLK15>	0	0	1	0	0	1	1	0	0x5E00_0160	

24.2.7.5 ID-Read (IA, ID)

ID-Read コマンドでのコード指定方法と読み出される内容を表 24-13 に示します。

24-13 ID-Read

Code	ID[7:0]	IA[15:14]	[31:0]
	0x0098	00	0x5E00_0000
	0x005A	01	0x5E00_4000
	Reserved	10	
	0x012F	11	0x5E00_C000

24.2.7.6 (MSA)

自動メモリスワップコマンドの第 4 バスライトサイクルで指定する FCSWPSR[10:0]への設定値を表 24-14 に示します。

24-14 FCSWPSR[10:0]

FCSWPSR[10:0]	MSA[11:4]							[31:0]
	[11]	[10:9]	[8]	[7]	[6]	[5]	[4]	
FCSWPSR[0]	1	"0"	0	1	0	0	0	0x5E00_0880
FCSWPSR[1]	1	"0"	0	1	0	0	1	0x5E00_0890
FCSWPSR[2]	1	"0"	0	1	0	1	0	0x5E00_08A0
FCSWPSR[3]	1	"0"	0	1	0	1	1	0x5E00_08B0
FCSWPSR[4]	1	"0"	0	1	1	0	0	0x5E00_08C0
FCSWPSR[5]	1	"0"	0	1	1	0	1	0x5E00_08D0
FCSWPSR[6]	1	"0"	0	1	1	1	0	0x5E00_08E0
FCSWPSR[7]	1	"0"	0	1	1	1	1	0x5E00_08F0
FCSWPSR[8]	1	"0"	1	0	0	0	0	0x5E00_0900
FCSWPSR[9]	1	"0"	1	0	0	0	1	0x5E00_0910
FCSWPSR[10]	1	"0"	1	0	0	1	0	0x5E00_0920

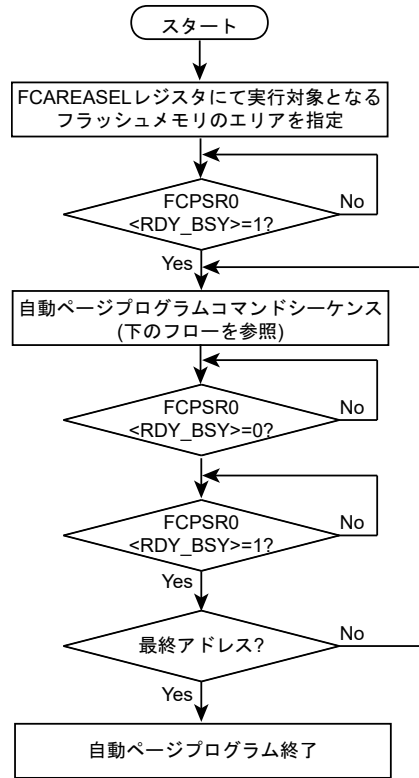
24.2.7.7

		1	2	3	4	5	6	7
Read/		0x5E00_0000						
		0x0000_00F0						
ID-Read		0x5E00_5400	0x5E00_AA00	0x5E00_5400	IA	0x5E00_0000		
		0x0000_00AA	0x0000_0055	0x0000_0090	0x0000_0000	ID		
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	0x5E00_5400	
		0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0010	
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	PBA			
		0x0000_00AA	0x0000_0055	0x0000_009A	0x0000_009A			
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_0000			
		0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_006A			
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	MSA			
		0x0000_00AA	0x0000_0055	0x0000_006A	0x0000_009A			

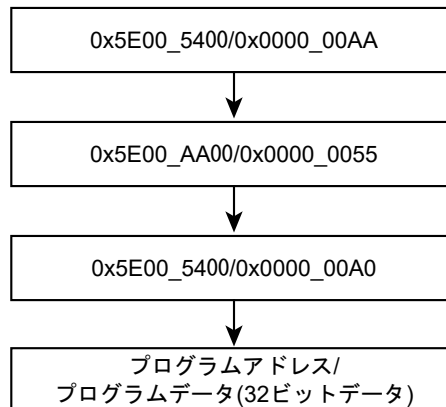
		1	2	3	4	5	6	7
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	PA	16		
		0x0000_00AA	0x0000_0055	0x0000_00A0	PD			
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	0x5E00_0000	
		0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0020	
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	BA	
		0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0030	
		0x5E00_5400	0x5E00_AA00	0x5E00_5400	0x5E00_5400	0x5E00_AA00	PGA	
		0x0000_00AA	0x0000_0055	0x0000_0080	0x0000_00AA	0x0000_0055	0x0000_0040	

## 24.2.8

### 24.2.8.1

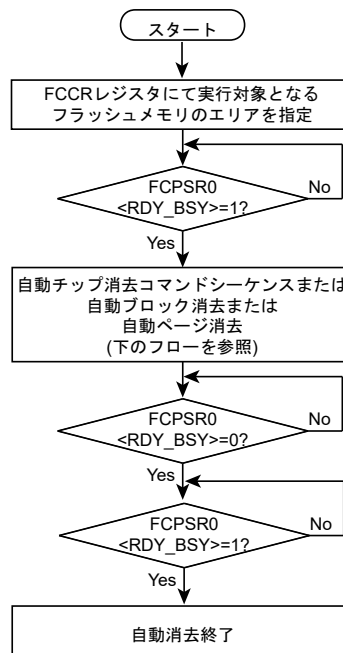


自動ページプログラムコマンドシーケンス(アドレス/コマンド)

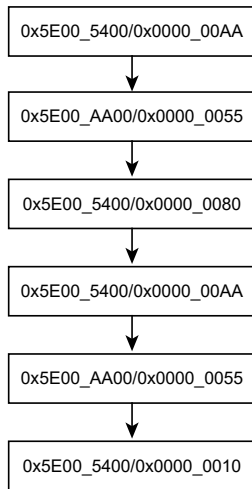


24-5

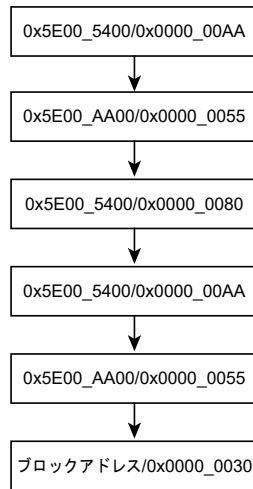
24.2.8.2



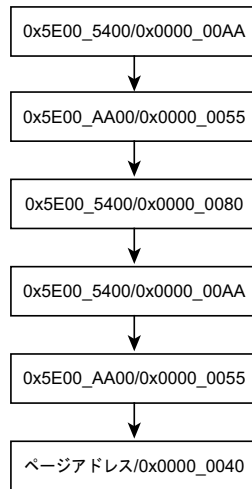
自動チップ消去コマンドシーケンス  
(アドレス/コマンド)



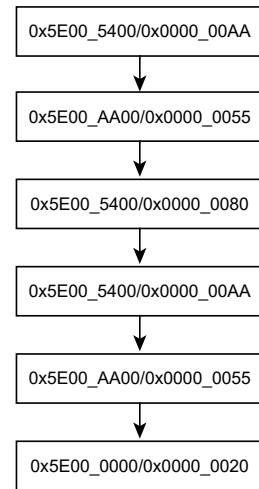
自動ブロック消去コマンドシーケンス  
(アドレス/コマンド)



自動ページ消去コマンドシーケンス  
(アドレス/コマンド)



自動エリア消去コマンドシーケンス  
(アドレス/コマンド)





24-15

	BOOT(PF0)
	RESET
	SC0TXD (PE0)
	RXD0 (PE1)

### 24.3.3

シングルブートモードでは、内蔵 RAM, 内蔵フラッシュメモリに対して表 24-16 のような制約がありますのでご注意ください。

24-16

RAM	0x2000_0000 ~ 0x2000_03FF      BOOT 0x2000_0400      RAM
	ID 0x5E07_FFF0 ~ 0x5E07_FFFF

) (0xFF)

### 24.3.4

内蔵ブートプログラムには、以下の動作コマンドが準備されています。

24-17

0x10	RAM
0x40	

#### 24.3.4.1 RAM

RAM 転送は、コントローラから送られてくるデータを内蔵 RAM へ格納します。転送が正常に終了するとユーザプログラムの実行を開始します。ユーザプログラム領域として、内蔵ブートプログラムで使用する領域(0x2000\_0000~0x2000\_03FF)を除く、0x2000\_0400 以降を使用可能です。実行開始アドレスは、RAM 格納開始アドレスになります。

この RAM 転送機能により、ユーザ独自のオンボードプログラミング制御を行うことができます。ユーザプログラムでオンボードプログラミングを実行するためには、24.2.7 で説明するフラッシュメモリコマンドシーケンスを使う必要があります。

#### 24.3.4.2

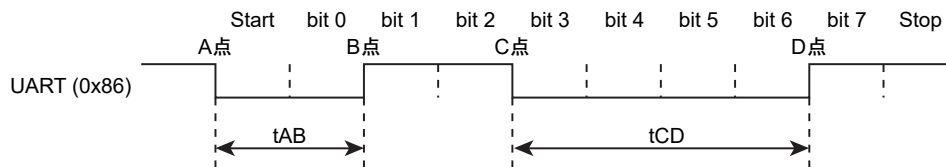
このコマンドはすべてのブロックのフラッシュメモリを消去します。ライト/消去プロテクトおよび、セキュリティ状態にかかわらず、メモリセルのすべてのブロックを消去し、すべてのブロックのライト/消去プロテクトを消去します。

## 24.3.5

内蔵ブートプログラム実行において、共通に行われる動作について説明します。

### 24.3.5.1

コントローラは、所望のボーレートで1バイト目を0x86にして送信してください。図24-7に波形を示します。

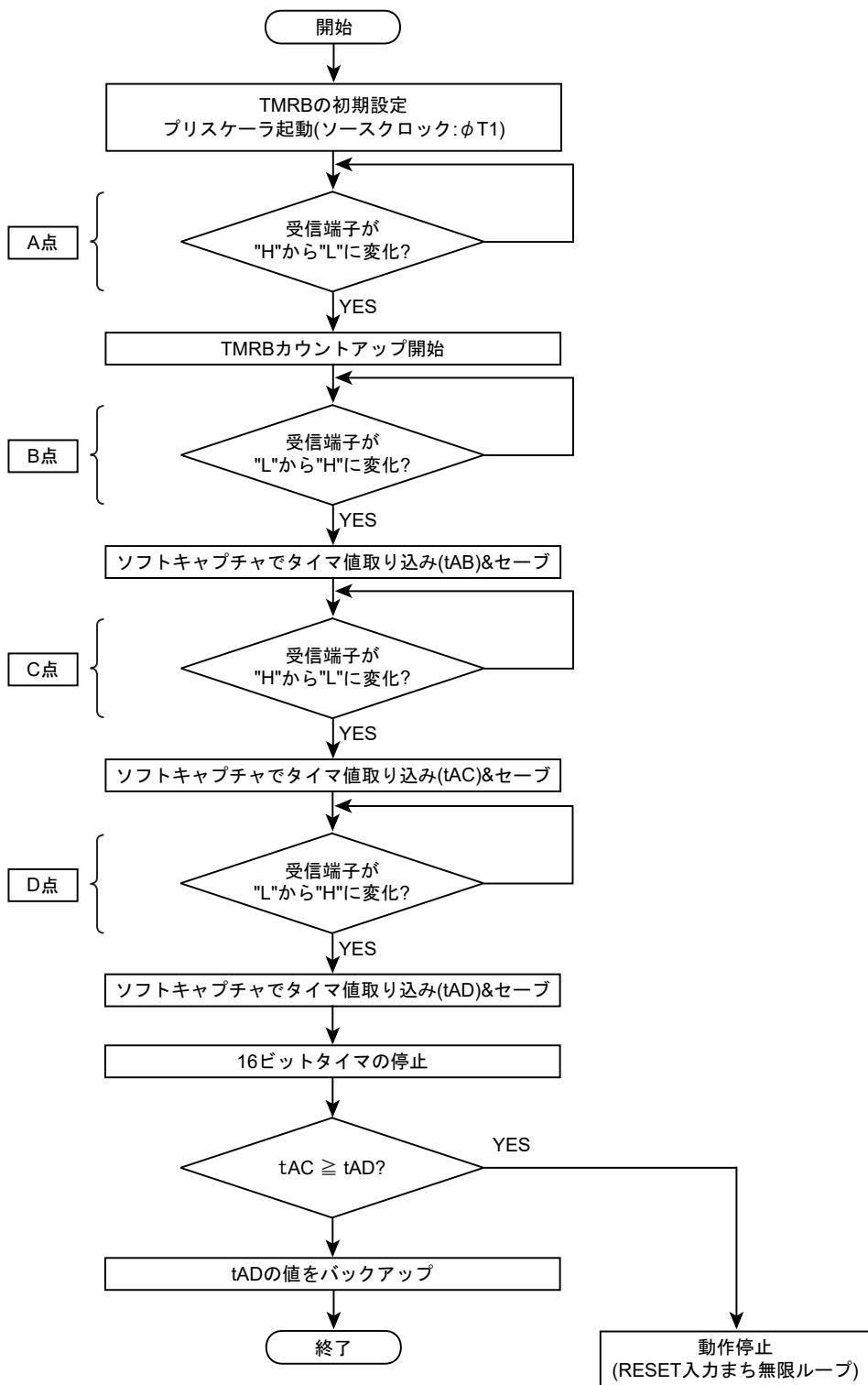


24-7

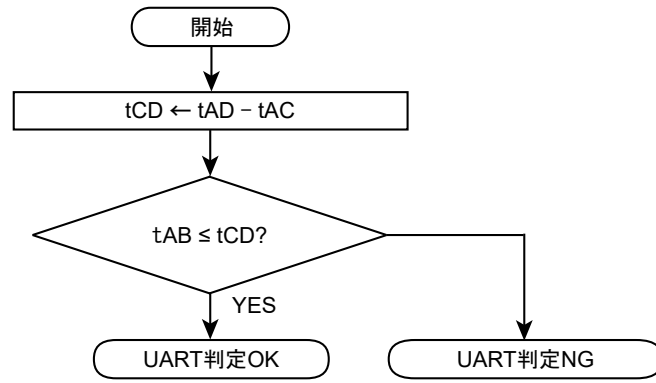
内蔵ブートプログラムは図24-8に示すフローチャートで、リセット解除後の1バイト目のシリアル動作モード判定データ(0x86)を、16ビットタイマ(TMRB)を用いて図24-7のtAB, tACと、tADの時間から求めています。図24-8のフローチャートに示すように、CPUが受信端子のレベルをモニタしてレベルの変化があると、そのときのタイマ値を取り込みます。このため、tAB, tACと、tADのタイマ値には誤差が生じます。また、ボーレートが速いときには、CPUは受信端子のレベルの変化を判断できない場合がありますので注意してください。

図24-9のフローチャートに示すように、シリアル動作モードの判定は、受信端子が"L"レベルのときの時間幅の大小関係で判定しています。tAB ≤ tCDの場合UARTと判定し、ボーレートの自動設定が可能かどうかをtADの時間から判定します。tAB > tCDの場合、UARTと判定しません。なお、先に述べたように、tAB, tAC, tADのタイマ値には誤差が生じているため、ボーレートが速く、動作周波数が低い場合、各タイマ値が小さくなり、意図しない判断を行うことがありますので注意してください(書き換えルーチン内でUARTの再設定を行ってください)。

例えば、コントローラはUARTで通信したいのに、UARTと判定されないことがあります。このようなことを考慮して、コントローラは1バイト目のデータを送信後、タイムアウト時間内にデータ0x86を正常受信できなければ通信不可能と判断してください。



24-8



24-9

24.3.5.2 ACK

内蔵ブートプログラムは処理状況を各種コードによってコントローラに送信します。表 24-18 から表 24-21 に各受信データに対する ACK 応答データを示します。

表 24-19 から表 24-21 に示す ACK 応答データの上位 4 ビットは、動作コマンドデータの上位 4 ビットになります。また 3 ビット目は受信エラーを表し、0 ビット目は動作コマンドエラー、CHECK SUM エラー、パスワードエラーの状態を表します。1 ビット目と 2 ビット目は常に 0 になります。

24-18 ACK

0x86	UART ( )
------	----------

) UART

24-19 ACK

0x?8 ( )	
0x?1 ( )	
0x10	RAM
0x40	

) 4 4

24-20 CHECK SUM ACK

0xN8 ( )	
0xN1 ( )	CHECK SUM
0xN0 ( )	CHECK SUM

) 4 4

24-21

ACK

0x54	
0x4F	
0x4C	
0x47	

## 24.3.5.3

内蔵ブートプログラムでは、以下の領域をパスワード要否判定およびパスワードのデータ領域として使用します。

	0x5E07_FFF0 (1byte)
	0x5E07_FFF4 ~ 0x5E07_FFFF (12byte)

RAM 転送コマンドでは、パスワード要否判定にかかわらずパスワード判定を行います。

フラッシュメモリチップ消去およびプロテクトビット消去コマンドでは、パスワード要否判定データが「要」の場合のみパスワード判定を行います。

	0xFF
	0xFF

パスワードが消去データ(0xFF)の場合、容易にパスワードの照合が可能になり、セキュリティの確保が難しくなります。シングルブートモードを使用しない場合も固有の値を置くことを推奨します。

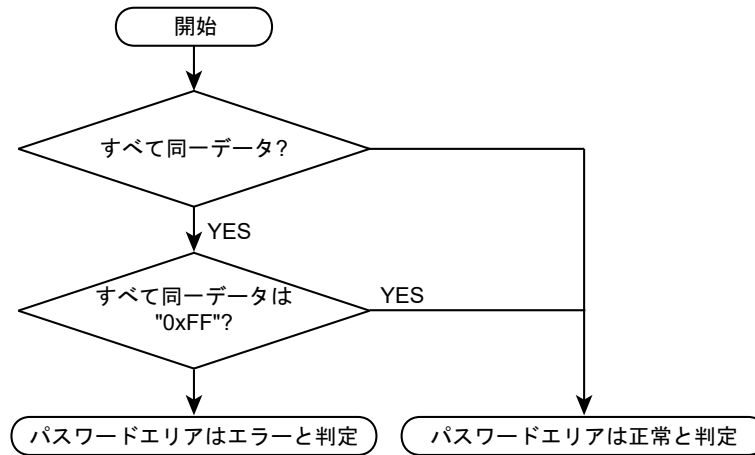
## (1) RAM

「RAM 転送コマンドの通信ルール」における No.5 のパスワード判定について説明します。

図 24-10 に示すようにパスワードエリアのデータが、0xFF 以外の同一データになっていた場合、パスワードエリアエラーと判定します。パスワードエリアエラーと判定された場合、パスワードデータの照合結果に関わらず、ACK 対応は 0x11 を送信します。

次に、受信データ(パスワードデータ)の照合を行います。12 バイト分すべてがフラッシュメモリ内のパスワードと一致しないと、パスワードエラーになります。パスワードエラーと判定された場合、ACK 応答は、パスワードエラーとなります。

セキュリティ機能が有効な状態でもパスワードの参照は行います。



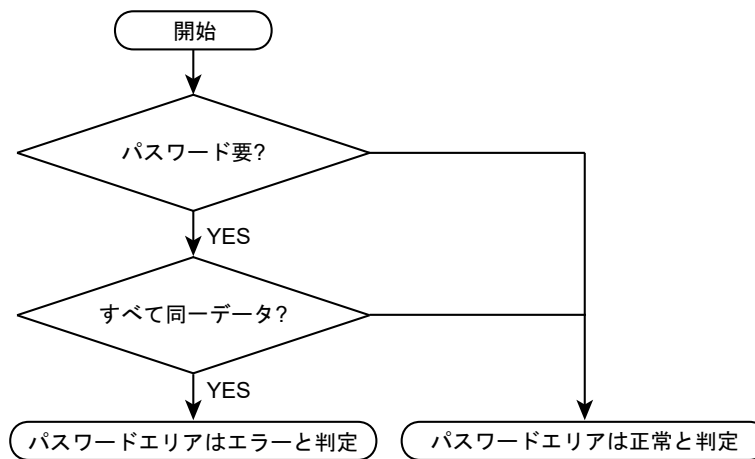
24-10

(2)

「フラッシュメモリチップ消去およびプロテクトビット消去の通信ルール」における No.5 のパスワード判定について説明します。

図 24-11 に示すように、消去パスワード要否の選択エリアがパスワード有効の場合は、パスワードエリアのデータが同一データになっていた場合、パスワードエリアエラーと判定します。パスワードエリアエラーと判定された場合、パスワードデータの照合結果に関わらず、CHECK SUM 値に対する ACK 対応は 0x41 を送信します。

次に、受信データ (パスワードデータ) の照合を行います。12 バイト分すべてがフラッシュメモリ内のパスワードと一致しないと、パスワードエラーになります。パスワードエラーと判定された場合、CHECK SUM 値に対する ACK 応答は、パスワードエラーとなります。セキュリティ機能が有効な状態でもパスワードの参照は行います。



24-11

24.3.5.4 CHECK SUM

CHECK SUM の計算方法は、送信データを符号なし 8 ビット加算(オーバーフローを無視)して得られた下位 8 ビット値の 2 の補数値を求めています。コントローラは CHECK SUM 値を送信するときは、本計算方法を使用してください。

例)CHECK SUM 計算例

2 バイトのデータ 0xE5、0xF6 の CHECK SUM 値を求める場合、まず符号なし 8 ビット加算を行います。

$$0xE5 + 0xF6 = 0x1DB$$

この値の下位 8 ビットに対しての 2 の補数をとると以下のようになり、この値が CHECK SUM 値になります。従って、コントローラには 0x25 を送信します。

$$0 - 0xDB = 0x25$$

24.3.6

シリアル動作モード判定の通信ルールを示します。表中の転送方向の表記が示す意味は以下のとおりです。

転送方向「C→T」：コントローラから TMPM475 へ

転送方向「C←T」：TMPM475 からコントローラへ

No			
1	C→T		24.3.5.1
		0x86	0x86 UART OK
2	C←T	ACK	ACK 1 UART
		: 0x86	0x86 1 (0x86) (5)
3			24.3.8 24.3.7 RAM

### 24.3.7 RAM

RAM 転送コマンドの通信ルールを示します。表中の転送方向の表記が示す意味は以下のとおりです。

転送方向「C→T」：コントローラから TMPM475 へ

転送方向「C←T」：TMPM475 からコントローラへ

#### RAM

No			
1	C→T	(0x10)	RAM (0x10)
2	C←T	ACK : 0x10 : 0x11 : 0x18	ACK 0x18 24-17 ACK 0x11 ACK 0x10
3	C→T	(12 )	24.3.5.3
4	C→T	(No.3) CHECK SUM	CHECK SUM (No.3) CHECK SUM 24.3.5.4 CHECK SUM
5	C←T	CHECK SUM ACK : 0x10 : 0x11 : 0x18	ACK 0x18 CHECK SUM 24.3.5.3 ACK 0x11 ACK 0x10
6	C→T	RAM 31 ~ 24	RAM 31 RAM 4 7 ~ 0 1 ~ 24 4 7 ~ 0 1 0x2000_0400 RAM
7	C→T	RAM 23 ~ 16	
8	C→T	RAM 15 ~ 8	
9	C→T	RAM 7 ~ 0	
10	C→T	RAM 15 ~ 8	RAM 15 ~ 8 2 7 ~ 0 1 RAM 0x2000_0400 RAM
11	C→T	RAM 7 ~ 0	
12	C→T	(No.6 ~ 11) CHECK SUM	(No.6 ~ 11) CHECK SUM

No			
13	C←T	CHECK SUM :0x10 :0x11 :0x18	ACK ACK 0x18 CHECK SUM ACK 0x11 ACK 0x10
14	C→T	RAM	RAM RAM
15	C→T	(No.14) CHECK SUM	(No.14) CHECK SUM
16	C←T	CHECK SUM :0x10 :0x11 :0x18	ACK ACK 0x18 CHECK SUM ACK 0x11 ACK 0x10 RAM (No.6 ~ 9)

### 24.3.8

フラッシュメモリチップ消去およびプロテクトビット消去コマンドの通信ルールを示します。表中の転送方向の表記が示す意味は以下のとおりです。

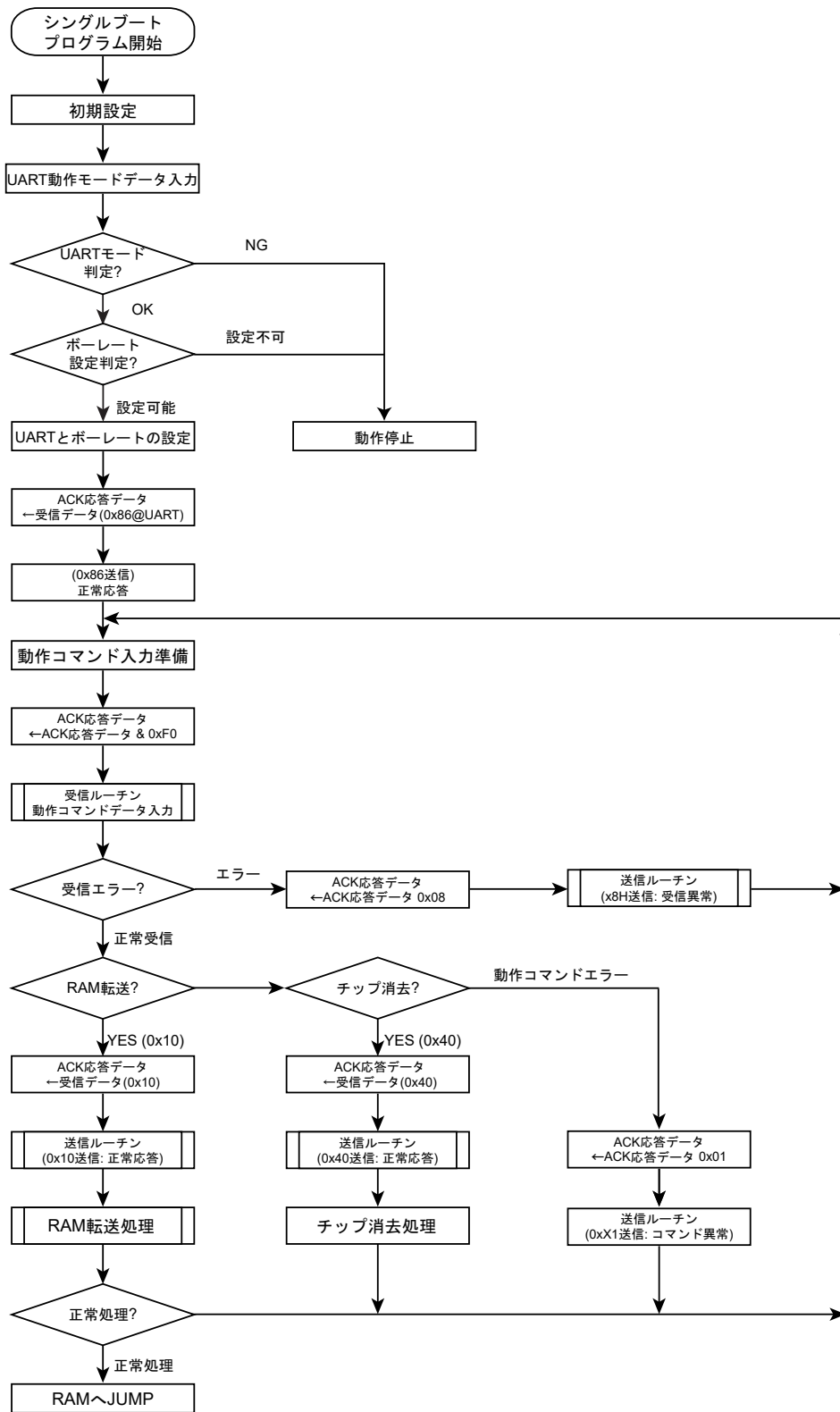
転送方向「C→T」：コントローラ→TMPM475

転送方向「C←T」：コントローラ←TMPM475

No			
1	C→T	(0x40)	(0x40)
2	C←T	ACK : 0x40 : 0x41 : 0x48	ACK ACK 0x48 24-17 ACK 0x41 ACK 0x40
3	C→T	(12 )	( :0xFF)  24.3.5.3
4	C→T	(No.3) CHECK SUM	CHECK SUM (No.3) CHECK SUM 24.3.5.4 CHECK SUM
5	C←T	CHECK SUM ACK : 0x40 : 0x41 : 0x48	ACK ACK 0x48 CHECK SUM ACK 0x41 ACK 0x40 ACK 0x41 ACK 0x40
6	C→T	(0x54)	(0x54)
7	C←T	ACK : 0x54 : 0x51 : 0x58	ACK ACK 0x58 (0x54) ACK 0x51 ACK 0x54
8	C←T	ACK : 0x4F : 0x4C	(0x4F) (0x4C) (0x47)

24.3.9

内蔵ブートプログラム全体フローチャートを示します。

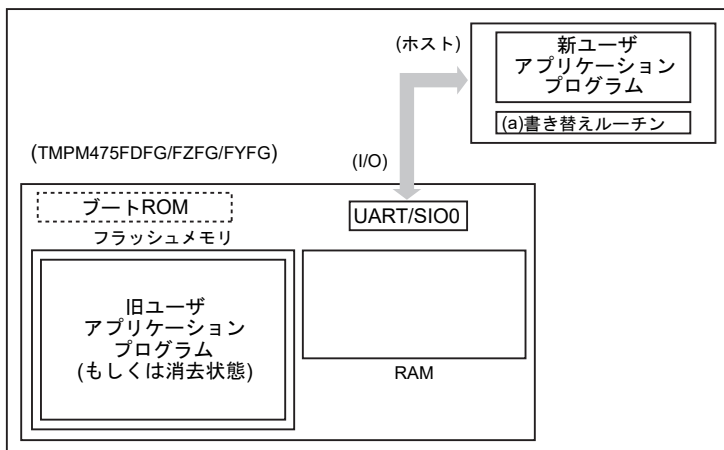


## 24.3.10 BOOT ROM

内蔵ブートプログラムを利用した書き替え手順を示します。

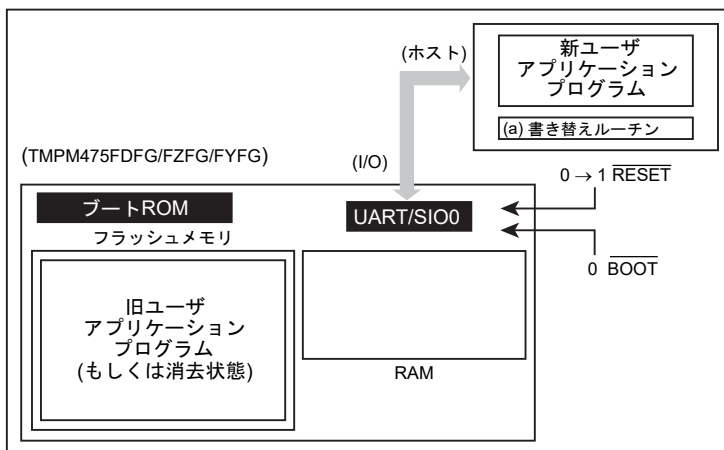
### 24.3.10.1 Step-1

フラッシュメモリの状態は旧バージョンのユーザプログラムが書かれた状態でも、消去されている状態でも構いません。書き替えルーチン、書き替えデータなどの転送は SIO (SIO0) を経由して行いますので、ボード上で本デバイスの SIO (SIO0) と外部ホストとをつなげます。書き替えを行うための(a)書き替えルーチンはホスト上に用意します。



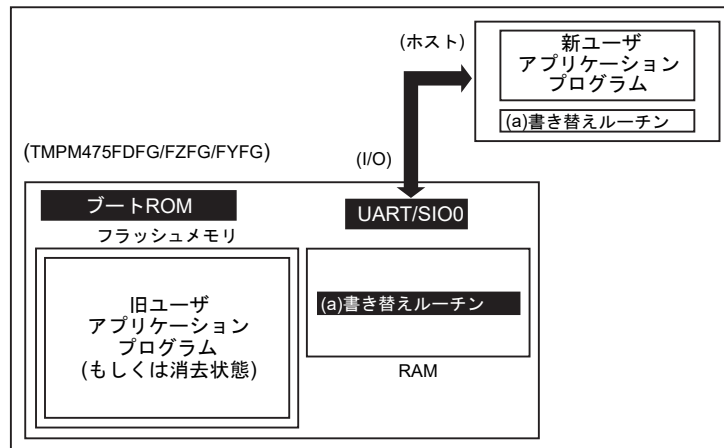
### 24.3.10.2 Step-2

シングルブートモードの端子条件設定でリセットを解除し、BOOT ROM で起動します。シングルブートモードの手順に従い、SIO0 を経由して転送元(ホスト)より(a)書き替えルーチンの転送を行います。最初にユーザアプリケーションプログラム上に記録されているパスワードとの照合を行います。(フラッシュメモリが消去されている状態でも、消去データ(0xFF)をパスワードとして照合を行います。)



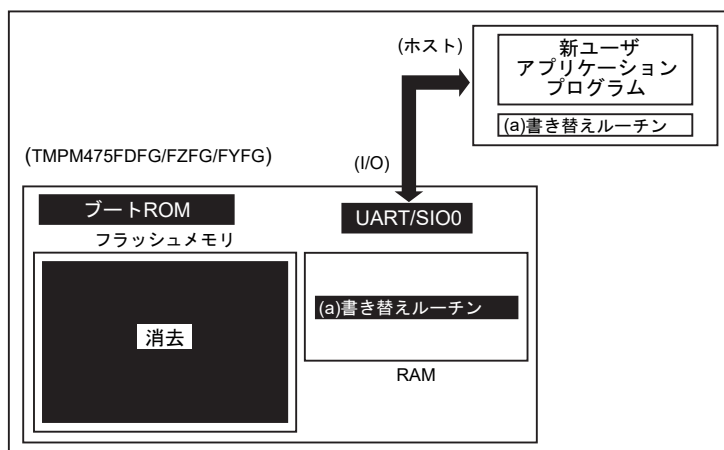
## 24.3.10.3 Step-3

パスワードの照合が終了すると、転送元(ホスト)から(a)書き替えルーチンを転送します。BOOT ROMはそのルーチンを内部RAMにロードします。ただし、RAM上のアドレス0x2000\_0400からRAMの最終番地の範囲に格納してください。



## 24.3.10.4 Step-4

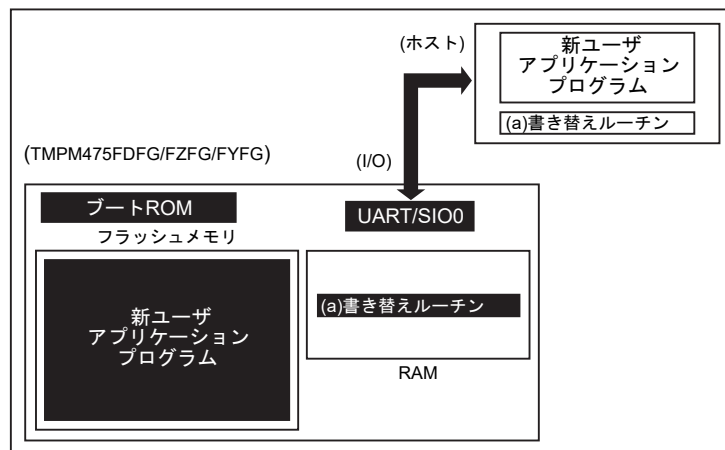
RAM上の(a)書き替えルーチンへジャンプし、旧ユーザアプリケーションプログラム領域の消去を行います。(任意の消去単位)



## 24.3.10.5 Step-5

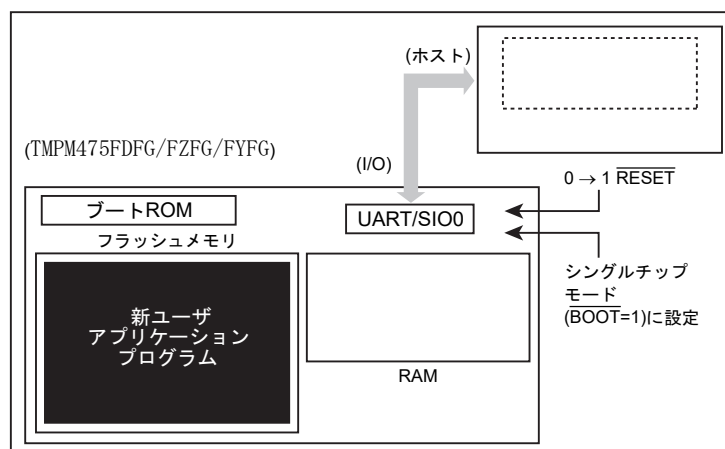
さらに、RAM上の(a)書き換えルーチンを実行して、転送元(ホスト)より新ユーザアプリケーションプログラムのデータをロードし、フラッシュメモリの消去した領域に書き込みを行います。書き込みが完了したら、ユーザプログラム領域のライト/消去プロテクトをオンにします。

下の例の場合、書き換えルーチンを転送したときと同じホストおよびSIO0経由で書き換えデータも転送されていますが、RAM上で動作を開始した以降では、ユーザ独自にデータバスおよび転送元を設定することもできます。方法に応じて、ボードのハードおよび書き換えルーチンを組み立ててください。



## 24.3.10.6 Step-6

書き込みが完了したら、一度ボードの電源を落とし、ホストと接続していたケーブルをはずします。この後、再度電源を入れ直し、シングルチップモード（ノーマルモード）起動し、新しいユーザアプリケーションプログラムを実行します。



## 24.4

ユーザのセット上で内蔵RAMに置くフラッシュメモリの書き替えプログラムにてフラッシュメモリの書き替えを実行する方法です。ユーザアプリケーション上で用意されているフラッシュメモリ書き替え用のプログラムで用いるデータ転送バスが、シリアル I/O と異なる場合に使用します。動作はシングルチップモード上で行います。このため、シングルチップモードにおいて通常のユーザアプリケーションプログラムが動作しているノーマルモードから、フラッシュを書き替えるためのユーザブートモードに移行する必要があります。従って、条件判定を行うプログラムをユーザアプリケーションの中で、リセット処理プログラムの中に組み込んでください。

このモード切り替えの条件設定は、ユーザのシステムセット条件に合わせて独自に構築してください。また、ユーザブートモード移行後に使用するユーザ独自のフラッシュメモリ書き替えルーチンも同様にユーザアプリケーションの中にあらかじめ組み込んでおき、ユーザブートモード移行後にこれらのルーチンを使用して書き替えを行ってください。また、シングルチップモード（通常動作モード）中に誤ってフラッシュの内容を書き替えないよう、書き替え処理が完了した後、必要なブロックにライト/消去プロテクトをかけておくことを推奨します。ユーザブートモード中は、すべての例外発生を禁止してください。

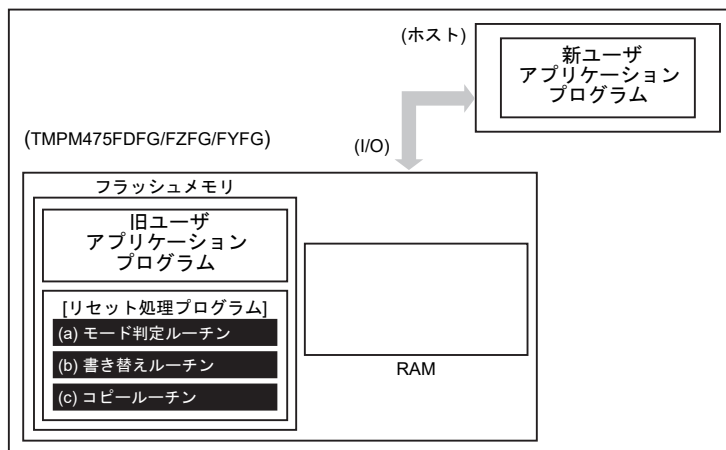
書き替えルーチンをフラッシュメモリに置く場合と、外部から転送する場合の2ケースを例に、以下(1-A), (1-B)にその手順を説明します。フラッシュメモリへの書き込み/消去方法の詳細は、「24.2 フラッシュメモリ詳細」を参照してください。

## 24.4.1 (1-A)

## 24.4.1.1 Step-1

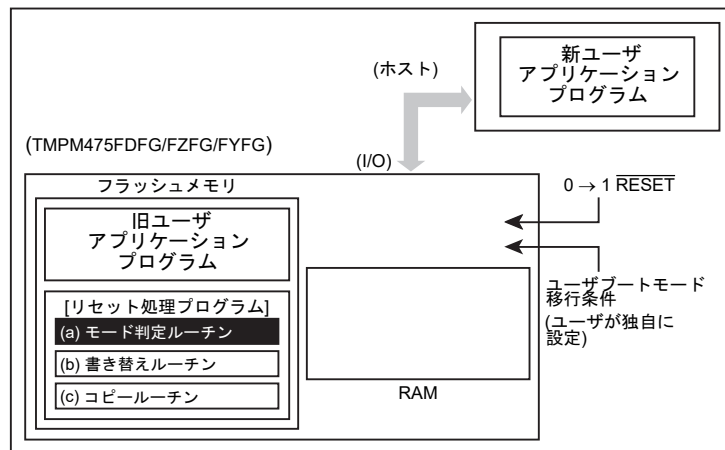
ユーザは、あらかじめどのような条件（例えば端子状態）に設定されたらユーザブートモードに移行するか、どの I/O バスを使用してデータ転送を行うかを決め、それに合った回路の設計、プログラムの作成を行います。ユーザは本デバイスをボードに組み込む前に、あらかじめフラッシュメモリ上の任意のブロックにライターなどを使用して以下に示す3つのプログラムを書き込んでおきます。

- (a) :  
 (b) :  
 (c) : (b) RAM



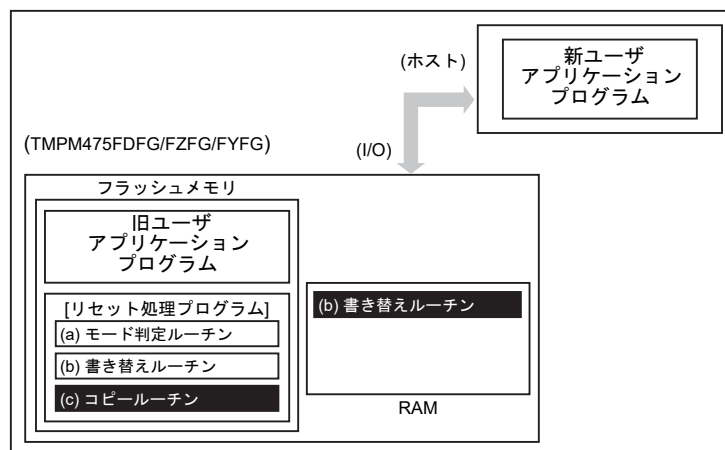
## 24.4.1.2 Step-2

以下リセット処理プログラム内にこれらのルーチンを組み込んだ場合について説明します。まず、リセット処理プログラムにおいてユーザブートモードへの移行を判定します。このとき、移行条件が整っていれば、プログラムは書き替えのためのユーザブートモードに移ります。(ユーザブートモードに移行した場合は、これ以降例外を発生させないでください)



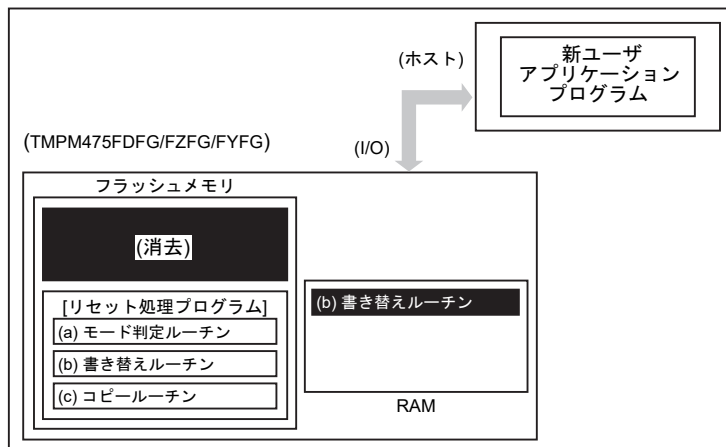
## 24.4.1.3 Step-3

ユーザブートモードに移ると、(c)コピールーチンを使用して、(b)書き替えルーチンを内蔵RAMにコピーします。



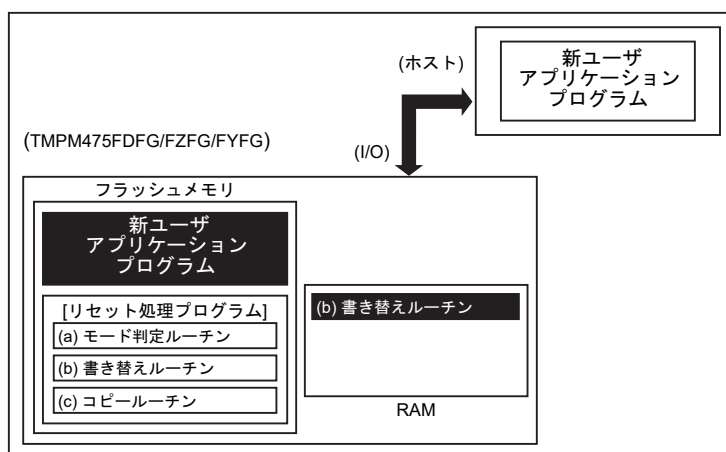
## 24.4.1.4 Step-4

RAM 上の書き替えルーチンへジャンプし、旧ユーザプログラム領域のライト/消去プロテクトを解除して、消去(任意の消去単位)を行います。



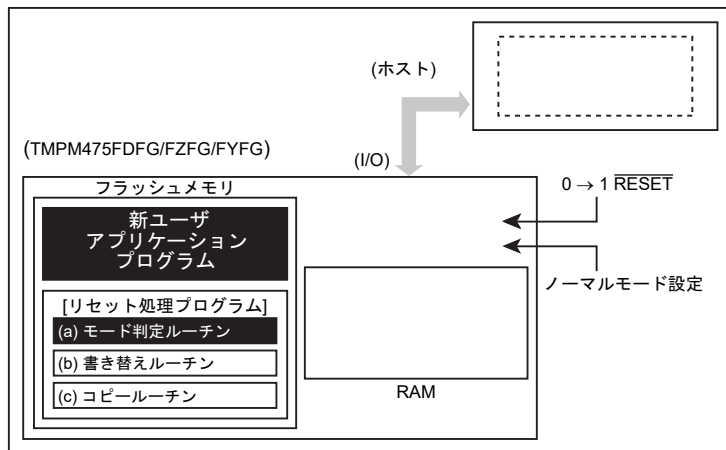
## 24.4.1.5 Step-5

さらに、RAM 上の書き替えルーチンを実行して、転送元 (ホスト) より新ユーザアプリケーションプログラムのデータをロードし、フラッシュメモリの消去した領域に書き込みを行います。書き込みが完了したら、ユーザプログラム領域のライト/消去プロテクトをオンにします。



## 24.4.1.6 Step-6

RESET 入力端子を"0"にしてリセットを行い、設定条件をノーマルモードの設定にします。リセット解除後、新ユーザアプリケーションプログラムで動作を開始します。



24.4.2 (1-B)

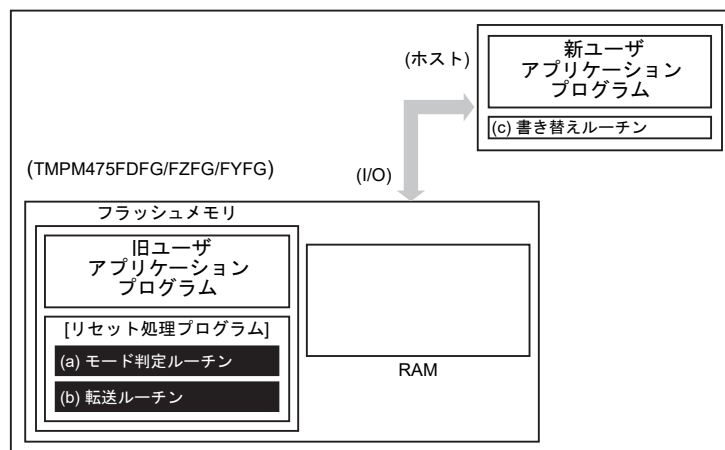
24.4.2.1 Step-1

ユーザは、あらかじめどのような条件(例えば端子状態)に設定されたらユーザブートモードに移行するか、どの I/O バスを使用してデータ転送を行うかを決め、それに合った回路の設計、プログラムの作成を行います。ユーザは本デバイスをボードに組み込む前に、あらかじめフラッシュメモリ上の任意のブロックにライタなどを使用して以下に示す2つのプログラムを書き込んでおきます。

- (a) :
- (b) :

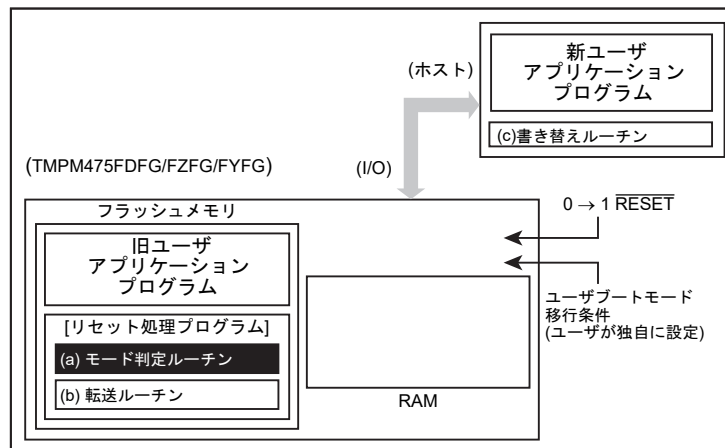
また、下記に示すプログラムはホスト上に用意します。

- (c) :



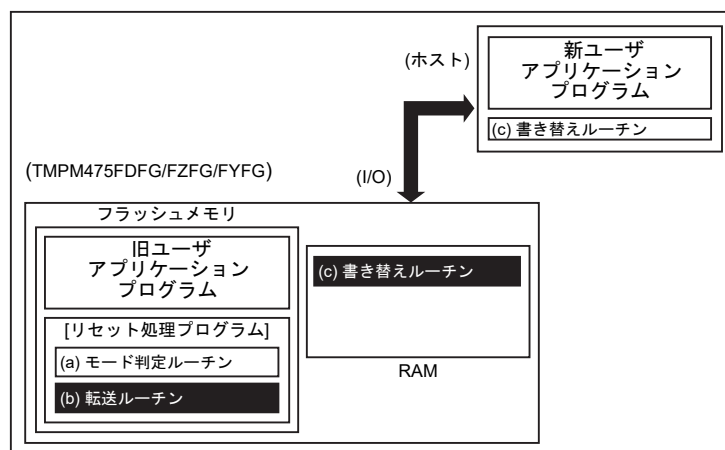
### 24.4.2.2 Step-2

以下、リセット処理プログラム内にこれらのルーチンを組み込んだ場合について説明します。まず、リセット解除後のリセット処理プログラムにおいてユーザブートモードへの移行を判定します。このとき、移行条件が整っていれば、プログラムは書き替えのためのユーザブートモードに移ります。(ユーザブートモードに移行した場合は、これ以降例外を発生させないでください)



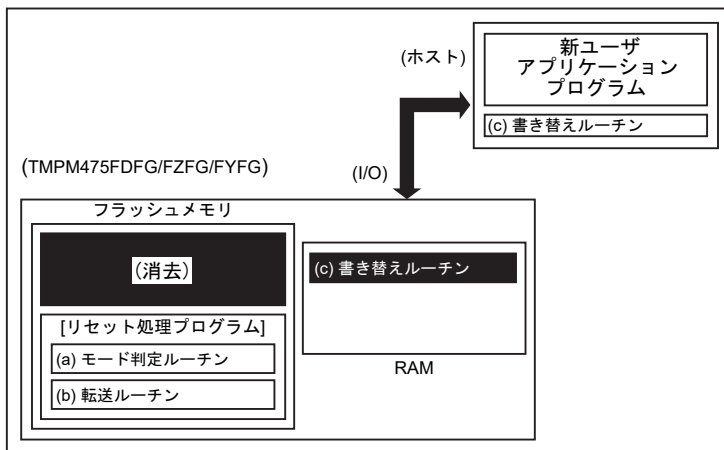
### 24.4.2.3 Step-3

ユーザブートモードに移ると、(b)転送ルーチンを使用して、転送元(ホスト)より(c)書き替えルーチンを内蔵RAMにロードします。



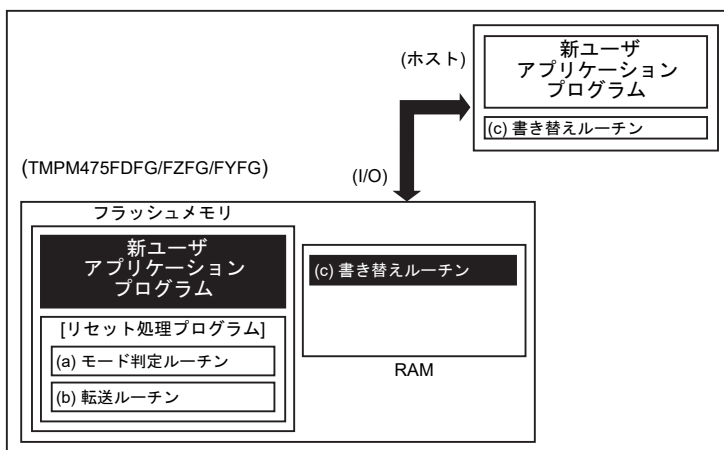
24.4.2.4 Step-4

RAM 上の書き替えルーチンへジャンプし、旧ユーザプログラム領域のライト/消去プロテクトを解除して、消去(任意の消去単位)を行います。



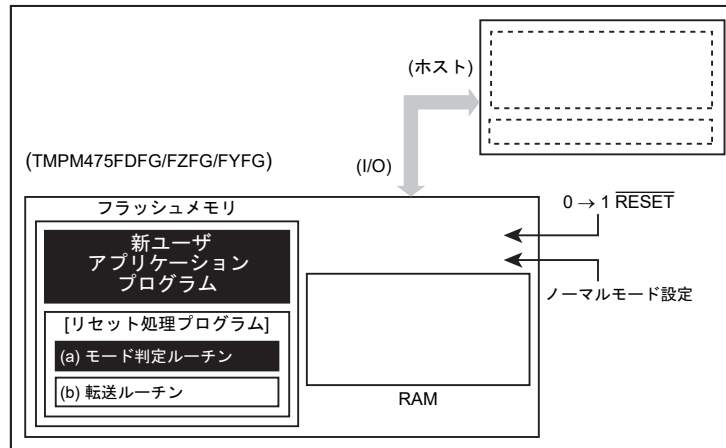
24.4.2.5 Step-5

さらに、RAM 上の(c)書き替えルーチンを実行して、転送元(ホスト)より新ユーザアプリケーションプログラムのデータをロードし、消去した領域に書き込みを行います。書き込みが完了したら、ユーザプログラム領域のライト/消去プロテクトをオンにします。



## 24.4.2.6 Step-6

$\overline{\text{RESET}}$  入力端子を"0"にしてリセットを行い、設定条件をノーマルモードに設定します。リセット解除後、新ユーザアプリケーションプログラムで動作を開始します。



## 24.5

メモリスワップ機能を利用して、ユーザブートプログラムが残るように Page0 と Page1 の領域を交換させてフラッシュメモリの書き替えを実行する方法です。

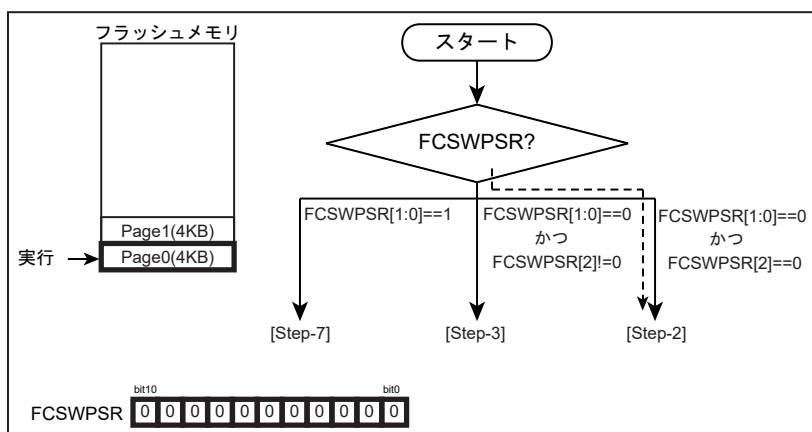
ユーザブートプログラムの書き替え手順の参考例を以下に示します。

(以下では、スワップサイズは 4K バイト、Page1 のプログラムは、Page0 からコピーするものとして、記載します)

### 24.5.1

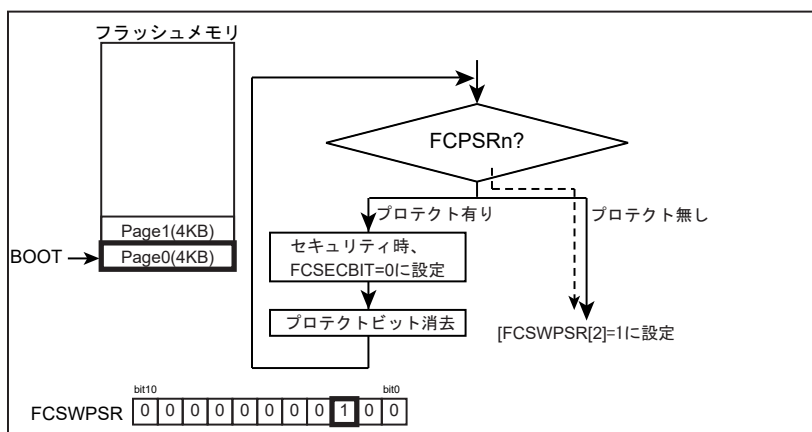
#### 24.5.1.1 Step-1

FCSWPSR[2:0]から 0x0 が読み出せることを確認します。



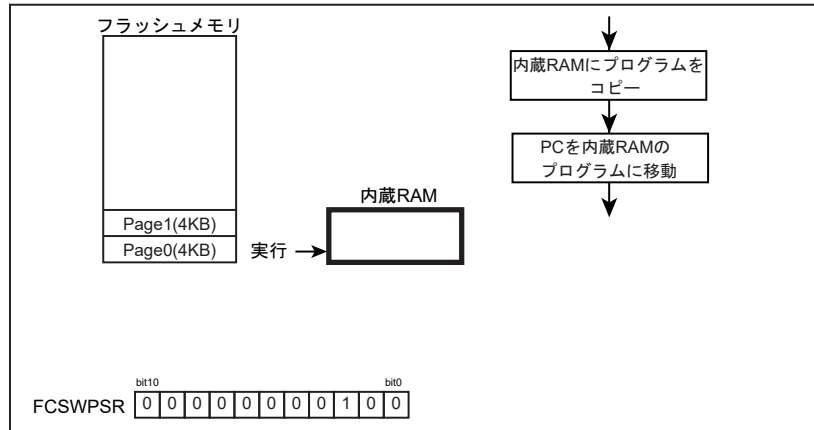
#### 24.5.1.2 Step-2

各 FCPSR レジスタの各ビットにてプロテクト状態をチェックし、プロテクトが無いことを確認します。その後、自動メモリスワップコマンドにて FCSWPSR[2]に"1"をセットします。



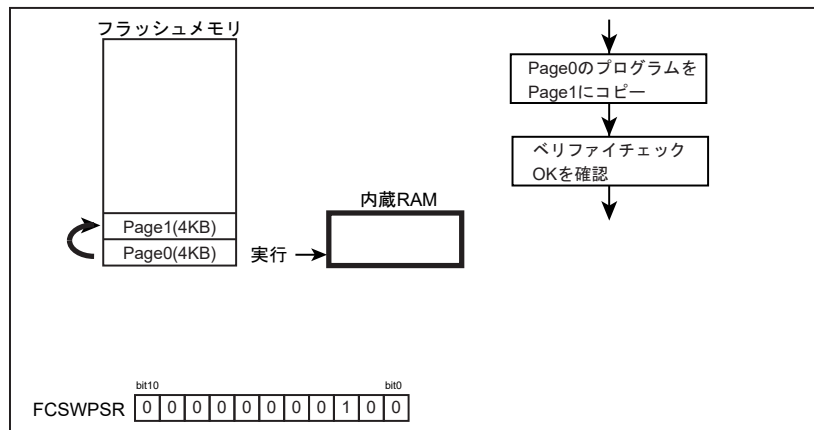
#### 24.5.1.3 Step-3

内蔵 RAM に書き替えルーチンを転送し、PC(プログラムカウンタ)を転送したプログラムに移動します。



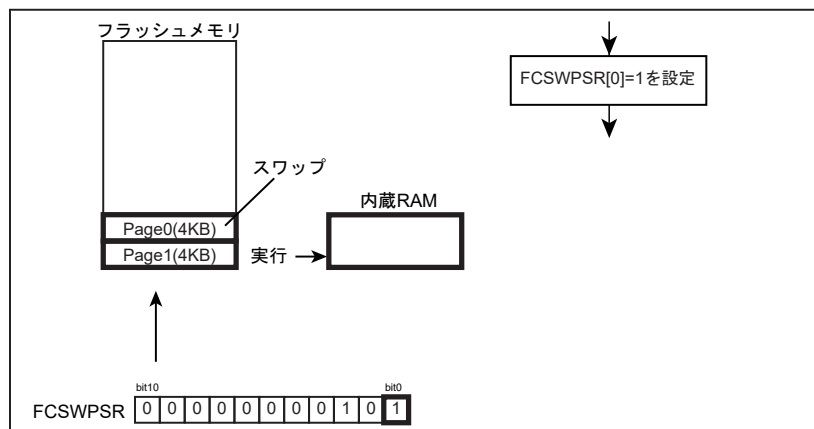
#### 24.5.1.4 Step-4

Page1 を消去し、その後 Page0 のプログラムを Page1 に書き込みます。



#### 24.5.1.5 Step-5

自動メモリスワップコマンドにて FCSWPSR[0]に"1"をセットし、Page0 と Page1 をスワップします。

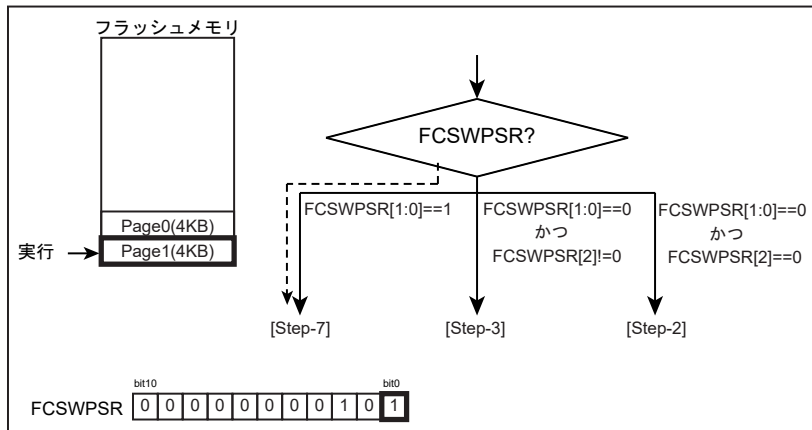


24.5.1.6 Step-6

リセット&リセット解除を行います。

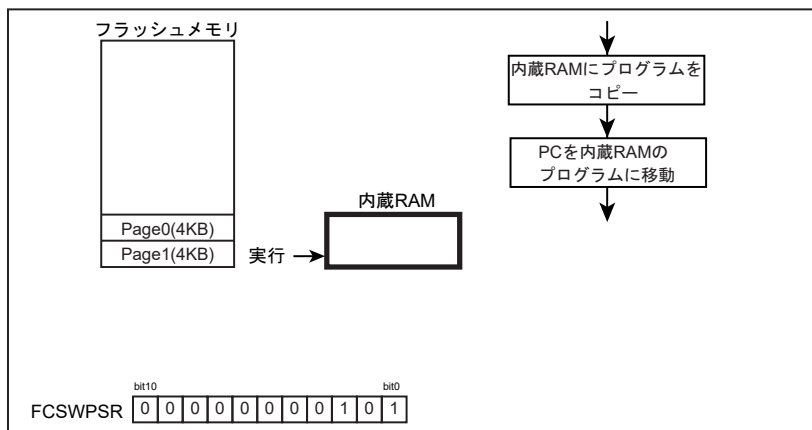
Page1 が 0 番地に割り付けられ、Page1 から起動します。

プログラムは、FCSWPSR[1:0]が"1"の条件用ルーチンへ分岐します。([Step-7]へ)



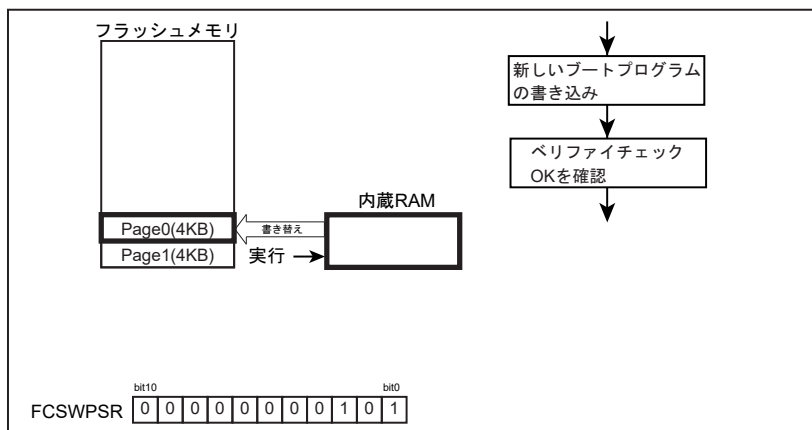
24.5.1.7 Step-7

内蔵 RAM に書き替えルーチンを転送し、PC(プログラムカウンタ)を転送したプログラムに移動します。



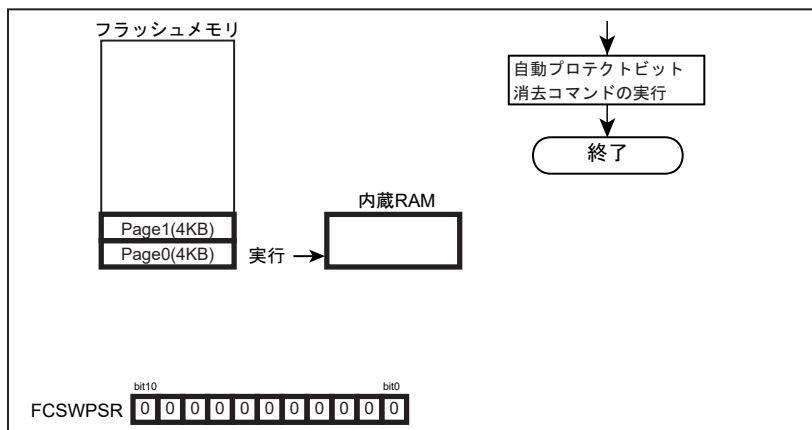
24.5.1.8 Step-8

新しいブートプログラムを Page0 に書き込みます。



### 24.5.1.9 Step-9

自動プロテクトビット消去コマンドを実行します。



## 25

## 25.1

TMPM475FDFG/FZFG/FYFG はデバッグツールと接続するためのデバッグインタフェースとして SWJ-DP (Serial Wire JTAG Debug Port)ユニット、ならびに内部プログラムをトレース出力するための ETM (Embedded Trace Macrocell™)ユニットを搭載しています。トレース出力はマイコン内部の TPIU (Trace Port Interface Unit)を通じてデバッグ用端子(TRACEDATA[3:0], SWV)に出力されます。

SWJ-DP, ETM, TPIU の詳細に関しましては ARM 社からリリースされる"ARM ドキュメンテーションセット Cortex-M4F 用"を参照してください。

## 25.2 SWJ-DP

シリアルワイヤデバッグポート(SWCLK, SWDIO)と、JTAG デバッグポート(TDI, TDO, TMS, TCK, TRST)をサポートしています。

Pin name	Function	Description	I/O
TMS	JTAG	JTAG Test Mode Selection	Input
SWDIO	SW	Serial Wire Data Input/Output	I/O
TCK	JTAG	JTAG Test Clock	Input
SWCLK	SW	Serial Wire Clock	Input
TDO	JTAG	JTAG Test Data Output	Output
SWV	SW	(Serial Wire Viewer Output)	(Output)( )
TDI	JTAG	JTAG Test Data Input	Input
TRST	JTAG	JTAG Test RESET	Input

) SWV

## 25.3 ETM

データ信号 4pin (TRACEDATA[3:0])と クロック信号 1pin (TRACECLK)および、1pin(SWV) によるトレース出力をサポートしています。

## 25.4

Cortex-M4F コアがホールドモードに入ると、ウォッチドッグタイマ(WDT)が自動的に停止します。また、16ビットタイマ(TMRB および TMR16A) はホールドモード時に動作するかどうかを指定することができます。その他の周辺機能は動作を続けます。

## 25.5

### 25.5.1

デバッグツールとの接続方法については、お使いになるツールメーカーが推奨する接続方法を参照してください。また、デバッグインタフェース端子はプルアップ/プルダウン抵抗を内蔵した端子です。外部にプルアップ/プルダウン抵抗を接続する際は注意してください。

) STOP

### 25.5.2

デバッグインタフェース端子は汎用ポートと兼用です。

リセット解除後、デバッグ端子となるもの以外は汎用ポート機能となります。必要に応じてデバッグ端子を使用する設定を行ってください。

デバッグ時にユーザプログラムでリセット解除後すぐに汎用ポートに設定を変更すると、デバッグツールからの制御が受け付けられず、デバッグツールによるデバッグができなくなる場合があります。

デバッグインタフェース端子の使用状況に合わせて、設定を変える場合は注意してください。

#### 25-1

	$\overline{\text{TRST}}$	TDI	TDO / SWV	TCK / SWCLK	TMS / SWDIO	TRACE DATA[3:0]	TRACE CLK
JTAG+SW ( )	o	o	o	o	o	3	3
JTAG+SW ( $\overline{\text{TRST}}$ )	3 (注)	o	o	o	o	x	x
JTAG+TRACE	o	o	o	o	o	o	o
SW	x	x	x	o	o	x	x
SW+SWV	x	x	o	o	o	x	x
デバッグ機能ディセーブル	x	x	x	x	x	x	x

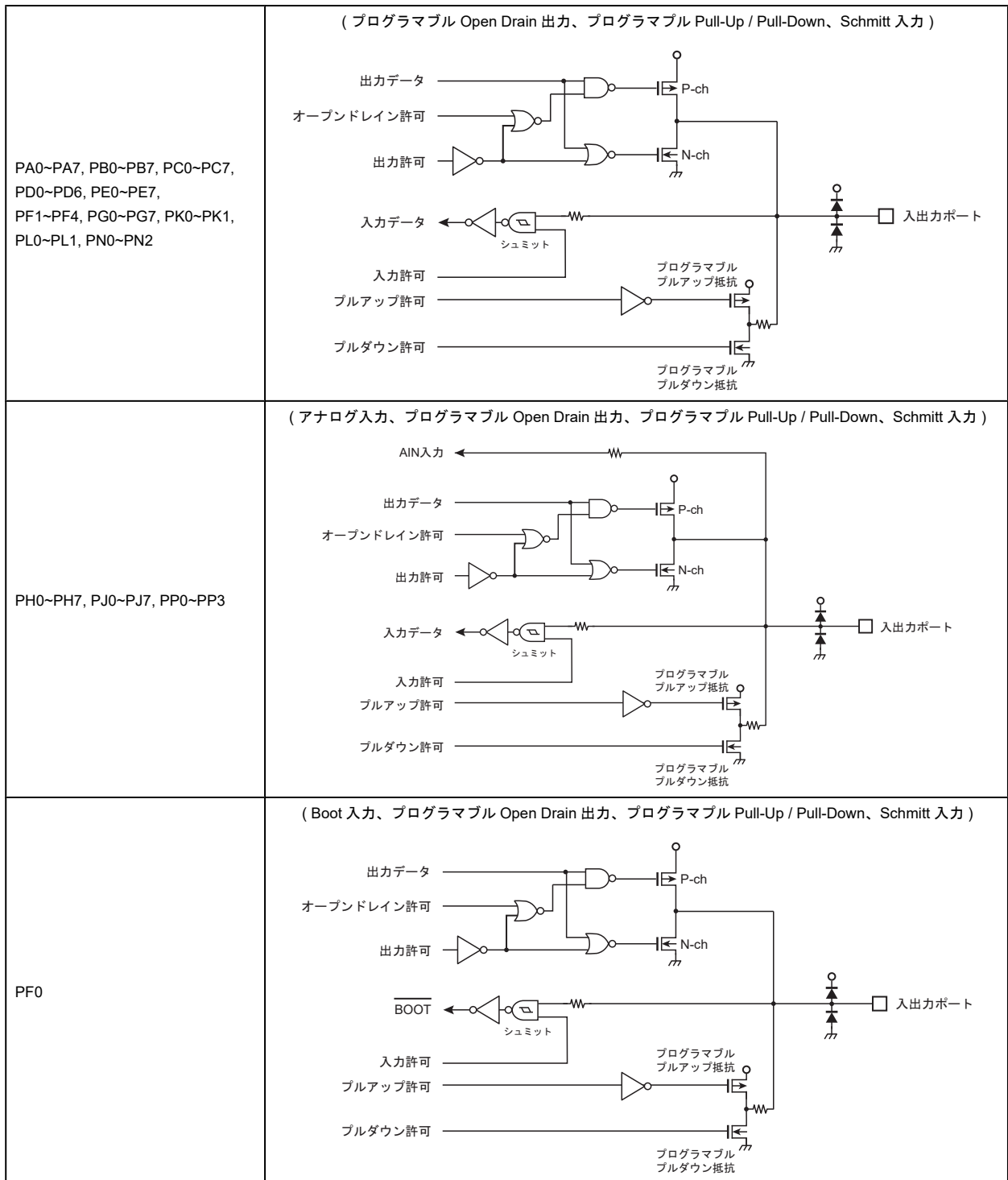
o : イネーブル x : ディセーブル(汎用ポートとして使用可能)

注)  $\overline{\text{TRST}}$ が割り当てられている端子は、 $\overline{\text{TRST}}$ を選択して、オープンにするか"High"レベルを入力してください。

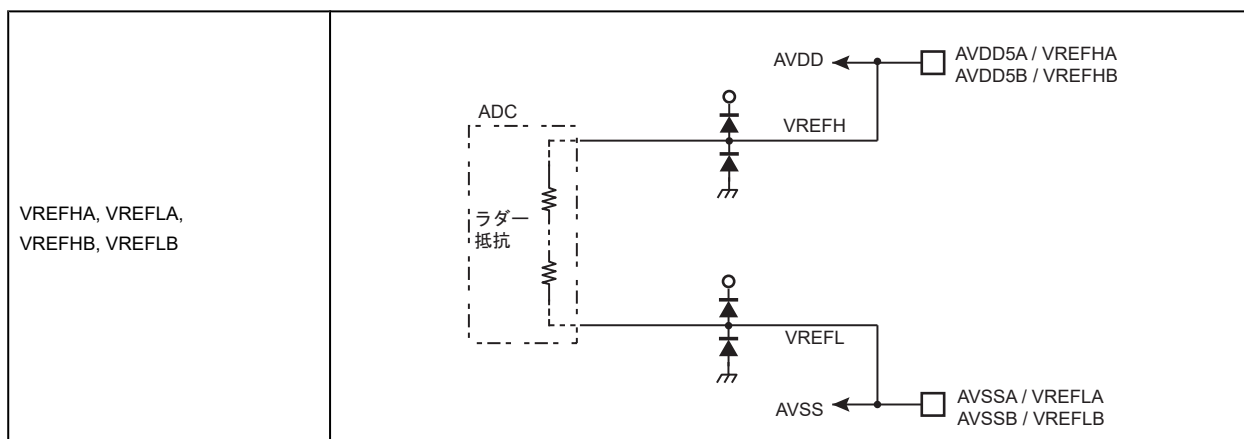
# 第 26 章 ポート等価回路図

ポート等価回路図は、基本的に標準 CMOS ロジック IC 「74HCxx」 シリーズと同じゲート記号を使って書かれています。

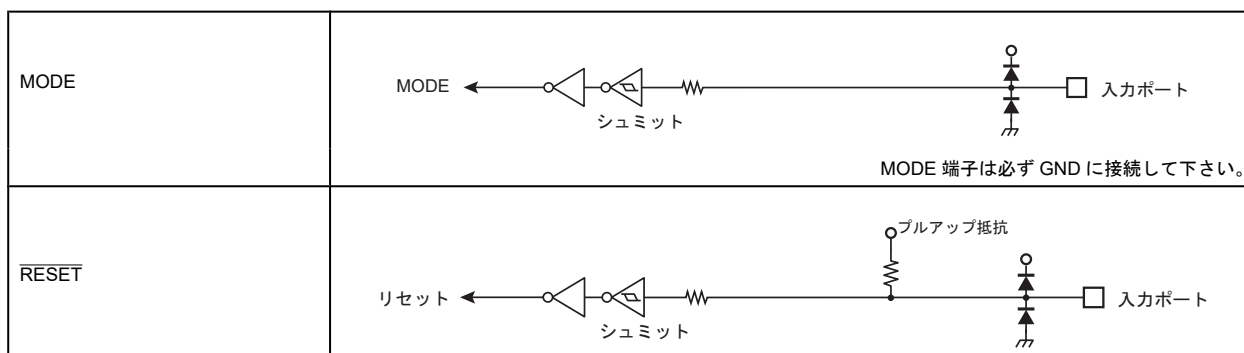
## 26.1 ポート



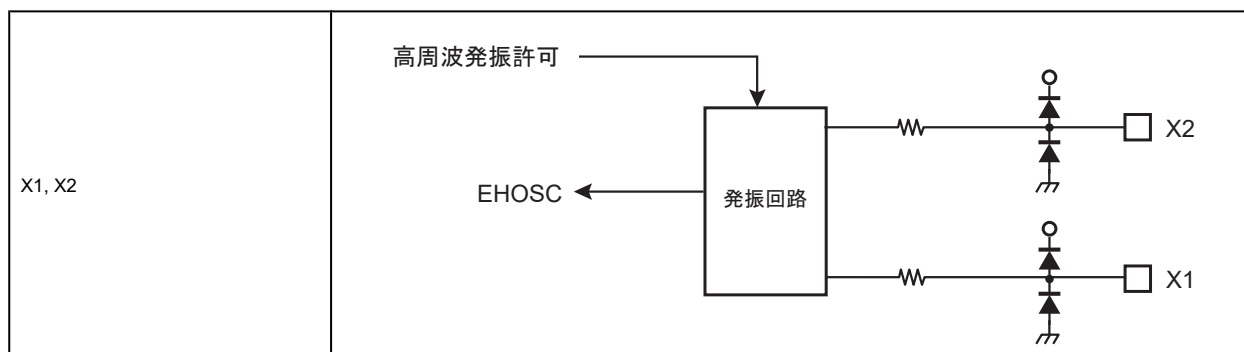
## 26.2 アナログ端子



## 26.3 制御端子



26.4 クロック端子





## 第 27 章 電気的特性

### 27.1 絶対最大定格

項目		記号	定格	単位
電源電圧		DVDD5	-0.3 to 6	V
		RVDD5	-0.3 to 6	
		AVDD5A/B	-0.3 to 6	
電圧保持用キャパシタ端子電圧		VOUT12	-0.3 to 3	V
		VOUT3	-0.3 to 3.9	
入力電圧		V <sub>IN</sub>	-0.3 ~ VDD + 0.3 (注 2)	V
低レベル 出力電流	1 端子	I <sub>OL</sub>	5	mA
	合計	ΣI <sub>OL</sub>	50	
高レベル 出力電流	1 端子	I <sub>OH</sub>	-5	
	合計	ΣI <sub>OH</sub>	-50	
消費電力 (Ta = 85 °C)		PD	600	mW
はんだ付け温度(10 s)		T <sub>SOLDER</sub>	260	°C
保存温度		T <sub>STG</sub>	-55 ~ 125	°C
動作温度		T <sub>OPR</sub>	-40 ~ 85	°C

注 1) 絶対最大定格とは、瞬時たりとも超えてはならない規格であり、どの 1 つの項目も超えることができない規格です。絶対最大定格(電流, 電圧, 消費電力, 温度)を超えると破壊や劣化の原因となり、破裂・燃焼による障害を負うことがあります。従って必ず絶対最大定格を超えないように、応用機器の設計を行ってください。

注 2) VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B

## 27.2 DC 電気的特性 (1/2)

DVSS = AVSSA = AVSSB = 0V, Ta = -40 ~ 85 °C

項目		記号	条件	Min	Typ. (注1)	Max	単位
電源電圧 (注 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f <sub>OSC</sub> = 8 ~ 10 MHz f <sub>sys</sub> = 1 ~ 120 MHz	4.5	-	5.5	V
電源電圧 (FLASH W/E 時) (注 2)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f <sub>OSC</sub> = 8 ~ 10 MHz f <sub>sys</sub> = 1 ~ 120 MHz (Ta (°C) = 0 ~ 70)	4.5	-	5.5	V
電源電圧 (電源上昇、 下降時) (注 3)	DVDD5 RVDD5 AVDD5A AVDD5B	VDD	f <sub>OSC</sub> = 8 ~ 10 MHz f <sub>sys</sub> = 1 ~ 120 MHz	3.9	-	5.5	V
低レベル 入力電圧	シュミット入力	V <sub>IL1</sub>	VDD = 4.5V ~ 5.5V (注 4)	-0.3	-	0.25 VDD	V
高レベル 入力電圧	シュミット入力	V <sub>IH1</sub>	VDD = 4.5V ~ 5.5V (注 4)	0.75VDD	-	VDD+0.3	V
電圧保持用キャパシタ容量(注 5)		C <sub>out</sub>	RVDD5 = 4.5V ~ 5.5V VOUT12, VOUT3	3.3	-	4.7	μF
低レベル出力電圧		V <sub>OL</sub>	I <sub>OL</sub> = 1.6 mA   VDD ≥ 4.5V (注 4)	-	-	0.4	V
高レベル出力電圧		V <sub>OH</sub>	I <sub>OH</sub> = -1.6 mA   VDD ≥ 4.5V (注 4)	4.1	-	-	V
入力リーク電流		I <sub>LI1</sub>	0.0 ≤ V <sub>IN</sub> ≤ VDD (注 4)	-	0.02	±5	μA
出力リーク電流		I <sub>LO</sub>	0.2 ≤ V <sub>IN</sub> ≤ VDD - 0.2 (注 4)	-	0.05	±10	
リセットブルアップ抵抗		R <sub>RST</sub>	4.5 ≤ VDD ≤ 5.5 (注 4)	-	50	150	kΩ
プログラマブルアップ/ダウン抵抗		P <sub>KH</sub>	4.5 ≤ VDD ≤ 5.5 (注 4)	-	50	150	kΩ
シュミット入力幅		V <sub>TH</sub>	4.5 ≤ VDD ≤ 5.5 (注 4)	0.3	0.6	-	V
Pin 容量(電源端子を除く)		C <sub>IO</sub>	f <sub>c</sub> = 1 MHz	-	-	10	pF

注 1) Typ. 値は特に指定のない限り Ta = 25 °C, DVDD5 = AVDD5A = AVDD5B = RVDD5 = 5V の値です。

注 2) DVDD5、AVDD5A、AVDD5B、RVDD5 は同電圧で使用してください。

注 3) 電源上昇時 (パワーオン時) と電圧検出回路 (VLTD) イネーブルでの電源電圧下降時における電圧範囲となります。なお、3.9V ≤ VDD < 4.5V の範囲では 12 ビット A/D コンバータ変換特性および AC 電気的特性は保証外となります。

注 4) VDD = DVDD5 = RVDD5 = AVDD5A = AVDD5B

注 5) VOUT12 端子と VOUT3 端子は同値の保持用キャパシタを介して GND に接続してください。VOUT12 端子と VOUT3 端子から IC 外部への電源供給はできません。

27.3 DC 電氣的特性 (2/2)

DVDD5 =RVDD5 = AVDD5A = AVDD5B = 4.5 V ~ 5.5 V, Ta = -40 ~ 85 °C

項目	記号	条件		Min	Typ. (注 1)	Max	単位
		システムクロック (fsys)	動作条件				
NORMAL	IDD	120 MHz	表 27-1 と表 27-2 を参照してください。	-	45	100	mA
IDLE				-	13	50	
NORMAL		80 MHz		-	34	75	
IDLE				-	9.8	38	
STOP		停止		-	1.6	25	

注 1) Typ. 値は、特に指定のない限り Ta=25°C、DVDD5 = AVDD5A = AVDD5B = RVDD5 = 5V です。

注 2) 80MHz 時の IDD は参考値です。

表 27-1 IDD 測定条件 (端子設定、発振回路)

		NORMAL	IDLE	STOP
端子設定	DVDD5 = AVDD5A = AVDD5B = RVDD5	4.5 ~ 5.5 V		
	x1, X2 端子	発振子接続		
	入力端子	固定		
	出力端子	開放		
動作条件 (発振回路)	外部高速発振器(EOSC)	発振		停止
	内部高速発振器 (IOSC)	停止		
	fsys 用 PLL	動作(12 通倍)		停止

表 27-2 IDD 測定条件 (CPU、周辺回路)

回路	搭載回路数	NORMAL	IDLE	STOP
CPU	1	動作 (ドライストン Ver. 2.1)	停止	停止
A-VE	2	動作	停止	停止
μDMAC	1	停止	停止	停止
ADC	2	動作	停止	停止
TMRB	10	動作	停止	停止
WDT	1	停止	停止	停止
SIO/UART	4	動作	停止	停止
I2C	1	停止	停止	停止
PMD	2	動作	停止	停止
A-ENC	2	動作	停止	停止
I/O port	-	動作	停止	停止
VLTD	1	動作	動作	停止
OFD	1	動作	停止	停止
CAN	1	停止	停止	停止

## 27.4 12 ビット A/D コンバータ変換特性

DVDD5 = RVDD5 = AVDD5A / VREFHA = AVDD5B / VREFHB = 4.5 V ~ 5.5 V

DVSS = AVSSA / VREFLA = AVSSB / VREFLB = 0V, Ta = -40 ~ 85 °C

項目	記号	条件	Min	Typ.	Max	単位
アナログ基準電圧(+)	VREFHA VREFHB	-	-	AVDD	-	V
アナログ入力電圧	VAIN	-	AVSS	-	AVDD	V
アナログ基準電圧電源電流 (注 1)	IREF	DVSS = AVSS	-	3.5	5.0	mA
消費電流(注 1) A/D 変換時	-	IREF を除く	-	-	6.0	mA
積分非直線性誤差	-	AIN 負荷抵抗 ≤ 600 Ω AIN 負荷容量 ≥ 0.1 μF 変換時間 ≥ 1.00 μs 変換クロック周波数=120MHz (注 4)	-	-	± 6	LSB
微分非直線性誤差			-	-	± 5	
オフセット誤差			-	-	± 5	
フルスケール誤差			-	-	± 6	
総合誤差			-	-	-10 ~ +6	

注 1) AD コンバータ 1 ユニットの電流です。

注 2) 1LSB = (AVDD - AVSS)/4096 [V]

注 3) AVDD = AVDD5A = AVDD5B、AVSS = AVSSA = AVSSB

注 4) 2 ユニット AD コンバータのみ動作時の特性です。

## 27.5 AC 電気的特性

### 27.5.1 AC 測定条件

本章に記載されている AC 特性は、特に指定のない限り以下の条件での測定結果です。

- ・ 出力レベル: High =  $0.8 \times VDD$ 、Low =  $0.2 \times VDD$
- ・ 入力レベル: DC 電気的特性の"低レベル入力電圧/高レベル入力電圧"参照
- ・ 負荷容量: CL = 30pF

注) VDD = DVDD5 = AVDD5A = AVDD5B

### 27.5.2 シリアルチャネル (SIO/UART)

#### 27.5.2.1 I/O インタフェースモード (VDD = 4.5 ~ 5.5V)

表中の x は SIO の動作クロックの周期を表します。SIO の動作クロックは、システムクロック fsys と同じ周期です。この周期は、クロックギアの設定に依存します。

##### (1) SCLK 入力モード (Ta = -40 ~ 85°C)

[データ入力]

項目	記号	計算式		80 MHz		120 MHz		単位
		Min	Max	Min	Max	Min	Max	
SCLK クロック High 幅(入力)	t <sub>SCH</sub>	4x	-	50	-	33.3	-	ns
SCLK クロック Low 幅(入力)	t <sub>SCL</sub>	4x	-	50	-	33.3	-	
SCLK 周期	t <sub>SCY</sub>	8x	-	100	-	66.6	-	
有効 Data 入力 ← SCLK 立ち上がり/立ち下がり (注 1)	t <sub>SRD</sub>	30	-	30	-	30	-	
SCLK 立ち上がり/立ち下がり(注 1) → Input Data 保持	t <sub>HSR</sub>	x + 30	-	42.5	-	38.3	-	

[データ出力]

項目	記号	計算式		80 MHz		120 MHz		単位
		Min	Max	Min	Max	Min	Max	
SCLK クロック High 幅(入力)	t <sub>SCH</sub>	4x	-	82.5 (注 3)	-	70.0 (注 3)	-	ns
SCLK クロック Low 幅(入力)	t <sub>SCL</sub>	4x	-	82.5 (注 3)	-	70.0 (注 3)	-	
SCLK 周期	t <sub>SCY</sub>	8x	-	165	-	140	-	
Output Data ← SCLK 立ち上がり/立ち下がり (注 1)	t <sub>OSS</sub>	t <sub>SCY</sub> / 2 - 3x - 45 (注 2)	-	0 (注 2)	-	0 (注 2)	-	
SCLK 立ち上がり/立ち下がり(注 1) → Output Data 保持	t <sub>OHS</sub>	t <sub>SCY</sub> / 2	-	82.5	-	70.0	-	

注 1) SCLK 立ち上がり/立ち下がり

SCLK 立ち上がりモードの場合は SCLK 立ち上がり、SCLK 立ち下がりモードの場合は SCLK 立ち下がりのタイミングです。

注 2) 計算値がマイナスにならない範囲の SCLK 周期で使用して下さい。

注 3) t<sub>OSS</sub> がマイナスにならない最小値を示しています。計算式による値ではありません。

(2) SCLK 出力モード (Ta = - 40 ~ 85°C)

[データ入出力]

項目	記号	計算式		80 to 120 MHz		単位
		Min	Max	Min	Max	
SCLK 周期(プログラマブル) (注 3)	t <sub>SCY</sub>	2x	-	50	-	ns
Output Data ← SCLK 立ち上がり/立ち下がり	t <sub>OSS</sub>	t <sub>SCY</sub> /2 - 25 (注 1)	-	0 (注 2)	-	
SCLK 立ち上がり/立ち下がり→ Output Data 保持	t <sub>OHS</sub>	t <sub>SCY</sub> /2 - 25 (注 1)	-	0 (注 2)	-	
有効 Data 入力 ← SCLK 立ち上がり/立ち下がり	t <sub>SRD</sub>	45	-	45	-	
SCLK 立ち上がり/立ち下がり→ input Data 保持	t <sub>HSR</sub>	0	-	0	-	

- 注 1) 計算値がマイナスにならない範囲の SCLK 周期で使用して下さい。
- 注 2) マイナスにならない最小値を示しています。計算値ではありません。
- 注 3) SCLK の周期 ≥ 50ns となるように、SCLK を調整してください。

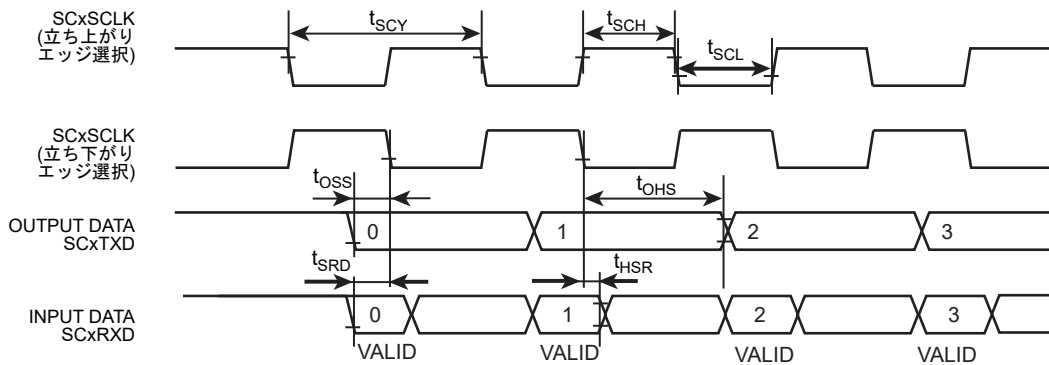


図 27-1 シリアルチャネルのタイミング (SIO)

## 27.5.3 シリアルバスインタフェース (I2C)

### 27.5.3.1 I2C モード

表中の  $x$  は I2C の動作クロックの周期を表します。I2C の動作クロックは、システムクロック  $f_{sys}$  と同じ周期です。この周期は、クロックギアの設定に依存します。

$n$  は SBIxCR<SCK>で指定した SCL 出力クロックの周波数選択値です。

項目	記号	計算式		標準モード		ファーストモード		単位
		Min	Max	Min	Max	Min	Max	
SCL クロック周波数	$t_{SCL}$	0	-	0	100	0	400	kHz
スタートコンディション保持	$t_{HD; STA}$	-	-	4.0	-	0.6	-	$\mu s$
SCL クロック Low 幅(入力) (注 1)	$t_{LOW}$	-	-	4.7	-	1.3	-	$\mu s$
SCL クロック High 幅(入力) (注 2)	$t_{HIGH}$	-	-	4.0	-	0.6	-	$\mu s$
再スタートコンディション セットアップ時間	$t_{SU; STA}$	(注 5)	-	4.7	-	0.6	-	$\mu s$
データ保持時間(入力) (注 3, 4)	$t_{HD; DAT}$	-	-	0.0	-	0.0	-	$\mu s$
データセットアップ時間	$t_{SU; DAT}$	-	-	250	-	100	-	ns
ストップコンディションセットアップ時間	$t_{SU; STO}$	-	-	4.0	-	0.6	-	$\mu s$
ストップコンディションとスタート コンディション間のバスフリー時間	$t_{BUF}$	(注 5)	-	4.7	-	1.3	-	$\mu s$

注 1) SCL クロック LOW 幅(出力):  $(2^{n-1} + 58)/x$

注 2) SCL クロック HIGH 幅(出力):  $(2^{n-1} + 14)/x$

通信規格上、標準モード/高速モードの最高速は 100 kHz/400 kHz です。内部 SCL クロックの周波数の設定は、使用される  $f_{sys}$  と上記注 1,注 2 の計算式にて設定されますのでご注意ください。

注 3) データ保持時間(出力)は内部 SCL から  $4x$  の時間です。

注 4) フィリップススペックでは内部で、SDA 入力時にデータホールド 300nsec を確保して SCL 立ち下がり時の不安定状態を回避することになっていますが、本製品では対応していません。また SCL のエッジスロープコントロール機能を持っていません。従って、SCL/SDA の  $tr/tf$  を含めて BUS 上で上表のデータ保持時間(入力)を守るように設計してください。

注 5) ソフトウェアに依存します。

注 6) フィリップススペックでは、ファーストモードデバイスへの電源供給が切れた場合、バスラインを妨げることがないように SDA および SCL 信号の I/O ピンと電源との接続が外れるようにする必要がありますが、本デバイスでは対応していません。

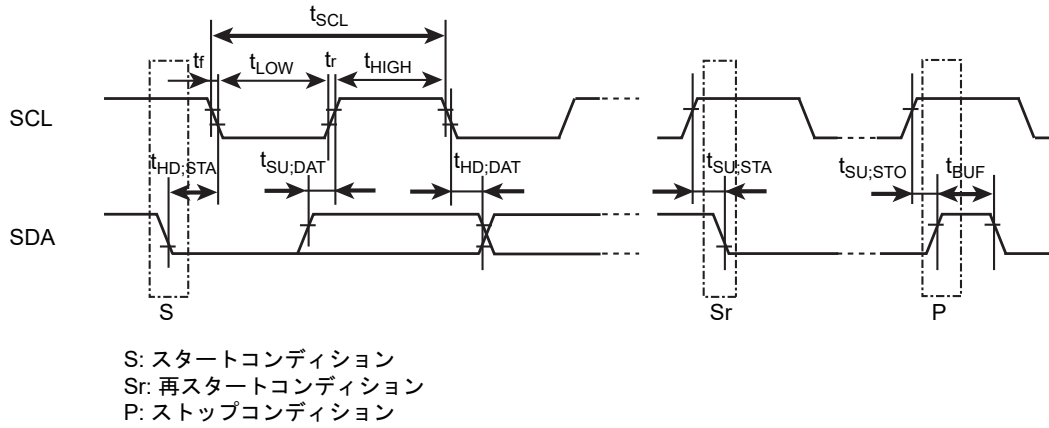


図 27-2 シリアルバスのタイミング(I2C)

## 27.5.4 16 ビットタイマ/カウンタ(TMRB)

### 27.5.4.1 イベントカウンタ

x は TMRB の動作クロックの周期を表します。TMRB の動作クロックは、システムクロック  $f_{sys}$  と同じ周期です。この周期は、クロックギアの設定に依存します。

項目	記号	計算式		80 MHz		120 MHz		単位
		Min	Max	Min	Max	Min	Max	
クロック低レベルパルス幅	$t_{VCKL}$	$2x + 100$	-	125	-	117	-	ns
クロック高レベルパルス幅	$t_{VCKH}$	$2x + 100$	-	125	-	117	-	ns

### 27.5.4.2 キャプチャ

x は TMRB の動作クロックの周期を表します。TMRB の動作クロックは、システムクロック  $f_{sys}$  と同じ周期です。この周期は、クロックギアの設定に依存します。

項目	記号	計算式		80 MHz		120 MHz		単位
		Min	Max	Min	Max	Min	Max	
低レベルパルス幅	$t_{CPL}$	$2x + 100$	-	125	-	117	-	ns
高レベルパルス幅	$t_{CPH}$	$2x + 100$	-	125	-	117	-	ns

## 27.5.5 外部割り込み

表中の x はシステムクロック  $f_{sys}$  の周期を表します。

### 1. STOP 解除割り込み以外

項目	記号	計算式		80 MHz		120 MHz		単位
		Min	Max	Min	Max	Min	Max	
INT0 ~ F 低レベルパルス幅	$t_{INTAL}$	$x + 100$	-	112.5	-	108.3	-	ns
INT0 ~ F 高レベルパルス幅	$t_{INTAH}$	$x + 100$	-	112.5	-	108.3	-	ns

### 2. STOP 解除割り込み

項目	記号	計算式		80 MHz		120 MHz		単位
		Min	Max	Min	Max	Min	Max	
INT0 ~ F 低レベルパルス幅	$t_{INTBL}$	100	-	100	-	100	-	ns
INT0 ~ F 高レベルパルス幅	$t_{INTBH}$	100	-	100	-	100	-	ns

## 27.5.6 デバッグ通信

### 27.5.6.1 AC 測定条件

- ・ 出力レベル: High =  $0.7 \times DVDD5$ , Low =  $0.3 \times DVDD5$
- ・ 負荷容量: TRACECLK CL = 25pF、TRACEDATA CL = 20pF

### 27.5.6.2 SWD インタフェース

項目	記号	Min	Max	単位
CLK 周期	$T_{dck}$	100	–	ns
CLK 立上がり → 出力データ保持	$T_{d1}$	4	–	
CLK 立上がり → 出力データ有効	$T_{d2}$	–	30	
入力データ有効 → CLK 立上がり	$T_{ds}$	20	–	
CLK 立上がり → 入力データ保持	$T_{dh}$	15	–	

### 27.5.6.3 JTAG インタフェース

項目	記号	Min	Max	単位
CLK 周期	$T_{dck}$	100	–	ns
CLK 立下がり → 出力データ保持	$T_{d3}$	4	–	
CLK 立下がり → 出力データ有効	$T_{d4}$	–	50	
入力データ有効 → CLK 立上がり	$T_{ds}$	20	–	
CLK 立上がり → 入力データ保持	$T_{dh}$	15	–	

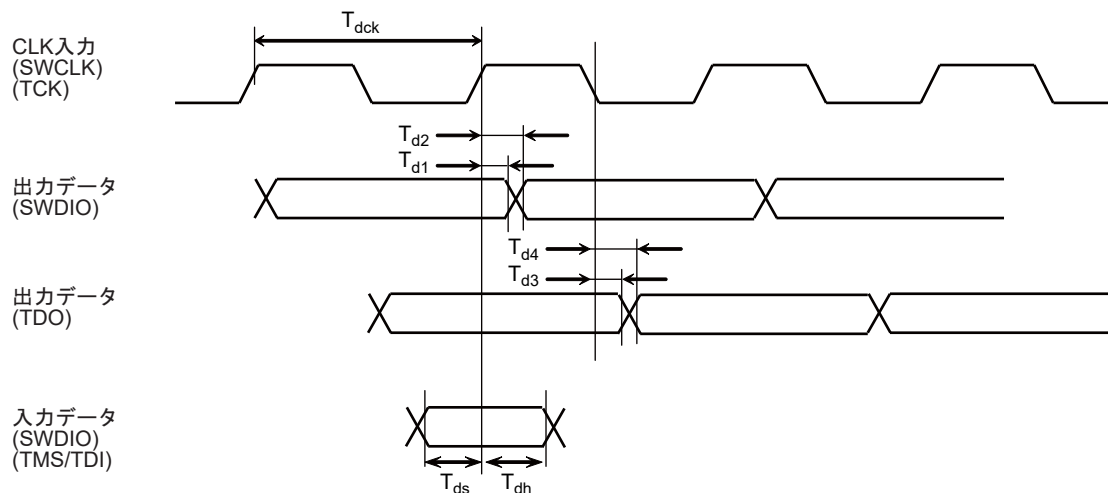


図 27-3 SWD/JTAG インタフェースのタイミング

## 27.5.7 ETM トレース

AC 測定条件は以下の通りです。

- ・ 出力レベル: High =  $0.7 \times DVDD5$ , Low =  $0.3 \times DVDD5$
- ・ 負荷容量: TRACECLK CL = 25pF、TRACEDATA CL = 20pF

項目	記号	Min	Max	単位
TRACECLK 周期	$t_{clk}$	33.3	-	ns
TRACEDATA 有効 ← TRACECLK 立上がり	$t_{setupr}$	2	-	
TRACECLK 立上がり → TRACEDATA 保持	$t_{holdr}$	1	-	
TRACEDATA 有効 ← TRACECLK 立下がり	$t_{setupf}$	2	-	
TRACECLK 立下がり → TRACEDATA 保持	$t_{holdf}$	1	-	

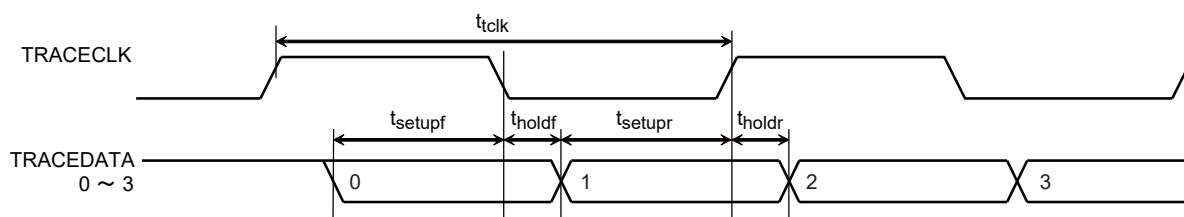


図 27-4 EMT トレースタイミング

## 27.5.8 Flash 特性

項目	条件	Min	Typ.	Max	単位
フラッシュメモリ 消去/書き込み回数	Ta = -40 to 85°C DVDD5 = RVDD5 = AVDD5A = AVDD5B = 4.5 ~ 5.5V	-	-	1000	回

## 27.5.9 内蔵発振回路特性

項目	記号	条件	Min	Typ.	Max	単位
発振周波数 (注 1)	fiosc	Ta = -40 to 85°C	-	-	10	MHz
発振周波数精度 (注 1), (注 2)	-	Ta = 0 to 50°C	-	-	± 1	%
		Ta = -40 to 0°C, Ta = 50 to 85°C	-	-	± 2	%

注 1) 弊社出荷テスト時の値です。リフロー実装直後とそれ以降において、記載の発振精度が必要な場合には、12 通倍 PLL 設定時の最大動作周波数 120MHz を超えないように、発振周波数をその都度トリミングしてください。

注 2) DVDD5 = RVDD5 = AVDD5A = AVDD5B = 5.0V, Ta=25°Cでのトリミング実施後の発振周波数に対する誤差です。

## 27.5.10 外部発振子

項目	記号	条件	Min	Typ.	Max	単位
高周波発振	feosc	Ta = -40 to 85°C	-	10	-	MHz

## 27.6 発振回路

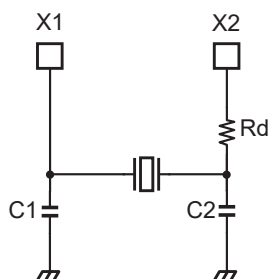


図 27-5 高周波発振回路例

注) 発振の安定には、発振子の位置、負荷容量を適切にする必要があります。これらは基板パターンにより大きな影響を受けます。安定した発振を得るため、ご使用される基板での評価をされるようお願いいたします。

### 27.6.1 セラミック発振子

本製品は(株)村田製作所製セラミック発振子を用いて評価しています。

(株)村田製作所の製品詳細につきましては、同社ホームページを参照してください。

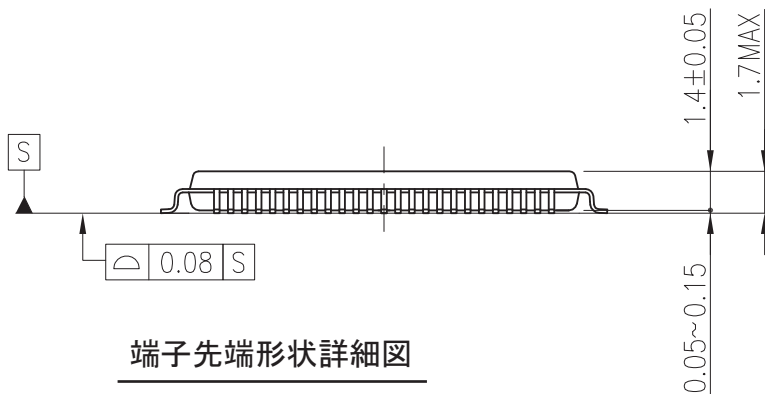
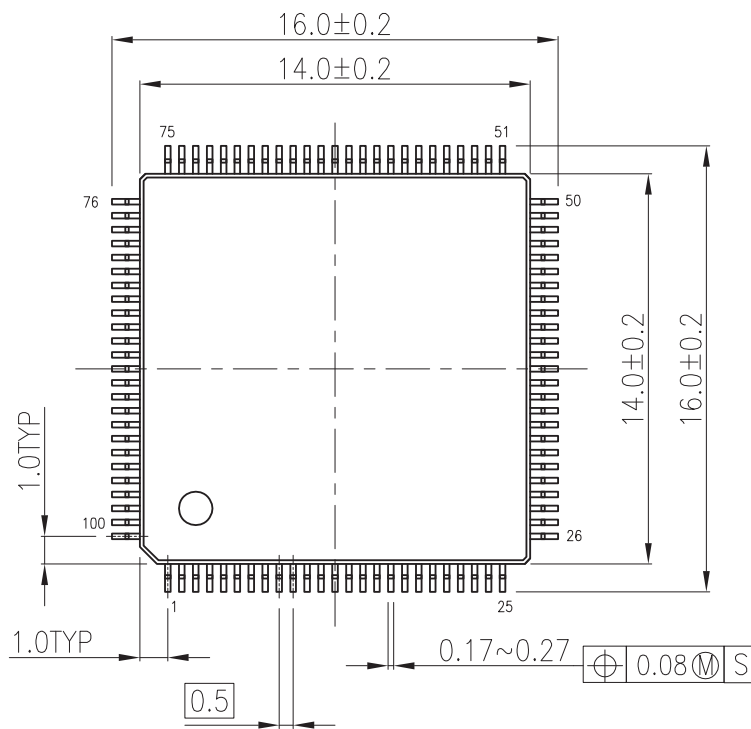


# 第 28 章 パッケージ寸法図

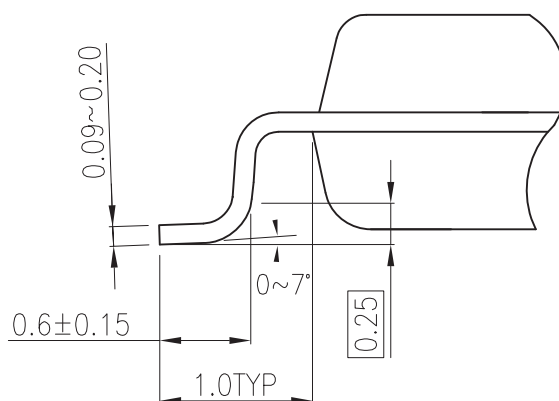
パッケージ型名 : P-LQFP100-1414-0.50-002

Unit: mm

外形寸法図



端子先端形状詳細図



## 製品取り扱い上のお願い

株式会社東芝およびその子会社ならびに関係会社を以下「当社」といいます。  
本資料に掲載されているハードウェア、ソフトウェアおよびシステムを以下「本製品」といいます。

- 本製品に関する情報等、本資料の掲載内容は、技術の進歩などにより予告なしに変更されることがあります。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。また、文書による当社の事前の承諾を得て本資料を転載複製する場合でも、記載内容に一切変更を加えたり、削除したりしないでください。
- 当社は品質、信頼性の向上に努めていますが、半導体・ストレージ製品は一般に誤作動または故障する場合があります。本製品をご使用頂く場合は、本製品の誤作動や故障により生命・身体・財産が侵害されることのないように、お客様の責任において、お客様のハードウェア・ソフトウェア・システムに必要な安全設計を行うことをお願いします。なお、設計および使用に際しては、本製品に関する最新の情報（本資料、仕様書、データシート、アプリケーションノート、半導体信頼性ハンドブックなど）および本製品が使用される機器の取扱説明書、操作説明書などをご確認の上、これに従ってください。また、上記資料などに記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を使用する場合は、お客様の製品単独およびシステム全体で十分に評価し、お客様の責任において適用可否を判断してください。
- 本製品は、特別に高い品質・信頼性が要求され、またはその故障や誤作動が生命・身体に危害を及ぼす恐れ、膨大な財産損害を引き起こす恐れ、もしくは社会に深刻な影響を及ぼす恐れのある機器（以下“特定用途”という）に使用されることは意図されていませんし、保証もされていません。特定用途には原子力関連機器、航空・宇宙機器、医療機器（生命直結機器）、車載・輸送機器、防衛関連機器などが含まれますが、本資料に個別に記載する用途は除きます。特定用途に使用された場合には、当社は一切の責任を負いません。なお、詳細は当社営業窓口まで、または当社 Web サイトのお問い合わせフォームからお問い合わせください。
- 本製品を分解、解析、リバースエンジニアリング、改造、改変、翻案、複製等しないでください。
- 本製品を、国内外の法令、規則及び命令により、製造、使用、販売を禁止されている製品に使用することはできません。
- 本資料に掲載してある技術情報は、製品の代表的動作・応用を説明するためのもので、その使用に際して当社及び第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。
- 別途、書面による契約またはお客様と当社が合意した仕様書がない限り、当社は、本製品および技術情報に関して、明示的にも黙示的にも一切の保証（機能動作の保証、商品性の保証、特定目的への合致の保証、情報の正確性の保証、第三者の権利の非侵害保証を含むがこれに限らない。）をしておりません。
- 本製品、または本資料に掲載されている技術情報を、大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替及び外国貿易法」、「米国輸出管理規則」等、適用ある輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
- 本製品のRoHS適合性など、詳細につきましては製品個別に必ず当社営業窓口までお問い合わせください。本製品のご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用ある環境関連法令を十分調査の上、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は一切の責任を負いかねます。