

TMS320x280x DSP

システム制御および割り込み

リファレンス・ガイド

この資料は、Texas Instruments Incorporated (TI) が英文で記述した資料を、皆様のご理解の一助として頂くために日本テキサス・インスツルメンツ (日本 TI) が英文から和文へ翻訳して作成したものです。
資料によっては正規英語版資料の更新に対応していないものがあります。
日本 TI による和文資料は、あくまでも TI 正規英語版をご理解頂くための補助的参考資料としてご使用下さい。
製品のご検討およびご採用にあたりましては必ず正規英語版の最新資料をご確認下さい。
TI および日本 TI は、正規英語版にて更新の情報を提供しているにもかかわらず、更新以前の情報に基づいて発生した問題や障害等につきましては如何なる責任も負いません。



ご注意

日本テキサス・インスツルメンツ株式会社（以下TIJといたします）及びTexas Instruments Incorporated（TIJの親会社、以下TIJ及びTexas Instruments Incorporatedを総称してTIといたします）は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従いまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIの標準契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従い販売時の仕様に対応した性能を有していること、またはお客様とTIとの間で合意された保証条件に従い合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定される危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは是認するということを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTI からライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、且つその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、且つ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

なお、日本テキサス・インスツルメンツ株式会社半導体集積回路製品販売用標準契約約款もご覧下さい。

<http://www.tij.co.jp/jsc/docs/stdterms.htm>

Copyright © 2006, Texas Instruments Incorporated

日本語版 日本テキサス・インスツルメンツ株式会社

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位（外装から取り出された内装及び個装）又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で（導電性マットにアースをとったもの等）、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。（但し、結露しないこと。）

- 直射日光があたる状態で保管・輸送しないこと。

3. 防湿梱包

- 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。

4. 機械的衝撃

- 梱包品（外装、内装、個装）及び製品単品を落下させたり、衝撃を与えないこと。

5. 熱衝撃

- はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。（個別推奨条件がある時はそれに従うこと。）

6. 汚染

- はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質（硫黄、塩素等ハロゲン）のある環境で保管・輸送しないこと。
- はんだ付け後は十分にフラックスの洗浄を行うこと。（不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。）

以上

以下余白

1	最初にお読みください	13
1	メモリ	17
1.1	フラッシュおよび OTP メモリ.....	18
1.1.1	フラッシュ・メモリ.....	18
1.1.2	OTP メモリ.....	18
1.2	フラッシュおよび OTP 電力モード.....	18
1.2.1	フラッシュおよび OTP の性能.....	20
1.2.2	28x フラッシュ・パイプライン・モード.....	21
1.2.3	フラッシュおよび OTP 内の予約ロケーション.....	23
1.2.4	フラッシュ・コンフィギュレーション・レジスタの変更手順.....	23
1.3	フラッシュおよび OTP レジスタ.....	25
2	コード・セキュリティ・モジュール (CSM)	31
2.1	機能説明.....	32
2.2	他のオンチップ・リソースに対する CSM の影響.....	34
2.3	ユーザー・アプリケーションへのコード・セキュリティの組み込み.....	35
2.3.1	セキュリティのアンロックを必要とする環境.....	36
2.3.2	パスワード・マッチ・フロー.....	37
2.3.3	コード・セキュリティあり / なしのデバイスに対するアンセキュア化の考慮事項.....	38
2.4	セキュリティ・ロジックを保護するための必要事項 / 禁止事項.....	40
2.4.1	必要事項.....	40
2.4.2	禁止事項.....	40
2.5	CSM 機能 - 要約.....	40
3	クロック	41
3.1	クロックとシステム制御.....	42
3.2	OSC および PLL ブロック.....	47
3.2.1	PLL ベースのクロック・モジュール.....	47
3.2.2	メイン・オシレータ障害検出.....	48
3.2.3	XCLKOUT 生成.....	50
3.2.4	PLL 制御 (PLLCCR) レジスタ.....	51
3.2.5	PLL 制御、ステータス、XCLKOUT レジスタの説明.....	53
3.2.6	外部基準オシレータ・クロック・オプション.....	55
3.3	低電力モード・ブロック.....	55
3.4	ウォッチドッグ・ブロック.....	59
3.4.1	ウォッチドッグ・タイマの処理.....	60
3.4.2	ウォッチドッグ・リセットまたはウォッチドッグ割り込みモード.....	60
3.4.3	低電力モードにおけるウォッチドッグ動作.....	61
3.4.4	エミュレーションについての考慮事項.....	61
3.4.5	ウォッチドッグ・レジスタ.....	62
3.5	32 ビット CPU タイマ 0/1/2.....	64

4	汎用入力 / 出力 (GPIO)	69
4.1	GPIO モジュールの概要	70
4.2	設定の概要	71
4.3	デジタル汎用 I/O 制御	73
4.4	入力フィルタリング (Qualification)	74
4.4.1	同期なし (非同期入力) :	74
4.4.2	SYSCLKOUT に対する同期のみ :	74
4.4.3	サンプリング・ウィンドウを使用したフィルタリング (Qualification) :	74
4.5	GPIO およびペリフェラル・マルチプレクス	78
4.6	レジスタ・ビット定義	86
5	ペリフェラル・フレーム	105
5.1	ペリフェラル・フレーム・レジスタ	106
5.2	EALLOW 保護されているレジスタ	108
5.3	デバイス・エミュレーション・レジスタ	112
5.4	ライト後のリード保護	113
6	ペリフェラル割り込み拡張 (PIE)	115
6.1	PIE コントローラの概要	116
6.1.1	割り込み動作シーケンス	116
6.2	ベクタ・テーブル・マッピング	119
6.3	割り込みソース	121
6.3.1	マルチプレクスされた割り込みの処理手順	123
6.3.2	マルチプレクスされたペリフェラル割り込みをイネーブルおよび ディスエーブルにするための手順	124
6.3.3	ペリフェラルから CPU へのマルチプレクスされた割り込みリクエストのフロー	125
6.3.4	PIE ベクタ・テーブル	126
6.4	PIE コンフィギュレーション・レジスタ	134
6.5	PIE 割り込みレジスタ	135
6.5.1	PIE 割り込みフラグ・レジスタ	135
6.5.2	PIE 割り込みイネーブル・レジスタ	136
6.5.3	CPU 割り込みフラグ・レジスタ (IFR)	137
6.5.4	割り込みイネーブル・レジスタ (IER) およびデバッグ割り込みイネーブル・レジスタ (DBGIER) 139	137
6.6	外部割り込み制御レジスタ	144

図目次

1-1	フラッシュ電力モード状態図	20
1-2	フラッシュ・パイプライン	22
1-3	フラッシュ設定方法のフロー・チャート	24
1-4	フラッシュ・オプション・レジスタ (FOPT)	26
1-5	フラッシュ電力レジスタ (FPWR)	26
1-6	フラッシュ・ステータス・レジスタ (FSTATUS)	27
1-7	フラッシュ・スタンバイ・ウェイト・レジスタ (FSTDBYWAIT)	28
1-8	フラッシュ・スタンバイ・アクティブ・ウェイト・カウンタ・レジスタ (FACTIVEWAIT)	28
1-9	フラッシュ・ウェイトステート・レジスタ (FBANKWAIT)	28
1-10	OTP ウェイト・ステート・レジスタ (FOTPWAIT)	30
2-1	CSM ステータスおよび制御レジスタ (CSMSCR)	35
2-2	パスワード・マッチ・フロー (PMF)	37
3-1	クロックおよびリセット・ドメイン	42
3-2	ペリフェラル・クロック制御 0 レジスタ (PCLKCR0)	43
3-3	ペリフェラル・クロック制御 1 レジスタ (PCLKCR1)	43
3-4	高速ペリフェラル・クロック・プリスケラ (HISPCP) レジスタ	46
3-5	低速ペリフェラル・クロック・プリスケラ・レジスタ (LOSPCP)	46
3-6	OSC および PLL ブロック	47
3-7	オシレータ障害検出ロジック図	48
3-8	XCLKOUT 生成	51
3-9	PLLCR 変更手順フロー・チャート	52
3-10	PLLCR レジスタのレイアウト	53
3-11	PLL ステータス・レジスタ (PLLSTS)	53
3-12	XCLKOUT レジスタ (XCLK)	55
3-13	低電力モード制御 0 レジスタ (LPMCR0)	57
3-14	ウォッチドッグ・モジュール	59
3-15	システム制御およびステータス・レジスタ (SCSR)	62
3-16	ウォッチドッグ・カウンタ・レジスタ (WDCNTR)	63
3-17	ウォッチドッグ・リセット・キー・レジスタ (WDKEY)	63
3-18	ウォッチドッグ制御レジスタ (WDCR)	63
3-19	CPU タイマ	64
3-20	CPU タイマ割り込み信号および出力信号	65
3-21	TIMERxTIM レジスタ (x=1、2、3)	66
3-22	TIMERxTIMH レジスタ (x=1、2、3)	66
3-23	TIMERxPRD レジスタ (x=1、2、3)	66
3-24	TIMERxPRDH レジスタ (x=1、2、3)	66
3-25	TIMERxTCR レジスタ (x=1、2、3)	67
3-26	TIMERxTPR レジスタ (x=1、2、3)	68
3-27	TIMERxTPRH レジスタ (x=1、2、3)	68
4-1	動作のモード	70
4-2	サンプリング・ウィンドウを使用した入力フィルタリング (Qualification)	75
4-3	入力フィルタリング (Qualification) クロック・サイクル	77
4-4	GPIO ポート A MUX 1 (GPAMUX1) レジスタ	86
4-5	GPIO ポート A MUX 2 (GPAMUX2) レジスタ	88

4-6	GPIO ポート B MUX 1 (GPBMUX1) レジスタ	91
4-7	GPIO ポート A Qualification 制御 (GPACTRL) レジスタ	92
4-8	GPIO ポート B Qualification 制御 (GPBCTRL) レジスタ	93
4-9	GPIO ポート A Qualification セレクト 1 (GPAQSEL1) レジスタ	94
4-10	GPIO ポート A Qualification セレクト 2 (GPAQSEL2) レジスタ	94
4-11	GPIO ポート B Qualification セレクト 1 (GPBQSEL1) レジスタ	95
4-12	GPIO ポート A 方向 (GPADIR) レジスタ	96
4-13	GPIO ポート B 方向 (GPBDIR) レジスタ	96
4-14	GPIO ポート A プルアップ・ディスエーブル (GPAPUD) レジスタ	97
4-15	GPIO ポート B プルアップ・ディスエーブル (GPBPUD) レジスタ	97
4-16	GPIO ポート A データ (GPADAT) レジスタ	98
4-17	GPIO ポート B データ (GPBDAT) レジスタ	99
4-18	GPIO ポート A セット、クリア、およびトグル (GPASET、GPACLEAR、GPATOGGLE) レジスタ	100
4-19	GPIO ポート B セット、クリア、およびトグル (GPBSET、GPBCLEAR、GPBTOGGLE) レジスタ	101
4-20	GPIO XINT1、XINT2、XNMI 割り込みセレクト (GPIOXINT1SEL、GPIOXINT2SEL、GPIOXNMISEL) レジスタ	102
4-21	GPIO 低電力モード・ウェイクアップ・セレクト (GPIOLPMSSEL) レジスタ	103
5-1	デバイス構成 (DEVICECNF) レジスタ	112
5-2	パート ID レジスタ	113
5-3	REVID レジスタ	113
6-1	概要 : PIE ブロックを使用した割り込みのマルチプレクス	116
6-2	典型的な PIE/CPU 割り込み応答 - INTx.y	118
6-3	リセット・フロー・チャート	121
6-4	外部および PIE 割り込みソース	122
6-5	マルチプレクスされた割り込みリクエストのフロー・チャート	125
6-6	PIECTRL レジスタ (アドレス CE0)	135
6-7	PIE 割り込み ACK レジスタ (PIEACK) (アドレス CE1)	135
6-8	PIEIFRx レジスタ (x=1 ~ 12)	136
6-9	PIEIERx レジスタ (x=1 ~ 12)	136
6-10	割り込みフラグ・レジスタ (IFR) - CPU レジスタ	137
6-11	割り込みイネーブル・レジスタ (IER) - CPU レジスタ	140
6-12	デバッグ割り込みイネーブル・レジスタ (DBGIER) - CPU レジスタ	142
6-13	外部割り込み 1 制御レジスタ (XINT1CR) (アドレス 7070h)	144
6-14	外部割り込み 2 制御レジスタ (XINT2CR) (アドレス 7071h)	144
6-15	外部 NMI 割り込み制御レジスタ (XNMICR) - アドレス 7077h	145
6-16	外部割り込み 1 カウンタ (XINT1CTR) (アドレス 7078h)	146
6-17	外部割り込み 2 カウンタ (XINT2CTR) - アドレス 7079h	146
6-18	外部 NMI 割り込みカウンタ (XNMICR) (アドレス 707Fh)	146

表目次

1-1	フラッシュ /OTP コンフィギュレーション・レジスタ	25
1-2	フラッシュ・オプション・レジスタ (FOPT) のフィールドの説明	26
1-3	フラッシュ電力レジスタ (FPWR) のフィールドの説明	26
1-4	フラッシュ・ステータス・レジスタ (FSTATUS) のフィールドの説明	27
1-5	フラッシュ・スタンバイ・ウェイト・レジスタ (FSTDBYWAIT) のフィールドの説明	28
1-6	フラッシュ・スタンバイ・アクティブ・ウェイト・カウンタ・レジスタ (FACTIVEWAIT) のフィールドの説明	28
1-7	フラッシュ・ウェイトステート・レジスタ (FBANKWAIT) のフィールドの説明	29
1-8	OTP ウェイト・ステート・レジスタ (FOTPWAIT) のフィールドの説明	30
2-1	セキュリティ・レベル	32
2-2	CSM の影響を受ける 280x リソース	34
2-3	CSM の影響を受けない 280x リソース	34
2-4	コード・セキュリティ・モジュール (CSM) レジスタ	35
2-5	CSM ステータスおよび制御レジスタ (CSMSCR) のフィールドの説明	36
3-1	PLL、クロック、ウォッチドッグ、および低電力モード・レジスタ	43
3-2	ペリフェラル・クロック制御 0 レジスタ (PCLKCR0) のフィールドの説明	44
3-3	ペリフェラル・クロック制御 1 レジスタ (PCLKCR1) のフィールドの説明	45
3-4	高速ペリフェラル・クロック・プリスケラ (HISPCP) のフィールドの説明	46
3-5	低速ペリフェラル・クロック・プリスケラ・レジスタ (LOSPCP) のフィールドの説明	46
3-6	可能な PLL 設定モード	48
3-7	PLLCR ビットの説明	53
3-8	PLL ステータス・レジスタ (PLLSTS) のフィールドの説明	54
3-9	XCLKOUT レジスタ (XCLK) のフィールドの説明	55
3-10	280x 低電力モード	56
3-11	低電力モード	56
3-12	低電力モード制御 0 レジスタ (LPMCR0) のフィールドの説明	58
3-13	ウォッチドッグ・キー・シーケンスの例	60
3-14	システム制御およびステータス・レジスタ (SCSR) のフィールドの説明	62
3-15	ウォッチドッグ・カウンタ・レジスタ (WDCNTR) のフィールドの説明	63
3-16	ウォッチドッグ・リセット・キー・レジスタ (WDKEY) のフィールドの説明	63
3-17	ウォッチドッグ制御レジスタ (WDCR) のフィールドの説明	63
3-18	CPU タイマ 0、1、2 設定および制御レジスタ	65
3-19	TIMERxTIM レジスタのフィールドの説明	66
3-20	TIMERxTIMH レジスタのフィールドの説明	66
3-21	TIMERxPRD レジスタのフィールドの説明	66
3-22	TIMERxPRDH レジスタのフィールドの説明	66
3-23	TIMERxTCR レジスタのフィールドの説明	67
3-24	TIMERxTPR レジスタのフィールドの説明	68
3-25	TIMERxTPRH レジスタのフィールドの説明	68
4-1	GPIO 制御レジスタ	71
4-2	GPIO 割り込み、および低電力モード・セレクト・レジスタ	71
4-3	GPIO データ・レジスタ	73
4-4	サンプリング周期	75
4-5	サンプリング周波数	75

4-6	ケース 1 : 3 サンプル・サンプリング・ウィンドウ幅	76
4-7	ケース 2 : 6 サンプル・サンプリング・ウィンドウ幅	76
4-8	ペリフェラル入力のデフォルト状態	79
4-9	2808 GPIO MUX 表	79
4-10	2806 GPIO MUX 表	80
4-11	2801 および 2802 GPIO MUX 表	81
4-12	ペリフェラルと GPIO の相互参照	83
4-13	GPIO ポート A MUX 1 (GPAMUX1) レジスタのフィールドの説明	86
4-14	GPIO ポート A MUX 2 (GPAMUX2) レジスタのフィールドの説明	88
4-15	GPIO ポート B MUX 1 (GPBMUX1) レジスタのフィールドの説明	91
4-16	GPIO ポート B MUX 2 (GPBMUX2) レジスタのフィールドの説明	91
4-17	GPIO ポート A Qualification 制御 (GPACTRL) レジスタのフィールドの説明	92
4-18	GPIO ポート B Qualification 制御 (GPBCTRL) レジスタのフィールドの説明	93
4-19	GPIO ポート A Qualification セレクト 1 (GPAQSEL1) レジスタのフィールドの説明	94
4-20	GPIO ポート A Qualification セレクト 2 (GPAQSEL2) レジスタのフィールドの説明	94
4-21	GPIO ポート B Qualification セレクト 1 (GPBQSEL1) レジスタのフィールドの説明	95
4-22	GPIO ポート B Qualification セレクト 2 (GPBQSEL2) レジスタのフィールドの説明	95
4-23	GPIO ポート A 方向 (GPADIR) レジスタのフィールドの説明	96
4-24	GPIO ポート B 方向 (GPBDIR) レジスタのフィールドの説明	96
4-25	GPIO ポート A 内部プルアップ・ディスエーブル (GPAPUD) レジスタのフィールドの説明	97
4-26	GPIO ポート B 内部プルアップ・ディスエーブル (GPBPUD) レジスタのフィールドの説明	97
4-27	GPIO ポート A データ (GPADAT) レジスタのフィールドの説明	98
4-28	GPIO ポート B データ (GPBDAT) レジスタのフィールドの説明	99
4-29	GPIO ポート A セット (GPASET) レジスタのフィールドの説明	100
4-30	GPIO ポート A クリア (GPACLEAR) レジスタのフィールドの説明	100
4-31	GPIO ポート A トグル (GPATOGGLE) レジスタのフィールドの説明	100
4-32	GPIO ポート B セット (GPBSET) レジスタのフィールドの説明	101
4-33	GPIO ポート B クリア (GPBCLEAR) レジスタのフィールドの説明	101
4-34	GPIO ポート B トグル (GPBTOGGLE) レジスタのフィールドの説明	101
4-35	GPIO XINT1 割り込みセレクト (GPIOXINT1SEL) レジスタのフィールドの説明	102
4-36	GPIO XINT2 割り込みセレクト (GPIOXINT2SEL) レジスタのフィールドの説明	102
4-37	GPIO XNMI 割り込みセレクト (GPIOXNMISEL) レジスタのフィールドの説明	102
4-38	GPIO 低電力モード・ウェイクアップ・セレクト (GPIOLPMSSEL) レジスタの フィールドの説明	103
5-1	ペリフェラル・フレーム 0 レジスタ	106
5-2	ペリフェラル・フレーム 1 レジスタ	106
5-3	ペリフェラル・フレーム 2 レジスタ	107
5-4	EALLOW 保護されているレジスタへのアクセス	108
5-5	EALLOW 保護されているデバイス・エミュレーション・レジスタ	108
5-6	EALLOW 保護されているフラッシュ /OTP コンフィギュレーション・レジスタ	108
5-7	EALLOW 保護されているコード・セキュリティ・モジュール (CSM) レジスタ	109
5-8	EALLOW 保護されている PIE ベクタ・テーブル	109
5-9	EALLOW 保護されている PLL、クロック、ウォッチドッグ、および低電力モード・レジスタ ...	110
5-10	EALLOW 保護されている GPIO MUX レジスタ	110
5-11	EALLOW 保護されている eCAN-A レジスタ	111

5-12	EALLOW 保護されている ePWM1 ~ ePWM3 レジスタ	111
5-13	EALLOW 保護されている ePWM4 ~ ePWM6 レジスタ	111
5-14	デバイス・エミュレーション・レジスタ	112
5-15	DEVICECNF レジスタのフィールドの説明	112
5-16	PARTID レジスタのフィールドの説明	113
5-17	REVID レジスタのフィールドの説明	113
5-18	PROTSTART および PROTRANGE レジスタ	114
5-19	PROTSTART の有効値	114
5-20	PROTRANGE の有効値	114
6-1	割り込みのイネーブル	118
6-2	割り込みベクタ・テーブル・マッピング	119
6-3	リセット動作後のベクタ・テーブル・マッピング	120
6-4	280x PIE マルチプレクス・ペリフェラル割り込みベクタ・テーブル	128
6-5	280x PIE ベクタ・テーブル	129
6-6	PIE コンフィギュレーションおよび制御レジスタ	134
6-7	PIECTRL レジスタ・アドレス・フィールドの説明	135
6-8	PIE 割り込み ACK レジスタ (PIEACK) のフィールドの説明	135
6-9	PIEIFRx レジスタのフィールドの説明	136
6-10	PIEIERx レジスタ (x=1 ~ 12) のフィールドの説明	136
6-11	割り込みフラグ・レジスタ (IFR) - CPU レジスタのフィールドの説明	138
6-12	割り込みイネーブル・レジスタ (IER) - CPU レジスタのフィールドの説明	140
6-13	デバッグ割り込みイネーブル・レジスタ (DBGIER) - CPU レジスタのフィールドの説明	142
6-14	外部割り込み 1 制御レジスタ (XINT1CR) のフィールドの説明	144
6-15	外部割り込み 2 制御レジスタ (XINT2CR) のフィールドの説明	144
6-16	外部 NMI 割り込み制御レジスタ (XNMICR) のフィールドの説明	145
6-17	XNMICR レジスタ設定および割り込みソース	145
6-18	外部割り込み 1 カウンタ (XINT1CTR) のフィールドの説明	146
6-19	外部割り込み 2 カウンタ (XINT2CTR) のフィールドの説明	146
6-20	外部 NMI 割り込みカウンタ (XNMICTR) のフィールドの説明	146

最初にお読みください

本書について

このリファレンス・ガイドは、TMS320x280x ファミリのプロセッサに搭載されているシステム制御および割り込みに適用できます。これには、280x ファミリ内のフラッシュベース、ROM ベース、および RAM ベースのデバイスがすべて含まれます。

本書では、280x デジタル・シグナル・プロセッサ (DSP) システム制御および割り込みがさまざまなペリフェラルでどのように機能するかを説明します。本書には以下に関する情報が含まれています。

- フラッシュ・メモリとワンタイム・プログラマブル (OTP) メモリ
- コード・セキュリティ・モジュール (CSM)。これは、TMS320C28x™ デバイスに組み込まれているセキュリティ機能です。
- オシレータ、PLL、XCLKOUT、ウォッチドッグ・モジュール、低電力モードなどのクロック・メカニズム。また、32 ビット CPU タイマについても説明します。
- 280x デバイスの共有ピンの動作を選択するために使用する GPIO MUX レジスタ
- 各種ペリフェラル・レジスタへのライトおよびリードを行うためのペリフェラル・フレームへのアクセス
- 多数の割り込みソースをより小さいセットの割り込み入力にマルチプレクスする、外部およびペリフェラル割り込み拡張 (PIE) ブロックの割り込みソース
- 『TMS320x280xDSP System Control and Interrupts Reference Guide』 (資料番号 SPRU712C) を翻訳しています。

表記規則

本書では、次の表記規則を使用しています。

- 16 進数は、末尾に h が付いて表されるか、または先頭に 0x が付いて表されます。たとえば、次の数字 40h または 0x40 は 16 進数の 40 (10 進数の 64) です。
- 本書内のレジスタは、図に示され、表で説明されます。
 - 各レジスタ図は、長方形で分割されたレジスタのフィールドを表します。各フィールドには、ビット名のラベルが付けられます。開始および終了ビット番号は上に、リード/ライト・プロパティは下にあります。説明文は、プロパティに対して使用される表記を説明します。
 - レジスタ図の予約ビットは、将来のデバイス拡張に使用されるビットを示します。

関連資料

以下の資料では、TI Web サイトで利用できる TMS320x280x および関連サポート・ツールについて説明しています。

データ・マニュアル

SPRS230: — [『TMS320F2801, TMS320F2806, TMS320F2808, UCD9501 Digital Signal Processors Data Manual』](#) には、ピンアウト、信号表記、F280x デバイスの電氣的仕様およびタイミング仕様が含まれています。

アプリケーション・レポート**SPRAA58: — [『TMS320x281x to TMS320x280x Migration Overview』](#)**

TI の TMS320x281x DSP と TMS320x280x DSP の違いを説明して、281x から 280x へのアプリケーション移行を支援します。本書の主な焦点は 281x から 280x への移行ですが、反対の移行 (280x から 281x へ) を検討しているユーザーにも役立ちます。

SPRA550: — [『3.3 V DSP for Digital Motor Control』](#)

3.3V 専用モータ・コントローラのシナリオについて説明し、ほとんどのアプリケーションについて 3.3V ~ 5V 間のインターフェイスの重要な問題が存在しないことを示します。オンチップ 3.3V アナログ・デジタル・コンバータ (ADC) と 5V ADC についても説明します。システム・ノイズと EMI 効果を低減できるコンポーネント・レイアウトとプリント回路基板 (PCB) 設計のガイドラインが要約されています。

SPRA820: — [『Online Stack Overflow Detection on the TMS320C28x DSP』](#)

TMS320C28x™ DSP でのオンライン・スタック・オーバーフロー検出の方法論を示します。DSP/BIOS™ アプリケーションと非 DSP/BIOS アプリケーションの両方でオーバーフロー検出を実装するための関数が含まれている C ソース・コードが提供されます。

SPRA861: — [『RAMDISK: A Sample User-Defined C I/O Driver』](#)

任意のデバイスでハイレベルな CIO 機能の高度なバッファリングを使用するための簡単な方法を提供します。このアプリケーション・レポートは、ユーザー定義デバイス・ドライバのサンプル実装を示します。

SPRA873: — [『Thermo-Electric Cooler Control Using a TMS320F2812 DSP & DRV592 Power Amplifier』](#)

TI の TMS320F2812 デジタル・シグナル・プロセッサ (DSP) と DRV592 パワー・アンプで構成される電子冷却システムを示します。DSP は、内蔵の 12 ビットアナログ・デジタル・コンバータを使用してサーミスタの値をリードし、H ブリッジ DRV592 パワー・アンプへのパルス幅変調波形の直接出力でデジタル比例 / 積分 / 微分フィードバック・コントローラを実装します。ソフトウェアおよびソフトウェア操作手順と共に実験システムの完全な説明が提供されます。

SPRA876: — [『Programming Examples for the TMS320F281x eCAN』](#)

複数のプログラミング例が含まれており、eCAN モジュールが各種動作モードについてどのように設定され、eCAN の迅速なプログラミングに役立つかを示しています。プロジェクトと CANalyzer 設定ファイルは、付属の SPRA876.zip ファイルにすべて含まれています。

SPRA953: — [『IC Package Thermal Metrics』](#)

従来と新規のサーマル・メトリクスについて説明し、システム・レベルのジャンクション温度推定を各種アプリケーションに関連づけて取り上げます。

SPRA958: — [『Running an Application from Internal Flash Memory on the TMS320F281x DSP \(Rev. B\)』](#)

オンチップ・フラッシュ・メモリからの実行について、アプリケーション・ソフトウェアを正しく設定するために必要な要件を扱います。DSP/BIOS™ と非 DSP/BIOS プロジェクトの両方に対する要件が示されます。コード・プロジェクト例が含まれています。

SPRA963: — [『Reliability Data for TMS320LF24x and TMS320F281x Devices』](#)

『Reliability Data for TMS320LF24x and TMS320F281x Devices』 アプリケーション・レポート

SPRA989: — [『F2810, F2811, and F2812 ADC Calibration』](#)

F2810/F2811/F2812 デバイスに搭載されている 12 ビット・アナログ・デジタル・コンバータ (ADC) の絶対精度を向上させる方法について説明しています。このアプリケーション・ノートには、F2812 eZdsp の RAM から実行するプログラム例 (ADCcalibration.zip) が付属しています。

SPRA991: — [『Simulation Fulfills its Promise for Enhancing Debug and Analysis - A White Paper』](#)

より効果的に代替システムを評価できるようにすることによって開発者の開発サイクルの短縮を可能にする拡張されたシミュレーションについて説明しています。

— ユーザーズ・ガイド —

SPRU051: — [『TMS320x281x, 280x Serial Communication Interface \(SCI\) Reference Guide』](#)では、UART として一般的に知られる SCI (2 線式非同期シリアル・ポート) について説明しています。SCI モジュールは、標準 NRZ (non-return-to-zero) フォーマットを使用する CPU と他の非同期ペリフェラルの間のデジタル通信をサポートします。

SPRU059: — [『TMS320x281x, 280x Serial Peripheral Interface \(SPI\) Reference Guide』](#)では、SPI (高速同期シリアル入力/出力 (I/O) ポート) について説明しています。このポートにより、プログラムされた長さ (1 ~ 16 ビット) のシリアル・ビット・ストリームを、プログラムされたビット転送レートで送受信することができます。

SPRU074: — [『TMS320x281x, 280x Enhanced Controller Area Network \(eCAN\) Reference Guide』](#) 確立されたプロトコルを使用して電氣的にノイズの多い環境で他のコントローラと逐次的に通信する eCAN について説明しています。32 個の個別に設定可能なメールボックスとタイムスタンプ機能により、eCAN モジュールは用途の広い堅牢なシリアル通信インターフェイスを提供します。281x DSP に実装されている eCAN モジュールは、CAN 2.0B 規格 (アクティブ) と互換性があります。

SPRU430: — [『TMS320C28x DSP CPU and Instruction Set Reference Guide』](#)では、中央演算処理ユニット (CPU) と、TMS320C28x 固定小数点デジタル・シグナル・プロセッサ (DSP) の命令セットについて説明しています。また、これらの DSP で利用できるエミュレーション機能についても説明します。

SPRU513: — [『TMS320C28x Assembly Language Tools User's Guide』](#) TMS320C28x デバイス用のアセンブリ言語ツール (アセンブル言語コードの開発に使用するアセンブラや他のツール)、アセンブラ疑似命令、マクロ、共通オブジェクト・ファイル・フォーマット、およびシンボリック・デバッグの疑似命令について説明しています。

SPRU514: — [『TMS320C28x Optimizing C Compiler User's Guide』](#) TMS320C28x C/C++ コンパイラについて説明しています。このコンパイラは、ANSI 標準 C/C++ ソース・コードを受け、TMS320C28x デバイス用の TMS320 DSP アセンブリ言語ソース・コードを生成します。

SPRU608: — [『The TMS320C28x Instruction Set Simulator Technical Overview』](#) C28x コアの命令セットをシミュレートするシミュレータ (TMS320C2000 IDE 用の Code Composer Studio 内で利用可能) について説明しています。

SPRU625: — [『TMS320C28x DSP/BIOS Application Programming Interface \(API\) Reference Guide』](#) DSP/BIOS を使用した開発について説明しています。

SPRU716: — [『TMS320x280x Analog-to-Digital Converter \(ADC\) Reference Guide』](#)では、12 ビット・パイプライン ADC であるオンチップ ADC モジュールの設定方法と使用方法について説明しています。

SPRU721: — [『TMS320x280x Inter-Integrated Circuit \(I2C\) Reference Guide』](#)では、TMS320x280x デジタル・シグナル・プロセッサ (DSP) で利用できる Inter-Integrated Circuit (I²C) モジュールの機能と動作について説明しています。

SPRU722: — [『TMS320x280x Boot ROM Reference Guide』](#) ブートローダ (工場プログラムされたブート・ローディング・ソフトウェア) の用途と機能について説明しています。また、デバイス・オンチップ・ブート ROM の他の内容についても説明し、そのメモリ内のどこに情報のすべてが配置されているかを示します。

SPRU790: — [『TMS320x280x Enhanced Quadrature Encoder Pulse \(eQEP\) Reference Guide』](#) eQEP モジュールについて説明しています。高性能のモーションおよび位置制御システムでは、このモジュールをリニアまたはロータリ・インクリメンタル・エンコーダとのインターフェイスに使用して、位置、方向、および速度情報をエンコーダから取得します。この資料には、モジュールとレジスタの説明が含まれます。

商標

- SPRU791:** — [『TMS320x280x Enhanced Pulse Width Modulator \(ePWM\) Module Reference Guide』](#)
デジタル・モータ制御、スイッチ・モード電源制御、UPS（無停電電源装置）、他の形式の電源変換など、エンハンスド PWM モジュールについて説明しています。
- SPRU807:** — [『TMS320x280x Enhanced Capture \(eCAP\) Module Reference Guide』](#)
エンハンスド・キャプチャ・モジュールについて説明しています。この資料には、モジュールとレジスタの説明が含まれます。
- SPRU924:** — [『High-Resolution Pulse Width Modulator \(HRPWM\)』](#)では、パルス幅モジュレータに対する高解像度拡張（HRPWM）の動作について説明しています。

商標

TMS320C28x、Code Composer Studio は、テキサス・インスツルメンツの商標です。
その他すべての商標は各社の所有物です。

メモリ

この章では、28x デジタル・シグナル・プロセッサ (DSP) デバイスのフラッシュ・メモリおよびワンタイム・プログラマブル (OTP) メモリのウェイト・ステートと動作モードを設定するための正しいシーケンスについて説明します。また、この章には、フラッシュおよび OTP 電力モードに関する情報と、フラッシュ・パイプライン・モードをイネーブルにすることによってフラッシュ性能を向上させる方法も含まれています。

ROM 専用デバイスでは、この情報はフラッシュおよび OTP を置き換える ROM に適用されます。

項目	ページ
1.1 フラッシュおよび OTP メモリ	18
1.2 フラッシュおよび OTP 電力モード	18
1.3 フラッシュおよび OTP レジスタ	25

1.1 フラッシュおよびOTP メモリ

このセクションでは、フラッシュとワンタイム・プログラマブル (OTP) という 2 種類のメモリの設定方法を説明します。

ROM 専用デバイスでは、この情報はフラッシュおよび OTP を置き換える ROM に適用されます。

1.1.1 フラッシュ・メモリ

オンチップ・フラッシュは、プログラム空間とデータ・メモリ空間の両方で一様にマップされます。28x デバイスでは、このフラッシュ・メモリは常にイネーブルになります。機能については下記のとおりです。

- **複数のセクタ**
イレーズすることができるフラッシュ・メモリの最小量はセクタです。複数のセクタがあると、プログラムされた一部のセクタを残して特定のセクタだけをイレーズすることができます。
- **コード・セキュリティ**
フラッシュは、コード・セキュリティ・モジュール (CSM) によって保護されます。ユーザーがパスワードをフラッシュにプログラムすることにより、無許可の人間がフラッシュにアクセスすることを防止できます。コード・セキュリティ・モジュールの使用については、[第 2 章](#)を参照してください。
- **低電力モード**
フラッシュが未使用のときに電力を節約するには、2 つのレベルの低電力モードを利用できます。使用可能なフラッシュ電力モードの詳細については、[セクション 1.2](#)を参照してください。
- **設定可能なウェイト・ステート**
設定可能なウェイト・ステートを CPU 周波数に基づいて調整すると、実行速度に対する最適な性能を提供できます。
- **強化された性能**
フラッシュ・パイプライン・モードは、リニア・コード実行の性能が向上します。

1.1.2 OTP メモリ

1K × 16 ブロックのワンタイム・プログラマブル (OTP) メモリは、プログラムとデータ・メモリ空間の両方で一様にマップされます。したがって、OTP を使用してデータまたはコードをプログラムすることができます。このブロックは、フラッシュとは違って 1 回だけプログラム可能であり、イレーズすることはできません。

1.2 フラッシュおよびOTP 電力モード

以下の動作状態はフラッシュおよびOTP メモリに適用されます。

- **リセットまたはスリープ状態**
これは、デバイス・リセット後の状態です。この状態では、バンクおよびポンプがスリープ状態 (最小電力) になります。フラッシュがスリープ状態のとき、フラッシュまたは OTP メモリ・マップ領域への CPU データ・リードやオペコード・フェッチは、電力モードでのスタンバイ状態への変更、およびアクティブ状態への変更を自動的に開始します。このアクティブ状態への移行中は、CPU は自動的にストールされます。アクティブ状態への移行が完了すると、CPU アクセスは通常どおり完了します。
- **スタンバイ状態**
この状態では、バンクおよびポンプはスタンバイ電力モード状態になります。この状態はスリープ状態よりも多くの電力を使用しますが、アクティブまたはリード状態への移行にかかる時間は短くなります。フラッシュがスタンバイ状態のときには、フラッシュまたは OTP メモリ・マップ領域への CPU データ・リードやオペコード・フェッチは、電力モードでのアクティブ状態への変更を自動的に開始します。このアクティブ状態への移行時間中は、CPU は自動的にストールされます。フラッシュ/OTP がアクティブ状態に達すると、CPU アクセスは通常どおり完了します。
- **アクティブまたはリード状態**
この状態では、バンクおよびポンプはアクティブ電力モード状態 (最大電力) になります。フラッシュ/OTP メモリ・マップ領域への CPU リードまたはフェッチ・アクセス・ウェイト・ステートは、FBANKWAIT または FOTPWAIT レジスタによって制御されます。また、フラッシュ・パイプライン

と呼ばれるプリフェッチ・メカニズムをイネーブルにして、リニア・コード実行のフェッチ性能を向上させることもできます。

注： ブート処理中には、280x ブート ROM はフラッシュ内にあるコード・セキュリティ・モジュール (CSM) パスワード・ロケーションのダミー・リードを実行します。このリードは、パスワードが格納されていない新規デバイスまたはイレーズ済みデバイスをアンロックするために実行されます。これにより、フラッシュ・プログラミング、または CSM 保護されている SARAM へのコードのロードを実行できるようになります。パスワードが格納されているデバイスでは、このリードには何の影響もなく、CSM はロックされたままになります (CSM については第 2 章を参照)。このリードの 1 つの効果は、フラッシュがスリープ (リセット) 状態からアクティブ状態に移行することです。

フラッシュ /OTP バンクおよびポンプは、常に同じ電力モードになります。使用可能な電力の状態図については、[図 1-1](#) を参照してください。フラッシュ /OTP メモリ電力モードは、以下のように変更できます。

- **より低い電力状態に移行するには**

より高い電力モードから、より低い電力モードへ PWR モード・ビットを変更します。この変更は、フラッシュ /OTP バンクを、より低い電力状態に瞬時に移行します。このレジスタは、フラッシュ /OTP メモリの外部で実行されているコードによってのみアクセスされる必要があります。

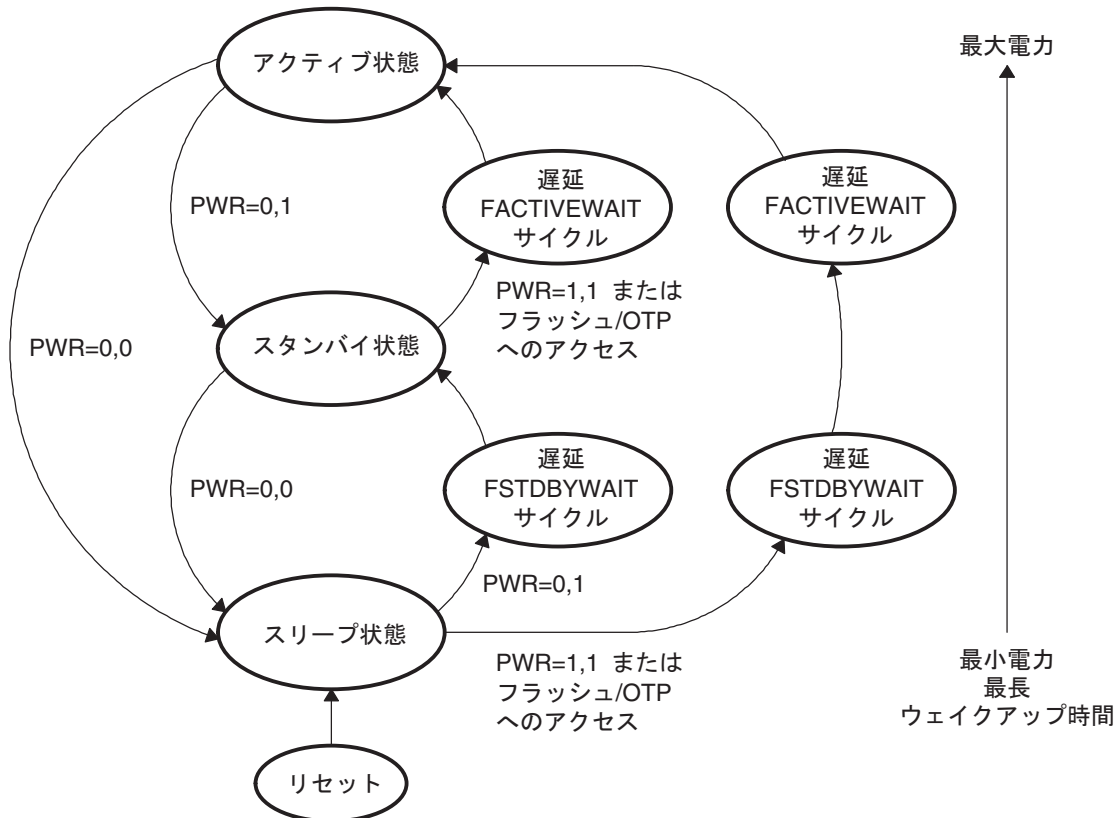
- **より高い電力状態に移行するには**

より低い電力状態から、より高い電力状態に移行するには、2 つのオプションがあります。

1. より低い状態から、より高い状態に FPWR レジスタを変更します。このアクセスは、フラッシュ /OTP メモリをより高い電力状態に引き上げます。
2. リード・アクセスまたはプログラム・オペコード・フェッチ・アクセスによって、フラッシュまたは OTP メモリにアクセスします。このアクセスは、自動的にフラッシュ /OTP メモリをアクティブ状態にします。

より低い電力状態から、より高い電力状態への移行時には、遅延が存在します。[図 1-1](#) を参照してください。この遅延は、フラッシュをより高い電力モードで安定できるようにするために必要です。この遅延中にフラッシュ /OTP メモリへのアクセスが発生した場合、CPU はその遅延が完了するまで自動的にストールします。

図 1-1 フラッシュ電力モード状態図



遅延の持続時間は、FSTDBYWAIT および FACTIVEWAIT レジスタによって決定されます。スリープ状態からスタンバイ状態への移行は、FSTDBYWAIT レジスタによって決定されたカウントの分だけ遅延されます。スタンバイ状態からアクティブ状態への移行は、FACTIVEWAIT レジスタによって決定されたカウントの分だけ遅延されます。スリープ・モード（最小電力）からアクティブ・モード（最大電力）への移行は、FSTDBYWAIT+FACTIVEWAIT の分だけ遅延されます。これらのレジスタは、デフォルト状態のままにしておく必要があります。

1.2.1 フラッシュおよびOTP の性能

フラッシュ/OTP に対する CPU リードまたはデータ・フェッチ動作は、以下のいずれかの形式をとることができます。

- 32 ビット命令フェッチ
- 16 ビットまたは 32 ビット・データ空間リード
- 16 ビット・プログラム空間リード

フラッシュがアクティブ電力状態になったら、バンク・メモリ・マップ領域へのリードまたはフェッチ・アクセスは、フラッシュ・アクセスまたは OTP アクセスとして分類できます。

メイン・フラッシュ・アレイは、行と列に構成されます。行には 2048 ビットの情報が含まれます。フラッシュおよび OTP へのアクセスは、以下の 3 つのタイプのいずれかです。

1. フラッシュ・メモリ・ランダム・アクセス

2048 ビット行への最初のアクセスは、ランダム・アクセスとみなされます。

2. フラッシュ・メモリ・ページ・アクセス

行への最初のアクセスはランダム・アクセスとみなされますが、同じ行内の後続のアクセスはページ・アクセスと呼ばれます。

ランダム・アクセスとページ・アクセスの両方に対するウェイト・ステートの数は、FBANKWAIT レジスタのプログラミングによって設定できます。ランダム・アクセスによって使用されるウェイト・ステートの数は RANDWAIT ビットによって制御され、ページ・アクセスによって使用されるウェイト・ステートの数は PAGEWAIT ビットによって制御されます。FBANKWAIT レジスタは、デフォルトでワーストケース・ウェイト・ステート・カウントになります。したがって、CPU クロック・レートとフラッシュのアクセス時間に基づいて性能を向上させるには、このレジスタを適切な数のウェイト・ステートに対して初期化する必要があります。PAGEWAIT ビットがゼロに設定されている場合、フラッシュは 0 ウェイト・アクセスをサポートします。これは、アクセス時間に対して十分に CPU 速度が低いことを前提としています。ランダムおよびページ・アクセス時間要件を決定するには、特定のデバイスに対するデータ・マニュアルを参照してください。

ROM デバイスでは、フラッシュ・デバイスとのタイミング互換性を可能にするために同じウェイト・ステート設定が保存されます。

3. OTP アクセス

OTP へのリードまたはフェッチ・アクセスは、FOTPWAIT レジスタの OTPWAIT ビットによって制御されます。

OTP へのアクセスはフラッシュよりも長くかかり、ページ・モードはありません。フラッシュの場合と同様に、ROM 専用デバイスの ROM に置き換えられた OTP は、OTP に対する同じウェイト・ステート設定を可能にします。OTP アクセス時間要件を決定するには、特定のデバイスに対するデータ・マニュアルを参照してください。

フラッシュを扱う場合のその他の留意点：

- フラッシュまたは OTP メモリ・マップ領域への CPU ライトは無視されます。それらは単一サイクルで完了します。
- コード・セキュリティ・モジュール (CSM) をセキュアにすると、セキュア・ゾーンの外側からのフラッシュ/TOP メモリ・マップ領域へのリードには、通常のアクセスと同じサイクル数がかかります。ただし、リード動作はゼロを返します。
- CSM パスワード・ロケーションのリードは、16 ウェイトステートについて物理的に組み込まれています。PAGEWAIT および RANDOMWAIT ビットは、これらのロケーションには影響しません。CSM の詳細については、[第 2 章](#)を参照してください。

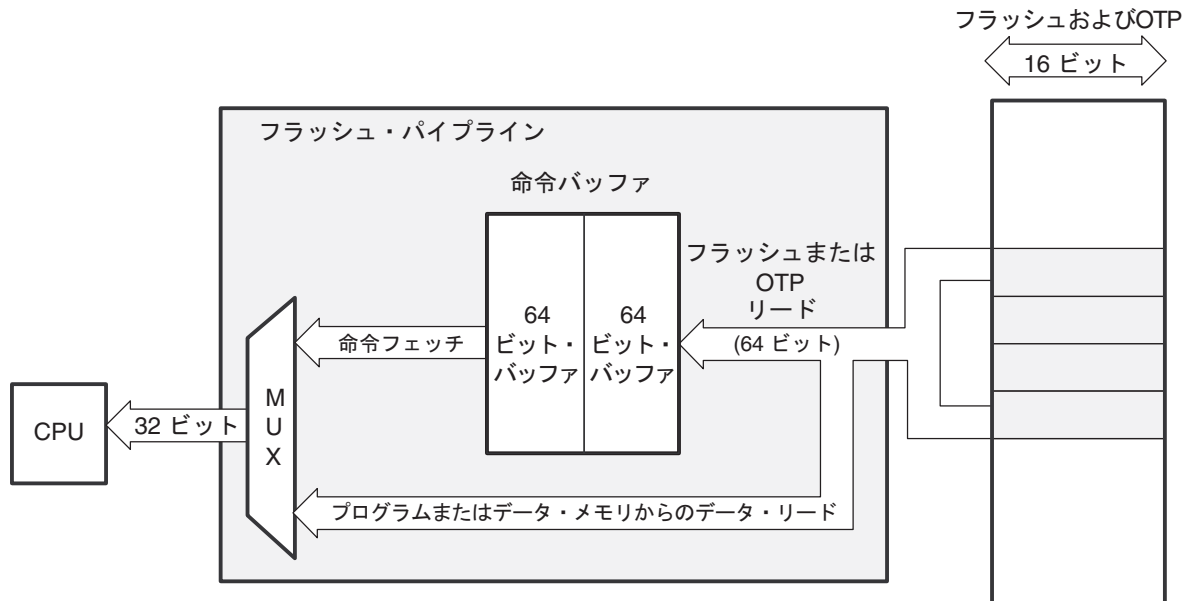
1.2.2 28x フラッシュ・パイプライン・モード

通常、フラッシュ・メモリはアプリケーション・コードを格納するために使用されます。コード実行時には、中断が発生した場合を除き、命令が連続メモリ・アドレスからフェッチされます。一般に、連続アドレスに存在するコードの部分はアプリケーション・コードの大部分を構成しており、リニア・コードと呼ばれます。リニア・コード実行の性能を向上させるために、フラッシュ・パイプライン・モードが実装されています。フラッシュ・パイプライン機能は、デフォルトでディスエーブルになります。FOPT レジスタの ENPIPE ビットをセットすると、このモードがイネーブルになります。フラッシュ・パイプライン・モードは、CPU パイプラインとは独立しています。フラッシュと ROM デバイスの間のコード・タイミング互換性を維持するため、フラッシュ・パイプライン・モードは ROM デバイスにも実装されています。

フラッシュまたは OTP からの命令フェッチは、アクセスごとに 64 ビットをリードします。フラッシュからのアクセスの開始アドレスは、命令ロケーションがフェッチされる 64 ビット内になるように自動的に 64 ビット境界にアラインされます。フラッシュ・パイプライン・モードをイネーブルにすると ([図 1-2](#) を参照)、命令フェッチからリードされた 64 ビットは、64 ビット幅 × 2 段の命令プリフェッチ・バッファに格納されます。このプリフェッチ・バッファの内容は、必要に応じて処理のために CPU へ送られます。

単一の 64 ビット・アクセス内には最大 2 つの 32 ビット命令または最大 4 つの 16 ビット命令があります。大部分の C28x 命令は 16 ビットです。そのため、フラッシュ・バンクからの各 64 ビット命令フェッチについては、CPU で処理する準備ができているプリフェッチ・バッファに最大 4 つの命令が存在する可能性があります。これらの命令を処理するのにかかる時間中に、フラッシュ・パイプラインはフラッシュ・バンクへの別のアクセスを自動的に開始して、次の 64 ビットをプリフェッチします。このようにして、フラッシュ・パイプライン・モードはバックグラウンドで機能し、命令プリフェッチ・バッファをできるだけ一杯の状態に維持します。この手法を使用すると、フラッシュまたは OTP からの連続コード実行の全体的な効率が大幅に改善されます。

図 1-2 フラッシュ・パイプライン



フラッシュ・パイプライン・プリフェッチは、分岐、BANZ、コール、ループなどの命令の実行によって引き起こされた PC 中断でのみ中止されます。これが発生した場合、プリフェッチは中止され、プリフェッチ・バッファの内容はフラッシュされます。これが発生した場合、2 つの可能なシナリオがあります。

1. 宛先アドレスがフラッシュまたは OTP 内にある場合、プリフェッチは中止し、宛先アドレスで再開します。
2. 宛先アドレスがフラッシュまたは OTP の外部にある場合、プリフェッチは中止され、分岐がフラッシュまたは OTP に戻るようになっている場合にのみ再び始まります。フラッシュ・パイプライン・プリフェッチ・メカニズムは、プログラム空間からの命令フェッチだけに適用されます。データ・メモリとプログラム・メモリからのデータ・リードは、プリフェッチ・バッファ機能を利用しないため、プリフェッチ・バッファをバイパスします。たとえば、MAC、DMAC、PREAD などの命令は、データ値をプログラム・メモリからリードします。このリードが発生すると、プリフェッチ・バッファはバイパスされますが、そのバッファはフラッシュされません。データ・リード動作の開始時に命令プリフェッチがすでに進行中の場合は、プリフェッチが完了するまでデータ・リードはストールされます。

1.2.3 フラッシュおよびOTP 内の予約ロケーション

コードとデータをフラッシュおよび OTP メモリに割り当てるときには、以下のことに注意してください。

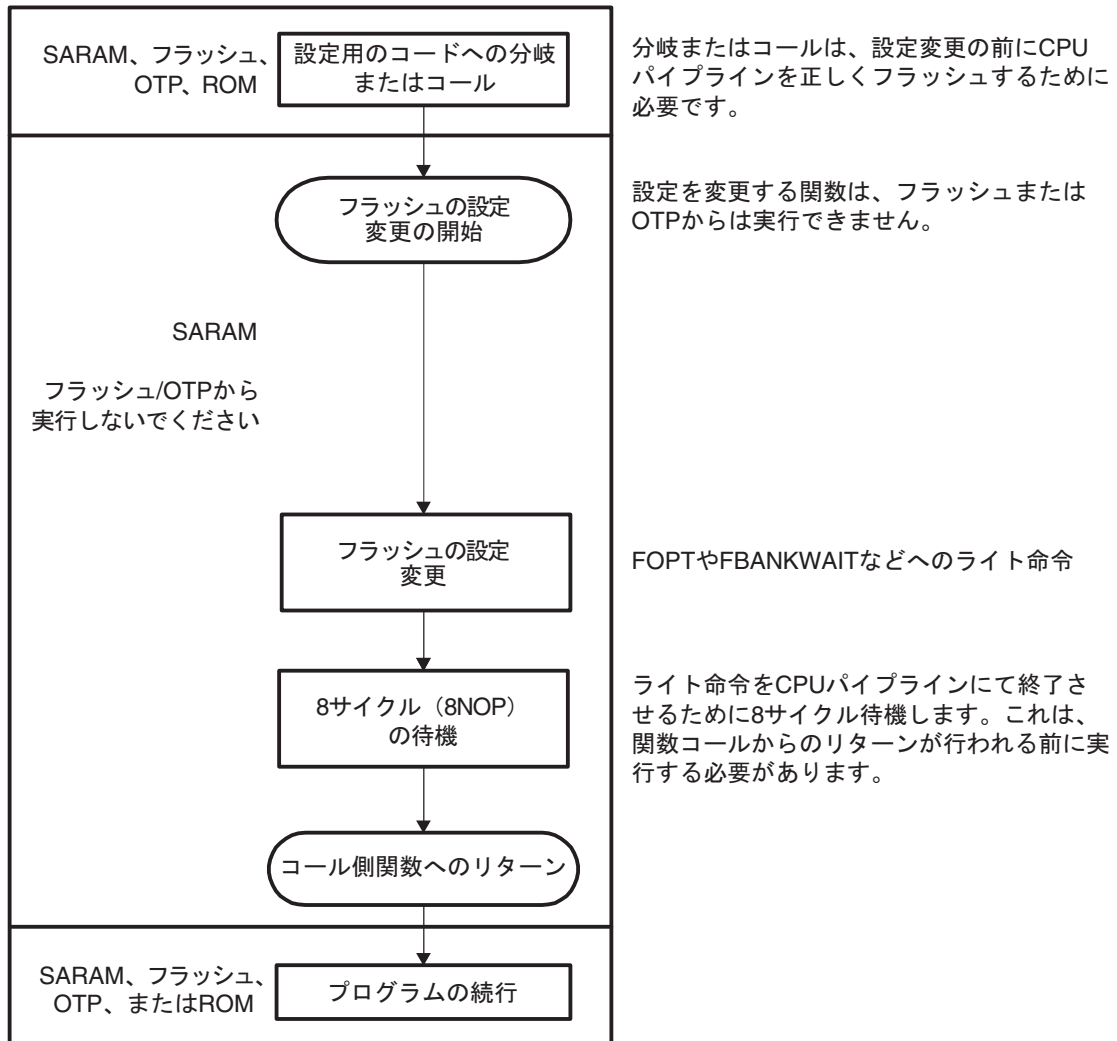
1. アドレス・ロケーションの 0x3F7FF6 と 0x3F7FF7 は、"entry into flash" 分岐命令用に予約されています。"boot to flash" ブート・オプションを使用すると、ブート ROM はアドレス 0x3F7FF6 にジャンプします。ユーザーによってここでプログラムされた分岐命令は、コード実行をアプリケーションのエントリ・ポイントにリダイレクトします。
2. コード・セキュリティ動作については、0x3F7F80 と 0x3F7FF5 の間のアドレスはプログラム・コードまたはデータとして使用できませんが、コード・セキュリティ・パスワードをプログラムするときには 0x0000 にプログラムする必要があります。セキュリティがそれほど重大でない場合は、アドレス 0x3F7F80 ~ 0x3F7FEF をコードまたはデータに対して使用できます。コード・セキュリティ・モジュールの使用については、第 2 章を参照してください。
3. 0x3F7FF0 から 0x3F7FF5 までのアドレスは、データ変数用に予約されており、プログラム・コードが含まれてはいけません。
4. あとでアプリケーションが ROM に移される場合は、チェックサムとデバイス番号識別子を格納するためにフラッシュおよび OTP 内の特定のロケーションが TI によって使用されます。これらのロケーションは、『TMS320C28x DSP CPU and Instruction Set Reference Guide』(SPRU430) の TI への ROM コード送付に関する章に記載されています。

1.2.4 フラッシュ・コンフィギュレーション・レジスタの変更手順

フラッシュの設定中は、フラッシュまたは OTP へのアクセスすることはできません。これには、CPU パイプライン、データ・リード、および命令プリフェッチ動作内にまだある命令が含まれます。設定変更中にアクセスが行われないことを保証するには、FOPT、FPWR、FBANKWAIT または FOTPWAIT レジスタを変更する任意のコードについて [図 1-3](#) の手順に従う必要があります。

この手順は、フラッシュおよび OTP が ROM に置き換えられているデバイスの ROM にも適用されます。

図 1-3 フラッシュ設定方法のフロー・チャート



1.3 フラッシュおよび OTP レジスタ

フラッシュおよび OTP メモリは、表 1-1 に示したレジスタによって設定できます。コンフィギュレーション・レジスタは、すべて EALLOW 保護されています。ビットの説明は図 1-4 ~ 図 1-10 にあります。

表 1-1 フラッシュ /OTP コンフィギュレーション・レジスタ

名前 ⁽¹⁾⁽²⁾	アドレス	サイズ (x16)	説明	ビットの説明
FOPT	0x0A80	1	フラッシュ・オプション・レジスタ	図 1-4
Reserved	0x0A81	1	予約	
FPWR	0x0A82	1	フラッシュ電力モード・レジスタ	図 1-5
FSTATUS	0x0A83	1	ステータス・レジスタ	図 1-6
FSTDBYWAIT ⁽³⁾	0x0A84	1	フラッシュ・スリープ・スタンバイ・ウェイト・レジスタ	図 1-7
FACTIVEWAIT ⁽³⁾	0x0A85	1	フラッシュ・スタンバイ・アクティブ・ウェイト・レジスタ	図 1-8
FBANKWAIT	0x0A86	1	フラッシュ・リード・アクセス・ウェイト・ステート・レジスタ	図 1-9
FOTPWAIT	0x0A87	1	OTP リード・アクセス・ウェイト・ステート・レジスタ	図 1-10

(1) これらのレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) これらのレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

(3) これらのレジスタはデフォルト状態のままにしておく必要があります。

注： フラッシュ・コンフィギュレーション・レジスタには、OTP またはフラッシュ・メモリから実行されているコードによってライトしたり、フラッシュまたは OTP にアクセスしている可能性があるときにライトしたりしないでください。フラッシュ・レジスタへのレジスタ・アクセスは、フラッシュ /OTP メモリの外部で実行しているコードから行う必要があります。また、フラッシュ /OTP に対するアクティビティがすべて完了するまではアクセスしないでください。これを防止するためのハードウェアは含まれていません。

要約すると、フラッシュ /OTP で実行しているコードからフラッシュ・レジスタをリードすることはできます。ただし、そのレジスタにはライトしないでください。

フラッシュ・コンフィギュレーション・レジスタへの CPU ライト・アクセスは、EALLOW 命令の実行によってのみイネーブルにすることができます。EDIS 命令が実行されると、ライト・アクセスはディスエーブルになります。これは、レジスタを誤アクセスから保護します。リード・アクセスは常に使用可能です。レジスタには、EALLOW を実行する必要なく JTAG ポートを通じてアクセスできます。EALLOW 保護については、[セクション 5.2](#) を参照してください。

これらのレジスタは、16 ビットと 32 ビットの両方のアクセスをサポートします。

図 1-4 フラッシュ・オプション・レジスタ (FOPT)

15	予約	1	0
	R-0		ENPIPE R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 1-2 フラッシュ・オプション・レジスタ (FOPT) のフィールドの説明

ビット	フィールド	値	説明 (1) (2) (3)
15-1	予約		
0	ENPIPE	0 1	<p>フラッシュ・パイプライン・モード・ビットをイネーブルにします。このビットがセットされると、フラッシュ・パイプライン・モードはアクティブです。パイプライン・モードは、プリフェッチ命令による命令フェッチの性能を向上させます。詳細については、セクション 1.2.2 を参照してください。</p> <p>パイプライン・モードがイネーブルになっている場合、フラッシュ・ウェイト・ステート（ページおよびランダム）はゼロより大きくなければなりません。</p> <p>フラッシュ・デバイスでは、ENPIPE はフラッシュおよび OTP からのフェッチに影響します。ROM デバイスでは、ENPIPE はフラッシュおよび OTP を置き換えた ROM ブロックからのフェッチに影響します。</p> <p>0 フラッシュ・パイプライン・モードはアクティブではありません。（デフォルト）</p> <p>1 フラッシュ・パイプライン・モードはアクティブです。</p>

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

(3) このレジスタにライトする場合は、[セクション 1.2.4](#) に記載されている手順に従ってください。

図 1-5 フラッシュ電力レジスタ (FPWR)

15	予約	2	1	0
	R-0			PWR R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 1-3 フラッシュ電力レジスタ (FPWR) のフィールドの説明

ビット	フィールド	値	説明 (1) (2)
15-2	予約		
1-0	PWR	00 01 10 11	<p>フラッシュ電力モード・ビット。これらのビットへのライトは、フラッシュ・バンクおよびポンプの現在の電力モードを変更します。フラッシュ・バンク電力モードの変更の詳細については、セクション 1.2 を参照してください。ROM デバイスでは、PWR の変更は ROM の消費電力に影響しません。スタンバイまたはスリープ・モードに移行すると、フラッシュ・デバイスの場合とまったく同様に ROM からの次のアクセスが遅延されます。</p> <p>00 ポンプおよびバンク・スリープ（最小電力）</p> <p>01 ポンプおよびバンク・スタンバイ</p> <p>10 予約（影響なし）</p> <p>11 ポンプおよびバンク・アクティブ（最大電力）</p>

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

図 1-6 フラッシュ・ステータス・レジスタ (FSTATUS)

15					9	8
予約					3VSTAT	
R-0					R/W1C-0	
7	4	3	2	1	0	
予約		ACTIVEWAITS	STDBYWAITS	PWRS		
R-0		R-0	R-0	R-0		

凡例：R/W = リード/ライト、R = リードのみ、W1C = 1 をライトしてクリア、-n = リセット後の値

表 1-4 フラッシュ・ステータス・レジスタ (FSTATUS) のフィールドの説明

ビット	フィールド	値	説明 (1) (2)
15-9	予約		予約
8	3VSTAT	0 1	フラッシュ電圧 (V_{DD3VFL}) ステータス・ラッチ・ビット。セットされた場合、このビットはポンプ・モジュールからの 3VSTAT 信号がハイ・レベルに行ったことを示します。この信号は、フラッシュ 3.3V 供給が許容範囲から外れたことを示します。 0 のライトは無視されます。 1 このビットが1のときは、フラッシュ 3.3V 供給が許容範囲から外れたことを示します。1 をライトすることによって、このビットをクリアしてください。
7-4	予約		予約
3	ACTIVEWAITS	0 1	バンクおよびポンプ・スタンバイ・アクティブ・ウェイト・カウンタ・ステータス・ビット。このビットは、それぞれのウェイト・カウンタがアクセスをタイムアウトしているかどうかを示します。 0 カウンタはカウントしていません。 1 カウンタはカウントしています。
2	STDBYWAITS	0 1	バンクおよびポンプ・スリープ・スタンバイ・ウェイト・カウンタ・ステータス・ビット。このビットは、それぞれのウェイト・カウンタがアクセスをタイムアウトしているかどうかを示します。 0 カウンタはカウントしていません。 1 カウンタはカウントしています。
1-0	PWRS	00 01 10 11	電力モード・ステータス・ビット。これらのビットは、フラッシュ /OTP が現在どの電力モードにあるのかを示します。 PWRS ビットは、適切なタイミング遅延の終了後にのみ新しい電力モードに設定されます。 00 スリープ・モードのポンプおよびバンク (最小電力) 01 スタンバイ・モードのポンプおよびバンク 10 予約 11 ポンプおよびバンクはアクティブでリード・モード (最大電力)

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

図 1-7 フラッシュ・スタンバイ・ウェイト・レジスタ (FSTDBYWAIT)

15	9	8	0
予約		STDBYWAIT	
R-0		R/W-1	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 1-5 フラッシュ・スタンバイ・ウェイト・レジスタ (FSTDBYWAIT) のフィールドの説明

ビット	フィールド	値	説明 (1) (2)
15-9	予約	0	予約
8-0	STDBYWAIT	11111111	このレジスタは、デフォルト状態のままにしておく必要があります。 バンクおよびポンプ・スリープ・スタンバイ・ウェイト・カウント 511 SYSCLKOUT サイクル (デフォルト)

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

図 1-8 フラッシュ・スタンバイ・アクティブ・ウェイト・カウンタ・レジスタ (FACTIVEWAIT)

7	9	8	0
予約		ACTIVEWAIT	
R-0		R/W-1	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 1-6 フラッシュ・スタンバイ・アクティブ・ウェイト・カウンタ・レジスタ (FACTIVEWAIT) のフィールドの説明

ビット	フィールド	値	説明 (1) (2)
15-9	予約	0	予約
8-0	ACTIVEWAIT	11111111	このレジスタは、デフォルト状態のままにしておく必要があります。 バンクおよびポンプ・スタンバイ・アクティブ・ウェイト・カウント： 511 SYSCLKOUT サイクル (デフォルト)

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

図 1-9 フラッシュ・ウェイトステート・レジスタ (FBANKWAIT)

15	12	11	8	7	4	3	0
予約		PAGEWAIT			予約		RANDWAIT
R-0		R/W-1			R-0		R/W-1

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 1-7 フラッシュ・ウェイトステート・レジスタ (FBANKWAIT) のフィールドの説明

ビット	フィールド	値	説明 (1)(2)(3)
15-12	予約		予約
11-8	PAGEWAIT	0000 0001 0010 0011 ... 1111	<p>フラッシュ・ページ・リード・ウェイト・ステート。これらのレジスタ・ビットは、ページ・リード動作のウェイト・ステートの数をフラッシュ・バンクに対する CPU クロック・サイクル (0 ~ 15 SYSCLKOUT サイクル) で指定します。詳細については、セクション 1.2.1 を参照してください。</p> <p>PAGED フラッシュまたは ROM アクセスに必要な最小時間については、デバイス別のデータ・マニュアルを参照してください。</p> <p>RANDWAIT は、PAGEWAIT 設定以上の値に設定する必要があります。RANDWAIT より大きい PAGEWAIT 値を検出するためのハードウェアは提供されません。</p> <p>ROM デバイスでは、これらのビットはフラッシュを置き換えた ROM ブロックのウェイト・ステートに影響します。</p> <p>アクセスあたり合計 1 SYSCLKOUT サイクルに対するページ・アクセスごとのゼロ・ウェイト・ステート。パイプライン・モードをイネーブルにした場合、PAGEWAIT は 0 より大きくなければなりません。</p> <p>0001 ページ・フラッシュ・アクセスごとに 1 ウェイト・ステート、またはアクセスごとに合計 2 SYSCLKOUT サイクル。</p> <p>0010 ページ・フラッシュ・アクセスごとに 2 ウェイト・ステート、またはアクセスごとに合計 3 SYSCLKOUT サイクル。</p> <p>0011 ページ・フラッシュ・アクセスごとにウェイト・ステート、またはアクセスごとに合計 4 SYSCLKOUT サイクル。</p> <p>...</p> <p>1111 ページ・フラッシュ・アクセスごとに 15 ウェイト・ステート、またはアクセスごとに合計 16 SYSCLKOUT サイクル。(デフォルト)</p>
7-4	予約		予約
3-0	RANDWAIT	0000 0001 0010 0011 ... 1111	<p>フラッシュ・ランダム・リード・ウェイト・ステート。これらのレジスタ・ビットは、ランダム・リード動作のウェイト・ステートの数をフラッシュ・バンクに対する CPU クロック・サイクル (1 ~ 15 SYSCLKOUT サイクル) で指定します。詳細については、セクション 1.2.1 を参照してください。</p> <p>RANDOM フラッシュまたは ROM アクセスに必要な最小時間については、デバイス別のデータ・マニュアルを参照してください。</p> <p>RANDWAIT は、0 より大きい値に設定されなければなりません。つまり、少なくとも 1 ランダム・ウェイト・ステートを使用する必要があります。また、RANDWAIT は、PAGEWAIT 設定以上の値に設定する必要があります。デバイスは、RANDWAIT より大きい PAGEWAIT 値の検出と修正を行いません。</p> <p>ROM デバイスでは、これらのビットはフラッシュを置き換えた ROM ブロックのウェイト・ステートに影響します。</p> <p>0000 不正な値。RANDWAIT は、0 より大きい値に設定されなければなりません。</p> <p>0001 ランダム・フラッシュ・アクセスごとに 1 ウェイト・ステート、またはアクセスごとに合計 2 SYSCLKOUT サイクル。</p> <p>0010 ランダム・フラッシュ・アクセスごとに 2 ウェイト・ステート、またはアクセスごとに合計 3 SYSCLKOUT サイクル。</p> <p>0011 ランダム・フラッシュ・アクセスごとにウェイト・ステート、またはアクセスごとに合計 4 SYSCLKOUT サイクル。</p> <p>...</p> <p>1111 ランダム・フラッシュ・アクセスごとに 15 ウェイト・ステート、またはアクセスごとに合計 16 SYSCLKOUT サイクル。(デフォルト)</p>

- (1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。
(2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。
(3) このレジスタにライトする場合は、[セクション 1.2.4](#) に記載されている手順に従ってください。

図 1-10 OTP ウェイト・ステート・レジスタ (FOTPWAIT)

15	5	4	0
予約		OTPWAIT	
R-0		R/W-1	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 1-8 OTP ウェイト・ステート・レジスタ (FOTPWAIT) のフィールドの説明

ビット	フィールド	値	説明 ^{(1) (2)(3)}
15-5	予約	0	予約
4-0	OTPWAIT		OTPWAIT OTP リード・ウェイト・ステート。これらのレジスタ・ビットは、リード動作のウェイト・ステートの数を OTP に対する CPU クロック・サイクル (1 ~ 31 SYSCLKOUT サイクル) で指定します。詳細については、「フラッシュ / OTP からの CPU リードまたはフェッチ・アクセス」のセクションを参照してください。OTP には PAGE モードはありません。 OTPWAIT は、0 より大きい値に設定されなければなりません。つまり、最低でも 1 ウェイト・ステートを使用する必要があります。OTP または ROM アクセスに必要な最小時間については、デバイス別のデータ・マニュアルを参照してください。 ROM デバイスでは、これらのビットは OTP を置き換えた ROM ブロックのウェイト・ステートに影響します。 00000 不正な値。OTPWAIT は、1 以上の値に設定されなければなりません。 00001 アクセスあたり合計 2 SYSCLKOUT サイクルのために、1 ウェイト・ステートが各 OTP アクセスに使用されます。 00010 アクセスあたり合計 3 SYSCLKOUT サイクルのために、2 ウェイト・ステートが各 OTP アクセスに使用されます。 00011 アクセスあたり合計 4 SYSCLKOUT サイクルのために、3 ウェイト・ステートが各 OTP アクセスに使用されます。 ... 11111 アクセスあたり合計 32 SYSCLKOUT サイクルのために、31 ウェイト・ステートが OTP アクセスに使用されます。

 (1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

 (2) このレジスタはコード・セキュリティ・モジュール (CSM) によって保護されます。詳細については、[第 2 章](#) を参照してください。

 (3) このレジスタにライトする場合は、[セクション 1.2.4](#) に記載されている手順に従ってください。

コード・セキュリティ・モジュール (CSM)

コード・セキュリティ・モジュール (CSM) は、28x デバイ스에組み込まれているセキュリティ機能です。このモジュールは、無許可の人間に対してオンチップ・メモリへのアクセス / 可視性を防止します。つまり、保護されているコードの複製 / リバース・エンジニアリングを防止します。

セキュアという単語は、オンチップ・メモリへのアクセスが保護されることを意味します。アンセキュアという単語は、オンチップ・セキュア・メモリへのアクセスが保護されないことを意味します。つまり、メモリの内容は、任意の手段 (たとえば、Code Composer Studio™ などのデバッグ・ツール) によってリードすることができます。

項目	ページ
2.1 機能説明	32
2.2 他のオンチップ・リソースに対する CSM の影響.....	34
2.3 ユーザー・アプリケーションへのコード・セキュリティの組み込み.....	35
2.4 セキュリティ・ロジックを保護するために行うこと / 行わないこと.....	40
2.5 CSM 機能 - 要約	40

2.1 機能説明

セキュリティ・モジュールは、CPU アクセスを特定のオンチップ・メモリに制限します。これは、実際には、JTAG ポートまたは外部ペリフェラルを通じて各種メモリへのリードおよびライト・アクセスをブロックします。セキュリティは、オンチップ・メモリのアクセスに関して定義され、保護されているユーザーのコードまたはデータの無許可のコピーを防止します。

オンチップ・セキュア・メモリ・ロケーションへの CPU アクセスが制限されていると、デバイスはセキュアです。セキュアの場合は、プログラム・カウンタが現在どこをポイントしているかに応じて 2 レベルの保護が可能です。コードがセキュア・メモリ内から実行されている場合は、JTAG を通じたアクセスだけが（つまり、エミュレータを通じて）ブロックされます。これは、セキュア・コードがセキュア・データにアクセスすることを可能にします。逆に、コードがノンセキュア・メモリから実行されている場合は、セキュア・メモリへのアクセスはすべてブロックされます。ユーザー・コードは、セキュア・メモリの中および外に動的にジャンプできます。そのため、ノンセキュア・メモリからのセキュア関数コールが可能になります。同様に、割り込みサービス・ルーチンは、たとえメイン・プログラム・ループがノンセキュア・メモリから実行されても、セキュア・メモリ内に配置できます。

セキュリティは、デバイスをセキュアまたはアンセキュアにするために使用される 128 ビットのデータ（8 つの 16 ビット・ワード）のパスワードによって保護されます。このパスワードは、パスワード・ロケーションと呼ばれる 8 ワードでフラッシュまたは ROM の最後に格納されます。

デバイスは、[セクション 2.3.2](#) に記載されているパスワード・マッチ・フロー（PMF）の実行によってアンセキュアになります。[表 2-1](#) はセキュリティのレベルを示します。

表 2-1. セキュリティ・レベル

正しいパスワードで実行された PMF?	動作モード	プログラム・フェッチ・ロケーション	セキュリティの説明
いいえ	セキュア	セキュア・メモリの外部	フェッチだけがセキュア・メモリに対して許可されます。
いいえ	セキュア	セキュア・メモリの内部	CPU にはフル・アクセスがあります。JTAG ポートはセキュア・メモリの内容をリードできません。
はい	非セキュア	任意の場所	セキュア・メモリに対する CPU および JTAG ポートのフル・アクセス

パスワードは、フラッシュ /ROM メモリ (0x003F 7FF8 ~ 0x003F 7FFF) のコード・セキュリティ・パスワード・ロケーション (PWL) に格納されます。これらのロケーションは、システム設計者によって事前に決定されたパスワードを格納します。

パスワード・ロケーションのすべての 128 ビットが 1 の場合、デバイスにはアンセキュアのラベルが付けられます。新規フラッシュ・デバイスはフラッシュをイレーズしてあるため（すべて 1）、そのデバイスをアンセキュア・モードにするにはパスワード・ロケーションのリードだけが要求されます。パスワード・ロケーションのすべての 128 ビットがゼロの場合は、KEY レジスタの内容にかかわらずデバイスはセキュアです。すべてゼロをパスワードとして使用したり、フラッシュのイレーズ中にデバイスをリセットしたりしないでください。イレーズ・ルーチン中にデバイスをリセットすると、すべてゼロまたは未知のパスワードになる可能性があります。パスワード・ロケーションがすべてゼロのときにデバイスがリセットされた場合は、[セクション 2.3.2](#) に記載されているパスワード・マッチ・フローによってデバイスをアンロックすることはできません。すべてゼロのパスワードの使用は、セキュア・コードのデバッグやフラッシュの再プログラムを行うことを制限します。

注： パスワード・ロケーションがすべてゼロまたは未知の値のときにデバイスがリセットされた場合は、セキュア SARAM からフラッシュ・イレーズ・ルーチンを実行する方法がフラッシュまたは OTP に組み込まれていない限り、デバイスは永続的にロックされます。この手順を実装してセキュリティ・ホールの導入を回避する場合は、注意を払う必要があります。

デバイスをアンセキュアにするために使用されるユーザー・アクセス可能レジスタ（8つの16ビット・ワード）は、キー・レジスタと呼ばれます。これらのレジスタは、アドレス 0x0000 0AE0 ~ 0x0000 0AE7 のメモリ空間でマップされ、EALLOW 保護されます。

注：コード・セキュリティ使用時の予約フラッシュ・ロケーション

コード・セキュリティ動作について、0x3F7F80 から 0x3F7FF5 のアドレスの間はプログラム・コードまたはデータとして使用できませんが、コード・セキュリティ・パスワードをプログラムするときには0x0000にプログラムする必要があります。セキュリティがそれほど重大でない場合は、アドレス 0x3F7F80 ~ 0x3F7FEF をコードまたはデータに対して使用できます。128 ビット・パスワード（0x3F 7FF8 ~ 0x3F 7FFF）は、ゼロにプログラムしないでください。ゼロにプログラムすると、デバイスが永続的にロックされます。

アドレス 0x3F7FF0 ~ 0x3F7FF5 は、データ変数用に予約されており、プログラム・コードが除外されている必要があります。

注：コード・セキュリティ・モジュールの免責事項

このデバイスに含まれているコード・セキュリティ・モジュール（CSM）は、関連メモリ（ROM またはフラッシュ）に格納されたデータをパスワード保護するように設計されており、このデバイスに適用できる保証期間について TI の発行済み仕様に準拠するために標準契約条件に従ってテキサス・インスツルメンツ（TI）によって保証されます。

ただし、TI は、CSM が損なわれる可能性や侵害される可能性がないこと、あるいは関連メモリに格納されたデータには他の手段でアクセスできないことを保証または表明しません。また、上記の場合を除き、TI は、特定の目的に対する商品性または適合性の黙示的保証を含め、このデバイスの CSM または動作に関する保証や表明を行いません。

どのような原因であろうとも、CSM またはこのデバイスの使用から何らかの方法で発生する必然的、例外的、間接的、偶発的、または懲罰的な損害について、そのような損害の可能性を TI が忠告したかどうかに関係なく、いかなる場合でも TI は責任を負いません。除外される損害にはデータの損失、信用の損失、使用の損失、業務の中断、または他の経済的損失が含まれますが、これらに限定されるわけではありません。

2.2 他のオンチップ・リソースに対する CSM の影響

CSM は、表 2-2 にリストされたオンチップ・リソースへのアクセスに影響を及ぼします。

表 2-2 CSM の影響を受ける 280x リソース

アドレス	ブロック
0x0000 0A80-0x0000 0A87	フラッシュ・コンフィギュレーション・レジスタ
0x0000 8000-0x0000 8FFF	L0 SARAM (4K × 16)
0x0000 9000-0x0000 9FFF	L1 SARAM (4K × 16)
0x003D 7800-0x003D 7BFF	ワンタイム・プログラマブル (OTP) または ROM (1K × 16)
0x003E 8000-0x003F 7FFF	フラッシュまたは ROM (64K × 16、32 × 16、または 16 × 16)
0x003F 8000-0x003F 8FFF	L0 SARAM (4K × 16)、ミラー
0x003F 9000-0x003F 9FFF	L1 SARAM (4K × 16)、ミラー

コード・セキュリティ・モジュールは、以下のオンチップ・リソースには何の影響も及ぼしません。

- セキュアとして指定されていないシングルアクセス RAM (SARAM) ブロック - デバイスがセキュア・モードかアンセキュア・モードかに関係なく、これらのメモリ・ブロックは自由にアクセス可能であり、コードはこれらのメモリ・ブロックから実行されます。
- ブート ROM の内容 - ブート ROM の内容に対する可視性は、CSM の影響を受けません。
- オンチップ・ペリフェラル・レジスタ - ペリフェラル・レジスタは、デバイスがセキュア・モードかアンセキュア・モードかに関係なく、オンチップまたはオフチップ・メモリから実行しているコードによって初期化することができます。
- PIE ベクタ・テーブル - ベクタ・テーブルは、デバイスがセキュア・モードかアンセキュア・モードかに関係なくリードおよびライトすることができます。表 2-2 と表 2-3 は、どのオンチップ・リソースが 280x デバイスの CSM の影響を受けるか (または影響を受けないか) を示します。他のデバイスについては、デバイス別のデータ・シートを参照してください。

表 2-3 CSM の影響を受けない 280x リソース

アドレス	ブロック
0x0000 0000-0x0000 03FF	M0 SARAM (1K × 16)
0x0000 0400-0x0000 07FF	M1 SARAM (1K × 16)
0x0000 0800-0x0000 0CFF	ペリフェラル・フレーム 0 (2K × 16)
0x0000 0D00-0x0000 0FFF	PIE ベクタ RAM (256 × 16)
0x0000 6000-0x0000 6FFF	ペリフェラル・フレーム 1 (4K × 16)
0x0000 A000-0x0000 BFFF	H0 SARAM (8K × 16)
0x0000 7000-0x0000 7FFF	ペリフェラル・フレーム 2 (4K × 16)
0x003F A000-0x003F BFFF	H0 SARAM (8K × 16) ミラー
0x003F F000-0x003F FFFF	ブート ROM (4K × 16)

要約すると、表 2-3 に示した保護されていないオンチップ・プログラム SARAM には、コード・セキュリティ・モジュールからの影響なしに JTAG コネクタを介してコードをロードすることができます。デバイスがセキュア・モードかアンセキュア・モードかに関係なく、コードをデバッグし、ペリフェラル・レジスタを初期化できます。

2.3 ユーザー・アプリケーションへのコード・セキュリティの組み込み

一般に、コード・セキュリティは、プロジェクトの開発フェーズでは必要ありません。ただし、いったんコードが開発されたら、セキュリティが必要になります。そのようなコードをフラッシュ・メモリでプログラムする（または ROM 化）前に、パスワードを選択してデバイスをセキュアにする必要があります。パスワードをプログラムすると、デバイスはセキュアになります（つまり、適切なロケーションでのパスワードのプログラミングと、デバイス・リセットの実行または FORCESEC ビット (CSMSCR.15) のセットは、デバイスをセキュアにします）。その時点以降は、任意の手段によって（JTAG や、外部 / オンチップ・メモリから実行されているコードなどによって）セキュア・メモリの内容をデバッグするためのアクセスは、正しいパスワードを入力する必要があります。パスワードは、一般的なエンドカスタマ用途などでセキュア・メモリの外部のコードを実行する場合は必要ありません。ただし、デバッグ目的でセキュア・メモリの内容にアクセスするには、パスワードが必要です。

表 2-4 コード・セキュリティ・モジュール (CSM) レジスタ

メモリ・アドレス	レジスタ名	リセット値	レジスタの説明
キー・レジスタ			
0x0000-0AE0	KEY0 ⁽¹⁾	0xFFFF	128 ビット・キー・レジスタの下位ワード
0x0000-0AE1	KEY1 ⁽¹⁾	0xFFFF	128 ビット・キー・レジスタの 2 番目のワード
0x0000-0AE2	KEY2 ⁽¹⁾	0xFFFF	128 ビット・キー・レジスタの 3 番目のワード
0x0000-0AE3	KEY3 ⁽¹⁾	0xFFFF	128 ビット・キーの 4 番目のワード
0x0000-0AE4	KEY4 ⁽¹⁾	0xFFFF	128 ビット・キーの 5 番目のワード
0x0000-0AE5	KEY5 ⁽¹⁾	0xFFFF	128 ビット・キーの 6 番目のワード
0x0000-0AE6	KEY6 ⁽¹⁾	0xFFFF	128 ビット・キーの 7 番目のワード
0x0000-0AE7	KEY7 ⁽¹⁾	0xFFFF	128 ビット・キー・レジスタの上位ワード
0x0000-0AEF	CSMSCR ⁽¹⁾	0x005F	CSM ステータスおよび制御レジスタ
フラッシュ・メモリ内のパスワード・ロケーション (PWL) - CSM パスワードのためだけに予約済み			
0x003F-7FF8	PWL0	User defined	128 ビット・パスワードの下位ワード
0x003F-7FF9	PWL1	User defined	128 ビット・パスワードの 2 番目のワード
0x003F-7FFA	PWL2	User defined	128 ビット・パスワードの 3 番目のワード
0x003F-7FFB	PWL3	User defined	128 ビット・パスワードの 4 番目のワード
0x003F-7FFC	PWL4	User defined	128 ビット・パスワードの 5 番目のワード
0x003F-7FFD	PWL5	User defined	128 ビット・パスワードの 6 番目のワード
0x003F-7FFE	PWL6	User defined	128 ビット・パスワードの 7 番目のワード
0x003F-7FFF	PWL7	User defined	128 ビット・パスワードの上位ワード

(1) これらのレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

図 2-1 CSM ステータスおよび制御レジスタ (CSMSCR)

15	14	7	6	1	0
FORCESEC	予約	予約	予約	予約	SECURE
R/W-1	R-0	R-10111			R-1

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 2-5 CSM ステータスおよび制御レジスタ (CSMSCR) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15	FORCESEC	0 1	1のライトは、キー・レジスタをクリアし、デバイスをセキュアにします。 リードは常にゼロを返します。 キー・レジスタをクリアし、デバイスをセキュアにします。デバイスを再びアンセキュアにするには、 セクション 2.3.2 に記載されているパスワード・マッチ・フローに従う必要があります。
14-1	予約		予約
0	SECURE	0 1	デバイスのセキュリティ状態を示すリード専用ビット。 0 デバイスはアンセキュアです (CSM アンロック済み)。 1 デバイスはセキュアです (CSM ロック済み)。

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#)を参照してください。

2.3.1 セキュリティのアンロックを必要とする環境

以下は、アンセキュアにすることが要求される可能性のある一般的な状況です。

- デバッガ (Code Composer Studio™ など) を使用したコード開発。
これは、製品の設計段階の間の最も一般的な環境です。
- TI のフラッシュ・ユーティリティ (Code Composer Studio™ F28xx オンチップ・フラッシュ・プログラマ・プラグインなど) を使用したフラッシュ・プログラミング。
コード開発およびテストの間はフラッシュ・プログラミングは一般的です。ユーザーが必要なパスワードを供給すると、フラッシュ・ユーティリティはフラッシュのプログラミングを試みる前にセキュリティ・ロジックをディスエーブルにします。新規デバイスはフラッシュがイレーズされているため、フラッシュ・ユーティリティは許可なしに新規デバイスのコード・セキュリティ・ロジックをディスエーブルにすることができます。ただし、すでにカスタム・パスワードが含まれているデバイスの再プログラミングは、デバイスをアンロックしプログラミングをイネーブルにするために、フラッシュ・ユーティリティに供給されるパスワードを必要とします。TI によって供給されたフラッシュ API を使用するカスタム・プログラミング・ソリューションでは、フラッシュ・プログラミング・アルゴリズムをセキュア・メモリから実行することによって、CSM のアンロックを回避できます。
- アプリケーションによって定義されたカスタム環境

上記に加えて、以下のような状況でもセキュア・メモリ内容へのアクセスが必要になることがあります。

- コードまたはデータをセキュア SARAM にロードするか、またはフラッシュをイレーズ/プログラムするためのオンチップ・ブートローダの使用。
- オンチップ・アンセキュア・メモリからのコードの実行で、ルックアップ・テーブル用のセキュア・メモリに対するアクセスの要求。外部コードからのパスワードの供給はコード・セキュリティを損なう可能性があるため、これは推奨される動作状態ではありません。

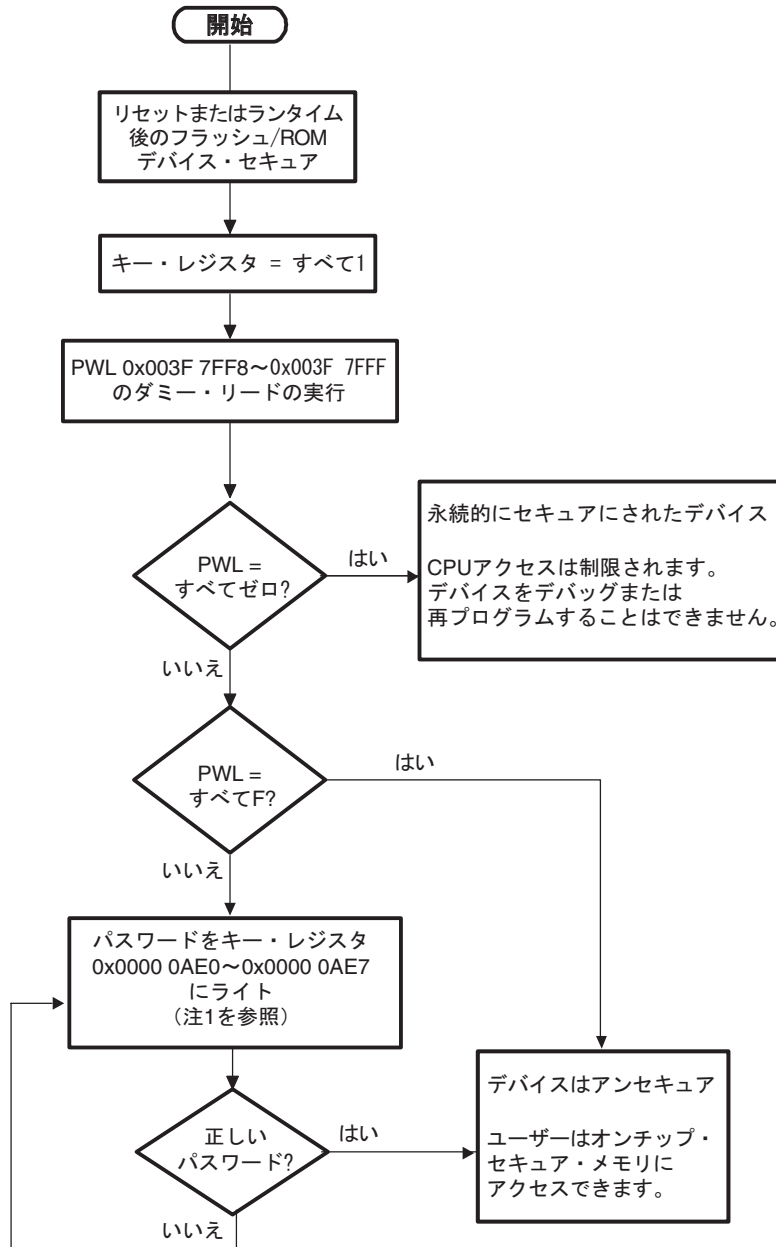
アンセキュア化のシーケンスは、上記のすべての状況において同一です。このシーケンスは、簡潔さのためにパスワード・マッチ・フロー (PMF) と呼ばれます。[図 2-2](#)では、ユーザーがデバイスをアンセキュアにしようとするたびに要求される動作のシーケンスを説明します。コード例は、説明をより分かりやすくするために記載されています。

2.3.2 パスワード・マッチ・フロー

パスワード・マッチ・フロー (PMF) は、基本的には、パスワード・ロケーション (PWL) からの 8 つのダミー・リードの後にキー・レジスタへの 8 つのライトが続くシーケンスです。

図 2-2 は、PMF がどのようにセキュリティ・ロジック・レジスタの初期化とセキュリティ・ロジックのディスエーブル化に役立つのかを示しています。

図 2-2 パスワード・マッチ・フロー (PMF)



A キー・レジスタは EALLOW 保護されます。

2.3.3 コード・セキュリティあり/なしのデバイスに対するアンセキュア化の考慮事項

ケース 1 とケース 2 は、コード・セキュリティしているデバイスとコード・セキュリティしていないデバイスに対するアンセキュア化の考慮事項を提供します。

ケース 1: コード・セキュリティしているデバイス

コード・セキュリティのあるデバイスでは、事前に決定されたパスワードがパスワード・ロケーション (メモリの 0x3F7FF8 ~ 0x3F7FFF) に格納されている必要があります。また、ロケーション 0x3F7F80 ~ 0x3F7FF5 は、すべて 0x0000 でプログラムされ、プログラムやデータ・ストレージに対して使用してはいけません。以下は、このデバイスをアンセキュアにする手順です。

1. パスワード・ロケーションのダミー・リードを実行します。
2. パスワードをキー・レジスタ (メモリ内のロケーション 0x0000 0AE0 ~ 0x0000 0AE7) にライトします。
3. パスワードが正しい場合は、デバイスがアンセキュアになります。それ以外の場合は、デバイスはセキュアのままになります。

ケース 2: コード・セキュリティしていないデバイス

コード・セキュリティのないデバイスでは、0x FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF (すべて 1 の 128 ビット) がパスワード・ロケーションに格納されている必要があります。以下は、このデバイスを使用する手順です。

1. リセット時に、CSM は、CSM によって保護されているメモリ領域をロックします。
2. パスワード・ロケーションのダミー・リードを実行します。
3. パスワードはすべて 1 であるため、これだけですべてのメモリ領域がアンロックされます。この操作が完了するとすぐに、セキュア・メモリは完全にアクセス可能になります。

注: たとえデバイスがパスワードで保護されていなくても (どのパスワード・ロケーションすべて 1)、CSM はリセット時にロックします。したがって、アクセスを実行するコードが CSM 保護メモリ領域の外部から実行されている場合は、セキュア・メモリのリード、ライト、またはプログラミングの前にダミー・リード動作がこれらのデバイスで引き続き実行されなければなりません。ブート ROM コードは、利便性のために、このダミー・リードを行います。

2.3.3.1 アンセキュアにする C コード例

```

volatile int *CSM = (volatile int *)0x000AE0;    //CSM register file
volatile int *PWL = (volatile int *)0x3F7FF8;    //Password location
volatile int tmp;

int i ;

// Read the 128bits of the password location (PWL)
// in Flash/ROM at address 0x3F7FF80x3F7FFF.
// If the device is secure, then the values read will
// not actually be loaded into the temp variable, so
// this is called a dummy read.
for (i = 0; i<8; i++) tmp = *PWL++;

// If the password location (PWL) is all = ones (0xFFFF),
// then the device will now be unsecure.  If the password
// is not all ones (0xFFFF), then the code below is required
// to unsecure the CSM.
// Write the 128bit password to the KEY registers
// If this password matches that stored in the
// PWL then the CSM will become unsecure.  If it does not
// match, then the device will remain secure.
// An example password of:
// 0x0123456789ABCDEF89AB45670123 is used.
asm("  EALLOW");    // Key registers are EALLOW protected
*CSM++ = 0x0123;    // Register KEY0 at 0xAE0
*CSM++ = 0x4567;    // Register KEY1 at 0xAE1
*CSM++ = 0x89AB;    // Register KEY2 at 0xAE2
*CSM++ = 0xCDEF;    // Register KEY3 at 0xAE3
*CSM++ = 0xCDEF;    // Register KEY4 at 0xAE4
*CSM++ = 0x89AB;    // Register KEY5 at 0xAE5
*CSM++ = 0x4567;    // Register KEY6 at 0xAE6
*CSM++ = 0x0123;    // Register KEY7 at 0xAE7
asm("  EDIS");

```

2.3.3.2 再セキュアにする C コード例

```

volatile int *CSMSCR = 0x000AFF;    //CSM register file
                                        //Set FORCESEC bit
asm("  EALLOW");    //CSMSCR register is EALLOW protected.
*CSMSCR = 0x8000;
asm ("EDIS");

```

2.4 セキュリティ・ロジックを保護するための必要事項 / 禁止事項

2.4.1 必要事項

- デバッグおよびコードの開発を簡単にするために、アンセキュア・モードでデバイスを使用します。つまり、パスワード・ロケーションでは 128 ビットのすべてを 1 として使用します（または、覚えやすいパスワードを使用します）。コードが確定した場合は、開発フェーズ後にパスワードを使用します。
- フラッシュ・ユーティリティを使用して COFF ファイルをプログラムする前に、パスワード・ロケーションに格納されたパスワードを再チェックします。
- コード実行のフローは、セキュリティを損なうことなくセキュア・メモリとアンセキュア・メモリの間で前後に自由に切り換えることができます。デバイスがセキュアにされている場合にセキュア・メモリにあるデータ変数にアクセスするには、コードがセキュア・メモリから現在実行されている必要があります。
- CSM を使用する場合は、ロケーション 0x3F7F80 ~ 0x3F7FF5 を 0x0000 でプログラムします。

2.4.2 禁止事項

- コード・セキュリティが要求される場合は、パスワード・ロケーション以外の場所でアプリケーションにパスワードを組み込まないでください。そうしないと、セキュリティが損なわれる可能性があります。
- すべてゼロの 128 ビットをパスワードとして使用しないでください。これは、キー・レジスタの内容にかかわらず自動的にデバイスをセキュアにします。デバイスはデバッグや、再プログラムができなくなります。
- フラッシュ・アレイに対するイレーズ動作中は、リセットしないでください。これは、ゼロまたは未知の値をパスワード・ロケーションに残すことがあります。リセット時にパスワード・ロケーションがすべてゼロの場合は、キー・レジスタの内容にかかわらずデバイスは常にセキュアになります。
- ロケーション 0x3F7F80 ~ 0x3F7FF5 を使用してプログラムやデータを格納しないでください。CSM を使用する場合は、これらのロケーションを 0x0000 にプログラムする必要があります。

2.5 CSM 機能 - 要約

1. [セクション 2.3.2](#)に記載されているパスワード・マッチ・フローが実行されるまでは、フラッシュはリセット後にセキュアにされます。
2. フラッシュまたは ROM の外部でコードを実行する標準的な方法は、フラッシュにコードをプログラムし（ROM デバイスの場合、プログラムはデバイス製造時にハードコードされる）、DSP の電源を投入することです。CSM の状態にかかわらず、命令フェッチは常にセキュア・メモリから許可されるため、パスワード・マッチ・フローを実行しなくてもコードは正しく機能します。
3. デバイスがセキュアの場合、アンセキュア・メモリからのコード実行によってセキュア・メモリを変更することはできません。
4. デバイスがセキュアの場合、アンセキュア・メモリからの任意のコード実行からセキュア・メモリをリードすることはできません。
5. デバイスがセキュアな場合、常にデバッガ（つまり、Code Composer Studio™）によってリードまたはライトすることはできません。
6. デバイスがアンセキュアの場合 CPU コードとデバッガの両方からのセキュア・メモリに対する完全なアクセスができます。

クロック

このセクションでは、オシレータ、PLL およびクロック・メカニズム、ウォッチドッグ機能、低電力モードについて説明します。

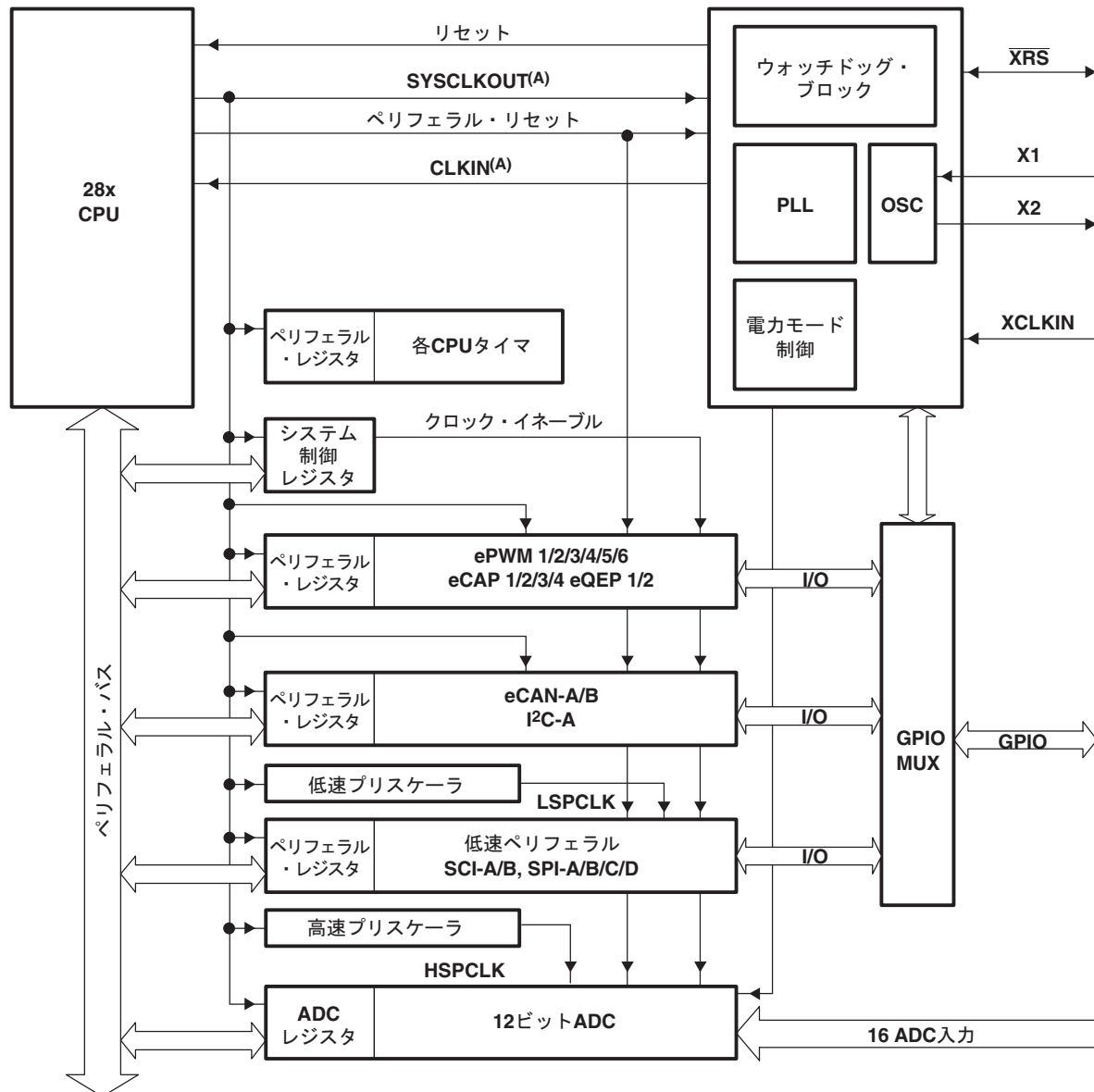
項目	ページ
3.1 クロックとシステム制御.....	42
3.2 OSC および PLL ブロック	47
3.3 低電力モード・ブロック.....	55
3.4 ウォッチドッグ・ブロック.....	59
3.5 32 ビット CPU タイマ 0/1/2.....	64

3.1 クロックとシステム制御

図 3-1 は、280x デバイスにおける各種クロックおよびリセット・ドメインを示します。

PLL、クロック、ウォッチドッグ、および低電力モードは、表 3-1 に記載されているレジスタによって制御されます。

図 3-1 クロックおよびリセット・ドメイン



A CLKIN は CPU へのクロックです。これは、SYSCLKOUT として CPU の外に出力します(つまり、CLKIN は SYSCLKOUT と同じ周波数である)。

表 3-1 PLL、クロック、ウォッチドッグ、および低電力モード・レジスタ

名前	アドレス	サイズ (x16)	説明 ⁽¹⁾	ビットの説明
XCLK	0x7010	1	XCLKOUT、X1、および XCLKIN レジスタ	図 3-12
PLLSTS ⁽²⁾	0x7011	1	PLL ステータス・レジスタ	図 3-11
予約	0x7012 0x7019	8		
HISPCP	0x701A	1	高速ペリフェラル・クロック (HSPCLK) プリスケアラ・レジスタ	図 3-4
LOSPCP	0x701B	1	低速ペリフェラル・クロック (LSPCLK) プリスケアラ・レジスタ	図 3-5
PCLKCR0	0x701C	1	ペリフェラル・クロック制御レジスタ 0	図 3-2
PCLKCR1	0x701D	1	ペリフェラル・クロック制御レジスタ 1	図 3-3
LPMCR0	0x701E	1	低電力モード制御レジスタ 0	図 3-5
予約	0x701F	1		
予約	0x7020	1		
PLLCR ⁽²⁾	0x7021	1	PLL 制御レジスタ	図 3-10
SCSR	0x7022	1	システム制御およびステータス・レジスタ	図 3-15
WDCNTR	0x7023	1	ウォッチドッグ・カウンタ・レジスタ	図 3-16
Reserved	0x7024	1		
WDKEY	0x7025	1	ウォッチドッグ・リセット・キー・レジスタ	図 3-17
予約	0x7026 0x7028	3		
WDCR	0x7029	1	ウォッチドッグ制御レジスタ	図 3-18

- (1) この表にあるレジスタは、すべて EALLOW 保護されます。詳細については、[セクション 5.2](#) を参照してください。
(2) PLL 制御レジスタ (PLLCR) および PLL ステータス・レジスタ (PLLSTS) は、XRS 信号またはウォッチドッグ・リセットによってのみ既知の状態にリセットされます。デバッグまたはミッシング・クロック検出ロジックによって発行されたリセットには何の効果もありません。

PCLKCR0 および PCLKCR1 レジスタは、各種ペリフェラル・モジュールへのクロックをイネーブル / ディスエーブルにします。ペリフェラル-GPIO マルチプレクスのため、すべてのペリフェラルを同時に使用することはできません。すべてのペリフェラルに対してクロックを同時にオンにすることは可能ですが、これを行った場合、消費電流が必要以上に多くなります。これを回避するには、アプリケーションによって要求されるクロックだけをイネーブルにします。

図 3-2 ペリフェラル・クロック制御 0 レジスタ (PCLKCR0)

15	14	13	12	11	10	9	8
ECANBENCLK	ECANAENCLK	予約	SCIBENCLK	SCIAENCLK	SPIBENCLK	SPIAENCLK	
R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
SPIDENCLK	SPICENCLK	予約	I2CAENCLK	ADCENCLK	TBCLKSYNC	予約	
R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

図 3-3 ペリフェラル・クロック制御 1 レジスタ (PCLKCR1)

15	14	13	12	11	10	9	8
EQEP2ENCLK	EQEP1ENCLK	予約	ECAP4ENCLK	ECAP3ENCLK	ECAP2ENCLK	ECAP1ENCLK	
R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
予約	EPWM6ENCLK	EPWM5ENCLK	EPWM4ENCLK	EPWM3ENCLK	EPWM2ENCLK	EPWM1ENCLK	
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-2 ペリフェラル・クロック制御 0 レジスタ (PCLKCR0) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15	ECANBENCLK	0 1	ECAN-B クロック・イネーブル。このビットは、2806、2802、および 2801 では予約されています。 ⁽²⁾ eCAN-B モジュールはクロック ON されません。(デフォルト) ⁽³⁾ eCAN-B モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
14	ECANAENCLK	0 1	ECAN-A クロック・イネーブル。 eCAN-A モジュールはクロック ON されません。(デフォルト) ⁽³⁾ eCAN-A モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
13-12	予約		予約
11	SCIBENCLK	0 1	SCI-B クロック・イネーブル。このビットは、2802 および 2801 では予約されています。 ⁽²⁾ SCI-B モジュールはクロック ON されません。(デフォルト) ⁽³⁾ SCI-B モジュールは、低速クロック (LSPCLK) によってクロック ON されます。
10	SCIAENCLK	0 1	SCI-A クロック・イネーブル。 SCI-A モジュールはクロック ON されません。(デフォルト) ⁽³⁾ SCI-A モジュールは、低速クロック (LSPCLK) によってクロック ON されます。
9	SPIBENCLK	0 1	SPI-B クロック・イネーブル。 SPI-B モジュールはクロック ON されません。(デフォルト) ⁽³⁾ SPI-B モジュールは、低速クロック (LSPCLK) によってクロック ON されます。
8	SPIAENCLK	0 1	SPI-A クロック・イネーブル。 SPI-A モジュールはクロック ON されません。(デフォルト) ⁽³⁾ SPI-A モジュールは、低速クロック (LSPCLK) によってクロック ON されます。
7	SPIDENCLK	0 1	SPI-D クロック・イネーブル。このビットは、2802 および 2801 では予約されています。 ⁽²⁾ SPI-D モジュールはクロック ON されません。(デフォルト) ⁽³⁾ SPI-D モジュールは、低速クロック (LSPCLK) によってクロック ON されます。
6	SPICENCLK	0 1	SPI-C クロック・イネーブル。このビットは、2802 および 2801 では予約されています。 ⁽²⁾ SPI-C モジュールはクロック ON されません。(デフォルト) ⁽³⁾ SPI-C モジュールは、低速クロック (LSPCLK) によってクロック ON されます。
5	予約	0	予約
4	I ² CAENCLK	0 1	I ² C クロック・イネーブル。 I ² C モジュールはクロック ON されません。(デフォルト) ⁽³⁾ I ² C モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
3	ADCENCLK	0 1	ADC クロック・イネーブル。 ADC はクロック ON されません。(デフォルト) ⁽³⁾ ADC モジュールは、高速クロック (HSPCLK) によってクロック ON されます。
2	TBCLKSYNC	0 1	ePWM モジュール基準時間クロック (TBCLK) Sync: イネーブルにされたすべての ePWM モジュールを基準時間クロック (TBCLK) にグローバルに同期させることを許可する: イネーブルにされた各 ePWM モジュール内の TBCLK (基準時間クロック) は停止されます。(デフォルト)。ただし、ePWM クロック・イネーブル・ビットが PCLKCR1 レジスタでセットされた場合は、たとえ TBCLKSYNC が 0 であっても、ePWM モジュールは引き続き SYSCLKOUT によってクロック ON されます。 イネーブルにされた各 ePWM モジュール・クロックは、アラインされた TBCLK の最初の立ち上がりエッジで開始されます。完全に同期化された TBCLK は、各 ePWM モジュールの TBCTL レジスタのプリスケアラ・ビットを同様にセットする必要があります。ePWM クロックをイネーブルにする正しい手順は以下のとおりです。 1. PCLKCR1 レジスタで ePWM モジュール・クロックをイネーブルにします。 2. TBCLKSYNC を 0 に設定します。 3. プリスケアラ値と ePWM モードを設定します。 4. TBCLKSYNC を 1 に設定します。
1	予約		予約

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) 特定のペリフェラルがないデバイスでは、クロック選択ビットが予約されます。これらのデバイスでは、ビットに 1 をライトしないでください。

(3) ペリフェラル・ブロックが使用されていない場合は、そのペリフェラルへのクロックをオフにして、消費電力を最小限にすることができます。

表 3-3 ペリフェラル・クロック制御 1 レジスタ (PCLKCR1) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15	EQEP2ENCLK	0	eQEP2 クロック・イネーブル。このビットは、2802 および 2801 では予約されています。 ⁽²⁾ eQEP2 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	eQEP2 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
14	EQEP1ENCLK	0	eQEP1 クロック・イネーブル。 eQEP1 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	eQEP1 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
13-12	予約		予約
11	ECAP4ENCLK	0	eCAP4 クロック・イネーブル。このビットは、2802 および 2801 では予約されています。 ⁽²⁾ eCAP4 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	eCAP4 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
10	ECAP3ENCLK	0	eCAP3 クロック・イネーブル。このビットは、2802 および 2801 では予約されています。 ⁽²⁾ eCAP3 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	eCAP3 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
9	ECAP2ENCLK	0	eCAP2 クロック・イネーブル。 eCAP2 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	eCAP2 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
8	ECAP1ENCLK	0	eCAP1 クロック・イネーブル。 eCAP1 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	eCAP1 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
7-6	予約		予約
5	EPWM6ENCLK	0	ePWM6 クロック・イネーブル。 ⁽⁴⁾ このビットは、2802 および 2801 では予約されています。 ⁽²⁾ ePWM6 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	ePWM6 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
4	EPWM5ENCLK	0	ePWM5 クロック・イネーブル。 ⁽⁴⁾ このビットは、2802 および 2801 では予約されています。 ⁽²⁾ ePWM5 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	ePWM5 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
3	EPWM4ENCLK	0	ePWM4 クロック・イネーブル。 ⁽⁴⁾ このビットは、2802 および 2801 では予約されています。 ⁽²⁾ ePWM4 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	ePWM4 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
2	EPWM3ENCLK	0	ePWM3 クロック・イネーブル。 ⁽⁴⁾ ePWM3 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	ePWM3 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
1	EPWM2ENCLK	0	ePWM2 クロック・イネーブル。 ⁽⁴⁾ ePWM2 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	ePWM2 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。
0	EPWM1ENCLK	0	ePWM1 クロック・イネーブル。 ⁽⁴⁾ ePWM1 モジュールはクロック ON されません。(デフォルト) ⁽³⁾
		1	ePWM1 モジュールは、システム・クロック (SYSCLKOUT) によってクロック ON されます。

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) 特定のペリフェラルがないデバイスでは、クロック選択ビットが予約されます。これらのデバイスでは、ビットに 1 をライトしないでください。

(3) ペリフェラル・ブロックが使用されていない場合は、そのペリフェラルへのクロックをオフにして、消費電力を最小限にすることができます。

(4) ePWM モジュール内の ePWM 基準時間クロック (TBCLK) を開始するには、PCLKCR0 の TBCLKSYNC ビットもセットする必要があります。

高速ペリフェラルおよび低速ペリフェラル・クロック・プリスケール (HISPCP および LOSPCP) レジスタは、それぞれのペリフェラル・クロックを設定するために使用されます。HISPCP ビット・レイアウトについては図 3-4 を、LOSPCP レイアウトについては図 3-5 を参照してください。

図 3-4 高速ペリフェラル・クロック・プリスケラ (HISPCP) レジスタ

15	予約	3	2	0
R-0			HSPCLK	
			R-0 R/W-001	

凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-4 高速ペリフェラル・クロック・プリスケラ (HISPCP) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15-3	予約		予約
2-0	HSPCLK		これらのビットは、SYSCLKOUT を基準として高速ペリフェラル・クロック (HSPCLK) レートを設定します。 HISPCP ⁽²⁾ ≠0 の場合、HSPCLK=SYSCLKOUT/(HISPCP × 2) HISPCP=0 の場合、HSPCLK=SYSCLKOUT
		000	高速クロック =SYSCLKOUT/1
		001	高速クロック =SYSCLKOUT/2 (リセット時のデフォルト)
		010	高速クロック =SYSCLKOUT/4
		011	高速クロック =SYSCLKOUT/6
		100	高速クロック =SYSCLKOUT/8
		101	高速クロック =SYSCLKOUT/10
		110	高速クロック =SYSCLKOUT/12
		111	高速クロック =SYSCLKOUT/14

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) この等式の HISPCP は、HISPCP レジスタにおけるビット 2:0 の値を示します。

図 3-5 低速ペリフェラル・クロック・プリスケラ・レジスタ (LOSPCP)

15	予約	3	2	0
R-0			LSPCLK	
			R/W-010	

凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-5 低速ペリフェラル・クロック・プリスケラ・レジスタ (LOSPCP) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15-3	予約		予約
2-0	LSPCLK		これらのビットは、SYSCLKOUT を基準として低速ペリフェラル・クロック (LSPCLK) レートを設定します。 LOSPCP ⁽²⁾ ≠0 の場合、LSPCLK=SYSCLKOUT/(LOSPCP × 2) LOSPCP=0 の場合、LSPCLK=SYSCLKOUT
		000	低速クロック =SYSCLKOUT/1
		001	低速クロック =SYSCLKOUT/2
		010	低速クロック =SYSCLKOUT/4 (リセット時のデフォルト)
		011	低速クロック =SYSCLKOUT/6
		100	低速クロック =SYSCLKOUT/8
		101	低速クロック =SYSCLKOUT/10
		110	低速クロック =SYSCLKOUT/12
		111	低速クロック =SYSCLKOUT/14

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

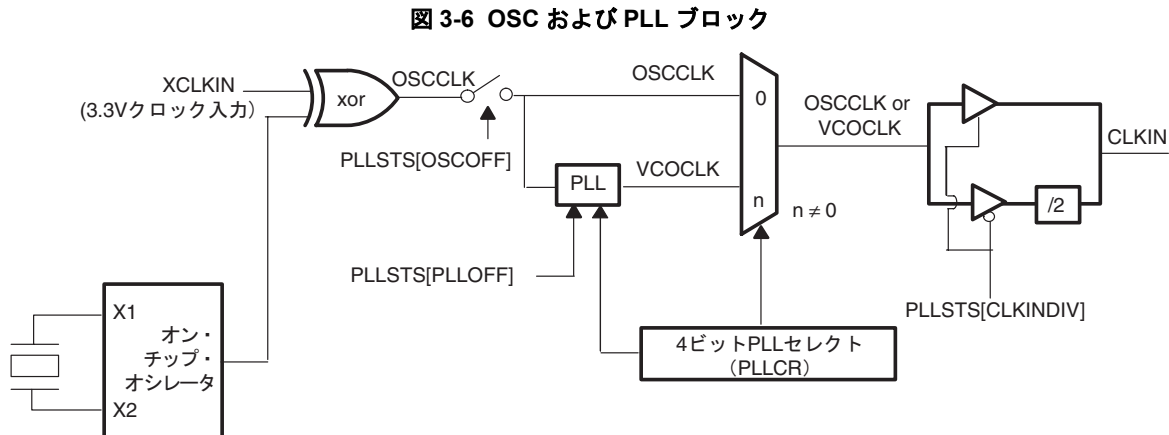
(2) この等式の LOSPCP は、LOSPCP レジスタにおけるビット 2:0 の値を示します。

3.2 OSC および PLL ブロック

オンチップ・オシレータおよびフェーズロック・ループ (PLL) ブロックは、低電力モード (LPM) エントリの制御のほかに、デバイスのクロックを提供します。

3.2.1 PLL ベースのクロック・モジュール

280x デバイスには、オンチップの PLL ベースのクロック・モジュールがあります。PLL には、さまざまな CPU クロック・レートを選択するための 4 ビット通倍率があります。図 3-6 は、280x の OSC および PLL ブロックを示します。



PLL ベースのクロック・モジュールは、以下の 2 つの動作モードを提供します。

- **水晶 / 発振子動作 :**

オンチップ・オシレータは、クロックをデバイスに提供するために 280x デバイスに接続される外部水晶 / 発振子の使用を可能にします。水晶 / 発振子は X1/X2 ピンに接続され、XCLKIN は Low に接続されます。

- **外部クロック・ソース動作 :**

オンチップ・オシレータが使用されない場合、このモードは内部オシレータをバイパスすることを可能にします。デバイス・クロックは、X1 または XCLKIN ピンの外部クロック・ソース入力から生成されます。

オプション 1 : XCLKIN ピンの外部クロック :

XCLKIN を外部クロック・ソースとして使用する場合は、X1 を Low に接続し、X2 を未接続のままにする必要があります。この場合、外部オシレータ・クロックは XCLKIN ピンに接続されます。これにより、3.3V のクロック・ソースが使用できるようになります。

オプション 2 : X1 ピンの外部クロック :

X1 をクロック・ソースとして使用する場合は、XCLKIN を Low に接続し、X2 を未接続のままにする必要があります。この場合、外部オシレータ・クロックは X1 ピンに接続されます。これにより、1.8V のクロック・ソースが使用できるようになります。

OSC 回路は、X1 および X2 ピンを使用して 280x デバイスに水晶を接続することができます。水晶を使用しない場合は、外部オシレータを XCLKIN ピンに直接接続でき、X2 ピンは未接続のままにされ、X1 ピンは Low に接続されます。『TMS320F2809, TMS320F2808, TMS320F2806 TMS320F2802, TMS320F2801, UCD9501 TMS320C2802, TMS320C2801 Digital Signal Processors Data Manual』(文献番号 SPRS230) を参照してください。

表 3-6 可能な PLL 設定モード

PLL モード	摘要	PLLSTS [CLKINDIV] ⁽¹⁾	CLKIN および SYSCLKOUT
PLL オフ	PLLSTS レジスタの PLLOFF ビットをセットし、起動します。PLL ブロックは、このモードではディスエーブルになります。これは、システム・ノイズを低減する場合や低電力動作の場合に役立つことがあります。PLLCR レジスタは、このモードに入る前に 0x0000 (PLL バイパス) に設定されなければなりません。CPU クロック (CLKIN) は、X1/X2、X1、または XCLKIN の入力クロックから直接供給されます。	0	OSCCLK/2
		1	OSCCLK
PLL バイパス	PLL バイパスは、電源投入時または外部リセット (\overline{XRS}) 後のデフォルトの PLL 設定です。このモードは、PLLCR レジスタが 0x0000 に設定されているときか、PLLCR レジスタの変更後に PLL が新しい周波数にロックしているときに選択されます。このモードでは、PLL 自体はバイパスされますが、PLL はオフになりません。	0	OSCCLK/2
		1	OSCCLK
PLL イネーブル	ゼロ以外の値を PLLCR レジスタにライトすることによってアクティブになります。PLLCR へのライト時には、デバイスは PLL がロックするまで PLL バイパス・モードに切り替わります。	0	OSCCLK*n/2
		1	OSCCLK*n

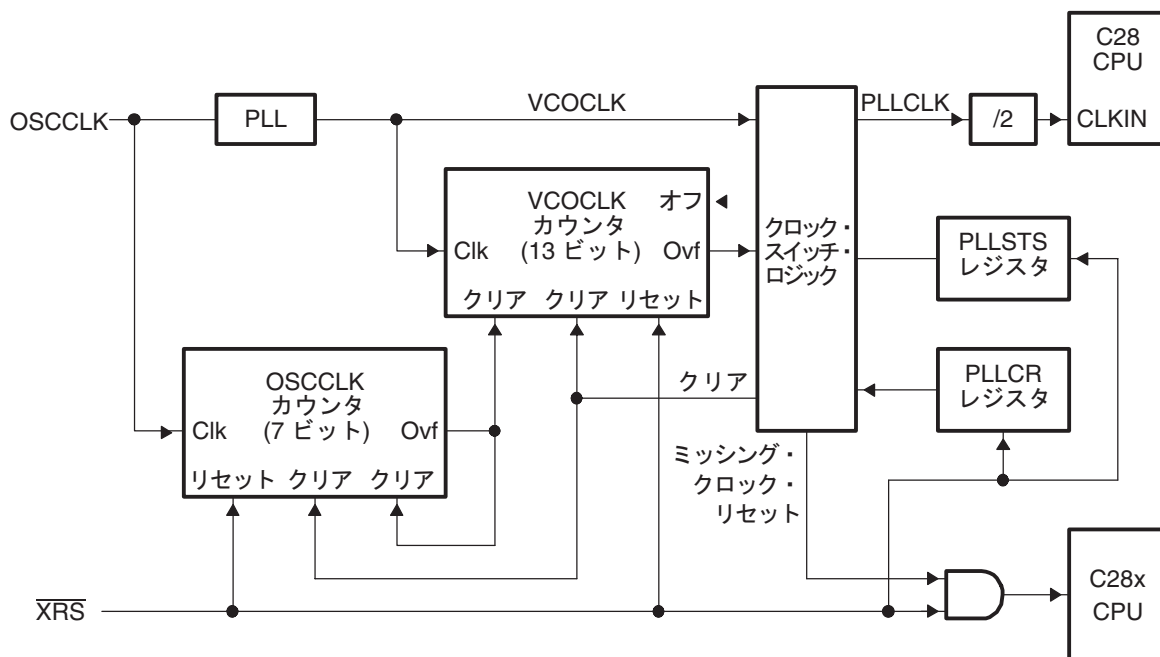
(1) PLLSTS[CLKINDIV] は、PLLCR へのライトの前は 0 でなければならず、PLLSTS[PLLLOCKS]=1 の後にのみ 1 に設定される必要があります。図 3-9 を参照してください。

3.2.2 メイン・オシレータ障害検出

振動により、DSP への外部クロック・ソースが切断され、デバイスのクロック供給が失敗することがあります。PLL がディスエーブルにされていない場合は、このセクションで説明するように、メイン・オシレータ障害ロジックは、デバイスがこの状況を検出して、デフォルトで既知の状態になります。

図 3-7 に示すように、2 つのカウンタは OSCCLK 信号の存在を監視するために使用されます。最初のカウンタは、X1/X2 または XCLKIN 入力からの OSCCLK 信号自体によってインクリメントされます。PLL がオフになっていない場合、2 番目のカウンタは PLL ブロックから出てくる VCOCLK によってインクリメントされます。これらのカウンタは、7 ビット OSCCLK カウンタがオーバーフローしたときに 13 ビット VCOCLK カウンタをクリアするように設定されます。通常の動作モードでは、OSCCLK が存在する限り、VCOCLK カウンタは決してオーバーフローしません。

図 3-7 オシレータ障害検出ロジック図



OSCCLK 入力信号が欠けている場合、PLL はデフォルトの「リンプ・モード」周波数を出力し、VCOCLK カウンタはインクリメントを続行します。OSCCLK 信号が欠けているため、OSCCLK カウンタはインクリメントしません。したがって、VCOCLK カウンタは定期的にクリアされません。その結果、VCOCLK カウンタはオーバーフローし、必要であればデバイスは CPU への CLKIN 入力を PLL のリンプ・モード出力周波数に切り替えます。

VCOCLK カウンタがオーバーフローすると、ミッシング・クロック検出ロジックは CPU、ペリフェラル、および他のデバイス・ロジックをリセットします。生成されたリセットは、ミッシング・クロック検出ロジック・リセット (MCLKRES) と呼ばれます。MCLKRES は内部リセットのみです。デバイスの外部 XRS ピンは、MCLKRES によって Low にされません。また、PLLCR および PLLSTS レジスタはリセットされません。

デバイスのリセットに加えて、ミッシング・オシレータ・ロジックは PLLSTS[MCLKSTS] レジスタ・ビットをセットします。MCLKCSTS ビットが 1 の場合、ミッシング・オシレータ検出ロジックがその部分をリセットすること、および CPU がリンプ・モード周波数またはその周波数の 2 分の 1 で現在動作していることを示します。

リセット後に PLLSTS[MCLKSTS] ビットをチェックして、ミッシング・クロックのために MCLKRES によってデバイスがリセットされたかどうかをソフトウェアでチェックする必要があります。MCLKSTS がセットされた場合、ファームウェアは、システム・シャットダウンなどのシステムに適したアクションをとる必要があります。ミッシング・クロック・ステータスは、PLLSTS[MCLKCLR] ビットに 1 をライトすることによってクリアできます。これは、ミッシング・クロック検出回路およびカウンタをリセットします。MCLKCLR ビットにライトした後でも OSCCLK 入力欠けている場合は、VCOCLK カウンタが再びオーバーフローし、プロセスが繰り返されます。

注： 正しい CPU 動作周波数が絶対に不可欠であるアプリケーションは、入力クロックが失敗した場合に DSP がリセットで保持されるようにするメカニズムを実装する必要があります。たとえば、コンデンサがフルに充電された場合に R-C 回路を使用して DSP の XRS ピンをトリガすることができます。I/O ピンを使用すると、定期的にコンデンサを放電して、フルに充電されることを防止できます。そのような回路は、フラッシュ・メモリや V_{DD3VFL} レールの障害の検出にも役立ちます。

以下の注意事項と制限事項を心に留めておく必要があります。

- **PLL 制御レジスタの変更には正しい手順を使用してください。**

PLLCR レジスタの変更には、[図 3-9](#) で概説されている手順に常に従ってください。

- **デバイスがリンプ・モードで動作している場合は、PLLCR レジスタにライトしないでください。**

PLLCR レジスタにライトすると、デバイスは OSCCLK/2 への CPU の CLKIN 入りに切り替わり、リンプ・モード後の動作が検出された場合は、OSCCLK が存在しない可能性があり、システムへのクロックが停止します。[図 3-9](#) に記載されているように、PLLCR レジスタにライトする前に PLLSTS[MCLKSTS] ビットが 0 であることを常にチェックしてください。

- **ウォッチドッグは、外部クロックなしでは機能しません。**

OSCCLK が存在しないと、ウォッチドッグは機能せず、リセット信号を生成できません。OSCCLK がなくなった場合にウォッチドッグをリンプ・モードに切り替えるための特殊なハードウェアは追加されていません。

- **リンプ・モードは、電源投入からは機能しないことがあります。**

電源投入から OSCCLK が欠けている場合は、PLL がリンプ・モードを生成しないことがあります。OSCCLK が最初に存在する場合にのみ、リンプ・モード・クロックが PLL によって生成されます。

- **デバイスがリンプ・モードで動作しているときには、HALT 低電力モードに入れないでください。**

デバイスがすでにリンプ・モードで動作しているときに HALT モードに入ろうとした場合、そのデバイスは正しく HALT に入らないことがあります。代わりに、そのデバイスは STANDBY モードに入ったりハングしたりする可能性があり、ユーザーは HALT モードを終了できないことがあります。そのため、HALT モードに入る前に PLLSTS[MCLKSTS] ビットが 0 であることを常にチェックしてください。

後続のセクションでは、さまざまな動作モードのミッシング・クロック検出ロジックの動作について説明します。

- **PLL バイパス・モード**

PLL コントローラ・レジスタが 0x0000 に設定されると、PLL はバイパスされます。PLLSTS[CLKINDIV] ビットの状態に応じて、OSCCLK または OSCCLK/2 が CPU の入力クロック (CLKIN) に直接接続されます。OSCCLK がミッシング・クロックとして検出された場合、デバイスは自動的に PLL に切り替わり、ミッシング・クロック検出ステータス・ビットをセットして、ミッシング・クロック・リセットを生成します。これで、デバイスは PLL リンブ・モード周波数またはその周波数の 2 分の 1 で動作するようになります。

- **PLL イネーブル・モード**

PLL 制御レジスタがゼロ以外の場合 (PLLCR=n で、n=0x0000 の場合)、PLL はイネーブルになります。このモードでは、OSCCLK*n/2 または OSCCLK*n が CPU の CLKIN に接続されます。OSCCLK がミッシングとして検出された場合、ミッシング・クロック検出ステータス・ビットがセットされ、デバイスはミッシング・クロック・リセットを生成します。これで、デバイスは PLL リンブ・モード周波数の 2 分の 1 で動作するようになります。

- **STANDBY 低電力モード**

このモードでは、CPU への CLKIN が停止されます。ミッシング入力クロックが検出された場合は、ミッシング・クロック・ステータス・ビットがセットされ、デバイスはミッシング・クロック・リセットを生成します。これが発生したときに PLL がバイパス・モードだった場合は、PLL リンブ周波数の 2 分の 1 が自動的に CPU にルーティングされます。これで、デバイスは PLLSTS[CLKINDIV] ビットの状態に応じて PLL リンブ・モード周波数またはその周波数の 2 分の 1 で動作するようになります。

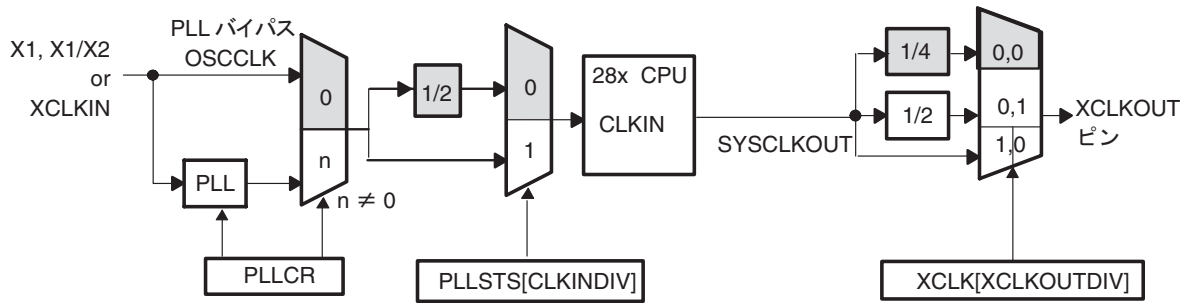
- **HALT 低電力モード**

HALT 低電力モードでは、デバイスへのクロックがすべてオフになります。デバイスが HALT モードから出ると、オシレータと PLL が起動します。ミッシング入力クロックを検出するために使用されるカウンタ (VCOCLK と OSCCLK) は、この起動が完了した後のみイネーブルになります。VCOCLK カウンタがオーバーフローした場合は、ミッシング・クロック検出ステータス・ビットがセットされ、デバイスはミッシング・クロック・リセットを生成します。オーバーフローの発生時に PLL がバイパス・モードだった場合は、PLL リンブ周波数の 2 分の 1 が自動的に CPU にルーティングされます。これで、デバイスは PLLSTS[CLKINDIV] ビットの状態に応じて PLL リンブ・モード周波数またはその周波数の 2 分の 1 で動作するようになります。

3.2.3 XCLKOUT 生成

図 3-8 に示されるように、XCLKOUT 信号は、システム・クロック SYSCLKOUT から直接派生されます。XCLKOUT は、XCLK レジスタの XCLKOUTDIV ビットによって決定された SYSCLKOUT に等しくするか、その 2 分の 1 にするか、または 4 分の 1 にすることができます。デフォルトでは、電源投入時は、 $XCLKOUT = SYSCLKOUT/4$ または $XCLKOUT = OSCCLK/8$ です。

図 3-8 XCLKOUT 生成



リセット時のデフォルト

リセットがアクティブの場合、XCLKOUT 信号はアクティブです。リセットが Low のときに XCLKOUT は SYSCLKOUT/4 を反映するため、ユーザーはこの信号を監視して、デバッグ時にデバイスが正しくクロック出力されているかどうかを確認することができます。XCLKOUT ピン上には内部プルアップまたはプルダウンはありません。

XCLKOUT が使用されていない場合は、XCLK レジスタの XCLKOUTDIV ビットを「1,1」に設定することにより、XCLKOUT をオフにすることができます。

3.2.4 PLL 制御 (PLLCR) レジスタ

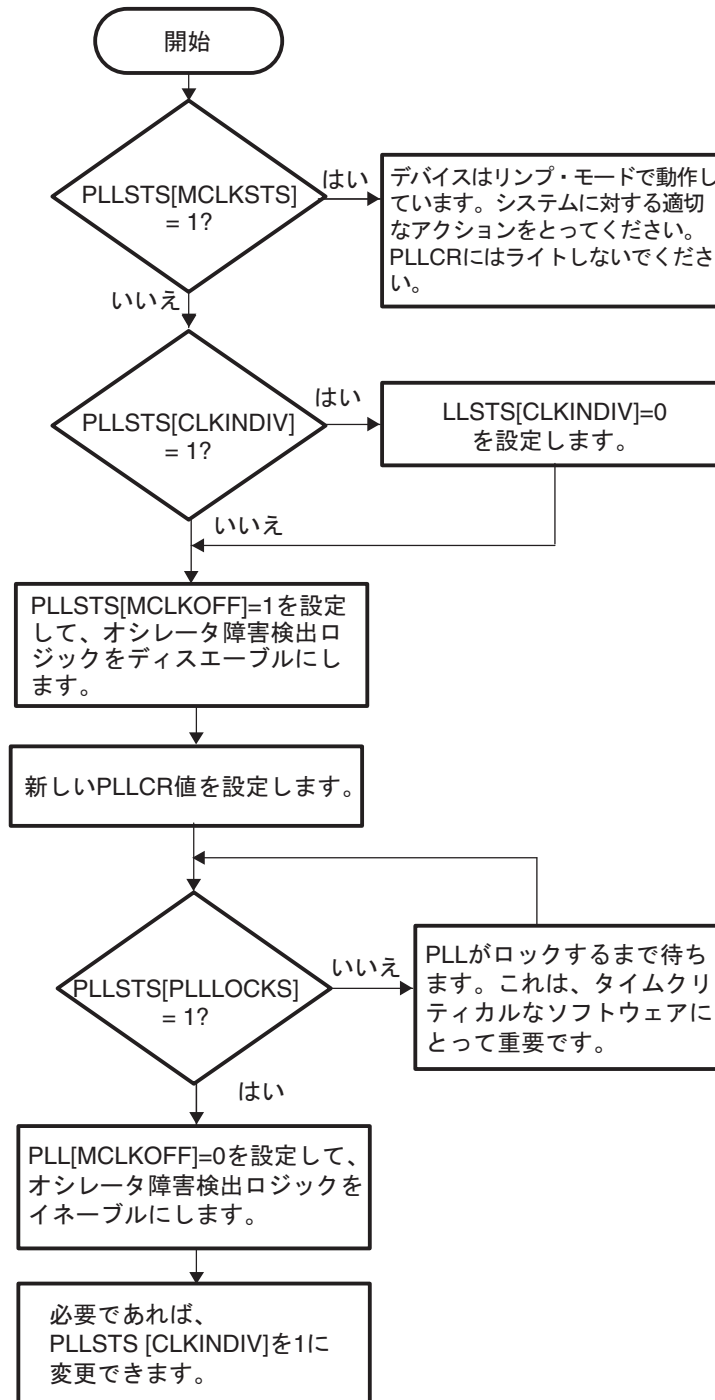
PLLCR レジスタは、デバイスの PLL 通倍器を変更するために使用されます。PLLCR レジスタにライトする前に、以下の 2 つの要件を満たす必要があります。

1. PLLSTS[CLKINDIV] ビットは 0 でなければなりません (イネーブルにされた CLKIN 2 分周)。PLL がロックを完了した後 (つまり、PLLSTS[PLLLOCKS]=1 の後) で PLLSTS[CLKINDIV] を 1 に変更してください。
2. デバイスは「リンプ・モード」で動作していない必要があります。つまり、PLLSTS[MCLKSTS] ビットは 0 でなければなりません。

CPU が PLLCR[DIV] ビットにライトすると、PLL ロジックは CPU クロック (CLKIN) を OSCCLK/2 に切り替えます。PLL が安定し、指定された周波数でロックすると、表 3-7 に示されているように、PLL は CLKIN を新しい値に切り替えます。これが発生すると、PLLSTS レジスタの PLLLOCKS ビットがセットされ、PLL がロックを完了していること、およびデバイスが新しい周波数で現在動作していることを示します。ユーザー・ソフトウェアは、PLLLOCKS ビットを監視して、PLL がロックを完了したかどうかを判断できます。PLLSTS[PLLLOCKS]=1 になったら、必要に応じて CLKINDIV を 1 に変更して、CLKIN 2 分周をディスエーブルにすることができます。

PLLCR レジスタにライトするときは常に、図 3-9 の手順に従ってください。

図 3-9 PLLCR 変更手順フロー・チャート



3.2.5 PLL 制御、ステータス、XCLKOUT レジスタの説明

PLLCR レジスタの DIV フィールドは、PLL がバイパスされるかどうかを制御し、バイパスされない場合は PLL クロックの通倍を設定します。PLL バイパスは、リセット後のデフォルト・モードです。PLLSTS[CLKINDIV] ビットがセットされている場合や、セットされている PLLSTS[MCLKSTS] ビットに示されるように PLL がリンプ・モードで動作している場合は、DIV フィールドにライトしないでください。図 3-9 に記載されている PLLCR の変更手順を参照してください。

図 3-10 PLLCR レジスタのレイアウト

15	予約	4	3	0
R-0			DIV R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-7 PLLCR ビットの説明

PLLCR[DIV] 値 ⁽²⁾	SYSCLKOUT (CLKIN) ⁽¹⁾	
	PLLSTS[CLKINDIV] = 0 ⁽³⁾	PLLSTS[CLKINDIV] = 1 ⁽³⁾
0000 (PLL バイパス)	OSCCLK/2 (デフォルト) ⁽⁴⁾	OSCCLK
0001	(OSCCLK*1)/2	OSCCLK*1
0010	(OSCCLK*2)/2	OSCCLK*2
0011	(OSCCLK*3)/2	OSCCLK*3
0100	(OSCCLK*4)/2	OSCCLK*4
0101	(OSCCLK*5)/2	OSCCLK*5
0110	(OSCCLK*6)/2	OSCCLK*6
0111	(OSCCLK*7)/2	OSCCLK*7
1000	(OSCCLK*8)/2	OSCCLK*8
1001	(OSCCLK*9)/2	OSCCLK*9
1010	(OSCCLK*10)/2	OSCCLK*10
1011 - 1111	予約	予約

- (1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。
- (2) CLKIN は CPU への入力クロックです。SYSCLKOUT は CPU からの出力クロックです。SYSCLKOUT の周波数は CLKIN と同じです。
- (3) PLLSTS[CLKINDIV] は、PLLCR へのライトの前は 0 でなければならず、PLLSTS[PLLLOCKS]=1 の後にのみ 1 に設定される必要があります。図 3-9 を参照してください。
- (4) PLL 制御レジスタ (PLLCR) および PLL ステータス・レジスタ (PLLSTS) は、XRS 信号またはウォッチドッグ・リセットによってのみデフォルト状態にリセットされます。デバッグまたはミッシング・クロック検出ロジックによって発行されたリセットには何の効果もありません。

図 3-11 PLL ステータス・レジスタ (PLLSTS)

予約							8
R-0							
7	6	5	4	3	2	1	0
予約	MCLKOFF	OSCOFF	MCLKCLR	MCLKSTS	PLLOFF	CLKINDIV	PLLLOCKS
R-0	R/W-0	R/W-0	R=0/W-0	R-0	R/W-0	R/W-0	R-1

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-8 PLL ステータス・レジスタ (PLLSTS) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾⁽²⁾
15-7	予約		予約
6	MCLKOFF	0 1	<p>ミッシング・クロック検出オフ・ビット</p> <p>0 メイン・オシレータ障害検出ロジックがイネーブルになります。(デフォルト)</p> <p>1 メイン・オシレータ障害検出ロジックはディスエーブルになり、PLL はリンプモード・クロックを送出しません。コードが検出回路の影響を受けてはいけない場合は、このモードを使用してください。たとえば、外部クロックがオフにされている場合です。</p>
5	OSCOFF	0 1	<p>オシレータ・クロック・オフ・ビット</p> <p>0 X1、X1/X2、または XCLKIN からの OSCCLK 信号は、PLL ブロックに供給されます。(デフォルト)</p> <p>1 X1、X1/X2、または XCLKIN からの OSCCLK 信号は、PLL ブロックに供給されません。これは、内部オシレータをシャット・ダウンしません。OSCOFF ビットは、ミッシング・クロック検出ロジックのテストのために使用されます。</p> <p>OSCOFF ビットがセットされている場合は、HALT または STANDBY モードに入れたり、PLLCR にライトしたりすると予測できない動作が発生する可能性があるため、これらの操作は行わないでください。</p> <p>OSCOFF ビットがセットされていると、ウォッチドッグの動作は、どの入力クロック・ソース (X1、X1/X2、または XCLKIN) が使用されているかに応じて異なります。</p> <ul style="list-style-type: none"> • X1 または X1/X2 : ウォッチドッグは機能しません。 • XCLKIN : ウォッチドッグは機能し、OSCOFF の設定前にディスエーブルにする必要があります。
4	MCLKCLR	0 1	<p>ミッシング・クロック・クリア・ビット。</p> <p>0 0 をライトしても影響はありません。このビットは常に 0 をリードします。</p> <p>1 ミッシング・クロック検出回路のクリアとリセットを強制します。OSCCLK がまだ欠けている場合は、検出回路はシステムへのリセット信号を再び生成し、ミッシング・クロック・ステータス・ビット (MCLKSTS) をセットします。また、CPU は、「リンプ・モード」周波数で動作している PLL によって動作されます。</p>
3	MCLKSTS	0 1	<p>ミッシング・クロック・ステータス・ビット。リセット後に、このビットのステータスをチェックして、ミッシング・オシレータ状況が検出されたかどうかを判断します。通常では、このビットは 0 になります。このビットへのライトは無視されます。このビットは、MCLKCLR ビットへのライトまたは外部リセットの強制によってクリアされます。</p> <p>0 通常の動作を示します。ミッシング・クロック状況は検出されていません。</p> <p>1 OSCCLK がミッシングとして検出されたことを示します。メイン・オシレータ障害検出ロジックはデバイスをリセットし、リンプ・モード周波数で動作している PLL によって CPU はクロックされます。</p>
2	PLLOFF	0 1	<p>PLL オフ・ビット。このビットは PLL をオフにします。これは、システム・ノイズ・テストに役立ちます。このモードは、PLLCR レジスタが 0x0000 に設定されている場合にのみ使用できます。</p> <p>0 PLL オン (デフォルト)</p> <p>1 PLL オフ。PLLOFF ビットがセットされている間は、PLL モジュールはパワーオフの状態になります。</p> <p>1 を PLLOFF にライトする前に、デバイスは PLL バイパス・モード (PLLCR=0x0000) になっている必要があります。PLL がオフ (PLLOFF=1) になっている間は、ゼロ以外の値を PLLCR にライトしないでください。</p> <p>PLLOFF=1 の場合 STANDBY および HALT 低電力モードは、期待どおりに機能します。HALT または STANDBY からのウェイクアップ後は、PLL モジュールはパワーが切断されたままになります。</p>
1	CLKINDIV	0 1	<p>クロック 2 分周イネーブルまたはディスエーブル : CPU への CLKIN 信号の 2 分周は、このビットによってイネーブルまたはディスエーブルにすることができます。CLKINDIV は、PLLCR にライトする前は 0 でなければならず、PLLLOCKS=1 の後にのみセットされる必要があります。図 3-9 を参照してください。</p> <p>0 CLKIN 2 分周はイネーブルになります (デフォルト)。CLKINDIV は、PLLCR へのライトの前に 0 に設定される必要があります。</p> <p>1 CLKIN 2 分周はディスエーブルになります。PLLLOCKS=1 の後にのみ、CLKINDIV を 1 に変更してください。</p>

(1) このレジスタは、XRS 信号またはウォッチドッグ・リセットによってのみ、デフォルト状態にリセットされます。このレジスタは、ミッシング・クロックやデバッグ・リセットによってリセットされません。

(2) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

表 3-8 PLL ステータス・レジスタ (PLLSTS) のフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾⁽²⁾
0	PLLLOCKS	0	PLL ロック・ステータス・ビット PLLCR レジスタがライトされていること、および PLL が現在ロックしていることを示します。CPU は、PLL がロックするまで OSCCLK/2 によってクロックされます。
		1	PLL がロックを完了して現在安定していることを示します。

図 3-12 XCLKOUT レジスタ (XCLK)

15	XCLKINCNT				12	11	X1CNT			8
	R/W-0						R/W-0			
7	予約	5	4	3	2	1	0			
	R-0		W-0	R-0	R-0	R/W-0				

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-9 XCLKOUT レジスタ (XCLK) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15-12	XCLKINCNT		このビットは、テキサス・インスツルメンツ専用予約されています。
11-8	X1CNT		このビットは、テキサス・インスツルメンツ専用予約されています。
7-5	予約		
4	XCLKINDAT		このビットは、テキサス・インスツルメンツ専用予約されています。
3	X1DAT		このビットは、テキサス・インスツルメンツ専用予約されています。
2	XCLKOUTDAT		このビットは、テキサス・インスツルメンツ専用予約されています。
1	XCLKOUTDIV		XCLKOUT 分周率。これらの 2 つのビットは、SYSCLKOUT を基準として XCLKOUT 周波数比率を選択します。その比率は以下のとおりです。
		00	XCLKOUT=SYSCLKOUT/4 (デフォルト)
		01	XCLKOUT = SYSCLKOUT/2
		10	XCLKOUT = SYSCLKOUT
		11	XCLKOUT= オフ (ピンはハイインピーダンス・モード)

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

3.2.6 外部基準オシレータ・クロック・オプション

TI は、お客様が発振子 / 水晶ベンダに DSP デバイスの動作特性を測定することを推奨しています。発振子 / 水晶ベンダは、タンク回路を調整するための装置と専門技術を持っています。また、正しいスタートアップと全動作範囲にわたる安定性を提供するための適切なタンク・コンポーネント値について、お客様に助言できます。

3.3 低電力モード・ブロック

280x デバイス上の低電力モードは、240x デバイスに似ています。[表 3-10](#) は、各種モードを要約しています。

各種低電力モードは、[表 3-11](#) に示されているように動作します。

低電力モードに入る場合および低電力モードから出る場合の正確なタイミングについては、*『TMS320F2809, TMS320F2808, TMS320F2806 TMS320F2802, TMS320F2801, UCD9501 TMS320C2802, TMS320C2801 Digital Signal Processors Data Manual』* (文献番号 SPRS230) を参照してください。

表 3-10 280x 低電力モード

モード	LPMCR0[1:0]	OSCCLK	CLKIN	SYSCLKOUT	終了 ⁽¹⁾
IDLE	00	オン	オン	オン ⁽²⁾	XRS、 ウォッチドッグ割り込み、 イネーブルにされている 任意の割り込み、 XNMI
STANDBY	01	オン (ウォッチドッグは 引き続き動作中)	オフ	オフ	XRS、 ウォッチドッグ割り込み、 GPIO ポート A 信号、 デバッグ ⁽³⁾
HALT	1X	オフ (オシレータおよび PLL はオフ、 ウォッチドッグは 機能していない)	オフ	オフ	XRS、 GPIO ポート A 信号、 デバッグ ⁽³⁾

- (1) 「終了」の列には、どの信号、またはどのような条件下で低電力モードが終了されるのかが記載されています。デバイスによって割り込みが認識されるためには、この信号を十分に長く Low に保持する必要があります。そうしないと、IDLE モードは終了されず、デバイスは示された低電力モードに戻ります。
- (2) 28x 上の IDLE モードの動作は、24x/240x 上の動作とは異なります。28x では CPU からのクロック出力 (SYSCLKOUT) は引き続き機能しますが、24x/240x ではクロックはオフになります。
- (3) 28x では、たとえ CPU へのクロック (CLKIN) がオフになっていても、JTAG ポートは引き続き機能することができます。

表 3-11 低電力モード

モード	説明
IDLE モード :	このモードは、イネーブルにされた任意の割り込み、または NMI によって終了されます。このモード中に LPM ブロック自体は何のタスクも実行しません。
STANDBY モード :	LPMCR0 レジスタの LPM ビットが 01 に設定された場合に、IDLE 命令が実行されるとデバイスが STANDBY モードに入ります。STANDBY モードでは、CPU へのクロック入力 (CLKIN) がディスエーブルになります。これにより、SYSCLKOUT から派生したクロックはすべてディスエーブルになります。オシレータ、PLL、およびウォッチドッグは、引き続き機能します。STANDBY モードに入る前に、以下のタスクを実行する必要があります。 <ul style="list-style-type: none"> PIE モジュールの WAKEINT 割り込みをイネーブルにします。この割り込みは、ウォッチドッグと低電力モード・モジュール割り込みの両方に接続されます。 必要であれば、GPIO_LPMSEL レジスタでいずれかの GPIO ポート A 信号を指定してデバイスをウェイクします。GPIO_LPMSEL レジスタは、GPIO モジュールの一部です。選択した GPIO 信号のほかに、XRS 入力とウォッチドッグ割り込みも、LPMCR0 レジスタでイネーブルにされた場合は、STANDBY モードからデバイスをウェイクすることができます。 デバイスをウェイクする信号について、LPMCR0 レジスタで入力フィルタリング (Qualification) を選択します。 <p>選択した外部信号が Low になる場合、その信号は、LPMCR0 レジスタのフィルタリング (Qualification) 周期によって指定された OSCCLK サイクルの数だけ Low のままでなければなりません。この時間中に信号が High にサンプリングされた場合は、フィルタリング (Qualification) が再開します。フィルタリング (Qualification) 周期が終了すると、PLL は CPU への CLKIN をイネーブルにし、WAKEINT 割り込みは PIE ブロックでラッチされます。その後、WAKEINT 割り込みがイネーブルになっている場合は、CPU が WAKEINT 割り込みに応答します。</p>

表 3-11 低電力モード（続き）

モード	説明
HALT モード :	<p>LPMCR0 レジスタの LPM ビットが 1x に設定された場合に、IDLE 命令が実行されるとデバイスが HALT モードに入ります。HALT モードでは、PLL やオシレータも含め、すべてのデバイス・クロックがシャットダウンされます。HALT モードに入る前に、以下のタスクを実行する必要があります。</p> <ul style="list-style-type: none"> • PIE モジュールの WAKEINT 割り込みをイネーブルにします。この割り込みは、ウォッチドッグと低電力モード・モジュール割り込みの両方に接続されます。 • 必要であれば、GPIOLPMSEL レジスタでいずれかの GPIO ポート A 信号を指定してデバイスをウェイクします。GPIOLPMSEL レジスタは、GPIO モジュールの一部です。選択した GPIO 信号のほかに、XRS 入力も STADBY モードからデバイスをウェイクします。 <p>デバイスがリンプ・モード (PLLSTS[MCLKSTS]=1) で動作しているときには、HALT 低電力モードに入らないでください。</p> <p>デバイスがすでにリンプ・モードで動作しているときに HALT モードに入ろうした場合、そのデバイスは正しく HALT モードに入らないことがあります。さらに、そのデバイスは STANDBY モードに入ったりハングしたりする可能性があり、ユーザーは HALT モードを終了できないことがあります。そのため、HALT モードに入る前に PLLSTS[MCLKSTS] ビットが 0 であることを常にチェックしてください。</p> <p>選択した外部信号が Low になる場合、その信号は非同期に LPM ブロックへ供給されます。オシレータはオンになり、電源投入を開始します。オシレータが電源投入を完了するためには、十分に長く信号を Low に保持する必要があります。信号が元の High に再びドライブされた場合、これは非同期に PLL をリリースし、PLL はロックを開始します。いったん PLL がロックすると、その PLL は CLKIN を CPU に供給し、その時点で CPU は WAKEINT 割り込みに応答します (WAKEINT 割り込みをイネーブルにした場合)。</p>

低電力モードは、LPMCR0 レジスタによって制御されます (図 3-13)。

図 3-13 低電力モード制御 0 レジスタ (LPMCR0)

15	14	8	7	2	1	0
WDINTE	予約	QUALSTDBY		LPM		
R/W-0	R-0	R/W-1		R/W-0		

凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-12 低電力モード制御 0 レジスタ (LPMCR0) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15	WDINTE	0 1	ウォッチドッグ割り込みイネーブル ウォッチドッグ割り込みは、STANDBY からデバイスをウェイクすることを許可されません。(デフォルト) ウォッチドッグは、STANDBY からデバイスをウェイクすることを許可されます。ウォッチドッグ割り込みも、SCSR レジスタでイネーブルにされる必要があります。
14-8	予約		予約
7-2	QUALSTDBY	000000 000001 ... 111111	OSCCLK クロック・サイクルの数を選択して、STANDBY モードからデバイスをウェイクする選択済みの GPIO 入力をフィルタリング (Qualification) します。このフィルタリング (Qualification) は、STANDBY モードでのみ使用されます。デバイスを STANDBY からウェイクすることができる GPIO 信号は、GPIOLPMSEL レジスタで指定されます。 2 OSCCLK (デフォルト) 3 OSCCLKs ... 65 OSCCLKs
1-0	LPM ⁽²⁾	00 01 10 11	これらのビットは、デバイスの低電力モードを設定します。 低電力モードを IDLE に設定する (デフォルト) 低電力モードを STANDBY に設定する 低電力モードを HALT に設定する ⁽³⁾ 低電力モードを HALT に設定する ⁽³⁾

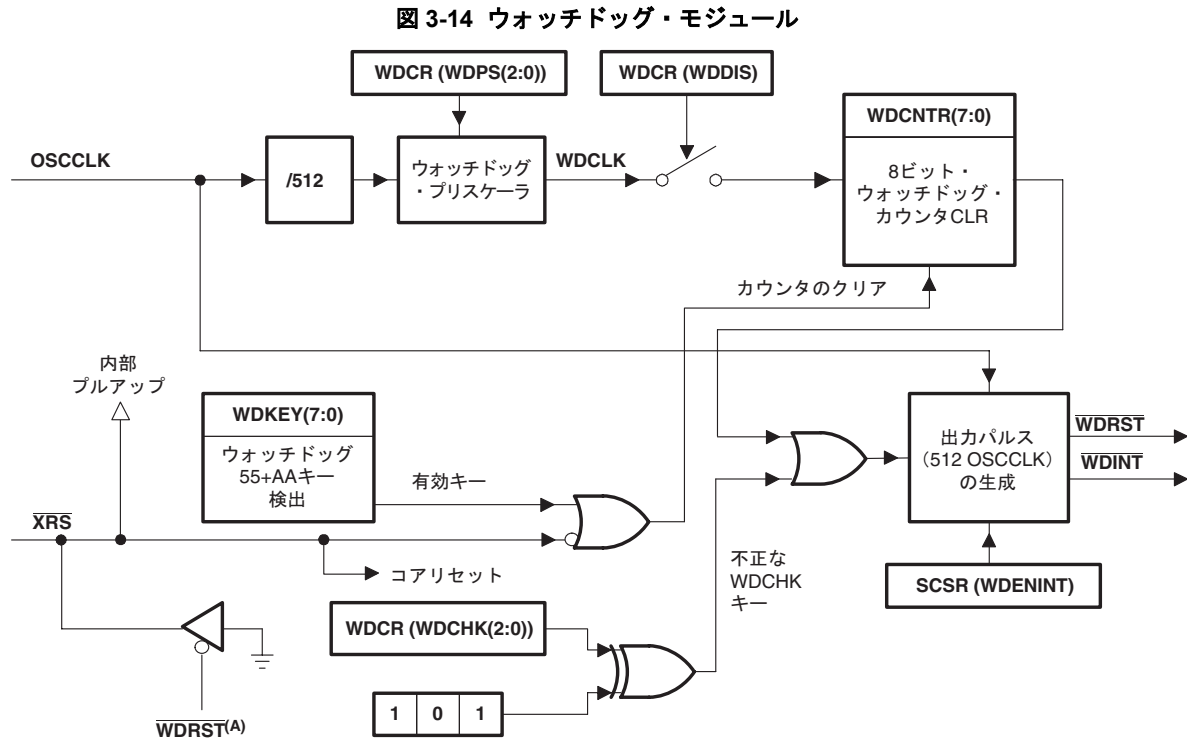
(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) 低電力モード・ビット (LPM) は、IDLE 命令が実行された場合にのみ有効になります。したがって、IDLE 命令を実行する前に LPM ビットを適切なモードに設定する必要があります。

(3) デバイスがすでにリンプ・モードで動作しているときに HALT モードに入ろうとした場合、そのデバイスは正しく HALT に入らないことがあります。さらに、そのデバイスは STANDBY モードに入ったりハングしたりする可能性があり、ユーザーは HALT モードを終了できないことがあります。そのため、HALT モードに入る前に PLLSTS[MCLKSTS] ビットが 0 であることを常にチェックしてください。

3.4 ウォッチドッグ・ブロック

ウォッチドッグ・モジュールは、8ビット・ウォッチドッグ・アップ・カウンタが最大値に達するたびに、512 オシレータクロック (OSCCLK) 幅の出力パルスを生成します。これを防止するには、ユーザーがカウンタをディスエーブルにするか、あるいは、ソフトウェアがウォッチドッグ・カウンタをリセットするウォッチドッグ・キー・レジスタに 0x55、0xAA の順番で定期的にライトする必要があります。図 3-14 は、ウォッチドッグ・モジュール内のさまざまな機能ブロックを示します。



- A WDRST および XRS 信号は、ウォッチドッグ・リセットが発生すると 512_OSCCLK サイクルについて Low にドライブされます。同様に、ウォッチドッグ割り込みをイネーブルにした場合、WDINT 信号は、割り込みが発生すると 512 OSCCLK サイクル Low にドライブされます。

注： 初期バージョンのシリコンでは、無効な値を WDKEY レジスタにライトすると、ウォッチドッグ・リセットまたは割り込みが強制されます。この機能は削除されています。この変更に関する詳細については、『TMS320F2808, TMS320F2806, TMS320F2802, TMS320F2801, UCD9501, TMS320C2802, TMS320C2801 DSP Silicon Errata』(SPRZ171) を参照してください。代わりにデバイスのリセットをウォッチドッグに強制するためには、不正な値を WDCR レジスタの WDCHK ビットにライトしてください。

3.4.1 ウォッチドッグ・タイマの処理

8 ビット・ウォッチドッグ・カウンタ (WDCNTR) がオーバーフローする前に正しいシーケンスで WDKEY レジスタにライトされると、WDCNTR はリセットされます。0x55 の値が WDKEY にライトされると、WDCNTR はリセット可能になります。WDKEY レジスタにライトされる次の値が 0xAA の場合、WDCNTR はリセットされます。0x55 または 0xAA 以外の WDKEY にライトされた値は、何のアクションも引き起こしません。0x55 および 0xAA 値の任意のシーケンスは、システム・リセットを引き起こすことなく WDKEY にライトすることができます。WDKEY に対して 0x55 のライトの後に 0xAA のライトが続いた場合にのみ、WDCNTR がリセットされます。

表 3-13 ウォッチドッグ・キー・シーケンスの例

ステップ	WDKEY にライトされる値	結果
1	0xAA	アクションなし
2	0xAA	アクションなし
3	0x55	次の値が 0xAA の場合、WDCNTR はリセット可能になります。
4	0x55	次の値が 0xAA の場合、WDCNTR はリセット可能になります。
5	0x55	次の値が 0xAA の場合、WDCNTR はリセット可能になります。
6	0xAA	WDCNTR はリセットされます。
7	0xAA	アクションなし
8	0x55	次の値が 0xAA の場合、WDCNTR はリセット可能になります。
9	0xAA	WDCNTR はリセットされます。
10	0x55	次の値が 0xAA の場合、WDCNTR はリセット可能になります。
11	0x32	WDKEY にライトされた不適当な値。 アクションなし。次が 0xAA であっても WDCNTR はリセットできない。
12	0xAA	前の無効値のためにアクションなし。
13	0x55	次の値が 0xAA の場合、WDCNTR はリセット可能になります。
14	0xAA	WDCNTR はリセットされます。

表 3-13 のステップ 3 は、WDCNTR のリセットを可能にする最初のアクションです。WDCNTR は、実際にはステップ 6 までリセットされません。ステップ 8 は WDCNTR のリセットを再び可能にし、ステップ 9 は WDCNTR をリセットします。ステップ 10 は、WDCNTR のリセットを再び可能にします。ステップ 11 で不正なキー値を WDKEY にライトしても何のアクションも起こりません。さらに、WDCNTR はリセット可能にならなくなり、ステップ 12 の 0xAA には何の作用もなくなります。

ウォッチドッグがデバイスをリセットするように設定されている場合、WDCR オーバーフロー、または WDCR[WDCHK] ビットに対する不正値を待つ処理は、デバイスをリセットし、WDCR レジスタのウォッチドッグ・フラグ (WDFLAG) をセットします。リセット後は、プログラムはこのフラグの状態をリードして、リセットの原因が判断できます。リセット後は、ソフトウェアによって WDFLAG をクリアし、後続リセットの原因が判断できるようにする必要があります。このフラグがセットされると、ウォッチドッグ・リセットは防止されません。

3.4.2 ウォッチドッグ・リセットまたはウォッチドッグ割り込みモード

ウォッチドッグを SCSR レジスタで設定すると、ウォッチドッグ・カウンタが最大値に達した場合にデバイスをリセット (WDRST) するか、または CPU 割り込み (WDINT) することができます。各状況の動作を以下に示します。

- **リセット・モード:**

ウォッチドッグがデバイスをリセットするように設定された場合は、ウォッチドッグ・カウンタが最大値に達すると、WDRST 信号は 512 OSCCLK サイクルにわたってデバイス・リセット (XRS) ピンを Low にします。

• **割り込みモード：**

ウォッチドッグが CPU 割り込みするように設定された場合、 $\overline{\text{WDINT}}$ 信号は 512 OSCCLK サイクルにわたって Low にドライブされます。これにより、PIE 内の WAKEINT 割り込みが PIE モジュールでイネーブルになっている場合、その割り込みが使用されます。ウォッチドッグ割り込みは、WDINT の立ち下がりがエッジでエッジ・トリガされます。したがって、WDINT がインアクティブになる前に WAKEINT 割り込みを再びイネーブルにした場合、ユーザーは別の割り込みをすぐに取得しません。次の WAKEINT 割り込みは、次のウォッチドッグ・タイムアウト時に発生します。

WDINT がまだロー・アクティブであるときに、ウォッチドッグが割り込みモードからリセット・モードへ再設定された場合は、デバイスがすぐにリセットします。SCSR レジスタの $\overline{\text{WDINT}}$ ビットをリードすると、ウォッチドッグをリセット・モードに再設定する前に $\overline{\text{WDINT}}$ 信号の現在の状態を確認できます。

3.4.3 低電力モードにおけるウォッチドッグ動作

STANDBY モードでは、ペリフェラルに対するクロックのすべてがデバイスでオフにされます。ウォッチドッグ・モジュールはオシレータ・クロック (OSCCLK) とは別に動作するため、機能できる状態を存続する唯一のペリフェラルです。WDINT 信号は低電力モード (LPM) ブロックに供給されます。これにより、WDINT 信号を使用して STANDBY 低電力モードからデバイスをウェイクすることができるようになります (イネーブルにした場合)。詳細については、デバイス・データ・マニュアルの低電力モード・ブロックのセクションを参照してください。

IDLE モードでは、ウォッチドッグ割り込み ($\overline{\text{WDINT}}$) 信号は、CPU への割り込みを生成して、CPU を IDLE モードからウェイクすることができます。ウォッチドッグは、PIE 内の WAKEINT 割り込みに接続されます。

注： ウォッチドッグ割り込みを使用して IDLE または STANDBY 低電力モード状態からウェイクアップする場合は、IDLE または STANDBY モードに戻ることを試みる前に、 $\overline{\text{WDINT}}$ 信号が High に再び戻ることを確認してください。ウォッチドッグ割り込みが生成されると、 $\overline{\text{WDINT}}$ 信号は 512 OSCCLK サイクルにわたって Low に保持されます。SCSR レジスタのウォッチドッグ割り込みステータス・ビット (WDINTS) をリードすることにより、WDINT の現状を判断できます。WDINTS は、2 SYSCLKOUT サイクルによって $\overline{\text{WDINT}}$ の状態に従います。

HALT モードでは、オシレータ (および PLL) はオフになるため (したがって、ウォッチドッグもオフになるため)、この機能は使用できません。

3.4.4 エミュレーションについての考慮事項

ウォッチドッグ・モジュールは、各種デバッグ状況では以下のように動作します。

CPU 一時停止：	CPU が一時停止されると、ウォッチドッグ・クロック (WDCLK) は一時停止されます。
ランフリー・モード：	CPU がランフリー・モードに置かれると、ウォッチドッグ・モジュールは動作を通常どおり再開します。
リアルタイム・シングル・ステップ・モード：	CPU がリアルタイム・シングル・ステップ・モードのときには、ウォッチドッグ・クロック (WDCLK) は一時停止されます。ウォッチドッグは、リアルタイム割り込み内であっても、一時停止のままになります。
リアルタイム・ランフリー・モード：	CPU がリアルタイム・ランフリー・モードのときには、ウォッチドッグは通常どおり動作します。

3.4.5 ウォッチドッグ・レジスタ

システム制御およびステータス・レジスタ (SCSR) には、ウォッチドッグ・オーバーライド・ビットとウォッチドッグ割り込みイネーブル/ディスエーブル・ビットが含まれます。図 3-15 は、SCSR レジスタのビット機能を表します。

図 3-15 システム制御およびステータス・レジスタ (SCSR)

15							8		
予約 R-0									
7	6	5	4	3	2	1	0		
予約 R-0					WDINTS R-1	WDENINT R/W-0	WDOVERRIDE R/W1C-1		

凡例 : R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-14 システム制御およびステータス・レジスタ (SCSR) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15-3	予約		
2	WDINTS	0 1	ウォッチドッグ割り込みステータス・ビット。WDINTS は、ウォッチドッグ・ブロックからの WDINT 信号の現状を反映します。WDINTS は、2 SYSCLKOUT サイクルによって WDINT の状態に従います。 ウォッチドッグ割り込みを使用してデバイスを IDLE または STANDBY 低電力モードからウェイクする場合は、このビットをリードして、IDLE または STANDBY モードに戻ることを試みる前に WDINT がアクティブでないことを確認してください。 0 ウォッチドッグ割り込み信号 (WDINT) はアクティブです。 1 ウォッチドッグ割り込み信号 (WDINT) はアクティブではありません。
1	WDENINT	0 1	ウォッチドッグ割り込みイネーブル。 0 ウォッチドッグ・リセット (WDRST) 出力信号はイネーブルにされ、ウォッチドッグ割り込み (WDINT) 出力信号はディスエーブルにされます。これは、リセット (XRS) 時のデフォルト状態です。ウォッチドッグ割り込みが発生すると、WDRST 信号は 512 OSCCLK サイクルにわたって Low のままになります。 WDINT が Low の間に WDENINT ビットがクリアされた場合は、リセットがすぐに発生します。WDINTS ビットをリードすると、WDINT 信号の状態を確認できます。 1 WDRST 出力信号はディスエーブルにされ、WDINT 出力信号はイネーブルにされます。ウォッチドッグ割り込みが発生すると、WDINT 信号は 512 OSCCLK サイクルにわたって Low のままになります。 ウォッチドッグ割り込みを使用してデバイスを IDLE または STANDBY 低電力モードからウェイクする場合は、WDINTS ビットを使用して、IDLE または STANDBY モードに戻ることを試みる前に WDINT がアクティブでないことを確認してください。
0	WDOVERRIDE	0 1	ウォッチドッグ・オーバーライド 0 0 をライトしても影響はありません。このビットがクリアされた場合、リセットが発生するまで、この状態のままになります。このビットの現在状態は、ユーザーがリードできます。 1 ウォッチドッグ制御 (WDCR) レジスタでウォッチドッグ・ディスエーブル (WDDIS) ビットの状態を変更できます。1 のライトによって WDOVERRIDE ビットがクリアされた場合は、WDDIS ビットを変更することはできません。

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

図 3-16 ウォッチドッグ・カウンタ・レジスタ (WDCNTR)

15	8	7	0
予約		WDCNTR	
R-0		R-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-15 ウォッチドッグ・カウンタ・レジスタ (WDCNTR) のフィールドの説明

ビット	フィールド	説明
15-8	予約	予約
2	WDCNTR	これらのビットには、WD カウンタの現在値になります。8 ビット・カウンタは、ウォッチドッグ・クロック (WDCLK) レートで継続的にインクリメントします。カウンタがオーバーフローした場合は、ウォッチドッグがリセットを開始します。WDKEY レジスタが有効な組み合わせでライトされた場合は、カウンタがゼロにリセットされます。ウォッチドッグ・クロック・レートは、WDCR レジスタで設定されます。

図 3-17 ウォッチドッグ・リセット・キー・レジスタ (WDKEY)

15	8	7	0
予約		WDKEY	
R-0		R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-16 ウォッチドッグ・リセット・キー・レジスタ (WDKEY) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15-8	予約		予約
7-0	WDKEY	0x55 + 0xAA Other value	WDKEY ライト・シーケンスの例については、表 3-13 を参照してください。 0x55 の後に 0xAA を WDKEY にライトすると、WDCNTR ビットがクリアされます。 0x55 または 0xAA 以外の値をライトすると、何のアクションも生成されません。0x55 の後に 0xAA 以外の値がライトされた場合、0x55 からのシーケンスを再開しなければなりません。 WDKEY からのリードは、WDCR レジスタの値を返します。

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

図 3-18 ウォッチドッグ制御レジスタ (WDCR)

15					8
予約					
7	6	5	3	2	0
WDFLAG	WDDIS	WDCHK		WDPS	
R/W1C-0	R/W-0	R/W-0		R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-17 ウォッチドッグ制御レジスタ (WDCR) のフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
15-8	予約		予約
7	WDFLAG	0 1	ウォッチドッグ・リセット・ステータス・フラグ・ビット リセットは、XRS ピンによって、または電源投入が原因で引き起こされています。1 をライトして状態をクリアするまで、ビットはラッチされた状態のままになります。0 のライトは無視されます。 ウォッチドッグ・リセット (WDRST) がリセット状態を生成したことを示します。

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

表 3-17. ウォッチドッグ制御レジスタ (WDCR) のフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾
6	WDDIS	0 1	ウォッチドッグ・ディスエーブル。リセットでは、ウォッチドッグ・モジュールはイネーブルにされます。 ウォッチドッグ・モジュールをイネーブルにします。WDDIS は、SCSR2 レジスタの WDOVERRIDE ビットが 1 に設定されている場合にのみ変更できます。(デフォルト) ウォッチドッグ・モジュールをディスエーブルにします。
5-3	WDCHK	0,0,0 その他	ウォッチドッグ・チェック。 目的がソフトウェアを介してデバイスをリセットすることでない限り、このレジスタへのライトが実行されるたび、常に「1,0,1」をこれらのビットにライトする必要があります。 ウォッチドッグがイネーブルになっている場合は、他の値をライトすると、即時のデバイス・リセットまたはウォッチドッグ割り込みが行われます。これらの 3 ビットは、常にゼロ (0, 0, 0) としてリードバックします。この機能を使用すると、DSP のソフトウェア・リセットを生成できます。
2-0	WDPS	000 001 010 011 100 101 110 111	ウォッチドッグ・プリスケール。これらのビットは、OSCCLK/512 を基準としてウォッチドッグ・カウンタ・クロック (WDCLK) レートを設定します。 WDCLK = OSCCLK/512/1 (デフォルト) WDCLK = OSCCLK/512/1 WDCLK = OSCCLK/512/2 WDCLK = OSCCLK/512/4 WDCLK = OSCCLK/512/8 WDCLK = OSCCLK/512/16 WDCLK = OSCCLK/512/32 WDCLK = OSCCLK/512/64

XRS ラインが Low の場合、WDFLAG ビットは強制的に Low になります。WDFLAG ビットは、WDRST 信号の立ち上がりエッジで (同期と 8192 SYSCLKOUT サイクルの遅延の後に) 検出され、XRS 信号が High である場合にのみセットされます。WDRST が High になるときに XRS 信号が Low の場合は、WDFLAG ビットは 0 のままになります。一般的なアプリケーションでは、WDRST 信号は XRS 入力につながります。したがって、ウォッチドッグ・リセットと外部デバイス・リセットを区別するためには、外部リセットの期間がウォッチドッグ・パルスよりも長くなければなりません。

3.5 32 ビット CPU タイマ 0/1/2

このセクションでは、280x デバイス上の 3 つの 32 ビット CPU タイマ (表 3-19) について説明します (TIMER0/1/2)。

CPU タイマ 1 および 2 は、TI ソフトウェア用 (DSP-BIOS など) に予約されています。CPU タイマ 0 は、ユーザー・アプリケーションで使用できます。

280x デバイスでは、CPU タイマ割り込み信号 (TINT0、TINT1、TINT2) は、表 3-20 に示したように接続されます。

図 3-19 CPU タイマ

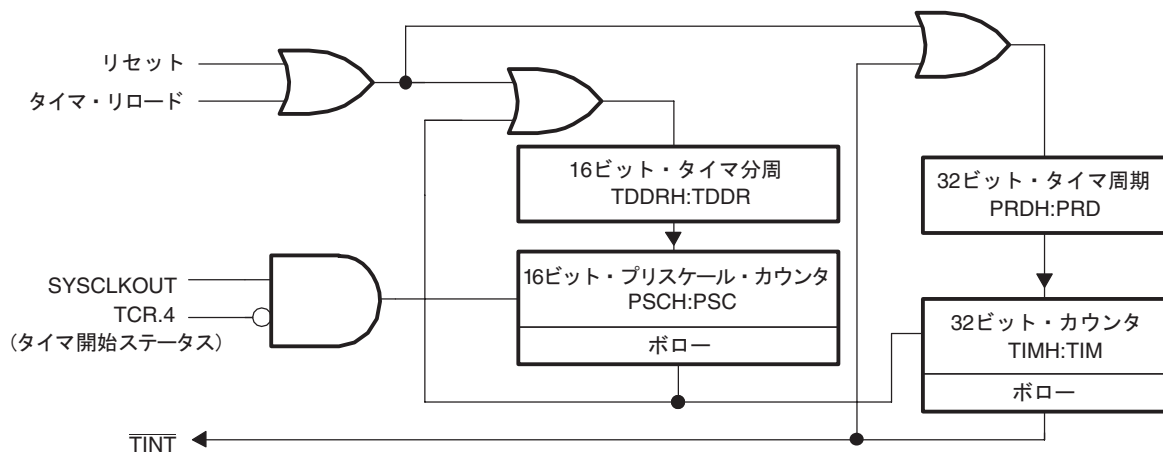
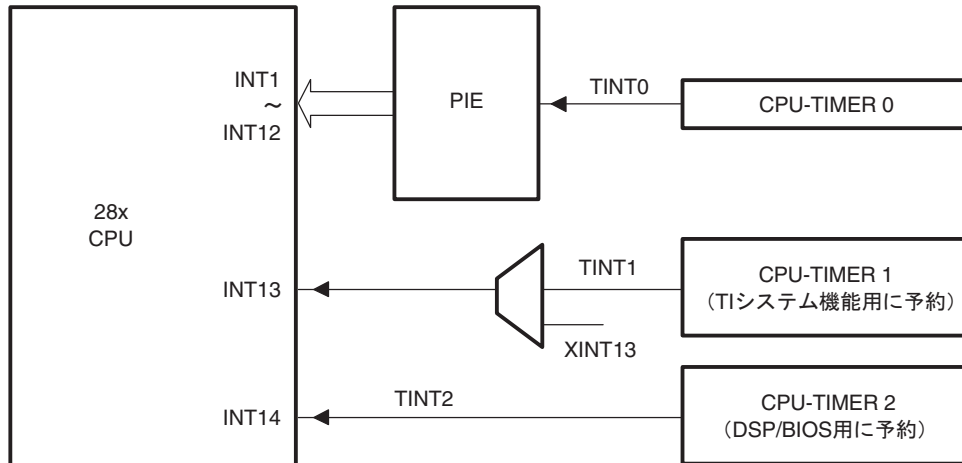


図 3-20 CPU タイマ割り込み信号および出力信号



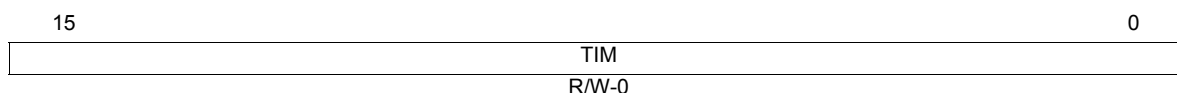
- A タイマ・レジスタは、28x プロセッサのメモリ・バスに接続されます。
- B タイマのタイミングは、プロセッサ・クロックの SYSCLKOUT に同期されます。

CPU タイマの一般的な動作は、以下のとおりです。32 ビット・カウンタ・レジスタ TIMH:TIM には、周期レジスタ PRDH:PRD の値がロードされます。カウンタ・レジスタは、28x の SYSCLKOUT レートでデクリメントします。カウンタが 0 に達すると、タイマ割り込み出力信号が割り込みパルスを生成します。表 3-18 に記載されているレジスタは、タイマを設定するために使用されます。

表 3-18 CPU タイマ 0、1、2 設定および制御レジスタ

名前	アドレス	サイズ (x16)	説明	ビットの説明
TIMER0TIM	0x0C00	1	CPU タイマ 0、カウンタ・レジスタ	図 3-21
TIMER0TIMH	0x0C01	1	CPU タイマ 0、カウンタ・レジスタ High	図 3-22
TIMER0PRD	0x0C02	1	CPU タイマ 0、周期レジスタ	図 3-23
TIMER0PRDH	0x0C03	1	CPU タイマ 0、周期レジスタ High	図 3-24
TIMER0TCR	0x0C04	1	CPU タイマ 0、制御レジスタ	図 3-25
予約	0x0C05	1		
TIMER0TPR	0x0C06	1	CPU タイマ 0、プリスケール・レジスタ	図 3-26
TIMER0TPRH	0x0C07	1	CPU タイマ 0、プリスケール・レジスタ High	図 3-27
TIMER1TIM	0x0C08	1	CPU タイマ 1、カウンタ・レジスタ	図 3-21
TIMER1TIMH	0x0C09	1	CPU タイマ 1、カウンタ・レジスタ High	図 3-22
TIMER1PRD	0x0C0A	1	CPU タイマ 1、周期レジスタ	図 3-23
TIMER1PRDH	0x0C0B	1	CPU タイマ 1、周期レジスタ High	図 3-24
TIMER1TCR	0x0C0C	1	CPU タイマ 1、制御レジスタ	図 3-25
予約	0x0C0D	1		
TIMER1TPR	0x0C0E	1	CPU タイマ 1、プリスケール・レジスタ	図 3-26
TIMER1TPRH	0x0C0F	1	CPU タイマ 1、プリスケール・レジスタ High	図 3-27
TIMER2TIM	0x0C10	1	CPU タイマ 2、カウンタ・レジスタ	図 3-21
TIMER2TIMH	0x0C11	1	CPU タイマ 2、カウンタ・レジスタ High	図 3-22
TIMER2PRD	0x0C12	1	CPU タイマ 2、周期レジスタ	図 3-23
TIMER2PRDH	0x0C13	1	CPU タイマ 2、周期レジスタ High	図 3-24
TIMER2TCR	0x0C14	1	CPU タイマ 2、制御レジスタ	図 3-25
予約	0x0C15	1		
TIMER2TPR	0x0C16	1	CPU タイマ 2、プリスケール・レジスタ	図 3-26
TIMER2TPRH	0x0C17	1	CPU タイマ 2、プリスケール・レジスタ High	図 3-27

図 3-21 TIMERxTIM レジスタ (x=1, 2, 3)

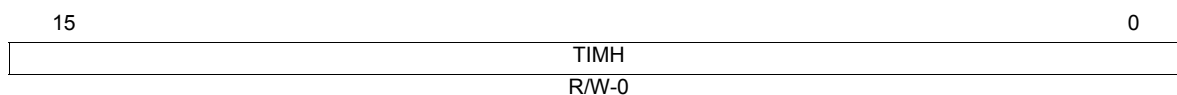


凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-19 TIMERxTIM レジスタのフィールドの説明

ビット	フィールド	説明
15-0	TIM	CPU タイマ・カウンタ・レジスタ (TIMH:TIM) : TIM レジスタは、タイマの現在の 32 ビット・カウンタの下位 16 ビットを保持します。TIMH レジスタは、タイマの現在の 32 ビット・カウンタの上位 16 ビットを保持します。TIMH:TIM は、(TDDR: TDDR+1) クロック・サイクルごとに 1 ずつデクリメントします。TDDR: TDDR はタイマ分周値です。TIMH:TIM がゼロまでデクリメントすると、TIMH:TIM レジスタには、PRDH:PRD レジスタに含まれている周期値がリロードされます。タイマ割り込み (TINT) 信号が生成されます。

図 3-22 TIMERxTIMH レジスタ (x=1, 2, 3)

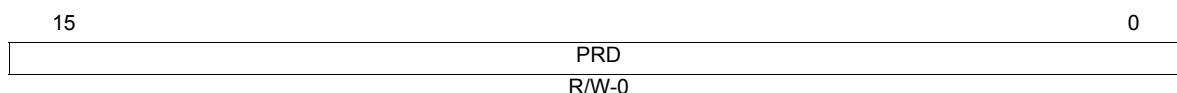


凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-20 TIMERxTIMH レジスタのフィールドの説明

ビット	フィールド	説明
15-0	TIMH	TIMERxTIM の説明を参照してください。

図 3-23 TIMERxPRD レジスタ (x=1, 2, 3)

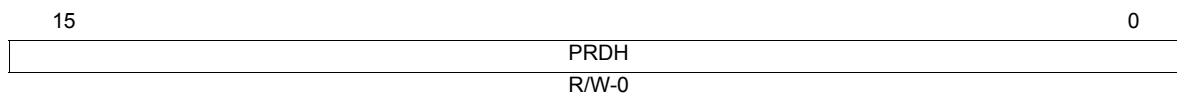


凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-21 TIMERxPRD レジスタのフィールドの説明

ビット	フィールド	説明
15-0	PRD	CPU タイマ周期レジスタ (PRDH:PRD) : PRD レジスタは、32 ビット周期の下位 16 ビットを保持します。PRDH レジスタは、32 ビット周期の上位 16 ビットを保持します。TIMH:TIM がゼロまでデクリメントすると、次のタイマ入カクロック・サイクル (プリスケアラの出力) の開始時に、PRDH:PRD レジスタに含まれている周期値が TIMH:TIM レジスタにリロードされます。タイマ制御レジスタ (TCR) のタイマ・リロード・ビット (TRB) をセットすると、PRDH:PRD の内容が TIMH:TIM にもロードされます。

図 3-24 TIMERxPRDH レジスタ (x=1, 2, 3)



凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-22 TIMERxPRDH レジスタのフィールドの説明

ビット	フィールド	説明
15-0	PRDH	TIMERxPRD の説明を参照してください。

図 3-25 TIMERxTCR レジスタ (x=1, 2, 3)

15	14	13	12	11	10	9	8
TIF	TIE	予約	FREE	SOFT	予約		
R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0		
7	6	5	4	3	2		0
予約	TRB	TSS	予約				
R-0	R/W-0	R/W-0	R-0				

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 3-23 TIMERxTCR レジスタのフィールドの説明

ビット	フィールド	値	説明
15	TIF	0 1	CPU タイマ割り込みフラグ。 CPU タイマは、ゼロまでデクリメントしていません。 0 のライトは無視されます。 このフラグは、CPU タイマがゼロまでデクリメントするとセットされます。 このビットに 1 をライトすると、フラグがクリアされます。
14	TIE	0 1	CPU タイマ割り込みイネーブル。 CPU タイマ割り込みがディスエーブルにされます。 CPU タイマ割り込みがイネーブルにされます。タイマがゼロまでデクリメントし、TIE がセットされた場合は、タイマは割り込みリクエストをアサートします。
13-12	予約		予約
11-10	FREE SOFT	FREE SOFT 0 0 0 1 1 0 1 1	CPU タイマ・エミュレーション・モード：これらのビットは、デバグがブレークポイントが検出されたときのタイマの状態を判断する特殊なエミュレーション・ビットです。FREE ビットが 1 に設定された場合、ソフトウェア・ブレークポイントでは、タイマは引き続き動作します（つまり、フリー・ラン）。このケースでは、SOFT は無視されます。ただし、FREE が 0 の場合は、SOFT は有効です。このケースでは、SOFT=0 の場合、タイマは TIMH:TIM の次のデクリメント時に停止します。SOFT ビットが 1 の場合は、TIMH:TIM がゼロまでデクリメントしているとタイマが停止します。 CPU タイマ・エミュレーション・モード TIMH:TIM の次のデクリメントの後に停止する（ハード・ストップ） TIMH:TIM が 0 までデクリメントした後に停止する（ソフト・ストップ） フリー・ラン フリー・ラン SOFT STOP モードでは、タイマはシャット・ダウンの前に割り込みを生成します（0 への到達は割り込みを引き起こす条件であるため）。
9-6	予約		予約
5	TRB	0 1	CPU タイマ・リロード・ビット。 TRB ビットは、常にゼロとしてリードされます。0 のライトは無視されます。 1 を TRB にライトすると、TIMH:TIM には PRDH:PRD の値がロードされ、プリスケアラ・カウンタ (PSCH:PSC) にはタイマ分周レジスタ (TDDR:TDDBR) の値がロードされます。
4	TSS	0 1	CPU タイマ停止ステータス・ビット。TSS は、CPU タイマを停止または開始する 1 ビットのフラグです。 0 のリードは、CPU タイマが動作中であることを示します。 CPU タイマを開始または再開するには、TSS を 0 に設定します。リセット時には、TSS は 0 にクリアされ、CPU タイマはすぐに開始します。 1 のリードは、CPU タイマが停止されることを示します。 CPU タイマを停止するには、TSS を 1 に設定します。
3-0	予約		予約

図 3-26 TIMERxTPR レジスタ (x=1、2、3)

15	8	7	0
PSC		TDDR	
R-0		R/W-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-24 TIMERxTPR レジスタのフィールドの説明

ビット	フィールド	説明
15-8	PSC	CPU タイマ・プリスケール・カウンタ。これらのビットは、タイマの現在のプリスケール・カウントを保持します。PSCH:PSC 値が 0 より大きいタイマ・クロック・ソース・サイクルごとに、PSCH:PSC が 1 ずつデクリメントします。PSCH:PSC が 0 に達した 1 タイマ・クロック (タイマ・プリスケール出力) サイクル後には、PSCH:PSC に TDDRH:TDDR の内容がロードされ、タイマ・カウンタ・レジスタ (TIMH:TIM) が 1 ずつデクリメントします。また、PSCH:PSC は、タイマ・リロード・ビット (TRB) がソフトウェアによってセットされるたびにリロードされます。PSCH:PSC は、レジスタのリードによってチェックできますが、直接設定することはできません。PSCH:PSC はタイマ分周レジスタ (TDDRH:TDDR) から値を取得する必要があります。リセット時には、PSCH:PSC は 0 に設定されます。
7-0	TDDR	CPU タイマ分周。(TDDRH:TDDR+1) タイマ・クロック・ソース・サイクルごとに、タイマ・カウンタ・レジスタ (TIMH:TIM) は 1 ずつデクリメントします。リセット時には、TDDRH:TDDR ビットは 0 にクリアされます。整数係数の分だけタイマ・カウント全体を増加させるには、この係数から 1 を引いた値を TDDRH:TDDR ビットにライトします。プリスケール・カウンタ (PSCH:PSC) 値が 0 の場合は、1 タイマ・クロック・ソース・サイクル後に、TDDRH:TDDR の内容が PSCH:PSC をリロードし、TIMH:TIM が 1 ずつデクリメントします。また、TDDRH:TDDR は、タイマ・リロード・ビット (TRB) がソフトウェアによってセットされるたびに PSCH:PSC をリロードします。

図 3-27 TIMERxTPRH レジスタ (x=1、2、3)

15	8	7	0
PSCH		TDDRH	
R-0		R/W-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 3-25 TIMERxTPRH レジスタのフィールドの説明

ビット	フィールド	説明
15-8	PSCH	TIMERxTPR の説明を参照してください。
7-0	TDDRH	TIMERxTPR の説明を参照してください。

汎用入力/出力 (GPIO)

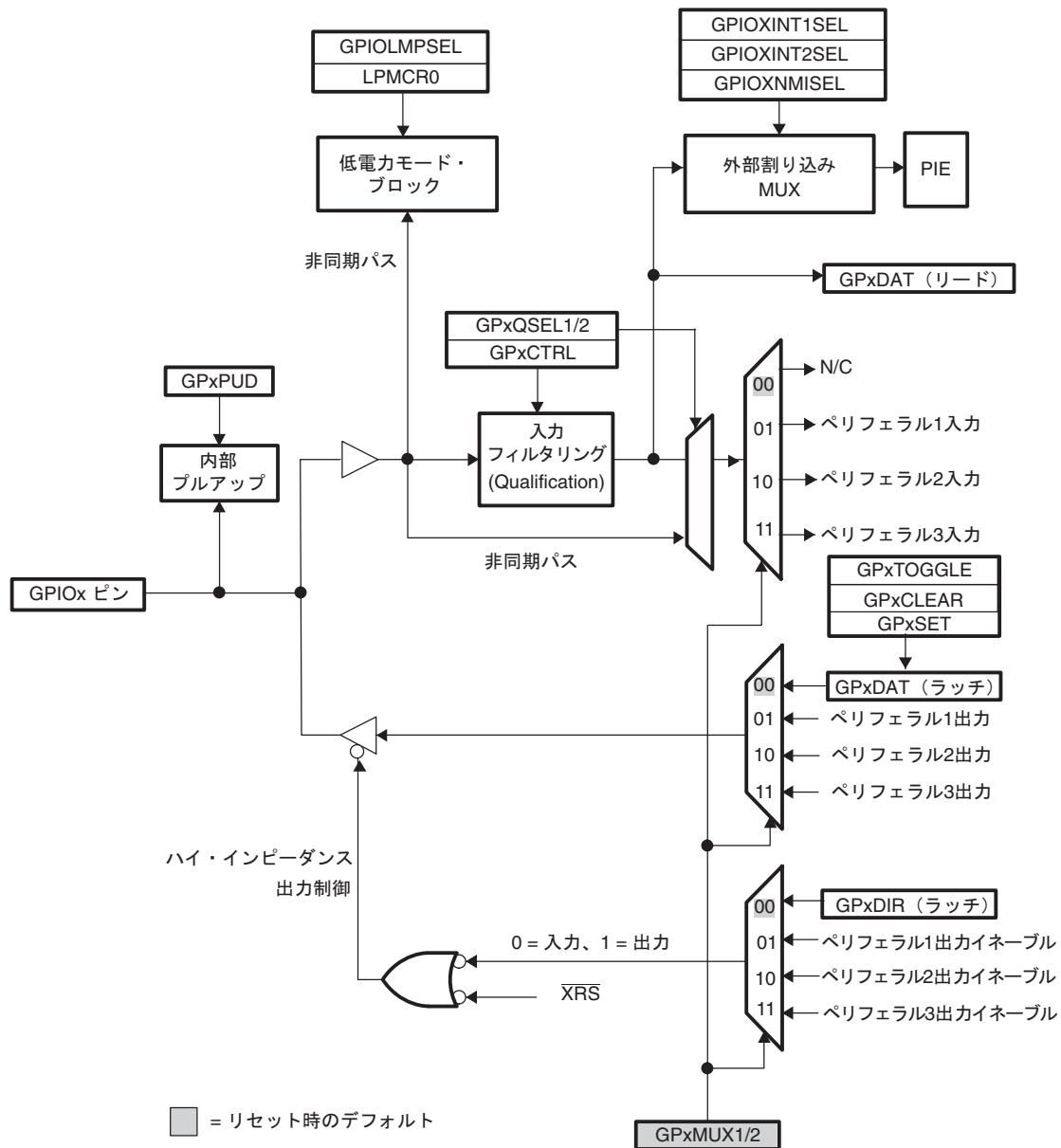
GPIO MUX レジスタは、280x デバイスの共有ピンの動作を選択するために使用されます。ピンには、GPIO0 ~ GPIO34 と名前が付けられています。これらのピンは、個別に選択してデジタル I/O (GPIO と呼ばれる) として動作させるか、または 3 つのペリフェラル I/O 信号のいずれかに (GPAMUX1、GPAMUX2、および GPBMUX1 レジスタで設定) 設定することができます。デジタル I/O モード用に選択した場合は、ピン方向を設定するためにレジスタが (GPADIR および GPBDIR レジスタによって) 提供されます。また、入力信号をフィルタリング (Qualification) して、不要なノイズを (GPAQSEL1、GPAQSEL2、GPACTRL、および GPBCTRL レジスタによって) 取り除くこともできます。

項目	ページ
4.1 GPIO モジュールの概要.....	70
4.2 設定の概要.....	71
4.3 デジタル汎用 I/O 制御.....	73
4.4 入力フィルタリング (Qualification).....	74
4.5 GPIO およびペリフェラル・マルチプレクス.....	78
4.6 レジスタ・ビット定義.....	86

4.1 GPIO モジュールの概要

280x デバイスでは、IO 機能に加えて、最大 3 つの独立したペリフェラル信号が単一の GPIO 対応ピンにマルチプレクスされます。280x デバイスには、2 つの 32 ビット IO ポートがあります。ポート A は GPIO0 ~ GPIO31 で構成され、ポート B は GPIO32 ~ GPIO34 で構成されます。ポート B の残りの IO は、将来の拡張のために現在予約されています。図 4-1 は、GPIO モジュールの動作の基本モードを示します。

図 4-1 動作のモード



- A x はポート (A または B) を表します。たとえば、GPxDIR は、選択された特定の GPIO ピンに応じて GPADIR または GPBDIR レジスタのいずれかを意味します。
 B GPxDAT ラッチ/リードは、同じメモリ・ロケーションでアクセスされます。

4.2 設定の概要

ピン機能割り当て、入力フィルタリング (Qualification)、および外部割り込み (XINT1、XINT2、XNMI) ソースは、すべて GPIO 設定制御レジスタによって制御されます。また、ピンを割り当てて、HALT および STANDBY 低電力モードからデバイスをウェイクし、内部プルアップ・レジスタをイネーブル/ディスエーブルにすることもできます。表 4-1 および表 4-2 には、GPIO ピンを設定してシステム要件に合わせるために使用されるレジスタが記載されています。

表 4-1 GPIO 制御レジスタ

名前 ⁽¹⁾	アドレス	サイズ (x16)	レジスタの説明	ビットの説明
GPACTRL	0x6F80	2	GPIO A 制御レジスタ (GPIO0 ~ GPIO31)	図 4-7
GPAQSEL1	0x6F82	2	GPIO A フィルタ (Qualifier) セレクト 1 レジスタ (GPIO0 ~ GPIO15)	図 4-9
GPAQSEL2	0x6F84	2	GPIO A フィルタ (Qualifier) セレクト 2 レジスタ (GPIO16 ~ GPIO31)	図 4-10
GPAMUX1	0x6F86	2	GPIO A MUX 1 レジスタ (GPIO0 ~ GPIO15)	図 4-4
GPAMUX2	0x6F88	2	GPIO A MUX 2 レジスタ (GPIO16 ~ GPIO31)	図 4-5
GPADIR	0x6F8A	2	GPIO A 方向レジスタ (GPIO0 ~ GPIO31)	図 4-12
GPAPUD	0x6F8C	2	GPIO A プルアップ・ディスエーブル・ レジスタ (GPIO0 ~ GPIO31)	図 4-14
予約	0x6F8E - 0x6F8F	2		
GPBCTRL	0x6F90	2	GPIO B 制御レジスタ (GPIO32 ~ GPIO34)	図 4-8
GPBQSEL1	0x6F92	2	GPIO B フィルタ (Qualifier) セレクト 1 レジスタ (GPIO32 ~ GPIO34)	図 4-11
GPBQSEL2	0x6F94	2	予約	
GPBMUX1	0x6F96	2	GPIO B MUX 1 レジスタ (GPIO32 ~ GPIO34)	図 4-6
GPBMUX2	0x6F98	2	予約	
GPBDIR	0x6F9A	2	GPIO B 方向レジスタ (GPIO32 ~ GPIO34)	図 4-13
GPBPUD	0x6F9C	2	GPIO B プルアップ・ディスエーブル・ レジスタ (GPIO32 ~ GPIO34)	図 4-15
予約	0x6F9E - 0x6FB0	34		

(1) この表にあるレジスタは EALLOW 保護されます。セクション 5.2 を参照してください。

表 4-2 GPIO 割り込み、および低電力モード・セレクト・レジスタ

名前 ⁽¹⁾	アドレス	サイズ (x16)	レジスタの説明	ビットの説明
GPIOXINT1SEL	0x6FE0	1	XINT1 ソース・セレクト・レジスタ (GPIO0 ~ GPIO31)	図 4-20
GPIOXINT2SEL	0x6FE1	1	XINT2 ソース・セレクト・レジスタ (GPIO0 ~ GPIO31)	図 4-20
GPIOXNMISEL	0x6FE2	1	XNMI ソース・セレクト・レジスタ (GPIO0 ~ GPIO31)	図 4-20
予約	0x6FE3 - 0x6FE7	5		
GPIO_LPMSEL	0x6FE8	1	LPM ウェイクアップ・ソース・セレクト・ レジスタ (GPIO0 ~ GPIO31)	図 4-21

(1) この表にあるレジスタは EALLOW 保護されます。詳細については、セクション 5.2 を参照してください。

GPIO モジュールの設定を計画するには、以下のステップについて検討してください。

1. デバイス・ピンアウトを計画する：

ピン・マルチプレクスの仕組みを通じて、280x デバイスは、35 本の GPIO としても使用できるピン (GPIO0 ~ GPIO34) を柔軟に割り当てることができます。作業を始める前に、各ピンに利用できるペリフェラル・オプションを調べて、ユーザー固有のシステムに対するピンアウトを計画してください。汎用入力または出力 (GPIO) として、あるいは利用可能な 3 つのペリフェラル機能のいずれかとしてピンを使用しますか？ピンを設定するためには、この情報を知る必要があります。

2. 内部プルアップ・レジスタをイネーブルまたはディスエーブルにする：

内部プルアップ・レジスタをイネーブルまたはディスエーブルにするには、GPIO プルアップ・ディスエーブルレジスタ (GPAPUD、GPBPUD) で、それぞれのビットにライトします。ePWM 出力ピン (GPIO0 ~ GPIO11) として機能できるピンの場合、内部プルアップ・レジスタはデフォルトでディスエーブルになります。その他の GPIO 対応ピンでは、デフォルトでプルアップがイネーブルになります。

3. 入力フィルタリング (Qualification) を選択する：

ピンが入力として使用される場合は、必要な入力フィルタリング (Qualification) を指定します (存在する場合)。入力フィルタリング (Qualification) は、GPACTRL、GPBCTRL、GPAQSEL1、GPAQSEL2、および GPBQSEL1 レジスタで設定します。デフォルトでは、すべての入力信号が SYSCLKOUT に対してのみ同期されます。

4. ピン機能を選択する：

GPAMUX1、GPAMUX2、および GPBMUX1 レジスタを設定して、ピンが GPIO または利用可能な 3 つのペリフェラル機能のいずれかになるようにします。デフォルトでは、すべての GPIO 対応ピンはリセット時に汎用入力ピンとして設定されます。

5. デジタル汎用 I/O の場合、ピンの方向を選択する：

ピンが GPIO として設定された場合は、GPADIR および GPBDIR レジスタでピンの方向を入力または出力として指定します。デフォルトでは、すべての GPIO ピンが入力です。ピンを入力から出力に変更するには、GPACLEAR、GPBCLEAR、GPASET、GPBSET、GPATOGGLE、または GPBTOGGLE レジスタに適切な値をライトすることによってドライブされる値を最初に出カラッチにロードします。出カラッチがロードされたら、GPADIR および GPBDIR レジスタでピン方向を入力から出力に変更します。すべてのピンの出カラッチは、リセット時にクリアされます。

6. 低電力モード・ウェイクアップ・ソースを選択する：

どのピン (存在する場合) が HALT および STANDBY 低電力モードからデバイスをウェイクすることができるのかを指定します。ピンは、GPIOLPMSEL レジスタで設定されます。

7. 外部割り込みソースを選択する：

XINT1、XINT2、および XNMI 割り込みのソースを指定します。割り込みごとに、いずれかのポート A 信号をソースとして指定できます。これは、GPIOXINT1SEL、GPIOXINT2SEL、および GPIOXNMISEL でソースを指定することによって行われます。割り込みの極性は、[セクション 6.6](#) で説明しているように、XINT1CR、XINT2CR、および XNMICR レジスタで設定できます。

4.3 デジタル汎用 I/O 制御

GPIO として設定されるピンについては、以下のレジスタによってピン上の値を変更できます。

表 4-3 GPIO データ・レジスタ

名前	アドレス	サイズ (x16)	レジスタの説明	ビットの説明
GPADAT	0x6FC0	2	GPIO A データ・レジスタ (GPIO0 ~ GPIO31)	図 4-16
GPASET	0x6FC2	2	GPIO A セット・レジスタ (GPIO0 ~ GPIO31)	図 4-18
GPACLEAR	0x6FC4	2	GPIO A クリア・レジスタ (GPIO0 ~ GPIO31)	図 4-18
GPATOGGLE	0x6FC6	2	GPIO A トグル・レジスタ (GPIO0 ~ GPIO31)	図 4-18
GPBDAT	0x6FC8	2	GPIO B データ・レジスタ (GPIO32 ~ GPIO34)	図 4-17
GPBSET	0x6FCA	2	GPIO B セット・レジスタ (GPIO32 ~ GPIO34)	図 4-19
GPBCLEAR	0x6FCC	2	GPIO B クリア・レジスタ (GPIO32 ~ GPIO34)	図 4-19
GPBTOGGLE	0x6FCE	2	GPIO B トグル・レジスタ (GPIO32 ~ GPIO34)	図 4-19
予約	0x70FC - 0x70FF	4		

• **GPADAT、GPBDAT レジスタ :**

各 I/O ポートには 1 つのデータ・レジスタがあります。データ・レジスタの各ビットは、1 つの GPIO ピンに対応しています。どのようにピンが設定されていても (GPIO またはペリフェラル機能)、データ・レジスタ内の対応するビットは、フィルタリング (Qualification) 後のピンの現在状態を反映します。GPADAT または GPBDAT レジスタへのライトは、対応する出力ラッチをクリアまたはセットします。また、ピンが汎用出力 (GPIO 出力) としてイネーブルになっている場合は、そのピンも Low または High のいずれかにドライブされます。ピンが GPIO 出力として設定されていない場合は、値はラッチされますが、ピンはドライブされません。ピンが後で GPIO 出力として設定された場合にのみ、ラッチされた値がピンへドライブされます。

GPxDAT レジスタを使用して出力ピンのレベルを変更するときには、別のピンのレベルを誤って変更しないように注意する必要があります。たとえば、read-modify-write 命令を使用して、GPADAT レジスタのビット 0 にライトすることによって GPIOA0 の出力ラッチ・レベルを変更する場合です。別の I/O ポート A 信号が命令のリード段階とライト段階の間でレベルを変更した場合は、問題が発生する可能性があります。また、その出力ラッチの状態を変更することもできます。このシナリオは、GPxSET、GPxCLEAR、および GPxTOGGLE レジスタを使用して代わりに出力ラッチをロードすることによって回避できます。

• **GPASET、GPBSET レジスタ :**

セット・レジスタは、他のピンを妨害することなく特定の GPIO ピンを High にドライブするために使用されます。各 I/O ポートには 1 つのセット・レジスタがあり、各ビットは 1 つの GPIO ピンに対応します。セット・レジスタは、常に 0 をリード・バックします。対応するピンが出力として設定されている場合は、セット・レジスタ内の対応するビットに 1 をライトすると、出力ラッチは High に設定され、対応するピンは High にドライブされます。ピンが GPIO 出力として設定されていない場合は、値はラッチされますが、ピンはドライブされません。ピンが後で GPIO 出力として設定された場合にのみ、ラッチされた値がピンへドライブされます。セット・レジスタ内の任意のビットに 0 をライトしても影響はありません。

• **GPACLEAR、GPBCLEAR レジスタ :**

クリア・レジスタは、他のピンを妨害することなく特定の GPIO ピンを Low にドライブするために使用されます。各 I/O ポートには 1 つのクリア・レジスタがあります。クリア・レジスタは、常に 0 をリード・バックします。対応するピンが汎用出力として設定されている場合は、クリア・レジスタ内の対応するビットに 1 をライトすると、出力ラッチはクリアされ、ピンは Low にドライブされます。ピンが GPIO 出力として設定されていない場合は、値はラッチされますが、ピンはドライブされません。ピンが後で GPIO 出力として設定された場合にのみ、ラッチされた値がピンへドライブされます。クリア・レジスタ内の任意のビットに 0 をライトしても影響はありません。

- **GPATOGGLE、GPBTOGGLE レジスタ :**

トグル・レジスタは、他のピンを妨害することなく特定の GPIO ピンを反対のレベルにドライブするために使用されます。各 I/O ポートには 1 つのトグル・レジスタがあります。トグル・レジスタは、常に 0 をリード・バックします。対応するピンが出力として設定されている場合は、トグル・レジスタ内の対応するビットに 1 をライトすると、出力ラッチは反転され、対応するピン出力は反転されます。つまり、出力ピンが Low にドライブされている場合は、トグル・レジスタ内の対応するビットに 1 をライトすると、ピンに High が出力されます。同様に、出力ピンが High の場合は、トグル・レジスタ内の対応するビットに 1 をライトすると、ピンに Low が出力されます。ピンが GPIO 出力として設定されていない場合は、値はラッチされますが、ピンはドライブされません。ピンが後で GPIO 出力として設定された場合にのみ、ラッチされた値がピンへドライブされます。トグル・レジスタ内の任意のビットに 0 をライトしても影響はありません。

4.4 入力フィルタリング (Qualification)

入力フィルタリング (Qualification) は、280x デバイスに搭載されている非常に柔軟な機能です。GPAQSEL1、GPAQSEL2、および GPBQSEL1 レジスタの設定によって、GPIO ピンごとに入力フィルタリング (Qualification) のタイプを選択できます。GPIO 入力ピンの場合、フィルタリング (Qualification) は、SYSCLKOUT に対する同期のみ、またはサンプリング・ウィンドウによるフィルタリング (Qualification) として指定できます。ペリフェラル入力として設定されるピンについては、SYSCLKOUT に対する同期またはサンプリング・ウィンドウによるフィルタリング (Qualification) のほかに、入力を非同期にすることもできます。このセクションの残りでは、利用可能なオプションについて説明します。

4.4.1 同期なし (非同期入力) :

このモードは、入力同期が必要でない場合やペリフェラル自体が同期を実行するペリフェラルに対して使用されます。例えば、SCI、SPI、eCAN、および I2C 等の通信ポートがこれに該当します。また、ePWM トリップ・ゾーン (TZ1 ~ TZ6) 機能のように SYSCLKOUT とは独立した方が望ましい場合もあります。

ピンが汎用デジタル入力ピン (GPIO) として使用される場合は、非同期オプションは無効です。ピンが GPIO 入力として設定され、非同期オプションが選択された場合は、[セクション 4.4.2](#) で説明するように、フィルタリング (Qualification) はデフォルトで SYSCLKOUT に対する同期になります。

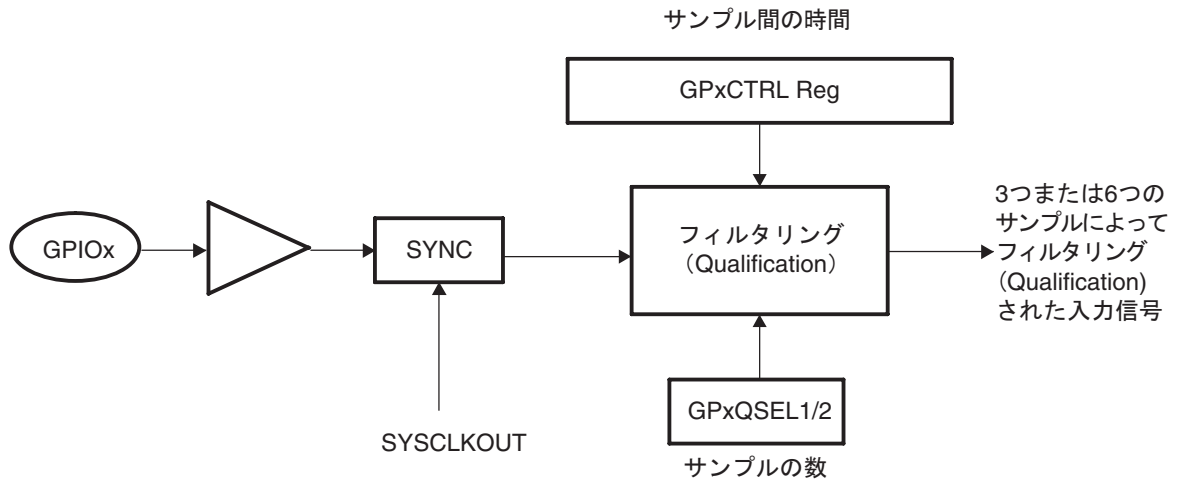
4.4.2 SYSCLKOUT に対する同期のみ :

これは、リセット時のすべてのピンのデフォルト・フィルタリング (Qualification) モードです。このモードでは、入力信号はシステム・クロック (SYSCLKOUT) に対してのみ同期されます。入力信号は非同期であるため、DSP への入力に変更されるために最大で SYSCLKOUT 周期の遅延が発生することがあります。信号に対して、さらなるフィルタリング (Qualification) は実行されません。

4.4.3 サンプリング・ウィンドウを使用したフィルタリング (Qualification) :

このモードでは、信号は最初にシステム・クロック (SYSCLKOUT) に対して同期され、その後で、入力の変更が許可される前に、指定された数のサイクルによってフィルタリング (Qualification) されます。[図 4-2](#) と [図 4-3](#) は、どのように入力フィルタリング (Qualification) を実行して不要なノイズを除去するのを示します。このタイプのフィルタリング (Qualification) については、2 つのパラメータ (1) サンプリング周期、または信号がサンプリングされる頻度、および 2) 取得されるサンプルの数。) がユーザーによって設定できます。

図 4-2 サンプリング・ウィンドウを使用した入力フィルタリング (Qualification)



サンプル間の時間 (サンプリング周期) :

信号をフィルタリング (Qualification) するために、入力信号が定期的にサンプリングされます。サンプリング周期は、ユーザーによって指定され、CPU クロック (SYSCLKOUT) を基準としてサンプル間の時間または信号がサンプリングされる頻度を決定します。

サンプリング周期は、GPACTRL または GPBCTRL レジスタのフィルタリング (Qualification) 周期 (QUALPRDn) ビットによって指定されます。サンプリング周期は、8 つの入力信号のグループで設定可能でした例えば、GPIO0 ~ GPIO7 は GPACTRL[QUALPRD0] 設定を使用し、GPIO8 ~ GPIO15 は GPACTRL[QUALPRD1] を使用します。表 4-4 と表 4-5 は、サンプリング周期またはサンプリング周波数と GPxCTRL[QUALPRDn] 設定の間の関係を示します。

表 4-4 サンプリング周期

	サンプリング周期
GPxCTRL[QUALPRDn]=0 の場合	$1 \times T_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn]≠0 の場合	$2 \times \text{GPxCTRL[QUALPRDn]} \times T_{\text{SYSCLKOUT}}$

$T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の期間です。

表 4-5 サンプリング周波数

	サンプリング周波数
GPxCTRL[QUALPRDn]=0 の場合	$f_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn]≠0 の場合	$f_{\text{SYSCLKOUT}} \times 1 \div (2 \times \text{GPxCTRL[QUALPRDn]})$

$f_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の周波数です。

これらの等式からは、サンプル間の最小時間と最大時間を特定の SYSCLKOUT 周波数について計算できます。

例：最大サンプリング周波数：

GPxCTRL[QUALPRDn]=0 の場合
サンプリング周波数は $f_{\text{SYSCLKOUT}}$ です。
たとえば、 $f_{\text{SYSCLKOUT}}=100\text{MHz}$ の場合
信号は、100MHz で、または 10ns ごとに 1 つのサンプルでサンプリングされます。

例：最小サンプリング周波数：

GPxCTRL[QUALPRDn]=0xFF (つまり、255) の場合
サンプリング周波数は、 $f_{\text{SYSCLKOUT}} \times 1 \div (2 \times \text{GPxCTRL[QUALPRDn]})$ です。
たとえば、 $f_{\text{SYSCLKOUT}}=100\text{MHz}$ の場合
信号は、 $100\text{MHz} \times 1 \div (2 \times 255)$ で、または 5.1uS ごとに 1 つのサンプルでサンプリングされます。

サンプルの数：

信号がサンプリングされる回数は、フィルタリング (Qualification) 選択 (GPAQSEL1、GPAQSEL2、および GPBQSEL1) レジスタで設定でき、3 サンプルまたは 6 サンプルのいずれかです。3 または 6 つの連続したサイクルが同一である場合、入力変更は DSP に渡されます。

合計サンプリング・ウィンドウ幅：

サンプリング・ウィンドウは、図 4-3 に示すように入力信号がサンプリングされる時間です。取得されるサンプルの数と共にサンプリング周期の等式を使用することにより、ウィンドウの合計幅を決定できます。

入力フィルタ (Qualifier) が入力の変化を検出するためには、信号のレベルがサンプリング・ウィンドウ幅以上の時間にわたって安定していなければなりません。

ウィンドウ内のサンプリング周期の数は、取得されるサンプルの数よりも常に 1 だけ少なくなります。3 サンプル・ウィンドウについては、サンプリング周期が表 4-4 で定義される場合、サンプリング・ウィンドウ幅は 2 サンプル周期の幅です。同様に、6 サンプル・ウィンドウについては、サンプリング・ウィンドウ幅は 5 サンプル周期の幅です。表 4-6 と表 4-7 は、GPxCTRL[QUALPRDn] と取得されるサンプルの数に基づいて合計サンプリング・ウィンドウ幅を決定する場合に使用できる計算を示します。

表 4-6 ケース 1：3 サンプル・サンプリング・ウィンドウ幅

	合計サンプリング・ウィンドウ幅
GPxCTRL[QUALPRDn]=0 の場合	$2 \times T_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn]≠0 の場合	$2 \times 2 \times \text{GPxCTRL[QUALPRDn]} \times T_{\text{SYSCLKOUT}}$

$T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の期間です。

表 4-7 ケース 2：6 サンプル・サンプリング・ウィンドウ幅

	合計サンプリング・ウィンドウ幅
GPxCTRL[QUALPRDn]=0 の場合	$5 \times T_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn]≠0 の場合	$5 \times 2 \times \text{GPxCTRL[QUALPRDn]} \times T_{\text{SYSCLKOUT}}$

$T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の期間です。

注： 外部信号の変化は、サンプリング周期と SYSCLKOUT の両方に関して非同期です。外部信号の非同期性のため、ロジックが信号の変化を検出することを保証するには、入力をサンプリング・ウィンドウ幅よりも長い時間にわたって安定した状態に保持する必要があります。必要となる追加の時間は、最大で追加サンプリング周期 + $T_{\text{SYSCLKOUT}}$ になる可能性があります。

フィルタリング (Qualification) ロジックが変化を検出するために必要となる、入力信号が安定している時間は、デバイス別のデータ・マニュアルに記載されています。

フィルタリング (Qualification) ウィンドウの例 :

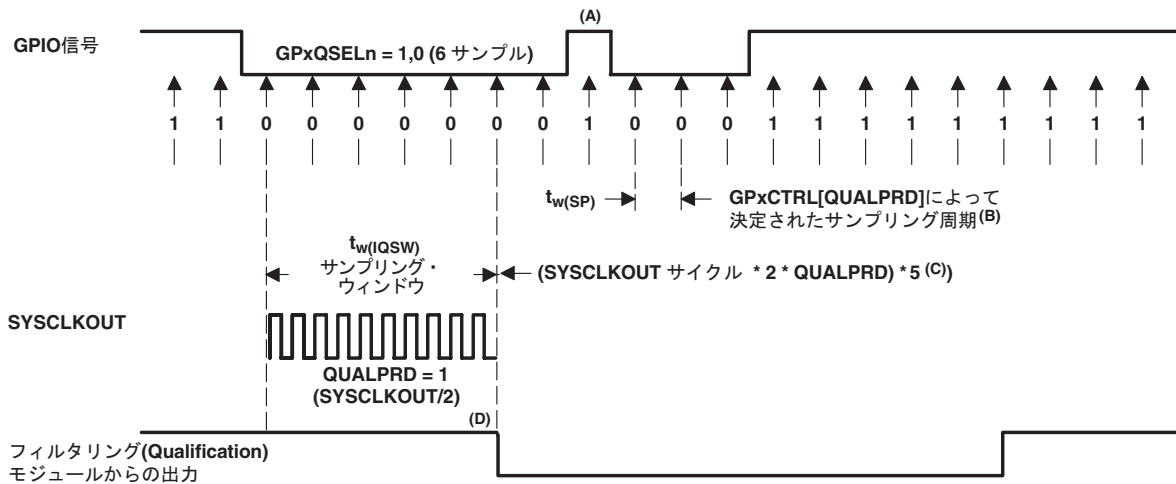
図 4-3 に示した例の場合、入力フィルタリング (Qualification) は以下のように設定されています。

- GPxQSEL1/2=1,0。これは、6 サンプル・フィルタリング (Qualification) を示します。
- GPxCTRL[QUALPRDn]=1。サンプリング周期は、 $t_w(SP)=2 \times GPxCTRL[QUALPRDn] \times T_{SYSCLKOUT}$ です。

この設定の結果は以下のようになります。

- サンプリング・ウィンドウの幅は :
 $t_w(IQSW) = 5 \times t_w(SP) = 5 \times 2 \times GPxCTRL[QUALPRDn] \times T_{SYSCLKOUT}$ or $5 \times 2 \times T_{SYSCLKOUT}$
- たとえば、 $T_{SYSCLKOUT}=10ns$ の場合、サンプリング・ウィンドウの持続時間は :
 $t_w(IQSW) = 5 \times 2 \times 10ns = 100ns$ 。
- サンプリング周期と SYSCLKOUT に対する入力の非同期性により、入力信号の変化を検出するために最大で追加サンプリング周期 ($t_w(SP)$) + $T_{SYSCLKOUT}$ が必要になることがあります。この例の場合 :
 $t_w(SP) + T_{SYSCLKOUT} = 20ns + 10ns = 30ns$
- 図 4-3 では、グリッチ (A) はフィルタリング (Qualification) ウィンドウよりも短く、入力フィルタ (Qualifier) によって無視されます。

図 4-3 入力フィルタリング (Qualification) クロック・サイクル



- このグリッチは入力フィルタ (Qualifier) によって無視されます。QUALPRD ビット・フィールドは、フィルタリング (Qualification) サンプリング周期を指定します。これは、00 から 0xFF まで変化する可能性があります。QUALPRD=00 の場合、サンプリング周期は 1 SYSCLKOUT サイクルです。その他の値 "n" については、 $2n$ SYSCLKOUT サイクルでのフィルタリング (Qualification) サンプリング周期 (つまり、 $2n$ SYSCLKOUT サイクルごとに、GPIO ピンがサンプリングされる)。
- GPxCTRL レジスタによって選択されたフィルタリング (Qualification) 周期は、8 GPIO ピンのグループに適用されます。
- フィルタリング (Qualification) ブロックは、3 つまたは 6 つのサンプルをとることができます。GPxQSELn レジスタは、どのサンプル・モードを使用するかを選択します。
- 上記例で、フィルタリング (Qualification) モジュールが変化を検出するためには、入力が 10 SYSCLKOUT サイクル以上にわたって安定している必要があります。つまり、入力は $(5 \times QUALPRD \times 2)$ SYSCLKOUT サイクルにわたって安定していなければなりません。それは、検出が発生するための 5 サンプル周期を保証します。外部信号は非同期にドライブされるため、13-SYSCLKOUT 幅のパルスは確実な認識を保証します。

4.5 GPIO およびペリフェラル・マルチプレクス

280x デバイスは、各ピンに対して 1 つの汎用入力 / 出力 (GPIO)、最大 3 つの異なるペリフェラル機能がマルチプレクスされています。これは、各アプリケーションに対する、柔軟なピンの割り当てを可能にします。

表 4-9、表 4-10、および表 4-11 は、GPIO ピンの可能な MUX 組み合わせの概要を示します。2 番目の列は、デバイス上のピンの I/O 名を示します。I/O 名は一意であるため、それは特定のピンを識別する最良の方法です。したがって、このセクションにおけるレジスタ記述は、特定のピンの GPIO 名だけを参照します。各ピンの選択を制御する MUX レジスタおよび特定のビットは、最初の列で示されます。

たとえば、GPIO7 ピンの MUX は、GPAMUX[15:14] へのライトによって制御されます。これらのビットへのライトにより、ピンは、GPIO7 として、または最大 3 つのペリフェラル機能のいずれかとして設定されます。2808 上の GPIO7 ピンは、以下のように設定できます。

GPAMUX1[15:14] ビット設定	選択されたピン機能
GPAMUX1[15:14]=0,0 の場合	GPIO7 として設定されたピン
GPAMUX1[15:14]=0,1 の場合	EPWM4B (O) として設定されたピン
GPAMUX1[15:14]=1,0 の場合	SPISTED (I/O) として設定されたピン
GPAMUX1[15:14]=1,1 の場合	ECAP2 (I/O) として設定されたピン

280x ファミリのすべてのデバイスは、同じマルチプレクスの仕組みを持っています。唯一の違いは、ペリフェラルが特定のデバイスで利用できない場合、その MUX 選択はそのデバイス上で予約され、使用すべきではないということです。

注： ペリフェラルにマップされていない予約済みの GPIO MUX 設定を選択する必要がある場合は、ピンの状態は不定となり、そのピンがドライブされる可能性があります。予約された設定は将来の拡張用であり、選択すべきではありません。デバイス MUX 表 (表 4-9、表 4-10、および表 4-11) では、これらのオプションは「予約」として示されます。

いくつかのペリフェラルは、MUX レジスタを介して複数のピンに割り当てることができます。たとえば、以下に示すように、CAP1 機能は、個々のシステム要件に応じて GPIO5 または GPIO24 ピンのいずれかに割り当てることができます。

CAP1 に割り当てられたピン	MUX 設定
選択 1 GPIO5	GPAMUX[11:10] = 1,1
または、選択 2 GPIO24	GPAMUX2[17:16] = 0,1

どのピンもペリフェラルへの入力として設定されていない場合や、複数のピンが同じペリフェラルの入力として設定されている場合は、表 4-8 に示すように、ペリフェラルへの入力はデフォルトで 0 または 1 のいずれかになります。たとえば、ECAP1 が GPIO5 と GPIO24 の両方に割り当てられた場合は、表 4-8 に示すように、eCAP1 ペリフェラルへの入力はデフォルトで High 状態になり、その入力は GPIO5 または GPIO24 に接続されません。

表 4-8 ペリフェラル入力のデフォルト状態

ペリフェラル入力	説明	デフォルト入力 ⁽¹⁾
TZ1-TZ6	トリップ・ゾーン 1 ~ 6	1
EPWMSYNCI	ePWM 同期入力	0
ECAP1-ECAP4	eCAP1 ~ 4 入力	1
EQEP1A, EQEP2A	eQEP 入力	1
EQEP1I, EQEP2I	eQEP インデックス	1
EQEP1S, EQEP2S	eQEP ストローブ	1
SPICLKA - SPICLKD	SPI-A - SPI-D クロック	1
SPISTE A - SPISTED	SPI-A - SPI-D 送信イネーブル	0
SPISIMOA - SPISIMOD	SPI-A - SPI-D スレーブイン、マスターアウト	1
SPISOMIA - SPISOMID	SPI-A - SPI-D スレーブアウト、マスターイン	1
SCIRXDA - SCIRXDB	SCI-A - SCI-B 受信	1
CANRXA - CANRXB	eCAN-A - eCAN-B 受信	1
SDAA	I2C データ	1
SCLA1	I2C クロック	1

(1) 複数のピンが GPxMUX1/2 レジスタでペリフェラル機能に割り当てられている場合や、どのピンも割り当てられていない場合は、この値がペリフェラル入力に割り当てられます。

表 4-9、表 4-10、および表 4-11 は、それぞれ 2808、2806、および 2801/2802 デバイスの MUX オプションを示します。表 4-12 は、ペリフェラルごとに並び替えた表のソートを提供します。この表を使用すると、特定のペリフェラル機能に割り当てることが可能な GPIO ピンをすぐに探すことができます。

表 4-9 2808 GPIO MUX 表

GPAMUX1/2 ⁽¹⁾ レジスタ・ビット	リセット時のデフォルト			
	基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
GPAMUX1				
1-0	GPIO0	EPWM1A (O)	予約 ⁽²⁾	予約 ⁽²⁾
3-2	GPIO1	EPWM1B (O)	SPISIMOD (I/O)	予約 ⁽²⁾
5-4	GPIO2	EPWM2A (O)	予約 ⁽²⁾	予約 ⁽²⁾
7-6	GPIO3	EPWM2B (O)	SPISOMID (I/O)	予約 ⁽²⁾
9-8	GPIO4	EPWM3A (O)	予約 ⁽²⁾	予約 ⁽²⁾
11-10	GPIO5	EPWM3B (O)	SPICLKD (I/O)	ECAP1 (I/O)
13-12	GPIO6	EPWM4A (O)	EPWMSYNCI (I)	EPWMSYNCO (O)
15-14	GPIO7	EPWM4B (O)	SPISTED (I/O)	ECAP2 (I/O)
17-16	GPIO8	EPWM5A (O)	CANTXB (O)	ADCSOAO (O)
19-18	GPIO9	EPWM5B (O)	SCITXDB (O)	ECAP3 (I/O)
21-20	GPIO10	EPWM6A (O)	CANRXB (I)	ADCSOBCO (O)
23-22	GPIO11	EPWM6B (O)	SCIRXDB (I)	ECAP4 (I/O)
25-24	GPIO12	TZ1 (I)	CANTXB (O)	SPISIMOB (I/O)
27-26	GPIO13	TZ2 (I)	CANRXB (I)	SPISOMIB (I/O)
29-28	GPIO14	TZ3 (I)	SCITXDB (O)	SPICLKB (I/O)
31-30	GPIO15	TZ4 (I)	SCIRXDB (I)	SPISTEB (I/O)
GPAMUX2				
1-0	GPIO16	SPISIMOA (I/O)	CANTXB (O)	TZ5 (I)
3-2	GPIO17	SPISOMIA (I/O)	CANRXB (I)	TZ6 (I)
5-4	GPIO18	SPICLKA (I/O)	SCITXDB (O)	予約 ⁽²⁾
7-6	GPIO19	SPISTE A (I/O)	SCIRXDB (I)	予約 ⁽²⁾

GPAMUX1/2 ⁽¹⁾ レジスタ・ビット	リセット時のデフォルト			
	基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
9-8	GPIO20	EQEP1A (I)	SPISIMOC (I/O)	CANTXB (O)
11-10	GPIO21	EQEP1B (I)	SPISOMIC (I/O)	CANRXB (I)
13-12	GPIO22	EQEP1S (I/O)	SPICLK (I/O)	SCITXDB (O)
15-14	GPIO23	EQEP1I (I/O)	SPISTEC (I/O)	SCIRXDB (I)
17-16	GPIO24	ECAP1 (I/O)	EQEP2A (I)	SPISIMOB (I/O)
19-18	GPIO25	ECAP2 (I/O)	EQEP2B (I)	SPISOMIB (I/O)
21-20	GPIO26	ECAP3 (I/O)	EQEP2I (I/O)	SPICLKB (I/O)
23-22	GPIO27	ECAP4 (I/O)	EQEP2S (I/O)	SPISTEB (I/O)
25-24	GPIO28	SCIRXDA (I)	予約 ⁽²⁾	TZ5 (I)
27-26	GPIO29	SCITXDA (O)	予約 ⁽²⁾	TZ6 (I)
29-28	GPIO30	CANRXA (I)	予約 ⁽²⁾	予約 ⁽²⁾
31-30	GPIO31	CANTXA (O)	予約 ⁽²⁾	予約 ⁽²⁾
GPBMUX1				
1-0	GPIO32	SDAA (I/O)	EPWMSYNCI (I)	ADCSOAO (O)
3-2	GPIO33	SCLA (I/O)	EPWMSYNCO (O)	ADCSOAO (O)
5-4	GPIO34	予約 ⁽²⁾	予約 ⁽²⁾	予約 ⁽²⁾

- (1) GPxMUX1/2 は、ピンに対する適切な MUX レジスタ (GPAMUX1、GPAMUX2、または GPBMUX1) を表します。
(2) 「予約」という語句は、GPxMUX1/2 レジスタ設定に割り当てられたペリフェラルが存在しないことを意味します。それを選択した場合、ピンの状態は不定になり、ピンがドライブされる可能性があります。この選択は、将来の拡張用の予約された設定です。

表 4-10 2806 GPIO MUX 表

GPAMUX1/2 ⁽¹⁾ レジスタ・ビット	リセット時のデフォルト			
	基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
GPAMUX1				
1-0	GPIO0	EPWM1A (O)	予約 ⁽²⁾	予約 ⁽²⁾
3-2	GPIO1	EPWM1B (O)	SPISIMOD (I/O)	予約 ⁽²⁾
5-4	GPIO2	EPWM2A (O)	予約 ⁽²⁾	予約 ⁽²⁾
7-6	GPIO3	EPWM2B (O)	SPISOMID (I/O)	予約 ⁽²⁾
9-8	GPIO4	EPWM3A (O)	予約 ⁽²⁾	予約 ⁽²⁾
11-10	GPIO5	EPWM3B (O)	SPICLKD (I/O)	ECAP1 (I/O)
13-12	GPIO6	EPWM4A (O)	EPWMSYNCI (I)	EPWMSYNCO (O)
15-14	GPIO7	EPWM4B (O)	SPISTED (I/O)	ECAP2 (I/O)
17-16	GPIO8	EPWM5A (O)	予約 ⁽²⁾	ADCSOAO (O)
19-18	GPIO9	EPWM5B (O)	SCITXDB (O)	ECAP3 (I/O)
21-20	GPIO10	EPWM6A (O)	予約 ⁽²⁾	ADCSOAO (O)
23-22	GPIO11	EPWM6B (O)	SCIRXDB (I)	ECAP4 (I/O)
25-24	GPIO12	TZ1 (I)	予約 ⁽²⁾	SPISIMOB (I/O)
27-26	GPIO13	TZ2 (I)	予約 ⁽²⁾	SPISOMIB (I/O)
29-28	GPIO14	TZ3 (I)	SCITXDB (O)	SPICLKB (I/O)
31-30	GPIO15	TZ4 (I)	SCIRXDB (I)	SPISTEB (I/O)
GPAMUX2				
1-0	GPIO16	SPISIMOA (I/O)	予約 ⁽²⁾	TZ5 (I)
3-2	GPIO17	SPISOMIA (I/O)	予約 ⁽²⁾	TZ6 (I)
5-4	GPIO18	SPICLKA (I/O)	SCITXDB (O)	予約 ⁽²⁾
7-6	GPIO19	SPISTEA (I/O)	SCIRXDB (I)	予約 ⁽²⁾
9-8	GPIO20	EQEP1A (I)	SPISIMOC (I/O)	予約 ⁽²⁾

GPAMUX1/2 ⁽¹⁾ レジスタ・ビット	リセット時のデフォルト			
	基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
11-10	GPIO21	EQEP1B (I)	SPISOMIC (I/O)	予約 ⁽²⁾
13-12	GPIO22	EQEP1S (I/O)	SPICLK (I/O)	SCITXDB (O)
15-14	GPIO23	EQEP11 (I/O)	SPISTEC (I/O)	SCIRXDB (I)
17-16	GPIO24	ECAP1 (I/O)	EQEP2A (I)	SPISIMOB (I/O)
19-18	GPIO25	ECAP2 (I/O)	EQEP2B (I)	SPISOMIB (I/O)
21-20	GPIO26	ECAP3 (I/O)	EQEP2I (I/O)	SPICLKB (I/O)
23-22	GPIO27	ECAP4 (I/O)	EQEP2S (I/O)	SPISTEB (I/O)
25-24	GPIO28	SCIRXDA (I)	予約 ⁽²⁾	TZ5 (I)
27-26	GPIO29	SCITXDA (O)	予約 ⁽²⁾	TZ6 (I)
29-28	GPIO30	CANRXA (I)	予約 ⁽²⁾	予約 ⁽²⁾
31-30	GPIO31	CANTXA (O)	予約 ⁽²⁾	予約 ⁽²⁾
GPBMUX1				
1-0	GPIO32	SDAA (I/OC)	EPWMSYNCI (I)	ADCSOAO (O)
3-2	GPIO33	SCLA (I/OC)	EPWMSYNCO (O)	ADCSOBO (O)
5-4	GPIO34	予約 ⁽²⁾	予約 ⁽²⁾	予約 ⁽²⁾

- (1) GPxMUX1/2 は、ピンに対する適切な MUX レジスタ (GPAMUX1、GPAMUX2、または GPBMUX1) を表します。
(2) 「予約」という語句は、GPxMUX1/2 レジスタ設定に割り当てられたペリフェラルが存在しないことを意味します。それを選択した場合、ピンの状態は不定になり、ピンがドライブされる可能性があります。この選択は、将来の拡張用の予約された設定です。

表 4-11 2801 および 2802 GPIO MUX 表

GPAMUX1/2 ⁽¹⁾ レジスタ・ビット	リセット時のデフォルト			
	基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
GPAMUX1				
1-0	GPIO0	EPWM1A (O)	予約 ⁽²⁾	予約 ⁽²⁾
3-2	GPIO1	EPWM1B (O)	予約 ⁽²⁾	予約 ⁽²⁾
5-4	GPIO2	EPWM2A (O)	予約 ⁽²⁾	予約 ⁽²⁾
7-6	GPIO3	EPWM2B (O)	予約 ⁽²⁾	予約 ⁽²⁾
9-8	GPIO4	EPWM3A (O)	予約 ⁽²⁾	予約 ⁽²⁾
11-10	GPIO5	EPWM3B (O)	予約 ⁽²⁾	ECAP1 (I/O)
13-12	GPIO6	予約 ⁽²⁾	EPWMSYNCI (I)	EPWMSYNCO (O)
15-14	GPIO7	予約 ⁽²⁾	予約 ⁽²⁾	ECAP2 (I/O)
17-16	GPIO8	予約 ⁽²⁾	予約 ⁽²⁾	ADCSOAO (O)
19-18	GPIO9	予約 ⁽²⁾	予約 ⁽²⁾	予約 ⁽²⁾
21-20	GPIO10	予約 ⁽²⁾	予約 ⁽²⁾	ADCSOBO (O)
23-22	GPIO11	予約 ⁽²⁾	予約 ⁽²⁾	予約 ⁽²⁾
25-24	GPIO12	TZ1 (I)	予約 ⁽²⁾	SPISIMOB (I/O)
27-26	GPIO13	TZ2 (I)	予約 ⁽²⁾	SPISOMIB (I/O)
29-28	GPIO14	TZ3 (I)	予約 ⁽²⁾	SPICLKB (I/O)
31-30	GPIO15	TZ4 (I)	予約 ⁽²⁾	SPISTEB (I/O)
GPAMUX2				
1-0	GPIO16	SPISIMOA (I/O)	予約 ⁽²⁾	TZ5 (I)
3-2	GPIO17	SPISOMIA (I/O)	予約 ⁽²⁾	TZ6 (I)
5-4	GPIO18	SPICLKA (I/O)	予約 ⁽²⁾	予約 ⁽²⁾
7-6	GPIO19	SPISTEA (I/O)	予約 ⁽²⁾	予約 ⁽²⁾
9-8	GPIO20	EQEP1A (I)	予約 ⁽²⁾	予約 ⁽²⁾
11-10	GPIO21	EQEP1B (I)	予約 ⁽²⁾	予約 ⁽²⁾

GPIO およびペリフェラル・マルチプレクス

GPAMUX1/2 ⁽¹⁾ レジスタ・ビット	リセット時のデフォルト 基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
13-12	GPIO22	EQEP1S (I/O)	予約 ⁽²⁾	予約 ⁽²⁾
15-14	GPIO23	EQEP1I (I/O)	予約 ⁽²⁾	予約 ⁽²⁾
17-16	GPIO24	ECAP1 (I/O)	予約 ⁽²⁾	SPISIMOB (I/O)
19-18	GPIO25	ECAP2 (I/O)	予約 ⁽²⁾	SPISOMIB (I/O)
21-20	GPIO26	予約 ⁽²⁾	予約 ⁽²⁾	SPICLKB (I/O)
23-22	GPIO27	予約 ⁽²⁾	予約 ⁽²⁾	SPISTEB (I/O)
25-24	GPIO28	SCIRXDA (I)	予約 ⁽²⁾	TZ5 (I)
27-26	GPIO29	SCITXDA (O)	予約 ⁽²⁾	TZ6 (I)
29-28	GPIO30	CANRXA (I)	予約 ⁽²⁾	予約 ⁽²⁾
31-30	GPIO31	CANTXA (O)	予約 ⁽²⁾	予約 ⁽²⁾
GPBMUX1				
1-0	GPIO32	SDAA (I/OC)	EPWMSYNCI (I)	ADCSOAO (O)
3-2	GPIO33	SCLA (I/OC)	EPWMSYNCO (O)	ADCSOCBO (O)
5-4	GPIO34	予約 ⁽²⁾	予約 ⁽²⁾	予約 ⁽²⁾

(1) GPxMUX1/2 は、ピンに対する適切な MUX レジスタ (GPAMUX1、GPAMUX2、または GPBMUX1) を表します。

(2) 「予約」という語句は、GPxMUX1/2 レジスタ設定に割り当てられたペリフェラルが存在しないことを意味します。それを選択した場合、ピンの状態は不定になり、ピンがドライブされる可能性があります。この選択は、将来の拡張用の予約された設定です。

表 4-12 は、GPIO 機能ではなくペリフェラル機能によって並び替えた GPxMUX1/2 レジスタ表です。この表は、2808 デバイスで利用できるペリフェラルのスーパーセットを示します。示されている一部のペリフェラルは、他のデバイスでは利用できず、予約になっているものもあります。各デバイス上の予約されたロケーションの動作は、表 4-9、表 4-10、および表 4-11 に記載されています。この表を使用すると、特定のペリフェラル機能に割り当てることが可能な GPIO ピンの識別に役立ちます。最終的な選択を特定のデバイスのデータ・マニュアルでクロスチェックすることをお奨めします。

表 4-12 ペリフェラルと GPIO の相互参照

基本 I/O 機能 (GPxMUX1/2 ビット=0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット=0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット=1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット=1,1)
ADC 外部変換開始 A			
GPIO8	EPWM5A	CANTXB	<u>ADCSOCAO</u>
GPIO32	SDAA	EPWMSYNCI	<u>ADCSOCAO</u>
ADC 外部変換開始 B			
GPIO10	EPWM6A	CANRXB	<u>ADCSOCBO</u>
GPIO10	EPWM6A	CANRXB	<u>ADCSOCBO</u>
GPIO33	SCLA	EPWMSYNCO	<u>ADCSOCBO</u>
eCAN-A			
GPIO30	CANRXA	予約	予約
GPIO31	CANTXA	予約	予約
eCAN-B			
GPIO8	EPWM5A	CANTXB	<u>ADCSOCAO</u>
GPIO10	EPWM6A	CANRXB	<u>ADCSOCBO</u>
GPIO12	<u>TZ1</u>	CANTXB	SPISIMOB
GPIO13	<u>TZ2</u>	CANRXB	SPISOMIB
GPIO16	SPISIMOA	CANTXB	<u>TZ5</u>
GPIO17	SPISOMIA	CANRXB	<u>TZ6</u>
GPIO20	EQEP1A	SPISIMOC	CANTXB
GPIO21	EQEP1B	SPISOMIC	CANRXB
eCAP1			
GPIO5	EPWM3B	SPICLKD	ECAP1
GPIO24	ECAP1	EQEP2A	SPISIMOB
eCAP2			
GPIO7	EPWM4B	SPISTED	ECAP2
GPIO25	ECAP2	EQEP2B	SPISOMIB
eCAP3			
GPIO9	EPWM5B	SCITXDB	ECAP3
GPIO26	ECAP3	EQEP2I	SPICLKB
eCAP4			
GPIO11	EPWM6B	SCIRXDB	ECAP4
GPIO27	ECAP4	EQEP2S	SPISTEB

表 4-12 ペリフェラルと GPIO の相互参照 (続き)

基本 I/O 機能 (GPxMUX1/2 ビット =0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット =0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット =1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット =1,1)
ePWM1-6			
GPIO0	EPWM1A	予約	予約
GPIO1	EPWM1B	SPISIMOD	予約
GPIO2	EPWM2A	予約	予約
GPIO3	EPWM2B	SPISOMID	予約
GPIO4	EPWM3A	予約	予約
GPIO5	EPWM3B	SPICLKD	ECAP1
GPIO6	EPWM4A	EPWMSYNCI	EPWMSYNCO
GPIO7	EPWM4B	SPISTED	ECAP2
GPIO8	EPWM5A	CANTXB	ADCSOCAO
GPIO9	EPWM5B	SCITXDB	ECAP3
GPIO10	EPWM6A	CANRXB	ADCSOCBO
GPIO11	EPWM6B	SCIRXDB	ECAP4
ePWM 同期入力			
GPIO6	EPWM4A	EPWMSYNCI	EPWMSYNCO
GPIO32	SDAA	EPWMSYNCI	ADCSOCAO
ePWM 同期出力			
GPIO6	EPWM4A	EPWMSYNCI	EPWMSYNCO
GPIO33	SCLA	EPWMSYNCO	ADCSOCBO
eQEP1			
GPIO20	EQEP1A	SPISIMOC	CANTXB
GPIO21	EQEP1B	SPISOMIC	CANRXB
GPIO22	EQEP1S	SPICLKC	SCITXDB
GPIO23	EQEP1I	SPISTEC	SCIRXDB
eQEP2			
GPIO24	ECAP1	EQEP2A	SPISIMOB
GPIO25	ECAP2	EQEP2B	SPISOMIB
GPIO26	ECAP3	EQEP2I	SPICLKB
GPIO27	ECAP4	EQEP2S	SPISTEB
I2C-A			
GPIO32	SDAA	EPWMSYNCI	ADCSOCAO
GPIO33	SCLA	EPWMSYNCO	ADCSOCBO
SCI-A			
GPIO28	SCIRXDA	予約	TZ5
GPIO29	SCITXDA	予約	TZ6
SCI-B			
GPIO9	EPWM5B	SCITXDB	ECAP3
GPIO11	EPWM6B	SCIRXDB	ECAP4
GPIO14	TZ3	SCITXDB	SPICLKB
GPIO15	TZ4	SCIRXDB	SPISTEB
GPIO18	SPICLKA	SCITXDB	予約
GPIO19	SPISTEA	SCIRXDB	予約
GPIO22	EQEP1S	SPICLKC	SCITXDB
GPIO23	EQEP1I	SPISTEC	SCIRXDB

表 4-12 ペリフェラルと GPIO の相互参照 (続き)

基本 I/O 機能 (GPxMUX1/2 ビット=0,0)	ペリフェラル選択 1 (GPxMUX1/2 ビット=0,1)	ペリフェラル選択 2 (GPxMUX1/2 ビット=1,0)	ペリフェラル選択 3 (GPxMUX1/2 ビット=1,1)
SPI-A			
GPIO16	SPISIMOA	CANTXB	<u>TZ5</u>
GPIO17	SPISOMIA	CANRXB	<u>TZ6</u>
GPIO18	SPICLKA	SCITXDB	予約
GPIO19	SPISTEA	SCIRXDB	予約
SPI-B			
GPIO12	<u>TZ1</u>	CANTXB	SPISIMOB
GPIO13	<u>TZ2</u>	CANRXB	SPISOMIB
GPIO14	<u>TZ3</u>	SCITXDB	SPICLKB
GPIO15	<u>TZ4</u>	SCIRXDB	SPISTEB
GPIO24	ECAP1	EQEP2A	SPISIMOB
GPIO25	ECAP2	EQEP2B	SPISOMIB
GPIO26	ECAP3	EQEP2I	SPICLKB
GPIO27	ECAP4	EQEP2S	SPISTEB
SPI-C			
GPIO20	EQEP1A	SPISIMOC	CANTXB
GPIO21	EQEP1B	SPISOMIC	CANRXB
GPIO22	EQEP1S	SPICLKC	SCITXDB
GPIO23	EQEP1I	SPISTEC	SCIRXDB
SPI-D			
GPIO1	EPWM1B	SPISIMOD	予約
GPIO3	EPWM2B	SPISOMID	予約
GPIO5	EPWM3B	SPICLKD	ECAP1
GPIO7	EPWM4B	SPISTED	ECAP2
トリップ・ゾーン 1 ~ 6			
GPIO12	<u>TZ1</u>	CANTXB	SPISIMOB
GPIO13	<u>TZ2</u>	CANRXB	SPISOMIB
GPIO14	<u>TZ3</u>	SCITXDB	SPICLKB
GPIO15	<u>TZ4</u>	SCIRXDB	SPISTEB
GPIO16	SPISIMOA	CANTXB	<u>TZ5</u>
GPIO17	SPISOMIA	CANRXB	<u>TZ6</u>
GPIO28	SCIRXDA	予約	<u>TZ5</u>
GPIO29	SCITXDA	予約	<u>TZ6</u>

4.6 レジスタ・ビット定義

図 4-4 GPIO ポート A MUX 1 (GPAMUX1) レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO15		GPIO14		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		GPIO8	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-13 GPIO ポート A MUX 1 (GPAMUX1) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-30	GPIO15	00 01 10 11	GPIO15 ピンを以下のように設定します。 GPIO15 (I/O)、汎用 I/O 15 (デフォルト) $\overline{TZ4}$ (I)、トリップ・ゾーン 4 SCIRXDB (I)、SCI-B 受信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPISTEB (I/O)、SPI-B 送信イネーブル
29-28	GPIO14	00 01 10 11	GPIO14 ピンを以下のように設定します。 GPIO14 (I/O)、汎用 I/O 14 (デフォルト) $\overline{TZ3}$ (I)、トリップ・ゾーン 3 SCITXDB (O)、SCI-B 送信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPICLKB (I/O)、SPI-B クロック・イン
27-26	GPIO13	00 01 10 11	GPIO13 ピンを以下のように設定します。 GPIO13 (I/O)、汎用 I/O 13 (デフォルト) $\overline{TZ2}$ (I)、トリップ・ゾーン 2 CANRXB (I)、eCAN-B 受信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ SPISOMIB (I/O)
25-24	GPIO12	00 01 10 11	GPIO12 ピンを以下のように設定します。 GPIO12 (I/O)、汎用 I/O 12 (デフォルト) $\overline{TZ1}$ (I)、トリップ・ゾーン 1 CANTXB (O)、eCAN-B 送信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ SPISIMOB (I/O)、SPI-B スレーブイン、マスターアウト
23-22	GPIO11	00 01 10 11	GPIO11 ピンを以下のように設定します。 GPIO11 (I/O)、汎用 I/O 11 (デフォルト) EPWM6B (O)、ePWM 6 出力 B。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SCIRXDB (I)、SCI-B 受信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ ECAP4 (I/O)、eCAP4。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) 予約されている設定を選択した場合は、ピンの状態は不定になり、ピンがドライブされる可能性があります。これらの選択は将来の拡張用に予約されており、使用すべきではありません。

表 4-13 GPIO ポート A MUX 1 (GPAMUX1) レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾
21-20	GPIO10	00 01 10 11	GPIO10 ピンを以下のように設定します。 GPIO10 (I/O)、汎用 I/O 10 (デフォルト) EPWM6A (O)、ePWM6 出力 A。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ CANRXB (I)、eCAN-B 受信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ ADCSOCBO (O)、ADC 変換開始 B
19-18	GPIO9	00 01 10 11	GPIO9 ピンを以下のように設定します。 ⁽¹⁾ GPIO9 (I/O)、汎用 I/O 9 (デフォルト) EPWM5B (O)、ePWM5 出力 B。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SCITXDB (O)、SCI-B 送信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ ECAP3 (I/O)、eCAP3
17-16	GPIO8	00 01 10 11	GPIO8 ピンを以下のように設定します。 GPIO8 (I/O)、汎用 I/O 8 (デフォルト) EPWM5A (O)、ePWM5 出力 A。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ CANTXB (O)、eCAN-B 送信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ ADCSOCAO (O)、ADC 変換開始 A
15-14	GPIO7	00 01 10 11	GPIO7 ピンを以下のように設定します。 GPIO7 (I/O)、汎用 I/O 7 (デフォルト) EPWM4B (O)、ePWM4 出力 B。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPISTED (I/O)、SPI-D 送信イネーブル。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ ECAP2 (I/O)、eCAP2
13-12	GPIO6	00 01 10 11	GPIO6 ピンを以下のように設定します。 GPIO6 (I/O)、汎用 I/O 6 (デフォルト) EPWM4A (O)、ePWM4 出力 A。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ EPWMSYNCI (I)、ePWM 同期入力 EPWMSYNCO (O)、ePWM 同期出力
11-10	GPIO5	00 01 10 11	GPIO5 ピンを以下のように設定します。 GPIO5 (I/O)、汎用 I/O 5 (デフォルト) EPWM3B (O)、ePWM3 出力 B SPICLKD (I/O)、SPI-D クロック。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ ECAP1 (I/O)、eCAP1
9-8	GPIO4	00 01 10 11	GPIO4 ピンを以下のように設定します。 GPIO4 (I/O)、汎用 I/O 4 (デフォルト) EPWM3A (O)、ePWM3 出力 A 予約 ⁽²⁾ 予約 ⁽²⁾

表 4-13 GPIO ポート A MUX 1 (GPAMUX1) レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾
7-6	GPIO3	00 01 10 11	GPIO3 ピンを以下のように設定します。 GPIO3 (I/O)、汎用 I/O 3 (デフォルト) EPWM2B (O)、ePWM2 出力 B SPISOMID (I/O)、SPI-B スレーブアウト、マスターイン。 このオプションは、2802 および 2801 デバイスでは予約されています。(2) 予約
5-4	GPIO2	00 01 10 11	GPIO2 ピンを以下のように設定します。 GPIO2 (I/O)、汎用 I/O 2 (デフォルト) EPWM2A (O)、ePWM2 出力 A 予約 ⁽²⁾ 予約 ⁽²⁾
3-2	GPIO1	00 01 10 11	GPIO1 ピンを以下のように設定します。 GPIO1 (I/O)、汎用 I/O 1 (デフォルト) EPWM1B (O)、ePWM1 出力 B SPISIMOD (I/O)、SPI-D スレーブイン、マスターアウト。 このオプションは、2802 および 2801 デバイスでは予約されています。(2) 予約 ⁽²⁾
1-0	GPIO0	00 01 10 11	GPIO0 ピンを以下のように設定します。 GPIO0 (I/O)、汎用 I/O 0 (デフォルト) EPWM1A (O)、ePWM1 出力 A 予約 ⁽²⁾ 予約 ⁽²⁾

図 4-5 GPIO ポート A MUX 2 (GPAMUX2) レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-14 GPIO ポート A MUX 2 (GPAMUX2) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-30	GPIO31	00 01 10 11	GPIO31 ピンを以下のように設定します。 GPIO31 (I/O)、汎用 I/O 31 (デフォルト) CANTXA (O)、eCAN-A 送信 予約 ⁽²⁾ 予約 ⁽²⁾

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) 予約されている設定を選択した場合は、ピンの状態は不定になり、ピンがドライブされる可能性があります。これらの選択は将来の拡張用に予約されており、使用すべきではありません。

表 4-14 GPIO ポート A MUX 2 (GPAMUX2) レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾
29-28	GPIO30	00 01 10 11	GPIO30 ピンを以下のように設定します。 GPIO30 (I/O)、汎用 I/O 30 (デフォルト) CANRXA (I)、eCAN-A 受信 予約 ⁽²⁾ 予約 ⁽²⁾
27-26	GPIO29	00 01 10 11	GPIO29 ピンを以下のように設定します。 GPIO29 (I/O)、汎用 I/O 29 (デフォルト) SCITXDA (O)、SCI-A 送信。 このオプションは、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ 予約 ⁽²⁾ TZ6 (I)、トリップ・ゾーン 6
25-24	GPIO28	00 01 10 11	GPIO28 ピンを以下のように設定します。 GPIO28 (I/O)、汎用 I/O 28 (デフォルト) SCIRXDA (I)、SCI-A 受信 予約 ⁽²⁾ TZ5 (I)、トリップ・ゾーン 5
23-22	GPIO27	00 01 10 11	GPIO27 ピンを以下のように設定します。 GPIO27 (I/O)、汎用 I/O 27 (デフォルト) ECAP4 (I/O)、eCAP4。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ EQEP2S (I/O)、eQEP2 ストロープ。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPISTEB (I/O)、SPI-B 送信イネーブル
21-20	GPIO26	00 01 10 11	GPIO26 ピンを以下のように設定します。 GPIO26 (I/O)、汎用 I/O 26 (デフォルト) ECAP3 (I/O)、eCAP3。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ EQEP2I (I/O)、eQEP2 インデックス。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPICLKB (I/O)、SPI-B クロック
19-18	GPIO25	00 01 10 11	GPIO25 ピンを以下のように設定します。 GPIO25 (I/O)、汎用 I/O 25 (デフォルト) ECAP2 (I/O)、eCAP2 EQEP2B (I)、eQEP2 入力 B。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPISOMIB (I/O)、SPI-B スレーブアウト、マスターイン
17-16	GPIO24	00 01 10 11	GPIO24 ピンを以下のように設定します。 GPIO24 (I/O)、汎用 I/O 24 (デフォルト) ECAP1 (I/O)、eCAP1 EQEP2A (I)、eQEP2 入力 A。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SPISIMOB (I/O)、SPI-B スレーブイン、マスターアウト

表 4-14 GPIO ポート A MUX 2 (GPAMUX2) レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾
15-14	GPIO23	00 01 10 11	GPIO23 ピンを以下のように設定します。 GPIO23 (I/O)、汎用 I/O 23 (デフォルト) EQEP1I (I/O)、eQEP1 インデックス SPISTEC (I/O)、SPI-C 送信イネーブル。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SCIRXDB (I/O)、SCI-B 受信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾
13-12	GPIO22	00 01 10 11	GPIO22 ピンを以下のように設定します。 GPIO22 (I/O)、汎用 I/O 22 (デフォルト) EQEP1S (I/O)、eQEP1 ストロブ SPICLK (I/O)、SPI-C クロック。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ SCITXDB (O)、SCI-B 送信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾
11-10	GPIO21	00 01 10 11	GPIO21 ピンを以下のように設定します。 GPIO21 (I/O)、汎用 I/O 21 (デフォルト) EQEP1B (I)、eQEP1 入力 B SPISOMIC (I/O)、SPI-C スレーブアウト、マスターイン。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ CANRXB (I)、eCAN-B 受信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾
9-8	GPIO20	00 01 10 11	GPIO20 ピンを以下のように設定します。 GPIO20 (I/O)、汎用 I/O 22 (デフォルト) EQEP1A (I)、eQEP1 入力 A SPISIMOC (I/O)、SPI-C スレーブイン、マスターアウト。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ CANTXB (O)、eCAN-B 送信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾
7-6	GPIO19	00 01 10 11	GPIO19 ピンを以下のように設定します。 GPIO19 (I/O)、汎用 I/O 19 (デフォルト) SPISTEA (I/O)、SPI-A 送信イネーブル SCIRXDB (I)、SCI-B 受信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ 予約 ⁽²⁾
5-4	GPIO18	00 01 10 11	GPIO18 ピンを以下のように設定します。 GPIO18 (I/O)、汎用 I/O 18 (デフォルト) SPICLKA (I/O)、SPI-A クロック SCITXDB (O)、SCI-B 送信。 このオプションは、2802 および 2801 デバイスでは予約されています。 ⁽²⁾ 予約 ⁽²⁾
3-2	GPIO17	00 01 10 11	GPIO17 ピンを以下のように設定します。 GPIO17 (I/O)、汎用 I/O 17 (デフォルト) SPISOMIA (I/O)、SPI-A スレーブアウト、マスターイン CANRXB (I)、eCAN-B 受信。 このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ TZ6 (I)、トリップ・ゾーン 6

表 4-14 GPIO ポート A MUX 2 (GPAMUX2) レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明 ⁽¹⁾
1-0	GPIO16		GPIO16 ピンを以下のように設定します。
		00	GPIO16 (I/O)、汎用 I/O 16 (デフォルト)
		01	SPISIMOA (I/O)、SPI-A スレーブイン、マスターアウト
		10	CANTXB (O)、eCAN-B 送信。
		11	このオプションは、2806、2802、および 2801 デバイスでは予約されています。 ⁽²⁾ TZ5 (I)、トリップ・ゾーン 5

図 4-6 GPIO ポート B MUX 1 (GPBMUX1) レジスタ

31	6	5	4	3	2	1	0
予約	GPIO34		GPIO33		GPIO32		
R-0	R/W-0		R/W-0		R/W-0		

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-15 GPIO ポート B MUX 1 (GPBMUX1) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-6	予約		予約
5-4	GPIO34		GPIO34 ピンを以下のように設定します。
		00	GPIO34 (I/O)、汎用 I/O 34 (デフォルト)
		01	予約 ⁽²⁾
		10	予約 ⁽²⁾
		11	予約 ⁽²⁾
3-2	GPIO33		GPIO33 ピンを以下のように設定します。
		00	GPIO33 (I/O)、汎用 I/O 33 (デフォルト)
		01	SCLA (I/O)、I2C クロック
		10	EPWMSYNCO (O)、ePWM 同期出力
		11	ADCSOCBO (O)
1-0	GPIO32		GPIO32 ピンを以下のように設定します。
		00	GPIO32 (I/O)、汎用 I/O 32 (デフォルト)
		01	SDAA (I/O)、I2C データ
		10	EWPMYSYNCI (I)、ePWM 同期入力
		11	ADCSOCAO (O)、ADC 変換開始 A

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) 予約されている設定を選択した場合は、ピンの状態は不定になり、ピンがドライブされる可能性があります。これらの選択は将来の拡張用に予約されており、使用すべきではありません。

表 4-16 GPIO ポート B MUX 2 (GPBMUX2) レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	予約	R/W	0,0	予約

図 4-7 GPIO ポート A Qualification 制御 (GPACTRL) レジスタ

31	24	23	16
QUALPRD3		QUALPRD2	
15	8	7	0
QUALPRD1 R/W-0		QUALPRD0 R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

GPACTRL1、GPACTRL2、および GPBCTRL1 レジスタは、3 または 6 サンプルのウィンドウを使用して入力フィルタリング (Qualification) について設定される場合、入力ピンのサンプリング周期を指定します。サンプリング周期は、SYSCLKOUT を基準としたフィルタリング (Qualification) サンプル間の時間の量です。サンプルの数は、GPAQSEL1、GPAQSEL2、または GPBQSEL1 レジスタで指定されます。

表 4-17 GPIO ポート A Qualification 制御 (GPACTRL) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
31-24	QUALPRD3		GPIO24 ~ GPIO31 ピンのサンプリング周期を指定します。
		0x00	サンプリング周期 = $T_{\text{SYSCLKOUT}}$ (2)
		0x01	サンプリング周期 = $2 \times T_{\text{SYSCLKOUT}}$
		0x02	サンプリング周期 = $4 \times T_{\text{SYSCLKOUT}}$
		0xFF	サンプリング周期 = $510 \times T_{\text{SYSCLKOUT}}$
23-16	QUALPRD2		QUALPRD2 GPIO16 ~ GPIO23 ピンのサンプリング周期を指定します。
		0x00	サンプリング周期 = $T_{\text{SYSCLKOUT}}$ (2)
		0x01	サンプリング周期 = $2 \times T_{\text{SYSCLKOUT}}$
		0x02	サンプリング周期 = $4 \times T_{\text{SYSCLKOUT}}$
		0xFF	サンプリング周期 = $510 \times T_{\text{SYSCLKOUT}}$
15-8	QUALPRD1		GPIO8 ~ GPIO15 ピンのサンプリング周期を指定します。
		0x00	サンプリング周期 = $T_{\text{SYSCLKOUT}}$ (2)
		0x01	サンプリング周期 = $2 \times T_{\text{SYSCLKOUT}}$
		0x02	サンプリング周期 = $4 \times T_{\text{SYSCLKOUT}}$
		0xFF	サンプリング周期 = $510 \times T_{\text{SYSCLKOUT}}$
7-0	QUALPRD0		QUALPRD0 GPIO0 ~ GPIO7 ピンのサンプリング周期を指定します。
		0x00	サンプリング周期 = $T_{\text{SYSCLKOUT}}$ (2)
		0x01	サンプリング周期 = $2 \times T_{\text{SYSCLKOUT}}$
		0x02	サンプリング周期 = $4 \times T_{\text{SYSCLKOUT}}$
		0xFF	サンプリング周期 = $510 \times T_{\text{SYSCLKOUT}}$

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) $T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の周期を示します。

図 4-8 GPIO ポート B Qualification 制御 (GPBCTRL) レジスタ

31	8	7	0
予約 R-0		QUALPRD0 R/W-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-18 GPIO ポート B Qualification 制御 (GPBCTRL) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-8	予約		予約
7-0	QUALPRD0	0x00 0x01 0x02 ... 0xFF	GPIO32 ~ GPIO34 ピンのフィルタリング (Qualification) サンプルング周期を指定します。 サンプルング周期 = $T_{\text{SYSCLKOUT}}$ ⁽²⁾ サンプルング周期 = $2 \times T_{\text{SYSCLKOUT}}$ サンプルング周期 = $4 \times T_{\text{SYSCLKOUT}}$... サンプルング周期 = $510 \times T_{\text{SYSCLKOUT}}$

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

(2) $T_{\text{SYSCLKOUT}}$ は、 SYSCLKOUT の周期を示します。

図 4-9 GPIO ポート A Qualification セレクト 1 (GPAQSEL1) レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO15		GPIO14		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		GPIO8	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-19 GPIO ポート A Qualification セレクト 1 (GPAQSEL1) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
31-0	GPIO15-GPIO0		GPIO0 ~ GPIO15 の入力フィルタリング (Qualification) タイプを選択します。各 GPIO 入力の入力フィルタリング (Qualification) は、図 4-9 に示すように 2 つのビットによって制御されます。
		00	SYSCLKOUT に対してのみ同期します。ペリフェラルと GPIO ピンの両方に対して有効。
		01	3 サンプルを使用したフィルタリング (Qualification)。GPIO またはペリフェラル機能として設定されたピンに対して有効。サンプル間の時間は、GPACTRL レジスタで指定されます。
		10	6 サンプルを使用したフィルタリング (Qualification)。GPIO またはペリフェラル機能として設定されたピンに対して有効。サンプル間の時間は、GPACTRL レジスタで指定されます。
		11	非同期。(同期またはフィルタリング (Qualification) なし)。このオプションは、ペリフェラルとして設定されたピンだけに適用されます。ピンが GPIO 入力として設定された場合、このオプションは 0,0、または SYSCLKOUT に対する同期と同じです。

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

図 4-10 GPIO ポート A Qualification セレクト 2 (GPAQSEL2) レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIO31		GPIO30		GPIO29		GPIO28		GPIO27		GPIO26		GPIO25		GPIO24	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO23		GPIO22		GPIO21		GPIO20		GPIO19		GPIO18		GPIO17		GPIO16	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-20 GPIO ポート A Qualification セレクト 2 (GPAQSEL2) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
31-0	GPIO31-GPIO16		GPIO16 ~ GPIO31 の入力フィルタリング (Qualification) タイプを選択します。各 GPIO 入力の入力フィルタリング (Qualification) は、図 4-10 に示すように 2 つのビットによって制御されます。
		00	SYSCLKOUT に対してのみ同期します。ペリフェラルと GPIO ピンの両方に対して有効。
		01	3 サンプルを使用したフィルタリング (Qualification)。GPIO またはペリフェラル機能として設定されたピンに対して有効。サンプル間の時間は、GPACTRL レジスタで指定されます。
		10	6 サンプルを使用したフィルタリング (Qualification)。GPIO またはペリフェラル機能として設定されたピンに対して有効。サンプル間の時間は、GPACTRL レジスタで指定されます。
		11	非同期。(同期またはフィルタリング (Qualification) なし)。このオプションは、ペリフェラルとして設定されたピンだけに適用されます。ピンが GPIO 入力として設定された場合、このオプションは 0,0 と同じか、または SYSCLKOUT に対して同期します。

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

図 4-11 GPIO ポート B Qualification セレクト 1 (GPBQSEL1) レジスタ

31	6	5	4	3	2	1	0
予約		GPIO34		GPIO33		GPIO32	
R-0		R/W-0		R/W-0		R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-21 GPIO ポート B Qualification セレクト 1 (GPBQSEL1) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-6	予約		
5-0	GPIO34-GPIO32	00 01 10 11	GPIO32 ~ GPIO34 の入力フィルタリング (Qualification) タイプを選択します。各 GPIO 入力の入力フィルタリング (Qualification) は、 図 4-11 に示すように 2 つのビットによって制御されます。 00 SYSCLKOUT に対してのみ同期します。ペリフェラルと GPIO ピンの両方に対して有効。 01 3 サンプルを使用したフィルタリング (Qualification)。GPIO またはペリフェラル機能として設定されたピンに対して有効。サンプル間の時間は、GPBCTRL レジスタで指定されます。 10 6 サンプルを使用したフィルタリング (Qualification)。GPIO またはペリフェラル機能として設定されたピンに対して有効。サンプル間の時間は、GPBCTRL レジスタで指定されます。 11 非同期。(同期またはフィルタリング (Qualification) なし)。このオプションは、ペリフェラルとして設定されたピンだけに適用されます。ピンが GPIO 入力として設定された場合、このオプションは 0,0、または SYSCLKOUT に対する同期と同じです。

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

表 4-22 GPIO ポート B Qualification セレクト 2 (GPBQSEL2) レジスタのフィールドの説明

ビット	フィールド	説明
31-0	予約	将来の拡張用に予約

ピンが対応する MUX レジスタで GPIO として設定されている場合、GPADIR および GPBDIR レジスタは、それらのピンの方向を制御します。方向レジスタは、ペリフェラル機能として設定されたピンには影響しません。

図 4-12 GPIO ポート A 方向 (GPADIR) レジスタ

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-23 GPIO ポート A 方向 (GPADIR) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
31-0	GPIO31-GPIO0		指定されたピンが対応する GPAMUX1 または GPAMUX2 レジスタで GPIO として設定されている場合は、GPIO ポート A ピンの方向を制御します。
		0	GPIO ピンを入力として設定します。(デフォルト)
		1	GPIO ピンを出力として設定します。
			GPADAT 出力ラッチに現在入っている値は、そのピンでドライブされます。ピンを入力から出力に変更する前に GPADAT ラッチを初期化するには、GPASET、GPACLEAR、および GPATOGGLE レジスタを使用します。

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

図 4-13 GPIO ポート B 方向 (GPBDIR) レジスタ

31	3	2	1	0
	予約	GPIO34	GPIO33	GPIO32
	R-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-24 GPIO ポート B 方向 (GPBDIR) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
31-3	予約		
2-0	GPIO34-GPIO32		指定されたピンが対応する GPBMUX1 または GPBMUX2 レジスタで GPIO として設定されている場合は、GPIO ポート B ピンの方向 (入力または出力) を制御します。
		0	GPIO ピンを入力として設定します。(デフォルト)
		1	GPIO ピンを出力として設定します。
			GPBDAT 出力ラッチに現在入っている値は、そのピンでドライブされます。ピンを入力から出力に変更する前に GPBDAT ラッチを初期化するには、GPBSET、GPBCLEAR、および GPBTOGGLE レジスタを使用します。

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

プルアップ・ディスエーブル (GPxPUD) レジスタでは、どのピンで内部プルアップ抵抗をイネーブルにする必要があるのかを設定できます。外部リセット信号 (XRS) が Low の場合、ePWM 出力 (GPIO0 ~ GPIO11) として設定できるピン上の内部プルアップは、すべて非同期にディスエーブルにされます。他のすべてのピン上の内部プルアップは、リセット時にイネーブルにされます。リセットから出てくると、このレジスタへのライトによってソフトウェアで選択的にプルアップをイネーブルまたはディスエーブルにしない限り、プルアップはデフォルト状態のままになります。プルアップ設定は、I/O として設定されたピンとペリフェラル機能として設定されたピンの両方に適用されます。

図 4-14 GPIO ポート A プルアップ・ディスエーブル (GPAPUD) レジスタ

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

凡例: R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-25 GPIO ポート A 内部プルアップ・ディスエーブル (GPAPUD) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-0	GPIO31-GPIO0	0	選択された GPIO ポート A ピン上の内部プルアップ抵抗を設定します。各 GPIO ピンは、図 4-14 に示されているように、このレジスタ内の 1 つのビットに対応します。
		1	指定されたピン上の内部プルアップをイネーブルにします。(GPIO12 ~ GPIO31 に対するデフォルト)
			指定されたピン上の内部プルアップをディスエーブルにします。(GPIO0 ~ GPIO11 に対するデフォルト)

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

図 4-15 GPIO ポート B プルアップ・ディスエーブル (GPBPUD) レジスタ

31	3	2	1	0
	予約	GPIO34	GPIO33	GPIO32
	R-0	R/W-0	R/W-0	R/W-0

凡例: R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-26 GPIO ポート B 内部プルアップ・ディスエーブル (GPBPUD) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-3	予約		予約
2-0	GPIO34- GPIO32	0	選択された GPIO ポート B ピン上の内部プルアップ抵抗を設定します。各 GPIO ピンは、図 4-15 に示されているように、このレジスタ内の 1 つのビットに対応します。
		1	指定されたピン上の内部プルアップをイネーブルにします。(デフォルト)
			指定されたピン上の内部プルアップをディスエーブルにします。

(1) このレジスタは EALLOW 保護されています。詳細については、セクション 5.2 を参照してください。

GPIO データ・レジスタは、ピンがどのモードにあるのかにかかわらず、GPIO ピンの現在のステータスを示します。ピンが GPIO 出力としてイネーブルにされている場合、このレジスタへのライトは、それぞれの GPIO ピンを High または Low に設定します。それ以外の場合は、ライトされた値はラッチされますが、無視されます。出力レジスタ・ラッチの状態は、次のライト動作まで現在の状態のままになります。リセットは、すべてのビットとラッチされた値をゼロにクリアします。GPADAT および GPBDAT レジスタからリードされた値は、GPADAT または GPBDAT レジスタの出力ラッチの状態ではなく、ピンの状態（フィルタリング（Qualification）後の）を反映します。

一般に、DAT レジスタは、ピンの現在の状態をリードするために使用されます。ピンの出力レベルを容易に変更するには、SET、CLEAR、および TOGGLE レジスタを参照してください。

図 4-16 GPIO ポート A データ（GPADAT）レジスタ

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値⁽¹⁾

(1) x = リセット後の GPADAT レジスタの状態は未知です。それはリセット後のピンのレベルに依存します。

表 4-27 GPIO ポート A データ（GPADAT）レジスタのフィールドの説明

ビット	フィールド	値	説明
31-0	GPIO31-GPIO0	0	<p>図 4-16 に示されているように、各ビットは 1 つの GPIO ポート A ピン（GPIO0 ~ GPIO31）に対応します。</p> <p>0 のリードは、ピンの設定対象となっているモードにかかわらず、ピンの状態が現在 Low であることを示します。</p> <p>対応する GPAMUX1/2 および GPADIR レジスタでピンが GPIO 出力として設定されている場合は、0 のライトは 0 の出力を強制します。それ以外の場合は、値はラッチされますが、ピンをドライブするためには使用されません。</p>
		1	<p>1 のリードは、ピンの設定対象となっているモードにかかわらず、ピンの状態が現在 High であることを示します。</p> <p>適切な GPAMUX1/2 および GPADIR レジスタでピンが GPIO 出力として設定されている場合は、1 のライトは 1 の出力を強制します。それ以外の場合は、値はラッチされますが、ピンをドライブするためには使用されません。</p>

図 4-17 GPIO ポート B データ (GPBDAT) レジスタ

31	3	2	1	0
予約	GPIO34	GPIO33	GPIO32	
R-0	R/W-x	R/W-x	R/W-x	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値⁽¹⁾

(1) x = リセット後の GPBDAT レジスタの状態は未知です。それはリセット後のピンのレベルに依存します。

表 4-28 GPIO ポート B データ (GPBDAT) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-3	予約		予約
2-0	GPIO34-GPIO32	0	<p>図 4-17 に示されているように、各ビットは 1 つの GPIO ポート B ピン (GPIO32 ~ GPIO34) に対応します。</p> <p>0 のリードは、ピンの設定対象となっているモードにかかわらず、ピンの状態が現在 Low であることを示します。</p> <p>対応する GPBMUX1 および GPBDIR レジスタでピンが GPIO 出力として設定されている場合は、0 のライトは 0 の出力を強制します。それ以外の場合は、値はラッチされますが、ピンをドライブするためには使用されません。</p>
		1	<p>1 のリードは、ピンの設定対象となっているモードにかかわらず、ピンの状態が現在 High であることを示します。</p> <p>GPBMUX1 および GPBDIR レジスタでピンが GPIO 出力として設定されている場合は、1 のライトは 1 の出力を強制します。それ以外の場合は、値はラッチされますが、ピンをドライブするためには使用されません。</p>

図 4-18 GPIO ポート A セット、クリア、およびトグル (GPASET、GPACLEAR、GPATOGGLE) レジスタ

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-29 GPIO ポート A セット (GPASET) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-0	GPIO31-GPIO0		図 4-18 に示されているように、各 GPIO ポート A ピン (GPIO0 ~ GPIO31) は、このレジスタ内の 1 つのビットに対応します。
		0	0 のライトは無視されます。このレジスタは、常に 0 をリード・バックします。
		1	1 のライトは、それぞれの出力データ・ラッチを High にします。ピンが GPIO 出力として設定されている場合、そのピンは High にドライブされます。ピンが GPIO 出力として設定されていない場合は、ラッチは High に設定されますが、ピンはドライブされません。

表 4-30 GPIO ポート A クリア (GPACLEAR) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-0	GPIO31 - GPIO0		図 4-18 に示されているように、各 GPIO ポート A ピン (GPIO0 ~ GPIO31) は、このレジスタ内の 1 つのビットに対応します。
		0	0 のライトは無視されます。このレジスタは、常に 0 をリード・バックします。
		1	1 のライトは、それぞれの出力データ・ラッチを Low にします。ピンが GPIO 出力として設定されている場合、そのピンは Low にドライブされます。ピンが GPIO 出力として設定されていない場合は、ラッチはクリアされますが、ピンはドライブされません。

表 4-31 GPIO ポート A トグル (GPATOGGLE) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-0	GPIO31-GPIO0		図 4-18 に示されているように、各 GPIO ポート A ピン (GPIO0 ~ GPIO31) は、このレジスタ内の 1 つのビットに対応します。
		0	0 のライトは無視されます。このレジスタは、常に 0 をリード・バックします。
		1	1 のライトは、それぞれの出力データ・ラッチを現在の状態からトグルします。ピンが GPIO 出力として設定されている場合、そのピンは現在の状態の反対方向にドライブされます。ピンが GPIO 出力として設定されていない場合は、ラッチはトグルされますが、ピンはドライブされません。

図 4-19 GPIO ポート B セット、クリア、およびトグル (GPBSET、GPBCLEAR、GPBTOGGLE) レジスタ

31	3	2	1	0
予約		GPIO34	GPIO33	GPIO32
R-0		R/W-0	R/W-0	R/W-0

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-32 GPIO ポート B セット (GPBSET) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-3	予約		予約
2-0	GPIO34-GPIO32	0 1	<p>図 4-19 に示されているように、各 GPIO ポート B ピン (GPIO32 ~ GPIO34) は、このレジスタ内の 1 つのビットに対応します。</p> <p>0 のライトは無視されます。このレジスタは、常に 0 をリード・バックします。</p> <p>1 のライトは、それぞれの出力データ・ラッチを High にします。ピンが GPIO 出力として設定されている場合、そのピンは High にドライブされます。ピンが GPIO 出力として設定されていない場合は、ラッチはセットされますが、ピンはドライブされません。</p>

表 4-33 GPIO ポート B クリア (GPBCLEAR) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-3	予約		予約
2-0	GPIO34-GPIO32	0 1	<p>図 4-19 に示されているように、各 GPIO ポート B ピン (GPIO32 ~ GPIO34) は、このレジスタ内の 1 つのビットに対応します。</p> <p>0 のライトは無視されます。このレジスタは、常に 0 をリード・バックします。</p> <p>1 のライトは、それぞれの出力データ・ラッチを Low にします。ピンが GPIO 出力として設定されている場合、そのピンは Low にドライブされます。ピンが GPIO 出力として設定されていない場合は、ラッチはクリアされますが、ピンはドライブされません。</p>

表 4-34 GPIO ポート B トグル (GPBTOGGLE) レジスタのフィールドの説明

ビット	フィールド	値	説明
31-3	予約		予約
2-0	GPIO34-GPIO32	0 1	<p>図 4-19 に示されているように、各 GPIO ポート B ピン (GPIO32 ~ GPIO34) は、このレジスタ内の 1 つのビットに対応します。</p> <p>0 のライトは無視されます。このレジスタは、常に 0 をリード・バックします。</p> <p>1 のライトは、それぞれの出力データ・ラッチを現在の状態からトグルします。ピンが GPIO 出力として設定されている場合、そのピンは現在の状態の反対方向にドライブされます。ピンが GPIO 出力として設定されていない場合は、ラッチはクリアされますが、ピンはドライブされません。</p>

図 4-20 GPIO XINT1、XINT2、XNMI 割り込みセレクト (GPIOXINT1SEL、GPIOXINT2SEL、GPIOXNMISEL) レジスタ

15	5	4	0
予約			
R-0	GPIOSEL		
	R/W-0		

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 4-35 GPIO XINT1 割り込みセレクト (GPIOXINT1SEL) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
15-5	予約		予約
4-0	GPIOSEL		どのポート A GPIO 信号 (GPIO0 ~ GPIO31) を XINT1 割り込みソースとして使用するのを選択します。また、 セクション 6.6 に記載されている XINT1CR レジスタで割り込みを設定することもできます。
		00000	GPIO0 ピンを XINT1 割り込みソースとして選択する (デフォルト)
		00001	GPIO1 ピンを XINT1 割り込みソースとして選択する
	
		11110	GPIO30 ピンを XINT1 割り込みソースとして選択する
		11111	GPIO31 ピンを XINT1 割り込みソースとして選択する

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#)を参照してください。

表 4-36 GPIO XINT2 割り込みセレクト (GPIOXINT2SEL) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
15-5	予約		予約
4-0	GPIOSEL		どのポート A GPIO 信号 (GPIO0 ~ GPIO31) を XINT2 割り込みソースとして使用するのを選択します。また、 セクション 6.6 に記載されている XINT2CR レジスタで割り込みを設定することもできます。
			信号を ADC 変換開始として使用するには、その信号を ADCTRL2 レジスタでイネーブルにします。ADCSOC は、常に立ち上がりエッジ・センシティブです。
		00000	GPIO0 ピンを XINT2 割り込みソースとして選択する (デフォルト)
		00001	GPIO1 ピンを XINT2 割り込みソースとして選択する
	
		11110	GPIO30 ピンを XINT2 割り込みソースとして選択する
		11111	GPIO31 ピンを XINT2 割り込みソースとして選択する

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#)を参照してください。

表 4-37 GPIO XNMI 割り込みセレクト (GPIOXNMISEL) レジスタのフィールドの説明

ビット	フィールド	値	説明 (1)
15-5	予約		予約
4-0	GPIOSEL		どのポート A GPIO 信号 (GPIO0 ~ GPIO31) を XNMI 割り込みソースとして使用するのを選択します。また、 セクション 6.6 に記載されている XNMICR レジスタで割り込みを設定することもできます。
		00000	GPIO0 ピンを XNMI 割り込みソースとして選択する (デフォルト)
		00001	GPIO1 ピンを XNMI 割り込みソースとして選択する
	
		11110	GPIO30 ピンを XNMI 割り込みソースとして選択する
		11111	GPIO31 ピンを XNMI 割り込みソースとして選択する

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#)を参照してください。

図 4-21 GPIO 低電力モード・ウェイクアップ・セレクト (GPIOPMSEL) レジスタ

31	30	29	28	27	26	25	24
GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 4-38 GPIO 低電力モード・ウェイクアップ・セレクト (GPIOPMSEL) レジスタのフィールドの説明

ビット	フィールド	値	説明 ⁽¹⁾
31-0	GPIO31 - GPIO0	0	低電力モード・ウェイクアップ選択。このレジスタの各ビットは、 図 4-21 に示されているように、1つの GPIO ポート A ピン (GPIO0 ~ GPIO31) に対応します。
		1	ビットがクリアされた場合、対応するピンの信号は HALT および STANDBY 低電力モードに影響しません。
			それぞれのビットが 1 に設定された場合、対応するピンの信号は、HALT と STANDBY の両方の低電力モードからデバイスをウェイクすることができます。

(1) このレジスタは EALLOW 保護されています。詳細については、[セクション 5.2](#) を参照してください。

ペリフェラル・フレーム

この章では、ペリフェラル・フレームについて説明します。また、デバイス・エミュレーション・レジスタについても説明します。

項目	ページ
5.1 ペリフェラル・フレーム・レジスタ	106
5.2 EALLOW 保護されているレジスタ	108
5.3 デバイス・エミュレーション・レジスタ	112
5.4 ライト後のリード保護	113

5.1 ペリフェラル・フレーム・レジスタ

280x デバイスには、3 つのペリフェラル・レジスタ空間があります。これらの空間は以下のように分類されます。

- ペリフェラル・フレーム 0：これらは、CPU メモリ・バスに直接マップされるペリフェラルです。[表 5-1](#) を参照してください。
- ペリフェラル・フレーム 1：これらは、32 ビット・ペリフェラル・バスにマップされるペリフェラルです。[表 5-2](#) を参照してください。
- ペリフェラル・フレーム 2：これらは、16 ビット・ペリフェラル・バスにマップされるペリフェラルです。[表 5-3](#) を参照してください。

表 5-1 ペリフェラル・フレーム 0 レジスタ

名前	アドレス範囲	サイズ (x16)	アクセス・タイプ ⁽¹⁾
デバイス・エミュレーション・レジスタ	0x0880 0x09FF	384	EALLOW 保護されている
FLASH レジスタ ⁽²⁾	0x0A80 0x0ADF	96	EALLOW 保護されている。CSM 保護されている
コード・セキュリティ・モジュール・レジスタ	0x0AE0 0x0AEF	16	EALLOW 保護されている
CPU-TIMER0/1/2 レジスタ	0x0C00 0x0C3F	64	EALLOW 保護されていない
PIE レジスタ	0x0CFF 0x0CE0	32	EALLOW 保護されていない
PIE ベクタ・テーブル	0x0D00 0x0DFF	256	EALLOW 保護されている

(1) レジスタが EALLOW 保護されている場合、EALLOW 命令を実行するまではライトを実行できません。EDIS 命令は、ライトをディスエーブルにして、不正コードまたはポインタがレジスタの内容を破損させることを防止します。

(2) フラッシュ・レジスタもコード・セキュリティ・モジュール (CSM) によって保護されます。

表 5-2 ペリフェラル・フレーム 1 レジスタ

名前	アドレス範囲	サイズ (x16)	アクセス・タイプ ⁽¹⁾
eCANA レジスタ	0x6000 0x60FF	256 (128 × 32)	一部の eCAN 制御レジスタ (および、他の eCAN 制御レジスタ内の選択されたビット)は、EALLOW 保護されています。
eCANA メールボックス RAM	0x6100 0x61FF	256 (128 × 32)	EALLOW 保護されていない
eCANB レジスタ	0x6200 0x62FF	256 (128 × 32)	一部の eCAN 制御レジスタ (および、他の eCAN 制御レジスタ内の選択されたビット)は、EALLOW 保護されています。
eCANB メールボックス RAM	0x6300 0x63FF	256 (128 × 32)	EALLOW 保護されていない
ePWM1 レジスタ	0x6800 0x683F	64 (32 × 32)	一部の ePWM レジスタは、EALLOW 保護されています。 セクション 5.2 を参照してください。
ePWM2 レジスタ	0x6840 0x687F	64 (32 × 32)	一部の ePWM レジスタは、EALLOW 保護されています。 セクション 5.2 を参照してください。
ePWM3 レジスタ	0x6880 0x68BF	64 (32 × 32)	一部の ePWM レジスタは、EALLOW 保護されています。 セクション 5.2 を参照してください。
ePWM4 レジスタ	0x68C0 0x68FF	64 (32 × 32)	一部の ePWM レジスタは、EALLOW 保護されています。 セクション 5.2 を参照してください。
ePWM5 レジスタ	0x6900 0x693F	64 (32 × 32)	一部の ePWM レジスタは、EALLOW 保護されています。 セクション 5.2 を参照してください。
ePWM6 レジスタ	0x6940 0x697F	64 (32 × 32)	一部の ePWM レジスタは、EALLOW 保護されています。 セクション 5.2 を参照してください。
予約	0x6980 0x69FF	128	

(1) ペリフェラル・フレーム 1 は、16 ビットおよび 32 ビットのアクセスを許可します。すべての 32 ビット・アクセスは偶数アドレス境界にアラインされます。

表 5-2 ペリフェラル・フレーム 1 レジスタ (続き)

名前	アドレス範囲	サイズ (x16)	アクセス・タイプ ⁽¹⁾
eCAP1 レジスタ	0x6A00 0x6A1F	32 (16 × 32)	EALLOW 保護されていない
eCAP2 レジスタ	0x6A20 0x6A3F	32 (16 × 32)	EALLOW 保護されていない
eCAP3 レジスタ	0x6A40 0x6A5F	32 (16 × 32)	EALLOW 保護されていない
eCAP4 レジスタ	0x6A60 0x6A7F	32 (16 × 32)	EALLOW 保護されていない
予約	0x6A80 0x6AFF	32 (16 × 32)	EALLOW 保護されていない
eQEP1 レジスタ	0x6B00 0x6B3F	64 (32 × 32)	EALLOW 保護されていない
eQEP2 レジスタ	0x6B40 0x6B7F	64 (32 × 32)	EALLOW 保護されていない
GPIO 制御レジスタ	0x6F80 0x6FBF	128 (64 × 32)	EALLOW 保護されている
GPIO データ・レジスタ	0x6FC0 0x6FDF	32 (16 × 32)	EALLOW 保護されていない
GPIO 割り込みおよび LPM セレクト・レジスタ	0x6FE0 0x6FFF	32 (16 × 32)	EALLOW 保護されている

表 5-3 ペリフェラル・フレーム 2 レジスタ

名前	アドレス範囲	サイズ (x16)	アクセス・タイプ ⁽¹⁾
システム制御レジスタ	0x7010 0x702F	32	EALLOW 保護されている
SPI-A レジスタ	0x7040 0x704F	16	EALLOW 保護されていない
SCI-A レジスタ	0x7050 0x705F	16	EALLOW 保護されていない
ADC レジスタ	0x7100 0x711F	32	EALLOW 保護されていない
SPI-B レジスタ	0x7740 0x774F	16	EALLOW 保護されていない
SCI-B レジスタ	0x7750 0x775F	16	EALLOW 保護されていない
SPI-C レジスタ	0x7760 0x776F	16	EALLOW 保護されていない
SPI-D レジスタ	0x7780 0x778F	16	EALLOW 保護されていない
I2C レジスタ	0x7900 0x793F	64	EALLOW 保護されていない

(1) ペリフェラル・フレームは 16 ビット・アクセスのみ実行できます。すべての 32 ビット・アクセスは無視されます (無効なデータが返される、またはライトされる可能性があります)。

5.2 EALLOW 保護されているレジスタ

280x デバイス上のいくつかの制御レジスタは、EALLOW 保護メカニズムによって誤った CPU ライトから保護されます。表 5-4 に示されているように、ステータス・レジスタ 1 (ST1) の EALLOW ビットは保護の状態かどうかを示します。

表 5-4 EALLOW 保護されているレジスタへのアクセス

EALLOW ビット	CPU ライト	CPU リード	JTAG ライト	JTAG リード
0	無視される	許可される	許可される (1)	許可される
1	許可される	許可される	許可される	許可される

(1) EALLOW ビットは、JTAG ポートを介して無効にされます。これにより、Code Composer Studio インターフェイスからのデバッグ時には、保護されたレジスタへのフル・アクセスが許可されます。

リセット時には、EALLOW ビットがクリアされ、EALLOW 保護がイネーブルになります。保護されている間は、CPU による保護されたレジスタへのライトはすべて無視され、CPU リード、JTAG リード、および JTAG ライトだけが許可されます。EALLOW 命令の実行によって、このビットがセットされた場合、CPU は保護されたレジスタに自由にライトすることを許可されます。レジスタの変更後は、EDI 命令を実行して EALLOW ビットをクリアすることにより、それらのレジスタをもう一度保護することができます。

以下のレジスタは、EALLOW 保護されています。

- デバイス・エミュレーション・レジスタ
- フラッシュ・レジスタ
- CSM レジスタ
- PIE ベクタ・テーブル
- システム制御レジスタ
- GPIO MUX レジスタ
- 特定の eCAN レジスタ

表 5-5 EALLOW 保護されているデバイス・エミュレーション・レジスタ

名前	アドレス	サイズ (x16)	説明
DEVICECNF	0x0880 0x0881	2	デバイス構成レジスタ
PROTSTART	0x0884	1	ブロック保護開始アドレス・レジスタ
PROTRANGE	0x0885	1	ブロック保護範囲アドレス・レジスタ

表 5-6 EALLOW 保護されているフラッシュ /OTP コンフィギュレーション・レジスタ

名前	アドレス	サイズ (x16)	説明
FOPT	0x0A80	1	フラッシュ・オプション・レジスタ
FPWR	0x0A82	1	フラッシュ電力モード・レジスタ
FSTATUS	0x0A83	1	ステータス・レジスタ
FSTDBYWAIT	0x0A84	1	フラッシュ・スリープ・スタンバイ・ウェイト・ステート・レジスタ
FACTIVEWAIT	0x0A85	1	フラッシュ・スタンバイ・アクティブ・ウェイト・ステート・レジスタ
FBANKWAIT	0x0A86	1	フラッシュ・リード・アクセス・ウェイト・ステート・レジスタ
FOTPWAIT	0x0A87	1	OTP リード・アクセス・ウェイト・ステート・レジスタ

表 5-7 EALLOW 保護されているコード・セキュリティ・モジュール (CSM) レジスタ

レジスタ名	アドレス	サイズ (x16)	レジスタの説明
KEY0	0x0AE0	1	128 ビット・キー・レジスタの下位ワード
KEY1	0x0AE1	1	128 ビット・キー・レジスタの 2 番目のワード
KEY2	0x0AE2	1	128 ビット・キー・レジスタの 3 番目のワード
KEY3	0x0AE3	1	128 ビット・キー・レジスタの 4 番目のワード
KEY4	0x0AE4	1	128 ビット・キー・レジスタの 5 番目のワード
KEY5	0x0AE5	1	128 ビット・キー・レジスタの 6 番目のワード
KEY6	0x0AE6	1	128 ビット・キー・レジスタの 7 番目のワード
KEY7	0x0AE7	1	128 ビット・キー・レジスタの上位ワード
CSMSCR	0x0AEF	1	CSM ステータスおよび制御レジスタ

表 5-8 EALLOW 保護されている PIE ベクタ・テーブル

名前	アドレス	サイズ (x16)	説明
未使用	0x0D00	2	予約
	0x0D02		
	0x0D04		
	0x0D06		
	0x0D08		
	0x0D0A		
	0x0D0C		
	0x0D0E		
	0x0D10		
	0x0D12		
	0x0D14		
	0x0D16		
	0x0D18		
INT13	0x0D1A	2	外部割り込み 13 (XINT13) または CPU タイマ 1 (RTOS 用)
INT14	0x0D1C	2	CPU タイマ 2 (RTOS 用)
DATALOG	0x0D1E	2	CPU データ・ロギング割り込み
RTOSINT	0x0D20	2	CPU リアルタイム OS 割り込み
EMUINT	0x0D22	2	CPU エミュレーション割り込み
NMI	0x0D24	2	外部ノンマスクابل割り込み
ILLEGAL	0x0D26	2	不正操作
USER1	0x0D28	2	ユーザー定義トラップ
.	.	.	.
USER12	0x0D3E	2	ユーザー定義トラップ
INT1.1	0x0D40	2	グループ 1 割り込みベクタ
.	.	.	.
INT1.8	0x0D4E	2	グループ 2 割り込みベクタ～グループ 11 割り込みベクタ
.	.	.	.
.	.	.	.
.	.	.	.
INT12.1	0x0DF0	2	グループ 12 割り込みベクタ
INT12.8	0x0DFE	2	

表 5-9 EALLOW 保護されている PLL、クロック、ウォッチドッグ、および低電力モード・レジスタ

名前	アドレス	サイズ (x16)	説明
XCLK	0x7010	1	XCLKOUT ピン制御、X1、および XCLKIN ステータス・レジスタ
PLLSTS	0x7011	1	PLL ステータス・レジスタ
HSPCP	0x701A	1	HSPCLK クロック用の高速ペリフェラル・クロック・プリスケアラ・レジスタ
LOSPCP	0x701B	1	HSPCLK クロック用の低速ペリフェラル・クロック・プリスケアラ・レジスタ
PCLKCR0	0x701C	1	ペリフェラル・クロック制御レジスタ 0
PCLKCR1	0x701D	1	ペリフェラル・クロック制御レジスタ 1
LPMCRO	0x701E	1	低電力モード制御レジスタ 0
PLLCR	0x7021	1	PLL 制御レジスタ
SCSR	0x7022	1	システム制御およびステータス・レジスタ
WDCNTR	0x7023	1	ウォッチドッグ・カウンタ・レジスタ
WDKEY	0x7025	1	ウォッチドッグ・リセット・キー・レジスタ
WDCR	0x7029	1	ウォッチドッグ制御レジスタ

表 5-10 EALLOW 保護されている GPIO MUX レジスタ

名前	アドレス	サイズ (x16)	説明
GPACTRL	0x6F80	2	GPIO A 制御レジスタ (GPIO0 ~ GPIO31)
GPAQSEL1	0x6F82	2	GPIO A フィルタ (Qualifier) セレクト 1 レジスタ (GPIO0 ~ GPIO15)
GPAQSEL2	0x6F84	2	GPIO A フィルタ (Qualifier) セレクト 2 レジスタ (GPIO16 ~ GPIO31)
GPAMUX1	0x6F86	2	GPIO A Mux 1 レジスタ (GPIO0 ~ GPIO15)
GPAMUX2	0x6F88	2	GPIO A Mux 2 レジスタ (GPIO16 ~ GPIO31)
GPADIR	0x6F8A	2	GPIO A 方向レジスタ (GPIO0 ~ GPIO31)
GPAPUD	0x6F8C	2	GPIO A プルアップ・ディスエーブル・レジスタ (GPIO0 ~ GPIO31)
GPBCTRL	0x6F90	2	GPIO B 制御レジスタ (GPIO32 ~ GPIO35)
GPBQSEL1	0x6F92	2	GPIO B フィルタ (Qualifier) セレクト 1 レジスタ (GPIO32 ~ GPIO35)
GPBQSEL2	0x6F94	2	予約
GPBMUX1	0x6F96	2	GPIO B Mux 1 レジスタ (GPIO32 ~ GPIO35)
GPBMUX2	0x6F98	2	予約
GPBDIR	0x6F9A	2	GPIO B 方向レジスタ (GPIO32 ~ GPIO35)
GPBPUD	0x6F9C	2	GPIO B プルアップ・ディスエーブル・レジスタ (GPIO32 ~ GPIO35)
GPIOXINT1SEL	0x6FE0	1	XINT1 GPIO 入力セレクト・レジスタ (GPIO0 ~ GPIO31)
GPIOXINT2SEL	0x6FE1	1	XINT2 GPIO 入力セレクト・レジスタ (GPIO0 ~ GPIO31)
GPIOXNMISEL	0x6FE2	1	XNMI GPIO 入力セレクト・レジスタ (GPIO0 ~ GPIO31)
GPIOLPMSEL	0x6FE8	2	LPM GPIO セレクト・レジスタ (GPIO0 ~ GPIO31)

表 5-11 EALLOW 保護されている eCAN-A レジスタ

名前	eCAN-A アドレス	eCAN-B アドレス	サイズ (x16)	説明
CANMC	0x6014	0x6214	2	マスタ制御レジスタ (1)
CANBTC	0x6016	0x6216	2	ビット・タイミング・コンフィギュレーション・レジスタ (2)
CANGIM	0x6020	0x6220	2	グローバル割り込みマスク・レジスタ (3)
CANMIM	0x6024	0x6224	2	メールボックス割り込みマスク・レジスタ
CANTSC	0x602E	0x622E	2	タイム・スタンプ・カウンタ
CANTIOC	0x602A	0x622A	1	CANTXA ピンの I/O 制御レジスタ (4)
CANRIOC	0x602C	0x622C	1	CANRXA ピンの I/O 制御レジスタ (5)

- (1) ビット CANMC[15-9] および [7-6] だけが保護されます。
 (2) ビット BCR[23-16] および [10-0] だけが保護されます。
 (3) ビット CANGIM[17-16]、[14-8]、および [2-0] だけが保護されます。
 (4) IOCNT1[3] だけが保護されます。
 (5) IOCNT2[3] だけが保護されます。

表 5-12 EALLOW 保護されている ePWM1 ~ ePWM3 レジスタ

名前	ePWM1 アドレス	ePWM2 アドレス	ePWM3 アドレス	サイズ x16	レジスタの説明
TZSEL	0x6812	0x6852	0x6892	1	トリップ・ゾーン・セレクト・レジスタ
TZCTL	0x6814	0x6854	0x6894	1	トリップ・ゾーン制御レジスタ
TZEINT	0x6815	0x6855	0x6895	1	トリップ・ゾーン・イネーブル割り込みレジスタ
TZCLR	0x6817	0x6857	0x6897	1	トリップ・ゾーン・クリア・レジスタ
TZFRC	0x6818	0x6858	0x6898	1	トリップ・ゾーン強制レジスタ
HRCNFG	0x6820	0x6860	0x68A0	1	HRPWM コンフィギュレーション・レジスタ

表 5-13 EALLOW 保護されている ePWM4 ~ ePWM6 レジスタ

名前	ePWM4 アドレス	ePWM5 アドレス	ePWM6 アドレス	サイズ x16	レジスタの説明
TZSEL	0x68D2	0x6912	0x6952	1	トリップ・ゾーン・セレクト・レジスタ
TZCTL	0x68D4	0x6914	0x6954	1	トリップ・ゾーン制御レジスタ
TZEINT	0x68D5	0x6915	0x6955	1	トリップ・ゾーン・イネーブル割り込みレジスタ
TZCLR	0x68D7	0x6917	0x6957	1	トリップ・ゾーン・クリア・レジスタ
TZFRC	0x68D8	0x6918	0x6958	1	トリップ・ゾーン強制レジスタ
HRCNFG	0x68E0	N/A	N/A	1	HRPWM コンフィギュレーション・レジスタ

5.3 デバイス・エミュレーション・レジスタ

これらのレジスタは、C28x CPU の保護モードを制御する場合と、いくつかの重要なデバイス信号を監視する場合に使用されます。これらのレジスタは、表 5-14 で定義されています。

表 5-14 デバイス・エミュレーション・レジスタ

名前	アドレス	サイズ (x16)	説明
DEVICECNF	0x0880 0x0881	2	デバイスコンフィギュレーションレジスタ
PARTID	0x0882	1	パート ID レジスタ
REVID	0x0883	1	リビジョン ID レジスタ
PROTSTART	0x0884	1	ブロック保護開始アドレス・レジスタ
PROTRANGE	0x0885	1	ブロック保護範囲アドレス・レジスタ

図 5-1. デバイス構成 (DEVICECNF) レジスタ

31		20	19	18	17	16
	予約		ENPROT		予約	
	R-0		R/W-1		R-111	
15		6	5	4	3	2
	予約		XRS	予約	VMAPS	予約
	R-0		R-P	R-0	R-1	R/W-011

凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 5-15 DEVICECNF レジスタのフィールドの説明

ビット	フィールド	値	説明
31-20	予約		予約
19	ENPROT		ライト・リード保護モード・ビットをイネーブルにします。
		0	ライト・リード保護モードをディスエーブルにします。
		1	PROTSTART および PROTRANGE レジスタによって指定されているライト・リード保護をイネーブルにします。
18-6	予約		予約
5	XRS		リセット入力信号ステータス。これは、XRS 入力ピンに直接接続されます。
4	予約		予約
3	VMAPS		VMAP 設定ステータス。これは、VMAP のステータスを示します。
2-0	予約		予約

図 5-2. パート ID レジスタ

15	8	7	0
PARTTYPE		PARTNO	
R		R	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 5-16 PARTID レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	値	説明
7-0	PARTNO	R	(1)		これらの 8 ビットは、以下のどのデバイスの機能セットかを示します。
				0x2C	2801
				0x34	2806
				0x3C	2808
					他のすべての値は予約されているか、または他のデバイスによって使用されます。
15-8	PARTTYPE	R	(1)		これらの 8 ビットは、フラッシュベースか ROM ベースかのデバイスのタイプを示します。
				0x00	フラッシュベースのデバイス。
				0xFF	ROM ベースのデバイス。
					他のすべての値は予約されています。

(1) レジスタの説明で示されているように、リセット値はデバイスに依存します。

図 5-3. REVID レジスタ

15	REVID	0
R		

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 5-17 REVID レジスタのフィールドの説明

ビット	フィールド	リセット	説明
15-0	REVID		(1) これらの 16 ビットは、シリコン・リビジョン番号を示します。この番号は、シリコンの最初のリビジョンで常に 0x0000 から始まり、任意の後続リビジョンでインクリメントされます。
		0x0000	リビジョン 0 (最初のシリコンの場合)
		0x0001	リビジョン A
		0x0002	リビジョン B など

(1) レジスタ・フィールドの説明に記載されているように、リセット値はシリコン・リビジョンに依存します。

5.4 ライト後のリード保護

PROTSTART および PROTRANGE レジスタは、CPU の「ライト後のリード」動作が保護されるメモリ・アドレス範囲を設定します (パイプライン順ではなくコードの順番どおりに動作する)。これは、特定のペリフェラル動作に対して必要な保護です。

例：以下のコード行はレジスタ 1 (REG1) ロケーションへのライトを実行し、その次の命令はレジスタ 2 (REG2) ロケーションからのリードを実行します。ブロック保護がディスエーブルになっているプロセッサ・メモリ・バスでは、以下のようにリード動作はライトの前に発行されます。

```
MOV @REG1,AL          -----+
TBIT @REG2,#BIT_X    -----|-----> Read
                        +-----> Write
```

ライト後のリード保護

ブロック保護がイネーブルにされている場合は、以下のようにリードはライトが発生するまでストールされます。



表 5-18 PROTSTART および PROTRANGE レジスタ

名前	アドレス	サイズ	タイプ	リセット	説明
PROTSTART	0x0884	16	R/W	0x0100(1)	PROTSTART レジスタは、プロセッサの 22 ビット・アドレスの上位 16 ビットの開始アドレスを設定します。したがって、最小分解能は 64 ワードです。
PROTRANGE	0x0885	16	R/W	0x00FF(1)	PROTRANGE レジスタは、開始アドレスからのブロック・サイズを設定します。このブロック・サイズは、64 ワードで始まり、2 進倍数 (64、128、256、512、1K、2K、4K、8K、16K、.....、2M) によってインクリメントされます。

(1) リセット時のこれらのレジスタのデフォルト値は、メモリ・マップ (アドレス範囲 0x4000 ~ 0x8000) のペリフェラル・フレーム 1、ペリフェラル・フレーム 2、および XINTF ゾーン 1 領域をカバーするように選択されます。

表 5-19 PROTSTART の有効値

開始アドレス	レジスタ名	レジスタ・ビット ⁽¹⁾															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000 0000	0x0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0000 0040	0x0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x0000 0080	0x0002	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x0000 00C0	0x0003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0x003F FF00	0xFFFC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0x003F FF40	0xFFFD	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0x003F FF80	0xFFFE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0x003F FFC0	0xFFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(1) レジスタ値を計算するための最も迅速な方法は、必要なブロック開始アドレスを 64 で割ることです。

表 5-20 PROTRANGE の有効値

ブロック・サイズ	レジスタ名	レジスタ・ビット ⁽¹⁾															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64	0x0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
128	0x0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
256	0x0003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
512	0x0007	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1K	0x000F	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
256K	0x0FFF	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
512K	0x1FFF	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1M	0x3FFF	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2M	0x7FFF	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4M	0xFFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(1) すべてのレジスタ値が有効なわけではありません。PROTSTART アドレス値は、範囲値の倍数でなければなりません。たとえば、ブロック・サイズが 4K に設定された場合、開始アドレスにすることができる位置は任意の 4K 境界だけです。

ペリフェラル割り込み拡張 (PIE)

ペリフェラル割り込み拡張 (PIE) ブロックは、多数の割り込みソースをより小さなセットの割り込み入力にマルチプレクスします。PIE ブロックは、8 つのグループに分類される 96 の個別割り込みをサポートできます。各グループは、12 のコア割り込みライン (INT1 ~ INT12) の 1 つに接続されます。96 の割り込みのそれぞれは、変更可能な専用 RAM ブロックに格納されている固有のベクトルによってサポートされます。CPU は、割り込みの処理時に、対応する割り込みベクタを自動的にフェッチします。ベクタをフェッチし、クリティカルな CPU レジスタを保存するには、9 CPU クロック・サイクルかかります。したがって、CPU は割り込みイベントにすぐに応答できます。割り込みの優先順位付けは、ハードウェアとソフトウェアによって制御されます。各個別割り込みは、PIE ブロック内でイネーブル/ディスエーブルにすることができます。

項目	ページ
6.1 PIE コントローラの概要	116
6.2 ベクタ・テーブル・マッピング	119
6.3 割り込みソース	121
6.4 PIE コンフィギュレーション・レジスタ	134
6.5 PIE 割り込みレジスタ	135
6.6 外部割り込み制御レジスタ	144

6.1 PIE コントローラの概要

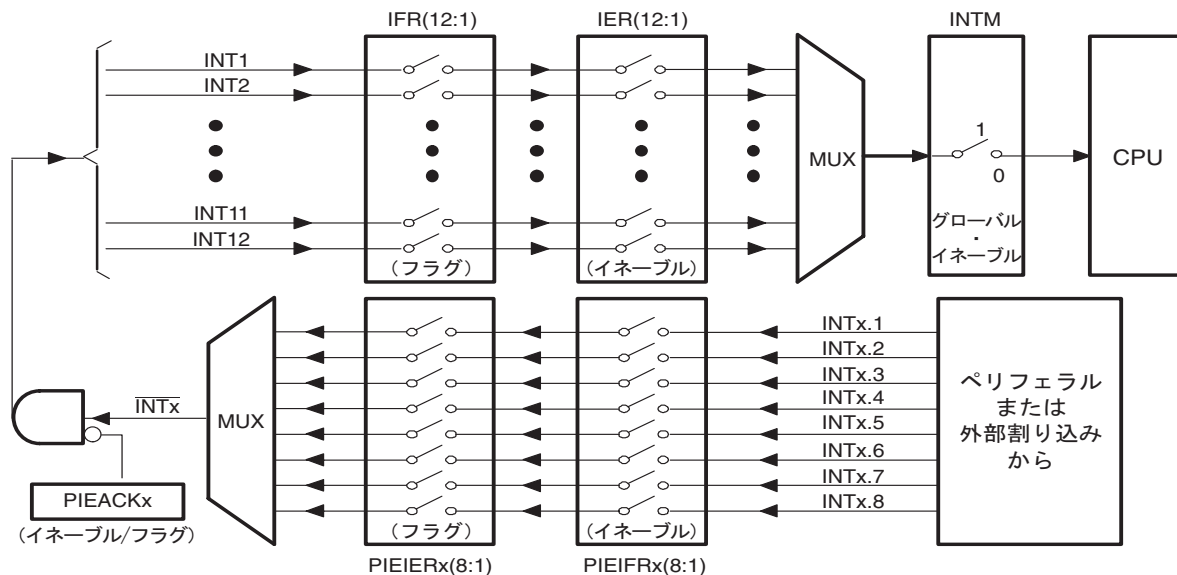
28x CPU は、1 つのノンマスカブル割り込み (NMI) と 16 のマスカブル優先順位付き割り込みリクエスト (INT1 ~ INT14, RTOSINT, および DLOGINT) を CPU レベルでサポートします。28x デバイスは多数のペリフェラルを持っており、各ペリフェラルは、ペリフェラル・レベルで多数のイベントに応じて 1 つ以上の割り込みを生成することができます。CPU はすべてのペリフェラル割り込みリクエストを CPU レベルで処理するための十分な能力を持っていないため、各種ソース (ペリフェラルや他の外部ピンなど) からの割り込みリクエストを調停するには、ペリフェラル割り込み拡張 (PIE) コントローラが必要です。

PIE ベクタ・テーブルは、システム内の各割り込みサービス・ルーチン (ISR) のアドレス (ベクタ) を格納するために使用されます。割り込みソース (すべてのマルチプレクスされた割り込みとマルチプレクスされていない割り込みを含む) ごとに 1 つのベクタが存在します。ユーザーは、デバイスの初期化時にベクタ・テーブルを埋めます。また、ユーザーは動作中にそれを更新できます。

6.1.1 割り込み動作シーケンス

図 6-1 は、すべての共用 PIE 割り込みに対する割り込み動作シーケンスの概要を示します。マルチプレクスされない割り込みソースは、CPU に直接供給されます。

図 6-1 概要 : PIE ブロックを使用した割り込みのマルチプレクス



• ペリフェラル・レベル

割り込み生成イベントはペリフェラル内で発生します。そのイベントに対応する割り込みフラグ (IF) ビットが、そのペリフェラルのレジスタにてセットされます。

対応する割り込みイネーブル (IE) がセットされた場合、ペリフェラルは PIE コントローラへの割り込みリクエストを生成します。割り込みがペリフェラル・レベルでイネーブルにされた場合、IF はソフトウェアによってクリアされるまでセットされたままになります。割り込みが後でイネーブルにされ、割り込みフラグがまだセットされている場合は、割り込みリクエストが PIE に要求されます。

ペリフェラル・レジスタ内の割り込みフラグは、手動でクリアする必要があります。特定のペリフェラルの詳細については、ペリフェラル・リファレンス・ガイドを参照してください。

- **PIE レベル**

PIE ブロックは、8つのペリフェラルおよび外部ピン割り込みを1つのCPU割り込みにマルチプレクスしています。これらの割り込みは、12のグループ（PIEグループ1～PIEグループ12）に分類されます。グループ内の割り込みは、1つのCPU割り込みにマルチプレクスされています。たとえば、PIEグループ1はCPU割り込み1（INT1）にマルチプレクスされ、PIEグループ12はCPU割り込み12（INT12）にマルチプレクスされます。残りのCPU割り込みに接続されている割り込みソースはマルチプレクスされません。マルチプレクスされていない割り込みの場合は、PIEはリクエストをCPUに直接渡します。

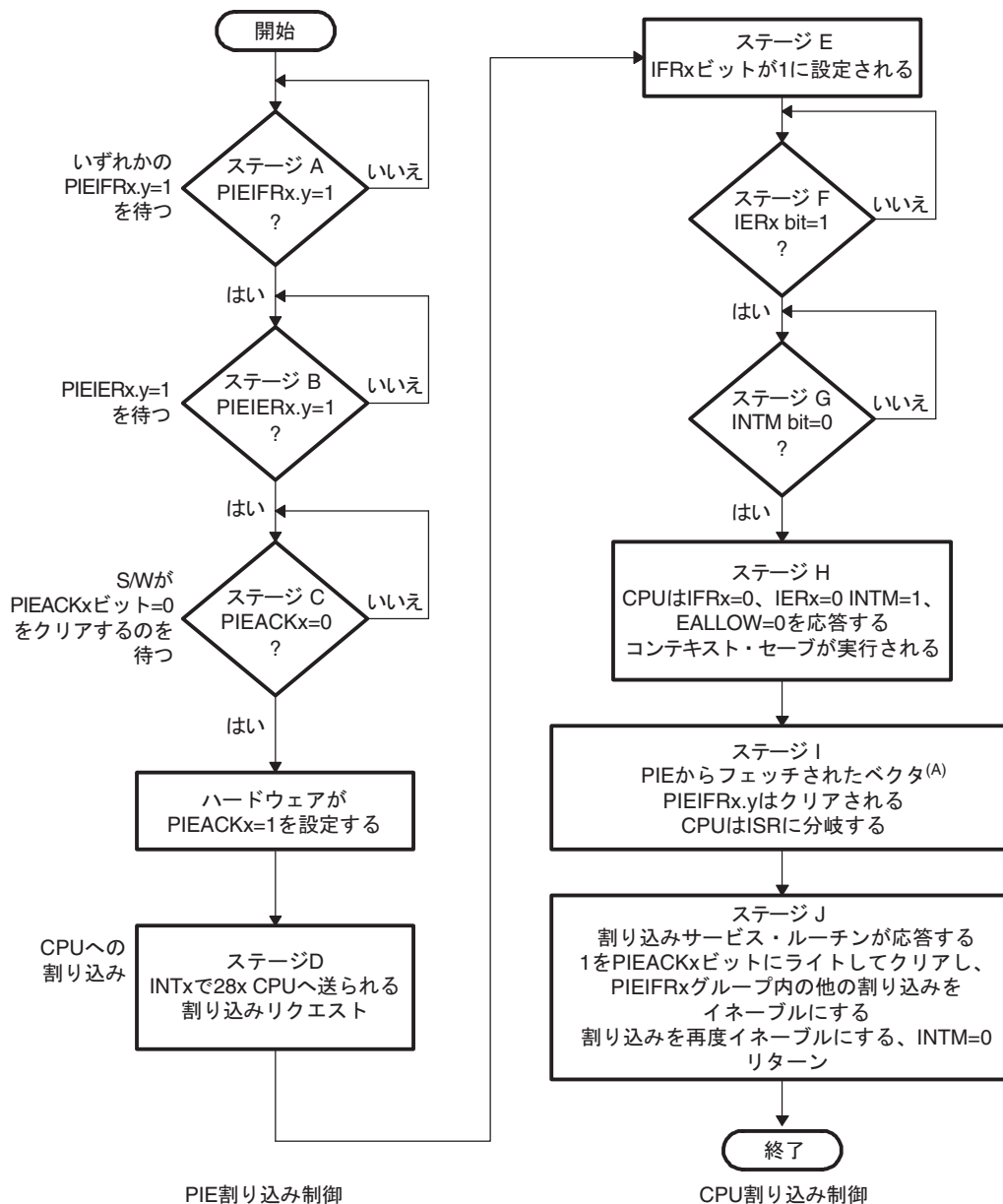
マルチプレクスされた割り込みソースの場合、PIEブロック内の各割り込みグループは、対応するフラグ・レジスタ（PIEIFRx）とイネーブル（PIEIERx）レジスタ（ x =PIEグループ1～PIEグループ12）を持ちます。各ビット（ y と呼ばれる）は、グループ内の8つのマルチプレクスされた割り込みの1つに対応します。したがって、PIEIFRx.yとPIEIERx.yは、PIEグループ x （ $x=1\sim 12$ ）内の割り込み y （ $y=1\sim 8$ ）に対応します。また、PIEACKx（ $x=1\sim 12$ ）と呼ばれるPIE割り込みグループごとに、1つのACKビット（PIEACK）があります。図6-2は、さまざまなPIEIFRおよびPIEIERレジスタ条件下でのPIEハードウェアの動作を表します。

リクエストがPIEコントローラに対して行われると、対応するPIE割り込みフラグ（PIEIFRx.y）ビットがセットされます。PIE割り込みイネーブル（PIEIERx.y）ビットも特定の割り込みに対してセットされている場合、PIEは対応するPIEACKxビットをチェックして、CPUがそのグループからの割り込みに対して準備ができているかどうかを確認します。PIEACKxビットがそのグループに対してクリアされた場合、PIEは割り込みリクエストをCPUに送ります。PIEACKxがセットされた場合、PIEはそれがクリアされるまで待ってから、INTxのリクエストを送ります。詳細については、セクション6.3を参照してください。

- **CPU レベル**

リクエストがCPUへ送られると、INTxに対応するCPUレベルの割り込みフラグ（IFR）ビットがセットされます。フラグがIFRでラッチされた後は、対応する割り込みは、CPU割り込みイネーブル（IER）レジスタまたはデバッグ割り込みイネーブル・レジスタ（DBGIER）とグローバル割り込みマスク（INTM）ビットで適切にイネーブルにされるまでは処理されません。

図 6-2 典型的な PIE/CPU 割り込み応答 - INTx.y



A 注：マルチプレクスされた割り込みの場合、PIE は、フラグが立ち、イネーブルとなっている最も高い優先順位の割り込みに応答します。フラグが立ちイネーブルになっている割り込みが存在しない場合は、グループ内の最も高い優先順位の割り込み (INTx.1。x は PIE グループ) が使用されます。詳細については、[セクション 6.3.3](#) を参照してください。

表 6-1 に示されているように、マスカブル割り込みを CPU レベルでイネーブルにするための必要条件は、使用されている割り込み処理プロセスに依存します。ほとんどの場合に発生する標準プロセスでは、DBGIER レジスタは使用されません。28x がリアルタイム・エミュレーション・モードになっていて、CPU が停止している場合は、異なるプロセスが使用されます。この特殊なケースでは、DBGIER が使用され、INTM ビットは無視されます。DSP がリアルタイム・モードになっていて、CPU が動作している場合は、標準割り込み処理プロセスが適用されます。

表 6-1 割り込みのイネーブル

割り込み処理プロセス	割り込みがイネーブルになる条件
標準	INTM=0、および IER のビットが 1
リアルタイム・モードで停止されている DSP	IER のビットが 1、および DBGIER が 1

その後で、CPU は割り込みを処理するための準備を整えます。この準備プロセスは、『TMS320C28x DSP CPU and Instruction Set Reference Guide』（文献番号 SPRU430）に詳しく記載されています。この準備では、対応する CPU IFR および IER ビットがクリアされ、EALLOW および LOOP がクリアされ、INTM および DBGM がセットされ、パイプラインがフラッシュされ、リターン・アドレスが格納されて、自動コンテキスト・セーブが実行されます。その後、ISR のベクタは PIE モジュールからフェッチされます。割り込みリクエストがマルチプレクスされた割り込みから来る場合、PIE モジュールはグループ PIEIERx および PIEIFRx レジスタを使用して、どの割り込みを処理する必要があるのかをデコードします。このデコード・プロセスは、[セクション 6.3.3](#) に詳しく記載されています。

実行される割り込みサービス・ルーチンのアドレスは、PIE 割り込みベクタ・テーブルから直接フェッチされます。PIE 内の可能な 96 の割り込みごとに 1 つの 32 ビット・ベクタが存在します。PIE モジュール内の割り込みフラグ (PIEIFRx.y) は、割り込みベクタがフェッチされると自動的にクリアされます。ただし、割り込みグループに対応する PIE ACK ビットは、同じ PIE グループからの割り込みを受信する準備ができたときに手動でクリアされる必要があります。

6.2 ベクタ・テーブル・マッピング

28xx デバイスでは、割り込みベクタ・テーブルをメモリ内の 5 つの異なるロケーションにマップすることができます。実際には、PIE ベクタ・テーブル・マッピングだけが 280x デバイスに対して使用されます。

このベクタ・マッピングは、以下のモード・ビット / 信号によって制御されます。

VMAP: VMAP は、ステータス・レジスタ 1 ST1 (ビット 3) にあります。デバイス・リセットは、このビットを 1 に設定します。このビットの状態は、ST1 へのライトによって、または SETC/CLRC VMAP 命令によって変更できます。通常動作の場合は、このビットをセットしたままにしてください。

MOM1MAP: MOM1MAP は、ステータス・レジスタ 1 ST1 (ビット 11) にあります。デバイス・リセットは、このビットを 1 に設定します。このビットの状態は、ST1 へのライトによって、または SETC/CLRC MOM1MAP 命令によって変更できます。通常の 28xx デバイス動作の場合は、このビットはセットされたままでなければなりません。MOM1MAP=0 は、TI テスト専用予約されています。

ENPIE: ENPIE は、PIECTRL レジスタ (ビット 0) にあります。このビットのデフォルト値は、リセット時に 0 (PIE ディスエーブル) に設定されます。このビットの状態は、PIECTRL レジスタ (アドレス 0x0000 0CE0) へのライトによってリセット後に変更できます。

これらのビットと信号を使用して、可能なベクタ・テーブル・マッピングが[表 6-2](#)に示されています。

表 6-2 割り込みベクタ・テーブル・マッピング

ベクタ MAPS	ベクタのフェッチ元	アドレス範囲	VMAP	MOM1MAP	ENPIE
M1 ベクタ (1)	M1 SARAM ブロック	0x000000-0x00003F	0	0	X
M0 ベクタ (1)	M0 SARAM ブロック	0x000000-0x00003F	0	1	X
BROM ベクタ	ブート ROM ブロック	0x3FFFC0-0x3FFFFFF	1	X	0
PIE ベクタ	PIE ブロック	0x000D00-0x000DFF	1	X	1

(1) ベクタ・マップ M0 および M1 ベクタは予約されたモードのみです。28x デバイスでは、これらは SARAM として使用されます。

M1 および M0 ベクタ・テーブル・マッピングは、TI テスト専用予約されています。他のベクタ・マッピングを使用する場合、M0 および M1 メモリ・ブロックは SARAM ブロックとして扱われ、何の制約もなしに自由に使用できます。

デバイス・リセット動作の後には、ベクタ・テーブルは[表 6-3](#)に示されているようにマップされます。

表 6-3 リセット動作後のベクタ・テーブル・マッピング

ベクタ MAPS	ベクタのフェッチ元	アドレス範囲	VMAP ⁽¹⁾	MOM1MAP ⁽¹⁾	ENPIE ⁽¹⁾
BROM ベクタ ⁽²⁾	ブート ROM ブロック	0x3FFFC0-0x3FFFFFF	1	1	0

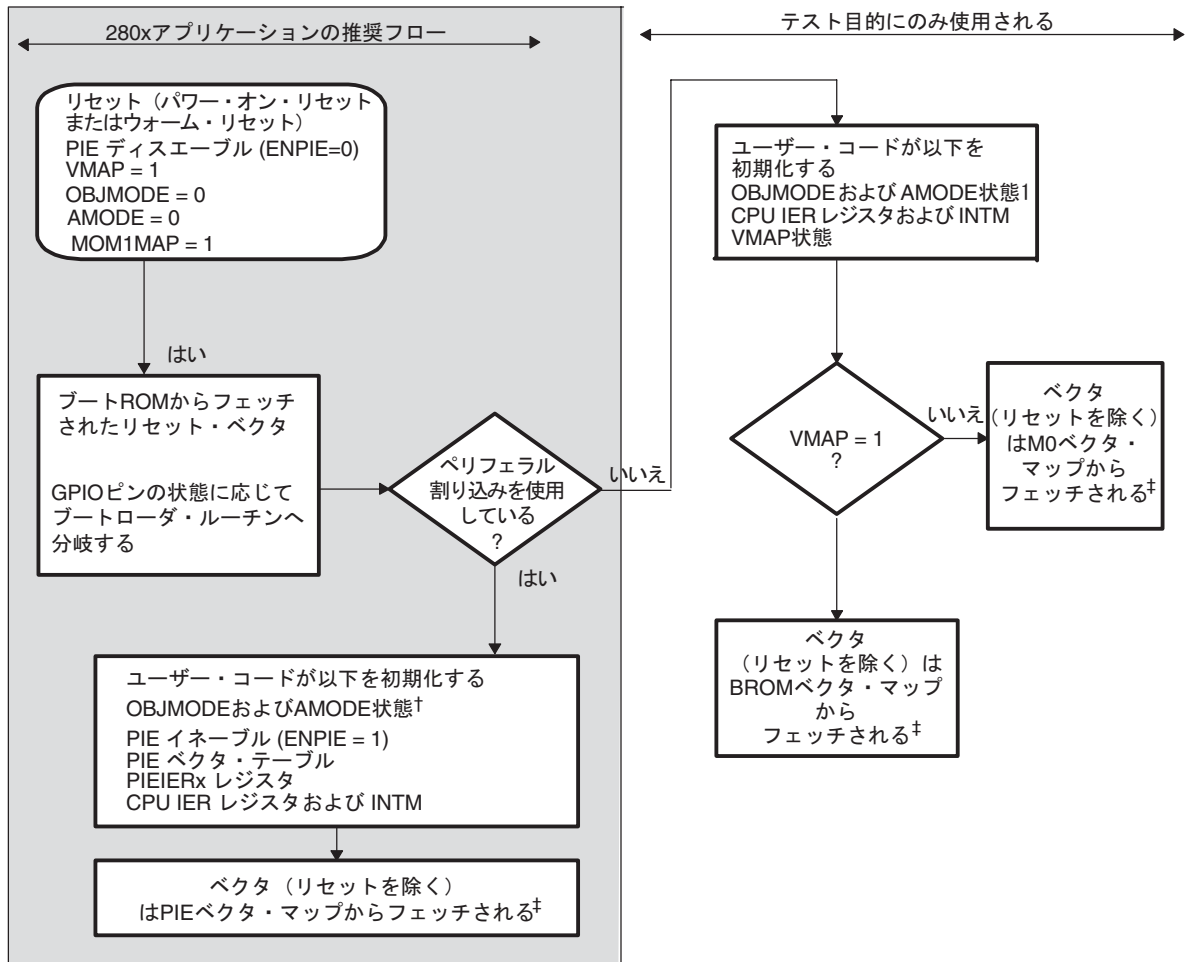
(1) 28x デバイスでは、VMAP および MOM1MAP モードはリセット時に 1 に設定されます。ENPIE モードは、リセット時に 0 に強制されます。

(2) リセット・ベクタは、常にブート ROM からフェッチされます。

リセットおよびブートの完了後は、PIE ベクタ・テーブルはユーザーのコードによって初期化される必要があります。その後で、アプリケーションは PIE ベクタ・テーブルをイネーブルにします。それ以降は、割り込みベクタは PIE ベクタ・テーブルからフェッチされます。注：リセットが発生すると、表 6-3 に示されているように、リセット・ベクタは常にベクタ・テーブルからフェッチされます。リセット後は、PIE ベクタ・テーブルは常にディスエーブルにされます。

図 6-3 は、ベクタ・テーブル・マッピングが選択されるプロセスを示します。

図 6-3 リセット・フロー・チャート



A 28x CPU の互換動作モードは、ステータス・レジスタ 1 (ST1) の OBJMODE および AMODE ビットの組み合わせによって決定されます。

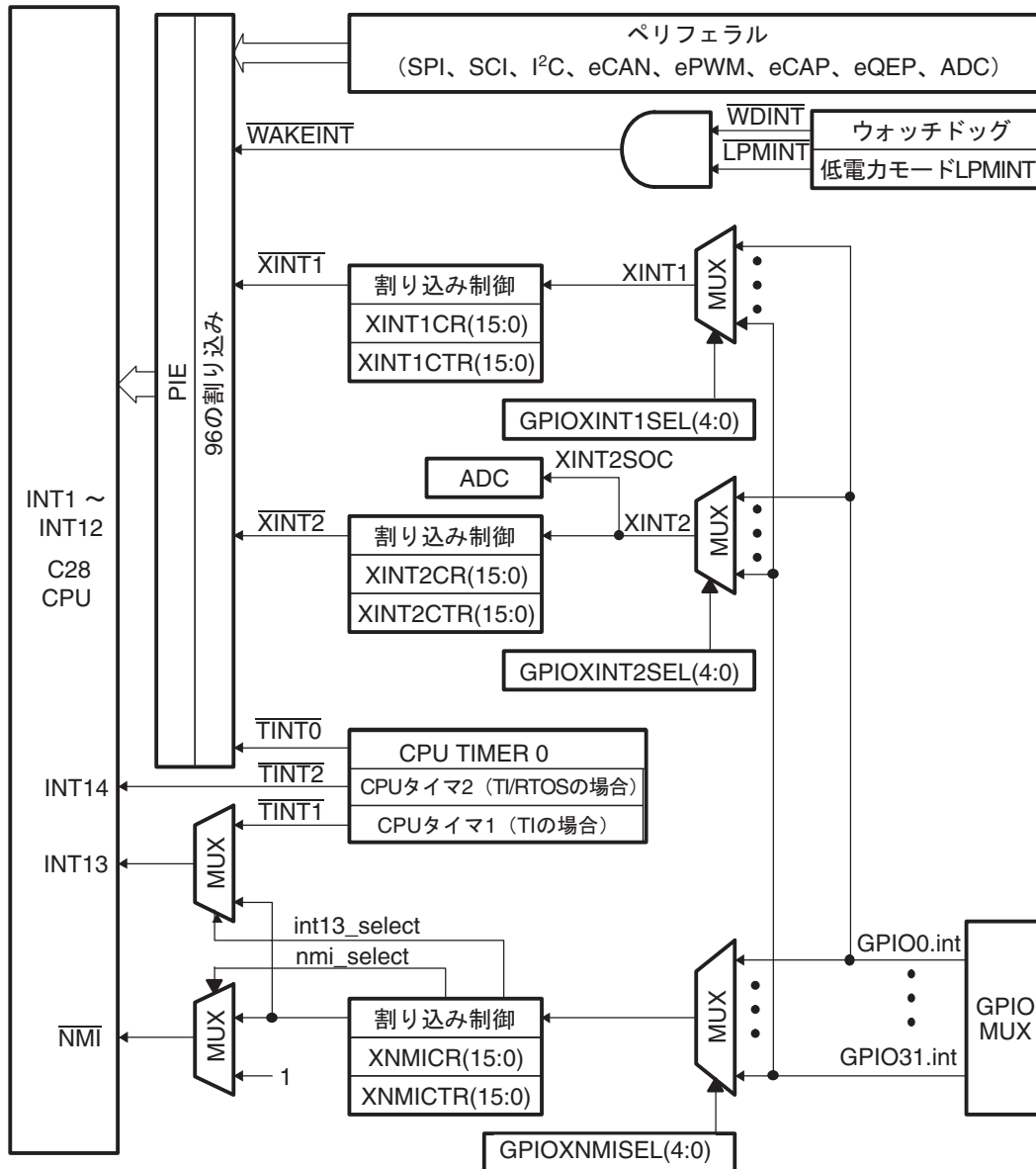
動作モード	OBJMODE	AMODE
C28x モード	1	0
C2xLP ソース互換	1	1
C27x オブジェクト互換	0	0 (リセット時のデフォルト)

B リセット・ベクタは、常にブート ROM からフェッチされます。

6.3 割り込みソース

図 6-4 は、さまざまな割り込みソースがどのように 280x デバイス内でマルチプレクスされるのを示します。このマルチプレクスの仕組みは、すべての 28x デバイスで全く同じとは限りません。詳細については、特定のデバイスのデータ・マニュアルを参照してください。

図 6-4 外部および PIE 割り込みソース



- A GPIO MUX では、XINT1、XINT2、および XNMI 信号は同期化され、ユーザーがプログラムできるクロック・サイクル数によって任意にフィルタリング (Qualification) されます。これは、入力ソースからグリッチを除去します。詳細については、[セクション 4.5](#)にある GPIO MUX 情報を参照してください。

6.3.1 マルチプレクスされた割り込みの処理手順

PIE モジュールは、8 つのペリフェラルおよび外部ピン割り込みを 1 つの CPU 割り込みにマルチプレクスしています。これらの割り込みは、12 のグループ (PIE グループ 1 ~ PIE グループ 12) に分類されます。各グループには、対応するイネーブル PIEIER およびフラグ PIEIFR レジスタがあります。これらのレジスタは、CPU への割り込みのフローを制御するために使用されます。また、PIE モジュールは、PIEIER および PIEIFR レジスタを使用して、どの割り込みサービス・ルーチンに CPU が分岐する必要があるのかをデコードします。

PIEIFR および PIEIER レジスタ内のビットをクリアするときには、以下の 3 つの主要ルールに従う必要があります。

ルール 1: ソフトウェアによって PIEIFR ビットを決してクリアしない

入ってくる割り込みは、PIEIFR レジスタへのライト動作または read-modify-write 動作が行われている間に失われることがあります。PIEIFR ビットをクリアするには、保留中の割り込みを処理する必要があります。通常のサービス・ルーチンを実行せずに PIEIFR ビットをクリアしたい場合は、以下の手順を使用してください。

1. EALLOW ビットをセットして、PIE ベクタ・テーブルに対する変更を許可します。
2. PIE ベクタ・テーブルを変更して、ペリフェラルのサービス・ルーチンに対するベクタがテンポラリー ISR を指すようにします。このテンポラリー ISR は、割り込みからのリターン (IRET) 動作だけを実行します。
3. 割り込みをイネーブルにして、その割り込みがテンポラリー ISR によって処理されるようにします。
4. テンポラリー割り込みルーチンの処理後には、PIEIFR ビットがクリアされます。
5. PIE ベクタ・テーブルを変更して、ペリフェラルのサービス・ルーチンを適切なサービス・ルーチンに再マッピングします。
6. EALLOW ビットをクリアします。

ルール 2: ソフトウェア優先順位付け割り込みの手順

『C280x C/C++ Header Files and Peripheral Examples in C』(文献番号 SPRC191) にある方法を使用してください。

1. CPU IER レジスタをグローバルな優先順位として使用し、個々の PIEIER レジスタをグループの優先順位に使用します。この場合、PIEIER レジスタは割り込み内でのみ変更されます。また、処理される割り込みと同じグループに対する PIEIER だけが変更されます。この変更は、PIEACK ビットが CPU への追加割り込みを抑制している間に行われます。
2. 関連のないグループからの割り込みを処理しているときには、グループに対する PIEIER ビットを決してディスエーブルにしないでください。

ルール 3: PIEIER を使用して割り込みをディスエーブルにする

PIEIER レジスタを使用して割り込みをイネーブルにし、その後でディスエーブルにした場合は、[セクション 6.3.2](#)に記載されている手順に従ってください。

6.3.2 マルチプレクスされたペリフェラル割り込みをイネーブルおよびディスエーブルにするための手順

割り込みをイネーブルまたはディスエーブルにするための正しい手順は、ペリフェラル割り込みイネーブル / ディスエーブル・フラグを使用する方法です。PIEIER および CPU IER レジスタの主な目的は、同じ PIE 割り込みグループ内の割り込みのソフトウェアによる優先順位付けです。ソフトウェア・パッケージ『C280x C/C++ Header Files and Peripheral Examples in C』（文献番号 SPRC191）には、このソフトウェアによる優先順位付け割り込みの方法を示す例が含まれています。

PIEIER レジスタ内のビットをこのコンテキストの外部でクリアする必要がある場合は、以下の 2 つの手順のいずれかに従ってください。最初の方法は、割り込みが失われないように、関連する PIE フラグ・レジスタを保存します。2 番目の方法は、関連する PIE フラグ・レジスタをクリアします。

方法 1：PIEIERx レジスタを使用して、割り込みをディスエーブルにし、関連する PIEIFRx フラグを保存します。

PIEIFRx レジスタ内の関連するフラグを保存している間に PIEIERx レジスタ内のビットをクリアするには、以下の手順に従う必要があります。

1. グローバル割り込みをディスエーブルにします (INTM=1)。
2. PIEIERx.y ビットをクリアして、特定のペリフェラルに対する割り込みをディスエーブルにします。これは、同じグループ内の 1 つ以上のペリフェラルに対して行うことができます。
3. 5 サイクル待ちます。この遅延は、CPU に入ってきた任意の割り込みのフラグが CPU IFR レジスタ内で立てられていることを確認するために必要です。
4. ペリフェラル・グループの CPU IFRx ビットをクリアします。これは、CPU IFR レジスタ上の安全な操作です。
5. ペリフェラル・グループの PIEACKx ビットをクリアします。
6. グローバル割り込みをイネーブルにします (INTM=0)。

方法 2：PIEIERx レジスタを使用して、割り込みをディスエーブルにし、関連する PIEIFRx フラグをクリアします。

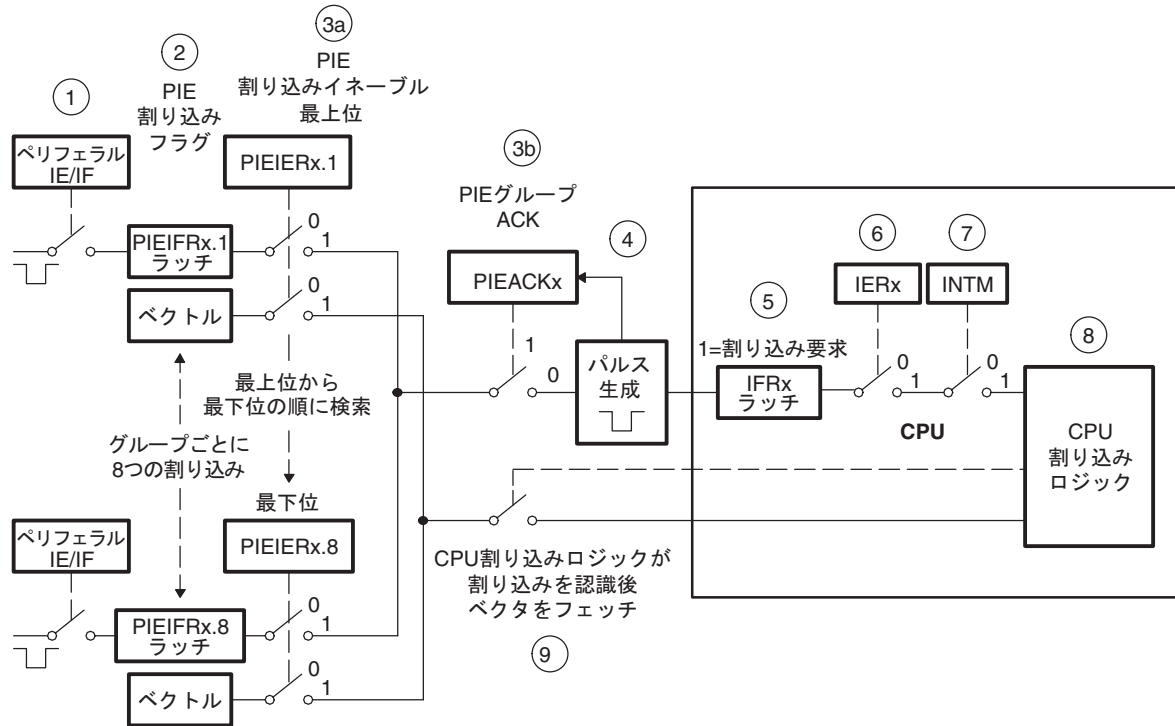
ペリフェラル割り込みのソフトウェア・リセットを実行し、PIEIFRx レジスタおよび CPU IFR レジスタ内の関連するフラグをクリアするには、以下の手順に従う必要があります。

1. グローバル割り込みをディスエーブルにします (INTM=1)。
2. EALLOW ビットをセットします。
3. PIE ベクタ・テーブルを変更して、対象となるペリフェラル割り込みのベクタを空の割り込みサービス・ルーチン (ISR) に一時的にマッピングします。この空の ISR は、割り込みからのリターン (IRET) 命令だけを実行します。これは、グループ内の他のペリフェラルからの割り込みを失うことなく単一の PIEIFRx.y ビットをクリアする安全な方法です。
4. ペリフェラル・レジスタでペリフェラル割り込みをディスエーブルにします。
5. グローバル割り込みをイネーブルにします (INTM=0)。
6. ペリフェラルからの保留中の割り込みが空の ISR ルーチンによって処理されるのを待ちます。
7. グローバル割り込みをディスエーブルにします (INTM=1)。
8. PIE ベクタ・テーブルを変更して、ペリフェラル・ベクタを元の ISR にマップします。
9. EALLOW ビットをクリアします。
10. 対象となるペリフェラルに対する PIEIER ビットをディスエーブルにします。
11. 対象となるペリフェラル・グループに対する IFR ビットをクリアします (これは、CPU IFR レジスタ上の安全な操作です)。
12. PIE グループの PIEACK ビットをクリアします。
13. グローバル割り込みをイネーブルにします。

6.3.3 ペリフェラルから CPU へのマルチプレクスされた割り込みリクエストのフロー

図 6-5 は、丸で囲んだ数字で示されたステップ付きのフローを表します。図に従って、ステップごとに説明します。

図 6-5 マルチプレクスされた割り込みリクエストのフロー・チャート



ステップ 1:

PIE グループ内の任意のペリフェラルまたは外部割り込みが割り込みを生成します。割り込みがペリフェラル・モジュール内でイネーブルにされた場合は、割り込みリクエストが PIE モジュールに送られます。

ステップ 2:

PIE モジュールは、PIE グループ x 内の割り込み y (INTx.y) が割り込みをアサートしたこと、および適切な PIE 割り込みフラグ・ビットがラッチされていること (PIEIFRx.y=1) を認識します。

ステップ 3:

割り込みリクエストが PIE から CPU へ送られるためには、以下の条件の両方が真でなければなりません。

3a: 適切なイネーブル・ビットがセットされている (PIEIERx.y=1)

3b: グループの PIEACKx ビットがクリアされている

ステップ 4:

3a と 3b の両方の条件が真の場合は、割り込みリクエストが CPU へ送られ、ACK ビットが再びセットされます (PIEACKx=1)。PIEACKx ビットは、それをクリアしてグループからの追加割り込みを PIE から CPU へ送ることができることを示すまでは、セットされたままになります。

ステップ 5:

CPU 割り込みフラグ・ビットは、保留中の割り込み x を CPU レベルで示すためにセットされます (CPU IFRx=1)。

ステップ 6 およびステップ 7:

CPU 割り込みがイネーブルにされて (CPU IER ビット x=1、または DBGIER ビット x=1)、グローバル割り込みマスクがクリアされた (INTM=0) 場合、CPU は INTx を処理します。

ステップ 8 :

CPU は割り込みを認識し、自動コンテキスト・セーブを実行し、IER ビットをクリアし、INTM をセットして、EALLOW をクリアします。割り込みの処理を準備するために CPU がとるステップのすべては、『TMS320C28x DSP CPU and Instruction Set Reference Guide』(文献番号 SPRU430) に記載されています。

ステップ 9 :

次に、CPU は PIE から適切なベクタを要求します。マルチプレクスされた割り込みの場合、PIE モジュールは PIEIERx および PIEIFRx レジスタの現在値を使用して、どのベクタ・アドレスを使用する必要があるのかをデコードします。以下の 2 つの可能なケースが存在します。

1. PIEIERx レジスタでイネーブルにする処理と PIEIFRx で保留中としてフラグを立てる処理の両方が行われるグループ内の最も高い優先順位の割り込みに対するベクタは、フェッチされて分岐アドレスとして使用されます。この方法では、さらに高い優先順位のイネーブルにされた割り込みがステップ 7 の後にフラグを立てられた場合、その割り込みが最初に処理されます。
2. グループ内のフラグを立てられた割り込みがどれもイネーブルにされていない場合は、PIE はそのグループ内の最も高い優先順位の割り込みに対するベクタで応答します。それは、INTx.1 に対して使用される分岐アドレスです。

この動作は、28x TRAP または INT 命令に相当します。

注 : PIEIERx レジスタは、どのベクタを分岐に使用するかを決定するために使用されるため、PIEIERx レジスタ内のビットをクリアするときには注意する必要があります。PIEIERx レジスタ内のビットをクリアするための適切な手順は、[セクション 6.3.2](#) に記載されています。これらのステップに従わないと、[図 6-5](#) のステップ 5 で割り込みが CPU に渡された後に PIEIERx レジスタに対して変更が発生することがあります。この場合、PIE は、保留中でイネーブルにされている他の割り込みが存在しない限り、あたかも TRAP または INT 命令が実行されたかのように応答します。

この時点で、PIEIFRx.y ビットはクリアされ、CPU は PIE からフェッチされた割り込みのベクタに分岐します。

6.3.4 PIE ベクタ・テーブル

PIE ベクタ・テーブル ([表 6-5](#) を参照) は、256 × 16 SARAM ブロックで構成されます。PIE ブロックが使用されていない場合に (データ空間でのみ) RAM として使用することもできます。PIE ベクタ・テーブルの内容は、リセット時には不定です。CPU の INT1 ~ INT12 の割り込み優先順位を固定です。PIE は、8 つの割り込みの各グループに対する優先順位を制御します。たとえば、INT1.1 が INT8.1 と同時に発生する必要がある場合は、両方の割り込みが PIE ブロックによって同時に CPU へ提示し、CPU は INT1.1 を最初に処理します。INT1.1 が INT1.8 と同時に発生する必要がある場合は、INT1.1 が CPU へ最初に送られ、その後 INT1.8 が続きます。割り込みの優先順位付けは、割り込み処理のベクタ・フェッチ部分の間に実行されます。

TRAP 1 ~ TRAP 12 命令または INTR INT1 ~ INTR INT12 命令は、各グループの最初のロケーション (INTR1.1 ~ INTR12.1) からベクタをフェッチします。同様に、それぞれの割り込みフラグがセットされている場合、OR IFR、#16 ビット操作を使用すると、INTR1.1 ~ INTR12.1 ロケーションからベクタがフェッチされます。他のすべての TRAP、INTR、OR IFR、#16 ビット操作は、それぞれのテーブル・ロケーションからベクタをフェッチします。INTR1 ~ INTR12 については、そのような操作の使用を避ける必要があります。TRAP #0 操作は、0x000000 のベクタ値を返します。ベクタ・テーブルは EALLOW 保護されています。

[表 6-4](#) にある可能な 96 のマルチプレクスされた割り込みの内、43 の割り込みが現在使用されています。残りの割り込みは、将来のデバイス用に予約されています。グループ内のどの割り込みもペリフェラルによって使用されていなければ、これらの予約された割り込みは、PIEIFRx レベルでイネーブルにされている場合にソフトウェア割り込みとして使用できます。その他の場合は、ペリフェラルから来る割り込みは、PIEIFR の変更時にフラグを誤ってクリアすることによって失われることがあります。

要約すると、予約された割り込みをソフトウェア割り込みとして使用できる安全なケースは2つあります。

1. グループ内のペリフェラルが割り込みをアサートしていない。
2. ペリフェラル割り込みがグループに割り当てられていない。たとえば、PIE グループ 11 および 12 には、どのペリフェラルも接続されていない。

PIE モジュールに接続されたペリフェラルおよび外部割り込みの割り込みグループリングは、表 6-4 に示されています。表の各行は、特定の CPU 割り込みにマルチプレクスされた8つの割り込みを示します。PIE ベクタ・テーブル全体は、マルチプレクスされた割り込みとマルチプレクスされていない割り込みの両方を含め、表 6-5 に示されています。

表 6-4 280x PIE マルチプレクス・ペリフェラル割り込みベクタ・テーブル

	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1
INT1.y	WAKEINT (LPM/WD) 0xD4E	TINT0 (TIMER 0) 0xD4C	ADCINT (ADC) 0xD4A	XINT2 0xD48	XINT1 0xD46	予約 0xD44	SEQ2INT (ADC) 0xD42	SEQ1INT (ADC) 0xD40
INT2.y	予約	予約	EPWM6_TZINT (ePWM6) 0xD5A	EPWM5_TZINT (ePWM5) 0xD58	EPWM4_TZINT (ePWM4) 0xD56	EPWM3_TZINT (ePWM3) 0xD54	EPWM2_TZINT (ePWM2) 0xD52	EPWM1_TZINT (ePWM1) 0xD50
INT3.y	予約	予約	EPWM6_INT (ePWM6) 0xD6A	EPWM5_INT (ePWM5) 0xD68	EPWM4_INT (ePWM4) 0xD66	EPWM3_INT (ePWM3) 0xD64	EPWM2_INT (ePWM2) 0xD62	EPWM1_INT (ePWM1) 0xD60
INT4.y	予約	予約	予約	予約	ECAP4_INT (eCAP4) 0xD76	ECAP3_INT (eCAP3) 0xD74	ECAP2_INT (eCAP2) 0xD72	ECAP1_INT (eCAP1) 0xD70
INT5.y	予約	予約	予約	予約	予約	予約	EQEP2_INT (eQEP2) 0xD82	EQEP1_INT (eQEP1) 0xD80
INT6.y	SPITXINTD (SPI-D) 0xD9E	SPITXINTD (SPI-D) 0xD9C	SPITXINTC (SPI-C) 0xD9A	SPITXINTC (SPI-C) 0xD98	SPITXINTB (SPI-B) 0xD96	SPITXINTB (SPI-B) 0xD94	SPITXINTA (SPI-A) 0xD92	SPITXINTA (SPI-A) 0xD90
INT7.y	予約	予約	予約	予約	予約	予約	予約	予約
INT8.y	予約	予約	予約	予約	予約	予約	I2CINT1A (I2C-A) 0xDB0	I2CINT2A (I2C-A) 0xDB0
INT9.y	ECAN1INTB (CAN-B) 0xDCE	ECAN0INTB (CAN-B) 0xDCC	ECAN1INTA (CAN-A) 0xDCA	ECAN0INTA (CAN-A) 0xDC8	SCITXINTB (SCI-B) 0xDC6	SCIRXINTB (SCI-B) 0xDC4	SCITXINTA (SCI-A) 0xDC2	SCIRXINTA (SCI-A) 0xDC0
INT10.y	予約	予約	予約	予約	予約	予約	予約	予約
INT11.y	予約	予約	予約	予約	予約	予約	予約	予約
INT12.y	予約	予約	予約	予約	予約	予約	予約	予約

表 6-5 280x PIE ベクタ・テーブル

名前	ベクトル ID ⁽¹⁾	アドレス ⁽²⁾	サイズ (x16)	説明 ⁽³⁾	CPU 優先順位	PIE グループ優先順位
Reset	0	0x0000 0D00	2	リセットは、常にブートROM内のロケーション 0x003F FFC0 からフェッチされます。	1 (最上位)	-
INT1	1	0x0000 0D02	2	未使用。PIE グループ 1 を参照	5	-
INT2	2	0x0000 0D04	2	未使用。PIE グループ 2 を参照	6	-
INT3	3	0x0000 0D06	2	未使用。PIE グループ 3 を参照	7	-
INT4	4	0x0000 0D08	2	未使用。PIE グループ 4 を参照	8	-
INT5	5	0x0000 0D0A	2	未使用。PIE グループ 5 を参照	9	-
INT6	6	0x0000 0D0C	2	未使用。PIE グループ 6 を参照	10	-
INT7	7	0x0000 0D0E	2	未使用。PIE グループ 7 を参照	11	-
INT8	8	0x0000 0D10	2	未使用。PIE グループ 8 を参照	12	-
INT9	9	0x0000 0D12	2	未使用。PIE グループ 9 を参照	13	-
INT10	10	0x0000 0D14	2	未使用。PIE グループ 10 を参照	14	-
INT11	11	0x0000 0D16	2	未使用。PIE グループ 11 を参照	15	-
INT12	12	0x0000 0D18	2	未使用。PIE グループ 12 を参照	16	-
INT13	13	0x0000 0D1A	2	外部割り込み 13 (XINT13) または CPU タイマ 1 (TI/RTOS 用) ⁽⁴⁾	17	-
INT14	14	0x0000 0D1C	2	CPU タイマ 2 (TI/RTOS 用)	18	-
DATALOG	15	0x0000 0D1E	2	CPU データ・ロギング割り込み	19 (最下位)	-
RTOSINT	16	0x0000 0D20	2	CPU リアルタイム OS 割り込み	4	-
EMUINT	17	0x0000 0D22	2	CPU エミュレーション割り込み	2	-
NMI	18	0x0000 0D24	2	外部ノンマスカブル割り込み	3	-
ILLEGAL	19	0x0000 0D26	2	不正オペレーション	-	-
USER1	20	0x0000 0D28	2	ユーザー定義トラップ	-	-
USER1	21	0x0000 0D2A	2	ユーザー定義トラップ	-	-
USER1	22	0x0000 0D2C	2	ユーザー定義トラップ	-	-
USER1	23	0x0000 0D2E	2	ユーザー定義トラップ	-	-
USER1	24	0x0000 0D30	2	ユーザー定義トラップ	-	-
USER1	25	0x0000 0D32	2	ユーザー定義トラップ	-	-
USER1	26	0x0000 0D34	2	ユーザー定義トラップ	-	-
USER1	27	0x0000 0D36	2	ユーザー定義トラップ	-	-
USER1	28	0x0000 0D38	2	ユーザー定義トラップ	-	-

(1) ベクトル ID は、DSP/BIOS によって使用されます。

(2) リセットは、常にブートROM内のロケーション 0x003F FFC0 からフェッチされます。

(3) PIE ベクタ・テーブル内のすべてのロケーションは EALLOW 保護されています。

(4) CPU タイマ 1 は、TI ソフトウェア用に予約されています。ただし、割り込み XINT13 は、お客様のアプリケーションによって自由に使用できます。

表 6-5 280x PIE ベクタ・テーブル (続き)

名前	ベクトル ID ⁽¹⁾	アドレス ⁽²⁾	サイズ (x16)	説明 ⁽³⁾	CPU 優先順位	PIE グループ優先順位
USER10	29	0x0000 0D3A	2	ユーザー定義トラップ	-	-
USER11	30	0x0000 0D3C	2	ユーザー定義トラップ	-	-
USER12	31	0x0000 0D3E	2	ユーザー定義トラップ	-	-
PIE グループ 1 ベクトル - CPU INT1 にマルチプレクスされる						
INT1.1	32	0x0000 0D40	2	SEQ1INT	5	1 (最上位)
INT1.2	33	0x0000 0D42	2	SEQ2INT	(ADC)	5
INT1.3	34	0x0000 0D44	2	予約	(ADC)	5
INT1.4	35	0x0000 0D46	2	XINT1		5
INT1.5	36	0x0000 0D48	2	XINT2		5
INT1.6	37	0x0000 0D4A	2	ADCINT	(ADC)	5
INT1.7	38	0x0000 0D4C	2	TINT0	(CPU タイマ 0)	5
INT1.8	39	0x0000 0D4E	2	WAKEINT	(LPM/WD)	5
PIE グループ 2 ベクトル - CPU INT2 にマルチプレクスされる						
INT2.1	40	0x0000 0D50	2	EPWM1_TZINT	(EPWM1)	1 (最上位)
INT2.2	41	0x0000 0D52	2	EPWM2_TZINT	(EPWM2)	2
INT2.3	42	0x0000 0D54	2	EPWM3_TZINT	(EPWM3)	3
INT2.4	43	0x0000 0D56	2	EPWM4_TZINT	(EPWM4)	4
INT2.5	44	0x0000 0D58	2	EPWM5_TZINT	(EPWM5)	5
INT2.6	45	0x0000 0D5A	2	EPWM6_TZINT	(EPWM6)	6
INT2.7	46	0x0000 0D5C	2	予約		7
INT2.8	47	0x0000 0D5E	2	予約		8 (最下位)
PIE グループ 3 ベクトル - CPU INT3 にマルチプレクスされる						
INT3.1	48	0x0000 0D60	2	EPWM1_INT	(EPWM1)	1 (最上位)
INT3.2	49	0x0000 0D62	2	EPWM2_INT	(EPWM2)	2
INT3.3	50	0x0000 0D64	2	EPWM3_INT	(EPWM3)	3
INT3.4	51	0x0000 0D66	2	EPWM4_INT	(EPWM4)	4
INT3.5	52	0x0000 0D68	2	EPWM5_INT	(EPWM5)	5
INT3.6	53	0x0000 0D6A	2	EPWM6_INT	(EPWM6)	6
INT3.7	54	0x0000 0D6C	2	予約		7
INT3.8	55	0x0000 0D6E	2	予約		8 (最下位)
PIE グループ 4 ベクトル - CPU INT4 にマルチプレクスされる						
INT4.1	56	0x0000 0D70	2	ECAP1_INT	(ECAP1)	1 (最上位)
INT4.2	57	0x0000 0D72	2	ECAP2_INT	(ECAP2)	2

表 6-5 280x PIE ベクタ・テーブル (続き)

名前	ベクトル ID ⁽¹⁾	アドレス ⁽²⁾	サイズ (x16)	説明 ⁽³⁾	CPU 優先順位	PIE グループ優先順位
INT4.3	58	0x0000 0D74	2	ECAP3_INT	8	3
INT4.4	59	0x0000 0D76	2	ECAP4_INT	8	4
INT4.5	60	0x0000 0D78	2	予約	8	5
INT4.6	61	0x0000 0D7A	2	予約	8	6
INT4.7	62	0x0000 0D7C	2	予約	8	7
INT4.8	63	0x0000 0D7E	2	予約	8	8 (最下位)
PIE グループ 5 ベクトル - CPU INT5 にマルチプレクスされる						
INT5.1	64	0x0000 0D80	2	EQEP1_INT	9	1 (最上位)
INT5.2	65	0x0000 0D82	2	EQEP1_INT	9	2
INT5.3	66	0x0000 0D84	2	予約	9	3
INT5.4	67	0x0000 0D86	2	予約	9	4
INT5.5	68	0x0000 0D88	2	予約	9	5
INT5.6	69	0x0000 0D8A	2	予約	9	6
INT5.7	70	0x0000 0D8C	2	予約	9	7
INT5.8	71	0x0000 0D8E	2	予約	9	8 (最下位)
PIE グループ 6 ベクトル - CPU INT6 にマルチプレクスされる						
INT6.1	72	0x0000 0D90	2	SPIRXINTA	10	1 (最上位)
INT6.2	73	0x0000 0D92	2	SPITXINTA	10	2
INT6.3	74	0x0000 0D94	2	SPIRXINTB	10	3
INT6.4	75	0x0000 0D96	2	SPITXINTB	10	4
INT6.5	76	0x0000 0D98	2	SPIRXINTC	10	5
INT6.6	77	0x0000 0D9A	2	SPITXINTC	10	6
INT6.7	78	0x0000 0D9C	2	SPIRXINTD	10	7
INT6.8	79	0x0000 0D9E	2	SPITXINTD	10	8 (最下位)
PIE グループ 7 ベクトル - CPU INT7 にマルチプレクスされる						
INT7.1	80	0x0000 0DA0	2	予約	11	1 (最上位)
INT7.2	81	0x0000 0DA2	2	予約	11	2
INT7.3	82	0x0000 0DA4	2	予約	11	3
INT7.4	83	0x0000 0DA6	2	予約	11	4
INT7.5	84	0x0000 0DA8	2	予約	11	5
INT7.6	85	0x0000 0DAA	2	予約	11	6
INT7.7	86	0x0000 0DAC	2	予約	11	7
INT7.8	87	0x0000 0DAE	2	予約	11	8 (最下位)
PIE グループ 8 ベクトル - CPU INT8 にマルチプレクスされる						

表 6-5 280x PIE ベクタ・テーブル (続き)

名前	ベクトル ID ⁽¹⁾	アドレス ⁽²⁾	サイズ (x16)	説明 ⁽³⁾	CPU 優先順位	PIE グループ優先順位
INT8.1	88	0x0000 0DB0	2	I2CINT1A	12	1 (最上位)
INT8.2	89	0x0000 0DB2	2	I2CINT2A	12	2
INT8.3	90	0x0000 0DB4	2	予約	12	3
INT8.4	91	0x0000 0DB6	2	予約	12	4
INT8.5	92	0x0000 0DB8	2	予約	12	5
INT8.6	93	0x0000 0DBA	2	予約	12	6
INT8.7	94	0x0000 0DBC	2	予約	12	7
INT8.8	95	0x0000 0DBE	2	予約	12	8 (最下位)
PIE グループ 9 ベクトル - CPU INT9 にマルチプレクスされる						
INT9.1	96	0x0000 0DC0	2	SCIRXINTA	13	1 (最上位)
INT9.2	97	0x0000 0DC2	2	SCITXINTA	13	2
INT9.3	98	0x0000 0DC4	2	SCIRXINTB	13	3
INT9.4	99	0x0000 0DC6	2	SCITXINTB	13	4
INT9.5	100	0x0000 0DC8	2	ECAN0INTA	13	5
INT9.6	101	0x0000 0DCA	2	ECAN1INTA	13	6
INT9.7	102	0x0000 0DCC	2	ECAN0INTA	13	7
INT9.8	103	0x0000 0DCE	2	ECAN1INTA	13	8 (最下位)
PIE グループ 10 ベクトル - CPU INT10 にマルチプレクスされる						
INT10.1	104	0x0000 0DD0	2	予約	14	1 (最上位)
INT10.2	105	0x0000 0DD2	2	予約	14	2
INT10.3	106	0x0000 0DD4	2	予約	14	3
INT10.4	107	0x0000 0DD6	2	予約	14	4
INT10.5	108	0x0000 0DD8	2	予約	14	5
INT10.6	109	0x0000 0DDA	2	予約	14	6
INT10.7	110	0x0000 0DDC	2	予約	14	7
INT10.8	111	0x0000 0DDE	2	予約	14	8 (最下位)
PIE グループ 11 ベクトル - CPU INT11 にマルチプレクスされる						
INT11.1	112	0x0000 0DE0	2	予約	15	1 (最上位)
INT11.2	113	0x0000 0DE2	2	予約	15	2
INT11.3	114	0x0000 0DE4	2	予約	15	3
INT11.4	115	0x0000 0DE6	2	予約	15	4
INT11.5	116	0x0000 0DE8	2	予約	15	5
INT11.6	117	0x0000 0DEA	2	予約	15	6
INT11.7	118	0x0000 0DEC	2	予約	15	7
INT11.8	119	0x0000 0DEE	2	予約	15	8 (最下位)

表 6-5 280x PIE ベクタ・テーブル (続き)

名前	ベクトル ID ⁽¹⁾	アドレス ⁽²⁾	サイズ (x16)	説明 ⁽³⁾	CPU 優先順位	PIE グループ優先順位
PIE グループ 12 ベクトル - CPU INT12 にマルチプレクスされる						
INT12.1	120	0x0000 0DF0	2	予約	16	1 (最上位)
INT12.2	121	0x0000 0DF2	2	予約	16	2
INT12.3	122	0x0000 0DF4	2	予約	16	3
INT12.4	123	0x0000 0DF6	2	予約	16	4
INT12.5	124	0x0000 0DF8	2	予約	16	5
INT12.6	125	0x0000 0DFA	2	予約	16	6
INT12.7	126	0x0000 0DFC	2	予約	16	7
INT12.8	127	0x0000 0DFE	2	予約	16	8 (最下位)

6.4 PIE コンフィギュレーション・レジスタ

PIE ブロックの機能を制御するレジスタは、表 6-6 に示されています。

表 6-6 PIE コンフィギュレーションおよび制御レジスタ

名前	アドレス	サイズ (x16)	説明
PIECTRL	0x0000-0CE0	1	PIE、制御レジスタ
PIEACK	0x0000-0CE1	1	PIE、ACK レジスタ
PIEIER1	0x0000-0CE2	1	PIE、INT1 グループ・イネーブル・レジスタ
PIEIFR1	0x0000-0CE3	1	PIE、INT1 グループ・フラグ・レジスタ
PIEIER2	0x0000-0CE4	1	PIE、INT2 グループ・イネーブル・レジスタ
PIEIFR2	0x0000-0CE5	1	PIE、INT2 グループ・フラグ・レジスタ
PIEIER3	0x0000-0CE6	1	PIE、INT3 グループ・イネーブル・レジスタ
PIEIFR3	0x0000-0CE7	1	PIE、INT3 グループ・フラグ・レジスタ
PIEIER4	0x0000-0CE8	1	PIE、INT4 グループ・イネーブル・レジスタ
PIEIFR4	0x0000-0CE9	1	PIE、INT4 グループ・フラグ・レジスタ
PIEIER5	0x0000-0CEA	1	PIE、INT5 グループ・イネーブル・レジスタ
PIEIFR5	0x0000-0CEB	1	PIE、INT5 グループ・フラグ・レジスタ
PIEIER6	0x0000-0CEC	1	PIE、INT6 グループ・イネーブル・レジスタ
PIEIFR6	0x0000-0CED	1	PIE、INT6 グループ・フラグ・レジスタ
PIEIER7	0x0000-0CEE	1	PIE、INT7 グループ・イネーブル・レジスタ
PIEIFR7	0x0000-0CEF	1	PIE、INT7 グループ・フラグ・レジスタ
PIEIER8	0x0000-0CF0	1	PIE、INT8 グループ・イネーブル・レジスタ
PIEIFR8	0x0000-0CF1	1	PIE、INT8 グループ・フラグ・レジスタ
PIEIER9	0x0000-0CF2	1	PIE、INT9 グループ・イネーブル・レジスタ
PIEIFR9	0x0000-0CF3	1	PIE、INT9 グループ・フラグ・レジスタ
PIEIER10	0x0000-0CF4	1	PIE、INT10 グループ・イネーブル・レジスタ
PIEIFR10	0x0000-0CF5	1	PIE、INT10 グループ・フラグ・レジスタ
PIEIER11	0x0000-0CF6	1	PIE、INT11 グループ・イネーブル・レジスタ
PIEIFR11	0x0000-0CF7	1	PIE、INT11 グループ・フラグ・レジスタ
PIEIER12	0x0000-0CF8	1	PIE、INT12 グループ・イネーブル・レジスタ
PIEIFR12	0x0000-0CF9	1	PIE、INT12 グループ・フラグ・レジスタ

6.5 PIE 割り込みレジスタ

図 6-6 PIECTRL レジスタ (アドレス CE0)

15	PIEVECT	1	0
R-0		ENPIE R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 6-7 PIECTRL レジスタ・アドレス・フィールドの説明

ビット	フィールド	値	説明
15-1	PIEVECT		これらのビットは、ベクタがフェッチされた PIE ベクタ・テーブル内のアドレスを示します。アドレスの最下位ビットは無視され、アドレスのビット 1 ~ 15 だけが示されます。ベクタ値をリードすると、どの割り込みがベクタ・フェッチを生成したかを判断できます。 例：PIECTRL=0x0D27 の場合は、アドレス 0x0D26 (不正操作) からのベクタがフェッチされました。
0	ENPIE	0 1	PIE ベクタ・テーブルからのベクタ・フェッチをイネーブルにします。 注：リセット・ベクタは、それがイネーブルになっている場合でも、PIE から決してフェッチされません。このベクタは、常にブート ROM からフェッチされます。 このビットが 0 に設定された場合は、PIE ブロックはディスエーブルにされ、ベクタはブート ROM 内の CPU ベクタ・テーブルからフェッチされます。PIE ブロックがディスエーブルになっている場合であっても、すべての PIE ブロック・レジスタ (PIEACK、PIEIFR、PIEIER) にアクセスできます。 ENPIE が 1 に設定されると、すべてのベクタ (リセットを除く) が PIE ベクタ・テーブルからフェッチされます。リセット・ベクタは、常にブート ROM からフェッチされます。

図 6-7 PIE 割り込み ACK レジスタ (PIEACK) (アドレス CE1)

15	12	11	0
予約	PIEACK		
R-0	R/W1C-1		

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 6-8 PIE 割り込み ACK レジスタ (PIEACK) のフィールドの説明

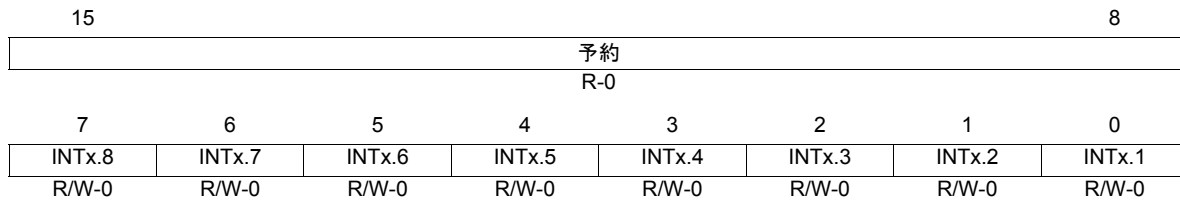
ビット	フィールド	値	説明
15-12	予約		予約
11-0	PIEACK	ビット x=0 ⁽¹⁾ ビット x=1	PIEACK 内の各ビットは、特定の PIE グループを意味します。ビット 0 は、ビット 11 まで INT1 にマルチプレクスされる中の PIE グループ 2 内の割り込みを意味します。ビット 11 は、CPU INT12 にマルチプレクスされる中の PIE グループ 12 を意味します。 ビットが 0 としてリードされた場合、それは、PIE が割り込みをそれぞれのグループから CPU に送信することができることを示します。 0 のライトは無視されます。 1 のリードは、それぞれのグループからの割り込みが CPU へ送られたかどうか、およびそのグループからの他のすべての割り込みが現在ブロックされているかどうかを示します。 割り込みがそのグループに対して保留中である場合は、それぞれの割り込みビットに 1 をライトすると、ビットはクリアされ、PIE ブロックは CPU 割り込み入力へパルスをドライブできるようになります。

(1) ビット x=PIEACK ビット 0 ~ PIEACK ビット 11。ビット 0 は、ビット 11 まで CPU INT1 を意味します。ビット 11 は CPU INT12 を意味します。

6.5.1 PIE 割り込みフラグ・レジスタ

12 の PIEIFR レジスタ、つまり、PIE モジュールによって使用される CPU 割り込み (INT1 ~ INT12) ごとに 1 つの PIEIFR レジスタが存在します。

図 6-8 PIEIFRx レジスタ (x=1 ~ 12)



凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-9 PIEIFRx レジスタのフィールドの説明

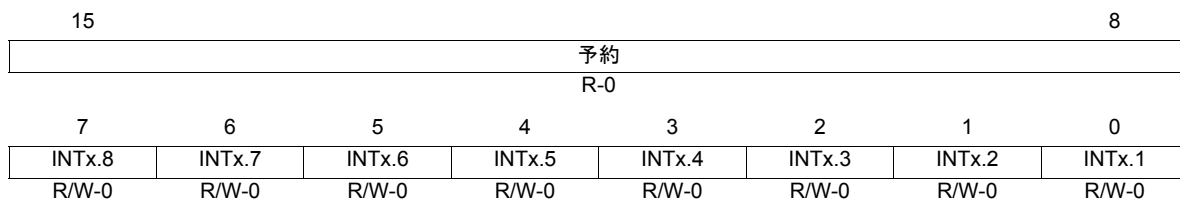
ビット	フィールド	説明
15-8	予約	予約
7	INTx.8	これらのレジスタ・ビットは、割り込みが現在アクティブかどうかを示します。それらは、CPU 割り込みフラグ・レジスタと非常によく似た動作をします。割り込みがアクティブのときには、それぞれのレジスタ・ビットがセットされます。ビットは、割り込みが処理された場合か、またはレジスタ・ビットへの 0 のライトによって、クリアされます。また、このレジスタをリードして、どの割り込みがアクティブまたは保留中かを判断することもできます。x=1 ~ 12。INTx は、CPU INT1 ~ INT12 を意味します。
6	INTx.7	
5	INTx.6	
4	INTx.5	
3	INTx.4	
2	INTx.3	
1	INTx.2	
0	INTx.1	
		PIEIFR レジスタ・ビットは、割り込み処理の割り込みベクタ・フェッチの間にクリアされます。ハードウェアは、PIEIFR レジスタへの CPU アクセスよりも優先されます。

注 : PIEIFR ビットは決してクリアしないでください。割り込みは、read-modify-write 動作の間に失われることがあります。フラグを立てられた割り込みをクリアする方法については、[セクション 6.3.1](#) を参照してください。

6.5.2 PIE 割り込みイネーブル・レジスタ

12 の PIEIER レジスタ、つまり、PIE モジュールによって使用される CPU 割り込み (INT1 ~ INT12) ごとに 1 つの PIEIER レジスタが存在します。

図 6-9 PIEIERx レジスタ (x=1 ~ 12)



凡例 : R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-10 PIEIERx レジスタ (x=1 ~ 12) のフィールドの説明

ビット	フィールド	説明
15-8	予約	予約
7	INTx.8	これらのレジスタ・ビットは、それぞれグループ内の割り込みをイネーブルにし、コア割り込みイネーブル・レジスタと非常によく似た動作をします。ビットを 1 に設定すると、それぞれの割り込みの処理がイネーブルになります。ビットを 0 に設定すると、割り込みの処理がディスエーブルになります。x=1 ~ 12。INTx は、CPU INT1 ~ INT12 を意味します。
6	INTx.7	
5	INTx.6	
4	INTx.5	
3	INTx.4	
2	INTx.3	
1	INTx.2	
0	INTx.1	

注： 通常動作中に PIEIER ビットをクリアする場合は注意を払う必要があります。これらのビットを処理するための正しい手順については、[セクション 6.3.2](#) を参照してください。

6.5.3 CPU 割り込みフラグ・レジスタ (IFR)

CPU 割り込みフラグ・レジスタ (IFR) は 16 ビットの CPU レジスタであり、保留中の割り込みを識別してクリアするために使用されます。IFR は、CPU レベルでのすべてのマスカブル割り込み (INT1 ~ INT14、DLOGINT、および RTOSINT) に対するフラグ・ビットを含んでいます。PIE がイネーブルにされると、PIE モジュールは INT1 ~ INT12 の割り込みソースをマルチプレクスします。

マスカブル割り込みが要求されると、対応するペリフェラル制御レジスタ内のフラグ・ビットが 1 に設定されます。対応するマスク・ビットも 1 である場合は、割り込みリクエストは CPU へ送られ、IFR 内の対応するフラグがセットされます。これは、割り込みが保留中であること、または確認を待っていることを示します。

保留中の割り込みを識別するには、PUSH IFR 命令を使用してから、スタック上の値をテストします。OR IFR 命令を使用して IFR ビットをセットし、AND IFR 命令を使用して保留中の割り込みを手動でクリアしてください。

保留中のすべての割り込みは、AND IFR #0 命令で、またはハードウェア・リセットによってクリアされます。

また、以下のイベントも IFR フラグをクリアします。

- CPU が割り込みを確認する
- 28x デバイスがリセットされる

注：

1. CPU IFR ビットをクリアするには、1 ではなくゼロをそのビットにライトする必要があります。
2. マスカブル割り込みが確認されると、IFR ビットだけが自動的にクリアされます。対応するペリフェラル制御レジスタ内のフラグ・ビットはクリアされません。制御レジスタ・フラグがクリアされることをアプリケーション上で必要とする場合は、そのビットをソフトウェアによってクリアする必要があります。
3. 割り込みが INTR 命令によって要求され、対応する IFR ビットがセットされている場合は、CPU はそのビットを自動的にクリアしません。IFR ビットがクリアされることをアプリケーションが必要とする場合は、そのビットをソフトウェアによってクリアする必要があります。
4. IMR および IFR レジスタは、コアレベル割り込みに関係します。すべてのペリフェラルは、固有の割り込みマスクおよびフラグ・ビットをそれぞれの制御 / コンフィギュレーション・レジスタに持っています。1 つのコアレベル割り込みのもとで複数のペリフェラル割り込みがグループ化されることに注意してください。

図 6-10 割り込みフラグ・レジスタ (IFR) - CPU レジスタ

15	14	13	12	11	10	9	8
RTOSINT	DLOGINT	INT14	INT13	INT12	INT11	INT10	INT9
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-11 割り込みフラグ・レジスタ (IFR) - CPU レジスタのフィールドの説明

ビット	フィールド	値	説明
15	RTOSINT	0 1	リアルタイム・オペレーティング・システム・フラグ。RTOSINT は、RTOS 割り込みのフラグです。 どの RTOS 割り込みも保留中ではありません。 少なくとも 1 つの RTOS 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
14	DLOGINT	0 1	DLOGINT データ・ロギング割り込みフラグ。DLOGINT は、データ・ロギング割り込みのフラグです。 どの DLOGINT も保留中ではありません。 少なくとも 1 つの DLOGINT 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
13	INT14	0 1	割り込み 14 フラグ。INT14 は、CPU 割り込みレベル INT14 に接続されている割り込みのフラグです。 どの INT14 割り込みも保留中ではありません。 少なくとも 1 つの INT14 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
12	INT13	0 1	割り込み 13 フラグ。INT13 は、CPU 割り込みレベル INT13 に接続されている割り込みのフラグです。 どの INT13 割り込みも保留中ではありません。 少なくとも 1 つの INT13 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
11	INT12	0 1	割り込み 12 フラグ。INT12 は、CPU 割り込みレベル INT12 に接続されている割り込みのフラグです。 どの INT12 割り込みも保留中ではありません。 少なくとも 1 つの INT12 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
10	INT11	0 1	割り込み 11 フラグ。INT11 は、CPU 割り込みレベル INT11 に接続されている割り込みのフラグです。 どの INT11 割り込みも保留中ではありません。 少なくとも 1 つの INT11 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
9	INT10	0 1	割り込み 10 フラグ。INT10 は、CPU 割り込みレベル INT10 に接続されている割り込みのフラグです。 どの INT10 割り込みも保留中ではありません。 少なくとも 1 つの INT6 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
8	INT9	0 1	割り込み 9 フラグ。INT9 は、CPU 割り込みレベル INT6 に接続されている割り込みのフラグです。 どの INT9 割り込みも保留中ではありません。 少なくとも 1 つの INT9 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
7	INT8	0 1	割り込み 8 フラグ。INT8 は、CPU 割り込みレベル INT6 に接続されている割り込みのフラグです。 どの INT8 割り込みも保留中ではありません。 少なくとも 1 つの INT8 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
6	INT7	0 1	割り込み 7 フラグ。INT7 は、CPU 割り込みレベル INT7 に接続されている割り込みのフラグです。 どの INT7 割り込みも保留中ではありません。 少なくとも 1 つの INT7 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。

表 6-11 割り込みフラグ・レジスタ (IFR) - CPU レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明
5	INT6	0 1	割り込み 6 フラグ。INT6 は、CPU 割り込みレベル INT6 に接続されている割り込みのフラグです。 どの INT6 割り込みも保留中ではありません。 少なくとも 1 つの INT6 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
4	INT5	0 1	割り込み 5 フラグ。INT5 は、CPU 割り込みレベル INT5 に接続されている割り込みのフラグです。 どの INT5 割り込みも保留中ではありません。 少なくとも 1 つの INT5 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
3	INT4	0 1	割り込み 4 フラグ。INT4 は、CPU 割り込みレベル INT4 に接続されている割り込みのフラグです。 どの INT4 割り込みも保留中ではありません。 少なくとも 1 つの INT4 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
2	INT3	0 1	割り込み 3 フラグ。INT3 は、CPU 割り込みレベル INT3 に接続されている割り込みのフラグです。 どの INT3 割り込みも保留中ではありません。 少なくとも 1 つの INT3 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
1	INT2	0 1	割り込み 2 フラグ。INT2 は、CPU 割り込みレベル INT2 に接続されている割り込みのフラグです。 どの INT2 割り込みも保留中ではありません。 少なくとも 1 つの INT2 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。
0	INT1	0 1	割り込み 1 フラグ。INT1 は、CPU 割り込みレベル INT1 に接続されている割り込みのフラグです。 どの INT1 割り込みも保留中ではありません。 少なくとも 1 つの INT1 割り込みが保留中です。0 をこのビットにライトして、それを 0 にクリアし、割り込みリクエストをクリアします。

6.5.4 割り込みイネーブル・レジスタ (IER) およびデバッグ割り込みイネーブル・レジスタ (DBGIER)

IER は 16 ビットの CPU レジスタです。IER は、すべてのマスク可能 CPU 割り込みレベル (INT1 ~ INT14、RTOSINT、および DLOGINT) に対するイネーブル・ビットを含んでいます。IER には NMI も XRS も含まれていません。したがって、IER は、これらの割り込みには影響しません。

ユーザーは、IER をリードして、イネーブルまたはディセーブルにされた割り込みレベルを識別することができます。また、IER にライトして、割り込みレベルをイネーブルまたはディセーブルにすることもできます。割り込みレベルをイネーブルにするには、OR IER 命令を使用して、対応する IER ビットを 1 に設定します。割り込みレベルをディセーブルにするには、AND IER 命令を使用して、対応する IER ビットをゼロに設定します。割り込みがディセーブルにされると、INTM ビットの値にかかわらず、その割り込みは確認されません。割り込みがイネーブルにされると、その割り込みは、対応する IFR ビットが 1 で INTM ビットがゼロの場合に確認されます。

OR IER および AND IER 命令を使用して IER ビットを変更する場合は、リアルタイム・オペレーティング・システムが存在しない限り、それらの命令がビット 15 (RTOSINT) の状態を変更しないことを確認してください。

ハードウェア割り込みが処理されるか、INTR 命令が実行されると、対応する IER ビットが自動的にクリアされます。割り込みが TRAP 命令によって要求される場合には、IER ビットは自動的にクリアされません。TRAP 命令のケースで、ビットをクリアする必要がある場合は、そのクリアを割り込みサービス・ルーチンによって行う必要があります。

リセット時には、すべての IER ビットが 0 にクリアされて、すべてのマスカブル CPU レベル割り込みがディスエーブルになります。

IER レジスタは図 6-11 に示されており、ビットの説明は図の後に続きます。

図 6-11 割り込みイネーブル・レジスタ (IER) - CPU レジスタ

15	14	13	12	11	10	9	8
RTOSINT	DLOGINT	INT14	INT13	INT12	INT11	INT10	INT9
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 6-12 割り込みイネーブル・レジスタ (IER) - CPU レジスタのフィールドの説明

ビット	フィールド	値	説明
15	RTOSINT	0	リアルタイム・オペレーティング・システム割り込みイネーブル。RTOSINT は、CPU RTOS 割り込みをイネーブルまたはディスエーブルにします。
		1	レベル INT6 はディスエーブルにします。
14	DLOGINT	0	データ・ロギング割り込みイネーブル。DLOGINT は、CPU データ・ロギング割り込みをイネーブルまたはディスエーブルにします。
		1	レベル INT6 はイネーブルにします。
13	INT14	0	割り込み 14 イネーブル。INT14 は、CPU 割り込みレベル INT14 をイネーブルまたはディスエーブルにします。
		1	レベル INT14 はディスエーブルにします。
12	INT13	0	割り込み 13 イネーブル。INT13 は、CPU 割り込みレベル INT13 をイネーブルまたはディスエーブルにします。
		1	レベル INT13 はイネーブルにします。
11	INT12	0	割り込み 12 イネーブル。INT12 は、CPU 割り込みレベル INT12 をイネーブルまたはディスエーブルにします。
		1	レベル INT12 はディスエーブルにします。
10	INT11	0	割り込み 11 イネーブル。INT11 は、CPU 割り込みレベル INT11 をイネーブルまたはディスエーブルにします。
		1	レベル INT11 はイネーブルにします。
9	INT10	0	割り込み 10 イネーブル。INT10 は、CPU 割り込みレベル INT10 をイネーブルまたはディスエーブルにします。
		1	レベル INT10 はディスエーブルにします。
8	INT9	0	割り込み 9 イネーブル。INT9 は、CPU 割り込みレベル INT9 をイネーブルまたはディスエーブルにします。
		1	レベル INT9 はイネーブルにします。

表 6-12 割り込みイネーブル・レジスタ (IER) - CPU レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明
7	INT8	0 1	割り込み 8 イネーブル。INT8 は、CPU 割り込みレベル INT8 をイネーブルまたはディスエーブルにします。 レベル INT8 はディスエーブルにします。 レベル INT8 はイネーブルにします。
6	INT7	0 1	割り込み 7 イネーブル。INT7 は、CPU 割り込みレベル INT7 をイネーブルまたはディスエーブルにします。 レベル INT7 はディスエーブルにします。 レベル INT7 はイネーブルにします。
5	INT6	0 1	割り込み 6 イネーブル。INT6 は、CPU 割り込みレベル INT6 をイネーブルまたはディスエーブルにします。 レベル INT6 はディスエーブルにします。 レベル INT6 はイネーブルにします。
4	INT5	0 1	割り込み 5 イネーブル。INT5 は、CPU 割り込みレベル INT5 をイネーブルまたはディスエーブルにします。 レベル INT5 はディスエーブルにします。 レベル INT5 はイネーブルにします。
3	INT4	0 1	割り込み 4 イネーブル。INT4 は、CPU 割り込みレベル INT4 をイネーブルまたはディスエーブルにします。 レベル INT4 はディスエーブルにします。 レベル INT4 はイネーブルにします。
2	INT3	0 1	割り込み 3 イネーブル。INT3 は、CPU 割り込みレベル INT3 をイネーブルまたはディスエーブルにします。 レベル INT3 はディスエーブルにします。 レベル INT3 はイネーブルにします。
1	INT2	0 1	割り込み 2 イネーブル。INT2 は、CPU 割り込みレベル INT2 をイネーブルまたはディスエーブルにします。 レベル INT2 はディスエーブルにします。 レベル INT2 はイネーブルにします。
0	INT1	0 1	割り込み 1 イネーブル。INT1 は、CPU 割り込みレベル INT1 をイネーブルまたはディスエーブルにします。 レベル INT1 はディスエーブルにします。 レベル INT1 はイネーブルにします。

デバッグ割り込みイネーブル・レジスタ (DBGIER) は、CPU がリアルタイム・エミュレーション・モードで停止している場合にのみ使用されます。DBGIER でイネーブルにされた割り込みは、タイムクリティカルな割り込みとして定義されます。CPU がリアルタイム・モードで停止されている場合は、処理される割り込みだけが、IER でもイネーブルにされるタイムクリティカルな割り込みです。CPU がリアルタイム・エミュレーション・モードで動作している場合は、標準の割り込み処理プロセスが使用され、DBGIER は無視されます。

IER の場合と同様に、DBGIER をリードして、イネーブルまたはディスエーブルにされた割り込みを識別し、DBGIER にライトして、割り込みをイネーブルまたはディスエーブルにすることができます。割り込みをイネーブルにするには、対応するビットを 1 に設定します。割り込みをディスエーブルにするには、対応するビットを 0 に設定します。PUSH DBGIER 命令を使用して DBGIER からリードし、POP DBGIER を使用して DBGIER レジスタにライトしてください。リセット時には、すべての DBGIER ビットが 0 に設定されます。

図 6-12 デバッグ割り込みイネーブル・レジスタ (DBGIER) - CPU レジスタ

15	14	13	12	11	10	9	8
RTOSINT	DLOGINT	INT14	INT13	INT12	INT11	INT10	INT9
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-13 デバッグ割り込みイネーブル・レジスタ (DBGIER) - CPU レジスタのフィールドの説明

ビット	フィールド	値	説明
15	RTOSINT	0 1	リアルタイム・オペレーティング・システム割り込みイネーブル。RTOSINT は、CPU RTOS 割り込みをイネーブルまたはディスエーブルにします。 レベル INT6 はディスエーブルにします。 レベル INT6 はイネーブルにします。
14	DLOGINT	0 1	データ・ロギング割り込みイネーブル。DLOGINT は、CPU データ・ロギング割り込みをイネーブルまたはディスエーブルにします。 レベル INT6 はディスエーブルにします。 レベル INT6 はイネーブルにします。
13	INT14	0 1	割り込み 14 イネーブル。INT14 は、CPU 割り込みレベル INT14 をイネーブルまたはディスエーブルにします。 レベル INT14 はディスエーブルにします。 レベル INT14 はイネーブルにします。
12	INT13	0 1	割り込み 13 イネーブル。INT13 は、CPU 割り込みレベル INT13 をイネーブルまたはディスエーブルにします。 レベル INT13 はディスエーブルにします。 レベル INT13 はイネーブルにします。
11	INT12	0 1	割り込み 12 イネーブル。INT12 は、CPU 割り込みレベル INT12 をイネーブルまたはディスエーブルにします。 レベル INT12 はディスエーブルにします。 レベル INT12 はイネーブルにします。
10	INT11	0 1	割り込み 11 イネーブル。INT11 は、CPU 割り込みレベル INT11 をイネーブルまたはディスエーブルにします。 レベル INT11 はディスエーブルにします。 レベル INT11 はイネーブルにします。
9	INT10	0 1	割り込み 10 イネーブル。INT10 は、CPU 割り込みレベル INT10 をイネーブルまたはディスエーブルにします。 レベル INT10 はディスエーブルにします。 レベル INT10 はイネーブルにします。
8	INT9	0 1	割り込み 9 イネーブル。INT9 は、CPU 割り込みレベル INT9 をイネーブルまたはディスエーブルにします。 レベル INT9 はディスエーブルにします。 レベル INT9 はイネーブルにします。
7	INT8	0 1	割り込み 8 イネーブル。INT8 は、CPU 割り込みレベル INT8 をイネーブルまたはディスエーブルにします。 レベル INT8 はディスエーブルにします。 レベル INT8 はイネーブルにします。

表 6-13 デバッグ割り込みイネーブル・レジスタ (DBGIER) - CPU レジスタのフィールドの説明 (続き)

ビット	フィールド	値	説明
6	INT7	0 1	割り込み 7 イネーブル。INT7 は、CPU 割り込みレベル INT7 をイネーブルまたはディスエーブルにします。 レベル INT7 はディスエーブルにします。 レベル INT7 はイネーブルにします。
5	INT6	0 1	割り込み 6 イネーブル。INT6 は、CPU 割り込みレベル INT6 をイネーブルまたはディスエーブルにします。 レベル INT6 はディスエーブルにします。 レベル INT6 はイネーブルにします。
4	INT5	0 1	割り込み 5 イネーブル。INT5 は、CPU 割り込みレベル INT5 をイネーブルまたはディスエーブルにします。 レベル INT5 はディスエーブルにします。 レベル INT5 はイネーブルにします。
3	INT4	0 1	割り込み 4 イネーブル。INT4 は、CPU 割り込みレベル INT4 をイネーブルまたはディスエーブルにします。 レベル INT4 はディスエーブルにします。 レベル INT4 はイネーブルにします。
2	INT3	0 1	割り込み 3 イネーブル。INT3 は、CPU 割り込みレベル INT3 をイネーブルまたはディスエーブルにします。 レベル INT3 はディスエーブルにします。 レベル INT3 はイネーブルにします。
1	INT2	0 1	割り込み 2 イネーブル。INT2 は、CPU 割り込みレベル INT2 をイネーブルまたはディスエーブルにします。 レベル INT2 はディスエーブルにします。 レベル INT2 はイネーブルにします。
0	INT1	0 1	割り込み 1 イネーブル。INT1 は、CPU 割り込みレベル INT1 をイネーブルまたはディスエーブルにします。 レベル INT1 はディスエーブルにします。 レベル INT1 はイネーブルにします。

6.6 外部割り込み制御レジスタ

いくつかのデバイスは、3つのマスクされた外部割り込み XINT1、XINT2、XINT13をサポートします。XINT13は、1つのノンマスクابل割り込み XNMI とマルチプレクスされます。これらの外部割り込みのそれぞれは、負または正のエッジ・トリガで選択でき、イネーブルまたはディスエーブルにすることもできます (XNMI を含む)。マスクされた割り込みは、有効な割り込みエッジの検出時にゼロにリセットされる 16 ビットのフリー・ランニング・アップ・カウンタも含んでいます。このカウンタを使用すると、割り込みを正確にタイムスタンプすることができます。

図 6-13 外部割り込み 1 制御レジスタ (XINT1CR) (アドレス 7070h)

15	4	3	2	1	0
予約	Polarity		予約	Enable	
R-0	R/W-0		R-0	R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 6-14 外部割り込み 1 制御レジスタ (XINT1CR) のフィールドの説明

ビット	フィールド	値	説明
15-4	予約		リードはゼロを返します。ライトは何の影響も及ぼしません。
3-2	Polarity	00	立ち下がリエッジ (High から Low への信号変化) で生成された割り込み
		01	立ち上がりエッジ (Low から High への信号変化) で生成された割り込み
		10	立ち下がリエッジ (High から Low への信号変化) で生成された割り込み
		11	立ち下がリエッジと立ち上がりエッジ (High から Low への信号変化と Low から High への信号変化) の両方で生成された割り込み
1	予約		リードはゼロを返します。ライトは何の影響も及ぼしません。
0	イネーブル	0	割り込みをディスエーブルにする
		1	割り込みをイネーブルにする

図 6-14 外部割り込み 2 制御レジスタ (XINT2CR) (アドレス 7071h)

15	4	3	2	1	0
予約	Polarity		予約	Enable	
R-0	R/W-0		R-0	R/W-0	

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 6-15 外部割り込み 2 制御レジスタ (XINT2CR) のフィールドの説明

ビット	フィールド	値	説明
15-4	予約		リードはゼロを返します。ライトは何の影響も及ぼしません。
3-2	Polarity	00	立ち下がリエッジ (High から Low への信号変化) で生成された割り込み
		01	立ち上がりエッジ (Low から High への信号変化) で生成された割り込み
		10	立ち下がリエッジ (High から Low への信号変化) で生成された割り込み
		11	立ち下がリエッジと立ち上がりエッジ (High から Low への信号変化と Low から High への信号変化) の両方で生成された割り込み
1	予約		リードはゼロを返します。ライトは何の影響も及ぼしません。
0	Enable	0	割り込みをディスエーブルにする
		1	割り込みをイネーブルにする

図 6-15 外部 NMI 割り込み制御レジスタ (XNMICR) - アドレス 7077h

15	4	3	2	1	0
予約		Polarity		Select	Enable
R-0		R/W-0		R-0	R/W-0

凡例：R/W = リード/ライト、R = リードのみ、-n = リセット後の値

表 6-16 外部 NMI 割り込み制御レジスタ (XNMICR) のフィールドの説明

ビット	フィールド	値	説明
15-4	予約		リードはゼロを返します。ライトは何の影響も及ぼしません。
3-2	Polarity	00 01 10 11	このリード/ライト・ビットは、ピン上の信号の立ち上がりエッジと立ち下がりエッジのどちらで割り込みが生成されるのかを決定します。 00 立ち下がりエッジ (High から Low への信号変化) で生成された割り込み 01 立ち上がりエッジ (Low から High への信号変化) で生成された割り込み 10 立ち下がりエッジ (High から Low への信号変化) で生成された割り込み 11 立ち下がりエッジと立ち上がりエッジ (High から Low への信号変化と Low から High への信号変化) の両方で生成された割り込み
1	Select		INT13 のソースを選択します。 INT13 に接続されているタイマ 1 INT13 に接続されている XNMI_XINT13
0	Enable	0 1	このリード/ライト・ビットは、外部割り込み NMI をイネーブルまたはディスエーブルにします。 0 XNMI 割り込みをディスエーブルにする 1 XNMI 割り込みをイネーブルにする

XNMI 制御レジスタ (XNMICR) を使用すると、CPU への NMI 割り込みをイネーブルまたはディスエーブルにすることができます。また、INT13 CPU 割り込みのソースを選択することもできます。図 6-4 に示されているように、INT13 割り込みのソースは、内部 CPU タイマ 1、または XNMI に割り当てられた外部 GPIO 信号のいずれかにすることができます。

F280x デバイスでは、CPU タイマ 1 は TI ソフトウェア用に予約されています。ただし、INT13 割り込みは、お客様の用途向けに引き続き XNMI_XINT13 に接続できます。

表 6-17 は、XNMICR レジスタの設定と 28x CPU への割り込みソースの関係を示します。

表 6-17 XNMICR レジスタ設定および割り込みソース

XNMICR ENABLE	レジスタ・ビット SELECT	28x CPU 割り込み		タイムスタンプ (XNMICTR)
		NMI ソース	INT13 ソース	
0	0	ディスエーブル	CPU タイマ 1	なし
0	1	ディスエーブル	XNMI	なし
1	0	XNMI	CPU タイマ 1	XNMI
1	1	ディスエーブル	XNMI	XNMI

各外部割り込みについては、割り込みエッジが検出されるたびに 0x000 にリセットされる 16 ビット・カウンタも存在します。これらのカウンタを使用すると、割り込みの発生を正確にタイムスタンプすることができます。

図 6-16 外部割り込み 1 カウンタ (XINT1CTR) (アドレス 7078h)

15	0
INTCTR[15-8]	
R-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-18 外部割り込み 1 カウンタ (XINT1CTR) のフィールドの説明

ビット	フィールド	説明
15-0	INTCTR	これは、SYSCLKOUT レートでクロックされるフリー・ランニング 16 ビット・アップ・カウンタです。カウンタ値は、有効な割り込みエッジが検出されると 0x0000 にリセットされ、次の有効なエッジが検出されるまでカウントを続行します。割り込みがディスエーブルにされると、カウンタは停止します。カウンタは、フリー・ランニング・カウンタであり、最大値に達すると折り返してゼロになります。カウンタはリード専用レジスタで、有効な割り込みエッジまたはリセットによってのみゼロにリセットできます。

図 6-17 外部割り込み 2 カウンタ (XINT2CTR) - アドレス 7079h

15	0
INTCTR[15-0]	
R-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-19 外部割り込み 2 カウンタ (XINT2CTR) のフィールドの説明

ビット	フィールド	説明
15-0	INTCTR	これは、SYSCLKOUT レートでクロックされるフリー・ランニング 16 ビット・アップ・カウンタです。カウンタ値は、有効な割り込みエッジが検出されると 0x0000 にリセットされ、次の有効なエッジが検出されるまでカウントを続行します。割り込みがディスエーブルにされると、カウンタは停止します。カウンタは、フリー・ランニング・カウンタであり、最大値に達すると折り返してゼロになります。カウンタはリード専用レジスタで、有効な割り込みエッジまたはリセットによってのみゼロにリセットできます。

図 6-18 外部 NMI 割り込みカウンタ (XNMICTR) (アドレス 707Fh)

15	0
INTCTR[15-0]	
R-0	

凡例：R/W = リード / ライト、R = リードのみ、-n = リセット後の値

表 6-20 外部 NMI 割り込みカウンタ (XNMICTR) のフィールドの説明

ビット	フィールド	説明
15-0	INTCTR	これは、SYSCLKOUT レートでクロックされるフリー・ランニング 16 ビット・アップ・カウンタです。カウンタ値は、有効な割り込みエッジが検出されると 0x0000 にリセットされ、次の有効なエッジが検出されるまでカウントを続行します。割り込みがディスエーブルにされると、カウンタは停止します。カウンタは、フリー・ランニング・カウンタであり、最大値に達すると折り返してゼロになります。カウンタはリード専用レジスタで、有効な割り込みエッジまたはリセットによってのみゼロにリセットできます。

ご注意

日本テキサス・インスツルメンツ株式会社(以下TIJといひます)及びTexas Instruments Incorporated(TIJの親会社、以下TIJおよびTexas Instruments Incorporatedを総称してTIといひます)は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従ひまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIとの間に取引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIの標準契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従ひ販売時の仕様に対応した性能を有していること、またはお客様とTIとの間で合意された保証条件に従ひ合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定される危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾することは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えたり、保証もしくは是認することを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、且つその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、且つ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

Copyright © 2007, Texas Instruments Incorporated
日本語版 日本テキサス・インスツルメンツ株式会社

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位(外装から取り出された内装及び個装)又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で(導電性マットにアースをとったもの等)、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。(但し、結露しないこと。)

- 直射日光があたる状態で保管・輸送しないこと。
3. 防湿梱包
 - 防湿梱包品は、開封後は個別推奨保管環境及び期間に従ひ基板実装すること。
 4. 機械的衝撃
 - 梱包品(外装、内装、個装)及び製品単品を落下させたり、衝撃を与えないこと。
 5. 熱衝撃
 - はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。(個別推奨条件がある時はそれに従うこと。)
 6. 汚染
 - はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質(硫黄、塩素等ハロゲン)のある環境で保管・輸送しないこと。
 - はんだ付け後は十分にフラックスの洗浄を行うこと。(不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。)

以上

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DSP	dsp.ti.com
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
Low Power Wireless	www.ti.com/lpw

Applications

Audio	www.ti.com/audio
Automotive	www.ti.com/automotive
Broadband	www.ti.com/broadband
Digital Control	www.ti.com/digitalcontrol
Military	www.ti.com/military
Optical Networking	www.ti.com/opticalnetwork
Security	www.ti.com/security
Telephony	www.ti.com/telephony
Video & Imaging	www.ti.com/video
Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2007, Texas Instruments Incorporated