

TMS320x281x DSP

システム制御および割り込み

リファレンス・ガイド

TMS320x281x 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の意味しません。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 ©2004, 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

- 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

- 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。
- 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

最初にお読みください

本書について

本書では、様々な 281x デジタル・シグナル・プロセッサ (DSP) のシステム制御および割り込みがペリフェラルでどのように動作するかについて説明します。これには、以下の情報が含まれます。

- 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
- 28x デバイ스에組み込まれたセキュリティ機能であるコード・セキュリティ・モジュール (CSM)
- オシレータ、PLL、ウォッチドッグ機能、および低電力モードを含むクロック・メカニズム
- 共有ピンの動作選択に使用される GPIO MUX レジスタ
- ペリフェラル・フレームおよびデバイス・エミュレーション・レジスタ
- 多数の割り込みソースをより小さい割り込み入力のセットに多重化するペリフェラル割り込み拡張 (PIE) ブロック

TI の関連資料

以下の資料では、TI の Web サイトで提供されている TMS320x281x および関連サポート・ツールについて説明しています。

『**TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, and TMS320C2812 Digital Signal Processors**』 (文書番号 SPRS174) データシートには、これらのデバイスの電気仕様とタイミング仕様、および使用可能なすべてのパッケージの信号説明とピンアウトが含まれます。

『**TMS320R2811 and TMS320R2812 Digital Signal Processors**』 (文書番号 SPRS257) データシートには、これらのデバイスの電気仕様とタイミング仕様、および使用可能なすべてのパッケージの信号説明とピンアウトが含まれます。

『**TMS320C28x DSP CPU 及びインストラクション・セット・リファレンス・ガイド**』 (文書番号 SPRU430) では、TMS320C28x™ 固定小数点デジタル・シグナル・プロセッサ (DSP) の中央演算処理ユニット (CPU) およびアセンブリ言語命令について説明します。また、これらの DSP で使用可能なエミュレーション機能についても説明します。

『**TMS320x281x Analog-to-Digital Converter (ADC) Reference Guide**』(文書番号 SPRU060)では、ADC モジュールについて説明します。このモジュールは、12 ビット・パイプライン ADC です。本書でコアと呼ぶこのコンバータのアナログ回路には、フロントエンド・アナログ・マルチプレクサ (MUX)、サンプル・ホールド (S/H) 回路、変換コア、電圧レギュレータ、およびその他のアナログ・サポート回路が含まれます。本書でラッパーと呼ぶデジタル回路は、プログラマブル変換シーケンサ、リザルト・レジスタ、アナログ回路へのインターフェイス、デバイス・ペリフェラル・バスへのインターフェイス、およびその他のオンチップ・モジュールへのインターフェイスが含まれます。

『**TMS320x281x Boot ROM Reference Guide**』(文書番号 SPRU095)では、ブートルoader (製造時にプログラムされたブートルoad・プログラム) の目的と機能について説明します。また、デバイス・オンチップ・ブートルoad のその他の内容についても説明し、すべての情報がそのメモリ内のどこに配置されているかを示します。

『**TMS320x281x, 280x Enhanced Controller Area Network (eCAN) Reference Guide**』(文書番号 SPRU074)では、一般的な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連続的に通信を行う eCAN について説明します。eCAN モジュールは、設定可能な 32 個のメールボックスとタイムスタンプ機能を備え、多用途で堅牢なシリアル通信インターフェイスを提供します。C28x DSP に実装された 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。

『**TMS320x281x Event Manager (EV) Reference Guide**』(文書番号 SPRU065)では、特にモーション制御およびモータ制御アプリケーションに役立つ幅広い機能を提供する EV モジュールについて説明します。EV モジュールには、汎用 (GP) タイマ、フル・コンペア /PWM ユニット、キャプチャ・ユニット、およびエンコーダ・パルス (QEP) 回路が含まれます。

『**TMS320x281x External Interface (XINTF) Reference Guide**』(文書番号 SPRU067)では、28x デジタル・シグナル・プロセッサ (DSP) の外部インターフェイス (XINTF) について説明します。

『**TMS320x281x Multi-channel Buffered Serial Ports (McBSPs) Reference Guide**』(文書番号 SPRU061)では、C28x デバイスで使用可能な McBSP について説明します。McBSP により、DSP とシステム内の他のデバイス間でダイレクト・インターフェイスを利用できます。

『**TMS320x281x, 280x Peripheral Reference Guide**』(文書番号 SPRU566)では、28x デジタル・シグナル・プロセッサ (DSP) のペリフェラル・リファレンス・ガイドについて説明します。

『**TMS320x281x, 280x Serial Communication Interface (SCI) Reference Guide**』(文書番号 SPRU051)では、一般に UART として知られる、2 線式非同期シリアル・ポートである SCI について説明します。SCI モジュールは、CPU と、NRZ (non-return-to-zero) 形式を使用する他の非同期ペリフェラル間のデジタル通信をサポートします。

『TMS320x281x, 280x Serial Peripheral Interface (SPI) Reference Guide』(文書番号 SPRU059) では、SPI について説明します。SPI は、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デバイスにシフトインおよびシフトアウトすることのできる高速同期シリアル入出力 (I/O) ポートです。SPI は、DSP コントローラと外部ペリフェラルまたは別のコントローラ間の通信に使用されます。

『TMS320x281x システム制御および割り込みリファレンス・ガイド』(文書番号 SPRU078) (本書) では、281x デジタル・シグナル・プロセッサ (DSP) の様々な割り込みおよびシステム制御機能について説明します。

『The TMS320C28x Instruction Set Simulator Technical Overview』(文書番号 SPRU608) では、C28x コアの命令セットをシミュレートする、Code Composer Studio for TMS320C2000 IDE で使用可能なシミュレータについて説明します。

『TMS320C28x DSP/BIOS Application Programming Interface (API) Reference Guide』(文書番号 SPRU625) では、DSP/BIOS を使用した開発について説明します。

『3.3 V DSP Digital Motor Control Application Report』(文書番号 SPRA550)。新世代のモータ制御デジタル・シグナル・プロセッサ (DSP) は、供給電圧を 5V から 3.3V に低電圧化し、低コストで高パフォーマンスを実現します。従来の 5V デジタル制御回路から 3.3V 設計への置き換えには追加のシステム・コストを伴わず、また TTL および CMOS 互換の構成要素とのインターフェイスや、パワー・トランジスタ・ゲート・ドライバなどの混合電圧 IC とのインターフェイスに重大な問題が発生することはありません。5V ベースの設計と同様、高/低電圧アナログおよび切り替え信号を使用するモータ制御システムなどの混合信号環境で 3.3V DSP、ADC、およびデジタル回路を使用するときは、技術的手法を用い、構成要素のレイアウトと PCB の設計を適切に行うことによってノイズや EMI 効果を最小限に抑える必要があります。また、TI の TMS320x24xx DSP コントローラの特長機能で使用可能なランダム PWM 手法などのソフトウェア技術を使用し、EMI 放射によるノイズ効果を大幅に低減することもできます。

このアプリケーション・レポートでは、低 HP モータ制御アプリケーションにおける 3.3V DSP と 5V DSP の設計について比較考察します。まず、3.3V のみのモータ・コントローラのシナリオについて説明し、多くのアプリケーションでは 3.3V と 5V 間のインターフェイスに重大な問題がないことを示します。次に、3.3V と 5V 間のインターフェイスを必要とする状況でのコスト効率の高い 3.3V と 5V 間のインターフェイス手法について説明します。また、オンチップ 3.3V ADC と 5V ADC の違いについて説明し、3.3V および 5V ADC 変換における感度とノイズについて取り上げます。最終セクションでは、システムのノイズと EMI 効果の低減に役立つ構成要素のレイアウトとプリント回路基板 (PCB) 設計のガイドラインを要約します。

『Thermo-Electric Cooler Control Using a TMS320F2812 DSP & DRV592 Power

Amplifier Application Note』 (文書番号 SPRA873)。このアプリケーション・レポートでは、TI の TMS320F2812 デジタル・シグナル・プロセッサ (DSP) と DRV592 パワー・アンプで構成された電子冷却システムを紹介し、DSP 内蔵の 12 ビット・アナログ・デジタル・コンバータによってサーミスタをリードし、パルス幅変調波形を H ブリッジ DRV592 パワー・アンプに直接出力するデジタル比例 / 積分 / 微分フィードバック・コントローラを実装します。ここで紹介するシステムでは、レーザ・マウントに最大 6.1 ワットの加熱または冷却を供給しますが、DRV592 アンプを適切に構成した場合、実際には最大 15 ワットの供給が可能です。閉じたループの TEC システムでは、必要な動作温度範囲に応じ、14 ~ 16 秒のステップ応答整定時間で ± 0.0006 °C の温度精度を実現可能と考えられます。実験システムの詳細、およびソフトウェアとソフトウェアの操作手順について説明します。

『Running an Application from Internal Flash Memory on the TMS320F281x DSP

Application Report』 (文書番号 SPRA958)。TMS320F28x DSP でオンチップ・フラッシュ・メモリからアプリケーションを実行するためには、特殊な要求がいくつか存在します。Code Composer Studio™ デバッガでは、初期化済みのセクションに関連する問題およびその問題とメモリとの関連性が隠蔽される可能性があるため、通常、RAM の開発中にこれらの要件が明白に示されることはありません。このアプリケーション・レポートでは、オンチップ・フラッシュ・メモリから実行するアプリケーション・ソフトウェアを適切に構成するための要件を示します。DSP/BIOS™ および非 DSP/BIOS プロジェクトの両方の要件を示します。パフォーマンスに関する考慮事項と技法についても説明します。eZdsp™ F2812 開発基板 (あるいは任意の F2812、F2811、または F2810 DSP 基板) でオンチップ・フラッシュから実行するサンプル・コード・プロジェクトを掲載します。また、完全を期すために、内部 RAM から実行するコード例も掲載します。これらのコード例は、必要に応じてコード開発の初めからも利用できます。

商標

Code Composer Studio および C28x は、テキサス・インスツルメンツの商標です。

余白

目次

1	メモリ	1-1
	フラッシュ・メモリおよびワンタイム・プログラマブル (OTP) メモリについて説明します。	
1.1	フラッシュ・メモリと OTP メモリ	1-2
1.1.1	フラッシュ・メモリ	1-2
1.1.2	OTP メモリ	1-2
1.2	フラッシュおよび OTP の電力モード	1-3
1.2.1	フラッシュおよび OTP のパフォーマンス	1-4
1.2.2	28x フラッシュ・パイプライン・モード	1-5
1.2.3	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	1-7
1.3	フラッシュ・レジスタと OTP レジスタ	1-8
2	コード・セキュリティ・モジュール (CSM)	2-1
	コード・セキュリティ・モジュール (CSM) について説明します。	
2.1	機能の説明	2-2
2.2	CSM によるほかのオンチップ・リソースへの影響	2-4
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	2-6
2.3.1	セキュリティのロック解除が必要な環境	2-7
2.3.2	パスワード・マッチ・フロー	2-8
2.3.3	コード・セキュリティを持つ / 持たないデバイスをアンセキュアにする際の 考慮事項	2-10
2.3.4	アンセキュアにする C コード例	2-11
2.3.5	再セキュアにする C コード例	2-12
2.4	セキュリティ・ロジックを保護するための必要事項と禁止事項	2-13
2.4.1	必要事項	2-13
2.4.2	禁止事項	2-13
2.5	CSM の機能 - 要約	2-14
3	クロック	3-1
	クロックとシステム制御について説明します。	
3.1	クロックとシステム制御	3-2
3.2	OSC と PLL ブロック	3-9
3.2.1	PLL ベースのクロック・モジュール	3-9
3.2.2	外部基準オシレータ・クロック・オプション	3-11
3.3	低電力モード・ブロック	3-12
3.4	ウォッチドッグ・ブロック	3-15
3.4.1	エミュレーションの考慮事項	3-18

3.5	32 ビット CPU タイマ 0/1/2	3-19
4	汎用 I/O (GPIO)	4-1
	<i>GPIO 共有ピンと動作の選択について説明します。</i>	
4.1	GPIO MUX	4-2
4.2	入力フィルタリング (Qualification)	4-6
4.3	レジスタの機能概要	4-8
4.4	レジスタ・ビットと I/O のマッピング	4-11
5	ペリフェラル・フレーム	5-1
	<i>28x デバイスのペリフェラルに関する設定方法について説明します。</i>	
5.1	ペリフェラル・フレーム・レジスタ	5-2
5.2	EALLOW 保護レジスタ	5-5
5.3	デバイス・エミュレーション・レジスタ	5-10
5.4	ライト後のリード保護	5-13
6	ペリフェラル割り込み拡張 (PIE)	6-1
	<i>ペリフェラル割り込み拡張 (PIE) について説明します。</i>	
6.1	PIE コントローラの概要	6-2
6.2	ベクタ・テーブルのマッピング	6-7
6.3	割り込みソース	6-10
6.3.1	マルチプレクスされた割り込みの処理手順	6-11
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にした後 ディスエーブルにする手順	6-12
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要求のフロー	6-14
6.3.4	PIE ベクタ・テーブル	6-16
6.4	PIE コンフィギュレーション・レジスタ	6-24
6.5	PIE 割り込みレジスタ	6-26
6.5.1	PIE 割り込みフラグ・レジスタ	6-27
6.5.2	PIE 割り込みイネーブル・レジスタ	6-28
6.5.3	CPU 割り込みフラグ・レジスタ (IFR)	6-28
6.5.4	割り込みイネーブル・レジスタ (IER) とデバッグ割り込みイネーブル・レジスタ (DBGIER)	6-32
6.6	外部割り込み制御レジスタ	6-39



1-1.	フラッシュの電力モード状態図	1-4
1-2.	フラッシュ・パイプライン	1-6
1-3.	フラッシュの設定方法のフロー・チャート	1-8
1-4.	フラッシュ・オプション (FOPT) レジスタ	1-10
1-5.	フラッシュ・パワー・レジスタ (FPWR)	1-10
1-6.	フラッシュ・ステータス・レジスタ (FSTATUS)	1-11
1-7.	フラッシュ・スタンバイ・ウェイト (FSTDBYWAIT) レジスタ	1-12
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 (FACTIVEWAIT) レジスタ	1-13
1-9.	フラッシュ・ウェイト・ステート (FBANKWAIT) レジスタ	1-13
1-10.	OTP ウェイト・ステート (FOTPWAIT) レジスタ	1-14
2-1.	CSM ステータスおよび制御 (CSMSCR) レジスタ	2-7
2-2.	パスワード・マッチ・フロー (PMF)	2-9
3-1.	クロックおよびリセット・ドメイン	3-2
3-2.	ペリフェラル・クロック制御 (PCLKCR) レジスタ	3-4
3-3.	システム制御およびステータス (SCSR) レジスタ	3-5
3-4.	高速ペリフェラル・クロック・プリスケラ (HISPCP) レジスタ	3-6
3-5.	低速ペリフェラル・クロック・プリスケラ (LOSPCP) レジスタ	3-7
3-6.	OSC と PLL ブロック	3-9
3-7.	PLLCR レジスタ	3-10
3-8.	低電力モード制御 0 (LPMCR0) レジスタ	3-13
3-9.	低電力モード制御 1 (LPMCR1) レジスタ	3-13
3-10.	ウォッチドッグ・モジュール	3-15
3-11.	ウォッチドッグ・カウンタ (WDCNTR) レジスタ	3-16
3-12.	ウォッチドッグ・リセット・キー (WDKEY) レジスタ	3-16
3-13.	ウォッチドッグ制御 (WDCR) レジスタ	3-17
3-14.	CPU タイマ	3-19
3-15.	CPU タイマの割り込み信号と出力信号	3-19
3-16.	TIMERxTIM レジスタ	3-21
3-17.	TIMERxTIMH レジスタ	3-21
3-18.	TIMERxPRD レジスタ (注 1)	3-22
3-19.	TIMERxPRDH レジスタ (注 1)	3-22
3-20.	TIMERxTCR レジスタ (注 1)	3-23
3-21.	TIMERxTPR レジスタ (注 1)	3-24
3-22.	TIMERxTPRH レジスタ (注 1)	3-25
4-1.	マルチプレクスされた GPIO/ ペリフェラル・ピン	4-5

4-2.	タイプ 1 の入力フィルタリング (Qualification)	4-6
4-3.	入力フィルタリング (Qualifier) クロック・サイクル	4-7
4-4.	タイプ 2 の入力フィルタリング (Qualification)	4-7
4-5.	GPIO A 入力 Qualification 制御 (GPAQUAL) レジスタ	4-12
4-6.	GPIO B 入力 Qualification 制御 (GPBQUAL) レジスタ	4-13
4-7.	GPIO D 入力 Qualification 制御 (GPDQUAL) レジスタ	4-14
4-8.	GPIO E 入力 Qualification 制御 (GPEQUAL) レジスタ	4-16
5-1.	デバイス・コンフィギュレーション (DEVICECNF) レジスタ	5-10
5-2.	パーツ ID レジスタ	5-11
5-3.	DEVICEID レジスタ	5-12
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	6-2
6-2.	代表的な PIE/CPU 割り込み応答 - INTx.y	6-5
6-3.	リセット・フロー図	6-9
6-4.	割り込みソース	6-10
6-5.	マルチプレクスされた割り込み要求のフロー図	6-14
6-6.	PIE 割り込み ACK レジスタ (PIEACK) レジスタ - アドレス CE1	6-26
6-7.	PIEIFRx レジスタ (x=1 ~ 12)	6-27
6-8.	PIEIERx レジスタ (x=1 ~ 12)	6-28
6-9.	割り込みフラグ・レジスタ (IFR) - CPU レジスタ	6-30
6-10.	割り込みイネーブル・レジスタ (IER) - CPU レジスタ	6-33
6-11.	デバッグ割り込みイネーブル・レジスタ (DBGIER) - CPU レジスタ	6-36
6-12.	外部割り込み 1 制御レジスタ (XINT1CR) - アドレス 7070h	6-39
6-13.	外部割り込み 2 制御レジスタ (XINT2CR) - アドレス 7071h	6-40
6-14.	外部 NMI 割り込み制御レジスタ (XNMICR) - アドレス 7077h	6-40
6-15.	外部割り込み 1 カウンタ (XINT1CTR) - アドレス 7078h	6-42
6-16.	外部割り込み 2 カウンタ (XINT2CTR) - アドレス 7079h	6-42
6-17.	外部 NMI 割り込みカウンタ (XNMICTR) - アドレス 707Fh	6-43

表

1-1.	フラッシュ /OTP コンフィギュレーション・レジスタ (1)	1-9
1-2.	フラッシュ・オプション (FOPT) レジスタのフィールドの説明	1-10
1-3.	フラッシュ電力レジスタ (FPWR) のフィールドの説明	1-11
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	1-11
1-5.	フラッシュ・スタンバイ・ウェイト (FSTDYWAIT) レジスタの フィールドの説明	1-12
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 (FACTIVEWAIT) レジスタのフィールドの説明 1-13	
1-7.	フラッシュ・ウェイト・ステート (FBANKWAIT) レジスタのフィールドの説明	1-13
1-8.	OTP ウェイト・ステート (FOTPWAIT) レジスタ	1-14
2-1.	セキュリティ・レベル	2-2
2-2.	CSM の影響を受ける F281x/C281x リソース	2-4
2-3.	CSM の影響を受けない 281x リソース	2-5
2-4.	コード・セキュリティ・モジュール (CSM) レジスタ	2-6
2-5.	CSM ステータスおよび制御 (CSMSCR) レジスタのフィールドの説明	2-7
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 (注 1)	3-3
3-2.	ペリフェラル・クロック制御 (PCLKCR) レジスタのフィールドの説明	3-4
3-3.	システム制御およびステータス (SCSR) レジスタのフィールドの説明	3-5
3-4.	高速ペリフェラル・クロック・プリスケラ (HISPCP) レジスタの フィールドの説明	3-7
3-5.	低速ペリフェラル・クロック・プリスケラ (LOSPCP) レジスタの フィールドの説明	3-7
3-6.	可能な PLL 設定モード	3-10
3-7.	PLLCR レジスタのフィールドの説明	3-10
3-8.	281x の低電力モード	3-12
3-9.	低電力モード制御 0 (LPMCR0) レジスタのフィールドの説明	3-13
3-10.	低電力モード制御 1 (LPMCR1) レジスタのフィールドの説明	3-14
3-11.	ウォッチドッグ・カウンタ (WDCNTR) レジスタのフィールドの説明	3-16
3-12.	ウォッチドッグ・リセット・キー (WDKEY) レジスタのフィールドの説明	3-16
3-13.	ウォッチドッグ制御 (WDCR) レジスタのフィールドの説明	3-17
3-14.	CPU タイマ 0、1、2 のコンフィギュレーション・レジスタと制御レジスタ	3-20
3-15.	TIMERxTIM レジスタのフィールドの説明	3-21
3-16.	TIMERxTIMH レジスタのフィールドの説明	3-21
3-17.	TIMERxPRD レジスタのフィールドの説明	3-22
3-18.	TIMERxPRDH レジスタのフィールドの説明	3-22
3-19.	TIMERxTCR レジスタのフィールドの説明	3-23

3-20.	TIMERxTPR レジスタのフィールドの説明	3-24
3-21.	TIMERxTPRH レジスタのフィールドの説明	3-25
4-1.	GPIO MUX レジスタ (注 1, 2)	4-2
4-2.	GPIO データ・レジスタ (注 1, 2)	4-3
4-3.	GPIO A レジスタ・ビットと I/O のマッピング	4-11
4-4.	GPIO A 入力 Qualification 制御 (GPAQUAL) レジスタのフィールドの説明	4-12
4-5.	GPIO B レジスタ・ビットと I/O のマッピング	4-12
4-6.	GPIO B 入力 Qualification 制御 (GPBQUAL) レジスタのフィールドの説明	4-13
4-7.	GPIO D レジスタ・ビットと I/O のマッピング	4-14
4-8.	GPIO D 入力 Qualification 制御 (GPDQUAL) レジスタのフィールドの説明	4-15
4-9.	GPIO E MUX レジスタ・ビットと I/O のマッピング	4-15
4-10.	GPIO E 入力 Qualification 制御 (GPEQUAL) レジスタのフィールドの説明	4-16
4-11.	GPIO F レジスタ・ビットと I/O のマッピング	4-16
4-12.	GPIO G レジスタ・ビットと I/O のマッピング	4-17
5-1.	ペリフェラル・フレーム 0 のレジスタ	5-2
5-2.	ペリフェラル・フレーム 1 のレジスタ	5-3
5-3.	ペリフェラル・フレーム 2 のレジスタ (注 1)	5-3
5-4.	EALLOW 保護レジスタへのアクセス	5-5
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	5-5
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	5-6
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) のレジスタ	5-6
5-8.	EALLOW 保護されている PIE ベクタ・テーブル	5-7
5-9.	EALLOW 保護されている PLL、クロッキング、ウォッチドッグ、低電力モードのレジスタ	5-8
5-10.	EALLOW 保護されている GPIO MUX レジスタ	5-8
5-11.	EALLOW 保護されている eCAN レジスタ	5-9
5-12.	デバイス・エミュレーション・レジスタ	5-10
5-13.	デバイス・コンフィギュレーション (DEVICECNF) レジスタの フィールドの説明	5-10
5-14.	パーツ ID レジスタのフィールドの説明	5-11
5-15.	DEVICEID レジスタのフィールドの説明	5-12
5-16.	PROTSTART および PROTRANGE レジスタ	5-13
5-17.	PROTSTART 有効値 (注 1)	5-14
5-18.	PROTRANGE 有効値 (注 2)	5-15
6-1.	割り込みをイネーブルにする	6-6
6-2.	割り込みベクタ・テーブル・マッピング (注 1)	6-7
6-3.	リセット・オペレーション後のベクタ・テーブル・マッピング (注 1)	6-8
6-4.	281x の PIE ベクタ・テーブル	6-16
6-5.	281x PIE ペリフェラル割り込み (注 1)	6-23
6-6.	PIE コンフィギュレーションおよび制御レジスタ	6-24
6-7.	PIECTRL レジスタ - アドレス CE0	6-26
6-8.	PIECTRL レジスタ - フィールドの説明	6-26
6-9.	PIE 割り込み ACK レジスタ (PIEACK) レジスタのフィールドの説明	6-26
6-10.	PIEIFRx レジスタ (x=1 ~ 12) フィールドの説明	6-27
6-11.	PIEIERx レジスタ (x=1 ~ 12) のフィールドの説明	6-28
6-12.	割り込みフラグ・レジスタ (IFR) のフィールドの説明	6-30
6-13.	割り込みイネーブル・レジスタ (IER) のフィールドの説明	6-33
6-14.	デバッグ割り込みイネーブル・レジスタ (DBGIER) のフィールドの説明	6-36
6-15.	外部割り込み 1 制御レジスタ (XINT1CR) のフィールドの説明	6-39

6-16.	外部割り込み 2 制御レジスタ (XINT2CR) のフィールドの説明	6-40
6-17.	外部 NMI 割り込み制御レジスタ (XNMICR) のフィールドの説明	6-41
6-18.	XNMICR レジスタのセットと割り込みソース	6-41
6-19.	外部割り込み 1 カウンタ (XINT1CTR) のフィールドの説明	6-42
6-20.	外部割り込み 2 カウンタ (XINT2CTR) のフィールドの説明	6-42
6-21.	外部 NMI 割り込みカウンタ (XNMICTR) のフィールドの説明	6-43

表

余白

メモリ

この章では、28x デジタル・シグナル・プロセッサ (DSP) デバイスおよびペリフェラルでフラッシュ・メモリおよびワンタイム・プログラマブル (OTP) メモリをどのように使用できるかについて説明します。また、メモリに関連するレジスタについても説明します。

項目	ページ
1.1 フラッシュ・メモリと OTP メモリ.....	1-2
1.2 フラッシュおよび OTP の電力モード.....	1-3
1.3 フラッシュ・レジスタと OTP レジスタ.....	1-8

1.1 フラッシュ・メモリとOTPメモリ

このセクションでは、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2種類のメモリを設定する方法について説明します。

1.1.1 フラッシュ・メモリ

フラッシュ・デバイスに含まれるオンチップ・フラッシュは、プログラムおよびデータ・メモリ空間両方にマップされます。このフラッシュ・メモリは、28x デバイスでは常にイネーブルになり、以下の機能を提供します。

- 複数セクタ
- コード・セキュリティ
- 低電力モード
- CPU 周波数に基づいて設定可能な、ウェイト・ステート
- リニア・コードの実行性能を向上させるフラッシュ・パイプライン・モード

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、データまたはコードをプログラムするために使用できます。このブロックは1度のみプログラム可能で、消去することはできません。

1.2 フラッシュおよびOTPの電力モード

フラッシュ・メモリおよびOTPメモリには、以下の動作状態が適用されます。

- リセットまたはスリープ状態：これは、デバイスの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はスリープ状態（最小電力）になります。フラッシュ/OTPメモリ・マップ領域に対するCPUリードまたはフェッチ・アクセスは、CPUをストールさせます。このアクセスにより、電力モードをアクティブまたはリード状態に自動的に変更します。
- 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。フラッシュ/OTPメモリ・マップ領域に対するCPUリードまたはフェッチ・アクセスは、CPUをストールさせます。このアクセスにより、電力モードをアクティブ状態に自動的に変更します。
- 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（最高電力）になります。フラッシュ/OTPメモリ・マップ領域に対するCPUリードまたはフェッチ・アクセスのウェイト・ステートは、FBANKWAITおよびFOTPWAITレジスタによって制御されます。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・コードの実行におけるフェッチ・パフォーマンスを向上させることもできます。

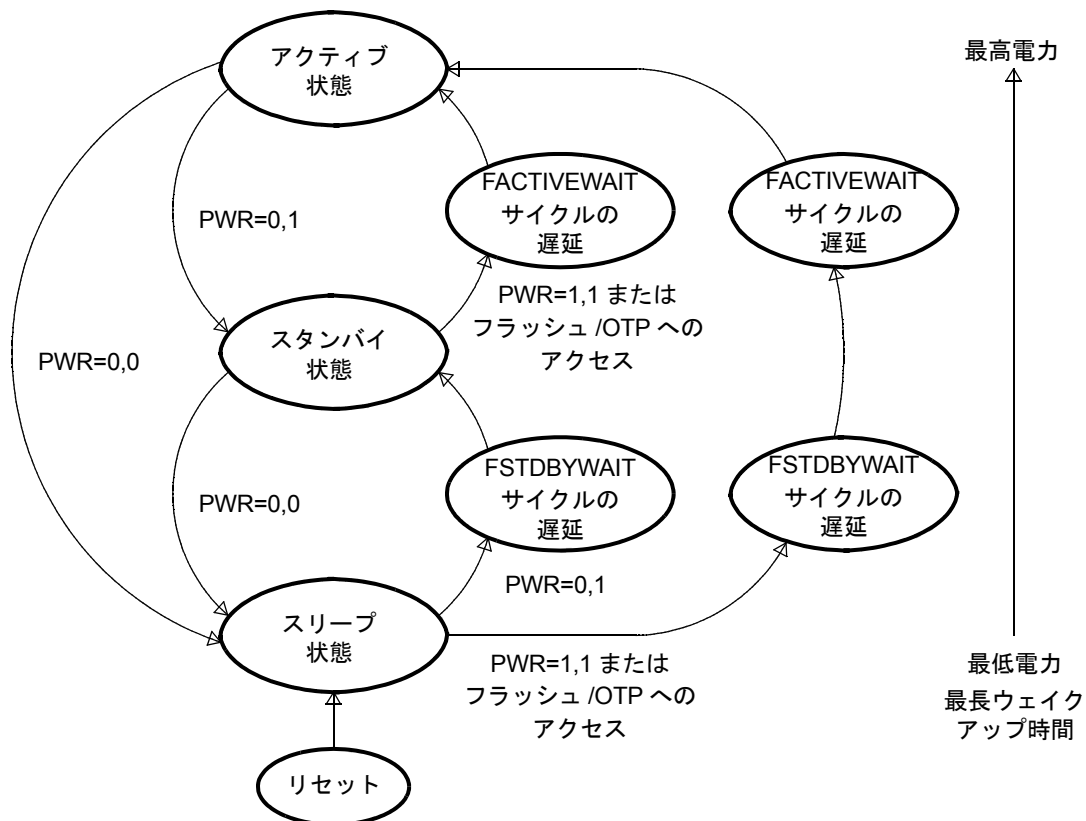
フラッシュ/OTPからのリードまたは実行オペレーション中、フラッシュ/OTPバンクおよびポンプは、常に同じ電力モードになります。これらの状態図については、図1-1を参照してください。

フラッシュ/OTPメモリの現行の電力状態は、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、PWRモード・ビットを高い電力モードから低い電力モードに変更します。この変更により、フラッシュ/OTPバンクはより低い電力状態に即座に移行します。このレジスタは、フラッシュ/OTPメモリ以外にあるコードによってのみアクセスされる必要があります。
- より低い電力状態からより高い電力状態に移行するには、以下の2つの方法があります。
 - FPWRレジスタを0から1に変更します。このアクセスにより、フラッシュ/OTPメモリはアクティブになります。
 - リード・アクセスまたはプログラム・フェッチ・アクセスによってフラッシュ・メモリまたはOTPメモリにアクセスします。このアクセスにより、フラッシュ/OTPメモリは自動的にアクティブ状態になります。

より低い電力状態からより高い電力状態に移行するときに、遅延が生じます。図1-1を参照してください。この遅延は、フラッシュがより高い電力モードで安定するために必要です。この遅延の間にフラッシュ/OTPメモリに対するアクセスが発生すると、CPUは遅延が完了するまで自動的にストールします。

図 1-1 フラッシュの電力モード状態図



遅延の期間は、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だけ遅れます。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だけ遅れます。スリープ・モード（最低電力）からアクティブ・モード（最高電力）への移行は、FSTDBYWAIT と FACTIVEWAIT を加算した分だけ遅れます。

1.2.1 フラッシュおよびOTPのパフォーマンス

フラッシュ / OTP に対する CPU リードまたは フェッチ・オペレーションは、以下のいずれかの形を取ることができます。

- 32 ビット命令フェッチ
- 16 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後、バンク・メモリ・マップ領域に対するリードまたはフェッチ・アクセスは、以下の3つのタイプに分類できます。

- フラッシュ・メモリ・ランダム・アクセス：ランダム・アクセスのウェイト・ステートの数は、FBANKWAIT レジスタの RANDWAIT ビットで構成されます。このレジスタは、デフォルトでワーストケース・カウント（最大ウェイト数）に設定されるため、ユーザーは、CPU のクロック・レートとフラッシュのアクセス時間に基づいてパフォーマンスを向上させるために、適切なウェイト・ステート数をプログラムする必要があります。
- フラッシュ・メモリ・ページ・アクセス：フラッシュ配列は行と列に編成されています。行には、2048 ビットの情報が含まれます。特定の行に対する最初のアクセスは、ランダム・アクセスとみなされます。同じ行内での以降のアクセスは、より高速なアクセス時間で行えます。これはページ・アクセスと呼ばれます。
フラッシュでは、ユーザーが FBANKWAIT レジスタの PAGEWAIT ビットに少ない数のウェイト・ステートを設定することで、このアクセスを利用できます。このモードは、データ空間およびプログラム空間リードのほか、命令フェッチで動作します。フラッシュのアクセス時間の詳細については、デバイスのデータシートを参照してください。
- OTP アクセス：OTP に対するリードまたはフェッチ・アクセスは、FOTPWAIT レジスタの OTPWAIT レジスタ・ビットで制御されます。OTP へのアクセスにはフラッシュより時間がかかり、ページ・モードはありません。

- 注：
- 1) フラッシュ/OTP メモリ・マップ領域へのライトは無視されます。これらは、単一サイクルで完了します。
 - 2) コード・セキュリティ・モジュール（CSM）がセキュアな場合、セキュア・ゾーン外部からフラッシュ/OTP メモリ・マップ領域へのリードは、通常のアクセスと同じサイクル数で行われます。ただし、リード・オペレーションはゼロを返します。
 - 3) フラッシュは、PAGEWAIT ビットがゼロに設定されている場合に、0 ウェイト・アクセスをサポートします。これは、CPU 速度が十分に低く、アクセス時間に適応できることを前提としています。

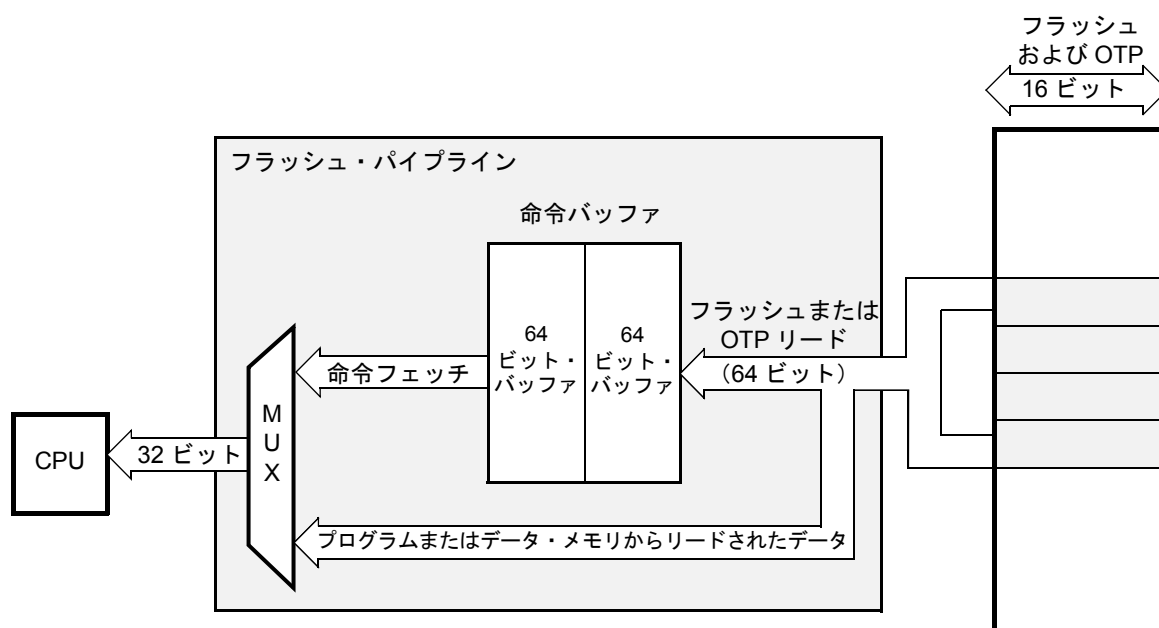
1.2.2 28x フラッシュ・パイプライン・モード

通常、フラッシュ・メモリは、アプリケーション・コードを保存するために使用されます。コードの実行中、中断が発生した場合を除き、命令は連続したメモリ・アドレスからフェッチされます。一般に、アプリケーション・コードの大部分は、連続したアドレスに置かれたコード部分から構成されます。このコード部分はリニア・コードと呼ばれます。リニア・コードの実行パフォーマンス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。このモードは、FOPT レジスタの ENPIPE ビットを設定することによって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 パイプラインから独立しています。また、フラッシュおよび ROM デバイス間のコード・タイミング互換性を維持できるようにするために、ROM デバイスにもフラッシュ・パイプライン・モードが実装されています。

フラッシュまたは OTP からの命令フェッチは、1 アクセスあたり 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、フェッチされる命令を含む 64 ビット境界に自動的にアラインされます。フラッシュ・パイプライン・モードがイネーブルの場合（図 1-2 を参照）、命令フェッチからの 64 ビット・リードは、64 ビット、2 レベルの命令プリフェッチ・バッファにストアされます。その後、このプリフェッチ・バッファの内容は、処理のために必要に応じて CPU に送信されます。

単一の64ビット・アクセスには、最大2個の32ビット命令または最大4個の16ビット命令が存在します。C28x命令のほとんどは16ビットであるため、フラッシュ・バンクからのすべての64ビット命令フェッチについて、CPUで処理可能な最大4個の命令がプリフェッチ・バッファに存在することになります。これらの命令を処理している間、フラッシュ・パイプラインは、次の64ビットをプリフェッチするために、フラッシュ・バンクに対する次のアクセスを自動的に開始します。この方法で、フラッシュ・パイプライン・モードは、命令プリフェッチ・バッファが可能な限りいっぱいになるようバックグラウンドで動作します。このモードを使用することで、フラッシュまたはOTPからの連続的なコード実行における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、またはループなどの命令の実行によってPCの流れが中断した場合のみ中止されます。この場合、プリフェッチが中止され、プリフェッチ・バッファの内容はフラッシュされます。これが発生する状況として、以下の2つが考えられます。

- 飛び先アドレスがフラッシュまたはOTPの場合、プリフェッチが中止され、その飛び先アドレスで再開されます。

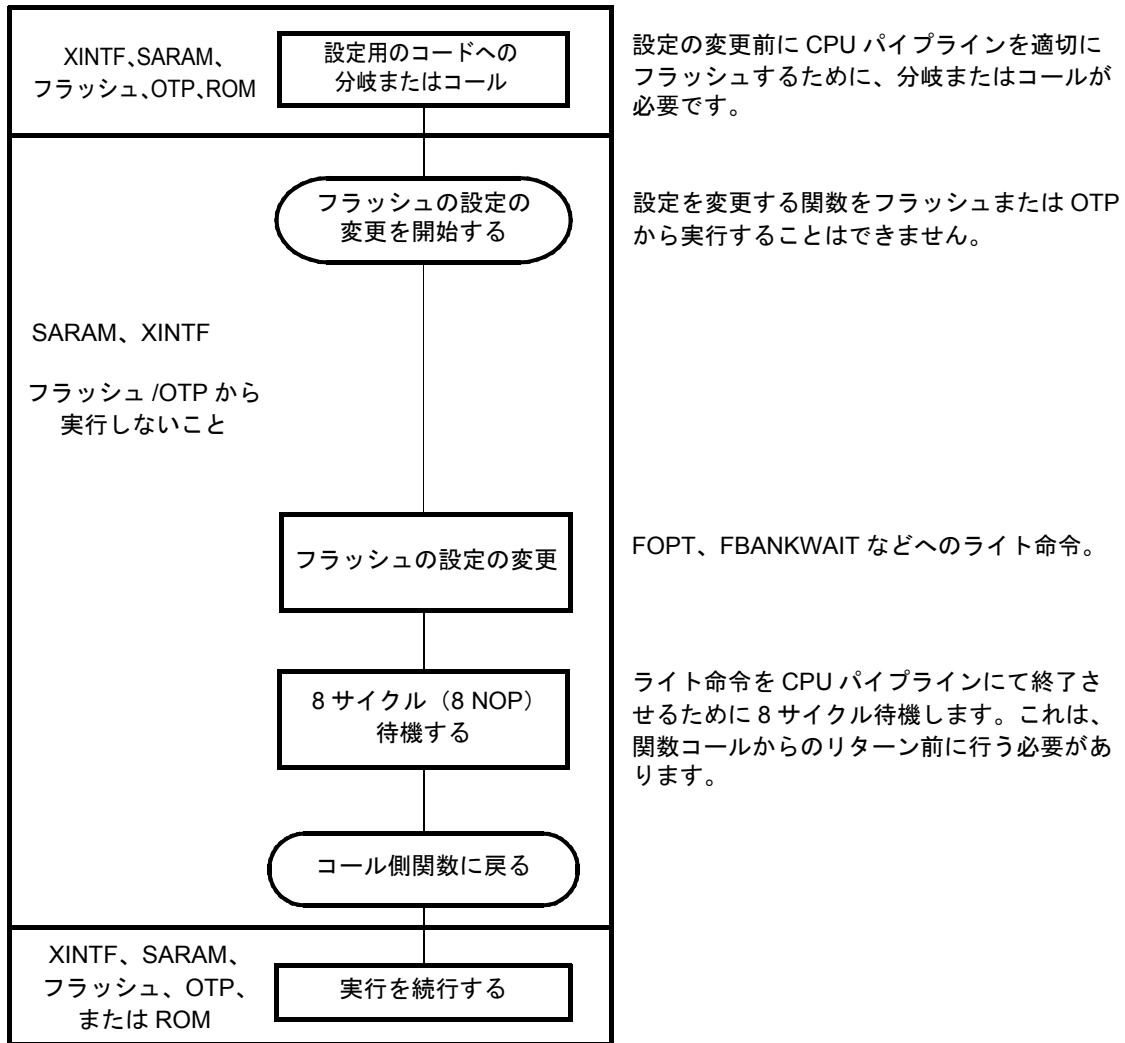
- 飛び先アドレスがフラッシュまたは OTP 以外の場合、プリフェッチが中止され、フラッシュまたは OTP への分岐が行われた場合のみ再度開始されます。フラッシュ・パイプライン・プリフェッチ・メカニズムは、プログラム空間からの命令フェッチにのみ適用されます。データ・メモリおよびプログラム・メモリからのデータ・リードは、プリフェッチ・バッファ機能を利用せず、プリフェッチ・バッファをバイパスします。たとえば、MAC、DMAC、PREAD などの命令は、プログラム・メモリからデータ値をリードします。このリードが行われるときにプリフェッチ・バッファはバイパスされますが、バッファはフラッシュされません。データ・リード・オペレーションが開始されたときにすでに命令プリフェッチが進行中である場合、プリフェッチが完了するまでデータ・リードはストールします。

1.2.3 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

フラッシュの設定を行っている間は、フラッシュまたは OTP へのアクセスをすることはできません。これには、CPU パイプラインに残された命令、データ・リード、および命令プリフェッチ・オペレーションが含まれます。設定の変更中にアクセスが行われないようにするには、FOPT、FPWR、FBANKWAIT、または FOTPWAIT レジスタを変更するすべてのコードについて、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 に置き換えられたデバイス上の ROM にも適用されます。

図 1-3 フラッシュの設定方法のフロー・チャート



1.3 フラッシュ・レジスタと OTP レジスタ

フラッシュ・メモリおよび OTP メモリは、表 1-1 に示されたレジスタによって設定することができます。ビットの説明は、図 1-4 から 図 1-6 に記します。

表 1-1 フラッシュ /OTP コンフィギュレーション・レジスタ⁽¹⁾

名前	アドレス	サイズ (× 16)	説明
コンフィギュレーション・レジスタ			
FOPT	0x0000-0A80	1	フラッシュ・オプション・レジスタ
予約済み	0x0000-0A81	1	予約済み
FPWR	0x0000-0A82	1	フラッシュ・パワー・モード・レジスタ
FSTATUS	0x0000-0A83	1	ステータス・レジスタ
FSTDBYWAIT	0x0000-0A84	1	フラッシュ・スリープ・スタンバイ・ウェイト・レジスタ
FACTIVEWAIT	0x0000-0A85	1	フラッシュ・スタンバイ・アクティブ・ウェイト・レジスタ
FBANKWAIT	0x0000-0A86	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	0x0000-0A87	1	OTP リード・アクセス・ウェイト・ステート・レジスタ

1) これらのレジスタは、EALLOW 保護されています。

注：フラッシュ・メモリまたは OTP メモリにアクセスしている間は、フラッシュ・コンフィギュレーション・レジスタにアクセスしないでください。

OTP メモリまたはフラッシュ・メモリから実行中のコードにより、またはアクセスが進行中である可能性のある間は、フラッシュ・コンフィギュレーション・レジスタにアクセスしないでください。フラッシュ・レジスタに対するすべてのレジスタ・アクセスは、フラッシュ /OTP メモリ外部で実行中の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アクセスしようとししないでください。これに対する保護を行うハードウェアはありません。

フラッシュ /OTP で実行中の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せんが、レジスタにライトしないでください。

レジスタに対する CPU ライト・アクセスは、EALLOW 命令の実行によってのみネーブルにできます。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。これにより、レジスタの誤アクセスから保護します。リード・アクセスは常に可能です。レジスタには JTAG ポートからアクセスでき、EALLOW を実行する必要はありません。これらのレジスタは、16 ビットおよび 32 ビット・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 (FOPT) レジスタ

15	1	0
予約済み		ENPIPE
R-0		R/W-0

凡例: R = リード・アクセス、-0 = リセット後の値

注: EALLOW 保護されたレジスタ

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

ビット	フィールド	説明
15-1	予約済み	
0	ENPIPE	<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する: このビットが設定されている場合、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、命令をプリフェッチすることによって命令フェッチのパフォーマンスを向上させます。</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を与えます。ROM デバイスでは、このビットは、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を与えます。</p> <p>パイプライン・モードがイネーブルの場合、フラッシュ・ウェイト・ステート (ページおよびランダム) は、ゼロより大きい値である必要があります。</p>

図 1-5 フラッシュ・パワー・レジスタ (FPWR)

15	2	1	0
予約済み			PWR
R-0			R/W-0

凡例: R = リード・アクセス、-0 = リセット後の値

注: EALLOW 保護されたレジスタ

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	説明
15-2	予約済み	
1-0	PWR	<p>フラッシュ・パワー・モード・ビット：これらのビットにライトすると、フラッシュ・バンクおよびポンプの現行の電力モードが変更されます。フラッシュ・バンクの電力モードの変更に関する詳細については、セクション 1.2 を参照してください。</p> <p>ROM デバイスでは、PWR を変更しても、ROM の電力消費には何の影響もありません。スタンバイまたはスリープ・モードへの移行は、フラッシュ・デバイスと同様に、ROM からの次のアクセスを遅らせます。</p> <p>00 ポンプおよびバンクはスリープ状態（最低電力）</p> <p>01 ポンプおよびバンクはスタンバイ状態</p> <p>10 予約済み（影響なし）</p> <p>11 ポンプおよびバンクはアクティブ状態（最高電力）</p>

図 1-6 フラッシュ・ステータス・レジスタ (FSTATUS)

15					9	8
予約済み					3VSTAT	
R-0					R/W1C-0	
7	4	3	2	1	0	
予約済み		ACTIVE-WAITS	STDBY-WAITS	PWRS		
R-0		R-0	R-0	R-0		

凡例： R = リード・アクセス、-0 = リセット後の値、W1C = クリアするには 1 をライトする

注： EALLOW 保護されたレジスタ

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	説明
15-9	予約済み	
8	3VSTAT	<p>V_{DD3V} ステータス・ラッチ・ビット：設定した場合、このビットは、ポンプ・モジュールからの 3VSTAT 信号が高レベルになったことを示します。この信号は、3V の供給電圧が許容範囲を超えたことを示します。このビットは、1 をライトすることによってクリアされます。0 のライトは無視されます。</p>
7-4	予約済み	
3	ACTIVewaITS	<p>バンクおよびポンプのスタンバイ・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のタイムアウトをカウントしているかどうかを示します。このビットがセットされている場合、カウンタはカウントしています。このビットが 0 の場合、カウンタはカウントしていません。</p>
2	STDBYwaITS	<p>バンクおよびポンプのスリープ・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のタイムアウトをカウントしているかどうかを示します。このビットがセットされている場合、カウンタはカウントしています。このビットが 0 の場合、カウンタはカウントしていません。</p>

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 (続き)

ビット	フィールド	説明
1-0	PWRS	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 /OPT の現在の電力モードを示します。 00 ポンプおよびバンクはスリープ状態 (最低電力) 01 ポンプおよびバンクはスタンバイ状態 10 予約済み 11 ポンプおよびバンクはアクティブ状態 (最高電力) 注： 上記ビットは、該当する遅延が期限切れになった場合のみ新しい電力モードに設定されます。

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

15	9	8	0
予約済み		STDBYWAIT	
R-0		R/W-1	

凡例： R = リード・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

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

ビット	フィールド	説明
15-9	予約済み	
8-0	STDBYWAIT	バンクおよびポンプのスリープ・スタンバイ・ウェイト・カウント。バンクおよびポンプ・モジュールがスリープ・モードのときに、FPWR レジスタの PWR ビットに対してライトが実行されるか (より高いデフォルトの電力モードに変更するため)、フラッシュ・バンクまたは OTP に対して CPU リードまたはフェッチ・アクセスが実行されると、これらのレジスタ・ビットに指定された値でカウンタが開始されます。バンクおよびポンプの電力モードは、スタンバイ・モードへと向かいます。その後、カウンタがゼロまでカウントダウンしてから、PWRS ビットがスタンバイ・モードに設定されます。フラッシュ・バンク /OPT に対する CPU リードまたはフェッチ・アクセスが処理を開始していた場合、アクセスが完了するまで CPU はストールします (ACTIVEWAIT ビットを参照)。STDBYWAIT ビットは、遅延の CPU クロック・サイクル数 (0..511 SYSCLKOUT サイクル) を指定します。詳細については、「フラッシュおよび OTP の電力モード」を参照してください。 このレジスタは、デフォルトの状態のままにしてください。

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 (FACTIVEWAIT) レジスタ

15	9	8	0
予約済み		ACTIVEWAIT	
R-0		R/W-1	

凡例： R = リード・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

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 (FACTIVEWAIT) レジスタのフィールドの説明

ビット	フィールド	説明
15-9	予約済み	
8-0	ACTIVEWAIT	バンクおよびポンプのスタンバイ・アクティブ・ウェイト・カウント：バンクおよびポンプ・モジュールがスタンバイ・モードのときに、FPWR レジスタの PWR ビットに対してライトが実行されるか（より高いデフォルトの電力モードに変更するため）、フラッシュ・バンクに対して CPU リードまたはフェッチ・アクセスが実行されると、これらのレジスタ・ビットに指定された値でカウンタが開始されます。バンクおよびポンプの電力モードは、アクティブ・モードへと向かいます。その後、カウンタがゼロまでカウントダウンしてから、CPU アクセスを進行させることができます。フラッシュ・バンクに対する CPU リードまたはフェッチ・アクセスが処理を開始していた場合、アクセスが完了するまで CPU はストールします (PAGEWAIT および RANDWAIT ビットを参照)。ACTIVEWAIT ビットは、遅延の CPU クロック・サイクル数 (0...511 SYSCLKOUT サイクル) を指定します。詳細については、「フラッシュおよび OTP の電力モード」を参照してください。 このレジスタは、デフォルトの状態のままにしてください。

図 1-9 フラッシュ・ウェイト・ステート (FBANKWAIT) レジスタ

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

凡例： R = リード・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

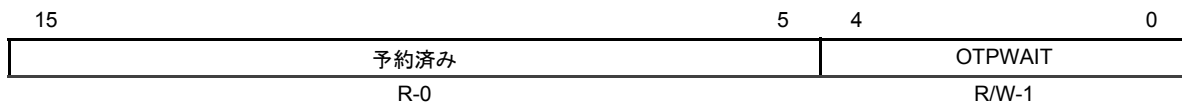
表 1-7 フラッシュ・ウェイト・ステート (FBANKWAIT) レジスタのフィールドの説明

ビット	フィールド	説明
15-12	予約済み	
11-8	PAGEWAIT	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、CPU クロック・サイクル (0...15 SYSCLKOUT サイクル) で、バンクに対するページ・リード・オペレーションのウェイト・ステート数を指定します。注 1 および 2 を参照してください。 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・ステートに影響を与えません。
7-4	予約済み	

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

ビット	フィールド	説明
3-0	RANDWAIT	<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、CPU クロック・サイクル (0...15 SYSCLKOUT サイクル) 内でのバンクに対するランダム・リード・オペレーションのウェイト・ステート数を指定します。注 1 および 2 を参照してください。RANDWAIT は、0 より大きい値に設定する必要があります。すなわち、少なくとも 1 個のランダム・ウェイト・ステートを使用する必要があります。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、各デバイスのデータシートを参照してください。</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・ステートに影響を与えます。</p> <p>注： 1) RANDWAIT は、PAGEWAIT の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 値を検出するハードウェアは提供されていません。</p> <p>2) フラッシュ・パイプライン・モードがイネーブルの場合、PAGEWAIT および RANDWAIT をゼロより大きい値に設定する必要があります。</p>

図 1-10 OTP ウェイト・ステート (FOTPWAIT) レジスタ



凡例： R = リード・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表 1-8 OTP ウェイト・ステート (FOTPWAIT) レジスタ

ビット	フィールド	説明
15-5	予約済み	
4-0	OTPWAIT	<p>OTP リード・ウェイト・ステート。これらのレジスタ・ビットは、CPU クロック・サイクル (0...31 SYSCLKOUT サイクル) で、OTP に対するリード・オペレーションのウェイト・ステート数を指定します。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」を参照してください。</p> <p>OTP には、ページ・モードはありません。</p> <p>OTPWAIT は、0 より大きい値に設定する必要があります。すなわち、最低 1 個のウェイト・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、デバイス固有のデータシートを参照してください。</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・ステートに影響を与えます。</p>

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 デバイスに組み込まれたセキュリティ機能です。CSM は、オンチップ・メモリに対するアクセス/可視性を無許可のユーザーから保護します。すなわち、保護されているコードを複製/リバース・エンジニアリングから保護します。

セキュアという用語は、オンチップ・メモリへのアクセスが保護されていることを意味します。アンセキュアという用語は、オンチップ・セキュア・メモリへのアクセスが保護されていないことを意味します。すなわち、メモリの内容を何らかの手段で (たとえば、Code Composer Studio™ などのデバッグ・ツールで) リードすることが可能です。

項目	ページ
2.1 機能の説明.....	2-2
2.2 CSM によるほかのオンチップ・リソースへの影響	2-4
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.....	2-6
2.4 セキュリティ・ロジックを保護するための必要事項と禁止事項	2-13
2.5 CSM の機能 - 要約.....	2-14

2.1 機能の説明

セキュリティ・モジュールは、特定のオンチップ・メモリに対する CPU アクセスを制限します。これにより、実質的に JTAG ポートまたは外部ペリフェラルによるさまざまなメモリへのリードおよびライト・アクセスがブロックされます。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ており、ユーザーのコードまたはデータを無許可コピーから保護します。

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場合、デバイスはセキュアです。セキュアな場合、プログラム・カウンタが現在どこを指しているかによって、2つのレベルの保護が可能です。コードがセキュア・メモリ内部から実行されている場合、JTAG（すなわち、エミュレータ）によるアクセスのみがブロックされます。この場合、セキュア・コードによるセキュア・データへのアクセスは許可されます。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、セキュア・メモリに対するすべてのアクセスはブロックされます。ユーザー・コードは、動的にセキュア・メモリにジャンプまたはセキュア・メモリからジャンプできるため、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、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る場合でも、割り込みサービス・ルーチンをセキュア・メモリに置くことができます。

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・データ（8 個の 16 ビット・ワード）のパスワードによって保護されます。

デバイスは、この章で後に説明するパスワード・マッチ・フロー（PMF）を実行すること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ています。

表 2-1 セキュリティ・レベル

PMF が正しい パスワードで実行 されたか？	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 外部	セキュア・メモリに対するフェッチのみが許可されます。
いいえ	セキュア	セキュア・メモリ 内部	CPU はフル・アクセスを持ちます。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	セキュア でない	任意	CPU および JTAG ポートは、セキュア・メモリへのフル・アクセスを持ちます。

パスワードは、フラッシュ/ROM メモリのコード・セキュリティ・パスワード・ロケーション（PWL）にストアされます（0x003F 7FF8 ~ 0x003F 7FFF）。これらのロケーションには、システム設計者によって事前定義されたパスワードがストアされます。

フラッシュ・デバイスでは、古いパスワードがわかっている場合、いつでもパスワードを変更できます。ROM デバイスでは、TI によるデバイスの製造後にパスワードを変更することはできません。

PWL の 128 ビットすべてが 1 である場合、デバイスはアンセキュアになります。新しいフラッシュ・デバイスはイレーズされたフラッシュ（すべて 1）なので、このデバイスをアンセキュア・モードにするには、PWL のリードのみが必要です。PWL の 128 ビットすべてが 0 である場合、キー・レジスタの内容に関わらず、デバイスはセキュアです。すべてゼロのパスワードを使用したり、フラッシュに対してクリア・ルーチンを実行した後にデバイスをリセットしないでください。PWL がすべてゼロのときにデバイスをリセットすると、デバイスのデバッグまたは再プログラミングができなくなります。要約すると、イレーズされたフラッシュ配列を持つデバイスはアンセキュアです。クリアされたフラッシュ配列を持つデバイスはセキュアです。

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、ユーザーがアクセス可能なレジスタ（8 個の 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、メモリ空間のアドレス 0x0000 0AE0 ~ 0x0000 0AE7 にマップされ、EALLOW 保護されます。

注：アドレス 0x3F7F80 ~ 0x3F7FF5 のセキュリティ

コード・セキュリティ・オペレーションでは、0x3F7F80 ~ 0x3F7FF5 のすべてのアドレスをプログラム・コードまたはデータとして使用することはできませんが、コード・セキュリティ・パスワードをプログラムするときは、これらのアドレスを 0x0000 にプログラムする必要があります。セキュリティが重要でない場合は、これらのアドレスをコードまたはデータに使用できます。

コード・セキュリティ・モジュールに関する免責事項

このデバイスに含まれるコード・セキュリティ・モジュール（以下「CSM」）は、関連するメモリ（ROM またはフラッシュ）にストアまたは保存されたデータをパスワード保護するために設計されました。テキサス・インスツルメンツ（以下「TI」）は、このデバイスに適用される保証期間に限り、CSM が、TI の標準契約約款に従い TI が発行した仕様書に適合していることを保証します。

上記にも関わらず TI は、CSM の支障をきたさないことまたはパスワード保護を破られないこと、あるいは関連するメモリにストアまたは保存されたデータにほかの手段でアクセスし得ないことを保証または表明するものではありません。さらに、上記に規定される場合を除き、TI は、CSM またはこのデバイスの動作に関し、商品性または特定の目的に対する適合性を黙示的な保証を含め、何ら保証または表明を行うものではありません。

TI は、いかなる場合も、お客様が CSM またはこのデバイスを使用することによって生じる結果的、特別的、間接的、偶発的、または懲罰的な損害に関してたとえ、そのような損害の可能性を示されていた場合であっても何ら責任を負うものではありません。除外される損害には、データの喪失、得意先の喪失、使用の喪失または事業の中断、あるいはその他の経済的損失が含まれますが、これらに限定されません。

2.2 CSM によるほかのオンチップ・リソースへの影響

CSM は、以下のオンチップ・リソースに対して何の影響も与えません。

- 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) - デバイスがセキュア・モードであるかアンセキュア・モードであるかに関わらず、これらのメモリ・ブロックには自由にアクセスでき、これらのブロックからコードを実行することができます。
- 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 - デバイスがセキュア・モードであるかアンセキュア・モードであるかに関わらず、ペリフェラル・レジスタは、オンチップまたはオフチップ・メモリから実行中のコードによって初期化できます。
- PIE ベクタ・テーブル - デバイスがセキュア・モードであるかアンセキュア・モードであるかに関わらず、ベクタ・テーブルはリードおよびライトできます。表 2-2 および表 2-3 は、281x デバイスの CSM の影響を受ける（または影響を受けない）オンチップ・デバイスを示しています。その他のデバイスについては、各デバイスのデータシートを参照してください。
- フラッシュ設定レジスタ - フラッシュと OTP のウェイト・ステートを設定するために使用されるレジスタは、CSM によって保護されます。C281x デバイスにおいては、これらのレジスタが ROM のウェイト・ステートを決定します。これらのレジスタを変更するために CSM を解除、もしくはそれらにアクセスするコードはセキュアなメモリから実行しなければいけません。

表 2-2 CSM の影響を受ける F281x/C281x リソース

アドレス	ブロック
0x0000 8000-0x0000 8FFF	L0 SARAM (4K X 16) ⁽¹⁾
0x0000 9000-0x0000 9FFF	L1 SARAM (4K X 16) ⁽¹⁾
0x003D 7800- 0x003D 7BFF	ワンタイム・プログラマブル(OTP)またはROM(1K × 16)
0x003D 8000-0x003F 7FFF	フラッシュまたは ROM (128K × 16 または 64K × 16)

1) R281x RAM デバイスでは、L0 および L1 SARAM は CSM の影響を受けます。ただし、CSM のパスワードはすべて 1 に設定されるため、CSM のブロック解除は容易に行えます。

表 2-3 CSM の影響を受けない 281x リソース

アドレス	ブロック
0x0000 0000-0x0000 03FF	M0 SARAM (1K X 16)
0x0000 0400-0x0000 07FF	M1 SARAM (1K X 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 2000-0x0000 3FFF	XINTF ゾーン 0
0x0000 4000-0x0000 5FFF	XINTF ゾーン 1
0x0000 6000-0x0000 6FFF	ペリフェラル・フレーム 1 (4K × 16)
0x0000 7000-0x0000 7FFF	ペリフェラル・フレーム 2 (4K × 16)
0x0008 0000-0x000F FFFF	XINTF ゾーン 2
0x0010 0000-0x0017 FFFF	XINTF ゾーン 6
0x003F 8000-0x003F 9FFF	H0 SARAM (8K X 16)
0x003F 0000-0x003F FFFF	XINTF ゾーン 7
0x003F F000-0x003F FFFF	ブート ROM (4K × 16)

要約すると、CSM の影響を受けることなく、表 2-3 の保護されていないオンチップ・プログラム RAM に JTAG コネクタを通してコードをロードすることが可能です。デバイスがセキュア・モードであるかアンセキュア・モードであるかに関わらず、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

通常、プロジェクトの開発時はコード・セキュリティは必要ありませんが、コード開発が終わった後はセキュリティが必要になります。このようなコードをフラッシュ・メモリにプログラム（またはROM化）する前に、デバイスをセキュアにするためのパスワードを決める必要があります。パスワードを設定すると、デバイスはセキュアになります（すなわち、適切なアドレスにパスワードをプログラムし、デバイスのリセットを実行するか FORCESEC ビット（CSMSCR.15）を設定することにより、デバイスをセキュアにする）。それ以降、何らかの手段（JTAG、外部 / オンチップ・メモリから実行されるコードなど）でセキュア・メモリの内容をデバッグするためにアクセスする場合には、正しいパスワードを入力する必要があります。セキュア・メモリからコードを実行するために（エンドユーザーによる通常の使用など）パスワードは必要ありませんが、デバッグの目的でセキュア・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 - ユーザーがアクセス可能			
0x0000-0AE0	KEY0 (注 1)	0xFFFF	128 ビット・キー・レジスタの下位ワード
0x0000-0AE1	KEY1 (注 1)	0xFFFF	128 ビット・キー・レジスタの 2 番目のワード
0x0000-0AE2	KEY2 (注 1)	0xFFFF	128 ビット・キー・レジスタの 3 番目のワード
0x0000-0AE3	KEY3 (注 1)	0xFFFF	128 ビット・キーの 4 番目のワード
0x0000-0AE4	KEY4 (注 1)	0xFFFF	128 ビット・キーの 5 番目のワード
0x0000-0AE5	KEY5 (注 1)	0xFFFF	128 ビット・キーの 6 番目のワード
0x0000-0AE6	KEY6 (注 1)	0xFFFF	128 ビット・キーの 7 番目のワード
0x0000-0AE7	KEY7 (注 1)	0xFFFF	128 ビット・キー・レジスタの上位ワード
0x0000-0AEF	CSMSCR (注 1)		CSM ステータスおよび制御レジスタ
メモリの PWL - パスワード用に予約済み (R281x デバイスでは、すべてのパスワードはすべて F に設定される)			
0x003F-7FF8	PWL0	ユーザー定義	128 ビット・パスワードの下位ワード
0x003F-7FF9	PWL1	ユーザー定義	128 ビット・パスワードの 2 番目のワード
0x003F-7FFA	PWL2	ユーザー定義	128 ビット・パスワードの 3 番目のワード
0x003F-7FFB	PWL3	ユーザー定義	128 ビット・パスワードの 4 番目のワード
0x003F-7FFC	PWL4	ユーザー定義	128 ビット・パスワードの 5 番目のワード
0x003F-7FFD	PWL5	ユーザー定義	128 ビット・パスワードの 6 番目のワード
0x003F-7FFE	PWL6	ユーザー定義	128 ビット・パスワードの 7 番目のワード
0x003F-7FFF	PWL7	ユーザー定義	128 ビット・パスワードの上位ワード

注 1 : EALLOW 保護

図 2-1 CSM ステータスおよび制御 (CSMSCR) レジスタ

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

凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

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

ビット	フィールド	説明
15	FORCESEC	1 をライトすると、キー・レジスタがクリアされ、デバイスがセキュアになります。リードは、常にゼロを返します。
0	SECURE	デバイスのセキュリティ状態を示すリード専用ビット。 1 デバイスはセキュアです (CSM はロックされている)。 0 デバイスはアンセキュアです (CSM はロック解除されている)。

2.3.1 セキュリティのロック解除が必要な環境

以下は、アンセキュアにすることが必要にな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、製品の設計時で最も一般的な環境です。
- TI のフラッシュ・ユーティリティを使用したフラッシュのプログラミング

フラッシュのプログラミングは、コードの開発およびテストにおいて一般的に行われます。ユーザーが必要なパスワードを設定すると、フラッシュ・ユーティリティは、フラッシュをプログラムす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。新しいデバイスはフラッシュが消去されているため、フラッシュ・ユーティリティは、パスワード認証なしで新しいデバイスのコード・セキュリティ・ロジックをディスエーブルにできます。ただし、デバイス (すでにユーザーのパスワードが設定されているデバイス) を再プログラムする場合は、プログラミングをイネーブルにするために、フラッシュ・ユーティリティにパスワードを設定する必要があります。

- アプリケーションによる特定環境

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

- フラッシュをプログラムするためのオンチップ・ブートローダの使用
- 外部メモリまたはオンチップ・アンセキュア・メモリからコードを実行するため、ルックアップ・テーブル用のセキュア・メモリへのアクセスが必要。外部コードからパスワードを設定すると、コード・セキュリティが損なわれる可能性があるため、この動作状態は推奨されません。

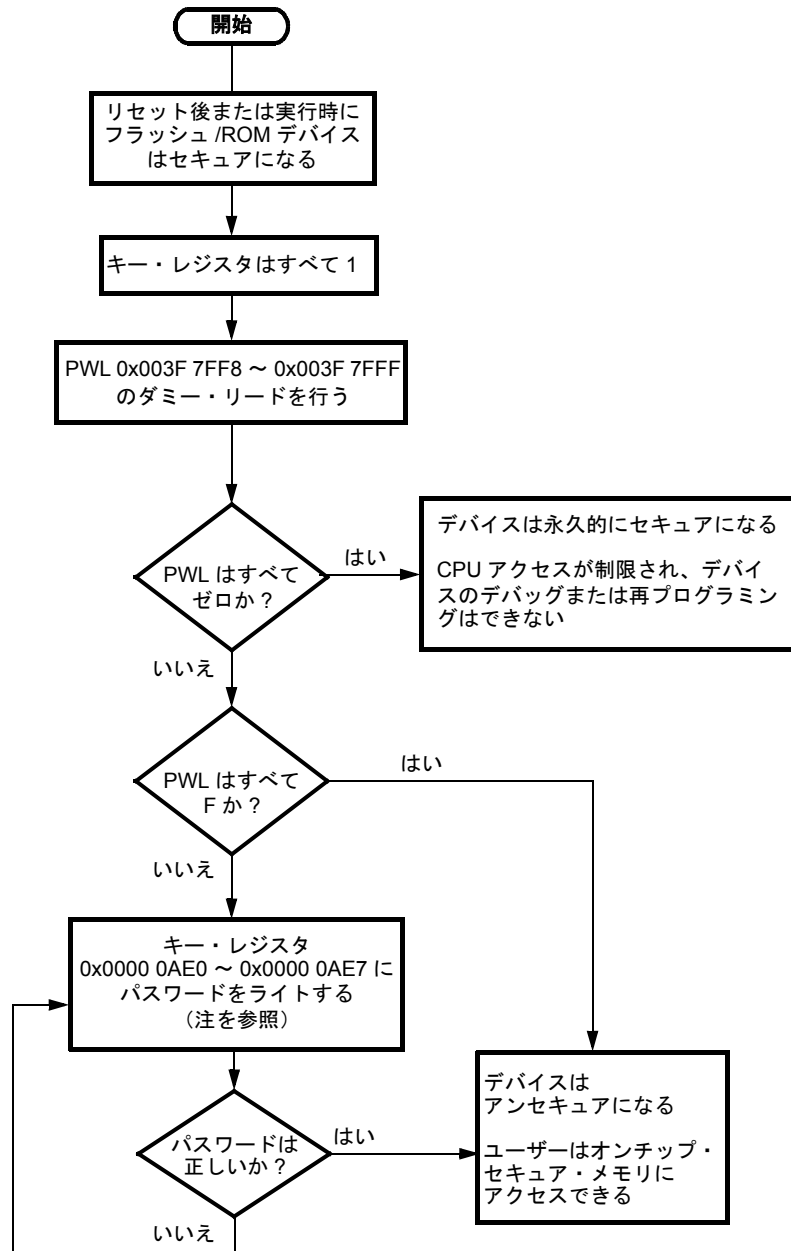
アンセキュアにする手順は、上記のすべての状況で同じです。この手順はパスワード・マッチ・フロー（PMF）と呼ばれます。図 2-2 は、ユーザーがデバイスをアンセキュアにしようとするたびに必要な操作の手順を説明しています。また、明確にするために、コード例を示します。

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、基本的にパスワード・ロケーション（PWL）からの 8 回のダミー・リードと、それに続くキー・レジスタへの 8 回のライトです。

図 2-2 は、セキュリティ・ロジック・レジスタを初期化し、セキュリティ・ロジックをディスエーブルにする上で PMF がどのように機能するかを示しています。

図 2-2 パスワード・マッチ・フロー (PMF)



注： キー・レジスタは EALLOW 保護されます。

2.3.3 コード・セキュリティを持つ/持たないデバイスをアンセキュアにする際の考慮事項

ケース 1 およびケース 2 に、コード・セキュリティが設定されている、またはされていないデバイスをアンセキュアにする際の考慮事項を挙げます。

ケース 1: コード・セキュリティが設定されているデバイス

コード・セキュリティが設定されているデバイスは、既にあるパスワードが PWL (メモリ内のロケーション 0x003F 7FF8 ~ 0x003F 7FFF) にストアされています。また、ロケーション 0x3F7F80 ~ 0x3F7FF5 をすべて 0x0000 にプログラムする必要があり、プログラムやデータをストアするためにこれらのロケーションを使用してはなりません。以下は、このデバイスをアンセキュアにするためのステップです。

- 1) PWL 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、デバイスはアンセキュアになります。それ以外の場合、セキュアのままになります。

ケース 2: コード・セキュリティが設定されていないデバイス

コード・セキュリティが設定されていないデバイスは、0x FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF (すべて 1 の 128 ビット) が PWL にストアされています。以下は、このデバイスを使用するためのステップです。

- 4) PWL のダミー・リードを実行します。
- 5) このオペレーションの完了後、すぐにセキュア・メモリへのフル・アクセスが可能になります。

注: デバイスがパスワードで保護されていない場合も、セキュア・メモリのリード、ライト、またはプログラミング前に、ダミー・リード・オペレーションを実行する必要があります。

2.3.4 アンセキュアにする C コード例

```

volatile int *CSM = (volatile int *)0x000AE0;    //CSM レジスタ・ファイル
volatile int *PWL = (volatile int *)0x3F7FF8;    // パスワード・ロケーション
volatile int tmp;
int i ;
// アドレス 0x3F7FF8 ~ 0x3F7FFF のフラッシュ /ROM にある
// 128 ビットのパスワード・ロケーション (PWL) をリードする。
// デバイスがセキュアな場合、リードされる値は実際には
// tmp 変数にロードされない。
// そのため、このリードは「ダミー・リード」と呼ばれる。
for (i = 0; i<8; i++) tmp = *PWL++;

// パスワード・ロケーション (PWL) がすべて 1 (0xFFFF) の場合、
// デバイスはアンセキュアになる。パスワードがすべて
// 1 (0xFFFF) でない場合、CSM をアンセキュアにするには
// 以下のコードが必要である。
// 128 ビットのパスワードをキー・レジスタにライトする。
// このパスワードが PWL にストアされているパスワードと
// マッチする場合、CSM はアンセキュアになる。
// マッチしない場合は、セキュアな状態が維持される。
// パスワードの例 :
// 0x0123456789ABCDEF89AB45670123
asm(" EALLOW");    // キー・レジスタは EALLOW 保護されている
*CSM++ = 0x0123;    // 0xAE0 の場合、KEY0
*CSM++ = 0x4567;    // 0xAE1 の場合、KEY1
*CSM++ = 0x89AB;    // 0xAE2 の場合、KEY2
*CSM++ = 0xCDEF;    // 0xAE3 の場合、KEY3
*CSM++ = 0xCDEF;    // 0xAE4 の場合、KEY4
*CSM++ = 0x89AB;    // 0xAE5 の場合、KEY5
*CSM++ = 0x4567;    // 0xAE6 の場合、KEY6
*CSM++ = 0x0123;    // 0xAE7 の場合、KEY7
asm(" EDIS");

```

2.3.5 再セキュアにする C コード例

```
volatile int *CSM = 0x000AE0;          //CSM レジスタ・ファイル
//FORCESEC ビットを設定
asm("  EALLOW");                      //CSMSCR レジスタは EALLOW 保護されている
*CSM = 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と禁止事項

2.4.1 必要事項

- デバック時およびコード開発時を簡単にするために、デバイスをアンセキュア・モードで使用してください。すなわち、PWL ワードとしてすべて 1 の 128 ビットを使用します（または、覚えやすいパスワードを使用）。コードが確定し、開発が終了したら、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、PWL のパスワードを再度チェックしてください。
- コード実行フローでは、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・メモリ間で自由に切り替えることができます。デバイスがセキュアな場合にセキュア・メモリに置かれたデータ変数にアクセスするには、コードがセキュア・メモリから実行されている必要があります。
- CSMを使用するときは、ロケーション0x3F7F80～0x3F7FF5を0x0000にプログラムしてください。

2.4.2 禁止事項

- コード・セキュリティが必要な場合、アプリケーション内の PWL 以外の場所にパスワードを埋め込まないでください。セキュリティが損なわれる可能性があります。
- パスワードとしてすべてゼロの 128 ビットを使用しないでください。キー・レジスタの内容に関わらず、デバイスが自動的にセキュアになります。この場合、デバイスのデバッグや再プログラミングができません。
- フラッシュのクリア後、イレーズする前にリセットしないでください。PWL がゼロのままになり、キー・レジスタの内容に関わらず、デバイスが自動的にセキュアになります。この場合、デバイスのデバッグや再プログラミングができません。
- プログラムやデータをストアするためにロケーション0x3F7F80～0x3F7FF5を使用しないでください。CSMを使用するときは、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 の機能 - 要約

- 1) リセット後、パスワード・マッチ・フロー (PMF) が実行されるまで、フラッシュはセキュアになります。
- 2) フラッシュまたは ROM から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 デバイスでは、プログラムはデバイスの製造時にハードコードされる)、マイクロコンピュータ・モードで DSP に電源を投入することです。CSM の状態に関わらず、セキュア・メモリからは常に命令フェッチが可能であるため、PMF を実行しなくても、コードは正常に機能します。
- 3) デバイスがセキュアな場合、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な場合、アンセキュア・メモリから実行中のコードからセキュア・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、デバッガ (Code Composer Studio™ など) によってセキュア・メモリを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な場合、CPU コードおよびデバッガの両方からセキュア・メモリへの完全なアクセスが与えられます。

クロック

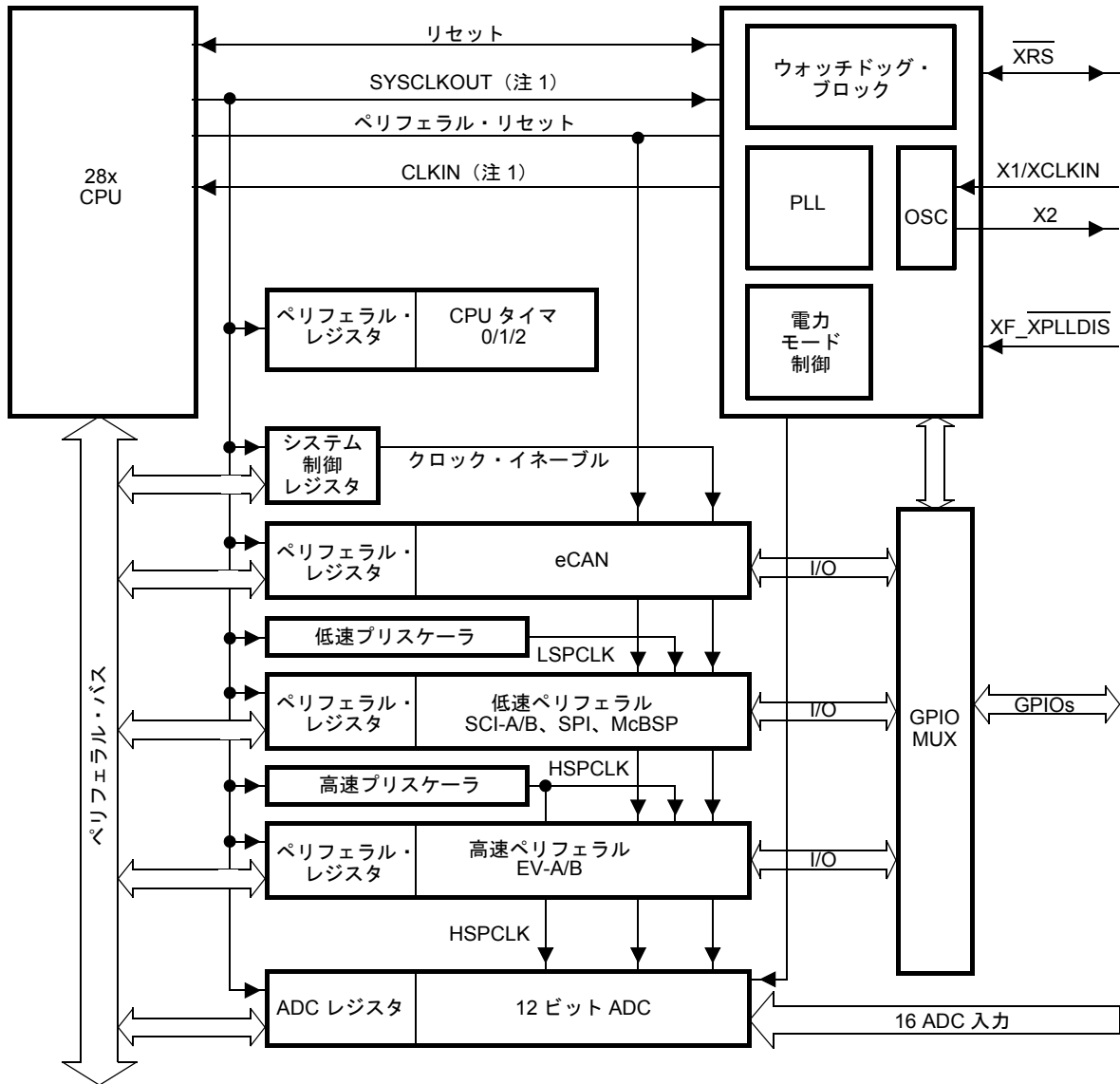
このセクションでは、281x のオシレータ、PLL およびクロックのメカニズム、ウォッチドッグ機能、低電力モードについて説明します。

項目	ページ
3.1 クロックとシステム制御	3-2
3.2 OSC と PLL ブロック	3-9
3.3 低電力モード・ブロック	3-12
3.4 ウォッチドッグ・ブロック	3-15
3.5 32 ビット CPU タイマ 0/1/2	3-19

3.1 クロックとシステム制御

図 3-1 は、281x デバイスのさまざまなクロックおよびリセット・ドメインを示しています。

図3-1 クロックおよびリセット・ドメイン



注 1: CLKIN は CPU へのクロックです。SYSCLKOUT として CPU から渡されます (すなわち、CLKIN の周波数は SYSCLKOUT と同じ)。

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 に挙げるレジスタによって制御されます。

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 (注1)

名前	アドレス	サイズ (× 16)	説明
予約済み	0x0000 7010 0x0000 7019	10	
HISPCP	0x0000 701A	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	0x0000 701B	1	LSPCLK 用の低速ペリフェラル・クロック・プリスケアラ・レジスタ
PCLKCR	0x0000 701C	1	ペリフェラル・クロック制御レジスタ
予約済み	0x0000 701D	1	
LPMCR0	0x0000 701E	1	低電力モード制御レジスタ 0
LPMCR1	0x0000 701F	1	低電力モード制御レジスタ 1
予約済み	0x0000 7020	1	
PLLCR	0x0000 7021	1	PLL 制御レジスタ (注2)
SCSR	0x0000 7022	1	システム制御およびステータス・レジスタ
WDCNTR	0x0000 7023	1	ウォッチドッグ・カウンタ・レジスタ
予約済み	0x0000 7024	1	
WDKEY	0x0000 7025	1	ウォッチドッグ・リセット・キー・レジスタ
予約済み	0x0000 7026 0x0000 7028	3	
WDCR	0x0000 7029	1	ウォッチドッグ制御レジスタ
予約済み	0x0000 702A 0x0000 702F	6	

注 1: この表に示されたすべてのレジスタには、EALLOW 命令を実行することによってのみアクセスできます。

注 2: PLL 制御レジスタ (PLLCR) は、XRS 信号によってのみ既知の状態にリセットされます。

PCLKCR レジスタは、281x デバイスのさまざまなペリフェラル・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。図 3-2 は、PCLKCR レジスタのビットの説明です。

図 3-2 ペリフェラル・クロック制御 (PCLKCR) レジスタ

15	14	13	12	11	10	9	8	
予約済み	ECANENCLK	予約済み	MCBSPENCLK	SCIBENCLK	SCIAENCLK	予約済み	SPIENCLK	
R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	
7	予約済み			4	3	2	1	0
R-0				ADCENCLK	予約済み	EVBENCLK	EVAENCLK	
R-0				R/W-0	R-0	R/W-0	R/W-0	

凡例： R = リード・アクセス、-0 = リセット後の値

注： 1) EALLOW 保護されたレジスタ

注： 2) ペリフェラル・ブロックを使用しない場合、電力消費を最小限にするために、そのペリフェラルへのクロックをオフにできます。

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

ビット	フィールド	説明
15	予約済み	予約済み
14	ECANENCLK	ECANENCLK が設定されている場合、CAN ペリフェラル内のシステム・クロックがイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
13	予約済み	予約済み
12	MCBSPENCLK	MCBSPENCLK が設定されている場合、McBSP ペリフェラル内の低速クロック (LSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
11	SCIBENCLK	SCIBENCLK が設定されている場合、SCI-B ペリフェラル内の低速クロック (LSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
10	SCIAENCLK	SCIAENCLK が設定されている場合、SCI-A ペリフェラル内の低速クロック (LSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
9	予約済み	予約済み
8	SPIAENCLK	SPIAENCLK が設定されている場合、SPI ペリフェラル内の低速クロック (LSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
7-4	予約済み	予約済み
3	ADCENCLK	ADCENCLK が設定されている場合、ADC ペリフェラル内の高速クロック (HSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
2	予約済み	予約済み

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

ビット	フィールド	説明
1	EVBENCLK	EVBENCLK が設定されている場合、EV-B ペリフェラル内の高速クロック (HSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。
0	EVAENCLK	EVAENCLK が設定されている場合、EV-A ペリフェラル内の高速クロック (HSPCLK) がイネーブルになります。低電力動作では、このビットは、ユーザーまたはリセットによってゼロに設定されます。

システム制御およびステータス・レジスタには、ウォッチドッグ・オーバーライド・ビットおよび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。表 3-3 は、SCSR レジスタのビット機能を示しています。

図 3-3 システム制御およびステータス (SCSR) レジスタ



凡例: R = リード・アクセス、-0 = リセット後の値、W1C = クリアするには 1 をライトする

注: EALLOW 保護されたレジスタ

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

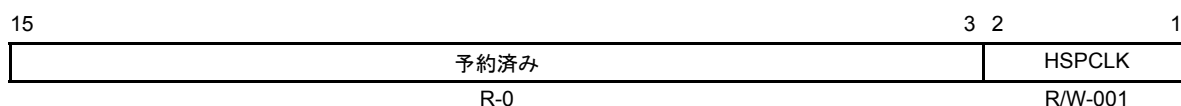
ビット	フィールド	説明
15-3	予約済み	
2	WDINTS	ウォッチドッグ割り込みステータス・ビット。WDINTS には、ウォッチドッグ・ブロックからの WDINT 信号の現行状態が反映されます。デバイスをアイドルまたはスタンバイからウェイクするためにウォッチドッグ割り込みを使用する場合、アイドルまたはスタンバイ・モードに戻す前に、このビットをチェックし、ウォッチドッグ割り込み信号がアクティブではない (WDINTS = 1) ことを確認する必要があります。

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

ビット	フィールド	説明
1	WDENINT	ウォッチドッグ・イネーブル割り込み。 0 $\overline{\text{WDRST}}$ 出力信号はイネーブル、 $\overline{\text{WDINT}}$ 出力信号はディスエーブルです。これは、リセット時 (XRS) のデフォルトの状態です。 1 ウォッチドッグ・リセット (WDRST) 出力信号はディスエーブル、ウォッチドッグ割り込み (WDINT) 出力信号はイネーブルです。
0	WDOVERRIDE	WDOVERRIDE が 1 に設定されている場合、ユーザーは、ウォッチドッグ制御 (WDCR) レジスタのウォッチドッグ・ディスエーブル (WDDIS) ビットの状態を変更することができます。1 をライトすることによって WDOVERRIDE ビットがクリアされた場合、ユーザーが WDDIS ビットを変更することはできません。また、現在ディスエーブルになっているウォッチドッグをイネーブルにします。0 をライトしても何の影響もありません。このビットがクリアされている場合、リセットが行われるまでこの状態のままになります。このビットの現行状態は、ユーザーがリード可能です。

HISPCP および LOSPCP レジスタは、それぞれ高速ペリフェラル・クロックと低速ペリフェラル・クロックを設定するために使用されます。HISPCP ビットの定義については図 3-4 を、LOSPCP ビットの定義については図 3-4 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



凡例: R = リード・アクセス、-0 = リセット後の値

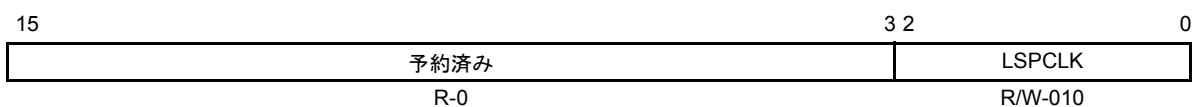
注: EALLOW 保護されたレジスタ

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

ビット	フィールド	値	説明
15-3	予約済み		
	HSPCLK		以下のビットは、SYSCLKOUT に対する高速ペリフェラル・クロック (HSPCLK) レートを設定します。 $HISPCP \neq 0$ の場合、 $HSPCLK = SYSCLKOUT / (HISPCP \text{ (注 1)} \times 2)$ $HISPCP = 0$ の場合、 $HSPCLK = SYSCLKOUT$
2-0		000	高速クロック = SYSCLKOUT/1
		001	高速クロック = SYSCLKOUT/2 (リセットのデフォルト)
		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

注 1: この等式の HISPCP は、HISPCP レジスタのビット 2:0 の値を示します。

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



凡例: R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注: EALLOW 保護されたレジスタ

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

ビット	フィールド	説明
15-3	予約済み	
	LSPCLK	以下のビットは、SYSCLKOUT に対する低速ペリフェラル・クロック (LSPCLK) レートを設定します。 $LOSPCP \neq 0$ の場合、 $LSPCLK = SYSCLKOUT / (LOSPCP \text{ (注 1)} \times 2)$ $LOSPCP = 0$ の場合、 $LSPCLK = SYSCLKOUT$
2-0		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 (リセットのデフォルト)

注 1: この等式の LOSPCP は、LOSPCP レジスタのビット 2:0 の値を示します。

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

ビット	フィールド	説明
		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

注 1: この等式の LOSPCP は、LOSPCP レジスタのビット 2:0 の値を示します。

3.2 OSC と 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、クロックをデバイスに供給し、低電力モード・エントリを制御します。

3.2.1 PLL ベースのクロック・モジュール

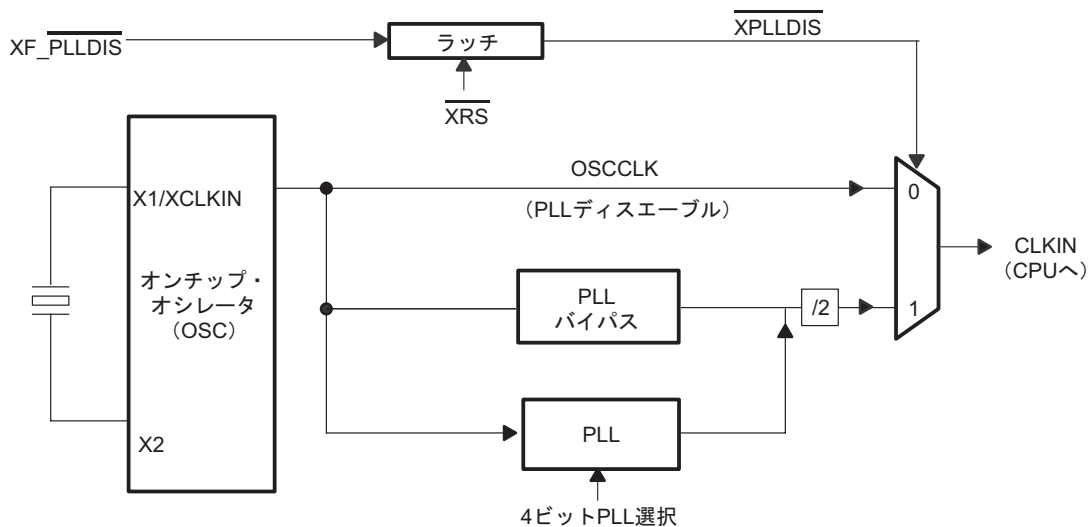
F281x デバイスは、オンチップの PLL ベースのクロック・モジュールを搭載しています。PLL は、異なる CPU クロック・レートを選択するための 4 ビットの通倍率制御を持ちます。

PLL ベースのクロック・モジュールには、以下の 2 つの動作モードがあります。

- 水晶動作
このモードでは、外部の水晶 / 発振子によってデバイスにクロックを供給できます。
- 外部クロック・ソース動作
このモードでは、内部のオシレータをバイパスできます。デバイス・クロックは、XTAL1/CLKIN ピンの外部クロック・ソース入力から生成されます。この場合、外部オシレータ・クロックは、XTAL1/CLKIN ピンに接続されます。

図 3-6 は、281x の OSC と PLL ブロックを示しています。

図 3-6 OSC と PLL ブロック



OSC 回路により、X1/XCLKIN ピンおよび X2 ピンを使用して水晶を 281x デバイスに接続することができます。水晶を使用しない場合、外部オシレータを直接 X1/XCLKIN ピンに接続し、X2 ピンは未接続のままにできます。

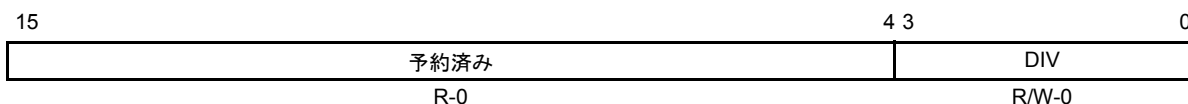
『TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 Digital Signal Processors Data Manual』(文書番号 SPRS174) および『TMS320R2811 and TMS320R2812 Digital Signal Processors』(文書番号 SPRS257)を参照してください。

表 3-6 可能な PLL 設定モード

PLL モード	備考	SYSCLKOUT
PLL ディスエーブル	リセット時に XPLLDIS ピンを Low にすることによって起動されます。PLL ブロックは完全にディスエーブルになります。CPU へのクロック入力 (CLKIN) は、X1/XCLKIN ピン上のクロック信号から直接供給します。	XCLKIN
PLL バイパス	PLL がディスエーブルでない場合の、電源投入時のデフォルトの PLL 設定。PLL 自体はバイパスされます。ただし、PLL ブロックの /2 モジュールは、CPU への供給前に、X1/XCLKIN ピンのクロック入力を 2 分周します。	XCLKIN/2
PLL イネーブル	ゼロ以外の値「n」を PLLCR レジスタにライトすることによって起動されます。PLL ブロックの /2 モジュールは、CPU への供給前に、PLL の出力を 2 分周します。	(XCLKIN*n)/2

PLLCR レジスタの DIV フィールド (ビット 3 ~ 0) は、デバイスの PLL 通倍器を変更するために使用されます。CPU が 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、新たに指定された周波数でロックすると、図 3-7 に示すように、PLL は CLKIN を新たな周波数に切り替えます。PLL が OSCCLK/2 から新たな周波数に切り替えるまでの時間は、131072 OSCCLK サイクルです。タイムクリティカルなソフトウェアの場合、PLL がロック・シーケンスを完了するために、PLLCR レジスタへのライト後に必要なロック期間のソフトウェア遅延を挿入する必要があります。

図 3-7 PLLCR レジスタ



凡例: R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注: EALLOW 保護されたレジスタ

表 3-7 PLLCR レジスタのフィールドの説明

ビット	名前	説明
15-4	予約済み	
3-0	DIV	DIV フィールドは、PLL をバイパスするかどうかを制御し、バイパスしない場合は PLL クロック通倍率を設定します。 0000 CLKIN = OSCCLK/2 (PLL バイパス) 0001 CLKIN = (OSCCLK*1.0)/2 0010 CLKIN = (OSCCLK*2.0)/2

注 1: PLLCR レジスタは、XRS リセット行によって既知の状態にリセットされます。リセットがデバッグによって開始された場合、PLL クロック通倍率は変更されません。

表 3-7 PLLCR レジスタのフィールドの説明 (続き)

ビット	名前	説明
	0011	$CLKIN = (OSCCLK * 3.0) / 2$
	0100	$CLKIN = (OSCCLK * 4.0) / 2$
	0101	$CLKIN = (OSCCLK * 5.0) / 2$
	0110	$CLKIN = (OSCCLK * 6.0) / 2$
	0111	$CLKIN = (OSCCLK * 7.0) / 2$
	1000	$CLKIN = (OSCCLK * 8.0) / 2$
	1001	$CLKIN = (OSCCLK * 9.0) / 2$
	1010	$CLKIN = (OSCCLK * 10.0) / 2$
	1011- 1111	予約済み

注 1: PLLCR レジスタは、 \overline{XRS} リセット行によって既知の状態にリセットされます。リセットがデバッグによって開始された場合、PLL クロック通倍率は変更されません。

3.2.2 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 チップを使用した場合のデバイスの動作特性を測定することをお奨めします。発振子 / 水晶ベンダは、タンク回路を調整するための機器と専門的知識を備えており、動作範囲にわたって起動と安定性を確保する適切なタンクコンポーネント値についてお客様に助言することもできます。

3.3 低電力モード・ブロック

281x デバイスの低電力モードは、240x デバイスと同様です。図 3-8 に、さまざまなモードを要約します。

表 3-8 281x の低電力モード

モード	LPMCR0[1:0]	OSCCLK	CLKIN	SYSCLKOUT	終了 (注 1)
アイドル	00	オン	オン	オン (注 2)	XRS、 WAKEINT、 任意のイネーブル割り込み、 XNMI_XINT13
スタンバイ	01	オン (ウォッチドッグ は引き続き動作)	オフ	オフ	XRS、 WAKEINT、 XINT1、 XNMI_XINT13、 T1/2/3/4CTRIP、 C1/2/3/4/5/6TRIP、 SCIRXDA、 SCIRXDB、 CANRX、 デバッグ (注 3)
停止	1X	オフ (オシレータ および PLL は オフ、 ウォッチドッグは 動作していない)	オフ	オフ	XRS、 XNMI_XINT13、 デバッグ (注 3)

注 1: 「終了」列は、低電力モードの終了時の信号または状態を示しています。この信号は、デバイスが割り込みを認識できるよう十分な期間にわたって Low にする必要があります。そうしないと、アイドル・モードが終了せず、デバイスは示された低電力モードに戻ります。

注 2: 28x のアイドル・モードの動作は、24x/240x と異なります。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。

注 3: 28x では、CPU へのクロック (CLKIN) がオフになっても、JTAG ポートは引き続き機能できます。

さまざまな低電力モードは、以下のように動作します。

- アイドル・モード:** このモードは、任意のイネーブルされた割り込みまたはプロセッサによって認識された NMI によって終了します。このモードで LPMCR0[1:0] ビットが 00 に設定されている間、LPM ブロックは何のタスクも実行しません。
- 停止モード:** XRS および XNMI_XINT13 外部信号のみが、デバイスを停止モードからウェイクできます。CPU への XNMI 入力は、XNMICR レジスタのイネーブル / ディスエーブル・ビットを持ちます。
- スタンバイ・モード:** LPMCR1 レジスタによって選択された場合、その他すべての信号 (XNMI を含む) がデバイスをスタンバイ・モードからウェイクします。デバイスをウェイクする信号を選択する必要があります。また、選択された信号は、デバイスをウェイクする前に、OSCCLK によってフィルタリング (Qualify) されます。この OSCCLK サイクル数は、LPMCR0 レジスタで指定されます。

低電力モードは、LPMCR0 レジスタ（図 3-8 を参照）および LPMCR1 レジスタ（図 3-9 を参照）によって制御されます。

図 3-8 低電力モード制御0 (LPMCR0) レジスタ

15	8 7	2 1	0
予約済み	QUALSTDBY	LPM	
R-0	R/W-1	R/W-0	

凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表 3-9 低電力モード制御0 (LPMCR0) レジスタのフィールドの説明

ビット	フィールド	説明 (注 1)
15-8	予約済み	予約済み
7-2	QUALSTDBY	デバイスをスタンバイ・モードからウェイクするときに、選択された入力をフィルタリング (Qualify) する OSCCLK クロック・サイクル数を選択します。 000000 = 2 OSCCLKs 000001 = 3 OSCCLKs 111111 = 65 OSCCLKs
1-0	LPM (注 2)	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。 00 低電力モードをアイドルに設定 01 低電力モードをスタンバイに設定 1X 低電力モードを停止に設定

注 1: これらのビットは、リセット (XRS) によってクリアされます。

注 2: 低電力モード・ビット (LPM) は、IDLE 命令が実行された場合のみ有効です。そのため、ユーザーは、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。

図 3-9 低電力モード制御1 (LPMCR1) レジスタ

15	14	13	12	11	10	9	8
CANRX	SCIRXB	SCIRXA	C6TRIP	C5TRIP	C4TRIP	C3TRIP	C2TRIP
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
C1TRIP	T4CTRIP	T3CTRIP	T2CTRIP	T1CTRIP	WDINT	XNMI	XINT1
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 = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表 3-10 低電力モード制御 1 (LPMCR1) レジスタのフィールドの説明

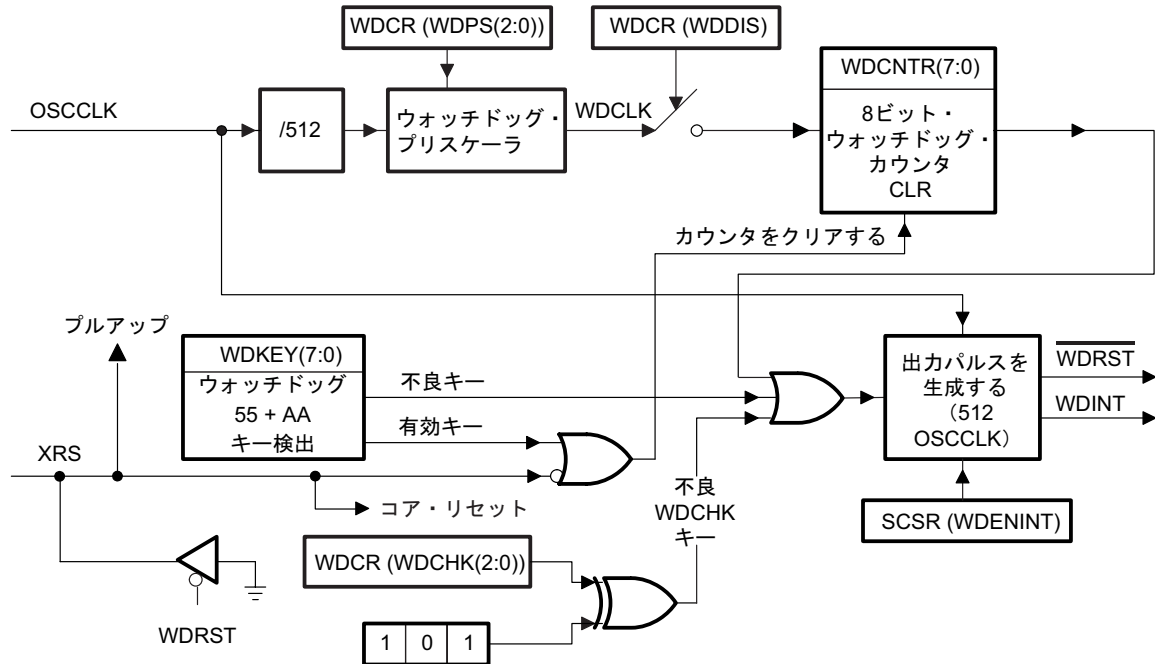
ビット	フィールド	説明
15	CANRX	
14	SCIRXB	
13	SCIRXA	
12	C6TRIP	
11	C5TRIP	
10	C4TRIP	
9	C3TRIP	
8	C2TRIP	それぞれのビットが 1 に設定されている場合、選択された信号によってデバイスをスタンバイ・モードからウェイクすることができます。このビットがクリアされている場合、信号は何の影響も与えません。
7	C1TRIP	
6	T4CTRIP	
5	T3CTRIP	
4	T2CTRIP	
3	T1CTRIP	
2	$\overline{\text{WDINT}}$	
1	XNMI_XINT13	
0	XINT1	

注 1: これらのビットは、リセット ($\overline{\text{XRS}}$) によってクリアされます。

3.4 ウォッチドッグ・ブロック

281x のウォッチドッグ・ブロックは、240x デバイスで使用されるものと似ています。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到達するたびに、512 オシレータ・クロック (OSCCLK) 幅の出力パルスを生成します。これを回避するには、ユーザーがカウンタをディスエーブルにするか、ソフトウェアが定期的に 0x55 と 0xAA を順番にウォッチドッグ・キー・レジスタにライトし、これによってウォッチドッグ・カウンタをリセットする必要があります。図 3-10 は、ウォッチドッグ・モジュール内の各機能ブロックを示しています。

図 3-10 ウォッチドッグ・モジュール



注 A : 512 OSCCLK サイクルの間、 $\overline{\text{WDRST}}$ 信号は Low でドライブされます。

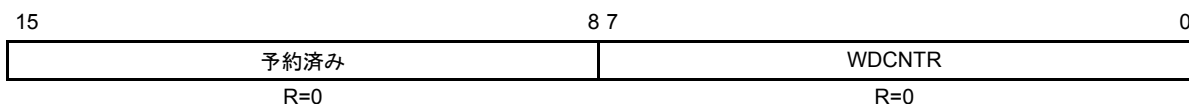
ウォッチドッグ割り込み ($\overline{\text{WDINT}}$) 信号により、ウォッチドッグをアイドル/スタンバイ・モード・タイマからのウェイクアップとして使用できます。ウォッチドッグ割り込みを使用してアイドルまたはスタンバイ低電力モード状態からウェイクする場合、アイドルまたはスタンバイ・モードに戻す前に、 $\overline{\text{WDINT}}$ 信号が High に戻っていることを確認する必要があります。この信号の状態は、SCSR レジスタの WDENINT ビットをリードすることによって判断できます。

スタンバイ・モードでは、デバイス内のすべてのペリフェラルがオフになります。引き続き機能する唯一のペリフェラルがウォッチドッグです。ウォッチドッグ・モジュールには、PLL クロックまたはオシレータ・クロックが流れます。 $\overline{\text{WDINT}}$ 信号が LPM ブロックに供給され、デバイスをスタンバイからウェイクすることができます (イネーブルになっている場合)。詳細については、このデータシートの「低電力モード・ブロック」を参照してください。

アイドル・モードでは、 $\overline{\text{WDINT}}$ 信号は CPU への割り込みを生成し (PIE の WAKEINT 割り込み)、CPU をアイドル・モードから別のモードに切り替えることができます。

停止モードでは、オシレータ (および PLL) がオフになっているため、ウォッチドッグもオフになり、この機能は使用できません。

図3-11 ウォッチドッグ・カウンタ (WDCNTR) レジスタ



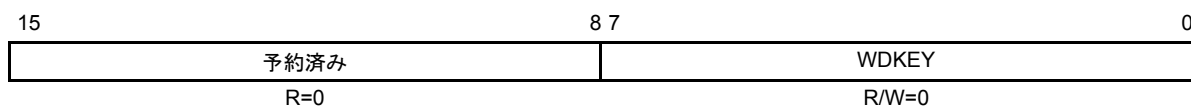
凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表3-11 ウォッチドッグ・カウンタ (WDCNTR) 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	予約済み	
7-0	WDCNTR	これらのビットには、WD カウンタの現行値が保持されます。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すると、ウォッチドッグはリセットを開始します。WDKEY レジスタに有効な組み合わせをライトすると、カウンタは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3-12 ウォッチドッグ・リセット・キー (WDKEY) レジスタ



凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表3-12 ウォッチドッグ・リセット・キー (WDKEY) 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	予約済み	予約済み
7-0	WDKEY	0x55 に続いて 0xAA をライトすると、WDCNTR ビットがクリアされます。その他の値をライトすると、ウォッチドッグ・リセットが生成されます。リードは、WDCR レジスタの値を返します。

図 3-13 ウォッチドッグ制御 (WDCR) レジスタ

15				8							
予約済み											
R=0											
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 = リード・アクセス、W = ライト・アクセス、W1C = クリアするには 1 をライトする、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表 3-13 ウォッチドッグ制御 (WDCR) 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	予約済み	
7	WDFLAG	ウォッチドッグ・リセット・ステータス・フラグ・ビット。WDFLAG が 1 にセットされている場合、ウォッチドッグ・リセット (WDRST) によってリセットが生成されたことを示します。0 の場合、前回リセットが外部デバイスまたは電源投入によるリセットであったことを示します。このビットは、ユーザーが 1 をライトしてこの状態をクリアするまで、ラッチされたままになります。0 をライトした場合は無視されます。
6	WDDIS	ウォッチドッグ・ディスエーブル。WDDIS に 1 をライトすると、ウォッチドッグ・モジュールがディスエーブルになります。0 をライトすると、このモジュールがイネーブルになります。WDDIS は、SCSR2 レジスタの WDOVERRIDE ビットが 1 に設定されている場合のみ変更できます。リセット時には、ウォッチドッグ・モジュールはイネーブルになります。
5-3	WDCHK(2-0)	ウォッチドッグ・チェック。このレジスタへのライトを実行するたびに、必ずこれらのビットに 1,0,1 をライトする必要があります。ほかの値をライトすると、コアに対して即時リセットが行われます (WD がイネーブルの場合)。WDCHK をリードすると必ず 000 を返します。
2-0	WDPS(2-0)	これらのビットは、OSCCLK/512 に対する以下のウォッチドッグ・カウンタ・クロック (WDCLK) レートを設定します。
	000	WDCLK = OSCCLK/512/1
	001	WDCLK = OSCCLK/512/1
	010	WDCLK = OSCCLK/512/2
	011	WDCLK = OSCCLK/512/4
	100	WDCLK = OSCCLK/512/8
	101	WDCLK = OSCCLK/512/16
	110	WDCLK = OSCCLK/512/32
	111	WDCLK = OSCCLK/512/64

XRS 行が 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が検出され (同期化および 4 サイクルの遅延の後)、XRS 信号が High の場合のみ 1 にセットされます。WDRST が High になったときに XRS 信号が Low の場合、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 入力に接続します。そのため、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には、外部リセットの期間がウォッチドッグ・パルスより長い必要があります。

3.4.1 エミュレーションの考慮事項

ウォッチドッグ・モジュールは、さまざまな状況のデバッグにおいて以下のように動作します。

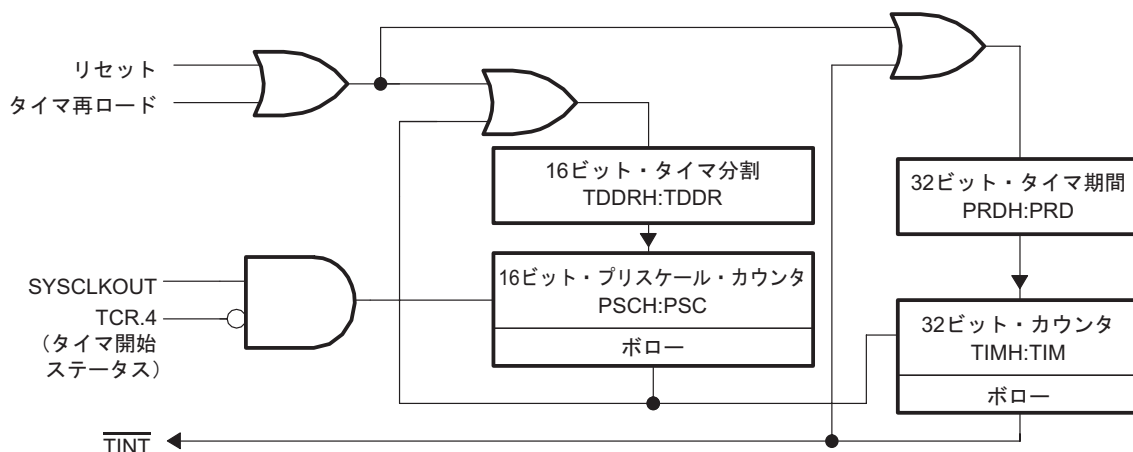
CPU の一時停止 :	CPU が一時停止すると、ウォッチドッグ・クロック (WDCLK) は一時停止します。
ランフリー・モード :	CPU がランフリー・モードに置かれると、ウォッチドッグ・モジュールは通常どおり動作を再開します。
リアルタイム・シングル・ステップ・モード :	CPU がリアルタイム・シングル・ステップ・モードになると、ウォッチドッグ・クロック (WDCLK) は一時停止します。ウォッチドッグは、リアルタイム割り込み内でも一時停止したままになります。
リアルタイム・ランフリー・モード :	CPU がリアルタイム・ランフリー・モードのときは、ウォッチドッグは通常どおり動作します。

3.5 32 ビット CPU タイマ 0/1/2

このセクションでは、281x デバイスの 3 つの 32 ビット CPU タイマ（タイマ 0/1/2）について説明します。

CPU タイマ 1 および 2 は、リアルタイム OS（DSP-BIOS など）用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。

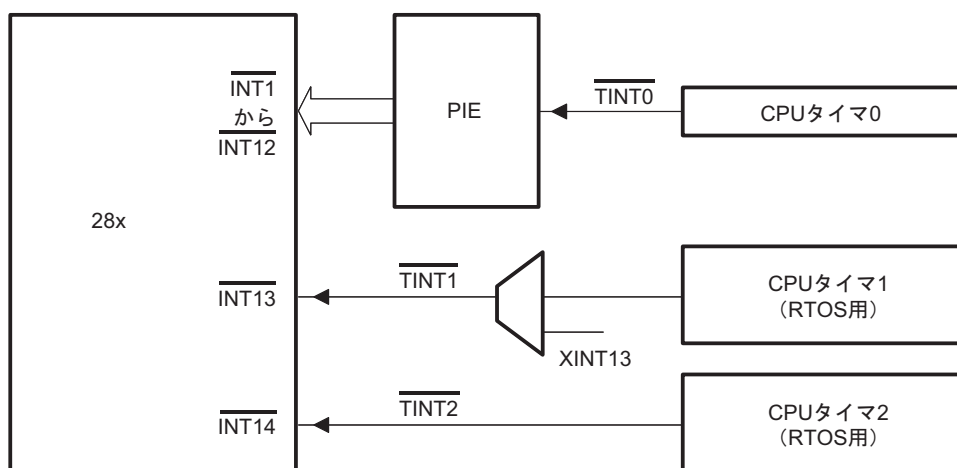
図 3-14 CPU タイマ



注 A： CPU タイマは、イベント・マネージャ・モジュール（EVA、EVB）を持つ汎用（GP）タイマとは異なります。

281x デバイスでは、CPU タイマの割り込み信号（ $\overline{TINT0}$ 、 $\overline{TINT1}$ 、 $\overline{TINT2}$ ）は、図 3-15 に示すように接続されます。

図 3-15 CPU タイマの割り込み信号と出力信号



注 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。

注 B： タイマのタイミングは、プロセッサ・クロックの SYSCLKOUT に同期化されます。

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 が、周期レジスタ PRDH:PRD の値でロードされます。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 に到達すると、タイマ割り込み出力信号により割り込みパルスが生成されます。表 3-14 に挙げるレジスタは、タイマを設定するために使用されます。

表 3-14 CPU タイマ 0、1、2 のコンフィギュレーション・レジスタと制御レジスタ

名前	アドレス	サイズ (× 16)	説明
TIMER0TIM	0X0000 0C00	1	CPU タイマ 0、カウンタ・レジスタ
TIMER0TIMH	0X0000 0C01	1	CPU タイマ 0、カウンタ・レジスタ上位
TIMER0PRD	0X0000 0C02	1	CPU タイマ 0、周期レジスタ
TIMER0PRDH	0X0000 0C03	1	CPU タイマ 0、周期レジスタ上位
TIMER0TCR	0X0000 0C04	1	CPU タイマ 0、制御レジスタ
予約済み	0X0000 0C05	1	
TIMER0TPR	0X0000 0C06	1	CPU タイマ 0、プリスケール・レジスタ
TIMER0TPRH	0X0000 0C07	1	CPU タイマ 0、プリスケール・レジスタ上位
TIMER1TIM	0X0000 0C08	1	CPU タイマ 1、カウンタ・レジスタ
TIMER1TIMH	0X0000 0C09	1	CPU タイマ 1、カウンタ・レジスタ上位
TIMER1PRD	0X0000 0C0A	1	CPU タイマ 1、周期レジスタ
TIMER1PRDH	0X0000 0C0B	1	CPU タイマ 1、周期レジスタ上位
TIMER1TCR	0X0000 0C0C	1	CPU タイマ 1、制御レジスタ
予約済み	0X0000 0C0D	1	
TIMER1TPR	0X0000 0C0E	1	CPU タイマ 1、プリスケール・レジスタ
TIMER1TPRH	0X0000 0C0F	1	CPU タイマ 1、プリスケール・レジスタ上位
TIMER2TIM	0X0000 0C10	1	CPU タイマ 2、カウンタ・レジスタ
TIMER2TIMH	0X0000 0C11	1	CPU タイマ 2、カウンタ・レジスタ上位
TIMER2PRD	0X0000 0C12	1	CPU タイマ 2、周期レジスタ
TIMER2PRDH	0X0000 0C13	1	CPU タイマ 2、周期レジスタ上位
TIMERTCR	0X0000 0C14	1	CPU タイマ 2、制御レジスタ
予約済み	0X0000 0C15	1	

表 3-14 CPU タイマ 0、1、2 のコンフィギュレーション・レジスタと制御レジスタ (続き)

名前	アドレス	サイズ (× 16)	説明
TIMER2TPR	0X0000 0C16	1	CPU タイマ 2、プリスケール・レジスタ
TIMER2TPRH	0X0000 0C17	1	CPU タイマ 2、プリスケール・レジスタ上位
予約済み	0X0000 0C18 0X0000 0C3F	40	

図 3-16 TIMERxTIM レジスタ



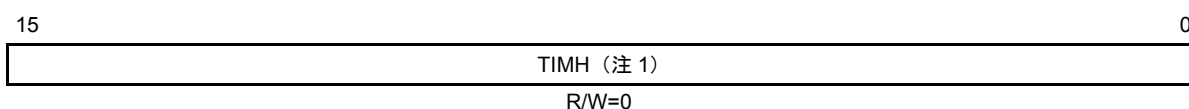
凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

表 3-15 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、タイマの現行の 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、タイマの現行の 32 ビット・カウンタの上位 16 ビットを保持します。TIMH:TIM は、(TDDRH: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。ここで、TDDRH:TDDR は、タイマ・プリスケール分周値です。TIMH:TIM がゼロまでデクリメントすると、PRDH:PRD レジスタに保持された周期値が TIMH:TIM レジスタに再ロードされます。タイマ割り込み (TINT) 信号が生成されます。

注 1 : x = 0、1、または 2

図 3-17 TIMERxTIMH レジスタ



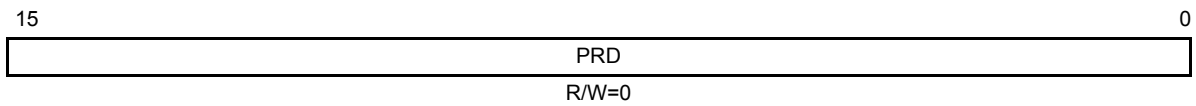
凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

表 3-16 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。

注 1 : x = 0、1、または 2

図 3-18 TIMERxPRD レジスタ (注 1)



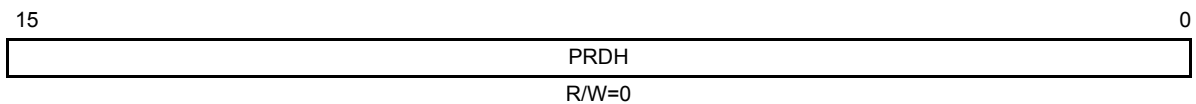
凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

表 3-17 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、次のタイマ入力クロック・サイクル (プリスケアラの出力) の開始時に、PRDH:PRD レジスタに保持された周期値が TIMH:TIM レジスタに再ロードされます。また、タイマ制御レジスタ (TCR) にタイマ再ロード・ビット (TRB) を 1 にセットすると、PRDH:PRD の内容が TIMH:TIM にロードされます。

注 1 : x = 0、1、または 2

図 3-19 TIMERxPRDH レジスタ (注 1)



凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

表 3-18 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。

注 1 : x = 0、1、または 2

図 3-20 TIMERxTCR レジスタ (注1)

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

凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

表 3-19 TIMERxTCR レジスタのフィールドの説明

ビット	フィールド	説明
15	TIF	CPU タイマ割り込みフラグ。 このフラグは、タイマがゼロまでデクリメントするとセットされます。TIF は、ソフトウェアで 1 をライトすることによってクリアできますが、タイマがゼロに到達することによってのみセットされます。 0 0 をライトしても何の影響もありません。 1 このビットに 1 をライトすると、クリアされます。
14	TIE	CPU タイマ割り込みイネーブル。 タイマがゼロまでデクリメントし、TIE がセットされている場合、タイマは割り込み要求をアサートします。
13-12	予約済み	予約済み
11	FREE	CPU タイマ・エミュレーション・モード
10	SOFT	これらのビットは、デバッガにてブレークポイントに遭遇したときのタイマの状態を決定する、特殊なエミュレーション・ビットです。FREE ビットが 1 に設定されている場合、ソフトウェア・ブレークポイントに遭遇しても、タイマは引き続き実行します (すなわち、フリー・ラン)。この場合、SOFT は考慮されません。ただし、FREE が 0 の場合、SOFT は影響を与えます。この場合、SOFT = 0 であれば、タイマは、次に TIMH:TIM がデクリメントしたときに停止します。SOFT ビットが 1 の場合、タイマは、TIMH:TIM がゼロまでデクリメントしたときに停止します。 FREE SOFT CPU タイマ・エミュレーション・モード 0 0 TIMH:TIM が次にデクリメントした後で停止 (ハード・ストップ) 0 1 TIMH:TIM が 0 までデクリメントした後で停止 (ソフト・ストップ) 1 0 フリー・ラン 1 1 フリー・ラン
		注： ソフト・ストップ・モードでは、タイマはシャットダウン前に割り込みを生成します (0 への到達は、割り込みを発生させる状態であるため)。
9-6	予約済み	予約済み

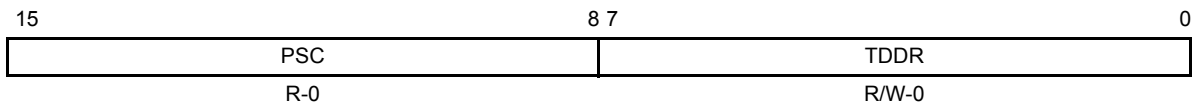
注 1: x = 0、1、または 2

表 3-19 TIMERxTCR レジスタのフィールドの説明 (続き)

ビット	フィールド	説明
5	TRB	CPU タイマ再ロード・ビット。 TRB に 1 をライトすると、PRDH:PRD の値が TIMH:TIM がロードされ、タイマ分周レジスタ (TDDRH:TDDR) の値がプリスケラ・カウンタ (PSCH:PSC) がロードされます。TRB ビットは、常にゼロとしてリードされます。
4	TSS	CPU タイマ停止ステータス・ビット。 TSS は、タイマを停止または開始する 1 ビット・フラグです。 0 タイマを開始または再度開始するには、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、タイマは即座に開始します。 1 タイマを停止するには、TSS を 1 に設定します。
3-0	予約済み	予約済み

注 1 : x=0、1、または 2

図 3-21 TIMERxTPR レジスタ (注 1)



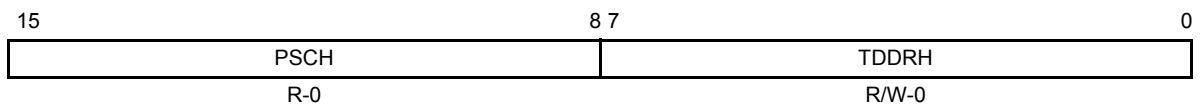
凡例： R=リード・アクセス、W=ライト・アクセス、-0=リセット後の値

表 3-20 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・プリスケラ・カウンタ。 これらのビットは、タイマの現行のプリスケラ・カウントを保持します。PSCH:PSC 値が 0 より大きい時はタイマ・クロック・ソース・サイクルごとに、PSCH:PSC は 1 ずつデクリメントします。PSCH:PSC が 0 に到達した後の 1 タイマ・クロック (タイマ・プリスケラの出力) サイクルで、PSCH:PSC には 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 デクリメントします。また、タイマ再ロード・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 は再ロードされます。PSCH:PSC は、このレジスタをリードすることによってチェックできませんが、直接設定することはできません。このレジスタの値は、タイマ分周レジスタ (TDDRH:TDDR) によって設定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。全体的なタイマ・カウントを整数倍増加させるには、この倍率から 1 を引いた値を TDDRH:TDDR ビットにライトします。プリスケラ・カウンタ (PSCH:PSC) 値が 0 になると、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 に再ロードされ、TIMH:TIM は 1 デクリメントします。また、タイマ再ロード・ビット (TRB) がソフトウェアによってセットされるたびに、TDDRH:TDDR が PSCH:PSC に再ロードされます。

注 1 : x=0、1、または 2

図 3-22 TIMERxTPRH レジスタ (注1)



凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

表 3-21 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。

注 1 : x = 0、1、または 2

余白

汎用 I/O (GPIO)

GPIO MUX レジスタは、281x デバイス上の共有ピンの動作を選択するために使用されます。ピンは、デジタル I/O として動作させるかペリフェラル I/O 信号に接続するかを個別に選択できます (GPxMUX レジスタによって)。デジタル I/O モードで動作させるよう選択した場合、ピンの方向を設定 (GPxDIR レジスタ)、入力信号をフィルタリング (Qualify) することによって不要なノイズを除去 (GPxQUAL レジスタ) ためのレジスタが用意されています。

項目	ページ
4.1 GPIO MUX.....	4-2
4.2 入力フィルタリング (Qualification)	4-6
4.3 レジスタの機能概要.....	4-8
4.4 レジスタ・ビットと I/O のマッピング.....	4-11

4.1 GPIO MUX

表 4-1 に、GPIO MUX レジスタを挙げます。

表 4-1 GPIO MUX レジスタ (注 1, 2)

名前	アドレス	サイズ (× 16)	レジスタの説明
GPAMUX	0x0000 70C0	1	GPIO A MUX 制御レジスタ
GPADIR	0x0000 70C1	1	GPIO A 方向制御レジスタ
GPAQUAL	0x0000 70C2	1	GPIO A 入力 Qualification 制御レジスタ
予約済み	0x0000 70C3	1	
GPBMUX	0x0000 70C4	1	GPIO B MUX 制御レジスタ
GPBDIR	0x0000 70C5	1	GPIO B 方向制御レジスタ
GPBQUAL	0x0000 70C6	1	GPIO B 入力 Qualification 制御レジスタ
予約済み	0x0000 70C7 - 0x0000 70CB	5	
GPDMUX	0x0000 70CC	1	GPIO D MUX 制御レジスタ
GPDDIR	0x0000 70CD	1	GPIO D 方向制御レジスタ
GPDQUAL	0x0000 70CE	1	GPIO D 入力 Qualification 制御レジスタ
予約済み	0x0000 70CF	1	
GPEMUX	0x0000 70D0	1	GPIO E MUX 制御レジスタ
GPEDIR	0x0000 70D1	1	GPIO E 方向制御レジスタ
GPEQUAL	0x0000 70D2	1	GPIO E 入力 Qualification 制御レジスタ
予約済み	0x0000 70D3	1	
GPFMUX	0x0000 70D4	1	GPIO F MUX 制御レジスタ
GPFDIR	0x0000 70D5	1	GPIO F 方向制御レジスタ
予約済み	0x0000 70D6 - 0x0000 70D7	2	
GPGMUX	0x0000 70D8	1	GPIO G MUX 制御レジスタ

注 1: 予約済みのロケーションでは、リード値は定義されていません。また、ライトを行っても何の効果もありません。

注 2: これらのレジスタは EALLOW 保護されています。これにより、偽ライトによる内容の上書きおよびシステムの破壊を防ぎます。

表 4-1 GPIO MUX レジスタ (注 1, 2) (続き)

名前	アドレス	サイズ (× 16)	レジスタの説明
GPGDIR	0x0000 70D9	1	GPIO G 方向制御レジスタ
予約済み	0x0000 70DA - 0x0000 70DF	6	

注 1: 予約済みのロケーションでは、リード値は定義されていません。また、ライトを行っても何の効果もありません。

注 2: これらのレジスタは EALLOW 保護されています。これにより、偽ライトによる内容の上書きおよびシステムの破壊を防ぎます。

デジタル I/O モードに設定されている場合、個々の I/O 信号のセット (GPxSET レジスタ)、個々の I/O 信号のクリア (GPxCLEAR レジスタ)、個々の I/O 信号のトグル (GPxTOGGLE)、または個々の I/O 信号のリード/ライト (GPxDAT レジスタ) を行うための追加のレジスタが用意されています。表 4-2 に、GPIO データ・レジスタを挙げます。

表 4-2 GPIO データ・レジスタ (注 1, 2)

名前	アドレス	サイズ (× 16)	レジスタの説明
GPADAT	0x0000 70E0	1	GPIO A データ・レジスタ
GPASET	0x0000 70E1	1	GPIO A セット・レジスタ
GPACLEAR	0x0000 70E2	1	GPIO A クリア・レジスタ
GPATOGGLE	0x0000 70E3	1	GPIO A トグル・レジスタ
GPBDAT	0x0000 70E4	1	GPIO B データ・レジスタ
GPBSET	0x0000 70E5	1	GPIO B セット・レジスタ
GPBCLEAR	0x0000 70E6	1	GPIO B クリア・レジスタ
GPBTOGGLE	0x0000 70E7	1	GPIO B トグル・レジスタ
予約済み	0x0000 70E8 - 0x0000 70EB	4	
GPDDAT	0x0000 70EC	1	GPIO D データ・レジスタ
GPDSET	0x0000 70ED	1	GPIO D セット・レジスタ
GPDCLEAR	0x0000 70EE	1	GPIO D クリア・レジスタ
GPDTOGGLE	0x0000 70EF	1	GPIO D トグル・レジスタ

注 1: 予約済みのロケーションでは、リード値は定義されていません。また、ライトを行っても何の効果もありません。

注 2: これらのレジスタは EALLOW 保護されておらず、通常、ユーザーによって定期的アクセスされます。

表 4-2 GPIO データ・レジスタ (注 1, 2) (続き)

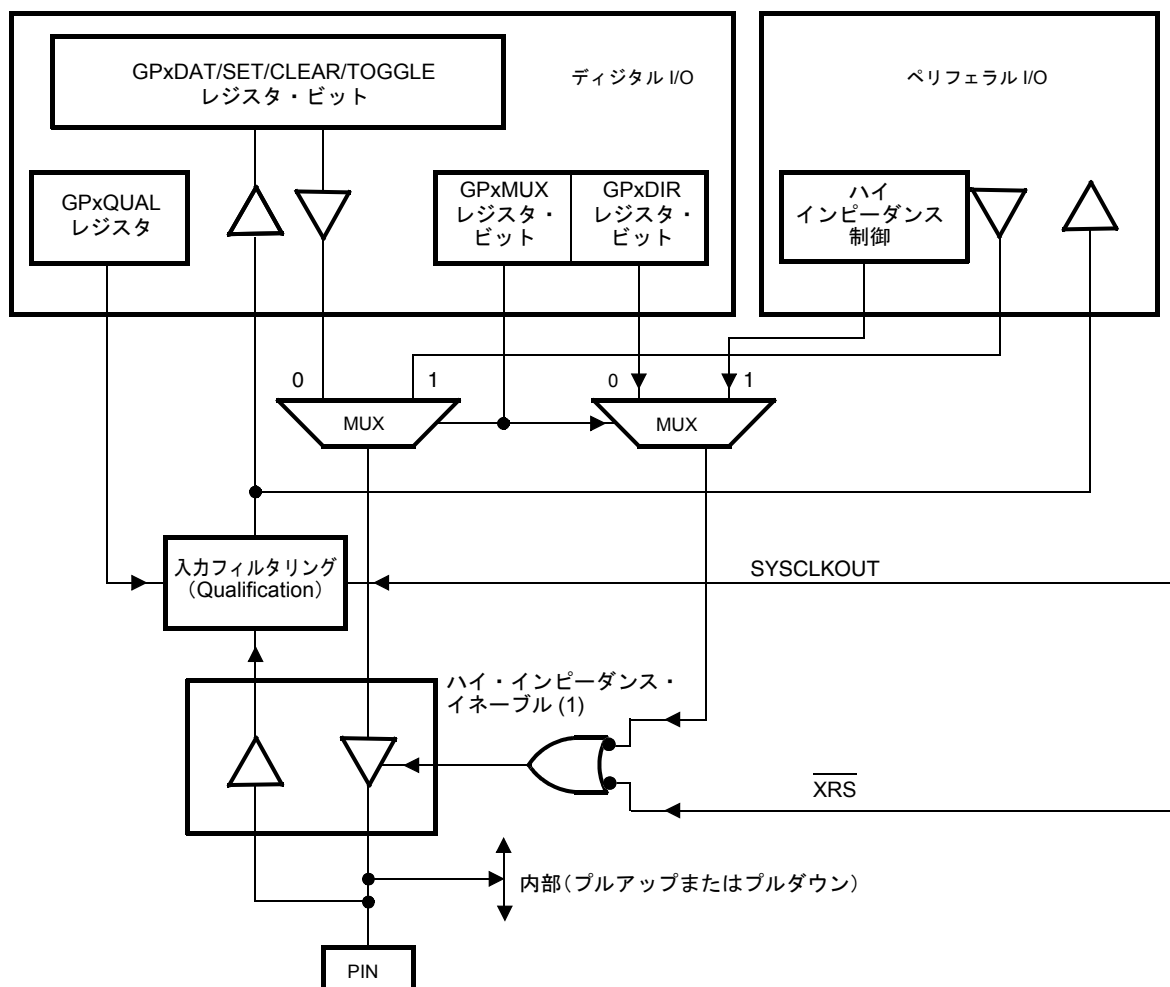
名前	アドレス	サイズ (× 16)	レジスタの説明
GPEDAT	0x0000 70F0	1	GPIO E データ・レジスタ
GPESSET	0x0000 70F1	1	GPIO E セット・レジスタ
GPECLEAR	0x0000 70F2	1	GPIO E クリア・レジスタ
GPETOGGLE	0x0000 70F3	1	GPIO E トグル・レジスタ
GPFDAT	0x0000 70F4	1	GPIO F データ・レジスタ
GPFSET	0x0000 70F5	1	GPIO F セット・レジスタ
GPF CLEAR	0x0000 70F6	1	GPIO F クリア・レジスタ
GPFTOGGLE	0x0000 70F7	1	GPIO F トグル・レジスタ
GPGDAT	0x0000 70F8	1	GPIO G データ・レジスタ
GPGSET	0x0000 70F9	1	GPIO G セット・レジスタ
GPGCLEAR	0x0000 70FA	1	GPIO G クリア・レジスタ
GPGTOGGLE	0x0000 70FB	1	GPIO G トグル・レジスタ
予約済み	0x0000 70FC - 0x0000 70FF	4	

注 1: 予約済みのロケーションでは、リード値は定義されていません。また、ライトを行っても何の効果もありません。

注 2: これらのレジスタは EALLOW 保護されておらず、通常、ユーザーによって定期的にアクセスされます。

図 4-1 は、さまざまなレジスタ・ビットによってさまざまな動作モードがどのように選択されるかを示しています。

図 4-1 マルチプレクスされた GPIO/ ペリフェラル・ピン



注： 1) GPxDAT レジスタにより、動作モードに関わらず、任意のピンの状態をリードできます。

注： 2) GPxQUAL レジスタは、フィルタリング (Qualification) サンプリング期間を指定します。サンプリング・ウィンドウは 6 サンプル幅で、出力は、図 4-3 に示すようにすべてのサンプルが同じ (すべて 0 またはすべて 1) 場合のみ変更されます。この機能により、入力信号から不要なスパイクが除去されます。

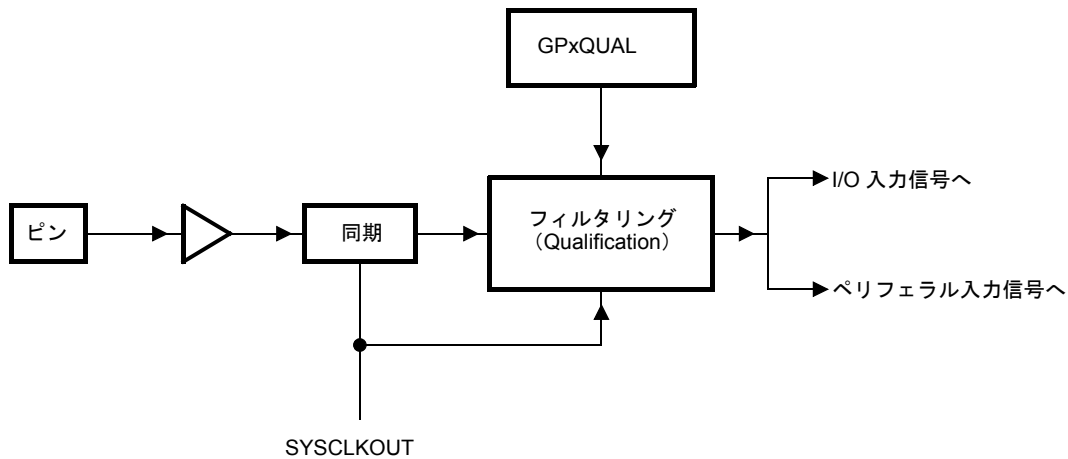
GPIO ピンの入力機能およびペリフェラルへの入力パスは、常にイネーブルです。基本 (ペリフェラル) 機能の出力パスと GPIO ピンの出力パス機能がマルチプレクスされています。ピンの出力バッファは入力バッファに接続されているため、ピン上のすべての GPIO 信号は、ペリフェラル・モジュールにも伝達されます。すなわち、ピンを GPIO 動作用に設定する場合、対応するペリフェラル機能 (および割り込み生成機能) をディスエーブルにする必要があります。そうしないと、割り込みがトリガされる可能性があります。

4.2 入力フィルタリング (Qualification)

入力に対して2つのタイプの入力フィルタリング (Qualification) が実行されます。

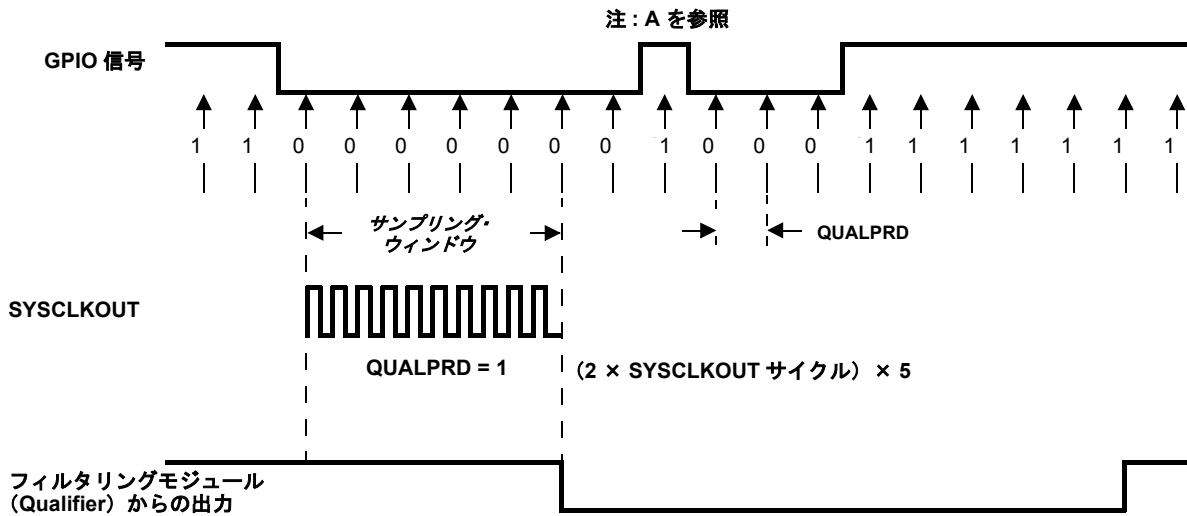
タイプ1のフィルタリング (Qualification) では、入力信号は、まず SYSCLKOUT に同期化され、入力の変更が許可される前に、指定された信号数によってフィルタリング (Qualify) されます。図 4-2 は、タイプ1の入力フィルタリング (Qualification) によってどのように不要なノイズが排除されるかを示しています。

図 4-2 タイプ1の入力フィルタリング (Qualification)



信号をフィルタリング (Qualify) するために、入力は SYSCLKOUT に同期化され、一定の期間でサンプリングされます。サンプリング期間は、GPxQUAL レジスタによって指定されます。サンプリング・ウィンドウは 6 サンプル幅で、入力は、図 4-3 に示すように 6 個のサンプルすべてが同じ場合のみ変更されます。入力信号は非同期であるため、フィルタリング (Qualification) サンプリング・ウィンドウの開始前に、最大 1 SYSCLKOUT の遅延が生じる可能性があります。

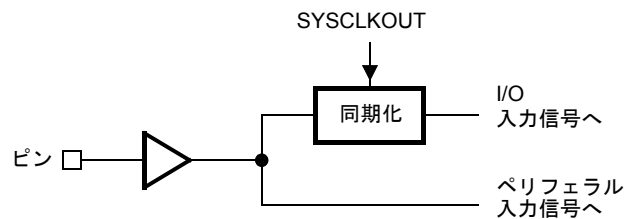
図 4-3 入力フィルタリング (Qualifier) クロック・サイクル



- 注: A このグリッチは、入力フィルタリングモジュール (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) サンプルング期間を指定します。このサンプルング期間は、00 ~ 0xFF の範囲で指定できます。QUALPRD = 00 の場合、入力フィルタリング (Qualification) は適用されません。その他の値「n」の場合は、2n SYSCLKOUT サイクルの限定サンプルング期間が適用されます (すなわち、2n SYSCLKOUT サイクルごとに、GPIO ピンがサンプルングされる)。与えられた入力を認識するためには、連続する 6 個のサンプルが同じ値である必要があります。
- B フィルタリングモジュール (Qualifier) 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一定である必要があります。すなわち、(5 × QUALPRD × 2) SYSCLKOUT サイクルにわたって一定であることが必要です。これにより、6 個のサンプリング・ウィンドウが確実に検出されます。外部信号は非同期で入力されるため、11 SYSCLKOUT 幅のパルスによって信頼性の高い認識を行うことができます。

図 4-4 は、タイプ 2 のフィルタリング (Qualification) を示しています。この場合、デジタル I/O 信号のみがコア・クロック (SYSCLKOUT) に同期化されます。一部のペリフェラルは独自に同期化を行うため、ペリフェラル信号は、フィルタリング (Qualification) または同期化が行われずにピンに直接接続されます。

図 4-4 タイプ 2 の入力フィルタリング (Qualification)



各信号に使用される入力フィルタリング (Qualification) のタイプは、セクション 4.3 のレジスタの機能概要に示されています。

4.3 レジスタの機能概要

各 GP I/O ポートは、MUX、方向、データ、セット、クリア、およびトグル・レジスタによって制御されます。以下のセクションでは、GPxMUX および GPxDIR レジスタのビット・マッピングを示します。対応するデータ、セット、クリア、およびトグル・レジスタのビットと I/O のマッピングは同じです。

GPxMUX レジスタ

各 I/O ポートは、1 個の MUX レジスタを持ちます。MUX レジスタは、各 I/O ピンの動作を、対応するペリフェラル動作または I/O 動作かを選択するために使用されます。リセット時には、すべての GP I/O ピンは I/O ピンとして設定されます。

GPxMUX. ビット = 0 の場合、ピンは I/O として設定されます。

GPxMUX. ビット = 1 の場合、ピンはペリフェラル機能用に設定されます。

図 4-1 に示すように、I/O の入力機能およびペリフェラルへの入力パスは、常にイネーブルです。I/O の出力と、ペリフェラルの出力パスがマルチプレクスされています。すなわち、ピンが I/O ピンとして設定されている場合、対応するペリフェラル機能をディスエーブルにする必要があります。そうしないと、割り込みがトリガされる可能性があります。

GPxDIR レジスタ

各 I/O ポートは、1 個の方向制御レジスタを持ちます。方向レジスタは、対応する I/O ピンを入力として設定するか出力として設定するかを制御します。リセット時には、すべての GP I/O ピンは入力として構成されます。

GPxDIR. ビット = 0 の場合、ピンは入力として設定されます。

GPxDIR. ビット = 1 の場合、ピンは出力として設定されます。

リセット時の、すべての GPxMUX および GPxDIR レジスタ・ビットのデフォルト値は 0 です。すなわち、リセット時には、すべての I/O ポートが入力ピンとして設定されます。GPxDIR レジスタ・ビットを使用してポートの方向を入力から出力に変更する前に、ピンの現行のレベルが GPxDAT レジスタに反映されます。GPxDAT レジスタについては、このセクションで後に説明します。ポートの方向を入力から出力に変更すると、すでに GPxDAT レジスタに保持されている値によってピンの状態が決定されます。

たとえば、ピンが内部プルアップを持つ場合、リセット後にプルアップによって GPxDAT レジスタ・ビットが強制的に 1 に設定され、ピンの現行状態が反映されます。ポートの方向が入力から出力に変化すると、すでに GPxDAT レジスタに保持されている 1 によって、ピンが強制的に同じレベルになります。この方法で、グリッチを発生させることなく、ピンを入力から出力に切り替えることができます。

GPxDAT レジスタ

各 I/O ポートは、1 個のデータ・レジスタを持ちます。データ・レジスタは、入力 I/O 信号のフィルタリング（Qualification）後の現行状態を反映した R/W レジスタです。このレジスタにライトすると、出力として設定されている I/O 信号の対応する出力状態が設定されます。

GPxDAT. ビット = 0 かつピンが出力の場合、ピンの出力を Low にします。

GPxDAT. ビット = 1 かつピンが出力の場合、ピンの出力を High にします。

GPxDAT レジスタを使用して出力ピンのレベルを変更する場合、間違っ別別のピンのレベルを変更しないよう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・ビットに 0 をライトし、GPIOA0 のレベルを変更する場合があります。このとき、命令のリード段階とライト段階の間で別の I/O ポート A 信号のレベルが変化すると、問題が発生する可能性があります。この場合、変更された信号レベルが、命令のリード段階でリードされた元の値で上書きされます。このような状況は、代わりに GPxSET、GPxCLEAR、および GPxTOGGLE を使用することによって回避できます。

GPxSET レジスタ

各 I/O ポートは、1 個のセット・レジスタを持ちます。セット・レジスタは、0 をリードバックするライト専用レジスタです。対応するピンが出力として設定されている場合、セット・レジスタのこのビットに 1 をライトすると、対応するピンの出力を High にします。ビットに 0 をライトしても何の影響もありません。

GPxSET. ビット = 0 の場合、無視されます。

GPxSET. ビット = 1 かつピンが出力の場合、ピンの出力を High にします。

GPxCLEAR レジスタ

各 I/O ポートは、1 個のクリア・レジスタを持ちます。クリア・レジスタは、0 をリードバックするライト専用レジスタです。対応するピンが出力として設定されている場合、クリア・レジスタのこのビットに 1 をライトすると、対応するピンの出力を Low にします。ビットに 0 をライトしても何の影響もありません。

GPxCLEAR. ビット = 0 の場合、無視されます。

GPxCLEAR. ビット = 1 かつピンが出力の場合、ピンの出力を Low にします。

GPxTOGGLE レジスタ

各 I/O ポートは、1 個のトグル・レジスタを持ちます。トグル・レジスタは、0 をリードバックするライト専用レジスタです。対応するピンが出力として設定されている場合、トグル・レジスタのこのビットに 1 をライトすると、対応するピン出力の出力信号が反転します。すなわち、出力ピンが Low の場合、トグル・レジスタの対応するビットに 1 をライトすると、High が出力されます。同様に、出力ピンが High の場合、トグル・レジスタの対応するビットに 1 をライトすると、Low が出力されます。ビットに 0 をライトしても何の影響もありません。

GPxTOGGLE. ビット = 0 の場合、無視されます。

GPxTOGGLE. ビット = 1 かつピンが出力の場合、ピン出力の出力信号を反転します。

4.4 レジスタ・ビットと I/O のマッピング

以下の表は、281x デバイスでのレジスタ・ビットと I/O のマッピングについて説明しています。各ポートにおける MUX、方向、セット、クリア、およびトグル制御レジスタのビット・マッピングは同じです。これらのレジスタの機能については、セクション 4.3 で詳しく説明しています。

一部の入力信号は、タイプ 1 またはタイプ 2 のフィルタリング (Qualification) が掛かります。各信号に対するフィルタリング (Qualification) のタイプは、レジスタ・ビットと I/O のマッピング表の「入力フィルタリング」列に示されています。入力フィルタリング (Qualification) のタイプの詳細については、セクション 4.2 で説明しています。

表 4-3 は、GPIO ポート A の MUX、方向、セット、クリア、およびトグル・レジスタ・ビットと I/O のマッピングを示しています。未使用の I/O は、将来使用するために予約されています。

表 4-3 GPIO A レジスタ・ビットと I/O のマッピング

レジスタ・ビット	ペリフェラル名 GPAMUX ビット = 1	GPIO 名 GPAMUX ビット = 0	GPAMUX/DIR タイプ	入力フィルタ リング
EV-A ペリフェラル				
0	PWM1 (O)	GPIOA0	R/W-0	タイプ 1
1	PWM2 (O)	GPIOA1	R/W-0	タイプ 1
2	PWM3 (O)	GPIOA2	R/W-0	タイプ 1
3	PWM4 (O)	GPIOA3	R/W-0	タイプ 1
4	PWM5 (O)	GPIOA4	R/W-0	タイプ 1
5	PWM6 (O)	GPIOA5	R/W-0	タイプ 1
6	T1PWM_T1CMP (O)	GPIOA6	R/W-0	タイプ 1
7	T2PWM_T2CMP (O)	GPIOA7	R/W-0	タイプ 1
8	CAP1_QEP1 (I)	GPIOA8	R/W-0	タイプ 1
9	CAP2_QEP2 (I)	GPIOA9	R/W-0	タイプ 1
10	CAP3_QEP11 (I)	GPIOA10	R/W-0	タイプ 1
11	TDIRA (I)	GPIOA11	R/W-0	タイプ 1
12	TCLKINA (I)	GPIOA12	R/W-0	タイプ 1
13	C1TRIP (I)	GPIOA13	R/W-0	タイプ 1

注： GPAMUX および GPADIR は、EALLOW 保護されたレジスタです。

表 4-3 GPIO A レジスタ・ビットとI/Oのマッピング (続き)

レジスタ・ビット	ペリフェラル名 GPAMUX ビット = 1	GPIO 名 GPAMUX ビット = 0	GPAMUX/DIR タイプ	入力フィルタ リング
14	C2TRIP (I)	GPIOA14	R/W-0	タイプ 1
15	C3TRIP (I)	GPIOA15	R/W-0	タイプ 1

注: GPAMUX および GPADIR は、EALLOW 保護されたレジスタです。

図 4-5 GPIO A 入力 Qualification 制御 (GPAQUAL) レジスタ

15	8 7	0
予約済み	QUALPRD	
R-0	R/W-0	

凡例: R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注: EALLOW 保護されたレジスタ

表 4-4 GPIO A 入力 Qualification 制御 (GPAQUAL) 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	予約済み	
7-0	QUALPRD	以下のフィルタリング (Qualification) サンプリング期間を指定します。 0x00 フィルタリング (Qualification) なし (SYSCLKOUT への同期化のみ) 0x01 QUALPRD = 2 SYSCLKOUT サイクル 0x02 QUALPRD = 4 SYSCLKOUT サイクル ... 0xFF QUALPRD = 510 SYSCLKOUT サイクル

表 4-5 は、GPIO ポート B の MUX、方向、セット、クリア、およびトグル・レジスタ・ビットと I/O のマッピングを示しています。未使用の I/O は、将来使用するために予約されています。

表 4-5 GPIO B レジスタ・ビットとI/Oのマッピング

レジスタ・ビット	ペリフェラル名 GPBMUX ビット = 1	GPIO 名 GPBMUX ビット = 0	GPBMUX/DIR タイプ	入力フィルタリング
EV-B ペリフェラル				
0	PWM7 (O)	GPIOB0	R/W-0	タイプ 1
1	PWM8 (O)	GPIOB1	R/W-0	タイプ 1
2	PWM9 (O)	GPIOB2	R/W-0	タイプ 1
3	PWM10 (O)	GPIOB3	R/W-0	タイプ 1

注: GPBMUX および GPBDIR は、EALLOW 保護されたレジスタです。

表 4-5 GPIO B レジスタ・ビットとI/Oのマッピング (続き)

レジスタ・ビット	ペリフェラル名 GPBMUX ビット = 1	GPIO 名 GPBMUX ビット = 0	GPBMUX/DIR タイプ	入力フィルタリング
4	PWM11 (O)	GPIOB4	R/W-0	タイプ 1
5	PWM12 (O)	GPIOB5	R/W-0	タイプ 1
6	T3PWM_T3CMP (O)	GPIOB6	R/W-0	タイプ 1
7	T4PWM_T4CMP (O)	GPIOB7	R/W-0	タイプ 1
8	CAP4_QEP3 (I)	GPIOB8	R/W-0	タイプ 1
9	CAP5_QEP4 (I)	GPIOB9	R/W-0	タイプ 1
10	CAP6_QEPI2 (I)	GPIOB10	R/W-0	タイプ 1
11	TDIRB (I)	GPIOB11	R/W-0	タイプ 1
12	TCLKINB (I)	GPIOB12	R/W-0	タイプ 1
13	C4TRIP (I)	GPIOB13	R/W-0	タイプ 1
14	C5TRIP (I)	GPIOB14	R/W-0	タイプ 1
15	C6TRIP (I)	GPIOB15	R/W-0	タイプ 1

注： GPBMUX および GPBDIR は、EALLOW 保護されたレジスタです。

図 4-6 GPIO B 入力 Qualification 制御 (GPBQUAL) レジスタ

15	8 7	0
予約済み	QUALPRD	
R-0	R/W-0	

凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表 4-6 GPIO B 入力 Qualification 制御 (GPBQUAL) 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	予約済み	
7-0	QUALPRD	以下のフィルタリング (Qualification) サンプリング期間を指定します。 0x00 フィルタリング (Qualification) なし (SYSCLKOUT への同期化のみ) 0x01 QUALPRD = 2 SYSCLKOUT サイクル 0x02 QUALPRD = 4 SYSCLKOUT サイクル ... 0xFF QUALPRD = 510 SYSCLKOUT サイクル

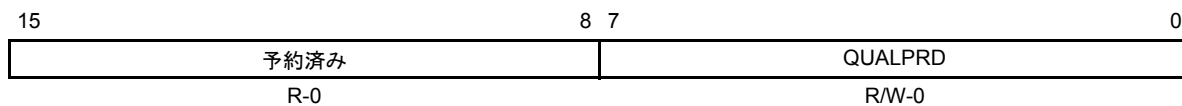
表 4-7 は、GPIO ポート D の MUX、方向、セット、クリア、およびトグル・レジスタ・ビットと I/O のマッピングを示しています。未使用の I/O は、将来使用するために予約されています。

表 4-7 GPIO D レジスタ・ビットと I/O のマッピング

レジスタ・ビット	ペリフェラル名 GPDMUX ビット = 1	GPIO 名 GPDMUX ビット = 0	GPDMUX/DIR タイプ	入力 フィルタリング
EV-A ペリフェラル				
0	T1CTTRIP_PDPINTA (I)	GPIOD0	R/W-0	タイプ 1
1	T2CTTRIP (I)	GPIOD1	R/W-0	タイプ 1
2	予約済み	予約済み	R-0	-
3	予約済み	予約済み	R-0	-
4	予約済み	予約済み	R-0	-
EV-B ペリフェラル				
5	T3CTTRIP_PDPINTB (I)	GPIOD5	R/W-0	タイプ 1
6	T4CTTRIP (I)	GPIOD6	R/W-0	タイプ 1
7	予約済み	予約済み	R-0	-
8	予約済み	予約済み	R-0	-
9	予約済み	予約済み	R-0	-
10	予約済み	予約済み	R-0	-
11	予約済み	予約済み	R-0	-
12	予約済み	予約済み	R-0	-
13	予約済み	予約済み	R-0	-
14	予約済み	予約済み	R-0	-
15	予約済み	予約済み	R-0	-

注： GPDMUX および GPDDIR は、EALLOW 保護されたレジスタです。

図 4-7 GPIO D 入力 Qualification 制御 (GPDQUAL) レジスタ



凡例： R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注： EALLOW 保護されたレジスタ

表 4-8 GPIO D 入力 Qualification 制御 (GPDQUAL) 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	予約済み	
7-0	QUALPRD	以下のフィルタリング (Qualification) サンプリング期間を指定します。 0x00 フィルタリング (Qualification) なし (SYSCLKOUT への同期化のみ) 0x01 QUALPRD = 2 SYSCLKOUT サイクル 0x02 QUALPRD = 4 SYSCLKOUT サイクル . 0xFF QUALPRD = 510 SYSCLKOUT サイクル

表 4-9 は、GPIO ポート E の MUX、方向、セット、クリア、およびトグル・レジスタビットと I/O のマッピングを示しています。未使用の I/O は、将来使用するために予約されています。

表 4-9 GPIO E MUX レジスタ・ビットと I/O のマッピング

レジスタ・ビット	ペリフェラル名 GPEMUX ビット = 0	GPIO 名 GPEMUX ビット = 0	GPEMUX/DIR タイプ	入力フィルタ リング
割り込み				
0	XINT1_XBIO (I)	GPIOE0	R/W-0	タイプ 1
1	XINT2_ADCSOC (I)	GPIOE1	R/W-0	タイプ 1
2	XNMI_XINT13 (I)	GPIOE2	R/W-0	タイプ 1
3	予約済み	予約済み	R-0	-
4	予約済み	予約済み	R-0	-
5	予約済み	予約済み	R-0	-
6	予約済み	予約済み	R-0	-
7	予約済み	予約済み	R-0	-
8	予約済み	予約済み	R-0	-
9	予約済み	予約済み	R-0	-
10	予約済み	予約済み	R-0	-
11	予約済み	予約済み	R-0	-
12	予約済み	予約済み	R-0	-
13	予約済み	予約済み	R-0	-
14	予約済み	予約済み	R-0	-
15	予約済み	予約済み	R-0	-

注： GPEMUX および GPEMUDIR は、EALLOW 保護されたレジスタです。

図 4-8 GPIO E 入力 Qualification 制御 (GPEQUAL) レジスタ

15	8 7	0
予約済み	QUALPRD	
R-0	R/W-0	

凡例: R = リード・アクセス、W = ライト・アクセス、-0 = リセット後の値

注: EALLOW 保護されたレジスタ

表 4-10 GPIO E 入力 Qualification 制御 (GPEQUAL) 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	予約済み	
7-0	QUALPRD	以下のフィルタリング (Qualification) サンプリング期間を指定します。 0x00 フィルタリング (Qualification) なし (SYSCLKOUT への同期化のみ) 0x01 QUALPRD = SYSCLKOUT/2 0x02 QUALPRD = SYSCLKOUT/4 . 0xFF QUALPRD = SYSCLKOUT/510

表 4-11 は、GPIO ポート F の MUX、方向、セット、クリア、およびトグル・レジスタ・ビットと I/O のマッピングを示しています。未使用の I/O は、将来使用するために予約されています。

表 4-11 GPIO F レジスタ・ビットと I/O のマッピング

レジスタ・ビット	ペリフェラル名 GPFMUX ビット = 1	GPIO 名 GPFMUX ビット = 0	GPFMUX/DIR タイプ	入力フィルタ リング
SPI ペリフェラル				
0	SPISIMO (O)	GPIOF0	R/W-0	タイプ 2
1	SPISOMI (I)	GPIOF1	R/W-0	タイプ 2
2	SPICLK (I/O)	GPIOF2	R/W-0	タイプ 2
3	SPISTE (I/O)	GPIOF3	R/W-0	タイプ 2
SCIA ペリフェラル				
4	SCITXDA (O)	GPIOF4	R/W-0	タイプ 2
5	SCIRXDA (I)	GPIOF5	R/W-0	タイプ 2
CAN ペリフェラル				
6	CANTX (O)	GPIOF6	R/W-0	タイプ 2
7	CANRX (I)	GPIOF7	R/W-0	タイプ 2

注: GPFMUX および GPFDIR は、EALLOW 保護されたレジスタです。

表 4-11 GPIO F レジスタ・ビットとI/Oのマッピング (続き)

レジスタ・ビット	ペリフェラル名 GPFMUX ビット = 1	GPIO 名 GPFMUX ビット = 0	GPFMUX/DIR タイプ	入力フィルタ リング
McBSP ペリフェラル				
8	MCLKX (I/O)	GPIOF8	R/W-0	タイプ 2
9	MCLKR (I/O)	GPIOF9	R/W-0	タイプ 2
10	MFSX (I/O)	GPIOF10	R/W-0	タイプ 2
11	MFSR (I/O)	GPIOF11	R/W-0	タイプ 2
12	MDX (O)	GPIOF12	R/W-0	タイプ 2
13	MDR (I)	GPIOF13	R/W-0	タイプ 2
XF CPU 出力信号				
14	XF (O)	GPIOF14	R/W-0	タイプ 2
15	予約済み	予約済み	R-0	-

注： GPFMUX および GPFDIR は、EALLOW 保護されたレジスタです。

表 4-12 は、GPIO ポート G の MUX、方向、セット、クリア、およびトグル・レジスタ・ビットと I/O のマッピングを示しています。未使用の I/O は、将来使用するために予約されています。

表 4-12 GPIO G レジスタ・ビットとI/Oのマッピング

レジスタ・ビット	ペリフェラル名 GPGMUX ビット = 1	GPIO 名 GPGMUX ビット = 0	GPGMUX/DIR タイプ	入力フィルタ リング
0	予約済み	予約済み	R-0	-
1	予約済み	予約済み	R-0	-
2	予約済み	予約済み	R-0	-
3	予約済み	予約済み	R-0	-
SCI-B ペリフェラル				
4	SCITXDB (O)	GPIOG4	R/W-0	タイプ 2
5	SCIRXDB (I)	GPIOG5	R/W-0	タイプ 2
6	予約済み	予約済み	R-0	-
7	予約済み	予約済み	R-0	-
8	予約済み	予約済み	R-0	-

注： GPGMUX および GPGDIR は、EALLOW 保護されたレジスタです。

表 4-12 GPIO G レジスタ・ビットとI/Oのマッピング (続き)

レジスタ・ビット	ペリフェラル名 GPGMUX ビット = 1	GPIO 名 GPGMUX ビット = 0	GPGMUX/DIR タイプ	入力フィルタ リング
9	予約済み	予約済み	R-0	-
10	予約済み	予約済み	R-0	-
11	予約済み	予約済み	R-0	-
12	予約済み	予約済み	R-0	-
13	予約済み	予約済み	R-0	-
14	予約済み	予約済み	R-0	-
15	予約済み	予約済み	R-0	-

注： GPGMUX および GPGDIR は、EALLOW 保護されたレジスタです。

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	5-2
5.2 EALLOW 保護レジスタ	5-5
5.3 デバイス・エミュレーション・レジスタ	5-10
5.4 ライト後のリード保護	5-13

5.1 ペリフェラル・フレーム・レジスタ

281x 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。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・バスへ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・バスへマップ
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・バスへマップ
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 のレジスタ

名前	アドレス範囲	サイズ (× 16)	アクセスタイプ (注 1)
デバイス・エミュレーション・レジスタ	0x0000 0880 0x0000 09FF	384	EALLOW 保護されている
予約済み	0x0000 0A00 0x0000 0B00	128	
フラッシュ・レジスタ (注 2)	0x0000 0A80 0x0000 0ADF	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	0x0000 0AE0 0x0000 0AEF	16	EALLOW 保護されている
予約済み	0x0000 0AF0 0x0000 0B1F	48	
XINTF レジスタ	0x0000 0B20 0x0000 0B3F	32	EALLOW 保護されていない
予約済み	0x0000 0B40 0x0000 0BFF	192	
CPU タイマ 0/1/2 レジスタ	0x0000 0C00 0x0000 0C3F	64	EALLOW 保護されていない
予約済み	0x0000 0C40 0x0000 0CDF	160	
PIE レジスタ	0x0000 0CE0 0x0000 0CFF	32	EALLOW 保護されていない

注 1: レジスタが EALLOW 保護されている場合は、EALLOW 命令を実行するまで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、不正コードやポインタがレジスタのコンテンツを壊さないようにします。

注 2: フラッシュ・レジスタは、コード・セキュリティ・モジュール (CSM) によっても保護されます。

表 5-1 ペリフェラル・フレーム 0 のレジスタ (続き)

名前	アドレス範囲	サイズ (× 16)	アクセスタイプ (注 1)
PIE ベクタ・テーブル	0x0000 0D00 0x0000 0DFF	256	EALLOW 保護されている
予約済み	0x0000 0E00 0x0000 0FFF	512	

注 1: レジスタが EALLOW 保護されている場合は、EALLOW 命令を実行するまで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、不正コードやポインタがレジスタのコンテンツを壊さないようにします。

注 2: フラッシュ・レジスタは、コード・セキュリティ・モジュール (CSM) によっても保護されます。

表 5-2 ペリフェラル・フレーム 1 のレジスタ

名前 (注 1)	アドレス範囲	サイズ (× 16)	アクセスタイプ
eCAN レジスタ	0x00 6000 0x00 60FF	256 (128 × 32)	一部の eCAN 制御レジスタ (および他の eCAN 制御レジスタ内で選択されているビット) は、EALLOW 保護されています。5-5 ページの「EALLOW 保護レジスタ」を参照してください。
eCAN メールボックス RAM	0x00 6100 0x00 61FF	256 (128 × 32)	EALLOW 保護されていない
予約済み	0x00 6200 0x00 6FFF	3584	

注 1: ペリフェラル・フレーム 1 は 16 ビットと 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 5-3 ペリフェラル・フレーム 2 のレジスタ (注 1)

名前	アドレス範囲	サイズ (× 16)	アクセスタイプ
予約済み	0x0000 7000 0x0000 700F	16	
システム制御レジスタ	0x0000 7010 0x0000 702F	32	EALLOW 保護されている
予約済み	0x0000 7030 0x0000 703F	16	
SPI レジスタ	0x0000 7040 0x0000 704F	16	EALLOW 保護されていない
SCI-A レジスタ	0x0000 7050 0x0000 705F	16	EALLOW 保護されていない
予約済み	0x0000 7060 0x0000 706F	16	

注 1: ペリフェラル・フレーム 2 は 16 ビット・アクセスのみを許可します。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か、入力されます)。

表 5-3 ペリフェラル・フレーム 2 のレジスタ (注 1) (続き)

名前	アドレス範囲	サイズ (× 16)	アクセスタイプ
外部割り込みレジスタ	0x0000 7070 0x0000 707F	16	EALLOW 保護されていない
予約済み	0x0000 7080 0x0000 70BF	64	
GPIO MUX レジスタ	0x0000 70C0 0x0000 70DF	32	EALLOW 保護されている
GPIO データ・レジスタ	0x0000 70E0 0x0000 70FF	32	EALLOW 保護されていない
ADC レジスタ	0x0000 7100 0x0000 711F	32	EALLOW 保護されていない
予約済み	0x0000 7120 0x0000 73FF	736	
EV-A レジスタ	0x0000 7400 0x0000 743F	64	EALLOW 保護されていない
予約済み	0x0000 7440 0x0000 74FF	192	
EV-B レジスタ	0x0000 7500 0x0000 753F	64	EALLOW 保護されていない
予約済み	0x0000 7540 0x0000 774F	528	
SCI-B レジスタ	0x0000 7750 0x0000 775F	16	EALLOW 保護されていない
予約済み	0x0000 7760 0x0000 77FF	160	
McBSP レジスタ	0x0000 7800 0x0000 783F	64	EALLOW 保護されていない
予約済み	0x0000 7840 0x0000 7FFF	1984	

注 1: ペリフェラル・フレーム 2 は 16 ビット・アクセスのみを許可します。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か、入力されます)。

5.2 EALLOW 保護レジスタ

281x デバイス上の一部の制御レジスタは、EALLOW 保護メカニズムによって偽 CPU ライトから保護されています。表 5-4 に示すように、ステータス・レジスタ 1 (ST1) の EALLOW ビットは保護の状態を示します。

表 5-4 EALLOW 保護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	許可	許可 (注 1)	許可
1	許可	許可	許可	許可

注 1: コード・コンポーザ・スタジオ・インターフェイスからのデバッグ中、EALLOW ビットは、JTAG ポート経由で無効にされ、保護レジスタへのフル・アクセスが許可されます。

リセットすると、EALLOW ビットがクリアされ、EALLOW 保護がイネーブルになります。保護中は、CPU による保護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、JTAG ライトのみが許可されます。EALLOW 命令を実行してこのビットを設定すると、CPU は保護レジスタへ自由にライトできるようになります。レジスタの変更後、EALLOW ビットをクリアする EDIS 命令を実行して再び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
- 一部の 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	アドレス範囲	サイズ (× 16)	説明
DEVICECNF	0x0000 0880 0x0000 0881	2	デバイス・コンフィギュレーション・レジスタ
PROTSTART	0x0000 0884	1	ブロック保護開始アドレス・レジスタ
PROTRANGE	0x0000 0885	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 (× 16)	説明
コンフィギュレーション・レジスタ			
FOPT	0x0000-0A80	1	フラッシュ・オプション・レジスタ
FPWR	0x0000-0A82	1	フラッシュ・パワー・モード・レジスタ
FSTATUS	0x0000-0A83	1	ステータス・レジスタ
FSTDBYWAIT	0x0000-0A84	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	0x0000-0A85	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	0x0000-0A86	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	0x0000-0A87	1	OTP リード・アクセス・ウェイト・ステートのレジスタ

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) のレジスタ

レジスタ名	メモリ・アドレス	リセット値	レジスタの説明
キー・レジスタ - ユーザーがアクセス可能			
KEY0	0x0000 - 0AE0	0xFFFF	128 ビット・キー・レジスタの下位ワード
KEY1	0x0000 - 0AE1	0xFFFF	128 ビット・キー・レジスタの 2 番目のワード
KEY2	0x0000 - 0AE2	0xFFFF	128 ビット・キー・レジスタの 3 番目のワード
KEY3	0x0000 - 0AE3	0xFFFF	128 ビット・キー・レジスタの 4 番目のワード
KEY4	0x0000 - 0AE4	0xFFFF	128 ビット・キー・レジスタの 5 番目のワード
KEY5	0x0000 - 0AE5	0xFFFF	128 ビット・キー・レジスタの 6 番目のワード
KEY6	0x0000 - 0AE6	0xFFFF	128 ビット・キー・レジスタの 7 番目のワード
KEY7	0x0000 - 0AE7	0xFFFF	128 ビット・キー・レジスタの上位ワード
CSMSCR	0x0000 - 0AEF		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 (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
未使用	0x0000 0D00	2	予約済み	-	-
	0x0000 0D02				
	0x0000 0D04				
	0x0000 0D06				
	0x0000 0D08				
	0x0000 0D0A				
	0x0000 0D0C				
	0x0000 0D0E				
	0x0000 0D10				
	0x0000 0D12				
	0x0000 0D14				
	0x0000 0D16				
	0x0000 0D18				
INT13	0x0000 0D1A	2	外部割り込み 13 (XINT13) または CPU タイマ 1 (RTOS 用)	17	-
INT14	0x0000 0D1C	2	CPU タイマ 2 (RTOS 用)	18	-
DATALOG	0x0000 0D1E	2	CPU データ・ログ割り込み	19 (最下位)	-
RTOSINT	0x0000 0D20	2	CPU リアルタイム OS 割り込み	4	-
EMUJINT	0x0000 0D22	2	CPU エミュレーション割り込み	2	-
NMI	0x0000 0D24	2	外部ノンマスクابل割り込み	3	-
ILLEGAL	0x0000 0D26	2	不正命令	-	-
USER0	0x0000 0D28	2	ユーザー定義のトラップ	-	-
.
USER11	0x0000 0D3E	2	ユーザー定義のトラップ	-	-
INT1.1	0x0000 0D40	2	グループ 1 割り込みベクタ	5	1 (最上位)
.
INT1.8	0x0000 0D4E	2			8 (最下位)
.	.	.	グループ 2 割り込みベクタ～グループ 11 割り込みベクタ	6 }	
.	.	.		15	
INT12.1	0x0000 0DF0	2	グループ 12 割り込みベクタ	16	1 (最上位)
.
INT12.8	0x0000 0DFE	2			8 (最下位)

表 5-9 EALLOW 保護されている PLL、クロッキング、ウォッチドッグ、低電力モードのレジスタ

名前	アドレス	サイズ (× 16)	説明
HISPCP	0x0000 701A	1	HSPCLK クロック用高速ペリフェラル・クロック・プリスケータ・レジスタ
LOSPCP	0x0000 701B	1	HSPCLK クロック用低速ペリフェラル・クロック・プリスケータ・レジスタ
PCLKCR	0x0000 701C	1	ペリフェラル・クロック制御レジスタ
LPMCR0	0x0000 701E	1	低電力モード制御レジスタ 0
LPMCR1	0x0000 701F	1	低電力モード制御レジスタ 1
PLLCR	0x0000 7021	1	PLL 制御レジスタ (注 2)
SCSR	0x0000 7022	1	システム制御およびステータス・レジスタ
WDCNTR	0x0000 7023	1	ウォッチドッグ・カウンタ・レジスタ
WDKEY	0x0000 7025	1	ウォッチドッグ・リセット・キー・レジスタ
WDCR	0x0000 7029	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 (× 16)	レジスタの説明
GPAMUX	0x0000 70C0	1	GPIO A MUX 制御レジスタ
GPADIR	0x0000 70C1	1	GPIO A 方向制御レジスタ
GPAQUAL	0x0000 70C2	1	GPIO A 入力 Qualification 制御レジスタ
GPBMUX	0x0000 70C4	1	GPIO B MUX 制御レジスタ
GPBDIR	0x0000 70C5	1	GPIO B 方向制御レジスタ
GPBQUAL	0x0000 70C6	1	GPIO B 入力 Qualification 制御レジスタ
GPDMUX	0x0000 70CC	1	GPIO D MUX 制御レジスタ
GPDDIR	0x0000 70CD	1	GPIO D 方向制御レジスタ
GPDQUAL	0x0000 70CE	1	GPIO D 入力 Qualification 制御レジスタ
GPEMUX	0x0000 70D0	1	GPIO E MUX 制御レジスタ
GPEDIR	0x0000 70D1	1	GPIO E 方向制御レジスタ
GPEQUAL	0x0000 70D2	1	GPIO E 入力 Qualification 制御レジスタ

表 5-10 EALLOW 保護されている GPIO MUX レジスタ (続き)

名前	アドレス	サイズ (× 16)	レジスタの説明
GPFMUX	0x0000 70D4	1	GPIO F MUX 制御レジスタ
GPFDIR	0x0000 70D5	1	GPIO F 方向制御レジスタ
GPGMUX	0x0000 70D8	1	GPIO G MUX 制御レジスタ
GPGDIR	0x0000 70D9	1	GPIO G 方向制御レジスタ

表 5-11 EALLOW 保護されている eCAN レジスタ

名前	アドレス	サイズ (× 16)	説明
CANMC	0x0000-6014	2	マスタ制御レジスタ (注 1)
CANBTC	0x0000-6016	2	ビット・タイミング・コンフィギュレーション・レジスタ (注 2)
CANGIM	0x0000-6020	2	グローバル割り込みマスク・レジスタ (注 3)
CANMIM	0x0000-6024	2	メールボックス割り込みマスク・レジスタ
CANTSC	0x0000-602E	2	タイム・スタンプ・カウンタ
CANTIOC	0x0000-602A	1	CANTXA ピン用入出力制御レジスタ (注 4)
CANRIOC	0x0000-602C	1	CANRXA ピン用入出力制御レジスタ (注 5)

- 注 1 : CANMC[15:9] と [7:6] のビットのみ保護されます。
 注 2 : BCR[23:16] と [10:0] のビットのみ保護されます。
 注 3 : CANGIM[17:16]、[14:8]、[2:0] のビットのみ保護されます。
 注 4 : IOCONT1[3] のみ保護されます。
 注 5 : IOCONT2[3] のみ保護されます。

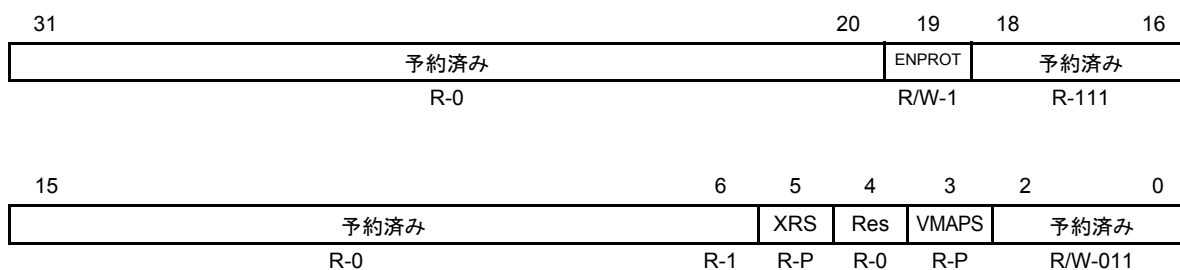
5.3 デバイス・エミュレーション・レジスタ

これらのレジスタは、C28x CPU の保護モードを制御し、一部のクリティカル・デバイス信号を監視するために使用します。これらのレジスタを表 5-12 で定義します。

表 5-12 デバイス・エミュレーション・レジスタ

名前	アドレス範囲	サイズ (× 16)	説明
DEVICECNF	0x0000 0880 0x0000 0881	2	デバイス・コンフィギュレーション・レジスタ
PARTID	0x0000 0882	1	パーツ ID レジスタ
REVID	0x0000 0883	2	デバイス ID レジスタ
PROTSTART	0x0000 0884	1	ブロック保護開始アドレス・レジスタ
PROTRANGE	0x0000 0885	1	ブロック保護範囲アドレス・レジスタ
予約済み	0x0000 0886 0x0000 09FF	378	

図 5-1 デバイス・コンフィギュレーション (DEVICECNF) レジスタ



凡例： R = リード、W = ライト、P = リセット後のピン値、-n = リセット値

注： EALLOW 保護されているレジスタ

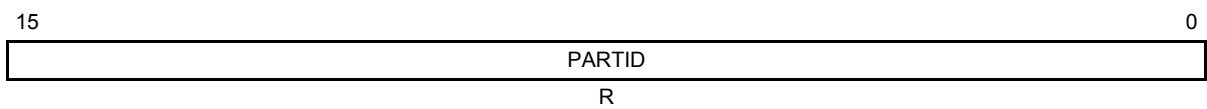
表 5-13 デバイス・コンフィギュレーション (DEVICECNF) レジスタのフィールドの説明

ビット	フィールド	説明
31-20	予約済み	
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 レジスタの指定に従って、ライト・リード保護をイネーブルにします。

表 5-13 デバイス・コンフィギュレーション (DEVICECNF) レジスタのフィールドの説明 (続き)

ビット	フィールド	説明
18-6	予約済み	
5	XRS	リセット入力信号のステータス。 $\overline{\text{XRS}}$ 入力ピンに直接接続されています。
4	予約済み	
3	VMAPS	VMAP 構成ステータス。VMAP のステータスを示します。
2-0	予約済み	

図 5-2 パーツ ID レジスタ



注： 1) リセット時の値はデバイスに依存します。

表 5-14 パーツ ID 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	PARTID	これらの 16 ビットは、以下のようにデバイスのパーツ番号を表します。
	0x0001	F281x (注 2)
	0x0002	F281x (注 2)
	0x0003	C281x
	0x002C	F2801
	0x0034	F2806
	0x003C	F2808

注： 2) リビジョン C 以上の F281x デバイスにおいて、リビジョンを区別する機能はなくなりました。したがって、0x0001 または 0x0002 の値はどんな F281x をも表します。

図 5-3 DEVICEID レジスタ

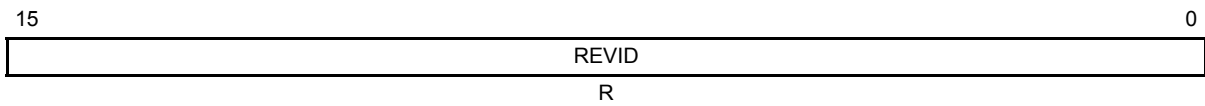


表 5-15 DEVICEID レジスタのフィールドの説明

ビット	フィールド	説明
15-0	REVID	これらの 16 ビットは、特定部分のシリコンのリビジョン番号を指定します。この番号は、シリコンの最初のリビジョンでは常に 0x0000 で始まり、後続のリビジョンでは番号がインクリメントします。
		0x0000 リビジョン 0（最初のシリコン）
		0x0001 リビジョン A
		0x0002 リビジョン B、以下同様

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操作を保護するメモリ・アドレス範囲を設定します（操作は通常のパイプライン順ではなく連続して発生します）。これは、一定のペリフェラル操作に必要な保護です。

例： 以下のコード行は、レジスタ 1 (REG1) へのライトを実行し、続いて、次の命令は、レジスタ 2 (REG2) からのリードを実行します。以下のように、ブロック保護がディセーブルとなっているプロセッサのメモリ・バス上では、リード操作はライト操作の前に実行されます。

```
MOV    @REG1,AL    ----- +
TBIT   @REG2,#BIT_X ----- |-----> リード
                               +-----> ライト
```

以下のように、ブロック保護がイネーブルとなっている場合は、ライトが発生するまでリードは停止します。

```
MOV    @REG1,AL    ----- +
TBIT   @REG2,#BIT_X ---    +      |
                               |    +-----> ライト
                               +-----> リード
```

注：C28x CPU は、同じメモリ・アドレスへのライト後のリードを自動的に保護します。上で説明した保護メカニズムは、アドレスが異なるメモリ内の同じ領域内にある (PROTSTART および PROTRANGE レジスタで定義) 場合のためのものです。

表 5-16 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	0x0000 0884	16	R/W	0x0100 (注 1)	PROTSTART レジスタは、開始アドレスをプロセッサの下位 22 ビット・アドレス範囲の 16 の最上位ビットに基づいて設定します。このため、最小単位は 64 ワードとなります。
PROTRANGE	0x0000 0885	16	R/W	0x00FF (注 1)	PROTRANGE レジスタはブロック・サイズを設定します (開始アドレスから)。64 ワードから開始し、2 進数の倍数となります (64、128、256、512、1K、2K、4K、8K、16K、...2M)。

注 1：これらのレジスタのリセット時のデフォルト値は、メモリ・マップ (アドレス範囲 0x0000 4000 ~ 0x0000 8000) のペリフェラル・フレーム 1、ペリフェラル・フレーム 2、XINTF ゾーン 1 の領域をカバーするように選択されます。

表 5-17 PROTSTART 有効値 (注 1)

開始アドレス	レジスタ値	レジスタ・ビット															
		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	0xFFFFC	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0x003F FF40	0xFFFFD	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0x003F FF80	0xFFFFE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0x003F FFC0	0xFFFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

注 1 : すばやくレジスタ値を計算する方法は、目的のブロック開始アドレスを 64 で割る方法です。

表 5-18 PROTRANGE 有効値 (注2)

ブロックサイズ	レジスタ値	レジスタ・ビット															
		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
2K	0x001F	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
4K	0x003F	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
8K	0x007F	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
16K	0x00FF	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
32K	0x01FF	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
64K	0x03FF	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
128K	0x07FF	0	0	0	0	0	1	1	1	1	1	1	1	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

注2: レジスタ値には有効でないものがあります。PROTSTART アドレス値は、範囲値の倍数でなくてはなりません。例えば、ブロック・サイズが4Kに設定されている場合は、開始アドレスは4K単位でしか設定できません。

余白

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、より小さい割り込み入力セットへマルチプレクスします。PIE ブロックは、8 つのブロックにグループ化された 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 つへ送られます。96 の割り込みはそれぞれ独自のベクタを持ち、これらのベクタは変更可能な専用の RAM ブロック内に保存されています。CPU は、割り込みを処理する際に、適切な割り込みベクタを自動的にフェッチします。9 CPU クロック・サイクルでベクタをフェッチし、クリティカルな CPU レジスタを保存します。このため、CPU は割り込みイベントにすばやく対応できます。割り込みの優先順位は、ハードウェアとソフトウェアによって制御されます。個々の割り込みは、PIE ブロック内でイネーブルまたはディスエーブルにできます。

項目	ページ
6.1 PIE コントローラの概要	6-2
6.2 ベクタ・テーブルのマッピング	6-7
6.3 割り込みソース	6-10
6.4 PIE コンフィギュレーション・レジスタ	6-24
6.5 PIE 割り込みレジスタ	6-26
6.6 外部割り込み制御レジスタ	6-39

6.1 PIE コントローラの概要

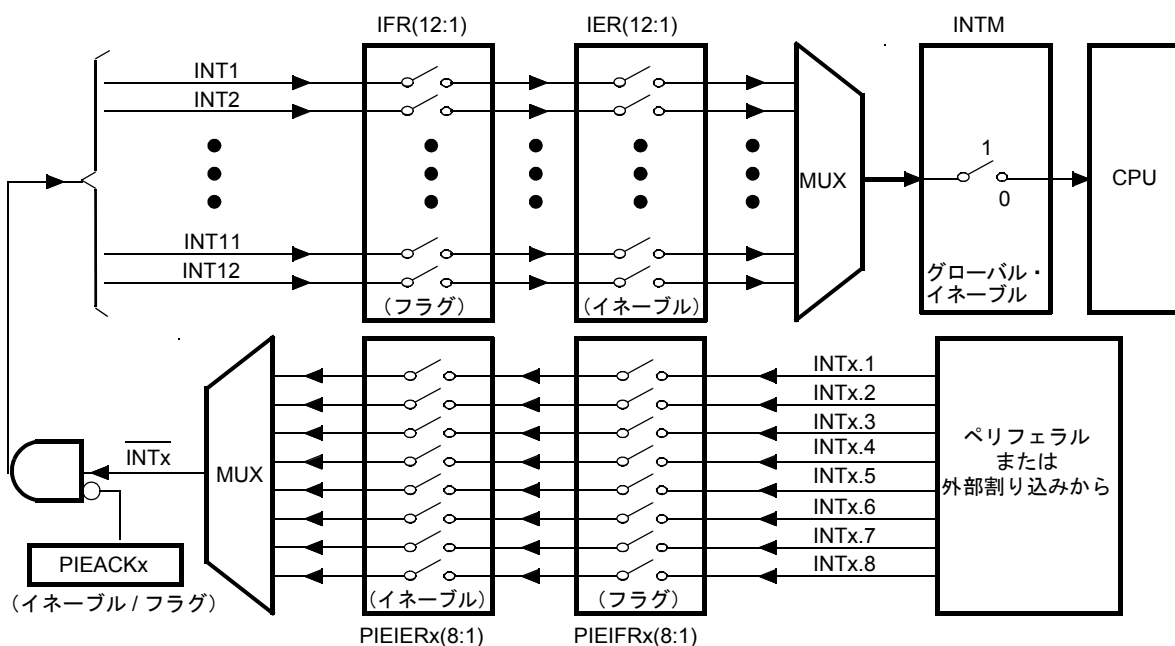
28x CPU は、CPU レベルで、1 つのノンマスクابل割り込み（NMI）と 16 のマスクابل優先割り込み要求（INT1 ~ INT14、RTOSINT、および DLOGINT）をサポートします。28x デバイスは、多くのペリフェラルを持ち、各ペリフェラルは、ペリフェラル・レベルの多くのイベントに対応して 1 つまたは複数の割り込みを生成できます。この CPU はすべてのペリフェラル割り込み要求を CPU レベルで処理する十分な能力を持たないため、ペリフェラルやその他の外部ピンなどのさまざまなソースからの割り込み要求を調停するペリフェラル割り込み拡張（PIE）コントローラが必要となります。

PIE ベクタ・テーブルは、システム内で各割り込みサービス・ルーチン（ISR）のアドレス（ベクタ）を保存するために使用します。すべてのマルチプレクスされたおよびマルチプレクスされていない割り込みを含め、割り込みソースあたり、1 つのベクタがあります。ベクタ・テーブルはデバイス初期化中に作成し、オペレーション中にアップデートできます。

割り込みオペレーションのシーケンス

図 6-1 は、すべてのマルチプレクスされた PIE 割り込みの割り込みオペレーション・シーケンスの概要を示しています。マルチプレクスされていない割り込みソースは CPU へ直接送信されます。

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

ペリフェラル内で割り込み生成イベントが発生します。そのイベントに対応する割り込みフラグ (IF) ビットが、そのペリフェラル用のレジスタ内でセットされます。

対応する割り込みイネーブル (IE) ビット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要求を生成します。割り込みがペリフェラル・レベルでイネーブルになっていない場合は、ソフトウェアがクリアするまで IF はセットされたままとなります。その後割り込みがイネーブルとなった場合も、割り込みフラグがセットされたままの場合は、割り込み要求は PIE へ要求されます。

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、ペリフェラルのリファレンス・ガイドを参照してください。

□ PIE レベル

PIE ブロックは、8つのペリフェラルと外部ピンの割り込みを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、PIE グループ1～12の12グループに分けられます。1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ます。例えば、PIE グループ1はCPU割り込み1 (INT1) へ、PIE グループ12はCPU割り込み12 (INT12) へマルチプレクスされます。残りのCPU割り込みに接続す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ない割り込みでは、PIE は要求を直接CPUへパスします。

マルチプレクスされた各割り込みに対し、PIE ブロック内の各割り込みグループに、フラグ・ビット (PIEIFRx.y) とイネーブル・ビット (PIEIERx.y) が用意されています。さらに、PIE 割り込みグループ (INT1～INT12) のそれぞれに1つずつACKビット (PIEACK) があり、このビットはPIEACKxと呼ばれます。図6-2は、さまざまなPIEIFRおよびPIEIERレジスタ条件下におけるPIEハードウェアの動作を示しています。

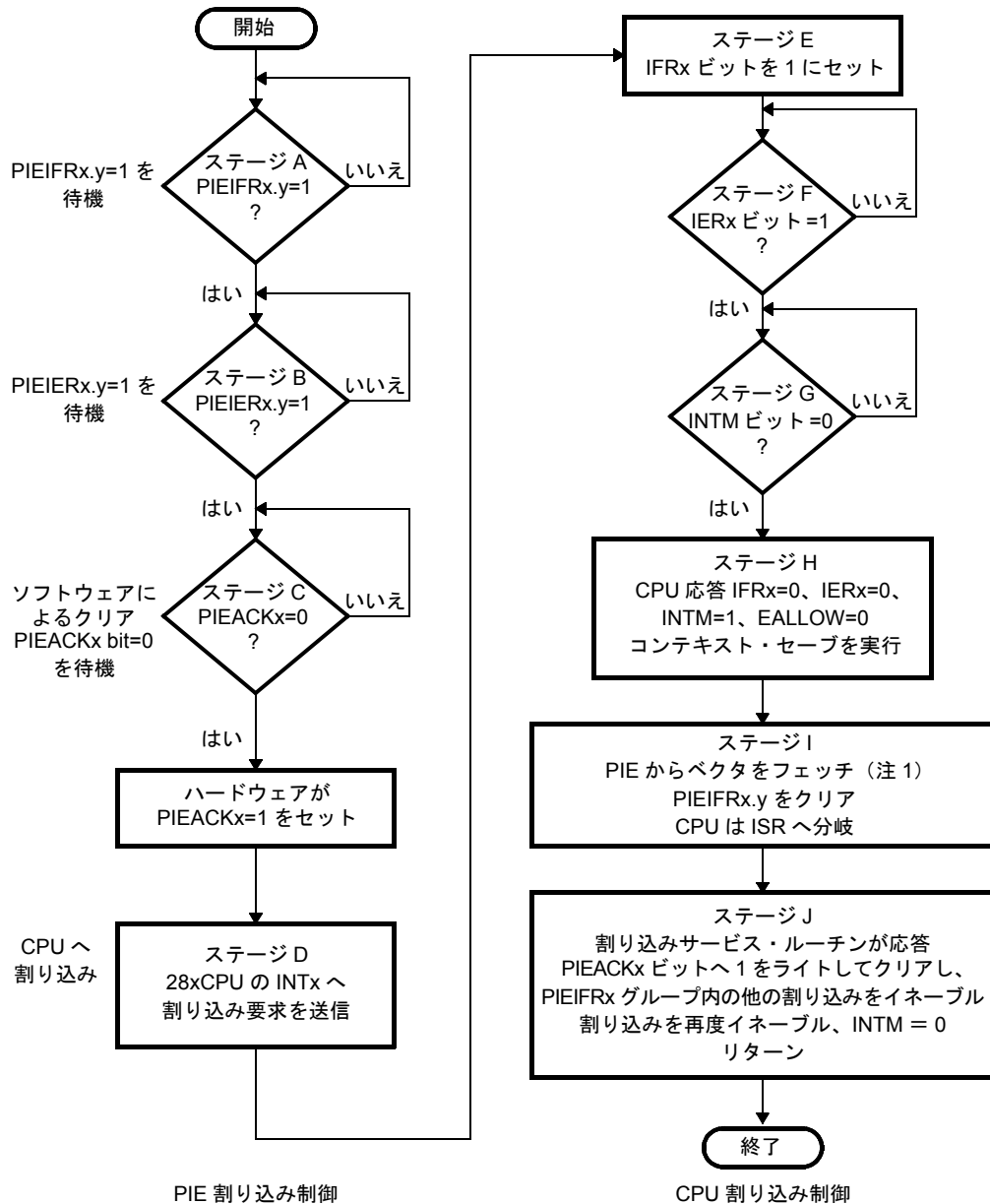
PIE コントローラへ要求が行われると、対応するPIE 割り込みフラグ (PIEIFRx.y) ビットがセットされます。その割り込みの対応するPIE 割り込みイネーブル (PIEIERx.y) ビットもセットされている場合は、PIE は、対応するPIEACKx ビットをチェックし、CPUがそのグループからの割り込みに対する準備ができているかどうかを確認します。そのグループのPIEACKx ビットがクリアされている場合は、PIE はその割り込み要求をCPUへ送信します。PIEACKx がセットされている場合は、PIE はクリアされるのを待ってから、要求をINTxへ送信します。詳細についてはセクション6.3を参照してください。

□ CPU レベル

要求がCPUへ送信されると、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内にラッチされると、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)、およびグローバル割り込みマスク (INTM) ビットが適切にイネーブルにされるまで処理されません。

表 6-1 に示すように、CPU レベルでマスカブル割り込みをイネーブルにするための必要条件は、使用する割り込み処理プロセスによって異なります。通常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・モードで、CPU が停止した場合は、別のプロセスが使用されます。この特殊なケースの場合に、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・モードで、CPU が動作中の場合は、標準の割り込み処理プロセスが適用されます。

図 6-2 代表的な PIE/CPU 割り込み応答 - INTx.y



注 1: マルチプレクスされた割り込みでは、PIE は、フラグが立ち、イネーブルとなっている優先順位が一番高い割り込みに応答します。フラグが立ち、かつ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 (INTx.1) で優先順位が一番高い割り込みが使用されます。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 割り込みをイネーブルにする

割り込み処理プロセス	割り込みをイネーブルにする条件
標準	INTM=0 かつ IER 内ビットが 1
DSP がリアルタイム・モードで、停止中	IER 内ビットが 1 で DBGIER が 1

次に CPU は割り込み処理の準備をします。この準備プロセスは、『TMS320C28x DSP CPU and Instruction Set Reference Guide』（文書番号 SPRU430）に詳細に記載されています。この準備では、対応する CPU IFR と IER ビットがクリアされ、EALLOW と LOOP がクリアされ、INTM と DBGM がセットされ、パイプラインはフラッシュされ、リターン・アドレスが保存され、自動コンテキスト・セーブが実行されます。続いて ISR のベクタが、PIE モジュールからフェッチされます。マルチプレクスされた割り込み要求の場合は、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かをデコードします。このデコード・プロセスについてはセクション 6.3.3 で詳しく説明します。

実行された割り込みサービス・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 96 の割り込みのそれぞれに、32 ビットのベクタが 1 つずつあります。割り込みベクタがフェッチされると、PIE モジュール内の割り込みフラグ (PIEIFRx.y) は、自動的にクリアされます。ただし、同じ PIE グループから追加割り込みを受け取る準備が整った場合は、その割り込みグループの PIE ACK ビットを手動でクリアする必要があります。

6.2 ベクタ・テーブルのマッピング

28xx デバイス上では、割り込みベクタ・テーブルは、メモリ内の 5 つの異なるロケーションにマッピングできます。実際には、F28xx デバイスには、PIE ベクタ・テーブル・マッピングのみが使用されます。

このベクタ・マッピングは、以下のモード・ビット / 信号で制御されます。

VMAP:	VMAP はステータス・レジスタ 1 ST1 (ビット 3) 内にあります。デバイスをリセットすると、このビットは 1 にセットされます。ST1 へのライトや、SETC/CLRC VMAP 命令によってこのビットの状態を変更できます。F2810/12 の通常のオペレーションではこのビットはセットされたままとなります。
M0M1MAP:	M0M1MAP はステータス・レジスタ 1 ST1 (ビット 11) 内にあります。デバイスをリセットすると、このビットは 1 にセットされます。ST1 へのライトや、SETC/CLRC M0M1MAP 命令によってこのビットの状態を変更できます。28xx デバイスの通常のオペレーションではこのビットはセットされたままとなります。M0M1MAP=0 は TI のテスト専用に予約されています。
MP/MC:	このビットは、XINTCNF2 レジスタ (ビット 8) 内にあります。外部インターフェイス (XINTF) のあるデバイスでは、このビットのデフォルト値はリセット時に XMP/MC 入力デバイス信号によってセットされます。XINTF のないデバイスでは、XMP/MC はデバイス内部で LOW に接続されています。このビットの状態は、リセット後、XINTCNF2 レジスタ (アドレス 0x0000 0B34) へライトすることにより変更できます。
ENPIE:	ENPIE は、PIECTRL レジスタ (ビット 0) 内にあります。このビットのデフォルト値はリセット時に 0 にセットされます (PIE をディスエーブルにします)。このビットの状態は、リセット後、PIECTRL レジスタ (アドレス 0x0000 0CE0) へライトすることにより変更できます。

これらのビットと信号を使用する可能性のある、ベクタ・テーブル・マッピングの例を表 6-2 に示します。

表 6-2 割り込みベクタ・テーブル・マッピング (注 1)

ベクタ・マップ	ベクタのフェッチ元	アドレス範囲	VMAP	M0M1MAP	MP/MC	ENPIE
M1 ベクタ (注 2)	M1 SARAM ブロック	0x000000-0x00003F	0	0	X	X
M0 ベクタ (注 2)	M0 SARAM ブロック	0x000000-0x00003F	0	1	X	X
BROM ベクタ	ROM ブロック	0x3FFFC0-0x3FFFFFF	1	X	0	0
XINTF ベクタ (注 3)	XINTF ゾーン 7 ブロック	0x3FFFC0-0x3FFFFFF	1	X	1	0
PIE ベクタ	PIE ブロック	0x000D00-0x000DFF	1	X	X	1

注 1: 281x デバイス上では、VMAP および M0M1MAP モードは、リセット時に 1 にセットされます。ENPIE モードは、リセット時に強制的に 0 にセットされます。

注 2: ベクタ・マップ MO と M1 ベクタは、予約モード専用です。28x デバイスではこれらは RAM として使用されます。

注 3: F2812、C2812、R2812 デバイスでのみ有効です。

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際、M0 および M1 メモリ・ブロックは、RAM ブロックとして扱われ、制限なしに自由に使用できます。

デバイスをリセットした後は、ベクタ・テーブルは表 6-3 のようにマッピングされます。

表 6-3 リセット・オペレーション後のベクタ・テーブル・マッピング (注1)

ベクタ・マップ	リセットのフェッチ元	アドレス範囲	VMAP	M0M1 MAP	MP/MC	ENPIE
BROM ベクタ	ROM ブロック	0x3FFFC0-0x3FFFFFF	1	1	0	0
XINTF ベクタ (注2)	XINTF ゾーン7 ブロック	0x3FFFC0-0x3FFFFFF	1	1	1	0

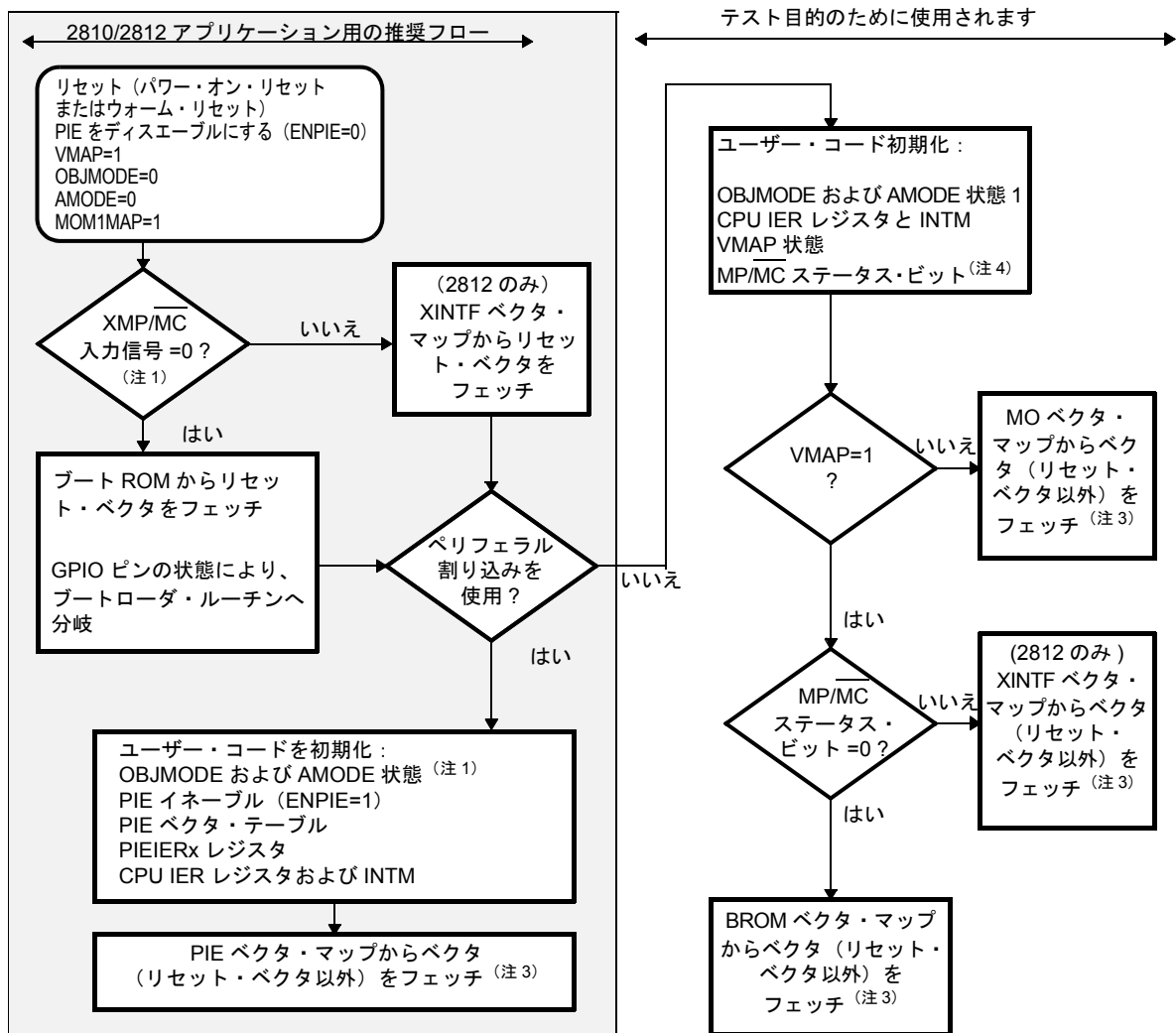
注1: 28x デバイス上では、VMAP および M0M1MAP モードは、リセット時に 1 にセットされます。ENPIE モードは、リセット時に強制的に 0 にセットされます。

注2: F2812、C2812、R2812 デバイスでのみ有効です。

リセットとブートの完了後、ユーザーのコードにより PIE ベクタ・テーブルを初期化する必要があります。次に、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。この時点より、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。ただし、リセットが行われると、リセットされたベクタは常に表 6-3 にあるベクタ・テーブルからフェッチされます。リセット後、PIE ベクタ・テーブルは常にディスエーブルになります。

図 6-3 は、ベクタ・テーブル・マッピングを選択するプロセスを示しています。

図 6-3 リセット・フロー図



注 1: XMP/MC 入力信号は、F2810 では内部で LOW 接続されています。

注 2: F2810 と F2812 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で判定します。

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)

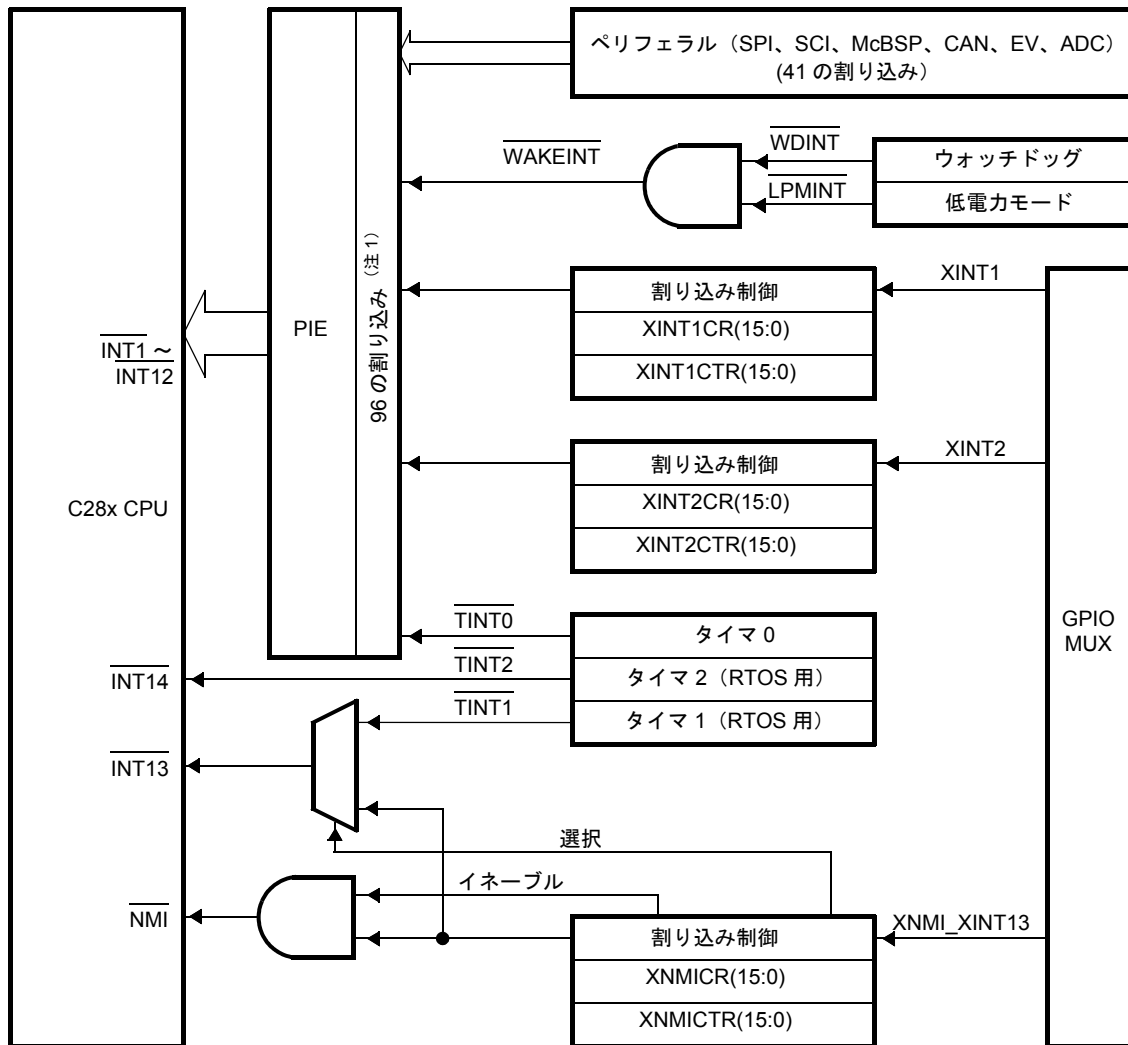
注 3: リセット・ベクタは XMP/MC 入力信号により、常に BROM または XINTF ベクタ・マップのどちらかからフェッチされます。

注 4: XMP/MC 信号の状態は、リセット時に MP/MC ビットにラッチされ、ソフトウェアによる変更が可能となります。

6.3 割り込みソース

表 6-4 は、さまざまな割り込みソースが F2810 および F2812 デバイス内でマルチプレクスされる様子を示しています。このマルチプレクス化の仕組みは、すべての F28xx デバイスで同一とは限りません。詳細については、各デバイスのデータ・シートを参照してください。

図 6-4 割り込みソース



注 1: 96 の割り込みのうち、45 の割り込みが現在ペリフェラルで使用されています。

- 注:
- 1) XINT1、XINT2、XNMI 信号は GPIO MUX 内で同期し、オプションでユーザーがプログラム可能なクロック・サイクル数でフィルタリング (Qualify) できます。これにより入力ソースのグリッチをフィルタアウトします。詳細については、GPIO MUX のセクションを参照してください。
 - 2) WAKEINT 入力を PIE ブロックへ送る前に (プロセッサの SYSCLKOUT を使用して) 同期させる必要があります。

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、PIE グループ 1 ~ 12 の 12 グループに分けられます。各グループには、そのグループに対応するイネーブル PIEIER とフラグ 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制御に使用します。PIE モジュールもまた、PIEIER および PIEIFR レジスタを使用して、CPU がどの割り込みサービス・ルーチンへ分岐すべきかを判断します。

PIEIFR および PIEIER レジスタ内のビットをクリアする際、主に以下の 3 つのルールに従う必要があります。

- 1) **PIEIFR ビットは絶対にクリアしない。** read - modify - write の操作中に、その時来ている割り込みが失われる恐れがあります。PIEIFR ビットをクリアするには、保留中の割り込みが処理される必要があります。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、以下の手順に従ってください。

ステップ 1: EALLOW ビットをセットし、PIE ベクタ・テーブルへの変更を許可します。

ステップ 2: ペリフェラルのサービス・ルーチンがテンポラリ ISR を指すように、PIE ベクタ・テーブルを変更します。このテンポラリ ISR は、割り込みオペレーションからのリターン (IRET) のみを実行します。

ステップ 3: テンポラリ ISR が割り込みを処理できるよう、割り込みをイネーブルにします。

ステップ 4: テンポラリ割り込みルーチンが処理されると、PIEIFR ビットはクリアされます。

ステップ 5: PIE ベクタ・テーブルを変更し、ペリフェラルのサービス・ルーチンを正しいサービス・ルーチンへマップし直します。

ステップ 6: EALLOW ビットをクリアします。

CPU IFR レジスタは CPU 内にあります。このビットのクリアは CPU IFR レジスタ内で実行されるため、その時来ている割り込みを失う恐れはありません。

- 2) **ソフトウェアにより優先順位を設定する割り込み。** 『C28x C/C++ Header Files and Peripheral Examples』(文書番号 SPRC097) に記載されている方法を使用します。

CPU IER レジスタをグローバル優先順位として、個別の PIEIER レジスタをグループ優先順位として使用します。この場合、PIEIER レジスタは割り込み内でのみ変更できます。また、処理される割り込みと同じグループの PIEIER のみを変更されます。この変更は PIEACK ビットが、CPU から返された追加の割り込みを保留する間に実行されます。

関係のないグループからの割り込みを処理する場合は、PIEIER ビットをディスエーブルにしないでください。

- 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た後ディスエーブルにする場合は、セクション 6.3.2 で説明す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にした後ディスエーブルにする手順

割り込みをイネーブルまたはディスエーブルにする正しい手順は、ペリフェラル割り込みのイネーブル/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、同じ割り込みグループ内でソフトウェアによる割り込み優先順位を設定することです。C28x ソフトウェア・パッケージの『C28x C/C++ Header Files and Peripheral Examples』（文書番号 SPRC097）には、ソフトウェアにより割り込みの優先順位を設定する方法の例が記載されています。この方法以外で PIEIER レジスタ内のビットを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 1 つを実行します。

1) PIEIERx レジスタを使って割り込みをディスエーブルにし、その対応する PIEIFRx フラグを保持する

PIEIFRx レジスタ内の関連フラグを保持しながら、PIEIERx レジスタ内のビットをクリアするには、以下の手順を実行します。

ステップ 1: グローバル割り込み (INTM=1) をディスエーブルにします。

ステップ 2: PIEIERx.y ビットをクリアし、特定ペリフェラルの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実行できます。

ステップ 3: 5 サイクル待ちます。この遅延が必要な理由は、CPU に入った割り込みに対し、CPU IFR レジスタの対応するフラグが立つようになるためです。

ステップ 4: 該当するペリフェラル・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オペレーションです。

ステップ 5: ペリフェラル・グループの PIEACKx ビットをクリアします。

ステップ 6: グローバル割り込み (INTM=0) をイネーブルにします。

2) PIEIERx レジスタを使って、割り込みをディスエーブルにし、その対応する PIEIFRx フラグをクリアする

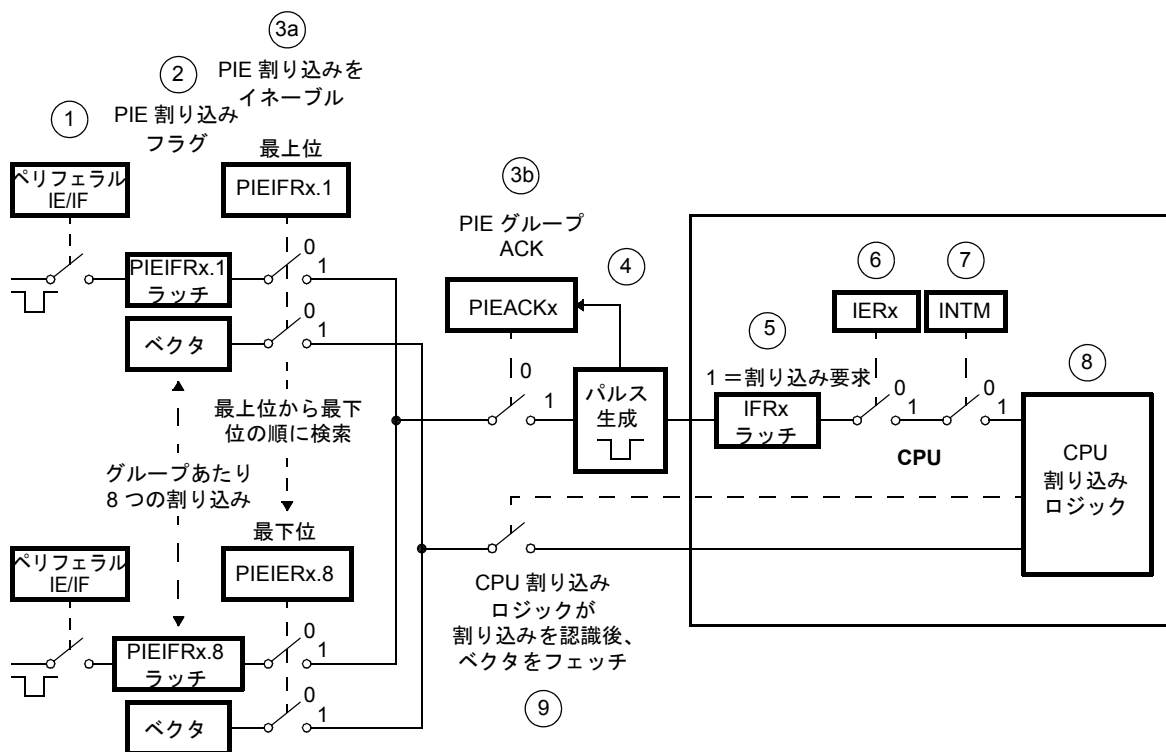
ペリフェラル割り込みのソフトウェア・リセットを実行し、PIEIFRx および CPU IFR レジスタ内の対応するフラグをクリアするには、以下の手順に従ってください。

- ステップ 1:** グローバル割り込み (INTM=1) をディスエーブルします。
- ステップ 2:** EALLOW ビットをセットします。
- ステップ 3:** PIE ベクタ・テーブルを変更し、対象となるペリフェラル割り込みのベクタを、一時的に空の割り込みサービス・ルーチン (ISR) へマッピングします。この空の ISR は割り込み命令からのリターン (IRET) のみを実行します。これは、グループ内の別のペリフェラルからの割り込みを失わずにその PIEIFRx.y ビットのみをクリアする安全な方法です。
- ステップ 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。
- ステップ 5:** グローバル割り込み (INTM=0) をイネーブルにします。
- ステップ 6:** ペリフェラルからの保留中の割り込みを空の ISR ルーチンが処理するまで待機します。
- ステップ 7:** グローバル割り込み (INTM=1) をディスエーブルにします。
- ステップ 8:** PIE ベクタ・テーブルを変更し、ペリフェラル・ベクタを元の ISR へマッピングし直します。
- ステップ 9:** EALLOW ビットをクリアします。
- ステップ 10:** 対象となるペリフェラルの PIEIER ビットをディスエーブルにします。
- ステップ 11:** 対象となるペリフェラル・グループの 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オペレーションです)。
- ステップ 12:** PIE グループの PIEACK ビットをクリアします。
- ステップ 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要求のフロー

図 6-5 では、丸で囲った番号順のステップでフローを示します。図に続いて、ステップを説明します。

図 6-5 マルチプレクスされた割り込み要求のフロー図



- ステップ 1:** PIE グループ内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。ペリフェラル・モジュール内で割り込みがイネーブルであれば、割り込み要求が PIE モジュールへ送られます。
- 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、該当する PIE 割り込みフラグ・ビットがラッチされたこと (PIEIFRx.y=1) を認識します。
- ステップ 3:** 割り込み要求が PIE から CPU へ送られるためには、以下の両方の条件が真である必要があります。
- 1) 適切なイネーブル・ビットがセットされている (PIEIERx.y=1)
 - 2) グループの PIEACKx ビットがクリアされている
- ステップ 4:** ステップ 3 で両方の条件が真である場合は、割り込み要求が CPU へ送られ、ACK ビットが再度セットされます (PIEACKx=1)。PIEACKx ビットは、グループからの追加割り込みを PIE から CPU へ送信可能であることを示すためユーザーがクリアするまでの間、セットされたままとなります。

- ステップ 5:** CPU 割り込みフラグ・ビットがセットされ (CPU IFR_x=1)、CPU レベルに保留中の割り込み x があることを示します。
- ステップ 6:** CPU 割り込みがイネーブルであり (CPU IER ビット $x=1$ 、または DBGIER ビット $x=1$)、かつ (7) グローバル割り込みマスクがクリアされている (INTM=0) 場合、CPU は INT_x を処理します。
- ステップ 7:** 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、EALLOW をクリアします。割り込みを処理する準備として CPU が実行するすべてのステップは、『TMS320C28x DSP CPU and Instruction Set Reference Guide』(文書番号 SPRU430) に記載されています。
- ステップ 8:** CPU は PIE の対応するベクタを要求します。
- ステップ 9:** マルチプレクスされた割り込みでは、PIE モジュールは、PIEIER_x および PIEIFR_x レジスタの現在値を使用して、使用するベクタ・アドレスをデコードします。
- 以下の 2 つの場合が考えられます。
- グループ内で最高の優先順位を持つ割り込みのベクタは、(a) PIEIER_x レジスタ内でイネーブルであり (b) かつ PIEIFR_x 内で保留中のフラグが立っている場合、このベクタが、フェッチされ分岐アドレスとして使用されます。このように、ステップ 4 の後、さらに高い優先順位を持つイネーブルな割り込みがフラグされている場合は、その割り込みが先に処理されます。
 - グループ内にフラグの付いた割り込みがイネーブルとなっていない場合は、PIE はそのグループ内で一番高い優先順位の割り込みを持つベクタで応答します。これは、INT_{x.1} に使用される分岐アドレスです。この動作は、28x TRAP または INT 命令に対応します。

注:

PIEIER_x レジスタは、どのベクタを分岐に使用するかの判定に使用されるため、PIEIER_x レジスタ内でビットをクリアする際には注意が必要です。PIEIER_x レジスタ内でビットをクリアする正しい手順は、セクション 6.3.2 で説明しています。この手順に従わないと、割り込みが図 6-5 の (5) で CPU へパスされた後で、PIEIER_x レジスタへの変更が行われる可能性があります。この場合、PIE は保留中でかつイネーブルとなっているその他の割り込みがない限り、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 PIEIFR_{x.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へ分岐します。

6.3.4 PIE ベクタ・テーブル

PIE ベクタ・テーブル (表 6-4 を参照) は、256 × 16 の SARAM ブロックで構成されます。このブロックは、PIE ブロックを使用しない場合は RAM として (データ空間としてのみ) 使用することもできます。PIE ベクタ・テーブルの内容は、リセット時には定義されていません。CPU の INT1 ~ INT12 の割り込み優先順位は固定です。PIE は、8 つの割り込みグループそれぞれの優先順位を制御します。たとえば、INT1.1 が INT8.1 と同時に発生した場合は、PIE ブロックは両方の割り込みを同時に CPU へ提示し、CPU は INT1.1 を先に処理します。INT1.1 が INT1.8 と同時に発生した場合は、INT1.1 が CPU へ先に送信され、その後 INT1.8 が送信されます。割り込みの優先順位付けは、割り込み処理のベクタ・フェッチ中に実行されます。

TRAP 1 ~ TRAP 12 命令または INTR INT1 ~ INTR INT12 命令は、各グループ (INTR1.1 ~ INT12.1) の先頭ロケーションからベクタをフェッチします。同じように、OR IFR の #16 ビット演算でも、各割り込みフラグがセットされている場合は、INTR1.1 ~ INTR12.1 のロケーションからのベクタがフェッチされます。その他すべての TRAP、INTR、OR IFR の #16 ビット演算は、ベクタをそれぞれのテーブルからフェッチします。INTR1 ~ INTR12 の演算ではこれらの使用を避けてください。TRAP #0 は、ベクタ値 0x000000 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 281x の PIE ベクタ・テーブル

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
Reset	0	0x0000 0D00	2	リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 内の 0x003F FFC0 ロケーションからフェッチされる	1 (最上位)	-
INT1	1	0x0000 0D02	2	使用しない。PIE グループ 1 を参照	5	-
INT2	2	0x0000 0D04	2	使用しない。PIE グループ 2 を参照	6	-
INT3	3	0x0000 0D06	2	使用しない。PIE グループ 3 を参照	7	-
INT4	4	0x0000 0D08	2	使用しない。PIE グループ 4 を参照	8	-
INT5	5	0x0000 0D0A	2	使用しない。PIE グループ 5 を参照	9	-
INT6	6	0x0000 0D0C	2	使用しない。PIE グループ 6 を参照	10	-
INT7	7	0x0000 0D0E	2	使用しない。PIE グループ 7 を参照	11	-
INT8	8	0x0000 0D10	2	使用しない。PIE グループ 8 を参照	12	-

- 注:
- 1) 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。
 - 2) ベクタ ID は DSP/BIOS で使用されます。
 - 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

表 6-4 281x の PIE ベクタ・テーブル (続き)

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
INT9	9	0x0000 0D12	2	使用しない。PIE グループ 9 を参照	13	-
INT10	10	0x0000 0D14	2	使用しない。PIE グループ 10 を参照	14	-
INT11	11	0x0000 0D16	2	使用しない。PIE グループ 11 を参照	15	-
INT12	12	0x0000 0D18	2	使用しない。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)	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	外部ノンマスクابل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	-	-
USER2	21	0x0000 0D2A	2	ユーザー定義トラップ	-	-
USER3	22	0x0000 0D2C	2	ユーザー定義トラップ	-	-
USER4	23	0x0000 0D2E	2	ユーザー定義トラップ	-	-
USER5	24	0x0000 0D30	2	ユーザー定義トラップ	-	-
USER6	25	0x0000 0D32	2	ユーザー定義トラップ	-	-
USER7	26	0x0000 0D34	2	ユーザー定義トラップ	-	-
USER8	27	0x0000 0D36	2	ユーザー定義トラップ	-	-
USER9	28	0x0000 0D38	2	ユーザー定義トラップ	-	-
USER10	29	0x0000 0D3A	2	ユーザー定義トラップ	-	-

- 注： 1) 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。
 2) ベクタ ID は DSP/BIOS で使用されます。
 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

表 6-4 281x の PIE ベクタ・テーブル (続き)

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
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 のベクタ - CPU INT1 ヘマルチプレクス化						
INT1.1	32	0x0000 0D40	2	PDPINTA (EV-A)	5	1 (最上位)
INT1.2	33	0x0000 0D42	2	PDPINTB (EV-B)	5	2
INT1.3	34	0x0000 0D44	2	予約済み	5	3
INT1.4	35	0x0000 0D46	2	XINT1	5	4
INT1.5	36	0x0000 0D48	2	XINT2	5	5
INT1.6	37	0x0000 0D4A	2	ADCINT (ADC)	5	6
INT1.7	38	0x0000 0D4C	2	TINT0 (CPU タイマ 0)	5	7
INT1.8	39	0x0000 0D4E	2	WAKEINT (LPM/WD)	5	8 (最下位)
PIE グループ 2 のベクタ - CPU INT2 ヘマルチプレクス化						
INT2.1	40	0x0000 0D50	2	CMP1INT (EV-A)	6	1 (最上位)
INT2.2	41	0x0000 0D52	2	CMP2INT (EV-A)	6	2
INT2.3	42	0x0000 0D54	2	CMP3INT (EV-A)	6	3
INT2.4	43	0x0000 0D56	2	T1PINT (EV-A)	6	4
INT2.5	44	0x0000 0D58	2	T1CINT (EV-A)	6	5
INT2.6	45	0x0000 0D5A	2	T1UFINT (EV-A)	6	6
INT2.7	46	0x0000 0D5C	2	T1OFINT (EV-A)	6	7
INT2.8	47	0x0000 0D5E	2	予約済み	6	8 (最下位)
PIE グループの 3 ベクタ - CPU INT3 ヘマルチプレクス化						
INT3.1	48	0x0000 0D60	2	T2PINT (EV-A)	7	1 (最上位)
INT3.2	49	0x0000 0D62	2	T2CINT (EV-A)	7	2

- 注: 1) 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。
 2) ベクタ ID は DSP/BIOS で使用されます。
 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

表 6-4 281x の PIE ベクタ・テーブル (続き)

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
INT3.3	50	0x0000 0D64	2	T2UFINT (EV-A)	7	3
INT3.4	51	0x0000 0D66	2	T2OFINT (EV-A)	7	4
INT3.5	52	0x0000 0D68	2	CAPINT1 (EV-A)	7	5
INT3.6	53	0x0000 0D6A	2	CAPINT2 (EV-A)	7	6
INT3.7	54	0x0000 0D6C	2	CAPINT3 (EV-A)	7	7
INT3.8	55	0x0000 0D6E	2	予約済み	7	8 (最下位)

PIE グループ 4 のベクタ - CPU INT4 ヘマルチプレクス化

INT4.1	56	0x0000 0D70	2	CMP4INT (EV-B)	8	1 (最上位)
INT4.2	57	0x0000 0D72	2	CMP5INT (EV-B)	8	2
INT4.3	58	0x0000 0D74	2	CMP6INT (EV-B)	8	3
INT4.4	59	0x0000 0D76	2	T3PINT (EV-B)	8	4
INT4.5	60	0x0000 0D78	2	T3CINT (EV-B)	8	5
INT4.6	61	0x0000 0D7A	2	T3UFINT (EV-B)	8	6
INT4.7	62	0x0000 0D7C	2	T3OFINT (EV-B)	8	7
INT4.8	63	0x0000 0D7E	2	予約済み	8	8 (最下位)

PIE グループ 5 のベクタ - CPU INT5 ヘマルチプレクス化

INT5.1	64	0x0000 0D80	2	T4PINT (EV-B)	9	1 (最上位)
INT5.2	65	0x0000 0D82	2	T4CINT (EV-B)	9	2
INT5.3	66	0x0000 0D84	2	T4UFINT (EV-B)	9	3
INT5.4	67	0x0000 0D86	2	T4OFINT (EV-B)	9	4
INT5.5	68	0x0000 0D88	2	CAPINT4 (EV-B)	9	5
INT5.6	69	0x0000 0D8A	2	CAPINT5 (EV-B)	9	6
INT5.7	70	0x0000 0D8C	2	CAPINT6 (EV-B)	9	7

- 注： 1) 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。
 2) ベクタ ID は DSP/BIOS で使用されます。
 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

表 6-4 281x の PIE ベクタ・テーブル (続き)

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
INT5.8	71	0x0000 0D8E	2	予約済み	9	8 (最下位)
PIE グループ 6 のベクタ - CPU INT6 ヘマルチプレクス化						
INT6.1	72	0x0000 0D90	2	SPIRXINTA (SPI)	10	1 (最上位)
INT6.2	73	0x0000 0D92	2	SPITXINTA (SPI)	10	2
INT6.3	74	0x0000 0D94	2	予約済み	10	3
INT6.4	75	0x0000 0D96	2	予約済み	10	4
INT6.5	76	0x0000 0D98	2	MRINT (McBSP)	10	5
INT6.6	77	0x0000 0D9A	2	MXINT (McBSP)	10	6
INT6.7	78	0x0000 0D9C	2	予約済み	10	7
INT6.8	79	0x0000 0D9E	2	予約済み	10	8 (最下位)
PIE グループ 7 のベクタ - CPU INT7 ヘマルチプレクス化						
INT7.1	80	0x0000 0DA0	2	予約済み	11	1 (最上位)
INT7.2	81	0x0000 0DA2	2	予約済み	11	2
INT7.3	82	0x0000 0DA4	2	予約済み	11	3
INT7.4	83	0x0000 0DA6	2	予約済み	11	4
INT7.5	84	0x0000 0DA8	2	予約済み	11	5
INT7.6	85	0x0000 0DAA	2	予約済み	11	6
INT7.7	86	0x0000 0DAC	2	予約済み	11	7
INT7.8	87	0x0000 0DAE	2	予約済み	11	8 (最下位)
PIE グループ 8 のベクタ - CPU INT8 ヘマルチプレクス化						
INT8.1	88	0x0000 0DB0	2	予約済み	12	1 (最上位)
INT8.2	89	0x0000 0DB2	2	予約済み	12	2
INT8.3	90	0x0000 0DB4	2	予約済み	12	3

- 注: 1) 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。
 2) ベクタ ID は DSP/BIOS で使用されます。
 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

表 6-4 281x の PIE ベクタ・テーブル (続き)

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
INT8.4	91	0x0000 0DB6	2	予約済み	12	4
INT8.5	92	0x0000 0DB8	2	予約済み	12	5
INT8.6	93	0x0000 0DBA	2	予約済み	12	6
INT8.7	94	0x0000 0DBC	2	予約済み	12	7
INT8.8	95	0x0000 0DBE	2	予約済み	12	8 (最下位)
PIE グループ 9 のベクタ - CPU INT9 ヘマルチプレクス化						
INT9.1	96	0x0000 0DC0	2	SCIRXINTA (SCI-A)	13	1 (最上位)
INT9.2	97	0x0000 0DC2	2	SCITXINTA (SCI-A)	13	2
INT9.3	98	0x0000 0DC4	2	SCIRXINTB (SCI-B)	13	3
INT9.4	99	0x0000 0DC6	2	SCITXINTB (SCI-B)	13	4
INT9.5	100	0x0000 0DC8	2	ECAN0INT (ECAN)	13	5
INT9.6	101	0x0000 0DCA	2	ECAN1INT (ECAN)	13	6
INT9.7	102	0x0000 0DCC	2	予約済み	13	7
INT9.8	103	0x0000 0DCE	2	予約済み	13	8 (最下位)
PIE グループ 10 のベクタ - CPU INT10 ヘマルチプレクス化						
INT10.1	104	0x0000 0DD0	2	予約済み	14	1 (最上位)
INT10.2	105	0x0000 0DD2	2	予約済み	14	2
INT10.3	106	0x0000 0DD4	2	予約済み	14	3
INT10.4	107	0x0000 0DD6	2	予約済み	14	4
INT10.5	108	0x0000 0DD8	2	予約済み	14	5
INT10.6	109	0x0000 0DDA	2	予約済み	14	6
INT10.7	110	0x0000 0DDC	2	予約済み	14	7
INT10.8	111	0x0000 0DDE	2	予約済み	14	8 (最下位)

- 注： 1) 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。
 2) ベクタ ID は DSP/BIOS で使用されます。
 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

表 6-4 281x の PIE ベクタ・テーブル (続き)

名前	ベクタ ID	アドレス	サイズ (× 16)	説明	CPU 優先順位	PIE グループ 優先順位
PIE グループ 11 のベクタ - CPU INT11 ヘマルチプレクス化						
INT11.1	112	0x0000 0DE0	2	予約済み	15	1 (最上位)
INT11.2	113	0x0000 0DE2	2	予約済み	15	2
INT11.3	114	0x0000 0DE4	2	予約済み	15	3
INT11.4	115	0x0000 0DE6	2	予約済み	15	4
INT11.5	116	0x0000 0DE8	2	予約済み	15	5
INT11.6	117	0x0000 0DEA	2	予約済み	15	6
INT11.7	118	0x0000 0DEC	2	予約済み	15	7
INT11.8	119	0x0000 0DEE	2	予約済み	15	8 (最下位)
PIE グループ 12 のベクタ - CPU INT12 ヘマルチプレクス化						
INT12.1	120	0x0000 0DF0	2	予約済み	16	1 (最上位)
INT12.2	121	0x0000 0DF2	2	予約済み	16	2
INT12.3	122	0x0000 0DF4	2	予約済み	16	3
INT12.4	123	0x0000 0DF6	2	予約済み	16	4
INT12.5	124	0x0000 0DF8	2	予約済み	16	5
INT12.6	125	0x0000 0DFA	2	予約済み	16	6
INT12.7	126	0x0000 0DFC	2	予約済み	16	7
INT12.8	127	0x0000 0DFE	2	予約済み	16	8 (最下位)

- 注:
- 1) 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。
 - 2) ベクタ ID は DSP/BIOS で使用されます。
 - 3) リセット・ベクタは常に、ブート ROM または XINTF ゾーン 7 のロケーション 0x003F FFC0 からフェッチされます。

PIE モジュールに接続されているペリフェラルおよび外部割り込みの割り込みグループを表 6-5 に示します。表内の各行は、特定の CPU 割り込みにマルチプレクスされている 8 つの割り込みを示しています。

表 6-5 281x PIE ペリフェラル割り込み (注1)

CPU 割り込み	PIE 割り込み							
	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1
INT1.y	WAKEINT (LPM/WD)	TINT0 (TIMER 0)	ADCINT (ADC)	XINT2	XINT1	予約済み	PDPINTB (EV-B)	PDPINTA (EV-A)
INT2.y	予約済み	T1OFINT (EV-A)	T1UFINT (EV-A)	T1CINT (EV-A)	T1PINT (EV-A)	CMP3INT (EV-A)	CMP2INT (EV-A)	CMP1INT (EV-A)
INT3.y	予約済み	CAPINT3 (EV-A)	CAPINT2 (EV-A)	CAPINT1 (EV-A)	T2OFINT (EV-A)	T2UFINT (EV-A)	T2CINT (EV-A)	T2PINT (EV-A)
INT4.y	予約済み	T3OFINT (EV-B)	T3UFINT (EV-B)	T3CINT (EV-B)	T3PINT (EV-B)	CMP6INT (EV-B)	CMP5INT (EV-B)	CMP4INT (EV-B)
INT5.y	予約済み	CAPINT6 (EV-B)	CAPINT5 (EV-B)	CAPINT4 (EV-B)	T4OFINT (EV-B)	T4UFINT (EV-B)	T4CINT (EV-B)	T4PINT (EV-B)
INT6.y	予約済み	予約済み	MXINT (McBSP)	MRINT (McBSP)	予約済み	予約済み	SPITXINTA (SPI)	SPIRXINTA (SPI)
INT7.y	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み
INT8.y	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み
INT9.y	予約済み	予約済み	ECAN1INT (CAN)	ECAN0INT (CAN)	SCITXINTB (SCI-B)	SCIRXINTB (SCI-B)	SCITXINTA (SCI-A)	SCIRXINTA (SCI-A)
INT10.y	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み
INT11.y	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み
INT12.y	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	予約済み

注 1: 96 の割り込みのうち、現在 45 の割り込みが使用されています。残りの割り込みは将来のデバイスのために予約されていますが、そのグループ内の割り込みがペリフェラルで使われていなければ、PIEIFRx レベルでイネーブルにすることでソフトウェア割り込みとして使用できます。そうでなければ、ペリフェラルからの割り込みは、PIEIFR をクリアする間に、偶然にそれらのフラグをクリアして失われるかもしれません。

要約すると、予約済みの割り込みをソフトウェア割り込みとして使用するためには、2 つの安全なケースがあります。

- 1) グループ内の割り込みがどのペリフェラルもアサートしていない場合
- 2) グループ内にペリフェラルの割り込みがアサインされていない場合 (例: PIE グループ 12)

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を表 6-6 に示します。

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 (× 16)	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

注： PIE コンフィギュレーションおよび制御レジスタは、EALLOW モードでは保護されません。PIE ベクタ・テーブルは保護されます。

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 (× 16)	説明
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
予約済み	0x0000-0CFA 0x0000-0CFF	6	予約済み

注： PIE コンフィギュレーションおよび制御レジスタは、EALLOW モードでは保護されません。PIE ベクタ・テーブルは保護されます。

6.5 PIE 割り込みレジスタ

表 6-7 PIECTRL レジスタ - アドレス CE0

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

凡例： R= リード・アクセス、W= ライト・アクセス、-0= リセット後の値

表 6-8 PIECTRL レジスタ - フィールドの説明

ビット	フィールド	説明
15-1	PIEVECT	これらのビットは、PIE ベクタ・テーブル内のフェッチされたベクタ・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、1 ~ 15 ビットのみが表示されます。ベクタ値を読み取れば、ベクタ・フェッチを生成した割り込みを判定できます。 例： If PIECTRL = 0x0d27 この場合、アドレス 0x0D26（不正操作）からベクタがフェッチされています。
0	ENPIE	PIE ブロックからのベクタ・フェッチをイネーブルにします。ENPIE が 1 にセットされているときは、PIE ベクタ・テーブルからすべてのベクタがフェッチされます。このビットが 0 にセットされているときは、PIE ブロックはディスエーブルとなり、ブート ROM または外部インターフェイスのゾーン 7 内の CPU ベクタ・テーブルからベクタがフェッチされます。PIE ブロックがディスエーブルとなっている場合も、すべての PIE ブロック・レジスタ（PIEACK、PIEIFR、PIEIER）へアクセスできます。 注： リセット・ベクタはイネーブルとなっている場合も、PIE からはフェッチされません。このベクタは XMPNMC 入力信号の状態により、常にブート ROM または XINTF ゾーン 7 からフェッチされます。

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

15	12	11	0
予約済み		PIEACKx	
R-0		R/W1C-0	

凡例： R= リード・アクセス、W1C= クリアするには 1 をライトする、-0= リセット後の値

表 6-9 PIE 割り込み ACK レジスタ (PIEACK) レジスタのフィールドの説明

ビット	フィールド	説明
15-12	予約済み	
11-0	PIEACKx	個々の割り込みビットへ 1 をライトすると、ビットがクリアされ、その PIE ブロックがイネーブルとなり、グループ割り込みに保留中の割り込みがある場合は、コアの割り込み入力へパルスを送信します。このレジスタのリードは、各グループ内で割り込みが保留中かどうかを示します。ビット 0 は INT1 を示し、ビットは 11 まであり、ビット 11 は INT12 を示します。 注： 0 のライトは無視されます。

6.5.1 PIE 割り込みフラグ・レジスタ

PIE モジュール (INT1 ~ INT12) が使用する各 CPU 割り込みあたり 1 つ、合計で 12 の PIEIFR レジスタがあります。

図 6-7 PIEIFRx レジスタ (x=1 ~ 12)



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

ビット	フィールド	説明
15-8	予約済み	
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。これらは、CPU 割り込みフラグ・レジスタと非常に似た動作をします。割り込みがアクティブなときは、それぞれのレジスタ・ビットがセットされます。割り込みが処理されたり、レジスタ・ビットに 0 がライトされると、ビットはクリアされます。このレジスタを読み、割り込みがアクティブか保留中かを判定することもできます。
6	INTx.7	
5	INTx.6	
4	INTx.5	
3	INTx.4	
2	INTx.3	
1	INTx.2	
0	INTx.1	

- 注：
- 1) 上記のすべてのレジスタのリセット値は、リセットによってセットされます。
 - 2) ハードウェアは、PIEIFR レジスタへの CPU アクセスよりも高い優先順位を持ちます。
 - 3) PIEIFR レジスタ・ビットは割り込み処理の割り込みベクタ・フェッチ中にクリアされます。

注：

PIEIFR ビットは絶対にクリアしないでください。read - modify - write の操作中に割り込みが失われる場合があります。フラグの付い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

PIE モジュール (INT1 ~ INT12) が使用する CPU 割り込みごとに 1 つ、合計で 12 の PIEIER レジスタがあります。

図 6-8 PIEIER_x レジスタ (x=1 ~ 12)



表 6-11 PIEIER_x 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	予約済み	
7	INTx.8	これらのレジスタ・ビットは、グループ内の割り込みを個別にイネーブルにし、コア割り込みイネーブル・レジスタと非常に良く似た動作をします。ビットを 1 にセットすると、各割り込み処理がイネーブルになります。ビットを 0 にセットすると、その割り込み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： 上記のすべてのレジスタのリセット値は、リセットによってセットされます。

注：

通常動作中の PIEIER ビットのクリアは、注意して行ってください。これらのビットの処理に関する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、保留中の割り込みの特定およびクリアに使用されます。IFR は、CPU レベル (INT1 ~ INT14、DLOGINT および RTOSINT) のすべてのマスカブル割り込みのフラグ・ビットを含みます。PIE がイネーブルの場合は、PIE モジュールは割り込みソースを INT1 ~ INT12 へマルチプレクスします。

マスカブル割り込みが要求された場合は、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 1 にセットされます。対応するマスク・ビットも 1 の場合は、割り込み要求は CPU へ送られ、IFR 内で対応するフラグをセットします。これは、割り込みは保留中または ACK 待機中であることを示します。

保留中の割り込みを特定するには、PUSH IFR 命令を使用し、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 命令を使用して手動で保留中の割り込みをクリアします。すべての保留中の割り込みは AND IFR #0 命令またはハードウェアのリセットによってクリアされます。

以下のイベントも IFR フラグをクリアします。

- 割り込みに対し CPU が承認したとき
- 28x デバイスがリセットされたとき

注：

- 1) IFR ビットをクリアするには、ビットに 1 ではなく 0 をライトする必要があります。
- 2) マスカブル割り込みが承認された場合、IFR ビットのみ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。アプリケーションで、制御レジスタ・フラグをクリアする必要がある場合、ソフトウェアがビットをクリアする必要があります。
- 3) INTR 命令が割り込みを要求し、対応する IFR ビットがセットされた場合、CPU はそのビットを自動でクリアしません。アプリケーションで IFR ビットのクリアが必要な場合、そのビットはソフトウェアでクリアする必要があります。
- 4) IMR および IFR レジスタは、コアレベルの割り込みに関係します。すべてのペリフェラルは、独自の割り込みマスクとフラグ・ビットをそれぞれの制御 / コンフィギュレーション・レジスタ内に保有します。一部のペリフェラル割り込みは、1 つのコア・レベルの割り込みとしてグループ化される点に注意してください。

図 6-9 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

注: R=リード・アクセス、W=ライト・アクセス、-0=リセット後の値

表 6-12 割り込みフラグ・レジスタ (IFR) のフィールドの説明

ビット	フィールド	説明
15	RTOSINT	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。 0 保留中の RTOS 割り込みはありません。 1 少なくとも 1 つの RTOS 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
14	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。 0 保留中の DLOGINT はありません。 1 少なくとも 1 つの DLOGINT 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
13	INT14	割り込み 14 フラグ。INT14 は CPU 割り込みレベル INT14 へ接続する割り込みのフラグです。 0 保留中の INT14 割り込みはありません。 1 少なくとも 1 つの INT14 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
12	INT13	割り込み 13 フラグ。INT13 は CPU 割り込みレベル INT13 へ接続する割り込みのフラグです。 0 保留中の INT13 割り込みはありません。 1 少なくとも 1 つの INT13 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
11	INT12	割り込み 12 フラグ。INT12 は CPU 割り込みレベル INT12 へ接続する割り込みのフラグです。 0 保留中の INT12 割り込みはありません。 1 少なくとも 1 つの INT12 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。

表 6-12 割り込みフラグ・レジスタ (IFR) のフィールドの説明 (続き)

ビット	フィールド	説明
10	INT11	割り込み 11 フラグ。INT11 は CPU 割り込みレベル INT11 へ接続する割り込みのフラグです。 0 保留中の INT11 割り込みはありません。 1 少なくとも 1 つの INT11 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
9	INT10	割り込み 10 フラグ。INT10 は CPU 割り込みレベル INT10 へ接続する割り込みのフラグです。 0 留中の INT10 割り込みはありません。 1 少なくとも 1 つの INT6 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
8	INT9	割り込み 9 フラグ。INT9 は CPU 割り込みレベル INT6 へ接続する割り込みのフラグです。 0 保留中の INT9 割り込みはありません。 1 少なくとも 1 つの INT9 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
7	INT8	割り込み 8 フラグ。INT8 は CPU 割り込みレベル INT6 へ接続する割り込みのフラグです。 0 保留中の INT8 割り込みはありません。 1 少なくとも 1 つの INT8 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
6	INT7	割り込み 7 フラグ。INT7 は CPU 割り込みレベル INT7 へ接続する割り込みのフラグです。 0 保留中の INT7 割り込みはありません。 1 少なくとも 1 つの INT7 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
5	INT6	割り込み 6 フラグ。INT6 は CPU 割り込みレベル INT6 へ接続する割り込みのフラグです。 0 保留中の INT6 割り込みはありません。 1 少なくとも 1 つの INT6 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
4	INT5	割り込み 5 フラグ。INT5 は CPU 割り込みレベル INT5 へ接続する割り込みのフラグです。 0 保留中の INT5 割り込みはありません。 1 少なくとも 1 つの INT5 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。

表 6-12 割り込みフラグ・レジスタ (IFR) のフィールドの説明 (続き)

ビット	フィールド	説明
3	INT4	割り込み 4 フラグ。INT4 は CPU 割り込みレベル INT4 へ接続する割り込みのフラグです。 0 保留中の INT4 割り込みはありません。 1 少なくとも 1 つの INT4 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
2	INT3	割り込み 3 フラグ。INT3 は CPU 割り込みレベル INT3 へ接続する割り込みのフラグです。 0 保留中の INT3 割り込みはありません。 1 少なくとも 1 つの INT3 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
1	INT2	割り込み 2 フラグ。INT2 は CPU 割り込みレベル INT2 へ接続する割り込みのフラグです。 0 保留中の INT2 割り込みはありません。 1 少なくとも 1 つの INT2 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。
0	INT1	割り込み 1 フラグ。INT1 は CPU 割り込みレベル INT1 へ接続する割り込みのフラグです。 0 保留中の INT1 割り込みはありません。 1 少なくとも 1 つの INT1 割り込みが保留中です。このビットに 0 をライトすると 0 へクリアされ、割り込み要求をクリアします。

6.5.4 割り込みイネーブル・レジスタ (IER) と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、すべてのマスカブル CPU 割り込みレベル (INT1 ~ INT14、RTOSINT、および DLOGINT) のイネーブル・ビットを含みます。IER には NMI も XRS も含まれません。そのため、IER はこれらの割り込みには影響を与えません。

IER を読むことで、イネーブルまたはディスエーブルにされた割り込みレベルを特定できますし、IER へライトして割り込みレベル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。割り込みレベルをイネーブルにするには、OR IER 命令を使って、対応する IER ビットを 1 にセットします。割り込みレベルをディスエーブルにするには、AND IER 命令を使って、対応する IER ビットを 0 にセットします。割り込みがディスエーブルになっている場合は、INTM ビットの値にかかわらず、承認されません。割り込みがイネーブルになっている、対応する IFR ビットが 1 で INTM ビットが 0 の場合は、承認されます。

OR IER および AND IER 命令を使って 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、ビット 15 (RTOSINT) の状態を変更してい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た場合、または INTR 命令が実行された場合は、対応する IER ビットは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た場合は、IER ビットは自動的にクリアされません。TRAP 命令割り込みにてビットをクリアする必要がある場合は、割り込みサービス・ルーチンによってクリアする必要があります。

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

IER レジスタを図 6-10 に示し、図の後でビットについて説明します。

図 6-10 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

注： R= リード・アクセス、W= ライト・アクセス、-0= リセット後の値

表 6-13 割り込みイネーブル・レジスタ (IER) のフィールドの説明

ビット	フィールド	説明
15	RTOSINT	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。 0 レベル INT6 をディスエーブルにします。 1 レベル INT6 をイネーブルにします。
14	DLOGINT	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。 0 レベル INT6 をディスエーブルにします。 1 レベル INT6 をイネーブルにします。
13	INT14	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。 0 レベル INT14 をディスエーブルにします。 1 レベル INT14 をイネーブルにします。
12	INT13	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。 0 レベル INT13 をディスエーブルにします。 1 レベル INT13 をイネーブルにします。

表 6-13 割り込みイネーブル・レジスタ (IER) のフィールドの説明 (続き)

ビット	フィールド	説明
11	INT12	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。 0 レベル INT12 をディスエーブルにします。 1 レベル INT12 をイネーブルにします。
10	INT11	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。 0 レベル INT11 をディスエーブルにします。 1 レベル INT11 をイネーブルにします。
9	INT10	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。 0 レベル INT10 をディスエーブルにします。 1 レベル INT10 をイネーブルにします。
8	INT9	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。 0 レベル INT9 をディスエーブルにします。 1 レベル INT9 をイネーブルにします。
7	INT8	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。 0 レベル INT8 をディスエーブルにします。 1 レベル INT8 をイネーブルにします。
6	INT7	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。 0 レベル INT7 をディスエーブルにします。 1 レベル INT7 をイネーブルにします。
5	INT6	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。 0 レベル INT6 をディスエーブルにします。 1 レベル INT6 をイネーブルにします。
4	INT5	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。 0 レベル INT5 をディスエーブルにします。 1 レベル INT5 をイネーブルにします。

表 6-13 割り込みイネーブル・レジスタ (IER) のフィールドの説明 (続き)

ビット	フィールド	説明
3	INT4	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。 0 レベル INT4 をディスエーブルにします。 1 レベル INT4 をイネーブルにします。
2	INT3	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。 0 レベル INT3 をディスエーブルにします。 1 レベル INT3 をイネーブルにします。
1	INT2	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。 0 レベル INT2 をディスエーブルにします。 1 レベル INT2 をイネーブルにします。
0	INT1	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。 0 レベル INT1 をディスエーブルにします。 1 レベル INT1 を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、リアルタイム・エミュレーション・モードで CPU が停止した場合のみ使用されます。DBGIER 内でイネーブルとなっている割り込みは、タイムクリティカルな割り込みとして定義されます。リアルタイム・モードで CPU が停止すると、IER 内でもイネーブルとなっているタイムクリティカルな割り込みのみが処理されます。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 と同じように、DBGIER を読めば、イネーブルまたはディスエーブルになっている割り込みを特定し、DBGIER ヘ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 にセットします。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 にセットします。DBGIER からリードするには、PUSH DBGIER 命令を、DBGIER レジスタにライトするには、POP DBGIER 命令を使用します。リセット時にはすべての DBGIER ビットは 0 にセットされます。

図 6-11 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							
7	6	5	4	3	2	1	0
INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1
R/W-0							

注: R=リード・アクセス、W=ライト・アクセス、-0=リセット後の値

表 6-14 デバッグ割り込みイネーブル・レジスタ (DBGIER) のフィールドの説明

ビット	フィールド	説明
15	RTOSINT	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。 0 レベル INT6 をディスエーブルにします。 1 レベル INT6 をイネーブルにします。
14	DLOGINT	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。 0 レベル INT6 をディスエーブルにします。 1 レベル INT6 をイネーブルにします。
13	INT14	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。 0 レベル INT14 をディスエーブルにします。 1 レベル INT14 をイネーブルにします。
12	INT13	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。 0 レベル INT13 をディスエーブルにします。 1 レベル INT13 をイネーブルにします。
11	INT12	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。 0 レベル INT12 をディスエーブルにします。 1 レベル INT12 をイネーブルにします。
10	INT11	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。 0 レベル INT11 をディスエーブルにします。 1 レベル INT11 をイネーブルにします。

表 6-14 デバッグ割り込みイネーブル・レジスタ (DBGIER) のフィールドの説明 (続き)

ビット	フィールド	説明
9	INT10	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。 0 レベル INT10 をディスエーブルにします。 1 レベル INT10 をイネーブルにします。
8	INT9	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。 0 レベル INT9 をディスエーブルにします。 1 レベル INT9 をイネーブルにします。
7	INT8	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。 0 レベル INT8 をディスエーブルにします。 1 レベル INT8 をイネーブルにします。
6	INT7	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。 0 レベル INT7 をディスエーブルにします。 1 レベル INT7 をイネーブルにします。
5	INT6	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。 0 レベル INT6 をディスエーブルにします。 1 レベル INT6 をイネーブルにします。
4	INT5	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。 0 レベル INT5 をディスエーブルにします。 1 レベル INT5 をイネーブルにします。
3	INT4	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。 0 レベル INT4 をディスエーブルにします。 1 レベル INT4 をイネーブルにします。
2	INT3	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。 0 レベル INT3 をディスエーブルにします。 1 レベル INT3 をイネーブルにします。

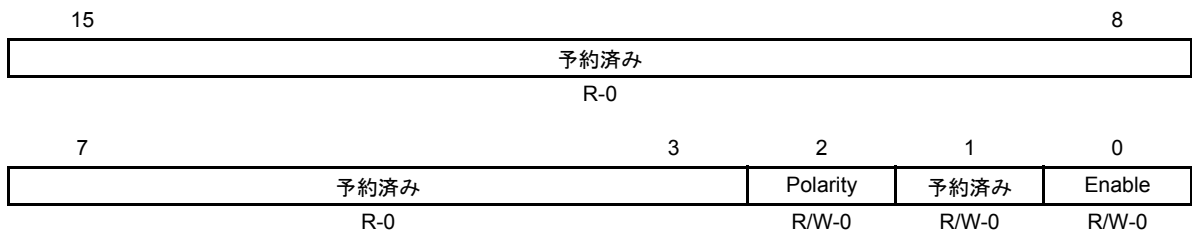
表 6-14 デバッグ割り込みイネーブル・レジスタ (DBGIER) のフィールドの説明 (続き)

ビット	フィールド	説明
1	INT2	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。 0 レベル INT2 をディスエーブルにします。 1 レベル INT2 をイネーブルにします。
0	INT1	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。 0 レベル INT1 をディスエーブルにします。 1 レベル INT1 をイネーブルにします。

6.6 外部割り込み制御レジスタ

デバイスにより、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つのノンマスクカブル割り込みXNMIとマルチプレクスされています。これらの外部割り込みはそれぞれ、トリガされた立ち下がりまたは立ち上がりのエッジに選択でき、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また、16ビットのフリー・ランニング・アップ・カウンタを含みます。このカウンタは、有効な割り込みエッジが検出されると0にリセットされます。このカウンタは、割り込みに正確なタイム・スタンプを付けるために使用できます。

図 6-12 外部割り込み1制御レジスタ (XINT1CR) - アドレス 7070h

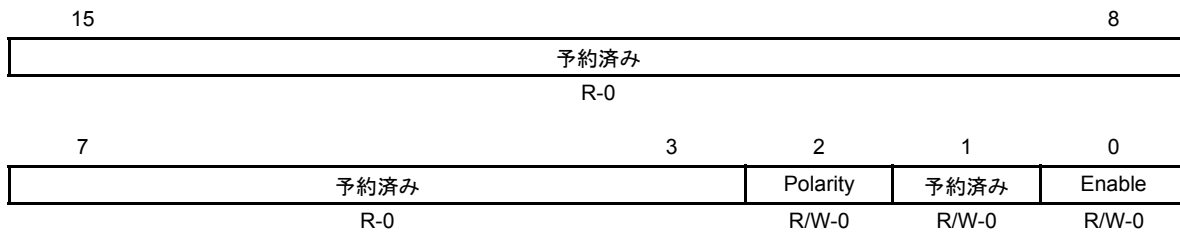


注： R=リード・アクセス、W=ライト・アクセス、-0=リセット後の値

表 6-15 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	説明
15-3	予約済み	リードは0を返し、ライトは影響を与えません。
2	Polarity	このリード/ライト・ビットは、割り込みが、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生成するかを決定します。 0 割り込みは立ち下がりエッジ (High から Low への移行部分) で生成されています。 1 割り込みは立ち上がりエッジ (Low から High への移行部分) で生成されています。
1	予約済み	リードは0を返し、ライトは影響を与えません。
0	Enable	このリード/ライト・ビットは、外部割り込み XINT1 をイネーブルまたはディスエーブルにします。 0 割り込みをディスエーブルにします。 1 割り込みをイネーブルにします。

図 6-13 外部割り込み 2 制御レジスタ (XINT2CR) - アドレス 7071h

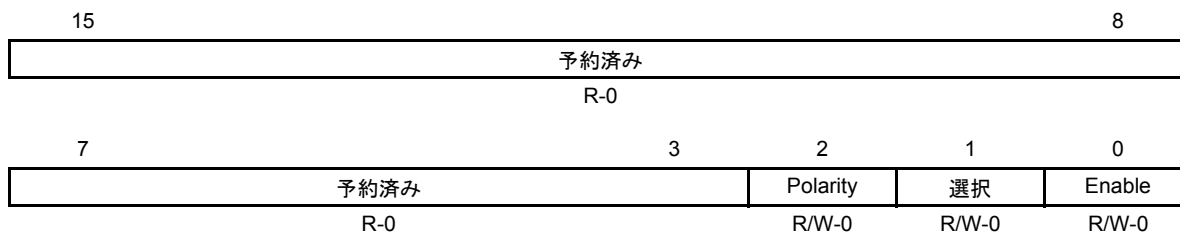


注: R=リード・アクセス、W=ライト・アクセス、-0=リセット後の値

表 6-16 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	説明
15-3	予約済み	リードは 0 を返し、ライトは影響を与えません。
2	Polarity	このリード / ライト・ビットは、割り込みが、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生成するかを決定します。 割り込みは立ち下がりエッジ (High から Low への移行部分) で生成されています。 割り込みは立ち上がりエッジ (Low から High への移行部分) で生成されています。
1	予約済み	リードは 0 を返し、ライトは影響を与えません。
0	Enable	このリード / ライト・ビットは、外部割り込み XINT2 をイネーブルまたはディスエーブルにします。 割り込みをディスエーブルにします。 割り込みをイネーブルにします。

図 6-14 外部 NMI 割り込み制御レジスタ (XNMICR) - アドレス 7077h



注: R=リード・アクセス、W=ライト・アクセス、-0=リセット後の値

表 6-17 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	説明
15-3	予約済み	リードは 0 を返し、ライトは影響を与えません。
2	Polarity	このリード / ライト・ビットは、割り込みが、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生成されるかを決定します。 0 割り込みは立ち下がりエッジ(High から Low への移行部分)上で生成されています。 1 割り込みは立ち上がりエッジ(Low から High への移行部分)上で生成されています。
1	選択	INT13 のソースを選択します。 0 INT13 に接続されているタイマ 1。 1 INT13 に接続されている XNMI_XINT13。
0	Enable	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します。 1 XNMI 割り込みをイネーブルにします。

XNMI 制御レジスタ (XNMICR) は、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ために使用できます。さらに、INT13 CPU 割り込みのソースを選択できます。図 6-4 が示すように、INT13 割り込みは、内部 CPU タイマ 1 または外部 XNMI_XINT13 信号のどちらかです。

F281x デバイス上では、CPU タイマ 1 は、TI ソフトウェア用に予約されています。しかし INT13 割り込みは、お客様が使用できる XNMI_XINT13 へも接続できます。

表 6-18 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ています。

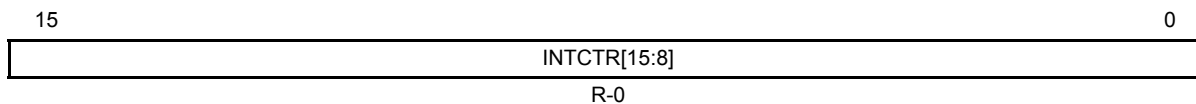
表 6-18 XNMICR レジスタのセットと割り込みソース

XNMICR	レジスタ・ビット		28xCPU 割り込み		タイム・スタンプ (XNMICR)
	ENABLE	SELECT	NMI	INT13	
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_XINT13	なし
1	0		XNMI_XINT13	CPU タイマ 1	XNMI_XINT13
1	1		XNMI_XINT13	XNMI_XINT13	XNMI_XINT13

ENABLE=1かつSELECT=1の場合、NMIとINT13 CPU割り込みの両方がXNMI_XINT13 信号に応答します。

各外部割り込みには、割り込みエッジが検出されると 0x0000 へリセットされる 16 ビットのカウンタがあります。これらのカウンタは、割り込みの発生時を正確にタイムスタンプするために使用できます。

図 6-15 外部割り込み 1 カウンタ (XINT1CTR) - アドレス 7078h

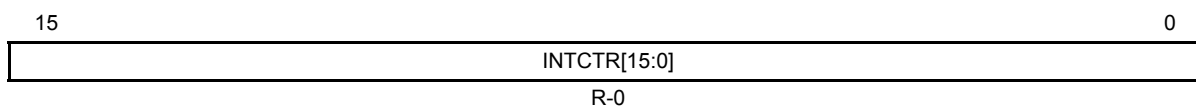


注： R= リード・アクセス、-0= リセット後の値

表 6-19 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	これはフリー・ランニングの 16 ビットのアップ・カウンタで、SYSCLKOUT レートでクロックします。このカウンタ値は、有効な割り込みエッジが検出されると 0x0000 へリセットされ、次の有効な割り込みエッジが検出されるまでの間、カウントを続けます。割り込みがディスエーブルになると、カウンタはストップします。このカウンタはフリー・ランニング・カウンタで、最大値に達すると 0 に戻ります。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、0 へリセットできます。

図 6-16 外部割り込み 2 カウンタ (XINT2CTR) - アドレス 7079h

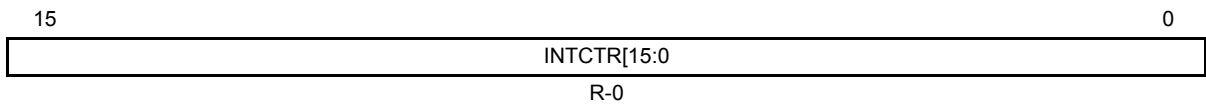


注： R= リード・アクセス、-0= リセット後の値

表 6-20 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	これはフリー・ランニングの 16 ビットのアップ・カウンタで、SYSCLKOUT レートでクロックします。このカウンタ値は、有効な割り込みエッジが検出されると 0x0000 へリセットされ、次の有効な割り込みエッジが検出されるまでの間、カウントを続けます。割り込みがディスエーブルになると、カウンタはストップします。このカウンタはフリー・ランニング・カウンタで、最大値に達すると 0 に戻ります。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、0 へリセットできます。

図 6-17 外部NMI 割り込みカウンタ (XNMICTR) - アドレス 707Fh



注： R= リード・アクセス、-0= リセット後の値

表 6-21 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	これはフリー・ランニングの 16 ビットのアップ・カウンタで、SYSCLKOUT レートでクロックします。このカウンタ値は、有効な割り込みエッジが検出されると 0x0000 へリセットされ、次の有効な割り込みエッジが検出されるまでの間、カウントを続けます。割り込みがディスエーブルになると、カウンタはストップします。このカウンタはフリー・ランニング・カウンタで、最大値に達すると 0 に戻ります。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、0 へリセットできます。

余白