

This document is a subset of chapters from the English TRM.

このリソースの元の言語は英語です。翻訳は概要を便宜的に提供するもので、自動化ツール(機械翻訳)を使用していることがあり、TI では翻訳の正確性および妥当性につきましては一切保証いたしません。実際の設計などの前には、ti.com で必ず最新の英語版をご参照くださいますようお願いいたします。

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020 – REVISED MARCH 2023



このマイクロコントローラのシステム・レベルの機能で、CPU とペリフェラルのクロック供給、リセット、割り込みのほか、オンチップ・メモリ、タイマ、セキュリティ機能の動作を構成します。

1.1 はじめに.....	4
1.2 パワー・マネージメント.....	5
1.3 デバイス識別レジスタと構成レジスタ.....	5
1.4 リセット.....	5
1.5 ペリフェラル割り込み.....	9
1.6 例外とノンマスカブル割り込み.....	21
1.7 クロック供給.....	24
1.8 32 ビット CPU タイマ 0/1/2.....	39
1.9 ウォッチドッグ・タイマ.....	41
1.10 低消費電力モード.....	44
1.11 メモリ・コントローラ・モジュール.....	47
1.12 JTAG.....	51
1.13 システム制御レジスタの構成の制限事項.....	52
1.14 ソフトウェア.....	53
1.15 システム制御レジスタ.....	59

1.1 はじめに

システム・レベルの構成は、サブモジュールのグループによって制御されます。これらのサブモジュールは、総称してシステム制御モジュールと呼ばれます。システム制御モジュールの機能は次のとおりです。

- パワーオン・リセットやブラウンアウト・リセットなどのシステム・レベルのリセット
- クロック・ソースの選択と PLL 構成
- クロック消失検出
- クロック・ゲーティング低消費電力モード
- ペリフェラル割り込み処理
- 特定のフォルト状態に対するノンマスカブル割り込み
- 3 つの 32 ビット・タイマ
- 割り込みまたはリセットを生成できるウィンドウ付きウォッチドッグ・タイマ
- RAM の初期化、書き込み保護、マスタシップの制御
- フラッシュ・メモリの ECC、ウェイト状態、キャッシュ構成
- デュアル・ゾーン・コード・セキュリティ・モジュール

1.1.1 SYSCTL 関連資料

基本資料

- 『[C2000 MCU JTAG コネクティビティのデバッグ](#)』アプリケーション・レポート

概要資料

- 『[C28x 割り込みのネスト](#)』
- 『[JTAG のデバッグ](#)』
- 『[JTAGLOCK 機能を使用したデバイスのセキュリティ強化](#)』アプリケーション・レポート
- 『[XDS ターゲット接続ガイド](#)』

エキスパート資料

- 『[C2000 CPU メモリの内蔵セルフ・テスト](#)』アプリケーション・レポート
- 『[C2000 メモリ・パワーオン・セルフ・テスト \(M-POST\)](#)』アプリケーション・レポート
- 『[TMS320C28x デバイス用の SDFlash を使用した外部不揮発性メモリのプログラミング](#)』アプリケーション・レポート
- 『[C2000 マイクロコントローラを使用したソフトウェア・フェーズ・ロック・ループ \(PLL\) 設計](#)』アプリケーション・レポート

1.1.2 システム構成レジスタのロック保護

いくつかのシステム構成レジスタは、「ロック」レジスタによって不要な CPU 書き込みから保護されています。これらの関連するロック・レジスタ・ビットをセットすると、対応するロックされたレジスタをソフトウェアで変更できなくなります。詳細については、レジスタの説明を参照してください。

1.1.3 EALLOW 保護

システム内の一部のレジスタは、EALLOW 保護メカニズムによってスプリアス CPU 書き込みから保護されています。このメカニズムでは、特別な CPU 命令 EALLOW および EDIS を使用して、保護されたレジスタへのアクセスをイネーブルおよびディセーブルにします。現在の保護状態は、表 1-1 に示すように、CPU ST1 レジスタの EALLOW ビットによって表されます。

レジスタ保護は、起動時にデフォルトでイネーブルになります。保護されている間は、保護されたレジスタへの CPU による書き込みはすべて無視されます。CPU 読み取り、JTAG 読み取り、および JTAG 書き込みのみが許可されます。EALLOW 命令を実行して保護をディセーブルにすると、CPU は保護されたレジスタに自由に書き込むことができます。レジスタを変更した後、EDIS 命令を実行して EALLOW ビットをクリアすることで、レジスタを再度保護できます。

クロック構成およびペリフェラル・クロック・イネーブル・レジスタへの書き込みは、特別なロック・レジスタに書き込むことで、次のリセットまでディセーブルにできます。

表 1-1. EALLOW 保護レジスタへのアクセス

EALLOW ビット	CPU 書き込み	CPU 読み取り	JTAG 書き込み	JTAG 読み取り
0	無視	許可 ⁽¹⁾	許可	許可
1	許可	許可	許可	許可

(1) EALLOW ビットは JTAG ポートによってオーバーライドされるため、Code Composer Studio™ インターフェイスからのデバッグ中は保護されたレジスタへのフル・アクセスが許可されます。

1.2 パワー・マネージメント

TMS320F280013x MCU 1.2V コアは、内部で LDO (VREG) によりパワーアップされます。VREG 電源は 3.3V レールから供給されるため、このデバイスは単一の 3.3V レールの動作をサポートできます。

1.3 デバイス識別レジスタと構成レジスタ

デバイス識別レジスタと構成レジスタは、デバイスの部品番号、製品ファミリ、リビジョン、ピン数、認定ステータス、および機能が利用可能かどうかに関する情報を提供します。

すべてのデバイス情報は、DEV_CFG_REGS 領域の一部です。識別レジスタは、PARTIDL、PARTIDH、および REVID です。

256 ビットの一意的 ID (UID) は、UID_REGS で利用できます。256 ビットは、次のレジスタに分割されています。

- UID_PSRAND0-4: 160 ビットの疑似ランダム・データ
- UID_UNIQUE: 64 ビットの固有データ。このレジスタの値は、同じ PARTIDH 内のすべてのデバイスで一意的になります。
- UID_CHECKSUM: UID_PSRAND0-4 と UID_UNIQUE の 32 ビットのフレッチャー・チェックサムで、出荷時のテストの際にリトル・エンディアンまたはビッグ・エンディアンとして計算されます。

1.4 リセット

このセクションでは、さまざまなリセットのタイプとこのデバイスへの影響について説明します。

1.4.1 リセット・ソース

表 1-2 に、各種のリセット信号と、デバイスへの影響をまとめます。

表 1-2. リセット信号

リセット・ソース	CPU コアのリセット (C28x, FPU, VCU)	ペリフェラルの リセット	JTAG / デバッグ・ロ ジックのリセット	IO	XRS 出力
POR	あり	あり	あり	ハイ・インピー ダンス	あり
BOR	あり	あり	あり	ハイ・インピー ダンス	あり
$\overline{\text{XRS}}$ ピン	あり	あり	なし	ハイ・インピー ダンス	-
WDRS	あり	あり	なし	ハイ・インピー ダンス	あり
NMIWDRS	あり	あり	なし	ハイ・インピー ダンス	あり
$\overline{\text{SYSRS}}$ (デバッガ・リセット)	あり	あり	なし	ハイ・インピー ダンス	なし
SCCRESET	あり	あり	なし	ハイ・インピー ダンス	なし
SIMRESET.XRS	あり	あり	なし	ハイ・インピー ダンス	あり

表 1-2. リセット信号 (continued)

リセット・ソース	CPU コアのリセット (C28x, FPU, VCU)	ペリフェラルの リセット	JTAG / デバッグ・ロ ジックのリセット	IO	XRS 出力
SIMRESET.CPU1RS	あり	あり	なし	ハイ・インピー ダンス	なし

リセットは、次の 2 つのグループに分けることができます。

- チップ・レベルのリセット ($\overline{\text{XRS}}$ 、POR、BOR、 $\overline{\text{WDRS}}$ 、SIMRESET.XRS、および $\overline{\text{NMIWDRS}}$)。デバイス全体またはほぼすべてをリセットします。
- システム・リセット ($\overline{\text{SYSRS}}$ 、SIMRESET.CPU1RS、および $\overline{\text{SCCRESET}}$)。デバイスの大規模なサブセットをリセットしますが、一部のシステム・レベルの構成は維持します。

リセット後、そのリセット要因に応じてリセット要因レジスタ (RESC) が更新されます。このレジスタのビットは、複数のリセットにわたって状態を維持します。これらのビットは、パワーオン・リセット (POR) または RESCCLR レジスタへの 1 の書き込みによってのみクリアできます。起動ルーチンの一部としてブート ROM によってクリアされるものもあります。

多くのペリフェラル・モジュールでは、SOFTPRESx レジスタから個別のリセットにアクセスできます。モジュールのリセット状態の詳細については、そのモジュールの章を参照してください。

リセット後、CPU はブート ROM にあるアドレス 0x3FFFC0 (リセット・ベクタ) から実行を開始します。ブート ROM コードを実行した後、CPU は通常、アドレス 0x80000 のフラッシュ・メモリの先頭に分岐します。ブート・プロセスの制御の詳細については、を参照してください。

注

POR の後、ブート ROM は M0/M1 および LSx RAM をクリアし、有効な ECC またはパリティが含まれていることを確認します。

1.4.2 外部リセット ($\overline{\text{XRS}}$)

外部リセット ($\overline{\text{XRS}}$) は、デバイスのメイン・チップ・レベル・リセットです。 $\overline{\text{XRS}}$ により、CPU、すべてのペリフェラルと I/O ピン構成、およびほとんどのシステム制御レジスタがリセットされます。 $\overline{\text{XRS}}$ 専用のオープン・ドレイン・ピンがあります。このピンは、アプリケーション内の他の IC のリセット・ピンを駆動するために使用でき、それ自体は外部ソースで駆動できます。 $\overline{\text{XRS}}$ はウォッチドッグ、NMI、およびパワーオン・リセット時に内部で駆動されます。

何らかの理由で $\overline{\text{XRS}}$ が Low に駆動されると、RESC レジスタの XRSn ビットがセットされます。その後、このビットはブート ROM によってクリアされます。

1.4.3 外部リセットのシミュレーション (SIMRESET.XRS)

場合によっては、ソフトウェアで外部リセット ($\overline{\text{XRS}}$) のシミュレーションを実行する必要があります。実行するには、ソフトウェアで SIMRESET レジスタの XRSn ビットを「1」に設定します。これにより、 $\overline{\text{XRS}}$ ピンがトグルされ、デバイス全体がリセットされます (外部リセットと同様)。

このリセット後、RESC レジスタの SIMRESET_XRSn ビットと XRSn ビットがセットされます。ソフトウェアはこれらのビットを読み取ってリセットの要因を把握し、RESCCLR レジスタの対応するビットに「1」を書き込むことでステータスをクリアできます。

1.4.4 パワーオン・リセット (POR)

パワーオン・リセット (POR) 回路は、電源投入時にデバイス全体にわたるクリーンなリセットを生成し、GPIO のグリッチを抑制します。 $\overline{\text{XRS}}$ ピンは POR の間、Low に保持されます。ほとんどのアプリケーションでは、 $\overline{\text{XRS}}$ が Low に保持されている間だけで他のシステム IC をリセットできますが、一部のアプリケーションではより長いパルスが必要になる場合があります。このような場合、 $\overline{\text{XRS}}$ ピンを外部から Low に駆動して、正しいリセット時間を提供できます。POR は、 $\overline{\text{XRS}}$ が実行するすべての機能と、リセット要因レジスタ (RESC)、NMI シャドウ・フラグ・レジスタ (NMISHDFLG)、X1 クロック・カウン

タ・レジスタ (X1CNT) といった他のいくつかのレジスタをリセットします。また、POR は JTAG ポートで使用されるデバッグ・ロジックもリセットします。

POR の後、RESC の POR ビットと XRSn ビットがセットされます。その後、これらのビットはブート ROM によってクリアされます。

1.4.5 ブラウンアウト・リセット (BOR)

ブラウンアウト・リセット (BOR) は、VDDIO 電源のグリッチや電源切断を監視する内部電源電圧スーパーバイザ (SVS) 回路です。VDDIO 電源電圧が動作電圧範囲を下回った場合、この回路は、障害が解消されて電源電圧が最小動作電圧に戻るまで、XRSn ピンを強制的に Low にします。BOR は、POR リセットと同じ方法ですべてをリセットします。

BOR 回路はデフォルトでイネーブルになっているため、電源投入時またはいずれかのタイプのリセット後は常にアクティブになります。BOR 回路をディセーブルにするには、VMONCTL レジスタの BORLVMONDIS ビットをセットします。

1.4.6 デバッグ・リセット ($\overline{\text{SYSRS}}$)

開発中に、デバッグの接続を切断したり、システム・レベルの構成を中断したりせずに、CPU とそのペリフェラルをリセットする必要が生じることがあります。この処理を容易にするために、CPU には独自のサブシステム・リセット機能があり、Code Composer Studio™ IDE を使用してデバッグによってトリガすることができます。このリセット ($\overline{\text{SYSRS}}$) により、CPU、そのペリフェラル、多くのシステム制御レジスタ (クロック・ゲーティングと LPM 構成を含む)、およびすべての I/O ピン構成がリセットされます。

$\overline{\text{SYSRS}}$ では、ICEPick デバッグ・モジュール、デバイス機能レジスタ、クロック・ソースおよび PLL 構成、クロック消失検出状態、PIE ベクタ・フェッチ・エラー・ハンドラ・アドレス、NMI フラグ、アナログ・トリム、または POR でのみリセットされる対象はリセットされません (セクション 1.4.4 を参照)。

1.4.7 CPU リセットのシミュレーション

場合によっては、ソフトウェアで CPU リセット ($\overline{\text{SYSRS}}$) のシミュレーションを実行する必要があります。実行するには、CPU1 ソフトウェアで SIMRESET レジスタの CPU1RSn ビットを「1」に設定します。これにより、CPU1.SYSRS 信号がトリガされ、CPU がリセットされます (デバッグ・リセットと同様)。

このリセット後、RESC レジスタの SIMRESET_CPU1RSn ビットがセットされます。ソフトウェアはこのビットを読み取ってリセットの要因を把握し、RESCCLR レジスタの対応するビットに「1」を書き込むことでステータスをクリアできます。

1.4.8 ウォッチドッグ・リセット ($\overline{\text{WDRS}}$)

デバイスにはウォッチドッグ・タイマがあり、ユーザーが指定した時間内にウォッチドッグ・タイマが CPU によって処理されない場合、オプションでリセットをトリガできます。このウォッチドッグ・リセット ($\overline{\text{WDRS}}$) により、512 INTOSC1 サイクルにわたって持続する $\overline{\text{XRS}}$ が生成されます。

ウォッチドッグ・リセット後、RESC の WDRSn ビットと XRSn ビットがセットされます。

1.4.9 NMI ウォッチドッグ・リセット ($\overline{\text{NMIWDRS}}$)

このデバイスにはノンマスカブル割り込み (NMI) モジュールがあり、システムのハードウェア・エラーを検出します。NMI モジュールにはウォッチドッグ・タイマがあり、ユーザーが指定した時間内に CPU がエラーに対処しない場合、リセットをトリガします。この NMI ウォッチドッグ・リセット ($\overline{\text{NMIWDRS}}$) により、512 INTOSC1 サイクルにわたって持続する $\overline{\text{XRS}}$ が生成されます。

NMI ウォッチドッグ・リセット後、RESC の NMIWDRSn ビットと XRSn ビットがセットされます。

1.4.10 DCSM セーフ・コード・コピー・リセット ($\overline{\text{SCCRESET}}$)

このデバイスには、フラッシュ・メモリの特定の領域への読み取りアクセスをブロックするデュアル・ゾーン・コード・セキュリティ・モジュール (DCSM) が搭載されています。CRC チェックを容易にするために、テキサス・インスツルメンツは、このようなメモリ領域に安全にアクセスするための ROM 関数を提供しています。セキュリティ侵害を防止するために、この関数を呼び出す前に割り込みをディセーブルにする必要があります。セーフ・コピーまたは CRC 関数でベクタ・フェッチが発生すると、DCSM はリセットをトリガします。このセキュリティ・リセット ($\overline{\text{SCCRESET}}$) は、 $\overline{\text{SYSRS}}$ と同様です。ただし、セキュリティ・リセットを実行すると、潜在的な攻撃者に対してアクセスを拒否するデバッグ・ロジックもリセットされます。

セキュリティ・リセット後、RESC の SCCRESETn ビットがセットされます。

1.5 ペリフェラル割り込み

このセクションでは、デバイスのペリフェラル割り込み処理について説明します。ノンマスカブル割り込みについては、[セクション 1.6](#) で説明します。ソフトウェア割り込みとエミュレーション割り込みについては、このドキュメントでは説明しません。これらの詳細については、『[TMS320C28x CPU および命令セット・リファレンス・ガイド](#)』を参照してください。

1.5.1 割り込みの概念

割り込みは、CPU の現在の実行を一時停止し、割り込みサービス・ルーチン (ISR) と呼ばれる別のコードに分岐させる信号です。これは、ペリフェラル・イベントを処理するための便利なメカニズムであり、レジスタのポーリングよりも CPU のオーバーヘッドやプログラムの複雑さを抑えることができます。ただし、割り込みはプログラム・フローと非同期であるため、割り込みとメイン・プログラム・コードの両方でアクセスされるリソースの競合を避けるように注意する必要があります。

割り込みは、一連のフラグ・レジスタとイネーブル・レジスタを経由して CPU に伝搬されます。フラグ・レジスタは、割り込みを処理するまで格納します。イネーブル・レジスタは、割り込みの伝搬をブロックします。割り込み信号が CPU に到達すると、CPU はベクタ・テーブルと呼ばれるリストから適切な ISR アドレスをフェッチします。

1.5.2 割り込みアーキテクチャ

C28x CPU には 14 本のペリフェラル割り込みラインがあります。そのうち 2 本 (INT13 および INT14) は、それぞれ CPU タイマ 1 および 2 に直接接続されています。残りの 12 本は、強化ペリフェラル割り込み拡張モジュール (ePIE、または短縮バージョンとして PIE) 経由でペリフェラル割り込み信号に接続されています。PIE は、各 CPU 割り込みラインに最大 8 本のペリフェラル割り込みを多重化します。また、ベクタ・テーブルを拡張して、各割り込みに固有の ISR を割り当てることもできます。これにより、CPU は多数のペリフェラルをサポートできます。

割り込みパスは、ペリフェラル、PIE、CPU の 3 つの段に分けられます。各段には、独自のイネーブル・レジスタとフラグ・レジスタがあります。このシステムにより、CPU は 1 つの割り込みを処理しながら他の割り込みを保留したり、ソフトウェア内でネストされた割り込みを実装および優先順位付けしたり、特定の重要なタスク中に割り込みを無効にしたりできます。

[図 1-1](#) に、このデバイスの割り込みアーキテクチャを示します。

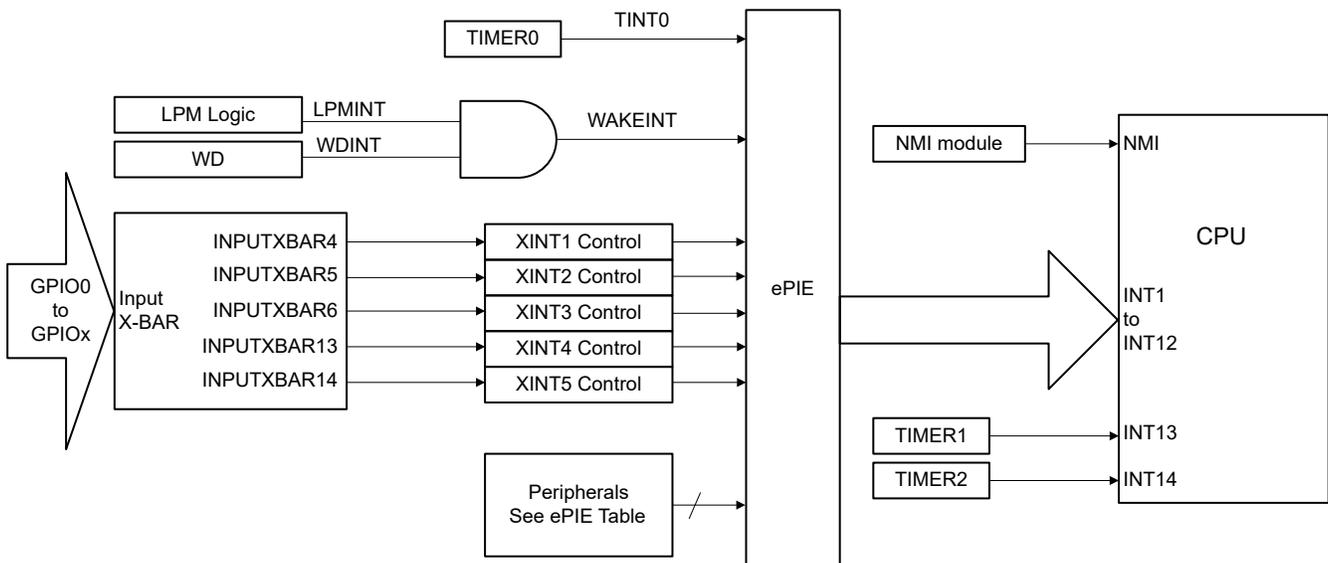


図 1-1. デバイス割り込みアーキテクチャ

1.5.2.1 ペリフェラル段

各ペリフェラルには独自の割り込み構成があり、そのペリフェラルの章で説明しています。一部のペリフェラルでは、同じ割り込み信号を複数のイベントでトリガできます。たとえば、通信ペリフェラルが同じ割り込みを使用して、データが受信されたこと、または送信エラーが発生したことを示す場合があります。割り込みの原因は、ペリフェラルのステータス・レジスタを読み取ることで特定できます。多くの場合、別の割り込みが生成される前に、ステータス・レジスタのビットを手動でクリアする必要があります。

1.5.2.2 PIE 段

PIE は、PIE チャンネルとも呼ばれる各ペリフェラル割り込み信号に対して個別のフラグ・レジスタ・ビットとイネーブル・レジスタ・ビットを提供します。これらのチャンネルは、関連する CPU 割り込みに従ってグループ化されています。各 PIE グループには、1 つの 8 ビット・イネーブル・レジスタ (PIEIERx)、1 つの 8 ビット・フラグ・レジスタ (PIEIFRx) があり、PIE アックノリッジ・レジスタ (PIEACK) の 1 ビットがあります。PIEACK レジスタ・ビットは、PIE グループ全体の共通割り込みマスクとして機能します。

CPU は割り込みを受信すると、PIE から ISR のアドレスをフェッチします。PIE は、フラグがセットされてイネーブルになっているグループ内で最も番号の小さいチャンネルのベクタを返します。このため、複数の割り込みが保留中の場合、割り込みの番号が小さいほど優先度が高くなります。

フラグがセットされてイネーブルになっている割り込みがない場合、PIE はチャンネル 1 のベクタを返します。この条件は、割り込みの伝搬中にソフトウェアで PIE の状態が変更されない限り発生しません。割り込みがイネーブルになった後で PIE 構成を安全に変更するための手順については、[セクション 1.5.4](#) を参照してください。

1.5.2.3 CPU 段

PIE と同様に、CPU は各割り込みに対してフラグ・レジスタ・ビットとイネーブル・レジスタ・ビットを提供します。イネーブル・レジスタ (IER) とフラグ・レジスタ (IFR) が 1 つずつあり、どちらも内部 CPU レジスタです。また、ST1 レジスタの INTM ビットによって制御されるグローバル割り込みマスクもあります。このマスクは、CPU の SETC 命令と CLRC 命令を使用してセットおよびクリアできます。C コードでは、C2000Ware の DINT マクロと EINT マクロをこの目的で使用できます。

IER および INTM への書き込みはアトミック操作です。特に、INTM がセットされている場合、パイプライン内の次の命令は、割り込みをディセーブルにした状態で実行されます。ソフトウェアの遅延は不要です。

1.5.3 割り込みエントリ・シーケンス

 [図 1-2](#) に、ペリフェラル割り込みが CPU にどのように伝搬されるかを示します。

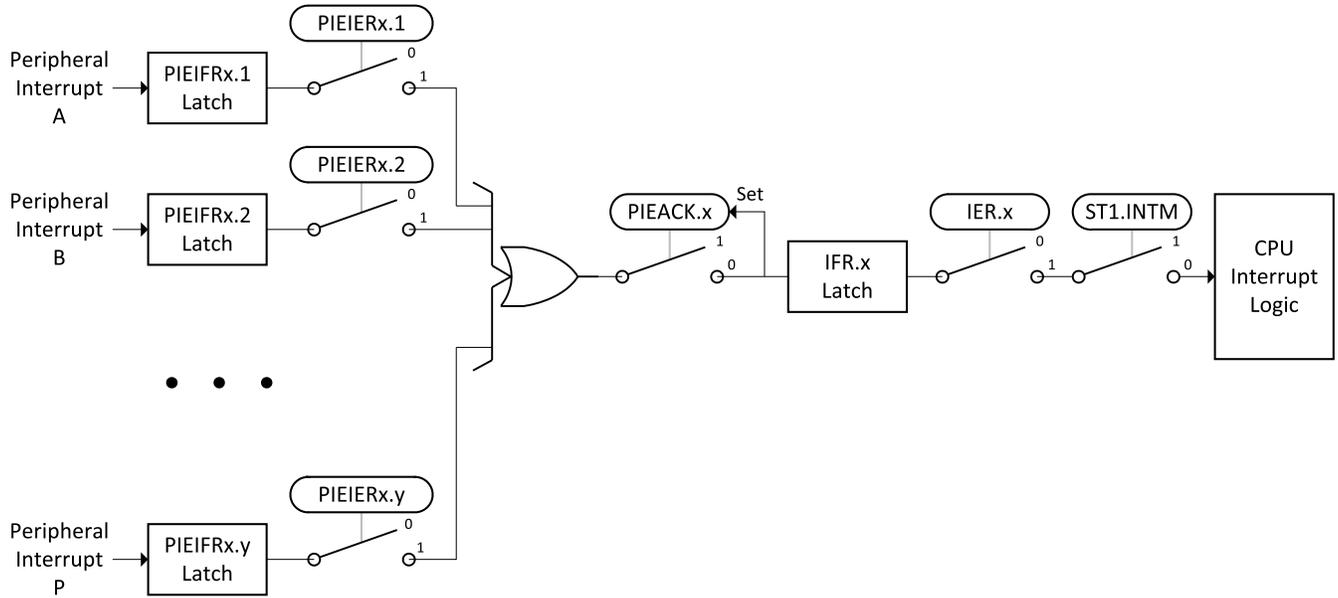


図 1-2. 割り込み伝搬パス

ペリフェラルが割り込み (PIE グループ x 、チャンネル y) を生成すると、次の一連のイベントがトリガされます。

1. 割り込みが $PIEIFRx.y$ にラッチされます。
2. $PIEIERx.y$ がセットされている場合、割り込みは伝搬されます。
3. $PIEACK.x$ がクリアされている場合、割り込みは伝搬され、 $PIEACK.x$ がセットされます。
4. 割り込みが $IFR.x$ にラッチされます。
5. $IER.x$ がセットされている場合、割り込みは伝搬されます。
6. $INTM$ がクリアされている場合、CPU は割り込みを受信します。
7. パイプラインの D2 またはそれ以降の段にあるすべての命令が完了するまで実行されます。前の段の命令はフラッシュされます。
8. CPU は、そのコンテキストをスタックに保存します。
9. $IFR.x$ と $IER.x$ がクリアされます。 $INTM$ がセットされます。 $EALLOW$ がクリアされます。
10. CPU は、PIE から ISR ベクタをフェッチします。 $PIEIFRx.y$ がクリアされます。
11. CPU は ISR に分岐します。

割り込みのレイテンシは、 $PIEIFRx.y$ が割り込みをラッチしてから、最初の ISR 命令が CPU パイプラインの実行段に入るまでの時間です。割り込みの最小レイテンシは 14 SYSCCLK サイクルです。レイテンシには、ISR またはスタック・メモリのウェイト状態が加算されます。外部割り込みにより、GPIO 同期のための少なくとも 2 SYSCCLK サイクルに加え、入力認定のための追加の時間が加算されます (使用する場合)。C28x RPT 命令を使用して作成されたループに割り込むことはできません。

1.5.4 割り込みの構成と使用

電源投入時に、割り込みはデフォルトでイネーブルにはなりません。PIEIER および IER レジスタがクリアされ、INTM がセットされます。アプリケーション・コードは、すべてのペリフェラル割り込みの構成とイネーブルを行う必要があります。

1.5.4.1 割り込みイネーブル

ペリフェラル割り込みをイネーブルにするには、次の手順を実行します。

1. 割り込みをグローバルにディセーブルにします (DINT または SETC INTM)。
2. PIECTRL レジスタの ENPIE ビットをセットして、PIE をイネーブルにします。
3. 表 1-3 に示す PIE ベクタ・テーブルの適切な場所に、各割り込みの ISR ベクタを書き込みます。ベクタ・テーブルは EALLOW 保護されていることに注意してください。
4. 各割り込みに適切な PIEIERx ビットを設定します。PIE グループとチャンネルの割り当てについては、表 1-3 を参照してください。
5. イネーブルされた割り込みを含む PIE グループの CPU IER ビットを設定します。
6. ペリフェラルで割り込みをイネーブルにします。
7. 割り込みをグローバルにイネーブルします (EINT または CLRC INTM)。

ステップ 4 は、CPU に直接接続する タイマ 1 および タイマ 2 割り込みには適用されません。

1.5.4.2 割り込みの処理

ISR は通常の間数に似ていますが、次のことを行う必要があります。

1. 特定の CPU レジスタの状態を保存および復元する (使用する場合)。
2. 割り込みグループの PIEACK ビットをクリアする。
3. IRET 命令を使用して復帰する。

要件 1 および 3 は、この関数が `__interrupt` キーワードを使用して定義されている場合、TMS320C28x C コンパイラによって自動的に処理されます。このキーワードの詳細については、『TMS320C28x C/C++ コンパイラの最適化 v6.2.4 ユーザー・ガイド』の「キーワード」セクションを参照してください。割り込みを処理するアセンブリ・コードの作成については、『TMS320C28x CPU および命令セット・リファレンス・ガイド』の「マスカブル割り込みの標準動作」セクションを参照してください。

割り込みグループの PIEACK ビットは、ユーザー・コードで手動でクリアする必要があります。これは通常、ISR の終了時に行います。PIEACK ビットがクリアされない場合、CPU はそのグループからそれ以上の割り込みを受信しません。これは、PIE を経由しない Timer1 および Timer2 割り込みには適用されません。

1.5.4.3 割り込みのディセーブル

すべての割り込みをディセーブルにするには、DINT または SETC INTM を使用して CPU のグローバル割り込みマスクをセットします。INTM をセットした後、または IER を変更した後に NOP を追加する必要はありません。次の命令は、割り込みをディセーブルにした状態で実行されます。

PIEIERx レジスタを使用して個別の割り込みをディセーブルにできますが、競合状態を避けるように注意する必要があります。PIEIER の書き込みが完了したときに割り込み信号がすでに伝搬されている場合、その信号は CPU に到達し、スプリアス割り込み条件をトリガする可能性があります。これを回避するには、次の手順を使用します。

1. 割り込みをグローバルにディセーブルにします (DINT または SETC INTM)。
2. 割り込みの PIEIER ビットをクリアします。
3. 伝搬する割り込みが CPU IFR レジスタに到達したことを確認するため、5 サイクル待ちます。
4. 割り込みの PIE グループの CPU IFR ビットをクリアします。
5. 割り込みの PIE グループの PIEACK ビットをクリアします。
6. 割り込みをグローバルにイネーブルにします (EINT または CLRC INTM)。

CPU IER レジスタを使用すると、割り込みグループをディセーブルにできます。この場合、競合状態は生じないため、特別な手順は必要ありません。

ソフトウェアで PIEIFR ビットをクリアしないでください。読み取り / 変更 / 書き込み動作により、受信割り込みが失われる可能性があるためです。PIEIFR ビットを安全にクリアする唯一の方法は、CPU が割り込みを取得することです。通常の ISR をバイパスするには、次の手順を使用します。

1. 割り込みをグローバルにディセーブルにします (DINT または SETC INTM)。
2. PIEIFR ビットの割り込みベクタを空の ISR にマップするように、PIE ベクタ・テーブルを変更します。この ISR には、割り込みからの復帰命令 (IRET) のみが含まれます。
3. ペリフェラル・レジスタで割り込みをディセーブルにします。
4. 割り込みをグローバルにイネーブルにします (EINT または CLRC INTM)。
5. 保留中の割り込みが空の ISR によって処理されるまで待ちます。
6. 割り込みをグローバルにディセーブルにします。
7. 割り込みベクタを元の ISR にマップするように、PIE ベクタ・テーブルを変更します。
8. 割り込みの PIE グループの PIEACK ビットをクリアします。
9. 割り込みをグローバルにイネーブルにします。

1.5.4.4 割り込みのネスト

デフォルトでは、割り込みはネストされません。IER レジスタと PIEIERx レジスタのソフトウェア制御により、割り込みのネストと優先度設定が可能です。C2000Ware にサンプル・コードがあり、資料は software-dl.ti.com/C2000/docs/c28x_interrupt_nesting/html/index.html で入手できます。

1.5.4.5 ベクタ・アドレスの有効性チェック

ePIE ベクタ・テーブル・メモリは、パリティ・チェックを使用して保護されます。ePIE からのベクタ・フェッチごとに、パリティ・チェックが実行されます。ベクタ・フェッチ中にパリティ・エラーが発生した場合、ePIE はユーザー定義のエラー・ハンドラルーチン (PIEVERRADDR が 0x003FFFFFF 以外の値で定義されている場合)、またはアドレス 0x3FFFBE のデフォルトのブート ROM ハンドラを返します。また、ePIE は EPWM にトリップ信号を送信します。

パリティ・チェックでは、ベクタ・フェッチ中にエラーが発生した場合、エラー・ハンドラの値のみが返されます。データ読み取り中のパリティ・エラーはメモリ・コントローラ・モジュールによって処理され、MEMORY_ERROR_REGS の UCERRFLG レジスタによってログに記録されます。エラーの原因となったアドレスは、UCCPUREADDR レジスタにあります。ログに記録されたエラー・アドレスが 0xD00~0xDFF の範囲内である場合、これは PIE パリティ・エラーです。さらに、ベクタ・フェッチ中のパリティ・エラーでは、修正不可能なエラーの NMI フラグはセットされません。

1.5.5 PIE チャンネル割り当て

表 1-3 に、各ペリフェラル割り込みの PIE グループとチャンネルの割り当てを示します。各行はグループであり、各列はそのグループ内のチャンネルです。複数の割り込みが保留されている場合、最も番号の小さいグループの最も番号の小さいチャンネルに対して、最初にサービスが提供されます。したがって、表の一番上にある割り込みは最も優先度が高く、一番下にある割り込みは最も優先度が低くなります。

表 1-3. PIE チャンネル割り当て

	INTx.1	INTx.2	INTx.3	INTx.4	INTx.5	INTx.6	INTx.7	INTx.8
INT1.y	ADCA1	ADCC1	-	XINT1	XINT2	SYS_ERR	TIMER0	WAKE
INT2.y	EPWM1_TZ	EPWM2_TZ	EPWM3_TZ	EPWM4_TZ	EPWM5_TZ	EPWM6_TZ	EPWM7_TZ	-
INT3.y	EPWM1	EPWM2	EPWM3	EPWM4	EPWM5	EPWM6	EPWM7	-
INT4.y	ECAP1	ECAP2	-	-	-	-	-	-
INT5.y	EQEP1	-	-	-	-	-	-	-
INT6.y	SPIA_RX	SPIA_TX	-	-	-	-	DCC0	-
INT7.y	-	-	-	-	-	-	-	-
INT8.y	I2CA	I2CA_FIFO	I2CB	I2CB_FIFO	SCIC_RX	SCIC_TX	-	-
INT9.y	SCIA_RX	SCIA_TX	SCIB_RX	SCIB_TX	CANA_0	CANA_1	-	-
INT10.y	ADCA_EVT	ADCA2	ADCA3	ADCA4	ADCC_EVT	ADCC2	ADCC3	ADCC4
INT11.y	-	-	-	-	-	-	-	-
INT12.y	XINT3	XINT4	XINT5	-	FLSS_INT	-	-	-

1.5.6 PIE 割り込み優先度

1.5.6.1 チャンネルの優先度

すべての PIE グループで、グループ内のチャンネルの番号が小さいほど優先度が高くなります。たとえば、PIE グループ 1 では、チャンネル 1.1 がチャンネル 1.3 よりも優先されます。これら 2 つのイネーブルな割り込みが同時に発生した場合、チャンネル 1.1 が最初に処理され、チャンネル 1.3 は保留のままになります。チャンネル 1.1 の ISR が完了した後、PIE グループ 1 に他のイネーブルな保留中の割り込みが存在しない場合、チャンネル 1.3 が処理されます。ただし、CPU が PIE グループからの割り込みをさらに処理するには、そのグループの PIEACK をクリアする必要があります。この例で、チャンネル 1.3 を処理するには、チャンネル 1.1 の ISR がグループ 1 の PIEACK をクリアする必要があります。

次の例では、別のシナリオについて説明します。チャンネル 1.1 は現在 CPU によって処理されており、チャンネル 1.3 は保留中で、チャンネル 1.1 の ISR が完了する前に、イネーブルになっているチャンネル 1.2 も受信したとします。チャンネル 1.2 の優先度はチャンネル 1.3 より高いため、CPU はチャンネル 1.2 を処理し、チャンネル 1.3 は引き続き保留のままになります。割り込みエントリ・シーケンス (セクション 1.5.3) のステップを使用すると、チャンネル 1.2 の割り込みはステップ 10 (CPU は、PIE から ISR ベクタをフェッチします。PIEIFRx.y がクリアされます) になってようやく発生しますが、チャンネル 1.3 よりも先に処理されます。

1.5.6.2 グループの優先度

一般に、優先度が最も高くなるのは、番号が最も小さい PIE グループ内で番号が最も小さいチャンネルです。チャンネル 1.1 とチャンネル 2.1 を例とします。これら 2 つのチャンネルの優先度は、それぞれのグループで最も高くなります。これら 2 つのイネーブルなチャンネルの割り込みが同時に発生し、他のイネーブルな保留中の割り込みが存在しない場合、チャンネル 1.1 が CPU によって最初に処理され、チャンネル 2.1 は保留のままになります。

ただし、チャンネルの優先度がグループの優先度より優先される場合もあります。この特殊なケースの発生は、CPU が現在割り込みエントリ・シーケンス (セクション 1.5.3) のどのステップにあるかによって変わってきます。

以下に、この特殊なケースの例を示します。

CPU はチャンネル 2.3 を処理しようとしており、現在は割り込みエントリ・シーケンス (セクション 1.5.3) のステップを実行しています。

1. CPU がステップ 10 (CPU は、PIE から ISR ベクタをフェッチします。PIEIFRx.y がクリアされます) に達すると、2 つのイネーブルな割り込みのチャンネル 1.1 とチャンネル 2.1 を受信します。
2. チャンネルの優先度により、チャンネル 2.1 はチャンネル 2.3 よりも先に処理されます。ただし、グループの優先度により、チャンネル 1.1 はチャンネル 2.1 および 2.3 よりも先に処理されることが決まっています。
3. ここではチャンネルの優先度が優先され、チャンネル 2.1 は 1.1 および 2.3 よりも先に処理されます。
4. チャンネル 2.1 が完了した後、チャンネル 1.1 に続いてチャンネル 2.3 が処理されます。

グループの優先度は、割り込みが現在処理されていない場合、つまり割り込みエントリ・シーケンス (セクション 1.5.3) が実行されていない場合にのみ保証されます。

1.5.7 システム・エラー

SYS_ERR は、複数の割り込みソースを統合します (図 1-3 を参照)。これらのソースは、**SYS_ERR_INT_FLG** レジスタのそれぞれのビットをセットします。**SYS_ERR_INT_FLG** レジスタのビットがセットされると、グローバル割り込み (**GINT**) ビットもセットされます。**SYS_ERR** 割り込みが生成される前に、**GINT** ビットをクリアする必要があります。ソース・フラグがセットされた状態で **GINT** ビットをクリアすると、別の **SYS_ERR** 割り込みが起動します。したがって、**GINT** ビットをクリアする前にソース・フラグをクリアすることを推奨します。

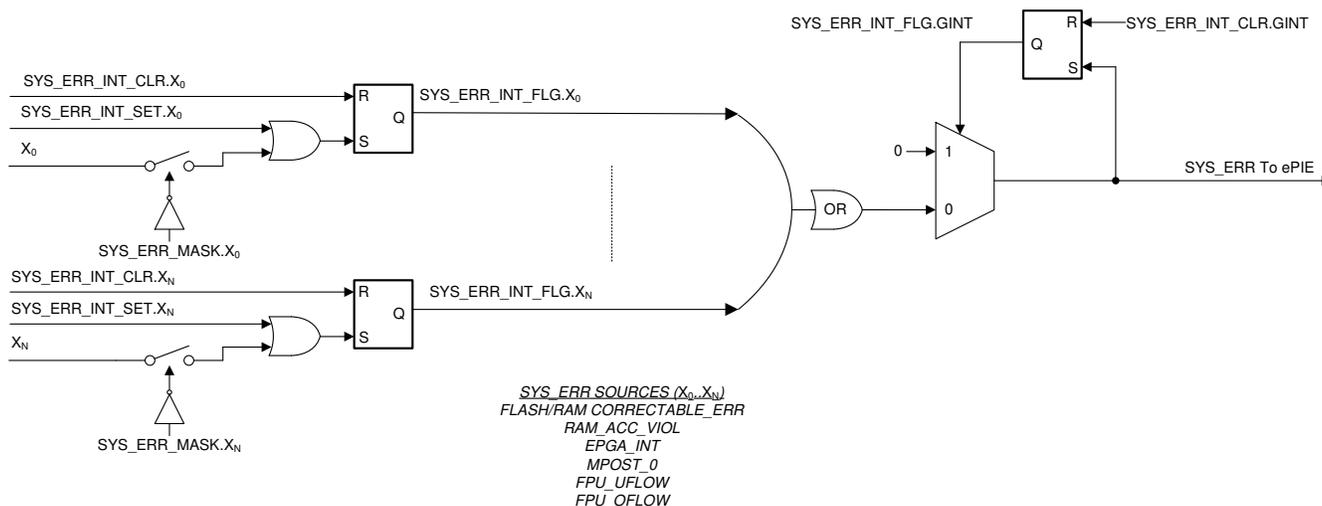


図 1-3. システム・エラー

1.5.8 ベクタ・テーブル

表 1-4 に CPU 割り込みベクタ・テーブルを示します。このデバイスでは、INT1～INT12 のベクタは使われていません。リセット・ベクタは、このテーブルからではなく、ブート ROM からフェッチされます。すべてのベクタは、EALLOW 保護されています。

表 1-5 に PIE ベクタ・テーブルを示します。

表 1-4. CPU 割り込みベクタ

名称	ベクタ ID	アドレス	サイズ (x16)	概要	コアの優先順位	ePIE グループの優先順位
リセット	0	0x0000 0D00	2	リセットは、ブート ROM の 0x003F_FFC0 の位置から常にフェッチされます	1 (最高)	-
INT1	1	0x0000 0D02	2	未使用。PIE Group 1 参照	5	-
INT2	2	0x0000 0D04	2	未使用。PIE Group 2 参照	6	-
INT3	3	0x0000 0D06	2	未使用。PIE Group 3 参照	7	-
INT4	4	0x0000 0D08	2	未使用。PIE Group 4 参照	8	-
INT5	5	0x0000 0D0A	2	未使用。PIE Group 5 参照	9	-
INT6	6	0x0000 0D0C	2	未使用。PIE Group 6 参照	10	-
INT7	7	0x0000 0D0E	2	未使用。PIE Group 7 参照	11	-
INT8	8	0x0000 0D10	2	未使用。PIE Group 8 参照	12	-
INT9	9	0x0000 0D12	2	未使用。PIE Group 9 参照	13	-
INT10	10	0x0000 0D14	2	未使用。PIE Group 10 参照	14	-
INT11	11	0x0000 0D16	2	未使用。PIE Group 11 参照	15	-
INT12	12	0x0000 0D18	2	未使用。PIE Group 12 参照	16	-
INT13	13	0x0000 0D1A	2	CPU TIMER1 割り込み	17	-
INT14	14	0x0000 0D1C	2	CPU TIMER2 割り込み	18	-
DATALOG	15	0x0000 0D1E	2	CPU データ・ロギング割り込み	19 (最低)	-
RTOSINT	16	0x0000 0D20	2	CPU リアルタイム OS 割り込み	4	-
RSVD	17	0x0000 0D22	2	予約済み	2	-
NMI	18	0x0000 0D24	2	ノンマスクابل割り込み	3	-
ILLEGAL	19	0x0000 0D26	2	不正命令 (ITRAP)	-	-
USER 1	20	0x0000 0D28	2	ユーザー定義トラップ	-	-
USER 2	21	0x0000 0D2A	2	ユーザー定義トラップ	-	-
USER 3	22	0x0000 0D2C	2	ユーザー定義トラップ	-	-
USER 4	23	0x0000 0D2E	2	ユーザー定義トラップ	-	-
USER 5	24	0x0000 0D30	2	ユーザー定義トラップ	-	-
USER 6	25	0x0000 0D32	2	ユーザー定義トラップ	-	-
USER 7	26	0x0000 0D34	2	ユーザー定義トラップ	-	-
USER 8	27	0x0000 0D36	2	ユーザー定義トラップ	-	-
USER 9	28	0x0000 0D38	2	ユーザー定義トラップ	-	-
USER 10	29	0x0000 0D3A	2	ユーザー定義トラップ	-	-
USER 11	30	0x0000 0D3C	2	ユーザー定義トラップ	-	-
USER 12	31	0x0000 0D3E	2	ユーザー定義トラップ	-	-

表 1-5. PIE 割り込みベクタ

名称	ベクタ ID	アドレス	サイズ (x16)	概要	コアの優先順位	ePIE グループの優先順位
PIE グループ 1 ベクタ - CPU INT1 に多重化						
INT1.1	32	0x0000 0D40	2	ADCA1 割り込み	5	1 (最高)
INT1.2	33	0x0000 0D42	2	ADCC1 割り込み	5	2
INT1.3	34	0x0000 0D44	2	予約済み	5	3
INT1.4	35	0x0000 0D46	2	XINT1 割り込み	5	4
INT1.5	36	0x0000 0D48	2	XINT2 割り込み	5	5
INT1.6	37	0x0000 0D4A	2	SYS_ERR 割り込み	5	6
INT1.7	38	0x0000 0D4C	2	TIMER0 割り込み	5	7
INT1.8	39	0x0000 0D4E	2	WAKE 割り込み	5	8 (最低)
PIE グループ 2 ベクタ - CPU INT2 に多重化						
INT2.1	40	0x0000 0D50	2	EPWM1 トリップ・ゾーン割り込み	6	1 (最高)
INT2.2	41	0x0000 0D52	2	EPWM2 トリップ・ゾーン割り込み	6	2
INT2.3	42	0x0000 0D54	2	EPWM3 トリップ・ゾーン割り込み	6	3
INT2.4	43	0x0000 0D56	2	EPWM4 トリップ・ゾーン割り込み	6	4
INT2.5	44	0x0000 0D58	2	EPWM5 トリップ・ゾーン割り込み	6	5
INT2.6	45	0x0000 0D5A	2	EPWM6 トリップ・ゾーン割り込み	6	6
INT2.7	46	0x0000 0D5C	2	EPWM7 トリップ・ゾーン割り込み	6	7
INT2.8	47	0x0000 0D5E	2	予約済み	6	8 (最低)
PIE グループ 3 ベクタ - CPU INT3 に多重化						
INT3.1	48	0x0000 0D60	2	EPWM1 割り込み	7	1 (最高)
INT3.2	49	0x0000 0D62	2	EPWM2 割り込み	7	2
INT3.3	50	0x0000 0D64	2	EPWM3 割り込み	7	3
INT3.4	51	0x0000 0D66	2	EPWM4 割り込み	7	4
INT3.5	52	0x0000 0D68	2	EPWM5 割り込み	7	5
INT3.6	53	0x0000 0D6A	2	EPWM6 割り込み	7	6
INT3.7	54	0x0000 0D6C	2	EPWM7 割り込み	7	7
INT3.8	55	0x0000 0D6E	2	予約済み	7	8 (最低)
PIE グループ 4 ベクタ - CPU INT4 に多重化						
INT4.1	56	0x0000 0D70	2	ECAP1 割り込み	8	1 (最高)
INT4.2	57	0x0000 0D72	2	ECAP2 割り込み	8	2
INT4.3	58	0x0000 0D74	2	予約済み	8	3
INT4.4	59	0x0000 0D76	2	予約済み	8	4
INT4.5	60	0x0000 0D78	2	予約済み	8	5
INT4.6	61	0x0000 0D7A	2	予約済み	8	6
INT4.7	62	0x0000 0D7C	2	予約済み	8	7
INT4.8	63	0x0000 0D7E	2	予約済み	8	8 (最低)

表 1-5. PIE 割り込みベクタ (continued)

名称	ベクタ ID	アドレス	サイズ (x16)	概要	コアの優先順位	ePIE グループの優先順位
PIE グループ 5 ベクタ - CPU INT5 に多重化						
INT5.1	64	0x0000 0D80	2	EQEP1 割り込み	9	1 (最高)
INT5.2	65	0x0000 0D82	2	予約済み	9	2
INT5.3	66	0x0000 0D84	2	予約済み	9	3
INT5.4	67	0x0000 0D86	2	予約済み	9	4
INT5.5	68	0x0000 0D88	2	予約済み	9	5
INT5.6	69	0x0000 0D8A	2	予約済み	9	6
INT5.7	70	0x0000 0D8C	2	予約済み	9	7
INT5.8	71	0x0000 0D8E	2	予約済み	9	8 (最低)
PIE グループ 6 ベクタ - CPU INT6 に多重化						
INT6.1	72	0x0000 0D90	2	SPIA RX 割り込み	10	1 (最高)
INT6.2	73	0x0000 0D92	2	SPIA TX 割り込み	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	予約済み	10	5
INT6.6	77	0x0000 0D9A	2	予約済み	10	6
INT6.7	78	0x0000 0D9C	2	DCC0 割り込み	10	7
INT6.8	79	0x0000 0D9E	2	予約済み	10	8 (最低)
PIE グループ 7 ベクタ - CPU INT7 に多重化						
INT7.1	80	0x0000 0DA0	2	予約済み	11	1 (最高)
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 (最低)
PIE グループ 8 ベクタ - CPU INT8 に多重化						
INT8.1	88	0x0000 0DB0	2	I2CA 割り込み	12	1 (最高)
INT8.2	89	0x0000 0DB2	2	I2CA FIFO 割り込み	12	2
INT8.3	90	0x0000 0DB4	2	I2CB 割り込み	12	3
INT8.4	91	0x0000 0DB6	2	I2CB FIFO 割り込み	12	4
INT8.5	92	0x0000 0DB8	2	SCIC RX 割り込み	12	5
INT8.6	93	0x0000 0DBA	2	SCIC TX 割り込み	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 (最低)

表 1-5. PIE 割り込みベクタ (continued)

名称	ベクタ ID	アドレス	サイズ (x16)	概要	コアの優先順位	ePIE グループの優先順位
PIE グループ 9 ベクタ - CPU INT9 に多重化						
INT9.1	96	0x0000 0DC0	2	SCIA RX 割り込み	13	1 (最高)
INT9.2	97	0x0000 0DC2	2	SCIA TX 割り込み	13	2
INT9.3	98	0x0000 0DC4	2	SCIB RX 割り込み	13	3
INT9.4	99	0x0000 0DC6	2	SCIB TX 割り込み	13	4
INT9.5	100	0x0000 0DC8	2	DCANA 割り込み 1	13	5
INT9.6	101	0x0000 0DCA	2	DCANA 割り込み 2	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 (最低)
PIE グループ 10 ベクタ - CPU INT10 に多重化						
INT10.1	104	0x0000 0DD0	2	ADCA イベント割り込み	14	1 (最高)
INT10.2	105	0x0000 0DD2	2	ADCA2 割り込み	14	2
INT10.3	106	0x0000 0DD4	2	ADCA3 割り込み	14	3
INT10.4	107	0x0000 0DD6	2	ADCA4 割り込み	14	4
INT10.5	108	0x0000 0DD8	2	ADCC イベント割り込み	14	5
INT10.6	109	0x0000 0DDA	2	ADCC2 割り込み	14	6
INT10.7	110	0x0000 0DDC	2	ADCC3 割り込み	14	7
INT10.8	111	0x0000 0DDE	2	ADCC4	14	8 (最低)
PIE グループ 11 ベクタ - CPU INT11 に多重化						
INT11.1	112	0x0000 0DE0	2	予約済み	15	1 (最高)
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 (最低)
PIE グループ 12 ベクタ - CPU INT12 に多重化						
INT12.1	120	0x0000 0DF0	2	XINT3 割り込み	16	1 (最高)
INT12.2	121	0x0000 0DF2	2	XINT4 割り込み	16	2
INT12.3	122	0x0000 0DF4	2	XINT5 割り込み	16	3
INT12.4	123	0x0000 0DF6	2	予約済み	16	4
INT12.5	124	0x0000 0DF8	2	FLSS_INT 割り込み	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 (最低)

1.6 例外とノンマスカブル割り込み

このセクションでは、ノンマスカブル割り込み (NMI) をトリガできるシステム・レベルのエラー条件について説明します。割り込みにより、アプリケーションはエラーに対処できます。

1.6.1 NMI の構成と使用

受信した NMI は、NMIFLG レジスタのステータス・ビットをセットし、NMI ウォッチドッグ・カウンタを開始します。このカウンタには SYSCLK からクロックが供給され、カウンタが NMIWDPRD レジスタの値に達すると、NMI ウォッチドッグ・リセット (NMIWDRS) がトリガされます。これを防止するために、NMI ハンドラは NMIFLGCLR レジスタを使用してフラグ・ビットをクリアする必要があります。すべてのフラグ・ビットがクリアされると、今後の NMI を取得できるように、NMIFLG レジスタの NMIINT ビットもクリアされる場合があります。

NMI モジュールは、起動プロセス中にブート ROM によってイネーブルになります。NMI に応答するには、NMI ハンドラ・ベクタを PIE ベクタ・テーブルに書き込む必要があります。

1.6.2 エミュレーションの考慮事項

NMI ウォッチドッグ・カウンタは、各デバッグ条件で次のように動作します。

CPU の一時停止	CPU が一時停止すると、NMI ウォッチドッグ・カウンタは一時停止します。
ランプリー・モード	CPU がランプリー・モードになると、NMI ウォッチドッグ・カウンタ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・モードのとき、NMI ウォッチドッグ・カウンタは一時停止します。カウンタは、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・ランプリー・モード	CPU がリアルタイム・ランプリー・モードのとき、NMI ウォッチドッグ・カウンタは通常どおり動作します。

1.6.3 NMI ソース

NMI をトリガできるハードウェア・エラーにはいくつかのタイプがあります。エラーに関する追加情報は通常、エラーを検出するモジュールから入手できます。

1.6.3.1 クロック消失検出

クロック消失検出ロジックは、OSCCLK の障害を監視します。OSCCLK ソースが停止すると、PLL がバイパスされ、OSCCLK が INTOSC1 に接続されて、CPU に対して NMI が起動されます。クロック消失検出の詳細については、[セクション 1.7.13.1](#) を参照してください。

1.6.3.2 RAM の修正不可能なエラー

RAM の読み取り時に 1 ビット・パリティ・エラー、2 ビット ECC データ・エラー、または 1 ビット ECC アドレス・エラーが発生すると、NMI がトリガされます。これは CPU 読み取りに適用されます。1 ビット ECC データ・エラーでは NMI はトリガされませんが、オプションで通常のペリフェラル割り込みをトリガすることもできます。RAM エラー検出の詳細については、[セクション 1.11.1.4](#) を参照してください。

1.6.3.3 フラッシュの修正不可能な ECC エラー

フラッシュの読み取り時に 2 ビット ECC データ・エラーまたは 1 ビット ECC アドレス・エラーが発生すると、NMI がトリガされます。1 ビット ECC データ・エラーでは NMI はトリガされませんが、オプションで通常のペリフェラル割り込みをトリガすることもできます。

1.6.3.4 ソフトウェア強制エラー

NMIFLGFRC レジスタの SWERR ビットへの書き込みによってのみトリガできる特別な NMI ソースがあります。SWERR フラグは実際のハードウェア障害によってセットされることはないため、NMI サブシステム用のセルフ・テスト・モードを実装するために使用できます。

1.6.4 不正命令トラップ (ITRAP)

CPU が不正な命令を実行しようとする、不正命令トラップ (ITRAP) と呼ばれる特別な割り込みが生成されます。この割り込みはノンマスカブルであり、PIE ベクタ・テーブルにベクタを持ちます。ITRAP の詳細については、『[TMS320C28x DSP CPU および命令セット・リファレンス・ガイド](#)』の「不正命令トラップ」セクションを参照してください。

注

RAM フェッチ・アクセス違反が発生すると、RAM アクセス違反に対する通常のペリフェラル割り込みに加えて、ITRAP がトリガされます。CPU は、ITRAP を最初に処理します。

1.6.5 ERRORSTS ピン

ERRORSTS ピンは「常時出力」ピンで、チップ内部でエラーが検出されるまで High に維持されます。エラーが発生すると、そのエラー・ソースに対応する内部エラー・ステータス・フラグがクリアされるまで、ERRORSTS ピンは Low (デフォルトの極性) になります。図 1-4 に、ERRORSTS ピンの機能を示します。

ERRORSTS ピンは、チップの電源レールが動作下限まで上昇するまでトライステートになります。ERRORSTS ピンはアクティブ Low ピン (デフォルトの極性) なので、電源投入時のこのピンの状態を考慮するユーザーは、このピンに外部プルダウンを接続できます。

このデバイスでは、ERRORSTS ピン・ロジックに対して次のような拡張が行われています。

- エラー・ピンの極性を ERRORCTL レジスタで設定できるようになりました (デフォルト設定はアクティブ Low 極性)。
- エラー・ピンのテストをイネーブルにするため、ソフトウェアからエラー・ピンを強制およびクリアする機能が導入されました。
- ERRORSTS にエラー・ソースが追加されました。
 - CPU1 ウォッチドッグ・リセット
 - PIE ベクタ・フェッチでのエラー

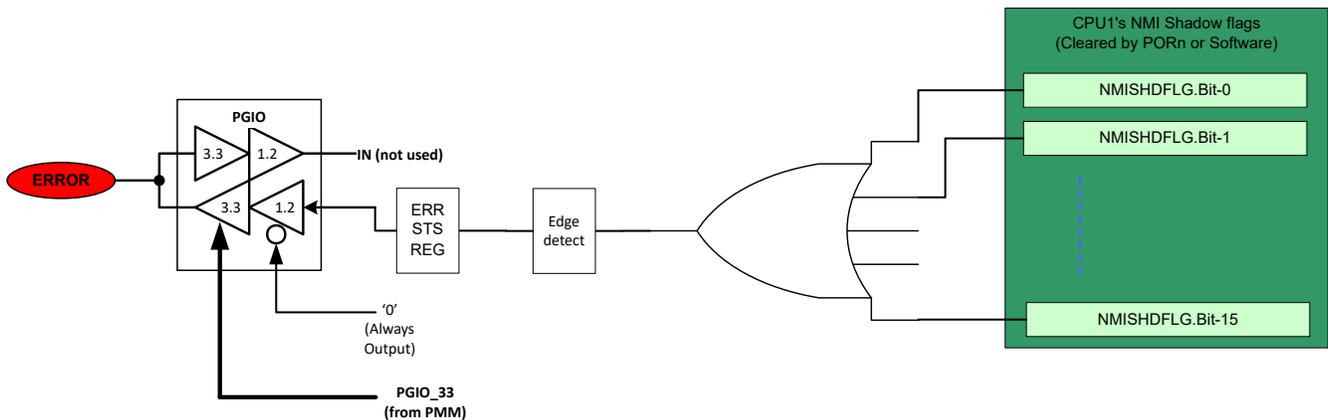


図 1-4. ERRORSTS ピン構造図

1.7 クロック供給

このセクションでは、このデバイスのクロック・ソースとクロック・ドメインについて説明し、クロック・ソースとクロック・ドメインをアプリケーションで使用するように構成する方法を示します。図 1-5 および図 1-6 に、デバイスのクロック供給システムの概要を示します。

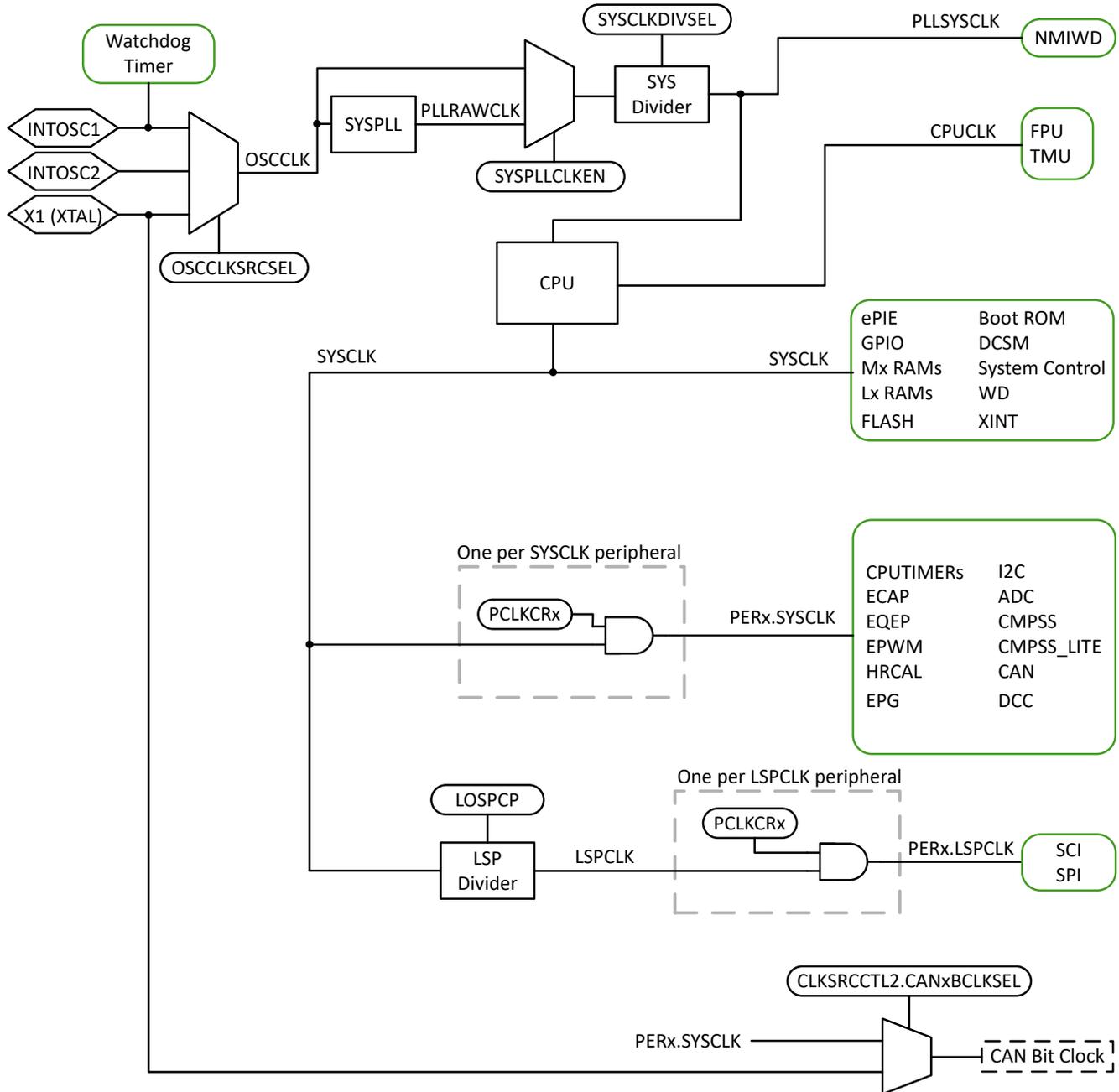


図 1-5. クロック供給システム

SYSPLL

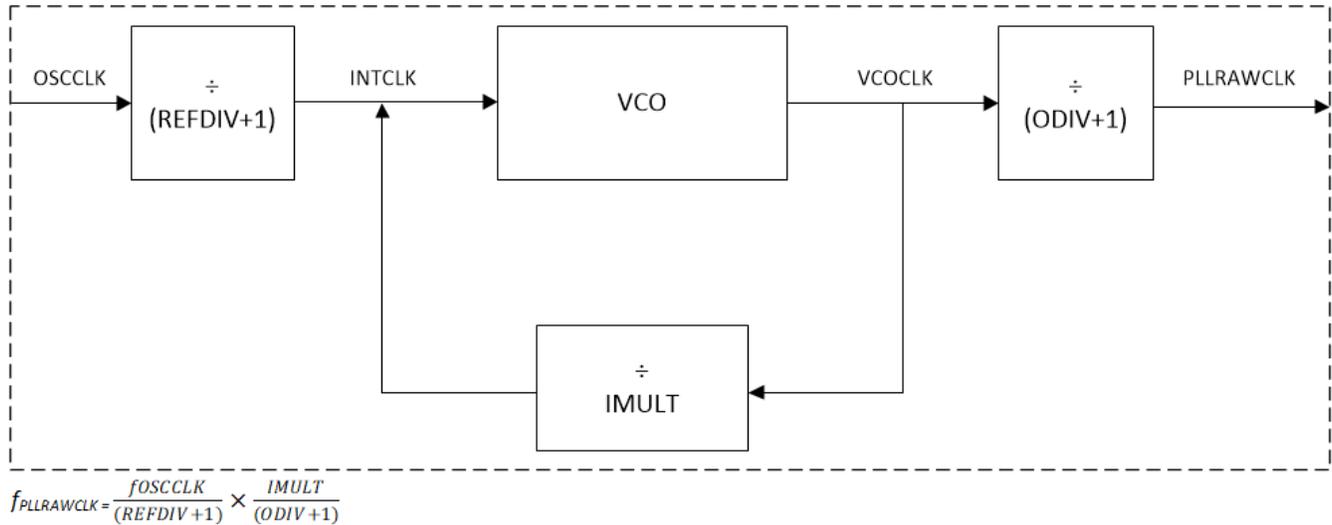


図 1-6. システム PLL

1.7.1 クロック・ソース

デバイス内のすべてのクロックは、4 つのクロック・ソースのいずれかから派生します。

1.7.1.1 プライマリ内部発振器 (INTOSC2)

電源投入時に、デバイスはオンチップの 10MHz 発振器 (INTOSC2) からクロック供給されます。INTOSC2 はプライマリ内部クロック・ソースで、リセット時のデフォルトのシステム・クロックです。INTOSC2 はブート ROM の実行に使用され、アプリケーションのシステム・クロック・ソースとして使用できます。INTOSC2 の周波数の許容誤差は非常に緩いため、CAN のタイミング要件を満たせないことに注意してください。CAN モジュールを使用するには、外部発振器が必要です。INTOSC2 をシステム・クロック・ソースとして使用する場合は、GPIO19 (X1) と GPIO18 (X2) を GPIO ピンとして使用できますが、ExtR 機能を使用する場合は除きます (GPIO19 は ExtR に使用されます)。

1.7.1.1.1 外部抵抗 (ExtR) モード

INTOSC2 の精度を向上させるために、外部抵抗 (ExtR) モードを使用できます。このモードでは、高精度の外部抵抗を基板に追加し、ExtR 対応ピンに接続できます。各パッケージでピンが ExtR 対応であるデバイスのデータシートを参照してください。

INTOSC2 は内部抵抗モードで起動するため、デフォルトでは ExtR を使用しません。標準の内部抵抗を使用する場合、INTOSC2 は引き続き完全に機能しますが、正確な ExtR を使用する場合よりも性能が低くなります。CLK_CFG_REGS CLKSRCCTL1.INTOSC2CLKMODE ビットを 1 に設定すると、より正確な外部抵抗 (ExtR) を使用するように変更できます。ExtR 機能を正しくイネーブルにする方法の詳細については、セクション 1.7.9 を参照してください。

1.7.1.1.1.1 外部高精度抵抗 (ExtR) を使用した場合の INTOSC2

INTOSC2 に外付けの高精度抵抗と組み合わせて使用することで、より高い精度を実現できます。

必要な外付けコンポーネントは次のとおりです。

- ExtR ピンと VSS 間での 100kΩ の高精度抵抗
- ノイズ・フィルタリング用の 10nF コンデンサ
- 電源ノイズの低減と過渡負荷への対応のため、VDDIO に最低 20μF の容量

図 1-7 に、これらの必要な外付けコンポーネントの配置例を示します。

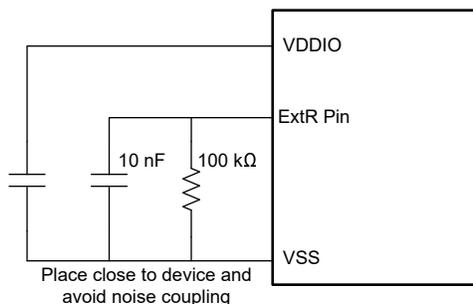


図 1-7. ExtR の回路例

ExtR モードでは、発振器の周波数誤差が、ExtR 抵抗の精度に直接比例します。

ExtR を使用する INTOSC の性能には、VDDIO 電源の品質が直接影響を与えます。ジッタ、ノイズ、その他の性能上の問題を回避するため、VDDIO に配置する容量値と回路構成は、可能な限りクリーンな電源を提供するように配慮して設計する必要があります。

ExtR ピンに抵抗を配置すると、このピンは GPIO または X1 として使用できなくなります。

表 1-6 に、ExtR の仕様値を示します。

表 1-6. ExtR の仕様

パラメータ	テスト条件	最小値	標準値	最大値	単位
$f_{INTOSC2-ExtR-ERR-PERC}$	誤差 0% の理想的な ExtR 抵抗 (100kΩ)	-0.7	0	+0.7	%
$f_{INTOSC2-ExtR}$	誤差 0% の理想的な ExtR 抵抗 (100kΩ)	9.93	10	10.07	MHz
$f_{ExtR-SETTLING}$	ExtR モードへの切り替え		1		ms
ExtR 抵抗、 R_{ExtR}			100		kΩ
ExtR デカップリング容量、 C_{ExtR}			10		nF
VDDIO デカップリング容量、 C_{VDDIO}		20			μF

表 1-7 に、抵抗のパラメータを指定して INTOSC2 の総合誤差を求める場合の計算例を示します。

表 1-7. 総合誤差の計算例:

パラメータ	値	単位
INTOSC2 の理想的な周波数変動	0.70	%
ExtR 抵抗の許容誤差	$R_{TOLERANCE}$	%
ExtR 抵抗の温度係数	R_{TEMPCO}	ppm/°C
動作温度	$T_{OPERATING_POINT}$	°C
ExtR データシートでの周囲温度	$T_{AMBIENT}$	°C
総合周波数誤差	$[(0.70/100) + (R_{TOLERANCE}/100) + ((R_{TEMPCO}/1E6) * \text{abs}(T_{OPERATING_POINT} - T_{AMBIENT}))] * 100$	%

表 1-8 に、上記の計算を使用した値の例を示します。

表 1-8. 総合誤差例の値

パラメータ	値	単位
INTOSC2 の理想的な周波数変動	0.70	%
ExtR 抵抗の許容誤差	0.10	%
ExtR 抵抗の温度係数	25	ppm/°C
動作温度	90	°C
ExtR データシートでの周囲温度	25	°C
総合周波数誤差の計算	$((0.70/100) + (0.10/100) + ((25/1E6) * \text{abs}(90 - 25))) * 100$	%
総合周波数誤差の計算	0.96	%

最高の性能を得るには、以下の基板レイアウト・ガイドラインに従ってください。

- ExtR の配線はできる限り短くします
- 最も近い VSS ピンに ExtR を接続します
- ExtR (R_{ExtR}) と C_{ExtR} は、C2000 デバイスと同じ側に配置し、配線は同じ層のみで行います
- 隣接する GPIO ピン (GPIO18、X2 など) との結合を低減するため、GPIO ピンへの配線は、反対側で別の層を使用していきます
- VSS との接続は、必ず、VSS 面において行い、C2000 デバイスの VSS ピンと直接接続します
- 図 1-8 に示すように、ExtR 配線の周囲に VSS によるガード・トレースを配置することを推奨します
- 隣接する層に信号線をルーティングしないよう、ExtR および C_{ExtR} の下層には、VSS または VDDIO の面を埋め込みます

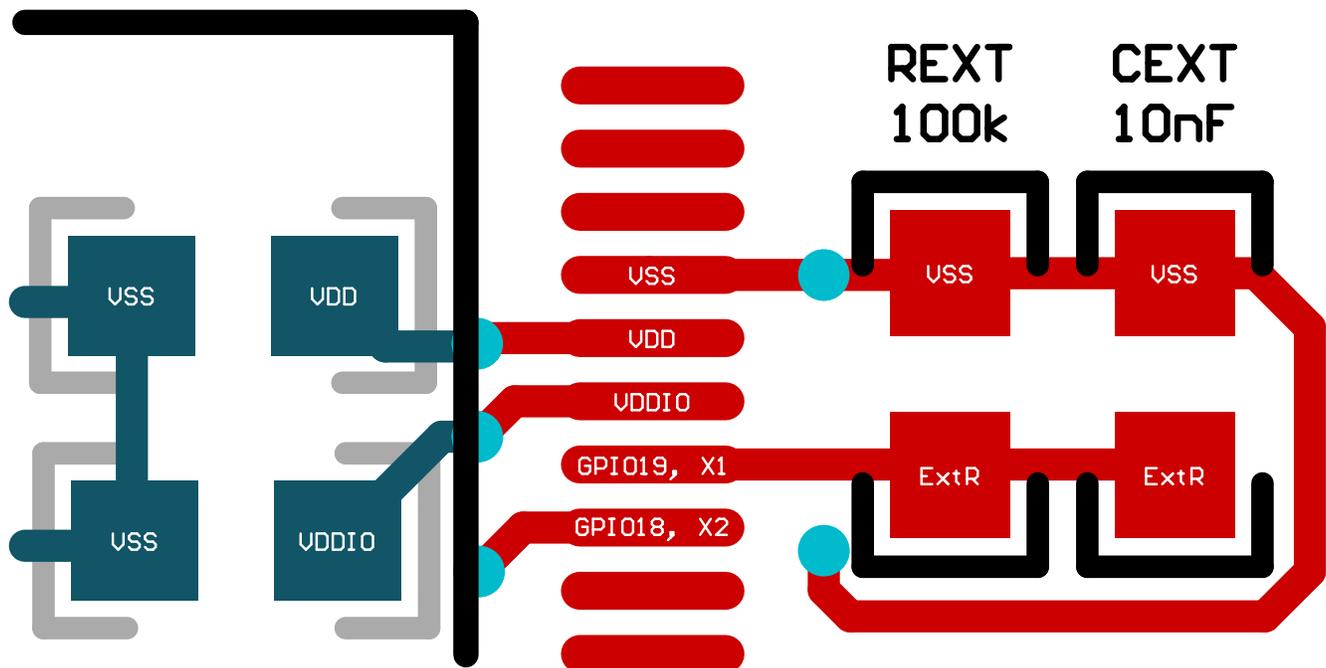


図 1-8. ExtR の PCB レイアウト例

1.7.1.2 バックアップ内部発振器 (INTOSC1)

このデバイスには、冗長オンチップ 10MHz 発振器 (INTOSC1) も内蔵されています。INTOSC1 は、通常、ウォッチドッグ・タイマとクロック消失検出回路 (MCD) へのみクロックを供給するバックアップ・クロック・ソースです。MCD がイネーブルで、システム・クロックが消失していることが検出されると、システム PLL がバイパスされ、すべてのシステム・クロックが INTOSC1 に自動的に接続されます。INTOSC1 は、デバッグ目的でシステム・クロック・ソースとして手動で選択することもできます。

1.7.1.3 Auxiliary Clock Input (AUXCLKIN)

An additional external clock source is supported on GPIO29 (AUXCLKIN). This must be a single-ended 3.3V external clock as shown in [図 1-9](#) and can be used as the clock source for DCAN and MCAN. Frequency limits and timing requirements are found in the [TMS320F280013x Real-Time Microcontrollers Data Sheet](#). The external clock can be connected directly to the GPIO29 pin.

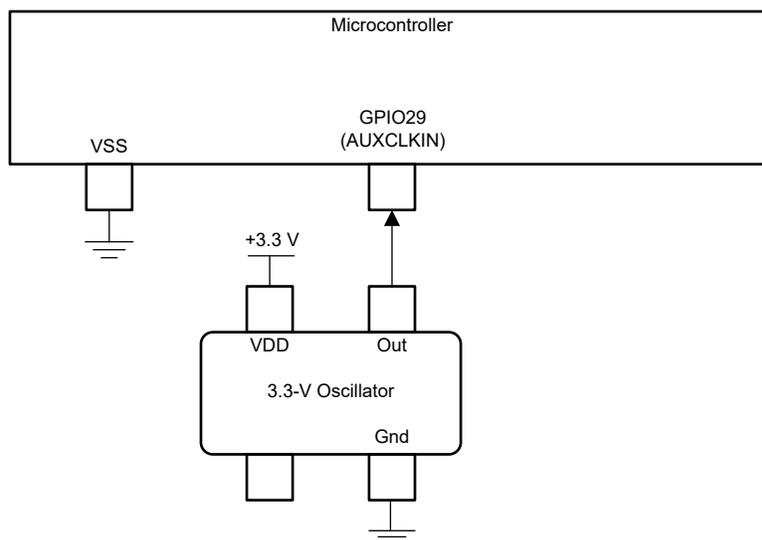


図 1-9. AUXCLKIN

1.7.1.4 外部発振回路 (XTAL)

このデバイスは、メイン・システムとして使用できる外部クロック・ソース (XTAL) と、CAN ビット・クロック・ソースをサポートしています。周波数制限およびタイミング要件については、デバイスのデータシートを参照してください。外部クロック・ソースは、X1/GPIO19 ピンと X2/GPIO18 ピンを使用します。電源投入後、[セクション 1.7.6](#) の手順に従って X1 および X2 ピンの機能をイネーブルにできます。

次の 3 種類の外部クロック・ソースがサポートされています。

- シングルエンド 3.3V 外部クロック。図 1-10 に示すように、クロック信号を X1 に接続できます。

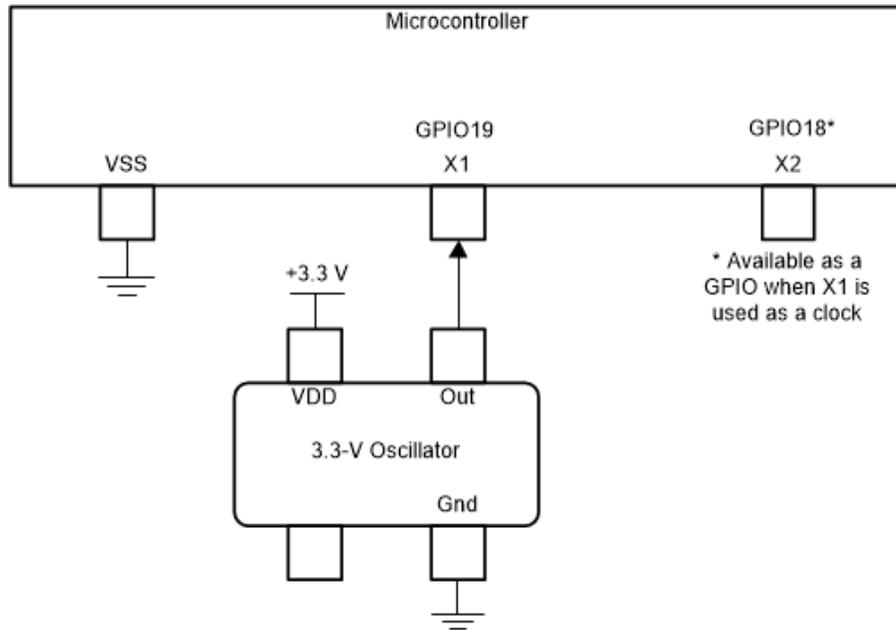


図 1-10. シングルエンド 3.3V 外部クロック

- 外部水晶振動子。図 1-11 に示すように、X1 と X2 の間に水晶振動子を接続して、さらに負荷コンデンサを VSS に接続します。

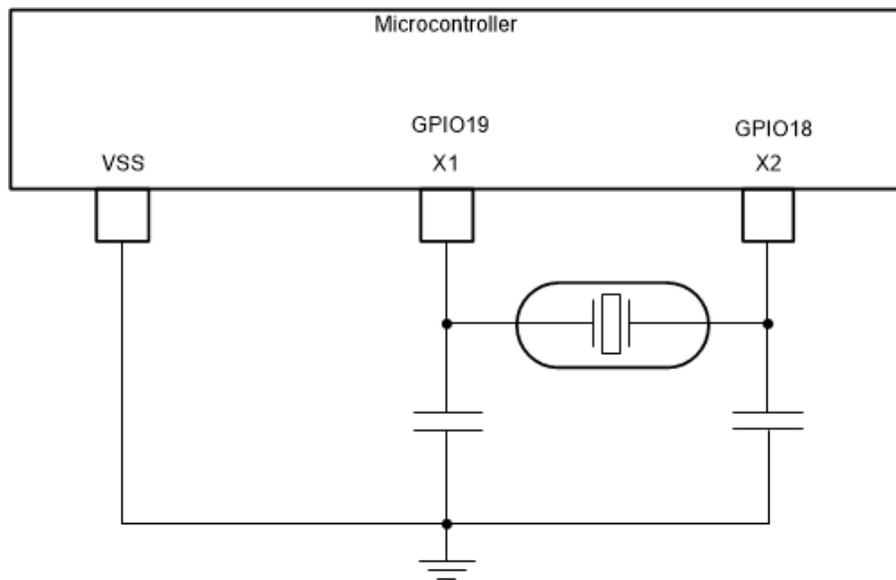


図 1-11. 外部水晶振動子

- 外部共振子。図 1-12 に示すように、X1 と X2 の間に共振子を接続し、そのグラウンドを VSS に接続します。

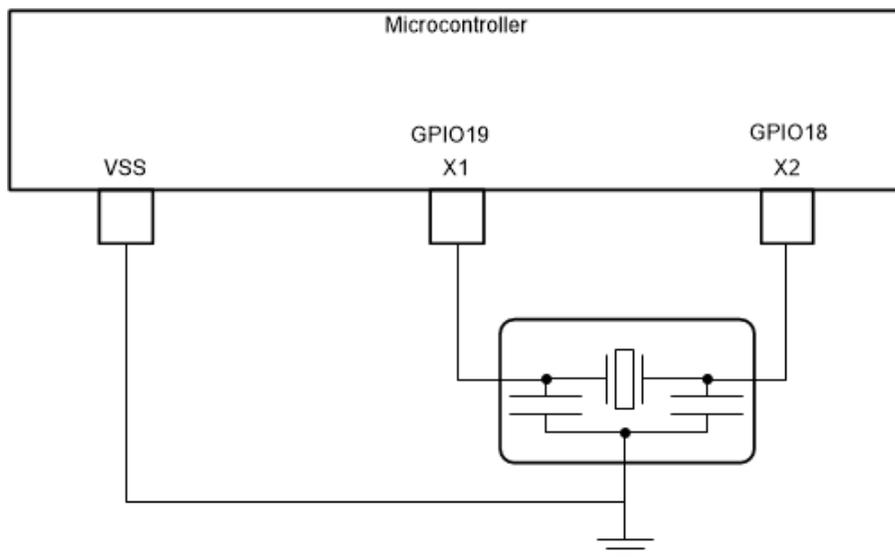


図 1-12. 外部共振子

表 1-9. 代替モード

XTALCR ビット ⁽¹⁾		動作モード	GPIO19 を X1 で使用可能?	GPIO18 を X2 で使用可能?
OSCOFF	SE			
0	0	水晶振動子モード: X1/X2 に水晶振動子を接続	不可	不可
0	1	シングルエンド・モード: X1 に外部クロック	不可	可
1	0	発振回路オフ	可	可
1	1	シングルエンド・モード: X1 に外部クロック ⁽²⁾	不可	可

(1) OSCOFF および SE によって、GPIO18 および GPIO19 の代替モードが決まります。

(2) このモードでは、X1 に約 1kΩ のプルダウンがあり、外部シングルエンド・クロックは、この負荷を駆動できる必要があります。

1.7.2 派生クロック

前のセクションで説明したクロック・ソースを通倍 (PLL 経由) および分周することで、アプリケーションに必要なクロック周波数を生成できます。このプロセスにより、このセクションで説明する一連の派生クロックが生成されます。

1.7.2.1 発振器クロック (OSCCLK)

INTOSC2、XTAL、または INTOSC1 のいずれかを、CPU およびほとんどのペリフェラルのマスタ・リファレンス・クロック (OSCCLK) として選択する必要があります。OSCCLK は、直接使用することも、より高い周波数に到達するようにシステム PLL 経由で供給することもできます。リセット時に、OSCCLK はデフォルトのシステム・クロックとなり、INTOSC2 に接続されます。

1.7.2.2 システム PLL 出カクロック (PLLRAWCLK)

システム PLL により、デバイスは最大定格動作周波数で動作することができます。また、システム PLL は、ほとんどのアプリケーションでメイン・システム・クロックを生成します。この PLL は、リファレンスとして OSCCLK を使用します。

PLLRAWCLK は、PLL の電圧制御発振器 (VCO) の出力です。構成手順については、セクション 1.7.6 を参照してください。

1.7.3 デバイス・クロック・ドメイン

デバイス・クロック・ドメインは、デバイス内の各種モジュールのクロック入力を供給します。これらのドメインは、直接、または追加の分周器を介して、派生クロックに接続されています。

1.7.3.1 システム・クロック (PLLSYSCLK)

NMI ウォッチドッグ・タイマには、独自のクロック・ドメイン (PLLSYSCLK) があります。名前に関係なく、PLLSYSCLK はシステム PLL (PLLRAWCLK) または OSCCLK に接続できます。選択したクロック・ソースは、SYSCLKDIVSEL レジスタを使用して構成される周波数分周器を経由して実行されます。PLLSYSCLK は HALT モードでゲートされます。

1.7.3.2 CPU クロック (CPUCLK)

CPU には、CPU とフラッシュ・ラッパーへのクロック供給に使用する独自のクロック (CPUCLK) があります。このクロックは PLLSYSCLK と同じですが、CPU が IDLE、STANDBY、または HALT モードに移行するとゲートされます。

1.7.3.3 CPU サブシステム・クロック (SYSCLK および PERx.SYSCLK)

CPU は、プライベート (M0 および M1)、ローカル共有 (LS0 および LS1)、ブート ROM、およびその他のペリフェラルにクロック (SYSCLK) を供給します。このクロックは PLLSYSCLK と同じですが、CPU が HALT モードまたは STANDBY モードに移行するとゲートされます。

各ペリフェラル・クロックには、PCLKCRx レジスタによって制御される独自の独立したクロック・ゲーティングがあります。

注

PCLKCRx を使用する場合、ペリフェラルへのクロックをイネーブルにした後、アプリケーションは 5 SYSCLK サイクル待機する必要があります。

1.7.3.4 低速ペリフェラル・クロック (LSPCLK および PERx.LSPCLK)

SCI および SPI モジュールは、CPU 周波数よりもはるかに低速なビット・レートで通信できます。これらのモジュールは、SYSCLK から派生した低速ペリフェラル・クロック (LSPCLK) を生成する共有クロック分周器に接続されています。LSPCLK はデフォルトでは 14 分周器を使用しますが、比率は LOSPCP レジスタで変更できます。各 SCI および SPI モジュールのクロック (PERx.LSPCLK) は、PCLKCRx レジスタを使用して個別にゲートできます。

1.7.3.5 CAN ビット・クロック

CAN ビット・クロックに必要な周波数の許容誤差は、ビット・タイミングの設定とネットワーク構成によって異なり、0.1% まで小さくすることができます。メイン・システム・クロック (SYSCLK の形式) の精度が十分でない可能性があるため、ビット・クロックを CLKSRCCTL2 レジスタ経由で XTAL に接続することもできます。CAN モジュールごとに独立した選択があります。

正常な動作を実現するには、CAN ビット・クロックの周波数を SYSCLK 周波数以下にする必要があります。

1.7.3.6 CPU タイマ 2 クロック (TIMER2CLK)

CPU タイマ 0 および 1 は PERx.SYSCLK に接続されています。タイマ 2 はデフォルトで PERx.SYSCLK に接続されていますが、TMR2CLKCTL レジスタを使用して INTOSC1、INTOSC2、または XTAL に接続することもできます。このレジスタには、タイマ 2 用の個別のプリスケール分周器もあります。SYSCLK 以外のソースを使用する場合は、ソースを SYSCLK 周波数の半分以下に分周する必要があります。

SYSCLK 以外のソースを使用する主な理由は、内部周波数測定です。ほとんどのアプリケーションでは、タイマ 2 は SYSCLK を使用して動作します。

1.7.4 XCLKOUT

デバッグとテストの目的で、クロックを直接観測する必要が生じることがあります。外部クロック出力 (XCLKOUT) 機能は、GPIO16 または GPIO18 の外部ピンにクロックを接続することで、この観測をサポートします。利用可能なクロック・ソースは PLLSYSCLK、PLLRAWCLK、SYSCLK、INTOSC1、INTOSC2、XTAL です。

XCLKOUT を使用するには、まず、CLKSRCCTL3 レジスタを使用してクロック・ソースを選択します。次に、XCLKOUTDIVSEL レジスタを使用して目的の出力分周器を選択します。最後に、GPIO 構成レジスタを使用して、GPIO16 または GPIO18 をマルチプレクサ・チャンネル 11 に接続します。

1.7.5 クロック接続

表 1-10 にクロック・ドメイン別に並べ替えたクロック接続を示し、表 1-11 にモジュール名別に並べ替えたクロック接続を示します。

表 1-10. クロック・ドメイン別に並べ替えたクロック接続

クロック・ドメイン	モジュール名
CPUCLK	FPU
	TMU
SYSCLK	ePIE
	Mx RAM
	LSx RAM
	フラッシュ
	ブート ROM
	GPIO 入力同期および認定
	WD
	XINT
	DCSM
PLLSYSCLK	CPU
	NMIWD
PERx.SYSCLK	Timer0~2
	DCC0
	ePWM1~7
	eCAP1~2
	eQEP1
	ADCA、C
	CMPSS1
	CMPSS_LITE2~4
	I2CA~B
	CANA
	HRCAL
	EPG
PERx.LSPCLK	SCIA~C
	SPIA
WDCLK (INTOSC1)	ウォッチドッグ・タイマ

表 1-11. モジュール名別に並べ替えたクロック接続

モジュール名	クロック・ドメイン
ADCA、C	PERx.SYSCLK
ブート ROM	SYSCLK
CANA	PERx.SYSCLK
CMPSS1	PERx.SYSCLK
CMPSS_LITE2~4	PERx.SYSCLK
CPU	PLLSYSCLK
CPU タイマ (0~2)	PERx.SYSCLK
DCC0	PERx.SYSCLK
DCSM	SYSCLK
eCAP1~2	PERx.SYSCLK
ePIE	SYSCLK
ePWM1~7	PERx.SYSCLK
eQEP1	PERx.SYSCLK
EPG	PERx.SYSCLK
フラッシュ	SYSCLK
FPU	CPUCLK
GPIO 入力同期および認定	SYSCLK
I2CA~B	PERx.SYSCLK
LSx RAM	SYSCLK
Mx RAM	SYSCLK
NMIWD	PLLSYSCLK
SCIA~C	PERx.LSPCLK
SPIA	PERx.LSPCLK
TMU	CPUCLK
ウォッチドッグ・タイマ	WDCLK (INTOSC1)

1.7.6 クロック・ソースと PLL のセットアップ

クロック構成を最終的に決定するのは、アプリケーションのニーズです。アプリケーションの性能、消費電力、総システム・コスト、EMC などの具体的な懸念事項については、このドキュメントでは説明しませんが、これらの事項から以下の質問に対する回答が得られます。

1. 望ましい CPU 周波数は？
2. CAN が必要か？
3. どのようなタイプの外部発振器またはクロック・ソースを使用できるか？

CAN が必要な場合は、正確な周波数を持つ外部クロック・ソースをリファレンス・クロックとして使用する必要があります。それ以外の場合は、INTOSC2 のみを使用し、必要となる外付け部品を増やさないようにすることができます。

1.7.7 外部水晶振動子または共振器の使用

X1 ピンと X2 ピンは GPIO19 と GPIO18 としても使用できます。電源投入時に、これらのピンは GPIO モードになり、オンチップの水晶発振器の電源がオフになります。ピンを X1 モードと X2 モードに切り替え、発振器をイネーブルにするには、次の手順を使用します。

1. XTALCR.OSCOFF ビットをクリアします。
2. 水晶振動子の電源がオンになるまで待ちます。標準的なウェイト時間は 1ms ですが、これは使用している水晶振動子によって異なります。

3. X1CNT.CLR に 1 を書き込んで X1 カウンタをクリアし、X1CNT レジスタの X1 カウンタ値が 2047 (0x7ff) で飽和しなくなるまでクリアを続けます。
4. X1CNT レジスタの X1 カウンタ値が 2047 (0x7ff) に達するまで待ちます。ステップ 3~4 をさらに 3 回繰り返します。
5. CLKSRCCTL1.OSCCLKSRCSEL に 1 を書き込んで、OSCCLK ソースとして XTAL を選択します。
6. MCDCCR レジスタの MCLKSTS ビットを確認します。セットされている場合、発振器は電源投入を完了しておらず、より多くの時間が必要になります。
 - a. MCDCCR.MCLKCLR に 1 を書き込んで、クロック消失ステータスをクリアします。
 - b. ステップ 2~7 を繰り返します。デバイスをリセットしないでください。リセットすると発振器の電源がオフになり、手順をステップ 1 からやり直すことが必要になります。
 - c. 発振器の電源投入が 10 ミリ秒で完了していない場合、実際のクロック障害が発生しています。
7. MCDCCR.MCLKSTS がクリアされている場合、発振器の起動は成功です。システム・クロックは現在、XTAL から派生しています。

1.7.8 外部発振器の使用

X1 ピンに接続された外部発振器を使用する手順は、水晶振動子または共振器を使用する手順と同様です。

1. XTALCR.OSCOFF ビットをクリアします。
2. XTALCR.SE ビットをセットしてシングルエンド・モードをイネーブルにします。
3. X1CNT.CLR に 1 を書き込んで X1 カウンタをクリアし、X1CNT レジスタの X1 カウンタ値が 2047 (0x7ff) で飽和しなくなるまでクリアを続けます。
4. X1CNT レジスタの X1 カウンタ値が 2047 (0x7ff) に達するまで待ちます。
5. ステップ 3~4 をさらに 3 回繰り返します。
6. CLKSRCCTL1.OSCCLKSRCSEL に 1 を書き込んで、OSCCLK ソースとして XTAL を選択します。
7. MCDCCR レジスタの MCLKSTS ビットを確認します。セットされている場合、外部発振器またはデバイスのいずれかに障害が発生しています。
8. MCLKSTS がクリアされている場合、外部クロックへの切り替えは成功です。システム・クロックは現在、XTAL から派生しています。

1.7.9 外部抵抗 (ExtR) と内部発振器の併用

INTOSC2 は内部抵抗モードで起動します。高精度の外部抵抗を使用できるようにするには、次の手順を使用します。

1. CLKSRCCTL1.OSCCLKSRCSEL をセットして OSCCLKSRC ソースを INTOSC1 に切り替えます。その後、クロック遷移に必要な遅延が生じます。
2. driverlib 関数「ASysCtl_setExtRCounterDelay(ASYSCCTL_EXTR_COUNTER_DELAY_2048);」を使用して、必要なクロック遷移時間を設定します。
3. driverlib 関数「SysCtl_setIntOSC2_Mode(SYSCTL_INTOSC2_MODE_EXTR);」を使用して ExtR モードをイネーブルにします。その後、10000 サイクルの遅延が生じます (driverlib の「SysCtl_delay(10000U);」関数を使用して合計 50009 サイクルを実現し、バッファ時間を追加できます)。
4. 「ASysCtl_getExtROscStatus()」が「ASYSCCTL_EXTR_ENABLE_COMPLETE」に等しくなるまで待ちます。
5. CLKSRCCTL1.OSCCLKSRCSEL をセットして OSCCLKSRC ソースを INTOSC2 に戻します。その後、クロック遷移に必要な遅延が生じます。

上記のステップが完了した後、INTOSC2 は ExtR を使用します。PLL ロックなどの動作は、すべてのステップと含まれているセリング遅延が完了した後に開始する必要があります。

1.7.10 PLL 設定の選択

図 1-6 に示す式を使用して、PLL を構成できます。

IMULT は整数値の通倍器です。

REFDIV は、OSCCLK のリファレンス分周器です。

ODIV は、PLLRAWCLK の出力分周器です。

PLLSYSCLKDIV は、システム・クロック分周器です。

通倍器と分周器のとりうる値については、それぞれのレジスタのドキュメントを参照してください。

通倍器と分周器の複数の組み合わせで、同じ出力周波数を生成できます。ただし、リファレンス・クロック周波数と通倍器 (VCO 周波数と呼ばれます) の積は、データシートで指定されている範囲内にする必要があります。

注

システム・クロック周波数 (PLLSYSCLK) は、データシートに規定されている限界値を超えてはなりません。この限界値については、発振器の公差は許容されません。

1.7.11 システム・クロックのセットアップ

アプリケーション要件を把握したら、具体的なクロック構成を決定できます。デフォルト構成では、分周器を 1 に設定したシステム・クロック (PLLSYSCLK) として INTOSC2 を使用します。目的のアプリケーション構成を設定するには、次の手順を使用します。

例については、C2000Ware インストール環境内のデバイスの SysCtl_setClock() 関数を参照してください。

システム PLL を設定するための推奨シーケンス:

1. SYSPLLCTL1[PLLCLKEN] をクリアして PLL をバイパスします。これを有効にするには、少なくとも 60 個の NOP 命令を許可します。
2. SYSPLLCTL1.PLEN=0 に書き込んで PLL の電源をオフにします。これを有効にするには、少なくとも 60 個の NOP 命令を許可します。
3. CLKSRCCTL1.OSCCLKSRCSEL に書き込んでリファレンス・クロック・ソース (OSCCLK) を選択します。これを有効にするには、少なくとも 300 個の NOP 命令を許可します。
4. SYSCLKDIVSEL[PLLSYSCLKDIV] をクリアして、システム・クロック分周器を /1 に設定し、最も高速な PLL 構成を確保します。
5. SYSPLLMULT に 32 ビットの値を一度に書き込むことで、IMULT、REFDIV、ODIV を同時にセットします。これにより、PLL が自動的にイネーブルになります。マルチプライヤと分周器の設定が、データシートで定義されている周波数仕様に違反しないようにします。
6. ロック・ステータス・ビットが High、つまり SYSPLLSTS.LOCKS=1 になるかをポーリングし、PLL がロックされるまで待ちます。
7. リファレンス・クロックを OSCCLK、測定中のクロックを PLLRAWCLK に設定して DCC を構成し、PLL の周波数を確認します。周波数が範囲外の場合、PLLRAWCLK を SYSCLK としてイネーブルにせず、ここで停止してトラブルシューティングを行います。構成と使用方法の詳細については、「デュアル・クロック・コンパレータ (DCC)」の章を参照してください。
8. SYSPLLCTL1[PLLCLKEN] をセットして、システム・クロックとして PLL に切り替えます。

注

1. OSCCLK ソースを変更する前に、SYSPLL をバイパスし、手動で電源をオフにする必要があります。
2. PLL をバイパスした後、つまり SYSPLLCTL1.PLLCLKEN=0 にした後、少なくとも 60 CPU クロック・サイクルの遅延が必要です。
3. PLL の電源をオフにした後、つまり SYSPLLCTL1.PLEN=0 にした後、少なくとも 60 CPU クロック・サイクルの遅延が必要です。
4. OSCCLK ソースを変更した後、少なくとも 300 CPU クロック・サイクルの遅延が必要です。
5. PLL SLIP ビットはサポートされていません。PLL クロックの有効性チェックには DCC を使用できます。この機能は、C2000Ware 内の SysCtl_setClock() 関数の一部として含まれています。

1.7.12 SYS PLL バイパス

アプリケーションでシステムから PLL クロックをバイパスする必要がある場合は、SYSPLLCTL1.PLLCLKEN=0 に設定する必要があります。バイパスが有効になるまでに、最大 60 CPU クロック・サイクルを要します。一方、PLLSYSCLKDIV が低い値 (たとえば、/2 から /1、/4 から /2 など) に低下した場合、デバイスに最大定格周波数を超えるクロックが供給される可能性があり、予測不能なデバイス動作が生じる可能性があります。したがって、PLL をイネーブル状態からバイパスした後、つまり PLLCLKEN=1 から PLLCLKEN=0 に変わった後、60 CPU クロック・サイクルの遅延が必要になります。

1.7.13 クロック (OSCCLK) の障害検出

安全診断を実現するために、クロック消失検出 (MCD) を使用できます。

表 1-12 に詳述します。

表 1-12. クロック・ソース (OSCCLK) の障害検出

クロック障害 検出回路	検出されるクロック	検出時間 (サイクル単位)	制約
クロック消失検出 (MCD)	INTOSC2、XTAL/X1	8192 INTOSC1 サイクル	INTOSC1 クロック障害は検出できません。

1.7.13.1 クロック消失検出

クロック消失検出 (MCD) ロジックは、リファレンス・クロック・ソースとして INTOSC1 を使用して、OSCCLK の障害を検出します。この回路は、OSCCLK の完全な消失のみを検出し、OSCCLK の周波数ドリフトは検出しません。

この回路は、INTOSC1 (2 次クロック) から供給される 10MHz クロックをバックアップ・クロックとして使用して、OSCLK (1 次クロック) を監視します。この回路は、以下のように機能します。

- 1 次クロック (OSCCLK) は、(MCDPCNT と呼ばれる) 7 ビット・カウンタの計測を維持します。このカウンタは、XRSn を使用して非同期的にリセットされます。
- 2 次クロック (INTOSC1) は、(MCDSCNT と呼ばれる) 13 ビット・カウンタの計測を維持します。このカウンタは、XRSn を使用して非同期的にリセットされます。
- MCDPCNT がオーバーフローするたびに、MCDSCNT カウンタがリセットされます。したがって、OSCCLK が存在する場合、または INTOSC1 の 1/64 以上の場合、MCDSCNT はオーバーフローしません。
- 何らかの理由で OSCCLK が停止した場合、または INTOSC1 の 1/64 未満である場合、MCDSCNT はオーバーフローし、OSCCLK でクロック消失状態が検出されます。
- MCDPCR レジスタを使用して (MCLKOFF ビットを 1 にすることで) MCD をディセーブルにしない限り、上記のチェックは継続的にアクティブになります。
- 回路が OSCCLK の消失を検出した場合、以下が発生します。
 - MCDSTS フラグがセットされます。
 - MCDSCNT カウンタは、クロック消失検出がさらに行われないうちに停止状態になります。
 - CLOCKFAIL 信号が High になり、PWM モジュールに対して TRIP イベントが生成され、CPU1.NMIWD に対して NMI が起動されます。
 - PLL は強制的にバイパスされ、OSCCLK ソースは INTOSC1 (新しいシステム・クロック周波数 = INTOSC1 周波数 10MHz/SYSDIV) に切り替えられます。一方、クロックが INTOSC1 に切り替わると、システムは PLL のリンプ・クロックで動作します。
 - この場合、SYSPLLMULT.IMULT は自動的にゼロに設定されます。
 - MCDSTS ビットがセットされている間、OSCCLKSRCSEL ビットは無効になり、OSCCLK は INTOSC1 に強制的に接続されます。
 - システムへの PLLRAWCLK は、自動的に INTOSC1 に切り替わります。
- MCLKCLR ビットが書き込まれると (W=1 ビット)、MCDSTS ビットがクリアされ、OSCCLK ソースは OSCCLKSRCSEL ビットによって決定されます。MCLKCLR に書き込むと、MCDPCNT カウンタと MCDSCNT カウンタもクリアされ、回路でクロック消失検出を再評価できるようになります。クロック消失検出後に PLL をロックする場合は、クロック・ソースを INTOSC1 に切り替え (OSCCLKSRCSEL レジスタを使用)、MCLKCLR を実行して PLL を再ロックします。
- 電源投入時に MCD がイネーブルになります。

図 1-13 に、クロック消失ロジックの機能フローを示します。

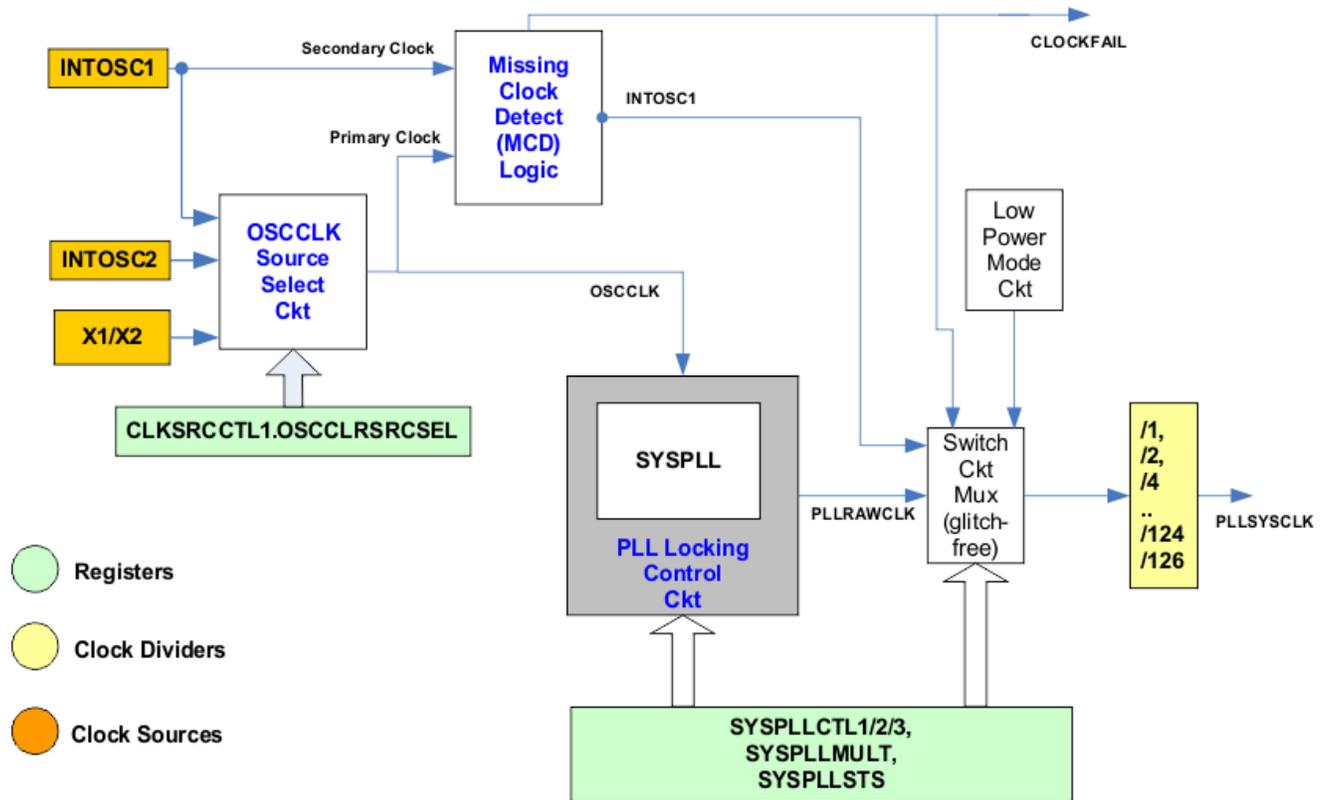


図 1-13. クロック消失検出ロジック

注

OSCCLK が停止状態のときに完全なクロック障害が発生した場合、CLOCKFAIL 信号が High になるまでに、最大 8192 INTOSC1 サイクル (すなわち 0.8192ms) がかかることがあります。その後、次のようになります。

- NMI が生成されます。
- OSCCLK が INTOSC1 に切り替えられます。
- PWM トリップが発生します。

1.8 32 ビット CPU タイマ 0/1/2

このセクションでは、[図 1-14](#) に示す 3 つの 32 ビット CPU タイマ (TIMER0/1/2) について説明します。

Timer0 および Timer1 は、ユーザー・アプリケーションで使用できます。Timer2 は、リアルタイム・オペレーティング・システム (TI-RTOS など) 用に予約されています。アプリケーションでこのタイマを利用するオペレーティング・システムを使用していない場合は、Timer2 をアプリケーションで使用できます。タイマ割り込み信号 (TINT0、TINT1、TINT2) は、[図 1-15](#) に示すように接続されています。

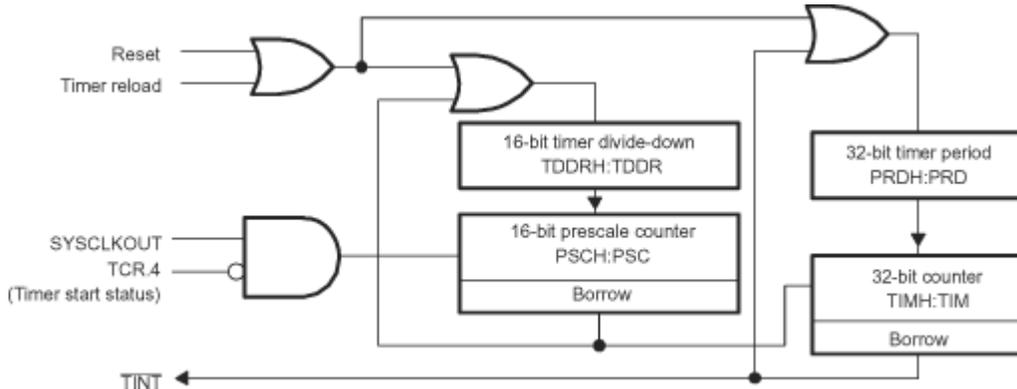
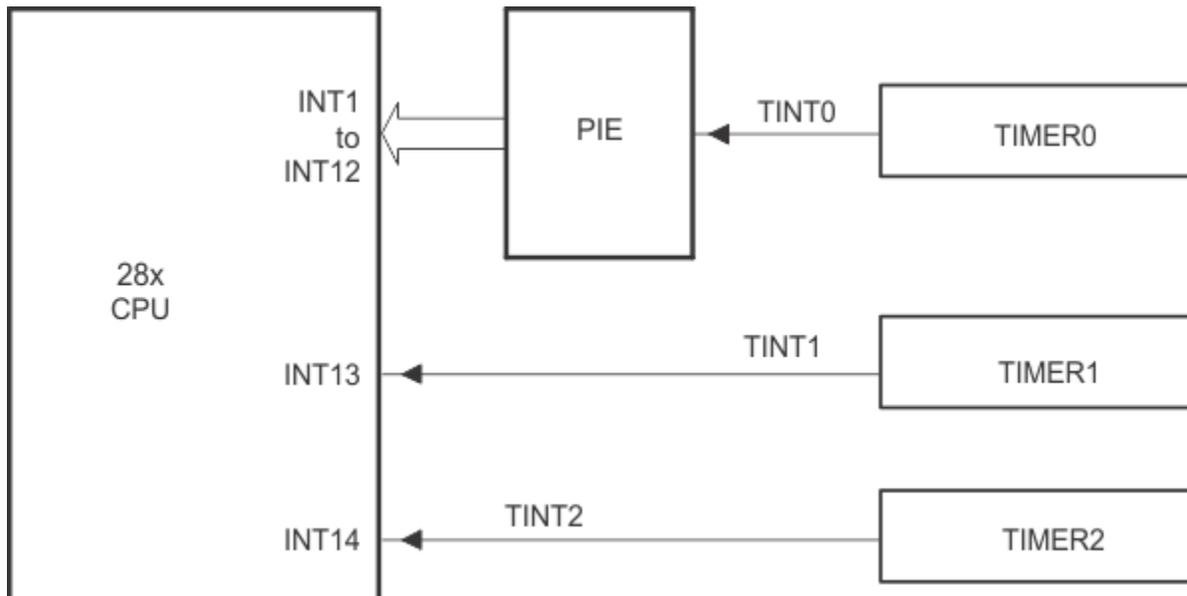


図 1-14. CPU タイマ



- A. タイマ・レジスタは、C28x プロセッサのメモリ・バスに接続されています。
- B. CPU タイマは SYSCLKOUT に同期しています。

図 1-15. CPU タイマの割り込み信号および出力信号

CPU タイマ動作の概要は、次のとおりです。

- 32 ビット・カウンタ・レジスタ TIMH:TIM には、周期レジスタ PRDH:PRD の値がロードされます。
- カウンタは、 $(TPR[TDDR:TDHR]+1)$ SYSCLK サイクルごとに 1 回デクリメントします。ここで、TDDR:TDHR はタイマ分周値です。

- カウンタが 0 に達すると、タイマ割り込み出力信号が割り込みパルスを発生します。
[セクション 1.15](#) に示すレジスタは、タイマの設定に使用します。

1.9 ウォッチドッグ・タイマ

ウォッチドッグ・モジュールは、プリスケールされたクロック (INTOSC1 に接続された WDCLK) から供給される 8 ビット・カウンタで構成されます。カウンタが最大値に達すると、モジュールは 512 WDCLK の出力パルスを生成します。このパルスは、割り込みまたはリセットを生成できます。ウォッチドッグ・カウンタをリセットするには、CPU は定期的に 0x55 + 0xAA シーケンスをウォッチドッグ・キー・レジスタに書き込む必要があります。カウンタはディセーブルにすることもできます。

カウンタのクロックは、2 つの分周器で WDCLK から分周されます。プリスケラは、1/1~1/64 の範囲の 2 のべき乗で調整できます。事前分周器のデフォルトは、下位互換性のために 1/512 ですが、1/2~1/4096 の範囲の 2 のべき乗で調整できます。これにより、安全重視のアプリケーションで幅広いタイムアウト値を実現できます。

図 1-16 に、ウォッチドッグ・モジュール内の各種機能ブロックを示します。

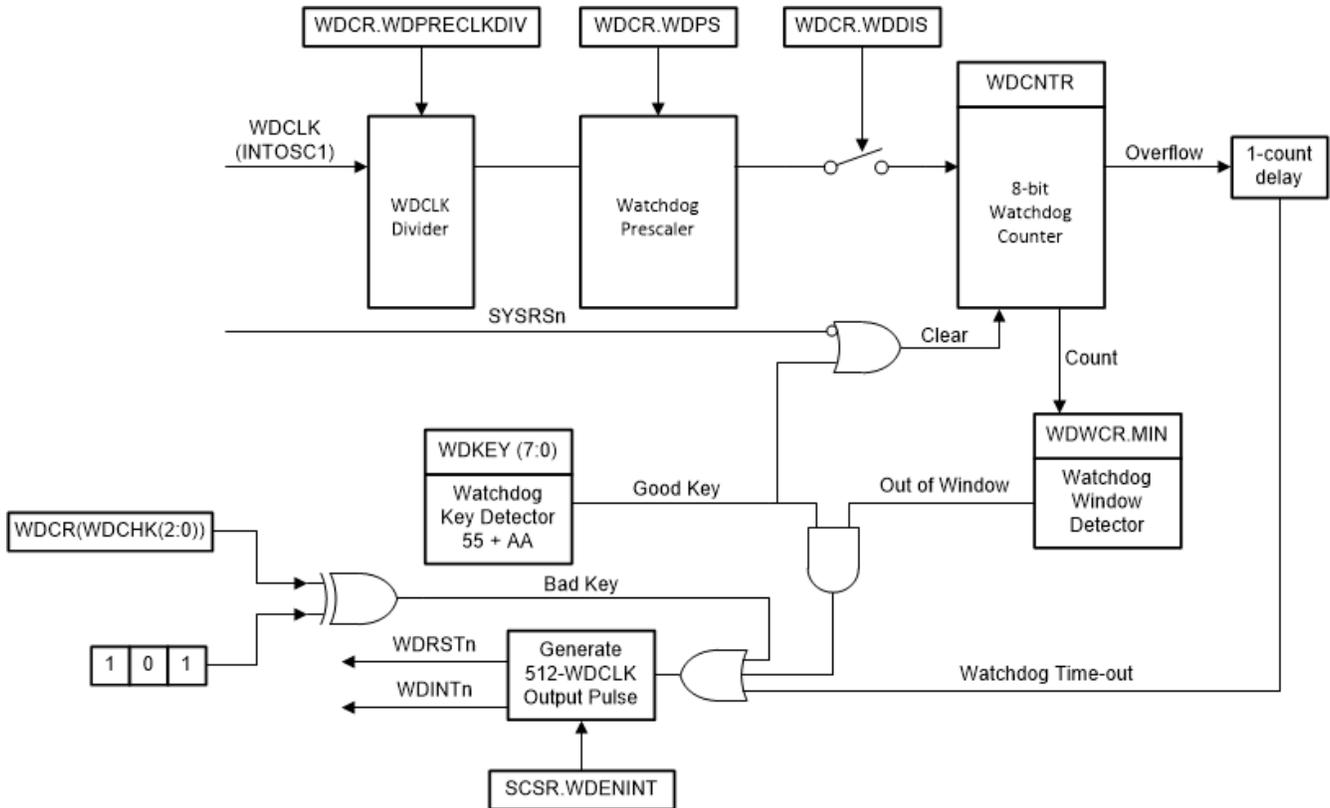


図 1-16. ウォッチドッグ・タイマ・モジュール

1.9.1 ウォッチドッグ・タイマの処理

8ビットのウォッチドッグ・カウンタがオーバーフローする前に正しいシーケンスが WDKEY レジスタに書き込まれると、ウォッチドッグ・カウンタ (WDCNTR) はリセットされます。WDKEY に値 0x55 が書き込まれると、WDCNTR でリセットがイネーブルになります。WDKEY レジスタに次に書き込まれた値が 0xAA の場合、WDCNTR はリセットされます。WDKEY に 0x55 または 0xAA 以外の値が書き込まれた場合、何も起こりません。WDKEY に 0x55 と 0xAA の値のシーケンスが書き込まれても、システム・リセットが発生しないことがあります。WDCNTR がリセットされるのは、WDKEY に 0x55 が書き込まれた後に 0xAA が書き込まれた場合だけです。

WDCNTR のリセットをイネーブルにする最初の動作を、表 1-13 のステップ 3 に示します。WDCNTR は、実際にはステップ 6 までリセットされません。ステップ 8 で WDCNTR のリセットを再度イネーブルにし、ステップ 9 で WDCNTR をリセットします。ステップ 10 で WDCNTR のリセットを再度イネーブルにします。ステップ 11 で WDKEY に誤ったキー値を書き込むと、何も起こりませんが、WDCNTR のリセットはイネーブルでなくなり、ステップ 12 の 0xAA は無効になります。

ウォッチドッグがデバイスをリセットするように設定されている場合、WDCR オーバーフローが発生するか、WDCR[WDCCHK] ビットに誤った値が書き込まれると、デバイスがリセットされ、リセット要因レジスタ (RESC) のウォッチドッグ・フラグ (WDRSn) がセットされます。リセット後、プログラムはこのフラグの状態を読み取って、ウォッチドッグによってリセットが行われたかどうかを判定できます。その後、プログラムは WDRSn をクリアし、それ以降のウォッチドッグ・リセットを検出できるようにします。フラグがセットされている場合、ウォッチドッグ・リセットは妨げられません。

表 1-13. ウォッチドッグ・キー・シーケンスの例

ステップ	WDKEY に書き込まれる値	結果
1	0xAA	何も起こらない
2	0xAA	何も起こらない
3	0x55	WDCNTR でリセットがイネーブルになり、次の値が 0xAA の場合にリセットされます。
4	0x55	WDCNTR でリセットがイネーブルになり、次の値が 0xAA の場合にリセットされます。
5	0x55	WDCNTR でリセットがイネーブルになり、次の値が 0xAA の場合にリセットされます。
6	0xAA	WDCNTR がリセットされます。
7	0xAA	何も起こらない
8	0x55	WDCNTR でリセットがイネーブルになり、次の値が 0xAA の場合にリセットされます。
9	0xAA	WDCNTR がリセットされます。
10	0x55	WDCNTR でリセットがイネーブルになり、次の値が 0xAA の場合にリセットされます。
11	0x32	WDKEY に不適切な値が書き込まれました。 何も起こらず、WDCNTR では次の 0xAA によるリセットがイネーブルでなくなります。
12	0xAA	前の無効な値が原因で、何も起こりません。
13	0x55	WDCNTR でリセットがイネーブルになり、次の値が 0xAA の場合にリセットされます。
14	0xAA	WDCNTR がリセットされます。

1.9.2 最小ウィンドウ・チェック

タイムアウト・メカニズムを補完するために、ウォッチドッグにはオプションの「ウィンドウ処理」機能も搭載されています。これは、カウンタ・リセット相互間に最小間隔を確保するものです。これにより、通常のプログラム・フローの大部分をバイパスしているにもかかわらず、ウォッチドッグ処理は実行している、というエラー状態からの保護に役立ちます。

このウィンドウの最小値を設定するためには、必要な最小ウォッチドッグ・カウントを WDWCR レジスタに書き込みます。この値は、次の WDKEY シーケンスの後で有効になります。それ以降は、WDCNTR が WDWCR よりも小さいときにウォッチドッグへのサービスを試みた場合、ウォッチドッグ割り込みまたはリセットがトリガされます。WDCNTR が WDWCR 以上の場合、ウォッチドッグは、通常どおりサービスを受けることができます。

リセット時には、ウィンドウ最小値は 0 であり、ウィンドウ機能はディセーブルになります。

1.9.3 ウォッチドッグ・リセットまたはウォッチドッグ割り込みモード

ウォッチドッグ・カウンタが最大値に達したとき、デバイスをリセットするか ($\overline{\text{WDRST}}$)、または、割り込みをアサートするか ($\overline{\text{WDINT}}$) を SCSR レジスタで構成できます。各条件の動作は次のとおりです。

- リセット・モード:

ウォッチドッグがデバイスをリセットするように構成されている場合、ウォッチドッグ・カウンタが最大値に達すると、 $\overline{\text{WDRST}}$ 信号は 512 OSCCLK サイクルにわたってデバイス・リセット ($\overline{\text{XRS}}$) ピンを Low にします。

- 割り込みモード:

ウォッチドッグ・カウンタが最大値に達すると、512 OSCCLK サイクルにわたって $\overline{\text{WDINT}}$ 信号を Low に駆動し、割り込みをアサートします。割り込みイネーブルの場合、 $\overline{\text{WDINT}}$ の立ち下がりがエッジによって WAKEINT 割り込みが PIE でトリガされます。PIE はエッジ・トリガなので、 $\overline{\text{WDINT}}$ がアクティブの間に WAKEINT を再度イネーブルにしても、重複割り込みは発生しません。

予期しない動作を回避するため、 $\overline{\text{WDINT}}$ がアクティブの間に、ソフトウェアはウォッチドッグの構成を変更しないでください。たとえば、 $\overline{\text{WDINT}}$ がアクティブの間に割り込みモードからリセット・モードに切り替えると、デバイスは直ちにリセットされます。 $\overline{\text{WDINT}}$ がアクティブの間にウォッチドッグを無効にすると、ウォッチドッグが後で再度有効になった場合に、重複割り込みが発生します。 $\overline{\text{WDINT}}$ がアクティブの間にデバッグ・リセットが発行されると、リセット要因レジスタ (RESC) にウォッチドッグ・リセットが表示されます。 SCSR レジスタの WDINTS ビットを読み取ることにより、 $\overline{\text{WDINT}}$ の現在の状態を確認できます。

1.9.4 低消費電力モードでのウォッチドッグの動作

IDLE モードでは、ウォッチドッグ割り込み ($\overline{\text{WDINT}}$) 信号は、CPU への割り込みを生成し、CPU を IDLE モードから復帰させます。他のペリフェラルと同様に、ウォッチドッグ割り込みは IDLE モード中に PIE で WAKE 割り込みをトリガします。ユーザー・ソフトウェアでは、割り込みを発生させたペリフェラルを特定する必要があります。

注:ウォッチドッグ割り込みを使用して IDLE 低消費電力モード状態からウェークアップする場合、ソフトウェアでは、IDLE モードへの再移行を試みる前に、 $\overline{\text{WDINT}}$ 信号が確実に High に戻るようにする必要があります。ウォッチドッグ割り込みが生成されると、 $\overline{\text{WDINT}}$ 信号は 512 OSCCLK サイクルにわたって Low に保持されます。 $\overline{\text{WDINT}}$ の現在の状態は、 SCSR レジスタのウォッチドッグ割り込みステータス・ビット (WDINTS) を読み取ることで特定できます。WDINTS は、2 SYSCLKOUT サイクルごとに $\overline{\text{WDINT}}$ の状態に追従します。

HALT モードでは、ユーザーが $\text{CLKSRCCTL1.WDHALTI} = 1$ に設定すると、内部発振器とウォッチドッグ・タイマがアクティブのままになります。ウォッチドッグ・リセットではシステムを HALT モードからウェークアップできますが、ウォッチドッグ割り込みではできません。

1.9.5 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、さまざまなデバッグ条件で次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・モードになると、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・モードのとき、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、ウォッチドッグは通常どおり動作します。

1.10 低消費電力モード

このデバイスには、クロック・ゲーティング低消費電力モードとして、HALT、IDLE、および STANDBY が備わっています。どの低消費電力モードに移行する場合も、LPMCR レジスタをセットして IDLE 命令を実行します。この命令の詳細については、『[TMS320C28x CPU および命令セット・リファレンス・ガイド](#)』を参照してください。

フラッシュ・プログラムまたは消去動作の実行中は、低消費電力モードに移行しないでください。HALT に移行すると、CPU とペリフェラルのすべての動作が停止します。これには、アクティブな送信や制御アルゴリズムが含まれます。HALT モードに移行する準備をするときは、アプリケーションでシステムが非アクティブな状態に移行する準備ができていないことを確認する必要があります。

HALT モード / 低消費電力モードに移行する前に、HALT からのウェークアップ用に選択したピンの GPIODAT レジスタの値 (GPIOLPMSEL0/1) を確認し、ウェークアップ・イベントがまだアサートされていないことを確認します。

1.10.1 クロック・ゲーティング低消費電力モード

このデバイスの IDLE モードおよび HALT モードは、他の C28x デバイスのモードと同様です。表 1-14 に、いずれかのクロック・ゲート低消費電力モードに移行したときのシステムへの影響を示します。

表 1-14. クロック・ゲーティング低消費電力モードによるデバイスへの影響

モジュール / クロック・ドメイン	IDLE	STANDBY	HALT
SYSCLK	アクティブ	ゲーテッド	ゲーテッド
CPUCLK	ゲーテッド	ゲーテッド	ゲーテッド
PERx.SYSCLK に接続されたモジュールへのクロック	アクティブ	ゲーテッド	ゲーテッド
WDCLK	アクティブ	アクティブ	CLKSRCCTL1.WDHALTI = 0 の場合、ゲーテッド
PLL	電源供給	電源供給	ソフトウェアは、HALT に入る前に PLL の電源をオフにする必要があります。
INTOSC1	電源供給	電源供給	CLKSRCCTL1.WDHALTI = 0 の場合、電源オフ
INTOSC2	電源供給	電源供給	CLKSRCCTL1.WDHALTI = 0 の場合、電源オフ
フラッシュ ⁽¹⁾	電源供給	電源供給	電源供給
XTAL ⁽²⁾	電源供給	電源供給	電源供給

- (1) フラッシュ・モジュールは、いずれの LPM においても、ハードウェアによってパワーダウンされることはありません。アプリケーションで必要な場合は、ソフトウェアを使用して電源をオフにすることができます。詳細については、「フラッシュ・モジュール」の章を参照してください。
- (2) XTAL は、いずれの LPM においても、ハードウェアによってパワーダウンされることはありません。XTALCR.OSCOFF ビットを 1 に設定すると、ソフトウェアによって電源をオフにできます。XTAL が不要な場合は、アプリケーション実行中の任意の時点でこれを実行できます。

1.10.2 IDLE

IDLE は C28x CPU の標準機能です。このモードでは、CPU クロックはゲートされますが、ペリフェラル・クロックはすべて動作したままになります。そのため、IDLE を使用すると、CPU がペリフェラル・イベントを待機している間、電力を節約できます。

イネーブルな割り込みが発生すると、CPU は IDLE モードからウェークアップします。

IDLE モードに移行するには、LPMCR.LPM を 0x0 に設定し、IDLE 命令を実行します。

イネーブルな割り込みイベントが発生すると、CPU は通常の動作を再開します。

1.10.3 STANDBY

STANDBY はより積極的な低消費電力モードであり、CPU クロックと、CPU の SYSCLK から派生したペリフェラル・クロックの両方をゲートします。ただし、ウォッチドッグはアクティブのままになります。STANDBY は、ウェークアップ信号がペリフェラル入力ではなく外部システム (または CPU サブシステム) から送信されるアプリケーションに最適です。

NMI (またはオプションで、ウォッチドッグ割り込みまたは構成済み GPIO) により、CPU を STANDBY モードからウェークアップできます。GPIO がアクティブ Low に駆動されたときに CPU をウェークアップするように、各 GPIO を構成できます。ウェークアップ時に、CPU は WAKEINT 割り込みを受信します (構成されている場合)。

STANDBY モードに移行するには:

1. LPMCR.LPM を 0x1 に設定します。
2. PIE で WAKEINT 割り込みをイネーブルにします。
3. ウォッチドッグ割り込みウェークアップの場合、LPMCR.WDINTE を 1 に設定し、割り込みを生成するようにウォッチドッグを構成します。
4. GPIO ウェークアップの場合、選択した GPIO を LPM モジュールに接続するように GPIOLPMSEL0 と GPIOLPMSEL1 を設定し、入力認定のための OSCCLK サイクル数を選択するように LPMCR.QUALSTDBY を設定します。
5. IDLE 命令を実行して STANDBY に移行します。

STANDBY モードからウェークアップするには:

1. ウェークアップをトリガするように目的の GPIO を構成します。
2. 選択した GPIO 信号を Low に駆動します。LPMCR レジスタの QUALSTDBY ビットで指定した OSCCLK サイクル数の間、信号は Low に維持される必要があります。この周期中に信号が High にサンプリングされると、カウントが再開されます。

認定周期が終了すると、PLL は CPU への CLKIN をイネーブルにし、WAKEINT 割り込みは PIE ブロックにラッチされます。

CPU が STANDBY モードから復帰し、通常の実行を再開できるようになります。

1.10.4 HALT

HALT は、ほぼすべてのシステム・クロックをゲートし、発振器とアナログ・ブロックをパワーダウンできる、グローバルな低消費電力モードです。

他の C2000™ デバイスとは異なり、HALT モードでは、HALT 移行時に XTAL が自動的にパワーダウンされることはありません。さらに、XTAL の電源がオンになっていない場合、HALT モードからウェークアップしても、XTAL の電源は自動的にオンになりません。アプリケーション・ソフトウェアで必要ないときに XTAL 回路の電源をオン/オフにするために、XTALCR.OSCOFF ビットが追加されました。

HALT モード中の消費電力を最小限に抑える必要があるアプリケーションでは、アプリケーション・ソフトウェアは、HALT に入る前に XTAL の電源をオフにする必要があります。OSCLK ソースが XTAL に設定されている場合、XTALCR.OSCOFF を設定する前に、まず OSSCLK ソースを INTOSC1 または INTOSC2 に切り替える必要があります。

各 GPIO は、システムを HALT からウェークアップするように設定できます。他のウェークアップ・オプションは利用できません。ただし、ウォッチドッグ・タイマーには引き続きクロックを供給することができ、タイムアウト・メカニズムが必要な場合はウォッチドッグ・リセットを生成するように設定できます。ウェークアップ時に、CPU は WAKEINT 割り込みを受信します。

HALT モードに移行するには:

1. PIE で WAKEINT 割り込みをイネーブルにします。
2. LPMCR.LPM を 0x2 に設定します。選択した GPIO を LPM モジュールに接続するように GPIOLPMSEL0 と GPIOLPMSEL1 を設定します。
3. CLKSRCCTL1.WDHALTI を 1 に設定すると、HALT 時にウォッチドッグ・タイマーがアクティブのままになり、INTOSC1 と INTOSC2 がパワーアップ状態のままになります。
4. CLKSRCCTL1.WDHALTI を 0 に設定すると、HALT 時にウォッチドッグ・タイマーがディセーブルになり、INTOSC1 と INTOSC2 がパワーダウンされます。
5. IDLE 命令を実行して HALT に移行します。

IDLE 命令がパイプライン内にある間に割り込みまたは NMI を受信すると、システムは WAKEINT ISR の実行を開始します。HALT からのウェークアップ後、ISR の実行が中断箇所から再開されます。

注

HALT モードに移行する前に、システム PLL がロックされている場合は (SYSPLL.LOCKS = 1)、システム・クロックに接続する必要もあります (PLLCTL1.PLLCLKEN = 1)。そうしないと、デバイスはウェークアップしません。

HALT モードからウェークアップするには:

1. 選択した GPIO を 5us 以上の間 Low に駆動します。これにより、WAKEINT PIE 割り込みがアクティブになります。
2. 再度ウェークアップ GPIO を High に駆動して、SYSPLL のパワーアップを開始します。
3. PLL をロックし、WAKEINT ISR をラッチできるように、16us + 1024 OSCLK サイクル待ちます。
4. WAKEINT ISR を実行します。

デバイスが HALT モードから復帰し、通常の実行を再開できるようになります。

1.11 メモリ・コントローラ・モジュール

このデバイスでは、M0、M1、および LSx RAM は CPU 専用です。

これらの RAM はすべて高度に構成可能であり、異なるマスタからの書き込みアクセスとフェッチ・アクセスを制御できます。すべての専用 RAM で ECC 機能 (データとアドレスの両方) がイネーブルであり、共有 RAM ではパリティ機能 (データとアドレスの両方) がイネーブルです。一部の専用メモリは、セキュア・メモリでもあります。詳細については、「デュアル・コード・セキュリティ・モジュール (DCSM)」の章を参照してください。各 RAM には独自のコントローラがあり、アクセス保護 / セキュリティ関連のチェックと、その RAM の ECC / パリティ機能を処理します。図 1-17 に、これらの RAM の構成を示します。

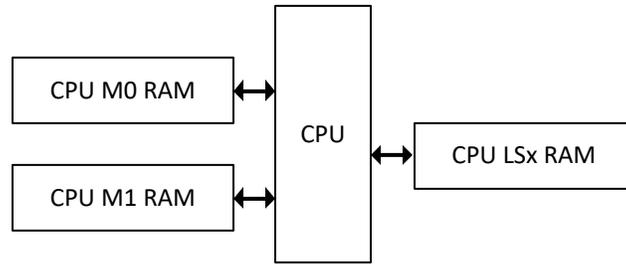


図 1-17. メモリ・アーキテクチャ

1.11.1 機能説明

このセクションでは、このデバイスの専用 RAM と共有 RAM についてさらに定義し、説明します。

1.11.1.1 専用 RAM (Mx RAM)

このデバイスには、2 つの専用 RAM ブロック M0 と M1 があります。M0 および M1 メモリは、CPU と緊密に結合されたメモリの小さなブロックです。CPU のみがこれらのメモリにアクセスできます。他のマスタは、これらのメモリにアクセスできません。

すべての専用 RAM には、ECC 機能があります。

1.11.1.2 ローカル共有 RAM (LSx RAM)

ローカル共有 RAM (LSx RAM) は、パリティ付きのセキュア・メモリです。これらのメモリは CPU 専用です。

1.11.1.3 アクセス保護

M0/M1 を除くすべての RAM ブロックは、異なるレベルの保護を備えています。この機能により、ユーザーは個別のマスタから個別の RAM ブロックへの特定のアクセスをイネーブルまたはディセーブルにできます。読み取りアクセスに対する保護は存在しないため、その RAM ブロックにアクセスできるすべてのマスタからの読み取りは常に許可されます。

以降のセクションでは、このデバイスの RAM ブロックで使用可能な各種の保護について説明します。

注

デバッグ・アクセスの場合、すべての保護がディセーブルになります。

1.11.1.3.1 CPU のフェッチ保護

CPU からのフェッチ・アクセスは、特定のレジスタの `FETCHPROTx` ビットを「1」に設定することで保護できます。CPU のフェッチ保護をイネーブルにしているメモリに CPU がフェッチ・アクセスを行うと、フェッチ保護違反が発生します。

フェッチ保護違反が発生すると、CPU の `ITRAP` が発生します。該当するアクセス違反フラグ・レジスタにフラグがセットされ、アクセス違反が発生したメモリ・アドレスが該当する CPU フェッチ・アクセス違反アドレス・レジスタにラッチされます。

1.11.1.3.2 CPU の書き込み保護

CPU からの書き込みアクセスは、特定のレジスタの `CPUWRPROTx` ビットを「1」に設定することで保護できます。書き込みアクセスが保護されているメモリに CPU が書き込みアクセスを行うと、書き込み保護違反が発生します。

書き込み保護違反が発生すると、書き込みは無視され、該当するアクセス違反フラグ・レジスタにフラグがセットされ、アクセス違反が発生したメモリ・アドレスが該当する CPU 書き込みアクセス違反アドレス・レジスタにラッチされます。また、割り込みイネーブル・レジスタでイネーブルになっている場合、アクセス違反割り込みが生成されます。

1.11.1.3.3 CPU の読み取り保護

読み取り保護違反が発生すると、該当するアクセス違反フラグ・レジスタにフラグがセットされ、アクセス違反が発生したメモリ・アドレスが該当する CPU 読み取りアクセス違反アドレス・レジスタにラッチされます。また、割り込みイネーブル・レジスタでイネーブルになっている場合、アクセス違反割り込みが生成されます。

1.11.1.4 メモリ・エラー検出、訂正、およびエラー処理

これらのデバイスには、安全基準の要件を満たすためのメモリ・エラー検出および訂正機能があります。このような要件により、有限の危険側故障に対する検出メカニズムの追加が保証されます。

このデバイスでは、すべての専用 RAM がエラー訂正コード (ECC) 保護をサポートし、共有 RAM はパリティ保護を備えています。使用される ECC 方式は、シングル・エラー訂正ダブル・エラー検出 (SECCDED) です。使用されるパリティ方式は偶数パリティです。ECC / パリティは、メモリに格納されているデータ・ビットとアドレスをカバーします。

ECC / パリティの計算は、メモリ・コントローラ・モジュール内で実行され、計算されます。ECC / パリティは、データとともにメモリに書き込まれます。ECC / パリティは 16 ビットのデータに対して計算されるため、32 ビットのデータごとに 3 つの 7 ビット ECC コード (または 3 ビットのパリティ) があり、そのうち 2 つはデータ用、3 番目のコードはアドレス用です。

1.11.1.4.1 エラー検出および訂正

メモリからデータを読み取るときに、エラー検出が実行されます。エラー検出は、データおよびアドレスに対して実行されます。パリティ・メモリについては、シングル・ビット・エラーのみが検出されます。一方、ECC メモリについては、シングル・ビット・エラーとともに、ダブル・ビット・エラーも検出されます。これらのエラーには、訂正可能なエラーおよび訂正不可能なエラーがあります。これらのエラーの特性を次に示します。

- パリティ・エラーは常に訂正不可能なエラーです
- シングル・ビット ECC エラーは訂正可能なエラーです
- ダブル・ビット ECC エラーは訂正不可能なエラーです
- アドレス ECC エラーも訂正不可能なエラーです

訂正可能なエラーはメモリ・コントローラ・モジュールによって訂正され、正しいデータがマスタへの読み取りデータとして返されます。また、その正しい値をメモリに書き戻して、同じメモリ・アドレスでさらにシングル・ビット・エラーが発生した場合のダブル・ビット・エラーを防止します。

1.11.1.4.2 エラー処理

訂正可能なエラーが発生するごとに、訂正可能エラー・カウント・レジスタのカウントが 1 ずつインクリメントされます。このカウント・レジスタの値が、訂正可能エラー・スレッシュホールド・レジスタで設定された値と等しくなった場合、訂正可能割り込みイネーブル・レジスタで割り込みがイネーブルであれば、CPU に対する割り込みが生成されます。ユーザーは、システム要件に基づいて、訂正可能エラー・スレッシュホールド・レジスタを構成する必要があります。また、エラーの発生したアドレスがレジスタにラッチされ、フラグもステータス・レジスタに設定されます。

訂正不可能エラーが発生した場合、CPU に対して NMI が生成されます。この場合にも、エラーの発生したアドレスがレジスタにラッチされ、フラグがステータス・レジスタに設定されます。

表 1-15 に、発生する可能性のあるさまざまなエラー状況を示します。これらはソフトウェアで適切に処理する必要があり、提供されているステータスおよび割り込み通知を使用します。

表 1-15. さまざまなシナリオでのエラー処理

アクセス・タイプ	エラー発生箇所	エラーのタイプ	状況の指標	エラー通知
読み取り	メモリからのデータの読み取り	訂正不可能エラー (パリティ RAM のシングル・ビット・エラー または ECC RAM のダブル・ビット・エラー)	あり - CPU 読み取りエラー・アドレス・レジスタ、CPU に返されるデータが正しくない	CPU アクセスに対する NMI
読み取り	メモリからのデータの読み取り	ECC RAM のシングル・ビット・エラー	あり - CPU 読み取りエラー・アドレス・レジスタ、シングル・エラー・カウンタがインクリメント	エラー・カウンタが、ユーザーによりプログラム可能なシングル・エラー・スレッシュホールドに達したときに割り込み発生
読み取り	PIE メモリからのデータの読み取り	パリティ・エラー	あり - PIE パリティ・エラーにより MEM_CFG_REGS のビットがセットされる	MEM_CFG_REGS のビットがセットされる
読み取り	アドレス	アドレス・エラー	あり - CPU 読み取りエラー・アドレス・レジスタ、CPU に返されるデータが正しくない	CPU アクセスに対する CPU への NMI

注

CPU のフェッチ中に訂正不可能エラーが発生した場合、NMI 例外よりも前に ITRAP を取得する可能性があります。NMI が生成される前に、CPU パイプラインに正しくない命令が入ってくるからです。

デバッグ・アクセス中は、訂正可能エラーと訂正不可能エラーの両方がマスクされます。

1.11.1.5 アプリケーション・テスト・フックによる誤り検出と訂正

誤り検出と訂正のロジックは、安全を重視するロジックの一部であるため、安全アプリケーションでは、ロジックが常に正しく動作している状態を確保する必要があります (実行時を含めて)。これを実現するために、テスト・モードが用意されており、ユーザーは (ECC/パリティ・ビットを変更せずに) データ・ビットを変更したり、あるいは ECC/パリティ・ビットを直接変更することができます。この機能を使えば、データに ECC/パリティ・エラーを注入できます。

注

ECC/パリティ・ビットとデータ・ビットのメモリ・マップは同じです。ECC/パリティ・ビットにアクセスするためには、別のテスト・モードを選択する必要があります。テスト・モードでは、すべてのメモリへのアクセス (データおよび ECC/パリティ) は、32 ビット・アクセスに限定して行う必要があります。

表 1-16 および 表 1-17 に、それぞれのアドレスを使用して RAMTEST モードで読み出す場合の ECC/パリティ・ビットのビット・マッピングを示します。

表 1-16. ECC/パリティ・アドレス・マップからの読み取りデータにおける ECC ビットのマッピング

読み取りデータにおけるデータ・ビットの位置	内容 (ECC メモリ)
6:0	下位 16 ビットのデータの ECC コード
7	未使用
14:8	上位 16 ビットのデータの ECC コード
15	未使用
22:16	アドレスの ECC コード
31:23	未使用

表 1-17. ECC/パリティ・アドレス・マップからの読み取りデータにおけるパリティ・ビットのマッピング

読み取りデータにおけるデータ・ビットの位置	内容 (パリティ・メモリ)
0	下位 16 ビットのデータのパリティ
7:1	未使用
8	上位 16 ビットのデータのパリティ
15:9	未使用
16	アドレスのパリティ
31:17	未使用

1.11.1.6 RAM の初期化

初期化されていない RAM 位置からの読み取り / フェッチで ECC またはパリティ・エラーが発生しないように、各メモリ・ブロックには RAM_INIT 機能が用意されています。この機能を使用すると、任意の RAM ブロックを 0x0 データおよびそれぞれの ECC / パリティ・ビットで適宜初期化できます。これを開始するには、INIT レジスタの特定の RAM ブロックの INIT ビットを「1」に設定します。RAM の初期化ステータスを確認するには、SW で、INITDONE レジスタのその RAM ブロックの INITDONE ビットがセットされるかをポーリングする必要があります。このビットがセットされない限り、その RAM メモリ・ブロックにはアクセスしないでください。

注

初期化の実行中は、どのマスタもメモリにアクセスしないようにする必要があります。RAMINITDONE がセットされる前にメモリにアクセスした場合、メモリの読み取り / 書き込みと初期化は正しく行われません。

1.12 JTAG

GEL ファイルは、特定の初期化タスクを実行します。これは、デバッグ環境のユーザーに役立ちます。ただし、スタンドアロンで (エミュレータを接続せずに) 実行すると、このような初期化を実行するための GEL ファイルがないため、アプリケーションは予想どおりに動作しない場合があります。たとえば、GEL ファイルはウォッチドッグをディセーブルにします。アプリケーション内でユーザー・コードによってウォッチドッグが処理されない場合 (またはウォッチドッグをディセーブルにできない場合)、デバッガを使用した場合と使用しない場合では、アプリケーションの動作に違いが生じます。

GEL ファイルで実行される一般的なタスク (ただし、ブート ROM ではない)。

リセット時:

- 一部のデバイスでフラッシュ ECC をディセーブルにします。
 - フラッシュ API 関数を使用している場合にのみ ECC をディセーブルにします。詳細については、『フラッシュ API ユーザー・ガイド』を参照してください。それ以外の場合は、常に ECC をプログラムし、ECC チェックをイネーブルにすることをお勧めします。
- ウォッチドッグをディセーブルにします。
- CLA クロックをイネーブルにします。
- リアルタイム・モードまたは C28x モードを選択します。

再起動時:

- リアルタイム・モードまたは C28x モードを選択します。
- IER と IFR をクリアします。

ターゲット接続時:

- リアルタイム・モードまたは C28x モードを選択します。

1.13 システム制御レジスタの構成の制限事項

システム制御のメモリマップされたレジスタは INTOSC1 クロック・ドメインで動作するため、これらのレジスタへの CPU 書き込みでは、以降の書き込み間に遅延が必要です。遅延がないと、2 番目の書き込みが失われる可能性があります。アプリケーションではこの点を考慮に入れ、表 1-18 に記載されているレジスタへの各書き込み後の NOP 命令の数に遅延を追加する必要があります。以降の書き込み間の遅延を計算する式は、次のとおりです。

$$\text{遅延 (SYSCLK サイクル)} = 3 \times (F_{\text{SYSCLK}} \div F_{\text{INTOSC1}}) + 9$$

たとえば、SYSCLK = 100200MHz の場合、次のようになります。

$$\text{遅延 (SYSCLK サイクル)} = 3 \times (100\text{MHz} \div 10\text{MHz}) + 9 = 39 \text{ SYSCLK サイクル}$$

表 1-18. 影響を受けるシステム制御レジスタ

書き込みのたびに遅延を必要とするレジスタ
CLBCLKCTL
PERCLKDIVSEL
SYSCLKDIVSEL
SYSPLLCTL1
SYSPLLMULT
WDCR
XCLKOUTDIVSEL
XTALCR
CLKSRCCTL1
CLKSRCCTL2
CLKSRCCTL3
CPU1TMR2CTL (TMR2CLKCTL)

1.14 ソフトウェア

1.14.1 SYSCTL の例

注:これらの例は、**C2000Ware** のインストール先の次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/sysctl

これらの例へのクラウド・アクセスは、dev.ti.com の **C2000Ware** の例のリンクから利用できます。

1.14.1.1 クロック消失検出 (MCD)

ファイル:sysctl_ex1_missing_clock_detection.c

この例では、クロック消失検出機能とその処理方法を示します。OSCCLK と MCD モジュールの接続を切断して MCD をシミュレートすると、NMI が生成されます。この NMI では、ISR で処理されるクロック障害が原因で、MCD が生成されたと判定されます。

MCD より前のクロック周波数は、デバイスの初期化に従って設定されます (120MHz)。MCD 後の周波数は 10MHz または INTOSC1 に移行します。

また、この例では、クロック消失の検出後に PLL をロックする方法も示します。最初にクロック・ソースを明示的に INTOSC1 に切り替え、クロック消失検出回路をリセットしてから、PLL を再ロックします。再ロック後のクロック周波数は 100MHz になりますが、INTOSC1 をクロック・ソースとして使用します。

外部接続

- なし。

監視変数

- fail** - クロック消失が検出されなかったか、正しく処理されなかったことを示します。
- mcd_clkfail_isr** - クロック消失障害が原因で NMI がトリガされ、それを処理するために ISR が呼び出されたことを示します。
- mcd_detect** - クロック消失が検出されたことを示します。
- result** - クロック消失検出が正常に処理されたことを示すステータス

1.14.1.2 XCLKOUT (外部クロック出力) 構成

ファイル:sysctl_ex2_xclkout_config.c

この例では、デバッグとテストの目的で、外部ピンを経由して内部クロックを観測するために XCLKOUT ピンを構成する方法を示します。

この例では、XCLKOUT クロック・ソースとして INTOSC1 を使用し、分周器を 8 に設定しています。XCLKOUT の予測される周波数 = (INTOSC1 周波数)/8 = 10/8 = 1.25MHz

オシロスコープを使用して、GPIO16 の XCLKOUT を表示します。

1.14.2 TIMER の例

注:これらの例は、**C2000Ware** のインストール先の次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/timer

これらの例へのクラウド・アクセスは、dev.ti.com の **C2000Ware** の例のリンクから利用できます。

1.14.2.1 CPU タイマ

ファイル:timer_ex1_cputimers.c

この例では、CPU Timer0、1、2 を設定し、タイマが割り込みをアサートするたびにカウンタをインクリメントします。

外部接続

- なし

監視変数

- cpuTimer0IntCount
- cpuTimer1IntCount
- cpuTimer2IntCount

1.14.3 MEMCFG の例

注:これらの例は、[C2000Ware](#) のインストール先の次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/memcfg

これらの例へのクラウド・アクセスは、dev.ti.com の [C2000Ware の例](#) のリンクから利用できます。

1.14.3.1 修正可能なメモリ・エラーと修正不可能なメモリ・エラーの処理

ファイル:memcfg_ex1_error_handling.c

この例では、メモリの読み取り / 書き込みに各種誤動作が発生した場合のエラー処理を示します。CPU の読み取り / 書き込み違反、修正可能なメモリ・エラー、および修正不可能なメモリ・エラーが発生した場合のエラー処理が示されています。修正可能なメモリ・エラーおよび違反の場合は CPU への SYS_INT 割り込みを生成できますが、修正不可能なエラーの場合は NMI が生成されます。

外部接続

- なし

監視変数

- testStatusGlobal - テストが正常に完了した場合は TEST_PASS と等価で、それ以外の場合は値が TEST_FAIL に設定されます。
- errCountGlobal - エラー・カウンタ

1.14.4 INTERRUPT の例

注:これらの例は、[C2000Ware](#) のインストール先の次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/interrupt

これらの例へのクラウド・アクセスは、dev.ti.com の [C2000Ware の例](#) のリンクから利用できます。

1.14.4.1 外部割り込み (ExternalInterrupt)

ファイル:interrupt_ex1_external.c

このプログラムは、GPIO0 を XINT1、GPIO1 を XINT2 として設定します。割り込みをトリガするために、他の 2 つの GPIO 信号が使用されます (GPIO10 は XINT1 をトリガし、GPIO11 は XINT2 をトリガします)。プログラムが正常に動作するには、これらの信号を外部から接続する必要があります。

XINT1 入力は SYSCLKOUT に同期されます。

XINT2 の認定は長く、510*SYSCLKOUT ごとに 6 サンプルとなります。

GPIO16 は割り込み外で High になり、割り込み内で Low になります。この信号はスコープで監視できます。

各割り込みは、最初に XINT1、次に XINT2 の順に起動されます。

外部接続

- GPIO10 を GPIO0 に接続します。GPIO0 は XINT1 に割り当てられます。
- GPIO11 を GPIO1 に接続します。GPIO1 は XINT2 に割り当てられます。

オシロスコープを使用して GPIO16 を監視します。GPIO16 は ISR 外で High になり、各 ISR 内で Low になります。

監視変数

- xint1Count - XINT1 割り込みを経由する回数
- xint2Count - XINT2 割り込みを経由する回数
- loopCount - アイドル・ループを経由する回数

1.14.4.2 I2C、SCI、SPI デジタル・ループバックの複数の割り込み処理

ファイル: `interrupt_ex2_with_i2c_sci_spi_loopback.c`

このプログラムを使用して、I2C、SCI、SPI デジタル・ループバックのような複数の通信ペリフェラルをすべて 1 つの例で使用する場合に複数の割り込みを処理する方法を示します。データ転送は FIFO 割り込みで行われます。

これらのモジュールの内部ループバック・テスト・モードが使用されます。TX と RX の両方の FIFO とそれらの割り込みが使用されます。ブート・モード・ピン構成以外に必要な他のハードウェア構成はありません。

データ・ストリームが送信され、受信したストリームと比較されます。I2C と SCI の場合、送信されるデータは次のようになります。

```
0000 0001
0001 0002
0002 0003
....
```

```
00FE 00FF
00FF 0000
```

など。

SPI の場合、送信されるデータは次のようになります。

```
0000 0001
0001 0002
0002 0003
```

....

```
FFFE FFFF
FFFF 0000
```

など。

このパターンは永続的に繰り返されます。

外部接続

- なし

監視変数

- `sDatai2cA` - I2C 経由で送信するデータ
- `rDatai2cA` - 受信した I2C データ
- `rDataPoint` - エラー・チェック用に受信 I2C ストリームの最後の位置を追跡するために使用します。
- `sDataspiA` - SPI 経由で送信するデータ
- `rDataspiA` - 受信した SPI データ
- `rDataPointspiA` - エラー・チェック用に受信 SPI ストリームの最後の位置を追跡するために使用します。
- `sDatasciA` - 送信する SCI データ
- `rDatasciA` - 受信した SCI データ
- `rDataPointA` - SCI データ・ストリーム内の現在位置を追跡します。これを使用して、受信データをチェックします。

1.14.4.3 CPU タイマ割り込みのソフトウェア優先度設定

ファイル: `interrupt_ex3_sw_prioritization.c`

この例では、CPU タイマ割り込みによる割り込みのソフトウェア優先度設定を示します。割り込みのソフトウェア優先度設定を行うには、割り込みのネストをイネーブルにします。

このデバイスでは、CPU タイマ 0、1、2 のハードウェア優先度の設定は、タイマ 0 の優先度が最も高く、タイマ 2 の優先度が最も低くなっています。この例では、ソフトウェアで CPU タイマ 0、1、2 の優先度を設定し、ソフトウェアでタイマ 2 の優先度を最も高く、タイマ 0 の優先度を最も低くして、実行順序のトレースを出力します。

ほとんどのアプリケーションでは、割り込みのハードウェア優先度設定で十分です。カスタムの優先度設定を必要とするアプリケーションのために、この例では、ソフトウェアを使用してこの設定を行う方法を示します。ユーザー固有の優先度は、`sw_prioritized_isr_level.h` ヘッダー・ファイルで設定できます。

割り込みのネストをイネーブルにするには、ISR で次のシーケンスに従う必要があります。ステップ 1: グローバル優先度を設定します。ユーザー優先度の高い CPU 割り込みを処理できるように、IER レジスタを変更します。注: 現時点で、IER はすでにスタックに保存されています。ステップ 2: グループの優先度を設定します。(オプション) ユーザー設定の優先度の高いグループ割り込みを処理できるように、適切な PIEIERx レジスタを変更します。この ISR で処理されるもの以外のグループから PIEIER レジスタ・ビットをクリアしないでください。クリアすると、誤った割り込みが発生する可能性があります。ステップ 3: 割り込みをイネーブルにします。これを行うには、次の 3 つのステップを実行します。a. PIEACK ビットをクリアします。b. 少なくとも 1 サイクル待ちます。c. INTM ビットをクリアします。ステップ 4: ISR のメイン部分を実行します。ステップ 5: INTM をセットして割り込みをディセーブルにします。ステップ 6: PIEIERx を復元します (ステップ 2 に応じてオプション)。ステップ 7: ISR から復帰します。

C28x デバイスでの割り込みのネストの詳細については、以下のリンクを参照してください。<C2000Ware>.html

外部接続

- なし

監視変数

- tracelSR - ISR が実行される順序を示します。

1.14.4.4 EPWM リアルタイム割り込み

ファイル: interrupt_ex4_epwm_realtime_interrupt.c

この例では、ePWM1 タイマを設定し、ISR が実行されるたびにカウンタをインクリメントします。ePWM 割り込みは、リアルタイム・モード機能とリアルタイム割り込み機能を示すために、タイム・クリティカルとして設定できます。

この例では 2 つの LED を使用しています。LED1 はメイン・ループでトグルされ、LED2 は EPWM タイマ割り込みでトグルされます。ePWM1 割り込みがタイム・クリティカルになり、HALT コマンド後にリアルタイム・モードで動作できるようにするには、FREE_SOFT ビットと DBGIER.INT3 ビットを設定する必要があります。

この例の実行方法

- 以下で説明するように監視変数を追加し、連続更新をイネーブルにします。
- リアルタイム・モードをイネーブルにします ([Run]->[Advanced]->[Enable Silicon Real-time Mode])。
- 初期状態では、DBGIER レジスタは 0 に設定され、EPWM エミュレーション・モードは EPWM_EMULATION_STOP_AFTER_NEXT_TB (FREE_SOFT = 0) に設定されています。
- アプリケーションの実行時に、両方の LED がトグルされ、監視変数 EPwm1TimerIntCount、EPwm1Regs.TBCTR が更新されることがわかります。
- アプリケーションが停止すると、両方の LED のトグルが停止し、監視変数は一定のままになります。デバッガが停止すると、EPWM カウンタは停止します。
- デバッガの停止中に EPWM カウンタの実行をイネーブルにするには、エミュレーション・モードを EPWM_EMULATION_FREE_RUN (FREE_SOFT = 2) に設定します。EPwm1Regs.TBCTR は動作していますが、EPwm1TimerIntCount は一定のままであることがわかります。つまり、EPWM カウンタは動作していますが、ISR は処理されていません。
- リアルタイム割り込みをイネーブルにするには、DBGIER.INT3 = 1 に設定します (EPWM1 割り込みは PIE グループ 3 の一部です)。EPwm1TimerIntCount がインクリメントされ、LED がトグルを開始していることがわかります。EPWM ISR は、デバッガの停止中も処理されます。

詳細については、以下のビデオをご覧ください。C2000 リアルタイム機能

外部接続

- なし

監視変数

- EPwm1TimerIntCount - EPWM1 ISR カウンタ
- EPwm1Regs.TBCTR.TBCTR - EPWM1 タイム・ベース・カウンタ
- EPwm1Regs.TBCTL.FREE_SOFT - 2 に設定すると、フリー・ランがイネーブルになります。
- DBGIER.INT3 - 1 に設定すると、リアルタイム割り込みがイネーブルになります。

1.14.5 LPM の例

注:これらの例は、[C2000Ware](#) のインストール先の次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/lpm

これらの例へのクラウド・アクセスは、dev.ti.com の [C2000Ware](#) の例のリンクから利用できます。

1.14.5.1 低消費電力モード: デバイスの IDLE モードと GPIO を使用したウェークアップ

ファイル: lpm_ex1_idlewake_gpio.c

この例では、デバイスを IDLE モードにし、GPIO0 の立ち下がりエッジでトリガされる XINT1 を使用してデバイスを IDLE からウェークアップします。

ウェークアップを行うには、外部エージェントによって GPIO0 ピンを High から Low にする必要があります。GPIO0 は XINT1 ピンとして構成され、立ち下がりエッジの検出時に XINT1 割り込みをトリガします。

最初に、GPIO0 を外部から High にします。XINT1 割り込みをトリガしてデバイスを IDLE モードからウェークアップするには、GPIO0 を Low (立ち下がりエッジ) にします。ウェークアップ・プロセスは、デバイスのデータシートに記載されている時間、GPIO0 を Low に保持するとすぐに開始されます。

GPIO1 は、IDLE モードに入る前に High にされ、外部割り込み ISR では Low にされます。

外部接続

- デバイスをウェークアップするには、GPIO0 を Low にする必要があります。
- デバイス・ウェークアップ時に、GPIO1 は Low になり、LED1 は点滅を開始します。

1.14.5.2 低消費電力モード: デバイスの IDLE モードとウォッチドッグを使用したウェークアップ

ファイル: lpm_ex2_idlewake_watchdog.c

この例では、デバイスを IDLE モードにし、ウォッチドッグ・タイマを使用してデバイスを IDLE からウェークアップします。

ウォッチドッグ・タイマがオーバーフローし、割り込みをトリガすると、デバイスは IDLE モードからウェークアップします。カウンタのオーバーフロー時間を変更するには、ウォッチドッグ・タイマのプリスケールを設定します。

GPIO1 は、IDLE モードに入る前に High にプルされ、ウェークアップ ISR では Low にされます。

外部接続

- デバイス・ウェークアップ時に、GPIO1 は Low になり、LED1 は点滅を開始します。

1.14.5.3 低消費電力モード: デバイスの STANDBY モードと GPIO を使用したウェークアップ

ファイル: lpm_ex3_standbywake_gpio.c

この例では、デバイスを STANDBY モードにします。STANDBY モードで消費電流を可能な限り低くする必要がある場合は、デバイスが STANDBY モードの間、JTAG コネクタをデバイス・ボードから取り外す必要があります。

この例では、デバイスを STANDBY モードにし、LPM ウェークアップ・ピンを使用してデバイスを STANDBY からウェークアップします。

ピン GPIO0 は LPM ウェークアップ・ピンとして構成され、Low パルスの検出時に WAKEINT 割り込みをトリガします。最初に、GPIO0 を外部から High にします。デバイスを STANDBY モードからウェークアップするには、GPIO0 を少なくとも (2+QUALSTDBY) OSCLK の間 Low にしてから、再度 High にします。

この例では、GPIO0 を使用してデバイスを STANDBY からウェークアップします。GPIO0 は、ピンで Low パルス (信号が High->Low->High になる) が検出されると、デバイスを STANDBY モードからウェークアップします。ウェークアップを行うには、このピンに外部エージェントからパルスが供給される必要があります。

GPIO1 は、STANDBY モードに入る前に High にされ、ウェークアップ ISR では Low にされます。

外部接続

- デバイスをウェークアップするには、GPIO0 を Low にする必要があります。

- デバイス・ウェークアップ時に、GPIO1 は Low になり、LED1 は点滅を開始します。

1.14.5.4 低消費電力モード: デバイスの STANDBY モードとウォッチドッグを使用したウェークアップ

ファイル: `lpm_ex4_standbywake_watchdog.c`

この例では、デバイスを STANDBY モードにします。STANDBY モードで消費電流を可能な限り低くする必要がある場合は、デバイスが STANDBY モードの間、JTAG コネクタをデバイス・ボードから取り外す必要があります。

この例では、デバイスを STANDBY モードにし、ウォッチドッグ・タイマを使用してデバイスを STANDBY からウェークアップします。

ウォッチドッグ・タイマがオーバーフローし、割り込みをトリガすると、デバイスは STANDBY モードからウェークアップします。ISR では、GPIO1 は Low にプルされます。GPIO1 がトグルされ、デバイスが STANDBY モードから復帰したことが示されます。カウンタのオーバーフロー時間を変更するには、ウォッチドッグ・タイマのプリスケールを設定します。

GPIO1 は、STANDBY モードに入る前に High にされ、ウェークアップ ISR では Low にされます。

外部接続

- デバイス・ウェークアップ時に、GPIO1 は Low になり、LED1 は点滅を開始します。

1.14.5.5 低消費電力モード: HALT モードと GPIO を使用したウェークアップ

ファイル: `lpm_ex5_haltwake_gpio.c`

この例では、デバイスを HALT モードにします。HALT モードで消費電流を可能な限り低くする必要がある場合は、デバイスが HALT モードの間、JTAG コネクタをデバイス・ボードから取り外す必要があります。

HALT モード中の消費電力を最小限に抑える必要があるアプリケーションでは、アプリケーション・ソフトウェアは、XTALCR.OSCOFF ビットをセットするか、または `driverlib` 関数 `SysCtl_turnOffOsc(SYSCTL_OSCSRC_XTAL)` を使用して、HALT に入る前に XTAL の電源をオフにする必要があります。OSCCLK ソースが XTAL に設定されている場合、XTALCR.OSCOFF を設定する前に、まず OSSCLK ソースを INTOSC1 または INTOSC2 に切り替える必要があります。

この例では、デバイスを HALT モードにし、LPM ウェークアップ・ピンを使用してデバイスを HALT からウェークアップします。

ピン GPIO0 は LPM ウェークアップ・ピンとして構成され、Low パルスの検出時に WAKEINT 割り込みをトリガします。ウェークアップを行うには、外部エージェントによって GPIO0 ピンを High から Low にする必要があります。

GPIO1 は、STANDBY モードに入る前に High にされ、ウェークアップ ISR では Low にされます。

外部接続

- デバイス・ウェークアップ時に、GPIO1 は Low になり、LED1 は点滅を開始します。

1.14.5.6 低消費電力モード: HALT モードとウェークアップ

ファイル: `lpm_ex6_haltwake_gpio_watchdog.c`

この例では、デバイスを HALT モードにします。HALT モードで消費電流を可能な限り低くする必要がある場合は、デバイスが HALT モードの間、JTAG コネクタをデバイス・ボードから取り外す必要があります。

HALT モード中の消費電力を最小限に抑える必要があるアプリケーションでは、アプリケーション・ソフトウェアは、XTALCR.OSCOFF ビットをセットするか、または `driverlib` 関数 `SysCtl_turnOffOsc(SYSCTL_OSCSRC_XTAL)` を使用して、HALT に入る前に XTAL の電源をオフにする必要があります。OSCCLK ソースが XTAL に設定されている場合、XTALCR.OSCOFF を設定する前に、まず OSSCLK ソースを INTOSC1 または INTOSC2 に切り替える必要があります。

この例では、デバイスを HALT モードにし、LPM ウェークアップ・ピンを使用してデバイスを HALT からウェークアップします。

ピン GPIO0 は LPM ウェークアップ・ピンとして構成され、Low パルスの検出時に WAKEINT 割り込みをトリガします。ウェークアップを行うには、外部エージェントによって GPIO0 ピンを High から Low にする必要があります。

この例では、ウォッチドッグ・タイマにクロックが供給され、タイムアウト・メカニズムとしてウォッチドッグ・リセットを生成するように設定されています。

GPIO1 は、STANDBY モードに入る前に High にされ、ウェークアップ ISR では Low にされます。

外部接続

- デバイス・ウェークアップ時に、GPIO1 は Low になり、LED1 は点滅を開始します。

1.14.6 WATCHDOG の例

注:これらの例は、[C2000Ware](#) のインストール先の次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/watchdog

これらの例へのクラウド・アクセスは、dev.ti.com の [C2000Ware の例](#)のリンクから利用できます。

1.14.6.1 ウォッチドッグ

ファイル: watchdog_ex1_service.c

この例では、ウォッチドッグを処理する方法、またはウォッチドッグを使用してウェークアップ割り込みを生成する方法を示します。デフォルトでは、この例は WAKE 割り込みを生成します。ウォッチドッグを処理し、割り込みを生成しないようにするには、メインの for ループにある SysCtl_serviceWatchdog() 行のコメントを解除します。

外部接続

- なし。

監視変数

- wakeCount - ウォッチドッグ ISR に移行した回数
- loopCount - ISR でないときに実行されるループの数

1.15 システム制御レジスタ

1.15.1 SYSCTRL ベース・アドレス表

表 1-19. SYSCTRL ベース・アドレス表

ビット・フィールド名		DriverLib 名	ベース・アドレス	パイプライン保護
インスタンス	構造			
CpuTimer0Regs	CPUTIMER_REGS	CPUTIMER0_BASE	0x0000_0C00	-
CpuTimer1Regs	CPUTIMER_REGS	CPUTIMER1_BASE	0x0000_0C08	-
CpuTimer2Regs	CPUTIMER_REGS	CPUTIMER2_BASE	0x0000_0C10	-
PieCtrlRegs	PIE_CTRL_REGS	PIECTRL_BASE	0x0000_0CE0	-
PieVectTable	PIE_VECT_TABLE	PIEVECTTABLE_BASE	0x0000_0D00	-
WdRegs	WD_REGS	WD_BASE	0x0000_7000	あり
NmiIntruptRegs	NMI_INTRUPT_REGS	NMI_BASE	0x0000_7060	あり
XintRegs	XINT_REGS	XINT_BASE	0x0000_7070	あり
SyncSocRegs	SYNC_SOC_REGS	SYNCSOC_BASE	0x0000_7940	あり
DevCfgRegs	DEV_CFG_REGS	DEVCFG_BASE	0x0005_D000	あり
ClkCfgRegs	CLK_CFG_REGS	CLKCFG_BASE	0x0005_D200	あり
CpuSysRegs	CPU_SYS_REGS	CPUSYS_BASE	0x0005_D300	あり
SysStatusRegs	SYS_STATUS_REGS	SYSTAT_BASE	0x0005_D400	あり
MemCfgRegs	MEM_CFG_REGS	MEMCFG_BASE	0x0005_F400	あり
AccessProtectionRegs	ACCESS_PROTECTION_REGS	ACCESSPROTECTION_BASE	0x0005_F500	あり
MemoryErrorRegs	MEMORY_ERROR_REGS	MEMORYERROR_BASE	0x0005_F540	あり
TestErrorRegs	TEST_ERROR_REGS	TESTERROR_BASE	0x0005_F590	あり
UidRegs	UID_REGS	UID_BASE	0x0007_1140	-

1.15.2 ACCESS_PROTECTION_REGS レジスタ

表 1-20 に、ACCESS_PROTECTION_REGS レジスタのメモリマップされたレジスタを示します。表 1-20 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-20. ACCESS_PROTECTION_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
20h	MAVFLG	マスタ・アクセス違反フラグ・レジスタ		表示
22h	MAVSET	マスタ・アクセス違反フラグ・セット・レジスタ	EALLOW	表示
24h	MAVCLR	マスタ・アクセス違反フラグ・クリア・レジスタ	EALLOW	表示
26h	MAVINTEN	マスタ・アクセス違反割り込みイネーブル・レジスタ	EALLOW	表示
28h	MCPUFAVADDR	マスタ CPU フェッチ・アクセス違反アドレス		表示
2Ah	MCPUWRAVADDR	マスタ CPU 書き込みアクセス違反アドレス		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-21 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-21. ACCESS_PROTECTION_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W1S	書き込み 1 でセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.2.1 MAVFLG レジスタ (オフセット = 20h) [リセット = 0h]

図 1-18 に MAVFLG を示し、表 1-22 にその説明を示します。

概略表に戻ります。

マスタ・アクセス違反フラグ・レジスタ

図 1-18. MAVFLG レジスタ

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

表 1-22. MAVFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R	0h	予約済み
2	RESERVED	R	0h	予約済み
1	CPUWRITE	R	0h	マスタ CPU 書き込みアクセス違反フラグ: 0:違反なし。 1:アクセス違反が発生しました。 リセット・タイプ:SYRSn
0	CPUFETCH	R	0h	マスタ CPU フェッチ・アクセス違反フラグ: 0:違反なし。 1:アクセス違反が発生しました。 リセット・タイプ:SYRSn

1.15.2.2 MAVSET レジスタ (オフセット = 22h) [リセット = 0h]

図 1-19 に MAVSET を示し、表 1-23 にその説明を示します。

概略表に戻ります。

マスタ・アクセス違反フラグ・セット・レジスタ

図 1-19. MAVSET レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	CPUWRITE	CPUFETCH
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-23. MAVSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	CPUWRITE	R-0/W1S	0h	0:動作なし。 1:MAVFLG レジスタの CPU 書き込みアクセス違反フラグがセットされ、イネーブルの場合は割り込みが生成されます。 リセット・タイプ:SYRSn
0	CPUFETCH	R-0/W1S	0h	0:動作なし。 1:MAVFLG レジスタの CPU フェッチ・アクセス違反フラグがセットされ、イネーブルの場合は割り込みが生成されます。 リセット・タイプ:SYRSn

1.15.2.3 MAVCLR レジスタ (オフセット = 24h) [リセット = 0h]

図 1-20 に MAVCLR を示し、表 1-24 にその説明を示します。

概略表に戻ります。

マスタ・アクセス違反フラグ・クリア・レジスタ

図 1-20. MAVCLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	CPUWRITE	CPUFETCH
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-24. MAVCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	CPUWRITE	R-0/W1S	0h	0:動作なし。 1:MAVFLG レジスタの CPU 書き込みアクセス違反フラグがクリアされます。 リセット・タイプ:SYSRSn
0	CPUFETCH	R-0/W1S	0h	0:動作なし。 1:MAVFLG レジスタの CPU フェッチ・アクセス違反フラグがクリアされます。 リセット・タイプ:SYSRSn

1.15.2.4 MAVINTEN レジスタ (オフセット = 26h) [リセット = 0h]

図 1-21 に MAVINTEN を示し、表 1-25 にその説明を示します。

概略表に戻ります。

マスタ・アクセス違反割り込みイネーブル・レジスタ

図 1-21. MAVINTEN レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	CPUWRITE	CPUFETCH
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-25. MAVINTEN レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	CPUWRITE	R/W	0h	0: CPU 書き込みアクセス違反割り込みがディセーブル。 1: CPU 書き込みアクセス違反割り込みがイネーブル。 リセット・タイプ: SYSRSn
0	CPUFETCH	R/W	0h	0: CPU フェッチ・アクセス違反割り込みがディセーブル。 1: CPU フェッチ・アクセス違反割り込みがイネーブル。 リセット・タイプ: SYSRSn

1.15.2.5 MCPUFAVADDR レジスタ (オフセット = 28h) [リセット = 0h]

図 1-22 に MCPUFAVADDR を示し、表 1-26 にその説明を示します。

概略表に戻ります。

マスタ CPU フェッチ・アクセス違反アドレス

図 1-22. MCPUFAVADDR レジスタ

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

表 1-26. MCPUFAVADDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	MCPUFAVADDR	R	0h	このレジスタは、マスタ CPU フェッチ・アクセス違反が発生したアドレス位置をキャプチャします。 リセット・タイプ: SYRSn

1.15.2.6 MCPUWRAVADDR レジスタ (オフセット = 2Ah) [リセット = 0h]

図 1-23 に MCPUWRAVADDR を示し、表 1-27 にその説明を示します。

概略表に戻ります。

マスタ CPU 書き込みアクセス違反アドレス

図 1-23. MCPUWRAVADDR レジスタ

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

表 1-27. MCPUWRAVADDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	MCPUWRAVADDR	R	0h	このレジスタは、マスタ CPU 書き込みアクセス違反が発生したアドレスの場所をキャプチャします。 リセット・タイプ: SYSRSn

1.15.3 CLK_CFG_REGS レジスタ

表 1-28 に、CLK_CFG_REGS レジスタのメモリマップされたレジスタを示します。表 1-28 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-28. CLK_CFG_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
2h	CLKCFGLOCK1	CLKCFG レジスタのロック・ビット	EALLOW	表示
8h	CLKSRCCTL1	クロック・ソース制御レジスタ -1	EALLOW	表示
Ah	CLKSRCCTL2	クロック・ソース制御レジスタ -2	EALLOW	表示
Ch	CLKSRCCTL3	クロック・ソース制御レジスタ -3	EALLOW	表示
Eh	SYSPLLCTL1	SYSPLL 制御レジスタ -1	EALLOW	表示
14h	SYSPLLMULT	SYSPLL 連倍器レジスタ	EALLOW	表示
16h	SYSPLLSTS	SYSPLL ステータス・レジスタ		表示
22h	SYSCLKDIVSEL	システム・クロック分周器選択レジスタ	EALLOW	表示
28h	XCLKOUTDIVSEL	XCLKOUT 分周器選択レジスタ	EALLOW	表示
2Ch	LOSPCP	低速クロック・ソース・プリスケラ	EALLOW	表示
2Eh	MDCDR	クロック消失検出制御レジスタ	EALLOW	表示
30h	X1CNT	X1 クロックの 10 ビット・カウンタ		表示
32h	XTALCR	XTAL 制御レジスタ	EALLOW	表示
3Ah	XTALCR2	パッド初期化用の XTAL 制御レジスタ	EALLOW	表示
3Ch	CLKFAILCFG	クロック故障要因構成	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-29 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-29. CLK_CFG_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R -0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	書き込み 1 でセット
WSonce	W Sonce	書き込み 1 回のみセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。

**表 1-29. CLK_CFG_REGS のアクセス・タイプ・コード
(continued)**

アクセス・タイプ	コード	説明
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.3.1 CLKCFGLOCK1 レジスタ (オフセット = 2h) [リセット = 0h]

図 1-24 に CLKCFGLOCK1 を示し、表 1-30 にその説明を示します。

概略表に戻ります。

CLKCFG レジスタのロック・ビット

注:

[1] このレジスタのいずれかのビットは、一度セットした後は、CPU1.SYSRSn によってのみクリアできます。このレジスタのいずれかのビットに 0 を書き込んでも、何も効果はありません。

[2] ロック・メカニズムは、書き込みにも適用されます。ロック保護機能を持つレジスタへの読み取りは常に許可されます

図 1-24. CLKCFGLOCK1 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						RESERVED	XTALCR
R-0-0h						R/WOnce-0h	R/WOnce-0h
15	14	13	12	11	10	9	8
LOSPCP	RESERVED	RESERVED	RESERVED	SYSCLKDIVSEL	RESERVED	RESERVED	RESERVED
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R-0-0h	R-0-0h
7	6	5	4	3	2	1	0
RESERVED	SYSPLLMULT	RESERVED	RESERVED	SYSPLLCTL1	CLKSRCCTL3	CLKSRCCTL2	CLKSRCCTL1
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h

表 1-30. CLKCFGLOCK1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-18	RESERVED	R-0	0h	予約済み
17	RESERVED	R/WOnce	0h	予約済み
16	XTALCR	R/WOnce	0h	XTALCR および XTAL CR2 レジスタの共通ロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
15	LOSPCP	R/WOnce	0h	LOSPCP レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
14	RESERVED	R/WOnce	0h	予約済み
13	RESERVED	R/WOnce	0h	予約済み
12	RESERVED	R/WOnce	0h	予約済み
11	SYSCLKDIVSEL	R/WOnce	0h	SYSCLKDIVSEL レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
10	RESERVED	R/WOnce	0h	予約済み
9	RESERVED	R-0	0h	予約済み
8	RESERVED	R-0	0h	予約済み
7	RESERVED	R/WOnce	0h	予約済み

表 1-30. CLKCFGLOCK1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	SYSPLLMULT	R/WOnce	0h	SYSPLLMULT レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
5	RESERVED	R/WOnce	0h	予約済み
4	RESERVED	R/WOnce	0h	予約済み
3	SYSPLLCTL1	R/WOnce	0h	SYSPLLCTL1 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
2	CLKSRCCTL3	R/WOnce	0h	CLKSRCCTL3 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
1	CLKSRCCTL2	R/WOnce	0h	CLKSRCCTL2 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
0	CLKSRCCTL1	R/WOnce	0h	CLKSRCCTL1 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn

1.15.3.2 CLKSRCCTL1 レジスタ (オフセット = 8h) [リセット = 0h]

図 1-25 に CLKSRCCTL1 を示し、表 1-31 にその説明を示します。

概略表に戻ります。

クロック・ソース制御レジスタ -1

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-25. CLKSRCCTL1 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
INTOSC2CLKMODE	RESERVED	WDHALTI	RESERVED	RESERVED	RESERVED	OSCCLSRCSEL	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h	R/W-0h	

表 1-31. CLKSRCCTL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-9	RESERVED	R-0	0h	予約済み
8	RESERVED	R/W	0h	予約済み
7	INTOSC2CLKMODE	R/W	0h	INTOSC2 の内部抵抗モードと外部抵抗モードを選択 0 -内部抵抗モード 1 -外部抵抗モード (ExtR) リセット・タイプ: XRSn
6	RESERVED	R/W	0h	予約済み
5	WDHALTI	R/W	0h	ウォッチドッグ HALT モード無視ビット: このビットは、WD が HALT モードで機能するかどうかを決定します。 0 = WD は HALT モードでは機能しません。システムが HALT モードに移行すると、WD へのクロックが遮断されます。 1 = WD は HALT モードで機能します。WD へのクロックは遮断されません リセット・タイプ: XRSn
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R-0	0h	予約済み

表 1-31. CLKSRCCTL1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
1-0	OSCCLKSRCSEL	R/W	0h	<p>発振器クロック・ソース選択ビット:このビットは、OSCCLK のソースを選択します。</p> <p>00 = INTOSC2 (リセット時のデフォルト)</p> <p>01 = 外部発振器 (XTAL)</p> <p>10 = INTOSC1</p> <p>11 = 予約済み (INTOSC1 になります)</p> <p>電源オン時または XRSn の後には、INTOSC2 がデフォルトで選択されます。ユーザーがこれらのビットを使用してクロック・ソースを変更するたびに、SYSPLLMULT[13:0] レジスタは強制的に 0 になり、PLL はバイパスされてパワーダウンされます。これにより、PLL のオーバーシュートの可能性を防止します。次に、適切な通倍器を構成するために、SYSPLLMULT レジスタに書き込む必要があります。</p> <p>ユーザーは、SYSPLLMULT に書き込む前、または今までのクロック・ソースを無効にする前に、変更が完了するのを待つため、10 OSCCLK サイクル待機する必要があります。</p> <p>注:</p> <p>[1] INTOSC1 は、クロック消失を検出した後にのみ使用することを推奨します。クロック消失を検出した後、INTOSC1 (バックアップ・クロック・ソース) で PLL を再ロックする場合には、MCLKCLR を実行して PLL をロックできます。</p> <p>リセット・タイプ: XRSn</p>

1.15.3.3 CLKSRCCTL2 レジスタ (オフセット = Ah) [リセット = 0h]

図 1-26 に CLKSRCCTL2 を示し、表 1-32 にその説明を示します。

概略表に戻ります。

クロック・ソース制御レジスタ -2

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-26. CLKSRCCTL2 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						RESERVED	
R-0-0h						R/W-0h	
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED		CANABCLKSEL		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-32. CLKSRCCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-18	RESERVED	R-0	0h	予約済み
17-16	RESERVED	R/W	0h	予約済み
15-14	RESERVED	R/W	0h	予約済み
13-12	RESERVED	R/W	0h	予約済み
11-10	RESERVED	R/W	0h	予約済み
9-8	RESERVED	R/W	0h	予約済み
7-6	RESERVED	R/W	0h	予約済み
5-4	RESERVED	R/W	0h	予約済み
3-2	CANABCLKSEL	R/W	0h	CANA ビット・クロック・ソース選択ビット: 00 = PERx.SYSCLK (リセット時のデフォルト) 01 = 外部発振器 (XTAL) 10 = AUXCLKIN (GPIO から) 11 = 予約済み クロック消失検出回路は、これらのビットに影響を及ぼしません。 リセット・タイプ: XRSn
1-0	RESERVED	R/W	0h	予約済み

1.15.3.4 CLKSRCCTL3 レジスタ (オフセット = Ch) [リセット = 0h]

図 1-27 に CLKSRCCTL3 を示し、表 1-33 にその説明を示します。

概略表に戻ります。

クロック・ソース制御レジスタ -3

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-27. CLKSRCCTL3 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み												XCLKOUTSEL			
R-0-0h												R/W-0h			

表 1-33. CLKSRCCTL3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3-0	XCLKOUTSEL	R/W	0h	XCLKOUT ソース選択ビット: このビットは、XCLKOUT のソースを選択します。 0x0 = PLLSYSCLK (リセット時のデフォルト) 0x1 = SYSPLLCLK 0x2 = SYSCLK 0x5 = INTOSC1 0x6 = INTOSC2 0x7 = XTAL OSC o/p クロック 0xC = PLLRAWCLK その他=予約済み リセット・タイプ: SYSRSn

1.15.3.5 SYSPLLCTL1 レジスタ (オフセット = Eh) [リセット = 0h]

図 1-28 に SYSPLLCTL1 を示し、表 1-34 にその説明を示します。

概略表に戻ります。

SYSPLL 制御レジスタ -1

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSPCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-28. SYSPLLCTL1 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	PLLCLKEN	PLLEN
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-34. SYSPLLCTL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-6	RESERVED	R-0	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	PLLCLKEN	R/W	0h	SYSPLL がバイパスされるか、あるいは PLLSYSCLK パスに含まれるか: このビットは、PLLSYSCLK が生成されたときに SYSPLL がバイパスされるかどうかを決定します 1 = PLLSYSCLK が SYSPLL クロック出力から供給されます。ユーザーは、このクロックをシステムに対して有効にする前に、PLL がロックされていることを確認する必要があります。 0 = SYSPLL はバイパスされます。システムへのクロックは、OSCCLK から直接供給されます。 リセット・タイプ XRSn
0	PLLEN	R/W	0h	SYSPLL のイネーブルまたはディセーブル: このビットは、SYSPLL をイネーブルにするかどうかを決定します 1 = SYSPLL がイネーブル 0 = SYSPLL が電源オフになります。システムへのクロックは、OSCCLK から直接供給されます。 リセット・タイプ XRSn

1.15.3.6 SYSPLLMULT レジスタ (オフセット = 14h) [リセット = 0h]

図 1-29 に SYSPLLMULT を示し、表 1-35 にその説明を示します。

概略表に戻ります。

SYSPLL 通倍器レジスタ

注: PLL を正しく動作させるには、FMULT および IMULT の各フィールドに同時に書き込む必要があります。このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-29. SYSPLLMULT レジスタ

31	30	29	28	27	26	25	24
RESERVED				REFDIV			
R-0-0h				R/W-0h			
23	22	21	20	19	18	17	16
RESERVED				ODIV			
R-0-0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R-0-0h		R/W-0h		R-0-0h		R/W-0h	
7	6	5	4	3	2	1	0
IMULT							
R/W-0h							

表 1-35. SYSPLLMULT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-29	RESERVED	R-0	0h	予約済み
28-24	REFDIV	R/W	0h	SYSPLL リファレンス・クロック分周器 PLL リファレンス分周器 = REFDIV + 1 リセット・タイプ: XRSn
23-21	RESERVED	R-0	0h	予約済み
20-16	ODIV	R/W	0h	SYSPLL 出力クロック分周器 PLL 出力分周器 = ODIV + 1 ODIV は、PLL 出力がデューティ・サイクル要件を満たすようにするため、1 以上に設定する必要があります。 リセット・タイプ: XRSn
15-14	RESERVED	R-0	0h	予約済み
13-12	RESERVED	R/W	0h	予約済み
11-10	RESERVED	R-0	0h	予約済み
9-8	RESERVED	R/W	0h	予約済み
7-0	IMULT	R/W	0h	SYSPLL 整数通倍器: 0000000 の場合、Fout = Fref (PLLBYPASS) 整数通倍器 = 1 0000001 整数通倍器 = 1 0000010 整数通倍器 = 2 0000011 整数通倍器 = 3 1111111 整数乗算器 = 127 APLL 通倍器の値として 0~3 は無効であることに注意してください。これらは内部的に 4 として処理されます。 リセットタイプ: XRSn

1.15.3.7 SYSPLLSTS レジスタ (オフセット = 16h) [リセット = 30h]

図 1-30 に SYSPLLSTS を示し、表 1-36 にその説明を示します。

概略表に戻ります。

SYSPLL ステータス・レジスタ

図 1-30. SYSPLLSTS レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	REF_LOSTS	RESERVED	SLIPS_NOTSU PPORTED	LOCKS
R-0-0h		R-1h	R-1h	W1C-0h	R-0h	R-0h	R-0h

表 1-36. SYSPLLSTS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-6	RESERVED	R-0	0h	予約済み
5	RESERVED	R	1h	予約済み
4	RESERVED	R	1h	予約済み
3	REF_LOSTS	W1C	0h	SYSPLL 「リファレンス消失」ステータス・ビット:このビットは、SYSPLL がロック範囲外であるかどうかを示します 0 = 「リファレンス消失」イベントは発生していません。 1 = 「リファレンス消失」イベントが発生しました。 リセット・タイプ: XRSn
2	RESERVED	R	0h	予約済み
1	SLIPS_NOTSUPPORTED	R	0h	RESERVED: このビットは予約されており、読み取った値は無視してください。SYSPLL スリップ・ステータスを評価するには、DCC を使用することをお勧めします。 DCC を使用して SYSPLL スリップ・ステータスを確認するには、C2000Ware の最新のサンプル・ソフトウェア内にある InitSysPll() または SysCtl_setClock() 関数を参照してください。 リセットタイプ: XRSn
0	LOCKS	R	0h	SYSPLL ロック・ステータス・ビット:このビットは、SYSPLL がロックされているかどうかを示します。 0 = SYSPLL がまだロックされていない 1 = SYSPLL がロックされている リセット・タイプ: XRSn

1.15.3.8 SYSCLKDIVSEL レジスタ (オフセット = 22h) [リセット = 0h]

図 1-31 に SYSCLKDIVSEL を示し、表 1-37 にその説明を示します。

概略表に戻ります。

システム・クロック分周器選択レジスタ

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-31. SYSCLKDIVSEL レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							PLLSYSCLKDI V_LSB
R-0-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED			PLLSYSCLKDIV				
R-0-0h			R/W-0h				

表 1-37. SYSCLKDIVSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-9	RESERVED	R-0	0h	予約済み
8	PLLSYSCLKDIV_LSB	R/W	0h	このビットは分周器の LSB であり、これを設定すると、分周値は {PLLSYSCLKDIV, PLLSYSCLKDIV_LSB} となって、奇数分周が可能になります。たとえば、PLLSYSCLKDIV = 0x1、PLLSYSCLKDIV_LSB = 0 の場合、2 の分周器が使用されます。あるいは、PLLSYSCLKDIV_LSB = 1 の場合、分周器の値は 3 です。 リセット・タイプ: XRSn
7-6	RESERVED	R-0	0h	予約済み
5-0	PLLSYSCLKDIV	R/W	0h	PLLSYSCLK 分周選択: このビットは、PLLSYSCLK の分周器設定を選択します。 000000 = /1 000001 = /2 000010 = /4 (デフォルト) 000011 = /6 000100 = /8 111111 = /126 リセット・タイプ: XRSn

1.15.3.9 XCLKOUTDIVSEL レジスタ (オフセット = 28h) [リセット = 3h]

図 1-32 に XCLKOUTDIVSEL を示し、表 1-38 にその説明を示します。

概略表に戻ります。

XCLKOUT 分周器選択レジスタ

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-32. XCLKOUTDIVSEL レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						XCLKOUTDIV	
R-0-0h						R/W-3h	

表 1-38. XCLKOUTDIVSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1-0	XCLKOUTDIV	R/W	3h	XCLKOUT 分周選択:このビットは、XCLKOUT の分周器設定を選択します。 00 = /1 01 = /2 10 = /4 11 = /8 (リセット時のデフォルト) リセット・タイプ:SYSRSn

1.15.3.10 LOSPCP レジスタ (オフセット = 2Ch) [リセット = 2h]

図 1-33 に LOSPCP を示し、表 1-39 にその説明を示します。

概略表に戻ります。

低速クロック・ソース・プリスケアラ

図 1-33. LOSPCP レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													LSPCLKDIV		
R-0-0h													R/W-2h		

表 1-39. LOSPCP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-3	RESERVED	R-0	0h	予約済み
2-0	LSPCLKDIV	R/W	2h	これらのビットは、低速パリアフェラル・クロック (LSPCLK) レートを構成します 000、LSPCLK = /1 001、LSPCLK = /2 010、LSPCLK = /4 (リセット時のデフォルト) 011、LSPCLK = /6 100、LSPCLK = /8 101、LSPCLK = /10 110、LSPCLK = /12 111、LSPCLK = /14 注: [1] このクロックは SCI および SPI モジュールのストローブとして使用されます。 リセット・タイプ:SYSRSn

1.15.3.11 MCDCCR レジスタ (オフセット = 2Eh) [リセット = 6000h]

図 1-34 に MCDCCR を示し、表 1-40 にその説明を示します。

概略表に戻ります。

クロック消失検出制御レジスタ

図 1-34. MCDCCR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h	R/W-1h	R/W-1h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	SYSREF_LAST_MCD_EN	SYSREF_LOST_SCLR	SYSREF_LOSTS	OSCOFF	MCLKOFF	MCLKCLR	MCLKSTS
R-0h	R/W-0h	R-0/W1S-0h	R-0h	R/W-0h	R/W-0h	R-0/W1S-0h	R-0h

表 1-40. MCDCCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-15	RESERVED	R-0	0h	予約済み
14	RESERVED	R/W	1h	予約済み
13	RESERVED	R/W	1h	予約済み
12	RESERVED	R/W	0h	予約済み
11	RESERVED	R-0/W1S	0h	予約済み
10	RESERVED	R	0h	予約済み
9	RESERVED	R/W	0h	予約済み
8	RESERVED	R-0/W1S	0h	予約済み
7	RESERVED	R	0h	予約済み
6	SYSREF_LOST_MCD_EN	R/W	0h	MCD の要因として「PLL リファレンス・クロック消失」を追加するかどうかを制御 0 = 「PLL リファレンス・クロック消失」は MCD に影響を与えません。 1 = 「PLL リファレンス・クロック消失」で MCD がアサートされます。 リセットタイプ: XRSn
5	SYSREF_LOSTSCLR	R-0/W1S	0h	MCD トリガのルートである PLLSTS から REF_LOST_STS をクリアします。 0 = REF_LOST_STS の現在の状態に影響なし 1 = REF_LOST_STS ビットを「0」にクリア。REF_LOST_STS へのクリア・パルス後に、このビットはクリアされます。 読み取り時は常に「0」となります。 リセット・タイプ: XRSn
4	SYSREF_LOSTS	R	0h	SYSPLL 「リファレンス消失」ステータス・ビット: このビットは、SYSPLL がロック範囲外であるかどうかを示します 0 = 「リファレンス消失」イベントは発生していません。 1 = 「リファレンス消失」イベントが発生しました。 リセット・タイプ: XRSn

表 1-40. MCD CR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	OSCOFF	R/W	0h	発振器のクロックを MCD ビットから切断: 0 = OSCCLK は MCD モジュールの OSCCLK カウンタに接続 1 = OSCCLK は MCD モジュールの OSCCLK カウンタから切断 リセット・タイプ: XRSn
2	MCLKOFF	R/W	0h	クロック消失検出オフ・ビット: 0 = クロック消失検出回路イネーブル 1 = クロック消失検出回路ディセーブル リセット・タイプ: XRSn
1	MCLKCLR	R-0/W1S	0h	クロック消失クリア・ビット: このビットに「1」を書き込むと、MCLKSTS ビットがクリアされ、クロック消失検出回路がリセットされます。 リセット・タイプ: XRSn
0	MCLKSTS	R	0h	クロック消失ステータス・ビット: 0 = OSCCLK は OK 1 = OSCCLK 消失を検出、CLOCKFAILn を生成 リセット・タイプ: XRSn

1.15.3.12 X1CNT レジスタ (オフセット = 30h) [リセット = 0h]

図 1-35 に X1CNT を示し、表 1-41 にその説明を示します。

概略表に戻ります。

X1 クロックの 10 ビット・カウンタ

図 1-35. X1CNT レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															CLR
R-0-0h															R-0/ W1C-0 h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					X1CNT										
R-0-0h					R-0h										

表 1-41. X1CNT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-17	RESERVED	R-0	0h	予約済み
16	CLR	R-0/W1C	0h	x1 カウンタ・クリア: このビット・フィールドに「1」を書き込むと、X1CNT がクリアされ、0x0 から再びカウントされます (X1 クロックが動作している場合)。 このビット・フィールドへの「0」の書き込みは無視されます リセット・タイプ: XRSn
15-11	RESERVED	R-0	0h	予約済み
10-0	X1CNT	R	0h	X1 カウンタ: - このカウンタは、X1 クロックの立ち上がりエッジごとにインクリメントします。 - 0x7ff の値に達すると停止します - INTOSC2 から X1 に切り替える前に、アプリケーションはこのカウンタをチェックし、飽和していることを確認する必要があります。これにより、X1/X2 に接続された水晶振動子が発振していることを保証できます。 リセット・タイプ: XRSn

1.15.3.13 XTALCR レジスタ (オフセット = 32h) [リセット = 5h]

図 1-36 に XTALCR を示し、表 1-42 にその説明を示します。

概略表に戻ります。

XTAL 制御レジスタ

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-36. XTALCR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED		SE	OSCOFF
R-0-0h				R/W-1h		R/W-0h	R/W-1h

表 1-42. XTALCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-3	RESERVED	R-0	0h	予約済み
2	RESERVED	R/W	1h	予約済み
1	SE	R/W	0h	XTAL 発振器の電源がオン (すなわち OSCOFF = 0) のとき、XTAL 発振器をシングルエンドまたは水晶振動子モードに設定します 0 XTAL 発振器は水晶振動子モード 1 XTAL 発振器はシングルエンド・モード (X1 経由) リセット・タイプ: XRSn
0	OSCOFF	R/W	1h	このビットが「1」のとき、XTAL 発振器マクロの電源をオフにします。したがって、X2 を XTAL 発振器で駆動することはできません。水晶振動子が X1/X2 に接続されている場合、ユーザーは、まずこのビットをクリアして、発振器の電源がオンになるのを待ち (X1CNT を使用)、その後でクロック・ソースを X1/X2 に切り替える必要があります。 リセット・タイプ: XRSn

1.15.3.14 XTALCR2 レジスタ (オフセット = 3Ah) [リセット = 3h]

図 1-37 に XTALCR2 を示し、表 1-43 にその説明を示します。

概略表に戻ります。

パッド初期化用の XTAL 制御レジスタ

図 1-37. XTALCR2 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													FEN	XOF	XIF
R-0-0h													R/W-0h R/W-1h R/W-1h		

表 1-43. XTALCR2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R/W	0h	予約済み
15-3	RESERVED	R-0	0h	予約済み
2	FEN	R/W	0h	XTAL 発振器パッドの初期化を構成します。 0: XOSC パッドは、GPIO 接続を介して駆動されません。 1: XOSC パッドは、XIF および XOF の値に従って、接続された GPIO を介して駆動されます。 このレジスタは、XOSC がオフ (SE なし、XTAL モードなし) のときのみ有効です。 XOSC オフ状態 (XOSCOFF = 1 および SE = 0) で、このレジスタが設定されている場合、これらの制御を変更したときに、このビットはリセットされ再設定されます。 リセットタイプ: XRSn
1	XOF	R/W	1h	起動前に XOSC の XO /X2 パッドを初期化する極性の選択 XOSC 開始 (XSCOFF = 1) の前に、この値がパッドに設定されず FEN = 0 または XOSC が XTAL モードもしくは SE モードの場合、この値はパッドに適用されません。 リセット・タイプ: XRSn
0	XIF	R/W	1h	起動前に XOSC の XI /X1 パッドを初期化する極性の選択 XOSC 開始 (XSCOFF = 1) の前に、この値がパッドに設定されず FEN = 0 または XOSC が XTAL モードもしくは SE モードの場合、この値はパッドに適用されません。 リセット・タイプ: XRSn

1.15.3.15 CLKFAILCFG レジスタ (オフセット = 3Ch) [リセット = 0h]

図 1-38 に CLKFAILCFG を示し、表 1-44 にその説明を示します。

概略表に戻ります。

クロック故障要因構成

図 1-38. CLKFAILCFG レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	DCC0_ERROR_EN
R-0-0h						R/W-0h	R/W-0h

表 1-44. CLKFAILCFG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	DCC0_ERROR_EN	R/W	0h	このフィールドにより、DCC0 エラー発生時にクロック故障 NMI がアサートされます。 0: DCC0 エラーはクロック故障 NMI に影響しません 1: DCC0 エラーが発生すると、クロック故障 NMI のアサートおよびエラー・ピンのアサートがトリガされます。 リセット・タイプ: XRSn

1.15.4 CPU_SYS_REGS Registers

表 1-45 に、CPU_SYS_REGS レジスタのメモリマップされたレジスタを示します。表 1-45 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-45. CPU_SYS_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	CPUSYSLOCK1	CPUSYS レジスタのロック・ビット	EALLOW	表示
2h	CPUSYSLOCK2	CPUSYS レジスタのロック・ビット	EALLOW	表示
Ah	PIEVERRADDR	PIE バクタ・フェッチエラー・アドレス・レジスタ	EALLOW	表示
22h	PCLKCR0	ペリフェラル・クロック・ゲーティング・レジスタ	EALLOW	表示
26h	PCLKCR2	ペリフェラル・クロック・ゲーティング・レジスタ - ETPWM	EALLOW	表示
28h	PCLKCR3	ペリフェラル・クロック・ゲーティング・レジスタ - ECAP	EALLOW	表示
2Ah	PCLKCR4	ペリフェラル・クロック・ゲーティング・レジスタ - EQEP	EALLOW	表示
30h	PCLKCR7	ペリフェラル・クロック・ゲーティング・レジスタ - SCI	EALLOW	表示
32h	PCLKCR8	ペリフェラル・クロック・ゲーティング・レジスタ - SPI	EALLOW	表示
34h	PCLKCR9	ペリフェラル・クロック・ゲーティング・レジスタ - I2C	EALLOW	表示
36h	PCLKCR10	ペリフェラル・クロック・ゲーティング・レジスタ - CAN	EALLOW	表示
3Ch	PCLKCR13	ペリフェラル・クロック・ゲーティング・レジスタ - ADC	EALLOW	表示
3Eh	PCLKCR14	ペリフェラル・クロック・ゲーティング・レジスタ - CMPSS	EALLOW	表示
4Ch	PCLKCR21	ペリフェラル・クロック・ゲーティング・レジスタ - DCC	EALLOW	表示
58h	PCLKCR27	ペリフェラル・クロック・ゲーティング・レジスタ - EPG	EALLOW	表示
70h	SIMRESET	シミュレーテッド・リセット・レジスタ		表示
76h	LPMCR	LPM 制御レジスタ	EALLOW	表示
78h	GPIOLPMSEL0	GPIO LPM ウェークアップ選択レジスタ	EALLOW	表示
7Ah	GPIOLPMSEL1	GPIO LPM ウェークアップ選択レジスタ	EALLOW	表示
7Ch	TMR2CLKCTL	タイマ 2 クロック測定機能制御レジスタ	EALLOW	表示
7Eh	RESCCLR	リセット要因クリア・レジスタ		表示
80h	RESC	リセット要因レジスタ		表示
A0h	USER_REG1_SYSRSn	SYSRSn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
A2h	USER_REG2_SYSRSn	SYSRSn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
A4h	USER_REG1_XRSn	XRSn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
A6h	USER_REG2_XRSn	XRSn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
A8h	USER_REG1_PORESETn	PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
AAh	USER_REG2_PORESETn	PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
ACh	USER_REG3_PORESETn	PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示
A Eh	USER_REG4_PORESETn	PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-46 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-46. CPU_SYS_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	書き込み 1 でセット
WSonce	W Sonce	書き込み 1 回のみセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.4.1 CPUSYSLOCK1 レジスタ (オフセット = 0h) [リセット = 0h]

図 1-39 に CPUSYSLOCK1 を示し、表 1-47 にその説明を示します。

概略表に戻ります。

CPUSYS レジスタのロック・ビット

注:

[1] このレジスタのいずれかのビットは、一度セットした後は、CPU1.SYSRSn によってのみクリアできます。このレジスタのいずれかのビットに 0 を書き込んでも、何も効果はありません。

[2] ロック・メカニズムは、書き込みにも適用されます。ロック保護機能を持つレジスタへの読み取りは常に許可されます

図 1-39. CPUSYSLOCK1 レジスタ

31		30		29		28		27		26		25		24	
RESERVED	RESERVED	PCLKCR22	PCLKCR21	PCLKCR20	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h
23		22		21		20		19		18		17		16	
GPIOLPMSEL1	GPIOLPMSEL0	LPMCR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PCLKCR14	PCLKCR13	PCLKCR12	PCLKCR11	PCLKCR10	PCLKCR9
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h
15		14		13		12		11		10		9		8	
RESERVED	RESERVED	PCLKCR10	PCLKCR9	PCLKCR8	PCLKCR7	PCLKCR6	PCLKCR5	PCLKCR4	PCLKCR3	PCLKCR2	PCLKCR1	PCLKCR0	RESERVED	RESERVED	RESERVED
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h
7		6		5		4		3		2		1		0	
PCLKCR4	PCLKCR3	PCLKCR2	RESERVED	PCLKCR0	PIEVERRADDR	RESERVED									
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h

表 1-47. CPUSYSLOCK1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R/WOnce	0h	予約済み
30	RESERVED	R/WOnce	0h	予約済み
29	PCLKCR22	R/WOnce	0h	PCLKCR22 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
28	PCLKCR21	R/WOnce	0h	PCLKCR21 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
27	PCLKCR20	R/WOnce	0h	PCLKCR20 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
26	RESERVED	R/WOnce	0h	予約済み
25	RESERVED	R/WOnce	0h	予約済み
24	RESERVED	R/WOnce	0h	予約済み
23	GPIOLPMSEL1	R/WOnce	0h	GPIOLPMSEL1 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
22	GPIOLPMSEL0	R/WOnce	0h	GPIOLPMSEL0 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn

表 1-47. CPUSYSLOCK1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
21	LPMCR	R/WOnce	0h	LPMCR レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
20	RESERVED	R/WOnce	0h	予約済み
19	RESERVED	R/WOnce	0h	予約済み
18	RESERVED	R/WOnce	0h	予約済み
17	PCLKCR14	R/WOnce	0h	PCLKCR14 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
16	PCLKCR13	R/WOnce	0h	PCLKCR13 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
15	RESERVED	R/WOnce	0h	予約済み
14	RESERVED	R/WOnce	0h	予約済み
13	PCLKCR10	R/WOnce	0h	PCLKCR10 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
12	PCLKCR9	R/WOnce	0h	PCLKCR9 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
11	PCLKCR8	R/WOnce	0h	PCLKCR8 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
10	PCLKCR7	R/WOnce	0h	PCLKCR7 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
9	PCLKCR6	R/WOnce	0h	PCLKCR6 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
8	RESERVED	R/WOnce	0h	予約済み
7	PCLKCR4	R/WOnce	0h	PCLKCR4 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
6	PCLKCR3	R/WOnce	0h	PCLKCR3 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
5	PCLKCR2	R/WOnce	0h	PCLKCR2 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
4	RESERVED	R/WOnce	0h	予約済み

表 1-47. CPUSYSLOCK1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	PCLKCR0	R/WOnce	0h	PCLKCR0 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
2	PIEVERRADDR	R/WOnce	0h	PIEVERRADDR レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
1	RESERVED	R/WOnce	0h	予約済み
0	RESERVED	R/WOnce	0h	予約済み

1.15.4.2 CPUSYSLOCK2 レジスタ (オフセット = 2h) [リセット = 0h]

図 1-40 に CPUSYSLOCK2 を示し、表 1-48 にその説明を示します。

概略表に戻ります。

CPUSYS レジスタのロック・ビット

注:

[1] このレジスタのいずれかのビットは、一度セットした後は、CPU1.SYSRSn によってのみクリアできます。このレジスタのいずれかのビットに 0 を書き込んでも、何も効果はありません。

[2] ロック・メカニズムは、書き込みのみ適用されます。ロック保護機能を持つレジスタへの読み取りは常に許可されます

図 1-40. CPUSYSLOCK2 レジスタ

31	30	29	28	27	26	25	24
USER_REG4_PORESETn	USER_REG3_PORESETn	USER_REG2_PORESETn	USER_REG1_PORESETn	USER_REG2_XRSn	USER_REG1_XRSn	USER_REG2_SYRSn	USER_REG1_SYRSn
R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED						
R/WOnce-0h	R-0-0h						
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	PCLKCR27	RESERVED	RESERVED	RESERVED
R-0-0h			R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h

表 1-48. CPUSYSLOCK2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	USER_REG4_PORESETn	R/WOnce	0h	USER_REG4_PORESETn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
30	USER_REG3_PORESETn	R/WOnce	0h	USER_REG3_PORESETn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
29	USER_REG2_PORESETn	R/WOnce	0h	USER_REG2_PORESETn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
28	USER_REG1_PORESETn	R/WOnce	0h	USER_REG1_PORESETn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
27	USER_REG2_XRSn	R/WOnce	0h	USER_REG2_XRSn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn
26	USER_REG1_XRSn	R/WOnce	0h	USER_REG1_XRSn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYRSn

表 1-48. CPUSYSLOCK2 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
25	USER_REG2_SYSRSn	R/WOnce	0h	USER_REG2_SYSRSn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
24	USER_REG1_SYSRSn	R/WOnce	0h	USER_REG1_SYSRSn レジスタのロック・ビット 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
23	RESERVED	R/WOnce	0h	予約済み
22-5	RESERVED	R-0	0h	予約済み
4	RESERVED	R/WOnce	0h	予約済み
3	PCLKCR27	R/WOnce	0h	PCLKCR27 レジスタのロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 リセット・タイプ: SYSRSn
2	RESERVED	R/WOnce	0h	予約済み
1	RESERVED	R/WOnce	0h	予約済み
0	RESERVED	R/WOnce	0h	予約済み

1.15.4.3 PIEVERRADDR レジスタ (オフセット= Ah) [リセット = 003FFFFFFh]

図 1-41 に PIEVERRADDR を示し、表 1-49 にその説明を示します。

概略表に戻ります。

PIE ベクタ・フェッチエラー・アドレス・レジスタ

図 1-41. PIEVERRADDR レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											ADDR																				
R-0-0h											R/W-003FFFFFFh																				

表 1-49. PIEVERRADDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-22	RESERVED	R-0	0h	予約済み
21-0	ADDR	R/W	003FFFFFFh	このレジスタは、PIE ベクタ・フェッチエラー・ハンドラ・ルーチンのアドレスを定義します。このレジスタを初期化するのはユーザーの責任です。このレジスタが初期化されていない場合、アドレス 0x3fffbe のデフォルトのエラー・ハンドラが実行されます。このレジスタの詳細については、「ブート ROM」セクションを参照してください。 リセット・タイプ: XRSn

1.15.4.4 PCLKCR0 レジスタ (オフセット = 22h) [リセット = 38h]

図 1-42 に PCLKCR0 を示し、表 1-50 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ

図 1-42. PCLKCR0 レジスタ

31	30	29	28	27	26	25	24
RESERVED							RESERVED
R-0-0h							R/W-0h
23	22	21	20	19	18	17	16
RESERVED				RESERVED	TBCLKSYNC	RESERVED	HRCAL
R-0-0h				R/W-0h	R/W-0h	R-0-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED				
R-0-0h	R/W-0h	R/W-0h	R-0-0h				
7	6	5	4	3	2	1	0
RESERVED		CPUTIMER2	CPUTIMER1	CPUTIMER0	RESERVED	RESERVED	RESERVED
R-0-0h		R/W-1h	R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

表 1-50. PCLKCR0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-25	RESERVED	R-0	0h	予約済み
24	RESERVED	R/W	0h	予約済み
23-20	RESERVED	R-0	0h	予約済み
19	RESERVED	R/W	0h	予約済み
18	TBCLKSYNC	R/W	0h	EPWM タイム・ベース・クロック同期:このビットを設定すると、同じ CPU サブシステム (CPUSEL ビットを使用してパーティショニングされたもの) に属するすべての PWM モジュールの PWM タイム・ベースがカウントを開始します。 リセット・タイプ:SYSRSn
17	RESERVED	R-0	0h	予約済み
16	HRCAL	R/W	0h	HRCAL クロック・イネーブル・ビット:セットすると、HRCAL モジュールへのクロックがイネーブルになります。 1:HRCAL クロックがイネーブル 0:HRCAL クロックがディセーブル リセット・タイプ:SYSRSn
15	RESERVED	R-0	0h	予約済み
14	RESERVED	R/W	0h	予約済み
13	RESERVED	R/W	0h	予約済み
12-6	RESERVED	R-0	0h	予約済み
5	CPUTIMER2	R/W	1h	CPUTIMER2 クロック・イネーブル・ビット: 0:モジュール・クロックがゲートオフ 1:モジュール・クロックがオン リセット・タイプ:SYSRSn
4	CPUTIMER1	R/W	1h	CPUTIMER1 クロック・イネーブル・ビット: 0:モジュール・クロックがゲートオフ 1:モジュール・クロックがオン リセット・タイプ:SYSRSn

表 1-50. PCLKCR0 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	CPUTIMER0	R/W	1h	CPUTIMER0 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	RESERVED	R/W	0h	予約済み

1.15.4.5 PCLKCR2 レジスタ (オフセット = 26h) [リセット = 0h]

図 1-43 に PCLKCR2 を示し、表 1-51 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - ETPWM

図 1-43. PCLKCR2 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1
R/W-0h							

表 1-51. PCLKCR2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R-0	0h	予約済み
15	RESERVED	R/W	0h	予約済み
14	RESERVED	R/W	0h	予約済み
13	RESERVED	R/W	0h	予約済み
12	RESERVED	R/W	0h	予約済み
11	RESERVED	R/W	0h	予約済み
10	RESERVED	R/W	0h	予約済み
9	RESERVED	R/W	0h	予約済み
8	RESERVED	R/W	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	EPWM7	R/W	0h	EPWM7 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
5	EPWM6	R/W	0h	EPWM6 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
4	EPWM5	R/W	0h	EPWM5 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
3	EPWM4	R/W	0h	EPWM4 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

表 1-51. PCLKCR2 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
2	EPWM3	R/W	0h	EPWM3 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
1	EPWM2	R/W	0h	EPWM2 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
0	EPWM1	R/W	0h	EPWM1 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.6 PCLKCR3 レジスタ (オフセット = 28h) [リセット = 0h]

図 1-44 に PCLKCR3 を示し、表 1-52 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - ECAP

図 1-44. PCLKCR3 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ECAP2	ECAP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-52. PCLKCR3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R-0	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	ECAP2	R/W	0h	ECAP2 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
0	ECAP1	R/W	0h	ECAP1 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.7 PCLKCR4 レジスタ (オフセット = 2Ah) [リセット = 0h]

図 1-45 に PCLKCR4 を示し、表 1-53 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - EQEP

図 1-45. PCLKCR4 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	EQEP1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-53. PCLKCR4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	EQEP1	R/W	0h	EQEP1 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYRSn

1.15.4.8 PCLKCR7 レジスタ (オフセット = 30h) [リセット = 0h]

図 1-46 に PCLKCR7 を示し、表 1-54 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - SCI

図 1-46. PCLKCR7 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	SCI_C	SCI_B	SCI_A
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-54. PCLKCR7 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	SCI_C	R/W	0h	SCI_C クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
1	SCI_B	R/W	0h	SCI_B クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
0	SCI_A	R/W	0h	SCI_A クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.9 PCLKCR8 レジスタ (オフセット = 32h) [リセット = 0h]

図 1-47 に PCLKCR8 を示し、表 1-55 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - SPI

図 1-47. PCLKCR8 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						RESERVED	RESERVED
R-0-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	SPI_A
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-55. PCLKCR8 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-18	RESERVED	R-0	0h	予約済み
17	RESERVED	R/W	0h	予約済み
16	RESERVED	R/W	0h	予約済み
15-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	SPI_A	R/W	0h	SPI_A クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.10 PCLKCR9 レジスタ (オフセット = 34h) [リセット = 0h]

図 1-48 に PCLKCR9 を示し、表 1-56 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲート・レジスタ - I2C

図 1-48. PCLKCR9 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						I2C_B	I2C_A
R-0-0h						R/W-0h	R/W-0h

表 1-56. PCLKCR9 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1	I2C_B	R/W	0h	I2C_B クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
0	I2C_A	R/W	0h	I2C_A クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.11 PCLKCR10 レジスタ (オフセット = 36h) [リセット = 0h]

図 1-49 に PCLKCR10 を示し、表 1-57 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - CAN

図 1-49. PCLKCR10 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	CAN_A						
R/W-0h	R/W-0h						

表 1-57. PCLKCR10 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R-0	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	CAN_A	R/W	0h	CAN_A クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.12 PCLKCR13 レジスタ (オフセット = 3Ch) [リセット = 0h]

図 1-50 に PCLKCR13 を示し、表 1-58 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - ADC

図 1-50. PCLKCR13 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	ADC_C	RESERVED	ADC_A
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-58. PCLKCR13 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	ADC_C	R/W	0h	ADC_C クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
1	RESERVED	R/W	0h	予約済み
0	ADC_A	R/W	0h	ADC_A クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.13 PCLKCR14 レジスタ (オフセット = 3Eh) [リセット = 0h]

図 1-51 に PCLKCR14 を示し、表 1-59 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - CMPSS

図 1-51. PCLKCR14 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	CMPSS4	CMPSS3	CMPSS2	CMPSS1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-59. PCLKCR14 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R-0	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	CMPSS4	R/W	0h	CMPSS4 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
2	CMPSS3	R/W	0h	CMPSS3 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
1	CMPSS2	R/W	0h	CMPSS2 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn
0	CMPSS1	R/W	0h	CMPSS1 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.14 PCLKCR21 レジスタ (オフセット = 4Ch) [リセット = 0h]

図 1-52 に PCLKCR21 を示し、表 1-60 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - DCC

図 1-52. PCLKCR21 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	DCC0
R-0-0h						R/W-0h	R/W-0h

表 1-60. PCLKCR21 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	DCC0	R/W	0h	DCC クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.15 PCLKCR27 レジスタ (オフセット = 58h) [リセット = 0h]

図 1-53 に PCLKCR27 を示し、表 1-61 にその説明を示します。

概略表に戻ります。

ペリフェラル・クロック・ゲーティング・レジスタ - EPG

図 1-53. PCLKCR27 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							EPG1
R-0-0h							R/W-0h

表 1-61. PCLKCR27 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-1	RESERVED	R-0	0h	予約済み
0	EPG1	R/W	0h	EPG1 クロック・イネーブル・ビット: 0: モジュール・クロックがゲートオフ 1: モジュール・クロックがオン リセット・タイプ: SYSRSn

1.15.4.16 SIMRESET レジスタ (オフセット = 70h) [リセット = 0h]

図 1-54 に SIMRESET を示し、表 1-62 にその説明を示します。

概略表に戻ります。

シミュレーション・リセット・レジスタ

注: このレジスタは、CPU1 にのみ存在します

図 1-54. SIMRESET レジスタ

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						XRSn	CPU1RSn
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

表 1-62. SIMRESET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	KEY	R-0/W	0h	このレジスタへの書き込みは、このフィールドに 0xa5a5 の値が書き込まれた場合にのみ成功します 注: [1] この KEY があるので、32 ビット書き込みのみが成功します (KEY が一致している場合)。このレジスタの上半分または下半分への 16 ビット書き込みは無視されます リセット・タイプ: XRSn
15-2	RESERVED	R-0	0h	予約済み
1	XRSn	R-0/W1S	0h	このフィールドに 1 を書き込むと、XRSn と同様のリセットが生成されます。 0 を書き込んでも何も影響はありません。 注: このピンに書き込むと、512 INTOSC1 クロック・サイクルにわたって XRSn ピンが Low になります。 リセット・タイプ: XRSn
0	CPU1RSn	R-0/W1S	0h	このフィールドに 1 を書き込むと、CPU1 へのリセットが生成されます。 0 を書き込んでも何も影響はありません。 リセット・タイプ: XRSn

1.15.4.17 LPMCR レジスタ (オフセット = 76h) [リセット = FCh]

図 1-55 に LPMCR を示し、表 1-63 にその説明を示します。

概略表に戻ります。

LPM 制御レジスタ

図 1-55. LPMCR レジスタ

31	30	29	28	27	26	25	24
RESERVED				RESERVED			
R/W1S-0h				R-0-0h			
23	22	21	20	19	18	17	16
RESERVED						RESERVED	
R-0-0h						R/W-0h	
15	14	13	12	11	10	9	8
WDINTE				RESERVED			
R/W-0h				R-0-0h			
7	6	5	4	3	2	1	0
QUALSTDBY						LPM	
R/W-3Fh						R/W-0h	

表 1-63. LPMCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R/W1S	0h	予約済み
30-18	RESERVED	R-0	0h	予約済み
17-16	RESERVED	R/W	0h	予約済み
15	WDINTE	R/W	0h	このビットが 1 に設定されると、ウォッチドッグ割り込み信号がイネーブルになり、デバイスが STANDBY モードからウェイクアップします。 注: [1] この信号を使用するには、ユーザーは SCSR レジスタの WDEINTn ビットを使用して WDINTn 信号もイネーブルにする必要があります。この信号は、デバイスを HALT モードからウェイクアップしません。ウォッチドッグ・モジュールへのクロックがオフになっているからです。 リセット・タイプ:SYSRSn
14-8	RESERVED	R-0	0h	予約済み
7-2	QUALSTDBY	R/W	3Fh	STANDBY モードからウェイクアップするときに、選択した入力を確認するための OSCCLK クロック・サイクル数を選択します。 000000 = 2 OSCCLK 000001 = 3 OSCCLK 111111 = 65 OSCCLK 注: 適切なウェイクアップを確保するためには、LPMCR.QUALSTDBY レジスタの値を INTOSC1/PLLSYSCLK の比率より大きい値に設定する必要があります。 リセット・タイプ:SYSRSn
1-0	LPM	R/W	0h	これらのビットは、デバイスの低消費電力モードを設定します。CPU が IDLE 命令を実行したときに有効になります (IDLE 命令がパイプラインの EXE フェーズから外に出たとき) 00: IDLE モード 01: STANDBY モード 1x: HALT モード リセット・タイプ:SYSRSn

1.15.4.18 GPIOLPMSEL0 レジスタ (オフセット = 78h) [リセット = 0h]

図 1-56 に GPIOLPMSEL0 を示し、表 1-64 にその説明を示します。

概略表に戻ります。

GPIO LPM ウェークアップ選択レジスタ

選択したピンを LPM 回路に接続します。選択したピンのウェークアップ機能については、TRM の LPM セクションを参照してください。

図 1-56. GPIOLPMSEL0 レジスタ

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

表 1-64. GPIOLPMSEL0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	GPIO31	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
30	GPIO30	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
29	GPIO29	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
28	GPIO28	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
27	GPIO27	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
26	GPIO26	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
25	GPIO25	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
24	GPIO24	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
23	GPIO23	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn

表 1-64. GPIO_LPMSEL0 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
22	GPIO22	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
21	GPIO21	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
20	GPIO20	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
19	GPIO19	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
18	GPIO18	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
17	GPIO17	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
16	GPIO16	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
15	GPIO15	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
14	GPIO14	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
13	GPIO13	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
12	GPIO12	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
11	GPIO11	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
10	GPIO10	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
9	GPIO9	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
8	GPIO8	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
7	GPIO7	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
6	GPIO6	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn

表 1-64. GPIOLPMSEL0 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	GPIO5	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
4	GPIO4	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
3	GPIO3	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
2	GPIO2	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
1	GPIO1	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
0	GPIO0	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn

1.15.4.19 GPIO_LPMSEL1 レジスタ (オフセット = 7Ah) [リセット = 0h]

図 1-57 に GPIO_LPMSEL1 を示し、表 1-65 にその説明を示します。

概略表に戻ります。

GPIO LPM ウェークアップ選択レジスタ

選択したピンを LPM 回路に接続します。選択したピンのウェークアップ機能については、TRM の LPM セクションを参照してください。

図 1-57. GPIO_LPMSEL1 レジスタ

31		30		29		28		27		26		25		24	
GPIO63		GPIO62		GPIO61		GPIO60		GPIO59		GPIO58		GPIO57		GPIO56	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23		22		21		20		19		18		17		16	
GPIO55		GPIO54		GPIO53		GPIO52		GPIO51		GPIO50		GPIO49		GPIO48	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15		14		13		12		11		10		9		8	
GPIO47		GPIO46		GPIO45		GPIO44		GPIO43		GPIO42		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7		6		5		4		3		2		1		0	
GPIO39		GPIO38		GPIO37		GPIO36		GPIO35		GPIO34		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-65. GPIO_LPMSEL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	GPIO63	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
30	GPIO62	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
29	GPIO61	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
28	GPIO60	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
27	GPIO59	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
26	GPIO58	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
25	GPIO57	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
24	GPIO56	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn
23	GPIO55	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ: SYSRSn

表 1-65. GPIOLPMSEL1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
22	GPIO54	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
21	GPIO53	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
20	GPIO52	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
19	GPIO51	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
18	GPIO50	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
17	GPIO49	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
16	GPIO48	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
15	GPIO47	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
14	GPIO46	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
13	GPIO45	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
12	GPIO44	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
11	GPIO43	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
10	GPIO42	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
9	GPIO41	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
8	GPIO40	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
7	GPIO39	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
6	GPIO38	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn

表 1-65. GPIOLPMSEL1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	GPIO37	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
4	GPIO36	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
3	GPIO35	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
2	GPIO34	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
1	GPIO33	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn
0	GPIO32	R/W	0h	0 ビンは LPM 回路から切断 1 ビンは LPM 回路に接続 リセット・タイプ:SYSRSn

1.15.4.20 TMR2CLKCTL レジスタ (オフセット = 7Ch) [リセット = 0h]

図 1-58 に TMR2CLKCTL を示し、表 1-66 に、その説明を示します。

概略表に戻ります。

タイマ 2 クロック測定機能制御レジスタ

このメモリ・マップされたレジスタは、レジスタへの次の書き込みまでに 45 SYSCLK サイクルの遅延が必要です。そうしないと、2 回目の書き込みが失われる可能性があります。この遅延は、45 個の NOP 命令を追加することで実現できます。

図 1-58. TMR2CLKCTL レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		TMR2CLKPRESCALE			TMR2CLKSRCSEL		
R-0-0h		R/W-0h			R/W-0h		

表 1-66. TMR2CLKCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-6	RESERVED	R-0	0h	予約済み
5-3	TMR2CLKPRESCALE	R/W	0h	CPU タイマ 2 クロック・プリスケール値:これらのビットは、CPU タイマ 2 に対して選択したクロック・ソースのプリスケール値を選択します。 0,0,0,/1 (リセット時のデフォルト) 0,0,1,/2, 0,1,0,/4 0,1,1,/8 1,0,0,/16 1,0,1,予備 (デフォルトは /16) 1,1,0,予備 (デフォルトは /16) 1,1,1,予備 (デフォルトは /16) 注: [1] CPU タイマ 2 クロック同期ロジックは、SYSCLK 以外のいずれかのクロック・ソースに対して構成されているときに入力クロックのエッジを検出し、CPU タイマ 2 に対して適切なクロック・パルスを生成します。SYSCLK が入力クロック・ソースとほぼ同じかそれ以下の場合、ユーザは、SYSCLK がプリスケール値の少なくとも 2 倍の速度になるように、プリスケール値を設定する必要があります。 リセット・タイプ:SYRSn
2-0	TMR2CLKSRCSEL	R/W	0h	CPU タイマ 2 クロック・ソース選択ビット:このビットは、CPU タイマ 2 のソースを選択します。 000 = SYSCLK を選択 (リセット時のデフォルト、プリスケールはパイパス) 001 = INTOSC1 010 = INTOSC2 011 = XTAL 100 = PUMPOSC (ラッパリーなしから) 101 = FOSCCLK (予約済み) 110 = AUXPLLCLK (予約済み) 111 = 予約済み リセット・タイプ:SYRSn

1.15.4.21 RESCCLR レジスタ (オフセット = 7Eh) [リセット = 0h]

図 1-59 に RESCCLR を示し、表 1-67 にその説明を示します。

概略表に戻ります。

リセット要因クリア・レジスタ

図 1-59. RESCCLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				SIMRESET_XR Sn	SIMRESET_CP U1RSn	RESERVED	SCCRESETn
R-0-0h				W1C-0h	W1C-0h	R-0-0h	W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	NMIWDRSn	WDRSn	XRSn	POR
R-0-0h	W1S-0h	W1S-0h	R-0-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h

表 1-67. RESCCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-12	RESERVED	R-0	0h	予約済み
11	SIMRESET_XRSn	W1C	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ: SYSRSn
10	SIMRESET_CPU1RSn	W1C	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ: SYSRSn
9	RESERVED	R-0	0h	予約済み
8	SCCRESETn	W1S	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ: SYSRSn
7	RESERVED	R-0	0h	予約済み
6	RESERVED	W1S	0h	予約済み
5	RESERVED	W1S	0h	予約済み
4	RESERVED	R-0	0h	予約済み

表 1-67. RESCCLR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	NMIWDRSn	W1S	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ:SYSRSn
2	WDRSn	W1S	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ:SYSRSn
1	XRSn	W1S	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ:SYSRSn
0	POR	W1S	0h	RESC の対応するステータス・ビットのクリア・ビット。RESCCLR の読み出しは常に 0 です。 このビットに 1 を書き込むと、RESC のステータス・ビットが 0 にクリアされます 0 を書き込んでも何も影響はありません。 リセット・タイプ:SYSRSn

1.15.4.22 RESC レジスタ (オフセット = 80h) [リセット = X]

図 1-60 に RESC を示し、表 1-68 にその説明を示します。

概略表に戻ります。

リセット要因レジスタ

図 1-60. RESC レジスタ

31	30	29	28	27	26	25	24
DCON	XRSn_pin_status	RESERVED					
R-0h	R-X	R-0-0h					
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				SIMRESET_XRSn	SIMRESET_CPU1RSn	RESERVED	SCCRESETn
R-0-0h				R-0h	R-0h	R-0-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	NMIWDRSn	WDRSn	XRSn	POR
R-0-0h	R-0h	R-0h	R-0-0h	R-0h	R-0h	R-1h	R-1h

表 1-68. RESC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	DCON	R	0h	このビットを読み取ると、C28x CPU へのデバッグ接続のステータスが表示されます。 0: デバッグが C28x CPU に接続されていません 1: デバッグが C28x CPU に接続されています 注: [1] このビットは、C28x CPU の DCON 出力信号に接続されます。 リセット・タイプ: 該当なし
30	XRSn_pin_status	R	X	このビットを読み出すと、XRSn ピンの現在のステータスが表示されます。 リセット値はピンのステータスを反映します。 リセット・タイプ: 該当なし
29-16	RESERVED	R-0	0h	予約済み
15-12	RESERVED	R-0	0h	予約済み
11	SIMRESET_XRSn	R	0h	このビットが設定されている場合、デバイスが SIMRESET_XRSn によってリセットされたことを示します。 リセット・タイプ: PORESETn
10	SIMRESET_CPU1RSn	R	0h	このビットが設定されている場合、デバイスが SIMRESET_CPU1RSn によってリセットされたことを示します。 リセット・タイプ: PORESETn
9	RESERVED	R-0	0h	予約済み
8	SCCRESETn	R	0h	このビットが設定されている場合、デバイスが SCCRESETn (DCSM によって作動) によってリセットされたことを示します。 リセット・タイプ: PORESETn
7	RESERVED	R-0	0h	予約済み
6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	RESERVED	R-0	0h	予約済み

表 1-68. RESC レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	NMIWDRSn	R	0h	このビットが設定されている場合、デバイスが NMIWDRSn によってリセットされたことを示します。 注:リセット後に NMI の正確な要因を知るためには、ソフトウェアで NMISHDFLG レジスタを読み取る必要があります リセット・タイプ:PORESETn
2	WDRSn	R	0h	このビットが設定されている場合、デバイスが WDRSn によってリセットされたことを示します。 注: [1] WD モジュール内のビットも同じ情報を提供します。このビットは、一貫性を維持するために用意されています。このレジスタは、C28x コアのリセット要因をソフトウェアが把握するためのワンストップ・ショップです。 リセット・タイプ:PORESETn
1	XRSn	R	1h	このビットが設定されている場合、デバイスが XRSn によってリセットされたことを示します。 リセット・タイプ:PORESETn
0	POR	R	1h	このビットが設定されている場合、デバイスが PORn によってリセットされたことを示します。 リセット・タイプ:PORESETn

1.15.4.23 USER_REG1_SYSRSn レジスタ (オフセット = A0h) [リセット = X]

図 1-61 に USER_REG1_SYSRSn を示し、表 1-69 に、その説明を示します。

概略表に戻ります。

SYSRSn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-61. USER_REG1_SYSRSn レジスタ

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

表 1-69. USER_REG1_SYSRSn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	SYSRSn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ:SYSRSn

1.15.4.24 USER_REG2_SYSRSn レジスタ (オフセット = A2h) [リセット = 0h]

図 1-62 に USER_REG2_SYSRSn を示し、表 1-70 にその説明を示します。

概略表に戻ります。

SYSRSn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-62. USER_REG2_SYSRSn レジスタ

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

表 1-70. USER_REG2_SYSRSn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	SYSRSn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ:SYSRSn

1.15.4.25 USER_REG1_XRSn レジスタ (オフセット = A4h) [リセット = 0h]

図 1-63 に USER_REG1_XRSn を示し、表 1-71 にその説明を示します。

概略表に戻ります。

XRSn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-63. USER_REG1_XRSn レジスタ

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

表 1-71. USER_REG1_XRSn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	XRSn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ: XRSn

1.15.4.26 USER_REG2_XRSn レジスタ (オフセット = A6h) [リセット = 0h]

図 1-64 に USER_REG2_XRSn を示し、表 1-72 にその説明を示します。

概略表に戻ります。

XRSn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-64. USER_REG2_XRSn レジスタ

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

表 1-72. USER_REG2_XRSn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	XRSn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ: XRSn

1.15.4.27 USER_REG1_PORESETn レジスタ (オフセット = A8h) [リセット = 0h]

図 1-65 に USER_REG1_PORESETn を示し、表 1-73 にその説明を示します。

概略表に戻ります。

PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-65. USER_REG1_PORESETn レジスタ

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

表 1-73. USER_REG1_PORESETn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	PORESETn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ: PORESETn

1.15.4.28 USER_REG2_PORESETn レジスタ (オフセット = AAh) [リセット = 0h]

図 1-66 に USER_REG2_PORESETn を示し、表 1-74 にその説明を示します。

概略表に戻ります。

PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-66. USER_REG2_PORESETn レジスタ

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

表 1-74. USER_REG2_PORESETn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	PORESETn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ: PORESETn

1.15.4.29 USER_REG3_PORESETn レジスタ (オフセット = ACh) [リセット = 0h]

図 1-67 に USER_REG3_PORESETn を示し、表 1-75 にその説明を示します。

概略表に戻ります。

PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-67. USER_REG3_PORESETn レジスタ

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

表 1-75. USER_REG3_PORESETn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	PORESETn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ:PORESETn

1.15.4.30 USER_REG4_PORESETn レジスタ (オフセット = AEh) [リセット = 0h]

図 1-68 に USER_REG4_PORESETn を示し、表 1-76 にその説明を示します。

概略表に戻ります。

PORESETn によってリセットされる、ソフトウェアで構成可能なレジスタ

図 1-68. USER_REG4_PORESETn レジスタ

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

表 1-76. USER_REG4_PORESETn レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	BITS	R/W	0h	PORESETn によってリセットされる R/W ビットであり、アプリケーション・ソフトウェアで使用されます リセット・タイプ:PORESETn

1.15.5 CPUTIMER_REGS レジスタ

表 1-77 に、CPUTIMER_REGS のメモリマップされたレジスタを示します。表 1-77 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-77. CPUTIMER_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	TIM	CPU タイマ、カウンタ・レジスタ		表示
2h	PRD	CPU タイマ、周期レジスタ		表示
4h	TCR	CPU タイマ、制御レジスタ		表示
6h	TPR	CPU タイマ、プリスケール・レジスタ		表示
7h	TPRH	CPU タイマ、プリスケール・レジスタ (上位ワード)		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-78 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-78. CPUTIMER_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.5.1 TIM レジスタ (オフセット = 0h) [リセット = FFFFh]

図 1-69 に TIM を示し、表 1-79 にその説明を示します。

概略表に戻ります。

CPU タイマ、カウンタ・レジスタ

図 1-69. TIM レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSW																LSW															
R/W-0h																R/W-FFFFh															

表 1-79. TIM レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	MSW	R/W	0h	CPU タイマ・カウンタ・レジスタ TIMH レジスタは、タイマの 32 ビット現在値の上位 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 はタイマのプリスケール分周値です。TIMH:TIM がデクリメントして 0 になると、PRDH:PRD レジスタに保存されている周期値が 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。 リセット・タイプ:SYSRSn
15-0	LSW	R/W	FFFFh	CPU タイマ・カウンタ・レジスタ TIM レジスタは、タイマの 32 ビット現在値の下位 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 はタイマのプリスケール分周値です。TIMH:TIM がデクリメントして 0 になると、PRDH:PRD レジスタに保存されている周期値が 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。 リセット・タイプ:SYSRSn

1.15.5.2 PRD レジスタ (オフセット = 2h) [リセット = FFFFh]

図 1-70 に PRD を示し、表 1-80 にその説明を示します。

概略表に戻ります。

CPU タイマ、周期レジスタ

図 1-70. PRD レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSW																LSW															
R/W-0h																R/W-FFFFh															

表 1-80. PRD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	MSW	R/W	0h	CPU タイマ周期レジスタ PRDH レジスタは、32 ビットの周期の上位 16 ビットを保持します TIMH:TIM がデクリメントして 0 になると、次のタイマ入力クロック・サイクル (プリスケアラ出力) の開始時に、PRDH:PRD レジスタに保存されている周期値が TIMH:TIM レジスタに再ロードされます。タイマ制御レジスタ (TCR) でタイマ再ロード・ビット (TRB) をセットしたときにも、PRDH:PRD の内容が TIMH:TIM にロードされます。 リセット・タイプ:SYSRSn
15-0	LSW	R/W	FFFFh	CPU タイマ周期レジスタ PRD レジスタは、32 ビットの周期の下位 16 ビットを保持します。 TIMH:TIM がデクリメントして 0 になると、次のタイマ入力クロック・サイクル (プリスケアラ出力) の開始時に、PRDH:PRD レジスタに保存されている周期値が TIMH:TIM レジスタに再ロードされます。タイマ制御レジスタ (TCR) でタイマ再ロード・ビット (TRB) をセットしたときにも、PRDH:PRD の内容が TIMH:TIM にロードされます。 リセット・タイプ:SYSRSn

1.15.5.3 TCR レジスタ (オフセット = 4h) [リセット = 1h]

図 1-71 に TCR を示し、表 1-81 にその説明を示します。

概略表に戻ります。

CPU タイマ、制御レジスタ

図 1-71. TCR レジスタ

15		14		13		12		11		10		9		8	
TIF		TIE		RESERVED				FREE		SOFT		RESERVED			
R/W1C-0h		R/W-0h		R-0h				R/W-0h		R/W-0h		R-0h			
7		6		5		4		3		2		1		0	
RESERVED				TRB		TSS		RESERVED							
R-0h				R/W-0h		R/W-0h		R-1h							

表 1-81. TCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	TIF	R/W1C	0h	CPU タイマ・オーバーフロー・フラグ。 TIF は、TIF が前回クリアされた後に、タイマ・オーバーフローが発生したかどうかを示します。TIF は自動的にクリアされず、次のタイマ割り込みをイネーブルにするためにクリアする必要はありません。 リセット・タイプ: SYSRSn 0h (R/W) = CPU タイマはゼロにデクリメントされていません。 0 を書き込んでも無視されます。 1h (R/W) = CPU タイマがゼロにデクリメントしたとき、このフラグがセットされます。 このビットに 1 を書き込むと、フラグがクリアされます。
14	TIE	R/W	0h	CPU タイマ割り込みイネーブル。 リセット・タイプ: SYSRSn 0h (R/W) = CPU タイマ割り込みがディセーブル。 1h (R/W) = CPU タイマ割り込みがイネーブル。タイマが 0 にデクリメントし、TIE がセットされている場合、タイマはその割り込み要求をアサートします。
13-12	RESERVED	R	0h	予約済み
11	FREE	R/W	0h	FREE ビットが 1 にセットされている場合、ソフトウェア・ブレイクポイントが発生したとき、タイマは引き続き実行されます。FREE が 0 の場合、SOFT ビットによってエミュレーション動作が制御されます。 リセット・タイプ: SYSRSn 0h (R/W) = TIMH:TIM の次のデクリメント後に停止 (ハード・ストップ) (SOFT ビットがエミュレーション動作を制御) 1h (R/W) = フリーラン (SOFT ビットは don't care (影響なし)、カウンタはフリーランニング)
10	SOFT	R/W	0h	FREE ビットが 1 にセットされている場合、ソフトウェア・ブレイクポイントが発生したとき、タイマは引き続き実行されます (すなわち、フリーラン)。この場合、SOFT は don't care (影響なし) です。しかし、FREE が 0 の場合には、SOFT が有効になります。 リセット・タイプ: SYSRSn 0h (R/W) = TIMH:TIM の次のデクリメント後に停止 (ハード・ストップ) (FREE=0 の場合のみ。FREE=1 の場合には、このビットは don't care) 1h (R/W) = TIMH:TIM が 0 にデクリメントした後に停止 (ソフト・ストップ) ソフト・ストップ・モードでは、シャットダウンする前にタイマが割り込みを生成します (0 への到達は、割り込みを生成する条件なので)。(FREE=0 の場合のみ。FREE=1 の場合には、このビットは don't care)
9-6	RESERVED	R	0h	予約済み

表 1-81. TCR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	TRB	R/W	0h	タイマの再ロード ・リセット・タイプ:SYSRSn 0h (R/W) = TRB ビットは常にゼロとして読み取られます。0 を書き込んでも無視されます。 1h (R/W) = TRB に 1 を書き込むと、TIMH:TIM に PRDH:PRD の値がロードされます。 プリスケアラ・カウンタ (PSCH:PSC) には、タイマ分周レジスタ (TDDR:H:TDDR) の値がロードされます。
4	TSS	R/W	0h	CPU タイマ停止ステータス・ビット。 TSS は、CPU タイマを停止または開始する 1 ビット・フラグです。 リセット・タイプ:SYSRSn 0h (R/W) = 読み出した値が 0 である場合、CPU タイマが動作していることを示します。 CPU タイマを開始または再起動するには、TSS を 0 に設定します。リセット時、TSS は 0 にクリアされ、CPU タイマは直ちに開始します。 1h (R/W) = 読み出した値が 1 である場合、CPU タイマが停止していることを示します。 CPU タイマを停止するには、TSS を 1 に設定します。
3-0	RESERVED	R	1h	予約済み

1.15.5.4 TPR レジスタ (オフセット = 6h) [リセット = 0h]

図 1-72 に TPR を示し、表 1-82 にその説明を示します。

概略表に戻ります。

CPU タイマ、プリスケール・レジスタ

図 1-72. TPR レジスタ

15	14	13	12	11	10	9	8
PSC							
R-0h							
7	6	5	4	3	2	1	0
TDDR							
R/W-0h							

表 1-82. TPR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	PSC	R	0h	CPU タイマ・プリスケール・カウンタ。 これらのビットには、タイマの現在のプリスケール・カウントが保持されま ず。PSCH:PSC 値が 0 より大きい場合、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 に達 すると、タイマ・クロック (タイマ・プリスケララの出力) の 1 サイクル後に、 PSCH:PSC に TDDRH:TDDR の内容がロードされ、タイマ・カウンタ・レ ジスタ (TIMH:TIM) が 1 だけデクリメントされます。また、ソフトウェアによ ってタイマ再ロード・ビット (TRB) がセットされた場合にも、PSCH:PSC が再 ロードされます。PSCH:PSC は、レジスタを読み取ることによって確認でき ますが、直接設定することはできません。このレジスタは、タイマ分周レジス タ (TDDRH:TDDR) から値を取得する必要があります。 リセット時、PSCH:PSC は 0 に設定されます。 リセット・タイプ:SYSRSn
7-0	TDDR	R/W	0h	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、 TDDRH:TDDR のビットは 0 にクリアされます。タイマ全体のカウントを整 数の係数だけ増やすには、その係数から 1 を減算した値を TDDRH:TDDR ビットに書き込みます。 プリスケララ・カウンタ (PSCH:PSC) の値が 0 になると、タイマ・クロック・ソ ース・サイクルの 1 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 だけ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設定されたときにも、 TDDRH:TDDR の値が PSCH:PSC に再ロードされます。 リセット・タイプ:SYSRSn

1.15.5.5 TPRH レジスタ (オフセット = 7h) [リセット = 0h]

図 1-73 に TPRH を示し、表 1-83 にその説明を示します。

概略表に戻ります。

CPU タイマ、プリスケール・レジスタ (上位ワード)

図 1-73. TPRH レジスタ

15	14	13	12	11	10	9	8
PSCH							
R-0h							
7	6	5	4	3	2	1	0
TDDRH							
R/W-0h							

表 1-83. TPRH レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	PSCH	R	0h	TIMERxTPR の説明を参照してください。 リセット・タイプ:SYSRSn
7-0	TDDRH	R/W	0h	TIMERxTPR の説明を参照してください。 リセット・タイプ:SYSRSn

1.15.6 DEV_CFG_REGS レジスタ

DEV_CFG_REGS レジスタのメモリマップされたレジスタを、表 1-84 に示します。表 1-84 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-84. DEV_CFG_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
8h	PARTIDL	デバイス部品識別番号の下位 32 ビット		表示
Ah	PARTIDH	デバイス部品識別番号の上位 32 ビット		表示
Ch	REVID	デバイス・リビジョン番号		表示
74h	TRIMERRSTS	TRIM エラー・ステータス・レジスタ		表示
86h	SOFTPRES2	EPWM ソフトウェア・リセット・レジスタ	EALLOW	表示
88h	SOFTPRES3	ECAP ソフトウェア・リセット・レジスタ	EALLOW	表示
8Ah	SOFTPRES4	EQEP ソフトウェア・リセット・レジスタ	EALLOW	表示
90h	SOFTPRES7	SCI ソフトウェア・リセット・レジスタ	EALLOW	表示
92h	SOFTPRES8	SPI ソフトウェア・リセット・レジスタ	EALLOW	表示
94h	SOFTPRES9	I2C ソフトウェア・リセット・レジスタ	EALLOW	表示
96h	SOFTPRES10	CAN ソフトウェア・リセット・レジスタ	EALLOW	表示
9Ch	SOFTPRES13	ADC ソフトウェア・リセット・レジスタ	EALLOW	表示
9Eh	SOFTPRES14	CMPSS ソフトウェア・リセット・レジスタ	EALLOW	表示
ACh	SOFTPRES21	DCC ソフトウェア・リセット・レジスタ	EALLOW	表示
B8h	SOFTPRES27	EPG ソフトウェア・リセット・レジスタ	EALLOW	表示
BAh	SOFTPRES28	フラッシュ・ソフトウェア・リセット・レジスタ	EALLOW	表示
130h	TAP_STATUS	JTAG ステート・マシンおよびデバッグ接続のステータス		表示
19Bh	ECAPTYPE	デバイスの ECAP タイプの構成	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-85 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-85. DEV_CFG_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み取りタイプ		
R	R	読み取り
R-0	R-0	読み取り 0 を返す
書き込みタイプ		
W	W	書き込み
W1C	W 1C	書き込み 1 でクリア
W1S	W 1S	書き込み 1 で設定
WOnce	W Once	書き込み 1 回書き込み
WSonce	W Sonce	書き込み 1 回設定
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		

**表 1-85. DEV_CFG_REGS のアクセス・タイプ・コード
(continued)**

アクセス・タイプ	コード	説明
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ・グループの一部であるレジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ配列の値を参照します。

1.15.6.1 PARTIDL レジスタ (オフセット = 8h) [リセット = X]

図 1-74 に、PARTIDL を示し、表 1-86 に、その説明を示します。

概略表に戻ります。

デバイス部品識別番号の下位 32 ビット

図 1-74. PARTIDL レジスタ

31	30	29	28	27	26	25	24
RESERVED				RESERVED			
R-X				R-0h			
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED	INSTASPIN		RESERVED	RESERVED	PIN_COUNT		
R-0h	R-X		R-0h	R-X	R-X		
7	6	5	4	3	2	1	0
QUAL		RESERVED	RESERVED		RESERVED		
R-X		R-0h	R-0h		R-0h		

表 1-86. PARTIDL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-28	RESERVED	R	X	予約済み
27-24	RESERVED	R	0h	予約済み
23-16	RESERVED	R	X	予約済み
15	RESERVED	R	0h	予約済み
14-13	INSTASPIN	R	X	1 = InstaSPIN-FOC 2 = NONE 3 = NONE リセット・タイプ: PORESETn
12	RESERVED	R	0h	予約済み
11	RESERVED	R	X	予約済み
10-8	PIN_COUNT	R	X	0 = 56 ピン QFN 1 = 64 ピン (QFP - Q100) 2 = 64 ピン (QFP) 3 = 80 ピン (QFP) 4 = 48 ピン (QFP) 5 = 32 ピン (QFN) 6 = 予約済み 7 = 48 ピン (QFN) 8 = 64 ピン (QFP) - VREGENZ ボンドアウト リセット・タイプ: PORESETn
7-6	QUAL	R	X	0 = エンジニアリング・サンプル (TMX) 1 = パイロット生産 (TMP) 2 = 完全認定済み (TMS) リセット・タイプ: PORESETn
5	RESERVED	R	0h	予約済み
4-3	RESERVED	R	0h	予約済み
2-0	RESERVED	R	0h	予約済み

1.15.6.2 PARTIDH レジスタ (オフセット = Ah) [リセット = X]

図 1-75 に、PARTIDH を示し、表 1-87 に、その説明を示します。

概略表に戻ります。

デバイス部品識別番号の上位 32 ビット

図 1-75. PARTIDH レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEVICE_CLASS_ID								PARTNO							
R-6h								R-X							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAMILY								RESERVED				RESERVED			
R-5h								R-0h				R-0h			

表 1-87. PARTIDH レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-24	DEVICE_CLASS_ID	R	6h	デバイス・クラス ID リセット・タイプ: PORESETn
23-16	PARTNO	R	X	部品番号指定子 0xFF=F2800137x 0xFE=F2800136x 0xFD=F2800135x (非 VREGENZ) 0xFC=F2800134x 0xFB=F2800133x 0xFA=F2800132x 0xF9=F2800135VPM リセット・タイプ: PORESETn
15-8	FAMILY	R	5h	デバイス・ファミリ このフィールドは、Delfino、Piccolo (Harmony クラス)、Delfino シングル・コア、Concerto などの C2000 デバイス・ファミリのいずれかにデバイスを分類します。Topogrande は Piccolo (Harmony クラス) に属し、値は 0x5 です。 リセット・タイプ: PORESETn
7-4	RESERVED	R	0h	予約済み
3-0	RESERVED	R	0h	予約済み

1.15.6.3 REVID レジスタ (オフセット = Ch) [リセット = 0h]

図 1-76 に、REVID を示し、表 1-88 に、その説明を示します。

概略表に戻ります。

デバイス・リビジョン番号

図 1-76. REVID レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REVID															
R-0-0h																R/WOnce-0h															

表 1-88. REVID レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R-0	0h	予約済み
15-0	REVID	R/WOnce	0h	デバイス・リビジョン ID. ブート ROM によりフラッシュ・トリム・セクタから読み込まれます。リセット値はダイ固有です。 リセット・タイプ: XRSn

1.15.6.4 TRIMERRSTS レジスタ (オフセット = 74h) [リセット = 0h]

図 1-77 に、TRIMERRSTS を示し、表 1-89 に、その説明を示します。

概略表に戻ります。

TRIM エラー・ステータス・レジスタ

図 1-77. TRIMERRSTS レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LERR															
R-0-0h																R/WSonce-0h															

表 1-89. TRIMERRSTS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R-0	0h	予約済み
15-0	LERR	R/WSonce	0h	TRIM 情報読み込みエラー・ステータス。これには、SRAM 修復時のエラーも含まれます。 0x1: 修正可能なシングル・ビット・エラー 0x2: 修正不可能なダブル・ビット・エラー 0x20: トリム・オーバー・タイムアウト・エラー その他: ゼロでない値は読み込み時のエラーを示す 注: [1] このビットはソフトウェアによって更新されます。ブート ROM 関連の要件が完了すると、詳細が入力されます。以下を示すビットがあります。 (i) トリム読み込み時のダブル・ビット・エラー (ii) トリム読み込み時のシングル・ビット・エラー (iii) SRAM 修復読み込み時のダブル・ビット・エラー (iv) SRAM 修復読み込み時のシングル・ビット・エラー (v) SRAM 修復の読み込みエラー (チェーンが破損) (vi) 待機時間の経過後も PWRUPSTS.TRIMOVER 信号がアサートされない リセット・タイプ: XRSn

1.15.6.5 SOFTPRES2 レジスタ (オフセット = 86h) [リセット = 0h]

図 1-78 に、SOFTPRES2 を示し、表 1-90 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-78. SOFTPRES2 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1
R/W-0h							

表 1-90. SOFTPRES2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R-0	0h	予約済み
15	RESERVED	R/W	0h	予約済み
14	RESERVED	R/W	0h	予約済み
13	RESERVED	R/W	0h	予約済み
12	RESERVED	R/W	0h	予約済み
11	RESERVED	R/W	0h	予約済み
10	RESERVED	R/W	0h	予約済み
9	RESERVED	R/W	0h	予約済み
8	RESERVED	R/W	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	EPWM7	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
5	EPWM6	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
4	EPWM5	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
3	EPWM4	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
2	EPWM3	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

表 1-90. SOFTPRES2 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
1	EPWM2	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
0	EPWM1	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.6 SOFTPRES3 レジスタ (オフセット = 88h) [リセット = 0h]

図 1-79 に、SOFTPRES3 を示し、表 1-91 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-79. SOFTPRES3 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ECAP2	ECAP1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-91. SOFTPRES3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R-0	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	ECAP2	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn
0	ECAP1	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn

1.15.6.7 SOFTPRES4 レジスタ (オフセット = 8Ah) [リセット = 0h]

図 1-80 に、SOFTPRES4 を示し、表 1-92 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-80. SOFTPRES4 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	EQEP1
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-92. SOFTPRES4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	EQEP1	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.8 SOFTPRES7 レジスタ (オフセット = 90h) [リセット = 0h]

図 1-81 に、SOFTPRES7 を示し、表 1-93 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-81. SOFTPRES7 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	SCI_C	SCI_B	SCI_A
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-93. SOFTPRES7 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	SCI_C	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
1	SCI_B	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
0	SCI_A	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.9 SOFTPRES8 レジスタ (オフセット = 92h) [リセット = 0h]

図 1-82 に、SOFTPRES8 を示し、表 1-94 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-82. SOFTPRES8 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED						RESERVED	RESERVED
R-0-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	SPI_A
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-94. SOFTPRES8 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-18	RESERVED	R-0	0h	予約済み
17	RESERVED	R/W	0h	予約済み
16	RESERVED	R/W	0h	予約済み
15-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	SPI_A	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.10 SOFTPRES9 レジスタ (オフセット = 94h) [リセット = 0h]

図 1-83 に、SOFTPRES9 を示し、表 1-95 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-83. SOFTPRES9 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						I2C_B	I2C_A
R-0-0h						R/W-0h	R/W-0h

表 1-95. SOFTPRES9 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1	I2C_B	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn
0	I2C_A	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn

1.15.6.11 SOFTPRES10 レジスタ (オフセット = 96h) [リセット = X]

図 1-84 に、SOFTPRES10 を示し、表 1-96 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-84. SOFTPRES10 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	CAN_A						
R-X	R-X	R-X	R-X	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-96. SOFTPRES10 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R-0	0h	予約済み
7	RESERVED	R	X	予約済み
6	RESERVED	R	X	予約済み
5	RESERVED	R	X	予約済み
4	RESERVED	R	X	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	CAN_A	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.12 SOFTPRES13 レジスタ (オフセット = 9Ch) [リセット = 0h]

図 1-85 に、SOFTPRES13 を示し、表 1-97 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-85. SOFTPRES13 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	ADC_C	RESERVED	ADC_A
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-97. SOFTPRES13 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-4	RESERVED	R-0	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	ADC_C	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn
1	RESERVED	R/W	0h	予約済み
0	ADC_A	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.13 SOFTPRES14 レジスタ (オフセット = 9Eh) [リセット = 0h]

図 1-86 に、SOFTPRES14 を示し、表 1-98 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-86. SOFTPRES14 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	CMPSS4	CMPSS3	CMPSS2	CMPSS1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-98. SOFTPRES14 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R-0	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	CMPSS4	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn
2	CMPSS3	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn
1	CMPSS2	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn
0	CMPSS1	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYRSn

1.15.6.14 SOFTPRES21 レジスタ (オフセット = ACh) [リセット = 0h]

図 1-87 に、SOFTPRES21 を示し、表 1-99 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-87. SOFTPRES21 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						RESERVED	DCC0
R-0-0h						R/W-0h	R/W-0h

表 1-99. SOFTPRES21 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	DCC0	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.15 SOFTPRES27 レジスタ (オフセット = B8h) [リセット = 0h]

図 1-88 に、SOFTPRES27 を示し、表 1-100 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-88. SOFTPRES27 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							EPG1
R-0-0h							R/W-0h

表 1-100. SOFTPRES27 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-1	RESERVED	R-0	0h	予約済み
0	EPG1	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 リセット・タイプ: SYSRSn

1.15.6.16 SOFTPRES28 レジスタ (オフセット = BAh) [リセット = 0h]

図 1-89 に、SOFTPRES28 を示し、表 1-101 に、その説明を示します。

概略表に戻ります。

ペリフェラル・ソフトウェア・リセット・レジスタ

このレジスタのビットを設定すると、該当するペリフェラルのリセットが実行されます。すべてのデータが失われ、ペリフェラルのレジスタがリセット状態に戻されます。これらのビットを設定した後は、手動でクリアする必要があります。

図 1-89. SOFTPRES28 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED							FLASHA
R-0-0h							R/W-0h

表 1-101. SOFTPRES28 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-1	RESERVED	R-0	0h	予約済み
0	FLASHA	R/W	0h	1: モジュールのリセットを実行 0: 通常のデバイス・リセット構造に従ってモジュールのリセットを実行 注: フラッシュへのリセットがアサートされるたびに、内部で 15us までストレッチされます。 リセット・タイプ: SYSRSn

1.15.6.17 TAP_STATUS レジスタ (オフセット = 130h) [リセット = 0h]

図 1-90 に、TAP_STATUS を示し、表 1-102 に、その説明を示します。

概略表に戻ります。

JTAG ステート・マシンおよびデバッガ接続のステータス

図 1-90. TAP_STATUS レジスタ

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

表 1-102. TAP_STATUS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	DCON	R	0h	IcePick からの DebugConnect 表示。 リセット・タイプ: PORESETn
30-16	RESERVED	R-0	0h	予約済み
15-0	TAP_STATE	R	0h	TAP 状態ベクトル。ビットが表す次のとおり、対応する POTAP* 出力を接続します。 0: TLR 1: IDLE 2: SELECTDR 3: CAPDR 4: SHIFTD 5: EXIT1DR 6: PAUSED 7: EXIT2DR 8: UPDTR 9: SELECTIR 10: CAPIR 11: SHIFDIR 12: EXIT1IR 13: PAUSEIR 14: EXIT2IR 15: UPDIR リセット・タイプ: PORESETn

1.15.6.18 ECAPTYPE レジスタ (オフセット = 19Bh) [リセット = 0h]

図 1-91 に、ECAPTYPE を示し、表 1-103 に、その説明を示します。

概略表に戻ります。

構成に基づいて、ECAP タイプに関連する機能を有効 / 無効にします。

図 1-91. ECAPTYPE レジスタ

15	14	13	12	11	10	9	8
LOCK	RESERVED						
R/WOnce-0h				R-0-0h			
7	6	5	4	3	2	1	0
RESERVED						TYPE	
R-0-0h						R/W-0h	

表 1-103. ECAPTYPE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	LOCK	R/WOnce	0h	1: このレジスタへの書き込みが許可されません。 0: このレジスタへの書き込みが許可されます。 リセット・タイプ: SYSRSn
14-2	RESERVED	R-0	0h	予約済み
1-0	タイプ	R/W	0h	"00,10,11": 1. ECAP レジスタに対する EALLOW 保護なし。 "01": 1. ECAP レジスタに対する EALLOW 保護あり。 リセット・タイプ: SYSRSn

1.15.7 MEM_CFG_REGS レジスタ

表 1-104 に、MEM_CFG_REGS レジスタのメモリマップされたレジスタを示します。表 1-104 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-104. MEM_CFG_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	DxLOCK	専用 RAM 構成ロック・レジスタ	EALLOW	表示
2h	DxCOMMIT	専用 RAM 構成ロック・コミット・レジスタ	EALLOW	表示
8h	DxACCPROT0	専用 RAM 構成レジスタ	EALLOW	表示
Ah	DxACCPROT1	専用 RAM 構成レジスタ	EALLOW	表示
10h	DxTEST	専用 RAM テスト・レジスタ		表示
12h	DxINIT	専用 RAM 初期化レジスタ	EALLOW	表示
14h	DxINITDONE	専用 RAM 初期化完了ステータス・レジスタ		表示
16h	DxRAMTEST_LOCK	専用 RAM テスト・レジスタのロック・レジスタ		表示
20h	LSxLOCK	ローカル共有 RAM 構成ロック・レジスタ	EALLOW	表示
22h	LSxCOMMIT	ローカル共有 RAM 構成ロック・コミット・レジスタ	EALLOW	表示
28h	LSxACCPROT0	ローカル共有 RAM 構成レジスタ 0	EALLOW	表示
30h	LSxTEST	ローカル共有 RAM テスト・レジスタ		表示
32h	LSxINIT	ローカル共有 RAM 初期化レジスタ	EALLOW	表示
34h	LSxINITDONE	ローカル共有 RAM 初期化完了ステータス・レジスタ		表示
36h	LSxRAMTEST_LOCK	ローカル共有 RAM テスト・レジスタのロック・レジスタ		表示
A0h	ROM_LOCK	ROM 構成ロック・レジスタ		表示
A2h	ROM_TEST	ROM テスト・レジスタ		表示
A4h	ROM_FORCE_ERROR	ROM 強制エラー・レジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-105 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-105. MEM_CFG_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R -0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W 1S	書き込み 1 でセット
WSonce	W Sonce	書き込み 1 回のみセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。

**表 1-105. MEM_CFG_REGS のアクセス・タイプ・コード
(continued)**

アクセス・タイプ	コード	説明
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.7.1 DxLOCK レジスタ (オフセット = 0h) [リセット = 0h]

図 1-92 に DxLOCK を示し、表 1-106 にその説明を示します。

概略表に戻ります。

専用 RAM 構成ロック・レジスタ

図 1-92. DxLOCK レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			LOCK_PIEVECT	RESERVED	RESERVED	LOCK_M1	LOCK_M0
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-106. DxLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-5	RESERVED	R	0h	予約済み
4	LOCK_PIEVECT	R/W	0h	PIEVECT RAM のアクセス保護、マスタ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みをロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みが可能です。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みはブロックされます。 リセット・タイプ: SYSRSn
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	LOCK_M1	R/W	0h	M1 RAM のアクセス保護、マスタ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みをロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みが可能です。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みはブロックされます。 リセット・タイプ: SYSRSn
0	LOCK_M0	R/W	0h	M0 RAM のアクセス保護、マスタ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みをロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みが可能です。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みはブロックされます。 リセット・タイプ: SYSRSn

1.15.7.2 DxCOMMIT レジスタ (オフセット = 2h) [リセット = 0h]

図 1-93 に DxCOMMIT を示し、表 1-107 に、その説明を示します。

概略表に戻ります。

専用 RAM 構成ロック・コミット・レジスタ

図 1-93. DxCOMMIT レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			COMMIT_PIEVECT	RESERVED	RESERVED	COMMIT_M1	COMMIT_M0
R-0h			R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h

表 1-107. DxCOMMIT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-5	RESERVED	R	0h	予約済み
4	COMMIT_PIEVECT	R/WOnce	0h	PIEVECT RAM のアクセス保護、マスタ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みを永続的にロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みは、DxLOCK レジスタの LOCK フィールドの値に応じて許可されます。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みは永続的にブロックされます。 リセット・タイプ: SYSRSn
3	RESERVED	R/WOnce	0h	予約済み
2	RESERVED	R/WOnce	0h	予約済み
1	COMMIT_M1	R/WOnce	0h	M1 RAM のアクセス保護、マスタ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みを永続的にロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みは、DxLOCK レジスタの LOCK フィールドの値に応じて許可されます。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みは永続的にブロックされます。 リセット・タイプ: SYSRSn
0	COMMIT_M0	R/WOnce	0h	M0 RAM のアクセス保護、マスタ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みを永続的にロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みは、DxLOCK レジスタの LOCK フィールドの値に応じて許可されます。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みは永続的にブロックされます。 リセット・タイプ: SYSRSn

1.15.7.3 DxACCPROT0 レジスタ (オフセット = 8h) [リセット = 0h]

図 1-94 に DxACCPROT0 を示し、表 1-108 に、その説明を示します。

概略表に戻ります。

専用 RAM 構成レジスタ

図 1-94. DxACCPROT0 レジスタ

31	30	29	28	27	26	25	24
RESERVED						RESERVED	RESERVED
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED						RESERVED	RESERVED
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						CPUWRPROT_M1	FETCHPROT_M1
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED						CPUWRPROT_M0	FETCHPROT_M0
R-0h						R/W-0h	R/W-0h

表 1-108. DxACCPROT0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-26	RESERVED	R	0h	予約済み
25	RESERVED	R/W	0h	予約済み
24	RESERVED	R/W	0h	予約済み
23-18	RESERVED	R	0h	予約済み
17	RESERVED	R/W	0h	予約済み
16	RESERVED	R/W	0h	予約済み
15-10	RESERVED	R	0h	予約済み
9	CPUWRPROT_M1	R/W	0h	M1 RAM に対する CPU 書き込み保護: 0: CPU の書き込みが可能です。 1: CPU の書き込みがブロックされます。 リセット・タイプ: SYSRSn
8	FETCHPROT_M1	R/W	0h	M1 RAM に対するフェッチ保護: 0: CPU のフェッチが可能です。 1: CPU のフェッチがブロックされます。 リセット・タイプ: SYSRSn
7-2	RESERVED	R	0h	予約済み
1	CPUWRPROT_M0	R/W	0h	M0 RAM に対する CPU 書き込み保護: 0: CPU の書き込みが可能です。 1: CPU の書き込みがブロックされます。 リセット・タイプ: SYSRSn
0	FETCHPROT_M0	R/W	0h	M0 RAM に対するフェッチ保護: 0: CPU のフェッチが可能です。 1: CPU のフェッチがブロックされます。 リセット・タイプ: SYSRSn

1.15.7.4 DxACCPROT1 レジスタ (オフセット = Ah) [リセット = 0h]

図 1-95 に DxACCPROT1 を示し、表 1-109 にその説明を示します。

概略表に戻ります。

専用 RAM 構成レジスタ

図 1-95. DxACCPROT1 レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						CPUWRPROT_ PIEVECT	RESERVED
R-0h						R/W-0h	R/W-0h

表 1-109. DxACCPROT1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R	0h	予約済み
1	CPUWRPROT_PIEVECT	R/W	0h	PIEVECT RAM に対する CPU 書き込み保護: 0: CPU の書き込みが可能です。 1: CPU の書き込みがブロックされます。 リセット・タイプ: SYSRSn
0	RESERVED	R/W	0h	予約済み

1.15.7.5 DxTEST レジスタ (オフセット = 10h) [リセット = 0h]

図 1-96 に DxTEST を示し、表 1-110 にその説明を示します。

概略表に戻ります。

専用 RAM テスト・レジスタ

図 1-96. DxTEST レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						TEST_PIEVECT	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED		TEST_M1		TEST_M0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-110. DxTEST レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-10	RESERVED	R	0h	予約済み
9-8	TEST_PIEVECT	R/W	0h	PIEVECT RAM のいくつかのモードを選択します。 00: 機能モード。 01: 書き込みはデータ・ビットに対してのみ許可されます。パリティ・ビットへの書き込み不可。 10: 書き込みはパリティ・ビットに対してのみ許可されます。データ・ビットへの書き込み不可。 11: 機能モード。 注: 0 以外の値を指定した場合、書き込み保護に優先して CPU による書き込みが可能になり、アクセス保護違反は発生しません。 リセット・タイプ: SYSRSn
7-6	RESERVED	R/W	0h	予約済み
5-4	RESERVED	R/W	0h	予約済み
3-2	TEST_M1	R/W	0h	M1 RAM のいくつかのモードを選択します。 00: 機能モード。 01: 書き込みはデータ・ビットに対してのみ許可されます。ECC ビットへの書き込み不可。 10: 書き込みは ECC ビットに対してのみ許可されます。データ・ビットへの書き込み不可。 11: 機能モードと同じですが、エラー発生時に割り込み/NMI は生成されません。 注: 0 以外の値を指定した場合、書き込み保護に優先して CPU による書き込みが可能になり、アクセス保護違反は発生しません。 リセット・タイプ: SYSRSn

表 1-110. DxTEST レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
1-0	TEST_M0	R/W	0h	M0 RAM のいくつかのモードを選択します。 00: 機能モード。 01: 書き込みはデータ・ビットに対してのみ許可されます。ECC ビットへの書き込み不可。 10: 書き込みは ECC ビットに対してのみ許可されます。データ・ビットへの書き込み不可。 11: 機能モードと同じですが、エラー発生時に割り込み/NMI は生成されません。 注: 0 以外の値を指定した場合、書き込み保護に優先して CPU による書き込みが可能になり、アクセス保護違反は発生しません。 リセット・タイプ: SYSRSn

1.15.7.6 DxINIT レジスタ (オフセット = 12h) [リセット = 0h]

図 1-97 に DxINIT を示し、表 1-111 にその説明を示します。

概略表に戻ります。

専用 RAM 初期化レジスタ

図 1-97. DxINIT レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			INIT_PIEVECT	RESERVED	RESERVED	INIT_M1	INIT_M0
R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-111. DxINIT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-5	RESERVED	R	0h	予約済み
4	INIT_PIEVECT	R-0/W1S	0h	PIEVECT RAM の RAM 初期化制御: 0:なし。 1:RAM 初期化を開始します。 リセット・タイプ:SYSRSn
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	INIT_M1	R-0/W1S	0h	M1 RAM の RAM 初期化制御: 0:なし。 1:RAM 初期化を開始します。 リセット・タイプ:SYSRSn
0	INIT_M0	R-0/W1S	0h	M0 RAM の RAM 初期化制御: 0:なし。 1:RAM 初期化を開始します。 リセット・タイプ:SYSRSn

1.15.7.7 DxINITDONE レジスタ (オフセット = 14h) [リセット = 0h]

図 1-98 に DxINITDONE を示し、表 1-112 にその説明を示します。

概略表に戻ります。

専用 RAM 初期化完了ステータス・レジスタ

図 1-98. DxINITDONE レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			INITDONE_PIE VECT	RESERVED	RESERVED	INITDONE_M1	INITDONE_M0
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

表 1-112. DxINITDONE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-5	RESERVED	R	0h	予約済み
4	INITDONE_PIEVECT	R	0h	PIEVECT RAM の RAM 初期化ステータス: 0: RAM の初期化が完了しました。 1: RAM の初期化が完了しました。 リセット・タイプ: SYRSn
3	RESERVED	R	0h	予約済み
2	RESERVED	R	0h	予約済み
1	INITDONE_M1	R	0h	M1 RAM の RAM 初期化ステータス: 0: RAM の初期化は行われていません。 1: RAM の初期化が完了しました。 リセット・タイプ: SYRSn
0	INITDONE_M0	R	0h	M0 RAM の RAM 初期化ステータス: 0: RAM の初期化は行われていません。 1: RAM の初期化が完了しました。 リセット・タイプ: SYRSn

1.15.7.8 DxRAMTEST_LOCK レジスタ (オフセット = 16h) [リセット = 0h]

図 1-99 に DxRAMTEST_LOCK を示し、表 1-113 にその説明を示します。

概略表に戻ります。

専用 RAM テスト・レジスタのロック・レジスタ

図 1-99. DxRAMTEST_LOCK レジスタ

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			PIEVECT	RESERVED	RESERVED	M1	M0
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-113. DxRAMTEST_LOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	KEY	R-0/W	0h	このレジスタの他のフィールドへの書き込みが成功するためには、同時にこのフィールドに 0xa5a5 の値を書き込む必要があります。 リセット・タイプ:SYSRSn
15-5	RESERVED	R	0h	予約済み
4	PIEVECT	R/W	0h	0: DxTEST.TEST_PIEVECT フィールドへの書き込みを許可します。 1: DxTEST.TEST_PIEVECT フィールドへの書き込みをブロックします リセット・タイプ:SYSRSn
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	M1	R/W	0h	0: DxTEST.TEST_M1 フィールドへの書き込みを許可します。 1: DxTEST.TEST_M1 フィールドへの書き込みをブロックします リセット・タイプ:SYSRSn
0	M0	R/W	0h	0: DxTEST.TEST_M0 フィールドへの書き込みを許可します。 1: DxTEST.TEST_M0 フィールドへの書き込みをブロックします リセット・タイプ:SYSRSn

1.15.7.9 LSxLOCK レジスタ (オフセット = 20h) [リセット = 0h]

図 1-100 に LSxLOCK を示し、表 1-114 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM 構成ロック・レジスタ

図 1-100. LSxLOCK レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	LOCK_LS1	LOCK_LS0
R/W-0h							

表 1-114. LSxLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-8	RESERVED	R	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	LOCK_LS1	R/W	0h	LS1 RAM のアクセス保護、マスタ選択、プログラムまたはデータ・メモリ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みをロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みが可能です。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みはブロックされます。 リセット・タイプ: SYSRSn
0	LOCK_LS0	R/W	0h	LS0 RAM のアクセス保護、マスタ選択、プログラムまたはデータ・メモリ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みをロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みが可能です。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みはブロックされます。 リセット・タイプ: SYSRSn

1.15.7.10 LSxCOMMIT レジスタ (オフセット = 22h) [リセット = 0h]

図 1-101 に LSxCOMMIT を示し、表 1-115 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM 構成ロック・コミット・レジスタ

図 1-101. LSxCOMMIT レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	COMMIT_LS1	COMMIT_LS0
R/WOnce-0h							

表 1-115. LSxCOMMIT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-8	RESERVED	R	0h	予約済み
7	RESERVED	R/WOnce	0h	予約済み
6	RESERVED	R/WOnce	0h	予約済み
5	RESERVED	R/WOnce	0h	予約済み
4	RESERVED	R/WOnce	0h	予約済み
3	RESERVED	R/WOnce	0h	予約済み
2	RESERVED	R/WOnce	0h	予約済み
1	COMMIT_LS1	R/WOnce	0h	LS1 RAM のアクセス保護、マスタ選択、プログラムまたはデータ・メモリ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みを永続的にロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みは、LSxLOCK レジスタの LOCK フィールドの値に基づいて許可されます。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みは永続的にブロックされます。 リセット・タイプ: SYSRSn
0	COMMIT_LS0	R/WOnce	0h	LS0 RAM のアクセス保護、マスタ選択、プログラムまたはデータ・メモリ選択、初期化制御、テスト・レジスタ・フィールドへの書き込みを永続的にロックします。 0: ACCPROT、INIT、MSEL の各フィールドへの書き込みは、LSxLOCK レジスタの LOCK フィールドの値に基づいて許可されます。 1: ACCPROT、INIT、MSEL の各フィールドへの書き込みは永続的にブロックされます。 リセット・タイプ: SYSRSn

1.15.7.11 LSxACCPROT0 レジスタ (オフセット = 28h) [リセット = 0h]

図 1-102 に LSxACCPROT0 を示し、表 1-116 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM 構成レジスタ 0

図 1-102. LSxACCPROT0 レジスタ

31	30	29	28	27	26	25	24
RESERVED						RESERVED	RESERVED
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED						RESERVED	RESERVED
R-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						CPUWRPROT_ LS1	FETCHPROT_ LS1
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED						CPUWRPROT_ LS0	FETCHPROT_ LS0
R-0h						R/W-0h	R/W-0h

表 1-116. LSxACCPROT0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-26	RESERVED	R	0h	予約済み
25	RESERVED	R/W	0h	予約済み
24	RESERVED	R/W	0h	予約済み
23-18	RESERVED	R	0h	予約済み
17	RESERVED	R/W	0h	予約済み
16	RESERVED	R/W	0h	予約済み
15-10	RESERVED	R	0h	予約済み
9	CPUWRPROT_LS1	R/W	0h	LS1 RAM に対する CPU 書き込み保護: 0: CPU の書き込みが可能です。 1: CPU の書き込みがブロックされます。 リセット・タイプ: SYSRSn
8	FETCHPROT_LS1	R/W	0h	LS1 RAM に対するフェッチ保護: 0: CPU のフェッチが可能です。 1: CPU のフェッチがブロックされます。 リセット・タイプ: SYSRSn
7-2	RESERVED	R	0h	予約済み
1	CPUWRPROT_LS0	R/W	0h	LS0 RAM に対する CPU 書き込み保護: 0: CPU の書き込みが可能です。 1: CPU の書き込みがブロックされます。 リセット・タイプ: SYSRSn
0	FETCHPROT_LS0	R/W	0h	LS0 RAM に対するフェッチ保護: 0: CPU のフェッチが可能です。 1: CPU のフェッチがブロックされます。 リセット・タイプ: SYSRSn

1.15.7.12 LSxTEST レジスタ (オフセット = 30h) [リセット = 0h]

図 1-103 に LSxTEST を示し、表 1-117 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM テスト・レジスタ

図 1-103. LSxTEST レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED		RESERVED		RESERVED		RESERVED	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		RESERVED		TEST_LS1		TEST_LS0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-117. LSxTEST レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-14	RESERVED	R/W	0h	予約済み
13-12	RESERVED	R/W	0h	予約済み
11-10	RESERVED	R/W	0h	予約済み
9-8	RESERVED	R/W	0h	予約済み
7-6	RESERVED	R/W	0h	予約済み
5-4	RESERVED	R/W	0h	予約済み
3-2	TEST_LS1	R/W	0h	LS1 RAM のいくつかのモードを選択します。 00: 機能モード。 01: 書き込みはデータ・ビットに対してのみ許可されます。パリティ・ビットへの書き込み不可。 10: 書き込みはパリティ・ビットに対してのみ許可されます。データ・ビットへの書き込み不可。 11: 機能モードと同じですが、エラー発生時に割り込み/NMI は生成されません。 注: 0 以外の値を指定した場合、書き込み保護に優先して CPU による書き込みが可能になり、アクセス保護違反は発生しません。 リセット・タイプ: SYSRSn
1-0	TEST_LS0	R/W	0h	LS0 RAM のいくつかのモードを選択します。 00: 機能モード。 01: 書き込みはデータ・ビットに対してのみ許可されます。パリティ・ビットへの書き込み不可。 10: 書き込みはパリティ・ビットに対してのみ許可されます。データ・ビットへの書き込み不可。 11: 機能モードと同じですが、エラー発生時に割り込み/NMI は生成されません。 注: 0 以外の値を指定した場合、書き込み保護に優先して CPU による書き込みが可能になり、アクセス保護違反は発生しません。 リセット・タイプ: SYSRSn

1.15.7.13 LSxINIT レジスタ (オフセット = 32h) [リセット = 0h]

図 1-104 に LSxINIT を示し、表 1-118 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM 初期化レジスタ

図 1-104. LSxINIT レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	INIT_LS1	INIT_LS0
R-0/W1S-0h							

表 1-118. LSxINIT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-8	RESERVED	R	0h	予約済み
7	RESERVED	R-0/W1S	0h	予約済み
6	RESERVED	R-0/W1S	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み
4	RESERVED	R-0/W1S	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	INIT_LS1	R-0/W1S	0h	LS1 RAM の RAM 初期化制御: 0:なし。 1:RAM 初期化を開始します。 リセット・タイプ:SYRSn
0	INIT_LS0	R-0/W1S	0h	LS0 RAM の RAM 初期化制御: 0:なし。 1:RAM 初期化を開始します。 リセット・タイプ:SYRSn

1.15.7.14 LSxINITDONE レジスタ (オフセット = 34h) [リセット = 0h]

図 1-105 に LSxINITDONE を示し、表 1-119 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM 初期化完了ステータス・レジスタ

図 1-105. LSxINITDONE レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	INITDONE_LS1	INITDONE_LS0
R-0h	R-0h						

表 1-119. LSxINITDONE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-8	RESERVED	R	0h	予約済み
7	RESERVED	R	0h	予約済み
6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	RESERVED	R	0h	予約済み
3	RESERVED	R	0h	予約済み
2	RESERVED	R	0h	予約済み
1	INITDONE_LS1	R	0h	LS1 RAM の RAM 初期化ステータス: 0:RAM の初期化は行われていません。 1:RAM の初期化が完了しました。 リセット・タイプ:SYRSn
0	INITDONE_LS0	R	0h	LS0 RAM の RAM 初期化ステータス: 0:RAM の初期化は行われていません。 1:RAM の初期化が完了しました。 リセット・タイプ:SYRSn

1.15.7.15 LSxRAMTEST_LOCK レジスタ (オフセット = 36h) [リセット = 0h]

図 1-106 に LSxRAMTEST_LOCK を示し、表 1-120 にその説明を示します。

概略表に戻ります。

ローカル共有 RAM テスト・レジスタのロック・レジスタ

図 1-106. LSxRAMTEST_LOCK レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
KEY																
R-0/W-0h																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED								RESE RVED	RESE RVED	RESE RVED	RESE RVED	RESE RVED	RESE RVED	LS1	LS0	
R-0h								R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-120. LSxRAMTEST_LOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	KEY	R-0/W	0h	このレジスタの他のフィールドへの書き込みが成功するためには、同時にこのフィールドに 0xa5a5 の値を書き込む必要があります。 リセット・タイプ: SYSRSn
15-8	RESERVED	R	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RESERVED	R/W	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	LS1	R/W	0h	0: LSxTEST.TEST_LS1 フィールドへの書き込みを許可します。 1: LSxTEST.TEST_LS1 フィールドへの書き込みをブロックします。 リセット・タイプ: SYSRSn
0	LS0	R/W	0h	0: LSxTEST.TEST_LS0 フィールドへの書き込みを許可します。 1: LSxTEST.TEST_LS0 フィールドへの書き込みをブロックします。 リセット・タイプ: SYSRSn

1.15.7.16 ROM_LOCK レジスタ (オフセット = A0h) [リセット = 0h]

図 1-107 に ROM_LOCK を示し、表 1-121 にその説明を示します。

概略表に戻ります。

ROM 構成ロック・レジスタ

図 1-107. ROM_LOCK レジスタ

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	LOCK_BOOTROM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-121. ROM_LOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	KEY	R-0/W	0h	このレジスタの他のフィールドへの書き込みが成功するためには、同時にこのフィールドに 0xa5a5 の値を書き込む必要があります。 リセット・タイプ:SYSRSn
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	LOCK_BOOTROM	R/W	0h	BOOTROM のテスト制御フィールド (TEST および FORCE_ERROR) への書き込みアクセスをロックします。 0: 書き込みアクセスを許可します 1: 書き込みアクセスをブロックします リセット・タイプ:SYSRSn

1.15.7.17 ROM_TEST レジスタ (オフセット = A2h) [リセット = 0h]

図 1-108 に ROM_TEST を示し、表 1-122 にその説明を示します。

概略表に戻ります。

ROM テスト・レジスタ

図 1-108. ROM_TEST レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED		RESERVED		TEST_BOOTROM	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-122. ROM_TEST レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-8	RESERVED	R/W	0h	予約済み
7-6	RESERVED	R/W	0h	予約済み
5-4	RESERVED	R/W	0h	予約済み
3-2	RESERVED	R/W	0h	予約済み
1-0	TEST_BOOTROM	R/W	0h	BOOTROM のいくつかのモードを選択します。 00: 機能モード。 01: 「00」と同じですが、データ読み取り時のパリティ・チェックがディセーブルになります (デバッグ用) 10: パリティ・ビットをメモリ・マップ上で見ることができます (デバッグ用) 11: 「00」と同じですが、エラー時に NMI が生成されません。診断に使用します。(診断用) リセット・タイプ: SYSRSn

1.15.7.18 ROM_FORCE_ERROR レジスタ (オフセット = A4h) [リセット = 0h]

図 1-109 に ROM_FORCE_ERROR を示し、表 1-123 にその説明を示します。

概略表に戻ります。

ROM 強制エラー・レジスタ

図 1-109. ROM_FORCE_ERROR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	FORCE_BOOT ROM_ERROR
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-123. ROM_FORCE_ERROR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R/W	0h	予約済み
2	RESERVED	R/W	0h	予約済み
1	RESERVED	R/W	0h	予約済み
0	FORCE_BOOTROM_ERROR	R/W	0h	反転されたパリティ・ビットをパリティ・チェック・ロジックに供給することで、パリティ・エラーを強制的に発生させます。 リセット・タイプ:SYSRSn

1.15.8 MEMORY_ERROR_REGS レジスタ

表 1-124 に、MEMORY_ERROR_REGS レジスタのメモリマップされたレジスタを示します。表 1-124 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-124. MEMORY_ERROR_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	UCERRFLG	訂正不可能エラー・フラグ・レジスタ		表示
2h	UCERRSET	訂正不可能エラー・フラグ・セット・レジスタ	EALLOW	表示
4h	UCERRCLR	訂正不可能エラー・フラグ・クリア・レジスタ	EALLOW	表示
6h	UCCPUREADDR	訂正不可能 CPU 読み取りエラー・アドレス		表示
1Ch	FLUCERRSTATUS	訂正不可能フラッシュ読み取り ECC エラー・ステータス		表示
1Eh	FLCERRSTATUS	訂正可能フラッシュ読み取り ECC エラー・ステータス		表示
20h	CERRFLG	訂正可能エラー・フラグ・レジスタ		表示
22h	CERRSET	訂正可能エラー・フラグ・セット・レジスタ	EALLOW	表示
24h	CERRCLR	訂正可能エラー・フラグ・クリア・レジスタ	EALLOW	表示
26h	CCPUREADDR	訂正可能 CPU 読み取りエラー・アドレス		表示
2Eh	CERRCNT	訂正可能エラー・カウント・レジスタ		表示
30h	CERRTHRES	訂正可能エラー・スレッシュホールド値レジスタ	EALLOW	表示
32h	CEINTFLG	訂正可能エラー割り込みフラグ・ステータス・レジスタ		表示
34h	CEINTCLR	訂正可能エラー割り込みフラグ・クリア・レジスタ	EALLOW	表示
36h	CEINTSET	訂正可能エラー割り込みフラグ・セット・レジスタ	EALLOW	表示
38h	CEINTEN	訂正可能エラー割り込みイネーブル・レジスタ	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-125 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-125. MEMORY_ERROR_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W1S	書き込み 1 でセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.8.1 UCERRFLG レジスタ (オフセット = 0h) [リセット = 0h]

図 1-110 に UCERRFLG を示し、表 1-126 にその説明を示します。

概略表に戻ります。

訂正不可能エラー・フラグ・レジスタ

図 1-110. UCERRFLG レジスタ

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

表 1-126. UCERRFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	RESERVED	R	0h	予約済み
3	RESERVED	R	0h	予約済み
2	RESERVED	R	0h	予約済み
1	RESERVED	R	0h	予約済み
0	CPURDERR	R	0h	訂正不可能 CPU 読み取りエラー・フラグ 0: エラーなし。 1: CPU 読み取り中に訂正不可能エラーが発生しました。 リセット・タイプ: SYRSn

1.15.8.2 UCERRSET レジスタ (オフセット = 2h) [リセット = 0h]

図 1-111 に UCERRSET を示し、表 1-127 にその説明を示します。

概略表に戻ります。

訂正不可能エラー・フラグ・セット・レジスタ

図 1-111. UCERRSET レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CPURDERR
R-0h		R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-127. UCERRSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-6	RESERVED	R	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み
4	RESERVED	R	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	RESERVED	R-0/W1S	0h	予約済み
0	CPURDERR	R-0/W1S	0h	0:動作なし。 1:UCERRFLG レジスタの CPU 読み取りエラー・フラグがセットされ、イネーブルになっている場合は割り込みが生成されます。 リセット・タイプ:SYSRSn

1.15.8.3 UCERRCLR レジスタ (オフセット = 4h) [リセット = 0h]

図 1-112 に UCERRCLR を示し、表 1-128 にその説明を示します。

概略表に戻ります。

訂正不可能エラー・フラグ・クリア・レジスタ

図 1-112. UCERRCLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CPURDERR
R-0h		R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-128. UCERRCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-6	RESERVED	R	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み
4	RESERVED	R	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	RESERVED	R-0/W1S	0h	予約済み
0	CPURDERR	R-0/W1S	0h	0:動作なし。 1:UCERRFLG レジスタの CPU 読み取りエラー・フラグがクリアされます。 リセット・タイプ:SYRSn

1.15.8.4 UCCPUREADDR レジスタ (オフセット = 6h) [リセット = 0h]

図 1-113 に UCCPUREADDR を示し、表 1-129 にその説明を示します。

概略表に戻ります。

訂正不可能 CPU 読み取りエラー・アドレス

図 1-113. UCCPUREADDR レジスタ

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

表 1-129. UCCPUREADDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	UCCPUREADDR	R	0h	このレジスタは、CPU の読み取り/フェッチアクセスによって訂正不可能 ECC/パリティ・エラーが発生したアドレス位置をキャプチャします。 リセット・タイプ:SYSRSn

1.15.8.5 FLUCERRSTATUS レジスタ (オフセット = 1Ch) [リセット = 0h]

図 1-114 に FLUCERRSTATUS を示し、表 1-130 にその説明を示します。

概略表に戻ります。

訂正不可能フラッシュ読み取り ECC エラー・ステータス

図 1-114. FLUCERRSTATUS レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED						DIAG_H_FAIL	UNC_ERR_H
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						DIAG_L_FAIL	UNC_ERR_L
R-0h						R-0h	R-0h

表 1-130. FLUCERRSTATUS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-10	RESERVED	R	0h	予約済み
9	DIAG_H_FAIL	R	0h	冗長および機能 ECC ロジック比較チェックのステータス。 0: 冗長および機能 ECC ロジック比較に成功。 1: 冗長および機能 ECC ロジック比較に失敗。 注: * このフィールドは、UNC_ERR_H と同時に更新されます * このフラグは、UCERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * UCERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ: SYSRSn
8	UNC_ERR_H	R	0h	訂正不可能エラー。この値が 1 のとき、128 ビット・アライン・アドレスの上位 64 ビットで訂正不可能エラーが発生したことを示します。 注: * このフラグは、UCERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * UCERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ: SYSRSn
7-2	RESERVED	R	0h	予約済み
1	DIAG_L_FAIL	R	0h	冗長および機能 ECC ロジック比較チェックのステータス。 0: 冗長および機能 ECC ロジック比較に成功。 1: 冗長および機能 ECC ロジック比較に失敗。 注: * このフィールドは、UNC_ERR_L と同時に更新されます * このフラグは、UCERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * UCERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ: SYSRSn

表 1-130. FLUCERRSTATUS レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	UNC_ERR_L	R	0h	訂正不可能エラー。この値が 1 のとき、128 ビット・アライン・アドレスの下位 64 ビットで訂正不可能エラーが発生したことを示します。 注: * このフラグは、UCERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * UCERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ:SYSRSn

1.15.8.6 FLCERRSTATUS レジスタ (オフセット = 1Eh) [リセット = 0h]

図 1-115 に FLCERRSTATUS を示し、表 1-131 にその説明を示します。

概略表に戻ります。

訂正可能フラッシュ読み取り ECC エラー・ステータス

図 1-115. FLCERRSTATUS レジスタ

31	30	29	28	27	26	25	24
RESERVED		ERR_TYPE_H	ERR_POS_H				
R-0h		R-0h	R-0h				
23	22	21	20	19	18	17	16
ERR_POS_H	ERR_TYPE_L	ERR_POS_L					
R-0h	R-0h	R-0h					
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		RESERVED	FAIL_1_H	FAIL_0_H	RESERVED	FAIL_1_L	FAIL_0_L
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-131. FLCERRSTATUS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	RESERVED	R	0h	予約済み
29	ERR_TYPE_H	R	0h	エラー・タイプ 0 は、128 ビット・アライン・アドレスの上位 64 ビットでシングル・ビット・エラーが発生したことを示します。 1 は、128 ビット・アライン・アドレスの上位 64 ビットの ECC チェック・ビットでシングル・ビット・エラーが発生したことを示します。 注: * このフィールドは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフィールドに影響を与えません。 リセット・タイプ: SYSRStn
28-23	ERR_POS_H	R	0h	エラー位置。128 ビット・アライン・アドレスの上位 64 ビットにおけるシングル・ビット・エラーのビット位置。この位置は、ERR_TYPE ビットがチェック・ビットであるかデータ・ビットであるかに応じて解釈されます。ERR_TYPE がチェック・ビット・エラーを示している場合、エラー位置の範囲は 0~7 です。それ以外の場合、範囲は 0~63 です。 注: * このフィールドは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフィールドに影響を与えません。 リセット・タイプ: SYSRStn
22	ERR_TYPE_L	R	0h	エラー・タイプ 0 は、128 ビット・アライン・アドレスの下位 64 ビットでシングル・ビット・エラーが発生したことを示します。 1 は、128 ビット・アライン・アドレスの下位 64 ビットの ECC チェック・ビットでシングル・ビット・エラーが発生したことを示します。 注: * このフィールドは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフィールドに影響を与えません。 リセット・タイプ: SYSRStn

表 1-131. FLCERRSTATUS レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
21-16	ERR_POS_L	R	0h	エラー位置。128 ビット・アライン・アドレスの下位 64 ビットにおけるシングル・ビット・エラーのビット位置。この位置は、ERR_TYPE ビットがチェック・ビットであるかデータ・ビットであるかに応じて解釈されます。ERR_TYPE がチェック・ビット・エラーを示している場合、エラー位置の範囲は 0~7 です。それ以外の場合、範囲は 0~63 です。 注: * このフィールドは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフィールドに影響を与えません。 リセット・タイプ:SYSRSn
15-6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	FAIL_1_H	R	0h	1 に対する誤り。 0 128 ビット・アライン・アドレスの上位 64 ビットで、1 に対する誤りのシングル・ビット・エラーは発生しませんでした。 1 この値が 1 のとき、128 ビット・アライン・アドレスの上位 64 ビットでシングル・ビット・エラーが発生し、訂正された値が 1 であることを示します。 注: * このフラグは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ:SYSRSn
3	FAIL_0_H	R	0h	0 に対する誤り。 0 128 ビット・アライン・アドレスの上位 64 ビットで、0 に対する誤りのシングル・ビット・エラーは発生しませんでした。 1 この値が 1 のとき、128 ビット・アライン・アドレスの上位 64 ビットでシングル・ビット・エラーが発生し、訂正された値が 0 であることを示します。 注: * このフラグは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ:SYSRSn
2	RESERVED	R	0h	予約済み
1	FAIL_1_L	R	0h	1 に対する誤り。 0 128 ビット・アライン・アドレスの下位 64 ビットで、1 に対する誤りのシングル・ビット・エラーは発生しませんでした。 1 この値が 1 のとき、128 ビット・アライン・アドレスの下位 64 ビットでシングル・ビット・エラーが発生し、訂正された値が 1 であることを示します。 注: * このフラグは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ:SYSRSn
0	FAIL_0_L	R	0h	0 に対する誤り。 0 128 ビット・アライン・アドレスの下位 64 ビットで、0 に対する誤りのシングル・ビット・エラーは発生しませんでした。 1 この値が 1 のとき、128 ビット・アライン・アドレスの下位 64 ビットでシングル・ビット・エラーが発生し、訂正された値が 0 であることを示します。 注: * このフラグは、CERRCLR.CPURDERR フラグに 1 を書き込むとクリアされます * CERRSET.CPURDERR は、このフラグに影響を与えません。 リセット・タイプ:SYSRSn

1.15.8.7 CERRFLG レジスタ (オフセット = 20h) [リセット = 0h]

図 1-116 に CERRFLG を示し、表 1-132 にその説明を示します。

概略表に戻ります。

訂正可能エラー・フラグ・レジスタ

図 1-116. CERRFLG レジスタ

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

表 1-132. CERRFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R	0h	予約済み
2	RESERVED	R	0h	予約済み
1	RESERVED	R	0h	予約済み
0	CPURDERR	R	0h	訂正可能 CPU 読み取りエラー・フラグ 0: エラーなし。 1: CPU 読み取り中に訂正可能エラーが発生しました。 リセット・タイプ: SYSRSn

1.15.8.8 CERRSET レジスタ (オフセット = 22h) [リセット = 0h]

図 1-117 に CERRSET を示し、表 1-133 にその説明を示します。

概略表に戻ります。

訂正可能エラー・フラグ・セット・レジスタ

図 1-117. CERRSET レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	CPURDERR
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-133. CERRSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	RESERVED	R-0/W1S	0h	予約済み
0	CPURDERR	R-0/W1S	0h	0:動作なし。 1:CERRFLG レジスタの CPU 読み取りエラー・フラグがセットされ、イネーブルになっている場合は割り込みが生成されます。 リセット・タイプ:SYSRSn

1.15.8.9 CERRCLR レジスタ (オフセット = 24h) [リセット = 0h]

図 1-118 に CERRCLR を示し、表 1-134 にその説明を示します。

概略表に戻ります。

訂正可能エラー・フラグ・クリア・レジスタ

図 1-118. CERRCLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	RESERVED	CPURDERR
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-134. CERRCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-4	RESERVED	R	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	RESERVED	R-0/W1S	0h	予約済み
1	RESERVED	R-0/W1S	0h	予約済み
0	CPURDERR	R-0/W1S	0h	0:動作なし。 1:CERRFLG レジスタの CPU 読み取りエラー・フラグがクリアされます。 リセット・タイプ:SYRSn

1.15.8.10 CCPUREADDR レジスタ (オフセット = 26h) [リセット = 0h]

図 1-119 に CCPUREADDR を示し、表 1-135 にその説明を示します。

概略表に戻ります。

訂正可能 CPU 読み取りエラー・アドレス

図 1-119. CCPUREADDR レジスタ

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

表 1-135. CCPUREADDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	CCPUREADDR	R	0h	このレジスタは、CPU の読み取り/フェッチアクセスによって訂正可能 ECC エラーが発生したアドレス位置をキャプチャします。 リセット・タイプ: SYSRSn

1.15.8.11 CERRCNT レジスタ (オフセット = 2Eh) [リセット = 0h]

図 1-120 に CERRCNT を示し、表 1-136 にその説明を示します。

概略表に戻ります。

訂正可能エラー・カウント・レジスタ

図 1-120. CERRCNT レジスタ

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

表 1-136. CERRCNT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	CERRCNT	R	0h	このレジスタには、訂正可能エラーが何回発生したかというカウントが保持されます。 リセット・タイプ:SYSRSn

1.15.8.12 CERRTHRES レジスタ (オフセット = 30h) [リセット = 0h]

図 1-121 に CERRTHRES を示し、表 1-137 にその説明を示します。

概略表に戻ります。

訂正可能エラー・スレッシュホールド値レジスタ

図 1-121. CERRTHRES レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CERRTHRES															
R-0h																R/W-0h															

表 1-137. CERRTHRES レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-0	CERRTHRES	R/W	0h	CERRCNT レジスタの値がこのレジスタに設定されている値より大きい場合、イネーブルになっていれば訂正可能割り込みが生成されます。 リセット・タイプ: SYSRSn

1.15.8.13 CEINTFLG レジスタ (オフセット = 32h) [リセット = 0h]

図 1-122 に CEINTFLG を示し、表 1-138 にその説明を示します。

概略表に戻ります。

訂正可能エラー割り込みフラグ・ステータス・レジスタ

図 1-122. CEINTFLG レジスタ

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

表 1-138. CEINTFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-1	RESERVED	R	0h	予約済み
0	CEINTFLAG	R	0h	訂正エラー・カウント合計スレッシュホールド超過フラグ 0: 訂正可能エラーの合計 < CERTRHRES レジスタで設定されたスレッシュホールド値。 1: 訂正可能エラーの合計 >= CERTRHRES レジスタで設定されたスレッシュホールド値。 リセット・タイプ: SYSRSn

1.15.8.14 CEINTCLR レジスタ (オフセット = 34h) [リセット = 0h]

図 1-123 に CEINTCLR を示し、表 1-139 にその説明を示します。

概略表に戻ります。

訂正可能エラー割り込みフラグ・クリア・レジスタ

図 1-123. CEINTCLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTCLR
R-0h							R-0/W1S-0h

表 1-139. CEINTCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-1	RESERVED	R	0h	予約済み
0	CEINTCLR	R-0/W1S	0h	0:動作なし。 1:CEINTFLG レジスタの訂正エラー・カウント合計超過フラグがクリアされます。 リセット・タイプ:SYSRSn

1.15.8.15 CEINTSET レジスタ (オフセット = 36h) [リセット = 0h]

図 1-124 に CEINTSET を示し、表 1-140 にその説明を示します。

概略表に戻ります。

訂正可能エラー割り込みフラグ・セット・レジスタ

図 1-124. CEINTSET レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTSET
R-0h							R-0/W1S-0h

表 1-140. CEINTSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-1	RESERVED	R	0h	予約済み
0	CEINTSET	R-0/W1S	0h	0:動作なし。 1:CEINTFLG レジスタの訂正エラー・カウント合計スレッシュホールド超過フラグがセットされ、イネーブルになっている場合は割り込みが生成されます。 リセット・タイプ:SYRSRn

1.15.8.16 CEINTEN レジスタ (オフセット = 38h) [リセット = 0h]

図 1-125 に CEINTEN を示し、表 1-141 にその説明を示します。

概略表に戻ります。

訂正可能エラー割り込みイネーブル・レジスタ

図 1-125. CEINTEN レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							CEINTEN
R-0h							R/W-0h

表 1-141. CEINTEN レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	RESERVED	R	0h	予約済み
15-1	RESERVED	R	0h	予約済み
0	CEINTEN	R/W	0h	0: 訂正可能エラー割り込みがディセーブル。 1: 訂正可能エラー割り込みがイネーブル。 リセット・タイプ: SYSRSn

1.15.9 NMI_INTRUPT_REGS レジスタ

NMI_INTRUPT_REGS レジスタのメモリマップされたレジスタを、表 1-142 に示します。表 1-142 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-142. NMI_INTRUPT_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	NMICFG	NMI 構成レジスタ	EALLOW	表示
1h	NMIFLG	NMI フラグ・レジスタ (SYSRsn クリア)		表示
2h	NMIFLGCLR	NMI フラグ・クリア・レジスタ	EALLOW	表示
3h	NMIFLGFRC	NMI フラグ強制レジスタ	EALLOW	表示
4h	NMIWDCNT	NMI ウォッチドッグ・カウンタ・レジスタ		表示
5h	NMIWDPRD	NMI ウォッチドッグ周期レジスタ	EALLOW	表示
6h	NMISHDFLG	NMI シャドウ・フラグ・レジスタ		表示
7h	ERRORSTS	エラー・ピン・ステータス		表示
8h	ERRORSTSCLR	ERRORSTS クリア・レジスタ	EALLOW	表示
9h	ERRORSTSFRC	ERRORSTS 強制レジスタ	EALLOW	表示
Ah	ERRORCTL	エラー・ピン制御レジスタ	EALLOW	表示
Bh	ERRORLOCK	エラー・ピン・レジスタのロック・レジスタ。	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-143 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-143. NMI_INTRUPT_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み取りタイプ		
R	R	読み取り
R-0	R-0	読み取り 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W1S	書き込み 1 で設定
WSonce	WSonce	書き込み 1 回設定
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ・グループの一部であるレジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ配列の値を参照します。

1.15.9.1 NMICFG レジスタ (オフセット = 0h) [リセット = 0h]

図 1-126 に、NMICFG を示し、表 1-144 に、その説明を示します。

[概略表](#)に戻ります。

NMI 構成レジスタ

図 1-126. NMICFG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							NMIE
R-0h							R/W1S-0h

表 1-144. NMICFG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	RESERVED	R	0h	予約済み
0	NMIE	R/W1S	0h	1 に設定すると、任意の条件で C28 CPU への NMI 割り込みが生成され、NMI ウォッチドッグ・カウンタが開始されます。このビットは、ブート・シーケンスの一部として、デバイスのセキュリティ関連の初期化の完了後に設定する必要があります。 0 NMI 無効 1 NMI 有効 リセット・タイプ:SYSRSn

1.15.9.2 NMIFLG レジスタ (オフセット = 1h) [リセット = 0h]

図 1-127 に、NMIFLG を示し、表 1-145 に、その説明を示します。

概略表に戻ります。

NMI フラグ・レジスタ (SYSRSn クリア)

図 1-127. NMIFLG レジスタ

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	NMIINT
R-0h	R-0h						

表 1-145. NMIFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	RESERVED	R	0h	予約済み
13	SWERR	R	0h	SW エラー強制 NMI フラグ:このビットは、NMI が NMIFLGFRC レジスタを介して強制されているかどうかを示します。このビットは、NMIFLGCLR レジスタの該当するビットにユーザーが書き込むか SYSRSn リセットでのみクリアできます。 0: SW エラー強制の生成なし 1: SW エラー NMI を SW で強制。 ユーザーがこのフラグをクリアするまで、他の NMI パルスは生成されません。 リセット・タイプ: SYSRSn
12	RESERVED	R	0h	予約済み
11	RESERVED	R	0h	予約済み
10	RESERVED	R	0h	予約済み
9	RESERVED	R	0h	予約済み
8	RESERVED	R	0h	予約済み
7	RESERVED	R	0h	予約済み
6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	RESERVED	R	0h	予約済み
3	RESERVED	R	0h	予約済み
2	UNCERR	R	0h	フラッシュ / RAM / ROM 修正不可能エラー NMI フラグ:このビットは、メモリ・アクセス (任意のマスタによる) で修正不可能なエラーが発生したかどうか、およびその状態がラッチされたかどうかを示します。このビットは、NMIFLGCLR レジスタの対応するクリア・ビットにユーザーが書き込むか SYSRSn リセットでのみクリアできます。 0: 保留中の修正不可能なエラー状態なし 1: 修正不可能なエラー状態が発生 リセット・タイプ: SYSRSn
1	CLOCKFAIL	R	0h	クロック障害割り込みフラグ:これらのビットは、CLOCKFAIL 状態がラッチされているかどうかを示します。これらのビットは、NMIFLGCLR レジスタの該当するビットにユーザーが書き込むか SYSRSn リセットでのみクリアできます。 0: 保留中の CLOCKFAIL 状態なし 1: CLOCKFAIL 状態が発生 リセット・タイプ: SYSRSn

表 1-145. NMIFLG レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	NMIINT	R	0h	NMI 割り込みフラグ: このビットは、NMI 割り込みが生成されたかどうかを示します。このビットは、NMIFLGCLR レジスタの該当するビットにユーザーが書き込むか、SYSRSn リセットでのみクリアできます。 0: NMI 割り込みの生成なし 1: NMI 割り込みを生成 ユーザーがこのフラグをクリアするまで、他の NMI 割り込みパルスは生成されません。 リセット・タイプ: SYSRSn

1.15.9.3 NMIFLGCLR レジスタ (オフセット = 2h) [リセット = 0h]

図 1-128 に、NMIFLGCLR を示し、表 1-146 に、その説明を示します。

概略表に戻ります。

NMI フラグ・クリア・レジスタ

図 1-128. NMIFLGCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0/W1S-0h						
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	NMIINT
R-0/W1S-0h							

表 1-146. NMIFLGCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	RESERVED	R-0/W1S	0h	予約済み
13	SWERR	R-0/W1S	0h	該当するビットに 1 を書き込むことにより、NMIFLG レジスタと NMISHDFLG レジスタの対応するフラグ・ビットがクリアされます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。 注： [1] ソフトウェアがビットを 0 にクリアしようとしているときに、同じサイクルでハードウェアがビットを 1 に設定しようとする、ハードウェアが優先されます。 [2] ユーザーは先に保留中の FAIL フラグをクリアしてから、NMIINT フラグをクリアする必要があります。 リセット・タイプ: SYSRSn
12	RESERVED	R-0/W1S	0h	予約済み
11	RESERVED	R-0/W1S	0h	予約済み
10	RESERVED	R-0/W1S	0h	予約済み
9	RESERVED	R-0/W1S	0h	予約済み
8	RESERVED	R-0/W1S	0h	予約済み
7	RESERVED	R-0/W1S	0h	予約済み
6	RESERVED	R-0/W1S	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み
4	RESERVED	R-0/W1S	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	UNCERR	R-0/W1S	0h	該当するビットに 1 を書き込むことにより、NMIFLG レジスタと NMISHDFLG レジスタの対応するフラグ・ビットがクリアされます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。 注： [1] ソフトウェアがビットを 0 にクリアしようとしているときに、同じサイクルでハードウェアがビットを 1 に設定しようとする、ハードウェアが優先されます。 [2] ユーザーは先に保留中の FAIL フラグをクリアしてから、NMIINT フラグをクリアする必要があります。 リセット・タイプ: SYSRSn

表 1-146. NMIFLGCLR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
1	CLOCKFAIL	R-0/W1S	0h	該当するビットに 1 を書き込むことにより、NMIFLG レジスタと NMISHDFLG レジスタの対応するフラグ・ビットがクリアされます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。 注: [1] ソフトウェアがビットを 0 にクリアしようとしているときに、同じサイクルでハードウェアがビットを 1 に設定しようとする、ハードウェアが優先されます。 [2] ユーザーは先に保留中の FAIL フラグをクリアしてから、NMIINT フラグをクリアする必要があります。 リセット・タイプ:SYSRSn
0	NMIINT	R-0/W1S	0h	該当するビットに 1 を書き込むことにより、NMIFLG レジスタと NMISHDFLG レジスタの対応するフラグ・ビットがクリアされます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。 注: [1] ソフトウェアがビットを 0 にクリアしようとしているときに、同じサイクルでハードウェアがビットを 1 に設定しようとする、ハードウェアが優先されます。 [2] ユーザーは先に保留中の FAIL フラグをクリアしてから、NMIINT フラグをクリアする必要があります。 リセット・タイプ:SYSRSn

1.15.9.4 NMIFLGFRFC レジスタ (オフセット = 3h) [リセット = 0h]

図 1-129 に、NMIFLGFRFC を示し、表 1-147 に、その説明を示します。

概略表に戻ります。

NMI フラグ強制レジスタ

図 1-129. NMIFLGFRFC レジスタ

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0/W1S-0h						
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	RESERVED
R-0/W1S-0h	R-0h						

表 1-147. NMIFLGFRFC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	RESERVED	R-0/W1S	0h	予約済み
13	SWERR	R-0/W1S	0h	これらのビットに 1 を書き込むと、NMIFLG レジスタと NMISHDFLG レジスタの該当する FAIL フラグが設定されます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。これは、NMI メカニズムをテストするための手段として使用できます。 リセット・タイプ: SYSRSn
12	RESERVED	R-0/W1S	0h	予約済み
11	RESERVED	R-0/W1S	0h	予約済み
10	RESERVED	R-0/W1S	0h	予約済み
9	RESERVED	R-0/W1S	0h	予約済み
8	RESERVED	R-0/W1S	0h	予約済み
7	RESERVED	R-0/W1S	0h	予約済み
6	RESERVED	R-0/W1S	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み
4	RESERVED	R-0/W1S	0h	予約済み
3	RESERVED	R-0/W1S	0h	予約済み
2	UNCERR	R-0/W1S	0h	これらのビットに 1 を書き込むと、NMIFLG レジスタと NMISHDFLG レジスタの該当する FAIL フラグが設定されます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。これは、NMI メカニズムをテストするための手段として使用できます。 リセット・タイプ: SYSRSn
1	CLOCKFAIL	R-0/W1S	0h	これらのビットに 1 を書き込むと、NMIFLG レジスタと NMISHDFLG レジスタの該当する FAIL フラグが設定されます。0 を書き込んだ場合は無視されます。読み取り時は常に 0 に戻ります。これは、NMI メカニズムをテストするための手段として使用できます。 リセット・タイプ: SYSRSn
0	RESERVED	R	0h	予約済み

1.15.9.5 NMIWDCNT レジスタ (オフセット = 4h) [リセット = 0h]

図 1-130 に、NMIWDCNT を示し、表 1-148 に、その説明を示します。

概略表に戻ります。

NMI ウォッチドッグ・カウンタ・レジスタ

図 1-130. NMIWDCNT レジスタ

15	14	13	12	11	10	9	8
NMIWDCNT							
R-0h							
7	6	5	4	3	2	1	0
NMIWDCNT							
R-0h							

表 1-148. NMIWDCNT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	NMIWDCNT	R	0h	<p>NMI ウォッチドッグ・カウンタ: この 16 ビットのインクリメント・カウンタは、イネーブルのいずれかの FAIL フラグが設定されるたびにインクリメントを開始します。カウンタが周期値に達すると、NMIRS_n 信号が発生し、システムがリセットされます。周期値に達するとカウンタがゼロにリセットされ、イネーブルの FAIL フラグが設定されるとカウンタが再開されます。</p> <p>イネーブルの FAIL フラグが設定されていない場合、カウンタはゼロにリセットされ、イネーブルの FAIL フラグが設定されるまでゼロのままになります。</p> <p>通常、NMI ウォッチドッグでリセットがトリガされる前に、生成された NMI 割り込みがソフトウェアで処理され、違反フラグがクリアされます。状況によっては、ソフトウェアの処理に関係なく、ウォッチドッグによるデバイスのリセットが許可される場合もあります。</p> <p>このカウンタのクロックは SYSCLKOUT レートで供給されます。</p> <p>リセット・タイプ: SYSR_S_n</p>

1.15.9.6 NMIWDPRD レジスタ (オフセット = 5h) [リセット = FFFFh]

図 1-131 に、NMIWDPRD を示し、表 1-149 に、その説明を示します。

概略表に戻ります。

NMI ウォッチドッグ周期レジスタ

図 1-131. NMIWDPRD レジスタ

15	14	13	12	11	10	9	8
NMIWDPRD							
R/W-FFFFh							
7	6	5	4	3	2	1	0
NMIWDPRD							
R/W-FFFFh							

表 1-149. NMIWDPRD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	NMIWDPRD	R/W	FFFFh	NMI ウォッチドッグ周期:この 16 ビットの値には、ウォッチドッグ・カウンタが一致したときにリセットが生成される周期値が格納されます。この値はリセット時に最大値に設定されます。ソフトウェアは、初期化時に周期値を減らすことができます。 現在のカウンタ値よりも小さい PERIOD 値を書き込むと、自動的に NMIRSn が強制され、ウォッチドッグ・カウンタがリセットされます。 リセット・タイプ:SYSRSn

1.15.9.7 NMISHDFLG レジスタ (オフセット = 6h) [リセット = 0h]

図 1-132 に、NMISHDFLG を示し、表 1-150 に、その説明を示します。

概略表に戻ります。

NMI シャドウ・フラグ・レジスタ

図 1-132. NMISHDFLG レジスタ

15	14	13	12	11	10	9	8
RESERVED	RESERVED	SWERR	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	UNCERR	CLOCKFAIL	RESERVED
R-0h	R-0h						

表 1-150. NMISHDFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	RESERVED	R	0h	予約済み
13	SWERR	R	0h	シャドウ NMI フラグ: デバイスで使用可能な NMI ソースのいずれかによって NMIFLG ビットが設定されると、このレジスタの対応するビットも設定されます。このレジスタのビットには、NMIFLGFRC レジスタと NMIFLGCLR レジスタも NMIFLG レジスタと同じように影響を及ぼすことに注意してください。このレジスタは PORESETn でのみリセットされます。 注: [1] このレジスタは、「システム制御リセットによるレジスタのクリア」の定義を保持するために追加されたものです。 リセット・タイプ: PORESETn
12	RESERVED	R	0h	予約済み
11	RESERVED	R	0h	予約済み
10	RESERVED	R	0h	予約済み
9	RESERVED	R	0h	予約済み
8	RESERVED	R	0h	予約済み
7	RESERVED	R	0h	予約済み
6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	RESERVED	R	0h	予約済み
3	RESERVED	R	0h	予約済み
2	UNCERR	R	0h	シャドウ NMI フラグ: デバイスで使用可能な NMI ソースのいずれかによって NMIFLG ビットが設定されると、このレジスタの対応するビットも設定されます。このレジスタのビットには、NMIFLGFRC レジスタと NMIFLGCLR レジスタも NMIFLG レジスタと同じように影響を及ぼすことに注意してください。このレジスタは PORESETn でのみリセットされます。 注: [1] このレジスタは、「システム制御リセットによるレジスタのクリア」の定義を保持するために追加されたものです。 リセット・タイプ: PORESETn

表 1-150. NMISHDFLG レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
1	CLOCKFAIL	R	0h	シェドウ NMI フラグ: デバイスで使用可能な NMI ソースのいずれかによって NMIFLG ビットが設定されると、このレジスタの対応するビットも設定されます。このレジスタのビットには、NMIFLGFRC レジスタと NMIFLGCLR レジスタも NMIFLG レジスタと同じように影響を及ぼすことに注意してください。このレジスタは PORESETn でのみリセットされます。 注: [1] このレジスタは、「システム制御リセットによるレジスタのクリア」の定義を保持するために追加されたものです。 リセット・タイプ: PORESETn
0	RESERVED	R	0h	予約済み

1.15.9.8 ERRORSTS レジスタ (オフセット = 7h) [リセット = 0h]

図 1-133 に、ERRORSTS を示し、表 1-151 に、その説明を示します。

概略表に戻ります。

エラー・ピン・ステータス

図 1-133. ERRORSTS レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						PINSTS	ERROR
R-0h						R-0h	R-0h

表 1-151. ERRORSTS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-2	RESERVED	R	0h	予約済み
1	PINSTS	R	0h	0: エラー・ピンは 0 1: エラー・ピンは 1 リセット・タイプ: PORESETn
0	エラー	R	0h	0: エラー・ソースはトリガされていません。 1: 1 つ以上のエラー・ソースがトリガされたか、ERRORSTS.FRRC.ERROR ビットへの 1 の書き込みによって ERRORSTS.ERROR が設定されました。ERROR フラグの設定後は、ERRORSTS.SCLR.ERROR ビットに 1 を書き込むことでクリアできます。このビットを設定できるイベント/トリガは次のとおりです。 1. C28x での NMI 割り込み 2. ウォッチドッグ・リセット 3. PIE ベクタ・フェッチのエラー 4. eFuse エラー このビットの読み取り時に、ピンのエラー・ピン状態が返されます。 リセット・タイプ: PORESETn

1.15.9.9 ERRORSTSCLR レジスタ (オフセット = 8h) [リセット = 0h]

図 1-134 に、ERRORSTSCLR を示し、表 1-152 に、その説明を示します。

概略表に戻ります。

ERRORSTS クリア・レジスタ

図 1-134. ERRORSTSCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							ERROR
R-0h							R-0/W1S-0h

表 1-152. ERRORSTSCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	RESERVED	R	0h	予約済み
0	エラー	R-0/W1S	0h	0: 影響なし 1: ERRORSTS.ERROR を 0 にクリア リセット・タイプ: PORESETn

1.15.9.10 ERRORSTSFRC レジスタ (オフセット = 9h) [リセット = 0h]

図 1-135 に、ERRORSTSFRC を示し、表 1-153 に、その説明を示します。

概略表に戻ります。

ERRORSTS 強制レジスタ

図 1-135. ERRORSTSFRC レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							ERROR
R-0h							R-0/W1S-0h

表 1-153. ERRORSTSFRC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	RESERVED	R	0h	予約済み
0	エラー	R-0/W1S	0h	0: 影響なし 1: ERRORSTS.ERROR を 1 に設定 リセット・タイプ: PORESETn

1.15.9.11 ERRORCTL レジスタ (オフセット = Ah) [リセット = 0h]

図 1-136 に、ERRORCTL を示し、表 1-154 に、その説明を示します。

概略表に戻ります。

エラー・ピン制御レジスタ

図 1-136. ERRORCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							ERRORPOLSE L
R-0h							R/W-0h

表 1-154. ERRORCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	RESERVED	R	0h	予約済み
0	ERRORPOLSEL	R/W	0h	0: ERRORSTS.ERROR が 1 の場合に値 0 でエラー・ピンが駆動されま す (それ以外の場合は 1)。 1: ERRORSTS.ERROR が 1 の場合に値 1 でエラー・ピンが駆動されま す (それ以外の場合は 0)。 リセット・タイプ: PORESETn

1.15.9.12 ERRORLOCK レジスタ (オフセット = Bh) [リセット = 0h]

図 1-137 に、ERRORLOCK を示し、表 1-155 に、その説明を示します。

概略表に戻ります。

エラー・ピン・レジスタのロック・レジスタ。

図 1-137. ERRORLOCK レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							ERRORCTL
R-0h							R/WSonce-0h

表 1-155. ERRORLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	RESERVED	R	0h	予約済み
0	ERRORCTL	R/WSonce	0h	0: ERRORCTL レジスタへの書き込みが許可されます。 1: ERRORCTL レジスタへの書き込みがブロックされます。 このビットに 0 を書き込んでも効果はありません。このビットは 1 を書き込むと設定され、SYSRSn でのみクリアされます。 リセット・タイプ: SYSRSn

1.15.10 PIE_CTRL_REGS レジスタ

PIE_CTRL_REGS レジスタのメモリマップされたレジスタを、表 1-156 に示します。表 1-156 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-156. PIE_CTRL_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	PIECTRL	ePIE 制御レジスタ		表示
1h	PIEACK	割り込みアクノリッジ・レジスタ		表示
2h	PIEIER1	割り込みグループ 1 イネーブル・レジスタ		表示
3h	PIEIFR1	割り込みグループ 1 フラグ・レジスタ		表示
4h	PIEIER2	割り込みグループ 2 イネーブル・レジスタ		表示
5h	PIEIFR2	割り込みグループ 2 フラグ・レジスタ		表示
6h	PIEIER3	割り込みグループ 3 イネーブル・レジスタ		表示
7h	PIEIFR3	割り込みグループ 3 フラグ・レジスタ		表示
8h	PIEIER4	割り込みグループ 4 イネーブル・レジスタ		表示
9h	PIEIFR4	割り込みグループ 4 フラグ・レジスタ		表示
Ah	PIEIER5	割り込みグループ 5 イネーブル・レジスタ		表示
Bh	PIEIFR5	割り込みグループ 5 フラグ・レジスタ		表示
Ch	PIEIER6	割り込みグループ 6 イネーブル・レジスタ		表示
Dh	PIEIFR6	割り込みグループ 6 フラグ・レジスタ		表示
Eh	PIEIER7	割り込みグループ 7 イネーブル・レジスタ		表示
Fh	PIEIFR7	割り込みグループ 7 フラグ・レジスタ		表示
10h	PIEIER8	割り込みグループ 8 イネーブル・レジスタ		表示
11h	PIEIFR8	割り込みグループ 8 フラグ・レジスタ		表示
12h	PIEIER9	割り込みグループ 9 イネーブル・レジスタ		表示
13h	PIEIFR9	割り込みグループ 9 フラグ・レジスタ		表示
14h	PIEIER10	割り込みグループ 10 イネーブル・レジスタ		表示
15h	PIEIFR10	割り込みグループ 10 フラグ・レジスタ		表示
16h	PIEIER11	割り込みグループ 11 イネーブル・レジスタ		表示
17h	PIEIFR11	割り込みグループ 11 フラグ・レジスタ		表示
18h	PIEIER12	割り込みグループ 12 イネーブル・レジスタ		表示
19h	PIEIFR12	割り込みグループ 12 フラグ・レジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-157 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-157. PIE_CTRL_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み取りタイプ		
R	R	読み取り
R-0	R-0	読み取り 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W1S	書き込み 1 で設定
リセットまたはデフォルト値		

**表 1-157. PIE_CTRL_REGS のアクセス・タイプ・コード
(continued)**

アクセス・タイプ	コード	説明
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ・グループの一部であるレジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ配列の値を参照します。

1.15.10.1 PIECTRL レジスタ (オフセット = 0h) [リセット = 0h]

図 1-138 に、PIECTRL を示し、表 1-158 に、その説明を示します。

概略表に戻ります。

ePIE 制御レジスタ

図 1-138. PIECTRL レジスタ

15	14	13	12	11	10	9	8
PIEVECT							
R-0h							
7	6	5	4	3	2	1	0
PIEVECT							ENPIE
R-0h							R/W-0h

表 1-158. PIECTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	PIEVECT	R	0h	これらのビットは、ePIE ベクタ・テーブルからフェッチされたベクタのベクタ・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、アドレスのビット 1 ~ 15 のみが表示されます。ユーザーはベクタ値を読み取ることで、どの割り込みでベクタ・フェッチが生成されたかを判別できます。 注:NMI の処理では、PIEVECT ビット・フィールドに他の割り込みの場合のようにベクタは反映されません。 リセット・タイプ:SYSRSn
0	ENPIE	R/W	0h	ePIE ブロックからのベクタ・フェッチをイネーブルにします。ペリフェラル割り込みが機能するには、このビットを 1 に設定する必要があります。ePIE レジスタ (PIEACK、PIEIFR、PIEIER) には、ePIE ブロックがディセーブルの場合でもすべてアクセスできます。 リセット・タイプ:SYSRSn

1.15.10.2 PIEACK レジスタ (オフセット = 1h) [リセット = 0h]

図 1-139 に、PIEACK を示し、表 1-159 に、その説明を示します。

概略表に戻ります。

アクアリッジ・レジスタ

ePIE から CPU 割り込みラインに割り込みが伝搬すると、割り込みグループの PIEACK ビットが設定されます。これにより、最初の割り込みが処理されている間、そのグループの他の割り込みが CPU に伝搬されなくなります。PIEACK ビットに 1 を書き込むとビットがクリアされ、対応するグループから別の割り込みが伝搬されます。PIE 割り込み用の ISR は、割り込みから復帰する前にグループの PIEACK ビットをクリアする必要があります。

0 を書き込んだ場合は無視されます。

図 1-139. PIEACK レジスタ

15	14	13	12	11	10	9	8
RESERVED				ACK12	ACK11	ACK10	ACK9
R-0-0h				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
ACK8	ACK7	ACK6	ACK5	ACK4	ACK3	ACK2	ACK1
R/W1S-0h							

表 1-159. PIEACK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11	ACK12	R/W1S	0h	アクアリッジ PIE 割り込みグループ 12 リセット・タイプ: SYSRSn
10	ACK11	R/W1S	0h	アクアリッジ PIE 割り込みグループ 11 リセット・タイプ: SYSRSn
9	ACK10	R/W1S	0h	アクアリッジ PIE 割り込みグループ 10 リセット・タイプ: SYSRSn
8	ACK9	R/W1S	0h	アクアリッジ PIE 割り込みグループ 9 リセット・タイプ: SYSRSn
7	ACK8	R/W1S	0h	アクアリッジ PIE 割り込みグループ 8 リセット・タイプ: SYSRSn
6	ACK7	R/W1S	0h	アクアリッジ PIE 割り込みグループ 7 リセット・タイプ: SYSRSn
5	ACK6	R/W1S	0h	アクアリッジ PIE 割り込みグループ 6 リセット・タイプ: SYSRSn
4	ACK5	R/W1S	0h	アクアリッジ PIE 割り込みグループ 5 リセット・タイプ: SYSRSn
3	ACK4	R/W1S	0h	アクアリッジ PIE 割り込みグループ 4 リセット・タイプ: SYSRSn
2	ACK3	R/W1S	0h	アクアリッジ PIE 割り込みグループ 3 リセット・タイプ: SYSRSn
1	ACK2	R/W1S	0h	アクアリッジ PIE 割り込みグループ 2 リセット・タイプ: SYSRSn
0	ACK1	R/W1S	0h	アクアリッジ PIE 割り込みグループ 1 リセット・タイプ: SYSRSn

1.15.10.3 PIEIER1 レジスタ (オフセット = 2h) [リセット = 0h]

図 1-140 に、PIEIER1 を示し、表 1-160 に、その説明を示します。

概略表に戻ります。

割り込みグループ 1 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-140. PIEIER1 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-160. PIEIER1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 1.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 1.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 1.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 1.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 1.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 1.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 1.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 1.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.4 PIEIFR1 レジスタ (オフセット = 3h) [リセット = 0h]

図 1-141 に、PIEIFR1 を示し、表 1-161 に、その説明を示します。

概略表に戻ります。

割り込みグループ 1 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-141. PIEIFR1 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-161. PIEIFR1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 1.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 1.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 1.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 1.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 1.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 1.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 1.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 1.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.5 PIEIER2 レジスタ (オフセット = 4h) [リセット = 0h]

図 1-142 に、PIEIER2 を示し、表 1-162 に、その説明を示します。

概略表に戻ります。

割り込みグループ 2 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-142. PIEIER2 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-162. PIEIER2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 2.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 2.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 2.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 2.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 2.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 2.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 2.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 2.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.6 PIEIFR2 レジスタ (オフセット = 5h) [リセット = 0h]

図 1-143 に、PIEIFR2 を示し、表 1-163 に、その説明を示します。

概略表に戻ります。

割り込みグループ 2 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-143. PIEIFR2 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-163. PIEIFR2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 2.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 2.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 2.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 2.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 2.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 2.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 2.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 2.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.7 PIEIER3 レジスタ (オフセット = 6h) [リセット = 0h]

図 1-144 に、PIEIER3 を示し、表 1-164 に、その説明を示します。

概略表に戻ります。

割り込みグループ 3 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-144. PIEIER3 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-164. PIEIER3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 3.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 3.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 3.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 3.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 3.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 3.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 3.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 3.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.8 PIEIFR3 レジスタ (オフセット = 7h) [リセット = 0h]

図 1-145 に、PIEIFR3 を示し、表 1-165 に、その説明を示します。

概略表に戻ります。

割り込みグループ 3 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-145. PIEIFR3 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-165. PIEIFR3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 3.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 3.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 3.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 3.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 3.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 3.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 3.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 3.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.9 PIEIER4 レジスタ (オフセット = 8h) [リセット = 0h]

図 1-146 に、PIEIER4 を示し、表 1-166 に、その説明を示します。

概略表に戻ります。

割り込みグループ 4 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-146. PIEIER4 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-166. PIEIER4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 4.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 4.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 4.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 4.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 4.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 4.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 4.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 4.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.10 PIEIFR4 レジスタ (オフセット = 9h) [リセット = 0h]

図 1-147 に、PIEIFR4 を示し、表 1-167 に、その説明を示します。

概略表に戻ります。

割り込みグループ 4 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-147. PIEIFR4 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-167. PIEIFR4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 4.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 4.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 4.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 4.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 4.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 4.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 4.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 4.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.11 PIEIER5 レジスタ (オフセット= Ah) [リセット= 0h]

図 1-148 に、PIEIER5 を示し、表 1-168 に、その説明を示します。

概略表に戻ります。

割り込みグループ 5 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-148. PIEIER5 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-168. PIEIER5 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 5.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 5.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 5.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 5.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 5.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 5.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 5.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 5.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.12 PIEIFR5 レジスタ (オフセット= Bh) [リセット= 0h]

図 1-149 に、PIEIFR5 を示し、表 1-169 に、その説明を示します。

概略表に戻ります。

割り込みグループ 5 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-149. PIEIFR5 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-169. PIEIFR5 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 5.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 5.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 5.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 5.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 5.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 5.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 5.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 5.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.13 PIEIER6 レジスタ (オフセット= Ch) [リセット= 0h]

図 1-150 に、PIEIER6 を示し、表 1-170 に、その説明を示します。

概略表に戻ります。

割り込みグループ 6 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-150. PIEIER6 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-170. PIEIER6 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 6.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 6.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 6.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 6.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 6.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 6.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 6.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 6.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.14 PIEIFR6 レジスタ (オフセット= Dh) [リセット= 0h]

図 1-151 に、PIEIFR6 を示し、表 1-171 に、その説明を示します。

概略表に戻ります。

割り込みグループ 6 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-151. PIEIFR6 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-171. PIEIFR6 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 6.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 6.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 6.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 6.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 6.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 6.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 6.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 6.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.15 PIEIER7 レジスタ (オフセット= Eh) [リセット= 0h]

図 1-152 に、PIEIER7 を示し、表 1-172 に、その説明を示します。

概略表に戻ります。

割り込みグループ 7 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-152. PIEIER7 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-172. PIEIER7 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 7.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 7.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 7.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 7.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 7.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 7.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 7.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 7.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.16 PIEIFR7 レジスタ (オフセット= Fh) [リセット= 0h]

図 1-153 に、PIEIFR7 を示し、表 1-173 に、その説明を示します。

概略表に戻ります。

割り込みグループ 7 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-153. PIEIFR7 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-173. PIEIFR7 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 7.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 7.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 7.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 7.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 7.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 7.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 7.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 7.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.17 PIEIER8 レジスタ (オフセット = 10h) [リセット = 0h]

図 1-154 に、PIEIER8 を示し、表 1-174 に、その説明を示します。

概略表に戻ります。

割り込みグループ 8 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-154. PIEIER8 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-174. PIEIER8 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 8.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 8.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 8.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 8.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 8.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 8.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 8.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 8.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.18 PIEIFR8 レジスタ (オフセット = 11h) [リセット = 0h]

図 1-155 に、PIEIFR8 を示し、表 1-175 に、その説明を示します。

概略表に戻ります。

割り込みグループ 8 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-155. PIEIFR8 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-175. PIEIFR8 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 8.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 8.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 8.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 8.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 8.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 8.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 8.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 8.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.19 PIEIER9 レジスタ (オフセット = 12h) [リセット = 0h]

図 1-156 に、PIEIER9 を示し、表 1-176 に、その説明を示します。

概略表に戻ります。

割り込みグループ 9 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-156. PIEIER9 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-176. PIEIER9 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 9.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 9.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 9.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 9.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 9.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 9.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 9.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 9.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.20 PIEIFR9 レジスタ (オフセット = 13h) [リセット = 0h]

図 1-157 に、PIEIFR9 を示し、表 1-177 に、その説明を示します。

概略表に戻ります。

割り込みグループ 9 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-157. PIEIFR9 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-177. PIEIFR9 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 9.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 9.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 9.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 9.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 9.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 9.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 9.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 9.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.21 PIEIER10 レジスタ (オフセット = 14h) [リセット = 0h]

図 1-158 に、PIEIER10 を示し、表 1-178 に、その説明を示します。

概略表に戻ります。

割り込みグループ 10 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-158. PIEIER10 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-178. PIEIER10 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 10.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 10.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 10.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 10.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 10.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 10.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 10.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 10.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.22 PIEIFR10 レジスタ (オフセット = 15h) [リセット = 0h]

図 1-159 に、PIEIFR10 を示し、表 1-179 に、その説明を示します。

概略表に戻ります。

割り込みグループ 10 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-159. PIEIFR10 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-179. PIEIFR10 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 10.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 10.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 10.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 10.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 10.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 10.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 10.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 10.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.23 PIEIER11 レジスタ (オフセット = 16h) [リセット = 0h]

図 1-160 に、PIEIER11 を示し、表 1-180 に、その説明を示します。

概略表に戻ります。

割り込みグループ 11 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-160. PIEIER11 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-180. PIEIER11 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 11.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 11.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 11.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 11.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 11.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 11.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 11.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 11.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.24 PIEIFR11 レジスタ (オフセット = 17h) [リセット = 0h]

図 1-161 に、PIEIFR11 を示し、表 1-181 に、その説明を示します。

概略表に戻ります。

割り込みグループ 11 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-161. PIEIFR11 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-181. PIEIFR11 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 11.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 11.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 11.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 11.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 11.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 11.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 11.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 11.1 のフラグ リセット・タイプ: SYSRSn

1.15.10.25 PIEIER12 レジスタ (オフセット = 18h) [リセット = 0h]

図 1-162 に、PIEIER12 を示し、表 1-182 に、その説明を示します。

概略表に戻ります。

割り込みグループ 12 イネーブル・レジスタ

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ます。これらのビットの動作は CPU 割り込みイネーブル・レジスタ (IER) のビットと非常によく似ています。

ビットを 1 に設定すると、対応する割り込みが CPU に伝搬されます。

ビットを 0 に設定すると、対応する割り込みが伝搬されなくなります。ペリフェラル割り込み信号は、ディセーブルの割り込みに対して PIEIFR ビットを引き続き設定できることに注意してください。

図 1-162. PIEIER12 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-182. PIEIER12 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 12.8 のイネーブル リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 12.7 のイネーブル リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 12.6 のイネーブル リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 12.5 のイネーブル リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 12.4 のイネーブル リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 12.3 のイネーブル リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 12.2 のイネーブル リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 12.1 のイネーブル リセット・タイプ: SYSRSn

1.15.10.26 PIEIFR12 レジスタ (オフセット = 19h) [リセット = 0h]

図 1-163 に、PIEIFR12 を示し、表 1-183 に、その説明を示します。

概略表に戻ります。

割り込みグループ 12 フラグ・レジスタ

これらのレジスタ・ビットは、グループ内の各割り込みが現在保留中であるかどうかを示します。これらのビットの動作は CPU 割り込みフラグ・レジスタ (IFR) のビットと非常によく似ています。

ペリフェラルが割り込みを送信すると、対応するビットが設定されます。このビットは割り込みが CPU に伝播されるとクリアされ、その時点で PIEACK が設定されます。

注: PIE IFR フラグを書き込むことでソフトウェア割り込みを作成できます。

IFR フラグは 0 を書き込むとクリアされます。したがって、割り込みを発生させようとして、意図せずに他の割り込みを中止してしまう可能性があります。テストでのみ使用するようにするか、アプリケーション・コードで使用する場合は細心の注意を払うことをお勧めします。PIE IFR レジスタの読み取りは安全です。

図 1-163. PIEIFR12 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
INTx8	INTx7	INTx6	INTx5	INTx4	INTx3	INTx2	INTx1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-183. PIEIFR12 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R/W	0h	予約済み
7	INTx8	R/W	0h	割り込み 12.8 のフラグ リセット・タイプ: SYSRSn
6	INTx7	R/W	0h	割り込み 12.7 のフラグ リセット・タイプ: SYSRSn
5	INTx6	R/W	0h	割り込み 12.6 のフラグ リセット・タイプ: SYSRSn
4	INTx5	R/W	0h	割り込み 12.5 のフラグ リセット・タイプ: SYSRSn
3	INTx4	R/W	0h	割り込み 12.4 のフラグ リセット・タイプ: SYSRSn
2	INTx3	R/W	0h	割り込み 12.3 のフラグ リセット・タイプ: SYSRSn
1	INTx2	R/W	0h	割り込み 12.2 のフラグ リセット・タイプ: SYSRSn
0	INTx1	R/W	0h	割り込み 12.1 のフラグ リセット・タイプ: SYSRSn

1.15.11 SYNC_SOC_REGS レジスタ

表 1-184 に、SYNC_SOC_REGS レジスタのメモリマップされたレジスタを示します。表 1-184 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-184. SYNC_SOC_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	SYNCSELECT	同期入力および出力選択レジスタ	EALLOW	表示
2h	ADCSOCOUTSELECT	外部 ADCSOC 選択レジスタ	EALLOW	表示
4h	SYNCSOCCLOCK	SYNCSEL および EXTADCSOC 選択ロック・レジスタ	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-185 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-185. SYNC_SOC_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
WSonce	W Sonce	書き込み 1 回のみセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.11.1 SYNCSELECT レジスタ (オフセット = 0h) [リセット = E03FFFFh]

図 1-164 に SYNCSELECT を示し、表 1-186 にその説明を示します。

概略表に戻ります。

同期入力および出力選択レジスタ

図 1-164. SYNCSELECT レジスタ

31	30	29	28	27	26	25	24
RESERVED			SYNCOUT				
R/W-7h			R/W-0h				
23	22	21	20	19	18	17	16
RESERVED						RESERVED	
R-0-0h						R/W-7h	
15	14	13	12	11	10	9	8
RESERVED	RESERVED			RESERVED			RESERVED
R/W-7h	R/W-7h			R/W-7h			R/W-7h
7	6	5	4	3	2	1	0
RESERVED		RESERVED			RESERVED		
R/W-7h		R/W-7h			R/W-7h		

表 1-186. SYNCSELECT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-29	RESERVED	R/W	7h	予約済み

表 1-186. SYNCSELECT レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
28-24	SYNCOUT	R/W	0h	<p>Syncout のソースを選択:</p> <p>00000: SYNCOUT ピンの駆動に EPWM1SYNCOUT を選択。 00001: SYNCOUT ピンの駆動に EPWM2SYNCOUT を選択。 00010: SYNCOUT ピンの駆動に EPWM3SYNCOUT を選択。 00011: SYNCOUT ピンの駆動に EPWM4SYNCOUT を選択。 00100: SYNCOUT ピンの駆動に EPWM5SYNCOUT を選択。 00101: SYNCOUT ピンの駆動に EPWM6SYNCOUT を選択。 00110: SYNCOUT ピンの駆動に EPWM7SYNCOUT を選択。 00111: 予約済み 01000: 予約済み 01001: 予約済み 01010: 予約済み 01011: 予約済み 01100: 予約済み 01101: 予約済み 01110: 予約済み 01111: 予約済み 10000: 予約済み 10001: 予約済み 10010: 予約済み 10011: 予約済み 10100: 予約済み 10101: 予約済み 10110: 予約済み 10111: 予約済み 11000: SYNCOUT ピンの駆動に ECAP1SYNCOUT を選択。 11001: SYNCOUT ピンの駆動に ECAP2SYNCOUT を選択。 11010: 予約済み 11011: 予約済み 11100: 予約済み 11101: 予約済み 11110: 予約済み 11111: 予約済み</p> <p>注: [1] 予約済みの領域を指定すると、デフォルトの 00 が選択されます。 リセット・タイプ: SYSRSn</p>
23-18	RESERVED	R-0	0h	予約済み
17-15	RESERVED	R/W	7h	予約済み
14-12	RESERVED	R/W	7h	予約済み
11-9	RESERVED	R/W	7h	予約済み
8-6	RESERVED	R/W	7h	予約済み
5-3	RESERVED	R/W	7h	予約済み
2-0	RESERVED	R/W	7h	予約済み

1.15.11.2 ADCSOCOUTSELECT レジスタ (オフセット = 2h) [リセット = 0h]

図 1-165 に ADCSOCOUTSELECT を示し、表 1-187 にその説明を示します。

概略表に戻ります。

外部 ADCSOC 選択レジスタ

図 1-165. ADCSOCOUTSELECT レジスタ

31	30	29	28	27	26	25	24
RESERVED				RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	PWM7SOCBEN	PWM6SOCBEN	PWM5SOCBEN	PWM4SOCBEN	PWM3SOCBEN	PWM2SOCBEN	PWM1SOCBEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED				RESERVED	RESERVED	RESERVED	RESERVED
R-0-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	PWM7SOCAEN	PWM6SOCAEN	PWM5SOCAEN	PWM4SOCAEN	PWM3SOCAEN	PWM2SOCAEN	PWM1SOCAEN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-187. ADCSOCOUTSELECT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-28	RESERVED	R-0	0h	予約済み
27	RESERVED	R/W	0h	予約済み
26	RESERVED	R/W	0h	予約済み
25	RESERVED	R/W	0h	予約済み
24	RESERVED	R/W	0h	予約済み
23	RESERVED	R/W	0h	予約済み
22	PWM7SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYRSn
21	PWM6SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYRSn
20	PWM5SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYRSn
19	PWM4SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYRSn
18	PWM3SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYRSn

表 1-187. ADCSOCOUTSELECT レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
17	PWM2SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYSRSn
16	PWM1SOCBEN	R/W	0h	ADCSOCBOn のソース選択: 0: 対応する EPWM SOCB 出力が選択されていません 1: 対応する EPWM SOCB 出力が選択されています リセット・タイプ: SYSRSn
15-12	RESERVED	R-0	0h	予約済み
11	RESERVED	R/W	0h	予約済み
10	RESERVED	R/W	0h	予約済み
9	RESERVED	R/W	0h	予約済み
8	RESERVED	R/W	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	PWM7SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn
5	PWM6SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn
4	PWM5SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn
3	PWM4SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn
2	PWM3SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn
1	PWM2SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn
0	PWM1SOCAEN	R/W	0h	ADCSOCAOn のソース選択: 0: 対応する EPWM SOCA 出力が選択されていません 1: 対応する EPWM SOCA 出力が選択されています リセット・タイプ: SYSRSn

1.15.11.3 SYNCSOCLOCK レジスタ (オフセット = 4h) [リセット = 0h]

図 1-166 に SYNCSOCLOCK を示し、表 1-188 にその説明を示します。

概略表に戻ります。

SYNCSEL および EXTADCSOC 選択ロック・レジスタ

図 1-166. SYNCSOCLOCK レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						ADCSOCOUTS ELECT	SYNCSELECT
R-0-0h						R/WOnce-0h	R/WOnce-0h

表 1-188. SYNCSOCLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R-0	0h	予約済み
1	ADCSOCOUTSELECT	R/WOnce	0h	ADCSOCOUTSELECT レジスタ・ロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 注: [1] このレジスタのいずれかのビットは、一度セットした後は、SYRSn によってのみクリアできます。このレジスタのいずれかのビットに 0 を書き込んでも、何も効果はありません。 [2] ロック・メカニズムは書き込みのみ適用されます。ロック保護機能を持つレジスタへの読み取りは常に許可されます リセット・タイプ: SYRSn
0	SYNCSELECT	R/WOnce	0h	SYNCSELECT レジスタ・ロック・ビット: 0: 対応するレジスタはロックされていません 1: 対応するレジスタはロックされています。 注: [1] このレジスタのいずれかのビットは、一度セットした後は、SYRSn によってのみクリアできます。このレジスタのいずれかのビットに 0 を書き込んでも、何も効果はありません。 [2] ロック・メカニズムは書き込みのみ適用されます。ロック保護機能を持つレジスタへの読み取りは常に許可されます リセット・タイプ: SYRSn

1.15.12 SYS_STATUS_REGS レジスタ

SYS_STATUS_REGS レジスタのメモリマップされたレジスタを、表 1-189 に示します。表 1-189 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-189. SYS_STATUS_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
10h	SYS_ERR_INT_FLG	システム内の複数の異なるエラーによる割り込みのステータス。		表示
12h	SYS_ERR_INT_CLR	SYS_ERR_INT_FLG クリア・レジスタ		表示
14h	SYS_ERR_INT_SET	SYS_ERR_INT_FLG 設定レジスタ	EALLOW	表示
16h	SYS_ERR_MASK	SYS_ERR_MASK レジスタ	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-190 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-190. SYS_STATUS_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み取りタイプ		
R	R	読み取り
R-0	R-0	読み取り 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W1S	書き込み 1 で設定
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ・グループの一部であるレジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ配列の値を参照します。

1.15.12.1 SYS_ERR_INT_FLG レジスタ (オフセット = 10h) [リセット = 0h]

図 1-167 に、SYS_ERR_INT_FLG を示し、表 1-191 に、その説明を示します。

概略表に戻ります。

システム内の複数の異なるエラーによる割り込みのステータス。

図 1-167. SYS_ERR_INT_FLG レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	EPG1_INT	RESERVED	RESERVED	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RAM_ACC_VIO L	RESERVED	CORRECTABL E_ERR	RESERVED	GINT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-191. SYS_ERR_INT_FLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-20	RESERVED	R	0h	予約済み
19	RESERVED	R	0h	予約済み
18	RESERVED	R	0h	予約済み
17	FPU_OFLOW	R	0h	0:FPU_OFLOW による割り込みは発生していません。 1:FPU_OFLOW による割り込みが発生しました。 リセット・タイプ:SYSRSn
16	FPU_UFLOW	R	0h	0:FPU_UFLOW による割り込みは発生していません。 1:FPU_UFLOW による割り込みが発生しました。 リセット・タイプ:SYSRSn
15	RESERVED	R	0h	予約済み
14	RESERVED	R	0h	予約済み
13	RESERVED	R	0h	予約済み
12	RESERVED	R	0h	予約済み
11	EPG1_INT	R	0h	0:EPG1_INT による割り込みは発生していません。 1:EPG1_INT による割り込みが発生しました。 リセット・タイプ:SYSRSn
10	RESERVED	R	0h	予約済み
9	RESERVED	R	0h	予約済み
8	RESERVED	R	0h	予約済み
7	RESERVED	R	0h	予約済み
6	RESERVED	R	0h	予約済み
5	RESERVED	R	0h	予約済み
4	RAM_ACC_VIOL	R	0h	0:設定保護ルールに違反しているマスタはありません。 1:1 つ以上のアクセス保護ルールに違反しているマスタ・アクセスが少なくとも 1 つあります。 リセット・タイプ:SYSRSn

表 1-191. SYS_ERR_INT_FLG レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	RESERVED	R	0h	予約済み
2	CORRECTABLE_ERR	R	0h	0: 検出された修正可能なエラーの数はフラッシュ / RAM の設定スレッショルドを超えていません。 1: 検出された修正可能なエラーの数がフラッシュ / RAM の設定スレッショルドを超えています。 リセット・タイプ: SYSRSn
1	RESERVED	R	0h	予約済み
0	GINT	R	0h	グローバル割り込みフラグ: 0: SYS_ERR_INT_FLG レジスタのいずれかのフラグが設定されると、SYS_ERR_INT のパルスが生成されて GINT フラグが設定されます。 1: GINT フラグがクリアされるまで、他の割り込みは発生しません。 リセット・タイプ: SYSRSn

1.15.12.2 SYS_ERR_INT_CLR レジスタ (オフセット = 12h) [リセット = 0h]

図 1-168 に、SYS_ERR_INT_CLR を示し、表 1-192 に、その説明を示します。

概略表に戻ります。

SYS_ERR_INT_FLG クリア・レジスタ

図 1-168. SYS_ERR_INT_CLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	EPG1_INT	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RAM_ACC_VIO L	RESERVED	CORRECTABL E_ERR	RESERVED	GINT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-192. SYS_ERR_INT_CLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-20	RESERVED	R	0h	予約済み
19	RESERVED	R-0/W1S	0h	予約済み
18	RESERVED	R-0/W1S	0h	予約済み
17	FPU_OFLOW	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの FPU_OFLOW フラグがクリアされます。 リセット・タイプ:SYSRSn
16	FPU_UFLOW	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの FPU_UFLOW フラグがクリアされます。 リセット・タイプ:SYSRSn
15	RESERVED	R-0/W1S	0h	予約済み
14	RESERVED	R-0/W1S	0h	予約済み
13	RESERVED	R-0/W1S	0h	予約済み
12	RESERVED	R-0/W1S	0h	予約済み
11	EPG1_INT	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの EPG1_INT フラグがクリアされます。 リセット・タイプ:SYSRSn
10	RESERVED	R-0/W1S	0h	予約済み
9	RESERVED	R-0/W1S	0h	予約済み
8	RESERVED	R-0/W1S	0h	予約済み
7	RESERVED	R-0/W1S	0h	予約済み
6	RESERVED	R-0/W1S	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み

表 1-192. SYS_ERR_INT_CLR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
4	RAM_ACC_VIOL	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの RAM_ACC_VIOL フラグがクリアされます。 リセット・タイプ:SYSRSn
3	RESERVED	R-0/W1S	0h	予約済み
2	CORRECTABLE_ERR	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの CORRECTABLE_ERR フラグがクリアされます。 リセット・タイプ:SYSRSn
1	RESERVED	R-0/W1S	0h	予約済み
0	GINT	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの GINT フラグがクリアされます。 リセット・タイプ:SYSRSn

1.15.12.3 SYS_ERR_INT_SET レジスタ (オフセット = 14h) [リセット = 0h]

図 1-169 に、SYS_ERR_INT_SET を示し、表 1-193 に、その説明を示します。

概略表に戻ります。

SYS_ERR_INT_FLG 設定レジスタ

図 1-169. SYS_ERR_INT_SET レジスタ

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	EPG1_INT	RESERVED	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RAM_ACC_VIO L	RESERVED	CORRECTABL E_ERR	RESERVED	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

表 1-193. SYS_ERR_INT_SET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-24	KEY	R-0/W	0h	このフィールドの値を 0xa5 にすると、このレジスタの他のビット・フィールドへの書き込みが有効になります。他の値を KEY フィールドに書き込むと、このレジスタの他のフィールドへの書き込みがブロックされます。 注: このレジスタのフィールドの更新は、KEY フィールドに正しい値が書き込まれている場合に、このレジスタへの 32 ビットの書き込みでのみ成功します。 リセット・タイプ: SYSRSn
23-20	RESERVED	R	0h	予約済み
19	RESERVED	R-0/W1S	0h	予約済み
18	RESERVED	R-0/W1S	0h	予約済み
17	FPU_OFLOW	R-0/W1S	0h	0: 効果はありません。 1: SYS_ERR_INT_FLG レジスタの FPU_OFLOW フラグが設定されます。 リセット・タイプ: SYSRSn
16	FPU_UFLOW	R-0/W1S	0h	0: 効果はありません。 1: SYS_ERR_INT_FLG レジスタの FPU_UFLOW フラグが設定されます。 リセット・タイプ: SYSRSn
15	RESERVED	R-0/W1S	0h	予約済み
14	RESERVED	R-0/W1S	0h	予約済み
13	RESERVED	R-0/W1S	0h	予約済み
12	RESERVED	R-0/W1S	0h	予約済み
11	EPG1_INT	R-0/W1S	0h	0: 効果はありません。 1: SYS_ERR_INT_FLG レジスタの EPG1_INT フラグが設定されます。 リセット・タイプ: SYSRSn
10	RESERVED	R-0/W1S	0h	予約済み
9	RESERVED	R-0/W1S	0h	予約済み
8	RESERVED	R-0/W1S	0h	予約済み

表 1-193. SYS_ERR_INT_SET レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
7	RESERVED	R-0/W1S	0h	予約済み
6	RESERVED	R-0/W1S	0h	予約済み
5	RESERVED	R-0/W1S	0h	予約済み
4	RAM_ACC_VIOL	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの RAM_ACC_VIOL フラグが設定されます。 リセット・タイプ:SYSRSn
3	RESERVED	R-0/W1S	0h	予約済み
2	CORRECTABLE_ERR	R-0/W1S	0h	0:効果はありません。 1:SYS_ERR_INT_FLG レジスタの CORRECTABLE_ERR フラグが設定されます。 リセット・タイプ:SYSRSn
1	RESERVED	R-0/W1S	0h	予約済み
0	RESERVED	R	0h	予約済み

1.15.12.4 SYS_ERR_MASK レジスタ (オフセット = 16h) [リセット = 0h]

図 1-170 に、SYS_ERR_MASK を示し、表 1-194 に、その説明を示します。

概略表に戻ります。

SYS_ERR_MASK レジスタ

図 1-170. SYS_ERR_MASK レジスタ

31	30	29	28	27	26	25	24
KEY							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				RESERVED	RESERVED	FPU_OFLOW	FPU_UFLOW
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	EPG1_INT	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RAM_ACC_VIO L	RESERVED	CORRECTABL E_ERR	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

表 1-194. SYS_ERR_MASK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-24	KEY	R/W	0h	このフィールドの値を 0xa5 にすると、このレジスタの他のビット・フィールドへの書き込みが有効になります。他の値を KEY フィールドに書き込むと、このレジスタの他のフィールドへの書き込みがブロックされます。 注: このレジスタのフィールドの更新は、KEY フィールドに正しい値が書き込まれている場合に、このレジスタへの 32 ビットの書き込みでのみ成功します。 リセット・タイプ: SYSRSn
23-20	RESERVED	R	0h	予約済み
19	RESERVED	R/W	0h	予約済み
18	RESERVED	R/W	0h	予約済み
17	FPU_OFLOW	R/W	0h	0: SYS_ERR_INT_FLG レジスタの FPU_OFLOW フラグがハードウェア・イベントで設定されます。 1: SYS_ERR_INT_FLG レジスタの FPU_OFLOW フラグがハードウェア・イベントで設定されません。 リセット・タイプ: SYSRSn
16	FPU_UFLOW	R/W	0h	0: SYS_ERR_INT_FLG レジスタの FPU_UFLOW フラグがハードウェア・イベントで設定されます。 1: SYS_ERR_INT_FLG レジスタの FPU_UFLOW フラグがハードウェア・イベントで設定されません。 リセット・タイプ: SYSRSn
15	RESERVED	R/W	0h	予約済み
14	RESERVED	R/W	0h	予約済み
13	RESERVED	R/W	0h	予約済み
12	RESERVED	R/W	0h	予約済み
11	EPG1_INT	R/W	0h	0: SYS_ERR_INT_FLG レジスタの EPG1_INT フラグがハードウェア・イベントで設定されます。 1: SYS_ERR_INT_FLG レジスタの EPG1_INT フラグがハードウェア・イベントで設定されません。 リセット・タイプ: SYSRSn

表 1-194. SYS_ERR_MASK レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
10	RESERVED	R/W	0h	予約済み
9	RESERVED	R/W	0h	予約済み
8	RESERVED	R/W	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5	RESERVED	R/W	0h	予約済み
4	RAM_ACC_VIOL	R/W	0h	0:SYS_ERR_INT_FLG レジスタの RAM_ACC_VIOL フラグがハードウェア・イベントで設定されます。 1:SYS_ERR_INT_FLG レジスタの RAM_ACC_VIOL フラグがハードウェア・イベントで設定されません。 リセット・タイプ:SYSRSn
3	RESERVED	R/W	0h	予約済み
2	CORRECTABLE_ERR	R/W	0h	0:SYS_ERR_INT_FLG レジスタの CORRECTABLE_ERR フラグがハードウェア・イベントで設定されます。 1:SYS_ERR_INT_FLG レジスタの CORRECTABLE_ERR フラグがハードウェア・イベントで設定されません。 リセット・タイプ:SYSRSn
1	RESERVED	R/W	0h	予約済み
0	RESERVED	R	0h	予約済み

1.15.13 TEST_ERROR_REGS レジスタ

表 1-195 に、TEST_ERROR_REGS のメモリマップされたレジスタを示します。表 1-195 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-195. TEST_ERROR_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	CPR_RAM_TEST_ERROR_STS	RAM テスト:エラー・ステータス・レジスタ		表示
2h	CPU_RAM_TEST_ERROR_STS_CLR	RAM テスト:エラー・ステータス・クリア・レジスタ		表示
4h	CPU_RAM_TEST_ERROR_ADDR	RAM テスト:エラー・アドレス・レジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-196 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-196. TEST_ERROR_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み取り 0 を返す
書き込みタイプ		
W1S	W 1S	書き込み 1 でセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.13.1 CPU_RAM_TEST_ERROR_STS レジスタ (オフセット = 0h) [リセット = 0h]

図 1-171 に CPU_RAM_TEST_ERROR_STS を示し、表 1-197 にその説明を示します。

概略表に戻ります。

RAM テスト: エラー・ステータス・レジスタ

図 1-171. CPU_RAM_TEST_ERROR_STS レジスタ

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

表 1-197. CPU_RAM_TEST_ERROR_STS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R	0h	予約済み
1	UNC_ERROR	R	0h	0: RAM/ROM テスト・モードで「訂正不可能エラー」が発生していないことを示します。 1: RAM/ROM テスト・モードで「訂正不可能エラー」が発生したことを示します。 リセット・タイプ: SYSRSn
0	COR_ERROR	R	0h	0: RAM/ROM テスト・モードで「訂正可能エラー」が発生していないことを示します。 1: RAM/ROM テスト・モードで「訂正可能エラー」が発生したことを示します。 リセット・タイプ: SYSRSn

1.15.13.2 CPU_RAM_TEST_ERROR_STS_CLR レジスタ (オフセット = 2h) [リセット = 0h]

図 1-172 に CPU_RAM_TEST_ERROR_STS_CLR を示し、表 1-198 にその説明を示します。

概略表に戻ります。

RAM テスト: エラー・ステータス・クリア・レジスタ

図 1-172. CPU_RAM_TEST_ERROR_STS_CLR レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						UNC_ERROR	COR_ERROR
R-0h						R-0/W1S-0h	R-0/W1S-0h

表 1-198. CPU_RAM_TEST_ERROR_STS_CLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-2	RESERVED	R	0h	予約済み
1	UNC_ERROR	R-0/W1S	0h	0: 影響なし。 1: CPU_RAM_TEST_ERROR_STS レジスタの対応するビットをクリアします。 リセット・タイプ: SYSRSn
0	COR_ERROR	R-0/W1S	0h	0: 影響なし。 1: CPU_RAM_TEST_ERROR_STS レジスタの対応するビットをクリアします。 リセット・タイプ: SYSRSn

1.15.13.3 CPU_RAM_TEST_ERROR_ADDR レジスタ (オフセット = 4h) [リセット = 0h]

図 1-173 に CPU_RAM_TEST_ERROR_ADDR を示し、表 1-199 にその説明を示します。

概略表に戻ります。

RAM テスト: エラー・アドレス・レジスタ

図 1-173. CPU_RAM_TEST_ERROR_ADDR レジスタ

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

表 1-199. CPU_RAM_TEST_ERROR_ADDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	ADDR	R	0h	RAM/ROM テスト・モードでエラーが検出された位置のアドレス。 リセット・タイプ: SYSRSn

1.15.14 UID_REGS レジスタ

UID_REGS レジスタのメモリマップされたレジスタを、表 1-200 に示します。表 1-200 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-200. UID_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	UID_PSRAND0	UID 疑似ランダム 160 ビット番号		表示
2h	UID_PSRAND1	UID 疑似ランダム 160 ビット番号		表示
4h	UID_PSRAND2	UID 疑似ランダム 160 ビット番号		表示
6h	UID_PSRAND3	UID 疑似ランダム 160 ビット番号		表示
8h	UID_PSRAND4	UID 疑似ランダム 160 ビット番号		表示
Ah	UID_UNIQUE0	UID の一意の 64 ビット番号		表示
Ch	UID_UNIQUE1	UID の一意の 64 ビット番号		表示
Eh	UID_CHECKSUM	UID チェックサム		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表しています。表 1-201 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-201. UID_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i, j, k, l, m, n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタが反復レジスタ・グループの一部であるレジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を参照します。

1.15.14.1 UID_PSRAND0 レジスタ (オフセット = 0h) [リセット = X]

図 1-174 に、UID_PSRAND0 を示し、表 1-202 に、その説明を示します。

概略表に戻ります。

UID 疑似ランダム 160 ビット番号

図 1-174. UID_PSRAND0 レジスタ

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

表 1-202. UID_PSRAND0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	RandomID	R	X	UID の疑似ランダム部分 リセット・タイプ: 該当なし

1.15.14.2 UID_PSRAND1 レジスタ (オフセット = 2h) [リセット = X]

図 1-175 に、UID_PSRAND1 を示し、表 1-203 に、その説明を示します。

概略表に戻ります。

UID 疑似ランダム 160 ビット番号

図 1-175. UID_PSRAND1 レジスタ

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

表 1-203. UID_PSRAND1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	RandomID	R	X	UID の疑似ランダム部分 リセット・タイプ: 該当なし

1.15.14.3 UID_PSRAND2 レジスタ (オフセット = 4h) [リセット = X]

図 1-176 に、UID_PSRAND2 を示し、表 1-204 に、その説明を示します。

概略表に戻ります。

UID 疑似ランダム 160 ビット番号

図 1-176. UID_PSRAND2 レジスタ

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

表 1-204. UID_PSRAND2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	RandomID	R	X	UID の疑似ランダム部分 リセット・タイプ: 該当なし

1.15.14.4 UID_PSRAND3 レジスタ (オフセット = 6h) [リセット = X]

図 1-177 に、UID_PSRAND3 を示し、表 1-205 に、その説明を示します。

概略表に戻ります。

UID 疑似ランダム 160 ビット番号

図 1-177. UID_PSRAND3 レジスタ

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

表 1-205. UID_PSRAND3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	RandomID	R	X	UID の疑似ランダム部分 リセット・タイプ: 該当なし

1.15.14.5 UID_PSRAND4 レジスタ (オフセット = 8h) [リセット = X]

図 1-178 に、UID_PSRAND4 を示し、表 1-206 に、その説明を示します。

概略表に戻ります。

UID 疑似ランダム 160 ビット番号

図 1-178. UID_PSRAND4 レジスタ

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

表 1-206. UID_PSRAND4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	RandomID	R	X	UID の疑似ランダム部分 リセット・タイプ: 該当なし

1.15.14.6 UID_UNIQUE0 レジスタ (オフセット = Ah) [リセット = X]

図 1-179 に、UID_UNIQUE0 を示し、表 1-207 に、その説明を示します。

概略表に戻ります。

UID の一意の 64 ビット番号

図 1-179. UID_UNIQUE0 レジスタ

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

表 1-207. UID_UNIQUE0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	UniqueID	R	X	UID の一意の部分。この識別子は、同じ PARTIDH を持つすべてのデバイスで一意になります。 リセット・タイプ: 該当なし

1.15.14.7 UID_UNIQUE1 レジスタ (オフセット = Ch) [リセット = X]

図 1-180 に、UID_UNIQUE1 を示し、表 1-208 に、その説明を示します。

概略表に戻ります。

UID の一意の 64 ビット番号

図 1-180. UID_UNIQUE1 レジスタ

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

表 1-208. UID_UNIQUE1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	UniqueID	R	X	UID の一意の部分。この識別子は、同じ PARTIDH を持つすべてのデバイスで一意になります。 リセット・タイプ: 該当なし

1.15.14.8 UID_CHECKSUM レジスタ (オフセット = Eh) [リセット = X]

図 1-181 に、UID_CHECKSUM を示し、表 1-209 に、その説明を示します。

概略表に戻ります。

UID_PSRAND レジスタと UID_UNIQUE レジスタのフレッチャー・チェックサム

図 1-181. UID_CHECKSUM レジスタ

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

表 1-209. UID_CHECKSUM レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	Checksum	R	X	UID_PSRANDx と UID_UNIQUE のフレッチャー・チェックサム リセット・タイプ: 該当なし

1.15.15 WD_REGS レジスタ

WD_REGS レジスタのメモリマップされたレジスタを、表 1-210 に示します。表 1-210 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-210. WD_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
22h	SCSR	システム制御およびステータス・レジスタ	EALLOW	表示
23h	WDCNTR	ウォッチドッグ・カウンタ・レジスタ	EALLOW	表示
25h	WDKEY	ウォッチドッグ・リセット・キー・レジスタ	EALLOW	表示
29h	WDCR	ウォッチドッグ制御レジスタ	EALLOW	表示
2Ah	WDWCR	ウォッチドッグ・ウィンドウ制御レジスタ	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-211 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-211. WD_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み取りタイプ		
R	R	読み取り
R-0	R -0	読み取り 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W 1S	書き込み 1 で設定
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ・グループの一部であるレジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ配列の値を参照します。

1.15.15.1 SCSR レジスタ (オフセット = 22h) [リセット = 5h]

図 1-182 に、SCSR を示し、表 1-212 に、その説明を示します。

概略表に戻ります。

システム制御およびステータス・レジスタ

図 1-182. SCSR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					WDINTS	WDENINT	WDOVERRIDE
R-0-0h					R-1h	R/W-0h	R/W1S-1h

表 1-212. SCSR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-3	RESERVED	R-0	0h	予約済み
2	WDINTS	R	1h	ウォッチドッグ割り込みステータス このビットは、アクティブ Low のウォッチドッグ割り込み信号の状態を示します (SYSCCLK と同期)。ウォッチドッグ割り込みを使用してシステムを低消費電力モードからウェークアップする場合、このビットが High のときにモードを移行する必要があります。同様に、ウォッチドッグをディセーブルにして再度イネーブルにする場合、安全に実行するにはこのビットを High にする必要があります。 リセット・タイプ: SYSRSn
1	WDENINT	R/W	0h	ウォッチドッグ割り込みイネーブル / リセット・ディセーブル このビットは、カウンタが満了したときにウォッチドッグで割り込み (WAKE/ WDOG) とリセット (WDRS) のどちらをトリガするかを決定します。 リセット・タイプ: SYSRSn
0	WDOVERRIDE	R/W1S	1h	ウォッチドッグ・イネーブル・ロック このビットに 1 を書き込むとビットがクリアされ、WDCR レジスタの WDDIS ビットがロックされます。このビットは、次のシステム・リセットまでこの状態のままになります。このビットを読み取ると、現在の値が返されます。このビットに 0 を書き込んでも効果はありません。 リセット・タイプ: SYSRSn

1.15.15.2 WDCNTR レジスタ (オフセット = 23h) [リセット = 0h]

図 1-183 に、WDCNTR を示し、表 1-213 に、その説明を示します。

概略表に戻ります。

ウォッチドッグ・カウンタ・レジスタ

図 1-183. WDCNTR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
WDCNTR							
R-0h							

表 1-213. WDCNTR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-0	WDCNTR	R	0h	ウォッチドッグ・カウンタ これらのビットには、ウォッチドッグ・カウンタの現在の値が格納されます。このカウンタは、WDCLK (INTOSC1) サイクルごとにインクリメントされます。カウンタがオーバーフローすると、SCSR レジスタの WDINTEN ビットの値に基づいて、割り込みまたはリセットのいずれかが生成されます。WDKEY レジスタに正しい値が書き込まれると、このカウンタはゼロにリセットされます。 リセット・タイプ: IORSn

1.15.15.3 WDKEY レジスタ (オフセット = 25h) [リセット = 0h]

図 1-184 に、WDKEY を示し、表 1-214 に、その説明を示します。

概略表に戻ります。

ウォッチドッグ・リセット・キー・レジスタ

図 1-184. WDKEY レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
WDKEY							
R/W-0h							

表 1-214. WDKEY レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-0	WDKEY	R/W	0h	ウォッチドッグ・カウンタ・リセット 0x55 に続けて 0xAA を書き込むと、ウォッチドッグ・カウンタがゼロにリセットされ、オーバーフローが回避されます。他の値を書き込んでも効果はありません。このレジスタを読み取ると、WDCR レジスタの値が返されます。 リセット・タイプ: IORSn

1.15.15.4 WDCR レジスタ (オフセット = 29h) [リセット = 0h]

図 1-185 に、WDCR を示し、表 1-215 に、その説明を示します。

概略表に戻ります。

ウォッチドッグ制御レジスタ

このメモリマップされたレジスタは、レジスタへの後続の書き込みとの間に 45 SYSCLK サイクルの遅延を必要とします。この遅延がないと、次の書き込みが失われる可能性があります。この遅延は、45 の NOP 命令を追加することで実現できます。

図 1-185. WDCR レジスタ

15	14	13	12	11	10	9	8
RESERVED				WDPRECLKDIV			
R-0-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	WDDIS	WDCHK		WDPS			
R/W1S-0h	R/W-0h	R-0/W-0h		R/W-0h			

表 1-215. WDCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-8	WDPRECLKDIV	R/W	0h	ウォッチドッグ・クロック・プリデバイダ これらのビットは、ウォッチドッグ・クロック・プリデバイダを決定します。これは、INTOSC1 とウォッチドッグ・カウンタ・クロック (WDCLK) の間の 2 つのデバイダのうち 1 つ目です。WDCLK の周波数は、次の式で与えられます。 $\text{PREDIVCLK} = \text{INTOSC1} / \text{プリデバイダ}$ $\text{WDCLK} = \text{PREDIVCLK} / \text{プリスケアラ}$ ウォッチドッグのリセットまたは割り込みのパルス長は 512 INTOSC1 サイクルであるため、カウンタ周期はそれよりも長くする必要があります。これを保証するには、プリスケアラとプリデバイダの積を 4 以上にする必要があります。デフォルトのプリデバイダの値は 512 です。 リセット・タイプ: IORSn
7	RESERVED	R/W1S	0h	予約済み
6	WDDIS	R/W	0h	ウォッチドッグ・ディセーブル このビットを設定すると、ウォッチドッグ・モジュールがディセーブルになります。このビットをクリアすると、ウォッチドッグ・モジュールがイネーブルになります。このビットは、SCSR レジスタの WDOVERRIDE ビットでロックできます。ウォッチドッグはリセット時にイネーブルになります。 リセット・タイプ: IORSn
5-3	WDCHK	R-0/W	0h	ウォッチドッグ・チェック・ビット このレジスタへの書き込み時に、これらのビットに値 101 (バイナリ) を書き込む必要があります。それ以外の値を書き込むと、ウォッチドッグのリセットまたは割り込みが直ちにトリガされます。 リセット・タイプ: IORSn
2-0	WDPS	R/W	0h	ウォッチドッグ・クロック・プリスケアラ これらのビットは、ウォッチドッグ・クロック・プリスケアラを決定します。これは、INTOSC1 とウォッチドッグ・カウンタ・クロック (WDCLK) の間の 2 つのデバイダのうち 2 つ目です。WDCLK の周波数は、次の式で与えられます。 $\text{PREDIVCLK} = \text{INTOSC1} / \text{プリデバイダ}$ $\text{WDCLK} = \text{PREDIVCLK} / \text{プリスケアラ}$ ウォッチドッグのリセットまたは割り込みのパルス長は 512 INTOSC1 サイクルであるため、カウンタ周期はそれよりも長くする必要があります。これを保証するには、プリスケアラとプリデバイダの積を 4 以上にする必要があります。デフォルトのプリスケアラの値は 1 です。 リセット・タイプ: IORSn

1.15.15.5 WDWCR レジスタ (オフセット = 2Ah) [リセット = 0h]

図 1-186 に、WDWCR を示し、表 1-216 に、その説明を示します。

概略表に戻ります。

ウォッチドッグ・ウィンドウ制御レジスタ

図 1-186. WDWCR レジスタ

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0-0h							R-0h
7	6	5	4	3	2	1	0
MIN							
R/W-0h							

表 1-216. WDWCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-9	RESERVED	R-0	0h	予約済み
8	RESERVED	R	0h	予約済み
7-0	MIN	R/W	0h	ウォッチドッグ・ウィンドウ・スレッシュホールド これらのビットは、ウォッチドッグ・カウンタ・リセット・ウィンドウの下限を指定します。カウンタ値がこのレジスタの値に達する前に WDKEY レジスタでカウンタがリセットされると、ウォッチドッグでリセットまたは割り込みが直ちにトリガされます。 リセット・タイプ: IORSn

1.15.16 XINT_REGS レジスタ

表 1-217 に、XINT_REGS レジスタのメモリマップされたレジスタを示します。表 1-217 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-217. XINT_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	XINT1CR	XINT1 構成レジスタ		表示
1h	XINT2CR	XINT2 構成レジスタ		表示
2h	XINT3CR	XINT3 構成レジスタ		表示
3h	XINT4CR	XINT4 構成レジスタ		表示
4h	XINT5CR	XINT5 構成レジスタ		表示
8h	XINT1CTR	XINT1 カウンタ・レジスタ		表示
9h	XINT2CTR	XINT2 カウンタ・レジスタ		表示
Ah	XINT3CTR	XINT3 カウンタ・レジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-218 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-218. XINT_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R -0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i,j,k,l,m,n		これらの変数をレジスタ名、オフセット、またはアドレスで使用する場合、反復レジスタ・グループの一部であるレジスタ配列の値を意味します。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用する場合、レジスタ配列の値を意味します。

1.15.16.1 XINT1CR レジスタ (オフセット = 0h) [リセット = 0h]

図 1-187 に XINT1CR を示し、表 1-219 にその説明を示します。

概略表に戻ります。

XINT1 構成レジスタ

図 1-187. XINT1CR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

表 1-219. XINT1CR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3-2	POLARITY	R/W	0h	00: 立ち下がりエッジ・トリガとして割り込みを選択 01: 立ち上がりエッジ・トリガとして割り込みを選択 10: 立ち下がりエッジ・トリガとして割り込みを選択 11: 立ち上がりまたは立ち下がりエッジ・トリガとして割り込みを選択 リセット・タイプ: SYSRSn
1	RESERVED	R-0	0h	予約済み
0	ENABLE	R/W	0h	0: 割り込みディセーブル 1: 割り込みイネーブル リセット・タイプ: SYSRSn

1.15.16.2 XINT2CR レジスタ (オフセット = 1h) [リセット = 0h]

図 1-188 に XINT2CR を示し、表 1-220 にその説明を示します。

概略表に戻ります。

XINT2 構成レジスタ

図 1-188. XINT2CR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

表 1-220. XINT2CR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3-2	POLARITY	R/W	0h	00: 立ち下がりエッジ・トリガとして割り込みを選択 01: 立ち上がりエッジ・トリガとして割り込みを選択 10: 立ち下がりエッジ・トリガとして割り込みを選択 11: 立ち上がりまたは立ち下がりエッジ・トリガとして割り込みを選択 リセット・タイプ: SYSRSn
1	RESERVED	R-0	0h	予約済み
0	ENABLE	R/W	0h	0: 割り込みディセーブル 1: 割り込みイネーブル リセット・タイプ: SYSRSn

1.15.16.3 XINT3CR レジスタ (オフセット = 2h) [リセット = 0h]

図 1-189 に XINT3CR を示し、表 1-221 にその説明を示します。

概略表に戻ります。

XINT3 構成レジスタ

図 1-189. XINT3CR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

表 1-221. XINT3CR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3-2	POLARITY	R/W	0h	00: 立ち下がりエッジ・トリガとして割り込みを選択 01: 立ち上がりエッジ・トリガとして割り込みを選択 10: 立ち下がりエッジ・トリガとして割り込みを選択 11: 立ち上がりまたは立ち下がりエッジ・トリガとして割り込みを選択 リセット・タイプ: SYSRSn
1	RESERVED	R-0	0h	予約済み
0	ENABLE	R/W	0h	0: 割り込みディセーブル 1: 割り込みイネーブル リセット・タイプ: SYSRSn

1.15.16.4 XINT4CR レジスタ (オフセット = 3h) [リセット = 0h]

図 1-190 に XINT4CR を示し、表 1-222 にその説明を示します。

概略表に戻ります。

XINT4 構成レジスタ

図 1-190. XINT4CR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

表 1-222. XINT4CR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3-2	POLARITY	R/W	0h	00: 立ち下がりエッジ・トリガとして割り込みを選択 01: 立ち上がりエッジ・トリガとして割り込みを選択 10: 立ち下がりエッジ・トリガとして割り込みを選択 11: 立ち上がりまたは立ち下がりエッジ・トリガとして割り込みを選択 リセット・タイプ: SYSRSn
1	RESERVED	R-0	0h	予約済み
0	ENABLE	R/W	0h	0: 割り込みディセーブル 1: 割り込みイネーブル リセット・タイプ: SYSRSn

1.15.16.5 XINT5CR レジスタ (オフセット = 4h) [リセット = 0h]

図 1-191 に XINT5CR を示し、表 1-223 にその説明を示します。

概略表に戻ります。

XINT5 構成レジスタ

図 1-191. XINT5CR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				POLARITY		RESERVED	ENABLE
R-0-0h				R/W-0h		R-0-0h	R/W-0h

表 1-223. XINT5CR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3-2	POLARITY	R/W	0h	00: 立ち下がりエッジ・トリガとして割り込みを選択 01: 立ち上がりエッジ・トリガとして割り込みを選択 10: 立ち下がりエッジ・トリガとして割り込みを選択 11: 立ち上がりまたは立ち下がりエッジ・トリガとして割り込みを選択 リセット・タイプ: SYSRSn
1	RESERVED	R-0	0h	予約済み
0	ENABLE	R/W	0h	0: 割り込みディセーブル 1: 割り込みイネーブル リセット・タイプ: SYSRSn

1.15.16.6 XINT1CTR レジスタ (オフセット = 8h) [リセット = 0h]

図 1-192 に XINT1CTR を示し、表 1-224 にその説明を示します。

概略表に戻ります。

XINT1 カウンタ・レジスタ

図 1-192. XINT1CTR レジスタ

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

表 1-224. XINT1CTR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	INTCTR	R	0h	これは、SYSCLKOUT レートでクロック供給される、フリーランニング 16 ビット・アップ・カウンタです。有効な割り込みエッジが検出されると、カウンタ値は 0x0000 にリセットされ、次の有効な割り込みエッジが検出されるまでカウントを続行します。カウンタは、対応する割り込み制御レジスタで選択した POLARITY (極性) エッジによってのみリセットされます。割り込みがディセーブルになると、カウンタは停止します。カウンタはフリーランニング・カウンタであり、最大値に達するとゼロに戻ります。カウンタは読み取り専用レジスタで、有効な割り込みエッジまたはリセットによってのみゼロにリセットできます。 リセット・タイプ: SYRSn

1.15.16.7 XINT2CTR レジスタ (オフセット = 9h) [リセット = 0h]

図 1-193 に XINT2CTR を示し、表 1-225 にその説明を示します。

概略表に戻ります。

XINT2 カウンタ・レジスタ

図 1-193. XINT2CTR レジスタ

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

表 1-225. XINT2CTR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	INTCTR	R	0h	これは、SYSCLKOUT レートでクロック供給される、フリーランニング 16 ビット・アップ・カウンタです。有効な割り込みエッジが検出されると、カウンタ値は 0x0000 にリセットされ、次の有効な割り込みエッジが検出されるまでカウントを続行します。カウンタは、対応する割り込み制御レジスタで選択した POLARITY (極性) エッジによってのみリセットされます。割り込みがディセーブルになると、カウンタは停止します。カウンタはフリーランニング・カウンタであり、最大値に達するとゼロに戻ります。カウンタは読み取り専用レジスタで、有効な割り込みエッジまたはリセットによってのみゼロにリセットできます。 リセット・タイプ: SYRSn

1.15.16.8 XINT3CTR レジスタ (オフセット= Ah) [リセット= 0h]

図 1-194 に XINT3CTR を示し、表 1-226 にその説明を示します。

概略表に戻ります。

XINT3 カウンタ・レジスタ

図 1-194. XINT3CTR レジスタ

15	14	13	12	11	10	9	8
INTCTR							
R-0h							
7	6	5	4	3	2	1	0
INTCTR							
R-0h							

表 1-226. XINT3CTR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	INTCTR	R	0h	これは、SYSCLKOUT レートでクロック供給される、フリーランニング 16 ビット・アップ・カウンタです。有効な割り込みエッジが検出されると、カウンタ値は 0x0000 にリセットされ、次の有効な割り込みエッジが検出されるまでカウントを続行します。カウンタは、対応する割り込み制御レジスタで選択した POLARITY (極性) エッジによってのみリセットされます。割り込みがディセーブルになると、カウンタは停止します。カウンタはフリーランニング・カウンタであり、最大値に達するとゼロに戻ります。カウンタは読み取り専用レジスタで、有効な割り込みエッジまたはリセットによってのみゼロにリセットできます。 リセット・タイプ: SYRSRn

1.15.17 Register to Driverlib Function Mapping

1.15.17.1 ASYSCTL レジスタと Driverlib 関数との対応

表 1-227. ASYSCTL レジスタと Driverlib 関数との対応

ファイル	Driverlib 関数
EXTROSCSR1	
-	
CONFIGLOCK	
-	
TSNSCTL	
asysctl.h	ASysCtl_enableTemperatureSensor
asysctl.h	ASysCtl_disableTemperatureSensor
ANARECTL	
adc.c	ADC_setVREF
adc.c	ADC_setOffsetTrim
asysctl.h	ASysCtl_setAnalogReferenceInternal
asysctl.h	ASysCtl_setAnalogReferenceExternal
asysctl.h	ASysCtl_setAnalogReference2P5
asysctl.h	ASysCtl_setAnalogReference1P65
VMONCTL	
-	
CMPPMXSEL	
asysctl.h	ASysCtl_selectCMPPMux
CMPLPMXSEL	
asysctl.h	ASysCtl_selectCMPLPMux
CMPHNMXSEL	
asysctl.h	ASysCtl_selectCMPHNMux
asysctl.h	ASysCtl_selectCMPHNMuxValue
CMPLNMXSEL	
asysctl.h	ASysCtl_selectCMPLNMux
asysctl.h	ASysCtl_selectCMPLNMuxValue
ADCDACLOOPBACK	
-	
CMPSSCTL	
asysctl.h	ASysCtl_enableCMPSSExternalDAC
asysctl.h	ASysCtl_disableCMPSSExternalDAC
LOCK	
asysctl.h	ASysCtl_lockTemperatureSensor
asysctl.h	ASysCtl_lockANAREF
asysctl.h	ASysCtl_lockVMON
asysctl.h	ASysCtl_lockCMPPMux
asysctl.h	ASysCtl_lockCMPLPMux
asysctl.h	ASysCtl_lockCMPHNMux
asysctl.h	ASysCtl_lockCMPLNMux
asysctl.h	ASysCtl_lockVREG
asysctl.h	ASysCtl_lockCMPSSCTL
AGPIOFILTER	

表 1-227. ASYSCTL レジスタと Driverlib 関数との対応 (continued)

ファイル	Driverlib 関数
-	
AGPIOCTRLA	
gpio.c	GPIO_setAnalogMode
AGPIOCTRLH	
-	

1.15.17.2 CPUTIMER レジスタと Driverlib 関数との対応
表 1-228. CPUTIMER レジスタと Driverlib 関数との対応

ファイル	Driverlib 機能
TIM	
cputimer.h	CPU_Timer_getTimerCount
PRD	
cputimer.h	CPUTimer_setPeriod
TCR	
cputimer.c	CPUTimer_setEmulationMode
cputimer.h	CPUTimer_clearOverflowFlag
cputimer.h	CPUTimer_disableInterrupt
cputimer.h	CPUTimer_enableInterrupt
cputimer.h	CPUTimer_reloadTimerCounter
cputimer.h	CPUTimer_stopTimer
cputimer.h	CPUTimer_resumeTimer
cputimer.h	CPUTimer_startTimer
cputimer.h	CPUTimer_getTimerOverflowStatus
TPR	
cputimer.h	CPUTimer_setPreScaler
TPRH	
cputimer.h	CPUTimer_setPreScaler

1.15.17.3 MEMCFG レジスタから Driverlib 関数への対応
表 1-229. MEMCFG レジスタから Driverlib 関数への対応

ファイル	Driverlib 関数
DXLOCK	
memcfg .c	MemCfg_lockConfig
memcfg .c	MemCfg_unlockConfig
DXCOMMIT	
memcfg .c	MemCfg_commitConfig
DXACCPROT0	
memcfg .c	MemCfg_setProtection
DXACCPROT1	
-	
DXTEST	
memcfg .c	MemCfg_setTestMode
DXINIT	
memcfg .c	MemCfg_initSections
memcfg .c	MemCfg_getInitStatus

表 1-229. MEMCFG レジスタから Driverlib 関数への対応 (continued)

ファイル	Driverlib 関数
DXINITDONE	
memcfg .c	MemCfg_getInitStatus
DXRAMTEST_LOCK	
memcfg .c	MemCfg_lockTestConfig
memcfg .c	MemCfg_unlockTestConfig
LSXLOCK	
memcfg .c	MemCfg_lockConfig
memcfg .c	MemCfg_unlockConfig
LSXCOMMIT	
memcfg .c	MemCfg_commitConfig
LSXACCPROT0	
memcfg .c	MemCfg_setProtection
LSXTEST	
memcfg .c	MemCfg_setTestMode
LSXINIT	
memcfg .c	MemCfg_initSections
memcfg .c	MemCfg_getInitStatus
LSXINITDONE	
memcfg .c	MemCfg_getInitStatus
LSXRAMTEST_LOCK	
memcfg .c	MemCfg_lockTestConfig
memcfg .c	MemCfg_unlockTestConfig
ROM_LOCK	
memcfg .c	MemCfg_lockTestConfig
memcfg .c	MemCfg_unlockTestConfig
ROM_TEST	
memcfg .c	MemCfg_setTestMode
ROM_FORCE_ERROR	
memcfg .c	MemCfg_forceMemError
MAVFLG	
memcfg.h	MemCfg_getViolationInterruptStatus
MAVSET	
memcfg.h	MemCfg_forceViolationInterrupt
MAVCLR	
memcfg.h	MemCfg_clearViolationInterruptStatus
MAVINTEN	
memcfg.h	MemCfg_enableViolationInterrupt
memcfg.h	MemCfg_disableViolationInterrupt
MCPUFAVADDR	
memcfg .c	MemCfg_getViolationAddress
MCPUWRVADDR	
memcfg .c	MemCfg_getViolationAddress
UCERRFLG	
memcfg.h	MemCfg_getUncorrErrorStatus
UCERRSET	

表 1-229. MEMCFG レジスタから Driverlib 関数への対応 (continued)

ファイル	Driverlib 関数
memcfg.h	MemCfg_forceUncorrErrorStatus
UCERRCLR	
memcfg.h	MemCfg_clearUncorrErrorStatus
UCCPUREADDR	
memcfg.c	MemCfg_getUncorrErrorAddress
FLUCERRSTATUS	
-	
FLCERRSTATUS	
-	
CERRFLG	
memcfg.h	MemCfg_getCorrErrorStatus
CERRSET	
memcfg.h	MemCfg_forceCorrErrorStatus
CERRCLR	
memcfg.h	MemCfg_clearCorrErrorStatus
CCPUREADDR	
memcfg.c	MemCfg_getCorrErrorAddress
CERRCNT	
memcfg.h	MemCfg_getCorrErrorCount
CERRTHRES	
memcfg.h	MemCfg_setCorrErrorThreshold
CEINTFLG	
memcfg.h	MemCfg_getCorrErrorInterruptStatus
CEINTCLR	
memcfg.h	MemCfg_clearCorrErrorInterruptStatus
CEINTSET	
memcfg.h	MemCfg_forceCorrErrorInterrupt
CEINTEN	
memcfg.h	MemCfg_enableCorrErrorInterrupt
memcfg.h	MemCfg_disableCorrErrorInterrupt
CPR_RAM_TEST_ERROR_STS	
memcfg.h	MemCfg_getDiagErrorStatus
memcfg.h	MemCfg_clearDiagErrorStatus
CPR_RAM_TEST_ERROR_STS_CLR	
memcfg.h	MemCfg_clearDiagErrorStatus
CPR_RAM_TEST_ERROR_ADDR	
memcfg.h	MemCfg_getDiagErrorAddress

1.15.17.4 NMI レジスタと Driverlib 関数

表 1-230. NMI レジスタと Driverlib 関数

ファイル	Driverlib 関数
CFG	
sysctl.h	SysCtl_enableNMIGlobalInterrupt
FLG	
sysctl.h	SysCtl_getNMIStatus

表 1-230. NMI レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
sysctl.h	SysCtl_getNMIFlagStatus
sysctl.h	SysCtl_isNMIFlagSet
sysctl.h	SysCtl_clearNMISStatus
sysctl.h	SysCtl_clearAllNMIFlags
sysctl.h	SysCtl_forceNMIFlags
FLGCLR	
sysctl.h	SysCtl_clearNMISStatus
sysctl.h	SysCtl_clearAllNMIFlags
FLGFRC	
sysctl.h	SysCtl_forceNMIFlags
WDCNT	
sysctl.h	SysCtl_getNMIWatchdogCounter
WDPRD	
sysctl.h	SysCtl_setNMIWatchdogPeriod
sysctl.h	SysCtl_getNMIWatchdogPeriod
SHDFLG	
sysctl.h	SysCtl_getNMIShadowFlagStatus
sysctl.h	SysCtl_isNMIShadowFlagSet
ERRORSTS	
sysctl.h	SysCtl_isErrorTriggered
sysctl.h	SysCtl_getErrorPinStatus
sysctl.h	SysCtl_forceError
sysctl.h	SysCtl_clearError
ERRORSTSCLR	
sysctl.h	SysCtl_clearError
ERRORSTSFRC	
sysctl.h	SysCtl_forceError
ERRORCTL	
sysctl.h	SysCtl_selectErrPinPolarity
ERRORLOCK	
sysctl.h	SysCtl_lockErrControl

1.15.17.5 PIE レジスタと Driverlib 関数

表 1-231. PIE レジスタと Driverlib 関数

ファイル	Driverlib 関数
CTRL	
interrupt.c	Interrupt_initModule
interrupt.h	Interrupt_defaultHandler
interrupt.h	Interrupt_enablePIE
interrupt.h	Interrupt_disablePIE
ACK	
interrupt.c	Interrupt_disable
interrupt.h	Interrupt_clearACKGroup
IER1	
interrupt.c	Interrupt_initModule

表 1-231. PIE レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
interrupt.c	Interrupt_enable
interrupt.c	Interrupt_disable
IFR1	
interrupt.c	Interrupt_initModule
IER2	
interrupt.c	Interrupt_initModule
IFR2	
interrupt.c	Interrupt_initModule
IER3	
interrupt.c	Interrupt_initModule
IFR3	
interrupt.c	Interrupt_initModule
IER4	
interrupt.c	Interrupt_initModule
IFR4	
interrupt.c	Interrupt_initModule
IER5	
interrupt.c	Interrupt_initModule
IFR5	
interrupt.c	Interrupt_initModule
IER6	
interrupt.c	Interrupt_initModule
IFR6	
interrupt.c	Interrupt_initModule
IER7	
interrupt.c	Interrupt_initModule
IFR7	
interrupt.c	Interrupt_initModule
IER8	
interrupt.c	Interrupt_initModule
IFR8	
interrupt.c	Interrupt_initModule
IER9	
interrupt.c	Interrupt_initModule
IFR9	
interrupt.c	Interrupt_initModule
IER10	
interrupt.c	Interrupt_initModule
IFR10	
interrupt.c	Interrupt_initModule
IER11	
interrupt.c	Interrupt_initModule
IFR11	
interrupt.c	Interrupt_initModule
IER12	

表 1-231. PIE レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
interrupt.c	Interrupt_initModule
IFR12	
interrupt.c	Interrupt_initModule

1.15.17.6 SYSCTL レジスタと Driverlib 関数
表 1-232. SYSCTL レジスタと Driverlib 関数

ファイル	Driverlib 関数
PARTIDL	
sysctl.c	SysCtl_getDeviceParametric
PARTIDH	
sysctl.c	SysCtl_getDeviceParametric
REVID	
sysctl.h	SysCtl_getDeviceRevision
TRIMERRSTS	
-	
SOFTPRES2	
-	SOFTPRES0 を参照
SOFTPRES3	
-	SOFTPRES0 を参照
SOFTPRES4	
-	SOFTPRES0 を参照
SOFTPRES7	
-	SOFTPRES0 を参照
SOFTPRES8	
-	SOFTPRES0 を参照
SOFTPRES9	
-	SOFTPRES0 を参照
SOFTPRES10	
-	SOFTPRES0 を参照
SOFTPRES13	
-	SOFTPRES0 を参照
SOFTPRES14	
-	SOFTPRES0 を参照
SOFTPRES21	
-	
SOFTPRES27	
-	
SOFTPRES28	
-	
TAP_STATUS	
-	
ECAPTYPE	
sysctl.c	SysCtl_configureType
sysctl.c	SysCtl_isConfigTypeLocked

表 1-232. SYSCTL レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
CLKCFGLOCK1	
sysctl.c	SysCtl_lockClkConfig
CLKSRCCTL1	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
sysctl.c	SysCtl_selectXTAL
sysctl.c	SysCtl_selectXTALSingleEnded
sysctl.c	SysCtl_selectOscSource
sysctl.h	SysCtl_enableWatchdogInHalt
sysctl.h	SysCtl_disableWatchdogInHalt
CLKSRCCTL2	
can.h	CAN_selectClockSource
CLKSRCCTL3	
sysctl.h	SysCtl_selectClockOutSource
SYSPLLCTL1	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
SYSPLLMULT	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
SYSPLLSTS	
sysctl.c	SysCtl_setClock
SYSCLKDIVSEL	
sysctl.c	SysCtl_getClock
sysctl.c	SysCtl_setClock
sysctl.h	SysCtl_setPLLSysClk
XCLKOUTDIVSEL	
sysctl.h	SysCtl_setXCik
LOSPCP	
sysctl.c	SysCtl_getLowSpeedClock
sysctl.h	SysCtl_setLowSpeedClock
MDCR	
sysctl.h	SysCtl_enableMCD
sysctl.h	SysCtl_disableMCD
sysctl.h	SysCtl_isMCDClockFailureDetected
sysctl.h	SysCtl_resetMCD
sysctl.h	SysCtl_connectMCDClockSource
sysctl.h	SysCtl_disconnectMCDClockSource
X1CNT	
sysctl.c	SysCtl_pollX1Counter
sysctl.h	SysCtl_getExternalOscCounterValue
sysctl.h	SysCtl_clearExternalOscCounterValue
XTALCR	
sysctl.c	SysCtl_setClock
sysctl.c	SysCtl_selectXTAL

表 1-232. SYSCTL レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
sysctl.c	SysCtl_selectXTALSingleEnded
sysctl.h	SysCtl_setExternalOscMode
sysctl.h	SysCtl_turnOnOsc
sysctl.h	SysCtl_turnOffOsc
XTALCR2	
sysctl.c	SysCtl_selectXTAL
CLKFAILCFG	
-	
CPUSYSLOCK1	
sysctl.c	SysCtl_lockSysConfig
CPUSYSLOCK2	
-	
PIEVERRADDR	
sysctl.h	SysCtl_getPIEErrAddr
PCLKCR0	
sysctl.h	SysCtl_enablePeripheral
sysctl.h	SysCtl_disablePeripheral
PCLKCR2	
-	PCLKCR0 を参照
PCLKCR3	
-	PCLKCR0 を参照
PCLKCR4	
-	PCLKCR0 を参照
PCLKCR7	
-	PCLKCR0 を参照
PCLKCR8	
-	PCLKCR0 を参照
PCLKCR9	
-	PCLKCR0 を参照
PCLKCR10	
-	PCLKCR0 を参照
PCLKCR13	
-	PCLKCR0 を参照
PCLKCR14	
-	PCLKCR0 を参照
PCLKCR21	
-	
PCLKCR27	
-	
SIMRESET	
sysctl.h	SysCtl_simulateReset
LPMCR	
sysctl.h	SysCtl_enterIdleMode
sysctl.h	SysCtl_enterStandbyMode
sysctl.h	SysCtl_enterHaltMode

表 1-232. SYSCTL レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
sysctl.h	SysCtl_setStandbyQualificationPeriod
sysctl.h	SysCtl_enableWatchdogStandbyWakeup
sysctl.h	SysCtl_disableWatchdogStandbyWakeup
GPIOLPMSEL0	
sysctl.h	SysCtl_enableLPMWakeupPin
sysctl.h	SysCtl_disableLPMWakeupPin
GPIOLPMSEL1	
sysctl.h	SysCtl_enableLPMWakeupPin
sysctl.h	SysCtl_disableLPMWakeupPin
TMR2CLKCTL	
cputimer.h	CPUTimer_selectClockSource
sysctl.h	SysCtl_setCputimer2Clk
RESCCLR	
sysctl.h	SysCtl_clearResetCause
sysctl.h	SysCtl_clearWatchdogResetStatus
RESC	
sysctl.h	SysCtl_getResetCause
sysctl.h	SysCtl_clearResetCause
sysctl.h	SysCtl_getWatchdogResetStatus
sysctl.h	SysCtl_clearWatchdogResetStatus
USER_REG1_SYSRSN	
sysctl.h	SysCtl_setUserRegister
sysctl.h	SysCtl_getUserRegister
USER_REG2_SYSRSN	
-	
USER_REG1_XRSN	
-	
USER_REG2_XRSN	
-	
USER_REG1_PORESETN	
-	
USER_REG2_PORESETN	
-	
USER_REG3_PORESETN	
-	
USER_REG4_PORESETN	
-	
SCSR	
sysctl.h	SysCtl_setWatchdogMode
sysctl.h	SysCtl_isWatchdogInterruptActive
sysctl.h	SysCtl_clearWatchdogOverride
WDCNTR	
sysctl.h	SysCtl_getWatchdogCounterValue
WDKEY	
sysctl.h	SysCtl_serviceWatchdog

表 1-232. SYSCTL レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
sysctl.h	SysCtl_enableWatchdogReset
sysctl.h	SysCtl_resetWatchdog
WDCR	
sysctl.h	SysCtl_resetDevice
sysctl.h	SysCtl_disableWatchdog
sysctl.h	SysCtl_enableWatchdog
sysctl.h	SysCtl_setWatchdogPredivider
sysctl.h	SysCtl_setWatchdogPrescaler
WDWCR	
sysctl.h	SysCtl_setWatchdogWindowValue
SYNCSELECT	
sysctl.h	SysCtl_setSyncOutputConfig
ADCSOCOUTSELECT	
sysctl.h	SysCtl_enableExtADCSOCSource
sysctl.h	SysCtl_disableExtADCSOCSource
SYNCSOCLOCK	
sysctl.h	SysCtl_lockExtADCSOCSelect
sysctl.h	SysCtl_lockSyncSelect
SYS_ERR_INT_FLG	
sysctl.h	SysCtl_getInterruptStatus
SYS_ERR_INT_CLR	
sysctl.h	SysCtl_clearInterruptStatus
SYS_ERR_INT_SET	
sysctl.h	SysCtl_setInterruptStatus
SYS_ERR_MASK	
sysctl.h	SysCtl_getInterruptStatusMask
sysctl.h	SysCtl_setInterruptStatusMask

1.15.17.7 XINT レジスタと Driverlib 関数

表 1-233. XINT レジスタと Driverlib 関数

ファイル	Driverlib 関数
1CR	
gpio.c	GPIO_setInterruptPin
gpio.h	GPIO_setInterruptType
gpio.h	GPIO_getInterruptType
gpio.h	GPIO_enableInterrupt
gpio.h	GPIO_disableInterrupt
gpio.h	GPIO_getInterruptCounter
2CR	
-	1CR を参照
3CR	
-	1CR を参照
4CR	
-	1CR を参照
5CR	

表 1-233. XINT レジスタと Driverlib 関数 (continued)

ファイル	Driverlib 関数
-	1CR を参照
1CTR	
gpio.h	GPIO_getInterruptCounter
2CTR	
-	
3CTR	
-	

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020 – REVISED MARCH 2023

Chapter 4 ROM コードとペリフェラル・ブート



この章では、ブート手順と使用可能なブート・モードについて説明します。また、メモリ・マップ、初期化、リセット処理、ステータス情報など、ROM コードのについても詳細に説明します。

1.1 はじめに.....	4
1.2 デバイス・ブート・シーケンス.....	5
1.3 デバイス・ブート・モード.....	5
1.4 デバイス・ブートの構成.....	6
1.5 デバイスのブート・フロー図.....	11
1.6 デバイスのリセットと例外処理.....	15
1.7 ブート ROM の説明.....	16
1.8 ブートローダ使用のアプリケーション・ノート.....	45

1.1 はじめに

この章では、CPU コアのブート読み取り専用メモリ (ROM) コードの機能について、ブートの手順も含めて説明することを目的としています。またこの章では、ブート ROM コードの機能と特徴について、および ROM メモリ・マップの内容の詳細についても取り扱います。リセットのたびにデバイスは、リセットの種類とブート構成に応じて、ROM 内のブート・シーケンスを実行します。このシーケンスにより、アプリケーション・コードを実行するための初期化が、デバイスで実行されます。またブート ROM には、CPU のためにアプリケーションを RAM にロードする際に使用するペリフェラル・ブートローダも含まれています。これらのブートローダは、安全性またはセキュリティのために無効化できます。

C28x CPU で利用可能なブート機能の詳細については、表 1-1 を参照してください。加えて表 1-2 では、デバイス上のさまざまな ROM のサイズを確認できます。

提供されているセキュリティ API の詳細については、セクション 1.7.9 を参照してください。

ソフトウェア・ライブラリで使用するために、ROM 内にはさまざまなテーブルが用意されています。詳細についてはセクション 1.7.6 を参照してください。

表 1-1. ブート・システムの概要

ブートの特徴	CPU
初期ブート・プロセス	デバイス・リセット。
ブート・モードの選択	GPIO
サポートされたブート・モード	フラッシュ・ブート セキュア・フラッシュ・ブート RAM ブート
サポートされるペリフェラル・ブート・ローダ	パラレル IO SCI / ウェイト CAN I2C SPI

表 1-2. ROM メモリ

ROM	CPU サイズ
セキュア、および非セキュアなブート ROM	64KB

1.1.1 ROM Related Collateral

Foundational Materials

- [Bootloading 101 \(Video\)](#)

Getting Started Materials

- [Secure BOOT On C2000 Device Application Report](#)

Expert Materials

- [C2000 Software Controlled Firmware Update Process Application Report](#)

1.2 デバイス・ブート・シーケンス

表 1-3 に、CPU コアがリセットされるたびにのブート ROM の一般的な手順を示します。

ブート中、ブート ROM コードによって、RAM 内のブート・ステータス・ロケーションが更新され、このプロセス中に実行されたアクションの詳細が記録されます。詳細については、[セクション 1.7.11](#) を参照してください。

表 1-3. デバイスのブート ROM シーケンス

ステップ	CPU アクション
1	リセット後、ウォッチドッグを初期化します
2	クロックを構成しフラッシュの電源を立ち上げます
3	ペリフェラル・トリミングとデバイス構成レジスタを OTP からロードします。
4	パワーオン・リセット (POR) 時に、すべての RAM を初期化します。
5	マスク不可能割り込み (NMI) 処理を有効化し、DCSM を初期化します。
6	デバイス・キャリブレーションを実行します。OTP 値をセットして指定されたペリフェラルのトリミングを行いません。
7	ブート・モードを決定するために GPIO ピンのポーリングが必要かどうかを判定し、必要な場合はブート・モード GPIO ピンを読み込み、実行するブート・モードを決定します。
8	ブート・モードとオプションに基づいて、適切なブート・シーケンスが実行されます。ブート・シーケンスのフローチャートについては、 セクション 1.5.1 を参照してください。

1.3 デバイス・ブート・モード

このセクションでは、デフォルトのブート・モードと、このデバイスでサポートされているすべてのカスタム・ブート・モードについて説明します。ブート ROM は、汎用入出力 (GPIO) ピンをブート・モード選択に使用して、ブート・モードの構成を決定します。

1.3.1 デフォルト・ブート・モード

表 1-4 に、デフォルトのブート・モード選択ピンで選択可能なブート・モード・オプションを示します。ユーザーは、ブートアップ・テーブルで選択可能なブート・モードと、使用するブート・モード選択ピン GPIO をカスタマイズするようにデバイスをプログラムすることができます。

表 1-4. デバイスのデフォルト・ブート・モード

ブート・モード	GPIO24 (デフォルト・ブート・モード選択ピン 1)	GPIO32 (デフォルト・ブート・モード選択ピン 0)
パラレル IO	0	0
SCI / ウェイト・ブート ⁽¹⁾	0	1
CAN	1	0
フラッシュ	1	1

(1) SCI オートパワー・ロック・プロセス中に SCI が「A」または「a」を待機し続ける限り、SCI ブート・モードが待機ブート・モードとして使用されます。

ブート・モードの詳細な機能については、[セクション 1.7.7.1](#) を参照してください。

ブート・モードの選択に使用する GPIO については、[セクション 1.7.8](#) を参照してください。

ブート構成の詳細については、[セクション 1.4](#) を参照してください。

注

サポートされているすべてのペリフェラル・ブート・モードは、ペリフェラル・モジュールの最初のインスタンス (SCIA、SPIA、I2CA、CANA など) を使用します。これらのブート・モードについて、この章で言及する場合、そのモードが実際に最初のモジュール・インスタンスであること指しています。たとえば SCI ブートと言えば、SCIA ポートでの SCI ブートを意味します。これは、他のペリフェラル・ブート・モードについても同様です。

1.3.2 カスタム・ブート・モード

ユーザー OTP 内でカスタム・ブート・テーブルがプログラムされていると、このカスタム・テーブルのエントリがブートに使用されます。ユーザーは、ユーザー OTP 内のロケーション `BOOTPIN_CONFIG` をプログラムすることで、エンド・システムの設計でのブート・モード選択ピンをカスタマイズできます。これにより、ユーザーは必要に応じて、0、1、2、3 のブート・モード選択ピンを使用できます。また、ユーザー OTP 内のロケーション `BOOTPDEF` にあるブート・モード定義表をプログラムしブート定義表をカスタマイズすることで、ブート元のロケーションを指定することもできます。表 1-5 に、各種ブート・モードのオプションを示します。

表 1-5. カスタム・ブート・モード

ブート・モード番号	ブート・モード
0	パラレル
1	SCI / ウェイト
2	CAN
3	フラッシュ
4	ウェイト
5	RAM
6	SPI
7	I2C
10	セキュア・フラッシュ

1.4 デバイス・ブートの構成

このセクションでは、利用可能なブート構成とその構成方法について詳しく説明します。このデバイスでは、0～3 本のブート・モード選択ピンと、1～8 個のブート・モードの構成がサポートされています。

デバイスをデフォルト設定からアプリケーションのカスタム設定に変更および構成するには、次のプロセスを使用します。

1. アプリケーションをブートできるようにするさまざまな方法をすべて決定します (たとえば、メイン・アプリケーション用のフラッシュ・ブートのプライマリ・ブート・オプション、ファームウェア更新用の CAN ブートのセカンダリ・ブート・オプション、デバッグ用の SCI ブートの 3 番目のブート・オプションなど)。
2. 必要なブート・モードの数に基づいて、ブート・モードの選択に必要なブート・モード選択ピン (BMSP) の数を決定します (たとえば、3 つのブート・モード・オプションから選択するには、2 つの BMSP が必要)
3. 必要な BMSP を物理的な GPIO ピンに割り当てます (たとえば、BMSP0 を GPIO10 に、BMSP1 を GPIO51 に、BMSP2 をデフォルト (ディセーブル) のままにするなど)。これらの構成を実行する詳細については、[セクション 1.4.1](#) を参照してください。
4. 決定したブート・モード定義を、BMSP のデコードされた値に相関付けるカスタム・ブート・テーブルのインデックスに割り当てます。たとえば、`BOOTDEF0` = フラッシュからブート、`BOOTDEF1` = CAN ブート、`BOOTDEF2` = SCI ブートに割り当て、その他のすべての `BOOTDEFx` はデフォルト / なしのままにします。カスタム・ブート・モード・テーブルの設定と構成の詳細については、[セクション 1.4.2](#) を参照してください。

さらに、[セクション 1.4.3](#) では、BMSP およびカスタム・ブート・テーブルの構成方法に関する使用例を提供しています。

1.4.1 ブート・モード・ピンの構成

このセクションでは、ユーザーが構成可能なデュアル・ゾーン・セキュリティ・モジュール (DCSM) OTP 内で、BOOTPIN-CONFIG のロケーション (表 1-6 を参照) をプログラムすることにより、ブート・モード選択ピンをユーザーがカスタマイズする方法について説明します。DCSM OTP 内のその領域は、Z1-OTP-BOOTPIN-CONFIG または Z2-OTP-BOOTPIN-CONFIG です。デバッグ時には、EMU-BOOTPIN-CONFIG が Z1-OTP-BOOTPIN-CONFIG / Z2-OTP-BOOTPIN-CONFIG をエミュレートする代用として機能するので、OTP に書き込まずにさまざまなブート・モードを検証するようにプログラムできます。このデバイスでは、必要に応じて、0、1、2、3 本のブート・モード選択ピンを使用するようにプログラムできます。

注

Z2-OTP-BOOTPIN-CONFIG を使用する場合には、このロケーションにプログラムされた構成が Z1-OTP-BOOTPIN-CONFIG の構成よりも優先されます。最初は Z1-OTP-BOOTPIN-CONFIG を使用し、OTP 構成に変更の必要性が生じた場合に、Z2-OTP-BOOTPIN-CONFIG に切り替えて使用することを推奨します。

表 1-6. BOOTPIN_CONFIG のビット・フィールド

ビット	名称	概要
31:24	キー	このレジスタのビットが有効であることをブート ROM コードに知らせるためには、この 8 ビットに 0x5A を書き込みます
23:16	ブート・モード選択ピン 2 (BMSP2)	BMSP0 の説明を参照してください。
15:8	ブート・モード選択ピン 1 (BMSP1)	BMSP0 の説明を参照してください。
7:0	ブート・モード選択ピン 0 (BMSP0)	ブート時に使用する GPIO ピンを設定します (最大 255)。 0x0 = GPIO0、0x01 = GPIO1、以下同様に続きます。 0xFF を書き込むと BMSP が無効化され、このピンはブート・モードの選択に使用されなくなります。

注

GPIO 224~253 はアナログ・ピンですが、ソフトウェアが GPIOHAMSEL レジスタ・ビットに書き込む場合は、これらのピンにデジタル入力を供給できます。

以下の GPIO を BMSP として使用することは**できません**。特定の BMSP 用に選択した場合、ブート ROM により BMSP0 および BMSP1 の工場出荷時デフォルト GPIO が自動的に選択されます。BMSP2 の工場出荷時デフォルトは 0xFF で、BMSP はディセーブルになります。

- GPIO 14 および GPIO 15 (どのパッケージでも利用不可)
- GPIO 25~GPIO 27 (どのパッケージでも利用不可)
- GPIO 30、GPIO 31、GPIO 34、GPIO 38 (どのパッケージでも利用不可)
- GPIO 42~GPIO 58 (どのパッケージでも利用不可)
- GPIO 62~GPIO 223 (どのパッケージでも利用不可)

表 1-7. スタンドアロンのブート・モード選択ピン・デコード

BOOTPIN_CONFIG キー	BMSP0	BMSP1	BMSP2	選択されるブート・モード
!= 0x5A	不定	不定	不定	工場出荷時にデフォルトで定義された BMSP でブートします。
= 0x5A	0xFF	0xFF	0xFF	ブート・テーブルで定義されたブート・モード 0 でブートします (すべての BMSP が無効)。
	有効な GPIO	0xFF	0xFF	この値の定義では BMSP0 でブートします (BMSP1 および BMSP2 は無効)。
	0xFF	有効な GPIO	0xFF	この値の定義では BMSP1 でブートします (BMSP0 および BMSP2 は無効)。
	0xFF	0xFF	有効な GPIO	この値の定義では BMSP2 でブートします (BMSP0 および BMSP1 は無効)。
	有効な GPIO	有効な GPIO	0xFF	この値の定義では BMSP0 と BMSP1 でブートします (BMSP2 は無効)。
	有効な GPIO	0xFF	有効な GPIO	この値の定義では BMSP0 と BMSP2 でブートします (BMSP1 は無効)。
	0xFF	有効な GPIO	有効な GPIO	この値の定義では BMSP1 と BMSP2 でブートします (BMSP0 は無効)。
	有効な GPIO	有効な GPIO	有効な GPIO	この値の定義では BMSP0、BMSP1、BMSP2 でブートします。
	無効な GPIO	有効な GPIO	有効な GPIO	BMSP0 は出荷時デフォルトの BMSP0 GPIO にリセットされます。この値の定義では BMSP0、BMSP1、BMSP2 でブートします。
	有効な GPIO	無効な GPIO	有効な GPIO	BMSP1 は、出荷時デフォルトの BMSP1 GPIO にリセットされます。この値の定義では BMSP0、BMSP1、BMSP2 でブートします。
有効な GPIO	有効な GPIO	無効な GPIO	BMSP2 は出荷時デフォルトの (無効) 状態にリセットされます。この値の定義では BMSP0、BMSP1 でブートされます。	

注

ブート・モードをデコードする際、BMSP0 がブート・テーブルでのインデックス値の最下位ビット、BMSP2 が最上位ビットになります。BMSP を無効化する場合は、BMSP2 の無効化から開始することをお勧めします。たとえば、BMSP2 のみを使用する (BMSP1 と BMSP0 は無効の) ケースでは、選択可能なブート・テーブル・インデックスは 0 と 4 のみです。BMSP0 のみを使用する場合、選択可能なブート・テーブル・インデックスは 0 と 1 です。

1.4.2 ブート・モード・テーブル・オプションの設定

このセクションでは、デバイスおよび関連するブート・オプションのブート定義表、BOOTDEF を構成する方法について説明します。ユーザーが構成可能な DCSM OTP の中に、Z1-OTP-BOOTDEF-LOW および Z1-OTP-BOOTDEF-HIGH として 64 ビットの領域があります。デバッグ時には、EMU-BOOTDEF-LOW および EMU-BOOTDEF-HIGH が、Z1-OTP-BOOTDEF-LOW および Z1-OTP-BOOTDEF-HIGH をエミュレートする代用として機能するので、OTP に書き込まずにさまざまなブート・モード・オプションを検証するようにプログラムできます。ブート定義表に対するカスタマイズの範囲は、使用されているブート・モード選択ピン (BMSP) の数によって異なります。たとえば、0 BMSP は 1 つのテーブル・エントリ、1 BMSP は 2 つのテーブル・エントリ、2 BMSP は 4 つのテーブル・エントリ、3 BMSP は 8 つのテーブル・エントリに対応します。BOOTPIN_CONFIG および BOOTDEF の値を設定する方法の例については、[セクション 1.4.3](#) を参照してください。

注

Z2-OTP-BOOTPIN-CONFIG が設定されている場合には、Z1-OTP-BOOTDEF-LOW および Z1-OTP-BOOTDEF-HIGH のロケーションの代わりに、Z2-OTP-BOOTDEF-LOW および Z2-OTP-BOOTDEF-HIGH のロケーションが使用されます。BOOTPIN_CONFIG 使用の詳細については、[セクション 1.4.1](#) を参照してください。

表 1-8. BOOTDEF のビット・フィールド

BOOTDEF 名	バイト位置	名称	概要
BOOT_DEF0	7:0	[3:0] BOOT_DEF0 モード	セクション 1.3.2 からのブート・モード番号を設定します。サポートされていないブート・モードを使用すると、デバイスは待機ブートかフラッシュからのブート・モードになります。
		[7:4] BOOT_DEF0 オプション	代替および追加のブート・オプションを設定します。これには、特定のブート・ペリフェラルに対する GPIO の変更や、別のフラッシュ・エントリ・ポイントの指定が含まれます。テーブルで設定できる有効な BOOTDEF 値については、 セクション 1.7.8 を参照してください。
BOOT_DEF1	15:8	BOOT_DEF1 モード / オプション	BOOT_DEF0 の説明を参照してください。
BOOT_DEF2	23:16	BOOT_DEF2 モード / オプション	
BOOT_DEF3	31:24	BOOT_DEF3 モード / オプション	
BOOT_DEF4	39:32	BOOT_DEF4 モード / オプション	
BOOT_DEF5	47:40	BOOT_DEF5 モード / オプション	
BOOT_DEF6	55:48	BOOT_DEF6 モード / オプション	
BOOT_DEF7	63:56	BOOT_DEF7 モード / オプション	

1.4.3 ブート・モードの使用例

このセクションでは、ブート・モード選択ピンとブート・モードの構成に関する、いくつかの使用例を示します。

1.4.3.1 ゼロ・ブート・モード選択ピン

この使用事例では、ブート・モード選択ピンを使用せず、デバイスが常にフラッシュからブートするアプリケーションにおけるシナリオを示します。

- OTP 内の BOOTPIN_CONFIG ロケーションを、次のようにプログラムします。
 - BOOTPIN_CONFIG.BMSP0 に 0xFF をセットします
 - BOOTPIN_CONFIG.BMSP1 に 0xFF をセットします
 - BOOTPIN_CONFIG.BMSP2 に 0xFF をセットします
 - ブート ROM でこれらのレジスタ・ビットを有効として扱い、カスタム・ブート・テーブルを使用するために、BOOTPIN_CONFIG.KEY に 0x5A を設定します。
- デバイスの BOOTDEF ロケーション・オプションをプログラムします。基本的に、これでデバイス固有のブート・モード・テーブルが設定されます。テーブルで設定できる有効な BOOTDEF 値については、[セクション 1.7.8](#) を参照してください。
 - フラッシュ・ブートのために、BOOTDEF.BOOTDEF0 に 0x03 を設定します (エントリ・アドレス・オプション 0)。これにより、フラッシュ・ブートをブート・テーブル・インデックス 0 に設定します。
 - 使用可能なフラッシュ・エントリ・ポイントについては、[セクション 1.7.2](#) を参照してください。

表 1-9. ゼロ・ブート・ピンのブート・テーブルの結果

ブート・モード・テーブル番号	ブート・モード
0	フラッシュ・ブート (0x03)

1.4.3.2 1 本のブート・モード選択ピン

この使用事例では、1 本のブート・モード選択ピンを使用して、アプリケーションが、フラッシュからブートするか CAN ブートを使用するかを選択するシナリオを示します。

- OTP 内の BOOTPIN_CONFIG ロケーションを、次のようにプログラムします。
 - GPIO のユーザー指定のために、BOOTPIN_CONFIG.BMSP0 を設定 (GPIO0 の場合は 0x0 など) にします
 - BOOTPIN_CONFIG.BMSP1 に 0xFF を設定します
 - BOOTPIN_CONFIG.BMSP2 に 0xFF を設定します
 - ブート ROM でこれらのレジスタ・ビットを有効として扱い、カスタム・ブート・テーブルを使用するには、BOOTPIN_CONFIG.KEY に 0x5A を設定します。
- デバイスの BOOTDEF ロケーションのオプションをプログラムします。基本的にはこれで、デバイス固有のブート・モード・テーブルが設定されます。テーブルで設定できる有効な BOOTDEF 値については、[セクション 1.7.8](#) を参照してください。
 - CAN ブートでは、BOOTDEF.BOOTDEF0 に 0x02 を設定します。これにより、CAN ブートがブート・テーブル・インデックス 0 に設定されます。
 - フラッシュからブートするには、BOOTDEF.BOOTDEF1 に 0x03 を設定します (エントリ・アドレス・オプション 0)。これにより、フラッシュ・ブートがブート・テーブル・インデックス 1 に設定されます。

表 1-10. 1 本のブート・ピンのブート・テーブルの結果

ブート・モード・テーブル番号	ブート・モード
0	CAN ブート (0x02)
1	フラッシュ・ブート (0x03)

1.4.3.3 3 本のブート・モード選択ピン

この使用事例では、3 本のブート・モード選択ピンを使用して、カスタム・ブート・テーブル内のさまざまなブート・モードを選択するアプリケーションのシナリオを示します。

- OTP 内の `BOOTPIN_CONFIG` ロケーションを、次のようにプログラムします。
 - GPIO のユーザー指定のために、`BOOTPIN_CONFIG.BMSP0` を設定 (GPIO0 の場合は `0x0` などに) します
 - GPIO のユーザー指定のために、`BOOTPIN_CONFIG.BMSP1` を設定 (GPIO1 の場合は `0x1` などに) します
 - GPIO のユーザー指定のために、`BOOTPIN_CONFIG.BMSP2` を設定 (GPIO2 の場合は `0x2` などに) します
 - ブート ROM でこれらのレジスタ・ビットを有効として扱い、カスタム・ブート・テーブルを使用するために、`BOOTPIN_CONFIG.KEY` に `0x5A` を設定します。
- デバイスの `BOOTDEF` ロケーション・オプションをプログラムします。基本的に、これでデバイス固有のブート・モード・テーブルが設定されます。テーブルで設定できる有効な `BOOTDEF` 値については、[セクション 1.7.8](#) を参照してください。
 - CAN ブートでは、`BOOTDEF.BOOTDEF0` に `0x02` を設定します。これにより、CAN ブートがブート・テーブル・インデックス 0 に設定されます。
 - フラッシュ・ブートのために、`BOOTDEF.BOOTDEF1` に `0x03` を設定します (エントリ・アドレス・オプション 0)。これにより、フラッシュ・ブートがブート・テーブル・インデックス 1 に設定されます。
 - 待機ブートのために、`BOOTDEF.BOOTDEF2` に `0x24` を設定します (代替オプション)。これにより、待機ブートをブート・テーブル・インデックス 2 に設定します。
 - SPI ブートのために、`BOOTDEF.BOOTDEF3` に `0x66` を設定します (代替 GPIO オプション 3)。これにより、SPI ブートをブート・テーブルのインデックス 3 に設定します。
 - フラッシュ・ブートのために、`BOOTDEF.BOOTDEF4` に `0x43` を設定します (エントリ・アドレス・オプション 2)。これにより、フラッシュ・ブートをブート・テーブル・インデックス 4 に設定します。

表 1-11. 3 本のブート・ピンのブート・テーブルの結果

ブート・モード・テーブル番号	ブート・モード
0	CAN ブート (0x02)
1	フラッシュ・ブート (0x03)
2	待機ブート - 代替 (0x24)
3	SPI - 代替 3 (0x66)
4	フラッシュ・ブート - 代替 2 (0x43)
5, 6, 7	この例では使用しません

1.5 デバイスのブート・フロー

このセクションでは、スタンドアロンおよびエミュレーション・ブート・フローの詳細なブート・フロー図を示します。

1.5.1 ブート・フロー

リセット時、CPU は [図 1-1](#) に示すブート・フローに従います。JTAG デバッガがデバイスに接続されているかどうかに応じて、CPU はエミュレーション・ブート・フローまたはスタンドアロン・ブート・フローのどちらかの後でブートを続行します。

注

BOR は POR と同様のフローに従います。

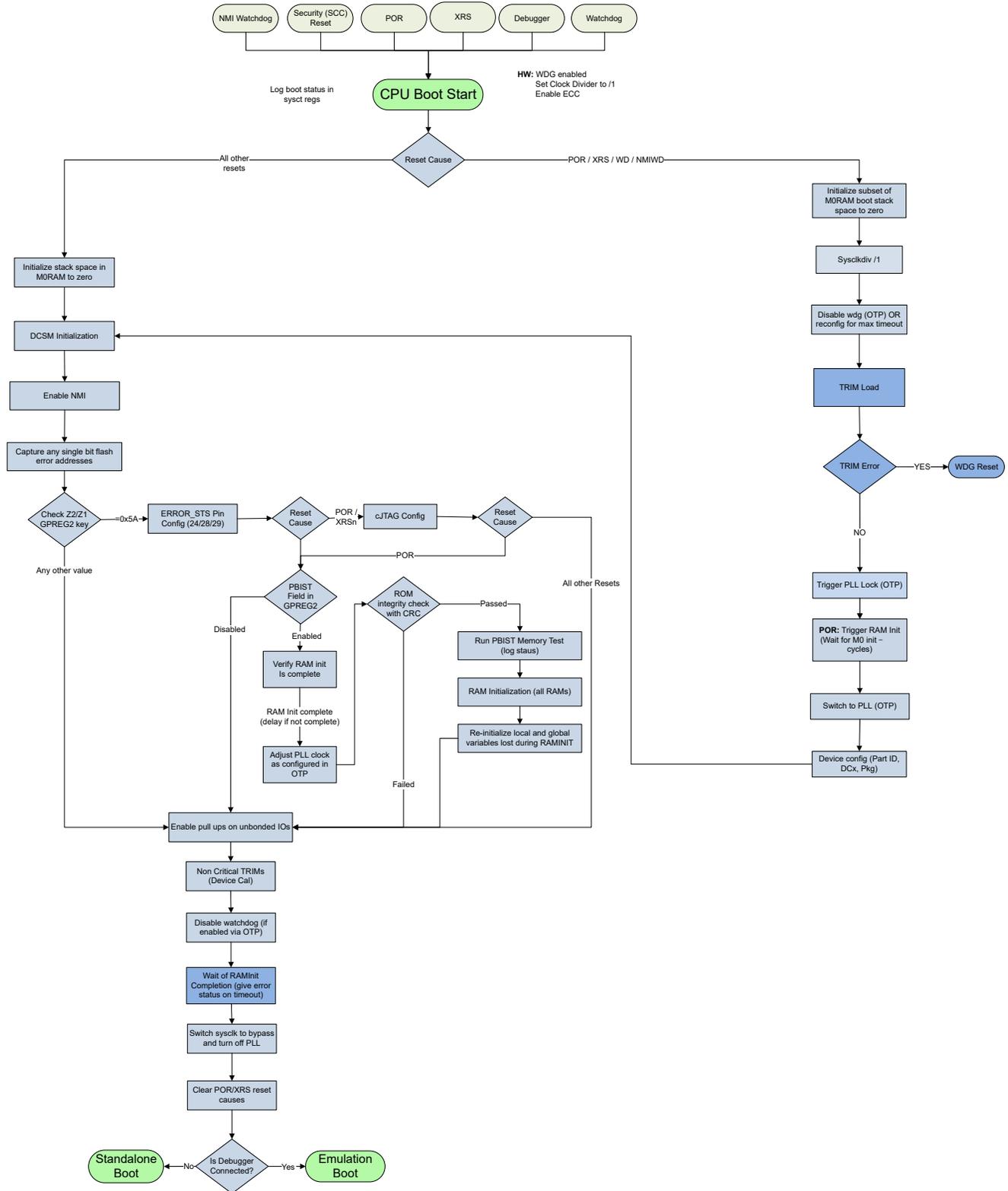


図 1-1. デバイス・ブート・フロー

1.5.2 エミュレーション・ブート・フロー

図 1-2 に、JTAG デバッガが接続されている場合のエミュレーション・ブート・フローを示します。

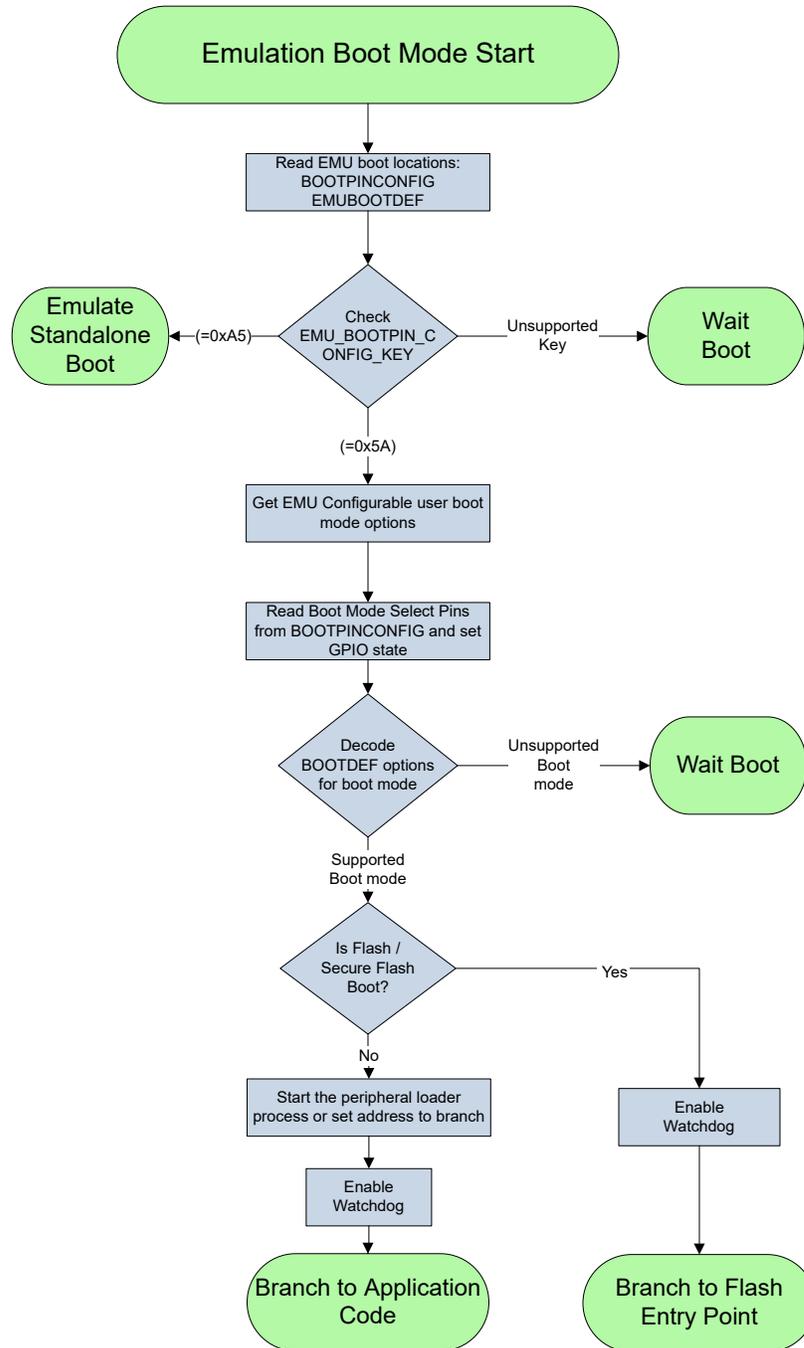


図 1-2. エミュレーション・ブート・フロー

1.5.3 スタンドアロン・ブート・フロー

図 1-3 に、JTAG デバッガがデバイスに接続されていない場合の、スタンドアロン・ブート・フローを示します。

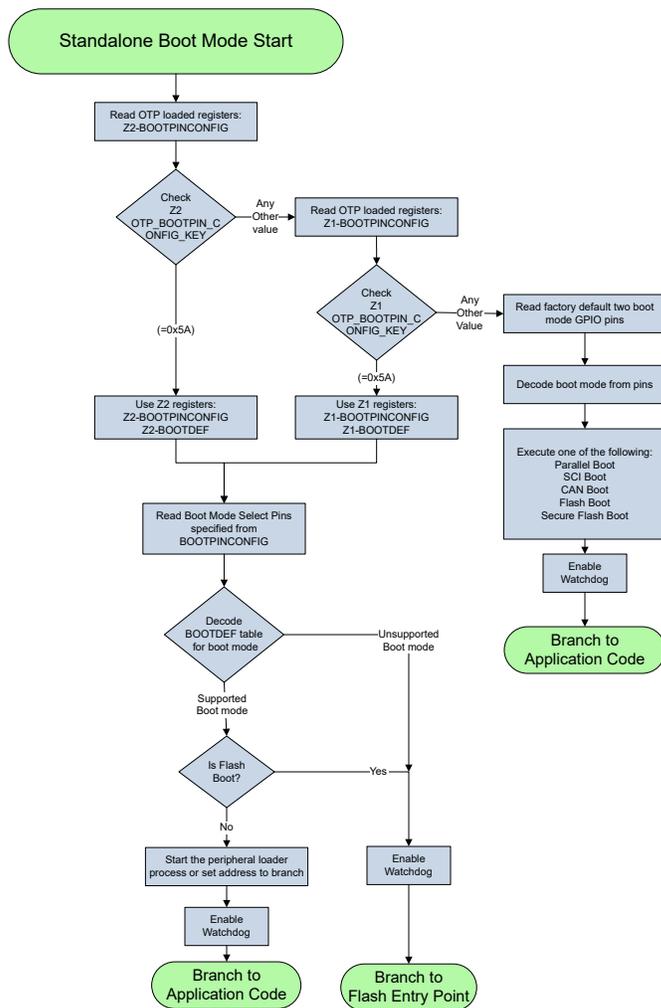


図 1-3. CPU スタンドアロン・ブート・フロー

1.6 デバイスのリセットと例外処理

1.6.1 リセットの原因と処理

表 1-12 に、特定のリセット原因に対し、各ブート ROM がリセット時に実行するアクションを示します。

表 1-12. ブート ROM のリセット原因とアクション

リセット・ソース	ブート ROM アクション
パワー・オン・リセット (POR)	1. クロック分周器を構成 2. フラッシュの電源立ち上げ 3. デバイスの構成と調整 4. RAM の初期化 5. デフォルトのブート・フローを続行
外部リセット (XRS) は以下を含みます。 <ul style="list-style-type: none"> • ウォッチドッグ・リセット • NMI ウォッチドッグ・リセット • SIMRESET XRS 	1. クロック分周器を構成 2. フラッシュの電源立ち上げ 3. デバイスの構成と調整 4. ブート・スタックの RAM をクリア 5. デフォルトのブート・フローを続行
セキュア・コピー・コード (SCC) リセット	1. ブート・スタックの RAM をクリア 2. デフォルトのブート・フローを続行
SIMRESET	1. ブート・スタックの RAM をクリア 2. デフォルトのブート・フローを続行
デバッグ・リセット	1. ブート・スタックの RAM をクリア 2. デフォルトのブート・フローを続行

1.6.2 例外と割り込みの処理

表 1-13 では、ブート中に例外が発生した場合に、ブート ROM が実行するアクションについて説明しています。ほとんどの場合、例外処理の考え方としては、エラーは記録して、引き続きブートを実行しアプリケーションに到達するというものです。

表 1-13. ブート ROM の例外とアクション

例外イベント・ソース	ブート ROM アクション	イベント記録
クロック・フェイル	NMI フラグをクリアしブートを続行	あり
RAM 修正不能エラー ROM パリティ・エラー	RAM の初期化を実行しデバイスをリセット	あり ⁽¹⁾
フラッシュ修正不能エラー	デバイスをリセット	あり
組み込みリアルタイム解析および診断 (ERAD) による NMI	NMI フラグをクリアしブートを続行	あり
RL NMI (CLB)	NMI フラグをクリアしブートを続行	あり
ソフトウェア NMI エラー (ソフトウェア・セルフ・テスト・エラー)	デバイスをリセット	なし
ITRAP 例外	不正な命令が実行されたメモリのアドレスを記録しデバイスをリセット	あり
サポートされていない PIE 割り込み	無視してブートを続行	なし

(1) RAM の修正不可能なエラーまたは ROM のパリティ・エラーにより、RAM に格納されているブート・ステータス情報はクリアされます。このエラーを修正するために、RAM の初期化が実行されるためです。ブート・ステータス情報が消去されているので、NMIWD リセットが発生し、すべての RAM が消去されたことがわかり、この例外を識別できます。

1.7 ブート ROM の説明

このセクションでは、デバイスのブート ROM の詳細について説明します。

1.7.1 ブート ROM 構成レジスタ

ブート ROM コードは、実行中に使用される複数のメモリ・アドレスおよびレジスタと関係しています。構成には、エミュレーション用とスタンドアロンのブート・フロー用という、2 種類のセットが存在します。RAM 内のエミュレーション用ロケーションでは OTP 構成をエミュレートします。ここには、必要な回数だけ書き込むことができます。ユーザーが構成可能な DCSM OTP のロケーションは、スタンドアロンのブート・フローで使用されデバイスの OTP をプログラムするため、書き込みできるのは 1 回だけです。『表 1-14』で、これらのロケーションの詳細を説明しています。BOOTPIN-CONFIG および BOOTCFG 用ビット・フィールドの構成の詳細については、[セクション 1.4.1](#) および [セクション 1.4.2](#) を参照してください。

さらにブート ROM では、DCSM のゾーン 1 およびゾーン 2 レジスタからのブート構成をサポートしています。ゾーン 2 での構成はゾーン 1 のものより優先されるため、ゾーン 1 の構成の使用が推奨されます。ゾーン 2 は二次的なオプションとして使用します。

表 1-14. ブート ROM レジスタ

ブート・フロー	レジスタ名	ブート ROM 名	レジスタ・アドレス	ユーザー OTP アドレス
エミュレーション	-	EMU-BOOTPIN-CONFIG	0x0000 0D00	-
	-	EMU-GPREG2	0x0000 0D02	-
	-	EMU-BOOTDEF-LOW	0x0000 0D04	-
	-	EMU-BOOTDEF-HIGH	0x0000 0D06	-
スタンドアロン (Z1 を使用)	Z1-GPREG1	Z1-OTP-BOOTPIN-CONFIG	0x0005 F008	0x0007 8008
	Z1-GPREG2	Z1-OTP-BOOT-GPREG2	0x0005 F00A	0x0007 800A
	Z1-GPREG3	Z1-OTP-BOOTDEF-LOW	0x0005 F00C	0x0007 800C
	Z1-GPREG4	Z1-OTP-BOOTDEF-HIGH	0x0005 F00E	0x0007 800E
スタンドアロン (Z2 を使用)	Z2-GPREG1	Z2-OTP-BOOTPIN-CONFIG	0x0005 F088	0x0007 8208
	Z2-GPREG2	Z2-OTP-BOOT-GPREG2	0x0005 F08A	0x0007 820A
	Z2-GPREG3	Z2-OTP-BOOTDEF-LOW	0x0005 F08C	0x0007 820C
	Z2-GPREG4	Z2-OTP-BOOTDEF-HIGH	0x0005 F08E	0x0007 820E

1.7.1.1 GPREG2 の使用と MPOST の構成

表 1-15 では、ユーザーが設定可能な OTP のロケーションである DCSM (Z1-OTP-BOOT-GPREG2 のビット・フィールド値が、ブート ROM でどのようにデコードされるかについて説明しています。

表 1-15. DCSM ゾーン GPREG2 ビット・フィールド

ビット	名称	概要	ブート ROM アクション
31:24	キー	このレジスタのビットが有効であることをブート ROM コードに知らせるためには、0x5A を書き込みます。	ユーザーが 0x5A に設定すると、ブート ROM はこのレジスタの値を使用します。他の値を設定した場合、ブート ROM はこのレジスタの値を無視します。
23:8	予約済み	予約済み	アクションなし
7:6	MPOST ⁽¹⁾	0x0 = PLL をディスエーブルにして MPOST を実行 (10MHz の内部発振器)	有効な値が設定されると、MPOST POR のメモリ・セルフ・テストがすべてのデバイス・メモリで実行されます
		0x1 = PLL を 95MHz でイネーブルにして MPOST を実行	
		0x2 = PLL を 47.5MHz でイネーブルにして MPOST を実行	
		0x3 = MPOST はディスエーブル	
5:4	ERROR_STS_PIN の構成	0X0 – GPIO24, MUX オプション 13	これにより、どの GPIO ピンで ERROR_PIN としての使用が想定されているかが示され、ブート ROM はそのピンにマルチプレクサを構成します。ERROR_STS ピンのマルチプレクサ構成はブート ROM によってロックされますが、コミットはされていません。
		0X1 – GPIO28, MUX オプション 13	
		0X2 – GPIO29, MUX オプション 13	
		0X3 – ERROR_STS 機能はディスエーブル (デフォルト)	
3:0	CJTAGNODEID	CJTAGNODEID[3:0]	ブート ROM はこの値を受け取り、CJTAGNODEID レジスタの下位 4 ビットをプログラムします。

(1) PLL を有効化して MPOST が動作するように構成し、PLL がロックに失敗した場合、MPOST の実行はすべてスキップされます。PLL が無効の状態では INTOSC2 を使用するように MPOST を構成している場合は、この動作は適用されません。

注

Z1-GPREG2 は Z1-GPREG1 と ECC を共有するため、ユーザーはこれら両方のロケーションを、一度にユーザー OTP でプログラムできます。

1.7.2 エントリ・ポイント

このセクションでは、各ブート・モードでのエントリ・ポイント・アドレスの詳細について説明します。これらのエントリ・ポイントは、選択したブート・モードに応じて、ブート終了時にどのアドレスに分岐するかをブート ROM に指示します。

表 1-16 に、フラッシュ・ブート・モードのエントリ・ポイント・アドレスの詳細を示します。

表 1-17 に、RAM ブート・モードのエントリ・ポイント・アドレスの詳細を示します。

表 1-18 に、セキュア・フラッシュ・ブート・モードのエントリ・ポイント・アドレスの詳細を示します。

表 1-16. フラッシュ・エントリ・ポイント・アドレス

オプション	BOOTDEFx 値	フラッシュ・セクタ	アドレス	サポートしているパッケージ
0	0x03	CPU バンク 0 セクタ 0	0x0008 0000	すべて
1	0x23	CPU バンク 0 セクタ 32	0x0008 8000	すべて
2	0x43	CPU バンク 0 セクタ 63 の最後	0x0008 FFF0	すべて
3	0x63	CPU バンク 0 セクタ 64	0x0009 0000	すべて
4	0x83	CPU バンク 0 セクタ 96 の最後	0x0009 8000	すべて

表 1-16. フラッシュ・エントリ・ポイント・アドレス (continued)

オプション	BOOTDEFx 値	フラッシュ・セクタ	アドレス	サポートしているパッケージ
5	0xA3	CPU バンク 0 セクタ 127 の最後	0x0009 FFF0	すべて

表 1-17. RAM エントリ・ポイント・アドレス

オプション	BOOTDEFx 値	RAM エントリ・ポイント	サポート対象のパッケージ
0	0x05	0x0000 0000	すべて

表 1-18. セキュア・フラッシュ・エントリ・ポイント・アドレス

オプション	BOOTDEFx 値	フラッシュ・セクタ	アドレス	サポートしているパッケージ
0 (デフォルト)	0x0A	CPU バンク 0 セクタ 0	0x0008 0000	すべて
1	0x2A	CPU バンク 0 セクタ 32	0x0008 8000	すべて
2	0x4A	CPU バンク 0 セクタ 63 の最後	0x0008 FFF0	すべて
3	0x6A	CPU バンク 0 セクタ 64	0x0009 0000	すべて
4	0x8A	CPU バンク 0 セクタ 96 の最後	0x0009 8000	すべて

1.7.3 待機ポイント

待機モードの CPU はブート ROM コード内でループ状態になり、ユーザー・アプリケーションのコードには分岐しません。デバイスは、手動での設定によるか、ブートアップ時に何らかの問題が発生した場合に、ウェイト・ブート・モードに移行します。デバッガを使用する場合は、JTAG での問題が発生しないようにするために、待機ブート・モードの使用を推奨します。待機ブート中は、デバッグ用に ESTOP が提供されます。

表 1-19. 待機ブート・オプション

オプション	BOOTDEFx 値	ウォッチドッグのステータス	サポート対象のパッケージ
0 (デフォルト)	0x04	イネーブル	すべて
1	0x24	ディセーブル	すべて

ブート ROM 実行中のシチュエーションによっては、CPU がコード内の待ちループに入る可能性があります。この状態が発生する理由は、さまざまです。CPU がこれらの状況に置かれた場合に、CPU の PC レジスタの値が落ち込むアドレス範囲の詳細を、表 1-20 に示します。

待機ブート・モードに移行するためのアクションを以下に示します。

- 待機ブートは、ユーザーがブート・モードとして設定します。
- ブート・モードが認識不能で、デバッガがデバイスに接続されています。
- エミュレーションの BOOTPIN_CONFIG キーが、0xA5 または 0x5A と等しくありません。
- エミュレーション・ブート中にエラーが発生し、有効なブート・モードとして認識されない値でブート・モード・ピンがデコードされます。

表 1-20. 待機ポイント・アドレス

アドレス範囲	概要
0x003F D4BC – 0x003F D4C2	待機ブート・モード内
0x003F E047 – 0x003F E04F	SCI ブートでオートボー・ロックを待機中
0x003F FE4C – 0x003F FED8	NMI ハンドラ内
0x003F FED9 – 0x003F FF0D	ITRAP の ISR 内
0x003F D611 - 0x003F D614 0x003F D4FB - 0x003F D512	パラレル・ブートで制御信号を待機中

1.7.4 セキュア・フラッシュ・ブート

セキュア・フラッシュ・ブート・モードは、ブート・フローがフラッシュ内で設定されたメモリ・アドレスに分岐するという点でフラッシュ・ブート・モードと類似しています。ただしこの分岐は、フラッシュ・メモリの内容が認証された後にのみ実行されます。フラッシュ認証では、暗号ベースのメッセージ認証プロトコル (CMAC) を使用して、設定されたフラッシュ・エントリ・ポイント・アドレスから 16KB 分のフラッシュを認証します。CMAC の計算には、CPU ユーザー OTP ゾーン 1 のヘッダー OTP CMACKEY ビット・フィールドで、ユーザー定義の 128 ビット・キーをプログラムする必要があります。さらに、16KB のフラッシュ・メモリ範囲に基づいてゴールデン CMAC タグを計算し、このタグをユーザー・コードとともに、フラッシュ内のハードコードされたアドレスに保存する必要があります。セキュア・フラッシュ・ブート中、計算された CMAC タグはフラッシュ内のユーザー・ゴールデン CMAC タグと比較されて、CMAC 認証の成功/失敗ステータスが判定されます。認証が成功すると、ブート・フローは継続され、アプリケーションの実行を開始するためにフラッシュに分岐されます。認証が失敗した場合、デバイスはリセットされます。

使用可能なセキュア・フラッシュ・ブート・エントリ・アドレスのオプションについては、[セクション 1.7.2](#) を参照してください。

CPU 用のセキュア・フラッシュ・ゴールデン CMAC タグ生成については、『[TMS320C28x アセンブリ言語ツール・ユーザー・ガイド](#)』にある、『[TMS320F2838x デバイスでのセキュア・フラッシュ・ブートの使用](#)』セクションを参照してください。

注

CMAC ゴールデン・シグネチャと CMAC キーは、どちらも 32 ビット・セクションの MSB に格納されます。各 32 ビット・セクションは LSB に格納されます。

キー: 2B7E1516 28AED2A6 ABF71588 09CF4F3C

(MSB は 2B、LSB は 3C)

CMACKEY0 = 0x2B7E1516

CMACKEY1 = 0x28AED2A6

CMACKEY2 = 0xABF71588

CMACKEY3 = 0x09CF4F3C

注

構成されたフラッシュ・エントリ・ポイントを包含するフラッシュ・セクタと、フラッシュの最初の 16KB は、セキュア・フラッシュ・ブートでのコア設定のために、ゾーン 1 に適切に割り当てられる必要があります。

セキュア・フラッシュ・ブートを使用する場合は、デバイスの JTAGLOCK を使用することを推奨します。

表 1-21. セキュア・フラッシュ・ブートの詳細

詳細	ロケーション・アドレス
CMAC シグネチャ・アドレス	フラッシュ・エントリ・ポイント・アドレス + 0x2
CMAC キー・アドレス (128 ビット・キー)	DCSM Z1 OTP CMACKEY0 / 1 / 2 / 3
フラッシュ・エントリ・ポイント (バンク 0、セクタ 0)	0x0008 0000
フラッシュ・エントリ・ポイント (バンク 0、セクタ 32)	0x0008 8000
フラッシュ・エントリ・ポイント (バンク 0、セクター 63 の終わり)	0x0008 FFF0
フラッシュ・エントリ・ポイント (バンク 0、セクタ 64)	0x0009 0000
フラッシュ・エントリ・ポイント (バンク 0、セクタ 96)	0x0009 8000
CMAC 計算のアドレス範囲	開始: フラッシュ・エントリ・ポイント・アドレス 終了: フラッシュ・エントリ・ポイント・アドレス + 16KB

表 1-22. セキュア・フラッシュ・タグとキーの詳細

名称	アドレス	詳細
CMA ゴールデン・タグ (128 ビット)	CPU: フラッシュ・エン트리・ポイント・アドレス + 0x2	フラッシュ内に配置され、エン트리・ポイント・アドレスから 2 ワードオフセット しています (CPU)。 CMAC 計算の実行時、メモリ内のゴールデン・タグのロケーションは、すべ て 0xF とみなされます。CPU でのリンカ構成の例については、 Example 1-1 を参照してください。 低いメモリ位置にタグの MSW が含まれ、高いメモリ位置に LSW が含ま れます。 例 (CPU 上): タグ = 0x00112233 44556677 8899AABB CCDDEEFF アドレス 0x0 = 0x00112233 アドレス 0x2 = 0x44556677 アドレス 0x4 = 0x8899AABB アドレス 0x6 = 0xCCDDEEFF
CMAC 128 ビット・キー	0x0007 8018	CPU ゾーン 1 のユーザー・ヘッダー OTP に配置 (CMACKEY0、CMACKEY1、CMACKEY2、CMACKEY3) キーの MSW は CMACKEY0 に格納され、LSW は CMACKEY3 に格 納されます。 例: キー = 0x00112233 44556677 8899AABB CCDDEEFF CMACKEY0 = 0x00112233 CMACKEY1 = 0x44556677 CMACKEY2 = 0x8899AABB CMACKEY3 = 0xCCDDEEFF

表 1-23. セキュア・フラッシュ認証失敗時のアクション

CPU	認証失敗時のアクション
C28x CPU	1. エミュレーションのみ- デバッガ停止 (ESTOP) 2. 無限ループで (WD リセットによるデバイス・リセットを) 待機

表 1-24. すべての CPU でのセキュア・フラッシュの推奨フロー

ステップ	アクション
1	CPU をセキュア・フラッシュ・ブート
2	設定されたエン트리・ポイントから最初の 16KB より後のフラッシュは、フラッシュですでに認証済みの 16KB 内に ある任意のアドレスに埋め込まれた、別の CMAC ゴールデン・タグを使用して認証を受けることができます。

CMAC 計算と認証用の API は、ROM の一部として提供されています。[セクション 1.7.9](#) に詳細を示します。

Example 1-1. セキュア・フラッシュ CPU1 リンカ・ファイルの例

```
MEMORY
{
  /* Code Start branch to _c_int00 */
  BEGIN : origin = 0x80000, length = 0x0002
  /* User calculated golden CMAC tag for Flash Sector 0 */
  GOLDEN_CMACE_TAG : origin = 0x80002, length = 0x0008
  /* Flash Sector 0 containing application code */
  FLASH_SECTOR_0 : origin = 0x8000A, length = 0x1FF6
  .
  .
  .
}
```

1.7.5 メモリ・マップ

1.7.5.1 ブート ROM メモリ・マップ

表 1-25 では、セキュアと非セキュアな ROM を含む ROM メモリ・マップの詳細を説明しています。

表 1-25. ブート ROM メモリ・マップ

メモリ	送信元アドレス	長さ (ワード)
ROM 署名	0x003F 9800	0x0002
バージョン	0x003F 9804	0x0004
IQMath テーブル	0x003F 9808	0x1674
FPU32 高速テーブル	0x003F AE7C	0x081A
FPU32 FFT 係数表	0x003F B696	0x0DF8
SW CRC テーブル	0x003F C48E	0x0200
ブート・コード	0x003F C68E	0x2372
AES テーブル	0x003F EA00	0x1400
割り込みハンドラ	0x003F FE00	0x010E
CRC テーブル	0x003F FFB6	0x0008
ベクタ・テーブル	0x003F FFBE	0x0042

1.7.5.2 予約済み RAM メモリ・マップ

表 1-26 に、ブート ROM で使用するために予約されている、RAM のメモリ使用の詳細を示します。これらのメモリ・セクションは、ユーザー・アプリケーションで予約できます。

表 1-26. 予約済み RAM メモリ・マップ

メモリ	概要	送信元アドレス	長さ (ワード)
RAM	ブート・ステータス、ブート・モード、MPOST ステータス、ブート・スタック	0x0000 0002	0x0126

1.7.6 ROM テーブル

表 1-27 では、用意された ROM 関数とテーブルを使用するためにアプリケーションに統合する、ブート ROM シンボル・ライブラリの詳細を説明します。

表 1-27. ROM シンボル・テーブル

ROM シンボル	ライブラリ名	ロケーション
ROM ブートローダと関数	F280013xCPU_BootROM_Symbols	C2000Ware の /libraries/boot_rom にあります
FPU32 テーブル	F280013xCPU_BootROM_Symbols	
IQmath	F280013xCPU_IQMathROM_Symbols	

1.7.7 ブート・モードとローダ

このデバイスでサポートされているブート・モードとブートローダの詳細については、このセクションを参照してください。

1.7.7.1 ブート・モード

このセクションでは、ペリフェラル・ブート・ローダを使用しないブート・モードの詳細について説明します。表 1-28 では、ペリフェラル・ブート・ローダを使用しないブート・モードの詳細について説明しています。

表 1-28. 使用可能なブート・モード

ブート・モード	CPU サポート
フラッシュ・ブート	C28x CPU
RAM ブート	C28x CPU
待機ブート	C28x CPU
セキュア・フラッシュ・ブート	C28x CPU

1.7.7.1.1 フラッシュ・ブート

フラッシュ・ブート・モードでは、フラッシュ内の設定されたメモリ・アドレスに分岐します。利用可能なすべてのフラッシュ・アドレス・オプションについては、[セクション 1.7.2](#) を参照してください。

1.7.7.1.2 RAM ブート

RAM ブート・モードは、RAM 内の設定されたメモリ・アドレスに分岐します。使用可能なすべての RAM アドレス・オプションについては、[セクション 1.7.2](#) を参照してください。

1.7.7.1.3 待機ブート

待機ブート・モードでは、[セクション 1.7.3](#) に示すメモリ・アドレスに分岐します。

1.7.7.2 ブートローダ

このセクションでは、ペリフェラル・ブート・ローダが使用できるブート・モードについて詳しく説明します。以下のブートローダでの使用がサポートされている、データ・ストリームの構造の詳細については、[セクション 1.8.1](#) を参照してください。

1.7.7.2.1 SCI ブート・モード

SCI ブート・モードでは、SCI-A から内部メモリに非同期でコードが転送されます。このブート・モードがサポートするのは、8 ビット・データ・ストリームの受信のみで、[図 1-4](#) に示すデータ・フローに従います。

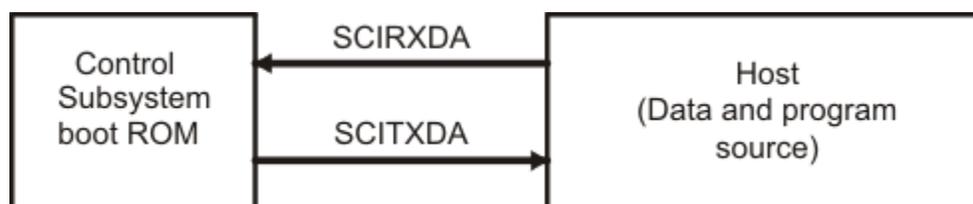


図 1-4. SCI ブートローダの動作の概要

このデバイスは、SCI-A ペリフェラルを介した通信により、外部ホストとやり取りをします。SCI ポートの自動ポー機能により、ホストとのポーレートをロックします。このため、SCI ローダには非常に高い柔軟性があり、このデバイスとの通信には、多数の異なるポーレートを 사용할 수 있습니다。

各データ転送の後、ブートローダは受信した 8 ビットの文字をホストにエコーバックします。これにより、ホストは、各文字がブートローダで受信されたことを確認できます。

高いボーレートでは、受信データ・ビットのスルーレートが、トランシーバおよびコネクタの性能の影響を受ける場合があります。通常のシリアル通信は正常に動作しますが、高い (通常は 100kbaud を超える) ボーレートにおける自動ボー検出の信頼性がこのスルーレートにより制限され、自動ボーロック機能が動作しない可能性があります。この問題を回避するために、以下を推奨します。

1. より低いボーレートを使用して、ホストと SCI ブートローダの間でボーロックを実現します。
2. この低いボーレートで、受け取るべきアプリケーションまたはカスタム・ローダをロードします。
3. その後ホストは、ロードされたアプリケーションとハンドシェイクし、SCI ボーレート・レジスタを目的の高いボーレート用に設定できます。

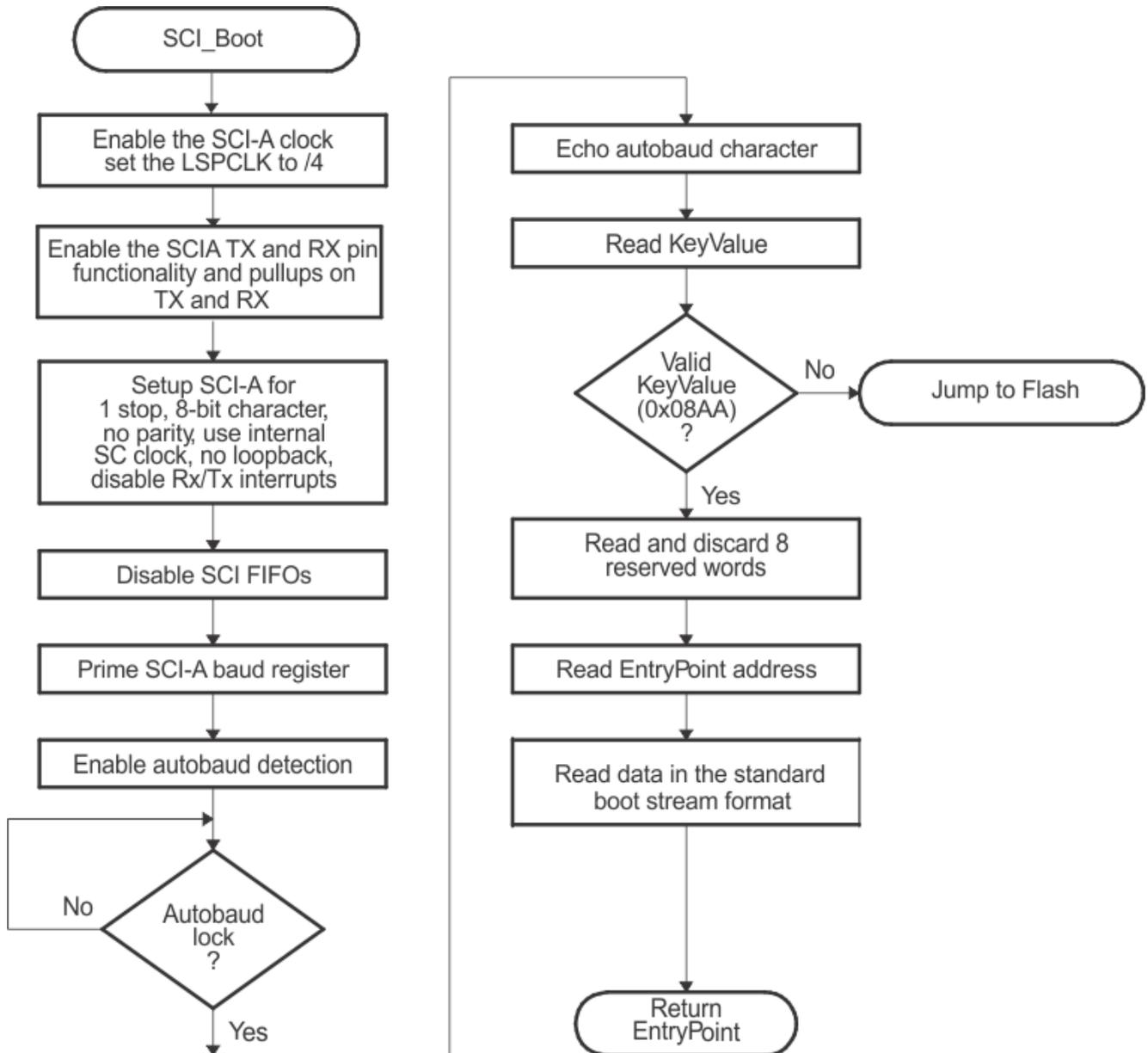


図 1-5. SCI ブート機能の概要

1.7.7.2.2 SPI ブート・モード

図 1-6 に示すように SPI ロードでは、16 ビットまたは 24 ビットでのアドレス指定が可能な SPI 互換のシリアル EEPROM、またはシリアル・フラッシュ・デバイスが、SPI-A ピン上に存在することを想定しています。SPI ブートローダは 8 ビットのデータ・ストリームをサポートしており、16 ビットのデータ・ストリームはサポートしていません。

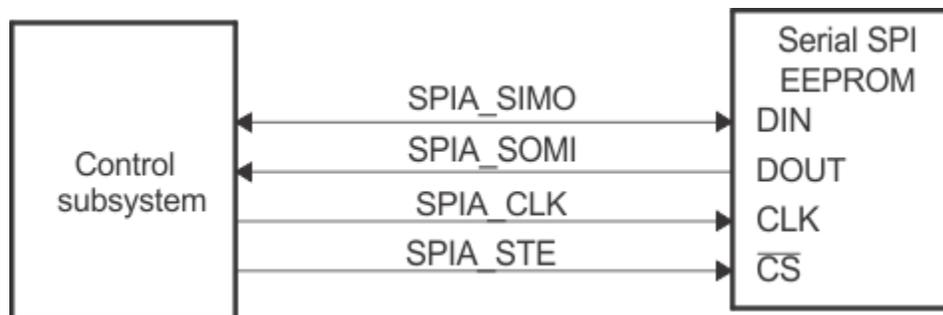


図 1-6. SPI ブートローダの動作の概要

SPI ブート ROM ロードは、シリアル SPI EEPROM またはフラッシュとインターフェイスを取るために、SPI モジュールを初期化します。このタイプのデバイスには、Xicor X25320 (4Kx8) および Xicor X25256 (32Kx8) SPI のシリアル SPI EEPROM、および Atmel AT25F1024A シリアル・フラッシュなどがありますが、これらに限定されません。

SPI ブート ROM ロードは、以下の設定で SPI を初期化します。FIFO イネーブル、8 ビット・キャラクタ、内蔵 SPICLK master モードおよびトーク・モード、クロック位相 = 1、極性 = 0、最も低速なボーレートを使用。

別のデバイスの SPI ポートからダウンロードを実行する場合は、そのデバイスを **slave** モードで動作するように設定し、シリアル SPI EEPROM をエミュレートする必要があります。SPI_Boot 機能を開始した直後に、SPI ピンのピン機能がプライマリに設定され、SPI が初期化されます。初期化は、可能な限り低速で実行されます。SPI が初期化され、キー値が読み取られたら、ボーレートまたは低速ペリフェラル・クロックを変更するように指定できます。表 1-29 に、SPI で使用される 8 ビット・データ・ストリームを示します。

表 1-29. SPI 8 ビット・データ・ストリーム

バイト	内容
1	LSB: AA (メモリ幅 = 8 ビットの場合のキー値)
2	MSB: 08H (メモリ幅 = 8 ビットの場合のキー値)
3	LSB: LOSPCP
4	MSB: SPIBRR
5	LSB: 将来使用のため予約済み
6	MSB: 将来使用のため予約済み
...	...
...	予約済み
...	...
17	LSB: 将来使用のため予約済み
18	MSB: 将来使用のため予約済み
19	LSB: エントリ・ポイントの上半分 (MSW) PC[23:16]
20	MSB: エントリ・ポイントの上半分 (MSW) PC[31:24] (注: 常に 0x00)
21	LSB: エントリ・ポイントの下半分 (LSW) PC[7:0]
22	MSB: エントリ・ポイントの下半分 (LSW) PC[15:8]
...	...
...	このセクションのデータ。
...	...
...	汎用データ・ストリームの説明で示された、サイズ / 送信先、アドレス / データ形式のデータ・ブロック

表 1-29. SPI 8 ビット・データ・ストリーム (continued)

バイト	内容
...	...
...	このセクションのデータ。
n	LSB: 00h
n+1	MSB: 00h - ソースの終了を示す

シリアル SPI EEPROM からのデータ転送は、「バースト」モードで実行されます。転送はすべてバイト・モード (8 ビット / 文字の SPI) で実行されます。以下に、このシーケンスについてステップバイステップで説明します。

1. SPI-A ポートが初期化されます。
2. 表 1-37 により構成した SPI オプションで定義される GPIO ピンは、シリアル SPI EEPROM またはフラッシュのチップ選択として使用されます。
3. SPI-A が、シリアル SPI EEPROM またはフラッシュに対し、読み取りコマンドを出力します。
4. SPI-A が、シリアル SPI EEPROM にアドレス 0x0000 を送信します。つまりホストは、アドレス 0x0000 から始まるダウンロード可能なパッケージが、EEPROM またはフラッシュ内に存在することを要求しています。このローダは、16 ビット・アドレスと 24 ビット・アドレスの両方に互換性があります。
5. 次にフェッチされたワードは、8 ビット・データ・ストリームのキー値 (0x08AA) と一致している必要があります。このワードの最下位バイトは最初に読み出されたバイトで、最上位バイトは次にフェッチされるバイトです。これは、SPI 上のすべてのワード転送に適合します。キー値が一致しない場合、ロードは中止され、ブートローダはフラッシュにジャンプします。
6. 次にフェッチされる 2 バイトは、低速ペリフェラル・クロック・レジスタ (LOSPCP) と SPI ボーレート・レジスタ (SPIBRR) の値を変更するために使用します。最初に読み取られたバイトは LOSPCP の値で、2 番目に読み取られたバイトは SPIBRR の値です。次の 7 つのワードは、将来の拡張のために予約されています。SPI ブートローダはこれら 7 つのワードを読み取り、破棄します。
7. 次の 2 つのワードで、ブート・ロード・プロセスの完了後に実行が続行される、32 ビットのエン트리・ポイント・アドレスを構成します。これは通常、SPI ポート経由でダウンロードされるプログラムのエン트리・ポイントです。
8. その後、SPI ポートを経由して、コードやデータの複数のブロックが、外部シリアル SPI EEPROM からメモリにコピーされます。コードのブロックは、先に述べた標準的なデータ・ストリーム構造の中で編成されています。これは、ブロック・サイズが 0x0000 となるまで行われます。その時点で、エン트리・ポイント・アドレスが呼び出しルーチンに返され、ブートローダが終了し、指定されたアドレスで実行が再開されます。

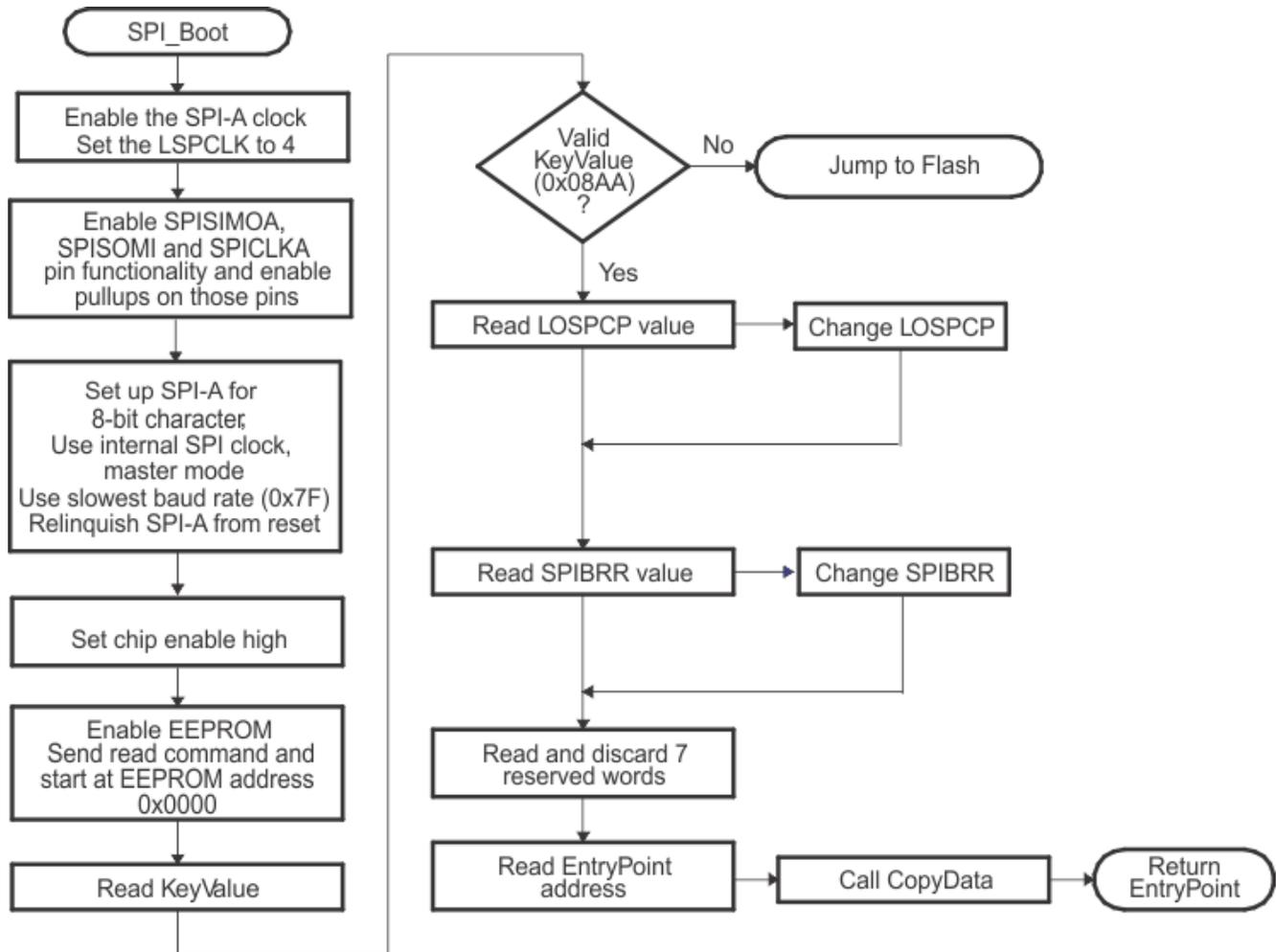


図 1-7. EEPROM フローからのデータ転送

1.7.7.2.3 I2C ブート・モード

I2C ブートローダは、[図 1-8](#) に示すように、I2C 互換の 8 ビット EEPROM デバイスが、I2C-A バスのアドレス 0x50 に存在することを想定しています。この EEPROM は、このセクションで説明するように、16 ビットのベース・アドレス・アーキテクチャを使用する、標準的な I2C EEPROM プロトコルに準拠している必要があります。

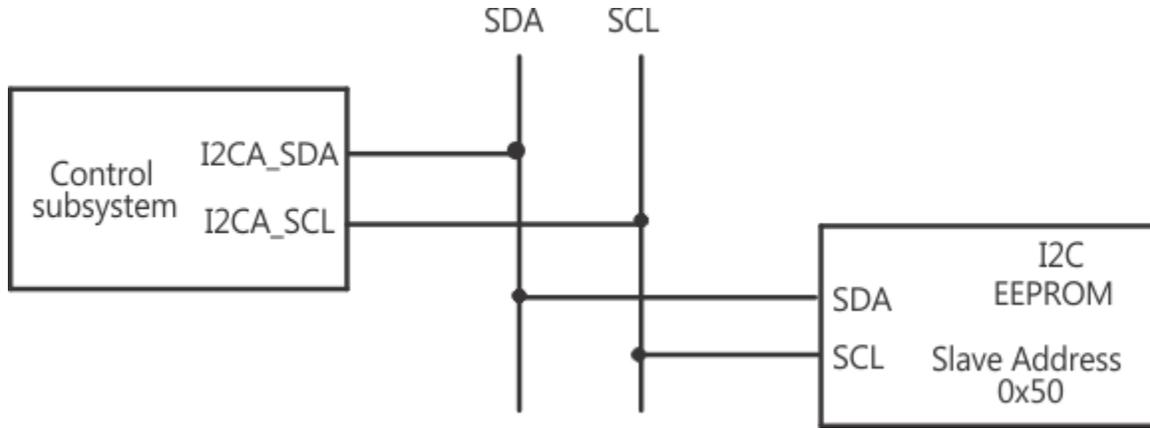


図 1-8. アドレス 0x50 の EEPROM デバイス

EEPROM 以外のデバイスからダウンロードを実行しようとする場合は、そのデバイスを **slave** モードで動作するように設定し、I2C EEPROM をエミュレートする必要があります。I2C ブート機能に移行した直後に、GPIO ピンは I2C-A 動作用に構成され、I2C が初期化されます。ブートを I2C モジュールから行なう場合は、次の要件を満たしている必要があります。

- デバイスへの入力周波数は適切な範囲内である必要があります。
- EEPROM は、**slave** アドレスである、0x50 に置かれる必要があります。

ビット周期プリスケアラ (I2CCLKH および I2CCLKL) は、10MHz のシステム・クロックにおいて、I2C が 50% デューティ・サイクルの 100kHz ビット・レート (標準 I2C モード) で実行するように、ブートローダによって構成されます。これらのレジスタは、EEPROM から最初の数バイトを受信した後で変更できます。これにより、残りのデータ読み取り時には、通信速度を 400kHz のビット・レート (高速 I2C モード) まで増やすことができます。

アービトレーション、バス・ビジー、および **slave** 信号はチェックされません。したがって、この初期化フェーズの間、他の **master** はバスを制御できません。I2C ブート・モード中に別の **master** が必要なアプリケーションの場合、その **master** では、ブートローダによる初期化の完了がアプリケーション・ソフトウェアから通知されるまで、I2C メッセージの送信を停止するように構成する必要があります。

確認応答なしビットは、EEPROM ベース・アドレスを初期化するために送信された最初のメッセージの間のみチェックされます。これは、アドレス 0x50 に EEPROM が存在していることを、処理続行の前に確認するためです。EEPROM が存在しない場合、データ読み取りメッセージ (I2C_Get Word) のアドレス・フェーズ中に、確認応答なしビットはチェックされません。データ読み取りメッセージ中に確認応答が受信されない場合、I2C バスは停止します。[表 1-30](#) に、I2C で使用される 8 ビット・データ・ストリームを示します。

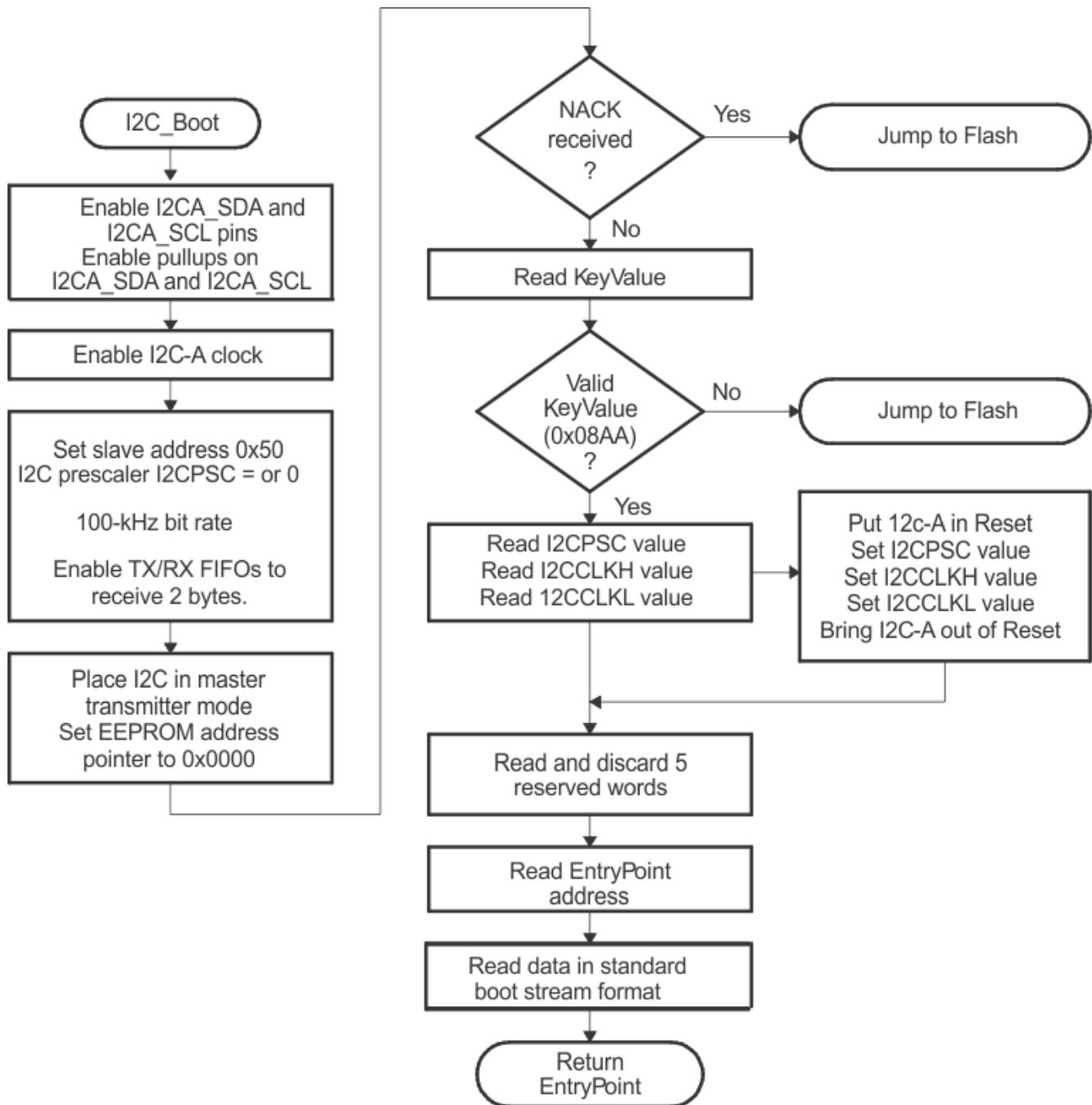


図 1-9. I2C ブート機能の概要

表 1-30. I2C 8 ビット・データ・ストリーム

バイト	内容
1	LSB: AA (メモリ幅 = 8 ビットの場合のキー値)
2	MSB: 08H (メモリ幅 = 8 ビットの場合のキー値)
3	LSB: I2CPSC[7:0]
4	予約済み
5	LSB: I2CCLKH[7:0]
6	MSB: I2CCLKH[15:8]
7	LSB: I2CCLKL[7:0]
8	MSB: I2CCLKL[15:8]
...	...
...	このセクションのデータです。
...	...
17	LSB: 将来使用のため予約済み。
18	MSB: 将来使用のため予約済み
19	LSB: エントリ・ポイントの上半分 PC
20	MSB: エントリ・ポイントの上半分、PC[22:16] (注: 常に 0x00)
21	LSB: エントリ・ポイントの下半分、PC[15:8]
22	MSB: エントリ・ポイントの下半分、PC[7:0]
...	...
...	このセクションのデータ。
...	...
...	汎用データ・ストリームの説明に示すように、サイズ/送信先、アドレス/データ形式のデータ・ブロック。
...	...
...	このセクションのデータ。
n	LSB: 00h
n+1	MSB: 00h - ソースの終了を示す

I2C ブートローダに必要な I2C EEPROM プロトコルを、[図 1-10](#) および [図 1-11](#) に示します。[図 1-10](#) に、EEPROM アドレス・ポイントを 0x0000 にセットし、キー値 (0x08AA) を読み取る最初の通信を示します。[図 1-11](#) に示した以降のすべての読み取りでは、同時に 2 バイトずつ読み取られます。

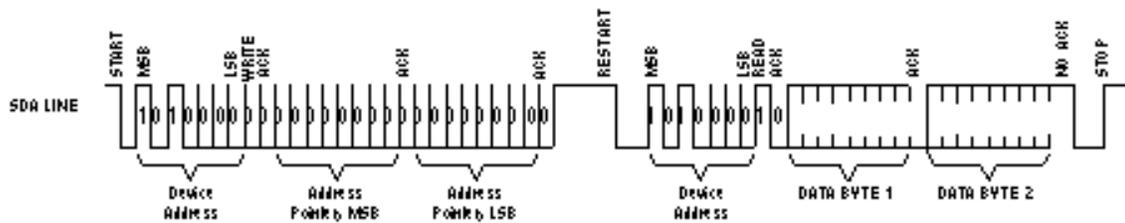


図 1-10. ランダム読み出し

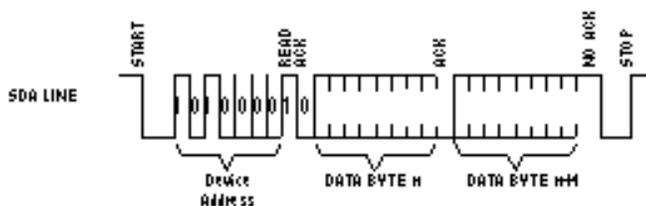


図 1-11. シーケンシャル読み出し

1.7.7.2.4 パラレル・ブート・モード

パラレル汎用 I/O (GPIO) ブート・モードでは、コードがホストから C28x の内部メモリに非同期で転送されます。各値は 8 ビット長で、[図 1-12](#) に示すのと同じデータ・フローに従います。

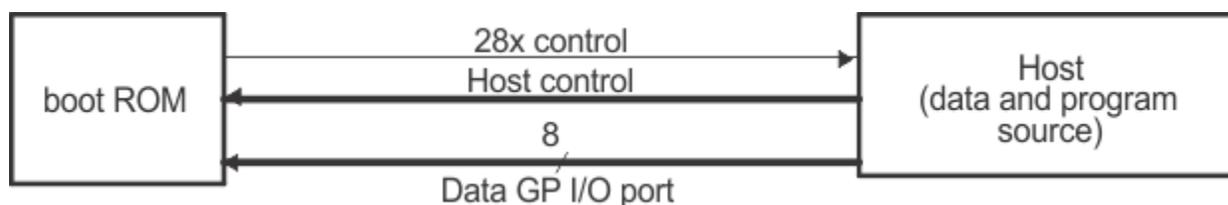


図 1-12. パラレル GPIO ブートローダの動作の概要

制御サブシステムは、ホスト制御と 28x の制御ラインをポーリングおよび駆動することで、外部ホスト・デバイスと通信します。GPIO [D0:D7] を使用して各ワードを正常に転送するには、[図 1-13](#) に示すハンドシェイク・プロトコルを使用する必要があります。このプロトコルは非常に堅牢であり、これにより、低速または高速のホストが master サブシステムと通信できます。

1 つの 16 ビット・ワードを形成するために、2 つの連続する 8 ビット・ワードが読み取られます。最下位バイト (LSB) が最初に読み取られ、その後に最上位バイト (MSB) が続きます。このケースでは、GPIO [D0:D7] からデータが読み取られます。

この 8 ビット・データ・ストリームを [表 1-31](#) に示します。

表 1-31. パラレル GPIO ブートの 8 ビット・データ・ストリーム

バイト	GPIO [D0:D7] (2 バイトの 1 バイト目)	GPIO [D0:D7] (2 バイトの 2 バイト目)	説明	
1	2	AA	08	0x08AA (メモリ幅 = 16 ビットの場合のキー値)
3	4	00	00	8 つの予約済みワード (ワード 2 から 9)
...
17	18	00	00	予約ワードの最後
19	20	BB	00	エントリ・ポイント PC[22:16]
21	22	DD	CC	エントリ・ポイント PC[15:0] (PC = 0x00BBCCDD)
23	24	NN	MM	ロードするデータの最初のブロックのブロック・サイズ = 0xMMNN ワード
25	26	BB	AA	最初のブロックの送信先アドレス Addr[31:16]
27	28	DD	CC	最初のブロックの送信先アドレス Addr[15:0] (Addr = 0xAABBCCDD)
29	30	BB	AA	ロードされているソースの最初のブロックの最初のワード = 0xAABB
...
...	このセクションのデータ。
...
.	BB	AA	AA	ロードされているソースの最初のブロックの最後のワード = 0xAABB

表 1-31. 並列 GPIO ブートの 8 ビット・データ・ストリーム (continued)

バイト	GPIO [D0:D7] (2 バイトの 1 バイト目)	GPIO [D0:D7] (2 バイトの 2 バイト目)	説明	
.	NN	MM	ロードする 2 番目のブロックのブロック・サイズ = 0xMMNN ワード	
.	BB	AA	2 番目のブロック・アドレスの送信先アドレス Addr[31:16]	
.	DD	CC	2 番目のブロック・アドレスの送信先アドレス Addr[15:0]	
.	BB	AA	ロードされているソース内の 2 番目のブロックの最初のワード	
.			...	
n	n+1	BB	AA	ロードされているソースの最後のブロックの最後のワード (必要に応じてセクションを追加)
n+2	n+3	00	00	0000H のブロック・サイズはソース・プログラムの終了を示します

最初にデバイスは、28x 制御ピンを Low にすることで、デバイスはデータ転送を開始する準備ができたことをホストに通知します。その後、ホスト・ロードは DSP 制御ピンを Low にすることで、データ転送を開始します。この完全なプロトコルを図 1-13 に示します。

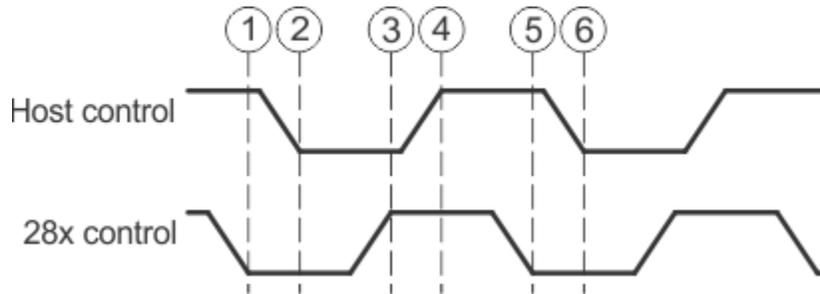


図 1-13. 並列 GPIO ブートローダのハンドシェイク・プロトコル

1. デバイスは、28x 制御ピンを Low にすることで、デバイスはデータの受信を開始する準備ができたことを示します。
2. ブートローダは、GPIO [D0:D7] にホストがデータを置くまで待機します。ホストは、ホスト制御ピンを Low にすることで、データの準備ができたことをデバイスに伝えます。
3. データを読み取ったデバイスは、28x 制御ピンを High にすることで、読み取りが完了したことをホストに伝えます。
4. ブートローダは、ホストがホスト制御ピンを High にしてデバイスを承認するまで待機します。
5. デバイスは、再度 28x 制御ピンを Low にすることで、さらにデータを受信する準備ができたことを示します。

このプロセスは、データ値が送信されるごとに繰り返されます。

図 1-14 に、並列 GPIO ブートローダ・フローの概要を示します。

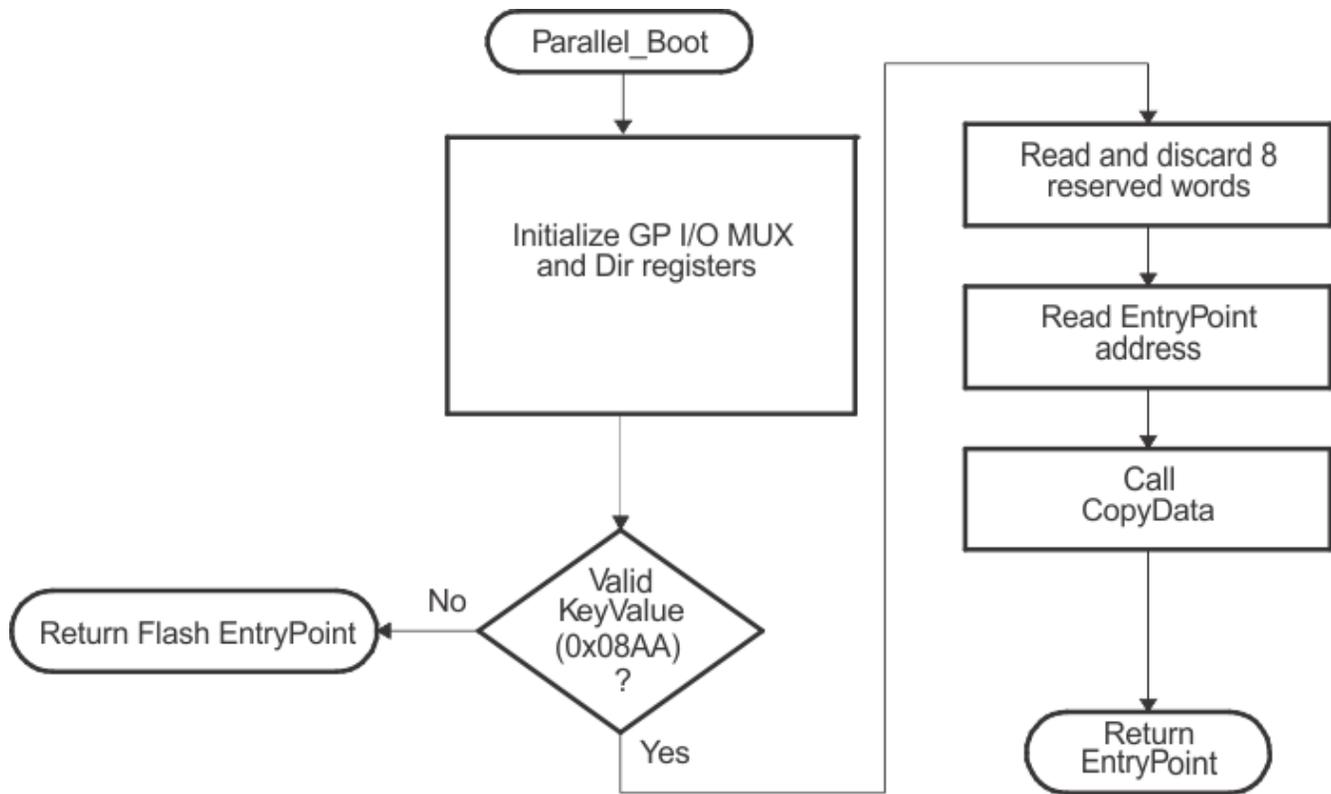


図 1-14. パラレル GPIO ブート機能の概要

図 1-15 に、ホスト側からの転送フローを示します。このモードでは、ホストはデバイスを待機し、またデバイスはホストを待機するため、CPU およびホストでの動作速度は重要ではありません。このプロトコルの手法は、ホストがデバイスより高速に動作している場合と、デバイスより低速で動作している場合の両方で動作します。

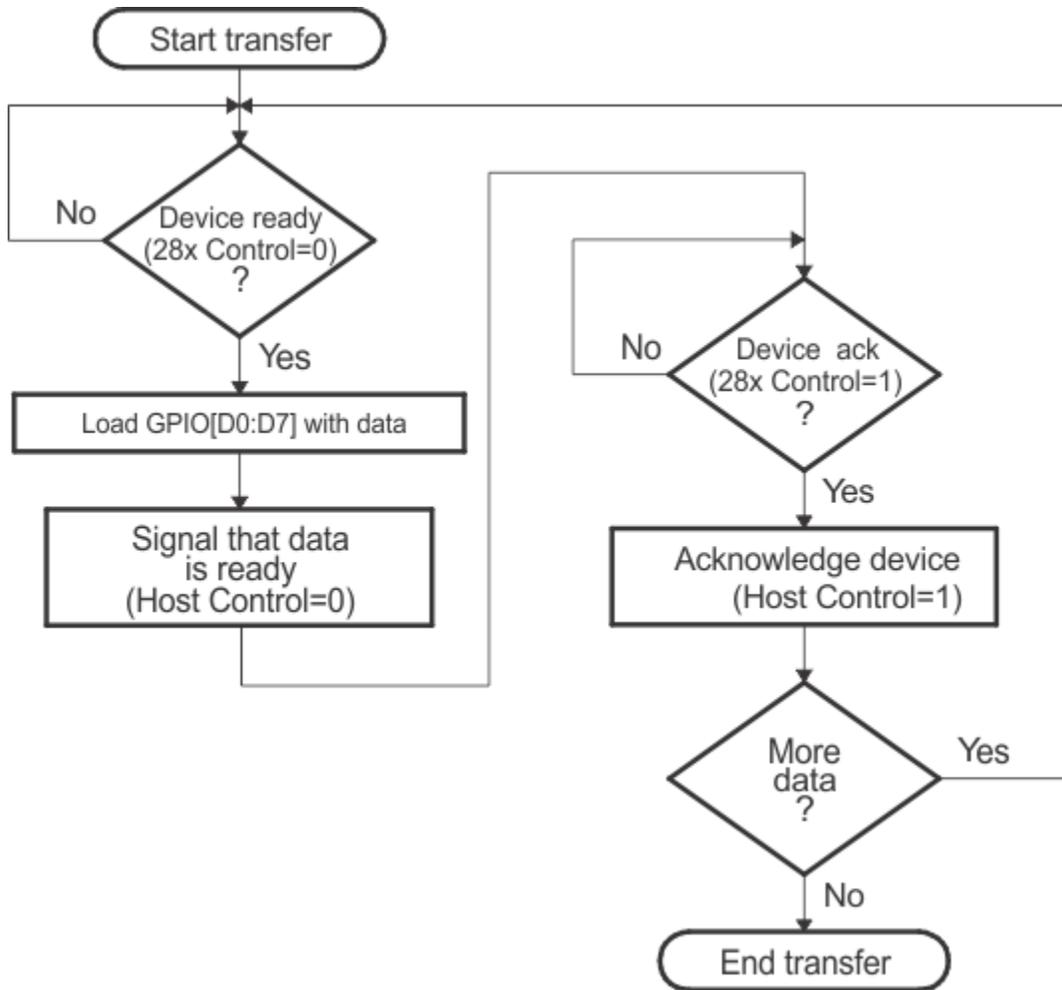


図 1-15. パラレル GPIO モード - ホスト転送フロー

図 1-16 に、パラレル・ポートから単一ワードのデータを読み取る場合に使用されるフローを示します。

- 8 ビット・データ・ストリーム

図 1-16 に示すように、この 8 ビット・ルーチンは、ポートから最初に読み取った上位 8 ビットを廃棄し、また、ビット位置 7 が D7 でマスクされビット位置 6 が D6 でマスクされた下位 8 ビットを、フェッチするワードの最下位バイト (LSB) として扱います。その後、このルーチンは 2 回目の読み取りを実行して、最上位バイト (MSB) をフェッチします。その後、このルーチンは 2 回目の読み取りを実行して、最上位バイト (MSB) をフェッチします。次に、このルーチンは MSB と LSB を 1 つの 16 ビット値に組み合わせ、それを呼び出しルーチンに戻します。

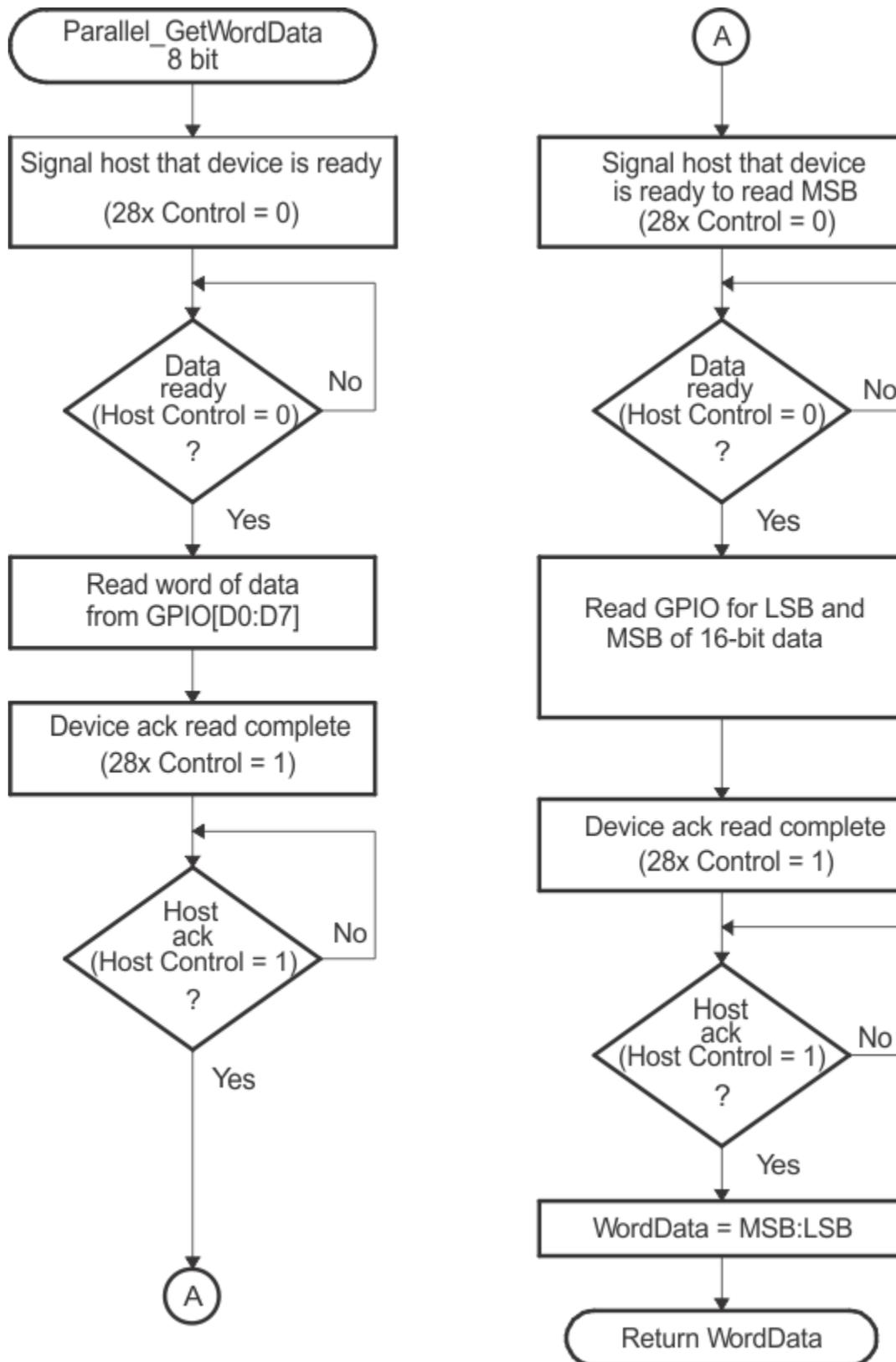


図 1-16. 8 ビット・パラレル GetWord 関数

1.7.7.2.5 CAN ブート・モード

CAN ブートローダは、[図 1-17](#) に示すように、CAN-A から内部メモリに対し非同期でコードを転送します。ホストでは、任意の CAN ノードを使用できます。この通信は、11 ビットの標準識別子 (MSGID は 0x1) でデータ・フレームごとに 2 バイトを使用して、最初に行われます。より高いデータ・スループットが求められる場合、ホストはカーネルをダウンロードして CAN を再構成できます。

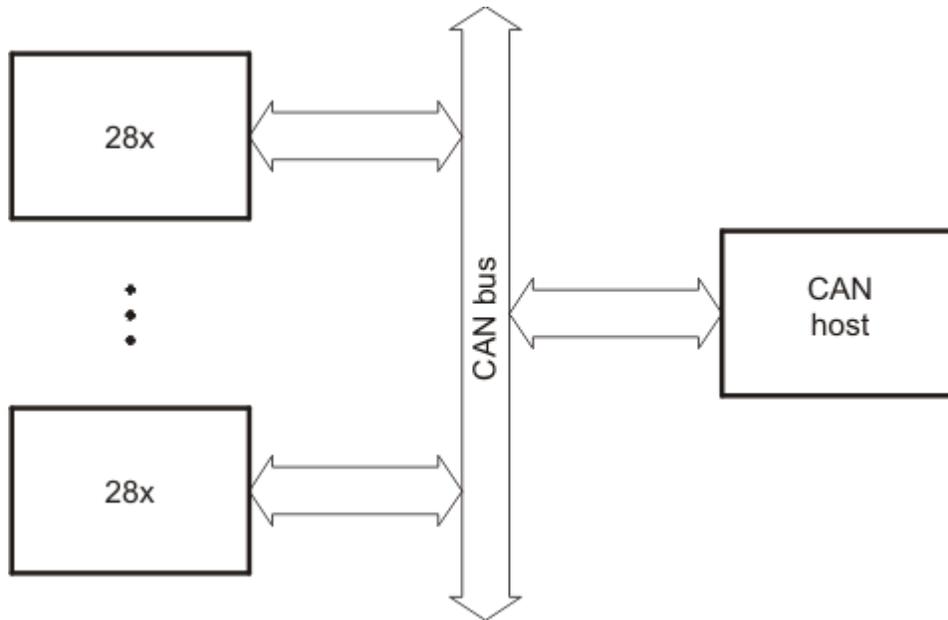


図 1-17. CAN-A ブートローダ動作の概要

[表 1-32](#) に示すように、20MHz の外部発振器により 100Kbps のビット・レートを実現するといったように、ビット・タイミング・レジスタをプログラムします。

表 1-32. 内部発振器のビット・レート値

OSCCLK	SYSCCLK	ビット・レート
20MHz	10MHz	100Kbps

表示されている SYSCCLKOUT の値は、デフォルトの PLL 設定でのリセット値です。BRP とビット時間の値は、それぞれ 10 と 20 にハードコードされています。

注

CAN ブート・ローダは、ビット・クロック・ソースとして XTAL を使用し、システム・クロック・ソースとしては INTOSC2 を使用します。

メールボックス 1 は、ブートローダ通信用の MSGID に標準の 0x1 を使用してプログラムされます。CAN ホストは、同時に 2 バイトのみを送信できます。最初に送信されるのは LSB で、次に MSB が送信されます。たとえば、0x08AA というワードをデバイスに送信する場合は、先に AA を送信し、次に 08 を送信します。CAN ブートローダのプログラム・フローは、SCI ブートローダと同じです。CAN ブートローダのデータ・シーケンスを、[表 1-33](#) に示します。

表 1-33. CAN 8 ビット・データ・ストリーム

バイト	2 バイトの 1 バイト目	2 バイトの 2 バイト目	説明	
1	2	AA	08	0x08AA (メモリ幅 = 16 ビットの場合のキー値)
3	4	00	00	予約済み
5	6	00	00	予約済み
7	8	00	00	予約済み
9	10	00	00	予約済み
11	12	00	00	予約済み
13	14	00	00	予約済み
15	16	00	00	予約済み
17	18	00	00	予約済み
19	20	BB	AA	エン트리・ポイント、PC[22:16]
21	22	DD	CC	エン트리・ポイント、PC[15:0] (PC = 0xAABBCCDD)
23	24	NN	MM	ロードするデータの最初のブロックのブロック・サイズ = 0xMMNN ワード
25	26	BB	AA	最初のブロックの送信先アドレス Addr[31:16]
27	28	DD	CC	最初のブロックの送信先アドレス Addr[15:0] (Addr = 0xAABBCCDD)
29	30	BB	AA	ロードされているソースの最初のブロックの最初のワード = 0xAABB
...			
...				このセクションのデータ。
...				...
.		BB	AA	ロードされているソースの最初のブロックの最後のワード = 0xAABB
.		NN	MM	ロードする 2 番目のブロックのブロック・サイズ = 0xMMNN ワード
.		BB	AA	2 番目のブロックの送信先アドレス Addr[31:16]
.		DD	CC	2 番目のブロックの送信先アドレス Addr[15:0]
.		BB	AA	ロードされているソース内の 2 番目のブロックの最初のワード
.			
n	n+1	BB	AA	ロードされているソースの最後のブロックの最後のワード (必要に応じてセクションを追加)
n+2	n+3	00	00	0000H のブロック・サイズはソース・プログラムの終了を示します

1.7.8 GPIO の割り当て

このセクションでは、BOOT_DEF メモリの領域 Z1-OTP-BOOTDEF-LOW / Z2-OTP-BOOTDEF-LOW および Z1-OTP-BOOTDEF-HIGH / Z2-OTP-BOOTDEF-HIGH でのブート・モードの設定に使用される GPIO とブート・オプションの値について詳細に説明します。BOOT_DEFx の構成方法については、[セクション 1.4.2](#) を参照してください。ブート・モード・オプションを選択するときは、使用する特定のデバイス・パッケージのピン・マルチプレクサ・オプションで、必要なピンが利用可能であることを確認します。

デフォルトのブート・モード GPIO ピン:

- ブート・モード・ピン 0 - GPIO32
- ブート・モード・ピン 1 - GPIO24

ブート・ピン選択のガイドライン:

- PWM 機能を持つピンは避けてください。
- アナログまたは USB ピンにすることはできません。
- ブート・モード選択ピンとデフォルトのブート・ペリフェラル・ピンは、すべてのパッケージで使用できます。
- JTAG エミュレーション・ピンと水晶振動子ピンは避けてください。
- ブート・モード選択ピンは入力にすることができます。
- ピンに PHY ブートストラップ機能を持たせることはできません。

表 1-34. SCI ブート・オプション

オプション	BOOTDEF 値	SCITXDA GPIO	SCIRXDA GPIO	サポート対象のパッケージ
0 (デフォルト)	0x01	GPIO29	GPIO28	すべて
1	0x21	GPIO1	GPIO0	すべて
2	0x41	GPIO8	GPIO9	48-QFN, 64-QFP
3	0x61	GPIO7	GPIO3	すべて
4	0x81	GPIO16	GPIO3	48-QFP, 48-QFN, 64-QFP

表 1-35. CAN ブート・オプション

オプション	BOOTDEF 値	CANTXA GPIO	CANRXA GPIO	サポート対象のパッケージ
0 (デフォルト)	0x02	GPIO4	GPIO5	すべて
1	0x22	GPIO32	GPIO33	48-QFP, 48-QFN, 64-QFP
2	0x42	GPIO2	GPIO3	48-QFP, 48-QFN, 64-QFP
3	0x62	GPIO13	GPIO12	48-QFP, 48-QFN, 64-QFP

注

F280013x と F280015x の CANTXA GPIO オプション 0 (デフォルト) の選択肢は異なります。他のすべての CAN ブート・オプション GPIO の選択肢は同じです。詳細については、該当するデバイスのデータシートを参照してください。

表 1-36. I2C ブート・オプション

オプション	BOOTDEF 値	SDAA GPIO	SCLA GPIO	サポート対象のパッケージ
0	0x07	GPIO0	GPIO1	すべて
1	0x27	GPIO32	GPIO33	48-QFP, 48-QFN, 64-QFP
2	0x47	GPIO5	GPIO4	すべて

表 1-37. SPI ブート・オプション

オプション	BOOTDEF 値	SPISIMOA	SPISOMIA	SPICLKA	SPISTEA	サポート対象のパッケージ
0	0x06	GPIO7	GPIO1	GPIO3	GPIO5	すべて
1	0x26	GPIO16	GPIO1	GPIO3	GPIO0	48-QFP、48-QFN、64-QFP
2	0x46	GPIO8	GPIO10	GPIO9	GPIO11	64-QFP
3	0x66	GPIO16	GPIO13	GPIO12	GPIO29	48-QFP、48-QFN、64-QFP

表 1-38. パラレル・ブート・オプション

オプション	BOOTDEF 値	D0~D7 GPIO	C28x (DSP) 制御 GPIO	ホスト制御 GPIO	サポート対象のパッケージ
0 (デフォルト)	0x00	D0 - GPIO0	GPIO224	GPIO242	すべて
		D1 - GPIO1			
		D2 - GPIO3			
		D3 - GPIO4			
		D4 - GPIO5			
		D5 - GPIO7			
		D6 - GPIO28			
		D7 - GPIO29			
1	0x20	D0 - GPIO0	GPIO12	GPIO13	48-QFP、48-QFN、64-QFP
		D1 - GPIO1			
		D2 - GPIO2			
		D3 - GPIO3			
		D4 - GPIO4			
		D5 - GPIO5			
		D6 - GPIO6			
		D7 - GPIO7			
2	0x40	D0 - GPIO0	GPIO16	GPIO29	48-QFP、48-QFN、64-QFP
		D1 - GPIO1			
		D2 - GPIO2			
		D3 - GPIO3			
		D4 - GPIO4			
		D5 - GPIO5			
		D6 - GPIO6			
		D7 - GPIO7			

1.7.9 セキュア ROM 関数 API

セキュア ROM 内には、アプリケーションが EXEONLY フラッシュ / RAM タスクをセキュアな方法で実行するために呼び出す、いくつかの関数が用意されています。

注

アプリケーションは、いずれかの EXEONLY 関数 API を呼び出す前に割り込みを無効にできます。

プログラム・カウンタ (PC) がセキュア ROM の EXEONLY 関数 API のコード内にあるときに、CPU (C28) によってベクトル・フェッチ要求が行われると、リセットが起動します (C28 からの場合 RSN)。その結果、PC が EXEONLY API 関数のいずれかを実行しているときに、NMI、ITRAP、またはバス障害が発生した場合も、この NMI / ITRAP / 障害を処理できなくなります。これは、サブシステムにリセットが発行されるためです。

コード・ゾーン 1 のセキュア・コピー関数、およびにより、EXEONLY フラッシュをセキュアな方法で EXEONLY RAM にコピーできます。このソースは EXEONLY フラッシュから、EXEONLY RAM にする必要があります。EXEONLY ROM または EXEONLY RAM の RAM へのコピーは、サポートされていません。フラッシュと RAM の両方を EXEONLY にセットし、またゾーンも同じに設定する必要があります。また、コピー・サイズがフラッシュ・セクタの境界を超えないようにする必要があります。これらの内、いずれかの要件に違反すると、障害ステータスが返されます。データのコピーが成功すると、コピーされた 16 ビット・ワードの個数が返されます。

表 1-39. セキュア・コピー・コード関数

CPU	関数プロトタイプ	関数のパラメータ	関数の戻り値
CPU (C28x)	<code>uint16_t SecureCopyCodeZ1(uint32_t size, uint16_t*dst, uint16_t*src)</code>	size : コピーする 16 ビット・ワードの個数 dst : EXEONLY RAM 内のコピー先メモリ・アドレス src : EXEONLY フラッシュ内のソース・メモリ・アドレス	0xFFFF : コピーされた 16 ビット・ワードの個数を返します。 0x0000 : 次のいずれかを示します。コピー長がゼロ。コピー・サイズがフラッシュ・セクタの境界を超えた。フラッシュと RAM が同じゾーンに属していない。フラッシュと RAM が EXEONLY に設定されていない。データ・コピー中にエラーが発生した。

ゾーン 1 のセキュア CRC 計算関数により、安全な方法で EXEONLY メモリの安全 CRC チェックを実行できます。CRC の長さで指定する値は 1 から 8 にする必要があります。この際、1 は 32 個の 16 ビット・ワードの CRC サイズ、8 は 4096 個の 16 ビット・ワードの CRC サイズを表します。ソース・アドレスでは CRC の開始アドレスを指定し、送信先アドレスでは結果の CRC 値が格納されるロケーションを指定します。ソースと送信先のメモリは、同じゾーンに構成する必要があります。さらに、CRC の長さを、フラッシュ・セクタまたは RAM ブロックの境界を超えないように指定する必要があります。これらの内、いずれかの要件に違反すると、障害ステータスが返されます。CRC が成功すると、CRC が実行された 16 ビット・ワードの個数が返されます。

表 1-40. セキュア CRC 計算関数

CPU	関数プロトタイプ	関数のパラメータ	関数の戻り値
CPU (C28x)	<code>uint16_t SecureCRCCalcZ1(uint16_t len_id, uint16_t*dst, uint16_t*src)</code>	len_id: CRC 長オプション (32, 64, 128, 256, 512, 1024, 2048, 4096 個の 16 ビット・ワード) を表す 1 から 8 の数値。 dst: 結果として生成される CRC の保存先メモリ・アドレス src: CRC 計算を開始するソース・メモリ・アドレス	0xFFFF : CRC 計算された 16 ビット・ワードの個数を返します 0x0000 : 次のいずれかを示します。長さオプションが無効。ソース・アドレスが長さ値の「法」となっていない。保存先アドレスがセキュア RAM 内にない。CRC サイズがフラッシュ・セクタまたは RAM ブロック境界をまたいでいる。ソースと保存先メモリが同じゾーンに属していない。

CMAC 計算および比較関数 を使用すると、フラッシュ・メモリ・ブロックの CMAC シグネチャを計算し、その結果をゴールデン・シグネチャと比較できます。これは、セキュア・ブート・モードでブート・イメージを認証するために使用されます。

表 1-41. CMAC 計算関数

CPU	関数プロトタイプ	関数のパラメータ	関数の戻り値
CPU (C28x)	<pre>uint32_t CPU1BROM_calculateCMAC(uint32_t startAddress, uint32_t endAddress, uint32_t signatureAddress)</pre>	<p>startAddress: CMAC が計算されるべきメモリの開始アドレス</p> <p>endAddress: CMAC が計算されるべきメモリの終了アドレス</p> <p>signatureAddress: ゴールデン CMAC シグネチャが格納されているロケーションのアドレス</p>	<p>0xFFFFFFFFU: 計算された CMAC シグネチャがゴールデン・シグネチャと一致しない (失敗)</p> <p>0xA5A5A5A5U: 指定されたメモリ範囲が 128 ビットの境界にアラインされていない、もしくは長さがゼロ</p> <p>0xE1E1E1E1U: AES エンジンがタイムアウト</p> <p>0x00000000U: エラーなし</p>

1.7.10 クロック初期化

ブートアップ中、ブート ROM はリセット・ソースに応じてデバイス・クロックを初期化することで、ブート時間応答を高速化しています。クロックの設定は、POR および XRS リセット・タイプに対してのみ、ブート ROM コードによって実行されます。その他のリセットの場合、ブート ROM は、リセット前のクロック設定を使用して実行を開始します。

注

クロックの構成は、ブートアップ時に CPU が実行します。ブート・プロセス中に PLL が使用された場合は、ユーザー・アプリケーションに分岐する前に、ブート ROM コードによって PLL がバイパスされます。

表 1-42. CPU ブート・クロック・ソース

ソース	周波数	概要
INTOSC2	10MHz	デフォルトのクロック・ソース
INTOSC1	10MHz	電源投入時またはデバイス・リセット直後にクロックの喪失が検出された場合、クロック・ソースとして設定されます
SYSPLL	95MHz, 47.5MHz	メイン・ブート・フローの一部として、または MPOST POR メモリ・テストでのブート・フローの一部として有効化します (オプション)。メモリ・テストの完了後、PLL はバイパスされ、無効化されます。MPOST POR メモリ・テストの有効化の詳細については、 セクション 1.7.11.2 を参照してください。

表 1-43. ブート後の CPU クロックの状態

リセット・ソース	クロックの状態
POR/XRS	<ol style="list-style-type: none"> INTOSC2 を使用 システム・クロック分周器を 1 に設定
その他のすべてのリセット	デバイス・リセット前のクロック設定を維持します。

1.7.11 ブート・ステータス情報

ブート ROM は、その実行中に発生するさまざまなアクションとイベントを記録しています。その理由は、デバイスのデフォルト状態では NMI などの例外が有効化されており、それぞれに応じて処理する必要があるからです。ブート ROM により、ブート・ステータス情報が RAM 内のロケーションに格納されます。これにより、ユーザー・アプリケーションはこのブート・ステータスを読み取り、アプリケーションのニーズに応じてそれらのイベントを処理するために、必要なアクションを実行できます。

1.7.11.1 ブート・ステータス

ブート ROM の状態とブート・ステータスは、32 ビット・アドレスで M0RAM 内に書き込まれます。このステータスは POR または XRS リセット発生時にクリアされます。他のリセットでは、前のステータスが保持されます。たとえば、このステータスに反映されている最新のブート ROM アクションを表示するためには、デバッガ・デバイスのリセットを実行する前にステータスをクリアしておく必要があります。

表 1-44. ブート・ステータス・アドレス

概要	アドレス
ブート ROM ステータス	0x0000 0002

表 1-45. ブート・ステータスのビット・フィールド

値	説明
0x4000 0000	フラッシュ 2T が未準備
0x2000 0000	TRIM 読み込みエラー
0x1000 0000	RAM 初期化エラー
0x0800 0000	フラッシュ検証エラー
0x0400 0000	DCSM 初期化 LP エラー
0x0200 0000	DCSM 初期化の LP が無効
0x0100 0000	SYSPLL が正常に有効化
0x0040 0000	クロック消失 NMI 発生
0x0020 0000	RAM 修正不能エラー NMI 発生
0x0010 0000	フラッシュ修正不能エラー NMI 発生
0x0008 0000	RL NMI 発生
0x0004 0000	ERAD NMI 発生
0x0002 0000	ブート ROM が PIE のミスマッチを検出
0x0001 0000	ブート ROM が ITRAP を検出
0x0000 8000	ブート ROM の実行完了
0x0000 4000	ウォッチドッグのセルフ・テストが失敗
0x0000 2000	ブート ROM が POR を処理
0x0000 1000	ブート ROM が XRS を処理
0x0000 0800	ブート ROM がすべてのリセットを処理済み
0x0000 0400	POR メモリ・テストが完了
0x0000 0200	DCSM の初期化が完了
0x0000 0100	RAM の初期化が完了
0x0000 000B	待機ブートが開始済み
0x0000 0009	CAN ブートが開始済み
0x0000 0008	I2C ブートが開始済み
0x0000 0007	SPI ブートが開始済み
0x0000 0006	SCI ブートが開始済み
0x0000 0005	RAM ブートが開始済み
0x0000 0004	パラレル・ブートが開始済み
0x0000 0003	セキュア・フラッシュ・ブートが開始済み
0x0000 0002	フラッシュ・ブートが開始済み
0x0000 0001	ブート ROM が実行開始済み

1.7.11.2 ブート・モードと MPOST (メモリ・パワー・オン・セルフテスト) のステータス

ブート・フロー中にブート・モードがデコードされると、そのブート・モードの値が RAM に書き込まれます。さらに、MPOST POR メモリ・テストが実行されると、そのテスト結果が RAM に書き込まれます。

詳細については、『[C2000™ メモリ・パワー・オン・セルフテスト \(M-POST\) アプリケーション・レポート](#)』を参照してください。

表 1-46. ブート・モードと MPOST ステータスのアドレス

概要	アドレス
ブート・モード	0x0000 0004
MPOST と POR のメモリ・テスト結果	0x0000 0006

1.7.12 ROM バージョン

ROM のリビジョンおよびリリース日の情報は、[表 1-47](#) で指定されている ROM のロケーションに格納されています。リビジョン番号の値「0x100」はバージョン「1.0」、「0x101」はバージョン「1.1」を表し、以下同様に続きます。リビジョンの日付が「0x0119」の値であれば、「01/19」または「2019 年 1 月」を表します。

表 1-47. ブート ROM のバージョン情報

内容	アドレス
リビジョン番号	0x003F 9804
改訂日	0x003F 9805
ビルド番号	0x003F 9806

1.8 ブートローダ使用のアプリケーション・ノート

1.8.1 ブートローダのデータ・ストリーム構造

このセクションでは、ブート ROM とホスト・デバイス間でブート・データの転送を可能にする、データ転送プロトコルまたはストリーム構造について詳しく説明します。このデータ転送プロトコルには、各 C2000 デバイス上のブートローダで互換性があります。

[表 1-48](#) および [Example 1-2](#) に、ブートローダが受信するデータ・ストリームの構造を示します。この基本的な構造は、すべてのブートローダについて同一です。また、C54x hex ユーティリティによって生成された、C54x ソース・データ・ストリームをベースにしています。C28x hex ユーティリティ (hex2000.exe) は、この構造をサポートするために更新されています。hex2000.exe ユーティリティは、C2000 コード生成ツールに包含されています。データ・ストリーム構造内のすべての値は 16 進数です。プロジェクトをこの形式に変換するための、C28x hex ユーティリティの詳細な使用方法については、[セクション 1.8.2](#) を参照してください。

データ・ストリームの最初の 16 ビット・ワードは、キー値と呼ばれます。キー値は、ブートローダに対して、受信ストリームの幅を示すために使用されます。8 または 16 ビット。すべてのブートローダが、8 ビットと 16 ビットの両方のストリームを受け入れるわけではないことに注意してください。有効なデータ・ストリーム幅については、各ローダの詳細情報を参照してください。8 ビットのデータ・ストリームの場合、キー値は 0x08AA です。16 ビットのデータ・ストリームの場合、キー値は 0x10AA です。無効なキー値がブートローダで受信された場合、そのロードは中止されます。

続く 8 個のワードは、レジスタ値を初期化するため、またはブートローダに値を渡すことでブートローダ機能を拡張するために使用されます。ブートローダがこれらの値を使用しない場合、その値は将来の使用のために予約されます。ブートローダは値を読み取ってから、その値を破棄します。現在、これらのワードを使用してレジスタを初期化するのは、SPI と I2C およびパラレル・ブートローダのみです。

10 番目と 11 番目のワードで、22 ビットのエン트리・ポイント・アドレスを構成します。このアドレスは、ブート・ロードの完了後に PC を初期化するために使用されます。このアドレスは、ほとんどの場合、ブートローダによってダウンロードされるプログラムのエン트리・ポイントになります。

データ・ストリームの 12 番目のワードは、転送される最初のデータ・ブロックのサイズを示します。このブロック・サイズは、8 ビットのデータ・ストリーム形式として定義されています。たとえば、8 ビットのデータ・ストリームから 20 個の 8 ビット・データ値によるブロックを転送する場合、そのブロック・サイズは 16 ビット・ワード 10 個分として 0x000A となります。

次の 2 つのワードでは、ローダーに対し、データ・ブロックの送信先アドレスを示します。サイズおよびアドレスの情報の後に、データ・ブロックを構成する 16 ビット・ワードが続きます。

この、ブロック・サイズ/送信先アドレスのパターンは、転送されるデータの各ブロックについて繰り返されます。すべてのブロックの転送が完了すると、信号のブロック・サイズとして 0x0000 がローダに送信され、転送の完了が伝えられます。この時点で、ローダは呼び出しルーチンに対しエン트리・ポイント・アドレスを返し、クリーンアップを実行してから終了します。その後、入力データ・ストリームの内容で定義されたとおりに、エン트리・ポイント・アドレスで実行が続行されます。

表 1-48. 8 ビット・データ・ストリームでの LSB/MSB ロード・シーケンス

		内容	
バイト		LSB (2 個のうちの最初のバイト)	MSB (2 個のうちの 2 番目のバイト)
1	2	LSB: AA (メモリ幅 = 8 ビットの場合のキー値)	MSB: 08H (メモリ幅 = 8 ビットの場合のキー値)
3	4	LSB: レジスタ初期化値または予約済み	MSB: レジスタ初期化値または予約済み
5	6	LSB: レジスタ初期化値または予約済み	MSB: レジスタ初期化値または予約済み
7	8	LSB: レジスタ初期化値または予約済み	MSB: レジスタ初期化値または予約済み
...
17	18	LSB: レジスタ初期化値または予約済み	MSB: レジスタ初期化値または予約済み
19	20	LSB: エン트리・ポイントの上半分、PC[23:16]	MSB: エン트리・ポイントの上半分、PC[31:24] (常に 0x00)
21	22	LSB: エン트리・ポイントの下半分、PC[7:0]	MSB: エン트리・ポイントの下半分、PC[15:8]
23	24	LSB: 最初にロードするブロックのブロック・サイズ (ワード)。ブロック・サイズが 0 の場合は、ソース・プログラムの終了を示します。それ以外の場合、他のブロックが続きますたとえば、ブロック・サイズが 0x000A であれば、そのブロックが 10 ワードまたは 20 バイトであることを示します。	MSB: ブロック・サイズ
25	26	LSB: MSW 送信先アドレス、最初のブロック Addr[23:16]	MSB: MSW 送信先アドレス、最初のブロック Addr[31:24]
27	28	LSB: LSW 送信先アドレス、最初のブロック Addr[7:0]	MSB: LSW 送信先アドレス、最初のブロック Addr[15:8]
29	30	LSB: ロードされている最初のブロックの最初のワード	MSB: ロードされている最初のブロックの最初のワード
...
.	.	LSB: ロードする最初のブロックの最後のワード	MSB: ロードする最初のブロックの最後のワード
.	.	LSB: 2 番目のブロックのブロック・サイズ	MSB: 2 番目のブロックのブロック・サイズ
.	.	LSB: 2 番目のブロックの MSW 送信先アドレス Addr[23:16]	MSB: 2 番目のブロックの MSW 送信先アドレス Addr[31:24]
.	.	LSB: 2 番目のブロックの LSW 送信先アドレス Addr[7:0]	MSB: 2 番目のブロックの LSW 送信先アドレス Addr[15:8]
.	.	LSB: ロードされている 2 番目のブロックの最初のワード	MSB: ロードされている 2 番目のブロックの最初のワード
...
.	.	LSB: 2 番目のブロックの最後のワード	MSB: 2 番目のブロックの最後のワード
.	.	LSB: 最後のブロックのブロック・サイズ	MSB: 最後のブロックのブロック・サイズ
.	.	LSB: 最後のブロックの MSW 送信先アドレス Addr[23:16]	MSB: 最後のブロックの MSW 送信先アドレス Addr[31:24]
.	.	LSB: 最後のブロックの LSW 送信先アドレス Addr[7:0]	MSB: 最後のブロックの LSW 送信先アドレス Addr[15:8]
.	.	LSB: ロードされている最後のブロックの最初のワード	MSB: ロードされている最後のブロックの最初のワード
...
...

表 1-48. 8 ビット・データ・ストリームでの LSB/MSB ロード・シーケンス (continued)

バイト		内容	
		LSB (2 個のうちの最初のバイト)	MSB (2 個のうちの 2 番目のバイト)
.	.	LSB: 最後のブロックの最後のワード	MSB: 最後のブロックの最後のワード
n	n+1	LSB: 00h	MSB: 00h - ソースの終了を示す

Example 1-2. データ・ストリーム構造 8 ビット

```

AA 08      ; 0x08AA 8bit key value
00 00 00 00 ; 8 reserved words
00 00 00 00
00 00 00 00
00 00 00 00
3F 00 00 80 ; 0x003F8000 EntryAddr, starting point after boot load completes
05 00      ; 0x0005 First block consists of 5 16-bit words
3F 00 10 90 ; 0x003F9010 First block will be loaded starting at 0x3F9010
01 00      ; Data loaded = 0x0001 0x0002 0x0003 0x0004 0x0005
02 00
03 00
04 00
05 00
02 00      ; 0x0002 - 2nd block consists of 2 16bit words
3F 00 00 80 ; 0x003F8000 2nd block will be loaded starting at 0x3F8000
00 77      ; Data loaded = 0x7700 0x7625
25 76
00 00      ; 0x0000 Size of 0 indicates end of data stream
After load has completed the following memory values will have been initialized as follows:
Location Value
0x3F9010 0x0001
0x3F9011 0x0002
0x3F9012 0x0003
0x3F9013 0x0004
0x3F9014 0x0005
0x3F8000 0x7700
0x3F8001 0x7625
PC Begins execution at 0x3F8000

```

1.8.2 hexC2000 ユーティリティ

ブートローダの機能を使用するには、[セクション 1.8.1](#) で説明されているように、データ・ストリームとブート・テーブルを生成する必要があります。C28x コード生成ツールに付属している HEX 変換ユーティリティ・ツールを使用すると、必須のブート・テーブルを含む必要なデータ・ストリームを生成できます。このセクションでは、hex2000 ユーティリティについて説明します。hex2000 によって実行されるファイル変換の例は、[Example 1-3](#) に示します。

hex ユーティリティでは、SCI、SPI、I2C、CAN、パラレル I/O ローダーに必要なブート・テーブルの作成をサポートしています。つまり、hex ユーティリティは、キー値、予約ビット、エントリ・ポイント、アドレス、ブロック開始アドレス、ブロック長、および終端値など、必要な情報をファイルに追加します。ブート・テーブルの内容は、ブート・モードと、hex 変換ユーティリティの実行時に選択したオプションによって若干異なります。ホストが必要とする実際のファイル形式 (ASCII、バイナリ、16 進、その他) は、特定のアプリケーションごとに異なるため、追加の変換が必要になる場合があります。

ブート・テーブルを構築するには、次のステップに従います。

1. コードをアセンブルまたはコンパイルします。

これにより、リンカが単一の出力ファイルを作成するために使用する、オブジェクト・ファイルが作成されます。

2. そのファイルをリンクします。

リンカは、すべてのオブジェクト・ファイルを、共通のオブジェクト・ファイル形式 (ELF) で単一の出力ファイルに結合します。コード・セクションをさまざまなメモリ・ブロックに割り当てるため、リンカは特定のリンカ・コマンド・ファイルを使用します。ブート・テーブル・データの各ブロックは、ELF ファイル内の初期化されたセクションに対応しています。初期化されていないセクションは、hex 変換ユーティリティによって変換されていません。以下の各オプションは便利です。

- リンカの **-m** オプションは、マップ・ファイルを生成するために使用します。このマップ・ファイルには、作成されたすべてのセクションと、それらのメモリ内のロケーション、および長さが記載されます。初期化されたセクションが想定通りに配置されたことを確認する場合に、このファイルをチェックすると便利です。
- リンカの **-w** オプションは、リンカがセクションをメモリ領域に自動的に割り当てた場合に、そのセクションを表示することをリンカに指定します。たとえば、コード内に、`.TI.ramfunc` というセクションがある場合です。

3. hex 変換ユーティリティを実行します。

望みのブート・モードに適したオプションを選択し、hex 変換ユーティリティを実行して、リンカによって生成された ELF ファイルをブート・テーブルに変換します。

コンパイルとリンク・プロセスの詳細については、『[TMS320C28x アセンブリ言語ツール・ユーザー・ガイド](#)』および『[TMS320C28x C/C++ コンパイラ最適化ユーザー・ガイド](#)』を参照してください。

表 1-49 に、ブートローダで使用可能な、hex 変換ユーティリティのオプションを示します。ブート・テーブルの生成に使用される hex2000 の操作の詳細については、『[TMS320C28x アセンブリ言語ツール・ユーザー・ガイド](#)』を参照してください。I2C ブートをサポートするための更新が行われています。最新情報については、CodeGen のリリース・ノートを参照してください。

表 1-49. ブート・ローダーのオプション

オプション	説明
-boot	すべてのセクションをブート可能な形式に変換します (SECTIONS 指令の代わりに使用)
-sci8	ブートローダ・テーブルのソースを SCI-A ポートの 8 ビット・モードとして指定します
-spi8	ブートローダ・テーブルのソースを SPI-A ポートの 8 ビット・モードとして指定します
-gpio8	ブートローダ・テーブルのソースを GPIO ポートの 8 ビット・モードとして指定します
-bootorg 値	ブートローダ・テーブルのソース・アドレスを指定します
-lospcp 値	LOSPCP レジスタの初期値を指定します。この値は、spi8 ブート・テーブル形式でのみ使用され、他のすべての形式では無視されます。値が 0x7F より大きい場合、値は 0x7F に切り詰められます。
-spibr 値	SPIBRR レジスタの初期値を指定します。この値は、spi8 ブート・テーブル形式でのみ使用され、他のすべての形式では無視されます。値が 0x7F より大きい場合、値は 0x7F に切り詰められます。
-e 値	ブート・ロード後に実行が開始されるエントリ・ポイントを指定します。この値では、アドレスまたはグローバル・シンボルを指定します。この値はオプションです。エントリ・ポイントを、コンパイル時にリンカ -e オプションを使用して定義すると、そのエントリ・ポイントをグローバル・シンボルに割り当てることができます。-e リンカ・オプションで別途定義されていない限り、通常、C 言語でのプログラムのエントリ・ポイントは <code>_c_int00</code> です。
-i2c8	ブートローダ・テーブルのソースを、8 ビットの I2C-A ポートとして指定します
-i2cpsc 値	I2CPCSC レジスタの値を指定します。この値は、EEPROM からデータを読み取る前、すべての I2C オプションのロードが完了した時点で有効になります。この値は、最小桁の 8 ビットに切り詰められ、I2C モジュール・クロックを 7 から 12 MHz の間に設定するために使用します。
-i2cclkh 値	I2CCLKH レジスタの値を指定します。この値は、EEPROM からデータを読み取る前、すべての I2C オプションのロードが完了した時点で有効になります。
-i2cckl 値	I2CCLKL レジスタの値を指定します。この値は、EEPROM からデータを読み取る前、すべての I2C オプションのロードが完了した時点で有効になります。

Example 1-3. HEX2000.exe コマンドの構文

```
C: HEX2000 GPIO34TOG.OUT -boot -gpio8 -a
where:
- boot Convert all sections into bootable form.
- gpio8 Use the GPIO in 8-bit mode data format. The eCAN
        uses the same data format as the GPIO in 8bit mode.
- a     Select ASCII-Hex as the output format.
```

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020 – REVISED MARCH 2023



GPIO モジュールは、デバイスのデジタルおよびアナログ I/O 多重化を制御し、共有ピンを使用してアプリケーションの柔軟性を最大限に高めます。ピンには汎用 I/O 名 (GPIO0、GPIO25、GPIO58 など) で名前が付けられています。これらのピンは、デジタル I/O (GPIO モードとも呼ばれます) として動作するように個別に選択することも、複数のペリフェラル I/O 信号のいずれかに接続することもできます。入力信号は、不要なノイズを除去するように条件を付けることができます。

1.1 はじめに.....	4
1.2 構成の概要.....	6
1.3 ADC ピンのデジタル入力 (AIO).....	7
1.4 ADC ピン上のデジタル入出力 (AGPIO).....	7
1.5 デジタル汎用 I/O 制御.....	7
1.6 入力認定.....	10
1.7 GPIO およびペリフェラルの多重化.....	13
1.8 内部プルアップ構成要件.....	17
1.9 ソフトウェア.....	18
1.10 GPIO レジスタ.....	19

1.1 はじめに

CPU 制御の I/O 機能に加えて、最大 12 個の独立したペリフェラル信号が 1 つの GPIO 対応ピンで多重化されます。各ピン出力は、ペリフェラルまたは CPU マスタのいずれかによって制御できます。

- CPU1

最大 8 個の I/O ポートが使用可能です。

- ポート A は GPIO0 ~ GPIO31 で構成されています
- ポート B は GPIO32 ~ GPIO63 で構成されています
- ポート C は GPIO64 ~ GPIO95 で構成されています
- ポート D は GPIO96 ~ GPIO127 で構成されています
- ポート E は GPIO128 ~ GPIO159 で構成されています
- ポート F は GPIO160 ~ GPIO191 で構成されています
- ポート G は GPIO192 ~ GPIO223 で構成されています
- ポート H は GPIO224 ~ GPIO255 で構成されています

注

一部の GPIO および I/O ポートは、特定のデバイスでは使用できない場合があります。利用可能な GPIO および I/O ポートについては、「GPIO レジスタ」セクションを参照してください。

このデバイスのアナログ信号は、デジタル入出力と多重化されています。これらのアナログ IO (AIO) ピンの一部は、デジタル出力機能がありません。その他のピンは、完全なデジタル入出力機能を持つアナログ・ピン (AGPIO) です。AIO (デジタル入力のみ) 機能を持つアナログ・ピンには、デバイス・データシートの「ピン属性」表に「AIO」が示されています。完全な入出力機能を持つアナログ・ピン (AGPIO ピン) には、デバイス・データシートの「ピン属性」表に「GPIO」が示されています。また、AGPIO ピンは、名前にアナログ信号と GPIO の両方のピン名が付いています。

図 1-1 に、シングル・ピンの GPIO ロジックを示します。

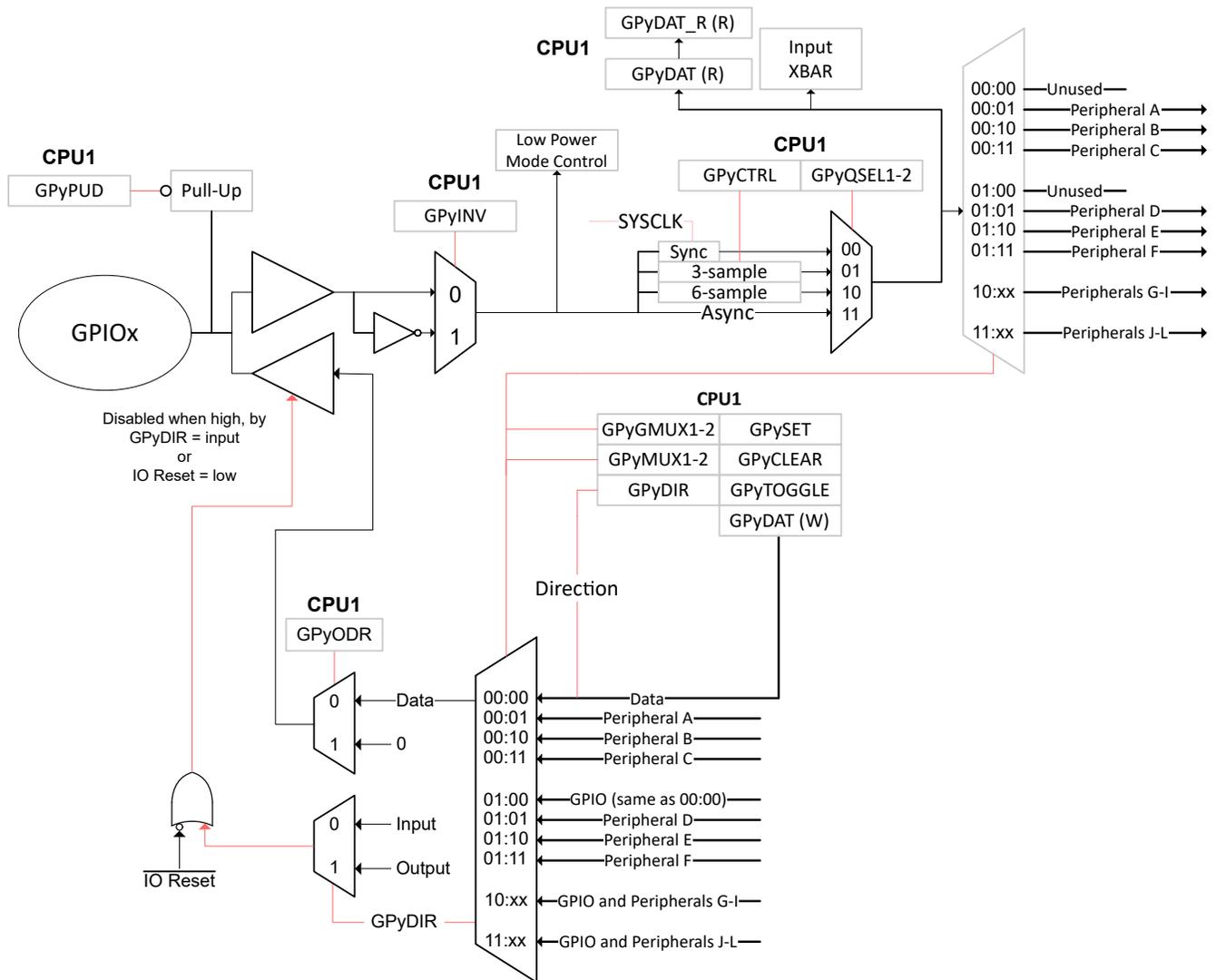


図 1-1. シングル・ピンの GPIO ロジック

図 1-1 には、注目すべき 2 つの重要な特長が示されています。1 つ目は、入力パスと出力パスが完全に分離されており、ピンのみで接続されていることです。2 つ目は、ペリフェラルの多重化がピンから遠く離れた場所で行われることです。その結果、CPU がペリフェラルの多重化とは独立してピンの物理状態を読み取ることが常に可能です。同様に、ペリフェラルの動作から外部割り込みを生成できます。入力認定やオープン・ドレイン出力などのすべてのピン・オプションは、すべてのペリフェラルで有効です。

注

JTAG は、反転や認定をサポートしない別の信号バスを使用します。

GPIO18/X2 と GPIO19/X1 は、発振器回路によって負荷が配置されているため、タイミングが異なります。GPIO18/X2 および GPIO19/X1 を GPIO として使用方法については、デバイス・データシートと、このドキュメントの「クロッキング」セクションを参照してください。

鋭いエッジ (大きい dv/dt) を持つデジタル信号を AIO または AGPIO に接続すると、隣接するアナログ信号に対してクロストークが発生する可能性があります。したがって、アナログ機能に隣接するチャネルを使用する場合は、AIO または AGPIO に接続する信号のエッジ・レートを制限します。

1.1.1 GPIO 関連の資料

基礎材料

- [C2000 アカデミー - GPIO](#)

入門資料

- 『C2000 デバイスで GPIO を最大限に使用方法』アプリケーション・レポート
- [\[FAQ\] C2000 GPIO に関する FAQ](#)

1.2 構成の概要

I/O ピンの構成は、いくつかの手順に分かれます。

1. デバイスのピン配置の計画

アプリケーションに必要なすべてのペリフェラルのリストを作成します。デバイスのデータ・マニュアルに記載されているペリフェラル・マルチプレクサの情報を使用して、ペリフェラル信号に使用する GPIO を選択します。、他の GPIO のどれを入力および出力として使用するかを決定します。

ペリフェラルの多重化を選択してから、GPyMUX1/2 および GPyGMUX1/2 レジスタに適切な値を書き込むことで、多重化を実装できます。ピンの GPyGMUX 値を変更するときは、マルチプレクサのグリッチを避けるため、対応する GPyMUX ビットを常に 0 に設定します。デフォルトでは、GPIO35 と GPIO37 を例外として、すべてのピンは汎用 I/O で、ペリフェラル信号ではありません。

2. (オプション) 内部プルアップ抵抗のイネーブル

プルアップ抵抗をイネーブルまたはディセーブルするには、GPIO プルアップ・ディセーブル・レジスタ (GPyPUD) の対応するビットに書き込みます。すべてのプルアップは、デフォルトでディセーブルされています。プルアップは、外部の信号により駆動されないとき、入力ピンを既知の状態に保つため使用できません。

3. 入力認定の選択

ピンを入力として使用する場合、必要な入力認定があるなら、指定します。入力認定のサンプリング期間は GPyCTRL レジスタで、認定の種類は GPyQSEL1 および GPyQSEL2 レジスタで選択します。デフォルトでは、GPIO35 と GPIO37 を例外として、すべての認定は PLLSYSCLK と同じサンプリング期間に同期されます。入力認定の説明については、[セクション 1.6](#) を参照してください。

4. 汎用 I/O ピンの方向の選択

GPIO として構成される各ピンについて、GPyDIR レジスタを使用し、ピンの方向として入力または出力を指定します。デフォルトでは、すべての GPIO ピンが入力です。ピンを出力に変更する前に、GPySET、GPyCLEAR、GPyDAT レジスタにその値を書き込み、駆動する値を出力ラッチにロードします。ラッチがロードされたら、GPyDIR に書き込んでピンの方向を変更します。デフォルトでは、すべての出力ラッチが 0 です。

GPyDAT_R レジスタを使用して、GPyDAT レジスタに書き込まれた値を読み取ることができます。

5. 低消費電力モードのウェークアップ・ソースの選択

GPIO 0~63 を使用して、低消費電力モードからシステムをウェークアップできます。ウェークアップする GPIO を 1 つ以上選択するには、GPIOLPMSEL0 および GPIOLPMSEL1 レジスタの適切なビットに書き込みます。これらのレジスタは、CPU システム・レジスタ領域の一部です。低消費電力モードと GPIO ウェークアップの詳細については、「システム制御および割り込み」の章の「低消費電力モード」セクションを参照してください。

6. 外部割り込みソースの選択

外部割り込みの構成は、2 手順のプロセスです。最初に割り込みをイネーブルにしてから、XINTnCR レジスタで極性を設定する必要があります。次に、入力の X-BAR 信号 4、5、6、13、14 の各ソースを選択して、XINT1~5 の GPIO ピンを設定する必要があります。入力 X-BAR アーキテクチャの詳細については、「クロスバー (X-BAR)」の章を参照してください。

1.3 ADC ピンのデジタル入力 (AIO)

一部の GPIO はアナログ・ピンと多重化されており、デジタル入力機能のみを備えています。これらは、「AIO」とも呼ばれます。このポートで AIO オプションのみがあるピンは、入力モードでのみ機能します。AIO 信号のリストについては、デバイスのデータシートを参照してください。デフォルトでは、これらのピンはアナログ・ピンとして機能し、GPIO はハイ・インピーダンス状態になります。GPYAMSEL レジスタは、これらのピンをデジタルまたはアナログ動作に構成するために使用します。

注

鋭いエッジ (大きい dv/dt) を持つデジタル信号を AIO に接続すると、隣接するアナログ信号に対してクロストークが発生する可能性があります。したがって、アナログ機能に隣接するチャンネルを使用する場合は、AIO に接続する信号のエッジ・レートを制限します。

1.4 ADC ピン上のデジタル入出力 (AGPIO)

一部の GPIO はアナログ・ピンと多重化されており、デジタル入出力機能を備えています。これらは、「AGPIO」とも呼ばれます。AIO とは異なり、AGPIO には完全な入力と出力の機能があります。

デフォルトでは、AGPIO は接続されていないため、構成する必要があります。表 1-1 に、AGPIO の構成方法を示します。

表 1-1. AGPIO の構成

AGPIOTRILA.GPIOy (デフォルト=0)	GPAXMSEL.GPIOy (デフォルト=1)	ピン接続先 :	
		ADC	GPIOy
0	0	-	あり
0	1	- ⁽¹⁾	- ⁽¹⁾
1	0	-	あり
1	1	あり	-

(1) デフォルトでは、AGPIO ピンに接続される信号はありません。ピン機能を使用するには、表内の他の行のいずれかを選択する必要があります。

注

鋭いエッジ (大きい dv/dt) を持つデジタル信号を AGPIO に接続すると、隣接するアナログ信号に対してクロストークが発生する可能性があります。したがって、アナログ機能に隣接するチャンネルを使用する場合は、AGPIO に接続する信号のエッジ・レートを制限できます。

1.5 デジタル汎用 I/O 制御

GPIO として構成されているピンの値は、以下のレジスタを使用して変更できます。

• GPyDAT レジスタ

各 I/O ポートには、1 つのデータ・レジスタがあります。データ・レジスタの各ビットは、1 つの GPIO ピンに対応します。ピンの構成 (GPIO またはペリフェラル機能) にかかわらず、データ・レジスタの対応するビットは、認定後のピンの現在の状態を反映します。GPyDAT レジスタに書き込むと、対応する出力ラッチがクリアまたは設定され、ピンが汎用出力 (GPIO 出力) としてイネーブルされている場合、ピンも Low または High に駆動されます。ピンが GPIO 出力として構成されていない場合、値はラッチされますが、ピンは駆動されません。ピンが後で GPIO 出力として構成された場合のみ、ラッチされた値がピンに駆動されません。

GPyDAT レジスタを使用して出力ピンのレベルを変更する場合、誤って別のピンのレベルを変更しないように注意してください。たとえば、読み取り / 変更 / 書き込み命令を使用して GPADAT レジスタのビット 0 に書き込むことで、GPIOA0 の出力ラッチのレベルを変更する場合、別の I/O ポート A 信号のレベルが命令の読み取り段階と書き込み段階の間で変化すると、問題が発生する可能性があります。以下は、なぜこのようなことが起こるのかについての分析です。

GPyDAT レジスタは、ラッチではなくピンの状態を反映します。これは、レジスタが実際のピンの値を反映していることを意味します。ただし、レジスタに書き込まれてから、新しいピンの値がレジスタに反映されるまでには遅延があります。これは、GPIO ピンの状態を変更するために、後続のプログラム・ステートメントでこのレジスタを使用する場合、問題となる可能性があります。以下に、2 つのプログラム・ステートメントが、現在 Low 状態の 2 つの異なる GPIO ピンを High 状態に駆動しようとする例を示します。

GPyDAT レジスタで読み取り / 変更 / 書き込み操作が使用される場合、最初の命令 (I1) の出力と入力の間には遅延があるため、2 番目の命令 (I2) は古い値を読み取り、その値を書き戻します。

```
GpioDataRegs.GPADAT.bit.GPIO1 = 1; //I1 performs read-modify-write of GPADAT
GpioDataRegs.GPADAT.bit.GPIO2 = 1; //I2 also a read-modify-write of GPADAT
//GPADAT gets the old value of GPIO1 due to the delay
```

2 番目の命令は、このペリフェラル・フレームでの書き込み後の読み取り保護により、最初の書き込みが完了するまで待機します。ただし、(I1) の書き込みから GPyDAT ビットにピンの新しい値 (1) が反映されるまでには多少の遅延があります。この遅延の間に、2 番目の命令は GPIO1 の古い値 (0) を読み取り、その値を GPIO2 の新しい値 (1) とともに書き戻します。したがって、GPIO1 ピンは Low のままです。

1 つの答えは、命令の間にいくつかの NOP を入れることです。より良い答えは、GPyDAT レジスタの代わりに GPySET/GPyCLEAR/GPyTOGGLE レジスタを使用することです。これらのレジスタは常に 0 を読み戻し、0 の書き込みを行っても、影響はありません。現在変更中の他のビットを妨げることなく、変更する必要があるビットのみを指定できます。

- **GPyDAT_R レジスタ**

GPyDAT_R レジスタは読み取り専用レジスタで、ピンのステータスの代わりに GPyDAT レジスタに書き込まれた値を返します。これらのレジスタに書き込みを行っても、影響はありません。

- **GPYSET レジスタ**

セット・レジスタは、他のピンを妨げることなく、指定された GPIO ピンを High に駆動するために使用されます。各 I/O ポートには 1 つのセット・レジスタがあり、各ビットは 1 つの GPIO ピンに対応します。セット・レジスタは常に 0 を読み戻します。対応するピンが出力として構成されている場合、セット・レジスタのそのビットに 1 を書き込むと、出力ラッチが High に設定され、対応するピンが High に駆動されます。ピンが GPIO 出力として構成されていない場合、値はラッチされますが、ピンは駆動されません。ピンが後で GPIO 出力として構成された場合のみ、ラッチされた値がピンに駆動されます。セット・レジスタの任意のビットに 0 の書き込みを行っても、影響はありません。

- **GPYCLEAR レジスタ**

クリア・レジスタは、他のピンを妨げることなく、指定された GPIO ピンを Low に駆動するために使用されます。各 I/O ポートには 1 つのクリア・レジスタがあります。クリア・レジスタは常に 0 を読み戻します。対応するピンが汎用出力として構成されている場合、クリア・レジスタの対応するビットに 1 を書き込むと、出力ラッチがクリアされ、ピンは Low に駆動されます。ピンが GPIO 出力として構成されていない場合、値はラッチされますが、ピンは駆動されません。ピンが後で GPIO 出力として構成された場合のみ、ラッチされた値がピンに駆動されます。クリア・レジスタの任意のビットに 0 の書き込みを行っても、影響はありません。

- **GPYTOGGLE レジスタ**

トグル・レジスタは、他のピンを妨げることなく、指定された GPIO ピンを反対のレベルに駆動するために使用されます。各 I/O ポートには 1 つのトグル・レジスタがあります。トグル・レジスタは常に 0 を読み戻します。対応するピンが出力として構成されている場合、トグル・レジスタのそのビットに 1 を書き込むと、出力ラッチが反転し、対応するピンが反対方向にプルされます。つまり、出力ピンが Low に駆動されている場合、トグル・レジスタの対応するビットに 1 を書き込むと、ピンが High になります。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 を書き込むと、ピンが Low になります。ピンが GPIO 出力として構成されていない場合、値はラッチされますが、ピンは駆動されません。ピンが後で GPIO 出力として構成された場合のみ、ラッチされた値がピンに駆動されます。トグル・レジスタの任意のビットに 0 の書き込みを行っても、影響はありません。

1.6 入力認定

入力認定方式は、非常に柔軟性が高くなるように設計されています。GPyQSEL1 および GPyQSEL2 レジスタを構成して、各 GPIO ピンの入力認定のタイプを選択します。GPIO 入力ピンの場合、認定は SYSCLKOUT にのみ同期するか、サンプリング・ウィンドウによる認定とするか指定できます。ペリフェラル入力として構成されているピンの場合、入力は SYSCLKOUT に同期するか、サンプリング・ウィンドウで認定するだけでなく、非同期にすることもできます。このセクションの残りの部分では、利用可能なオプションについて説明します。

1.6.1 同期なし (非同期入力)

このモードは、入力同期が不要なペリフェラルや、ペリフェラル自身が同期を行うときに使用されます。例として、通信ポートの McBSP、SCI、SPI、I²C などがあります。また、ePWM トリップ・ゾーン (TZn) 信号は、SYSCLKOUT が存在するかどうかにかかわらず機能できます。

注

ペリフェラル自身が同期を行っているとき、入力同期を使用すると、予期しない結果を招くことがあります。この場合、GPIO ピンが非同期用に構成されていることを、使用者が確認する必要があります。

1.6.2 SYSCLKOUT のみへの同期

これは、リセット時のすべてのピンのデフォルトの認定モードです。このモードでは、入力信号はシステム・クロック (SYSCLKOUT) にのみ同期されます。受信信号は非同期であるため、デバイスへの入力を変更するには SYSCLKOUT 周期の遅延が必要です。この信号に対して、これ以上の認定は行われません。

1.6.3 サンプリング・ウィンドウを使用した認定

このモードでは、信号は最初にシステム・クロック (SYSCLKOUT) に同期され、その後、指定されたサイクル数が認められてから、入力の変更が許可されます。図 1-2 および 図 1-3 に、不要なノイズを除去するために入力認定がどのように実行されるかを示します。このタイプの認定では、ユーザーは以下の 2 つのパラメータを指定します。1) サンプリング周期、つまり信号をサンプリングする頻度、2) 取得するサンプリング数。

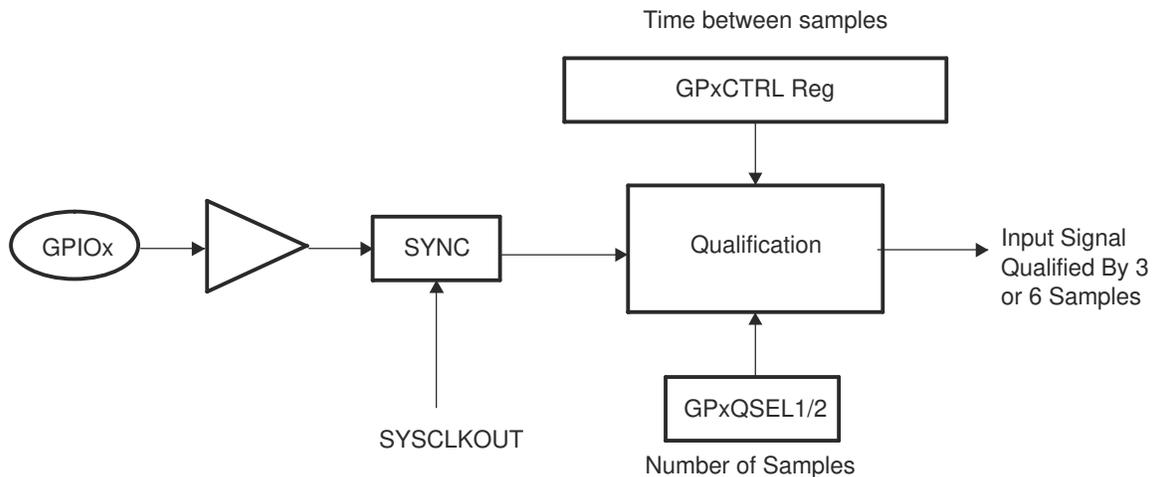


図 1-2. サンプリング・ウィンドウを使用した入力認定

サンプル間の時間 (サンプリング周期) :

信号を認定するために、入力信号が一定の周期でサンプリングされます。サンプリング周期はユーザーが指定し、CPU クロック (SYSCLKOUT) に対するサンプリング間の時間、つまり信号がサンプリングされる頻度を決定します。

サンプリング周期は、GPxCTRL レジスタの認定周期 (QUALPRDn) ビットによって指定されます。サンプリング周期は、8 つの入力信号のグループごとに構成可能です。たとえば、GPIO0 ~ GPIO7 は GPaCTRL[QUALPRD0] の設定を使用し、GPIO8 ~ GPIO15 は GPaCTRL[QUALPRD1] の設定を使用します。表 1-2 および表 1-3 に、サンプリング周期またはサンプリング周波数と GPxCTRL[QUALPRDn] の設定の関係を示します。

表 1-2. サンプリング周期

サンプリング周期	
GPxCTRL[QUALPRDn] = 0 の場合	$1 \times T_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn] ≠ 0 の場合	$2 \times \text{GPxCTRL[QUALPRDn]} \times T_{\text{SYSCLKOUT}}$
ここで、 $T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の期間です	

表 1-3. サンプリング周波数

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

これらの式から、特定の SYSCLKOUT の周波数に対するサンプル間の最小時間と最大時間を計算できます。

例：最大サンプリング周波数：

GPxCTRL[QUALPRDn] = 0 の場合

サンプリング周波数は $f_{\text{SYSCLKOUT}}$ です

たとえば、 $f_{\text{SYSCLKOUT}} = 60\text{MHz}$ の場合

信号は 60MHz、つまり 16.67ns ごとに 1 サンプルでサンプリングされます。

例：最小サンプリング周波数：

GPxCTRL[QUALPRDn] = 0xFF (255) の場合

サンプリング周波数は $f_{\text{SYSCLKOUT}} \times 1 \div (2 \times \text{GPxCTRL[QUALPRDn]})$ です

たとえば、 $f_{\text{SYSCLKOUT}} = 60\text{MHz}$ の場合

信号は $60\text{MHz} \times 1 \div (2 \times 255)$ (117.647kHz)、つまり 8.5μs ごとに 1 サンプルでサンプリングされます。

サンプリング数：

信号がサンプリングされる回数は、認定選択 (GPyQSEL1、GPyQSEL2) レジスタで指定されているように、3 サンプルまたは 6 サンプルのいずれかです。連続した 3 サイクルまたは 6 サイクルが同じである場合、入力の変化はデバイスに渡されます。

サンプリング・ウィンドウの合計幅：

サンプリング・ウィンドウは、図 1-3 に示すように、入力信号がサンプリングされる時間です。サンプリング周期の式と取得するサンプルの数を使用することで、ウィンドウの合計幅を決定できます。

入力クオリファイヤが入力の変化を検出するには、信号のレベルがサンプリング・ウィンドウの幅以上の期間安定している必要があります。

ウィンドウ内のサンプリング周期の数は、常に取得されたサンプルの数より 1 つ少なくなります。3 サンプル・ウィンドウの場合、サンプリング・ウィンドウの幅は 2 サンプル周期の幅であり、サンプリング周期は表 1-2 で定義されています。同様に、6 サンプル・ウィンドウの場合、サンプリング・ウィンドウの幅は 5 サンプル周期の幅です。表 1-4 および表 1-5 に、GPxCTRL[QUALPRDn] および取得したサンプル数に基づいて、サンプリング・ウィンドウの合計幅を決定するために使用される計算を示します。

表 1-4. ケース 1 : 3 サンプルのサンプリング・ウィンドウの幅

サンプリング・ウィンドウの合計幅	
GPxCTRL[QUALPRDn] = 0 の場合	$2 \times T_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn] ≠ 0 の場合	$2 \times 2 \times \text{GPxCTRL[QUALPRDn]} \times T_{\text{SYSCLKOUT}}$
ここで、 $T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の期間です	

表 1-5. ケース 2 : 6 サンプルのサンプリング・ウィンドウの幅

サンプリング・ウィンドウの合計幅	
GPxCTRL[QUALPRDn] = 0 の場合	$5 \times T_{\text{SYSCLKOUT}}$
GPxCTRL[QUALPRDn] ≠ 0 の場合	$5 \times 2 \times \text{GPxCTRL[QUALPRDn]} \times T_{\text{SYSCLKOUT}}$
ここで、 $T_{\text{SYSCLKOUT}}$ は、SYSCLKOUT の期間です	

注

外部信号の変化は、サンプリング周期と SYSCLKOUT の両方に関して非同期です。外部信号の非同期特性により、ロジックが信号の変化を確実に検出できるように、入力サンプリング・ウィンドウの幅よりも長い時間、安定に保持される必要があります。追加に必要な時間は、最大で追加のサンプリング周期 + $T_{\text{SYSCLKOUT}}$ になります。

認定ロジックが変化を検出するために入力信号を安定させておく必要時間は、データ・マニュアルに記載されています。

認定ウィンドウの例 :

図 1-3 に示す例では、入力認定は次のように構成されています。

- GPxQSEL1/2 = 1, 0 これは、6 サンプルの認定を示しています。
- GPxCTRL [QUALPRDn] = 1 サンプル周期は $t_w(SP) = 2 \times GPxCTRL[QUALPRDn] \times T_{SYSCLKOUT} = 2 \times T_{SYSCLKOUT}$ です。

この構成では、次の結果が得られます。

- サンプルング・ウィンドウの幅は次のとおりです。

$$t_w(IQSW) = 5 \times t_w(SP) = 5 \times 2 \times GPxCTRL[QUALPRDn] \times T_{SYSCLKOUT} = 5 \times 2 \times T_{SYSCLKOUT}$$

- たとえば、 $T_{SYSCLKOUT} = 16.67ns$ の場合、サンプルング・ウィンドウの期間は次のようになります。

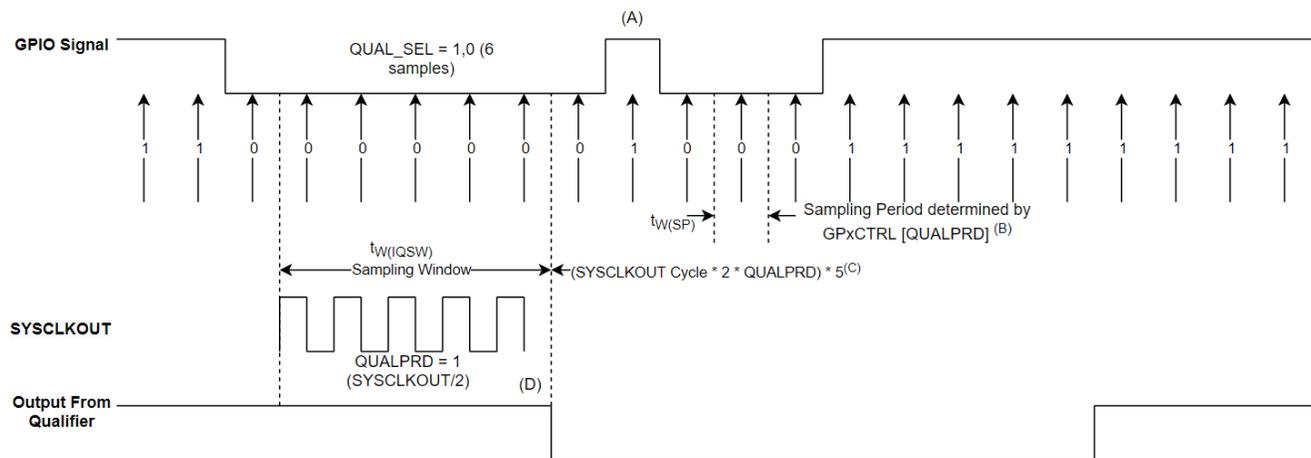
サンプルング周期、 $t_w(SP) = 2 \times T_{SYSCLKOUT} = 2 \times 16.67ns = 33.3ns$

サンプルング・ウィンドウ、 $t_w(IQSW) = 5 \times t_w(SP) = 5 \times 33.3ns = 166.7ns$

- サンプルング周期および SYSCLKOUT に対する入力の非同期特性を考慮すると、入力信号の変化を検出するには、最大で 1 つの追加サンプルング周期と SYSCLK 周期が必要です。この例では以下の式で計算されます。

$$t_w(IQSW) + t_w(SP) + T_{SYSCLKOUT} = 166.7ns + 33.3ns + 16.67ns = 216.7ns$$

- 図 1-3 では、グリッチ (A) は認定ウィンドウより短く、入力クオリファイヤによって無視されます。



A. This glitch will be ignored by the input qualifier. The QUALPRD bit field specifies the qualification sampling period. It can vary from 00 to 0xFF. If QUALPRD = 00, then the sampling period is 1 SYSCLKOUT cycle. For any other value "n", the qualification sampling period in 2n SYSCLKOUT cycles (i.e., at every 2n SYSCLKOUT cycles, the GPIO pin will be sampled).

B. The qualification period selected via the GPxCTRL register applies to groups of 8 GPIO pins.

C. The qualification block can take either three or six samples. The QUAL_SEL Register selects which sample mode is used.

D. In the example shown, for the qualifier to detect the change, the input should be stable for 10 SYSCLKOUT cycles or greater. In other words, the inputs should be stable for $(5 \times \text{QUALPRD} \times 2)$ SYSCLKOUT cycles. That would ensure 5 sampling periods for detection to occur. Since external signals are driven asynchronously, an 13-SYSCLKOUT-wide pulse ensures reliable recognition.

図 1-3. 入力クオリファイヤのクロック・サイクル

1.7 GPIO およびペリフェラルの多重化

1.7.1 GPIO 多重化

最大 12 の異なるペリフェラル機能が、汎用入出力 (GPIO) 機能とともに各ピンに多重化されます。これにより、特定のアプリケーションに最適なペリフェラルの組み合わせとピン配置を選択することができます。多重化の組み合わせと定義については、表 1-6 を参照してください。

表 1-6. GPIO 多重化ピン

0, 4, 8, 12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO0	EPWM1_A	CANA_RX	OUTPUTXBAR7	SCIA_RX	I2CA_SDA	SPIA_STE				EQEP1_INDEX		EPWM3_A	
GPIO1	EPWM1_B			SCIA_TX	I2CA_SCL	SPIA_SOMI	EQEP1_STROBE					EPWM3_B	
GPIO2	EPWM2_A			OUTPUTXBAR1		SPIA_SIMO	SCIA_TX		I2CB_SDA		CANA_TX	EPWM4_A	
GPIO3	EPWM2_B	OUTPUTXBAR2		OUTPUTXBAR2		SPIA_CLK	SCIA_RX		I2CB_SCL		CANA_RX	EPWM4_B	
GPIO4	EPWM3_A	I2CA_SCL		OUTPUTXBAR3	CANA_TX						SPIA_SOMI	EPWM1_A	
GPIO5	EPWM3_B	I2CA_SDA	OUTPUTXBAR3		CANA_RX	SPIA_STE			SCIA_RX			EPWM1_B	
GPIO6	EPWM4_A	OUTPUTXBAR4	SYNCOUT	EQEP1_A								EPWM2_A	
GPIO7	EPWM4_B	EPWM2_A	OUTPUTXBAR5	EQEP1_B		SPIA_SIMO			SCIA_TX		CANA_TX	EPWM2_B	
GPIO8	EPWM5_A		ADCSCAO	EQEP1_STROBE	SCIA_TX	SPIA_SIMO	I2CA_SCL						
GPIO9	EPWM5_B	SCIB_TX	OUTPUTXBAR6	EQEP1_INDEX	SCIA_RX	SPIA_CLK					I2CB_SCL		
GPIO10	EPWM6_A		ADCSCBO	EQEP1_A	SCIB_TX	SPIA_SOMI	I2CA_SDA						
GPIO11	EPWM6_B	CANA_RX	OUTPUTXBAR7	EQEP1_B	SCIB_RX	SPIA_STE				SPIA_SIMO			
GPIO12	EPWM7_A			EQEP1_STROBE	SCIB_TX				SPIA_CLK	CANA_RX			
GPIO13	EPWM7_B			EQEP1_INDEX	SCIB_RX				SPIA_SOMI	CANA_TX			
GPIO16	SPIA_SIMO		OUTPUTXBAR7	EPWM5_A	SCIA_TX		EQEP1_STROBE		XCLKOUT				
GPIO17	SPIA_SOMI		OUTPUTXBAR8	EPWM5_B	SCIA_RX		EQEP1_INDEX		CANA_TX		EPWM6_A		
GPIO18	SPIA_CLK	SCIB_TX	CANA_RX	EPWM6_A	I2CA_SCL				XCLKOUT				X2
GPIO19	SPIA_STE	SCIB_RX	CANA_TX	EPWM6_B	I2CA_SDA								X1
GPIO20	EQEP1_A		CANA_TX		SPIA_SIMO				I2CA_SCL			SCIC_TX	
GPIO21	EQEP1_B		CANA_RX		SPIA_SOMI				I2CA_SDA			SCIC_RX	
GPIO22	EQEP1_STROBE		SCIB_TX				SCIC_TX				EPWM4_A		
GPIO23	EQEP1_INDEX		SCIB_RX				SCIC_RX				EPWM4_B		
GPIO24	OUTPUTXBAR1		SPIA_STE	EPWM4_A	SPIA_SIMO				SCIA_TX	ERRORSTS			
GPIO28	SCIA_RX		EPWM7_A	OUTPUTXBAR5	EQEP1_A			SCIC_TX	SPIA_CLK	ERRORSTS	I2CB_SDA		
GPIO29	SCIA_TX		EPWM7_B	OUTPUTXBAR6	EQEP1_B			SCIC_RX	SPIA_STE	ERRORSTS	I2CB_SCL		
GPIO32	I2CA_SDA	EQEP1_INDEX	SPIA_CLK	EPWM4_B	SCIC_TX			CANA_TX		ADCSCBO			
GPIO33	I2CA_SCL			OUTPUTXBAR4	SCIC_RX			CANA_RX		ADCSCAO			
GPIO35	SCIA_RX	SPIA_SOMI	I2CA_SDA	CANA_RX		SCIC_RX	EQEP1_A		EPWM5_B			TDI	
GPIO37	OUTPUTXBAR2	SPIA_STE	I2CA_SCL	SCIA_TX	CANA_TX	SCIC_TX	EQEP1_B		EPWM5_A			TDO	
GPIO39										SYNCOUT	EQEP1_INDEX		
GPIO40				EPWM2_B			SCIB_TX	EQEP1_A					
GPIO41	EPWM7_A			EPWM2_A			SCIB_RX	EQEP1_B					
GPIO224				OUTPUTXBAR3	SPIA_SIMO		EPWM1_A	CANA_TX	EQEP1_A		SCIC_TX		
GPIO226				EPWM6_A	SPIA_CLK		EPWM1_B		EQEP1_STROBE		SCIC_RX		

表 1-6. GPIO 多重化ピン (続き)

0、4、8、12	1	2	3	5	6	7	9	10	11	13	14	15	ALT
GPIO227	I2CB_SCL		EPWM3_A	OUTPUTXBAR1	EPWM2_B								
GPIO228			ADCSOCAO	CANA_TX	SPIA_SOMI		EPWM2_B		EQEP1_B				
GPIO230	I2CB_SDA		EPWM3_B	CANA_RX	EPWM2_A	I2CA_SDA							
GPIO242				OUTPUTXBAR2	SPIA_STE		EPWM4_A	CANA_RX	EQEP1_INDEX				
AIO225													
AIO231													
AIO232													
AIO233													
AIO237													
AIO238													
AIO239													
AIO241													
AIO244													
AIO245													

1.7.2 ペリフェラルの多重化

たとえば、GPIO6 ピンの多重化は、GPAGMUX[13:12] および GPAMUX[13:12] に書き込むことで制御されます。これらのビットに書き込むことで、GPIO6 は汎用デジタル I/O またはいくつかの異なるペリフェラル機能の 1 つとして構成されます。表 1-7 に、GPyGMUX および GPyMUX の選択の例と、単一の GPIO に対するオプションを示します。

注

以下の表は一例です。このデバイスでは GPIO6 を利用できない可能性があります。GPIO6 が利用可能な場合、以下の表に記載されている機能は、実際に利用可能な機能と一致しない可能性があります。このデバイスの GPIO と対応する多重化オプションの正しいリストについては、このドキュメントの「GPIO 多重化」セクションを参照してください。

表 1-7. GPIO およびペリフェラルの多重化

GPAGMUX1[13:12]	GPAMUX1[13:12]	ピンの機能
00	00	GPIO6
00	01	ペリフェラル 1
00	10	ペリフェラル 2
00	11	ペリフェラル 3
01	00	GPIO6
01	01	ペリフェラル 4
01	10	ペリフェラル 5
01	11	
10	00	GPIO6
10	01	
10	10	ペリフェラル 6
10	11	ペリフェラル 7
11	00	GPIO6
11	01	ペリフェラル 8
11	10	ペリフェラル 9
11	11	ペリフェラル 10

デバイスの多重化方式は異なります。特定のデバイスでペリフェラルが利用できない場合、その多重化の選択はそのデバイスで予約されており、使用できません。

注意

ペリフェラル・モードにも GPIO モードにもマップされていない予約済み GPIO 多重化構成が選択されている場合、ピンの状態は未定義であり、ピンは駆動されます。未実装の構成は将来の拡張用であるため、選択しないでください。デバイス多重化テーブル (データ・マニュアルを参照) では、これらのオプションは予約済みと示されるか、空白のままになっています。

一部のペリフェラルは、マルチプレクサ・レジスタによって複数のピンに割り当てることができます。たとえば、OUTPUTXBAR1 は、個々のシステム要件に応じて、GPIO p、q、または r (ここで、p、q、および r は GPIO 番号の例) に割り当てることができます。表 1-8 にこの例を示します。

注

以下の表は一例です。ビット範囲が、このデバイスの OUTPUTXBAR1 に対応していない可能性があります。このデバイスの GPIO と対応する多重化オプションの正しいリストについては、このドキュメントの「GPIO 多重化」セクションを参照してください。

1 つも GPIO ピンがペリフェラル入力ピンとして構成されていない場合、または 1 つ以上の GPIO ピンがペリフェラル入力ピンとして構成されている場合、その GPIO はハードワイヤードのデフォルト値に設定されません。

表 1-8. ペリフェラルの多重化 (複数のピンが割り当て済み)

GMUX の構成	MUX の構成	
選択 1: GPIOp	GPyGMUX1[5:4]=01	GPyMUX1[5:4]=01
または選択 2: GPIOq	GPyGMUX2[17:16]=00	GPyMUX2[17:16]=01
または選択 3: GPIOr	GPyGMUX1[7:6]=01	GPyMUX1[7:6]=01

1.8 内部プルアップ構成要件

リセット時、GPIO は入力モードになり、内部プルアップはディセーブルになります。駆動されていない入力は、中間レール電圧まで変動し、入力バッファに無駄なシュートスルー電流が発生する可能性があります。ユーザーは、常に各 GPIO を次の構成のいずれかにする必要があります。

- 入力モードで、別のコンポーネントによって、基板上で V_{ih} より高いか V_{il} より低いレベルに駆動
- 入力モードで、GPIO 内部プルアップをイネーブル
- 出力モード

ピン数の少ないパッケージのデバイスでは、フローティング入力を防ぐために、ボンディングされていない GPIO のプルアップはデフォルトでイネーブルになっています。ユーザーは、アプリケーション・コードでこれらのプルアップをディセーブルにしないように注意する必要があります。

よりピン数の多いパッケージのデバイスでは、フローティング入力を防ぐために、内部でボンディングされていない GPIO のプルアップをイネーブルにする必要があります。テキサス・インスツルメンツは controlSUITE/C2000Ware で関数を提供しており、ユーザーはこの関数を呼び出して、使用しているパッケージのボンディングされていない GPIO のプルアップをイネーブルにできます。この関数 GPIO_EnabledUnbondedIOPullups() は、(Device)_SYSCtrl.c ファイルに存在し、デフォルトでは InitSysCtrl() から呼び出されます。ユーザーは、アプリケーション・コードでこれらのプルアップをディセーブルにしないように注意する必要があります。

1.9 ソフトウェア

1.9.1 GPIO の例

注：これらの例は、[C2000Ware](#) のインストール先の以下の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/gpio

これらの例へのクラウド・アクセスは、以下のリンクから利用できます：dev.ti.com [C2000Ware Examples](#)。

1.9.1.1 デバイス GPIO の設定

FILE : gpio_ex1_setup.c

デバイスの GPIO を 2 つの異なる構成にします。このコードは、GPIO の設定方法を説明するために冗長になっています。実際のアプリケーションでは、コードの行を組み合わせ、コード・サイズと効率を向上させることができます。

この例では GPIO のみをセットアップします。セットアップ後、実際にピンで行われることは何もありません。

一般的には、以下の通りです。

- すべてのプルアップ抵抗がイネーブルになります。ePWM の場合、これは望ましくない場合があります。
- 通信ポート (SPI、SCI、I2C) の入力認定は非同期です
- トリップ・ピン (TZ) の入力認定は非同期です
- eCAP および eQEP 信号の入力認定は SYSCLKOUT に同期します
- 一部の I/O および `__interrupts` の入力認定には、サンプリング・ウィンドウがある場合があります

1.9.1.2 デバイス GPIO トグル

FILE : gpio_ex2_toggle.c

sysconfig ファイルによってデバイスの GPIO を構成します。GPIO ピンは無限ループ内でトグルされます。

1.9.1.3 デバイス GPIO 割り込み

FILE : gpio_ex3_interrupt.c

sysconfig ファイルによってデバイスの GPIO を構成します。1 つの GPIO 出力ピンと 1 つの GPIO 入力ピンが構成されます。次に、この例では、GPIO 出力ピンをトグルする外部割り込みのソースとして GPIO 入力ピンを構成します。

1.9.1.4 外部割り込み (XINT)

FILE : gpio_ex4_aio_external_interrupt.c

この例では、AIO ピンがデジタル入力として構成されています。他の 2 つの GPIO 信号 (AIO ピンに外部接続) はソフトウェアでトグルされ、AIO224 および AIO225 (XINT1 に割り当てられた AIO224 と XINT2 に割り当てられた AIO225) を介して外部割り込みをトリガします。プログラムを正常に動作させるには、これらの信号を外に接続する必要があります。各割り込みは順番に発生します。最初に XINT1、次に XINT2 です。

- GPIO5 は、割り込みの外部では High になり、割り込みの内部では Low になります。この信号はスコープで監視できます。外部接続
- GPIO0 を AIO224 に接続します。AIO224 は XINT1 に割り当てられます
- GPIO1 を AIO225 に接続します。AIO225 は XINT2 に割り当てられます
- GPIO5 はオシロスコープで監視できます。

監視変数

- XINT1 割り込みの回数である xint1Count
- XINT2 割り込みの回数である xint2Count
- アイドル・ループの回数である loopCount

1.9.2 LED の例

注：これらの例は、[C2000Ware](#) のインストール先の以下の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/led

これらの例へのクラウド・アクセスは、以下のリンクから利用できます：dev.ti.com [C2000Ware Examples](#)。

1.9.2.1 LED 点滅の例

FILE : led_ex1_blinky.c

この例では、LED を点滅させる方法を示します。

外部接続

- なし。

監視変数

- なし。

1.10 GPIO レジスタ

このセクションでは、汎用入出力レジスタについて説明します。

1.10.1 GPIO ベース・アドレスの表

表 1-9. GPIO ベース・アドレスの表

ビット・フィールド名		DriverLib 名	ベース・アドレス	パイプライン保護
インスタンス	構造			
GpioCtrlRegs	GPIO_CTRL_REGS	GPIOCTRL_BASE	0x0000_7C00	あり
GpioDataRegs	GPIO_DATA_REGS	GPIODATA_BASE	0x0000_7F00	あり
GpioDataReadRegs	GPIO_DATA_READ_REGS	GPIODATAREAD_BASE	0x0000_7F80	あり

1.10.2 GPIO_CTRL_REGS レジスタ

GPIO_CTRL_REGS レジスタにメモリマップされたレジスタを、表 1-10 に示します。表 1-10 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-10. GPIO_CTRL_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	GPACTRL	GPIO A 認定サンプリング期間制御 (GPIO0 ~ 31)	EALLOW	表示
2h	GPAQSEL1	GPIO A 修飾子選択 1 レジスタ (GPIO0 ~ 15)	EALLOW	表示
4h	GPAQSEL2	GPIO A 修飾子選択 2 レジスタ (GPIO16 ~ 31)	EALLOW	表示
6h	GPAMUX1	GPIO A マルチプレクサ 1 レジスタ (GPIO0 ~ 15)	EALLOW	表示
8h	GPAMUX2	GPIO A マルチプレクサ 2 レジスタ (GPIO16 ~ 31)	EALLOW	表示
Ah	GPADIR	GPIO A 方向レジスタ (GPIO0 ~ 31)	EALLOW	表示
Ch	GPAPUD	GPIO A プルアップ・ディセーブル・レジスタ (GPIO0 ~ 31)	EALLOW	表示
10h	GPAINV	GPIO A 入力極性反転レジスタ (GPIO0 ~ 31)	EALLOW	表示
12h	GPAODR	GPIO A オープン・ドレイン出力レジスタ (GPIO0 ~ GPIO31)	EALLOW	表示
14h	GPAAMSEL	GPIO A アナログ・モード選択レジスタ (GPIO0 ~ GPIO31)	EALLOW	表示
20h	GPAGMUX1	GPIO A ベリフェラル・グループ・マルチプレクサ (GPIO0 ~ 15)	EALLOW	表示
22h	GPAGMUX2	GPIO A ベリフェラル・グループ・マルチプレクサ (GPIO16 ~ 31)	EALLOW	表示
3Ch	GPALOCK	GPIO A ロック構成レジスタ (GPIO0 ~ 31)	EALLOW	表示
3Eh	GPACR	GPIO A ロック・コミット・レジスタ (GPIO0 ~ 31)	EALLOW	表示
40h	GPBCTRL	GPIO B 認定サンプリング期間制御 (GPIO32 ~ 63)	EALLOW	表示
42h	GPBQSEL1	GPIO B フィルタ選択 1 レジスタ (GPIO32 ~ 47)	EALLOW	表示
46h	GPBMUX1	GPIO B マルチプレクサ 1 レジスタ (GPIO32 ~ 47)	EALLOW	表示
4Ah	GPBDIR	GPIO B 方向レジスタ (GPIO32 ~ 63)	EALLOW	表示
4Ch	GPBPUD	GPIO B プルアップ・ディセーブル・レジスタ (GPIO32 ~ 63)	EALLOW	表示
50h	GPBINV	GPIO B 入力極性反転レジスタ (GPIO32 ~ 63)	EALLOW	表示
52h	GPBODR	GPIO B オープン・ドレイン出力レジスタ (GPIO32 ~ GPIO63)	EALLOW	表示
60h	GPBGMUX1	GPIO B ベリフェラル・グループ・マルチプレクサ (GPIO32 ~ 47)	EALLOW	表示
7Ch	GPBLOCK	GPIO B ロック構成レジスタ (GPIO32 ~ 63)	EALLOW	表示
7Eh	GPBCR	GPIO B ロック・コミット・レジスタ (GPIO32 ~ 63)	EALLOW	表示
1C0h	GPHCTRL	GPIO H 認定サンプリング期間制御 (GPIO224 ~ 255)	EALLOW	表示
1C2h	GPHQSEL1	GPIO H 修飾子選択 1 レジスタ (GPIO224 ~ 239)	EALLOW	表示
1C4h	GPHQSEL2	GPIO H 修飾子選択 2 レジスタ (GPIO240 ~ 255)	EALLOW	表示
1C6h	GPHMUX1	GPIO H マルチプレクサ 1 レジスタ (GPIO224 ~ 239)	EALLOW	表示
1C8h	GPHMUX2	GPIO H マルチプレクサ 2 レジスタ (GPIO240 ~ 255)	EALLOW	表示

表 1-10. GPIO_CTRL_REGS レジスタ (続き)

オフセット	略称	レジスタ名	書き込み保護	セクション
1CAh	GPHDIR	GPIO H 方向レジスタ (GPIO224 ~ 255)	EALLOW	表示
1CCh	GPHPUD	GPIO H プルアップ・ディセーブル・レジスタ (GPIO224 ~ 255)	EALLOW	表示
1D0h	GPHINV	GPIO H 入力極性反転レジスタ (GPIO224 ~ 255)	EALLOW	表示
1D2h	GPHODR	GPIO H オープン・ドレイン出力レジスタ (GPIO224 ~ GPIO255)	EALLOW	表示
1D4h	GPHAMSEL	GPIO H アナログ・モード選択レジスタ (GPIO224 ~ GPIO255)	EALLOW	表示
1E0h	GPHGMUX1	GPIO H ベリフェラル・グループ・マルチプレクサ (GPIO224 ~ 239)	EALLOW	表示
1E2h	GPHGMUX2	GPIO H ベリフェラル・グループ・マルチプレクサ (GPIO240 ~ 255)	EALLOW	表示
1FCh	GPHLOCK	GPIO H ロック構成レジスタ (GPIO224 ~ 255)	EALLOW	表示
1FEh	GPHCR	GPIO H ロック・コミット・レジスタ (GPIO224 ~ 255)	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。このセクションでアクセス・タイプに使用しているコードを、[表 1-11](#) に示します。

表 1-11. GPIO_CTRL_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
書き込みタイプ		
W	W	書き込み
WSonce	W Sonce	書き込み 1 回設定
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列の変数		
i, j, k, l, m, n		レジスタ名、オフセット、またはアドレスでこれらの変数を使用すると、レジスタ配列の値を参照することになり、ここではレジスタが反復レジスタ・グループの一部になります。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		レジスタ名、オフセット、またはアドレスでこの変数を使用すると、レジスタ配列の値を参照することになります。

1.10.2.1 GPACTRL レジスタ (オフセット = 0h) [リセット = 0h]

GPACTRL を [図 1-4](#) に示します。説明については、[表 1-12](#) を参照してください。

[概略表](#)に戻ります。

GPIO A 認定サンプリング期間制御 (GPIO0 ~ 31)

図 1-4. GPACTRL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUALPRD3								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 1-12. GPACTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-24	QUALPRD3	R/W	0h	GPIO24 から GPIO31 までの認定サンプリング期間 : 0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ : SYSRSn
23-16	QUALPRD2	R/W	0h	GPIO16 から GPIO23 までの認定サンプリング期間 : 0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ : SYSRSn
15-8	QUALPRD1	R/W	0h	GPIO8 から GPIO15 までの認定サンプリング期間 : 0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ : SYSRSn
7-0	QUALPRD0	R/W	0h	GPIO0 から GPIO7 までの認定サンプリング期間 : 0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ : SYSRSn

1.10.2.2 GPAQSEL1 レジスタ (オフセット = 2h) [リセット = 0h]

GPAQSEL1 を [図 1-5](#) に示します。説明については、[表 1-13](#) を参照してください。

[概略表](#)に戻ります。

GPIO A 修飾子選択 1 レジスタ (GPIO0 ~ 15)

入力認定タイプ :

0、0 同期

0、1 認定 (3 サンプル)

1、0 認定 (6 サンプル)

1、1 非同期 (同期または認定なし)

図 1-5. GPAQSEL1 レジスタ

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

表 1-13. GPAQSEL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	GPIO13	R/W	0h	GPIO13 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
25-24	GPIO12	R/W	0h	GPIO12 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
23-22	GPIO11	R/W	0h	GPIO11 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
21-20	GPIO10	R/W	0h	GPIO10 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
19-18	GPIO9	R/W	0h	GPIO9 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn

表 1-13. GPAQSEL1 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
17-16	GPIO8	R/W	0h	GPIO8 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
15-14	GPIO7	R/W	0h	GPIO7 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
13-12	GPIO6	R/W	0h	GPIO6 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
11-10	GPIO5	R/W	0h	GPIO5 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
9-8	GPIO4	R/W	0h	GPIO4 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
7-6	GPIO3	R/W	0h	GPIO3 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
5-4	GPIO2	R/W	0h	GPIO2 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
3-2	GPIO1	R/W	0h	GPIO1 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
1-0	GPIO0	R/W	0h	GPIO0 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn

1.10.2.3 GPAQSEL2 レジスタ (オフセット = 4h) [リセット = 0h]

GPAQSEL2 を [図 1-6](#) に示します。説明については、[表 1-14](#) を参照してください。

[概略表](#)に戻ります。

GPIO A 修飾子選択 2 レジスタ (GPIO16 ~ 31)

入力認定タイプ :

0、0 同期

0、1 認定 (3 サンプル)

1、0 認定 (6 サンプル)

1、1 非同期 (同期または認定なし)

図 1-6. GPAQSEL2 レジスタ

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

表 1-14. GPAQSEL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	GPIO29	R/W	0h	GPIO29 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
25-24	GPIO28	R/W	0h	GPIO28 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	GPIO24	R/W	0h	GPIO24 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
15-14	GPIO23	R/W	0h	GPIO23 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
13-12	GPIO22	R/W	0h	GPIO22 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn

表 1-14. GPAQSEL2 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
11-10	GPIO21	R/W	0h	GPIO21 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
9-8	GPIO20	R/W	0h	GPIO20 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
7-6	GPIO19	R/W	0h	GPIO19 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
5-4	GPIO18	R/W	0h	GPIO18 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
3-2	GPIO17	R/W	0h	GPIO17 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
1-0	GPIO16	R/W	0h	GPIO16 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn

1.10.2.4 GPAMUX1 レジスタ (オフセット = 6h) [リセット = 0h]

GPAMUX1 を [図 1-7](#) に示します。説明については、[表 1-15](#) を参照してください。

[概略表](#)に戻ります。

GPIO A マルチプレクサ 1 レジスタ (GPIO0~15)

GPIO のピン多重化の選択を定義します。

注：

中間ペリフェラルの選択が GPIO にマップされないよう、このレジスタの前に対応する GPyGMUXn.GPIOz を構成する必要があります。詳細については、GPIO の章を参照してください。

図 1-7. GPAMUX1 レジスタ

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

表 1-15. GPAMUX1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	GPIO13	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
25-24	GPIO12	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
23-22	GPIO11	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
21-20	GPIO10	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
19-18	GPIO9	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
17-16	GPIO8	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
15-14	GPIO7	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
13-12	GPIO6	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
11-10	GPIO5	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
9-8	GPIO4	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
7-6	GPIO3	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	GPIO2	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
3-2	GPIO1	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
1-0	GPIO0	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.5 GPAMUX2 レジスタ (オフセット = 8h) [リセット = 0h]

GPAMUX2 を図 1-8 に示します。説明については、表 1-16 を参照してください。

概略表に戻ります。

GPIO A マルチプレクサ 2 レジスタ (GPIO16~31)

GPIO のピン多重化の選択を定義します。

注：

中間ペリフェラルの選択が GPIO にマップされないよう、このレジスタの前に対応する GPyGMUXn.GPIOz を構成する必要があります。詳細については、GPIO の章を参照してください。

図 1-8. GPAMUX2 レジスタ

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

表 1-16. GPAMUX2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	GPIO29	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
25-24	GPIO28	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	GPIO24	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
15-14	GPIO23	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
13-12	GPIO22	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
11-10	GPIO21	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
9-8	GPIO20	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
7-6	GPIO19	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	GPIO18	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
3-2	GPIO17	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
1-0	GPIO16	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.6 GPADIR レジスタ (オフセット= Ah) [リセット= 0h]

GPADIR を図 1-9 に示します。説明については、表 1-17 を参照してください。

概略表に戻ります。

GPIO A 方向レジスタ (GPIO0~31)

指定されたピンが GPIO モードで構成されているとき、GPIO ピンの方向を制御します。

0 : ピンを入力として構成します。

1 : ピンを出力として構成します。

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

図 1-9. GPADIR レジスタ

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

表 1-17. GPADIR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	GPIO29	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
28	GPIO28	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	GPIO24	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
23	GPIO23	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
22	GPIO22	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
21	GPIO21	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
20	GPIO20	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
19	GPIO19	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
18	GPIO18	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn
17	GPIO17	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ: SYSRSn

表 1-17. GPADIR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
16	GPIO16	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	GPIO13	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
12	GPIO12	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
11	GPIO11	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
10	GPIO10	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
9	GPIO9	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
8	GPIO8	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
7	GPIO7	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
6	GPIO6	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
5	GPIO5	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
4	GPIO4	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
3	GPIO3	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
2	GPIO2	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
1	GPIO1	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
0	GPIO0	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn

1.10.2.7 GPAPUD レジスタ (オフセット = Ch) [リセット = FFFFFFFFh]

GPAPUD を図 1-10 に示します。説明については、表 1-18 を参照してください。

概略表に戻ります。

GPIO A プルアップ・ディセーブル・レジスタ (GPIO0~31)

は、GPIO のプルアップをディセーブルします。

0 : プルアップをイネーブルします。

1 : プルアップをディセーブルします。

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。

図 1-10. GPAPUD レジスタ

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

表 1-18. GPAPUD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	1h	予約済み
30	予約済み	R/W	1h	予約済み
29	GPIO29	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
28	GPIO28	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
27	予約済み	R/W	1h	予約済み
26	予約済み	R/W	1h	予約済み
25	予約済み	R/W	1h	予約済み
24	GPIO24	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
23	GPIO23	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
22	GPIO22	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
21	GPIO21	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
20	GPIO20	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
19	GPIO19	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn
18	GPIO18	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ : SYSRSn

表 1-18. GPAPUD レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
17	GPIO17	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
16	GPIO16	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
15	予約済み	R/W	1h	予約済み
14	予約済み	R/W	1h	予約済み
13	GPIO13	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
12	GPIO12	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
11	GPIO11	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
10	GPIO10	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
9	GPIO9	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
8	GPIO8	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
7	GPIO7	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
6	GPIO6	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
5	GPIO5	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
4	GPIO4	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
3	GPIO3	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
2	GPIO2	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
1	GPIO1	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn
0	GPIO0	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ：SYSRSn

1.10.2.8 GPAINV レジスタ (オフセット = 10h) [リセット = 0h]

GPAINV を [図 1-11](#) に示します。説明については、[表 1-19](#) を参照してください。

[概略表](#)に戻ります。

GPIO A 入力極性反転レジスタ (GPIO0 ~ 31)

デバイスへの非反転または反転のいずれかの GPIO 入力を選択します。

0 : 非反転 GPIO 入力を選択

1 : 反転 GPIO 入力を選択

レジスタを読み取ると、レジスタ設定の現在の値が返されます。

図 1-11. GPAINV レジスタ

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

表 1-19. GPAINV レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	GPIO29	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
28	GPIO28	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	GPIO24	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
23	GPIO23	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
22	GPIO22	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
21	GPIO21	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
20	GPIO20	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
19	GPIO19	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
18	GPIO18	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn
17	GPIO17	R/W	0h	このピンの入力反転の制御 リセット・タイプ : SYSRSn

表 1-19. GPAINV レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
16	GPIO16	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	GPIO13	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
12	GPIO12	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
11	GPIO11	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
10	GPIO10	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
9	GPIO9	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
8	GPIO8	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
7	GPIO7	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
6	GPIO6	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
5	GPIO5	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
4	GPIO4	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
3	GPIO3	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
2	GPIO2	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
1	GPIO1	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
0	GPIO0	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn

1.10.2.9 GPAODR レジスタ (オフセット = 12h) [リセット = 0h]

GPAODR を [図 1-12](#) に示します。説明については、[表 1-20](#) を参照してください。

[概略表](#)に戻ります。

GPIO A オープン・ドレイン出力レジスタ (GPIO0 ~ GPIO31)

GPIO ピンの通常出力とオープン・ドレイン出力のいずれかを選択します。

0 : 通常出力

1 : オープン・ドレイン出力

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] オープン・ドレイン出力モードで、バッファが出力モードに構成されていると、0 の値は PAD 上の駆動で出力され、1 の値はバッファのトライステートの駆動で出力されます。

図 1-12. GPAODR レジスタ

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

表 1-20. GPAODR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	GPIO29	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
28	GPIO28	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	GPIO24	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
23	GPIO23	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
22	GPIO22	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
21	GPIO21	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
20	GPIO20	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
19	GPIO19	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn

表 1-20. GPAODR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
18	GPIO18	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
17	GPIO17	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
16	GPIO16	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	GPIO13	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
12	GPIO12	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
11	GPIO11	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
10	GPIO10	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
9	GPIO9	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
8	GPIO8	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
7	GPIO7	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
6	GPIO6	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
5	GPIO5	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
4	GPIO4	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
3	GPIO3	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
2	GPIO2	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
1	GPIO1	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
0	GPIO0	R/W	0h	このピン出力オープン・ドレインの制御 リセット・タイプ: SYSRSn

1.10.2.10 GPAAMSEL レジスタ (オフセット = 14h) [リセット = FFFFFFFFh]

GPAAMSEL を [図 1-13](#) に示します。説明については、[表 1-21](#) を参照してください。

[概略表](#)に戻ります。

GPIO A アナログ・モード選択レジスタ (GPIO0 ~ GPIO31)

GPIO ピンのデジタル機能とアナログ機能のいずれかを選択します。

0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できません

1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できます

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは影響を与えません。

図 1-13. GPAAMSEL レジスタ

31	30	29	28	27	26	25	24
予約済み	予約済み	予約済み	GPIO28	予約済み	予約済み	予約済み	予約済み
R/W-1h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO21	GPIO20	予約済み	予約済み	予約済み	予約済み
R/W-1h							
15	14	13	12	11	10	9	8
予約済み	予約済み	GPIO13	GPIO12	予約済み	予約済み	予約済み	予約済み
R/W-1h							
7	6	5	4	3	2	1	0
予約済み							
R/W-1h							

表 1-21. GPAAMSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	1h	予約済み
30	予約済み	R/W	1h	予約済み
29	予約済み	R/W	1h	予約済み
28	GPIO28	R/W	1h	このピンのアナログ・モード選択 リセット・タイプ : SYSRSn
27	予約済み	R/W	1h	予約済み
26	予約済み	R/W	1h	予約済み
25	予約済み	R/W	1h	予約済み
24	予約済み	R/W	1h	予約済み
23	予約済み	R/W	1h	予約済み
22	予約済み	R/W	1h	予約済み
21	GPIO21	R/W	1h	このピンのアナログ・モード選択 リセット・タイプ : SYSRSn
20	GPIO20	R/W	1h	このピンのアナログ・モード選択 リセット・タイプ : SYSRSn
19	予約済み	R/W	1h	予約済み
18	予約済み	R/W	1h	予約済み
17	予約済み	R/W	1h	予約済み
16	予約済み	R/W	1h	予約済み

表 1-21. GPAAMSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
15	予約済み	R/W	1h	予約済み
14	予約済み	R/W	1h	予約済み
13	GPIO13	R/W	1h	このピンのアナログ・モード選択 リセット・タイプ: SYSRSn
12	GPIO12	R/W	1h	このピンのアナログ・モード選択 リセット・タイプ: SYSRSn
11	予約済み	R/W	1h	予約済み
10	予約済み	R/W	1h	予約済み
9	予約済み	R/W	1h	予約済み
8	予約済み	R/W	1h	予約済み
7	予約済み	R/W	1h	予約済み
6	予約済み	R/W	1h	予約済み
5	予約済み	R/W	1h	予約済み
4	予約済み	R/W	1h	予約済み
3	予約済み	R/W	1h	予約済み
2	予約済み	R/W	1h	予約済み
1	予約済み	R/W	1h	予約済み
0	予約済み	R/W	1h	予約済み

1.10.2.11 GPAGMUX1 レジスタ (オフセット = 20h) [リセット = 0h]

GPAGMUX1 を図 1-14 に示します。説明については、表 1-22 を参照してください。

概略表に戻ります。

GPIO A ペリフェラル・グループ・マルチプレクサ (GPIO0 ~ 15)

GPIO のピン多重化の選択を定義します。

注：

[1] GPIOx のピン多重化の完全な選択を行うには、GPAMUXy.GPIOx も構成する必要があります。

図 1-14. GPAGMUX1 レジスタ

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

表 1-22. GPAGMUX1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	GPIO13	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
25-24	GPIO12	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
23-22	GPIO11	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
21-20	GPIO10	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
19-18	GPIO9	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
17-16	GPIO8	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
15-14	GPIO7	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
13-12	GPIO6	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
11-10	GPIO5	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
9-8	GPIO4	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
7-6	GPIO3	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	GPIO2	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
3-2	GPIO1	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
1-0	GPIO0	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.12 GPAGMUX2 レジスタ (オフセット = 22h) [リセット = 0h]

GPAGMUX2 を図 1-15 に示します。説明については、表 1-23 を参照してください。

概略表に戻ります。

GPIO A ペリフェラル・グループ・マルチプレクサ (GPIO16 ~ 31)

GPIO のピン多重化の選択を定義します。

注：

[1] GPIOx のピン多重化の完全な選択を行うには、GPAMUXy.GPIOx も構成する必要があります。

図 1-15. GPAGMUX2 レジスタ

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

表 1-23. GPAGMUX2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	GPIO29	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
25-24	GPIO28	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	GPIO24	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
15-14	GPIO23	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
13-12	GPIO22	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
11-10	GPIO21	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
9-8	GPIO20	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
7-6	GPIO19	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	GPIO18	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
3-2	GPIO17	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
1-0	GPIO16	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.13 GPALOCK レジスタ (オフセット = 3Ch) [リセット = 0h]

GPALOCK を図 1-16 に示します。説明については、表 1-24 を参照してください。

概略表に戻ります。

GPIO A ロック構成レジスタ (GPIO0~31)

GPIO 用の GPIO 構成ロック。

0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx レジスタで、同じピンを制御するビットを変更可能

1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる

図 1-16. GPALOCK レジスタ

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

表 1-24. GPALOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	GPIO29	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
28	GPIO28	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	GPIO24	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
23	GPIO23	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
22	GPIO22	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
21	GPIO21	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
20	GPIO20	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
19	GPIO19	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
18	GPIO18	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn

表 1-24. GPALOCK レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
17	GPIO17	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
16	GPIO16	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	GPIO13	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
12	GPIO12	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
11	GPIO11	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
10	GPIO10	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
9	GPIO9	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
8	GPIO8	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
7	GPIO7	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
6	GPIO6	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
5	GPIO5	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
4	GPIO4	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
3	GPIO3	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
2	GPIO2	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
1	GPIO1	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn
0	GPIO0	R/W	0h	このピンの構成ロック・ビット リセット・タイプ：SYSRSn

1.10.2.14 GPACR レジスタ (オフセット = 3Eh) [リセット = 0h]

GPACR を図 1-17 に示します。説明については、表 1-25 を参照してください。

概略表に戻ります。

GPIO A ロック・コミット・レジスタ (GPIO0 ~ 31)

GPIO 用の GPIO 構成ロック・コミット :

1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック

0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能

図 1-17. GPACR レジスタ

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

表 1-25. GPACR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/WOnce	0h	予約済み
30	予約済み	R/WOnce	0h	予約済み
29	GPIO29	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
28	GPIO28	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
27	予約済み	R/WOnce	0h	予約済み
26	予約済み	R/WOnce	0h	予約済み
25	予約済み	R/WOnce	0h	予約済み
24	GPIO24	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
23	GPIO23	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
22	GPIO22	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
21	GPIO21	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
20	GPIO20	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
19	GPIO19	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
18	GPIO18	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn
17	GPIO17	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ : SYSRSn

表 1-25. GPACR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
16	GPIO16	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
15	予約済み	R/WOnce	0h	予約済み
14	予約済み	R/WOnce	0h	予約済み
13	GPIO13	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
12	GPIO12	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
11	GPIO11	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
10	GPIO10	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
9	GPIO9	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
8	GPIO8	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
7	GPIO7	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
6	GPIO6	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
5	GPIO5	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
4	GPIO4	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
3	GPIO3	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
2	GPIO2	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
1	GPIO1	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
0	GPIO0	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn

1.10.2.15 GPBCTRL レジスタ (オフセット = 40h) [リセット = 0h]

GPBCTRL を [図 1-18](#) に示します。説明については、[表 1-26](#) を参照してください。

[概略表](#)に戻ります。

GPIO B 認定サンプリング期間制御 (GPIO32 ~ 63)

図 1-18. GPBCTRL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								予約済み								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 1-26. GPBCTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-24	予約済み	R/W	0h	予約済み
23-16	予約済み	R/W	0h	予約済み
15-8	QUALPRD1	R/W	0h	GPIO40 から GPIO47 までの認定サンプリング期間： 0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ：SYSRSn
7-0	QUALPRD0	R/W	0h	GPIO32 から GPIO39 までの認定サンプリング期間： 0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ：SYSRSn

1.10.2.16 GPBQSEL1 レジスタ (オフセット = 42h) [リセット = CC0h]

GPBQSEL1 を [図 1-19](#) に示します。説明については、[表 1-27](#) を参照してください。

[概略表](#)に戻ります。

GPIO B 修飾子選択 1 レジスタ (GPIO32 ~ 47)

入力認定タイプ :

0、0 同期

0、1 認定 (3 サンプル)

1、0 認定 (6 サンプル)

1、1 非同期 (同期または認定なし)

図 1-19. GPBQSEL1 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み		予約済み		予約済み		予約済み		予約済み		予約済み		GPIO41		GPIO40	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39		予約済み		GPIO37		予約済み		GPIO35		予約済み		GPIO33		GPIO32	
R/W-0h		R/W-0h		R/W-3h		R/W-0h		R/W-3h		R/W-0h		R/W-0h		R/W-0h	

表 1-27. GPBQSEL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	GPIO41	R/W	0h	GPIO41 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
17-16	GPIO40	R/W	0h	GPIO40 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
15-14	GPIO39	R/W	0h	GPIO39 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
13-12	予約済み	R/W	0h	予約済み
11-10	GPIO37	R/W	3h	GPIO37 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
9-8	予約済み	R/W	0h	予約済み

表 1-27. GPBQSEL1 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
7-6	GPIO35	R/W	3h	GPIO35 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
5-4	予約済み	R/W	0h	予約済み
3-2	GPIO33	R/W	0h	GPIO33 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
1-0	GPIO32	R/W	0h	GPIO32 の入力認定タイプの選択 : 0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn

1.10.2.17 GPBMUX1 レジスタ (オフセット = 46h) [リセット = CC0h]

GPBMUX1 を図 1-20 に示します。説明については、表 1-28 を参照してください。

概略表に戻ります。

GPIO B マルチプレクサ 1 レジスタ (GPIO32 ~ 47)

GPIO のピン多重化の選択を定義します。

注：

中間ペリフェラルの選択が GPIO にマップされないよう、このレジスタの前に対応する GPyGMUXn.GPIOz を構成する必要があります。詳細については、GPIO の章を参照してください。

図 1-20. GPBMUX1 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み	GPIO41	GPIO40													
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37		予約済み	GPIO35		予約済み	GPIO33		GPIO32					
R/W-0h	R/W-0h	R/W-3h		R/W-0h	R/W-3h		R/W-0h	R/W-0h		R/W-0h		R/W-0h			

表 1-28. GPBMUX1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	GPIO41	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
17-16	GPIO40	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
15-14	GPIO39	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
13-12	予約済み	R/W	0h	予約済み
11-10	GPIO37	R/W	3h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
9-8	予約済み	R/W	0h	予約済み
7-6	GPIO35	R/W	3h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	予約済み	R/W	0h	予約済み
3-2	GPIO33	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
1-0	GPIO32	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.18 GPBDIR レジスタ (オフセット = 4Ah) [リセット = 0h]

GPBDIR を図 1-21 に示します。説明については、表 1-29 を参照してください。

概略表に戻ります。

GPIO B 方向レジスタ (GPIO32 ~ 63)

指定されたピンが GPIO モードで構成されているとき、GPIO ピンの方向を制御します。

0 : ピンを入力として構成します。

1 : ピンを出力として構成します。

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

図 1-21. GPBDIR レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/W-0h							

表 1-29. GPBDIR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	予約済み	R/W	0h	予約済み
20	予約済み	R/W	0h	予約済み
19	予約済み	R/W	0h	予約済み
18	予約済み	R/W	0h	予約済み
17	予約済み	R/W	0h	予約済み
16	予約済み	R/W	0h	予約済み
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	予約済み	R/W	0h	予約済み
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み
10	予約済み	R/W	0h	予約済み

表 1-29. GPBDIR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO41	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
8	GPIO40	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
7	GPIO39	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
6	予約済み	R/W	0h	予約済み
5	GPIO37	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
4	予約済み	R/W	0h	予約済み
3	GPIO35	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
2	予約済み	R/W	0h	予約済み
1	GPIO33	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
0	GPIO32	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn

1.10.2.19 GPBPUD レジスタ (オフセット = 4Ch) [リセット = FFFFFFFFh]

GPBPUD を図 1-22 に示します。説明については、表 1-30 を参照してください。

概略表に戻ります。

GPIO B プルアップ・ディセーブル・レジスタ (GPIO32 ~ 63)

GPIO のプルアップをディセーブルします。

0 : プルアップをイネーブルします。

1 : プルアップをディセーブルします。

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。

図 1-22. GPBPUD レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-1h							
23	22	21	20	19	18	17	16
予約済み							
R/W-1h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/W-1h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/W-1h							

表 1-30. GPBPUD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	1h	予約済み
30	予約済み	R/W	1h	予約済み
29	予約済み	R/W	1h	予約済み
28	予約済み	R/W	1h	予約済み
27	予約済み	R/W	1h	予約済み
26	予約済み	R/W	1h	予約済み
25	予約済み	R/W	1h	予約済み
24	予約済み	R/W	1h	予約済み
23	予約済み	R/W	1h	予約済み
22	予約済み	R/W	1h	予約済み
21	予約済み	R/W	1h	予約済み
20	予約済み	R/W	1h	予約済み
19	予約済み	R/W	1h	予約済み
18	予約済み	R/W	1h	予約済み
17	予約済み	R/W	1h	予約済み
16	予約済み	R/W	1h	予約済み
15	予約済み	R/W	1h	予約済み
14	予約済み	R/W	1h	予約済み
13	予約済み	R/W	1h	予約済み
12	予約済み	R/W	1h	予約済み

表 1-30. GPBPUD レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
11	予約済み	R/W	1h	予約済み
10	予約済み	R/W	1h	予約済み
9	GPIO41	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn
8	GPIO40	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn
7	GPIO39	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn
6	予約済み	R/W	1h	予約済み
5	GPIO37	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn
4	予約済み	R/W	1h	予約済み
3	GPIO35	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn
2	予約済み	R/W	1h	予約済み
1	GPIO33	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn
0	GPIO32	R/W	1h	このピンのプルアップのディセーブル制御 リセット・タイプ: SYSRSn

1.10.2.20 GPBINV レジスタ (オフセット = 50h) [リセット = 0h]

GPBINV を図 1-23 に示します。説明については、表 1-31 を参照してください。

概略表に戻ります。

GPIO B 入力極性反転レジスタ (GPIO32 ~ 63)

デバイスへの非反転または反転のいずれかの GPIO 入力を選択します。

0 : 非反転 GPIO 入力を選択

1 : 反転 GPIO 入力を選択

レジスタを読み取ると、レジスタ設定の現在の値が返されます。

図 1-23. GPBINV レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/W-0h							

表 1-31. GPBINV レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	予約済み	R/W	0h	予約済み
20	予約済み	R/W	0h	予約済み
19	予約済み	R/W	0h	予約済み
18	予約済み	R/W	0h	予約済み
17	予約済み	R/W	0h	予約済み
16	予約済み	R/W	0h	予約済み
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	予約済み	R/W	0h	予約済み
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み
10	予約済み	R/W	0h	予約済み

表 1-31. GPBINV レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO41	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
8	GPIO40	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
7	GPIO39	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
6	予約済み	R/W	0h	予約済み
5	GPIO37	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
4	予約済み	R/W	0h	予約済み
3	GPIO35	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
2	予約済み	R/W	0h	予約済み
1	GPIO33	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn
0	GPIO32	R/W	0h	このピンの入力反転の制御 リセット・タイプ：SYSRSn

1.10.2.21 GPBODR レジスタ (オフセット = 52h) [リセット = 0h]

GPBODR を図 1-24 に示します。説明については、表 1-32 を参照してください。

概略表に戻ります。

GPIO B オープン・ドレイン出力レジスタ (GPIO32 ~ GPIO63)

GPIO ピンの通常出力とオープン・ドレイン出力のいずれかを選択します。

0 : 通常出力

1 : オープン・ドレイン出力

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] オープン・ドレイン出力モードで、バッファが出力モードに構成されていると、0 の値は PAD 上の駆動で出力され、1 の値はバッファのトライステートの駆動で出力されます。

図 1-24. GPBODR レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/W-0h							

表 1-32. GPBODR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	予約済み	R/W	0h	予約済み
20	予約済み	R/W	0h	予約済み
19	予約済み	R/W	0h	予約済み
18	予約済み	R/W	0h	予約済み
17	予約済み	R/W	0h	予約済み
16	予約済み	R/W	0h	予約済み
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	予約済み	R/W	0h	予約済み

表 1-32. GPBODR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み
10	予約済み	R/W	0h	予約済み
9	GPIO41	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
8	GPIO40	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
7	GPIO39	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
6	予約済み	R/W	0h	予約済み
5	GPIO37	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
4	予約済み	R/W	0h	予約済み
3	GPIO35	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
2	予約済み	R/W	0h	予約済み
1	GPIO33	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
0	GPIO32	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn

1.10.2.22 GPBGMUX1 レジスタ (オフセット = 60h) [リセット = CC0h]

GPBGMUX1 を図 1-25 に示します。説明については、表 1-33 を参照してください。

概略表に戻ります。

GPIO B ペリフェラル・グループ・マルチプレクサ (GPIO32 ~ 47)

GPIO のピン多重化の選択を定義します。

注：

[1] GPIOx のピン多重化の完全な選択を行うには、GPAMUXy.GPIOx も構成する必要があります。

図 1-25. GPBGMUX1 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
予約済み	GPIO41	GPIO40													
R/W-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32								
R/W-0h	R/W-0h	R/W-3h	R/W-0h	R/W-3h	R/W-0h										

表 1-33. GPBGMUX1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	GPIO41	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
17-16	GPIO40	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
15-14	GPIO39	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
13-12	予約済み	R/W	0h	予約済み
11-10	GPIO37	R/W	3h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
9-8	予約済み	R/W	0h	予約済み
7-6	GPIO35	R/W	3h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	予約済み	R/W	0h	予約済み
3-2	GPIO33	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
1-0	GPIO32	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.23 GPBLOCK レジスタ (オフセット = 7Ch) [リセット = 0h]

GPBLOCK を図 1-26 に示します。説明については、表 1-34 を参照してください。

概略表に戻ります。

GPIO B ロック構成レジスタ (GPIO32 ~ 63)

GPIO 用の GPIO 構成ロック。

0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx レジスタで、同じピンを制御するビットを変更可能

1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる

図 1-26. GPBLOCK レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/W-0h							

表 1-34. GPBLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	予約済み	R/W	0h	予約済み
20	予約済み	R/W	0h	予約済み
19	予約済み	R/W	0h	予約済み
18	予約済み	R/W	0h	予約済み
17	予約済み	R/W	0h	予約済み
16	予約済み	R/W	0h	予約済み
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	予約済み	R/W	0h	予約済み
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み

表 1-34. GPBLOCK レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
10	予約済み	R/W	0h	予約済み
9	GPIO41	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
8	GPIO40	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
7	GPIO39	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
6	予約済み	R/W	0h	予約済み
5	GPIO37	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
4	予約済み	R/W	0h	予約済み
3	GPIO35	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
2	予約済み	R/W	0h	予約済み
1	GPIO33	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn
0	GPIO32	R/W	0h	このピンの構成ロック・ビット リセット・タイプ: SYSRSn

1.10.2.24 GPBCR レジスタ (オフセット = 7Eh) [リセット = 0h]

GPBCR を [図 1-27](#) に示します。説明については、[表 1-35](#) を参照してください。

[概略表](#)に戻ります。

GPIO B ロック・コミット・レジスタ (GPIO32 ~ 63)

GPIO 用の GPIO 構成ロック・コミット :

1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック

0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能

図 1-27. GPBCR レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/WOnce-0h							
23	22	21	20	19	18	17	16
予約済み							
R/WOnce-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/WOnce-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/WOnce-0h							

表 1-35. GPBCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/WOnce	0h	予約済み
30	予約済み	R/WOnce	0h	予約済み
29	予約済み	R/WOnce	0h	予約済み
28	予約済み	R/WOnce	0h	予約済み
27	予約済み	R/WOnce	0h	予約済み
26	予約済み	R/WOnce	0h	予約済み
25	予約済み	R/WOnce	0h	予約済み
24	予約済み	R/WOnce	0h	予約済み
23	予約済み	R/WOnce	0h	予約済み
22	予約済み	R/WOnce	0h	予約済み
21	予約済み	R/WOnce	0h	予約済み
20	予約済み	R/WOnce	0h	予約済み
19	予約済み	R/WOnce	0h	予約済み
18	予約済み	R/WOnce	0h	予約済み
17	予約済み	R/WOnce	0h	予約済み
16	予約済み	R/WOnce	0h	予約済み
15	予約済み	R/WOnce	0h	予約済み
14	予約済み	R/WOnce	0h	予約済み
13	予約済み	R/WOnce	0h	予約済み
12	予約済み	R/WOnce	0h	予約済み
11	予約済み	R/WOnce	0h	予約済み
10	予約済み	R/WOnce	0h	予約済み

表 1-35. GPBCR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO41	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
8	GPIO40	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
7	GPIO39	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
6	予約済み	R/WOnce	0h	予約済み
5	GPIO37	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
4	予約済み	R/WOnce	0h	予約済み
3	GPIO35	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
2	予約済み	R/WOnce	0h	予約済み
1	GPIO33	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn
0	GPIO32	R/WOnce	0h	このピンの構成ロック・コミット・ビット リセット・タイプ: SYSRSn

1.10.2.25 GPHCTRL レジスタ (オフセット = 1C0h) [リセット = 0h]

GPHCTRL を [図 1-28](#) に示します。説明については、[表 1-36](#) を参照してください。

[概略表](#)に戻ります。

GPIO H 認定サンプリング期間制御 (GPIO224 ~ 255)

図 1-28. GPHCTRL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み								QUALPRD2								QUALPRD1								QUALPRD0							
R/W-0h								R/W-0h								R/W-0h								R/W-0h							

表 1-36. GPHCTRL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-24	予約済み	R/W	0h	予約済み
23-16	QUALPRD2	R/W	0h	0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/512 リセット・タイプ：SYSRSn
15-8	QUALPRD1	R/W	0h	0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/511 リセット・タイプ：SYSRSn
7-0	QUALPRD0	R/W	0h	0x00、QUALPRDx = PLLSYSCLK 0x01、QUALPRDx = PLLSYSCLK/2 0x02、QUALPRDx = PLLSYSCLK/4 0xFF、QUALPRDx = PLLSYSCLK/510 リセット・タイプ：SYSRSn

1.10.2.26 GPHQSEL1 レジスタ (オフセット = 1C2h) [リセット = 0h]

GPHQSEL1 を [図 1-29](#) に示します。説明については、[表 1-37](#) を参照してください。

[概略表](#)に戻ります。

GPIO H 修飾子選択 1 レジスタ (GPIO224 ~ 239)

入力認定タイプ :

0、0 同期

0、1 認定 (3 サンプル)

1、0 認定 (6 サンプル)

1、1 非同期 (同期または認定なし)

図 1-29. GPHQSEL1 レジスタ

31	30	29	28	27	26	25	24
GPIO239		GPIO238		GPIO237		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
予約済み		予約済み		GPIO233		GPIO232	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
GPIO231		GPIO230		予約済み		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		GPIO225		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-37. GPHQSEL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	GPIO239	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
29-28	GPIO238	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
27-26	GPIO237	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	GPIO233	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn

表 1-37. GPHQSEL1 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
17-16	GPIO232	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
15-14	GPIO231	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
13-12	GPIO230	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
11-10	予約済み	R/W	0h	予約済み
9-8	GPIO228	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
7-6	GPIO227	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
5-4	GPIO226	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
3-2	GPIO225	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn
1-0	GPIO224	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ: SYSRSn

1.10.2.27 GPHQSEL2 レジスタ (オフセット = 1C4h) [リセット = 0h]

GPHQSEL2 を [図 1-30](#) に示します。説明については、[表 1-38](#) を参照してください。

[概略表](#)に戻ります。

GPIO H 修飾子選択 2 レジスタ (GPIO240 ~ 255)

入力認定タイプ :

0、0 同期

0、1 認定 (3 サンプル)

1、0 認定 (6 サンプル)

1、1 非同期 (同期または認定なし)

図 1-30. GPHQSEL2 レジスタ

31	30	29	28	27	26	25	24
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
予約済み		予約済み		GPIO245		GPIO244	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
予約済み		GPIO242		GPIO241		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-38. GPHQSEL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	予約済み	R/W	0h	予約済み
15-14	予約済み	R/W	0h	予約済み
13-12	予約済み	R/W	0h	予約済み
11-10	GPIO245	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
9-8	GPIO244	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
7-6	予約済み	R/W	0h	予約済み

表 1-38. GPHQSEL2 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
5-4	GPIO242	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
3-2	GPIO241	R/W	0h	0、0、同期 0、1、認定 (3 サンプル) 1、0、認定 (6 サンプル) 1、1、非同期 (同期または認定なし) リセット・タイプ : SYSRSn
1-0	予約済み	R/W	0h	予約済み

1.10.2.28 GPHMUX1 レジスタ (オフセット = 1C6h) [リセット = 0h]

GPHMUX1 を図 1-31 に示します。説明については、表 1-39 を参照してください。

概略表に戻ります。

GPIO H マルチプレクサ 1 レジスタ (GPIO224 ~ 239)

GPIO のピン多重化の選択を定義します。

注：

中間ペリフェラルの選択が GPIO にマップされないよう、このレジスタの前に対応する GPyGMUXn.GPIOz を構成する必要があります。詳細については、GPIO の章を参照してください。

図 1-31. GPHMUX1 レジスタ

31	30	29	28	27	26	25	24
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
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
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
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
予約済み	GPIO230		予約済み	GPIO228		GPIO228	
R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		予約済み		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-39. GPHMUX1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	予約済み	R/W	0h	予約済み
15-14	予約済み	R/W	0h	予約済み
13-12	GPIO230	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
11-10	予約済み	R/W	0h	予約済み
9-8	GPIO228	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
7-6	GPIO227	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	GPIO226	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
3-2	予約済み	R/W	0h	予約済み
1-0	GPIO224	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.29 GPHMUX2 レジスタ (オフセット = 1C8h) [リセット = 0h]

GPHMUX2 を図 1-32 に示します。説明については、表 1-40 を参照してください。

概略表に戻ります。

GPIO H マルチプレクサ 2 レジスタ (GPIO240 ~ 255)

GPIO のピン多重化の選択を定義します。

注：

中間ペリフェラルの選択が GPIO にマップされないよう、このレジスタの前に対応する GPyGMUXn.GPIOz を構成する必要があります。詳細については、GPIO の章を参照してください。

図 1-32. GPHMUX2 レジスタ

31	30	29	28	27	26	25	24
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
予約済み		GPIO242		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-40. GPHMUX2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	予約済み	R/W	0h	予約済み
15-14	予約済み	R/W	0h	予約済み
13-12	予約済み	R/W	0h	予約済み
11-10	予約済み	R/W	0h	予約済み
9-8	予約済み	R/W	0h	予約済み
7-6	予約済み	R/W	0h	予約済み
5-4	GPIO242	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ : SYSRSn
3-2	予約済み	R/W	0h	予約済み
1-0	予約済み	R/W	0h	予約済み

1.10.2.30 GPHDIR レジスタ (オフセット = 1CAh) [リセット = 0h]

GPHDIR を図 1-33 に示します。説明については、表 1-41 を参照してください。

概略表に戻ります。

GPIO H 方向レジスタ (GPIO224 ~ 255)

指定されたピンが GPIO モードで構成されているとき、GPIO ピンの方向を制御します。

0 : ピンを入力として構成します。

1 : ピンを出力として構成します。

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

図 1-33. GPHDIR レジスタ

31	30	29	28	27	26	25	24
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
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
予約済み	予約済み	予約済み	予約済み	予約済み	GPIO242	予約済み	予約済み
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
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
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
予約済み	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	予約済み	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-41. GPHDIR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	予約済み	R/W	0h	予約済み
20	予約済み	R/W	0h	予約済み
19	予約済み	R/W	0h	予約済み
18	GPIO242	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ : SYSRSn
17	予約済み	R/W	0h	予約済み
16	予約済み	R/W	0h	予約済み
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み
13	予約済み	R/W	0h	予約済み
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み

表 1-41. GPHDIR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
10	予約済み	R/W	0h	予約済み
9	予約済み	R/W	0h	予約済み
8	予約済み	R/W	0h	予約済み
7	予約済み	R/W	0h	予約済み
6	GPIO230	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
5	予約済み	R/W	0h	予約済み
4	GPIO228	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
3	GPIO227	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
2	GPIO226	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn
1	予約済み	R/W	0h	予約済み
0	GPIO224	R/W	0h	GPIO モードでのこのピンの方向を定義します リセット・タイプ：SYSRSn

1.10.2.31 GPHPUD レジスタ (オフセット = 1CCh) [リセット = FFFFFFFFh]

GPHPUD を図 1-34 に示します。説明については、表 1-42 を参照してください。

概略表に戻ります。

GPIO H プルアップ・ディセーブル・レジスタ (GPIO224 ~ 255)

GPIO のプルアップをディセーブルします。

0 : プルアップをイネーブルします。

1 : プルアップをディセーブルします。

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。

図 1-34. GPHPUD レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-1h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO245	GPIO244	予約済み	GPIO242	GPIO241	予約済み
R/W-1h							
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	予約済み	予約済み	予約済み	GPIO233	GPIO232
R/W-1h							
7	6	5	4	3	2	1	0
GPIO231	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-1h							

表 1-42. GPHPUD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	1h	予約済み
30	予約済み	R/W	1h	予約済み
29	予約済み	R/W	1h	予約済み
28	予約済み	R/W	1h	予約済み
27	予約済み	R/W	1h	予約済み
26	予約済み	R/W	1h	予約済み
25	予約済み	R/W	1h	予約済み
24	予約済み	R/W	1h	予約済み
23	予約済み	R/W	1h	予約済み
22	予約済み	R/W	1h	予約済み
21	GPIO245	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn

表 1-42. GPHPUD レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
20	GPIO244	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
19	予約済み	R/W	1h	予約済み
18	GPIO242	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
17	GPIO241	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
16	予約済み	R/W	1h	予約済み
15	GPIO239	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
14	GPIO238	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
13	GPIO237	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
12	予約済み	R/W	1h	予約済み
11	予約済み	R/W	1h	予約済み
10	予約済み	R/W	1h	予約済み

表 1-42. GPHPUD レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO233	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
8	GPIO232	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
7	GPIO231	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
6	GPIO230	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
5	予約済み	R/W	1h	予約済み
4	GPIO228	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
3	GPIO227	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn

表 1-42. GPHPUD レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
2	GPIO226	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
1	GPIO225	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn
0	GPIO224	R/W	1h	0 : プルアップをイネーブルします。 1 : プルアップをディセーブルします。 レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] IORSn 信号が Low のとき、GPIO ピンのプルアップは非同期にディセーブルされます。リセットから復帰したとき、ユーザーがソフトウェアでこのレジスタに書き込み、選択的にプルアップをイネーブルにするまで、プルアップはディセーブルのままです。 リセット・タイプ : SYSRSn

1.10.2.32 GPHINV レジスタ (オフセット = 1D0h) [リセット = 0h]

GPHINV を図 1-35 に示します。説明については、表 1-43 を参照してください。

概略表に戻ります。

GPIO H 入力極性反転レジスタ (GPIO224 ~ 255)

デバイスへの非反転または反転のいずれかの GPIO 入力を選択します。

0 : 非反転 GPIO 入力を選択

1 : 反転 GPIO 入力を選択

レジスタを読み取ると、レジスタ設定の現在の値が返されます。

図 1-35. GPHINV レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO245	GPIO244	予約済み	GPIO242	GPIO241	予約済み
R/W-0h							
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	予約済み	予約済み	予約済み	GPIO233	GPIO232
R/W-0h							
7	6	5	4	3	2	1	0
GPIO231	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h							

表 1-43. GPHINV レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	GPIO245	R/W	0h	0 : 非反転 GPIO 入力を選択 1 : 反転 GPIO 入力を選択 注 : [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ : SYSRSn
20	GPIO244	R/W	0h	0 : 非反転 GPIO 入力を選択 1 : 反転 GPIO 入力を選択 注 : [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ : SYSRSn
19	予約済み	R/W	0h	予約済み

表 1-43. GPHINV レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
18	GPIO242	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
17	GPIO241	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
16	予約済み	R/W	0h	予約済み
15	GPIO239	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
14	GPIO238	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
13	GPIO237	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み
10	予約済み	R/W	0h	予約済み
9	GPIO233	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
8	GPIO232	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
7	GPIO231	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
6	GPIO230	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn
5	予約済み	R/W	0h	予約済み
4	GPIO228	R/W	0h	0: 非反転 GPIO 入力を選択 1: 反転 GPIO 入力を選択 注: [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ: SYSRSn

表 1-43. GPHINV レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3	GPIO227	R/W	0h	0 : 非反転 GPIO 入力を選択 1 : 反転 GPIO 入力を選択 注 : [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ : SYSRSn
2	GPIO226	R/W	0h	0 : 非反転 GPIO 入力を選択 1 : 反転 GPIO 入力を選択 注 : [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ : SYSRSn
1	GPIO225	R/W	0h	0 : 非反転 GPIO 入力を選択 1 : 反転 GPIO 入力を選択 注 : [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ : SYSRSn
0	GPIO224	R/W	0h	0 : 非反転 GPIO 入力を選択 1 : 反転 GPIO 入力を選択 注 : [1] レジスタを読み出すと、レジスタ設定の現在の値が返されます。 リセット・タイプ : SYSRSn

1.10.2.33 GPHODR レジスタ (オフセット = 1D2h) [リセット = 0h]

GPHODR を図 1-36 に示します。説明については、表 1-44 を参照してください。

概略表に戻ります。

GPIO H オープン・ドレイン出力レジスタ (GPIO224 ~ GPIO255)

GPIO ピンの通常出力とオープン・ドレイン出力のいずれかを選択します。

0 : 通常出力

1 : オープン・ドレイン出力

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] オープン・ドレイン出力モードで、バッファが出力モードに構成されていると、0 の値は PAD 上の駆動で出力され、1 の値はバッファのトライステートの駆動で出力されます。

図 1-36. GPHODR レジスタ

31	30	29	28	27	26	25	24
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
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
予約済み	予約済み	予約済み	予約済み	予約済み	GPIO242	予約済み	予約済み
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
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
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
予約済み	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	予約済み	GPIO224
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-44. GPHODR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	予約済み	R/W	0h	予約済み
20	予約済み	R/W	0h	予約済み
19	予約済み	R/W	0h	予約済み
18	GPIO242	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ : SYSRSn
17	予約済み	R/W	0h	予約済み
16	予約済み	R/W	0h	予約済み
15	予約済み	R/W	0h	予約済み
14	予約済み	R/W	0h	予約済み

表 1-44. GPHODR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
13	予約済み	R/W	0h	予約済み
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み
10	予約済み	R/W	0h	予約済み
9	予約済み	R/W	0h	予約済み
8	予約済み	R/W	0h	予約済み
7	予約済み	R/W	0h	予約済み
6	GPIO230	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
5	予約済み	R/W	0h	予約済み
4	GPIO228	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
3	GPIO227	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
2	GPIO226	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn
1	予約済み	R/W	0h	予約済み
0	GPIO224	R/W	0h	このピンの出力オープン・ドレインの制御 リセット・タイプ: SYSRSn

1.10.2.34 GPHAMSEL レジスタ (オフセット = 1D4h) [リセット = FFFFFFFFh]

GPHAMSEL を図 1-37 に示します。説明については、表 1-45 を参照してください。

概略表に戻ります。

GPIO H アナログ・モード選択レジスタ (GPIO224 ~ GPIO255)

GPIO ピンのデジタル機能とアナログ機能のいずれかを選択します。

0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できません

1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できます

レジスタを読み出すと、レジスタ設定の現在の値が返されます。

注 :

[1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、t

図 1-37. GPHAMSEL レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-1h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO245	GPIO244	予約済み	GPIO242	GPIO241	予約済み
R/W-1h							
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	予約済み	予約済み	予約済み	GPIO233	GPIO232
R/W-1h							
7	6	5	4	3	2	1	0
GPIO231	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-1h							

表 1-45. GPHAMSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	1h	予約済み
30	予約済み	R/W	1h	予約済み
29	予約済み	R/W	1h	予約済み
28	予約済み	R/W	1h	予約済み
27	予約済み	R/W	1h	予約済み
26	予約済み	R/W	1h	予約済み
25	予約済み	R/W	1h	予約済み
24	予約済み	R/W	1h	予約済み
23	予約済み	R/W	1h	予約済み
22	予約済み	R/W	1h	予約済み
21	GPIO245	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できません 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できます レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn

表 1-45. GPHAMSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
20	GPIO244	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
19	予約済み	R/W	1h	予約済み
18	GPIO242	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
17	GPIO241	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
16	予約済み	R/W	1h	予約済み
15	GPIO239	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
14	GPIO238	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn

表 1-45. GPHAMSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
13	GPIO237	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
12	予約済み	R/W	1h	予約済み
11	予約済み	R/W	1h	予約済み
10	予約済み	R/W	1h	予約済み
9	GPIO233	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
8	GPIO232	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
7	GPIO231	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
6	GPIO230	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注： [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
5	予約済み	R/W	1h	予約済み

表 1-45. GPHAMSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	GPIO228	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
3	GPIO227	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
2	GPIO226	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
1	GPIO225	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn
0	GPIO224	R/W	1h	0 : ピンのアナログ機能がディセーブルされ、他の GPIO 構成レジスタによる指定に従って、デジタル機能を使用できます 1 : ピンのアナログ機能がイネーブルされ、アナログ機能を使用できません レジスタを読み出すと、レジスタ設定の現在の値が返されます。 注 : [1] このレジスタおよびビットは、ユニファイド I/O パッドを介してアナログ機能を共有する GPIO 信号に対してのみ有効です。すべての IO について、これらのレジスタの対応ビットは何の効果もありません。 リセット・タイプ : SYSRSn

1.10.2.35 GPHGMUX1 レジスタ (オフセット = 1E0h) [リセット = 0h]

GPHGMUX1 を図 1-38 に示します。説明については、表 1-46 を参照してください。

概略表に戻ります。

GPIO H ベリフェラル・グループ・マルチプレクサ (GPIO224 ~ 239)

GPIO のピン多重化の選択を定義します。

注：

[1] GPIOx のピン多重化の完全な選択を行うには、GPAMUXy.GPIOx も構成する必要があります。

図 1-38. GPHGMUX1 レジスタ

31	30	29	28	27	26	25	24
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
予約済み		GPIO230		予約済み		GPIO228	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
GPIO227		GPIO226		予約済み		GPIO224	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-46. GPHGMUX1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	予約済み	R/W	0h	予約済み
15-14	予約済み	R/W	0h	予約済み
13-12	GPIO230	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
11-10	予約済み	R/W	0h	予約済み
9-8	GPIO228	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
7-6	GPIO227	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
5-4	GPIO226	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn
3-2	予約済み	R/W	0h	予約済み
1-0	GPIO224	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ：SYSRSn

1.10.2.36 GPHGMUX2 レジスタ (オフセット = 1E2h) [リセット = 0h]

GPHGMUX2 を図 1-39 に示します。説明については、表 1-47 を参照してください。

概略表に戻ります。

GPIO H ベリフェラル・グループ・マルチプレクサ (GPIO240 ~ 255)

GPIO のピン多重化の選択を定義します。

注：

[1] GPIOx のピン多重化の完全な選択を行うには、GPAMUXy.GPIOx も構成する必要があります。

図 1-39. GPHGMUX2 レジスタ

31	30	29	28	27	26	25	24
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
予約済み		予約済み		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
予約済み		GPIO242		予約済み		予約済み	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-47. GPHGMUX2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-30	予約済み	R/W	0h	予約済み
29-28	予約済み	R/W	0h	予約済み
27-26	予約済み	R/W	0h	予約済み
25-24	予約済み	R/W	0h	予約済み
23-22	予約済み	R/W	0h	予約済み
21-20	予約済み	R/W	0h	予約済み
19-18	予約済み	R/W	0h	予約済み
17-16	予約済み	R/W	0h	予約済み
15-14	予約済み	R/W	0h	予約済み
13-12	予約済み	R/W	0h	予約済み
11-10	予約済み	R/W	0h	予約済み
9-8	予約済み	R/W	0h	予約済み
7-6	予約済み	R/W	0h	予約済み
5-4	GPIO242	R/W	0h	GPIO のピン多重化の選択を定義します リセット・タイプ : SYSRSn
3-2	予約済み	R/W	0h	予約済み
1-0	予約済み	R/W	0h	予約済み

1.10.2.37 GPHLOCK レジスタ (オフセット = 1FCh) [リセット = 0h]

GPHLOCK を図 1-40 に示します。説明については、表 1-48 を参照してください。

概略表に戻ります。

GPIO H ロック構成レジスタ (GPIO224 ~ 255)

GPIO 用の GPIO 構成ロック。

0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx レジスタで、同じピンを制御するビットを変更可能

1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる

図 1-40. GPHLOCK レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO245	GPIO244	予約済み	GPIO242	GPIO241	予約済み
R/W-0h							
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	予約済み	予約済み	予約済み	GPIO233	GPIO232
R/W-0h							
7	6	5	4	3	2	1	0
GPIO231	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h							

表 1-48. GPHLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/W	0h	予約済み
30	予約済み	R/W	0h	予約済み
29	予約済み	R/W	0h	予約済み
28	予約済み	R/W	0h	予約済み
27	予約済み	R/W	0h	予約済み
26	予約済み	R/W	0h	予約済み
25	予約済み	R/W	0h	予約済み
24	予約済み	R/W	0h	予約済み
23	予約済み	R/W	0h	予約済み
22	予約済み	R/W	0h	予約済み
21	GPIO245	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
20	GPIO244	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
19	予約済み	R/W	0h	予約済み

表 1-48. GPHLOCK レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
18	GPIO242	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
17	GPIO241	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
16	予約済み	R/W	0h	予約済み
15	GPIO239	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
14	GPIO238	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
13	GPIO237	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
12	予約済み	R/W	0h	予約済み
11	予約済み	R/W	0h	予約済み
10	予約済み	R/W	0h	予約済み
9	GPIO233	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
8	GPIO232	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
7	GPIO231	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn

表 1-48. GPHLOCK レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
6	GPIO230	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
5	予約済み	R/W	0h	予約済み
4	GPIO228	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
3	GPIO227	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
2	GPIO226	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
1	GPIO225	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
0	GPIO224	R/W	0h	1 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットの変更がロックされる 0 : GPyMUX1、GPyMUX2、GPyDIR、GPyINV、GPyODR、GPyAMSEL、GPyGMUX1、GPyGMUX2、GPyCSELx の各レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn

1.10.2.38 GPHCR レジスタ (オフセット = 1FEh) [リセット = 0h]

GPHCR を [図 1-41](#) に示します。説明については、[表 1-49](#) を参照してください。

[概略表](#)に戻ります。

GPIO H ロック・コミット・レジスタ (GPIO224 ~ 255)

GPIO 用の GPIO 構成ロック・コミット :

1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック

0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能

図 1-41. GPHCR レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/WOnce-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO245	GPIO244	予約済み	GPIO242	GPIO241	予約済み
R/WOnce-0h							
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	予約済み	予約済み	予約済み	GPIO233	GPIO232
R/WOnce-0h							
7	6	5	4	3	2	1	0
GPIO231	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/WOnce-0h							

表 1-49. GPHCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	予約済み	R/WOnce	0h	予約済み
30	予約済み	R/WOnce	0h	予約済み
29	予約済み	R/WOnce	0h	予約済み
28	予約済み	R/WOnce	0h	予約済み
27	予約済み	R/WOnce	0h	予約済み
26	予約済み	R/WOnce	0h	予約済み
25	予約済み	R/WOnce	0h	予約済み
24	予約済み	R/WOnce	0h	予約済み
23	予約済み	R/WOnce	0h	予約済み
22	予約済み	R/WOnce	0h	予約済み
21	GPIO245	R/WOnce	0h	1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
20	GPIO244	R/WOnce	0h	1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn
19	予約済み	R/WOnce	0h	予約済み
18	GPIO242	R/WOnce	0h	1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn

表 1-49. GPHCR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
17	GPIO241	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
16	予約済み	R/WOnce	0h	予約済み
15	GPIO239	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
14	GPIO238	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
13	GPIO237	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
12	予約済み	R/WOnce	0h	予約済み
11	予約済み	R/WOnce	0h	予約済み
10	予約済み	R/WOnce	0h	予約済み
9	GPIO233	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
8	GPIO232	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
7	GPIO231	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
6	GPIO230	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
5	予約済み	R/WOnce	0h	予約済み
4	GPIO228	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
3	GPIO227	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
2	GPIO226	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn
1	GPIO225	R/WOnce	0h	1: GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0: GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ: SYSRSn

表 1-49. GPHCR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
0	GPIO224	R/WOnce	0h	1 : GPyLOCK レジスタで、同じピンを制御するビットの変更をロック 0 : GPyLOCK レジスタで、同じピンを制御するビットを変更可能 リセット・タイプ : SYSRSn

1.10.3 GPIO_DATA_REGS レジスタ

表 1-50 に、GPIO_DATA_REGS レジスタのメモリマップされたレジスタを示します。表 1-50 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-50. GPIO_DATA_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	GPADAT	GPIO A データ・レジスタ (GPIO0 ~ 31)		表示
2h	GPASET	GPIO A データ・セット・レジスタ (GPIO0 ~ 31)		表示
4h	GPACLEAR	GPIO A データ・クリア・レジスタ (GPIO0 ~ 31)		表示
6h	GPATOGGLE	GPIO A データ・トグル・レジスタ (GPIO0 ~ 31)		表示
8h	GPBDAT	GPIO B データ・レジスタ (GPIO32 ~ 63)		表示
Ah	GPBSET	GPIO B データ・セット・レジスタ (GPIO32 ~ 63)		表示
Ch	GPBCLEAR	GPIO B データ・クリア・レジスタ (GPIO32 ~ 63)		表示
Eh	GPBTOGGLE	GPIO B データ・トグル・レジスタ (GPIO32 ~ 63)		表示
38h	GPHDAT	GPIO H データ・レジスタ (GPIO224 ~ 255)		表示
3Ah	GPHSET	GPIO H データ・セット・レジスタ (GPIO224 ~ 255)		表示
3Ch	GPHCLEAR	GPIO H データ・クリア・レジスタ (GPIO224 ~ 255)		表示
3Eh	GPHTOGGLE	GPIO H データ・トグル・レジスタ (GPIO224 ~ 255)		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-51 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-51. GPIO_DATA_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み出しで 0 が返される
書き込みタイプ		
W	W	書き込み
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用される場合、レジスタが繰り返し使用されるレジスタのグループの一部である、レジスタ配列の値を参照します。レジスタ・グループは階層構造を形成し、配列は数式で表現されます。

表 1-51. GPIO_DATA_REGS のアクセス・タイプ・コード (続き)

アクセス・タイプ	コード	説明
y		この変数がレジスタ名、オフセット、またはアドレスで使用される場合、レジスタ配列の値を参照します。

1.10.3.1 GPADAT レジスタ (オフセット=0h) [リセット=0h]

図 1-42 に、GPADAT を示し、表 1-52 に、その説明を示します。

概略表に戻ります。

GPIO A データ・レジスタ (GPIO0 ~ 31)

このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。

設計者注：

[1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。

図 1-42. GPADAT レジスタ

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

表 1-52. GPADAT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R/W	0h	予約済み
30	RESERVED	R/W	0h	予約済み
29	GPIO29	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
28	GPIO28	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
27	RESERVED	R/W	0h	予約済み
26	RESERVED	R/W	0h	予約済み
25	RESERVED	R/W	0h	予約済み
24	GPIO24	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
23	GPIO23	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
22	GPIO22	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
21	GPIO21	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
20	GPIO20	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
19	GPIO19	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn

表 1-52. GPADAT レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
18	GPIO18	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
17	GPIO17	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
16	GPIO16	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
15	RESERVED	R/W	0h	予約済み
14	RESERVED	R/W	0h	予約済み
13	GPIO13	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
12	GPIO12	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
11	GPIO11	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
10	GPIO10	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
9	GPIO9	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
8	GPIO8	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
7	GPIO7	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
6	GPIO6	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
5	GPIO5	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
4	GPIO4	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
3	GPIO3	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
2	GPIO2	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
1	GPIO1	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
0	GPIO0	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn

1.10.3.2 GPASET レジスタ (オフセット=2h) [リセット=0h]

図 1-43 に、GPASET を示し、表 1-53 に、その説明を示します。

概略表に戻ります。

GPIO A データ・セット・レジスタ (GPIO0~31)

1 を書き込むと、GPIO 出力データ・ラッチが強制的に 1 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-43. GPASET レジスタ

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

表 1-53. GPASET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	GPIO29	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
28	GPIO28	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	GPIO24	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
23	GPIO23	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
22	GPIO22	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
21	GPIO21	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
20	GPIO20	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
19	GPIO19	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
18	GPIO18	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
17	GPIO17	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn

表 1-53. GPASET レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
16	GPIO16	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	GPIO13	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
12	GPIO12	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
11	GPIO11	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
10	GPIO10	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
9	GPIO9	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
8	GPIO8	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
7	GPIO7	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
6	GPIO6	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
5	GPIO5	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
4	GPIO4	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
3	GPIO3	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
2	GPIO2	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
1	GPIO1	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn
0	GPIO0	R-0/W	0h	このピンの出力設定ビット リセット・タイプ: SYSRSn

1.10.3.3 GPACLEAR レジスタ (オフセット=4h) [リセット=0h]

図 1-44 に、GPACLEAR を示し、表 1-54 に、その説明を示します。

概略表に戻ります。

GPIO A データ・クリア・レジスタ (GPIO0~31)

1 を書き込むと、GPIO0 出力データ・ラッチが強制的に 0 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-44. GPACLEAR レジスタ

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

表 1-54. GPACLEAR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	GPIO29	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
28	GPIO28	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	GPIO24	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
23	GPIO23	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
22	GPIO22	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
21	GPIO21	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
20	GPIO20	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
19	GPIO19	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
18	GPIO18	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
17	GPIO17	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn

表 1-54. GPACLEAR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
16	GPIO16	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	GPIO13	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
12	GPIO12	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
11	GPIO11	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
10	GPIO10	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
9	GPIO9	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
8	GPIO8	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
7	GPIO7	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
6	GPIO6	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
5	GPIO5	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
4	GPIO4	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
3	GPIO3	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
2	GPIO2	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
1	GPIO1	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn
0	GPIO0	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ: SYSRSn

1.10.3.4 GPATOGGLE レジスタ (オフセット=6h) [リセット=0h]

図 1-45 に、GPATOGGLE を示し、表 1-55 に、その説明を示します。

概略表に戻ります。

GPIO A データ・トグル・レジスタ (GPIO0~31)

1 を書き込むと、GPIO0 出力データ・ラッチが 1 から 0 または 0 から 1 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-45. GPATOGGLE レジスタ

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

表 1-55. GPATOGGLE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	GPIO29	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
28	GPIO28	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	GPIO24	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
23	GPIO23	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
22	GPIO22	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
21	GPIO21	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
20	GPIO20	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
19	GPIO19	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
18	GPIO18	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
17	GPIO17	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn

表 1-55. GPATOGGLE レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
16	GPIO16	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	GPIO13	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
12	GPIO12	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
11	GPIO11	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
10	GPIO10	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
9	GPIO9	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
8	GPIO8	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
7	GPIO7	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
6	GPIO6	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
5	GPIO5	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
4	GPIO4	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
3	GPIO3	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
2	GPIO2	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
1	GPIO1	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn
0	GPIO0	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ: SYSRSn

1.10.3.5 GPBDAT レジスタ (オフセット=8h) [リセット=0h]

図 1-46 に、GPBDAT を示し、表 1-56 に、その説明を示します。

概略表に戻ります。

GPIO B データ・レジスタ (GPIO32~63)

このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。

設計者注：

[1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。

図 1-46. GPBDAT レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R/W-0h							

表 1-56. GPBDAT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R/W	0h	予約済み
30	RESERVED	R/W	0h	予約済み
29	RESERVED	R/W	0h	予約済み
28	RESERVED	R/W	0h	予約済み
27	RESERVED	R/W	0h	予約済み
26	RESERVED	R/W	0h	予約済み
25	RESERVED	R/W	0h	予約済み
24	RESERVED	R/W	0h	予約済み
23	RESERVED	R/W	0h	予約済み
22	RESERVED	R/W	0h	予約済み
21	RESERVED	R/W	0h	予約済み
20	RESERVED	R/W	0h	予約済み
19	RESERVED	R/W	0h	予約済み
18	RESERVED	R/W	0h	予約済み
17	RESERVED	R/W	0h	予約済み
16	RESERVED	R/W	0h	予約済み
15	RESERVED	R/W	0h	予約済み
14	RESERVED	R/W	0h	予約済み

表 1-56. GPBDAT レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
13	RESERVED	R/W	0h	予約済み
12	RESERVED	R/W	0h	予約済み
11	RESERVED	R/W	0h	予約済み
10	RESERVED	R/W	0h	予約済み
9	GPIO41	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
8	GPIO40	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
7	GPIO39	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
6	RESERVED	R/W	0h	予約済み
5	GPIO37	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
4	RESERVED	R/W	0h	予約済み
3	GPIO35	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
2	RESERVED	R/W	0h	予約済み
1	GPIO33	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn
0	GPIO32	R/W	0h	このピンのデータ・レジスタ リセット・タイプ: SYSRSn

1.10.3.6 GPBSET レジスタ (オフセット=Ah) [リセット=0h]

図 1-47 に、GPBSET を示し、表 1-57 に、その説明を示します。

概略表に戻ります。

GPIO B データ・セット・レジスタ (GPIO32~63)

1 を書き込むと、GPIO 出力データ・ラッチが強制的に 1 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-47. GPBSET レジスタ

31	30	29	28	27	26	25	24
予約済み							
R-0/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R-0/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R-0/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R-0/W-0h							

表 1-57. GPBSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	RESERVED	R-0/W	0h	予約済み
28	RESERVED	R-0/W	0h	予約済み
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	RESERVED	R-0/W	0h	予約済み
23	RESERVED	R-0/W	0h	予約済み
22	RESERVED	R-0/W	0h	予約済み
21	RESERVED	R-0/W	0h	予約済み
20	RESERVED	R-0/W	0h	予約済み
19	RESERVED	R-0/W	0h	予約済み
18	RESERVED	R-0/W	0h	予約済み
17	RESERVED	R-0/W	0h	予約済み
16	RESERVED	R-0/W	0h	予約済み
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	RESERVED	R-0/W	0h	予約済み
12	RESERVED	R-0/W	0h	予約済み
11	RESERVED	R-0/W	0h	予約済み
10	RESERVED	R-0/W	0h	予約済み

表 1-57. GPBSET レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO41	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
8	GPIO40	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
7	GPIO39	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
6	RESERVED	R-0/W	0h	予約済み
5	GPIO37	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
4	RESERVED	R-0/W	0h	予約済み
3	GPIO35	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
2	RESERVED	R-0/W	0h	予約済み
1	GPIO33	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
0	GPIO32	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn

1.10.3.7 GPBCLEAR レジスタ (オフセット=Ch) [リセット=0h]

図 1-48 に、GPBCLEAR を示し、表 1-58 に、その説明を示します。

概略表に戻ります。

GPIO B データ・クリア・レジスタ (GPIO32 ~ 63)

1 を書き込むと、GPIO0 出力データ・ラッチが強制的に 0 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-48. GPBCLEAR レジスタ

31	30	29	28	27	26	25	24
予約済み							
R-0/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R-0/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R-0/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R-0/W-0h							

表 1-58. GPBCLEAR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	RESERVED	R-0/W	0h	予約済み
28	RESERVED	R-0/W	0h	予約済み
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	RESERVED	R-0/W	0h	予約済み
23	RESERVED	R-0/W	0h	予約済み
22	RESERVED	R-0/W	0h	予約済み
21	RESERVED	R-0/W	0h	予約済み
20	RESERVED	R-0/W	0h	予約済み
19	RESERVED	R-0/W	0h	予約済み
18	RESERVED	R-0/W	0h	予約済み
17	RESERVED	R-0/W	0h	予約済み
16	RESERVED	R-0/W	0h	予約済み
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	RESERVED	R-0/W	0h	予約済み
12	RESERVED	R-0/W	0h	予約済み
11	RESERVED	R-0/W	0h	予約済み
10	RESERVED	R-0/W	0h	予約済み

表 1-58. GPBCLEAR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO41	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn
8	GPIO40	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn
7	GPIO39	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn
6	RESERVED	R-0/W	0h	予約済み
5	GPIO37	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn
4	RESERVED	R-0/W	0h	予約済み
3	GPIO35	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn
2	RESERVED	R-0/W	0h	予約済み
1	GPIO33	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn
0	GPIO32	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ：SYSRSn

1.10.3.8 GPBTOGGLE レジスタ (オフセット=Eh) [リセット=0h]

図 1-49 に、GPBTOGGLE を示し、表 1-59 に、その説明を示します。

概略表に戻ります。

GPIO B データ・トグル・レジスタ (GPIO32 ~ 63)

1 を書き込むと、GPIO0 出力データ・ラッチが 1 から 0 または 0 から 1 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-49. GPBTOGGLE レジスタ

31	30	29	28	27	26	25	24
予約済み							
R-0/W-0h							
23	22	21	20	19	18	17	16
予約済み							
R-0/W-0h							
15	14	13	12	11	10	9	8
予約済み	予約済み	予約済み	予約済み	予約済み	予約済み	GPIO41	GPIO40
R-0/W-0h							
7	6	5	4	3	2	1	0
GPIO39	予約済み	GPIO37	予約済み	GPIO35	予約済み	GPIO33	GPIO32
R-0/W-0h							

表 1-59. GPBTOGGLE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	RESERVED	R-0/W	0h	予約済み
28	RESERVED	R-0/W	0h	予約済み
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	RESERVED	R-0/W	0h	予約済み
23	RESERVED	R-0/W	0h	予約済み
22	RESERVED	R-0/W	0h	予約済み
21	RESERVED	R-0/W	0h	予約済み
20	RESERVED	R-0/W	0h	予約済み
19	RESERVED	R-0/W	0h	予約済み
18	RESERVED	R-0/W	0h	予約済み
17	RESERVED	R-0/W	0h	予約済み
16	RESERVED	R-0/W	0h	予約済み
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	RESERVED	R-0/W	0h	予約済み
12	RESERVED	R-0/W	0h	予約済み
11	RESERVED	R-0/W	0h	予約済み
10	RESERVED	R-0/W	0h	予約済み

表 1-59. GPBTOGGLE レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	GPIO41	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
8	GPIO40	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
7	GPIO39	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
6	RESERVED	R-0/W	0h	予約済み
5	GPIO37	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
4	RESERVED	R-0/W	0h	予約済み
3	GPIO35	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
2	RESERVED	R-0/W	0h	予約済み
1	GPIO33	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
0	GPIO32	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn

1.10.3.9 GPHDAT レジスタ (オフセット=38h) [リセット=0h]

図 1-50 に、GPHDAT を示し、表 1-60 に、その説明を示します。

概略表に戻ります。

GPIO H データ・レジスタ (GPIO224 ~ 255)

このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。

設計者注：

[1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。

図 1-50. GPHDAT レジスタ

31	30	29	28	27	26	25	24
予約済み							
R/W-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	GPIO245	GPIO244	予約済み	GPIO242	GPIO241	予約済み
R/W-0h							
15	14	13	12	11	10	9	8
GPIO239	GPIO238	GPIO237	予約済み	予約済み	予約済み	GPIO233	GPIO232
R/W-0h							
7	6	5	4	3	2	1	0
GPIO231	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	GPIO225	GPIO224
R/W-0h							

表 1-60. GPHDAT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R/W	0h	予約済み
30	RESERVED	R/W	0h	予約済み
29	RESERVED	R/W	0h	予約済み
28	RESERVED	R/W	0h	予約済み
27	RESERVED	R/W	0h	予約済み
26	RESERVED	R/W	0h	予約済み
25	RESERVED	R/W	0h	予約済み
24	RESERVED	R/W	0h	予約済み
23	RESERVED	R/W	0h	予約済み
22	RESERVED	R/W	0h	予約済み

表 1-60. GPHDAT レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
21	GPIO245	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
20	GPIO244	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
19	RESERVED	R/W	0h	予約済み
18	GPIO242	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
17	GPIO241	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
16	RESERVED	R/W	0h	予約済み

表 1-60. GPHDAT レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
15	GPIO239	R/W	0h	このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。 設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn
14	GPIO238	R/W	0h	このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。 設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn
13	GPIO237	R/W	0h	このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。 設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn
12	RESERVED	R/W	0h	予約済み
11	RESERVED	R/W	0h	予約済み
10	RESERVED	R/W	0h	予約済み
9	GPIO233	R/W	0h	このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。 設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn

表 1-60. GPHDAT レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
8	GPIO232	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
7	GPIO231	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
6	GPIO230	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
5	RESERVED	R/W	0h	予約済み
4	GPIO228	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>

表 1-60. GPHDAT レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3	GPIO227	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
2	GPIO226	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
1	GPIO225	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>
0	GPIO224	R/W	0h	<p>このレジスタを読み取ると、ピンがどのモードにあるかに関係なく、GPIO ピンの現在のステータスが表示されます。このレジスタに書き込むと、ピンが GPIO 出力モードでイネーブルになっている場合は、GPIO ピンが High または Low に設定され、それ以外の場合は、書き込まれた値はラッチされますが、無視されます。出力レジスタのラッチの状態は、次の書き込み操作まで現在の状態のままになります。システムをリセットすると、すべてのビットとラッチされた値がゼロにクリアされます。</p> <p>設計者注： [1] GPIODAT レジスタの読み取りは、GPIODAT レジスタの出力ラッチの状態ではなく、PIN の状態 (認定後) を反映する必要があります。 リセット・タイプ：SYSRSn</p>

1.10.3.10 GPHSET レジスタ (オフセット=3Ah) [リセット=0h]

図 1-51 に、GPHSET を示し、表 1-61 に、その説明を示します。

概略表に戻ります。

GPIO H データ・セット・レジスタ (GPIO224 ~ 255)

1 を書き込むと、GPIO 出力データ・ラッチが強制的に 1 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-51. GPHSET レジスタ

31	30	29	28	27	26	25	24
予約済み							
R-0/W-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	予約済み	予約済み	予約済み	GPIO242	予約済み	予約済み
R-0/W-0h							
15	14	13	12	11	10	9	8
予約済み							
R-0/W-0h							
7	6	5	4	3	2	1	0
予約済み	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	予約済み	GPIO224
R-0/W-0h							

表 1-61. GPHSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	RESERVED	R-0/W	0h	予約済み
28	RESERVED	R-0/W	0h	予約済み
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	RESERVED	R-0/W	0h	予約済み
23	RESERVED	R-0/W	0h	予約済み
22	RESERVED	R-0/W	0h	予約済み
21	RESERVED	R-0/W	0h	予約済み
20	RESERVED	R-0/W	0h	予約済み
19	RESERVED	R-0/W	0h	予約済み
18	GPIO242	R-0/W	0h	このピンの出力設定ビット リセット・タイプ : SYSRSn
17	RESERVED	R-0/W	0h	予約済み
16	RESERVED	R-0/W	0h	予約済み
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	RESERVED	R-0/W	0h	予約済み
12	RESERVED	R-0/W	0h	予約済み
11	RESERVED	R-0/W	0h	予約済み
10	RESERVED	R-0/W	0h	予約済み

表 1-61. GPHSET レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	RESERVED	R-0/W	0h	予約済み
8	RESERVED	R-0/W	0h	予約済み
7	RESERVED	R-0/W	0h	予約済み
6	GPIO230	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
5	RESERVED	R-0/W	0h	予約済み
4	GPIO228	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
3	GPIO227	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
2	GPIO226	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn
1	RESERVED	R-0/W	0h	予約済み
0	GPIO224	R-0/W	0h	このピンの出力設定ビット リセット・タイプ：SYSRSn

1.10.3.11 GPHCLEAR レジスタ (オフセット=3Ch) [リセット=0h]

図 1-52 に、GPHCLEAR を示し、表 1-62 に、その説明を示します。

概略表に戻ります。

GPIO H データ・クリア・レジスタ (GPIO224 ~ 255)

1 を書き込むと、GPIO0 出力データ・ラッチが強制的に 0 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-52. GPHCLEAR レジスタ

31	30	29	28	27	26	25	24
予約済み							
R-0/W-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	予約済み	予約済み	予約済み	GPIO242	予約済み	予約済み
R-0/W-0h							
15	14	13	12	11	10	9	8
予約済み							
R-0/W-0h							
7	6	5	4	3	2	1	0
予約済み	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	予約済み	GPIO224
R-0/W-0h							

表 1-62. GPHCLEAR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	RESERVED	R-0/W	0h	予約済み
28	RESERVED	R-0/W	0h	予約済み
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	RESERVED	R-0/W	0h	予約済み
23	RESERVED	R-0/W	0h	予約済み
22	RESERVED	R-0/W	0h	予約済み
21	RESERVED	R-0/W	0h	予約済み
20	RESERVED	R-0/W	0h	予約済み
19	RESERVED	R-0/W	0h	予約済み
18	GPIO242	R-0/W	0h	このピンの出力クリア・ビット リセット・タイプ : SYSRSn
17	RESERVED	R-0/W	0h	予約済み
16	RESERVED	R-0/W	0h	予約済み
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	RESERVED	R-0/W	0h	予約済み
12	RESERVED	R-0/W	0h	予約済み
11	RESERVED	R-0/W	0h	予約済み
10	RESERVED	R-0/W	0h	予約済み

表 1-62. GPHCLEAR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	RESERVED	R-0/W	0h	予約済み
8	RESERVED	R-0/W	0h	予約済み
7	RESERVED	R-0/W	0h	予約済み
6	GPIO230	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ : SYSRSn
5	RESERVED	R-0/W	0h	予約済み
4	GPIO228	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ : SYSRSn
3	GPIO227	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ : SYSRSn
2	GPIO226	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ : SYSRSn
1	RESERVED	R-0/W	0h	予約済み
0	GPIO224	R-0/W	0h	このピンの出カクリア・ビット リセット・タイプ : SYSRSn

1.10.3.12 GPHTOGGLE レジスタ (オフセット=3Eh) [リセット=0h]

図 1-53 に、GPHTOGGLE を示し、表 1-63 に、その説明を示します。

概略表に戻ります。

GPIO H データ・ トグル・ レジスタ (GPIO224 ~ 255)

1 を書き込むと、GPIO0 出力データ・ ラッチが 1 から 0 または 0 から 1 になります。

0 の書き込みは無視されます。

常に 0 を読み戻します。

図 1-53. GPHTOGGLE レジスタ

31	30	29	28	27	26	25	24
予約済み							
R-0/W-0h							
23	22	21	20	19	18	17	16
予約済み	予約済み	予約済み	予約済み	予約済み	GPIO242	予約済み	予約済み
R-0/W-0h							
15	14	13	12	11	10	9	8
予約済み							
R-0/W-0h							
7	6	5	4	3	2	1	0
予約済み	GPIO230	予約済み	GPIO228	GPIO227	GPIO226	予約済み	GPIO224
R-0/W-0h							

表 1-63. GPHTOGGLE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31	RESERVED	R-0/W	0h	予約済み
30	RESERVED	R-0/W	0h	予約済み
29	RESERVED	R-0/W	0h	予約済み
28	RESERVED	R-0/W	0h	予約済み
27	RESERVED	R-0/W	0h	予約済み
26	RESERVED	R-0/W	0h	予約済み
25	RESERVED	R-0/W	0h	予約済み
24	RESERVED	R-0/W	0h	予約済み
23	RESERVED	R-0/W	0h	予約済み
22	RESERVED	R-0/W	0h	予約済み
21	RESERVED	R-0/W	0h	予約済み
20	RESERVED	R-0/W	0h	予約済み
19	RESERVED	R-0/W	0h	予約済み
18	GPIO242	R-0/W	0h	出カトグル・ レジスタ GPIO ピン リセット・ タイプ : SYSRSn
17	RESERVED	R-0/W	0h	予約済み
16	RESERVED	R-0/W	0h	予約済み
15	RESERVED	R-0/W	0h	予約済み
14	RESERVED	R-0/W	0h	予約済み
13	RESERVED	R-0/W	0h	予約済み
12	RESERVED	R-0/W	0h	予約済み
11	RESERVED	R-0/W	0h	予約済み
10	RESERVED	R-0/W	0h	予約済み

表 1-63. GPHTOGGLE レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9	RESERVED	R-0/W	0h	予約済み
8	RESERVED	R-0/W	0h	予約済み
7	RESERVED	R-0/W	0h	予約済み
6	GPIO230	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
5	RESERVED	R-0/W	0h	予約済み
4	GPIO228	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
3	GPIO227	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
2	GPIO226	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn
1	RESERVED	R-0/W	0h	予約済み
0	GPIO224	R-0/W	0h	出カトグル・レジスタ GPIO ピン リセット・タイプ : SYSRSn

1.10.4 GPIO_DATA_READ_REGS レジスタ

GPIO_DATA_READ_REGS レジスタのメモリマップされたレジスタを、表 1-64 に示します。表 1-64 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-64. GPIO_DATA_READ_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	GPADAT_R	GPIO A データ読み取りレジスタ		表示
2h	GPBDAT_R	GPIO B データ読み取りレジスタ		表示
Eh	GPHDAT_R	GPIO H データ読み取りレジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。このセクションでアクセス・タイプに使用しているコードを、表 1-65 に示します。

表 1-65. GPIO_DATA_READ_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列の変数		
i, j, k, l, m, n		レジスタ名、オフセット、またはアドレスでこれらの変数を使用すると、レジスタ配列の値を参照することになり、ここではレジスタが反復レジスタ・グループの一部になります。レジスタ・グループは階層構造を形成し、配列は式で表されます。
y		レジスタ名、オフセット、またはアドレスでこの変数を使用すると、レジスタ配列の値を参照することになります。

1.10.4.1 GPADAT_R レジスタ (オフセット = 0h) [リセット = 0h]

GPADAT_R を [図 1-54](#) に示します。説明については、[表 1-66](#) を参照してください。

[概略表](#)に戻ります。

GPIO A データ読み取りレジスタ。

読み取り時の GPADAT レジスタの内容を返します。このレジスタへの書き込みは無効です

図 1-54. GPADAT_R レジスタ

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

表 1-66. GPADAT_R レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	DATA	R	0h	このレジスタを読み取ると、GPADAT レジスタの内容が返されます。 書き込みは無効です リセット・タイプ：CPU1.SYSRSn

1.10.4.2 GPBDAT_R (オフセット = 2h) [リセット = 0h]

GPBDAT_R を [図 1-55](#) に示します。説明については、[表 1-67](#) を参照してください。

[概略表](#)に戻ります。

GPIO B データ読み取りレジスタ。

読み取り時の GPBDAT レジスタの内容を返します。このレジスタへの書き込みは無効です

図 1-55. GPBDAT_R レジスタ

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

表 1-67. GPBDAT_R レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	DATA	R	0h	このレジスタを読み取ると、GPBDAT レジスタの内容が返されます。 書き込みは無効です リセット・タイプ：CPU1.SYSRSn

1.10.4.3 GPHDAT_R (オフセット = Eh) [リセット = 0h]

GPHDAT_R を図 1-56 に示します。説明については、表 1-68 を参照してください。

概略表に戻ります。

GPIO H データ読み取りレジスタ。

読み取り時の GPHDAT レジスタの内容を返します。このレジスタへの書き込みは無効です

図 1-56. GPHDAT_R レジスタ

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

表 1-68. GPHDAT_R レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	DATA	R	0h	このレジスタを読み取ると、GPHDAT レジスタの内容が返されます。 書き込みは無効です リセット・タイプ：CPU1.SYSRSn

1.10.5 GPIO レジスタを Driverlib 関数に

表 1-69. GPIO レジスタを Driverlib 関数に

ファイル	Driverlib 関数
GPACTRL	
gpio.c	GPIO_setQualificationPeriod
GPAQSEL1	
gpio.c	GPIO_setQualificationMode
gpio.c	GPIO_getQualificationMode
GPAQSEL2	
-	GPAQSEL1 を参照
GPAMUX1	
gpio.c	GPIO_setPinConfig
GPAMUX2	
-	GPAMUX1 を参照
GPADIR	
gpio.c	GPIO_setDirectionMode
gpio.c	GPIO_getDirectionMode
GPAPUD	
gpio.c	GPIO_setPadConfig
gpio.c	GPIO_getPadConfig
GPAINV	
gpio.c	GPIO_setPadConfig
gpio.c	GPIO_getPadConfig
GPAODR	
gpio.c	GPIO_setPadConfig
gpio.c	GPIO_getPadConfig
GPAAMSEL	
-	
GPAGMUX1	
gpio.c	GPIO_setPinConfig

表 1-69. GPIO レジスタを Driverlib 関数に (続き)

ファイル	Driverlib 関数
GPAGMUX2	
-	GPAGMUX1 を参照
GPALOCK	
gpio.h	GPIO_lockPortConfig
gpio.h	GPIO_unlockPortConfig
GPACR	
gpio.h	GPIO_commitPortConfig
GPBCTRL	
-	GPACTRL を参照
GPBQSEL1	
-	GPAQSEL1 を参照
GPBMUX1	
-	GPAMUX1 を参照
GPBDIR	
-	GPADIR を参照
GPBPUD	
-	GPAPUD を参照
GPBINV	
-	GPAINV を参照
GPBODR	
-	GPAODR を参照
GPBGMUX1	
-	GPAGMUX1 を参照
GPBLOCK	
-	GPALOCK を参照
GPBCR	
-	GPACR を参照
GPHCTRL	
-	GPACTRL を参照
GPHQSEL1	
-	GPAQSEL1 を参照
GPHQSEL2	
-	GPAQSEL1 を参照
GPHMUX1	
-	GPAMUX1 を参照
GPHMUX2	
-	GPAMUX1 を参照
GPHDIR	
-	GPADIR を参照
GPHPUD	
-	GPAPUD を参照
GPHINV	
-	GPAINV を参照
GPHODR	
-	GPAODR を参照

表 1-69. GPIO レジスタを Driverlib 関数に (続き)

ファイル	Driverlib 関数
GPHAMSEL	
-	
GPHGMUX1	
-	GPAGMUX1 を参照
GPHGMUX2	
-	GPAGMUX1 を参照
GPHLOCK	
-	GPALOCK を参照
GPHCR	
-	GPACR を参照
GPADAT	
gpio.h	GPIO_readPin
gpio.h	GPIO_readPortData
gpio.h	GPIO_writePortData
GPASET	
gpio.h	GPIO_writePin
gpio.h	GPIO_setPortPins
GPACLEAR	
gpio.h	GPIO_writePin
gpio.h	GPIO_clearPortPins
GPATOGGLE	
gpio.h	GPIO_togglePin
gpio.h	GPIO_togglePortPins
GPBDAT	
-	GPADAT を参照
GPBSET	
-	GPASET を参照
GPBCLEAR	
-	GPACLEAR を参照
GPBTOGGLE	
-	GPATOGGLE を参照
GPHDAT	
-	GPADAT を参照
GPHSET	
-	GPASET を参照
GPHCLEAR	
-	GPACLEAR を参照
GPHTOGGLE	
-	GPATOGGLE を参照
GPADAT_R	
-	
GPBDAT_R	
-	
GPHDAT_R	

表 1-69. GPIO レジスタを Driverlib 関数に (続き)

ファイル	Driverlib 関数
-	

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020 – REVISED MARCH 2023



この章で説明する A/D コンバータ (ADC) モジュールは、タイプ 5 ADC です。同じタイプのモジュールを含むデバイスのリスト、タイプ間の違い、同じタイプのデバイス間の違いについては、『[C2000 リアルタイム制御 MUC ペリフェラル・リファレンス・ガイド](#)』を参照してください。

1.1 概要.....	4
1.2 ADC の構成可能性.....	7
1.3 SOC の動作原理.....	10
1.4 SOC の構成例.....	13
1.5 ADC 変換の優先度.....	15
1.6 バースト・モード.....	18
1.7 EOC と割り込み動作.....	20
1.8 後処理ブロック.....	23
1.9 開放 / 短絡検出回路 (OSDETECT).....	27
1.10 パワーアップ・シーケンス.....	29
1.11 ADC 較正.....	29
1.12 ADC タイミング.....	31
1.13 追加情報.....	34
1.14 ソフトウェア.....	40
1.15 ADC レジスタ.....	45

1.1 概要

ADC モジュールは、12 ビットの逐次比較 (SAR) 型の ADC です。ADC は、コアとラッパーで構成されています。コアは、チャンネル選択マルチプレクサ、サンプル / ホールド (S/H) 回路、逐次比較回路、基準電圧回路、その他のアナログ・ サポート回路を含むアナログ回路で構成されています。ラッパーは、ADC を構成および制御するデジタル回路で構成されています。これらの回路には、プログラマブル変換用のロジック、結果レジスタ、アナログ回路へのインターフェイス、ペリフェラル・ バスへのインターフェイス、後処理回路、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。

各 ADC モジュールは、単一のサンプル / ホールド (S/H) 回路で構成されています。ADC モジュールは、同じチップ上で複数回複製された設計になっており、複数の ADC を同時にサンプリングすることも、独立して動作させることもできます。ADC ラッパーは、変換開始 (SOC) に基づいています ([セクション 1.3](#) を参照)。

1.1.1 ADC の関連資料

基礎資料

- 『C2000 MCU 用 ADC 入力回路の評価 (TINA-TI シミュレーション・ ツールを使用)』アプリケーション・ ノート
- C2000 Academy - ADC
- TI Design / シミュレーション・ ツール向け PSpice
- 『リアルタイム制御リファレンス・ ガイド』
 - 「ADC」セクションを参照
- TI プレジジョン・ ラボ - ADC
- TI プレジジョン・ ラボ : SAR ADC のリファレンス入力の駆動 (ビデオ)
- TI プレジジョン・ ラボ : A/D コンバータ (ADC) の概要 (ビデオ)
- TI プレジジョン・ ラボ : SAR ADC 入力ドライバの設計 (ビデオ)
- TI e2e : VDDA を VREFHI に接続
- TI e2e : ADC 入力保護のトポロジ
- TI e2e : サンプリングを行った場合に ADC の入力電圧が低下する理由
 - ADC を使用した高インピーダンス分圧器のサンプリング
- 『データ・ コンバータの理解』アプリケーション・ レポート

入門資料

- ADC 割り込みを使用した ADC-PWM の同期
 - 注 : これはテキサス・ インスツルメンツ以外 (サード・ パーティ) サイトです。
- C2000 MCU 用 A/D コンバータ (ADC) トレーニング (ビデオ)
- 『F2800x C2000 リアルタイム MCU シリーズ用ハードウェア設計ガイド』

エキスパート資料

- アナログ技術者向けカリキュレータ
- 『アナログ回路設計式一覧ポケット・ ガイド』
- 『C2000 ADC 用の電荷共有駆動回路 (PSPICE-FOR-TI シミュレーション・ ツールを使用)』アプリケーション・ ノート
- 『C2000 ADC 用の電荷共有駆動回路 (TINA-TI シミュレーション・ ツールを使用)』アプリケーション・ ノート
- オシロスコープを使用したマイクロコントローラの内蔵 ADC のデバッグ
- 『ADC メモリのクロストークを低減する方法』アプリケーション・ ノート
- TI プレジジョン・ ラボ : ADC AC 仕様 (ビデオ)
- TI プレジジョン・ ラボ : ADC 誤差の発生源 (ビデオ)
- TI プレジジョン・ ラボ : ADC ノイズ (ビデオ)
- TI プレジジョン・ ラボ : A/D コンバータ (ADC) ドライブ・ トポロジ (ビデオ)
- TI プレジジョン・ ラボ : データ・ コンバータの電氣的オーバーストレス (ビデオ)
- TI プレジジョン・ ラボ : 高速 ADC の基礎 (ビデオ)
- TI プレジジョン・ ラボ : 逐次比較型 / デルタ・ シグマ : 違いの理解 (ビデオ)

- TI e2e : ADC の帯域幅の明確化
- TI e2e : ADC の較正と未調整誤差の合計
- TI e2e : ADC リファレンス・ドライバ・オプション
- TI e2e : オーバーサンプリングによる ADC の分解能
- TI e2e : インターリーブ・モード用の ADC 構成
- TI e2e : 単一 ADC による同時サンプリング

1.1.2 特長

各 ADC には次のような特長があります。

- VREFHI および VREFLO ピンによって設定されるレシオメトリック外部リファレンス
- 2.5V または 3.3V の内部リファレンスを選択可能
- シングルエンド信号変換
- 最大 21 チャンネルの入カマルチプレクサ
- 16 本の構成可能な SOC
- 16 個の個別にアドレス指定可能な結果レジスタ
- 複数のトリガ・ソース
 - S/W - ソフトウェアを即時に起動
 - すべての ePWM - ADCSOC A または B
 - GPIO XINT2
 - CPU タイマ 0/1/2
 - ADCINT1/2
- 4 つのフレキシブルな PIE 割り込み
- バースト・モード
- 4 つの後処理ブロック、それぞれに次の機能を搭載：
 - 飽和オフセット較正
 - 設定点からの誤差の計算
 - 高、低、ゼロクロス比較、割り込みおよび ePWM トリップ機能付き
 - トリガからサンプルまでの遅延キャプチャ

注

すべてのチャンネルがすべての ADC からピンへ接続できるわけではありません。使用可能なチャンネルを確認するには、デバイスのデータ・マニュアルを参照してください。

1.2 ADC の構成可能性

一部の ADC 構成は、SOC によって個別に管理されますが、他の構成は、ADC モジュールごとにまとめて管理されます。表 1-1 に、基本的な ADC オプションとその構成レベルを示します。以降のセクションでは、これらの構成について説明します。

表 1-1. ADC オプションおよび構成レベル

オプション	構成機能
クロック	モジュールごと ⁽¹⁾
分解能	構成不可 (12 ビットのみ)
信号モード	構成不可 (シングルエンドのみ)
リファレンス電圧源	すべてのモジュールを外部または内部に構成
トリガ・ソース	SOC ごと ⁽¹⁾
変換対象チャンネル	SOC ごと
アキュイジション・ウィンドウ幅	SOC ごと ⁽¹⁾
EOC の位置	モジュールごと
バースト・モード	モジュールごと ⁽¹⁾

(1) これらの設定について、異なる ADC モジュールに異なる値を書き込むと、ADC が非同期に動作する可能性があります。ADC が同期動作または非同期動作している場合のガイダンスについては、セクション 1.13.1 を参照してください。

1.2.1 クロック構成

ベース ADC クロックは、システム・クロック (SYSCLK) から直接供給されます。ADC アキュイジション・ウィンドウの生成には SYSCLK が使用されます。レジスタ ADCCTL2 には、ADCCLK を指定する PRESCALE フィールドがあります。ADCCLK は、コンバータのクロック供給に使用されます。

コアが電圧を変換結果に処理するには、約 10.5 ADCCLK サイクル必要です。必要なアキュイジション・ウィンドウの幅は、ユーザーが決定する必要があります。セクション 1.13.2 を参照してください。

注

ADCCTL2.PRESCALE に適切な値を決定するには、デバイスのデータ・マニュアルを参照して、最大 SYSCLK および ADCCLK 周波数を決定します。

1.2.2 分解能

ADC の分解能によって、アナログ範囲をデジタル値に量子化する方法が決まります。この ADC は、12 ビットの分解能をサポートします。

1.2.3 基準電圧

1.2.3.1 外部リファレンス・モード

ADC モジュールは、VREFHI 入力と VREFLO 入力を共有します。外部リファレンス・モードでは、これらのピンをレシオメトリック・リファレンスとして使用して、ADC 変換の入力範囲を判定します。

リファレンス電圧の供給方法については、[セクション 1.13.6](#) を参照してください。

注

- 外部 VREFHI および VREFLO ピンのないデバイスでは、VREFHI は内部的に VDDA に接続され、VREFLO は内部的にデバイスのアナログ・グランド VSSA に接続されます。VREFHI ピンおよび VREFLO ピンのないパッケージでは、内部リファレンス・モードは使用できません。VREFHI ピンと VREFLO ピンのあるパッケージについては、デバイスのデータシートを参照してください。
 - VREFHI および VREFLO の許容電圧範囲については、デバイスのデータ・マニュアルを参照してください。
 - 外部リファレンス・モードでは、VREFHI ピンに外付けコンデンサが必要です。必要な値については、デバイスのデータ・マニュアルを参照してください。
-

1.2.3.2 内部リファレンス・モード

内部リファレンス・モードでは、デバイスは電圧出力を VREFHI ピンに駆動します。VREFHI ピンと VREFLO ピンは、ADC の変換範囲を設定します。内部リファレンス・モードは、VREFHI ピンと VREFLO ピンのないパッケージでは使用できません。VREFHI ピンと VREFLO ピンのあるパッケージについては、デバイスのデータシートを参照してください。

内部リファレンス電圧は、2.5V または 1.65V のいずれかに構成できます。1.65V の内部リファレンス電圧を選択すると、ADC 入力信号は変換前に内部的に 2 で除算されるので、ADC 変換範囲は実質的に VREFLO ~ 3.3V になります。

注

内部リファレンス・モードでは、VREFHI ピンに外付けコンデンサも必要です。必要な値については、デバイスのデータシートを参照してください。

1.2.3.3 リファレンス・モードの選択

電圧リファレンスを構成するには、使用するヘッダー・ファイルに応じて、ADC_setVREF() または SetVREF() のいずれかの関数を使用します (C2000Ware で提供)。これらの関数を使用することにより、適切なトリムが ADC トリム・レジスタに確実にロードされます。この関数は、デバイスのリセット後に少なくとも 1 回呼び出す必要があります。ANAREFCTL レジスタに直接書き込んで電圧リファレンスを設定しないでください。

1.2.4 信号モード

ADC はシングルエンド信号処理をサポートしています。

シングルエンド・モードでは、コンバータへの入力電圧は VREFLO を基準として、1 本のピン (ADCINx) でサンプリングされます。

1.2.5 予測される変換結果

表 1-2 に、アナログ入力電圧に基づく理想的なデジタル変換結果を示します。小数値は切り捨てられます。

表 1-2. アナログから 12 ビット・ デジタルへの変換式

アナログ入力	デジタル結果
$ADCINy \leq VREFLO$	$ADCRESULTx = 0$
$VREFLO < ADCINy < VREFHI$	$ADCRESULTx = 4096 \left(\frac{ADCINy - VREFLO}{VREFHI - VREFLO} \right)$
$ADCINy \geq VREFHI$	$ADCRESULTx = 4095$

1.2.6 変換結果の解釈

得られた ADC 変換結果に基づき、対応する理想的なアナログ入力を表 1-3 に示します。これは、この変換結果を生成することが可能なアナログ電圧範囲の中央に対応します。

表 1-3. 12 ビット・ デジタルのアナログへの変換式

デジタル値	等価のアナログ値
$ADCRESULTy = 0$	$ADCINx \leq VREFLO$
$0 < ADCRESULTy < 4095$	$ADCINx = (VREFHI - VREFLO) \left(\frac{ADCRESULTy}{4096} \right) + VREFLO$
$ADCRESULTy = 4095$	$ADCINx \geq VREFHI$

1.3 SOC の動作原理

ADC のトリガと変換のシーケンスは、構成可能な変換開始 (SOC) によって実現されます。各 SOC は、1 つのチャンネルの 1 つの変換を定義する構成セットです。このセットには、変換を開始するトリガ・ソース、変換するチャンネル、アキュイジション (サンプル) ウィンドウ幅の 3 つの構成があります。SOC 用に構成されたトリガが受信されると、ラッパーは指定されたアキュイジション・ウィンドウ幅を使用して、指定されたチャンネルがキャプチャされることを確認します。

必要に応じて、複数の SOC に同じトリガ、チャンネル、アキュイジション・ウィンドウを構成できます。複数の SOC を同じトリガを使用するように構成すると、そのトリガで変換シーケンスを生成できます。複数の SOC を同じトリガとチャンネルを使用するように構成すると、オーバーサンプリングが可能になります。

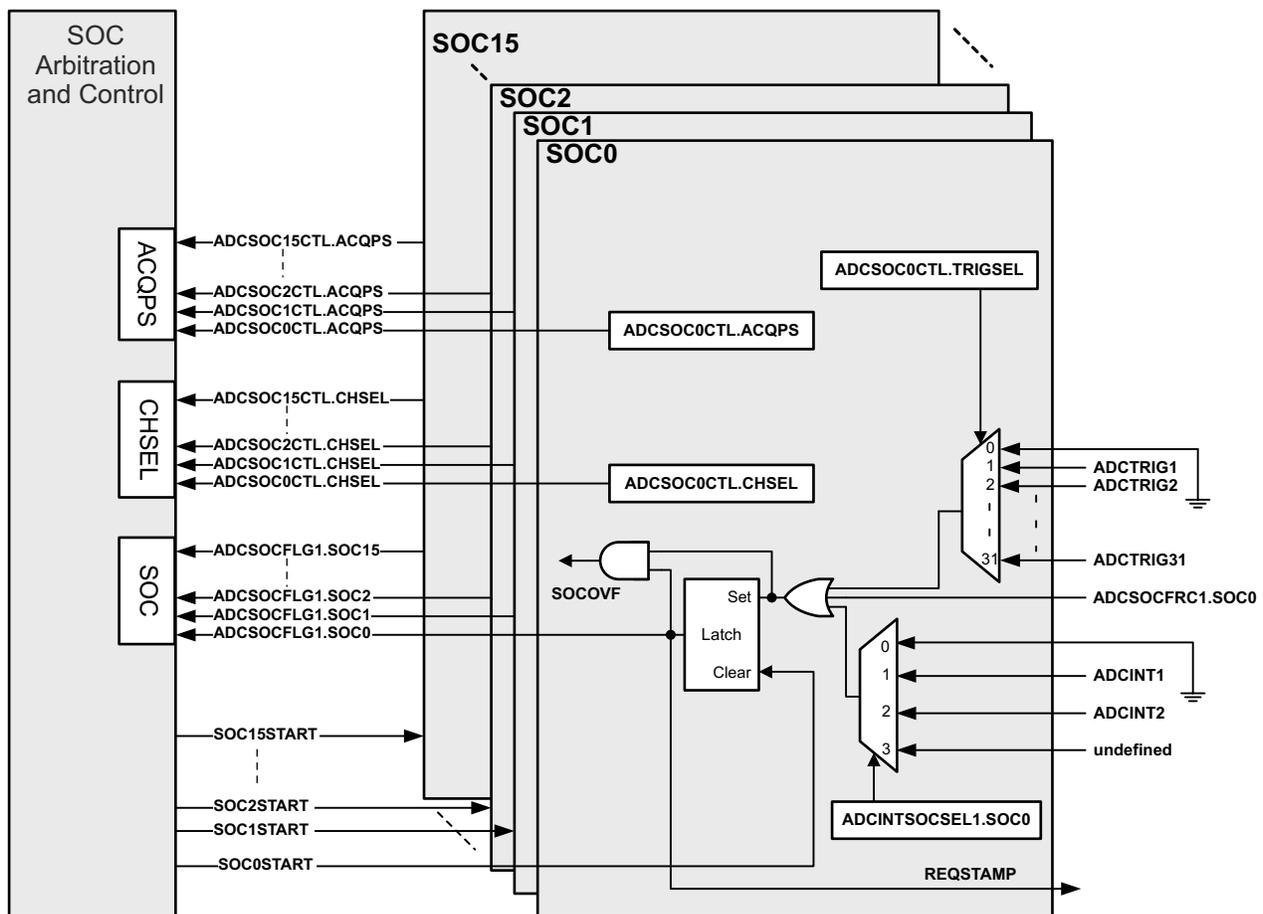


図 1-2. SOC の機能ブロック図

1.3.1 SOC 構成

各 SOC には、専用の構成レジスタ ADCSOCxCTL があります。このレジスタで、SOCx のトリガ・ソース、変換するチャンネル、およびアキュイジション (サンプル) ウィンドウ幅を構成できます。

1.3.2 トリガの動作

各 SOC は、多くの入力トリガのいずれかで開始するように構成できます。SOCx の主なトリガは、ADCSOCxCTL.TRIGSEL レジスタで選択します。以下のいずれかを選択できます。

- デイセーブル (ソフトウェアのみ)
- CPU タイマ 0/1/2
- GPIO : 入力クロスバー INPUT5
- 各 ePWM モジュールからの ADCSOCA または ADCSOCB

また、ADCINT1 フラグまたは ADCINT2 フラグが設定されたときに、各 SOC をトリガすることもできます。これには、ADCINTSOCSEL1 レジスタ (SOC0 ~ SOC7) または ADCINTSOCSEL2 レジスタ (SOC8 ~ SOC15) を構成します。これは、連続変換を作成する場合に便利です。

1.3.3 ADC アキュイジション (サンプル / ホールド) ウィンドウ

外部信号源によって、アナログ信号を迅速かつ効果的に駆動する能力は異なります。定格分解能を達成するには、信号源から ADC コアのサンプリング・コンデンサを信号電圧の 0.5LSB 以内に充電する必要があります。アキュイジション・ウィンドウは、サンプリング・コンデンサを充電可能な時間の長さで、SOCx の ADCSOCxCTL-ACQPS レジスタで構成できます。

ACQPS は 9 ビットのレジスタで、0 ~ 511 の値に設定できます。この値から、アキュイジション・ウィンドウの幅は次のようになります。

アキュイジション・ウィンドウ = (ACQPS + 1) (システム・クロック (SYSCLK) のサイクル時間)

- アキュイジション・ウィンドウの幅は、ADC クロック (ADCCLK) ではなく、システム・クロック (SYSCLK) に基づきます。
- 選択したアキュイジション・ウィンドウの幅は、1 ADCCLK サイクル以上である必要があります。
- データ・マニュアルに、最小のアキュイジション・ウィンドウ幅 (ナノ秒) が規定されています。この要件を満たすアキュイジション・ウィンドウ幅を選択するのは、ユーザーの責任です。

1.3.4 ADC 入力モデル

シングルエンド動作の場合、シングルエンド入力モデルの値に対する ADC 入力特性 (図 1-3 を参照) は、デバイスのデータ・マニュアルに記載されています。

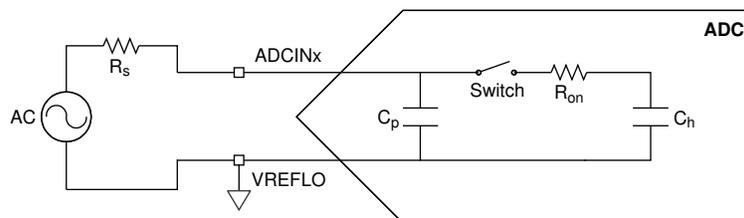


図 1-3. シングルエンド入力モデル

この入力モデルは、アキュイジション・ウィンドウの幅を決定する際に、実際の信号源インピーダンスと合わせて使用できます。詳細については、[セクション 1.13.2](#) を参照してください。

1.3.5 チャンネルの選択

各 SOC は、どの ADC チャンネルを変換するようにも構成できます。これは、SOCx の ADCSOCxCTL.CHSEL レジスタで選択します。表 1-4 に、上述の内容を示します。

表 1-4. 入力ピンのチャンネル選択

入力モード	CHSEL	入力
シングル・エンド	0	ADCIN0
	1	ADCIN1
	2	ADCIN2
	3	ADCIN3
	4	ADCIN4
	5	ADCIN5
	6	ADCIN6
	7	ADCIN7
	8	ADCIN8
	9	ADCIN9
	10	ADCIN10
	11	ADCIN11
	12	ADCIN12
	13	ADCIN13
	14	ADCIN14
	15	ADCIN15
	16	ADCIN16
	17	ADCIN17
	18	ADCIN18
	19	ADCIN19
	20	ADCIN20

1.4 SOC の構成例

以下のセクションでは、変換を生成するために SOC を構成する方法の具体的な例をいくつか示します。

1.4.1 ePWM トリガからの単一変換

ePWM タイマが期間に一致したときにチャンネル ADCINA1 で変換を 1 回実行するように ADCA を構成するには、いくつかの設定が必要です。まず、ePWM3 を SOCA または SOCB 信号を生成するように構成する必要があります (この文では SOC は ePWM モジュールの信号)。この方法については、「拡張パルス幅変調器モジュール (ePWM)」の章を参照してください。SOCB を選択したとします。

SOC5 を任意に選択します。16 個の SOC のどれでも使用できます。

100ns のサンプル・ウィンドウが必要で、SYSCLK 周波数が 120MHz であるとする、アキュイジション・ウィンドウの幅は $100\text{ns}/8.333\text{ns} = 12$ サイクルになります。したがって、ACQPS フィールドは $12 - 1 = 11$ に設定できます。

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1; //SOC5 converts ADCINA1
AdcaRegs.ADCSOC5CTL.bit.ACQPS = 11; //SOC5 uses a sample duration of 12 SYSCLK cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 10; //SOC5 begins conversion on ePWM3 SOCB
```

このように構成すると、ePWM3 が期間に一致し、SOCB 信号が生成されたときに、ADC がアイドル状態であれば、ADC が直ちにチャンネル ADCINA1 (SOC5) のサンプリングを開始します。ADC がビジーの場合、SOC5 が優先度を獲得したときに、ADCINA1 のサンプリングが開始します (セクション 1.5 を参照)。ADC 制御ロジックは、指定したアキュイジション・ウィンドウ幅 100ns で ADCINA1 をサンプリングします。アキュイジションが完了した直後に、ADC がサンプリングされた電圧をデジタル値に変換し始めます。ADC 変換が完了すると、結果は ADCRESULT5 レジスタから取得できます (正確なサンプリング、変換、結果のラッチのタイミングについては、セクション 1.12 を参照)。

1.4.2 ePWM トリガからのオーバーサンプリング変換

ADCINA1 を 4 回オーバーサンプリングするように ADC を構成するには、前の例と同じ構成を SOC5、SOC6、SOC7、SOC8 に適用します。

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1; //SOC5 converts ADCINA1
AdcaRegs.ADCSOC5CTL.bit.ACQPS = 9; //SOC5 uses a sample duration of 10 SYSCLK cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 10; //SOC5 begins conversion on ePWM3 SOCB
AdcaRegs.ADCSOC6CTL.bit.CHSEL = 1; //SOC6 converts ADCINA1
AdcaRegs.ADCSOC6CTL.bit.ACQPS = 9; //SOC6 uses a sample duration of 10 SYSCLK cycles
AdcaRegs.ADCSOC6CTL.bit.TRIGSEL = 10; //SOC6 begins conversion on ePWM3 SOCB
AdcaRegs.ADCSOC7CTL.bit.CHSEL = 1; //SOC7 converts ADCINA1
AdcaRegs.ADCSOC7CTL.bit.ACQPS = 9; //SOC7 uses a sample duration of 10 SYSCLK cycles
AdcaRegs.ADCSOC7CTL.bit.TRIGSEL = 10; //SOC7 begins conversion on ePWM3 SOCB
AdcaRegs.ADCSOC8CTL.bit.CHSEL = 1; //SOC8 converts ADCINA1
AdcaRegs.ADCSOC8CTL.bit.ACQPS = 9; //SOC8 uses a sample duration of 10 SYSCLK cycles
AdcaRegs.ADCSOC8CTL.bit.TRIGSEL = 10; //SOC8 begins conversion on ePWM3 SOCB
```

このように構成すると、ePWM3 が期間に一致し、SOCB 信号が生成されたときに、ADC がアイドル状態であれば、ADC が直ちにチャンネル ADCINA1 (SOC5) のサンプリングを開始します。ADC がビジーの場合、SOC5 が優先度を獲得したときに、ADCINA1 のサンプリングが開始します (セクション 1.5 を参照)。SOC5 の変換が完了すると、SOC6 が ADCINA1 の変換を開始し、SOC5 の結果が ADCRESULT5 レジスタに格納されます。4 つの変換すべてが順に完了し、SOC5、SOC6、SOC7、SOC8 の結果がそれぞれ ADCRESULT5、ADCRESULT6、ADCRESULT7、ADCRESULT8 に格納されます。

注

ePWM トリガが受信されたときのラウンド・ロビン・ポインタの位置によっては、ADC が SOC5 より先に SOC6、SOC7、SOC8 の変換を開始する可能性があります、それはまれです。変換する次の SOC の選択方法については、セクション 1.5 を参照してください。

1.4.3 CPU タイマ・トリガからの複数変換

この例では、異なるアキュイジション・ウィンドウ要件で複数の信号をサンプリングする方法を示します。トリガの生成には、CPU1 のタイマ 2 を使用します。CPU タイマの構成方法については、「システム制御と割り込み」の章を参照してください。

多くの信号を使用したサンプリング方式を設計する最初の手順は、信号とそれらに必要なアキュイジション・ウィンドウを列挙することです。これから、各信号に必要な SYSCLK サイクル数を計算し、ACQPS レジスタの設定を求めます。これを表 1-5 に示します。SYSCLK は 120MHz であると想定しています (サイクル時間 8.333ns)。

表 1-5. 複数信号サンプリングの要件例

信号名	アキュイジション・ウィンドウ要件 (ns)	アキュイジション・ウィンドウの SYSCLK サイクル数	ACQPS レジスタ値
信号 1	> 200ns	200ns/8.333ns = 24	24 - 1 = 23
信号 2	> 740ns	740ns/8.333ns = 89 (切り上げ)	89 - 1 = 88
信号 3	> 183.33ns	183.333ns/8.333ns = 22	22 - 1 = 21
信号 4	> 485ns	485ns/8.333ns = 59 (切り上げ)	59 - 1 = 58

次に、各信号に接続する ADC ピンを決定します。これは、アプリケーション・ボードのレイアウトに大きく依存します。ピンを選択すれば、CHSEL の値を決定するのは簡単です (表 1-6 を参照)。

表 1-6. 複数信号サンプリングの接続例

信号名	ADC ピン	CHSEL レジスタ値
信号 1	ADCINA5	5
信号 2	ADCINA0	0
信号 3	ADCINA3	3
信号 4	ADCINA2	2

この表形式の情報を使用すると、SOC 構成を簡単に生成できます。

```

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 5;           //SOC0 converts ADCINA5
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 23;          //SOC0 uses a sample duration of 24 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 3;         //SOC0 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC1CTL.bit.CHSEL = 0;           //SOC1 converts ADCINA0
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 88;          //SOC1 uses a sample duration of 89 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 3;         //SOC1 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC2CTL.bit.CHSEL = 3;           //SOC2 converts ADCINA3
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 21;          //SOC2 uses a sample duration of 22 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 3;         //SOC2 begins conversion on CPU1 Timer 2
AdcaRegs.ADCSOC3CTL.bit.CHSEL = 2;           //SOC3 converts ADCINA2
AdcaRegs.ADCSOC3CTL.bit.ACQPS = 58;          //SOC3 uses a sample duration of 59 SYSCLK cycles
AdcaRegs.ADCSOC3CTL.bit.TRIGSEL = 3;         //SOC3 begins conversion on CPU1 Timer 2

```

この構成では、CPU1 のタイマ 2 がイベントを生成すると、SOC0、SOC1、SOC2、SOC3 がこの順でサンプリングおよび変換されます。ADCINA5 (信号 1) の変換結果は、ADCRESULT0 に格納されます。同様に、ADCINA0 (信号 2)、ADCINA3 (信号 3)、ADCINA2 (信号 4) の結果はそれぞれ ADCRESULT1、ADCRESULT2、ADCRESULT3 に格納されます。

注

CPU タイマ・トリガが受信されたときのラウンド・ロビン・ポインタの位置によっては、ADC が SOC0 より先に SOC1、SOC2、SOC3 の変換を開始する可能性があります。それはまれです。変換する次の SOC の選択方法については、[セクション 1.5](#) を参照してください。

1.4.4 SOC のソフトウェア・トリガ

特定のトリガを受信するように SOC を構成しているかどうかにかかわらず、いつでもソフトウェア・トリガを設定して SOC を変換できます。これには、ADCSOCFRC1 レジスタのビットを書き込みます。

前の例で CPU1 タイマ 2 でトリガが生成されるのを待機せずにソフトウェアでトリガするには、次の文を使用します。

```
AdcaRegs.ADCSOCFRC1.a11 = 0x000F; //set SOC flags for SOC0 to SOC3
```

1.5 ADC 変換の優先度

複数の SOC フラグを同時に設定した場合、2 種類の優先度決定方式のいずれかによって変換後の順序が決まります。デフォルトの優先度決定方式はラウンド・ロビンです。この方式では、どの SOC も本質的にほかの SOC より優先されるということはありません。優先度は、ラウンド・ロビン・ポインタ (RRPOINTER) によって決まります。ADCSOCPRIORITYCTL レジスタの RRPOINTER は、最後に変換された SOC をポイントします。次に優先となるのは RRPOINTER 値よりも次に大きい値の SOC で、SOC15 に達すると SOC0 に戻ります。0 は変換がすでに実行されたことを示すため、リセット時の値は 16 です。RRPOINTER が 16 の場合、SOC0 が最優先となります。RRPOINTER は、ADC モジュールがリセットされたとき、または SOCPRIORITYCTL レジスタが書き込まれたときにリセットされます。ADC モジュールをリセットするには、ADC インスタンスに対応する SOFTPRES ビットを書き込んでクリアします。

図 1-4 に、ラウンド・ロビン優先度決定方式の例を示します。

ADCSOCPRIORITYCTL レジスタの SOCPRIORITY フィールドを使用して、1 つの SOC からすべての SOC に高優先度を割り当てることができます。ある SOC を高優先度に設定すると、電流変換の完了後にラウンド・ロビンの輪に割り込みが発生し、その SOC が次に変換されます。変換が完了すると、割り込みの発生により中断したラウンド・ロビンの輪が再開します。高優先度の 2 つの SOC が同時にトリガされると、番号が低い方の SOC が優先されます。

最初に SOC0 に高優先度モードが割り当てられ、その後は昇順に割り当てられます。SOCPRIORITY フィールドに書き込む値によって、高優先度でない最初の SOC が定義されます。SOCPRIORITY に値 4 を書き込んだ場合、SOC0、SOC1、SOC2、SOC3 が高優先度として定義され、SOC0 の優先度が最高になります。

図 1-5 に、高優先度 SOC を使用した例を示します。

- A** After reset, SOC0 is highest priority SOC ;
SOC7 receives trigger ;
SOC7 configured channel is converted immediately .
- B** RRPOINTER changes to point to SOC 7 ;
SOC8 is now highest priority SOC .
- C** SOC2 & SOC12 triggers rcvd . simultaneously ;
SOC12 is first on round robin wheel ;
SOC12 configured channel is converted while
SOC2 stays pending .
- D** RRPOINTER changes to point to SOC 12 ;
SOC2 configured channel is now converted .
- E** RRPOINTER changes to point to SOC 2 ;
SOC3 is now highest priority SOC .

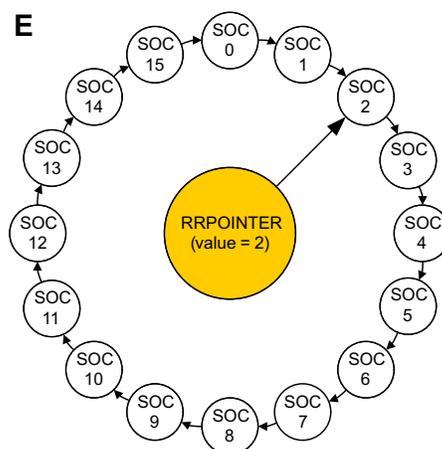
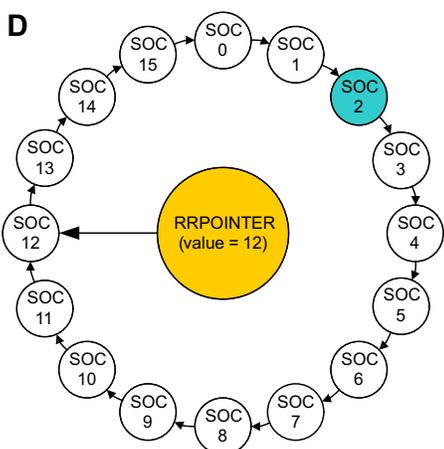
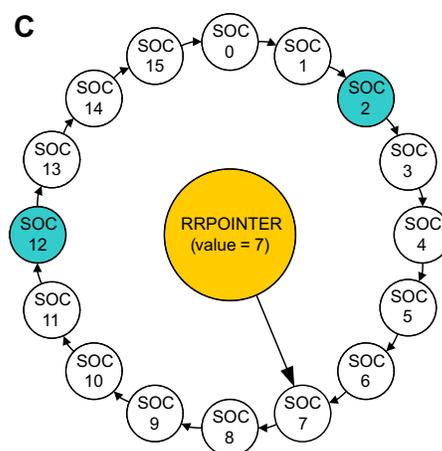
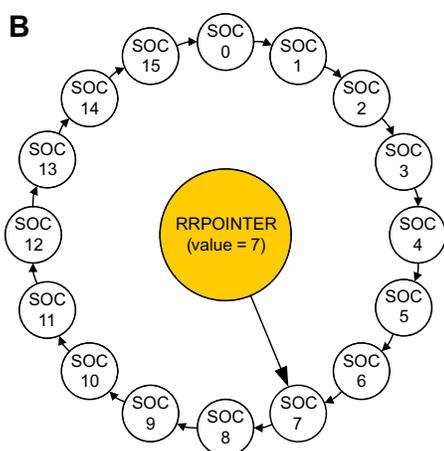
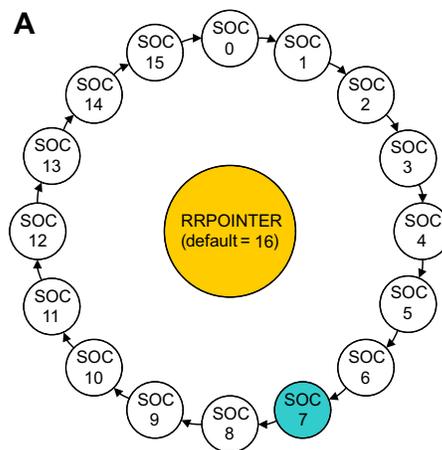


図 1-4. ラウンド・ロビン優先度決定方式の例

Example when SOC PRIORITY = 4

- A After reset, SOC4 is 1st on round robin wheel ;
SOC7 receives trigger ;
SOC7 configured channel is converted immediately .
- B RRPOINTER changes to point to SOC 7 ;
SOC8 is now 1st on round robin wheel .
- C SOC2 & SOC12 triggers rcvd. simultaneously ;
SOC2 interrupts round robin wheel and SOC 2 configured channel is converted while SOC 12 stays pending .
- D RRPOINTER stays pointing to 7 ;
SOC12 configured channel is now converted .
- E RRPOINTER changes to point to SOC 12 ;
SOC13 is now 1st on round robin wheel .

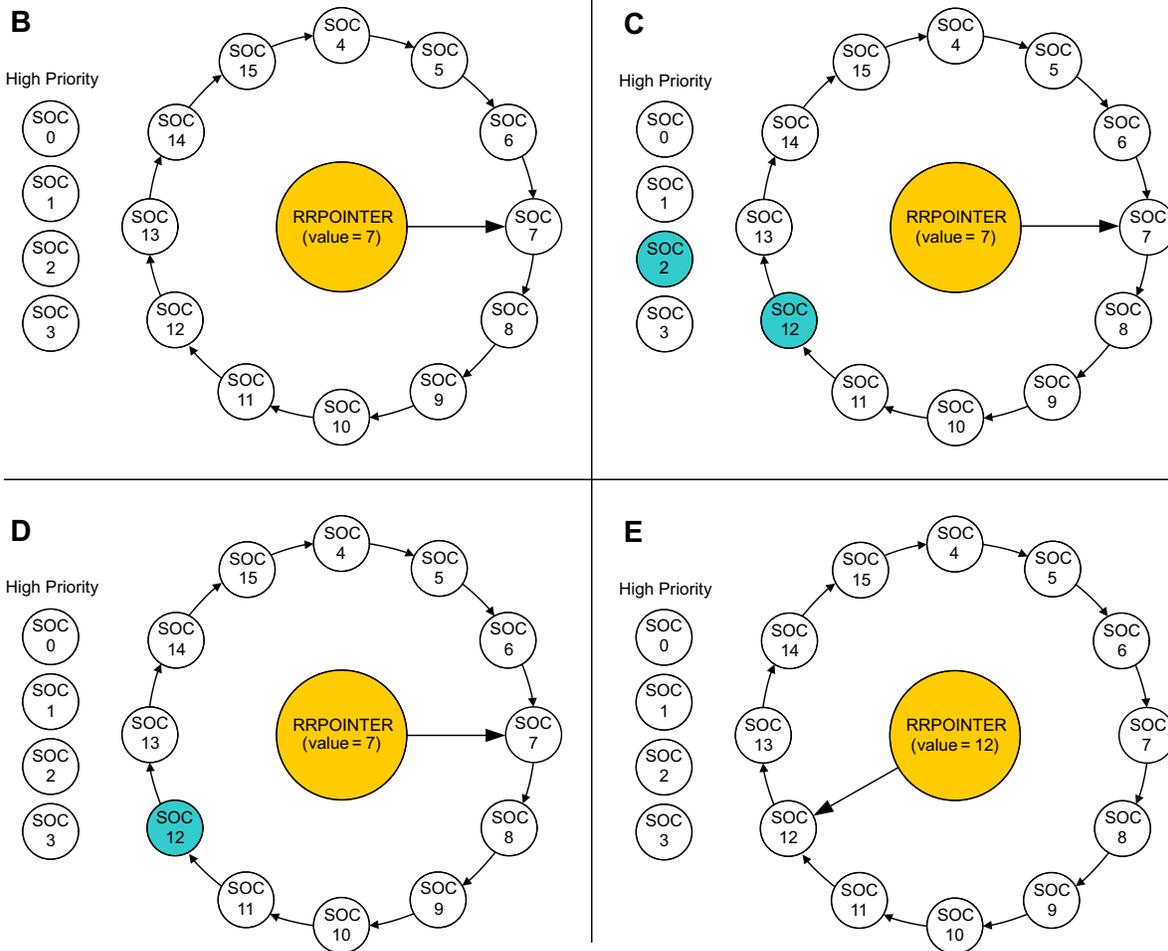


図 1-5. 高優先度を設定した例

1.6 バースト・モード

バースト・モードでは、1つのトリガを使用してラウンド・ロビン SOC を一度に1つ以上変換できます。ADCBURSTCTL レジスタのビット BURSTEN を設定すると、ADC ラッパーがバースト・モード用に設定されます。この設定では、ラウンド・ロビン動作に構成されている (高優先度ではない) SOC の TRIGSEL フィールドは無視されます。ラウンド・ロビン SOC はすべて、TRIGSEL フィールドではなく、ADCBURSTCTL レジスタの BURSTTRIG フィールドに基づいてトリガされます。バースト・トリガが受信されると、ADC ラッパーはすべてのラウンド・ロビン SOC を変換するように設定するのではなく、(ADCBURSTCTL.BURSTSIZE + 1) 個の SOC のみを設定します。最初にラウンド・ロビン・ポインタに基づく最優先の SOC が設定され、BURSTSIZE で指定されている数の SOC が設定されるまでそれに続く SOC が設定されます。

注

ADC をバースト・モードに設定する場合、次のバースト・トリガが受信される前に、各バースト変換が確実に完了するようにするのはユーザーの責任です。(ADCBURSTCTL.BURSTSIZE + 1) の値は、ラウンド・ロビン方式に設定された SOC の数以下である必要があります。

たとえば SOC PRIORITY = 12 (SOC12、SOC13、SOC14、SOC15 がラウンド・ロビン方式) の場合、バースト・モードを正しく動作させるには、ADCBURSTCTL.BURSTSIZE は 3 以下に設定する必要があります。

1.6.1 バースト・モードの例

バースト・モードを使用して、1つおきのトリガで異なる信号セットをサンプリングできます。次の例では、ADCIN7 と ADCIN5 は CPU1 のタイマ 2 からの最初のトリガで変換され、それ以降 1つおきのトリガで変換されます。ADCIN2 と ACIN3 は CPU1 タイマ 2 からの 2 番目のトリガで変換され、それ以降 1つおきのトリガで変換されます。すべての信号は 20 SYSCLK サイクル幅のアクイジション・ウィンドウで変換されますが、必要に応じて各 SOC に異なる幅を設定できます。

```
AdcaRegs.BURSTCTL.BURSTEN = 1;           //Enable ADC burst mode
AdcaRegs.BURSTCTL.BURSTTRIG = 3;         //CPU1 Timer 2 will trigger burst of conversions
AdcaRegs.BURSTCTL.BURSTSIZE = 1;        //conversion bursts are 1 + 1 = 2 conversions long
AdcaRegs.SOCPRICL.bit.SOC PRIORITY = 12; //SOC0 to SOC11 are high priority
AdcaRegs.ADCSOC12CTL.bit.CHSEL = 7;     //SOC12 will convert ADCINA7
AdcaRegs.ADCSOC12CTL.bit.ACQPS = 19;    //SOC12 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC13CTL.bit.CHSEL = 5;     //SOC13 will convert ADCINA5
AdcaRegs.ADCSOC13CTL.bit.ACQPS = 19;    //SOC13 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC14CTL.bit.CHSEL = 2;     //SOC14 will convert ADCINA2
AdcaRegs.ADCSOC14CTL.bit.ACQPS = 19;    //SOC14 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC15CTL.bit.CHSEL = 3;     //SOC15 will convert ADCINA3
AdcaRegs.ADCSOC15CTL.bit.ACQPS = 19;    //SOC15 will use sample duration of 20 SYSCLK cycles
```

CPU1 タイマ 2 から最初のトリガが受信されると、ADC がアイドル状態であれば SOC12 と SOC13 は直ちに変換されます。ADC がビジーの場合、SOC12 と SOC13 はそれぞれが最優先になった時点で変換されます。SOC12 と SOC13 の結果は、それぞれ ADCRESULT12 と ADCRESULT13 に格納されます。SOC13 の完了後、ラウンド・ロビン・ポインタは SOC14 に最優先を割り当てます。そのため、CPU1 タイマ 2 から次のトリガが受信されると、SOC14 と SOC15 が保留になり、変換されます。SOC14 と SOC15 の結果は、それぞれ ADCRESULT14 と ADCRESULT15 に格納されます。それ以降のトリガでは、SOC12 と SOC13 の変換と SOC14 と SOC15 の変換が交互に実行されます。

上の例では 2 つの変換セットが交互に実行されますが、同様の方法で 3 つ以上の変換を順繰りに実行することもできます。

1.6.2 バースト・モードと高優先度を使用した例

図 1-6 に、バースト・モードと高優先度の SOC を使用した場合の優先度の例を示します。

Example when SOCPRRIORITY = 4, BURSTEN = 1, and BURSTSIZE = 1

- A After reset, SOC4 is 1st on round robin wheel; BURSTTRIG trigger is received; SOC4 & SOC5 are set and configured channels converted immediately.
- B RRPOINTER changes to point to SOC5; SOC6 is now 1st on round robin wheel.
- C BURSTTRIG & SOC1 triggers rcvd. simultaneously; SOC1, SOC6, and SOC7 are set; SOC1 interrupts round robin wheel and SOC1 configured channel is converted while SOC6 and SOC7 stay pending.
- D RRPOINTER stays pointing to 5; SOC6/SOC7 configured channels are now converted.
- E RRPOINTER changes to point to SOC7; SOC8 is now 1st on round robin wheel, waiting for BURSTTRIG.

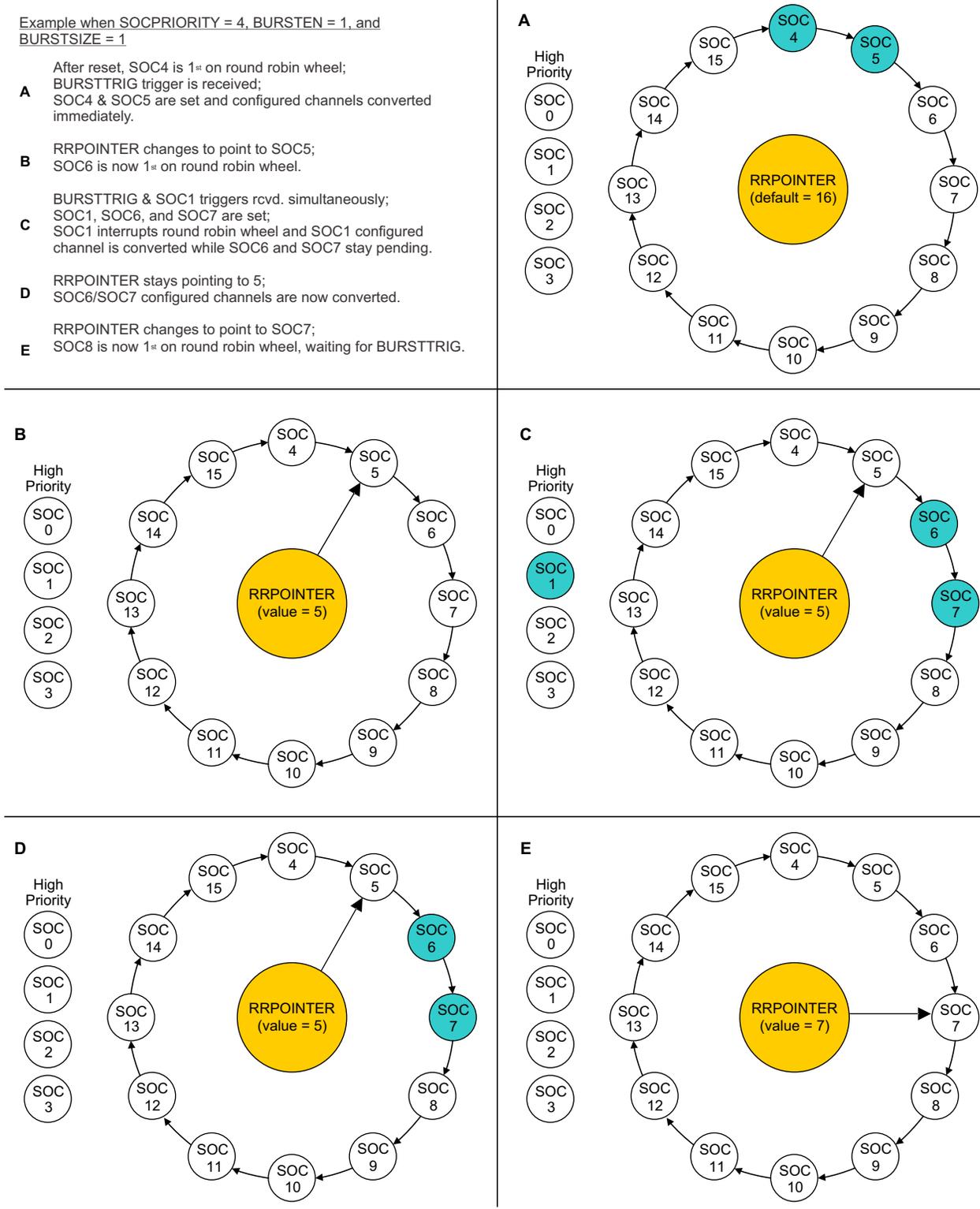


図 1-6. バースト・モードと高優先度を使用した例

1.7 EOC と割り込み動作

各 SOC には、対応する変換終了 (EOC) 信号があります。この EOC 信号を使用して、ADC 割り込みをトリガできます。EOC パルスをアキュイジション・ウィンドウの最後または電圧変換の終了時に生成するように、ADC を構成できます。これは、ADCCTL1 レジスタのビット INTPULSEPOS を使用して構成します。EOC パルスの正確な位置については、[セクション 1.12](#) を参照してください。

各 ADC モジュールには、構成可能な ADC 割り込みが 4 つあります。これらの割り込みは、個々の EOC 信号のいずれかでトリガできます。各 ADCINT のフラグ・ビットを直接読み取って、関連する SOC が完了したが、割り込みを PIE に渡すことができるかを判定できます。

注

ADCCTL1.ADCBSY ビットがクリアの場合、一連の SOC のすべての変換が完了したことを示しているわけではなく、ADC が次の変換を処理する準備ができていないことを示します。一連の SOC が完了したかどうかを判定するには、シーケンス内の最後の SOC に ADCINT フラグをリンクし、その ADCINT フラグを監視します。

ADC 割り込み構造のブロック図を、[図 1-7](#) に示します。

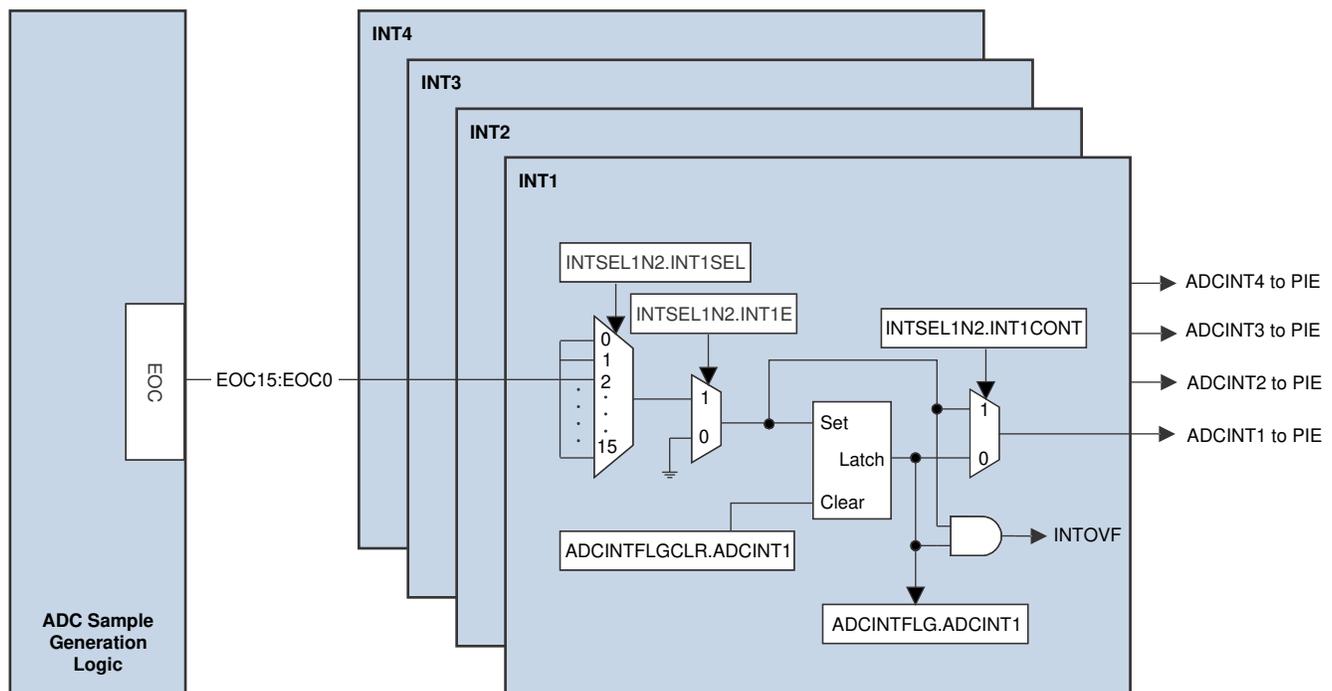


図 1-7. ADC の EOC 割り込み

1.7.1 割り込みオーバーフロー

EOC 信号が ADCINTFLG レジスタのフラグを設定するときに、そのフラグがすでに設定されている場合、割り込みオーバーフローが発生します。デフォルトでは、オーバーフロー割り込みは PIE モジュールに渡されません。ADCINTFLG レジスタの特定のフラグでオーバーフローが発生すると、ADCINOVF レジスタの対応するフラグが設定されます。このオーバーフローフラグは、オーバーフローが発生したことを検出するためにのみ使用され、それ以上の割り込みが PIE モジュールに渡されることはブロックしません。

ADC 割り込みオーバーフローが発生すると、アプリケーションは ISR 内またはバックグラウンドループ内で適切な ADCINOVF フラグをチェックし、オーバーフローが検出された場合は適切な操作を実行します。次のコードスニペットに、ADCINT フラグをクリアしようとした後に ISR 内で ADCINOVF フラグをチェックする方法を示します。

```
// Clear the interrupt flag
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;    //clear INT1 flag for ADC-A

// Check if an overflow has occurred
if(1 == AdcaRegs.ADCINOVF.bit.ADCINT1)    //ADCINT overflow occurred
{
    AdcaRegs.ADCINOVFCLR.bit.ADCINT1 = 1  //Clear overflow flag
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1 //Re-clear ADCINT flag
}
```

```
//
// Clear the interrupt flag
//
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);

//
// Check if an overflow has occurred
//
if(true == ADC_getInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1))
{
    ADC_clearInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1);
    ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
```

1.7.2 割り込みを続行モード

ADCINTSEL1N2 および ADCINTSEL3N4 レジスタの INTxCONT ビットは、ADCINTFLG が前の割り込みからまだクリアされていない場合に、割り込みをどのように処理するかを構成します。このモードはデフォルトでディセーブルになっており、追加の重複する割り込みは PIE に対して発行されません。このモードをアクティブにすると、ADC 割り込みは常に PIE に渡されます。この構成にかかわらず、ADCINTFLG が設定されているときに割り込みが発生した場合、ADCINOVF レジスタは設定されたままになります。

1.7.3 Early 割り込み構成モード

Early 割り込みモードをイネーブルにすると、ADC 結果が使用可能になる前に、アプリケーションが ADC 割り込みサービスルーチンに入ることができます。これにより、ADC の結果が使用可能になったときにすぐにアプリケーションで ADC 結果を処理できるように、アプリケーションで必要な事前処理を行うことができます。ただし、Early 割り込みのタイミングが早すぎ、アプリケーションが更新された ADC の結果を使用できるようになるまで時間を浪費することになる場合があります。この状況を回避するため、Early 割り込みモードのときに ADC 割り込みに移行する時間を、ADCINTCYCLE レジスタの DELAY フィールドで構成できます。

- この割り込み時間を設定するには、ADC が Early 割り込みモードである必要があります。これには、ADCCTL1 のビット INTPULSEPOS を 0 にクリアします。
- ADCINTCYCLE レジスタの DELAY 値は、SOC パルスの立ち下がりエッジ後、ADCINT フラグが設定されるまでの追加の SYSCLK サイクル数を設定します。
- DELAY 値が EOC を超える場合、ADC 割り込みと EOC が生成されます。

- INTPULSEPOS が 1 に設定されているときに DELAY に値を書き込んでも、割り込み生成に影響はありません。

1.8 後処理ブロック

各 ADC モジュールには、4 つの後処理ブロック (PPB) が含まれています。これらのブロックは、ADCPPBxCONFIG.CONFIG ビット・フィールドを使用して、個の RESULT レジスタのいずれかに関連付けることができます。後処理ブロックには、以下の機能があります。

- ADCIN チャンネルに関連付けられているオフセットを削除
- リファレンス値を減算
- ゼロクロス・ポイントをフラグ (PWM をトリップして割り込みを生成するオプションあり)
- 比較結果の上限または下限をフラグ (PWM をトリップして割り込みを生成するオプションあり)
- 関連する SOC トリガからサンプリングが実際に開始するまでの遅延時間を記録

図 1-8 に、各 PPB の構造を示します。以下のセクションでは、各サブモジュールの使用方法について説明します。

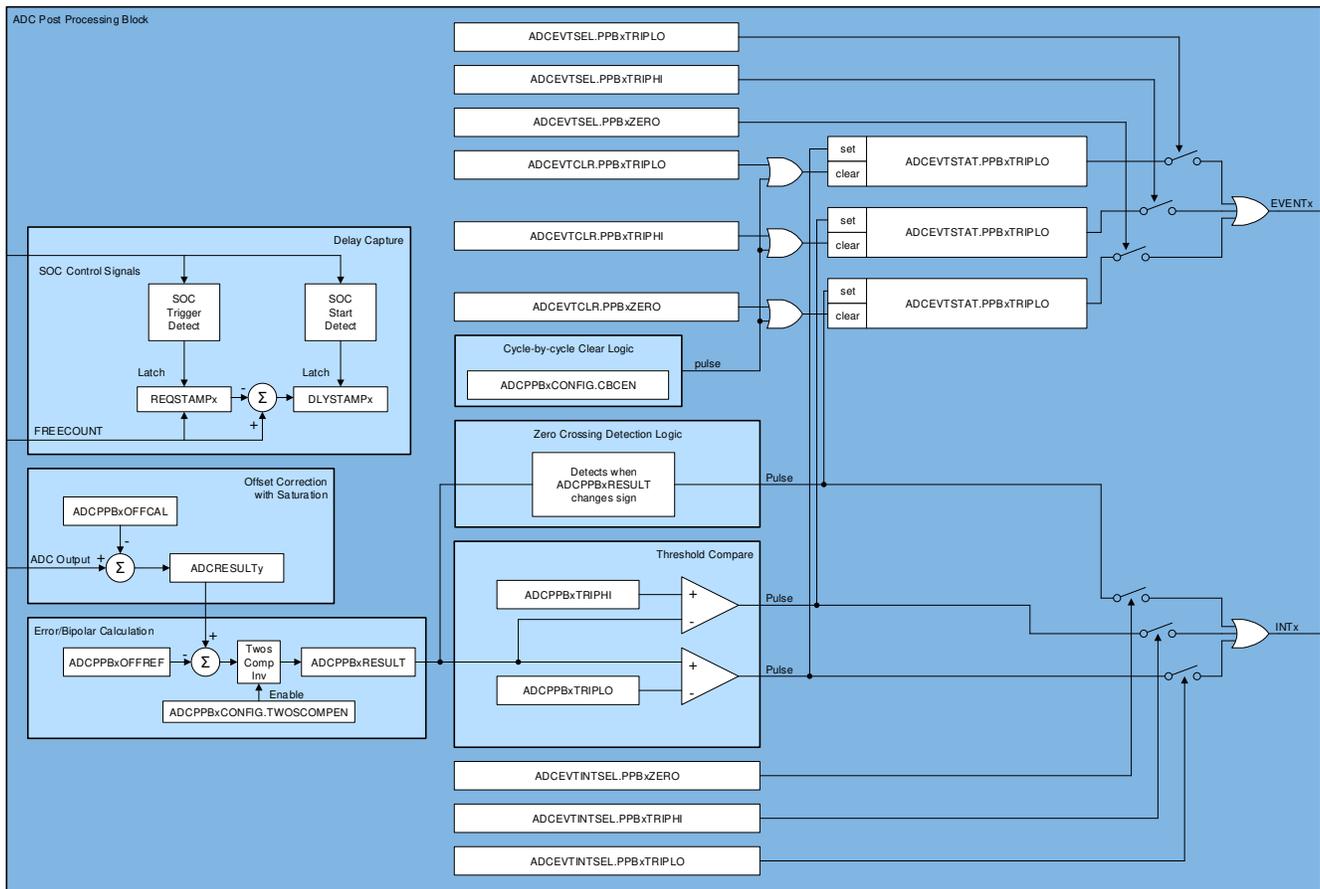


図 1-8. ADC PPB のブロック図

1.8.1 PPB オフセット補正

多くのアプリケーションでは、外部センサと信号源によりオフセットが生成されます。これらのオフセットはチャンネルごとに異なるので、ADC オフセットのグローバル・トリミングはこれらを補正するのに十分ではありません。後処理ブロックはオーバーヘッドなしでこれらのオフセットを除去できるため、厳密な制御ループで多数のサイクルを節約できます。

オフセットを補正するには、まず ADCPPBxCONFIG.CONFIG を目的の SOC に設定し、ADCPPBxOFFCAL.OFFCAL レジスタにオフセット補正値を書き込みます。後処理ブロックは、未処理の変換結果に OFFCAL レジスタの値を自動的に加算または減算し、その値を ADCRESULT レジスタに格納します。この加算 / 減算は、下限では 0、上限では 4095 で飽和します。

注

- OFFCAL レジスタに 0 を書き込むと、オフセット補正機能が実質的にディセーブルになり、未処理の結果は変更されずに ADCRESULT レジスタに渡されます。
- 複数の PPB に同じ SOC を設定することもできます。この場合、実際に適用される OFFCAL 値は、最大値の PPB から供給されます。
- PPB を SOC0 に使用する場合は、すべての PPB でこの SOC がデフォルトで設定されているため、注意が必要です。これにより、番号の小さい PPB のオフセット補正が、番号の大きい PPB で上書きされる可能性があります。

1.8.2 PPB の誤差計算

多くのアプリケーションでは、設定ポイントまたは予測値からの誤差を、ADC 変換のデジタル出力から計算する必要があります。それ以外の場合、制御計算には、バイポーラ信号が必要であるか、または有益です。PPB はこれらの機能を自動的に実行できるため、サンプリングから出力までのレイテンシを短縮し、ソフトウェアのオーバーヘッドを低減できます。

誤差を計算するには、まず ADCPPBxCONFIG.CONFIG を目的の SOC に設定し、ADCPPBxOFFCAL.OFFREF レジスタに値を書き込みます。後処理ブロックは、ADCRESULT 値から OFFREF レジスタの値を自動的に減算し、その値を ADCPPBxRESULT レジスタに格納します。この減算により、符号拡張 32 ビット結果が生成されます。ADCPPBxCONFIG レジスタの TWOSCOMPEN ビットを設定することで、計算値を ADCPPBxRESULT レジスタに格納する前に選択的に反転することもできます。

注

- ADCPPBxOFFREF レジスタに 12 ビットを超える値を書き込まないでください。
- ADCPPBxRESULT レジスタは各 PPB に固有であるため、複数の PPB に同じ SOC を設定し、各 PPB で異なる結果を取得できます。
- ADCPPBxOFFREF レジスタに 0 を書き込むと、誤差計算機能が実質的にディセーブルになり、ADCRESULT 値は変更されずに ADCPPBxRESULT レジスタに渡されます。
- ADCPPBxOFFREF に新しい値を書き込むと、ADCPPBxRESULT レジスタは即更新されません。ただし、PPB からのフラグは、次の変換終了 (EOC) まで変更されません。たとえば、ADCPPBxOFFREF レジスタを変更して ADCPPBxRESULT の符号を変更したが、次の変換で結果が OFFREF 変更前と同じ符号に戻る場合、ADCPPBxZERO フラグは設定されません。

1.8.3 PPB の制限値検出とゼロクロス検出

多くのアプリケーションでは、ADC 変換結果の制限値チェックが実行されます。PPB は、上限値と下限値に対して、または ADCPPBxRESULT の符号が変更されたときに、自動的にチェックを実行できます。これらの比較に基づいて、PPB は PWM に対してトリップと割り込みを自動的に生成し、サンプルから ePWM までのレイテンシを短縮して、ソフトウェアのオーバーヘッドを低減できます。この機能により、安全性が重視される

アプリケーションで、ADC 変換結果が範囲外になったときに、CPU の介入なしで ePWM をトリップすることもできます。

この機能をイネーブルにするには、まず ADCPPBxCONFIG.CONFIG を目的の SOC に設定し、レジスタ ADCPPBxTRIPHI.LIMITHI と ADCPPBxTRIPLO.LIMITLO の一方または両方に値を書き込みます (ゼロクロス検出では、これ以外の構成は必要ありません)。これらの制限値を超えると、ADCEVTSTAT レジスタの PPBxTRIPHI ビットまたは PPBxTRIPLO ビットが設定されます。ADCEVTSTAT レジスタの PPBxZERO ビットは、ADCPPBxRESULT レジスタの符号の変化ではなく、変換終了 (EOC) によってゲートされることに注意してください。ADCEVTCLR レジスタには、これらのイベント・フラグをクリアするための対応ビットがあります。ADCEVTSEL レジスタには、イベントを PWM を介して伝搬できるようにする対応ビットがあります。ADCEVTINTSEL レジスタには、イベントを PIE を介して伝搬できるようにする対応ビットがあります。

図 1-9 に示すように、1 つの PIE 割り込みが ADC モジュールのすべての PPB で共有されます。

注

- 同じ ADC モジュールからの異なる PPB イベントに対して異なるアクションを実行する必要がある場合、ADCEVTINT ISR で ADCEVTSTAT レジスタの PPB イベント・フラグを読み取り、割り込みを発生させたイベントを判定する必要があります。
- 上限値比較、下限値比較、ゼロクロスに異なる ePWM トリップを生成する必要がある場合は、複数の PPB を同じ SOC に設定することで、これを実現できます。
- ゼロクロス検出回路では、結果 0 は正と見なされます。

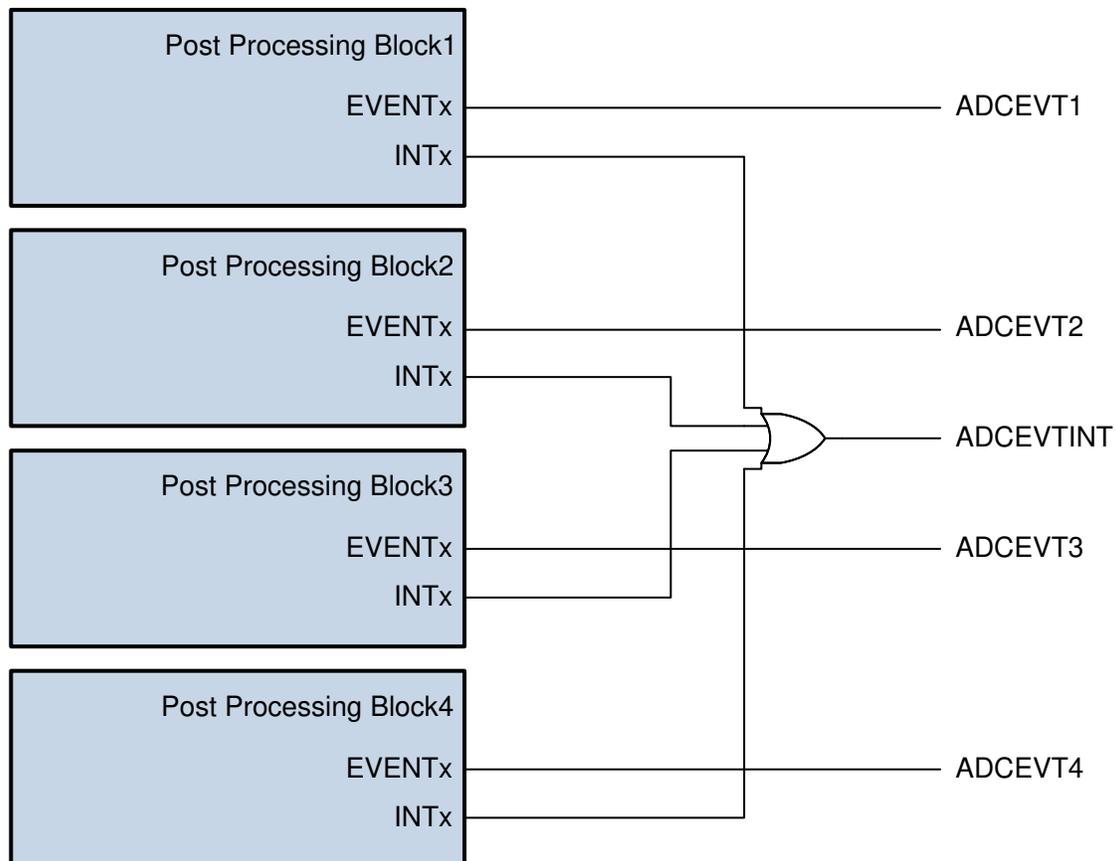


図 1-9. ADC PPB 割り込みイベント

1.8.4 PPB サンプル遅延のキャプチャ

同じ ADC 上で複数の制御ループを非同期で実行する場合、2 つ以上のループからの ADC 要求が競合し、いずれかのサンプルが遅延する可能性があります。これは、システムの測定誤差となります。この遅延の発生のタイミングと遅延時間を把握することにより、ソフトウェアで外挿手法を使用して誤差を低減できます。

このため、各 PPB には ADCPPBxSTAMP レジスタにフィールド DLYSTAMP があります。このフィールドには、関連付けられた SOC がトリガされてから SOC の変換が開始するまでの SYSCLK サイクル数が含まれます。

これは、SYSCLK をベースとするグローバルな 12 ビット・フリー・ランニング・カウンタ (値は ADCCOUNTER レジスタのフィールド FREECOUNT に格納) を使用して実現されます。関連する SOC のトリガが到着すると、このカウンタの値がビット・フィールド ADCPPBxTRIPLO.REQSTAMP にロードされます。その SOC の実際のサンプル・ウィンドウが開始すると、REQSTAMP の値が現在の FREECOUNT 値から減算され、DLYSTAMP に格納されます。

注

SOC トリガから SOC アクイジションの実際に開始するまでに 4096 SYSCLK サイクルを超える時間が経過すると、FREECOUNT レジスタが複数回オーバーフローし、DLYSTAMP 値が不正確なものになる可能性があります。非常に低速な変換を使用する場合は、このような状況を回避する必要があります。

関連する SOC がソフトウェアを使用してトリガされた場合、サンプル遅延キャプチャは機能しません。ただし、異なる SOC のソフトウェア・トリガによって PPB に関連付けられている SOC が遅延された場合、サンプル遅延キャプチャで遅延が正しく記録されます。

1.9 開放 / 短絡検出回路 (OSDETECT)

開放 / 短絡検出回路 (OSDETECT) は、システム内のピン・フォルトを検出するのに使用できます。この回路は、[図 1-10](#) に示すように、チャンネル選択マルチプレクサの後、S+H 回路の前の ADC 入力に接続されます。

注

- 分圧抵抗の許容誤差は大きく異なる可能性があるため、この機能を使用して変換の精度を確認することはできません。
- アナログ入力チャンネルの実装と利用可能性については、データシートを参照してください。
- 駆動インピーダンスが高いため、S+H 期間を ADC の最小値よりもかなり長くする必要があります。

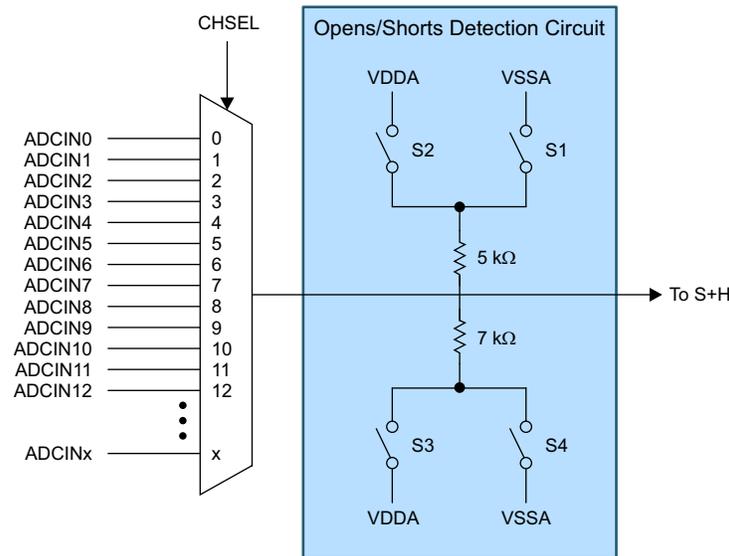


図 1-10. 開放 / 短絡検出回路

この回路は、ADCOSDETECT レジスタの DETECTCFG フィールドに値を書き込むことで動作させることができます。これにより、回路が変換の S+H フェーズ中に入力に電圧を供給します。[表 1-7](#) に、異なる DETECTCFG 設定における OSDETECT 回路の電圧と駆動強度を示します。

表 1-7. DETECTCFG の設定

ADCOSDETECT.DETE CTCFG	ソース電圧	S4	S3	S2	S1	駆動インピーダ ンス
0	Off	オープン	オープン	オープン	オープン	オープン
1	ゼロスケール	クローズド	オープン	オープン	クローズド	5K 7K
2	フルスケール	オープン	クローズド	クローズド	オープン	5K 7K
3	5/12 VDDA	オープン	クローズド	オープン	クローズド	5K 7K
4	7/12 VDDA	クローズド	オープン	クローズド	オープン	5K 7K
5	ゼロスケール	オープン	オープン	オープン	クローズド	5K
6	フルスケール	オープン	オープン	クローズド	オープン	5K
7	ゼロスケール	クローズド	オープン	オープン	オープン	7K

1.9.1 実装

OSDETECT を実装した代表的な回路は、[図 1-11](#) に示すように、直列抵抗 R_S 、シャント・コンデンサ C_P 、等価 OSDETECT 抵抗、 $R_{OSDETECT}$ 、電圧 $V_{OSDETECT}$ で構成されます。これに基づいて、サンプリング・コンデンサに流入する信号レベルを計算できます。 $R_{OSDETECT}$ および $V_{OSDETECT}$ は、OSDETECT 回路から供給される等価入力抵抗および電圧源です。[表 1-7](#) に、異なる構成設定でのこれらの値を示します。OSDETECT 機能がイネーブルのときに信号ソース V_S が駆動している場合の S/H への入力信号を求めるには、[図 1-11](#) を参照してください。

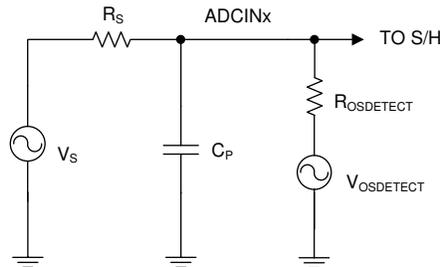


図 1-11. OSDETECT イネーブルと等価の入力回路

入力インピーダンス R_S と C_P は信号ソースに不可欠な要素です。または、信号のプリコンディショニングや、S/H 要件を満たすための信号セトリング時間の制御のために、この設計に実装することもできます。OSDETECT 機能を使用するときは、入力パスを考慮する必要があります。これが、変換結果に影響を及ぼす可能性があります。たとえば、この機能がイネーブルのとき入力信号を駆動すると、信号 V_S が R_S 経由で OSDETECT 回路に接続され、ADC の結果に影響を与えることがあります。 C_P の値を大きくする場合 (数百 pF 以上)、変換前に入力の信号を安定させるため、ACQPS の値を大きくする必要があります。

回路をイネーブルにするには：

1. ADC を変換用に構成します (例：チャンネル、SOC、ACQPS、プリスケラ、トリガなど)。
2. [表 1-7](#) を参照して、目的の分圧器接続に合わせて ADCOSDETECT レジスタを設定します。
3. 変換を開始し、変換結果を調べます。

入力側を駆動しているものと R_S および C_P の値に基づいて結果を解釈します。 V_S 信号を入力ピンから切断できる場合、以下のセクションで説明するように、この回路を使用して入力ピンの開放と短絡を検出できます。

1.9.2 開放入力ピンの検出

さまざまな OSDETECT 設定を順に使用すると、入力信号がソース電圧にプルされます。駆動強度が良好な (ピンが開放されていない) 入力は、ほとんど影響を受けません。ピンが開放されている場合、サンプリングされた電圧は [表 1-7](#) で指定されているソース電圧に近くなります。

1.9.3 短絡入力ピンの検出

さまざまな OSDETECT 設定を順に使用すると、入力信号がソース電圧にプルされます。駆動強度が有限 (ピンが短絡していない) の入力は、各ソース電圧にプルされます。ピンが短絡している場合、信号は同じ電圧のままになります。

1.10 パワーアップ・シーケンス

デバイスの電源投入時またはシステム・レベルのリセット時に、ADC はパワーダウンされ、ディセーブルになります。ADC に電源を投入するときは、以下のシーケンスを使用します。

1. PCLKCR13 レジスタで、使用する ADC クロックをイネーブルにするビットを設定します。
2. ADCCTL2 の PRESCALE フィールドで、使用する ADC クロック分周器を設定します。
3. ADCCTL1 の ADCPWDNZ ビットを設定して、ADC に電源を投入します。
4. 遅延時間を確保してからサンプリングを開始します。必要な時間については、データ・マニュアルを参照してください。

複数の ADC に同時に電源を投入する場合は、手順 1 と手順 3 はすべての ADC に対して 1 つの書き込み命令で実行できます。また、すべての ADC の電源投入が開始した後に遅延を確保するのであれば、必要な遅延は 1 回のみです。

1.11 ADC 較正

テキサス・インスツルメンツでは、製造およびテスト・プロセス中に、ADC のゲイン、オフセット、直線性を較正しています。これらのトリム設定は、テキサス・インスツルメンツの予約済み OTP メモリに組み込まれており、C 呼び出し可能な関数を使用してロードできます。

- Device_cal() 関数は、ADC のトリム値を OTP メモリからそれぞれのトリム・レジスタにコピーします。
- Device_cal() 内のトリム関数は、C2000ware で ADC_setOFFSETRIM()、ADC_setINLTRIM() として呼び出すことができます。これらの関数は、テスト・プロセス中に値が保存されるテキサス・インスツルメンツの予約済み OTP メモリ・ソース位置と、トリム値のコピー先であるアナログ・モジュール・レジスタの宛先からトリム値をフェッチします。

適切な工場出荷時のトリム値がロードされるまでは、ADC およびその他のアナログ・モジュールはデータ・マニュアルの仕様範囲内では動作するよう規定されていません。同様に、工場出荷時の設定以外のトリム値をトリム・レジスタに配置した場合、ADC (および他のモジュール) はデータ・マニュアルの仕様範囲内で動作するよう規定されていません。

ブート ROM により較正関数が呼び出されるので、ユーザーの介入なしでトリム値を初期設定できます。トリム値がモジュールのリセットでクリアされたり何らかの理由で変更された場合は、(ヘッダー・ファイルで定義して) 較正関数を呼び出すことができます。

1.11.1 ADC ゼロ・オフセット較正

ゼロ・オフセット誤差は、VREFLO で電圧を変換するときに発生する 0 との差として定義されます。ゼロ・オフセット誤差は、正の値にも負の値にもなります。この誤差を修正するため、同じ振幅で逆極性の調整が ADCOFFTRIM レジスタに書き込まれます。このレジスタに含まれる値は、結果が ADC 結果レジスタで使用可能になる前に適用されます。この動作は ADC コア内に完全に抱合されているため、結果のタイミングは影響を受けず、どんなトリム値に対しても ADC の全ダイナミック・レンジが維持されます。

GetAdcOffsetTrimOTP(Uint16) 関数を使用すると、ADCOFFTRIM レジスタに工場較正されたオフセット誤差訂正をロードできます。ADCOFFTRIM レジスタを変更することにより、アプリケーション環境によって発生する追加のオフセット誤差を必要に応じて補償できますが、データ・マニュアルで規定された性能を達成するのにこのような変更は通常必要ありません。

注

コンバータの分解能にかかわらず、各 ADCOFFTRIM ステップのサイズは (VREFHI-VREFLO)/65536 となります。

12 ビットのシングルエンド・モードで ADC オフセットを再較正するには、以下の手順に従います。

1. ADCOFFTRIM を +112 ステップ (0x70) に設定します。これにより、ADC コア内に存在する可能性のある負のオフセットを考慮した人工的なオフセットが追加されます。
2. VREFLO (内部接続) で 16 変換を複数回実行し、結果を蓄積します (例: $32 \times 16 = 512$ 回の変換)。
3. 累積結果を 16 の倍数で除算します (例: 512 変換の場合は 32 で除算)。
4. ADCOFFTRIM を 112 から手順 3 の結果を引いた値に設定します。

1.12 ADC タイミング

アナログ電圧をデジタル値に変換するプロセスは、S+H フェーズと変換フェーズに分けられます。ADC のサンプル / ホールド回路 (S+H) には SYSCLK からクロックが供給され、ADC 変換プロセスでは ADCCLK からクロックが供給されます。ADCCLK は、ADCCTL2 レジスタの PRESCALE フィールドに基づいて SYSCLK を分周することにより生成されます。

S+H 期間は、変換される SOC の ACQPS フィールドの値に 1 を加え、SYSCLK 周期を乗算した値です。この期間が、ADCCLK の 1 周期と、データ・マニュアルで規定されている最小 S+H 期間の両方を超えていることを確認する必要があります。変換時間は、約 10.5 ADCCLK サイクルです。正確な変換時間は、常に SYSCLK サイクルの整数倍です。正確なタイミングについては、[セクション 1.12.1](#) のタイミング図と表を参照してください。

1.12.1 ADC のタイミング図

次の図に、以下の前提による 2 つの SOC の ADC 変換タイミングを示します。

- SOC0 および SOC1 は、同じトリガを使用するように構成されています。
- トリガが発生したときに、他の SOC が変換中または保留中になっていることはありません。
- ラウンド・ロビン・ポインタは、最初に SOC0 で変換が発生する状態になっています。
- ADCINTSEL は、SOC0 の変換終了時に ADCINT フラグを設定するように構成されています (このフラグを CPU に通知して割り込みを発生させるかどうかは、PIE モジュールの構成によって決まります)。

[表 1-8](#) に、以下のタイミング図のパラメータを示します。に [表 1-9](#) に、ADC のタイミングを示します。

表 1-8. ADC タイミング・パラメータの説明

パラメータ	説明
t_{SH}	S+H ウィンドウの幅。 このウィンドウの終了時点における S+H コンデンサでの値が、デジタル値に変換される電圧になります。この時間は、(ACQPS + 1) SYSCLK サイクルで与えられます。ACQPS は、各 SOC に対して個別に構成できるため、異なる SOC の t_{SH} は、同じであるとは限りません。 注：デバイスのクロック設定に関係なく、S+H コンデンサでの値は、S+H ウィンドウ終了の約 5ns 前にキャプチャされます。
t_{LAT}	S+H ウィンドウ終了から、ADC 結果が ADCRESULTx レジスタにラッチされるまでの時間。 この時間より前に ADCRESULTx レジスタを読み取った場合、前回の変換結果が返されます。
t_{EOC}	S+H ウィンドウ終了から、次の ADC 変換を開始できる S+H ウィンドウまでの時間。その後は、変換結果がラッチされる前にサンプリングを開始できます。
t_{INT}	S+H ウィンドウの終了から ADCINT フラグが設定されるまでの時間 (そのように構成されている場合)。 ADCCTL1 レジスタの INTPULSEPOS ビットが設定されている場合、 t_{INT} は変換 (EOC) 信号の終了時点と一致します。 INTPULSEPOS ビットが 0 の場合、 t_{INT} は、S+H ウィンドウの終了時点と一致します。 t_{INT} によって ADC 結果レジスタの読み取りが () トリガされる場合、結果がラッチされた後に読み取りが発生するように注意する必要があります (そうでなければ、前回の結果が読み取られます)。 INTPULSEPOS ビットが 0 であり、ADCINTCYCLE レジスタの OFFSET フィールドが 0 でない場合、ADCINT フラグが設定される前に、OFFSET SYSCLK サイクルの遅延が発生します。この遅延時間を使えば、ちょうどサンプリングの用意ができた時点で ISR に入ることができます。

- パラメータ t_{SH} は、S+H ウィンドウの幅です。このウィンドウの終了時点における S+H コンデンサでの値が、デジタル値に変換される電圧になります。この時間は、(ACQPS + 1) SYSCLK サイクルで与えられます。ACQPS は、各 SOC に対して個別に構成できるため、異なる SOC の t_{SH} は、同じであるとは限りません。
- パラメータ t_{LAT} は、S+H ウィンドウ終了から、ADC 変換結果が ADCRESULTx レジスタにラッチされるまでの時間です。この時間より前に ADCRESULTx レジスタを読み取った場合、前回の変換結果が返されます。
- パラメータ t_{EOC} は、S+H ウィンドウの終了から、次の ADC 変換 S+H ウィンドウを開始できるようになるまでの時間です。
- パラメータ t_{INT} は、S+H ウィンドウの終了から ADCINT フラグが設定されるまでの時間です (そのように構成されている場合)。ADCCTL1 レジスタの INTPULSEPOS ビットが設定されている場合、これは変換結果

が結果レジスタにラッチされる時点と一致します。このビットが 0 の場合は、S+H ウィンドウの終了時点と一致します。

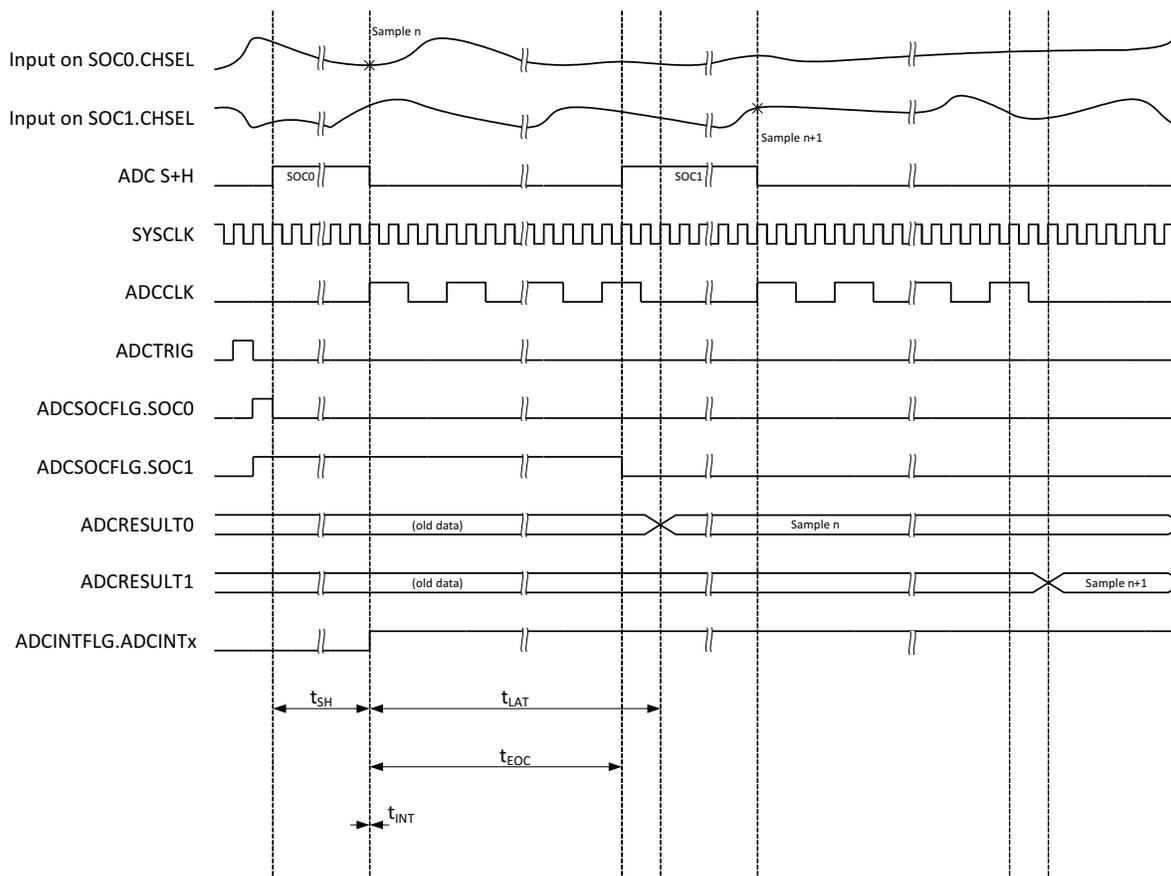


図 1-12. 12 ビット・モードの Early 割り込みモードでの ADC タイミング

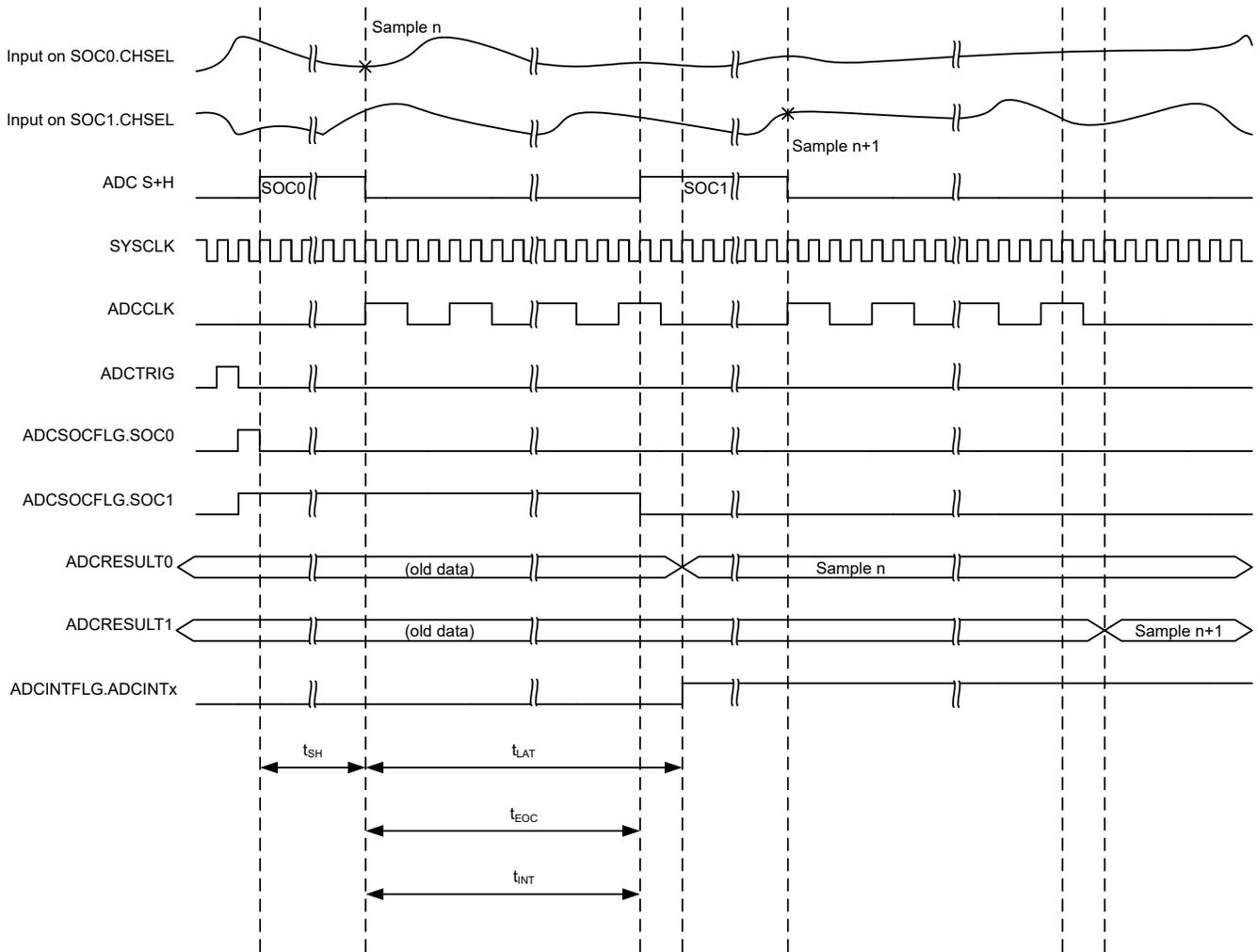


図 1-13. 12 ビット・モードの Late 割り込みモードでの ADC タイミング

表 1-9. 12 ビット・モードでの ADC タイミング

ADCCLK プリスケール		SYSCLK サイクル			
ADCCTL2 プリスケール	プリスケール比	t_{EOC}	t_{LAT}	t_{INT} (Early) ⁽¹⁾	t_{INT} (Late)
0	1	11	13	0	11
2	2	21	23	0	21
4	3	31	34	0	31
6	4	41	44	0	41
8	5	51	55	0	51
10	6	61	65	0	61
12	7	71	76	0	71
14	8	81	86	0	81

(1) デフォルトでは、INTPULSEPOS が 0 の場合、S+H ウィンドウの後 1 SYSCLK サイクル経過してから t_{INT} が発生します。これは、ADCINTCYCLE レジスタの OFFSET フィールドに書き込むことで変更できます。

1.13 追加情報

以下のセクションに、実用的な追加情報を示します。

1.13.1 同期動作の確実な実現

最高の性能を実現するため、デバイス上のすべての ADC を同期動作させることができます。デバイスのデータ・マニュアルでは、動作モードで異なるパラメータに対して、同期モードと非同期モードの性能を規定しています。

同期動作を確実にするため、デバイス上のすべての ADC をロックステップで動作させることができます。これには、すべての ADC のサンプリング・フェーズと変換フェーズが正確に揃うような構成をすべての ADC に書き込みます。最も簡単な方法は、各 ADC のトリガ選択と ACQPS (S+H 期間) の SOC 構成に同じ値を書き込むことです。また、同期 ADC は、SOC の優先制御、バースト・モード、バースト・トリガ、バースト・サイズも同じ値に構成する必要があります。

1.13.1.1 基本的な同期動作

以下の例では、同じトリガ選択と ACQPS 値を使用して、ADCA と ADCC に SOC を 2 つずつ構成します。これにより、ADCA と ADCC が同期動作するようになります。複数の ADC を搭載したデバイスでは、同じ原理を使用してすべての ADC を同期できます。

Example: Basic Synchronous Operation

```

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 will convert ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 will convert ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 will use sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 will begin conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4; //SOC1 will convert ADCINA4
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 will use sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC1CTL.bit.CHSEL = 1; //SOC1 will convert ADCINC1
AdccRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 will use sample duration of 31 SYSCLK cycles
AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 will begin conversion on ePWM3 SOCB

```

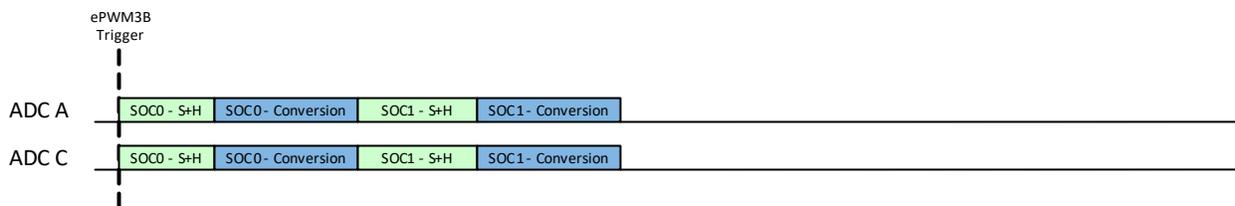


図 1-14. 例：基本的な同期動作

図 1-14 に、いくつかの注目点があります。まず、同じ番号の SOC では ACQPS 値を同じにする必要がありますが、異なる番号の SOC には異なる ACQPS 値を使用できます。このため、同期動作には単一のグローバル S+H 時間は必要なく、同時にサンプリングされるチャンネルでのみ S+H 時間を同じにする必要があります。この例のもう 1 つの重要な点は、どのチャンネル選択値をどの SOC にも使用できることです。最後に、この例ではラウンド・ロビン動作を想定しています。高優先度の SOC を使用する場合は、すべての ADC で優先度を同じに設定する必要があります。

1.13.1.2 複数のトリガ・ソースを使用した同期動作

SOC の各セットに同じトリガが選択され、同じ ACQPS 設定が使用されていれば、複数のトリガ・ソースを使用して同期動作を実現できます。

以下の例は、3 つの SOC と 2 つのトリガ・ソースを使用した ADCA と ADCC の同期動作を示しています。
図 1-15 は、相対トリガ・タイミングの任意の組み合わせで同期動作を達成できることを示しています。

Example: Synchronous Operation With Multiple Trigger Sources

```

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 will convert ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 will convert ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 will use sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 will begin conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4; //SOC1 will convert ADCINA4
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 will use sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC1CTL.bit.CHSEL = 1; //SOC1 will convert ADCINC1
AdccRegs.ADCSOC1CTL.bit.ACQPS = 30; //SOC1 will use sample duration of 31 SYSCLK cycles
AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10; //SOC1 will begin conversion on ePWM3 SOCB

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 0; //SOC2 will convert ADCINA0
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 19; //SOC2 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 2; //SOC2 will begin conversion on CPU Timer1
AdccRegs.ADCSOC2CTL.bit.CHSEL = 2; //SOC2 will convert ADCINC2
AdccRegs.ADCSOC2CTL.bit.ACQPS = 19; //SOC2 will use sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC2CTL.bit.TRIGSEL = 2; //SOC2 will begin conversion on CPU Timer1
    
```

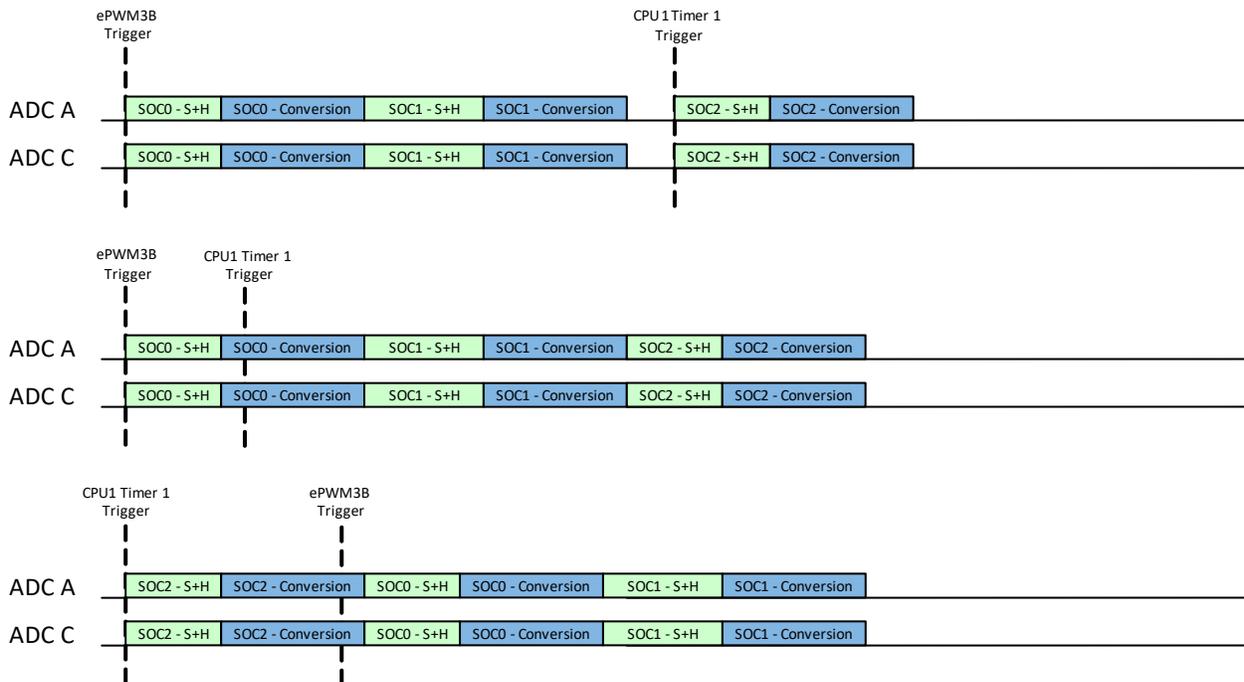


図 1-15. 例：複数のトリガ・ソースを使用した同期動作

ソフトウェア・トリガ以外であれば、TRIGSEL フィールドで選択可能なトリガ・ソースでも使用できます。すべての ADC に対して同時にソフトウェア・トリガを発行する方法はないため、非同期動作になる可能性が高くなります。すべての ADC に対して ADCINTSOCSEL1 と ADCINTSOCSEL2 レジスタの構成を同じにし、変換チェーンの開始にソフトウェア・トリガを使用していなければ、ADCINT1 または ADCINT2 もトリガとして使用できます。

1.13.1.3 SOC の数が等しくない場合の同期動作

1つのトリガ・ソースのみを使用する場合、1つのADCで使用されるSOCの数が他のADCよりも多い場合でも、同期動作を維持できます。

Example: Synchronous Operation With Uneven SOC Numbers

```

AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4;      //SOC0 will convert ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0;      //SOC0 will convert ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;     //SOC0 will use sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10;   //SOC0 will begin conversion on ePWM3 SOCB

AdcaRegs.ADCSOC1CTL.bit.CHSEL = 4;      //SOC1 will convert ADCINA4
AdcaRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 will use sample duration of 31 SYSCLK cycles
AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC1CTL.bit.CHSEL = 1;      //SOC1 will convert ADCINC1
AdccRegs.ADCSOC1CTL.bit.ACQPS = 30;     //SOC1 will use sample duration of 31 SYSCLK cycles
AdccRegs.ADCSOC1CTL.bit.TRIGSEL = 10;   //SOC1 will begin conversion on ePWM3 SOCB

AdcaRegs.ADCSOC2CTL.bit.CHSEL = 0;      //SOC2 will convert ADCINA0
AdcaRegs.ADCSOC2CTL.bit.ACQPS = 19;     //SOC2 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC2CTL.bit.TRIGSEL = 10;   //SOC2 will begin conversion on ePWM3 SOCB

```

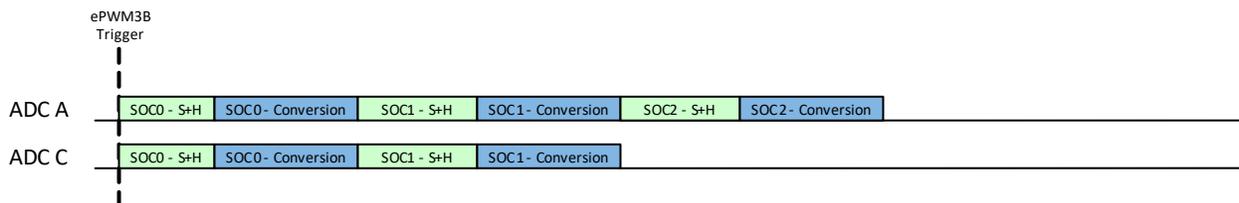


図 1-16. 例 : SOC の数が等しくない場合の同期動作

すべてのSOCが変換を完了する前にトリガが再び発生した場合は、ADCCはSOC0での変換を直ちに開始し、ADCAはSOC2が完了するまでSOC0での変換は開始しないことに注意してください。この場合は非同期動作になるため、トリガをオーバーフローさせないよう注意する必要があります。

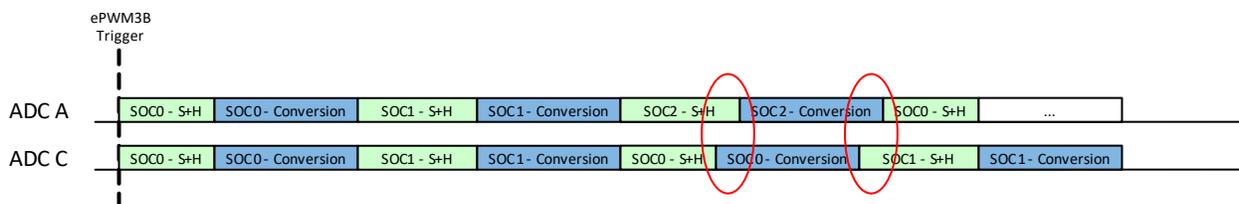


図 1-17. 例 : SOC の数が等しくない場合の非同期動作 – トリガ・オーバーフロー

1.13.1.4 重複しない変換

ユーザーが変換のタイミングが重複しないことを確実にしている場合は、同期動作と同等の性能を達成するために、すべての ADC ですべての SOC を同じ構成にする必要はありません。たとえば、システム内の 2 つの ADC トリガが常に 180 度位相差のある 2 つの ePWM ソースから供給される場合、SOC0 を ADCA と ADCC の両方で異なるトリガ・ソースおよび異なる ACQPS 値で使用できます。

Example: Operation with Non-overlapping Conversions

```
//ePWM3 SOCA and SOCB are 180 degrees out of phase
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 4; //SOC0 will convert ADCINA4
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10; //SOC0 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 0; //SOC0 will convert ADCINC0
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19; //SOC0 will use sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 9; //SOC0 will begin conversion on ePWM3 SOCA
```

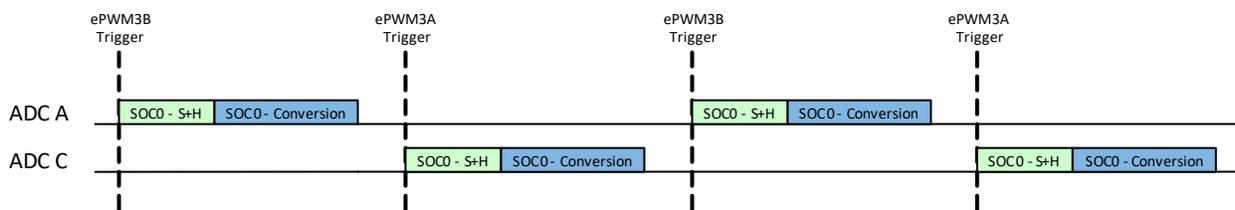


図 1-18. 例：重複しない変換による同期等価動作

1.13.2 アクイジション・ウィンドウ幅の選択

正しく動作させるには、ADC への入力信号がサンプル / ホールド・コンデンサ C_h を充電するのに十分な時間を確保する必要があります。通常、許容されるセトリング誤差によって、サンプリング・コンデンサが最終値の $\frac{1}{2}$ LSB または $\frac{1}{4}$ LSB 以内に充電されるように、S+H 期間を選択します。

必要なセトリング・タイムを決定するには、ADC および ADC 駆動回路をシミュレーションして、適切なセトリング性能が得られるようにするのが最適な方法です。ADC シグナル・コンディショニング回路の設計と評価の詳細については、『C2000 MCU 用の ADC 入力回路の評価』と『C2000 ADC 用の電荷共有駆動回路』を参照してください。

必要なセトリング時間の概算値は、RC セトリング・モデルを使用して決定することもできます。モデルの時定数を求めるには、次の式を使用します。

$$\tau = (R_S + R_{on}) \times C_h + R_S \times (C_S + C_P) \tag{1}$$

必要な時定数の数は、次の式で求められます。

$$k = \ln \left(\frac{2^n}{\text{settling error}} \right) - \ln \left(\frac{C_S + C_P}{C_H} \right) \tag{2}$$

合計 S+H 時間は、次以上に設定します。

$$t = k \cdot \tau \tag{3}$$

以下のパラメータは、デバイスのデータ・マニュアルの ADC 入力モデルから得られます。

- n = ADC の分解能 (ビット数)
- R_{ON} = ADC サンプリング・スイッチの抵抗 (Ω)
- C_H = ADC サンプリング・コンデンサ (pF)
- C_P = ADC チャネルの寄生入力容量 (pF)

以下のパラメータは、アプリケーションの設計によって異なります。

- セトリング誤差 = 許容されるセトリング誤差 (LSB 単位)
- R_S = ADC 駆動回路のソース・インピーダンス (通常は Ω または k Ω)
- C_S = ADC 入力ピンの容量 (通常は pF または nF)

たとえば、パラメータが次の値であるとします。

- n = 12 ビット
- R_{ON} = 500 Ω
- C_H = 12.5pF
- C_P = 12.7pF
- セトリング誤差 = $\frac{1}{4}$ LSB
- R_S = 180 Ω
- C_S = 150pF

時定数は次のように計算されます。

$$\tau = (180\Omega + 500\Omega) \times 12.5\text{pF} + 180\Omega \times (150\text{pF} + 12.7\text{pF}) = 37.8\text{ns} \quad (4)$$

必要な時定数の数は次のとおりです。

$$k = \ln\left(\frac{2^{12}}{0.25}\right) - \ln\left(\frac{150\text{pF} + 12.7\text{pF}}{12.5\text{pF}}\right) = 9.70 - 2.57 = 7.13 \quad (5)$$

合計 S+H 時間は、 $37.8\text{ns} \times 7.13 = 270\text{ns}$ 以上に設定します。

SYSCLOCK = 120MHz の場合、各 SYSCLOCK サイクルは 8.33ns です。S+H 期間は $270\text{ns}/8.33\text{ns} = 32.4$ SYSCLOCK サイクルとなるので、この入力の ACQPS は $\text{CEILING}(32.4) - 1 = 31$ 以上に設定します。

これにより必要なアキュジション・ウィンドウを概算できますが、より良い方法は、ADC 入力モデル、ソース・インピーダンス / 容量のモデル、ボード寄生成分を含む回路を SPICE (または類似のソフトウェア) で設定してシミュレーションし、サンプリング・コンデンサが必要な精度で安定することを確認することです。

注

デバイスのデータ・マニュアルには、ADC の最小 S+H ウィンドウ期間が規定されています。この仕様より期間が短くなる ACQPS 値を使用しないでください。

1.13.3 同時サンプリングの実現

各 ADC にはデュアル S+H 回路はありませんが、同時サンプリングは簡単に実現できます。これには、2 つ以上の ADC モジュールで同じトリガ・ソースを使用するように SOC トリガを設定します。次の例に、ePWM3 イベントに基づく 2 ADC の同時サンプリングを示します。ADCINA3、ADCINC5 がサンプリングされます。20 SYSCLK サイクルのアクイジション・ウィンドウが使用されますが、異なる期間が可能で

```
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 3;           //SOC0 will convert ADCINA3
AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;          //SOC0 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 10;        //SOC0 will begin conversion on ePWM3 SOCB
AdccRegs.ADCSOC0CTL.bit.CHSEL = 5;           //SOC0 will convert ADCINC5
AdccRegs.ADCSOC0CTL.bit.ACQPS = 19;          //SOC0 will use sample duration of 20 SYSCLK cycles
AdccRegs.ADCSOC0CTL.bit.TRIGSEL = 10;        //SOC0 will begin conversion on ePWM3 SOCB
```

ePWM3 トリガを受信すると、の ADC すべてがすぐに変換を開始します。すべての結果は、各 ADC の ADCRESULT0 レジスタに格納されます。ここでは、トリガを受信したときにすべての ADC がアイドル状態であると想定しています。1 つ以上の ADC がビジーの場合、サンプルはまったく同時には発生しません。

1.13.4 結果レジスタのマッピング

ADC の結果および ADC PPB の結果は、システム内の各メモリ・バス・コントローラに複製されます。すべてのバス・コントローラには、特定の部品ファミリおよび型番に存在する C28x CPU、C28x DMA、CLA があります。それぞれのバス・コントローラごとに、結果レジスタへの読み取りアクセスを許可するアクセス構成は必要ありません。また、複数のバス・コントローラが ADC 結果を同時に読もうとした場合に、競合は発生しません。

1.13.5 内部温度センサ

内部温度センサは、デバイスの接合部温度を測定します。センサの出力は、ADC を使用して内部接続を介してサンプリングできます。これは、TSNSCTL レジスタの ENABLE ビットを設定することにより、ADCC の ADCIN12 チャンネル、CMPSS2_HP5 入力でイネーブルにできます。

温度センサの読み取り値を温度に変換するには、温度センサの読み取り値を ADC driverlib の ADC_getTemperatureC() 関数に渡します。

1.13.6 外部リファレンス回路の設計

図 1-19 に、外部電圧リファレンス生成回路の基本的な構成を示します。リファレンス電圧は、帯域幅が良好で出力インピーダンスの低い高精度オペアンプでバッファしてから、リファレンス・ピンに駆動できます。高周波電流を吸収されやすくするため、高電位リファレンス・ピンと低電位リファレンス・ピン間のコンデンサは、PCB 上のピンにできるだけ近づけて配置します。オペアンプの安定性を確保するため、このコンデンサと直列に抵抗 (通常は 1Ω 未満) を接続することが必要な場合があります。

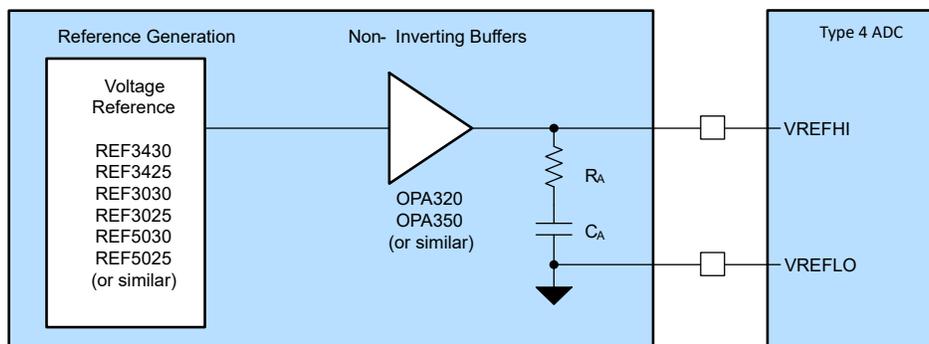


図 1-19. ADC リファレンス・システム

1.14 ソフトウェア

1.14.1 ADC の例

注：これらの例は、[C2000Ware](#) インストール・ディレクトリの次の場所にあります。
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/adc

これらのサンプルは、クラウドの [dev.ti.com C2000Ware サンプル](#) から入手できます。

1.14.1.1 ADC ソフトウェア・トリガ

ファイル：adc_ex1_soc_software.c

この例では、ソフトウェア・トリガに基づいて、ADCA と ADCC のいくつかの電圧を変換します。

ADCC は ADCA が完了するまで変換を実行しないため、ADC は非同期には実行されません。ただしこの場合、(ePWM トリガを使用するなどして) ADC を同期させて並列に変換するよりも、はるかに効率が低くなります。

外部接続

- A0、A1、C2、C3 ピンを変換する信号に接続する必要があります。

注目する変数

- *myADC0Result0* - ピン A0 の電圧のデジタル表現
- *myADC0Result1* - ピン A1 の電圧のデジタル表現
- *myADC1Result0* - ピン C2 の電圧のデジタル表現
- *myADC1Result1* - ピン C3 の電圧のデジタル表現

1.14.1.2 ADC ePWM のトリガ

ファイル：adc_ex2_soc_epwm.c

この例では、ADCA で変換を定期的にトリガするように ePWM1 を設定します。

外部接続

- A0 を変換する信号に接続する必要があります。

注目する変数

- *myADC0Results* - ピン A0 からの A/D 変換サンプルのシーケンス。サンプル間の時間は、ePWM タイマの期間に基づいて決定されます。

1.14.1.3 ADC 温度センサ変換

ファイル：adc_ex3_temp_sensor.c

この例では、ADC を定期的にトリガするように ePWM を設定します。ADC は内部接続を温度センサに変換します。これを、ADC_getTemperatureC() 関数を呼び出して温度として解釈します。

注目する変数

- *sensorSample* - 温度センサからの未処理の読み取り値
- *sensorTemp* - センサ・サンプルを温度 (°C) として解釈した値。

1.14.1.4 ADC 同期 SOC ソフトウェア強制 (adc_soc_software_sync)

ファイル：adc_ex4_soc_software_sync.c

この例では、入力クロスバーの入力 5 をソフトウェア強制として使用し、ADCA と ADCC のいくつかの電圧を変換します。入力 5 は GPIO0 をトグルすることによりトリガされますが、どの予備 GPIO でも使用できます。この方法により、両方の ADC がまったく同時に変換を開始するようになります。

外部接続

- A2、A3、C2、C3 ピンを変換する信号に接続する必要があります。

注目する変数

- *myADC0Result0* : ピン A2 の電圧のデジタル表現
- *myADC0Result1* : ピン A3 の電圧のデジタル表現

- `myADC1Result0` : ピン C2 の電圧のデジタル表現
- `myADC1Result1` - ピン C3 の電圧のデジタル表現

1.14.1.5 ADC 連続トリガ (`adc_soc_continuous`)

ファイル : `adc_ex5_soc_continuous.c`

この例では、ADC を設定して連続変換を実行し、最大サンプリング・レートを実現します。

外部接続

- A0 ピンを変換する信号に接続する必要があります。

注目する変数

- `adcAResults` - ピン A0 からの A/D 変換サンプルのシーケンス。サンプリングの間隔は、ADC の速度に基づく最小値です。

1.14.1.6 ADC PPB オフセット (`adc_ppb_offset`)

ファイル : `adc_ex7_ppb_offset.c`

この例では、ソフトウェアで ADC をトリガします。一部の SOC には、後処理ブロックによって自動オフセット調整が適用されます。プログラムの実行後、メモリには ADC と後処理ブロック (PPB) の結果が格納されます。

外部接続

- A2、C2 ピンを変換する信号に接続できます。

注目する変数

- `myADC0Result` : ピン A2 の電圧のデジタル表現
- `myADC0PPBResult` : ピン A2 の電圧をデジタル表現から、自動的に追加されたオフセット 100LSB を減算した結果
- `myADC1Result` : ピン C2 の電圧のデジタル表現
- `myADC1PPBResult` : ピン C2 の電圧のデジタル表現に、自動的に追加されたオフセット 100LSB を加算した結果

1.14.1.7 ADC PPB の制限値 (`adc_ppb_limits`)

ファイル : `adc_ex8_ppb_limits.c`

この例では、ADC を定期的なトリガするように ePWM を設定します。結果が定義された範囲外である場合、後処理ブロックが割り込みを生成します。

デフォルトの制限値は 1000LSB と 3000LSB です。VREFHI を 3.3V に設定すると、入力電圧が約 2.4V を上回った場合、または約 0.8V を下回った場合に、PPB が割り込みを生成します。

外部接続

- A0 を変換する信号に接続できます。

注目する変数

- なし

1.14.1.8 ADC PPB 遅延キャプチャ (`adc_ppb_delay`)

ファイル : `adc_ex9_ppb_delay.c`

この例では、後処理ブロックを使用した遅延キャプチャを示します。

次の 2 つの非同期 ADC トリガが設定されています。

- ePWM1、期間 2048、SOC0 をトリガしてピン A0 に変換
 - ePWM2、期間 9999、SOC1 をトリガしてピン A2 を変換
- 各変換の終了時に ISR が生成されます。SOC0 の ISR では、変換カウンタがインクリメントされ、PPB がチェックされて、サンプルが遅延されているかどうか判定されます。プログラムの実行後、メモリには次の内容が格納されます。

`conversion` : SOC0 を使用した遅延された変換シーケンス

- *delay* : 遅延された各変換に対応する遅延

1.14.1.9 ADC ePWM による複数の SOC のトリガ

ファイル : `adc_ex10_multiple_soc_epwm.c`

この例では、ADCA と ADCC で一連の変換を定期的なトリガするように ePWM1 を設定します。複数の ADC が連携して動作し、複数の ADC で使用可能な並列処理を使用して、1 つの変換バッチを処理する方法を示します。

ADCA 割り込み ISR を使用して、ADCA と ADCC の両方の結果を読み取ります。

外部接続

- A0、A1、A2、C2、C3、C4 ピンは、変換する信号に接続する必要があります。

注目する変数

- *adcAResult0* - ピン A0 の電圧のデジタル表現
- *adcAResult1* - ピン A1 の電圧のデジタル表現
- *adcAResult2* - ピン A2 の電圧のデジタル表現
- *adcCResult0* - ピン C2 の電圧のデジタル表現
- *adcCResult1* - ピン C3 の電圧のデジタル表現
- *adcCResult2* - ピン C4 の電圧のデジタル表現

1.14.1.10 ADC バースト・モード

ファイル : `adc_ex11_burst_mode_epwm.c`

この例では、バースト・モードを使用して ADCA を定期的なトリガするように ePWM1 を設定します。これにより、各バーストで異なるチャンネルをサンプリングできます。

各バーストは 3 回の変換をトリガします。A0 と A1 は各バーストの一部であり、3 番目の変換で A2、A3、A4 が循環します。これにより、重要度の高い信号を高速でサンプリングでき、優先度の低い信号は低いレートでサンプリングできます。

ADCA 割り込み ISR を使用して、ADCA の結果を読み取ります。

外部接続

- A0、A1、A2、A3、A4

注目する変数

- *adcAResult0* - ピン A0 の電圧のデジタル表現
- *adcAResult1* - ピン A1 の電圧のデジタル表現
- *adcAResult2* - ピン A2 の電圧のデジタル表現
- *adcAResult3* - ピン A3 の電圧のデジタル表現
- *adcAResult4* - ピン A4 の電圧のデジタル表現

1.14.1.11 ADC バースト・モードのオーバーサンプリング

ファイル : `adc_ex12_burst_mode_oversampling.c`

この例では、A0 と A1 をサンプリングするために、ADCA 上の SOC0 と SOC1 を定期的なトリガするように ePWM1 を設定します。また、ePWM1 を使用して ADC バースト・モードもトリガします。バースト SOC は、複数の ePWM 期間にわたって A2 をオーバーサンプリングするために、複数の変換を蓄積するのに使用されます。

外部接続

- A0、A1、A2

注目する変数

- *adcAResult0* - ピン A0 の電圧のデジタル表現
- *adcAResult1* - ピン A1 の電圧のデジタル表現
- *adcAResult2* - ピン A2 の電圧のデジタル表現

1.14.1.12 ADC SOC オーバーサンプリング

ファイル : `adc_ex13_soc_oversampling.c`

この例では、複数の SOC を含む ADCA で定期的に一連の変換をトリガするように ePWM1 を設定します。これらの SOC はすべて A2 を変換し、A2 のオーバーサンプリングを実現します。

ADCA 割り込み ISR を使用して、ADCA の結果を読み取ります。

外部接続

- A0、A1、A2 ピンは、変換する信号に接続する必要があります。

注目する変数

- `adcAResult0` - ピン A0 の電圧のデジタル表現
- `adcAResult1` - ピン A1 の電圧のデジタル表現
- `adcAResult2` - ピン A2 の電圧のデジタル表現

1.14.1.13 ADC PPB PWM トリップ (`adc_ppb_pwm_trip`)

ファイル : `adc_ex14_ppb_pwm_trip.c`

この例では、ADC 制限値検出 PPB ブロックによる ePWM トリップを示します。ADCAINT1 は、ソフトウェアによる初期強制トリガ後に、ADCA チャンネル 2 を定期的トリガするよう構成されています。制限値検出の後処理ブロック (PPB) が構成されており、ADC の結果が定義された範囲外である場合、後処理ブロックが ADCxEVTy イベントを生成します。このイベントは、ePWM クロスバー、対応する ePWM のトリップ・ゾーン、およびデジタル比較サブモジュールを構成することにより、ePWM トリップ・ソースとして構成されません。この例は次を示しています。

- ワンショット
- サイクル単位
- デジタル比較サブモジュールを使用して、ADCAEVT1 ソースで PWM を直接トリップ

デフォルトの制限値は 0LSB と 3600LSB です。VREFHI を 3.3V に設定すると、入力電圧が約 2.9V を上回った場合に PPB がトリップ・イベントを生成します。

外部接続

- A2 を変換する信号に接続できます。
- オシロスコープで以下の信号を観測します。
 - ePWM1(GPIO0 ~ GPIO1)
 - ePWM2(GPIO2 ~ GPIO3)
 - ePWM3(GPIO4 ~ GPIO5)

注目する変数

- `adcA2Results` - ピン A2 の電圧のデジタル表現

1.14.1.14 ADC の開放 / 短絡検出 (`adc_open_shorts_detection`)

ファイル : `adc_ex15_open_shorts_detection.c`

この例では、システム内のピン・フォルトを検出するための ADC の開放 / 短絡検出 (ADCOSDETECT) 回路構成を示します。この例では、必須の ADC 構成とともに開放 / 短絡検出回路を有効にし、通常の ADC 変換を開始する前に ADCA A0 入力ピンの状態を診断します。

ADC の OSDetect 回路をイネーブルにするには、次の手順に従います。

1. ADC を変換用に構成します (例 : チャンネル、SOC、ACQPS、プリスケラ、トリガ)。OSDetect 機能は 12 ビットでのみ使用できます。
2. 目的の分圧器接続に合わせて、ADCOSDETECT レジスタを設定します。使用可能な OSDetect 構成の詳細については、「開放 / 短絡検出回路 (OSDETECT)」セクションを参照してください。
3. 変換を開始し、変換結果を調べます。注 : 結果は、入力側を駆動しているものと R_s および C_p の値に基づいて解釈する必要があります。Vs 信号を入力ピンから切断できる場合、この回路を使用して入力ピンの開放と短絡を検出できます。この例では、ADCA A0 チャンネルが構成されており、次のアルゴリズムを使用して A0 ピンのステータスを確認します。

- a. ステップ 1 : フルスケール OSDETECT モードを構成し、ADC の結果 (resultHi) をキャプチャします。
- b. ステップ 2 : ゼロスケール OSDETECT モードを構成し、ADC の結果 (resultLo) をキャプチャします。
- c. ステップ 3 : OSDETECT モードをディセーブルにし、ADC の結果 (resultNormal) をキャプチャします。
- d. ステップ 4 : ADC ピンの状態を判定します。
 - i. a. ピンが開放されている場合、resultLo は Vreflo と等しくなり、resultHi は Vrefhi と等しくなります。
 - ii. b. ピンが Vrefhi に短絡している場合、resultLo は Vrefhi にほぼ等しくなり、resultHi は Vrefhi に等しくなります。
 - iii. c. ピンが Vreflo に短絡している場合、resultLo は Vreflo と等しくなり、resultHi は Vreflo とほぼ等しくなります。
 - iv. d. ピンが有効な信号に接続されている場合、resultLo は osdLoLimit より大きく、resultNormal より小さくなり、resultHi は osdHiLimit より小さく、resultNormal より大きくなります。
- e. ステップ 5 : osDetectStatusVal > 4 の場合、ピン・フォルトが発生していないことを意味します。
 - i. a. osDetectStatusVal == 1 の場合、ピン A0 は開放されています。
 - ii. b. osDetectStatusVal == 2 の場合、ピン A0 は VREFLO に短絡しています。
 - iii. c. osDetectStatusVal == 4 の場合、ピン A0 は VREFHI に短絡しています。
 - iv. d. osDetectStatusVal == 8 の場合、ピン A0 は GOOD/VALID 状態です。
 - v. e. osDetectStatusVal > 4 の場合、ピン A0 は VALID 状態です。

ADC を OSDETECT モードで構成する際は、以下の点に注意してください。

1. 分圧抵抗の許容誤差は大きく異なる可能性があるため、この機能を使用して変換の精度を確認することはできません。
2. アナログ入力チャンネルの実装と利用可能性については、デバイスのデータシートを参照してください。
3. 駆動インピーダンスが高いため、S+H 期間を ADC の最小値よりもかなり長くする必要があります。

外部接続

- A0 ピンを変換する信号に接続できます。

注目する変数

- `osDetectStatusVal` : ピン A0 の電圧の OS 検出ステータス
- `adcAResult0` : ピン A0 の電圧のデジタル表現

1.15 ADC レジスタ

このセクションでは、A/D コンバータのレジスタについて説明します。

1.15.1 ADC ベース・アドレス表

表 1-10. ADC ベース・アドレス表

ビット・フィールド名		DriverLib 名	ベース・アドレス	パイプライン保護
インスタンス	構造			
AdcaResultRegs	ADC_RESULT_REGS	ADCARESULT_BASE	0x0000_0B00	-
AdccResultRegs	ADC_RESULT_REGS	ADCCRESULT_BASE	0x0000_0B40	-
AdcaRegs	ADC_REGS	ADCA_BASE	0x0000_7400	あり
AdccRegs	ADC_REGS	ADCC_BASE	0x0000_7500	あり

1.15.2 ADC_RESULT_REGS レジスタ

表 1-11 に、ADC_RESULT_REGS レジスタのメモリマップされたレジスタを示します。表 1-11 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-11. ADC_RESULT_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	ADCRESULT0	ADC 結果 0 レジスタ		表示
1h	ADCRESULT1	ADC 結果 1 レジスタ		表示
2h	ADCRESULT2	ADC 結果 2 レジスタ		表示
3h	ADCRESULT3	ADC 結果 3 レジスタ		表示
4h	ADCRESULT4	ADC 結果 4 レジスタ		表示
5h	ADCRESULT5	ADC 結果 5 レジスタ		表示
6h	ADCRESULT6	ADC 結果 6 レジスタ		表示
7h	ADCRESULT7	ADC 結果 7 レジスタ		表示
8h	ADCRESULT8	ADC 結果 8 レジスタ		表示
9h	ADCRESULT9	ADC 結果 9 レジスタ		表示
Ah	ADCRESULT10	ADC 結果 10 レジスタ		表示
Bh	ADCRESULT11	ADC 結果 11 レジスタ		表示
Ch	ADCRESULT12	ADC 結果 12 レジスタ		表示
Dh	ADCRESULT13	ADC 結果 13 レジスタ		表示
Eh	ADCRESULT14	ADC 結果 14 レジスタ		表示
Fh	ADCRESULT15	ADC 結果 15 レジスタ		表示
10h	ADCPPB1RESULT	ADC 後処理ブロック 1 の結果レジスタ		表示
12h	ADCPPB2RESULT	ADC 後処理ブロック 2 の結果レジスタ		表示
14h	ADCPPB3RESULT	ADC 後処理ブロック 3 の結果レジスタ		表示
16h	ADCPPB4RESULT	ADC 後処理ブロック 4 の結果レジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-12 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-12. ADC_RESULT_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ・アレイ変数		
i、j、k、l、m、n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、反復レジスタ・グループの一部であるレジスタのレジスタ・アレイの値を示します。レジスタ・グループは階層構造を形成し、アレイは式で表されません。

表 1-12. ADC_RESULT_REGS のアクセス・タイプ・コード (continued)

アクセス・タイプ	コード	説明
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ・アレイの値を示します。

1.15.2.1 ADCRESULT0 レジスタ (オフセット = 0h) [リセット = 0h]

図 1-20 に ADCRESULT0 を示し、表 1-13 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 0 レジスタ

図 1-20. ADCRESULT0 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-13. ADCRESULT0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 0 16 ビット ADC の結果。ADC が SOC0 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.2 ADCRESULT1 レジスタ (オフセット = 1h) [リセット = 0h]

図 1-21 に ADCRESULT1 を示し、表 1-14 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 1 レジスタ

図 1-21. ADCRESULT1 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-14. ADCRESULT1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 1 16 ビット ADC の結果。ADC が SOC1 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.3 ADCRESULT2 レジスタ (オフセット = 2h) [リセット = 0h]

図 1-22 に ADCRESULT2 を示し、表 1-15 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 2 レジスタ

図 1-22. ADCRESULT2 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-15. ADCRESULT2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 2 16 ビット ADC の結果。ADC が SOC2 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.4 ADCRESULT3 レジスタ (オフセット = 3h) [リセット = 0h]

図 1-23 に ADCRESULT3 を示し、表 1-16 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 3 レジスタ

図 1-23. ADCRESULT3 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-16. ADCRESULT3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 3 16 ビット ADC の結果。ADC が SOC3 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.5 ADCRESULT4 レジスタ (オフセット = 4h) [リセット = 0h]

図 1-24 に ADCRESULT4 を示し、表 1-17 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 4 レジスタ

図 1-24. ADCRESULT4 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-17. ADCRESULT4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 4 16 ビット ADC の結果。ADC が SOC4 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.6 ADCRESULT5 レジスタ (オフセット = 5h) [リセット = 0h]

図 1-25 に ADCRESULT5 を示し、表 1-18 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 5 レジスタ

図 1-25. ADCRESULT5 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-18. ADCRESULT5 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 5 16 ビット ADC の結果。ADC が SOC5 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.7 ADCRESULT6 レジスタ (オフセット = 6h) [リセット = 0h]

図 1-26 に ADCRESULT6 を示し、表 1-19 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 6 レジスタ

図 1-26. ADCRESULT6 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-19. ADCRESULT6 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 6 16 ビット ADC の結果。ADC が SOC6 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.8 ADCRESULT7 レジスタ (オフセット = 7h) [リセット = 0h]

図 1-27 に ADCRESULT7 を示し、表 1-20 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 7 レジスタ

図 1-27. ADCRESULT7 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-20. ADCRESULT7 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 7 16 ビット ADC の結果。ADC が SOC7 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.9 ADCRESULT8 レジスタ (オフセット = 8h) [リセット = 0h]

図 1-28 に ADCRESULT8 を示し、表 1-21 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 8 レジスタ

図 1-28. ADCRESULT8 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-21. ADCRESULT8 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 8 16 ビット ADC の結果。ADC が SOC8 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.10 ADCRESULT9 レジスタ (オフセット = 9h) [リセット = 0h]

図 1-29 に ADCRESULT9 を示し、表 1-22 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 9 レジスタ

図 1-29. ADCRESULT9 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-22. ADCRESULT9 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 9 16 ビット ADC の結果。ADC が SOC9 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.11 ADCRESULT10 レジスタ (オフセット= Ah) [リセット= 0h]

図 1-30 に ADCRESULT10 を示し、表 1-23 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 10 レジスタ

図 1-30. ADCRESULT10 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-23. ADCRESULT10 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 10 16 ビット ADC の結果。ADC が SOC10 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.12 ADCRESULT11 レジスタ (オフセット = Bh) [リセット = 0h]

図 1-31 に ADCRESULT11 を示し、表 1-24 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 11 レジスタ

図 1-31. ADCRESULT11 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-24. ADCRESULT11 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 11 16 ビット ADC の結果。ADC が SOC11 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.13 ADCRESULT12 レジスタ (オフセット = Ch) [リセット = 0h]

図 1-32 に ADCRESULT12 を示し、表 1-25 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 12 レジスタ

図 1-32. ADCRESULT12 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-25. ADCRESULT12 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 12 16 ビット ADC の結果。ADC が SOC12 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.14 ADCRESULT13 レジスタ (オフセット = Dh) [リセット = 0h]

図 1-33 に ADCRESULT13 を示し、表 1-26 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 13 レジスタ

図 1-33. ADCRESULT13 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-26. ADCRESULT13 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 13 16 ビット ADC の結果。ADC が SOC13 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.15 ADCRESULT14 レジスタ (オフセット = Eh) [リセット = 0h]

図 1-34 に ADCRESULT14 を示し、表 1-27 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 14 レジスタ

図 1-34. ADCRESULT14 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-27. ADCRESULT14 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 14 16 ビット ADC の結果。ADC が SOC14 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.16 ADCRESULT15 レジスタ (オフセット = Fh) [リセット = 0h]

図 1-35 に ADCRESULT15 を示し、表 1-28 にその説明を示します。

[概略表](#)に戻ります。

ADC 結果 15 レジスタ

図 1-35. ADCRESULT15 レジスタ

15	14	13	12	11	10	9	8
結果							
R-0h							
7	6	5	4	3	2	1	0
結果							
R-0h							

表 1-28. ADCRESULT15 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	結果	R	0h	ADC の結果 15 16 ビット ADC の結果。ADC が SOC15 の変換を完了すると、デジタル結果がこのビット・フィールドに配置されます。 リセット・タイプ : SYSRSn

1.15.2.17 ADCPPB1RESULT レジスタ (オフセット = 10h) [リセット = 0h]

図 1-36 に ADCPPB1RESULT を示し、表 1-29 にその説明を示します。

概略表に戻ります。

ADC 後処理ブロック 1 の結果レジスタ

図 1-36. ADCPPB1RESULT レジスタ

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

表 1-29. ADCPPB1RESULT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	SIGN	R	0h	符号拡張ビット。これらのビットは、ビット 16 の値を反映します。 注：この後処理ブロックに関連付けられている変換が 12 ビットの変換である場合、SIGN ビットはビット 12 まで拡張され、すべてのビットがビット 12 と同じ値になります。 リセット・タイプ：SYSRSn
15-0	PPBRESULT	R	0h	ADC 後処理ブロックの結果 1 変換処理の結果は、オフセット / リファレンスの減算後にこのレジスタに保存されます。 この結果は、関連する ADCRESULT が使用可能になった 1 SYSCLK サイクル後に使用できます。ADCINTFLG をポーリングして PPBRESULT を読み取るタイミングを決定する場合、変換後の更新された処理結果がレジスタに配置されていることを確実にするため、NOP 命令を追加する必要がある場合があります。 注：この後処理ブロックに関連付けられた変換が 12 ビットの変換である場合、PPBRESULT ビットはビット 12:0 に制限されます。 リセット・タイプ：SYSRSn

1.15.2.18 ADCPPB2RESULT レジスタ (オフセット = 12h) [リセット = 0h]

図 1-37 に ADCPPB2RESULT を示し、表 1-30 にその説明を示します。

[概略表](#)に戻ります。

ADC 後処理ブロック 2 の結果レジスタ

図 1-37. ADCPPB2RESULT レジスタ

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

表 1-30. ADCPPB2RESULT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	SIGN	R	0h	符号拡張ビット。これらのビットは、ビット 16 の値を反映します。 注：この後処理ブロックに関連付けられている変換が 12 ビットの変換である場合、SIGN ビットはビット 12 まで拡張され、すべてのビットがビット 12 と同じ値になります。 リセット・タイプ：SYSRSn
15-0	PPBRESULT	R	0h	ADC 後処理ブロックの結果 2 変換処理の結果は、オフセット / リファレンスの減算後にこのレジスタに保存されます。 この結果は、関連する ADCRESULT が使用可能になった 1 SYSCLK サイクル後に使用できます。ADCINTFLG をポーリングして PPBRESULT を読み取るタイミングを決定する場合、変換後の更新された処理結果がレジスタに配置されていることを確実にするため、NOP 命令を追加する必要がある場合があります。 注：この後処理ブロックに関連付けられた変換が 12 ビットの変換である場合、PPBRESULT ビットはビット 12:0 に制限されます。 リセット・タイプ：SYSRSn

1.15.2.19 ADCPPB3RESULT レジスタ (オフセット = 14h) [リセット = 0h]

図 1-38 に ADCPPB3RESULT を示し、表 1-31 にその説明を示します。

概略表に戻ります。

ADC 後処理ブロック 3 の結果レジスタ

図 1-38. ADCPPB3RESULT レジスタ

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

表 1-31. ADCPPB3RESULT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	SIGN	R	0h	符号拡張ビット。これらのビットは、ビット 16 の値を反映します。 注：この後処理ブロックに関連付けられている変換が 12 ビットの変換である場合、SIGN ビットはビット 12 まで拡張され、すべてのビットがビット 12 と同じ値になります。 リセット・タイプ：SYSRSn
15-0	PPBRESULT	R	0h	ADC 後処理ブロックの結果 3 変換処理の結果は、オフセット / リファレンスの減算後にこのレジスタに保存されます。 この結果は、関連する ADCRESULT が使用可能になった 1 SYSCLK サイクル後に使用できます。ADCINTFLG をポーリングして PPBRESULT を読み取るタイミングを決定する場合、変換後の更新された処理結果がレジスタに配置されていることを確実にするため、NOP 命令を追加する必要がある場合があります。 注：この後処理ブロックに関連付けられた変換が 12 ビットの変換である場合、PPBRESULT ビットはビット 12:0 に制限されます。 リセット・タイプ：SYSRSn

1.15.2.20 ADCPPB4RESULT レジスタ (オフセット = 16h) [リセット = 0h]

図 1-39 に ADCPPB4RESULT を示し、表 1-32 にその説明を示します。

[概略表](#)に戻ります。

ADC 後処理ブロック 4 の結果レジスタ

図 1-39. ADCPPB4RESULT レジスタ

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

表 1-32. ADCPPB4RESULT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	SIGN	R	0h	符号拡張ビット。これらのビットは、ビット 16 の値を反映します。 注：この後処理ブロックに関連付けられている変換が 12 ビットの変換である場合、SIGN ビットはビット 12 まで拡張され、すべてのビットがビット 12 と同じ値になります。 リセット・タイプ：SYSRSn
15-0	PPBRESULT	R	0h	ADC 後処理ブロックの結果 4 変換処理の結果は、オフセット / リファレンスの減算後にこのレジスタに保存されます。 この結果は、関連する ADCRESULT が使用可能になった 1 SYSCLK サイクル後に使用できます。ADCINTFLG をポーリングして PPBRESULT を読み取るタイミングを決定する場合、変換後の更新された処理結果がレジスタに配置されていることを確実にするため、NOP 命令を追加する必要がある場合があります。 注：この後処理ブロックに関連付けられた変換が 12 ビットの変換である場合、PPBRESULT ビットはビット 12:0 に制限されます。 リセット・タイプ：SYSRSn

1.15.3 ADC_REGS レジスタ

表 1-33 に、ADC_REGS レジスタのメモリマップされたレジスタを示します。表 1-33 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-33. ADC_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	ADCCTL1	ADC 制御 1 レジスタ	EALLOW	表示
1h	ADCCTL2	ADC 制御 2 レジスタ	EALLOW	表示
2h	ADCBURSTCTL	ADC バースト制御レジスタ	EALLOW	表示
3h	ADCINTFLG	ADC 割り込みフラグ・レジスタ		表示
4h	ADCINTFLGCLR	ADC 割り込みフラグ・クリア・レジスタ		表示
5h	ADCINTOVF	ADC 割り込みオーバーフロー・レジスタ		表示
6h	ADCINTOVFCLR	ADC 割り込みオーバーフロー・クリア・レジスタ		表示
7h	ADCINTSEL1N2	ADC 割り込み 1 および 2 選択レジスタ	EALLOW	表示
8h	ADCINTSEL3N4	ADC 割り込み 3 および 4 選択レジスタ	EALLOW	表示
9h	ADCSOCPRICL	ADC SOC 優先度制御レジスタ	EALLOW	表示
Ah	ADCINTSOCSEL1	ADC 割り込み SOC 選択 1 レジスタ	EALLOW	表示
Bh	ADCINTSOCSEL2	ADC 割り込み SOC 選択 2 レジスタ	EALLOW	表示
Ch	ADCSOCFLG1	ADC SOC フラグ 1 レジスタ		表示
Dh	ADCSOCFRC1	ADC SOC 強制 1 レジスタ		表示
Eh	ADCSOCOVF1	ADC SOC オーバーフロー 1 レジスタ		表示
Fh	ADCSOCOVFCLR1	ADC SOC オーバーフロー・クリア 1 レジスタ		表示
10h	ADCSOC0CTL	ADC SOC0 制御レジスタ	EALLOW	表示
12h	ADCSOC1CTL	ADC SOC1 制御レジスタ	EALLOW	表示
14h	ADCSOC2CTL	ADC SOC2 制御レジスタ	EALLOW	表示
16h	ADCSOC3CTL	ADC SOC3 制御レジスタ	EALLOW	表示
18h	ADCSOC4CTL	ADC SOC4 制御レジスタ	EALLOW	表示
1Ah	ADCSOC5CTL	ADC SOC5 制御レジスタ	EALLOW	表示
1Ch	ADCSOC6CTL	ADC SOC6 制御レジスタ	EALLOW	表示
1Eh	ADCSOC7CTL	ADC SOC7 制御レジスタ	EALLOW	表示
20h	ADCSOC8CTL	ADC SOC8 制御レジスタ	EALLOW	表示
22h	ADCSOC9CTL	ADC SOC9 制御レジスタ	EALLOW	表示
24h	ADCSOC10CTL	ADC SOC10 制御レジスタ	EALLOW	表示
26h	ADCSOC11CTL	ADC SOC11 制御レジスタ	EALLOW	表示
28h	ADCSOC12CTL	ADC SOC12 制御レジスタ	EALLOW	表示
2Ah	ADCSOC13CTL	ADC SOC13 制御レジスタ	EALLOW	表示
2Ch	ADCSOC14CTL	ADC SOC14 制御レジスタ	EALLOW	表示
2Eh	ADCSOC15CTL	ADC SOC15 制御レジスタ	EALLOW	表示
30h	ADCEVTSTAT	ADC イベント・ステータス・レジスタ		表示
32h	ADCEVTCLR	ADC イベント・クリア・レジスタ		表示
34h	ADCEVTSEL	ADC イベント選択レジスタ	EALLOW	表示
36h	ADCEVTINTSEL	ADC イベント割り込み選択レジスタ	EALLOW	表示
38h	ADCOSDETECT	ADC 開放 / 短絡検出レジスタ	EALLOW	表示
39h	ADCCOUNTER	ADC カウンタ・レジスタ		表示
3Ah	ADCREV	ADC リビジョン・レジスタ		表示
3Bh	ADCOFFTRIM	ADC オフセット・トリム・レジスタ	EALLOW	表示

表 1-33. ADC_REGS レジスタ (continued)

オフセット	略称	レジスタ名	書き込み保護	セクション
40h	ADCPPB1CONFIG	ADC PPB1 構成レジスタ	EALLOW	表示
41h	ADCPPB1STAMP	ADC PPB1 サンプル遅延タイムスタンプ・レジスタ		表示
42h	ADCPPB1OFFCAL	ADC PPB1 オフセット較正レジスタ	EALLOW	表示
43h	ADCPPB1OFFREF	ADC PPB1 オフセット・リファレンス・レジスタ		表示
44h	ADCPPB1TRIPHI	ADC PPB1 トリップ上限レジスタ	EALLOW	表示
46h	ADCPPB1TRIPLO	ADC PPB1 トリップ下限 / トリガ・タイムスタンプ・レジスタ	EALLOW	表示
48h	ADCPPB2CONFIG	ADC PPB2 構成レジスタ	EALLOW	表示
49h	ADCPPB2STAMP	ADC PPB2 サンプル遅延タイムスタンプ・レジスタ		表示
4Ah	ADCPPB2OFFCAL	ADC PPB2 オフセット較正レジスタ	EALLOW	表示
4Bh	ADCPPB2OFFREF	ADC PPB2 オフセット・リファレンス・レジスタ		表示
4Ch	ADCPPB2TRIPHI	ADC PPB2 トリップ上限レジスタ	EALLOW	表示
4Eh	ADCPPB2TRIPLO	ADC PPB2 トリップ下限 / トリガ・タイムスタンプ・レジスタ	EALLOW	表示
50h	ADCPPB3CONFIG	ADC PPB3 構成レジスタ	EALLOW	表示
51h	ADCPPB3STAMP	ADC PPB3 サンプル遅延タイムスタンプ・レジスタ		表示
52h	ADCPPB3OFFCAL	ADC PPB3 オフセット較正レジスタ	EALLOW	表示
53h	ADCPPB3OFFREF	ADC PPB3 オフセット・リファレンス・レジスタ		表示
54h	ADCPPB3TRIPHI	ADC PPB3 トリップ上限レジスタ	EALLOW	表示
56h	ADCPPB3TRIPLO	ADC PPB3 トリップ下限 / トリガ・タイムスタンプ・レジスタ	EALLOW	表示
58h	ADCPPB4CONFIG	ADC PPB4 構成レジスタ	EALLOW	表示
59h	ADCPPB4STAMP	ADC PPB4 サンプル遅延タイムスタンプ・レジスタ		表示
5Ah	ADCPPB4OFFCAL	ADC PPB4 オフセット較正レジスタ	EALLOW	表示
5Bh	ADCPPB4OFFREF	ADC PPB4 オフセット・リファレンス・レジスタ		表示
5Ch	ADCPPB4TRIPHI	ADC PPB4 トリップ上限レジスタ	EALLOW	表示
5Eh	ADCPPB4TRIPLO	ADC PPB4 トリップ下限 / トリガ・タイムスタンプ・レジスタ	EALLOW	表示
6Fh	ADCINTCYCLE	ADC Early 割り込み生成サイクル	EALLOW	表示
72h	ADCINLTRIM2	ADC 直線性トリム 2 レジスタ	EALLOW	表示
74h	ADCINLTRIM3	ADC 直線性トリム 3 レジスタ	EALLOW	表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-34 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-34. ADC_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み取り 0 を返す

**表 1-34. ADC_REGS のアクセス・タイプ・コード
(continued)**

アクセス・タイプ	コード	説明
書き込みタイプ		
W	W	書き込み
W1C	W 1C	1 を書き込むことで クリア
W1S	W 1S	1 を書き込むことで セット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト 値
レジスタ・アレイ変数		
i、j、k、l、m、 n		これらの変数がレジスタ名、オフ セット、またはアドレスで使用さ れている場合、反復レジスタ・グ ループの一部であるレジスタのレ ジスタ・アレイの値を示します。 レジスタ・グループは階層構造 を形成し、アレイは式で表されま す。
y		この変数がレジスタ名、オフセッ ト、またはアドレスで使用されて いる場合、レジスタ・アレイの値 を示します。

1.15.3.1 ADCCTL1 (オフセット = 0h) [リセット = 0h]

図 1-40 に ADCCTL1 を示し、表 1-35 にその説明を示します。

概略表に戻ります。

ADC 制御 1 レジスタ

図 1-40. ADCCTL1 レジスタ

15	14	13	12	11	10	9	8
RESERVED		ADCBSY	RESERVED	ADCBSYCHN			
R-0h		R-0h	R-0h	R-0h			
7	6	5	4	3	2	1	0
ADCPWDNZ	RESERVED				INTPULSEPOS	RESERVED	
R/W-0h	R-0h			R/W-0h		R-0h	

表 1-35. ADCCTL1 レジスタ・フィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	RESERVED	R	0h	予約済み
13	ADCBSY	R	0h	ADC ビジー。ADC SOC が生成されたときに設定され、S/H パルスの負のエッジ後にハードウェアの 4 ADC クロックでクリアされます。ADC がサンプリング可能かどうかを判定するために ADC ステート・マシンで使用されます。 0 : ADC は次のチャンネルをサンプリング可能 1 : ADC はビジーで別のチャンネルをサンプリング不可能 リセット・タイプ : SYSRSn
12	RESERVED	R	0h	予約済み
11-8	ADCBSYCHN	R	0h	ADC ビジー・チャンネル。ADC 変換開始 (SOC) が生成されるタイミングを設定します。 ADCBSY = 0 : 最後に変換された SOC の値を保持 ADCBSY = 1 : 現在処理中の SOC を反映 0h : SOC0 を現在処理中か、最後に変換 1h : SOC1 を現在処理中か、最後に変換 2h : SOC2 を現在処理中か、最後に変換 3h : SOC3 を現在処理中か、最後に変換 4h : SOC4 を現在処理中か、最後に変換 5h : SOC5 を現在処理中か、最後に変換 6h : SOC6 を現在処理中か、最後に変換 7h : SOC7 を現在処理中か、最後に変換 8h : SOC8 を現在処理中か、最後に変換 9h : SOC9 を現在処理中か、最後に変換 Ah : SOC10 を現在処理中か、最後に変換 Bh : SOC11 を現在処理中か、最後に変換 Ch : SOC12 を現在処理中か、最後に変換 Dh : SOC13 を現在処理中か、最後に変換 Eh : SOC14 を現在処理中か、最後に変換 Fh : SOC15 を現在処理中か、最後に変換 リセット・タイプ : SYSRSn
7	ADCPWDNZ	R/W	0h	ADC パワーダウン (アクティブ Low)。このビットは、アナログ・コア内のすべてのアナログ回路のパワーアップとパワーダウンを制御します。 0 : コア内のすべてのアナログ回路をパワーダウン 1 : コア内のすべてのアナログ回路をパワー・アップ リセット・タイプ : SYSRSn
6-3	RESERVED	R	0h	予約済み

表 1-35. ADCCTL1 レジスタ・フィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
2	INPULSEPOS	R/W	0h	ADC 割り込みパルスの位置。 0：割り込みパルスは、ADC が (アキュジション・ウィンドウの最後で) 変換を開始してから、ADCINTCYCLE.OFFSET レジスタで指定されている SYSCLK サイクル数が経過したときに生成されます。 1：割り込みパルスは、変換の最後に、ADC の結果が結果レジスタにラッチされる 1 サイクル前に生成されます。 リセット・タイプ：SYSRSn
1-0	RESERVED	R	0h	予約済み

1.15.3.2 ADCCTL2 レジスタ (オフセット = 1h) [リセット = 0h]

図 1-41 に ADCCTL2 を示し、表 1-36 にその説明を示します。

[概略表](#)に戻ります。

ADC 制御 2 レジスタ

図 1-41. ADCCTL2 レジスタ

15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED		PRESCALE			
R/W-0h	R/W-0h	R-0h		R/W-0h			

表 1-36. ADCCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-13	RESERVED	R	0h	予約済み
12-8	RESERVED	R	0h	予約済み
7	RESERVED	R/W	0h	予約済み
6	RESERVED	R/W	0h	予約済み
5-4	RESERVED	R	0h	予約済み
3-0	PRESCALE	R/W	0h	ADC クロック・プリスケアラ。 0000 ADCCLK = 入力クロック / 1.0 0001 無効 0010 ADCCLK = 入力クロック / 2.0 0011 ADCCLK = 入力クロック / 2.5 0100 ADCCLK = 入力クロック / 3.0 0101 ADCCLK = 入力クロック / 3.5 0110 ADCCLK = 入力クロック / 4.0 0111 ADCCLK = 入力クロック / 4.5 1000 ADCCLK = 入力クロック / 5.0 1001 ADCCLK = 入力クロック / 5.5 1010 ADCCLK = 入力クロック / 6.0 1011 ADCCLK = 入力クロック / 6.5 1100 ADCCLK = 入力クロック / 7.0 1101 ADCCLK = 入力クロック / 7.5 1110 ADCCLK = 入力クロック / 8.0 1111 ADCCLK = 入力クロック / 8.5 リセット・タイプ : SYSRSn

1.15.3.3 ADCBURSTCTL レジスタ (オフセット = 2h) [リセット = 0h]

図 1-42 に ADCBURSTCTL を示し、表 1-37 にその説明を示します。

概略表に戻ります。

ADC バースト制御レジスタ

図 1-42. ADCBURSTCTL レジスタ

15	14	13	12	11	10	9	8
BURSTEN	RESERVED			BURSTSIZE			
R/W-0h	R-0h			R/W-0h			
7	6	5	4	3	2	1	0
RESERVED		BURSTTRIGSEL					
R-0h		R/W-0h					

表 1-37. ADCBURSTCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	BURSTEN	R/W	0h	SOC バースト・モード・イネーブル。このビットにより、SOC バースト・モード動作がイネーブルになります。 0: ビット・モードがディセーブル。 1: バースト・モードがイネーブル。 リセット・タイプ: SYSRSn
14-12	RESERVED	R	0h	予約済み
11-8	BURSTSIZE	R/W	0h	SOC バースト・サイズ選択。このビット・フィールドは、バースト変換シーケンスが開始したときに変換する SOC の数を決定します。変換する最初の SOC はラウンド・ロビン・ポイントで指定されます。このポイントは、SOC が変換されるごとに次に進みます。 0h: 1 個の SOC を変換 1h: 2 個の SOC を変換 2h: 3 個の SOC を変換 3h: 4 個の SOC を変換 4h: 5 個の SOC を変換 5h: 6 個の SOC を変換 6h: 7 個の SOC を変換 7h: 8 個の SOC を変換 8h: 9 個の SOC を変換 9h: 10 個の SOC を変換 Ah: 11 個の SOC を変換 Bh: 12 個の SOC を変換 Ch: 13 個の SOC を変換 Dh: 14 個の SOC を変換 Eh: 15 個の SOC を変換 Fh: 16 個の SOC を変換 リセット・タイプ: SYSRSn
7-6	RESERVED	R	0h	予約済み

表 1-37. ADCBURSTCTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5-0	BURSTTRIGSEL	R/W	0h	<p>SOC バースト・トリガ・ソース選択。バースト変換シーケンスを開始するトリガを設定します。</p> <p>注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。</p> <p>00h BURSTTRIG0 - ソフトウェアのみ 01h BURSTTRIG1 - CPU1 タイマ 0、TINT0n 02h BURSTTRIG2 - CPU1 タイマ 1、TINT1n 03h BURSTTRIG3 - CPU1 タイマ 2、TINT2n 04h BURSTTRIG4 - GPIO、入カク로스バー INPUT5 05h BURSTTRIG5 - ePWM1、ADCSOCA 06h BURSTTRIG6 - ePWM1、ADCSOCB 07h BURSTTRIG7 - ePWM2、ADCSOCA 08h BURSTTRIG8 - ePWM2、ADCSOCB 09h BURSTTRIG9 - ePWM3、ADCSOCA 0Ah BURSTTRIG10 - ePWM3、ADCSOCB 0Bh BURSTTRIG11 - ePWM4、ADCSOCA 0Ch BURSTTRIG12 - ePWM4、ADCSOCB 0Dh BURSTTRIG13 - ePWM5、ADCSOCA 0Eh BURSTTRIG14 - ePWM5、ADCSOCB 0Fh BURSTTRIG15 - ePWM6、ADCSOCA 10h BURSTTRIG16 - ePWM6、ADCSOCB 11h BURSTTRIG17 - ePWM7、ADCSOCA 12h BURSTTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み</p> <p>リセット・タイプ：SYSRSn</p>

1.15.3.4 ADCINTFLG レジスタ (オフセット = 3h) [リセット = 0h]

図 1-43 に ADCINTFLG を示し、表 1-38 にその説明を示します。

概略表に戻ります。

ADC 割り込みフラグ・レジスタ

図 1-43. ADCINTFLG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0h	R-0h	R-0h	R-0h

表 1-38. ADCINTFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R	0h	予約済み
3	ADCINT4	R	0h	ADC 割り込み 4 フラグ。これらのフラグは、最後のクリア後に関連する ADCINT パルスが生成されたかどうかを示します。 0 : ADC 割り込みパルス生成なし 1 ADC 割り込みパルスが生成された ADC 割り込みが割り込み継続モード (INTSELxNy レジスタ) の場合、フラグ・ビットが設定されていても、選択した EOC イベントが発生するたびに割り込みパルスが生成されます。継続モードがイネーブルでない場合、ADCINTFLGCLR レジスタを使用してこのフラグ・ビットをクリアするまで、それ以上の割り込みパルスは生成されません。代わりに、ADCINTOVF レジスタで ADC 割り込みオーバーフロー・イベントが発生します。 リセット・タイプ : SYSRSn
2	ADCINT3	R	0h	ADC 割り込み 3 フラグ。これらのフラグは、最後のクリア後に関連する ADCINT パルスが生成されたかどうかを示します。 0 : ADC 割り込みパルス生成なし 1 ADC 割り込みパルスが生成された ADC 割り込みが割り込み継続モード (INTSELxNy レジスタ) の場合、フラグ・ビットが設定されていても、選択した EOC イベントが発生するたびに割り込みパルスが生成されます。継続モードがイネーブルでない場合、ADCINTFLGCLR レジスタを使用してこのフラグ・ビットをクリアするまで、それ以上の割り込みパルスは生成されません。代わりに、ADCINTOVF レジスタで ADC 割り込みオーバーフロー・イベントが発生します。 リセット・タイプ : SYSRSn
1	ADCINT2	R	0h	ADC 割り込み 2 フラグ。これらのフラグは、最後のクリア後に関連する ADCINT パルスが生成されたかどうかを示します。 0 : ADC 割り込みパルス生成なし 1 ADC 割り込みパルスが生成された ADC 割り込みが割り込み継続モード (INTSELxNy レジスタ) の場合、フラグ・ビットが設定されていても、選択した EOC イベントが発生するたびに割り込みパルスが生成されます。継続モードがイネーブルでない場合、ADCINTFLGCLR レジスタを使用してこのフラグ・ビットをクリアするまで、それ以上の割り込みパルスは生成されません。代わりに、ADCINTOVF レジスタで ADC 割り込みオーバーフロー・イベントが発生します。 リセット・タイプ : SYSRSn

表 1-38. ADCINTFLG レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	ADCINT1	R	0h	<p>ADC 割り込み 1 フラグ。これらのフラグは、最後のクリア後に関連する ADCINT パルスが生成されたかどうかを示します。</p> <p>0 : ADC 割り込みパルス生成なし 1 ADC 割り込みパルスが生成された</p> <p>ADC 割り込みが割り込み継続モード (INTSELxNy レジスタ) の場合、フラグ・ビットが設定されていても、選択した EOC イベントが発生するたびに割り込みパルスが生成されます。継続モードがイネーブルでない場合、ADCINTFLGCLR レジスタを使用してこのフラグ・ビットをクリアするまで、それ以上の割り込みパルスは生成されません。代わりに、ADCINTOVF レジスタで ADC 割り込みオーバーフロー・イベントが発生します。</p> <p>リセット・タイプ : SYSRStn</p>

1.15.3.5 ADCINTFLGCLR レジスタ (オフセット = 4h) [リセット = 0h]

図 1-44 に ADCINTFLGCLR を示し、表 1-39 にその説明を示します。

概略表に戻ります。

ADC 割り込みフラグ・クリア・レジスタ

図 1-44. ADCINTFLGCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

表 1-39. ADCINTFLGCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R	0h	予約済み
3	ADCINT4	R-0/W1C	0h	ADC 割り込み 4 フラグ・クリア。読み取り値は 0 です。 0: アクションなし 1: ADCINTFLG レジスタの該当するフラグ・ビットをクリアします。ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先され、オーバーフロー・ビットは設定されません。 リセット・タイプ: SYSRSn
2	ADCINT3	R-0/W1C	0h	ADC 割り込み 3 フラグ・クリア。読み取り値は 0 です。 0: アクションなし 1: ADCINTFLG レジスタの該当するフラグ・ビットをクリアします。ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先され、オーバーフロー・ビットは設定されません。 リセット・タイプ: SYSRSn
1	ADCINT2	R-0/W1C	0h	ADC 割り込み 2 フラグ・クリア。読み取り値は 0 です。 0: アクションなし 1: ADCINTFLG レジスタの該当するフラグ・ビットをクリアします。ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先され、オーバーフロー・ビットは設定されません。 リセット・タイプ: SYSRSn
0	ADCINT1	R-0/W1C	0h	ADC 割り込み 1 フラグ・クリア。読み取り値は 0 です。 0: アクションなし 1: ADCINTFLG レジスタの該当するフラグ・ビットをクリアします。ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先され、オーバーフロー・ビットは設定されません。 リセット・タイプ: SYSRSn

1.15.3.6 ADCINTOVF レジスタ (オフセット = 5h) [リセット = 0h]

図 1-45 に ADCINTOVF を示し、表 1-40 にその説明を示します。

概略表に戻ります。

ADC 割り込みオーバーフロー レジスタ

図 1-45. ADCINTOVF レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0h	R-0h	R-0h	R-0h

表 1-40. ADCINTOVF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R	0h	予約済み
3	ADCINT4	R	0h	ADC 割り込み 4 オーバーフロー フラグ ADCINT パルスの生成時にオーバーフローが発生したかどうかを示します。対応する ADCINTFLG ビットが設定され、選択された追加の EOC トリガが生成されると、オーバーフロー状態が発生します。 0 : ADC 割り込みオーバーフロー イベントの検出なし。 1 : ADC 割り込みオーバーフロー イベント検出。 オーバーフロー ビットは、連続モード ビットの状態には関係しません。オーバーフロー状態は、このモードの選択に関係なく生成されます。 リセット・タイプ : SYSRSn
2	ADCINT3	R	0h	ADC 割り込み 3 オーバーフロー フラグ ADCINT パルスの生成時にオーバーフローが発生したかどうかを示します。対応する ADCINTFLG ビットが設定され、選択された追加の EOC トリガが生成されると、オーバーフロー状態が発生します。 0 : ADC 割り込みオーバーフロー イベントの検出なし。 1 : ADC 割り込みオーバーフロー イベント検出。 オーバーフロー ビットは、連続モード ビットの状態には関係しません。オーバーフロー状態は、このモードの選択に関係なく生成されます。 リセット・タイプ : SYSRSn
1	ADCINT2	R	0h	ADC 割り込み 2 オーバーフロー フラグ ADCINT パルスの生成時にオーバーフローが発生したかどうかを示します。対応する ADCINTFLG ビットが設定され、選択された追加の EOC トリガが生成されると、オーバーフロー状態が発生します。 0 : ADC 割り込みオーバーフロー イベントの検出なし。 1 : ADC 割り込みオーバーフロー イベント検出。 オーバーフロー ビットは、連続モード ビットの状態には関係しません。オーバーフロー状態は、このモードの選択に関係なく生成されます。 リセット・タイプ : SYSRSn

表 1-40. ADCINTOVF レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	ADCINT1	R	0h	<p>ADC 割り込み 1 オーバーフロー・フラグ ADCINT パルスの生成時にオーバーフローが発生したかどうかを示します。対応する ADCINTFLG ビットが設定され、選択された追加の EOC トリガが生成されると、オーバーフロー状態が発生します。</p> <p>0 : ADC 割り込みオーバーフロー・イベントの検出なし。 1 : ADC 割り込みオーバーフロー・イベント検出。</p> <p>オーバーフロー・ビットは、連続モード・ビットの状態には関係しません。オーバーフロー状態は、このモードの選択に関係なく生成されます。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.7 ADCINTOVFCLR レジスタ (オフセット = 6h) [リセット = 0h]

図 1-46 に ADCINTOVFCLR を示し、表 1-41 にその説明を示します。

概略表に戻ります。

ADC 割り込みオーバーフロー・クリア・レジスタ

図 1-46. ADCINTOVFCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

表 1-41. ADCINTOVFCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R	0h	予約済み
3	ADCINT4	R-0/W1C	0h	ADC 割り込み 4 オーバーフロー・クリア・ビット 0 : アクションなし。 1 : ADCINTOVF レジスタの対応するオーバーフロー・ビットをクリアします。ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCINTOVF レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCINTOVF ビットは設定されます。 リセット・タイプ : SYSRSn
2	ADCINT3	R-0/W1C	0h	ADC 割り込み 3 オーバーフロー・クリア・ビット 0 : アクションなし。 1 : ADCINTOVF レジスタの対応するオーバーフロー・ビットをクリアします。ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCINTOVF レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCINTOVF ビットは設定されます。 リセット・タイプ : SYSRSn
1	ADCINT2	R-0/W1C	0h	ADC 割り込み 2 オーバーフロー・クリア・ビット 0 : アクションなし。 1 : ADCINTOVF レジスタの対応するオーバーフロー・ビットをクリアします。ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCINTOVF レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCINTOVF ビットは設定されます。 リセット・タイプ : SYSRSn
0	ADCINT1	R-0/W1C	0h	ADC 割り込み 1 オーバーフロー・クリア・ビット 0 : アクションなし。 1 : ADCINTOVF レジスタの対応するオーバーフロー・ビットをクリアします。ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCINTOVF レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCINTOVF ビットは設定されます。 リセット・タイプ : SYSRSn

1.15.3.8 ADCINTSEL1N2 レジスタ (オフセット = 7h) [リセット = 0h]

図 1-47 に ADCINTSEL1N2 を示し、表 1-42 にその説明を示します。

概略表に戻ります。

ADC 割り込み 1 および 2 選択レジスタ

図 1-47. ADCINTSEL1N2 レジスタ

15	14	13	12	11	10	9	8
RESERVED	INT2CONT	INT2E	RESERVED			INT2SEL	
R-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	INT1CONT	INT1E	RESERVED			INT1SEL	
R-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	

表 1-42. ADCINTSEL1N2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	INT2CONT	R/W	0h	ADCINT2 割り込み継続モード 0: ADCINT2 フラグ (ADCINTFLG レジスタ) がユーザーによってクリアされるまで、ADCINT2 パルスは生成しない。 1: フラグ・ビットがクリアされているかどうかにかかわらず、EOC パルスが生成されるたびに ADCINT2 パルスを生成。 リセット・タイプ: SYSRSn
13	INT2E	R/W	0h	ADCINT2 割り込みイネーブル 0: ADCINT2 はディセーブル 1: ADCINT2 はイネーブル リセット・タイプ: SYSRSn
12	RESERVED	R	0h	予約済み
11-8	INT2SEL	R/W	0h	ADCINT2 EOC ソース選択 0h: EOC0 で ADCINT2 をトリガ 1h: EOC1 で ADCINT2 をトリガ 2h: EOC2 で ADCINT2 をトリガ 3h: EOC3 で ADCINT2 をトリガ 4h: EOC4 で ADCINT2 をトリガ 5h: EOC5 で ADCINT2 をトリガ 6h: EOC6 で ADCINT2 をトリガ 7h: EOC7 で ADCINT2 をトリガ 8h: EOC8 で ADCINT2 をトリガ 9h: EOC9 で ADCINT2 をトリガ Ah: EOC10 で ADCINT2 をトリガ Bh: EOC11 で ADCINT2 をトリガ Ch: EOC12 で ADCINT2 をトリガ Dh: EOC13 で ADCINT2 をトリガ Eh: EOC14 で ADCINT2 をトリガ Fh: EOC15 で ADCINT2 をトリガ リセット・タイプ: SYSRSn
7	RESERVED	R	0h	予約済み
6	INT1CONT	R/W	0h	ADCINT1 割り込み継続モード 0: ADCINT1 フラグ (ADCINTFLG レジスタ) がユーザーによってクリアされるまで、ADCINT1 パルスは生成しない。 1: フラグ・ビットがクリアされているかどうかにかかわらず、EOC パルスが生成されるたびに ADCINT1 パルスを生成。 リセット・タイプ: SYSRSn
5	INT1E	R/W	0h	ADCINT1 割り込みイネーブル 0: ADCINT1 はディセーブル 1: ADCINT1 はイネーブル リセット・タイプ: SYSRSn
4	RESERVED	R	0h	予約済み

表 1-42. ADCINTSEL1N2 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3-0	INT1SEL	R/W	0h	ADCINT1 EOC ソース選択 0h : EOC0 で ADCINT1 をトリガ 1h : EOC1 で ADCINT1 をトリガ 2h : EOC2 で ADCINT1 をトリガ 3h : EOC3 で ADCINT1 をトリガ 4h : EOC4 で ADCINT1 をトリガ 5h : EOC5 で ADCINT1 をトリガ 6h : EOC6 で ADCINT1 をトリガ 7h : EOC7 で ADCINT1 をトリガ 8h : EOC8 で ADCINT1 をトリガ 9h : EOC9 で ADCINT1 をトリガ Ah : EOC10 で ADCINT1 をトリガ Bh : EOC11 で ADCINT1 をトリガ Ch : EOC12 で ADCINT1 をトリガ Dh : EOC13 で ADCINT1 をトリガ Eh : EOC14 で ADCINT1 をトリガ Fh : EOC15 で ADCINT1 をトリガ リセット・タイプ : SYSRSn

1.15.3.9 ADCINTSEL3N4 レジスタ (オフセット = 8h) [リセット = 0h]

図 1-48 に ADCINTSEL3N4 を示し、表 1-43 にその説明を示します。

概略表に戻ります。

ADC 割り込み 3 および 4 選択レジスタ

図 1-48. ADCINTSEL3N4 レジスタ

15	14	13	12	11	10	9	8
RESERVED	INT4CONT	INT4E	RESERVED			INT4SEL	
R-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	INT3CONT	INT3E	RESERVED			INT3SEL	
R-0h	R/W-0h	R/W-0h	R-0h			R/W-0h	

表 1-43. ADCINTSEL3N4 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	INT4CONT	R/W	0h	ADCINT4 割り込み継続モード 0 : ADCINT4 フラグ (ADCINTFLG レジスタ) がユーザーによってクリアされるまで、ADCINT4 パルスは生成しない。 1 : フラグ・ビットがクリアされているかどうかにかかわらず、EOC パルスが生成されるたびに ADCINT4 パルスを生成。 リセット・タイプ : SYSRSn
13	INT4E	R/W	0h	ADCINT4 割り込みイネーブル 0 : ADCINT4 はディセーブル 1 : ADCINT4 はイネーブル リセット・タイプ : SYSRSn
12	RESERVED	R	0h	予約済み
11-8	INT4SEL	R/W	0h	ADCINT4 EOC ソース選択 0h : EOC0 で ADCINT4 をトリガ 1h : EOC1 で ADCINT4 をトリガ 2h : EOC2 で ADCINT4 をトリガ 3h : EOC3 で ADCINT4 をトリガ 4h : EOC4 で ADCINT4 をトリガ 5h : EOC5 で ADCINT4 をトリガ 6h : EOC6 で ADCINT4 をトリガ 7h : EOC7 で ADCINT4 をトリガ 8h : EOC8 で ADCINT4 をトリガ 9h : EOC9 で ADCINT4 をトリガ Ah : EOC10 で ADCINT4 をトリガ Bh : EOC11 で ADCINT4 をトリガ Ch : EOC12 で ADCINT4 をトリガ Dh : EOC13 で ADCINT4 をトリガ Eh : EOC14 で ADCINT4 をトリガ Fh : EOC15 で ADCINT4 をトリガ リセット・タイプ : SYSRSn
7	RESERVED	R	0h	予約済み
6	INT3CONT	R/W	0h	ADCINT3 割り込み継続モード 0 : ADCINT3 フラグ (ADCINTFLG レジスタ) がユーザーによってクリアされるまで、ADCINT3 パルスは生成しない。 1 : フラグ・ビットがクリアされているかどうかにかかわらず、EOC パルスが生成されるたびに ADCINT3 パルスを生成。 リセット・タイプ : SYSRSn
5	INT3E	R/W	0h	ADCINT3 割り込みイネーブル 0 : ADCINT3 はディセーブル 1 : ADCINT3 はイネーブル リセット・タイプ : SYSRSn
4	RESERVED	R	0h	予約済み

表 1-43. ADCINTSEL3N4 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3-0	INT3SEL	R/W	0h	ADCINT3 EOC ソース選択 0h : EOC0 で ADCINT3 をトリガ 1h : EOC1 で ADCINT3 をトリガ 2h : EOC2 で ADCINT3 をトリガ 3h : EOC3 で ADCINT3 をトリガ 4h : EOC4 で ADCINT3 をトリガ 5h : EOC5 で ADCINT3 をトリガ 6h : EOC6 で ADCINT3 をトリガ 7h : EOC7 で ADCINT3 をトリガ 8h : EOC8 で ADCINT3 をトリガ 9h : EOC9 で ADCINT3 をトリガ Ah : EOC10 で ADCINT3 をトリガ Bh : EOC11 で ADCINT3 をトリガ Ch : EOC12 で ADCINT3 をトリガ Dh : EOC13 で ADCINT3 をトリガ Eh : EOC14 で ADCINT3 をトリガ Fh : EOC15 で ADCINT3 をトリガ リセット・タイプ : SYSRSn

1.15.3.10 ADCSOCPRCTL レジスタ (オフセット= 9h) [リセット= 200h]

図 1-49 に ADCSOCPRCTL を示し、表 1-44 にその説明を示します。

概略表に戻ります。

ADC SOC 優先度制御レジスタ

図 1-49. ADCSOCPRCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED						RRPOINTER	
R-0h						R-10h	
7	6	5	4	3	2	1	0
RRPOINTER			SOCPRIORITY				
R-10h			R/W-0h				

表 1-44. ADCSOCPRCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	RESERVED	R	0h	予約済み
9-5	RRPOINTER	R	10h	ラウンド・ロビン・ポインタ。最後に変換されたラウンド・ロビン SOCx の値を保持します。ラウンド・ロビン方式で変換順序を決定するために使用されます。 00h : SOC0 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC1 です。 01h : SOC1 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC2 です。 02h : SOC2 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC3 です。 03h : SOC3 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC4 です。 04h : SOC4 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC5 です。 05h : SOC5 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC6 です。 06h : SOC6 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC7 です。 07h : SOC7 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC8 です。 08h : SOC8 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC9 です。 09h : SOC9 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC10 です。 0Ah : SOC10 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC11 です。 0Bh : SOC11 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC12 です。 0Ch : SOC12 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC13 です。 0Dh : SOC13 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC14 です。 0Eh : SOC14 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC15 です。 0Fh : SOC15 が最後に変換されたラウンド・ロビン SOC。次に最優先となるのは SOC0 です。 10h : 変換された SOC がいないことを示すリセット値。SOC0 がラウンド・ロビンで最優先されます。SOFTPRES または ADCSOCPRCTL レジスタへの書き込みにより ADC モジュールがリセットされたときに、この値に設定されます。ADCSOCPRCTL レジスタへの書き込みによりリセットされた場合、進行中の変換は完了し、新しい優先度が有効になります。 その他の値は無効。 リセット・タイプ : SYSRSn

表 1-44. ADCSOCPRCTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
4-0	SOC PRIORITY	R/W	0h	<p>SOC 優先度 優先度モードのカットオフ・ポイントと SOCx のラウンド・ロビン・アービトレーションを決定します。 00h : すべてのチャンネルで SOC 優先度はラウンド・ロビン・モードで処理。 01h : SOC0 は高優先度、他のチャンネルはラウンドロビン・モード。 02h : SOC0 ~ SOC1 は高優先度モード、SOC2 ~ SOC15 はラウンド・ロビン・モード。 03h : SOC0 ~ SOC2 は高優先度モード、SOC3 ~ SOC15 はラウンド・ロビン・モード。 04h : SOC0 ~ SOC3 は高優先度モード、SOC4 ~ SOC15 はラウンド・ロビン・モード。 05h : SOC0 ~ SOC4 は高優先度モード、SOC5 ~ SOC15 はラウンド・ロビン・モード。 06h : SOC0 ~ SOC5 は高優先度モード、SOC6 ~ SOC15 はラウンド・ロビン・モード。 07h : SOC0 ~ SOC6 は高優先度モード、SOC7 ~ SOC15 はラウンド・ロビン・モード。 08h : SOC0 ~ SOC7 は高優先度モード、SOC8 ~ SOC15 はラウンド・ロビン・モード。 09h : SOC0 ~ SOC8 は高優先度モード、SOC9 ~ SOC15 はラウンド・ロビン・モード。 0Ah : SOC0 ~ SOC9 は高優先度モード、SOC10 ~ SOC15 はラウンド・ロビン・モード。 0Bh : SOC0 ~ SOC10 は高優先度モード、SOC11 ~ SOC15 はラウンド・ロビン・モード。 0Ch : SOC0 ~ SOC11 は高優先度モード、SOC12 ~ SOC15 はラウンド・ロビン・モード。 0Dh : SOC0 ~ SOC12 は高優先度モード、SOC13 ~ SOC15 はラウンド・ロビン・モード。 0Eh : SOC0 ~ SOC13 は高優先度モード、SOC14 ~ SOC15 はラウンド・ロビン・モード。 0Fh : SOC0 ~ SOC14 は高優先度モード、SOC15 はラウンド・ロビン・モード。 10h : すべての SOC が高優先度モード、SOC 番号によりアービトレーション。 その他の選択は無効。 リセット・タイプ : SYSRSn</p>

1.15.3.11 ADCINTSOCSEL1 レジスタ (オフセット = Ah) [リセット = 0h]

図 1-50 に ADCINTSOCSEL1 を示し、表 1-45 にその説明を示します。

概略表に戻ります。

ADC 割り込み SOC 選択 1 レジスタ

図 1-50. ADCINTSOCSEL1 レジスタ

15	14	13	12	11	10	9	8
SOC7		SOC6		SOC5		SOC4	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SOC3		SOC2		SOC1		SOC0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-45. ADCINTSOCSEL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	SOC7	R/W	0h	SOC7 ADC 割り込みトリガ選択。SOC7 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC7 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC7 をトリガ。10：ADCINT2 で SOC7 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
13-12	SOC6	R/W	0h	SOC6 ADC 割り込みトリガ選択。SOC6 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC6 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC6 をトリガ。10：ADCINT2 で SOC6 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
11-10	SOC5	R/W	0h	SOC5 ADC 割り込みトリガ選択。SOC5 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC5 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC5 をトリガ。10：ADCINT2 で SOC5 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
9-8	SOC4	R/W	0h	SOC4 ADC 割り込みトリガ選択。SOC4 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC4 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC4 をトリガ。10：ADCINT2 で SOC4 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn

表 1-45. ADCINTSOCSEL1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
7-6	SOC3	R/W	0h	SOC3 ADC 割り込みトリガ選択。SOC3 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。 00：SOC3 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。 01：ADCINT1 で SOC3 をトリガ。 10：ADCINT2 で SOC3 をトリガ。 11：無効な選択。 リセット・タイプ：SYSRSn
5-4	SOC2	R/W	0h	SOC2 ADC 割り込みトリガ選択。SOC2 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。 00：SOC2 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。 01：ADCINT1 で SOC2 をトリガ。 10：ADCINT2 で SOC2 をトリガ。 11：無効な選択。 リセット・タイプ：SYSRSn
3-2	SOC1	R/W	0h	SOC1 ADC 割り込みトリガ選択。SOC1 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。 00：SOC1 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。 01：ADCINT1 で SOC1 をトリガ。 10：ADCINT2 で SOC1 をトリガ。 11：無効な選択。 リセット・タイプ：SYSRSn
1-0	SOC0	R/W	0h	SOC0 ADC 割り込みトリガ選択。SOC0 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。 00：SOC0 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。 01：ADCINT1 で SOC0 をトリガ。 10：ADCINT2 で SOC0 をトリガ。 11：無効な選択。 リセット・タイプ：SYSRSn

1.15.3.12 ADCINTSOCSEL2 レジスタ (オフセット = Bh) [リセット = 0h]

図 1-51 に ADCINTSOCSEL2 を示し、表 1-46 にその説明を示します。

概略表に戻ります。

ADC 割り込み SOC 選択 2 レジスタ

図 1-51. ADCINTSOCSEL2 レジスタ

15	14	13	12	11	10	9	8
SOC15		SOC14		SOC13		SOC12	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SOC11		SOC10		SOC9		SOC8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-46. ADCINTSOCSEL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	SOC15	R/W	0h	SOC15 ADC 割り込みトリガ選択。SOC15 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC15 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC15 をトリガ。10：ADCINT2 で SOC15 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
13-12	SOC14	R/W	0h	SOC14 ADC 割り込みトリガ選択。SOC14 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC14 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC14 をトリガ。10：ADCINT2 で SOC14 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
11-10	SOC13	R/W	0h	SOC13 ADC 割り込みトリガ選択。SOC13 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC13 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC13 をトリガ。10：ADCINT2 で SOC13 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
9-8	SOC12	R/W	0h	SOC12 ADC 割り込みトリガ選択。SOC12 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC12 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC12 をトリガ。10：ADCINT2 で SOC12 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn

表 1-46. ADCINTSOCSEL2 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
7-6	SOC11	R/W	0h	SOC11 ADC 割り込みトリガ選択。SOC11 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC11 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC11 をトリガ。10：ADCINT2 で SOC11 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
5-4	SOC10	R/W	0h	SOC10 ADC 割り込みトリガ選択。SOC10 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC10 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC10 をトリガ。10：ADCINT2 で SOC10 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
3-2	SOC9	R/W	0h	SOC9 ADC 割り込みトリガ選択。SOC9 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC9 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC9 をトリガ。10：ADCINT2 で SOC9 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn
1-0	SOC8	R/W	0h	SOC8 ADC 割り込みトリガ選択。SOC8 をトリガする ADCINT を選択します。ADCSOCxCTL レジスタの TRIGSEL フィールドの設定に加えて、このフィールドで選択するトリガが選択されます。00：SOC8 をトリガする ADCINT なし。SOC0 トリガは TRIGSEL フィールドのみで決定されます。01：ADCINT1 で SOC8 をトリガ。10：ADCINT2 で SOC8 をトリガ。11：無効な選択。 リセット・タイプ：SYSRSn

1.15.3.13 ADCSOCFLG1 レジスタ (オフセット = Ch) [リセット = 0h]

図 1-52 に ADCSOCFLG1 を示し、表 1-47 にその説明を示します。

概略表に戻ります。

ADC SOC フラグ 1 レジスタ

図 1-52. ADCSOCFLG1 レジスタ

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-47. ADCSOCFLG1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SOC15	R	0h	SOC15 変換開始フラグ。SOC15 変換の状態を示します。 0 : SOC15 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC15 用に保留中。 SOC15 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
14	SOC14	R	0h	SOC14 変換開始フラグ。SOC14 変換の状態を示します。 0 : SOC14 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC14 用に保留中。 SOC14 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
13	SOC13	R	0h	SOC13 変換開始フラグ。SOC13 変換の状態を示します。 0 : SOC13 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC13 用に保留中。 SOC13 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
12	SOC12	R	0h	SOC12 変換開始フラグ。SOC12 変換の状態を示します。 0 : SOC12 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC12 用に保留中。 SOC12 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn

表 1-47. ADCSOCFLG1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
11	SOC11	R	0h	SOC11 変換開始フラグ。SOC11 変換の状態を示します。 0 : SOC11 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC11 用に保留中。 SOC11 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
10	SOC10	R	0h	SOC10 変換開始フラグ。SOC10 変換の状態を示します。 0 : SOC10 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC10 用に保留中。 SOC10 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
9	SOC9	R	0h	SOC9 変換開始フラグ。SOC9 変換の状態を示します。 0 : SOC9 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC9 用に保留中。 SOC9 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
8	SOC8	R	0h	SOC8 変換開始フラグ。SOC8 変換の状態を示します。 0 : SOC8 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC8 用に保留中。 SOC8 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
7	SOC7	R	0h	SOC7 変換開始フラグ。SOC7 変換の状態を示します。 0 : SOC7 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC7 用に保留中。 SOC7 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn
6	SOC6	R	0h	SOC6 変換開始フラグ。SOC6 変換の状態を示します。 0 : SOC6 用に保留中のサンプルなし。 1 : トリガを受信し、サンプルが SOC6 用に保留中。 SOC6 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn

表 1-47. ADCSOCFLG1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	SOC5	R	0h	<p>SOC5 変換開始フラグ。SOC5 変換の状態を示します。</p> <p>0 : SOC5 用に保留中のサンプルなし。</p> <p>1 : トリガを受信し、サンプルが SOC5 用に保留中。</p> <p>SOC5 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
4	SOC4	R	0h	<p>SOC4 変換開始フラグ。SOC4 変換の状態を示します。</p> <p>0 : SOC4 用に保留中のサンプルなし。</p> <p>1 : トリガを受信し、サンプルが SOC4 用に保留中。</p> <p>SOC4 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
3	SOC3	R	0h	<p>SOC3 変換開始フラグ。SOC3 変換の状態を示します。</p> <p>0 : SOC3 用に保留中のサンプルなし。</p> <p>1 : トリガを受信し、サンプルが SOC3 用に保留中。</p> <p>SOC3 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
2	SOC2	R	0h	<p>SOC2 変換開始フラグ。SOC2 変換の状態を示します。</p> <p>0 : SOC2 用に保留中のサンプルなし。</p> <p>1 : トリガを受信し、サンプルが SOC2 用に保留中。</p> <p>SOC2 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
1	SOC1	R	0h	<p>SOC1 変換開始フラグ。SOC1 変換の状態を示します。</p> <p>0 : SOC1 用に保留中のサンプルなし。</p> <p>1 : トリガを受信し、サンプルが SOC1 用に保留中。</p> <p>SOC1 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
0	SOC0	R	0h	<p>SOC0 変換開始フラグ。SOC0 変換の状態を示します。</p> <p>0 : SOC0 用に保留中のサンプルなし。</p> <p>1 : トリガを受信し、サンプルが SOC0 用に保留中。</p> <p>SOC0 の変換が開始すると、このビットはクリアされます。同じサイクルでこのビットの設定要求とクリア要求の両方を受信した場合、それらのソースに関係なく、このビットは設定され、クリア要求は無視されます。この場合、このビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.14 ADCSOCFRC1 レジスタ (オフセット = Dh) [リセット = 0h]

図 1-53 に ADCSOCFRC1 を示し、表 1-48 にその説明を示します。

概略表に戻ります。

ADC SOC 強制 1 レジスタ

図 1-53. ADCSOCFRC1 レジスタ

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0/W1S-0h							
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0/W1S-0h							

表 1-48. ADCSOCFRC1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SOC15	R-0/W1S	0h	<p>SOC15 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC15 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC15 フラグ・ビットを 1 に強制。SOC15 が優先になった時点で変換が開始します。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC15 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn</p>
14	SOC14	R-0/W1S	0h	<p>SOC14 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC14 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC14 フラグ・ビットを 1 に強制。SOC14 が優先になった時点で変換が開始します。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC14 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn</p>
13	SOC13	R-0/W1S	0h	<p>SOC13 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC13 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC13 フラグ・ビットを 1 に強制。SOC13 が優先になった時点で変換が開始します。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC13 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。 リセット・タイプ : SYSRSn</p>

表 1-48. ADCSOCFRC1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
12	SOC12	R-0/W1S	0h	<p>SOC12 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC12 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC12 フラグ・ビットを 1 に強制。SOC12 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC12 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
11	SOC11	R-0/W1S	0h	<p>SOC11 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC11 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC11 フラグ・ビットを 1 に強制。SOC11 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC11 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
10	SOC10	R-0/W1S	0h	<p>SOC10 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC10 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC10 フラグ・ビットを 1 に強制。SOC10 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC10 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
9	SOC9	R-0/W1S	0h	<p>SOC9 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC9 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC9 フラグ・ビットを 1 に強制。SOC9 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC9 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>

表 1-48. ADCSOCFRC1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8	SOC8	R-0/W1S	0h	<p>SOC8 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC8 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC8 フラグ・ビットを 1 に強制。SOC8 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC8 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
7	SOC7	R-0/W1S	0h	<p>SOC7 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC7 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC7 フラグ・ビットを 1 に強制。SOC7 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC7 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
6	SOC6	R-0/W1S	0h	<p>SOC6 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC6 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC6 フラグ・ビットを 1 に強制。SOC6 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC6 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
5	SOC5	R-0/W1S	0h	<p>SOC5 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC5 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC5 フラグ・ビットを 1 に強制。SOC5 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC5 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>

表 1-48. ADCSOCFRC1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
4	SOC4	R-0/W1S	0h	<p>SOC4 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC4 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC4 フラグ・ビットを 1 に強制。SOC4 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC4 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
3	SOC3	R-0/W1S	0h	<p>SOC3 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC3 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC3 フラグ・ビットを 1 に強制。SOC3 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC3 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
2	SOC2	R-0/W1S	0h	<p>SOC2 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC2 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC2 フラグ・ビットを 1 に強制。SOC2 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC2 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>
1	SOC1	R-0/W1S	0h	<p>SOC1 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC1 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC1 フラグ・ビットを 1 に強制。SOC1 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC1 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>

表 1-48. ADCSOCFRC1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	SOC0	R-0/W1S	0h	<p>SOC0 強制変換開始ビット。1 を書き込むと、ADCSOCFLG1 レジスタの SOC0 フラグが強制的に 1 になります。これを使用すると、ソフトウェアで変換を開始できます。0 の書き込みは無視されます。このビットの読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC0 フラグ・ビットを 1 に強制。SOC0 が優先になった時点で変換が開始します。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCFLG1 レジスタの SOC0 ビットをクリアしようとした場合、ソフトウェアが優先され、ADCSOCFLG1 ビットは設定されます。この場合、ADCSOCFLG1 ビットが以前に設定されたかどうかにかかわらず、ADCSOCOVF1 レジスタのオーバーフロー・ビットは影響を受けません。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.15 ADCSOCOVF1 レジスタ (オフセット = Eh) [リセット = 0h]

図 1-54 に ADCSOCOVF1 を示し、表 1-49 にその説明を示します。

概略表に戻ります。

ADC SOC オーバーフロー 1 レジスタ

図 1-54. ADCSOCOVF1 レジスタ

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-49. ADCSOCOVF1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SOC15	R	0h	SOC15 変換開始オーバーフロー・フラグ。既存の SOC15 イベントがすでに保留中のときに、ハードウェアで SOC15 イベントが生成されたことを示します。 0 : SOC15 イベント・オーバーフローなし。 1 : SOC15 イベント・オーバーフロー発生。 オーバーフロー状態により SOC15 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。 リセット・タイプ : SYSRSn
14	SOC14	R	0h	SOC14 変換開始オーバーフロー・フラグ。既存の SOC14 イベントがすでに保留中のときに、ハードウェアで SOC14 イベントが生成されたことを示します。 0 : SOC14 イベント・オーバーフローなし。 1 : SOC14 イベント・オーバーフロー発生。 オーバーフロー状態により SOC14 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。 リセット・タイプ : SYSRSn
13	SOC13	R	0h	SOC13 変換開始オーバーフロー・フラグ。既存の SOC13 イベントがすでに保留中のときに、ハードウェアで SOC13 イベントが生成されたことを示します。 0 : SOC13 イベント・オーバーフローなし。 1 : SOC13 イベント・オーバーフロー発生。 オーバーフロー状態により SOC13 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。 リセット・タイプ : SYSRSn
12	SOC12	R	0h	SOC12 変換開始オーバーフロー・フラグ。既存の SOC12 イベントがすでに保留中のときに、ハードウェアで SOC12 イベントが生成されたことを示します。 0 : SOC12 イベント・オーバーフローなし。 1 : SOC12 イベント・オーバーフロー発生。 オーバーフロー状態により SOC12 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。 リセット・タイプ : SYSRSn

表 1-49. ADCSOCOVF1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
11	SOC11	R	0h	<p>SOC11 変換開始オーバーフロー・フラグ。既存の SOC11 イベントがすでに保留中のときに、ハードウェアで SOC11 イベントが生成されたことを示します。</p> <p>0 : SOC11 イベント・オーバーフローなし。 1 : SOC11 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC11 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
10	SOC10	R	0h	<p>SOC10 変換開始オーバーフロー・フラグ。既存の SOC10 イベントがすでに保留中のときに、ハードウェアで SOC10 イベントが生成されたことを示します。</p> <p>0 : SOC10 イベント・オーバーフローなし。 1 : SOC10 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC10 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
9	SOC9	R	0h	<p>SOC9 変換開始オーバーフロー・フラグ。既存の SOC9 イベントがすでに保留中のときに、ハードウェアで SOC9 イベントが生成されたことを示します。</p> <p>0 : SOC9 イベント・オーバーフローなし。 1 : SOC9 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC9 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
8	SOC8	R	0h	<p>SOC8 変換開始オーバーフロー・フラグ。既存の SOC8 イベントがすでに保留中のときに、ハードウェアで SOC8 イベントが生成されたことを示します。</p> <p>0 : SOC8 イベント・オーバーフローなし。 1 : SOC8 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC8 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
7	SOC7	R	0h	<p>SOC7 変換開始オーバーフロー・フラグ。既存の SOC7 イベントがすでに保留中のときに、ハードウェアで SOC7 イベントが生成されたことを示します。</p> <p>0 : SOC7 イベント・オーバーフローなし。 1 : SOC7 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC7 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
6	SOC6	R	0h	<p>SOC6 変換開始オーバーフロー・フラグ。既存の SOC6 イベントがすでに保留中のときに、ハードウェアで SOC6 イベントが生成されたことを示します。</p> <p>0 : SOC6 イベント・オーバーフローなし。 1 : SOC6 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC6 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>

表 1-49. ADCSOCOVF1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	SOC5	R	0h	<p>SOC5 変換開始オーバーフロー・フラグ。既存の SOC5 イベントがすでに保留中のときに、ハードウェアで SOC5 イベントが生成されたことを示します。</p> <p>0 : SOC5 イベント・オーバーフローなし。 1 : SOC5 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC5 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
4	SOC4	R	0h	<p>SOC4 変換開始オーバーフロー・フラグ。既存の SOC4 イベントがすでに保留中のときに、ハードウェアで SOC4 イベントが生成されたことを示します。</p> <p>0 : SOC4 イベント・オーバーフローなし。 1 : SOC4 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC4 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
3	SOC3	R	0h	<p>SOC3 変換開始オーバーフロー・フラグ。既存の SOC3 イベントがすでに保留中のときに、ハードウェアで SOC3 イベントが生成されたことを示します。</p> <p>0 : SOC3 イベント・オーバーフローなし。 1 : SOC3 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC3 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
2	SOC2	R	0h	<p>SOC2 変換開始オーバーフロー・フラグ。既存の SOC2 イベントがすでに保留中のときに、ハードウェアで SOC2 イベントが生成されたことを示します。</p> <p>0 : SOC2 イベント・オーバーフローなし。 1 : SOC2 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC2 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
1	SOC1	R	0h	<p>SOC1 変換開始オーバーフロー・フラグ。既存の SOC1 イベントがすでに保留中のときに、ハードウェアで SOC1 イベントが生成されたことを示します。</p> <p>0 : SOC1 イベント・オーバーフローなし。 1 : SOC1 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC1 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>
0	SOC0	R	0h	<p>SOC0 変換開始オーバーフロー・フラグ。既存の SOC0 イベントがすでに保留中のときに、ハードウェアで SOC0 イベントが生成されたことを示します。</p> <p>0 : SOC0 イベント・オーバーフローなし。 1 : SOC0 イベント・オーバーフロー発生。</p> <p>オーバーフロー状態により SOC0 イベントの処理が停止することはありません。これは、ハードウェア・トリガが処理されなかったことを示すものです。ADCSOCFRC1 レジスタへの書き込みは、このビットに影響を与えません。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.16 ADCSOCOVFCLR1 レジスタ (オフセット = Fh) [リセット = 0h]

図 1-55 に ADCSOCOVFCLR1 を示し、表 1-50 にその説明を示します。

概略表に戻ります。

ADC SOC オーバーフロー・クリア 1 レジスタ

図 1-55. ADCSOCOVFCLR1 レジスタ

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0/W1S-0h							
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0/W1S-0h							

表 1-50. ADCSOCOVFCLR1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SOC15	R-0/W1S	0h	SOC15 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC15 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。 0 : アクションなし。 1 : SOC15 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn
14	SOC14	R-0/W1S	0h	SOC14 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC14 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。 0 : アクションなし。 1 : SOC14 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn
13	SOC13	R-0/W1S	0h	SOC13 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC13 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。 0 : アクションなし。 1 : SOC13 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn
12	SOC12	R-0/W1S	0h	SOC12 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC12 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。 0 : アクションなし。 1 : SOC12 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn

表 1-50. ADCSOCOVFCLR1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
11	SOC11	R-0/W1S	0h	<p>SOC11 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC11 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC11 オーバーフロー・フラグをクリア。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。</p> <p>リセット・タイプ : SYSRSn</p>
10	SOC10	R-0/W1S	0h	<p>SOC10 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC10 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC10 オーバーフロー・フラグをクリア。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。</p> <p>リセット・タイプ : SYSRSn</p>
9	SOC9	R-0/W1S	0h	<p>SOC9 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC9 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC9 オーバーフロー・フラグをクリア。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。</p> <p>リセット・タイプ : SYSRSn</p>
8	SOC8	R-0/W1S	0h	<p>SOC8 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC8 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC8 オーバーフロー・フラグをクリア。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。</p> <p>リセット・タイプ : SYSRSn</p>
7	SOC7	R-0/W1S	0h	<p>SOC7 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC7 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC7 オーバーフロー・フラグをクリア。</p> <p>ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。</p> <p>リセット・タイプ : SYSRSn</p>

表 1-50. ADCSOCOVFCLR1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	SOC6	R-0/W1S	0h	<p>SOC6 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC6 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC6 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>
5	SOC5	R-0/W1S	0h	<p>SOC5 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC5 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC5 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>
4	SOC4	R-0/W1S	0h	<p>SOC4 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC4 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC4 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>
3	SOC3	R-0/W1S	0h	<p>SOC3 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC3 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC3 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>
2	SOC2	R-0/W1S	0h	<p>SOC2 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC2 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC2 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>

表 1-50. ADCSOCOVFCLR1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
1	SOC1	R-0/W1S	0h	<p>SOC1 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC1 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC1 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>
0	SOC0	R-0/W1S	0h	<p>SOC0 変換開始オーバーフロー・フラグのクリア。1 を書き込むと、ADCSOCOVF1 レジスタの SOC0 オーバーフロー・フラグがクリアされます。0 の書き込みは無視されます。読み取り値は常に 0 です。</p> <p>0 : アクションなし。 1 : SOC0 オーバーフロー・フラグをクリア。 ソフトウェアがこのビットを設定しようとしたのと同じクロック・サイクルでハードウェアが ADCSOCOVF1 レジスタのオーバーフロー・ビットを設定しようとした場合、ハードウェアが優先され、ADCSOCOVF1 ビットは設定されます。 リセット・タイプ : SYSRSn</p>

1.15.3.17 ADCSOC0CTL レジスタ (オフセット = 10h) [リセット = 0h]

図 1-56 に ADCSOC0CTL を示し、表 1-51 にその説明を示します。

概略表に戻ります。

ADC SOC0 制御レジスタ

図 1-56. ADCSOC0CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-51. ADCSOC0CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC0 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC0 フィールドとともに、SOC0 が最優先になったときに、ADCSOCFLG1 レジスタの SOC0 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・ トリガ構成に加え、SOC のソフトウェア・ トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カク로스パー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・ タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC0 チャンネル選択。ADC が SOC0 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・ タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-51. ADCSOC0CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC0 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.18 ADCSOC1CTL レジスタ (オフセット = 12h) [リセット = 0h]

図 1-57 に ADCSOC1CTL を示し、表 1-52 にその説明を示します。

概略表に戻ります。

ADC SOC1 制御レジスタ

図 1-57. ADCSOC1CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-52. ADCSOC1CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC1 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC1 フィールドとともに、SOC1 が最優先になったときに、ADCSOCFLG1 レジスタの SOC1 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC1 チャンネル選択。ADC が SOC1 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-52. ADCSOC1CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC1 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.19 ADCSOC2CTL レジスタ (オフセット = 14h) [リセット = 0h]

図 1-58 に ADCSOC2CTL を示し、表 1-53 にその説明を示します。

概略表に戻ります。

ADC SOC2 制御レジスタ

図 1-58. ADCSOC2CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-53. ADCSOC2CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC2 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC2 フィールドとともに、SOC2 が最優先になったときに、ADCSOCFLG1 レジスタの SOC2 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・ トリガ構成に加え、SOC のソフトウェア・ トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・ タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC2 チャンネル選択。ADC が SOC2 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・ タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-53. ADCSOC2CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC2 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.20 ADCSOC3CTL レジスタ (オフセット = 16h) [リセット = 0h]

図 1-59 に ADCSOC3CTL を示し、表 1-54 にその説明を示します。

概略表に戻ります。

ADC SOC3 制御レジスタ

図 1-59. ADCSOC3CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-54. ADCSOC3CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC3 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC3 フィールドとともに、SOC3 が最優先になったときに、ADCSOCFLG1 レジスタの SOC3 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カク로스パー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC3 チャンネル選択。ADC が SOC3 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-54. ADCSOC3CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC3 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.21 ADCSOC4CTL レジスタ (オフセット = 18h) [リセット = 0h]

図 1-60 に ADCSOC4CTL を示し、表 1-55 にその説明を示します。

概略表に戻ります。

ADC SOC4 制御レジスタ

図 1-60. ADCSOC4CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-55. ADCSOC4CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC4 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC4 フィールドとともに、SOC4 が最優先になったときに、ADCSOCFLG1 レジスタの SOC4 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC4 チャンネル選択。ADC が SOC4 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-55. ADCSOC4CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC4 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.22 ADCSOC5CTL レジスタ (オフセット = 1Ah) [リセット = 0h]

図 1-61 に ADCSOC5CTL を示し、表 1-56 にその説明を示します。

概略表に戻ります。

ADC SOC5 制御レジスタ

図 1-61. ADCSOC5CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-56. ADCSOC5CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC5 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC5 フィールドとともに、SOC5 が最優先になったときに、ADCSOCFLG1 レジスタの SOC5 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・ トリガ構成に加え、SOC のソフトウェア・ トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カク로스パー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・ タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC5 チャンネル選択。ADC が SOC5 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・ タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-56. ADCSOC5CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC5 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.23 ADCSOC6CTL レジスタ (オフセット = 1Ch) [リセット = 0h]

図 1-62 に ADCSOC6CTL を示し、表 1-57 にその説明を示します。

概略表に戻ります。

ADC SOC6 制御レジスタ

図 1-62. ADCSOC6CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-57. ADCSOC6CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC6 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC6 フィールドとともに、SOC6 が最優先になったときに、ADCSOCFLG1 レジスタの SOC6 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・ トリガ構成に加え、SOC のソフトウェア・ トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・ タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC6 チャンネル選択。ADC が SOC6 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・ タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-57. ADCSOC6CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC6 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.24 ADCSOC7CTL レジスタ (オフセット = 1Eh) [リセット = 0h]

図 1-63 に ADCSOC7CTL を示し、表 1-58 にその説明を示します。

概略表に戻ります。

ADC SOC7 制御レジスタ

図 1-63. ADCSOC7CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-58. ADCSOC7CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC7 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC7 フィールドとともに、SOC7 が最優先になったときに、ADCSOCFLG1 レジスタの SOC7 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC7 チャンネル選択。ADC が SOC7 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-58. ADCSOC7CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC7 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.25 ADCSOC8CTL レジスタ (オフセット = 20h) [リセット = 0h]

図 1-64 に ADCSOC8CTL を示し、表 1-59 にその説明を示します。

概略表に戻ります。

ADC SOC8 制御レジスタ

図 1-64. ADCSOC8CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-59. ADCSOC8CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC8 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC8 フィールドとともに、SOC8 が最優先になったときに、ADCSOCFLG1 レジスタの SOC8 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・ トリガ構成に加え、SOC のソフトウェア・ トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・ タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC8 チャンネル選択。ADC が SOC8 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・ タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-59. ADCSOC8CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC8 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.26 ADCSOC9CTL レジスタ (オフセット = 22h) [リセット = 0h]

図 1-65 に ADCSOC9CTL を示し、表 1-60 にその説明を示します。

概略表に戻ります。

ADC SOC9 制御レジスタ

図 1-65. ADCSOC9CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-60. ADCSOC9CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC9 トリガ・ ソース選択。このビット・ フィールドは、ADCINTSOCSEL1 レジスタの SOC9 フィールドとともに、SOC9 が最優先になったときに、ADCSOCFLG1 レジスタの SOC9 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・ トリガ構成に加え、SOC のソフトウェア・ トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カク로스パー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・ タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC9 チャンネル選択。ADC が SOC9 を受信したときに変換するチャンネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・ タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-60. ADCSOC9CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC9 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.27 ADCSOC10CTL レジスタ (オフセット = 24h) [リセット = 0h]

図 1-66 に ADCSOC10CTL を示し、表 1-61 にその説明を示します。

概略表に戻ります。

ADC SOC10 制御レジスタ

図 1-66. ADCSOC10CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-61. ADCSOC10CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC10 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC10 フィールドとともに、SOC10 が最優先になったときに、ADCSOCFLG1 レジスタの SOC10 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC10 チャネル選択。ADC が SOC10 を受信したときに変換するチャネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-61. ADCSOC10CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC10 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.28 ADCSOC11CTL レジスタ (オフセット = 26h) [リセット = 0h]

図 1-67 に ADCSOC11CTL を示し、表 1-62 にその説明を示します。

概略表に戻ります。

ADC SOC11 制御レジスタ

図 1-67. ADCSOC11CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-62. ADCSOC11CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC11 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC11 フィールドとともに、SOC11 が最優先になったときに、ADCSOCFLG1 レジスタの SOC11 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC11 チャネル選択。ADC が SOC11 を受信したときに変換するチャネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-62. ADCSOC11CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC11 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.29 ADCSOC12CTL レジスタ (オフセット = 28h) [リセット = 0h]

図 1-68 に ADCSOC12CTL を示し、表 1-63 にその説明を示します。

概略表に戻ります。

ADC SOC12 制御レジスタ

図 1-68. ADCSOC12CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-63. ADCSOC12CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC12 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC12 フィールドとともに、SOC12 が最優先になったときに、ADCSOCFLG1 レジスタの SOC12 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC12 チャネル選択。ADC が SOC12 を受信したときに変換するチャネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-63. ADCSOC12CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC12 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.30 ADCSOC13CTL レジスタ (オフセット = 2Ah) [リセット = 0h]

図 1-69 に ADCSOC13CTL を示し、表 1-64 にその説明を示します。

概略表に戻ります。

ADC SOC13 制御レジスタ

図 1-69. ADCSOC13CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-64. ADCSOC13CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC13 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC13 フィールドとともに、SOC13 が最優先になったときに、ADCSOCFLG1 レジスタの SOC13 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC13 チャネル選択。ADC が SOC13 を受信したときに変換するチャネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-64. ADCSOC13CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC13 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ...</p> <p>1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn</p>

1.15.3.31 ADCSOC14CTL レジスタ (オフセット = 2Ch) [リセット = 0h]

図 1-70 に ADCSOC14CTL を示し、表 1-65 にその説明を示します。

概略表に戻ります。

ADC SOC14 制御レジスタ

図 1-70. ADCSOC14CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-65. ADCSOC14CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC14 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC14 フィールドとともに、SOC14 が最優先になったときに、ADCSOCFLG1 レジスタの SOC14 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスバー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC14 チャネル選択。ADC が SOC14 を受信したときに変換するチャネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-65. ADCSOC14CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	<p>SOC14 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。</p> <p>000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.32 ADCSOC15CTL レジスタ (オフセット = 2Eh) [リセット = 0h]

図 1-71 に ADCSOC15CTL を示し、表 1-66 にその説明を示します。

概略表に戻ります。

ADC SOC15 制御レジスタ

図 1-71. ADCSOC15CTL レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED					TRIGSEL					CHSEL					
R-0h					R/W-0h					R/W-0h					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL	RESERVED					ACQPS									
R/W-0h		R-0h					R/W-0h								

表 1-66. ADCSOC15CTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-27	RESERVED	R	0h	予約済み
26-20	TRIGSEL	R/W	0h	SOC15 トリガ・ソース選択。このビット・フィールドは、ADCINTSOCSEL1 レジスタの SOC15 フィールドとともに、SOC15 が最優先になったときに、ADCSOCFLG1 レジスタの SOC15 フラグを設定して変換を開始するトリガを設定します。 注：ハードウェア・トリガ構成に加え、SOC のソフトウェア・トリガに SOCFRC1 レジスタも使用できます。 00h ADCTRIG0 - ソフトウェアのみ 01h ADCTRIG1 - CPU1 タイマ 0、TINT0n 02h ADCTRIG2 - CPU1 タイマ 1、TINT1n 03h ADCTRIG3 - CPU1 タイマ 2、TINT2n 04h ADCTRIG4 - GPIO、入カクロスパー INPUT5 05h ADCTRIG5 - ePWM1、ADCSOCA 06h ADCTRIG6 - ePWM1、ADCSOCB 07h ADCTRIG7 - ePWM2、ADCSOCA 08h ADCTRIG8 - ePWM2、ADCSOCB 09h ADCTRIG9 - ePWM3、ADCSOCA 0Ah ADCTRIG10 - ePWM3、ADCSOCB 0Bh ADCTRIG11 - ePWM4、ADCSOCA 0Ch ADCTRIG12 - ePWM4、ADCSOCB 0Dh ADCTRIG13 - ePWM5、ADCSOCA 0Eh ADCTRIG14 - ePWM5、ADCSOCB 0Fh ADCTRIG15 - ePWM6、ADCSOCA 10h ADCTRIG16 - ePWM6、ADCSOCB 11h ADCTRIG17 - ePWM7、ADCSOCA 12h ADCTRIG18 - ePWM7、ADCSOCB 13h - 3Fh - 予約済み リセット・タイプ：SYSRSn
19-15	CHSEL	R/W	0h	SOC15 チャネル選択。ADC が SOC15 を受信したときに変換するチャネルを選択します。 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 12h ADCIN18 13h ADCIN19 14h ADCIN20 15h - 1Fh 予約済み リセット・タイプ：SYSRSn
14-9	RESERVED	R	0h	予約済み

表 1-66. ADCSOC15CTL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
8-0	ACQPS	R/W	0h	SOC15 アクイジション・プリスケール。この SOC のサンプル / ホールド・ウィンドウを制御します。ADC を正しく動作させるために、アクイジション時間は少なくとも、1 ADCCLK サイクルにわたって継続する必要があります。デバイスのデータシートにも、サンプル / ホールド・ウィンドウの最小期間が記載されています。 000h : サンプル・ウィンドウは 1 システム・クロック・サイクル 001h : サンプル・ウィンドウは 2 システム・クロック・サイクル 002h : サンプル・ウィンドウは 3 システム・クロック・サイクル ... 1FFh : サンプル・ウィンドウは 512 システム・クロック・サイクル リセット・タイプ : SYSRSn

1.15.3.33 ADCEVTSTAT レジスタ (オフセット = 30h) [リセット = 0h]

図 1-72 に ADCEVTSTAT を示し、表 1-67 にその説明を示します。

概略表に戻ります。

ADC イベント・ステータス・レジスタ

図 1-72. ADCEVTSTAT レジスタ

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-67. ADCEVTSTAT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	PPB4ZERO	R	0h	後処理ブロック 4 のゼロクロス・フラグ。設定されている場合、ADCPP4RESULT レジスタの符号が変化したことを示します。このビットは、EOC 信号によってゲートされます。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
13	PPB4TRIPLO	R	0h	後処理ブロック 4 のトリップ下限フラグ。設定されている場合、デジタル比較トリップ下限イベントが発生したことを示します。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
12	PPB4TRIPHI	R	0h	後処理ブロック 4 のトリップ上限フラグ。設定されている場合、デジタル比較トリップ上限イベントが発生したことを示します。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
11	RESERVED	R	0h	予約済み
10	PPB3ZERO	R	0h	後処理ブロック 3 のゼロクロス・フラグ。設定されている場合、ADCPP3RESULT レジスタの符号が変化したことを示します。このビットは、EOC 信号によってゲートされます。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn

表 1-67. ADCEVTSTAT レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
9	PPB3TRIPLO	R	0h	後処理ブロック 3 のトリップ下限フラグ。設定されている場合、デジタル比較トリップ下限イベントが発生したことを示します。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
8	PPB3TRIPHI	R	0h	後処理ブロック 3 のトリップ上限フラグ。設定されている場合、デジタル比較トリップ上限イベントが発生したことを示します。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
7	RESERVED	R	0h	予約済み
6	PPB2ZERO	R	0h	後処理ブロック 2 のゼロクロス・フラグ。設定されている場合、ADCPPB2RESULT レジスタの符号が変化したことを示します。このビットは、EOC 信号によってゲートされます。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
5	PPB2TRIPLO	R	0h	後処理ブロック 2 のトリップ下限フラグ。設定されている場合、デジタル比較トリップ下限イベントが発生したことを示します。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
4	PPB2TRIPHI	R	0h	後処理ブロック 2 のトリップ上限フラグ。設定されている場合、デジタル比較トリップ上限イベントが発生したことを示します。 注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
3	RESERVED	R	0h	予約済み

表 1-67. ADCEVTSTAT レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
2	PPB1ZERO	R	0h	<p>後処理ブロック 1 のゼロクロス・フラグ。設定されている場合、ADCPPB1RESULT レジスタの符号が変化したことを示します。このビットは、EOC 信号によってゲートされます。</p> <p>注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。</p> <p>注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。</p> <p>リセット・タイプ：SYSRSn</p>
1	PPB1TRIPLO	R	0h	<p>後処理ブロック 1 のトリップ下限フラグ。設定されている場合、デジタル比較トリップ下限イベントが発生したことを示します。</p> <p>注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。</p> <p>注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。</p> <p>リセット・タイプ：SYSRSn</p>
0	PPB1TRIPHI	R	0h	<p>後処理ブロック 1 のトリップ上限フラグ。設定されている場合、デジタル比較トリップ上限イベントが発生したことを示します。</p> <p>注：これらのビットは、ADCEVTINTSEL の対応するイネーブルが設定されていない場合でも設定されます。このため、ISR で ADCEVTSTAT レジスタと ADCEVTINTSEL レジスタの両方を調べて、割り込みのソースを判定する必要があります。</p> <p>注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。</p> <p>リセット・タイプ：SYSRSn</p>

1.15.3.34 ADCEVTCLR レジスタ (オフセット = 32h) [リセット = 0h]

図 1-73 に ADCEVTCLR を示し、表 1-68 にその説明を示します。

概略表に戻ります。

ADC イベント・クリア・レジスタ

図 1-73. ADCEVTCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

表 1-68. ADCEVTCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	PPB4ZERO	R-0/W1S	0h	後処理ブロック 4 のゼロクロス・クリア。ADCEVTSTAT レジスタの対応するゼロクロス・フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
13	PPB4TRIPLO	R-0/W1S	0h	後処理ブロック 4 のトリップ下限クリア。ADCEVTSTAT レジスタの対応するトリップ下限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
12	PPB4TRIPHI	R-0/W1S	0h	後処理ブロック 4 のトリップ上限クリア。ADCEVTSTAT レジスタの対応するトリップ上限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
11	RESERVED	R	0h	予約済み
10	PPB3ZERO	R-0/W1S	0h	後処理ブロック 3 のゼロクロス・クリア。ADCEVTSTAT レジスタの対応するゼロクロス・フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
9	PPB3TRIPLO	R-0/W1S	0h	後処理ブロック 3 のトリップ下限クリア。ADCEVTSTAT レジスタの対応するトリップ下限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
8	PPB3TRIPHI	R-0/W1S	0h	後処理ブロック 3 のトリップ上限クリア。ADCEVTSTAT レジスタの対応するトリップ上限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
7	RESERVED	R	0h	予約済み

表 1-68. ADCEVTCLR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	PPB2ZERO	R-0/W1S	0h	後処理ブロック 2 のゼロクロス・クリア。ADCEVTSTAT レジスタの対応するゼロクロス・フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
5	PPB2TRIPLO	R-0/W1S	0h	後処理ブロック 2 のトリップ下限クリア。ADCEVTSTAT レジスタの対応するトリップ下限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
4	PPB2TRIPHI	R-0/W1S	0h	後処理ブロック 2 のトリップ上限クリア。ADCEVTSTAT レジスタの対応するトリップ上限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
3	RESERVED	R	0h	予約済み
2	PPB1ZERO	R-0/W1S	0h	後処理ブロック 1 のゼロクロス・クリア。ADCEVTSTAT レジスタの対応するゼロクロス・フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
1	PPB1TRIPLO	R-0/W1S	0h	後処理ブロック 1 のトリップ下限クリア。ADCEVTSTAT レジスタの対応するトリップ下限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn
0	PPB1TRIPHI	R-0/W1S	0h	後処理ブロック 1 のトリップ上限クリア。ADCEVTSTAT レジスタの対応するトリップ上限フラグをクリアします。 注：ソフトウェアがクリア・ビットを設定しようとしているのと同じサイクルでハードウェアがフラグ・ビットを設定しようとした場合、ハードウェアが優先されます。 リセット・タイプ：SYSRSn

1.15.3.35 ADCEVTSEL レジスタ (オフセット = 34h) [リセット = 0h]

図 1-74 に ADCEVTSEL を示し、表 1-69 にその説明を示します。

概略表に戻ります。

ADC イベント選択レジスタ

図 1-74. ADCEVTSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

表 1-69. ADCEVTSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	PPB4ZERO	R/W	0h	後処理ブロック 4 のゼロクロス・イベント・イネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
13	PPB4TRIPLO	R/W	0h	後処理ブロック 4 のトリップ下限イベント・イネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
12	PPB4TRIPHI	R/W	0h	後処理ブロック 4 のトリップ上限イベント・イネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
11	RESERVED	R	0h	予約済み
10	PPB3ZERO	R/W	0h	後処理ブロック 3 のゼロクロス・イベント・イネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
9	PPB3TRIPLO	R/W	0h	後処理ブロック 3 のトリップ下限イベント・イネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
8	PPB3TRIPHI	R/W	0h	後処理ブロック 3 のトリップ上限イベント・イネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
7	RESERVED	R	0h	予約済み

表 1-69. ADCEVTSEL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	PPB2ZERO	R/W	0h	後処理ブロック 2 のゼロクロス・イベント・イネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
5	PPB2TRIPLO	R/W	0h	後処理ブロック 2 のトリップ下限イベント・イネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
4	PPB2TRIPHI	R/W	0h	後処理ブロック 2 のトリップ上限イベント・イネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
3	RESERVED	R	0h	予約済み
2	PPB1ZERO	R/W	0h	後処理ブロック 1 のゼロクロス・イベント・イネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
1	PPB1TRIPLO	R/W	0h	後処理ブロック 1 のトリップ下限イベント・イネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
0	PPB1TRIPHI	R/W	0h	後処理ブロック 1 のトリップ上限イベント・イネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PWM ブロックへのイベント信号をアクティブにできます。このフラグは、PWM ブロックに追加のイベントを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn

1.15.3.36 ADCEVTINTSEL レジスタ (オフセット = 36h) [リセット = 0h]

図 1-75 に ADCEVTINTSEL を示し、表 1-70 にその説明を示します。

概略表に戻ります。

ADC イベント割り込み選択レジスタ

図 1-75. ADCEVTINTSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	RESERVED	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	RESERVED	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

表 1-70. ADCEVTINTSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	PPB4ZERO	R/W	0h	後処理ブロック 4 のゼロクロス割り込みイネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
13	PPB4TRIPLO	R/W	0h	後処理ブロック 4 のトリップ下限割り込みイネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
12	PPB4TRIPHI	R/W	0h	後処理ブロック 4 のトリップ上限割り込みイネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
11	RESERVED	R	0h	予約済み
10	PPB3ZERO	R/W	0h	後処理ブロック 3 のゼロクロス割り込みイネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
9	PPB3TRIPLO	R/W	0h	後処理ブロック 3 のトリップ下限割り込みイネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
8	PPB3TRIPHI	R/W	0h	後処理ブロック 3 のトリップ上限割り込みイネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ：SYSRSn
7	RESERVED	R	0h	予約済み

表 1-70. ADCEVTINTSEL レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	PPB2ZERO	R/W	0h	後処理ブロック 2 のゼロクロス割り込みイネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ : SYSRSn
5	PPB2TRIPLO	R/W	0h	後処理ブロック 2 のトリップ下限割り込みイネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ : SYSRSn
4	PPB2TRIPHI	R/W	0h	後処理ブロック 2 のトリップ上限割り込みイネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ : SYSRSn
3	RESERVED	R	0h	予約済み
2	PPB1ZERO	R/W	0h	後処理ブロック 1 のゼロクロス割り込みイネーブル。このビットを設定すると、対応するゼロ・クロス・フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ : SYSRSn
1	PPB1TRIPLO	R/W	0h	後処理ブロック 1 のトリップ下限割り込みイネーブル。このビットを設定すると、対応するトリップ下限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ : SYSRSn
0	PPB1TRIPHI	R/W	0h	後処理ブロック 1 のトリップ上限割り込みイネーブル。このビットを設定すると、対応するトリップ上限フラグの立ち上がりで PIE へのイベント割り込み信号をアクティブにできます。このフラグは、PIE に追加の割り込みを発生させる前にクリアする必要があります。 リセット・タイプ : SYSRSn

1.15.3.37 ADCOSDETECT レジスタ (オフセット = 38h) [リセット = 0h]

図 1-76 に ADCOSDETECT を示し、表 1-71 にその説明を示します。

概略表に戻ります。

ADC 開放 / 短絡検出レジスタ

図 1-76. ADCOSDETECT レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					DETECTCFG		
R-0h					R/W-0h		

表 1-71. ADCOSDETECT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-3	RESERVED	R	0h	予約済み
2-0	DETECTCFG	R/W	0h	ADC 開放 / 短絡検出の構成。このビット・フィールドは、開放 / 短絡検出回路の状態を定義します。 0h：開放 / 短絡検出回路はディセーブル。 1h：開放 / 短絡検出回路はゼロスケールでイネーブル。 2h：開放 / 短絡検出回路はフルスケールでイネーブル。 3h：開放 / 短絡検出回路は (公称) 5/12 スケールでイネーブル。 4h：開放 / 短絡検出回路は (公称) 7/12 スケールでイネーブル。 5h：開放 / 短絡検出回路は、VSSA への (公称) 5K プルダウンでイネーブル。 6h：開放 / 短絡検出回路は、VDDA への (公称) 5K プルアップでイネーブル。 7h：開放 / 短絡検出回路は、VSSA への (公称) 7K プルダウンでイネーブル。 リセット・タイプ：SYSRSn

1.15.3.38 ADCCOUNTER レジスタ (オフセット = 39h) [リセット = 0h]

図 1-77 に ADCCOUNTER を示し、表 1-72 にその説明を示します。

[概略表](#)に戻ります。

ADC カウンタ・レジスタ

図 1-77. ADCCOUNTER レジスタ

15	14	13	12	11	10	9	8
RESERVED				FREECOUNT			
R-0h				R-0h			
7	6	5	4	3	2	1	0
FREECOUNT							
R-0h							

表 1-72. ADCCOUNTER レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-0	FREECOUNT	R	0h	ADC フリーランニング・カウンタ値。このビット・フィールドは、フリーランニング ADC カウンタのステータスを反映します。 リセット・タイプ : SYSRSn

1.15.3.39 ADCREV レジスタ (オフセット= 3Ah) [リセット = 105h]

図 1-78 に ADCREV を示し、表 1-73 にその説明を示します。

概略表に戻ります。

ADC リビジョン・レジスタ

図 1-78. ADCREV レジスタ

15	14	13	12	11	10	9	8
REV							
R-1h							
7	6	5	4	3	2	1	0
タイプ							
R-5h							

表 1-73. ADCREV レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	REV	R	1h	ADC のリビジョン。複数のリビジョン間の相違点を文書化できるようにします。最初のバージョンは 00h とラベル付けされます。 リセット・タイプ：SYSRSn
7-0	タイプ	R	5h	ADC のタイプ。この ADC では常に 5 に設定されます。 リセット・タイプ：SYSRSn

1.15.3.40 ADCOFFTRIM レジスタ (オフセット = 3Bh) [リセット = 0h]

図 1-79 に ADCOFFTRIM を示し、表 1-74 にその説明を示します。

[概略表](#)に戻ります。

ADC オフセット・トリム・レジスタ

図 1-79. ADCOFFTRIM レジスタ

15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
OFFTRIM							
R/W-0h							

表 1-74. ADCOFFTRIM レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-8	RESERVED	R/W	0h	予約済み
7-0	OFFTRIM	R/W	0h	ADC オフセット・トリム ADC のオフセット誤差を考慮して、 コンバータの変換結果を上または下に調整します。デバイスのブ ート時に工場出荷時のトリム設定がロードされます。 オフセットは +7 から -8LSB の範囲で補正できます。値は 16 × オフ セット (8 ビット 2 の補数) です。 7 LSB (16*7) = 112 6 LSB (16*6) = 96 5 LSB (16*5) = 80 4 LSB (16*4) = 64 3 LSB (16*3) = 48 2 LSB (16*2) = 32 1 LSB (16*1) = 16 0 LSB (16*0) = 0 -1 LSB (16*(-1)) = 240 : : -7LSB(16*(-7)) = 144 リセット・タイプ : XRSn

1.15.3.41 ADCPPB1CONFIG レジスタ (オフセット = 40h) [リセット = 0h]

図 1-80 に ADCPPB1CONFIG を示し、表 1-75 にその説明を示します。

概略表に戻ります。

ADC PPB1 構成レジスタ

図 1-80. ADCPPB1CONFIG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		CBCEN	TWOSCOMPEN	CONFIG			
R-0h		R/W-0h	R/W-0h	R/W-0h			

表 1-75. ADCPPB1CONFIG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-6	RESERVED	R	0h	予約済み
5	CBCEN	R/W	0h	ADC 後処理ブロックのサイクル単位イネーブル。このビットを設定すると、イベント条件が存在しなくなった場合に、変換後のハードウェア処理回路で変換の ADCEVTSTAT を自動的にクリアできるようになります。 リセット・タイプ：SYSRSn
4	TWOSCOMPEN	R/W	0h	ADC 後処理ブロック 1 の 2 の補数イネーブル。このビットを設定すると、ADCPPB1RESULT レジスタに結果を保存する前に、変換後のハードウェア処理回路でオフセット / リファレンス減算ユニットの出力に対して 2 の補数を実行できるようになります。 0 : ADCPPB1RESULT = ADCRESULTx - ADCPPB1OFFREF 1 : ADCPPB1RESULT = ADCPPB1OFFREF - ADCRESULTx リセット・タイプ：SYSRSn
3-0	CONFIG	R/W	0h	ADC 後処理ブロック 1 の構成。このビット・フィールドは、この後処理ブロックに関連付ける SOC/EOC/RESULT を指定します。 0000 : SOC0/EOC0/RESULT0 を後処理ブロック 1 に関連付け 0001 : SOC1/EOC1/RESULT1 を後処理ブロック 1 に関連付け 0010 : SOC2/EOC2/RESULT2 を後処理ブロック 1 に関連付け 0011 : SOC3/EOC3/RESULT3 を後処理ブロック 1 に関連付け 0100 : SOC4/EOC4/RESULT4 を後処理ブロック 1 に関連付け 0101 : SOC5/EOC5/RESULT5 を後処理ブロック 1 に関連付け 0110 : SOC6/EOC6/RESULT6 を後処理ブロック 1 に関連付け 0111 : SOC7/EOC7/RESULT7 を後処理ブロック 1 に関連付け 1000 : SOC8/EOC8/RESULT8 を後処理ブロック 1 に関連付け 1001 : SOC9/EOC9/RESULT9 を後処理ブロック 1 に関連付け 1010 : SOC10/EOC10/RESULT10 を後処理ブロック 1 に関連付け 1011 : SOC11/EOC11/RESULT11 を後処理ブロック 1 に関連付け 1100 : SOC12/EOC12/RESULT12 を後処理ブロック 1 に関連付け 1101 : SOC13/EOC13/RESULT13 を後処理ブロック 1 に関連付け 1110 : SOC14/EOC14/RESULT14 を後処理ブロック 1 に関連付け 1111 : SOC15/EOC15/RESULT15 を後処理ブロック 1 に関連付け リセット・タイプ：SYSRSn

1.15.3.42 ADCPPB1STAMP レジスタ (オフセット = 41h) [リセット = 0h]

図 1-81 に ADCPPB1STAMP を示し、表 1-76 にその説明を示します。

[概略表](#)に戻ります。

ADC PPB1 サンプル遅延タイムスタンプ・レジスタ

図 1-81. ADCPPB1STAMP レジスタ

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

表 1-76. ADCPPB1STAMP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-0	DLYSTAMP	R	0h	ADC 後処理ブロック 1 の遅延タイムスタンプ。SOC がサンプリングを開始すると、REQSTAMP に含まれる値が ADCCOUNTER.FREECOUNT の値から減算され、このビット・フィールドにロードされます。このシステム・クロック・サイクル数の遅延が、SOC トリガとサンプルの実際の開始との間に挿入されます。 リセット・タイプ : SYSRSn

1.15.3.43 ADCPPB1OFFCAL レジスタ (オフセット = 42h) [リセット = 0h]

図 1-82 に ADCPPB1OFFCAL を示し、表 1-77 にその説明を示します。

概略表に戻ります。

ADC PPB1 オフセット 校正レジスタ

図 1-82. ADCPPB1OFFCAL レジスタ

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

表 1-77. ADCPPB1OFFCAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	RESERVED	R	0h	予約済み
9-0	OFFCAL	R/W	0h	<p>ADC 後処理ブロック 1 のオフセット補正。このビット・フィールドを使用して、ADCIN 回路に固有のシステム・レベル・オフセットをデジタル処理により除去できます。ADC 出力からこの 10 ビット符号付き値が減算されてから、ADCRESULT レジスタに保存されます。</p> <p>000h : 変更なし。ADC 出力を ADCRESULT に直接格納。 001h : ADC の出力 - 1 を ADCRESULT に格納。 002h : ADC の出力 - 2 を ADCRESULT に格納。 ... 200h : ADC の出力 + 512 を ADCRESULT に格納。 ... 3FFh : ADC の出力 + 1 を ADCRESULT に格納。</p> <p>注 : 16 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および FFFFh で飽和します。12 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および 0FFFh で飽和します。</p> <p>注 : 複数の PPB が同じ SOC を指定している場合は、番号が最大の PPB の OFFCAL のみが適用されます。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.44 ADCPPB1OFFREF レジスタ (オフセット = 43h) [リセット = 0h]

図 1-83 に ADCPPB1OFFREF を示し、表 1-78 にその説明を示します。

概略表に戻ります。

ADC PPB1 オフセット・リファレンス・レジスタ

図 1-83. ADCPPB1OFFREF レジスタ

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

表 1-78. ADCPPB1OFFREF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	OFFREF	R/W	0h	<p>ADC 後処理ブロック 1 のオフセット補正。このビット・フィールドを使用して、フィードバック誤差を計算するか、またはリファレンス値を減算してユニポーラ信号をバイポーラに変換できます。この 16 ビット符号なし値は、オプションの 2 の補数関数に渡される前に ADCRESULT レジスタから減算され、ADCPPB1RESULT レジスタに格納されます。この減算は飽和しません。</p> <p>0000h : 変更なし。ADCRESULT の値が渡される。 0001h : ADCRESULT - 1 が渡される。 0002h : ADCRESULT - 2 が渡される。 ... 8000h : ADCRESULT - 32,768 が渡される。 ... FFFFh : ADCRESULT - 65,535 が渡される。</p> <p>注 : 12 ビット・モードでは、このレジスタのサイズは 16 ビットから変更されません。12 ビット・モードでは、このレジスタに 12 ビット値のみが書き込まれるようにするのはユーザーの責任です。 リセット・タイプ : SYSRSn</p>

1.15.3.45 ADCPPB1TRIPHI レジスタ (オフセット = 44h) [リセット = 0h]

図 1-84 に ADCPPB1TRIPHI を示し、表 1-79 にその説明を示します。

概略表に戻ります。

ADC PPB1 トリップ上限レジスタ

図 1-84. ADCPPB1TRIPHI レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							HSIGN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
LIMITHI							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITHI							
R/W-0h							

表 1-79. ADCPPB1TRIPHI レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-17	RESERVED	R	0h	予約済み
16	HSIGN	R/W	0h	上限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITHI ビット・フィールドになります。 リセット・タイプ: SYSRSn
15-0	LIMITHI	R/W	0h	ADC 後処理ブロック 1 のトリップ上限。デジタル・コンパレータのトリップ上限を設定します。16 ビット・モードでは、17 ビットすべてが ADCPPB1RESULT レジスタの PPBRESULT ビット・フィールドの 17 ビットと比較されます。12 ビット・モードでは、ビット 12:0 が ADCPPB1RESULT レジスタの PPBRESULT ビット・フィールドのビット 12:0 と比較されます。 リセット・タイプ: SYSRSn

1.15.3.46 ADCPPB1TRIPLO レジスタ (オフセット = 46h) [リセット = 0h]

図 1-85 に ADCPPB1TRIPLO を示し、表 1-80 にその説明を示します。

概略表に戻ります。

ADC PPB1 トリップ下限 / トリガ・ タイムスタンプ・ レジスタ

図 1-85. ADCPPB1TRIPLO レジスタ

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				RESERVED			LSIGN
R-0h				R-0h			R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

表 1-80. ADCPPB1TRIPLO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-20	REQSTAMP	R	0h	ADC 後処理ブロック 1 の要求タイムスタンプ。トリガで ADCSOCFLG1 レジスタの関連する SOC フラグが設定されると、ADCCOUNTER.FREECOUNT の値がこのビット・ フィールドにロードされます。 リセット・ タイプ : SYSRSn
19-17	RESERVED	R	0h	予約済み
16	LSIGN	R/W	0h	下限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITLO ビット・ フィールドになります。 リセット・ タイプ : SYSRSn
15-0	LIMITLO	R/W	0h	ADC 後処理ブロック 1 のトリップ下限。デジタル・ コンパレータのトリップ下限を設定します。16 ビット・ モードでは、17 ビットすべてが ADCPPB1RESULT レジスタの PPBRESULT ビット・ フィールドの 17 ビットと比較されます。12 ビット・ モードでは、ビット 12:0 が ADCPPB1RESULT レジスタの PPBRRESULT ビット・ フィールドのビット 12:0 と比較されます。 リセット・ タイプ : SYSRSn

1.15.3.47 ADCPPB2CONFIG レジスタ (オフセット = 48h) [リセット = 0h]

図 1-86 に ADCPPB2CONFIG を示し、表 1-81 にその説明を示します。

概略表に戻ります。

ADC PPB2 構成レジスタ

図 1-86. ADCPPB2CONFIG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		CBCEN	TWOSCOMPEN	CONFIG			
R-0h		R/W-0h	R/W-0h	R/W-0h			

表 1-81. ADCPPB2CONFIG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-6	RESERVED	R	0h	予約済み
5	CBCEN	R/W	0h	ADC 後処理ブロックのサイクル単位イネーブル。このビットを設定すると、イベント条件が存在しなくなった場合に、変換後のハードウェア処理回路で変換の ADCEVTSTAT を自動的にクリアできるようになります。 リセット・タイプ：SYSRSn
4	TWOSCOMPEN	R/W	0h	ADC 後処理ブロック 2 の 2 の補数イネーブル。このビットを設定すると、ADCPPB2RESULT レジスタに結果を保存する前に、変換後のハードウェア処理回路でオフセット / リファレンス減算ユニットの出力に対して 2 の補数を実行できるようになります。 0 : ADCPPB2RESULT = ADCRESULTx - ADCPPB2OFFREF 1 : ADCPPB2RESULT = ADCPPB2OFFREF - ADCRESULTx リセット・タイプ：SYSRSn
3-0	CONFIG	R/W	0h	ADC 後処理ブロック 2 の構成。このビット・フィールドは、この後処理ブロックに関連付ける SOC/EOC/RESULT を指定します。 0000 : SOC0/EOC0/RESULT0 を後処理ブロック 2 に関連付け 0001 : SOC1/EOC1/RESULT1 を後処理ブロック 2 に関連付け 0010 : SOC2/EOC2/RESULT2 を後処理ブロック 2 に関連付け 0011 : SOC3/EOC3/RESULT3 を後処理ブロック 2 に関連付け 0100 : SOC4/EOC4/RESULT4 を後処理ブロック 2 に関連付け 0101 : SOC5/EOC5/RESULT5 を後処理ブロック 2 に関連付け 0110 : SOC6/EOC6/RESULT6 を後処理ブロック 2 に関連付け 0111 : SOC7/EOC7/RESULT7 を後処理ブロック 2 に関連付け 1000 : SOC8/EOC8/RESULT8 を後処理ブロック 2 に関連付け 1001 : SOC9/EOC9/RESULT9 を後処理ブロック 2 に関連付け 1010 : SOC10/EOC10/RESULT10 を後処理ブロック 2 に関連付け 1011 : SOC11/EOC11/RESULT11 を後処理ブロック 2 に関連付け 1100 : SOC12/EOC12/RESULT12 を後処理ブロック 2 に関連付け 1101 : SOC13/EOC13/RESULT13 を後処理ブロック 2 に関連付け 1110 : SOC14/EOC14/RESULT14 を後処理ブロック 2 に関連付け 1111 : SOC15/EOC15/RESULT15 を後処理ブロック 2 に関連付け リセット・タイプ：SYSRSn

1.15.3.48 ADCPPB2STAMP レジスタ (オフセット = 49h) [リセット = 0h]

図 1-87 に ADCPPB2STAMP を示し、表 1-82 にその説明を示します。

[概略表](#)に戻ります。

ADC PPB2 サンプル遅延タイムスタンプ・レジスタ

図 1-87. ADCPPB2STAMP レジスタ

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

表 1-82. ADCPPB2STAMP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-0	DLYSTAMP	R	0h	ADC 後処理ブロック 2 の遅延タイムスタンプ。SOC がサンプリングを開始すると、REQSTAMP に含まれる値が ADCCOUNTER.FREECOUNT の値から減算され、このビット・フィールドにロードされます。このシステム・クロック・サイクル数の遅延が、SOC トリガとサンプルの実際の開始との間に挿入されます。 リセット・タイプ : SYSRSn

1.15.3.49 ADCPPB2OFFCAL レジスタ (オフセット = 4Ah) [リセット = 0h]

図 1-88 に ADCPPB2OFFCAL を示し、表 1-83 にその説明を示します。

概略表に戻ります。

ADC PPB2 オフセット 校正レジスタ

図 1-88. ADCPPB2OFFCAL レジスタ

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

表 1-83. ADCPPB2OFFCAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	RESERVED	R	0h	予約済み
9-0	OFFCAL	R/W	0h	<p>ADC 後処理ブロック 2 のオフセット補正。このビット・フィールドを使用して、ADCIN 回路に固有のシステム・レベル・オフセットをデジタル処理により除去できます。ADC 出力からこの 10 ビット符号付き値が減算されてから、ADCRESULT レジスタに保存されます。</p> <p>000h : 変更なし。ADC 出力を ADCRESULT に直接格納。 001h : ADC の出力 - 1 を ADCRESULT に格納。 002h : ADC の出力 - 2 を ADCRESULT に格納。 ... 200h : ADC の出力 + 512 を ADCRESULT に格納。 ... 3FFh : ADC の出力 + 1 を ADCRESULT に格納。</p> <p>注 : 16 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および FFFFh で飽和します。12 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および 0FFFh で飽和します。</p> <p>注 : 複数の PPB が同じ SOC を指定している場合は、番号が最大の PPB の OFFCAL のみが適用されます。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.50 ADCPPB2OFFREF レジスタ (オフセット = 4Bh) [リセット = 0h]

図 1-89 に ADCPPB2OFFREF を示し、表 1-84 にその説明を示します。

概略表に戻ります。

ADC PPB2 オフセット・リファレンス・レジスタ

図 1-89. ADCPPB2OFFREF レジスタ

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

表 1-84. ADCPPB2OFFREF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	OFFREF	R/W	0h	<p>ADC 後処理ブロック 2 のオフセット補正。このビット・フィールドを使用して、フィードバック誤差を計算するか、またはリファレンス値を減算してユニポーラ信号をバイポーラに変換できます。この 16 ビット符号なし値は、オプションの 2 の補数関数に渡される前に ADCRESULT レジスタから減算され、ADCPPB2RESULT レジスタに格納されます。この減算は飽和しません。</p> <p>0000h : 変更なし。ADCRESULT の値が渡される。 0001h : ADCRESULT - 1 が渡される。 0002h : ADCRESULT - 2 が渡される。 ... 8000h : ADCRESULT - 32,768 が渡される。 ... FFFFh : ADCRESULT - 65,535 が渡される。</p> <p>注 : 12 ビット・モードでは、このレジスタのサイズは 16 ビットから変更されません。12 ビット・モードでは、このレジスタに 12 ビット値のみが書き込まれるようにするのはユーザーの責任です。 リセット・タイプ : SYSRSn</p>

1.15.3.51 ADCPPB2TRIPHI レジスタ (オフセット = 4Ch) [リセット = 0h]

図 1-90 に ADCPPB2TRIPHI を示し、表 1-85 にその説明を示します。

概略表に戻ります。

ADC PPB2 トリップ上限レジスタ

図 1-90. ADCPPB2TRIPHI レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							HSIGN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
LIMITHI							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITHI							
R/W-0h							

表 1-85. ADCPPB2TRIPHI レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-17	RESERVED	R	0h	予約済み
16	HSIGN	R/W	0h	上限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITHI ビット・フィールドになります。 リセット・タイプ: SYSRSn
15-0	LIMITHI	R/W	0h	ADC 後処理ブロック 2 のトリップ上限。デジタル・コンパレータのトリップ上限を設定します。16 ビット・モードでは、17 ビットすべてが ADCPPB2RESULT レジスタの PPBRESULT ビット・フィールドの 17 ビットと比較されます。12 ビット・モードでは、ビット 12:0 が ADCPPB2RESULT レジスタの PPBRESULT ビット・フィールドのビット 12:0 と比較されます。 リセット・タイプ: SYSRSn

1.15.3.52 ADCPPB2TRIPLO レジスタ (オフセット = 4Eh) [リセット = 0h]

図 1-91 に ADCPPB2TRIPLO を示し、表 1-86 にその説明を示します。

概略表に戻ります。

ADC PPB2 トリップ下限 / トリガ・ タイムスタンプ・ レジスタ

図 1-91. ADCPPB2TRIPLO レジスタ

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				RESERVED			LSIGN
R-0h				R-0h			R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

表 1-86. ADCPPB2TRIPLO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-20	REQSTAMP	R	0h	ADC 後処理ブロック 2 の要求タイムスタンプ。トリガで ADCSOCFLG1 レジスタの関連する SOC フラグが設定されると、ADCCOUNTER.FREECOUNT の値がこのビット・ フィールドにロードされます。 リセット・ タイプ : SYSRSn
19-17	RESERVED	R	0h	予約済み
16	LSIGN	R/W	0h	下限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITLO ビット・ フィールドになります。 リセット・ タイプ : SYSRSn
15-0	LIMITLO	R/W	0h	ADC 後処理ブロック 2 のトリップ下限。デジタル・ コンパレータのトリップ下限を設定します。16 ビット・ モードでは、17 ビットすべてが ADCPPB2RESULT レジスタの PPBRESULT ビット・ フィールドの 17 ビットと比較されます。12 ビット・ モードでは、ビット 12:0 が ADCPPB2RESULT レジスタの PPBRRESULT ビット・ フィールドのビット 12:0 と比較されます。 リセット・ タイプ : SYSRSn

1.15.3.53 ADCPPB3CONFIG レジスタ (オフセット = 50h) [リセット = 0h]

図 1-92 に ADCPPB3CONFIG を示し、表 1-87 にその説明を示します。

概略表に戻ります。

ADC PPB3 構成レジスタ

図 1-92. ADCPPB3CONFIG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		CBCEN	TWOSCOMPEN	CONFIG			
R-0h		R/W-0h	R/W-0h	R/W-0h			

表 1-87. ADCPPB3CONFIG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-6	RESERVED	R	0h	予約済み
5	CBCEN	R/W	0h	ADC 後処理ブロックのサイクル単位イネーブル。このビットを設定すると、イベント条件が存在しなくなった場合に、変換後のハードウェア処理回路で変換の ADCEVTSTAT を自動的にクリアできるようになります。 リセット・タイプ：SYSRSn
4	TWOSCOMPEN	R/W	0h	ADC 後処理ブロック 3 の 2 の補数イネーブル。このビットを設定すると、ADCPPB3RESULT レジスタに結果を保存する前に、変換後のハードウェア処理回路でオフセット / リファレンス減算ユニットの出力に対して 2 の補数を実行できるようになります。 0 : ADCPPB3RESULT = ADCRESULTx - ADCPPB3OFFREF 1 : ADCPPB3RESULT = ADCPPB3OFFREF - ADCRESULTx リセット・タイプ：SYSRSn
3-0	CONFIG	R/W	0h	ADC 後処理ブロック 3 の構成。このビット・フィールドは、この後処理ブロックに関連付ける SOC/EOC/RESULT を指定します。 0000 : SOC0/EOC0/RESULT0 を後処理ブロック 3 に関連付け 0001 : SOC1/EOC1/RESULT1 を後処理ブロック 3 に関連付け 0010 : SOC2/EOC2/RESULT2 を後処理ブロック 3 に関連付け 0011 : SOC3/EOC3/RESULT3 を後処理ブロック 3 に関連付け 0100 : SOC4/EOC4/RESULT4 を後処理ブロック 3 に関連付け 0101 : SOC5/EOC5/RESULT5 を後処理ブロック 3 に関連付け 0110 : SOC6/EOC6/RESULT6 を後処理ブロック 3 に関連付け 0111 : SOC7/EOC7/RESULT7 を後処理ブロック 3 に関連付け 1000 : SOC8/EOC8/RESULT8 を後処理ブロック 3 に関連付け 1001 : SOC9/EOC9/RESULT9 を後処理ブロック 3 に関連付け 1010 : SOC10/EOC10/RESULT10 を後処理ブロック 3 に関連付け 1011 : SOC11/EOC11/RESULT11 を後処理ブロック 3 に関連付け 1100 : SOC12/EOC12/RESULT12 を後処理ブロック 3 に関連付け 1101 : SOC13/EOC13/RESULT13 を後処理ブロック 3 に関連付け 1110 : SOC14/EOC14/RESULT14 を後処理ブロック 3 に関連付け 1111 : SOC15/EOC15/RESULT15 を後処理ブロック 3 に関連付け リセット・タイプ：SYSRSn

1.15.3.54 ADCPPB3STAMP (オフセット = 51h) [リセット = 0h]

図 1-93 に ADCPPB3STAMP を示し、表 1-88 にその説明を示します。

[概略表](#)に戻ります。

ADC PPB3 サンプル遅延タイムスタンプ・レジスタ

図 1-93. ADCPPB3STAMP レジスタ

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

表 1-88. ADCPPB3STAMP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-0	DLYSTAMP	R	0h	ADC 後処理ブロック 3 の遅延タイムスタンプ。SOC がサンプリングを開始すると、REQSTAMP に含まれる値が ADCCOUNTER.FREECOUNT の値から減算され、このビット・フィールドにロードされます。このシステム・クロック・サイクル数の遅延が、SOC トリガとサンプルの実際の開始との間に挿入されます。 リセット・タイプ : SYSRSn

1.15.3.55 ADCPPB3OFFCAL レジスタ (オフセット = 52h) [リセット = 0h]

図 1-94 に ADCPPB3OFFCAL を示し、表 1-89 にその説明を示します。

概略表に戻ります。

ADC PPB3 オフセット 較正レジスタ

図 1-94. ADCPPB3OFFCAL レジスタ

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

表 1-89. ADCPPB3OFFCAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	RESERVED	R	0h	予約済み
9-0	OFFCAL	R/W	0h	<p>ADC 後処理ブロック 3 のオフセット補正。このビット・フィールドを使用して、ADCIN 回路に固有のシステム・レベル・オフセットをデジタル処理により除去できます。ADC 出力からこの 10 ビット符号付き値が減算されてから、ADCRESULT レジスタに保存されます。</p> <p>000h : 変更なし。ADC 出力を ADCRESULT に直接格納。 001h : ADC の出力 - 1 を ADCRESULT に格納。 002h : ADC の出力 - 2 を ADCRESULT に格納。 ... 200h : ADC の出力 + 512 を ADCRESULT に格納。 ... 3FFh : ADC の出力 + 1 を ADCRESULT に格納。</p> <p>注 : 16 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および FFFFh で飽和します。12 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および 0FFFh で飽和します。</p> <p>注 : 複数の PPB が同じ SOC を指定している場合は、番号が最大の PPB の OFFCAL のみが適用されます。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.56 ADCPPB3OFFREF レジスタ (オフセット = 53h) [リセット = 0h]

図 1-95 に ADCPPB3OFFREF を示し、表 1-90 にその説明を示します。

概略表に戻ります。

ADC PPB3 オフセット・リファレンス・レジスタ

図 1-95. ADCPPB3OFFREF レジスタ

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

表 1-90. ADCPPB3OFFREF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	OFFREF	R/W	0h	<p>ADC 後処理ブロック 3 のオフセット補正。このビット・フィールドを使用して、フィードバック誤差を計算するか、またはリファレンス値を減算してユニポーラ信号をバイポーラに変換できます。この 16 ビット符号なし値は、オプションの 2 の補数関数に渡される前に ADCRESULT レジスタから減算され、ADCPPB3RESULT レジスタに格納されます。この減算は飽和しません。</p> <p>0000h : 変更なし。ADCRESULT の値が渡される。 0001h : ADCRESULT - 1 が渡される。 0002h : ADCRESULT - 2 が渡される。 ... 8000h : ADCRESULT - 32,768 が渡される。 ... FFFFh : ADCRESULT - 65,535 が渡される。</p> <p>注 : 12 ビット・モードでは、このレジスタのサイズは 16 ビットから変更されません。12 ビット・モードでは、このレジスタに 12 ビット値のみが書き込まれるようにするのはユーザーの責任です。 リセット・タイプ : SYSRSn</p>

1.15.3.57 ADCPPB3TRIPHI レジスタ (オフセット = 54h) [リセット = 0h]

図 1-96 に ADCPPB3TRIPHI を示し、表 1-91 にその説明を示します。

概略表に戻ります。

ADC PPB3 トリップ上限レジスタ

図 1-96. ADCPPB3TRIPHI レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							HSIGN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
LIMITHI							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITHI							
R/W-0h							

表 1-91. ADCPPB3TRIPHI レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-17	RESERVED	R	0h	予約済み
16	HSIGN	R/W	0h	上限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITHI ビット・フィールドになります。 リセット・タイプ: SYSRSn
15-0	LIMITHI	R/W	0h	ADC 後処理ブロック 3 のトリップ上限。デジタル・コンパレータのトリップ上限を設定します。16 ビット・モードでは、17 ビットすべてが ADCPPB3RESULT レジスタの PPBRESULT ビット・フィールドの 17 ビットと比較されます。12 ビット・モードでは、ビット 12:0 が ADCPPB3RESULT レジスタの PPBRESULT ビット・フィールドのビット 12:0 と比較されます。 リセット・タイプ: SYSRSn

1.15.3.58 ADCPPB3TRIPLO レジスタ (オフセット = 56h) [リセット = 0h]

図 1-97 に ADCPPB3TRIPLO を示し、表 1-92 にその説明を示します。

概略表に戻ります。

ADC PPB3 トリップ下限 / トリガ・ タイムスタンプ・ レジスタ

図 1-97. ADCPPB3TRIPLO レジスタ

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				RESERVED			LSIGN
R-0h				R-0h			R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

表 1-92. ADCPPB3TRIPLO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-20	REQSTAMP	R	0h	ADC 後処理ブロック 3 の要求タイムスタンプ。トリガで ADCSOCFLG1 レジスタの関連する SOC フラグが設定されると、ADCCOUNTER.FREECOUNT の値がこのビット・ フィールドにロードされます。 リセット・ タイプ : SYSRSn
19-17	RESERVED	R	0h	予約済み
16	LSIGN	R/W	0h	下限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITLO ビット・ フィールドになります。 リセット・ タイプ : SYSRSn
15-0	LIMITLO	R/W	0h	ADC 後処理ブロック 3 のトリップ下限。デジタル・ コンパレータのトリップ下限を設定します。16 ビット・ モードでは、17 ビットすべてが ADCPPB3RESULT レジスタの PPBRESULT ビット・ フィールドの 17 ビットと比較されます。12 ビット・ モードでは、ビット 12:0 が ADCPPB3RESULT レジスタの PPBRRESULT ビット・ フィールドのビット 12:0 と比較されます。 リセット・ タイプ : SYSRSn

1.15.3.59 ADCPPB4CONFIG レジスタ (オフセット = 58h) [リセット = 0h]

図 1-98 に ADCPPB4CONFIG を示し、表 1-93 にその説明を示します。

概略表に戻ります。

ADC PPB4 構成レジスタ

図 1-98. ADCPPB4CONFIG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		CBCEN	TWOSCOMPEN	CONFIG			
R-0h		R/W-0h	R/W-0h	R/W-0h			

表 1-93. ADCPPB4CONFIG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-6	RESERVED	R	0h	予約済み
5	CBCEN	R/W	0h	ADC 後処理ブロックのサイクル単位イネーブル。このビットを設定すると、イベント条件が存在しなくなった場合に、変換後のハードウェア処理回路で変換の ADCEVTSTAT を自動的にクリアできるようになります。 リセット・タイプ：SYSRSn
4	TWOSCOMPEN	R/W	0h	ADC 後処理ブロック 4 の 2 の補数イネーブル。このビットを設定すると、ADCPPB4RESULT レジスタに結果を保存する前に、変換後のハードウェア処理回路でオフセット / リファレンス減算ユニットの出力に対して 2 の補数を実行できるようになります。 0 : ADCPPB4RESULT = ADCRESULTx - ADCPPB4OFFREF 1 : ADCPPB4RESULT = ADCPPB4OFFREF - ADCRESULTx リセット・タイプ：SYSRSn
3-0	CONFIG	R/W	0h	ADC 後処理ブロック 4 の構成。このビット・フィールドは、この後処理ブロックに関連付ける SOC/EOC/RESULT を指定します。 0000 : SOC0/EOC0/RESULT0 を後処理ブロック 4 に関連付け 0001 : SOC1/EOC1/RESULT1 を後処理ブロック 4 に関連付け 0010 : SOC2/EOC2/RESULT2 を後処理ブロック 4 に関連付け 0011 : SOC3/EOC3/RESULT3 を後処理ブロック 4 に関連付け 0100 : SOC4/EOC4/RESULT4 を後処理ブロック 4 に関連付け 0101 : SOC5/EOC5/RESULT5 を後処理ブロック 4 に関連付け 0110 : SOC6/EOC6/RESULT6 を後処理ブロック 4 に関連付け 0111 : SOC7/EOC7/RESULT7 を後処理ブロック 4 に関連付け 1000 : SOC8/EOC8/RESULT8 を後処理ブロック 4 に関連付け 1001 : SOC9/EOC9/RESULT9 を後処理ブロック 4 に関連付け 1010 : SOC10/EOC10/RESULT10 を後処理ブロック 4 に関連付け 1011 : SOC11/EOC11/RESULT11 を後処理ブロック 4 に関連付け 1100 : SOC12/EOC12/RESULT12 を後処理ブロック 4 に関連付け 1101 : SOC13/EOC13/RESULT13 を後処理ブロック 4 に関連付け 1110 : SOC14/EOC14/RESULT14 を後処理ブロック 4 に関連付け 1111 : SOC15/EOC15/RESULT15 を後処理ブロック 4 に関連付け リセット・タイプ：SYSRSn

1.15.3.60 ADCPPB4STAMP レジスタ (オフセット = 59h) [リセット = 0h]

図 1-99 に ADCPPB4STAMP を示し、表 1-94 にその説明を示します。

[概略表](#)に戻ります。

ADC PPB4 サンプル遅延タイムスタンプ・レジスタ

図 1-99. ADCPPB4STAMP レジスタ

15	14	13	12	11	10	9	8
RESERVED				DLYSTAMP			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DLYSTAMP							
R-0h							

表 1-94. ADCPPB4STAMP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-0	DLYSTAMP	R	0h	ADC 後処理ブロック 4 の遅延タイムスタンプ。SOC がサンプリングを開始すると、REQSTAMP に含まれる値が ADCCOUNTER.FREECOUNT の値から減算され、このビット・フィールドにロードされます。このシステム・クロック・サイクル数の遅延が、SOC トリガとサンプルの実際の開始との間に挿入されます。 リセット・タイプ : SYSRSn

1.15.3.61 ADCPPB4OFFCAL レジスタ (オフセット = 5Ah) [リセット = 0h]

図 1-100 に ADCPPB4OFFCAL を示し、表 1-95 にその説明を示します。

概略表に戻ります。

ADC PPB4 オフセット 校正レジスタ

図 1-100. ADCPPB4OFFCAL レジスタ

15	14	13	12	11	10	9	8
RESERVED						OFFCAL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OFFCAL							
R/W-0h							

表 1-95. ADCPPB4OFFCAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	RESERVED	R	0h	予約済み
9-0	OFFCAL	R/W	0h	<p>ADC 後処理ブロック 4 のオフセット補正。このビット・フィールドを使用して、ADCIN 回路に固有のシステム・レベル・オフセットをデジタル処理により除去できます。ADC 出力からこの 10 ビット符号付き値が減算されてから、ADCRESULT レジスタに保存されます。</p> <p>000h : 変更なし。ADC 出力を ADCRESULT に直接格納。 001h : ADC の出力 - 1 を ADCRESULT に格納。 002h : ADC の出力 - 2 を ADCRESULT に格納。 ... 200h : ADC の出力 + 512 を ADCRESULT に格納。 ... 3FFh : ADC の出力 + 1 を ADCRESULT に格納。</p> <p>注 : 16 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および FFFFh で飽和します。12 ビット・モードでは、ADCRESULT レジスタに格納される前に、減算は 0000h および 0FFFh で飽和します。</p> <p>注 : 複数の PPB が同じ SOC を指定している場合は、番号が最大の PPB の OFFCAL のみが適用されます。</p> <p>リセット・タイプ : SYSRSn</p>

1.15.3.62 ADCPPB4OFFREF レジスタ (オフセット = 5Bh) [リセット = 0h]

図 1-101 に ADCPPB4OFFREF を示し、表 1-96 にその説明を示します。

概略表に戻ります。

ADC PPB4 オフセット・リファレンス・レジスタ

図 1-101. ADCPPB4OFFREF レジスタ

15	14	13	12	11	10	9	8
OFFREF							
R/W-0h							
7	6	5	4	3	2	1	0
OFFREF							
R/W-0h							

表 1-96. ADCPPB4OFFREF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	OFFREF	R/W	0h	<p>ADC 後処理ブロック 4 のオフセット補正。このビット・フィールドを使用して、フィードバック誤差を計算するか、またはリファレンス値を減算してユニポーラ信号をバイポーラに変換できます。この 16 ビット符号なし値は、オプションの 2 の補数関数に渡される前に ADCRESULT レジスタから減算され、ADCPPB4RESULT レジスタに格納されます。この減算は飽和しません。</p> <p>0000h : 変更なし。ADCRESULT の値が渡される。 0001h : ADCRESULT - 1 が渡される。 0002h : ADCRESULT - 2 が渡される。 ... 8000h : ADCRESULT - 32,768 が渡される。 ... FFFFh : ADCRESULT - 65,535 が渡される。</p> <p>注 : 12 ビット・モードでは、このレジスタのサイズは 16 ビットから変更されません。12 ビット・モードでは、このレジスタに 12 ビット値のみが書き込まれるようにするのはユーザーの責任です。 リセット・タイプ : SYSRSn</p>

1.15.3.63 ADCPPB4TRIPHI レジスタ (オフセット = 5Ch) [リセット = 0h]

図 1-102 に ADCPPB4TRIPHI を示し、表 1-97 にその説明を示します。

概略表に戻ります。

ADC PPB4 トリップ上限レジスタ

図 1-102. ADCPPB4TRIPHI レジスタ

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							HSIGN
R-0h							R/W-0h
15	14	13	12	11	10	9	8
LIMITHI							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITHI							
R/W-0h							

表 1-97. ADCPPB4TRIPHI レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-17	RESERVED	R	0h	予約済み
16	HSIGN	R/W	0h	上限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITHI ビット・フィールドになります。 リセット・タイプ: SYSRSn
15-0	LIMITHI	R/W	0h	ADC 後処理ブロック 4 のトリップ上限。デジタル・コンパレータのトリップ上限を設定します。16 ビット・モードでは、17 ビットすべてが ADCPPB4RESULT レジスタの PPBRESULT ビット・フィールドの 17 ビットと比較されます。12 ビット・モードでは、ビット 12:0 が ADCPPB4RESULT レジスタの PPBRESULT ビット・フィールドのビット 12:0 と比較されます。 リセット・タイプ: SYSRSn

1.15.3.64 ADCPPB4TRIPLO レジスタ (オフセット = 5Eh) [リセット = 0h]

図 1-103 に ADCPPB4TRIPLO を示し、表 1-98 にその説明を示します。

概略表に戻ります。

ADC PPB4 トリップ下限 / トリガ・ タイムスタンプ・ レジスタ

図 1-103. ADCPPB4TRIPLO レジスタ

31	30	29	28	27	26	25	24
REQSTAMP							
R-0h							
23	22	21	20	19	18	17	16
REQSTAMP				RESERVED			LSIGN
R-0h				R-0h			R/W-0h
15	14	13	12	11	10	9	8
LIMITLO							
R/W-0h							
7	6	5	4	3	2	1	0
LIMITLO							
R/W-0h							

表 1-98. ADCPPB4TRIPLO レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-20	REQSTAMP	R	0h	ADC 後処理ブロック 4 の要求タイムスタンプ。トリガで ADCSOCFLG1 レジスタの関連する SOC フラグが設定されると、ADCCOUNTER.FREECOUNT の値がこのビット・ フィールドにロードされます。 リセット・ タイプ : SYSRSn
19-17	RESERVED	R	0h	予約済み
16	LSIGN	R/W	0h	下限符号ビット。16 ビット ADC モードでは、この符号ビット (17 番目のビット) が LIMITLO ビット・ フィールドになります。 リセット・ タイプ : SYSRSn
15-0	LIMITLO	R/W	0h	ADC 後処理ブロック 4 のトリップ下限。デジタル・ コンパレータのトリップ下限を設定します。16 ビット・ モードでは、17 ビットすべてが ADCPPB4RESULT レジスタの PPBRESULT ビット・ フィールドの 17 ビットと比較されます。12 ビット・ モードでは、ビット 12:0 が ADCPPB4RESULT レジスタの PPBRRESULT ビット・ フィールドのビット 12:0 と比較されます。 リセット・ タイプ : SYSRSn

1.15.3.65 ADCINTCYCLE レジスタ (オフセット = 6Fh) [リセット = 0h]

図 1-104 に ADCINTCYCLE を示し、表 1-99 にその説明を示します。

概略表に戻ります。

ADC Early 割り込み生成サイクル

図 1-104. ADCINTCYCLE レジスタ

15	14	13	12	11	10	9	8
DELAY							
R/W-0h							
7	6	5	4	3	2	1	0
DELAY							
R/W-0h							

表 1-99. ADCINTCYCLE レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	DELAY	R/W	0h	ADC Early 割り込み生成サイクル遅延：割り込みの生成を ADCSOC の立ち下がりエッジから遅延させるシステム・クロック・サイクル数を定義します。 リセット・タイプ：SYSRSn

1.15.3.66 ADCINLTRIM2 レジスタ (オフセット = 72h) [リセット = X]

図 1-105 に ADCINLTRIM2 を示し、表 1-100 にその説明を示します。

[概略表](#)に戻ります。

ADC 直線性トリム 2 レジスタ

図 1-105. ADCINLTRIM2 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM63TO32																															
R/W-X																															

表 1-100. ADCINLTRIM2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	INLTRIM63TO32	R/W	X	ADC 直線性トリム・ビット 63~32。 このレジスタは、テキサス・インスツルメンツのエラッタまたはその他の資料で特に記述されている場合を除き、変更しないでください。このレジスタの内容を変更すると、このモジュールの動作がデータシートの仕様の範囲外になる可能性があります。 リセット・タイプ：XRSn

1.15.3.67 ADCINLTRIM3 レジスタ (オフセット = 74h) [リセット = X]

図 1-106 に ADCINLTRIM3 を示し、表 1-101 にその説明を示します。

概略表に戻ります。

ADC 直線性トリム 3 レジスタ

図 1-106. ADCINLTRIM3 レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INLTRIM95TO64																															
R/W-X																															

表 1-101. ADCINLTRIM3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-0	INLTRIM95TO64	R/W	X	ADC 直線性トリム・ビット 95~64。 このレジスタは、テキサス・インスツルメンツのエラッタまたはその他の資料で特に記述されている場合を除き、変更しないでください。このレジスタの内容を変更すると、このモジュールの動作がデータシートの仕様の範囲外になる可能性があります。 リセット・タイプ：XRSn

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020



拡張パルス幅変調器 (ePWM) ペリフェラルは、民生用および産業用機器で採用されている多くのパワー・エレクトロニクス・システムを制御するための重要な要素です。これらのシステムには、デジタル・モーター制御、スイッチ・モード電源制御、無停電電源 (UPS)、その他の形式の電力変換が含まれます。ePWM ペリフェラルは、デジタル / アナログ変換 (DAC) 機能も実行できます。この場合、デューティ・サイクルが DAC アナログ値に相当します (パワー DAC と呼ばれます)。

この章は、レジスタ保護機能が追加された ePWM タイプ 4 に適用できます。タイプ間の相違点を判断するための、同じタイプの ePWM モジュールを備えたすべてのデバイスのリストと、タイプ内の個々のデバイスの相違点のリストは、『[C2000 リアルタイム制御ペリフェラル・リファレンス・ガイド](#)』をご覧ください。

1.1 はじめに.....	4
1.2 デバイス・ピンの設定.....	11
1.3 ePWM モジュールの概要.....	11
1.4 時間ベース (TB) サブモジュール.....	13
1.5 カウンタ比較 (CC) サブモジュール.....	29
1.6 アクション・フィルタ (AQ) サブモジュール.....	35
1.7 デッドバンド・ジェネレータ (DB) サブモジュール.....	52
1.8 PWM チョッパ (PC) サブモジュール.....	59
1.9 トリップ・ゾーン (TZ) サブモジュール.....	63
1.10 イベント・トリガ (ET) サブモジュール.....	70
1.11 デジタル比較 (DC) サブモジュール.....	75
1.12 ePWM クロスバー.....	86
1.13 電源トポロジへの応用.....	87
1.14 レジスタ・ロック保護.....	105
1.15 高分解能パルス幅変調器 (HRPWM).....	106
1.16 ソフトウェア.....	132
1.17 ePWM レジスタ.....	137

1.1 はじめに

この章では、以下の各サブモジュールに関する概要と情報について説明します。

- 時間ベース (TB) サブモジュール
- カウンタ比較 (CC) サブモジュール
- アクション・フィルタ (AQ) サブモジュール
- デッドバンド・ジェネレータ (DB) サブモジュール
- PWM チョップ (PC) サブモジュール
- トリップ・ゾーン (TZ) サブモジュール
- イベント・トリガ (ET) サブモジュール
- デジタル比較 (DC) サブモジュール

ePWM タイプ 4 は、タイプ 2 と機能的に互換性があります (タイプ 3 は存在しません)。タイプ 4 では、タイプ 2 の機能に加えて、以下の拡張が行われています。

- **レジスタ・アドレス・マップ** ePWM タイプ 4 の新機能のために、追加のレジスタが必要です。アラインメントのしやすさと使いやすさを実現するため、ePWM レジスタのアドレス空間が再割り当てされました。
- **遅延トリップ機能**：ピーク電流モード制御タイプのアプリケーション・シナリオで必要とされる遅延トリップ機能などをサポートするためのデッドバンド挿入機能を実現するため、変更が追加されました。これは、コンパレータ・イベントをトリガ・イベント (イベント T1 および T2) としてアクション・フィルタに取り込むことを可能にすることで実現されています。コンパレータ T1/T2 イベントを使って PWM を編集する場合、PWM 波形の変更は即座には行われません。代わりに、その波形は次の TBCLK に同期します。
- **デッドバンド・ジェネレータ・サブモジュールの拡張**：動的な構成変更を可能にするための DBCTL レジスタのシャドウイング。
- **レジスタのワンショットおよびグローバル・ロード**：ePWM タイプ 4 を使用すると、多相アプリケーションなどで部分負荷を防止するための、シャドウ・レジスタからアクティブ・レジスタへのワンショットおよびグローバル・ロード機能を実現できます。また、ePWM タイプ 4 では、シャドウ・アクティブ・ロード・イベントのプリスケールを設定できます。ePWM タイプ 4 のグローバル・ロードを使うと、割り込みがなくなり、すべてのレジスタが同時にロードされるため、ePWM ソフトウェアが簡単になります。
- **トリップ・ゾーン・サブモジュールの拡張**：各 TZ ソースのトリップ・ステータスを反映するため、独立したフラグが追加されています。バレー・スイッチングなどの特定のパワー・コンバータ・スイッチング手法をサポートするため、トリップ・ゾーン・サブモジュールに変更が加えられました。
- **デジタル比較サブモジュールの拡張**：ブランキング・ウィンドウ・フィルタのレジスタ幅が 8 ビットから 16 ビットに拡大されました。プログラマビリティを向上させるため、DCCAP 機能が拡張されました。
- **PWM 同期関連の拡張機能**：ePWM タイプ 4 では、CMPC および CMPD イベントに基づいて PWM SYNCOUT を生成できます。これらのイベントは PWMSYNC パルス選択にも使用できます。

ePWM タイプ 2 はタイプ 1 と完全互換です。タイプ 2 では、タイプ 1 の機能に加えて、以下の拡張が行われています。

- **高分解能デッドバンド機能**：半周期クロック・モードのデッドバンド RED および FED に高分解能機能が追加されています。
- **デッドバンド・ジェネレータ・サブモジュールの拡張**：ePWM タイプ 2 には、いずれかの PWM 出力の RED と FED の両方を有効化する機能があります。14 ビット・カウンタによりデッドバンドが拡大されており、デッドバンド・レジスタとデッドバンド高分解能レジスタはシャドウイングされます。
- **ePWMxB 出力で利用可能な高分解能拡張機能**：ePWMxB 出力での高分解能周期とデューティ・サイクルの制御を有効化できます。[セクション 1.15](#) で、これについて詳細に説明しています。
- **カウンタ比較サブモジュールの拡張**：ePWM タイプ 2 を使用すると、CMPC と CMPD を比較する追加のカウンタによって、割り込みと SOC イベントを生成できます。
- **イベント・トリガ・サブモジュールの拡張**：割り込み要求と ADC 変換開始を発行するためのプリスケラ・ロジックがカウントできるイベント回数が、最大 15 回まで拡張されました。このプリスケラ・ロジックを使うと、同期イベント時にイベント・カウンタをソフトウェアで初期化できます。
- **デジタル比較サブモジュールの拡張**：デジタル比較トリップ選択ロジック [DCTRIPSEL] は、個別の DCxEVT を生成するため、最大 14 の外部および内部トリップ・ソースのすべての論理和をとる機能に加えて、最大 12 の外部トリップ・ソースを入力クロスバー・ロジックによって選択する機能を備えています。

- **TBPRD および CMPx レジスタへの同時書き込み**：この機能により、任意の ePWM モジュールの TBPRD、CMPA：CMPAHR、CMPB：CMPBHR、CMPC、CMPD への書き込みをその他の任意の ePWM モジュールと関連付けることができ、必要に応じて、すべての ePWM モジュールのこれらのレジスタへの書き込みを特定の ePWM モジュールと関連付けることもできます。
- **TBPRD および CMP レジスタ同期時のシャドウ - アクティブ・ロード**：この機能は、TBPRD および CMPA/B/C/D レジスタの同時書き込みに対応します。

効率的な PWM ペリフェラルには、CPU のオーバーヘッドと介入を最小限に抑えながら、複雑なパルス幅の波形を生成する機能が欠かせません。また、これらの PWM ペリフェラルは、分かりやすく使いやすいと同時に、詳細な設定が可能で非常に柔軟である必要があります。ここで説明する ePWM ユニットの、必要なすべてのタイミングおよび制御リソースを PWM チャンネルごとに割り当てることで、これらの要件に対応しています。リソースの交換も共有も行われていません。その代わりに本 ePWM は、必要に応じて連携して動作できる、独立したリソースを備えた複数の小さなシングル・チャンネル・サブモジュールで構成されています。このモジュール式手法により直交アーキテクチャが可能となり、ペリフェラルの構造をより透過的に観察できるようになるため、ユーザーはその動作をすぐに理解できます。

このドキュメントでは、信号またはサブモジュール名の中の文字「x」を使ってデバイス上の一般的な ePWM インスタンスを表しています。たとえば、出力信号 EPWMxA および EPWMxB は ePWMx インスタンスからの出力信号を意味しています。したがって、EPWM1A と EPWM1B は ePWM1 に属し、同様に EPWM4A と EPWM4B は ePWM4 に属します。

タイプ 0 からタイプ 1 への拡張

- **デッドバンド分解能の向上**：半周期クロック処理で分解能を倍増できるように、デッドバンド・クロッキングが拡張されました。
- **割り込みおよび SOC 生成の拡張**：TBCTR == ゼロと TBCTR == 周期の両方のイベント時に、割り込みと ADC 変換開始を生成できるようになりました。この機能はデュアル・エッジ PWM 制御を可能にします。また、デジタル比較サブモジュールで設定されたイベントから ADC 変換開始を生成できます。
- **高分解能周期機能**：高分解能周期を有効化できます。[セクション 1.15](#) で、これについて詳細に説明しています。
- **デジタル比較サブモジュール**：デジタル比較サブモジュールは、デジタル比較信号に対するフィルタリング、ブランキング、トリップ機能の向上を実現することで、イベント・トリガおよびトリップ・ゾーン・サブモジュールを拡張します。このような機能は、ピーク電流モードの制御とアナログ・コンパレータのサポートに不可欠です。

注

その他の EPWM 同期信号 (EPWMSYNCl および EPWMSYNCO) との混同を避けるため、CMPSS に向かう同期信号の名前が PWMSYNC から EPWMSYNCPER (SYNCPER/PWMSYNCPER/EPWMxSYNCPER) に更新されました。これらの信号の説明については、[表 1-2](#) を参照してください。

1.1.1 EPWM の関連資料

基礎的資料

- [C2000 アカデミー - EPWM](#)
- 『リアルタイム制御リファレンス・ガイド』
– EPWM セクションを参照してください。

入門用資料

- 『C2000 ePWM 開発者ガイド』アプリケーション・レポート
- 『C2000 MCU の拡張パルス幅変調器 (ePWM) のトレーニング』(動画)
- 『柔軟な PWM による多軸駆動の実現』、『マルチレベル・インバータ』アプリケーション・レポート
- 『C2000 ePWM モジュール入門』(動画)
- 『TMS320F280x デジタル・シグナル・コントローラでの D/A コンバータとしての PWM 出力の使い方』アプリケーション・レポート

- 第 1~6 章は、PWM を使って DAC を実装する方法を理解するのに役立つ基本的な資料、派生資料、説明です。以降の章は、システムに実装するための入門用および専門的資料です。
- 『[拡張パルス幅変調器 \(ePWM\) モジュールの使い方](#)』アプリケーション・レポート

専門的資料

- [C2000 リアルタイム・マイクロコントローラ - リファレンス・デザイン](#)
 - 対象となる特定の最終アプリケーションに関連する TI designs を参照してください。
- 『[C2000 タイプ 4 PWM モジュールを使った CRM/ZVS PFC の実装](#)』アプリケーション・レポート
- 『[新しい ePWM 機能の活用による多相制御](#)』アプリケーション・レポート

1.1.2 サブモジュールの概要

ePWM モジュールは、2 つの PWM 出力 (EPWMxA と EPWMxB) で構成された 1 つの PWM チャネル全体を表します。図 1-1 に示すように、1 つのデバイス内で複数の ePWM モジュールがインスタンス化されています。各 ePWM インスタンスは、1 つの例外を除き同一です。一部のインスタンスは、PWM 出力をより精密に制御できるようにハードウェア拡張機能を内蔵しています。この拡張機能とは、高分解能パルス幅変調器 (HRPWM) です ([セクション 1.15](#) を参照)。この機能を備えた ePWM インスタンスを確認するには、本デバイスのデータシートを参照してください。各 ePWM モジュールは、1 から始まる数で示されます。たとえば、システム内で ePWM1 が第 1 のインスタンス、ePWM3 が第 3 のインスタンス、ePWMx が任意のインスタンスを示します。

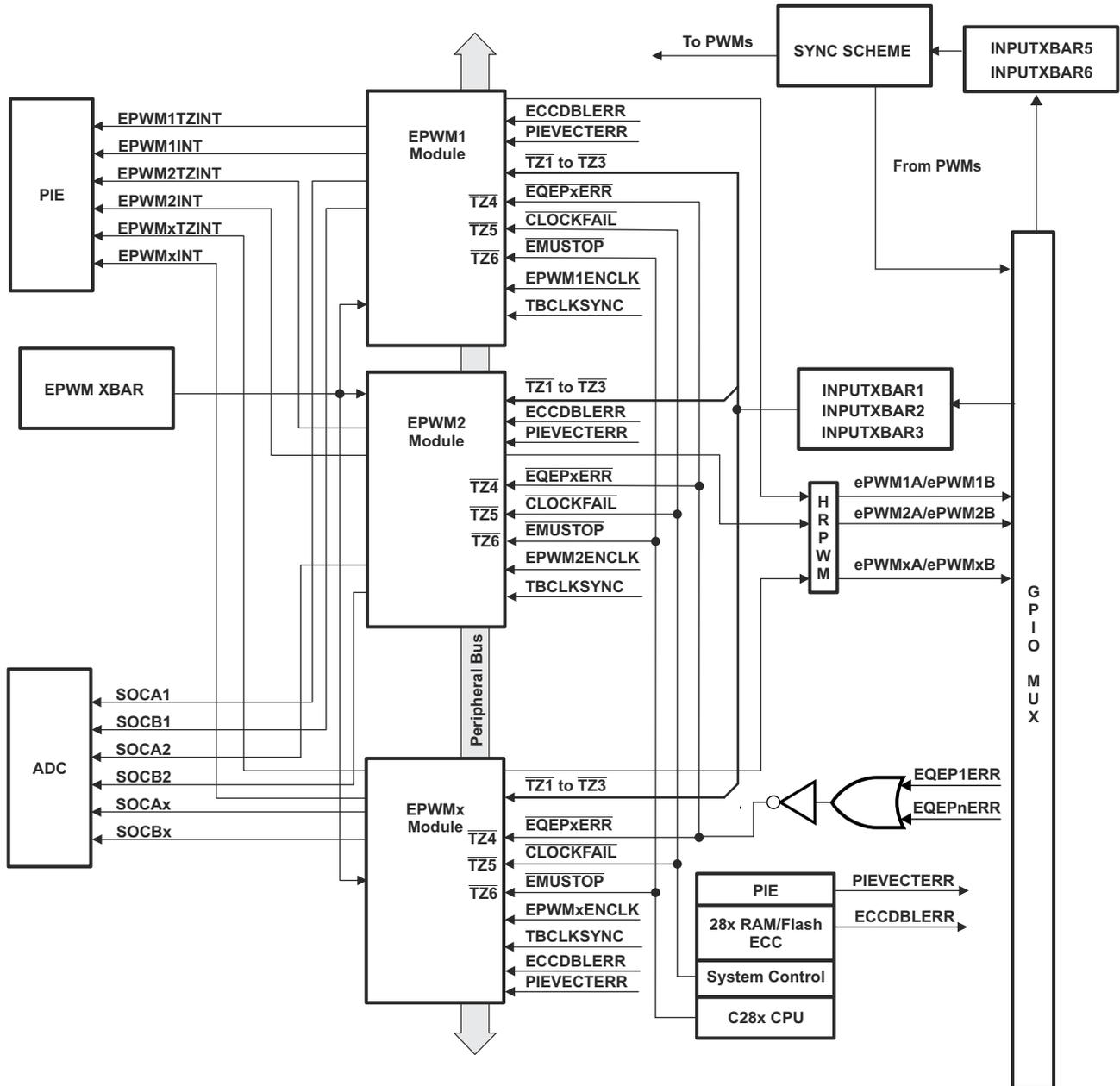
ePWM モジュールは、必要に応じて 1 つのシステムとして動作できるクロック同期方式によって互いに連結されています。さらに、この同期方式をキャプチャ・バリフェラル・サブモジュール (eCAP) に拡張することもできます。サブモジュールの数はデバイスによって異なり、ターゲット・アプリケーションの要求に基づいて決定されます。サブモジュールはスタンドアロンでも動作できます。

各 ePWM モジュールは以下の機能をサポートしています。

- 周期および周波数制御機能を備えた専用の 16 ビット時間ベース・カウンタ
- 以下の構成で使用できる 2 つの PWM 出力 (EPWMxA、EPWMxB)
 - シングル・エッジ動作の 2 つの独立した PWM 出力
 - デュアル・エッジ対称動作の 2 つの独立した PWM 出力
 - デュアル・エッジ非対称動作の 1 つの独立した PWM 出力
- ソフトウェアによる PWM 信号の非同期オーバーライド制御
- その他の ePWM モジュールに対する遅れまたは進み動作のためのプログラマブルな位相制御のサポート
- ハードウェアによってサイクルごとにロック (同期) された位相関係
- 独立した立ち上がりおよび立ち下がりエッジ遅延制御によるデッドバンド生成
- フォルト条件でのサイクル・バイ・サイクル・トリップとワンショット・トリップの両方のプログラマブルなトリップ・ゾーン割り当て
- High、Low、ハイ・インピーダンス状態のいずれかのロジック・レベルを PWM 出力に出力できるトリップ条件
- CPU 割り込みと ADC 変換開始 (SOC) の両方をすべてのイベントがトリガ可能
- 割り込み時の CPU のオーバーヘッドを最小化するプログラマブルなイベント・プリスケールリング
- パルス・トランスによるゲート駆動に便利な、高周波数キャリア信号による PWM チョッピング

各 ePWM モジュールは入力 / 出力信号 (図 1-1 を参照) に接続されています。これらの信号については、以降のセクションで詳しく説明します。

ePWM モジュールの接続順序は、図 1-1 に示す順序とは異なる場合があります。特定のデバイスの同期方式については、[セクション 1.4.3.3](#) を参照してください。各 ePWM モジュールは 8 つのサブモジュールで構成されており、図 1-2 に示す信号によってシステム内で接続されています。



A. この信号は、eQEP サブモジュールを備えたデバイスにのみ存在します。

図 1-1. 複数の ePWM モジュール

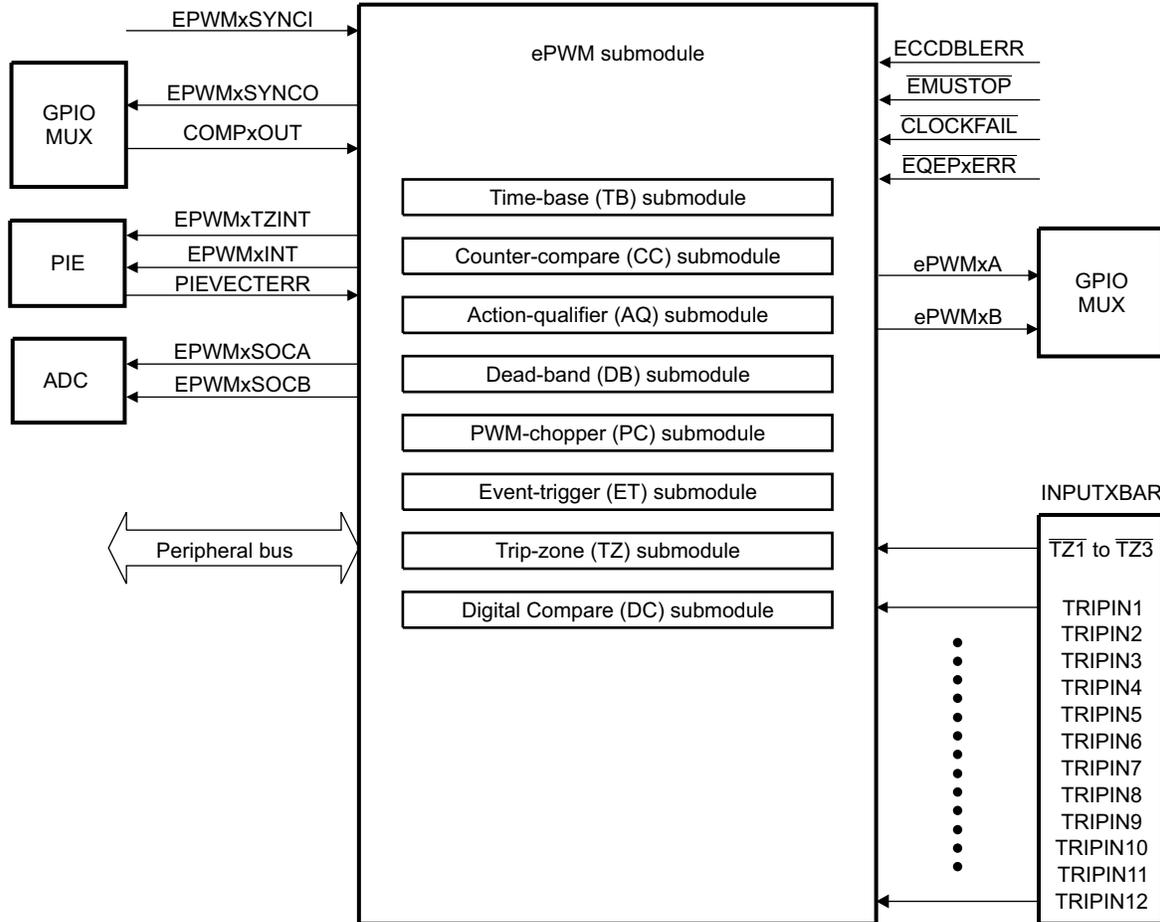


図 1-2. ePWM モジュールのサブモジュールと信号接続

図 1-3 に、1 つの ePWM モジュールの内部の詳細を示します。ePWM モジュールで使用される主な信号は次のとおりです。

- **PWM 出力信号 (EPWMxA、EPWMxB)**

PWM 出力信号は、で説明されている GPIO ペリフェラルを介して、本デバイスの外部で利用できます。

- **トリップ・ゾーン信号 (TZ1 ~ TZ6)**

これらの入力信号は、ePWM モジュールの外部のフォルト条件を ePWM モジュールに通知します。デバイス上の各サブモジュールは、トリップ・ゾーン信号を使用するように、または無視するように設定できます。TZ1 ~ TZ3 のトリップ・ゾーン信号は、入力クロスバー・ロジックを使った、GPIO ペリフェラル経由の非同期入力として構成できます (図 1-51 を参照)。TZ4 は反転 EQEPx エラー信号 (EQEPxERR) に接続されており、(EQEP モジュールを備えたデバイスの場合、) 任意の EQEP サブモジュールから生成できます。TZ5 はシステム・クロック障害ロジックに接続されており、TZ6 は CPU からの EMUSTOP 出力に接続されています。そのため、クロック障害または CPU 停止時のトリップ・アクションを設定できます。

- **時間ベース同期入力 (EPWMxSYNCl)、出力 (EPWMxSYNCO)、ペリフェラル (EPWMxSYNCPER) 信号**

各 ePWM モジュールは、EPWMSYNClNSEL を使って、その他の ePWM モジュールまたはペリフェラルと同期させることができます。各 ePWM モジュールは同期出力信号を生成することもできます。EPWMxSYNCO のソースは、EPWMSYNCOOUTEN と TBCTL2.OSHTSYNClMODE によって、選択および有効化できます。詳細については、[セクション 1.4.3.3](#) を参照してください。

各 ePWM モジュールは、EPWMxSYNCPER と呼ぶ別の PWMSYNCl 信号も生成します。同期の目的で、EPWMxSYNCPER は CMPSS に接続されています。機能は、HRPCTL レジスタを使って設定されますが、HRPWM とは関係ありません。CMPSS が EPWMxSYNCPER を使用方法の詳細については、それぞれの章を参照してください。

- **ADC 変換開始信号 (EPWMxSOCA、EPWMxSOCB)**

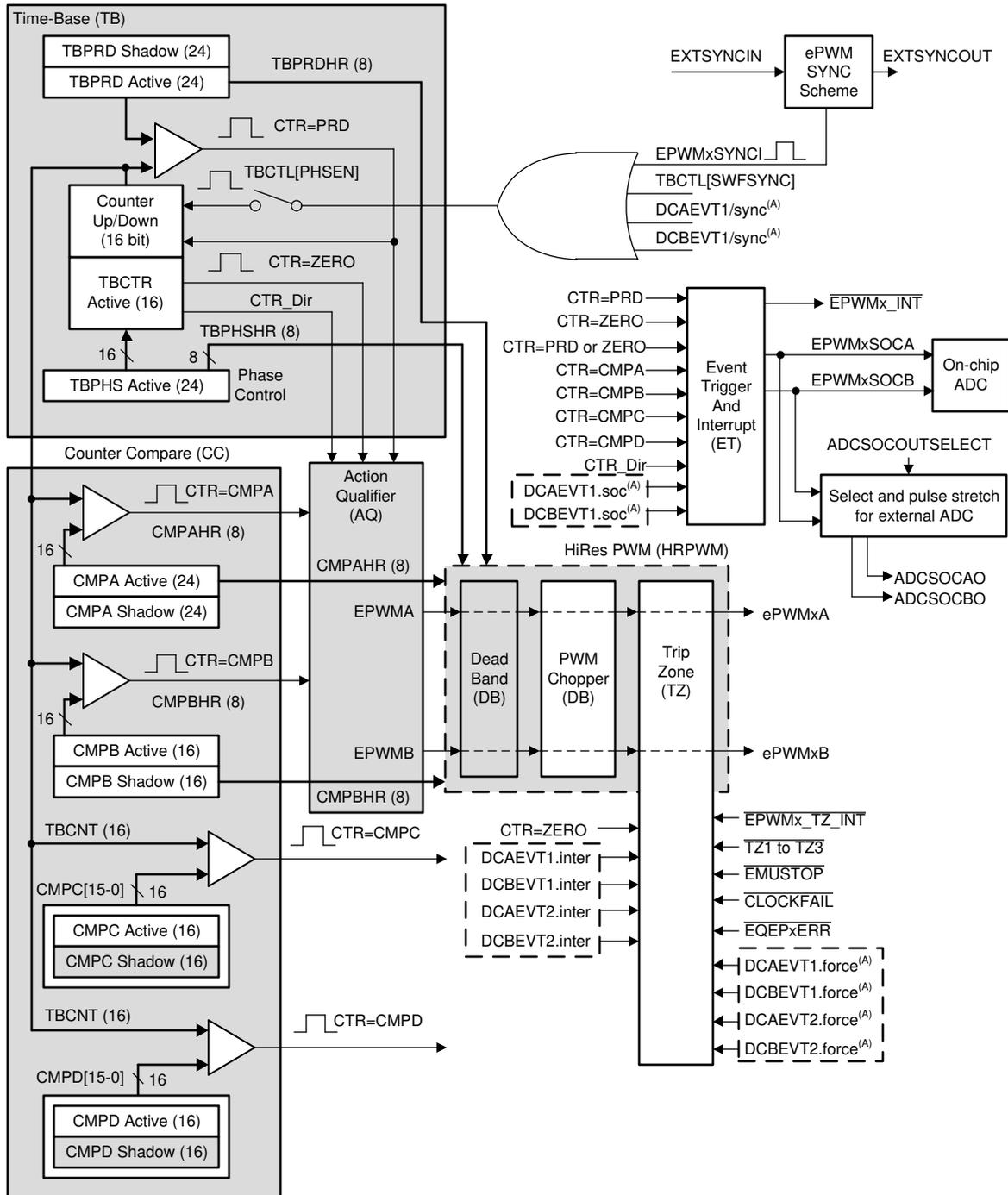
各 ePWM モジュールは 2 つの ADC 変換開始信号を持っています。すべての ePWM モジュールは変換開始をトリガできます。変換開始をトリガするイベントは、いずれも ePWM のイベント・トリガ・サブモジュールで設定されます。

- **コンパレータ出力信号 (COMPxOUT)**

コンパレータ・モジュールからの出力信号は、入力クロスバーと EPWM クロスバーを経由して、12 のトリップ入力 [TRIPIN1 ~ TRIPIN12] のいずれかまたはすべてに入力でき、トリップ・ゾーン信号と組み合わせてデジタル比較イベントを生成できます。

- **ペリフェラル・バス**

ペリフェラル・バスは 32 ビット幅であるため、ePWM レジスタ・ファイルへの 16 ビットと 32 ビットの両方の書き込みが可能です。



A. これらのイベントは、TRIPIN 入力のレベルに基づいて、ePWM デジタル比較 (DC) サブモジュールによって生成されます。

図 1-3. ePWM モジュールと重要な内部信号の相互接続

1.2 デバイス・ピンの設定

デバイス入力ピンをモジュールに接続するには、入カクロスバーと EPWM クロスバーを使用する必要があります。外部信号が必要とされる可能性がある場合の例として、TZx、TRIPx、EXTSYNCIN があります。本デバイス上の任意の GPIO を入力として設定できます。適切な GPxQSEL レジスタ・ビットを 11b に設定することで、GPIO 入力のクオリフィケーションを非同期モードに設定できます。内部プルアップは GPyPUD レジスタで設定できます。GPIO モードを使用するため、GPyINV レジスタは信号を反転できます。また、一部の TRIPx (TRIP6 を除く TRIP4 ~ 12) 信号は、入カクロスバーに加えて ePWM クロスバーを経由して配線する必要があります。

このペリフェラルのために GPIO マルチプレクサ・レジスタを設定する必要があります。ピンのグリッチを回避するため、(対応する GPyMUX ビットをデフォルトのゼロに保持しながら) GPyGMUX ビットを最初に設定し、次に GPyMUX レジスタを目的の値に書き込む必要があります。

GPIO マルチプレクサ、GPIO の設定、クロスバーの設定の詳細については、「[汎用入出力 \(GPIO\)](#)」の章を参照してください。

1.3 ePWM モジュールの概要

各 ePWM ペリフェラルには 8 つのサブモジュールが内蔵されています。これらのサブモジュールはそれぞれ、ソフトウェアで設定できる特定のタスクを実行します。

表 1-1 に、主要なサブモジュールと、それらの主要な設定パラメータのリストを示します。たとえば、PWM 波形のデューティ・サイクルを調整または制御する必要がある場合、関連する詳細については、[セクション 1.5](#) のカウンタ比較サブモジュールを参照してください。

表 1-1. サブモジュールの設定パラメータ

サブモジュール	設定パラメータまたはオプション
時間ベース (TB)	<ul style="list-style-type: none"> ePWM クロック (EPWMCLK) を分周して時間ベース・クロック (TBCLK) を生成します。 PWM 時間ベース・カウンタ (TBCTR) の周波数または周期を設定します。 時間ベース・カウンタのモードを設定します。 <ul style="list-style-type: none"> カウント・アップ・モード：非対称型 PWM 用 カウント・ダウン・モード：非対称型 PWM 用 カウント・アップ・ダウン・モード：対称型 PWM 用 別の ePWM モジュールに対する時間ベース位相を設定します。 ハードウェアまたはソフトウェアを使用して、モジュール間で時間ベース・カウンタを同期させます。 同期イベント後の時間ベース・カウンタの方向 (アップまたはダウン) を設定します。 EPWMXLINK の設定に対応するすべての PWM で、TBPRD レジスタに同時に書き込みます。 エミュレータがデバイスを停止させた際の時間ベース・カウンタの挙動を設定します。 ePWM モジュールの同期出力のソースを指定します。 このモジュールのレジスタのワンショットおよびグローバル・ロードを設定します。
カウンタ比較 (CC)	<ul style="list-style-type: none"> 出力 EPWMxA と出力 EPWMxB の PWM デューティ・サイクルを指定します。 EPWMxA または EPWMxB 出力でスイッチング・イベントが発生する時点を指定します。 追加のコンパレータを使用して、割り込みおよび変換開始生成のプログラマブル遅延を指定します。 EPWMXLINK の設定に対応するすべての PWM で、CMPA、CMPB、CMPC、CMPD レジスタに同時に書き込みます。 このモジュールのレジスタのワンショットおよびグローバル・ロードを設定します。

表 1-1. サブモジュールの設定パラメータ (続き)

サブモジュール	設定パラメータまたはオプション
アクション・フィルタ (AQ)	<ul style="list-style-type: none"> 時間ベース・カウンタ比較、トリップ・ゾーン・サブモジュール、コンパレータ・イベントが発生したときに実行されるアクションの種類を指定します。 <ul style="list-style-type: none"> アクションは何も実行されません。 出力 EPWMxA および EPWMxB を High に切り替えます。 出力 EPWMxA および EPWMxB を Low に切り替えます。 出力 EPWMxA および EPWMxB をトグルします。 ソフトウェア制御によって PWM 出力状態を強制します。 ソフトウェアによって PWM デッドバンドを設定および制御します。 このモジュールのレジスタのワンショットおよびグローバル・ロードを設定します。
デッドバンド・ジェネレータ (DB)	<ul style="list-style-type: none"> 上側スイッチと下側スイッチの間の、従来型の相補型デッドバンドの関係を制御します。 出力の立ち上がりエッジ遅延値を指定します。 出力の立ち下がりエッジ遅延値を指定します。 デッドバンド・モジュールを完全にバイパスします。この場合、PWM 波形は変更されずに通過します。 分解能を 2 倍にするための半周期クロックを有効化できます。 ePWMxA 出力に対して ePWMxB の位相をシフトできます。 このモジュールのレジスタのワンショットおよびグローバル・ロードを設定します。
PWM チョップパ (PC)	<ul style="list-style-type: none"> チョッピング (キャリア) 周波数を生成します。 チョッピングされたパルス列の最初のパルスのパルス幅。 2 つ目以降のパルスのデューティ・サイクル。 PWM チョップパ・モジュールを完全にバイパスします。この場合、PWM 波形は変更されずに通過します。
トリップ・ゾーン (TZ)	<ul style="list-style-type: none"> トリップ・ゾーン信号またはデジタル比較イベントの 1 つまたはすべてにตอบสนองするように、またはいずれにもตอบสนองしないように ePWM モジュールを設定します。 フォルトが発生したときに実行されるアクションを指定します。 <ul style="list-style-type: none"> EPWMxA と EPWMxB を強制的に High にします。 EPWMxA と EPWMxB を強制的に Low にします。 EPWMxA と EPWMxB を強制的にハイ・インピーダンス状態にします。 トリップ条件を無視するように EPWMxA と EPWMxB を設定します。 ePWM が各トリップ・ゾーン信号にตอบสนองする頻度を設定します。 <ul style="list-style-type: none"> ワンショット サイクル・バイ・サイクル トリップ・ゾーンが割り込みを開始できるようにします。 トリップ・ゾーン・モジュールを完全にバイパスします。 サイクルごとにトリップをクリアするためのプログラマブル・オプション 必要に応じて、時間ベース・カウンタがカウント・ダウンしている際に実行されるトリップ・アクションを個別に設定します。
イベント・トリガ (ET)	<ul style="list-style-type: none"> 割り込みをトリガする ePWM イベントを有効化します。 ADC 変換開始イベントをトリガする ePWM イベントを有効化します。 イベントがトリガを引き起こす頻度を指定します (すべてのイベントごと、または 2~15 回のイベントに 1 回)。 イベント・フラグのポーリング、セット、クリア
デジタル比較 (DC)	<ul style="list-style-type: none"> コンパレータ (COMP) モジュールの出力と、入力クロスバーを使ってイベントとフィルタ処理済みイベントを生成するように設定されたトリップ・ゾーン信号を有効化します。 TCTR カウンタのキャプチャ、ブランキング・ウィンドウの生成、PWM 出力または時間ベース・カウンタへの (キャプチャされた値に基づく) 遅延の挿入を行うイベント・フィルタリング・オプションを指定します。

1.4 時間ベース (TB) サブモジュール

各 ePWM モジュールは、ePWM モジュールのすべてのイベント・タイミングを決定する専用の時間ベース・サブモジュールを持っています。同期ロジックを内蔵しているため、複数の ePWM モジュールの時間ベースが 1 つのシステムとして連携して動作できます。

図 1-4 に、ePWM 内の時間ベース・サブモジュールを示します。

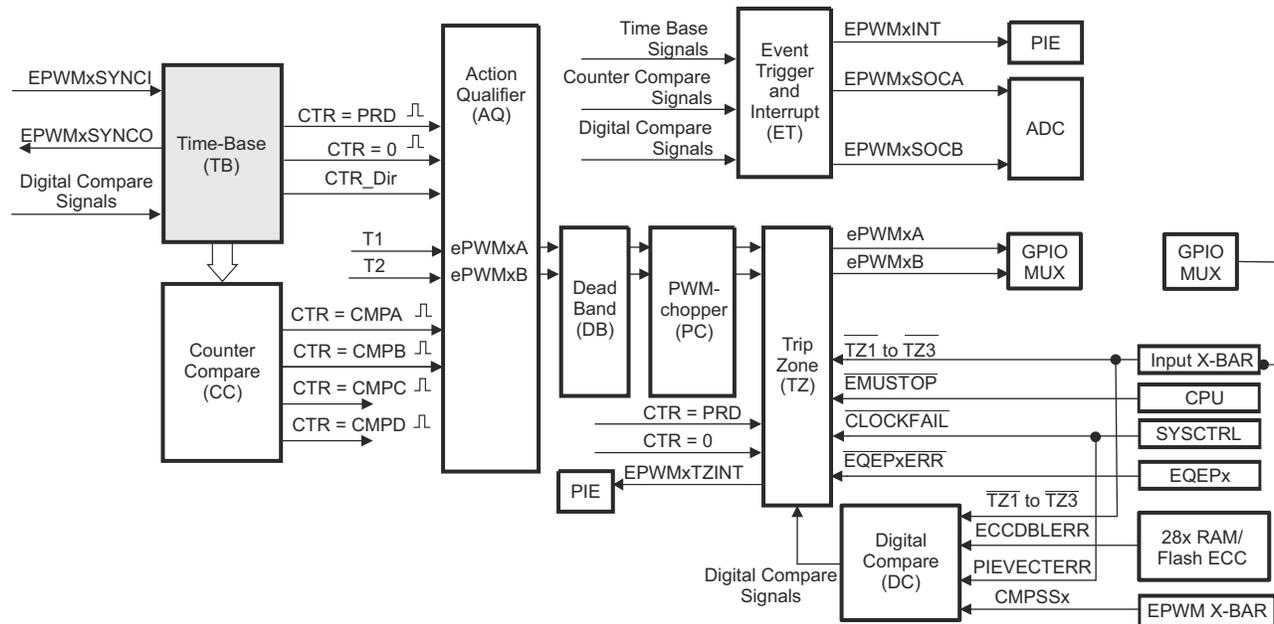


図 1-4. 時間ベース・サブモジュール

1.4.1 時間ベース・サブモジュールの目的

時間ベース・サブモジュールは、次のように設定できます。

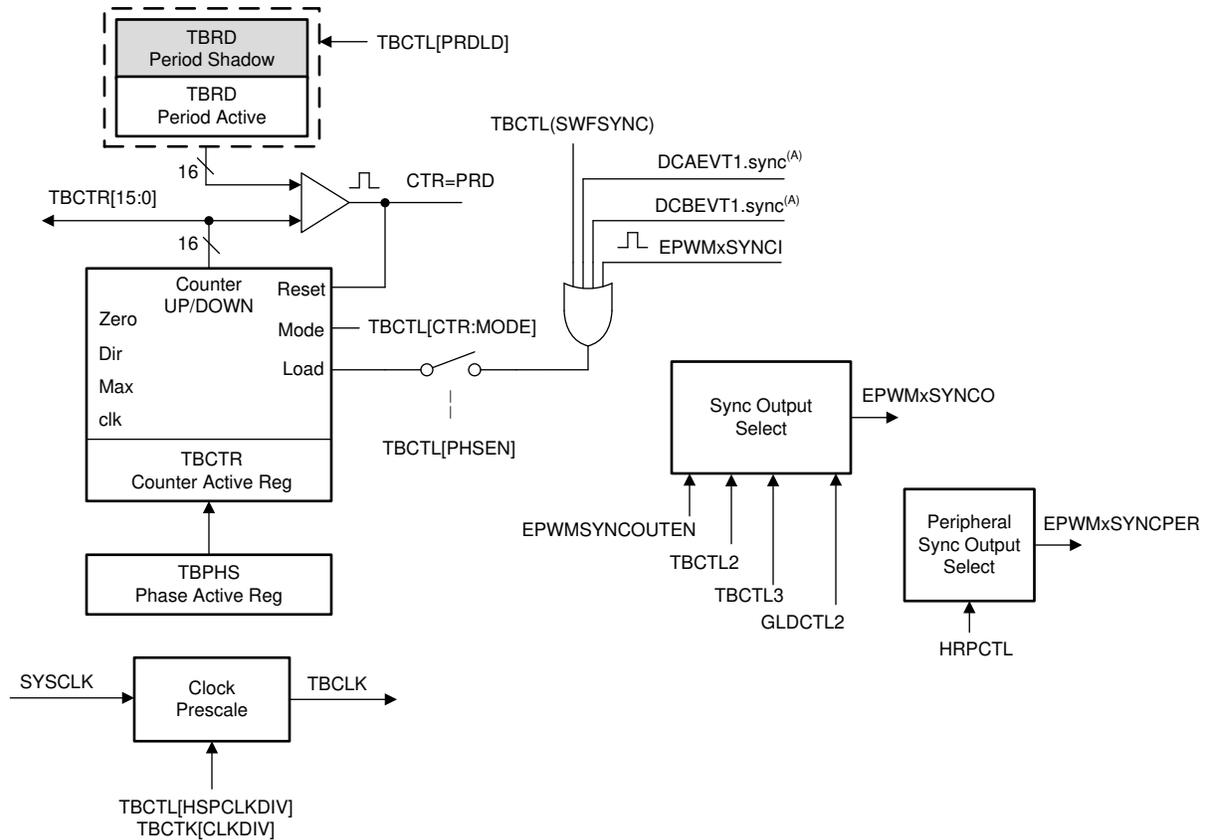
- イベントの発生頻度を制御するため、ePWM 時間ベース・カウンタ (TBCTR) の周波数または周期を設定します。
- その他の ePWM モジュールとの時間ベース同期を管理します。
- その他の ePWM モジュールとの位相関係を維持します。
- 時間ベース・カウンタをカウント・アップ、カウント・ダウン、カウント・アップ・ダウン・モードに設定します。
- 以下のイベントを生成します。
 - CTR = PRD : 時間ベース・カウンタが指定された周期と一致 (TBCTR = TBPRD)
 - CTR = ゼロ : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- 時間ベース・クロック (ePWM クロック (EPWMCLK) を分周したもの) のレートを設定します。これにより、時間ベース・カウンタをより低速でインクリメント / デクリメントできます。

注

TBCTR がカウントしている間、アプリケーション・コードが TBCTR の値をソフトウェアで更新する必要がある場合、時間ベース関連イベントが実行されるには、時間ベース・モジュールが少なくとも 1 TBCLK サイクルを必要とすることに注意することが重要です。したがって、カウンタがカウント・アップしている場合は TBCTR = PRD の代わりに TBCTR = PRD - 1 を TBCTR に書き込み、カウンタがカウント・ダウンしている場合は TBCTR = 0 の代わりに TBCTR = 1 を TBCTR に書き込むことで、これらのイベントは実行されます。

1.4.2 時間ベース・サブモジュールの制御と監視

図 1-5 のブロック図に、時間ベース・サブモジュールの重要な信号とレジスタを示します。表 1-2 に、時間ベース・サブモジュールに関連する主な信号の説明を示します。



A. These signals are generated by the digital compare (DC) submodule.

図 1-5. 時間ベース・サブモジュールの信号とレジスタ

表 1-2. 主な時間ベース信号

信号	説明
EPWMxSYNCl	時間ベース同期入力 時間ベース・カウンタをその他の ePWM モジュールのカウンタと同期させるために使われる入力パルス。同期に使用できるすべての信号の詳細については、EPWMSYNClNSEl を参照してください。特定のデバイスの同期順序の詳細については、 セクション 1.4.3.3 を参照してください。
EPWMxSYNCO	時間ベース同期出力。 この出力パルスは、その他の ePWM モジュールのカウンタを同期させるために使われます。EPWMSYNCOUEN、TBCTL2、TBCTL3、GLDCTL2 を使って、この出力パルスのソースを選択します。
EPWMxSYNCPER	時間ベース・ペリフェラル同期出力。 この出力信号は、CMPSS を ePWM に同期させるために使用されます。この出力信号は、HRPCTL レジスタを使って設定できます。この信号は HRPWM とは関係がないことに注意します。
CTR = PRD	時間ベース・カウンタが指定された周期と一致。 この信号は、カウンタ値がアクティブな周期レジスタ値と一致すると常に生成されます。それは TBCTR = TBPRD のときです。
CTR = ゼロ	時間ベース・カウンタがゼロと一致 この信号は、カウンタ値がゼロになると常に生成されます。それは TBCTR が 0x00 と一致したときです。
CTR = CMPB	時間ベース・カウンタがアクティブなカウンタ比較 B レジスタと一致 (TBCTR = CMPB)。 このイベントは、カウンタ比較サブモジュールによって生成され、同期出力ロジックによって使用されます。
CTR_dir	時間ベース・カウンタの方向。 ePWM の時間ベース・カウンタの現在の方向を示します。カウンタが増加している間、信号は High になり、カウンタが減少している間、信号は Low になります。
CTR_max	時間ベース・カウンタが最大値と一致 (TBCTR = 0xFFFF)。 TBCTR の値が最大値に達したときに生成されるイベントです。この信号はステータス・ビットとしてのみ使用されます。
TBCLK	時間ベース・クロック これは、ePWM クロック (EPWMCLK) の分周クロックであり、ePWM 内のすべてのサブモジュールで使用されます。このクロックによって、時間ベース・カウンタがインクリメントまたはデクリメントするレートが決まります。

1.4.3 PWM 周期および周波数の計算

PWM イベントの周波数は、時間ベース周期 (TBPRD) レジスタと時間ベース・カウンタのモードによって制御されます。図 1-6 に、周期が 4 に設定されている (TBPRD = 4) ときのアップ・カウント、ダウン・カウント、アップ・ダウン・カウントの時間ベース・カウンタ・モードの周期 (T_{pwm}) と周波数 (F_{pwm}) の関係を示します。各ステップの時間の増分は、ePWM クロック (EPWMCLK) を分周した時間ベース・クロック (TBCLK) によって定義されます。

時間ベース・カウンタには、時間ベース制御レジスタ (TBCTL) によって選択される 3 つの動作モードがあります。

- **アップ・ダウン・カウント・モード** : アップ・ダウン・カウント・モードでは、時間ベース・カウンタはゼロから始まり、周期 (TBPRD) 値に達するまでインクリメントします。その周期値に達すると、時間ベース・カウンタは、カウンタがゼロになるまでデクリメントします。ゼロに達すると、カウンタはこのパターンを繰り返してインクリメントを開始します。
- **アップ・カウント・モード** : アップ・カウント・モードでは、時間ベース・カウンタはゼロから始まり、カウンタが周期レジスタの値 (TBPRD) に達するまでインクリメントします。その周期値に達すると、時間ベース・カウンタはゼロにリセットされ、再びインクリメントを開始します。
- **ダウン・カウント・モード** : ダウン・カウント・モードでは、時間ベース・カウンタは周期 (TBPRD) 値から始まり、カウンタがゼロに達するまでデクリメントします。カウンタがゼロに達すると、時間ベース・カウンタは周期値にリセットされ、再びデクリメントを開始します。

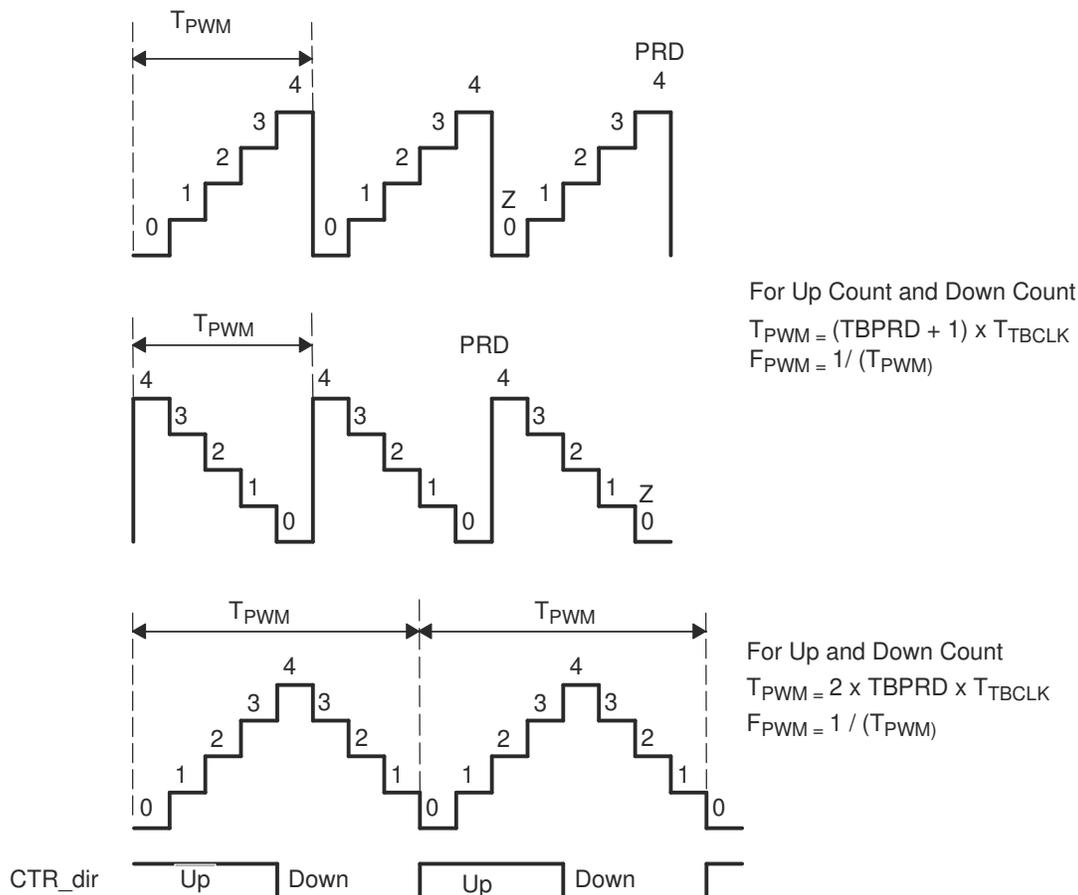


図 1-6. 時間ベースの周波数と周期

1.4.3.1 時間ベース周期シャドウ・レジスタ

時間ベース周期レジスタ (TBPRD) にはシャドウ・レジスタがあります。シャドウイングによって、レジスタの更新をハードウェアと同期させることができます。以下の定義は、ePWM モジュールのすべてのシャドウ・レジスタを説明するために使用されます。

- **アクティブ・レジスタ**：アクティブ・レジスタはハードウェアを制御し、ハードウェアが引き起こす (起動する) アクションを制御します。
- **シャドウ・レジスタ**：シャドウ・レジスタ・バッファは、一時的な保管場所をアクティブ・レジスタに提供するためのものであり、制御ハードウェアには直接的な影響を及ぼしません。計画された適切な時点で、シャドウ・レジスタの内容はアクティブ・レジスタに転送されます。これにより、ソフトウェアによってレジスタが非同期的に変更されることによるデータ破損または誤動作を防止できます。

シャドウ周期レジスタのメモリ・アドレスは、アクティブ・レジスタと同じです。どちらのレジスタに対して書き込みまたは読み出しが行われるかは、TBCTL[PRDL] ビットによって決定されます。このビットは、TBPRD シャドウ・レジスタを次のように有効化 / 無効化します。

- **時間ベース周期シャドウ・モード**：TBCTL[PRDL] = 0 になると、TBPRD シャドウ・レジスタは有効化されます。TBPRD メモリ・アドレスに対する読み出しと書き込みは、シャドウ・レジスタに対して行われます。時間ベース・カウンタがゼロになると (TBCTR = 0x00)、または TBCTL2[PRDLDSYNC] ビットで決定される同期イベントが発生すると、シャドウ・レジスタの内容がアクティブ・レジスタに転送されます (TBPRD (アクティブ) ← TBPRD (シャドウ))。TBCTL[PRDL] = 0 の場合にのみ、PRDLDSYNC ビットは有効です。デフォルトでは、TBPRD シャドウ・レジスタは有効化されています。同期入力の信号源については、[セクション 1.4.3.3](#) で説明します。

グローバル・ロード制御機能は、グローバル・ロード構成レジスタ (GLDCFG) の適切なビットを設定することで、時間ベース周期レジスタと組み合わせて使用することもできます。グローバル・ロード・モードが選択されている場合、このモードが有効化されたすべてのレジスタにおいて、シャドウ・レジスタからアクティブ・レジスタへの内容の転送は、グローバル・シャドウ・アクティブ・ロード制御レジスタ (GLDCTL) の構成ビットで設定されたイベントと同じイベントで行われます。グローバル・ロード制御機能については、[セクション 1.4.7](#) を参照してください。

- **時間ベース周期即時ロード・モード**：即時ロード・モードが選択されている場合 (TBCTL[PRDL] = 1)、TBPRD メモリ・アドレスに対する読み出しまたは書き込みは、アクティブ・レジスタに対して直接行われます。

1.4.3.2 時間ベース・クロック同期

ペリフェラル・クロック・イネーブル・レジスタの TBCLKSYNC ビットを使用すると、有効化されているすべての ePWM モジュールを、すべてのユーザーが時間ベース・クロック (TBCLK) にグローバルに同期させることができます。このビットをセットすると、有効化されているすべての ePWM モジュール・クロックは、TBCLK の最初の立ち上がりエッジを揃えて開始します。TBCLK を完全に同期させるには、各 ePWM モジュールのプリスケアラをまったく同じに設定する必要があります。

ePWM クロックを有効化するための正しい手順は、次のとおりです。

1. PCLKCRx レジスタで ePWM モジュールのクロックを有効化します。
2. TBCLKSYNC = 0 に設定します。
3. ePWM モジュールを設定します。
4. TBCLKSYNC = 1 に設定します。

1.4.3.3 時間ベース・カウンタ同期

ePWM 同期方式を使うと、ePWM モジュールを柔軟に同期させることができます。各 ePWM モジュールには、同期入力 (SYNCl)、同期出力 (SYNCO)、ペリフェラル同期出力 (SYNCPER) が備わっています。図 1-7 で、EXTSYNClN1 には INPUTXBAR5 から信号が供給され、EXTSYNClN2 には INPUTXBAR6 から信号が供給されています。このクロスバーは、同期入力として任意の GPIO を選択するように構成できます。INPUTXBAR5 と INPUTXBAR6 を含むすべての同期入力の一覧表については、表 1-3 を参照してください。図 1-8 に、EXTSYNCOuT として使用できる信号源を示します。

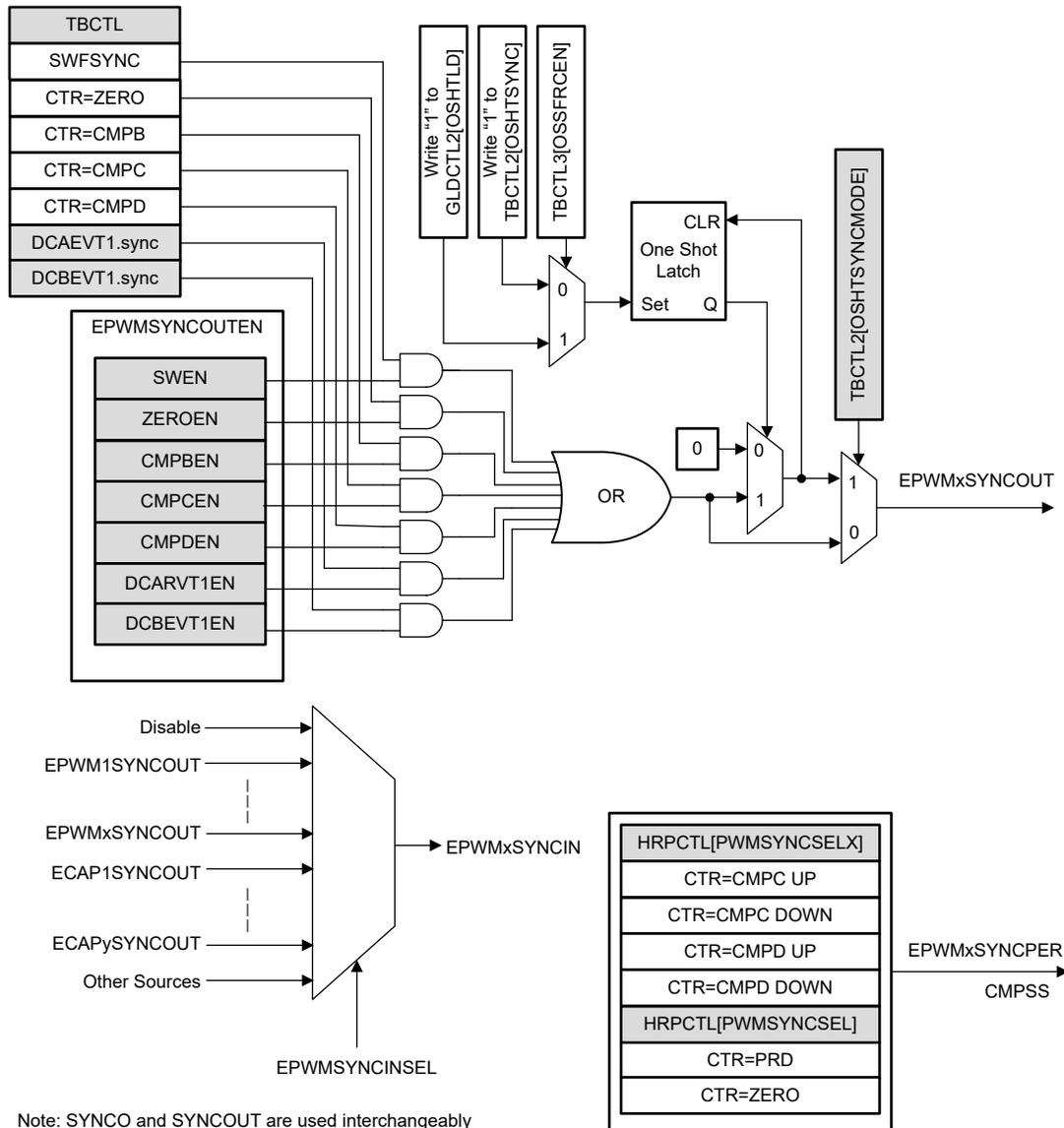


図 1-7. 時間ベース・カウンタ同期方式

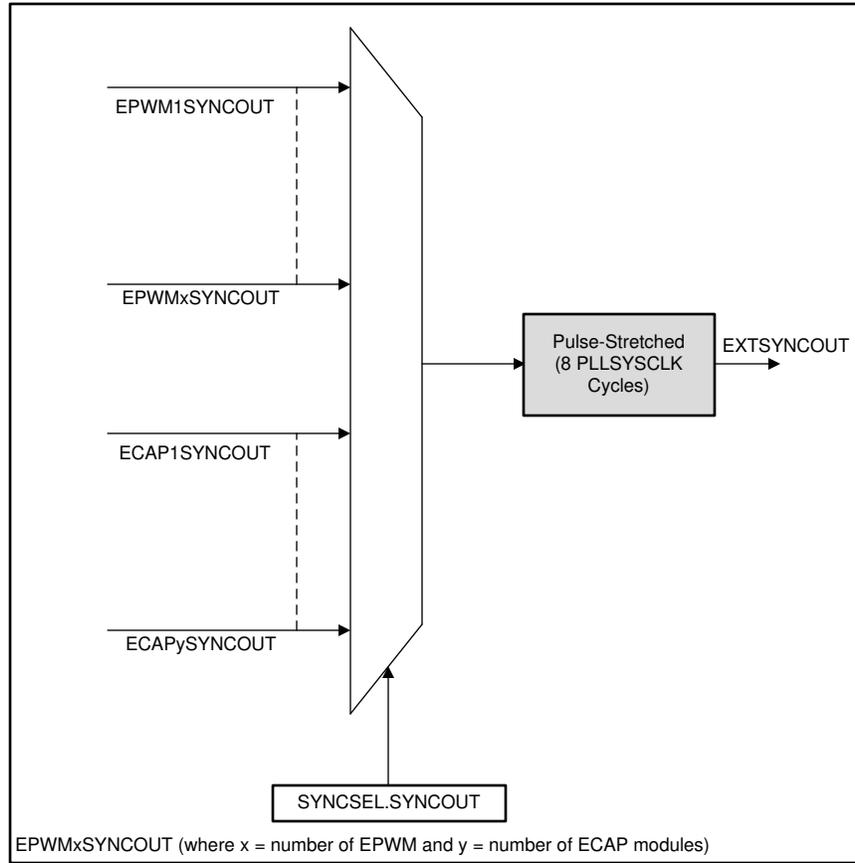


図 1-8. ePWM 外部同期出力

注

ご使用のデバイスで利用可能な ePWM および eCAP モジュールの数については、データシートを参照してください。

各 ePWM モジュールは、同期入力を使用するように、または無視するように設定できます。TBCTL[PHSEN] ビットがセットされている場合、次の条件のいずれかが発生すると、ePWM モジュールの時間ベース・カウンタ (TBCTR) に位相レジスタ (TBPHS) の内容が自動的にロードされます。

- **EPWMxSYNCl** : 同期入力パルス : 入力同期パルスが検出されると、位相レジスタの値がカウンタ・レジスタにロードされます (TBPHS → TBCTR)。この動作は、次の有効な時間ベース・クロック (TBCLK) のエッジで行われます。
- **ソフトウェア強制同期パルス** : TBCTL[SWFSYNCl] 制御ビットに 1 を書き込むと、ソフトウェア強制同期が実行されます。このパルスは、同期入力信号と論理和がとられるため、EPWMxSYNCl のパルスと同じ効果を及ぼします。
- **デジタル比較イベント同期パルス** : DCAEVT1 および DCBEVT1 デジタル比較イベントは、EPWMxSYNCl と同じ効果を及ぼす同期パルスを生成するように設定できます。

注

EPWMxSYNCl 信号が High に保持されている場合、同期は連続的には行われません。EPWMxSYNCl は立ち上がりエッジによって活性化されます。

この機能により、ePWM モジュールを別の ePWM モジュールの時間ベースに自動的に同期させることができます。異なる ePWM モジュールで生成される波形に、進みまたは遅れ位相制御を追加することで、それらを同期させることができます。アップ・ダウン・カウント・モードでは、同期イベントの直後に、TBCTL[PHSDIR] ビットが時間ベース・カウンタの方向を設定します。新しい方向は、同期イベントの前の方向とは無関係です。カウント・アップまたはカウント・ダウン・モードでは、PHSDIR ビットは無視されます。図 1-9 ~ 図 1-12 に、例を示します。

TBCTL[PHSEN] ビットをクリアすると、同期入力パルスを無視するように ePWM が設定されます。

1.4.3.4 ePWM 同期選択

表 1-3 に、ePWM 同期入力および出力のソースを示します。

表 1-3. ePWM 同期選択

EPWMSYNClSEL.SEL、ECAPSYNClSEL.SEL	同期ソース
0x0	予約済み
0x1	EPWM1.SYNClOUT
0x2	EPWM2.SYNClOUT
0x3	EPWM3.SYNClOUT
0x4	EPWM4.SYNClOUT
0x5	EPWM5.SYNClOUT
0x6	EPWM6.SYNClOUT
0x7	EPWM7.SYNClOUT
0x8	予約済み
0x9	予約済み
0xA	予約済み
0xB	予約済み
0xC	予約済み
0xD	予約済み
0xE	予約済み
0xF	予約済み
0x10	予約済み
0x11	ECAP1.SYNClOUT
0x12	ECAP2.SYNClOUT
0x13	予約済み
0x14	予約済み

表 1-3. ePWM 同期選択 (続き)

EPWMSYNCINSEL.SEL、ECAPSYNCINSEL.SEL	同期ソース
0x15	予約済み
0x16	予約済み
0x17	予約済み
0x18	INPUT-XBAR.INPUT5
0x19	INPUT-XBAR.INPUT6
0x1A	予約済み
0x1B	予約済み
0x1C	予約済み
0x1D	予約済み
0x1E	予約済み
0x1F	予約済み

1.4.4 複数の ePWM モジュールの時間ベース・クロックの位相ロック

TBCLKSYNC ビットを使うと、デバイス上のすべての有効化された ePWM モジュールの時間ベース・クロックをグローバルに同期させることができます。このビットは本デバイスのクロック・イネーブル・レジスタの一部であり、このマニュアルの「システムの制御と割り込み」セクションに記載されています。TBCLKSYNC = 0 になると、すべての ePWM モジュールの時間ベース・クロックは停止します (デフォルト)。TBCLKSYNC = 1 になると、対応するすべての ePWM 時間ベース・クロックは、TBCLK の立ち上がりエッジを揃えて開始します。TBCLK を完全に同期させるには、各 ePWM モジュールの TBCTL レジスタのプリスケアラ・ビットをまったく同じに設定する必要があります。ePWM クロックを有効化するための適切な手順は次のとおりです。

1. 個別の ePWM モジュール・クロックを有効化します。これについては、「システムの制御と割り込み」の章に記載されています。
2. TBCLKSYNC = 0 に設定します。これにより、すべての有効化された ePWM モジュール内の時間ベース・クロックが停止します。
3. プリスケアラ値と目的の ePWM モードを設定します。
4. TBCLKSYNC = 1 に設定します。

1.4.5 ePWM モジュール間の TBPRD および CMPx レジスタへの同時書き込み

可変周波数アプリケーションの場合、ePWM モジュール間の TBPRD レジスタと CMPx レジスタを同時に書き込む必要があります。これにより、ePWM モジュール間ですべてのレジスタが更新される前に、CTR = 0 または CTR = PRD パルスによってこれらのレジスタのシャドウ - アクティブ・ロードが強制的に行われるという状況 (その結果、一部のレジスタは新しいシャドウ値からロードされ、その他のレジスタは古いシャドウ値からロードされます) を防止できます。これに対応するため、TPRD : TBPRDHR、CMPA : CMPAHR、CMPB : CMPBHR、CMPC、CMPD レジスタの PWM モジュール間の ePWM レジスタ・リンク機能が追加されました。

特定の ePWM モジュール # A では、ユーザー・コードによって、EPWMXLINK のリンクされたレジスタ・ビット・フィールドに B+1 が書き込まれます。「B」はリンク先の ePWM モジュール番号です (つまり、ePWM モジュール「B」の TBPRD : TBPRDHR、CMPA : CMPAHR、CMPB : CMPBHR、CMPC への同時書き込みは、ePWM モジュール「A」の対応するレジスタに書き込まれます)。たとえば、CMPA : CMPAHR が ePWM1 にリンクされるように、ePWM3 の EPWMXLINK レジスタが設定されている場合、ePWM1 の CMPA : CMPAHR に書き込むと、ePWM3 の CMPAHR にも同時に同じ値が書き込まれます。ePWM4 にも、ePWM1 にリンクされた CMPA : CMPAHR レジスタが存在する場合、ePWM1 に書き込むと、ePWM3 と ePWM4 の両方の CMPA : CMPAHR レジスタに同じ値が書き込まれます。

EPWMXLINK のレジスタの説明には、対応する ePWM のリンクされたレジスタ・ビット・フィールド値が明確に説明されています。EPWMXLINK の使い方の例として、ePWM2 の CMPA を ePWM1 の CMPA にリンクすることが挙げられます。この場合、ePWM1 の CMPA に書き込むと、ePWM2 の CMPA の値も変更されます。

1.4.6 時間ベース・カウンタ・モードとタイミング波形

時間ベース・カウンタは、次の4つのモードのいずれかで動作します。

- 非対称アップ・カウント・モード
- 非対称ダウン・カウント・モード
- 対称アップ・ダウン・カウント
- 固定 (時間ベース・カウンタは現在の値で一定に保持されます。)

最初の3つのモードの動作を示すため、イベントが生成されるタイミングと、EPWMxSYNCl 信号に対する時間ベースの応答方法を、以下のタイミング図に示します。

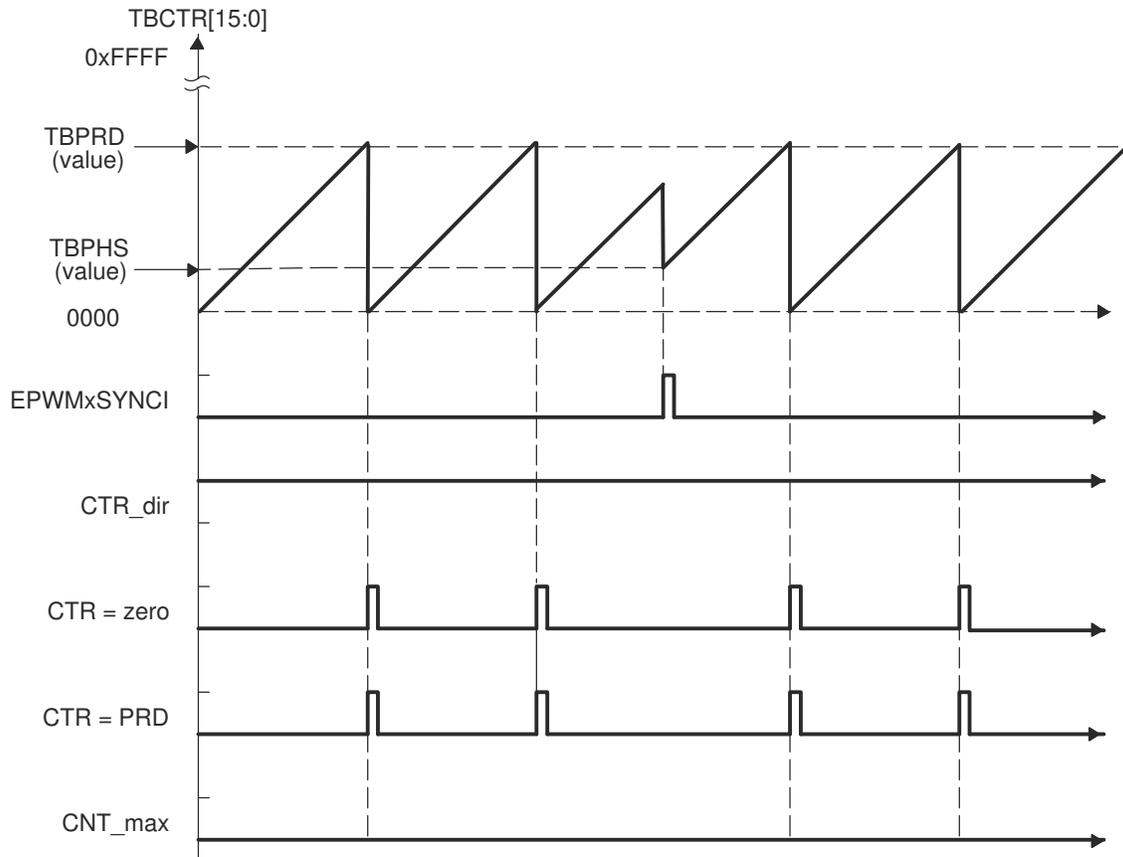


図 1-9. 時間ベースのアップ・カウント・モード波形

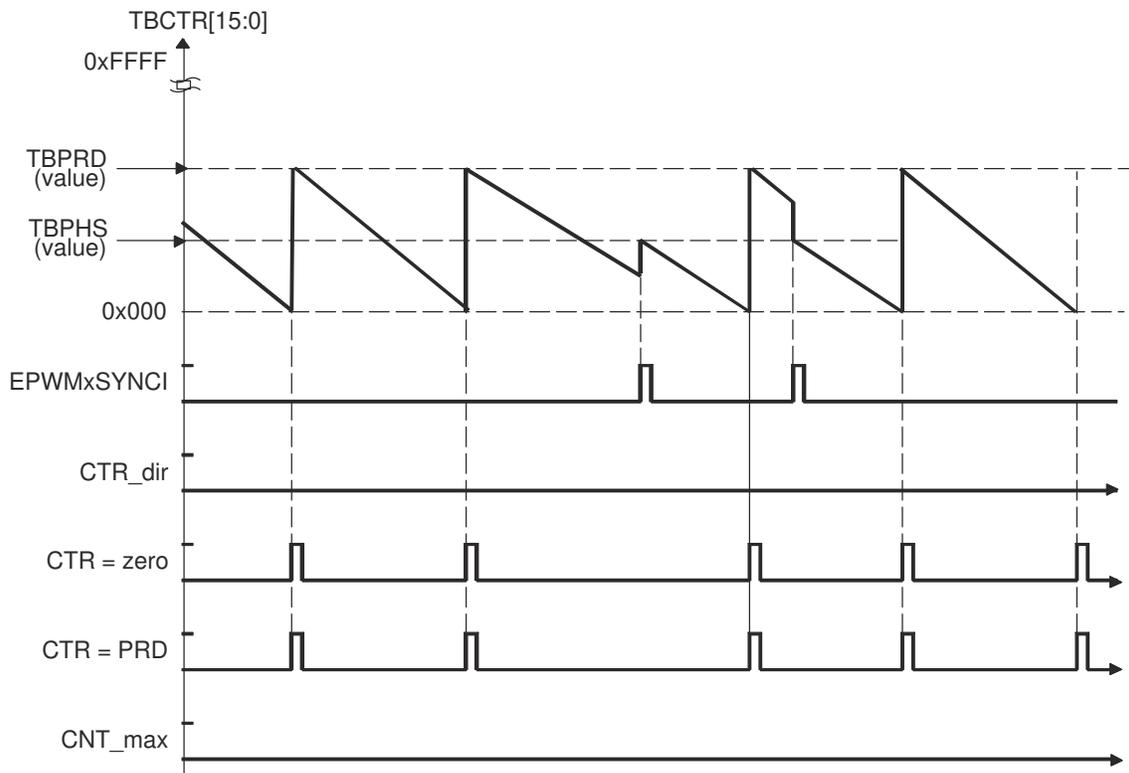


図 1-10. 時間ベースのダウン・カウント・モード波形

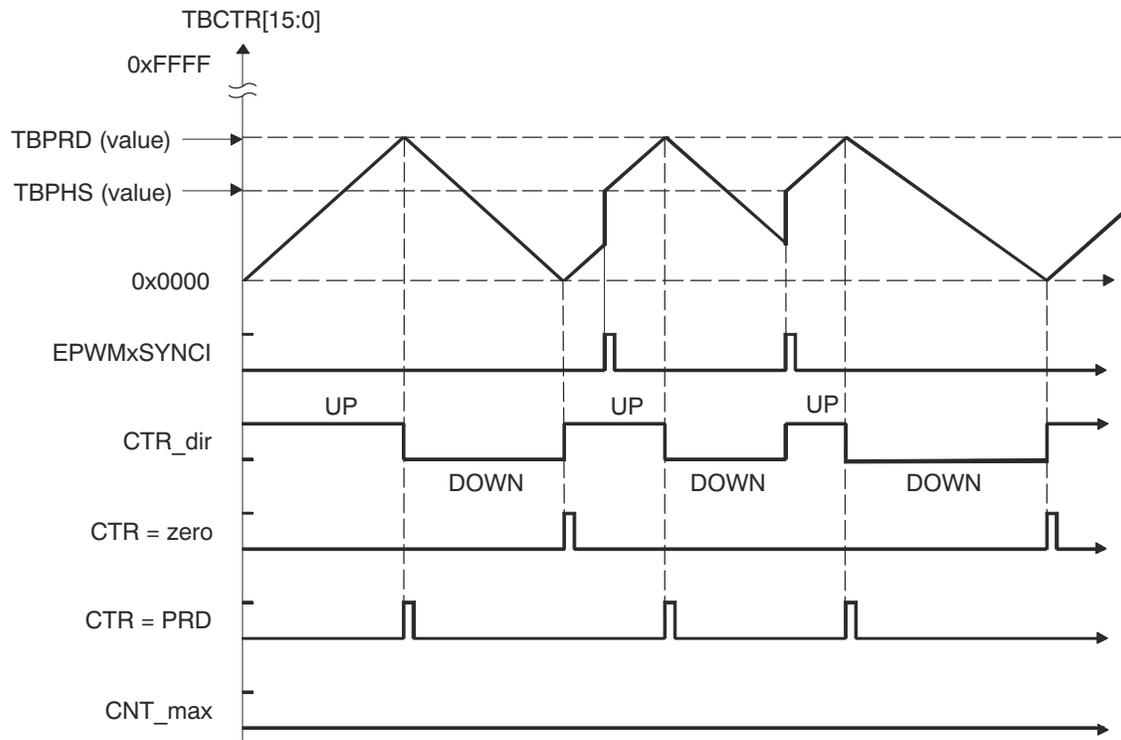


図 1-12. 時間ベースのアップ・ダウン・カウント波形、TBCTL[PHSDIR = 1] により同期イベント後カウント・アップ

1.4.7 グローバル・ロード

図 1-13 に、グローバル・ロード機能に関連する信号とレジスタを示します。

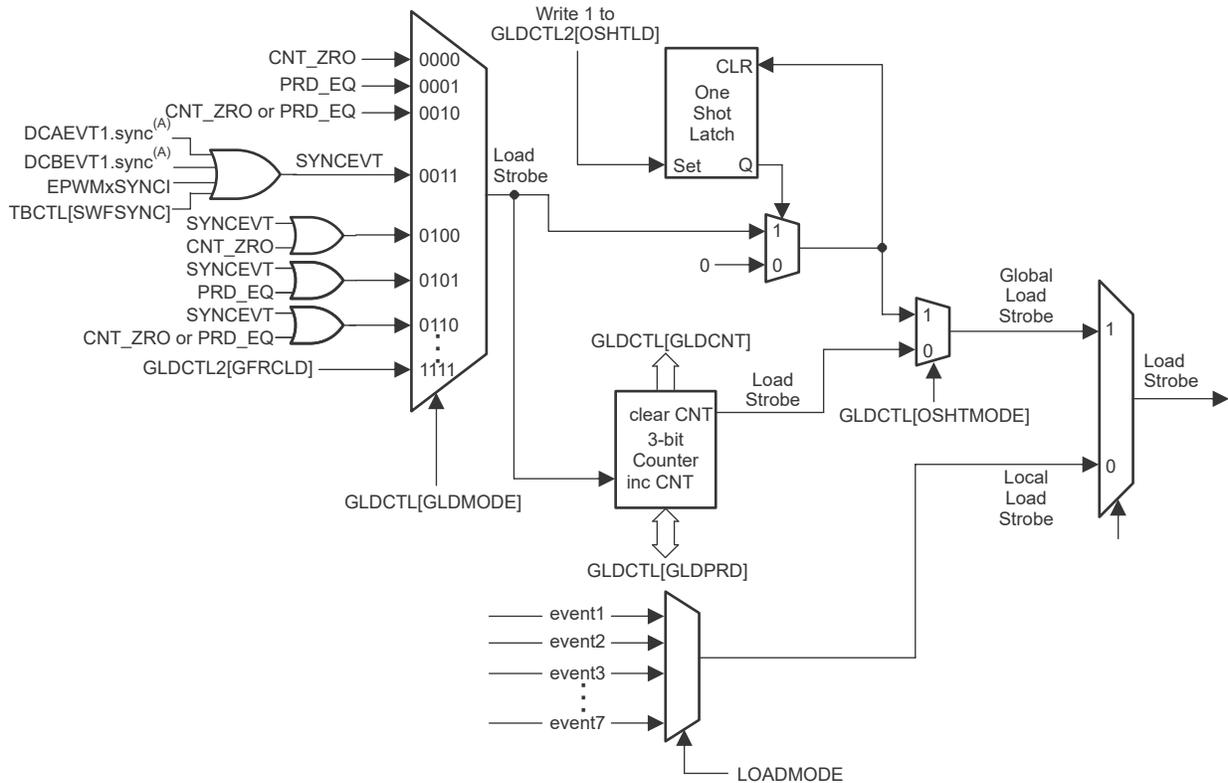


図 1-13. グローバル・ロード : 信号とレジスタ

注

SYNCEVT 信号は、PHSEN がセットされている場合にのみ伝搬されます。

この機能が有効化されている場合、このモードが有効化されたすべてのレジスタにおいて、シャドウ・レジスタからアクティブ・レジスタへの内容の転送は、グローバル・シャドウ・アクティブ・ロード制御レジスタの構成ビット (GLDCTL[GLDMODE]) で設定されたイベントと同じイベントで行われます。GLDCTL[GLD] = 1 の場合、個別のシャドウ・レジスタのシャドウ・アクティブ・ロード・イベント選択ビットは無視され、GLDCFG[REGx] によって有効化された対応するレジスタに対してグローバル・ロード・モードが効力を生じません。

GLDCTL[GLD] = 1 かつ GLDCFG[REGx] = 0 の場合、グローバル・ロード・モードは対応するレジスタ (REGx) に影響を及ぼしません。個別のシャドウ・レジスタのシャドウ・アクティブ・ロード・イベント選択ビットは、シャドウ・レジスタからアクティブ・レジスタへの内容の転送方法を決定します。

1.4.7.1 グローバル・ロード・パルス・プリスケラ

この機能を使用すると、選択されたグローバル・ロード・パルス (GLDCTL[GLDMODE]) が「N」回発生するとシャドウ・アクティブ転送が 1 回行われるように選択できます。このプリスケール機能は、グローバル・ロード機能を使用できないレジスタまたは使用するように設定されていないレジスタ (つまり、GLDCTL[GLD] = 「0」または GLDCFG[REGx] = 「0」) では利用できません。

1.4.7.2 ワンショット・ロード・モード

この機能を使うと、ユーザーはシャドウ・レジスタからアクティブ・レジスタへの転送を 1 回実行させることができます。GLDCTL2[OSHTLD] = 「1」の場合、シャドウ - アクティブ・レジスタ転送は、(グローバル・ロード機能を使うように設定されたレジスタの場合、) GLDCTL[GLDMODE] によって選択されたイベント時に実行されます。

GLDCTL2[GFRCLD] を使うことで、シャドウ・レジスタからアクティブ・レジスタに内容をソフトウェアによって強制的にロードできます。EPWMXLINK[GLDCTL2LINK] を使うことで、GLDCTL2 レジスタを複数の PWM モジュールに関連付けることもできます。これにより、上述のワンショット・ロード・モード機能に加えて、特定の PWM イベント時に (または、必要に応じて同じクロック・サイクル内で) 1 つ以上の PWM モジュール内の複数の PWM レジスタを適切に更新できます。これは、可変周波数アプリケーションと多相インターリーブ・アプリケーションで非常に有用です。

注

高分解能モードが有効化されている場合、ワンショット・ロード・モードは**使えません**。

1.4.7.3 ワンショット同期モード

ワンショット同期モードを有効化して SYNCOUT パルスを生成するには、[図 1-14](#) に示すように、TBCTL2[OSHTSYNCMODE] ビットをセットし、TBCTL2[OSHTSYNC] ビットをセットします。

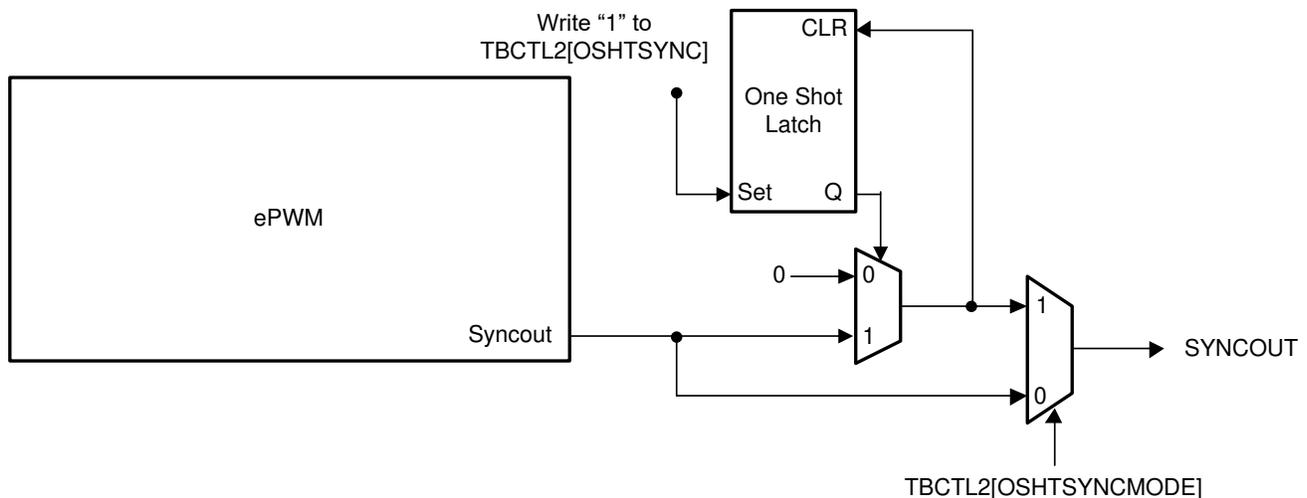


図 1-14. ワンショット同期モード

1.5 カウンタ比較 (CC) サブモジュール

図 1-15 に、ePWM 内のカウンタ比較サブモジュールを示します。

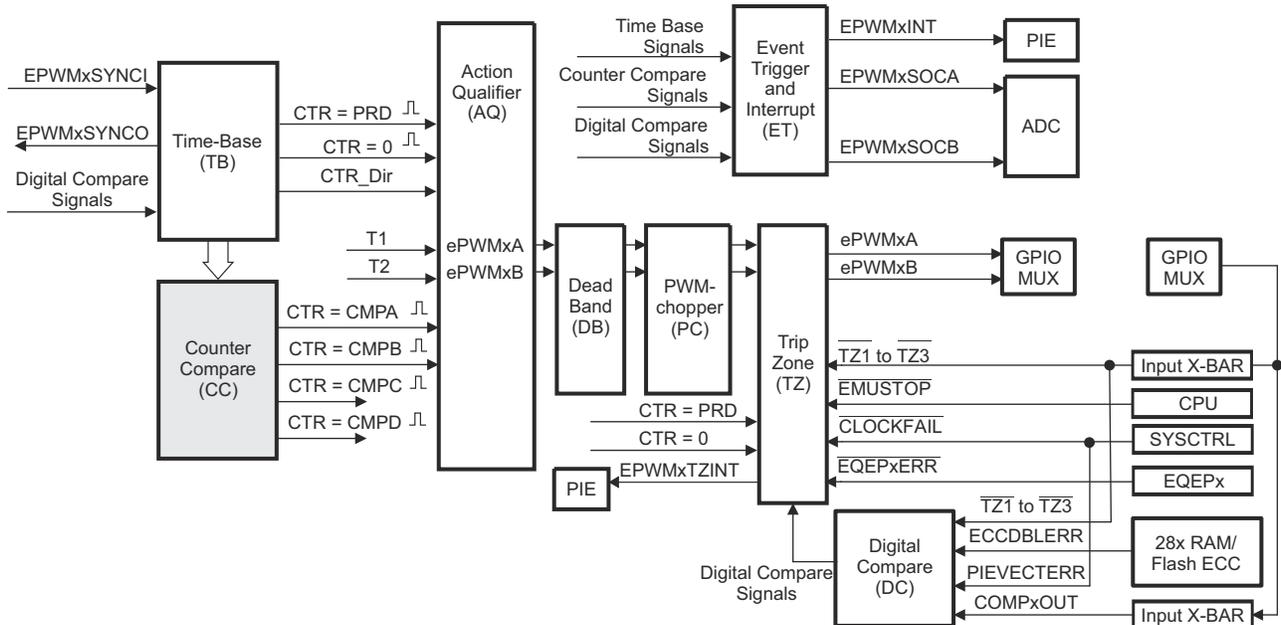


図 1-15. カウンタ比較サブモジュール

1.5.1 カウンタ比較サブモジュールの目的

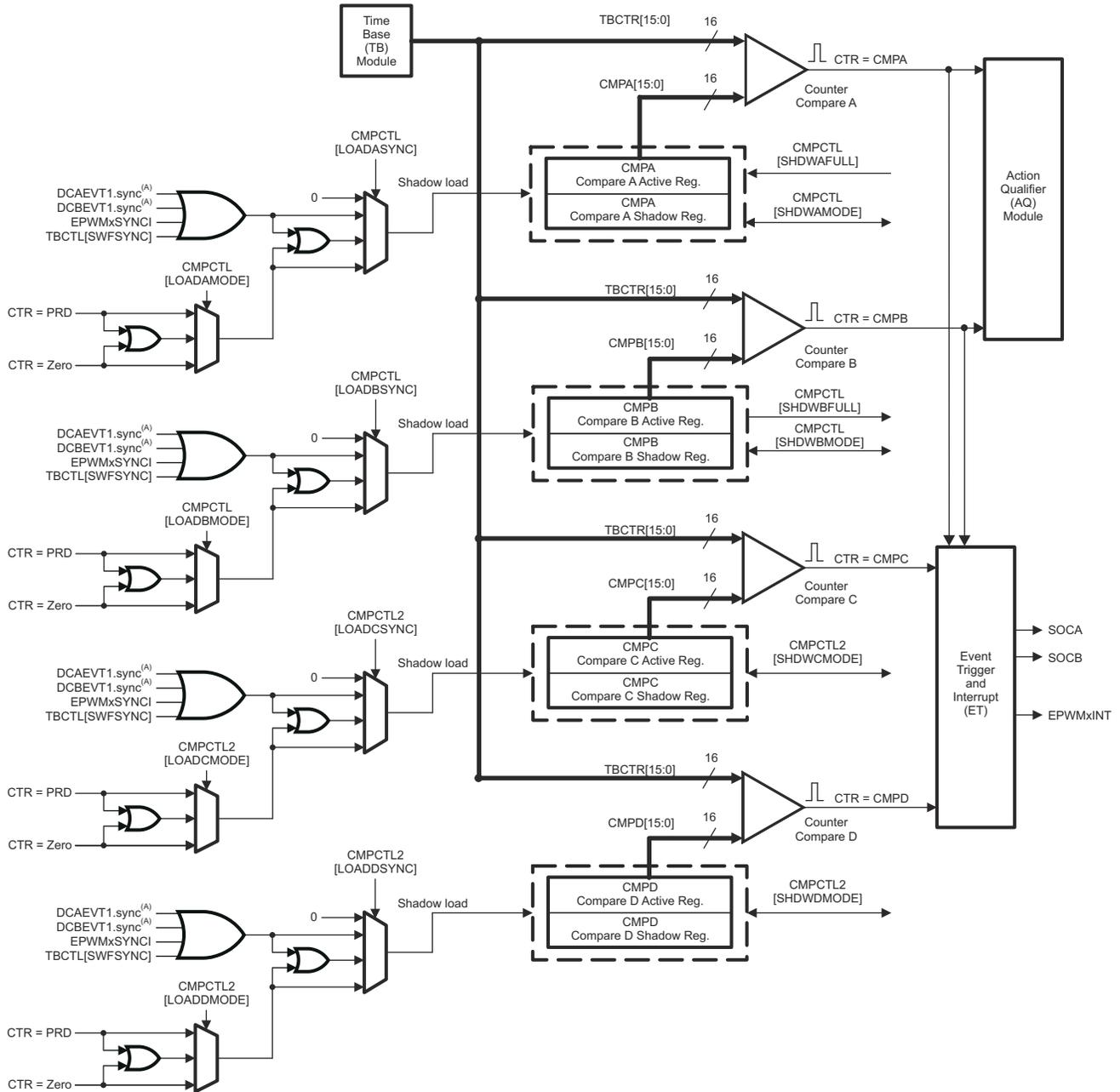
カウンタ比較サブモジュールは、時間ベース・カウンタ値を入力として取り込みます。この値はカウンタ比較 A (CMPA)、カウンタ比較 B (CMPB)、カウンタ比較 C (CMPC)、カウンタ比較 D (CMPD) レジスタと絶えず比較されます。時間ベース・カウンタが比較レジスタのいずれかと等しくなると、カウンタ比較ユニットは適切なイベントを生成します。

カウンタ比較は以下の処理を行います。

- CMPA、CMPB、CMPC、CMPD レジスタを使って、プログラム可能なタイム・スタンプに基づいてイベントを生成します。
 - CTR = CMPA : カウンタ比較 A レジスタに等しい時間ベース・カウンタ (TBCTR = CMPA)
 - CTR = CMPB : カウンタ比較 B レジスタに等しい時間ベース・カウンタ (TBCTR = CMPB)
 - CTR = CMPC : カウンタ比較 C レジスタに等しい時間ベース・カウンタ (TBCTR = CMPC)
 - CTR = CMPD : カウンタ比較 D レジスタに等しい時間ベース・カウンタ (TBCTR = CMPD)
- カウンタ比較 A (CMPA) とカウンタ比較 B (CMPB) を使用してアクション・フィルタ・サブモジュールが適切に設定されている場合、PWM デューティ・サイクルを制御します。
- アクティブ PWM サイクル中のデータ破損またはグリッチを防止するため、新しい比較値をシャドウイングします。

1.5.2 カウンタ比較サブモジュールの制御と監視

図 1-16 に、カウンタ比較サブモジュールの動作を示します。



A. これらのイベントは、TRIPIN 入力 (たとえば CMPSSx および TZ 信号) のレベルに基づいて、ePWM デジタル比較 (DC) サブモジュールによって生成されます。

図 1-16. カウンタ比較サブモジュールの詳細図

1.5.3 カウンタ比較サブモジュールの動作の概要

カウンタ比較サブモジュールは、アクション・フィルタおよびイベント・トリガ・サブモジュールで使用できるイベントを生成します。次の 4 つの独立した比較イベントがあります。

1. CTR = CMPA : 時間ベース・カウンタがカウンタ比較 A レジスタと一致 (TBCTR = CMPA)。
2. CTR = CMPB : 時間ベース・カウンタがカウンタ比較 B レジスタと一致 (TBCTR = CMPB)。
3. CTR = CMPC : 時間ベース・カウンタがカウンタ比較 C レジスタと一致 (TBCTR = CMPC)。このイベントは、イベント・トリガ・サブモジュールでのみ、イベントを生成するために使用できます。
4. CTR = CMPD : 時間ベース・カウンタがカウンタ比較 D レジスタと一致 (TBCTR = CMPD)。このイベントは、イベント・トリガ・サブモジュールでのみ、イベントを生成するために使用できます。

アップ・カウントまたはダウン・カウント・モードでは、各イベントはサイクルごとに 1 回のみ発生します。アップ・ダウン・カウント・モードでは、比較値が 0x00 と TBPRD の間の値である場合、各イベントはサイクルごとに 2 回発生します。比較値が 0x00 または TBPRD と等しい場合、サイクルごとに 1 回発生します。これらのイベントはアクション・フィルタ・サブモジュールに適用されます。アクション・フィルタ・サブモジュールでは、これらのイベントがカウンタの方向によってクオリファイされ、有効化されている場合、アクションに変換されます。詳細については、[セクション 1.6.1](#) を参照してください。

カウンタ比較レジスタ CMPA および CMPB にはそれぞれ 1 つのシャドウ・レジスタが関連付けられています。シャドウイングを行うことで、ハードウェアと同期してレジスタを更新し続けることができます。シャドウイングを使うと、アクティブ・レジスタに対する更新は、計画された適切な時点にのみ行われます。これにより、ソフトウェアによってレジスタが非同期的に変更されることによるデータ破損または誤動作を防止できます。アクティブ・レジスタとシャドウ・レジスタのメモリ・アドレスは同じです。書き込みまたは読み出し対象のレジスタは、CMPCTL[SHDWAMODE] および CMPCTL[SHDWBMODE] ビットによって決定されます。これらのビットは、CMPC シャドウ・レジスタと CMPD シャドウ・レジスタをそれぞれ有効化および無効化します。2 つのロード・モードの挙動は次のとおりです。

シャドウ・モード :

CMPA のシャドウ・モードは、CMPCTL[SHDWAMODE] ビットをクリアすることで有効化され、CMPB のシャドウ・レジスタは、CMPCTL[SHDWBMODE] ビットをクリアすることで有効化されます。CMPA と CMPB のシャドウ・モードは、どちらもデフォルトで有効化されています。

シャドウ・レジスタが有効化されている場合、CMPCTL[LOADAMODE]、CMPCTL[LOADBMODE]、CMPCTL[LOADASYNC]、CMPCTL[LOADBSYNC] レジスタ・ビットで指定されたように、以下のイベントのいずれかでシャドウ・レジスタの内容がアクティブ・レジスタに転送されます。

- CTR = PRD : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
- CTR = ゼロ : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- CTR = PRD かつ CTR = ゼロ
- DCAEVT1、DCBEVT1、EPWMxSYNCl、TBCTL[SWFSYNC] のいずれかによって発生する同期イベント
- 同期イベントと LOADAMODE/LOADBMODE による選択の両方

アクション・フィルタに転送されるイベントを生成するため、アクティブ・レジスタの内容のみがカウンタ比較サブモジュールによって使われます。

注

CMPA/CMPB と LOADAMODE/LOADBMODE の有効な設定については、[セクション 1.6.5](#) を参照してください。

即時ロード・モード :

即時ロード・モードが選択されている場合 (つまり、CMPCTL[SHDWAMODE] = 1 または CMPCTL[SHDWBMODE] = 1)、レジスタの読み出しまたは書き込みはアクティブ・レジスタに対して直接行われます。

追加コンパレータ

ePWM タイプ 2 以降のカウンタ比較サブモジュールは、2 つの比較レジスタに基づいて 2 つの独立した比較イベントの生成を制御します。これらのイベントはイベント・トリガ・サブモジュールに入力されます。

1. CTR = CMPC : 時間ベース・カウンタがカウンタ比較 C レジスタと一致 (TBCTR = CMPC)。
2. CTR = CMPD : 時間ベース・カウンタがカウンタ比較 D レジスタと一致 (TBCTR = CMPD)。

カウンタ比較レジスタ CMPC および CMPD にはそれぞれ 1 つのシャドウ・レジスタが関連付けられています。デフォルトでは、このレジスタはシャドウイングされます。アクティブ・レジスタとシャドウ・レジスタのメモリ・アドレスは同じです。アクティブ CMPC および CMPD レジスタの値は時間ベース・カウンタ (TBCTR) と比較されます。値が等しくなると、カウンタ比較モジュールはイベント「時間ベース・カウンタがカウンタ比較 C またはカウンタ比較 D と一致」をそれぞれ生成します。このレジスタのシャドウイングは、CMPCTL2[SHDWCMODE] および CMPCTL2[SHDWDMODE] ビットによって有効化および無効化されます。これらのビットは、CMPC シャドウ・レジスタと CMPD シャドウ・レジスタをそれぞれ有効化および無効化します。2 つのロード・モードの挙動は次のとおりです。

シャドウ・モード :

CMPC のシャドウ・モードは、CMPCTL2[SHDWCMODE] ビットをクリアすることで有効化され、CMPD のシャドウ・レジスタは、CMPCTL2[SHDWDMODE] ビットをクリアすることで有効化されます。CMPC と CMPD のシャドウ・モードは、どちらもデフォルトで有効化されています。

シャドウ・レジスタが有効化されている場合、CMPCTL2[LOADCMODE]、CMPCTL2[LOADDMODE]、CMPCTL2[LOADCSYNC]、CMPCTL2[LOADDSYNC] レジスタ・ビットで指定されたように、以下のイベントのいずれかでシャドウ・レジスタの内容がアクティブ・レジスタに転送されます。

- CTR = PRD : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
- CTR = ゼロ : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- CTR = PRD かつ CTR = ゼロ
- DCAEVT1、DCBEVT1、EPWMxSYNCl、TBCTL[SWFSYNC] のいずれかによって発生する同期イベント
- 同期イベントと LOADCMODE/LOADDMODE による選択の両方

アクション・フィルタに転送されるイベントを生成するため、アクティブ・レジスタの内容のみがカウンタ比較サブモジュールによって使われます。

即時ロード・モード :

即時ロード・モードが選択されている場合 (つまり、CMPCTL2[SHDWCMODE] = 1 または CMPCTL2[SHDWDMODE] = 1)、レジスタの読み出しまたは書き込みはアクティブ・レジスタに対して直接行われます。

グローバル・ロードのサポート

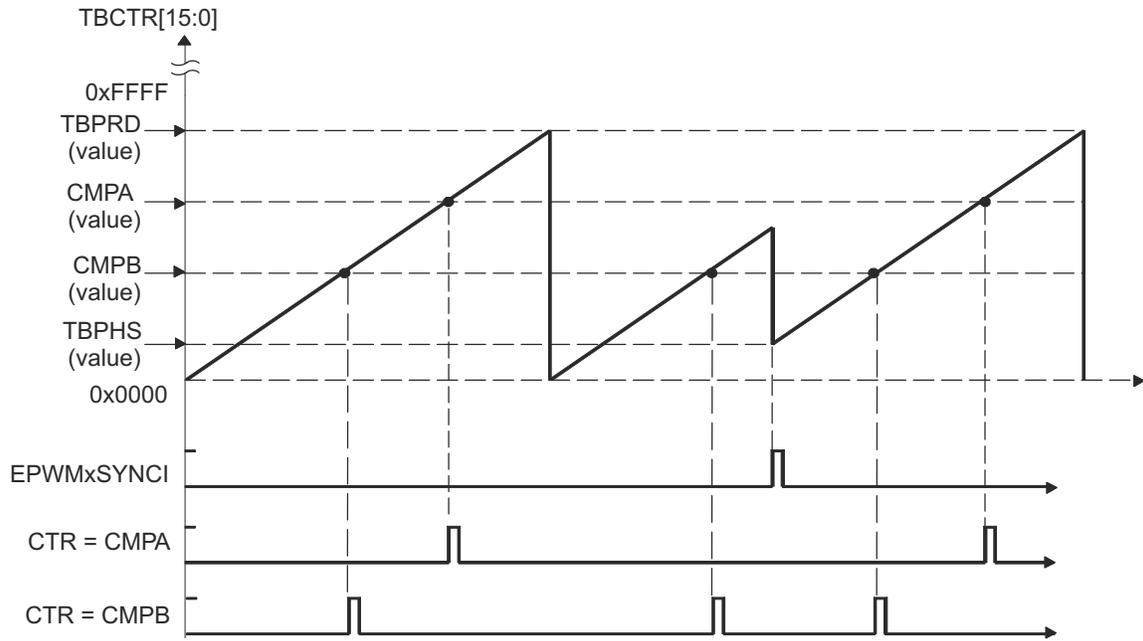
グローバル・ロード制御機能は、グローバル・ロード構成レジスタ (GLDCFG) の適切なビットを設定することで、すべてのカウンタ比較レジスタのためにも使用できます。グローバル・ロード・モードが選択されている場合、このモードが有効化されたすべてのレジスタにおいて、シャドウ・レジスタからアクティブ・レジスタへの内容の転送は、グローバル・シャドウ・アクティブ・ロード制御レジスタ (GLDCTL) の構成ビットで設定されたイベントと同じイベントで行われます。グローバル・ロード制御機能については、[セクション 1.4.7](#) で説明します。

1.5.4 カウント・モードのタイミング波形

カウンタ比較モジュールは、3 つのカウント・モードのすべてで比較イベントを生成できます。

- アップ・カウント・モード : 非対称型 PWM 波形を生成するために使います。
- ダウン・カウント・モード : 非対称型 PWM 波形を生成するために使います。
- アップ・ダウン・カウント・モード : 対称型 PWM 波形を生成するために使います。

最初の 3 つのモードの動作を最もよく説明するため、[図 1-17](#) ~ [図 1-20](#) のタイミング図に、イベントが生成されるタイミングと EPWMxSYNCl 信号の相互作用の仕方を示しています。



EPWMxSYNCl 外部同期イベントは、TBCTR カウント・シーケンスに不連続性を生じさせることがあります。これが原因で、比較イベントがスキップされる可能性があります。このスキップは通常動作と見なされるため、考慮に入れる必要があります。

図 1-17. アップ・カウント・モードでのカウンタ比較イベント波形

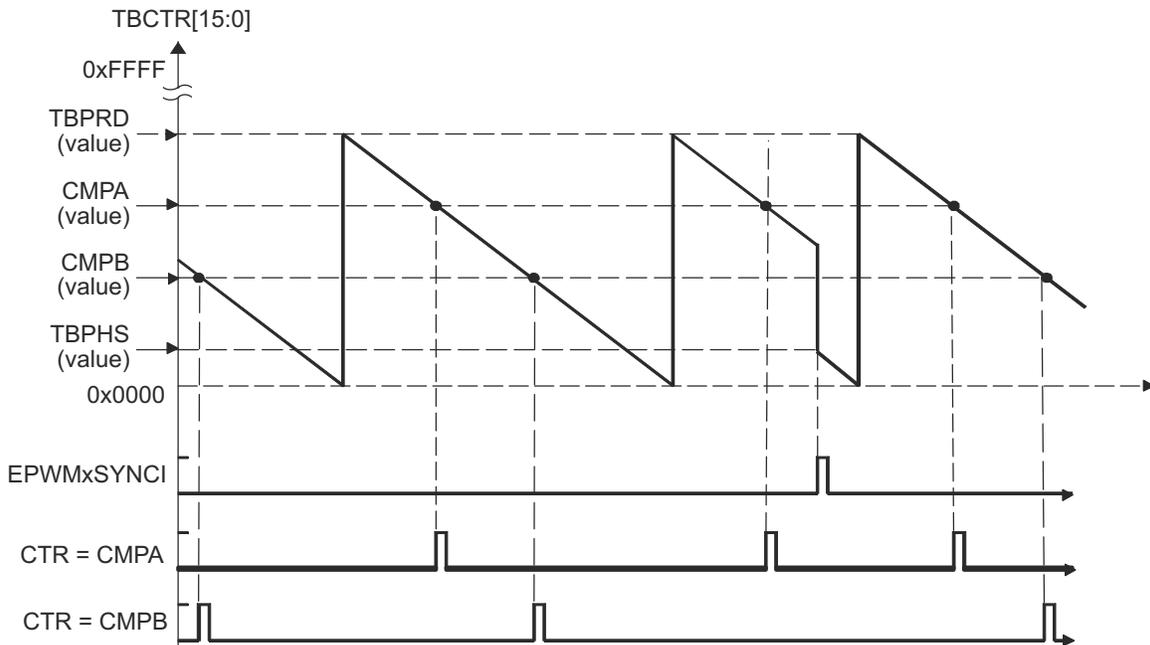


図 1-18. ダウン・カウント・モードでのカウンタ比較イベント

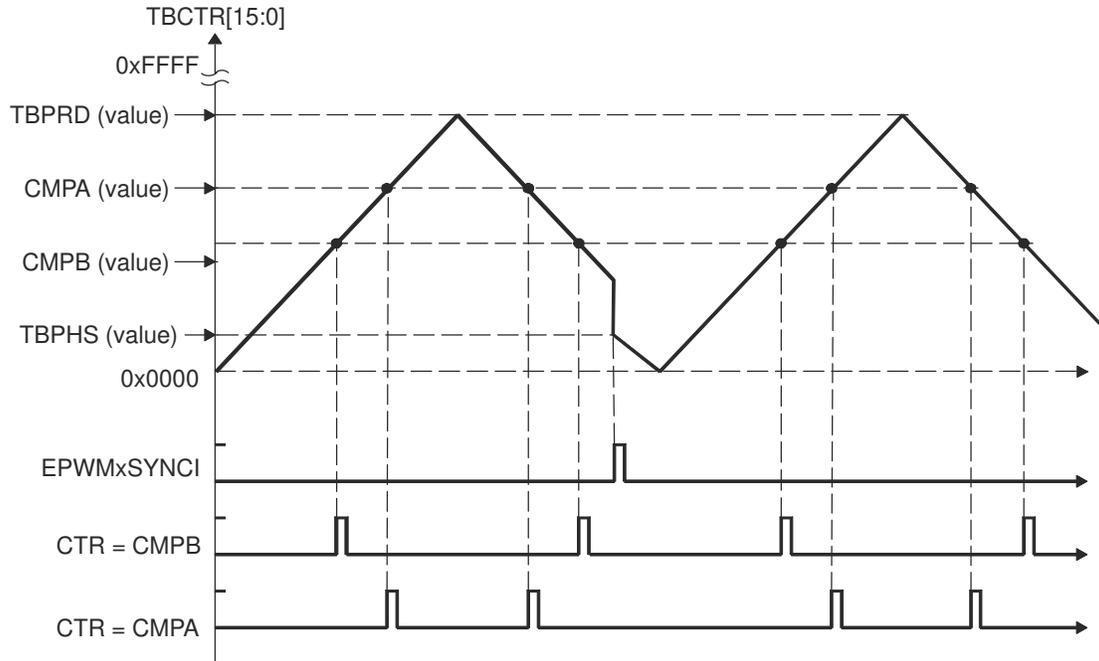


図 1-19. アップ・ダウン・カウント・モードでのカウンタ比較イベント : TBCTL[PHSDIR = 0] (同期イベント後 TBCTL カウント・ダウン)

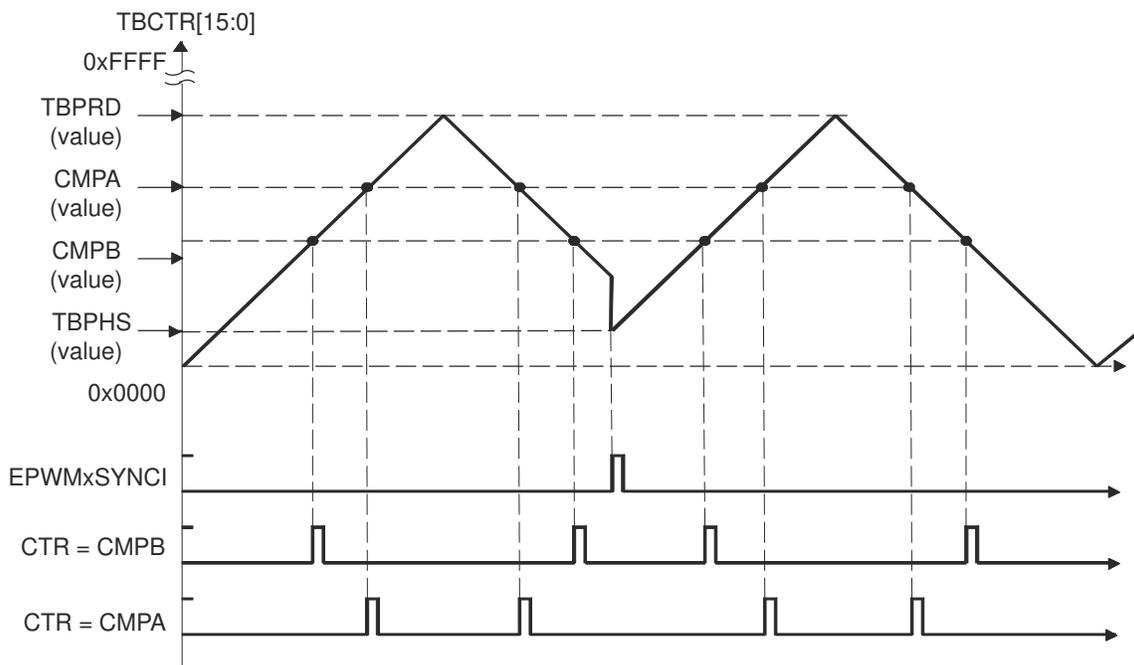


図 1-20. アップ・ダウン・カウント・モードでのカウンタ比較イベント : TBCTL[PHSDIR = 1] (同期イベント後 TBCTL カウント・アップ)

1.6 アクション・フィルタ (AQ) サブモジュール

アクション・フィルタ・サブモジュールは、波形の構成と PWM の生成において最も重要な役割を果たします。アクション・フィルタ・サブモジュールは、各種のアクションに変換するイベントの種類を決定することで、EPWMxA および EPWMxB 出力で必要なスイッチング波形を生成します。

図 1-21 に、ePWM 内のアクション・フィルタ・サブモジュールを示します。

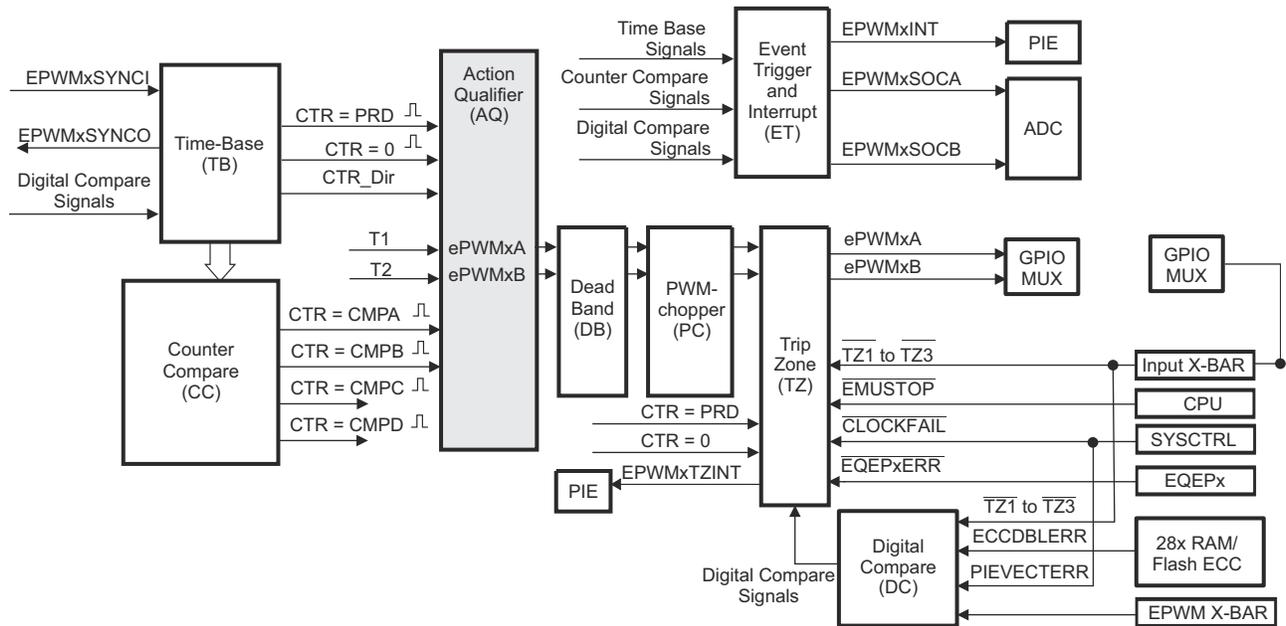


図 1-21. アクション・フィルタ・サブモジュール

1.6.1 アクション・フィルタ・サブモジュールの目的

アクション・フィルタ・サブモジュールは以下の処理を行います。

- 以下のイベントに基づいて、アクションのクオリフィケーションと生成 (セット、クリア、トグル) を行います。
 - CTR = PRD : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
 - CTR = ゼロ : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
 - CTR = CMPA : 時間ベース・カウンタがカウンタ比較 A レジスタと一致 (TBCTR = CMPA)
 - CTR = CMPB : 時間ベース・カウンタがカウンタ比較 B レジスタと一致 (TBCTR = CMPB)
- T1、T2 イベントは以下の処理を行います。コンパレータ・イベント、トリップ・イベント、同期イベントに基づいてイベントをトリガします。
- これらのイベントが同時に発生した場合の優先順位を管理します。
- 時間ベース・カウンタがインクリメントしている際とデクリメントしている際のイベントを独立して制御します。

1.6.2 アクション・フィルタ・サブモジュールの制御とステータス・レジスタの定義

図 1-22 に、アクション・フィルタ・サブモジュールの動作を示します。この動作は、[セクション 1.17](#) のレジスタによって監視されます。

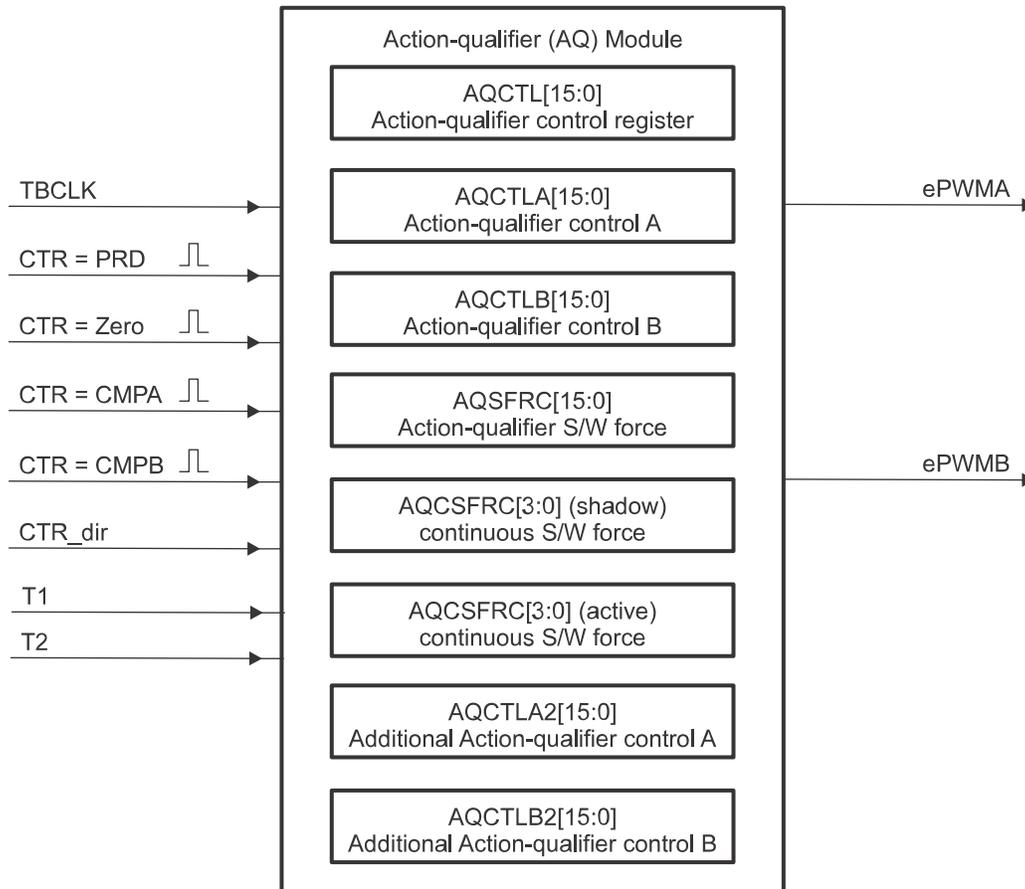


図 1-22. アクション・フィルタ・サブモジュールの入力と出力

便宜のため、あり得る入力イベントを再度[表 1-4](#)にまとめます。

表 1-4. アクション・フィルタ・サブモジュールのあり得る入力イベント

信号	説明	比較対象のレジスタ
CTR = PRD	時間ベース・カウンタが周期値と一致	TBCTR = TBPRD
CTR = ゼロ	時間ベース・カウンタがゼロと一致	TBCTR = 0x00
CTR = CMPA	時間ベース・カウンタがカウンタ比較 A と一致	TBCTR = CMPA
CTR = CMPB	時間ベース・カウンタがカウンタ比較 B と一致	TBCTR = CMPB
T1 イベント	コンパレータに基づくトリップまたは同期入力イベント	なし
T2 イベント	コンパレータに基づくトリップまたは同期入力イベント	なし
ソフトウェアによって強制されたイベント	ソフトウェアによって開始された非同期イベント	

ソフトウェアによって強制されたアクションは、便利な非同期イベントです。この制御は、AQSFRC および AQCSFRC レジスタを使って処理されます。

注

CSFA をシャドウ・モードで使用しない場合、シャドウ・モードを無効化するように RLDCSF ビットを設定する必要があります。

アクション・フィルタ・サブモジュールは、特定のイベントが発生した際の 2 つの出力 (EPWMxA、EPWMxB) の挙動を制御します。アクション・フィルタ・サブモジュールへのイベント入力、カウンタの方向 (アップまたはダウン) によってさらにクオリファイされ (絞り込まれ) ます。これにより、カウント・アップとカウント・ダウンの両方の局面で、独立したアクションを出力で実行できます。

出力 (EPWMxA、EPWMxB) で実行できるアクションは次のとおりです。

- **High にセット** : 出力 (EPWMxA または EPWMxB) を High レベルに設定します。
- **Low にクリア** : 出力 (EPWMxA または EPWMxB) を Low レベルに設定します。
- **トグル** : EPWMxA または EPWMxB が現在 High にプルされている場合、出力を Low にプルします。EPWMxA または EPWMxB が現在 Low にプルされている場合、出力を High にプルします。
- **何もしない** : 出力 (EPWMxA、EPWMxB) を、現在設定されているレベルに維持します。「何もしない」を選択すると、イベントが EPWMxA および EPWMxB 出力のアクションを引き起こすことを禁止できますが、このイベントは依然として割り込みと ADC 変換開始をトリガできます。詳細については、[セクション 1.10](#) の説明を参照してください。

アクションは、どちらの出力 (EPWMxA または EPWMxB) に対しても個別に指定されます。任意の (またはすべての) イベントは、特定の出力でアクションを生成するように設定できます。たとえば、CTR = CMPA と CTR = CMPB の両方が出力 EPWMxA で動作できます。すべてのフィルタ・アクションは、「ePWM レジスタ」セクションに記載された制御レジスタを使って設定されます。

分かりやすくするため、この章のイラストでは記号で表した一連のアクションを使用しています。図 1-23 に、これらの記号をまとめます。各記号は、時間的に 1 つのアクションを 1 つのマーカとして表しています。CMPA および CMPB アクションが時間的に移動可能であり、それらの時間的位置がカウンタ比較 A および B レジスタによってそれぞれ設定されるのに対して、一部のアクションは時間的に (ゼロまたは周期値に) 固定されています。アクションをターンオフ (無効化) するには、「何もしない」(リセット時のデフォルト) を選択します。

SW force	TB Counter equals			Trigger Events			Actions
	Zero	Comp A	Comp B	Period	T1	T2	
							Do Nothing
							Clear Lo
							Set Hi
							Toggle

図 1-23. EPWMxA および EPWMxB 出力のあり得るアクション・フィルタ動作

アクション・フィルタ・トリガ・イベント・ソース選択レジスタ (AQTSRCSEL) は、T1 および T2 イベントのソースを選択するために使います。アクション・フィルタ・サブモジュールでの T1/T2 の選択とトリップ / デジタル比較イベントの設定は、トリップ・ゾーン・サブモジュールでのそのイベントの設定とは無関係です。PWM 生成を制御するため、特定のトリップ・イベントを、トリップ・ゾーン・サブモジュールでトリップ・アクションを引き起こすように設定することも、引き起こさないように設定することもできますが、同じイベントをアクション・フィルタが使うことで、T1/T2 を生成させることもできます。

1.6.3 アクション・フィルタ・イベントの優先度

ePWM アクション・フィルタは、同時に複数のイベントを受け取ることができます。この場合、ハードウェアによってイベントに優先度が割り当てられます。一般に、時間的により遅く発生したイベントはより高い優先度を持ち、ソフトウェアによって強制されたイベントは常に最も高い優先度を持ちます。表 1-5 に、アップ・ダウン・カウント・モードのイベント優先レベルを示します。優先レベル 1 が最も高い優先度、レベル 10 が最も低い優先度です。TBCTR の方向によって、優先度はわずかに変化します。

表 1-5. アップ・ダウン・カウント・モードのアクション・フィルタ・イベントの優先度

優先レベル	TBCR = ゼロから TBCTR = TBPRD まで TBCTR がインクリメントしている場合のイベント	TBCTR = TBPRD から TBCTR = 1 まで TBCTR がデクリメントしている場合のイベント
1 (最高)	ソフトウェアによって強制されたイベント	ソフトウェアによって強制されたイベント
2	アップ・カウント時の T1 (T1U)	ダウン・カウント時の T1 (T1D)
3	アップ・カウント時の T2 (T2U)	ダウン・カウント時の T2 (T2D)
4	アップ・カウント時にカウンタが CMPB と一致 (CBU)	ダウン・カウント時にカウンタが CMPB と一致 (CBD)
5	アップ・カウント時にカウンタが CMPA と一致 (CAU)	ダウン・カウント時にカウンタが CMPA と一致 (CAD)
6	カウンタがゼロと一致	カウンタが周期と一致 (TBPRD)
7	ダウン・カウント時の T1 (T1D)	アップ・カウント時の T1 (T1U)
8	ダウン・カウント時の T2 (T2D)	アップ・カウント時の T2 (T2U)
9	ダウン・カウント時にカウンタが CMPB と一致 (CBD)	アップ・カウント時にカウンタが CMPB と一致 (CBU)
10 (最低)	ダウン・カウント時にカウンタが CMPA と一致 (CAD)	アップ・カウント時にカウンタが CMPA と一致 (CAU)

表 1-6 に、アップ・カウント・モードのアクション・フィルタの優先度を示します。この場合、カウンタの方向は常に昇順に定義されているため、ダウン・カウント・イベントは決して発生しません。

表 1-6. アップ・カウント・モードのアクション・フィルタ・イベントの優先度

優先レベル	イベント
1 (最高)	ソフトウェアによって強制されたイベント
2	カウンタが周期と一致 (TBPRD)
3	アップ・カウント時の T1 (T1U)
4	アップ・カウント時の T2 (T2U)
5	アップ・カウント時にカウンタが CMPB と一致 (CBU)
6	アップ・カウント時にカウンタが CMPA と一致 (CAU)
7 (最低)	カウンタがゼロと一致

表 1-7 に、ダウン・カウント・モードのアクション・フィルタの優先度を示します。この場合、カウンタの方向は常に降順に定義されているため、アップ・カウント・イベントは決して発生しません。

表 1-7. ダウン・カウント・モードのアクション・フィルタ・イベントの優先度

優先レベル	イベント
1 (最高)	ソフトウェアによって強制されたイベント
2	カウンタがゼロと一致
3	ダウン・カウント時の T1 (T1D)
4	ダウン・カウント時の T2 (T2D)
5	ダウン・カウント時にカウンタが CMPB と一致 (CBD)
6	ダウン・カウント時にカウンタが CMPA と一致 (CAD)

表 1-7. ダウン・カウント・モードのアクション・フィルタ・イベントの優先度 (続き)

優先レベル	イベント
7 (最低)	カウンタが周期と一致 (TBPRD)

周期より大きい比較値を設定することもできます。この場合、表 1-8 に示すようにアクションが実行されません。

表 1-8. CMPA/CMPB が周期より大きい場合の挙動

カウンタ・モード	アップ・カウント時比較イベント CAD/CBD	ダウン・カウント時比較イベント CAD/CBD
アップ・カウント・モード	CMPA/CMPB ≤ TBPRD の場合、比較一致 (TBCTR = CMPA または CMPB) 時にイベントが発生します。 CMPA/CMPB > TBPRD の場合、イベントは発生しません。	何も発生しません。
ダウン・カウント・モード	何も発生しません。	CMPA/CMPB < TBPRD の場合、比較一致 (TBCTR = CMPA または CMPB) 時にイベントが発生します。 CMPA/CMPB ≥ TBPRD の場合、周期一致 (TBCTR = TBPRD) 時にイベントが発生します。
アップ・ダウン・カウント・モード	CMPA/CMPB < TBPRD でありかつカウンタがインクリメントしている場合、比較一致 (TBCTR = CMPA または CMPB) 時にイベントが発生します。 CMPA/CMPB ≥ TBPRD の場合、周期一致 (TBCTR = TBPRD) 時にイベントが発生します。	CMPA/CMPB < TBPRD でありかつカウンタがデクリメントしている場合、比較一致 (TBCTR = CMPA または CMPB) 時にイベントが発生します。 CMPA/CMPB ≥ TBPRD の場合、周期一致 (TBCTR = TBPRD) 時にイベントが発生します。

1.6.4 AQCTLA および AQCTLB シャドウ・モード動作

位相が変化した場合でも、1 周期の終わりに行われる必要があるアクション・フィルタ・モードの変更を可能にするため、ePWM タイプ 2 以降では、AQCTLA および AQCTLB レジスタのシャドウイングが追加されています。また、これらのレジスタの同期時のシャドウからアクティブへのロードもサポートされています。このレジスタのシャドウイングは、AQCTL[SHDWAQAMODE] および AQCTL[SHDWAQBMODE] ビットによって有効化および無効化されます。これらのビットは、AQCTLA シャドウ・レジスタと AQCTLA シャドウ・レジスタをそれぞれ有効化および無効化します。2 つのロード・モードの挙動は次のとおりです。

シャドウ・モード：

AQCTLA のシャドウ・モードは AQCTL[SHDWAQAMODE] ビットをセットすることで有効化され、AQCTLB のシャドウ・レジスタは AQCTL[SHDWAQBMODE] ビットをセットすることで有効化されます。デフォルトでは、シャドウ・モードは AQCTLA と AQCTLB の両方で無効化されています

シャドウ・レジスタが有効化されている場合、AQCTL[LDAQAMODE]、AQCTL[LDAQBMODE]、AQCTL[LDAQASYNC]、AQCTL[LDAQBSYNC] レジスタ・ビットで指定されたように、以下のイベントのいずれかでシャドウ・レジスタの内容がアクティブ・レジスタに転送されます。

- CTR = PRD : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
- CTR = ゼロ : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- CTR = PRD かつ CTR = ゼロ
- DCAEVT1、DCBEVT1、EPWMxSYNCl、TBCTL[SWFSYNCl] のいずれかによって発生する同期イベント
- 同期イベントと LDAQAMODE/LDAQBMODE による選択の両方

グローバル・ロードのサポート

グローバル・ロード制御機能は、グローバル・ロード構成レジスタ (GLDCFG) の適切なビットを設定することで、AQCTLA : AQCTLA2、AQCTLB : AQCTLB2、AQCSFRC レジスタのためにも使用できます。グローバル・ロード・モードが選択されている場合、このモードが有効化されたすべてのレジスタにおいて、シャドウ・レジスタからアクティブ・レジスタへの内容の転送は、グローバル・シャドウ - アクティブ・ロード制御レジスタ (GLDCTL) の構成ビットで設定されたイベントと同じイベントで行われます。グローバル・ロード制御機能については、[セクション 1.4.7](#) で説明します。

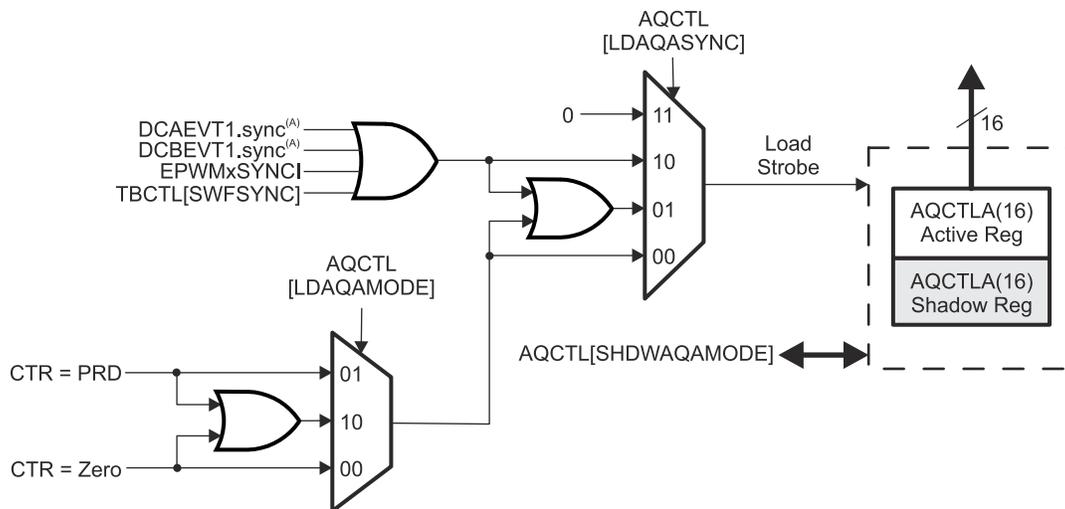
即時ロード・モード：

即時ロード・モードが選択されている場合 (つまり、AQCTL[SHDWAQAMODE] = 0 または AQCTL[SHDWAQBMODE] = 0)、レジスタの読み出しまたは書き込みはアクティブ・レジスタに対して直接行われます。図 1-24 と 図 1-25 を参照してください。

注

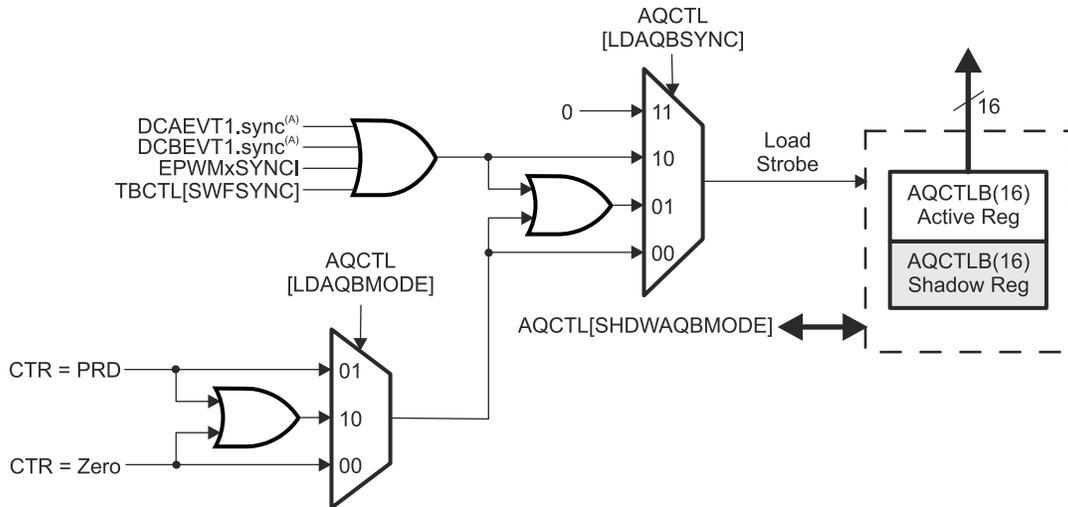
CMPA = 0 または CMPB = 0 境界での、アクション・フィルタ出力 A/B 制御レジスタ [AQCTLA および AQCTLB] のシャドウ・アクティブ・ロード

カウンタ比較 A レジスタ (CMPA) またはカウンタ比較 B レジスタ (CMPB) の値が 0 に設定されており、かつ AQCTLA と AQCTLB のアクション・フィルタ動作が、シャドウ・アクティブ・ロードと同時に実行されるように設定されている (つまり、CMPA = 0 かつ、TBCTR = 0 時に AQCTLA のシャドウ・アクティブ・ロード (AQCTL レジスタの LDAQAMODE および LDAQAMODE ビットを使用)) 場合、両者のイベントは競合します。TBCTR = 0 境界でのアクション・フィルタ出力 A/B 制御レジスタのシャドウ・アクティブ・ロードを使用する場合、ゼロ以外のカウンタ比較を使用することを推奨します。



- A. これらのイベントは、TRIPIN 入力 (たとえば CMPSSx および T \bar{Z} 信号) のレベルに基づいて、ePWM デジタル比較 (DC) サブモジュールによって生成されます。

図 1-24. AQCTL[SHDWAQAMODE]



- A. これらのイベントは、TRIPIN 入力 (たとえば CMPSSx および \overline{TZ} 信号) のレベルに基づいて、ePWM デジタル比較 (DC) サブモジュールによって生成されます。

図 1-25. AQCTL[SHDWAQBMODE]

1.6.5 代表的な波形に必要な設定

注

このドキュメントの波形は、静的な比較レジスタ値に対する ePWM の挙動を示しています。実行中のシステムでは、アクティブ比較レジスタ (CMPA および CMPB) は通常、各周期に 1 回、それぞれのシャドウ・レジスタによって更新されます。更新が行われるタイミング (時間ベース・カウンタがゼロに達したときと時間ベース・カウンタが周期値に達したときのどちらか) を指定します。新しい値に基づくアクションが 1 周期だけ遅延する場合や、古い値に基づくアクションが 1 周期余分に効力を生じる場合があります。PWM の設定の仕方によっては、この状況を回避できます。それらには以下が含まれますが、以下に限定されません。

対称型 PWM を生成するためにアップ・ダウン・カウント・モードを使用 :

- ゼロで CMPA/CMPB をロードする場合、1 以上の CMPA/CMPB 値を使用します。
- 周期値で CMPA/CMPB をロードする場合、TPRD - 1 以下の CMPA/CMPB 値を使用します。

これは、非常に短い場合はシステムが無視しがちな、少なくとも 1 TBCLK サイクルの幅のパルスが 1 つの PWM 周期内に常に存在することを意味します。

非対称型 PWM を生成するためにアップ・ダウン・カウント・モードを使用 :

- 50%~0% の非対称型 PWM を実現するには、以下の設定を使います。周期値で CMPA/CMPB をロードし、PWM をクリアするために周期アクションを使い、PWM をセットするために比較アップ・アクションを使います。50%~0% の PWM デューティを実現するため、比較値を 0 から TBPRD まで変化させます。

非対称型 PWM を生成するためにアップ・カウント・モードを使用 :

- 0~100% の非対称型 PWM を実現するには、TBPRD で CMPA/CMPB をロードする必要があります。TBCTR = PRD で CMPA/CMPB がロードされない場合、書き込みのタイミングと CMPA/CMPB に書き込まれた値によっては、境界条件が発生する可能性があります。PWM をセットするためにゼロ・アクションを使い、PWM をクリアするために比較アップ・アクションを使います。0~100% の PWM デューティを実現するため、比較値を 0 から TBPRD + 1 まで変化させます。

デッドバンドが有効化された非対称型 PWM を生成するためにアップ・カウント・モードを使用 :

- 0%~100% の PWM を実現するには、以下の設定を使います。「CMPX < デッドバンド」または「CMPX > PRD - デッドバンド」の条件がほとんど満たされるほど、CMPA の値が 0 または PRD に近い場合、CMPX の AQCTL レジスタで指定されたアクションは効力を生じません。これを回避するため、CAU と CAD の両方のイベントのために High と Low のどちらかのパルスを生成 (両方をセットまたは両方をクリア) するためにのみ、これらの条件の下で AQCTL 設定を変更する必要があります。このソフトウェア更新が、必ず PWM キャリア・サイクルと同期して行われるようにします。また、シャドウ・モードを必ず有効化します。

デッドバンドが有効化された非対称型 PWM を生成するためにアップ・ダウン・カウント・モードを使用 :

- 0%~100% の PWM を実現するには、以下の設定を使います。「CMPX < デッドバンド / 2」または「CMPX > PRD - (デッドバンド) / 2」の条件がほとんど満たされるほど、CMPA の値が 0 または PRD に近い場合、CMPX の AQCTL レジスタで指定されたアクションは効力を生じません。これを回避するため、CAU と CAD の両方のイベントのために High と Low のどちらかのパルスを生成 (両方をセットまたは両方をクリア) するためにのみ、これらの条件の下で AQCTL 設定を変更する必要があります。このソフトウェア更新が、必ず PWM キャリア・サイクルと同期して行われるようにします。また、シャドウ・モードを必ず有効化します。

『0~100% デューティ・サイクル制御のための拡張パルス幅変調器 (ePWM) モジュールの使い方』を参照してください。

図 1-26 に、TBCTR のアップ・ダウン・カウント・モードを使って対称型 PWM 波形を生成する方法を示します。このモードでは、波形のアップ・カウント部とダウン・カウント部で同じ比較一致を使うことで、0%~

100% DC 変調を実現しています。この例では、比較を行うために CMPA が使われています。カウンタがインクリメントしている場合、CMPA の一致によって PWM 出力は High にプルされます。同様に、カウンタがデクリメントしている場合、比較一致によって PWM 信号は Low にプルされます。CMPA = 0 の場合、PWM 信号は全周期にわたって High になり、100% デューティ波形が得られます。CMPA = TBPRD の場合、PWM 信号は Low になり、0% デューティが得られます。

この構成を実際を使用する際に、ゼロで CMPA/CMPB をロードする場合、1 以上の CMPA/CMPB 値を使用します。周期値で CMPA/CMPB をロードする場合、TPRD - 1 以下の CMPA/CMPB 値を使用します。これは、非常に短い場合はシステムが無視しがちな、少なくとも 1 TBCLK サイクルの幅のパルスが 1 つの PWM 周期内に常に存在することを意味します。

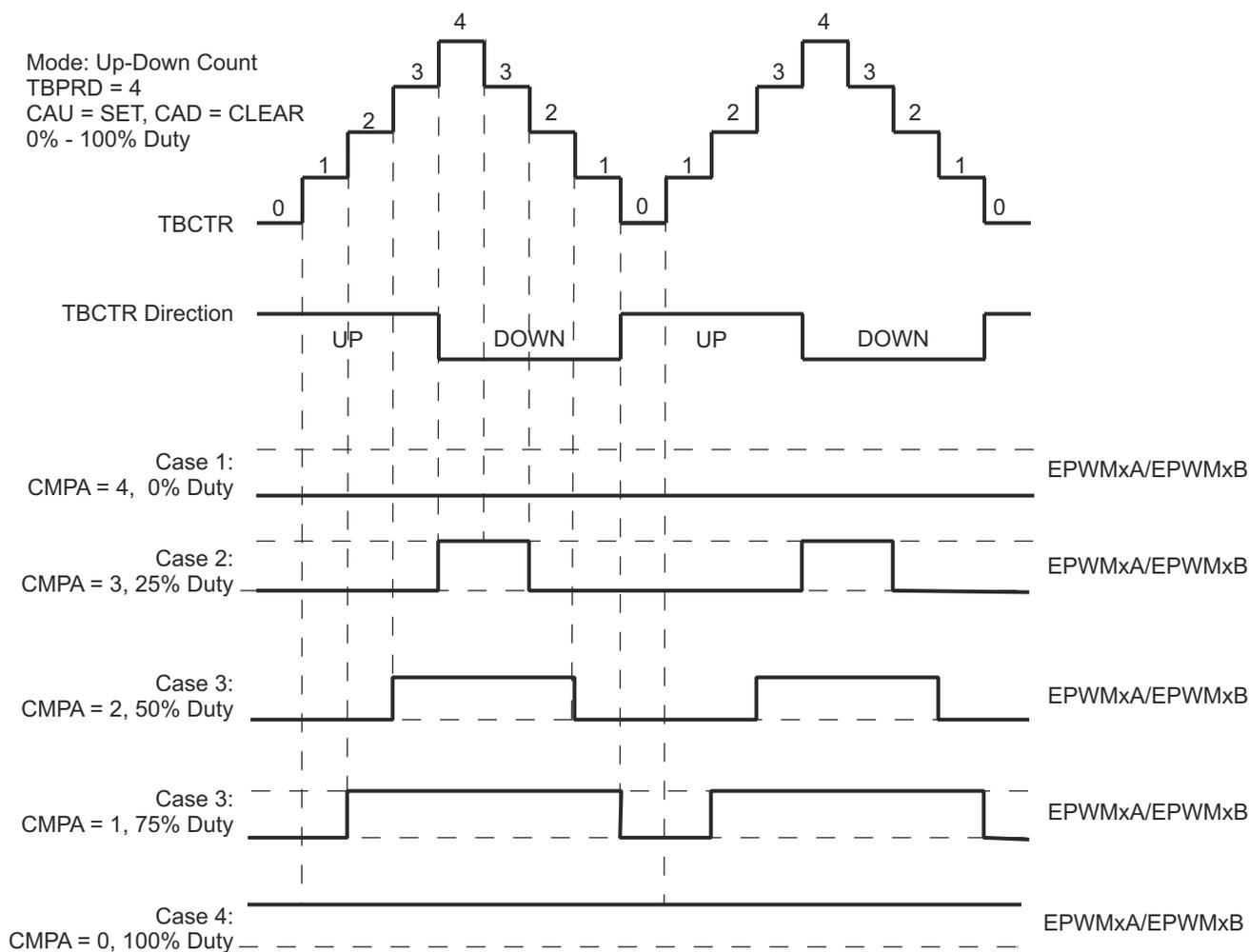
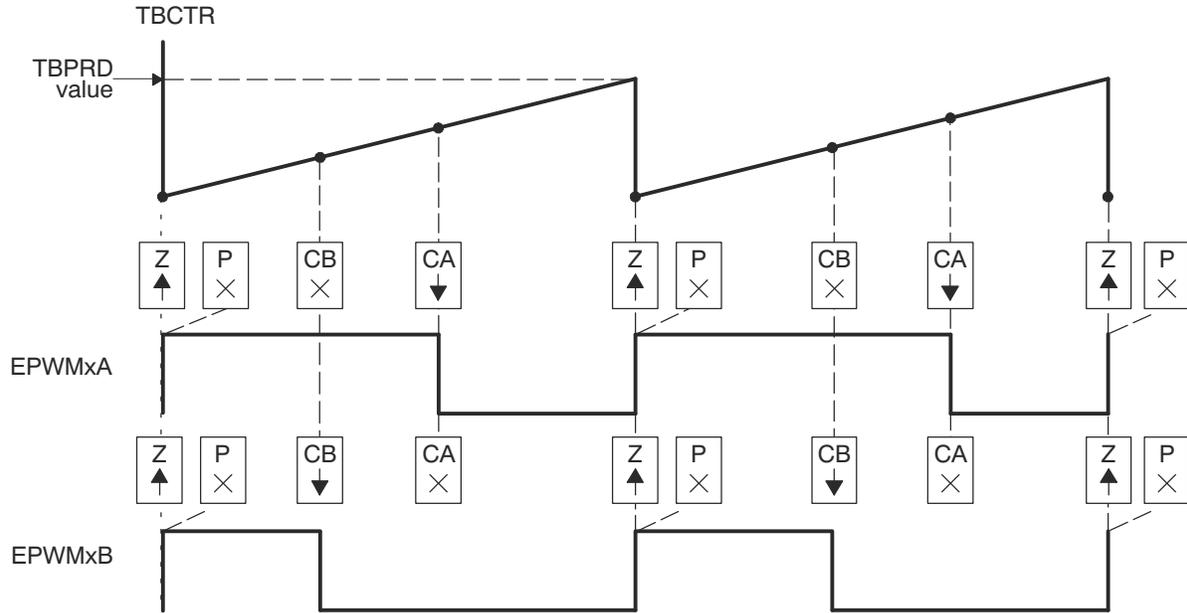


図 1-26. アップ・ダウン・カウント・モードの対称型波形

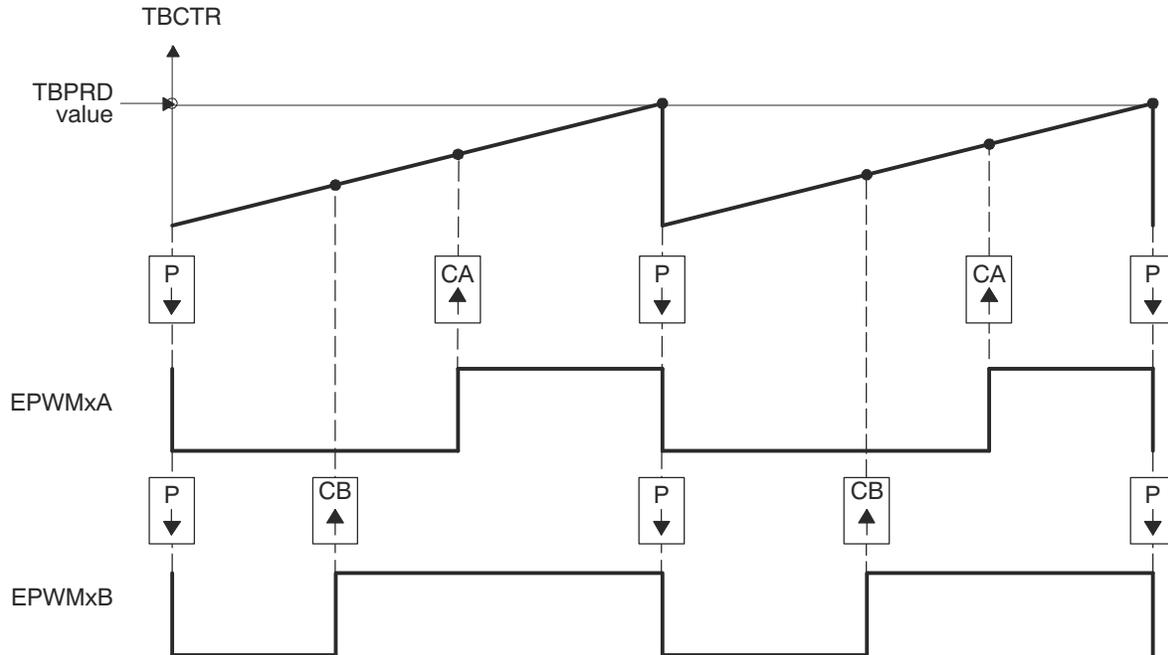
図 1-27 から 図 1-32 までの PWM 波形は、いくつかの代表的なアクション・フィルタの構成を示しています。図と例で使用されている規則は以下のとおりです。

- TBPRD、CMPA、CMPB は、それぞれのレジスタに書き込まれた値を参照します。アクティブ・レジスタ (シャドウ・レジスタではなく) がハードウェアによって使用されます。
- CMPx は CMPA または CMPB を指します。
- EPWMxA と EPWMxB は ePWMx からの出力信号を指します。
- アップ・ダウンとはカウント・アップおよびカウント・ダウン・モードを意味し、アップとはアップ・カウント・モードを意味し、ダウンとはダウン・カウント・モードを意味します。
- Sym = 対称型、Asym = 非対称型



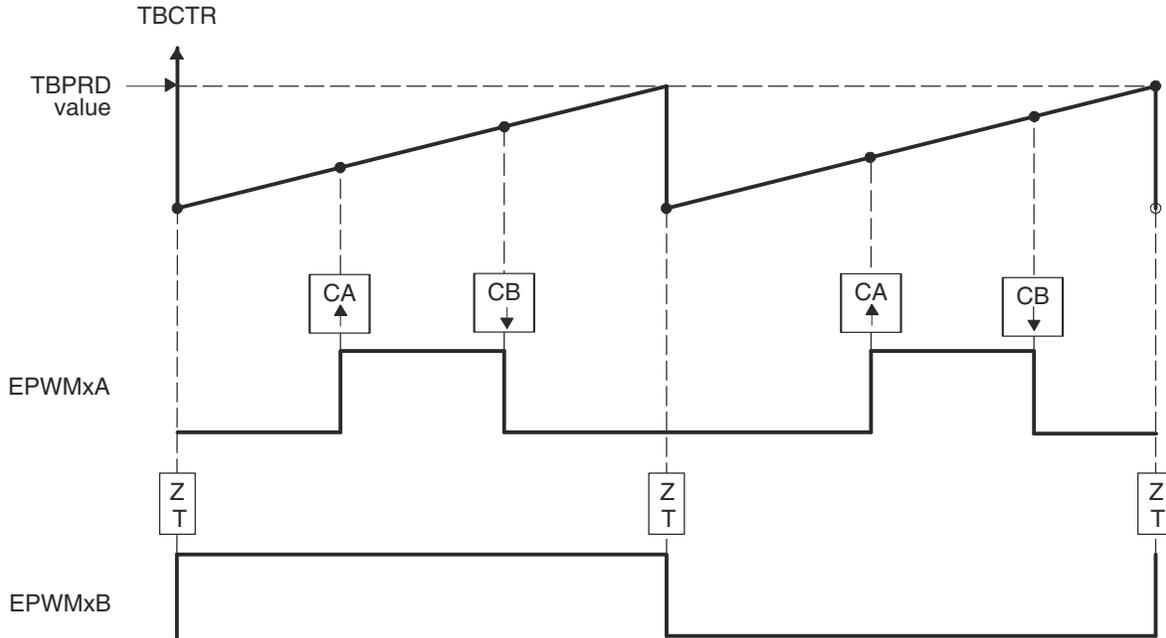
- PWM 周期 = $(TBPRD + 1) \times T_{TBCLK}$
- EPWMxA のデューティ変調は CMPA によって設定され、アクティブ High (つまり、High 期間のデューティが CMPA に比例) です。
- EPWMxB のデューティ変調は CMPB によって設定され、アクティブ High (つまり、High 期間のデューティが CMPB に比例) です。
- 「何もしない」アクション (X) は完全性を示すために表示されていますが、以降の図には表示されていません。
- ゼロおよび周期値でのアクションは、同時に行われているように見えますが、実際には 1 TBCLK 周期離れています。TBCTR は周期値から 0000 に戻り、動作を繰り返します。

図 1-27. アップ、シングル・エッジ非対称型波形、EPWMxA および EPWMxB 独立変調あり — アクティブ High



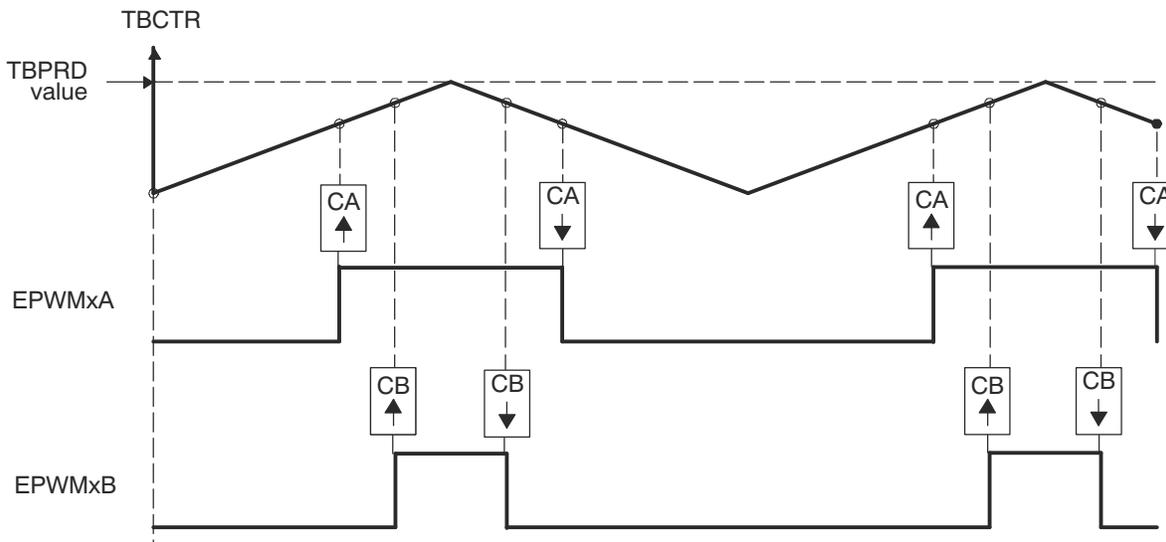
- A. PWM 周期 = $(TBPRD + 1) \times T_{TBCLK}$
- B. EPWMxA のデューティ変調は CMPA によって設定され、アクティブ Low (つまり、Low 期間のデューティが CMPA に比例) です。
- C. EPWMxB のデューティ変調は CMPB によって設定され、アクティブ Low (つまり、Low 期間のデューティが CMPB に比例) です。
- D. ゼロおよび周期値でのアクションは、同時に行われているように見えますが、実際には 1 TBCLK 周期離れています。TBCTR は周期値から 0000 に戻り、動作を繰り返します。

図 1-28. アップ、シングル・エッジ非対称型波形、EPWMxA および EPWMxB 独立変調あり — アクティブ Low



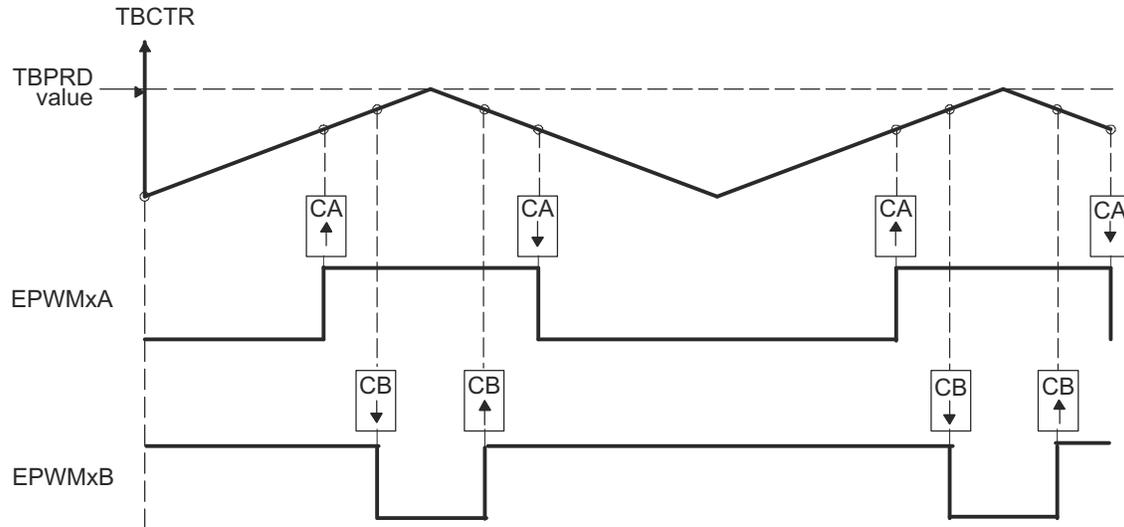
- A. PWM 周波数 = $1 / ((TBPRD + 1) \times T_{TBCLK})$
- B. PWM サイクル内の任意の位置 (0000 ~ TBPRD) にパルスを配置できます。
- C. High 時間デューティは (CMPB - CMPA) に比例します。

図 1-29. アップ・カウント、パルス配置非対称型波形、EPWMxA 独立変調あり



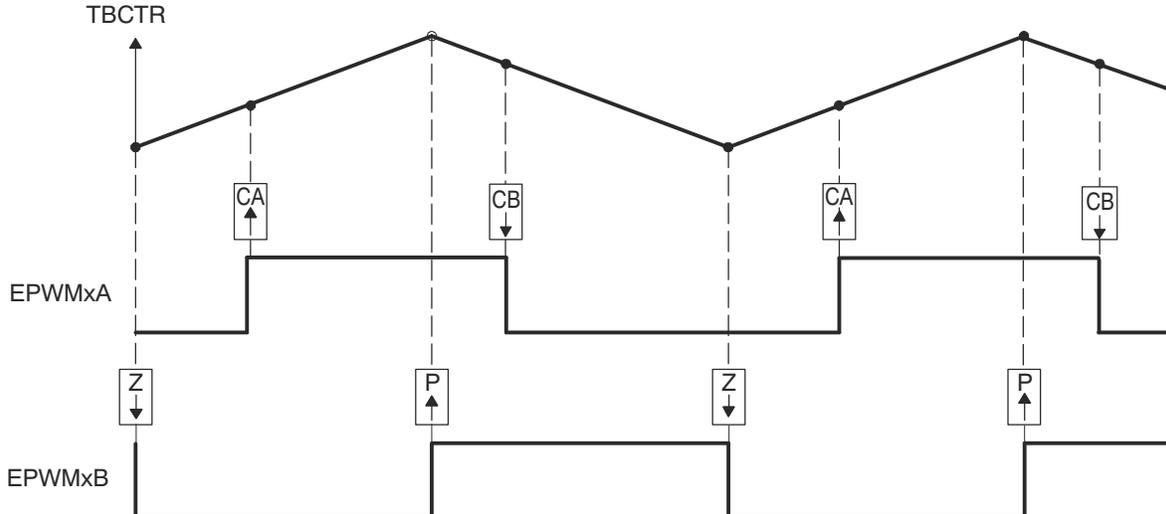
- A. PWM 周期 = $2 \times TBPRD \times T_{TBCLK}$
- B. EPWMxA のデューティ変調は CMPA によって設定され、アクティブ Low (つまり、Low 期間のデューティが CMPA に比例) です。
- C. EPWMxB のデューティ変調は CMPB によって設定され、アクティブ Low (つまり、Low 期間のデューティが CMPB に比例) です。
- D. 出力 EPWMxA および EPWMxB は、独立したパワー・スイッチを駆動できます。

図 1-30. アップ・ダウン・カウント、デュアル・エッジ対称型波形、EPWMxA および EPWMxB 独立変調あり
— アクティブ Low



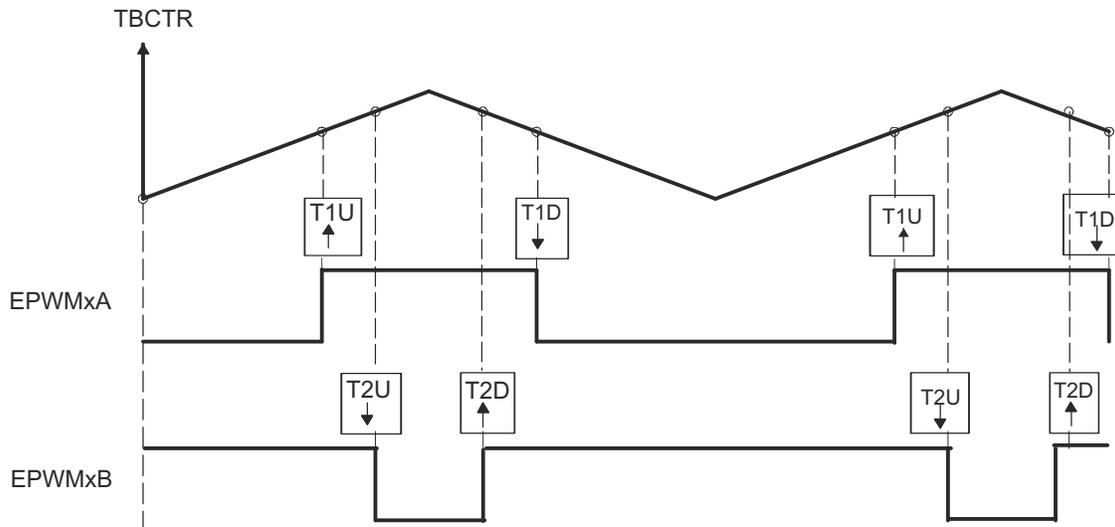
- A. PWM 周期 = $2 \times \text{TBPRD} \times T_{\text{TBCLK}}$
- B. EPWMxA のデューティ変調は CMPA によって設定され、アクティブ Low (つまり、Low 期間のデューティが CMPA に比例) です。
- C. EPWMxB のデューティ変調は CMPB によって設定され、アクティブ High (つまり、High 期間のデューティが CMPB に比例) です。
- D. 出力 EPWMx は、上側 / 下側 (相補型) パワー・スイッチを駆動できます。
- E. デッドバンド = $\text{CMPB} - \text{CMPA}$ (エッジの配置をソフトウェアで詳細に設定可能)。伝統的なエッジ遅延方式が必要な場合、デッドバンド・モジュールも利用できます。

図 1-31. アップ・ダウン・カウント、デュアル・エッジ対称型波形、EPWMxA および EPWMxB 独立変調あり — 相補型



- A. PWM 周期 = $2 \times \text{TBPRD} \times \text{TBCLK}$
- B. 立ち上がりエッジと立ち下がりエッジは、PWM サイクル内で非対称に配置できます。これにより、パルス配置手法が可能になります。
- C. EPWMxA のデューティ変調は CMPA と CMPB によって設定されます。
- D. EPWMxA の Low 時間デューティは $(\text{CMPA} + \text{CMPB})$ に比例します。
- E. この例をアクティブ High に変更するには、CMPA と CMPB の動作を反転させる必要があります (つまり、CMPA でクリア、CMPB でセット)。
- F. EPWMxB のデューティ変調は 50% に固定されています (EPWMxB の予備のアクション・リソースを利用)。

図 1-32. アップ・ダウン・カウント、デュアル・エッジ非対称型波形、EPWMxA 独立変調あり — アクティブ Low



- A. PWM 周期 = $2 \times \text{TBPRD} \times \text{TTBCLK}$
- B. カウンタがカウント・アップしている際とカウンタがカウント・ダウンしている際に、独立した T1 イベント・アクションを使用して EPWMxA 出力を生成します。
- C. カウンタがカウント・アップしている際とカウンタがカウント・ダウンしている際に、独立した T2 イベント・アクションを使用して EPWMxB 出力を生成します。
- D. T1 のソースとして TZ1 を選択しています。
- E. T2 のソースとして TZ2 を選択しています。

図 1-33. アップ・ダウン・カウント、T1 および T2 イベントを利用した PWM 波形生成

1.7 デッドバンド・ジェネレータ (DB) サブモジュール

図 1-34 に、ePWM 内のデッドバンド・サブモジュールを示します。

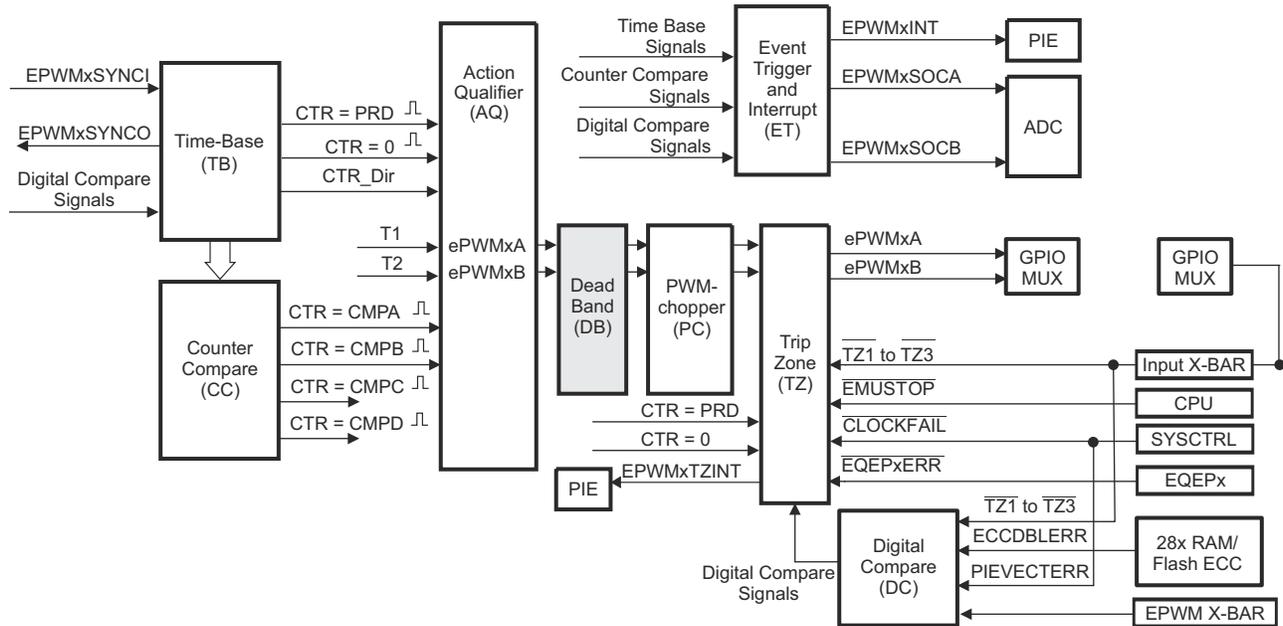


図 1-34. デッドバンド・サブモジュール

1.7.1 デッドバンド・サブモジュールの目的

「アクション・フィルタ (AQ) モジュール」セクションでは、ePWM モジュールの CMPA と CMPB の両方のリソースを使用してエッジの配置を完全に制御することで、必要なデッドバンドを生成する方法について説明しました。ただし、より伝統的なエッジ遅延ベースのデッドバンドと極性の制御が必要な場合、ここで説明するデッドバンド・サブモジュールを使用する必要があります。

デッドバンド・サブモジュールの主な特長は次のとおりです。

- 1 つの EPWMxA 入力から、デッドバンドの関係を持つ適切な信号ペア (EPWMxA と EPWMxB) を生成します。
- 信号ペアを以下のように設定します。
 - アクティブ High (AH)
 - アクティブ Low (AL)
 - アクティブ High 相補 (AHC)
 - アクティブ Low 相補 (ALC)
- 立ち上がりエッジにプログラマブルな遅延 (RED) を追加します。
- 立ち下がりエッジにプログラマブルな遅延 (FED) を追加します。
- 信号パスから完全にバイパスできます (図の点線に注意)。

1.7.2 デッドバンド・サブモジュールのその他の動作モード

タイプ 1 ePWM では、一方のチャンネルに RED を出力し、他方のチャンネルに FED を出力できます。

以下の箇条書きに、デッドバンド動作モードに関するタイプ 1 モジュールとタイプ 4 モジュールの相違点を示します。

- [図 1-35](#) の S6、S7、S8 を追加することで、RED と FED を A チャンネルと B チャンネルのどちらにも適用できるようにしています。また、RED と FED の両方を一緒に A チャンネルと B チャンネルのどちらかに適用できるため、A チャンネルに対して B チャンネルの位相をシフトできます。

注

デッドバンド・サブモジュールの追加動作モードを使用した、A チャンネルを基準とした B チャンネルの位相シフトには、RED および FED 遅延の選択と ePWMxA および ePWMxB 出力の動作デューティ・サイクルに関して制約があります。

- デッドバンド・カウンタも 14 ビットに増えました。
- デッドバンド・レジスタとデッドバンド高分解能レジスタがシャドウイングされるようになりました。
- DBREDHR および DBFEDHR レジスタを使った高分解能デッドバンド (RED および FED) が可能になりました。

注

高分解能デッドバンドが有効化されている場合、PWM チョップは無効です。

高分解能デッドバンド (RED および FED) には、半周期クロック・モード (DBCTL[HALFCYCLE] = 1) が必要です。

ePWMxA と ePWMxB の両方に、RED と FED の両方を一緒に適用することはできません。RED と FED を一緒に適用できるのは、OutA と OutB のどちらか一方のみです。

A チャンネルを基準とした B チャンネルの位相シフト : DEDB_MODE ビットを使い、立ち上がりエッジと立ち下がりエッジを位相シフト量だけ遅延させることで、PWMxA を基準として PWMxB を生成する場合。PWMxA のデューティ・サイクル値がこの位相シフト量より小さい場合、PWMxA の立ち下がりエッジは、PWMxB の遅延した立ち上がりエッジよりも優先されます。デッドバンド・モジュールに供給される電流波形のデューティ・サイクル値が、必要な位相シフト量よりも必ず大きくなるようにすることを推奨します。

タイプ 4 の動作はタイプ 3 PWM と同じですが、ePWM モジュールのタイプ 4 アクション・フィルタとデッドバンド出力は、タイプ 2 ePWM モジュールと比べると、1 TBCLK サイクル遅延します。いかなる場合でも、PWMA 信号と PWMB 信号の両方が遅延します。

シャドウ・モード：

DBRED のシャドウ・モードは、DBCTL[SHDWDBREDDMODE] ビットをセットすることで有効化され、DBFED のシャドウ・レジスタは、DBCTL [SHDWDBFEDMODE] ビットをセットすることで有効化されます。DBRED と DBFED の両方で、シャドウ・モードはデフォルトで無効化されています。

シャドウ・レジスタが有効化されている場合、DBCTL [LOADREDDMODE] および DBCTL [LOADFEDMODE] レジスタ・ビットで指定されたように、以下のイベントのいずれかでシャドウ・レジスタの内容がアクティブ・レジスタに転送されます。

- CTR = PRD：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
- CTR = ゼロ：時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- CTR = PRD かつ CTR = ゼロ

DBCTL レジスタは、シャドウイングできます。DBCTL2[SHDWDBCTLMODE] ビットをセットすると、DBCTL のシャドウ・モードが有効化されます。シャドウ・レジスタが有効化されている場合、DBCTL2[LOADDBCTLMODE] レジスタ・ビットで指定されたように、以下のイベントのいずれかでシャドウ・レジスタの内容がアクティブ・レジスタに転送されます。

- CTR = PRD：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
- CTR = ゼロ：時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- CTR = PRD かつ CTR = ゼロ

グローバル・ロードのサポート

グローバル・ロード制御機能は、グローバル・ロード構成レジスタ (GLDCFG) の適切なビットを設定することで、DBRED：DBREDHR、DBFED：DBFEDHR、DBCTL レジスタのためにも使用できます。グローバル・ロード・モードが選択されている場合、このモードが有効化されたすべてのレジスタにおいて、シャドウ・レジスタからアクティブ・レジスタへの内容の転送は、グローバル・シャドウ・アクティブ・ロード制御レジスタ (GLDCTL) の構成ビットで設定されたイベントと同じイベントで行われます。グローバル・ロード制御機能については、[セクション 1.4.7](#) で説明します。

注

DB カウンタのカウント中に DBRED/DBFED アクティブに新しいシャドウ値がロードされると、新しい DBRED/DBFED 値は、現在のエッジではなく、次の PWMx エッジにのみ影響を及ぼします。

「グローバル・シャドウ・アクティブ・ロード」が CTR = ゼロ時に行われるように設定されている場合、デッドバンド値としてゼロは使用できません。同様に、「グローバル・シャドウ・アクティブ・ロード」が CTR = PRD 時に行われるように設定されている場合、デッドバンド値として PRD は使用できません。

TBPRDHR は、グローバル・ロードと組み合わせて使うことはできません。アプリケーションで高分解能周期を変更する必要がある場合、ユーザーは ePWM の ISR から個別の周期レジスタに書き込む必要があります (その ISR は PWM スイッチング周期と同期している必要があります)。その際に、グローバル・ロード・ワンショット・ビットも書き込まれます。

1.7.3 デッドバンド・サブモジュールの動作の概要

図 1-35 に、デッドバンド・サブモジュールの設定オプションを示します。

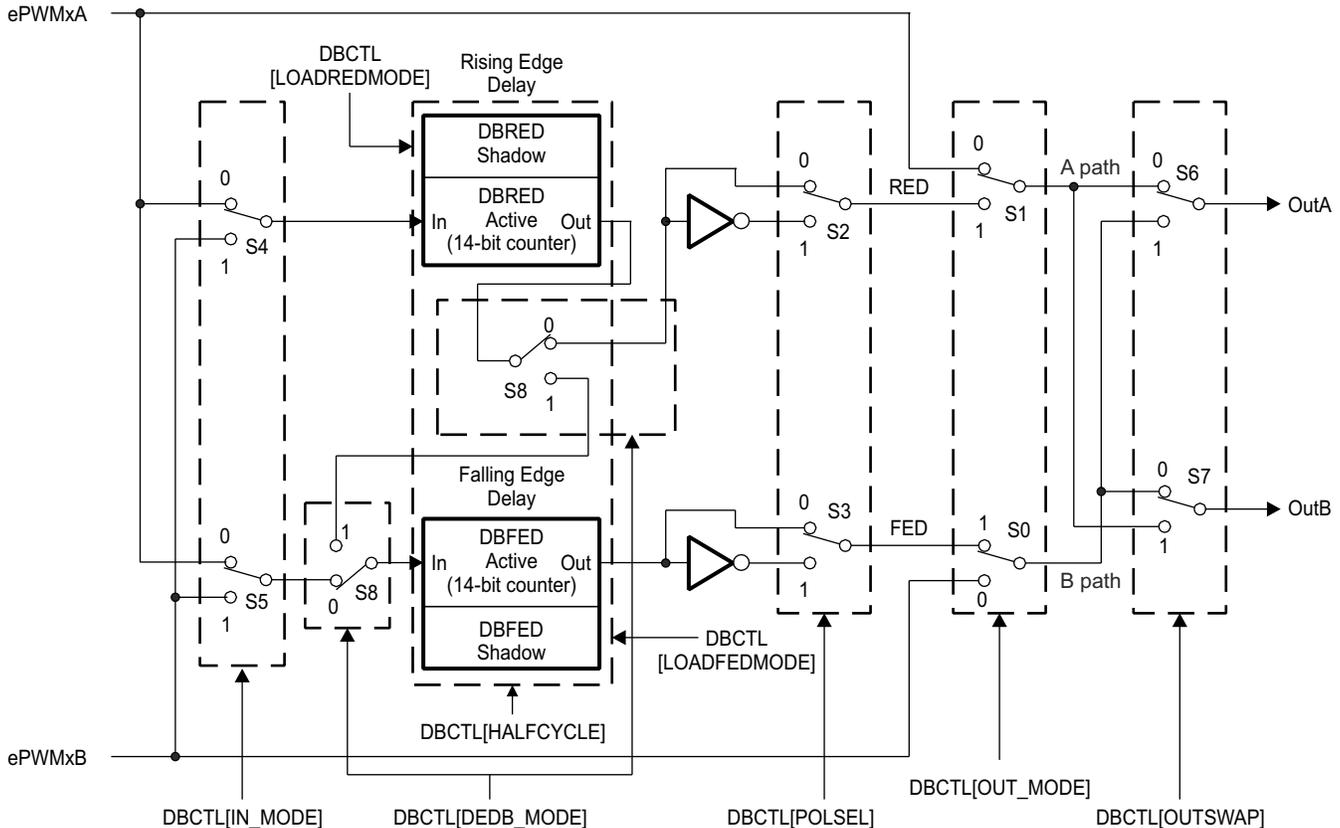


図 1-35. デッドバンド・サブモジュールの設定オプション

すべての組み合わせがサポートされていますが、すべてが代表的な使用モードであるとは限りません。表 1-9 に、いくつかの伝統的なデッドバンド設定を示します。これらのモードは、EPWMxA の入力が立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方のソースになるように、DBCTL[IN_MODE] が設定されていることを前提としています。入力信号ソースを変更することで、拡張モードまたは従来とは異なるモードを実現できます。表 1-9 に示すモードは次のカテゴリに分類されます。

- **モード 1：立ち下がりエッジ遅延 (FED) と立ち上がりエッジ遅延 (RED) の両方をバイパス：**PWM 信号パスからデッドバンド・サブモジュールを完全に無効化できます。
- **モード 2~5：伝統的なデッドバンド極性設定：**これらは、業界で利用可能なパワー・スイッチ・ゲート・ドライバが必要とするすべてのアクティブ High およびアクティブ Low モードに対応できる代表的な極性設定を示しています。図 1-36 に、これらの代表例の波形を示します。図 1-36 と等価な波形を生成するには、EPWMxA として示された信号を生成するようにアクション・フィルタ・サブモジュールを設定することに注意します。
- **モード 6：立ち上がりエッジ遅延のバイパスとモード 7：立ち下がりエッジ遅延のバイパス：**最後に、表 1-9 の最後 2 つの項目は、立ち下がりエッジ遅延 (FED) と立ち上がりエッジ遅延 (RED) のどちらかのブロックをバイパスする組み合わせを示しています。

図 1-36 に、代表例の波形 (0% < デューティ < 100%) を示します。

表 1-9. 伝統的なデッドバンドの動作モード

モード	モードの説明	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	EPWMxA と EPWMxB のパススルー (遅延なし)	X	X	0	0
2	アクティブ High 相補 (AHC)	1	0	1	1
3	アクティブ Low 相補 (ALC)	0	1	1	1
4	アクティブ High (AH)	0	0	1	1
5	アクティブ Low (AL)	1	1	1	1
6	EPWMxA 出力 = EPWMxA 入力 (遅延なし)	0 または 1	0 または 1	0	1
	EPWMxB 出力 = EPWMxA 入力、立ち下がりエッジ遅延あり				
7	EPWMxA 出力 = EPWMxA 入力、立ち上がりエッジ遅延あり	0 または 1	0 または 1	1	0
	EPWMxB 出力 = EPWMxB 入力、遅延なし				

表 1-10. 追加のデッドバンドの動作モード

モードの説明	DBCTL[DEDB-MODE]	DBCTL[OUTSWAP]	
	S8	S6	S7
EPWMxA および EPWMxB 信号は OUT-MODE ビットの設定に従います。	0	0	0
EPWMxA = OUT-MODE ビットの設定に従う A パス	0	0	1
EPWMxB = OUT-MODE ビットの設定に従う A パス (「A 信号パスの立ち上がりエッジを遅延」または「A 信号パスの遅延をバイパス」)			
EPWMxA = OUT-MODE ビットの設定に従う B パス (「B 信号パスの立ち下がりエッジを遅延」または「B 信号パスの遅延をバイパス」)	0	1	0
EPWMxB = OUT-MODE ビットの設定に従う B パス			
EPWMxA = OUT-MODE ビットの設定に従う B パス (「B 信号パスの立ち下がりエッジを遅延」または「B 信号パスの遅延をバイパス」)	0	1	1
EPWMxB = OUT-MODE ビットの設定に従う A パス (「A 信号パスの立ち上がりエッジを遅延」または「A 信号パスの遅延をバイパス」)			
A 信号パスのみで、S4 スイッチ (IN-MODE ビット) の選択に従って、EPWMxA/EPWMxB に立ち上がりエッジ遅延が適用されます。	0	X	X
B 信号パスのみで、S5 スイッチ (IN-MODE ビット) の選択に従って、EPWMxA/EPWMxB に立ち下がりエッジ遅延が適用されます。			
S4 スイッチ (IN-MODE ビット) によって選択されたソースに立ち上がりエッジ遅延と立ち下がりエッジ遅延が適用され、B 信号パスにのみ出力されます。(1)	1	X	X

- (1) このビットが 1 に設定されている場合、ユーザーは常に「A パス = InA となるように OUT_MODE をセットする」と「EPWMxA = B パスとなるように OUTSWAP ビットをセットする」のどちらかを行うことができます。それ以外の場合、EPWMxA は無効です。

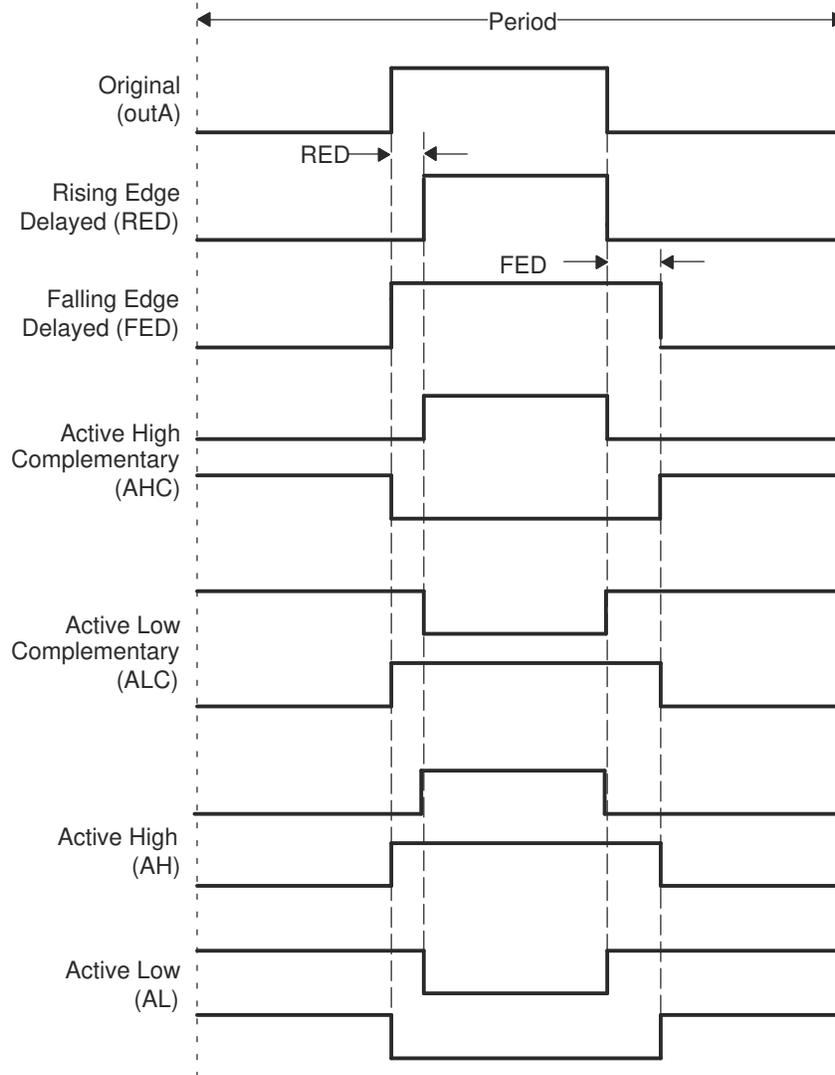


図 1-36. 代表例のデッドバンド波形 (0% < デューティ < 100%)

デッドバンド・サブモジュールは、立ち上がりエッジ遅延 (RED) と立ち下がりエッジ遅延 (FED) に対して独立した値をサポートしています。遅延量は、DBRED および DBFED レジスタを使って設定されます。これらは 10 ビット・レジスタであり、これらの値は、信号エッジが遅延される時間ベース・クロック (TBCLK) 周期の数を表します。たとえば、立ち下がりエッジ遅延と立ち上がりエッジ遅延を計算する式は次のとおりです。

$$FED = DBFED \times T_{TBCLK}$$

$$RED = DBRED \times T_{TBCLK}$$

ここで、 T_{TBCLK} は TBCLK (EPWMCLK を分周したもの) の周期です。

便宜のため、TBCLK の各種選択肢の遅延値を表 1-11 に示します。これらの遅延値の計算に使われた ePWM 入力クロック周波数は 100MHz です。

表 1-11. DBFED と DBRED の関数としてデッドバンド遅延値 (単位 : μ s)

デッドバンド値	デッドバンド遅延 (μ s)			
	DBFED、DBRED	TBCLK = EPWMCLK/1	TBCLK = EPWMCLK/2	TBCLK = EPWMCLK/4
1		0.01 μ s	0.02 μ s	0.04 μ s
5		0.05 μ s	0.10 μ s	0.20 μ s
10		0.10 μ s	0.20 μ s	0.40 μ s
100		1.00 μ s	2.00 μ s	4.00 μ s
200		2.00 μ s	4.00 μ s	8.00 μ s
400		4.00 μ s	8.00 μ s	16.00 μ s
500		5.00 μ s	10.00 μ s	20.00 μ s
600		6.00 μ s	12.00 μ s	24.00 μ s
700		7.00 μ s	14.00 μ s	28.00 μ s
800		8.00 μ s	16.00 μ s	32.00 μ s
900		9.00 μ s	18.00 μ s	36.00 μ s
1000		10.00 μ s	20.00 μ s	40.00 μ s

半周期クロックが有効化されている場合、立ち下がりエッジ遅延と立ち上がりエッジ遅延を計算するための式は次のようになります。

$$FED = DBFED \times T_{TBCLK}/2$$

$$RED = DBRED \times T_{TBCLK}/2$$

1.8 PWM チョッパ (PC) サブモジュール

PWM チョッパ・サブモジュールを使用すると、アクション・フィルタおよびデッドバンド・サブモジュールによって生成された PWM 波形を高周波数キャリア信号で変調できます。この機能は、パルス・トランスを使ったゲート・ドライバでパワー・スイッチング素子を制御する必要がある場合に重要です。

図 1-37 に、ePWM 内の PWM チョッパ・サブモジュールを示します。

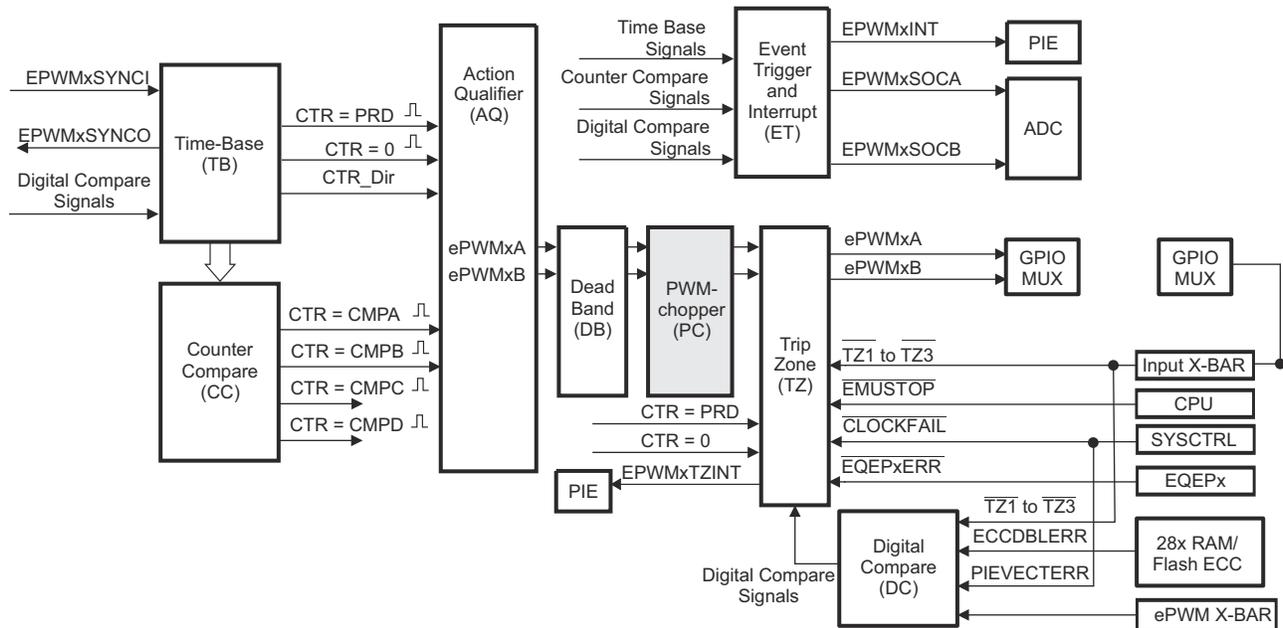


図 1-37. PWM チョッパ・サブモジュール

1.8.1 PWM チョッパ・サブモジュールの目的

PWM チョッパ・サブモジュールの主な機能は次のとおりです。

- プログラマブルなチョッピング (キャリア) 周波数
- プログラマブルな 1 つ目のパルスのパルス幅
- プログラマブルな 2 つ目以降のパルスのデューティ・サイクル
- 不要な場合、完全にバイパス可能

1.8.2 PWM チョッパ・サブモジュールの動作の概要

図 1-38 に、PWM チョッパ・サブモジュールの動作の詳細を示します。キャリア・クロックは EPWMCLK から生成されます。クロック周波数とデューティ・サイクルは、PCCTL レジスタの CHPFREQ および CHPDUTY ビットを使って制御されます。ワンショット・ブロックは、パワー・スイッチを確実にかつ高速にターンオンさせるための高エネルギーの第 1 パルスを供給する機能である一方、それ以降のパルスはパルスを持続し、パワー・スイッチを確実にオン状態に維持します。ワンショットの幅は、OSHTWTH ビットを使って設定されます。PWM チョッパ・サブモジュールは、CHPEN ビットを使って完全に無効化 (バイパス) できます。

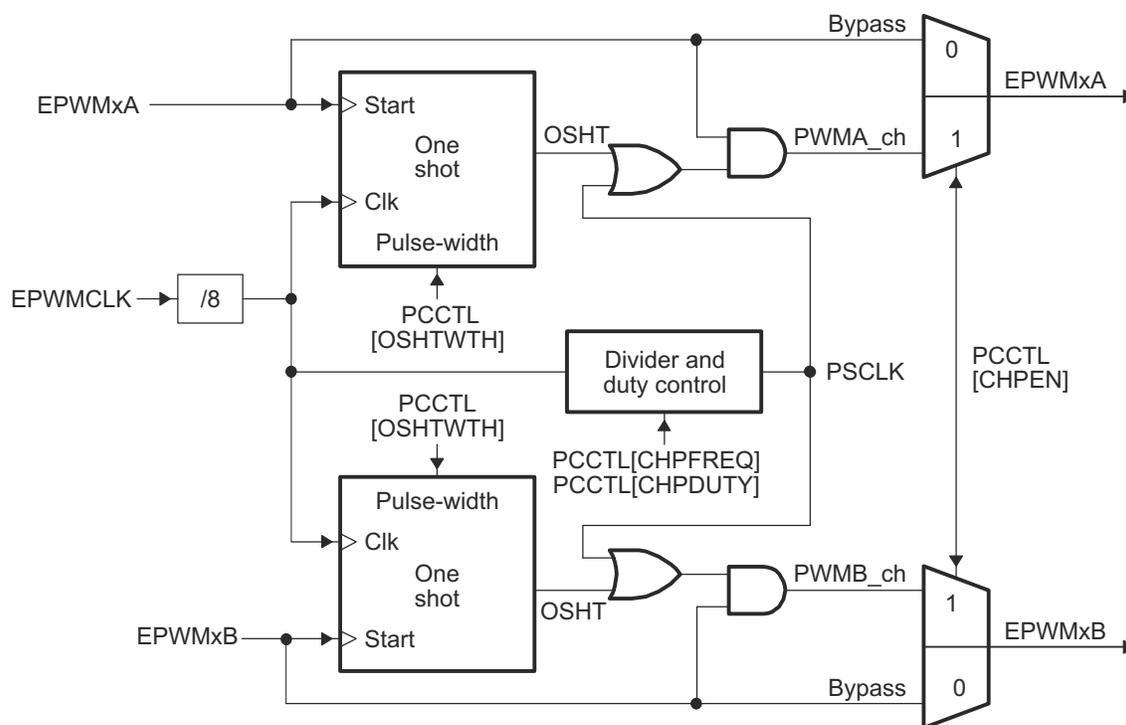


図 1-38. PWM チョッパ・サブモジュールの動作の詳細

1.8.3 波形

図 1-39 に、チョッピング動作のみの波形の概略図を示します。ワンショットおよびデューティ・サイクル制御は示されていません。ワンショットおよびデューティ・サイクル制御の詳細については、以下のセクションで説明します。

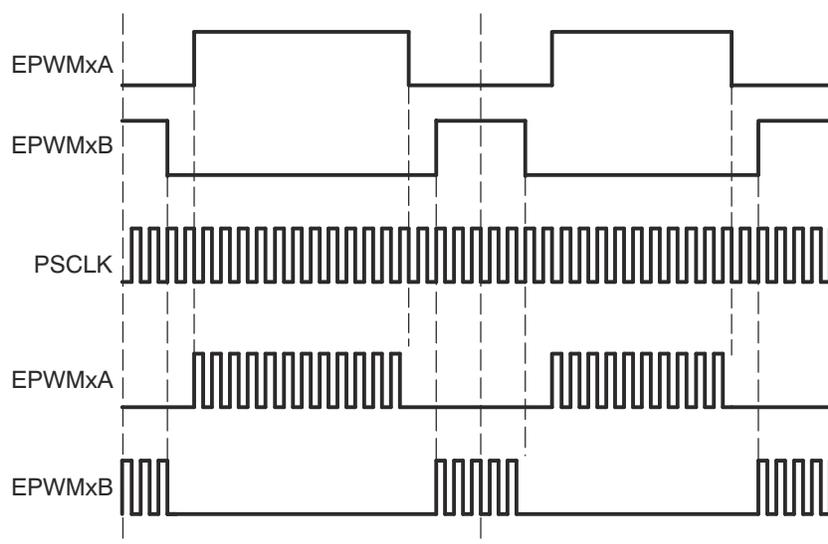


図 1-39. チョッピング動作のみを説明する単純な PWM チョッパ・サブモジュール波形

1.8.3.1 ワンショット・パルス

最初のパルスの幅は、16種類の可能なパルス幅値のいずれかに設定できます。最初のパルスの幅または周期は、次の式で与えられます。

$$T_{1\text{stpulse}} = T_{\text{EPWMCLK}} \times 8 \times \text{OSHTWTH}$$

ここで、 T_{EPWMCLK} はシステム・クロック (EPWMCLK) の周期、OSHTWTH は4ビットの制御ビット (値は1~16) です。

図 1-40 に、最初のパルスとそれに続く維持パルスを示し、表 1-12 に、EPWMCLK = 80MHz で使用可能なパルス幅の値を示します。

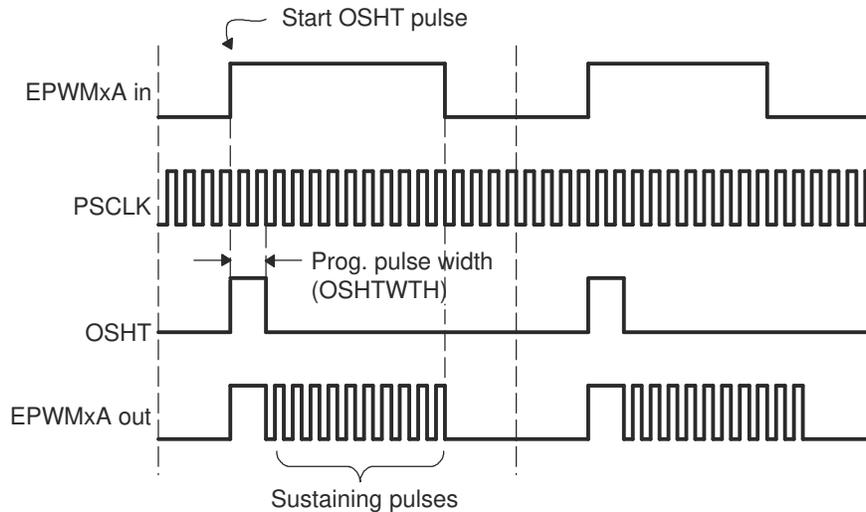


図 1-40. 最初のパルスとそれに続く維持パルスを説明する PWM チョップパ・サブモジュール波形

表 1-12. EPWMCLK = 80MHz で使用可能なパルス幅の値

OSHTWTH (16 進数)	パルス幅 (ns)
0	100
1	200
2	300
3	400
4	500
5	600
6	700
7	800
8	900
9	1000
A	1100
B	1200
C	1300
D	1400
E	1500
F	1600

1.8.3.2 デューティ・サイクル制御

パルス・トランスを使ったゲート駆動設計では、トランスと関連回路の磁氣的性質または特性を把握する必要があります。飽和はそのような検討事項の1つです。ゲート駆動設計を支援するため、2つ目以降のパルスのデューティ・サイクルはプログラマブルになっています。これらの維持パルスにより、パワー・スイッチのゲートでの適切な駆動強度と極性がオン期間中確実に維持されます。そのため、プログラマブルなデューティ・サイクルのおかげで、ソフトウェア制御を使って設計を調整または最適化できます。

図 1-41 に、CHPDUTY ビットのプログラミングにより可能なデューティ・サイクル制御を示します。12.5% ~ 87.5% の範囲の 7 種類のデューティ比のいずれかを選択できます。

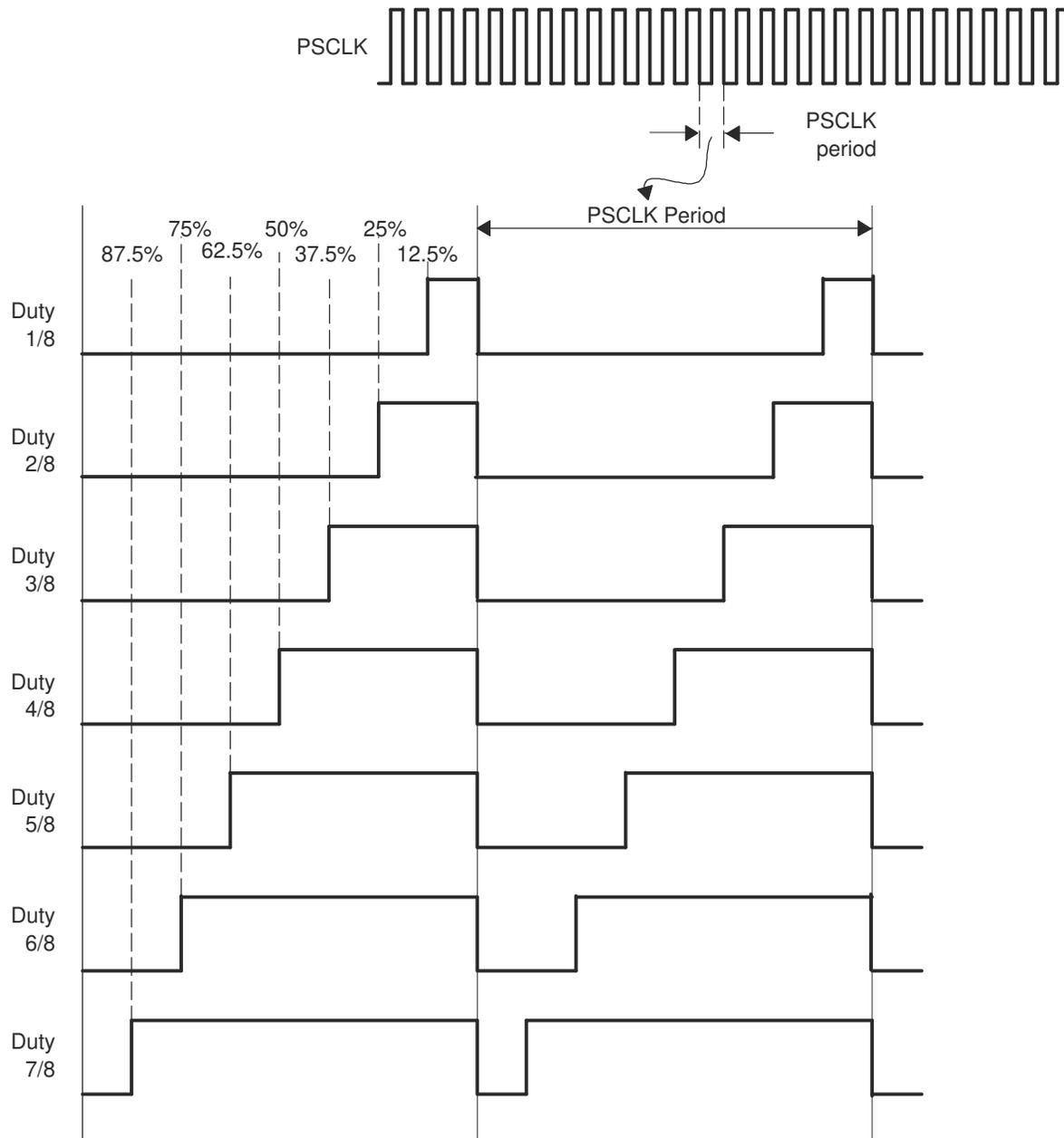


図 1-41. 維持パルスのパルス幅 (デューティ・サイクル) 制御を説明する PWM チョップ・サブモジュール波形

1.9 トリップ・ゾーン (TZ) サブモジュール

各 ePWM モジュールは 6 つの \overline{TZn} 信号 ($\overline{TZ1} \sim \overline{TZ6}$) に接続されています。 $\overline{TZ1} \sim \overline{TZ3}$ には GPIO マルチプレクサから信号が供給されます。 $\overline{TZ4}$ には、EQEP モジュールを備えたデバイスの反転された EQEPxERR 信号が供給されます。 $\overline{TZ5}$ はシステム・クロック障害ロジックに接続されており、 $\overline{TZ6}$ には CPU の EMUSTOP 出力が供給されます。これらの信号は外部のフォルトまたはトリップ条件を示しています。フォルトが発生した際にそれに応じて ePWM 出力が応答するように、ePWM 出力を設定できます。

図 1-42 に、ePWM 内のトリップ・ゾーン・サブモジュールを示します。

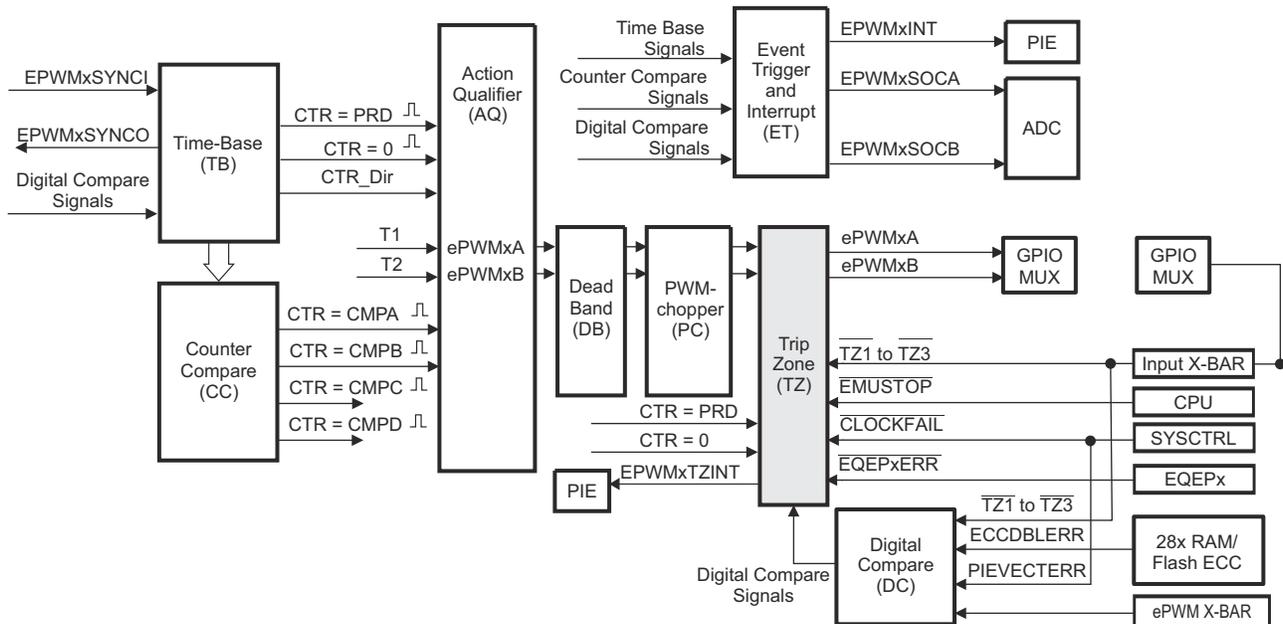


図 1-42. トリップ・ゾーン・サブモジュール

1.9.1 トリップ・ゾーン・サブモジュールの目的

トリップ・ゾーン・サブモジュールの主な機能は次のとおりです。

- トリップ入力 ($\overline{TZ1} \sim \overline{TZ6}$) は、任意の ePWM モジュールに柔軟に割り当てることができます。
- フォルト条件が発生すると、出力 EPWMxA および EPWMxB を以下のいずれかに強制的に設定できます。
 - High
 - Low
 - ハイ・インピーダンス
 - アクションは何も実行されません。
- 深刻な短絡または過電流条件に備えたワンショット・トリップ (OSHT) のサポート。
- 電流制限動作のためのサイクル・バイ・サイクル (CBC) トリップのサポート。
- 内蔵アナログ・コンパレータ・モジュール出力の状態と $\overline{TZ1} \sim \overline{TZ3}$ 信号に基づくデジタル比較トリップ (DC) のサポート。
- トリップ・ゾーン入力とデジタル比較 (DC) サブモジュール DCAEVT1/2 または DCBEVT1/2 強制イベントはどちらも、ワンショット動作とサイクル・バイ・サイクル動作のどちらかに割り当てることができます。
- 割り込み生成は、すべてのトリップ・ゾーン入力が可能です。
- ソフトウェア強制トリップもサポートされています。
- トリップ・ゾーン・サブモジュールが不要な場合、トリップ・ゾーン・サブモジュールを完全にバイパスできます。

1.9.2 トリップ・ゾーン・サブモジュールの動作の概要

以下のセクションでは、トリップ・ゾーン・サブモジュールの動作の概要と設定オプションについて説明します。

トリップ・ゾーン信号 $\overline{TZ1} \sim \overline{TZ6}$ (総称して \overline{TZn} とも呼びます) はアクティブ Low の入力信号です。これらの信号のいずれかが Low になった場合、または TZDCSEL レジスタ・イベントの選択に基づいて DCAEVT1/2 または DCBEVT1/2 の強制が行われた場合、それはトリップ・イベントが発生したことを示します。各 ePWM モジュールは、各トリップ・ゾーン信号または DC イベントを無視するように、または使用するよう個別に設定できます。特定の ePWM モジュールで使用されるトリップ・ゾーン信号または DC イベントは、その特定の ePWM モジュールの TZSEL レジスタによって決定されます。トリップ・ゾーン信号は、ePWM クロック (EPWMCLK) と同期させ、GPIO マルチプレクサ・ブロック内でデジタル・フィルタ処理されるように設定することも、そのように設定しないこともできます。ePWM モジュールのフォルト条件をトリガするには、 \overline{TZn} 入力での最小 $3 \cdot TBCLK$ の Low パルス幅で十分です。パルス幅がこれよりも小さい場合、CBC または OST ラッチでこのトリップ条件をラッチできません。非同期トリップを使うと、何らかの理由でクロックが失われた場合でも、 \overline{TZn} 入力に存在する有効なイベントによって出力が確実にトリップされます。GPIO またはペリフェラルは、適切に設定する必要があります。詳細については、「システムの制御と割り込み」の章を参照してください。

各 \overline{TZn} 入力は、ePWM モジュールに対して、サイクル・バイ・サイクル・トリップ・イベントとワンショット・トリップ・イベントのどちらかを生成するように個別に設定できます。DCAEVT1 および DCBEVT1 イベントは、ePWM モジュールを直接トリップさせるようにも、モジュールに対してワンショット・トリップ・イベントを生成するようにも設定できます。同様に、DCAEVT2 および DCBEVT2 イベントは、ePWM モジュールを直接トリップさせるようにも、モジュールに対してサイクル・バイ・サイクル・トリップ・イベントを生成するようにも設定できます。この設定は、TZSEL[DCAEVT1/2]、TZSEL[DCBEVT1/2]、TZSEL[CBCn]、TZSEL[OSHTn] の各制御ビット (n はトリップ入力に対応) によってそれぞれ決定されます。

• サイクル・バイ・サイクル (CBC) :

サイクル・バイ・サイクル・トリップ・イベントが発生すると、TZCTL[TZA] および TZCTL[TZB] ビットで設定されたアクションが EPWMxA および EPWMxB 出力で即座に実行されます。表 1-13 に、実行可能な各種アクションを示します。TZCTL2 レジスタのビットを適切に設定することで、カウンタがカウント・アップしている間、またはカウント・ダウンしている間のイベントの発生に基づいて、独立したアクションを指定できます。TZCTL2 レジスタで指定されたアクションは、TZCTL2 の ETZE ビットがセットされている場合にのみ効力を生じます。

また、TZEINT レジスタと PIE ペリフェラルで有効化されている場合、サイクル・バイ・サイクル・トリップ・イベントが発生すると、サイクル・バイ・サイクル・トリップ・イベント・フラグ (TZFLG[CBC]) がセットされ、EPWMx_TZINT 割り込みが生成されます。CBC イベントの原因となったイベントに対応するフラグは、レジスタ TZCBCFLG でもセットされます。

TZEINT レジスタを使って CBC 割り込みが有効化されており、TZSEL レジスタを使って DCAEVT2 または DCBEVT2 が CBC トリップ・ソースとして選択されている場合、CBC 機構による DC イベント・トリガ割り込みのように、DCAEVT2 割り込みも DCBEVT2 割り込みも TZEINT レジスタで有効化する必要はありません。

トリップ・イベントがもはや存在しなくなった場合、その入力で指定された条件は、TZCLR[CBCPULSE] による選択に基づいて自動的にクリアされます。したがって、このモードでは、PWM サイクルごとにトリップ・イベントがクリアまたはリセットされます。TZFLG[CBC] および TZCBCFLG フラグ・ビットは、TZCLR[CBC] および TZCBCCLR フラグ・ビットに書き込むことによって個別にクリアされるまで、セットされたまま維持されます。TZFLG[CBC] および TZCBCFLG レジスタ・ビットがクリアされた際にサイクル・バイ・サイクル・トリップ・イベントが引き続き存在する場合、これらのビットは再び即座にセットされます。

• ワンショット (OSHT) :

ワンショット・トリップ・イベントが発生すると、TZCTL[TZA] および TZCTL[TZB] ビットで設定されたアクションが EPWMxA および EPWMxB 出力で即座に実行されます。表 1-13 に、実行可能な各種アクションを示します。TZCTL2 レジスタのビットを適切に設定することで、カウンタがカウント・アップして

いる間とカウント・ダウンしている間のイベントの発生に基づいて、独立したアクションを指定できます。TZCTL2 レジスタで指定されたアクションは、TZCTL2 の ETZE ビットがセットされている場合にのみ効力を生じます。

また、TZEINT レジスタと PIE ペリフェラルで有効化されている場合、ワンショット・トリップ・イベントが発生すると、ワンショット・トリップ・イベント・フラグ (TZFLG[OST]) がセットされ、EPWMx_TZINT 割り込みが生成されます。OST イベントの原因となったイベントに対応するフラグは、レジスタ TZOSTFLG でもセットされます。ワンショット・トリップ条件は、TZCLR[OST] ビットに書き込むことで個別にクリアする必要があります。必要に応じて、TZOSTFLG レジスタの対応するビットに個別に書き込むことで、TZOSTFLG レジスタ・ビットをクリアできます。

TZEINT レジスタを使ってワンショット割り込みが有効化されており、TZSEL レジスタを使って DCAEVT1 または DCBEVT1 が OSHT トリップ・ソースとして選択されている場合、OSHT 機構による DC イベント・トリガ割り込みのように、DCAEVT1 割り込みも DCBEVT1 割り込みも TZEINT レジスタで有効化する必要はありません。

注

OST の TRIPIN ソースが非アクティブになったことを確認した後、TZFLG および TZOSTFLG フラグをクリアします。さもないと、割り込みが有効化されている場合、フラグがクリアされるタイミングによっては、OST 割り込みが発生し、OST フラグがゼロになる可能性があります。

• デジタル比較イベント (DCAEVT1/2 および DCBEVT1/2) :

デジタル比較 DCAEVT1/2 または DCBEVT1/2 イベントは、TZDCSEL レジスタで選択された DCAH/DCAL 信号と DCBH/DCBL 信号の組み合わせに基づいて生成されます。DCAH/DCAL および DCBH/DCBL 信号のソースとなる信号は、DCTRISEL レジスタを使って選択され、トリップ・ゾーン入力ピンとアナログ・コンパレータ CMPSSx 信号のどちらかに指定できます。デジタル比較サブモジュール信号の詳細については、[セクション 1.11](#) を参照してください。

デジタル比較イベントが発生すると、TZCTL[DCAEVT1/2] および TZCTL[DCBEVT1/2] ビットで設定されたアクションが EPWMxA および EPWMxB 出力で即座に実行されます。[表 1-13](#) に、実行可能な各種アクションを示します。TZCTLDCA および TZCTLDCB レジスタのビットを適切に設定することで、カウンタがカウント・アップしている間とカウント・ダウンしている間のイベントの発生に基づいて、独立したアクションを指定できます。TZCTLDCA および TZCTLDCB レジスタで指定されたアクションは、TZCTL2 の ETZE ビットがセットされている場合にのみ効力を生じます。

また、TZEINT レジスタと PIE ペリフェラルで有効化されている場合、関連する DC トリップ・イベント・フラグ (TZFLG[DCAEVT1/2]、TZFLG[DCBEVT1/2]) がセットされ、EPWMx_TZINT 割り込みが生成されません。

DC トリップ・イベントがもはや存在しなくなった場合、そのピンで指定された条件は自動的にクリアされます。TZFLG[DCAEVT1/2] または TZFLG[DCBEVT1/2] フラグ・ビットは、TZCLR[DCAEVT1/2] または TZCLR[DCBEVT1/2] ビットに書き込むことによって個別にクリアされるまで、セットされたまま維持されます。TZFLG[DCAEVT1/2] または TZFLG[DCBEVT1/2] フラグがクリアされた際に DC トリップ・イベントが引き続き存在する場合、そのフラグは再び即座にセットされます。

トリップ・イベントが発生した際に実行されるアクションは、TZCTL、TZCTL2、TZCTLDCA、TZCTLDCB レジスタ・ビット・フィールドを使って、各 ePWM 出力ピンに対して個別に設定できます。[表 1-13](#) に示す各種のアクションを、トリップ・イベント時に実行できます。

表 1-13. トリップ・イベント時に実行可能なアクション

TZCTL レジスタ・ビット・フィールドの設定	EPWMxA および EPWMxB	コメント
0,0	ハイ・インピーダンス	トリップされます。
0,1	High 状態に強制	トリップされます。
1,0	Low 状態に強制	トリップされます。
1,1	変更なし	何もしません。 出力は変化しません。

Example 1-1. トリップ・ゾーンの設定
シナリオ A :

$\overline{TZ1}$ のワンショット・トリップ・イベントにより、EPWM1A と EPWM1B の両方が Low にプルされ、EPWM2A と EPWM2B は High に強制されます。

- ePWM1 レジスタを次のように設定します。
 - TZSEL[OSHT1] = 1 : ePWM1 のワンショット・イベント・ソースとして $\overline{TZ1}$ を有効化します。
 - TZCTL[TZA] = 2 : トリップ・イベント時に EPWM1A は Low に強制されます。
 - TZCTL[TZB] = 2 : トリップ・イベント時に EPWM1B は Low に強制されます。
- ePWM2 レジスタを次のように設定します。
 - TZSEL[OSHT1] = 1 : ePWM2 のワンショット・イベント・ソースとして $\overline{TZ1}$ を有効化します。
 - TZCTL[TZA] = 1 : トリップ・イベント時に EPWM2A は High に強制されます。
 - TZCTL[TZB] = 1 : トリップ・イベント時に EPWM2B は High に強制されます。

シナリオ B :

$\overline{TZ5}$ でサイクル・バイ・サイクル・イベントが発生すると、EPWM1A と EPWM1B の両方が Low にプルされます。

$\overline{TZ1}$ または $\overline{TZ6}$ でワンショット・イベントが発生すると、EPWM2A はハイ・インピーダンス状態になります。

- ePWM1 レジスタを次のように設定します。
 - TZSEL[CBC5] = 1 : ePWM1 のサイクル・バイ・サイクル・イベント・ソースとして $\overline{TZ5}$ を有効化します。
 - TZCTL[TZA] = 2 : トリップ・イベント時に EPWM1A は Low に強制されます。
 - TZCTL[TZB] = 2 : トリップ・イベント時に EPWM1B は Low に強制されます。
- ePWM2 レジスタを次のように設定します。
 - TZSEL[OSHT1] = 1 : ePWM2 のワンショット・イベント・ソースとして $\overline{TZ1}$ を有効化します。
 - TZSEL[OSHT6] = 1 : ePWM2 のワンショット・イベント・ソースとして $\overline{TZ6}$ を有効化します。
 - TZCTL[TZA] = 0 : トリップ・イベント時に EPWM2A はハイ・インピーダンス状態になります。
 - TZCTL[TZB] = 3 : EPWM2B はトリップ・イベントを無視します。

注

GPIO と入力クロスバー / ePWM クロスバー・オプションを設定する際、クロスバー入力の選択を変更すると、不要なイベントが発生する可能性があることに注意します。したがって、ePWM トリップ・ゾーンを有効化する前に、GPIO とクロスバー入力構成を設定するのが理想的です。ePWM トリップ・ゾーンが有効化されている間に GPIO / クロスバーの設定を変更する必要がある場合、ユーザーは TZSEL レジスタをクリアし、入力クロスバーの選択が変更された後にトリップ選択 (TZSEL) を再設定することで、トリップを無効化できます。

1.9.3 トリップ・ イベント割り込みの生成

図 1-43 と図 1-44 に、トリップ・ ゾーン・ サブモジュールの制御ロジックと割り込みロジックをそれぞれ示します。DCAEVT1/2 および DCBEVT1/2 信号については、セクション 1.11 で詳細に説明します。

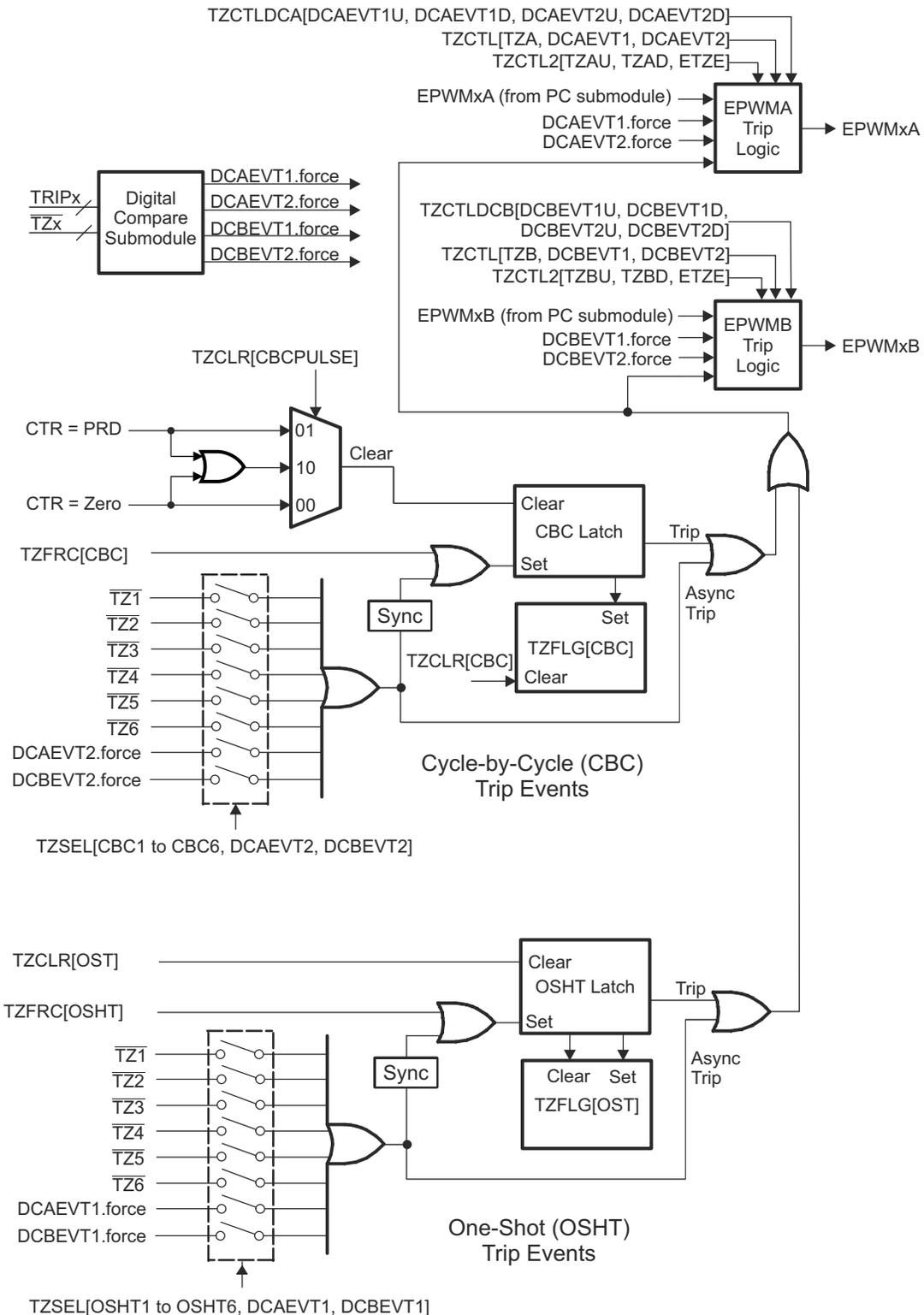


図 1-43. トリップ・ ゾーン・ サブモジュール・ モード制御ロジック

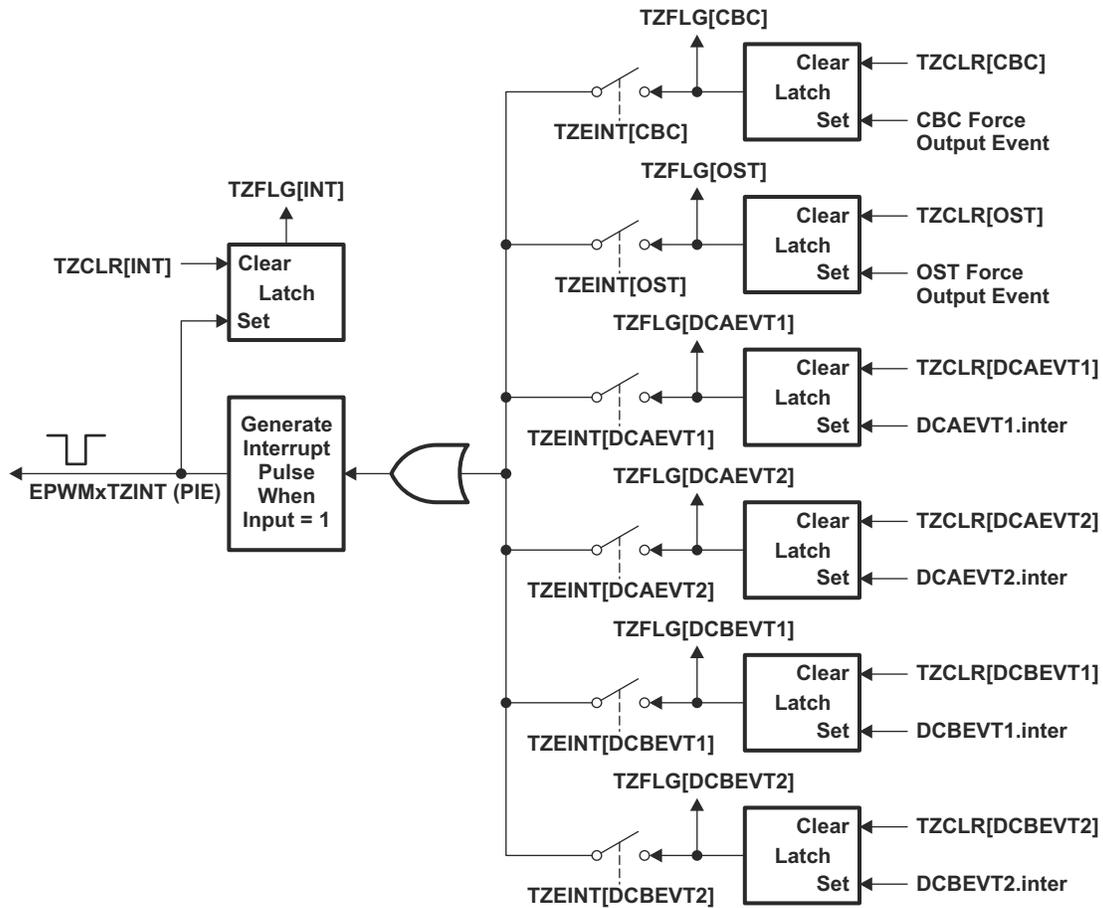


図 1-44. トリップ・ゾーン・サブモジュール割り込みロジック

CBC、OST、DCxEV_{Ty} のためのこれらの個別フラグは、EPWMxTZINT 割り込みのソースを検出するために使用できます。EPWMxTZINT 割り込みを生成するために複数のソースを使用する場合、フラグの読み出しとクリアは、特定のイベントに基づいて各種アクションを実行します。

1.10 イベント・トリガ (ET) サブモジュール

イベント・トリガ・サブモジュールの主な機能は次のとおりです。

- 時間ベース、カウンタ比較、デジタル比較サブモジュールによって生成されたイベント入力を受け取ります。
- アップ/ダウン・イベントのクオリフィケーションのために時間ベースの方向情報を使用します。
- 以下のタイミングで割り込み要求と ADC 変換開始を発行するため、プリスケアラ・ロジックを使用します。
 - イベントごと
 - 2 回のイベントごと
 - 最大 15 回のイベントごと
- イベント・カウンタおよびフラグを使用して、イベント生成を完全に可視化します。
- 割り込みと ADC 変換開始のソフトウェアによる強制が可能です。

イベント・トリガ・サブモジュールは、選択されたイベントが発生した際に CPU への割り込みと ADC への変換開始パルスを生成するため、時間ベース・サブモジュール、カウンタ比較サブモジュール、デジタル比較サブモジュールによって生成されたイベントを管理します。

図 1-45 に、ePWM 内のイベント・トリガ・サブモジュールを示します。

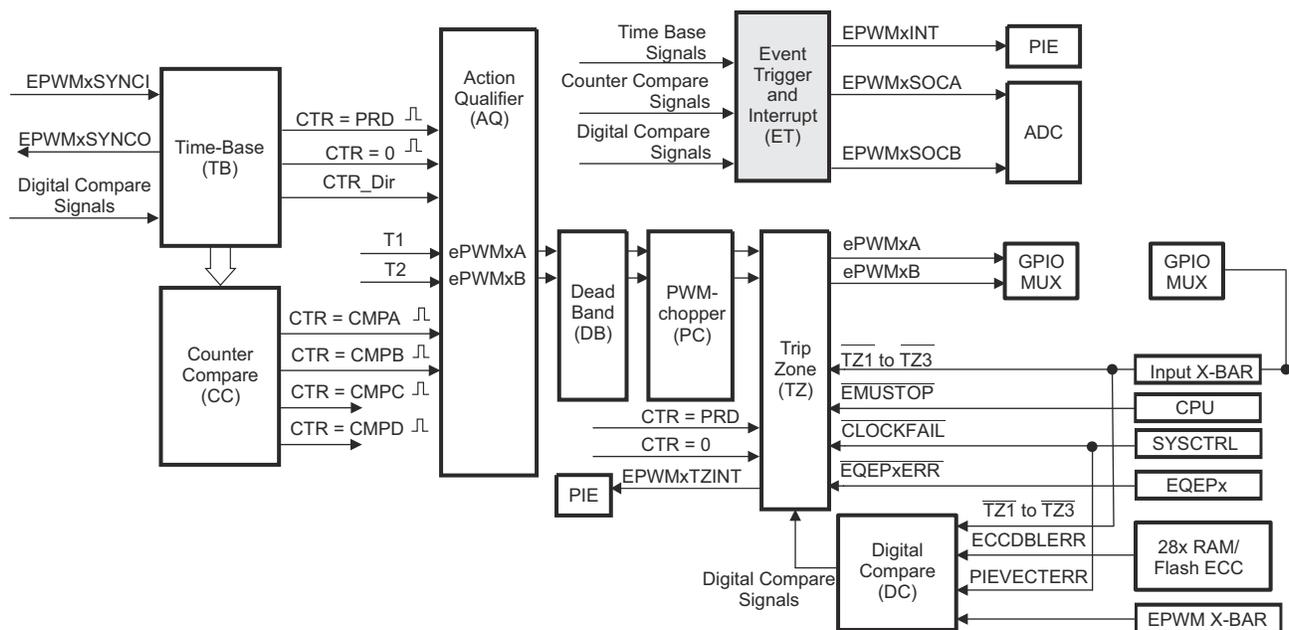


図 1-45. イベント・トリガ・サブモジュール

1.10.1 ePWM イベント・トリガ・サブモジュールの動作の概要

イベント・トリガ・サブモジュールは各種のイベント条件 (図 1-46 の左側に入力として示します) を監視します。また、割り込み要求または ADC 変換開始を発行する前にこれらのイベントをプリスケールするように設定できます。イベント・トリガ・プリスケール・ロジックは、以下のタイミングで割り込み要求と ADC 変換開始を発行できます。

- イベントごと
- 2 回のイベントごと
- 最大 15 回のイベントごと

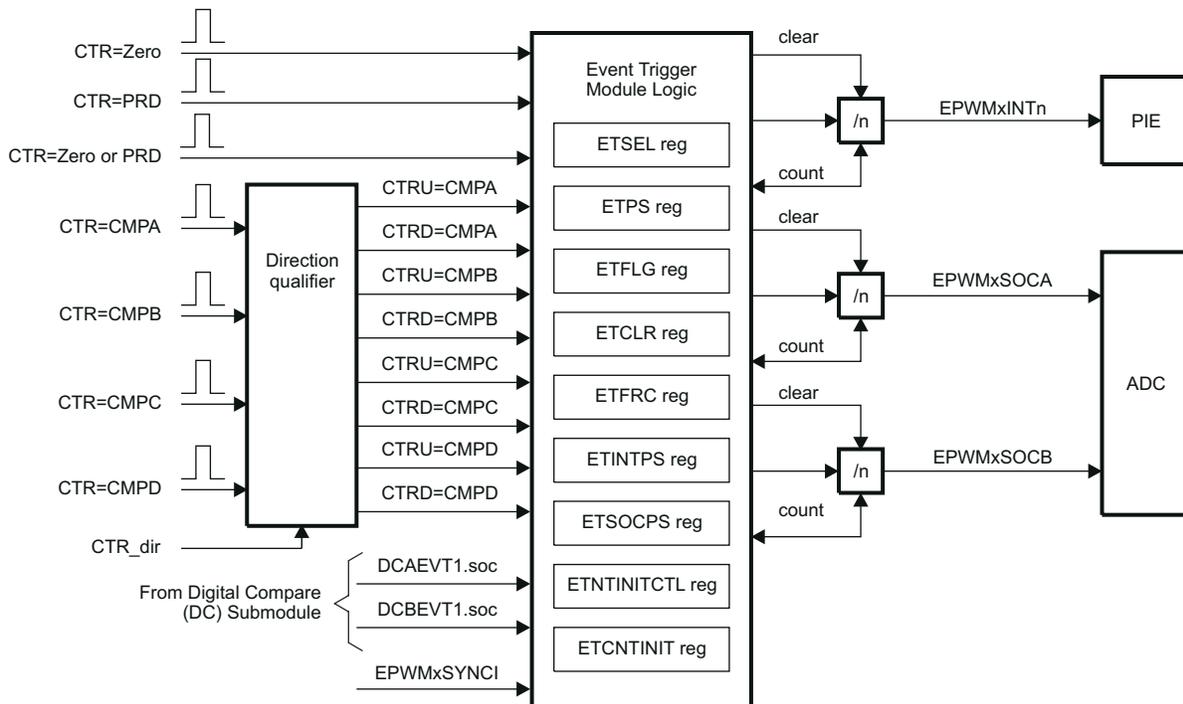


図 1-46. イベント入力とプリスケール済み出力を説明するイベント・トリガ・サブモジュールのブロック図

- ETSEL - 割り込みをトリガし、または ADC 変換を開始するイベントの種類を選択します。
- ETPS - 上記のイベント・プリスケール・オプションを設定します。
- ETFLG - 選択され、かつプリスケールされたイベントのステータスを示すフラグ・ビットです。
- ETCLR - これらのビットを使用すると、ソフトウェアを使用して ETFLG レジスタ内のフラグ・ビットをクリアできます。
- ETFRC - これらのビットを使用すると、イベントをソフトウェアで強制的にトリガできます。デバッグまたはソフトウェアによる介入に便利です。
- ETINTPS - 割り込みイベント・プリスケール・オプションを設定します。最大 15 のイベントのカウンタと周期をサポートしています。
- ETSOCPs - SOC イベントのプリスケール・オプションを設定します。最大 15 のイベントのカウンタと周期をサポートしています。
- ETNTINITCTL - これらのビットを使用すると、同期イベントまたはソフトウェア強制を使った ETCNTINIT による初期化を実行できます。
- ETCNTINIT - これらのビットを使用すると、同期イベント (またはソフトウェア強制) 時に、ユーザーが設定した値で INT/SOCA/SOCB カウンタを初期化できます。

図 1-47、図 1-48、図 1-49 に、各種レジスタ・ビットと割り込みおよび ADC 変換開始ロジックとの関係の詳細を示します。

図 1-47 に、イベント・トリガの割り込み生成ロジックを示します。割り込み周期 (ETPS[INTPRD]) ビットは、割り込みパルスが生成されるのに必要なイベント数を指定します。選択肢は以下のとおりです。

- 割り込みを生成しません。
- 1回のイベントごとに1回の割り込みを生成します。
- 2回のイベントごとに1回の割り込みを生成します。
- 3回のイベントごとに1回の割り込みを生成します。

ETPS[INTPSSSEL] ビットでの選択によって、ETINTPS レジスタの INTCNT2 および INTPRD2 ビット・フィールドがイベントの頻度 (0~15 回のイベントごとに1回の割り込み) を決定するかどうかが決まります。

割り込みの原因となり得るイベントは、割り込み選択ビット (ETSEL[INTSEL]、ETSEL[INTSELCMP]) によって設定されます。そのイベントは以下のいずれかです。

- 時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)
- 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)
- 時間ベース・カウンタがゼロまたは周期と一致 (TBCTR = 0x00 || TBCTR = TBPRD)
- 時間ベース・カウンタが比較 A レジスタ (CMPA) と一致 (そのタイムがインクリメントしている場合)
- 時間ベース・カウンタが比較 A レジスタ (CMPA) と一致 (そのタイムがデクリメントしている場合)
- 時間ベース・カウンタが比較 B レジスタ (CMPB) と一致 (そのタイムがインクリメントしている場合)
- 時間ベース・カウンタが比較 B レジスタ (CMPB) と一致 (そのタイムがデクリメントしている場合)
- 時間ベース・カウンタが比較 C レジスタ (CMPC) と一致 (そのタイムがインクリメントしている場合)
- 時間ベース・カウンタが比較 C レジスタ (CMPC) と一致 (そのタイムがデクリメントしている場合)
- 時間ベース・カウンタが比較 D レジスタ (CMPD) と一致 (そのタイムがインクリメントしている場合)
- 時間ベース・カウンタが比較 D レジスタ (CMPD) と一致 (そのタイムがデクリメントしている場合)

発生したイベントの数は、ETPS[INTPSSSEL] を使って行われた選択に基づいて、割り込みイベント・カウンタ ETPS[INTCNT] または ETINTPS[INTCNT2] レジスタ・ビットから読み出すことができます。つまり、指定されたイベントが発生すると、ETPS[INTCNT] または ETINTPS[INTCNT2] ビットが、ETPS[INTPRD] または ETINTPS[INTPRD2] (ETPS[INTPSSSEL] で行われた選択によって再度決定されます) で指定された値に達するまでインクリメントされます。ETPS[INTCNT] = ETPS[INTPRD] になると、カウンタはカウントを停止し、カウンタ出力はセットされます。このカウンタは、PIE に割り込みが発行されたときのみクリアされます。

ETPS[INTCNT] が ETPS[INTPRD] に達すると、以下の動作が行われます。以下の動作は、ETINTPS[INTCNT2] と ETINTPS[INTPRD2] にも適用できます。

- 割り込みが有効化されており (ETSEL[INTEN] = 1)、割り込みフラグがクリアされている (ETFLG[INT] = 0) 場合、割り込みパルスが生成され、割り込みフラグがセットされ (ETFLG[INT] = 1)、イベント・カウンタがクリアされます (ETPS[INTCNT] = 0)。カウンタはイベントのカウントを再開します。
- 割り込みが無効化されており (ETSEL[INTEN] = 0)、割り込みフラグがセットされている (ETFLG[INT] = 1) 場合、カウンタが周期値に達すると (ETPS[INTCNT] = ETPS[INTPRD])、カウンタはイベントのカウントを停止します。
- 割り込みが有効化されているが、割り込みフラグがすでにセットされている場合、ENTFLG[INT] フラグがクリアされるまで、カウンタは出力を High に保持します。これにより、1つの割り込みが処理されている間、1つの割り込みを保留できます。

INTPRD ビットに 0 を書き込むと、カウンタが自動的にクリアされ (INTCNT = 0)、カウンタ出力がリセットされます (そのため、割り込みは生成されません)。ETFRC[INT] ビットに 1 を書き込むと、イベント・カウンタ INTCNT がインクリメントされます。INTCNT = INTPRD になると、このカウンタは前述のように動作します。INTPRD = 0 の場合、カウンタは無効化されるため、イベントは何も検出されず、ETFRC[INT] ビットも無視されます。ETINTPS[INTCNT2] と ETINTPS[INTPRD2] にも同じことが当てはまります。

INTCNT と INTPRD を使用する場合、前述の定義は、1回のイベントごと、2回のイベントごと、3回のイベントごとのいずれかに1回の割り込みが生成されることを意味します。INTCNT2 と INTPRD2 を使用する場合、最大 15 回のイベントごとに1回の割り込みが生成されます。

INTCNT2 の値は、ETCNTINITCTL[INTINITEN] による選択に基づいて、ETCNTINIT[INTINIT] の値で初期化できます。ETCNTINITCTL[INTINITEN] がセットされている場合、同期イベントまたはソフトウェア強制時の、ETCNTINIT[INTINIT] の内容による INTCNT2 カウンタの初期化は、ETCNTINITCTL[INTINITFRC] によって決定されます。

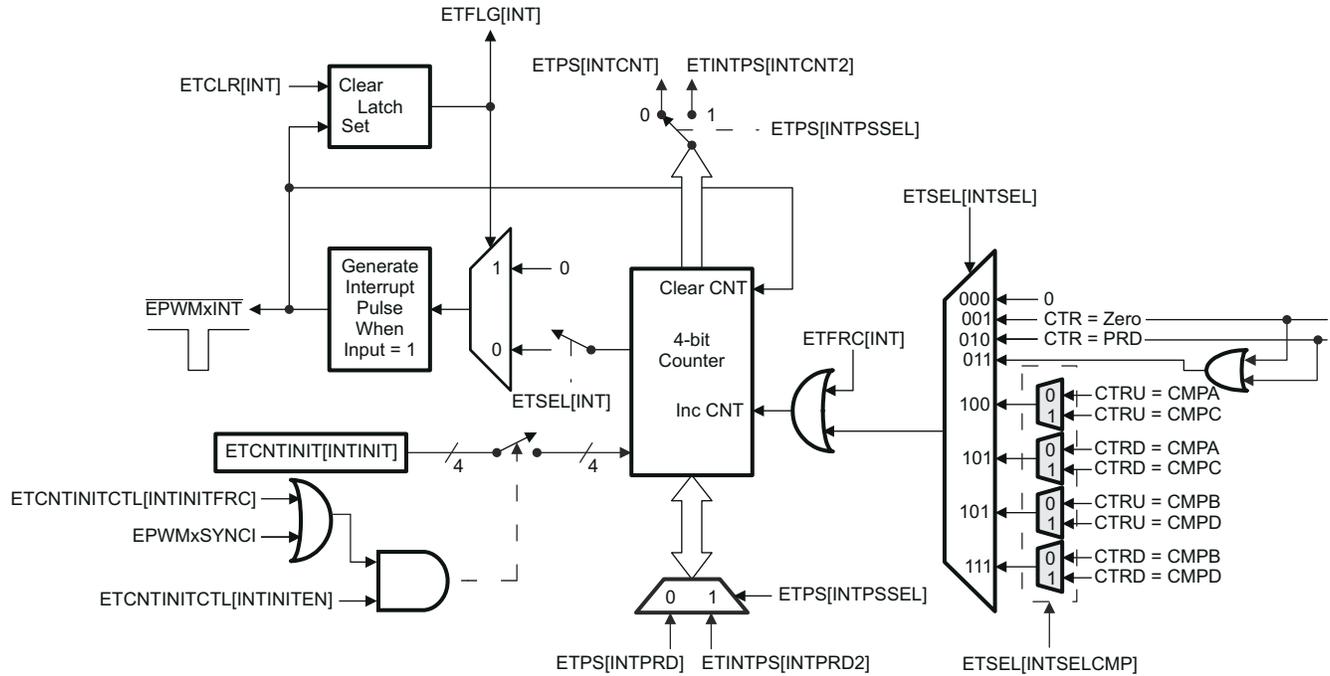
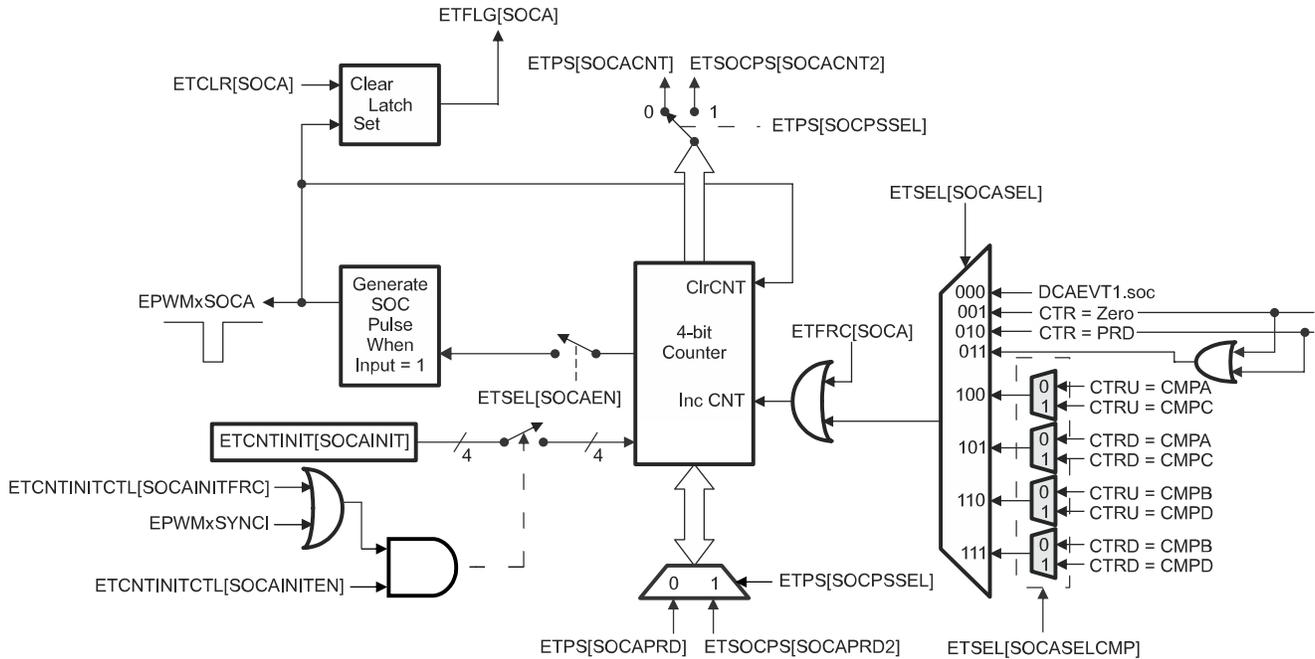


図 1-47. イベント・トリガ割り込みジェネレータ

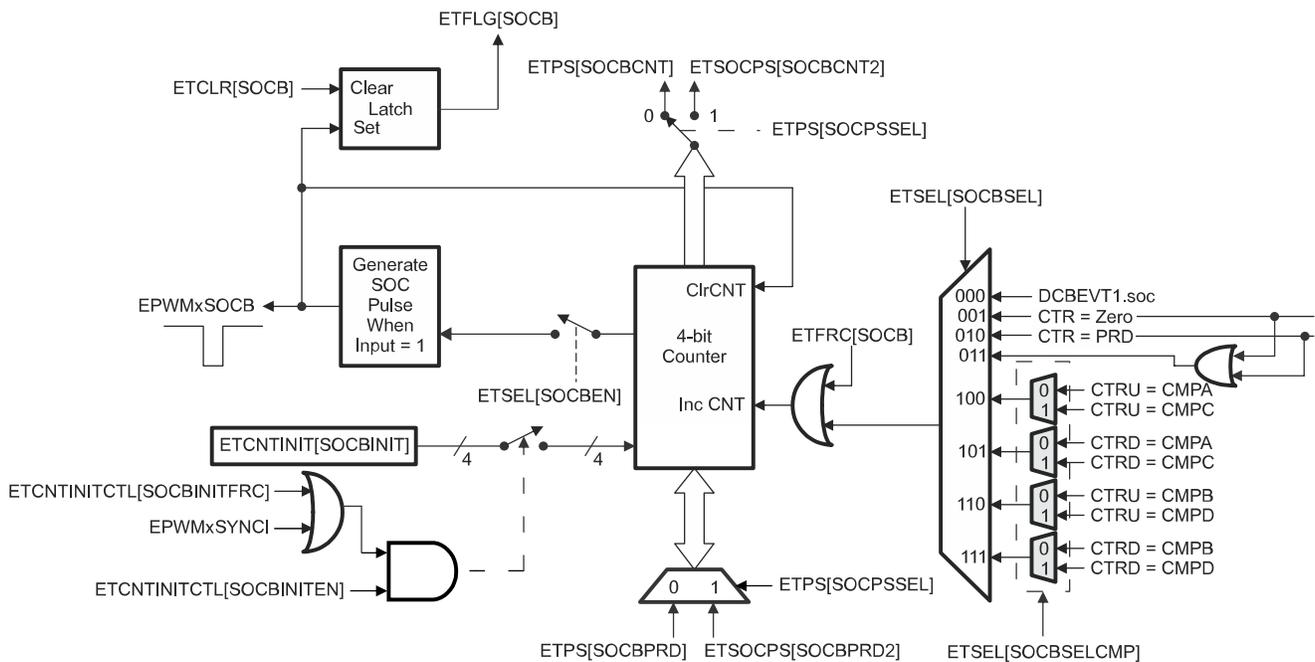
図 1-48 に、イベント・トリガの変換開始 A (SOCA) パルス・ジェネレータの動作を示します。ETSEL レジスタで定義された SOCASELCMP および SOCBSELCMP ビット・フィールドなどの拡張機能のおかげで、CMPC および CMPD イベントがそれぞれ変換開始をトリガできます。ETPS[SOCPSSEL] ビット・フィールドは、SOCACNT2 と SOCAPRD2 が制御を受け持つかどうかを決定します。ETPS[SOCACNT] カウンタおよび ETPS[SOCAPRD] 周期の値は、パルスが連続的に生成されることを除いて、割り込みジェネレータと同様に振る舞います。つまり、パルス生成時にパルス・フラグ ETFLG[SOCA] がラッチされますが、割り込みジェネレータはさらなるパルス生成を停止しません。有効化および無効化ビット ETSEL[SOCAEN] はパルス生成を停止しますが、割り込み生成ロジックと同様、周期値に達するまで入力イベントがカウントされます。SOCA および SOCB パルスをトリガするイベントは、ETSEL[SOCASEL] および ETSEL[SOCBSEL] ビットで個別に設定できます。起こり得るイベントは、デジタル比較 (DC) サブモジュールからの DCAEVT1.soc および DCBEVT1.soc イベント信号の追加によって、割り込み生成ロジックのために指定できるイベントと同じです。SOCACNT2 の初期化方式は、イネーブル機能、値の初期化機能、同期 / ソフトウェア強制選択機能をそれぞれ備えた割り込みジェネレータと非常によく似ています。



注：DCAEVT1.soc 信号は、デジタル比較 (DC) サブモジュール (セクション 1.11) によって生成されます。

図 1-48. イベント・トリガ SOCA パルス・ジェネレータ

図 1-49 に、イベント・トリガの変換開始 B (SOCB) パルス・ジェネレータの動作を示します。イベント・トリガ SOCB パルス・ジェネレータは、SOCA と同じ方法で動作します。



注：DCBEVT1.soc 信号は、デジタル比較 (DC) サブモジュール (セクション 1.11) によって生成されます。

図 1-49. イベント・トリガ SOCB パルス・ジェネレータ

1.11 デジタル比較 (DC) サブモジュール

図 1-50 に、デジタル比較 (DC) サブモジュール信号の、ePWM システム内のその他のサブモジュールとのインターフェイスを示します。

eCAP 入力信号は入カクロスバー信号から供給されます (図 1-51 を参照)。

このデバイスでは、どの GPIO ピンも、トリップ・ゾーン・サブモジュールとデジタル比較サブモジュールへのトリップ・ゾーン入力とトリップ入力として柔軟に割り当てることができます。入カクロスバー入力選択 (INPUTxSELECT) レジスタは、どの GPIO ピンを特定のトリップ・ゾーン入力/トリップ入力に割り当てるかを定義します。

デジタル比較 (DC) サブモジュールは、ePWM モジュールの外部の信号 (たとえば、アナログ・コンパレータからの CMPSSx 信号) を比較し、PWM イベント / アクションを直接生成してから、イベント・トリガ、トリップ・ゾーン、時間ベース・サブモジュールに供給します。また、DC イベント信号からのノイズまたは不要なパルスをフィルタ処理するため、ブランキング・ウィンドウ機能をサポートしています。

注

トリップ信号の誤ラッチを防止するため、ユーザーは、クロックを有効化する前に、選択されたピンを正しい状態に駆動し、各 ePWM パリフェラルのトリップ入力を設定する必要があります。

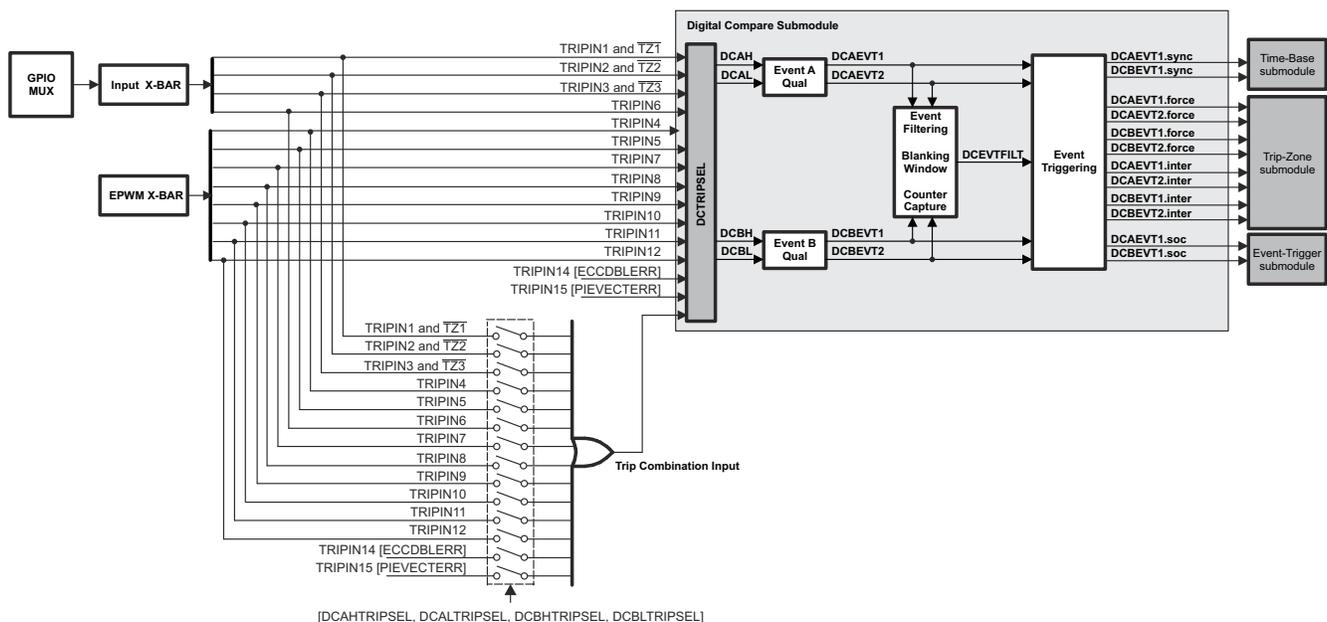


図 1-50. デジタル比較サブモジュールのハイレベル・ブロック図

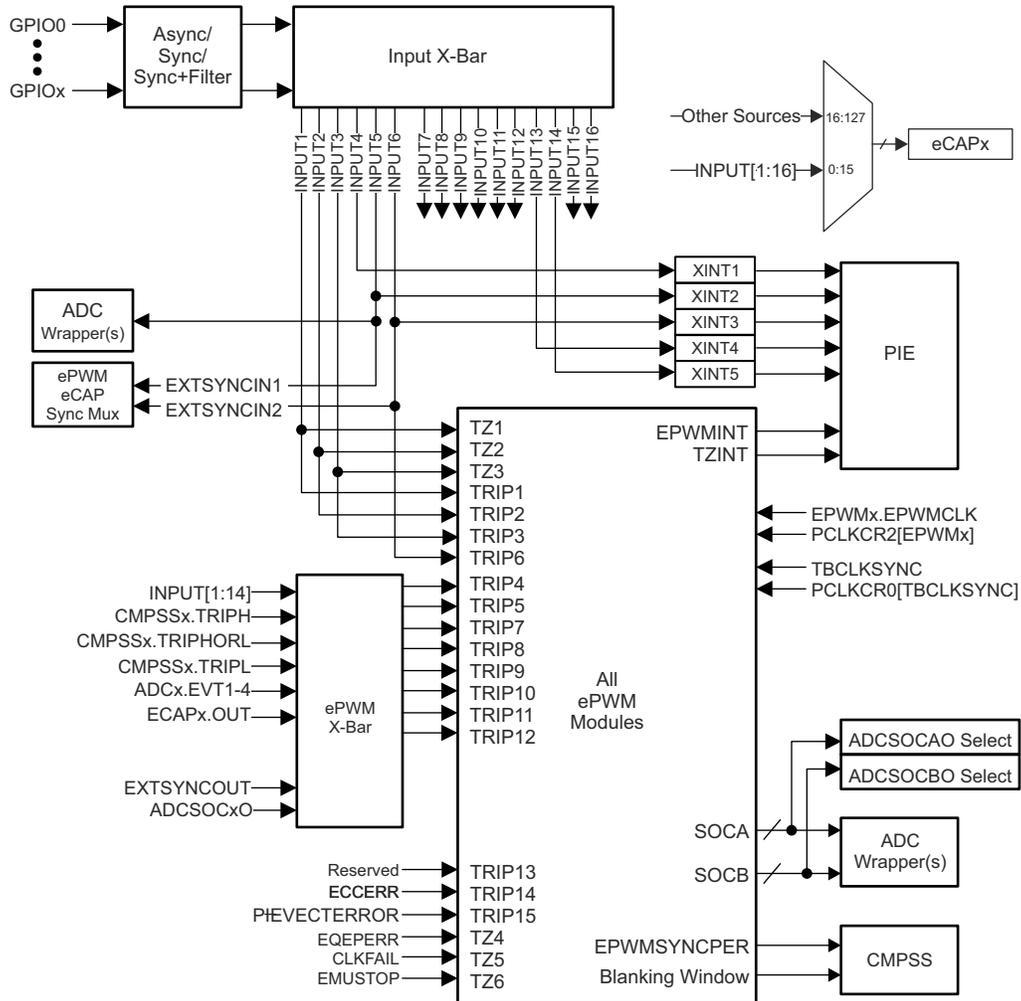


図 1-51. GPIO マルチプレクサとトリップ入力の接続

1.11.1 デジタル比較サブモジュールの目的

デジタル比較サブモジュールの主な機能は次のとおりです。

- GPIO ペリフェラル、内部 PIE、ECC エラー信号、TZ1、TZ2、TZ3 入力を使い、入力クロスバーと EPWM クロスバーを通して外部から信号を供給されたアナログ・コンパレータ (COMP) モジュール出力が、デジタル比較 A High/Low (DCAH、DCAL) およびデジタル比較 B High/Low (DCBH、DCBL) 信号を生成します。
- DCAH/L および DCBH/L 信号は、以下の目的で、フィルタ処理できるイベントと、トリップ・ゾーン、イベント・トリガ、時間ベースの各サブモジュールに直接適用できるイベントのどちらかをトリガします。
 - トリップ・ゾーン割り込みを生成します。
 - ADC 変換開始を生成します。
 - 強制的にイベントを発生させます。
 - ePWM モジュールの TBCTR を同期させるための同期イベントを生成します。
- ノイズを除去するため、イベント・フィルタリング (ブランキング・ウィンドウ・ロジック) が入力信号を選択的に無視することもできます。

1.11.2 CMPSS を使用した拡張トリップ・アクション

一度に複数の CMPSS が DCA/BEVTx イベントとトリップ・アクションに影響を及ぼすことができるように、ePWM モジュールの外部のソースからのすべてのトリップ入力 (最大 15) を 1 つにまとめ、DCTRIPSEL レジスタを使って「組み合わせ入力」として DCAH、DCAL、DCBH、DCBL に供給する OR ロジックが備わっています。これは、DCTRIPSEL レジスタの「トリップ組み合わせ入力」(値 0xF) を選択することで使えるようになります。

DCAH、DCAL、DCBH、DCBL 信号を生成するため、組み合わせロジックに接続するトリップ入力を個別に選択できます。これは、DCAHTRIPSEL、DCALTRIPSEL、DCBHTRIPSEL、DCBLTRIPSEL レジスタの選択によって実現されます。組み合わせ入力用に選択された入力は DCTRIPSEL レジスタに渡されます。

1.11.3 CMPSS を使用したサイクルごとの ePWM のトリップ

ePWM をサイクルごとにトリップさせるために CMPSS を使用している場合、ある PWM サイクルのアサートされたコンパレータ・トリップ状態が次のサイクルまで延びないようにするための手段を講じることができます。CMPSS を使うと、トリップ条件を下流の ePWM モジュールに通知できます。ピーク電流モード制御などのアプリケーションに必要なトリップ・イベントは、PWM サイクルごとに 1 つのみです。特定の条件では、予防措置を講じない場合、長期間続く、または遅れて発生する (PWM サイクルの終わり近くになって到着する) トリップ・イベントが、次の PWM サイクルに持ち越される可能性があります。CMPSS デジタル・フィルタと ePWM デジタル比較 (DC) サブモジュールのどちらかがコンパレータのトリップ信号をクオリファイするように設定されている場合、トリップ信号の論理の変化に ePWM トリップ・ロジックが応答する前に、「N」個のクオリフィケーション・クロック・サイクルが取り込まれます。ePWM トリップ条件がクオリファイされると、そのトリップ条件は、コンパレータ・トリップ信号がデアサートされた後、N クロック・サイクルにわたってアクティブに維持されます。クオリファイされたコンパレータ・トリップ信号が、PWM サイクルの終了前の N クロック・サイクル以内にアサートされたまま維持されている場合、次の PWM サイクルが開始されるまでそのトリップ条件はクリアされません。そのため、新しい PWM サイクルは、サイクルが始まるとすぐにトリップ条件を検出してしまいます。

この不要なトリップ条件を回避するため、アプリケーションは、ePWM トリップ・ロジックによって検出されたクオリファイ済みトリップ信号が各 PWM サイクルの終了前に確実にデアサートされるようにするための手段を講じることができます。これは、以下のようなさまざまな方法で実現できます。

- PWM サイクルの終了前の N クロック・サイクル以内にコンパレータ・トリップがアサートされないように、システムを設計します。
- PWMSYNCPER 信号の少なくとも 2 クロック・サイクル前に、ePWM イベント・フィルタを使ってコンパレータ・トリップ信号のブランキングを有効化し、次の PWM サイクルまで少なくとも N クロック・サイクルの間ブランキングを継続します。
- CMPSS の COMPxLATCH パスを使用する場合、PWM サイクルの終了の少なくとも N クロック・サイクル前に COMPxLATCH をクリアします。このラッチは、(COMPSTSLR を使って) ソフトウェアによって、または PWMSYNCPER 信号を早く生成することでクリアできます。このデバイスの ePWM モジュールは、

PWM サイクル内で PWMSYNCPER を任意に配置するため、(HRPCTL を使って) CMPC または CMPD 一致時に PWMSYNCPER を生成する機能を備えています。

1.11.4 デジタル比較サブモジュールの動作の概要

以下のセクションでは、デジタル比較サブモジュールの動作の概要と設定オプションについて説明します。

1.11.4.1 デジタル比較イベント

[セクション 1.11.1](#) で説明したように、トリップ・ゾーン入力 ($\overline{TZ1}$ 、 $\overline{TZ2}$ 、 $\overline{TZ3}$) とアナログ・コンパレータ (COMP) モジュールからの CMPSSx 信号を、DCTRIPSEL ビットを使って選択し、デジタル比較 A High および Low (DCAH/L) 信号とデジタル比較 B High および Low (DCBH/L) 信号を生成できます。次に、TZDCSEL レジスタの設定が、選択された DCAH/L および DCBH/L 信号でのアクションをクオリファイします。これらのアクションは、DCAEVT1/2 および DCBEVT1/2 イベントを生成します (イベント・クオリフィケーション A および B)。

注

\overline{TZn} 信号は、DCEVT トリップ機能として使う場合、通常の入力信号として扱われ、アクティブ High またはアクティブ Low の入力として定義できます。 \overline{TZn} 、DCAEVTx.force、DCBEVTx.force のいずれかの信号がアクティブになると、ePWM 出力は非同期的にトリップされます。ラッチ状態を維持するには、少なくとも $3 \cdot TBCLK$ の同期パルス幅が必要です。同期パルス幅が $3 \cdot TBCLK$ 未満である場合、CBC または OST ラッチがトリップ条件をラッチできない場合があります。

その後、DCAEVT1/2 および DCBEVT1/2 イベントをフィルタ処理してフィルタ処理済みイベント信号 (DCEVTFILT) を生成します。または、フィルタ処理をバイパスできます。フィルタ処理については、[セクション 1.11.4.2](#) で詳しく説明します。DCAEVT1/2 および DCBEVT1/2 イベント信号とフィルタ処理済み DCEVTFILT イベント信号のどちらかが、トリップ・ゾーン・モジュールに対する強制信号、TZ 割り込み信号、ADC 変換開始 (SOC) 信号、PWM 同期信号を生成できます。

- 強制信号** : DCAEVT1/2.force 信号は、(TZCTL、TZCTLDCA レジスタ設定を使って) EPWMxA ピンの出力に直接影響を及ぼすトリップ・ゾーン条件を強制的にトリガします。あるいは DCAEVT1/2.force 信号は、(TZCTL、TZCTLDCA レジスタ設定を使って) EPWMxB ピンの出力に直接影響を及ぼすトリップ・ゾーン条件を強制的にトリガします。(TZSEL レジスタを使って) DCAEVT1.force 信号と DCBEVT1.force 信号が EPWMxA と EPWMxB を同時にワンショット・トリップ・ソースとして選択できる場合、DCAEVT1.force 信号と DCBEVT1.force 信号は、TZCTL または TZCTL2 レジスタ設定を使ってトリップ・アクションに影響を及ぼすことができます。あるいは、DCAEVT2.force 信号と DCBEVT2.force 信号が EPWMxA と EPWMxB を同時にサイクル・バイ・サイクル・トリップ・ソースとして選択できる場合、DCAEVT2.force 信号と DCBEVT2.force 信号は、TZCTL または TZCTL2 レジスタ設定を使ってトリップ・アクションに影響を及ぼすことができます。DCBEVT1/2.force 信号は同様に振る舞いますが、EPWMxA 出力ピンではなく、EPWMxB 出力ピンに影響を及ぼします。

TZCTL、TZCTL2、TZCTLDCA、TZCTLDCA レジスタで競合するアクションの優先順位は次のとおりです (最も高い優先度がより低い優先度に優先します)。

出力 EPWMxA :

- TZA (最高) -> DCAEVT1 -> DCAEVT2 (最低)
- TZAU (最高) -> DCAEVT1U -> DCAEVT2U (最低)
- TZAD (最高) -> DCAEVT1D -> DCAEVT2D (最低)

出力 EPWMxB :

- TZB (最高) -> DCBEVT1 -> DCBEVT2 (最低)
- TZBU (最高) -> DCBEVT1U -> DCBEVT2U (最低)
- TZBD (最高) -> DCBEVT1D -> DCBEVT2D (最低)

- 割り込み信号** : DCAEVT1/2.interrupt 信号は、PIE へのトリップ・ゾーン割り込みを生成します。この割り込みを有効化するには、TZEINT レジスタの DCAEVT1、DCAEVT2、DCBEVT1、DCBEVT2 ビットをセットします。これらのイベントのいずれかが発生すると、EPWMxTZINT 割り込みがトリガされます。その割り込みをクリアするには、TZCLR レジスタの対応するビットをセットする必要があります。
- 変換開始 (SOC) 信号** : DCAEVT1.soc 信号は、イベント・トリガ・サブモジュールと接続されており、ETSEL[SOCASEL] ビットを使用して、ADC 変換開始 A (SOCA) パルスを生成するイベントとして選択できます。同様に、DCBEVT1.soc 信号は、ETSEL[SOCBSEL] ビットを使用して、ADC 変換開始 B (SOCA) パルスを生成するイベントとして選択できます。

- **同期信号**：時間ベース・カウンタへの同期パルスを生成するため、DCAEVT1.sync および DCBEVT1.sync イベントは EPWMxSYNCl 入力信号および TBCTL[SWFSYNC] 信号と論理和がとられます。

図 1-52 と 図 1-53 に、DCxEVT1、DCxEVT2、DCEVTFLT 信号を処理して、デジタル比較 A および B イベント強制、割り込み、変換開始、同期信号を生成する方法を示します。

位相シフト・フルブリッジ (PSFB) コンバータのような一部のアプリケーションでは、CBC トリップ・イベントと OST トリップ・イベントの際に各種アクションを実行する必要があります。これは、DCxEVT1LAT を使用して実現できます。

- このラッチは、DCxCTL.EVTy.LATCLRSEL の設定に基づいて、「CNT = 0」イベント、「CTR = PRD」イベント、「CNT = 0 または CTR = PRD」イベントでクリアできます。これは CBC ラッチ・クリア機能に似ています。
- DCxEVTy.force 信号は、DCxCTL.EVTyLATSEL の値に基づいて、ラッチ付きバージョンとラッチなしバージョンのどちらかを選択できます。
- DCxEVTyLAT 信号のステータスは、DCxCTL.EVTyLAT フィールドを読み出すことでアクセスできます。

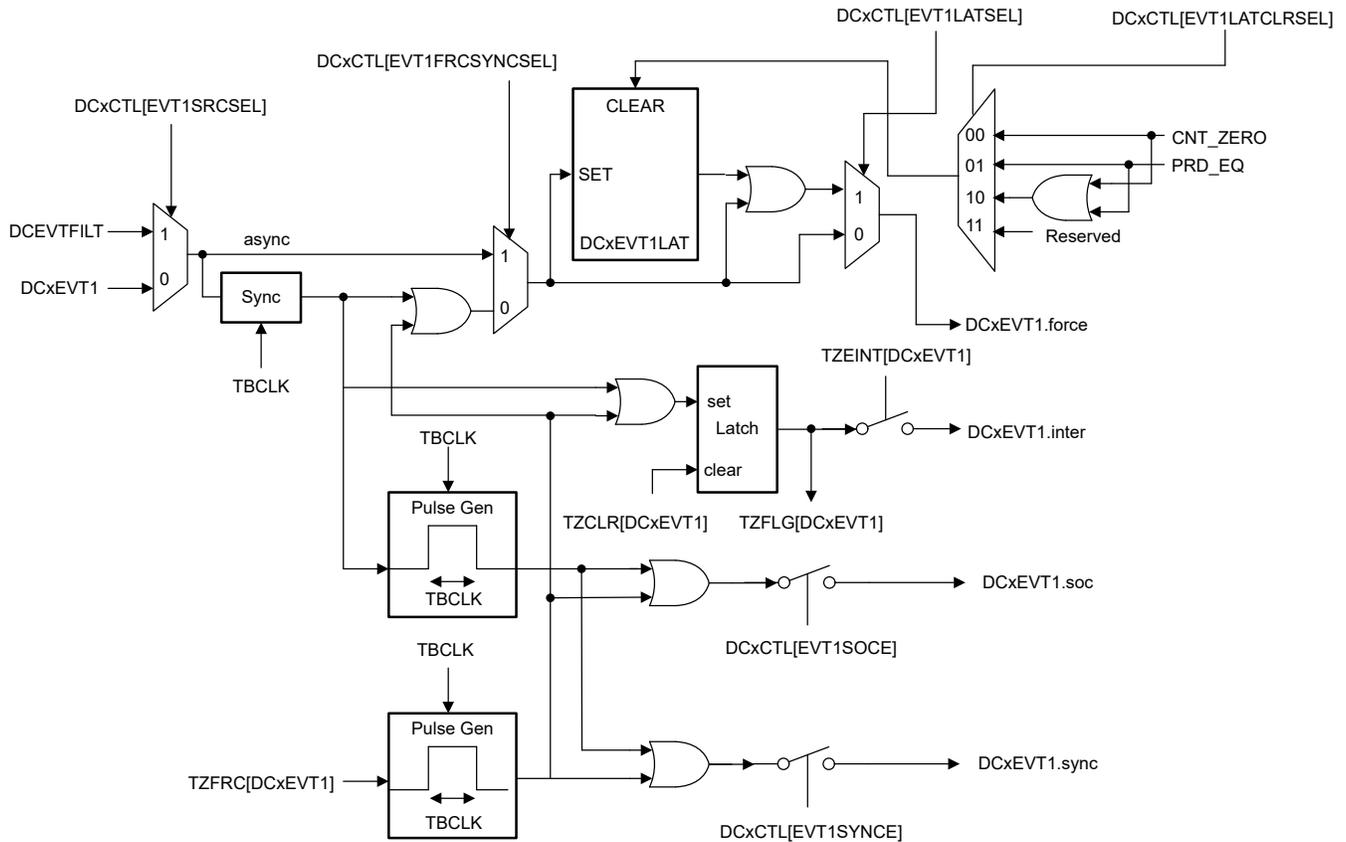


図 1-52. DCxEVT1 イベント・トリガ

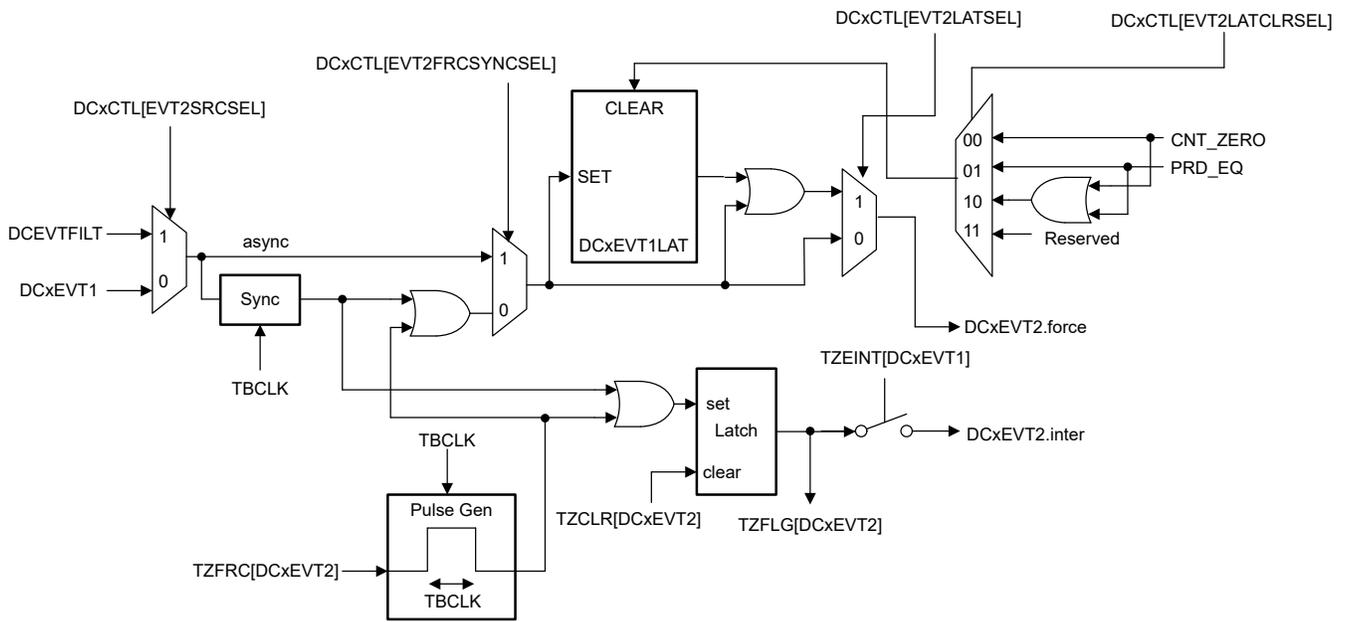


図 1-53. DCxEVT2 イベント・トリガ

1.11.4.2 イベント・フィルタリング

イベント・フィルタリング・ロジックを使って、特定の期間にわたってイベントを無視することで、DCAEVT1/2 および DCBEVT1/2 イベントをフィルタ処理してノイズを除去することもできます。DCAEVT1/2 および DCBEVT1/2 イベントをトリガするためにアナログ・コンパレータ出力が選択される場合と、PWM 出力をトリップさせる前に、または割り込みまたは ADC 変換開始を生成する前に、信号に含まれる可能性があるノイズを除去するためにブランキング・ロジックが使われる場合に、この機能は便利です。イベント・フィルタリングは、トリップ・イベントの TBCTR 値をキャプチャすることもできます。図 1-54 に、イベント・フィルタリング・ロジックの詳細を示します。

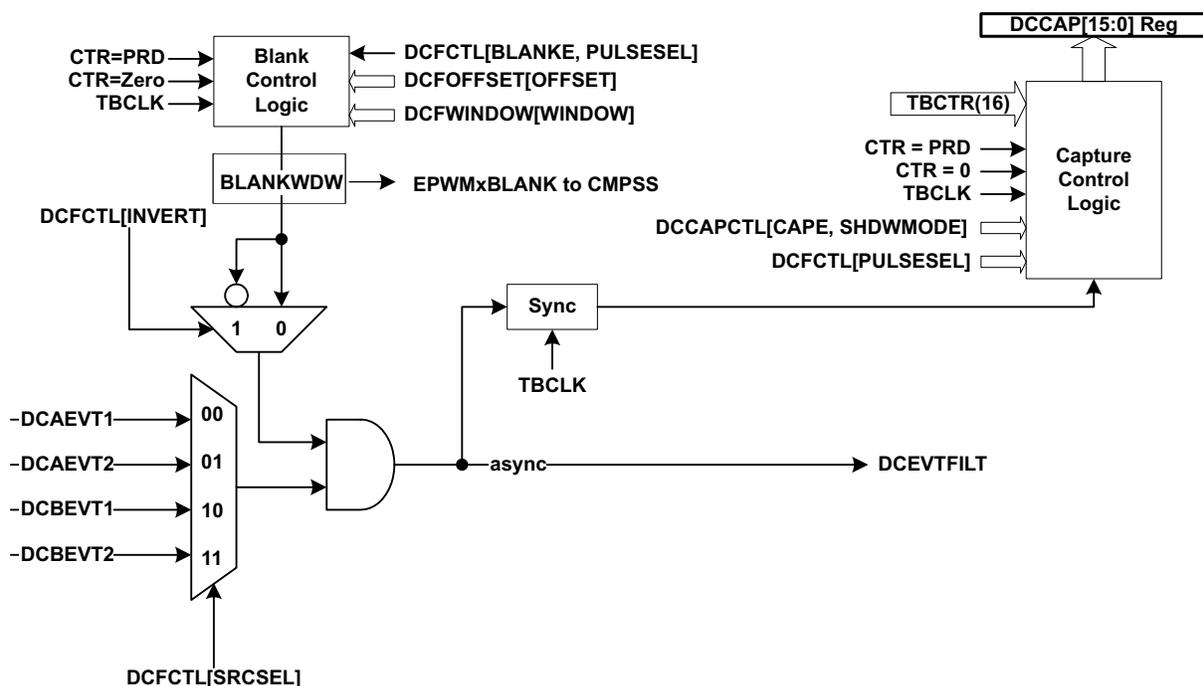


図 1-54. イベント・フィルタリング

ブランキング・ロジックが有効化されている場合、デジタル比較イベント (DCAEVT1、DCAEVT2、DCBEVT1、DCBEVT2) の 1 つがフィルタリングのために選択されます。ブランキング・ウィンドウ (このウィンドウがアクティブである間に信号上に発生するすべてのイベントをフィルタで除去します) は、CTR = PRD パルスと CTR = 0 パルスのどちらか、または CTR = PRD と CTR = 0 の両方 (DCFCTL[PULSESEL] ビットで設定) に整列させることができます。TBCLK カウントのオフセット値は DCFOFFSET レジスタに書き込まれます。このレジスタは、CTR = PRD または CTR = 0 パルスの後のブランキング・ウィンドウの開始タイミングを決定します。ブランキング・ウィンドウの期間 (オフセット・カウンタが満了した後の TBCLK カウント数で表されます) はアプリケーションによって DCFWINDOW レジスタに書き込まれます。ブランキング・ウィンドウ中、すべてのイベントは無視されます。ブランキング・ウィンドウが終了する前と終了した後に、イベントは変換開始、同期、割り込み、強制信号を従来と同様に生成できます。

注

ブランキング・ウィンドウが満了した後、トリップ入力が少なくとも 3 ePWM サイクルの間有効に維持されるように、ePWM ブランキング・ウィンドウを適切に設定する必要があります。

図 1-55 に、ePWM 周期内のオフセットとブランキング・ウィンドウの各種タイミング条件を示します。ブランキング・ウィンドウが CTR = 0 または CTR = PRD 境界をまたぐ場合、CTR = 0 または CTR = PRD パルスの後、次のウィンドウは依然として同じオフセット値で開始されることに注意します。

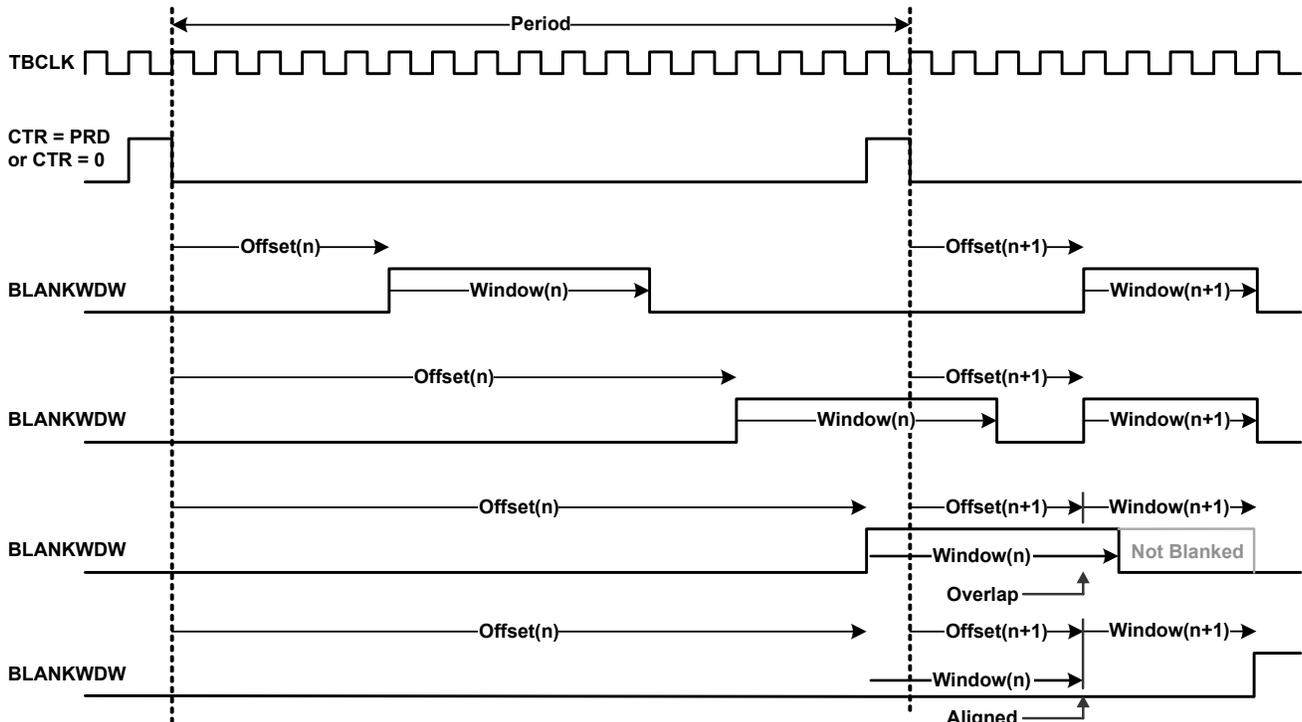


図 1-55. ブランキング・ウィンドウのタイミング図

BLANKPULSEMIX 信号

DCFCTL マルチプレクサ (ブランク制御ロジックとキャプチャ制御ロジックで利用可能) には、BLANKPULSEMIX 信号を選択するという新しい選択肢が追加されました。DCFCTL[PULSESEL] で選択することで、BLANKPULSEMIX 信号が使われます。

1.11.4.3 バレー・スイッチング

バレー・スイッチング機能は、イベント・フィルタリングとイベント・フィルタリング・ロジック (セクション 1.11.4.2 を参照) の組み合わせで表されます。この機能を使うと、外部回路を何も追加しなくても、プログラム可能なバレー・スイッチングを実現できます。このモジュールは、以下の機能を実行するオンチップ・ハードウェア機能を備えています。

- 発振周期をキャプチャします。
- PWM のスイッチングの瞬間を精密に遅延させます。
- 遅延が作用するまでのエッジの数を設定できます。
- トリガとイベントに関する複数の選択肢を提供します。
- システム / 動作条件の変化に応じて最適な性能が得られるように、簡単に適応させることができます。

バレー・スイッチングをサポートするため、DCxEVTy 信号をさらに処理する必要があります。バレー・スイッチング機能を有効化する方法について簡単に説明します。

1. ブランキング・ウィンドウを追加するためのオプション (ブランク制御ロジック) を使って、バレー・スイッチング・ブロックへの入力として DCxEVTy イベントの 1 つ (DCFCTL[SRSEL]) を選択します。ここで、バレー・スイッチング・ブロックへの入力として、上記のコンパレータ出力 (または外部入力) を選択します。
2. エッジ選択ロジックによって、「n」個の立ち上がりエッジ、立ち下がりエッジ、両方のエッジをキャプチャするようにエッジ・フィルタ (DCFCTL[EDGEMODE、EDGECOUNT]) を設定します。
3. エッジ・フィルタをリセットおよびリスタートするための適切なイベント (VCAPCTL[TRIGSEL]) を選択します。エッジ・キャプチャ・イベントは、この選択されたエッジによってトリガされ、または作動可能になります。
4. バレー・キャプチャ・ロジックを有効化 (VCAPCTL[VCAPE]) します。
5. 発振周期測定のカウント開始を示すスタート・エッジ (VCNTCFG[STARTEDGE]) を選択します。ここで、16 ビット・カウンタがカウントを開始します。
6. 16 ビット・カウンタがカウントを停止するエッジを示すストップ・エッジ (VCNTCFG[STOPEDGE]) を選択します。キャプチャされたカウンタ値 (CNTVAL) は、発振周期情報を提供します。
 - STOPEDGE の値は STARTEDGE の値より常に大きい必要があります。
7. キャプチャされた遅延 (CNTVAL) がエッジ・フィルタ処理された DCxEVTy 信号に適用されるように設定します。CNTVAL 値は、そのまま適用することも、ソフトウェアでプログラムされた値と組み合わせて適用することも (オフセット調整に便利) (SWVDELVAL) できます。または、SWVDELVAL を使っても使わなくても、遅延の一部のみを適用することもできます。これは、バレー・ポイントに対応する遅延を正しく適用するのに役立ちます。(VCAPCTL[VDELAYDIV])
8. 上記のキャプチャ値に基づいてハードウェア遅延が適用されるように VCAPCTL[EDGEFILTDLYSEL] を設定します。

カウンタが停止すると、カウンタ値が CNTVAL レジスタにコピーされ、カウンタはゼロにリセットされます。VCAPCTL[TRIGSEL] で選択されたイベントの発生によってロジックが再度トリガされるまで、それ以上のキャプチャは行われません。この実装では、ソフトウェア・トリガは VCAPCTL[TRIGSEL] のソースとして使用されます。トリガ・イベントが発生すると、カウンタの現在のステータスに関係なく、カウンタはリセットされ、STARTEDGE の発生と同時にゼロからカウントを開始します。同様に、トリガ・イベントが発生すると、エッジ・フィルタはリセットされ、STARTEDGE の発生と同時にゼロからカウントを開始します。

その後、バレー・スイッチング・ブロック (DCEVTFILT) の出力は、PWM 時間ベースを同期させるために使われます。その過程を図 1-56 に示します。

注

バレー・スイッチング・ハードウェアおよびソフトウェアの使い方を示す具体的なアプリケーション例は、C2000Ware で提供しています。

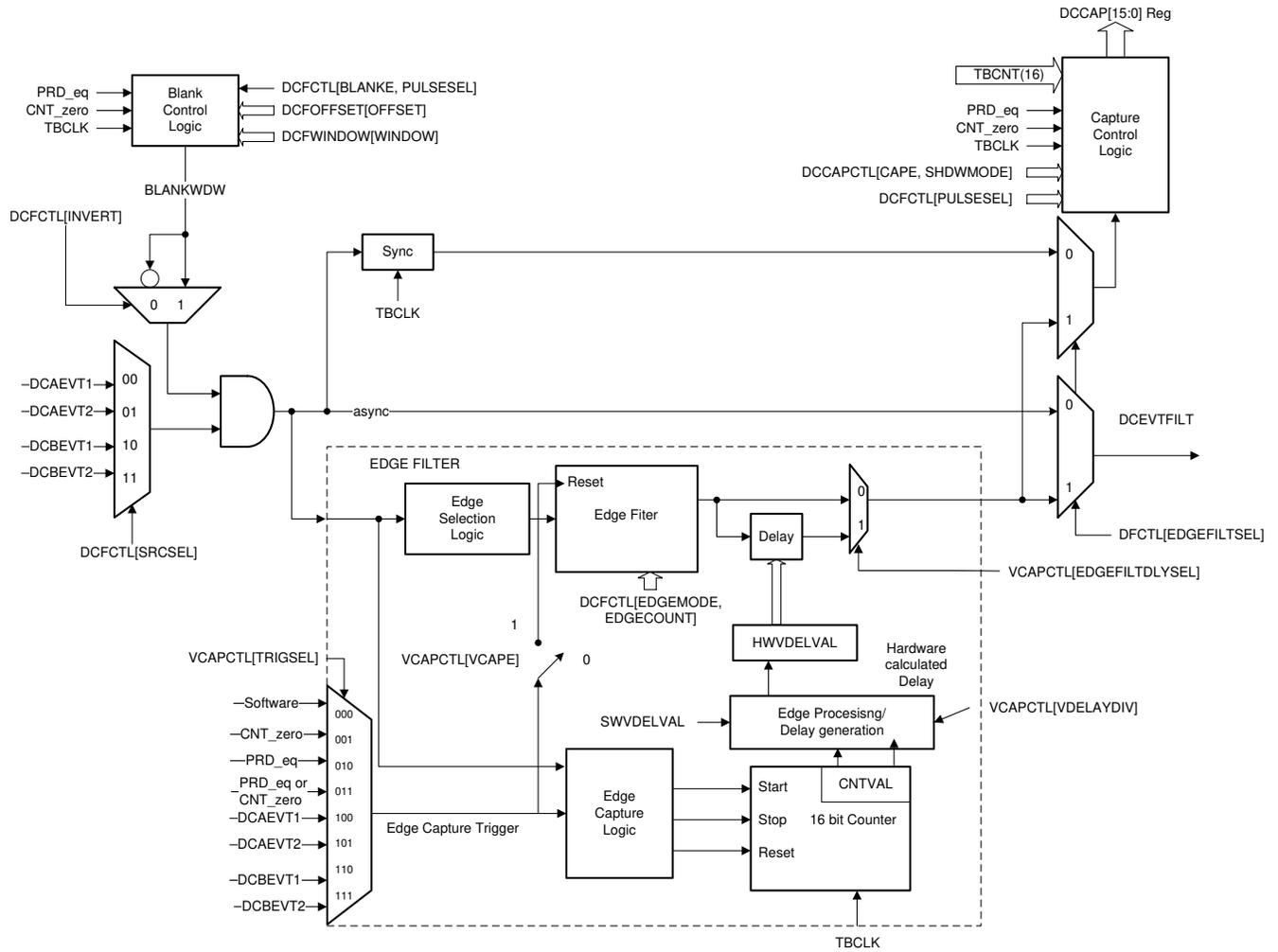


図 1-56. バレー・スイッチング

1.12 ePWM クロスバー

図 1-57 に、ePWM クロスバーのアーキテクチャを示します。このモジュールを使用すると、8 つの専用 ePWM トリップ入力 (TRIP4、TRIP5、TRIP7、TRIP8、TRIP9、TRIP10、TRIP11、TRIP12) のすべてへの各種トリガ・ソースを選択できます。

注

クロスバー・フラグを含むクロスバー・モジュールの詳細については、「クロスバー」の章を参照してください。

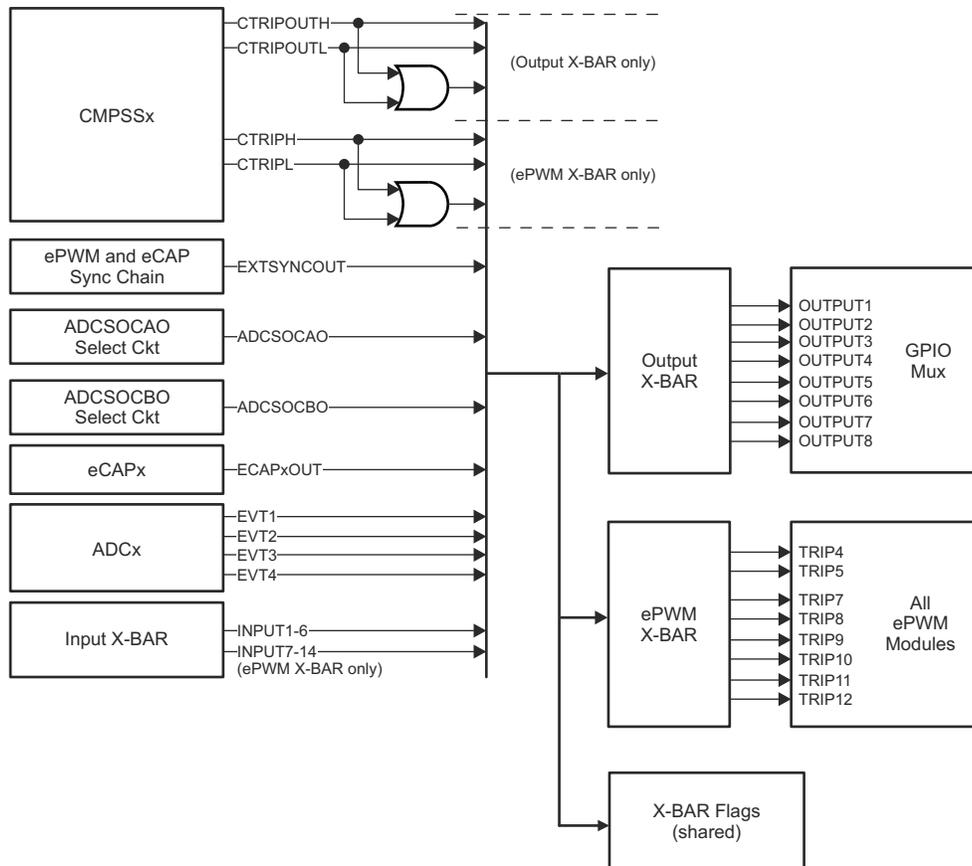


図 1-57. ePWM クロスバー

1.13 電源トポロジへの応用

ePWM モジュールは、スタンドアロン・モジュールとして完全に動作するために、またはその他の同一の ePWM モジュールと同期して動作するために必要なすべてのローカル・リソースを備えています。

1.13.1 複数のモジュールの動作の概要

この章ではこれまで、すべての説明は 1 つのモジュールの動作についての説明でした。システム内で互いに連携して動作する複数のモジュールを理解しやすくするため、この図に記載された ePWM モジュールは、より簡略化されたブロック図 (図 1-58) で表されています。この ePWM の概略ブロックには、連携して動作する複数の ePWM モジュールを使ってマルチスイッチ・パワー・トポロジを制御するのに必要な主要リソースのみが示されています。

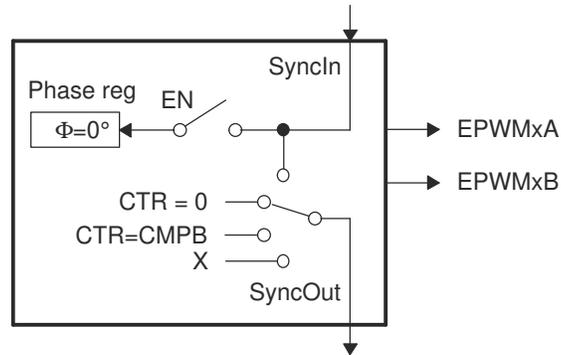


図 1-58. 簡素化された ePWM モジュール

1.13.2 主な設定機能

各モジュールに対して使用できる主な設定オプションは次のとおりです。

- SyncIn のオプション
 - 受信同期ストロブ時に位相レジスタで自身のカウンタをロードします。— イネーブル (EN) スイッチは閉じた状態
 - 何もしない、または受信同期ストロブを無視します。— イネーブル・スイッチは開いた状態
 - 同期フリースルー - SyncIn と SyncOut を接続
 - マスタ・モード。PWM 境界で同期させます。— CTR = PRD と SyncOut を接続
 - マスタ・モード。プログラマブルな任意の時点で同期させます。— CTR = CMPB と SyncOut を接続
 - モジュールをスタンダアロン・モードに入っており、その他のモジュールと同期させません。— X (無効) と SyncOut を接続
- SyncOut のオプション
 - 同期フリースルー - SyncIn と SyncOut を接続
 - マスタ・モード。PWM 境界で同期させます。— CTR = PRD と SyncOut を接続
 - マスタ・モード。プログラマブルな任意の時点で同期させます。— CTR = CMPB と SyncOut を接続
 - モジュールをスタンダアロン・モードに入っており、その他のモジュールと同期させません。— X (無効) と SyncOut を接続

SyncOut の各選択肢に対して、モジュールは、SyncIn ストロブ入力時に新しい位相値で自身のカウンタをロードするか、値を無視するか (つまり、イネーブル・スイッチを使って) を選択することもできます。各種の組み合わせが可能です。最も一般的な 2 つ (マスタモジュール・モードとスレーブ・モジュール・モード) を [図 1-59](#) に示します。

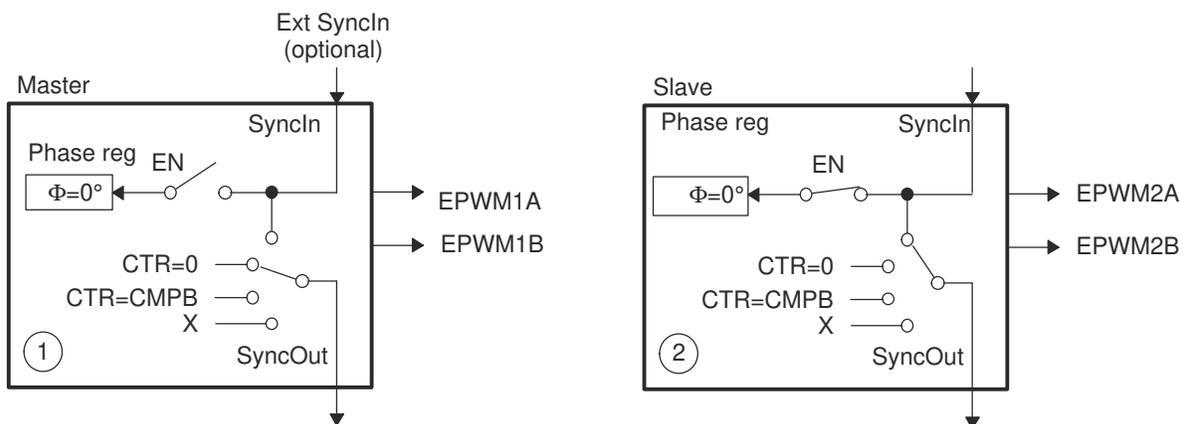
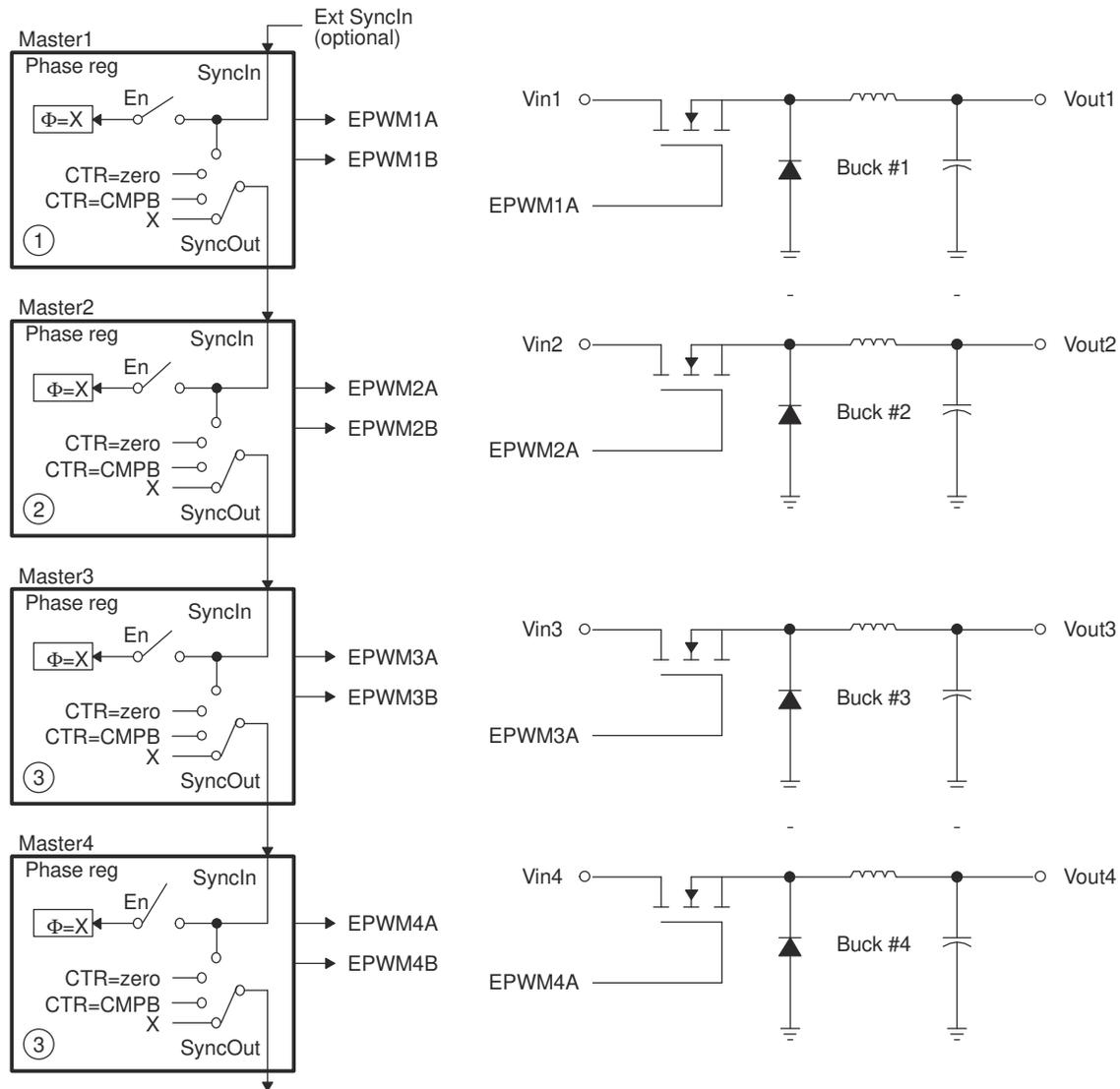


図 1-59. 代表的なマスタとして構成された EPWM1、スレーブとして構成された EPWM2

1.13.3 独立した周波数による複数の降圧コンバータの制御

降圧は、最もシンプルなパワー・コンバータ・トポロジの1つです。マスタとして構成された1つのePWMモジュールが、同じPWM周波数で2つの降圧段を制御できます。各降圧コンバータに対して独立した周波数制御が必要な場合、各コンバータ段に1つのePWMモジュールを割り当てる必要があります。図1-60に、各段が独立した周波数で動作している4つの降圧段を示します。この場合、4つのePWMモジュールはすべてマスタとして構成されており、同期は使われていません。図1-61に、図1-60に示す設定によって生成される波形を示します。4つの段が存在しますが、3つの波形のみが示されていることに注意します。



A. $\phi = X$ は位相レジスタの値が「ドントケア」であることを示します。

図 1-60. 4 つの降圧段の制御ここで、 $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$

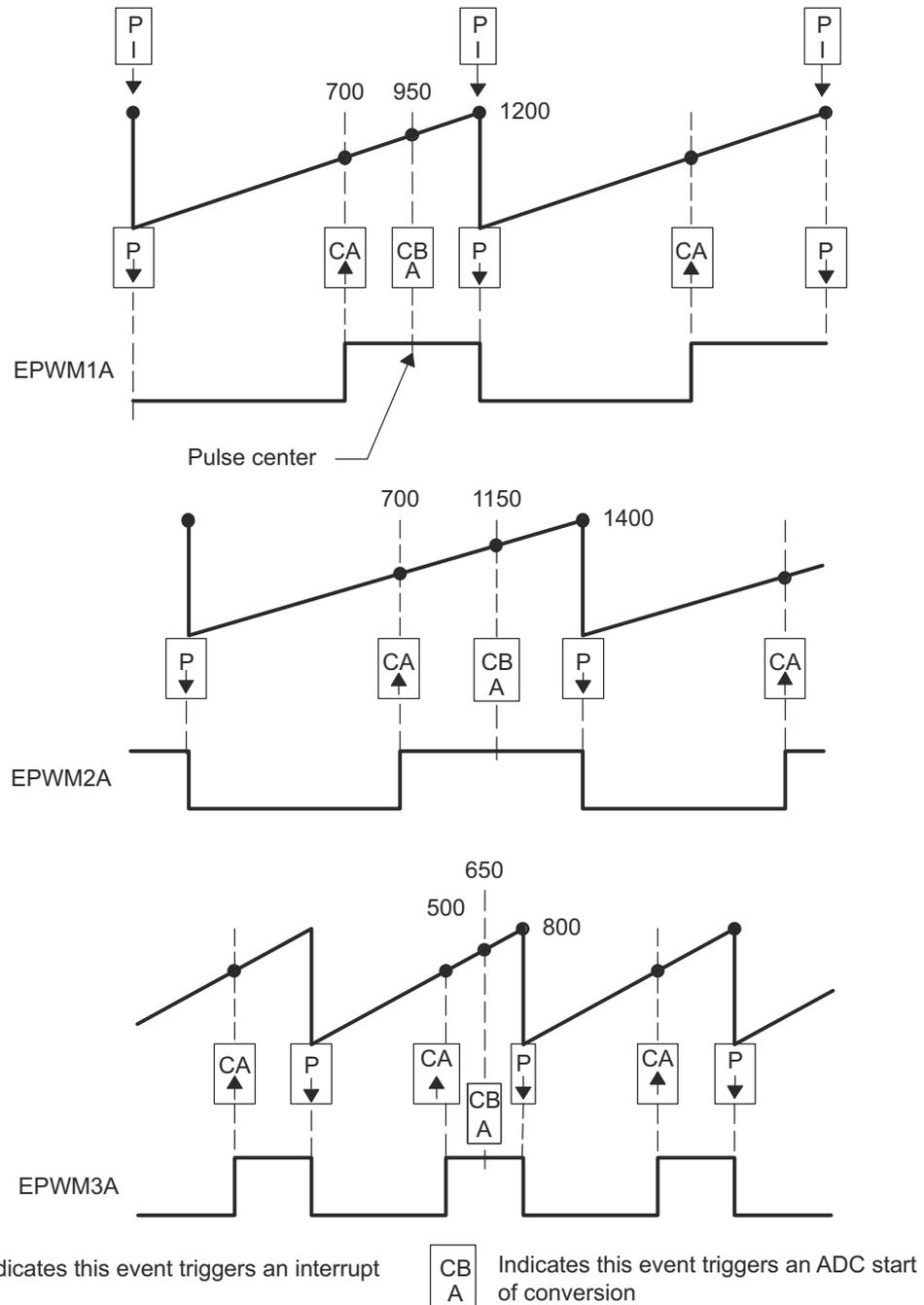
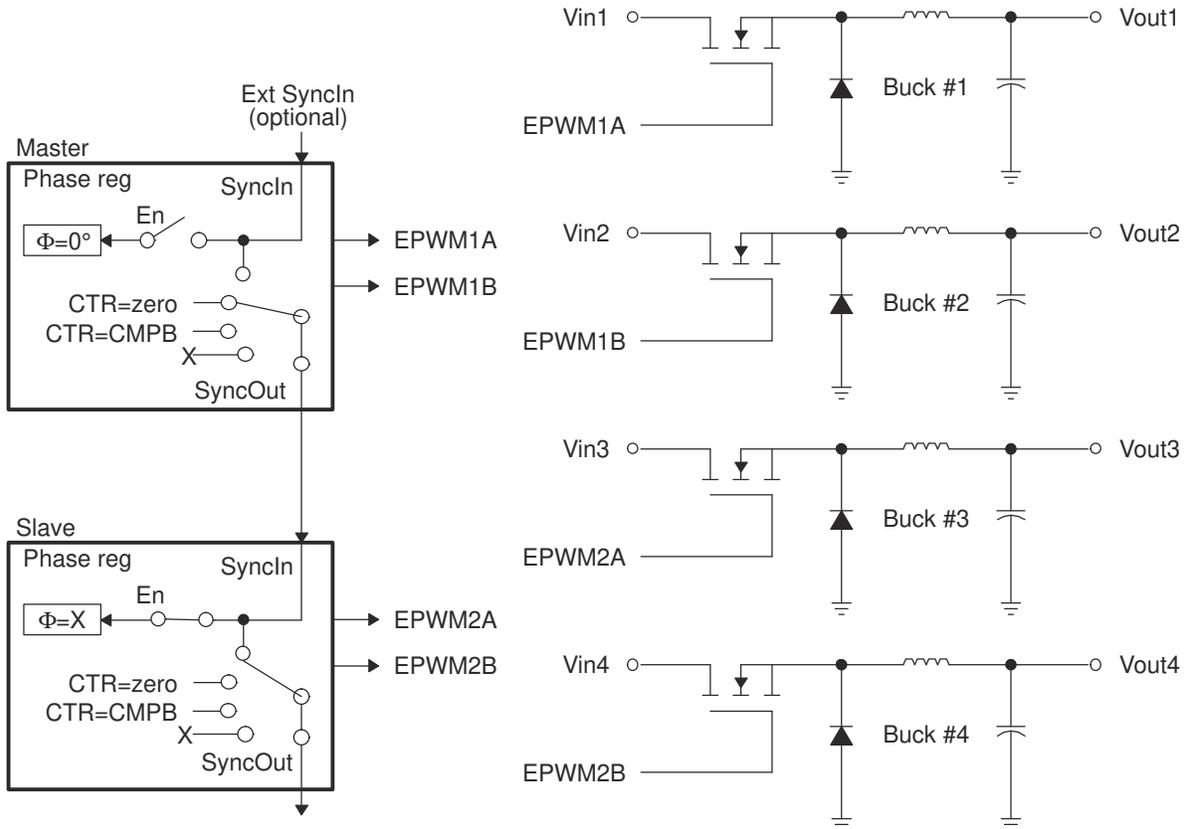


図 1-61. 4 つの降圧段を制御するための降圧波形 (注 : ここには 3 つの降圧段のみを示します。)

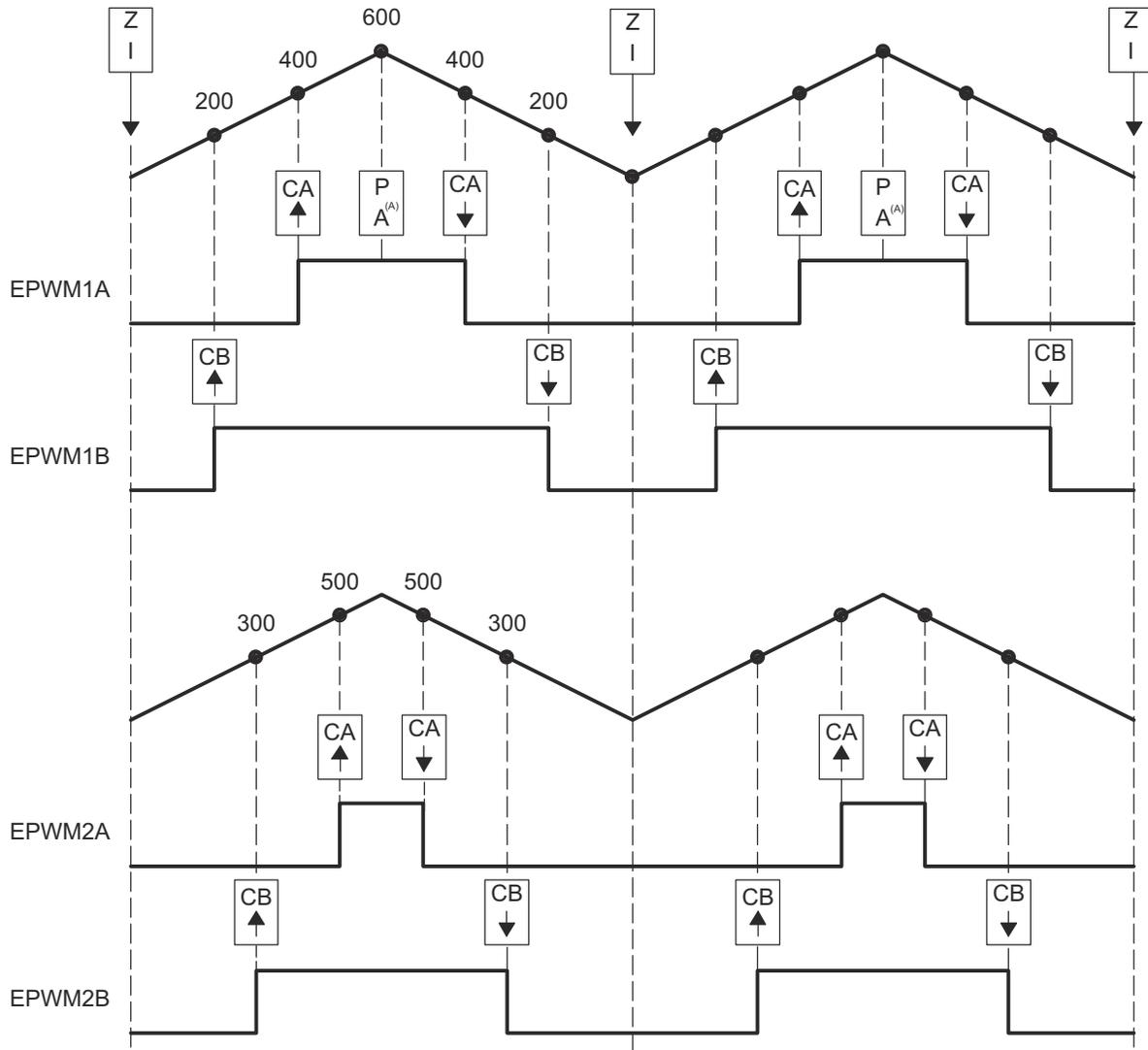
1.13.4 同じ周波数で複数の降圧コンバータを制御

同期させる必要がある場合、ePWM モジュール 2 はスレーブとして構成され、モジュール 1 の整数 (N) 倍の周波数で動作します。マスタからスレーブへの同期信号によって、これらのモジュールはロックされた状態に維持されます。図 1-62 に、そのような構成を示します。図 1-63 に、その構成によって生成される波形を示します。



A. $\phi = X$ は位相レジスタの値が「ドントケア」であることを示します。

図 1-62. 4 つの降圧段の制御(注 : $F_{PWM2} = N \times F_{PWM1}$)



A. ADC 変換を開始します。

図 1-63. 4 つの降圧段を制御するための降圧波形 (注 : $F_{PWM2} = F_{PWM1}$)

1.13.5 複数のハーフ H ブリッジ (HHB) コンバータの制御

複数のスイッチング素子を制御する必要があるトポロジにも、これらの同じ ePWM モジュールを使って対応できます。1 つの ePWM モジュールでハーフ H ブリッジ段を制御することもできます。この制御は複数の段に拡張できます。図 1-64 に、ステージ 2 がステージ 1 の整数 (N) 倍の周波数で動作できる 2 つの同期されたハーフ H ブリッジ段の制御を示します。図 1-65 に、図 1-64 に示す構成によって生成される波形を示します。

ePWM モジュール 2 (スレーブ) は、同期フロースルー用に構成されています。必要に応じて、この構成を使うことで、第 3 のハーフ H ブリッジを ePWM モジュール 3 によって制御でき、さらに最も重要なことには、マスタ ePWM モジュール 1 との同期を維持できます。

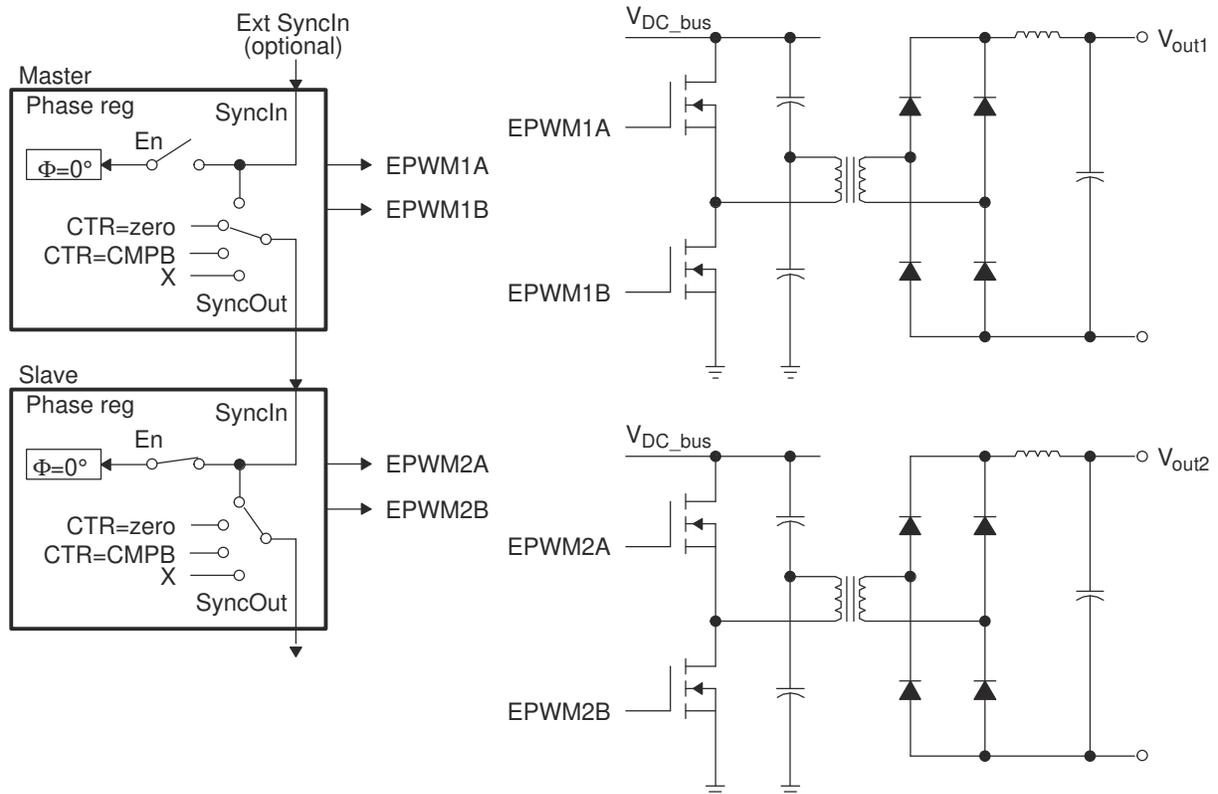


図 1-64. 2 つのハーフ H ブリッジ段の制御 ($F_{PWM2} = N \times F_{PWM1}$)

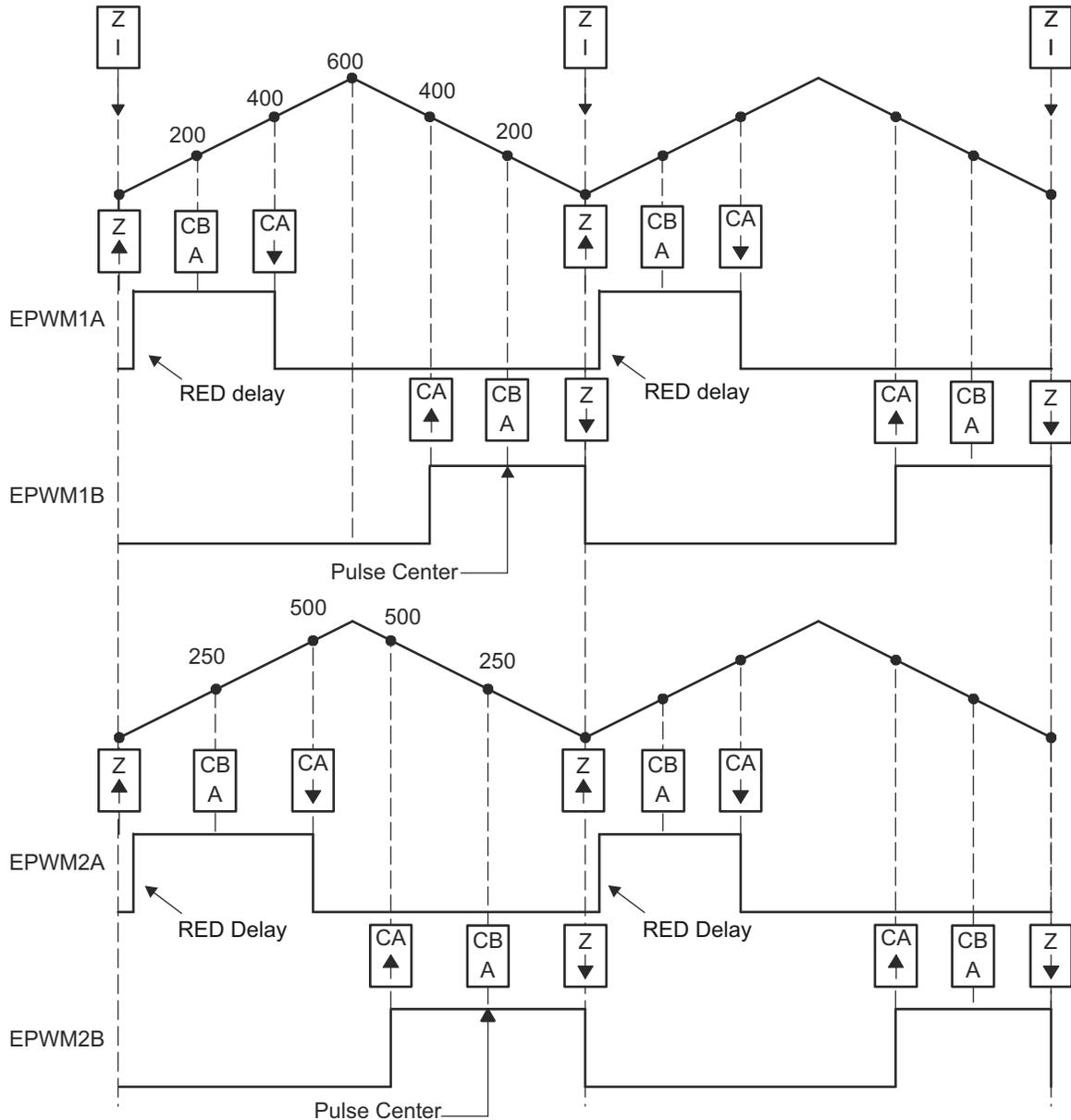


図 1-65. 2 つのハーフブリッジ段を制御するためのハーフブリッジ波形 (注: ここで、 $F_{PWM2} = F_{PWM1}$)

1.13.6 モーター (ACI および PMSM) 用デュアル 3 相インバータの制御

複数のモジュールで 1 つの電力段を制御するという考え方は、3 相インバータの場合に拡張できます。このような場合、インバータの各レッグに 1 つずつ、3 つの PWM モジュールを使って 6 つのスイッチング素子が制御されます。各レッグは同じ周波数でスイッチングする必要があり、すべてのレッグは同期している必要があります。1 つのマスタ + 2 つのスレーブの構成は、この要件に簡単に対応できます。図 1-66 に、6 つの PWM モジュールが 2 つの独立した 3 相インバータ (各インバータが 1 つのモーターを駆動) を制御する方法を示します。

前のセクションに示した例のように、各インバータを異なる周波数で動作させることができます (図 1-66 に示すように、モジュール 1 とモジュール 4 がマスタ)。または、1 つのマスタ (モジュール 1) と 5 つのスレーブを使うことで両方のインバータを同期させることもできます。この場合、モジュール 4、5、6 の周波数 (すべて同じ) は、モジュール 1、2、3 の周波数 (同様にすべて同じ) の整数倍にすることができます。

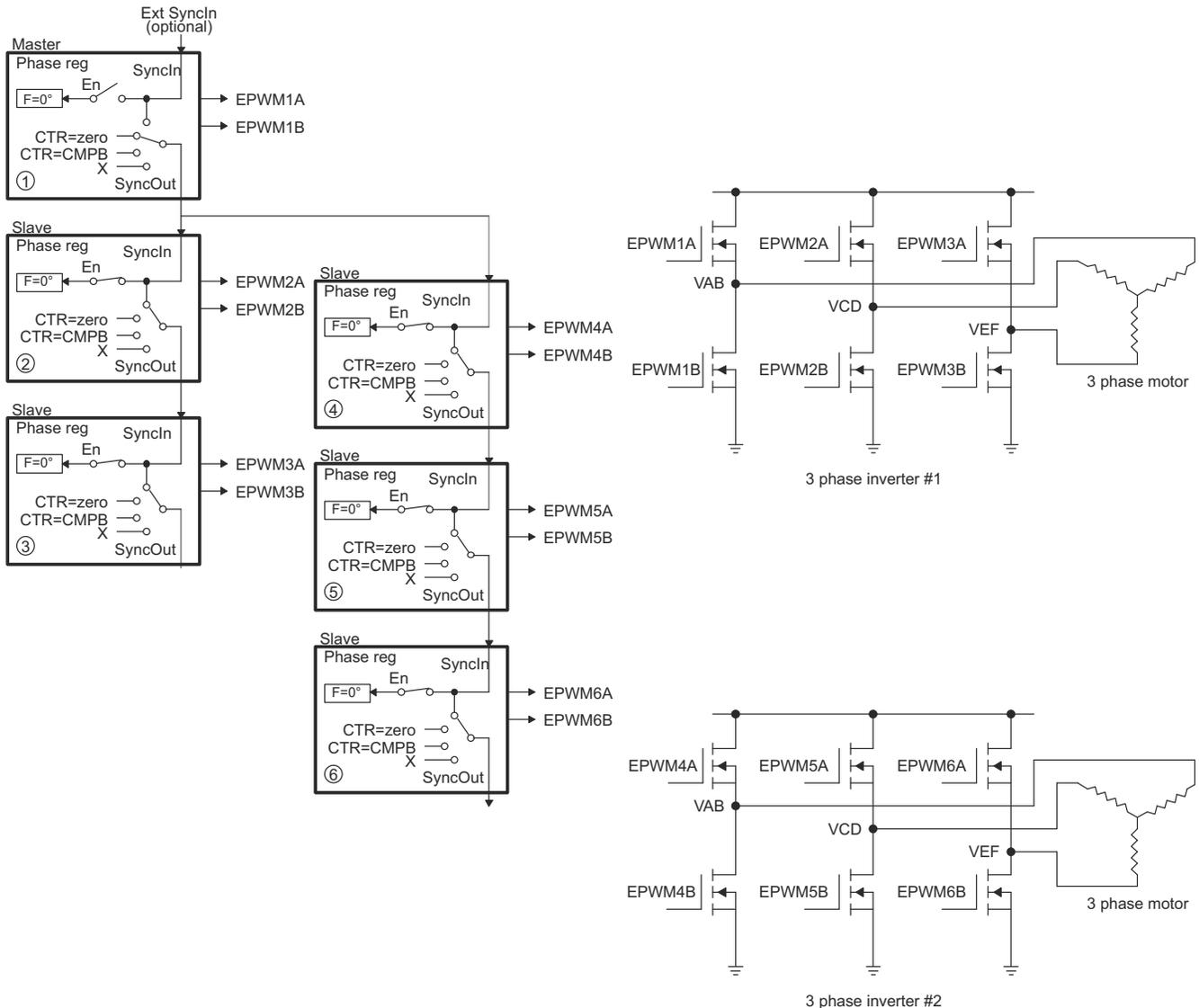


図 1-66. モーター制御で一般的に使用されているデュアル 3 相インバータ段の制御

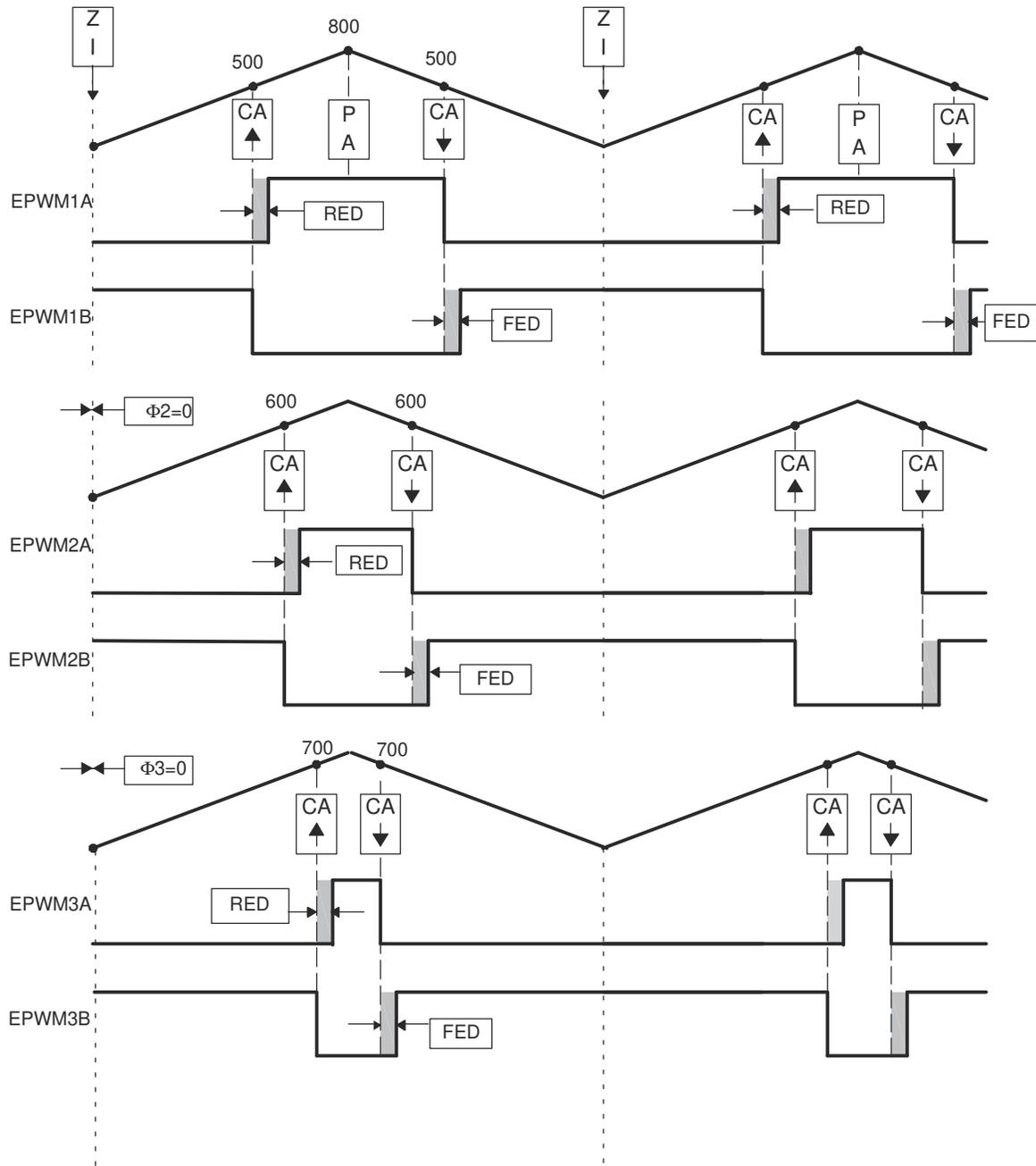


図 1-67. デュアル 3 相インバータ段を制御するための 3 相インバータ波形 (1 つのインバータのみを表示)

1.13.7 PWM モジュール間の位相制御を使用した実用的応用

これまで紹介した中で、位相レジスタ (TBPHS) を使用した例はありませんでした。TBPHS がゼロに設定されているか、値がドントケアであるかのどちらかでした。しかし、TBPHS に適切な値を設定することで、複数の PWM モジュールを使って、レグ (または段) 間の位相関係に基づいて動作する別の種類の電源トポロジに対応することもできます。「時間ベース・サブモジュール」セクションで説明したように、SyncIn パルスによって TBPHS レジスタの内容が TBCTR レジスタにロードされるように PWM モジュールを設定することができます。この概念を図示するため、[図 1-68](#) に、120° の位相関係 (つまり、スレーブの位相がマスタの位相より進んでいる) を持つマスタおよびスレーブ・モジュールを示します。

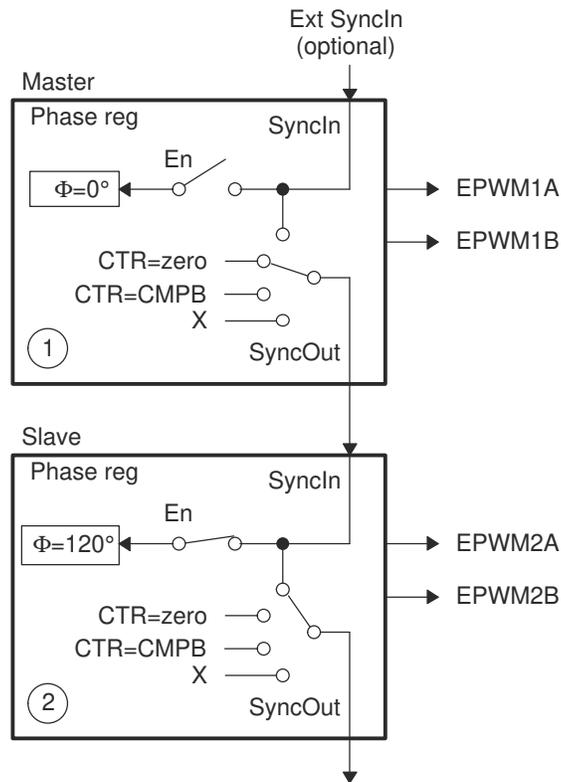


図 1-68. 位相制御のための 2 つの PWM モジュールの設定

[図 1-69](#) に、この構成に関連するタイミング波形を示します。ここで、マスタとスレーブの両方に対して TBPRD = 600 としています。スレーブの場合、TBPHS = 200 (つまり、 $200/600 \times 360^\circ = 120^\circ$) です。マスタが SyncIn パルス (CTR = PRD) を生成するたびに、TBPHS = 200 の値がスレーブの TBCTR レジスタにロードされるため、スレーブの時間ベースはマスタの時間ベースより常に 120° 進んでいます。

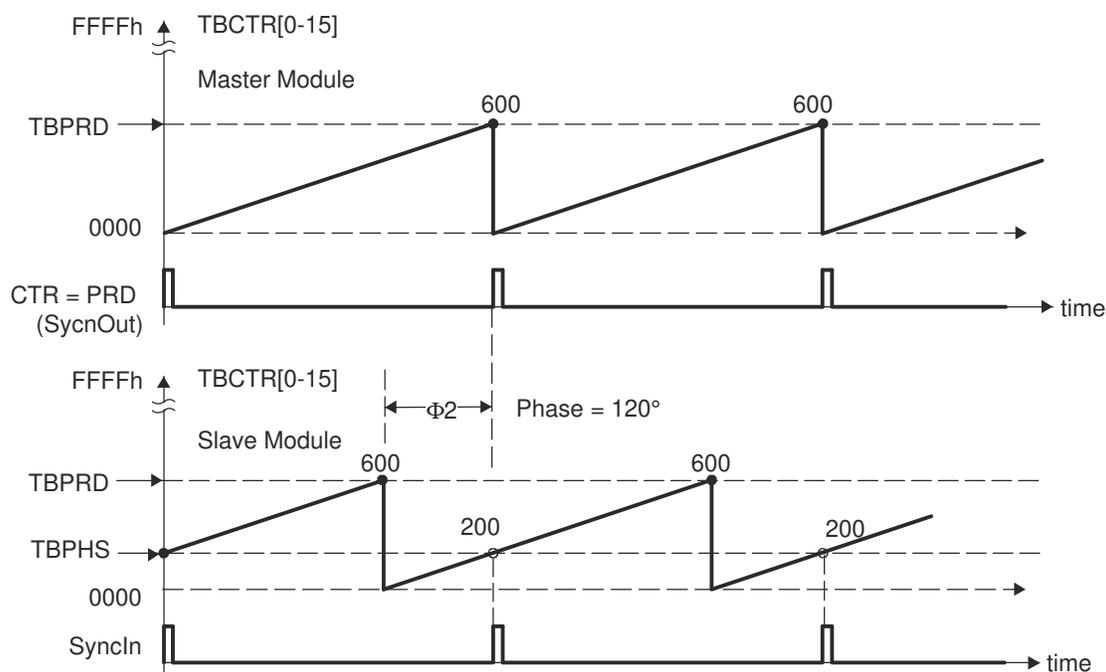


図 1-69. 2 つのモジュール間の位相制御に関連するタイミング波形

1.13.8.3 相インターリーブ DC/DC コンバータの制御

図 1-70 に、モジュール間の位相オフセットを使う一般的な電源トポロジを示します。このシステムは 3 つの PWM モジュールを使っており、モジュール 1 をマスタとして設定しています。正しく機能させるには、隣接モジュール間の位相関係を $F = 120^\circ$ にする必要があります。これは、スレーブの TBPHS レジスタ 2 および 3 に、周期値の 1/3 と 2/3 の値をそれぞれ設定することで実現されます。たとえば、周期レジスタに 600 カウンツの値がロードされた場合、TBPHS (スレーブ 2) = 200、TBPHS (スレーブ 3) = 400 となります。両方のスレーブ・モジュールがマスタ・モジュール 1 と同期します。

TBPHS の値を適切に設定することで、この概念は 4 つ以上の位相に拡張できます。以下の式に、N 相の TBPHS 値を示します。

$$TBPHS(N,M) = (TBPRD/N) \times (M - 1)$$

ここで

N = 位相の数

M = PWM モジュール番号

たとえば、3 相の場合 (N = 3)、TBPRD = 600、

$TBPHS(3,2) = (600/3) \times (2 - 1) = 200$ (つまり、スレーブ・モジュール 2 の位相値)

$TBPHS(3,3) = 400$ (つまり、スレーブ・モジュール 3 の位相値)

図 1-71 に、図 1-70 の構成の波形を示します。

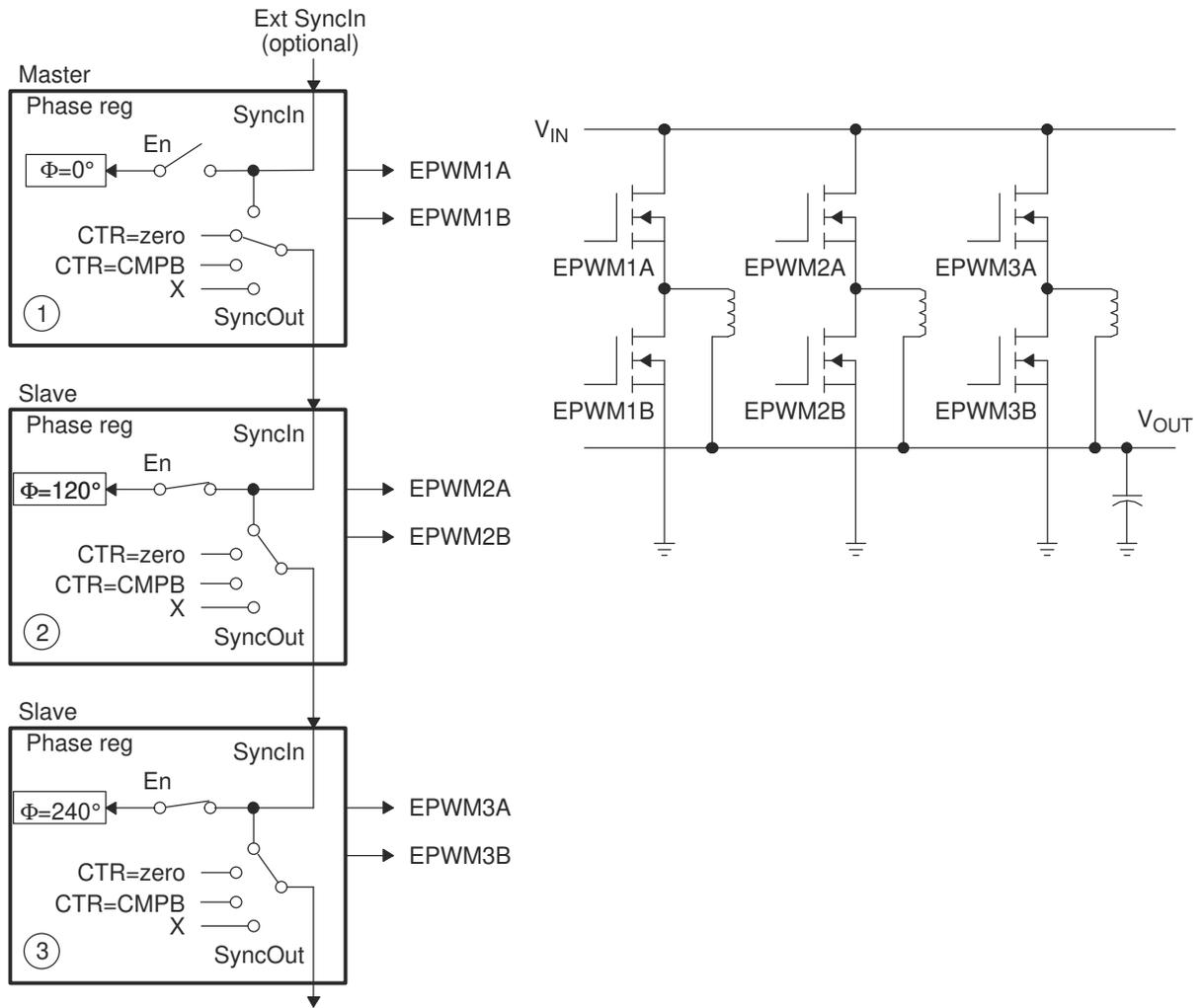


図 1-70. 3 相インターリーブ DC/DC コンバータの制御

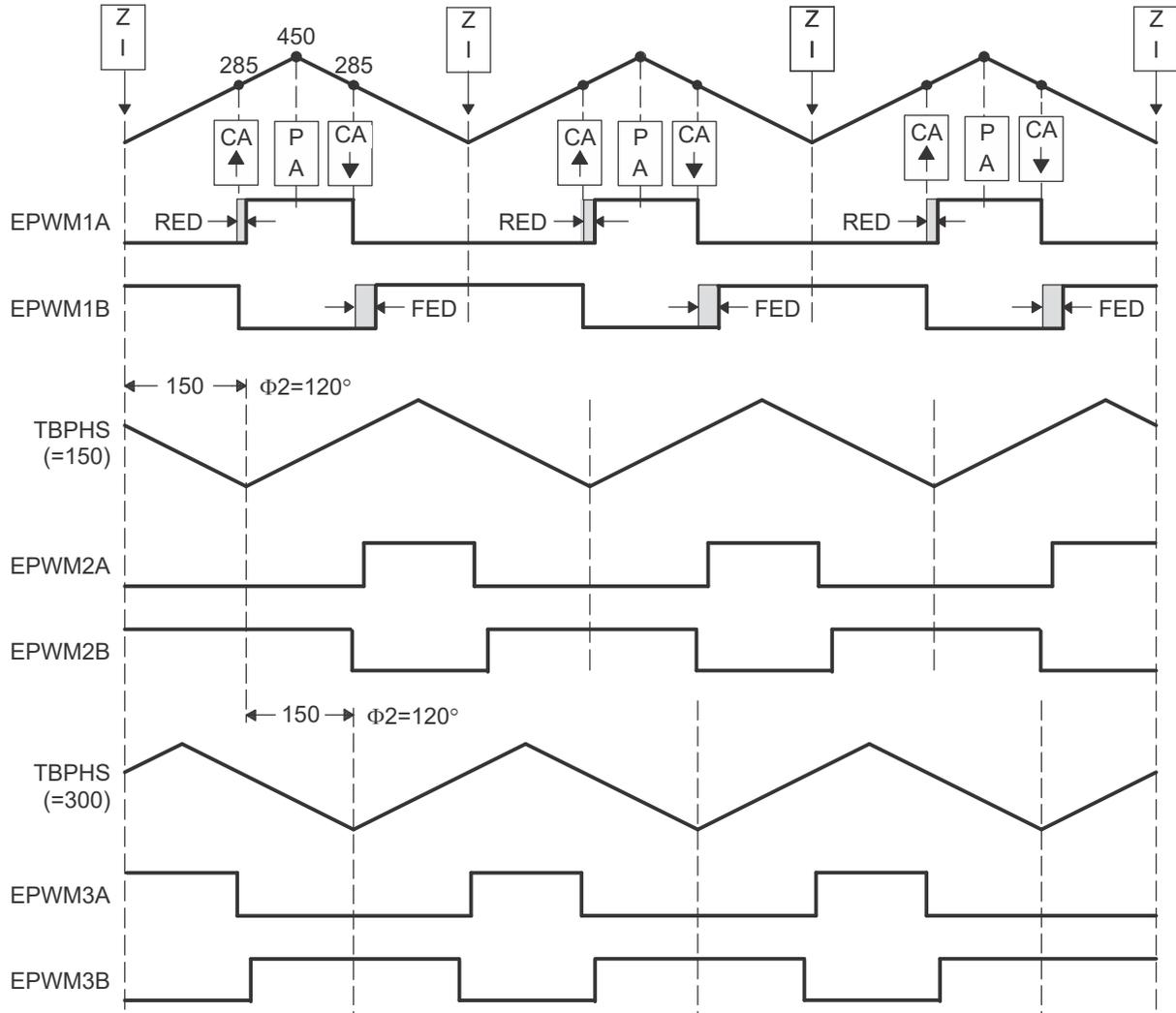


図 1-71. 3 相インターリーブ DC/DC コンバータ波形による 3 相インターリーブ DC/DC コンバータの制御

1.1.3.9 ゼロ電圧スイッチング・フル・ブリッジ (ZVSFB) コンバータの制御

図 1-72 に示す例は、レッグ (モジュール) 間の位相関係が静的 (一定) であることを想定しています。そのような場合、デューティ・サイクルを変調することで制御が行われます。サイクルごとに位相値を動的に変更することもできます。この機能は、位相シフト・フル・ブリッジまたはゼロ電圧スイッチング・フル・ブリッジとして知られる種類の電源トポロジを制御するために役立ちます。この場合、制御されるパラメータはデューティ・サイクルではなく (デューティ・サイクルは約 50% で一定に維持されます)、レッグ間の位相関係です。そのようなシステムは、1 つの出力段を制御するために 2 つの PWM モジュールのリソースを割り当てることで実装できます。その結果、4 つのスイッチング素子を制御する必要があります。図 1-73 に、フル H ブリッジを制御するために同期したマスタとスレーブのモジュールの組み合わせを示します。この場合、マスタとスレーブのモジュールはどちらも同じ PWM 周波数でスイッチングする必要があります。この位相は、スレーブ位相レジスタ (TBPHS) を使って制御されます。マスタ位相レジスタは使われないため、ゼロに初期化できます。

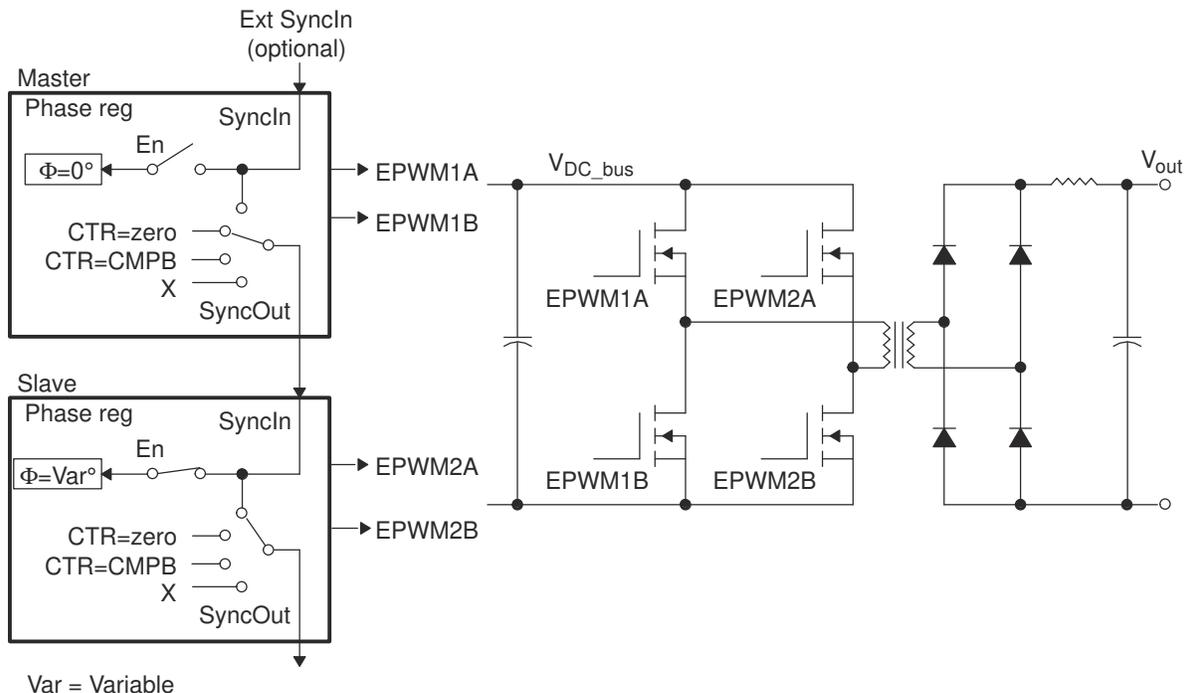


図 1-72. フル H ブリッジ段の制御 ($F_{PWM2} = F_{PWM1}$)

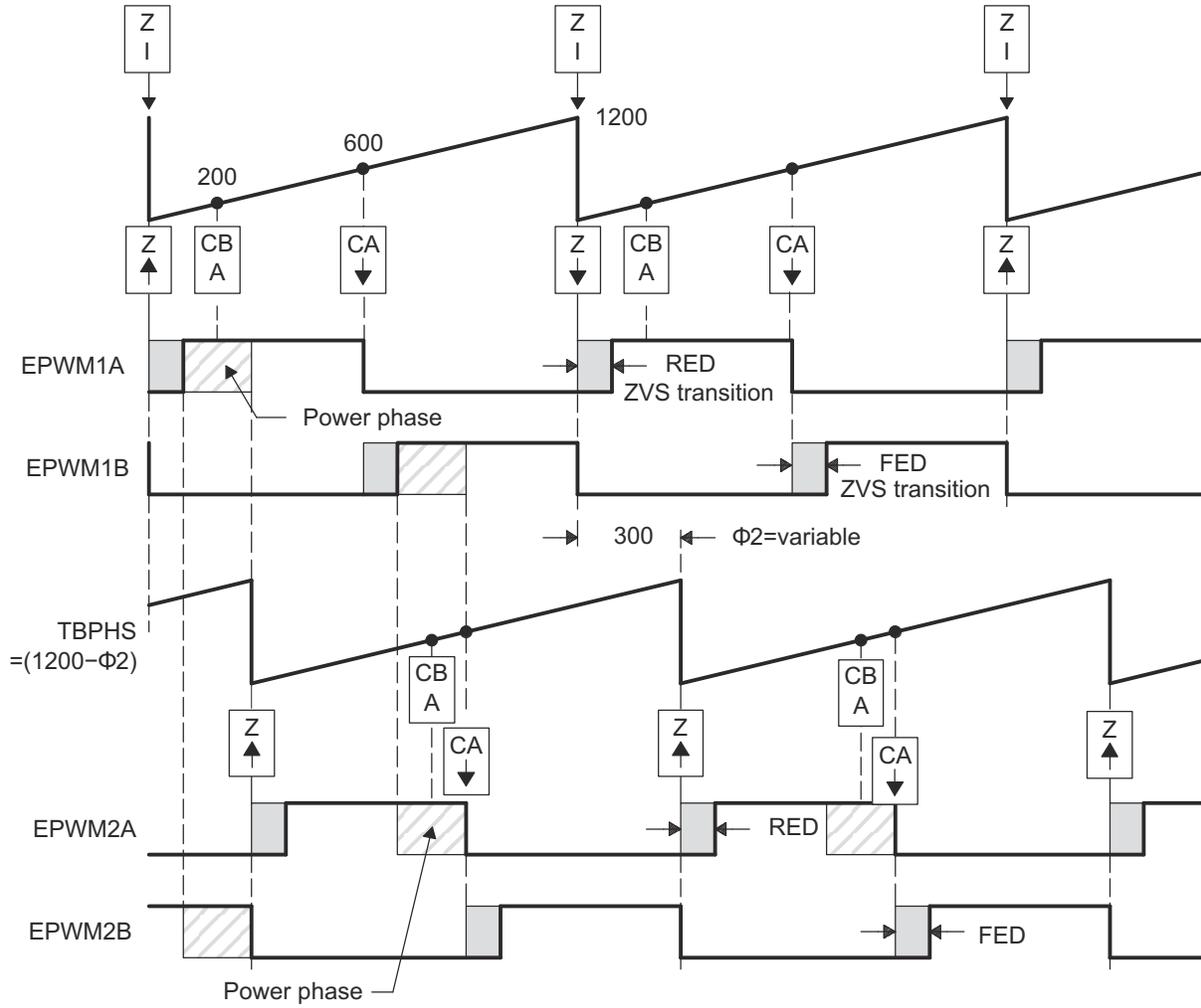


図 1-73. ZVS フル Hブリッジ波形

1.13.10 ピーク電流モード制御降圧モジュールの制御

ピーク電流制御手法には、過電流の自動制限、入力電圧変動の高速な補正、磁気飽和の低減など、多くの利点があります。図 1-74 に、降圧コンバータ・トポロジのための ePWM1A とオンチップ・アナログ・コンパレータの使い方を示します。出力電流は電流センス抵抗を使って検出され、オンチップ・コンパレータの正端子に供給されます。内蔵プログラマブル 12 ビット DAC は、コンパレータの負端子に基準ピーク電流を供給するために使えます。その代わりに、この入力に外部基準電圧を接続することもできます。コンパレータ出力は、デジタル比較サブモジュールへの入力です。ePWM モジュールは、検出された電流がピーク基準値に達するとすぐに ePWM1A 出力をトリップするように設定されています。サイクル・バイ・サイクル (サイクルごと) のトリップ機能が使われています。図 1-75 に、本構成によって生成された波形を示します。

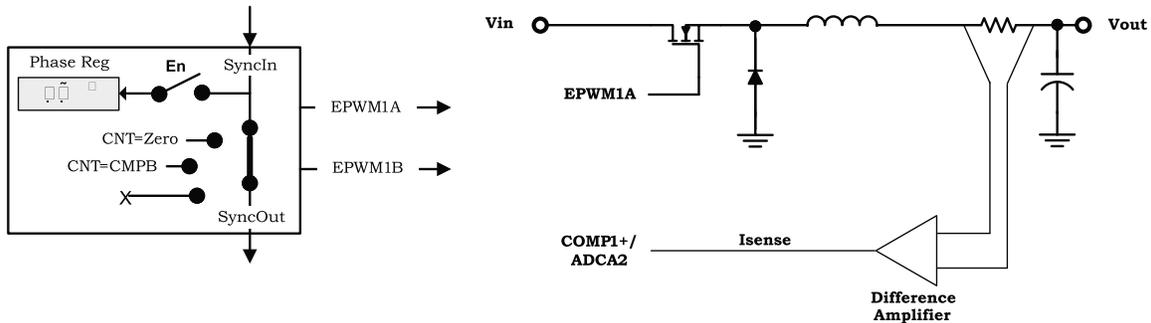


図 1-74. 降圧コンバータのピーク電流モード制御

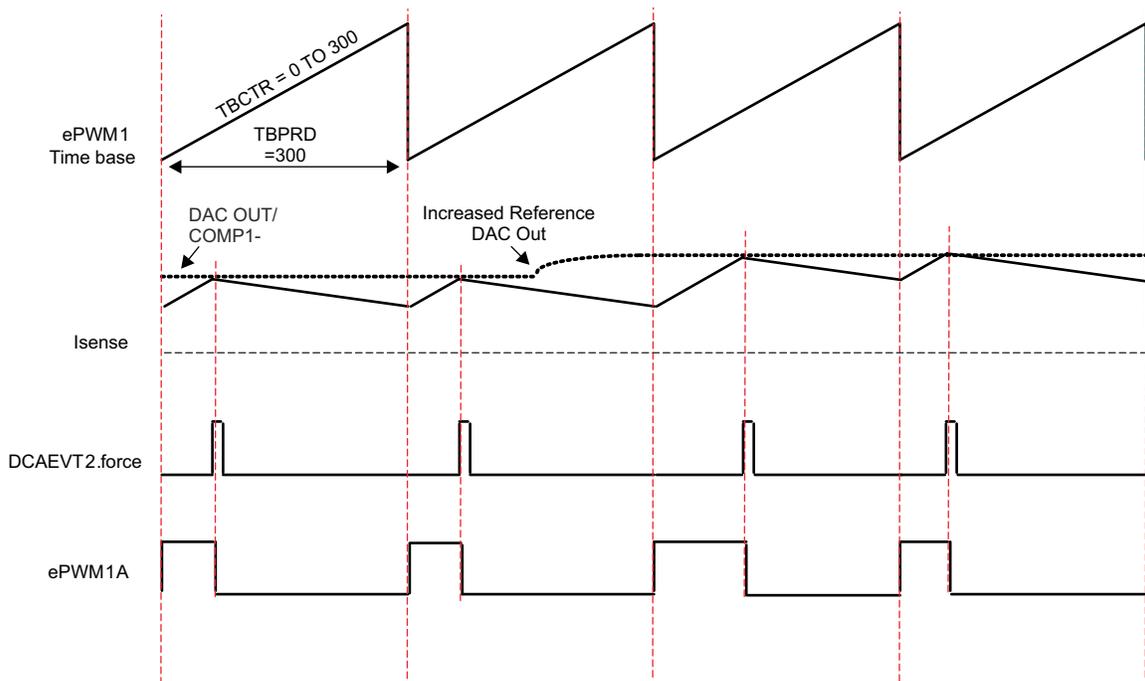
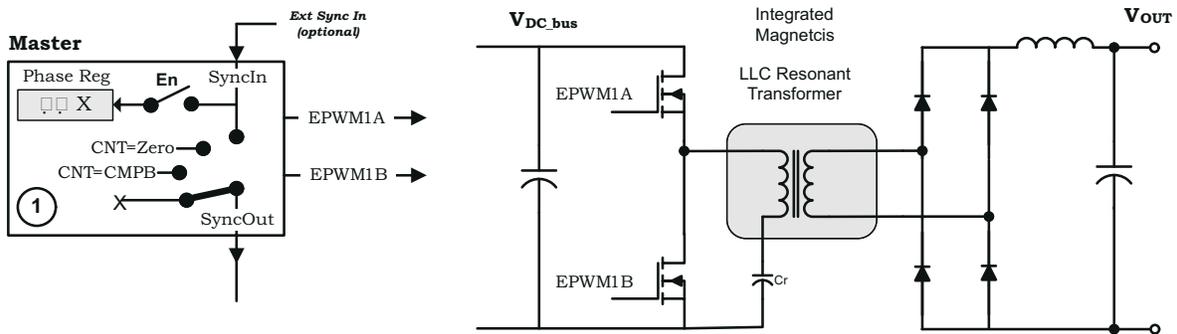


図 1-75. 降圧コンバータ制御のピーク電流モード制御波形

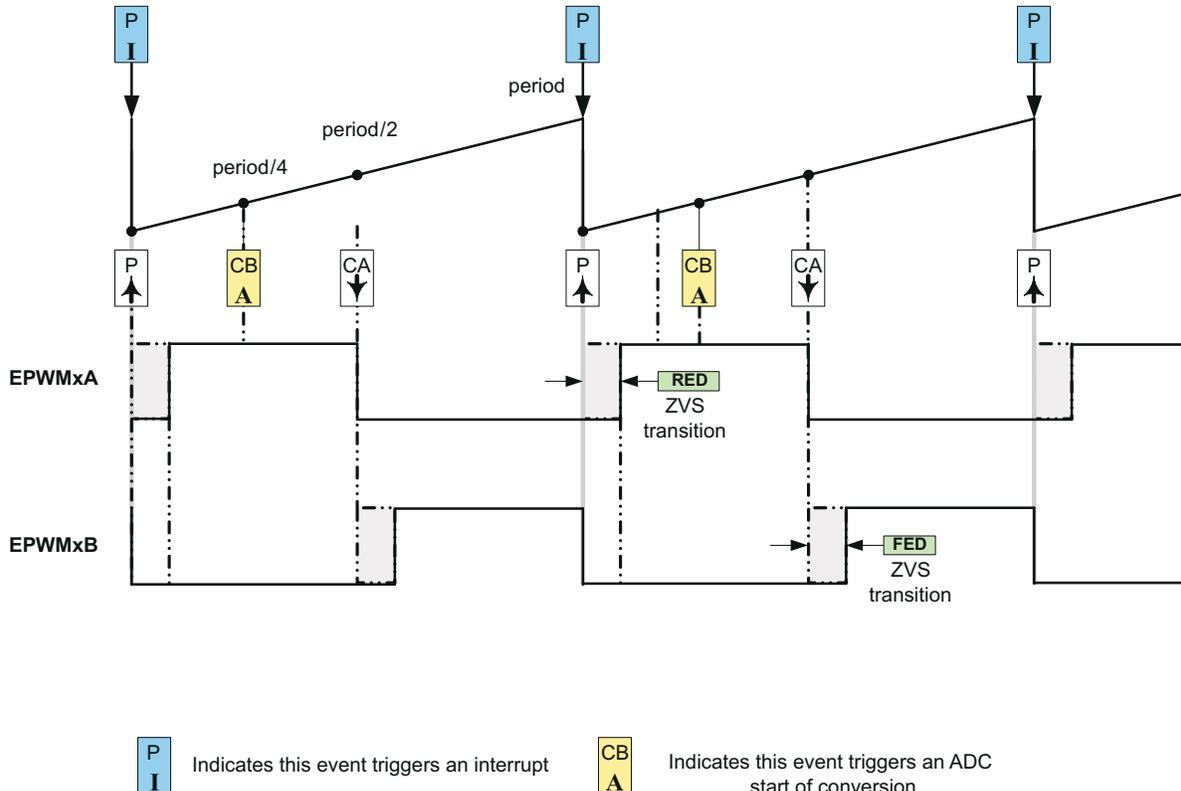
1.13.11 Hブリッジ LLC 共振コンバータの制御

共振コンバータのさまざまなトポロジは、長年にわたってパワー・エレクトロニクス分野で広く知られています。これらに加えてHブリッジ LLC 共振コンバータ・トポロジは、高効率と高電力密度が求められる多くのコンシューマ・エレクトロニクス・アプリケーションにおいて、最近人気を集めています。この例では、ePWM1のシングル・チャンネル構成について詳細に説明しますが、この構成は簡単にマルチチャンネルに拡張できます。この例では、制御されるパラメータはデューティ・サイクルではなく(デューティ・サイクルは約50%で一定に保たれます)、周波数です。デッドバンドは制御されず、300ns(つまり、100MHzのTBCLKで30)で一定に保たれますが、ソフト・スイッチングを行うのに十分な時間遅延を調整することで効率を向上させるため、ユーザーはデッドバンドをリアルタイムで更新できます。



NOTE $\Theta = X$ indicates value in phase register is 'don't care'

図 1-76. 2つの共振コンバータ段の制御



P I Indicates this event triggers an interrupt **CB A** Indicates this event triggers an ADC start of conversion

図 1-77. Hブリッジ LLC 共振コンバータの PWM 波形

1.14 レジスタ・ロック保護

コード暴走時の誤書き込みによって重要な ePWM レジスタのデータが破損しないように、レジスタ・ロック保護機能が追加されました。レジスタ EPWMLOCK にはロック・ビットの定義が含まれています (表 1-14 に、ロック・ビットと対応するレジスタを示します)。このレジスタには KEY フィールドもあります。このレジスタへの書き込みは、KEY フィールドに 0xa5a5 の値が書き込まれている場合にのみ成功します。詳細については、本レジスタの説明を参照してください。

表 1-14. ロック・ビットと対応するレジスタ

ビット・フィールド	定義	ロックされるレジスタ
HRLOCK	HRPWM レジスタ・セットのロック	HRCNFG、HRPWR、HRMSTEP、HRPCTL
GLLOCK	グローバル・ロード・レジスタ・セットのロック	GLDCTL、GLDCFG
TZCFGLOCK	TripZone レジスタ・セットのロック	TZSEL、TZDSEL、TZCTL、TZCTL2、TZCTLDCA、TZCTLDCB、TZEINT
TZCLRLOCK	TripZone クリア・レジスタ・セットのロック	TZCLR、TZCBCCLR、TZOSTCLR、TZFRC
DCLOCK	デジタル比較レジスタ・セットのロック	DCTRIPSEL、DCACTL、DCBCTL、DCCFCTL、DCCAPCTL、DCAHTRIPSEL、DCALTRIPSEL、DCBHTRIPSEL、DCBLTRIPSEL

注

同じレジスタに KEY フィールドが存在するため、KEY が一致する場合、32 ビット書き込みのみが成功します。このレジスタの上位半分または下位半分への 16 ビット書き込みは無視されます。

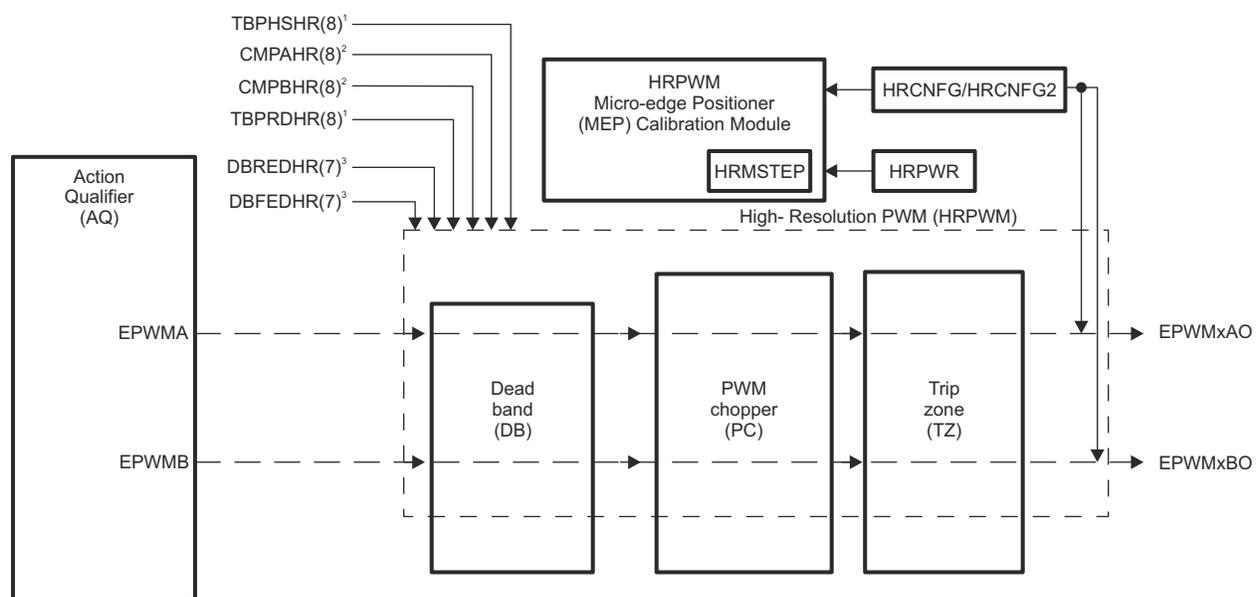
1.15 高分解能パルス幅変調器 (HRPWM)

図 1-78 に、HRPWM のブロック図を示します。このモジュールは、従来方式のデジタル・パルス幅変調器 (PWM) の時間分解能を拡張します。HRPWM は通常、PWM 分解能が約 9~10 ビットを下回る際に使用されま
す。HRPWM の主な機能は次のとおりです。

- 拡張された時間分解能
- デューティ・サイクルと位相シフトの両方の制御方式で使用可能
- 比較 A、比較 B、位相レジスタの拡張機能によるより細かい時間粒度制御またはエッジ位置設定
- PWM の A および B 信号パス (つまり EPWMxA および EPWMxB 出力) を使用して実装
- デッドバンド高分解能制御による、半周期クロック動作時の立ち下がりおよび立ち上がりエッジ遅延
- マイクロ・エッジ・ポジショナ (MEP) ロジックが、設計通りに動作しているかどうかをチェックするための自己チェック診断ソフトウェア・モード
- EPWMxA および EPWMxB 出力の高分解能出力の交換が可能
- EPWMxA 信号の反転出力を使用した EPWMxB 信号出力の高分解能出力が可能
- ePWM モジュールを備えたデバイスの EPWMxA および EPWMxB 出力の高分解能周期、デューティ、位相制御が可能

注

お使いのデバイスが高分解能周期対応 ePWM モジュールを備えているかどうかを確認するには、そのデバイスのデータシートを参照してください。



- ePWM 時間ベース (TB) サブモジュールから
- ePWM カウンタ比較 (CC) サブモジュールから
- ePWM デッドバンド (DB) サブモジュールから

図 1-78. HRPWM のブロック図

ePWM ペリフェラルは、D/A コンバータ (DAC) と数学的に等価な機能を実行するために使用されます。図 1-79 に示すように、従来の方法で生成された PWM の実効分解能は、PWM 周波数 (または周期) とシステム・クロック周波数の関数です。

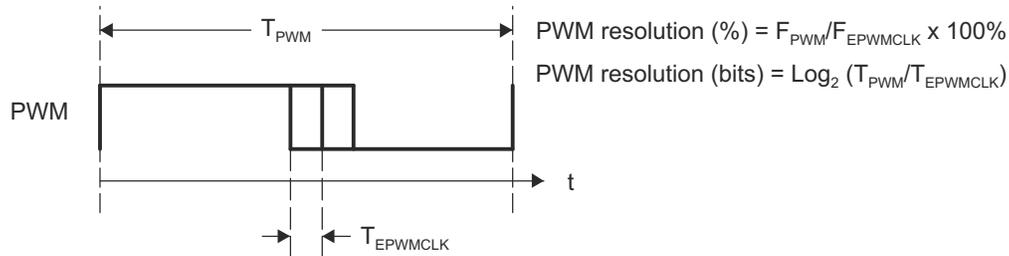


図 1-79. 従来の方法で生成された PWM の分解能の計算

PWM モードにおいて、必要とされる PWM 動作周波数では十分な分解能が得られない場合、HRPWM を使うことを考慮します。HRPWM によって得られる性能向上の例として、表 1-15 に、各種 PWM 周波数における分解能 (単位: ビット) で示します。これらの値は 180ps の MEP ステップ・サイズを前提としています。MEP の性能仕様の標準値と最大値については、本デバイスのデータ・マニュアルを参照してください。

表 1-15. PWM と HRPWM の分解能

PWM 周波数 (kHz)	通常分解能 (PWM) 100MHz の EPWMCLK		高分解能 (HRPWM)	
	ビット	%	ビット	%
20	12.3	0.02	18.1	0.000
50	11	0.05	16.8	0.001
100	10	0.1	15.8	0.002
150	9.4	0.15	15.2	0.003
200	9	0.2	14.8	0.004
250	8.6	0.25	14.4	0.005
500	7.6	0.5	13.4	0.009
1000	6.6	1	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2	11.4	0.036

アプリケーションによって異なる可能性がありますが、一般的な低周波数 PWM 動作 (250kHz 未満) では HRPWM は不要です。HRPWM 機能は、以下のような電力変換トポロジの高周波数 PWM 要件に最も有用です。

- 単相降圧、昇圧、フライバック
- 多相降圧、昇圧、フライバック
- 位相シフト・フルブリッジ
- D クラス・パワー・アンプの直接変調

1.15.1 HRPWM の動作の説明

HRPWM は、マイクロ・エッジ・ポジショナ (MEP) 技術に基づいています。MEP のロジックは、従来型 PWM ジェネレータの 1 つの粗いシステム・クロックを分周することで、エッジの位置を非常に細かく調整できます。時間ステップの精度は 150ps のオーダーです。特定のデバイスの代表的な MEP ステップ・サイズについては、そのデバイスのデータ・マニュアルを参照してください。すべての動作条件の下で設計どおりに MEP ロジックが動作しているかどうかを確認するため、HRPWM は自己チェック・ソフトウェア診断モードも備えています。ソフトウェアの診断と機能の詳細については、[セクション 1.15.1.7](#) を参照してください。

図 1-80 に、MEP ステップを単位として見た、1 つの粗いシステム・クロックとエッジ位置との関係を示します。これらは、比較 A 拡張レジスタ (CMPAHR) の 8 ビット・フィールドを使って制御されます。同じ動作ロジックが CMPBHR にも適用されます。

HRPWM 波形を生成するには、特定の周波数と極性の従来型 PWM を生成するように ePWM レジスタを設定します。HRPWM は、ePWM レジスタと連携して動作することで、エッジの分解能を拡張します。多くの設定の組み合わせが可能ですが、必要かつ実用的な組み合わせはごくわずかのみです。[セクション 1.15.1.8](#) で、これらの方法について説明しています。

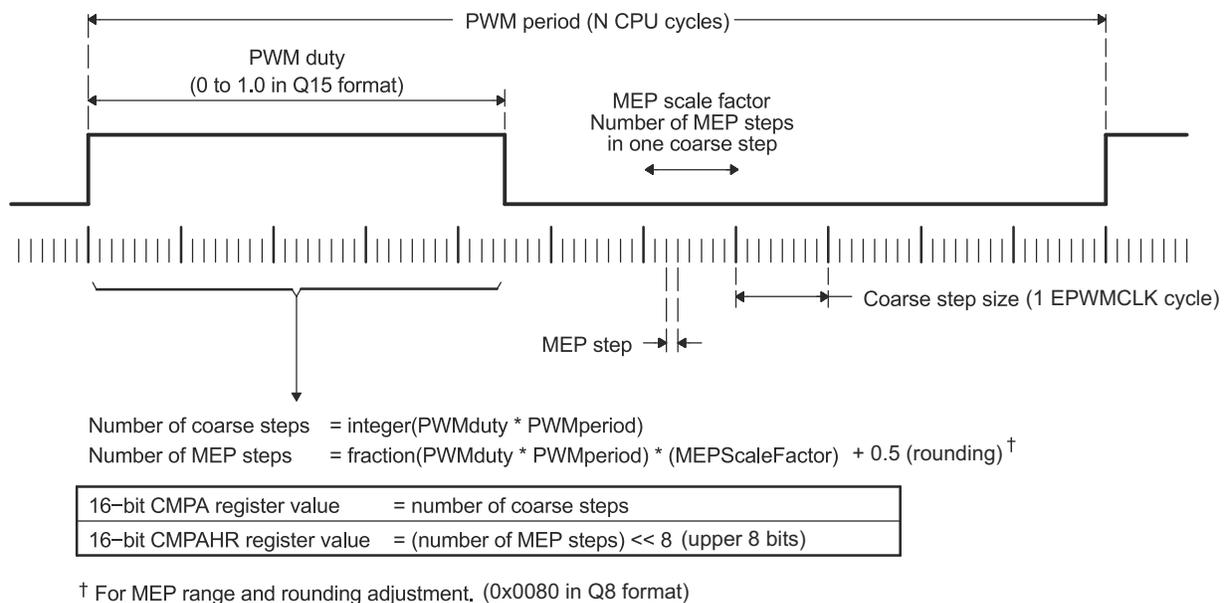
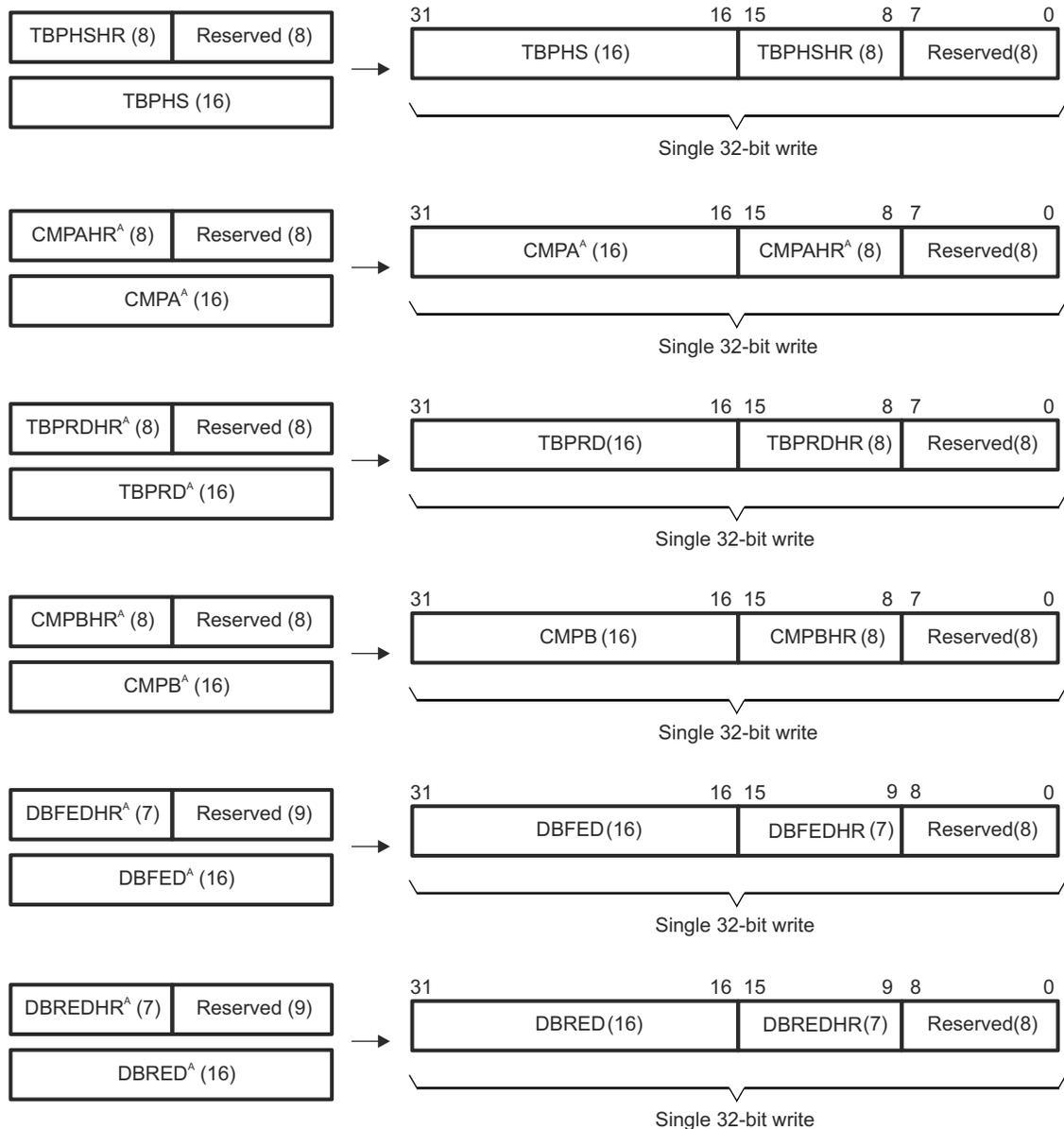


図 1-80. MEP を使った動作ロジック

1.15.1.1 HRPWM 機能の制御

HRPWM の MEP は 6 つの拡張レジスタによって制御されます。これらの HRPWM レジスタは、PWM 動作を制御するために使われる 16 ビットの TBPHS、TBPRD、CMPA、CMPBM、DBREDM、DBFEDM レジスタと連結されています。

- TBPHSHR - 時間ベース位相高分解能レジスタ
- CMPAHR - カウンタ比較 A 高分解能レジスタ。CMPAHR はチャンネル A の AQ 出力と組み合わせて使用するためのものであり、CMPA とは関係ありません。
- TBPRDHR - 時間ベース周期高分解能レジスタ(一部のデバイスで使用できます)。
- CMPBHR - カウンタ比較 B 高分解能レジスタ。CMPBHR はチャンネル B の AQ 出力と組み合わせて使用するためのものであり、CMPB とは関係ありません。
- DBREDHR - デッドバンド・ジェネレータ立ち上がりエッジ遅延高分解能レジスタ
- DBFEDHR - デッドバンド・ジェネレータ立ち下がりエッジ遅延高分解能レジスタ



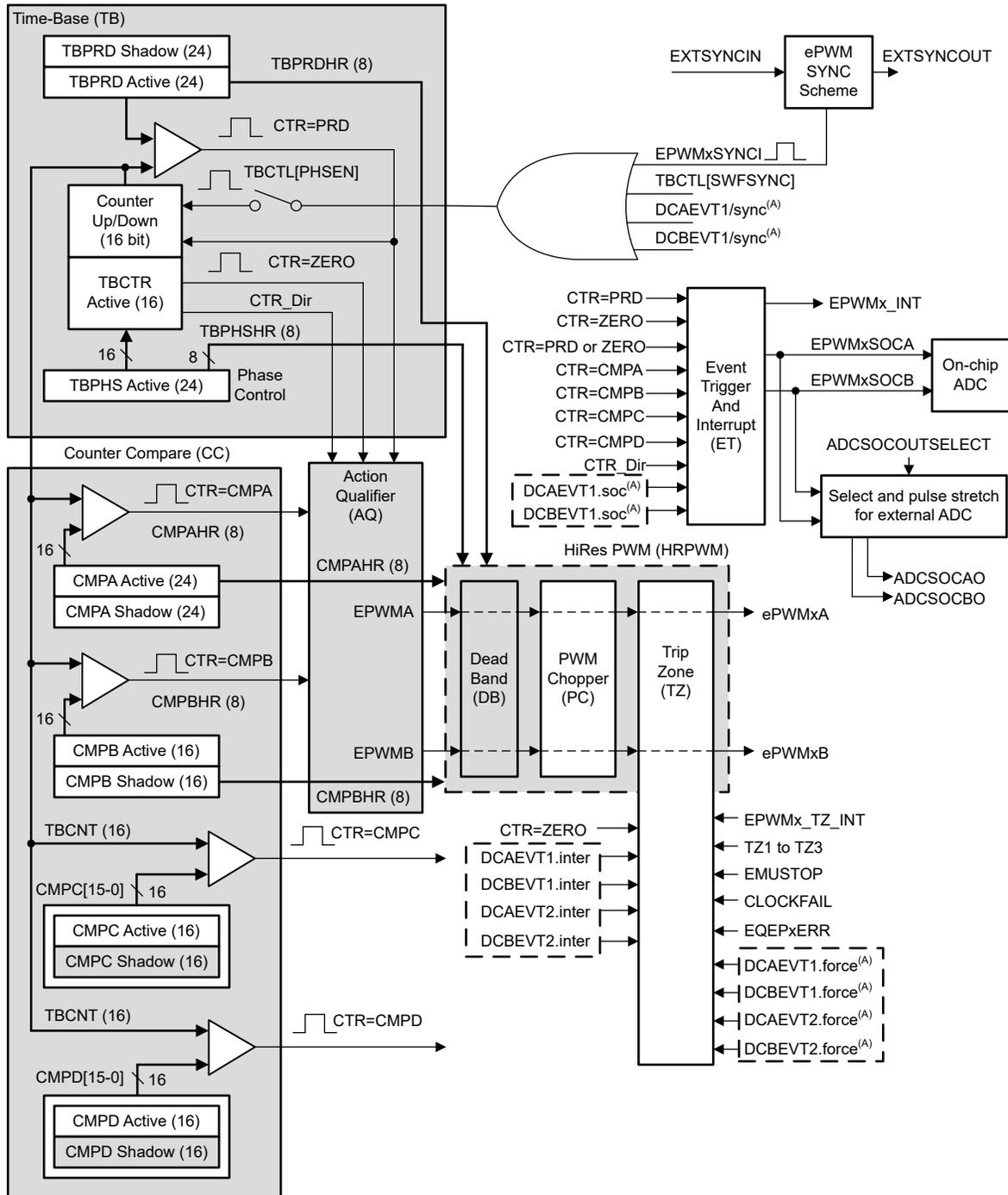
A. 使用しているデバイスによっては、これらのレジスタをミラーリングし、2つの異なるメモリ位置に書き込むこともできます。

図 1-81. HRPWM 拡張レジスタとメモリの構成

注

デッドバンドの立ち上がりエッジ遅延と立ち下がりエッジ遅延に関する HRPWM 機能は、デッドバンド半周期クロック動作中にのみ適用できます。同じ理由で、MEP ステップ数はデューティおよび位相高分解能レジスタの半分のサイズ [ビット 15 : 9] になっています。

HRPWM 機能は、チャンネル A および B の PWM 信号パスを使って制御されます。デッドバンド信号パスでの HRPWM のサポートは、HRCNFG2 レジスタを適切に設定することで利用できます。図 1-82 に、HRPWM の 8 ビット拡張レジスタとのインターフェイスを示します。



A. これらのイベントは、TRIPIN 入力のレベルに基づいて、ePWM デジタル比較 (DC) サブモジュールによって生成されます。

図 1-82. HRPWM システムのインターフェイス

1.15.1.2 HRPWM のソース・クロック

各 HRPWM モジュールには、それぞれの EPWMxCLK からクロックが供給されます。HRCAL は独立したクロックを備えています。たとえば、HRPWM1 には EPWM1CLK からクロックが供給され、HRPWM2 には EPWM2CLK からクロックが供給されます。図 1-83 に、それぞれの ePWM クロック・ソースからクロックが供給される HRCAL および HRPWM モジュールを示します。

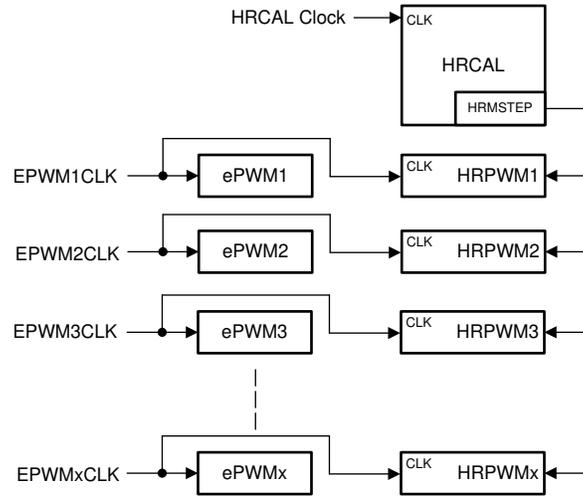


図 1-83. HRPWM と HRCAL のソース・クロック

1.15.1.3 HRPWM の設定

ePWM が従来型 PWM (特定の周波数と極性) を出力するように設定されている場合、その特定の ePWM モジュール・レジスタ空間の HRCNFG レジスタに値を書き込むことで、HRPWM が設定されます。このレジスタでは、以下の設定を選択できます。

- エッジ・モード** この MEP は、立ち上がりエッジ (RE)、立ち下がりエッジ (FE)、両方のエッジ (BE) を同時に精密に位置制御するように設定できます。FE と RE は、デューティ・サイクル制御 (CMPA または CMPB 高分解能制御) を必要とする電源トポロジに使用されます。一方 BE は、位相シフト・フル・ブリッジ (TBPHS または TBPRD 高分解能制御) などの位相シフトを必要とするトポロジに使用されます。
- 制御モード** MEP は、CMPAHR/CMPBHR レジスタ (デューティ・サイクル制御の場合) と TBPHSHR レジスタ (位相制御の場合) のどちらかで制御されるように設定されます。RE または FE 制御モードは、CMPAHR または CMPBHR レジスタと組み合わせて使用できます。BE 制御モードは、TBPHSHR レジスタと組み合わせて使用できます。MEP を TBPRDHR レジスタで制御 (周期制御) する場合、それぞれの高分解能レジスタを使用してデューティ・サイクルと位相も制御できます。
- シャドウ・モード** このモードは、通常の PWM モードと同じシャドウイング (ダブル・バッファリング) 機能を備えています。この機能は、CMPAHR、CMPBHR、TBPRDHR レジスタで動作している場合にのみ有効であり、CMPA/CMPB レジスタの通常のロード機能と同じものを選択できます。TBPHSHR を使う場合、この機能は何の影響も及ぼしません。
- 高分解能 B 信号制御** 高分解能 ePWMxA 信号の反転信号を ePWMxB ピンに出力することで、ePWM チャネルの B 信号パスは高分解能出力を生成できます。タイプ 2 またはタイプ 4 HRPWM モジュールは、A 信号パスから独立して、B 信号パスでも同様に高分解能機能を有効化できます。
- ePWMxA 出力と ePWMxB 出力の交換** このモードを使うと、高分解能 A および B 出力を交換できます。このモード選択により、「A と B の出力を変更しない」と「A 出力を B に出力し、B 出力を A に出力する」のどちらかが可能です。

自動変換モード

このモードは、係数最適化 (SFO) ソフトウェアと組み合わせてのみ使用します。タイプ 4 HRPWM モジュールについては、CMPAHR を例とした自動変換モードの説明を以下に示します。自動変換が有効化されている場合、 $CMPAHR = \text{fraction}(PWMduty * PWMperiod) \ll 8$ です。係数最適化ソフトウェアは、バックグラウンド・コードで MEP 係数を計算し、粗ステップあたりの MEP ステップの計算値で HRMSTEP レジスタを自動的に更新します。次に、MEP 較正モジュールは HRMSTEP および CMPAHR レジスタの値を使って、分数デューティ・サイクルで表される適切な MEP ステップ数を自動的に計算し、それに応じて高分解能 ePWM 信号エッジを移動させます。自動変換が無効化されている場合、CMPAHR レジスタはタイプ 0 HRPWM モジュールと同様に動作し、 $CMPAHR = (\text{fraction}(PWMduty * PWMperiod) * MEP \text{ 係数} + 0.5) \ll 8$ となります。このモードでは、すべての計算はコードによって実行される必要があり、HRMSTEP レジスタは無視されます。高分解能周期の自動変換は、高分解能デューティ・サイクルの自動変換と同様に動作します。高分解能周期モードでは、自動変換は常に有効化されている必要があります。

注

HRPWM モジュールがアップ・ダウン・カウンタ・モードに設定されている場合、HRPWM レジスタのシャドウ・モードは、ゼロと周期値の両方でロードするように設定する必要があります。ユーザーからの新しい値は、CTR = ゼロの場合にのみシャドウ・レジスタにロードされますが、それらのレジスタのシャドウ・モードは、ゼロと周期値の両方に設定する必要があります。CTR = PRD イベントは、HRPWM モジュール内の特定の内部ロジックのために使用されます。

自動変換モードは、CMPBHR、DBREDHR、DBFEDHR の計算を実行します。係数最適化ソフトウェアは、バックグラウンド・コードで MEP 係数を計算し、粗ステップあたりの MEP ステップの計算値で HRMSTEP レジスタを自動的に更新します。次に、MEP 較正モジュールは HRMSTEP および CMPAHR レジスタの値を使って、分数成分で表される適切な MEP ステップ数を自動的に計算し、それに応じて高分解能 ePWM 信号エッジを移動させます。自動変換が無効化されている場合、CMPBHR は CMPAHR と同様に動作します。 $CMPBHR = (\text{fraction}(PWMduty * PWMperiod) * MEP \text{ 係数} + 0.5) \ll 8$ 。

アプリケーションがこれらのモジュールのいずれかにアクセスする必要がある場合、ePWM1 と HRPWM の両方の保護を無効化する必要があります。

1.15.1.4 デッドバンド立ち上がりエッジおよび立ち下がりエッジ遅延での高分解能の設定

ePWM が従来型 PWM (特定の周波数と極性、半周期クロック・モードでデッドバンドが有効) を出力するように設定されている場合、その特定の ePWM モジュール・レジスタ空間の HRCNFG2 レジスタを設定することで、デッドバンド RED および FED ラインでの高分解能動作が有効化されます。このレジスタでは、以下の設定を選択できます。

- エッジ・モード** この MEP は、デッドバンドの立ち上がりエッジ (RED)、デッドバンドの立ち下がりエッジ (FED)、両方のエッジ (DBRED 信号の立ち上がりエッジと DBFED 信号の立ち下がりエッジ) を同時に精密に位置制御するように設定できます。
- 制御モード** 高分解能モードで DBRED および DBFED のアクティブ・レジスタにシャドウ値をロードする時間イベントを選択します。その ePWM の DBCTL[LOADREDMODE] および DBCTL[LOADFEDMODE] ビットでの選択と一致するようにパルスを選択します。

1.15.1.5 動作原理

MEP ロジックは、255 (8 ビット) の離散的な時間ステップのいずれかにエッジを配置できます (代表的な MEP ステップ・サイズについては、本デバイスのデータ・マニュアルを参照してください)。時間ステップが適用され、幅広い PWM 周波数、システム・クロック周波数、その他の動作条件にわたってエッジ配置の精度が維持されるように、この MEP は TBM および CCM レジスタと連携して動作します。表 1-16 に、HRPWM でサポートされる動作周波数の代表的な範囲を示します。

表 1-16. MEP ステップ、PWM 周波数、分解能の関係

システム (MHz)	EPWMCLK ごとの MEP ステップ (1) (2) (3)	PWM の最小値 (Hz) (4)	PWM の最大値 (MHz)	最大値での分解能 (ビット) (5)
60.0	93	916	3.00	10.9
70.0	79	1068	3.50	10.6
80.0	69	1221	4.00	10.4
90.0	62	1373	4.50	10.3
100.0	56	1526	5.00	10.1

- (1) TBCLK = EPWMCLK。
- (2) 180ps の MEP 時間分解能に基づくデータの表 (これはサンプル値です。MEP 制約については、本デバイスのデータシートを参照してください)。
- (3) この例では、適用 MEP ステップ = $T_{EPWMCLK}/180ps$ です。
- (4) PWM 周波数の最小値は、周期の最大値 (TBPRD = 65535) に基づいています。PWM モードは非対称のアップ・カウントです。
- (5) 分解能 (単位: ビット) は、決められた PWM 周波数の最大値に対して与えられます。

1.15.1.5.1 エッジ位置調整

注

以下の例は、[CMPA : CMPAHR] レジスタの組み合わせを使用して示しています。デューティ・サイクル制御に [CMPBM : CMPBHRM] を使おうとする場合、動作原理と式は同じです。

一般的な電力制御ループでは、デジタル・コントローラがデューティ・コマンドを発行します。そのコマンドは通常、単位あたりまたはパーセンテージの数値で表されます。特定の動作点において、求められるデューティ・サイクルが 0.405 (40.5%) のオン時間であり、必要なコンバータ PWM 周波数が 1.25MHz であると仮定します。100MHz のシステム・クロックを使用する従来型の PWM 生成では、デューティ・サイクルの選択肢は 40.5% 付近にあります。図 1-84 に示すように、32 カウント (デューティ = 40%) の比較値は、40.5% に最も近い実現可能な値です。これは、目標値である 324ns に対して、320ns のエッジ位置に相当します。表 1-17 に、このデータを示します。

MEP を利用することで、目標値である 324ns にさらに近いエッジ位置を実現できます。表 1-17 は、CMPA 値に加えて、MEP (CMPAHR レジスタ) の 22 ステップでエッジが 323.96ns に配置されることを示しています。その結果、誤差はほぼゼロになります。この例では、MEP が 180ps のステップ分解能を持っていると仮定しています。

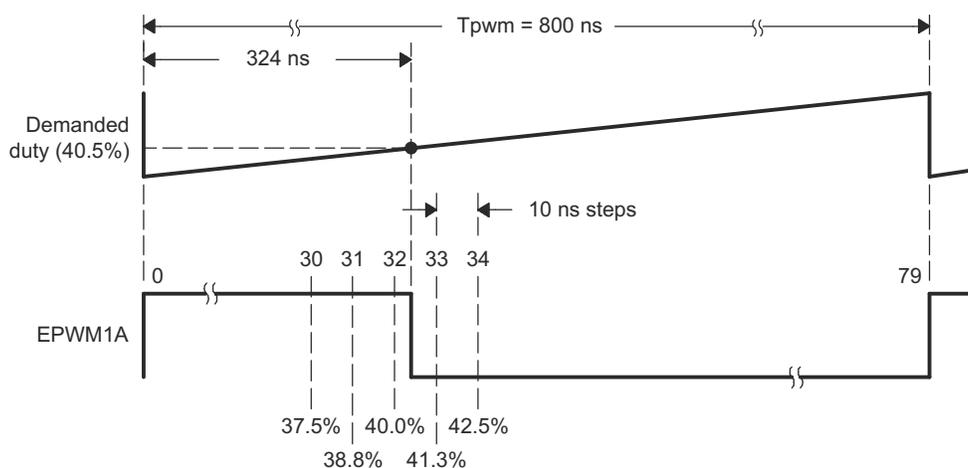


図 1-84. 要求デューティ = 40.5% のために必要な PWM 波形

表 1-17. CMPA とデューティとの関係 (左)、および [CMPA : CMPAHR] とデューティとの関係 (右)

CMPA (カウント) ⁽¹⁾ ⁽²⁾ ⁽³⁾	デューティ (%)	High 時間 (ns)	CMPA (カウント)	CMPAHR (カウント)	デューティ (%)	High 時間 (ns)
28	35.0	280	32	18	40.405	323.24
29	36.3	290	32	19	40.428	323.42
30	37.5	300	32	20	40.450	323.60
31	38.8	310	32	21	40.473	323.78
32	40.0	320	32	22	40.495	323.96
33	41.3	330	32	23	40.518	324.14
34	42.5	340	32	24	40.540	324.32
			32	25	40.563	324.50
必須			32	26	40.585	324.68
32.40	40.5	324	32	27	40.608	324.86

- (1) 上記の例で前提とされている MEP ステップ・サイズ = 180ps。MEP の標準値と最大値については、個々のデバイスのデータシートを参照してください。
- (2) TBCLK = 100MHz、10ns
- (3) PWM 周期レジスタ値が 80 カウントの場合、PWM 周期 = $80 \times 10\text{ns} = 800\text{ns}$ 、PWM 周波数 = $1/800\text{ns} = 1.25\text{MHz}$ 。

1.15.1.5.2 スケーリングに関する考慮事項

エッジを時間的に正確に配置する方法は、CMPA および MEP (CMPAHR) レジスタの標準的なリソースを使って実証済みです。しかし、実用的なアプリケーションでは、単位あたりの (分数) デューティ・サイクルから、[CMPA : CMPAHR] レジスタの組み合わせに書き込まれる最終的な整数 (非分数) 表現まで、CPU にシームレスに割り当てる必要があります。

これを行うため、関連するスケーリングまたはマッピング手順を最初に検討します。制御ソフトウェアでは、デューティ・サイクルを単位あたりまたはパーセンテージで表現するのが一般的です。これには、クロック数または High 時間 (単位: ナノ秒 (ns)) で表された最終的な絶対デューティ・サイクルを考慮せずに、必要なすべての算術計算を実行できるという利点があります。さらに、そうすることで、各種 PWM 周波数で実行する各種コンバータ・タイプ間でのコードの移植性が向上します。

マッピング手順を実装するには、2 段階のスケーリング手順が必要です。

この例の前提:

TBCLK	= 10ns (100MHz)
PWM 周波数	= 1.25MHz (1/800ns)
必要な PWM デューティ・サイクル、 PWMDuty	= 0.405 (40.5%)
粗ステップを単位とする PWM 周期、 PWMPeriod (800ns/10ns)	= 80
180ps (10n/180ps) での粗ステップあたりの MEP ステップ数、 MEP_ScaleFactor	= 55
CMPAHR を 1~255 の範囲内に維持し、端数を丸めるための定数 (デフォルト値)	= 0.5 (Q8 形式で 0080h)

ステップ 1 : CMPA レジスタのパーセンテージ - 整数デューティ値変換

CMPA レジスタ値	= $\text{int}(\text{PWMDuty} * \text{PWMPeriod})$ (int は整数部を意味します。)
	= $\text{int}(0.405 * 80)$
	= $\text{int}(32.4)$
CMPA レジスタ値	= 32 (20h)

ステップ 2 : CMPAHR レジスタの分数値変換

CMPAHR	= $(\text{frac}(\text{PWMDuty} * \text{PWMPeriod}) * \text{MEP_ScaleFactor} + 0.5) \ll 8$ (frac は分数部を意味します。)
	= $(\text{frac}(32.4) * 55 + 0.5) \ll 8$ (シフトとは、CMPAHR の上位バイトに値を移動することです。)
	= $(0.4 * 55 + 0.5) \ll 8$
	= $(22 + 0.5) \ll 8$
	= $22.5 * 256$ (左に 8 桁シフトすることは、256 を掛けることと同じです。)
	= 5760 (1680h)
CMPAHR	= 1680h CMPAHR 値 = 1600h (下位 8 ビットはハードウェアによって無視されます。)

注

AUTOCONV ビット (HRCNFG.6) がセットされており、MEP_ScaleFactor が HRMSTEP レジスタに存在する場合、CMPAHR/CMPBHR レジスタ値 = $\text{frac}(\text{PWMDuty} * \text{PWMperiod} \ll 8)$ です。残りの変換計算はハードウェア内で自動的に実行され、MEP でスケールされた正しい信号エッジが ePWM チャネル出力に現れます。AUTOCONV がセットされていない場合、上記の計算をソフトウェアで実行する必要があります。

MEP 係数 (MEP_ScaleFactor) は、システム・クロックと DSP 動作条件によって変化します。テキサス・インスツルメンツは、MEP 係数最適化 (SFO) ソフトウェアの C 関数を提供しています。この関数は、各 HRPWM に内蔵された診断機能を使用して、特定の動作点に最適な係数を返します。

この係数は限られた範囲にわたってゆっくりと変化するため、C 関数の最適化は、バックグラウンド・ループで非常に低速で実行できます。

CPU の 32 ビット・データ機能がこれを 1 つの連結値 (つまり [CMPA : CMPAHR], [CMPB : CMPBHR] など) として書き込むことができるように、CMPA、CMPB、CMPAHR、CMPBHR レジスタはメモリ内で構成されています。

マッピング手順は、C とアセンブリの両方で実装されています ([セクション 1.15.1.8](#) を参照)。実際の実装では、32 ビット CPU アーキテクチャを利用しており、[セクション 1.15.1.5.2](#) に示す手順とは多少異なります。

サイクルごとにカウントするタイムクリティカルな制御ループの場合、アセンブリ・バージョンを推奨します。これは、Q15 デューティ値を入力として使い、1 つの [CMPA : CMPAHR] 値を書き込む、サイクル数が最適化された機能 (11 EPWMCLK サイクル) です。

1.15.1.5.3 デューティ・サイクル範囲の制約

高分解能モードでは、MEP は PWM 周期の全体では作動せず、以下のタイミングで動作可能になります。

- 周期が開始して 3 EPWMCLK サイクル後 (高分解能周期 (TBPRDHR) 制御が有効化されていない場合)。
- HRPCTL レジスタを使用して高分解能周期 (TBPRDHR) 制御が有効化されている場合、MEP は以下の期間に動作可能になります。
 - アップカウント・モードの場合：周期の冒頭の 3 EPWMCLK サイクル後から、周期の末尾の 3 EPWMCLK サイクル前まで。
 - アップ・ダウン・カウント・モードの場合：カウント・アップ時は、CTR = 0 の 3 サイクル後から、CTR = PRD の 3 サイクル前まで。カウント・ダウン時は、CTR = PRD の 3 サイクル後から、CTR = 0 の 3 サイクル前まで。
- DBREDHR または DBFEDHR を使用する場合、DBRED または DBFED (高分解能変位を伴うエッジに対応するレジスタ) は 7 以上である必要があります。

[図 1-85](#) ~ [図 1-88](#) に、デューティ・サイクル範囲の制約を示します。この制約により、MEP のデューティ・サイクルが制限されます。たとえば、0% のデューティ・サイクルまで、高精度のエッジ制御が利用できる訳ではありません。高分解能周期制御を無効化した場合、通常の PWM デューティ制御は 0% のデューティ・サイクルまで完全に機能します。ほとんどのアプリケーションでは、コントローラのレギュレーション点は通常、0% デューティ・サイクルの近くには設計されていないため、これは問題になりません。利用可能なデューティ・サイクル範囲の詳細については、[表 1-18](#) を参照してください。高分解能周期制御が有効化されている場合 (HRPCTL[HRPE] = 1)、禁止された範囲にデューティ・サイクルが入らないようにする必要があります。さもないと、ePWMxA 出力で未定義の動作が行われる可能性があります。

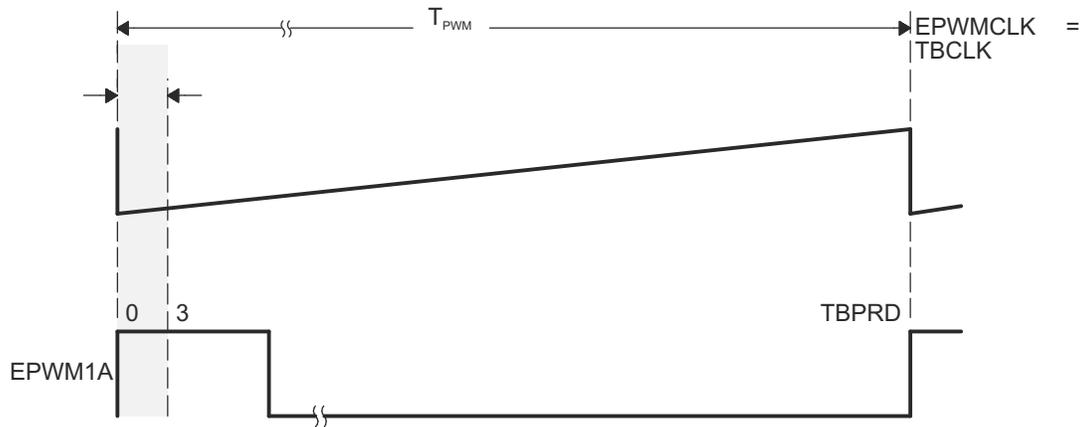


図 1-85. 低 % デューティ・サイクル範囲の制約の例 (HRPCTL[HRPE] = 0)

表 1-18. 3 EPWMCLK/TBCLK サイクルのデューティ・サイクル範囲の制約

PWM 周波数 ⁽¹⁾ (kHz)	3 サイクル 最小デューティ	3 サイクル 最大デューティ ⁽²⁾
200	0.6%	99.4%
400	1.2%	98.8%
600	1.8%	98.2%
800	2.4%	97.6%
1000	3%	97%
1200	3.6%	96.4%
1400	4.2%	95.8%
1600	4.8%	95.2%
1800	5.4%	94.6%
2000	6%	94%

(1) EPWMCLK = TBCLK = 100MHz

(2) この制約は、高分解能周期 (TBPRDHR) 制御が有効化されている場合のみ適用されます。

アプリケーションで、HRPWM が最小デューティ・サイクル制約を下回って動作する必要がある場合、高分解能周期が無効化された状態 (HRPCTL[HRPE] = 0) で、立ち上がりエッジ位置 (REP) を MEP で制御するカウントダウン・モードで動作するように HRPWM を設定できます。これを図 1-86 に示します。この構成では、最小デューティ・サイクル制約はもはや問題になりません。ただし、表 1-18 に示す値と同じパーセントの最大デューティ制約があります。

注意

アプリケーションで高分解能周期制御が有効化されている場合 (HRPCTL[HRPE] = 1)、禁止された範囲にデューティ・サイクルが入らないようにする必要があります。さもないと、ePWM 出力で未定義の動作が行われる可能性があります。

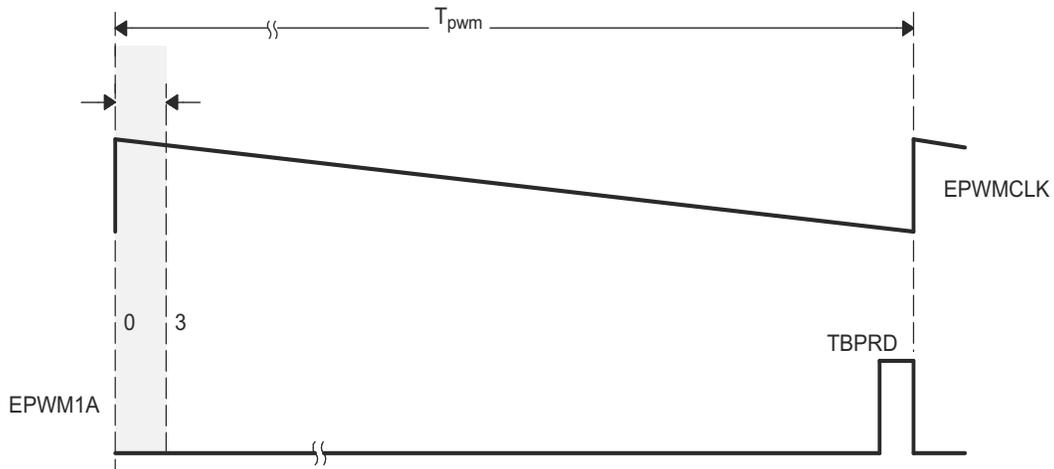


図 1-86. 高 % デューティ・サイクル範囲の制約の例 (HRPCTL[HRPE] = 0)

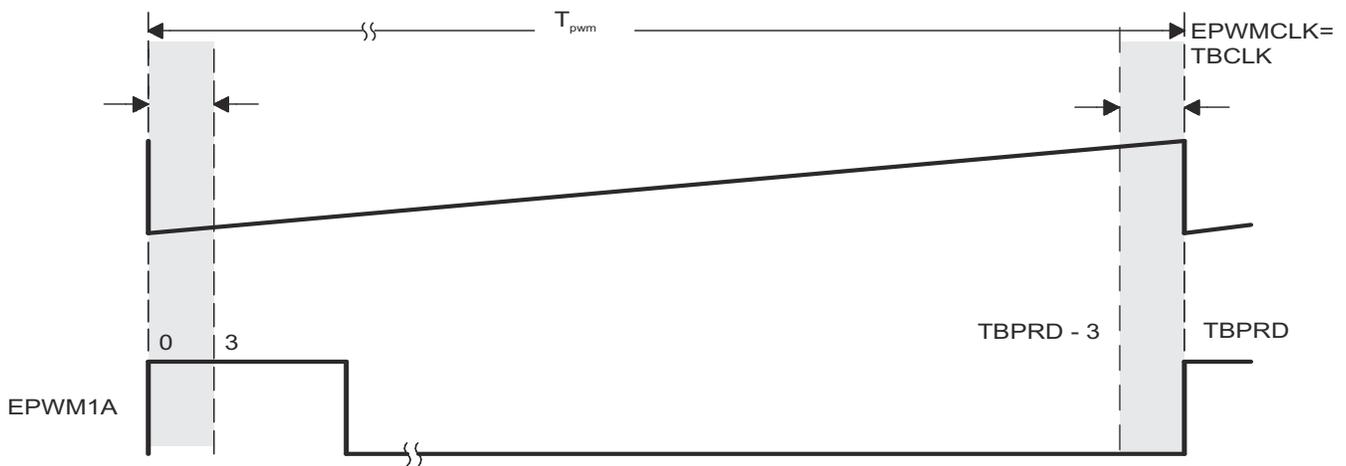


図 1-87. アップ・カウントのデューティ・サイクル範囲の制約の例 (HRPCTL[HRPE] = 1)

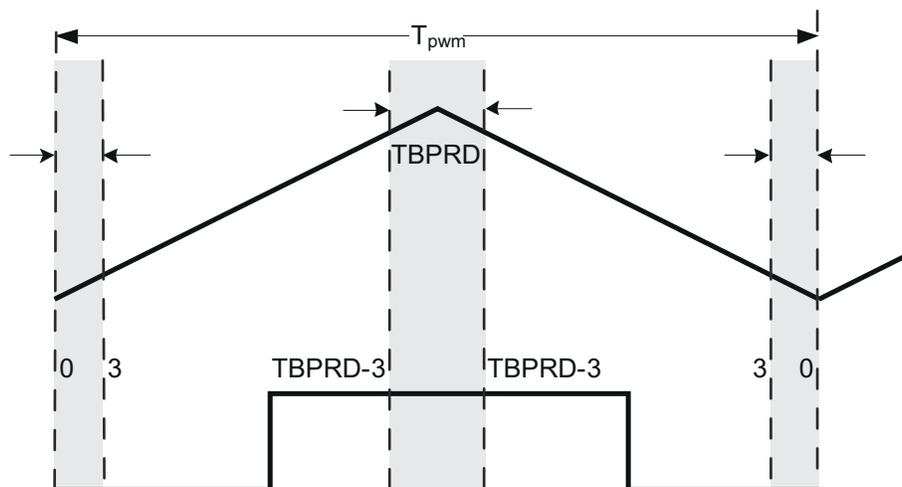


図 1-88. アップ・ダウン・カウントのデューティ・サイクル範囲の制約の例 (HRPCTL[HRPE] = 1)

1.15.1.5.4 高分解能周期

MEP ロジックを使った高分解能周期制御は、タイプ 1 以上の ePWM モジュールを備えたデバイスでサポートされています。

注

ePWMxA のみで高分解能周期制御が有効化されており、逆に、ePWMxB 出力では有効化されていない場合、非高分解能出力ではアップ・カウント・モードで ± 1 TBCLK のサイクル・ジッタ、アップ・ダウン・カウント・モードで ± 2 TBCLK のサイクル・ジッタが発生します。

セクション 1.15.1.5.2 のデューティ・サイクルを説明したスケーリング手順は、以下の高分解能周期にも適用されます。

この例の前提：

TBCLK	= 10ns (100MHz)
必要な PWM 周波数	= 175kHz (571.428 の周期)
180ps での粗ステップあたりの MEP ステップ数 (MEP_ScaleFactor)	= 55 (10ns/180ps)
TBPRDHR を 1~255 の範囲内に維持し、端数を丸めるための定数 (デフォルト値)	= 0.5 (Q8 形式で 0080h)

問題：

アップ・カウント・モードでは

- TBPRD = 571 の場合、PWM 周波数 = 174.82kHz (周期 = $(571 + 1) * T_{TBCLK}$) です。
- TBPRD = 570 の場合、PWM 周波数 = 175.13kHz (周期 = $(570 + 1) * T_{TBCLK}$) です。

アップ・ダウン・カウント・モードでは

- TBPRD = 286 の場合、PWM 周波数 = 174.82kHz (周期 = $(286 * 2) * T_{TBCLK}$) です。
- TBPRD = 285 の場合、PWM 周波数 = 175.44kHz (周期 = $(285 * 2) * T_{TBCLK}$) です。

ソリューション：

粗ステップあたり 55 MEP ステップ (各 180ps) :

ステップ 1 : TBPRD レジスタのパーセンテージ - 整数周期値変換

整数周期値	= $571 * T_{TBCLK}$ = $\text{int}(571.428) * T_{TBCLK}$ = $\text{int}(\text{PWMperiod}) * T_{TBCLK}$
アップ・カウント・モードでは TBPRD	= 570 (TBPRD = 周期値 - 1) = 023Ah
アップ・ダウン・カウント・モードでは TBPRD	= 285 (TBPRD = 周期値 / 2) = 011Dh

ステップ 2 : TBPRDHR レジスタの分数値変換

アップ・カウント・モードでは

 TBPRDHR レジスタ値 $= (\text{frac}(\text{PWMperiod}) * \text{MEP_ScaleFactor} + 0.5)$

自動変換が有効化されており、かつ HRMSTEP =

 MEP_ScaleFactor 値 (55) $= \text{frac}(\text{PWMperiod}) \ll 8$ (シフトとは、TBPRDHR の上位バイトに値を移動することです。)

 TBPRDHR レジスタ値 $= \text{frac}(571.428) \ll 8$
 $= 0.428 \times 256$
 $= 6D00h$

その後、ハードウェアによって TBPRDHR MEP 遅延が以下のようにスケーリングされるように、自動変換が自動的に計算を実行します。

 $= ((\text{TBPRDHR}(15 : 0) \gg 8) \times \text{HRMSTEP} + 80h) \ll 8$
 $= (006Dh \times 55 + 80h) \gg 8$
 $= (17EBh) \gg 8$

周期の MEP 遅延

 $= 0017h$ MEP ステップ

アップ・ダウン・カウント・モードでは

 TBPRDHR レジスタ値 $= (\text{frac}(\text{PWMperiod}) * \text{MEP_ScaleFactor} + 0.5)$

自動変換が有効化されており、かつ HRMSTEP =

 MEP_ScaleFactor 値 (55) $= \text{frac}(\text{PWMperiod} / 2) \ll 8$ (シフトとは、TBPRDHR の上位バイトに値を移動することです。)

 TBPRDHR レジスタ値 $= \text{frac}(285.714) \ll 8$
 $= 0.714 \times 256$
 $= B600h$

その後、ハードウェアによって TBPRDHR MEP 遅延が以下のようにスケーリングされるように、自動変換が自動的に計算を実行します。

 $= ((\text{TBPRDHR}(15 : 0) \gg 8) \times \text{HRMSTEP} + 80h) \ll 8$
 $= (00B6h \times 55 + 80h) \gg 8$
 $= (279Ah) \gg 8$

周期の MEP 遅延

 $= 0027h$ MEP ステップ

1.15.1.5.4.1 高分解能周期の設定

高分解能周期を使用するには、示された正しい順序で ePWMx モジュールを初期化する必要があります。

以下の手順では、EPWMxA での高分解能動作のため、シャドウ・レジスタと、対応する HRCNFG ビットとを CMPA と組み合わせて使用します。EPWMxB で高分解能動作を行うには、B チャネル・フィールドを使って適切な置き換えを行います。

1. ePWMx クロックを有効化します。
2. HRPWM クロックを有効化します。
3. TBCLKSYNC を無効化します。
4. ePWMx レジスタ (AQ、TBPRD、CC など) を設定します。
 - ePWMx は、アップ・カウント・モードまたはアップ・ダウン・カウント・モードにのみ設定できます。高分解能周期はダウン・カウント・モードには対応していません。
 - TBPRD および CC レジスタは、シャドウ・ロード用に設定する必要があります。
 - CMPCTL[LOADAMODE]
 - アップ・カウント・モードでは CMPCTL[LOADAMODE] = 1 (CTR = PRD 時にロード)
 - アップ・ダウン・カウント・モードでは CMPCTL[LOADAMODE] = 2 (CTR = 0 時または CTR = PRD 時にロード)
5. HRCNFG レジスタを以下のように設定します。
 - HRCNFG[HRLOAD] = 2 (CTR = 0 時と CTR = PRD 時のどちらかにロード)
 - HRCNFG[AUTOCONV] = 1 (自動変換を有効化)
 - HRCNFG[EDGMODE] = 3 (両方のエッジで MEP 制御)
6. 高分解能周期での TBPHS : TBPHSHR 同期の場合、HRPCTL[TBPSHRLOADE] = 1 かつ TBCTL[PHSEN] = 1 に設定します。アップ・ダウン・カウント・モードでは、TBPHSHR の内容に関係なく、これらのビットを 1 に設定する必要があります。
7. 高分解能周期制御を有効化 (HRPCTL[HRPE] = 1) します。
8. TBCLKSYNC を有効化します。
9. TBCTL[SWFSYNC] = 1
10. 自動変換が有効化されているため、HRMSTEP には正確な MEP 係数 (EPWMCLK の粗ステップごとの MEP ステップ数) が含まれている必要があります。MEP 係数は、[セクション 1.15.2](#) に記載された SFO() 関数を使って求めることができます。
11. 高分解能周期を制御するには、TBPRDHR(M) レジスタに書き込みます。

注

高分解能周期モードが有効化されている場合、EPWMxSYNC パルスは $\pm 1 \sim 2$ サイクルのジッタを PWM にもたらしめます (アップ・カウント・モードでは ± 1 サイクル、アップ・ダウン・モードでは ± 2 サイクル)。このため、EPWMxSYNCO のソースは CTR = 0 にも CTR = CMPB にも設定することはできません。さもなければ、同期パルスによって PWM サイクルごとにジッタが発生します。

EPWMxSYNCI が EPWMxSYNCO のソースである場合、ソフトウェア同期パルスは高分解能周期の初期化中に 1 回のみ発行できます。PWM 動作中にソフトウェア同期パルスが印加されると、同期パルスのタイミングで PWM 出力にジッタが現れます。

1.15.1.6 デッドバンド高分解能動作

注

アップ・カウント・モードでは、いずれかの高分解能モードが有効化されている場合、デッドバンド・モジュールは利用できません。

この例の前提：

システム・クロック	= 10ns (100MHz)
半周期モードでデッドバンドが有効、TBCLK = EPWMCLK	
必要な PWM 周波数	1.33MHz (1/750ns)
必要な PWM デューティ・サイクル	0.5 (50%)
必要なデッドバンド立ち上がりエッジ遅延	デューティの 5%
必要なデッドバンド立ち上がりエッジ遅延 (ns)	(0.05 * 375ns) = 18.75ns

注

HRPWM を使用する際のデューティ・サイクル制限と同様に、高分解能デッドバンドを使用するには、DBRED と DBFED の値を 3 より大きくする必要があります。

DBFED と DBRED の関数としてのデッドバンド遅延値：

半周期クロックが有効化されている場合、立ち下がりエッジ遅延と立ち上がりエッジ遅延を計算するための式は次のようになります。

$$FED = DBFED * TBCLK / 2$$

$$RED = DBRED * TBCLK / 2$$

DBRED と DBFED の計算値：

$$\text{必要なデッドバンド立ち上がりエッジ遅延 (ns)} = 18.75\text{ns}$$

$$DBRED = RED / (TBCLK / 2)$$

$$DBRED = 18.75\text{ns} / 5\text{ns}$$

$$\text{必要な DBRED} = 3.75\text{ns}$$

粗ステップあたり 55 MEP ステップ (各 180ps)：

ステップ 1：DBREDM レジスタの整数デッドバンド値変換

DBRED の整数値	= int (RED / (TBCLK / 2))
	= int (3.75)
DBRED	= 3

ステップ 2 : デッドバンド高分解能レジスタ DBREDHR の分数値変換

DBREDHR レジスタ値	$= (\text{frac}(\text{必要な DBRED}) * \text{MEP_ScaleFactor} + 0.5) \ll 8$ (シフトとは、DBREDHR の上位バイトに値を移動させることです。)
	$= (\text{frac}(3.75) * 55 + 0.5) \ll 8$
	$= (0.75 * 55 + 0.5) \ll 8$
	$= (41.75) * 256$ (左に 8 桁シフトすることは、256 を掛けることと同じです。)
DBREDHR 値	= 29C0h MEP ステップ 上記の DBREDHR の計算値の下位 9 ビットをハードウェアは無視します。

注

AUTOCONV ビット (HRCNFG.6) がセットされており、MEP_ScaleFactor が HRMSTEP レジスタに存在する場合、DBREDHR : DBRED = $\text{frac}(\text{必要な DB 値}) \ll 8$ です。残りの変換計算はハードウェア内で自動的に実行され、MEP でスケールされた正しい信号エッジが ePWM チャネル出力に現れます。AUTOCONV がセットされていない場合、上記の計算をソフトウェアで実行する必要があります。

1.15.1.7 係数最適化ソフトウェア (SFO)

マイクロ・エッジ・ポジショナ (MEP) ロジックは、255 の離散的な時間ステップのいずれかにエッジを配置できます。すでに説明したように、これらのステップのサイズは 150ps のオーダーです (お使いのデバイスの MEP ステップ・サイズの標準値については、そのデバイスのデータ・マニュアルを参照してください)。MEP ステップ・サイズは、最も厳しいプロセス・パラメータ、動作温度および電圧に基づいて変化します。MEP ステップ・サイズは、電圧の低下と温度の上昇に伴って増加し、電圧の上昇と温度の低下に伴って減少します。HRPWM 機能を使用するアプリケーションでは、テキサス・インスツルメンツが提供する MEP 係数最適化 (SFO) ソフトウェア関数を使用できます。SFO 関数は、HRPWM の動作中に、EPWMCLK 周期あたりの MEP ステップ数を動的に決定するのに役立ちます。

MEP 機能を効果的に利用するには、MEP 係数の正しい値をソフトウェアが認識する必要があります。これを実現するため、HRPWM モジュールは、任意の動作条件に最適な MEP 係数値を決定するために使用できる自己チェックおよび診断機能を内蔵しています。テキサス・インスツルメンツは、このハードウェアを利用して最適な MEP 係数を決定する 1 つの SFO 関数を含む C 呼び出し可能ライブラリを提供しています。そのような理由で、MEP 制御および診断レジスタは、テキサス・インスツルメンツ用に予約済みです。

SFO ライブラリの SFO_TI_Build_V8.lib ソフトウェアの詳細な説明については、[セクション 1.15.2](#) を参照してください。

1.15.1.8 最適化されたアセンブリ・コードを使用した HRPWM の例

HRPWM 機能の使い方を理解するための最善の方法は、次の 2 つの実例を参照することです。

1. アクティブ High 極性の非対称 PWM (カウント・アップ) を使用したシンプルな降圧コンバータ。
2. シンプルな R+C 再構成フィルタを使用した DAC 機能。

以下のすべての例には、C で記述された初期化および構成コードが用意されています。これらを理解しやすくするため、以下に示す #define 文が使われています。

Example 1-2 では、150ps の MEP ステップ・サイズを前提としており、SFO ライブラリは使用していません。

Example 1-2. HRPWM ヘッダ・ファイルの #Define 文

```
// HRPWM (High Resolution PWM) //
=====
// HRCNFG
#define HR_Disable 0x0
#define HR_REP 0x1           // Rising Edge position
#define HR_FEP 0x2           // Falling Edge position
#define HR_BEP 0x3           // Both Edge position #define HR_CMP 0x0 // CMPAHR controlled
#define HR_PHS 0x1           // TBPHSHR controlled #define HR_CTR_ZERO 0x0 // CTR = Zero event
#define HR_CTR_PRD 0x1       // CTR = Period event
#define HR_CTR_ZERO_PRD 0x2 // CTR = ZERO or Period event
#define HR_NORM_B 0x0        // Normal ePWMxB output
#define HR_INVERT_B 0x1      // ePWMxB is inverted ePWMxA output
```

1.15.1.8.1 シンプルな降圧コンバータの実装

この例での、PWM の要件を以下に示します。

- PWM 周波数 = 1MHz (つまり、TBPRD = 100)
- PWM モード = 非対称、アップ・カウント
- 分解能 = 12.7 ビット (MEP ステップ・サイズ = 150ps)

図 1-89 と図 1-90 に、必要な PWM 波形を示します。すでに説明したように、ePWM1 モジュールの設定は通常の場合とほぼ同じですが、適切な MEP オプションを有効化 / 選択する必要がある点が異なります。

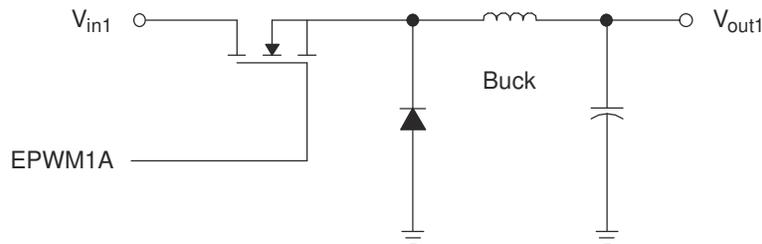


図 1-89. 1 つの PWM を使用したシンプルな降圧制御コンバータ

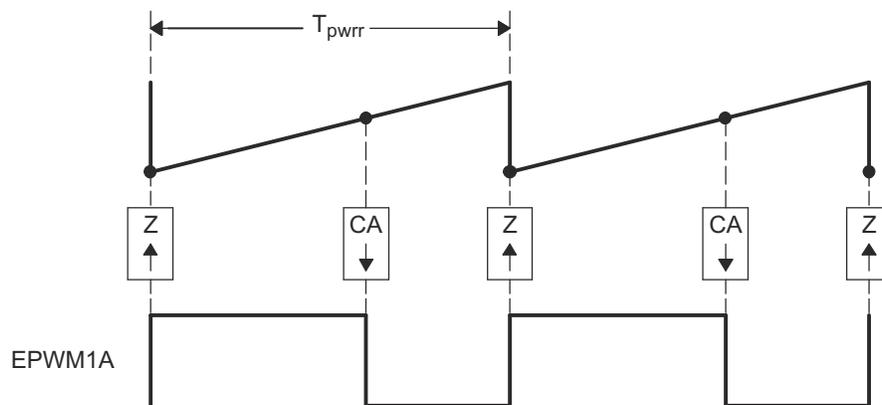


図 1-90. シンプルな降圧制御コンバータのために生成された PWM 波形

ここに示すサンプル・コードは、2 つの主要な部分で構成されています。

- 初期化コード (1 回実行)
- ランタイム・コード (通常は ISR 内で実行)

Example 1-3 に、初期化コードを示します。最初の部分は、従来型 PWM 用に構成されています。2 番目の部分では、HRPWM リソースを設定しています。

この例では、150ps の MEP ステップ・サイズを前提としており、SFO ライブラリは使用していません。

Example 1-4 に、HRPWM 降圧コンバータのランタイム・コードのアセンブリ例を示します。

Example 1-3. HRPWM 降圧コンバータの初期化コード

```

void HrBuckDrvCnf(void)
{
// Config for conventional PWM first
EPwm1Regs.TBCTL.bit.PRDL = TB_IMMEDIATE;           // set Immediate load
EPwm1Regs.TBPRD = 100;                             // Period set for 1000 khz PWM
hrbuck_period = 200;                                // used for Q15 to Q0 scaling
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;           // EPWM1 is the Master

EPwm1Regs.EPWSYNCOUTEN.all = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
// Note: ChB is initialized here only for comparison purposes, it is not required

EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;      // optional
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;       // optional
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;                // optional
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;             // optional
// Now configure the HRPWM resources
EALLOW;                                           // Note these registers are protected
                                                    // and act only on ChA
EPwm1Regs.HRCNFG.all = 0x0;                       // clear all bits first
EPwm1Regs.HRCNFG.bit.EDGMODE = HR_FEP;           // Control Falling Edge Position
EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP;          // CMPAHR controls the MEP
EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO;      // Shadow load on CTR=Zero
EDIS;
MEP_ScaleFactor = 66*256;                         // Start with typical Scale Factor
                                                    // value for 100 MHz
                                                    // Note: Use SFO functions to update
                                                    // MEP_ScaleFactor dynamically
}
    
```

Example 1-4. HRPWM 降圧コンバータのランタイム・コード

```

EPWM1_BASE .set 0x6800
CMPAHR1 .set EPWM1_BASE+0x8
;=====
HRBUCK_DRV; (can execute within an ISR or loop)
;=====
    MOVW DP, #_HRBUCK_In
    MOVL XAR2,@_HRBUCK_In      ; Pointer to Input Q15 Duty (XAR2)
    MOVL XAR3,#CMPAHR1        ; Pointer to HRPWM CMPA reg (XAR3)

; Output for EPWM1A (HRPWM)
    MOV T,*XAR2 ; T <= Duty
    MPYU ACC,T,@_hrbuck_period ; Q15 to Q0 scaling based on Period
    MOV T,@_MEP_ScaleFactor   ; MEP scale factor (from optimizer s/w)
    MPYU P,T,@AL              ; P <= T * AL, Optimizer scaling
    MOVH @AL,P                ; AL <= P, move result back to ACC
    ADD ACC, #0x080           ; MEP range and rounding adjustment
    MOVL *XAR3,ACC           ; CMPA: CMPAHR(31:8) <= ACC

; Output for EPWM1B (Regular Res) Optional - for comparison purpose only
    MOV *+XAR3[2],AH        ; Store ACCH to regular CMPB
    
```

1.15.1.8.2 R+C 再構成フィルタを使用した DAC 機能の実装

この例での、PWM の要件を以下に示します。

- PWM 周波数 = 400kHz (つまり、TBPRD = 250)
- PWM モード = 非対称、アップ・カウント
- 分解能 = 14 ビット (MEP ステップ・サイズ = 150ps)

図 1-91 と図 1-92 に、DAC の機能と必要な PWM 波形を示します。すでに説明したように、ePWM1 モジュールの設定は通常の場合とほぼ同じですが、適切な MEP オプションを有効化 / 選択する必要がある点が異なります。

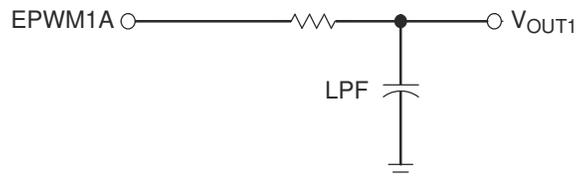


図 1-91. PWM を使った DAC のためのシンプルな再構成フィルタ

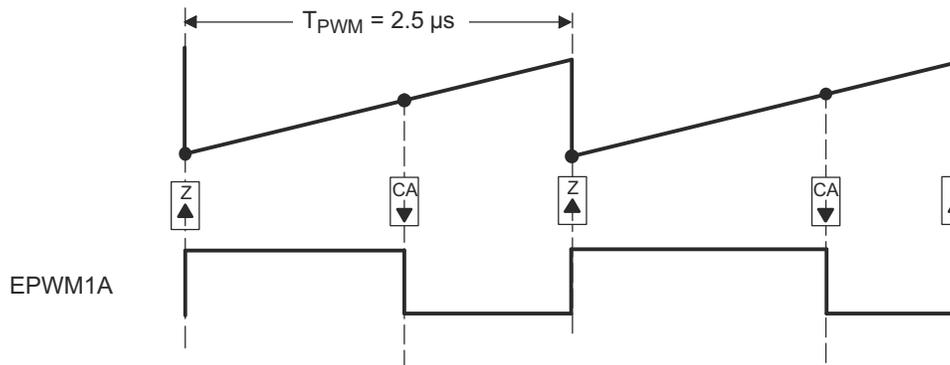


図 1-92. PWM DAC 機能用に生成された PWM 波形

ここに示すサンプル・コードは、2 つの主要な部分で構成されています。

- 初期化コード (1 回実行)
- ランタイム・コード (通常は ISR 内で実行)

この例では、代表的な MEP_SP を前提としており、SFO ライブラリは使用していません。

Example 1-5 に、初期化コードを示します。最初の部分は、従来型 PWM 用に構成されています。2 番目の部分では、HRPWM リソースを設定しています。

Example 1-6 に、高速 ISR ループ内で実行できるランタイム・コードのアセンブリ例を示します。

Example 1-5. PWM DAC 機能の初期化コード

```

void HrPwmDacDrvCnf(void)
{
// Config for conventional PWM first
EPwm1Regs.TBCTL.bit.PRDL = TB_IMMEDIATE;           // Set Immediate load
EPwm1Regs.TBPRD = 250;                             // Period set for 400 kHz PWM
hrDAC_period = 250;                                 // Used for Q15 to Q0 scaling
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE;           // EPWM1 is the Master

EPwm1Regs.EPWSYNCOUTEN.all = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1;
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
// Note: ChB is initialized here only for comparison purposes, it is not required

EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;    // optional
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;      // optional

EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;               // optional
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;            // optional
// Now configure the HRPWM resources
EALLOW;                                          // Note these registers are protected
                                                // and act only on ChA.
EPwm1Regs.HRCNFG.all = 0x0;                     // Clear all bits first
EPwm1Regs.HRCNFG.bit.EDGMODE = HR_FEP;          // Control falling edge position
EPwm1Regs.HRCNFG.bit.CTLMODE = HR_CMP;          // CMPAHR controls the MEP.
EPwm1Regs.HRCNFG.bit.HRLOAD = HR_CTR_ZERO;      // Shadow load on CTR=Zero.
EDIS;
MEP_ScaleFactor = 66*256;                       // Start with typical Scale Factor
                                                // value for 100 MHz.
                                                // Use SFO functions to update MEP_ScaleFactor
                                                // dynamically.
}
    
```

Example 1-6. PWM DAC 機能のランタイム・コード

```

EPWM1_BASE .set 0x6800
CMPAHR1 .set EPWM1_BASE+0x8
;=====
HRPWM_DAC_DRV; (can execute within an ISR or loop)
;=====
    MOVW DP, #_HRDAC_In
    MOVL XAR2,@_HRDAC_In           ; Pointer to input Q15 duty (XAR2)
    MOVL XAR3,#CMPAHR1            ; Pointer to HRPWM CMPA reg (XAR3)

; Output for EPWM1A (HRPWM)
    MOV T,*XAR2                   ; T <= duty
    MPY ACC,T,@_hrDAC_period      ; Q15 to Q0 scaling based on period
    ADD ACC,@_hrDAC_period<<15   ; Offset for bipolar operation
    MOV T,@_MEP_ScaleFactor       ; MEP scale factor (from optimizer s/w)
    MPYU P,T,@AL                  ; P <= T * AL, optimizer scaling
    MOVH @AL,P                    ; AL <= P, move result back to ACC
    ADD ACC,#0x080                ; MEP range and rounding adjustment
    MOVL *XAR3,ACC                ; CMPA: CMPAHR(31:8) <= ACC

; Output for EPWM1B (Regular Res) Optional - for comparison purpose only
    MOV *+XAR3[2],AH              ; Store ACCH to regular CMPB
    
```

1.15.2 SFO ライブラリ・ソフトウェア - SFO_TI_Build_V8.lib

表 1-19 に、SFO_TI_Build_V8.lib ライブラリの各種機能を示します。

表 1-19. SFO ライブラリの特長

	SFO_TI_Build_V8.lib	単位
チェックの完了確認	あり	関数の戻り値
割り込みを使わずに反復的に呼び出された場合に、SFO() が MEP_ScaleFactor を更新するのに必要なサイクル数の標準値	130,000	EPWMCLK サイクル

1.15.2.1 係数最適化関数 - int SFO()

SFO 診断を実行し、その時々に適したデバイスの MEP 係数 (粗 EPWMCLK ステップあたりの MEP ステップ数) を決定するため、このルーチンはマイクロ・エッジ・ポジショナ (MEP) 較正モジュールを駆動します。

EPWMCLK = TBCLK = 100MHz であり、MEP ステップ・サイズが 150ps であると仮定すると、100MHz での係数値 (標準値) は TBCLK ユニットの (10ns) あたり 66 MEP ステップです。

この関数は、次のように MEP 係数値を返します。

MEP_ScaleFactor = EPWMCLK あたりの MEP ステップ数

この関数を使う際の制約：

- SFO() は、最小 EPWMCLK = TBCLK = 50MHz で使用できます。MEP 診断ロジックは、TBCLK ではなく EPWMCLK を使用するため、EPWMCLK の制限は重要な制約です。50MHz 未満では、デバイス・プロセス変動に伴って、低温かつ高コア電圧の条件下で、255 の MEP ステップでは EPWMCLK サイクル全体を網羅できない程度まで MEP ステップ・サイズが小さくなることがあります。
- MEP 較正モジュールで SFO 診断を実行するため、いつでも SFO() を呼び出すことができます。

使い方：

- SFO() は、ePWM チャンネルが HRPWM モードで動作している間、バックグラウンドでいつでも呼び出すことができます。この関数は MEP 較正モジュールの診断ロジック (ePWM チャンネルから独立して動作します) を利用しているため、得られた係数の計算結果は、HRPWM モードで動作するすべての ePWM チャンネルに適用できます。
- このルーチンは、較正が完了して新しい係数の計算が完了している場合は 1 を返し、較正がまだ実行されている場合は 0 を返します。エラーが発生し、粗 EPWMCLK サイクルあたり 255 微ステップの最大値を MEP_ScaleFactor が超えた場合、このルーチンは 2 を返します。この場合、HRMSTEP レジスタは最後の MEP 係数値を自動変換で 256 未満に維持します。
- 高分解能周期制御が使われていない場合、HRPWM で動作するすべての ePWM モジュールには、3 EPWMCLK サイクルという最小デューティ・サイクルが必要です。高分解能周期制御が有効化されている場合、追加の制限があります ([セクション 1.15.1.5.3](#) を参照)。
- SFO() 関数は、係数の計算結果を使って HRMSTEP レジスタも更新します。HRCNFG[AUTOCONV] ビットがセットされている場合、バックグラウンドで SFO() を実行している間、アプリケーション・ソフトウェアは $CMPAHR = \text{fraction}(\text{PWMduty} * \text{PWMperiod}) \ll 8$ または $CMPBHR = \text{fraction}(\text{PWMduty} * \text{PWMperiod}) \ll 8$ または $TBPRDHR = \text{fraction}(\text{PWMperiod})$ の設定のみを行います。次に、MEP 較正モジュールは、分数デューティ・サイクルまたは周期で表された適切な MEP ステップ数を自動的に計算し、それに応じて高分解能 ePWM 信号エッジを動かすため、HRMSTEP と CMPAHR/CMPBHR/TBPRDHR レジスタの値を使います。
- HRCNFG[AUTOCONV] ビットがクリアされている場合、HRMSTEP レジスタは無視されます。アプリケーション・ソフトウェアは、以下のように、必要な計算を個別に実行する必要があります。
 - $CMPAHR = (\text{fraction}(\text{PWMduty} * \text{PWMperiod}) * \text{MEP 係数}) \ll 8 + 0x080$ 。
 - TBPHSHR、CMPBHR、DBREDHR、DBFEDHR についても同様です。TBPRDHR を使う場合、自動変換を有効化する必要があります。

以下のコード・スニペットは、Driverlib 機能を使用した HRPWM DUTY の使い方を示しています。

```
float32_t dutyFine = 85.62;
float32_t count = (dutyFine * (float32_t)(EPWM_TIMER_TBPRD << 8))/100;
uint32_t compCount = (count);
HRPWM_setCounterCompareValue(EPWM1_BASE, HRPWM_COUNTER_COMPARE_A, compCount);
HRPWM_setCounterCompareValue(EPWM1_BASE, HRPWM_COUNTER_COMPARE_B, compCount);
```

このルーチンは、ごくわずかな CPU サイクル数しか使わない低速ループのバックグラウンド・タスクとして実行できます。SFO 関数を実行する必要がある反復頻度は、アプリケーションの動作環境によって異なります。すべてのデジタル CMOS デバイスと同様、温度と電源電圧の変動は MEP の動作に影響します。ただし、ほとんどのアプリケーションでは、これらのパラメータはゆっくりと変化します。そのため、大抵、5~10 秒に 1 回 SFO 関数を実行するだけで十分です。より速い変化が予想される場合、アプリケーションに対応して、より頻繁に SFO 関数を実行することもできます。SFO 関数の反復頻度には上限はありません。したがって、バックグラウンド・ループが許す限りの頻度で SFO 関数を実行できます。

HRPWM 機能を使用している間、PWM 周期の最初の 3 EPWMCLK サイクル (と TBPRDHR を使用している場合、PWM 周期の最後の 3 EPWMCLK サイクル) の間、HRPWM ロジックは動作を停止します。この構成でアプリケーションを実行している間、高分解能周期制御が無効化されており (HRPCTL[HRPE=0])、CMPA/CMPB レジスタの値が 3 サイクル未満である場合、CMPAHR/CMPBHR レジスタはゼロにクリアされるはずですが、高分解能周期制御が有効化されている場合 (HRPCTL[HRPE=1])、CMPA レジスタの値は 3 未満にも (TBPRD - 3) より大きい値にもなりません。これにより、PWM 信号の予期しない遷移を防止できます。

1.15.2.2 ソフトウェアの使い方

ソフトウェア・ライブラリ関数 SFO() は、HRPWM をサポートする ePWM モジュールの MEP 係数を計算します。この係数は 1~255 の範囲の整数であり、システム・クロック周期として使用可能なマイクロ・ステップ・エッジ位置の数を表します。この係数の値は、MEP_ScaleFactor と呼ぶ整数変数で返されます。表 1-20 に、例を示します。

表 1-20. 係数値

ソフトウェア関数呼び出し	機能説明	更新された変数
SFO()	HRMSTEP レジスタの MEP 係数を返します。	MEP_ScaleFactor と HRMSTEP レジスタ。

ePWM の HRPWM 機能を使用するには、ここで説明するように、SFO 機能を使用することを推奨します。

ステップ 1. 「インクルード」ファイルの追加

SFO_V8.h ファイルは、以下のようにインクルードする必要があります。SFO ライブラリ関数を使用する場合、このインクルード・ファイルは必須です。SFO() を機能させるには、適切な (デバイス名)_Device.h および (デバイス名)_Epwm_defines.h をプロジェクトに含める必要があります。カスタマイズされたヘッダ・ファイルがエンド・アプリケーションで使われる場合、これらのインクルード・ファイルは任意です。

Example 1-7. 「インクルード」ファイルの追加方法の例

```
#include "F28x7x_Device.h" // F28x7x Headerfile
#include "F28x7x_EPwm_defines.h" // init defines
#include "SFO_V8.h" // SFO lib functions (needed for HRPWM)
```

ステップ 2.要素の宣言

以下に示すように、係数の値の整数変数を宣言します。

Example 1-8. エレメントの宣言

```
int MEP_ScaleFactor = 0;    //scale factor value
volatile struct EPWM_REGS *ePWM[] = {0, &EPwm1Regs, &EPwm2Regs, &EPwm3Regs,
&EPwm4Regs};
```

ステップ 3.MEP_ScaleFactor の初期化

SFO() 関数には、MEP_ScaleFactor の係数の初期値は不要です。アプリケーション・コードで MEP_ScaleFactor 変数を使用する前に、SFO() を呼び出すことで、MEP 較正モジュールを駆動して MEP_ScaleFactor 値を計算できます。

MEP_ScaleFactor を使用する前のワンタイム初期化コードの一部として、以下をインクルードします。

Example 1-9. 係数値による初期化

```
MEP_ScaleFactor initialized using function SFO ()
while (SFO() == 0) {} // MEP_ScaleFactor calculated by MEP Cal Module
```

ステップ 4.アプリケーション・コード

アプリケーションが実行している間、デバイスの温度と電源電圧の両方が変動することが予想される場合があります。各 ePWM モジュールのために適切な係数が確実に使われるように、低速バックグラウンド・ループの一部として SFO 関数を定期的に再実行できます。その例を以下に示します。

注

テキサス・インスツルメンツの Web サイトで提供しているデバイス固有の C/C++ ヘッダ・ファイルの HRPWM_SFO の例とペリフェラルの例を参照してください。

Example 1-10. SFO 関数呼び出し

```
main ()
{
    int status;
    // User code
    // ePWM1, 2, 3, 4 are running in HRPWM mode
    // The status variable returns 1 once a new MEP_ScaleFactor has been
    // calculated by the MEP Calibration Module running SFO
    // diagnostics.
    status = SFO();
    if(status==2) {ESTOP0;} // The function returns a 2 if MEP_ScaleFactor is greater
    // than the maximum 255 allowed (error condition)
}
```

1.16 ソフトウェア

1.16.1 EPWM の例

注：これらの例は、以下の [C2000Ware](#) インストールの場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/epwm

これらの例へのクラウド・アクセスは、dev.ti.com [C2000Ware の例](#)のリンクから可能です。

1.16.1.1 ePWM トリップ・ゾーン

ファイル：epwm_ex1_trip_zone.c

この例では、ePWM1 と ePWM2 を次のように設定します。

- ePWM1 ではワンショットのトリップ・ソースとして TZ1 を使う。
- ePWM2 ではサイクル・バイ・サイクル・トリップ・ソースとして TZ1 を使う。

最初は TZ1 を High に接続します。テスト中、オシロスコープ上で ePWM1 または ePWM2 出力を監視します。その効果を確認するには、TZ1 を Low にプルします。

外部接続

- ePWM1A は GPIO0 に接続
- ePWM2A は GPIO2 に接続
- TZ1 は GPIO12 に接続

この例では、入カクロスバーも使用しています。GPIO12 (外部トリガ) は入カクロスバーに配線され、そこから TZ1 に配線されています。

TZ イベントは、ePWM1A がワンショット・トリップを受け、ePWM2A がサイクル・バイ・サイクル・トリップを受けるように定義されています。

1.16.1.2 ePWM アップ・ダウン・カウント・アクション・フィルタ

ファイル：epwm_ex2_updown_aq.c

この例では、ePWMxA と ePWMxB で独立した変調を行う波形を生成するように、ePWM1、ePWM2、ePWM3 を設定します。

比較値 CMPA および CMPB は ePWM の ISR 内で変更されます。

この例では、TB カウンタはアップ / ダウン・カウント・モードになっています。

ePWM1A/B(GPIO0 & GPIO1)、ePWM2A/B(GPIO2 & GPIO3)、ePWM3A/B(GPIO4 & GPIO5) の波形をオシロスコープで観測します。

1.16.1.3 ePWM 同期

ファイル：epwm_ex3_synchronization.c

この例では、ePWM1、ePWM2、ePWM3、ePWM4 を次のように設定します。

- ePWM1 では同期ソースとして位相シフトを使わない。
- ePWM2 では 300 TBCLK の位相シフトを行う。
- ePWM3 では 600 TBCLK の位相シフトを行う。
- ePWM4 では 900 TBCLK の位相シフトを行う。

外部接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO2 EPWM2A
- GPIO3 EPWM2B
- GPIO4 EPWM3A
- GPIO5 EPWM3B

- GPIO6 EPWM4A
- GPIO7 EPWM4B

観測 変数

- なし。

1.16.1.4 ePWM デジタル比較

ファイル : epwm_ex4_digital_compare.c

この例では、ePWM1 を次のように設定します。

- DCAEVT1 によって ePWM1 の ePWM 出力を Low に強制
- GPIO24 は、入カクロスバーの INPUT1 への入力として使用
- (入カクロスバーの) INPUT1 は、DCAEVT1 のソースとして使用
- GPIO24 のプルアップ抵抗は有効 (トリップをテストするには、このピンを GND にプルします。)

外部 接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO24 TZ1 (ePWM をトリップさせるには、このピンを Low にプルします。)

観測 変数

- なし。

1.16.1.5 ePWM デジタル比較イベント・フィルタ・ブランキング・ウィンドウ

ファイル : epwm_ex5_digital_compare_event_filter.c

この例では、ePWM1 を次のように設定します。

- DCAEVT1 によって ePWM1 の ePWM 出力を Low に強制
- GPIO24 は、入カクロスバーの INPUT1 への入力として使用
- (入カクロスバーの) INPUT1 は、DCAEVT1 のソースとして使用
- GPIO24 のプルアップ抵抗は有効 (トリップをテストするには、このピンを GND にプルします。)
- DCBEVT1 によって ePWM1 の ePWM 出力を Low に強制
- GPIO24 は、入カクロスバーの INPUT1 への入力として使用
- (入カクロスバーの) INPUT1 は、DCAEVT1 のソースとして使用
- GPIO24 のプルアップ抵抗は有効 (トリップをテストするには、このピンを GND にプルします。)
- DCBEVT1 は、DCBEVT1 をフィルタ処理した信号を使用
- DCFILT 信号は、DC ブランキング・ウィンドウ期間中 DCBEVT1 を無視するため、ブランキング・ウィンドウを使用

外部 接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO24 TRIPIN1 (ePWM をトリップさせるには、このピンを Low にプルします。)

観測 変数

- なし。

1.16.1.6 ePWM バレー・スイッチング

ファイル : epwm_ex6_valley_switching.c

この例では、ePWM1 を次のように設定します。

- DCAEVT1 によって ePWM1 の ePWM 出力を Low に強制
- GPIO24 は、入カクロスバーの INPUT1 への入力として使用
- (入カクロスバーの) INPUT1 を DCAEVT1 のソースとして使用
- GPIO24 は出力に設定され、メイン・ループでトグルされて PWM をトリップ
- DCBEVT1 によって ePWM1 の ePWM 出力を Low に強制

- GPIO24 は、入カククロスバーの INPUT1 への入力として使用
- (入カククロスバーの) INPUT1 を DCAEVT1 のソースとして使用
- GPIO24 は出力に設定され、メイン・ループでトグルされて PWM をトリップ
- DCBEVT1 は、DCBEVT1 をフィルタ処理した信号を使用
- DCFILT 信号は、
- ソフトウェアで定義された DELAY 値だけ DCFILT 信号を遅延させるため、バレー・スイッチング・モジュールを使用

外部 接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO24 TRIPIN1 (出カピン、ソフトウェアでトグル)

観測 変数

- なし。

1.16.1.7 ePWM デジタル比較エッジ・フィルタ

ファイル : epwm_ex7_edge_filter.c

この例では、ePWM1 を次のように設定します。

- CBC ソースとしての DCBEVT2 によって ePWM1 の ePWM 出力を Low に強制
- GPIO24 は、入カククロスバーの INPUT1 への入力として使用
- (入カククロスバーの) INPUT1 は、DCBEVT2 のソースとして使用
- GPIO24 は出力に設定され、メイン・ループでトグルされて PWM をトリップ
- DCBEVT2 は DCFILT のソース
- DCFILT は DCBEVT2 のエッジをカウントし、DCBEVT2 の 4 番目のエッジで ePWM をトリップするための信号を生成

外部 接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO24 TRIPIN1 (出カピン、ソフトウェアでトグル)

観測 変数

- なし。

1.16.1.8 ePWM デッドバンド

ファイル : epwm_ex8_deadband.c

この例では、ePWM1 ~ ePWM6 を次のように設定します。

- ePWM1、デッドバンド無効 (基準)
- ePWM2、デッドバンド・アクティブ High
- ePWM3、デッドバンド・アクティブ Low
- ePWM4、デッドバンド・アクティブ High、相補
- ePWM5、デッドバンド・アクティブ Low、相補
- ePWM6、デッドバンド出カスワップ (スイッチ A および B 出力)

外部 接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO2 EPWM2A
- GPIO3 EPWM2B
- GPIO4 EPWM3A
- GPIO5 EPWM3B
- GPIO6 EPWM4A
- GPIO7 EPWM4B

- GPIO8 EPWM5A
- GPIO9 EPWM5B
- GPIO10 EPWM6A
- GPIO11 EPWM6B

観測 変数

- なし。

1.16.1.9 ePWM チョップパ

ファイル : epwm_ex10_chopper.c

この例では、ePWM1、ePWM2、ePWM3、ePWM4 を次のように設定します。

- ePWM1、チョップパ無効 (基準)
- ePWM2、チョップパ有効 (1/8 デューティ・ サイクル)
- ePWM3、チョップパ有効 (6/8 デューティ・ サイクル)
- ePWM4、チョップパ有効 (1/2 デューティ・ サイクル)、ワンショット・ パルス有効

外部 接続

- GPIO0 EPWM1A
- GPIO1 EPWM1B
- GPIO2 EPWM2A
- GPIO3 EPWM2B
- GPIO4 EPWM3A
- GPIO5 EPWM3B
- GPIO6 EPWM4A
- GPIO7 EPWM4B

観測 変数

- なし。

1.16.1.10 EPWM 設定信号

ファイル : epwm_ex11_configure_signal.c

この例では、目的の周波数とデューティの信号が生成されるように、ePWM1、ePWM2、ePWM3 を設定します。また、設定されたジュール間の位相も設定します。

デューティ 0.5、10kHz の信号を ePWMxA と ePWMxB (ePWMxB は反転) に設定します。また、ePWM1 信号と ePWM3 信号の間に 120 度の位相を設定します。

テスト中、ePWM1、ePWM2、ePWM3 の出力をオシロスコープで監視します。

- ePWM1A は GPIO0 に接続
- ePWM1B は GPIO1 に接続
- ePWM2A は GPIO2 に接続
- ePWM2B は GPIO3 に接続
- ePWM3A は GPIO4 に接続
- ePWM3B は GPIO5 に接続

1.16.1.11 ワンショット・ モードの実現

ファイル : epwm_ex12_monoshot_mode.c

この例では、外部トリガに基づいてワンショット PWM 出力を生成 (つまり、外部トリガを受信した時点で 1 つのパルス出力を生成) する方法を示します。次のパルスは、次のトリガが受信されたときにのみ生成されます。この例では、目的の出力を実現するため、外部同期機能および T1 アクション・ フィルタ・ イベント機能を利用しています。

ePWM1 は、ワンショット出力を生成するために使われ、ePWM2 は、そのための外部トリガを生成するために使われます。入力クロスバーを使用して ePWM2A がトリガとして自動的に入力されるため、外部接続は不要です。

ePWM1 は、外部トリガを受信した際、 $0.5\mu\text{s}$ の 1 つのパルスを生成するように設定されています。これは、位相同期機能を有効化し、EPWMxSYNCl を EXTSYNClN1 として設定することで実現されます。また、この EPWMxSYNCl は、出力を High に設定するためにアクション・フィルタの T1 イベントとしても設定されます。一方、出力を Low に設定するために CTR = PRD アクションが使われます。

ePWM2 は、(立ち上がりエッジ・トリガを模擬するため) 入力クロスバーを使用して EXTSYNClN1 に接続された 1% デューティの 100kHz 信号を生成するように設定されています。

GPIO0 (EPWM1A : ワンショット出力) と GPIO2 (EPWM2 : 外部トリガ) をオシロスコープで観察します。

注：以下の例では、ePWM タイマが依然としてワンショット・モードではなく連続モードで動作しているため、実装の信頼性をより高めるため、「clb_ex17_one_shot_pwm」の例に示す CLB ベースのワンショット PWM 実装を参照してください。

1.16.1.12 EPWM アクション・フィルタ (epwm_up_aq)

ファイル : epwm_ex13_up_aq.c

この例では、EPWMxA と EPWMxB で独立した変調を行う波形を生成するように、ePWM1、ePWM2、ePWM3 を設定します。

比較値 CMPA および CMPB は ePWM の ISR 内で変更されます。

この例では、TB カウンタはアップ・カウント・モードになっています。

EEPWM1A/B(GPIO0 & GPIO1)、EPWM2A/B(GPIO2 & GPIO3)、EPWM3A/B(GPIO4 & GPIO5) の波形をオシロスコープで観測します。

1.17 ePWM レジスタ

このセクションでは、拡張パルス幅変調器レジスタについて説明します。

1.17.1 EPWM のベース・アドレス表

表 1-21. EPWM のベース・アドレス表

ビット・フィールド名		DriverLib 名	ベース・アドレス	パイプライン保護
インスタンス	構造			
EPwm1Regs	EPWM_REGS	EPWM1_BASE	0x0000_4000	あり
EPwm2Regs	EPWM_REGS	EPWM2_BASE	0x0000_4100	あり
EPwm3Regs	EPWM_REGS	EPWM3_BASE	0x0000_4200	あり
EPwm4Regs	EPWM_REGS	EPWM4_BASE	0x0000_4300	あり
EPwm5Regs	EPWM_REGS	EPWM5_BASE	0x0000_4400	あり
EPwm6Regs	EPWM_REGS	EPWM6_BASE	0x0000_4500	あり
EPwm7Regs	EPWM_REGS	EPWM7_BASE	0x0000_4600	あり

1.17.2 EPWM_REGS レジスタ

EPWM_REGS レジスタのメモリマップされたレジスタを、表 1-23 に示します。表 1-23 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更してはいけません。

表 1-22. Scattershot によるレビュー

バージョン	4.0
入力 IP-XACT ファイル	epwm_registers_topoa_ipxact.xml
出力 DZ ファイル	EPWM_REGS_Registers_epwm_registers_topoa_ipxact_dz.xml
最後の実行	2023 年 5 月 12 日 (金) 11 : 08 : 38

表 1-23. EPWM_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	TBCTL	時間ベース制御レジスタ		表示
1h	TBCTL2	時間ベース制御レジスタ 2		表示
3h	EPWMSYNCINSEL	EPWMxSYNCIN ソース選択レジスタ		表示
4h	TBCTR	時間ベース・カウンタ・レジスタ		表示
5h	TBSTS	時間ベース・ステータス・レジスタ		表示
6h	EPWMSYNCOUTEN	EPWMxSYNCOUT ソース・イネーブル・レジスタ		表示
7h	TBCTL3	時間ベース制御レジスタ 3		表示
8h	CMPCTL	カウンタ比較制御レジスタ		表示
9h	CMPCTL2	カウンタ比較制御レジスタ 2		表示
Ch	DBCTL	デッドバンド・ジェネレータ制御レジスタ		表示
Dh	DBCTL2	デッドバンド・ジェネレータ制御レジスタ 2		表示
10h	AQCTL	アクション・フィルタ制御レジスタ		表示
11h	AQTSRCSEL	アクション・フィルタ・トリガ・イベント・ソース選択レジスタ		表示
14h	PCCTL	PWM チョップ制御レジスタ		表示
18h	VCAPCTL	バレー・キャプチャ制御レジスタ		表示
19h	VCNTCFG	バレー・カウンタ構成レジスタ		表示
20h	HRCNFG	HRPWM 構成レジスタ	EALLOW	表示
21h	HRPWR	HRPWM 電力レジスタ	EALLOW	表示
22h	HRCAL	HRPWM 較正レジスタ	EALLOW	表示
23h	HRPRD	HRPWM 周期レジスタ	EALLOW	表示
24h	HRCNT0	HRPWM カウンタ 0 レジスタ	EALLOW	表示
25h	HRCNT1	HRPWM カウンタ 1 レジスタ	EALLOW	表示
26h	HRMSTEP	HRPWM MEP ステップ・レジスタ	EALLOW	表示
27h	HRCNFG2	HRPWM 構成 2 レジスタ	EALLOW	表示
2Dh	HRPCTL	高分解能周期制御レジスタ	EALLOW	表示
2Eh	TRREM	HRPWM 高分解能剰余レジスタ	EALLOW	表示
34h	GLDCTL	グローバル PWM ロード制御レジスタ	EALLOW	表示
35h	GLDCFG	グローバル PWM ロード構成レジスタ	EALLOW	表示
38h	EPWMXLINK	EPWMx リンク・レジスタ		表示
3Eh	EPWMREV	EPWM リビジョン・レジスタ		表示
3Fh	HRPWMREV	高分解能リビジョン・レジスタ		表示
40h	AQCTLA	出力 A のアクション・フィルタ制御レジスタ		表示
41h	AQCTLA2	出力 A の追加アクション・フィルタ制御レジスタ		表示

表 1-23. EPWM_REGS レジスタ (続き)

オフセット	略称	レジスタ名	書き込み保護	セクション
42h	AQCTLB	出力 B のアクション・フィルタ制御レジスタ		表示
43h	AQCTLB2	出力 B の追加アクション・フィルタ制御レジスタ		表示
47h	AQSFR	アクション・フィルタ・ソフトウェア強制レジスタ		表示
49h	AQCSFR	アクション・フィルタ連続ソフトウェア強制レジスタ		表示
50h	DBREDHR	デッドバンド・ジェネレータ立ち上がりエッジ遅延高分解能ミラー・レジスタ		表示
51h	DBRED	デッドバンド・ジェネレータ立ち上がりエッジ遅延高分解能ミラー・レジスタ		表示
52h	DBFEDHR	デッドバンド・ジェネレータ立ち下がりエッジ遅延高分解能レジスタ		表示
53h	DBFED	デッドバンド・ジェネレータ立ち下がりエッジ遅延カウント・レジスタ		表示
60h	TBPHS	時間ベース位相 High		表示
62h	TBPRDHR	時間ベース周期高分解能レジスタ		表示
63h	TBPRD	時間ベース周期レジスタ		表示
64h	TBPRDHRB	EPWMxB の計算結果		表示
6Ah	CMPA	カウンタ比較 A レジスタ		表示
6Ch	CMPB	カウンタ比較 B レジスタ		表示
6Fh	CMPC	カウンタ比較 C レジスタ		表示
71h	CMPD	カウンタ比較 D レジスタ		表示
74h	GLDCTL2	グローバル PWM ロード制御レジスタ 2		表示
77h	SWVDELVAL	ソフトウェア・バレー・モード遅延レジスタ		表示
80h	TZSEL	トリップ・ゾーン選択レジスタ	EALLOW	表示
82h	TZDSEL	トリップ・ゾーン・デジタル・コンパレータ選択レジスタ	EALLOW	表示
84h	TZCTL	トリップ・ゾーン制御レジスタ	EALLOW	表示
85h	TZCTL2	追加のトリップ・ゾーン制御レジスタ	EALLOW	表示
86h	TZCTLDCA	トリップ・ゾーン制御レジスタ・デジタル比較 A	EALLOW	表示
87h	TZCTLDCB	トリップ・ゾーン制御レジスタ・デジタル比較 B	EALLOW	表示
8Dh	TZEINT	トリップ・ゾーン・イネーブル割り込みレジスタ	EALLOW	表示
93h	TZFLG	トリップ・ゾーン・フラグ・レジスタ		表示
94h	TZCBCFLG	トリップ・ゾーン CBC フラグ・レジスタ		表示
95h	TZOSTFLG	トリップ・ゾーン OST フラグ・レジスタ		表示
97h	TZCLR	トリップ・ゾーン・クリア・レジスタ	EALLOW	表示
98h	TZCBCCLR	トリップ・ゾーン CBC クリア・レジスタ	EALLOW	表示
99h	TZOSTCLR	トリップ・ゾーン OST クリア・レジスタ	EALLOW	表示
9Bh	TZFRC	トリップ・ゾーン強制レジスタ	EALLOW	表示
A4h	ETSEL	イベント・トリガ選択レジスタ		表示
A6h	ETPS	イベント・トリガ・プリスケール・レジスタ		表示
A8h	ETFLG	イベント・トリガ・フラグ・レジスタ		表示
AAh	ETCLR	イベント・トリガ・クリア・レジスタ		表示
ACH	ETFRC	イベント・トリガ強制レジスタ		表示
Aeh	ETINTPS	イベント・トリガ割り込みプリスケール・レジスタ		表示

表 1-23. EPWM_REGS レジスタ (続き)

オフセット	略称	レジスタ名	書き込み保護	セクション
B0h	ETSOCPS	イベント・トリガ変換開始プリスケール・レジスタ		表示
B2h	ETCNTINITCTL	イベント・トリガ・カウンタ初期化制御レジスタ		表示
B4h	ETCNTINIT	イベント・トリガ・カウンタ初期化レジスタ		表示
C0h	DCTRIPSEL	デジタル比較トリップ選択レジスタ	EALLOW	表示
C3h	DCACTL	デジタル比較 A 制御レジスタ	EALLOW	表示
C4h	DCBCTL	デジタル比較 B 制御レジスタ	EALLOW	表示
C7h	DCFCTL	デジタル比較フィルタ制御レジスタ	EALLOW	表示
C8h	DCCAPCTL	デジタル比較キャプチャ制御レジスタ	EALLOW	表示
C9h	DCFOFFSET	デジタル比較フィルタ・オフセット・レジスタ		表示
CAh	DCFOFFSETCNT	デジタル比較フィルタ・オフセット・カウンタ・レジスタ		表示
CBh	DCFWINDOW	デジタル比較フィルタ・ウィンドウ・レジスタ		表示
CCh	DCFWINDOWCNT	デジタル比較フィルタ・ウィンドウ・カウンタ・レジスタ		表示
CDh	BLANKPULSEMIXSEL	ブランキング・ウィンドウ・トリガ・パルス選択レジスタ	EALLOW	表示
CFh	DCCAP	デジタル比較カウンタ・キャプチャ・レジスタ		表示
D2h	DCAHTRIPSEL	デジタル比較 AH トリップ選択	EALLOW	表示
D3h	DCALTRIPSEL	デジタル比較 AL トリップ選択	EALLOW	表示
D4h	DCBHTRIPSEL	デジタル比較 BH トリップ選択	EALLOW	表示
D5h	DCBLTRIPSEL	デジタル比較 BL トリップ選択	EALLOW	表示
FAh	EPWMLOCK	EPWM ロック・レジスタ		表示
FDh	HWVDELVAL	ハードウェア・バレー・モード遅延レジスタ		表示
FEh	VCNTVAL	ハードウェア・バレー・カウンタ・レジスタ		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-24 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-24. EPWM_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み出し 0 を返す
書き込みタイプ		
W	W	書き込み
W1C	W1C	1 を書き込むことで クリア
W1S	W1S	1 を書き込むことで セット
WOnce	WOnce	書き込み 1 回書き込む
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト 値
レジスタ・アレイ変数		

**表 1-24. EPWM_REGS のアクセス・タイプ・コード
(続き)**

アクセス・タイプ	コード	説明
i、j、k、l、m、 n		これらの変数が、レジスタ名、オフセット、アドレスで使用される場合、これらの変数はレジスタ配列の値として参照されます (そのレジスタ配列は繰り返しレジスタ・グループの一部です)。これらのレジスタ・グループは階層構造を成しており、その配列は 1 つの式で表されます。
y		この変数が、レジスタ名、オフセット、アドレスで使用される場合、この変数はレジスタ配列の値を参照します。

1.17.2.1 TBCTL レジスタ (オフセット = 0h) [リセット = 83h]

図 1-93 に、TBCTL を示し、表 1-25 に、その説明を示します。

概略表に戻ります。

時間ベース制御レジスタ

図 1-93. TBCTL レジスタ

15	14	13	12	11	10	9	8
FREE_SOFT		PHSDIR	CLKDIV			HSPCLKDIV	
R/W-0h		R/W-0h	R/W-0h			R/W-1h	
7	6	5	4	3	2	1	0
HSPCLKDIV	SWFSYNC	RESERVED		PRDL	PHSEN	CTRMODE	
R/W-1h	R-0/W1S-0h	R-0h		R/W-0h	R/W-0h	R/W-3h	

表 1-25. TBCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	FREE_SOFT	R/W	0h	エミュレーション・モード・ビット。これらのビットは、エミュレーション・イベント中の ePWM 時間ベース・カウンタの挙動を選択します。 00：次の時間ベース・カウンタ・インクリメントまたはデクリメントの後に停止します。 01：カウンタがサイクル全体を完了した時点で停止します。 - アップ・カウント・モード：時間ベース・カウンタ = 周期 (TBCTR = TBPRD) で停止します。 - ダウン・カウント・モード：時間ベース・カウンタ = 0x00 (TBCTR = 0x00) で停止します。 - アップ・ダウン・カウント・モード：時間ベース・カウンタ = 0x00 (TBCTR = 0x00) で停止します。 1x：自走 リセットの種類：SYSRSn
13	PHSDIR	R/W	0h	位相方向ビット このビットは、時間ベース・カウンタがアップ・ダウン・カウント・モードに設定されている場合にのみ使用されます。 PHSDIR ビットは、同期イベントが発生した後に時間ベース・カウンタ (TBCTR) がカウントする方向を示し、新しい位相値が位相レジスタ (TBPHS) からロードされます。 これは、同期イベント前のカウンタの方向とは無関係です。 アップ・カウントおよびダウン・カウント・モードでは、このビットは無視されます。 0：同期イベント後、カウント・ダウンします。 1：同期イベント後、カウント・アップします。 リセットの種類：SYSRSn
12-10	CLKDIV	R/W	0h	時間ベース・クロック・プリスケール・ビット これらのビットは、時間ベース・クロック・プリスケール値を選択します (TBCLK = EPWMCLK/(HSPCLKDIV * CLKDIV))。 000：/1 (リセット時のデフォルト) 001：/2 010：/4 011：/8 100：/16 101：/32 110：/64 111：/128 リセットの種類：SYSRSn

表 1-25. TBCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9-7	HSPCLKDIV	R/W	1h	高速時間ベース・クロック・プリスケール・ビット これらのビットは、時間ベース・クロック・プリスケール値の一部を決定します。 $TBCLK = EPWMCLK / (HSPCLKDIV \times CLKDIV)$ 。この除数は、イベント・マネージャ (EV) ペリフェラルで使用される TMS320x281x システムの HSPCLK に相当します。 000 : /1 001 : /2 (リセット時のデフォルト) 010 : /4 011 : /6 100 : /8 101 : /10 110 : /12 111 : /14 リセットの種類 : SYSRSn
6	SWFSYNC	R-0/W1S	0h	ソフトウェア強制同期パルス 0 : 0 を書き込んでも何の影響も及ぼさず、読み出すと常に 0 が返されます。 1 : 1 を書き込むと、1 回限りの同期パルスが生成されます。 EPWMSYNCOUTEN.SWEN ビットをセットすると、SWFSYNC が有効化され、EPWMxSYNCO に効果を及ぼすことができます。 リセットの種類 : SYSRSn
5-4	RESERVED	R	0h	予約済み
3	PRDL2	R/W	0h	シャドウからのアクティブ周期レジスタのロード選択 0 : 時間ベース・カウンタ (TBCTR) がゼロと一致し、かつ / または TBCTL2[PRDLDSYNC] ビットで設定された同期イベントが発生した場合、周期レジスタ (TBPRD) はそのシャドウ・レジスタからロードされます。 TBPRD レジスタに対する書き込み / 読み出しによって、シャドウ・レジスタへのアクセスが行われます。 1 : 即時モード (シャドウ・レジスタはバイパス) : TBPRD レジスタに対する書き込みまたは読み出しによって、アクティブ・レジスタへのアクセスが行われます。 リセットの種類 : SYSRSn
2	PHSEN	R/W	0h	位相レジスタからのカウンタ・レジスタのロード・イネーブル 0 : 時間ベース位相レジスタ (TBPHS) から時間ベース・カウンタ (TBCTR) へのロードは行われません。 1 : EPWMxSYNCI 入力信号またはソフトウェア強制同期信号 (ビット 6 を参照) が発生した際、位相レジスタ (TBPHS) とシャドウ・アクティブ・ロード・イベントによってカウンタがロードされます。 リセットの種類 : SYSRSn
1-0	CTRMODE	R/W	3h	カウンタ・モード 時間ベース・カウンタ・モードは通常は 1 回設定され、通常動作中は変更されません。カウンタのモードを変更すると、その変更は次の TBCLK エッジで効力を生じ、現在のカウンタ値はモード変更前の値からインクリメントまたはデクリメントします。これらのビットによって、時間ベース・カウンタの動作モードが次のように設定されます。 00 : アップ・カウント・モード 01 : ダウン・カウント・モード 10 : アップ・ダウン・カウント・モード 11 : カウンタ動作を停止 (リセット時のデフォルト) リセットの種類 : SYSRSn

1.17.2.2 TBCTL2 レジスタ (オフセット = 1h) [リセット = 0h]

図 1-94 に、TBCTL2 を示し、表 1-26 に、その説明を示します。

概略表に戻ります。

時間ベース制御レジスタ 2

図 1-94. TBCTL2 レジスタ

15		14		13		12		11		10		9		8	
PRDLDSYNC				RESERVED				RESERVED							
R/W-0h				R-0h				R-0-0h							
7		6		5		4		3		2		1		0	
OSHTSYNC		OSHTSYNCMODE		SELFCLRTRREM RESERVED		RESERVED									
R-0/W1S-0h		R/W-0h		R/W-0h R/W-0h		R-0-0h									

表 1-26. TBCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	PRDLDSYNC	R/W	0h	同期イベント時のシャドウからのアクティブ周期レジスタのロード 00: TBPRD のシャドウ - アクティブ・ロードは、TBCTR = 0 になったときにのみ行われます (レガシーと同じ)。 01: TBPRD のシャドウ - アクティブ・ロードは、TBCTR = 0 になったときと、同期が発生したときの両方で行われます。 10: TBPRD のシャドウ - アクティブ・ロードは、同期を受信したときにのみ行われます。 11: 予約済み 注: このビット選択は、TBCTL[PRDL] = 0 の場合にのみ有効です。 リセットの種類: SYSRSn
13-12	RESERVED	R	0h	予約済み
11-8	RESERVED	R-0	0h	予約済み
7	OSHTSYNC	R-0/W1S	0h	ワンショット同期ビット 0: 0 を書き込んでも何もしません。 1: 1 つの同期パルスを伝播させます。 リセットの種類: SYSRSn
6	OSHTSYNCMODE	R/W	0h	ワンショット同期イネーブル・ビット 0: ワンショット同期モードを無効化します。 1: ワンショット同期モードを有効化します。 リセットの種類: SYSRSn
5	SELFCLRTRREM	R/W	0h	自己同期動作を可能にするためのループバック同期パルス 0: TRREM のセルフ・クリア機能を無効化します。 1: TRREM のセルフ・クリア機能を有効化します。 リセットの種類: SYSRSn
5	RESERVED	R/W	0h	予約済み
4-0	RESERVED	R-0	0h	予約済み

1.17.2.3 EPWMSYNCINSEL レジスタ (オフセット = 3h) [リセット = 1h]

図 1-95 に、EPWMSYNCINSEL を示し、表 1-27 に、その説明を示します。

概略表に戻ります。

EPWMxSYNCIN ソース選択レジスタ

図 1-95. EPWMSYNCINSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SEL			
R-0h				R/W-1h			

表 1-27. EPWMSYNCINSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-5	RESERVED	R	0h	予約済み
4-0	SEL	R/W	1h	これらのビットによって、EPWMxSYNCl 信号のソースが決定されます。 0x00 無効 「ePWM 同期選択」表に定義されているその他の値 リセットの種類：SYSRSn

1.17.2.4 TBCTR レジスタ (オフセット = 4h) [リセット = 0h]

図 1-96 に、TBCTR を示し、表 1-28 に、その説明を示します。

概略表に戻ります。

時間ベース・カウンタ・レジスタ

図 1-96. TBCTR レジスタ

15	14	13	12	11	10	9	8
TBCTR							
R/W-0h							
7	6	5	4	3	2	1	0
TBCTR							
R/W-0h							

表 1-28. TBCTR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	TBCTR	R/W	0h	時間ベース・カウンタ・レジスタ リセットの種類：SYSRSn

1.17.2.5 TBSTS レジスタ (オフセット = 5h) [リセット = 1h]

図 1-97 に、TBSTS を示し、表 1-29 に、その説明を示します。

概略表に戻ります。

時間ベース・ステータス・レジスタ

図 1-97. TBSTS レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					CTRMX	SYNCI	CTDIR
R-0-0h					R/W1C-0h	R/W1C-0h	R-1h

表 1-29. TBSTS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-3	RESERVED	R-0	0h	予約済み
2	CTRMX	R/W1C	0h	時間ベース・カウンタ最大ラッチ・ステータス・ビット 0: 0 の読み出しは、時間ベース・カウンタが最大値に達していないことを示します。0 を書き込んでも何もしません。 1: このビットの 1 の読み出しは、時間ベース・カウンタが最大値 0xFFFF に達したことを示します。このビットに 1 を書き込むと、ラッチされたイベントがクリアされます。 リセットの種類: SYSRSn
1	SYNCI	R/W1C	0h	入力同期ラッチ・ステータス・ビット 0: 0 を書き込んでも何もしません。0 の読み出しは、外部同期イベントが発生していないことを示します。 1: このビットの 1 の読み出しは、外部同期イベントが発生したことを示します (EPWMxSYNCI)。このビットに 1 を書き込むと、ラッチされたイベントがクリアされます。 リセットの種類: SYSRSn
0	CTDIR	R	1h	時間ベース・カウンタ方向ステータス・ビット 0: 時間ベース・カウンタは現在カウント・ダウン中です。 1: 時間ベース・カウンタは現在カウント・アップ中です。 注: このビットは、カウンタが停止していない場合にのみ有効です。 リセットの種類: SYSRSn

1.17.2.6 EPWMSYNCOOUT レジスタ (オフセット = 6h) [リセット = 1h]

図 1-98 に、EPWMSYNCOOUT を示し、表 1-30 に、その説明を示します。

概略表に戻ります。

EPWMxSYNCOOUT ソース・ イネーブル・ レジスタ

図 1-98. EPWMSYNCOOUT レジスタ

15								14								13								12								11								10								9								8							
RESERVED																																																															
R-0h																																																															
7								6								5								4								3								2								1								0							
RESERVED								DCBEVT1EN								DCAEVT1EN								CMPDEN								CMPCEN								CMPBEN								ZEROEN								SWEN							
R-0h								R/W-0h								R/W-0h								R/W-0h								R/W-0h								R/W-0h								R/W-0h								R/W-1h							

表 1-30. EPWMSYNCOOUT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R	0h	予約済み
7	RESERVED	R	0h	予約済み
6	DCBEVT1EN	R/W	0h	このビットは、DCBEVT1.sync イベントが EPWMxSYNCO 信号をセットできるようにします。 0 無効化 1 DCBEVT1.sync イベント時に、EPWMxSYNCO 信号は 1 PWM クロック周期のパルスを生成します。 リセットの種類: SYSRSn
5	DCAEVT1EN	R/W	0h	このビットは、DCAEVT1.sync イベントが EPWMxSYNCOOUT 信号をセットできるようにします。 0 無効化 1 DCAEVT1.sync イベント時に、EPWMxSYNCOOUT 信号は 1 PWM クロック周期のパルスを生成します。 リセットの種類: SYSRSn
4	CMPDEN	R/W	0h	このビットは、TBCTR = CMPD イベントが EPWMxSYNCO 信号をセットできるようにします。 0 無効化 1 時間ベース・カウンタがカウンタ比較 D イベント (TBCTR = CMPD) と一致すると、EPWMxSYNCO 信号は 1 PWM クロック周期のパルスを生成します。 リセットの種類: SYSRSn
3	CMPCEN	R/W	0h	このビットは、TBCTR = CMPC イベントが EPWMxSYNCO 信号をセットできるようにします。 0 無効化 1 時間ベース・カウンタがカウンタ比較 C イベント (TBCTR = CMPC) と一致すると、EPWMxSYNCO 信号は 1 PWM クロック周期のパルスを生成します。 リセットの種類: SYSRSn
2	CMPBEN	R/W	0h	このビットは、TBCTR = CMPB イベントが EPWMxSYNCO 信号をセットできるようにします。 0 無効化 1 時間ベース・カウンタがカウンタ比較 B イベント (TBCTR = CMPB) と一致すると、EPWMxSYNCO 信号は 1 PWM クロック周期のパルスを生成します。 リセットの種類: SYSRSn
1	ZEROEN	R/W	0h	このビットは、TBCTR = 0x0000 イベントが EPWMxSYNCOOUT 信号をセットできるようにします。 0 無効化 1 TBCTR の値が 0x0000 に変化すると、EPWMxSYNCOOUT 信号は 1 PWM クロック周期のパルスを生成します。 リセットの種類: SYSRSn

表 1-30. EPWMSYNCOUEN レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
0	SWEN	R/W	1h	このビットは、TBCTL.SWFSYNC ビットが EPWMxSYNCO 信号を セットできるようにします。 0 無効化 1 TBCTL.SWFSYNC ビットがセットされると、EPWMxSYNCO 信 号は 1 PWM クロック周期のパルスを生成します。 リセットの種類：SYSRSn

1.17.2.7 TBCTL3 レジスタ (オフセット = 7h) [リセット = 0h]

図 1-99 に、TBCTL3 を示し、表 1-31 に、その説明を示します。

概略表に戻ります。

時間ベース制御レジスタ 3

図 1-99. TBCTL3 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							OSSFRGEN
R-0h							R/W-0h

表 1-31. TBCTL3 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-1	RESERVED	R	0h	予約済み
0	OSSFRGEN	R/W	0h	このビットは、どちらのビットが EPWMxSYNCOOUT ワンショット・ラッチをセットするかを決定します。 0 TBCTL2[OSHTSYNC] がワンショット・ラッチをセットします。 1 GLDCTL2[OSHTLD] がワンショット・ラッチをセットします。 リセットの種類 SYSRSn

1.17.2.8 CMPCTL レジスタ (オフセット = 8h) [リセット = 0h]

図 1-100 に、CMPCTL を示し、表 1-32 に、その説明を示します。

概略表に戻ります。

カウンタ比較制御レジスタ

図 1-100. CMPCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED		LOADBSYNC		LOADASYNC		SHDWBFULL	SHDWAFULL
R-0-0h		R/W-0h		R/W-0h		R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h		R/W-0h	

表 1-32. CMPCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	RESERVED	R-0	0h	予約済み
13-12	LOADBSYNC	R/W	0h	同期イベント時のシャドウからのアクティブ CMPB レジスタのロード 00 : CMPB : CMPBHR のシャドウ - アクティブ・ロードは、LOADBMODE (ビット 1, 0) に従って行われます (従来と同じ)。 01 : CMPB : CMPBHR のシャドウ - アクティブ・ロードは、LOADBMODE ビットに従って、または同期が発生したときに行われます。 10 : CMPB : CMPBHR のシャドウ - アクティブ・ロードは、同期を受信したときのみ行われます。 11 : 予約済み 注 : このビットは、CMPCTL[SHDWBMODE] = 0 の場合にのみ有効です。 リセットの種類 : SYSRSn
11-10	LOADASYNC	R/W	0h	同期イベント時のシャドウからのアクティブ CMPA レジスタのロード 00 : CMPA : CMPAHR のシャドウ - アクティブ・ロードは、LOADBMODE (ビット 1, 0) に従って行われます (従来と同じ)。 01 : CMPA : CMPAHR のシャドウ - アクティブ・ロードは、LOADBMODE ビットに従って、または同期が発生したときに行われます。 10 : CMPA : CMPAHR のシャドウ - アクティブ・ロードは、同期を受信したときのみ行われます。 11 : 予約済み 注 : このビットは、CMPCTL[SHDWAMODE] = 0 の場合にのみ有効です。 リセットの種類 : SYSRSn
9	SHDWBFULL	R	0h	カウンタ比較 B (CMPB) シャドウ・レジスタ・フル・ステータス・フラグ このビットは、ロードストローブが発生すると自動的にクリアされます。 0 : CMPB シャドウ・レジスタはまだ完全には満たされていません。 1 : CMPB シャドウ・レジスタが完全に満たされていることを示します。 CPU への書き込みは現在のシャドウ値を上書きします。 リセットの種類 : SYSRSn

表 1-32. CMPCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
8	SHDWAFULL	R	0h	カウンタ比較 A (CMPA) シャドウ・レジスタ・フル・ステータス・フラグ CMPA : CMPAHR レジスタへの 32 ビット書き込みまたは CMPA レジスタへの 16 ビット書き込みが行われると、このフラグ・ビットがセットされます。CMPAHR レジスタへの 16 ビット書き込みは、このフラグに影響を及ぼしません。このビットは、ロードストローブが発生すると自動的にクリアされます。 0 : CMPA シャドウ・レジスタはまだ完全には満たされていません。 1 : CMPA シャドウ・レジスタが完全に満たされていることを示します。CPU への書き込みは現在のシャドウ値を上書きします。 リセットの種類 : SYSRSn
7	RESERVED	R-0	0h	予約済み
6	SHDWBMODE	R/W	0h	カウンタ比較 B (CMPB) レジスタ動作モード 0 : シャドウ・モード。ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われます。 1 : 即時モード。アクティブ比較 B レジスタのみが使用されます。即時の比較アクションの場合、すべての書き込みと読み出しはアクティブ・レジスタに対して直接行われます。 リセットの種類 : SYSRSn
5	RESERVED	R-0	0h	予約済み
4	SHDWAMODE	R/W	0h	カウンタ比較 A (CMPA) レジスタ動作モード 0 : シャドウ・モード。ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われます。 1 : 即時モード。アクティブ比較レジスタのみが使用されます。即時の比較アクションの場合、すべての書き込みと読み出しはアクティブ・レジスタに対して直接行われます。 リセットの種類 : SYSRSn
3-2	LOADBMODE	R/W	0h	シャドウからのアクティブ・カウンタ比較 B (CMPB) のロード選択モード このビットは、即時モード (CMPCTL[SHDWBMODE] = 1) では何の影響も及ぼしません。 00 : CTR = ゼロ時にロード : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 : CTR = PRD 時にロード : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 : CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 : 停止 (ロード不可) リセットの種類 : SYSRSn
1-0	LOADAMODE	R/W	0h	シャドウからのアクティブ・カウンタ比較 A (CMPA) のロード選択モード このビットは、即時モード (CMPCTL[SHDWAMODE] = 1) では何の影響も及ぼしません。 00 : CTR = ゼロ時にロード : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 : CTR = PRD 時にロード : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 : CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 : 停止 (ロード不可) リセットの種類 : SYSRSn

1.17.2.9 CMPCTL2 レジスタ (オフセット = 9h) [リセット = 0h]

図 1-101 に、CMPCTL2 を示し、表 1-33 に、その説明を示します。

概略表に戻ります。

カウンタ比較制御レジスタ 2

図 1-101. CMPCTL2 レジスタ

15	14	13	12	11	10	9	8
RESERVED		LOADDSYNC		LOADCSYNC		RESERVED	
R-0-0h		R/W-0h		R/W-0h		R-0-0h	
7	6	5	4	3	2	1	0
RESERVED	SHDWDMODE	RESERVED	SHDWCMODE	LOADDMODE		LOADCMODE	
R-0-0h	R/W-0h	R-0-0h	R/W-0h	R/W-0h		R/W-0h	

表 1-33. CMPCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	RESERVED	R-0	0h	予約済み
13-12	LOADDSYNC	R/W	0h	同期イベント時のシャドウからのアクティブ CMPD レジスタのロード 00: CMPD のシャドウ - アクティブ・ロードは、LOADDMODE に従って行われます。 01: CMPD のシャドウ - アクティブ・ロードは、LOADDMODE ビットに従って、または同期が発生したときに行われます。 10: CMPD シャドウ - アクティブ・ロードは、同期を受信したときのみ行われます。 11: 予約済み 注: このビットは、CMPCTL2[SHDWDMODE] = 0 の場合にのみ有効です。 リセットの種類: SYSRSn
11-10	LOADCSYNC	R/W	0h	同期イベント時のシャドウからのアクティブ CMPC レジスタのロード 00: CMPC のシャドウ - アクティブ・ロードは、LOADCMODE に従って行われます。 01: CMPC のシャドウ - アクティブ・ロードは、LOADCMODE ビットに従って、または同期が発生したときに行われます。 10: CMPC のシャドウ - アクティブ・ロードは、同期を受信したときのみ行われます。 11: 予約済み 注: このビットは、CMPCTL2[SHDWCMODE] = 0 の場合にのみ有効です。 リセットの種類: SYSRSn
9-7	RESERVED	R-0	0h	予約済み
6	SHDWDMODE	R/W	0h	カウンタ比較 D レジスタ動作モード 0: シャドウ・モード - ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われず。 1: 即時モード - アクティブ比較レジスタのみが使用されます。即時の比較アクションの場合、CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 リセットの種類: SYSRSn
5	RESERVED	R-0	0h	予約済み
4	SHDWCMODE	R/W	0h	カウンタ比較 C レジスタ動作モード 0: シャドウ・モード - ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われず。 1: 即時モード - アクティブ比較レジスタのみが使用されます。即時の比較アクションの場合、CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 リセットの種類: SYSRSn

表 1-33. CMPCTL2 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-2	LOADDMODE	R/W	0h	シャドウからのアクティブ・カウンタ比較 D (CMPD) のロード選択モード 00 : CTR = ゼロ時にロード : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 : CTR = PRD 時にロード : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 : CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 : 停止 (ロード不可) 注 : 即時モードでは何の影響も及ぼしません。 リセットの種類 : SYSRSn
1-0	LOADCMODE	R/W	0h	シャドウからのアクティブ・カウンタ比較 C (CMPC) のロード選択モード 00 : CTR = ゼロ時にロード : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 : CTR = PRD 時にロード : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 : CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 : 停止 (ロード不可) 注 : 即時モードでは何の影響も及ぼしません。 リセットの種類 : SYSRSn

1.17.2.10 DBCTL レジスタ (オフセット = Ch) [リセット = 0h]

図 1-102 に、DBCTL を示し、表 1-34 に、その説明を示します。

概略表に戻ります。

デッドバンド・ジェネレータ制御レジスタ

図 1-102. DBCTL レジスタ

15		14		13		12		11		10		9		8	
HALFCYCLE		DEDB_MODE		OUTSWAP		SHDWDBFED MODE		SHDWDBRED MODE		LOADFEDMODE					
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h			
7		6		5		4		3		2		1		0	
LOADREDMODE				IN_MODE				POLSEL				OUT_MODE			
R/W-0h				R/W-0h				R/W-0h				R/W-0h			

表 1-34. DBCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	HALFCYCLE	R/W	0h	半周期クロック・イネーブル・ビット 0：全周期クロックが有効。デッドバンド・カウンタは TBCLK のレートのクロックで動作します。 1：半周期クロックが有効。デッドバンド・カウンタは TBCLK*2 のクロックで動作します。 リセットの種類：SYSRSn
14	DEDB_MODE	R/W	0h	デッドバンド・デュアル・エッジ B モード制御 (S8 スイッチ) 0：A 信号パスのみで、S4 スイッチ (IN-MODE ビット) の選択に従って、立ち上がりエッジ遅延が InA/InB に適用されます。B 信号パスのみで、S5 スイッチ (INMODE ビット) の選択に従って、立ち下がりエッジ遅延が InA/InB に適用されます。 1：S4 スイッチ (INMODE ビット) によって選択されたソースに立ち上がりエッジ遅延と立ち下がりエッジ遅延が適用され、B 信号パスにのみ出力されます。注：このビットを 1 に設定している場合、ユーザーは常に OUT_MODE ビット (A パス = InA の場合) と OUTSWAP ビット (OutA = B パスの場合) のどちらかをセットする必要があります。 それ以外の場合、OutA は無効になります。 リセットの種類：SYSRSn
13-12	OUTSWAP	R/W	0h	デッドバンド出力スワップ制御 ビット 13 は S6 スイッチを制御し、ビット 12 は S7 スイッチを制御します。 00：OutA および OutB 信号は OUT-MODE ビットの設定に従います。 01：OutA = OUT-MODE ビットの設定に従う A パス。 OutB = OUT-MODE ビットの設定に従う A パス (立ち上がりエッジ遅延または遅延がバイパスされた A 信号パス)。 10：OutA = OUT-MODE ビットの設定に従う B パス (立ち下がりエッジ遅延または遅延がバイパスされた B 信号パス)。 OutB = OUT-MODE ビットの設定に従う B パス。 11：OutA = OUT-MODE ビットの設定に従う B パス (立ち下がりエッジ遅延または遅延がバイパスされた B 信号パス)。 OutB = OUT-MODE ビットの設定に従う A パス (立ち上がりエッジ遅延または遅延がバイパスされた A 信号パス)。 リセットの種類：SYSRSn

表 1-34. DBCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
11	SHDWDBFEDMODE	R/W	0h	FED デッドバンド・ロード・モード 0: 即時モード。アクティブな DBFED レジスタのみが使用されます。即時の「FED デッドバンド・アクション」の場合、CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 1: シャドウ・モード。ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われます。リセット時のデフォルトは即時モードです (レガシーとの互換性を確保するため)。 リセットの種類: SYSRSn
10	SHDWDBREDDMODE	R/W	0h	RED デッドバンド・ロード・モード 0: 即時モード。アクティブな DBRED レジスタのみが使用されます。即時の「RED デッドバンド・アクション」の場合、CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 1: シャドウ・モード。ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われます。リセット時のデフォルトは即時モードです (レガシーとの互換性を確保するため)。 リセットの種類: SYSRSn
9-8	LOADFEDMODE	R/W	0h	シャドウからのアクティブ DBFED のロード選択モード 00: カウンタ = 0 (CNT_eq) 時にロード 01: カウンタ = 周期 (PRD_eq) 時にロード 10: カウンタ = 0 時とカウンタ = 周期時のどちらかにロード 11: 停止 (ロード不可) 注: 即時モードでは何の影響も及ぼしません。 リセットの種類: SYSRSn
7-6	LOADREDDMODE	R/W	0h	シャドウからのアクティブ DBRED のロード選択モード 00: カウンタ = 0 (CNT_eq) 時にロード 01: カウンタ = 周期 (PRD_eq) 時にロード 10: カウンタ = 0 時とカウンタ = 周期時のどちらかにロード 11: 停止 (ロード不可) 注: 即時モードでは何の影響も及ぼしません。 リセットの種類: SYSRSn
5-4	IN_MODE	R/W	0h	デッドバンド入力モード制御 ビット 5 は S5 スイッチを制御し、ビット 4 は S4 スイッチを制御します。これにより、立ち下がりエッジ遅延と立ち上がりエッジ遅延の入力ソースを選択できます。伝統的なデッドバンド波形を生成するため、デフォルトは、EPWMxA 入力を立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方のソースにすることです。 00: (アクション・フィルタからの) EPWMxA 入力を立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方のソースにします。 01: (アクション・フィルタからの) EPWMxB 入力を立ち上がりエッジ遅延のソースにします。 (アクション・フィルタからの) EPWMxA 入力を立ち下がりエッジ遅延のソースにします。 10: (アクション・フィルタからの) EPWMxA 入力を立ち上がりエッジ遅延のソースにします。 (アクション・フィルタからの) EPWMxB 入力を立ち下がりエッジ遅延のソースにします。 11: (アクション・フィルタからの) EPWMxB 入力を立ち下がりエッジ遅延と立ち上がりエッジ遅延の両方のソースにします。 リセットの種類: SYSRSn

表 1-34. DBCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-2	POLSEL	R/W	0h	極性選択制御 ビット 3 は S3 スイッチを制御し、ビット 2 は S2 スイッチを制御します。これにより、遅延された信号がデッドバンド・サブモジュールから送信される前に、その信号の 1 つを選択的に反転できます。以下の説明は、デジタル・モーター制御インバータの 1 つのレッグに見られる伝統的な上側 / 下側スイッチ制御に対応しています。これらは「DBCTL[OUT_MODE] = 1,1」かつ「DBCTL[IN_MODE] = 0x0」を仮定しています。他の拡張モードも可能ですが、代表的な使用モードとはみなされていません。 00: アクティブ High (AH) モード。EPWMxA も EPWMxB も反転されません (デフォルト)。 01: アクティブ Low 相補 (ALC) モード。EPWMxA が反転されます。 10: アクティブ High 相補 (AHC) モード。EPWMxB が反転されます。 11: アクティブ Low (AL) モード。EPWMxA と EPWMxB の両方が反転されます。 リセットの種類: SYSRSn
1-0	OUT_MODE	R/W	0h	デッドバンド出力モード制御 ビット 1 は S1 スイッチを制御し、ビット 0 は S0 スイッチを制御します。 00: DBM は完全に無効化 (バイパス) されます。このモードでは、POLSEL ビットと IN-MODE ビットは何の影響も及ぼしません。 01: Apath = InA (A 信号バスの遅延はバイパスされます。) Bpath = FED (B 信号バスは立ち下がりエッジ遅延) 10: Apath = RED (A 信号バスは立ち上がりエッジ遅延) Bpath = InB (B 信号バスの遅延はバイパスされます。) 11: DBM は完全に有効化されます (つまり、RED と FED の両方が作用します。) リセットの種類: SYSRSn

1.17.2.11 DBCTL2 レジスタ (オフセット = Dh) [リセット = 0h]

図 1-103 に、DBCTL2 を示し、表 1-35 に、その説明を示します。

概略表に戻ります。

デッドバンド・ジェネレータ制御レジスタ 2

図 1-103. DBCTL2 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED					SHDWDBCTLM ODE	LOADDBCTLMODE	
R-0-0h					R/W-0h	R/W-0h	

表 1-35. DBCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-3	RESERVED	R-0	0h	予約済み
2	SHDWDBCTLMODE	R/W	0h	DBCTL ロード・モード 0：即時モード - アクティブ DBCTL レジスタのみが使用されます。CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 1：シャドウ・モード - DBCTL レジスタのビット [5 : 0] に対するすべての書き込みと読み出しはシャドウイングされます。その他のすべてのビットのアクセスは引き続きアクティブ・レジスタに対して行われます。 リセットの種類：SYSRSn
1-0	LOADDBCTLMODE	R/W	0h	シャドウからのアクティブ DBCTL のロード選択モード 00：カウンタ = 0 (CNT_eq) 時にロード 01：カウンタ = 周期 (PRD_eq) 時にロード 10：カウンタ = 0 時とカウンタ = 周期時のどちらかにロード 11：停止 (ロード不可) 注：即時モードでは何の影響も及ぼしません。 リセットの種類：SYSRSn

1.17.2.12 AQCTL レジスタ (オフセット = 10h) [リセット = 0h]

図 1-104 に、AQCTL を示し、表 1-36 に、その説明を示します。

概略表に戻ります。

アクション・フィルタ制御レジスタ

図 1-104. AQCTL レジスタ

15		14		13		12		11		10		9		8	
RESERVED								LDAQBSYNC				LDAQASYNC			
R-0-0h								R/W-0h				R/W-0h			
7		6		5		4		3		2		1		0	
RESERVED		SHDWAQBMODE		RESERVED		SHDWAQAMODE		LDAQBMODE				LDAQAMODE			
R-0-0h		R/W-0h		R-0-0h		R/W-0h		R/W-0h				R/W-0h			

表 1-36. AQCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-10	LDAQBSYNC	R/W	0h	同期イベント時のシャドウからのアクティブ AQCTLB レジスタのロード 00：AQCTLB のシャドウ - アクティブ・ロードは、LDAQBMODE に従って行われます。 01：AQCTLB のシャドウ - アクティブ・ロードは、LDAQBMODE ビットに従って、または同期が発生したときに行われます。 10：AQCTLB のシャドウ - アクティブ・ロードは、同期を受信したときのみ行われます。 11：予約済み 注：このビットは、AQCTL[SHDWAQBMODE] = 1 の場合にのみ有効です。 リセットの種類：SYSRSn
9-8	LDAQASYNC	R/W	0h	同期イベント時のシャドウからのアクティブ AQCTLA レジスタのロード 00：AQCTLA のシャドウ - アクティブ・ロードは、LDAQAMODE に従って行われます。 01：AQCTLA のシャドウ - アクティブ・ロードは、LDAQAMODE ビットに従って、または同期が発生したときに行われます。 10：AQCTLA のシャドウ - アクティブ・ロードは、同期を受信したときのみ行われます。 11：予約済み 注：このビットは、AQCTL[SHDWAQAMODE] = 1 の場合にのみ有効です。 リセットの種類：SYSRSn
7	RESERVED	R-0	0h	予約済み
6	SHDWAQBMODE	R/W	0h	アクション・フィルタ B レジスタの動作モード 1：シャドウ・モード - ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われます。 0：即時モード - アクティブ・アクション・フィルタ・レジスタのみが使用されます。CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 リセットの種類：SYSRSn
5	RESERVED	R-0	0h	予約済み

表 1-36. AQCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	SHDWAQAMODE	R/W	0h	アクション・フィルタ A レジスタの動作モード 1: シャドウ・モード - ダブル・バッファとして動作します。CPU を介したすべての書き込みはシャドウ・レジスタに対して行われます。 0: 即時モード - アクティブ・アクション・フィルタ・レジスタのみが使用されます。CPU を介したすべての書き込み / 読み出しはアクティブ・レジスタに対して直接行われます。 リセットの種類: SYSRSn
3-2	LDAQBMODE	R/W	0h	シャドウからのアクティブ・アクション・フィルタ B のロード選択モード 00: CTR = ゼロ時にロード: 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01: CTR = PRD 時にロード: 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10: CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11: 停止 (ロード不可) 注: 即時モードでは何の影響も及ぼしません。 リセットの種類: SYSRSn
1-0	LDAQAMODE	R/W	0h	シャドウからのアクティブ・アクション・フィルタ A のロード選択モード 00: CTR = ゼロ時にロード: 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01: CTR = PRD 時にロード: 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10: CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11: 停止 (ロード不可) 注: 即時モードでは何の影響も及ぼしません。 リセットの種類: SYSRSn

1.17.2.13 AQTSRCSEL レジスタ (オフセット = 11h) [リセット = 0h]

図 1-105 に、AQTSRCSEL を示し、表 1-37 に、その説明を示します。

概略表に戻ります。

アクション・フィルタ・トリガ・イベント・ソース選択レジスタ

図 1-105. AQTSRCSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
T2SEL				T1SEL			
R/W-0h				R/W-0h			

表 1-37. AQTSRCSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-4	T2SEL	R/W	0h	T2 イベント・ソース選択ビット 0000 : DCAEVT1 0001 : DCAEVT2 0010 : DCBEVT1 0011 : DCBEVT2 0100 : TZ1 0101 : TZ2 0110 : TZ3 0111 : EPWMxSYNCl 1000 : DCEVTFILT その他 : 予約済み リセットの種類 : SYSRSn
3-0	T1SEL	R/W	0h	T1 イベント・ソース選択ビット 0000 : DCAEVT1 0001 : DCAEVT2 0010 : DCBEVT1 0011 : DCBEVT2 0100 : TZ1 0101 : TZ2 0110 : TZ3 0111 : EPWMxSYNCl 1000 : DCEVTFILT その他 : 予約済み リセットの種類 : SYSRSn

1.17.2.14 PCCTL レジスタ (オフセット = 14h) [リセット = 0h]

図 1-106 に、PCCTL を示し、表 1-38 に、その説明を示します。

概略表に戻ります。

PWM チョップ制御レジスタ

図 1-106. PCCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED						CHPDUTY	
R-0-0h						R/W-0h	
7	6	5	4	3	2	1	0
CHPFREQ			OSHTWTH			CHPEN	
R/W-0h			R/W-0h			R/W-0h	

表 1-38. PCCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-11	RESERVED	R-0	0h	予約済み
10-8	CHPDUTY	R/W	0h	チョッピング・クロック・デューティ・サイクル 000 : デューティ = 1/8 (12.5%) 001 : デューティ = 2/8 (25.0%) 010 : デューティ = 3/8 (37.5%) 011 : デューティ = 4/8 (50.0%) 100 : デューティ = 5/8 (62.5%) 101 : デューティ = 6/8 (75.0%) 110 : デューティ = 7/8 (87.5%) 111 : 予約済み リセットの種類 : SYSRSn
7-5	CHPFREQ	R/W	0h	チョッピング・クロック周波数 000 : 1 分周 (分周なし、100MHz の TBCLK で 12.5MHz) 001 : 2 分周 (100MHz の TBCLK で 6.25MHz) 010 : 3 分周 (100MHz の TBCLK で 4.16MHz) 011 : 4 分周 (100MHz の TBCLK で 3.12MHz) 100 : 5 分周 (100MHz の TBCLK で 2.50MHz) 101 : 6 分周 (100MHz の TBCLK で 2.08MHz) 110 : 7 分周 (100MHz の TBCLK で 1.78MHz) 111 : 8 分周 (100MHz の TBCLK で 1.56MHz) リセットの種類 : SYSRSn
4-1	OSHTWTH	R/W	0h	ワンショット・パルス幅 0000 : 1 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 80ns) 0001 : 2 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 160ns) 0010 : 3 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 240ns) 0011 : 4 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 320ns) 0100 : 5 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 400ns) 0101 : 6 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 480ns) 0110 : 7 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 560ns) 0111 : 8 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 640ns) 1000 : 9 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 720ns) 1001 : 10 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 800ns) 1010 : 11 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 880ns) 1011 : 12 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 960ns) 1100 : 13 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 1040ns) 1101 : 14 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 1120ns) 1110 : 15 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 1200ns) 1111 : 16 x EPWMCLK / 8 の幅 (100MHz の EPWMCLK で 1280ns) リセットの種類 : SYSRSn
0	CHPEN	R/W	0h	PWM チョッピング・イネーブル 0 : PWM チョッピング機能を無効化 (バイパス) します。 1 : チョッピング機能を有効化します。 リセットの種類 : SYSRSn

1.17.2.15 VCAPCTL レジスタ (オフセット = 18h) [リセット = 0h]

図 1-107 に、VCAPCTL を示し、表 1-39 に、その説明を示します。

概略表に戻ります。

バレー・キャプチャ制御レジスタ

図 1-107. VCAPCTL レジスタ

15		14		13		12		11		10		9		8	
RESERVED										EDGEFILTDLYSEL		VDELAYDIV			
R-0-0h										R/W-0h		R/W-0h			
7		6		5		4		3		2		1		0	
VDELAYDIV		RESERVED				TRIGSEL				VCAPSTART		VCAPE			
R/W-0h		R-0-0h				R/W-0h				R-0/W1S-0h		R/W-0h			

表 1-39. VCAPCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-11	RESERVED	R-0	0h	予約済み
10	EDGEFILTDLYSEL	R/W	0h	バレー・スイッチング・モードの遅延選択 0 : エッジ・フィルタ出力に遅延を適用しません。 1 : エッジ・フィルタ出力に HWDELAYVAL の遅延を適用します。 リセットの種類 : SYSRSn
9-7	VDELAYDIV	R/W	0h	バレー遅延モード除算イネーブル 000 : HWVDELVAL = SWVDELVAL 001 : HWVDELVAL = VCNTVAL + SWVDELVAL 010 : HWVDELVAL = VCNTVAL >> 1 + SWVDELVAL 011 : HWVDELVAL = VCNTVAL >> 2 + SWVDELVAL 100 : HWVDELVAL = VCNTVAL >> 4 + SWVDELVAL 注 : これらのビットを使って、連続したエッジ・キャプチャの間の遅延値を除算することもできます。 リセットの種類 : SYSRSn
6-5	RESERVED	R-0	0h	予約済み
4-2	TRIGSEL	R/W	0h	バレー・キャプチャ・トリガの選択 000 : キャプチャ・シーケンスは、VCAPCTL[VCAPSTART] への書き込みにより、ソフトウェアによってトリガされます。 001 : キャプチャ・シーケンスは CNT_zero イベントによってトリガされます。 010 : キャプチャ・シーケンスは PRD_eq イベントによってトリガされます。 011 : キャプチャ・シーケンスは CNT_zero または PRD_eq イベントによってトリガされます。 100 : キャプチャ・シーケンスは DCAEVT1 イベントによってトリガされます。 101 : キャプチャ・シーケンスは DCAEVT2 イベントによってトリガされます。 110 : キャプチャ・シーケンスは DCBEVT1 イベントによってトリガされます。 111 : キャプチャ・シーケンスは DCBEVT2 イベントによってトリガされます。 注 : このレジスタ・フィールドで選択されたイベントによってトリガされるバレー・キャプチャ・シーケンス。選択されたイベントが発生すると、キャプチャ・シーケンスが作動可能になります。イベント・キャプチャは、DCFCTL[SRCSEL] レジスタで選択されたイベントに基づいて実行されます。 注 : DCFCTL[SRCSEL] と VCAPCTL[TRIGSEL] の両方のレジスタで同じイベントを選択することはできません。 注 : VCAPCTL[TRIGSEL] で選択されたイベントが発生すると、現在のキャプチャ・ステータスに関係なく、キャプチャ・シーケンスが再トリガされます。 リセットの種類 : SYSRSn

表 1-39. VCAPCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	VCAPSTART	R-0/W1S	0h	バレー・キャプチャの開始 0 : 0 を書き込んでも何もしません。 1 : VCAPCTL[TRIGSEL] = 0x0 の場合、キャプチャ・シーケンスを 1 回トリガします。 注：このビットは、ソフトウェアを使用してバレー・キャプチャ・シーケンスを開始するために使用されます。このビットを有効化するには、VCAPCTL[TRIGSEL] をソフトウェア・トリガとして選択する必要があります。1 を書き込むと、1 つのキャプチャ・シーケンス・トリガが生成されます。 リセットの種類：SYSRSn
0	VCAPE	R/W	0h	バレー・キャプチャの有効化 / 無効化 0 : 無効化 1 : 有効化 リセットの種類：SYSRSn

1.17.2.16 VCNTCFG レジスタ (オフセット = 19h) [リセット = 0h]

図 1-108 に、VCNTCFG を示し、表 1-40 に、その説明を示します。

概略表に戻ります。

バレー・カウンタ構成レジスタ

図 1-108. VCNTCFG レジスタ

15	14	13	12	11	10	9	8
STOPEDGESTS	RESERVED			STOPEDGE			
R-0h	R-0-0h			R/W-0h			
7	6	5	4	3	2	1	0
STARTEDGESTS	RESERVED			STARTEDGE			
R-0h	R-0-0h			R/W-0h			

表 1-40. VCNTCFG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	STOPEDGESTS	R	0h	ストップ・エッジ・ステータス・ビット 0 : ストップ・エッジが発生していません。 1 : ストップ・エッジが発生しました。 注：このビットは、(VCAPCTL[TRIGSEL] によって選択されたトリガ・パルスの発生時の)トリガ・シーケンスが作動可能になり、かつ STOPEDGE が発生した後にのみセットされます。 注：このビットは、VCAPCTL[TRIGSEL] によって選択されたトリガ・パルスが発生するとリセットされます。 リセットの種類 SYSRSn
14-12	RESERVED	R-0	0h	予約済み
11-8	STOPEDGE	R/W	0h	カウンタ・ストップ・エッジの選択 VCAPCTL[TRIGSEL] によって選択されたトリガ・パルスの発生時のカウンタ動作が作動可能な場合、このビット・フィールドによって選択されたイベント数に達した時点でバレー・カウンタはカウントを停止します。以下が発生すると、カウントを停止します。 0000 : 停止しません。 0001 : 最初のエッジ 0010 : 2 番目のエッジ 0011 : 3 番目のエッジ ... 1111 : 15 番目のエッジ リセットの種類 : SYSRSn
7	STARTEDGESTS	R	0h	スタート・エッジ・ステータス・ビット 0 : スタート・エッジが発生していません。 1 : スタート・エッジが発生しました。 注：このビットは、(VCAPCTL[TRIGSEL] で選択されたトリガ・パルスの発生時の)トリガ・シーケンスが作動可能になり、かつ STARTEDGE が発生した後にのみセットされます。 注：このビットは、VCAPCTL[TRIGSEL] によって選択されたトリガ・パルスが発生するとリセットされます。 リセットの種類 SYSRSn
6-4	RESERVED	R-0	0h	予約済み

表 1-40. VCNTCFG レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-0	STARTEDGE	R/W	0h	カウンタ・スタート・エッジの選択 VCAPCTL[TRIGSEL] によって選択されたトリガ・パルスの発生時のカウンタ動作が作動可能な場合、このビット・フィールドによって選択されたイベント数に達した時点でバレー・カウンタはカウントを開始します。以下が発生すると、カウントを開始します。 0000 : 開始しません。 0001 : 最初のエッジ 0010 : 2 番目のエッジ 0011 : 3 番目のエッジ ... 1111 : 15 番目のエッジ リセットの種類 : SYSRSn

1.17.2.17 HRCNFG レジスタ (オフセット = 20h) [リセット = 0h]

図 1-109 に、HRCNFG を示し、表 1-41 に、その説明を示します。

概略表に戻ります。

HRPWM 構成レジスタ

このレジスタは、HRPWM 機能を備えた EPWM モジュールでのみアクセスできます。

図 1-109. HRCNFG レジスタ

15	14	13	12	11	10	9	8
LINESEL RESERVED		RESERVED	HRLOADB		CTLMODEB	EDGMODEB	
R/W-0h R/W-0h		R-0-0h	R/W-0h		R/W-0h	R/W-0h	
7	6	5	4	3	2	1	0
SWAPAB	AUTOCONV	SELOUTB	HRLOAD		CTLMODE	EDGMODE	
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	

表 1-41. HRCNFG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	LINESEL	R/W	0h	ディレイ・ライン選択ビット：特定の ePWM/EPWM モジュールがキャリブレーションのために CALIN に送信するため、4 本のディレイ・ラインのいずれかを選択します。 リセットの種類：SYSRSn
15-14	RESERVED	R/W	0h	予約済み
13	RESERVED	R-0	0h	予約済み
12-11	HRLOADB	R/W	0h	シャドウ・モード・ビット CMPBHR シャドウ値をアクティブ・レジスタにロードする時間イベントを選択します。 00：CTR = ゼロ時にロード：時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01：CTR = PRD 時にロード：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10：CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11：予約済み リセットの種類：SYSRSn
10	CTLMODEB	R/W	0h	制御モード・ビット MEP を制御するレジスタ (CMP/TBPRD または TBPHS) を選択します。 0：CMPBHR(8) または TBPRDHR(8) レジスタがエッジの位置を制御します (つまり、これはデューティ (周期) 制御モードです)。(リセット時のデフォルト) 1：TBPHSHR(8) レジスタがエッジの位置を制御します (つまり、これは位相制御モードです)。 リセットの種類：SYSRSn
9-8	EDGMODEB	R/W	0h	エッジ・モード・ビット マイクロエッジ位置 (MEP) ロジックによって制御される PWM のエッジを以下のように選択します。 00：HRPWM 機能は無効 (リセット時のデフォルト) 01：立ち上がりエッジを MEP で制御 (CMPBHR) 10：立ち下がりエッジを MEP で制御 (CMPBHR) 11：両方のエッジを MEP で制御 (TBPHSHR または TBPRDHR) リセットの種類：SYSRSn
7	SWAPAB	R/W	0h	ePWM A および B 出力信号の交換 このビットを使うと、A 信号出力と B 信号出力を交換できます。その選択は次のとおりです。 0：ePWMxA と ePWMxB の出力は変更されません。 1：ePWMxA 信号は ePWMxB 出力に現れ、ePWMxB 信号は ePWMxA 出力に現れます。 リセットの種類：SYSRSn

表 1-41. HRCNFG レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
6	AUTOCONV	R/W	0h	自動変換デレイ・ライン値 CMPAHR/TBPRDHR/TBPHSHR レジスタの分数デューティ・サイクル / 周期 / 位相を HRMSTEP レジスタの MEP 係数で自動的にスケールアップするか、アプリケーション・ソフトウェア内の計算によって個別にスケールアップするかを選択します。SFO ライブラリ機能は、適切な MEP 係数で HRMSTEP レジスタを自動的に更新します。 0 : 自動的な HRMSTEP スケールアップを無効化します。 1 : 自動的な HRMSTEP スケールアップを有効化します。 アプリケーション・ソフトウェアが分数デューティ・サイクルまたは位相を個別にスケールアップする場合 (つまり、ソフトウェアがデューティ・サイクルを $CMPAHR = (\text{fraction}(PWMduty * PWMperiod) * MEP \text{ 係数}) \ll 8 + 0x080$ に設定する場合)、このモードを無効化する必要があります。 リセットの種類 : SYSRSn
5	SELOUTB	R/W	0h	EPWMxB 出力選択ビット このビットは、ePWMxB チャンネル出力にどちらの信号を出力するかを選択します。 信号の反転は、高分解能モードを考慮して行われるので、高分解能のエッジの配置は維持されます。この反転は、ePWMxB 信号の変更の最後の段階として行われます。 0 : ePWMxB 出力は反転されません。 1 : ePWMxB 出力は、ePWMxA 信号を反転させたものです。 リセットの種類 : SYSRSn
4-3	HRLOAD	R/W	0h	シャドウ・モード・ビット CMPAHR シャドウ値をアクティブ・レジスタにロードする時間イベントを選択します。 00 : CTR = ゼロ時にロード : 時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 : CTR = PRD 時にロード : 時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 : CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 : 予約済み リセットの種類 : SYSRSn
2	CTLMODE	R/W	0h	制御モード・ビット MEP を制御するレジスタ (CMP/TBPRD または TBPHS) を選択します。 0 : CMPAHR(8) または TBPRDHR(8) レジスタがエッジの位置を制御します (つまり、これはデューティ (周期) 制御モードです)。(リセット時のデフォルト) 1 : TBPHSHR(8) レジスタがエッジの位置を制御します (つまり、これは位相制御モードです)。 リセットの種類 : SYSRSn
1-0	EDGMODE	R/W	0h	エッジ・モード・ビット マイクロエッジ位置 (MEP) ロジックによって制御される PWM のエッジを以下のように選択します。 00 : HRPWM 機能は無効 (リセット時のデフォルト) 01 : 立ち上がりエッジを MEP で制御 (CMPAHR) 10 : 立ち下がりエッジを MEP で制御 (CMPAHR) 11 : 両方のエッジを MEP で制御 (TBPHSHR または TBPRDHR) リセットの種類 : SYSRSn

1.17.2.18 HRPWR レジスタ (オフセット = 21h) [リセット = 0h]

図 1-110 に、HRPWR を示し、表 1-42 に、その説明を示します。

概略表に戻ります。

HRPWM 電力レジスタ

このレジスタは、HRPWM 機能を備えた EPWM モジュールでのみアクセスできます。

図 1-110. HRPWR レジスタ

15	14	13	12	11	10	9	8
CALPWRON	RESERVED					CALSEL RESERVED	
R/W-0h		R-0-0h			R/W-0h R/W-0h		
7	6	5	4	3	2	1	0
CALSEL RESERVED		TESTSEL RESERVED	CALSTS RESERVED	CNTSEL RESERVED	CALSTART RESERVED	RESERVED	
R/W-0h R/W-0h		R/W-0h R/W-0h	R-0h R-0h	R/W-0h R/W-0h	R/W-0h R/W-0h	R/W-0h	

表 1-42. HRPWR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	CALPWRON	R/W	0h	MEP 較正電力ビット (ePWM1 でのみ利用可能) 0 : HRPWM の MEP 較正ロジックを無効化し、消費電力を低減します。 1 : MEP 較正ロジックを有効化します。 リセットの種類 : SYSRSn
14-10	RESERVED	R-0	0h	予約済み
9-6	CALSEL	R/W	0h	較正用の EPWM デイレイ・ラインの選択 : リセットの種類 : SYSRSn
9-6	RESERVED	R/W	0h	予約済み
5	TESTSEL	R/W	0h	テスト・モード選択ビット : このビットは、発振器較正モードでダミー遅延を追加 (周波数を下げるために小さい遅延を使用) するかどうかを選択します。 リセットの種類 : SYSRSn
5	RESERVED	R/W	0h	予約済み
4	CALSTS	R	0h	較正ステータス・ビット : このビットが 1 に設定されている場合、較正が進行中であることを示します。このビットは以下の場合に 0 に設定されます。 リセットの種類 : SYSRSn
4	RESERVED	R	0h	予約済み
3	CNTSEL	R/W	0h	カウンタ選択ビット : このビットの機能は変更されました。HRCNT0 または HRCNT1 が 0xFFFF に達すると、両方のカウンタが停止します。このビットは、較正の開始時期に影響を及ぼします。 リセットの種類 : SYSRSn
3	RESERVED	R/W	0h	予約済み
2	CALSTART	R/W	0h	較正の開始 / 停止ビット : リセットの種類 : SYSRSn
2	RESERVED	R/W	0h	予約済み
1-0	RESERVED	R/W	0h	予約済み

1.17.2.19 HRCAL レジスタ (オフセット = 22h) [リセット = 0h]

図 1-111 に、HRCAL を示し、表 1-43 に、その説明を示します。

[概略表](#)に戻ります。

HRPWM 較正レジスタ

図 1-111. HRCAL レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
HRCAL							
R/W-0h							

表 1-43. HRCAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-0	HRCAL	R/W	0h	これらの 8 ビットは、較正ディレイ・ライン (DCAL) に対してのみ、発振器較正時の遅延素子の数を選択するために使用されます。ユーザーは目的の遅延時間を設定してから、較正の実行を開始します。較正の実行結果に基づいて、次の較正を実行するために遅延が増減されます。 リセットの種類 : SYSRSn

1.17.2.20 HRPRD レジスタ (オフセット = 23h) [リセット = 0h]

図 1-112 に、HRPRD を示し、表 1-44 に、その説明を示します。

[概略表](#)に戻ります。

HRPWM 周期レジスタ

図 1-112. HRPRD レジスタ

15	14	13	12	11	10	9	8
HRPRD							
R/W-0h							
7	6	5	4	3	2	1	0
HRPRD							
R/W-0h							

表 1-44. HRPRD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	HRPRD	R/W	0h	これらの 8 ビットは、較正ディレイ・ライン (DCAL) に対してのみ、発振器較正時の遅延素子の数を選択するために使用されます。 リセットの種類：SYSRSn

1.17.2.21 HRCNT0 レジスタ (オフセット = 24h) [リセット = 0h]

図 1-113 に、HRCNT0 を示し、表 1-45 に、その説明を示します。

[概略表](#)に戻ります。

HRPWM カウンタ 0 レジスタ

図 1-113. HRCNT0 レジスタ

15	14	13	12	11	10	9	8
HRCNT0							
R/W-0h							
7	6	5	4	3	2	1	0
HRCNT0							
R/W-0h							

表 1-45. HRCNT0 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	HRCNT0	R/W	0h	HRCNT0 カウンタはリング発振器のクロック・パルスごとにインクリメントします。 リセットの種類: SYSRSn

1.17.2.22 HRCNT1 レジスタ (オフセット = 25h) [リセット = 0h]

図 1-114 に、HRCNT1 を示し、表 1-46 に、その説明を示します。

[概略表](#)に戻ります。

HRPWM カウンタ 1 レジスタ

図 1-114. HRCNT1 レジスタ

15	14	13	12	11	10	9	8
HRCNT1							
R/W-0h							
7	6	5	4	3	2	1	0
HRCNT1							
R/W-0h							

表 1-46. HRCNT1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	HRCNT1	R/W	0h	HRCNT1 カウンタは、システム・クロック・パルスごとにインクリメントします。 リセットの種類：SYSRSn

1.17.2.23 HRMSTEP レジスタ (オフセット = 26h) [リセット = 0h]

図 1-115 に、HRMSTEP を示し、表 1-47 に、その説明を示します。

概略表に戻ります。

HRPWM MEP ステップ・レジスタ

このレジスタは、HRPWM 機能を備えた EPWM モジュールでのみアクセスできます。このレジスタには 16 ビット・アクセスのみが許可されます。32 ビット・モードでのデバッグ・アクセスは、誤った値を表示させる場合があります。

図 1-115. HRMSTEP レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
HRMSTEP							
R/W-0h							

表 1-47. HRMSTEP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-0	HRMSTEP	R/W	0h	高分解能 MEP ステップ 自動変換が有効化されている場合 (HRCNFG[AUTOCONV] = 1)、この 8 ビット・フィールドには MEP_ScaleFactor (粗ステップあたりの MEP ステップ数) が含まれます。この MEP_ScaleFactor は、CMPAHR、CMPBHR、DBFEDHR、DBREDHR、TBPHSHR、TBPRDHR レジスタの値を、高分解能 ePWM 出力のスケールリングされたマイクロエッジ遅延にハードウェアが自動的に変換するために使用されます。このレジスタの値は、各較正実行の終了時に SFO 較正ソフトウェアによって書き込まれます。 リセットの種類 : SYSRSn

1.17.2.24 HRCNFG2 レジスタ (オフセット = 27h) [リセット = 0h]

図 1-116 に、HRCNFG2 を示し、表 1-48 に、その説明を示します。

概略表に戻ります。

HRPWM 構成 2 レジスタ

このレジスタは、HRPWM 機能を備えた EPWM モジュールでのみアクセスできます。このレジスタには 16 ビット・アクセスのみが許可されます。32 ビット・モードでのデバッグ・アクセスは、誤った値を表示させる場合があります。

図 1-116. HRCNFG2 レジスタ

15		14		13		12		11		10		9		8	
NOBYPASS RESERVED		DELLOADFRC RESERVED		RESERVED											
R/W-0h R/W-0h		R-0/W1S-0h R-0/W1S-0h		R-0-0h											
7		6		5		4		3		2		1		0	
RESERVED		RESERVED		CTLMODEDBFED		CTLMODEDBFED		CTLMODEDBRED		CTLMODEDBRED		EDGMODEDB		EDGMODEDB	
R-0-0h		R-0-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-48. HRCNFG2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	NOBYPASS	R/W	0h	非バイパス・ディレイ・ライン更新ビット：内部的なテストのため、このビットは、ディレイ・ラインが更新される前に 1 SYSCLK サイクルのバイパスを無効化します。 リセットの種類：SYSRSn
15	RESERVED	R/W	0h	予約済み
14	DELLOADFRC	R-0/W1S	0h	ソフトウェアによるディレイ・ライン・ロード強制：内部的なテストのため、ディレイ・ラインを強制的に更新するパルスがソフトウェアによって生成されます (PRD_eq/CNT_zero ストローブと同様)。 リセットの種類：SYSRSn
14	RESERVED	R-0/W1S	0h	予約済み
13-6	RESERVED	R-0	0h	予約済み
5-4	CTLMODEDBFED	R/W	0h	シャドウ・モード・ビット - 選択は DBCTL[LOADFEDMODE] と一致している必要があります。 DBFEDHR のシャドウ値をアクティブ・レジスタにロードする時間イベントを選択します。 00 CTR = ゼロ時にロード：時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 CTR = PRD 時にロード：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 予約済み リセットの種類：SYSRSn
3-2	CTLMODEDBRED	R/W	0h	シャドウ・モード・ビット - 選択は DBCTL[LOADREDMODE] と一致している必要があります。 DBREDHR のシャドウ値をアクティブ・レジスタにロードする時間イベントを選択します。 00 CTR = ゼロ時にロード：時間ベース・カウンタがゼロと一致 (TBCTR = 0x0000) 01 CTR = PRD 時にロード：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 10 CTR = ゼロ時と CTR = PRD 時のどちらかにロード 11 予約済み リセットの種類：SYSRSn

表 1-48. HRCNFG2 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1-0	EDGMOEEDB	R/W	0h	エッジ・モード・ビット マイクロエッジ位置 (MEP) ロジックによって制御される PWM のエッジを以下のように選択します。 00 HRPWM 機能は無効 (リセット時のデフォルト) 01 立ち上がりエッジを MEP で制御 (DBREDHR) 10 立ち下がりエッジを MEP で制御 (DBFEDHR) 11 両方のエッジを MEP で制御 (立ち上がりエッジは DBREDHR、立ち下がりエッジは DBFEDHR) リセットの種類: SYSRSn

1.17.2.25 HRPCTL レジスタ (アドレス = 2Dh) [リセット = 0h]

図 1-117 に、HRPCTL を示し、表 1-49 に、その説明を示します。

概略表に戻ります。

高分解能周期制御レジスタ

このレジスタは、HRPWM 機能を備えた EPWM モジュールでのみアクセスできます。

図 1-117. HRPCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	PWMSYNCESELX			HRPSYNCE RESERVED	TBPHSHRLOA DE	PWMSYNCESEL	HRPE
R-0-0h	R/W-0h			R/W-0h R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-49. HRPCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-7	RESERVED	R-0	0h	予約済み
6-4	PWMSYNCESELX	R/W	0h	EPWMSYNCPER の拡張選択ビット 000 : EPWMSYNCPER は PWMSYNCESEL によって設定 -> デフォルト条件 (前の EPWM バージョンと互換) 001 : 予約済み 010 : 予約済み 011 : 予約済み 100 : CTR = CMPC、カウント方向はアップ 101 : CTR = CMPC、カウント方向はダウン 110 : CTR = CMPD、カウント方向はアップ 111 : CTR = CMPD、カウント方向はダウン リセットの種類 : SYSRSn
3	HRPSYNCE	R/W	0h	同期イネーブル・ビット (TRSYNCE) / 高分解能周期同期イネーブル・ビット (HRPSYNCE) リセットの種類 : SYSRSn
3	RESERVED	R/W	0h	予約済み
2	TBPHSHRLOADE	R/W	0h	TBPHSHR ロード・イネーブル このビットを使用すると、SYNCIN、TBCTL[SWFSYNC]、デジタル比較イベントのいずれかと ePWM モジュールを高分解能位相で同期させることができます。これにより、同じ周波数で動作する複数の ePWM モジュールの位相を高分解能で揃えることができます。 0 : SYNCIN、TBCTL[SWFSYNC]、デジタル比較イベントの高分解能位相の同期を無効化します。 1 : SYNCIN、TBCTL[SWFSYNC]、デジタル・コンパレータ同期イベントの高分解能位相を同期させます。この位相は、高分解能位相 TBPHSHR レジスタの内容を使って同期します。SYNCIN または TBCTL[SWFSYNC] イベント時に TBPHS レジスタの値で TBCTR レジスタをロードできる TBCTL[PHSEN] ビットは、独立して機能します。ただし、高分解能周期機能と組み合わせて位相を制御する場合、このビットも有効化する必要があります。 アップ・ダウン・カウント・モードで高分解能周期が有効化されている場合、TBPHSHR = 0x0000 であっても、このビットと TBCTL[PHSEN] ビットを 1 に設定する必要があります。高分解能デューティのみが有効化されている場合、このビットをセットする必要はありません。 リセットの種類 : SYSRSn

表 1-49. HRPCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	PWMSYNCSEL	R/W	0h	PWMSYNC ソース選択ビット：このビットは、CMPSS と GPDAC に送られる EPWMSYNCPER 信号のソースを選択します。 0 CTR = PRD：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 1 CTR = ゼロ：時間ベース・カウンタがゼロと一致 (TBCTR = 0x00) リセットの種類：SYSRSn
0	HRPE	R/W	0h	高分解能周期イネーブル・ビット 0：高分解能周期機能は無効です。このモードでは、ePWM はタイプ 4 ePWM として動作します。 1：高分解能周期は有効です。このモードでは、HRPWM モジュールはデューティと周波数の両方の高分解能を制御できます。高分解能周期が有効化されている場合、TBCTL[CTRMODE] = 0,1 (ダウン・カウント・モード) はサポートされません。 リセットの種類：SYSRSn

1.17.2.26 TRREM レジスタ (アドレス = 2Eh) [リセット = 0h]

図 1-118 に、TRREM を示し、表 1-50 に、その説明を示します。

概略表に戻ります。

HRPWM 高分解能剰余レジスタ

このレジスタは、HRPWM 機能を備えた EPWM モジュールでのみアクセスできます。

図 1-118. TRREM レジスタ

15	14	13	12	11	10	9	8
RESERVED						TRREM	
R-0-0h						R/W-0h	
7	6	5	4	3	2	1	0
TRREM							
R/W-0h							

表 1-50. TRREM レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-11	RESERVED	R-0	0h	予約済み
10-0	TRREM	R/W	0h	HRPWM 剰余ビット：この 11 ビット値は、HRPWM アルゴリズム計算の剰余部分を記録します。 この値は、HRPWM ハードウェア計算の剰余部分を記録します。 注： 1. TRREM レジスタの下位 8 ビットは、SYNCIN または TBCTL[SWFSYNC] イベントまたは DC イベント (有効化されている場合) の TBPHSHR 値によって自動的に初期化できます。ユーザーは CPU を使用して値を書き込むこともできます。 2. TRREM レジスタ更新の優先度： TRREM にコピーされた同期 (ソフトウェアまたはハードウェア) TBPHSHR：最も高い優先度 HRPWM ハードウェア (TRREM レジスタを更新)：次の優先度 TRREM レジスタへの CPU 書き込み：最も低い優先度 3. TRREM レジスタのビット 10 は非対称モードでは使用されません。このビットは強制的にゼロに設定できます。 TRREM は、アップ・モードとアップ・ダウン・モードでそれぞれ 0x0 と 0x100 に初期化されます。 非対称モード： TRREM[7 : 0] = TBPHSHR[15 : 8] TRREM[10, 9, 8] = 0,0,0 対称モード： TRREM[7 : 0] = TBPHSHR[15 : 8] TRREM[10, 9, 8] = 0,0,1 リセットの種類：SYSRSn

1.17.2.27 GLDCTL レジスタ (オフセット = 34h) [リセット = 0h]

図 1-119 に、GLDCTL を示し、表 1-51 に、その説明を示します。

概略表に戻ります。

グローバル PWM ロード制御レジスタ

図 1-119. GLDCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED			GLDCNT			GLDPRD	
R-0-0h			R-0h			R/W-0h	
7	6	5	4	3	2	1	0
GLDPRD	RESERVED	OSHTMODE	GLDMODE			GLD	
R/W-0h	R-0-0h	R/W-0h	R/W-0h			R/W-0h	

表 1-51. GLDCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-13	RESERVED	R-0	0h	予約済み
12-10	GLDCNT	R	0h	グローバル・ロード・ストローブ・カウンタ・レジスタ これらのビットは、選択されたイベントの発生回数を示します。 000 : イベントなし 001 : 1 回のイベント 010 : 2 回のイベント 011 : 3 回のイベント 100 : 4 回のイベント 101 : 5 回のイベント 110 : 6 回のイベント 111 : 7 回のイベント リセットの種類 : SYSRSn
9-7	GLDPRD	R/W	0h	グローバル・ロード・ストローブ周期選択レジスタ これらのビットは、ロード・ストローブが生成されるのに必要な特定のイベントの発生回数を選択します。 000 : カウンタを無効化します。 001 : GLDCNT = 001 (最初のイベント) でストローブを生成します。 010 : GLDCNT = 010 (2 番目のイベント) でストローブを生成します。 011 : GLDCNT = 011 (3 番目のイベント) でストローブを生成します。 100 : GLDCNT = 011 (4 番目のイベント) でストローブを生成します。 101 : GLDCNT = 001 (5 番目のイベント) でストローブを生成します。 110 : GLDCNT = 010 (6 番目のイベント) でストローブを生成します。 111 : GLDCNT = 011 (7 番目のイベント) でストローブを生成します。 リセットの種類 : SYSRSn
6	RESERVED	R-0	0h	予約済み
5	OSHTMODE	R/W	0h	ワンショット・ロード・モード制御ビット 0 : ワンショット・ロード・モードが無効化され、選択されたすべてのロード・ストローブで、シャドウからアクティブへのロードが連続的に行われます。 1 : ワンショット・モードが有効化されます。GLDCTL2[OSHTLD] に 1 が書き込まれるまで、すべてのロード・ストローブが阻止されます。 注 : ワンショット・モードは、グローバル・シャドウ・アクティブ・ロード・モードを有効にした状態 (GLDCTL[GLD] = 1) のみ使用できます。 リセットの種類 : SYSRSn

表 1-51. GLDCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4-1	GLDMODE	R/W	0h	シャドウ - アクティブ・モード・リロード用グローバル・ロード・パルス選択 0000 : カウンタ = 0 (CNT_ZRO) 時にロード 0001 : カウンタ = 周期 (PRD_EQ) 時にロード 0010 : カウンタ = 0 時とカウンタ = 周期時のどちらかにロード 0011 : SYNCEVT (DCAEVT1.sync、DCBEVT1.sync、EPWMxSYNCl、TBCTL[SWFSYNC] の論理和) 時にロード 0100 : SYNCEVT 時または CNT_ZRO 時にロード 0101 : SYNCEVT 時または PRD_EQ 時にロード 0110 : SYNCEVT 時、CNT_ZRO 時、PRD_EQ 時にロード 1000 : 予約済み ... 1110 : 予約済み 1111 : GLDCTL2[GFRCLD] の書き込みでロード リセットの種類 : SYSRSn
0	GLD	R/W	0h	グローバル・シャドウ - アクティブ・ロード・イベント制御 0 : すべてのシャドウ・レジスタのシャドウ - アクティブ・リロードは、指定された個別のリロード制御ビットに従って行われます (以前の EPWM バージョンと互換性があります)。 1 : セットされている場合、すべてのシャドウ - アクティブ・リロード・イベントは GLDCTL レジスタの GLDMODE ビットによって設定されます。すべてのシャドウ・レジスタは、シャドウ - アクティブ・リロードからの同じリロード・パルスを使用します。個別の LOADMODE ビットは無視されます。 リセットの種類 : SYSRSn

1.17.2.28 GLDCFG レジスタ (オフセット = 35h) [リセット = 0h]

図 1-120 に、GLDCFG を示し、表 1-52 に、その説明を示します。

概略表に戻ります。

グローバル PWM ロード構成レジスタ

図 1-120. GLDCFG レジスタ

15		14		13		12		11		10		9		8	
RESERVED										AQCSFRC	AQCTLB_AQC TLB2	AQCTLA_AQC TLA2			
R-0-0h										R/W-0h	R/W-0h	R/W-0h		R/W-0h	
7		6		5		4		3		2		1		0	
DBCTL	DBFED_DBFE DHR	DBRED_DBRE DHR	CMPD		CMPC		CMPB_CMPBH R		CMPA_CMPAH R		TBPRD_TBPR DHR				
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		

表 1-52. GLDCFG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-11	RESERVED	R-0	0h	予約済み
10	AQCSFRC	R/W	0h	AQCSFRC のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
9	AQCTLB_AQCTLB2	R/W	0h	AQCTLB_AQCTLB2 のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
8	AQCTLA_AQCTLA2	R/W	0h	AQCTLA_AQCTLA2 のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
7	DBCTL	R/W	0h	DBCTL のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
6	DBFED_DBFEDHR	R/W	0h	DBFED_DBFEDHR のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
5	DBRED_DBREDHR	R/W	0h	DBRED_DBREDHR のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn

表 1-52. GLDCFG レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	CMPD	R/W	0h	CMPD のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
3	CMPC	R/W	0h	CMPC のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
2	CMPB_CMPBHR	R/W	0h	CMPB_CMPBHR のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
1	CMPA_CMPAHR	R/W	0h	CMPA_CMPAHR のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn
0	TBPRD_TBPRDHR	R/W	0h	TBPRD_TBPRDHR のグローバル・ロード・イベント設定 0 : GLDCTL(GLD) = 1 であっても、レジスタはローカル・リロード設定を使用します (リロードは以前の EPWM と互換性があります)。 1 : このビットがセットされており、かつ GLDCTL(GLD) = 1 の場合、レジスタはグローバル・ロード設定を使用します。 リセットの種類 : SYSRSn

1.17.2.29 EPWMXLINK レジスタ (オフセット = 38h) [リセット = X]

図 1-121 に、EPWMXLINK を示し、表 1-53 に、その説明を示します。

概略表に戻ります。

EPWMx リンク・レジスタ

このレジスタは、どの EPWM をその他の EPWM モジュールとリンクさせるかを制御します。デフォルトのリセット値はモジュールごとに異なります。モジュールの意図しないリンクを防止するため、リセット値では、各 EPWM モジュールはそれ自体にリンクされます。

図 1-121. EPWMXLINK レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GLDCTL2LINK				RESERVED								CMPDLINK			
R/W-X				R-0-0h								R/W-X			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPCLINK				CMPBLINK				CMPALINK				TBPRDLINK			
R/W-X				R/W-X				R/W-X				R/W-X			

表 1-53. EPWMXLINK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-28	GLDCTL2LINK	R/W	X	GLDCTL2 リンク・ビット 以下のビット選択によって選択された ePWM モジュールの GLDCTL2 レジスタに書き込むと、現在の ePWM モジュールの GLDCTL2 レジスタにも同時に書き込まれます。 0000 : ePWM1 0001 : ePWM2 ... 最後の ePWM インスタンスまで。その他はすべて予約済みです。 リセットの種類 : SYSRSn
27-20	RESERVED	R-0	0h	予約済み
19-16	CMPDLINK	R/W	X	CMPD リンク・ビット 以下のビット選択によって選択された ePWM モジュールの CMPD レジスタに書き込むと、現在の ePWM モジュールの CMPD レジスタにも同時に書き込まれます。 0000 : ePWM1 0001 : ePWM2 ... 最後の ePWM インスタンスまで。その他はすべて予約済みです。 リセットの種類 : SYSRSn
15-12	CMPCLINK	R/W	X	CMPC リンク・ビット 以下のビット選択によって選択された ePWM モジュールの CMPC レジスタに書き込むと、現在の ePWM モジュールの CMPC レジスタにも同時に書き込まれます。 0000 : ePWM1 0001 : ePWM2 ... 最後の ePWM インスタンスまで。その他はすべて予約済みです。 リセットの種類 : SYSRSn
11-8	CMPBLINK	R/W	X	CMPB_CMPBHR リンク・ビット 以下のビット選択によって選択された ePWM モジュールの CMPB_CMPBHR レジスタに書き込むと、現在の ePWM モジュールの CMPB_CMPBHR レジスタにも同時に書き込まれます。 0000 : ePWM1 0001 : ePWM2 ... 最後の ePWM インスタンスまで。その他はすべて予約済みです。 リセットの種類 : SYSRSn

表 1-53. EPWMXLINK レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
7-4	CMPALINK	R/W	X	CMPA_CMPAHR リンク・ビット 以下のビット選択によって選択された ePWM モジュールの CMPA_CMPAHR レジスタに書き込むと、現在の ePWM モジュール の CMPA_CMPAHR レジスタにも同時に書き込まれます。 0000 : ePWM1 0001 : ePWM2 ... 最後の ePWM インスタンスまで。その他はすべて予約済みです。 リセットの種類 : SYSRSn
3-0	TBPRDLINK	R/W	X	TBPRD_TBPRDHR リンク・ビット 以下のビット選択によって選択された ePWM モジュールの TBPRD : TBPRDHR レジスタに書き込むと、現在の ePWM モジュー ールの TBPRD_TBPRDHR レジスタにも同時に書き込まれます。 0000 : ePWM1 0001 : ePWM2 ... 最後の ePWM インスタンスまで。その他はすべて予約済みです。 リセットの種類 : SYSRSn

1.17.2.30 EPWMREV レジスタ (オフセット = 3Eh) [リセット = X]

図 1-122 に、EPWMREV を示し、表 1-54 に、その説明を示します。

概略表に戻ります。

EPWM リビジョン・レジスタ

図 1-122. EPWMREV レジスタ

15	14	13	12	11	10	9	8
TYPE							
R-5h							
7	6	5	4	3	2	1	0
REV							
R-X							

表 1-54. EPWMREV レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	TYPE	R	5h	EPWM タイプ・ビット：これらのビットは EPWM のタイプを指定します。EPWM の機能が変更された場合、または何らかの機能が追加または削除された場合、これらのビットは変更されます。 リセットの種類：SYSRSn
7-0	REV	R	X	EPWM シリコン・リビジョン・ビット：これらのビットは EPWM のリビジョンを指定します。これらのビットは、何らかのバグ修正が行われた場合に変更されます。 リセットの種類：SYSRSn

1.17.2.31 HRPWMREV レジスタ (オフセット = 3Fh) [リセット = 300h]

図 1-123 に、HRPWMREV を示し、表 1-55 に、その説明を示します。

[概略表](#)に戻ります。

高分解能リビジョン・レジスタ

図 1-123. HRPWMREV レジスタ

15	14	13	12	11	10	9	8
TYPE							
R-3h							
7	6	5	4	3	2	1	0
REV							
R-0h							

表 1-55. HRPWMREV レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	TYPE	R	3h	HRPWM タイプ・ビット：これらのビットは HRPWM のタイプを指定します。HRPWM の機能が変更された場合、または何らかの機能が追加または削除された場合、これらのビットは変更されます。 リセットの種類：SYSRSn
7-0	REV	R	0h	HRPWM シリコン・リビジョン・ビット：これらのビットは HRPWM のリビジョンを指定します。これらのビットは、何らかのバグ修正が行われた場合に変更されます。 リセットの種類：SYSRSn

1.17.2.32 AQCTLA レジスタ (オフセット = 40h) [リセット = 0h]

図 1-124 に、AQCTLA を示し、表 1-56 に、その説明を示します。

概略表に戻ります。

出力 A のアクション・フィルタ制御レジスタ

図 1-124. AQCTLA レジスタ

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-56. AQCTLA レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-10	CBD	R/W	0h	ダウン・カウント時の TBCTR = CMPB でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
9-8	CBU	R/W	0h	アップ・カウント時の TBCTR = CMPB でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
7-6	CAD	R/W	0h	ダウン・カウント時の TBCTR = CMPA でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
5-4	CAU	R/W	0h	アップ・カウント時の TBCTR = CMPA でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn

表 1-56. AQCTLA レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-2	PRD	R/W	0h	TBCTR = TBPRD でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
1-0	ZRO	R/W	0h	TBCTR = 0 でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn

1.17.2.33 AQCTLA2 レジスタ (オフセット = 41h) [リセット = 0h]

図 1-125 に、AQCTLA2 を示し、表 1-57 に、その説明を示します。

概略表に戻ります。

出力 A の追加アクション・フィルタ制御レジスタ

図 1-125. AQCTLA2 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
T2D		T2U		T1D		T1U	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-57. AQCTLA2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-6	T2D	R/W	0h	ダウン・カウント時の T2 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
5-4	T2U	R/W	0h	アップ・カウント時の T2 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
3-2	T1D	R/W	0h	ダウン・カウント時の T1 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
1-0	T1U	R/W	0h	アップ・カウント時の T1 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxA 出力を Low に強制します。 10：セット：EPWMxA 出力を High に強制します。 11：EPWMxA 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn

1.17.2.34 AQCTLB レジスタ (オフセット = 42h) [リセット = 0h]

図 1-126 に、AQCTLB を示し、表 1-58 に、その説明を示します。

概略表に戻ります。

出力 B のアクション・フィルタ制御レジスタ

図 1-126. AQCTLB レジスタ

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-58. AQCTLB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-10	CBD	R/W	0h	ダウン・カウント時の TBCTR = CMPB でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
9-8	CBU	R/W	0h	アップ・カウント時の TBCTR = CMPB でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
7-6	CAD	R/W	0h	ダウン・カウント時の TBCTR = CMPA でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
5-4	CAU	R/W	0h	アップ・カウント時の TBCTR = CMPA でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn

表 1-58. AQCTLB レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-2	PRD	R/W	0h	TBCTR = TBPRD でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
1-0	ZRO	R/W	0h	TBCTR = 0 でのアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn

1.17.2.35 AQCTLB2 レジスタ (オフセット = 43h) [リセット = 0h]

図 1-127 に、AQCTLB2 を示し、表 1-59 に、その説明を示します。

概略表に戻ります。

出力 B の追加アクション・フィルタ制御レジスタ

図 1-127. AQCTLB2 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
T2D		T2U		T1D		T1U	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-59. AQCTLB2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-6	T2D	R/W	0h	ダウン・カウント時の T2 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
5-4	T2U	R/W	0h	アップ・カウント時の T2 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
3-2	T1D	R/W	0h	ダウン・カウント時の T1 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn
1-0	T1U	R/W	0h	アップ・カウント時の T1 でイベントが発生した場合のアクション 注：当然のことながら、カウント・アップ・ダウン・モードでカウンタが 0 と一致すると、その方向は 1 (カウント・アップ) に設定されます。 00：何もしません (アクションは無効)。 01：クリア：EPWMxB 出力を Low に強制します。 10：セット：EPWMxB 出力を High に強制します。 11：EPWMxB 出力をトグル：Low 出力信号は High に強制され、High 信号は Low に強制されます。 リセットの種類：SYSRSn

1.17.2.36 AQSFRFC レジスタ (オフセット = 47h) [リセット = 0h]

図 1-128 に、AQSFRFC を示し、表 1-60 に、その説明を示します。

概略表に戻ります。

アクション・フィルタ・ソフトウェア強制レジスタ

図 1-128. AQSFRFC レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RLDCSF		OTSFB		ACTSFB		OTSFA	ACTSFA
R/W-0h		R-0/W1S-0h		R/W-0h		R-0/W1S-0h	R/W-0h

表 1-60. AQSFRFC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-6	RLDCSF	R/W	0h	シャドウからの AQSFRFC アクティブ・レジスタのリロードの選択肢 00: 時間ベース・カウンタがゼロになるとロードします。 01: 時間ベース・カウンタが周期に等しくなるとロードします。 10: 時間ベース・カウンタがゼロまたは周期に等しくなるとロードします。 11: 即座にロードします (アクティブ・レジスタは CPU によって直接アクセスされ、シャドウ・レジスタからはロードされません)。 リセットの種類: SYSRSn
5	OTSFB	R-0/W1S	0h	出力 B でのワントタイム・ソフトウェア強制イベント 0: 0 (ゼロ) を書き込んでも何もしません。常に 0 が読み出されます。このレジスタへの書き込みが完了すると、このビットは自動的にクリアされます (つまり、強制イベントが開始されます)。これはワンショット強制イベントです。出力 B のその他の後続イベントが、このイベントに優先することがあります。 1: 1 つのソフトウェア強制イベントを開始します。 リセットの種類: SYSRSn
4-3	ACTSFB	R/W	0h	ワントタイム・ソフトウェア強制 B が呼び出された際のアクション 00: 何も起こりません (アクションは無効)。 01: クリア (Low) 10: セット (High) 11: トグル (Low -> High, High -> Low) 注: このアクションは、カウンタの方向 (CNT_dir) によってはクオリファイされません。 リセットの種類: SYSRSn
2	OTSFA	R-0/W1S	0h	出力 A でのワントタイム・ソフトウェア強制イベント 0: 0 (ゼロ) を書き込んでも何もしません。常に 0 が読み出されます。このレジスタへの書き込みが完了すると、このビットは自動的にクリアされます (つまり、強制イベントが開始されます)。これはワンショット強制イベントです。出力 A のその他の後続イベントが、このイベントに優先することがあります。 1: 1 つのソフトウェア強制イベントを開始します。 リセットの種類: SYSRSn
1-0	ACTSFA	R/W	0h	ワントタイム・ソフトウェア強制 A が呼び出された際のアクション 00: 何も起こりません (アクションは無効)。 01: クリア (Low) 10: セット (High) 11: トグル (Low -> High, High -> Low) 注: このアクションは、カウンタの方向 (CNT_dir) によってはクオリファイされません。 リセットの種類: SYSRSn

1.17.2.37 AQCSFRC レジスタ (オフセット = 49h) [リセット = 0h]

図 1-129 に、AQCSFRC を示し、表 1-61 に、その説明を示します。

概略表に戻ります。

アクション・フィルタ連続ソフトウェア強制レジスタ

図 1-129. AQCSFRC レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				CSFB		CSFA	
R-0-0h				R/W-0h		R/W-0h	

表 1-61. AQCSFRC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3-2	CSFB	R/W	0h	出力 B での連続ソフトウェア強制 即時モードでは、次の TBCLK エッジで連続強制が効力を生じます。シャドウ・モードでは、アクティブ・レジスタへのシャドウ・ロードの後の、次の TBCLK エッジで連続強制が効力を生じます。シャドウ・モードを設定するには、AQSFRC[RLDCSF] を使います。 00：ソフトウェア強制は無効化され、何の影響も及ぼしません。 01：出力 B を連続的に Low に強制します。 10：出力 B を連続的に High に強制します。 11：ソフトウェア強制は無効化され、何の影響も及ぼしません。 リセットの種類：SYSRSn
1-0	CSFA	R/W	0h	出力 A での連続ソフトウェア強制 即時モードでは、次の TBCLK エッジで連続強制が効力を生じます。シャドウ・モードでは、アクティブ・レジスタへのシャドウ・ロードの後の、次の TBCLK エッジで連続強制が効力を生じます。 00：ソフトウェア強制は無効化され、何の影響も及ぼしません。 01：出力 A を連続的に Low に強制します。 10：出力 A を連続的に High に強制します。 11：ソフトウェア強制は無効化され、何の影響も及ぼしません。 リセットの種類：SYSRSn

1.17.2.38 DBREDHR レジスタ (オフセット = 50h) [リセット = 0h]

図 1-130 に、DBREDHR を示し、表 1-62 に、その説明を示します。

概略表に戻ります。

デッドバンド・ジェネレータ立ち上がりエッジ遅延高分解能ミラー・レジスタ

図 1-130. DBREDHR レジスタ

15	14	13	12	11	10	9	8
DBREDHR							RESERVED
R/W-0h							R-0h
7	6	5	4	3	2	1	0
DBREDHR_DELAY RESERVED							RESERVED
R-0h R-0h							R-0h

表 1-62. DBREDHR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-9	DBREDHR	R/W	0h	デッドバンド立ち上がりエッジ遅延高分解能ビット リセットの種類: SYSRSn
8	RESERVED	R	0h	予約済み
7-1	DBREDHR_DELAY	R	0h	これらの 7 ビットには OTTO 計算の結果が含まれています (自動変換が有効化されている場合)。 リセットの種類: SYSRSn
7-1	RESERVED	R	0h	予約済み
0	RESERVED	R	0h	予約済み

1.17.2.39 DBRED レジスタ (オフセット = 51h) [リセット = 0h]

図 1-131 に、DBRED を示し、表 1-63 に、その説明を示します。

[概略表](#)に戻ります。

デッドバンド・ジェネレータ立ち上がりエッジ遅延高分解能ミラー・レジスタ

図 1-131. DBRED レジスタ

15	14	13	12	11	10	9	8
RESERVED				DBRED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DBRED							
R/W-0h							

表 1-63. DBRED レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	RESERVED	R	0h	予約済み
13-0	DBRED	R/W	0h	立ち上がりエッジ遅延値 リセットの種類: SYSRSn

1.17.2.40 DBFEDHR レジスタ (オフセット = 52h) [リセット = 0h]

図 1-132 に、DBFEDHR を示し、表 1-64 に、その説明を示します。

概略表に戻ります。

デッドバンド・ジェネレータ立ち下がりエッジ遅延高分解能レジスタ

図 1-132. DBFEDHR レジスタ

15	14	13	12	11	10	9	8
DBFEDHR							RESERVED
R/W-0h							R-0h
7	6	5	4	3	2	1	0
DBFEDHR_DELAY RESERVED							RESERVED
R-0h R-0h							R-0h

表 1-64. DBFEDHR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-9	DBFEDHR	R/W	0h	デッドバンド立ち下がりエッジ遅延高分解能ビット リセットの種類: SYSRSn
8	RESERVED	R	0h	予約済み
7-1	DBFEDHR_DELAY	R	0h	これらの 7 ビットには OTTO 計算の結果が含まれています (自動変換が有効化されている場合)。 リセットの種類: SYSRSn
7-1	RESERVED	R	0h	予約済み
0	RESERVED	R	0h	予約済み

1.17.2.41 DBFED レジスタ (オフセット = 53h) [リセット = 0h]

図 1-133 に、DBFED を示し、表 1-65 に、その説明を示します。

[概略表](#)に戻ります。

デッドバンド・ジェネレータ立ち下がりエッジ遅延カウント・レジスタ

図 1-133. DBFED レジスタ

15	14	13	12	11	10	9	8
RESERVED				DBFED			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DBFED							
R/W-0h							

表 1-65. DBFED レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	RESERVED	R	0h	予約済み
13-0	DBFED	R/W	0h	立ち下がりエッジ遅延カウント 14 ビット・カウンタ リセットの種類：SYSRSn

1.17.2.42 TBPHS レジスタ (オフセット = 60h) [リセット = 0h]

図 1-134 に、TBPHS を示し、表 1-66 に、その説明を示します。

概略表に戻ります。

時間ベース位相 High

図 1-134. TBPHS レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS																TBPHSHR															
R/W-0h																R/W-0h															

表 1-66. TBPHS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	TBPHS	R/W	0h	位相オフセット・レジスタ これらのビットは、同期入力信号を供給している時間ベースに対する、選択された ePWM の時間ベース・カウンタの位相を設定します。 - TBCTL[PHSEN] = 0 の場合、同期イベントは無視され、時間ベース・カウンタには位相がロードされません。 - TBCTL[PHSEN] = 1 の場合、同期イベントが発生すると、時間ベース・カウンタ (TBCTR) に位相 (TBPHS) がロードされます。この同期イベントは、入力同期信号 (EPWMxSYNCl) またはソフトウェアによる強制同期によって開始されます。 リセットの種類: SYSRSn
15-0	TBPHSHR	R/W	0h	位相オフセット (高分解能) レジスタ。 TBPHSHR は使用できません。TBPHSHR の機能を再現するには、代わりに TRREM (HRPWM 剰余レジスタ) を使用する必要があります。 このレジスタの下位 8 ビットは無視されます。書き込みは無視され、読み出しにはゼロが返されます。 リセットの種類: SYSRSn

1.17.2.43 TBPRDHR レジスタ (オフセット = 62h) [リセット = 0h]

図 1-135 に、TBPRDHR を示し、表 1-67 に、その説明を示します。

[概略表](#)に戻ります。

時間ベース周期高分解能レジスタ

図 1-135. TBPRDHR レジスタ

15	14	13	12	11	10	9	8
TBPRDHR							
R/W-0h							
7	6	5	4	3	2	1	0
TBPRDHR							
R/W-0h							

表 1-67. TBPRDHR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	TBPRDHR	R/W	0h	周期高分解能ビット 上位 8 ビットには周期値の高分解能部分が含まれます。TBPRDHR レジスタは TBCTL[PRDL] ビットの影響を受けません。このレジスタからの読み出しには常にシャドウ・レジスタが反映されます。同様に、書き込みもシャドウ・レジスタに対して行われます。TBPRDHR レジスタは、高分解能周期機能が有効化されている場合にのみ使用されます。このレジスタは、高分解能周期制御をサポートしている ePWM モジュールでのみ利用できます。このレジスタの下位 8 ビットは無視されます。書き込みは無視され、読み出しにはゼロが返されます。リセットの種類 SYSRSn

1.17.2.44 TBPRD レジスタ (オフセット = 63h) [リセット = 0h]

図 1-136 に、TBPRD を示し、表 1-68 に、その説明を示します。

概略表に戻ります。

時間ベース周期レジスタ

図 1-136. TBPRD レジスタ

15	14	13	12	11	10	9	8
TBPRD							
R/W-0h							
7	6	5	4	3	2	1	0
TBPRD							
R/W-0h							

表 1-68. TBPRD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	TBPRD	R/W	0h	時間ベース周期レジスタ これらのビットは時間ベース・カウンタの周期を決定します。このレジスタにより、PWM 周波数が設定されます。このレジスタのシャドウイングは、TBCTL[PRDL] ビットによって有効化および無効化されます。デフォルトでは、このレジスタはシャドウイングされます。 - TBCTL[PRDL] = 0 の場合、シャドウイングは有効化され、すべての書き込みまたは読み出しは自動的にシャドウ・レジスタに対して行われます。この場合、時間ベース・カウンタがゼロになると、アクティブ・レジスタはシャドウ・レジスタからロードされます。 - TBCTL[PRDL] = 1 の場合、シャドウイングは無効化され、すべての書き込みまたは読み出しはアクティブ・レジスタ (つまりハードウェアをアクティブに制御しているレジスタ) に対して直接行われます。 - アクティブ・レジスタとシャドウ・レジスタは、同じメモリ・マップ・アドレスを共有します。 リセットの種類 : SYSRSn

1.17.2.45 TBPRDHRB レジスタ (オフセット = 64h) [リセット = 0h]

図 1-137 に、TBPRDHRB を示し、表 1-69 に、その説明を示します。

[概略表](#)に戻ります。

EPWMxB の計算結果

図 1-137. TBPRDHRB レジスタ

15	14	13	12	11	10	9	8
TBPRDHRB							
R/W-0h							
7	6	5	4	3	2	1	0
TBPRDHRB_DELAY RESERVED							
R/W-0h R/W-0h							

表 1-69. TBPRDHRB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	TBPRDHRB	R/W	0h	TBPRD 高分解能計算 (2) EPWMxB の HRPWM の式の結果 リセットの種類: SYSRSn
7-0	TBPRDHRB_DELAY	R/W	0h	TBPRDHRB 遅延 リセットの種類: SYSRSn
7-0	RESERVED	R/W	0h	予約済み

1.17.2.46 CMPA レジスタ (オフセット = 6Ah) [リセット = 0h]

図 1-138 に、CMPA を示し、表 1-70 に、その説明を示します。

概略表に戻ります。

カウンタ比較 A レジスタ

図 1-138. CMPA レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA																CMPAHR															
R/W-0h																R/W-0h															

表 1-70. CMPA レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	CMPA	R/W	0h	比較 A レジスタ アクティブ CMPA レジスタの値は時間ベース・カウンタ (TBCTR) と絶えず比較されます。値が等しくなると、カウンタ比較モジュールは「時間ベース・カウンタがカウンタ比較 A と一致」イベントを生成します。このイベントはアクション・フィルタに転送され、そこでクオリファイされ、1 つ以上のアクションに変換されます。これらのアクションは、AQCTLA および AQCTLB レジスタの設定に応じて、EPWMxA と EPWMxB のどちらかの出力に適用されます。AQCTLA および AQCTLB レジスタで定義できるアクションは、次のとおりです。 - 何もしない イベントは無視されます。 - クリア：EPWMxA および / または EPWMxB 信号を Low にプルします。 - セット：EPWMxA および / または EPWMxB 信号を High にプルします。 - EPWMxA および / または EPWMxB 信号をトグルします。 このレジスタのシャドウイングは、CMPCTL[SHDWAMODE] ビットによって有効化および無効化されます。デフォルトでは、このレジスタはシャドウイングされます。 - CMPCTL[SHDWAMODE] = 0 の場合、シャドウイングは有効化され、すべての書き込みまたは読み出しはシャドウ・レジスタに対して自動的に行われます。この場合、CMPCTL[LOADAMODE] ビット・フィールドによって、シャドウ・レジスタからアクティブ・レジスタへのロードをトリガするイベントの種類が決定されます。 - 書き込みの前に、CMPCTL[SHDWAFULL] ビットを読み出すことで、シャドウ・レジスタが現在完全に満たされているかどうかを確認できます。 - CMPCTL[SHDWAMODE] = 1 の場合、シャドウ・レジスタは無効化され、すべての書き込みまたは読み出しはアクティブ・レジスタ (ハードウェアをアクティブに制御しているレジスタ) に対して直接行われます。 - どちらのモードでも、アクティブ・レジスタとシャドウ・レジスタは、同じメモリ・マップ・アドレスを共有します。 リセットの種類：SYSRSn
15-0	CMPAHR	R/W	0h	比較 A HRPWM 拡張レジスタ 上位 8 ビットにはカウンタ比較 A 値の高分解能部分が含まれます。CMPA : CMPAHR は、1 つの 32 ビット読み出し / 書き込みでアクセスできます。シャドウイングは、CMPA レジスタで説明されているように、CMPCTL[SHDWAMODE] ビットによって有効化および無効化されます。 このレジスタの下位 8 ビットは無視されます。 リセットの種類：SYSRSn

1.17.2.47 CMPB レジスタ (オフセット = 6Ch) [リセット = 0h]

図 1-139 に、CMPB を示し、表 1-71 に、その説明を示します。

概略表に戻ります。

カウンタ比較 B レジスタ

図 1-139. CMPB レジスタ

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB																CMPBHR															
R/W-0h																R/W-0h															

表 1-71. CMPB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	CMPB	R/W	0h	比較 B レジスタ アクティブ CMPB レジスタの値は時間ベース・カウンタ (TBCTR) と絶えず比較されます。値が等しくなると、カウンタ比較モジュールは「時間ベース・カウンタがカウンタ比較 B と一致」イベントを生成します。このイベントはアクション・フィルタに転送され、そこでクオリファイされ、1 つ以上のアクションに変換されます。これらのアクションは、AQCTLA および AQCTLB レジスタの設定に応じて、EPWMxA と EPWMxB のどちらかの出力に適用されます。AQCTLA および AQCTLB レジスタで定義できるアクションは、次のとおりです。 - 何もしない イベントは無視されます。 - クリア：EPWMxA および / または EPWMxB 信号を Low にプルします。 - セット：EPWMxA および / または EPWMxB 信号を High にプルします。 - EPWMxA および / または EPWMxB 信号をトグルします。 このレジスタのシャドウイングは、CMPCTL[SHDWBMODE] ビットによって有効化および無効化されます。デフォルトでは、このレジスタはシャドウイングされます。 - CMPCTL[SHDWBMODE] = 0 の場合、シャドウイングは有効化され、すべての書き込みまたは読み出しはシャドウ・レジスタに対して自動的に行われます。この場合、CMPCTL[LOADBMODE] ビット・フィールドによって、シャドウ・レジスタからアクティブ・レジスタへのロードをトリガするイベントの種類が決定されます。 - 書き込みの前に、CMPCTL[SHDWBFULL] ビットを読み出すことで、シャドウ・レジスタが現在完全に満たされているかどうかを確認できます。 - CMPCTL[SHDWBMODE] = 1 の場合、シャドウ・レジスタは無効化され、すべての書き込みまたは読み出しはアクティブ・レジスタ (ハードウェアをアクティブに制御しているレジスタ) に対して直接行われます。 - どちらのモードでも、アクティブ・レジスタとシャドウ・レジスタは、同じメモリ・マップ・アドレスを共有します。 リセットの種類：SYSRSn
15-0	CMPBHR	R/W	0h	比較 B 高分解能ビット このレジスタの下位 8 ビットは無視されます。 リセットの種類：SYSRSn

1.17.2.48 CMPC レジスタ (オフセット = 6Fh) [リセット = 0h]

図 1-140 に、CMPC を示し、表 1-72 に、その説明を示します。

概略表に戻ります。

カウンタ比較 C レジスタ

リンク機能へのアクセスは、常に 16 ビットである必要があります。

図 1-140. CMPC レジスタ

15	14	13	12	11	10	9	8
CMPC							
R/W-0h							
7	6	5	4	3	2	1	0
CMPC							
R/W-0h							

表 1-72. CMPC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	CMPC	R/W	0h	比較 C レジスタ アクティブ CMPC レジスタの値は時間ベース・カウンタ (TBCTR) と絶えず比較されます。値が等しくなると、カウンタ比較モジュールは「時間ベース・カウンタがカウンタ比較 C と一致」イベントを生成します。 このレジスタのシャドウイングは、CMPCTL2[SHDWCMODE] ビットによって有効化および無効化されます。デフォルトでは、このレジスタはシャドウイングされます。 - CMPCTL2[SHDWCMODE] = 0 の場合、シャドウイングは有効化され、すべての書き込みまたは読み出しはシャドウ・レジスタに対して自動的に行われます。この場合、CMPCTL2[LOADCMODE] ビット・フィールドによって、シャドウ・レジスタからアクティブ・レジスタへのロードをトリガするイベントの種類が決定されます。 - CMPCTL2[SHDWCMODE] = 1 の場合、シャドウ・レジスタは無効化され、すべての書き込みまたは読み出しはアクティブ・レジスタ (ハードウェアをアクティブに制御しているレジスタ) に対して直接行われます。 - どちらのモードでも、アクティブ・レジスタとシャドウ・レジスタは、同じメモリ・マップ・アドレスを共有します。 リセットの種類 : SYSRSn

1.17.2.49 CMPD レジスタ (オフセット = 71h) [リセット = 0h]

図 1-141 に、CMPD を示し、表 1-73 に、その説明を示します。

概略表に戻ります。

カウンタ比較 D レジスタ

リンク機能へのアクセスは、常に 16 ビットである必要があります。

図 1-141. CMPD レジスタ

15	14	13	12	11	10	9	8
CMPD							
R/W-0h							
7	6	5	4	3	2	1	0
CMPD							
R/W-0h							

表 1-73. CMPD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	CMPD	R/W	0h	比較 D レジスタ アクティブ CMPD レジスタの値は時間ベース・カウンタ (TBCTR) と絶えず比較されます。値が等しくなると、カウンタ比較モジュールは「時間ベース・カウンタがカウンタ比較 D と一致」イベントを生成します。 このレジスタのシャドウイングは、CMPCTL2[SHDWDMODE] ビットによって有効化および無効化されます。デフォルトでは、このレジスタはシャドウイングされます。 - CMPCTL2[SHDWDMODE] = 0 の場合、シャドウイングは有効化され、すべての書き込みまたは読み出しはシャドウ・レジスタに対して自動的に行われます。この場合、CMPCTL2[LOADDMODE] ビット・フィールドによって、シャドウ・レジスタからアクティブ・レジスタへのロードをトリガするイベントの種類が決定されます。 - CMPCTL2[SHDWDMODE] = 1 の場合、シャドウ・レジスタは無効化され、すべての書き込みまたは読み出しはアクティブ・レジスタ (ハードウェアをアクティブに制御しているレジスタ) に対して直接行われます。 - どちらのモードでも、アクティブ・レジスタとシャドウ・レジスタは、同じメモリ・マップ・アドレスを共有します。 リセットの種類 : SYSRSn

1.17.2.50 GLDCTL2 レジスタ (オフセット = 74h) [リセット = 0h]

図 1-142 に、GLDCTL2 を示し、表 1-74 に、その説明を示します。

概略表に戻ります。

グローバル PWM ロード制御レジスタ 2

図 1-142. GLDCTL2 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						GFRCLD	OSHTLD
R-0-0h						R-0/W1S-0h	R-0/W1S-0h

表 1-74. GLDCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-2	RESERVED	R-0	0h	予約済み
1	GFRCLD	R-0/W1S	0h	ワンショット・モードでの強制ロード・イベント 0: 0 の書き込みは無視されます。常に 0 が読み出されます。 1: イベント・プリスケール・カウンタの入力に 1 つのロード・イベントを強制します。このビットは、グローバル・ロード・モードでのイベントのテスト / ソフトウェア強制ロードに使うことを意図しています。 リセットの種類: SYSRSn
0	OSHTLD	R-0/W1S	0h	ワンショット・モードでのリロード・イベントの有効化 0: 0 の書き込みは無視されます。常に 0 が読み出されます。 1: ワンショット・ラッチ条件を有効化します。選択されたロード・ストローブが発生すると、アクティブ・リロードに対して 1 回シャドウイングが行われ、ラッチがクリアされます。そのため、このビットに 1 を書き込むと、1 つのロード・ストローブ・イベントが通過し、それ以上のストローブ・イベントは阻止されます。 リセットの種類: SYSRSn

1.17.2.51 SWVDELVAL レジスタ (オフセット = 77h) [リセット = 0h]

図 1-143 に、SWVDELVAL を示し、表 1-75 に、その説明を示します。

概略表に戻ります。

ソフトウェア・バレー・モード遅延レジスタ

図 1-143. SWVDELVAL レジスタ

15	14	13	12	11	10	9	8
SWVDELVAL							
R/W-0h							
7	6	5	4	3	2	1	0
SWVDELVAL							
R/W-0h							

表 1-75. SWVDELVAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	SWVDELVAL	R/W	0h	ソフトウェア・バレー遅延値レジスタ このレジスタは、VCAPCTL[VDELAYDIV] ビットで定義されたハードウェア計算遅延 HWDELAYVAL のオフセット値を定義するために使用することもできます。 リセットの種類 : SYSRSn

1.17.2.52 TZSEL レジスタ (オフセット = 80h) [リセット = 0h]

図 1-144 に、TZSEL を示し、表 1-76 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン選択レジスタ

図 1-144. TZSEL レジスタ

15		14		13		12		11		10		9		8	
DCBEVT1	DCAEVT1	OSHT6	OSHT5	OSHT4	OSHT3	OSHT2	OSHT1								
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h								
7		6		5		4		3		2		1		0	
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1								
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h								

表 1-76. TZSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	DCBEVT1	R/W	0h	デジタル比較出力 B イベント 1 の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして DCBEVT1 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして DCBEVT1 を有効化します。 リセットの種類: SYSRSn
14	DCAEVT1	R/W	0h	デジタル比較出力 A イベント 1 の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして DCAEVT1 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして DCAEVT1 を有効化します。 リセットの種類: SYSRSn
13	OSHT6	R/W	0h	トリップ・ゾーン 6 (TZ6) の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして TZ6 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして TZ6 を有効化します。 リセットの種類: SYSRSn
12	OSHT5	R/W	0h	トリップ・ゾーン 5 (TZ5) の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして TZ5 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして TZ5 を有効化します。 リセットの種類: SYSRSn
11	OSHT4	R/W	0h	トリップ・ゾーン 4 (TZ4) の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして TZ4 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして TZ4 を有効化します。 リセットの種類: SYSRSn
10	OSHT3	R/W	0h	トリップ・ゾーン 3 (TZ3) の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして TZ3 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして TZ3 を有効化します。 リセットの種類: SYSRSn
9	OSHT2	R/W	0h	トリップ・ゾーン 2 (TZ2) の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして TZ2 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして TZ2 を有効化します。 リセットの種類: SYSRSn

表 1-76. TZSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
8	OSHT1	R/W	0h	トリップ・ゾーン 1 (TZ1) の選択 0: この ePWM モジュールのワンショット・トリップ・ソースとして TZ1 を無効化します。 1: この ePWM モジュールのワンショット・トリップ・ソースとして TZ1 を有効化します。 リセットの種類: SYSRSn
7	DCBEVT2	R/W	0h	デジタル比較出力 B イベント 2 の選択 0: この ePWM モジュールの CBC トリップ・ソースとして DCBEVT2 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして DCBEVT2 を有効化します。 リセットの種類: SYSRSn
6	DCAEVT2	R/W	0h	デジタル比較出力 A イベント 2 の選択 0: この ePWM モジュールの CBC トリップ・ソースとして DCAEVT2 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして DCAEVT2 を有効化します。 リセットの種類: SYSRSn
5	CBC6	R/W	0h	トリップ・ゾーン 6 (TZ6) の選択 0: この ePWM モジュールの CBC トリップ・ソースとして TZ6 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして TZ6 を有効化します。 リセットの種類: SYSRSn
4	CBC5	R/W	0h	トリップ・ゾーン 5 (TZ5) の選択 0: この ePWM モジュールの CBC トリップ・ソースとして TZ5 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして TZ5 を有効化します。 リセットの種類: SYSRSn
3	CBC4	R/W	0h	トリップ・ゾーン 4 (TZ4) の選択 0: この ePWM モジュールの CBC トリップ・ソースとして TZ4 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして TZ4 を有効化します。 リセットの種類: SYSRSn
2	CBC3	R/W	0h	トリップ・ゾーン 3 (TZ3) の選択 0: この ePWM モジュールの CBC トリップ・ソースとして TZ3 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして TZ3 を有効化します。 リセットの種類: SYSRSn
1	CBC2	R/W	0h	トリップ・ゾーン 2 (TZ2) の選択 0: この ePWM モジュールの CBC トリップ・ソースとして TZ2 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして TZ2 を有効化します。 リセットの種類: SYSRSn
0	CBC1	R/W	0h	トリップ・ゾーン 1 (TZ1) の選択 0: この ePWM モジュールの CBC トリップ・ソースとして TZ1 を無効化します。 1: この ePWM モジュールの CBC トリップ・ソースとして TZ1 を有効化します。 リセットの種類: SYSRSn

1.17.2.53 TZDCSEL レジスタ (オフセット = 82h) [リセット = 0h]

図 1-145 に、TZDCSEL を示し、表 1-77 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン・デジタル・コンパレータ選択レジスタ

図 1-145. TZDCSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2			DCBEVT1
R-0-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCBEVT1		DCAEVT2			DCAEVT1		
R/W-0h		R/W-0h			R/W-0h		

表 1-77. TZDCSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-9	DCBEVT2	R/W	0h	デジタル比較出力 B イベント 2 の選択 000 : イベントは無効 001 : DCBH = Low、DCBL = ドントケア 010 : DCBH = High、DCBL = ドントケア 011 : DCBL = Low、DCBH = ドントケア 100 : DCBL = High、DCBH = ドントケア 101 : DCBL = High、DCBH = Low 110 : 予約済み 111 : 予約済み リセットの種類 : SYSRSn
8-6	DCBEVT1	R/W	0h	デジタル比較出力 B イベント 1 の選択 000 : イベントは無効 001 : DCBH = Low、DCBL = ドントケア 010 : DCBH = High、DCBL = ドントケア 011 : DCBL = Low、DCBH = ドントケア 100 : DCBL = High、DCBH = ドントケア 101 : DCBL = High、DCBH = Low 110 : 予約済み 111 : 予約済み リセットの種類 : SYSRSn
5-3	DCAEVT2	R/W	0h	デジタル比較出力 A イベント 2 の選択 000 : イベントは無効 001 : DCAH = Low、DCAL = ドントケア 010 : DCAH = High、DCAL = ドントケア 011 : DCAL = Low、DCAH = ドントケア 100 : DCAL = High、DCAH = ドントケア 101 : DCAL = High、DCAH = Low 110 : 予約済み 111 : 予約済み リセットの種類 : SYSRSn
2-0	DCAEVT1	R/W	0h	デジタル比較出力 A イベント 1 の選択 000 : イベントは無効 001 : DCAH = Low、DCAL = ドントケア 010 : DCAH = High、DCAL = ドントケア 011 : DCAL = Low、DCAH = ドントケア 100 : DCAL = High、DCAH = ドントケア 101 : DCAL = High、DCAH = Low 110 : 予約済み 111 : 予約済み リセットの種類 : SYSRSn

1.17.2.54 TZCTL レジスタ (オフセット = 84h) [リセット = 0h]

図 1-146 に、TZCTL を示し、表 1-78 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン制御レジスタ

図 1-146. TZCTL レジスタ

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2		DCBEVT1	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
DCAEVT2		DCAEVT1		TZB		TZA	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-78. TZCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-10	DCBEVT2	R/W	0h	EPWMxB でのデジタル比較出力 B イベント 2 のアクション 00: ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 01: EPWMxB を High 状態に強制します。 10: EPWMxB を Low 状態に強制します。 11: 何もしません (トリップ・アクションは無効)。 リセットの種類: SYSRSn
9-8	DCBEVT1	R/W	0h	EPWMxB でのデジタル比較出力 B イベント 1 のアクション 00: ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 01: EPWMxB を High 状態に強制します。 10: EPWMxB を Low 状態に強制します。 11: 何もしません (トリップ・アクションは無効)。 リセットの種類: SYSRSn
7-6	DCAEVT2	R/W	0h	EPWMxA でのデジタル比較出力 A イベント 2 のアクション 00: ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 01: EPWMxA を High 状態に強制します。 10: EPWMxA を Low 状態に強制します。 11: 何もしません (トリップ・アクションは無効)。 リセットの種類: SYSRSn
5-4	DCAEVT1	R/W	0h	EPWMxA でのデジタル比較出力 A イベント 1 のアクション 00: ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 01: EPWMxA を High 状態に強制します。 10: EPWMxA を Low 状態に強制します。 11: 何もしません (トリップ・アクションは無効)。 リセットの種類: SYSRSn
3-2	TZB	R/W	0h	EPWMxB での、TZ1~TZ6、DCAEVT1/2、DCBEVT1/2 のトリップ・アクション トリップ・イベントが発生した場合、出力 EPWMxB で次のアクションが実行されます。イベントを発生させることができるトリップ・ゾーン・ピンは、TZSEL レジスタで設定されます。 00: ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 01: EPWMxB を High 状態に強制します。 10: EPWMxB を Low 状態に強制します。 11: 何もしません (EPWMxB では何も実行されません)。 リセットの種類: SYSRSn

表 1-78. TZCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1-0	TZA	R/W	0h	EPWMxA での、TZ1~TZ6、DCAEVT1/2、DCBEVT1/2 のトリップ・アクション トリップ・イベントが発生した場合、出力 EPWMxA で次のアクションが実行されます。イベントを発生させることができるトリップ・ゾーン・ピンは、TZSEL レジスタで設定されます。 00：ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 01：EPWMxA を High 状態に強制します。 10：EPWMxA を Low 状態に強制します。 11：何もありません (EPWMxA では何も実行されません)。 リセットの種類：SYSRStn

1.17.2.55 TZCTL2 レジスタ (オフセット = 85h) [リセット = 0h]

図 1-147 に、TZCTL2 を示し、表 1-79 に、その説明を示します。

概略表に戻ります。

追加のトリップ・ゾーン制御レジスタ

図 1-147. TZCTL2 レジスタ

15		14		13		12		11		10		9		8	
ETZE		RESERVED						TZBD				TZBU			
R/W-0h		R-0-0h						R/W-0h				R/W-0h			
7		6		5		4		3		2		1		0	
TZBU		TZAD						TZAU							
R/W-0h		R/W-0h						R/W-0h							

表 1-79. TZCTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	ETZE	R/W	0h	TZCTL2 イネーブル 0 : TZCTL からトリップ・アクションを使います (従来の ePWM と互換)。 1 : TZCTL2、TZCTLDCA、TZCTLDCA で設定されたトリップ・アクションを使います。TZCTL での設定は無視されます。 リセットの種類 : SYSRSn
14-12	RESERVED	R-0	0h	予約済み
11-9	TZBD	R/W	0h	カウント・ダウン方向の場合の EPWMxB での TZ1~TZ6 のトリップ・アクション 000 : ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxB = High 状態)。 010 : Low 状態に強制します (EPWMxB = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn
8-6	TZBU	R/W	0h	カウント・アップ方向の場合の EPWMxB での TZ1~TZ6、DCAEVT1/2、DCBEVT1/2 のトリップ・アクション 000 : ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxB = High 状態)。 010 : Low 状態に強制します (EPWMxB = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn
5-3	TZAD	R/W	0h	カウント・ダウン方向の場合の EPWMxA での TZ1~TZ6、DCAEVT1/2、DCBEVT1/2 のトリップ・アクション 000 : ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxA = High 状態)。 010 : Low 状態に強制します (EPWMxA = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn

表 1-79. TZCTL2 レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
2-0	TZAU	R/W	0h	カウント・アップ方向の場合の EPWMxA での TZ1 ~ TZ6、 DCAEVT1/2、DCBEVT1/2 のトリップ・アクション 000 : ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状 態) 001 : High 状態に強制します (EPWMxA = High 状態)。 010 : Low 状態に強制します (EPWMxA = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn

1.17.2.56 TZCTLDCA レジスタ (オフセット = 86h) [リセット = 0h]

図 1-148 に、TZCTLDCA を示し、表 1-80 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン制御レジスタ・デジタル比較 A

図 1-148. TZCTLDCA レジスタ

15	14	13	12	11	10	9	8
RESERVED				DCAEVT2D			DCAEVT2U
R-0-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCAEVT2U		DCAEVT1D			DCAEVT1U		
R/W-0h		R/W-0h			R/W-0h		

表 1-80. TZCTLDCA レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-9	DCAEVT2D	R/W	0h	カウント・ダウン方向の場合の EPWMxA でのデジタル比較出力 A イベント 2 のアクション 000 : ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxA = High 状態)。 010 : Low 状態に強制します (EPWMxA = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn
8-6	DCAEVT2U	R/W	0h	カウント・アップ方向の場合の EPWMxA でのデジタル比較出力 A イベント 2 のアクション 000 : ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxA = High 状態)。 010 : Low 状態に強制します (EPWMxA = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn
5-3	DCAEVT1D	R/W	0h	カウント・ダウン方向の場合の EPWMxA でのデジタル比較出力 A イベント 1 のアクション 000 : ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxA = High 状態)。 010 : Low 状態に強制します (EPWMxA = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn

表 1-80. TZCTLDCR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
2-0	DCAEVT1U	R/W	0h	カウント・アップ方向の場合の EPWMxA でのデジタル比較出力 A イベント 1 のアクション 000 : ハイ・インピーダンス (EPWMxA = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxA = High 状態)。 010 : Low 状態に強制します (EPWMxA = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn

1.17.2.57 TZCTLDCB レジスタ (オフセット = 87h) [リセット = 0h]

図 1-149 に、TZCTLDCB を示し、表 1-81 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン制御レジスタ・デジタル比較 B

図 1-149. TZCTLDCB レジスタ

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2D			DCBEVT2U
R-0-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCBEVT2U		DCBEVT1D			DCBEVT1U		
R/W-0h		R/W-0h			R/W-0h		

表 1-81. TZCTLDCB レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R-0	0h	予約済み
11-9	DCBEVT2D	R/W	0h	カウント・ダウン方向の場合の EPWMxB でのデジタル比較出力 B イベント 2 のアクション 000 : ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxB = High 状態)。 010 : Low 状態に強制します (EPWMxB = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn
8-6	DCBEVT2U	R/W	0h	カウント・アップ方向の場合の EPWMxB でのデジタル比較出力 B イベント 2 のアクション 000 : ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxB = High 状態)。 010 : Low 状態に強制します (EPWMxB = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn
5-3	DCBEVT1D	R/W	0h	カウント・ダウン方向の場合の EPWMxB でのデジタル比較出力 B イベント 1 のアクション 000 : ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxB = High 状態)。 010 : Low 状態に強制します (EPWMxB = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn

表 1-81. TZCTLDCB レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
2-0	DCBEVT1U	R/W	0h	カウント・アップ方向の場合の EPWMxB でのデジタル比較出力 B イベント 1 のアクション 000 : ハイ・インピーダンス (EPWMxB = ハイ・インピーダンス状態) 001 : High 状態に強制します (EPWMxB = High 状態)。 010 : Low 状態に強制します (EPWMxB = Low 状態)。 011 : トグル (Low -> High、High -> Low) 100 : 予約済み 101 : 予約済み 110 : 予約済み 111 : 何もしません (トリップ・アクションは無効)。 リセットの種類 : SYSRSn

1.17.2.58 TZEINT レジスタ (オフセット = 8Dh) [リセット = 0h]

図 1-150 に、TZEINT を示し、表 1-82 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン・イネーブル割り込みレジスタ

図 1-150. TZEINT レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0-0h

表 1-82. TZEINT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-7	RESERVED	R-0	0h	予約済み
6	DCBEVT2	R/W	0h	デジタル比較出力 B イベント 2 割り込みイネーブル 0: 無効化 1: 有効化 リセットの種類: SYSRSn
5	DCBEVT1	R/W	0h	デジタル比較出力 B イベント 1 割り込みイネーブル 0: 無効化 1: 有効化 リセットの種類: SYSRSn
4	DCAEVT2	R/W	0h	デジタル比較出力 A イベント 2 割り込みイネーブル 0: 無効化 1: 有効化 リセットの種類: SYSRSn
3	DCAEVT1	R/W	0h	デジタル比較出力 A イベント 1 割り込みイネーブル 0: 無効化 1: 有効化 リセットの種類: SYSRSn
2	OST	R/W	0h	トリップ・ゾーン・ワンショット割り込みイネーブル 0: ワンショット割り込み生成を無効化します。 1: 割り込み生成を有効化します。 ワンショット・トリップ・イベントにより、EPWMx_TZINT PIE 割り込みが生成されます。 リセットの種類: SYSRSn
1	CBC	R/W	0h	トリップ・ゾーンのサイクル・バイ・サイクル割り込みイネーブル 0: サイクル・バイ・サイクル割り込み生成を無効化します。 1: 割り込み生成を有効化します。 サイクルごとのトリップ・イベントにより、EPWMx_TZINT PIE 割り込みが生成されます。 リセットの種類: SYSRSn
0	RESERVED	R-0	0h	予約済み

1.17.2.59 TZFLG レジスタ (オフセット = 93h) [リセット = 0h]

図 1-151 に、TZFLG を示し、表 1-83 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン・フラグ・レジスタ

図 1-151. TZFLG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-83. TZFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-7	RESERVED	R-0	0h	予約済み
6	DCBEVT2	R	0h	デジタル比較出力 B イベント 2 のラッチ・ステータス・フラグ 0: DCBEVT2 でトリップ・イベントが発生していないことを示します。 1: DCBEVT2 のために設定されたイベントに対してトリップ・イベントが発生したことを示します。 リセットの種類 SYSRSn
5	DCBEVT1	R	0h	デジタル比較出力 B イベント 1 のラッチ・ステータス・フラグ 0: DCBEVT1 でトリップ・イベントが発生していないことを示します。 1: DCBEVT1 のために設定されたイベントに対してトリップ・イベントが発生したことを示します。 リセットの種類 SYSRSn
4	DCAEVT2	R	0h	デジタル比較出力 A イベント 2 のラッチ・ステータス・フラグ 0: DCAEVT2 でトリップ・イベントが発生していないことを示します。 1: DCAEVT2 のために設定されたイベントに対してトリップ・イベントが発生したことを示します。 リセットの種類 SYSRSn
3	DCAEVT1	R	0h	デジタル比較出力 A イベント 1 のラッチ・ステータス・フラグ 0: DCAEVT1 でトリップ・イベントが発生していないことを示します。 1: DCAEVT1 のために設定されたイベントに対してトリップ・イベントが発生したことを示します。 リセットの種類 SYSRSn
2	OST	R	0h	A のワンショット・トリップ・イベントのラッチ・ステータス・フラグ 0: ワンショット・トリップ・イベントは発生していません。 1: ワンショット・トリップ・ソースとして選択されたピンでトリップ・イベントが発生したことを示します。 TZCLR レジスタに適切な値を書き込むことで、このビットはクリアされます。 リセットの種類: SYSRSn

表 1-83. TZFLG レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	CBC	R	0h	サイクル・バイ・サイクル・トリップ・イベントのラッチ・ステータス・フラグ 0：サイクル・バイ・サイクル・トリップ・イベントは発生していません。 1：サイクル・バイ・サイクル・トリップ・ソースとして選択された信号でトリップ・イベントが発生したことを示します。 TZFLG[CBC] ビットは、ユーザーが個別にクリアするまでセットされたまま維持されます。CBC ビットがクリアされた際にサイクル・バイ・サイクル・トリップ・イベントが引き続き存在する場合、CBC は再び即座にセットされます。トリップ条件がもはや存在しなくなった場合、ePWM 時間ベース・カウンタがゼロ (TBCTR = 0x00) になると、その信号で指定された条件は自動的にクリアされます。サイクル内のどの場所で CBC フラグがクリアされたとしても、信号の条件は、TBCTR = 0x00 の場合にのみクリアされます。TZCLR レジスタに適切な値を書き込むことで、このビットはクリアされます。 リセットの種類：SYSRSn
0	INT	R	0h	ラッチ・トリップ割り込みステータス・フラグ 0：割り込みが生成されていないことを示します。 1：トリップ条件のせいで EPWMx_TZINT PIE 割り込みが生成されたことを示します。 このフラグがクリアされるまで、さらなる EPWMx_TZINT PIE 割り込みは生成されません。CBC と OST のどちらかがセットされているときに割り込みフラグがクリアされると、さらなる割り込みパルスが生成されます。すべてのフラグ・ビットをクリアすると、さらなる割り込みは禁止されます。TZCLR レジスタに適切な値を書き込むことで、このビットはクリアされます。 リセットの種類：SYSRSn

1.17.2.60 TZCBCFLG レジスタ (オフセット = 94h) [リセット = 0h]

図 1-152 に、TZCBCFLG を示し、表 1-84 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン CBC フラグ・レジスタ

図 1-152. TZCBCFLG レジスタ

15								14								13								12								11								10								9								8							
RESERVED																																																															
R-0-0h																																																															
7								6								5								4								3								2								1								0							
DCBEVT2								DCAEVT2								CBC6								CBC5								CBC4								CBC3								CBC2								CBC1							
R-0h								R-0h								R-0h								R-0h								R-0h								R-0h								R-0h								R-0h							

表 1-84. TZCBCFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7	DCBEVT2	R	0h	デジタル比較 B 出カイベント 2 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、DCBEVT2 でトリップが発生していないことを示します。 1: 1 の読み出しは、DCBEVT2 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
6	DCAEVT2	R	0h	デジタル比較 A 出カイベント 2 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、DCAEVT2 でトリップが発生していないことを示します。 1: 1 の読み出しは、DCAEVT2 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
5	CBC6	R	0h	CBC6 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、CBC6 でトリップが発生していないことを示します。 1: 1 の読み出しは、CBC6 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
4	CBC5	R	0h	CBC5 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、CBC5 でトリップが発生していないことを示します。 1: 1 の読み出しは、CBC5 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
3	CBC4	R	0h	CBC4 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、CBC4 でトリップが発生していないことを示します。 1: 1 の読み出しは、CBC4 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
2	CBC3	R	0h	CBC3 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、CBC3 でトリップが発生していないことを示します。 1: 1 の読み出しは、CBC3 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn

表 1-84. TZCBCFLG レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	CBC2	R	0h	CBC2トリップ・ラッチのラッチ・ステータス・フラグ 0:0の読み出しは、CBC2でトリップが発生していないことを示します。 1:1の読み出しは、CBC2の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
0	CBC1	R	0h	CBC1トリップ・ラッチのラッチ・ステータス・フラグ 0:0の読み出しは、CBC1でトリップが発生していないことを示します。 1:1の読み出しは、CBC1の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn

1.17.2.61 TZOSTFLG レジスタ (オフセット = 95h) [リセット = 0h]

図 1-153 に、TZOSTFLG を示し、表 1-85 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン OST フラグ・レジスタ

図 1-153. TZOSTFLG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
DCBEVT1	DCAEVT1	OST6	OST5	OST4	OST3	OST2	OST1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-85. TZOSTFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7	DCBEVT1	R	0h	デジタル比較 B 出カイベント 1 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、DCBEVT1 でトリップが発生していないことを示します。 1: 1 の読み出しは、DCBEVT1 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
6	DCAEVT1	R	0h	デジタル比較 A 出カイベント 1 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、DCAEVT1 でトリップが発生していないことを示します。 1: 1 の読み出しは、DCAEVT1 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
5	OST6	R	0h	OST6 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、OST6 でトリップが発生していないことを示します。 1: 1 の読み出しは、OST6 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
4	OST5	R	0h	OST5 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、OST5 でトリップが発生していないことを示します。 1: 1 の読み出しは、OST5 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
3	OST4	R	0h	OST4 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、OST4 でトリップが発生していないことを示します。 1: 1 の読み出しは、OST4 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
2	OST3	R	0h	OST3 トリップ・ラッチのラッチ・ステータス・フラグ 0: 0 の読み出しは、OST3 でトリップが発生していないことを示します。 1: 1 の読み出しは、OST3 の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn

表 1-85. TZOSTFLG レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	OST2	R	0h	OST2トリップ・ラッチのラッチ・ステータス・フラグ 0:0の読み出しは、OST2でトリップが発生していないことを示します。 1:1の読み出しは、OST2の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn
0	OST1	R	0h	OST1トリップ・ラッチのラッチ・ステータス・フラグ 0:0の読み出しは、OST1でトリップが発生していないことを示します。 1:1の読み出しは、OST1の選択されたイベントでトリップが発生したことを示します。 リセットの種類: SYSRSn

1.17.2.62 TZCLR レジスタ (オフセット = 97h) [リセット = 0h]

図 1-154 に、TZCLR を示し、表 1-86 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン・クリア・レジスタ

図 1-154. TZCLR レジスタ

15	14	13	12	11	10	9	8
CBCPULSE		RESERVED					
R/W-0h		R-0-0h					
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0-0h	R-0/W1S-0h						

表 1-86. TZCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	CBCPULSE	R/W	0h	サイクル・バイ・サイクル (CBC) トリップ・ラッチのクリア・パルス このビット・フィールドは、CBC トリップ・ラッチをクリアするパルスを決定します。 00: CTR = ゼロ・パルスが CBC トリップ・ラッチをクリアします。(従来の設計と同じです。) 01: CTR = PRD パルスが CBC トリップ・ラッチをクリアします。 10: CTR = ゼロまたは CTR = PRD パルスが CBC トリップ・ラッチをクリアします。 11: CBC トリップ・ラッチはクリアされません。 リセットの種類: SYSRSn
13-7	RESERVED	R-0	0h	予約済み
6	DCBEVT2	R-0/W1S	0h	デジタル比較出力 B イベント 2 のクリア・フラグ 0: 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1: 1 を書き込むと、DCBEVT2 イベント・トリップ条件がクリアされます。 リセットの種類: SYSRSn
5	DCBEVT1	R-0/W1S	0h	デジタル比較出力 B イベント 1 のクリア・フラグ 0: 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1: 1 を書き込むと、DCBEVT1 イベント・トリップ条件がクリアされます。 リセットの種類: SYSRSn
4	DCAEVT2	R-0/W1S	0h	デジタル比較出力 A イベント 2 のクリア・フラグ 0: 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1: 1 を書き込むと、DCAEVT2 イベント・トリップ条件がクリアされます。 リセットの種類: SYSRSn
3	DCAEVT1	R-0/W1S	0h	デジタル比較出力 A イベント 1 のクリア・フラグ 0: 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1: 1 を書き込むと、DCAEVT1 イベント・トリップ条件がクリアされます。 リセットの種類: SYSRSn
2	OST	R-0/W1S	0h	ワンショット・トリップ (OST) ラッチのクリア・フラグ 0: 何もしません。常に 0 が読み出されます。 1: このトリップ (セット) 状態をクリアします。 リセットの種類: SYSRSn

表 1-86. TZCLR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	CBC	R-0/W1S	0h	サイクル・バイ・サイクル (CBC) トリップ・ラッチのクリア・フラグ 0: 何もしません。常に 0 が読み出されます。 1: このトリップ (セット) 状態をクリアします。 リセットの種類: SYSRSn
0	INT	R-0/W1S	0h	グローバル割り込みクリア・フラグ 0: 何もしません。常に 0 が読み出されます。 1: この ePWM モジュールのトリップ割り込みフラグ (TZFLG[INT]) をクリアします。 注: このフラグがクリアされるまで、さらなる EPWMx_TZINT PIE 割り込みは生成されません。TZFLG[INT] ビットがクリアされても、その他のフラグ・ビットのいずれかがセットされている場合、さらなる割り込みパルスが生成されます。すべてのフラグ・ビットをクリアすると、さらなる割り込みは禁止されます。 リセットの種類: SYSRSn

1.17.2.63 TZCBCCLR レジスタ (オフセット = 98h) [リセット = 0h]

図 1-155 に、TZCBCCLR を示し、表 1-87 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン CBC クリア・レジスタ

図 1-155. TZCBCCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0/W1S-0h							

表 1-87. TZCBCCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7	DCBEVT2	R-0/W1S	0h	CBC に選択されたデジタル比較出力 B イベント 2 のクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[DCBEVT2] ビットがクリアされます。 リセットの種類 : SYSRSn
6	DCAEVT2	R-0/W1S	0h	CBC として設定されたデジタル比較出力 A イベント 2 のクリア・フラグ 2 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[DCAEVT2] ビットがクリアされます。 リセットの種類 : SYSRSn
5	CBC6	R-0/W1S	0h	サイクル・バイ・サイクル (CBC6) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[CBC6] ビットがクリアされます。 リセットの種類 : SYSRSn
4	CBC5	R-0/W1S	0h	サイクル・バイ・サイクル (CBC5) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[CBC5] ビットがクリアされます。 リセットの種類 : SYSRSn
3	CBC4	R-0/W1S	0h	サイクル・バイ・サイクル (CBC4) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[CBC4] ビットがクリアされます。 リセットの種類 : SYSRSn
2	CBC3	R-0/W1S	0h	サイクル・バイ・サイクル (CBC3) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[CBC3] ビットがクリアされます。 リセットの種類 : SYSRSn
1	CBC2	R-0/W1S	0h	サイクル・バイ・サイクル (CBC2) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[CBC2] ビットがクリアされます。 リセットの種類 : SYSRSn

表 1-87. TZCBCCLR レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
0	CBC1	R-0/W1S	0h	サイクル・バイ・サイクル (CBC1) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZCBCFLG[CBC1] ビットがクリアされます。 リセットの種類 : SYSRSn

1.17.2.64 TZOSTCLR レジスタ (オフセット = 99h) [リセット = 0h]

図 1-156 に、TZOSTCLR を示し、表 1-88 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン OST クリア・レジスタ

図 1-156. TZOSTCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
DCBEVT1	DCAEVT1	OST6	OST5	OST4	OST3	OST2	OST1
R-0/W1S-0h							

表 1-88. TZOSTCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7	DCBEVT1	R-0/W1S	0h	OST として設定されたデジタル比較出力 B イベント 1 のクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[DCBEVT1] ビットがクリアされます。 リセットの種類 : SYSRSn
6	DCAEVT1	R-0/W1S	0h	OST として設定されたデジタル比較出力 A イベント 1 のクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[DCAEVT1] ビットがクリアされます。 リセットの種類 : SYSRSn
5	OST6	R-0/W1S	0h	ワンショット (OST6) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[OST6] ビットがクリアされます。 リセットの種類 : SYSRSn
4	OST5	R-0/W1S	0h	ワンショット (OST5) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[OST5] ビットがクリアされます。 リセットの種類 : SYSRSn
3	OST4	R-0/W1S	0h	ワンショット (OST4) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[OST4] ビットがクリアされます。 リセットの種類 : SYSRSn
2	OST3	R-0/W1S	0h	ワンショット (OST3) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[OST3] ビットがクリアされます。 リセットの種類 : SYSRSn
1	OST2	R-0/W1S	0h	ワンショット (OST2) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[OST2] ビットがクリアされます。 リセットの種類 : SYSRSn
0	OST1	R-0/W1S	0h	ワンショット (OST1) トリップ・ラッチのクリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、TZOSTFLG[OST1] ビットがクリアされます。 リセットの種類 : SYSRSn

1.17.2.65 TZFRC レジスタ (オフセット = 9Bh) [リセット = 0h]

図 1-157 に、TZFRC を示し、表 1-89 に、その説明を示します。

概略表に戻ります。

トリップ・ゾーン強制レジスタ

図 1-157. TZFRC レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0-0h

表 1-89. TZFRC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-7	RESERVED	R-0	0h	予約済み
6	DCBEVT2	R-0/W1S	0h	デジタル比較出力 B イベント 2 の強制フラグ 0 : 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1 : 1 を書き込むと、DCBEVT2 イベント・トリップ条件が強制的にトリガされ、TZFLG[DCBEVT2] ビットがセットされます。 リセットの種類 : SYSRSn
5	DCBEVT1	R-0/W1S	0h	デジタル比較出力 B イベント 1 の強制フラグ 0 : 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1 : 1 を書き込むと、DCBEVT1 イベント・トリップ条件が強制的にトリガされ、TZFLG[DCBEVT1] ビットがセットされます。 リセットの種類 : SYSRSn
4	DCAEVT2	R-0/W1S	0h	デジタル比較出力 A イベント 2 の強制フラグ 0 : 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1 : 1 を書き込むと、DCAEVT2 イベント・トリップ条件が強制的にトリガされ、TZFLG[DCAEVT2] ビットがセットされます。 リセットの種類 : SYSRSn
3	DCAEVT1	R-0/W1S	0h	デジタル比較出力 A イベント 1 の強制フラグ 0 : 0 を書き込んでも何もしません。このビットは常に 0 を読み出します。 1 : 1 を書き込むと、DCAEVT1 イベント・トリップ条件が強制的にトリガされ、TZFLG[DCAEVT1] ビットがセットされます。 リセットの種類 : SYSRSn
2	OST	R-0/W1S	0h	ソフトウェアによるワンショット・トリップ・イベントの強制 0 : 0 の書き込みは無視されます。常に 0 が読み出されます。 1 : ワンショット・トリップ・イベントが強制的にトリガされ、TZFLG[OST] ビットがセットされます。 リセットの種類 : SYSRSn
1	CBC	R-0/W1S	0h	ソフトウェアによるサイクル・バイ・サイクル・トリップ・イベントの強制 0 : 0 の書き込みは無視されます。常に 0 が読み出されます。 1 : サイクル・バイ・サイクル・トリップ・イベントが強制的にトリガされ、TZFLG[CBC] ビットがセットされます。 リセットの種類 : SYSRSn
0	RESERVED	R-0	0h	予約済み

1.17.2.66 ETSEL レジスタ (オフセット = A4h) [リセット = 0h]

図 1-158 に、ETSEL を示し、表 1-90 に、その説明を示します。

概略表に戻ります。

イベント・トリガ選択レジスタ

図 1-158. ETSEL レジスタ

15		14		13		12		11		10		9		8	
SOCBEN		SOCBSEL				SOCAEN		SOCASEL							
R/W-0h		R/W-0h				R/W-0h		R/W-0h							
7		6		5		4		3		2		1		0	
RESERVED		INTSELCMP	SOCBSELCMP	SOCASELCMP	INTEN		INTSEL								
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		

表 1-90. ETSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SOCBEN	R/W	0h	ADC 変換開始 B (EPWMxSOCB) パルスの有効化 0 : EPWMxSOCB を無効化します。 1 : EPWMxSOCB パルスを有効化します。 リセットの種類 : SYSRSn
14-12	SOCBSEL	R/W	0h	EPWMxSOCB 選択オプション これらのビットは、EPWMxSOCB パルスを生成するタイミングを決定します。 000 : DCBEVT1.soc イベントの有効化 001 : イベント「時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)」を有効化します。 010 : イベント「時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)」を有効化します。 011 : イベント「時間ベース・カウンタがゼロまたは周期と一致 (TBCTR = 0x00 または TBCTR = TBPRD)」を有効化します。このモードは、アップ・ダウン・カウント・モードで有効です。 100 : イベント「時間ベース・カウンタが CMPA (そのタイマがインクリメントしている場合) または CMPC (そのタイマがインクリメントしている場合) と一致」を有効化します。 101 : イベント「時間ベース・カウンタが CMPA (そのタイマがデクリメントしている場合) または CMPC (そのタイマがデクリメントしている場合) と一致」を有効化します。 110 : イベント「時間ベース・カウンタが CMPB (そのタイマがインクリメントしている場合) または CMPD (そのタイマがインクリメントしている場合) と一致」を有効化します。 111 : イベント「時間ベース・カウンタが CMPB (そのタイマがデクリメントしている場合) または CMPD (そのタイマがデクリメントしている場合) と一致」を有効化します。(*) イベントの選択は、SOCBSELCMP ビットによって決定されます。 リセットの種類 : SYSRSn
11	SOCAEN	R/W	0h	ADC 変換開始 A (EPWMxSOCA) パルスの有効化 0 : EPWMxSOCA を無効化します。 1 : EPWMxSOCA パルスを有効化します。 リセットの種類 : SYSRSn

表 1-90. ETSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
10-8	SOCASEL	R/W	0h	EPWMxSOCA 選択オプション これらのビットは、EPWMxSOCA パルスを生成するタイミングを決定します。 000 : DCAEVT1.soc イベントを有効化します。 001 : イベント「時間ベース・カウンタがゼロと一致(TBCTR = 0x00)」を有効化します。 010 : イベント「時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)」を有効化します。 011 : イベント「時間ベース・カウンタがゼロまたは周期と一致 (TBCTR = 0x00 または TBCTR = TBPRD)」を有効化します。このモードは、アップ・ダウン・カウント・モードで有効です。 100 : イベント「時間ベース・カウンタが CMPA (そのタイマがインクリメントしている場合) または CMPC (そのタイマがインクリメントしている場合) と一致」を有効化します。 101 : イベント「時間ベース・カウンタが CMPA (そのタイマがデクリメントしている場合) または CMPC (そのタイマがデクリメントしている場合) と一致」を有効化します。 110 : イベント「時間ベース・カウンタが CMPB (そのタイマがインクリメントしている場合) または CMPD (そのタイマがインクリメントしている場合) と一致」を有効化します。 111 : イベント「時間ベース・カウンタが CMPB (そのタイマがデクリメントしている場合) または CMPD (そのタイマがデクリメントしている場合) と一致」を有効化します。(*) イベントの選択は、SOCASELCMP ビットによって決定されます。 リセットの種類 : SYSRSn
7	RESERVED	R-0	0h	予約済み
6	INTSELCMP	R/W	0h	EPWMxINT 比較レジスタ選択オプション 0 : INTSEL 選択マルチプレクサに対して、イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがデクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがデクリメントしている場合)」を有効化します。 1 : INTSEL 選択マルチプレクサに対して、イベント「時間ベース・カウンタが CMPC と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPC と一致 (そのタイマがデクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPD と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPD と一致 (そのタイマがデクリメントしている場合)」を有効化します。 リセットの種類 : SYSRSn
5	SOCBSELCMP	R/W	0h	EPWMxSOCB 比較レジスタ選択オプション 0 : SOCBSEL 選択マルチプレクサに対して、イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがデクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがデクリメントしている場合)」を有効化します。 1 : SOCBSEL 選択マルチプレクサに対して、イベント「時間ベース・カウンタが CMPC と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPC と一致 (そのタイマがデクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPD と一致 (そのタイマがインクリメントしている場合)」を有効化します。/ イベント「時間ベース・カウンタが CMPD と一致 (そのタイマがデクリメントしている場合)」を有効化します。 リセットの種類 : SYSRSn

表 1-90. ETSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	SOCASELCMP	R/W	0h	EPWMxSOCA 比較レジスタ選択オプション 0 : SOCASEL 選択マルチプレクサに対して、イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがインクリメントしている場合)」を有効化します。 / イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがデクリメントしている場合)」を有効化します。 / イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがインクリメントしている場合)」を有効化します。 / イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがデクリメントしている場合)」を有効化します。 1 : SOCASEL 選択マルチプレクサに対して、イベント「時間ベース・カウンタが CMPC と一致 (そのタイマがインクリメントしている場合)」を有効化します。 / イベント「時間ベース・カウンタが CMPC と一致 (そのタイマがデクリメントしている場合)」を有効化します。 / イベント「時間ベース・カウンタが CMPD と一致 (そのタイマがインクリメントしている場合)」を有効化します。 / イベント「時間ベース・カウンタが CMPD と一致 (そのタイマがデクリメントしている場合)」を有効化します。 リセットの種類 : SYSRSn
3	INTEN	R/W	0h	ePWM 割り込み (EPWMx_INT) 生成の有効化 0 : EPWMx_INT 生成を無効化します。 1 : EPWMx_INT 生成を有効化します。 リセットの種類 : SYSRSn
2-0	INTSEL	R/W	0h	ePWM 割り込み (EPWMx_INT) 選択オプション 000 : 予約済み 001 : イベント「時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)」を有効化します。 010 : イベント「時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)」を有効化します。 011 : イベント「時間ベース・カウンタがゼロまたは周期と一致 (TBCTR = 0x00 または TBCTR = TBPRD)」を有効化します。このモードは、アップ・ダウン・カウント・モードで有効です。 100 : イベント「時間ベース・カウンタが CMPA (そのタイマがインクリメントしている場合) または CMPC (そのタイマがインクリメントしている場合) と一致」を有効化します。 101 : イベント「時間ベース・カウンタが CMPA (そのタイマがデクリメントしている場合) または CMPC (そのタイマがデクリメントしている場合) と一致」を有効化します。 110 : イベント「時間ベース・カウンタが CMPB (そのタイマがインクリメントしている場合) または CMPD (そのタイマがインクリメントしている場合) と一致」を有効化します。 111 : イベント「時間ベース・カウンタが CMPB (そのタイマがデクリメントしている場合) または CMPD (そのタイマがデクリメントしている場合) と一致」を有効化します。 (*) イベントの選択は、INTSELCMP ビットによって決定されます。 リセットの種類 : SYSRSn

1.17.2.67 ETPS レジスタ (オフセット = A6h) [リセット = 0h]

図 1-159 に、ETPS を示し、表 1-91 に、その説明を示します。

概略表に戻ります。

イベント・トリガ・プリスケール・レジスタ

図 1-159. ETPS レジスタ

15	14	13	12	11	10	9	8
SOCBCNT		SOCBPRD		SOCACNT		SOCAPRD	
R-0h		R/W-0h		R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED		SOCPSSEL	INTPSSEL	INTCNT		INTPRD	
R-0-0h		R/W-0h	R/W-0h	R-0h		R/W-0h	

表 1-91. ETPS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	SOCBCNT	R	0h	ePWM ADC 変換開始 B イベント (EPWMxSOCB) カウンタ・レジスタ これらのビットは、選択された ETSSEL[SOCBSEL] イベントの発生回数を示します。 00：イベントは発生していません。 01：1 回のイベントが発生しました。 10：2 回のイベントが発生しました。 11：3 回のイベントが発生しました。 リセットの種類：SYSRSn
13-12	SOCBPRD	R/W	0h	ePWM ADC 変換開始 B イベント (EPWMxSOCB) 周期の選択 これらのビットは、EPWMxSOCB パルスが生成されるのに必要な、選択された ETSSEL[SOCBSEL] イベントの発生回数を決定します。パルスが生成されるには、そのパルスが有効化されている (ETSSEL[SOCBEN] = 1) 必要があります。前回の変換開始 (ETFLG[SOCB] = 1) によってステータス・フラグがセットされていても、SOCB パルスは生成されます。SOCB パルスが生成されると、ETPS[SOCBCNT] ビットは自動的にクリアされます。 00：SOCB イベント・カウンタを無効化します。EPWMxSOCB パルスは生成されません。 01：最初のイベント (ETPS[SOCBCNT] = 0,1) で EPWMxSOCB パルスが生成されます。 10：2 番目のイベント (ETPS[SOCBCNT] = 1,0) で EPWMxSOCB パルスが生成されます。 11：3 番目のイベント (ETPS[SOCBCNT] = 1,1) で EPWMxSOCB パルスが生成されます。 リセットの種類：SYSRSn
11-10	SOCACNT	R	0h	ePWM ADC 変換開始 A イベント (EPWMxSOCA) カウンタ・レジスタ これらのビットは、選択された ETSSEL[SOCASEL] イベントの発生回数を示します。 00：イベントは発生していません。 01：1 回のイベントが発生しました。 10：2 回のイベントが発生しました。 11：3 回のイベントが発生しました。 リセットの種類：SYSRSn

表 1-91. ETPS レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
9-8	SOCAPRD	R/W	0h	ePWM ADC 変換開始 A イベント (EPWMxSOCA) 周期の選択 これらのビットは、EPWMxSOCA パルスが生成されるのに必要な、選択された ETSSEL[SOCASEL] イベントの発生回数を決定します。パルスが生成されるには、そのパルスが有効化されている (ETSSEL[SOCAEN] = 1) 必要があります。前回の変換開始 (ETFLG[SOCA] = 1) によってステータス・フラグがセットされていても、SOCA パルスは生成されます。SOCA パルスが生成されると、ETPS[SOCACNT] ビットは自動的にクリアされます。 00 : SOCA イベント・カウンタを無効化します。EPWMxSOCA パルスは生成されません。 01 : 最初のイベント (ETPS[SOCACNT] = 0,1) で EPWMxSOCA パルスが生成されます。 10 : 2 番目のイベント (ETPS[SOCACNT] = 1,0) で EPWMxSOCA パルスが生成されます。 11 : 3 番目のイベント (ETPS[SOCACNT] = 1,1) で EPWMxSOCA パルスが生成されます。 リセットの種類 : SYSRSn
7-6	RESERVED	R-0	0h	予約済み
5	SOCPSSEL	R/W	0h	EPWMxSOC A/B プリスケール選択ビット 0 : ETPS [SOCACNT/SOCBCNT] および [SOCAPRD/SOCBPRD] レジスタを選択して、イベントの頻度 (0~3 回のイベントごとに 1 回の変換開始パルス) を決定します。 1 : ETSOCP [SOCACNT2/SOCBCNT2] および [SOCAPRD2/SOCBPRD2] レジスタを選択して、イベントの頻度 (0~15 回のイベントごとに 1 回の変換開始パルス) を決定します。 リセットの種類 : SYSRSn
4	INTPSSEL	R/W	0h	EPWMxINTn プリスケール選択ビット 0 : ETPS [INTCNT, INTPRD] レジスタを選択して、イベントの頻度 (0~3 回のイベントごとに 1 回の割り込み) を決定します。 1 : ETINTPS [INTCNT2, INTPRD2] レジスタを選択して、イベントの頻度 (0~15 回のイベントごとに 1 回の割り込み) を決定します。 リセットの種類 : SYSRSn
3-2	INTCNT	R	0h	ePWM 割り込みイベント (EPWMx_INT) カウンタ・レジスタ これらのビットは、選択された ETSSEL[INTSEL] イベントの発生回数を示します。これらのビットは、割り込みパルスが生成されると自動的にクリアされます。割り込みが無効化されている場合 (ETSSEL[INT] = 0)、または割り込みフラグがセットされている場合 (ETFLG[INT] = 1)、カウンタは周期値に達する (ETPS[INTCNT] = ETPS[INTPRD]) とイベントのカウンタを停止します。 00 : イベントは発生していません。 01 : 1 回のイベントが発生しました。 10 : 2 回のイベントが発生しました。 11 : 3 回のイベントが発生しました。 リセットの種類 : SYSRSn

表 1-91. ETPS レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1-0	INTPRD	R/W	0h	<p>ePWM 割り込み (EPWMx_INT) 周期の選択</p> <p>これらのビットは、割り込みが生成されるのに必要な、選択された ETSSEL[INTSEL] イベントの発生回数を決定します。割り込みが生成されるには、その割り込みが有効化されている (ETSEL[INT] = 1) 必要があります。前回の割り込み (ETFLG[INT] = 1) によって割り込みステータス・フラグがセットされている場合、ETCLR[INT] ビットによってそのフラグがクリアされるまで、割り込みは生成されません。これにより、別の割り込みがまだ処理されている間、1 つの割り込みを保留できます。割り込みが生成されると、ETPS[INTCNT] ビットは自動的にクリアされます。</p> <p>割り込みが有効化されており、かつそのステータス・フラグがクリアされている場合、現在のカウンタ値と同じ INTPRD 値を書き込むと、割り込みがトリガされます。</p> <p>現在のカウンタ値より小さい INTPRD 値を書き込むと、未定義の状態になります。ゼロまたはゼロ以外の新しい INTPRD 値が書き込まれるのと同時にカウンタ・イベントが発生すると、カウンタがインクリメントされます。</p> <p>00: 割り込みイベント・カウンタを無効化します。割り込みは生成されず、ETFRC[INT] は無視されます。</p> <p>01: 最初のイベント (INTCNT = 01) で割り込みが生成されます。</p> <p>10: 2 番目のイベント (ETPS[INTCNT] = 1,0) で割り込みが生成されます。</p> <p>11: 3 番目のイベント (ETPS[INTCNT] = 1,1) で割り込みが生成されます。</p> <p>リセットの種類: SYSRSn</p>

1.17.2.68 ETFLG レジスタ (オフセット = A8h) [リセット = 0h]

図 1-160 に、ETFLG を示し、表 1-92 に、その説明を示します。

概略表に戻ります。

イベント・トリガ・フラグ・レジスタ

図 1-160. ETFLG レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0-0h				R-0h	R-0h	R-0-0h	R-0h

表 1-92. ETFLG レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3	SOCB	R	0h	ラッチ付き ePWM ADC 変換開始 A (EPWMxSOCB) ステータス・フラグ ETFLG[INT] フラグとは異なり、EPWMxSOCB 出力は、フラグ・ビットがセットされている場合でもパルスを出し続けます。 0：イベントが発生していないことを示します。 1：EPWMxSOCB で変換開始パルスが生成されたことを示します。 フラグ・ビットがセットされていても、EPWMxSOCB 出力は生成され続けます。 リセットの種類：SYSRSn
2	SOCA	R	0h	ラッチ付き ePWM ADC 変換開始 A (EPWMxSOCA) ステータス・フラグ ETFLG[INT] フラグとは異なり、EPWMxSOCA 出力は、フラグ・ビットがセットされている場合でもパルスを出し続けます。 0：イベントが発生していないことを示します。 1：EPWMxSOCA で変換開始パルスが生成されたことを示します。 フラグ・ビットがセットされていても、EPWMxSOCA 出力は生成され続けます。 リセットの種類：SYSRSn
1	RESERVED	R-0	0h	予約済み
0	INT	R	0h	ラッチ付き ePWM 割り込み (EPWMx_INT) ステータス・フラグ 0：イベントが発生していないことを示します。 1：ePWMx 割り込み (EPWMx_INT) が生成されたことを示します。 このフラグ・ビットがクリアされるまで、さらなる割り込みは生成されません。ETFLG[INT] ビットがまだセットされている間、最大 1 つの割り込みを保留できます。1 つの割り込みが保留されている場合、ETFLG[INT] ビットがクリアされるまで、割り込みは生成されません。 リセットの種類：SYSRSn

1.17.2.69 ETCLR レジスタ (オフセット = AAh) [リセット = 0h]

図 1-161 に、ETCLR を示し、表 1-93 に、その説明を示します。

概略表に戻ります。

イベント・トリガ・クリア・レジスタ

図 1-161. ETCLR レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0-0h	R-0/W1S-0h

表 1-93. ETCLR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3	SOCB	R-0/W1S	0h	ePWM ADC 変換開始 A (EPWMxSOCB) フラグ・クリア・ビット 0 : 0 を書き込んでも何もしません。常に 0 が読み出されます。 1 : ETFLG[SOCB] フラグ・ビットがクリアされます。 リセットの種類 : SYSRSn
2	SOCA	R-0/W1S	0h	ePWM ADC 変換開始 A (EPWMxSOCA) フラグ・クリア・ビット 0 : 0 を書き込んでも何もしません。常に 0 が読み出されます。 1 : ETFLG[SOCA] フラグ・ビットがクリアされます。 リセットの種類 : SYSRSn
1	RESERVED	R-0	0h	予約済み
0	INT	R-0/W1S	0h	ePWM 割り込み (EPWMx_INT) フラグ・クリア・ビット 0 : 0 を書き込んでも何もしません。常に 0 が読み出されます。 1 : ETFLG[INT] フラグ・ビットがクリアされ、さらなる割り込みパ ルスが生成されるようになります。 リセットの種類 : SYSRSn

1.17.2.70 ETFRC レジスタ (オフセット = ACh) [リセット = 0h]

図 1-162 に、ETFRC を示し、表 1-94 に、その説明を示します。

概略表に戻ります。

イベント・トリガ強制レジスタ

図 1-162. ETFRC レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0-0h				R-0/W1S-0h	R-0/W1S-0h	R-0-0h	R-0/W1S-0h

表 1-94. ETFRC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-4	RESERVED	R-0	0h	予約済み
3	SOCB	R-0/W1S	0h	SOCB 強制ビット SOCB パルスは、ETSEL レジスタでイベントが有効化されている場合にのみ生成されます。ETFLG[SOCB] フラグ・ビットは、どのような場合でもセットされます。 0：このビットへの 0 の書き込みは無視されます。常に 0 が読み出されます。 1：EPWMxSOCB にパルスが生成され、SOCBFLG ビットがセットされます。このビットはテストのために使用されます。 リセットの種類：SYSRSn
2	SOCA	R-0/W1S	0h	SOCA 強制ビット SOCA パルスは、ETSEL レジスタでイベントが有効化されている場合にのみ生成されます。ETFLG[SOCA] フラグ・ビットは、どのような場合でもセットされます。 0：このビットへの 0 の書き込みは無視されます。常に 0 が読み出されます。 1：EPWMxSOCOA にパルスが生成され、SOCAFLG ビットがセットされます。このビットはテストのために使用されます。 リセットの種類：SYSRSn
1	RESERVED	R-0	0h	予約済み
0	INT	R-0/W1S	0h	INT 強制ビット 割り込みは、ETSEL レジスタでイベントが有効化されている場合にのみ生成されます。INT フラグ・ビットは、どのような場合でもセットされます。 0：このビットへの 0 の書き込みは無視されます。常に 0 が読み出されます。 1：EPWMxINT で割り込みが生成され、INT フラグ・ビットがセットされます。このビットはテストのために使用されます。 リセットの種類：SYSRSn

1.17.2.71 ETINTPS レジスタ (オフセット = AEh) [リセット = 0h]

図 1-163 に、ETINTPS を示し、表 1-95 に、その説明を示します。

概略表に戻ります。

イベント・トリガ割り込みプリスケール・レジスタ

図 1-163. ETINTPS レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
INTCNT2				INTPRD2			
R-0h				R/W-0h			

表 1-95. ETINTPS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	RESERVED	R-0	0h	予約済み
7-4	INTCNT2	R	0h	EPWMxINT カウンタ 2 ETPS[INTPSSEL]=1 の場合、これらのビットは、選択されたイベントの発生回数を示します。 0000 : イベントなし 0001 : 1 回のイベント 0010 : 2 回のイベント 0011 : 3 回のイベント 0100 : 4 回のイベント ... 1111 : 15 回のイベント リセットの種類 : SYSRSn
3-0	INTPRD2	R/W	0h	EPWMxINT 周期 2 の選択 ETPS[INTPSSEL] = 1 の場合、これらのビットは、割り込みが生成されるのに必要な、選択されたイベントの発生回数を選択します。 0000 : カウンタを無効化します。 0001 : 最初のイベント (INTCNT = 1) で割り込みが生成されます。 0010 : 2 番目のイベント (INTCNT = 2) で割り込みが生成されます。 0011 : 3 番目のイベント (INTCNT = 3) で割り込みが生成されます。 0100 : 4 番目のイベント (INTCNT = 4) で割り込みが生成されます。 ... 1111 : 15 番目のイベント (INTCNT = 15) で割り込みが生成されます。 リセットの種類 : SYSRSn

1.17.2.72 ETSOCPS レジスタ (オフセット = B0h) [リセット = 0h]

図 1-164 に、ETSOCPS を示し、表 1-96 に、その説明を示します。

概略表に戻ります。

イベント・トリガ変換開始プリスケール・レジスタ

図 1-164. ETSOCPS レジスタ

15	14	13	12	11	10	9	8
SOCBCNT2				SOCBPRD2			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
SOCACNT2				SOCAPRD2			
R-0h				R/W-0h			

表 1-96. ETSOCPS レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	SOCBCNT2	R	0h	EPWMxSOCB カウンタ 2 ETPS[SOCPSSEL] = 1 の場合、これらのビットは、選択されたイベントの発生回数を示します。 0000 : イベントなし 0001 : 1 回のイベント 0010 : 2 回のイベント 0011 : 3 回のイベント 0100 : 4 回のイベント ... 1111 : 15 回のイベント リセットの種類 : SYSRSn
11-8	SOCBPRD2	R/W	0h	EPWMxSOCB 周期 2 の選択 ETPS[SOCPSSEL] = 1 の場合、これらのビットは、SOCB パルスが生成されるのに必要な、選択されたイベントの発生回数を決定します。 0000 : カウンタを無効化します。 0001 : 最初のイベント (SOCBCNT2 = 1) で変換開始パルスが生成されます。 0010 : 2 番目のイベント (SOCBCNT2 = 2) で変換開始パルスが生成されます。 0011 : 3 番目のイベント (SOCBCNT2 = 3) で変換開始パルスが生成されます。 0100 : 4 番目のイベント (SOCBCNT2 = 4) で変換開始パルスが生成されます。 ... 1111 : 15 番目のイベント (SOCBCNT2 = 15) で変換開始パルスが生成されます。 リセットの種類 : SYSRSn
7-4	SOCACNT2	R	0h	EPWMxSOCA カウンタ 2 ETPS[SOCPSSEL] = 1 の場合、これらのビットは、選択されたイベントの発生回数を示します。 0000 : イベントなし 0001 : 1 回のイベント 0010 : 2 回のイベント 0011 : 3 回のイベント 0100 : 4 回のイベント ... 1111 : 15 回のイベント リセットの種類 : SYSRSn

表 1-96. ETSOCPS レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-0	SOCAPRD2	R/W	0h	EPWMxSOCA 周期 2 の選択 ETPS[SOCPSSEL] = 1 の場合、これらのビットは、SOCA パルスが生成されるのに必要な、選択されたイベントの発生回数を決定します。 0000 : カウンタを無効化します。 0001 : 最初のイベント (SOCACNT2 = 1) で変換開始パルスが生成されます。 0010 : 2 番目のイベント (SOCACNT2 = 2) で変換開始パルスが生成されます。 0011 : 3 番目のイベント (SOCACNT2 = 3) で変換開始パルスが生成されます。 0100 : 4 番目のイベント (SOCACNT2 = 4) で変換開始パルスが生成されます。 ... 1111 : 15 番目のイベント (SOCACNT2 = 15) で変換開始パルスが生成されます。 リセットの種類 : SYSRSn

1.17.2.73 ETCNTINITCTL レジスタ (オフセット = B2h) [リセット = 0h]

図 1-165 に、ETCNTINITCTL を示し、表 1-97 に、その説明を示します。

概略表に戻ります。

イベント・トリガ・カウンタ初期化制御レジスタ

図 1-165. ETCNTINITCTL レジスタ

15		14		13		12		11		10		9		8	
SOCBINITEN		SOCAINITEN		INTINITEN		SOCBINITFRC		SOCAINITFRC		INTINITFRC		RESERVED			
R/W-0h		R/W-0h		R/W-0h		R-0/W1S-0h		R-0/W1S-0h		R-0/W1S-0h		R-0-0h			
7		6		5		4		3		2		1		0	
RESERVED															
R-0-0h															

表 1-97. ETCNTINITCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SOCBINITEN	R/W	0h	EPWMxSOCB カウンタ 2 初期化イネーブル 0：何もしません。 1：ETCNTINIT[SOCBINIT] の内容による EPWMxSOCB カウンタの初期化 (同期イベントまたはソフトウェア強制時) を有効化します。 リセットの種類：SYSRSn
14	SOCAINITEN	R/W	0h	EPWMxSOCA カウンタ 2 初期化イネーブル 0：何もしません。 1：ETCNTINIT[SOCAINIT] の内容による EPWMxSOCA カウンタの初期化 (同期イベントまたはソフトウェア強制時) を有効化します。 リセットの種類：SYSRSn
13	INTINITEN	R/W	0h	EPWMxINT カウンタ 2 初期化イネーブル 0：何もしません。 1：ETCNTINIT[INTINIT] の内容による EPWMxINT カウンタ 2 の初期化 (同期イベントまたはソフトウェア強制時) を有効化します。 リセットの種類：SYSRSn
12	SOCBINITFRC	R-0/W1S	0h	EPWMxSOCB カウンタ 2 初期化の強制 0：何もしません。 1：ETCNTINIT[SOCBINIT] の内容で ET EPWMxSOCB カウンタを強制的に初期化します。 リセットの種類：SYSRSn
11	SOCAINITFRC	R-0/W1S	0h	EPWMxSOCA カウンタ 2 初期化の強制 0：何もしません。 1：ETCNTINIT[SOCAINIT] の内容で ET EPWMxSOCA カウンタを強制的に初期化します。 リセットの種類：SYSRSn
10	INTINITFRC	R-0/W1S	0h	EPWMxINT カウンタ 2 初期化の強制 0：何もしません。 1：ETCNTINIT[INTINIT] の内容で ET EPWMxINT カウンタを強制的に初期化します。 リセットの種類：SYSRSn
9-0	RESERVED	R-0	0h	予約済み

1.17.2.74 ETCNTINIT レジスタ (オフセット = B4h) [リセット = 0h]

図 1-166 に、ETCNTINIT を示し、表 1-98 に、その説明を示します。

概略表に戻ります。

イベント・トリガ・カウンタ初期化レジスタ

図 1-166. ETCNTINIT レジスタ

15	14	13	12	11	10	9	8
RESERVED				SOCBINIT			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
SOCAINIT				INTINIT			
R/W-0h				R/W-0h			

表 1-98. ETCNTINIT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	RESERVED	R	0h	予約済み
11-8	SOCBINIT	R/W	0h	EPWMxSOCB カウンタ 2 初期化ビット ET EPWMxSOCB カウンタは、ePWM 同期イベントまたはソフトウェア強制時に、このレジスタの内容によって初期化されます。 リセットの種類：SYSRSn
7-4	SOCAINIT	R/W	0h	EPWMxSOCA カウンタ 2 初期化ビット ET EPWMxSOCA カウンタは、ePWM 同期イベントまたはソフトウェア強制時に、このレジスタの内容によって初期化されます。 リセットの種類：SYSRSn
3-0	INTINIT	R/W	0h	EPWMxINT カウンタ 2 初期化ビット ET EPWMxINT カウンタは、ePWM 同期イベントまたはソフトウェア強制時に、このレジスタの内容によって初期化されます。 リセットの種類：SYSRSn

1.17.2.75 DCTRIPSEL レジスタ (オフセット = C0h) [リセット = 0h]

図 1-167 に、DCTRIPSEL を示し、表 1-99 に、その説明を示します。

概略表に戻ります。

デジタル比較トリップ選択レジスタ

図 1-167. DCTRIPSEL レジスタ

15	14	13	12	11	10	9	8
DCBLCOMPSEL				DCBHCOMPSEL			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
DCALCOMPSEL				DCAHCOMPSEL			
R/W-0h				R/W-0h			

表 1-99. DCTRIPSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	DCBLCOMPSEL	R/W	0h	デジタル比較 B Low 入力選択ビット 0000 : TRIPIN1 0001 : TRIPIN2 0010 : TRIPIN3 0011 : TRIPIN4 ... 1011 : TRIPIN12 1100 : 予約済み 1101 : TRIPIN14 1110 : TRIPIN15 1111 : トリップ組み合わせ入力 (DCBLTRIPSEL レジスタにより選択されたすべてのトリップ入力の論理和) リセットの種類 : SYSRSn
11-8	DCBHCOMPSEL	R/W	0h	デジタル比較 B High 入力選択ビット 0000 : TRIPIN1 0001 : TRIPIN2 0010 : TRIPIN3 0011 : TRIPIN4 ... 1011 : TRIPIN12 1100 : 予約済み 1101 : TRIPIN14 1110 : TRIPIN15 1111 : トリップ組み合わせ入力 (DCBHTRIPSEL レジスタにより選択されたすべてのトリップ入力の論理和) リセットの種類 : SYSRSn
7-4	DCALCOMPSEL	R/W	0h	デジタル比較 A Low 入力選択ビット 0000 : TRIPIN1 0001 : TRIPIN2 0010 : TRIPIN3 0011 : TRIPIN4 ... 1011 : TRIPIN12 1100 : 予約済み 1101 : TRIPIN14 1110 : TRIPIN15 1111 : トリップ組み合わせ入力 (DCALTRIPSEL レジスタにより選択されたすべてのトリップ入力の論理和) リセットの種類 : SYSRSn

表 1-99. DCTRIPSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
3-0	DCAHCOMPSEL	R/W	0h	デジタル比較 A High 入力選択ビット 0000 : TRIPIN1 0001 : TRIPIN2 0010 : TRIPIN3 0011 : TRIPIN4 ... 1011 : TRIPIN12 1100 : 予約済み 1101 : TRIPIN14 1110 : TRIPIN15 1111 : トリップ組み合わせ入力 (DCAHTRIPSEL レジスタにより選 択されたすべてのトリップ入力の論理和) リセットの種類 : SYSRSn

1.17.2.76 DCACTL レジスタ (オフセット = C3h) [リセット = 0h]

図 1-168 に、DCACTL を示し、表 1-100 に、その説明を示します。

概略表に戻ります。

デジタル比較 A 制御レジスタ

図 1-168. DCACTL レジスタ

15		14		13		12		11		10		9		8	
EVT2LAT		EVT2LATCLRSEL		EVT2LATSEL		RESERVED		EVT2FRCSYN CSEL		EVT2SRCSEL					
R-0h		R/W-0h		R/W-0h		R-0-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7		6		5		4		3		2		1		0	
EVT1LAT		EVT1LATCLRSEL		EVT1LATSEL		EVT1SYNCE		EVT1SOCE		EVT1FRCSYN CSEL		EVT1SRCSEL			
R-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-100. DCACTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	EVT2LAT	R	0h	DCAEVT2LAT 信号のステータスを示します。 0 : DCAEVT2LAT ラッチはクリアされています。 1 : DCAEVT2LAT ラッチはセットされています。 リセットの種類 : SYSRSn
14-13	EVT2LATCLRSEL	R/W	0h	DCAEVT2 ラッチ・クリア・ソースの選択 : 00 : CNT_ZERO イベントが DCAEVT2 ラッチをクリアします。 01 : PRD_EQ イベントが DCAEVT2 ラッチをクリアします。 10 : CNT_ZERO イベントまたは PRD_EQ イベントが DCAEVT2 ラッチをクリアします。 11 : 予約済み。 リセットの種類 : SYSRSn
12	EVT2LATSEL	R/W	0h	DCAEVT2 被ラッチ信号の選択 : 0 : DCAEVT2.force のソースとして DCAEVT2 被ラッチ信号を選択しません。 1 : DCAEVT2.force のソースとして DCAEVT2 被ラッチ信号を選択します。 リセットの種類 : SYSRSn
11-10	RESERVED	R-0	0h	予約済み
9	EVT2FRCSYN CSEL	R/W	0h	DCAEVT2 強制同期信号の選択 0 : ソースは EPWMCLK と同期しています。 1 : ソースは非同期的にパススルー (通過) します。 リセットの種類 : SYSRSn
8	EVT2SRCSEL	R/W	0h	DCAEVT2 ソース信号の選択 0 : ソースは DCAEVT2 信号です。 1 : ソースは DCEVTFILT 信号です。 リセットの種類 : SYSRSn
7	EVT1LAT	R	0h	DCAEVT1LAT 信号のステータスを示します。 0 : DCAEVT1LAT ラッチはクリアされています。 1 : DCAEVT1LAT ラッチはセットされています。 リセットの種類 : SYSRSn
6-5	EVT1LATCLRSEL	R/W	0h	DCAEVT1 ラッチ・クリア・ソースの選択 : 00 : CNT_ZERO イベントが DCAEVT1 ラッチをクリアします。 01 : PRD_EQ イベントが DCAEVT1 ラッチをクリアします。 10 : CNT_ZERO イベントまたは PRD_EQ イベントが DCAEVT1 ラッチをクリアします。 11 : 予約済み。 リセットの種類 : SYSRSn

表 1-100. DCACTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	EVT1LATSEL	R/W	0h	DCAEVT1 被ラッチ信号の選択： 0：DCAEVT1.force のソースとして DCAEVT1 被ラッチ信号を選択しません。 1：DCAEVT1.force のソースとして DCAEVT1 被ラッチ信号を選択します。 リセットの種類：SYSRSn
3	EVT1SYNCE	R/W	0h	DCAEVT1 同期、有効化 / 無効化 0：同期の生成を無効化します。 1：同期の生成を有効化します。 リセットの種類：SYSRSn
2	EVT1SOCE	R/W	0h	DCAEVT1 SOC、有効化 / 無効化 0：変換開始の生成を無効化します。 1：変換開始の生成を有効化します。 リセットの種類：SYSRSn
1	EVT1FRCSYNCSSEL	R/W	0h	DCAEVT1 強制同期信号の選択 0：ソースは EPWMCLK と同期しています。 1：ソースは非同期的にパススルー (通過) します。 リセットの種類：SYSRSn
0	EVT1SRCSEL	R/W	0h	DCAEVT1 ソース信号の選択 0：ソースは DCAEVT1 信号です。 1：ソースは DCEVTFILT 信号です。 リセットの種類：SYSRSn

1.17.2.77 DCBCTL レジスタ (オフセット = C4h) [リセット = 0h]

図 1-169 に、DCBCTL を示し、表 1-101 に、その説明を示します。

概略表に戻ります。

デジタル比較 B 制御レジスタ

図 1-169. DCBCTL レジスタ

15		14		13		12		11		10		9		8	
EVT2LAT		EVT2LATCLRSEL		EVT2LATSEL		RESERVED		EVT2FRCSYN CSEL		EVT2SRCSEL					
R-0h		R/W-0h		R/W-0h		R-0-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7		6		5		4		3		2		1		0	
EVT1LAT		EVT1LATCLRSEL		EVT1LATSEL		EVT1SYNCE		EVT1SOCE		EVT1FRCSYN CSEL		EVT1SRCSEL			
R-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-101. DCBCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	EVT2LAT	R	0h	DCBEVT2LAT 信号のステータスを示します。 0 DCBEVT2LAT ラッチはクリアされています。 1 DCBEVT2LAT ラッチはセットされています。 リセットの種類：SYSRSn
14-13	EVT2LATCLRSEL	R/W	0h	DCBEVT2 ラッチ・クリア・ソースの選択： 00 CNT_ZERO イベントが DCBEVT2 ラッチをクリアします。 01 PRD_EQ イベントが DCBEVT2 ラッチをクリアします。 10 CNT_ZERO イベントまたは PRD_EQ イベントが DCBEVT2 ラッチをクリアします。 11 予約済み。 リセットの種類：SYSRSn
12	EVT2LATSEL	R/W	0h	DCBEVT2 被ラッチ信号の選択： 0 DCBEVT2.force のソースとして DCBEVT2 被ラッチ信号を選択しません (図「DCBEVT1.force/DCBEVT2.force 生成への変更」を参照)。 1 DCBEVT2.force のソースとして DCBEVT2 被ラッチ信号を選択します。 リセットの種類：SYSRSn
11-10	RESERVED	R-0	0h	予約済み
9	EVT2FRCSYN CSEL	R/W	0h	DCBEVT2 強制同期信号の選択 0：ソースは EPWMCLK と同期しています。 1：ソースは非同期的にパススルー (通過) します。 リセットの種類：SYSRSn
8	EVT2SRCSEL	R/W	0h	DCBEVT2 ソース信号の選択 0：ソースは DCBEVT2 信号です。 1：ソースは DCEVTFILT 信号です。 リセットの種類：SYSRSn
7	EVT1LAT	R	0h	DCBEVT1LAT 信号のステータスを示します。 0 DCBEVT1LAT ラッチはクリアされています。 1 DCBEVT1LAT ラッチはセットされています。 リセットの種類：SYSRSn
6-5	EVT1LATCLRSEL	R/W	0h	DCBEVT1 ラッチ・クリア・ソースの選択： 00 CNT_ZERO イベントが DCBEVT1 ラッチをクリアします。 01 PRD_EQ イベントが DCBEVT1 ラッチをクリアします。 10 CNT_ZERO イベントまたは PRD_EQ イベントが DCBEVT1 ラッチをクリアします。 11 予約済み。 リセットの種類：SYSRSn

表 1-101. DCBCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	EVT1LATSEL	R/W	0h	DCBEVT1 被ラッチ信号の選択： 0 DCBEVT1.force のソースとして DCBEVT1 被ラッチ信号を選択 しません (図「DCBEVT1.force/DCBEVT2.force 生成への変更」を参 照)。 1 DCBEVT1.force のソースとして DCBEVT1 被ラッチ信号を選択し ます。 リセットの種類：SYSRSn
3	EVT1SYNCE	R/W	0h	DCBEVT1 同期、有効化 / 無効化 0：同期の生成を無効化します。 1：同期の生成を有効化します。 リセットの種類：SYSRSn
2	EVT1SOCE	R/W	0h	DCBEVT1 SOC、有効化 / 無効化 0：変換開始の生成を無効化します。 1：変換開始の生成を有効化します。 リセットの種類：SYSRSn
1	EVT1FRCSYNCSSEL	R/W	0h	DCBEVT1 強制同期信号の選択 0：ソースは EPWMCLK と同期しています。 1：ソースは非同期的にパススルー (通過) します。 リセットの種類：SYSRSn
0	EVT1SRCSEL	R/W	0h	DCBEVT1 ソース信号の選択 0：ソースは DCBEVT1 信号です。 1：ソースは DCEVTFILT 信号です。 リセットの種類：SYSRSn

1.17.2.78 DCFCTL レジスタ (オフセット = C7h) [リセット = 0h]

図 1-170 に、DCFCTL を示し、表 1-102 に、その説明を示します。

概略表に戻ります。

デジタル比較フィルタ制御レジスタ

図 1-170. DCFCTL レジスタ

15	14	13	12	11	10	9	8
EDGESTATUS			EDGECOUNT			EDGEMODE	
R-0h			R/W-0h			R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	EDGEFILTSEL	PULSESEL		BLANKINV	BLANKE	SRCSEL	
R-0-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	

表 1-102. DCFCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-13	EDGESTATUS	R	0h	エッジ・ステータス： これらのビットは、現在キャプチャされているエッジの総数を反映します。その値が EDGECOUNT と一致すると、このステータス・ビットはゼロに設定されます。その後 DCEVTFILT 信号に出力できる TBCLK 幅のパルスが生成されます。エッジ・カウンタは、EDGECOUNT 値に 000 を書き込むことでリセットできます。 リセットの種類：SYSRSn
12-10	EDGECOUNT	R/W	0h	エッジ・カウンタ：これらのビットは、DCEVTFILT 信号に TBCLK 幅のパルスを生成するのに必要なエッジのカウント数を選択します。 000：エッジなし、現在の EDGESTATUS ビットを 0,0,0 にリセットします。 001：1つのエッジ 010：2つのエッジ 011：3つのエッジ 100：4つのエッジ 101：5つのエッジ 110：6つのエッジ 111：7つのエッジ リセットの種類：SYSRSn
9-8	EDGEMODE	R/W	0h	エッジ・モードの選択： 00：Low から High へのエッジ 01：High から Low へのエッジ 10：両方のエッジ 11：予約済み リセットの種類：SYSRSn
7	RESERVED	R-0	0h	予約済み
6	EDGEFILTSEL	R/W	0h	エッジ・フィルタの選択： 0：エッジ・フィルタが選択されていません。 1：エッジ・フィルタが選択されています。 リセットの種類：SYSRSn
5-4	PULSESEL	R/W	0h	ブランキングおよびキャプチャ・アラインメント用パルスの選択 00：時間ベース・カウンタが周期と一致 (TBCTR = TBPRD) 01：時間ベース・カウンタがゼロと一致 (TBCTR = 0x00) 10：時間ベース・カウンタがゼロまたは周期と一致 (TBCTR = 0x00 または TBCTR = TBPRD) 11：BLANKPULSEMIX リセットの種類：SYSRSn
3	BLANKINV	R/W	0h	ブランキング・ウィンドウの反転 0：ブランキング・ウィンドウは反転されません。 1：ブランキング・ウィンドウは反転されます。 リセットの種類：SYSRSn

表 1-102. DCFCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
2	BLANKE	R/W	0h	ブランキング・ウィンドウの有効化 / 無効化 0 : ブランキング・ウィンドウは無効化されます。 1 : ブランキング・ウィンドウは有効化されます。 リセットの種類 : SYSRSn
1-0	SRCSEL	R/W	0h	フィルタ・ブロック信号ソースの選択 00 : ソースは DCAEVT1 信号です。 01 : ソースは DCAEVT2 信号です。 10 : ソースは DCBEVT1 信号です。 11 : ソースは DCBEVT2 信号です。 リセットの種類 : SYSRSn

1.17.2.79 DCCAPCTL レジスタ (オフセット = C8h) [リセット = 0h]

図 1-171 に、DCCAPCTL を示し、表 1-103 に、その説明を示します。

概略表に戻ります。

デジタル比較キャプチャ制御レジスタ

図 1-171. DCCAPCTL レジスタ

15		14		13		12		11		10		9		8	
CAPMODE		CAPCLR		CAPSTS		RESERVED									
R/W-0h		R-0/W1S-0h		R-0h		R-0-0h									
7		6		5		4		3		2		1		0	
RESERVED												SHDWMODE		CAPE	
R-0-0h												R/W-0h		R/W-0h	

表 1-103. DCCAPCTL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	CAPMODE	R/W	0h	カウンタ・キャプチャ・モード 0 : DCEVTFILT が発生すると、カウンタ・キャプチャが有効化されている場合、現在の TBCNT 値がアクティブ・レジスタにキャプチャされます。それぞれのトリップ・イベントが発生すると、次の PRD_eq または CNT_zero イベント (DCFCTL レジスタの PULSESEL ビットで選択されたイベント) がキャプチャ機能を再トリガするまで、さらなるトリップ (キャプチャ) イベントは無視されます。 DCCAPCTL レジスタの SHDWMODE ビットによってアクティブ・モードが有効化されている場合、このレジスタの CPU 読み出しはアクティブ・レジスタ値を返します。 DCCAPCTL レジスタの SHDWMODE ビットによってシャドウ・モードが有効化されている場合、アクティブ・レジスタは PRD_eq または CNT_zero イベント (DCFCTL レジスタの PULSESEL ビットによって選択された方) のシャドウ・レジスタにコピーされます。このレジスタの CPU 読み出しはシャドウ・レジスタ値を返します。 1 : DCEVTFILT が発生すると、カウンタ・キャプチャが有効化されている場合、現在の TBCNT 値がアクティブ・レジスタにキャプチャされます。それぞれのトリップ・イベントが発生すると、CAPSTS フラグがセットされ、このビットがクリアされるまで、さらなるトリップ (キャプチャ) イベントは無視されます。DCCAPCTL レジスタの CAPCLR ビットに書き込むと、CAPSTS がクリアされ、キャプチャ機能が再トリガされます。 DCCAPCTL レジスタの SHDWMODE ビットによってアクティブ・モードが有効化されている場合、このレジスタの CPU 読み出しはアクティブ・レジスタ値を返します。 DCCAPCTL レジスタの SHDWMODE ビットによってシャドウ・モードが有効化されている場合、アクティブ・レジスタは PRD_eq または CNT_zero イベント (DCFCTL レジスタの PULSESEL ビットによって選択された方) のシャドウ・レジスタにコピーされます。このレジスタの CPU 読み出しはシャドウ・レジスタ値を返します。 リセットの種類 : SYSRSn
14	CAPCLR	R-0/W1S	0h	DC キャプチャ・ラッチ・ステータス・クリア・フラグ 0 : 0 を書き込んでも何もしません。 1 : 1 を書き込むと、この CAPSTS (セット) 条件はクリアされます。 リセットの種類 : SYSRSn
13	CAPSTS	R	0h	キャプチャ・イベントのラッチ・ステータス・フラグ 0 : DC キャプチャ・イベントは発生していません。 1 : DC キャプチャ・イベントが発生しました。 リセットの種類 : SYSRSn
12-2	RESERVED	R-0	0h	予約済み

表 1-103. DCCAPCTL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
1	SHDWMODE	R/W	0h	TBCTR カウンタ・キャプチャ・シャドウ選択モード 0 : シャドウ・モードを有効化します。TBCTR = TBPRD または TBCTR = ゼロ・イベント時に (DCFCTL[PULSESEL] ビットの設定に従って)、DCCAP アクティブ・レジスタはシャドウ・レジスタにコピーされます。DCCAP レジスタの CPU 読み出しは、シャドウ・レジスタの内容を返します。 1 : アクティブ・モード。このモードでは、シャドウ・レジスタは無効化されます。DCCAP レジスタからの CPU 読み出しは、常にアクティブ・レジスタの内容を返します。 リセットの種類 : SYSRSn
0	CAPE	R/W	0h	TBCTR カウンタ・キャプチャの有効化 / 無効化 0 : 時間ベース・カウンタのキャプチャを無効化します。 1 : 時間ベース・カウンタのキャプチャを有効化します。 リセットの種類 : SYSRSn

1.17.2.80 DCFOFFSET レジスタ (オフセット = C9h) [リセット = 0h]

図 1-172 に、DCFOFFSET を示し、表 1-104 に、その説明を示します。

概略表に戻ります。

デジタル比較フィルタ・ オフセット・ レジスタ

図 1-172. DCFOFFSET レジスタ

15	14	13	12	11	10	9	8
DCFOFFSET							
R/W-0h							
7	6	5	4	3	2	1	0
DCFOFFSET							
R/W-0h							

表 1-104. DCFOFFSET レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	DCFOFFSET	R/W	0h	ブランキング・ ウィンドウ・ オフセット これらの 16 ビットは、ブランキング・ ウィンドウ基準からブランキング・ ウィンドウが適用される時点までの TBCLK サイクル数を指定します。ブランキング・ ウィンドウ基準は周期とゼロのどちらかです (DCFCTL[PULSESEL] ビットの設定に従います)。このオフセット・ レジスタはシャドウイングされ、アクティブ・ レジスタは、DCFCTL[PULSESEL] で設定された基準点でロードされます。オフセット・ カウンタも初期化され、アクティブ・ レジスタがロードされると、カウンタ・ ダウンし始めます。カウンタが満了すると、ブランキング・ ウィンドウが適用されます。ブランキング・ ウィンドウが現在アクティブな場合、ブランキング・ ウィンドウ・ カウンタが再スタートします。 リセットの種類 : SYSRSn

1.17.2.81 DCFOFFSETCNT レジスタ (オフセット = CAh) [リセット = 0h]

図 1-173 に、DCFOFFSETCNT を示し、表 1-105 に、その説明を示します。

概略表に戻ります。

デジタル比較フィルタ・ オフセット・ カウンタ・ レジスタ

図 1-173. DCFOFFSETCNT レジスタ

15	14	13	12	11	10	9	8
DCFOFFSETCNT							
R-0h							
7	6	5	4	3	2	1	0
DCFOFFSETCNT							
R-0h							

表 1-105. DCFOFFSETCNT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	DCFOFFSETCNT	R	0h	ブランキング・ オフセット・ カウンタ これらの 16 ビットは読み出し専用であり、オフセット・ カウンタの現在の値を示しています。このカウンタはゼロまでカウントされた後、次の周期に再びロードされるまで、またはゼロ・ イベントまで停止します (DCCTL[PULSESEL] ビットの設定に従います)。オフセット・ カウンタはフリー/ソフト・ エミュレーション・ ビットの影響を受けません。つまり、エミュレーション停止によって本デバイスが停止した場合、常にカウント・ ダウンし続けます。 リセットの種類: SYSRSn

1.17.2.82 DCFWINDOW レジスタ (オフセット = CBh) [リセット = 0h]

図 1-174 に、DCFWINDOW を示し、表 1-106 に、その説明を示します。

概略表に戻ります。

デジタル比較フィルタ・ウィンドウ・レジスタ

図 1-174. DCFWINDOW レジスタ

15	14	13	12	11	10	9	8
DCFWINDOW							
R/W-0h							
7	6	5	4	3	2	1	0
DCFWINDOW							
R/W-0h							

表 1-106. DCFWINDOW レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	DCFWINDOW	R/W	0h	ブランキング・ウィンドウ幅 00h : ブランキング・ウィンドウは生成されません。 01-FFFFh : TBCLK サイクル内のブランキング・ウィンドウの幅を指定します。オフセット・カウンタが満了すると、ブランキング・ウィンドウが始まります。このような状況が発生すると、ウィンドウ・カウンタはロードされ、カウント・ダウンを始めます。ブランキング・ウィンドウが現在アクティブであり、オフセット・カウンタが満了した場合、ブランキング・ウィンドウ・カウンタは再スタートせず、ブランキング・ウィンドウは通常より早く終了します。この状況を避けるために注意を払う必要があります。ブランキング・ウィンドウは PWM 周期の境界を越える可能性があります。 リセットの種類 : SYSRSn

1.17.2.83 DCFWINDOWCNT レジスタ (オフセット = CCh) [リセット = 0h]

図 1-175 に、DCFWINDOWCNT を示し、表 1-107 に、その説明を示します。

[概略表](#)に戻ります。

デジタル比較フィルタ・ ウィンドウ・ カウンタ・ レジスタ

図 1-175. DCFWINDOWCNT レジスタ

15	14	13	12	11	10	9	8
DCFWINDOWCNT							
R-0h							
7	6	5	4	3	2	1	0
DCFWINDOWCNT							
R-0h							

表 1-107. DCFWINDOWCNT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	DCFWINDOWCNT	R	0h	ブランキング・ ウィンドウ・ カウンタ この 16 ビットは読み出し専用であり、ウィンドウ・ カウンタの現在の値を示します。このカウンタはゼロまでカウントされた後、オフセット・ カウンタが再度ゼロに達して再びロードされるまで停止します。 リセットの種類 : SYSRSn

1.17.2.84 BLANKPULSEMIXSEL レジスタ (オフセット = CDh) [リセット = 0h]

図 1-176 に、BLANKPULSEMIXSEL を示し、表 1-108 に、その説明を示します。

概略表に戻ります。

ブランキング・ウィンドウ・トリガ・パルス選択レジスタ

図 1-176. BLANKPULSEMIXSEL レジスタ

15		14		13		12		11		10		9		8	
RESERVED												CDD		CDU	
R-0-0h												R/W-0h		R/W-0h	
7		6		5		4		3		2		1		0	
CCD		CCU		CBD		CBU		CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

表 1-108. BLANKPULSEMIXSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	RESERVED	R-0	0h	予約済み
9	CDD	R/W	0h	ブランキング・ウィンドウ・トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPD と一致 (そのタイムがデクリメントしている場合)」を有効化します。 0 : CMPD ダウン・カウンタ一致イネーブル・イベントを無効化します。 1 : CMPD ダウン・カウンタ一致イネーブル・イベントを有効化します。 リセットの種類 : SYSRSn
8	CDU	R/W	0h	ブランキング・ウィンドウ・トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPD と一致 (そのタイムがインクリメントしている場合)」を有効化します。 0 : CMPD アップ・カウンタ一致イネーブル・イベントを無効化します。 1 : CMPD アップ・カウンタ一致イネーブル・イベントを有効化します。 リセットの種類 : SYSRSn
7	CCD	R/W	0h	ブランキング・ウィンドウ・トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPC と一致 (そのタイムがデクリメントしている場合)」を有効化します。 0 : CMPC ダウン・カウンタ一致イネーブル・イベントを無効化します。 1 : CMPC ダウン・カウンタ一致イネーブル・イベントを有効化します。 リセットの種類 : SYSRSn
6	CCU	R/W	0h	ブランキング・ウィンドウ・トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPC と一致 (そのタイムがインクリメントしている場合)」を有効化します。 0 : CMPC アップ・カウンタ一致イネーブル・イベントを無効化します。 1 : CMPC アップ・カウンタ一致イネーブル・イベントを有効化します。 リセットの種類 : SYSRSn
5	CBD	R/W	0h	ブランキング・ウィンドウ・トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPB と一致 (そのタイムがデクリメントしている場合)」を有効化します。 0 : CMPB ダウン・カウンタ一致イネーブル・イベントを無効化します。 1 : CMPB ダウン・カウンタ一致イネーブル・イベントを有効化します。 リセットの種類 : SYSRSn

表 1-108. BLANKPULSEMIXSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
4	CBU	R/W	0h	混合 ET 割り込みトリガ信号 (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPB と一致 (そのタイマがインクリメントしている場合)」を有効化します。 0: CMPB アップ・カウンタ一致イネーブル・ イベントを無効化します。 1: CMPB アップ・カウンタ一致イネーブル・ イベントを有効化します。 リセットの種類: SYSRSn
3	CAD	R/W	0h	ブランキング・ ウィンドウ・ トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがデクリメントしている場合)」を有効化します。 0: CMPA ダウン・カウンタ一致イネーブル・ イベントを無効化します。 1: CMPA ダウン・カウンタ一致イネーブル・ イベントを有効化します。 リセットの種類: SYSRSn
2	CAU	R/W	0h	ブランキング・ ウィンドウ・ トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが CMPA と一致 (そのタイマがインクリメントしている場合)」を有効化します。 0: CMPA アップ・カウンタ一致イネーブル・ イベントを無効化します。 1: CMPA アップ・カウンタ一致イネーブル・ イベントを無効化します。 リセットの種類: SYSRSn
1	PRD	R/W	0h	ブランキング・ ウィンドウ・ トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタが周期と一致 (TBCTR = TBPRD)」を有効化します。 0: 周期一致イベントを無効化します。 1: 周期一致イベントを有効化します。 リセットの種類: SYSRSn
0	ZRO	R/W	0h	ブランキング・ ウィンドウ・ トリガ (BLANKPULSEMIX) に対して、イベント「時間ベース・カウンタがゼロと一致 (TBCTR = 0x00)」を有効化します。 0: ゼロ一致イベントを無効化します。 1: ゼロ一致イベントを有効化します。 リセットの種類: SYSRSn

1.17.2.85 DCCAP レジスタ (オフセット = CFh) [リセット = 0h]

図 1-177 に、DCCAP を示し、表 1-109 に、その説明を示します。

概略表に戻ります。

デジタル比較カウンタ・キャプチャ・レジスタ

図 1-177. DCCAP レジスタ

15	14	13	12	11	10	9	8
DCCAP							
R-0h							
7	6	5	4	3	2	1	0
DCCAP							
R-0h							

表 1-109. DCCAP レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	DCCAP	R	0h	デジタル比較時間ベース・カウンタ・キャプチャ時間ベース・カウンタ・キャプチャを有効化するには、DCCAPCLT[CAPE] ビットを 1 に設定します。有効化されている場合、フィルタ処理済み (DCEVTFLT) イベントの Low から High へのエッジ遷移時に時間ベース・カウンタ (TBCTR) の値を反映します。次の周期またはゼロまで (DCFCTL[PULSESEL] ビットによる選択に従って)、さらなるキャプチャ・イベントは無視されます。DCCAP のシャドウイングは、DCCAPCTL[SHDWMODE] ビットによって有効化および無効化されます。デフォルトでは、このレジスタはシャドウイングされます。 - DCCAPCTL[SHDWMODE] = 0 の場合、シャドウイングは有効化されます。このモードでは、TBCTR = TBPRD または TBCTR = ゼロ時に (DCFCTL[PULSESEL] ビットの設定に従って)、アクティブ・レジスタはシャドウ・レジスタにコピーされます。このレジスタの CPU 読み出しはシャドウ・レジスタ値を返します。 - DCCAPCTL[SHDWMODE] = 1 の場合、シャドウ・レジスタは無効化されます。このモードでは、CPU 読み出しはアクティブ・レジスタ値を返します。アクティブ・レジスタとシャドウ・レジスタは、同じメモリ・マップ・アドレスを共有します。 リセットの種類 : SYSRSn

1.17.2.86 DCAHTRIPSEL レジスタ (オフセット = D2h) [リセット = 0h]

図 1-178 に、DCAHTRIPSEL を示し、表 1-110 に、その説明を示します。

概略表に戻ります。

デジタル比較 AH トリップ選択

図 1-178. DCAHTRIPSEL レジスタ

15		14		13		12		11		10		9		8	
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9								
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h								
7		6		5		4		3		2		1		0	
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1								
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h								

表 1-110. DCAHTRIPSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	TRIPINPUT15	R/W	0h	トリップ入力 15 0: 組み合わせ論理和入力としてトリップ入力 15 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 15 を選択します。 リセットの種類: SYSRSn
13	TRIPINPUT14	R/W	0h	トリップ入力 14 0: 組み合わせ論理和入力としてトリップ入力 14 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 14 を選択します。 リセットの種類: SYSRSn
12	RESERVED	R/W	0h	予約済み
11	TRIPINPUT12	R/W	0h	トリップ入力 12 0: 組み合わせ論理和入力としてトリップ入力 12 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 12 を選択します。 リセットの種類: SYSRSn
10	TRIPINPUT11	R/W	0h	トリップ入力 11 0: 組み合わせ論理和入力としてトリップ入力 11 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 11 を選択します。 リセットの種類: SYSRSn
9	TRIPINPUT10	R/W	0h	トリップ入力 10 0: 組み合わせ論理和入力としてトリップ入力 10 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 10 を選択します。 リセットの種類: SYSRSn
8	TRIPINPUT9	R/W	0h	トリップ入力 9 0: 組み合わせ論理和入力としてトリップ入力 9 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 9 を選択します。 リセットの種類: SYSRSn
7	TRIPINPUT8	R/W	0h	トリップ入力 8 0: 組み合わせ論理和入力としてトリップ入力 8 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 8 を選択します。 リセットの種類: SYSRSn
6	TRIPINPUT7	R/W	0h	トリップ入力 7 0: 組み合わせ論理和入力としてトリップ入力 7 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 7 を選択します。 リセットの種類: SYSRSn

表 1-110. DCAHTRIPSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
5	TRIPINPUT6	R/W	0h	トリップ入力 6 0: 組み合わせ論理和入力としてトリップ入力 6 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 6 を選択します。 リセットの種類: SYSRSn
4	TRIPINPUT5	R/W	0h	トリップ入力 5 0: 組み合わせ論理和入力としてトリップ入力 5 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 5 を選択します。 リセットの種類: SYSRSn
3	TRIPINPUT4	R/W	0h	トリップ入力 4 0: 組み合わせ論理和入力としてトリップ入力 4 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 4 を選択します。 リセットの種類: SYSRSn
2	TRIPINPUT3	R/W	0h	トリップ入力 3 0: 組み合わせ論理和入力としてトリップ入力 3 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 3 を選択します。 リセットの種類: SYSRSn
1	TRIPINPUT2	R/W	0h	トリップ入力 2 0: 組み合わせ論理和入力としてトリップ入力 2 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 2 を選択します。 リセットの種類: SYSRSn
0	TRIPINPUT1	R/W	0h	トリップ入力 1 0: 組み合わせ論理和入力としてトリップ入力 1 を選択しません。 1: DCAH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 1 を選択します。 リセットの種類: SYSRSn

1.17.2.87 DCALTRIPSEL レジスタ (オフセット = D3h) [リセット = 0h]

図 1-179 に、DCALTRIPSEL を示し、表 1-111 に、その説明を示します。

概略表に戻ります。

デジタル比較 AL トリップ選択

図 1-179. DCALTRIPSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-111. DCALTRIPSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	TRIPINPUT15	R/W	0h	トリップ入力 15 0: 組み合わせ論理和入力としてトリップ入力 15 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 15 を選択します。 リセットの種類: SYSRSn
13	TRIPINPUT14	R/W	0h	トリップ入力 14 0: 組み合わせ論理和入力としてトリップ入力 14 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 14 を選択します。 リセットの種類: SYSRSn
12	RESERVED	R/W	0h	予約済み
11	TRIPINPUT12	R/W	0h	トリップ入力 12 0: 組み合わせ論理和入力としてトリップ入力 12 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 12 を選択します。 リセットの種類: SYSRSn
10	TRIPINPUT11	R/W	0h	トリップ入力 11 0: 組み合わせ論理和入力としてトリップ入力 11 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 11 を選択します。 リセットの種類: SYSRSn
9	TRIPINPUT10	R/W	0h	トリップ入力 10 0: 組み合わせ論理和入力としてトリップ入力 10 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 10 を選択します。 リセットの種類: SYSRSn
8	TRIPINPUT9	R/W	0h	トリップ入力 9 0: 組み合わせ論理和入力としてトリップ入力 9 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 9 を選択します。 リセットの種類: SYSRSn
7	TRIPINPUT8	R/W	0h	トリップ入力 8 0: 組み合わせ論理和入力としてトリップ入力 8 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 8 を選択します。 リセットの種類: SYSRSn
6	TRIPINPUT7	R/W	0h	トリップ入力 7 0: 組み合わせ論理和入力としてトリップ入力 7 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 7 を選択します。 リセットの種類: SYSRSn

表 1-111. DCALTRIPSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
5	TRIPINPUT6	R/W	0h	トリップ入力 6 0: 組み合わせ論理和入力としてトリップ入力 6 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 6 を選択します。 リセットの種類: SYSRSn
4	TRIPINPUT5	R/W	0h	トリップ入力 5 0: 組み合わせ論理和入力としてトリップ入力 5 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 5 を選択します。 リセットの種類: SYSRSn
3	TRIPINPUT4	R/W	0h	トリップ入力 4 0: 組み合わせ論理和入力としてトリップ入力 4 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 4 を選択します。 リセットの種類: SYSRSn
2	TRIPINPUT3	R/W	0h	トリップ入力 3 0: 組み合わせ論理和入力としてトリップ入力 3 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 3 を選択します。 リセットの種類: SYSRSn
1	TRIPINPUT2	R/W	0h	トリップ入力 2 0: 組み合わせ論理和入力としてトリップ入力 2 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 2 を選択します。 リセットの種類: SYSRSn
0	TRIPINPUT1	R/W	0h	トリップ入力 1 0: 組み合わせ論理和入力としてトリップ入力 1 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 1 を選択します。 リセットの種類: SYSRSn

1.17.2.88 DCBHTRIPSEL レジスタ (オフセット = D4h) [リセット = 0h]

図 1-180 に、DCBHTRIPSEL を示し、表 1-112 に、その説明を示します。

概略表に戻ります。

デジタル比較 BH トリップ選択

図 1-180. DCBHTRIPSEL レジスタ

15	14	13	12	11	10	9	8
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-112. DCBHTRIPSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	TRIPINPUT15	R/W	0h	トリップ入力 15 0: 組み合わせ論理和入力としてトリップ入力 15 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 15 を選択します。 リセットの種類: SYSRSn
13	TRIPINPUT14	R/W	0h	トリップ入力 14 0: 組み合わせ論理和入力としてトリップ入力 14 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 14 を選択します。 リセットの種類: SYSRSn
12	RESERVED	R/W	0h	予約済み
11	TRIPINPUT12	R/W	0h	トリップ入力 12 0: 組み合わせ論理和入力としてトリップ入力 12 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 12 を選択します。 リセットの種類: SYSRSn
10	TRIPINPUT11	R/W	0h	トリップ入力 11 0: 組み合わせ論理和入力としてトリップ入力 11 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 11 を選択します。 リセットの種類: SYSRSn
9	TRIPINPUT10	R/W	0h	トリップ入力 10 0: 組み合わせ論理和入力としてトリップ入力 10 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 10 を選択します。 リセットの種類: SYSRSn
8	TRIPINPUT9	R/W	0h	トリップ入力 9 0: 組み合わせ論理和入力としてトリップ入力 9 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 9 を選択します。 リセットの種類: SYSRSn
7	TRIPINPUT8	R/W	0h	トリップ入力 8 0: 組み合わせ論理和入力としてトリップ入力 8 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 8 を選択します。 リセットの種類: SYSRSn
6	TRIPINPUT7	R/W	0h	トリップ入力 7 0: 組み合わせ論理和入力としてトリップ入力 7 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 7 を選択します。 リセットの種類: SYSRSn

表 1-112. DCBHTRIPSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
5	TRIPINPUT6	R/W	0h	トリップ入力 6 0: 組み合わせ論理和入力としてトリップ入力 6 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 6 を選択します。 リセットの種類: SYSRSn
4	TRIPINPUT5	R/W	0h	トリップ入力 5 0: 組み合わせ論理和入力としてトリップ入力 5 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 5 を選択します。 リセットの種類: SYSRSn
3	TRIPINPUT4	R/W	0h	トリップ入力 4 0: 組み合わせ論理和入力としてトリップ入力 4 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 4 を選択します。 リセットの種類: SYSRSn
2	TRIPINPUT3	R/W	0h	トリップ入力 3 0: 組み合わせ論理和入力としてトリップ入力 3 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 3 を選択します。 リセットの種類: SYSRSn
1	TRIPINPUT2	R/W	0h	トリップ入力 2 0: 組み合わせ論理和入力としてトリップ入力 2 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 2 を選択します。 リセットの種類: SYSRSn
0	TRIPINPUT1	R/W	0h	トリップ入力 1 0: 組み合わせ論理和入力としてトリップ入力 1 を選択しません。 1: DCBH マルチプレクサへの組み合わせ論理和入力としてトリップ入力 1 を選択します。 リセットの種類: SYSRSn

1.17.2.89 DCBLTRIPSEL レジスタ (オフセット = D5h) [リセット = 0h]

図 1-181 に、DCBLTRIPSEL を示し、表 1-113 に、その説明を示します。

概略表に戻ります。

デジタル比較 BL トリップ選択

図 1-181. DCBLTRIPSEL レジスタ

15		14		13		12		11		10		9		8	
RESERVED	TRIPINPUT15	TRIPINPUT14	RESERVED	TRIPINPUT12	TRIPINPUT11	TRIPINPUT10	TRIPINPUT9								
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h								
7		6		5		4		3		2		1		0	
TRIPINPUT8	TRIPINPUT7	TRIPINPUT6	TRIPINPUT5	TRIPINPUT4	TRIPINPUT3	TRIPINPUT2	TRIPINPUT1								
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h								

表 1-113. DCBLTRIPSEL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RESERVED	R	0h	予約済み
14	TRIPINPUT15	R/W	0h	トリップ入力 15 0: 組み合わせ論理和入力としてトリップ入力 15 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 15 を選択します。 リセットの種類: SYSRSn
13	TRIPINPUT14	R/W	0h	トリップ入力 14 0: 組み合わせ論理和入力としてトリップ入力 14 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 14 を選択します。 リセットの種類: SYSRSn
12	RESERVED	R/W	0h	予約済み
11	TRIPINPUT12	R/W	0h	トリップ入力 12 0: 組み合わせ論理和入力としてトリップ入力 12 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 12 を選択します。 リセットの種類: SYSRSn
10	TRIPINPUT11	R/W	0h	トリップ入力 11 0: 組み合わせ論理和入力としてトリップ入力 11 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 11 を選択します。 リセットの種類: SYSRSn
9	TRIPINPUT10	R/W	0h	トリップ入力 10 0: 組み合わせ論理和入力としてトリップ入力 10 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 10 を選択します。 リセットの種類: SYSRSn
8	TRIPINPUT9	R/W	0h	トリップ入力 9 0: 組み合わせ論理和入力としてトリップ入力 9 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 9 を選択します。 リセットの種類: SYSRSn
7	TRIPINPUT8	R/W	0h	トリップ入力 8 0: 組み合わせ論理和入力としてトリップ入力 8 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 8 を選択します。 リセットの種類: SYSRSn
6	TRIPINPUT7	R/W	0h	トリップ入力 7 0: 組み合わせ論理和入力としてトリップ入力 7 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 7 を選択します。 リセットの種類: SYSRSn

表 1-113. DCBLTRIPSEL レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
5	TRIPINPUT6	R/W	0h	トリップ入力 6 0: 組み合わせ論理和入力としてトリップ入力 6 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 6 を選択します。 リセットの種類: SYSRSn
4	TRIPINPUT5	R/W	0h	トリップ入力 5 0: 組み合わせ論理和入力としてトリップ入力 5 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 5 を選択します。 リセットの種類: SYSRSn
3	TRIPINPUT4	R/W	0h	トリップ入力 4 0: 組み合わせ論理和入力としてトリップ入力 4 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 4 を選択します。 リセットの種類: SYSRSn
2	TRIPINPUT3	R/W	0h	トリップ入力 3 0: 組み合わせ論理和入力としてトリップ入力 3 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 3 を選択します。 リセットの種類: SYSRSn
1	TRIPINPUT2	R/W	0h	トリップ入力 2 0: 組み合わせ論理和入力としてトリップ入力 2 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 2 を選択します。 リセットの種類: SYSRSn
0	TRIPINPUT1	R/W	0h	トリップ入力 1 0: 組み合わせ論理和入力としてトリップ入力 1 を選択しません。 1: DCAL マルチプレクサへの組み合わせ論理和入力としてトリップ入力 1 を選択します。 リセットの種類: SYSRSn

1.17.2.90 EPWMLOCK レジスタ (オフセット = FAh) [リセット = 0h]

図 1-182 に、EPWMLOCK を示し、表 1-114 に、その説明を示します。

概略表に戻ります。

EPWM ロック・レジスタ

図 1-182. EPWMLOCK レジスタ

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			DCLOCK	TZCLRLOCK	TZCFGLOCK	GLLOCK	HRLOCK
R-0h			R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h	R/WOnce-0h

表 1-114. EPWMLOCK レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
31-16	KEY	R-0/W	0h	このレジスタへの書き込みは、このフィールドに 0xa5a5 の値が書き込まれた場合にのみ成功します。 注： [1] この KEY のおかげで、32 ビット書き込みのみが成功します (KEY が一致する場合)。このレジスタの上位半分または下位半分への 16 ビット書き込みは無視されます。 リセットの種類：SYSRSn
15-5	RESERVED	R	0h	予約済み
4	DCLOCK	R/WOnce	0h	0：デジタル比較レジスタ 0xC0～0xD9 のオフセットは EALLOW によって保護されます。 1：デジタル比較レジスタ 0xC0～0xD9 のオフセットはロックされており、書き込むことはできません。 リセットの種類：SYSRSn
3	TZCLRLOCK	R/WOnce	0h	0：トリップ・ゾーン・レジスタ 0x97～0x9B のオフセットは EALLOW によって保護されます。 1：トリップ・ゾーン・レジスタ 0x97～0x9B のオフセットはロックされており、書き込むことはできません。 リセットの種類：SYSRSn
2	TZCFGLOCK	R/WOnce	0h	0：トリップ・ゾーン・レジスタ 0x80～0x8D と TZTRIPOUTSEL 0x9D のオフセットは EALLOW によって保護されます。 1：トリップ・ゾーン・レジスタ 0x80～0x8D と TZTRIPOUTSEL 0x9D のオフセットはロックされており、書き込むことはできません。 リセットの種類：SYSRSn
1	GLLOCK	R/WOnce	0h	0：グローバル・ロード・レジスタ 0x34～0x35 のオフセットは EALLOW によって保護されます。 1：グローバル・ロード・レジスタ 0x34～0x35 のオフセットはロックされており、書き込むことはできません。 リセットの種類：SYSRSn

表 1-114. EPWMLOCK レジスタのフィールドの説明 (続き)

ビット	フィールド	タイプ	リセット	説明
0	HRLOCK	R/WOnce	0h	0 : HRPWM レジスタ 0x20 ~ 0x2D のオフセットは EALLOW によって保護されます。 1 : HRPWM レジスタ 0x20 ~ 0x2D のオフセットはロックされており、書き込むことはできません。 リセットの種類 : SYSRSn

1.17.2.91 HWVDELVAL レジスタ (オフセット = FDh) [リセット = 0h]

図 1-183 に、HWVDELVAL を示し、表 1-115 に、その説明を示します。

[概略表](#)に戻ります。

ハードウェア・バレー・モード遅延レジスタ

図 1-183. HWVDELVAL レジスタ

15	14	13	12	11	10	9	8
HWVDELVAL							
R-0h							
7	6	5	4	3	2	1	0
HWVDELVAL							
R-0h							

表 1-115. HWVDELVAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	HWVDELVAL	R	0h	ハードウェア・バレー遅延値レジスタ この読み出し専用レジスタは、VCAPCTL[VDELAYDIV] で定義された式で計算されたハードウェア遅延値を反映します。この値はハードウェア計算による最新の値を反映しており、バレー・キャプチャ・シーケンスがトリガされ、VCAP1 および VCAP2 値が更新されるたびに变化する可能性があります。 リセットの種類：SYSRSn

1.17.2.92 VCNTVAL レジスタ (オフセット = FEh) [リセット = 0h]

図 1-184 に、VCNTVAL を示し、表 1-116 に、その説明を示します。

概略表に戻ります。

ハードウェア・バレー・カウンタ・レジスタ

図 1-184. VCNTVAL レジスタ

15	14	13	12	11	10	9	8
VCNTVAL							
R-0h							
7	6	5	4	3	2	1	0
VCNTVAL							
R-0h							

表 1-116. VCNTVAL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	VCNTVAL	R	0h	バレー時間ベース・カウンタ・レジスタ このレジスタは、VCNTCFG レジスタで選択された STOPEDGE が発生した際にキャプチャされる VCNT 値を反映します。 リセットの種類：SYRSn

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020 – REVISED MARCH 2023



この章では、I2C (Inter-Integrated Circuit) モジュールの機能と動作について説明します。I2C (Inter-Integrated Circuit) モジュールは、これらのデバイスの 1 つと、他のデバイスとの間のインターフェイスとして機能します。他のデバイスは、NXP Semiconductors の I2C バス (Inter-IC Bus) 規格バージョン 2.1 に準拠し、I2C バス経由で接続されている必要があります。この 2 線式シリアル・バスに接続されている外部コンポーネントは、I2C モジュールによって、デバイスとの間で 1 ビットから 8 ビットまでのデータを送受信できます。この章では、読者が I2C バスの仕様を理解していることを想定しています。

注

I2C モジュールで送受信されるデータの単位は 8 ビットより短いこともありますが、利便性のため、このドキュメント全体を通してデータの単位をデータ・バイトと呼びます。データ・バイトのビット数は、MODE レジスタ I2CMDR の BC ビットを使用して選択できます。

1.1 はじめに.....	4
1.2 デバイス・ピンの構成.....	9
1.3 I2C モジュールの動作の詳細.....	9
1.4 I2C モジュールにより生成される割り込み要求.....	21
1.5 I2C モジュールのリセットまたはディセーブル.....	24
1.6 ソフトウェア.....	25
1.7 I2C レジスタ.....	27

1.1 はじめに

I2C モジュールは、スレーブまたはマスタ I2C 互換のあらゆるデバイスをサポートします。あるデバイスから他のデバイスに双方向転送を行うように接続された複数の I2C モジュールの例を、[図 1-1](#) に示します。

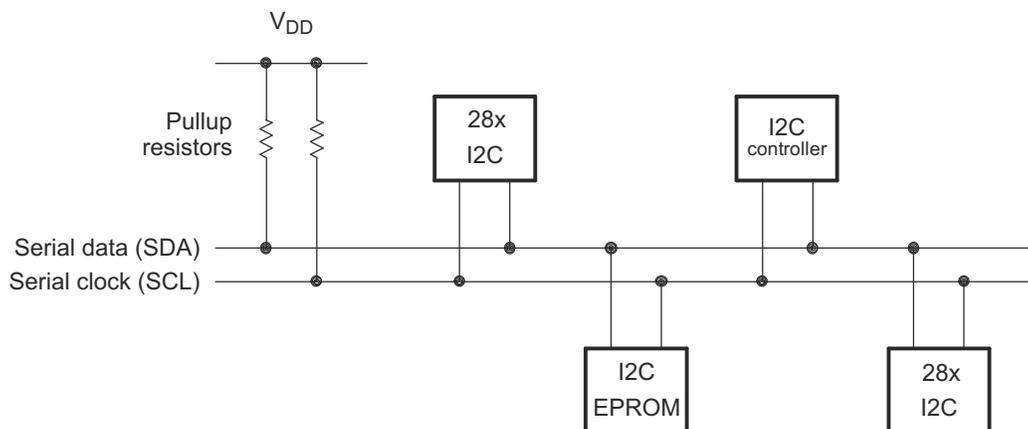


図 1-1. 複数の I2C モジュールが接続された状態

1.1.1 I2C 関連の資料

基本資料

- [『C2000 アカデミー - I2C』](#)
- [『I2C ハードウェアの概要』](#) (ビデオ)
- [『I2C プロトコルの概要』](#) (ビデオ)
- [『I2C バス・アプリケーション・レポートについて』](#)

入門用資料

- [『TMS320F280x DSP を I2C プロセッサとして構成する方法』](#)アプリケーション・レポート
- [『I2C バッファの概要』](#) (ビデオ)
- [『I2C の動的アドレッシング』](#)アプリケーション・レポート
- [『I2C トランスレータの概要』](#) (ビデオ)
- [『C2000 I2C モジュールを使用した EEPROM のインターフェイス』](#)アプリケーション・レポート
- [『I2C バッファを使用する理由、条件、方法』](#)アプリケーション・レポート

高度な資料

- [『I2C バスのプルアップ抵抗の計算』](#)アプリケーション・レポート
- [『リピータを使用する I2C バスの最高クロック周波数』](#)アプリケーション・レポート

1.1.2 特長

I2C モジュールの主な機能は次のとおりです。

- NXP Semiconductors の I2C バス仕様 (バージョン 2.1) に準拠:
 - 8 ビット形式の転送をサポート
 - 7 ビットおよび 10 ビットのアドレッシング・モード
 - ゼネラル・コール
 - START バイト・モード
 - 複数のマスタ・トランスミッタとスレーブ・レシーバをサポート
 - 複数のスレーブ・トランスミッタとマスタ・レシーバをサポート
 - マスタ送信 / 受信および受信 / 送信の組み合わせモード
 - 10kbps～最大 400kbps (ファスト・モード) のデータ転送レートをサポート
- 受信 FIFO およびトランスミッタ FIFO (深さ 16 × 8 ビット FIFO)
- 2 つの ePIE 割り込みをサポート
 - I2Cx 割り込み - 次のいずれかのイベントで I2Cx 割り込みを生成するように構成できます。
 - 送信データ準備完了
 - 受信データ準備完了
 - レジスタ・アクセス・レディ
 - アクノリッジ受信なし
 - アービトレーション・ロスト
 - ストップ条件検出
 - スレーブとしてアドレス指定
 - I2Cx_FIFO 割り込み:
 - 送信 FIFO 割り込み
 - 受信 FIFO 割り込み
- モジュールのイネーブルおよびディセーブル機能
- フリー・データ形式モード

1.1.3 サポート対象外の機能

I2C モジュールは次の機能をサポートしていません。

- ハイスピード・モード (HS)
- CBUS 互換モード

1.1.4 機能概要

I2C バスに接続された各デバイスは、一意のアドレスで認識されます。各デバイスは、その機能に応じて、トランスミッタまたはレシーバとして動作できます。I2C バスに接続されているデバイスは、データ転送を実行するときに、マスタまたはスレーブとも考えることができます。マスタ・デバイスは、バス上でデータ転送を開始し、その転送を許可するクロック信号を生成するデバイスです。この転送中に、このマスタによってアドレス指定されるデバイスは、スレーブとみなされます。I2C モジュールは、マルチマスタ・モードをサポートしており、I2C バスの制御能力を持つデバイスを同じ I2C バスに複数接続できます。

I2C モジュールには、[図 1-2](#) に示すように、データ通信のシリアル・データ・ピン (SDA) とシリアル・クロック・ピン (SCL) があります。これら 2 本のピンは、C28x デバイスと、I2C バスに接続されている他のデバイスとの間で情報を伝送します。SDA ピンと SCL ピンはどちらも双方向で、プルアップ抵抗で正の電源電圧に接続する必要があります。バスが解放されると、両方のピンが High になります。これら 2 つのピンのドライバは、必要な有線 AND 機能を実行するため、オープン・ドレイン構成になっています。

主な転送方式は 2 つあります。

- 標準モード: 正確に n 個のデータ値を送信します。ここで、 n は I2C モジュール・レジスタでプログラムする値です。詳細については、[セクション 1.7](#) の I2CCNT レジスタを参照してください。
- 反復モード: ソフトウェアから STOP 条件または新しい START 条件が開始されるまで、データ値の送信を続けます。RM ビットについては、[セクション 1.7](#) の I2CMDR レジスタを参照してください。

I2C モジュールは、以下の主要ブロックで構成されています。

- シリアル・インターフェイス: 1 つのデータ・ピン (SDA) と 1 つのクロック・ピン (SCL)
- SDA ピンと CPU の間で伝送される受信データと送信データを一時的に保持するデータ・レジスタと FIFO
- 制御およびステータス・レジスタ
- CPU が I2C モジュールのレジスタと FIFO にアクセスするための、ペリフェラル・バス・インターフェイス。
- (デバイス・クロック・ジェネレータからの) I2C 入力クロックと SCL ピンのクロックを同期し、クロック速度の異なるマスタとのデータ転送を同期するクロック・シンクロナイザ
- I2C モジュールに駆動される入力クロックを分周するプリスケアラ
- SDA および SCL ピンの両方のノイズ・フィルタ
- I2C モジュール (I2C モジュールが マスタのとき) と他のマスタとの間で調停を処理するアービトラータ
- 割り込みを CPU に送信する割り込み生成ロジック
- I2C モジュールでのデータ受信およびデータ送信と FIFO アクセスとを同期するための FIFO 割り込み生成ロジック

FIFO 以外のモードでの送受信に使用される 4 つのレジスタを、[図 1-2](#) に示します。CPU は送信するデータを I2CDXR に書き込み、受信したデータを I2CDRR から読み取ります。I2C モジュールをトランスミッタとして構成すると、I2CDXR に書き込まれたデータが I2CXSR にコピーされ、SDA ピンで 1 ビットずつシフト・アウトされます。I2C モジュールをレシーバとして構成すると、受信データは I2CRSR にシフト・インされ、I2CDRR にコピーされます。

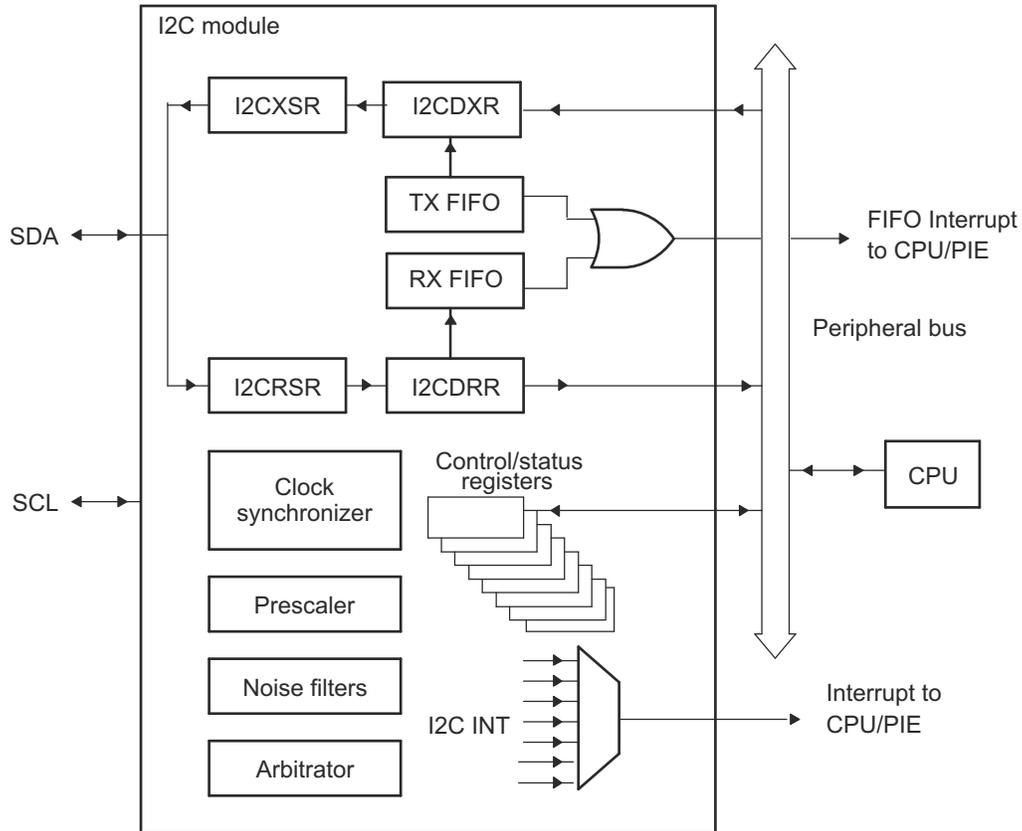


図 1-2. I2C モジュールの概念ブロック図

1.1.5 クロックの生成

I2C モジュールのクロックによって、I2C モジュールの動作周波数が決まります。I2C モジュールのプログラマブル・プリスケアラは、SYSCLK を分周して I2C モジュール・クロックを生成し、その I2C モジュール・クロックをさらに分周して SCL ピンに I2C マスタ・クロックを生成します。I2C モジュールのクロック生成図を、図 1-3 に示します。

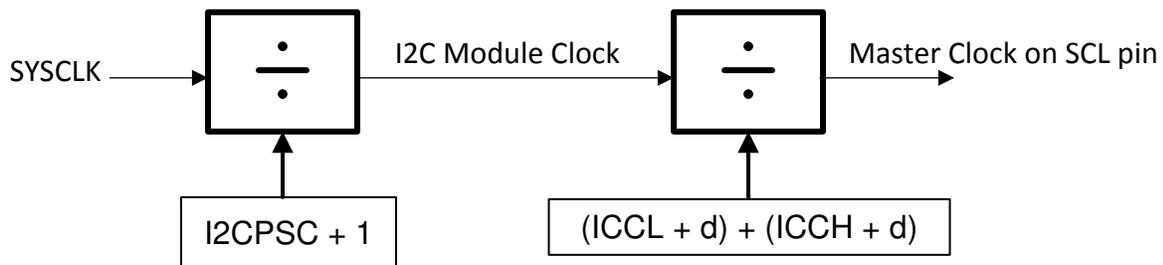


図 1-3. I2C モジュールのクロック図

注

すべての I2C プロトコルのタイミング仕様を満たすには、I2C モジュールのクロックが 7~12MHz の必要があります。

分周値を指定するには、プリスケアラ・レジスタ I2CPSC の IPSC フィールドを初期化します。得られる周波数は次のとおりです：

$$\text{I2C Module Clock (Fmod)} = \frac{\text{SYSCLK}}{(\text{I2CPSC} + 1)} \quad (1)$$

プリスケアラは、I2C モジュールが RESET 状態 (I2CMDR で IRS = 0) のときのみ初期化が必要です。プリスケールされた周波数は、IRS が 1 に変更されたときのみ有効になります。IRS = 1 で IPSC 値を変更しても効果はありません。

I2C モジュールが I2C バスでマスタになるよう構成されている場合、マスタ・クロックが SCL ピンに現れます。このクロックは、I2C モジュールとスレーブとの通信タイミングを制御します。図 1-3 に示すように、I2C モジュールの 2 番目のクロック・デバイダは、モジュールのクロックを分周して、マスタ・クロックを生成します。クロック・デバイダは、I2CCLKL の ICCL 値を使用してモジュール・クロック信号の Low 部分を分周し、I2CCLKH の ICCH 値を使用してモジュール・クロック信号の High 部分を分周します。マスタ・クロックの周波数の式については、[セクション 1.1.6](#) を参照してください。

1.1.6 I2C クロック・デバイダ・レジスタ (I2CCLKL および I2CCLKH)

[セクション 1.1.5](#) で説明されているように、I2C モジュールがマスタのとき、I2C モジュールのクロックは、SCL ピンで マスタ・クロックとして使用するためさらに分周されます。図 1-4 に示すように、マスタ・クロックの形状は、2 つの分周値によって決められます。

- I2CCLKL 内の ICCL。それぞれのマスタ・クロック・サイクルについて、信号が Low になる時間の長さは ICCL によって決定されます。
- I2CCLKH の ICCH。それぞれのマスタ・クロック・サイクルについて、信号が High になる時間の長さは ICCH によって決定されます。

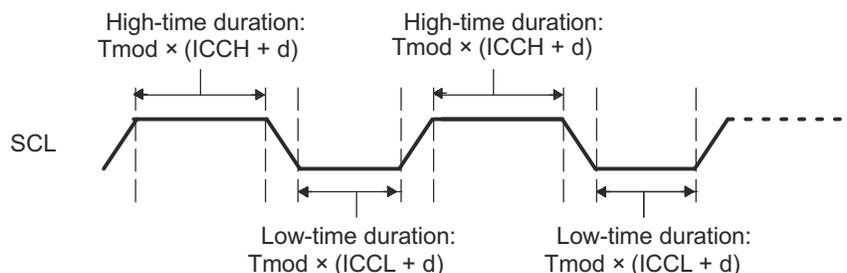


図 1-4. クロック分周値 (ICCL および ICCH) の役割

1.1.6.1 マスタ・クロック周期の式

マスタ・クロックの周期 (Tmst) は、I2C モジュール・クロックの周期 (Tmod) の倍数です。

$$\text{Master Clock period (Tmst)} = \frac{[(\text{ICCH} + d) + (\text{ICCL} + d)]}{\text{I2C Module Clock (Fmod)}} \quad (2)$$

ここで、d は表 1-1 に示すように、分周値 IPSC に依存します。IPSC は I2CPSC レジスタで指定されます。

表 1-1. 分周値 IPSC と遅延 d との関係

IPSC	d
0	7
1	6
1 より大きい	5

1.2 デバイス・ピンの構成

このペリフェラルをデバイスのピンに接続するには、GPIO マルチプレクサ・レジスタを構成する必要があります。ピンのグリッチを回避するには、まず GPyGMUX ビットを最初に設定し (対応する GPyMUX ビットはデフォルトの 0 のまま)、次に GPyMUX レジスタに目的の値を書き込む必要があります。

一部の IO 機能は、このペリフェラルとは独立した GPIO レジスタ設定によって定義されます。入力信号については、適切な GPxQSELn レジスタ・ビットを 11b に設定して、GPIO 入力認定を非同期モードに設定する必要があります。内部プルアップは GPyPUD レジスタで構成できます。

GPIO マルチプレクサと設定の詳細については、「GPIO」の章を参照してください。

1.3 I2C モジュールの動作の詳細

このセクションでは、I2C バス・プロトコルの概要と実装方法について説明します。

1.3.1 入力と出力の電圧レベル

転送されるデータ・ビットごとに、マスタ デバイスにより 1 つのクロック・パルスが生成されます。I2C バスには、さまざまなテクノロジーのデバイスが接続可能なため、ロジック 0 (Low) とロジック 1 (High) のレベルは固定されておらず、関連する V_{DD} のレベルに依存します。詳細については、デバイスのデータ・マニュアルを参照してください。

1.3.2 データの有効性

SDA のデータは、クロック信号 (SCL) の High 期間中は安定している必要があります (図 1-5 を参照)。データ・ライン SDA の High または Low の状態は、SCL のクロック信号が Low のときのみ変化します。

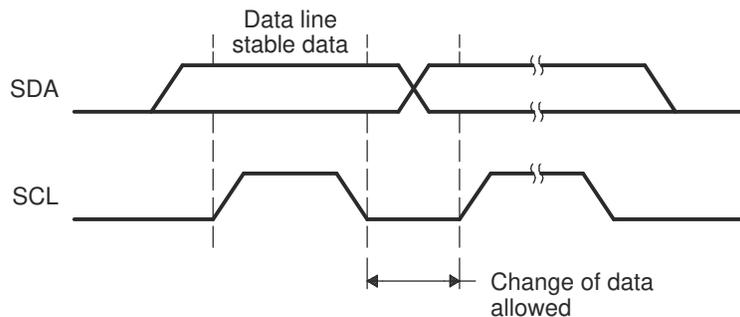


図 1-5. I2C バスでのビット転送

1.3.3 動作モード

I2C モジュールには 4 つの基本動作モードがあり、マスタおよびスレーブとしてのデータ転送をサポートします。モードの名前と説明については、表 1-2 を参照してください。

I2C モジュールがマスタなら、I2C モジュールはマスタ・トランスミッタとして開始され、通常は特定のスレーブのアドレスを送信します。スレーブにデータを渡すときは、I2C モジュールがマスタ・トランスミッタのまま維持される必要があります。スレーブからデータを受信するには、I2C モジュールがマスタ・レシーバ・モードに変更される必要があります。

I2C モジュールがスレーブなら、I2C モジュールはスレーブ・レシーバとして開始され、通常は I2C モジュールがマスタからのスレーブ・アドレスを認識したときにアクノリッジを送信します。マスタが I2C モジュールにデータを送信するなら、モジュールはスレーブ・レシーバのまま維持される必要があります。マスタが I2C モジュールからデータを要求したなら、モジュールがスレーブ・トランスミッタ・モードに変更される必要があります。

表 1-2. I2C モジュールの動作モード

動作モード	概要
スレーブ・レシーバ・モード	I2C モジュールはスレーブで、マスタからデータを受信します。 すべてのスレーブはこのモードで開始します。このモードでは、SDA で受信されたシリアル・データ・ビットが、マスタによって生成されたクロック・パルスとともにシフトインされます。I2C モジュールは、スレーブなので、クロック信号を生成しませんが、バイトが受信された後でデバイスの介入が必要な間 SCL を Low に保持できます (I2CSTR で RSFULL = 1)。詳細については、 セクション 1.3.7 を参照してください。
スレーブ・トランスミッタ・モード	I2C モジュールはスレーブで、マスタにデータを送信します。 このモードには、スレーブ・レシーバ・モードからのみ移行できます。I2C モジュールは最初に、マスタからのコマンドを受信する必要があります。7 ビット / 10 ビットのアドレッシング形式のいずれかを使用しているとき、スレーブ・アドレス・バイトが独自アドレス (I2COAR にあるもの) と同一で、マスタが R/W = 1 を送信すれば、I2C モジュールはスレーブ・トランスミッタ・モードに移行します。I2C モジュールはスレーブ・トランスミッタなので、マスタにより生成されたクロック・パルスを使用して、シリアル・データを SDA にシフト・アウトします。I2C モジュールは、スレーブのとき、クロック信号を生成しませんが、バイトが受信された後でデバイスの介入が必要な間 SCL を Low に保持できます (I2CSTR で XSMT = 0)。詳細については、 セクション 1.3.7 を参照してください。
マスタ・レシーバ・モード	I2C モジュールはマスタで、スレーブからデータを受信します。 このモードには、マスタ・トランスミッタ・モードからのみ移行できます。I2C モジュールは最初に、スレーブにコマンドを送信する必要があります。7 ビット / 10 ビットのアドレッシング形式のいずれかを使用しているとき、I2C モジュールはスレーブ・アドレス・バイトと R/W = 1 を送信した後でマスタ・レシーバ・モードに移行します。SDA 上のシリアル・データ・ビットは、I2C モジュールによって SCL に生成されるクロック・パルスを使用して、I2C モジュールにシフト・インされます。クロック・パルスは禁止され、バイトを受信した後にデバイスの介入が必要な場合 (I2CSTR で RSFULL = 1)、SCL が Low に保持されます。
マスタ・トランスミッタ・モード	I2C モジュールはマスタで、制御情報とデータをスレーブに送信します。 すべてのマスタは、このモードで開始します。このモードでは、7 ビット / 10 ビットのアドレッシング形式のいずれかで組み立てられたデータが SDA 上でシフト・アウトされます。ビット・シフトは、SCL 上の I2C モジュールによって生成されるクロック・パルスと同期されます。クロック・パルスは禁止され、バイトを送信した後にデバイスの介入が必要な場合 (I2CSTR で XSMT = 0)、SCL が Low に保持されます。

要約すると、SCL は次の条件で Low に保持されます。

- スレーブ・レシーバ・モードで、オーバーラン条件が検出されたとき (RSFULL = 1)。
- スレーブ・トランスミッタ・モードで、アンダーフロー条件が検出されたとき (XSMT = 0)。

I2C のスレーブ・モードは、I2C の マスタ・モードがデータを要求したとき、要求を受け付けてデータを提供する必要があります。

- スレーブ・レシーバ・モードで SCL を解放するには、I2CDRR からデータを読み取ります。
- スレーブ・トランスミッタ・モードで SCL を解放するには、I2CDXR にデータを書き込みます。
- データを処理せず強制的に解放するには、I2CMDR.IRS ビットを使用してモジュールをリセットします。

表 1-3. I2CMDR の RM、STT、STP ビットによって定義されるマスタ・トランスミッタ / レシーバのバス・アクティビティ

RM	STT	STP	バス・アクティビティ ⁽¹⁾	概要
0	0	0	なし	アクティビティなし
0	0	1	P	STOP 条件
0	1	0	S-A-D..(n)..D	START 条件、スレーブ・アドレス、n 個のデータ・バイト (n = I2CCNT の値)
0	1	1	S-A-D..(n)..D-P	START 条件、スレーブ・アドレス、n 個のデータ・バイト、STOP 条件 (n = I2CCNT の値)
1	0	0	なし	アクティビティなし
1	0	1	P	STOP 条件
1	1	0	S-A-D-D-D	反復モードの転送: START 条件、スレーブ・アドレス、STOP 条件または次の START 条件まで連続的にデータを転送
1	1	1	なし	予約ビットの組み合わせ (アクティビティなし)

(1) S = START 条件、A = アドレス、D = データ・バイト、P = STOP 条件

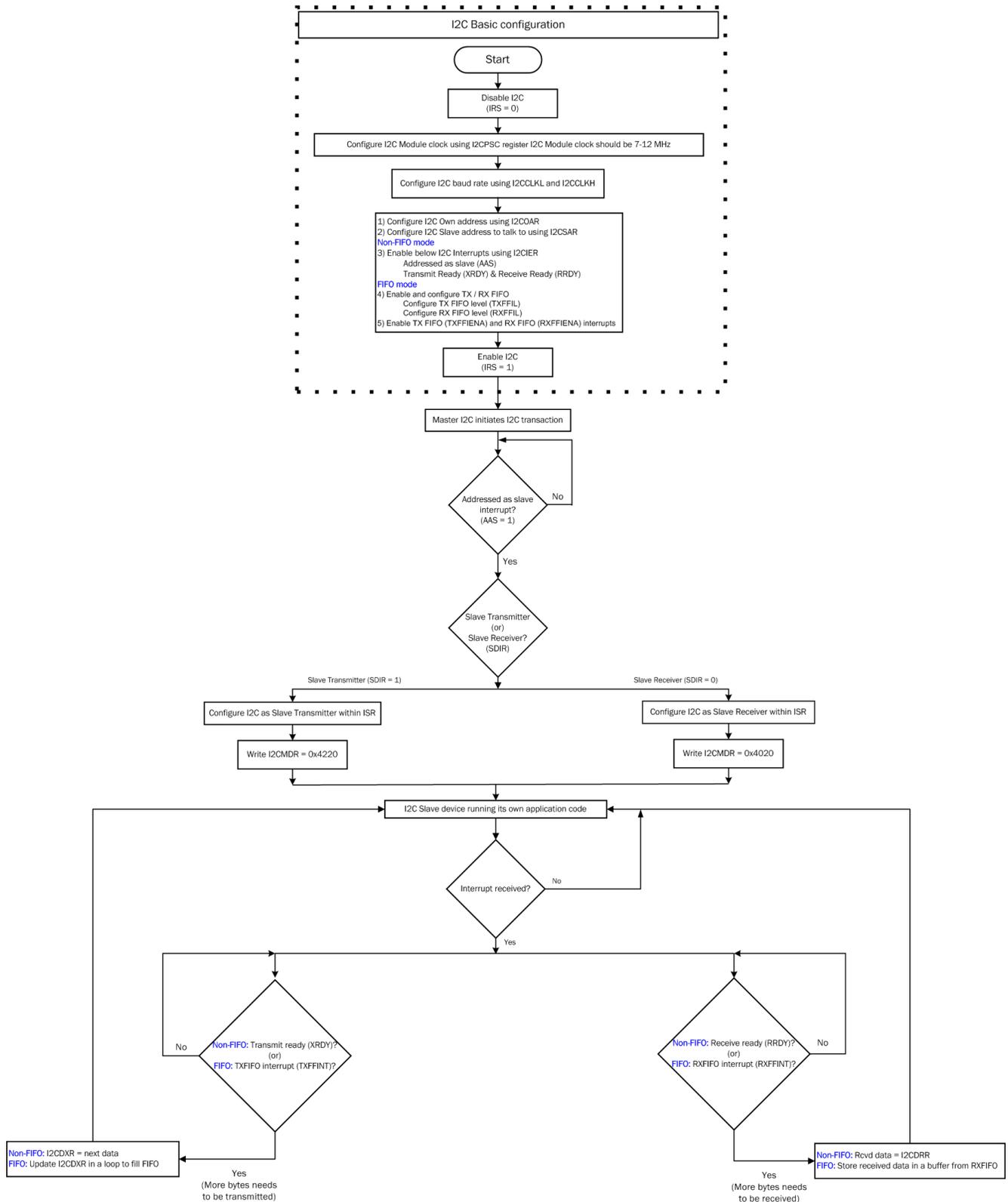


図 1-6. I2C スレーブ の TX/RX フローチャート

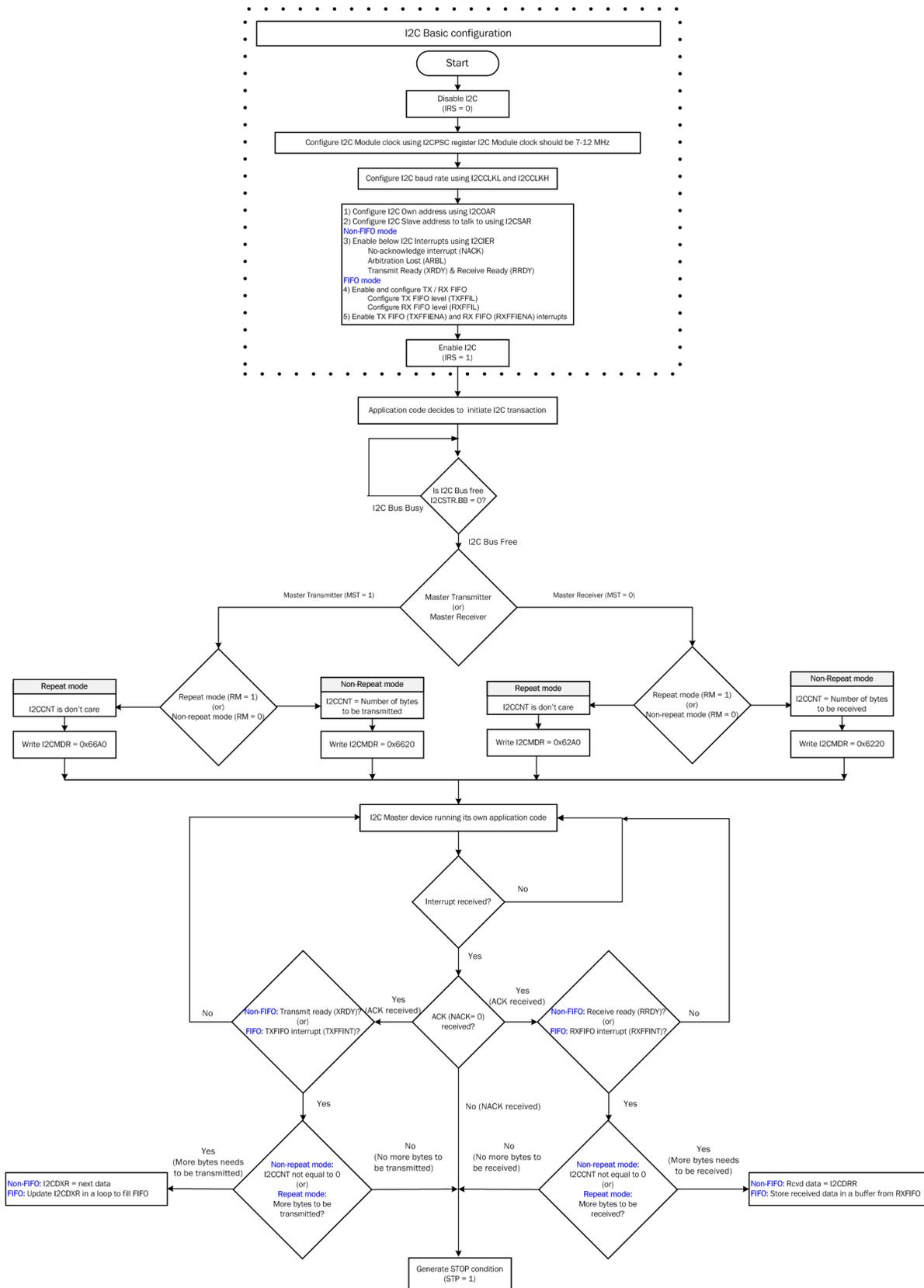


図 1-7. I2C マスタの TX/RX フローチャート

1.3.4 I2C モジュールの START および STOP 条件

I2C バス上でモジュールがマスタに構成されているときは、I2C モジュールで START および STOP 条件を生成できます。図 1-8 に示すように:

- START 条件は、SCL が High のときに SDA ラインが High から Low に遷移することと定義されます。マスタは、この条件を駆動して、データ転送の開始を示します。
- STOP 条件は、SCL が High のときに SDA ラインが Low から High に遷移することと定義されます。マスタは、この条件を駆動して、データ転送の終了を示します。

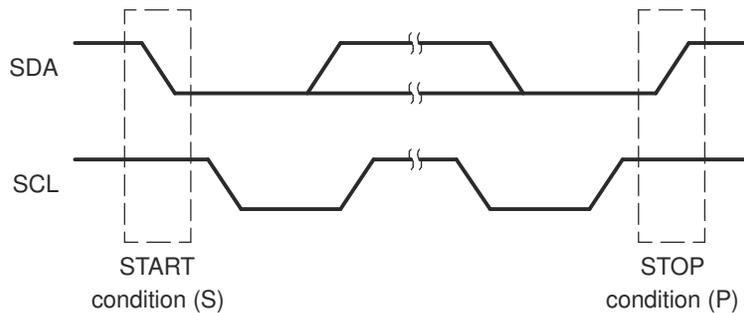


図 1-8. I2C モジュールの START および STOP 条件

START 条件の後には、次に STOP 条件が発生するまで、I2C バスはビジーと見なされ、I2CSTR のバス・ビジー (BB) ビットは 1 です。STOP 条件から次の START 条件まで、バスはフリーとみなされ、BB は 0 です。

I2C モジュールが START 条件でデータ転送を開始するには、I2CMDR のマスタ・モード・ビット (MST) と START 条件ビット (STT) の両方を 1 にする必要があります。I2C モジュールが STOP 条件でデータ転送を終了するには、STOP 条件ビット (STP) が 1 に設定されている必要があります。BB ビットが 1 に設定され、STT ビットが 1 に設定されると、反復 START 条件が生成されます。I2CMDR およびビット (MST、STT、STP を含む) の説明については、[セクション 1.7](#) を参照してください。

I2C ペリフェラルは、リセット中 ($IRS = 0$) に START 条件または STOP 条件を検出できません。I2C ペリフェラルがリセット中 ($IRS = 0$)、BB ビットはクリア状態 ($BB = 0$) のままです。I2C ペリフェラルがリセットから復帰したとき (IRS が 1 に設定されている)、BB ビットは START 条件または STOP 条件が検出されるまで、I2C バスのステータスを正しく反映しません。

I2C を使用して最初のデータ転送を開始する前に、以下の手順に従います。

1. IRS ビットを 1 に設定して I2C ペリフェラルのリセット状態を終了した後で、アプリケーションの最も長いデータ転送に必要な合計時間よりも長い時間だけ待機します。I2C がリセットを終了してから一定時間待つことで、I2C バスに少なくとも 1 つの START または STOP 条件が発生し、BB ビットによってキャプチャされることを保証できます。この期間が経過すると、BB ビットは I2C バスの状態を正しく反映します。
2. 処理を進める前に、BB ビットを確認し、 $BB = 0$ (バスがビジーではない) であることを確認します。
3. データ転送を開始します。

転送の間に I2C ペリフェラルをリセットしなければ、BB ビットが実際のバス・ステータスを反映していることを確認できません。転送の間に I2C ペリフェラルをリセットする必要がある場合は、I2C ペリフェラルがリセット状態を終了するたびに、手順 1~3 を繰り返します。

1.3.5 非反復モードと反復モードの比較

非反復モード

- I2CMDR.RM = 0 のとき、I2C モジュールは非反復モードに構成されます。
- I2CCNT レジスタは、送信または受信されるバイト数を決定します。
- I2CMDR で STP = 0 なら、内部データ・カウンタが 0 にカウントダウンすると ARDY ビットがセットされます。
- STP = 1 なら、ARDY ビットはセットされず、内部データ・カウンタが 0 になると I2C モジュールが STOP 条件を生成します。

注

非反復モード (RM = 0) で、I2CCNT が 0 に設定されているなら、I2C ステート・マシンは 0 バイトではなく 65536 バイトを送受信すると想定されます。

反復モード:

- I2CMDR.RM = 1 のとき、I2C モジュールは反復モードに構成されます。
- 送信または受信されるバイト数は、I2CCNT レジスタの内容では決定されません。
- 送受信されるバイト数は、ソフトウェアで制御できます。
- ARDY ビットは、各バイトの送信および受信の終了時にセットされます。

注

非反復モードまたは反復モードで I2C トランザクションを開始した後は、STOP 条件で I2C トランザクションが完了するまで、別のモードに切り替えることはできません。

1.3.6 シリアル・データ形式

I2C バスでのデータ転送の例を、[図 1-9](#) に示します。I2C モジュールは、1~8 ビットのデータ値をサポートしています。[図 1-9](#) では、8 ビットのデータが転送されます。SDA ラインに出力される各ビットは SCL ラインの 1 パルスに相当し、値は常に最上位ビット (MSB) から先に転送されます。送信または受信できるデータ値の数に制限はありません。[図 1-9](#) で使用されているシリアル・データ形式は、7 ビットのアドレッシング形式です。I2C モジュールは、[図 1-10](#) から [図 1-12](#) までに示す形式をサポートしています。これらの形式については、図の後の段落で説明します。

注

[図 1-9](#) から [図 1-12](#) までにおいて、n = I2CMDR のビット数 (BC) フィールドで指定されたデータ・ビット数 (1~8) です。

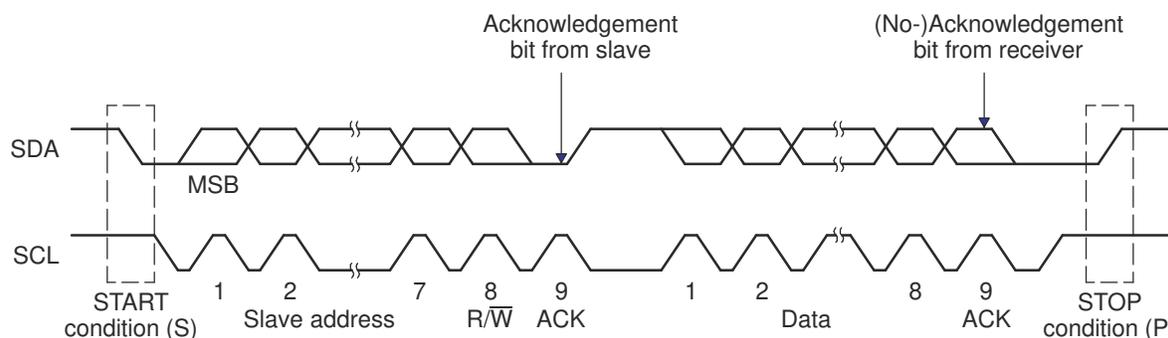


図 1-9. I2C モジュールのデータ転送 (8 ビット・データ構成による 7 ビット・アドレッシングの例)

1.3.6.1 7 ビット・アドレッシング形式

7 ビット・アドレッシング形式は、リセット後のデフォルトの形式です。拡張アドレスをディセーブル (I2CMDR.XA = 0) し、フリー・データ形式をディセーブル (I2CMDR.FDF = 0) すると、7 ビット・アドレッシング形式がイネーブルされます。

この形式 (図 1-10 を参照) では、START 条件 (S) 後の最初のバイトは 7 ビットのスレーブのアドレスの後に R/\bar{W} ビットが続きます。 R/\bar{W} はデータの方法を決定します。

- $R/\bar{W} = 0$: I2C マスタはアドレス指定されたスレーブにデータを書き込み (送信) します。I2CMDR.TRX = 1 を設定 (トランスミッタ・モード) すると、このモードになります。
- $R/\bar{W} = 1$: I2C マスタは、スレーブからデータを読み取り (受信) します。I2CMDR.TRX = 0 を設定 (レシーバ・モード) すると、このモードになります。

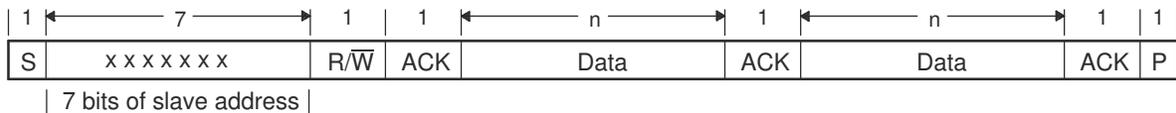


図 1-10. I2C モジュールの 7 ビット・アドレッシング形式 (I2CMDR の FDF = 0、XA = 0)

各バイトの後に、アクノリッジ (ACK) 専用の追加クロック・サイクルが挿入されます。マスタから最初のバイトの後に、スレーブによって ACK ビットが挿入されると、その後にトランスミッタ (R/\bar{W} ビットに応じて マスタまたはスレーブ) から n ビットのデータが続きます。ここで n は、I2CMDR のビット・カウント (BC) フィールドにより決定される 1 から 8 までの数値です。データ・ビットが転送された後で、レシーバは ACK ビットを挿入します。

1.3.6.2 10 ビット・アドレッシング形式

10 ビット・アドレッシング形式をイネーブルするには、拡張アドレス (I2CMDR.XA = 1) を設定し、フリー・データ形式をディセーブル (I2CMDR.FDF = 0) します。

10 ビット・アドレッシング形式 (図 1-11 を参照) は 7 ビット・アドレッシング形式と似ていますが、マスタがスレーブのアドレスを 2 つの別のバイト転送で送信します。最初のバイトは 11110b、10 ビットのスレーブ・アドレスの MSB 2 ビット、および R/\bar{W} で構成されます。2 番目のバイトは、10 ビットのスレーブのアドレスの残り 8 ビットです。2 バイトの転送が終わるごとに、スレーブはアクノリッジを送信する必要があります。マスタが 2 番目のバイトをスレーブに書き込んでから、マスタはデータを書き込むか、反復 START 条件を使用してデータの方法を変更できます。10 ビット・アドレッシングの使用の詳細については、NXP Semiconductors の I2C バス仕様を参照してください。

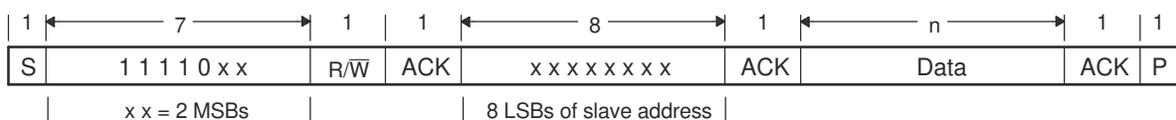


図 1-11. I2C モジュールの 10 ビット・アドレッシング形式 (I2CMDR で FDF = 0、XA = 1)

1.3.6.3 フリー・データ形式

フリー・データ形式をイネーブルするには、I2CMDR を FDF = 1 に設定します。

この形式 (図 1-12 を参照) では、START 条件 (S) 後に来る最初のバイトはデータ・バイトです。各データ・バイトの後に ACK ビットが挿入されます。このビットは、I2CMDR の BC フィールドに応じて 1~8 ビットに設定できます。アドレスやデータ方向のビットは送信されません。したがって、トランスミッタとレシーバの両方がフリー・データ形式をサポートし、転送中にデータの方向が変化しない必要があります。

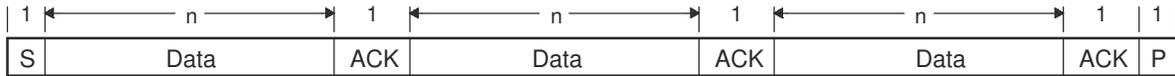


図 1-12. I2C モジュールのフリー・データ形式 (I2CMDR で FDF = 1)

注

デジタル・ループバック・モード (I2CMDR.DLB = 1) では、フリー・データ形式はサポートされていません。

表 1-4. I2CMDR の MST ビットと FDF ビットが I2CMDR の TRX ビットの役割に与える影響

MST	FDF	I2C モジュールの状態	TRX の機能
0	0	スレーブ・モードで、フリー・データ形式モードではない	TRX は無視されます I2C モジュールは、マスタからのコマンドに応じて、レシーバまたはトランスミッタとして応答します。
0	1	スレーブ・モードで、フリー・データ形式モード	フリー・データ形式モードでは、I2C モジュールが送信中を通してトランスミッタまたはレシーバに維持される必要があります。TRX は、I2C モジュールの役割を指定します。 TRx = 1: I2C モジュールはトランスミッタです。 TRx = 0: I2C モジュールはレシーバです。
1	0	マスタ・モードで、フリー・データ形式モードではない	TRx = 1: I2C モジュールはトランスミッタです。 TRx = 0: I2C モジュールはレシーバです。
1	1	マスタ・モードで、フリー・データ形式モード	TRx = 0: I2C モジュールはレシーバです。 TRx = 1: I2C モジュールはトランスミッタです。

1.3.6.4 反復 START 条件の使用

I2C マスタは、STOP 条件を駆動して I2C バスの制御権を渡す必要なしに、複数のスレーブ・アドレスと通信できます。これは、各データ・タイプの最後に別の START 条件を駆動することで行えます。反復 START 条件は、7 ビット・アドレッシング、10 ビット・アドレッシング、およびフリー・データ形式で使用できます。7 ビット・アドレッシング形式での反復 START 条件を、図 1-13 に示します。

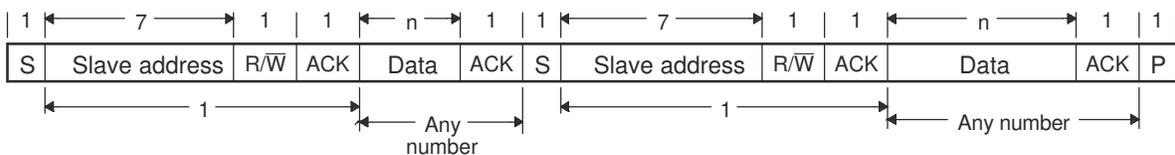


図 1-13. 反復 START 条件 (この例は 7 ビット・アドレッシング形式)

注

図 1-13 で、n = I2CMDR のビット数 (BC) フィールドで指定されたデータ・ビット数 (1~8) です。

1.3.7 クロックの同期

通常の条件では、クロック信号の SCL を生成するのはマスタデバイスのみです。しかし、調停手順のときは 2 つ以上のマスタが存在するため、クロックを同期してデータ出力を比較可能にする必要があります。クロックの同期を、[図 1-14](#) に図示します。SCL の有線 AND の性質は、最初に SCL で Low 期間を生成するデバイスが、他のデバイスより優先されることを示しています。この High から Low への遷移時に、他のデバイスのクロック・ジェネレータは、強制的に独自の Low 期間を開始します。SCL は、Low 期間が最も長いデバイスによって Low に保持されます。Low 期間が完了した他のデバイスは、High 期間を開始する前に、SCL の解放を待つ必要があります。SCL 上で同期された信号が得られ、最も遅いデバイスが Low 期間の長さを、最も速いデバイスが High 期間の長さを決定します。

デバイスがクロック・ラインをさらに長い時間プルダウンすると、すべてのクロック・ジェネレータが待機状態に移行する必要があります。これによって、スレーブが速いマスタの速度を引き下げ、遅いデバイスが受信したバイトの保存や、送信するバイトの準備を行うための十分な時間を確保できます。

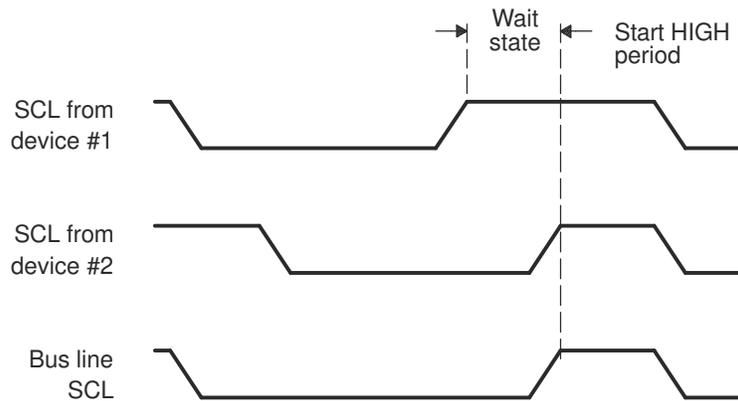


図 1-14. 調停中の 2 つの I2C クロック・ジェネレータの同期

1.3.8 調停

2 つ以上のマスタ・トランスミッタがほぼ同時に同じバスで送信の開始を試みると、調停手順が開始されます。調停手順では、競合するトランスミッタによってシリアル・データ・バス (SDA) に現れるデータを使用します。2 つのデバイス間の調停手順を、[図 1-15](#) に示します。SDA ラインを High にする最初のマスタ・トランスミッタは、SDA を Low に駆動する別のマスタ・トランスミッタによって優先されます。調停手順では、シリアル・データ・ストリームを最小のバイナリ値で送信するデバイスが優先されます。2 つ以上のデバイスが送信する最初のバイトが同一なら、次のバイトでも調停が続けられます。

I2C モジュールが調停に負けた側のマスタなら、スレーブ・レシーバ・モードに切り替わり、調停に負けた (ARBL) フラグをセットして、調停に負けた割り込み要求を生成します。

シリアル転送中に、START 条件の繰り返し、または STOP 条件が SDA に転送されたとき、調停手順がまだ進行中なら、関係するマスタ・トランスミッタは形式フレームの同じ位置で、START 条件の繰り返し、または STOP 条件を送信する必要があります。次のものの間では調停は許可されません。

- START 条件の繰り返しとデータ・ビット
- STOP 条件とデータ・ビット
- START 条件の繰り返しと STOP 条件

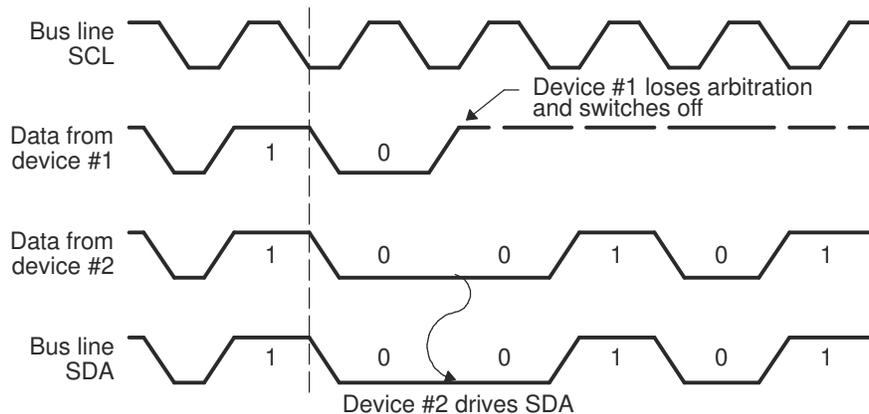


図 1-15. 2 つのマスタ・トランスミッタ間の調停手順

1.3.9 デジタル・ループバック・モード

I2C モジュールは、デジタル・ループバックと呼ばれる自己テスト・モードをサポートしています。I2CMDR レジスタの DLB ビットをセットすると、このモードがイネーブルされます。このモードでは、I2CDXR レジスタから送信されたデータが I2CDRR レジスタで受信されます。データは内部パスに従い、I2CDRR に達するまでに n サイクルを要します。ここで、

$$n = 8 * (\text{SYSCLK}) / (\text{I2C モジュール・クロック (Fmod)})$$

送信クロックと受信クロックは同じです。外部 SDA ピンに現れるアドレスは、I2COAR レジスタのアドレスです。デジタル・ループバック・モードでの信号の配線を、図 1-16 に示します。

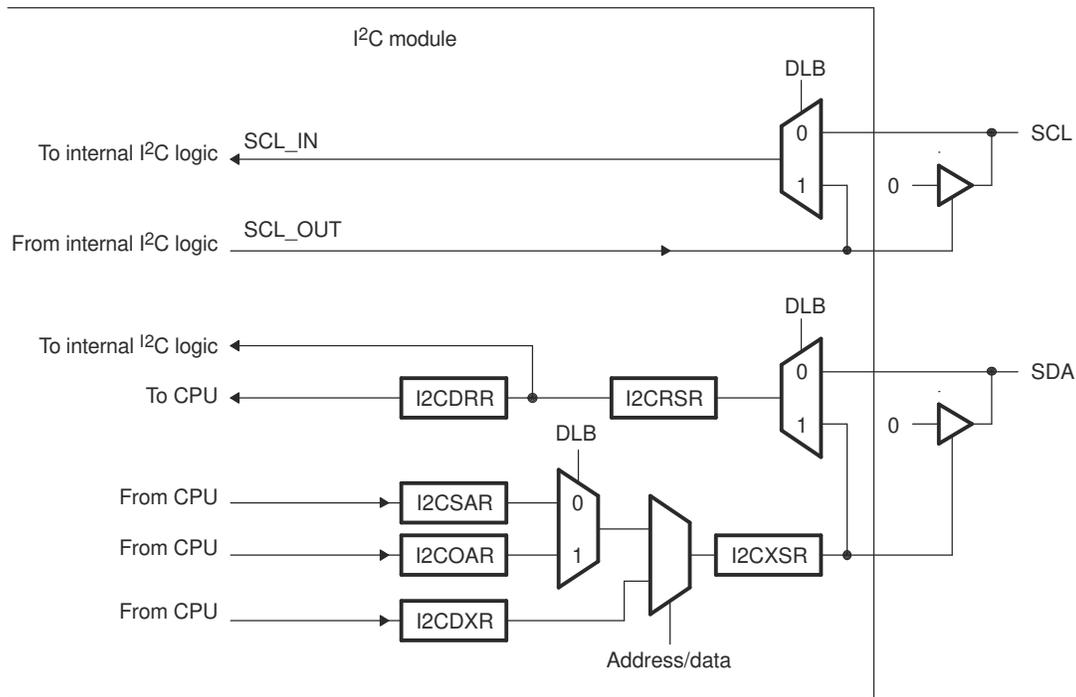


図 1-16. デジタル・ループバック・モード (DLB) ビットの影響を示すピン図

注

デジタル・ループバック・モードでは、フリー・データ形式 (I2CMDR.FDF = 1) はサポートされていません。

1.3.10 NACK ビットの生成

I2C モジュールがレシーバ (マスタまたはスレーブ) のとき、トランスミッタから送信されたビットをアクノリッジ、または無視します。新しいビットを無視するには、バスのアクノリッジ・サイクル中に、I2C モジュールが NACK (非アクノリッジ) ビットを送信する必要があります。I2C モジュールに NACK ビットの送信を許可する各種の方法の要約を、表 1-5 に示します。

表 1-5. NACK ビットを生成する方法

I2C モジュールの条件	NACK ビット生成のオプション
スレーブ・レシーバ・モード	オーバーラン条件を許容 (I2CSTR で RSFULL = 1) モジュールのリセット (I2CMDR で IRS = 0) 受信する最後のデータ・ビットの立ち上がりエッジより前に、I2CMDR の NACKMOD ビットをセットする
マスタ・レシーバ・モードと 反復モード (I2CMDR で RM = 1)	STOP 条件の生成 (I2CMDR で STP = 1) モジュールのリセット (I2CMDR で IRS = 0) 受信する最後のデータ・ビットの立ち上がりエッジより前に、I2CMDR の NACKMOD ビットをセットする
マスタ・レシーバ・モードと 非反復モード (I2CMDR で RM = 0)	I2CMDR で STP = 1 なら、内部データ・カウンタは 0 までカウントダウンできるため、STOP 条件が強制的に発生する STP = 0 なら、STP = 1 にして STOP 条件を生成する モジュールをリセットする (I2CMDR で IRS = 0)。= 1 なら STOP 条件が生成される 受信する最後のデータ・ビットの立ち上がりエッジより前に、I2CMDR の NACKMOD ビットをセットする

1.4 I2C モジュールにより生成される割り込み要求

各 I2C モジュールは、2 つの CPU 割り込みを生成できます。

1. 基本 I2C 割り込み: この割り込みをトリガできる基本 I2C 割り込みソースについては、[セクション 1.4.1](#) を参照してください。
2. I2C FIFO 割り込み: この割り込みをトリガできる I2C FIFO 割り込みソースについては、[セクション 1.4.2](#) を参照してください。

1.4.1 基本的な I2C 割り込み要求

I2C モジュールは、[表 1-6](#) で説明されている割り込み要求を生成します。[図 1-17](#) に示すように、すべての要求はアービターによって、CPU への単一の I2C 割り込み要求に多重化されます。それぞれの割り込み要求について、ステータス・レジスタ (I2CSTR) にフラグ・ビット、割り込みイネーブル・レジスタ (I2CIER) にイネーブル・ビットがあります。指定されたイベントのいずれかが発生すると、フラグ・ビットがセットされます。対応するイネーブル・ビットが 0 なら、割り込み要求はブロックされます。イネーブル・ビットが 1 なら、要求は I2C 割り込みとして CPU に転送されます。

I2C 割り込みは、CPU のマスク可能割り込みの 1 つです。他のマスク可能な割り込み要求と同様に、CPU で正しくイネーブルされていれば、CPU は対応する割り込みサービス・ルーチン (I2CINT1A_ISR) を実行します。I2C 割り込みの I2CINT1A_ISR は、割り込みソース・レジスタ I2CISRC を読み出して、割り込みソースを判定できます。その後で、I2CINT1A_ISR は適切なサブルーチンに分岐します。

CPU が I2CISRC を読み取った後で、次のイベントが発生します。

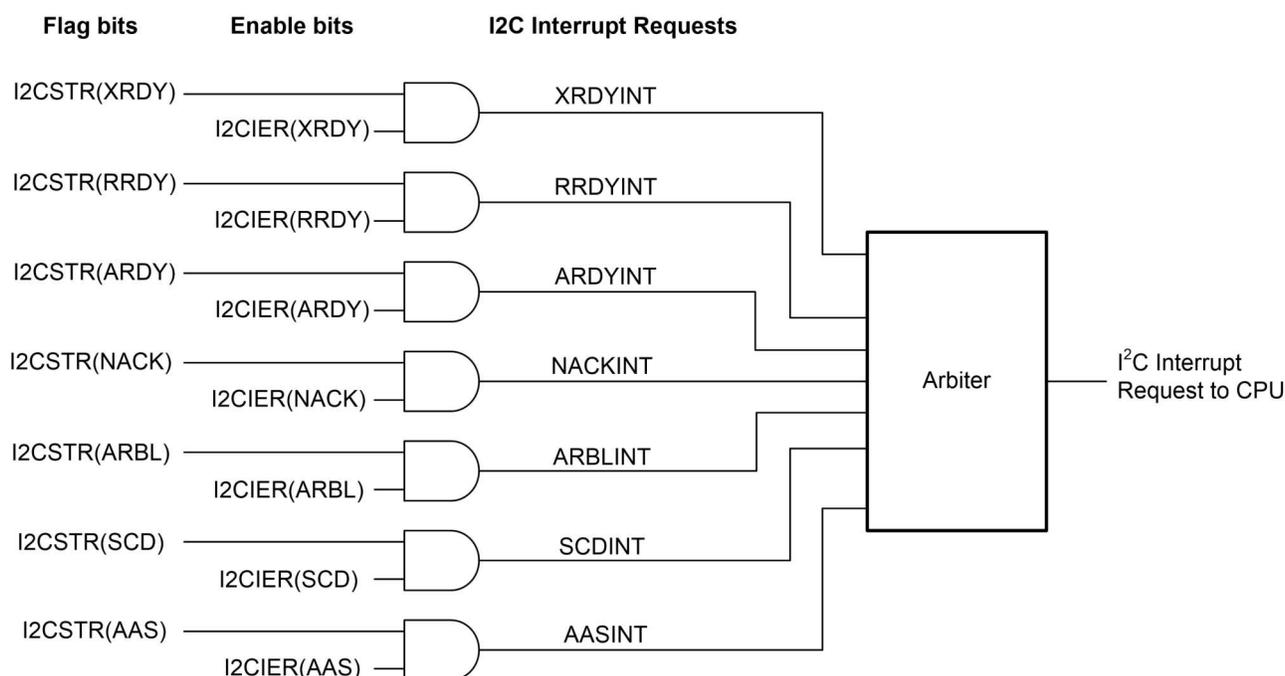
1. I2CSTR で、ソース割り込みのフラグがクリアされます。例外: I2CSTR の ARDY、RRDY、XRDY ビットは、I2CISRC の読み取り時にクリアされません。これらのビットをクリアするには、そのビットに 1 を書き込みます。
2. アービターは、残りの割り込み要求のうち、最も優先度の高いものを決定し、その割り込みのコードを I2CISRC に書き込んで、割り込み要求を CPU に転送します。

表 1-6. 基本的な I2C 割り込み要求の説明

I2C 割り込み要求	割り込みソース
XRDYINT	送信準備完了状況: 以前のデータがデータ送信レジスタ (I2CDXR) から送信シフト・レジスタ (I2CXHR) にコピー済みで、新しいデータを受け付ける準備ができています。 XRDYINT を使用する代わりに、CPU はステータス・レジスタ I2CSTR の XRDY ビットをポーリングできます。FIFO モードでは、XRDYINT を使用しないでください。代わりに FIFO 割り込みを使用してください。
RRDYINT	受信準備完了状況: 受信シフト・レジスタ (I2CRSR) から I2CDRR にデータがコピー済みで、データ受信レジスタ (I2CDRR) を読み取る準備ができています。 RRDYINT を使用する代わりに、CPU は I2CSTR の RRDY ビットをポーリングできます。FIFO モードでは、RRDYINT を使用しないでください。代わりに FIFO 割り込みを使用してください。
ARDYINT	レジスタ・アクセス準備完了状況: 以前にプログラムされたアドレス、データ、コマンドの値が使用済みで、I2C モジュールのレジスタにアクセスする準備ができています。 ARDYINT を生成するイベントは、I2CSTR の ARDY ビットを設定するイベントと同じです。 ARDYINT を使用する代わりに、CPU は ARDY ビットをポーリングできます。
NACKINT	アクリッジなしの状況: I2C モジュールはマスタ・トランスミッタとして構成されており、スレーブ・レシーバからアクリッジを受信しませんでした。 NACKINT を使用する代わりに、CPU は I2CSTR の NACK ビットをポーリングできます。
ARBLINT	調停に負けた状況: I2C モジュールは、別のマスタ・トランスミッタとの間で調停に負けました。 ARBLINT を使用する代わりに、CPU は I2CSTR の ARBL ビットをポーリングできます。
SCDINT	STOP 条件が検出された: I2C バスで STOP 条件が検出されました。 SCDINT を使用する代わりに、CPU はステータス・レジスタ I2CSTR の SCD ビットをポーリングできます。
AASINT	スレーブとしてアドレス指定された状況: I2C は、I2C バス上の別のマスタによって、スレーブ・デバイスとしてアドレス指定されました。

表 1-6. 基本的な I2C 割り込み要求の説明 (continued)

I2C 割り込み要求	割り込みソース
	AASINT を使用する代わりに、CPU はステータスレジスタ I2CSTR の AAS ビットをポーリングできます。


図 1-17. I2C 割り込み要求のイネーブルパス

基本的な I2C 割り込み要求の優先度を、優先度が高いものから低いものへの順にリストで示します。

1. ARBLINT
2. NACKINT
3. ARDYINT
4. RRDYINT
5. XRDYINT
6. SCDINT
7. AASINT

通常の送信割り込みタイミングにより、バイトの途中でトランザクションが中断された場合でも、古いデータが送信バッファに残る可能性があります。これを回避するには、I2CEMDR レジスタの FCM ビットをセットします。このビットがセットされている場合、送信データ準備完了割り込みは、バス・トランザクションにデータが必要な場合のみ生成されます。マスタ・モードでは、アドレス・バイトの ACK が受信されたとき、最初の割り込みが生成されます。スレーブ・モードでは、アドレスが一致したとき、最初の割り込みが生成されます。データがアクロリッジされると、以後の割り込みが生成されます。このモードでは、送信準備完了割り込みと同時に XRDY がアサートされます。

I2C モジュールの I2CEMDR レジスタには、下位互換ビット (BC) があります。図 1-18 のタイミング図は、スレーブ・トランスミッタとして構成されているとき、下位互換ビットが I2C モジュールのレジスタと割り込みにどのような影響を及ぼすかを示すものです。

Slave Transmitter

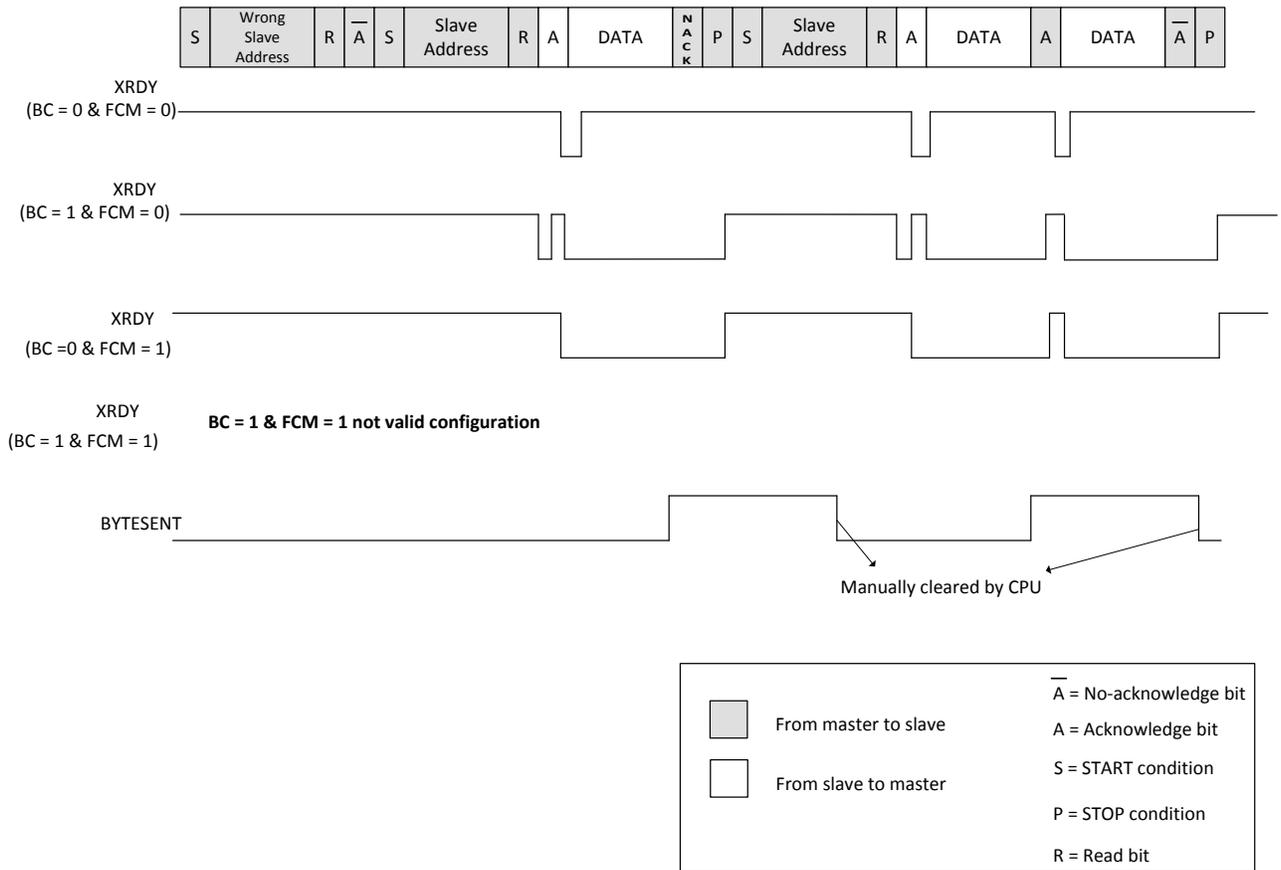


図 1-18. 下位互換モードおよび上位互換ビット、スレーブ・トランスミッタ

1.4.2 I2C の FIFO 割り込み

7 つの基本的な I2C 割り込みに加えて、送信および受信 FIFO にはそれぞれ割り込み (I2CINT2A) を生成する機能があります。送信 FIFO は、定義された数のバイトを送信した後で、最大 16 個の割り込みを生成するように構成できます。受信 FIFO は、定義された数のバイトを受信した後で、最大 16 個の割り込みを生成するように構成できます。これら 2 つの割り込みソースは、1 つのマスク可能 CPU 割り込みに OR 接続されています。I2C FIFO 割り込みの構造を、[図 1-19](#) に示します。その後で、割り込みサービスルーチンは FIFO 割り込みステータス・フラグを読み取って、割り込みの発生源を特定できます。I2C 送信 FIFO レジスタ (I2CFFTX) と I2C 受信 FIFO レジスタ (I2CFFRX) の説明を参照してください。

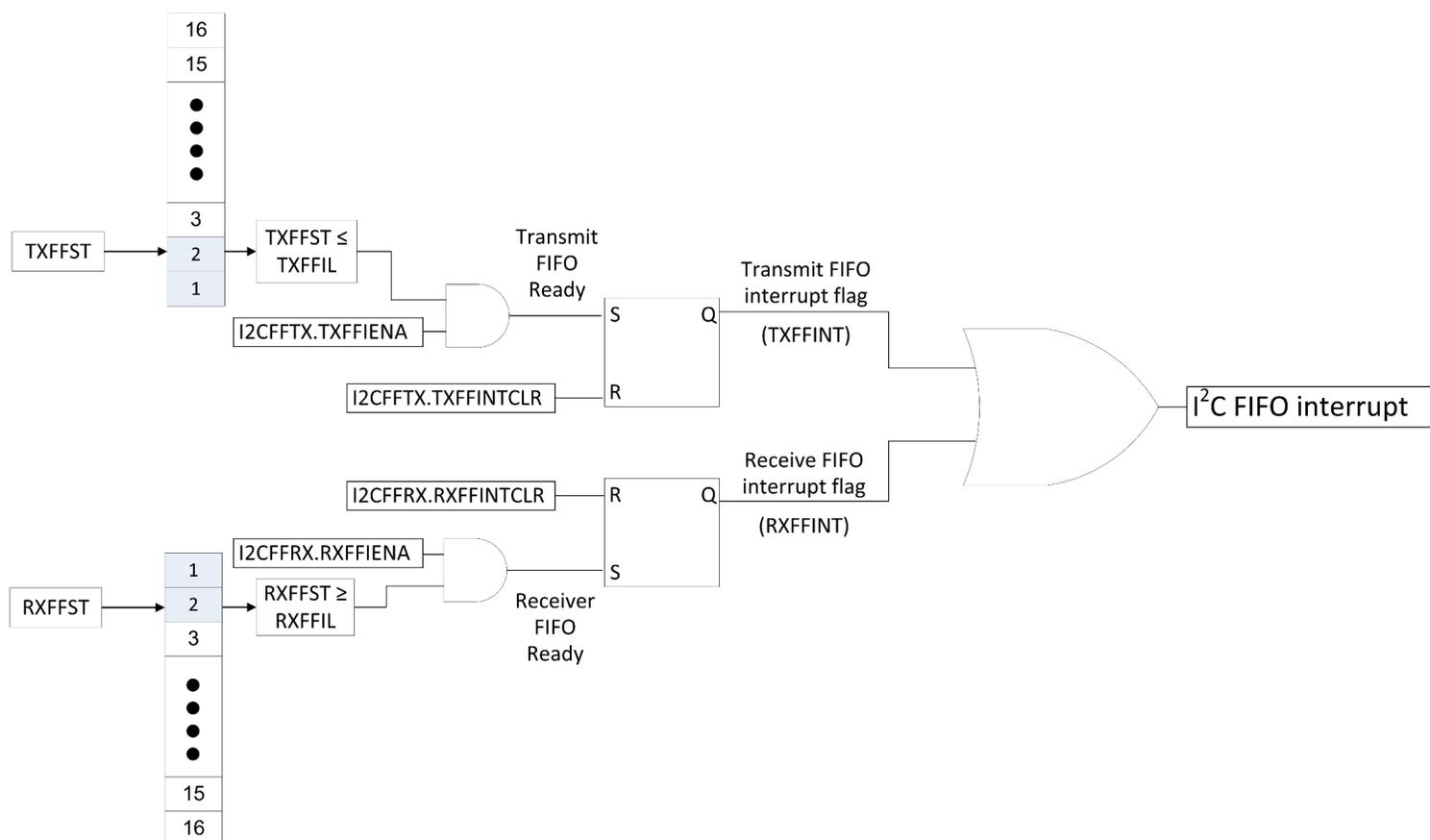


図 1-19. I2C_FIFO_interrupt

1.5 I2C モジュールのリセットまたはディセーブル

I2C モジュールは、2 つの方法でリセットまたはディセーブルできます。

- I2C モード・レジスタ (I2CMODR) の I2C リセット・ビット (IRS) に 0 を書き込む。(I2CSTR 内の) すべてのステータス・ビットは強制的にデフォルト値になり、IRS が 1 に変更されるまで I2C モジュールはディセーブルのままです。SDA ピンと SCL ピンは高インピーダンス状態です。
- $\overline{\text{XRS}}$ ピンを Low に駆動して、デバイスのリセットを開始する。デバイス全体がリセットされ、ピンを High に駆動するまで RESET 状態に保持されます。 $\overline{\text{XRS}}$ ピンが解放されると、すべての I2C モジュール・レジスタがデフォルト値にリセットされる。IRS ビットは強制的に 0 になり、これにより I2C モジュールがリセットされます。IRS に 1 を書き込むまで、I2C モジュールは RESET 状態を維持します。

I2C モジュールを構成または再構成するときは、IRS が 0 の必要があります。IRS を強制的に 0 にすることで、電力を節約し、エラー条件をクリアできます。

1.6 ソフトウェア

1.6.1 I2C の例

注:これらの例は、[C2000Ware](#) のインストールで、次の場所にあります。

C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/i2c

これらの例へのクラウド・アクセスは、[dev.ti.com C2000Ware Examples](http://dev.ti.com/C2000Ware_Examples) リンクにあります。

1.6.1.1 FIFO 割り込み用の C28x-I2C ライブラリのソース・ファイル

ファイル: i2cLib_FIFO_controller_interrupt.c

1.6.1.2 ポーリングを使用する FIFO 用の C28x-I2C ライブラリのソース・ファイル

ファイル: i2cLib_FIFO_polling.c

1.6.1.3 FIFO 割り込み用の C28x-I2C ライブラリのソース・ファイル

ファイル: i2cLib_FIFO_controller_target_interrupt.c

1.6.1.4 FIFO 割り込み付きの I2C デジタル・ループバック

ファイル: i2c_ex1_loopback.c

このプログラムは、I2C モジュールの内部ループバック・テスト・モードを使用します。TX と RX の両方の I2C FIFO と、それらの割り込みが使用されます。ピン・マルチプレクサと I2C の初期化は、`sysconfig` ファイルを使用して実行します。

データ・ストリームが送信されてから、受信したストリームと比較されます。送信されるデータは次のようなものです。

0000 0001

0001 0002

0002 0003

....

00FE 00FF

00FF 0000

など。

このパターンは永続的に繰り返されます。

外部接続

- なし

調べる変数

- sData - 送信するデータ
- rData - 受信したデータ
- rDataPoint - エラー・チェックの目的で、受信ストリームの最後の位置を追跡するために使用されます

1.6.1.5 I2C EEPROM

FILE: i2c_ex2_eeprom.c

This program will write 1-14 words to EEPROM and read them back. The data written and the EEPROM address written to are contained in the message structure, `i2cMsgOut`. The data read back will be contained in the message structure `i2cMsgIn`.

External Connections

- Connect external I2C EEPROM at address 0x50
- Connect `DEVICE_GPIO_PIN_SDAA` on to external EEPROM SDA (serial data) pin
- Connect `DEVICE_GPIO_PIN_SCL` on to external EEPROM SCL (serial clock) pin

Watch Variables

- `i2cMsgOut` - Message containing data to write to EEPROM
- `i2cMsgIn` - Message containing data read from EEPROM

1.6.1.6 I2C Digital External Loopback with FIFO Interrupts

FILE: i2c_ex3_external_loopback.c

This program uses the I2CA and I2CB modules for achieving external loopback. The I2CA TX FIFO and the I2CB RX FIFO are used along with their interrupts.

A stream of data is sent on I2CA and then compared to the received stream on I2CB. The sent data looks like this:

```
0000 0001
0001 0002
0002 0003
....
00FE 00FF
00FF 0000
etc..
```

This pattern is repeated forever.

External Connections

- Connect SCLA (DEVICE_GPIO_PIN_SCLA) to SCLB (DEVICE_GPIO_PIN_SCLB)
- and SDA A (DEVICE_GPIO_PIN_SDA A) to SDA B (DEVICE_GPIO_PIN_SDA B)
- Connect DEVICE_GPIO_PIN_LED1 to an LED used to depict data transfers.

Watch Variables

- *sData* - Data to send
- *rData* - Received data
- *rDataPoint* - Used to keep track of the last position in the receive stream for error checking

1.6.1.7 I2C EEPROM

ファイル: i2c_ex4_eeprom_polling.c

このプログラムは、I2Cポーリング方式のEEPROMを使用して、EEPROMの書き込み / 読み取りコマンドを実行する方法を示しています。この例で使用しているのはAT24C256です。

1.6.1.8 FIFO 割り込みによる I2C コントローラとターゲットとの通信

ファイル: i2c_ex5_controller_target_interrupt.c

このプログラムでは、コントローラとターゲットの両方の構成でI2CAおよびI2CBモジュールを使用する方法を示します。この例では、I2C FIFO 割り込みを使用しており、ポーリングは使用していません。

例 1: I2CA をコントローラ・トランスミッタ、I2CB をターゲット・レシーバとして使用します。例 2: I2CA をコントローラ・レシーバ、I2CB をターゲット・トランスミッタとして使用します。例 3: I2CB をコントローラ・トランスミッタ、I2CA をターゲット・レシーバとして使用します。例 4: I2CB をコントローラ・レシーバ、I2CA をターゲット・トランスミッタとして使用します。

メモリ・ウィンドウで変数を調べる

- *I2CA_TXdata*
- *I2CA_RXdata*
- *I2CB_TXdata*
- エラー・チェック用の I2CB_RXdata ストリーム

1.6.1.9 I2C EEPROM

ファイル: i2c_ex6_eeprom_interrupt.c

このプログラムは、I2C 割り込み EEPROM を使用して、各種の EEPROM で書き込みおよび読み取りコマンドを実行する方法を示しています。この例で使用しているのは AT24C256 です。

//例 1:EEPROM のバイト書き込み //例 2:EEPROM のバイト読み取り //例 3:EEPROM のワード (16 ビット) 書き込み //例 4:EEPROM のワード (16 ビット) 読み取り //例 5:EEPROM のページ書き込み //例 6:EEPROM のワードのページ単位読み取り

調べる変数

- TX_MsgBuffer - 送信するデータを格納するメッセージ・バッファ
- RX_MsgBuffer - 受信したデータを格納するメッセージ・バッファ

1.7 I2C レジスタ

このセクションでは、C28x I2C モジュールのレジスタについて説明します。

1.7.1 I2C ベース・アドレス表

表 1-7. I2C ベース・アドレス表

ビット・フィールド名		DriverLib 名	ベース・アドレス	パイプライン保護
インスタンス	構造			
I2caRegs	I2C_REGS	I2CA_BASE	0x0000_7300	あり
I2cbRegs	I2C_REGS	I2CB_BASE	0x0000_7340	あり

1.7.2 I2C_REGS レジスタ

I2C_REGS レジスタのメモリマップされたレジスタを、表 1-8 に示します。表 1-8 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更しないでください。

表 1-8. I2C_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	I2COAR	I2C の独自アドレス		表示
1h	I2CIER	I2C の割り込みイネーブル		表示
2h	I2CSTR	I2C のステータス		表示
3h	I2CCLKL	I2C クロックの Low 時間分周器		表示
4h	I2CCLKH	I2C クロックの High 時間分周器		表示
5h	I2CCNT	I2C のデータ数		表示
6h	I2CDRR	I2C データ受信		表示
7h	I2CSAR	I2C スレーブ・アドレス		表示
8h	I2CDXR	I2C データ送信		表示
9h	I2CMDR	I2C のモード		表示
Ah	I2CISRC	I2C の割り込みソース		表示
Bh	I2CEMDR	I2C 拡張モード		表示
Ch	I2CPSC	I2C プリスケーラ		表示
20h	I2CFFTX	I2C FIFO 送信		表示
21h	I2CFFRX	I2C FIFO 受信		表示

表の小さなセルに収まるように、複雑なビット・アクセス・タイプを記号で表記しています。このセクションでアクセス・タイプに使用しているコードを、表 1-9 に示します。

表 1-9. I2C_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み取りでは 0 が返される
書き込みタイプ		
W	W	書き込み
W1C	W1C	1 を書き込むことでクリア
W1S	W1S	1 を書き込むことでセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ配列の変数		
i, j, k, l, m, n		これらの変数をレジスタ名、オフセット、またはアドレスで使用すると、レジスタ配列の値を指すことになり、各レジスタは反復レジスタのグループの一部になります。このレジスタ・グループは階層構造を形成し、配列は式で表されます。
y		この変数をレジスタ名、オフセット、またはアドレスで使用すると、レジスタ配列の値を指すことになり、

1.7.2.1 I2COAR レジスタ (オフセット = 0h) [リセット = 0h]

I2COAR を図 1-20 に示します。また、その説明を表 1-10 に示します。

概略表に戻ります。

I2C の独自アドレス・レジスタ (I2COAR) は 16 ビット・レジスタです。I2C モジュールは、このレジスタを使用して独自のスレーブ・アドレスを指定します。このアドレスは、I2C バスに接続されている他のスレーブとは区別されます。7 ビット・アドレッシング・モードを選択した場合 (I2CMDR で $XA = 0$)、ビット 6~0 のみが使用され、ビット 9~7 には 0 が書き込まれます。

図 1-20. I2COAR レジスタ

15	14	13	12	11	10	9	8
予約済み						OAR	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
OAR							
R/W-0h							

表 1-10. I2COAR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	予約済み	R	0h	予約済み
9-0	OAR	R/W	0h	7 ビット・アドレッシング・モード (I2CMDR で $XA = 0$): ビット 6-0 は 00h~7Fh で、I2C モジュールの 7 ビット・スレーブ・アドレスを示します。ビット 9-7 には 0 を書き込みます。 10 ビット・アドレッシング・モード (I2CMDR で $XA = 1$): ビット 9-0 は 000h~3FFh で、I2C モジュールの 10 ビット・スレーブ・アドレスを示します。 リセット・タイプ: SYSRSn

1.7.2.2 I2CIER レジスタ (オフセット = 1h) [リセット = 0h]

I2CIER を図 1-21 に示します。また、その説明を表 1-11 に示します。

概略表に戻ります。

I2CIER は、CPU が I2C 割り込み要求を個別にイネーブルまたはディセーブルするために使用されます。

図 1-21. I2CIER レジスタ

15	14	13	12	11	10	9	8
予約済み							
R-0h							
7	6	5	4	3	2	1	0
予約済み	AAS	SCD	XRDY	RRDY	ARDY	NACK	ARBL
R-0h	R/W-0h						

表 1-11. I2CIER レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-7	予約済み	R	0h	予約済み
6	AAS	R/W	0h	スレーブ割り込みイネーブルとしてアドレス指定される リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル
5	SCD	R/W	0h	STOP 条件検出割り込みイネーブル リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル
4	XRDY	R/W	0h	送信データ準備完了割り込みイネーブル・ビット。 FIFO モードを使用するときは、このビットをセットしません。 リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル
3	RRDY	R/W	0h	受信データ準備完了割り込みイネーブル・ビット。 FIFO モードを使用するときは、このビットをセットしません。 リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル
2	ARDY	R/W	0h	レジスタ・アクセス準備完了割り込みイネーブル リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル
1	NACK	R/W	0h	非アクリッジ割り込みイネーブル リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル
0	ARBL	R/W	0h	調停に負けた割り込みイネーブル リセット・タイプ: SYSRSn 0h (R/W) = 割り込み要求ディセーブル 1h (R/W) = 割り込み要求イネーブル

1.7.2.3 I2CSTR レジスタ (オフセット = 2h) [リセット = 410h]

I2CSTR を図 1-22 に示します。また、その説明を表 1-12 に示します。

概略表に戻ります。

I2C ステータス・レジスタ (I2CSTR) は 16 ビットのレジスタで、どの割り込みが発生したかを判定し、ステータス情報を読み取るために使用されます。

図 1-22. I2CSTR レジスタ

15	14	13	12	11	10	9	8
予約済み	SDIR	NACKSNT	BB	RSFULL	XSMT	AAS	AD0
R-0h	R/W1C-0h	R/W1C-0h	R-0h	R-0h	R-1h	R-0h	R-0h
7	6	5	4	3	2	1	0
予約済み	BYTESENT	SCD	XRDY	RRDY	ARDY	NACK	ARBL
R-0h	R/W1C-0h	R/W1C-0h	R-1h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 1-12. I2CSTR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	予約済み	R	0h	予約済み
14	SDIR	R/W1C	0h	スレーブ方向ビット リセット・タイプ:SYSRSn 0h (R/W) = I2C がスレーブ・トランスミッタとしてアドレス指定されています。SDIR は、次のいずれかのイベントによってクリアされます。 - 手動でクリア。このビットをクリアするには、1 を書き込みます。 - デジタル・ループバック・モードがイネーブルされる。 - I2C バスで START または STOP 条件が発生する。 1h (R/W) = I2C はスレーブ・トランスミッタとしてアドレス指定されています。
13	NACKSNT	R/W1C	0h	NACK 送信済みビット。 このビットは、I2C モジュールがレシーバ・モードのときに使用されます。 NACKSNT が影響を受ける例の 1 つは、NACK モードが使用されている場合です (の NACKMOD の説明を参照) リセット・タイプ:SYSRSn 0h (R/W) = NACK は送信されていません。NACKSNT ビットは、次のいずれかのイベントによってクリアされます。 - 手動でクリア。このビットをクリアするには、1 を書き込みます。 - I2C モジュールがリセットされる (I2CMDR の IRS ビットに 0 が書き込まれた、またはデバイス全体がリセットされた)。 1h (R/W) = NACK が送信済み:I2C バスのアクリッジ・サイクル中に、非アクリッジ・ビットが送信されました。
12	BB	R	0h	バス・ビジー・ビット。 BB は、I2C バスがビジーか、フリーで別のデータ転送に使用可能かを示します。詳細については、表の後にある段落を参照してください リセット・タイプ:SYSRSn 0h (R/W) = バスがフリー。BB は、次のいずれかのイベントによってクリアされます。 - I2C モジュールが STOP ビット (バスがフリー) を送信または受信する。 - I2C モジュールがリセットされる。 1h (R/W) = バスがビジー:I2C モジュールは、バス上で START ビットを送信または受信しました。

表 1-12. I2CSTR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
11	RSFULL	R	0h	受信シフトレジスタ・フル・ビット。 RSFULL は、受信中のオーバーラン状態を示します。オーバーランは、シフトレジスタ (I2CRSR) に新しいデータが受信され、受信レジスタ (I2CDRR) から古いデータが読み出されていないときに発生します。新しいビットが SDA ピンから到着すると、I2CRSR のビットが上書きされます。新しいデータは、前のデータが読み取られるまで ICDRR にコピーされません。 リセット・タイプ:SYSRSn 0h (R/W) = オーバーランが検出されていません。RSFULL は、次のいずれかのイベントによってクリアされます。 - I2CDRR が CPU によって読み取られる。I2CDRR のエミュレータ読み取りは、このビットに影響を与えません。 - I2C モジュールがリセットされる。 1h (R/W) = オーバーランが検出されました
10	XSMT	R	1h	送信シフトレジスタの空のビット。 XSMT = 0 は、トランスミッタのアンダーフローが発生したことを示します。アンダーフローが発生するのは、送信シフトレジスタ (I2CXSR) が空で、最後の I2CDXR から I2CXSR への転送以後に、データ送信レジスタ (I2CDXR) がロードされていない場合です。次の I2CDXR から I2CXSR への転送は、新しいデータが I2CDXR に入るまで発生しません。新しいデータが時間内に転送されないと、前のデータが SDA ピンで再送信される可能性があります。 リセット・タイプ:SYSRSn 0h (R/W) = アンダーフローが検出されました (空) 1h (R/W) = アンダーフローが検出されていません (空ではない)。XSMT は、次のいずれかのイベントによってクリアされます。 - I2CDXR にデータが書き込まれる。 - I2C モジュールがリセットされる
9	AAS	R	0h	スレーブとしてアドレス指定ビット リセット・タイプ:SYSRSn 0h (R/W) = 7 ビット・アドレッシング・モードでは、NACK、STOP 条件、または反復 START 条件を受信すると、AAS ビットがクリアされます。10 ビット・アドレッシング・モードでは、NACK、STOP 条件、または I2C ペリフェラルの独自スレーブ・アドレスとは異なるスレーブ・アドレスを受信すると、AAS ビットがクリアされます。 1h (R/W) = I2C モジュールは独自スレーブ・アドレスまたはすべて 0 のアドレス (ゼネラル・コール) を認識しました。
8	AD0	R	0h	アドレス 0 ビット リセット・タイプ:SYSRSn 0h (R/W) = AD0 は、START 条件または STOP 条件によってクリアされました。 1h (R/W) = すべて 0 のアドレス (ゼネラル・コール) が検出されました。
7	予約済み	R	0h	予約済み
6	BYTESENT	R/W1C	0h	バイト送信完了の表示。 マスタ/スレーブが SCL/SDA ラインでバイトを正常に送信したとき、BYTESENT が設定されます。これは診断レジスタで、ソフトウェアから明示的にクリアする必要があります。コーナー条件を回避するため、自動的にクリアする機能は組み込まれていません。このため、クリアしないと、古いステータスがいつまでも反映されます。 リセット・タイプ:SYSRSn 0h (R/W) = I2C モジュールは次のデータ・バイトの送信を完了していません。BYTESENT は、次のいずれかのイベントによってクリアされます。 - 手動でクリア。このビットをクリアするには、1 を書き込みます。 - I2C モジュールがリセットされる。 1h (R/W) = I2C モジュールが 1 バイトの送信を完了しました。

表 1-12. I2CSTR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	SCD	R/W1C	0h	<p>ストップ条件検出ビット。 SCD は、I2C が STOP 条件を送信または受信したときにセットされます。 I2C モジュールは、SCD ビットがセットされるまで、I2CMDR[STP] ビットのクリアを遅延させます。 リセット・タイプ:SYSRSn</p> <p>0h (R/W) = SCD が最後にクリアされた後で、STOP 条件は検出されていません。SCD は、次のいずれかのイベントによってクリアされます。</p> <ul style="list-style-type: none"> - I2CISRC に値 110b が含まれているとき、CPU によって読み出される (STOP 条件が検出される)。I2CISRC のエミュレータ読み取りは、このビットに影響を与えません。 - SCD が手動でクリアされる。このビットをクリアするには、1 を書き込みます。 - I2C モジュールがリセットされる。 <p>1h (R/W) = I2C バスで STOP 条件が検出されました。</p>
4	XRDY	R	1h	<p>送信データ準備完了割り込みフラグ・ビット。FIFO モードでない場合、XRDY はデータ送信レジスタ (I2CDXR) が新しいデータを受け付ける準備ができていないことを示します。 FCM=0: 以前のデータが I2CDXR から送信シフト・レジスタ (I2CXHR) にコピーされたとき、CPU は XRDY をポーリングするか、XRDY 割り込み要求を使用できます。FIFO モードでは、代わりに TXFFINT を使用します。 FCM=1: XRDY は、次のデータが必要なおきのみアサートされ、I2CDXR への書き込みでアサート解除されます。FCM モードでは、ポーリングと割り込みベースの両方のデータ転送が許可されます。 リセット・タイプ:SYSRSn</p> <p>0h (R/W) = I2CDXR の準備ができていません。データが I2CDXR に書き込まれると、XRDY はクリアされます。</p> <p>1h (R/W) = I2CDXR 準備完了: データは I2CDXR から I2CXHR にコピーされました。 また、I2C モジュールがリセットされると、XRDY は強制的に 1 になります。</p>
3	RRDY	R/W1C	0h	<p>受信データ準備完了割り込みフラグ・ビット。 FIFO モードでないとき、RRDY は受信シフト・レジスタ (I2CRSR) から I2CDRR にデータがコピー済みで、データ受信レジスタ (I2CDRR) を読み取る準備ができていないことを示します。CPU は RRDY をポーリングするか、RRDY 割り込み要求を使用できます。FIFO モードでは、代わりに RXFFINT を使用します。 リセット・タイプ:SYSRSn</p> <p>0h (R/W) = I2CDRR の準備ができていません。RRDY は、次のいずれかのイベントによってクリアされます。</p> <ul style="list-style-type: none"> - I2CDRR が CPU によって読み取られる。I2CDRR のエミュレータ読み取りは、このビットに影響を与えません。 - RRDY が手動でクリアされる。このビットをクリアするには、1 を書き込みます。 - I2C モジュールがリセットされる。 <p>1h (R/W) = I2CDRR 準備完了: データは I2CRSR から I2CDRR にコピーされました。</p>

表 1-12. I2CSTR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
2	ARDY	R/W1C	0h	<p>レジスタ・アクセス準備完了割り込みフラグ・ビット (I2C モジュールがマスタ・モードのときのみ適用可能)。</p> <p>ARDY は、以前にプログラムされたアドレス、データ、コマンドの値が使用済みで、I2C モジュールのレジスタにアクセスする準備ができています。CPU は ARDY をポーリングするか、ARDY 割り込み要求を使用できます</p> <p>リセット・タイプ:SYSRSn</p> <p>0h (R/W) = レジスタにアクセスする準備ができていません。ARDY は、次のいずれかのイベントによってクリアされます。</p> <ul style="list-style-type: none"> - I2C モジュールが、現在のレジスタの内容を使用して起動する。 - ARDY が手動でクリアされる。このビットをクリアするには、1 を書き込みます。 - I2C モジュールがリセットされる。 <p>1h (R/W) = レジスタにアクセスする準備ができています。</p> <p>非反復モード (I2CMDR で RM = 0) では、次のようになります。I2CMDR で STP = 0 なら、内部データ・カウンタが 0 にカウントダウンすると ARDY ビットがセットされます。STP = 1 の場合、ARDY は影響を受けません (代わりに、カウンタが 0 に達したときに I2C モジュールが STOP 条件を生成します)。</p> <p>反復モード (RM = 1) では、次のようになります。ARDY は、I2CDXR から送信される各バイトの末尾に設定されます。</p>
1	NACK	R/W1C	0h	<p>非アクリッジ割り込みフラグ・ビット。</p> <p>NACK は、I2C モジュールがマスタ・トランスミッタのときに適用されます。</p> <p>NACK は、I2C モジュールがスレーブ・レシーバから、アクリッジ・ビット (ACK) と非アクリッジ・ビット (NACK) のどちらを検出したかを示します。CPU は NACK をポーリングするか、NACK 割り込み要求を使用できます</p> <p>リセット・タイプ:SYSRSn</p> <p>0h (R/W) = ACK を受信し、NACK を受信していません。このビットは、次のいずれかのイベントによってクリアされます。</p> <ul style="list-style-type: none"> - スレーブ・レシーバからアクリッジ・ビット (ACK) が送信される。 - NACK が手動でクリアされる。このビットをクリアするには、1 を書き込みます。 - CPU が割り込みソース・レジスタ (I2CISRC) を読み取り、レジスタに NACK 割り込みのコードが含まれている。I2CISRC のエミュレータ読み取りは、このビットに影響を与えません。 - I2C モジュールがリセットされる。 <p>1h (R/W) = NACK ビットが受信されました。ハードウェアは、非アクリッジ (NACK) ビットが受信されたことを検出しました。</p> <p>注: I2C モジュールがゼネラル・コール転送を行っているときは、1 つ以上のスレーブがアクリッジを送信しても、NACK は 1 のままです。</p>

表 1-12. I2CSTR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	ARBL	R/W1C	0h	<p>調停に負けた割り込みフラグ・ビット (I2C モジュールがマスタ・トランスミッタのときのみ適用可能)。</p> <p>ARBL は主に、I2C モジュールが他のマスタ・トランスミッタとの調停に負けたことを示します。CPU は ARBL をポーリングするか、ARBL 割り込み要求を使用できます。</p> <p>リセット・タイプ: SYSRSn</p> <p>0h (R/W) = 調停に負けていません。AL は、次のいずれかのイベントによってクリアされます。</p> <ul style="list-style-type: none"> - AL が手動でクリアされる。このビットをクリアするには、1 を書き込みます。 - CPU が割り込みソース・レジスタ (I2CISRC) を読み取り、レジスタに AL 割り込みのコードが含まれている。I2CISRC のエミュレータ読み取りは、このビットに影響を与えません。 - I2C モジュールがリセットされる。 <p>1h (R/W) = 調停に負けました。AL は、次のいずれかのイベントによって設定されます。</p> <ul style="list-style-type: none"> - I2C モジュールが、ほぼ同時に送信を開始した複数の競合トランスミッタとの調停に負けたことを検出した。 - BB (バス・ビジー) ビットが 1 にセットされているときに、I2C モジュールが転送の開始を試みる。 <p>AL が 1 になると、I2CMDR の MST ビットと STP ビットがクリアされ、I2C モジュールがスレーブ・レシーバになります。</p>

1.7.2.4 I2CCLKL レジスタ (オフセット = 3h) [リセット = 0h]

I2CCLKL を図 1-23 に示します。また、その説明を表 1-13 に示します。

[概略表](#)に戻ります。

I2C クロックの Low 時間分周器

図 1-23. I2CCLKL レジスタ

15	14	13	12	11	10	9	8
I2CCLKL							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCLKL							
R/W-0h							

表 1-13. I2CCLKL レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	I2CCLKL	R/W	0h	クロックの Low 時間の分周値。 マスタ・クロックの Low 期間を生成するには、モジュール・クロックの周期に (ICCL + d) を掛けます。d はプリスケアラに基づく調整係数です。詳細については、「はじめに」の「クロック・デバイダ・レジスタ」セクションを参照してください。 注: I2C クロックを正しく生成するには、これらのビットを 0 以外の値に設定する必要があります。 リセット・タイプ: SYRSn

1.7.2.5 I2CCLKH レジスタ (オフセット = 4h) [リセット = 0h]

I2CCLKH を図 1-24 に示します。また、その説明を表 1-14 に示します。

[概略表](#)に戻ります。

I2C クロックの High 時間分周器

図 1-24. I2CCLKH レジスタ

15	14	13	12	11	10	9	8
I2CCLKH							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCLKH							
R/W-0h							

表 1-14. I2CCLKH レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	I2CCLKH	R/W	0h	クロックの High 時間の分周値。 マスタ・クロックの High 期間を生成するには、モジュール・クロックの周期に (ICCL + d) を掛けます。d はプリスケアラに基づく調整係数です。詳細については、「はじめに」の「クロック・デバイダ・レジスタ」セクションを参照してください。 注: I2C クロックを正しく生成するには、これらのビットを 0 以外の値に設定する必要があります。 リセット・タイプ: SYRSRn

1.7.2.6 I2CCNT レジスタ (オフセット = 5h) [リセット = 0h]

I2CCNT を図 1-25 に示します。また、その説明を表 1-15 に示します。

概略表に戻ります。

I2CCNT は 16 ビットのレジスタで、I2C モジュールがトランスミッタとして構成されているときに送信する、またはマスタ・レシーバとして構成されているときに受信する、データ・バイトの数を示します。反復モード (RM = 1) では、I2CCNT は使用されません。

I2CCNT に書き込まれた値は、内部データ・カウンタにコピーされます。内部データ・カウンタは、転送されるバイトごとに 1 ずつデクリメントされます (I2CCNT は変更されません)。マスタ・モード (I2CMDR で STP = 1) で STOP 条件が要求された場合、I2C モジュールは、カウントダウンが完了した時点 (つまり、最後のバイトが転送された時点) で STOP 条件を示し、転送を終了します。

図 1-25. I2CCNT レジスタ

15	14	13	12	11	10	9	8
I2CCNT							
R/W-0h							
7	6	5	4	3	2	1	0
I2CCNT							
R/W-0h							

表 1-15. I2CCNT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-0	I2CCNT	R/W	0h	データ・カウント値。I2CCNT は、転送または受信するデータ・バイト数を示します。 STOP 条件が指定されているなら (STP=1)、各バイトが送信された後、I2CCNT は 0 に達するまでデクリメントされ、その後で STOP 条件が生成されます。 I2CMDR の RM ビットが 1 にセットされているとき、I2CCNT の値は無視されます。 リセット・タイプ: SYSRSn 0h (R/W) = データ・カウント値は 65536 1h (R/W) = データ・カウント値は 1 2h (R/W) = データ・カウント値は 2 FFFFh (R/W) = データ・カウント値は 65535

1.7.2.7 I2CDRR レジスタ (オフセット = 6h) [リセット = 0h]

I2CDRR を図 1-26 に示します。また、その説明を表 1-16 に示します。

概略表に戻ります。

I2CDRR は、CPU が受信データを読み取るために使用される 16 ビット・レジスタです。I2C モジュールは、1~8 ビットのデータ・バイトを受信できます。ビット数は、I2CMDR のビット数 (BC) ビットで選択されます。一度に 1 ビットずつ、SDA ピンから受信シフト・レジスタ (I2CRSR) にシフトインされます。完全なデータ・バイトが受信されると、I2C モジュールはデータ・バイトを I2CRSR から I2CDRR にコピーします。CPU は I2CRSR に直接アクセスできません。

I2CDRR に 8 ビット未満のデータ・バイトがある場合、データの値は右揃えされ、I2CDRR (7-0) の他のビットは未定義になります。たとえば、BC = 011 (3 ビットのデータ・サイズ) なら、受信データは I2CDRR (2-0) で、I2CDRR (7-3) の内容は未定義です。

受信 FIFO モードでは、I2CDRR レジスタが受信 FIFO バッファとして動作します。

図 1-26. I2CDRR レジスタ

15	14	13	12	11	10	9	8
予約済み							
R-0h							
7	6	5	4	3	2	1	0
DATA							
R-0h							

表 1-16. I2CDRR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	予約済み	R	0h	予約済み
7-0	データ	R	0h	受信データ リセット・タイプ: SYSRSn

1.7.2.8 I2CSAR レジスタ (オフセット = 7h) [リセット = 3FFh]

I2CSAR を図 1-27 に示します。また、その説明を表 1-17 に示します。

概略表に戻ります。

I2C スレーブ・アドレス・レジスタ (I2CSAR) は 16 ビットのレジスタで、I2C モジュールがマスタのとき、次に送信されるスレーブ・アドレスを格納します。I2CSAR の SAR フィールドには、7 ビットまたは 10 ビットのスレーブ・アドレスが含まれています。I2C モジュールがフリー・データ形式を使用していない (I2CMDR で FDF = 0) とき、このアドレスを使用してスレーブとのデータ転送を開始します。アドレスは 0 か、特定のスレーブのアドレスです。アドレスが 0 なら、すべてのスレーブへのゼネラル・コールとなります。7 ビット・アドレッシング・モードが選択されると (I2CMDR で XA = 0)、I2CSAR のビット 6-0 のみが使用され、ビット 9-7 には 0 を書き込みます。

図 1-27. I2CSAR レジスタ

15	14	13	12	11	10	9	8
予約済み						SAR	
R-0h						R/W-3FFh	
7	6	5	4	3	2	1	0
SAR							
R/W-3FFh							

表 1-17. I2CSAR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-10	予約済み	R	0h	予約済み
9-0	SAR	R/W	3FFh	7 ビット・アドレッシング・モード (I2CMDR で XA = 0): ビット 6-0 は 00h~7Fh で、I2C モジュールがマスタ・トランスミッタ・モードのときに送信する 7 ビットのスレーブ・アドレスです。ビット 9-7 には 0 を書き込みます。 10 ビット・アドレッシング・モード (I2CMDR で XA = 1): ビット 9-0 は 000h~3FFh で、I2C モジュールがマスタ・トランスミッタ・モードのときに送信する 10 ビットのスレーブ・アドレスです。 リセット・タイプ: SYSRSn

1.7.2.9 I2CDXR レジスタ (オフセット = 8h) [リセット = 0h]

I2CDXR を図 1-28 に示します。また、その説明を表 1-18 に示します。

概略表に戻ります。

CPU は送信データを I2CDXR に書き込みます。この 16 ビット・レジスタは、1~8 ビットのデータ・バイトを受け付けます。I2CDXR に書き込む前に、適切な値を I2CMDR のビット数 (BC) ビットにロードして、データ・バイトのビット数を指定します。8 ビットより少ないデータ・バイトを書き込むときは、I2CDXR の値が右揃えになっていることを確認してください。データ・バイトが I2CDXR に書き込まれた後で、I2C モジュールはデータ・バイトを送信シフト・レジスタ (I2CXSR) にコピーします。CPU は I2CXSR に直接アクセスできません。I2C モジュールは、データバイトを I2CXSR から 1 ビットずつ SDA ピンにシフトアウトします。

受信 FIFO モードでは、I2CDXR レジスタが送信 FIFO バッファとして動作します。

図 1-28. I2CDXR レジスタ

15	14	13	12	11	10	9	8
予約済み							
R-0h							
7	6	5	4	3	2	1	0
DATA							
R/W-0h							

表 1-18. I2CDXR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	予約済み	R	0h	予約済み
7-0	DATA	R/W	0h	送信データ リセット・タイプ: SYSRSn

1.7.2.10 I2CMR レジスタ (オフセット = 9h) [リセット = 0h]

I2CMR を図 1-29 に示します。また、その説明を表 1-19 に示します。

概略表に戻ります。

I2C モード・レジスタ (I2CMR) は、I2C モジュールの制御ビットを含む 16 ビット・レジスタです。

図 1-29. I2CMR レジスタ

15	14	13	12	11	10	9	8
NACKMOD	FREE	STT	予約済み	STP	MST	TRX	XA
R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RM	DLB	IRS	STB	FDL	BC		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

表 1-19. I2CMR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	NACKMOD	R/W	0h	<p>NACK モード・ビット。このビットは、I2C モジュールがレシーバとして動作しているときのみ適用されます。</p> <p>リセット・タイプ: SYRSn</p> <p>0h (R/W) = スリープ・レシーバ・モードの場合: I2C モジュールは、バスのアクリッジ・サイクルごとに、アクリッジ (ACK) ビットをトランスミッタに送信します。I2C モジュールは、NACKMOD ビットがセットされているときのみ、非アクリッジ (NACK) ビットを送信します。</p> <p>マスタ・レシーバ・モードのとき: I2C モジュールは、内部データ・カウンタが 0 にカウントダウンするまで、アクリッジ・サイクルごとに ACK ビットを送信します。この時点で、I2C モジュールは NACK ビットをトランスミッタに送信します。NACK ビットを早期に送信するには、NACKMOD ビットを設定する必要があります。</p> <p>1h (R/W) = スリープ・レシーバまたはマスタ・レシーバ・モードで: I2C モジュールは、バスの次のアクリッジ・サイクルで、トランスミッタに NACK ビットを送信します。NACK ビットが送信されると、NACKMOD はクリアされます。</p> <p>重要: 次のアクリッジ・サイクルで NACK ビットを送信するには、最後のデータ・ビットの立ち上がりエッジより前に NACKMOD を設定する必要があります。</p>
14	FREE	R/W	0h	<p>このビットは、デバッグ・ブレイクポイントが発生したときに I2C モジュールが実行するアクションを制御します。</p> <p>リセット・タイプ: SYRSn</p> <p>0h (R/W) = I2C モジュールがマスタのとき: ブレイクポイントが発生したときに SCL が Low なら、I2C モジュールは即座に停止し、I2C モジュールがトランスミッタかレシーバかにかかわらず、SCL を Low に駆動し続けます。SCL が High なら、I2C モジュールは SCL が Low になるまで待機し、その後停止します。</p> <p>I2C モジュールがスリープのとき: ブレイクポイントが発生すると、現在の送受信が完了したときに I2C モジュールが強制的に停止します。</p> <p>1h (R/W) = I2C モジュールはフリーで動作します。 すなわち、ブレイクポイントが発生しても動作を継続します。</p>
13	STT	R/W	0h	<p>START 条件ビット (I2C モジュールがマスタのときのみ適用可能)。RM、STT、STP ビットは、I2C モジュールがデータ送信を開始および停止するタイミングを決定します (表 9-6 を参照)。STT ビットと STP ビットを使用して反復モードを終了でき、IRS = 0 のときはこのビットに書き込みできないことに注意してください。</p> <p>リセット・タイプ: SYRSn</p> <p>0h (R/W) = マスタ・モードでは、START 条件が生成された後、STT は自動的にクリアされます。</p> <p>1h (R/W) = マスタ・モードで STT を 1 にセットすると、I2C モジュールは I2C バスに START 条件を生成します</p>

表 1-19. I2CMDR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
12	予約済み	R	0h	予約済み
11	STP	R/W	0h	<p>STOP 条件ビット (I2C モジュールがマスタのときのみ適用可能)。 マスタ・モードでは、RM、STT、STP ビットによって、I2C モジュールがデータ送信を開始および停止するタイミングが決定されます。 STT ビットと STP ビットを使用して反復モードを終了でき、IRS = 0 のときはこのビットに書き込めないことに注意してください。非反復モードでは、STOP 条件を生成する前に少なくとも 1 バイトを転送する必要があります。I2C モジュールは、I2CSTR[SCD] ビットがセットされるまで、このビットのクリアを遅延させます。I2C ステート・マシンの中断を避けるため、ユーザーはこのビットがクリアされるまで待つてから、新しいメッセージを開始する必要があります。 リセット・タイプ: SYSRSn 0h (R/W) = STOP 条件が生成された後で、STP は自動的にクリアされます 1h (R/W) = I2C モジュールの内部データ・カウンタが 0 までカウントダウンされたときに STOP 条件を生成するよう、デバイスによって STP が設定されています。</p>
10	MST	R/W	0h	<p>マスタ・モード・ビット。 MST は、I2C モジュールがスレーブ・モードかマスタ・モードかを決定します。I2C マスタが STOP 条件を生成すると、MST は自動的に 1 から 0 に変更されます リセット・タイプ: SYSRSn 0h (R/W) = スレーブ・モード。I2C モジュールはスレーブで、マスタからシリアル・クロックを受信します。 1h (R/W) = マスタ・モード。I2C モジュールはマスタで、SCL ピンにシリアル・クロックを生成します。</p>
9	TRX	R/W	0h	<p>トランスミッタ・モード・ビット。 該当するとき、TRX は I2C モジュールがトランスミッタ・モードかレシーバ・モードかを選択します。 リセット・タイプ: SYSRSn 0h (R/W) = レシーバ・モード。I2C モジュールはレシーバで、データを SDA ピンで受信します。 1h (R/W) = トランスミッタ・モード。I2C モジュールはトランスミッタで、データを SDA ピンで送信します。</p>
8	XA	R/W	0h	<p>拡張アドレス・イネーブル・ビット。 リセット・タイプ: SYSRSn 0h (R/W) = 7 ビット・アドレッシング・モード (通常アドレス・モード)。I2C モジュールは 7 ビットのスレーブ・アドレスを送信し (I2CSAR のビット 6-0 から)、独自スレーブ・アドレスは 7 ビット (I2COAR のビット 6-0) です。 1h (R/W) = 10 ビット・アドレッシング・モード (拡張アドレス・モード)。I2C モジュールは 10 ビットのスレーブ・アドレスを送信し (I2CSAR のビット 9-0 から)、独自スレーブ・アドレスは 10 ビット (I2COAR のビット 9-0) です。</p>
7	RM	R/W	0h	<p>反復モード・ビット (I2C モジュールがマスタ・トランスミッタの場合のみ適用可能) RM、STT、STP ビットは、I2C モジュールがデータ送信を開始および停止するタイミングを決定します リセット・タイプ: SYSRSn 0h (R/W) = 非反復モード。データ・カウント・レジスタ (I2CCNT) の値によって、I2C モジュールが送受信するバイト数が決まります。 1h (R/W) = 反復モード。STP ビットが手動でセットされるまで、I2CDXR レジスタが書き込まれるたびに (または、FIFO モードでは送信 FIFO が空になるまで)、データ・バイトが送信されます。I2CCNT の値は無視されず。ARDY ビット / 割り込みを使用して、I2CDXR (または FIFO) が追加データを受け付けられるタイミング、またはデータがすべて送信され、CPU が STP ビットに書き込みできるタイミングを判定できます。</p>

表 1-19. I2CMDR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	DLB	R/W	0h	デジタル・ループバック・モード・ビット。 リセット・タイプ: SYSRSn 0h (R/W) = デジタル・ループバック・モードがディセーブル。 1h (R/W) = デジタル・ループバック・モードがイネーブル。このモードで正常に動作するには、MST ビットが 1 の必要があります。 デジタル・ループバック・モードでは、I2CDXR から送信されたデータは、n デバイス・サイクル後に内部バスにより I2CDRR で受信されます。ここで、 $n = ((I2C \text{ 入力クロック周波数} / \text{モジュール・クロック周波数}) \times 8)$ 送信クロックは受信クロックでもあります。SDA ピンで送信されるアドレスは、I2COAR のアドレスです。 注: デジタル・ループバック・モードでは、フリー・データ形式 (FDF = 1) はサポートされません。
5	IRS	R/W	0h	I2C モジュール・リセット・ビット。 リセット・タイプ: SYSRSn 0h (R/W) = I2C モジュールはリセット / ディセーブル状態です。このビットが 0 にクリアされると、すべてのステータス・ビット (I2CSTR) がそれぞれのデフォルト値に設定されます。 1h (R/W) = I2C モジュールがイネーブル。これは、I2C ペリフェラルが I2C バスを保持しているときに、I2C バスを解放する効果があります。
4	STB	R/W	0h	START バイト・モード・ビット。このビットは、I2C モジュールがマスタのときのみ適用されます。Philips Semiconductors の I2C バス仕様のバージョン 2.1 で説明されているように、START バイトを使用して、START 条件を検出するために余分な時間を必要とするスレーブを支援できます。I2C モジュールがスレーブのとき、STB ビットの値に関係なく、マスタからの START バイトは無視されます。 リセット・タイプ: SYSRSn 0h (R/W) = I2C モジュールは START バイト・モードではありません。 1h (R/W) = I2C モジュールは START バイト・モードです。START 条件ビット (STT) を設定すると、I2C モジュールは START 条件以外のものを含む転送を開始します。具体的には、次のものが生成されます。 1. SATR 条件 2. START バイト (0000 0001b) 3. ダミーのアクノリッジ・クロック・パルス 4. 反復 START 条件 その後で、I2C モジュールは通常と同じに I2CSAR 内のスレーブ・アドレスを送信します。
3	FDF	R/W	0h	フリー・データ形式モード・ビット。 リセット・タイプ: SYSRSn 0h (R/W) = フリー・データ形式モードがディセーブル。XA ビットで選択される 7/10 ビットのアドレッシング形式が転送に使用されます。 1h (R/W) = フリー・データ形式モードがイネーブル。セクション 9.2.5 で説明しているフリー・データ (アドレスなし) 形式が転送に使用されます。 デジタル・ループバック・モード (DLB=1) では、フリー・データ形式はサポートされていません。

表 1-19. I2CMDR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
2-0	BC	R/W	0h	<p>ビット・カウント・ビット。 BC は、I2C モジュールによって次に受信または送信される、データ・バイトのビット数 (1~8) を定義します。BC で選択されるビット数は、他のデバイスのデータ・サイズと一致している必要があります。BC = 000b のとき、データ・バイトは 8 ビットであることに注意してください。アドレス・バイトは常に 8 ビットなので、BC の影響を受けません。</p> <p>注: ビット数が 8 未満なら、受信データは I2CDRR (7-0) で右揃えされ、空きビットは未定義です。また、I2CDXR に書き込まれる送信データも右揃えの必要があります</p> <p>リセット・タイプ: SYSRSn</p> <p>0h (R/W) = データバイトごとに 8 ビット 1h (R/W) = データバイトごとに 1 ビット 2h (R/W) = データバイトごとに 2 ビット 3h (R/W) = データバイトごとに 3 ビット 4h (R/W) = データバイトごとに 4 ビット 5h (R/W) = データバイトごとに 5 ビット 6h (R/W) = データバイトごとに 6 ビット 7h (R/W) = データバイトごとに 7 ビット</p>

1.7.2.11 I2CISRC レジスタ (オフセット = Ah) [リセット= 0h]

I2CISRC を図 1-30 に示します。また、その説明を表 1-20 に示します。

概略表に戻ります。

I2C 割り込みソース・レジスタ (I2CISRC) は、I2C 割り込みを生成したイベントを判定するために CPU で使用される 16 ビット・レジスタです。

図 1-30. I2CISRC レジスタ

15	14	13	12	11	10	9	8
予約済み				WRITE_ZEROS			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
予約済み				INTCODE			
R-0h				R-0h			

表 1-20. I2CISRC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-12	予約済み	R	0h	予約済み
11-8	WRITE_ZEROS	R/W	0h	TI の内部テスト・ビット これらの予約済みビット位置には、常に 0 を書き込みます。 リセット・タイプ:SYSRSn
7-3	予約済み	R	0h	予約済み
2-0	INTCODE	R	0h	割り込みコード・ビット。 INTCODE のバイナリ・コードは、I2C 割り込みを生成したイベントを示します。 CPU 読み取りにより、このフィールドはクリアされます。優先順位の低い別の割り込みが保留中でイネーブルなら、その割り込みに対応する値がロードされます。それ以外の場合、値はクリアされたままです。 以下の割り込みイベントは、優先順位の高いものから低いものへの順にリストされています。つまり、INTCODE 1 (調停に負けた) は最高の優先順位で、INTCODE 7 (スレーブとしてアドレス指定された) は最低の優先順位です。 調停に負けた、非アクリッジの条件が検出された、または STOP 条件が検出された場合、CPU の読み取りによって、I2CSTR レジスタの関連する割り込みフラグ・ビットもクリアされます。 エミュレータの読み取りは、このフィールドや、I2CSTR レジスタのステータス・ビットの状態には影響を与えません。 リセット・タイプ:SYSRSn 0h (R/W) = なし 1h (R/W) = 調停に負けた 2h (R/W) = 非アクリッジ条件が検出された 3h (R/W) = レジスタへのアクセス準備完了 4h (R/W) = 受信データ準備完了 5h (R/W) = 送信データ準備完了 6h (R/W) = STOP 条件が検出された 7h (R/W) = スレーブとしてアドレス指定された

1.7.2.12 I2CEMDR レジスタ (オフセット = Bh) [リセット = 1h]

I2CEMDR を図 1-31 に示します。また、その説明を表 1-21 に示します。

[概略表](#)に戻ります。

I2C 拡張モード

図 1-31. I2CEMDR レジスタ

15	14	13	12	11	10	9	8
予約済み							
R-0h							
7	6	5	4	3	2	1	0
予約済み						FCM	BC
R-0h						R/W-0h	R/W-1h

表 1-21. I2CEMDR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-2	予約済み	R	0h	予約済み
1	FCM	R/W	0h	上位互換モード。 このビットをプログラムすると、非 FIFO モードでの Tx バッファのデータ・ステータスに関係なく、Tx データが必要な場合のみ Tx 要求の機能が有効になります。 このレジスタは XRDY の動作に影響するため、IRS (I2CMDR[5]) を解放した後で設定する必要があります。 リセット・タイプ:SYSRSn 0h (R/W) = 従来の機能で、バッファをシフト・レジスタにコピーするとき、または START 条件がアクティブなときに Tx データを要求します。不正な起動、調停に負けた、NACK 条件の後では、古いデータが再利用されます。 1h (R/W) = 新しい機能で、ACK (アドレス / データ) がアクティブなときのみデータを要求します。
0	BC	R/W	1h	下位互換モード。 このビットは、スレーブ・トランスミッタ・モードでの I2CSTR レジスタの送信ステータス・ビット (XRDY および XSMT) のタイミングに影響を及ぼします。 リセット・タイプ:SYSRSn 0h (R/W) = 詳細については、「下位互換モード・ビット、スレーブ・トランスミッタ」の図を参照してください。 1h (R/W) = 詳細については、「下位互換モード・ビット、スレーブ・トランスミッタ」の図を参照してください。

1.7.2.13 I2CPSC レジスタ (オフセット = Ch) [リセット = 0h]

I2CPSC を図 1-32 に示します。また、その説明を表 1-22 に示します。

概略表に戻ります。

I2C プリスケアラ・レジスタ (I2CPSC) は、I2C 入力クロックを分周して I2C モジュールの動作に必要なモジュール・クロックを取得するために使用される、16 ビット・レジスタ (図 14-21 を参照) です。モジュールのクロック周波数としてサポートされている範囲の値については、デバイス固有のデータ・マニュアルを参照してください。

IPSC は、I2C モジュールがリセット中 (I2CMDR で IRS = 0) に初期化する必要があります。プリスケールされた周波数は、IRS が 1 に変更されたときのみ有効になります。IRS = 1 で IPSC 値を変更しても効果はありません。

図 1-32. I2CPSC レジスタ

15	14	13	12	11	10	9	8
予約済み							
R-0h							
7	6	5	4	3	2	1	0
IPSC							
R/W-0h							

表 1-22. I2CPSC レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-8	予約済み	R	0h	予約済み
7-0	IPSC	R/W	0h	I2C プリスケアラの分周値。 IPSC は、I2C モジュールのモジュール・クロックを生成するために、CPU クロックをどれだけ分割するかを決定します。 モジュール・クロック周波数 = I2C 入力クロック周波数 / (IPSC + 1) 注: IPSC は、I2C モジュールがリセット中 (I2CMDR で IRS = 0) に初期化する必要があります。 リセット・タイプ: SYSRSn

1.7.2.14 I2CFFTX レジスタ (オフセット = 20h) [リセット = 0h]

I2CFFTX を図 1-33 に示します。また、その説明を表 1-23 に示します。

概略表に戻ります。

I2C 送信 FIFO レジスタ (I2CFFTX) は 16 ビットのレジスタで、I2C FIFO モードのイネーブル・ビットと、I2C ペリフェラルで動作する送信 FIFO モード用の制御およびステータス・ビットが含まれています。

図 1-33. I2CFFTX レジスタ

15	14	13	12	11	10	9	8
予約済み	I2CFFEN	TXFFRST					TXFFST
R-0h	R/W-0h	R/W-0h					R-0h
7	6	5	4	3	2	1	0
TXFFINT	TXFFINTCLR	TXFFIENA					TXFFIL
R-0h	R-0/W1S-0h	R/W-0h					R/W-0h

表 1-23. I2CFFTX レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	予約済み	R	0h	予約済み
14	I2CFFEN	R/W	0h	I2C FIFO モード・イネーブル・ビット。 送信 FIFO と受信 FIFO のどちらかを正しく動作させるには、このビットをイネーブルする必要があります。 リセット・タイプ: SYSRSn 0h (R/W) = I2C FIFO モードをディセーブルします。 1h (R/W) = I2C FIFO モードをイネーブルします。
13	TXFFRST	R/W	0h	送信 FIFO リセット リセット・タイプ: SYSRSn 0h (R/W) = 送信 FIFO ポインタを 0000 にリセットし、送信 FIFO を RESET 状態に保持します。 1h (R/W) = 送信 FIFO 動作をイネーブルします。
12-8	TXFFST	R	0h	送信 FIFO のステータスが含まれています。 xxxxx 送信 FIFO には xxxxx バイトが含まれています。 00000 送信 FIFO は空です。 注: これらのビットは 0 にリセットされるため、送信 FIFO 動作がイネーブルされ、I2C がリセットから復帰すると、送信 FIFO 割り込みフラグがセットされます。これにより、送信 FIFO 割り込みがイネーブルなら、生成されます。これによる悪影響を回避するため、送信 FIFO 動作がイネーブルされ、I2C がリセットから復帰した後で、TXFFINTCLR に 1 を書き込みます。 リセット・タイプ: SYSRSn
7	TXFFINT	R	0h	送信 FIFO 割り込みフラグ。 このビットは、CPU が TXFFINTCLR ビットに 1 を書き込むことでクリアされます。TXFFIENA ビットがセットされているなら、このビットがセットされたとき割り込みが発生します。 リセット・タイプ: SYSRSn 0h (R/W) = 送信 FIFO 割り込み条件が発生していません。 1h (R/W) = 送信 FIFO 割り込み条件が発生しました。
6	TXFFINTCLR	R-0/W1S	0h	送信 FIFO 割り込みフラグのクリア リセット・タイプ: SYSRSn 0h (R/W) = 0 の書き込みは影響を与えません。読み出しでは 0 が返されます。 1h (R/W) = このビットに 1 を書き込むと、TXFFINT フラグがクリアされます。

表 1-23. I2CFFTX レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	TXFFIENA	R/W	0h	送信 FIFO 割り込みイネーブル リセット・タイプ:SYSRSn 0h (R/W) = ディセーブル。TXFFINT フラグは、設定時に割り込みを生成しません。 1h (R/W) = イネーブル。TXFFINT フラグをセットすると、割り込みが生成されます。
4-0	TXFFIL	R/W	0h	送信 FIFO 割り込みレベル。 これらのビットは、送信割り込みフラグをセットするステータス・レベルを設定します。TXFFST の 4-0 ビットがこれらのビット以下の値に達すると、TXFFINT フラグがセットされます。TXFFIENA ビットがセットされていると、割り込みが発生します。このデバイスの I2C には 16 レベルの送信 FIFO があるため、これらのビットは 16 を超える FIFO レベルの割り込み用に構成できません。 リセット・タイプ:SYSRSn

1.7.2.15 I2CFFRX レジスタ (オフセット = 21h) [リセット = 0h]

I2CFFRX を図 1-34 に示します。また、その説明を表 1-24 に示します。

概略表に戻ります。

I2C 受信 FIFO レジスタ (I2CFFRX) は 16 ビットのレジスタで、I2C ペリフェラルでの受信 FIFO 動作モードの制御およびステータス・ビットが含まれています。

図 1-34. I2CFFRX レジスタ

15		14		13		12		11		10		9		8	
予約済み				RXFFRST				RXFFST							
R-0h				R/W-0h				R-0h							
7		6		5		4		3		2		1		0	
RXFFINT		RXFFINTCLR		RXFFIENA						RXFFIL					
R-0h		R-0/W1S-0h		R/W-0h						R/W-0h					

表 1-24. I2CFFRX レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15-14	予約済み	R	0h	予約済み
13	RXFFRST	R/W	0h	I2C 受信 FIFO リセット・ビット リセット・タイプ:SYSRSn 0h (R/W) = 受信 FIFO ポインタを 0000 にリセットし、受信 FIFO を RESET 状態に保持します。 1h (R/W) = 受信 FIFO 動作をイネーブルします。
12-8	RXFFST	R	0h	受信 FIFO のステータスが含まれています: xxxxx 受信 FIFO には xxxxx バイトが含まれています 00000 受信 FIFO は空です。 リセット・タイプ:SYSRSn
7	RXFFINT	R	0h	受信 FIFO 割り込みフラグ。 このビットは、CPU が RXFFINTCLR ビットに 1 を書き込むことでクリアされます。RXFFIENA ビットがセットされているなら、このビットがセットされたとき割り込みが発生します リセット・タイプ:SYSRSn 0h (R/W) = 受信 FIFO 割り込み条件が発生していません。 1h (R/W) = 受信 FIFO 割り込み条件が発生しました。
6	RXFFINTCLR	R-0/W1S	0h	受信 FIFO 割り込みフラグ・クリア・ビット。 リセット・タイプ:SYSRSn 0h (R/W) = 0 の書き込みは影響を与えません。読み出しでは 0 が返されます。 1h (R/W) = このビットに 1 を書き込むと、RXFFINT フラグがクリアされます。
5	RXFFIENA	R/W	0h	受信 FIFO 割り込みイネーブル・ビット。 リセット・タイプ:SYSRSn 0h (R/W) = ディセーブル。RXFFINT フラグがセットされたとき、割り込みが発生しません。 1h (R/W) = イネーブル。RXFFINT フラグがセットされたとき、割り込みが発生します。

表 1-24. I2CFFRX レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
4-0	RXFFIL	R/W	0h	受信 FIFO 割り込みレベル・ これらのビットは、受信割り込みフラグをセットするステータス・レベルを設定します。RXFFST の 4-0 ビットがこれらのビット以下の値に達すると、RXFFINT フラグがセットされます。RXFFIENA ビットがセットされていると、割り込みが発生します。 注: これらのビットは 0 にリセットされるため、受信 FIFO 動作がイネーブルされ、I2C がリセットから復帰すると、受信 FIFO 割り込みフラグがセットされます。これにより、受信 FIFO 割り込みがイネーブルなら、発生します。これを回避するには、RXFFRST ビットをセットするときと同じ命令で、またはその前に、これらのビットを変更します。このデバイスの I2C には 16 レベルの受信 FIFO があるため、これらのビットは 16 を超える FIFO レベルの割り込み用に構成できません。 リセット・タイプ: SYSRSn

1.7.3 I2C レジスタから Driverlib への関数

表 1-25. I2C レジスタから Driverlib への関数

ファイル	Driverlib 関数
OAR	
i2c.h	I2C_setOwnSlaveAddress
IER	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
STR	
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_isBusBusy
i2c.h	I2C_getStatus
i2c.h	I2C_clearStatus
CLKL	
i2c.c	I2C_initMaster
CLKH	
i2c.c	I2C_initMaster
CNT	
i2c.h	I2C_setDataCount
DRR	
i2c.h	I2C_getData
SAR	
i2c.h	I2C_setSlaveAddress
DXR	
i2c.h	I2C_putData
MDR	
i2c.h	I2C_enableModule
i2c.h	I2C_disableModule
i2c.h	I2C_setConfig
i2c.h	I2C_setBitCount
i2c.h	I2C_sendStartCondition
i2c.h	I2C_sendStopCondition
i2c.h	I2C_sendNACK

表 1-25. I2C レジスタから Driverlib への関数 (continued)

ファイル	Driverlib 関数
i2c.h	I2C_getStopConditionStatus
i2c.h	I2C_setAddressMode
i2c.h	I2C_setEmulationMode
i2c.h	I2C_enableLoopback
i2c.h	I2C_disableLoopback
ISRC	
i2c.h	I2C_getInterruptSource
EMDR	
i2c.h	I2C_setExtendedMode
PSC	
i2c.c	I2C_initMaster
i2c.c	I2C_configureModuleFrequency
i2c.h	I2C_getPreScaler
FFTX	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_enableFIFO
i2c.h	I2C_disableFIFO
i2c.h	I2C_setFIFOInterruptLevel
i2c.h	I2C_getFIFOInterruptLevel
i2c.h	I2C_getTxFIFOStatus
FFRX	
i2c.c	I2C_enableInterrupt
i2c.c	I2C_disableInterrupt
i2c.c	I2C_getInterruptStatus
i2c.c	I2C_clearInterruptStatus
i2c.h	I2C_enableFIFO
i2c.h	I2C_disableFIFO
i2c.h	I2C_setFIFOInterruptLevel
i2c.h	I2C_getFIFOInterruptLevel
i2c.h	I2C_getRxFIFOStatus

Technical Reference Manual



Literature Number: JAJU890A
OCTOBER 2020 – REVISED MARCH 2023



この章では、シリアル通信インターフェイス (SCI) モジュールの機能と動作について説明します。SCI は、一般的には UART と呼ばれる 2 線式非同期シリアル・ポートです。SCI モジュールは、CPU と、標準 NRZ (non-return-to-zero) フォーマットを使用する他の非同期ペリフェラルとの間のデジタル通信をサポートします。SCI のレシーバおよびトランスミッタは、処理オーバーヘッドを削減するための 16 段 FIFO を備えており、それぞれ独自のイネーブル・ビットと割り込みビットがあります。この両方が、半二重通信用として個別に動作することも、全二重通信用として同時に動作することもできます。

データの整合性を確保するために、SCI は、受信データのブレイク検出、パリティ、オーバーラン、およびフレーミングエラーをチェックします。ビット・レートは、16 ビットのポー選択レジスタにより、さまざまな速度にプログラム可能です。

1.1 概要.....	4
1.2 アーキテクチャ.....	5
1.3 SCI モジュール信号の概要.....	5
1.4 デバイス・ピンの構成.....	7
1.5 マルチプロセッサおよび非同期通信モード.....	7
1.6 SCI プログラマブル・データ・フォーマット.....	8
1.7 SCI マルチプロセッサ通信.....	9
1.8 アイドル・ライン・マルチプロセッサ・モード.....	10
1.9 アドレス・ビット・マルチプロセッサ・モード.....	12
1.10 SCI 通信フォーマット.....	13
1.11 SCI ポート割り込み.....	16
1.12 SCI ポーレートの計算.....	16
1.13 SCI 拡張機能.....	17
1.14 ソフトウェア.....	20
1.15 SCI レジスタ.....	22

1.1 概要

図 1-1 に SCI インターフェイスを示します。

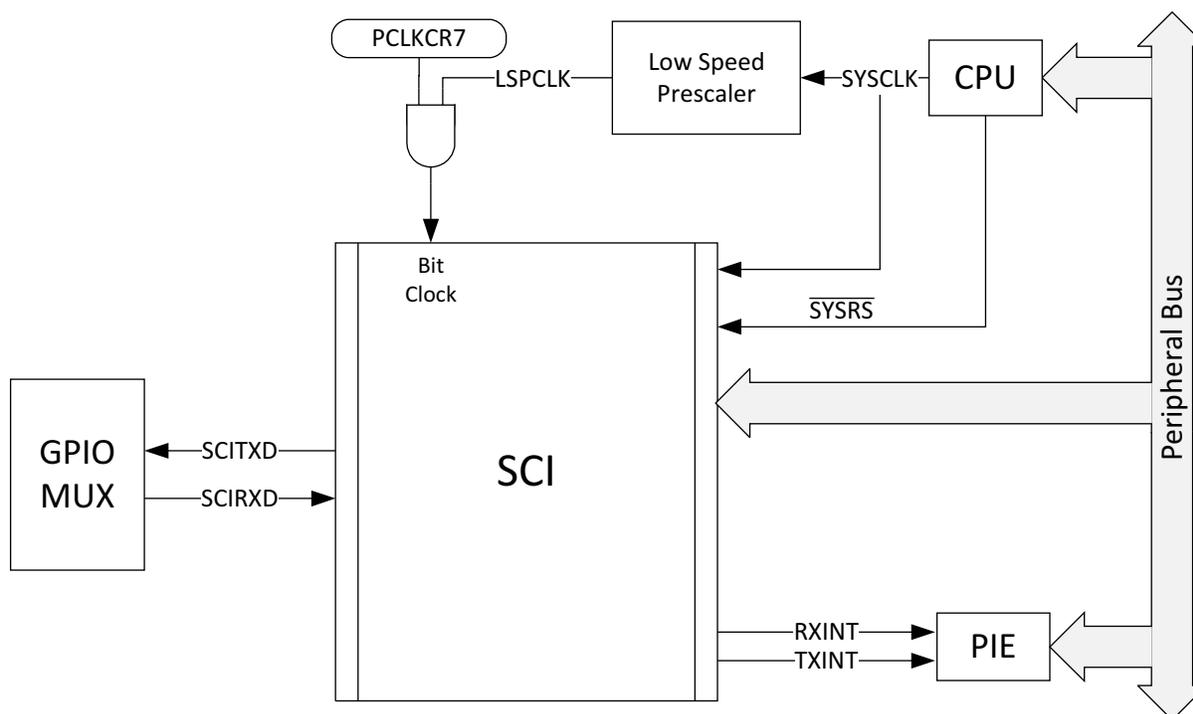


図 1-1. SCI CPU インターフェイス

1.1.1 SCI 関連資料

基礎資料

- [C2000 Academy - SCI](#)
- [1 分間で RS-485 を紹介 \(ビデオ\)](#)
- [RS-232、RS-422、RS-485 : 違い\(ビデオ\)](#)

入門資料

- [\[FAQ\] C2000 SCI がデータを正しく送受信しない場合、どのように修正すればよいですか?](#)

1.1.2 特長

SCI モジュールには次のような特長があります。

- 2 本の外部ピン (SCI に使用しない場合は、いずれのピンも GPIO として使用可能) :
 - SCITXD : SCI 送信出力ピン
 - SCIRXD : SCI 受信入力ピン
- 64K までの異なるレートにプログラム可能なボーレート
- データ・ワード・フォーマット
 - 1 スタート・ビット
 - データ・ワード (1~8 ビットの範囲でワード長をプログラム可能)
 - パリティ・ビット (偶数 / 奇数 / なしを選択可能)
 - 1 または 2 ストップ・ビット
 - アドレスをデータと区別するための追加ビット (アドレス・ビット・モードのみ)
- 4 つのエラー検出フラグ : パリティ、オーバーラン、フレーム、ブレイク検出
- 2 つのウェイクアップ・マルチプロセッサ・モード : アイドル・ラインおよびアドレス・ビット
- 半二重または全二重動作

- ダブル・バッファ付きの受信および送信機能
- トランスミッタおよびレシーバの動作は、ステータス・フラグによる割り込み駆動、またはポーリング・アルゴリズムによって実現可能
- トランスミッタおよびレシーバの割り込み用に個別のイネーブル・ビット (BRKDT を除く)
- NRZ (non-return-to-zero) フォーマット

拡張機能：

- 自動ポー検出ハードウェア・ロジック
- 16 段の送信 / 受信 FIFO

1.1.3 ブロック図

図 1-2 に、SCI モジュールのブロック図を示します。SCI ポートの動作は、[セクション 1.15](#) に示すレジスタによって設定および制御されます。

1.2 アーキテクチャ

図 1-2 に、全二重動作で使用される主な要素を示します。次のものが含まれます。

- トランスミッタ (TX) とその主要なレジスタ (図 1-2 の上半分)
 - SCITXBUF – 送信データ・バッファ・レジスタ。送信するデータ (CPU によってロードされる) が入っています
 - TXSHF レジスタ – 送信シフト・レジスタ。SCITXBUF レジスタからデータを受け取って、データを一度に 1 ビットずつシフトして SCITXD ピンに出力します
- レシーバ (RX) とその主要なレジスタ (図 1-2 の下半分)
 - RXSHF レジスタ – 受信シフト・レジスタ。データを SCIRXD ピンから 1 ビットずつシフトインします
 - SCIRXBUF – 受信データ・バッファ・レジスタ。このデータを CPU が読み取ります。リモート・プロセスからのデータは RXSHF レジスタにロードされ、次に SCIRXBUF レジスタおよび SCIRXEMU レジスタにロードされます。
- プログラム可能なポー・ジェネレータ
- 制御およびステータス・レジスタ

SCI レシーバとトランスミッタは、独立して動作するか、または同時に動作できます。

1.3 SCI モジュール信号の概要

表 1-1 に、各 SCI 信号の概要を示します。

表 1-1. SCI モジュール信号の概要

信号名	説明
外部信号	
SCIRXD	SCI 非同期シリアル・ポート受信データ
SCITXD	SCI 非同期シリアル・ポート送信データ
制御	
ポー・クロック	LSPCLK プリスケール・クロック
割り込み信号	
TXINT	送信割り込み
RXINT	受信割り込み

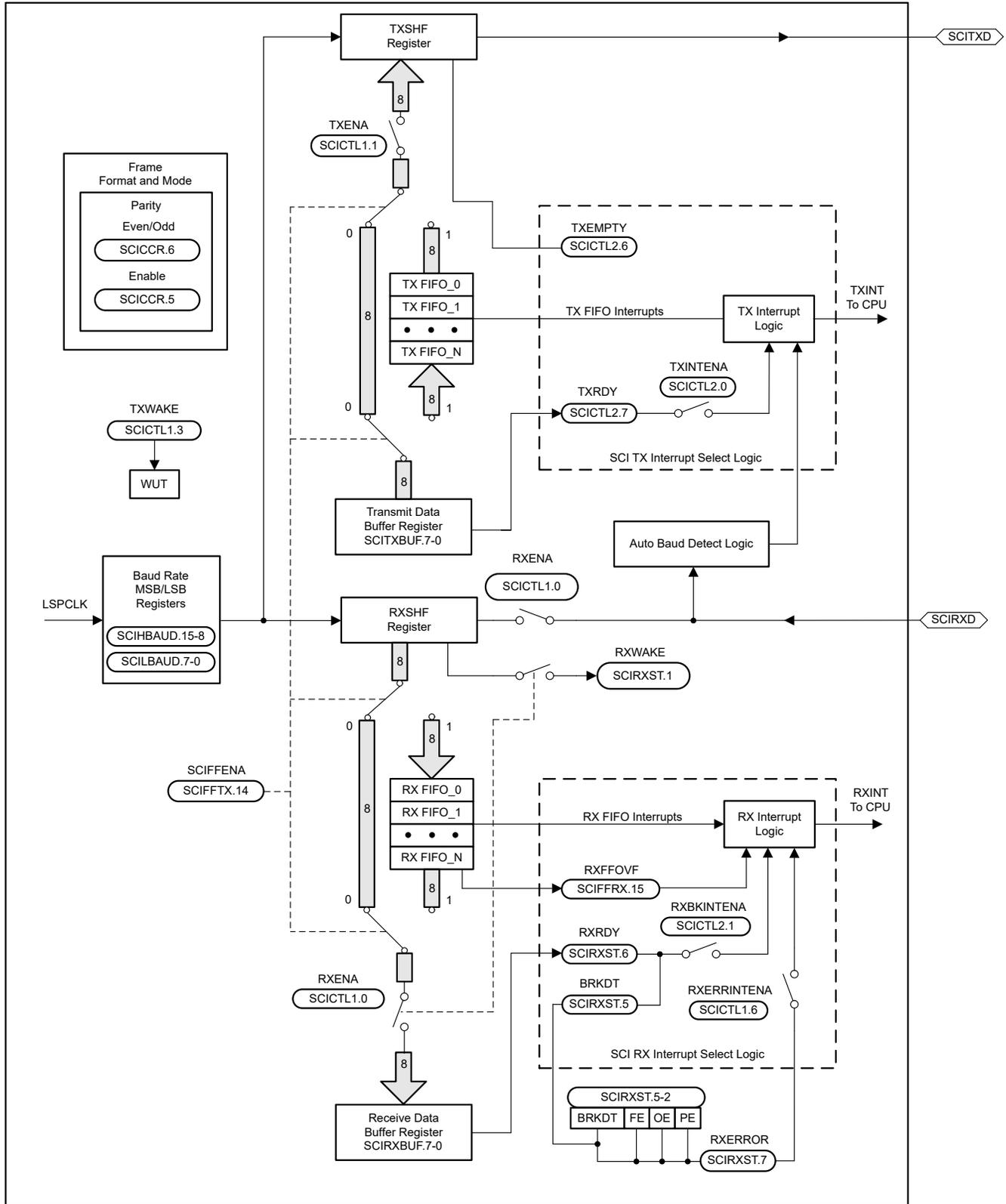


図 1-2. シリアル通信インターフェイス (SCI) モジュールのブロック図

1.4 デバイス・ピンの構成

このペリフェラルをデバイス・ピンに接続するには、GPIO マルチプレクサ・レジスタを構成する必要があります。ピン上のグリッチを回避するには、まず GPyGMUX ビットを設定し (対応する GPyMUX ビットはデフォルトの 0 のまま)、その後 GPyMUX レジスタに目的の値を書き込む必要があります。

一部の IO 機能は、このペリフェラルとは独立した GPIO レジスタ設定によって定義されます。入力信号では、適切な GPxQSELn レジスタ・ビットを 11b に設定して、GPIO 入力認定を非同期モードに設定する必要があります。内部プルアップは GPyPUD レジスタで構成できます。

GPIO マルチプレクサと設定の詳細については、「汎用入出力 (GPIO)」の章を参照してください。

1.5 マルチプロセッサおよび非同期通信モード

SCI には、アイドル・ライン・マルチプロセッサ・モード ([セクション 1.8](#) を参照) とアドレス・ビット・マルチプロセッサ・モード ([セクション 1.9](#) を参照) の 2 つのマルチプロセッサ・プロトコルがあります。これらのプロトコルにより、複数のプロセッサ間で効率的なデータ転送が可能になります。

多くの一般的なペリフェラルとのインターフェイスを実現するため、SCI には UART (Universal Asynchronous Receiver/Transmitter) 通信モードがあります。非同期モード ([セクション 1.10](#) を参照) では、RS-232-C フォーマットを使用する端末やプリンタなど、多くの標準デバイスとインターフェイスをとるために 2 本のラインが必要です。データ転送特性は以下のとおりです。

- 1 スタート・ビット
- 1~8 データ・ビット
- 偶数 / 奇数パリティ・ビット、またはパリティ・ビットなし
- 1 または 2 ストップ・ビット

1.6 SCI プログラマブル・データ・フォーマット

SCI データは、受信と送信の両方で NRZ (non-return-to-zero) フォーマットです。図 1-3 に示す NRZ データ・フォーマットは、次のもので構成されています。

- 1 スタート・ビット
- 1~8 データ・ビット
- 偶数 / 奇数パリティ・ビット (オプション)
- 1 または 2 ストップ・ビット
- アドレスをデータと区別するための追加ビット (アドレス・ビット・モードのみ)

データの基本単位は文字と呼ばれ、長さは 1~8 ビットです。データの各文字は、スタート・ビット、1 つまたは 2 つのストップ・ビット、オプションのパリティおよびアドレス・ビットでフォーマットされます。フォーマット情報を持つデータの文字をフレームと呼びます (図 1-3)。

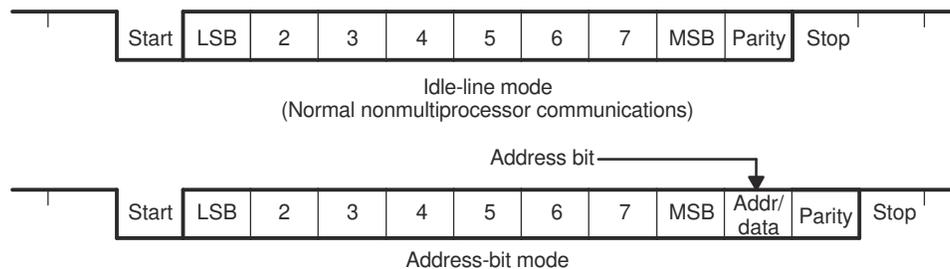


図 1-3. 代表的な SCI データ・フレームのフォーマット

データ・フォーマットをプログラムするには、SCICCR レジスタを使用します。表 1-2 に、データ・フォーマットのプログラムに使用されるビットを示します。

表 1-2. SCICCR を使用したデータ・フォーマットのプログラミング

ビット	ビット名	割り当て	機能
2~0	SCICCHAR	SCICCR.2:0	文字 (データ) の長さ (1~8 ビット) を選択します。
5	PARITYENA (ENABLE)	SCICCR.5	1 に設定するとパリティ機能がイネーブルになり、0 にクリアするとパリティ機能がディセーブルになります。
6	PARITY (EVEN/ODD)	SCICCR.6	パリティがイネーブルの場合に、0 にクリアすると奇数パリティが選択され、1 に設定すると偶数パリティが選択されます。
7	STOPBITS	SCICCR.7	送信されるストップ・ビット数を指定します。0 にクリアするとストップ・ビット数は 1、1 に設定するとストップ・ビット数は 2 になります。

1.7 SCI マルチプロセッサ通信

マルチプロセッサ通信フォーマットにより、1つのプロセッサが同じシリアル・リンク上の他のプロセッサにデータ・ブロックを効率的に送信できます。1つのシリアル・ラインでは、一度に1つの転送しか行えません。つまり、シリアル・ライン上には一度に1つのトークラしか存在できません。

アドレス・バイト

トークラが送信する情報ブロックの最初のバイトには、すべてのリスナが読み取るアドレス・バイトが含まれています。アドレスが一致するリスナに対してのみ、アドレス・バイトの後にあるデータ・バイトによって割り込みが生成されます。アドレスが一致しないリスナに対しては、次のアドレス・バイトまで割り込みは生成されません。

スリープ・ビット

シリアル・リンク上のすべてのプロセッサでは、アドレス・バイトが検出されたときのみプロセッサが割り込まれるように、SCI SLEEP ビット (SCICTL1 のビット 2) が 1 に設定されます。プロセッサがブロック・アドレスを読み取り、それがアプリケーション・ソフトウェアで設定された CPU デバイス・アドレスに一致している場合、SCI が各データ・バイトの受信時に割り込みを生成できるように、プログラムで SLEEP ビットをクリアする必要があります。

SLEEP ビットが 1 の場合にもレシーバは動作しますが、アドレス・バイトが検出され、受信フレームのアドレス・ビットが 1 (アドレス・ビット・モードのとき) である場合を除き、レシーバは RXRDY、RXINT、またはレシーバのエラー・ステータス・ビットを 1 に設定しません。SCI では SLEEP ビットは変更されません。SLEEP ビットはソフトウェアで変更する必要があります。

1.7.1 アドレス・バイトの認識

プロセッサがアドレス・バイトを認識する方法は、使用されているマルチプロセッサ・モードによって異なります。次に例を示します。

- アイドル・ライン・モード ([セクション 1.8](#)) では、アドレス・バイトの前に何も無い空間が確保されます。このモードには追加のアドレス/データ・ビットはなく、10 バイトを超えるデータを含むブロックを処理する場合にアドレス・ビット・モードよりも効率的です。アイドル・ライン・モードは、標準的な非マルチプロセッサ SCI 通信に使用する必要があります。
- アドレス・ビット・モード ([セクション 1.9](#)) では、アドレスをデータと区別するために、各バイトにビット (アドレス・ビット) が追加されます。このモードでは、アイドル・モードとは異なり、データのブロック間で待機する必要がないため、小さなデータ・ブロックを多数処理する場合により効率的です。ただし、送信速度が高い場合、プログラムが十分に高速ではないため、送信ストリームで 10 ビットのアイドルを回避できません。

1.7.2 SCI TX および RX 機能の制御

マルチプロセッサ・モードは、ADDR/IDLE MODE ビット (SCICCR、ビット 3) を使用してソフトウェアで選択できます。どちらのモードでも、TXWAKE フラグ・ビット (SCICTL1、ビット 3)、RXWAKE フラグ・ビット (SCIRXST、ビット 1)、および SLEEP フラグ・ビット (SCICTL1、ビット 2) を使用して、これらのモードの SCI トランスミッタおよびレシーバ機能を制御します。

1.7.3 受信シーケンス

どちらのマルチプロセッサ・モードでも、受信シーケンスは次のようになります。

1. アドレス・ブロックを受信すると、SCI ポートがウェイクアップし、割り込みを要求します (割り込みを要求するため、SCICTL2 のビット 1 RX/BK INT ENA をイネーブルにする必要あり)。SCI ポートが送信先アドレスを含むブロックの最初のフレームを読み取ります。
2. 割り込みによりソフトウェア・ルーチンに入り、受信アドレスがチェックされます。このアドレス・バイトが、メモリに格納されているデバイス・アドレス・バイトと比較されます。
3. このチェックでブロックのアドレス指定先がデバイスの CPU であることが示された場合、CPU は SLEEP ビットをクリアし、ブロックの残りの部分を読み取ります。それ以外の場合は、SLEEP ビットが設定されたままソフトウェア・ルーチンが終了し、次のブロック開始まで割り込みを受信しません。

1.8 アイドル・ライン・マルチプロセッサ・モード

アイドル・ラインのマルチプロセッサ・プロトコル (ADDR/IDLE MODE ビット = 0) では、ブロック間のアイドル時間をブロック内のフレーム間よりも長くすることで、ブロックを分離します。フレームの後に High レベル・ビットが 10 以上のアイドル時間がある場合、新しいブロックが開始したことを示します。1 ビットの時間は、ポー値 (ビット/秒) から直接計算されます。図 1-4 に、アイドル・ライン・マルチプロセッサ通信のフォーマットを示します (ADDR/IDLE MODE ビットは SCICCR のビット 3)。

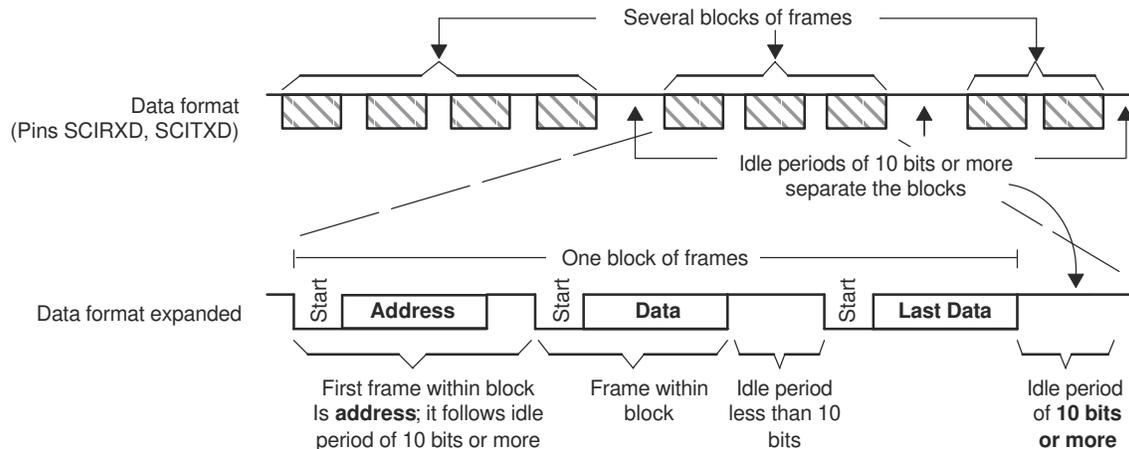


図 1-4. アイドル・ライン・マルチプロセッサ通信のフォーマット

1.8.1 アイドル・ライン・モードの手順

アイドル・ライン・モードの手順は次のとおりです。

1. SCI がブロック開始信号を受信してウェークアップします。
2. プロセッサが次の SCI 割り込みを認識します。
3. 割り込みサービス・ルーチンが受信したアドレス (リモート・トランスミッタから送信) を自身のアドレスと比較します。
4. CPU のがアドレス指定されている場合、サービス・ルーチンは SLEEP ビットをクリアし、残りのデータ・ブロックを受信します。
5. CPU のアドレスが指定されていない場合、SLEEP ビットは設定されたままになります。これにより、CPU は次のブロック開始を検出するまで、SCI ポートから割り込まれることなく、メイン・プログラムの実行を継続できます。

1.8.2 ブロック開始信号

ブロック開始信号を送信するには、2 つの方法があります。

- 方法 1: 前のブロックの最後のデータ・フレーム送信から新しいブロックのアドレス・フレームの送信までの時間を遅延させることで、意図的に 10 ビット以上のアイドル時間が存在するようにします。
- 方法 2: SCITXBUF レジスタに書き込む前に、SCI ポートで TXWAKE ビット (SCICTL1、ビット 3) を 1 に設定します。これにより、11 ビットのアイドル時間が送信されます。この方法では、シリアル通信ラインが必要以上にアイドル状態になることはありません。アイドル時間を送信するため、TXWAKE を設定した後、アドレスを送信する前に、ドントケア・バイトを SCITXBUF に書き込む必要があります。

1.8.3 ウェークアップ時 (WUT) フラグ

TXWAKE ビットには、ウェークアップ時 (WUT) フラグが関連付けられています。WUT は内部フラグであり、TXWAKE でダブル・バッファされます。TXSHF が SCITXBUF からロードされると、WUT が TXWAKE からロードされ、TXWAKE ビットが 0 にクリアされます。この方式を、[図 1-5](#) に示します。

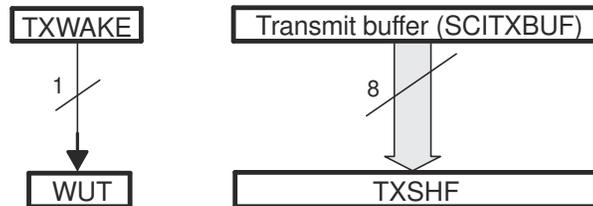


図 1-5. WUT および TXSHF のダブル・バッファ

1.8.3.1 ブロック開始信号の送信

一連のブロックの送信中に、ブロック・スタート信号を 1 フレーム時間だけ送信するには、次の手順に従います。

1. TXWAKE ビットに 1 を書き込みます。
2. SCITXBUF レジスタ (送信データ・バッファ) にデータ・ワード (重要ではない内容 : ドントケア) を書き込み、ブロック開始信号を送信します。ブロック開始信号が送信されている間に書き込まれた最初のデータ・ワードは抑制され、その後は無視されます。TXSHF (送信シフト・レジスタ) が再び解放されると、SCITXBUF の内容が TXSHF にシフトされ、TXWAKE の値が WUT にシフトされて、TXWAKE がクリアされます。

TXWAKE は 1 に設定されているため、スタート・ビット、データ・ビット、パリティ・ビットは、直前のフレームの最後のストップ・ビットに続いて送信される 11 ビットのアイドル期間に置き換えられます。

3. 新しいアドレス値を SCITXBUF に書き込みます。

SCITXBUF レジスタにまずドントケア・データ・ワードを書き込んで、TXWAKE ビットの値を WUT にシフトできるようにする必要があります。ドントケア・データ・ワードが TXSHF レジスタにシフトされた後、TXSHF と WUT は両方ともダブル・バッファであるため、SCITXBUF (および必要に応じて TXWAKE) に再度書き込むことができるようになります。

1.8.4 レシーバの動作

レシーバは、SLEEP ビットに関係なく動作します。ただし、RXRDY もエラー・ステータス・ビットも設定せず、アドレス・フレームが検出されるまで受信割り込みも要求しません。

1.9 アドレス・ビット・マルチプロセッサ・モード

アドレス・ビット・プロトコル (ADDR/IDLE MODE ビット = 1) では、フレームの最後のデータ・ビットの直後にアドレス・ビットと呼ばれるビットがあります。アドレス・ビットは、ブロックの最初のフレームでは 1 に、他のすべてのフレームでは 0 に設定されます。アイドル期間のタイミングは関係しません (図 1-6 を参照)。

1.9.1 アドレスの送信

TXWAKE ビット値は、アドレス・ビットに配置されます。送信中に、SCITXBUF レジスタが TXSHF レジスタにロードされ、TXWAKE が WUT にロードされると、TXWAKE は 0 にリセットされ、WUT が現在のフレームのアドレス・ビットになります。したがって、アドレスを送信する手順は次のようになります。

1. TXWAKE ビットを 1 に設定し、適切なアドレス値を SCITXBUF レジスタに書き込みます。

このアドレス値が TXSHF レジスタに転送されてシフトアウトされると、アドレス・ビットが 1 として送信されます。これにより、シリアル・リンク上の他のプロセッサにアドレスを読み取るよう通知されます。

2. TXSHF および WUT がロードされた後、SCITXBUF および TXWAKE に書き込みます。TXSHF と WUT は両方ともダブル・バッファであるため、すぐに書き込むことができます。
3. TXWAKE ビットを 0 のままにして、ブロック内のアドレス以外のフレームを送信します。

注

一般的に、11 バイト以下のデータ・フレームにはアドレス・ビット・フォーマットが使用されます。このフォーマットでは、送信されるすべてのデータ・バイトに 1 ビット値 (アドレス・フレームの場合は 1、データ・フレームの場合は 0) が追加されます。12 バイト以上のデータ・フレームには、アイドル・ライン・フォーマットが使用されます。

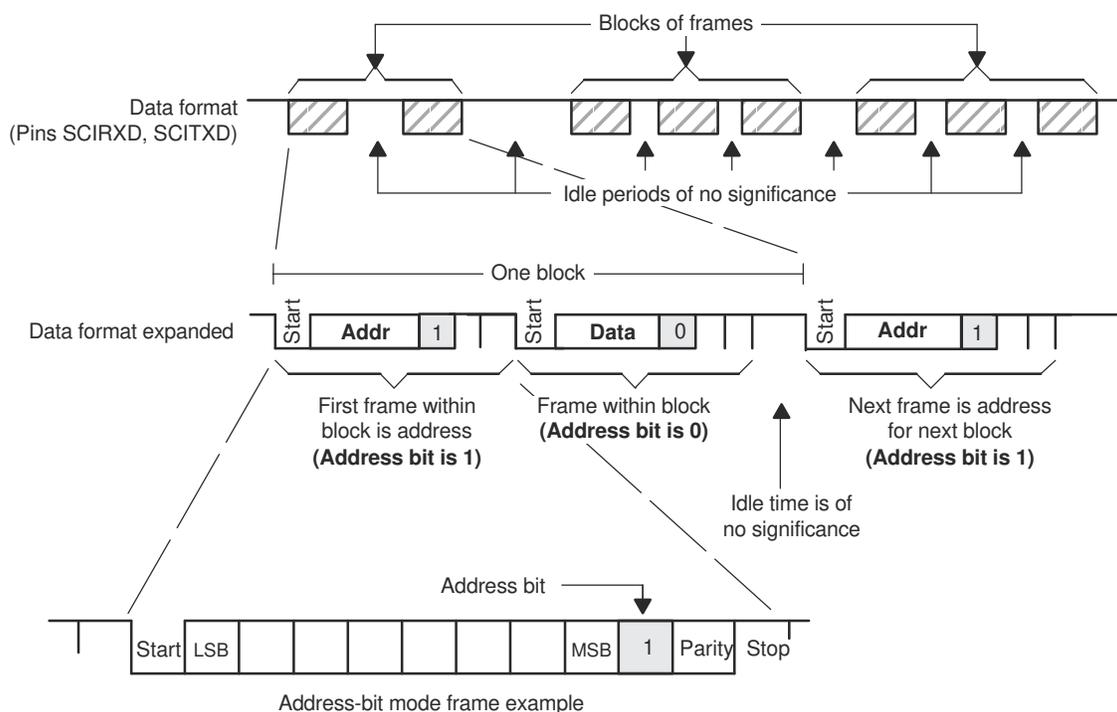


図 1-6. アドレス・ビット・マルチプロセッサ通信フォーマット

1.10 SCI 通信フォーマット

SCI 非同期通信フォーマットは、1 ライン (単方向) または 2 ライン (双方向) のいずれかの通信を使用します。このモードでは、フレームはスタート・ビット、1~8 データ・ビット、オプションの偶数 / 奇数パリティ・ビット、1 つまたは 2 つのストップ・ビットで構成されます (図 1-7 を参照)。データ・ビットごとに 8 つの SCICLK 周期があります。

レシーバは、有効なスタート・ビットを受信すると動作を開始します。図 1-7 に示すように、有効なスタート・ビットは、連続する 4 つの内部 SCICLK 周期が 0 ビットであることにより示されます。いずれかのビットが 0 でない場合、プロセッサは最初からやり直して、別のスタート・ビットを探します。

スタート・ビットに続くビットでは、プロセッサがビットの中間に 3 つのサンプルを作成することによりビット値を判断します。これらのサンプルは、4、5、および 6 番目の SCICLK 周期で発生し、ビット値は多数決 (3 つのうち 2 つ) で決定されます。図 1-7 に、スタート・ビットを含むこの非同期通信フォーマットを示します。多数決が行われた位置も示されています。

レシーバはフレームと同期するため、外部の送受信デバイスが同期シリアル・クロックを使用する必要はありません。クロックはローカルで生成できます。

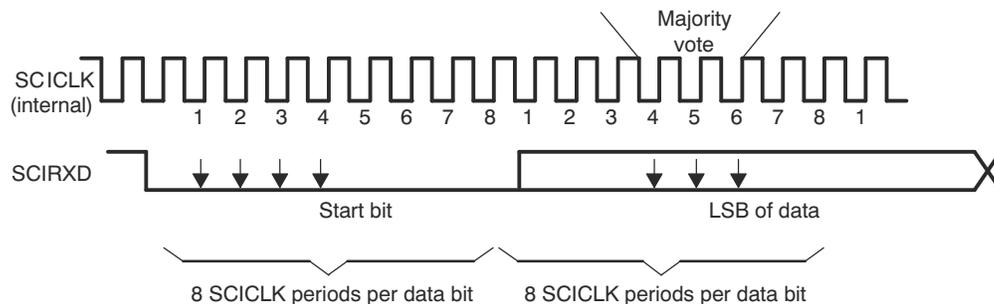


図 1-7. SCI 非同期通信フォーマット

1.10.1 通信モードにおけるレシーバ信号

図 1-8 に、以下の条件を前提としたレシーバ信号のタイミング例を示します。

- アドレス・ビット・ウェークアップ・モード (アイドル・ライン・モードではアドレス・ビットなし)
- 6 ビット/文字

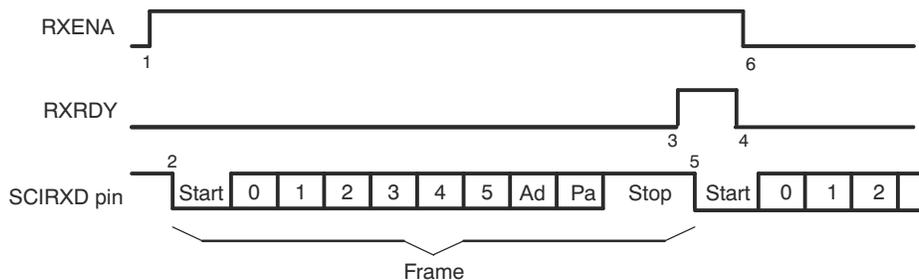


図 1-8. 通信モードにおける SCI RX 信号

注 :

1. フラグ・ビット RXENA (SCICTL1、ビット 0) が High になり、レシーバがイネーブルになります。
2. データが SCIRXD ピンに到着し、スタート・ビットが検出されます。
3. データが RXSHF からレシーバ・バッファ・レジスタ (SCIRXBUF) にシフトされ、割り込みが要求されます。フラグ・ビット RXRDY (SCIRXST、ビット 6) が High になり、新しい文字が受信されたことを示します。
4. プログラムが SCIRXBUF を読み取り、フラグ RXRDY が自動的にクリアされます。
5. 次のデータ・バイトが SCIRXD ピンに到着し、スタート・ビットが検出されてクリアされます。

- ビット RXENA が Low になり、レシーバがディセーブルになります。データは RXSHF で引き続き収集されますが、レシーバ・バッファ・レジスタには転送されません。

1.1.0.2 通信モードにおけるトランスミッタ信号

図 1-9 に、以下の条件を前提としたトランスミッタ信号のタイミング例を示します。

- アドレス・ビット・ウェークアップ・モード (アイドル・ライン・モードではアドレス・ビットなし)
- 3 ビット/文字

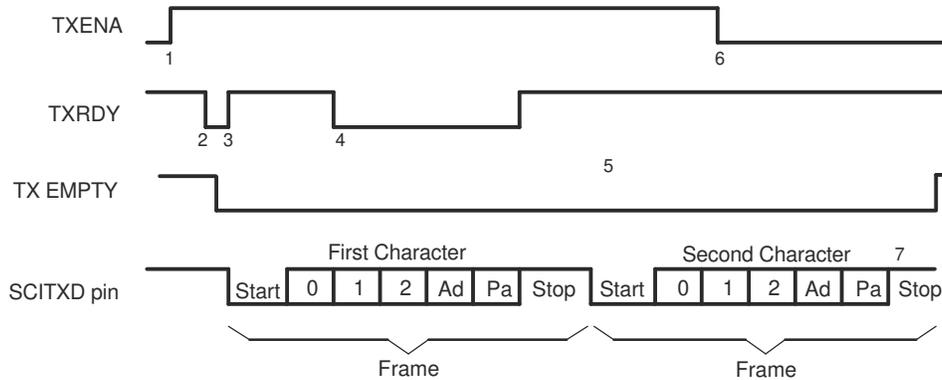


図 1-9. 通信モードにおける SCI TX 信号

注：

1. ビット TXENA (SCICTL1、ビット 1) が High になると、トランスミッタがデータを送信できるようになります。
2. SCITXBUF に書き込まれるため、(1) トランスミッタは空ではなくなり、(2) TXRDY は Low になります。
3. SCI がシフト・レジスタ (TXSHF) にデータを転送します。トランスミッタで 2 番目の文字 (TXRDY が High になる) を転送する準備ができ、割り込みを要求します (割り込みをイネーブルにするにはビット TX INT ENA (SCICTL2、ビット 0) を設定)。
4. TXRDY が High になった後、プログラムが 2 番目の文字を SCITXBUF に書き込みます (項目 3)。2 番目の文字が SCITXBUF に書き込まれると、TXRDY は再び Low になります。
5. 最初の文字の送信が完了します。シフト・レジスタ TXSHF への 2 番目の文字の転送が開始します。
6. ビット TXENA が Low になってトランスミッタがディセーブルになり、SCI が現在の文字の送信を終了します。
7. 2 番目の文字の送信が完了します。トランスミッタは空になり、新しい文字の準備ができます。

1.11 SCI ポート割り込み

SCI レシーバとトランスミッタは、割り込み制御できます。SCICTL2 レジスタには、アクティブな割り込み状態を示すフラグ・ビット (TXRDY) が 1 つあり、SCIRXST レジスタには 2 つの割り込みフラグ・ビット (RXRDY および BRKDT) と、FE、OE、BRKDT、PE の各状態の論理 OR である RX ERROR 割り込みフラグがあります。トランスミッタとレシーバには、個別の割り込みイネーブル・ビットがあります。ディセーブの場合、割り込みはアサートされませんが、状態フラグはアクティブで、送信および受信状態を示します。

SCI には、レシーバ用とトランスミッタ用に独立したペリフェラル割り込みベクトルがあります。ペリフェラル割り込み要求は、高優先度と低優先度のいずれかに指定できます。これは、ペリフェラルから PIE コントローラに出力される優先ビットによって示されます。RX と TX の両方の割り込み要求が同じ優先度の場合、常にレシーバがトランスミッタよりも優先され、レシーバのオーバーランの可能性が低減されます。

ペリフェラル割り込みの動作については、「システム制御および割り込み」の章の「ペリフェラル割り込み」セクションを参照してください。

- RX/BK INT ENA ビット (SCICTL2、ビット 1) が設定されている場合、次のいずれかのイベントが発生すると、レシーバ・ペリフェラル割り込み要求がアサートされます。
 - SCI が完全なフレームを受信し、RXSHF レジスタのデータを SCIRXBUF レジスタに転送する。この動作により RXRDY フラグ (SCIRXST、ビット 6) が設定され、割り込みが開始します。
 - ブレーク検出条件が発生する (欠けているストップ・ビットの後、SCIRXD が 9.625 ビット期間 Low になる)。この動作により BRKDT フラグ・ビット (SCIRXST、ビット 5) が設定され、割り込みが開始します。
- TX INT ENA ビット (SCICTL2.0) が設定されている場合、SCITXBUF レジスタのデータが TXSHF レジスタに転送されるたびにトランスミッタ・ペリフェラルの割り込み要求がアサートされ、CPU が SCITXBUF に書き込むことができることを示します。この動作により TXRDY フラグ・ビット (SCICTL2、ビット 7) が設定され、割り込みが開始します。

注

RXRDY ビットと BRKDT ビットによる割り込み生成は、RX/BK INT ENA ビット (SCICTL2、ビット 1) によって制御されます。RX ERROR ビットによる割り込み生成は、RX ERR INT ENA ビット (SCICTL1、ビット 6) によって制御されます。

1.12 SCI ボーレートの計算

内部で生成されるシリアル・クロックは、低速ペリフェラル・クロック (LSPCLK) とボー選択レジスタによって決定されます。SCI は、ボー選択レジスタの 16 ビット値を使用して、該当する LSPCLK で使用可能な 64K の異なるシリアル・クロック・レートのいずれかを選択します。

SCI 非同期ボーの計算に使用する式については、ボー選択レジスタのビットの説明を参照してください。表 1-3 に、一般的な SCI ビット・レートのボー選択値を示します。最大ボーレートは LSPCLK/16 です。たとえば、LSPCLK が 100MHz の場合、最大ボーレートは 6.25Mbps です。

表 1-3. 共通 SCI ビット・レートの非同期ボー・レジスタ値

理想的なボー	BRR	LSPCLK クロック周波数、100MHz	
		実際のボー	誤差 %
2400	5207 (1457h)	2400	0
4800	2603 (A2Bh)	4800	0
9600	1301 (515h)	9601	0.01
19200	650 (28Ah)	19201	0.01
38400	324 (144h)	38462	0.16

1.13 SCI 拡張機能

C28x SCI には、自動ポー検出機能と、送信 / 受信 FIFO が搭載されています。次のセクションで、FIFO の動作について説明します。

1.13.1 SCI FIFO の説明

以下に、FIFO の機能についての説明と、FIFO を使用して SCI をプログラミングする際に役立つ情報を示します。

- リセット** : リセット時には、SCI は標準 SCI モードで起動し、FIFO 機能はディセーブルになります。FIFO レジスタ SCIFFTX、SCIFFRX、および SCIFFCT は非アクティブのままになります。
- 標準 SCI** : 標準 SCI モードは通常の動作で、TXINT/RXINT 割り込みをモジュールの割り込みソースとして使用します。
- FIFO イネーブル** : FIFO モードは、SCIFFTX レジスタの SCIFFEN ビットをセットするとイネーブルになります。SCIRST は、動作のどの段階でも FIFO モードをリセットできます。
- アクティブ・レジスタ** : SCI レジスタと SCI FIFO レジスタ (SCIFFTX、SCIFFRX、SCIFFCT) がすべてアクティブになります。
- 割り込み** : FIFO モードには、送信 FIFO (TXINT) と受信 FIFO (RXINT) の 2 つの割り込みがあります。RXINT は、SCI FIFO 受信、受信エラー、および受信 FIFO オーバーフロー状態の一般的な割り込みです。標準 SCI の TXINT はディセーブルで、この割り込みは SCI 送信 FIFO 割り込みとして機能します。
- バッファ** : 送信バッファと受信バッファに 2 つの 16 段 FIFO が追加されます。送信 FIFO レジスタは 8 ビット幅で、受信 FIFO レジスタは 10 ビット幅です。標準 SCI の 1 ワード送信バッファ (SCITXBUF) は、送信 FIFO およびシフト・レジスタの前の遷移バッファとして機能します。SCITXBUF は、FIFO (FIFO がイネーブルの場合) または TXSHF (FIFO がディセーブルの場合) のいずれかにロードされます。FIFO がイネーブルのとき、SCITXBUF はシフト・レジスタの最後のビットがシフトアウトされた後にのみ FIFO にロードされるため、SCITXBUF を追加段のバッファとして扱うことはできません。FIFO がイネーブルの場合、オプションの遅延値 (SCIFFCT) の後、TXSHF は (TXBUF ではなく) FIFO から直接ロードされます。SCI で FIFO モードがイネーブルのとき、SCITXBUF に書き込まれた文字は SCI-TXFIFO にキューされ、SCI-RXFIFO で受信された文字は SCIRXBUF を使用してを読み取ることができます。
- 転送の遅延** : FIFO 内のワードが送信シフト・レジスタに転送されるレートはプログラム可能です。SCIFFCT レジスタ・ビット (7~0) FFXDLY7~FFXDLY0 は、ワード転送間の遅延を定義します。遅延は SCI ポー・クロック・サイクル数で定義します。8 ビット・レジスタでは、0 ポー・クロック・サイクルの最小遅延と、256 ポー・クロック・サイクルの最大遅延を定義できます。遅延が 0 の場合、SCI モジュールはデータを連続モードで送信し、FIFO ワードは連続してシフト・アウトされます。遅延を 256 クロック・サイクルにすると、SCI モジュールは最大遅延モードでデータを送信し、FIFO ワードは各ワード間に 256 ポー・クロックの遅延が挿入されてシフト・アウトされます。遅延をプログラム可能なため、低速 SCI/UART との通信を CPU の介入をほとんど必要とせずに行うことができます。
- FIFO ステータス・ビット** : 送信 FIFO と受信 FIFO の両方に、ステータス・ビット TXFFST または RXFFST (ビット 12~8) があり、FIFO で利用可能なワード数をいつでも定義できます。送信 FIFO リセット・ビット TXFIFO と受信リセット・ビット RXFIFO は、これらのビットが 0 にクリアされると、FIFO ポインタを 0 にリセットします。これらのビットが 1 に設定されると、FIFO は最初から動作を再開します。
- プログラマブル割り込みレベル** : 送信 FIFO と受信 FIFO の両方で CPU 割り込みを生成できます。割り込みトリガは、送信 FIFO のステータス・ビット TXFFST (ビット 12~8) が、割り込みトリガ・レベル・ビット TXFFIL (ビット 4~0) と一致した (またはそれ以下の) ときに生成されます。これにより、SCI の送信セクションと受信セクションにプログラム可能な割り込みトリガが供給されます。これらのトリガ・レベル・ビットのデフォルト値は、受信 FIFO では 0x11111、送信 FIFO では 0x00000 です。

図 1-10 および表 1-4 に、FIFO/FFO 以外のモードでの SCI 割り込みの動作と構成を示します。

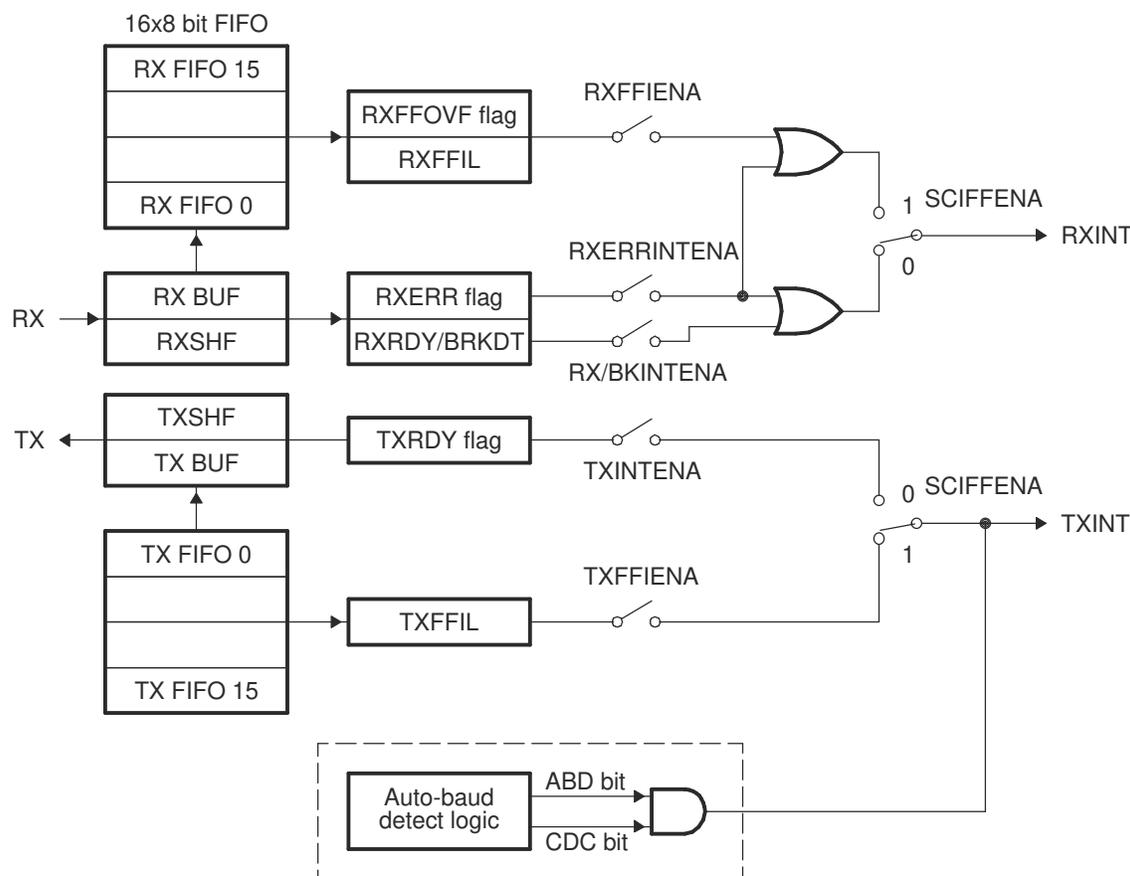


図 1-10. SCI FIFO 割り込みフラグとイネーブル・ロジック

表 1-4. SCI 割り込みフラグ

FIFO オプション ⁽¹⁾	SCI 割り込みソース	割り込みフラグ	割り込みイネーブル	FIFO イネーブル SCIFFENA	割り込みライン
SCI、FIFO デイセーブル	受信エラー	RXERR ⁽²⁾	RXERRINTENA	0	RXINT
	受信ブレーク	BRKDT	RX/BKINTENA	0	RXINT
	データ受信	RXRDY	RX/BKINTENA	0	RXINT
	送信空	TXRDY	TXINTENA	0	TXINT
SCI、FIFO イネーブル	受信エラーおよび受信ブレーク	RXERR	RXERRINTENA	1	RXINT
	FIFO 受信	RXFFIL	RXFFIENA	1	RXINT
	送信空	TXFFIL	TXFFIENA	1	TXINT
自動ボー	自動ボー検出	ABD	ドント・ケア	x	TXINT

(1) FIFO モードの TXSHF は、遅延値の直後に直接ロードされます。TXBUF は使用されません。

(2) RXERR は、BRKDT、FE、OE、PE フラグでセットできます。FIFO モードでは、BRKDT 割り込みは RXERR フラグでのみトリガできます。

1.13.2 SCI 自動ポー

ほとんどの SCI モジュールには、自動ポー検出口ジックが内蔵されたハードウェアがありません。これらの SCI モジュールは、クロック・レートが PLL のリセット値によって変化する組み込みコントローラと統合されています。組み込みコントローラのクロックは、最終的な設計の後で変化的ることがよくあります。拡張機能セットでは、このモジュールはハードウェアの自動ポー検出口ジックをサポートしています。次のセクションでは、自動ポー検出機能のイネーブル・シーケンスについて説明します。

1.13.3 自動ポー検出シーケンス

SCIFFCT のビット ABD および CDC は、自動ポー・ロジックを制御します。SCIRST ビットをイネーブルにして、自動ポー・ロジックを動作させることができます。

CDC が 1 (自動ポー・アライメントがイネーブル) のときに ABD を設定すると、SCI 送信 FIFO 割り込みが発生します (TXINT)。割り込みサービス後に、CDC ビットをソフトウェアでクリアする必要があります。割り込みサービス後に CDC を設定したままにした場合、割り込みを繰り返し発生させることはできません。

1. SCIFFCT の CDC ビット (ビット 13) を設定し、ABDCLR ビット (ビット 14) に 1 を書き込んで ABD ビット (ビット 15) をクリアして、SCI の自動ポー検出モードをイネーブルにします。
2. ポー・レジスタを 1 または 500kbps のポーレート制限以下に初期化します。
3. SCI がホストから適切なポーレートで文字「A」または「a」を受信できるようにします。最初の文字が「A」または「a」の場合、自動ポー検出ハードウェアが受信ポーレートを検出し、ABD ビットを設定します。
4. 自動ポー検出ハードウェアは、ポーレート・レジスタを等価の 16 進ポー値で更新します。このロジックは、CPU への割り込みも生成します。
5. SCIFFCT レジスタの ABD CLR (ビット 14) に 1 を書き込んで割り込みクリア ADB ビットに応答し、CDC ビットに 0 に書き込んでクリアすることにより自動ポー・ロックをディセーブルします。
6. 受信バッファで文字「A」または「a」を読み取り、バッファとバッファのステータスを空にします。
7. CDC が 1 (自動ポー・アライメントがイネーブル) のときに ABD を設定すると、SCI 送信 FIFO 割り込みが発生します (TXINT)。割り込みサービス後に、CDC ビットをソフトウェアでクリアする必要があります。

注

ポーレートが高い場合、受信データ・ビットのスルーレートがトランシーバおよびコネクタの性能の影響を受けることがあります。通常のシリアル通信はうまく動作しますが、このスルーレートにより、高いポーレート (通常 100k ポー超) での自動ポー検出の信頼性が制限され、自動ポーロック機能が正常に機能しなくなる可能性があります。

この問題を回避するため、以下のことを推奨します。

- より低いポーレートを使用して、ホストと C28x SCI ブート・ローダの間でポーのロックを達成します。
- これにより、ホストがロードされた C28x アプリケーションとハンドシェイクし、SCI ポーレート・レジスタを適切な高いポーレートに設定できます。

1.14 ソフトウェア

1.14.1 SCI の例

注：これらの例は、[C2000Ware](#) インストール・ディレクトリの次の場所にあります。
C2000Ware_VERSION#/driverlib/DEVICE_GPN/examples/CORE_IF_MULTICORE/sci

これらのサンプルは、クラウドの [dev.ti.com C2000Ware サンプル](#) から入手できます。

1.14.1.1 UART を使用したボーレートの調整例

ファイル：baud_tune_via_uart.c

この例では、別のデバイスからの UART 入力に基づいて C2000 デバイスの UART/SCI ボーレートを調整するプロセスを示します。UART にはクロック信号がないため、信頼性の高い通信を行うには、ボーレートがおおよそ一致している必要があります。この例では、デバイス間のクロックの不一致が通信で許容される範囲を超えており、ボード間でボーレートの調整が必要な場合を示します。信頼性の高い通信では、EFFECTIVE ボーレートを一致させることのみが必要なため、2つのボードのうちどちらを調整してもかまいません。調整するのが精度の低いクロック・ソースを搭載したボードである必要はありません。2つのデバイスのいずれかがもう一方に合わせて調整されていれば、適切な通信を確立できます。

このデバイスのボーレートを調整するには、(目的のボーレートの) SCI データをこのデバイスに送信する必要があります。入力 SCI のボーレートは、以下で選択した TARGETBAUD の \pm MARGINPERCENT 内である必要があります。これら 2 つの変数は以下に定義されており、アプリケーションの要件に基づいて選択できます。MARGINPERCENT が高いほど、ノイズの多い状況でより多くのデータが「正しい」と見なされ、精度が低下する可能性があります。TARGETBAUD はボーレートと想定された値ですが、クロックの差異により、他のデバイスとの通信の堅牢性を高めるために調整する必要があります。

注：ボーレートが低いほど、レジスタ・オプションの粒度が高くなるため、これらの速度では調整の影響を受けやすくなります。

制御カードの外部接続

- SCIA_RX/eCAP1 は GPIO9 上にあり、受信 SCI 通信に接続します。
- SCIA_TX は GPIO8 上にあり、外部で観測できます。

注目する変数

- *avgBaud* - 調整後に検出および設定されたボーレート

1.14.1.2 SCI FIFO デジタル・ループバック

ファイル：sci_ex1_loopback.c

このプログラムでは、ペリフェラルの内部ループバック・テスト・モードを使用します。ブート・モード・ピン構成以外のハードウェア構成は不要です。PinMux および SCI モジュールは、sysconfig ファイルを使用して構成します。

このテストでは、SCI モジュールのループバック・テスト・モードを使用して、0x00~0xFF までの文字を送信します。このテストでは、文字を送信し、受信バッファの文字が一致するかどうかをチェックします。

注目する変数

- *loopCount* - 送信された文字数
- *errorCount* - 検出されたエラーの数
- *sendChar* - 送信された文字
- *receivedChar* - 受信した文字

1.14.1.3 割り込み付き SCI デジタル・ループバック

ファイル：sci_ex2_loopback_interrupts.c

このテストでは、ペリフェラルの内部ループバック・テスト・モードを使用します。ブート・モード・ピン構成以外のハードウェア構成は不要です。割り込みと SCI FIFO の両方が使用されます。

データ・ストリームが送信され、受信したストリームと比較されます。SCI-A の送信データは次のようになります。

```
00 01
01 02
02 03
```

....

```
FE FF
FF 00
など
```

このパターンは永続的に繰り返されます。

注目する変数

- *sDataA* - 送信中のデータ
- *rDataA* - 受信したデータ
- *rDataPointA* - データ・ストリーム内の位置。これを使用して、受信データをチェックします。

1.14.1.4 SCI エコーバック

ファイル : `sci_ex3_echoback.c`

このテストは、SCI-A ポートを使用してデータを受信し、エコーバックします。

Putty などの端末を使用して SCI からデータを表示し、SCI に情報を送信できます。SCI ポートで受信した文字は、ホストに送り返されます。

アプリケーションの実行 端末を使用して次の設定で COM ポートを開きます。

- 正しい COM ポートを検索
- ビット/秒 : 9600
- データ・ビット : 8
- パリティ : なし
- ストップ・ビット : 1
- ハードウェア制御 : なし

プログラムにより初期メッセージが表示され、端末に送り返す文字を入力するように求められます。

注目する変数

- *loopCounter* - 送信された文字数

外部接続

制御カード J1:A から PC に USB ケーブルを接続します。

1.14.1.5 標準出力のリダイレクト例

ファイル : `sci_ex4_stdout_redirect.c` このテストは、SCI-A ポートを使用してデータを端末に送信します。

Putty などの端末を使用して SCI からデータを表示できます。SCI ポートで受信した文字は、ホストに送り返されます。

アプリケーションの実行 端末を使用して次の設定で COM ポートを開きます。

- 正しい COM ポートを検索
- ビット/秒 : 9600
- データ・ビット : 8
- パリティ : なし
- ストップ・ビット : 1
- ハードウェア制御 : なし

このプログラムは、3 つの文 (SCIA に 1 つ、CCS に 1 つ、SCIA に 1 つ) を出力します。

外部接続

トランシーバとケーブルを使用して SCI-A ポートを PC に接続します。

- `DEVICE_GPIO_PIN_SCIRXDA` は SCI_A-RXD (シリアル DB9 ケーブルのピン 3、PC-TX に接続)
- `DEVICE_GPIO_PIN_SCITXDA` は SCI_A-TXD (シリアル DB9 ケーブルのピン 2、PC-RX に接続)

1.15 SCI レジスタ

このセクションでは、シリアル通信インターフェイス・モジュールのレジスタについて説明します。

1.15.1 SCI ベース・アドレスの表

表 1-5. SCI ベース・アドレスの表

ビット・フィールド名		DriverLib 名	ベース・アドレス	バイプライン保護
インスタンス	構造			
SciaRegs	SCI_REGS	SCIA_BASE	0x0000_7200	あり
ScibRegs	SCI_REGS	SCIB_BASE	0x0000_7210	あり
ScicRegs	SCI_REGS	SCIC_BASE	0x0000_7220	あり

1.15.2 SCI_REGS レジスタ

表 1-7 に、SCI_REGS レジスタのメモリマップを示します。表 1-7 に記載されていないレジスタ・オフセット・アドレスは、すべて予約済みと見なされます。これらのレジスタの内容を変更しないでください。

表 1-6. スキャッターショット・レビュー

バージョン	0.0
入力 IP-XACT ファイル	sci_registers_ipxact.xml
出力 DZ ファイル	SCI_REGS_Registers_sci_registers_ipxact_dz.xml
最後の実行	2023 年 5 月 12 日 金曜日 11:31:40

表 1-7. SCI_REGS レジスタ

オフセット	略称	レジスタ名	書き込み保護	セクション
0h	SCICCR	通信制御レジスタ		表示
1h	SCICTL1	制御レジスタ 1		表示
2h	SCIHBAUD	ボーレート (高) レジスタ		表示
3h	SCILBAUD	ボーレート (低) レジスタ		表示
4h	SCICTL2	制御レジスタ 2		表示
5h	SCIRXST	受信ステータス・レジスタ		表示
6h	SCIRXEMU	受信エミュレーション・バッファ・レジスタ		表示
7h	SCIRXBUF	受信データ・バッファ		表示
9h	SCITXBUF	送信データ・バッファ		表示
Ah	SCIFFTX	FIFO 送信レジスタ		表示
Bh	SCIFFRX	FIFO 受信レジスタ		表示
Ch	SCIFFCT	FIFO 制御レジスタ		表示
Fh	SCIPRI	SCI 優先順位制御		表示

表の小さい枠に収まるように、複雑なビット・アクセス・タイプを記号で表わしています。表 1-8 に、このセクションでアクセス・タイプに使用しているコードを示します。

表 1-8. SCI_REGS のアクセス・タイプ・コード

アクセス・タイプ	コード	説明
読み出しタイプ		
R	R	読み出し
R-0	R-0	読み取り 0 を返す
書き込みタイプ		
W	W	書き込み
W1S	W1S	1 を書き込むことでセット
リセットまたはデフォルト値		
-n		リセット後の値またはデフォルト値
レジスタ・アレイ変数		
i, j, k, l, m, n		これらの変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタが反復レジスタ・グループの一部であるレジスタ・アレイの値を示します。レジスタ・グループは階層構造を形成し、アレイは式で表されません。

**表 1-8. SCI_REGS のアクセス・タイプ・コード
(continued)**

アクセス・タイプ	コード	説明
y		この変数がレジスタ名、オフセット、またはアドレスで使用されている場合、レジスタ・アレイの値を示します。

1.15.2.1 SCICCR レジスタ (オフセット = 0h) [リセット = 0h]

図 1-11 に SCICCR を示し、表 1-9 にその説明を示します。

概略表に戻ります。

SCICCR は、SCI が使用する文字フォーマット、プロトコル、および通信モードを定義します。

図 1-11. SCICCR レジスタ

15		14		13		12		11		10		9		8	
RESERVED															
R-0h															
7		6		5		4		3		2		1		0	
STOPBITS	PARITY	PARITYENA	LOOPBKENA	ADDRIDLE_MODE	SCICCHAR										
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h										

表 1-9. SCICCR レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7	STOPBITS	R/W	0h	SCI ストップ・ビット数。 このビットは、送信されるストップ・ビットの数を指定します。レシーバは、1つのストップ・ビットのみをチェックします。 リセット・タイプ: SYSRSn 0h (R/W) = ストップ・ビット数 1 1h (R/W) = ストップ・ビット数 2
6	PARITY	R/W	0h	SCI パリティの奇数 / 偶数の選択。 PARITY ENABLE ビット (SCICCR、ビット 5) が設定されている場合、PARITY (ビット 6) は奇数パリティまたは偶数パリティを指定します (送信文字と受信文字の両方で、値が 1 のビット数が奇数または偶数) リセット・タイプ: SYSRSn 0h (R/W) = 奇数パリティ 1h (R/W) = 偶数パリティ
5	PARITYENA	R/W	0h	SCI パリティ・イネーブル。 このビットは、パリティ機能をイネーブルまたはディセーブルにします。SCI がアドレス・ビット・マルチプロセッサ・モードの場合 (このレジスタのビット 3 を使用して設定)、パリティ計算にアドレス・ビットが含まれます (パリティがイネーブルの場合)。8 ビット未満の文字の場合、残りの未使用ビットはパリティ計算からマスクして除外する必要があります。 リセット・タイプ: SYSRSn 0h (R/W) = パリティはディセーブル 送信時にパリティ・ビットは生成されず、受信時にパリティ・ビットはないと想定 1h (R/W) = パリティはイネーブル
4	LOOPBKENA	R/W	0h	ループバック・テスト・モード・イネーブル。 ループバック・テスト・モード (Tx ピンを Rx ピンに内部接続) をイネーブルにします。 リセット・タイプ: SYSRSn 0h (R/W) = ループバック・テスト・モードはディセーブル 1h (R/W) = ループバック・テスト・モードはイネーブル

表 1-9. SCICCR レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	ADDRIDLE_MODE	R/W	0h	<p>SCI マルチプロセッサ・モード制御ビット。 このビットは、マルチプロセッサ・プロトコルのいずれかを選択します。マルチプロセッサ通信は、SLEEP 機能 (SCICTL1、ビット 2) と TXWAKE 機能 (SCICTL1、ビット 3) を使用するため、他の通信モードとは異なります。アドレス・ビット・モードではフレームにビットが追加されるため、通常の通信ではアイドル・ライン・モードが使用されます。アイドル・ライン・モードではビットは追加されず、RS-232 タイプの通信と互換性があります。</p> <p>リセット・タイプ : SYSRSn 0h (R/W) = アイドル・ライン・モード・プロトコルを選択 1h (R/W) = アドレス・ビット・モード・プロトコルを選択</p>
2~0	SCICHAR	R/W	0h	<p>文字の長さの制御ビット 2~0。 これらのビットは、SCI 文字の長さを 1~8 ビットに選択します。SCIRXBUF および SCIRXEMU では、8 ビット未満の文字は右揃えされ、SCIRXBUF では先頭に 0 がパディングされます。SCITXBUF は先頭に 0 をパディングする必要はありません。</p> <p>リセット・タイプ : SYSRSn 0h (R/W) = SCICHAR_LENGTH_1 1h (R/W) = SCICHAR_LENGTH_2 2h (R/W) = SCICHAR_LENGTH_3 3h (R/W) = SCICHAR_LENGTH_4 4h (R/W) = SCICHAR_LENGTH_5 5h (R/W) = SCICHAR_LENGTH_6 6h (R/W) = SCICHAR_LENGTH_7 7h (R/W) = SCICHAR_LENGTH_8</p>

1.15.2.2 SCICTL1 レジスタ (オフセット = 1h) [リセット = 0h]

図 1-12 に SCICTL1 を示し、表 1-10 にその説明を示します。

概略表に戻ります。

SCICTL1 は、レシーバ/トランスミッタのイネーブル、TXWAKE および SLEEP 機能、SCI ソフトウェアのリセットを制御します。

図 1-12. SCICTL1 レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RXERRINTENA	SWRESET	RESERVED	TXWAKE	SLEEP	TXENA	RXENA
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 1-10. SCICTL1 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~7	RESERVED	R	0h	予約済み
6	RXERRINTENA	R/W	0h	SCI 受信エラー割り込みイネーブル。 このビットを設定すると、エラーが発生したために RX ERROR ビット (SCIRXST、ビット 7) が設定された場合に割り込みがイネーブルになります。 リセット・タイプ: SYSRSn 0h (R/W) = 受信エラー割り込みはディセーブル 1h (R/W) = 受信エラー割り込みはイネーブル
5	SWRESET	R/W	0h	SCI ソフトウェア・リセット (アクティブ Low)。 このビットに 0 を書き込むと、SCI ステート・マシンと動作フラグ (レジスタ SCICTL2 および SCIRXST) がリセット状態に初期化されます。SW RESET ビットは、どの構成ビットにも影響しません。 影響を受けるすべてのロジックは、SW RESET に 1 が書き込まれるまで、指定されたリセット状態に保持されます (このセクションでは、リセット後のビット値を各レジスタ図の下に表示)。そのため、システム・リセット後、このビットに 1 を書き込んで SCI を再度イネーブルにします。レシーバ・ブレイク検出後にこのビットをクリアします (BRKDT フラグ、ビット SCIRXST、ビット 5)。 SW RESET は SCI の動作フラグに影響しますが、構成ビットには影響せず、リセット値を復元することはありません。SW RESET がアサートされると、ビットがデアサートされるまでフラグは固定されます。 影響を受けるフラグは次のとおりです。 SW SCI フラグ・レジスタ・ビット後の値 RESET 1 TXRDY SCICTL2、ビット 7 1 TX EMPTY SCICTL2、ビット 6 0 RXWAKE SCIRXST、ビット 1 0 PE SCIRXST、ビット 2 0 OE SCIRXST、ビット 3 0 FE SCIRXST、ビット 4 0 BRKDT SCIRXST、ビット 5 0 RXRDY SCIRXST、ビット 6 0 RX ERROR SCIRXST、ビット 7 リセット・タイプ: SYSRSn 0h (R/W) = このビットに 0 を書き込むと、SCI ステート・マシンと動作フラグ (レジスタ SCICTL2 および SCIRXST) がリセット状態に初期化されます。 1h (R/W) = システム・リセット後、このビットに 1 を書き込んで SCI を再度イネーブルにします。
4	RESERVED	R	0h	予約済み

表 1-10. SCICTL1 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
3	TXWAKE	R/W	0h	<p>SCI トランスミッタのウェークアップ方法の選択。 TXWAKE ビットは、ADDR/IDLE MODE ビット (SCICCR、ビット 3) で指定されている送信モード (アイドル・ラインまたはアドレス・ビット) に応じて、データ送信機能の選択を制御します。 リセット・タイプ: SYSRSn 0h (R/W) = 送信機能は選択されていません。アイドル・ライン・モード: TXWAKE に 1 を書き込んでからレジスタ SCITXBUF にデータを書き込み、11 データ・ビットのアイドル期間を生成します。アドレス・ビット・モード: TXWAKE に 1 を書き込んでから SCITXBUF にデータを書き込み、そのフレームのアドレス・ビットを 1 に設定します。 1h (R/W) = 選択される送信機能は、モード (アイドル・ラインまたはアドレス・ビット) によって異なります。TXWAKE は SW RESET ビット (SCICTL1、ビット 5) ではクリアされません。 システム・リセットまたは TXWAKE を WUT フラグに転送することによってクリアされます。</p>
2	SLEEP	R/W	0h	<p>SCI スリープ。 TXWAKE ビットは、ADDR/IDLE MODE ビット (SCICCR、ビット 3) で指定されている送信モード (アイドル・ラインまたはアドレス・ビット) に応じて、データ送信機能の選択を制御します。マルチプロセッサ構成では、このビットによってレシーバのスリープ機能が制御されます。このビットをクリアすると、SCI はスリープ・モードから復帰します。 レシーバは SLEEP ビットが設定されている場合でも動作します。ただし、レシーバ・バッファ・レディ・ビット (SCIRXST、ビット 6、RXRDY) およびエラー・ステータス・ビット (SCIRXST、ビット 5~2: BRKDT、FE、OE、PE) はアドレス・バイトが検出されない場合は更新されません。アドレス・バイトが検出されても、SLEEP はクリアされません。 リセット・タイプ: SYSRSn 0h (R/W) = スリープ・モードはディセーブル 1h (R/W) = スリープ・モードはイネーブル</p>
1	TXENA	R/W	0h	<p>SCI トランスミッタ・イネーブル。 データは、TXENA が設定されている場合にのみ、SCITXD ピンを経由して送信されます。リセットすると、以前に SCITXBUF に書き込まれたすべてのデータが送信された後に転送が停止します。TXENA がディセーブルのときに SCITXBUF に書き込まれたデータは、TXENA が後でイネーブルになっても送信されません。 リセット・タイプ: SYSRSn 0h (R/W) = トランスミッタはディセーブル 1h (R/W) = トランスミッタはイネーブル</p>
0	RXENA	R/W	0h	<p>SCI レシーバ・イネーブル。 データが SCIRXD ピンで受信され、レシーバ・シフト・レジスタに送信されてから、レシーバ・バッファに送信されます。このビットは、レシーバ (バッファへの転送) をイネーブルまたはディセーブルにします。 RXENA をクリアすると、受信した文字が 2 つのレシーバ・バッファに転送されなくなり、レシーバ割り込みの生成も停止します。ただし、レシーバ・シフト・レジスタは引き続き文字を収集できます。そのため、文字の受信中に RXENA を設定すると、レシーバ・バッファ・レジスタ SCIRXEMU および SCIRXBUF に完全な文字が転送されます。 リセット・タイプ: SYSRSn 0h (R/W) = 受信した文字を SCIRXEMU および SCIRXBUF レシーバ・バッファに転送しない 1h (R/W) = 受信した文字を SCIRXEMU および SCIRXBUF に送信</p>

1.15.2.3 SCIHBAUD レジスタ (オフセット = 2h) [リセット = 0h]

図 1-13 に SCIHBAUD を示し、表 1-11 にその説明を示します。

[概略表](#)に戻ります。

SCIHBAUD および SCILBAUD の値は、SCI のボーレートを指定します。

図 1-13. SCIHBAUD レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
BAUD							
R/W-0h							

表 1-11. SCIHBAUD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7~0	BAUD	R/W	0h	SCI 16 ビット・ボー選択レジスタ SCIHBAUD (MSbyte)。 内部で生成されるシリアル・クロックは、低速ペリフェラル・クロック (LSPCLK) 信号と 2 つのボー選択レジスタによって決定されます。SCI は、これらのレジスタの 16 ビット値を使用して、通信モードに使用する 64K シリアル・クロック・レートのいずれかを選択します。 $BRR = (SCIHBAUD \ll 8) + (SCILBAUD)$ SCI ボーレートは、次の式で計算されます。 $SCI \text{ 非同期ボー} = LSPCLK / ((BRR + 1) * 8)$ または $BRR = LSPCLK / (SCI \text{ 非同期ボー} * 8) - 1$ 上記の式は、 $0 < BRR < 65536$ の場合にのみ適用されます。BRR = 0 の場合、 $SCI \text{ 非同期ボー} = LSPCLK / 16$ 説明：BRR = ボー選択レジスタの 16 ビット値 (10 進) リセット・タイプ：SYSRSn

1.15.2.4 SCILBAUD レジスタ (オフセット = 3h) [リセット = 0h]

図 1-14 に SCILBAUD を示し、表 1-12 にその説明を示します。

概略表に戻ります。

SCIHBAUD および SCILBAUD の値は、SCI のボーレートを指定します。

図 1-14. SCILBAUD レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
BAUD							
R/W-0h							

表 1-12. SCILBAUD レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7~0	BAUD	R/W	0h	「SCIHBAUD の詳細説明」を参照 リセット・タイプ : SYSRSn

1.15.2.5 SCICTL2 レジスタ (オフセット = 4h) [リセット = C0h]

図 1-15 に SCICTL2 を示し、表 1-13 にその説明を示します。

概略表に戻ります。

SCICTL2 は、受信準備完了、ブレーク検出、送信準備完了の割り込み、トランスミッタ・レディ・フラグ、空フラグを有効にします。

図 1-15. SCICTL2 レジスタ

15	14	13	12	11	10	9	8	
RESERVED								
R-0h								
7	6	5	4	3	2	1	0	
TXRDY	TXEMPTY	RESERVED				RXBKINTENA	TXINTENA	
R-1h	R-1h	R-0h				R/W-0h	R/W-0h	

表 1-13. SCICTL2 レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7	TXRDY	R	1h	トランスミッタ・バッファ・レジスタ・レディ・フラグ。 このビットを設定すると、送信データ・バッファ・レジスタ (SCITXBUF) が別の文字を受信する準備ができたことを示します。SCITXBUF にデータを書き込むと、このビットは自動的にクリアされます。このフラグを設定すると、割り込みイネーブル・ビット TX INT ENA (SCICTL2.0) も設定されている場合に、トランスミッタ割り込み要求がアサートされます。TXRDY は、SW RESET ビット (SCICTL1.5) をイネーブルにするか、システム・リセットによって 1 に設定されます。 リセット・タイプ: SYSRSn 0h (R/W) = SCITXBUF はフル 1h (R/W) = SCITXBUF は次の文字を受信可能
6	TXEMPTY	R	1h	トランスミッタの空フラグ。 このフラグの値は、トランスミッタのバッファ・レジスタ (SCITXBUF) とシフト・レジスタ (TXSHF) の内容を示します。アクティブ SW RESET (SCICTL1.5) またはシステム・リセットにより、このビットが設定されます。このビットは割り込み要求を発生させません。 リセット・タイプ: SYSRSn 0h (R/W) = トランスミッタ・バッファまたはシフト・レジスタ、あるいはその両方にデータがロードされている 1h (R/W) = トランスミッタ・バッファとシフト・レジスタの両方が空
5~2	RESERVED	R	0h	予約済み
1	RXBKINTENA	R/W	0h	レシーバ・バッファ/ブレーク割り込みイネーブル。 このビットは、RXRDY フラグまたは BRKDT フラグ (ビット SCIRXST.6 および .5) が設定されているときに発生する割り込み要求を制御します。ただし、RX/BK INT ENA ではこれらのフラグが設定されないようにすることはできません。 リセット・タイプ: SYSRSn 0h (R/W) = RXRDY/BRKDT 割り込みはディセーブル 1h (R/W) = RXRDY/BRKDT 割り込みはイネーブル

表 1-13. SCICTL2 レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
0	TXINTENA	R/W	0h	<p>SCITXBUF レジスタ割り込みイネーブル。</p> <p>このビットは、TXRDY フラグ・ビット (SCICTL2.7) が設定されているときに発生する割り込み要求を制御します。ただし、TXRDY フラグ (SCITXBUF が別の文字を受信する準備ができたことを示す) が設定されないようにすることはできません。</p> <p>0 = TXRDY 割り込みはディセーブル 1 = TXRDY 割り込みはネーブル。</p> <p>FIFO 以外のモードでは、最初の送信割り込みを発生させるため、ダミー (または有効な) データを SCITXBUF に書き込む必要があります。これは、初めて送信割り込みをイネーブルにしたときと、送信割り込みを再度イネーブル (ディセーブルにしてからイネーブル) にしたときです。データを SCITXBUF に書き込んだ後に TXINTENA をイネーブルにした場合、割り込みは生成されません。</p> <p>リセット・タイプ: SYSRSn 0h (R/W) = TXRDY 割り込みはディセーブル 1h (R/W) = TXRDY 割り込みはイネーブル</p>

1.15.2.6 SCIRXST レジスタ (オフセット = 5h) [リセット = 0h]

図 1-16 に SCIRXST を示し、表 1-14 にその説明を示します。

概略表に戻ります。

SCIRXST には、レシーバ・ステータス・フラグである 7 ビットが含まれています (そのうち 2 ビットは割り込み要求を生成可能)。レシーバ・バッファ (SCIRXEMU および SCIRXBUF) に完全な文字が転送されるたびに、ステータス・フラグが更新されます。

図 1-16. SCIRXST レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RXERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

表 1-14. SCIRXST レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7	RXERROR	R	0h	SCI レシーバ・エラー・フラグ。 RX ERROR フラグは、レシーバ・ステータス・レジスタのいずれかのエラー・フラグが設定されていることを示します。RX ERROR は、ブレイク検出、フレーミング・エラー、オーバーラン、パリティ・エラー・イネーブル・フラグ (ビット 5~2: BRKDT、FE、OE、PE) の論理 OR です。 RX ERR INT ENA ビット (SCICTL1.6) が設定されている場合、このビットが 1 になると割り込みが発生します。このビットは、割り込みサービス・ルーチンでの高速エラー状態チェックに使用できません。このエラー・フラグを直接クリアすることはできません。アクティブな SW RESET またはシステム・リセットによってクリアされます。 リセット・タイプ: SYSRSn 0h (R/W) = 設定されたエラー・フラグなし 1h (R/W) = 1 つ以上のエラー・フラグが設定されている
6	RXRDY	R	0h	SCI レシーバ・レディ・フラグ。 SCIRXBUF レジスタから新しい文字を読み取る準備ができるとレシーバがこのビットを設定し、RX/BK INT ENA ビット (SCICTL2.1) が 1 のときにレシーバ割り込みが生成されます。RXRDY は、SCIRXBUF レジスタの読み取り、アクティブな SW RESET、またはシステム・リセットによってクリアされます。 リセット・タイプ: SYSRSn 0h (R/W) = SCIRXBUF に新しい文字がない 1h (R/W) = SCIRXBUF から文字を読み取る準備完了
5	BRKDT	R	0h	SCI ブレイク検出フラグ。 このビットは、ブレイク条件が発生すると設定されます。ブレイク条件は、欠けている最初のストップ・ビット後に、SCI レシーバ・データ・ライン (SCIRXD) が 9.625 ビット以上連続して Low であった場合に発生します。ブレイクが発生すると、RX/BK INT ENA ビットが 1 である場合にレシーバ割り込みが生成されますが、レシーバ・バッファはロードされません。レシーバの SLEEP ビットが 1 に設定されている場合でも、BRKDT 割り込みが発生する可能性があります。BRKDT は、アクティブな SW RESET またはシステム・リセットによってクリアされます。ブレイクの検出後に文字を受信してもクリアされません。さらに多くの文字を受信するには、SW RESET ビットをトグルするが、システム・リセットによって SCI をリセットする必要があります。 リセット・タイプ: SYSRSn 0h (R/W) = ブレイク条件なし 1h (R/W) = ブレイク条件発生

表 1-14. SCIRXST レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
4	FE	R	0h	<p>SCI フレーミング・エラー・フラグ。 このビットは、ストップ・ビットが見つからない場合に設定されます。最初のストップ・ビットのみがチェックされます。ストップ・ビットが欠けているということは、スタート・ビットとの同期が失われ、文字が正しくフレーミングされていないことを示します。FE ビットは、SW RESET ビットをクリアするか、システム・リセットによってリセットされます。注：RX がスリープ・モードの場合以外は、FE は BRKDT より前に設定されます。スリープ・モードでは、RX WAKEUP がなく、RXD ラインが 10 ビット以上 Low の場合、BRKDT は設定されますが、FE は設定されません。</p> <p>リセット・タイプ：SYSRSn 0h (R/W) = フレーミング・エラーの検出なし 1h (R/W) = フレーミング・エラー検出</p>
3	OE	R	0h	<p>SCI オーバーラン・エラー・フラグ。 このビットは、前の文字が CPU または DMAC で完全に読み取られる前に、文字がレジスタ SCIRXEMU および SCIRXBUF に転送されると設定されます。前の文字は上書きされ、失われます。OE フラグ・ビットは、アクティブな SW RESET またはシステム・リセットによってリセットされます。</p> <p>リセット・タイプ：SYSRSn 0h (R/W) = オーバーラン・エラーの検出なし 1h (R/W) = オーバーラン・エラー検出</p>
2	PE	R	0h	<p>SCI パリティ・エラー・フラグ。 このフラグ・ビットは、文字を受信したときに、1 の数とそのパリティ・ビットが一致していない場合に設定されます。アドレス・ビットは計算に含まれていません。パリティの生成と検出がイネーブルでない場合、PE フラグはディセーブルされ、0 として読み取られます。PE ビットは、アクティブな SW RESET またはシステム・リセットによってリセットされます。</p> <p>リセット・タイプ：SYSRSn 0h (R/W) = パリティ・エラーの検出なし、またはパリティはディセーブル 1h (R/W) = パリティ・エラー検出</p>
1	RXWAKE	R	0h	<p>レシーバ・ウェークアップ検出フラグ リセット・タイプ：SYSRSn 0h (R/W) = レシーバ・ウェークアップ条件の検出なし 1h (R/W) = レシーバ・ウェークアップ条件検出アドレス・ビット・マルチプロセッサ・モード (SCICCR.3 = 1) では、RXWAKE は SCIRXBUF に含まれる文字のアドレス・ビットの値を示します。アイドル・ライン・マルチプロセッサ・モードでは、SCIRXD データ・ラインがアイドルとして検出されると、RXWAKE が設定されます。RXWAKE は読み取り専用フラグで、次のいずれかによってクリアされます。</p> <ul style="list-style-type: none"> - アドレス・バイトの後に最初のバイトを SCIRXBUF に転送 (FIFO 以外のモードのみ) - SCIRXBUF の読み取り - アクティブな SW RESET - システム・リセット
0	RESERVED	R	0h	予約済み

1.15.2.7 SCIRXEMU レジスタ (オフセット = 6h) [リセット = 0h]

図 1-17 に SCIRXEMU を示し、表 1-15 にその説明を示します。

概略表に戻ります。

通常の SCI データ受信動作は、SCIRXBUF レジスタから受信したデータを読み取ります。SCIRXEMU レジスタは、主にエミュレータ (EMU) によって使用されます。これは、RXRDY フラグをクリアせずに、画面更新のために受信したデータを継続的に読み取ることができるためです。SCIRXEMU は、システム・リセットによりクリアされます。これは、エミュレータの監視ウィンドウで SCIRXBUF レジスタの内容を表示するために使用するレジスタです。SCIRXEMU は物理的に実装されているわけではなく、RXRDY フラグをクリアせずに SCIRXBUF レジスタにアクセスするための異なるアドレスの場所にすぎません。

図 1-17. SCIRXEMU レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
ERXDT							
R-0h							

表 1-15. SCIRXEMU レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7~0	ERXDT	R	0h	受信エミュレーション・バッファ・データ リセット・タイプ: SYSRSn

1.15.2.8 SCIRXBUF レジスタ (オフセット = 7h) [リセット = 0h]

図 1-18 に SCIRXBUF を示し、表 1-16 にその説明を示します。

概略表に戻ります。

受信した現在のデータが RXSHF からレシーバ・バッファにシフトされると、フラグ・ビット RXRDY が設定され、データを読み取る準備ができます。RXBKINTENA ビット (SCICTL2.1) が設定されている場合、このシフトによって割り込みも発生します。SCIRXBUF を読み出すと、RXRDY フラグがリセットされます。SCIRXBUF は、システム・リセットによりクリアされます。

図 1-18. SCIRXBUF レジスタ

15	14	13	12	11	10	9	8
SCIFFFE	SCIFFPE	RESERVED					
R-0h	R-0h	R-0h					
7	6	5	4	3	2	1	0
SAR							
R-0h							

表 1-16. SCIRXBUF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SCIFFFE	R	0h	SCIFFFE。SCI FIFO フレーミング・エラー・フラグ・ビット (FIFO がイネーブルの場合のみ) リセット・タイプ : SYSRSn 0h (R/W) = 文字の受信中にフレーム・エラーなし (ビット 7~0)。このビットは FIFO の最上位の文字に関連しています。 1h (R/W) = 文字の受信中にフレーム・エラー発生 (ビット 7~0)。このビットは FIFO の最上位の文字に関連しています。
14	SCIFFPE	R	0h	SCIFFPE。SCI FIFO パリティ・エラー・フラグ・ビット (FIFO がイネーブルの場合のみ) リセット・タイプ : SYSRSn 0h (R/W) = 文字の受信中にパリティ・エラーなし (ビット 7~0)。このビットは FIFO の最上位の文字に関連しています。 1h (R/W) = 文字の受信中にパリティ・エラー発生 (ビット 7~0)。このビットは FIFO の最上位の文字に関連しています。
13~8	RESERVED	R	0h	予約済み
7~0	SAR	R	0h	受信文字ビット リセット・タイプ : SYSRSn

1.15.2.9 SCITXBUF レジスタ (オフセット = 9h) [リセット = 0h]

図 1-19 に SCITXBUF を示し、表 1-17 にその説明を示します。

[概略表](#)に戻ります。

送信するデータ・ビットは、SCITXBUF に書き込まれます。8 ビット未満の文字では左端のビットが無視されるため、これらのビットは右揃えにする必要があります。このレジスタから TXSHF トランスミッタ・シフト・レジスタにデータが転送されると、TXRDY フラグ (SCICTL2.7) が設定され、SCITXBUF が別のデータ・セットを受信する準備ができたことが示されます。ビット TXINTENA (SCICTL2.0) が設定されている場合、このデータ転送により割り込みも発生します。

図 1-19. SCITXBUF レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
TXDT							
R/W-0h							

表 1-17. SCITXBUF レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7~0	TXDT	R/W	0h	送信データ・バッファ リセット・タイプ: SYSRSn

1.15.2.10 SCIFFTX レジスタ (オフセット= Ah) [リセット= A000h]

図 1-20 に SCIFFTX を示し、表 1-18 にその説明を示します。

概略表に戻ります。

SCIFFTX は、SCI 送信および受信チャネルの送信 FIFO 割り込み、FIFO 拡張、およびリセットを制御します。

図 1-20. SCIFFTX レジスタ

15		14		13		12		11		10		9		8	
SCIRST		SCIFFENA		TXFIFORESET						TXFFST					
R/W-1h		R/W-0h		R/W-1h								R-0h			
7		6		5		4		3		2		1		0	
TXFFINT		TXFFINTCLR		TXFFIENA						TXFFIL					
R-0h		R-0/W1S-0h		R/W-0h								R/W-0h			

表 1-18. SCIFFTX レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	SCIRST	R/W	1h	SCI リセット 0 = 0 を書き込んで SCI 送信および受信チャネルをリセットします。SCI FIFO レジスタ構成ビットはそのままになります。 1 = SCI FIFO で送信または受信を再開します。自動ポーリングを機能させるためには、SCIRST を 1 にする必要があります。 リセット・タイプ: SYSRSn
14	SCIFFENA	R/W	0h	SCI FIFO イネーブル ・リセット・タイプ: SYSRSn 0h (R/W) = SCI FIFO 拡張はディセーブル 1h (R/W) = SCI FIFO 拡張はイネーブル
13	TXFIFORESET	R/W	1h	送信 FIFO リセット リセット・タイプ: SYSRSn 0h (R/W) = FIFO ポインタをゼロにリセットし、リセットに保持 1h (R/W) = 送信 FIFO 動作を再度イネーブル
12~8	TXFFST	R	0h	FIFO ステータス リセット・タイプ: SYSRSn 0h (R/W) = 送信 FIFO が空 1h (R/W) = 送信 FIFO に 1 ワード 2h (R/W) = 送信 FIFO に 2 ワード 3h (R/W) = 送信 FIFO に 3 ワード 4h (R/W) = 送信 FIFO に 4 ワード 5h (R/W) = 送信 FIFO に 5 ワード 6h (R/W) = 送信 FIFO に 6 ワード 7h (R/W) = 送信 FIFO に 7 ワード 8h (R/W) = 送信 FIFO に 8 ワード 9h (R/W) = 送信 FIFO に 9 ワード Ah (R/W) = 送信 FIFO に 10 ワード Bh (R/W) = 送信 FIFO に 11 ワード Ch (R/W) = 送信 FIFO に 12 ワード Dh (R/W) = 送信 FIFO に 13 ワード Eh (R/W) = 送信 FIFO に 14 ワード Fh (R/W) = 送信 FIFO に 15 ワード 10h (R/W) = 送信 FIFO に 16 ワード
7	TXFFINT	R	0h	送信 FIFO 割り込み リセット・タイプ: SYSRSn 0h (R/W) = TXFIFO 割り込みなし、読み取り専用ビット 1h (R/W) = TXFIFO 割り込み発生、読み取り専用ビット
6	TXFFINTCLR	R-0/W1S	0h	送信 FIFO クリア ・リセット・タイプ: SYSRSn 0h (R/W) = 0 を書き込んでも TXFFINT フラグ・ビットに影響なし、ビットは 0 をリードバック 1h (R/W) = 1 を書き込んでビット 7 の TXFFINT フラグをクリア

表 1-18. SCIFFTX レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
5	TXFFIENA	R/W	0h	送信 FIFO の割り込みイネーブル リセット・タイプ : SYSRSn 0h (R/W) = TX FIFO 割り込みはディセーブル 1h (R/W) = TX FIFO 割り込みはイネーブル。この割り込みは、送信 FIFO ステータス (TXFFST) ビットが割り込みトリガ・レベル・ビット TXFFIL (ビット 4~0) と一致した (またはそれ以下の) 場合にトリガされます。
4~0	TXFFIL	R/W	0h	TXFFIL4~0 送信 FIFO 割り込みレベル・ビット。 送信 FIFO は、FIFO ステータス・ビット (TXFFST4~0) が FIFO レベル・ビット (TXFFIL4~0) 以下である場合に割り込みを生成します。割り込みを生成するためにこれらのビットに割り当てることができる最大値は、TX FIFO の深度を超えることはできません。これらのビットのリセット後のデフォルト値は 00000b です。ISR をサービスするための CPU オーバーヘッドと SCI バス帯域幅の最適な使用状況を比較して、アプリケーションのニーズに最適な TXFFIL を設定する必要があります。 リセット・タイプ : SYSRSn

1.15.2.11 SCIFFRX レジスタ (オフセット = Bh) [リセット = 201Fh]

図 1-21 に SCIFFRX を示し、表 1-19 にその説明を示します。

概略表に戻ります。

SCIFFRX は、受信 FIFO 割り込み、受信 FIFO リセット、および受信 FIFO オーバーフローのステータスを制御します。

図 1-21. SCIFFRX レジスタ

15		14		13		12		11		10		9		8	
RXFFOVF		RXFFOVRCLR		RXFIFORESET						RXFFST					
R-0h		R-0/W1S-0h		R/W-1h								R-0h			
7		6		5		4		3		2		1		0	
RXFFINT		RXFFINTCLR		RXFFIENA						RXFFIL					
R-0h		W-0h		R/W-0h								R/W-1Fh			

表 1-19. SCIFFRX レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	RXFFOVF	R	0h	受信 FIFO オーバーフロー。 これはフラグとして機能しますが、割り込みを生成することはできません。この条件は、受信割り込みがアクティブなときに発生します。受信割り込みは、このフラグ条件を処理する必要があります。 リセット・タイプ : SYSRSn 0h (R/W) = 受信 FIFO のオーバーフローなし、読み取り専用ビット 1h (R/W) = 受信 FIFO のオーバーフロー発生、読み取り専用ビット FIFO に 16 ワード以上を受信すると、最初に受信したワードは失われます。
14	RXFFOVRCLR	R-0/W1S	0h	RXFFOVF クリア リセット・タイプ : SYSRSn 0h (R/W) = 0 を書き込んでも RXFFOVF フラグ・ビットに影響はなし、ビットは 0 をリードバック 1h (R/W) = 1 を書き込んでビット 15 の RXFFOVF フラグをクリア
13	RXFIFORESET	R/W	1h	受信 FIFO リセット リセット・タイプ : SYSRSn 0h (R/W) = 0 を書き込んで FIFO ポインタを 0 にリセットし、リセット状態を保持。 1h (R/W) = 受信 FIFO 動作を再イネーブル
12~8	RXFFST	R	0h	FIFO ステータス リセット・タイプ : SYSRSn 0h (R/W) = 受信 FIFO が空 1h (R/W) = 受信 FIFO に 1 ワード 2h (R/W) = 受信 FIFO に 2 ワード 3h (R/W) = 受信 FIFO に 3 ワード 4h (R/W) = 受信 FIFO に 4 ワード 5h (R/W) = 受信 FIFO に 5 ワード 6h (R/W) = 受信 FIFO に 6 ワード 7h (R/W) = 受信 FIFO に 7 ワード 8h (R/W) = 受信 FIFO に 8 ワード 9h (R/W) = 受信 FIFO に 9 ワード Ah (R/W) = 受信 FIFO に 10 ワード Bh (R/W) = 受信 FIFO に 11 ワード Ch (R/W) = 受信 FIFO に 12 ワード Dh (R/W) = 受信 FIFO に 13 ワード Eh (R/W) = 受信 FIFO に 14 ワード Fh (R/W) = 受信 FIFO に 15 ワード 10h (R/W) = 受信 FIFO に 16 ワード
7	RXFFINT	R	0h	受信 FIFO 割り込み リセット・タイプ : SYSRSn 0h (R/W) = RXFIFO 割り込みなし、読み取り専用ビット 1h (R/W) = RXFIFO 割り込み発生、読み取り専用ビット

表 1-19. SCIFFRX レジスタのフィールドの説明 (continued)

ビット	フィールド	タイプ	リセット	説明
6	RXFFINTCLR	W	0h	受信 FIFO 割り込みクリア リセット・タイプ：SYSRSn 0h (R/W) = 0 を書き込んでも RXFIFINT フラグ・ビットに影響なし。ビットは 0 をリードバック。 1h (R/W) = 1 を書き込んでビット 7 の RXFFINT フラグをクリア
5	RXFFIENA	R/W	0h	受信 FIFO 割り込みイネーブル リセット・タイプ：SYSRSn 0h (R/W) = RX FIFO 割り込みはディセーブル 1h (R/W) = RX FIFO 割り込みはイネーブル。この割り込みは、受信 FIFO ステータス (RXFFST) ビットが割り込みトリガ・レベル・ビット RXFFIL (ビット 4~0) と一致した (またはそれ以下の) 場合にトリガされます。
4~0	RXFFIL	R/W	1Fh	受信 FIFO 割り込みレベル・ビット 受信 FIFO は、FIFO ステータス・ビット (RXFFST4~0) が FIFO レベル・ビット (RXFFIL4~0) 以上である場合に割り込みを生成します。割り込みを生成するためにこれらのビットに割り当てることができる最大値は、RX FIFO の深度を超えることはできません。これらのビットのリセット後のデフォルト値は 11111b です。ISR をサーブिसするための CPU オーバーヘッドと受信 SCI データの最適な使用状況を比較して、アプリケーションのニーズに最適な RXFFIL を設定する必要があります。 リセット・タイプ：SYSRSn

1.15.2.12 SCIFFCT レジスタ (オフセット = Ch) [リセット = 0h]

図 1-22 に SCIFFCT を示し、表 1-20 にその説明を示します。

概略表に戻ります。

SCIFFCT には自動ポー検出のステータスが含まれ、自動ポー・フラグをクリアし、検出ビットをキャリブレーションします。

図 1-22. SCIFFCT レジスタ

15	14	13	12	11	10	9	8
ABD	ABDCLR	CDC	RESERVED				
R-0h	W-0h	R/W-0h	R-0h				
7	6	5	4	3	2	1	0
FFTXDLY							
R/W-0h							

表 1-20. SCIFFCT レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15	ABD	R	0h	自動ポー検出 (ABD) ビット リセット・タイプ : SYSRSn 0h (R/W) = 自動ポー検出は未完了。「A」、「a」文字が正しく受信されていません。 1h (R/W) = SCI 受信レジスタで自動ポー・ハードウェアが「A」または「a」文字を検出。自動ポー検出完了。
14	ABDCLR	W	0h	ABD クリア・ビット リセット・タイプ : SYSRSn 0h (R/W) = 0 を書き込んでも ABD フラグ・ビットに影響なし。ビットは 0 をリードバック。 1h (R/W) = 1 を書き込んでビット 15 の ABD フラグをクリア。
13	CDC	R/W	0h	CDC キャリブレーション A 検出ビット リセット・タイプ : SYSRSn 0h (R/W) = 自動ポー・アライメントはディセーブル 1h (R/W) = 自動ポー・アライメントはイネーブル
12~8	RESERVED	R	0h	予約済み
7~0	FFTXDLY	R/W	0h	FIFO 転送遅延。これらのビットは、FIFO 送信バッファからシフト・レジスタに送信する各転送の遅延を定義します。遅延は SCI シリアル・ポー・クロック・サイクル数で定義します。8 ビット・レジスタでは、0 ポー・クロック・サイクルの最小遅延と、256 ポー・クロック・サイクルの最大遅延を定義できます。 FIFO モードでは、シフト・レジスタと FIFO の間のバッファ (TXBUF) は、シフト・レジスタが最後のビットのシフトを完了した後でのみ入力する必要があります。これは、データ・ストリームへの転送間の遅延を渡すために必要です。FIFO モードでは、TXBUF を追加のバッファ・レベルとして扱わないでください。送信の遅延機能は、標準的な UARTS のように、RTS/CTS 制御を使用しない自動フロー方式を作成するのに役立ちます。 SCI がストップ・ビット 1 つで構成されている場合、1 つのフレームとその次のフレームの間に FFTXDLY によって挿入される遅延は、FFTXDLY で設定されているポー・クロック・サイクル数になります。 SCI がストップ・ビット 2 つで構成されている場合、1 つのフレームとその次のフレームの間に FFTXDLY によって挿入される遅延は、FFTXDLY で設定されているポー・クロック・サイクル数から 1 を引いたサイクル数になります。 リセット・タイプ : SYSRSn

1.15.2.13 SCIPRI レジスタ (オフセット = Fh) [リセット = 0h]

図 1-23 に SCIPRI を示し、表 1-21 にその説明を示します。

概略表に戻ります。

SCIPRI は、エミュレーションの一時停止イベントが発生したときの動作を決定します。

図 1-23. SCIPRI レジスタ

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			FREESOFT			RESERVED	
R-0h			R/W-0h			R-0h	

表 1-21. SCIPRI レジスタのフィールドの説明

ビット	フィールド	タイプ	リセット	説明
15~8	RESERVED	R	0h	予約済み
7~5	RESERVED	R	0h	予約済み
4~3	FREESOFT	R/W	0h	これらのビットは、エミュレーションの一時停止イベントが発生したとき (デバッガがブレークポイントに達したときなど) の動作を決定します。ペリフェラルは、実行中の処理を継続するか (フリーラン・モード)、直ちに停止するか、現在の動作 (現在の受信 / 送信シーケンス) が完了した時点で停止できます (ストップ・モード)。 リセット・タイプ : SYSRSn 0h (R/W) = 一時停止が発生したときに即停止 1h (R/W) = 現在の受信 / 送信シーケンスを完了してから停止 2h (R/W) = フリーラン 3h (R/W) = フリーラン
2~0	RESERVED	R	0h	予約済み

1.15.3 SCI レジスタから Driverlib 関数へのマッピング

表 1-22. SCI レジスタから Driverlib 関数へのマッピング

ファイル	Driverlib 機能
SCICCR	
sci.c	SCI_setConfig
sci.h	SCI_setParityMode
sci.h	SCI_getParityMode
sci.h	SCI_setAddrMultiProcessorMode
sci.h	SCI_setIdleMultiProcessorMode
sci.h	SCI_getConfig
sci.h	SCI_enableLoopback
sci.h	SCI_disableLoopback
SCICTL1	
sci.c	SCI_enableInterrupt
sci.c	SCI_disableInterrupt
sci.c	SCI_setWakeFlag
sci.h	SCI_enableModule
sci.h	SCI_disableModule
sci.h	SCI_performSoftwareReset
SCIHBAUD	
sci.c	SCI_setConfig

表 1-22. SCI レジスタから Driverlib 関数へのマッピング (continued)

ファイル	Driverlib 機能
sci.c	SCI_setBaud
sci.h	SCI_lockAutobaud
sci.h	SCI_getConfig
SCILBAUD	
sci.c	SCI_setConfig
sci.c	SCI_setBaud
sci.h	SCI_lockAutobaud
sci.h	SCI_getConfig
SCICTL2	
sci.c	SCI_enableInterrupt
sci.c	SCI_disableInterrupt
sci.c	SCI_getInterruptStatus
sci.h	SCI_isSpaceAvailableNonFIFO
sci.h	SCI_isTransmitterBusy
SCIRXST	
sci.c	SCI_getInterruptStatus
sci.h	SCI_isDataAvailableNonFIFO
sci.h	SCI_getRxStatus
SCIRXEMU	
-	
SCIRXBUF	
sci.c	SCI_readCharArray
sci.h	SCI_readCharBlockingFIFO
sci.h	SCI_readCharBlockingNonFIFO
sci.h	SCI_readCharNonBlocking
SCITXBUF	
sci.c	SCI_writeCharArray
sci.h	SCI_writeCharBlockingFIFO
sci.h	SCI_writeCharBlockingNonFIFO
sci.h	SCI_writeCharNonBlocking
SCIFFTX	
sci.c	SCI_enableInterrupt
sci.c	SCI_disableInterrupt
sci.c	SCI_getInterruptStatus
sci.c	SCI_clearInterruptStatus
sci.h	SCI_setFIFOInterruptLevel
sci.h	SCI_getFIFOInterruptLevel
sci.h	SCI_disableModule
sci.h	SCI_enableFIFO
sci.h	SCI_disableFIFO
sci.h	SCI_isFIFOEnabled
sci.h	SCI_resetTxFIFO
sci.h	SCI_resetChannels
sci.h	SCI_getTxFIFOStatus
sci.h	SCI_isTransmitterBusy

表 1-22. SCI レジスタから Driverlib 関数へのマッピング (continued)

ファイル	Driverlib 機能
SCIFFRX	
sci.c	SCI_enableInterrupt
sci.c	SCI_disableInterrupt
sci.c	SCI_getInterruptStatus
sci.c	SCI_clearInterruptStatus
sci.h	SCI_setFIFOInterruptLevel
sci.h	SCI_getFIFOInterruptLevel
sci.h	SCI_enableFIFO
sci.h	SCI_resetRxFIFO
sci.h	SCI_getRxFIFOStatus
sci.h	SCI_getOverflowStatus
sci.h	SCI_clearOverflowStatus
SCIFFCT	
sci.h	SCI_lockAutobaud
SCIPRI	
-	

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス・デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、または [ti.com](https://www.ti.com) やかかる TI 製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。

お客様がいかなる追加条項または代替条項を提案した場合でも、TI はそれらに異議を唱え、拒否します。

郵送先住所 : Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated