

Errata

TMS320F28003x Real-Time MCUs Silicon Errata**シリコン リビジョン 0****概要**

この文書では、機能仕様に対する既知の例外 (アドバイザリ) について説明します。本文書には、使用上の注意事項も記載されています。使用上の注意は、デバイスの動作が推定または文書化された動作と一致しない可能性がある状況を示しています。これには、デバイスの性能や機能の正確さに影響を与える動作が含まれる場合があります。

目次

1 使用上の注意およびアドバイザリ マトリックス	2
1.1 使用上の注意マトリックス.....	2
1.2 アドバイザリ マトリックス.....	2
2 命名法、パッケージのマーキングとリビジョンの識別	4
2.1 デバイスおよび開発ツールの命名規則.....	4
2.2 サポート対象デバイス.....	4
2.3 パッケージの記号表記およびリビジョンの識別.....	5
3 シリコン リビジョン 0 の使用上の注意とアドバイザリ	8
3.1 シリコン リビジョン 0 の使用上の注記.....	8
3.2 シリコン リビジョン 0 のアドバイザリ.....	11
4 ドキュメントのサポート	41
5 商標	41
6 改訂履歴	41

図の一覧

図 2-1. PZ パッケージのパッケージ マーキング.....	5
図 2-2. PN パッケージのパッケージ マーキング.....	5
図 2-3. PM パッケージのパッケージ マーキング.....	6
図 2-4. PT パッケージのパッケージ マーキング.....	6
図 3-1. AGPIO と AIO アナログ ピン タイプを含むアナログ サブシステム図.....	18
図 3-2. 望ましくないトリップ イベントとブランキング ウィンドウの期限切れ.....	22
図 3-3. 結果として望ましくない ePWM 出力が発生する可能性があります.....	22
図 3-4. パイプラインにストールがない場合の問題のパイプライン図.....	25
図 3-5. 命令 I1 の E3 スロットにストールがある場合の問題のパイプライン図.....	26
図 3-6. 回避方法が適用されたパイプライン図.....	27

表の一覧

表 1-1. 使用上の注意マトリックス.....	2
表 1-2. アドバイザリ マトリックス.....	2
表 2-1. リビジョンの識別.....	7
表 3-1. ADCCTL2 レジスタ.....	12
表 3-2. 特定のアナログ入力ピンの使用事例の組み合わせ.....	18
表 3-3. 標準モードホストを備えたターゲット トランスミッタとしての C2000 のデータ立ち上がり時間要件.....	28
表 3-4. 一般的なバス容量 (C_b) のプルアップ抵抗 (R_p) 値.....	29
表 3-5. アドバイザリによって影響を受けるメモリ.....	31

1 使用上の注意およびアドバイザリ マトリックス

表 1-1 に、すべての使用上の注意と、該当するシリコンのリビジョンを示します。表 1-2 にすべてのアドバイザリ、影響を受けるモジュール、および適用可能なシリコン リビジョンを一覧表示します。

1.1 使用上の注意マトリックス

表 1-1. 使用上の注意マトリックス

番号	タイトル	影響を受けるシリコン のリビジョン
		0
セクション 3.1.1	PIE: 双方向 PIEACK 書き込みと手動 CPU 割り込みマスク クリア後のスプリアス ネスト割り込み	あり
セクション 3.1.2	繰り返しブロックでネストされた割り込みを使用する際の注意	あり
セクション 3.1.4	セキュリティ: プライマリ防御層はチップの境界を保護します。これは、JTAGLOCK およびフラッシュからのゼロ ビン ブート機能を有効化することから始まります	あり
セクション 3.1.3	I2C スレーブ モード: 割り込み応答時間が長いアプリケーションでステータス フラグを管理する	あり

1.2 アドバイザリ マトリックス

表 1-2. アドバイザリ マトリックス

モジュール	説明	影響を受けるシリ コンのリビジョン
		0
ADC	ADC: INTxCONT (割り込み継続モード) が設定されていない場合、割り込みは停止する可能性があります	あり
ADC	ADC: ADCCLK 分周器により ADC の性能が低下	あり
ADC	ADC: 結果の DMA 読み取り結果	あり
アナログ サブシステム	アナログ サブシステム: ADCDACLOOPBACK レジスタのビット 1 への書き込みは機能しません	あり
BOR	BOR: 2.45V ~ 3.0V の VDDIO は、複数の XRSn パルスを生成する可能性があります	あり
C28	C28: 高速整数除算 (FID) と DIVF64 の不正確な結果	あり
CMPSS	CMPSS: COMPxLATCH は、特定の条件では正しくクリアされないことがあります	あり
CMPSS	CMPSS: コンパレータ入力ピンに AGPIO 機能があり、ADC が入力ピンをサンプリングしている場合、CMPSS グリッチが発生する可能性があります	あり
DCAN	DCAN: DCAN FIFO モードでは、受信したメッセージは FIFO バッファの順序に従わずに配置できます	あり
MCAN	MCAN: 同じメッセージ ID で設定された専用 Tx バッファから送信する場合のメッセージ順序の反転	あり
ePWM	ePWM: ePWM グリッチは、ブランキング ウィンドウの終了時にトリップがアクティブのままの場合、発生する可能性があります	あり
ePWM	ePWM ワンショット / CBC トリップ イベント DCxEVTy.force はトリップ条件を設定しません	あり
ePWM	ePWM: ブランキング ウィンドウ開始後の最初の 3 サイクルの間、トリップ イベントはブランキング ウィンドウによってフィルタされません	あり
eQEP	eQEP: インデックス中の方向変更時に位置カウンタが正しくリセットされません	あり
FPU	FPU: FPU-to-CPU レジスタ移動操作の前には FPU 2p 操作	あり
I2C	I2C: ターゲットトランスミッタ モード、標準モードの SDA タイミングの制限	あり
LIN	LIN: 同期フィールドが誤っている場合、不整合同期フィールド エラー (ISFE) フラグ/割り込みが設定されません	あり
メモリ	メモリ: 有効なメモリを超えたプリフェッチ	あり
ブート ROM	ブート ROM: TMX デバイス上の MCAN ブートローダーによるデータ オーバーラン	あり
ブート ROM	セキュア ライブ ファームウェア アップデート (LFU) ブート モードは非推奨です	あり
フラッシュ	フラッシュ: フラッシュのプリフェッチを無効にせずに Fapi_setActiveFlashBank() を実行すると、ITRAP が発生することがあります	あり
システム	システム: Pagesel=0 モードでは、HIC 不正読み取りエラー フラグがアサートされません	あり
システム	システム: CLKSRCCTL1 への複数の連続した書き込みを行うと、システムがハングする可能性があります	あり
診断	HWBIST 使用中にスプリアス割り込みを回避する	あり
PLL	PLL リファレンス クロック消失検出: クロック消失フラグが正しくアクティブ化されない可能性があります	あり
SDFM	SDFM: スレッシュホールド設定 (LLT、HLT)、フィルタ タイプ、COSR 設定を動的に変更すると、スプリアス コンパレータ イベントがトリガされます	あり
SDFM	SDFM: データ フィルタ設定 (フィルタ タイプや DOSR など) を動的に変更すると、誤ったデータ確認イベントがトリガされます	あり

表 1-2. アドバイザリ マトリックス (続き)

モジュール	説明	影響を受けるシリコンのリビジョン
		0
SDFM	SDFM:SD 変調器の 3 クロック サイクル以内に SDCPARMx レジスタのビット フィールド CEVT1SEL、CEVT2SEL、および HZEN に連続して 2 回書き込みを行うと、SDFM ステートマシンが破損し、誤ったコンパレータ イベントが発生する可能性があります	あり
ウォッチドッグ	ウォッチドッグ:WDKEY レジスタは EALLOW 保護されていません	あり

2 命名法、パッケージのマーキングとリビジョンの識別

2.1 デバイスおよび開発ツールの命名規則

製品開発サイクルの段階を示すために、TI では DSP デバイスとサポート ツールすべての型番に接頭辞を割り当てます。DSP 商用ファミリの製品には、次の 3 つの接頭辞のいずれかが付いています。TMX、TMP、TMS (例: TMS320F280039C)。テキサス インストルメンツでは、サポート ツールに対して使用可能な 3 つの接頭辞のうち、TMDX、TMDS の 2 つを推奨しています。これらの接頭辞は、製品開発の進展段階を表します。段階には、エンジニアリング プロトタイプ(TMX および TMDX)から、完全認定済みの量産デバイス/ツール(TMS および TMDS)まであります。

デバイスの開発進展フロー:

TMX 実験的デバイス。最終デバイスの電気的特性を必ずしも表さず、量産アセンブリ・フローを使用しない可能性があります。

TMP プロトタイプ・デバイス。最終的なシリコン・ダイとは限らず、最終的な電気的特性を満たさない可能性があります。

TMS 認定済みのシリコン・ダイの量産バージョン。

サポート・ツールの開発進展フロー:

TMDX 開発サポート製品。テキサス・インストルメンツの社内認定試験はまだ完了していません。

TMDS 完全に認定済みの開発サポート製品です。

TMX および TMP デバイスと TMDX 開発サポート・ツールは、以下の免責事項の下で出荷されます。

「開発中の製品は、社内での評価用です」。

量産デバイスおよび TMDS 開発サポート・ツールの特性は完全に明確化されており、デバイスの品質と信頼性が十分に示されています。テキサス・インストルメンツの標準保証が適用されます。

プロトタイプ・デバイス (X または P) の方が標準的な量産デバイスに比べて故障率が大いといと予測されます。これらのデバイスは予測される最終使用時の故障率が未定義であるため、テキサス・インストルメンツでは、それらのデバイスを量産システムで使用しないよう推奨しています。認定済みの量産デバイスのみを使用する必要があります。

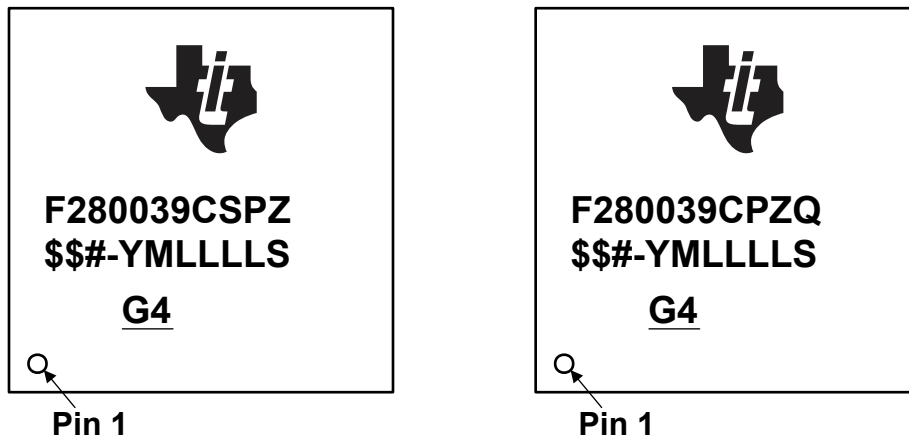
2.2 サポート対象デバイス

本文書は、以下のデバイスをサポートしています。

- [TMS320F280039C](#)
- [TMS320F280039C-Q1](#)
- [TMS320F280039](#)
- [TMS320F280039-Q1](#)
- [TMS320F280038C-Q1](#)
- [TMS320F280038-Q1](#)
- [TMS320F280037C](#)
- [TMS320F280037C-Q1](#)
- [TMS320F280037](#)
- [TMS320F280037-Q1](#)
- [TMS320F280036C-Q1](#)
- [TMS320F280036-Q1](#)
- [TMS320F280034](#)
- [TMS320F280034-Q1](#)
- [TMS320F280033](#)
- [TMS320F280033-Q1](#)

2.3 パッケージの記号表記およびリビジョンの識別

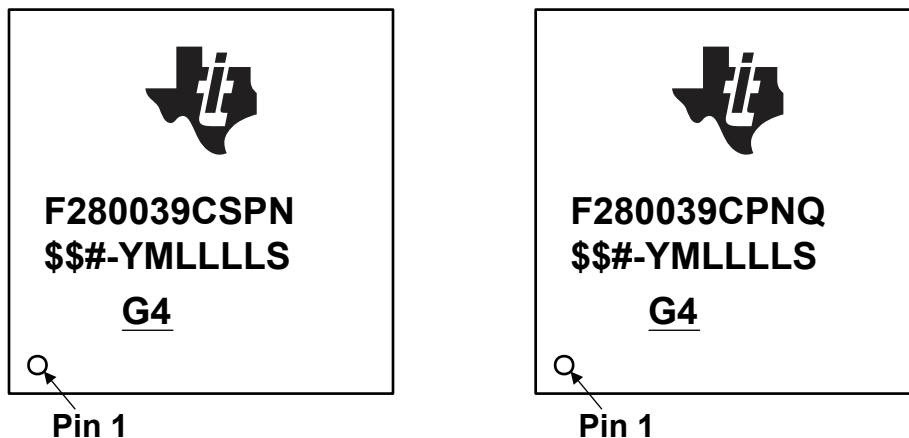
パッケージ マーキングを [図 2-1](#)、[図 2-2](#)、[図 2-3](#)、および [図 2-4](#) に示します。[表 2-1](#) に、シリコンのリビジョン コードを示します。



\$\$ = Wafer Fab Code (one or two characters)
= Silicon Revision Code
YM = 2-digit Year/Month Code
LLLL = Assembly Lot Code
S = Assembly Site Code per QSS 005-120

G4 = Green (Low Halogen and RoHS-compliant)

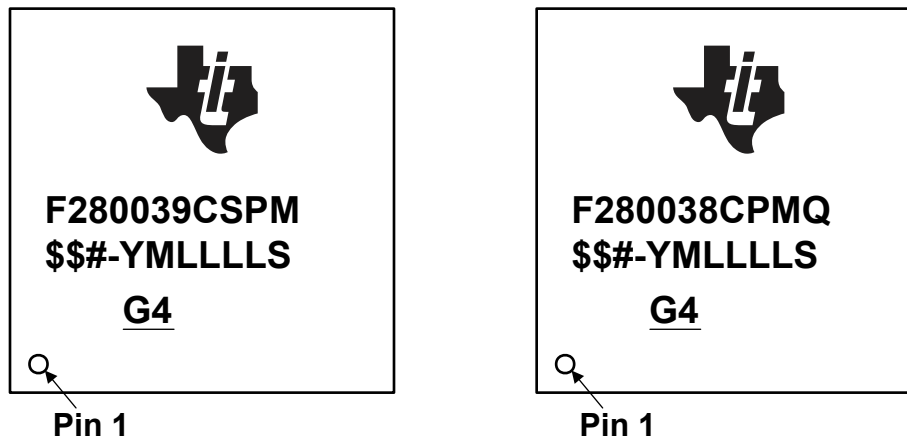
図 2-1. PZ パッケージのパッケージ マーキング



\$\$ = Wafer Fab Code (one or two characters)
= Silicon Revision Code
YM = 2-digit Year/Month Code
LLLL = Assembly Lot Code
S = Assembly Site Code per QSS 005-120

G4 = Green (Low Halogen and RoHS-compliant)

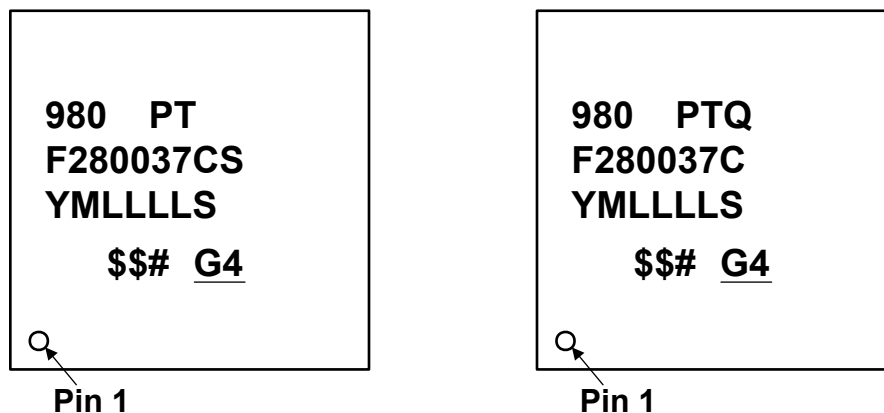
図 2-2. PN パッケージのパッケージ マーキング



\$\$ = Wafer Fab Code (one or two characters)
= Silicon Revision Code
YM = 2-digit Year/Month Code
LLLL = Assembly Lot Code
S = Assembly Site Code per QSS 005-120

G4 = Green (Low Halogen and RoHS-compliant)

図 2-3. PM パッケージのパッケージ マーキング



980 = TI EIA Code
YM = 2-digit Year/Month Code
LLLL = Assembly Lot Code
S = Assembly Site Code per QSS 005-120
\$\$ = Wafer Fab Code (one or two characters)
= Silicon Revision Code

G4 = Green (Low Halogen and RoHS-compliant)

図 2-4. PT パッケージのパッケージ マーキング

表 2-1. リビジョンの識別

シリコンのリビジョン コード	シリコンのリビジョン	REVID ⁽¹⁾ アドレス: 0x5D00C	備考 ⁽²⁾
空白	0	0x0000 0000	このシリコン リビジョンは TMX および TMS とし て供給されます。

- (1) シリコンのリビジョン ID
- (2) 注文可能なデバイス番号については、[TMS320F28003x リアルタイム マイクロ コントローラ](#)のデータシートのパッケージ情報表を参照してください。

3 シリコン リビジョン 0 の使用上の注意とアドバイザー

3.1 シリコン リビジョン 0 の使用上の注記

このセクションでは、シリコン リビジョン 0 に適用されるすべての使用上の注意を示しています。

3.1.1 PIE: 双方向 PIEACK 書き込みと手動 CPU 割り込みマスク クリア後のスプリアス ネスト割り込み

影響を受けるリビジョン: 0

ネストされた割り込みに使用される特定のコード・シーケンスでは、CPU と PIE が矛盾した状態に移行し、望ましくない割り込みをトリガできるようになります。この状態に入るために必要な条件は次のとおりです。

1. PIEACK クリアの後に、グローバル割り込み友好状態 (EINT または ASM (「CLRC INTM」)) が直ちに続きます。
2. ネストされた割り込みにより、そのグループの一つ以上の PIEIER ビットがクリアされます。

不要な割り込みがトリガされるかどうかは、システム内の他の割り込みの構成とタイミングによって異なります。これは、ほとんどのアプリケーションではまれなイベントまたは存在しないイベントであると予想されます。発生した場合、不要な割り込みはネストされた割り込みの PIE グループの最初の割り込みになり、ネストされた割り込みが CPU 割り込みを再度イネーブルにした後にトリガされます (EINT または ASM (「CLRC INTM」))。

回避方法: PIEACK 書き込みと CPU 割り込み有効化の間に NOP を追加します。コード例を以下に示します。

```
//Bad interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF;    //Enable nesting in the PIE
EINT;                                //Enable nesting in the CPU

//Good interrupt nesting code
PieCtrlRegs.PIEACK.all = 0xFFFF;    //Enable nesting in the PIE
asm(" NOP");                          //Wait for PIEACK to exit the pipeline
EINT;                                //Enable nesting in the CPU
```

3.1.2 繰り返しブロックでネストされた割り込みを使用する際の注意

影響を受けるリビジョン: 0

ネスト機能を使用するために、ユーザーが割り込みサービス ルーチン (ISR) 内で EINT 命令を使用して割り込みを有効にしている場合、ユーザーは ISR を終了する前に DINT アセンブリ命令を使用して割り込みを無効にする必要があります。この操作を怠ると、RB レジスタのビットが正しく復元されず、コードが未定義の動作が発生する可能性があります。

RPTB ASM 命令がアプリケーション内で使用されていない場合、問題はありません。C 言語でコーディングしている場合、生成された逆アセンブリの解析を実行して、これを確認する必要があります。

ISR を C 言語でコーディングしている場合は、C28x C コンパイラが上記の処理をすることがあり、何もする必要はありません。ISR が C28x アセンブリ言語でコーディングされている場合、上記のガイダンスに従う必要があります。

注

2016 年 4 月以降の CGT パッケージでリリースされた CGT v15.12.2.LTS は、この要件に自動的に対応します。DINT は、CGT ツールの以前のバージョンでのみ追加する必要があります。

3.1.3 I2C スレーブ モード: 割り込み応答時間が長いアプリケーションでステータス フラグを管理する

影響を受けるリビジョン: 0

アプリケーションが I2C 割り込みの処理を遅延させている間に I2C バスのアクティビティが進行し続けた場合、その後のバス イベントによって、I2C ステータスレジスタ (I2CSTR) 内の複数のステータス フラグが同時に有効になる可能性があります。I2CIER で、これらの割り込みイベントのいずれかが複数が有効になっている場合、以後の割り込みが失われる可能性があります。

- これは、I2C マスタとして動作している場合は当てはまりません。
- これは、ソフトウェアによって各 I2CSTR フラグがクリアされてから、次のフラグがセットされる場合は当てはまりません。

回避方法: I2CISRC.INTCODE フィールドの値が 0 (保留中の割り込みなし) になるまで I2CISRC.INTCODE フィールドをポーリングし、割り込みソースに応じたアクションを実行してください。

```
__interrupt void I2CA_ISR(void){
    uint16_t isrc;
    while((isrc = I2C_getInterruptSource(I2CA_BASE)) != I2C_INTSRC_NONE){
        switch(isrc) {
            case I2C_INTSRC_ARB_LOST:
                /* ARBLINT ISR code */
                break;
            case I2C_INTSRC_NO_ACK:
                /* NACKINT ISR code */
                break;
            case I2C_INTSRC_REG_ACCESS_RDY:
                /* ARDYINT ISR code */
                I2C_clearStatus(I2CA_BASE, I2C_STS_REG_ACCESS_RDY);
                break;
            case I2C_INTSRC_RX_DATA_RDY:
                /* RRDYINT ISR code, including I2C_getData() call */
                break;
            case I2C_INTSRC_TX_DATA_RDY:
                /* XRDYINT ISR code, including I2C_putData() call */
                break;
            case I2C_INTSRC_STOP_CONDITION:
                /* SCDINT ISR code */
                break;
            case I2C_INTSRC_ADDR_TARGET:
                /* AASINT ISR code */
                break;
        }
        NOP;
        NOP;
    }
    Interrupt_clearACKGroup();
}
```

3.1.4 セキュリティ: プライマリ防御層はチップの境界を保護します。これは、JTAGLOCK およびフラッシュからのゼロピンブート機能を有効化することから始まります

影響を受けるリビジョン: 0

デバイスのセキュリティは、不正なコードがデバイスに侵入して実行されることを許可されていないという前提に依存しています。そのため、セキュリティを懸念するユーザーが常に有効にする必要のある 2 つの機能をデバイスに搭載しています。

- **JTAGLOCK**

フラッシュの USER OTP 領域で有効にすると、JTAGLOCK 機能はデバイス上のリソースへの JTAG アクセス (デバッグ接続など) を無効にし、権限のない者が JTAG インターフェイスを使用してデバイスにコードをダウンロードすることをブロックします。JTAGLOCK が有効な場合でも、ユーザーは許可された当事者がパスワードを入力してロックを解除できるようにしたり、すべてゼロのパスワード値をプログラムして永続的にロックしたりすることができます。

- **ゼロピンブートからフラッシュ**

TI ROM に組み込まれた外部ブートローダは、ダウンロードされたコードの認証を実行しません。USER OTP でフラッシュブートモードとともにゼロピンブートオプションを有効にすると、ベースブート ROM の実行が終了した後、ブートプロセスが直ちに内部フラッシュにジャンプするように強制され、すべてのピンベースの外部ブートローダオプション (SCI、CAN、パラレルなど) がブート時に実行されなくなります。最高のセキュリティを実現するために、セキュアフラッシュブートモードを選択できます。これにより、ベースブート ROM にジャンプする前にフラッシュコードを事前にチェックできるようになります。

JTAG が永続的にロックされ、Zero-pin Boot to Flash (ゼロピンブートからフラッシュ) オプションが有効になっている場合、JTAG または内蔵ブートローダを介してデバイスと通信するプログラミングツールは動作しません。ファームウェアのアップグレードを実行する機能が必要な場合、更新を安全に管理および実行するため、コードをフラッシュに事前格納する必要があります。

3.2 シリコン リビジョン 0 のアドバイザー

このセクションでは、シリコン リビジョン 0 に適用されるすべてのアドバイザーを示しています。

アドバイザー **ADC:INTxCONT (割り込み継続モード) が設定されていない場合、割り込みは停止する可能性があります**

影響を受けるリビジョン 0

詳細

ADCINTSELxNx[INTxCONT] = 0 の場合、ADCINTFLG が設定されると割り込みは停止し、追加の ADC 割り込みは発生しません。

ADCINTFLGCLR レジスタのソフトウェア書き込みとともに ADC 割り込みが同時に発生すると、ADCINTFLG が予期せず設定されたままになり、将来の ADC 割り込みをブロックします。

回避方法

1. ADCINTFLG が追加の ADC 割り込みをブロックしないように、Continue-to-Interrupt モードを使用します。

```
ADCINTSEL1N2[INT1CONT] = 1;
ADCINTSEL1N2[INT2CONT] = 1;
ADCINTSEL3N4[INT3CONT] = 1;
ADCINTSEL3N4[INT4CONT] = 1;
```

2. この状態を回避するために、次の ADC 割り込みが発生する前に、ADC ISR をサービスし、ADCINTFLG をクリアするのに十分な時間を常に確保してください。
3. ADCINTFLG をクリアするとき、ISR のオーバーフロー状態を確認します。
ADCINTFLGCLR への書き込み直後に ADCINTOVF をチェックし、これが設定されている場合は、ADCINTFLGCLR をもう一度書き込んで ADCINTFLG がクリアされていることを確認します。ADCINTOVF レジスタが設定され、ADC 変換割り込みが失われたことを示します。

```
AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;           //clear INT1 flag
if(1 == AdcaRegs.ADCINTOVF.bit.ADCINT1)           //ADCINT overflow
{
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;         //clear INT1 again
    // If the ADCINTOVF condition will be ignored by the application
    // then clear the flag here by writing 1 to ADCINTOVFCLR.
    // If there is a ADCINTOVF handling routine, then either insert
    // that code and clear the ADCINTOVF flag here or do not clear
    // the ADCINTOVF here so the external routine will detect the
    // condition.
    // AdcaRegs.ADCINTOVFCLR.bit.ADCINT1 = 1;      // clear OVF
}
```

アドバイザー

ADC:ADCCLK 分周器により ADC の性能が低下

影響を受けるリビジョン 0

詳細

分数 SYSCLK から ADCCLK への分周器 (ADCCTL2.PRESCALE フィールドで制御) を使用すると、このデバイスで ADC 性能が低下することが示されています。表 3-1 を参照してください。

表 3-1. ADCCTL2 レジスタ

性能が低下			
ビット	フィールド	値	説明
3-0	PRESCALE	0001	ADCCLK = SYSCLK/1.5
		0003	ADCCLK = SYSCLK/2.5
		...	
通常の性能			
ビット	フィールド	値	説明
3-0	PRESCALE	0000	ADCCLK = SYSCLK/1.0
		0002	ADCCLK = SYSCLK/2.0
		...	

回避方法

偶数 PRESCALE クロック分周器の値を使用します。偶数 PRESCALE 値では、整数クロック分周器が発生し、ADC の性能には影響しません。

アドバイザー

ADC:結果の DMA 読み取り結果

影響を受けるリビジョン 0

詳細

ADCINT フラグは、ADCRESULT の値がラッチされる前に設定できます (『TMS320F28003x リアルタイム マイコン』データシートの「ADC タイミング」表の t_{LAT} および $t_{INT(LATE)}$ 列を参照)。DMA は、ADCINT トリガが設定されてから 3 サイクル後、すぐに ADCRESULT の値を読み取ることができます。その結果、以下の条件がすべて満たされると、ユーザーが最新の結果を期待しているにもかかわらず、DMA が 1 つ前の ADCRESULT 値を読み取る可能性があります。

- ADC は後期割り込みモードです。
- ADC は、 $t_{INT(LATE)}$ が t_{LAT} より 3 サイクル以上前に発生するモードで動作します (ADCCTL2 [PRESCALE] > 2)。
- DMA は ADCINT 信号からトリガされます。
- DMA は、最初に他の値を読み取らずに、その ADCINT 信号に関連付けられている ADCRESULT 値を直ちに読み取ります。
- ADCINT トリガを受信したとき、DMA はアイドル状態でした。

上述の DMA のみが古いデータを読み取る可能性があります。次の DMA 以外の方法では常に、期待されるデータを読み取ります。

- ADCINT フラグが CLA タスクをトリガする。
- ADCINT フラグが CPU ISR をトリガする。
- CPU は ADCINT フラグをポーリングする。

回避方法

ADCINT フラグから 2 つの DMA チャンネルをトリガします。最初のチャンネルがダミー トランザクションとして機能するため、2 番目のチャンネルは常に新しい ADC 結果を読み取るための十分な遅延が得られます。

アドバイザリ アナログ サブシステム: **ADCDACLOOPBACK** レジスタのビット 1 への書き込みは機能しません

影響を受けるリビジョン 0

詳細

レジスタ ビット ADCDACLOOPBACK.ENLB2ADCB は機能せず、ビットに書き込んでも影響はありません。そのため、このレジスタを使用して ADCB の ADC-DAC ループバック機能をアクティブにはできません。

回避方法

- 診断安全対策として ADC の機能をテストするには、2 つのバッファ付き DAC の 1 つを使用して既知の電圧を出力し、ADCB を使用して DAC 電圧をサンプリングします。DACA の出力はチャンネル B15 と同じピンにあり、DACB の出力は B7 にあります。
- INTERNALTEST/ADC ゲイン バランシング メカニズムを使用して、 $0.9 \cdot V_{REFHI}$ をサンプリングします。詳細については、『[TMS320F28003x リアルタイム マイコン テクニカル リファレンス マニュアル](#)』の「A/D コンバータ (ADC)」の章にある「内部テスト モード」セクションを参照してください。

アドバイザリ **BOR: 2.45V ~ 3.0V の VDDIO は、複数の XRSn パルスを生成する可能性があります**

影響を受けるリビジョン 0

詳細

VDDIO 電源電圧が 2.45V ~ 3.0V のとき、BOR は XRSn のアサートおよびディアサートを繰り返すことがあります。XRSn ピンをシステム内の他のデバイスのリセットとして直接使用しないことを推奨します。

F28003x BOR は、これらの XRSn パルスが発生した場合でも、デバイスを既知のリセット状態に内部で保持するのに有効です。デバイスはアプリケーション コードやブートローダに分岐せず、VDDIO 電源が 3.0V を上回るまで、他のすべてのピンはリセット状態に保持されます。

回避方法

1. パワーアップ、パワーダウン、BOR イベント中は、追加の XRSn 遷移は無視します。追加の XRSn パルスは、F28003x デバイスの動作自体には影響しません。
2. XRSn パルスによって他のシステム コンポーネントで望ましくないシステム動作が発生する場合は、XRSn を使用して他のデバイスを駆動しないでください。これらのアプリケーションには、外部電圧スーパーバイザを使用できます。
3. 通常のパワーアップおよびパワーダウン時にこれらのパルスを回避する必要があるアプリケーションでは、以下のように動作します。
 - a. パワーアップ: [TMS320F28003x リアルタイム マイクロ コントローラ](#) データ シートの「推奨動作条件」表の SR_{SUPPLY} 要件に従い、追加の XRSn Low パルスは発生しません。
 - b. パワーダウン: パワーダウン時に XRSn がアサートされるのを防ぐため、VDDIO が 25 μ s 内で 3.0V~2.45V の範囲を通過するように電源を設計してください。XRSn の電圧上昇が許容される場合、XRSn に実装された RC 回路の時定数を計算し、その電圧がシステム指定のスレッシュホールドを超えないようにできます。

アドバイザリ

C28: 高速整数除算 (FID) と DIVF64 の不正確な結果

影響を受けるリビジョン 0

詳細

22.6.3.LTS または 21.6.2.LTS より前のコンパイラ バージョンを使用すると、特定の 2 サイクル FPU 命令で不正確な結果が発生する可能性があります。

以下の影響を受ける 2p FPU 命令の後に、2 つ未満の遅延スロットを持つ FID または DIVF64 命令のいずれかが続くと、レジスタの破損が発生し、不正確な結果につながる可能性があります。

2 サイクル (2p) FPU 命令で、FID または DIVF64 命令の前に 2 つの遅延スロットが必要:

- ADDF32
- SUBF32
- MPYF32
- MACF32

高速整数除算 (FID) 命令:

- ABSI32DIV32
- ABSI32DIV32U
- ABSI64DIV32
- ABSI64DIV32U
- ABSI64DIV64
- ABSI64DIV64U
- ENEGI32DIV32
- ENEGI64DIV32
- ENEGI64DIV64
- MNEGI32DIV32
- MNEGI64DIV32
- MNEGI64DIV64
- NEGI32DIV32
- NEGI64DIV32
- NEGI64DIV64
- SUBC2UI64
- SUBC4UI32

DIVF64 命令:

- PREDIVF64
- SUBC3F64
- POSTDIVF64

問題の例:

問題は、次のような順序で再現できます。

ADDF32	R0H, R0H, #32768	; 2p FPU instruction
NOP		; Only one delay slot (insufficient)
ABSI32DIV32	R2H, R1H, R3H	; Fast Integer Division instruction

この場合、R0H の値が R1H にリークし、不正確な結果が発生する可能性があります。NOP を追加すると、問題が解決します。

ADDF32	R0H, R0H, #32768	; 2p FPU instruction
NOP		; First delay slot
NOP		; Second delay slot (required)

アドバイザリ (続き)**C28: 高速整数除算 (FID) と DIVF64 の不正確な結果**

`ABSI32DIV32 R2H, R1H, R3H ; Fast Integer Division instruction`

回避方法

1. この問題を修正したコンパイラ バージョン 22.6.3.LTS、21.6.2.LTS またはそれ以降に更新してください。
2. アセンブリコードを使用する場合は、影響を受ける命令の組み合わせの間に 2 つの遅延スロットを設けてください。

アドバイザリ

CMPSS:COMPxLATCH は、特定の条件では正しくクリアされないことがあります

影響を受けるリビジョン 0

詳細

CMPSS ラッチ パスは、ソフトウェアによって (COMPSTSCLR を介して) または PWMSYNC によってクリアされるまで、ローカル ラッチ (COMPxLATCH) 内でトリップ状態を維持するように設計されています。

COMPxLATCH は、信号がデジタル化され、デジタル フィルタによって認定された後、コンパレータ出力によって間接的に設定されます。コンパレータ出力が COMPxLATCH に達することが期待される最大レイテンシは、CMPSS モジュール クロック サイクルでは次のように表されます。

$$\text{LATENCY} = 1 + (1 \times \text{FILTER_PRESCALE}) + (\text{FILTER_THRESH} \times \text{FILTER_PRESCALE})$$

ソフトウェアまたは PWMSYNC によって COMPxLATCH がクリアされると、ラッチ自体は必要に応じてクリアされますが、COMPxLATCH 前のデータ パスは、追加のレイテンシ数のモジュール クロック サイクルについて、コンパレータの出力値を反映していない可能性があります。

COMPxLATCH がクリアされたときにデジタル フィルタ出力が論理 1 に解決されると、次のクロック サイクルでラッチが再度セットされます。

回避方法

COMPxLATCH をクリアする前に、デジタル フィルタ出力を論理 0 に解決できます。

ソフトウェアによって COMPxLATCH がクリアされた場合、ラッチをクリアする前に、デジタル フィルタの出力状態を COMPSTS レジスタで確認できます。レイテンシ値が大きいと許容できない遅延が発生する場合、デジタル フィルタを再初期化することでフィルタ FIFO をフラッシュできます (CTRIPxFILCTL 経由)。

PWMSYNC によって COMPxLATCH がクリアされた場合、PWMSYNC が生成される前に少なくともレイテンシ サイクルにわたってコンパレータトリップ条件がクリアされるように、ユーザー アプリケーションを設計する必要があります。

アドバイザー

CMPSS : コンパレータ入力ピンに **AGPIO** 機能があり、**ADC** が入力ピンをサンプリングしている場合、**CMPSS** グリッチが発生する可能性があります

影響を受けるリビジョン 0

詳細

表 3-2 に、注意する必要がある特定のアナログ入力ピンの使用事例の組み合わせを示します。このテーブルに示すように、**CMPSS** 入力、**ADC** サンプリング、**AGPIO** の組み合わせには注意するか、回避方法を使用する必要があります。

表 3-2. 特定のアナログ入力ピンの使用事例の組み合わせ

特定のアナログ ピンで使用する機能	使用部品				
CMPSS コンパレータ入力	あり	-	あり	-	あり
ADC サンプリング	あり	あり	-	あり	あり
AGPIO アナログ ピン タイプ	あり	あり	あり	-	-
AIO アナログ ピン タイプ	-	-	-	あり	あり
結果	回避方法が必要		特別な分析や回避方法は不要		

AGPIO アナログ ピン パスには、 53Ω の追加の直列スイッチが含まれています。これにより、図 3-1 に示すように、**ADC** および **CMPSS** コンパレータと共有される低容量の絶縁型ノードが作成されます。**ADC** が (**ADC** サンプル / ホールド コンデンサに保存されている前の電圧に応じて) チャンネルをサンプリングするとき、このノードに外乱が生じ、それによって最大 50ns の誤 **CMPSS** 事象が発生する可能性があります。下の回避方法を実装することで、この潜在的な外乱に対応できます。

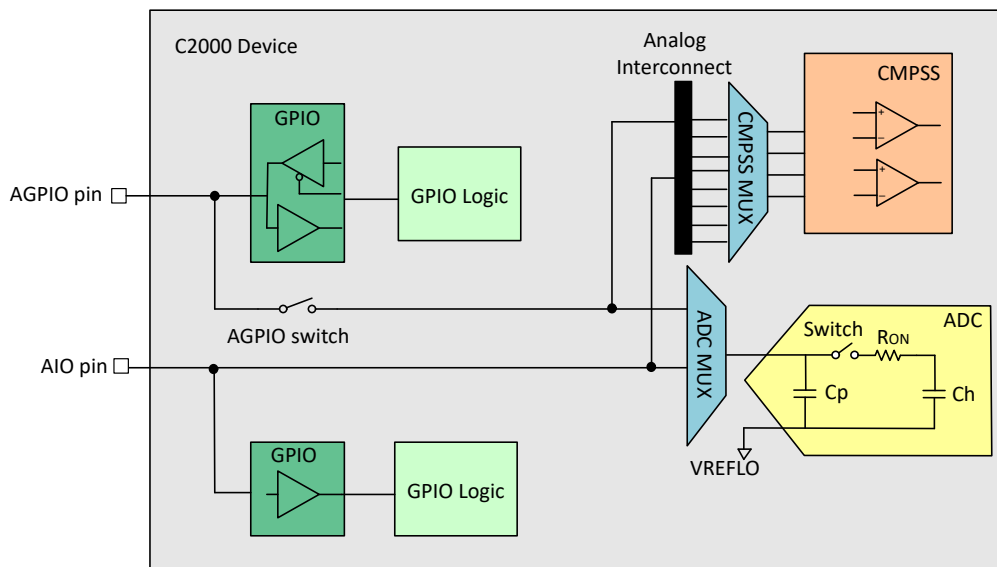


図 3-1. AGPIO と AIO アナログ ピン タイプを含むアナログ サブシステム図

回避方法

1. **ADC** と **CMPSS** の両方を同時に必要とするアナログ チャンネルには、別のピン (**AIO** ピン タイプ) を使用します。
2. **CMPSS** デジタル フィルタを 50ns 以上の設定で使用します。これにより、一時的な外乱がフィルタリングされます。
3. 外乱によって誤トリップが発生しないように、**ADC** の サンプル / ホールド コンデンサを事前に調整します。たとえば、影響を受けるチャンネルが読み取られる直前に、**ADC** の別のチャンネルから 3.3V 接続のダミー読み取りを実行します。これにより、外乱が正の方向になり、誤トリ

アドバイザリ (続き)

CMPSS: コンパレータ入力ピンに AGPIO 機能があり、ADC が入力ピンをサンプリングしている場合、CMPSS グリッチが発生する可能性があります

ップから離れます。誤トリップの極性が反転した場合、0V 信号の逆のダミー読み取りが使用されます。

アドバイザリ

DCAN :DCAN FIFO モードでは、受信したメッセージは **FIFO** バッファの順序に従わず配置されます

影響を受けるリビジョン 0**詳細**

DCAN FIFO モードでは、同じアービトレーション ID とマスク ID を持つ受信メッセージは受信した順序で FIFO に配置されることになります。その後、CPU は IF1/IF2 インターフェイス レジスタを介して、FIFO から受信したメッセージを取得します。一部のメッセージは、受信した順序で FIFO に配置される場合があります。アプリケーションの処理にとってメッセージの順序が重要である場合、この動作により DCAN FIFO モードが適切に使用できなくなります。

回避方法

DMA を使用して、IF3 レジスタ経由で FIFO を読み出します。FIFO にメッセージが受信されるたびに、データも IF3 レジスタにコピーされ、DMA モジュールに対して DMA 要求が生成されてデータが読み出されます。

アドバイザリ **MCAN: 同じメッセージ ID で設定された専用 Tx バッファから送信する場合のメッセージ順序の反転**

影響を受けるリビジョン 0

詳細

複数の Tx バッファが同じメッセージ ID で設定されます。これらの Tx バッファの送信は、個別の Tx 要求間に遅延を加えて、昇順に要求されます。個々の Tx 要求間の遅延によっては、Tx バッファは Tx バッファ番号の予想される昇順で送信されない場合があります。

回避方法

まず、同じメッセージ ID を持つ Tx メッセージのグループをメッセージ RAM に書き込みます。次に、**TXBAR** への単一の書き込みアクセスによって、これらすべてのメッセージの同時送信を要求します。

特定の順序で同じメッセージ ID を持つ複数のメッセージを送信するには、専用 Tx バッファの代わりに Tx FIFO を使用します。

アドバイザー

ePWM:ePWM グリッチは、ブランキング ウィンドウの終了時にトリップがアクティブのままの場合、発生する可能性があります

影響を受けるリビジョン 0**詳細**

ブランキング ウィンドウは通常、システムへの誤ったトリップを引き起こす可能性のある遷移中の PWM トリップ イベントをマスクするために使用されます。ブランキング ウィンドウ サイクルの終了後、3 未満の ePWM クロックの間 ePWM トリップ イベントがアクティブのまま維持されている場合、ePWM 出力に望ましくないグリッチが発生する可能性があります。

図 3-2 に、不要な EPWM 出力が発生する可能性のある時間を示します。

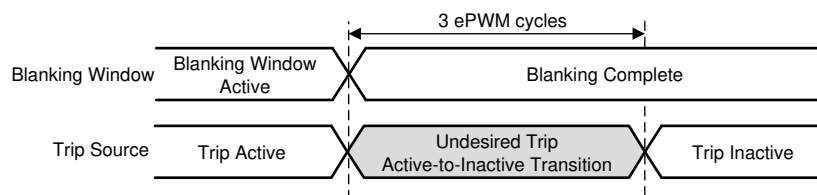


図 3-2. 望ましくないトリップ イベントとブランキング ウィンドウの期限切れ

図 3-3 に、ブランキング ウィンドウが閉じる前または 3 サイクル後にトリップイベントが 1 サイクル以内に終了した場合に可能性のある 2 つの ePWM 出力を示します。

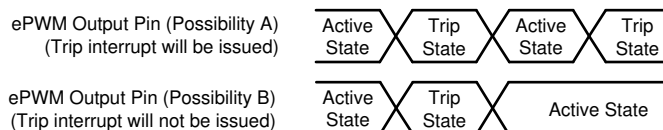


図 3-3. 結果として望ましくない ePWM 出力が発生する可能性があります

回避方法

ブランキング ウィンドウを延長または短縮して、不要なトリップ動作を回避します。

アドバイザリ **ePWM: ePWM ワンショット / CBC トリップ イベント DCxEVTy.force はトリップ条件を設定しません**

影響を受けるリビジョン 0

詳細

DCxEVTy.force 信号は、ワンショットトリップ イベントとサイクルバイサイクルトリップ イベントの入力として使用できます。DCxCTL[EVT1/2FRCSYNCSEL] = 1 に設定しても (非同期パス経由で DCxEVT 信号を渡します)、ワンショットまたはサイクルバイサイクルのトリップ イベント条件が適切に設定されず、トリップ イベントが検出されません。

回避方法

DCxCTL.EVT1/2FRCSYNCSEL と DCxCTL.EVT1/2LATSEL を備えたデバイスには、2 つの選択肢があります。それ以外の場合では、同期パスが、ワンショットまたはサイクルバイサイクルのトリップ イベントで使用できる唯一の選択肢となります。

非同期パス (非同期パスを介して DCxEVT 信号を渡します)

非同期パスを使用してワンショットまたはサイクルバイサイクルのトリップ条件をラッチするには、以下の設定を行う必要があります。

PWM をトリップするために、XBAR_invertEPWMSignal() を使用して EPWMXBAR の状態を反転できます。

1. DCxCTL.EVT1/2FRCSYNCSEL = 1
2. DCxCTL.EVT1/2LATSEL = 1

同期パス (同期パスを介して DCxEVT 信号を渡します)

1. DCxCTL.EVT1/2FRCSYNCSEL = 0

アドバイザリ **ePWM: ブランキング ウィンドウ開始後の最初の 3 サイクルの間、トリップ イベントはブランキング ウィンドウによってフィルタされません**

影響を受けるリビジョン 0

詳細

ブランキング ウィンドウは、ブランキング ウィンドウの開始後の最初の 3 サイクルのトリップ イベントをブランクにしません。DCEVTFILT は、DCxEVTy 信号の変更を継続的に反映することができます。DCEVTFILT が有効化されている場合、設定されている後続のサブシステム (たとえば、トリップゾーン サブモジュール、TZ 割り込み、ADC SOC、PWM 出力) に影響を及ぼす可能性があります。

回避方法

ブランキングが必要な前に、ブランキング ウィンドウを 3 サイクル開始します。周期境界でブランキング ウィンドウが必要な場合、次の周期の開始 3 サイクル前にブランキング ウィンドウを開始します。これは、ブランキング ウィンドウが期間の境界を越えて持続するため、機能します。

アドバイザー**eQEP: インデックス中の方向変更時に位置カウンタが正しくリセットされません****影響を受けるリビジョン 0****詳細**

PCRM = 0 構成を使用しているときに、インデックス入力がアクティブのときに方向変更が発生すると、位置カウンタ(QPOSCNT)が誤ってリセットされ、カウンタ値の予期しない変化が発生する可能性があります。その結果、位置カウンタの期待値から最大 ± 4 カウントが変化し、予期しないそれ以降のエラー フラグ設定が発生する可能性があります。

PCRM = 0 構成を使用しているとき、[つまり、インデックス イベント時の位置カウンタリセット (QEPCTL[PCRM] = 00)]、順方向移動中にインデックス イベントが発生した場合、位置カウンタは次の eQEP クロックで 0 にリセットされます。リバース移動中にインデックス イベントが発生すると、次の eQEP クロックの QPOSMAX レジスタの値に位置カウンタがリセットされます。eQEP ペリフェラルは、QEPSTS レジスタの最初のインデックス マーカー (QEPSTS[FIMF]) と最初のインデックス イベント マーカー (QEPSTS[FIDF]) の発生を記録します。また、インデックス イベントのリセット操作に同じ相対直交遷移が使用されるように、最初のインデックス マーカーの直交エッジを記憶します。

インデックス パルスがアクティブの間に方向変更が発生した場合、モジュールは引き続き相対的な直交遷移を探して、位置カウンタのリセットを実行します。これにより、位置カウンタ値が予期せず変化します。

同時に方向を変更せずに次のインデックス イベントが発生すると、カウンタが正しくリセットされ、期待どおりに動作します。

回避方法

インデックスがアクティブなときに方向が変更される可能性があり、結果として位置カウンタ値が変更されるとアプリケーションに影響を与える可能性がある場合は、PCRM=0 構成を使用しないでください。

アプリケーションに応じて位置カウンタのリセットを実行するその他のオプション (インデックス イベント初期化 (IEI) など) には、この問題はありません。

アドバイザリ

FPU: FPU から CPU へのレジスタ移動操作の前にする FPU 2p 操作

影響を受けるリビジョン 0

詳細

このアドバイザリは、マルチサイクル (2p) FPU 命令の後に FPU から CPU へのレジスタ転送が行われる場合に適用されます。FPU-to-CPU READ 命令ソースレジスタが 2p 命令宛先と同じである場合、2p 命令が完了する前に、読み出しは FPU レジスタの値となる可能性があります。これは、2p 命令がパイプラインの E3 フェーズ中に結果のデータ転送に依存するために発生します。E3 フェーズでパイプライン ストールが発生した場合、結果は読み取り命令の時間内に転送されません。

このアドバイザリの影響を受ける 2p 命令は、MPYF32、ADDF32、SUBF32、および MACF32 です。FPU レジスタの読み出しの宛先は、CPU レジスタ (ACC、P、T、XAR0 ~ XAR7) で無ければいけません。レジスタの読み出しが FPU-to-FPU レジスタ転送である場合、このアドバイザリは適用されません。

次の例では、2p 命令 MPYF32 は R6H を宛先として使用します。FPU レジスタ READ MOV32 は、同じレジスタ R6H をソースとして使用し、CPU レジスタを宛先として使用します。E3 パイプラインフェーズでストールが発生した場合、MOV32 は MPYF32 命令が完了する前に R6H の値を読み取ります。

問題の例:

```
MPYF32 R6H, R5H, R0H ; 2p FPU instruction that writes to R6H
|| MOV32 *XAR7++, R4H
F32TOUI16R R3H, R4H ; delay slot
ADDF32 R2H, R2H, R0H
|| MOV32 *--SP, R2H ; alignment cycle
MOV32 @XAR3, R6H ; FPU register read of R6H
```

図 3-4 に、パイプラインにストールがない場合の問題のパイプライン図を示します。

	Instruction	FPU pipeline-->										Comments
		F1	F2	D1	D2	R1	R2	E	W			
I1	MPYF32 R6H, R5H, R0H MOV32 *XAR7++, R4H	I1										
I2	F32TOUI16R R3H, R4H	I2	I1									
I3	ADDF32 R3H, R2H, R0H MOV32 *--SP, R2H	I3	I2	I1								
I4	MOV32 @XAR3, R6H	I4	I3	I2	I1							
			I4	I3	I2	I1						
				I4	I3	I2	I1					
					I4	I3	I2	I1				
						I4	I3	I2	I1			
							I4	I3	I2	I1		I4 samples the result as it enters the R2 phase. The product R6H=R5H*R0H (I1) finishes computing in the E3 phase, but is forwarded as an operand to I4. This makes I4 appear to be a 2p instruction, but I4 actually takes 3p cycles to compute.
								I4	I3	I2		
									I4	I3		

図 3-4. パイプラインにストールがない場合の問題のパイプライン図

アドバイザリ (続き)

FPU: FPU から CPU へのレジスタ移動操作の前にする FPU 2p 操作

図 3-5 に、命令 I1 の E3 スロットにストールがある場合の問題のパイプライン図を示します。

	Instruction	F1	F2	D1	D2	R1	R2	E	W		Comments
		FPU pipeline-->				R1	R2	E1	E2	E3	
I1	MPYF32 R6H, R5H, R0H MOV32 *XAR7++, R4H	I1									
I2	F32TOUI16R R3H, R4H	I2	I1								
I3	ADDF32 R3H, R2H, R0H MOV32 *--SP, R2H	I3	I2	I1							
I4	MOV32 @XAR3, R6H	I4	I3	I2	I1						
			I4	I3	I2	I1					
				I4	I3	I2	I1				
					I4	I3	I2	I1			
						I4	I3	I2	I1		
							I4	I3	I2	I1 (STALL)	I4 samples the result as it enters the R2 phase, but I1 is stalled in E3 and is unable to forward the product of R5H*R0H to I4 (R6H does not have the product yet due to a design bug). So, I4 reads the old value of R6H.
							I4	I3	I2	I1	There is no change in the pipeline as it was stalled in the previous cycle. I4 had already sampled the old value of R6H in the previous cycle.
							I4	I3	I2		Stall over

図 3-5. 命令 I1 の E3 スロットにストールがある場合の問題のパイプライン図

回避方法

このシナリオでは、MPYF32、ADDF32、SUBF32、および MACF32 を 3p サイクル命令として扱います。命令のディレイ スロットには、3 つの NOP 命令または矛盾しない命令を配置する必要があります。

C28x コード生成ツール v.6.2.0 以降のバージョンでは、正しい命令シーケンスが生成され、アセンブリ コードのエラーが検出されます。以前のバージョンの v6.0.5 (6.0.x 分岐の場合) および v6.1.2 (6.1.x 分岐の場合) では、コンパイラは正しい命令シーケンスを生成しますが、アセンブラはアセンブリ コードのエラーを検出しません。

回避方法の例:

```

MPYF32 R6H, R5H, R0H
|| MOV32 *XAR7++, R4H      ; 3p FPU instruction that writes to R6H
F32TOUI16R R3H, R4H      ; delay slot
ADDF32 R2H, R2H, R0H
|| MOV32 *--SP, R2H      ; delay slot
NOP                       ; alignment cycle
MOV32 @XAR3, R6H         ; FPU register read of R6H

```

図 3-6 に、回避方法が適用されたパイプライン図を示します。

アドバイザリ (続き)

FPU: FPU から CPU へのレジスタ移動操作の前にする FPU 2p 操作

	Instruction	F1	F2	D1	D2	R1	R2	E	W		Comments
		FPU pipeline-->				R1	R2	E1	E2	E3	
I1	MPYF32 R6H, R5H, R0H MOV32 *XAR7++, R4H	I1									
I2	F32TOUI16R R3H, R4H	I2	I1								
I3	ADDF32 R3H, R2H, R0H MOV32 *--SP, R2H	I3	I2	I1							
I4	NOP	I4	I3	I2	I1						
I5	MOV32 @XAR3, R6H	I5	I4	I3	I2	I1					
			I5	I4	I3	I2	I1				
				I5	I4	I3	I2	I1			
					I5	I4	I3	I2	I1		
						I5	I4	I3	I2	I1 (STALL)	Due to one extra NOP, I5 does not reach R2 when I1 enters E3; thus, forwarding is not needed.
						I5	I4	I3	I2	I1	There is no change due to the stall in the previous cycle.
							I5	I4	I3	I2	I1 moves out of E3 and I5 moves to R2. R6H has the result of R5H*R0H and is read by I5. There is no need to forward the result in this case.
								I5	I4	I3	

図 3-6. 回避方法が適用されたパイプライン図

アドバイザー

I2C: ターゲット トランスミッタ モード、標準モードの SDA タイミング制限

影響を受けるリビジョン 0

詳細

MCU に搭載された I2C 周辺装置はファースト モード デバイスであり、スタンダードモードのホストで使用する場合、SCL (クロック) ラインをクロック ストレッチします。

スタンダード モード システムで使用されるファースト モード デバイスについては、I2C 仕様に SCL ラインを解放する前に、 $t_{\text{SU:DAT}}$ (データセット アップ時間) + $t_{\text{r(max)}}$ (立ち上がり時間) を満たすという要件があります。NXP セミコンダクタ I²C バス仕様およびユーザー マニュアル (UM10204) の「標準、高速、および高速モード プラス I²C バス デバイスの SDA および SCL バスラインの特性」表の脚注 4 を参照してください。

ただし、C2000 I2C クロックは、上記のシナリオでは、SCL ラインを $6 * f_{\text{mod}}$ クロック (C2000 の I2C クロック レート) の固定値だけストレッチします。C2000™ マイクロ コントローラがスタンダードモードのホストとともにターゲット トランスミッタとして動作しているとき、SDA の t_{r} が長すぎると、データ (SDA) の準備が完了する前に、C2000 によってクロックライン (SCL) が解放される可能性があります。

NXP セミコンダクタ I²C バス仕様とユーザー マニュアル (UM10204) の「プルアップ抵抗のサイズ設定」セクションでは、式 1 に示されている立ち上がり時間 (t_{r}) とバス容量 (C_{b}) に基づいて、適切な PU 抵抗 (R_{p}) の選択について詳細に説明しています。

$$R_{\text{p(max)}} = \frac{t_{\text{r}}}{0.8473 \times C_{\text{b}}} \quad (1)$$

回避方法

1. 強力なプルアップで t_{r} を減らす

$t_{\text{SU:DAT}} + t_{\text{r(max)}}$ が満たされることを保証するために、ユーザーは、システムの f_{mod} クロックの値に基づいて、表 3-3 の SDA データ立ち上がり時間要件列にリストされている制約を満たすように SDA ラインのプルアップ抵抗を構成できます。これにより、C2000 が SCL 信号を解放したときに、SDA ラインに存在するデータが有効になることが保証されます。

表 3-4 に、特定の f_{mod} クロック (MHz) および C_{b} (バス容量) に対して推奨される R_{p} 抵抗値を示します。 C_{b} の他の値については、式 1 を使用して、システムに必要な R_{p} の値を計算してください。

表 3-3. 標準モードホストを備えたターゲット トランスミッタとしての C2000 のデータ立ち上がり時間要件

f_{mod} クロック (MHz)	f_{mod} 期間 (ns)	C2000 I2C からの SCL クロック ストレッチ遅延 (ns): ($6 * f_{\text{mod}}$ クロック)	データ セットアップ時間 (ns): $t_{\text{SU:DAT}}$ (標準モード)	SDA データの立ち上がり時間の要件 (ns): t_{r}
7	142.9	857	250	607
8	125	750		500
9	111	666		416
10	100	600		350
11	90.9	545		295
12	83.3	500		250

アドバイザリ (続き)

I2C: ターゲット トランスミッタ モード、標準モードの SDA タイミング制限

表 3-4. 一般的なバス容量 (C_b) のプルアップ抵抗 (R_p) 値

f_{mod} クロック (MHz)	SDA データの立ち上がり 時間の要件 (ns): t_r	R_p (k Ω) $C_b = 100\text{pF}$ の 場合	R_p (k Ω) $C_b = 200\text{pF}$ の 場合	R_p (k Ω) $C_b = 300\text{pF}$ の 場合	R_p (k Ω) $C_b = 400\text{pF}$ の 場合
7	607	7.1	3.5	2.3	1.7
8	500	5.9	2.9	1.9	1.4
9	416	4.9	2.4	1.6	1.2
10	350	4.1	2.0	1.3	1.0
11	295	3.4	1.7	1.1	0.8
12	250	2.9	1.4	0.9	0.7

2. $t_r = 1000\text{ns}$

この回避方法は、一般的な I2C の使用上の制限により推奨されません。可能な場合は回避方法 1 を使用してください。

システムで SDA ラインの立ち上がり時間が 1000ns 必要な場合は、C2000 I2C f_{mod} クロックを 4.8MHz に設定して、クロックストレッチ ($6 * f_{\text{mod}}$ クロック) によってこの要件を満たすことができます。その結果、 $t_r = (1/4.8\text{MHz}) * 6 = 1000\text{ns}$ となります。この回避方法は、C2000 I2C が I2C バス上のターゲットとなるシステムでのみ有効です。4.8MHz は、 f_{mod} クロックに 7Mhz から 12Mhz へデータシートに必要な範囲外であることに注意してください。 f_{mod} を 4.8Mhz で使用すると、データシートの必要な範囲外であっても、C2000 I2C に対してスタンダード モードのホストバス上でターゲット モードで動作します。この回避方法に記載されているものを除く他の構成で $f_{\text{mod}} = 4.8\text{Mhz}$ を使用すると、他のタイミングパラメータに違反し、許可されません。

アドバイザリ

LIN:同期フィールドが誤っている場合、不整合同期フィールド エラー (ISFE) フラグ/割り込みが設定されません

影響を受けるリビジョン 0**詳細**

LIN 通信中、(RX で)受信した同期フィールドが誤っている場合(つまり、同期フィールドが 0x55 以外の値を受信した場合)、LIN は SCIFLR.ISFE レジスタの ISFE フラグを設定したり、ISFE 割り込みをトリガしません。データを受信したり、RX 受信割り込みが設定されたりせずに、通信が終了します。アプリケーションが同期フィールドのエラーを検出する方法はありません。アプリケーションは、同期フィールドが完全にブランクであるか、または同期フィールドが指定された許容範囲内に受信されていないかどうかを検出できます(『[TMS320F28003x リアルタイム マイコン テクニカル リファレンス マニュアル](#)』を参照)。

回避方法

方法 1: SCIFLR.RXRDY フラグをポーリングし続け、一定時間内に設定されていない場合はタイムアウトします。

ガイドラインとして、以下の手順に従ってください。

1. 設定する SCIFLR.BUSY フラグをポーリングします。
2. BUSY フラグが High になったら、SCIFLR.RXRDY フラグをポーリングします。このループ内に同時に SW タイムアウトがあります。このタイムアウトは、ユーザー定義の時間間隔内に RXRDY フラグが設定されていない場合、ループをタイムアウトし、ループを終了します。

方法 2: RX 割り込みがトリガされていない場合に割り込みを行うように CPU タイマを構成します。この方法では、CPU 帯域幅は使用されません。

ガイドラインとして、以下の手順に従ってください。

1. LINRX が High から Low になった (LIN がビジーであることを示す)ときに ISR をトリガするように XINT を構成します。
2. XINT ISR 内で CPU タイマを構成し、フレーム完了のタイミングを開始します。
3. 正しい同期フィールドを使用してフレームを正しく受信すると、LIN RX ISR がトリガされます。内部でタイマをオフにすることにより、誤ったタイムアウトが発生しないようにできます。
4. フレームが正しく受信されない場合、LIN RX ISR はトリガされず、CPU タイマ ISR (タイムアウトが発生)がトリガされます。これは、同期フィールドのエラーを示します。

アドバイザリ メモリ: 有効なメモリを超えたプリフェッチ

影響を受けるリビジョン 0

詳細

C28x CPU は、パイプラインで現在アクティブな命令を超える命令をプリフェッチします。プリフェッチが有効なメモリの終了後に発生した場合、CPU は無効なオペコードを受信する可能性があります。

回避方法

M1、GS3 –プリフェッチ キューは 8×16 ワードの深さです。したがって、コードは 有効なメモリの終わりから 8 ワード以内にはなりません。2 つの有効なメモリ ブロック間の境界を越えてプリフェッチすることは問題ありません。

事例 1: M1 はアドレス 0x7FF で終了し、その後に別のメモリ ブロックが続くことはありません。M1 のコードは、アドレス 0x7F7 以下に格納されないようにする必要があります。アドレス 0x7F8 ~ 0x7FF はコードには使用できません。

事例 2: M0 はアドレス 0x3FF で終了し、有効なメモ (M1) がその後に続きます。M0 のコードは、アドレス 0x3FF まで保存できます。コードはアドレス 0x7F7 までの M1 にクロスすることもできます。

フラッシュ -プリフェッチ キューは 16×16 ワードの深さです。したがって、コードは 有効なメモリの終わりから 16 ワード以内にはなりません。そうでない場合、フラッシュ ECC の修正不可能なエラーが発生します。

表 3-5. アドバイザリによって影響を受けるメモリ

メモリ タイプ	影響を受けるアドレス
M1	0x0000 07F8–0x0000 07FF
GS3	0x0000 FFF8–0x0000 FFFF
フラッシュ	0x000A FFF0–0x000A FFFF

アドバイザリ **ブート ROM:TMX デバイス上の MCAN ブートローダーによるデータ オーバーラン**

影響を受けるリビジョン 0 (TMX のみ)**詳細**

TMX デバイスでは、MCAN ブートローダーを使用すると、F28003x 側で通信データのオーバーランが発生し、ブートローダーの障害につながります。

回避方法

ホストは、通信が成功するように、連続するフレーム間の 400µs の遅延を確保する必要があります。この問題は TMS デバイスで修正されています。

アドバイザリ

セキュア ライブ ファームウェア アップデート (LFU) ブート モードは非推奨です

影響を受けるリビジョン 0

詳細

セキュア LFU ブート モードが設定されている場合、メモリが保護されている場合にデバイスがリセットされます。

回避方法

最新のファームウェア バンク選択アルゴリズムは、ROM から呼び出す代わりに、フラッシュ カスタム ブートローダーの一部として実装できます。この同じカスタム ブートローダーは、最新のファームウェア バンクを選択した後、フラッシュ コンテンツ認証のために、(セキュア ROM に格納されている) CMAC 認証アルゴリズムを呼び出せます。このカスタム ブートローダーが配置されているフラッシュ メモリは、Z1 セキュア メモリとして構成する必要があります。

アドバイザリ フラッシュ:フラッシュのプリフェッチを無効にせずに **Fapi_setActiveFlashBank()** を実行すると、**ITRAP** が発生する

影響を受けるリビジョン 0

詳細

フラッシュのプリフェッチ メカニズムは、RAM から実行されるコードによって、フラッシュへのアクティブなフェッチ アクセスがない場合にのみ無効にする必要があります。
Fapi_setActiveFlashBank() により、プリフェッチ メカニズムが無効になります。プリフェッチの無効化処理中にフラッシュからコードを実行すると、**ITRAP** が発生する可能性があります。そのため、**Fapi_setActiveFlashBank()** を実行する前にプリフェッチを無効にする必要があります。

回避方法

Fapi_setActiveFlashBank() を呼び出す前に、プリフェッチ メカニズムを無効にします。プリフェッチは、**Fapi_setActiveFlashBank()** の実行後に有効にできます。プリフェッチ メカニズムをディセーブルおよびイネーブルにするコードは、RAM から実行する必要があります。この回避方法を実装する場合は、プリフェッチがすでに無効になっているため、**Fapi_setActiveFlashBank()** をフラッシュから実行できることに注意してください (消去 / プログラム操作の対象となるバンクからは実行しないでください)。

C2000Ware_5_02_00_00 以降で提供されている Flash API の使用例には、この回避方法の実装方法が示されています。

TMS320F28003x サンプルのパス:

C2000Ware_5_02_00_00\driverlib\f28003x\examples\flash\flashapi_ex1_programming.c

この例では、**Fapi_setActiveFlashBank()** を呼び出す前に、
Flash_DisablePrefetch_SW_Workaround() を実行してプリフェッチを無効にします。
Fapi_setActiveFlashBank() の呼び出し後にプリフェッチを有効にするため、
Flash_EnablePrefetch_SW_Workaround() を RAM から実行します。

アドバイザリ

システム: **Pagesel=0** モードでは、**HIC** 不正読み取りエラー フラグがアサートされません

影響を受けるリビジョン 0

詳細

保留中の書き込み場所と同じアドレス (不正なアクセス シーケンス) をホスト読み取りアクセスが開始された場合、**Pagesel=0** モードでは不正読み出しエラー フラグはアサートされません。エラーフラグは、**Pagesel=1** モードで同じシーケンスに設定されます。これは不正なシーケンスであり、通常のアプリケーションフローにおいてソフトウェアが書き込み保留中の領域に対する読み出しを開始することは想定されていないため、影響は軽微です。

回避方法

なし

アドバイザリ

システム: **CLKSRCCTL1** への複数の連続した書き込みを行うと、システムがハングする可能性があります

影響を受けるリビジョン 0

詳細

書き込み間に遅延なしで **CLKSRCCTL1** レジスタに複数回書き込むと、システムがハングアップし、外部 **XRSn** リセットまたはウォッチドッグリセットでのみ回復できます。この条件の発生は、**SYSCLK** と **OSCCLKSRCSEL** で選択したクロックとの間のクロック比によって異なり、毎回発生するとは限りません。

デバッガの使用中にこの問題が発生した場合、一時停止すると、プログラム カウンタはブート ROM リセット ベクタになります。

回避方法を実装すると、**SYSCLK/OSCCLK** の比率でこの条件を回避できます。

回避方法

CLKSRCCTL1 レジスタへの書き込みごとに、**NOP** 命令を使用して、300 **SYSCLK** サイクルのソフトウェア遅延を追加します。

例:

```
ClkCfgRegs.CLKSRCTL1.bit.INTOSC2OFF=0;           // Turn on INTOSC2
asm(" RPT #250 || NOP");                          // Delay of 250 SYSCLK cycles
asm(" RPT #50 || NOP");                            // Delay of 50 SYSCLK cycles
ClkCfgRegs.CLKSRCTL1.bit.OSCCLKSRCSEL = 0;         // Clk Src = INTOSC2
asm(" RPT #250 || NOP");                          // Delay of 250 SYSCLK cycles
asm(" RPT #50 || NOP");                            // Delay of 50 SYSCLK cycles
```

C2000Ware_3_00_00_00 以降のリビジョンでこの回避策が実装されます。

アドバイザー HWBIST 使用中にスプリアス割り込みを回避する

影響を受けるリビジョン 0

詳細

HWBIST には、CPU のテスト中に受信した割り込みをログに記録し、HWBIST の完了後にそれらを再発行する機能があります。割り込みログ記録が有効化される直前のクロック サイクルで受信された割り込みは、HWBIST が実行される前に実行されます。次のサイクルで割り込みログ記録が有効化されると、割り込みがログに記録され、HWBIST が完了した後に再発行されます。

CPU タイマ 1 および CPU タイマ 2 の割り込みイベントは、2 SYSCLK サイクルの間有効です。最初のサイクルが割り込みログ記録の有効化の 1 サイクル前に発生し、2 番目のサイクルが割り込みログ記録の有効化と重なった場合、割り込みはログ記録の前に 1 回実行されますが (CPU タイマの TCR.TIF フラグをクリア)、その後割り込みロガーによってログに記録され、HWBIST の完了後に再度トリガされます。TCR.TIF フラグは以前の ISR によってすでにクリアされているため、これは予期しないスプリアス割り込みとなります。

なお、このシナリオは非 PIE の CPU タイマ割り込みにのみ適用されることに注意してください。CPU タイマ 0 の割り込みは PIE によって管理され、そのパルス幅はわずか 1 SYSCLK サイクルです。

回避方法

割り込みログを有効にする前に CPU タイマ 1 および 2 の割り込みを無効にし、後でそれらを復元します。この回避策は、`stl_hwbist_s.asm` ファイルの C2000 ソフトウェア診断ライブラリにすでに実装されています。手順は次のとおりです。

1. CPU タイマ 1 および 2 のタイマ割り込みイネーブル ビット TCR.TIE をクリアします。
2. 通常の HWBIST シーケンス (レジスタの保存、割り込みログの有効化、HWBIST の実行、レジスタの復元、割り込みログの終了) を実行します。
3. CPU タイマの TCR.TIF フラグがセットされているかどうかを確認します。フラグがセットされている場合、対応する CPU IFR ビットを設定して割り込みをトリガします。
4. TCR.TIE を復元します。

アドバイザリ **PLL リファレンス クロック消失検出: クロック消失フラグが正しくアクティブ化されない可能性があります**

影響を受けるリビジョン 0

詳細

レジスタ ビット `SYSPLLSTS.REF_LOSTS` の設定が誤っており、リファレンス クロックの損失を示す可能性があります。

回避方法

PLL リファレンス クロック消失検出機能は使用しないでください。代わりに、欠落クロック検出 (MCD) 機能またはデュアル クロック コンパレータ (DCC) を使用して、リファレンス クロックの損失を検出します。

- MCD 方式については、`sysctl` フォルダにある C2000Ware サンプルの `sysctl_ex1_missing_clock_detection` を参照してください。
- DCC 方式については、`dcc` フォルダにある C2000Ware サンプルの `dcc_ex4_clock_fail_detect` を参照してください。

アドバイザリ **SDFM:スレッショルド設定 (LLT、HLT)、フィルタタイプ、COSR 設定を動的に変更すると、スプリアス コンパレータ イベントがトリガされます**

影響を受けるリビジョン 0

詳細

SDFM コンパレータ設定 (フィルタタイプ、下限/上限スレッショルド、コンパレータ OSR (COSR) 設定など) が実行中に動的に変更されると、誤ったコンパレータ イベントがトリガされてしまいます。スプリアス コンパレータ イベントは、適切に設定されている場合、対応する CPU 割り込み、CLA タスク、ePWM クロスバー イベント、GPIO 出力クロスバー イベントをトリガします。

回避方法

コンパレータ設定を動的に変更する必要がある場合は、以下の手順に従って、誤ったコンパレータ イベントによって CPU 割り込み、CLA タスク、または X-BAR イベント (ePWM X-BAR/GPIO 出力 X-BAR イベント) が生成されないようにしてください。

1. コンパレータ フィルタを無効にします。
2. コンパレータ フィルタの少なくともレイテンシ + 3 SD-Cx クロックサイクルの遅延。
3. フィルタタイプ、COSR、下限/上限スレッショルドなどのコンパレータ フィルタ設定を変更します。
4. コンパレータ フィルタの少なくともレイテンシ + 5 SD-Cx クロックサイクルの遅延。
5. コンパレータ フィルタを有効にします。

アドバイザリ **SDFM:データフィルタ設定 (フィルタタイプや DOSR など) を動的に変更すると、誤ったデータ確認イベントがトリガされます**

影響を受けるリビジョン 0

詳細

実行時に SDFM データ設定 (フィルタタイプや DOSR 設定など) が動的に変更されると、誤ったデータ フィルタ準備完了イベントがトリガされます。スプリアス データ準備完了イベントは、適切に構成されている場合、対応する CPU 割り込み、CLA タスク、DMA トリガをトリガします。

回避方法

SDFM データ フィルタ設定を動的に変更する必要がある場合は、以下の手順に従って、スプリアス データ フィルタ準備完了イベントが生成されないようにします。

1. データフィルタを無効にします。
2. データフィルタの少なくともレイテンシー + 3 SD-Cx クロック サイクルの遅延。
3. フィルタタイプや DOSR などのデータ フィルタ設定を変更します。
4. データフィルタの少なくともレイテンシー + 5 SD-Cx クロック サイクルの遅延。
5. データ フィルタを有効にします。

アドバイザリ

SDFM:SD 変調器の 3 クロック サイクル以内に SDCPARMx レジスタのビット フィールド CEVT1SEL、CEVT2SEL、および HZEN に連続して 2 回書き込みを行うと、SDFM ステートマシンが破損し、誤ったコンパレータ イベントが発生する可能性があります

影響を受けるリビジョン 0

詳細

3 つの SD モジュレータ クロック サイクル内で SDCPARMx レジスタ ビット フィールド CEVT1SEL、CEVT2SEL、および HZEN に連続書き込みをすると、SDFM ステート マシンが破損する可能性があります、誤ったコンパレータ イベントが発生する可能性があります。これにより、適切に構成されている場合は、CPU 割り込み、CLA タスク、ePWM XBAR イベント、および GPIO 出力 X-BAR イベントがトリガされる可能性があります。

回避方法

3 つの SD モジュレータ クロック サイクル内での連続書き込みを避けるか、SDCPARMx レジスタ ビット フィールドを 1 回のレジスタ書き込みで構成します。

アドバイザリ ウォッチドッグ:WDKEY レジスタは EALLOW 保護されていません

影響を受けるリビジョン 0

詳細

WDKEY レジスタは EALLOW 保護されていません。このレジスタに書き込むために EALLOW および EDIS 命令を発行する必要はありません。WDKEY が EALLOW 保護されている他のデバイスでソフトウェアの再利用をイネーブルにするには、EALLOW および EDIS を推奨します。

回避方法

なし

4 ドキュメントのサポート

デバイス固有のデータシートおよび関連ドキュメントについては、TI の Web サイト <https://www.ti.com> をご覧ください。

TMS320F28003x デバイスの詳細については、以下のドキュメントを参照してください。

- 『[TMS320F28003x リアルタイム マイクロ コントローラ](#)』データシート
- 『[TMS320F28003x リアルタイム マイクロコントローラ テクニカル リファレンス マニュアル](#)』

5 商標

C2000™ is a trademark of Texas Instruments.

すべての商標は、それぞれの所有者に帰属します。

6 改訂履歴

Changes from May 30, 2024 to August 31, 2026 (from Revision D (May 2024) to Revision E (August 2026))

	Page
• セクション 3.1.2 使用上の注意: タイトルを「ネストされた割り込みを使用する際の注意」から「繰り返しブロックでネストされた割り込みを使用する際の注意」に変更しました。使用上の注意の説明を更新。.....	8
• 新しい I2C スレーブ モードの使用上の注意を追加: セクション 3.1.3	8
• アドバイザリ C28: 高速整数除算 (FID) と DIVF64 の不正確な結果.....	15
• 追加のアドバイザリ - ePWM: ePWM ワンショット / CBC トリップ イベント DCxEVTy.force はトリップ条件を設定しません.....	23

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みいかなる保証も、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、TI は一切の責任を拒否します。

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](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。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。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月