参考資料

Errata

F28E12x Real-Time MCUs Silicon Errata

シリコン リビジョン 0

TEXAS INSTRUMENTS

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

図の一覧

表の一覧

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 3.1.3	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	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
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
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
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
フラッシュ	フラッシュ：シングル ビット ECC エラー割り込みは生成されません	あり
FPU	FPU：FPU-to-CPU レジスタ移動操作の前にする FPU 2p 操作	あり
GPIO	GPIO：オープン ドレイン構成による短い High パルスを駆動する可能性	あり
MCD	MCD：PLL が有効 (PLLCLKEN = 1) のとき、クロック消失検出を無効化	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
PLL	PLL：PLL が初回ロック試行で正常にロックしない可能性	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
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
ウォッチドッグ	ウォッチドッグ：WDHALTI ビットへの書き込みが PLL ロック ステータスに影響	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

テキサス・インスツルメンツでは、サポート ツールについては、使用可能な 3 つの接頭辞のうち TMDX および 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 (TMDX) から、完全認定済みの量産ツール (TMDS) まであります。

デバイスの開発進展フロー:

X 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。

P 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。

空白 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・インスツ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。

X および P 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・インスツ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きいと予測されます。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・インスツルメンツでは、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。

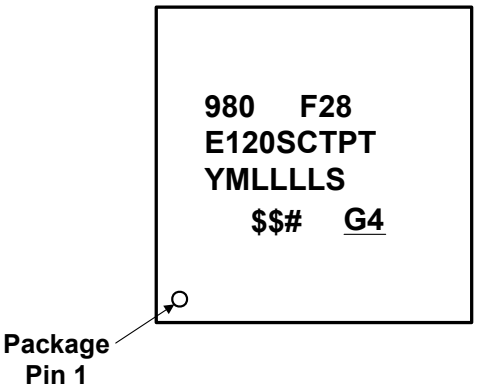
2.2 サポート対象デバイス

本文書は、以下のデバイスをサポートしています。

- F28E120SC
- F28E120SB

2.3 パッケージの記号表記およびリビジョンの識別

図 2-1、図 2-2 および図 2-3 にパッケージの記号表記を示します。表 2-1 に、シリコンのリビジョン コードを示します。



YMLLLLS = Lot Trace Code

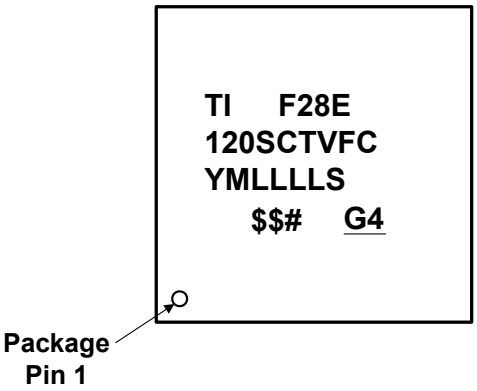
980 = TI EIA Code
YM = 2-digit Year/Month Code
LLLL = Assembly Lot Code
S = Assembly Site Code
$$ = Wafer Fab Code (one or two characters) as applicable
 # = Silicon Revision Code

G4 = Green (Low Halogen and RoHS-compliant)

注

YM =「5A」の F28E120SCTPT ユニットにおいて、シリコンのリビジョン コードが誤って「A」と印字されていました。これらのユニットにおけるシリコンのリビジョン コードは 0 です (# は空欄)。

図 2-1. PT パッケージのパッケージ マーキング

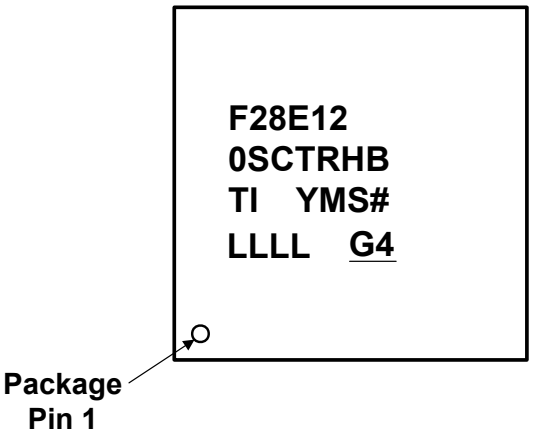


YMLLLLS = Lot Trace Code

YM = 2-digit Year/Month Code
LLLL = Assembly Lot Code
S = Assembly Site Code
$$ = Wafer Fab Code (one or two characters) as applicable
 # = Silicon Revision Code

G4 = Green (Low Halogen and RoHS-compliant)

図 2-2. VFC パッケージのパッケージ マーキング



YM = 2-digit Year/Month Code
S = Assembly Site Code
 # = Silicon Revision Code
LLLL = Assembly Lot Code

G4 = Green (Low Halogen and RoHS-compliant)

図 2-3. RHB 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:0x5D006	備考(2)
空白	0	0x0000 0001	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、*F28E12x 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リ

このセクションには、このシリコン リビジョンの使用上の注意およびアドバイザリが記載されています。

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 ツールの以前のリビジョンでのみ追加する必要があります。

3.1.3 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 (デバッガ接続など) 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 のアドバイザリ

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リを示しています。

アドバイザリ	***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
}
```

アドバイザリ ***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。

F28E12x 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、F28E12x 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の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：*F28E12x 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 µs 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ョルドを超えないようにできます。

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。

LATENCY = 1 + (1 x FILTER_PRESCALE) + (FILTER_THRESH x 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。

アドバイザリ ***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 ピンで使用される機能	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。

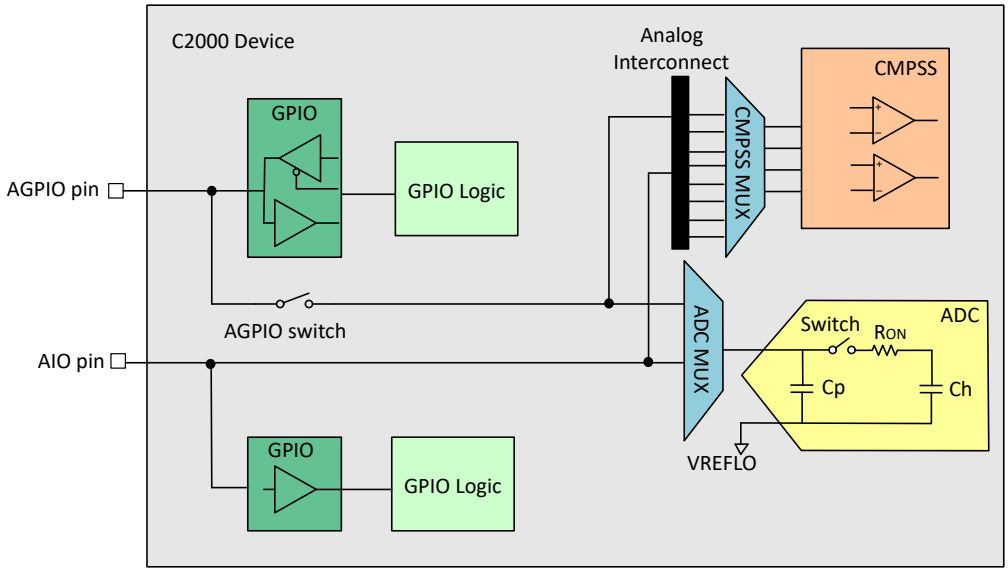


図 3-1. AGPIO と AIO アナログ ピン タイプを含むアナログ サブシステム図

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、影響を受けるチャネルが読み取られる直前に、ADC の別のチャネルから 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、誤トリッ

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**

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 ***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	フラッシュ : シングル ビット ECC エラー割り込みは生成されません
影響を受けるリビジョン	0

詳細

シングル ビット ECC エラー スレッショルドが 0 に設定されている場合、シングル ビット エラーが発生しても、シングル ビット エラー割り込みは生成されません。

回避方法

エラー スレッショルド ビット フィールド (FLASH_ECC_REGS ERR_THRESHOLD.ERR_THRESHOLD フィールド) を 1 以上の値に設定します。スレッショルド ビット フィールドのデフォルト値は 0 であることに注意してください。

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 3-2 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	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**	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 3-2. 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 3-3 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 3-3. 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 v.6.1.2 (6.1.x 分岐の場) 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 3-4 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 3-4. 回避方法が適用されたパイプライン図

アドバイザリ ***GPIO：オープンドレイン構成による短い High パルスを駆動する可能性***

影響を受けるリビジョン 0

詳細

各 GPIO は、GPxODR レジスタを使用してオープン ドレイン モードに設定できます。ただし、内部デバイスのタイミングの問題により、GPIO が高インピーダンス状態への遷移中、または高インピーダンス状態からの遷移中に最大 0 ～ 10ns の間、ロジック High を駆動することがあります。

この望ましくない High レベルにより、他のドライバが同時に低レベルを駆動している場合、GPIO がライン上の別のオープンドレイン ドライバと競合する可能性があります。この競合は、両方のデバイスにストレスを加え、信号に短時間の中間電圧レベルが生じるため、望ましくありません。レシーバ ロジックにこの短いパルスをフィルタリングするのに十分なロジック フィルタリングがない場合、この中間電圧レベルは誤って高レベルと解釈される可能性があります。

回避方法

競合が問題となる場合、GPIO のオープンドレイン機能を使用せず、ソフトウェアでオープン ドレイン モードをエミュレートします。オープン ドレイン エミュレーションは、GPIO データ (GPxDAT) を静的 0 に設定し、GPIO 方向ビット (GPxDIR) をトグルして、駆動を Low にして有効 / 無効にすることにより実現できます。実装例については、以下のコードを参照してください。

```
void main(void)
{ ...

  // GPIO configuration
     EALLOW;                                 // disable pullup
     GpioCtrlRegs.GPxPUD.bit.GPIOx = 1;      // disable open-drain mode
     GpioCtrlRegs.GPxODR.bit.GPIOx = 0;      // set GPIO to drive static 0 before
                                             // enabling output
     GpioDataRegs.GPxCLEAR.bit.GPIOx = 1;
     EDIS;
  ...

  // application code
  ...

  // To drive 0, set GPIO direction as output
     GpioCtrlRegs.GPxDIR.bit.GPIOx = 1;

  // To tri-state the GPIO(logic 1),set GPIO as input
     GpioCtrlRegs.GPxDIR.bit.GPIOx = 0;
}
```

アドバイザリ ***MCD:PLL が有効 (PLLCLKEN = 1) のとき、クロック消失検出を無効化***

影響を受けるリビジョン 0

詳細

PLL には、入力 OSCCLK が存在しない場合でも低速の PLLRAWCLK 出力を提供するリンプ モード機能があります。独立して、Missing Clock Detect (MCD) 回路は、OSCCLK 入力の欠落が検出されると、システム クロック ソースを強制的に INTOSC1 に切り替えます。これらのシステム クロック ソースを切り替える MCD マルチプレクサは、両方のクロック ソース (PLLRAWCLK と INTOSC1) がアクティブな場合、グリッチがないことが保証されません。まれに、クロックの欠落障害イベント中に予期しないデバイス動作が発生する可能性があります。

回避方法

システムで PLL が使用されている場合 (PLLCLKEN = 1)、MCDCR.MCLKOFF = 1 を書き込んで MCD を無効にします。

デュアル クロック コンパレータ (DCC) 回路は、クロック イベントの欠落により SYSCLK 周波数が目的の周波数から外れてリンプ モードに低下したかどうかを迅速に検出するように構成できます。

システムが PLL バイパス モード (PLLCLKEN = 0) で動作している場合でも、MCD 回路を使用して欠落したクロック イベントを検出し、クロック ソースを INTOSC1 に切り替えることができます。

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 – 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。

表 3-3. 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

アドバイザリ ***PLL :PLL が初回ロック試行で正常にロックしない可能性***

影響を受けるリビジョン 0

詳細

PLL が最初のロック試行で正しくロックしない場合があります。PLLSTS[LOCKS] ビットはセットされますが、PLL は予期しないランダムな周波数でロックします。PLL を無効化して再度有効化した際に、このロック不良が再発する可能性があります。

SYSPLL が正しくロックしていない状態で CPU クロック ソースとして選択されている場合、あいまいな周波数にロックされているため、CPU が予期しない動作を示します。

この一過性の問題の発生率は低い。PLL の無効化および再有効化を行うシステムで発生する可能性があります。回避方法を実装することで、PLL が正しい周波数でロックするまで再試行を行うようにできます。

回避方法

TI は、PLL がロック状態に入り、デュアルクロックコンパレータ (DCC) によって意図した周波数で動作していることが検証されるまで、PLL ロック シーケンスを連続して再試行することを推奨しています。

ロック シーケンスは、まず PLL を無効化し、その後に再起動して LOCKS ビットがセットされるまで待機し、続いて DCC を使用して PLL の周波数を検証するという手順で構成されます。PLL が正しい周波数で動作していることが確認された後に、CPU クロック ソースとして選択することができます。

TI は、C2000Ware v6.00.01 以降に含まれる SysCtl_setClock() 関数の使用を推奨しており、この関数には、PLL クロックをユーザー定義の回数まで再試行して設定する回避方法の実装も含まれています。

アドバイザリ (続き)　**PLL :PLL が初回ロック試行で正常にロックしない可能性**

DCC の使用方法の詳細については、C2000Ware SysCtl_IsPLLValid () 関数を参照してください。また、この回避方法はシステムレベルでも適用可能であり、デバイスが正常に動作していない場合には、スーパーバイザがデバイスをリセットすることで対処することもできます。、適用することもできます。

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.CLKSRCCTL1.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.CLKSRCCTL1.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

アドバイザリ	***ウォッチドッグ:WDHALTI ビットへの書き込みが PLL ロック ステータスに影響***
影響を受けるリビジョン	0
詳細	システム PLL がロックされたあと (SYSPLLSTS.LOCKS = 1)、CLKSRCCTL1.WDHALTI に書き込むと、PLL ロックが解除されます (SYSPLLSTS.LOCKS = 0)。
回避方法	PLL をロックする前に、WDHALTI の構成を行います。

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。

F28E12x 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。

- 『F28E12x リアルタイム マイクロコントローラ』データシート
- 『F28E12x 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

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 JULY 28, 2025 to OCTOBER 31, 2025 (from Revision * (July 2025) to Revision 0 (October 2025)) **Page**

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。

最終更新日：2025 年 10 月