ANALOG DEVICES

Blackfin® 組込みプロセッサ

シリコン・アノーマリ・リスト　ADSP-BF538/BF538F

ADSP-BF538/BF538F シリコン・アノーマリについて

これらのアノーマリには、Blackfin ADSP-BF538/BF538F 製品においてすでに知られているレビジョン間の相違、および ADSP-BF538/BF538F データシートとハードウェア・リファレンス・ブックで規定される機能に対する相違を示します。

シリコン・レビジョン

シリコン・レビジョン番号は"-x.x"の形式で、すべてのデバイスに表示してあります。レビジョンを識別するために、DSPID コア MMR レジスタのインプリメンテーション・フィールド・ビット<15:0>を下図のように使うことができます。

Silicon REVISION	DSPID<15:0>
0.5	0x0005
0.4	0x0004
0.3	0x0003

アノーマリ・リストのレビジョン履歴

次のレビジョン履歴には、アノーマリ・リストのレビジョンとアノーマリ・リストの各レビジョンでの主要な変更を記載します。

Date	Anomaly List Revision	Data Sheet Revision	Additions and Changes
07/10/2009	H	B	Added Anomalies: 05000443、05000461、05000462
09/18/2008	G	B	Added Anomalies: 05000425、05000426、05000436 Revised Anomalies: 05000283、05000315
06/18/2008	F	B	Added Silicon Revision 0.5 Added Anomalies: 05000416
02/08/2008	E	0	Added Anomalies: 05000374、05000402、05000403
11/15/2007	D	0	Added Anomalies: 05000355、05000357、05000366、05000371、05000357
03/13/2007	C	PrD	Added Silicon Revision 0.4 Added Anomalies: 05000315、05000317
12/07/2006	B	PrD	Added Anomalies: 05000245、05000310、05000312、05000313
09/15/2006	A	PrD	Initial Revision

NR003277H

アナログ・デバイセズ株式会社

本　　社／〒105-6891　東京都港区海岸 1-16-1　ニューピア竹芝サウスタワービル
電話 03（5402）8200
大阪営業所／〒532-0003　大阪府大阪市淀川区宮原 3-5-36　新大阪トラストタワー
電話 06（6350）6868

シリコン・アノーマリの一覧

次の表に、ADSP-BF538/BF538F アノーマリの一覧と各アノーマリの適用されるシリコン・レビジョンを示します。

番号	ID	説明	0.3	0.4	0.5
1	05000074	スロット 1 で dsp32shiftimm を使用しスロット 2 で P レジスタ・ストアを行うマルチ発行命令はサポートされていません:	x	x	x
2	05000119	ペリフェラル受信チャンネル DMA が停止した後の DMA_RUN ビットが無効:	x	x	x
3	05000122	16 ビット・システム MMR レジスタをアクセスするとき Rx.H を使用できない:	x	x	x
4	05000166	8～16 の PPI データ長で上位ビットがゼロにならない:	x	x	x
5	05000179	汎用 TX または RX モードで、PPI_COUNT を 0 に設定できない:	x	x	x
6	05000180	0 フレーム同期を持つ PPI モードで PPI_DELAY が機能しない:	x	x	x
7	05000193	極性設定を変更した際のエッジ検出入力での偽 I/O ピン割り込みの発生:	x	x	x
8	05000199	キャリ・フィックス時にカレント DMA アドレスが不正な値を表示する:	x	.	.
9	05000219	ブート時の NMI イベントで予期しない状態が発生する:	x	x	x
10	05000229	SPI スレーブ・ブート・モードでレジスタがリセット値から変更される:	x	x	x
11	05000233	2 または 3 内部フレーム同期送信モードで PPI_FS3 が出力されない:	x	x	x
12	05000245	条件付き分岐のシャドウ内のアクセスで偽ハードウェア・エラーが発生する:	x	x	x
13	05000253	タイマの最大外部クロック速度:	x	x	x
14	05000270	活発な I/O 動作により、内部電圧レギュレータ(Vddint)の出力電圧が低下する:	x	.	.
15	05000272	あるデータ・キャッシュ・ライトスルー・モードが Vddint <= 0.9V で失敗する:	x	x	x
16	05000273	同期 SDRAM メモリへの書き込みが失われる:	x	.	.
17	05000277	エッジ検出の 1SCLK サイクル後の I/O データ・レジスタへの書き込みが割り込みをクリアする:	x	.	.
18	05000278	DMA 実行中にペリフェラルをディスエーブルすると DMA システムが不安定になる:	x	.	.
19	05000281	ISR コンテキストが回復されないとき偽ハードウェア・エラー例外が発生する:	x	.	.
20	05000282	32 ビット・データとトラフィック制御によるメモリ DMA の破壊:	x	.	.
21	05000283	特定のステージでシステム MMR 書き込みが停止されると、直ちにこれがストールする:	x	.	.
22	05000288	FIFO が満杯になると SPORT が不正なデータを受信する:	x	.	.
23	05000291	CAN メールボックスとアクセプタンス・マスク領域からの読み出しが失敗する:	x	.	.
24	05000293	ハイバネートリーク電流が規定値より大きい:	x	.	.
25	05000294	外部フレーム同期を使う PPI TX モードに対するタイマ・ピンの制約	x	x	x
26	05000301	メモリ―メモリ間 DMA のソース/ディステネーション・ディスクリプタは同じメモリ空間にある必要がある。:	x	.	.
27	05000304	CAN/DMA MMR レジスタへの書き込み後、SSYNC が正常に処理されないことがある:	x	.	.
28	05000307	ハイバネート中に SCKELOW ビットが状態を維持しない:	x	.	.
29	05000310	予約済みメモリの境界でのフェッチにより、偽ハードウェア・エラーが発生する:	x	x	x
30	05000312	SSYNC、CSYNC、または LT、LB、LC レジスタへのロードが割り込みを受けるとエラーが発生する:	x	x	.
31	05000313	シングル・フレーム同期モードの最初の転送で PPI がレベル検出になる:	x	.	.
32	05000315	次のシステム MMR アクセスでシステム MMR 書き込みの停止が異常完了する:	x	.	.
33	05000317	PORTFIO または PORTFIO_TOGGLE への書き込みで PFx グリッチが発生する:	x	.	.
34	05000355	ハイバネート・ウェイクアップ・ソースがアクティブのままであると、レギュレータの設定ができなくなる:	x	x	.
35	05000357	チャンネル 0 をディスエーブルすると、シリアル・ポート(SPORT)マルチチャンネル送信に不具合が発生する:	x	x	.
36	05000366	ITU-R 656 モードで PPI アンダーフロー・エラーが検出されなくなる:	x	x	x
37	05000371	サブルーチンの継続時間が 5 サイクルを下回るとき、RETS レジスタが壊れる可能性がある:	x	x	.
38	05000374	ペリフェラル・ウェイクアップをイネーブルしてハイバネート状態に入ると、大きな電流が流れる:	.	x	.
39	05000375	GPIO ピン PC1 と PC4 が通常出力として機能できる:	x	.	.
40	05000402	プロセッサが非キャッシャブル・メモリから実行されると SSYNC によりストールする:	x	.	.
41	05000403	レベル検出の外部 GPIO ウェイクアップにより無限ストールが発生する:	x	x	x

番号	ID	説明	0.3	0.4	0.5
42	05000416	投機的フェッチにより、不要な外部 FIFO 動作が発生する:	x	x	x
43	05000425	特定の設定でマルチチャンネル SPORT チャンネルがずれる:	x	x	x
44	05000426	間接ポインタ命令の投機的フェッチにより、偽ハードウェア・エラーが発生する:	x	x	x
45	05000436	ハイバネート状態に入るとき特定の GPIO ピンの状態が変わることがある:	.	x	x
46	05000443	ハードウェア・ループの終わりにある IFLUSH 命令により無限ストールが発生する	x	x	x
47	05000461	RETI が無効なメモリを指すと偽ハードウェア・エラーが発生する:	x	x	x
48	05000462	スタートアップ時の同期問題により SPORT 送信チャンネルでデータ不整列が発生する:	x	x	x

キー: x = アノーマリが存在するレビジョン
. =適用なし

シリコン・アノーマリの詳細リスト

次のリストに、ADSP-BF538/BF538F のすでに知られているすべてのシリコン・アノーマリを示します。説明、対策、適用シリコン・レビジョンを含みます。

1. 05000074—スロット 1 に dsp32shiftimm を使用しスロット 2 で P レジスタ・ストアを行うマルチ発行命令はサポートされていません:

説明:

スロット 1 に dsp32shiftimm を使用しスロット 2 で P レジスタ・ストアを行うマルチ発行命令はサポートされていません。これは例外発生の原因になります。

次のタイプの命令はサポートされていません。これは、スロット 1 に dsp32shiftimm を伴う状態で、P3 レジスタがスロット 2 でストアされるためです。

```
R0 = R0 << 0x1 || [ P0 ] = P3 || NOP; // Not Supported - Exception
```

これは rotate 命令にも適用されます。

```
R0 = ROT R0 by 0x1 || [ P0 ] = P3 || NOP; // Not Supported - Exception
```

サポートされている命令の例:

```
R0 = R0 << 0x1 || [ P0 ] = R1 || NOP;
R0 = R0 << 0x1 || R1 = [ P0 ] || NOP;
R0 = R0 << 0x1 || P3 = [ P0 ] || NOP;
R0 = ROT R0 by R0.L || [ P0 ] = P3 || NOP;
```

対策:

アセンブリ・プログラムで、マルチ発行命令を 2 つの命令に分離してください。この対策は開発ツール・チェーンおよび/またはオペレーティング・システムのソース・コードに組み込まれる予定です。VisualDSP++ や VDK のような ADI がサポートしているツール・チェーンとオペレーティング・システム（VisualDSP++、VDK、GNU Tool Chain、Linux kernel）の詳細については、該当するドキュメントとリリース・ノートのヘルプ・ページ"Silicon Anomaly Tools Support" をご覧ください。

その他のツール・チェーンとオペレーティング・システムについては、該当するサポート・ドキュメントを参照してください。

適用レビジョン:

0.3、0.4、0.5

2. 05000119—ペリフェラル受信チャンネル DMA が停止した後の DMA_RUN ビットが無効:

説明:

ペリフェラル受信 DMA が完了した後、DMAx_IRQ_STATUS:DMA_RUN ビットが不定状態になります。

対策:

DMA 割り込みビットおよび/または DMAx_IRQ_STATUS:DMA_DONE ビットを使って、チャンネル動作の完了を調べる必要があります。

適用レビジョン:

0.3、0.4、0.5

3. 05000122—16 ビット・システム MMR レジスタをアクセスするとき Rx.H を使用できない:

説明:

16 ビット・システム MMR レジスタをアクセスする際、データ・レジスタの上位半分を使用できません。上位半分のレジスタを使用すると、不正なデータがシステム MMR レジスタに書き込まれますが、例外は発生しません。例えば、次のアクセスは失敗します。

```
W[P0] = R5.H; // P0 points to a 16-bit System MMR
```

対策:

16 ビット・システム MMR レジスタをアクセスする際、他の形式の 16 ビット転送を使用してください。例えば、

```
W[P0] = R5.L; // P0 points to a 16-bit System MMR
R4.L = W[P0];
R3 = W[P0](Z);
W[P0] = R3;
```

この対策は開発ツール・チェーンおよび/またはオペレーティング・システムのソース・コードに組み込まれる予定です。VisualDSP++ や VDK のような ADI がサポートしているツール・チェーンとオペレーティング・システム(VisualDSP++、 VDK、GNU Tool Chain、Linux kernel)の詳細については、該当するドキュメントとリリース・ノートのヘルプ・ページ"Silicon Anomaly Tools Support" をご覧ください。

その他のツール・チェーンとオペレーティング・システムについては、該当するサポート・ドキュメントを参照してください。

適用レビジョン:

0.3、0.4、0.5

4. 05000166—8～16 の PPI データ長で上位ビットがゼロにならない:

説明:

PPI データ長が 8～16 ビットの場合、メモリに受信された PPI データの上位ビット(PPI データの一部ではない)はゼロである必要があります。例えば、10 ビット PPI データ長を使用する場合、メモリ内の上位 6 ビットはゼロである必要があります。しかし、PPI は上位 6 PPI データ・ピン(PFx ピンとして共用)にあるデータは何でもキャプチャします。

対策:

ソフトウェアでの対策は、受信データを処理する際に上位 6 ビットをマスクして無視することです。

適用レビジョン:

0.3、0.4、0.5

5. 05000179—汎用 TX または RX モードで、PPI_COUNT を 0 に設定できない:

説明:

汎用モードでは、PPI は少なくとも 2 ワードのブロックを受信または送信する必要があります。シングル・ワード転送 (PPI_COUNT 値 = 0) は機能しません。

対策:

なし

適用レビジョン:

0.3、0.4、0.5

6. 05000180— 0 フレーム同期を持つ PPI モードで PPI_DELAY が機能しない:

説明:

セルフトリガーの PPI の連続サンプリング動作では、PPI_DELAY レジスタで指定される遅延カウントが無視されます。このモードがイネーブルされると直ちに、データが転送されます。

対策:

遅延が必要な場合は、ソフトウェアで受信データを無視するか、あるいは少なくとも 1 フレーム同期を持つモードを使用してください。

適用レビジョン:

0.3、0.4、0.5

7. 05000193—極性設定を変更した際のエッジ検出入力での偽 I/O ピン割り込みの発生:

説明:

次のシナリオを考えてみます:

1) ピンをエッジ検出入力に設定します。

2) 立ち上がりエッジで割り込みが発生します。

3) 入力レベルは一定で 0 です。

4) 極性設定を変更して、立ち下がりエッジで割り込みが発生するようにします。

この場合、入力にエッジが実際に存在しなくとも誤割り込みが発生します。これは、後続のすべての書き込みで極性レジスタのこのビットに 1 を書き込むときにも発生します。極性レジスタが 0 にリセットされると、期待通りに割り込みは発生しません。

外部ピン・レベルが 1 である逆の場合には、極性ビットが(0 から 1 へではなく)1 から 0 へ変わったとき(さらに極性レジスタのこのビットへ後続書き込みで 1 を書き込むとき)、誤割り込みが発生します。

複数の I/O ピンがエッジ検出割り込みに設定されている場合は、極性レジスタに対する任意の変更により、これらすべての I/O ピンが上記の影響を受けます。この場合の対策は、これらすべてのピンに適用する必要があります。

同様の考慮は、入力イネーブル・レジスタにも適用されます。エッジ検出割り込みのイネーブル中にこの設定を変更したときにも、不要な割り込みが発生する原因になります。

対策:

極性 (および/または入力イネーブル) レジスタを変更する前に、割り込みをディスエーブルし (PFA ビットまたは PFB IMASK ビットをクリア)、レジスタ設定を変更し、通常通りに割り込み要求をクリアし (データを書き込むかレジスタをクリア)、そして割り込みを再度イネーブルしてください。

適用レビジョン:

0.3、0.4、0.5

8. 05000199—キャリ・フィックス時にカレント DMA アドレスが不正な値を表示する:

説明:

キャリ・フィックス・サイクルでアクティブ・チャンネルの DMA カレント・アドレス・レジスタ (DMAx_CURR_ADDR)が読み出されると、レジスタの上位半分が 1 だけずれます。LSB は新しい値で更新されますが、MSB は前の値を維持します。レジスタを 2 回目に読み出すと、正しい値が返されます。

アドレスが 64k 境界を超えるように DMA アドレスが変更されると、キャリ・フィックス・サイクルが発生します。DMA アドレスが 64k アドレス境界を跨ぐことがない場合、読み出しは正しく行われます。

対策:

1) 64K アドレス境界を超える DMA アドレスは回避してください。または 2) DMA カレントアドレス・レジスタを 2 回読み出して、読み出した値を確認してください。

適用レビジョン:

0.3

9. 05000219—ブート時の NMI イベントで予期しない状態が発生する:

説明:

ブート時 NMI ピンがアサートされると、ブート ROM 内にハンドラがないためブート・プロセスが失敗します。動作は予測できません。

対策:

ブート・シーケンス時に NMI ピンをアサートしないでください。

適用レビジョン:

0.3、0.4、0.5

10. 05000229—SPI スレーブ・ブート・モードでレジスタがリセット値から変更される:

説明:

このブート・モードでは、DMA5_CONFIG レジスタと SPI_CTL レジスタが、ユーザのアプリケーション・コードを実行する前にデフォルト (リセット) 状態に回復されません。ストップ・モードでは DMA5 チャンネルがイネーブルされたままになり、RX DMA モードで SPI がイネーブルされたままになります。

対策:

ユーザのアプリケーションで、SPI または DMA チャンネル 5 を使用する前にこれらのレジスタをリセットする必要があります。

適用レビジョン:

0.3、0.4、0.5

11. 05000233—2 または 3 内部フレーム同期送信モードで PPI_FS3 が出力されない:

説明:

このモードでは、PPI_CONTROL レジスタの PORT_CFG フィールドに #b11 が設定されると (Sync PPI_FS3～PPI_FS2)、PPI_FS3 フレーム同期信号が PF3 フラグ・ピンへ出力されません。一方、PORT_CFG フィールドに#b01 が設定されると (Sync PPI_FS3～PPI_FS1)、PF3 に正しく出力されます。

対策:

なし

適用レビジョン:

0.3、0.4、0.5

12. 05000245—条件付き分岐のシャドウ内のアクセスで偽ハードウェア・エラーが発生する:

説明:

条件付きジャンプが採用したパスと反対側のコントロール・フロー上で、あるロードが予約済みまたは不正メモリにアクセスすると、偽ハードウェア・エラーが発生します。

次のシーケンスに、これが発生する状況を示します:

シーケンス#1:

"非採用と予測される"分岐に対しては、パイプラインは非採用と予測された分岐命令にシーケンシャルに続く命令をロードします。パイプラインのデザインにより、これらの命令は、分岐の誤予測によりアボートされる前に、投機的に実行することができます。このアノーマリは、分岐に続く 3 個のいずれかの命令スロットにハードウェア・エラーを発生させるロードが含まれている場合に発生します:

```
BRCC X [predicted not taken]
R0 = [P0];                    // If any of these three loads accesses non-existent
R1 = [P1];                    // memory, such as external SDRAM when the SDRAM
R2 = [P2];                    // controller is off, then a hardware error will result.
```

シーケンス#2:

"採用が予測される"分岐の場合は、分岐のディステネーションにある 1 命令スロットは、ハードウェア・エラーが発生するアクセスを含むことができません:

```
BRCC X (BP)
Y: ... ...
X: R0 = [P0];                 // If this instruction accesses non-existent memory,
                              // such as external SDRAM when the SDRAM controller
                              // is off, then a hardware error will result.
```

対策:

アセンブリでプログラムする場合は、上記条件を回避する必要があります。

この対策は開発ツール・チェーンおよび/またはオペレーティング・システムのソース・コードに組み込まれる予定です。VisualDSP++ や VDK のような ADI がサポートしているツール・チェーンとオペレーティング・システム(VisualDSP++、 VDK、GNU Tool Chain、Linux kernel)の詳細については、該当するドキュメントとリリース・ノートのヘルプ・ページ"Silicon Anomaly Tools Support" をご覧ください。

その他のツール・チェーンとオペレーティング・システムについては、該当するサポート・ドキュメントを参照してください。

適用レビジョン:

0.3、0.4、0.5

13. 05000253—タイマの最大外部クロック速度:

説明:

汎用タイマは、TMRx ピンに PWM 出力波形を発生します。このタイミングは、システム・クロック(SCLK)の周期または外部から入力されたクロック(TMRCLK または TACLK)の周期で決定されます。正常動作のためには、SCLK が使用するソース(TMRCLK または TACLK)より高速である必要があります。

データシートとハードウェア・リファレンス・マニュアルの仕様では、TMRCLK と TACLK の速度は最大 1/2 SCLK まで可能です。

最大レートはこの規定値より低くなります。最小 SCLK/TMRCLK または SCLK/TACLK 比は 2.5～2.7 です。

正確な値はまだキャラクタライゼーションされていません。

対策:

最小 SCLK/TMRCLK または SCLK/TACLK 比を 3 にして安全に使用してください。

適用レビジョン:

0.3、0.4、0.5

14. 05000270—活発な I/O 動作により、内部電圧レギュレータ(Vddint)の出力電圧が低下する:

説明:

活発な I/O 動作により、VDDint が低下することがあります。ループの設定ポイントの発生に使用されるリファレンス電圧が、電源ノイズにより低下します。電圧は、アプリケーションの動作周波数を満たすために必要な最小値より低いレベルに低下することがあります。VDDint は、I/O 動作が停止すると設定された値に戻ります。

対策:

この問題は、外部電圧レギュレータを使用した場合には発生しません。この問題がアプリケーションで発生するか否かを調べるときは、次の条件/セットアップで VDDint 波形を観測してください:

- VDDext 電源の偏差に基づいて最大 VDDext を入力します。
- 定常状態(非スタートアップ)でアプリケーションを実行します。
- オシロスコープを最小グラウンドと信号ループで VDDint に接続します。
- 設定する値より 5%低い VDDint 値でトリガーするようにオシロスコープを設定します。次の項目により、この問題を軽減できることがあります:
- 可能な場合、SCLK 周波数を下げて I/O 動作を減らします。
- 観測される低下値に近い値(50mV の整数倍)だけ電圧レギュレータの設定値を大きくします。
- VDDext に十分なバイパスを設けます。

適用レビジョン:

0.3

15. 05000272—あるデータ・キャッシュ・ライトスルー・モードが Vddint <= 0.9V で失敗する:

説明:

ライト・スルー・モードでデータ・キャッシュをイネーブルし、DCPLB の AOW ビットをセットしないで、Vddint を 0.9V 以下にすると、データが破壊されることがあります。

対策:

Vddint <= 0.9V のとき、ライト・バック・モードでデータ・キャッシュを動作させるか、またはライト・スルー・モードの動作中に、DCPLB の AOW ビットをセットしてください。Vddint が 0.9V より高いときは、このアノーマリは発生しません。

適用レビジョン:

0.3、0.4、0.5

16. 05000273—同期 SDRAM メモリへの書き込みが失われる:

説明:

コア・クロックがシステム・クロックより少なくとも 2 倍高速でない場合、SDRAM メモリに対する 32 ビット以上の幅の書き込みが失われてしまうことがあります。キャッシュ・ビクティムは実質的に 256 ビット幅の書き込みであるため、キャッシュ・ビクティムにより、このアノーマリも発生します。

対策:

1)コア・クロック(CCLK)がシステム・クロック(SCLK)より少なくとも 2 倍高速であることを確認するか、あるいは
2)すべての外部メモリ書き込みが 16 ビット以下の幅であることを確認します:

```
W[P2] = R0; // 16-bit write
B[P2] = R0; // 8-bit write
```

データ・キャッシュを使う場合、キャッシュ・ビクティムがないライト・スルー・ポリシーを使う必要があります。

適用レビジョン:

0.3

17. 05000277—エッジ検出の 1SCLK サイクル後の I/O データ・レジスタへの書き込みが割り込みをクリアする:

説明:

任意の I/O データ・レジスタへの書き込み(データ、レジスタのクリア、セット、トグル)がエッジ検出割り込みでエッジが検出された 1 システム・クロック・サイクルに発生した場合、セットされてから 1 システム・クロック・サイクル後にこのビットがクリアされます。

ビットが割り込みを発生するように設定されている場合は、割り込みが発生しますが、割り込みを発生させたビットが表示されません。コア・クロックが動作していないとき、または SIC_IMASK ビットが割り込みをイネーブルするように設定されていないときは、割り込みが失われます。

対策:

1 つのエッジ検出ソースのみが 1 つの割り込みに割り当てられた場合、それが割り込み原因と見なされるので、SIC_ISR レジスタと I/O レジスタの読み出し命令は不要です。イネーブルされると、すべての割り込みが正しく実行されることに注意してください。

エッジ検出割り込みの代わりにレベル検出割り込みを使ってください。割り込みサービス・ルーチンの再起動を防止して、次のエッジを検出するために、受信エッジの間で極性をトグルさせてください。2 つのエッジの間の遅延が割り込みサービス・ルーチンのサービスに十分であるか、または要求ラインに使用できる場合、これが適用できます。極性のトグルは、両エッジを探す場合に使用することができますが、1 個だけのエッジの場合は、他方の割り込みを無視する必要があります。

適用レビジョン:

0.3

18. 05000278—DMA 実行中にペリフェラルをディスエーブルすると DMA システムが不安定になる:

説明:

DMA の実行中で、かつ関係する DMA チャンネルがディスエーブルされる前に、ペリフェラル(PPI、SPORT、SPI など)がディスエーブルされると、DMA システムが壊れます。複数の DMA チャンネルが並行動作するアプリケーションでは、このアノーマリはデータが喪失するか、またはごちゃ混ぜのデータが転送されることを示しています。アノーマリは 1 個の DMA チャンネルを使うアプリケーションにも影響しますが、ユーザ・コードによりペリフェラルがシャットダウンされる場合その影響は見えません。

対策:

DMA チャンネルが動作中の場合、ペリフェラルの関係する DMA チャンネルをディスエーブルした後にペリフェラル自体をディスエーブルしてください。

DMA チャンネルが停止している場合は、ペリフェラルをディスエーブルした後に、関係する DMA チャンネルをディスエーブルする必要があります。チャンネルがディスエーブルされると、DMA ユニットはペリフェラル割り込みを無視して、それを直接割り込みコントローラへ渡すため、偽割り込みが発生します。

適用レビジョン:

0.3

19. 05000281— ISR コンテキストが回復されないとき偽ハードウェア・エラーが発生する:

説明:

ケースによっては、コンテキストを回復しない既存の割り込みサービス・ルーチン(ISR)が必要になります。次のシーケンスを考えてみます:

```
ISR_Exit:
RAISE 14; // instruction A
RTI; // instruction B
```

このシーケンスは現在の割り込みレベルから戻ると、レベル 14 の割り込みサービス・ルーチンを直ちに実行します。理想的に、後者はユーザ・レベルに戻る前にコンテキストを回復するため、最初の ISR で時間を節約します。

問題の説明のため、最初の割り込みは次の命令で:

```
Rx = [Py]; // instruction C
```

あるいは同様の命令で発生するものとします。

プロセッサは ISR へジャンプします(RETI には命令 C のアドレスが含まれます)。プロセッサが上記命令 B に到達したとき ISR が Py を変更すると、命令 C を投機的にフェッチし、これが無効アドレスを指すようになります。命令 A のため、命令 B は実行されませんが、ハードウェア・エラー状態がラッチされます。ハードウェア例外が、次のシステム MMR 読み出しで発生します。

対策:

多くの場合、割り込みから戻る前にコンテキストが回復されるため、これは問題になりません。

上記のようなケースでは、投機的フェッチがハードウェア・エラーを発生しないロケーションで RETI レジスタをロードすることで十分です(上記"`raise; rti;`"シーケンスの前)。

適用レビジョン:

0.3

20. 05000282—32 ビット・データとトラフィック制御によるメモリ DMA の破壊:

説明:

このアノーマリは次のケースに適用されます。

1)メモリ DMA (MDMA)チャンネルを 32 ビット・モード(WDSIZE in MDMA_yy_CONFIG = 0b10)で使用し、
さらに、

2) 同じ方向のグループ・アクセスに対してトラフィック制御をイネーブル (DMA_TC_PER レジスタは非ゼロ・フィールドを含む)。

この特別なケースでは、上位と下位ワードが反転し、および/または割り込みが失われます。

対策:

MDMA チャンネルが 16 ビット・モードで使用される場合、またはトラフィック制御がディスエーブルされる場合(DMA_TC_PER = 0x0000)、このアノーマリは適用されません。

注:このデバイスでは、L1 から外部メモリへとその逆の転送では、16 ビット MDMA の方が 32 ビット・モードより効率が優れています。

適用レビジョン:

0.3

21. 05000283—特定のステージでシステム MMR 書き込みが停止されると、直ちにこれがストールする:

説明:

次のシーケンスを考えてみます:
1)システム MMR 書き込みがストールさせられます。
2)システム MMR 書き込みがストールさせられている間(書き込みが停止)、割り込み/例外が発生します。
3)割り込み/例外サービス・ルーチンが SSYNC 命令を実行します。

このアノーマリが発生するためには、プログラム・フローの変化により、実行パイプラインの特定の 1 ステージで書き込みが停止させられる必要があります。この場合、このアノーマリのために、MMR ロジックが停止させられたシステム MMR アクセスがまだ有効であると見なすようになります。このため、SSYNC がプロセッサを無限に、または高い優先順位からの割り込みまたはイベントにより、割り込まれるまでストールさせられます。

同様に、システム MMR 書き込みが条件付き分岐のような命令自体により停止させられる場合、ストア・バッファがフルで、低速な外部メモリへ出力しているときに無限ストールが発生します。

```
cc = r0 == r0;                  // always true
if cc jump skip; W[p0] = r1.l; // System MMR access is fetched and killed
skip: ...
```

注:ユーザがデバッグ・ツールを使ってプロセッサをハンドラ内で停止させようとすると、無限ストールによりエミュレーション・イベントもロックアウトされます。

対策:

対策は、アプリケーションに副作用を与えない別のシステム MMR アクセス停止により、MMR ロジックをリセットすることです。例えば、CHIPID レジスタからの読み出しを使います。各割り込み/例外ハンドラの開始に実行される次のコードは、このアノーマリの対策になります:

```
cc = r0 == r0;                  // always true
p0.h = 0xffc0;                  // System MMR space CHIPID
p0.l = 0x0014;
if cc jump skip;                // always skip MMR access, but MMR access is fetched and killed
r0 = [p0];                      // bogus System MMR read to work around the anomaly
skip: ...                       // continue with handler code
```

条件付き分岐により MMR 書き込みが停止されるケースでは、2 個の NOP または他の非 MMR 命令を条件付き分岐の直後のロケーションへ挿入することで十分です。

注:デバッグ・セッションでのロックアップを防止するためには、ハンドラ・コード内でプロセッサを停止させることが必要な場合、必ず上記コードの後ろにブレーク・ポイントを設定してください。

適用レビジョン:

0.3

22. 05000288—FIFO が満杯になると SPORT が不正なデータを受信する:

説明:

SPORT が次のように設定されると、不正なデータを受信します:

1)セカンダリ受信データをイネーブル(RXSE=1)、すなわちワード長> 16 ビット。
さらに、

2) RX FIFO に 8 ワードのデータが詰まっている。
さらに、

3)さらに 1 ワードが SPORT に入力される。

この場合、データを保持する余裕が残っているため、オーバーフローがアサートされません。FIFO からデータが取り出されることなく次のデータが受信されると、オーバーフローがアサートされます。

このアノーマリにより、セカンダリ・データ(RxSEC=1)の代わりにプライマリ・データが受信されるかまたはワードがスワップ(SLEN>0xF)します。後続のワードは正常に受信されます。

対策:

問題の説明で示した状態を回避してください。

FIFO オーバーフローの近くで動作させないようにしてください。

適用レビジョン:

0.3

23. 05000291—CAN メールボックスとアクセプタンス・マスク領域からの読み出しが失敗する:

説明:

コアと CAN コントローラが CAN メールボックス RAM または CAN アクセプタンス・マスク RAM を同時にアクセスしようとすると、調停不具合が発生します。この不具合が発生すると、コア読み出しにおいて、コアが読み出そうとしているロケーションからデータを取得する代わりに、CAN コントローラが現在アクセス中の CAN メールボックスまたはアクセプタンス・マスク RAM ロケーションからのデータが返されます。CAN コントローラは、次の 3 つの時間ウインドウで RAM をアクセスします。

1)各フレーム間の間隙で、待ち状態の送信要求を調べて、優先順位の高いイネーブルされた TX メールボックスを探します。CAN コントローラは、内部 TX バッファ内で CAN TX メッセージを構成して CAN バスへ出力するために、まず優先順位の高い TX メールボックスの CAN メールボックス RAM (ID、長さ、データ)を読み出します。

2) CAN コントローラは、CAN RX ピン上で検出された各メッセージの調停フェーズの後、すべてのメッセージ・センターに問い合わせて、ID レジスタとアクセプタンス・マスクを同時にスキャンすることにより一致するメッセージ ID を探します。一致の有無に関係なく、すべてのメッセージ・センターが調べられます。

3) メッセージ ID の一致が 2)で検出された場合、CAN コントローラは CAN メールボックス RAM をアクセスし、メッセージの ID、長さ、データ (イネーブルされている場合はタイムスタンプ)をターゲットのメッセージ・センターに格納します。

対策:

CAN メールボックス RAM またはアクセプタンス・マスク RAM からのコア読み出しが必要な場合は、目的の CAN RAM ロケーションに対して読み出しを約 80 SCLK サイクル間をおいて 3 回行い、多数決判断により正常な読み出しを決めます。この 80 SCLK の遅延により、不具合が発生している同じ不良ウインドウで 2 回もの読み出しが行われないようにします。この対策では特別なタイミングが必要とされるため、割り込みをディスエーブルして、調停ロジックが失敗する複数のウインドウに跨がって読み出しが発生しないようにする必要があります。次の疑似コードは 1 つの例です:

```
R0 = CAN_MB00_DATA0;              // Base Address of CAN Mailbox RAM
R1 = MAILBOX_NUMBER << 5;         // << 5 Provides Offset into CAN RAM
P0 = R0 + R1;                     // Add offset to base and set p0 pointer
P1 = TEMP_AREA;                   // 3 local copies of mailbox (DATA3/2/1/0, LENGTH, ID0/1)
P2 = CAN_STATUS;                  // System MMR reads insert needed SCLK domain delays
CLI R1;                           // Disable interrupts (begin critical region)
P3=3; LSETUP(l10b,l10e) LC0=P3;// Read entire mailbox 3 times
l10b: R0 = W[P0+CAN_MB00_DATA3-CAN_MB00_DATA0](Z); W[P1++] = R0;      // Read/Store DATA3
   R0 = W[P0+CAN_MB00_DATA2-CAN_MB00_DATA0](Z); W[P1++] = R0;  // Read/Store DATA2
   R0 = W[P0+CAN_MB00_DATA1-CAN_MB00_DATA0](Z); W[P1++] = R0;  // Read/Store DATA1
   R0 = W[P0+CAN_MB00_DATA0-CAN_MB00_DATA0](Z); W[P1++] = R0;  // Read/Store DATA0
   R0 = W[P0+CAN_MB00_LENGTH-CAN_MB00_DATA0](Z); W[P1++] = R0; // Read/Store LENGTH
   R0 = W[P0+CAN_MB00_ID0-CAN_MB00_DATA0](Z); W[P1++] = R0;    // Read/Store ID0
   R0 = W[P0+CAN_MB00_ID1-CAN_MB00_DATA0](Z); W[P1++] = R0;    // Read/Store ID1
   P3=16; LSETUP(l20b,l20e) LC1=P3;
   l20b:
   l20e: R0 = W[P2](Z);           // 16 reads provides required delay in SCLK domain
l10e: NOP;
STI R1;                           // Restore interrupts (end critical region)
P1 = TEMP_AREA;                   // Restore P1 for comparison
P2 = GOOD_READ_DATA;              // Local copy of mailbox area to write good data to
P3=7; LSETUP(l30b,l30e) LC0=P3;// Compare loop for all 7 registers in mailbox RAM
l30b: R0 = W[P1](Z); R1 = W[P1+7*2](Z); R2 = W[P1+7*4](Z); // Get the values read
   CC = R1 == R2;                 // Compare 2nd read to 3rd read
   IF CC R0 = R2;                 // If equal, this is the good value. Save to R0.
   P1+= 2;                        // Check next register in mailbox RAM
l30e: W[P2++] = R0;               // If 2nd read != 3rd read, 1st read was good
```

コンパイラが発生するコードはタイミング要求を満たさないため、上記コードを C/C++に変換することは推奨できません。C/C++ コードを使う場合は、この ASM コードをインライン・アセンブリし、必要に応じて、C/C++ データ・タイプを使えるようにマクロを生成することができます。

適用レビジョン:

0.3

24. 05000293—ハイバネート・リーク電流が規定値より大きい:

説明:

プロセッサのデータシートでは、ハイバネート・リーク電流は 50uA と規定されています。このアノーマリのため、実際の測定値は約 200uA になります。

対策:

なし。

適用レビジョン:

0.3

25. 05000294—外部フレーム同期を使う PPI TX モードに対するタイマ・ピンの制約

説明:

ある PPI 設定では、汎用タイマをフレーム同期信号として使うことができます。PPI を 1 個または複数の外部フレーム同期を使う送信モードに設定する場合、汎用タイマの機能が制約されます。

対策:

外部フレーム同期を使うすべての送信モードで、フレーム同期に接続されるタイマ・ピン (TMR1 および/または TMR2)は、入力に設定する必要があります。出力パッド・ディスエーブル機能をアクティブ (OUT_DIS = 1)にして、EXT_CLK モード (TMODE = 11)でタイマをイネーブルする必要があります。

PPI_FS1 の場合:

```
*pTIMER1_CONFIG = OUT_DIS | EXT_CLK;
*pTIMER_ENABLE |= TIMEN1;
```

PPI_FS2 の場合:

```
*pTIMER2_CONFIG = OUT_DIS | EXT_CLK;
*pTIMER_ENABLE |= TIMEN2;
```

注: 外部フレーム同期として使用する場合、タイマは汎用として使えません。

適用レビジョン:

0.3、0.4、0.5

26. 05000301—メモリーメモリ間 DMA のソース/ディステネーション・ディスクリプタは同じメモリ空間にある必要がある:

説明:

MemDMA のソースとディステネーション・ディスクリプタが異なるメモリ空間(1 つは内部メモリ、他方は外部メモリ)にあり、かつトラフィック制御がターンオンされている場合、フェッチ中のディスクリプタ・ワードのソース・ディスクリプタ・カウントが、カレント・ディステネーション・ディスクリプタ・カウント(元のソース・ディスクリプタ・カウントに一致しないことがあります)の値により破壊されることがあります。このために、ソースがフェッチするディスクリプタ・エレメント数が意図した値に一致しなくなります。

これにより発生する 1 つの結果として、ディスクリプタの幾つかのエレメントがロードされなくなります。別の可能な結果としては、余分なディスクリプタ・エレメントのフェッチが行われることがあります。余分なフェッチ数が多いとき、先頭の数レジスタ(例えば、次のディスクリプタ・ポインタ)で正常データが不正データにより上書きされる場合、ディスクリプタ・エレメント・ポインタもオーバーフローして、レジスタの設定された開始点に戻ります。この最後のケースでは、DMA が無効なポインタをフェッチしたとき、次のディスクリプタをフェッチするまで DMA がエラーしたように見えません。

対策:

ソースとディステネーションのディスクリプタを同じメモリ空間に配置してください。両方とも、外部メモリまたは内部メモリに配置する必要があります。

適用レビジョン:

0.3

27. 05000304—CAN/DMA MMR レジスタへの書き込み後、SSYNC が正常に処理されないことがある:

説明:

DMA コントローラと CAN ペリフェラルは、同じ空間に対するアクセス中、それぞれ MMR 空間に対するペリフェラル・アクセス・バスへのアクセスを遅延させることができます。この遅延は、書き込みの後に続く、アプリケーション・コード内の後続 SSYNC 命令の継続時間を超えることがあり、このために望ましくない結果が発生します。

DMA コントローラの場合、DMA チャンネルがメモリからディスクリプタをフェッチする許可を得ているとき、同じ DMA コントローラに対応するシステム MMR に対する他のアクセスをディスクリプタ・フェッチが完了するまで待たせます。この待ち時間はフェッチされるディスクリプタのサイズに応じて、SCLK で数サイクルを要することがあります。この動作による悪影響は、ISR コードが正しいシーケンスを実行して割り込み要求をクリアする DMA 割り込みの場合に発生します:

```
p0.h = hi(DMA3_IRQ_STATUS);
p0.l = lo(DMA3_IRQ_STATUS);
r0.l = 0x0001; w[p0] = r0.l;    // Write-1-to-Clear Interrupt Request
ssync;                          // Allow write to complete
rti;
```

同じ DMA コントローラの別の DMA チャンネルが書き込みのときに、ディスクリプタをフェッチ中である場合、この書き込みが遅延されるため、この遅延が後続の SSYNC 命令の継続時間を超えると、ISR コードが RTI 命令を実行して、再度 ISR が起動されます。これは、DMAx_IRQ_STATUS ビットがまだクリアされていないために発生します。この動作は、ディスクリプタ・フェッチでビジー中の DMA コントローラに関係するすべてのシステム MMR に対しても当てはまります。

CAN コントローラの場合、CAN コントローラがメッセージの送信または受信を準備しているとき、RAM 領域に対するアクセスにより同様の結果が発生します。この RAM 領域を構成する CAN レジスタは、アクセプタンス・マスクレジスタ(CAN_AmxxL と CAN_AMxxH)、メールボックス RAM レジスタ(CAN_MBxx_DATA0、CAN_MBxx_DATA1、CAN_MBxx_DATA2、CAN_MBxx_DATA3、CAN_MBxx_LENGTH、CAN_MBxx_TIMESTAMP、CAN_MBxx_ID0、CAN_MBxx_ID1)です。

対策:

MMR レジスタからのダミー読み出しが SSYNC の前に挿入されると、これにより、前の書き込みが完了した後に読み出しを実行できることが保証されます。例えば、上の DMA コントローラの例を使用して、IRQ ステータス・レジスタを書き込んだ後にこのレジスタをリードバックします:

```
p0.h = hi(DMA3_IRQ_STATUS);
p0.l = lo(DMA3_IRQ_STATUS);
r0.l = 0x0001; w[p0] = r0.l;    // Write-1-to-Clear Interrupt Request
r0.l = w[p0];                   // Insert dummy read before ssync
ssync;                          // Allow write to complete
rti;
```

適用レビジョン:

0.3

28. 05000307—ハイバネート中に SCKELOW ビットが状態を維持しない:

説明:

VR_CTL の SCKELOW ビット(ビット 15)がハイバネート・リセット・シーケンス中にステータスを維持しないため、ブート・シーケンスでこれを読み出して、プロセッサがコールド・スタートしたのか、またはハイバネート状態から抜け出たのか調べることができません。

対策:

ハイバネート機能を使用している場合は、ハイバネートに入る前にアプリケーション・コードにより VR_CTL を外部メモリの特定ロケーションへコピーすることができます。初期化ブロックでこれを調べると、コンテキストの待避が事前に行われた否かを知ることができます。例えば、ソース・コード内に 32 ビットのデータ・エレメントを生成して、LDF を使って、特定のロケーションへ待避させます。LDF 内に:

```
RESOLVE(32BitDataLabel, 32BIT_ADDR_IN_EXTERNAL_SDRAM_DATA_SEGMENT);
```

次に、ソース・コード内でハイバネートに入る準備をしているとき、この疑似コードを次のように使うことができます:

```
#define VR_CTL_HIBERNATE_VALUE 0x000080DC  // Your value for VR_CTL goes here
PTR_TO_32BitDataLabel = VR_CTL_HIBERNATE_VALUE;   // 32-Bit Access
VR_CTL = VR_CTL_HIBERNATE_VALUE;     // 16-Bit Access
CLI/IDLE PLL Programming Sequence;  // Latch write to VR_CTL
```

次に、初期化ブロックで、この疑似コードを使ってこの 32 ビット値を調べて、このロケーションが前に書き込まれたか否かを知ることができます:

```
Read PTR_TO_32BitDataLabel;
Compare to VR_CTL_HIBERNATE_VALUE;
if TRUE
Execute post-hibernate boot sequence;
else
Perform full boot;
```

適用レビジョン:

0.3

29. 05000310—予約済みメモリの境界でのフェッチにより、偽ハードウェア・エラーが発生する:

説明:

予約済みメモリ、または有効な CPLB でカバーされている L1 命令キャッシュ・メモリ(命令キャッシュがイネーブルされている場合)の境界でのフェッチにより、偽ハードウェア・エラーが発生します(外部メモリ・アドレシング・エラー)。

対策:

1)境界の前では、予約済みメモリ空間を使い少なくとも 76 バイトを空けてください。これにより偽ハードウェア・エラーの発生を防止することができます。

2)ハードウェア・エラーが有効か否かをアクションの前に判断するハードウェア・エラー・ハンドラを設けてください。これは、CODE_FAULT_ADDR レジスタ値が有効ページ内のアドレスであることを確認することにより行うことができます。この場合、アクションは不要です。

このアノーマリは、L1_code_cache（命令キャッシュがイネーブルされている場合)の境界でも発生することに注意してください。

適用レビジョン:

0.3、0.4、0.5

30. 05000312—SSYNC、CSYNC、または LT、LB、LC レジスタへのロードが割り込みを受けるとエラーが発生する:

説明:

命令キャッシュがイネーブルされた場合、次の命令が割り込みされると、無効なコードが実行されます:

- CSYNC
- SSYNC
- LCx =
- LTx = (LCx が非ゼロの場合)
- LBx = (LCx が非ゼロの場合)

この問題が発生すると、不正命令例外などの様々な不具合が発生します。その他にも、例外、ハードウェア・エラー、または有効な命令であるが予期したものと異なる命令として表示されます。

対策:

すべての **SSYNC**、**CSYNC**、"**LCx =**"、"**LTx =**"、"**LBx =**"命令の前に **cli** を配置して、割り込みをディスエーブルし、これらの各命令の後ろに **sti** を配置して割り込みを再イネーブルしてください。既に割り込みが禁止されているコード内でこれらの命令が実行されると、この問題は発生しません。

割り込みのネスティングをイネーブルする割り込みサービス・ルーチンでは、必ず LCx、LTx、LBx レジスタをプッシュした後に RETI をプッシュして、割り込みのネスティンクをイネーブルしてください。ISR コンテキストの回復時に逆を行うと、RETI をポップした後にループ・レジスタがロードされることが保証されるため、割り込みのネストがディスエーブルされて、このアノーマリ状況からロードが保護されます。例えば、

```
INT_HANDLER:
[--sp] = astat;
[--sp] = lc0; // push loop registers before pushing RETI
[--sp] = lt0;
[--sp] = lb0;
[--sp] = lc1;
[--sp] = lt1;
[--sp] = lb1;
[--sp] = reti; // push RETI to enable nested interrupts
[--sp] = ...
// body of interrupt handler
... = [sp++]; reti = [sp++]; // pop RETI to disable interrupts
lb1 = [sp++]; // it is now safe to load the loop registers
lt1 = [sp++];
lc1 = [sp++];
lb0 = [sp++];
lt0 = [sp++];
lc0 = [sp++];
astat = [sp++];
```

最後に、この対策はスーパーバイザ・モード命令を使って割り込みをディスエーブルおよびイネーブルするため、ユーザ・モードには適用されません。ユーザ領域では、CSYNC 命令または SSYNC 命令を使用しないでください。また、ループ・レジスタを直接ロードしないでください。その代わり、LSETUP 命令を使って構成できるハードウェア・ループを使ってください。これにより、ループ範囲が 2046 バイトに制限されます。

適用レビジョン:

0.3、0.4

31. 05000313—シングル・フレーム同期モードの最初の転送で PPI がレベル検出になる:

説明:

PPI をシングル外部フレーム同期でトリガーするように設定すると、最初の転送を除くすべての転送で、フレーム同期のエッジが必要とされます。最初の転送でのみ、フレーム同期入力がレベル検出されます。フレーム同期がアクティブ状態にあると、このために PPI が転送を開始します。このため、PPI が早めに開始されてしまいます。

このアノーマリは、PPI が 2 または 3 フレーム同期を使用する場合には適用されません。

対策:

PPI でシングル外部フレーム同期を使用するとき、PPI がイネーブルされたときにフレーム同期が非アクティブ状態となるようにしてください。

適用レビジョン:

0.3

32. 05000315—次のシステム MMR アクセスでシステム MMR 書き込みの停止が異常完了する:

説明:

次のシーケンスを考えてみます:
1)システム MMR 書き込みがストールさせられます。
2)システム MMR 書き込みがストールさせられている間(書き込みが停止)、割り込み/例外が発生します。
3)割り込み/例外サービス・ルーチンが任意の MMR をアクセス(読み出しまたは書き込み)します。

このアノーマリが発生するためには、プログラム・フローの変化により、実行パイプラインの特定の 1 ステージで書き込みが停止させられる必要があります。この場合、このアノーマリのために、MMR ロジックが停止させられたシステム MMR アクセスがまだ有効であると見なすようになります。ハンドラ内でのシステム MMR に対する次のアクセス(読み出し/書き込み)により、前にストールした書き込みが異常完了させられます。

同様に、システム MMR 書き込みが条件付き分岐のような命令自体により停止させられる場合、ストア・バッファがフルで、低速な外部メモリへ出力しているときに異常書き込みが発生します。

```
cc = r0 == r0;                  // always true
if cc jump skip;
W[p0] = r1.l;                   // System MMR access is fetched and killed
skip: ...
```

注:デバッグ・ツールを使った次のシステム MMR アクセスの前に、ハンドラ内でプロセッサが停止した場合、プロセッサは書き込みの完了を待つため無限にストールして、エミュレート・イベントがロックアウトされます。

対策:

対策は、分岐のシャドウ内の別のシステム MMR アクセス停止により、MMR ロジックをリセットすることです。例えば、システム MMR CHIPID レジスタからの読み出しをセットアップして、その後それを停止させると、システムに他の副作用を与えないアクセス停止が生成されます。したがって、各ハンドラの開始に実行される次のコードは、このアノーマリの対策になります:

```
cc = r0 == r0;                  // always true
p0.h = 0xffc0;                  // System MMR space CHIPID
p0.l = 0x0014;
if cc jump skip;                // always skip System MMR access, but it is fetched and killed
r0 = [p0];                      // bogus System MMR read to work around the anomaly
skip: ...                       // continue with handler code
```

条件付き分岐によりシステム MMR 書き込みが停止されるケースでは、2 個の NOP または他の非 MMR 命令を条件付き分岐の直後のロケーションへ挿入することで十分です。

注:デバッグ・セッションでのロックアップを防止するためには、ハンドラ・コード内でプロセッサを停止させることが必要な場合、必ず上記コードの後ろに所望のブレーク・ポイントを設定してください。

適用レビジョン:

0.3

33. 05000317—PORTFIO または PORTFIO_TOGGLE への書き込みで PFx グリッチが発生する:

説明:

PORTFIO または PORTFIO_ TOGGLE へ書き込みを行うと、PFx ピンでグリッチが発生することがあります。

対策:

書き込み対象のレジスタを読み出した後に、書き込みを行ってください。

適用レビジョン:

0.3

34. 05000355—ハイバネート・ウェイクアップ・ソースがアクティブのままであると、レギュレータの設定ができなくなる:

説明:

プロセッサがハイバネート状態に入った後、後続のペリフェラル・ウェイクアップ・イベントによりデバイスをハイバネートから抜け出させることができます。開始されたリセット・シーケンス中ウェイクアップ・ソースのアサートが続くと、IDLE シーケンスを実行したとき、VR_CTL レジスタに対する書き込みがレギュレータにラッチされません。ウェイクアップ・ソースのアサートが解除されるまで、レギュレータ回路へのラッチができなくなります。

書き込みは実質的に失われますが、書き込まれた値は VR_CTL レジスタの物理メモリ・マップド・アドレスに行きます。VR_CTL への書き込みがさらに実行されない場合は、"失われた"書き込みが次の **IDLE** 命令が実行されたとき、レギュレータ・ハードウェアにラッチされます。

対策:

ペリフェラルハイバネート・ウェイクアップ・ソースのアサートを解除した後に、VR_CTL への書き込みの後に必要とされる **IDLE** シーケンスを使って、レギュレータの設定を行ってください。

適用レビジョン:

0.3、0.4

35. 05000357—チャンネル 0 をディスエーブルすると、シリアル・ポート(SPORT)マルチチャンネル送信に不具合が発生する:

説明:

チャンネル 0 をディスエーブルしてマルチチャンネル・モードを設定すると、次のすべての条件を満たすとき、DMA 送信データが正しくない SPORT チャンネルへ送信されます:

1) 外部受信フレーム同期 (SPORTx_RCR1 で IRFS = 0)

2) ウインドウ・オフセット = 0 (SPORTx_MCMC1 で WOFF = 0)

3) マルチチャンネル・フレーム遅延 = 0 (SPORTx_MCMC2 で MFD = 0)

4) DMA 送信パッキングをディスエーブル (SPORTx_MCMC2 で MCDTXPE = 0)

この特別な設定を使用したとき、チャンネル 0 がディスエーブルされていても、非パック・モードでチャンネル 0 プレースホルダ内にある内容は何であっても最初に送信されるため、マルチチャンネル送信データが壊されます。このため、出力ウインドウで 1 ワードのデータ・シフトが発生し、これがシリアル・ストリーム内の後続の各ウインドウで繰り返されます。例えば、チャンネル 0 がディスエーブルされ、チャンネル 1〜7 が送信用にイネーブルされて、非パック送信バッファが{0、1、2、3、4、5、6、7}であり、かつウインドウ・サイズが 8 チャンネルの場合、一連の出力ウインドウ内の予測されるデータ・シーケンスは次のようになります:

1234567--1234567--1234567--1234567

このアノーマリにより、出力は次のように見えます:

0123456--7012345--6701234--5670123

対策:

これに対しては幾つかの対策があります:

1) マルチチャンネル・モードをディスエーブルします。

2) 内部受信フレーム同期を使います。

3) マルチチャンネル・フレーム遅延 >0 を使います。

4) ウインドウ・オフセット >0 を使います。

5) DMA 送信パッキングをイネーブルします。

6) チャンネル 0 をディスエーブルしないようにします。

適用レビジョン:

0.3、0.4

36. 05000366—ITU-R 656 モードで PPI アンダーフロー・エラーが検出されなくなる:

説明:

PPI ポートが ITU-R 656 出力モードに設定されると、PPI FIFO アンダーランが発生したとき、FIFO アンダーラン・ビット(in PPI_STATUS の UNDR)セットされなくなります。帯域幅が十分でない場合、または調停による遅延のために PPI DMA がバス・アクセスの取得に失敗した場合に、アンダーランが発生します。

対策:

なし。

適用レビジョン:

0.3、0.4、0.5

37. 05000371—サブルーチンの継続時間が 5 サイクルを下回るとき、RETS レジスタが壊れる可能性がある:

説明:

RTS 命令がサブルーチンの開始から 4 実行サイクル以内に配置されると、正常にリターンしなくなることがあります。例えば、

```
CALL STUB_CODE;
...
...
...
STUB_CODE:
RTS;
```

これが発生した場合、RETS 内のビットに不具合が発生すると、これによりプロセッサが誤ったアドレスに分岐して、無効なコードが実行されることがあります。

対策:

サブルーチン内で RTS の前に少なくとも 4 実行サイクルある場合、CALL 命令と RTS 命令がこの問題に遭遇するような方法で並ぶことはありません。NOP は 1 サイクル命令であるため、次に示す対策はすべての不具合ケースに対して有効です:

```
CALL STUB_CODE;
...
...
...
STUB_CODE:
NOP; // These 4 NOPs can be any combination of instructions
NOP; // that results in at least 4 core clock cycles.
NOP;
NOP;
RTS;
```

分岐予測は、このシナリオの要因にはなりません。4 サイクル以内で RTS 命令に到達するサブルーチン内部の条件付きジャンプは、この不具合が発生する原因にはなりません。非同期イベント(割り込み、例外、NMI)も、この不具合を発生しません。

この対策は開発ツール・チェーンおよび/またはオペレーティング・システムのソース・コードに組み込まれる予定です。VisualDSP++ や VDK のような ADI がサポートしているツール・チェーンとオペレーティング・システム(VisualDSP++、 VDK、GNU Tool Chain、Linux kernel)の詳細については、該当するドキュメントとリリース・ノートのヘルプ・ページ"Silicon Anomaly Tools Support" をご覧ください。

その他のツール・チェーンとオペレーティング・システムについては、該当するサポート・ドキュメントを参照してください。

適用レビジョン:

0.3、0.4

38. 05000374—ペリフェラル・ウェイクアップをイネーブルしてハイバネート状態に入ると、大きな電流が流れる:

説明:

プロセッサをハイバネート状態にする前に VR_CTL レジスタで任意のペリフェラル・ウェイクアップ・ビットを 1 に設定すると、プロセッサの消費電流は規定の 50 uA ではなく約 10 mA になります。

イネーブルしたペリフェラル・ウェイクアップが存在しない場合 (すなわち、プロセッサを休眠から抜け出させるのは、/RESET ピンのアサーションだけ) は、この大きな消費電流は発生しません。

対策:

なし。

適用レビジョン:

0.3、0.4

39. 05000375—GPIO ピン PC1 と PC4 が通常出力として機能できる:

説明:

プロセッサの GPIO の PC1 ピンと PC4 ピンは、5V 対応入力/オープン・ドレイン出力ピンと規定されていますが、GPIO 出力として設定することができ、出力ピンをハイ・レベルとロー・レベルに駆動する機能を持っています。

対策:

このアノーマリを修正したシリコンへ移行するデザインでは、PC1 および/または PC4 をフル機能の GPIO 出力として使う場合、出力ピンをハイ・レベルに駆動するために外部プルアップ抵抗が必要です。この外部プルアップ抵抗が存在する場合は、このアノーマリがあるシリコン向けに書かれたソフトウェアを変更する必要はありません。

適用レビジョン:

0.3

40. 05000402—プロセッサが非キャッシャブル・メモリから実行されると SSYNC によりストールする:

説明:

割り込みをディスエーブルして SSYNC 命令を非キャッシャブル L2 メモリから実行すると、プロセッサがストールすることがあります。

対策:

任意の割り込みをイネーブルすると、ストールが発生しますが、非同期イベントによりストールから抜け出ます。割り込みをイネーブルしない場合、または割り込みが発生しない場合は、ストールが無限に続くためプロセッサをリセットする必要があります。

ストール状態を回避するためには、次の条件を満たす必要があります。

1) SSYNC が L1 メモリ内またはキャッシャブル L2 メモリ内にある。

2)ループの上部は非キャッシャブル L2 メモリにある場合、SSYNC がループの底部にない。

3) SSYNC がキャッシャブル L2 ページ内にあるとき、次の(アドレス・シーケンシャル)ページが L1 または非キャッシャブル L2 メモリの場合、ページ(CPLB により指定)の終わりから 64 ビット・ワードで少なくとも 8 ワードだけ離れている。

上記条件のいずれかを満たさない場合は、別の対策として、タイマの 1 つを設定して、SSYNC 命令の前でタイムアウト周期により割り込みを発生させてストールを破るようにすることです。

適用レビジョン:

0.3

41. 05000403—レベル検出の外部 GPIO ウェイクアップにより無限ストールが発生する:

説明:

レベル検出の GPIO イベントを使ってプロセッサを低消費電力のスリープ動作モードからウェイクアップさせる場合、ウェイクアップ・パルス幅が狭すぎるとき、プロセッサが無限にストールすることがあります。これが発生した場合、レベルが GPIO ピンで検出されたことにより PLL がスリープ・モードから遷移し始めますが、コアがアイドル状態から抜け出して実行を再開する十分な時間を確保する前にそのトリガー・レベルがなくなると、スリープ・モードに戻ってしまいます。

このため、プロセッサは正常にウェイクアップしません。この時点ではハードウェア・リセットでのみこのストール状態から抜け出すことができます。

対策:

このアノーマリを回避する方法は 2 つあります:

1)ウェイクアップ・イベントの発生に使用するピンにエッジ検出機能を使います。

2)ウェイクアップ信号のエッジのノイズを除去して、少なくとも 3 システム・クロック(SCLK)サイクル間トリガー・レベルを維持します。

適用レビジョン:

0.3、0.4、0.5

42. 05000416—投機的フェッチにより、不要な外部 FIFO 動作が発生する:

説明:

外部 FIFO デバイスが同期メモリ・バンクに接続されると、プロセッサが投機的にメモリ・アクセスを行うことができるので、不正な動作が発生します。これは、FIFO がデータを Blackfin に提供するため、フェッチが投機的に行われるごとに、あるいは投機的アクセスがキャンセルされたとき、データが失われるためです。"投機的"フェッチとは、パイプライン内で起動され、完了前に停止される読み出しです。これは、フローの変更(割り込みまたは例外など)の際に、または分岐のシャドウをアクセスする際に発生します。この動作は、Blackfin プログラマズ・リファレンスで説明されています。

発生する別のケースとしては、フロー変更が例外から発生するようなハードウェア・ループの一部としてアクセスが実行されるケースがあります。例外はディスエーブルできないため、割り込みをディスエーブルした場合でも、例外から投機的フェッチが発生する例を次に示します:

```
CLI R3;                          /* Disable Interrupts */
LSETUP( loop_s, loop_e) LC0 = P2;
loop_s: R0 = W[P0];              /* Read from a FIFO Device */
loop_e: W[P1++] = R0;            /* Write that Generates a Data CPLB Page Miss */
STI R3;                          /* Enable Interrupts */
RTS;
```

この例では、ハードウェア・ループ内での読み出しが、割り込みをディスエーブルして FIFO に対して行われています。ループ内での書き込みからデータ CPLB 例外が発生すると、ループ内での読み出しが投機的に行われます。

対策:

まず、コア読み出しによりアクセスを行う場合、コア読み出しを行う前に割り込みをターンオフさせます。次に、読み出し命令が見えないように、割り込みがターンオフされる前にパイプラインの読み出しフェーズを保護する必要があります。:

```
CLI R0;
NOP; NOP; NOP; /* Can Be Any 3 Instructions */
R1 = [P0];
STI R0;
```

例外から同じ不要な動作が発生しないように保護するため、読み出しをフローの変更から離す必要があります:

```
CLI R3;                              /* Disable Interrupts */
LSETUP( loop_s, loop_e) LC0 = P2;
loop_s: NOP;                         /* 2 NOPs to Pad Read */
   NOP;
   R0 = W[P0];
loop_e: W[P1++] = R0;
STI R3;                              /* Enable Interrupts */ RTS;
```

ループを構成して、最後に NOP のパディングを配置することができます:

```
LSETUP( .Lword_loop_s, .Lword_loop_e) LC0 = P2;
.Lword_loop_s: R0 = W[P0];
   W[P1++] = R0;
   NOP;                              /* 2 NOPs to Pad Read */
.Lword_loop_e: NOP;
```

これらの両シーケンスは、フロー変更から読み出しが投機的に実行されることを防止します。挿入された 2 個の NOP は、投機的アクセスを防止するため、パイプライン内に十分な分離を提供します。これらの NOP は、任意の 2 個の命令にすることができます。

DMA 転送を使って行う読み出しは、投機的アクセスから保護する必要はありません。

適用レビジョン:

0.3、0.4、0.5

43. 05000425—特定の設定でマルチチャンネル SPORT チャンネルがずれる:

説明:

マルチチャンネル・モードでシリアル・ポートを使う場合、SPORT に非常に特別な設定が行われると、送信チャンネルと受信チャンネルがずれます:

1)ウインドウ・オフセット(WOFF) = 0。

2)ウインドウ・サイズが 8 の奇数倍(すなわち WSIZE が偶数> 0)。

3) RFS パルス間の間隔がウインドウ継続時間に一致する。

注: このアノーマリは WSIZE = 0 の場合は適用されません。

この設定を使用すると、最初のウインドウが完了したとき、マルチチャンネル・モード・チャンネル・イネーブル・レジスタが誤ラッチされるために、不正チャンネル割り当てに従って TDV 信号が駆動され、不正なチャンネルで受信データがサンプルされます。

したがって、最初のウインドウは正常に送受信されますが、2 番目以後のすべてのウインドウがずれてしまい、送受信されたデータが壊れます。

このエラーは、外部クロック、内部クロック、RFS に対して発生します。

対策:

幾つかの対策が可能です:

1) 0 以外のウインドウ・オフセットを使います。

2) 8 の偶数倍のウインドウ・サイズを使います。

3)内部 RFS の場合、SPORTx_RFSDIV が少なくともウインドウ・サイズ(イネーブルされるチャンネル数* SLEN)に一致することを確認します。

適用レビジョン:

0.3、0.4、0.5

44. 05000426—間接ポインタ命令の投機的フェッチにより、偽ハードウェア・エラーが発生する:

説明:

間接ジャンプまたは条件付きジャンプが採用したパスと反対側のコントロール・フロー上の予約済みメモリまたは不正メモリを指すポインタを使うコールがあると、偽ハードウェア・エラーが発生します。これは、一般に無効な関数ポインタ(例えば-1 を設定)を使う場合に発生します。例えば、

```
CC = P2 == -0x1;
IF CC JUMP skip;
CALL (P2);
skip:
RETS;
```

IF CC JUMP 命令をコミットできる前に、パイプラインがアドレス-1 (0xffffffff)に対する投機的命令フェッチを発行するため、偽ハードウェア・エラーが発生します。これは、オフェンデイング命令が実際に実行されないための偽ハードウェア・エラーです。これは、条件付き分岐(不採用が予測される)の 2 つの命令内で次のようにポインタが使用されると発生します:

```
BRCC X [predicted not taken]
Y: JUMP (P-reg);              // If either of these two p-regs describe non-existent
   CALL (P-reg);              // memory, such as external SDRAM when the SDRAM
X: RETS;                      // controller is off, then a hardware error will result.
```

対策:

If 命令キャッシュがオンであるか、または ICPLB がイネーブルされる場合、このアノーマリは適用されません。

命令キャッシュがオフであり、ICPLB がディスエーブルされる場合、間接ポインタ命令は分岐命令から 2 命令離れる必要があります。これは NOP を使って実現することができます:

```
BRCC X [predicted not taken]
Y: NOP;                       // These two NOPs will properly pad the indirect pointer
   NOP;                       // used in the next line.
   JUMP (P-reg);
   CALL (P-reg);
X: RETS;
```

適用レビジョン:

0.3、0.4、0.5

45. 05000436—ハイバネート状態に入るとき特定の GPIO ピンの状態が変わることがある:

説明:

リセット時に、GPIO ポート C、D、E ピンのペリフェラル機能を選択し、GPIO ピンとして使うために、PORTxIO_FER レジスタをソフトウェアから設定する必要があります。ハイバネート状態に入ると、GPIO ポートがリセットされるため、すべてのポート C、D、E ピンはそれぞれのペリフェラル機能に設定されます。次に、直ちにプロセッサはすべてのペリフェラル出力ピンをペリフェラル・ピンの非アクティブ状態に駆動した後に、プロセッサはハイバネート状態になります。この時点で、ピンはプロセッサのデータシートの規定に従いスリーステートになります。例えば、CANTX ピン (PC0)は非アクティブのハイ・レベルに駆動された直後にスリーステートになるため、ピンが通常動作で GPIO として使われている場合は、予期しない信号が発生することがあります。ピンが既にハイ・レベルを駆動している場合には問題ありませんが、ロー・レベルを駆動している場合には、予期しないハイ・パルスがシステムに悪影響を与えることがあります。次の表に、影響のある GPIO ピンを示します。

PIN NAME	CURRENT STATE: LOW	CURRENT STATE:	HIGH CURRENT STATE: HI-Z
CANTX/PC0	Low --> High --> Hi-Z	High --> Hi-Z	Hi-Z --> High --> Hi-Z
TX1/PD11	Low --> High --> Hi-Z	High --> Hi-Z	Hi-Z --> High --> Hi-Z
TX2/PD13	Low --> High --> Hi-Z	High --> Hi-Z	Hi-Z --> High --> Hi-Z

この表の見方は、PIN NAME について、休眠シーケンスを実行したときのピン状態が CURRENT STATE: X の場合、ピンは CURRENT STATE: X の列に示す状態に駆動されことを示しています。この表で、正常動作は現在の状態から直接 Hi-Z になることです。このアノーマリでは現在の状態と Hi-Z との間の中間状態になります。

対策:

システム内の他のデバイスでピンが Hi-Z になる前のこのピン変化を許容できない場合は、上の表に示すピンを使わないことが最善の対策です。

適用レビジョン:

0.4、0.5

46. 05000443—ハードウェア・ループの終わりにある IFLUSH 命令により無限ストールが発生する

説明:

IFLUSH 命令がループの終わりに配置されると、プロセッサが無限にストールします。例えば、次の 2 つのコード例ではループから抜け出すことはできません。

```
P1 = 2;
LSETUP (LOOP1_S, LOOP1_E) LC1 = P1;
LOOP1_S: NOP;
LOOP1_E: IFLUSH[P0++];
LSETUP (LOOP2_S, LOOP2_E) LC1 = P1;
LOOP2_S: NOP; NOP; NOP; NOP; // Any number of instructions...
LOOP2_E: IFLUSH[P0++];
```

対策:

ハードウェア・ループの終わりに IFLUSH 命令を配置しないでください。ループの終わりの IFLUSH を任意の命令で置き換えると、次のように問題が回避されます:

```
LSETUP (LOOP_S, LOOP_E) LC1 = P1;
LOOP_S: IFLUSH[P0++];
LOOP_E: NOP; // Pad the loop end
```

適用レビジョン:

0.3、0.4、0.5

47. 05000461—RETI が無効なメモリを指すと偽ハードウェア・エラーが発生する:

説明:

存在しないメモリを対象とする CALL/JUMP 命令を使用すると、ハードウェア・エラー状態が発生します。割り込みをイネーブルしていると、ハードウェア割り込み (IRQ5) が発生します。RETI レジスタには無効なロケーションが格納されているため、別の割り込みをサービスする場合でも、RTI 命令を実行する前にレジスタ値を変更する必要があります。次のシーケンスを考えてみます:

```
P2.L = LO (0xFFAFFFFC);          // Load Address in Illegal Memory to P2
P2.H = HI (0xFFAFFFFC);
CALL(P2);                        // Call to Bad Address Generates Hardware Error IRQ5
....
IRQ5_code:                       // Hardware Error Interrupt Routine
RAISE 14;                        // (1)
RTI;                             // (2)
IRQ14_code:
[--SP] = ( R7:0, P5:0 );         // (3)
[--SP] = RETI;                   // (4)
....
```

ハードウェア・エラーが発生すると、プログラム・カウンタは無効なロケーション 0xFFAFFFFC を指します。このロケーションは、IRQ5 ハードウェア・エラー・イベントのサービス中に RETI レジスタにロードされます。RTI 命令 (2) が実行されると、RETI レジスタで指定される命令(不正アドレス)のフェッチが要求されます。ハードウェアがレベル 14 割り込みが待ち状態になるのを確認する前に、RETI レジスタで指定される命令(不正アドレス)のフェッチが要求されます。このフェッチにより、この命令が実行されなくとも、もう 1 つのハードウェア・エラーがラッチされます。実行は IRQ14 (3)まで進みます。割り込みが再イネーブルされると(4)、直ちに待ち状態のハードウェア・エラーが発生します。

対策:

1. コードで無効なポインタに対するジャンプまたは呼び出しを行わないようにします。

2. ハードウェア・エラーからリターンしたとき、常に RETI レジスタにメモリ・フェッチでハードウェア・エラーを発生しない値を設定します。

適用レビジョン:

0.3、0.4、0.5

48. 05000462—スタートアップ時の同期問題により SPORT 送信チャンネルでデータ不整列が発生する:

説明:

SPORT を外部 SPORT クロックを使うマルチチャンネル・モードに設定した場合、SPORT をイネーブルしたときに同期問題が発生します。この同期問題は、外部 SPORT クロックと Blackfin プロセッサの内部システム・クロック (SCLK) との間のスキューにより、SPORT 内のチャンネル・カウンタが同期外れを起こしたときに現れます。この問題が発生すると、ウインドウの開始点で"無効" チャンネルが挿入されるため、送信チャンネルの残りの部分がアクティブ・ウインドウ内で 1 ロケーション右にシフトされます。最後のチャンネル・データが、2 番目のウインドウで最初にイネーブルされた送信チャンネル・データとして送信され、その後にもう 1 つの "無効" チャンネルが挿入されます。すべてのデータがそっくりそのままシーケンシャルに送信されますが、フレーム同期からずれた正しくないチャンネルで送信されるため、データを再生することはできません。

対策:

このエラーが発生した場合、SPORT を再起動させて再度このエラーをチェックする必要があります。この不具合は極めて希であるため、この不具合を示す連続する再起動に遭遇する確率は極めて小さいものです。

SPORT 割り込みのタイミングを使用するソフトウェア・ソリューションが可能です。SPORT ISR 内で、最初に割り込みが発生したとき CYCLES レジスタにゼロを設定して、2 回目に割り込みが発生したときにこれを読み出します。これにより、SPORT 割り込み自体の頻度についてコア・クロックの基準時点を設定します。2 回目に読み出した値がマルチチャンネル・ウインドウの継続時間(コア・クロック数)を超えた場合には、ストリームに"無効" チャンネルが挿入されているので、SPORT の再起動が必要です。

ハードウェア対策は、マルチチャンネル・モード SPORT の設定方法に大きく依存します。マルチチャンネル・モードでは、TFS は送信データ有効 (TDV) 信号として機能するため、送信チャンネル中、常にアクティブ状態に駆動されます (SPORTx_TCR1 レジスタの LTFS ビットから指定されます)。このため、各アプリケーションでの "無効" チャンネルの検出のためのアクティブ・フレーム内でのチャンネル設定方法に応じて、TDV 信号を TDV ピン状態変化の検出時に割り込みを発生するように設定された GPIO ピンの 1 つに接続することができます。ウインドウ内のすべてのチャンネルが送信チャンネルとして設定され、かつウインドウ・オフセットがなく、かつマルチチャンネル・フレーム遅延がない場合は、RFS パルスの受信後直ちに TDV はアクティブになる必要があります。RFS パルスの幅がウインドウ・サイズに一致する場合 (すなわちアクティブ・ウインドウの後から次の RFS が検出されるまで余分なクロックがない場合)、TDV は動作中アクティブを維持する必要があります。したがって、SPORT がオンの間に TDV が非アクティブになる場合は、不具合の発生を意味し、SPORT の再起動が必要となり、不具合が検出されなくなるまで、このテストを設定したまま動作を繰り返します。

ウインドウ・オフセット、マルチチャンネル・フレーム遅延、アクティブ・ウインドウの終わりと次のフレーム同期との間に余分なクロック、および/またはアクティブ・ウインドウ内に非送信チャンネルが、それぞれ存在するアプリケーションでは、最初の TDV アサーションをマニュアルで行って、"無効" チャンネルを検出する必要があります。次に 1 つの方法を示します:

1. TFS (TDV)を GPIO 割り込みに接続し、TDV がアクティブになったときに割り込みが発生するように設定します。

2. RFS を GPIO 割り込みに接続し、RFS がアクティブになったときに割り込みが発生するように設定します。

3. SPORT 受信クロックを、EXT_CLK モードに設定された TMRx ピンに接続します。

アクティブ RFS パルスの GPIO 割り込みによりウインドウの開始が通知されたとき、SPORT 受信クロックの追跡に使用するタイマをイネーブルします。TDV 信号変化の GPIO 割り込みが発生したとき、TIMERx_COUNTER レジスタをチェックして、フレームの開始から経過した SPORT クロック数を求めます。この値が期待値を 1 チャンネル分だけ超えた場合、エラーの発生を意味し、SPORT の再起動とテストの繰り返しが必要です。起動条件が検出されない場合は、GPIO 割り込みもディスエーブルする必要があります。

適用レビジョン:

0.3、0.4、0.5