ANALOG DEVICES

Blackfin® 組込みプロセッサ

シリコン・アノーマリ・リスト　ADSP-BF531/BF532/BF533

ADSP-BF531/BF532/BF533 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-BF531/BF532/BF533 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-BF531/BF532/BF533 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>
0.6	0x0006
0.5	0x0005
0.4	0x0003*
0.3	0x0003

*アノーマリ 05000234 参照

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
09/18/2008	E	F	Added Anomalies - 05000425, 05000426 Revised Anomalies - 05000283, 05000315
06/18/2008	D	F	Added Silicon Revision 0.6 Added Anomalies – 05000416
02/08/2008	C	E	Added Anomalies - 05000363, 05000400, 05000402, 05000403
12/10/2007	B	E	Added Anomalies - 05000366, 05000371
09/04/2007	A	E	Initial Consolidated Revision - Replaces anomaly lists for ADSP-BF531 (Rev W), ADSP-BF532 (Rev AB) and ADSP-BF533 (Rev X) Added Anomalies – 05000357 Revised Anomalies - 05000311

NR003532E

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-BF531/BF532/BF533 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。

No.	ID	説明	0.3	0.4	0.5	0.6
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	x
2	05000099	UART ライン・ステータス・レジスタ(UART_LSR)ビットが同時に更新されない:	x	x	.	.
3	05000105	ウォッチポイント・ステータス・レジスタ(WPSTAT)ビットが各対応するマッチでセットされます:	x	x	x	x
4	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	x
5	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	x
6	05000158	命令 DMA がデータ・キャッシュのフィルに失敗することがある(ブートの意味):	x	x	.	.
7	05000166	8～16 の PPI データ長が上位ビットをゼロにしない:	x	x	x	x
8	05000167	外部フレーム同期がアクティブの間に SPORT をターンオンさせるとデータが破壊する:	x	x	x	x
9	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	.	.
10	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	x
11	05000183	外部フレーム同期を使う PPI TX モードでのタイマ・ピンの制約:	x	.	.	.
12	05000189	投機的フェッチ・キャンセル時の偽保護例外:	x	.	.	.
13	05000193	極性設定を変更した際のエッジ検出入力で偽 I/O ピン割り込みが発生する:	x	.	.	.
14	05000194	特定のモードで SPORT を再起動するとデータ破壊が発生する:	x	.	.	.
15	05000198	先行するメモリ読み出しがストールした際に MMR アクセスが失敗する:	x	x	.	.
16	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	.	.	.
17	05000200	非アクティブ・チャンネル中のある条件で SPORT TFS と DT の駆動が正しくない:	x	x	.	.
18	05000201	SPORT マルチチャンネル・モードのアクティブ・フレームで受信フレーム同期が無視されない:	x	.	.	.
19	05000202	特定のデュアル DAG 状況で無限ストールの可能性がある:	x	x	.	.
20	05000203	DMA エラーまたは DMA 停止を発生させる特定のシーケンス:	x	.	.	.
21	05000204	キャッシュモードがライトスルーの"キャッシュ・ラインをリード時にのみ割り当て"のとき読み出しデータが不正:	x	.	.	.
22	05000207	停電状態からの回復:	x	.	.	.
23	05000208	PLL_STAT レジスタの VSTAT ステータス・ビットが機能しない:	x	x	x	x
24	05000209	演算ユニット内のスピード・パスが特定の命令に影響を与える:	x	.	.	.
25	05000215	UART TX 割り込みが誤ってマスクされる:	x	x	.	.
26	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	x
27	05000225	UART スタート・ビットが不正なパルス幅になる:	x	x	.	.
28	05000227	スクラッチパッド・メモリ・バンクを読み出すと、不正なデータが返される:	x	x	.	.
29	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	x
30	05000230	UART レシーバがある条件でボーレート差に対して弱くなる:	x	x	.	.
31	05000231	UART STB ビットがレシーバ設定値に悪影響を与える:	x	x	.	.
32	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	.
33	05000234	DSPID レジスタのレビジョン番号が不正:	.	x	.	.
34	05000242	PLL_CTL レジスタの DF ビットがハードウェア・リセットに応答しない:	x	x	.	.
35	05000244	I キャッシュがオンの場合、コントロールの変更付近で CSYNC/SSYNC/IDLE により不具合が発生する	x	x	.	.
36	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	x
37	05000246	データ CPLB が偽ハードウェア・エラーを防止しない	x	x	.	.
38	05000250	マルチチャンネル(TDM)モードのある条件でデータ・ワードのビット・シフトが正しくない	.	x	.	.
39	05000253	タイマの最大外部クロック速度	x	x	.	.
40	05000254	外部クロックを使ったシングルショット PWM 出力モードのパルス幅が正しくない	.	.	x	x
41	05000255	RTC Seconds 割り込みが機能しないでハイバネート状態になる	x	x	.	.
42	05000257	短いハードウェア・ループ時の割り込み/例外により不正な命令がフェッチされることがある	x	x	.	.
43	05000258	ICPLB データ・レジスタのビット 9 とビット 12 が異なるとき、命令キャッシュが壊れる	x	x	.	.
44	05000260	ICPLB_ステータス MMR レジスタが壊れることがある	x	x	.	.
45	05000261	DCPLB FAULT ADDR MMR レジスタが壊れる	x	x	.	.
46	05000262	データ・キャッシュへのストアが失われることがある	x	x	.	.
47	05000263	ICPLB 例外を受理するとハードウェア・ループが壊れる	x	x	.	.
48	05000264	ハードウェア・ループ内の最後から 2 番目の命令で CSYNC/SSYNC/IDLE により無限ストールが発生する	x	x	.	.
49	05000265	外部 SPORT TX と RX クロックの低速な入力エッジ・レートでノイズに対して弱くなる	x	x	x	x
50	05000269	活発な I/O 動作により、内部電圧レギュレータ出力電圧(Vddint)が上昇する	x	x	.	.
51	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	x	.	.
52	05000271	内部電圧レギュレータの自発的リセット	x	.	.	.

No.	ID	説明	0.3	0.4	0.5	0.6
53	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	x
54	05000273	同期 SDRAM メモリへの書き込みが失われる	x	x	x	.
55	05000276	非ゼロ PPI_DELAY を持つ外部フレーム同期 PPI モードに対するタイミング条件の変更	x	x	x	x
56	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	x	x	.
57	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	x	x	.
58	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	x	x	.
59	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	x	x	.
60	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	x	x	.
61	05000288	FIFO が満杯になると SPORT が不正なデータを受信する	x	x	x	.
62	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	x	x	.
63	05000302	DMA MMR レジスタに対する書き込みの後の SSYNC が正しく処理されない	x	x	.	.
64	05000305	PLL_CTL レジスタの SPORT_HYS ビットが機能しない	x	x	.	.
65	05000306	PPI_コントロール・レジスタの ALT_TIMING ビットが機能しない	x	x	.	.
66	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	x
67	05000311	特定のシーケンスでフラグ(GPIO)ピンが誤動作する	x	x	x	.
68	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	x	.
69	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	x	x	.
70	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	x	x	.
71	05000319	LQFP パッケージで内部電圧レギュレータ値 1.05V、1.10V、1.15V が使用できない	x	x	x	.
72	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	x	.
73	05000363	UART ブレーク信号の問題	x	x	.	.
74	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	x
75	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	x	.
76	05000400	特別なモードで PPI が正常に起動されない	.	.	x	.
77	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 によりストールされる	.	.	x	.
78	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	x
79	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	x
80	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	x
81	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	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-BF531/BF532/BF533 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
```

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;
```

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。VisualDSP++ランタイム・ライブラリはサポートしてない命令を使用しません。また、このアノーマリの影響を受けるデバイスとシリコン・レビジョンをターゲットとするとき、VisualDSP++ Blackfin コンパイラはサポートしてない命令を生成しません。

適用レビジョン:

0.3、0.4、0.5、0.6

2. 05000099 - UART ライン・ステータス・レジスタ(UART_LSR)ビットが同時に更新されない:

説明:

UART ライン・ステータス・レジスタ(UART_LSR)ビットが同時に更新されません。UART_LSR レジスタを直接ポーリングすると、見落とすライン・ステータス状態があります。

対策:

ポーリング・モードを使用する場合、UART_LSR と UART_RBR の読み出し、および UART_THR の書き込みを安全に行うタイミングを調べるために SIC_ISR レジスタをポーリングする必要があります。受信データがレディになるタイミングと受信エラーの有無を調べるためのコーディング例を次に示します。ポーリング・モードに対して、UART_TX、UART_RX、UART エラーの各割り込みは SIC_IMASK の中でマスクされています(割り込みの発生なし)。

```
#define IRQ_UART_RX 0x4000
#define IRQ_UART_ERROR 0x40

  p0.l = lo(UART_GCTL);
  p0.h = hi(UART_GCTL);

  p2.l = lo(SIC_ISR);
  p2.h = hi(SIC_ISR);

  r1 = PEN | WLS(8) (z);
  w[p0+UART_LCR-UART_GCTL] = r1;

  r1 = ERBFI | ELSI (z);
  w[p0+UART_IER-UART_GCTL] = r1;

  receive_polling:
  r2 = w[p2] (z);
  CC = bittst (r2, bitpos (IRQ_UART_RX));
  if !CC jump receive_polling;

data_ready: csync;
  r1 = w[p0+UART_LSR-UART_GCTL] (z);
  r0 = w[p0+UART_RBR-UART_GCTL] (z);

  CC = bittst (r2, bitpos (IRQ_UART_ERROR));
  if CC jump error_handler;

  [i0++] = r0;
  jump receive_polling;
```

適用レビジョン:

0.3、0.4

3. 05000105 - ウォッチポイント・ステータス・レジスタ(WPSTAT)ビットが各対応するマッチでセットされます:

説明:

ウォッチポイント・データ・アドレス・カウンタ(WPDACTL:WPDCNTENx)がイネーブルの場合でも、対応するウォッチポイント・ステータス・レジスタ・ビット(WPSTAT:STATDAx)は、すべての条件一致時にセットされます。カウンタの満了時ではありません。

ウォッチポイント命令アドレス・カウンタ(WPIACTL:WPICNTENx)とステータス・ビット(WPSTAT:STATIAx)にも同じことがいえます。

対策:

ウオッチポイント割り込みが発生したとき、WPSTAT ビットのセットを確認すると共に、それらのカウンタ・イネーブル・ビットとカウンタ・レジスタ値(WPIACNTn または WPDACTn)を確認する必要があります。

注: カウンタ・レジスタは、0x0000 にデクリメントしていくので、カウンタが満了し、かつ、その満了からさらに 1 回の条件一致が検出された時に、その値が 0x0000 になります。

適用レビジョン:

0.3、0.4、0.5、0.6

4. 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、0.6

5. 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。たとえば(P0 が 16 ビット・システム MMR を指す場合)、次のアクセスは失敗します:

```
w[P0] = R5.H;
```

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。たとえば(P0 が 16 ビット・システム MMR を指す場合):

```
w[p0] = r5.l;
r4.l = w[p0];
r3 = w[p0](z);
w[p0] = r3;
```

VisualDSP++ Blackfin コンパイラは通常、コードの生成時に問題の命令を発行しません。定数アドレスを用いて MMR ロードを行うケース(たとえば*MMR_Reg = value)では、レジスタ・ハーフをスワップするパック命令を挿入します。しかし、コンパイル時に未知のポインタ(たとえば関数に対するパラメータ)を MMR に対するポインタとして識別することはできません。VisualDSP++ランタイム・ライブラリもこのアノーマリを回避します。

適用レビジョン:

0.3、0.4、0.5、0.6

6. 05000158 - 命令 DMA がデータ・キャッシュのフィルに失敗することがある(ブートの意味):

説明:

DMA またはコア MMR DTEST レジスタ・アクセスが L1 命令メモリに対して発生した後、対応するポートに対するデータ・キャッシュ・フィルでデータ破壊が生ずることがあります。この状況は、コアがデータ・キャッシュ・フィルと同じデータ・メモリ・バンクをアクセスしているためにストールが発生した場合にのみ発生します。

このデータ・キャッシュ・フィルは、L1 命令メモリ DMA または MMR アクセス後の多くのサイクルで発生します。不具合の一例は、データ・キャッシュをイネーブルするプログラムがブートする場合です。ブート時に発生する命令 DMA は、メイン・プログラムの実行中のある時点で発生するデータ・キャッシュ・フィルによるデータ破壊を引き起こす原因となります。

ポート A の場合、データ・ロケーション 0xFF80xxxx と命令ロケーション 0xFFA00000～0xFFA07FFF (該当する場合)で不具合が発生します。ポート B の場合、データ・ロケーション 0xFF90xxxx と命令ロケーション 0xFFA08000～0xFFA0FFFF (該当する場合)で不具合が発生します。

低い優先順位のデータ・キャッシュ・フィルが進行中に、直前の DMA トランザクションが L1 命令メモリに対するものである場合にも、この問題が発生します。

対策:

この問題に対する最善の対策は、使用している DCPLB データ・レジスタのビット 9 をセットすることです。このビットはハードウェア・リファレンス・マニュアルでは予約済みビットと表示されていますが、このビットの機能を記載するようにドキュメントの変更中です。

次善の対策は、プログラムの開始時にソフトウェア・コア・リセットを実行することです:

```
P0.H = HI(SYSCR);              // Check System Reset Configuration Register
P0.L = LO(SYSCR);
R0.L = W[P0];
CC = BITTST(R0,4);             // Check NO BOOT ON SOFTWARE RESET Bit
IF CC JUMP _No_Boot_Set;
BITSET(R0,4);                  // Set NO BOOT ON SOFTWARE RESET Bit
W[P0] = R0;
SSYNC;                         // Ensure write completes before executing
RAISE RAISE 1;

  _No_Boot_Set:
BITCLR(R0,4);                  // Reset SYSCR to original value
W[P0] = R0;
```

プログラムがデータ・キャッシングと L1 命令メモリ DMA を実行中の場合は、命令 DMA を実行した後に命令 SRAM とデータ・ポートを共用するデータ SRAM に対してデータ DMA を実行してください。命令 DMA が開始した時点からデータ DMA が完了するまでの間、データ・キャッシングやビクティム処理を引き起こすアクセスを行ってはいけません。また、元の問題が命令メモリ DMA から発生した場合には、対応するデータ・バンクに対する MMR DTEST アクセスもこの問題を解消します。

プログラムがコア MMR DTEST レジスタを経由して L1 命令メモリをアクセスする場合、このアクセスの後に対応するデータ・バンクに対するコア MMR DTEST レジスタ・アクセスを続けてください。

プロセッサが命令メモリへ DMA を行わない場合またはデータ・キャッシュを使用しない場合は、この問題に遭遇しません。

VisualDSP++ランタイム・ライブラリのキャッシュ・サポート機能は、必要な場合このアノーマリに対する対策を含んでいます。

適用レビジョン:

0.3、0.4

7. 05000166 - 8～16 の PPI データ長が上位ビットをゼロにしない:

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。たとえば、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、0.6

8. 05000167 - 外部フレーム同期がアクティブの間に SPORT をターンオンさせるとデータが破壊する:

説明:

SPORT は、外部フレーム同期をレベル検出します。SPORT を外部フレーム同期に設定し、かつ SPORT を最初にイネーブルしたときフレーム同期がアクティブの場合、イネーブルされると直ちに SPORT はデータ受信を開始します。これがフレームの中程で発生して、不正なデータが受信されてしまいます。

このアノーマリはステレオ・シリアル・モード(I2S およびその派生)にも適用されます。ただし、LRFS ビットまたは RRFST ビット(いずれか一方のみ)がセットされている場合は除きます。

対策:

SPORT が完全にイネーブルされるまで、外部フレーム同期を遅延させてください。SPORT がイネーブルされるまで遅延させることができない外部フレーム同期を持つシリアル・デバイスを使用している場合は、プログラマブル・フラグ・ピンをフレーム同期に接続することができます。SPORT フレーム同期を連続的にサンプルするように PFx ピンを設定して、RFS / TFS 信号が非アクティブ状態になったときに SPORT をイネーブルします。

ステレオ・シリアル・モードの場合、可能なとき LRFS ビットまたは RRFST ビット(いずれか一方のみ)をセットしてください。

適用レビジョン:

0.3、0.4、0.5、0.6

9. 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

10. 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、0.6

11. 05000183 - 外部フレーム同期を使う PPI TX モードでのタイマ・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:

このアノーマリは、" 2 個の外部フレーム同期を使う PPI 送信"モードにのみ適用され、TMR2 ピンのみが影響を受けます。タイマ 2 をイネーブルする必要があるため、汎用として使うことはできません。

適用レビジョン:

0.3

12. 05000189 - 投機的フェッチ・キャンセル時の偽保護例外:

説明:

投機的フェッチのキャンセルにより偽コードまたは偽データが発生した場合、これらの保護例外が発生します。たとえば、有効ページの最後のワードにあるジャンプ命令または rts 命令が別の有効ページへ分岐し、かつ無効または存在しないメモリ・ロケーションから投機的命令フェッチが行われている場合に、例外が発生します。同様に、ポスト・インクリメントの間接データ・メモリ・アクセスで、保護された、または存在しないメモリ・ロケーション(たとえば、R0 = [P0++]、ここで P0 はページの最終ワードを指します)に対して投機的アクセスが行われた場合も、不正な例外が発生します。

対策:

1)ページ境界に分岐命令またはデータを配置しないでください。境界の前では、予約済みメモリ空間として少なくとも 76 バイトを空けてください。これにより偽例外の発生を防止することができます。

2)例外が有効か否かをアクションの前に判断する例外ハンドラを設けてください。これは、CODE_FAULT_ADDR (または DATA_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。

デフォルトの VisualDSP++ LDF には、このハードウェア・アノーマリに対する対策が含まれています。該当するシリコン・レビジョンでは対策が自動的にイネーブルされます。あるいは、リンク時にマクロ __WORKAROUND_AVOID_LDF_BLOCK_BOUNDARIES を定義して、対策を手動でイネーブルすることができます。

イネーブルされると、LDF は有効なメモリ・ブロックの境界に 76 バイトを予約します。

適用レビジョン:

0.3

13. 05000193 - 極性設定を変更した際のエッジ検出入力で偽 I/O ピン割り込みが発生する:

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)、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

14. 05000194 - 特定のモードで SPORT を再起動するとデータ破壊が発生する:

説明:

立ち下がりエッジを選択して内部 SPORT クロックまたは外部 SPORT クロックを使用する場合(SPORTx_TCR1:TCKE、SPORTx_RCR1:RCKE)、SPORT をディスエーブルした後に再イネーブルすると、最初に送信または受信されたワードが壊れる可能性があります。

対策:

内部 SPORT クロックの場合: SPORT をディスエーブルした後に SPORT コンフィギュレーション 1 レジスタ(SPORTx_TCR1:ITCLK、SPORTx_RCR1:IRCLK)のビット 1 に 0 を書き込みます。これにより、SPORT クロックが外部に切り替えられて、SPORT を完全にリセットできるようになります。

立ち下がりエッジを選択した外部 SPORT クロックの場合: SPORT をディスエーブルした後に、SPORT コンフィギュレーション 1 レジスタ(SPORTx_TCR1:TCKE、SPORTx_RCR1:RCKE)のビット 14 に 0 を書き込みます。これにより、SPORT が完全にリセットできるようになります。

適用レビジョン:

0.3

15. 05000198 - 先行するメモリ読み出しがストールした際に MMR アクセスが失敗する:

説明:

メモリ読み出しのストール後に MMR 読み出しが続くと、MMR 読み出しが失敗します(不正なデータが読み出されます)。同様に、メモリ読み出しのストール後に MMR 書き込みが続くと、書き込みが実行されません(失われます)。メモリ読み出しを含む命令は、

```
reg = [ Preg ], etc.
reg = [ Ireg ], etc.
stack pop, stack pop multiple
UNLINK
TESTSET
PREFETCH
```

および並行発行の上記すべて。メモリ書き込みを含む命令は、

```
[ Preg ] = reg, etc.
[ Ireg ] = reg, etc.
stack push, stack push multiple
LINK
```

および並行発行の上記すべて。

対策:

MMR アクセスの前に NOP (または任意の非メモリ・アクセス)を配置すると、この問題が防止されます。

VisualDSP++Blackfin コンパイラには、このハードウェア・アノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ'-workaround sdram-mmr-read'を指定することにより、対策を手動でイネーブルすることができます。

イネーブルされると、コンパイラは、ロードと MMR ロードとの間に NOP 命令を挿入します。コンパイラは、定数アドレスで MMR ロードを行う場合(たとえば*MMR_ADDR = value)は NOP 命令を挿入しますが、コンパイル時に未知のポインタ(たとえば関数へのパラメータ)を MMR へのポインタとして識別できません。

対策をイネーブルすると、__WORKAROUND_SDRAM_MMR_READ が、コンパイル、アセンブル、リンク・ステージで定義されます。

適用レビジョン:

0.3、0.4

16. 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。

2) あるいは 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:

17. 05000200 - 非アクティブ・チャンネル中のある条件で SPORT TFS と DT の駆動が正しくない:

説明:
マルチチャンネル・モードでは、SPORT の MRCS レジスタを使って、アクティブにするチャンネル(送信または受信する)と無視するチャンネルを選択します。各無視されるチャンネルでは、DT 出力がスリー・ステートになります。"マルチチャンネル DMA パッキング"がディスエーブルされたとき、問題が発生します。このモードでは、非アクティブ・チャンネルの場合、データ・ワードの上位ビット(MSB)が DT ラインに出力され、これに対応して TFS が 1 ビットの継続時間ハイ・レベルに駆動されて有効データを表示します。

対策:
可能な対策は、"マルチチャンネル DMA パッキング"をイネーブルすることです。このモードでは、アクティブ・チャンネルだけでメモリとの間で DMA が実行され、非アクティブ・チャンネルは実質的に無効にされますが、このモードでは、非パックド DMA モードのようにアクティブ/非アクティブ・チャンネルをダイナミックに変更することはできません。したがって、これはすべてのアプリケーションで可能ではありません。

適用レビジョン:
0.3、0.4

18. 05000201 - SPORT マルチチャンネル・モードのアクティブ・フレームで受信フレーム同期が無視されない:

説明:
アクティブ・フレームで外部 RFS が無視されません。これにより、データ転送が停止することがあります。RFSDIV がウインドウ・サイズより小さい値に設定されると、同じ問題は内部フレーム同期でも発生します。

対策:
カレント・フレームがアクティブのとき、外部フレーム同期を回避してください。RFSDIV をウインドウ・サイズより小さい値に設定しないでください。

適用レビジョン:
0.3

19. 05000202 - 特定のデュアル DAG 状況で無限ストールの可能性がある:

説明:
この問題が発生するためには、プロセッサが非キャッシャブルまたはライトスルー・キャッシャブルの L2 または書き込まれたばかりの外部メモリ・アドレスに対するデータ・メモリ読み出しを行う必要があります。"最近の書き込み"は、再度読み出されたとき現に書き込みバッファに保持されている必要があります。この読み出しは、この"最近の書き込み"が書き込みバッファからディステネーション・メモリ・ロケーションへ排出された同じクロック・サイクルで実行される必要があります。
"最近の書き込み"の読み出しの直前に、デュアル DAG アクセスを実行する必要があります。デュアル DAG アクセスは互いに衝突する必要がありますが、"最近の書き込み"アドレスと何かの関係があると考えられます。
"最近の書き込み"の読み出しの直後に、読み出し、書き込み、またはプリフェッチを行わない必要があります(そうしないと、このアノーマリは回避されます)。

注: デュアル DAG の衝突は、両 DAG が L1 メモリまたは L1 キャッシュ内の同じサブバンクをアクセスするときに発生します。

対策:
1) デュアル DAG アクセスの衝突の直後に、最近書き込まれた L2 アドレスを読み出さないでください。
   または
2) オフェンシブなデュアル DAG/L2 読み出し命令を SSYNC の前に配置して、前の書き込みが書き込みバッファ内に存在しなくなるようにしてください。または
3) すべての L2 をライト・バック・キャッシャブルに設定してください(書き込みバッファの使用を回避)。

VisualDSP++Blackfin コンパイラには、このハードウェア・アノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ'-workaround infinite-stall-202'を指定することにより、対策を手動でイネーブルすることができます。イネーブルされると、コンパイラは、PREFETCH[SP]命令を挿入してアノーマリ状態を回避します。
対策をイネーブルすると、__WORKAROUND_INFINITE_STALL_202 が、コンパイル、アセンブル、リンク・ステージで定義されます。

適用レビジョン:
0.3、0.4

20. 05000203 - DMA エラーまたは DMA 停止を発生させる特定のシーケンス:

説明:

1 つの L1 メモリ・バンク(バンク A またはバンク B)からの 3 回以上の連続 DMA 読み出しが長時間ストールすると(コアまたはキャッシュ動作に起因)、問題が発生します。この状況では、後続の L1 DMA 読み出しが別のバンクから開始されると、直前の 2 回の読み出しのデータが衝突して、データが壊れる可能性があります。

対策:

この問題には次の 2 つの対策があります:

1)同じ L1 メモリ・バンク(たとえばバンク A またはバンク B)内にすべての DMA データを配置します。

2)任意の時間に任意の L1 メモリ・バンクから 1 つの DMA チャンネルのみが読み出すようにします。

適用レビジョン:

0.3

21. 05000204 - キャッシュ・モードがライトスルーの"キャッシュ・ラインをリード時のみ割り当て"のとき読み出しデータが不正:

説明:

ライトスルー・キャッシュがイネーブルされ、かつ DCPLB_DATAx:CPLB_L1_AOW = 0 (キャッシュ・ラインをリード時のみ割り当て)のとき、次のシナリオで不正なデータが読み出されることがあります。

- 書き込み割り当てなしのモードでライトスルー・キャッシャブルであるアドレスに対して書き込むと、キャッシュ・ミスが発生する。
- 書き込みがまだストア・バッファ内である間に(書き込みバッファまたはディステネーション・メモリ・ロケーションに入る前に)上記アドレスが読み出された。
- その後、ストア・バッファから取り出した後に再度上記アドレスを読み出した。キャッシュから返されたデータは不正であるが、ディステネーション・メモリ・ロケーションの値は正しい。

対策:

データ・キャッシュをライトスルーに設定するとき、DCPLB_DATAx:CPLB_L1_AOW = 1 (キャッシュ・ラインを読み出しと書き込みに割り当て)を設定して、このアノーマリの発生を防止してください。

適用レビジョン:

0.3

22. 05000207 - "停電"状態からの回復:

説明:

"停電"が発生すると、ハードウェア・リセット・ピンを使って内部電圧レギュレータをリセットすることができません。"停電"とは、VDDext がデータシートで規定された範囲を下回るが、正しい値に戻る前に 0 V まで低下しない状態と定義されます。

対策:

"停電"から回復するためには、プロセッサが完全にパワーダウンして、電源が戻る必要があります。

適用レビジョン:

0.3

23. 05000208 - PLL_STAT レジスタの VSTAT ステータス・ビットが機能しない:

説明:

PLL_STAT レジスタの VSTAT ステータス・ビットが機能しません。内部電圧レギュレータが安定したか否かを調べるとき、この値に依存することは推奨されません。

対策:

内部電圧レギュレータを介して電圧を変えるときは、電圧が変化するために少なくとも 40usec 待ってください。40usec 後に、VSTAT ビットの状態に関係なく新しい値が設定されます。

適用レビジョン:

0.3、0.4、0.5、0.6

24. 05000209 - 演算ユニット内のスピード・パスが特定の命令に影響を与える:

説明:

次の命令では、前の命令がその命令のオペランドを生成するとき正しくない動作が発生することがあります。影響を受ける命令は:

```
EXTRACT (x)
DEPOSIT (x)
SIGNBITS
EXPADJ
```

Signbits 命令に対して例を示します:

```
r0 = ashift r2 by r3.l;
r1.l = signbits r0;
```

対策:

この問題の回避には次の 2 つの対策があります:

1) signbits 命令の前に nop を配置します:

```
r0 = ashift r2 by r3.l;
nop;
r1.l = signbits r0;
```

2) signbits のオペランド・レジスタが前の命令に依存していないことを確認します:

```
r0 = ashift r2 by r3.l;
// ** another useful instruction that is not updating r0 **;
r1.l = signbits r0;
```

VisualDSP++Blackfin コンパイラには、このハードウェア・アノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ'-workaround dreg-comp-latency'を指定することにより、対策を手動でイネーブルすることができます。VisualDSP++ランタイム・ライブラリも、必要に応じてこのアノーマリ状態を回避します。イネーブルされると、コンパイラは 2 つの命令の間に NOP 命令を挿入します。この最初の命令では値を DREG に割り当て、2 つ目の命令では DREG を SIGNBITS、EXTRACT、DEPOSIT または EXPADJ の各命令に対するパラメータとして使います。

この対策がイネーブルされると、コンパイラもマクロ __WORKAROUND_DREG_COMP_LATENCY と __WORKAROUNDS_ENABLED をソース、アセンブリ、リンク・ビルド・ステージで定義します。

適用レビジョン:

0.3

25. 05000215 - UART TX 割り込みが誤ってマスクされる:

説明:

UART TX 割り込みで、IIR レジスタが読み出され、かつ THR レジスタが書き込まれないと(TX 割り込みをクリアするため)、UART TX 割り込みがマスクされます。ストリングの終わりに到達した場合に、ISR 内でこれが発生することがあります。この場合、ETBEI ビットをイネーブル/ディスエーブルしても、割り込みイネーブルの状態は影響を受けませんが、これを行う必要があります。

対策:

UART TX 割り込み内で ETBEI ビットをクリアしてストリングを終わらせて、RTI を実行してください。割り込みを再イネーブルするときは、ETBEI ビットをセットするだけで済みます。これにより、THR レジスタがエンプティのとき、プロセッサが UART TX 割り込みサービス・ルーチン内へ直接進みます。

適用レビジョン:

0.3、0.4

26. 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、0.6

27. 05000225 - UART スタート・ビットが不正なパルス幅になる:

説明:

UART インターフェースから送信されたワードのスタート・ビットの幅が不正になります。

1 より大きいクロック分周比に対して(UART_DLL レジスタと UART_DLH レジスタにより指定)、パルス幅は公称ビット時間の 14/16 または 15/16 の値と見なすことができます。データ、パリティ、ストップの各ビットは正しい継続時間になります。データは正しく受信されます。UART のタイミングについては、アノーマリ 05000230 と 05000231 を参照してください。

対策:

なし

適用レビジョン:

0.3、0.4

28. 05000227 - スクラッチパッド・メモリ・バンクを読み出すと、不正なデータが返される:

説明:

スクラッチパッド・メモリを読み出すと、ある条件下で不正なデータが返されます。スクラッチパッド・メモリの読み出しの直後にさらに読み出し(非スクラッチパッド・ロケーションを含む任意ロケーション)が続く場合で、かつアクセスされるアドレスが同じ下位アドレス・ビットを持たない場合に、この問題が発生します。これは、転送の 1 つがバイト・アクセスである必要があることを意味しています。他のアクセスは、最初のアクセスとは異なるバイト境界でのバイト、16 ビット、または 32 ビット・アクセスである必要があります。さらに、スクラッチパッド・メモリ読み出しの直前の命令は、2 個の DAG バンクの衝突、非 L1 メモリ・データのフェッチ、またはキャッシュ・ライン・フィルに起因するメモリ・ストールを発生する必要があります。

命令がスクラッチパッド読み出しの直後に読み出しを行わない場合は、問題は発生しません。また、連続する非バイト読み出しも正常に機能します。

対策:

アセンブリ・プログラマにとって最もシンプルな対策は、各スクラッチパッド読み出しの後ろに非読み出し命令を配置することです。C プログラマにとって 1 つのソリューションは、スクラッチパッドにデータをマップしないことです。

VisualDSP++Blackfin コンパイラには、このハードウェア・アノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ' -workaround scratchpad-read'を指定することにより、対策を手動でイネーブルすることができます。対策をイネーブルして、(2)と(3)の少なくとも一方がバイト・ロードである 3 個のロード命令シーケンスが発生すると(またはマルチ発行命令の一部として発生すると)、nop が(1)と(2)の間または(2)と(3)の間に挿入されます:

```
A load instruction (1);
A load instruction (2);
A load instruction (3);
```

マクロ__WORKAROUND_SCRATCHPAD_READ が、コンパイル、アセンブル、リンク・ステージで定義されます。

VisualDSP++ランタイム・ライブラリも、該当するシリコン・レビジョンとデバイス番号に対してこのアノーマリ状態を回避します。

適用レビジョン:

0.3、0.4

29. 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、0.6

30. 05000230 - UART レシーバがある条件でボーレート差に対して弱くなる:

説明:

同期通信で、トランスミッタとレシーバのビット・クロックが公称値から一定パーセント値だけ異なることがあります。正確な差は、送信ワード構成と信号品質のようなその他の外部ファクタに依存します。

Blackfin UART レシーバでは、ビット・クロックが送信側クロックより低速または高速である場合、偏差が異なります。Blackfin UART レシーバは、送信側が少し低いビット・レートで動作する場合、その差に良く耐えますが、送信側が Blackfin レシーバより少し高いビット・レートで動作する場合は、連続転送(ワード間にギャップがない場合)で通信に失敗することがあります。

この理由は、標準インプリメンテーションの場合、レシーバは他のビットと同様に、ストップ・ビットを 16 回サンプルした後に、新しいスタート・ビット条件を検出するためです。ストップ・ビットが検出されたか否かの判定は 9 番目のサンプルの後に行われます。ストップ・ビットが検出されると、レシーバは直ちに次のワードを読み出すことができるべきですが、Blackfin レシーバでは残りの 7 サンプルも使用してしまいます。

これによる影響は、サンプリング・エラーが上記のようなデータ転送で蓄積することです。

シングル転送またはワード間にギャップがある転送ではサンプリング・エラーが蓄積されないため、このアノーマリの影響ありません。

対策:

送信側は、低い(または等しい)ビット・レートで動作する必要があります。

これを保証できない(または不可能な)場合は、送信側を 1 ストップ・ビットより長く送信するように設定し、ワード間に必要なギャップを挿入してください。

双方向通信チャンネルでトランスミッタとレシーバが Blackfin デバイスである場合は、上記対策でこの問題を解決できません。これについては、アノーマリ 05000225 と 05000231 を参照してください。

適用レビジョン:

0.3、0.4

31. 05000231 - UART STB ビットがレシーバ設定値に悪影響を与える:

説明:

STB ビットは、トランスミッタで発生するストップ・ビット数を制御しますが、この設定値も、レシーバでサンプルするストップ・ビット数に悪影響を与えます。正しい動作では、レシーバが常に 1 ストップ・ビットをサンプルしてテストしますが、レシーバは、STB ビットで設定されるストップ・ビット数をサンプルしてテストします。この不正な動作は、フレーム・エラー検出にも影響を与えます。

このアノーマリは、アノーマリ 05000230 に対する対策を、2 個の Blackfin デバイスで構成される双方向リンクのケースに適用できなくすることに注意してください。

対策:

なし

適用レビジョン:

0.3、0.4

32. 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

33. 05000234 - DSPID レジスタのレビジョン番号が不正:

説明:

DSPID レジスタのシリコン・レビジョン情報が誤っています。

対策:

REVID レジスタ(アドレス 0xFFC0 0014)の上位 4 ビットは、シリコン・レビジョン情報を取得するときに読み出すことができます。このロケーションの残りのビットは予約済みです。

適用レビジョン:

0.4

34. 05000242 - PLL_CTL レジスタの DF ビットがハードウェア・リセットに応答しない:

説明:

ハードウェア・リセットの前に DF ビットをセットすると、ハードウェア・リセットの後に PLL は CLKIN の 2 分周を続けますが、PLL_CTL レジスタの DF ビット自体はクリアされます。

対策:

リセット後に CLKIN の 2 分周が不要な場合には、DF をクリアして PLL を再設定してください。

適用レビジョン:

0.3、0.4

35. 05000244 - I キャッシュがオンの場合、コントロールの変更付近で CSYNC/SSYNC/IDLE により不具合が発生する:

説明:

命令キャッシュをイネーブルすると、コントロールの変更付近(非同期の例外/割り込みも含む)で CSYNC/SSYNC/IDLE により予期しない結果が発生することがあります。

この問題を発生する最も一般的なシーケンスの例は、BRCC (NP)とそれに続く次の 3 つの命令内の何処かに CSYNC/SSYNC/IDLE がある場合です:

```
BRCC X [predicted not taken]
nop
nop CSYNC/SSYNC/IDLE // this instruction is bad in any of the 3 instructions following BRCC X
```

この問題に遭遇するもう 1 つのシーケンスは、CSYNC/SSYNC/IDLE を指す BRCC (BP)の後ろに、投機的にフェッチされた CSYNC/SYNC/IDLE が 2 サイクル以内に BRCC に"追いつく"ことを可能にする命令のストールが続く場合です:

```
BRCC X (bp)
Y: ...
   ...
X: CSYNC/SSYNC/IDLE
```

このシーケンスでは、不具合の再現が極めて困難です。BRCC の前のストールと非常に特殊なキャッシュ動作との正確な組み合わせが必要です。

対策:

命令キャッシュをオフにすることが、このアノーマリを回避する 1 つの方法です。

アセンブリ・コードでは上記シナリオを回避する必要があります。VisualDSP++ Blackfin アセンブラは、CSYNC/SSYNC/IDLE 命令がこのアノーマリ状態が発生しそうな場合に警告 ea5507 を発生します。

非同期イベントに無関係なすべてのケースに対しては、VisualDSP++ Blackfin コンパイラが該当するシリコン・レビジョンとデバイス番号に対して自動的にイネーブルされる対策を含んでいます。この対策は、-workaround speculative-syncs コンパイラ・フラグを使って手動でイネーブルすることもできます。イネーブルすると、__WORKAROUND_SPECULATIVE_SYNCS マクロが、コンパイル、アセンブル、リンクのビルド・ステージで定義されます。この対策では NOP を挿入して、次の形式のアノーマリに対してアノーマリ状態を回避します:

```
IF CC JUMP ...;
CSYNC/SSYNC/IDLE
CSYNC/SSYNC/IDLE
```

```
IF CC JUMP X (BP);
...
...
X: CSYNC/SSYNC/IDLE
```

```
LSETUP(LT, LB) CSYNC/SSYNC/IDLE
     LT: CSYNC/SSYNC/IDLE
         CSYNC/SSYNC/IDLE
         IF CC JUMP X:
     LB: NOP;
X: ...
```

VisualDSP++ランタイム・ライブラリも、該当するシリコン・レビジョンとデバイス番号に対してこのアノーマリ状態を回避します。

非同期割り込みイベントの場合、割り込みをディスエーブルし、SSYNC/CSYNC/IDLE の前に 2 個の NOP を詰めることにより SSYNC/CSYNC/IDLE 命令を保護することができます:

```
CLI R0;
NOP; NOP;                    // 2 Padding Instructions
CSYNC/SSYNC/IDLE
STI R0;
```

例外に対しては、メモリのキャッシャブル領域に対する任意のアクセスの後に 3 個の NOP を詰める必要があります。

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。

適用レビジョン:

0.3、0.4

36. 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。

VisualDSP++Blackfin コンパイラには、このハードウェア・アノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ'-workaround speculative-loads'を指定することにより、対策を手動でイネーブルすることができます。

対策がイネーブルされると、コンパイラは、NOP 命令を挿入してアノーマリ状態を回避します。

対策をイネーブルすると、__WORKAROUND_SPECULATIVE_LOADS が、コンパイル、アセンブル、リンク・ビルド・フェーズで定義されます。

この対策の重複適用を回避する種々のチェックがコンパイラ内にあります。たとえば、対策を適用する前に、ロードについて次を確認します:

- SP または FP を使用してないこと(この場合スタックからであるため不正ではありません)
- 揮発性と評価されるタイプのアドレスでないこと(この場合既知の有効アドレスから)
- コンパイラに未知のアドレスからであること
- 分岐ターゲットで重複されていないこと(この場合、投機的実行可能)
- ジャンプの前に実行されないこと(この場合、投機的実行可能)

VisualDSP++ランタイム・ライブラリも、該当するシリコン・レビジョンとデバイス番号に対してこのアノーマリ状態を回避します。

適用レビジョン:

0.3、0.4、0.5、0.6

37. 05000246 - データ CPLB が偽ハードウェア・エラーを防止しない:

説明:

データ CPLB がイネーブルされたとき、予約済みまたは未定義メモリに対する投機的アクセスにより発生するハードウェア・エラーは起こらないはずなのに、発生します。

対策:

なし。投機的フェッチの一部として予約済みメモリをアクセスしないでください。

このアノーマリの影響を受けるシリコン・レビジョンをビルドするとき、またはスイッチ'-workaround no-cplb-spec-protect-246'を使用するとき、VisualDSP++ Blackfin コンパイラは、'-cplbs'スイッチの使用を無視します。逆に-cplbs を使用すると、コンパイラは投機的フェッチの発行を無視します。

適用レビジョン:

0.3、0.4

38. 05000250 - マルチチャンネル(TDM)モードのある条件でデータ・ワードのビット・シフトが正しくない:

説明:

マルチチャンネル・モードでは、フレーム同期の幅が実際のデータ・フレーム幅より 1 ビットだけ大きい場合(すなわち"非アクティブ・ビット"が 1 つある場合)、送信フレームの先頭ワードが 1 ビット左にシフトされるため、LSB が 2 番目のワードの MSB になってしまいます。他の全ワードは正常に送信されます。

フレーム同期幅が実際のフレーム幅と一致する場合、または 1 ビット大きい場合はすべてのデータが正常に送信されます。

たとえば、16 ビット・データが 8 ワードある場合、データ・フレームでは 128 ビットになります。

```
If RFSDIV = 127 --> all data words are CORRECT
If RFSDIV = 128 --> first word is INCORRECT
If RFSDIV = 129 --> all data words are CORRECT
If RFSDIV = 130 --> all data words are CORRECT
```

対策:

RFSDIV レジスタ値にデータ・ビット数+/- 1 を設定して上記ケースを回避してください。

適用レビジョン:

0.4

39. 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 まで可能としていますが、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

40. 05000254 - 外部クロックを使ったシングルショット PWM 出力モードのパルス幅が正しくない:

説明:

タイマが PWM_OUT モードで、かつシステム・クロックではなく外部クロック(PPI_CLK またはフラグ・ピンに入力される信号)で駆動され、かつシングル・パルス・モード(PERIOD_CNT = 0)である場合、発生されるパルス幅が+1 または-1 カウントだけずれることがあります。他の全モードはこのアノーマリの影響を受けません。

対策:

推奨対策は、シングルパルス・モードではなく連続モードを使用することです。タイマをイネーブルした後、直ちにディスエーブルすることができます。タイマはシングル・パルスを発生し、周期の終わりまでカウントした後に実質的に自分自身をディスエーブルします。発生される波形は所望の長さになります。

PULSEWIDTH が所望の幅である場合、次のシーケンスによりシングル・パルスが発生されます:

```
TIMERx_CONFIG = PWM_OUT|CLK_SEL|PERIOD_CNT|IRQ_ENA;   // Optional:PULSE_HI|TIN_SEL|EMU_RUN
TIMERx_PERIOD = PULSEWIDTH + 2;                       // Slightly bigger than the width
TIMERx_WIDTH = PULSEWIDTH;
TIMER_ENABLE = TIMENx;
TIMER_DISABLE = TIMDISx;
<wait for interrupt (at end of period)>
```

適用レビジョン:

0.5、0.6

41. 05000255 - RTC Seconds 割り込みが機能しないでハイバネート状態になる:

説明:

低消費電力のハイバネート状態は、内部電圧レギュレータをディスエーブルすると(Vddint = 0 Volts)、開始されます。リアルタイム・クロック(RTC)は、特定のイベントで電圧レギュレータをウェイクアップさせるように設定されます。

RTC ウェイクアップ・イベントが Seconds イベント(RTC_ICTL = 0x0004)である場合、ウェイクアップ信号が 1 秒間誤ってアクティブになります。この場合、常に電圧レギュレータは直ちにウェイクアップしてしまうため、これをディスエーブルすることはできません。

これはハイバネート状態の場合だけです。ディープ・スリープとその他の低消費電力モードは、Seconds イベントで正常に動作します。

長い周期の RTC イベントの場合、RTC からのウェイクアップ・イベントの後にプロセッサが再度ハイバネート状態を開始できるまでの最小時間は 1sec であることに注意してください。たとえば、Minute イベントの場合、プロセッサは最初の 1sec 間、ハイバネート・モードを再開できません。

対策:

可能な対策は次のステップを実行することです:

1)プリスケーラ(RTC_PREN = 0)をディスエーブルします。したがって、RTC は毎秒 32768 ティックを発生します。
2) Seconds イベントの代わりに Stopwatch イベントを使います。各ウェイクアップ・イベントでストップウォッチ・レジスタに 32768 (または、さらに一般的に所望の周波数に対応する値)を設定します。
3)この場合、ウェイクアップ信号は約 15 μ sec の長さになります。これは、ウェイクアップ後ハイバネート・モードを再開するまでにアプリケーションが待たなければならない(または何かを有用なことを行う)最小時間です。

カウンタは 1Hz でなく 32768Hz でインクリメントされるため、この対策は RTC が実際の時刻を表するために使われないことに注意してください。

適用レビジョン:

0.3、0.4

42. 05000257 - 短いハードウェア・ループ時の割り込み/例外により不正な命令がフェッチされることがある:

説明:

4 命令長より短いハードウェア・ループが、割り込みまたは例外に起因してループの終わりに中断されると、予期しない動作が発生することがあります。この状況では、命令フェッチのレイテンシを小さくするために使用されているプロセッサのループ・バッファが正しく動作しなくなるため、ループの終わりで正しくない命令がフェッチされてしまうことがあります。

対策:

このアノーマリには幾つかの対策があります。1 つ目は、すべての割り込み/例外ハンドラ内でループ・カウンタ・レジスタ(LC0 と LC1)に書き込みを行うことにより、ループ・バッファをクリアすることです:

```
R0=LC0;
LC0=R0;
R0=LC1;
LC1=R0;
```

2 つ目は、コンテキスト・スイッチ・コード内にループ・カウンタを使用することです:

```
[--SP] = LC0;
[--SP] = LC1;

<interrupt code>

LC1 = [SP++];
LC0 = [SP++];
```

最後の対策は、ループ長を 4 命令以上にするため、ループに NOP を追加することです。

また、イベント・ハンドラでハードウェア・ループを使用している場合、LCx レジスタに書き込みが行われるごとに、対応するループ・バッファがクリアされるため、上記ステップは不要です。

VisualDSP++Blackfin コンパイラには、このアノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ '-workaround short-loop-exceptions-257'を指定することにより、対策を手動でイネーブルすることができます。対策をイネーブルすると、コンパイラは LC0 レジスタと LC1 レジスタの退避と回復を割り込みハンドラと例外ハンドラに追加します(既に存在しない場合)。これらのレジスタが処理ルーチン内で使用されている場合には、通常コンパイラはこれらを退避させるだけです。

対策をイネーブルすると、__WORKAROUND_SHORT_LOOP_EXCEPTIONS が、コンパイル、アセンブル、リンク・ビルド・フェーズで定義されます。

適用レビジョン:

0.3、0.4

43. 05000258 - ICPLB データ・レジスタのビット 9 とビット 12 が異なるとき、命令キャッシュが壊れる:

説明:

ICPLB データ MMR のビット 9 とビット 12 が異なるとき、キャッシュが正しく更新されません。たとえば、特定のキャッシュ・ラインについて、そのキャッシュ・ラインの値がキャッシュ内に存在しない間、キャッシュ・タグが有効になることがあります。

対策:

各 ICPLB エントリ内で、ビット 12 の状態をビット 9 に設定してください。

VisualDSP++Blackfin ランタイム・ライブラリには、このアノーマリに対する対策が含まれています。

_cplb_mgr and _cplb_init ルーチン(これはランタイム・ライブラリのデフォルト・キャッシュ・サポートに含まれています)がビット 12 (CPLB_L1_CHBL)と同じ状態をビット 9 (予約済み)に設定します。

適用レビジョン:

0.3、0.4

44. 05000260 - ICPLB_ステータス MMR レジスタが壊れることがある:

説明:

例外を発生させた CPLB を調べるとき、ICPLB ステータス・レジスタに頼ることができません。このレジスタは次の場合に壊れます:

1)ページの最後の 64 ビット内へジャンプする命令があり(ICPLB により指定)、さらに、

2)これらの最後の 64 ビット内に存在する命令が命令例外を発生し、さらに、

3)投機的命令フェッチが、次のページまでインクリメントして、別の命令例外の原因に遭遇する

これらすべての条件が満たされると、ICPLB_STATUS は最初の例外原因ではなく投機的命令フェッチの方を反映するようになります。

対策:

命令保護違反と ICPLB の複数ヒットをこのレジスタを使わずに処理してください。

ICPLB_FAULT_ADDR レジスタを使って、例外を発生したアドレスを調べてください:

1)CPLB ミスの場合、問題のアドレスをカバーする CPLB が例外により単純にスワップ・インされます。

2)CPLB の複数ヒットの場合、ICPLB_FAULT_ADDR レジスタを使って、例外を発生したアドレスを探し、次にすべての CPLB エントリについて問題のアドレスをカバーする CPLB を探してください。

3)保護違反例外の場合、処理はユーザ固有になります。

適用レビジョン:

0.3、0.4

45. 05000261 - DCPLB FAULT ADDR MMR レジスタが壊れる:

説明:

DCPLB_FAULT_ADDR MMR レジスタが壊れることがあります。これが発生した場合、アボートされたデータ・メモリ・アクセスは、保護例外とストール (2 個の DAG の衝突、キャッシャブル・メモリへのアドレシングやミス、または L2 からの単なるフェッチなどにより発生) を発生します。

対策:

1)データ例外ハンドラへの最初のエントリ時には、そのハンドラから(サービスの実行なしに)直ちにリターンし、同じデータ CPLB 例外ハンドラ内の 2 つ目のパスで DCPLB_FAULT_ADDR を信頼します。例外の原因がサービスされていなければ、この例外が再発生して 2 つ目のパスを発生させます。例外ハンドラは、戻った直後に誤って生成されることがないので、2 つ目のパスでは、DCPLB_FAULT_ADDR レジスタが正しくなります。(高い優先順位の例外ではなく) 同じ例外の 2 つ目のパスで正しく応答できるように、最初のパスで RETX レジスタのコピーを取得して、2 つ目のパスと比較する必要があります。

または

2)例外の誤処理により発生する偽例外に耐性を持たせます。保護違反、CPLB ミス、CPLB 複数ヒットの 3 種類のデータ・メモリ例外の場合、推奨ソフトウェア対策は次の通りです:

a)データ保護例外の場合、ハンドラ内で DCPLB_FAULT_ADDR レジスタの代わりに DCPLB_STATUS レジスタを使用します。これにより、フル・アドレスの例外の代わりに保護違反のページが提供されます。理想的ではないですが、これが十分な情報を提供するものと思われます。

b)データ CPLB ミス例外の場合、DCPLB_FAULT_ADDR レジスタを使いますが、このレジスタに報告されるアドレスが現在の真の例外ではなく前にキャンセルされた投機的例外であることの警告を受けるようにします。したがって、このアドレスは:

i)ロード済みディスクリプタを持っているページを指す

または

ii)実際にフェッチされることのないアドレスを指す

i)の場合、CPLB ミス・ハンドラが冗長な CPLB エントリを発生することがありますが(さらにページ・チェックが実行されない限り)、この希に発生させる冗長なページ・ディスクリプタを削除する複数 CPLB ヒット・ハンドラが存在する場合、これにより耐性を持つようになります。

ii)の場合、DCPLB_FAULT_ADDR レジスタは例外ハンドラから戻った直後に不正になることがないため、誤った DCPLB_FAULT_ADDR レジスタ値の繰り返しに起因する例外ハンドラの無限ループを発生させることなく、CPLB ミス・ハンドラから無意味なアドレスを無視することができます。

c)データ CPLB 複数ヒット例外の場合、上記問題に対応するようなハンドラを設けます。複数 CPLB 例外が発生しないことに頼らないでください。

VisualDSP++Blackfin ランタイム・ライブラリには、このアノーマリに対する対策が含まれています。この対策では、DCPLB ミス例外が特定の PC から発生した最初のとき、これを無視します。障害アドレスの修正は 2 回目に保証されます。

適用レビジョン:

0.3、0.4

46. 05000262 - データ・キャッシュへのストアが失われることがある:

説明:

連続する 2 つのデュアル DAG 動作の最初にターゲットされたサブバンクに対して確定した待機中の書き込みが次の場合に失われます:

1)データ・キャッシュがイネーブルされ、さらに、

2)最初のデュアル DAG アクセスで、DAG0 がキャッシュ・ミスになり、DAG1 が読み出しで、かつ両アクセスが同じ非 L1 サブバンクに対するエイリアスで、さらに、

3)2 つ目のデュアル DAG が次の命令であり、かつ DAG1 が L1 SRAM のアクセス(読み出しまたは書き込み)であり、さらに、

4)最初のデュアル DAG アクセス後 3 クロック・サイクル以内にフローの予期しない変更があり、ユーザがフロー変化を制御していない場合

対策:

1)データ・キャッシュを使わないか、または、

2)最初のデュアル DAG アクセスが次の場合に、連続するデュアル DAG メモリ・アクセスを回避してください:

a)両 DAG が L2 をターゲットにし、さらに、

b)両 DAG が同じサブバンクをエイリアスし、さらに、

c) DAG1 による読み出しを含み、その直後に、DAG1 が L1 アクセスを行う 2 つ目のデュアル DAG アクセスが続く場合

VisualDSP++Blackfin コンパイラには、このアノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ '-workaround stores-to-data-cache-262'を指定することにより、対策を手動でイネーブルすることができます。

対策がイネーブルされると、コンパイラが、アノーマリを発生するデュアル DAG 命令を発行しないようにします。コンパイラは、対策が不要なケースを調べるために使用可能なすべてのバンク情報を使用しようとします。

対策をイネーブルすると、__WORKAROUND_LOST_STORES_TO_DATA_CACHE_262 が、コンパイル、アセンブル、リンク・ビルド・フェーズで定義されます。

VisualDSP++ランタイム・ライブラリは、該当する場合、このアノーマリを回避するようにビルドおよび修正されています。

適用レビジョン:

0.3、0.4

47. 05000263 - ICPLB 例外を受理するとハードウェア・ループが壊れる:

説明:

ハードウェア・ループ・ロジックにエラーがあり、このためプロセッサのループ実行中に命令 ICPLB 例外が発生すると、不正な命令が実行されることがあります。

対策:

1)ハードウェア・ループの使用を回避するか、または
2)ハードウェア・ループが L1 メモリ内にのみあることを確認するか、または
3) L1 メモリの外部にあるハードウェア・ループの実行中に ICPLB 例外が発生しないことを確認してください。

ハードウェア・ループが L1 メモリ内にある場合、このループは、たとえば、CPLB ページ境界を超えて有効な CPLB 定義を持たないページへ行くなどにより、ICPLB 例外を発生しません。また、ターゲット・アドレスで ICPLB 例外が発生したとき、ループ内から非 L1 メモリへ分岐させないでください。ループ中に割り込みが発生して、割り込みサービス・ルーチン(ISR)が非 L1 メモリ内にある場合は、ISR から ICPLB 例外を発生させないでください。

適用レビジョン:

0.3、0.4

48. 05000264 - ハードウェア・ループ内の最後から 2 番目の命令で CSYNC/SSYNC/IDLE により無限ストールが発生する:

説明:

SSYNC、CSYNC、または IDLE がハードウェア・ループの最後から 2 番目の命令として配置されている場合、同期を実行しようとすると、プロセッサが無限ストールに入る可能性があります。

対策:

ハードウェア・ループの最後から 2 番目の命令として、SSYNC、CSYNC、または IDLE 命令を使わないでください。

割り込みまたは例外によりプロセッサがストールから抜け出すため、DMA または割り込みの実行中はこの問題は表面化しません。

VisualDSP++Blackfin コンパイラには、このアノーマリに対する対策が含まれています。コンパイラは、該当するシリコン・レビジョンとデバイス番号に対してこの対策を自動的にイネーブルします。あるいは、コンパイラ・フラグ'-workaround pre-loop-end-sync-stall-264'を指定することにより、対策を手動でイネーブルすることができます。

対策をイネーブルすると、コンパイラはハードウェア・ループの最後から 2 番目の命令が、このアノーマリを発生する可能性を持つ CSYNC、SSYNC または IDLE 命令にならないようにします。

対策をイネーブルすると、__WORKAROUND_PRE_LOOP_END_SYNC_STALL_264 が、コンパイル、アセンブル、リンク・ビルド・フェーズで定義されます。

適用レビジョン:

0.3、0.4

49. 05000265 - 外部 SPORT TX と RX クロックの低速な入力エッジ・レートでノイズに対して弱くなる:

説明:

ノイズの多いボード環境と外部 SPORT の受信クロック(RSCLK)と送信クロック(TSCLK)の低速入力エッジ・レートとの組み合わせにより、多様な問題が観測されます。RSCLK/TSCLK で予期しない高周波変化が発生すると、SPORT がノイズから生ずる余分なグリッチ・クロック・パルスを認識することがあります。

RSCLK/TSCLK での高周波変化は、多くの場合外部シリアル・クロックの立ち上がりまたは立ち下がりエッジのノイズから発生します。このノイズと低速で変化するシリアル・クロック信号とが組み合わさると、クロック入力の高い感度のため、幅の狭い余分なビット・クロックが発生することがあります。低速なスルーレート入力により、スイッチング・ポイント付近のクロック入力のノイズで、クロック入力がスイッチング・ポイントを横切り、さらに再度横切るようになります。この発振により、シリアル・ポートの内部ロジックでグリッチ・クロック・パルスが発生します。

このグリッチ・クロック・パルスに起因して観測される問題は:

- ステレオ・シリアル・モードの場合、これは左/右データの入れ換えとなるフレーム同期外れとして現れます。
- マルチチャンネル・モードの場合、これは不正確なまたはスキップされたフレームを表す MFD カウントとして現れます。
- ノーマル(アーリー)フレーム同期モードの場合、受信データ・ワードが 1 ビット右にシフトされます。MSB が誤って符号拡張モードでキャプチャされます。
- すべてのモードで、フレーム同期の開始と、受信または送信最終ビットとの間のすべての入力クロックにノイズがある場合、受信または送信データ・ワードが部分的に右シフトされて現れます。

ステレオ・シリアル・モードでは(SPORTx_RCR2 のビット 9 がセット)、RSCLK/TSCLK に予期しない高周波変化があると、SPORT がワード・クロックの立ち上がりまたは立ち下がりエッジを誤ることがあります。このために、ステレオ・シリアル・データの左または右ワードが失われます。これは、ステレオ・オーディオ信号を聞いているときに左/右チャンネルの入れ換えとして観測されます。SPORT の内部ロジックでノイズから余分なビット・クロック・パルスが発生すると、FS エッジ検出ロジックが狭い幅のパルスを発生し、同時に、SPORT が次の‘通常の’ビット・クロック周期内で外部 FS 信号の検出ができなくなります。エッジ検出ロジックから出力される幅の狭い FS パルスは、SPORT のシーケンシャルなロジックから無視されます。FS ロジックのエッジ検出部分は既に‘トリガー済み’であるため、次の‘通常の’ RSCLK は RFS の変化をこれ以上検出しません。I2S/EIAJ モードでは、このために、2 つの左/右チャンネルとして 1 ステレオ・サンプルが検出/転送され、すべての後続チャンネルはメモリ内でワードが入れ代わります。

マルチチャンネル・モードでは、マルチチャンネル・フレーム遅延(MFD)ロジックが余分な同期パルスを受信して、カウントを早めるかダブル・カウントしてしまいます(カウントが既に開始している場合)。MFD がゼロのときは、カウントが 1 サイクル早く開始されているため 15 へロールオーバーすることがあります。

アーリー・フレーム同期モードでは、FS がアクティブになった同じクロック・サイクルの駆動エッジでノイズが発生すると、FS ロジックは余分なパルスを受信して 1 サイクル早くワード長のカウントを開始します。最初のビットが 2 回サンプルされ、最後のビットがスキップされます。

すべてのモードでは、FS がアクティブになった後の任意のサイクルでノイズが発生すると、ビット・カウント・ロジックが余分なパルスを受信して、カウントが早く進み過ぎます。これが動作ユニットで発生すると、1 サイクル進んでワード長のカウントが終わります。ノイズが発生したビットは 2 回サンプルされて、最後のビットはスキップされます。

対策:

1)ビット・クロックのスルーレートを大きくするか、またはシリアル・ビット・クロックの立ち上がりと立ち下がり時間を短くして、ノイズに対する感度を小さくして、変化の周囲のノイズにより、狭い幅のノイズからビット・クロック・パルスが発生しないようにします。狭い高周波数パルスは、SPORT またはフレーム同期の検出に影響を与えません。EMI 条件を満たし適切な場合にはエッジをできるだけ急峻にします。
2)可能な場合、内部発生のビット・クロックとフレーム同期を使います。
3)正しい PCB デザイン方法に従います。TSCLK ラインから RSCLK をシールドして、シリアル・クロックの混入を小さくします。
4)ボード上の RSCLK、TSCLK、フレーム同期の各パターンを分離して、FS がスイッチするとき駆動エッジで発生する混入を小さくします。

ステレオ・シリアル・モードで観測される問題に対する特別な対策は、フレーム同期信号を遅延させて、ノイズから混入するビット・クロック・パルスにより、フレーム同期処理を開始させないようにすることです。これは、ビット・クロック・パターンの直列抵抗よりフレーム同期パターンの直列抵抗が大きい場合に実現できます。ビット・クロックが VDD の 10%閾値(立ち下がりエッジ・ビット・クロック)または 90%閾値(立ち上がりエッジ・ビット・クロック)を通過するまでフレーム同期変化が 50%ポイントを通過しないようにします。

ノイズ耐性を向上させるため、PLL_CTL レジスタのビット 15 をセットし、次に該当する PLL プログラム・シーケンスを実行して、入力ピンでオプションのヒステリシスをイネーブルすることができます。

適用レビジョン:

0.3、0.4、0.5、0.6

50. 05000269 - 活発な I/O 動作により、内部電圧レギュレータ出力電圧(Vddint)が上昇する:

説明:

内部電圧レギュレータは、特に VDDext が高いときに、活発な I/O 動作により混入する電源とグラウンドのノイズ過渡電圧に敏感です。このために、VDDint が設定された値より高くなることがあります。場合によっては、値がデータシートの上限を超えることがあります。VDDint は、I/O 動作が減少または停止すると設定された値に戻ります。

BGA パッケージを採用するデバイスは、LQFP パッケージを採用するデバイスより敏感です。

現在まで、規定値の上限を超える電圧は、疑似的に I/O を活発に動作させたテストを実行中にのみ観測されています(たとえば、アドレス・ラインとデータ・ラインのすべてのビットが各クロックでトグル)。仕様外の動作は、アプリケーション・コードを実行するユーザのアプリケーションでは観測されていません。

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、グラウンドと信号のループを最小にして VDDint に接続します。
- 通常の過渡電圧を回避するため、設定する値より 10%大きい値と絶対最大電圧値(最大 Vddint 仕様についてはデータシートを参照)との間の 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。

すべてのデバイスが、この問題に同じように敏感ではないことに注意してください。最小 10 個のデバイスで上記監視を繰り返してください。

問題が発生する場合には、I/O が活発に動作している間 VDDint 値が大きくなります。VDDint の最大値が最大 VDDint 以下に維持される場合には、信頼性の問題はありませんが、消費電力は大きくなります。

問題は、VDDext、I/O の動作度、VDDint の設定値の関数であるため、次の技術を使ってこの問題を軽減/改善することができます:

- 問題は、VDDext 値が 3.3V より高いときに発生しやすいため、偏差+/- 2%以下の 3.3V (または以下)のレギュレータを使うことが最適です。
- VDDext に十分なバイパスを設けます。
- 可能な場合 I/O 動作を減らします(たとえば、動作 SCLK 周波数レートを下げます)。
- アプリケーション・ノート EE-228 の条件に従います。さらに、ユーザ・アプリケーションの定格条件を満たす最小ゲート電荷定格を持つ PMOS FET を使用します。

適用レビジョン:

0.3、0.4

51. 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、グラウンドと信号のループを最小にして 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、0.4

52. 05000271 - 内部電圧レギュレータの自発的リセット:

説明:

内部電圧レギュレータがデフォルトと異なる電圧レベルに設定されると、電圧レギュレータ回路のノイズにより、1.2V のデフォルト値へ自発的にリセットされることがあります。

VR_CTL レジスタの VLEV ビット・フィールドは影響を受けないため、レジスタ値の読み出しでこの状態を検出できないことに注意してください。

対策:

なし

適用レビジョン:

0.3

53. 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、0.6

54. 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、0.4、0.5

55. 05000276 - 非ゼロ PPI_DELAY を持つ外部フレーム同期 PPI モードに対するタイミング条件の変更:

説明:

プロセッサのデータシートの PPI タイミング図は、PPI_DELAY レジスタがゼロに設定された PPI モードにのみ適用されます。

対策:

PPI_DELAY レジスタの非ゼロ値に対して、次の情報が適用されます:

データシートでは、POLC = 0 のとき、フレーム同期は PPI クロックの立ち下がりエッジでサンプルされ、対応するセットアップ時間は、このエッジに対して表示されています。PPI_DELAY レジスタが非ゼロ値のとき、フレーム同期セットアップ時間が PPI クロック周期の 1/2 だけ大きくなります。データのサンプルを示す既存の図のポイントで、遅延のカウントが開始されます。

データシートでは、POLC = 1 のとき、フレーム同期は PPI クロックの立ち上がりエッジでサンプルされ、対応するセットアップ時間は、このエッジに対して表示されています。PPI_DELAY レジスタが非ゼロ値のとき、フレーム同期セットアップ時間が PPI クロック周期の 1/2 だけ大きくなります。データのサンプルを示す既存の図のポイントで、遅延のカウントが開始されます。

適用レビジョン:

0.3、0.4、0.5、0.6

56. 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・サイクル後に発生した場合、このビットがセットされてから 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、0.4、0.5

57. 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、0.4、0.5

58. 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:

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 instruction C のアドレスが含まれます)。プロセッサが上記 instruction B に到達したとき ISR が Py を変更すると、instruction 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。instruction A のため、instruction 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、0.4、0.5

59. 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、0.4、0.5

60. 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。たとえば、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、0.4、0.5

61. 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) 1 ワードがさらに 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、0.4、0.5

62. 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(たとえば、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、0.4、0.5

63. 05000302 - DMA MMR レジスタに対する書き込みの後の SSYNC が正しく処理されない:

説明:

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、SSYNC 命令の有無に関係なく、同じ DMA コントローラに対応するシステム MMR に対する書き込みがディスクリプタのフェッチ完了まで遅延させられます。

この動作による 1 つの悪影響は、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 割り込みの場合です:

```
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 に対する書き込みに対しても当てはまります。

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。たとえば、上の例を使用して、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、0.4

64. 05000305 - PLL_CTL レジスタの SPORT_HYS ビットが機能しない:

説明:

PLL_CTL レジスタの SPORT ヒステリシス・ビット(SPORT_HYS、ビット 15)が機能しません。このビットを読み出すと常に 0 が返され、1 を書き込むと無視されます。

対策:

なし。

適用レビジョン:

0.3、0.4

65. 05000306 - PPI_コントロール・レジスタの ALT_TIMING ビットが機能しない:

説明:

PPI_CONTROL レジスタの ALT_TIMING ビット(ビット 8)が機能しません。このビットは、PPI データがサンプルされるエッジ(PPI フレーム同期が基準)をソフトウェアから設定できるようにします。このビットを読み出すと常に 0 が返され、1 を書き込むと無視されます。

対策:

なし。

適用レビジョン:

0.3、0.4

66. 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)ページ境界に分岐命令またはデータを配置しないでください。境界の前では、予約済みメモリ・スペースを使い少なくとも 76 バイトを空けてください。これにより偽例外の発生を防止することができます。

2)例外が有効か否かをアクションの前に判断する例外処理を設けてください。これは、CODE_FAULT_ADDR (または DATA_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、0.6

67. 05000311 - 特定のシーケンスでフラグ(GPIO)ピンが誤動作する:

説明:

GPIO /フラグ IO システム MMR (PORTFIO_プレフィックスまたは FIO_プレフィックスを持つ任意のレジスタ)に対するアクセスの後ろに、別のシステム MMR アクセス(GPIO /フラグ IO ブロック外)が続くと、GPIO の入力ドライバが短時間アクティブになります。このため、ポート F ラッチに保持されている出力値が誤ってクリアされて、出力ピンの状態に不要な変化が発生します。

MMR アクセスのある組み合わせでのみこの不具合が発生し、GPIO レジスタ・アクセスのタイプ(書き込み、読み出し、読み出しのアボート)により不具合が変わります。幾つかの不具合は非常に希に発生するため、システム評価時に検出されない可能性があります。このため、アドレス・ビット 4、5、または 6 が GPIO レジスタと続いてアクセスされる MMR との間で異なっている任意の MMR の組み合わせでこの不具合が発生するものと見なされています。さらに、この問題が発生するためには、連続する命令内で 2 回の MMR アクセスが起こる必要はありません。アクセスは、無制限のサイクル数/命令数で分離することができます。

複数の GPIO/FIO レジスタに対するアクセスは互いに妨げにならないため、入力として設定された GPIO フラグ・ピンは影響を受けません。

対策:

GPIO レジスタに対するアクセスの各シーケンスは、安全なレジスタの読み出しで終了する必要があります。"安全なレジスタ"とは、直前にアクセスされた GPIO レジスタと同じアドレス・ビット 4、5、6 を持つ任意の非 GPIO システム MMR と定義されます。対策では、この規則が条件付きジヤンプや割り込みのような不連続なプログラム・フローにより乱されないようにする必要があります。歓迎される副作用として、GPIO MMR 読み出しのアボートが完全に回避されます。たとえば、次のシーケンスは安全です:

```
P5.H = hi(PORTFIO); P5.L = lo(PORTFIO);     /* PORTFIO is the same as FIO_FLAG_D */
P4.H = hi(SYSCR); P4.L = lo(SYSCR);

cli R7;                     /* avoid interrupts */
nop; nop; nop;              /* three cycles after CLI before 1st FIO read access */

/* any GPIO sequence */
R6 = w[P5](z);
R5 = w[P5+PORTFIO_MASKA-PORTFIO](z);        /* PORTFIO_MASKA_D read */
R6 = R5 & R6;

w[P5+PORTFIO_CLEAR-PORTFIO] = R6;           /* last GPIO access to PORTFIO_CLEAR (0xFFC00704) */
R5 = w[P4](z);                              /* dummy read from SYSCR (0xFFC00104) */
sti R7;                     /* restore interrupts */
```

SYSCR と PORTFIO_CLEAR のアドレス・ビット 4、5、6 が b#000 であることに注意してください。このため、SYSCR 読み出しが、PORTFIO_CLEAR のアクセスにより発生するこのクリティカルな状況を安全に解決します。

各 GPIO/FIO レジスタの安全なレジスタの包括的なリストを次に示します。

If the last GPIO access was to...	Then the "safe registers" are...
PORTFIO/_CLEAR/_SET/_TOGGLE (FIO_FLAG_D/C/S/T)	SYSCR, PPI_STATUS, or SPI_STAT
PORTFIO_MASKA/_CLEAR/_SET/_TOGGLE (FIO_MASKA_D/C/S/T)	UART_SCR, TIMER1_CONFIG, or EBIU_SDSTAT
PORTFIO_MASKB/_CLEAR/_SET/_TOGGLE (FIO_MASKB_D/C/S/T)	UART_GCTL, TIMER2_CONFIG, or DMA0_IRQ_STATUS
PORTFIO_DIR/POLAR/EDGE/BOTH (FIO_DIR/POLAR/EDGE/BOTH)	SPORT0_STAT, SPORT1_STAT, or DMA0_CURR_X_COUNT
PORTFIO_INEN (FIO_INEN)	TIMER_ENABLE, TIMER_STATUS, or DMA1_CONFIG

特別なツールの影響:

VisualDSP++ 4.5 (2007 年 2 月アップデート)では、コンパイラが CHIPID のダミー読み出しを挿入しましたが、ダミー読み出しアドレスが変わるため(さらに割り込みをディスエーブルする必要があるため)、この対策は VisualDSP++ 4.5 (2007 年 6 月アップデート)でヘッダー05000311.h により置き換えられました。このヘッダーでは、コンパイラ制御ではなくユーザ制御のもとでこれを実行するマクロが定義されています。詳細については、ヘッダー・ファイルを参照してください。

適用レビジョン:

0.3、0.4、0.5

68. 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 = (only when LCx is non-zero)
LBx = (only when LCx is non-zero)
```

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。その他にもエラーが、例外、ハードウェア・エラー、または有効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。たとえば、

```
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、0.5

69. 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、0.4、0.5

70. 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。たとえば、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、0.4、0.5

71. 05000319 – LQFP パッケージで内部電圧レギュレータ値 1.05V、1.10V、1.15V が使用できない:

説明:

VR_CTL レジスタに VLEV 値 0xA、0xB、0xC (それぞれ 1.05V、1.10V、1.15V)を設定すると、レギュレータを介してコアに加える実際の Vddint が規定偏差-5%を下回ります。

この問題は、LQFP パッケージを採用したデバイスにのみ発生します。

対策:

これらの値の設定を回避するか、またはレギュレータに次に高い設定値を設定して、アプリケーションが動作しているコア・クロックに対する最小閾値を Vddint が上回るようにしてください。

適用レビジョン:

0.3、0.4、0.5

72. 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値によらず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。たとえば、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、0.5

73. 05000363 - UART ブレーク信号の問題:

説明:

ブレーク信号が受信されると、UART コントローラはエラー割り込みを 1 回発生する必要がありますが、ブレーク信号がアクティブとなる各ビット時間に対してエラー割り込みを発生して、コントローラが多くのエラー割り込みを発生してしまいます。たとえば、57600 のボー・レートで約 250ms 間ブレーク信号がラインをロー・レベルに維持すると、ストリーム内にスタート・ビットまたはストップ・ビットがないことと無関係に、約 1400 回のエラー割り込みが発生されます。内部的には、ブレーク信号の間にサンプルされるデータはすべて 0 であるため、UART_RBR レジスタ内でデータは 0 として受信されます。

別の問題は、ブレーク信号自体のタイミングです。次の有効なキャラクタが送信されるタイミングに応じて、ブレーク信号の結果により次の有効なキャラクタが 2 つの無効なキャラクタに分割されます。これは受信される先頭ビットが前の不正データの後ろに追加され、有効なキャラクタの残りの部分が次のキャラクタとしてサンプルされるためです。ブレークの後にデータが連続ストリームとして続く場合、この動作は UART を介して受信される後続キャラクタのストリームで伝搬することがあります。

対策:

1 つのブレーク信号により発生する多くの割り込みに対して、ソフトウェア内で多くの割り込みを 1 つにまとめてそれをサービスする以外に対策はありません。発生される各エラー割り込みは個別にサービスする必要があります。たとえば、ソフトウェアではフラグを使ってこれを制御することができます。UART エラー割り込みハンドラがフラグをセットし、後続の割り込み要求をこのフラグがクリアされるまでスキップすると、複数のブレークを 1 つとして処理することができます。UART RX 割り込み内で同じフラグをクリアすることにより、ブレークが完了して、UART から新しい有効なデータが受信されたことを表示することができます。

ブレークの後に分割されるデータは、回復できません。ホストがブレークの後に次のデータの送信を 1 キャラクタ分のビット時間だけ遅延させると、Blackfin UART はブレーク信号から回復して、有効なデータの受信を再開できるようになります。たとえば、1 スタート・ビット、1 パリティ・ビット、2 ストップ・ビットを持つ 8 ビット・データの場合、ホストはブレーク信号の後、少なくとも 12 UART ビット時間待って次の有効データを発行する必要があります。

適用レビジョン:

0.3、0.4

74. 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(PPI_STATUS の UNDR)が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、0.6

75. 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。たとえば、

```
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。

VisualDSP++ 4.5 アップデート 6 と VisualDSP++ 5.0 アップデート 2 から、ツールにはこのアノーマリの対策が含まれています。C/C++コンパイラの対策では、RTS の前に NOP 命令または無条件 JUMP 命令を挿入して stub 関数コードの生成を回避しています。2 個を超える NOP が必要とされる場合には、コード・サイズを最適化(-Os)するときに、JUMP 対策の派生が使用されます。このアノーマリを発生させるコードに対しては、検出して警告(ea5516)を発するようにアセンブラが変更されています。ランタイム・ライブラリと VDK サポート・ライブラリも、このアノーマリを変更するように修正されています。

VisualDSP++では、影響を受けるプロセッサに対してビルドするとき、これらの対策が自動的にイネーブルされます。コンパイラの対策は、-workaround avoid-quick-rts-371 スイッチを使って手動でイネーブルすることができます。アセンブラの警告は、-anomaly-detect 05000371 スイッチを使って制御されます。この対策をイネーブルすると、__WORKAROUND_AVOID_QUICK_RTS_371 マクロが、コンパイル、アセンブル、リンクのステージで定義されます。

適用レビジョン:

0.3、0.4、0.5

76. 05000400 - 特別なモードで PPI が正常に起動されない:

説明:

PPI ポートが 2 内部フレーム同期を使う送信モードに設定された場合、PPI フレーム同期 3 ピン(PPI_FS3)がフローティングのままのとき、PPI が正常に起動されません。

対策:

PPI を 2 内部フレーム同期を使う送信モードに設定するときは、PPI_FS3 ピンをプルダウンする必要があります。

適用レビジョン:

0.5

77. 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 によりストールされる:

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が破られます。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)の終わりから少なくとも 8 ワード(1 ワード = 64 ビット)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:

0.5

78. 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、0.6

79. 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に、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、0.6

80. 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)フレーム遅延(MFD) > 0。

3)ウインドウ・サイズが 8 の奇数倍(すなわち WSIZE が偶数)。

4) RFS パルス間の間隔がウインドウ継続時間に一致する。

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、このために、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) 0 のフレーム遅延を使います。

3) 8 の偶数倍のウインドウ・サイズを使います。

4)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、0.6

81. 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ヤ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(たとえば-1 を設定)を使う場合に発生します。たとえば、

```
      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は実際に実行されないため偽ハードウェア・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:

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、0.6