ANALOG DEVICES

Blackfin® 組込みプロセッサ

シリコン・アノーマリ・リスト　ADSP-BF534/BF536/BF537

ADSP-BF534/BF536/BF537 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-BF534/BF536/BF537 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-BF534/BF536/BF537 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>
0.3	0x0003
0.2	0x0002

対象製品

各アノーマリは、特定のシリコン・レビジョンに適用されます。影響を受けるレビジョンについては、一覧または詳細リストを参照してください。さらに、このアノーマリ・リストに記載するすべてのプロセッサが、同じ機能セットを持っているわけではありません。したがって、ペリフェラル固有のアノーマリはすべてのプロセッサに適用できません。詳細については、下の表を参照してください。"x"マークは、このペリフェラルに関係するアノーマリは、表示されたモデルのみに適用されることを示し、そのペリフェラルに対する固有なアノーマリのリストを最も右の列に示します。

Peripheral	ADSP-BF534	ADSP-BF536	ADSP-BF537	Anomalies
Ethernet MAC	-	x	x	05000252, 05000256, 05000285, 05000316, 05000321, 05000322, 05000341

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	D	E	Added Anomalies—05000416, 05000425, 05000426 Revised Anomaly—05000283, 05000315
02/08/2008	C	D	Added Anomalies—05000402, 05000403 Removed Anomaly—05000359
12/10/2007	B	D	Removed Silicon Revision 0.1 Added Anomalies—05000350, 05000355, 05000366, 05000371 Removed Anomaly—05000167
09/04/2007	A	D	Initial Consolidated Revision—Replaces anomaly lists for (Rev M), ADSP-BF536 (Rev L) and ADSP-BF537 (Rev M)ADSP-BF534 Added Anomalies—05000167, 05000341, 05000357, 05000359

NR003531D

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-BF534/BF536/BF537 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	0.2	0.3
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
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
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
4	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
5	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	.
6	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
7	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	.
8	05000252	アーリー・フレームをアボートした後に EMAC TX DMA がエラーを発生する	x	.
9	05000253	タイマの最大外部クロック速度	x	.
10	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
11	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	.
12	05000256	EMAC MDIO 入力が不正な MDC エッジでラッチされる	x	.
13	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	.
14	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	.
15	05000260	ICPLB_ステータス MMR レジスタが壊れることがある	x	.
16	05000261	DCPLB FAULT ADDR MMR レジスタが壊れる	x	.
17	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	.
18	05000263	ICPLB 例外を受理するとハードウェア・ループが壊れる	x	.
19	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	.
20	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
21	05000268	DEB_TRAFFIC_PERIOD がゼロでないときペリフェラル DMA が動作中にメモリ DMA がエラーを発生する	x	.
22	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	.
23	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
24	05000273	同期 SDRAM メモリへの書き込みが失われる	x	.
25	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	.
26	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	.
27	05000280	Atmel 社のデータ・フラッシュ・デバイスで SPI マスタ・ブート・モードが正常に動作しない	x	x
28	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	.
29	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	.
30	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	.
31	05000285	EMAC_SYSCTL レジスタの TXDWA ビットが機能しない	x	.
32	05000288	FIFO が満杯になると SPORT が不正なデータを受信する	x	.
33	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
34	05000304	CAN/DMA MMR レジスタへの書き込み後、SSYNC は正常に処理されないことがある	x	.
35	05000305	PLL_CTL レジスタの SPORT_HYS ビットが機能しない	x	.
36	05000307	ハイバネート中に SCKELOW ビットが状態を維持しない	x	.
37	05000309	UART クロックのディスエーブル中に UART THR に書き込みを行うと不正なスタート・ビットが送信される	x	.
38	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
39	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
40	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
41	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	.
42	05000316	EMAC RMII モード:全二重モードで衝突が発生する	x	.

No.	ID	説明	0.2	0.3
43	05000321	EMAC RMII モード: TX フレームが半二重でステータス"No Carrier"のエラーを発生する	x	.
44	05000322	10-Base-T 速度の EMAC RMII モード: RX フレームが正常に受信されない	x	x
45	05000341	イーサネット MAC MDIO 読み出しで IEEE 仕様を満たさない	.	x
46	05000350	UART ブート後に UART がディスエーブルされる	x	.
47	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
48	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
49	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
50	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
51	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	.
52	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
53	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
54	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
55	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

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-BF534/BF536/BF537 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:

0.2, 0.3

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:

0.2、0.3

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。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:

0.2、0.3

4. 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:

0.2、0.3

5. 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"追い付く"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 ...;          IF CC JUMP X (BP);          LSETUP(LT, LB)
CSYNC/SSYNC/IDLE         ...                             LT: CSYNC/SSYNC/IDLE
CSYNC/SSYNC/IDLE         ...                                 CSYNC/SSYNC/IDLE
CSYNC/SSYNC/IDLE         X: 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:

0.2

6. 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)
· 揮発性評価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:

0.2、0.3

7. 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。
RFSDIV = 127→すべてのデータ・ワードが正しい
RFSDIV = 128→先頭ワードが誤り
RFSDIV = 129→すべてのデータ・ワードが正しい
RFSDIV = 130→すべてのデータ・ワードが正しい

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:

0.2

8. 05000252—アーリー・フレームをアボートした後に EMAC TX DMA がエラーを発生する:

説明:

データとディスクリプタが異なるメモリ空間にあるとき(すなわちデータが外部メモリで、ディスクリプタが L1)、EMAC が TX アーリー・アボート動作を実行すると、DMA エラーが発生することがあります。

TX アーリー・アボートは希なケースです。次の 1)と 2)が同時に生じるときに発生します:
1)フレームの初期に、EMAC TX FIFO が初期の FIFO フィルを実行中である(すなわち、まだ 90 番目バイトまたはフレームの終わりの内の小さい方に到達していない)
2)次の 4 ケースのフレーム・アボートのいずれかが発生する:
a) DMA アンダーラン
b)コリジョンが多い
c)ディフェラルが多い、ディフェラル・チェックのイネーブル時(OPMODE:DC = 1)
d)レイト・コリジョン、レイト・コリジョン・リトライのディスエーブル時(OPMODE:LCTRE = 0)

アーリー・アボートの 4 つの原因の他に、このエラーは、次のいずれかの場合に、フレーム後期(DMA が 90 バイトを送信した後)でも発生することがあります:
a) DMA アンダーラン
b)レイト・コリジョン、レイト・コリジョン・リトライのディスエーブル時(OPMODE:LCTRE = 0)

対策:

対策は、メモリ空間を次のように構成することです:

	Descriptor	Data Buffer
Data 1	in A	in B
Data 2	in C	in D
Status	in D	in E

各文字(A～E)はメモリ空間を表します。異なる文字が同じメモリ空間または異なるメモリ空間を表すことがありますが、同じ文字が使われる項目の各グループは同じメモリ空間内にある必要があります。

適用レビジョン:

0.2

9. 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:

0.2

10. 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:

0.3

11. 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 µsec 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:

0.2

12. 05000256—EMAC MDIO 入力が不正な MDC エッジでラッチされる:

説明:

MDIO 入力が MDC の立ち下がりエッジより 1SCLK サイクル後にラッチされます。このために、MDC クロックが非常に短いサイクル時間に設定された場合、PHY デバイスによっては通信エラーが発生することがあります。MDC/MDIO は、PHY デバイスの内部コントロール・レジスタとの通信に使用されます。

注: MDC クロックはプロセッサから駆動され、EMAC ペリフェラル内のプログラマブルなクロック分周器により SCLK から発生されます。分周比レジスタ MDCDIV に N が設定されると、MDC の周期は 2(N+1)×SCLK になります。

PHY は通常、MDC の立ち上がりエッジで MDIO を駆動し、プロセッサは上述のように MDC の立ち下がりの 1SCLK 後に MDIO をサンプルします。これは、PHY のクロック—出力間遅延と伝搬遅延の和(tdMDIO)が MDC 周期と 1SCLK サイクルの和の 1/2 を超えることができないことを意味しています。プロセッサの MDC 分周比レジスタに N を設定する場合、伝搬遅延は[2(N+1)]/2 + 1 = N+2 SCLK サイクルを超えることはできません。

対策:

PHY がこの問題に遭遇する場合には、PHY の tdMDIO を調べて、MDCDIV > (tdMDIO/tSCLK) + 2 に設定してください。

適用レビジョン:

0.2

13. 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る。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:

0.2

14. 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:

0.2

15. 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、例外により問題のアドレスをカバーする CPLB が単純にスワップ・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:

0.2

16. 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、次のようになります:

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:

0.2

17. 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:

0.2

18. 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:

0.2

19. 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:

0.2

20. 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・ポイントを通過し、さらに再通過するようになります。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 へロールオーバーできます。

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:

0.2、0.3

21. 05000268—DEB_TRAFFIC_PERIOD がゼロでないときペリフェラル DMA が動作中にメモリ DMA がエラーを発生する:

説明:

メモリ DMA チャンネルが任意のペリフェラル DMA チャンネルと同時にアクティブで、かつ DMA_TC_PER レジスタの DEB_TRAFFIC_PERIOD ビットが非ゼロの場合、メモリ DMA 転送内でデータが壊れることがあります。

対策:

アプリケーション内でメモリ DMA チャンネルをペリフェラル DMA チャンネルと同時に使用しないか、または、メモリ DMA をペリフェラル DMA と同時に使用する場合には、DMA_TC_PER レジスタの DEB_TRAFFIC_PERIOD ビットに b#0000 を設定してください。

適用レビジョン:

0.2

22. 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:

0.2

23. 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:

0.2、0.3

24. 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:

0.2

25. 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:

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:

0.2

26. 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:

0.2

27. 05000280—Atmel 社のデータ・フラッシュ・デバイスで SPI マスタ・ブート・モードが正常に動作しない:

説明:

ブート・ストリームが SPI デバイスのページ・サイズより大きい場合、ブート ROM コードにより、Atmel 社のシリアル・データフラッシュ・デバイスからのブートが失敗します。これは、データフラッシュが標準アドレッシング・モードで動作する場合にのみ発生します。

対策:

シリーズ D デバイス(AT45DB642D)を 2 の累乗アドレッシング・モードで使用してください。

適用レビジョン:

0.2、0.3

28. 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:

0.2

29. 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:

0.2

30. 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:

0.2

31. 05000285—EMAC_SYSCTL レジスタの TXDWA ビットが機能しない:

説明:

EMAC_SYSCTL レジスタの EMAC TX DMA Word Alignment ビット(TXDWA、ビット 4)を使うと、TX フレーム・データを奇数または偶数整列のワード・アドレスから開始するようにソフトウェアから指定できますが、このビットが機能しません。このビットを読み出すと常に 0 が返され、このビットへの書き込みは無視されます。

対策:

なし。

適用レビジョン:

0.2

32. 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:

0.2

33. 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:

0.2、0.3

34. 05000304—CAN/DMA MMR レジスタへの書き込み後、SSYNC は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、同じスペースに対するアクセス中、それぞれ MMR スペース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、Acceptance Mask レジスタ (CAN_AmxxL と CAN_AMxxH)、Mailbox 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。たとえば、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:

0.2

35. 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:

0.2

36. 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:

0.2

37. 05000309—UART クロックのディスエーブル中に UART THR に書き込みを行うと不正なスタート・ビットが送信される:

説明:

内部 UART クロックがオフのときに(UCEN=0 in UART_GCTL)、UART Transmit Hold レジスタ(UART_THR)が書き込まれると、UART クロックが後でイネーブルされるまで(UART_GCTL 内で UCEN=1)、UART TX ピンがロー・レベルに駆動されます。この時点で、UART ロジックが実際の UART パケット(スタート・ビット、データ、パリティ、ストップ・ビットを含む)を TX ピンへ出力します。

実際には、TX ピンの元の立ち下がりエッジ(UART_THR への書き込みが発生したタイミング)が、接続されているレシーバによりスタート・ビットとして解釈されてしまいます。UART クロックがイネーブルされたとき、UART ロジックはメッセージを正常に送信するため、レシーバがパリティ/ストップ情報の受信を期待している間 UART がデータを出力するので、これによりフレーム・エラーが発生するものと思われます。

対策:

UCEN=0 のとき、UART_THR レジスタへ書き込みを行わないでください。

適用レビジョン:

0.2

38. 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:

0.2、0.3

39. 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:

0.2、0.3

40. 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:

0.2, 0.3

41. 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:

0.2

42. 05000316—EMAC RMII モード:全二重モードで衝突が発生する:

説明:

RMII モードの全二重動作で、TX_EN と CRS_DV がアサートされると、衝突が検出されます。これは、送信データと着信データが同時に得られたことを意味します。トランスミッタを暫くの間低速にして、もう一度試みます。この問題により、送信帯域幅が低下することがあります。

対策:

なし。

適用レビジョン:

0.2

43. 05000321—EMAC RMII モード: TX フレームが半二重でステータス"No Carrier"のエラーを発生する:

説明:

RMII モードの半二重動作で、EMAC_TX_STAT レジスタの No Carrier ビット(CSR)が常にセットされます(キャリアなし)。これにより、TX フレームでステータス TX_OK が報告されなくなります。このため、TX フレームが送信されたことをソフトウェアから知ることができません。この問題は、PHY が CRS_DV のアサートを解除して衝突を防止しなければならないことと、送信で同じ信号が TX_CRS に使われることとの矛盾により発生しています。これと同じ理由で、TX ステータス・ワードで(TX_LOSS ビットがセット)キャリア喪失の偽レポートも発生します。

対策:

なし。

適用レビジョン:

0.2

44. 05000322—10-Base-T 速度の EMAC RMII モード: RX フレームが正常に受信されない:

説明:

10-Base-T 速度の RMII モードで動作するように設定されたイーサネット MAC が RX フレームを正常にデコードできません。

対策:

ネットワークに高速(100-Base-T)および/または MII 接続を使用してください。

適用レビジョン:

0.2、0.3

45. 05000341—イーサネット MAC MDIO 読み出しで IEEE 仕様を満たさない:

説明:

イーサネット MAC (EMAC) MII インターフェースを使って PHY レジスタからデータを読み出す際に、MDC の立ち下がりエッジの後の、システム・クロックの立ち上がりエッジで PHY からの MDIO 信号を EMAC がラッチします。IEEE 802.3 規格では、PHY は MDC の立ち上がりエッジで MDIO を駆動し、0～300ns の出力遅延を持つと規定しています。SCLK/MDC クロックと回路配線の組み合わせによっては、EMAC がただしい MDIO データ値をラッチできないことがあります。

対策:

次の条件を満たして、EMAC が正しい MDIO データをラッチできるようにします:

2 * Tprop + 300ns (または実際の PHY 遅延) < 0.5 * MDC 周期+ tck、ここで、

· Tprop は PHY と Blackfin MII との間の伝搬時間。

· tck は MDC の立ち下がりエッジと SCLK の立ち上がりエッジとの間の遅延(-0.5 * SCLK 周期+ 4ns)。

この条件を満たすために、PLL_DIV レジスタと EMAC_SYSCTL レジスタを設定することにより、それぞれ SCLK と MDC を調節することができます。

適用レビジョン:

0.3

46. 05000350—UART ブート後に UART がディスエーブルされる:

説明:

UART を使用したブート・シーケンスが正常に完了した後にブート・カーネルが UART ポートをディスエーブルします。このため、UART0_DLL レジスタと UART0_DLH レジスタがリセットされてしまうので、ブート時の自動ボー・シーケンスで得られたビット・レート情報をアプリケーションからアクセスすることができません。この動作により、通常行われている、UART ブートの正常完了を通知するアクノリッジをプロセッサからホストへ送信できなくなります。

対策:

アプリケーションから UART ボー検出ルーチンを再起動して、UART ホストへのリンクを再確率する必要があります。

適用レビジョン:

0.2

47. 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:

ペリフェラルのハイバネート・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:

0.2、0.3

48. 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) External Receive Frame Sync (SPORTx_RCR1 で IRFS = 0)
2) Window Offset = 0 (SPORTx_MCMC1 で WOFF = 0)
3) Multichannel Frame Delay = 0 (SPORTx_MCMC2 で MFD = 0)
4) DMA Transmit Packing 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。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
2)内部受信フレーム同期を使用
3) Multichannel Frame Delay > 0 を使用
4) Window Offset > 0 を使用
5) DMA Transmit Packing をイネーブル
6)チャンネル 0 をディスエーブルしない

適用レビジョン:

0.2、0.3

49. 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:

0.2、0.3

50. 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:

0.2、0.3

51. 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 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:

0.2

52. 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:

0.2、0.3

53. 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:

0.2、0.3

54. 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:

0.2、0.3

55. 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(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 命令離す必要があります。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:

0.2、0.3