

ADSP-21369 シリコン・アノーマリについて

これらのアノーマリには、SHARC ADSP-21369 製品レビジョン間の現在、既知の相違、および ADSP-21369 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間の現在、既知の相違を示します。

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(製品表示についてはデータシートを参照してください)。

JTAG コードまたは DSP コードを使って REVPID レジスタを読み出すことにより、シリコン・レビジョンを電氣的に読み出すこともできます。

次の DSP コードを使ってレジスタを読み出すことができます。

```
<UREG> = REVPID;
```

Silicon REVISION	REVPID[7:4]
0.2	0010
0.1	0001
0.0	0000

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各アノーマリ・リストの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
06/16/2009	G	D	Modified anomalies: 08000001 , Added common note on Tools action for all the core related anomalies
01/20/2009	F	D	Added anomaly: 08000026
03/16/2007	E	A	Modified anomalies: 08000001 , 08000014 , 08000015 , 08000016 , 08000017 , 08000018 , 08000019 , 08000020 , 08000024
01/05/2007	D	A	Added anomaly: 08000024 Modified anomaly: 08000020
08/14/2006	C	A	Added anomalies: 08000018 , 08000019 , 08000020 Modified anomalies: 08000014 , 08000017
03/21/2006	B	PrB	Added anomalies: 08000014 , 08000015 , 08000016 , 08000017 Modified anomalies: 08000002 , 08000006
08/10/2005	A	PrB	Initial release

アナログ・デバイセズ社は、提供する情報が正確で信頼できるものであることを期していますが、その情報の利用に関して、あるいは利用によって生じる第三者の特許やその他の権利の侵害に関して一切の責任を負いません。また、アナログ・デバイセズ社の特許または特許の権利の使用を明示的または暗示的に許諾するものでもありません。仕様は、予告なく変更される場合があります。本紙記載の商標および登録商標は、各社の所有に属します。

※日本語データシートは REVISION が古い場合があります。最新の内容については、英語版をご参照ください。

シリコン・アノーマリの一覧

次の表に、ADSP-21369 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	0.0	0.1	0.2
1	08000001	コア・ストールが正常に実行されない	X	X	X
2	08000002	内部メモリに対する DMA の実行中に、ある条件下でメモリ書き込み動作に失敗することがある	X	.	.
3	08000006	tDCAD 仕様の違反	X	.	.
4	08000014	外部メモリ・アクセスと関係する CORE—DMA 間の競合がある場合、命令フェッチに失敗する	X	X	.
5	08000015	266MHz を超える動作ではコア電圧を上げることが必要	X	X	.
6	08000016	SPIBAUD=2 を使うときの SPIDMA の問題	X	X	X
7	08000017	166MHz での SDRAM 動作が tSSDAT 仕様を満たさない	X	X	.
8	08000018	ある条件下で TCK が外部 PM バス・アクセスに影響を与える	X	X	.
9	08000019	PDAP データには外部ポート・ピンを使えないので PDAP には DAI ピンを使う必要がある	X	X	X
10	08000020	ASRC(非同期サンプル・レート・コンバータ)の入力シフト・レジスタ不具合からディジチチェーン接続された TDM モードが影響を受ける	X	X	.
11	08000024	ハード(ホット)リセット後の外部 SDRAM 読み出しの途中でパラレル EPROM/FLASH ブートが発行されると、このブートが失敗する	X	X	X
12	08000026	DB 修飾子を使って IRQx/タイマ割り込みから抜け出るとき、スタックの不正なポップが発生することがある	X	X	X

キー: x=アノーマリが存在するレビジョン
.=適用なし

シリコン・アノーマリの詳細リスト

次のリストに、説明、対策、適用シリコン・レビジョンの識別を含む、ADSP-21369 のすべての既知シリコン・アノーマリを示します。

1. 08000001—コア・ストールが正常に実行されない:

説明:

次に示す特定の条件で、通常実行されるコア・ストールの1つのタイプが実行されません。場合によっては、これにより予期しないコード動作が発生します。

PCSTK load と RTS/RTI の組み合わせ:

通常、PCSTK をロードした直後に RTS/RTI を実行すると、復帰する前に PCSTK のライトバックを待つためストールが発生します。

例 1:

```
[1] PCSTK = DM(I0,M0);  
[2] RTS;
```

例 2:

```
[1] PCSTK = DM(I0,M0);  
[2] RTI;
```

異常動作:

命令 1 でのメモリ・アクセスがメモリ・マップド IOP レジスタに対するものである場合、

または、

命令 1 でのメモリ・アクセスが外部メモリに対するものである場合、

または

DMA が命令 1 内のメモリ・アクセスと同じバンクを同時にアクセスする場合、

ストールが発生しないため、PCSTK からの値を用いることなく RTS/RTI は未知のロケーションへ分岐します。

対策:

命令 1 と命令 2 の間に nop を挿入してください。

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

適用レビジョン:

0.0、0.1、0.2

2. 08000002—内部メモリに対する DMA の実行中に、ある条件下でメモリ書き込み動作に失敗することがある:

説明:

命令がレジスタを変更し、かつ同じレジスタがメモリ書き込みのソースになっており、さらに次のサイクルでのメモリ書き込みと同じブロックに対して DMA が発生すると、次の条件下で、レジスタの古い値ではなく変更された値がメモリに書き込まれます。

レジスタを変更する命令の例と、その同じレジスタがメモリに書き込まれる例を次に示します:

```
R0 = R1-R2, DM(I0,M0) = R0; // The memory access can be either DM/PM
DM(I0,M0) = R0, R0 = PM(I8,M8);
```

問題が発生する条件は、

1.失敗する命令が、1、2、または4命令の長さを持つループの先頭の命令である。

例:

```
lcntr = 8, do (pc,1) until lce;
DM(I0,M0) = R0, R0 = PM(I8,M8); //and DMA occurs in next cycle.
// The new value of R0 (fetched through PM) is written into DM memory
lcntr=0x7, do ST2_IN_BFLY_T2 until lce; //2 instr long loop
f4=f2+f4, dm(i3,m6)=r0, r0=pm(i11,m11);
ST2_IN_BFLY_T2: f4=pass f2, dm(i4,m6)=r4, r2=pm(i11,m11);
```

2.失敗する命令での計算で、次の命令内にある乗算の両オペランドが生成される場合。

例:

```
F0=F0+F4, F1=F0-F4, DM(I0,M0) = R1; // and DMA occurs in next cycle.
// The new value of R1 (output of compute) is written into DM memory
F4=F0*F1;
```

3.失敗する命令の後ろに条件付き分岐が続き、かつ次の2つの内のいずれかが起こる場合:

a.失敗する命令内での計算が、分岐の条件に影響を与える。

例:

```
F0=F0+F4, PM(I10,M10) = R0; // and DMA occurs in next cycle.
//The new value of R0 (output of compute) is written into PM memory
IF EQ JUMP(PC,0x12);
```

b.失敗する命令の前にある命令での計算が分岐の条件に影響を与える。

例:

```
F0=F0+F4;
DM(I0,M0) = R0, R0 = PM(I8,M8); // and DMA occurs in next cycle.
//The new value of R0 (fetched through PM) is written into DM memory
IF EQ JUMP(PC,0x12);
```

4.失敗する命令に浮動小数点乗算が含まれ、かつその後ろに ALU またはシフタで実行される固定小数点オペランド・レジスタを使う計算動作が続く場合。

例:

```
F0=F0*F4, DM(I0,M0) = R0;
F5=FLOAT R1; //The new value of R0 (output of multiply) is written into PM memory
```

対策:

DMA をメモリの別のブロックに移すか、または:

1. ケース 1 の場合、ループを広げてループ長を 4 より長くします。
2. ケース 2、3a、3b、4 の場合、失敗する命令とそれに続く命令の間に無関係な命令を挿入します。

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

適用レビジョン:

0.0

3. 08000006—tDCAD 仕様の違反:**説明:**

166 MHz SDCLK で SDRAM コントローラを使用すると、SDRAM コントローラ仕様 tDCAD (SDCLK からのデータ遅延) が、最初のアクセスでのみ特定のデータ・ピンで満たされません。4.0 ns max と規定されており、6.0 ns まで大きくなることがあります。これにより、SDRAM の仕様を満たせなくなります。この違反により、SDRAM アクセス・エラーが発生したことはまだありません。

対策:

なし。

適用レビジョン:

0.0

4. 08000014—外部メモリ・アクセスと関係する CORE—DMA 間の競合がある場合、命令フェッチに失敗する:

説明:

次の条件で命令フェッチに失敗します:

外部アクセス、命令パイプライン・ストール、DMA ブロック競合の組み合わせが発生すると、命令フェッチに失敗します。下記 3 条件すべてが満たされるときにのみ、失敗が発生します:

1. 連続して実行される 2 つの命令が外部メモリを対象とするデータ・アクセスを行う。
2. プロセッサ・コアが最初のアクセスでストールすると、同時にもう 1 つのストールも発生する。後者のストールとしては、"ADSP-2136x SHARC Processor- Programming Reference"の"PROGRAM SEQUENCER"の章(2005 年 11 月のレビジョン 1.0 のページ 3-11~3-26)で説明するストールが該当します。ただし、次の 2 つのストールは例外です:

- a) DAG レジスタ(index、modifier など)依存性に起因して発生するストール
- b) エミュレータにより発生するストール

外部データ・アクセスを行う命令は、命令パイプラインの"アドレス"ステージでストールすることがあります。これは、外部メモリ・デバイスがアクセス完了通知を待つ必要があるためです。その他のストールについては、上記のプログラミング・リファレンスを参照してください。

3. 内部メモリ・ブロックの競合は、DMA と命令フェッチ(最後の外部アクセス命令に続く 3 番目の命令)との間で発生します。この命令フェッチの失敗により、対応するオペコードが破壊されて、予期しない結果が発生します。

例えば、次の命令シーケンスを考えてみます:

- a) 乗算依存のストール:

```
I1: r12=r5+r4, r0=dm(ext1);
I2: r12=r12*r12, r1=dm(ext2);
I3: <any_instruction>;
I4: <any_instruction>;
I5: <any_instruction>;
```

I1 と I2 は外部メモリ・アクセスを行う連続する命令とします。両命令は、外部アクセスに必要な長い時間のためにパイプライン内の"アドレス"ステージでストールします。I1 がストールすると、同時にコアから発生するもう 1 つの依存性ストールが発生します。このストールは、I2 での乗算の両オペランド(R12)が、R12 に書き込む I1 の実行に依存するためです。これらの両ストールが解消すると、パイプラインが先に進み、I5 の命令フェッチが開始されます。I5 が"フェッチ 1"ステージにある場合に、コードが格納されている内部メモリの同じブロックに対する DMA が発生すると、内部メモリ・ブロックの競合が発生します。このアノーマリにより、I5 のオペコードが破壊されているために命令フェッチが失敗します。

- b) 上記アクセスがループ内の場合:

```
lcntr = 100 , do (PC,3) until lce;
I1: r12=r5+r4, r0=dm(ext1);
I2: r12=r12*r12, r1=dm(ext2);
I3: <any_instruction>;
```

この場合も、同じ理由で問題が発生します。I5 の代わりに、2 番目ループ繰り返して I2 が失敗する点が違います。

- c) 計算での条件付き分岐の依存性:

```
I1: R0=R5+R8, R2 = DM(I0, M0) ;
I2: IF eq JUMP (pc, 3), ELSE R1=DM(I5, M5);
I3: <any_instruction>;
I4: <any_instruction>;
I5: <any_instruction>;
```

この場合、プロセッサ・コアが、別の理由、すなわち I1 での計算に依存した I2 内の条件付き分岐によりストールを発生させます。I5 のフェッチは失敗します。

問題を生じない、DAG 依存ストールの例:

```
a)
I0: I5 = buffer1; // buffer1 in external memory.
I1: r12=r5+r4, r0=DM(I0,M1);
I2: R1 = DM(I5,M5);

b)
I0: R2=R3-R4
I1: IF EQ DM(I1,M1) = R15;
I2: R0 = DM(I1,M2);
```

対策:

- 1.コードと DMA データを内部メモリの異なるブロックに配置してください。
- 2.両命令の間に NOP 命令を挿入して 2 つの命令を分離し、外部メモリ・アクセスを行ってください。NOP の代わりに、外部メモリをアクセスしない命令を使うこともできます。この対策を行うと、外部アクセスでのデータ・スループットが低下して、コード実行速度が低下することに注意してください。

適用レビジョン:

0.0、0.1

5. 08000015—266 MHz を超える動作ではコア電圧を上げることが必要:

説明:

レビジョン 0.0 と 0.1 の内部メモリ内のメモリ・スピード・パスにより、メモリ使用速度が公称コア電圧に制限されています。このため、コア電源電圧を上げる必要があります。この問題は、レビジョン 0.0 と 0.1 のシリコンを 266 MHz を超える速度で動作させる場合にのみ影響します。MQFP パッケージを採用した 266 MHz 製品には影響なく、データシートに規定する電圧で動作します。

0.2 レビジョンの機能テストでは、この問題が解決されていることが示されました。0.2 シリコンのデータシートのタイミング条件と特性は、現在 1.2v の公称コア電圧でキャラクタライズされていますが、レビジョン 0.2 は、Vddint = 1.2v +/-5% で動作させることができます。すべてのレビジョンの 400 MHz での動作では、Vddint = 1.3v +/-5% が必要であることを注意してください。

ADSP-21367/8/9 データシートは、公称 1.2v と 333 MHz のコア・クロック・レートでのタイミング情報に更新される予定です。

対策:

1. 333 MHz を超えてレビジョン 0.0 または 0.1 シリコンを使用する場合には、vddINT 電圧を 1.3 V (+/-5%) に調節してください。

適用レビジョン:

0.0、0.1

6. 08000016—SPIBAUD=2 を使うときの SPI DMA の問題:**説明:**

SPI ペリフェラルを DMA マスター受信モードで使う場合に、SPI を連続 DMA アクセスに設定し、かつアクセスとアクセスの間にディスエーブルしないで SPI ポートを使うと、FIFO と受信バッファがフルになった後でも、SPI が 32 ビット・ワードに対して余分なクロックを発生することがあります。この余分なクロックにより、受信バッファのオーバーフロー・エラーが発生して、32 ビット・ワードが失われてしまうことがあります。これは、SPIBAUD=2 の設定でのみ発生します。

対策:

1)連続する DMA アクセスの間に DMA をディスエーブルしないでください。または
2)連続するアクセスとアクセスの間に DMA をディスエーブルする必要がある場合には、SPI もディスエーブルし、かつ SPI を再イネーブルする前に FIFO と受信バッファをクリアし、その後で SPI をイネーブルし、次に DMA ディスクリプタを設定して DMA をイネーブルしてください。

適用レビジョン:

0.0、0.1、0.2

7. 08000017—166MHz での SDRAM 動作が tSSDAT 仕様を満たさない:**説明:**

SDRAM は最大 133 MHz の SDCLK レートまで正常に動作しますが、タイミングの問題により、SDRAM コントローラの tSSDAT タイミング条件が、166 MHz の SDCLK レートで使用した場合、SDRAM のセットアップ・タイム特性(多くの SDRAM バンダー仕様の tAC)を満たしません。この問題は、166 MHz の SDCLK レートでのみ発生します。SDRAM コントローラを 166 MHz の SDCLK レートで使用すると、データ・アクセス・エラーが発生します。

対策:

SDRAM コントローラを 133 MHz レートで使用してください。

適用レビジョン:

0.0、0.1

8. 08000018—ある条件下で TCK が外部 PM バス・アクセスに影響を与える:**説明:**

プロセッサによっては、TCK の駆動中に、PM データ・バスを使う外部メモリ・アクセスで問題が生ずることがあります。場合によっては、駆動されるアドレス値が変わってしまうことがあります。この問題は、外部メモリに対する PM バス・アクセスのみに影響を与え、駆動される TCK JTAG 入力に依存します。TCK が非アクティブのときに問題は観測されません。TCK は一般に、インサーキット・エミュレータまたはバウンダリ・スキャン・コントローラからのみ駆動されます。プロセッサの通常の非デバッグ動作では、TCK は駆動されないため、この問題は容易に回避できます。

対策:

エミュレータまたはバウンダリ・スキャン・コントローラを使わない通常のプロセッサ動作では、TCK 入力が駆動されないため、この問題は発生しません。エミュレータまたはバウンダリ・スキャン・コントローラをアクティブ(TCK を駆動)にしてプロセッサ・アプリケーションをデバッグする場合に、外部スペースに対する PM バス書き込みが正しくないと見られるときは、DM バスを使って外部書き込みを実行してください。TCK が非アクティブのときこの問題は観測されないため、これはデバッグ中にのみ必要であることに注意してください。

適用レビジョン:

0.0、0.1

9. 08000019—PDAP データには外部ポート・ピンを使えないので PDAP には DAI ピンを使う必要がある:**説明:**

SHARC では、PDAP データ信号を外部ポート・データ・ピンに接続するか、または SRU を使ってこれらの信号を DAI ピンに接続することにより、PDAP 入力の使用方法を選択できます。しかし、シリコン上の問題に起因し、PDAP ではデータ信号を SRU により DAI ピンに接続する必要があり、外部ポート・データ・ピンを介するデータ信号を使うことができません。この問題は、他の外部ポートまたは SDRAM の使用に影響を与えません。これは、PDAP の使用に対する制約になるだけです。

対策:

なし。

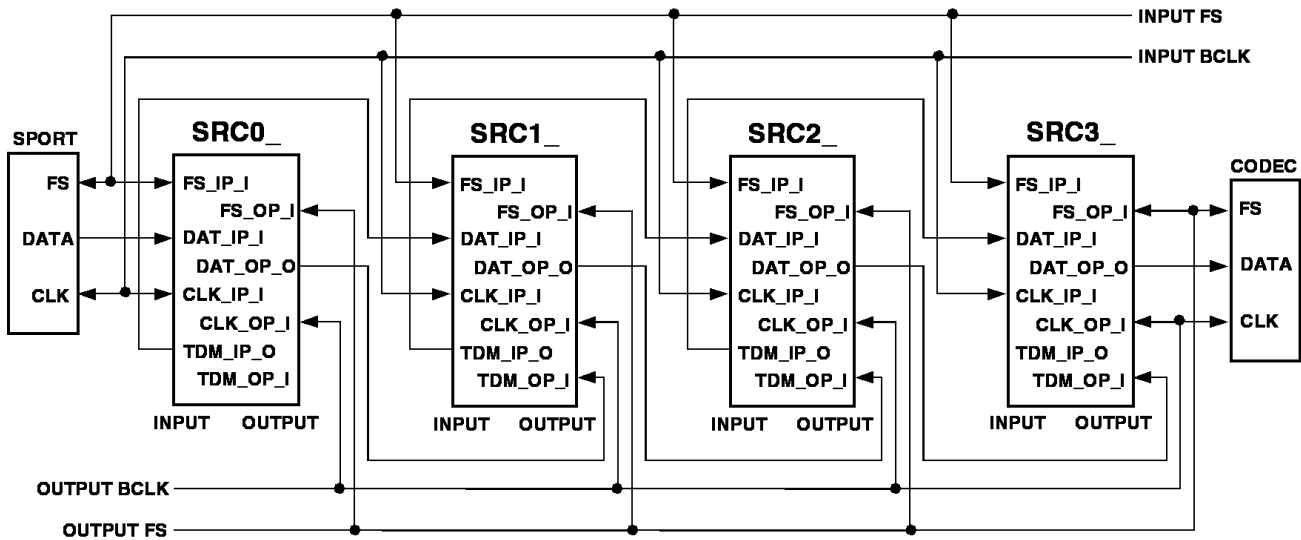
適用レビジョン:

0.0、0.1、0.2

10. 08000020—ASRC (非同期サンプル・レート・コンバータ)の入カシフト・レジスタ不具合からディジーチェーン接続された TDM モードが影響を受ける:

説明:

TDM ディジーチェーン・モードでは、TDM モードを実現するため ASRC を次のように接続することができます:



NOTE: PREFIX ALL SRC SIGNAL NAMES WITH SRCX_.
FOR A COMPLETE LIST OF SIGNAL NAMES, SEE THE DAI CHAPTER IN THE HARDWARE REFERENCE.

各 ASRC には 2 個の 64 ビット・シフト・レジスタがあります(1 つは入力側、もう 1 つは出力側)。ASRC がディジーチェーン接続されたとき、これらの 64 ビット・レジスタは、データのフレーム全体を保持する 1 つの長いシフト・レジスタと見なすことができます。

TDM チェイニング・モードで:

1. 各 ASRC の 64 ビット・シフト・レジスタは、2 個の 32 ビット・チャンネルに分割されます。
2. 1 つの ASRC 入力の TDM 出力がチェーン内の次の ASRC のデータ入力に接続されます。
3. 1 つの ASRC のデータ出力が、チェーン内の次の ASRC の TDM 入力に接続されます。

次の図に、TDM ディジーチェーンの入力側と出力側でのチャンネル割り当てを示します。

32-BIT CHANNEL		1	2	3	4	5	6	7	8	
BIT		0								255
ASRC	IN SIDE	ASRC 3 LEFT	ASRC 3 RIGHT	ASRC 2 LEFT	ASRC 2 RIGHT	ASRC 1 LEFT	ASRC 1 RIGHT	ASRC 0 LEFT	ASRC 0 RIGHT	
	OUT SIDE	ASRC 0 LEFT	ASRC 0 RIGHT	ASRC 1 LEFT	ASRC 1 RIGHT	ASRC 2 LEFT	ASRC 2 RIGHT	ASRC 3 LEFT	ASRC 3 RIGHT	

TDM ディジーチェーン・モードで ASRC を使う場合、チェーン内の各 ASRC の出力で入力データの MSB が失われます。アノマリは、バイパス・モードと非バイパス・モードでの TDM モードに適用されます。バイパス・モードでは、ASRC により入力されるデータが変更されないため、次の例にバイパス・モードの予想不具合パターンを示します。

BYPASS モードの 8 チャンネルによる TDM ディジーチェーンの場合、チェーン内の最終 ASRC の出力では、チャンネル 1 とチャンネル 2 に対して 3 ビット左シフトし、チャンネル 3 とチャンネル 4 に対して 2 ビット左シフトし、チャンネル 5 とチャンネル 6 に対して 1 ビット左シフトします。チャンネル 7 とチャンネル 8 ではデータ・シフトはありません。出力端では、チャンネル 1 とチャンネル 2 では上位 3 ビットが、チャンネル 3 とチャンネル

ル4で上位2ビットが、チャンネル5とチャンネル6では上位1ビットが、それぞれ失われます。

次の8チャンネル入力データのセットに対して、

```
0xF1111100 0xF1111100 // Channels 1 and 2
0xF1111100 0xF1111100 // Channels 3 and 4
0xF1111100 0xF1111100 // Channels 5 and 6
0xF1111100 0xF1111100 // Channels 7 and 8
```

出力データは次のようになります。

```
0x888888FF 0x888888FF //3-bit left shift and 3 MSBs are lost
0xC44444FF 0xC44444FF //2-bit left shift and 2 MSBs are lost
0xE22222FF 0xE22222FF //1-bit left shift and one MSB is lost
0xF11111FF 0xF11111FF
```

上記出力データで、各ワードの下位8ビットは比の情報を持っています。

NON-BYPASSモードのTDMディジチェーンの場合、チェーン内の各ASRCでの上位ビットの喪失によりTDMディジチェーンの出力が完全に破壊されます。

対策:

このアノマリの対策は、TDMディジチェーン・モードでASRCに入力データを提供するデバイス上のソフトウェアでのみ実行する必要があります。ASRCからの出力データに対策を行うことはできません。ソフトウェア内で実施する対策により、アプリケーションに対するコアMIPSがある程度増えます。ADC/CODECのようなデバイスの場合、対策を直接実施することができないことがあります。このようなシステムに対しては、SHARCの2つのシリアル・ポートをこの目的で使う必要があります。一方のシリアル・ポートはADC/CODECからのデータの受信に使い、他方は変更した入力データをASRCへ送信するために使います。

ASRCをTDMディジチェーン・モードで使う場合、ASRCは32ビット入力データの上位24ビットだけを使い、下位8ビットは使いません。このアノマリは、次のようにデータの上位24ビットを下位8ビットへシフトすることにより実施されます。

1. チャンネル1とチャンネル2の入力データを3ビット右へシフトします。データを持っている上位24ビットは、ビット28~5に配置されるようになります。
2. チャンネル3とチャンネル4の入力データを2ビット右へシフトします。データを持っている上位24ビットは、ビット29~6に配置されるようになります。
3. チャンネル5とチャンネル6の入力データを1ビット右へシフトします。データを持っている上位24ビットは、ビット30~7に配置されるようになります。
4. チャンネル7とチャンネル8の入力データは変更する必要がありません。

変更された入力データがTDMディジチェーンに入力されると、このアノマリのために、変更されたデータの上位1ビットがチェーン内の各ASRCで失われます。データを持つ上位24ビットが入力の下位ビットにシフトされるため、実際の上位ビットはチェーン内の各ASRCで失われません。このアノマリのため、ASRCによりデータが左へシフトされるので、出力端では正しく受信されます。

この対策は、ASRCのBYPASSモードとNON-BYPASSモードに適用されます。両ケースとも、24ビットのオーディオ・データが保持されるため、分解能の低下はありません。この対策は、ASRC TDMディジチェーン・モードの位相一致モードにも影響を与えません。これは、位相情報は出力SRCシフト・レジスタの下位8ビットに存在し、このシフト・レジスタはこのアノマリの影響を受けないためです。

次の実際の8チャンネル入力データのセットに対して、

```
0xF1111100 0xF1111100 // Channels 1 and 2
0xF1111100 0xF1111100 // Channels 3 and 4
0xF1111100 0xF1111100 // Channels 5 and 6
0xF1111100 0xF1111100 // Channels 7 and 8
```

この対策により変更されたデータは次のようになります:

```
0xFE222220 0xFE222220 // Channels 1 and 2 with three bit right shift
0xFC444440 0xFC444440 // Channels 3 and 4 with two bit right shift
0xF8888880 0xF8888880 // Channels 5 and 6 with one bit right shift
0xF1111100 0xF1111100 // Channels 7 and 8
```

出力端でのデータは、BYPASSモードで次のようになります:

```
0xF11111FF 0xF11111FF // Channels 1 and 2
0xF11111FF 0xF11111FF // Channels 3 and 4
0xF11111FF 0xF11111FF // Channels 5 and 6
0xF11111FF 0xF11111FF // Channels 7 and 8
```

適用レビジョン:

0.0、0.1

11. 08000024—ハード(ホット)リセット後の外部 SDRAM 読み出しの途中でパラレル EPROM/FLASH ブートが発行されると、このブートが失敗する:

説明:

SHARC プロセッサが外部 SDRAM メモリからデータをフェッチ中に、ハード(ホット)リセットが起動されると、SDRAM デバイスは/RESET がアサートされている間中データ・バスの駆動を続けるため、DSP が/RESET から抜け出した後でも外部データ・バスの解放に失敗します。

DSP がパラレル外部 FLASH または EPROM (同じデータ・バスの一部を外部 SDRAM と共用)を使ってブートするように設定されている場合、FLASH メモリからブートしようとする DSP と、データ・バスの駆動を続ける SDRAM との間で外部データ・バス上でのバス競合が発生して、不正なブート・データがロードされるため、ブート動作は失敗します。一旦、このような状態が発生すると、後続 DSP リセットによりこの状況を抜け出すことができません。この状態から回復するためには、電源のオン/オフ・シーケンスが必要になります。

マイクロコントローラまたはホスト・プロセッサが SHARC の/RESET ピンを制御している場合は、マイクロコントローラ/ホスト・プロセッサと DSP との間のシンプルな通信プロトコルにより、/RESET がアサートされる前に、DMA 転送/外部データ・フェッチの完了を確認するため、この問題を回避することができます。

DSP 動作の通常モードでは、この問題に遭遇する確率が大幅に小さくなることに注意してください。例えば、

- キャッシュがイネーブルされた場合(MODE2 レジスタの CADIS ビットがセットされない)
- SDCTL レジスタのビット 31(NO_BSTOP ビット)がセットされて、SDRAM からのバースト・モード・アクセスがディスエーブルされた場合

上記条件により、この問題に遭遇する確率は小さくなりますが、これらを対策として推奨することはできません。これについては、対策のセクションに記載する技術を参照してください。

このアノマリは、SDRAM メモリへの書き込みでは観測されず、外部 SDRAM からの読み出しでのみ観測されます。このアノマリは、パワーオン・リセットやソフトウェア・リセットなどのその他のリセットでも観測されません。

対策:

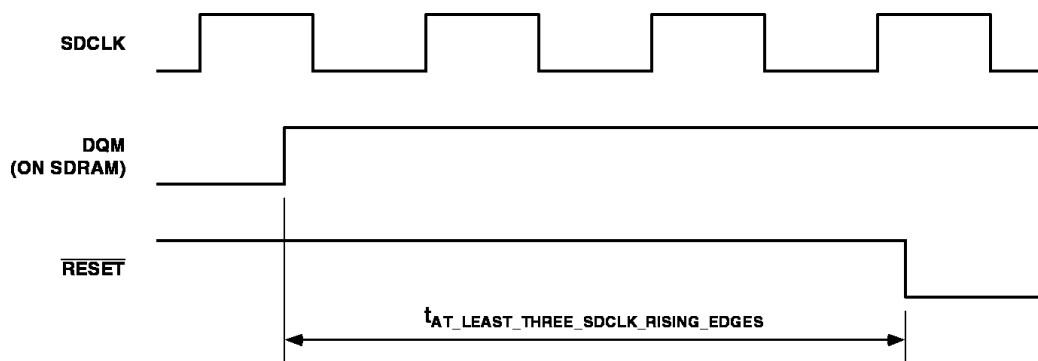
ソフトウェア:

DSP を SPI ポート(マスターまたはスレーブ)を使用するブートに設定して、ローダ・カーネル内で次のシーケンスを実行します:

- SDCTL レジスタと SDRRC レジスタを適切な値に初期化します。
 - 外部 SDRAM メモリのダミー読み出しアクセスを実行します。
- これは、SDRAM コントローラの状態と外部 SDRAM メモリの状態を再設定および再初期化する効果を持つため、問題状態に遭遇することを回避します。

ハードウェア:

SDRAM の DQM 信号と DSP への/RESET_IN との間に少なくとも 3 SDCLK クロック・サイクルの外部遅延を与えても、問題を回避することができます(下図参照)。



適用レビジョン:

0.0, 0.1, 0.2

12. 08000026—DB 修飾子を使って IRQx/タイマ割り込みから抜け出るとき、スタックの不正なポップが発生することがある:**説明:**

遅延分岐修飾子(DB)を使って、IRQx (ハードウェア)割り込みまたはタイマ割り込みの割り込みサービス・ルーチンからリターンする場合、ステータス・スタックからの ASTATx/ASTATy/MODE1 レジスタの自動ポップがおかしくなる。

このアノーマリの影響を受ける特定の命令は、"RTI (DB);"と"JUMP (CI) (DB);"です。

このアノーマリは、IRQx 割り込みとタイマ割り込みにのみ影響を与えます。これら以外に、シーケンサに値をステータス・スタックにプッシュさせる割り込みはありません。

対策:

IRQx またはタイマ ISR から抜け出る命令で(DB)修飾子を使わないでください。遅延スロット内の命令は、分岐の前に移動させてください。

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

適用レビジョン:

0.0、0.1、0.2