

ADSP-21364 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-21364 製品レビジョン間の現在、既知の相違、および ADSP-21364 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]
0.5	0101
0.3	0011
0.1	0001

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
08/13/2009	N	D	Modified anomalies: 07000009
06/02/2009	M	D	Modified anomalies: 07000009 , Added common note on Tools action for all the core related anomalies
02/25/2009	L	D	Added anomalies: 07000021
07/19/2007	K	0	Modified anomalies: 07000010 - Removed some special characters
12/28/2006	J	0	Removed anomalies: 07000020-This information is added to the HRM
12/05/2006	I	0	Modified anomalies: 07000019 -Changed the Title
11/24/2006	H	0	Added anomalies: 07000020
11/09/2006	G	0	Modified anomalies: 07000019 -Added the Workaround
09/28/2006	F	0	Added Anomalies: 07000019
06/21/2006	E	0	Modified Anomalies: 07000011

NR002951N

アナログ・デバイス社は、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・デバイス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。
©2009 Analog Devices, Inc. All rights reserved.

アナログ・デバイス株式会社

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・アノマリの一覧

次の表に、ADSP-21364 アノマリの一覧と各アノマリの適用されるシリコン・レビジョンを示します。

番号	ID	説明	0.1	0.3	0.5
1	07000005	パラレル・ポート・バスの競合	x	.	.
2	07000006	ある条件で遅延間接ジャンプと呼び出しが失敗する	x	.	.
3	07000007	SPDIF レシーバ出力データ順が変化する	x	x	.
4	07000008	命令キャッシュの初期化が必要	x	x	.
5	07000009	正しく実行されないコア・ストールがある	x	x	x
6	07000010	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	x	x
7	07000011	SPI を最大 SPICLK 周波数で受信 DMA モードで使用すると、余分なワードに対して余分なクロックを発生することがある	x	x	x
8	07000019	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	x
9	07000021	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-21364 のすべての既知シリコン・アノーマリを示します。

1. 07000005—パラレル・ポート・バスの競合:

説明:

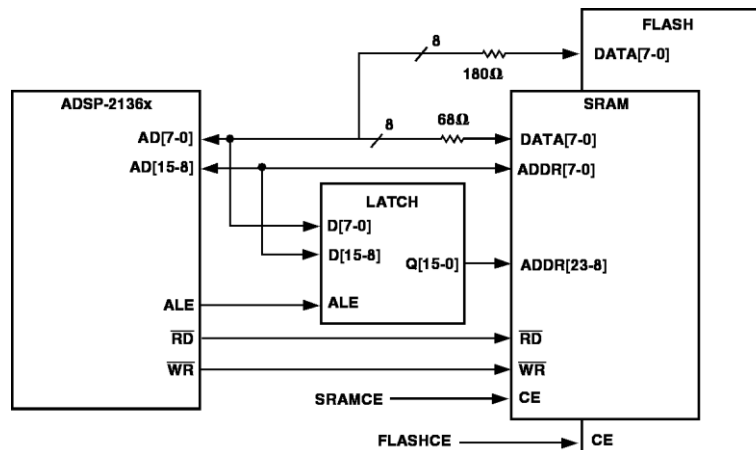
ADSP-2136x プロセッサのパラレル・ポートでバス競合問題が見つかっています。DMA-READ 転送の使用中に外部メモリ・アドレス・ページを超えると、ALE サイクルが挿入されて、外部ラッチが保持している上位アドレス・ビットが更新されます。パラレル・ポートが読み出しサイクルから ALE サイクルへ変化するとき、外部 SRAM または FLASH がバスをスリー・ステートにするために十分な時間が確保できません。/OE のアサーション解除からバスのスリー・ステートまでのタイミングは一般に、SRAM デバイスよりフラッシュの方が長いため、SRAM のケースと FLASH のケースに対して別々の検査が必要になります。

対策:

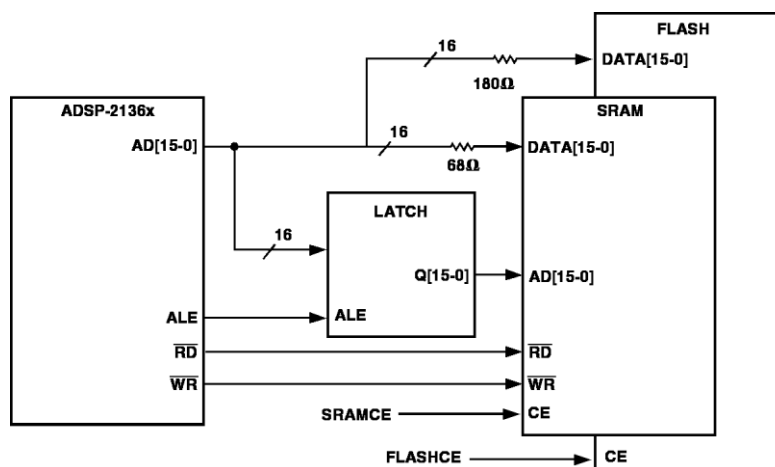
a) SRAM の競合：ハードウェアによる対策:

一般に 8ns の SRAM では、/OE のアサーション解除からスリー・ステートになるまで 5~7ns 必要です。ADSP-2136x では約 4.0 ns しか確保していないため、3.0ns の競合が発生する可能性があります。この時点での推奨される対策は、下図のように SRAM データ・ラインの前に直列抵抗を接続することです。この抵抗は、最大ピン電流を約 22mA に制限すると同時にタイミング条件を満たす十分高速な立ち上がり/立ち下がりレートを提供します。推奨抵抗値は 68Ω です。このオプションはフル速度で動作し、ソフトウェアの変更は不要です。

メモリ読み出し—パラレル・ポート8ビット:



メモリ読み出し—パラレル・ポート16ビット:



b) SRAM の競合: ソフトウェアによる対策 (1):

パラレル・ポート・ホールド・サイクルをイネーブルすることにより対応できます。これにより、外部デバイスのディスエーブルに対してさらに 6.5ns が追加されます(PPCTL レジスタの PPBHC ビットをセットするとバス・ホールド・サイクルが選択されます)。このオプションにより、SRAM 転送あたり 5ns 増加するため、全体の外部帯域幅が低下します。その他、このオプションはソフトウェアには影響ありません。

ソフトウェアによる対策 (2):

DMA 転送が常に 256 バイト (または以下) の外部バースト (8 ビット・モード) に収まるようにソフトウェアを構成します。これにより、読み出し後の ALE サイクルがなくなります。このオプションでは、DMA アクセスで外部ページを超えないようソフトウェアを記述する必要がありますが、ページ内ではフル速度動作させることが可能です。

c) FLASH の競合: ハードウェアによる対策:

一般に FLASH メモリでは、OE のディスエーブルからバス・スリー・ステートまで 30ns を要します。複数の FLASH デバイスのベンチ・テストでは、このディスエーブル時間は一般に 10ns より小さいことが分かりました。ただし、ワースト・ケース仕様での安全な動作のためには、規定の最大 FLASH ディスエーブルを満たす必要があります。FLASH のスリー・ステート時間が 30ns だとしても、ALE の立ち下がりがエッジでバスをまだ駆動しているため、ALE により上位アドレスがラッチされます。このため、上記 SRAM のケースとは異なり FLASH の場合は、競合中も有効ロジック・レベルが AD[15:0] ラインからラッチへ供給される必要があります。これを実現するため、大きな抵抗を使う必要があります。推奨抵抗値は 180Ω です。この抵抗値は、デジタル・ラインでの輻射を削減する目的で使用する場合に比べると大きい値ですが、FLASH は低速デバイスであるため、立ち上がり/立ち下がり時間の増加は相応して小さいものです。正しく動作させるためには、この抵抗により増加する遅延を待ち状態の追加により考慮する必要があります。1 個の FLASH を使ったシミュレーションでは、遅延が 5ns より小さいことが示されています。これらの抵抗値を使うことができない場合は、FLASH のデータ・パスにスリー・ステート・バッファを接続することが可能です。バッファの通過遅延は比較的小さいのですが、このようなバッファのターンオフ時間は必要とされる 2.5ns を超えるため、ADSP-2136x デバイスとバッファの間に小さい直列抵抗を接続して、パッド電流を抑える必要があります。

d) FLASH の競合: ソフトウェアによる対策:

パラレル・ポート・バス・ホールド・サイクルは、FLASH に対しては短過ぎます。FLASH のアクセスを行うときは、256 バイト (または以下) の外部バースト(8 ビット・モード)で DMA を実行する方法が推奨されます。パラレル・ポート・ブート時、プロセッサは連続する DMA 読み出しで 8 ビット・モードを使うため、競合が発生することに注意してください。

適用レビジョン:

0.1

2. 07000006—ある条件で遅延間接ジャンプと呼び出しが失敗する:

説明:

ブロック n に対する遅延間接ジャンプ (または間接呼び出し)の後ろにブロック n に対する dm アクセスが続き、さらに dm アクセスと同時にブロック n に対する DMA アクセスも発生すると、このブロック n に対する遅延間接ジャンプ (または間接呼び出し)が機能しません。

たとえば、

```
back: jump (m14,i12) (db); // M14 + I12 is in Block 0
      r0=dm(i6,m7); //I6 + M7 is to Block 0 & DMA access to Block 0 comes here
      nop;
      nop;
      jump back;
      nop; nop;
```

上記例で、ロケーション M14+I12 からの命令フェッチは発生しません。

注: これらの命令は、DM アクセスおよび DMA アクセス (DMA データまたは TCB)と同じメモリ・ブロック内にある必要があります。i12、i6、m14、m7 の値は、アクセスが同じメモリ・ブロック内に留まるように初期化される必要があります。

対策:

遅延間接分岐の後ろに nop を追加すると、この問題は解決されます。

DMA データ (チェイニングを使用の場合は TCB も)を別のメモリ・ブロックへ移動してください。

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:

0.1

3. 07000007—SPDIF レシーバ出力データ順が変化する:**説明:**

SPDIF レシーバ (DIR)は、SPDIF レシーバ出力内のすべてのサンプルに対して、右チャンネル情報を送った後に左チャンネル情報を送りますが、これでも出力は I2S 規格を満たしています。

```
Ln represents the left channel from frame n
Rn represents the right channel from frame n
t=0 -----> t=3*Fs
```

SPDIF RX が受信して、I2S フォーマットで次のように入力する必要があります。

L0 -> R0 -> L1 -> R1 -> L2 -> R2 -> L3 -> R3 -> ...

異常出力は次のようになります。

R0 -> L0 -> R1 -> L1 -> R2 -> L2 -> R3 -> ...

対策:

SPDIF レシーバ出力がシリアル・ポートを介してプロセッサの内部メモリへ入力される場合は、データをソフトウェアで処理して、左チャンネル・データをメモリへ書き込んだ後に右チャンネル・データを書き込むことにより、順序の一時的変化を直します。AC-3 などのようなエンコードされたデータ・ストリームを受信する SPDIF レシーバを使うアプリケーションでは、サンプルの一時的順序は非常に重要です。デコーディング・アルゴリズムとストリーム検出アルゴリズムが機能するためには、この対策の実施、または入力データの順序変化に合わせてアルゴリズムを変更することが必要です。

SPDIF ストリームが 2 チャンネルの PCM オーディオ・データを構成する幾つかのアプリケーションでは、対策は不要です。データは正しい I2S チャンネルに保持されるため、右と左チャンネルの交換はありません。

適用レビジョン:

0.1、0.3

4. 07000008—命令キャッシュの初期化が必要:**説明:**

ADSP-2136x プロセッサの命令キャッシュが、パワーアップ時に正しく初期化されません。このためキャッシュからの最初のデータ読み出し(キャッシュ・ヒット)を正しく実行できません。これにより、プロセッサが無効な命令を実行する可能性があります。この種類のシナリオは、ループ内に PM データ・アクセスを含むコードに当てはまります(ループの 2 回目の繰り返しで、プログラム・シーケンサにより PM データ・アクセス命令がキャッシュから読み出されますが、この問題のためにキャッシュからの命令が無効になります)。

対策:

この問題に対する対策は、プロセッサの初期化ルーチンの中に次のコードを配置することです。このコードがキャッシュを正しく初期化し、このコードを実行した後、命令キャッシュは正しく動作します。命令キャッシュを信頼して使用するためには、このコードを実行しておくことが重要です。

```
BIT SET MODE2 CADIS;           // Disable the cache
NOP;                           // One nop for effect latency
READ CACHE 0;                  // Read cache instruction
BIT CLR MODE2 CADIS;          // Re-enabling the cache.
```

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:

0.1、0.3

5. 07000009—正しく実行されないコア・ストールがある:**説明:**

次に示す特定の条件で、通常実行される 3 つのタイプのコア・ストールが実行されません。場合によっては、このために予期しないコード動作が発生します。

1.乗算のストール:

通常の動作では、乗算命令の両オペランドが前の命令で乗算器または ALU により発生される場合、パイプラインが 1 サイクル間ストールする必要があります。

```
[1] F0=F0+F4, F1=F0-F4, R14 = DM(I0,M0);
[2] F4=F0*F1; <---- this should stall
```

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 内でのメモリ・アクセスと同じバンクを DMA が同時にアクセスする場合、
ストールは発生しないので、F4 内で正しい積が得られません。

2.レジスタ書き込みモード 1/ モード 2:

通常の動作では、MODE1 レジスタまたは MODE2 レジスタに対するすべての書き込みの後にストール・サイクルが追加される必要があります。

- MODE1 レジスタと MODE2 レジスタの値がビット操作命令により変更されたときにもストールが追加されます。

- PUSH STS 命令または POP STS 命令が実行されたとき、またはシーケンサがステータス・スタックの push/pop を使う ISR (IRQ 割り込みやコア・タイマ割り込みのようなハードウェア割り込み)への分岐/復帰のときか、それからのリターンのとき、MODE1 の値も変化します。このような場合、パイプラインは 1 サイクル間ストールします。

Example1:

```
[1] MODE1= DM(I0,M0); /*Enable bit reverse addressing for I8 */
[2] PM(I8,M8)=R14; /* stalls for a cycle but unaffected by mode setting */
[3] PM(I8,M8)=R14; /* performs bit reversed mode of addressing */
[4] ...
```

Example2:

```
[1] MODE1= DM(I0,M0); /*Enable bit reverse addressing for I8 */
[2] r0 = r1 + r2; /* stalls for a cycle */
[3] PM(I8,M8)=R14; /* performs bit reversed mode of addressing */
[4] ...
```

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 内でのメモリ・アクセスと同じバンクを DMA が同時にアクセスする場合、
ストールが発生しないため、命令 3 はモード変更の影響を受けません。

3.PCSTK ロードと RTS/RTI の組み合わせ:

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。

Example1:

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

Example2:

```
[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 内でのメモリ・アクセスと同じバンクを DMA が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。

注: 乗算ストールのケースは、シリコンの 0.4 および 0.5 レビジョンで解決されています。

対策:

ケース 1 とケース 3 の場合は、命令 1 と命令 2 の間に nop を追加してください。ケース 2 の場合は、命令 2 と命令 3 の間に nop を追加してください。

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:

0.1、0.3、0.5

6. 07000010—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:

1. `R0 = R1-R2, DM(I0,M0) = R0; // The memory access can be either DM/PM`
2. `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 個、2 個、または 4 個分の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. 失敗する命令の計算内で、次の命令の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オペランド・レジスタを使う演算動作が続く場合。

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:

0.1、0.3、0.5

7.07000011—SPI を最大 SPICLK 周波数で受信 DMA モードで使用すると、余分なワードに対して余分なクロックが発生することがある:**説明:**

次の 3 つの条件を満たすとき、SPI は余分なワードに対して余分なクロックが発生することがあります。

1. SPI を 32 ビットの受信マスター DMA モードに設定して、連続的に DMA 転送を行う。
2. SPI 自体をディスエーブルしないで DMA 転送を連続的に行う間に、SPI DMA をディスエーブルした後に再イネーブルする。
3. SPI が最大 SPICLK 周波数 (SPIBAUD = 2) で動作する。

上記 3 つの条件が満たされると、SPI は FIFO および受信バッファがフルになった後でも余分なワードに対してクロックが発生することがあります。

次のワードに対するこのクロックにより、受信バッファ・オーバーフロー・エラーが発生して、次の受信データ・ワードが失われてしまうことがあります。

対策:

1. 連続する DMA アクセスの間に SPI DMA 転送をディスエーブルしないでください。
2. 連続する各 DMA 転送間で SPI DMA をディスエーブルする必要があるアプリケーションでは、次のシーケンスに従ってください。
 - SPI DMA をディスエーブルします。
 - SPI をディスエーブルします。
 - FIFO と受信バッファをクリアします。
 - DMA ディスクリプタを設定して、SPI をイネーブルします。
 - SPI DMA をイネーブルします。

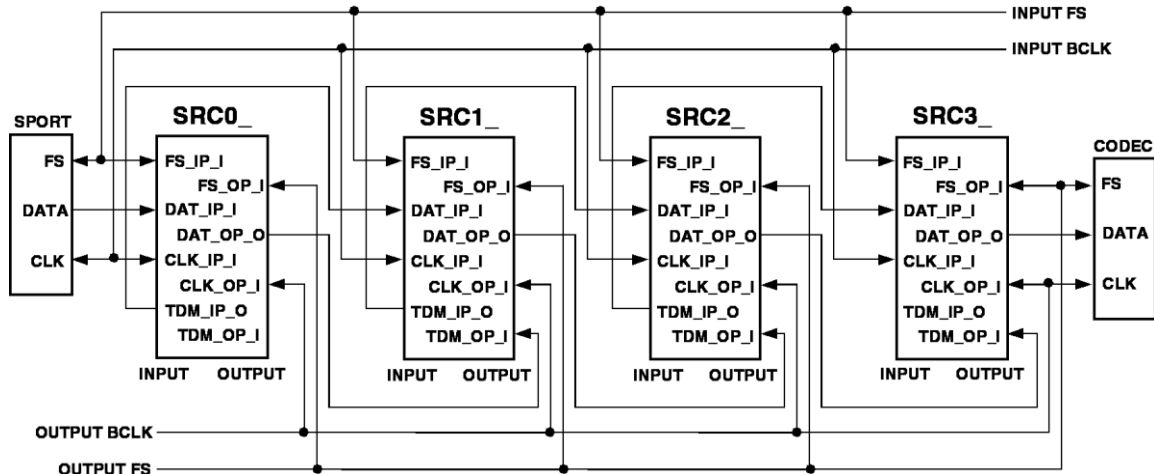
適用レビジョン:

0.1、0.3、0.5

8. 07000019—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。このアノマリは、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 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 ディジーチェーンの場合、チェーン内の各 ASRC での上位ビットの喪失により TDM ディジー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 ディジーチェーン・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 ディジーチェーン・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 ディジー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 ディジーチェーン・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:

0.1、0.3、0.5

9. 07000021—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異常動作します。

このアノ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:

0.1、0.3、0.5