

**ADSP-21371/ADSP-21375 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-21371/ADSP-21375製品レビジョン間の現在、既知の相違、およびADSP-21371/ADSP-21375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]
0.0	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
06/07/2009	F	Rev B	Added anomalies: <a href="#">09000023</a> and <a href="#">09000024</a> .
02/02/2009	E	Rev B	Revised anomaly <a href="#">09000018</a> . Added anomalies: <a href="#">09000019</a> , <a href="#">09000020</a> , <a href="#">09000021</a> , and <a href="#">09000022</a>
06/05/2008	D	Rev A	Added anomalies: <a href="#">09000014</a> , <a href="#">09000015</a> , and <a href="#">09000018</a> .
03/21/2008	C	Rev A	Consolidated anomaly lists for ADSP-21371/ADSP-21375. Added new anomalies <a href="#">09000009</a> and <a href="#">09000011</a> . Consolidated old anomalies <a href="#">09000006</a> (for ADSP-21371) and <a href="#">09000001</a> (for ADSP-21375) into <a href="#">09000013</a> , and <a href="#">09000007</a> (for ADSP-21371) and <a href="#">09000002</a> (for ADSP-21375) into <a href="#">09000007</a> .
01/03/2007	B	PrA	Removed anomalies: <a href="#">09000008</a> -This information is added to the HRM
11/06/2006	A	PrA	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。

NR003347F

©2009 Analog Devices, Inc. All rights reserved.

## シリコン・アノマリの一覧

次の表に、ADSP-21371/ADSP-21375 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。

No.	ID	説明	0.0
1	09000007	バウンダリ・スキャン・コントローラを使ってシフト出力したとき、IDCODE レジスタの先頭ビットがロジック 1 にならない	x
2	09000009	外部メモリ・アクセス時に BTC でエラーが発生する	x
3	09000011	特別な条件下で、外部メモリから内部メモリへの間接分岐により命令キャッシュが破壊されることがある	x
4	09000013	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
5	09000014	遅延分岐(DB)スロットでの条件付き外部データ・アクセスが正しく実行されない	x
6	09000015	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
7	09000018	特定の乗算動作を外部メモリ・アクセスと同じ命令に含めないようにする必要がある	x
8	09000019	外部コア・データ・アクセスがある場合 EMUCLK/EMUCLK2 カウンタが正しく動作しない	x
9	09000020	外部メモリから実行中の PMDA 命令に対して割り込みが発生すると、正しくない命令アドレスがキャッシュされる	x
10	09000021	外部メモリ・アクセスの直後の IOP レジスタ・アクセスが機能しない	x
11	09000022	外部データ・アクセスの実効レイテンシが 1 サイクルではなく 2 サイクルになる	x
12	09000023	内部メモリから LCNTR、CURLCNTR、LADDR へ書き込みを行うと、DMA ブロックの競合が発生したときに失敗する	x
13	09000024	特定のコア・ストールが期待通りに実行されない	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-21371/ADSP-21375 のすべての既知シリコン・アノーマリを示します。

### 1. 09000007—バウンダリ・スキャン・コントローラを使ってシフト出力したとき、IDCODE レジスタの先頭ビットがロジック 1 にならない:

#### 説明:

新しく設けたデバイス識別レジスタ(IDCODE)の LSB は、IEEE 1149.1 規格に基づき 1 である必要があります。

このケースでは、シフト出力された先頭ビットが LSB ではなく MSB であるため、テスト・スキャン・コントローラによるこのレジスタの有無の自動検出(通常、先頭ビットがシフト出力されると直ちに開始されます)が失敗します。

#### 対策:

アナログ・デバイセズから提供するプロセッサ固有の BSDL ファイルを使い、IDCODE レジスタの LSB (ビット 0)に対して発生する構文エラーを無視してください。

#### 適用レビジョン:

0.0

### 2. 09000009—外部メモリ・アクセス時に BTC でエラーが発生する:

#### 説明:

プロセッサ・コアが外部メモリ・データをアクセスしている場合、BTC (Background Telemetry Channel)が動作しません。外部メモリの実行は、BTC の動作を阻害しません。

#### 対策:

プロセッサが外部メモリからデータをフェッチすることが予測される場合には、BTC を使わないでください。

#### 適用レビジョン:

0.0

### 3. 09000011—特別な条件下で、外部メモリから内部メモリへの間接分岐により命令キャッシュが破壊されることがある:

#### 説明:

キャッシュをイネーブルして外部メモリからコードを実行する時、次のすべての条件が成立する場合、

1. 間接分岐命令の処理(モディファイ・レジスタとインデックス・レジスタの対により提供される分岐アドレスを持つ JUMP/CALL)。
2. 間接分岐命令のターゲット・アドレスが内部メモリ・アドレス空間内のロケーションを指定する。
3. 間接分岐の直前に命令フェッチ・シーケンスのリダイレクトが発生する(これは、JUMP、CALL、RTS などの明確な命令により、または割り込みのサービス、ループ内でのループバックなどにより、暗黙的に発生します)。

そうすると、間接分岐命令の後ろに続く 3 番目の命令がプログラム・シーケンサにより不正にキャッシュされます。

次の例に、これらの条件を満たす状況を示します:

```
// following code resides in external memory
jump _some_location;
jump(m12,i12); // target points to internal memory
nop;
nop;
r4=5; // <-- this instruction is improperly cached
```

もう 1 つの例:

```
// following code resides in external memory
lcntr=<count>, do (pc, end) until lce;
...
...
call(m12,i12); // target points to internal memory
nop;
end: nop;
r4=5; // <--- this instruction is improperly cached
```

#### 対策:

1. 外部メモリ内では、間接分岐命令に続く 3 番目の命令を実行しないようにしてください。次に示すコード・シーケンス例では、これを実現しています:

```
// following code resides in external memory
jump _some_location;
call (m12,i12); // target is in internal memory
nop;
jump(pc,2);
<instruction_3>; // <-- this instruction is improperly cached
<instruction_4>; // start of useful code.
```

上記対策が機能するためには、<instruction\_3>が他の分岐のターゲットにならないようにすることに注意してください。

2. 内部メモリをターゲットとして持つ間接分岐を外部メモリ内のコード内に配置しないでください。分岐のターゲット・アドレスが内部メモリ・ロケーションを指さない場合は、このアノーマリは発生しません。
3. 次のロケーションのいずれかに "**FLUSH CACHE;**" 命令を追加してください:
  - 問題となる間接分岐命令のすべてのターゲット・アドレス。
  - 問題となる各間接分岐命令の直前。
  - 問題となるすべての間接分岐を(DB)修飾子を使用する遅延分岐へ変換して、2 つの遅延分岐スロット内のいずれかに "**FLUSH CACHE;**" 命令を配置します。
4. **MODE2** レジスタの CADIS ビットを設定してキャッシュ機能をディスエーブルしてください。この場合、すべての命令キャッシュ機能がディスエーブルされるため(内部メモリと外部メモリ)、性能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

#### 4. 09000013 -ハード(ホット)リセット後の外部 SDRAM 読み出しの途中にパラレル EPROM/FLASH ブートが発行されると、このブートが失敗する:

##### 説明:

プロセッサが外部 SDRAM メモリからのデータをフェッチ中に、または外部 SDRAM メモリからのコードを実行中に、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 ADSP-21375 の /RESET ピンを制御していて、DSP のリセットを実行する場合には、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、DSP のプログラム実行を内部メモリにジャンプさせるか(外部コード実行の場合)、または /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(外部 SDRAM からのデータ・アクセスの場合)、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、およびシーケンサによる外部 SDRAM からのフェッチ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、ソフトウェア・リセット、ランニング・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:

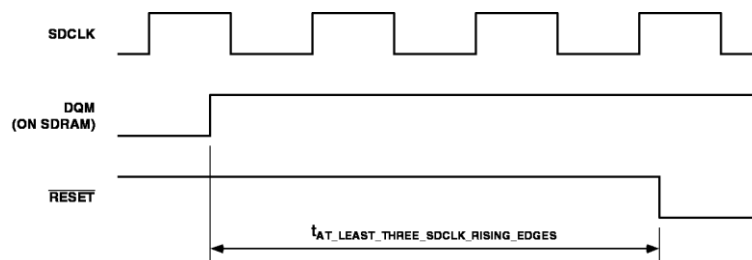
DSP を SPI ポート(マスターまたはスレーブ)を使用するブートに設定して、ローダ・カーネル内で次のシーケンスを実行します:

- 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

**5. 09000014—遅延分岐(DB)スロットでの条件付き外部データ・アクセスが正しく実行されない:****説明:**

条件の評価を誤るために、分岐命令の遅延スロット内の条件付き外部データ・アクセス命令(JUMP/CALL/RTS/RTI など)が正しく実行されないことがあります。これは、内部メモリと外部メモリでの実行で発生します。

たとえば、

```
// Code residing in either internal or external memory
jump (some_addr) (DB);
<instruction_1>;
if eq r0=dm(ext_addr); //Condition evaluation may be wrong
```

**対策:**

すべての分岐命令の遅延スロット内で条件付き外部データ・アクセス命令を使わないでください。

**適用レビジョン:**

0.0

**6. 09000015—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。このアノマリは、内部メモリと外部メモリでの実行で発生します。

**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。

**適用レビジョン:**

0.0

**7. 09000018—特定の乗算動作を外部メモリ・アクセスと同じ命令に含めないようにする必要がある:****説明:**

乗算結果レジスタ(MRF/MRB)をディステネーションとして使用する特定の乗算動作は、外部メモリ・アクセスと同じ命令に含めることはできません。

こういった動作が外部データ・アクセスと組み合わせると、MR\*レジスタは期待する値で更新されません。これは、内部メモリと外部メモリでの実行で発生します。

さらに、乗算器動作が外部メモリ・データ・アクセスと同じ命令に含まれると、対応するステータス・フラグ ASTATx/y レジスタと STKYx/y レジスタが正しく更新されないため、該当する割り込みも開始されません。

**対策:**

これらの特定の乗算動作は、外部メモリ・アクセスから分離した命令で行ってください。

**適用レビジョン:**

0.0

**8. 09000019—外部コア・データ・アクセスがある場合 EMUCLK/EMUCLK2 カウンタが正しく動作しない:****説明:**

外部メモリが関係するコア・アクセスに対して、EMUCLK カウンタと EMUCLK2 カウンタが正しくインクリメントしません。このため、外部データ・アクセスがある場合、カウント値は実際の間隔より小さくなります。このアノーマリは、外部メモリが関係する、コアからのデータ・アクセスに固有なものです。DMA アクセスおよび/または命令アクセスには影響ありません。

**対策:**

1. 外部メモリへのコア・アクセスが必要な場合には他のカウンタを使ってください。
2. EMUCLK カウンタと EMUCLK2 カウンタの使用中に、外部メモリが関係するコア・アクセスを回避してください。

**適用レビジョン:**

0.0

**9. 09000020—外部メモリから実行中の PMDA 命令に対して割り込みが発生すると、正しくない命令アドレスがキャッシュされる:****説明:**

外部メモリから PMDA 命令(PM()アクセスを含む命令)を実行中に、コアに対して割り込みが発生すると、その時点でキャッシュされる命令が正しくキャッシュされません。このアノーマリは、PMDA 命令を含む命令に限定されることに注意してください。外部メモリから実行中のすべての命令に影響するものではありません。

**対策:**

1. ユーザ・アプリケーションに外部メモリから実行される PMDA 命令が含まれ、さらに割り込みが発生して、この期間に実行しなければならない場合には、影響を受ける割り込みサービス・ルーチン(ISR)内の先頭命令として"flush cache"命令を挿入してください。
2. MODE2 の CADIS ビットを使ってキャッシュをディスエーブルしてください。これにより、内部メモリ・キャッシュと外部メモリ・キャッシュがディスエーブルされるため、性能が低下します。
3. 可能な場合、外部メモリから実行される PMDA 命令の付近でグローバル割り込みをディスエーブルした後に再度イネーブルしてください。割り込みをディスエーブルする際の遅延を考慮するように注意してください。

**適用レビジョン:**

0.0

**10. 09000021—外部メモリ・アクセスの直後の IOP レジスタ・アクセスが機能しない:****説明:**

IOP レジスタにアクセスする命令が外部メモリへアクセスする別の命令の直後に続くと、IOP レジスタ・アクセスが正しく実行されません。

**対策:**

NOP などの別の命令を両命令の間に挿入して分離してください。

**適用レビジョン:**

0.0



**11. 09000022—外部データ・アクセスの実効レイテンシが1サイクルではなく2サイクルになる:****説明:**

次のレジスタのビットが外部データ・アクセスを含む命令に影響を与える場合、これらのレジスタの実効レイテンシ1(これらのレジスタに対する書き込みが有効になるために必要なサイクル数)が、代わりに実効レイテンシ2になります:

**MODE1、MODE2、MMASK、ASTATx、ASTATy、STKYx、STKYy**

例えば、次の命令シーケンスを考えます:

```
bit set MODE1 BR8;
nop;                //Sufficient if not immediately followed by external memory access instruction
nop;                //Extra NOP needed if following instruction accesses external memory
pm(i8,m12)=f9;     //i8 is pointing to an address in external memory
```

上記以外のレジスタはこのアノーマリの影響を受けません。

このアノーマリは、命令自体が内部メモリまたは外部メモリのいずれにあるかに無関係であることに注意してください。このアノーマリは、レジスタを変更した直後の2つの命令内で外部メモリ・データ・アクセスがある場合に発生します。

**対策:**

実効レイテンシ1の上記レジスタを更新する場合、レジスタ更新直後の2つの命令内で外部メモリに対するアクセス(PMバスまたはDMバスを使用)を実行しないことが推奨されます。このようなレジスタ更新の後に2つのNOPを挿入することが推奨されます。

**適用レビジョン:**

0.0

**12. 09000023—内部メモリからLCNTR、CURLCNTR、LADDRへ書き込みを行うと、DMAブロックの競合が発生したときに失敗する:****説明:**

内部メモリからLCNTR、CURLCNTR、LADDRに対する書き込み(DMアクセスまたはPMアクセス)を行うと、同じブロックに対するDMA転送が同じサイクルで発生したときに失敗します。

例えば、次の命令を考えます:

```
CURLCNTR = dm (i0,m0);
```

ここで、アドレス(I0+M0)で指定される同じメモリ・ブロックが関係するDMAアクセスについて考えます。DMAと上記書き込みが、上記命令と同じサイクルでDMA転送が発生する場合、上記書き込みが失敗します。

(I0+M0)がメモリ・マップドI/Oレジスタを指定する場合にも、このアノーマリが発生することに注意してください。

**対策:**

1) 別の内部メモリ・ブロックをソース/ターゲットとするようにDMAを変更して、DMAブロックの競合を回避してください。

2) メモリからこれらのレジスタへ直接ロードする代わりに、次に示す2ステップ・プロセスのようにこれらのレジスタを間接的にロードすることができます。

```
r0 = dm (i0,m0);
CURLCNTR = r0;
```

**適用レビジョン:**

0.0



**13. 09000024—特定のコア・ストールが期待通りに実行されない:****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、これにより正しくないコード動作が発生します。

PCSTK をロードした直後に RTI/RTS 命令を実行すると、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] PCSTK = DM(I0,M0);  
[2] RTS;
```

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 RTI/RTS は不定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。

**適用レビジョン:**

0.0