参考資料

Application Note

Jacinto™ 7 および Sitara™ プロセッサにおける DDR インライン ECC エラー アドレス デコード

Linjun Meng, Zekun Bai, Tommy Song *Central FAE*

概要

メモリ エラーの検出および報告によってシステムの信頼性を向上させることを目的として、テキサス インスツルメンツの Jacinto™ 7 および Sitara™ プロセッサでは DDR インライン ECC がサポートされています。ECC イベントが発生すると、DDR コントローラが ECC エラー ログ レジスタに情報を記録します。しかし、ECC パリティ データは DDR メモリ内に格納され、DDR コントローラによって透過的に管理されるため、ログに記録された値は CPU で認識可能なメモリ アドレスに直接対応しません。このアプリケーション ノートでは、ECC エラー ログ レジスタを解釈する方法、DDR インライン ECC メモリ構成を理解する方法、および ERR_ADR および ERR_MSK 値をシステムおよび物理 DDR アドレスに変換する方法について説明します。ECC のデバッグと故障分析を簡素化するため、実用的なデコードの例と付属のデコード ツールも用意されています。

目次

図の一覧

表の一覧

商標

1 概要

DDR インライン ECC は、メモリ エラーの検出と訂正を行い、システムの堅牢性を向上させるため、車載、産業、機能安全アプリケーションで一般的に使用されます。ECC イベントが発生すると、DDR コントローラは影響を受けるメモリ領域に関する情報を ECC エラー ログ レジスタに記録します。

これらのレジスタは有用なデバッグ情報を提供しますが、ログに記録された値をメモリ アドレスとして直接解釈することはできません。DDR インライン ECC により、DDR メモリに追加の ECC ストレージ領域が導入されますが、メモリ構成はデバイス ファミリによって異なります。その結果、実際に障害のあるメモリ位置を特定する前に、さらにデコードが必要になります。

このドキュメントでは、以下について説明します。

- デバッグ時に ECC エラー ログ レジスタが使用される仕組み
- ERR_ADR がメモリ アドレスを直接表さない理由
- DDR インライン ECC がメモリ レイアウトに与える影響
- ERR_ADR および ERR_MSK 値をデコードする方法
- 対応するシステムおよび物理 DDR アドレスの判別方法
- 付属のデコード ツールの使用方法

2 詳細説明

2.1 ECC エラー ログの概要

DDR コントローラが ECC イベントを検出すると、コントローラは一連の ECC エラー ログ レジスタに情報を記録します。ECC 分析時は、表 2-1 に示されているレジスタが通常使用されます。

表 2-1. デバッグ時に使用される ECC エラー レジスタ

登録	使用
ECC_1B_ERR_CNT	検出されたシングル ビット ECC エラーの数を数えます
ECC_1B_ERR_THRSH	シングル ビット ECC イベントの割り込みスレッショルドを定義します
ECC_1B_ERR_ADR_LOG	シングル ビット エラーに関連付けられた ECC 保護ブロックを識別します
ECC_1B_ERR_MSK_LOG	影響を受ける 8 バイト セグメントを識別します
ECC_2B_ERR_ADR_LOG	ダブル ビット エラーに関連付けられた ECC 保護ブロックを識別します
ECC_2B_ERR_MSK_LOG	影響を受ける 8 バイト セグメントを識別します

詳細なレジスタ定義は該当するテクニカル リファレンス マニュアルに記載されているため、このドキュメントでは繰り返していません。このアプリケーション ノートでは、報告された値がどのようにデコードされるかに焦点を当てます。

2.1.1 ERR_ADR および ERR_MSK の理解

ERR_ADR は、障害のあるメモリ アドレスを直接表すものではありません。代わりに、以下を行います。

- ERR_ADR は ECC 保護ブロックを識別します。
- ERR_MSK は、そのブロック内の影響を受ける 8 バイト セグメントを識別します。
- 両方の値を合わせて解釈する必要があります。
- 最終的なメモリ アドレスを決定するには、追加のデコードが必要です。

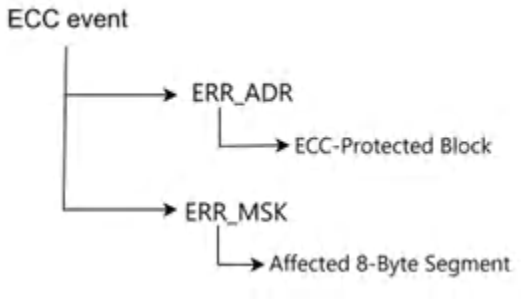

図 2-1. ERR_ADR と ERR_MSK の関係

2.2 アドレス デコードが必要な理由

DDR インライン ECC が有効になると、ECC パリティ情報とユーザー データが DDR メモリ内に保存されます。ただし、ソフトウェアでは ECC ストレージ領域が認識されず、DDR コントローラによって内部的に管理されます。その結果、CPU で認識可能なシステム アドレスは物理 DDR アドレスとは異なります。

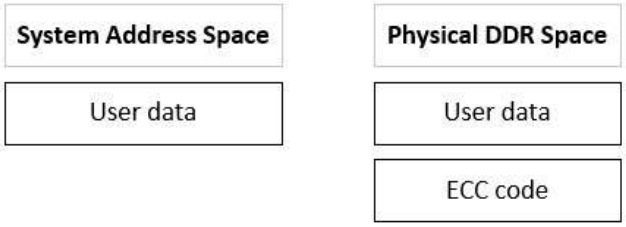

図 2-2. システム アドレス空間と物理 DDR 空間の間の関係

その結果、以下のようになります。

- ERR_ADR はダイレクト メモリ アドレスではありません。
- 物理 DDR アドレスには ECC パリティ ストレージが含まれます。
- アドレス変換は、DDR インライン ECC メモリ レイアウトを考慮する必要があります。

2.3 DDR インライン ECC メモリ レイアウト

DDR インライン ECC メモリ構成は、アドレス デコード プロセスの基盤となります。ECC レポートの粒度はプロセッサ ファミリによって異なりますが、サポートされているすべてのデバイスは同じ原理を使用します。

- ユーザー データ
- ECC 保護データ

デバイス ファミリ間の主な違いは、各 ECC セグメントで保護されているユーザー データのサイズです。

2.3.1 TDA4VM/TDA4VE/TDA4VH

特性:

- DDR 幅:32 ビット
- VBUSM 幅:512 ビット
- ERR_ADR の粒度:64 バイト

メモリ レイアウト:

表 2-2. TDA4VM/TDA4VE/TDA4VH のメモリ レイアウト

システム オフセット	DDR コンテンツ	DDR 物理オフセット
0x000	512B ユーザー データ 0	0x000～0x1FF
認識不可	データ ブロック 0 の ECC	0x200～0x23F
認識不可	データ ブロック 1 の ECC	0x240～0x27F
0x200	512B ユーザー データ 1	0x280～0x47F

したがって、次のようになります。

```
512B User Data0+ 64B ECC code0 + 64B ECC Code1 + 512B User Data1 = 1152B (0x480)
```

システムで認識可能なメモリの各 0x400 バイトは、物理 DDR 空間で 0x480 バイトを占有します。

2.3.2 DRA821/DRA829/TDA4VEN/AM62A/AM62P

特性:

- DDR 幅:32 ビット
- VBUSM 幅:256 ビット
- ERR_ADR の粒度:32 バイト

メモリ レイアウト:

表 2-3. DRA821/DRA829/TDA4VEN/AM62A/AM62P のメモリ レイアウト

システム オフセット	DDR コンテンツ	DDR 物理オフセット
0x00	256B ユーザー データ 0	0x000～0xFF
認識不可	データ ブロック 0 の ECC	0x100～0x11F
認識不可	データ ブロック 1 の ECC	0x120～0x13F
0x100	256B ユーザー データ 1	0x140～0x23F

したがって、次のようになります。

```
256B User Data0+ 32B ECC Code0 + 32B ECC Code1 + 256B User Data1 = 576B (0x240)
```

システムで認識可能なメモリの各 0x200 バイトは、物理 DDR 空間で 0x240 バイトを占有します。

2.3.3 AM62/AM62L/AM64

セクション 2.3.2 に示されているように、AM62、AM62L、AM64 は、デバイスと同じ物理 DDR インライン ECC メモリ構成を使用します。ただし、ECC エラー報告の粒度は異なります。これらのデバイスは 16 ビット DDR インターフェイスを使用しているためです。その結果、以下のようになります。

- ERR_ADR の粒度 = 16 バイト
- ERR_MSK 幅 = 4 ビット

物理的な DDR レイアウトは変更されません。

2.4 アドレス デコード方法

ERR_ADR レジスタには、直接システム アドレスが含まれていません。代わりに、レジスタは ECC 保護されたメモリ ブロックのインデックスを記録します。アドレスの粒度は、次のプロセッサ ファミリによって異なります。

- AM62/AM62L/AM64：16 バイト
- AM62A/AM62P/DRA821/DRA829/TDA4VEN：32 バイト
- TDA4VM/TDA4VE/TDA4VH：64 バイト

ECC ブロック開始アドレスは ERR_ADR から派生され、ERR_MSK はレポート範囲内で影響を受ける 8 バイト セグメントを識別します。

影響を受けるメモリの場所を特定するには、次の手順を実行する必要があります。

1. ERR_ADR で表される ECC ブロックを決定します。
2. ERR_MSK を使用して、影響を受ける 8 バイト セグメントを識別します。
3. 結果のオフセットをシステム アドレスに変換します。
4. ECC メモリのレイアウトを適用して、物理 DDR アドレスを決定します。

2.5 AM62A デコード例

仮定:

ERR_ADR = 0x101

ERR_MSK = 0x10

AM62A で使用される要素:

- 32 バイトの ERR_ADR 粒度
- 8 ビットの ERR_MSK
- 256 ビットの VBUSM 幅

ECC ブロック開始オフセット:

0x101 × 0x20 = 0x2020

ERR_ADR は奇数であるため、対応する ECC ブロックは同じ ERR_MSK フィールドを、隣接する偶数ブロックと共有します。したがって、ERR_ADR のパリティに従ってオフセットを調整する必要があります。

ERR_MSK のビット 4 がアサートされます。

システム アドレス:

= 0x2020 - 0x20 + (4 × 8) + 0x80000000

= 0x80002020

16 の ERR_ADR エントリごとに、1 つの 0x240 バイトの物理 DDR セグメント (ECC ストレージを含む) に対応します。したがって、ブロック数は ERR_ADR/16 から取得されます。

物理アドレス:

ブロック数 = 0x101/0x10 = 0x10

その配置されたブロックの場合:オフセット = 0x101 % 0x10 = 1

DDR での位置:0x10 × 0x240 + 0x20 + 0x80000000 = 0x80002420

2.6 TDA4VE デュアル DDR インターリーブの例

TDA4VE 構成:

TDA4VE は、インターリーブ モードで動作する 2 つの DDR コントローラを使用しています。各 DDR コントローラは、独立した ECC エラー ログ レジスタセットを維持します。したがって、ECC イベントを報告する DDR コントローラによって、障害のある場所を含むチャネルが決定されます。

- デュアル DDR コントローラ
- 32 ビット DDR インターフェイス幅
- 512 ビットの VBUSM 幅
- ERR_ADR の粒度 = 64 バイト
- ERR_MSK 幅 = 16 ビット
- デフォルトの DDR インターリーブ粒度 = 128 バイト

デュアル DDR インターリーブ モードでは、連続する 128 バイトのメモリ領域が DDR0 および DDR1 に交互に割り当てられます。TDA4VE デュアル DDR インターリーブ モードの場合、16 の ERR_ADR エントリごとに 1 つの反復アドレスパターンを形成します。表 2-4 に、以下に示す ERR_ADR mod 16 に対応する論理および物理オフセットを示します。

表 2-4. 論理オフセットおよび物理オフセットの ADR マッピング テーブル

MOD(ADR,16)	DDR0 論理オフセット	DDR1 論理オフセット	DDR0 物理オフセット	DDR1 物理オフセット
0	0x0	0x80	0x0	0x80
1	0x40	0xC0	0x40	0xC0
2	0x100	0x180	0x100	0x180
3	0x140	0x1C0	0x140	0x1C0
4	0x200	0x280	0x200	0x280
5	0x240	0x2C0	0x240	0x2C0
6	0x300	0x380	0x300	0x380
7	0x340	0x3C0	0x340	0x3C0
ハミング コード	認識不可	認識不可	0x400	0x480
ハミング コード	認識不可	認識不可	0x440	0x4C0
8	0x400	0x480	0x500	0x580
9	0x440	0x4C0	0x540	0x5C0
10	0x500	0x580	0x600	0x680
11	0x540	0x5C0	0x640	0x6C0
12	0x600	0x680	0x700	0x780
13	0x640	0x6C0	0x740	0x7C0
14	0x700	0x780	0x800	0x880
15	0x740	0x7C0	0x840	0x8C0

仮定:

DDR0_1B_ERR_ADR = 0x49

DDR0_1B_ERR_MSK = 0x0400

システム アドレスをデコード

アドレスの粒度 = 64 バイト

システムは、128 バイトのインターリーブ粒度を持つデュアル DDR インターリーブ モードに構成されます。

16 の ERR_ADR エントリごと:

占有される論理アドレス空間は以下のとおりです。

16 × 128B = 0x800B

占有される物理 DDR アドレス空間は以下のとおりです。

16 × 128B + ECC ストレージ

= 0x900B

ECC ブロック開始オフセットは以下のとおりです。

floor (0x49/16) × 0x800

0x4 × 0x800

= 0x2000

ECC ブロック セグメントのオフセットは以下のとおりです。

0x49 mod 16 = 9

アドレス マッピング表から、ERR_ADR mod 16 = 9 に対応する DDR0 論理オフセットは 0x440 です。

ERR_MSK のビット 10 が以下に設定されます。

エラー オフセット = 10×8 = 0x50

ERR_ADR は奇数であるため、対応する ECC ブロックは同じ ERR_MSK フィールドを前のアドレス ブロックと共有します。したがって、次のようになります。

システム オフセット

= 0x440 - 0x40 + 0x50 + 0x2000

= 0x2450

システム アドレス

= システムオフセット + 0x80000000

= 0x80002450

物理アドレスの計算:

ECC ブロックの開始オフセットは以下のとおりです。

floor (0x49/16) × 0x900

= 0x4 × 0x900

= 0x2400

ECC ブロック セグメントのオフセットは以下のとおりです。

0x49 mod 16 = 9

アドレス マッピング表から、ERR_ADR mod 16 = 9 に対応する DDR0 物理オフセットは 0x540 です。

ERR_MSK のビット 10 が以下に設定されます。

エラー オフセット = 10×8 = 0x50

ERR_ADR は奇数であるため、対応する ECC ブロックは同じ ERR_MSK フィールドを前のアドレス ブロックと共有します。したがって、次のようになります。

物理オフセット

= 0x540 - 0x40 + 0x50 + 0x2400

= 0x2950

物理アドレス

= 物理オフセット + 0x80000000

= 0x80002950

2.7 デデコード ツールの使用

スプレッドシート ツール TI TDA4x インライン ECC エラー レジスタ トランスレータは、ERR_ADR および ERR_MSK 値を、対応するシステムおよび物理 DDR アドレスに変換することで、アドレス デコード プロセスを自動化します。このツールでは、ECC エラー レジスタに関連する影響を受ける 8 バイト領域も強調表示されます。

3 まとめ

DDR インライン ECC は、Jacinto 7 および Sitara プロセッサのメモリ破損イベントを検出する効果的なメカニズムを提供します。ECC エラー ログ レジスタは、影響を受ける ECC 保護メモリ ブロックに関する情報を記録しますが、ログに記録された値はシステム メモリ アドレスに直接対応しません。ECC メモリ構成を理解し、このドキュメントに記載されているデコード方法を適用することで、ユーザーは ECC イベントに関連するメモリの場所を正確に特定できます。付属のデデコード ツールを使用することで、デバッグと故障分析をさらに簡素化できます。

4 参考資料

1. テキサス インスツルメンツ、『J721E DRA829/TDA4VM プロセッサ テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
2. テキサス インスツルメンツ、『J721S2/TDA4VE/AM68A テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
3. テキサス インスツルメンツ、『J784S4/TDA4VH テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
4. テキサス インスツルメンツ、『J7200 DRA821 テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
5. テキサス インスツルメンツ、『AM62Ax Sitara™ プロセッサ テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
6. テキサス インスツルメンツ、『AM62Px Sitara™ プロセッサ テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
7. テキサス インスツルメンツ、『AM62x Sitara™ プロセッサ テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。
8. テキサス インスツルメンツ、『AM64x/AM243x テクニカル リファレンス マニュアル』、テクニカル リファレンス マニュアル。

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、 TI の販売条件 、 TI の総合的な品質ガイドライン 、 ti.com または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。 TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

最終更新日：2025 年 10 月