

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

SIII51015-1.3

概要

航空電子、テレコム、システム・コントロール、および軍用アプリケーションの分野で使用されるクリティカルな用途では、以下ができることが重要です。

- Stratix III デバイ스에 格納されたコンフィギュレーション・データが正確であるかを確認する。
- システムにコンフィギュレーション・エラーの発生を警告する。

Stratix® III ファミリーでは、エラー検出機能が強化されています。Stratix III デバイスでは、Stratix II デバイスよりも短時間で SEU (Single Event Upset) のエラー検出および回復を実行することができます。



SEU に関する情報は、アルテラ・ウェブサイト (www.altera.co.jp) の製品ページにあります。

Stratix III デバイスに組み込まれた専用回路は、オプションで継続的かつ自動的に SEU をチェック可能な CRC (Cyclic Redundancy Check) エラー検出機能で構成されています。

この項では、Stratix III デバイスがユーザー・モードのときにエラー検出 CRC 機能をアクティブにして使用する方、および CRC エラーにより発生したコンフィギュレーション・エラーから回復する方法について説明します。



Stratix III デバイスの場合、Quartus® II ソフトウェア v6.1 からエラー検出 CRC 機能が提供されています。



Stratix III デバイスは、1.1 V の VCCL でエラー検出 CRC 機能のみをサポートしています。この機能は、Stratix III が 0.9 V の VCCL で動作している場合にはサポートされません。

Stratix III ファミリーの CRC エラー検出機能を使用しても、フィッティングや性能には影響を与えません。




CRC について詳しくは、「AN 357: アルテラ FPGA デバイスの CRC によるエラー検出」を参照してください。

エラー検出の基礎

エラー検出では、媒体から受信したデータが送信中に破壊されたかどうかを判定します。これを実行するために、トランスミッタはデータのチェックサム値を計算し、オリジナルのデータ・フレームにチェックサムを付加するファンクションを使用します。レシーバは同じ計算方法を使用して、受信データ・フレームのチェックサムを生成し、これを送信されたチェックサムと比較します。この2つのチェックサム値が等しい場合、受信データ・フレームは正しく、送信中またはストレージ中にデータ破壊は発生していません。

エラー検出 CRC 機能も、同じ概念を使用しています。Stratix III デバイスが正常にコンフィギュレーションされ、ユーザー・モードになっている場合は、エラー検出 CRC 機能により、コンフィギュレーション・データの完全性が確保されます。

 以下の2つの CRC エラー・チェックがあります。1つはコンフィギュレーション時に常に行われるチェック、もう1つはユーザー・モードにおいてバックグラウンドで実行されるオプションの CRC エラー・チェックです。この2つの CRC エラー・チェックは、同じ CRC 多項式を使用しますが、エラー検出の実装方法が異なります。



詳細は、15-2 ページの「[コンフィギュレーション・エラー検出](#)」および15-3 ページの「[ユーザー・モード・エラー検出](#)」を参照してください。

コンフィギュレーション・エラー検出

コンフィギュレーション・モードでは、フレーム・ベースの CRC がコンフィギュレーション・データ内に格納され、各データ・フレームの CRC 値を保持しています。

コンフィギュレーション中には、Stratix III デバイスが受信したデータ・フレームに基づいて CRC 値を計算し、それをデータ・ストリームのフレーム CRC 値と比較します。コンフィギュレーションは、デバイスがエラーを検出するか、コンフィギュレーションが完了するまで継続します。

Stratix III デバイスでは、CRC 値はコンフィギュレーション・ステージ中に計算されます。並列 CRC エンジンには、フレームごとに 16 の CRC チェック・ビットを生成し、それらをコンフィギュレーション・ランダム・アクセス・メモリ (CRAM) に格納します。CRC チェック・ビットの格納に使用される CRAM チェインの幅は 16 ビットで、長さはデバイスのフレームの数値と同じです。

ユーザー・ モード・ エラー検出

Stratix III デバイスは、CRAM セル内のソフト・エラーによるデータ破壊を検出するエラー検出回路を内蔵しています。この機能により、CRAM のすべての内容を読み出して検証し、コンフィギュレーション計算済みの CRC 値に一致させることができます。ソフト・エラーとは、イオン化した粒子によって生じる CRAM ビット状態の変化のことです。

エラー検出機能は、コンフィギュレーション済みの CRAM ビットの CRC を継続的に計算し、それを計算済みの CRC と比較します。両方の CRC が一致する場合、現在のコンフィギュレーション CRAM ビットにはエラーはありません。このエラー検出プロセスは、デバイスがリセットされる (nCONFIG を Low に設定) まで継続します。

CRC エラー検出オプションがイネーブルされている場合、デバイスがユーザー・モードになるとすぐにエラー検出プロセスをイネーブルすることができます。内部 100 MHz コンフィギュレーション・オシレータは、2 ~ 256 (2 のべき乗) で分周され、エラー検出プロセス時にクロック・ソースとして使用されます。Quartus II ソフトウェアのオプション設定で、クロック分周係数を設定します。

シングル 16 ビット CRC 計算は、フレームごとに実行されます。フレームの CRC 計算が終了すると、エラー検出回路で CRAM ビット・エラーが検出されず、出力信号 CRC_ERROR が 0 の場合、16 ビット・シグネチャは hex 0000 になります。回路がデバイスのフレーム内で CRAM ビット・エラーを検出した場合、シグネチャはゼロ以外になります。これにより、CRC エンジンエラー・ビット位置の検索を開始します。

Stratix III デバイスのエラー検出機能は、各フレームの CRC チェック・ビットを計算し、チップ内でビット・エラーを検出すると、CRC_ERROR ピンを High にプルアップします。フレーム内では、すべてのシングル・ビット、ダブル・ビット、およびスリー・ビット・エラーを検出することができます。SEU イベントによって CRAM ビットが 4 ビット以上反転する確率は、ほとんどありません。一般に、すべてのエラー・パターンが検出される確率は 99.998 % です。

CRC エンジンは、ビット位置をレポートし、すべてのシングル・ビット・エラーおよび 99.641 % 以上のダブル隣接エラーのタイプを判別します。その他のエラー・パターンが発生する確率は非常に低いため、CRC エンジンによるビットの反転位置に関するレポートは保証されません。

JTAG (Joint Test Action Group) およびコア・インタフェースを介して、エラー・ビット位置を読み出すこともできます。CRC が別のフレームで次のエラーを検出する前に、これらのビットを JTAG 命令、SHIFT_EDERROR_REG、またはコア・インタフェースを介してシフト・アウトする必要があります。次のフレームにもエラーがある場合は、1 フレーム CRC 検証の時

間内にこれらのビットをシフト・アウトする必要があります。エラー検出クロック周波数を低くして、この時間間隔を延長することを選択できますが、これによって SEU イベントのエラー回復時間が長くなります。Stratix III デバイスの最小アップデート間隔については、表 15-7 を参照してください。次のエラー位置が検出される前にこれらのビットがシフト・アウトされない場合、前のエラー位置とエラー・メッセージが新しい情報によって上書きされます。CRC 回路が動作を継続し、エラーが検出された場合は、リコンフィギュレーションを完了するか、または CRC エラーを無視するかを判断します。

エラー検出ロジックは、現在のフレームでエラーが発生したかどうかに関係なく、データの次のフレームに対する CRC_ERROR と 16 ビット・シグネチャの計算を継続します。これらの信号をモニタし、ソフト・エラーが発生した場合は、適切な処置を行う必要があります。

Stratix III デバイスのエラー検出回路は、16 ビット CRC-ANSI 規格（16 ビット多項式）を CRC ジェネレータとして使用します。

各フレームの計算された 16 ビット CRC シグネチャは、コア内のレジスタに格納されます。ストレージ・レジスタの合計サイズは、16（1 フレームあたりのビット数）× フレーム数になります。

Stratix III デバイスのエラー検出機能は、メモリ・ブロックと I/O バッファはチェックしません。メモリ・ブロックは、内容にエラーがないかチェックするのに使用されるパリティ・ビットをサポートしています。I/O バッファのビットは、ストレージ・エレメントとして CRAM セルよりもソフトウェア・エラーに強いフリップ・フロップを使用しているため、エラー検出中はチェックされません。

M144K TriMatrix メモリ・ブロックは、ブロック内のエラーをチェックし訂正するエラー訂正コード・ブロックを内蔵しています。しかし、MLAB メモリ・ブロックとして使用されるロジック・アレイ・ブロック（LAB）では、エラー検出検証中は無視されます。これにより、CRC_ERROR 信号が以前にチェックした CRAM フレームのエラー・ステータスに応じて、High または Low になったままの場合があります。



Stratix III TriMatrix メモリ・ブロックにおけるエラー検出について詳しくは、「Stratix III デバイス・ハンドブック Volume 1」の「Stratix III デバイスの TriMatrix エンベデッド・メモリ・ブロック」の章を参照してください。

エラー検出ブロックのテスト機能を提供するために、JTAG 命令 EDERROR_INJECT が用意されています。この命令は、Stratix III デバイスのエラー・インジェクションに使用される 21 ビット JTAG フォールト・インジェクション・レジスタの内容を変更することができ、エラー検出ブロックのテストを実施できます。


 デバイスがユーザー・モードのときは、EDERROR_INJECT JTAG 命令しか実行できません。


表 15-1 に、EDERROR_INJECT JTAG 命令の説明を示します。

JTAG 命令	命令コード	説明
EDERROR_INJECT	00 0001 0101	この命令は、エラー・インジェクションに使用される 21 ビット JTAG フォールト・インジェクション・レジスタを制御します。

Jam™ ファイル (.jam) を作成して、テストと検証プロセスを自動化することができます。これにより、デバイスをリコンフィギュレーションすることなく、CRC 機能をイン・システムでオン・ザ・フライに検証できます。CRC 回路に切り換えて、SEU で誘発された実際のエラーをチェックできます。

シングル・エラー、ダブル・エラー、または相互に隣接するダブル・エラーをコンフィギュレーション・メモリに導入することができます。これにより、デザインの検証およびシステムの障害許容特性評価を容易に行うことができます。JTAG フォールト・インジェクション・レジスタで EDERROR_INJECT 命令を使用して、リードバック・ビットを反転します。Stratix III デバイスを、強制的にエラー・テスト・モードにします。

JTAG フォールト・インジェクション・レジスタの内容は、最後と最初のフレームの処理中はフォールト・インジェクション・レジスタにロードされません。処理が終了した段階でのみロードされます。

 最初のデータ・フレームでのみエラー・インジェクションを実行することができますが、いつでもエラー情報をモニタすることが可能です。




JTAG フォールト・インジェクション・レジスタおよびフォールト・インジェクション・レジスタについて詳しくは、15-10 ページの「エラー検出レジスタ」を参照してください。

表 15-2 に、フォールト・インジェクション・レジスタの実装方法を示し、エラー・インジェクションについて説明します。

ビット	ビット [20..19]		ビット [18..8]	ビット [7..0]	
説明	エラーのタイプ		インジェクトされたエラーのバイト位置	エラー・バイト値	
内容	エラーのタイプ (1)		最初のデータ・フレームでインジェクトされたエラーの位置を示します。	ビット・エラーの位置を示し、エラー・インジェクション・タイプ選択に対応します。	
	ビット [20]	ビット [19]			
	0	1			シングル・バイト・エラー・インジェクション
	1	0			ダブル隣接バイト・エラー・インジェクション
	0	0	ノー・エラー・インジェクション		

表 15-2 の注：

- (1) 1は有効な選択肢でないため、ビット [20] とビット [19] は1に設定することはできません。エラー検出回路は、ノー・エラー・インジェクションとしてデコードします。

 アルテラでは、テストの完了後にデバイスをリコンフィギュレーションすることを推奨しています。

SEU の自動検出

Stratix III デバイスは、SEU 検出の自動チェック用のオンチップ回路を備えています。高中性子束環境で、デバイスにエラーのない動作を要求するアプリケーションでは、継続的なデータ完全性を確保するために定期的なチェックが必要です。エラー検出 CRC 機能は、データの信頼性を確保しており、SEU を軽減するために最良の選択肢の1つです。

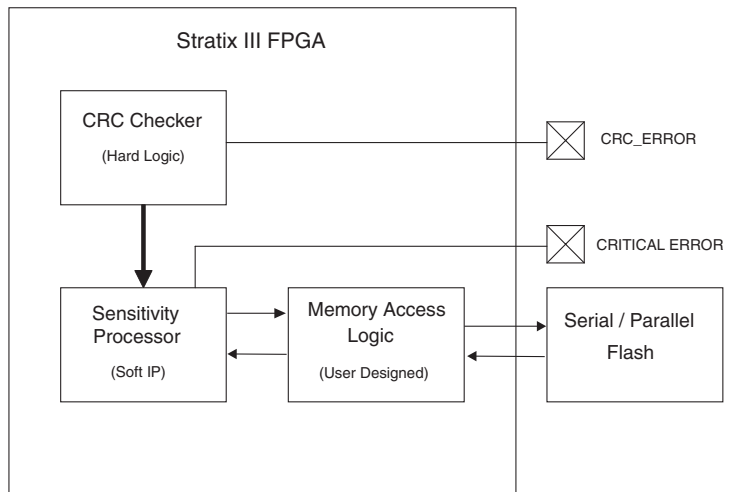
このエラー検出 CRC 機能は、Stratix III デバイス内の既存の回路を用いて実装できるため、外部ロジックは必要ありません。CRC_ERROR ピンは、コンフィギュレーション CRAM データが破壊されているときはソフト・エラーをレポートするため、デバイスをリコンフィギュレーションするか、またはエラーを無視するかを決定しなければなりません。

重大エラー検出

CRC 回路がエラーを検出すると、センシティブリティ・プロセッサはユーザーが設計したロジックを通じてマスクされたコンフィギュレーション・ビットストリームにアクセスして識別したエラーの重要度を判断し、リコンフィギュレーションを実行するようシステムに警告します。重大でないエラーの場合、エラー検出回路は CRC_ERROR および次のデータ・フレームの 16 ビット・シグネチャの計算を継続します。

この機能は、センシティブリティ・プロセッサ・リファレンス・デザインを使用して、トリプル・モジュール冗長デザイン手法を実装し、エラー検出ブロックとコア IP ロジック間の信号をインターフェースします。同じ回路の 3 つのコピーを実装し、出力信号でビット単位の「多数決」を実行します。CRAM ビットの 3 ビットが SEU イベントによって反転する可能性はほとんどありません。図 15-1 に、重大エラー検出実装のブロック図を示します。

図 15-1. 重大エラー検出実装のブロック図



このリファレンス・デザインは、Quartus II ソフトウェアの将来のバージョンでサポートされます。

エラー検出 ピンの説明

選択したエラー検出機能のタイプに応じて、異なるエラー検出ピンを使用してユーザー・モード時のデータをモニタする必要があります。

CRC_ERROR ピン

表 15-3 は、CRC_ERROR ピンを説明しています。

ピン名	ピン・タイプ	説明
CRC_ERROR	I/O、出力またはオープン・ドレイン・出力 (オプション)	エラー検出回路がコンフィギュレーション CRAM ビットでエラーを検出したことを示すアクティブ High 信号です。このピンはオプションで、エラー検出 CRC 回路がイネーブルされるときに使用されます。エラー検出 CRC 回路がディセーブルされると、ユーザー I/O ピンになります。WYSIWYG 機能を使用するとき、CRC エラー出力は CRC_ERROR ピンへの専用パスです。Quartus II のデフォルト設定では、CRC_ERROR ピンは専用出力です。CRC_ERROR が専用出力として使用される場合、ユーザーはピンが存在するバンクの V _{CCIO} が信号を受信するシステムの入力電圧仕様に適合していることを確認する必要があります。オプションで、Quartus II ソフトウェアの Device & Pin Options ダイアログ・ボックスの Error Detection CRC タブをイネーブルにして、ユーザーはこのピンをオープン・ドレイン出力に設定できます。このピンをオープン・ドレインとして使用することで、電圧レベリングという利点を得られます。ユーザーは、このピンを 10 k Ω 抵抗を介して V _{CCPGM} に接続することで、このピンをオープン・ドレインとして使用できます。または、信号を受信するシステムの電圧入力電圧仕様に応じて、ユーザーはプルアップ抵抗を異なるプルアップ電圧に接続できます。



WYSIWYG (What You See Is What You Get) は、Quartus II ソフトウェアの Verilog Quartus Mapping (VQM) ネットリストで最適化を行う最適化手法です。

CRITICAL ERROR ピン

Stratix III デバイスの CRC_ERROR ピン情報は、アルテラ・ウェブサイト (www.altera.co.jp) の資料ページのデバイス・ピンアウトに掲載されています。表 15-4 は、CRITICAL ERROR ピンを説明しています。

ピン名	ピン・タイプ	説明
CRITICAL ERROR	I/O、出力	センシティビティ・プロセッサ・リファレンス・デザインがコンフィギュレーション CRAM ビットでエラーを検出したことを示すアクティブ High 信号です。このピンはオプションで、重大エラー検出がイネーブルされるときに使用されます。

Stratix III デバイスの CRITICAL ERROR ピン情報は、アルテラ・ウェブサイト (www.altera.co.jp) の資料ページのデバイス・ピンアウトの将来のバージョンに掲載されます。


エラー検出 ブロック

Quartus II ソフトウェアでは、Stratix III デバイスのエラー検出ブロックをイネーブルできます (15-14 ページの「ソフトウェア・サポート」を参照)。このブロックには、デバイス内のコンフィギュレーション CRAM ビット用の 16 ビット CRC シグネチャの計算に必要なロジックが含まれています。

この CRC 回路はエラーが発生した場合でも継続して動作します。ソフト・エラーが発生すると、デバイスは CRC_ERROR ピンを High に設定します。以下の 2 つのタイプの CRC 検出により、コンフィギュレーション・ビットをチェックします。

- 1 つは、CRC_ERROR ピンで使用するユーザー・モード中の CRAM エラー・チェック機能 (16 ビット CRC) です。
 - データの各フレームでは、フレーム・データの最後に計算済み 16 ビット CRC が CRC 回路に入り、エラーがあるかどうか判断します。
 - エラーがある場合、検索エンジンがエラーの位置の検索を開始します。
 - エラー検出ブロックが動作し続けている間、JTAG 命令またはコア・インタフェース・ロジックによりエラー・メッセージをシフト・アウトできます。

- JTAGインタフェースは、最初のフレームの16ビットCRC結果を読み出し、16ビットCRCビットをテスト用に16ビットストレージ・レジスタにシフトします。
 - シングル・エラー、ダブル・エラー、または相互に隣接するダブル・エラーを、テストおよびデザイン検証のためにコンフィギュレーション・メモリで意図的に発生させることができます。
- もう1つは、各コンフィギュレーション・データ・フレームに組み込まれている16ビットCRCです。
- コンフィギュレーション中、データのフレームがStratix III デバイスにロードされた後、計算済みCRCがCRC回路にシフト・インされます。
 - 同時に、シフト・インされたデータ・フレームのCRC値が計算されます。計算済みCRCと計算されたCRC値が一致しない場合、nSTATUSがLowに設定されます。どのデータ・フレームにも16ビットCRCがあるため、コンフィギュレーション・ビット・ストリーム全体では多数の16ビットCRC値があります。コンフィギュレーション・データ・フレームの長さはデバイスごとに異なります。

 「エラー検出ブロック」の項では、最初のタイプ、すなわちデバイスがユーザー・モード時の16ビットCRCのみ扱います。

エラー検出レジスタ

エラー検出回路には、計算されたCRCシグネチャを格納する1組の16ビット・レジスタがあります。シンドローム・レジスタにゼロ以外の値がある場合、CRC_ERRORピンはHighに設定されます。図15-2に、エラー検出回路、シンドローム・レジスタ、およびエラー・インジェクション・ブロックのブロック図を示します。

図 15-2. エラー検出ブロック図

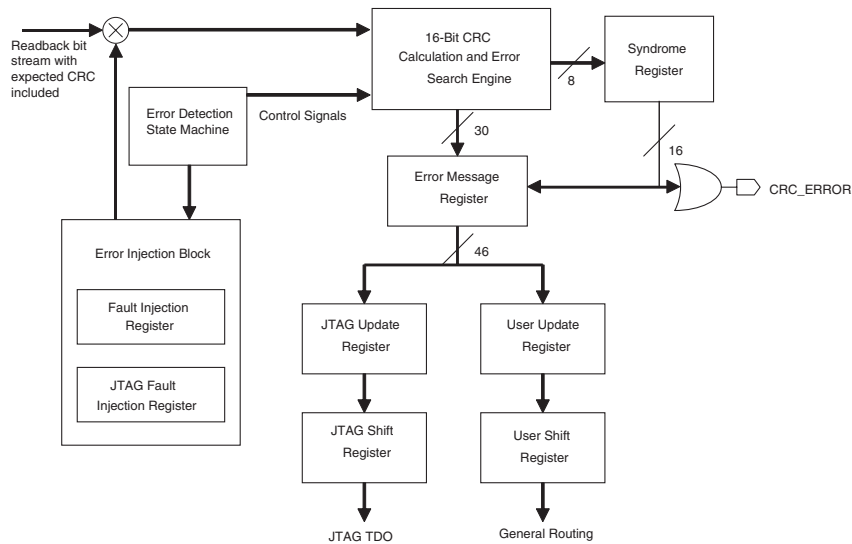


表 15-5 は、図 15-2 に示すレジスタを定義しています。

レジスタ	説明
シンドローム・レジスタ	このレジスタには、エラー検出検証サイクルでの現在のフレームのCRCシグネチャが含まれています。CRC_ERROR 信号は、このレジスタの内容から派生したものです。
エラー・メッセージ・レジスタ	この 46 ビット・レジスタには、エラー・タイプ、エラーの位置、および実際の症状に関する情報が含まれています。レポートされるエラー・タイプおよびエラーの位置は、シングルおよびダブル隣接ビット・エラーです。他のタイプのエラーに対する位置ビットは、エラー・メッセージ・レジスタでは識別されません。レジスタの内容は、JTAG SHIFT_EDERROR_REG 命令またはコア・インタフェースを通してコアにシフト・アウトできます。

表 15-5. エラー検出レジスタ (2 / 2)	
レジスタ	説明
JTAG アップデート・レジスタ	このレジスタは、46 ビット・レジスタの内容が検証されてから 1 サイクル後にエラー・メッセージ・レジスタの内容で自動的にアップデートされます。このレジスタには、JTAG シフト・レジスタにサンプリングされる前にアサートする必要があるクロック・イネーブルが含まれています。この要求条件により、JTAG アップデート・レジスタへのエラー・メッセージ・レジスタの内容の書き込みと、JTAG シフト・レジスタによるその内容の読み出しが同時に発生しないようにしています。
ユーザー・アップデート・レジスタ	このレジスタは、46 ビット・レジスタの内容が検証されてから 1 サイクル後にエラー・メッセージ・レジスタの内容により自動的にアップデートされます。このレジスタには、ユーザー・シフト・レジスタにサンプリングされる前にアサートする必要があるクロック・イネーブルが含まれています。この要求条件により、ユーザー・アップデート・レジスタへのエラー・メッセージ・レジスタの内容の書き込みと、ユーザー・シフト・レジスタによるその内容の読み出しが同時に発生しないようにしています。
JTAG シフト・レジスタ	このレジスタは、JTAG インタフェースからアクセスでき、JTAG 命令 SHIFT_EDERROR_REG により JTAG アップデート・レジスタの内容をサンプリングし、読み出すことが可能です。
ユーザー・シフト・レジスタ	このレジスタは、コア・ロジックからアクセスでき、ユーザー・ロジックによりユーザー・アップデート・レジスタの内容をサンプリングし、読み出すことが可能です。
JTAG フォールト・インジェクション・レジスタ	この 21 ビット・レジスタは、JTAG 命令 EDERROR_INJECT によって完全に制御されます。このレジスタは、ビットストリームに必要なエラー・インジェクションの情報を保持します。
フォールト・インジェクション・レジスタ	フォールト・インジェクション・レジスタがアップデートされると、JTAG フォールト・インジェクション・レジスタの内容がこの 21 ビット・レジスタにロードされます。

エラー検出の タイミング

Quartus II ソフトウェアを通して CRC 機能がイネーブルされると、デバイスはコンフィギュレーションおよび初期化の完了後、ユーザー・モードに入ったときに自動的に CRC プロセスをアクティブにします。

フレーム内でエラーが検出された場合、**Error Message Register** がアップデートされた後、CRC_ERROR はエラー位置検索の最後に High にドライブされます。このサイクルの最後で、CRC_ERROR ピンは最小 32 クロック・サイクル間 Low にプルダウンされます。次のフレームにもエラーが含まれる場合、**Error Message Register** が新しい値を上書きされた後、CRC_ERROR は再び High にドライブされます。ユーザーは、CRC_ERROR ピンの各立ち上がりエッジにおけるエラー・メッセージのアンロードを開始できます。エラー検出はデバイスがリセットされるまで動作します。

エラー検出回路は、最大周波数およびその分周周波数で内部コンフィギュレーション・オシレータを動作させます。表 15-6 に、最小および最大エラー検出周波数を示します。

デバイス・タイプ	エラー検出周波数	最大エラー検出周波数	最小エラー検出周波数	有効な除数 (n)
Stratix III	100 MHz/2 ⁿ	50 MHz	390 kHz	1, 2, 3, 4, 5, 6, 7, 8

Quartus II ソフトウェアで分周係数を指定することによって、低いクロック周波数を設定することができます (15-14 ページの「ソフトウェア・サポート」を参照)。除数は 2 のべき乗です。ここで n は 1 ~ 8 です。除数の範囲は 2 ~ 256 です。以下の式を参照してください。

$$\text{Error detection frequency} = \frac{100\text{MHz}}{2^n}$$

エラー・メッセージ・レジスタの情報を失わないように、エラー・メッセージをモニタする必要があります。エラー・メッセージ・レジスタは、エラーが発生するたびにアップデートされます。エラー・メッセージ・レジスタの各アップデートの最小時間間隔は、デバイスおよびエラー検出クロック周波数によって異なります。表 15-7 に、Stratix III デバイスにおけるエラー・メッセージ・レジスタの各アップデートの推定最小時間間隔を示します。

デバイス	タイミング間隔 (μs)
EP3SL50	11
EP3SL70	11

表 15-7. エラー・メッセージ・レジスタの最小アップデート間隔
注(1) (2 / 2)

デバイス	タイミング間隔 (μs)
EP3SL110	16
EP3SL150	16
EP3SL200	21
EP3SE260	21
EP3SL340	23
EP3SE50	11
EP3SE80	16
EP3SE110	16

表 15-7 の注:

(1) これらのタイミング数値は暫定仕様です。

ソフトウェア・サポート

Quartus II ソフトウェアは、バージョン 6.1 以降で Stratix III のデバイスエラー検出 CRC 機能をサポートしています。この機能をイネーブルすると、オプションの兼用 CRC_ERROR ピンに CRC_ERROR 出力を生成します。

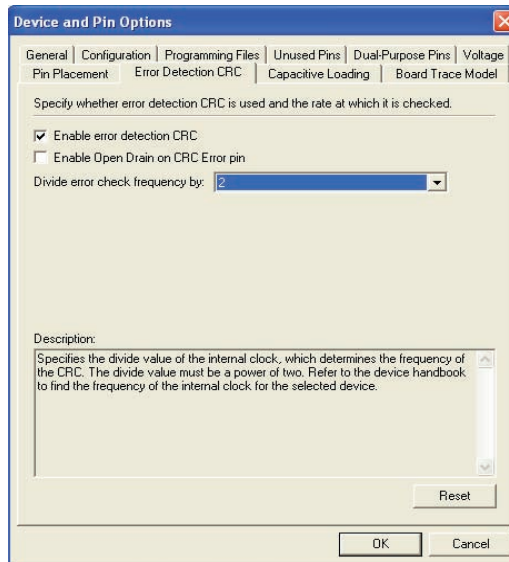
エラー検出 CRC 機能は、Quartus II ソフトウェアの **Device and Pin Options** ダイアログ・ボックスで制御されます。

CRC を使用したエラー検出機能をイネーブルするには、以下のステップを実行します。


1. Quartus II ソフトウェアを開いて、Stratix III デバイスを使用してプロジェクトを読み込みます。
2. Assignments メニューの **Settings** をクリックします。Settings ダイアログ・ボックスが表示されます。
3. **Category** リストで、**Device** を選択します。Device ページが表示されます。
4. **Device and Pin Options** をクリックします。Device and Pin Options ダイアログ・ボックスが表示されます (図 15-3 を参照)。

5. **Device and Pin Options** ダイアログの **Error Detection CRC** タブをクリックします。
6. **Enable error detection CRC** (図 15-3) をオンにします。

図 15-3. Quartus II ソフトウェアにおけるエラー検出 CRC 機能のイネーブル



7. 表 15-6 に示すとおり、**Divide error check frequency by** ボックスに有効な除数を入力します。

 この除数値は、CRC 回路をクロックするコンフィギュレーション・オシレータ出力クロックの周波数を分周します。

8. **OK** をクリックします。

CRC エラーからの回復

Stratix III デバイスを搭載するシステムは、デバイスのリコンフィギュレーションを制御しなければなりません。CRC_ERROR ピンでエラーを検出した後、nCONFIG を Low にストロープして、システムがデバイスをリコンフィギュレーションするのに安全な状況で実行するようシステムに指示します。

デバイスをリコンフィギュレーションすることによってデータ・ビットが正しい値に書き直されると、デバイスは正しく機能します。

アルテラ・デバイスではソフト・エラーはめったに発生しませんが、高信頼性アプリケーションではこれらのエラーに対応したデザインが要求される場合があります。

まとめ

エラー検出 CRC 機能の目的は、ソフト・エラーに起因する Stratix III デバイス内の任意のコンフィギュレーション CRAM ビットでの反転を検出することです。エラー検出回路を使用して、コンフィギュレーション CRAM ビットの完全性を連続的に確認することができます。



詳細は、「[Robust SEU Mitigation with Stratix III FPGAs White Paper](#)」を参照してください。

参考資料

この章では以下のドキュメントを参照しています。

- [「AN 357: アルテラ FPGA デバイスの CRC によるエラー検出」](#)
- [「Robust SEU Mitigation with Stratix III FPGAs White Paper」](#)
- [「Stratix III デバイスの TriMatrix エンベデッド・メモリ・ブロック」](#)

改訂履歴

表 15-8 に、本資料の改訂履歴を示します。

表 15-8. 改訂履歴		
日付およびバージョン	変更内容	概要
2008 年 5 月 v1.3	<ul style="list-style-type: none"> ● 「コンフィギュレーション・エラー検出」、「ユーザー・モード・エラー検出」、および「エラー検出のタイミング」の項を更新。 ● 表 15-3、表 15-6、および表 15-7を更新。 ● 図 15-2 および図 15-3を更新。 	テキスト、表、および図の更新。
2007 年 10 月 v1.2	<ul style="list-style-type: none"> ● 表 15-3をマイナー・アップデート。 ● 新しい項、「参考資料」を追加。 ● 参照にライブ・リンクを追加。 	マイナー・アップデート。
2007 年 5 月 v1.1	<ul style="list-style-type: none"> ● ページ 2、3、4 および 14 のマイナーな修正。表 15-5を更新。 	マイナー・アップデート。
2006 年 11 月 v1.0	<ul style="list-style-type: none"> ● 初版 	—

