

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。

SV51011-1.2


この章では、Stratix® V デバイスがユーザー・モードのときにエラー検出 CRC (Cyclic Redundancy Check) 機能をアクティブにして使用方法、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。このエラー検出機能は、Stratix V デバイスで強化されています。

この章は、以下の項で構成されています。

- 10-1 ページの「エラー検出の基礎」
- 10-2 ページの「コンフィギュレーション・エラー検出」
- 10-2 ページの「ユーザー・モード・エラー検出および訂正」
- 10-6 ページの「エラー検出ピンの説明」
- 10-7 ページの「エラー検出ブロック」
- 10-9 ページの「エラー検出のタイミング」
- 10-12 ページの「ソフトウェア・サポート」
- 10-12 ページの「CRC エラーからの回復」

航空電子、テレコム、システム・コントロール、および軍事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 V 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ことを確認する。
- 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 V デバイスの場合、Quartus®II ソフトウェア v10.0 からエラー検出 CRC 機能が提供されています。

Stratix V ファミリのエラー検出 CRC 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。

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。レシーバはこのファンクションを使用して、受信データ・フレームのチェックサムを計算し、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 V 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。

©2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX are Reg. U.S. Pat. & Tm. Off. and/or trademarks of Altera Corporation in the U.S. and other countries. All other trademarks and service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



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 16 ビット・コンフィギュレーション 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 V デバイスが受信したデータ・フレームに基づいて 16 ビット・コンフィギュレーション CRC 値を計算し、それをデータ・ストリームの計算済み 16 ビット・コンフィギュレーション CRC 値と比較します。これらの 16 ビット・コンフィギュレーション CRC 値が一致しない場合、nSTATUS が Low に設定されます。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続きます。

ユーザー・モード・エラー検出および訂正

Stratix V デバイスは、自動 SEU (Single Event Upset) 検出内蔵回路を提供します。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 V 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。Stratix V デバイスはエラー検出回路を内蔵し、コンフィギュレーション・ランダム・アクセス・メモリ (CRAM) セル内のソフト・エラーによるデータ破壊を検出します。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 32 ビット・エラー検出 CRC の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。

デバイスがユーザー・モードになった後エラー検出プロセスをイネーブルするためには、Quartus II ソフトウェアの **Device and Pin Options** ダイアログ・ボックスの **Error Detection CRC** ページで、**Enable Error Detection CRC_ERROR pin** オプションをオンにしてください。

このエラー検出機能は、デバイスの内容に基づいて継続的にコンフィギュレーションされた CRAM ビットの 32 ビット・エラー検出 CRC 値を計算し、それをコンフィギュレーションの終了時に取得した計算済みの 32 ビット・エラー検出 CRC 値と比較します。この 32 ビット・エラー検出 CRC 値は、コンフィギュレーション・ステージに計算されます。エラー検出回路は、フレームごとに 32 の CRC チェック・ビットを生成し、それらを CRAM に格納します。両方の 32 ビット・エラー検出 CRC 値が一致する場合、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) まで続きます。

シングル 32 ビット・エラー検出 CRC 計算は、フレームごとに実行されます。エラーが検出されていない場合、フレームの CRC 計算を完成すると、32 ビット・シグネチャは 0x00000000 となります。エラー検出回路がエラーを検出していない場合、出力信号 CRC_ERROR は Low に設定されます。CRAM ビット・エラーが検出された場合、シグネチャは 0 以外になり、そしてエラー検出回路はエラー・ビット位置の検索を開始します。

Stratix V デバイスのエラー検出回路は、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、3 ビット、4 ビットおよび 5 ビットのビット・エラーを検出することができます。SEU によって CRAM ビットが 5 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 99.9999% です。

エラー検出回路は、ビット位置をレポートし、シングル・ビット・エラーおよび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、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 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。エラー検出回路が別のフレームで次のエラーを検出する前に、エラー・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、エラー・メッセージ・レジスタ (EMR) からシフト・アウトする必要があります。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。

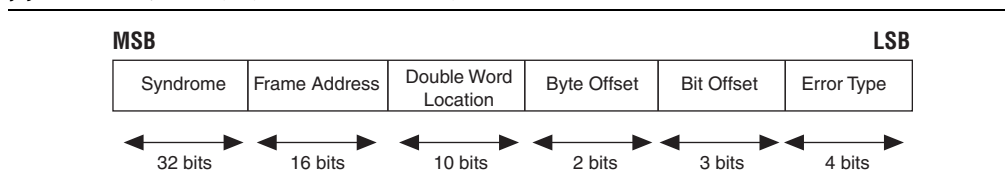
表 10-1 に、JTAG 命令の SHIFT_EDERROR_REG の命令コードを示します。

表 10-1. SHIFT_EDERROR_REG JTAG 命令

JTAG 命令	命令コード	説明
SHIFT_EDERROR_REG	00 0001 0111	JTAG 命令は、TDI ピンと TDO ピンの間で、EMR をエラー検出ブロックの JTAG ピンに接続します。

図 10-1 に、EMR の内容を示します。

図 10-1. エラー・メッセージ・レジスタ



エラーのタイプは、EMR の最初の 4 ビットで特定されます。表 10-2 に、EMR 内で表現されるエラーのタイプを示します。

表 10-2. エラー・メッセージ・レジスタ内のエラー・タイプ

エラーのタイプ				説明
ビット 3	ビット 2	ビット 1	ビット 0	
0	0	0	0	CRC エラーなし。
0	0	0	1	シングル・ビット・エラーの位置が特定されました。
0	0	1	0	隣接のダブル・ビット・エラーの位置が特定されました。
1	1	1	1	複数のエラーが発生しました。
その他				予約

EMR からエラー情報をシフトアウトするためのタイミング要件については、10-9 ページの「エラー検出のタイミング」を参照してください。

エラー検出回路は、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 32 ビット・エラー検出 CRC 値と 32 ビット・シグネチャの計算を継続します。CRC_ERROR 信号をモニタし、CRC 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 V デバイスのエラー検出回路は、32 ビット CRC-ANSI 規格（32 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 32 ビット CRC シグネチャは、CRAM に格納されます。ストレージの合計サイズは、「32（1 フレームあたりのビット数）× フレーム数」になります。

Stratix V デバイスのエラー検出 CRC 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。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。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。MLAB および M20K 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。

 Stratix V メモリ・ブロックにおけるエラー検出について詳しくは、「[TriMatrix Embedded Memory Blocks in Stratix V Devices](#)」の章を参照してください。

Stratix V デバイスでは、エラー検出回路はエラー検出機能に加えて、エラー訂正および内部スクラビングをサポートします。内部スクラビングとは、検出されたソフト・エラーを内部で訂正する機能です。これはフレームごとに実行されます。内部スクラビングがイネーブルされると、デバイスは動作している間に、CRAM ビット内のシングル・ビット・エラーまたは隣接のダブル・エラーを訂正します。

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 V デバイスのエラー・インジェクションに使用される 47 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 10-3 に、EDERROR_INJECT JTAG 命令を示します。

表 10-3. EDERROR_INJECT JTAG 命令

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 47 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 JTAG 命令を使用して、リードバック・ビットを反転します。Stratix V デバイスを、強制的にエラー・テスト・モードにします。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。


 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、10-8 ページの「エラー検出レジスタ」を参照してください。

表 10-4 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。

表 10-4. フォールト・インジェクション・レジスタ

説明	ビット [46..43]				エラー・インジェクションのタイプ	ビット [42..32]	ビット [31..0]
	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
	ビット [46]	ビット [45]	ビット [44]	ビット [43]			
内容	0	0	0	0	エラー・インジェクションなし	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。
	0	0	0	1	シングル・エラー・インジェクション		
	0	0	1	0	隣接のダブル・エラー・インジェクション		
	その他				予約		

エラー検出ピンの説明

表 10-5 に、CRC_ERROR について説明します。

表 10-5. CRC_ERROR Pin Description

ピン名	ピン・タイプ	説明
CRC_ERROR	I/O、出力、 または出力 オープン・ ドレイン	<p>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、このピンがユーザー I/O ピンになります。WYSIWYG 機能を使用するときに、crcerror ポートを WYSIWYG アトムから専用の CRC_ERROR ピンまたは任意のユーザー I/O に配線することができます。crcerror ポートをユーザー I/O に配線するには、crcerror ポートと I/O の間に D タイプ・フリップ・フロップを挿入する必要があります。</p> <p>デフォルトでは、エラー検出 CRC 回路がイネーブルされると、Quartus II ソフトウェアは CRC_ERROR ピンを出力オープン・ドレインとして設定します。オプションで、Quartus II ソフトウェアの Device and Pin Option ダイアログ・ボックスの Error Detection CRC ページで、Enable open-drain on CRC_ERROR pin をオフにすることで、このピンを出力に設定できます。CRC_ERROR が出力として使用される場合、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。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。このピンを 10 kΩ 抵抗を介して V_{CCPGM} に接続することで、CRC_ERROR 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仕様に応じて、ユーザーはこのピンを異なるプルアップ電圧に接続できます。</p>


エラー検出ブロック

エラー検出ブロックには、Stratix V デバイス内のコンフィギュレーション CRAM ビット用の 32 ビットエラー検出 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。CRC 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。表 10-6 に、コンフィギュレーション・ビットをチェックするための 2 種類の CRC 検出を示します。

表 10-6. 2 種類の CRC 検出

ユーザー・モード CRC エラー検出	コンフィギュレーション CRC エラー検出
<ul style="list-style-type: none"> ■ これは、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 (32 ビット・エラー検出 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 32 ビット・エラー検出 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を検索します。 ■ 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 32 ビット・エラー検出 CRC 結果を読み出し、32 ビット・エラー検出 CRC ビットをテスト用に 32 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、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。 	<ul style="list-style-type: none"> ■ これは、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・コンフィギュレーション 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 V デバイスにロードされた後、計算済みのコンフィギュレーション 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 16 ビット・コンフィギュレーション CRC 値が計算されます。計算済みのコンフィギュレーション CRC と計算されたコンフィギュレーション 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・コンフィギュレーション CRC があります。データ・フレームが多数あるため、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・コンフィギュレーション 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。

 「エラー検出レジスタ」の項では、ユーザー・モード CRC エラー検出を中心に説明します。

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 32 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 10-2 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・ブロックを示します。

図 10-2. 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

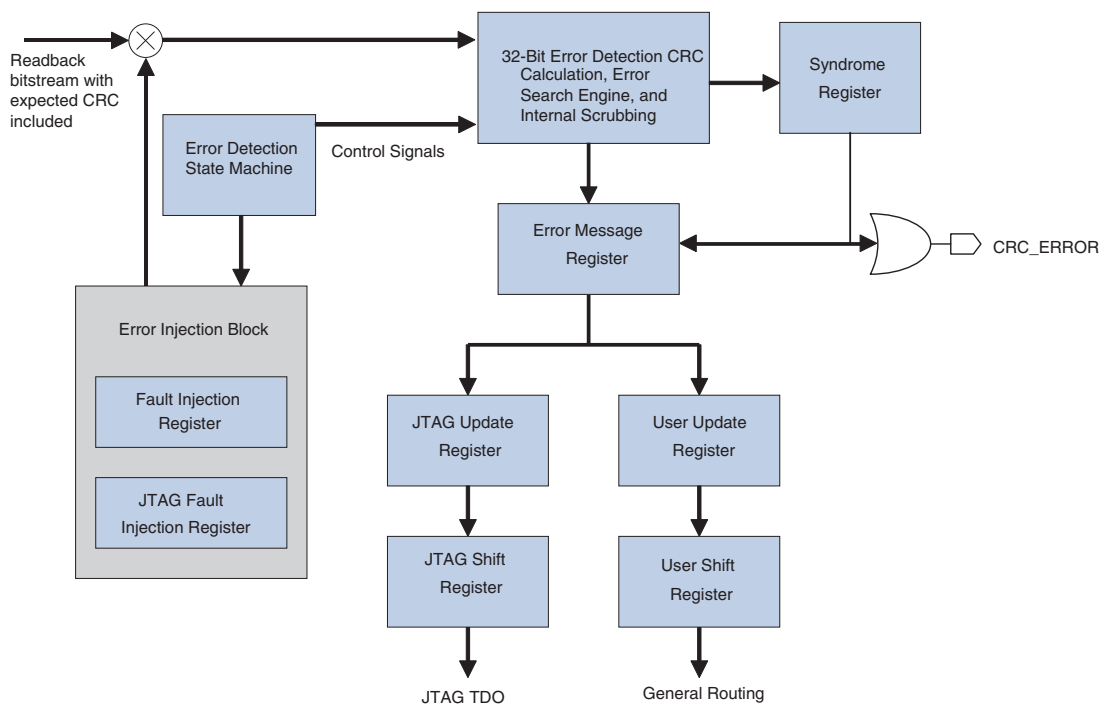


表 10-7 は、図 10-2 に示すレジスタについて説明します。

表 10-7. エラー検出レジスタ (1 / 2)

レジスタ	説明
シンドローム・レジスタ	この 32 ビットのレジスタには、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 67 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、シングルおよび隣接のダブル・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、EMR では識別されません。レジスタの内容は、SHIFT_EDERROR_REG JTAG 命令またはコア・インタフェースを通してコアにシフト・アウトされます。
JTAG アップデート・レジスタ	この 67 ビット・レジスタは、内容が検証されてから 1 サイクル後に EMR 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 EMR 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。

表 10-7. エラー検出レジスタ (2 / 2)

レジスタ	説明
ユーザー・アップデート・レジスタ	この 67 ビット・レジスタは、内容が検証されてから 1 サイクル後に EMR 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、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。この要件により、ユーザー・アップデート・レジスタへの EMR 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、SHIFT_EDERROR_REG JTAG 命令により JTAG アップデート・レジスタの内容をサンプリングし、読み出すことができます。
ユーザー・シフト・レジスタ	この 67 ビット・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、読み出すことができます。
JTAG フォールト・インジェクション・レジスタ	この 47 ビット・レジスタは、EDERROR_INJECT 命令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 47 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

QuartusII ソフトウェアを通してエラー検出 CRC 機能がイネーブルされると、デバイスはユーザー・モードに入った後、自動的に CRC エラー検出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、EMR 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。次のフレームにエラーが含まれる場合、EMR 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 10-8 に、最小および最大エラー検出周波数を示します。


表 10-8. 最小および最大エラー検出周波数

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 V	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 (10-12 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 (式 10-1)。

式 10-1.

$$\text{エラー検出周波数} = \frac{100\text{MHz}}{2^n}$$

 Stratix V デバイスでの CRC 計算はフレームごとに実行されるため、エラー検出周波数がフレームで行われるエラー検出プロセスの周波数を反映しています。

EMR はエラーが発生するたびにアップデートされます。次のエラー位置が検出される前にエラー位置とエラー・メッセージがシフト・アウトされていない場合、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。この問題を回避するには、1 フレームの CRC 検証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。EMR 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。ただし、エラー検出クロック周波数を低くすると、SEU イベントのエラー回復時間が長くなります。

表 10-9 に、Stratix V デバイスにおける EMR の各アップデートの推定最小時間間隔を示します。

表 10-9. エラー・メッセージ・レジスタの最小アップデート間隔 — 暫定値

ファミリ	デバイス	タイミング間隔 (μs)
Stratix V GX	5SGXA3	2.73
	5SGXA4	2.73
	5SGXA5	3.59
	5SGXA7	3.59
	5SGXA9	4.87
	5SGXAB	4.87
	5SGXB5	3.73
	5SGXB6	3.73
Stratix V GT	5SGTC5	3.59
	5SGTC7	3.59
Stratix V GS	5SGSD2	未定
	5SGSD3	未定
	5SGSD4	2.99
	5SGSD5	2.99
	5SGSD6	4.55
	5SGSD8	4.55
Stratix V E	5SEE9	4.87
	5SEEB	4.87

エラー検出回路が最初のフレームから最後のフレームまでチェックするのに必要な CRC 計算時間は、デバイスとエラー検出クロック周波数によって決まります。最小 CRC 計算時間は、最大エラー検出周波数によって算出されます (除数 1 を使用)。最大 CRC 計算時間は、最小検出周波数によって算出されます (除数 8 を使用)。式 10-2 によって、ほかの有効な除数 (n) で、特定のデバイス集積度の CRC 計算時間を見積もることができます。

式 10-2.

$$\text{CRC計算時間} = \frac{\text{最小計算時間}}{2} \times 2^n$$

表 10-10 に、Stratix V デバイスの最小および最大クロック周波数での各 CRC 計算に対する見積り時間を示します。

表 10-10. 特定のデバイス集積度に対して — 暫定値

ファミリ	デバイス	最小時間 (μs)	最大時間 (μs)
Stratix V GX	5SGXA3	2.71	5.42
	5SGXA4	2.71	5.42
	5SGXA5	3.57	7.14
	5SGXA7	3.57	7.14
	5SGXA9	4.85	9.70
	5SGXAB	4.85	9.70
	5SGXB5	3.71	7.42
	5SGXB6	3.71	7.42
Stratix V GT	5SGTC5	3.57	7.14
	5SGTC7	3.57	7.14
Stratix V GS	5SGSD2	TBD	未定
	5SGSD3	TBD	未定
	5SGSD4	2.97	5.94
	5SGSD5	2.97	5.94
	5SGSD6	4.53	9.06
	5SGSD8	4.53	9.06
Stratix V E	5SEE9	4.85	9.70
	5SEEB	4.85	9.70

ソフトウェア・サポート

QuartusII ソフトウェアは、バージョン 10.0 以降で Stratix V のデバイスエラー検出 CRC 機能をサポートしています。**Device and Pin Options** ダイアログ・ボックスでこの機能をイネーブルすると、オプションの CRC_ERROR 兼用ピンに CRC_ERROR 出力が生成されます。

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. QuartusII ソフトウェアを開いて、Stratix V 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 **Device** をクリックします。**Device** ダイアログ・ボックスが表示されます。
3. **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。
4. **Category** リストで、**Error Detection CRC** をクリックします。
5. **Enable Error Detection CRC_ERROR pin** をオンにします。
6. デフォルトでは、エラー検出 CRC 回路がイネーブルされる場合、Quartus II ソフトウェアは CRC_ERROR ピンを出力オープン・ドレインとして設定します。**Enable open-drain on CRC_ERROR pin** をオフにすることで、このピンを出力として設定できます。
7. 9 ページの表 10-8 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。
8. 内蔵エラー検出機能のオン/オフは、**Enable internal scrubbing** オプションによって制御されます。
9. **OK** をクリックします。

CRC エラーからの回復

Stratix V 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 にストロブして、システムが安全な状況で実行するようシステムに指示します。

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。

改訂履歴

表 10-11 に、本資料の改訂履歴を示します。

表 10-11. 改訂履歴

日付	バージョン	変更内容
2011 年 5 月	1.2	<ul style="list-style-type: none">■ 11.0 リリースにより、この章を Volume 2 に移動■ 表 10-9 および 表 10-10 を更新■ テキストのマイナーな編集
2010 年 12 月	1.1	10.1 リリース。内容の変更はなし
2010 年 7 月	1.0	初版

