



インテル® Stratix® 10 SEU (Single Event Upset) 緩和 ユーザーガイド

インテル® Quartus® Prime 開発デザインスイートの更新情報: **18.1**



UG-S10SEU | 2018.10.10

最新版をウェブからダウンロード: [PDF](#) | [HTML](#)

目次

1. Intel® Stratix® 10 SEU 緩和の概要	3
1.1. インテル Stratix 10® デバイスでの SEU の緩和.....	3
1.2. コンフィグレーション RAM.....	4
1.3. メモリーブロックの誤り訂正コードのサポート.....	4
1.4. トリプル・モジュール・リダンダンシー.....	5
1.5. 故障率.....	6
2. インテル Stratix 10 CRAM 用の緩和手法	7
2.1. CRAM エラー検出と訂正.....	7
2.1.1. エラー・メッセージ・キュー.....	7
2.1.2. SEU_ERROR ピンの動作.....	8
2.2. 内部スクラビングとプライオリティー・スクラビング.....	8
2.2.1. 内部スクラビング.....	9
2.2.2. プライオリティー・スクラビング.....	9
2.3. SEU センシティブィティー・プロセッシング.....	9
2.3.1. Advanced SEU Detection IP コア.....	9
2.4. デザイン階層のセンシティブィティーの指定.....	14
2.4.1. 階層タグ付け.....	15
2.5. 機能不調に対するシステムのレスポンスの評価.....	15
2.5.1. インテル Quartus Prime Fault Injection Debugger.....	15
3. Intel Stratix 10 SEU 緩和実装ガイド	17
3.1. SEU_ERROR ピンの設定.....	17
3.2. インテル Quartus Prime SEU ソフトウェアの設定.....	17
3.3. プライオリティー・スクラビングの有効化.....	18
3.4. 階層タグ付けの実行.....	18
3.5. センシティブィティー・マップ・ヘッダー・ファイルのメモリーへのプログラミング.....	19
3.6. センシティブィティー・マップ・ヘッダーのルックアップの実行.....	20
3.7. Fault Injection Debugger の使用.....	22
3.7.1. デバイスと Fault Injection Debugger のコンフィグレーション.....	22
3.7.2. フォールト・インジェクションの領域制約.....	22
3.7.3. エラーの挿入.....	23
4. Advanced SEU Detection Intel Stratix 10 FPGA IP 参照事項	25
4.1. Advanced SEU Detection IP コアのパラメーター設定.....	25
4.2. Advanced SEU Detection IP コアのポート.....	26
5. Intel Stratix 10 Fault Injection Debugger 参照事項	28
5.1. Fault Injection Debugger インターフェイス・パラメーター.....	28
5.2. Fault Injection Debugger のコマンドライン・インターフェイス.....	28
6. Intel Stratix 10 SEU 緩和 ユーザーガイド・アーカイブ	30
7. インテル Stratix 10 SEU 緩和 ユーザーガイドの改訂履歴	31

1. Intel® Stratix® 10 SEU 緩和の概要

SEU (Single Event Upset) とは、FPGA の内部メモリーエレメントの状態がまれに意図せぬ変化をすることで、宇宙放射線の影響によって発生します。この状態変化はソフトエラーであり、FPGA が永久的に破損することはありません。FPGA は、意図せぬメモリー状態になるため、バックグラウンド・スクラビングによって問題が修正されるまでは、誤って動作する可能性があります。

インテル® Quartus® Prime 開発ソフトウェアで提供されている機能の中には、SEU の影響、つまりソフトエラーの検出と修正、またデザインに対する SEU の影響の特性評価を行うためのものがあります。また、一部の Intel FPGA には、エラーの検出と修正に役立つ専用の回路が含まれています。

Intel FPGA のメモリーは、ユーザーロジック(ブロックメモリーとレジスター)および Configuration Random Access Memory (CRAM) 内にあります。インテル Quartus Prime Programmer では CRAM のロードには .sof ファイルが使用されます。その後 CRAM では、すべての FPGA ロジックと配線のコンフィグレーションが行われます。SEU で CRAM ビットを検出した場合、デバイスで CRAM ビットを使用していなければその影響は無害ですが、影響が深刻になる可能性があるのは、SEU によって重要なロジックまたは内部信号配線が影響を受けた場合です。

多くの場合、デザインで SEU が発生する可能性は低いので緩和は必要ありませんが、複数の高密度コンポーネントを含むシステムなどの非常に複雑なシステムでは、エラー率が重要なシステムデザインの要素になる可能性があります。システムに複数の FPGA が含まれていて、非常に高い信頼性と可用性が必要な場合は、ソフトエラーの影響を考慮し、この章で説明する手法で、このような種類のエラーを検出して修復してください。

関連情報

- [Introduction to Single-Event Upsets](#)
- [Understanding Single Event Functional Interrupts in FPGA Designs](#)
- [Intel Stratix 10 SEU 緩和 ユーザーガイド・アーカイブ \(30 ページ\)](#)
旧バージョンの Advanced SEU Detection Intel® Stratix® 10 FPGA IP ユーザーガイドの一覧を提供しています。

1.1. インテル Stratix 10® デバイスでの SEU の緩和

インテル Stratix 10® SEU 緩和機能がシステムに有益な点は次の通りです。

- システムをつねに正常に機能させます。
- SEU イベント起因のシステム誤動作を防止します。
- システムに対してクリティカルな SEU イベントを処理します。

表 1. インテル Stratix 10 デバイス向け SEU 緩和のエリアとアプローチ

エリア	SEU 緩和アプローチ
誤り検出訂正	誤り検出訂正 (EDC) 機能を有効にすると、CRAM SEU イベントが検出され、CRAM 内容が自動的に訂正されます。
メモリーブロック誤り訂正コードのサポート	インテル Stratix 10 では、M20K メモリーブロックのデザインに特別なレイアウト手法と誤り訂正コード (ECC) を使用しており、SEU Failures in time (FIT) レートをほぼゼロに低減します。
SEU センシティブィティ・プロセッシング	センシティブィティ・プロセッシングを使用すると、CRAM ビット位置の SEU が、コンパイルされた FPGA のデザイン・ビットストリームファイルの機能に対してクリティカルであるかどうかを識別することができます。
フォールト・インジェクション	フォールト・インジェクション機能を使用すると、SEU イベントに対するシステムの反応を検証することができます。これには CRAM ステートを変更してエラーをトリガーします。
階層的タグ付け	センシティブィティ・プロセッシングおよびフォールト・インジェクションの補完機能。SEU をレポートし、デザインロジックの特定部分へのインジェクションを制約します。
トリプル・モジュール・リダンダンシー (TMR)	TMR 手法は、ステートマシン等のクリティカルなロジックで実行することができます。

1.2. コンフィグレーション RAM

FPGA では、ユーザーロジック(パルク・メモリーおよびレジスター)内と Configuration RAM (CRAM) 内の両方のメモリーを使用します。CRAM は、ユーザーデザインがロードされたメモリーであり、デバイス内の全てのロジックと配線のコンフィグレーションを行います。SEU で CRAM ビットを検出しても、その CRAM ビットを使用していなければ影響は無害です。一方で、機能エラーが生じる可能性があるのは、内部信号配線やデザインの一部のルックアップ・テーブル・ロジック・ビットが影響を受けた場合です。

関連情報

Intel® Stratix® 10 Configuration User Guide

Intel® Stratix® 10 デバイスの CRAM およびユーザーデザインに関する詳細情報を提供しています。

1.3. メモリーブロックの誤り訂正コードのサポート

ECC ではデータエラーの検出と訂正をメモリーの出力で行います。

M20K ブロックと eSRAM ブロックのみで ECC 機能をサポートしています。

ECC 機能を使用する場合、次の機能は使用できません。

- バイトイネーブル
- コヒーレンシー読み出し

M20K ブロック

M20K ブロックの場合、ECC では、シングルエラー訂正、ダブル隣接エラー訂正、およびトリプル隣接エラーを 32 ビットワードで実行しますが、2 ビット以上の非隣接エラーの検出または訂正は保証されていません。



M20K ブロックに内蔵された ECC のサポートは、×32 幅シングル・デュアルポート・モードで有効です。

- ECC 機能を使用すると、M20K の動作は、ECC シングル・デュアルポート以外のモードに比べて低速になります。ただし、オプションの ECC パイプライン・レジスターを出力デコーダーの前で有効にすると、パイプライン ECC モード以外のモードと比べて、より高いパフォーマンスが実現可能です。ただし、1 サイクルのレイテンシーが追加されます。
- e (エラー) と ue (訂正不可エラー) の 2 つの ECC ステータスフラグ信号によって、M20K ECC ステータスが表示されます。ステータスフラグは、メモリーブロックからの通常の出力の一部です。ECC を使用すると、この 2 つパリティビットへのアクセスはできません。これは、ECC ステータスフラグによってパリティビットが置き換えられるからです。

eSRAM ブロック

eSRAM ブロックの場合、ECC ではシングルエラー訂正とダブルエラー検出を 64 ビットワードで実行します。

eSRAM ブロックに内蔵された ECC のサポートは、×64 幅シングル・デュアルポート・モードで有効です。

- 2 つの ECC ステータスフラグ信号の `c{7:0}_error_correct_0` (エラー訂正済み) および `c{7:0}_error_detect_0` (エラー検出済み) では、eSRAM ECC ステータスを示します。

関連情報

- [Embedded Memory User Guide](#)
エンベデッド・メモリー IP コアを使用した ECC の実装に関する詳細情報を提供しています。
- [Intel® Stratix® 10 Embedded Memory User Guide](#)
Intel® Stratix® 10 メモリーブロック内の ECC に関する詳細情報を提供しています。

1.4. トリプル・モジュール・リダンダンシー

トリプル・モジュール・リダンダンシー (TMR) は、SEU が原因のダウンタイムを許容できないシステムに対して使用します。TMR は、確立した SEU 軽減手法として、ハードウェア・フォールト・トレランスの向上のために使用されます。TMR のデザインには、同一のハードウェア・インスタンスが 3 つあり、出力に投票ハードウェアが備えられています。SEU によっていずれかのハードウェアのインスタンスが影響を受ける場合、投票ロジックでは過半数出力を記録します。この操作によって、誤動作しているハードウェアがマスクされます。

TMR を使用すると、シングル SEU ではデザインのダウンタイムはなくて済みます。システムで障害のあるモジュールが検出された場合、システムではエラーをスクラブするためにそのモジュールを再プログラムします。誤り検出訂正時間は、SEU イベントの MTBF より何桁も少なくなります。したがって、システムではソフト割り込みを修復して、別の SEU によって TMR アプリケーション内の別のインスタンスが影響を受けないようにします。

TMR の欠点は、ハードウェア・リソースのコストです。3 倍のハードウェアが必要になり、加えて投票ロジックも必要です。このハードウェア・コストを最小限に抑えるために、TMR の実装は、デザイン内の最もクリティカルな部分のみで実行します。

複数の方法を使用して、TMR デザインの生成のために、指定機能の自動複製および必要な投票ロジックの自動合成を行います。Synopsys では自動 TMR 合成を提供しています。

1.5. 故障率

Soft Error Rate (SER) または SEU の信頼性は、Failure in Time (FIT) で表されます。1FIT ユニツトは、操作 10 億時間あたり 1 つのソフトエラーの発生です。

- たとえば、5,000 個の FIT があるデザインでは、10 億時間(または 8,333.33 年)で平均 5,000 の SEU イベントが発生します。SEU イベントは統計的に独立しているため、FIT は加法的です。1 つの FPGA に 5,000 個の FIT がある場合、10 個の FPGA には 50,000 個の FIT(または 8,333 年間で 50K の障害)があります。

もう 1 つの信頼性測定法は、平均故障時間 (MTTF) です。これは、FIT または 1/FIT の逆数です。

- 標準故障単位で 5,000 の FIT / 10 億時間の場合、MTTF は、
 $1 / (5,000 / 1\text{Bh}) = 10 \text{ 億} / 5,000 = 200,000 \text{ 時間} = 22.83 \text{ 年}$ です。
 $1 / (5,000 / 1\text{Bh}) = 1 \text{ billion} / 5,000 = 200,000 \text{ hours} = 22.83$

SEU イベントはポアソン分布に従います。平均故障間時間 (MTBF) の累積分布関数 (CDF) は指数分布です。故障率計算の詳細については、*Intel FPGA Reliability Report* を参照してください。

中性子 SEU の発生率は、高度、緯度、およびその他の環境要因により異なります。インテル Quartus Prime 開発ソフトウェアで提供している SEU FIT レポートは、ニューヨーク州マンハッタンの海拔でのコンパイルに基づいています。JESD 89A 仕様でテスト・パラメーターを定義しています。

ヒント: このデータを他の場所や高度に変換するには、www.seutest.com の計算機などを使用します。さらに、プロジェクト内の SEU 率を調整するために、相対中性子束 (www.seutest.com で計算) をプロジェクトの .qsf ファイルに含めてください。

2. インテル Stratix 10 CRAM 用の緩和手法

この章では、インテル Stratix 10 CRAM 用の SEU 緩和手法について説明します。エンベデッド・メモリーの ECC 機能の詳細については、インテル *Stratix 10 Embedded Memory User Guide* を参照してください。

関連情報

- [Embedded Memory User Guide](#)
エンベデッド・メモリー IP コアを使用した ECC の実装に関する詳細情報を提供しています。
- [Intel® Stratix® 10 Embedded Memory User Guide](#)
Intel® Stratix® 10 メモリーブロック内の ECC に関する詳細情報を提供しています。

2.1. CRAM エラー検出と訂正

Intel® Stratix® 10 デバイスの機能では、オンチップ EDC 回路でソフトエラーを検出します。SEU イベントによって生じたエラーが訂正可能な場合は、Intel Stratix 10 内部スクラビング機能を有効にすると、FPGA によってエラーが訂正されます。

表 2. エラータイプの検出と訂正

エラータイプ	検出	訂正
シングル・ビット・エラー	可能	可能
ダブル隣接エラー ⁽¹⁾	可能	—
複数ビットエラー ⁽¹⁾	8CRAM ビットの長方形ボックスに収まる最大 8CRAM ビット (8x1、4x2、1x8 または 2x4 エラー) までを検出	—

2.1.1. エラー・メッセージ・キュー

インテル Stratix 10 デバイス・エラー・メッセージ・キューでは、エラーメッセージの保管を SEU エラーの検出時に行います。エラー・メッセージ・キューには、異なるメッセージを最大 4 つまで格納できます。各エラーメッセージには、セクターアドレス、エラーの種類、およびエラーの位置に関する情報が含まれています。エラー・メッセージ・キューの内容を取得するには、次のツールを使用します。

- Fault Injection Debugger ツール
- Advanced SEU Detection Intel Stratix 10 FPGA IP

⁽¹⁾ Intel Stratix 10 デバイスによってサポートされています。インテル Quartus Prime でこの機能を有効にするサポートは、将来のリリースで利用可能になる予定です。詳細については、最寄りの Intel FPGA 担当者にお問い合わせください。

表 3. エラー・メッセージ・キューの説明

名前	幅	ビット	説明
セクターアドレス	32	31:24	予約済み
		23:16	エラーのあるセクターのアドレス
		15:4	予約済み
		3:0	セクター 1 で検出されたエラー数
エラー位置 ⁽²⁾ ビットポジション	32	31:29	ビット 31:29 - エラーの種類: • 01 = シングルビット • 10 = マルチビット
		28	訂正ステータス: • 0 = 未修正 • 1 = 訂正済
		27:24	予約済み
		23:12	フレーム内のビット位置
		11:0	行インデックスとフレーム・インデックスの組み合わせ

注意: インテルでは、**Internal Scrubbing** 機能をオンにすることをお勧めします。あるセクターでエラーが検出されたときに **Internal Scrubbing** オプションを有効にしていなかった場合、その特定のセクターに対する SEU 機能はオフになります。さらに、その後続 SEU が同じセクター内で発生した場合、訂正が可能でも不可能でもエラーは検出されません。

関連情報

- [Advanced SEU Detection IP コアのポート \(26 ページ\)](#)
- [Fault Injection Debugger の使用 \(22 ページ\)](#)

2.1.2. SEU_ERROR ピンの動作

エラー・メッセージ・キューにエラーメッセージが 1 つ以上含まれている場合、SEU_ERROR 信号は High になります。キューにエラーメッセージがあると、信号は High の状態のままです。SEU_ERROR 信号が Low になるのは、SEU エラー・メッセージ・キューが空のときだけです。これは、すべてのエラーメッセージをシフトアウトした後に発生します。

SEU_ERROR ピン機能に設定して、SEU_ERROR ピンの動作を観察してください。

関連情報

[インテル Quartus Prime SEU ソフトウェアの設定 \(17 ページ\)](#)

2.2. 内部スクラビングとプライオリティー・スクラビング

Intel Stratix 10 デバイスでは、自動 CRAM エラー訂正をサポートしています。このとき、元の CRAM の内容を元のプログラミング・ビットストリームの外部コピーから再ロードする必要はありません。

⁽²⁾ エラー位置によって提供されるのは、シングル・ビット・エラーのビット位置のみです。複数ビットエラーの場合、bit [23:0] は 0 を返します。



2.2.1. 内部スクラビング

内部スクラビング機能では、シングルビット・エラーを自動的に訂正します。

内部スクラビング機能を有効にした場合でも、リカバリーシーケンスの計画が必要です。スクラビング機能では、CRAM アレイを復元し、意図したコンフィグレーションにすることができますが、ソフトエラーの検出と訂正の間にはレイテンシーがあります。このレイテンシー中は、Intel Stratix 10 デバイスがエラーで動作している可能性があります。

2.2.2. プライオリティー・スクラビング

デザインの一部を指定して、内部スクラビングの優先順位が高いセクターとすることができます。Intel Stratix 10 EDC 回路では、優先順位の高いセクターで発生するエラーの検出および訂正後に他のセクターのエラーを検出して訂正します。

関連情報

[プライオリティー・スクラビングの有効化 \(18 ページ\)](#)

2.3. SEU センシティブィティ・プロセッシング

動作中の FPGA をリコンフィグレーションすると、その FPGA を使用しているシステムに重大な影響を与えます。SEU リカバリーの計画時には、FPGA のステートを現在のシステムのステートと整合させるために必要な時間を考慮してください。たとえば、内部ステートマシンが不正なステートにある場合、リセットが必要になることがあります。さらに、周囲のロジックでもこういった想定外の動作の考慮が必要になることがあります。

SEU では、実装されたデザインで使用していない CRAM ビットに影響を与えることがよくあります。十分に利用されている FPGA デザインであっても、コンフィグレーション・ビットの多くは使用されていません。これは、デザインで使用していないロジックおよび配線ワイヤを制御しているためです。実装方法によりませんが、最も密に使用されているデバイスでも使用されているのは CRAM ビットの 40% だけです。つまり、SEU イベントのうち 40% のみで対応が必要であり、SEU イベントの 60% は無視できるということです。使用しているビットはクリティカル・ビットとして判断され、使用していないビットはクリティカルではないビットと判断されます。

さらに、実装されたデザインのうちの一部が、FPGA 機能で使用されていない場合もあります。例として挙げられるのは、テスト回路が実装はされているがデバイスの動作には重要ではない場合や、その他のクリティカルではない機能で、ログには記録されるが再プログラムやリセットの必要がない場合です。

SEU センシティブィティ・プロセッシングの実行には、Advanced SEU Detection IP コアを使用できます。

2.3.1. Advanced SEU Detection IP コア

Advanced SEU Detection IP コアでは、以下の機能を提供しています。

- Secure Device Manager (SDM) と通信して SEU イベントを検出し、SDM からコマンドまたはレスポンスを送受信して SEU エラーをレポートします。
- Sensitivity Map Header (.smh) リビジョン 4 ファイルを読み出し、オンチップまたはオフチップ・ルックアップ・センシティブィティ・プロセッシングが可能になり、デバイスで発生した SEU エラーのクリティカルリティが、.smh ファイルで指定された領域に基づいてレポートされます。

Advanced SEU Detection IP コアを使用すると、実行時に SEU エラーのセンシティブティイー・プロセッシングを実行できます。Advanced SEU Detection IP コアでは、次の実装をサポートしています。

- オンチップ・ルックアップ・センシティブティイー・プロセッシング - センシティブティイー・プロセッシング・ソフト IP では、エラー位置のレポートおよびルックアップを提供します。
- オフチップ・ルックアップ・センシティブティイー・プロセッシング - マイクロプロセッサなどの外部ユニットでは、エラー位置の探索を実行するのにエラー・メッセージ・キュー情報を使用します。

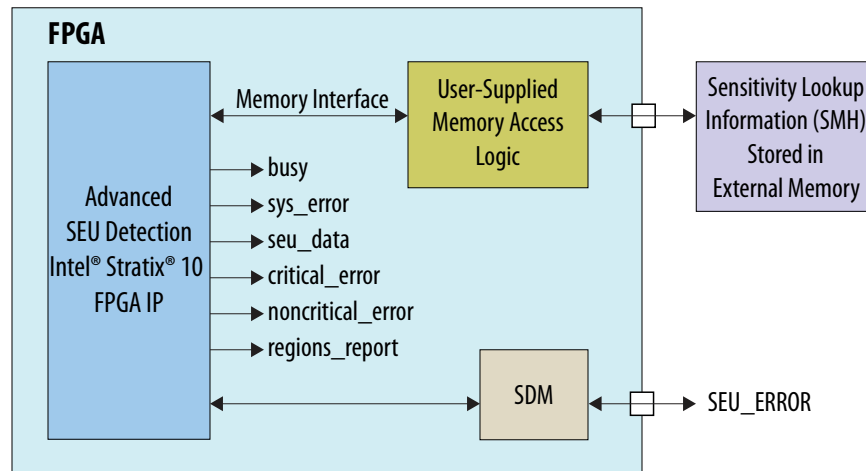
関連情報

- [SMH ルックアップ \(13 ページ\)](#)
- [センシティブティイー・マップ・ヘッダーのルックアップの実行 \(20 ページ\)](#)
- [階層タグ付けの実行 \(18 ページ\)](#)
- [Advanced SEU Detection IP コアのパラメーター設定 \(25 ページ\)](#)
- [Advanced SEU Detection IP コアのポート \(26 ページ\)](#)
- [インテル Quartus Prime SEU ソフトウェアの設定 \(17 ページ\)](#)

2.3.1.1. オンチップ・ルックアップ・センシティブティイー・プロセッシング

Advanced SEU Detection IP コアでは、エラー・メッセージ・キューの内容を読み出し、シングル・ビット・エラーの位置をセンシティブティイー・マップと比較します。この検査によって、障害がデバイスの動作に影響を与えるかどうかを判断します。

図 -1: Advanced SEU Detection IP コアを使用したオンチップ・ルックアップ・センシティブティイー・プロセッシングのシステム概要



オンチップ・ルックアップ・センシティブティイー・プロセッシングは次のとおりです。

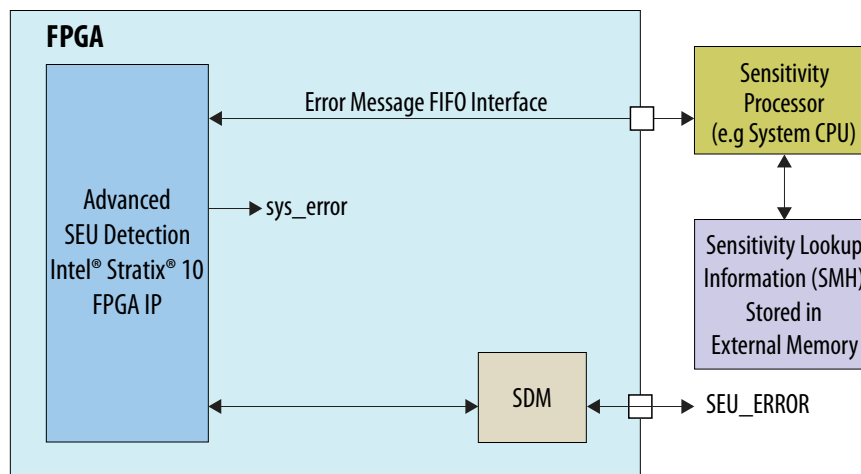
1. SEU エラーが発生すると、SEU_ERROR がアサートされます。
2. Advanced SEU Detection IP コアでは、SEU エラーメッセージを SDM から取得します。
注意: Advanced SEU Detection SEU エラーメッセージの取得中にシステムでエラーが発生すると、IP コアでは sys_error 信号がアサートされます。
3. Advanced SEU Detection IP コアによってセンシティブティイー・プロセッシングが開始されます。この処理中、

- Advanced SEU Detection IP コアによって busy 信号がアサートされます。
 - Advanced SEU Detection IP コアでは、.smh ファイルを読み出します。メモリー・アクセス・ロジックおよび外部メモリーの情報を入力してください。
4. Advanced SEU Detection IP コアでは、busy 信号をアサートして、センシティブィティ・プロセッシングの完了を示します。また、SEU エラーのクリティカルリティを、次の信号を介してレポートします。
- critical_error
 - noncritical_error
 - regions_report
 - seu_data (オプション)

2.3.1.2. オンチップ・ルックアップ・センシティブィティ・プロセッシング

Advanced SEU Detection IP コアでは、エラー・メッセージ・キューの内容を読み取り、システム・プロセッサに情報を提示します。プロセッサによって、障害がデバイスの動作に影響を与えるかどうかを判断します。システム・プロセッサでは、アルゴリズムを実装し、.smh に対するルックアップを実行します。

図 -2: Advanced SEU Detection IP コアを使用したオフチップ・ルックアップ・センシティブィティ・プロセッシングのシステム概要



オフチップ・ルックアップ・センシティブィティ・プロセッシングは次のとおりです。

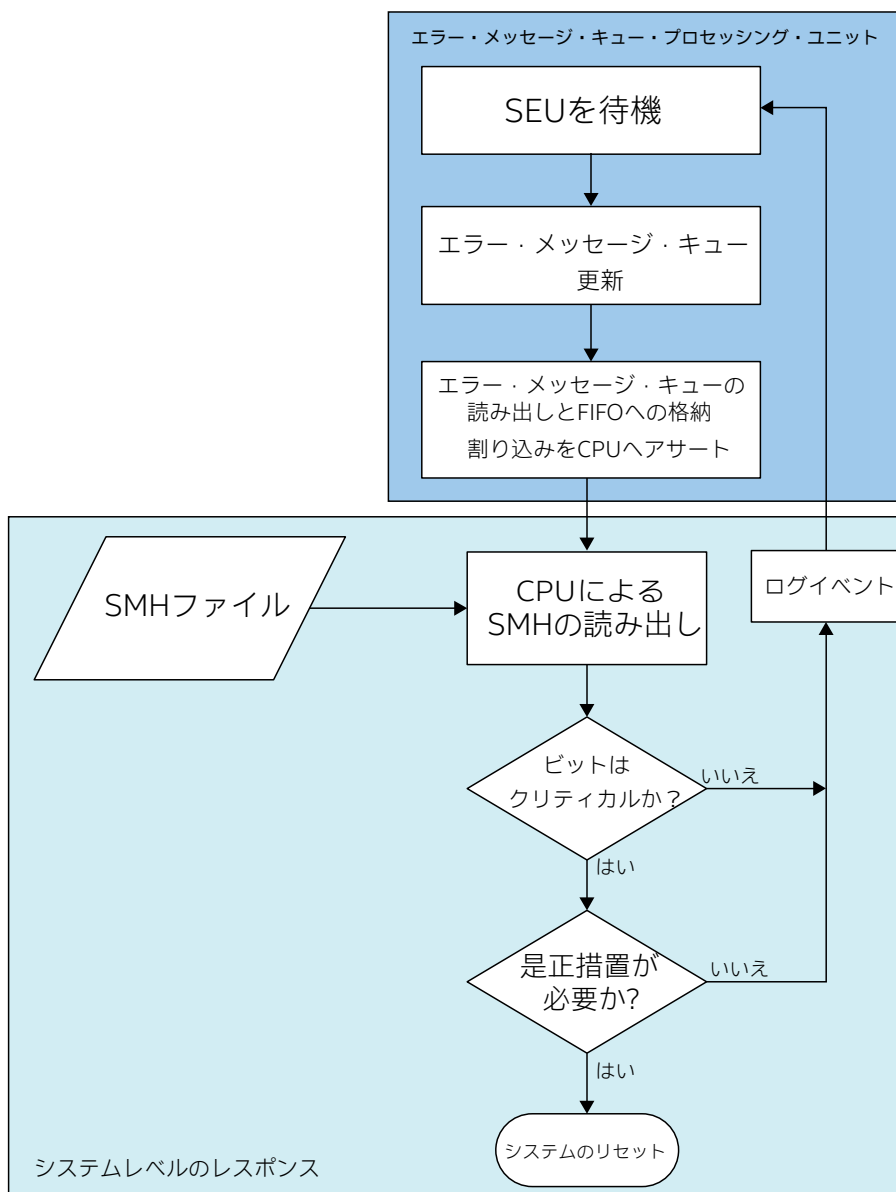
1. SEU エラーがあると、SEU_ERROR がアサートされます。
2. Advanced SEU Detection IP コアでは、SDM から SEU エラーメッセージを取得し、内部 FIFO に格納します。

注意: Advanced SEU Detection エラーメッセージの取得中にシステムでエラーが発生すると、IP コアでは `sys_error` 信号がアサートされます。

- Advanced SEU Detection IP コアでは、`seu_avst_valid` 信号をアサートして、エラーメッセージが有効であることを示します。
- 外部センシティブティビティ・プロセッサでは、Advanced SEU Detection IP コアの `seu_avst_valid` 信号を監視する必要があります。有効なエラーメッセージがある場合、プロセッサでは、Avalon-ST インターフェイスを介して SEU エラーの読み出しを開始し、センシティブティビティ・マップに対してロックアップを実行し、SEU エラーのクリティカルリティを判断します。

2.3.1.2.1. オフチップ・ロックアップ・センシティブティビティ・プロセッシングの動作フロー

図 -3: オフチップ・ロックアップ・センシティブティビティ・プロセッシングの動作フロー





2.3.1.3. SMH ルックアップ

.smh ファイルは、デザイン上の CRAM ビット設定のハッシュを表します。CRAM に関連するグループは、センシティブティアー・アレイ内のシグナルビットにマッピングされます。SEU イベント中、アプリケーションでは .smh に対してルックアップを実行し、ビットの使用を見極めます。ビット位置情報によって、動作システムの有効なソフトエラー率を低減することができます。

次の基準でデザイン内の CRAM 位置のクリティカリティが決定されます。

- 配線 - 使用された配線ラインを制御するすべてのビット
- アダプティブ・ロジック・モジュール (ALM) - ALM をコンフィグレーションすると、IP コアではその ALM センシティブに関連するすべての CRAM ビットを考慮します。
- ロジック・アレイ・ブロック (LAB) 制御ライン - LAB で ALM を使用すると、IP コアでは LAB センシティブに供給するコントロール信号に関連するすべてのビットを考慮します。
- M20K メモリーブロックとデジタル信号処理 (DSP) ブロック - ブロックを使用すると、IP コアではそのブロック・センシティブに関連するすべての CRAM ビットを考慮します。

関連情報

[Advanced SEU Detection IP コア \(9 ページ\)](#)

2.3.1.3.1. SMH リビジョン 4 ファイル・フォーマット

表 4. インテル Stratix 10 デバイスの SMH リビジョン 4 ファイル・フォーマット

ブロック	サブブロック	32 ビットワード	ビット	詳細
センシティブティアー・マップ・ヘッダー	—	0	[31:0]	SMH フォーマットとそのバージョンの識別ワード 0xXB445341。
		1	[31:8]	予約済み
			[7:0]	ASD 領域のビットマスクサイズ。ASD 領域のビットマスクサイズは、デザイン内の最大 ASD 領域 ID に対する 2 の上限電力です。1、2、4、8、16、または 32 にすることができます。
Sector Information ブロック	セクター 0 情報	0	[31:0]	セクター 0 エンコード方式のアドレス。
		1	[31:0]	エンコードされたセクター 0 センシティブティアー・データのアドレス。
		2	[23:8]	セクター 0 で使用される ASD 領域のビットマスク数 (つまり、SMH タグ数)。値 0 は、セクター内にセンシティブティアー・ビットがないことを示します。
	[7:0]		セクター 0 SMH タグサイズはビット単位です。1、2、4、8 にすることができます。	

セクター N 情報	N*3 .. N*3+2	
セクターエンコード化ブロック	セクターエンコード化 0	0	[31:16]	識別ワード 0xEEEE
			[15:0]	単一フレームエンコード化 (つまり、ビット->タグ・インデックス) のサイズをバイト単位でマッピングします。
		1	[31:0]	フレーム情報のアドレス (FADD)

continued...

ブロック	サブブロック	32 ビットワード	ビット	詳細
		2	[31:0]	フレーム情報のアドレス (EADD)
		FADD	[31:20]	フレーム 0 のエンコード化マップのインデックス
			[19:0]	センシティブィティデータは、フレーム 0 のセクター・センシティブィティデータにオフセットされます。
	
		FAAD+K	[31:20]	最後のフレームのエンコード化マップのインデックス
			[19:0]	センシティブィティデータは、最後のフレームのセクター・センシティブィティデータにオフセットされます。
		EADD		フレームエンコード化マップ 0。16 ビット「ビット・グループ・センシティブィティ・タグ・インデックス」へのフレーム内の「ビット位置」のフレーム・センシティブィティデータへのマッピングが含まれています。 感度データは必要ないので、フレーム内のすべてのファントムビットに対して、ビット・グループ・センシティブィティ・インデックス」は 0xFFFF に設定されます。
	
	
		セクターエンコード化 M		...
セクター・センシティブィティデータ	セクター 0 データ	0	[31:16]	セクターデータ識別ワード (0xDDDD)
			[15:0]	予約済み
		1..L		セクター領域マップ: L = ('ASD region bitmask size' * 'number of ASD region masks for sector'+31)/32
		L+1		エンコードされたフレームのセンシティブィティデータ。各フレームのデータは次の場所にあります。 offset = L+1+frame sensitivity data offset * sector SMH tag size
	
	セクター N データ		...	

2.4. デザイン階層のセンシティブィティの指定

インテル Quartus Prime 開発ソフトウェアでは、各ロジックブロックのクリティカルリティを示すため、パーティションを生成し、各パーティションにセンシティブィティ ID タグを割り当てます。インテル Quartus Prime 開発ソフトウェアでは、この情報を Sensitivity Map Header File (.smh) に格納します。

システム動作中にエラーが発生した場合、システムでは、エラーの影響を判断するために .smh ファイル内の分類を調べます。その後、システムでは分類に基づいて訂正措置を講じることができます。

注意: .smh ファイルの生成には、ライセンス版 インテル Quartus Prime 開発ソフトウェアが必要です。

.smh ファイルにアクセスするには、Advanced SEU Detection IP コアのインスタンスをデザインに追加してください。

2.4.1. 階層タグ付け

インテル Quartus Prime 階層タグ付け機能を使用すると、デザイン効率の高い FIT レートの向上のために、デバイス動作のクリティカルなロジックのみをタグ付けすることができます。

システム回復手順は、SEU によって障害を受けたロジックの知識に基づいて定義することもできます。この手法により、FPGA および FPGA が存在しているシステムのダウンタイムが短縮されます。階層タグ付けのその他の利点は次のとおりです。

- 重大性が低いエラーに対して破壊的な回復手順を回避することで、システムの安定性が向上します。
- 異なるデザインロジックに対する多様な訂正措置が可能になります。

.smh ファイルには、デザイン・センシティブ・ビット用のマスクが圧縮フォーマットで格納されています。インテル Quartus Prime 開発ソフトウェアでは、デザイン全体のセンシティブティーマスクを生成します。

2.5. 機能不調に対するシステムのレスポンスの評価

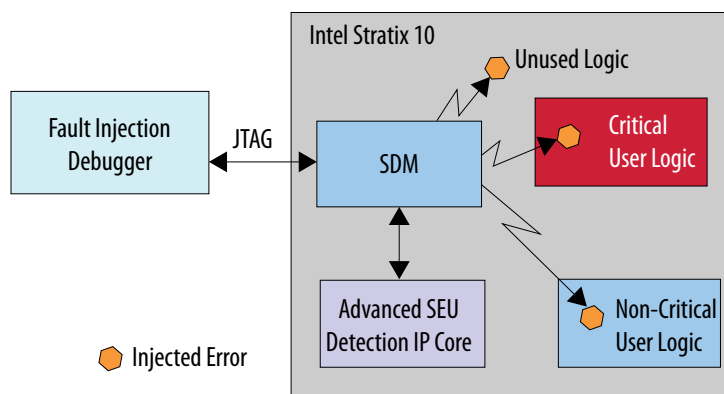
SEU では、あらゆるメモリーエレメントが検出される可能性があります。したがって、システムのテストを行って、包括的な回復レスポンスを確実なものにしてください。インテル Quartus Prime 開発ソフトウェアに含まれている Fault Injection Debugger では、SEU の回復を支援します。Fault Injection Debugger を GUI でグラフィカルに使用することも、コマンドライン・アサインメントを使用することもできます。

2.5.1. インテル Quartus Prime Fault Injection Debugger

シングル・イベント・アップセット (SEU) の検出およびデバッグに、インテル Quartus Prime 開発ソフトウェアの Fault Injection Debugger を使用します。デバッガーを使用して、Intel Stratix 10 FPGA デバイスのコンフィグレーション RAM (CRAM) にエラーを挿入します。

Fault Injection Debugger を使用すると、システム内の FPGA を操作してランダムな CRAM ビットフリップを挿入することができます。このようなシミュレートされた SEU 検出により、FPGA およびシステムによる SEU の検出および回復方法の観察が可能になります。結果に応じて、システムの回復順序の調整ができます。

図 -4: Intel Stratix 10 デバイスの Fault Injection Debugger の概要ブロック図





Fault Injection Debugger を使用すると、次の事項が実行できます。

- 次のいずれかへのシングル・ビット・エラーの挿入
 - ランダムな位置
 - 指定された領域
- エラー・メッセージ・キューの読み出しによるエラー情報のレポート

注意: Fault Injection Debugger ツールを使用するためには、ライセンス版 インテル Quartus Prime 開発ソフトウェアが必要です。

関連情報

- [Fault Injection Debugger の使用 \(22 ページ\)](#)
- [Fault Injection Debugger インターフェイス・パラメーター \(28 ページ\)](#)
- [Fault Injection Debugger のコマンドライン・インターフェイス \(28 ページ\)](#)
- [エラーの挿入 \(23 ページ\)](#)

3. Intel Stratix 10 SEU 緩和実装ガイド

3.1. SEU_ERROR ピンの設定

インテル Quartus Prime 開発ソフトウェアの SEU_ERROR ピン機能を設定するには、次の手順を実行します。

1. **Assignments** メニューで **Device** をクリックします。
2. **Device and Pin Options** で Configuration category を選択し、**Configuration Pins Options** をクリックします。
3. **Configuration Pin** 画面で **USE SEU_ERROR output** をオンにします。
4. 未使用の SDM ピンをドロップダウンから選択し、SEU_ERROR ピン機能を実装します。
5. **OK** をクリックし、**Configuration Pin** ウィンドウを閉じます。

3.2. インテル Quartus Prime SEU ソフトウェアの設定

Device and Pin Options ウィンドウの **Error Detection CRC** 設定を使用して、最小 SEU 間隔を設定し、エラー検出および内部スクラビング機能を有効にし、.smh ファイルの生成を有効にします。

1. インテル Quartus Prime メニューで **Assignments > Device** をクリックします。
2. **Device** ウィンドウで **Device and Pin Options** をクリックします。
3. **Device and Pin Options** ウィンドウで、**Error Detection CRC** カテゴリーを選択し、次の設定を指定します。

設定	詳細
Enable error detection check	オンにするとエラー検出機能が有効になります。このオプションが必要なのは、センシティブ処理やフォールト・インジェクション、あるいは SEU_ERROR ピンの動作を観察する場合です。
Minimum SEU interval	0 から 10000 ミリ秒の値を指定して、同じビットの 2 つの検査間の最小時間を設定します。できるだけ頻繁に確認するには 0 を指定します。
Enable internal scrubbing	オンにするとエラー訂正機能が有効になります。このオプションはセンシティブ処理に必要です。
Generate SEU sensitivity map file (.smh)	オンにすると .smh が生成されます。このオプションはセンシティブ処理に必要です。

4. **OK** をクリックします。

関連情報

- [SEU_ERROR ピンの動作 \(8 ページ\)](#)
- [Advanced SEU Detection IP コア \(9 ページ\)](#)
- [階層タグ付けの実行 \(18 ページ\)](#)
- [Fault Injection Debugger の使用 \(22 ページ\)](#)

3.3. プライオリティー・スクラビングの有効化

インテル Quartus Prime 領域およびデザイン・パーティション機能を使用して、プライオリティーの高い内部スクラビングのエリアを指定します。

1. インテル Quartus Prime 開発ソフトウェアで **Assignments > Logic Lock Regions Window** を選択します。
2. **Logic Lock Regions Window** で領域を作成し、それをデザイン・パーティション内に配置します。
3. クリティカルなデザインモジュール、エンティティー、またはロジックグループを追加し、それらを保存してリージョンにロックします。
4. インテル Quartus Prime 開発ソフトウェアで **Assignments > Assignment Editor** を選択します。
5. **Assignment Editor** で、**Priority SEU Area** を Logic Lock 領域を配置するデザイン・パーティションに対して割り当てます。

Assignment Editor を使用する代わりに、プロジェクトの Quartus Settings File (.qsf) に次の命令を含めることもできます。`set_instance_assignment -name PRIORITY_SEU_AREA ON -to <partition name>`

インテル Quartus Prime 開発ソフトウェアでは、プライオリティー・セクターの内部スクラビング・スケジュールを「できるだけ早く」に設定します。他のセクターの内部スクラビング・スケジュールは、プロジェクトの **Minimum SEU interval** グローバル・アサインメントに従います。

関連情報

[Logic Lock Regions, Design Optimization User Guide: インテル Quartus Prime プロ・エディション](#)

Logic Lock 領域の作成と使用に関する詳細情報を提供しています。

3.4. 階層タグ付けの実行

タグ付け用の FPGA 領域を定義するために、ASD Region の場所を割り当てます。デザイン階層の任意の部分に ASD Region 値を指定するには、Design Partitions Window を使用します。

1. インテル Quartus Prime ソフトウェアで **Assignments > Design Partitions Window** を選択します。
2. ヘッダー行の任意の場所を右クリックし、ASD Region をオンにして **ASD Region** 列を表示します (表示されていない場合)。
3. ロジック・センシビティ ID 値を任意のパーティションに対して 0~32 の範囲で入力し、特定の ASD Region に割り当てます。

Logic Sensitivity ID は、パーティションに関連付けられたセンシビティ・タグを表します。

- センシビティ・タグ 1 は、割り当てなしと同じで、基本的なセンシビティ・レベルを示します。これは、「デザインで使用される領域」です。
- センシビティ・タグ 0 は、予約済みで未使用の CRAM ビットを示します。明示的にパーティションを 0 に設定して、そのパーティションがクリティカルではないことを示すことができます。この設定では、パーティションがセンシビティ・マッピングから除外されます。

注意: 複数のデザイン・パーティションに同じセンシビティ・タグを使用できます。



図 -5: Design Partitions ウィンドウの ASD Region 列

Partition Name	Hierarchy Path	Type	Preservation Level	Empty	Color	ASD Region
<<new>>						
state_m	inst1	Reconfigurable	Not Set	No	Red	0
taps		Default	synthesized	No	Yellow	3
hsvlues	inst2	Periphery Reuse Core	Final	No	Green	2

デザインをコンパイルすると、インテル Quartus Prime 開発ソフトウェアでは、センシティブティー・データが標準の Intel hex (ビッグ・エンディアン) .smh ファイルとして生成されます。これは .sof ファイルの生成時に行われます。

関連情報

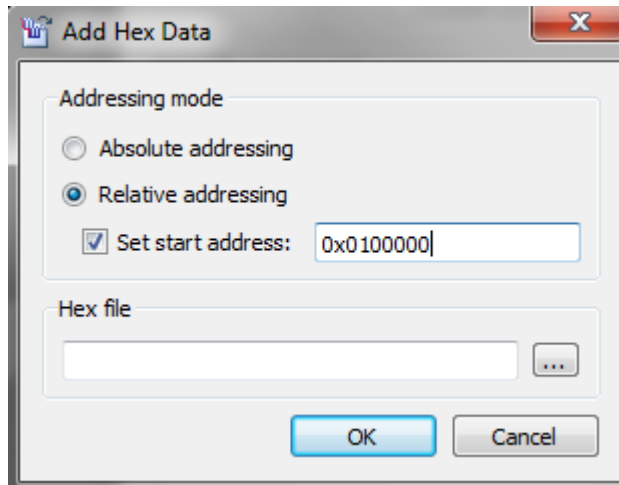
- [Advanced SEU Detection IP コア \(9 ページ\)](#)
- [インテル Quartus Prime SEU ソフトウェアの設定 \(17 ページ\)](#)

3.5. センシティブティー・マップ・ヘッダー・ファイルのメモリーへのプログラミング

.smh のプログラムは、どのタイプのメモリーに対しても行うことができます。例えば、CFI フラッシュメモリーを使用するには次の手順を行います。

1. .smh の名前を <file_name>.hex に変更、または、必要に応じてリトル・エンディアンの <file_name>.hex に変換します。
2. インテル Quartus Prime 開発ソフトウェアで **File > Convert Programming Files** をクリックします。
3. **Convert Programming Files** ウィンドウの **Output programming file** で目的のオプションを選択します。
4. 16 進データを追加するには、次の手順に従います。
 - a. **Add Hex Data** をクリックします。
 - b. **Add Hex Data** ダイアログボックスで **Set start address** をオンにし、開始アドレスを入力します。
 - c. **Hex file box** でブラウザをクリックして .hex ファイルを選択し、**OK** をクリックします。

図 -6: Add Hex Data ダイアログボックスの追加



5. **Generate** をクリックします。

3.6. センシティブィティーマップヘッダーのルックアップの実行

次のオプションを インテル Quartus Prime 開発ソフトウェアで有効にしてから、SMH ルックアップの実行に Advanced SEU Detection Intel Stratix 10 FPGA IP を使用してください。

- エラー検出 CRC
- SEU センシティブィティーマップファイル (.smh) の生成

センシティブィティーマップファイルのルックアップを インテル Stratix 10 デバイスに対して実行する場合は、次の手順を実行します。

1. .smh ファイルヘッダーを読み出して、汎用 .smh 情報を取得します。
 - アドレス = 0
 - ワード 0 = SMH_signature
 - ワード 1 = (reserved, region_mask_size)
 - ワード 2 = sector_info_base_address
2. 32 ビットワードのセクター情報エントリーを 3 つ読み取ります。
 - a. セクター暗号化スキーム 32 ビットアドレス
 - b. セクター .smh データ 32 ビットアドレス
 - c. 8 ビットのセクター .smh タグサイズ(1、2、4、または 8 ビットにすることができます)
 - d. セクターで使用される ASD 領域ビットマスクの数である 16 ビットの ASD 領域マップサイズ



- アドレス = `sector_info_base_address + (sector_index*3)`
 - ワード 0 = `encoding_scheme_address`
 - ワード 1 = `sector_data_address`
 - ワード 2 = `(reserved, regions_map_size, smh_tag_size)`
3. 次のセクター暗号化スキーム情報を読み出して、エラー位置フレーム・インデックスおよびフレーム内のビット位置を確認します。
- a. セクター暗号化スキームのヘッダー情報の最初の 3 つのワードを読み出して、暗号化スキームのパラメーターを取得します。
 - アドレス = `encoding_scheme_address`
 - ワード 0 = `(reserved, frame_encoding_map_size)`
 - ワード 1 = `frame_info_base_offset`
 - ワード 2 = `frame_encoding_base_offset`
 - b. 32 ビットのフレーム情報文字列でフレーム番号を確認します。
 - アドレス = `encoding_scheme_address + frame_info_base_offset + frame_index`
 - ワード 0 = `(frame_encoding_index, frame_data_offset)`
 - c. ビット位置のフレーム・センシティビティ・データへの 16 ビット・インデックスを取得します。

```
int16* frame_encoding_map = encoding_scheme_address +
frame_encoding_base_offset + (frame_encoding_map_size *
frame_encoding_index)/4;

int16 tag_index = frame_encoding_map[bit_position];
```
4. セクター .smh データで次のデータを確認し、影響を受ける ASD 領域を確立します。
- a. `frame_data_offset` および `tag_index` の `smh_tag_size` ビット長 .smh タグ 2 (20 ページ)。

```
int8* frame_data = (sector_data_address + 1 +
(regions_map_size*region_mask_size+31)/32 +
frame_data_offset*smh_tag_size);

int8 sensitivity_byte =
frame_data[tag_index*smh_tag_size/8];

int8 smh_tag = (sensitivity_byte >> (tag_index*smh_tag_size
%8)) & ((0x1<<smh_tag_size)-1);
```
 - b. ゼロの SMH タグは、ビットエラーの位置がどの領域に対してもクリティカルではないことを示します。ゼロ以外のタグは、領域マップ内のインデックスを示します。SMH タグの領域マスクを取得するには次を実行します。

```
int32* region_masks = sector_data_address+1;
int32 region_mask_offset = (smh_tag-1)*region_mask_size;
int32 region_mask_word = region_masks[region_mask_offset/
32];

int32 region_mask = (region_mask_word >> region_mask_offset
%32) & ((0x1<<(region_mask_size)-1);
```

関連情報

[Advanced SEU Detection IP コア \(9 ページ\)](#)

3.7. Fault Injection Debugger の使用

フォールト・インジェクション機能を有効にするには、デザインで **Enable error detection CRC** オプションを有効にしてください。

Fault Injection Debugger を起動するには、インテル Quartus Prime 開発ソフトウェアで **Tools > Fault Injection Debugger** を選択します。

Fault Injection Debugger を使用するには、ツールからデバイスへの接続に JTAG インターフェイスを利用します。次に、JTAG チェーンのコンフィグレーションを行います。JTAG チェーンのコンフィグレーションには、次の手順を実行してください。

1. **Hardware Setup** をクリックします。このツールでは、コンピューターに接続されているプログラミング・ハードウェアが表示されます。
2. 使用するプログラミング・ハードウェアを選択します。
3. **Close** をクリックします。
4. **Auto Detect** をクリックすると、デバイスチェーンに JTAG チェーン内のプログラマブル・デバイスが追加されます。

関連情報

- [インテル Quartus Prime Fault Injection Debugger \(15 ページ\)](#)
- [エラー・メッセージ・キュー \(7 ページ\)](#)
- [インテル Quartus Prime SEU ソフトウェアの設定 \(17 ページ\)](#)

3.7.1. デバイスと Fault Injection Debugger のコンフィグレーション

Fault Injection Debugger では、Software Object File (.sof) を使用します。

.sof を指定するには次を実行します。

1. コンフィグレーションを行う Intel Stratix 10 デバイスを **Device chain** ボックスで選択します。
2. **Select File** をクリックします。
3. .sof に移動して **OK** をクリックします。Fault Injection Debugger で **.sof** を読み出します。
4. **Program / Configure** をオンにします。
5. **Start** をクリックします。

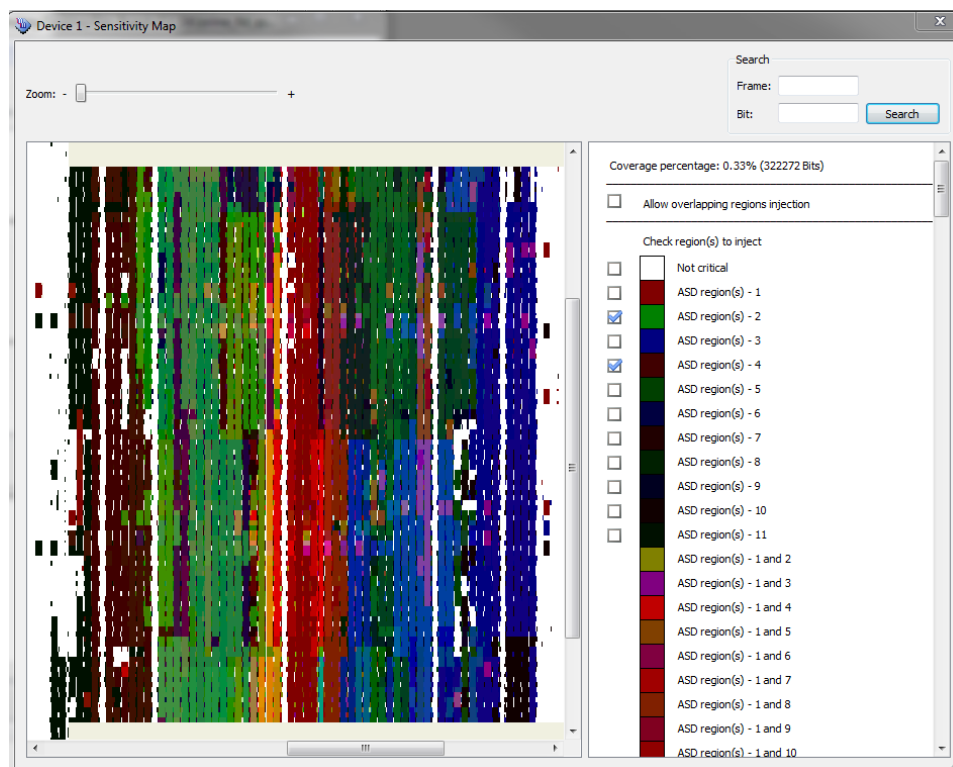
3.7.2. フォールト・インジェクションの領域制約

SMH ファイルのロード後、Fault Injection Debugger に指示して、特定の ASD 領域のみで操作するようにします。

フォールト・インジェクションを行う ASD 領域を指定するには、次を実行します。

1. **Device chain** ボックスで FPGA を右クリックし、**Show Device Sensitivity Map** をクリックします。
2. フォールト・インジェクション用の ASD 領域を選択します。

図 -7: Device Sensitivity Map ビューア



3.7.3. エラーの挿入

エラーの挿入は、次の方法で行うことができます。

- ランダムな位置へのエラー挿入には、Fault Injection Debugger のオプションを使用します。
- 特定の位置へのエラー挿入には、コマンドライン・インターフェイスを使用します。

関連情報

[インテル Quartus Prime Fault Injection Debugger \(15 ページ\)](#)

3.7.3.1. ランダムな位置へのエラー挿入

ランダムな位置へのエラー挿入には、Fault Injection Debugger のオプションを使用して、次の手順を実行します。

1. **Inject Fault** オプションをオンにします。
2. エラー挿入を繰り返し実行するか、停止するまで実行するかを選択します。
 - 停止するまで実行することを選択した場合、Fault Injection Debugger でのエラー挿入の間隔は、**Tools > Options** ダイアログボックスで指定した通りになります。

— エラー挿入を特定の回数だけ反復実行する場合は、回数を入力します。

3. **Start** をクリックします。

インテル Quartus Prime メッセージ・ウィンドウでは、挿入されたエラーに関するメッセージが表示されます。挿入されたエラーの詳細については、**Read EMR** をクリックしてください。Fault Injection Debugger では、エラー・メッセージ・キューを読み出し、その内容を Messages ウィンドウに表示します。

注意: **Read EMR** で取得できるのは、エラー・メッセージ・キューの内容です。

3.7.3.2. 特定の位置へのエラー挿入

次の引数を使用し、特定の位置へのエラー挿入をコマンドライン・インターフェイスを使用して行います。

```
quartus_fid -cable=<cable_num> --index= "@<device_num>=<sof_file>" --user  
"@<device_num>=<sector_location> <frame_location> <bit_location>"
```

コマンドライン・インターフェイスの引数について詳しくは、[Fault Injection Debugger のコマンドライン・インターフェイス](#) (28 ページ) を参照してください。

4. Advanced SEU Detection Intel Stratix 10 FPGA IP 参照事項

Advanced SEU Detection IP コアにさまざまなパラメーターを設定し、動作、ポート、および信号をカスタマイズすることができます。

インテル Quartus Prime 開発ソフトウェアでは、カスタマイズした Advanced SEU Detection IP コアの生成を、パラメーター・エディターで設定したパラメーター・オプションに基づいて行います。

4.1. Advanced SEU Detection IP コアのパラメーター設定

表 5. Advanced SEU Detection IP コアのパラメーター設定

パラメーター	値	デフォルト値	詳細
Use on-chip sensitivity processing	<ul style="list-style-type: none"> • On • Off 	On	外部メモリー・インターフェイスの使用を選択し、センシティブデータにアクセスして IP による SEU 位置ルックアップを実行します。
Largest ASD region ID used	1 ~ 32	1	デザインで使用する最大の ASD 領域 ID を指定します。使用可能なのは Use on-chip sensitivity processing がオンの場合です。デザインで使用できる領域 ID 分類の最大数は 16 です。 ⁽³⁾
Sensitivity data start address	0x0	0x0	定数オフセットが指定され、外部メモリー・インターフェイスによって生成されたすべてのアドレスに追加されます。使用可能なのは Use on-chip sensitivity processing がオンの場合です。
Show raw SEU error message	<ul style="list-style-type: none"> • On • Off 	Off	選択すると raw SEU エラーメッセージが表示されます。使用可能なのは Use on-chip sensitivity processing がオンの場合です。
SEU error fifo depth	<ul style="list-style-type: none"> • 2 • 4 • 8 • 16 • 32 • 64 	4	格納する SEU エラー数を指定します。使用可能なのは Use on-chip sensitivity processing がオンの場合です。
Use with Fault Injection Debugger Tool	<ul style="list-style-type: none"> • On • Off 	Off	オンにすると Fault Injection Debugger Tool と併用できます。

関連情報

[Advanced SEU Detection IP コア \(9 ページ\)](#)

⁽³⁾ 使用中の領域 ID 数は、SMH で指定された領域マスクによって制限されます。

4.2. Advanced SEU Detection IP コアのポート

図 -8: Advanced SEU Detection IP コアのオンチップ・センシティビティ・プロセッシング図

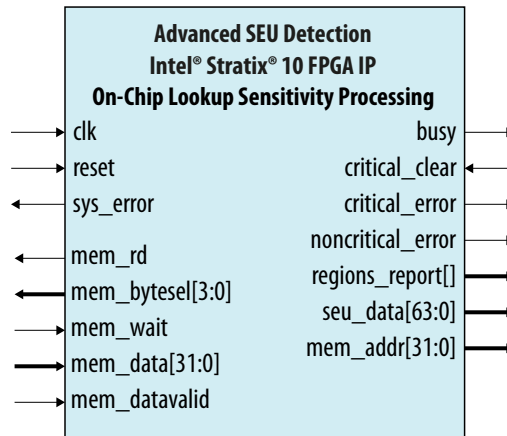


表 6. Advanced SEU Detection IP コアのオンチップ・センシティビティ・ポート

ポート	幅	入力/出力	詳細
clk	1	入力	ユーザー入力クロック。最大周波数は 250MHz です。
reset	1	入力	アクティブ High のシンクロナス・リセット信号。
busy	1	出力	ロジック High によって、Advanced SEU Detection IP コアが SEU データ処理でビジー状態にあることが示されます。信号が Low になるのは、処理が完了して、critical_error または noncritical_error 信号がアサートされたときです。
sys_error	1	出力	ロジック High によって、SEU エラーの取得中にシステムエラーがあったことが示されます。
critical_clear	1	入力	High にアサートして、最後に処理した SEU データ入力のエラーレポート (critical_error、noncritical_error、regions_report、および seu_data) を消去します。
critical_error	1	出力	ロジック High によって、SEU エラーがクリティカル領域にあると SMH ルックアップによって判断されたことが示されます。
noncritical_error	1	出力	ロジック High によって、SEU エラーが非クリティカル領域にあると SMH ルックアップによって判断されたことが示されます。
regions_report	1~32	出力	領域 ID によって、SMH ルックアップでレポートされたエラーを示します。この信号のポート幅は、 Largest ASD region ID used パラメーターで設定されます。
seu_data	64	出力	表示される SEU エラーメッセージは、最後に処理された SEU データ入力に関するものです。ポートが使用可能なのは、 Show raw SEU error message パラメーターがオンの場合です。エラーメッセージの詳細については、 <i>Error Message Queue</i> を参照してください。
mem_addr	32	出力	バイト・アドレッシング単位の Avalon-MM アドレスバス。
mem_rd	1	出力	Avalon-MM 読み出しコントロール信号。
mem_wait	1	入力	Avalon-MM waitrequest 信号。
mem_data	32	入力	Avalon-MM データバス。
mem_datavalid	1	入力	Avalon-MM データ有効信号。



図 -9: Advanced SEU Detection IP コアのオフチップ・センシティビティ・プロセッシング・ブロック

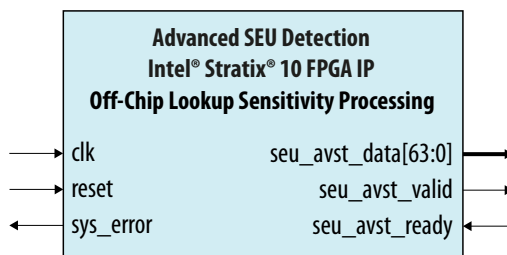


表 7. Advanced SEU Detection IP コアのオフチップ・センシティビティ・ポート

ポート	幅	入力/出力	詳細
clk	1	入力	ユーザー入力クロック。最大周波数は 250MHz です。
reset	1	入力	アクティブ High のシンクロナス・リセット信号。
sys_error	1	出力	ロジック High によって、SEU エラーの取得中にシステムエラーがあったことが示されます。
seu_avst_data	64	出力	Avalon-ST データ信号。FIFO エントリからの SEU エラーメッセージを提供します。
seu_avst_valid	1	出力	Avalon-ST データ有効信号。seu_avst_data 信号には有効データが含まれています。
seu_avst_ready	1	入力	Avalon-ST ready 信号。

関連情報

- [Advanced SEU Detection IP コア \(9 ページ\)](#)
- [エラー・メッセージ・キュー \(7 ページ\)](#)

5. Intel Stratix 10 Fault Injection Debugger 参照事項

5.1. Fault Injection Debugger インターフェイス・パラメーター

パラメーター	詳細
Hardware Setup	Hardware Setup ウィンドウを開きます。
Start	プログラムを開始するかデバイスのコンフィグレーションを行います。
Auto Detect	指定したハードウェアの JTAG チェーンをスキャンして、デバイスチェーンをグラフィック表示します。
Select File	.sof ファイルを選択します。
Program/Configure	Programmer バックエンド・エンジンを呼び出して、デバイスのプログラムまたはコンフィグレーションを行います。
Inject Fault	フォールトを挿入します(ランダムな場所のみ)。
Run For	ツールがエラー挿入を停止するまでの、フォールト・インジェクションの反復回数を設定します。
Run until stopped	Stop をクリックするまで、ツールではフォールト・インジェクションを続行します。
Start	フォールト・インジェクションを開始します。
Stop	フォールト・インジェクションを停止します。
Read EMR	エラー・メッセージ・キューを読み出します。

関連情報

インテル Quartus Prime Fault Injection Debugger (15 ページ)

5.2. Fault Injection Debugger のコマンドライン・インターフェイス

コマンドラインで Fault Injection Debugger を実行するには、quartus_fid 実行可能ファイルを使用します。これは、フォールト・インジェクションをスクリプトから実行する場合に便利です。

表 8. Intel Stratix 10 デバイス用 Fault Injection Debugger のコマンドライン・インターフェイス 引数

短い引数	長い引数	詳細
l	list	インストールされているすべてのハードウェアを表示します。
c	cable	ケーブル番号を選択します。
a	auto	自動検出操作。この操作にはケーブルを 1 本だけ選択してください。
i	index	アクティブなデバイスまたはソフトエラーを挿入するデバイスを指定するオプション。 完全な構文: @<device_position>=<file_path>#<operation>

continued...

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2015
登録済



短い引数	長い引数	詳細
		場所: <ul style="list-style-type: none"> device_position は、TDI の直近から数えたアクティブデバイスの位置です。 file_path は、アクティブデバイスのプログラミング・ファイルです。 operation は実行する操作です。⁽⁴⁾ <ul style="list-style-type: none"> P - Program/Configure I - Inject fault
s	smh	センシティブティー・マップのヘッダーファイルを指定するオプション。 完全な構文: @<device_position>=<file_path>#<region_info> 場所: <ul style="list-style-type: none"> device_position は、TDI の直近から数えたアクティブデバイスの位置です。 file_path は、アクティブデバイスの SMH ファイルです。 region_info は意図した SMH 領域情報です。形式は次の通りです。 <targeted_regions><allow non critical><allow overlapping> <ul style="list-style-type: none"> targeted_regions = 領域のバイナリ表現 <ul style="list-style-type: none"> 領域 1 = 1 領域 2 = 2 領域 3 = 4 領域 4 = 8 領域 1 および 2 = 3 (1 + 2 から) 領域 1 および 3 = 5 (1 + 4 から) allow non critical - N = 非クリティカル・ビットへの挿入を許可 allow overlapping - O = 重複する領域を含むビットへの挿入を許可 例: <ul style="list-style-type: none"> リージョン 1 または 3 のみへの挿入 region_info = 5 リージョン 2 または非クリティカル・ビットへの挿入 region_info = 4N リージョン 4 または非クリティカル・ビットを持つ任意のビットへの挿入 region_info = 8NO
u	user	ユーザー固有の障害を指定するオプション。 完全な構文: @<device_position>=<sector-frame-bit-pair ?>#1 <sector-frame-bit-pair ?>#2 ... <sector-frame-bit?>#n 場所: <ul style="list-style-type: none"> device_position TDI の直近から数えたアクティブデバイスの位置です。 sector-frame-bit-pair エラーが挿入されたフレームビットとセクター位置です。⁽⁵⁾
n	number	挿入するソフトエラーの数を指定するオプション。
t	time	挿入の間隔時間を指定するオプション。

関連情報

[インテル Quartus Prime Fault Injection Debugger \(15 ページ\)](#)

⁽⁴⁾ 操作の指定がない場合は、インジェクション・フォールトがデフォルトの操作になります。

⁽⁵⁾ フレームビットの最大ペアは引数 n によって異なります。



6. Intel Stratix 10 SEU 緩和 ユーザーガイド・アーカイブ

IP コアのバージョンが一覧にない場合は、以前の IP コアバージョン用のユーザーガイドが適用されます。

IP コアバージョン	User Guide
18.0	Intel Stratix 10 SEU Mitigation User Guide

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/ またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

**ISO
9001:2015**
登録済

7. インテル Stratix 10 SEU 緩和 ユーザーガイドの改訂履歴

ドキュメントバージョン	インテル Quartus Prime バージョン	変更内容
2018.10 10	18.1	<ul style="list-style-type: none"> 二重隣接エラーの訂正サポートを削除しました。 エラー・メッセージ・キューに関するトピックを更新して、エラー・メッセージ・キュー内のエラー数の記述を削除しました。 内部スクラビングに関するトピックを追加しました。 プライオリティー・スクラビングに関するトピックを追加しました。 インテル Quartus Prime SEU 設定に関するトピックを更新して内容をより明確にしました。 プライオリティー・スクラビング有効化の手順を追加しました。 Largest ASD region ID used パラメーターの許容値を「1~255」から「1~32」に更新しました。
2018.08 07	18.0	<ul style="list-style-type: none"> 複数ビットエラーの修正サポートを削除しました。 オフチップ・ルックアップ・センシティビティー・プロセッシングに関するトピックの信号名を訂正して、<code>seu_avst_ready</code> から <code>seu_avst_valid</code> に変更しました。 IP コア名を "Intel FPGA Stratix 10 Advanced SEU Detection IP" から "Advanced SEU Detection Intel FPGA Stratix 10 IP" に更新しました。
2018.05 07	18.0	<ul style="list-style-type: none"> <code>smh</code> 引数を Fault Injection コマンドライン・インターフェイス・コマンドに追加しました。 Fault Injection のコマンドライン・インターフェイスの説明にある <code>user</code> コマンドを更新しました。 <i>Failure Rates</i> の項を追加しました。 <i>Constraining Regions for Fault Injection</i> の項を追加しました。 特定の場所にエラーを挿入するための引数を更新しました。 メモリーブロックの誤り訂正コードのサポートトピック内の eSRAM ブロックの ECC ステータスフラグ信号を更新しました。

日付	バージョン	変更内容
2017 年 12 月	2017.12 29	<ul style="list-style-type: none"> Fault Injection ツールの情報を追加しました。 Advanced SEU Detection IP コアを追加しました。 実装ガイドを更新して、Fault Injection ツールおよび Advanced SEU Detection IP コアの実装を含めました。 概要の章を改編しました。
2016 年 12 月	2016.12.09	<ul style="list-style-type: none"> <i>SEU_ERROR Pin Settings</i> を追記。 Added <i>Enable internal scrubbing</i>.
2016 年 10 月	2016.10.31	初版

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。