XAPP1098 (v1.2) 2017 年 2 月 22 日

UltraScale FPGA および UltraScale+ FPGA での不正操作防止デザインの開発

著者: Ed Peterson

概要

このアプリケーション ノートでは、UltraScale™ および UltraScale+™ FPGA で実現した、システム内に存在する知的所有権 (IP) や機密データを保護する不正操作防止 (AT) の機能について説明し、それらの実例を紹介します。この保護機能 (不正操作防止機能) は、FPGA がビットストリームでコンフィギュレーションされる間およびその前後に有効になっている必要があります。機密データには、FPGA ロジックの機能を設定するコンフィギュレーション データ、ビットストリームに含まれる重要なデータ/パラメーター (ブロック RAM の初期値やフリップフロップの初期状態など) が含まれます。また、コンフィギュレーション後の通常動作時に FPGA に対して動的に書き込みまたは読み出しが実行される外部データも機密データとなります。

ここでは、UltraScale および UltraScale+ FPGA で利用できるシリコン AT 機能の概要を説明し、これらの機能が必要な理由、各機能の使用例、およびインプリメンテーションの詳細を示します。また、さらなる不正操作防止を実現するための、その他の方法についても説明します。

このアプリケーション ノートを参考にすることで、UltraScale および UltraScale+ FPGA で利用できる AT ベスト プラクティスを確実に実施できます。これらのベスト プラクティスは、商用デザインのクローニングやオーバービルドを防止する目的でも、軍事システムの重要なクリティカル テクノロジ (CT) のリバース エンジニアリングを防止する目的でも、またはその中間的な目的でも広く適用できます。

このアプリケーション ノートでは、ザイリンクス FPGA のアーキテクチャ [参照 1] とデザインに関してある程度の知識があり、Vivado® ツールを使用したフロー設計手法 [参照 2] とコンフィギュレーション [参照 3] の経験があることを前提としています。『デザインの安全性の確保』[参照 4] および『Virtex-6 および 7 シリーズ FPGA での不正操作防止デザインの開発』[参照 5] でも、さまざまなセキュリティ上の脅威と FPGA でのソリューションについて説明しています。

はじめに

ザイリンクスは FPGA AT ソリューションにおいて何世代にもわたって業界を牽引してきました。UltraScale および UltraScale+ FPGA では、非対称認証、サイドチャネル攻撃からの保護などのシリコン AT 機能により最先端の保護を実現できます。また、コンフィギュレーション後のさまざまな不正操作防止機能を提供するために、Security Monitor [参照 6] という IP コアを提供しています。一部制約のため、Security Monitor の使用には制限があります。詳細は、ザイリンクス販売代理店へお問い合わせください。

不正防止において攻撃者よりも常に一歩リードするということは、継続的なプロセスであり、既存の脆弱性と攻撃を把握して攻撃に対抗するために新しい軽減手法 (対抗措置) を開発する作業が伴います。ザイリンクスは、商用市場と防衛市場の両方を含めた AT 機能を重視するユーザーに対して、数世代にわたってセキュアな FPGA テクノロジを低コストで提供しています。

ザイリンクス FPGA のさまざまな AT 機能を利用することで、プログラムや顧客要件に基づいてどの程度の AT を FPGA デザインに含めるかを選択できます。個々のシリコン AT 機能を有効にして使用することも、各 AT 機能を組み合わせて使用する (FPGA デザイン内で AT 機能を組み合わせて、ベスト プラクティスのガイダンスに従うなど) ことも可能です。

どの程度の AT を含めるかの判断は、主に次の 3 つの要素を考慮します。

- 価値: 知的所有権の知覚価値、および知的所有権が侵害された場合に経済的または国家安全保障に与える損害。特定の AT 機能は実装に高いコストがかかるため、保護対象のテクノロジやデータの価値に対するコストを慎重に検討する必要があります。
- 攻撃者: システムへのアクセス手段、および攻撃に伴う専門レベルとリソース。たとえば、システムへのアクセスは「銃、門、警備員」で防ぐことはできるか、一般市場で簡単に入手できるか、攻撃者は個人レベルのハッカーなのか、大規模なグループなのか、など。攻撃者の能力は極端なものから中間レベルに至るまでさまざまです。

- 設計段階: システム開発サイクルのどの段階で FPGA デザインに AT を使用すると決定するか。ザイリンクスは、スケジュールとコスト両方の要件を満たすことができるように、FPGA AT 機能の使用に関する決定は可能な限り早期 (システムに CT を定義した後など) に行うことを強く推奨しています。開発プロセスの後の段階で AT 機能を挿入すると必ずコストと時間が増加し、多くの場合効果も低減します。

また、特定の能動的 AT 機能を有効にした場合に消費する FPGA のロジック リソースも考慮する必要があります。通常、全体的なリソース消費による影響はかなり抑えられますが、機能の実装方法や FPGA のサイズによって異なり、大規模な FPGA ほど影響は小さくなります。

ザイリンクスでは、シリコンの AT 機能を能動的セキュリティと受動的セキュリティに分類しています。概して、受動的セキュリティ機能はツール フローの一部であるか FPGA に組み込まれているため、FPGA ロジック デザインに対して追加の作業は必要ありません。また、受動的セキュリティ機能は一時的という特性を持ち、FPGA の通常の動作サイクルの異なるタイミングで次のような機能を果たします。

- コンフィギュレーション前 (例: コンフィギュレーション ビットストリームの公開キー認証)
- コンフィギュレーション中 (例: 差分電力解析 (DPA) によるサイドチャネル攻撃の防止)
- コンフィギュレーション後 (例: リードバックの無効化によるユーザー データの保護)

これに対して、能動的セキュリティ機能は FPGA ロジック デザインに含める必要があります。これらの機能は、ユーザー ビットストリームで FPGA がコンフィギュレーションされ、デザインがアクティブになった後にのみ有効になります。例として、KEYCLEARB をアサートしてバックアップ バッテリ付きの AES (Advanced Encryption Standard) キーをゼロにする場合や、PROGRAM_B インターセプトを処理する場合が挙げられます。

最小限の措置として、ビットストリームの暗号化や認証などの適切な受動的セキュリティ機能はデザインに含めるように常に計画する必要があります。これらの機能はデザインの機能性に影響を与えることはありませんが、ロジスティック上の課題 (例: キー管理)、シスム上の課題 (例: キー ストレージ用にバックアップ バッテリ付き RAM (BBRAM) を使用する場合はバッテリが必要)、およびコンフィギュレーション時間の増加 (例: 公開キー認証によりコンフィギュレーション時間が増加) などが生じる可能性があります。それ以外では、これらの機能を無償で利用して、相当の改ざん防止策を講じることができます。これらの AT 機能は、実際の FPGA ロジック デザインに影響を与えないため、既にフィールド展開されたシステムや開発段階後期のデザインで使用を検討するに十分に値します。

このアプリケーション ノートで説明している AT 機能とガイダンスは、主に 3 つの AT カテゴリに分類されます。

- 防止 (例: JTAG ポートのブロック)
- 検出 (例: 電圧および温度の監視)
- 応答 (例: BBRAM キー消去ペナルティ)

表 1 に、UltraScale および UltraScale+ FPGA のビルトイン シリコン AT 機能の概要とそのカテゴリを示します。

表 1: AT 機能の分類と概要

UltraScale および UltraScale+ FPGA のシリコン AT 機能	種類	カテゴリ	ライフ サイクル(1)
ビットストリームの機密性と認証 (対称)(2)	受動的	防止	コンフィギュレーション前 および コンフィギュレーション中
揮発性の 256 ビット BBRAM キー ストレージ	受動的	防止	コンフィギュレーション前
不揮発性の 256 ビット eFUSE キー ストレージ(3)	受動的	防止	コンフィギュレーション前
書き込み専用キーの読み込みと整合性チェック (BBRAM および eFUSE)(2)	受動的	防止	コンフィギュレーション前
難読化キーの読み込みとストレージ(2)	受動的	防止	コンフィギュレーション前
ビットストリームの認証 (非対称)	受動的	防止	コンフィギュレーション前
不揮発性の 384 ビット eFUSE 公開キー ハッシュ ストレージ(2)(3)	受動的	防止	コンフィギュレーション前
DPA の保護(2)	受動的	防止	コンフィギュレーション中
ハード化されたリードバック無効化回路	受動的	防止	コンフィギュレーション後

表 1: AT 機能の分類と概要 (続き)

UltraScale および UltraScale+ FPGA のシリコン AT 機能	種類	カテゴリ	ライフ サイクル[1]
JTAG ポートの恒久的な無効化 (eFUSE)[2][3]	受動的 または能動的	防止 または応答	コンフィギュレーション前 または コンフィギュレーション後
JTAG ポートの一時的な無効化	受動的 または能動的	防止	コンフィギュレーション後
JTAG ポートの監視	能動的	検出	コンフィギュレーション後
コンフィギュレーション メモリの整合性チェック	能動的	検出	コンフィギュレーション後
固有識別子 (Device DNA およびユーザー eFUSE)	能動的	検出	コンフィギュレーション後
オンチップ温度および電圧の監視/警告	能動的	検出 および応答	コンフィギュレーション後
連続した内部クロック ソース	能動的	検出	コンフィギュレーション後
外部 PROGRAM_B インターセプト	能動的	防止 および検出	コンフィギュレーション後
コンフィギュレーション メモリの消去	能動的	応答	コンフィギュレーション後
キーの俊敏性 (BBRAM のみ)[2]	能動的	応答	コンフィギュレーション後
BBRAM キーのゼロ化 (消去 + 検証)[2]	能動的	応答	コンフィギュレーション後
不揮発性 (eFUSE) 不正操作イベントのログ記録[2]	能動的	応答	コンフィギュレーション後
ビットストリーム復号器の永久的な (eFUSE) 無効化[2][3]	能動的	防止 または応答	コンフィギュレーション後
グローバル トライステート (GTS) の有効化	能動的	応答	コンフィギュレーション後
グローバル セット/リセット (GSR) の有効化	能動的	応答	コンフィギュレーション後

注記:

1. FPGA の動作サイクルでこの機能が有効になるとき (コンフィギュレーションの前、実行中、後) を示します。
2. UltraScale および UltraScale+ FPGA の新機能または強化された機能です。
3. 一部の「永久的な」不正操作ペナルティ (eFUSE ベース) のアサートは元に戻すことができないため、デバイスをザイリンクスに返却可能かどうかに影響する可能性があります。詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 3] に記載されている eFUSE セキュリティ レジスタ (FUSE_SEC) の表を参照してください。

以降では前述の機能の仕組みと必要性、および適切な使用方法の具体例 (単独使用、またはほかのビルトイン機能やユーザー ロジックと組み合わせて使用) について詳しく説明します。さらに、FPGA デザインやシステム全体の不正操作防止レベルを強化するために有効な手法やテクニックについても具体的に説明します。

FPGA レベルで有効化される AT 機能は常に、包括的なシステム レベル AT ソリューションの一環として考える必要があります。この資料で説明する機能やテクニックは、FPGA にとって非常に優れた AT という傘のような機能を果たしますが、AT が効果を最も発揮するのは、常にシステム全体を考慮して多層アプローチで展開された場合です。

受動的 AT シリコン機能

ビットストリームの機密性と認証 (対称)

暗号化したビットストリームを外部フラッシュなどに格納し、FPGA コンフィギュレーション中に FPGA の内部復号化エンジンで復号化することで、非常に高レベルの機密性が確保されます。これにより、同じ対称型の秘密キーを共有するユーザーのみがビットストリーム内の情報にアクセスできるようになります。ビットストリームの暗号化と復号化により、システムの停止中やコンフィギュレーション中の機密性が確保され、ブロック RAM やフリップフロップの初期化データなどの FPGA デザインのコンテンツが保護されます。ザイリンクスでは、外部に格納されるビットストリームは常に暗号化された形式で保持することを強く推奨しています。

注記: UltraScale および UltraScale+ FPGA では、NIST (National Institute of Standards and Technology) が認定する Galois/Counter Mode (GCM) の 256 ビット キーの AES を使用しています。ザイリンクスに関する NIST の CAVP 認定については、http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html で検証番号 2800 を参照してください。

このセキュリティ機能を利用するには、まず、Vivado ツールで write_bitstream Tcl コマンドおよび XDC ファイルに定義されている適切なプロパティを使用してコンフィギュレーション ビットストリームを暗号化する必要があります [参照 2] [参照 3] [参照 7]。Vivado ツールではユーザーが指定したキーを用いて暗号化が実行されます。AES キーが提供されていない場合は、Vivado ツールによってオプションで自動的にキーが生成されます。ただし、Vivado ツールで生成されるキーは疑似ランダムであり、実際のランダム キーほどセキュリティ レベルは高くありません。このキーは、Vivado ハードウェア マネージャーを使用して JTAG ポート経由で FPGA へロードされます。

UltraScale および UltraScale+ FPGA でビットストリームの復号化が有効になっており、かつ非対称 RSA-2048 認証が eFUSE で有効になっていない場合、AES-GCM は認証付きの暗号化/復号化アルゴリズムであるため、対称認証が自動的に有効になります。AES-GCM では、機密性を確保するカウンター モードをユニバーサル ハッシュ (認証タグ) 関数に基づいた認証メカニズムと組み合わせています。そのため AES-GCM では、機密性だけでなく、完全性と認証も同時に実現できます。この暗号化ベースの厳密な認証スキームにより、ビットストリームの変更が試みられると、それがシングル ビットであってもビットストリームのシグネチャが大幅に変更されるため、デバイスは起動されません。基本的に、対称認証を利用すればソースが正規であることが保証されます。つまり、デバイスをコンフィギュレーションすることが認可されているユーザーのみが、デバイスのコンフィギュレーションを実行できます。

認証チェックに合格すると、デバイスは通常動作を開始し、スタートアップ コマンドが実行されます。ビットストリームの読み込み後に DONE 出力信号が High にアサートされない、INIT_B 信号が Low をアサートする 、およびコンフィギュレーション ステータス レジスタの HMAC_ERROR ビットが High をアサートする場合、認証が失敗したことを示します。認証失敗は、ビットストリームが不正操作された可能性があることを示しますが、ビットストリームの読み込みに使用したチャネルがノイズの影響を受け、コンフィギュレーション プロセス中にビット破損が発生した可能性も考えられます。

また、AES-GCM は高スループットのハードウェア インプリメンテーションを容易に実行できるように設計されています。従来ファミリでは 8 ビット幅形式に制限されていましたが、UltraScale および UltraScale+ FPGA の AES-GCM 復号化では、32 ビット幅形式の暗号化ビットストリームを使用できます。そのため、非暗号化ビットストリームでのデバイス コンフィギュレーション時間が暗号化ビットストリームの場合とほぼ同じになります。

揮発性キーおよび不揮発性キーのストレージ

256 ビットの対称 AES-GCM キーは、FPGA 内の揮発性 BBRAM または不揮発性 eFUSE のいずれかのワンタイム プログラマブル (OTP) ストレージへロードできます。いずれのストレージにキーを格納するかを判断するには、BBRAM (表 2) と eFUSE (表 3) それぞれのストレージの長所と短所を理解しておく必要があります。

表 2: BBRAM ストレージ: 長所と短所

長所	短所
• 揮発性、再プログラム可能 • 受動的および能動的にキーを消去可能 (つまり、証拠を削除できる) • 不正操作防止(1)	• 外部バッテリが必要 • 多くのバッテリ ベンダーは高温や長期利用における動作仕様を定義してない (一部のベンダーは、これらの問題に対処するために、ベータボルタ式バッテリの提供を開始)

注記:

1. BBRAM からキーを読み出すための物理的パスはありません (書き込み専用パスはある)。

表 3: eFUSE ストレージ: 長所と短所

長所	短所
• 外部バッテリが不要 • スプーフィングが困難 (ボード上のデバイスの交換が必要) ◦ eFUSE キーで暗号化されたビットストリームでのみ FPGA をコンフィギュレーション可能。cfg_aes_only[1] eFUSE ビットもセットされている場合には、その他すべてのビットが拒否される。	• 恒久的。キーの消去およびアップデートが不可[2] • BBRAM ソリューションよりもセキュリティ レベルが低い (デバイス レベルの証拠が残る)

注記:

eFUSE 制御レジスタ FUSE_SEC 内にある cfg_aes_only オプションを使用する場合には、重要なポイントが 2 つあります。

1. ビットストリームの間接フラッシュ プログラム手法 [参照 8] を使用する場合、暗号化されたビットストリームをオンボード フラッシュに読み込んだ後で、このオプション (cfg_aes_only) を有効にする必要があります。これは、ザイリンクスの間接プログラム コア ビットストリームは暗号化されていないためです。それ以外の場合、FPGA は eFUSE キーを使用して間接プログラム ビットストリームを復号化しようとし、コンフィギュレーションがエラーとなり、外部フラッシュのアップデートもエラーとなります。また、フラッシュに格納されたファームウェアで以降に実行される、フィールド アップデートにも影響します。
2. DPA 攻撃を防止する eFUSE ベースのキーを消去せずに、恒久的な AES-GCM 復号無効化機能 (eFUSE を使用) を使用できますが、eFUSE キーがデバイスに保持されたままになり、さらに高度な物理的攻撃の対象となります。

ビットストリームの暗号化とキー ストレージの詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 3]、『Vivado Design Suite ユーザー ガイド: プログラムおよびデバッグ』(UG908) [参照 7]、『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) [参照 9]、および『暗号化を使用して UltraScale FPGA ビットストリームを保護』(XAPP1267) [参照 10] を参照してください。

書き込み専用キーの読み込みと整合性チェック

BBRAM と eFUSE の両方の 256 ビット対称キーが、Vivado ハードウェア マネージャーを使用して外部 JTAG 経由で読み込まれます。内部 MASTER_JTAG ポートを使用して BBRAM キーをアップデートする方法は、「キーの俊敏性 (応答)」を参照してください。UltraScale および UltraScale+ FPGA では、このキーを読み込むパスはデバイスに対して書き込み専用となります。いずれのキーもリードバックするための物理的なデータパスはありません。従来ファミリでは、読み込みプロセス中はプロトコルによってキーが保護されており、「キー アクセス モード」に入ると、既存のキーとコンフィギュレーション メモリが即座に消去されてから新しいキーを書き込んでリードバックし、整合性チェックを実行できるようになっていました。JTAG 経由でデバイスにキーを書き込んだ場合、キーの整合性チェックは JTAG 経由でデバイスに CRC32 の想定値を書き込むことで開始されます。実際の CRC32 整合性チェックは格納されたキーに対してデバイスによって (内部で) 算出され、JTAG ポートから受信した CRC32 の想定値と比較されます。デバイスは、実際のキー情報ではなく、合/否の結果を JTAG ポートへ出力して整合性ステータスを示します。キー用の物理的なリードバック パスを排除することで、格納されたキーのセキュリティが強化されます。

注記: BBRAM ベースのキーの場合は、キーを書き込む前に、BBRAM にある既存のキーをゼロ化 (消去および検証) してください。

難読化キーの読み込みとストレージ

必要に応じて、UltraScale または UltraScale+ FPGA の eFUSE アレイまたは BBRAM (JTAG を使用) に書き込まれて格納されたキー データを難読化できます。キー データは、ザイリンクスのみが把握し、UltraScale FPGA および UltraScale+ FPGA それぞれに共通かつ固定の難読化キーを使用して暗号化されます (UltraScale FPGA の難読化キーは UltraScale+ FPGA の難読化キーとは異なる)。これにより、委託製造メーカーにおける秘密のレッド キーの保護など、商用量産時のセキュリティ レベルが強化されます。内部に格納された難読化キーは、暗号化ビットストリームの読み込み開始時に解読され、その後ビットストリームの復号化に使用されます。この機能は、FUSE_SEC 制御レジスタの制御ビット (eFUSE ベースのキー) または BBRAM に書き込まれた制御ビット (BBRAM ベースのキー) で有効化されます。図 1 に、この動作の概要を示します。

注記: 難読化キー ストレージは、BBRAM キー ストレージのコンフィギュレーション カウント機能による DPA 対抗措置と併用できません (「DPA の保護」を参照)。

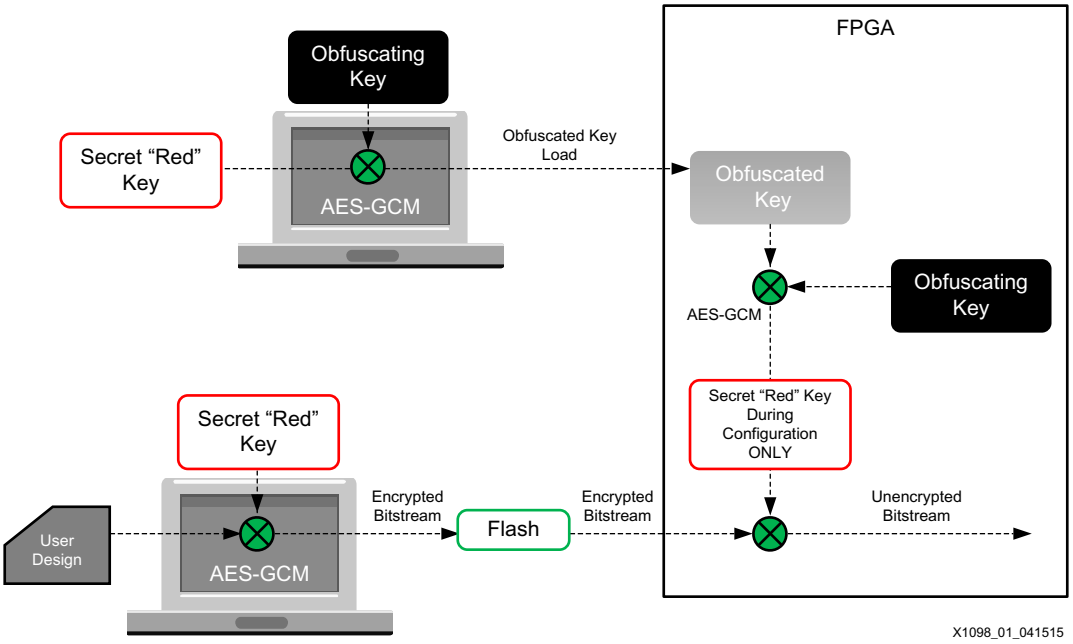

図 1: 難読化キーの読み込みとストレージの概要

ビットストリームの認証 (非対称)

UltraScale および UltraScale+ FPGA では、暗号化ビットストリーム全体をデバイスに読み込んで、オンチップの復号化エンジンに送信する前に認証できます。つまり、認証後の復号化が可能です。ビットストリームが変更された場合、それがシングル ビットの場合でも、デバイスの非対称認証機能がこの変更を検出し、復号化エンジンを無効にする (暗号化ビットストリームに対して有効になっている場合) だけでなく、デバイスの起動も防止します。つまり、この機能を有効にすると、認可されているビットストリームでのみ UltraScale または UltraScale+ FPGA をコンフィギュレーションできます。非対称認証が有効になっている場合、AES-GCM アルゴリズムの対称認証機能は実行されません。つまり、定期的かつ総合的な対称認証チェックは実行されません。

この方法では RSA-2048 非対称デジタル署名 (認証) アルゴリズムを使用しているため、認証タスク用にデバイスに秘密キーを格納する必要はありません。代わりに、非対称認証機能ではユーザー定義の公開キー情報が格納されます。領域が限られるため、この認証方法では 2048 ビット公開キーの 384 ビット SHA-3 ハッシュを使用し、ハッシュ値は UltraScale または UltraScale+ FPGA の eFUSE ビットにプログラムされます。必要に応じて、秘密キーと公開キーのペアを定義します。キー ペアは OpenSSL や SafeNet などのさまざまなオープン ソース製品や商用製品を使用して生成できます。この認証方法には秘密情報が必要がないため、サイドチャネル解析などの攻撃が発生した場合に、攻撃者に悪用されるような情報は流出しません。

次のような理由から、RSA 非対称認証を使用します。

1. 復号化する前に、ビットストリーム全体を認証します。この手法は、「DPA の保護」で説明している DPA 攻撃への対抗措置に含まれています。
2. 認可されていないユーザーが、悪意のある可能性がある独自デザインを UltraScale または UltraScale+ FPGA で実行するのを防ぎます。認可されているユーザーが eFUSE ビットに公開キー ハッシュをプログラムしており、RSA_AUTH_ALL_EFUSE eFUSE がプログラム (RSA 認証が適用) されている場合は、認可されたビットストリームのみが読み込まれます。
3. 暗号化されていないビットストリームの認証。FPGA デザインに CT は含まれていなくても、認証が必要となる場合があります。次がその例になります。
    a. デザインに、AES 暗号化アルゴリズムなどの広く知られている機能が含まれている場合。デザインを機密扱いにする必要はありませんが、たとえば外部ピンでレッド キーやデータを出力するなど、変更されないようにする必要があります。
    b. FPGA デザインに、基本機能から高度な機能に至るまで、さまざまなレベルの機能が含まれている場合。たとえば、顧客によってアクセスできる機能が異なり、すべての機能にアクセスできる顧客と基本機能にのみアクセスできる顧客がいる場合があります。攻撃者が非暗号化ビットストリームを変更し、検知されずに拡張機能を試したり、使用することはできません。

図 2 に、RSA/SHA-3(1) を使用してビットストリームが作成される仕組みと、ビットストリーム全体がオンチップで内部認証される仕組みについて概要を示します。図の上半分は、RSA 認証のビットストリームを作成するためにソフトウェア ツールによって実行され、図の下半分は、ビットストリームを認証するために UltraScale または UltraScale+ FPGA 内部で実行されます。

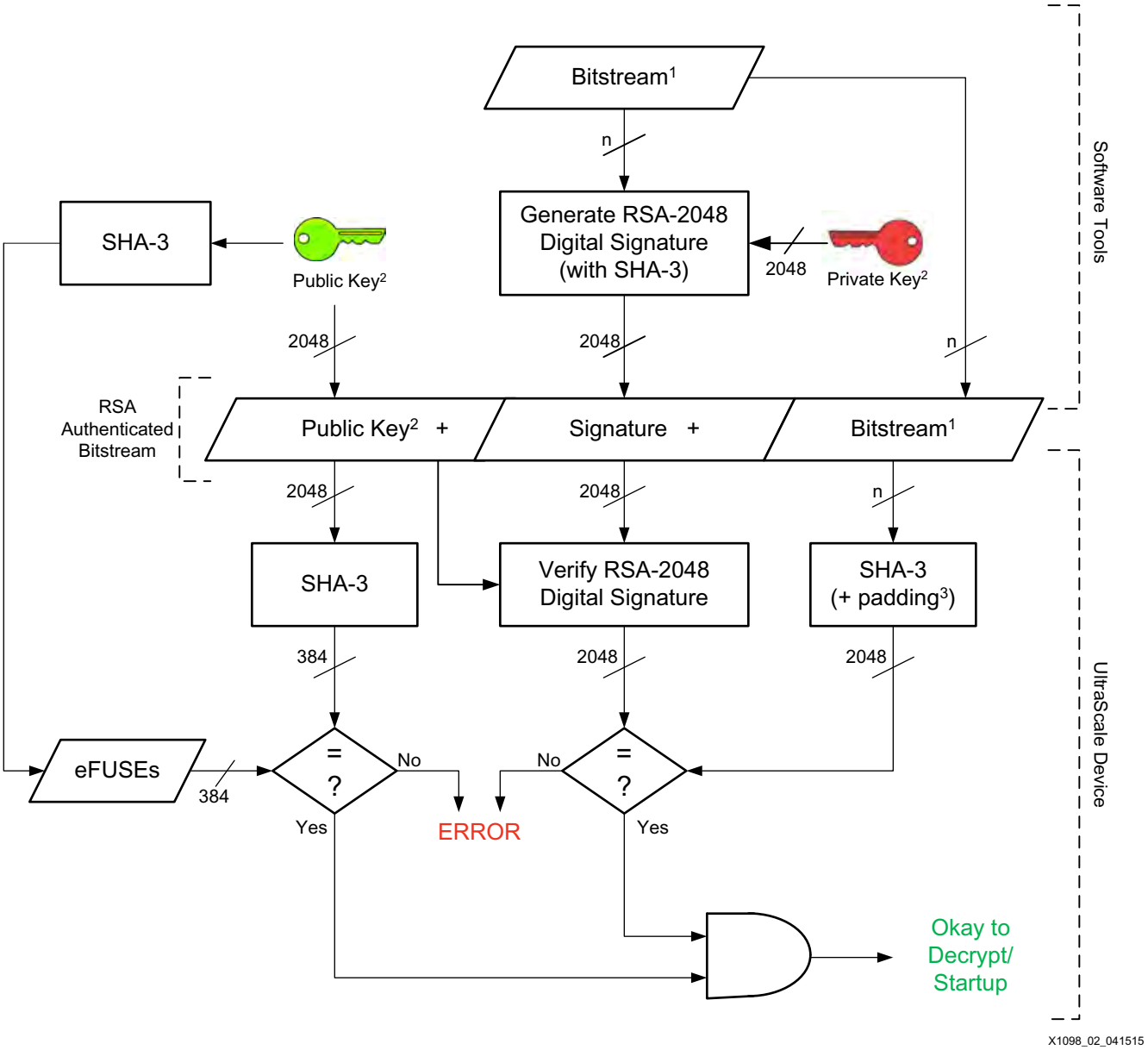

図 2: RSA 署名を使用したビットストリームの作成

図 2 について説明します。

1. ビットストリームが暗号化される場合 (暗号化テキスト) と暗号化されない場合 (プレーンテキスト) があります。
2. ユーザーが生成した秘密/公開キー ペアです。
3. PKCS #1 v1.5 パディング スキームです。

1. RSA-2048 および SHA-3 は現時点では NIST 標準に準拠していません。詳細は、ザイリンクスまでお問い合わせください。

図 3 に、暗号化されて RSA で認証されるビットストリームの実際の形式を示し、プレーンテキスト、認証対象のプレーンテキスト、および認証対象の暗号化テキストがそれぞれどの部分であるかを示します。

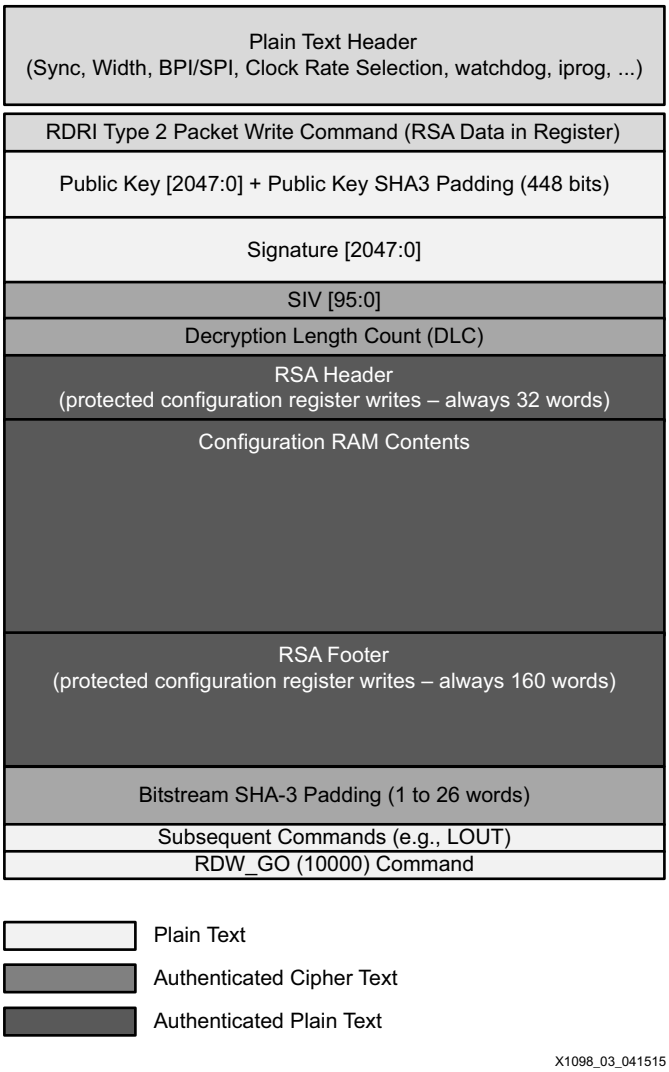

図 3: RSA を使用するビットストリームの形式

ビットストリーム全体はデバイスに読み込まれた後、使用前に認証されるため、暗号化されたビットストリームに RSA 非対称認証を使用する場合、追加のコンフィギュレーション時間が生じます。ただし、使用するコンフィギュレーション ポートによっては、コンフィギュレーション時間全体としてはそれほど大幅には増加しません。これは、ビットストリーム全体が読み込まれて認証された後では、復号化プロセスで 32 ビット幅のデータ バス全体を利用できるためです。厳密なコンフィギュレーション時間の詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 3] を参照してください。

シリコン デザインに一部制約があり、UltraScale および UltraScale+ FPGA で RSA 非対称認証を使用する際はいくつかの制限があります。

- RSA 認証のビットストリームは圧縮できません。圧縮の詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) を参照してください。
- パーシャル リコンフィギュレーション (PR) のビットストリームは、ビルトイン シリコン機能を使用して RSA 認証できません。独自の認証機能を FPGA ロジックで使用できます。PR の詳細は、『Vivado Design Suite ユーザー ガイド: パーシャル リコンフィギュレーション』(UG909) を参照してください。

- Tandem ビットストリームは RSA 認証できません。Tandem コンフィギュレーションは、オープン PCIe システムでのエニュメレーション ニーズに対応するために PCIe® デザインの高速コンフィギュレーションを可能にする、ザイリンクス ソリューションです。
- UltraScale FPGA の場合のみ、RSA を有効にすると、ビットストリームのコンフィギュレーション幅に制限が発生します。一部の狭いコンフィギュレーション幅がサポートされません。詳細は、『UltraScale FPGA RSA 認証およびサポート コンフィギュレーション モード』(XCN15038) [参照 11] を参照してください。

DPA の保護

攻撃者は通常、セキュリティ機能を直接攻撃する (実行不可能なキー総当たり攻撃を使用して FPGA ビットストリームの AES-256 復号化を解読するなど) のではなく、サイドチャネル解析などのより簡単な方法を探します。サイドチャネルは、電子デバイス内に存在する意図的でない情報流出パスです。観測期間が十分な場合は、このパスから、暗号化機能のレッド キー データなどの秘密情報を抽出できる可能性があります。

差分電力解析 (DPA) はサイドチャネル テクニックで、電源ラインに直列に低抵抗を挿入するか、近接した位置での電磁プローブによって電圧をモニターすることで、機能中の電子デバイスにおけるデジタル スイッチングによる消費電力変化のサンプルを観測し、記録します。その後、信号処理と統計手法を用いて記録されたデータからキー データが抽出されます。攻撃者の能力が向上するにつれ、必要となるサンプル データの数も減少しています。

ザイリンクスは、任意のキーで不正に収集される可能性のあるサイドチャネル データの量を制限することで DPA 耐性を提供します。このプロトコル ベースのデータ制限手法を UltraScale および UltraScale+ FPGA で使用することにより、オンチップのビットストリーム復号器の DPA 攻撃を軽減します。この手法では、保護レベルがプログラム可能であり、攻撃者の能力が向上するのに伴い保護レベルも強化できるため、長期にわたって最大限の柔軟性が得られます。

その際、無効/ランダムなビットストリーム データと有効なビットストリーム データの 2 種類のデータを制限する必要があります。効果を得るには、これら両方のビットストリームに対する攻撃への対抗措置が必要です。

無効/ランダムなビットストリーム データ

従来、DPA 攻撃者は、大量のランダム データを復号器の暗号化テキスト入力ポートに送り込むだけで、解析用のサイドチャネル情報を収集していました。UltraScale および UltraScale+ FPGA でこの無効/ランダムなビットストリームを検出するには次の 2 つの方法があり、いずれかを選択します。

1. コンフィギュレーション カウントにより、無効なビットストリームを復号化プロセス中にリアルタイムですばやく検出します。UltraScale および UltraScale+ FPGA により、8 ワードごとに 32 ビットの定期的な対称認証チェックが追加されます。この定期的な認証がエラーとなった場合は、復号化が即座に停止され、UltraScale または UltraScale+ FPGA で無効なコンフィギュレーション試行としてマークされます。この方法は、BBRAM ベースの AES-GCM キーの場合にのみ使用する必要があり、無効なコンフィギュレーション試行が特定の回数に達すると、不正操作ペナルティが発生して BBRAM キーがゼロ化されます。許容されるコンフィギュレーション試行数はプログラムできます。

   注記: 難読化キー ストレージは、BBRAM キー ストレージのコンフィギュレーション カウント機能による DPA 対抗措置と併用できません。

2. 暗号化ビットストリームを非対称認証 (RSA) を使用して認証後復号化してから、復号器に送信します。シグネチャ検証プロセスに対するサイドチャネル攻撃は、検出するような秘密情報がないため意味がありません (公開キーと公開キー ハッシュは誰でも把握できる)。この方法は eFUSE または BBRAM ベースの AES-GCM キーに使用でき、不正操作ペナルティは発生しません。この方法を使用する場合、対称認証チェックは実行されません。

コンフィギュレーション カウント ソリューション (ソリューション 1) では、コンフィギュレーション試行を追跡する関連ダウン カウンターが BBRAM 内にあります (最大初期カウント値は 255 であり、キー読み込み時にユーザーが設定)。このカウンターは、無効なコンフィギュレーション試行をカウントするか、または無効か有効かに関係なくすべてのコンフィギュレーション試行をカウントするように設定できます。カウンターは各コンフィギュレーション試行の前にデクリメントされます。無効なコンフィギュレーションのみをカウントするように設定した場合、コンフィギュレーションが正常に実行されると、カウンターがインクリメントされます。すべてのコンフィギュレーションをカウントするように設定した場合は、各コンフィギュレーションの終了時にカウンターはデクリメントされたままになります。

カウンターがターミナル カウント 0 に達すると、BBRAM キーは不正操作ペナルティとしてゼロ化されます。初期カウント値が小さいほど、収集可能なサイドチャネル データの量が少なくなるため、保護レベルが高くなります。また、カウンターの動作と BBRAM の整合性を検証するために、セキュリティ チェックが実行されます。

ソリューション 1の場合、意図的にランダム (無効) なビットストリームとシグナル インテグリティの問題が原因で無効なデータを区別する方法はありません。そのため、FPGA が十分な回数コンフィギュレーションされると、最終的には BBRAM キーがゼロ化されます。このソリューションを使用する場合、シグナル インテグリティの問題が生じないように、メモリ デバイスから FPGA のコンフィギュレーション ポートへのデータパスを堅牢に設計することが重要です。

ランダム データ ソリューション (ソリューション 2) の場合について、図 4 に、認証後復号化によってランダム データの復号化を防止する仕組みの概要を示します。

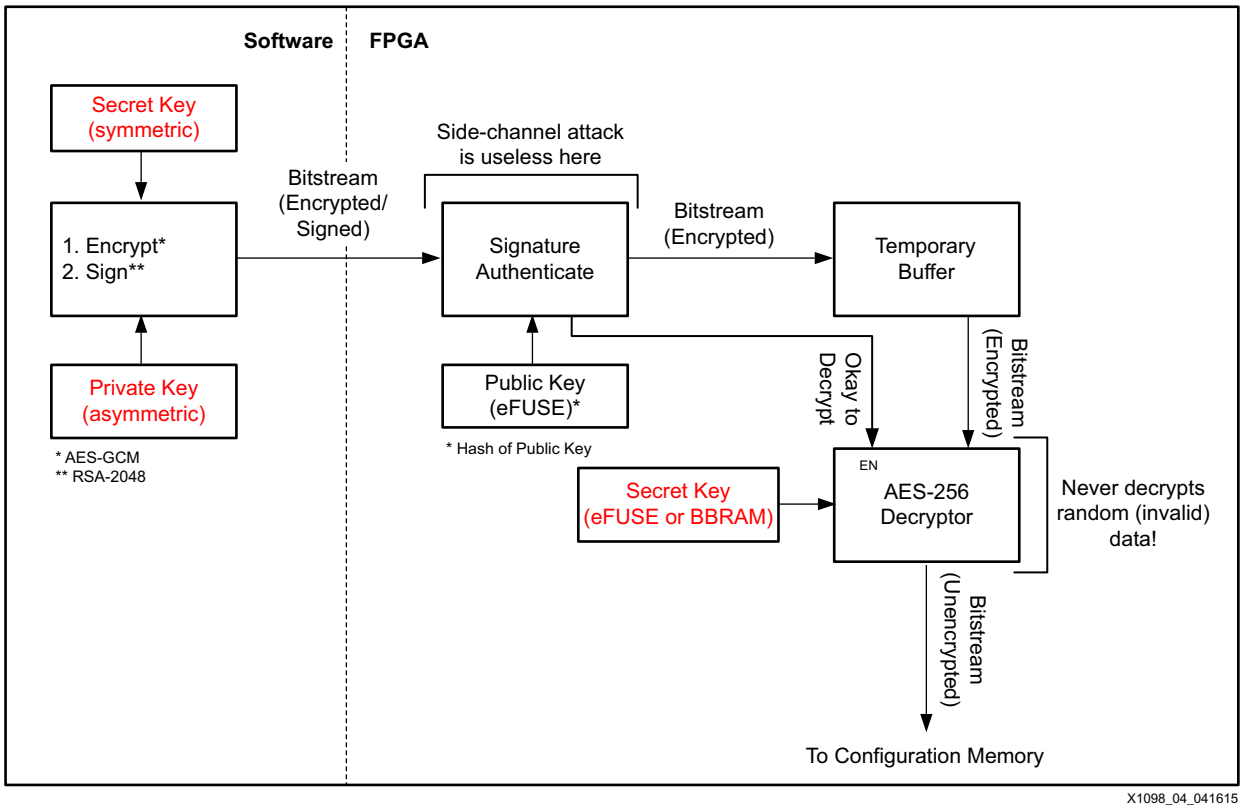

図 4: 認証後復号化によるランダム データ攻撃の防止

図 4 に示す一時バッファーは実際にはコンフィギュレーション メモリであり、通常はユーザー ロジックを定義するデータを保持するために使用されます。ただし、ここでは RSA 認証ステップが完了するまで暗号化されたビットストリームを安全に保持するために使用されています。認証に合格すると、暗号化されたビットストリーム データの各フレームが復号化され、コンフィギュレーション メモリ内の同じ場所に戻されます。これは、読み出し-復号化-書き込み (RDW) フェーズです。「ビットストリームの認証 (非対称)」で説明したように、この RDW フェーズは、コンフィギュレーションに使用されている指定の CCLK 周波数 (内部指定の CCLK 周波数または EMCCLK から派生した CCLK 周波数) で動作します。

コンフィギュレーション カウントのトレードオフ

個々の暗号化ブロックに関連する一定のオーバーヘッドが発生するため、コンフィギュレーション カウントのランダム データ攻撃ソリューションを使用する場合、作成するブロックが小さいほどビットストリームのサイズが大きくなります。そのため、コンフィギュレーションのストレージおよび時間と、セキュリティ レベルとの間でトレードオフを検討する必要があります。コンフィギュレーション カウントでキー ローリングを使用する場合は常に、ビットストリームのサイズが少なくとも 14% 程度増加します。ブロック サイズが小さくなるほど安全性は高くなりますが、ビットストリームのサイズが増加し、コンフィギュレーション時間も長くなります。図 5 に、キーごとのブロック数 (x 軸) が少なくなるにつれて、ビットストリームのサイズ (y 軸) がどのように増加するかを示します。

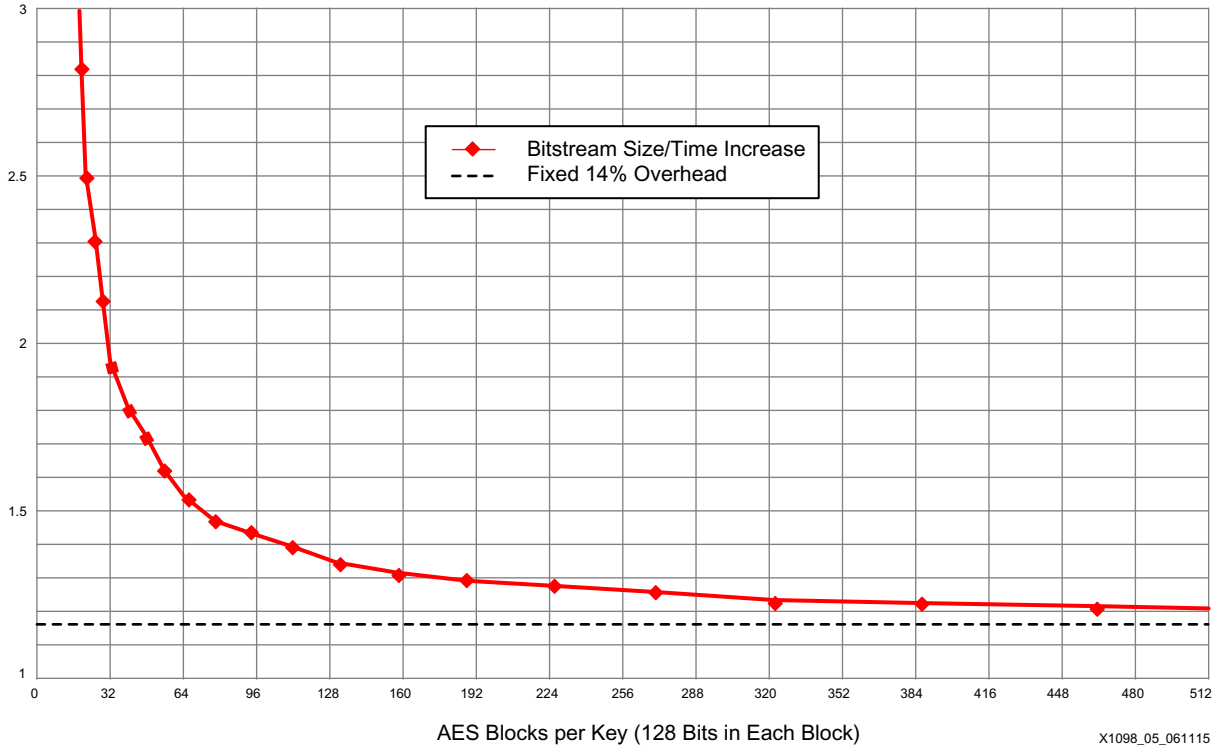

図 5: コンフィギュレーション カウントでのキー ローリングのトレードオフ

認証後復号化のトレードオフ

暗号化されたビットストリーム全体は復号化の前に FPGA のコンフィギュレーション メモリに読み込まれるため、ビットストリームのサイズはデバイスのメモリ容量によって制限されます。この場合、コンフィギュレーションの総時間に対する影響の割合を検討する必要があります。この影響は、コンフィギュレーションに使用する外部ポートによって異なります。前述したように、内部 RDW フェーズは常に 32 ビット幅で、コンフィギュレーションに使用する指定のクロック周波数で実行されます。そのため、外部ポートがそれよりも低速でありボトルネックとなる場合、全体的なコンフィギュレーション時間にはそれほど影響しない可能性があります。反対に、高速な外部ポートを使用する場合、コンフィギュレーション時間の全体的な増加割合が大きくなります。次に、例をいくつか示します。

- JTAG およびマスター SPI (x1):
  全体的なコンフィギュレーション時間が ~8% 増加
  シリアル ロード フェーズの後に 32 ビット パラレル RDW フェーズ
- マスター SPI クワッド (x4):
  全体的なコンフィギュレーション時間が 32% 増加
  クワッド SPI ロード フェーズの後に 32 ビット パラレル RDW フェーズ
- マスター SPI デュアル クワッド (x8):
  全体的なコンフィギュレーション時間が 63% 増加
  デュアル クワッド SPI ロード フェーズの後に 32 ビット パラレル RDW フェーズ

- マスター (CCLK) 非同期 BPI (x16):
  コンフィギュレーション時間が ~125% 増加
  16 ビット パラレル ロード フェーズの後に 32 ビット パラレル RDW フェーズ
- マスター (EMCCLK) 同期 BPI (x16):
  コンフィギュレーション時間が ~125% 増加
  16 ビット パラレル ロード フェーズの後に 32 ビット パラレル RDW フェーズ

注記: これらすべての SPI モード (RSA が有効) が UltraScale+ FPGA でサポートされています。UltraScale FPGA でサポートされている SPI モード (RSA を有効化) の詳細は、『UltraScale FPGA RSA 認証およびサポート コンフィギュレーション モード』(XCN15038) [参照 11] を参照してください。

重要: 前述のリストの後半の例では、コンフィギュレーション時間の増加率が最も大きくなっていますが、コンフィギュレーションの総時間はリストの最初の例よりも大幅に短くなっています。

重要: 暗号化されたビットストリーム全体が復号化の前に FPGA のコンフィギュレーション メモリに収まる必要があるため、キー ローリング ブロックをどの程度小さくできるかには限界があります。認証後復号化の場合、キー ローリング ブロックのサイズは、UltraScale FPGA では 246 個の AES 復号化ブロックよりも小さくできず、UltraScale+ FPGA では 186 個の AES 復号化ブロックよりも小さくできません (各 AES 復号化ブロックは 128 ビット)。

有効なビットストリーム データ

最新の FPGA のビットストリーム長は、有効なビットストリームの 1 回のロードで DPA 攻撃を実行できるほどの長さになっています (入力をさまざまに変化させることで有効なビットストリーム データは十分にランダムになる)。これを防ぐために、UltraScale および UltraScale+ FPGA のビットストリームを複数の小さいブロックに分割し、各ブロックを専用の固有なユーザー定義キーを使用して暗号化できます。セキュリティ要件に応じて、各ブロックのサイズをプログラムできます。ブロックのサイズが小さいほど、特定のキーに対応する収集可能なサイドチャネル データが少なくなるため、セキュリティが向上します。

オンチップ メモリ (BBRAM または eFUSE) にすべての復号化キーを格納せずに済むように、UltraScale および UltraScale+ FPGA ではキー ローリング手法が使用され、最初のキー (key 0) のみがオンチップに格納され、以降の各ブロックのキーは前のブロック内で暗号化 (ラップ) されます。図 6 にこの概念を示します。

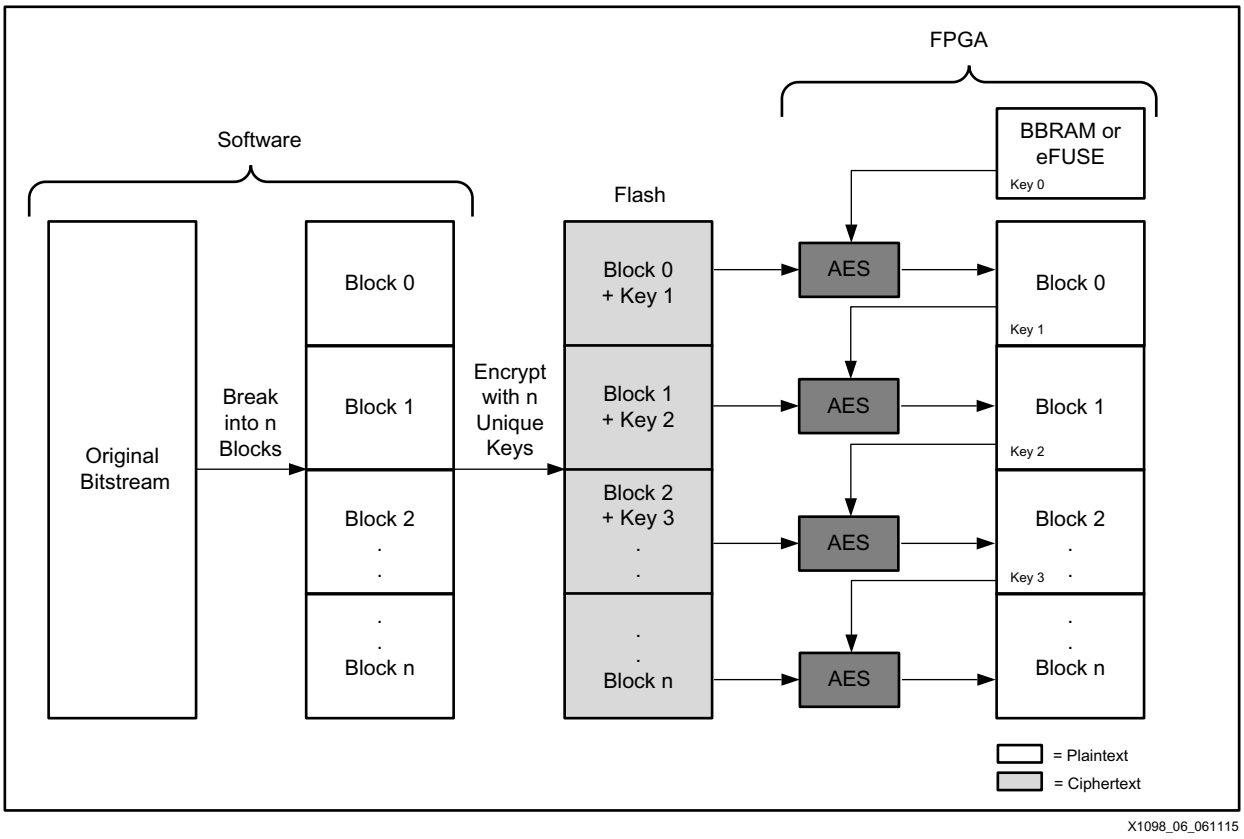

図 6: キー ローリング

有効なビットストリームを何度でも無制限にロードすることは可能ですが、追加のコンフィギュレーションにより新たなサイド チャネル情報が攻撃者に流出することはありません。攻撃者が実行できるのは、最初のコンフィギュレーションに含まれているサイド チャネル データの信号対ノイズ比を低減させることだけです。キー ローリング手法をコンフィギュレーション カウントまたは認証後復号化と併用すると、攻撃者が暗号化テキスト ポートに適用する値を取得することはできません。

重要: キー ローリング (有効データ攻撃への対抗措置) をいずれかのランダム データ攻撃への対抗措置 (コンフィギュレーション カウントまたは認証後復号化) と併用する必要があります。

キー ローリング手法をコンフィギュレーション カウントまたは認証後復号化と組み合わせる際に考慮すべきトレードオフがあります。

DPA ソリューションの比較

コンフィギュレーション カウントと認証後復号化の 2 つのランダム データ対抗措置から、UltraScale または UltraScale+ FPGA で一度に有効化できるのは 1 つのみであるため、これらのトレードオフを理解しておく必要があります。包括的な DPA 対抗措置を使用するために、キー ローリングは必ずこれらの措置のいずれかと併用する必要があります。表 4 にトレードオフの概要を示します。

表 4: コンフィギュレーション カウントおよび認証後復号化のトレードオフの比較

メトリクス	コンフィギュレーション カウント	認証後復号化
サポートされるキー ストレージ	BBRAM	BBRAM および eFUSE
各キーで公開される最小データ(1)	4 ブロック	246 ブロック (UltraScale FPGA)/ 186 ブロック (UltraScale+ FPGA)
ビットストリーム サイズの増加	あり	なし
コンフィギュレーション時間の増加	キーの有効期間に基づく	コンフィギュレーション モードに基づく
eFUSE プログラムの必要性	なし	あり(2)
すべてのビットストリーム機能 (圧縮、PR、Tandem) のサポート	あり	なし
フィールドでのキー保守	あり	なし

注記:

1. DPA 攻撃に関する最新の公開文献は、www.dpacontest.org/home を参照してください。
2. RSA 公開キーのハッシュに必要です。

ハード化されたリードバック無効化回路

暗号化されたビットストリームまたは認証されたビットストリームが FPGA にロードされた場合、JTAG をはじめとする外部インターフェイスによる内部コンフィギュレーション メモリのリードバックは一切実行できなくなります。ハード化された三重冗長ロジックにより、すべての外部リードバックが自動的にブロック (無効化) されます。暗号化されたビットストリームをロードした後にコンフィギュレーション メモリをリードバックできるのは、内部コンフィギュレーション アクセス ポート (ICAPE3) を介した場合のみです。ビットストリームはロード プロセス中に認証されるため、ICAPE3 は信頼できるチャネルと見なされます。ICAPE3 が信頼できるチャネルと見なされるもう 1 つの理由は、ICAPE3 が FPGA ロジック内のデザインへ直接アクセスすることによってのみ利用できるためです。デザインで ICAPE3 がインスタンシエートされていない場合は、ICAPE3 は一切使用できません。

注記: ビットストリーム内の特定の制御ビットを使用する Vivado ツールのセキュリティ オプションで、リードバックを有効/無効にできます。このビットはコンフィギュレーション中に変更可能です。そのため、暗号化または認証されたビットストリームを使用していないデバイスでは、攻撃者によりリードバックの無効設定は容易に変更されてしまいます。ハード化されたリードバック無効化回路にはこのような弱点はなく、暗号化または認証されたビットストリームを使用している場合、セキュリティ オプションよりも常に優先されます。ハード手法とソフト手法の両方のリードバック無効化を使用することが推奨されます。

JTAG ポートの無効化 (受動的)

一時的な方法と恒久的な方法の 2 つの受動的な方法で、外部 JTAG ポートを無効にできます。一時的な方法では、Vivado ツールで write_bitstream Tcl コマンドを使用するときに、set_property BITSTREAM.GENERAL.DISABLE_JTAG YES [current_design] オプションを追加します。この場合、コンフィギュレーション ビットストリームがロードされた後で外部 JTAG ポートが無効になります。この際、内部 MASTER_JTAG ポートも無効になります。

注記: Vivado ツールのオプションを使用した場合、攻撃者が後でこのオプションを無効にしようとすると、認証によってビット反転攻撃が検出され、デバイスは起動されません。この変更は、有効になっている認証方法に応じて、RSA (非対称) アルゴリズムまたは AES-GCM (対称) アルゴリズムのいずれかによって検出されます。

Vivado ハードウェア マネージャーを使用して、外部 JTAG ポートを恒久的に無効にするように eFUSE ビット (FUSE_SEC レジスタ内の FUSE_SHAD_SEC[3] ビット) をプログラムできます。この方法では、実質的には外部ピンが無効にされて FPGA ロジック内部用に使用されます。この方法は、認証が必要な場合や、ユーザー デザインをロードする間のみ JTAG の無効化が必要な場合に、外部 JTAG の無効化として効果的です。製造施設で最後の手順の 1 つとして、eFUSE をプログラムする前に回路基板上でテスト用に JTAG ベースのバウンダリスキャンを使用できるように、この方法を実施することが考えられます。

能動的 AT シリコン機能

「はじめに」で説明したように、能動的 AT 機能を利用するには FPGA ロジック デザインに手を加える必要があります。たとえば、不正操作イベントに応答して KEYCLEARB 入力を駆動するには、デザインに STARTUP プリミティブをインスタンシエートする必要があります。表 5 に、これらの能動的 AT 機能、使用例、および実装方法について概要を示します。

表 5: 能動的セキュリティ機能の使用例

機能	使用例	方法
JTAG ポートの恒久的な無効化 (eFUSE)(1)	不正操作イベントに応答して不正な JTAG アクセスを恒久的に防止します。	MASTER_JTAG プリミティブをインスタンシエートするか、DISABLE_JTAG eFUSE (恒久的な無効化) を内部でプログラムします。認証も強制する必要があります (aes_efuse_only または symmetric with rsa_auth_all を使用した対称認証)。
JTAG ポートの一時的な無効化	不正な JTAG アクセスを防止します。	MASTER_JTAG プリミティブをインスタンシエートするか、DISABLE_JTAG 属性を TRUE に設定して BSCANE2 プリミティブをインスタンシエートするか、BITSTREAM.GENERAL.DISABLE_JTAG オプションを使用します。
JTAG ポートの監視	不正な JTAG アクセスを検出します。	BSCANE2 プリミティブをインスタンシエートして、FPGA ロジックに監視/応答機能を追加します。MASTER_JTAG を使用する場合、内部 JTAG ポートの活動が検出されます。
コンフィギュレーション メモリの整合性チェック	コンフィギュレーション メモリの整合性をバックグラウンドでチェックします (干渉なしのランタイム チェック)。	Soft Error Mitigation (SEM) IP コアをインスタンシエートします [参照 12]。
固有識別子 (Device DNA およびユーザー eFUSE)	固有識別子が認識されない場合、デザインの動作を不可にするか、制限付きで動作可能にします。	固有識別子を読み出して処理し、それらが有効かどうかを判断できる FPGA ロジックを構築します。
オンチップ温度および電圧の監視/警告	通常の環境範囲内でデバイスが動作していることを確認します。	システム モニター (SYSMONE1) プリミティブをインスタンシエートし、環境状態をチェック/応答する FPGA ロジックを構築します。
連続した内部クロック ソース	外部クロック ソースを削除するだけでは、能動的 AT 機能が無効化されないようにします。	STARTUPE3 プリミティブをインスタンシエートし、CFGMCLK 出力に接続して、ユーザー定義の AT 機能のクロック ソースとして使用します。クロックが無効化されないように、ビットストリーム暗号化も使用する必要があります。

表 5: 能動的セキュリティ機能の使用例 (続き)

機能	使用例	方法
外部 PROGRAM_B インターセプト	デバイス コンフィギュレーションを遅延させて、リセットできないデータ エレメント (トランシーバーの FIFO などの、コンフィギュレーション前にハウス クリーニングでは自動的にクリアされないエレメント) を消去できるようにします。	STARTUPE3 プリミティブをインスタンシエートして、PROG_REQ 受信後の PROG_ACK アサートの正しい条件を決定する FPGA ロジックを構築します。
コンフィギュレーション メモリの消去	不正操作イベントに応答してコンフィギュレーション メモリを消去します。	ICAPE3 プリミティブをインスタンシエートして、IPROG コマンド送信の正しい条件を決定する FPGA ロジックを構築します。
キーの俊敏性 (BBRAM のみ)(1)	ボードやモジュールをセキュアな施設に戻さずに BBRAM キーをフィールドで安全にアップデートします。	キー管理イベントに応答して FPGA ロジックでセキュアなキー交換を実行できるロジックと共に、MASTER_JTAG プリミティブをインスタンシエートします。
BBRAM キーのゼロ化 (消去 + 検証)(1)	不正操作イベントに応答してバックアップ バッテリ付きキーをゼロ化します。	STARTUPE3 プリミティブをインスタンシエートして、KEYCLEARB アサートおよび STATUS レジスタ内の検証ビット読み出しの正しい条件を決定する FPGA ロジックを構築します。
不揮発性 (eFUSE) 不正操作イベント ログ(1)	以降のフォレンジック分析に備えて、不正操作イベントを不揮発性メモリ (eFUSE) に安全に記録します。	MASTER_JTAG プリミティブをインスタンシエートして、eFUSE ビットに不正操作イベントを記録するための FPGA ロジック機能を構築します (eFUSE ビットのプログラム時は POST_CRC を一時停止する)。
ビットストリーム復号器の恒久的な無効化 (eFUSE)(1)	予防手段として、または不正操作イベントに応答して、ビットストリーム復号器のサイドチャネル解析を恒久的に防止します。	MASTER_JTAG プリミティブをインスタンシエートし、eFUSE を無効化する復号器をプログラムする FPGA ロジックを構築します。
GTS	不正操作イベントに応答して出力を遮断し、デバイスからの情報流出を防ぎます。	STARTUPE3 プリミティブをインスタンシエートし、GTS アサートの正しい条件を決定する FPGA ロジックを構築します。
GSR	不正操作イベントに応答してユーザーのフリップフロップ ステートを初期状態に戻し、デバイス内の考えられる CT を効率的に消去します。	STARTUPE3 プリミティブをインスタンシエートし、GSR アサートの正しい条件を決定する FPGA ロジックを構築します。

注記:

1. UltraScale および UltraScale+ FPGA の新機能または強化機能です。

JTAG ポートの無効化 (能動的)

通常、攻撃者はシステムへ侵入しようとするとき、まず外部 JTAG ポートを狙います。UltraScale および UltraScale+ FPGA では、JTAG ポートをブロックするための能動的な方法が複数用意されており、JTAG ポートを一時的にも恒久的にもブロックできできます。以前の FPGA ファミリと同様に、1 つ目の方法は 1 つの BSCANE2 プリミティブを使用し、DISABLE_JTAG 属性を TRUE に設定してこのプリミティブをインスタンシエートする方法です。これにより、JTAG チェーン (外部 JTAG ポートと内部 MASTER_JTAG ポートの両方) と、FPGA デバイスと同じチェーン内のその他のすべてのデバイス (アップストリームとダウンストリーム両方) が切断されます。次に、UltraScale または UltraScale+ FPGA のデザインでの VHDL インスタンシエーション [参照 13] の例を示します。

```
BSCAN_U0 : BSCANE2
 generic map (
   DISABLE_JTAG => TRUE,
   JTAG_CHAIN   => 1 -- can be 1, 2, 3, or 4 depending on chain location
 )
 port map (
   CAPTURE => open,
   DRCK    => open,
   RESET   => open,
   RUNTEST => open,
   SEL     => open,
   SHIFT   => open,
   TCK     => tck_signal,
   TDI     => tdi_signal, TMS     => tms_signal, UPDATE  => open,
   TDO     => '1'
 );
```

注記: TCK、TDI、および TMS 信号は JTAG アクティビティの監視に使用できるため、上記ではこれらのポートのみが接続されています (「JTAG の監視 (検出)」を参照)。

外部 JTAG ポートをブロックする 2 つ目の方法は、UltraScale および UltraScale+ FPGA の新しいプリミティブ MASTER_JTAG をインスタンシエートする方法です。このプリミティブを使用して FPGA の外部 JTAG ピンよりも優先的にデバイス内部から JTAG ポートへのフルアクセスを許可します。この方法では、実質的には外部ピンが無効になり、FPGA ロジック内部用に使用されます。この方法は、認証が必要な場合や、ユーザー デザインをロードする間のみ JTAG の無効化が必要な場合に、外部 JTAG の無効化として効果的です。この方法を使用すると、ユーザー デザインによって JTAG を完全に制御できますが、外部 JTAG ピンのアクティビティは監視できません。MASTER_JTAG にはユーザー デザインでアクセスできるため、このプリミティブを使用し、不正操作イベントに応答して外部 JTAG ポートを恒久的に無効化する eFUSE ビットをプログラムできます。MASTER_JTAG を使用してデバイス内で eFUSE ビットをプログラムする方法の詳細は、ザイリンクス FAE にお問い合わせください。

FPGA ロジック デザインに、専用の JTAG ポートに接続された JTAG ベースのデバッグ ツールがある場合、JTAG チェーンを切断すると、これらのツールが機能しなくなります。FPGA のデバッグ段階では JTAG チェーンを保持し、以降の開発サイクルで JTAG ベースのデバッグ ツールが必要なくなったときにこれを切断できます。

JTAG チェーンを切断するオプションが有効になっている場合は、デバイスの JTAG コンフィギュレーションは使用できません。シリアル、シリアル ペリフェラル インターフェイス (SPI)、バイト ペリフェラル インターフェイス (BPI)、SelectMAP などのその他のコンフィギュレーション インターフェイスの 1 つを選択する必要があります [参照 3]。FPGA デバイスの実際のコンフィギュレーションが遅延している間、JTAG バウンダリスキャン ボード レベルのテストは正常に動作します。

JTAG の監視 (検出)

デバイス内から JTAG アクティビティを検出するには、BSCANE2 プリミティブで JTAG TCK、TDI、または TMS ラインのすべての組み合わせを監視します。外部 JTAG コマンドではこれらのラインをトグルする必要があるため、信号ラインの動作検出器によって JTAG アクティビティを検出できます。たとえば、TDI ラインの立ち上がりエッジを監視する場合は、図 7 の回路を使用できます。

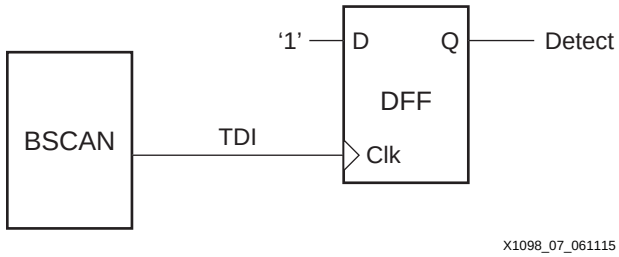

図 7: JTAG アクティビティ検出器の例

TDI ラインで立ち上がりエッジがラッチされると、デバイスがリコンフィギュレーションされるまで、つまり PROGRAM_B 入力がアサートされるまで、DFF の Detect 出力は 1 を保持します。この方法を拡張して、TCK、TDI、または TMS 信号ラインの立ち上がり/立ち下がりエッジを監視できます。JTAG 検出器 DFF のいずれかの出力が設定された場合には、これを使用して不正操作ペナルティを適用できます。

コンフィギュレーション メモリの整合性チェック (検出)

復号化されたビットストリームでコンフィギュレーションされた内部コンフィギュレーション メモリ セルが破損していると、FPGA が予期しない不正な動作をする可能性があります。このような破損は、コンフィギュレーション後の意図的な不正操作攻撃や、シングル イベント アップセット (SEU) などの意図しないイベントが原因で発生する可能性があります。SEM IP コア [参照 12] を使用すると、デザインのバックグランドでコンフィギュレーション データを継続的にリードバックしてビット反転を検出できます。SEM IP コアを使用すると、SEU の訂正も可能です。

固有識別子 (検出)

固有識別子 (UI) には、Device DNA とユーザー eFUSE の 2 種類あります。これらの UI をクローニング防止セキュリティ対策 (ユーザーのビットストリームを盗用して自分のデバイスをプログラムする行為への対策) として使用したり、UI の値に基づいて特定の機能を有効化または無効化 (アップグレード またはダウングレード) するために使用できます。

Device DNA は 96 ビットのデバイス固有のシリアル番号で構成されており、製造過程で FPGA 上のワンタイム プログラマブル (OTP) eFUSE ビットにザイリンクスが設定します (FPGA ロジックによるこの値への読み出しアクセスは DNA_PORTE2 プリミティブを介して実行し、JTAG を介した外部読み出しも可能)。ユーザー eFUSE にはユーザーが読み出し/書き込み可能な 32 ビットの OTP 領域があり、JTAG 経由でユーザーが設定します (FPGA ロジックによるこの値への読み出しアクセスは、EFUSE_USR プリミティブを介して実行)。セキュリティ上の目的に合わせて、これらの両方の UI を個別に使用することも、併用することも可能です。

注記: Device DNA またはユーザー eFUSE を使用する場合、固有の ID を持つことになりますが、暗号化のように高い機密性や認証機能 (AES-GCM など) は提供されません。クローニング対策には AES-GCM 暗号化が推奨されます。その上にこれらの UI を利用すれば、AT 全体にもう一重のセキュリティを追加できます。

これらの UI を使用して、ビットストリームを 1 つのデバイス (Device DNA の場合) または複数のデバイス (ユーザー eFUSE の場合) に関連付けられます。ユーザーが FPGA デザインに UI の比較機能を構築し、比較結果を用いて FPGA の動作を制御できます。たとえば、UI 比較でエラーが発生した場合にデザインの機能を停止あるいは制限できます。UI の使用例を次に示します。

1. セットアップ: 図 8 に示すように、JTAG を介して FPGA から UI 値を読み出し、UI 値からハッシュを生成し、FPGA からアクセスできるフラッシュ デバイスに格納します (堅牢な一方向関数を使用、キー タイプが最も安全)。

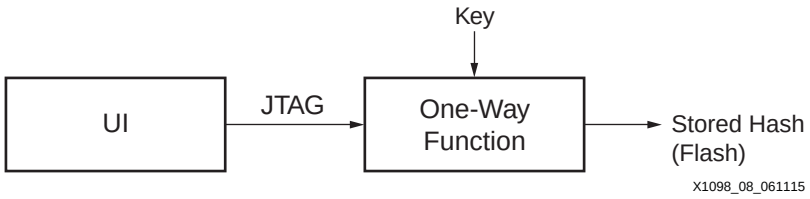

図 8: 秘密キーを使用した DNA 値の暗号化

図 8 に示す一方向関数のキーを、暗号化されたビットストリーム内に格納できます。ビットストリーム暗号化を使用しない場合、この方法でキーの機密性を確保するには、ビットストリームを複雑化する必要があります。

2. FPGA をコンフィギュレーションします。
3. 比較: FPGA デザインで DNA_PORT プリミティブまたは EFUSE_USR プリミティブのいずれか一方または両方から UI 値を読み出し、同じアルゴリズムを使用してチェックサムを計算します。その後、計算されたハッシュとフラッシュから読み出したハッシュを比較します。ハッシュが一致した場合は、デザインの動作が許可されます。

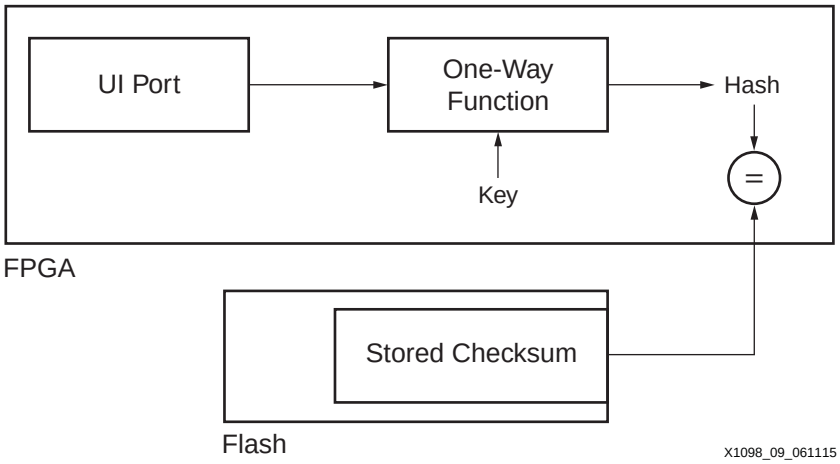

図 9: ハッシュの比較

これらの UI の詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 3] を参照してください。また、Spartan-3 ジェネレーション FPGA での Device DNA の詳細は、『Spartan-3 ジェネレーション FPGA を使用したセキュリティ ソリューション』(WP266) [参照 14] を参照してください。

オンチップ温度および電圧の監視/警告 (検出/応答)

攻撃者は、FPGA の通常動作時の電圧や温度を変更してデバイスに予想外の動作をさせることで、デバイスからデータを抽出したり、特定のセキュリティ機能を迂回するように仕掛ける可能性があります。たとえば、『Federal Information Processing Standards Publication (FIPS) 140-2Security Requirements For Cryptographic Modules』[参照 15] (暗号化モジュールに関するセキュリティ要件) には、「特に、暗号化モジュールは、指定された通常動作範囲外の動作温度や電圧の変動を監視し、適切に対応する必要がある」と規定されています。

このような要件を満たすために、オンチップの専用 IP ブロックである UltraScale および UltraScale+ FPGA のシステム モニター (SYSMONE1) を使用できます [参照 16]。SYSMONE1 は FPGA 内部にあるマルチチャネル ADC であり、オンチップ電源電圧、多数の I/O バンク電圧、オンチップ ダイ温度、および FPGA の外部ピンに供給される複数のユーザー アナログ電圧を監視できます。SYSMONE1 は FPGA ロジック デザインに簡単にインスタンシエートできます。このようなオンチップの監視機能はオフチップの場合よりも不正操作が難しく、安全です。また、SYSMONE1 には外部基準電圧または内部基準電圧 (V_{REF}) を使用できます。外部基準電圧の方が高精度ですが、内部基準電圧は攻撃者による不正操作が非常に難しいため、内部基準電圧の方が安全です。内部基準電圧と外部基準電圧の選択は外部ピンで制御されます。したがって、内部 V_{REF} が使用されていることを確認する場合は適切な SYSMONE2 ステータス レジスタを読み出します [参照 16]。

オンチップのパラメーターを監視する SYSMONE1 には警告の上限と下限を直接プログラムできます。さらに、FPGA ロジックを使用して、外部アナログ電圧不正操作ループや電圧センサーの出力といった外部電圧入力の警告の制限値を作成できます。警告信号のステータスをデザインまたはシステムで使用して、これらの信号がアクティブになった場合に実行する適切な一連の動作 (適切な不正操作ペナルティを特定するなど) を判断できます。アナログ入力の帯域幅は制限されています (最大入力周波数の詳細は、『UltraScale アーキテクチャ システム モニター ユーザー ガイド』(UG580) [参照 16] を参照)。

温度や電圧の急な変化を検出しなければならない場合、オフチップ ソリューションが必要になる可能性があります。検出に必要な帯域幅はシステム設計者が定義します。『Sorcerer's Apprentice Guide to Fault Attacks』[参照 17] では、チップを攻撃するさまざまな手法を説明しており、その 1 つに電源電圧の変があります。

連続した内部クロック ソース (検出)

ビットストリームの暗号化を使用する場合、STARTUPE3 ブロック プリミティブの出力となっている連続したクロック ソース CFGMCLK (コンフィギュレーション内部オシレーター クロック出力) を利用できます (PROGRAM_B インターセプトでの STARTUPE3 ブロックの VHDL インスタンシエーション例を参照)。このクロックは常にアクティブであるため、ユーザー クロックまたはその他の重要なユーザー信号の監視機能のベースとして使用できます。CFGMCLK は公称値 50MHz ± 15% [参照 18] [参照 19] から大幅に変動する可能性がありますが、重要なユーザー クロックやユーザー信号が有効な動作を続け、下限と上限の周波数範囲内でトグルしている (CFGMCLK の変動を考慮) ことを監視する上で非常に有効です 。重要なユーザー クロックやユーザー信号がこの範囲から外れた場合は、デザインが誤動作したか不正操作された可能性があり、適切な対策を講じることができます。

CFGMCLK は、FPGA ロジック内のその他のユーザー定義 AT 機能のクロック ソースとしても使用できます。外部クロック ソースを削除するだけでは AT 機能を停止できないことに注意が必要です。

外部 PROGRAM_B インターセプト (防止および検出)

コンフィギュレーション時に FPGA のすべてのメモリ エレメントが消去されるわけではありません。たとえば、FIFO を使用する GTH トランシーバーと GTY トランシーバーは、外部 PROGRAM_B ピンがアサートされた後でも状態を保持する場合があります (PROGRAM_B のアサートにより、FPGA がリセットされてビットストリームでリコンフィギュレーションされる)。攻撃者は PROGRAM_B ピンをアサートして、FPGA コンフィギュレーション後に消去されていないメモリ エレメントのコンテンツを破棄するように設計した独自のビットストリームで、ユーザー ビットストリームを置き換えることができます。STARTUP ブロックの PROGRAM_B 要求/肯定応答ペア (PREQ/PACK) などの PROGRAM_B インターセプト機能を使用すると、PROGRAM_B のアサート前にメモリ エレメントをデザインでまずクリアしたり、その他のハウスキーピング タスクを実行できるように、FPGA のリコンフィギュレーションを恒久的に遅延できます。

PROGRAM_B インターセプトの別の使用例として、動作中に PROGRAM_B がアサートすべきでないフィールド展開済みのシステムがあります。この場合、アサートの発生は不正操作が行われたことを示します。攻撃者がまず実行する操作の 1 つが、PROGRAM_B をアサートして FPGA のスタートアップ動作を観察することです。デザインで PREQ がアクティブになると、ペナルティが適用され、PACK をアサートすることで PROGRAM_B のアサートが可能になります。

PROGRAM_B インターセプト セキュリティ機能が有効になっている状態で暗号化されたビットストリームがロードされ、PROGRAM_B ピンが外部でアサートされる (または内部 IPROG コマンドが ICAPE3 に送信される) たびに、STARTUPE3 ブロックの PREQ 出力が High にアサートされます。PACK 入力を High (立ち上がりエッジを認識) に駆動するか、デバイスの電源を切って入れ直すまで、コンフィギュレーションが恒久的に延期されます。

適切なセキュリティ ジェネリックが設定され、PREQ/PACK 信号が接続されている STARTUPE3 ブロックの VHDL インスタンシエーション例を次に示します。

```
STARTUP_U0 : STARTUPE3
  generic map (
    PROG_USR => TRUE ) -- turn on PROGRAM_B intercept security feature
  port map (
    CFGCLK    => open,
    CFGMCLK   => cfgmclk_signal,
    DI        => open,
    EOS       => open,
    PREQ      => preq_signal, -- PROGRAM request to FPGA logic output
    DO        => (others => '0'),
    DTS       => (others => '0'),
    FCSBO     => '0',
    FCSBTS    => '0',
    GSR       => gsr_signal,
    GTS       => gts_signal,
    KEYCLEARB => keyclearb_signal,
    PACK      => pack_signal, -- PROGRAM acknowledge input (rising edge)
    USRCCLKO  => '0',
    USRCCLKTS => '0',
    USRDONEO  => '0',
    USRDONETS => '0'
);
```

注記: PROGRAM_B インターセプトでは、外部の試行だけでなく、プログラムによる *すべての*試行がインターセプトされます。MASTER_JTAG の IPROG や JPROGRAM も例外ではありません。

コンフィギュレーション メモリの消去 (応答/ペナルティ)

IPROG は ICAPE3 インターフェイス経由で送信される内部コマンドであり、FPGA コンフィギュレーション メモリ、すべてのフリップフロップ コンテンツ、およびキー拡張メモリを消去しますが、キー自体は消去しません。IPROG は外部 PROGRAM_B ピンのアサートに相当します。このコマンドは、コンフィギュレーション メモリ (コンフィギュレーション データ、ブロック RAM、およびフリップフロップ ステート) を効率的に消去します。

KEYCLEARB と IPROG の両方のペナルティが適用された後は、既存のビットストリームを FPGA で復号化できなくなるため、FPGA が動作不可能な状態になります。暗号化ビットストリームでデバイスをコンフィギュレーションできない場合、不正操作イベントが発生したことを示します。この時点で、デザインでは、KEYCLEARB ペナルティと IPROG ペナルティの後で暗号化されていないビットストリームをロードし、CT を流出させずに一部の基本機能を利用できます。当然ながら、eFUSE ベースのキーを使用しており、cfg_aes_only eFUSE がプログラムされている場合は、暗号化されていないビットストリームはロードできません。

コンフィギュレーション エンジンに IPROG コマンドを送信するには、デザインで ICAPE3 プリミティブをインスタンシエートして、適切なコマンド シーケンスを ICAPE3 に書き込む必要があります。詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 3] の「IPROG リコンフィギュレーション」を参照してください。

キーの俊敏性 (応答)

キーの俊敏性とは、FPGA ロジックによって BBRAM 内の AES ビットストリーム復号化キーをアップデートまたは変更できることです。これは、eFUSE ベースのキーには当てはまりません。

図 10 は、最初のキーが BBRAM に既にロードされた UltraScale/UltraScale+ FPGA を示しています。次に外部の暗号化ビットストリームをロードし、復号化してから、最初の FPGA ロジック デザインを起動して実行できます。

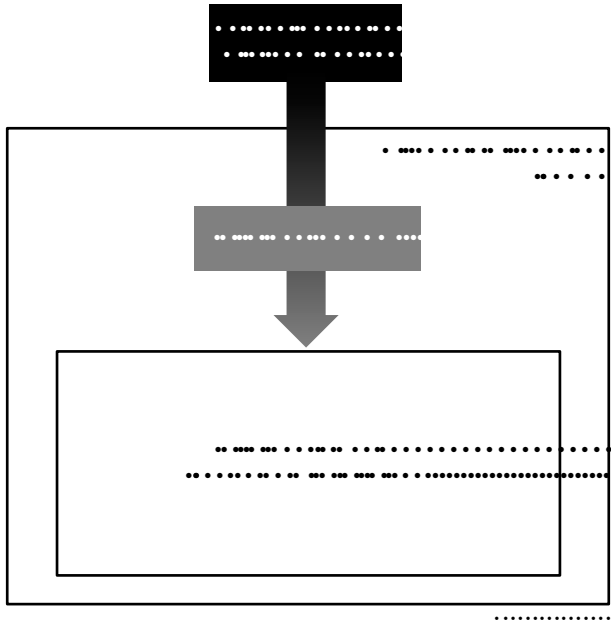

図 10: キーが BBRAM にロード された UltraScale/UltraScale+ FPGA

将来的な可能性として、セキュリティ侵害、適切な暗号化の運用、または新しいデザインのためにキー管理イベントが発生し、キーの変更やアップデートが必要になることがあります。何らかのセキュアなキー交換アルゴリズム (Diffie-Hellman プロトコルなどの FPGA ロジックで動作する IP コア) を使用して、インターネット経由の場合を含め、外部の暗号化キーを取り込むことができます。このキーを IP コアで復号化して BBRAM にロードし、MASTER_JTAG プリミティブを介して内部的に整合性チェックを実行できます (図 11)。

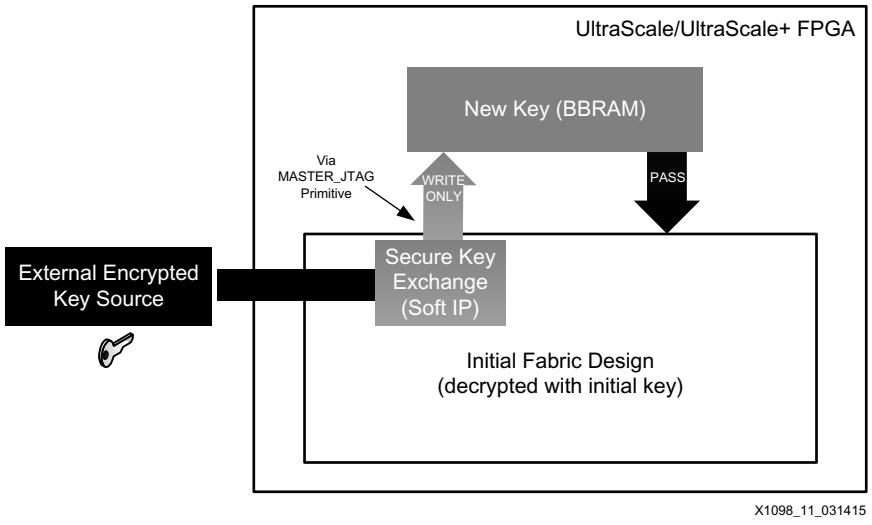

図 11: キー管理イベント

図 12 に示すように、外部の暗号化ビットストリーム (新しいキーで暗号化) をロードして復号化し、この新しいキーで復号化された新しいファブリック デザインを使用できます。

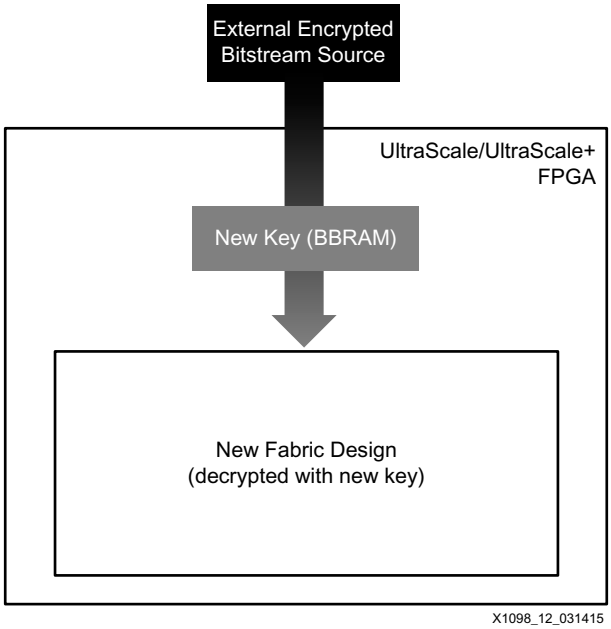

図 12: 新しいファブリック デザイン

この手法を使用すると、フィールドでキーをアップデートでき、安全な施設にボードやシステムを移動させる必要がありません。

BBRAM キーのゼロ化 (応答/ペナルティ)

FPGA の最も重要な部分は、ビットストリームの復号化に使用する AES-GCM キーです。攻撃者がこのキーを利用できるようになると、元のビットストリームのコンテンツが容易に解読されてしまいます。デザインを KEYCLEARB(1) 入力 (「外部 PROGRAM_B インターセプト (防止および検出)」の STARTUP ブロック プリミティブにある) に接続することで、内部または外部の不正操作に応答するペナルティとしてこの信号をアサートできます。BBRAM の 256 ビット キー、eFUSE キーのレジスタ付きコンテンツ、および復号化ブロック内の 1920 ビット拡張キーをゼロ化すると、オフチップの不揮発性メモリに格納されている暗号化ビットストリームが攻撃者にとって無用になります。UltraScale および UltraScale+ FPGA では、キー データの消去が検証されたかどうかを示す、つまりゼロ化を証明するコンフィギュレーション ステータス ビット(2) も使用できます。

KEYCLEARB のアサート判断およびその他の不正操作イベントへの応答は、必ずしも FPGA 内で実行されるとは限りません。システム レベルやモジュール レベルの不正操作バウンダリの侵害やなど、システム内の別の場所で発生した不正操作イベントに起因する可能性もあります。

キーがゼロ化されると、同じキーで再プログラムするか、暗号化されていないビットストリームに対して IPROG を使用してリコンフィギュレーションするまで [参照 3] (機能が制限される可能性がある)、FPGA デバイスが無用になります。必ず適切な条件でのみ KEYCLEARB 入力がアサートされるようにします。多くの場合、このような状態になったフィールドの装置は取り外して中央拠点や製造工場に送り返し、キーをロードして再び有効にする必要があります。

「コンフィギュレーション メモリの消去 (応答/ペナルティ)」で説明しているように、KEYCLEARB を IPROG コマンドと組み合わせて、不正操作イベントに応答してコンフィギュレーション メモリを消去することも可能です。KEYCLEARB は、IPROG コマンドの送信前にアサートする必要があります。つまり、KEYCLEARB がアサートされればすぐに IPROG コマンドを ICAPE3 に送信できます。

1. KEYCLEARB 信号は、不揮発性 eFUSE キーには影響を与えません。
2. これは、ICAP でアクセス可能なステータス レジスタのビット 21 です。

不揮発性 (eFUSE) 不正操作イベント ログ (応答)

UltraScale および UltraScale+ FPGA は新しい 128 ビットのユーザー eFUSE レジスタを備えています。このレジスタを使用すると、不揮発性領域に対するニーズに柔軟に対応できます。eFUSE レジスタ ビットのプログラムとリードバックには、JTAG 命令しか使用できません (詳細は、『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570) [参照 3] を参照)。これらの eFUSE ビットは、外部 JTAG ポートまたは内部 MASTER_JTAG を介してプログラムできます。MASTER_JTAG を使用してデバイス内で eFUSE ビットをプログラムする方法の詳細は、ザイリンクス FAE にお問い合わせください。

重要: eFUSE ビットをプログラムした後で別の eFUSE (書き込み不可) をプログラムでき、eFUSE レジスタ ビットがそれ以上プログラムされるのを防止 (「ドアをロック」) できます。そのため、攻撃者は不正操作ログ情報を上書きできません。

ビットストリーム復号器の恒久的な (eFUSE) 無効化 (防止/応答/ペナルティ)

攻撃者がサイドチャネル情報を収集するのを防ぐために、eFUSE をプログラムして AES-GCM 復号器を恒久的に無効化できます。eFUSE は、外部 JTAG ポートまたは内部 MASTER_JTAG を介してプログラムできます。MASTER_JTAG を使用してデバイス内で eFUSE ビットをプログラムする方法の詳細は、ザイリンクス FAE にお問い合わせください。この機能を不正操作イベントへの応答として使用したり、暗号化されたワンタイム コンフィギャラブル デバイスを作成するために使用できます。また、cfg_aes_only eFUSE ビットをプログラムすると、デバイスは再コンフィギュレーション不可となり、動作しません。

グローバル トライステート (応答/ペナルティ)

システム デザインによっては、重要なレッド情報が外部 FPGA ピンから読み出される可能性があります (暗号化モジュールなど)。不正操作イベントに応答して STARTUPE3 ブロックで GTS 入力がアサートされると (「外部 PROGRAM_B インターセプト (防止および検出)」のサンプル コードを参照)、すべての FPGA 出力が即座にハイ インピーダンス状態になり、FPGA 外部へのデータ フローが停止します。これは、レッド データのフローを早急に停止するために、IPROG または KEYCLEARB より前に講じる緊急措置です。

グローバル リセット (応答/ペナルティ)

ユーザー キー (AES BBRAM ビットストリーム復号化キーではない) などの重要なデータや機密パラメーターは FPGA ロジック レジスタに格納できます。不正操作イベントに応答して STARTUP ブロックで GSR 入力がアサートされる (「外部 PROGRAM_B インターセプト (防止および検出)」を参照) と、すべての FPGA レジスタ (フリップフロップ) がデフォルト ステートに戻ります。これは、FPGA 内のすべての機密データを早急に消去するために、KEYCLEARB と共に実行される緊急措置です。GSR がシフト レジスタ ルックアップ テーブル (SRL) やブロック RAM のコンテンツに影響することはありません。これらはデザインまたは IPROG コマンドで消去する必要があります。

不正操作防止のガイダンス

ここでは、前述したビルトイン シリコン AT 機能と組み合わせて、UltraScale および UltraScale+ FPGA を使用して不正操作防止デザインを作成するためのガイダンスおよび技術的ヒントを説明します。

必要な場合にのみ CT を読み込む (防止)

デザインを非クリティカル テクノロジ ブロックとクリティカル テクノロジ ブロックが含まれた複数のセクションに分割できる場合、デザインの非 CT 部分のみを常に配置し、必要な場合にのみ FPGA のパーシャル リコンフィギュレーション (PR) 機能 [参照 20]を使用して CT をロードできます。CT の役割が完了したら、PR 領域のブラック ボックス バージョンをロードして CT を消去できます。CT のパーシャル ビットストリームは FPGA ロジック内で任意のアルゴリズムで復号化できます。不正操作イベントに応答し、PR 領域と CT のキー (通常ブロック RAM または FPGA ロジック レジスタに格納される) の両方を消去できます。

例として、図 13 と図 14 に、FPGA、CPU、および外部メモリ デバイス (FPGA コンフィギュレーション、PR、CPU コード、および CPU ブート コード用) で構成された一般的なシステムを示します。図 13 の PR 領域 (「User CT Logic Region」と表示) は空の状態です。

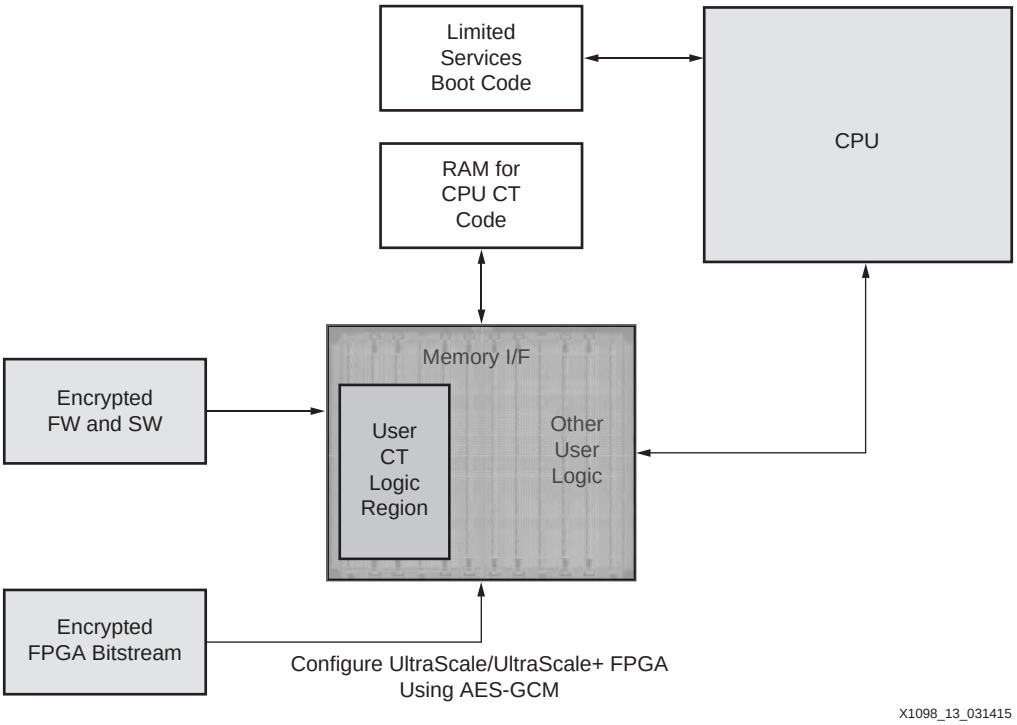

図 13: 使用モデル: システムの CT の保護

図 14 の場合、PR (ユーザー CT ロジック) によって、FPGA の PR 領域に CT ロジックが動的にロードされます。CT の役割が完了するか不正操作イベントが発生すると、PR モジュールのブラック ボックス バージョンをロードすることで、図 13 に示す状態に PR 領域を戻すことができます。

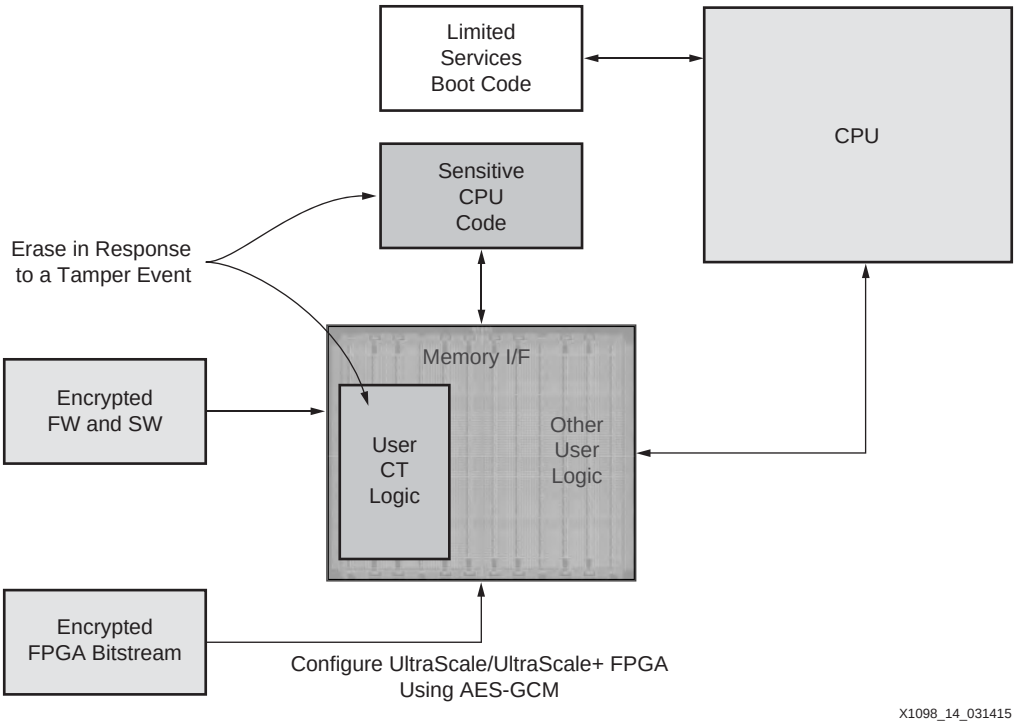

図 14: 使用モデル: システムの CT の保護 - 不正操作へ応答

注記: 不正操作イベントが発生した場合、外部の重要な CPU コード メモリは消去され、暗号化された、消去された、または重要度の低い外部メモリ コンテンツのみが残ります。

これらの例では、PR の実行に ICAPE3 が使用されます。ICAPE3 は信頼できるチャネルであるため、cfg_aes_only eFUSE ビットが設定された eFUSE キーを使用している場合でも、暗号化または非暗号化 PR ビットストリームが許可されます。FPGA ロジックにある、ユーザーが選択した復号化認証エンジンで復号化および認証が実行される、暗号化された認証済み PR ビットストリームを使用することを常に推奨します。セキュアな PR に関する参考資料として、『PRC/EPRC: パーシャル リコンフィギュレーションのデータ整合性とセキュリティ コントローラー』(XAPP887) [参照 20] を参照してください。

外部シャントによるキーの消去

BBRAM キーを消去する別の方法として、外部シャントを使用して V_{BATT} ラインをグランド接続する方法があります。KEYCLEARB などの能動的機能は FPGA に電源が投入されてコンフィギュレーションされた後にのみ使用可能なため、FPGA の主電源 (V_{CCINT} および V_{CCAUX}) が供給されていない場合に、この方法を用いることができます。たとえば、FPGA に主電源が供給される前にシステム レベルの不正操作イベントが検出された場合 (不正操作スイッチがアクティブになるなど)、FPGA の V_{BATT} ピンへの外部バッテリ電源ラインをオープンにして、何らかのトランジスタ シャントで V_{BATT} ピンをグランド接続できます。V_{BATT} ピンをグランドにシャントする前にバッテリ接続がオープンになるように、回路を設計する必要があります。また、別の方法として、抵抗を介してバッテリを V_{BATT} ピンに接続する方法もあります (V_{BATT} ピンの最大入力電流は 150nA)。適切な抵抗値を選択することで、バッテリから過度な電流フローを生じさせることなく V_{BATT} ピンをグランドに直接シャントできます。

FPGA に電源が投入されていない場合 (V_{CCINT} や V_{CCAUX} などが存在しない場合、ただし V_{BATT} を除く)、V_{BATT} ピンを –55°C でグランドに適切にシャントし、BBRAM に格納された AES キーが消去されるまでワースト ケースで最大 50ms かかります。

重要: セキュリティ レベルを高めながら流出を最小限に抑えるには、電圧ができるだけ低いバッテリを使用します。バッテリ電圧レベルの詳細は、UltraScale および UltraScale+ のアーキテクチャ データ シート [参照 18] [参照 19] [参照 21] [参照 22] を参照してください。

予防策としての BBRAM キーのゼロ化

キーのゼロ化の使用例として、予防策として用いる方法があります。ビットストリームをロードして復号化した後、システムを配備する前に BBRAM キーを意図的にゼロ化できます。この方法は、発射後のミサイルなど、配備後に電源を切って入れ直す必要のないシステムにのみ有効です。これは eFUSE キーにも使用でき、eFUSE キーの読み出しと書き込みが無効になっていない場合に限り、配備前に MASTER_JTAG を介してすべて 1 に書き換えます。

脆弱なキーや重複したキーは使用しない

ユーザー キーにはすべて 0、すべて 1、または反復パターンを使用しないようにします。可能な限り、キーを再利用しないようにします。キーの利用者を厳しく制御する必要があります。つまり、キー値を知る必要がある人のみがキー データにアクセスできるようにします。ランダムなソースを使用してキーを生成することが理想的です。脆弱なキーは使用しないようにします。たとえば、すべて 0 のランダム キーは理論的には可能です。Vivado ツールで AES-GCM キーを自動的に生成できますが、その場合、現在の日付と時刻に基づいた疑似ランダム プロセスが使用されます。最も安全なキーは、真のランダム プロセスによって作成されます。

暗号化システムにおいて、キー管理は非常に重要、かつおそらく最も複雑な要素です。キーに関するその他の役立つ情報として、NIST のキー管理ガイドライン [参照 23] を参照してください。

不正操作ステータス出力をシステムへ送信

不正操作イベントが発生した場合、ペナルティのアサートに加えて、デザインによってシステムに不正操作ステータス情報を送信できます (ユーザー eFUSE 領域にローカルにログを記録する代わり、またはこれに加えて)。今後の監査用に、システムにこの情報を保存できます。IPROG コマンド (不正操作ペナルティ) が実行される前にデータを転送するように設計する必要があります。

FPGA プローブ ポイントへのアクセス制限

攻撃されにくい FPGA を構築するには、多層アプローチが良い例です。CT を含むすべてのデバイス周囲に堅牢な不正操作バウンダリ (不正操作検出スイッチなど) を使用できます。たとえば、不正操作スイッチがアクティブになった場合に FPGA の V_{BATT} ラインでシャントをアクティブにできます。プリント回路基板には FPGA 信号用に埋め込み型のビアと配線を使用し、電源配線を埋め込み層内に配置して (アクセスが困難になる)、適切なデカップリングを行います (可能な場合は、埋め込みキャパシタンス テクノロジを使用)。JTAG バウンダリスキャン技術は、ボード レベルのファクトリ テストとして信頼性が確保されるため、量産ボードからテスト ポイントを削除する必要があります。詳細は、『UltraFast 設計手法ガイド (Vivado Design Suite 用) 』(UG949) [参照 24] および『UltraFast 設計手法チェックリスト』(XTP301) [参照 25] を参照してください。

まとめ

このアプリケーション ノートでは、UltraScale および UltraScale+ FPGA で現在利用できる AT 機能の概要を説明し、これらの AT 機能を効果的に使用する実際の例を紹介しました。設計サイクル初期に AT 機能を活用し、そのガイダンスに従うことで、不正操作防止 FPGA 対応のシステム デザインを実現できます。

1 つの AT 機能や手法で常に 100% の効果を挙げたりシステム全体のすべての AT ニーズを満たすことは不可能ですが、攻撃者の作業をできるだけ困難でコストの高いものにしたり、多層アプローチを採用することで、ほとんどの場合に満足できる結果を得ることができます。

FPGA を含めた新しい集積回路の開発およびテストに使用されるツールとテクノロジは日々進化し、改善されています。同時に、攻撃者が使用するツールも進化と改善を繰り返しているため、利用できる AT 機能や手法を把握しておくことが重要です。また、ザイリンクスは、これらの開発の最先端をリードし、現在および将来においてカスタマー IP を保護するために機能強化と新機能の開発に尽力しています。

参考資料

注記: 日本語版のバージョンは、英語版より古い場合があります。

1. ザイリンクス入門
   japan.xilinx.com/company/gettingstarted/index.htm
2. ザイリンクス Vivado Design Suite
   http://japan.xilinx.com/products/design-tools/vivado.html
3. 『UltraScale アーキテクチャ コンフィギュレーション ユーザー ガイド』(UG570: 英語版、日本語版)
4. 『デザインの安全性の確保』(WP365)
5. 『Virtex-6 および 7 シリーズ FPGA での不正操作防止デザインの開発』(XAPP1084: 英語版、日本語版)
6. Security Monitor 製品概要 japan.xilinx.com/publications/prod_mktg/CS1140_AD_SecMonIP_ProdBrf_Update_June_2012.pdf
7. 『Vivado Design Suite ユーザー ガイド: プログラムおよびデバッグ』(UG908: 英語版、日本語版)
8. 間接プログラムの概要 - SPI または BPI フラッシュ メモリ
   https://japan.xilinx.com/support/documentation/sw_manuals_j/xilinx11/pim_c_introduction_indirect_programming.htm
9. 『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835: 英語版、日本語版)
10. 『暗号化を使用して UltraScale FPGA ビットストリームを保護』(XAPP1267: 英語版、日本語版)
11. 『UltraScale FPGA RSA 認証およびサポート コンフィギュレーション モード』(XCN15038: 英語版、日本語版)
12. SEM (ソフト エラー軽減) コア
    japan.xilinx.com/products/intellectual-property/sem.html
13. 『UltraScale アーキテクチャ ライブラリ ガイド』(UG974: 英語版、日本語版)
14. 『Spartan-3 ジェネレーション FPGA を使用したセキュリティ ソリューション』(WP266)
15. 『Security Requirements for Cryptographic Modules』(FIPS PUB 140-2) www.nist.gov/itl/upload/fips1402.pdf
16. 『UltraScale アーキテクチャ システム モニター ユーザー ガイド』(UG580: 英語版、日本語版)
17. Hagai Bar-El、Hamid Choukri、David Naccache、Michael Tunstall、および Claire Whelan。『The Sorcerer's Apprentice Guide to Fault Attacks』
    http://eprint.iacr.org/2004/100.pdf
18. 『Kintex UltraScale FPGA データシート: DC 特性および AC スイッチ特性』(DS892: 英語版、日本語版)
19. 『Virtex UltraScale FPGA データシート: DC 特性および AC スイッチ特性』(DS893: 英語版、日本語版)
20. 『PRC/EPRC: パーシャル リコンフィギュレーションのデータ整合性とセキュリティ コントローラー』(XAPP887)
21. 『Kintex UltraScale+ FPGA データシート: DC 特性および AC スイッチ特性』(DS922: 英語版、日本語版)
22. 『Virtex UltraScale+ FPGA データシート: DC 特性および AC スイッチ特性』(DS923: 英語版、日本語版)

23. NIST キー管理ガイドライン
csrc.nist.gov/groups/ST/toolkit/key_management.html
24. 『UltraFast 設計手法ガイド (Vivado Design Suite 用)』(UG949: 英語版、日本語版)
25. 『UltraFast 設計手法チェックリスト』(XTP301: 英語版、日本語版)
26. Vivado Design Suite のパーシャル リコンフィギュレーション
japan.xilinx.com/products/design-tools/vivado/implementation/partial-reconfiguration.html
27. Xilinx ChipScope Pro およびシリアル I/O ツールキット
japan.xilinx.com/tools/cspro.htm
28. 『Secure Hash Standard』(FIPS PUB 180-2) csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf
29. ISE Design Suite のパーシャル リコンフィギュレーション
japan.xilinx.com/tools/partial-reconfiguration.htm
30. 『組み込み型マイクロコントローラーを使用するザイリンクスのインシステム プログラミング機能』(XAPP058: 英語版、日本語版)
31. 『エンベデッド JTAG ACE プレーヤー』(XAPP424)
32. Google での differential+power+analysis の検索結果
scholar.google.com/scholar?q=Differential+power+analysis&hl=en&as_sdt=0&as_vis=1&oi=scholart

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2017 年 2 月 22 日	1.2	初版

法的通知

本通知に基づいて貴殿または貴社 (本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ) に開示される情報 (以下「本情報」といいます) は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、およびすべて受領者の責任で (with all faults) という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず (商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、すべての保証および条件を負わない (否認する) ものとします。また、(2) ザイリンクスは、本情報 (貴殿または貴社による本情報の使用を含む) に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない (契約上、不法行為上 (過失の場合を含む)、その他のいかなる責任の法理によるかを問わない) ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害 (第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます) が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので https://japan.xilinx.com/legal.htm#tosで見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うことになります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。https://japan.xilinx.com/legal.htm#tos で見られるザイリンクスの販売条件を参照してください。本資料は英語版 (v1.2) 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。資料によっては英語版の更新に対応していないものがあります。日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com までお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。

自動車用のアプリケーションの免責条項

ザイリンクスの製品は、フェイルセーフとして設計されたり意図されてはおらず、また、フェイルセーフの動作を要求するアプリケーション (具体的には、(I) エアバッグの展開、(II) 車のコントロール (フェイルセーフまたは余剰性の機能 (余剰性を実行するためのザイリンクスの装置にソフトウェアを使用することは含まれません) および操作者がミスをした際の警告信号がある場合を除きます)、(III) 死亡や身体傷害を導く使用、に関するアプリケーション) を使用するために設計されたり意図されたりもしていません。顧客は、そのようなアプリケーションにザイリンクスの製品を使用する場合のリスクと責任を単独で負います。

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。