WP267 (v1.0) 2007 年 8 月 15 日

Spartan-3A/3AN/3A DSP FPGA の 高度なセキュリティ手法

著者 : Glenn Crow

FPGA は、新しいプロトコルと規格の組み込みおよびそのサポートを容易に実現し、迅速な市場への製品投入タイミングを保ちつつ、製品のカスタマイズを可能にします。インターネットと市場の世界規模への拡大に伴い、外部への製造委託がより一般的になったため、デザインのセキュリティがさらに大きな課題となっています。業界の専門誌に記載されるように、リバース エンジニアリング、クローニング、オーバービルディング、およびタンパリングはセキュリティ上、大きな問題となっています。専門家は、商品偽造によって年間数十億ドルの利益の損失が発生していると予測しています。Anti-counterfeiting Coalition (模倣対策連合) の報告によると、これらの偽造商品は経済を脅かし、消費者市場に世界規模の多大な影響を与えています。このホワイト ペーパーでは、デザインのセキュリティに対する最大の脅威を特定し、ザイリンクスの新しい、低コストの Spartan™-3A、Spartan-3AN、および Spartan-3A DSP FPGA の高度なセキュリティ オプションについて述べ、ユーザーの製品および利益をどのように守るかについて説明します。

最も頻繁に行われるセキュリティ侵害手法

現在、最も頻繁に使用されるセキュリティ侵害手法は、リバース エンジニアリング、オーバービルディング、クローニングおよびタンパリングです。

リバース エンジニアリングでは、窃取したデザインから競合製品を再現または再構築し、これらを市場で販売します。この場合、デザインをより短時間で構築できるため、研究開発 (R & D) コストを最小化できることが特徴となります。これは、電子産業界の創成期から最も一般的な侵害手法です。

オーバービルディングは、外部に製造委託するビジネス モデルで潜在的な懸念事項となっています。この場合、製品が不正に規定量より多く製造され、元の機器製造業者の許可なく、ほかの経路から販売されます。こうして製造された製品が市場で販売される際に、非常に多大な悪影響があることは明らかです。通常、「オーバービルト」された製品は低コストで販売され、製品の市場への投入も非常に早くなります。

クローニングは、同じまたは異なるラベル表示で、デザイン、IP (知的財産)、または製品の複製を製造することです。この場合、研究開発コストが発生せず、クローンした製品の市場への投入までの期間が大幅に削減されます。

タンパリングでは、権限を持たないサービスへアクセスするため、機密データを窃取するため、あるいはアプリケーションを妨害するために、オリジナルのデザインの変更および差し替えを試みます。タンパリングは、金融、防衛、特別なサービスを提供するオーディオ / メディア プロバイダにとって非常に大きな懸念事項となっています。

Spartan-3A/3AN/3A DSP FPGA のセキュリティ手法

FPGA を保護するセキュリティのレベルと種類は、コストに関係します。まず、突破されないセキュリティはないと認識することが重要です。結局のところ、システムへの侵入は完全には阻止できません。侵入者はデータやデザインが欲しければ、いかなる手段を用いてでもそれを入手します。これは単なるハッカーではなく、十分な資金のある政府または競合企業の場合があります。これを心に留めて、突破されないソリューションを作り出すのではなく、クローニング、オーバービルディング、リバース エンジニアリング、タンパリングに共通する脅威から十分に保護する方法を準備する必要があります。セキュリティについて考察する際、製品の要件に適切なものは何か考える必要があります。たとえば 10,000 ドルのコストがかかるシステムがあることを考えると、製品のコストが 10 ドルである場合、この価格帯のシステムに適切なレベルのセキュリティがあります。これは、FPGA ユーザー自身が評価する必要があります。評価終了後、それに基づいて、どの製品に対して、どのセキュリティ レベルをインプリメントするかを決定できます。ザイリンクスでは、セキュリティの問題を解決する多様なソリューションを提供しています。このソリューションには、単純なものから複雑なものまであります。Spartan-3 ジェネレーション FPGA 内でのセキュリティのインプリメンテーションで、より基本的と考えられるソリューションについては、WP266 :『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』で説明します。

このホワイト ペーパーでは、次のより高度な手法について説明します。

- ビットストリーム生成時のセキュリティ レベル
- アクティブ ディフェンス (JTAG バウンダリ スキャン)
- ビットストリームの検証 (CRC (巡回冗長検査))
- 高度なデータ変換

ザイリンクスでは、Spartan ファミリ以外でも、高度なソリューションを Virtex™ FPGA ファミリで提供しています。

ビットストリーム生成時のセキュリティ レベル

デザインのテストおよびデバッグ段階で、デザイン製品化後のメンテナンスまたは任意の点検のために ICAP (内部コンフィギュレーション アクセス ポート) または ChipScope™ Pro Analyzer コアをデザインに残すように決定できます。ChipScope Pro Analyzer などのソフトウェア機能の一部には、内部ロジックのステータスの読み出しにこれらのマクロが必要です。これは設計者にとっては便利ですが、セキュリティ ホールになる可能性があります。

Bitstream Generator は、NCD ファイルと呼ばれる物理的なインプリメンテーション ファイルの内容に基づいて、コンフィギュレーション ファイル (**.bit**) を作成します。BIT ファイルは、プログラム済み FPGA の動作を定義します。Bitstream Generator には多数のオプションがあり、これらの一部には通常使用されないものがあります。これらのオプションの 1 つにセキュリティ レベル設定があり、Bitstream Generator には、設定が 4 つあります。最初のレベルはデフォルトで、その他 3 つはオプションであり、追加のセキュリティを提供します。表 1 に示すように、リードバック動作はオプションで完全に無効にするか、ICAP を介した FPGA アプリケーションからの内部アクセス以外を無効にできます。

表 1 : ビットストリーム ジェネレータのセキュリティ レベル設定

セキュリティ レベル	説明
なし	デフォルト。すべてのコンフィギュレーションおよびリードバック機能へのアクセスが無制限
レベル 1	コンフィギュレーションまたは JTAG ポート (外部ピン) の両方からのリードバック機能がすべて無効。ICAP を介したリードバックは有効
レベル 2	全ポートでのリードバック動作がすべて無効
レベル 3	すべてのコンフィギュレーションおよび JTAG ポートからのコンフィギュレーションとリードバック機能がすべて無効。(リードバックとコンフィギュレーションに関しては) レベル 3 で発行および実行できる唯一のコマンドは REBOOT。このコマンドは、デバイスのコンフィギュレーションを消去する。コマンドがデバイス内から実行されることを除いて、これはデバイスで PROG_B ピンを有効にすることと機能的に同一

Bitstream Generator の全オプションの詳細については、UG332 :『Spartan-3 ジェネレーション コンフィギュレーション ユーザーガイド』を参照してください。上記のセキュリティ設定、レベル 1、2、および 3 を使用すると、次に示すソリューションで ICAP プリミティブを必要とするものが禁止されます。

アクティブ ディフェンス (JTAG)

JTAG インターフェイスを使用したデバイスは、リバース エンジニアリングの攻撃を受けやすいことが、共通の懸念事項として挙げられます。JTAG は、バウンダリ スキャン チェーンを用いたシステム、デバイス、IP、または標準的な製品のリバース エンジニアリングに使用できます。これらの攻撃には、十分な資金、知識、技術、装置、および時間が必要です。これらを実行する組織、競合、政府は、製品の動作を把握し、そのコスト削減、または機能の追加を試みる可能性が高いと考えられます。このセクションでは、デザインに機能を組み込み、JTAG を使用したリバース エンジニアリングを検出および防止する手法を説明します。

基本的に、JTAG バウンダリ スキャンは、PCB 上の I/O の接続テストおよびデバッグ用に設計され、後に、そのロジックをチップ内に含めるために採用されました。バウンダリ スキャンで INTEST コマンドを使用してデータをブロックするか、IC にシフトし、その後で IC をクロッキングして結果のデータをリードバックします。この動作で、IC またはブロックにアーキテクチャもしくはロジックを構築できます。一方で、図 1 に示すように、これはデザインまたはシステムをリバース エンジニアリングする手法の 1 つにもなります。したがって、JTAG ポートの不正使用は、ユーザーの、そして製品のセキュリティへの懸念事項となっています。

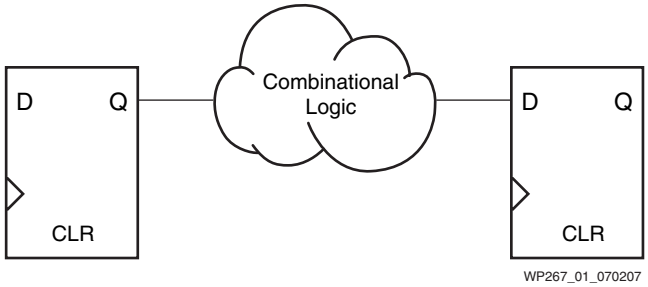

図 1 : 標準的なバウンダリ スキャン チェーン

Spartan-3A/3AN/3A DSP デバイスは JTAG 準拠であり、FPGA のコンフィギュレーションおよびリードバックが可能です。JTAG 準拠とは、JTAG ピンの使用を禁止できないことを意味しています。しかし、バウンダリ スキャン ブロックを用いて、JTAG ポートの不正使用を検知および阻止するセキュリティを組み込むことができます。

バウンダリ スキャン ブロック

バウンダリ スキャン信号へは、BSCAN_SPARTAN3A マクロ ブロック (図 2 参照) からアクセスできます。このブロックの単純なインスタンシエーションで、FPGA 内部から JTAG ピンの動作を監視できます。

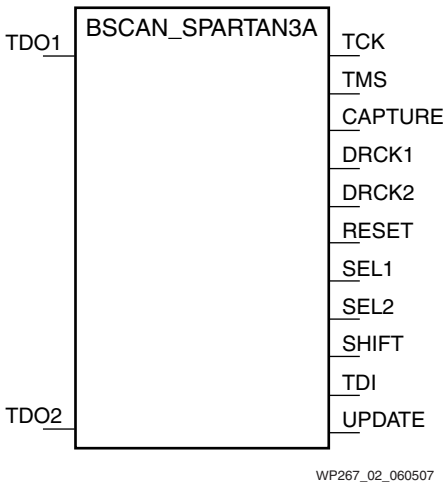

図 2 : BSCAN_SPARTAN3A

```
-- BSCAN_SPARTAN3A: Boundary Scan primitive for connecting internal
-- logic to JTAG interface
-- Spartan-3A
-- Xilinx HDL Libraries Guide, version 9.1i
BSCAN_SPARTAN3A_inst : BSCAN_SPARTAN3A
port map (
TCK => TCK,
TMS => TMS,
CAPTURE => CAPTURE,      -- CAPTURE output from TAP controller
DRCK1 => DRCK1,          -- Data register output for USER1 functions
DRCK2 => DRCK2,          -- Data register output for USER2 functions
RESET => RESET,          -- Reset output from TAP controller
SEL1 => SEL1,            -- USER1 active output
SEL2 => SEL2,            -- USER2 active output
SHIFT => SHIFT,          -- SHIFT output from TAP controller
TDI => TDI,              -- TDI output from TAP controller
UPDATE => UPDATE,        -- UPDATE output from TAP controller
TDO1 => TDO1,            -- Data input for USER1 function
TDO2 => TDO2,            -- Data input for USER2 function
);
-- End of BSCAN_SPARTAN3A_inst instantiation
```

バウンダリ スキャン ブロックがセキュリティを向上する方法

前述のように、このブロックから JTAG ポートの動作が内部で監視できます。ポートで動作が検出された場合、FPGA のコンフィギュレーションを完全に消去するか、選択した機能をバイパス / 阻止するようロジックを設計できます。ICAP は、Spartan-3A/3AN/3A DSP デバイスのコンフィギュレーションの消去に使用できます。ICAP の詳細は、UG332 :『Spartan-3 ジェネレーション コンフィギュレーション ユーザー ガイド』を参照してください。

図 3 に、主要ロジックおよび機能のバイパス例を示します。検出ロジックの出力で制御されるバイパス マルチプレクサを、主要入力機能に組み込みます。通常動作では、信号はロジックに送信されますが、JTAG 動作が検出されると、信号は切断され、設定値がロジックを介して送信されます。したがって INTEST 出力は、内部ロジックのリバースエンジニアリングでは完全に使用不可となります。

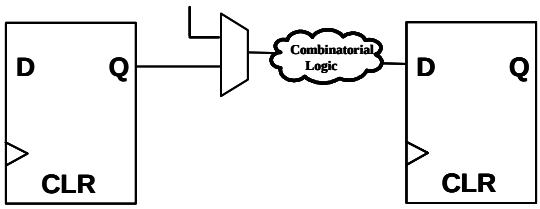

図 3 : ユーザー設定のバウンダリ スキャン チェーン

図 4 に示す「検出ロジック」は、1 つのゲートを用いた単純なものにもできますが、アプリケーションによってはさらに複雑なロジックが必要な場合があります (図 4)。

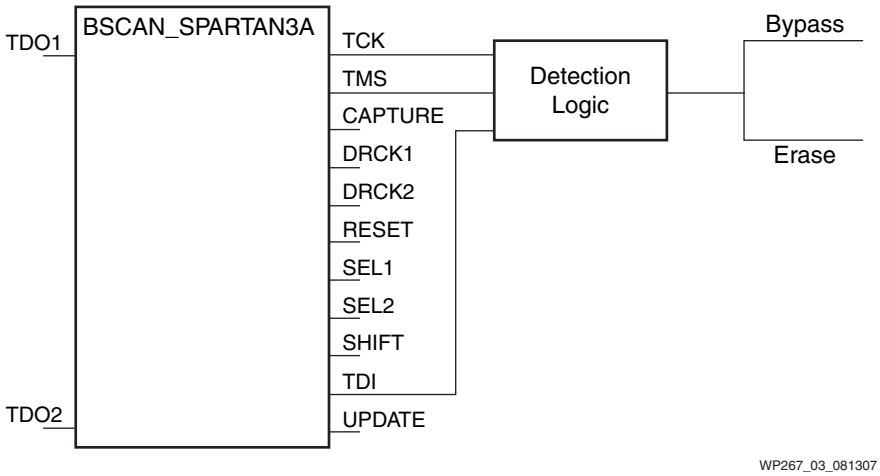

図 4 : 検出ロジック

フィールドでの JTAG によるセキュリティ更新および分析

ほとんどの場合、システムまたはデバイスが配置され、動作が開始すると、JTAG インターフェイスはアクセスまたは使用されません。しかし、これには常に例外があります。たとえば、システムのフィールドでの更新または診断が要求される場合は、JTAG ポートが必要です。JTAG ポートを介した不正なアクセスからデバイスを保護する検出セキュリティがインプリメントされていると、これによって許可されているアクセスが禁止されることがあります。デバイスをインプリメントする方法には複数あります。まず、検出ロジックが INTEST テスト命令でのみ有効になるよう設計することが挙げられます。この場合、JTAG は、BYPASS、IDCODE、USERCODE、および EXTEST などのほかのモードすべてで通常動作します。したがって、更新および診断の際、JTAG ポートへのフィールド アクセスが単純になります。

さらに複雑なセキュリティでは、たとえば、特定のアクセス ルーチンまたはコード シーケンスの監視用に検出ロジックを設計し、JTAG の通常動作モードへのアクセスを可能にできます。これは、フィールドでシステム機能の内部テストおよび検証に、INTEST 命令へのアクセスが必要な場合に便利です。ここでは、診断テストおよび更新が完了するまで JTAG 命令を使用可能にできます。更新が完了すると、更新された FPGA の再起動によって検出ロジックをリセットできます。診断のみを実行しているシステムでは、コード シーケンスが発行され、検出ロジックの監視が再開されます。

いずれの手法でも、JTAG ポートを介してセキュリティを危険にさらすことなく、必要なフィールド用タスクを実行できます。検出されたシーケンスが不正である場合、ICAP を使用してリセット (消去) します。

アクティブ ディフェンス ロジック リソースの要件

Spartan-3 ジェネレーションでは、数多くの機能がシリコンに組み込まれています。これらの機能に、JTAG ステート マシンおよび ICAP へのインターフェイス ロジックがあります。BSCAN_SPARTAN3A ブロックはシリコンに組み込まれているため、ロジック リソースが不要です。しかし、インスタンシエート済みの JTAG ブロックに接続されたユーザー ロジックは、ロジックおよびインターコネクト リソースを消費します。このロジックは、ユーザー ロジックおよび機能の複雑性によって 1 ロジックを使用する単純なものにも、10 以上のロジック セルを使用したものにもできます。

アクティブ ディフェンスの結論

小型の追加ロジックで、リバース エンジニアリングを検出し、それに対するセキュリティを向上できます。これは、Spartan-3A/3AN/3A DSP FPGA 設計時にバウンダリ スキャン ブロックおよび単純な検出ロジックをインスタンシエートすることで実現できます。

ビットストリームの検証

このセクションでは、コンフィギュレーション ビットストリームのタンパリング防止手法について説明します。デザインのタンパリングでは、権限を持たないサービスへアクセスするため、機密データを窃取するため、あるいはアプリケーションを妨害するために、オリジナルのデザインの変更および差し替えを試みます。通常動作中にデバイスのコンフィギュレーションを検証することで、改ざんされたコンフィギュレーションを検出でき、タンパリングの対処方法が決定可能です。検証回路のインプリメンテーションには多くの方法があります。ICAP および CRC を使用した単純な例を図 5 に示します。

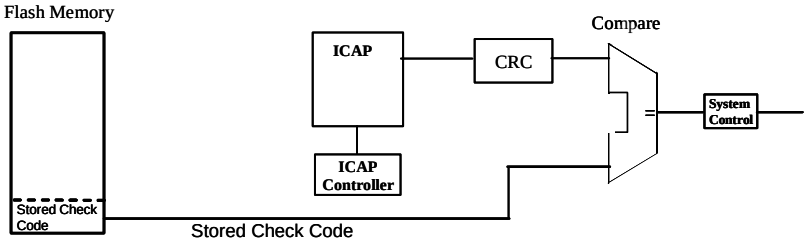

図 5 : ビットストリームの検証

ICAP ブロック

ICAP ブロックは、ファブリックおよび FPGA コンフィギュレーション コントローラ間のインターフェイスを有効にします。このブロック プリミティブは、FPGA にポートが組み込まれているため、インスタンシエーションに追加のロジック セルが不要である点でバウンダリスキャン ブロック プリミティブと類似しています。デバイスのコンフィギュレーション後、コンフィギュレーション ビットストリームを読み出すには、ICAP マクロをインスタンシエートする必要があります。ICAP ブロックは、Spartan-3A/3AN/3A DSP の MultiBoot 機能で共通に使用されます。ICAP が MultiBoot およびビットストリームの検証など、2 つ以上の機能で使用される場合、ICAP 接続時に信号の優先順位および制御を考慮する必要があります。これは、マルチプレクサのように単純にも、より複雑な調停ロジックにもできます。

図 6 に、ICAP プリミティブの回路図シンボルを、その次に VHDL インスタンシエーション テンプレートを示します。

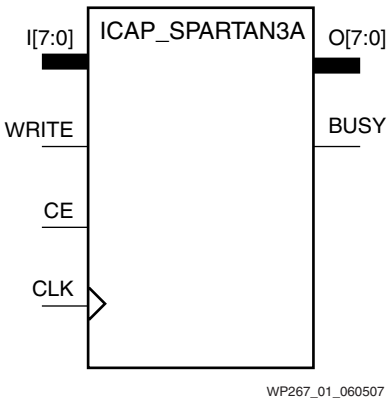

図 6 : ICAP_SPARTAN3A

```
-- ICAP_SPARTAN3A: Internal Configuration Access Port
--                 Spartan-3A
-- Xilinx HDL Libraries Guide, Version 9.1.3i

ICAP_SPARTAN3A_inst : ICAP_SPARTAN3A
port map (
   BUSY => BUSY,      -- Busy output
   0 => 0,            -- 8-bit data output
```

```
      CE => CE,          -- Clock enable input
      CLK => CLK,        -- Clock input
      I => I,            -- 8-bit data input
      WRITE => WRITE     -- Write input
   );

   -- End of ICAP_SPARTAN3A_inst instantiation
```

CRC (冗長巡回検査)

CRC は、通常、データ送受信時のエラー検出に使用されるチェック サムの一種です。これは、Bluetooth、Ethernet、USB、衛星通信、および FPGA のコンフィギュレーションに使用されています。ザイリンクスの FPGA には、デバイスがコンフィギュレーションをロードする際、ビットストリームを検証するセルフ チェック機能があります。CRC が算出され、生成されたビットストリーム内の保存された値と比較されます。2 つの値が同じ場合は、DONE ピンが High になり、コンフィギュレーションの成功を示します。

CRC アルゴリズムは単純ですが、データの完全性のチェックに非常に効果的です。また、ハッシュ アルゴリズムも、FPGA コンフィギュレーションの検証に使用できます。CRC またはハッシュ アルゴリズムの選択は、完全に設計者にゆだねられています。

単純なビットストリームの検証

ICAP ブロックは、デバイスのコンフィギュレーションの読み出しに使用されます。値は CRC に送信され、アクティブ値 (結果の値) が生成されます。その後、アクティブ値は、CRC の保存された値と比較されます。この例では、保存された値は空のコンフィギュレーション メモリ ロケーションにあります。2 つの値が同一である場合、コンフィギュレーションは正しいと判断されます。値が異なる場合は、デバイスがタンパリングされているので、設計者はその対応を決定します。一般的な対応手法は次のとおりです。

- コンフィギュレーションの再ロード

  ICAP ブロックを使用して、FPGA の消去および再コンフィギュレーションが可能です。主要なコンフィギュレーションがタンパリングされている場合は、FPGA の再コンフィギュレーションが繰り返し実行されます。

- 機能停止

  デザインは、完全に機能を停止します。これは、トライステート、ゲートを介したクロック、フリップフロップ クロック イネーブルなどのグローバル制御信号を使用して、Spartan FPGA に容易にインプリメント可能です。

デザイン要件に従って、動作が追加可能です。

ロジック リソース要件

組み込まれた ICAP ブロックを使用する場合、FPGA のロジック リソースは消費されません。さまざまな CRC およびハッシング アルゴリズムが選択可能で、これらには、数個のロジック セルを使用した単純なものから、数百のロジック セルを用いたより複雑なアルゴリズムまであります。

ビットストリーム検証のまとめ

データおよびアクセスの保護は、一部のデザインによっては、機能よりも重要です。単純なビットストリームの検証で、データ、アクセス、およびデザインの機能へのタンパリングによる攻撃が防止できます。

高度なデータ変換

Device DNA および保存されたチェック コードは外部に対する機密事項ではなく、誰でもこの情報にアクセス可能です。Device DNA セキュリティの詳細は、WP266 :『Spartan-3 FPGA を使用したセキュリティ ソリューション』を参照してください。

Device DNA デザイン レベル セキュリティにおける実際の機密事項は「セキュリティ アルゴリズム」です。一部のデザインでは、強引な攻撃に対してセキュリティを強化するため、Device DNA のデフォルトの 57 ビット以上がセキュリティ要件とされています。Device

DNA は、セキュリティ強化のためにビットを追加できるよう設計されています。Device DNA のビットが長くなると、強引な攻撃にさらに時間を要します。ここでの強引な攻撃とは、保存されたチェック コードを生成するため、クローニングやオーバービルディングによってセキュリティ アルゴリズムを見つけ出すことです。いずれ、強引な攻撃は時間がかかりすぎたり、不可能であったり、価値がないものとなります。強引な攻撃に要する時間の合計は、Device DNA および保存されたチェック コードのビット数の組み合わせによって左右されます。

図 7 に、Device DNA のデータ変換機能を使用してセキュリティを追加し、強引な攻撃に対応する手法を示します。この例では、Device DNA に 64,000 ビット を追加するようデザインが構築されています。このビットは、Spartan-3AN のユーザー Flash メモリに保存されます。また、これは、コンフィギュレーション メモリまたはシステム メモリにも同様に容易に保存できます。このデザインでは、Device DNA の後にソーターが挿入されます。ソーターはデマルチプレクサと、デマルチプレクサのセレクト ラインの制御用にデコードされたカウンタで単純に構成されています。デマルチプレクサの 1 つ目の出力はセキュリティ アルゴリズムにデータを送信し、2 つ目はビットをごみ箱に捨てます。この単純な回路によって Device DNA と保存されたチェック コードの関係が変更され、セキュリティ アルゴリズムへの強引な攻撃やリバース エンジニアリングがより困難になります。

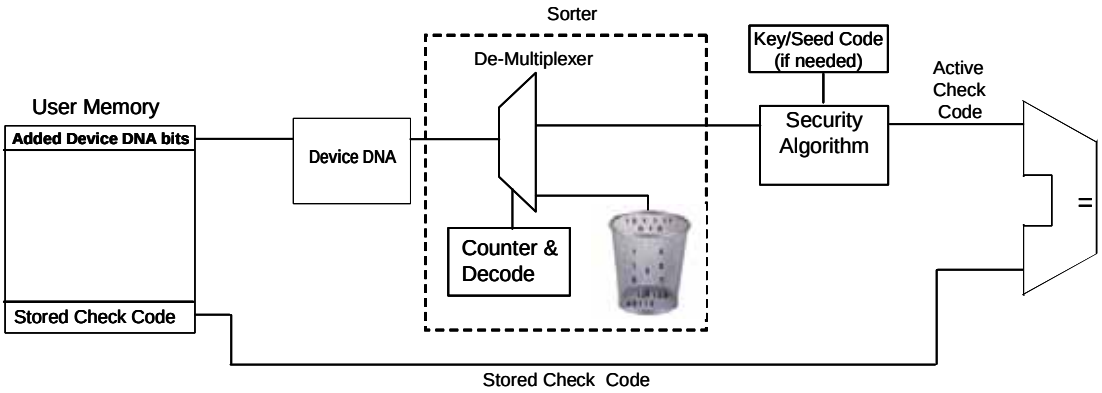

図 7 : Device DNA のデータ変換

保存されたチェック コードおよびアルゴリズム制御における高度なデータ変換

データ変換を更に進めると、保存されたチェック コードを組み込むことができます。図 8 に、データ変換ソーターを拡張し、追加の Device DNA ビットおよび保存されたチェック コードを組み合わせる例を示します。この場合、クローニングまたはオーバービルディングを試みるものには、Device DNA の FPGA への読み出しのみしか確認できません。これによって、クローニングまたはオーバービルディングする際、最初に Device DNA、保存されたチェック コード、およびごみ箱に捨てられたビットを、次にセキュリティ アルゴリズムを、それぞれリバース エンジニアリングすることが困難になります。この例では、3 つ目のデマルチプレクサ出力が追加されることで、保存されたチェック コードが分離され、コンパレータに送信されます。

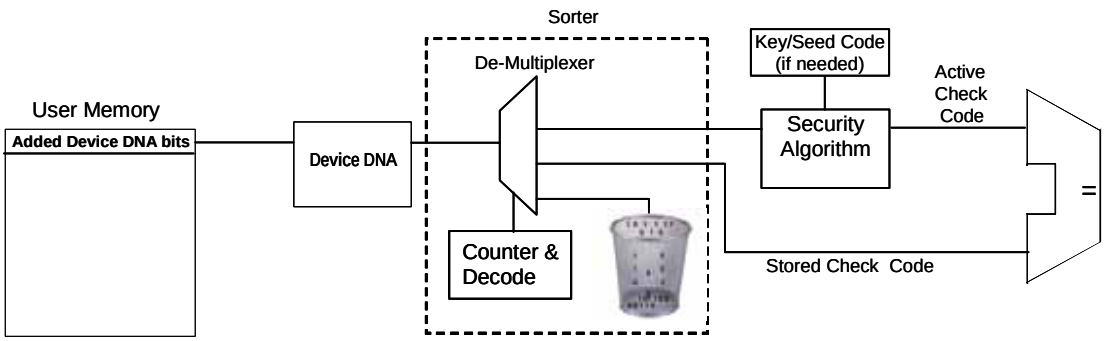

図 8 : 保存されたチェック コードのデータ変換

さらに、このデータ変換では、図 9 に示すようにデマルチプレクサに 4 つ目の出力を追加し、これをセキュリティ アルゴリズムに直接接続できます。選択したセキュリティ アルゴリズムに基づいてシード値、セキュリティ キー、または、アルゴリズム自体を変更でき、これによってクローニングまたはオーバービルディングを防ぐセキュリティがさらに追加されます。また、このデータ変換では、FPGA のハードウェア デザインは 100% 同一のままですが、セキュリティ アルゴリズムは変更されます。この追加されたセキュリティ アルゴリズムの変更で、製造フローまたはフィールドでのデザイン セキュリティの更新が容易になります。

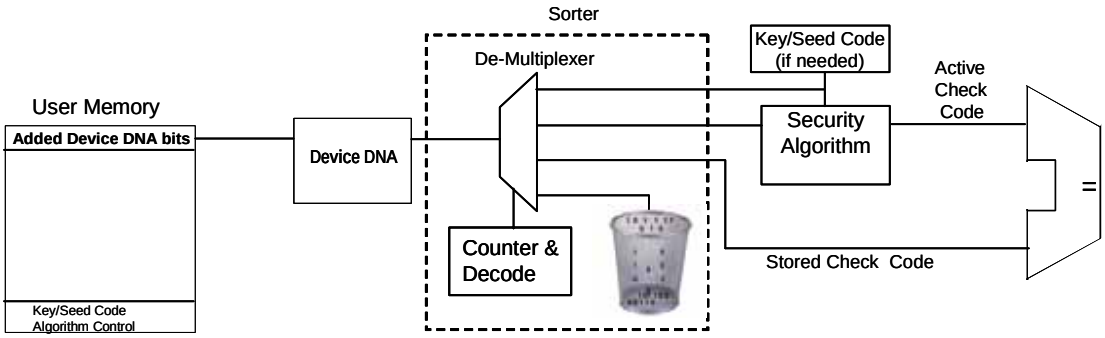

図 9 : デマルチプレクサに 4 つ目の出力を追加

ロジック リソースの要件

データ変換ソーターはデマルチプレクサおよびデマルチプレクサのセレクト ラインを制御するためにデコードされたカウンタで構成されています。このセレクト ラインは数十程度の少数のロジックセルでインプリメント可能です。

高度なデータ変換のまとめ

高度なデータ変換は、強引な攻撃を試みるクローニングおよびオーバービルディングから FPGA デザインを防御しながら、セキュリティの更新を単純かつ迅速に実現します。

まとめ

このホワイト ペーパーでは、クローニング、不正なオーバービルディング、リバース エンジニアリングおよびタンパリングからデザインまたはシステムを防ぐ、複数の高度な手法を紹介しました。説明した高度な手法の一部は、レイヤの手法でもあります。これは、多様な手法を統合すると同時に、複数の脆弱性を軽減します。

改定履歴

次の表に、この資料の改定履歴を示します。

日付	バージョン	改訂内容
2007/08/15	1.0	初版リリース