XAPP1239 (v1.0) 2015 年 4 月 15 日

暗号化を使用して 7 シリーズ FPGA のビットストリームを保護

著者 : Kyle Wilkinson

概要

このアプリケーションでは、ザイリンクスの Vivado® Design Suite で暗号化ビットストリームおよび暗号化キーを生成する簡単な手順について説明します。また、Vivado Design Suite を使用して、暗号化キーと暗号化ビットストリームをザイリンクスの 7 シリーズ FPGA へプログラムする手順も説明します。

はじめに

ザイリンクスの 7 シリーズ デバイスにはオンチップ AES (Advanced Encryption Standard) 復号化ロジックがあり、デザインを高度なセキュリティで保護します。暗号化された 7 シリーズ FPGA デザインは、意図しない FPGA 上で使用する目的で複製またはリバースエンジニアリングされることはありません。7 シリーズ FPGA の AES システムは、ソフトウェア ベースのビットストリーム暗号化機能と、オンチップのビットストリーム復号化機能、および暗号化キーを格納する専用メモリで構成されます。オプションとしてザイリンクスの Vivado ツールを使用して、暗号化キーと暗号化されたビットストリームを生成できます。真のランダムなソースからユーザーが生成したキーを使用することを推奨します。7 シリーズ デバイスでは、この暗号化キーをデバイス内部の専用 RAM (外部の小型バックアップ バッテリに接続された RAM - BBRAM) または eFUSE のいずれかに格納できます。暗号化キーは、JTAG ポートを介してのみデバイスにプログラムできます。7 シリーズ デバイスでは、コンフィギュレーション実行中に反対の処理、つまり取り込んでいるビットストリームの復号化が行われます。7 シリーズ FPGA の AES 暗号化ロジックは、256 ビットの暗号化キーを使用します。オンチップの AES 復号化ロジックは、ビットストリーム復号化以外の用途には使用できません。AES 復号化ロジックはユーザー デザインでは使用できず、コンフィギュレーション ビットストリーム以外のデータの復号化には使用できません。

AES (Advanced Encryption Standard) および認証

7 シリーズ FPGA の暗号化システムは、AES (Advanced Encryption Standard) 暗号化アルゴリズムを使用します。AES は、NIST (National Institute of Standards and Technology) および米国商務省が認証する公式規格です (http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf)。7 シリーズ FPGA の AES 暗号化システムは 256 ビットの暗号化キーを使用し (NIST が定める 128 ビットおよび 192 ビットの暗号化キーはインプリメントしない)、一度に 128 ビットのデータ ブロックを暗号化または復号化します。NIST によると、256 ビット キーの場合、キーの組み合わせは $1.1x10^{77}$ 通り考えられます。AES のような対称暗号化アルゴリズムでは、暗号化と復号化に同じキーが使用されます。したがって、データの安全性はキーの安全性に依存しています。

7 シリーズ FPGA でサポートされる AES は、ザイリンクスの Virtex®-6 デバイスでサポートされているものと同じです。(AES の有効性は検証されています。詳細は、http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html#2363 にある「Advanced Encryption Standard Algorithm Validation List」を参照してください。) 256 ビットの暗号化キーを eFUSE ビットまたはバックアップ バッテリ付き RAM のいずれかにユーザーが格納し、ザイリンクス ビットストリーム ライターで AES を用いてビットストリームを暗号化します。この暗号化機能では、CBC (Cipher Block Chaining) モードの 256 ビット AES 暗号を使用してビットストリームを暗号化できます。ユーザーが 128 ビットの初期ベクターと 256 ビットのキーを用意するか、ソフトウェアでランダム キーを選択します。Vivado Design Suite でキーを生成する方法は、真のランダム プロセスで固有のキーを生成する方法よりもセキュリティ レベルが低くなります (XAPP1084 [参照 1] 参照)。バックアップ バッテリ付き RAM に格納した AES キーを FPGA ロジックで消去するなど、一部のセキュリティ機能を利用するには、デバイスを暗号化ビットストリームでコンフィギュレーションする必要があります。

7 シリーズ デバイスは、オンチップのビットストリーム HMAC (keyed-Hash Message Authentication Code) もハードウェアにインプリメントしているため、AES 暗号化に加え、多重のセキュリティを実現しています。(http://www.nist.gov/itl/upload/FIPS-198-1_final.pdf にある FIPS PUB 198-1「HMAC Federal Information Processing Standards」を参照)。さらなるセキュリティによって復号化ビットストリームに厳密な認証が課され、シングル ビットの改ざんも許されないよう厳重に保護されます。AES と HMAC のキーが入手されない限り、ビットストリームの読み込み、改変、不正入手、複製が行われることはありません。AES は、デザインを複製やリバース エンジニアリングから保護するための基本的なセキュリティを提供します。一方 HMAC は、FPGA のコンフィギュレーション用に提供されたビットストリームが改変されておらず、そのまま読み込んで問題ないことを保証します。シングル ビットの反転を含め、ビットストリームに対するあらゆる改ざんを検出できます。

HMAC アルゴリズムで使用するキーを、ザイリンクス ツールで指定します。または、ツールでランダム キーを自動生成することも可能です。HMAC キーと AES キーは別のものです。ザイリンクス ツールは、キーと SHA (Secure Hash Algorithm) を使用して 256 ビットの MAC (Message Authentication Code) を生成します。MAC および HMAC キーは AES 暗号化されたビットストリームの一部として送信され、ビットストリームのデータが本物で改ざんされていないことを検証します。この認証機能は、あらゆる種類の制御ビットとデータ ビットを含む、ビットストリーム全体に適用されます。7 シリーズ FPGA でセキュリティ ソリューションを使用する場合は、常に HMAC と AES の両方を用いることになります。

暗号化ビットストリームのインプリメンテーションの概要

ザイリンクスの 7 シリーズ FPGA に暗号化されたデザインをインプリメントする場合の 6 つの基本手順を次に示します。

1. AES キーの格納場所を選択 : セキュリティ オプションに応じて BBRAM または eFUSE のいずれかを選択します (BBRAM と eFUSE のトレードオフについては XAPP1084 [参照 1] を参照)。
2. 指定した AES キーの格納場所に基づいて、ボード デザインのハードウェア要件を満たします。
3. Vivado Design Suite ツールを使用して AES キーを生成するか、またはソフトウェアおよび暗号化ビットストリームにユーザー独自の AES/HMAC キーを提供 (常にこちらの方法が高いセキュリティ レベルを提供) します。
4. JTAG インターフェイスを介して、FPGA に AES キーをプログラムします。
5. JTAG またはその他のコンフィギュレーション モード (SPI、BPI など) を使用して、FPGA に暗号化されたビット ファイルをプログラムし、DONE ピンがアサートされていることを確認します。
6. ハードウェア検証を実行して正しく動作していることを確認します。

ハードウェア ボード要件

暗号化デザイン フローを実行するにあたって、いくつか基本的なハードウェア要件があります。

- プログラム機能およびデバッグ機能 : FPGA へ接続する JTAG コネクタ。
- BBRAM キー ストレージ : V_{CCBATT} へのバッテリー (バッテリー電圧要件はデータシートを参照)
- eFUSE キー ストレージ : eFUSE にプログラムする前に、BBRAM でテストできるように V_{CCBATT} を V_{CCAUX} に接続することを推奨。

ソフトウェア要件

Vivado Design Suite 2014.3.1 またはそれ以降のバージョンが必要です。

AES キー ストレージ

AES キーの格納場所には、バッテリ バックアップ RAM (BBRAM) と eFUSE の 2 つの選択肢があります。ストレージ オプションとして BBRAM または eFUSE のいずれかを選択するときには、各オプションの長所/短所をよく理解した上でデザイン要件に最適なオプションを判断してください。それぞれの長所と短所の詳細は、この後のセクションで説明しています。これらのストレージ オプションに関するその他の情報は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) [参照 2] を参照してください。

BBRAM

暗号化キーを FPGA のバックアップ バッテリ付き RAM に格納した場合、暗号化キーのメモリ セルは揮発性であるため、継続的に電力を供給する必要があります。通常の動作中、これらのメモリ セルには補助電圧入力 (V_{CCAUX}) から電源が供給されますが、V_{CCAUX} が切断されたときにキーを維持できるように、独立した $_{CCBATT}$ 電源入力を推奨しています。このため、AES キーは、バックアップ バッテリーのあるボードでインシステム プログラムすることを推奨します。そうしないと、電源/バッテリーが切断されるとキーを失うことになります。BBRAM を格納場所に選択した場合の長所および短所を表 1 に示します。

表 1 : 格納場所としての BBRAM の長所/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	• 外部バッテリーが必要 • 多くのバッテリー ベンダーは高温/長期利用における動作仕様を定義していない

eFUSE

eFUSE は、コンフィギュレーション設定に使用する不揮発性のワンタイム プログラマブル技術です。ヒューズ リンクは、一定期間に大量の電流を流すことでプログラム (バーン、ブローとも呼ぶ) します。ユーザー プログラム可能な eFUSE のプログラムにはザイリンクス コンフィギュレーション ツールが使用できます。繰り返しになりますが、eFUSE ビットはワンタイム プログラマブル (OTP) です。一度プログラムすると再プログラムはできません。たとえばレジスタへのアクセスを無効にした場合、後から有効に変更できません。FPGA ロジックからアクセスできるのは FUSE_USER レジスタのみで、その他すべての eFUSE ビットは、FPGA ロジックからアクセスできません。eFUSE を格納場所に選択した場合の長所および短所を表 2 に示します。

表 2 : 格納場所としての eFUSE の長所/短所

長所	短所
• 外部バッテリーが不要 • CFG_AES_Only セキュリティ eFUSE ビットがセットされている場合は、eFUSE キーで暗号化されているビットストリームのみ FPGA へロード可能	• 恒久的 : キーの消去が不可 • BBRAM ソリューションよりもセキュリティレベルが低い

eFUSE レジスタ

7 シリーズ FPGA には、全部で 4 つの eFUSE レジスタ (FUSE_KEY、FUSE_CNTL、FUSE_USER、FUSE_DNA) があります。このアプリケーション ノートでは、FUSE_KEY、FUSE_CNTL、および FUSE_USER のみについて言及しています。表 3 に eFUSE レジスタの説明を示します。

表 3 : eFUSE レジスタの説明

レジスタ名	サイズ (ビット)	内容	説明
FUSE_KEY	256	ビットストリーム暗号化キー [0:255] (ビット 255 が最初にシフト)	AES ビットストリームの復号化で使用するキーを格納します。バックアップ バッテリ付きの SRAM にキーを格納する代わりに、eFUSE にキーを格納して使用できます。 7 シリーズ FPGA の復号化エンジンは、この AES キーを使用して暗号化ビットストリームを読み込みます。AES キーは、FUSE_CNTL レジスタの読み出し/書き込みアクセス ビットの設定に基づき、JTAG ポートを介して読み出しまたはプログラムが可能です。
FUSE_CNTL	14	制御ビット CNTL [13:0] (ビット 0 が最初にシフト)	キーの使用や eFUSE レジスタへの読み出し/書き込みアクセスを制御します。このレジスタは、JTAG ポートを介して読み出しまたはプログラムが可能です。
FUSE_USER	32	ユーザー定義 [31:0] (ビット 0 が最初にシフト)	32 ビットのユーザー定義コードを格納します。このレジスタは、EFUSE_USR プリミティブを使用して FPGA ロジックから読み出し可能です。EFUSE_USR プリミティブの詳細は、『7 シリーズ ライブラリ ガイド』を参照してください。 このコードは、FUSE_CNTL レジスタの読み出し/書き込みアクセス ビットの設定に基づき、JTAG ポートを介して読み出しまたはプログラムが可能です。

eFUSE 制御レジスタ (FUSE_CNTL) の説明

このレジスタには、ユーザー プログラマブル ビットが含まれています。表 4 で説明するこれらのビットを使用して、AES キーの使用法やその他の eFUSE レジスタの読み出し/書き込み保護を設定します。

表 4 : eFUSE 制御レジスタ ビットの説明

ビット インデックス 番号	FUSE_CNTL ビット名	eFUSE 制御ビットの説明	推奨設定
0	CFG_AES_Only	• AES 復号化のみを使用してコンフィギュレーションします。 • 1 にプログラムされている場合、eFUSE に格納された AES キーの使用を強制します。 • プログラムされていない (0) 場合、ビットストリームのセキュリティ オプションで AES 復号化の使用するか否かが選択されます。 **注意** : このビットが 1 にプログラムされている場合、AES キーが確認されない限りデバイスは使用できません。このビットがプログラムされている場合は、RMA (Return Material Authorization) による返品を受け付けていません。また Vivado による SPI/BPI フラッシュの間接プログラミング フローも使用できません。	なし (0 を保持し、顧客のセキュリティ要件を未決定にしておくことを推奨)
1	AES_Exclusive	• 1 にプログラムされている場合、外部コンフィギュレーション インターフェイスからのパーシャル リコンフィギュレーションが無効になります。 • プログラムされていない (0) 場合、外部インターフェイスからのパーシャル コンフィギュレーションが有効になりますが、キーを使用してパーシャル ビットストリームを暗号化する必要があります。 **注意** : このビットが 1 にプログラムされていると、RMA (Return Material Authorization) による返品は、デバイス解析およびデバッグの面で制限があります。このビットをプログラムする代わりに、ビットストリームのセキュリティ レベルを Level2 に設定することでも外部インターフェイスからのパーシャル コンフィギュレーションを無効にできます。	なし (0 を保持)
2	W_EN_B_Key_User	• キーおよびユーザー定義の eFUSE 値への書き込みイネーブルです (アクティブ Low)。 • 1 にプログラムされている場合、AES キーおよびユーザー定義値のビットのプログラムが無効になります。 **推奨** : キーをプログラムした後にこのビットプログラムすることで、eFUSE AES キー値への意図しない変更/破損を防止します。	あり (1 にプログラム)
3	R_EN_B_Key	• キーの読み出しイネーブルです (アクティブ Low)。 • 1 にプログラムされている場合、AES キーの読み出しと、AES キーおよびユーザー定義値のビットのプログラムが無効になります。 **注意** : eFUSE キーは JTAG インターフェイスを介して読み出すことができるため、キーのプログラム完了後は、このビットを未プログラム (0) のままにしないでください。	あり (1 にプログラム)

表 4 : eFUSE 制御レジスタ ビットの説明 (続き)

ビット インデックス 番号	FUSE_CNTL ビット名	eFUSE 制御ビットの説明	推奨設定
4	R_EN_B_User	• ユーザー定義の eFUSE 値への読み出しイネーブルです (アクティブ Low)。 • 1 にプログラムされている場合、JTAG を介すユーザー定義値の読み出しが無効になります。また、その影響を受けて、AES キーおよびユーザー定義値のビットのプログラムも無効になります。 **注記** : ユーザー定義値は、EFUSE_USR プリミティブを介して FPGA デザインから常時アクセス可能です。	なし (0 を保持)
5	W_EN_B_Cntl	• FUSE_CNTL eFUSE ビットの書き込みイネーブルです (アクティブ Low)。 • 1 にプログラムされている場合、FUSE_CNTL ビットのプログラムが無効になります。 **推奨** : FUSE_CNTL レジスタのビットをプログラムした後に、このビットを 1 にプログラムすることによって、FUSE_CNTL eFUSE ビットへの意図しない変更を防ぐことができます。	あり (1 にプログラム)

FUSE_CNTL[0] をプログラムしない場合 :

- 暗号化は、ビットストリーム オプションで無効/有効にできる
- eFUSE に格納されている AES キーを使用するか、バックアップ バッテリ付きの RAM に格納されている AES キーを使用するかをビットストリーム オプションで選択できる

注意 : FUSE_CNTL[0] をプログラムすると、外部コンフィギュレーション ポート経由で FPGA をコンフィギュレーションできるのは eFUSE キーで暗号化したビットストリームのみとなります。これにより、ザイリンクスのテスト ビットストリームや構築済みビットストリームを使用したデバイス コンフィギュレーションができなくなります。このため、ザイリンクスは FUSE_CNTL[0] ビットがプログラムされているデバイスに対する RMA 要求や Vivado による SPI/BPI フラッシュの間接プログラムをサポートしていません。

FUSE_CNTL[1] がプログラムされると、外部コンフィギュレーション ポートは、初期コンフィギュレーション後にコンフィギュレーション メモリへのアクセスがブロックされます。デバイスを再コンフィギュレーションするには、電源の再投入、JPROGRAM または IPROG コマンドの発行、PROGRAM_B ピンのパルスのいずれかが必要になります。

暗号化キーと暗号化ビットストリームの生成

Vivado ツールで提供されているビットストリーム ジェネレーターの write_bitstream は、暗号化したビットストリームと暗号化していないビットストリームのどちらも作成できます。ビットストリームを AES で暗号化するには、write_bitstream のプロパティでビットストリーム暗号化を有効にするオプションを選択します。ユーザーは、256 ビットのキーをビットストリーム ジェネレーターに入力できますが、Vivado ツールにランダム キーを生成させることも可能です。これにより、ビットストリーム ジェネレーターで暗号化ビットストリーム ファイル (.BIT) および暗号化キーファイル (.NKY) が生成されます。表 5 に、XDC ファイルで定義可能な write_bitstream のプロパティおよびそれらの説明を示します。

表 5 : Write_bitstream 暗号化プロパティ

Write_bitstream プロパティ	デフォルト値	設定可能な値	説明
BITSTREAM.ENCRYPTION.ENCRYPT	No	No、Yes	ビットストリームを暗号化します。
BITSTREAM.ENCRYPTION. ENCRYPTKEYSELECT	bbram	BBRAM、eFUSE	使用されている AES 暗号化キーの格納場所 (バックアップ バッテリ付き RAM (BBRAM) または eFUSE レジスタ) を示します。 **注記** : このプロパティは、Encrypt オプションが Yes に設定されている場合のみ有効です。
BITSTREAM.ENCRYPTION.HKEY	Pick	Pick、<16 進文字列>	HKEY は、ビットストリーム暗号化の HMAC 認証キーをセットします。7 シリーズ デバイスには、ハードウェアにオンチップのビットストリーム キー付き HMAC (Hash Message Authentication Code) アルゴリズムがインプリメントされており、AES 復号化のみの場合よりもセキュリティが強化されています。これらのデバイスに対するビットストリームの読み込み、変更、妨害、複製に AES キーと HMAC キーの両方が必要です。pick に設定すると、ビットストリーム ジェネレーターがランダムな値を選択します。このオプションを使用するには、最初に ENCRYPT オプションを Yes に設定する必要があります。
BITSTREAM.ENCRYPTION.KEY0	Pick	Pick、<16 進文字列>	Key0 は、ビットストリーム暗号化の AES 認証キーをセットします。pick に設定すると、ビットストリーム ジェネレーターがランダムな値を選択します。このオプションを使用するには、最初に ENCRYPT オプションを Yes に設定する必要があります。
BITSTREAM.ENCRYPTION.KEYFILE	None	<文字列>	入力暗号化ファイル名 (ファイル拡張子 .nky) を指定します。このオプションを使用するには、最初に ENCRYPT オプションを Yes に設定する必要があります。
BITSTREAM.ENCRYPTION.STARTCBC	Pick	Pick、<32 ビットの 16 進文字列>	暗号文ブロック連鎖 (CBC) の開始値を設定します。pick に設定すると、ランダムな値が選択されます。

BBAM キー ストレージとユーザー指定のカスタム AES キーを使用した場合の XDC ファイルの例を示します。これらの暗号化プロパティは、[Edit Device Properties] の GUI でも操作可能です。

```
24 #Encryption Settings
25
26 set_property BITSTREAM.ENCRYPTION.ENCRYPT YES [current_design]
27 set_property BITSTREAM.ENCRYPTION.ENCRYPTKEYSELECT BBRAM [current_design]
28 #set_property BITSTREAM.ENCRYPTION.ENCRYPTKEYSELECT eFUSE [current_design]
29 set_property BITSTREAM.ENCRYPTION.KEY0 256'h12345678ABCDDCBA1234578ABCDDCBA1234578
       ABCDDCBA1234578ABCDDCBA [current_design]
30
```

NKY ファイルの生成は、ビットストリーム生成と同時に実行されます。NKY ファイルは、ビット ファイルと同じ top_level 名を使用し、同じインプリメンテーション ディレクトリに配置されます。

NKY ファイルのフォーマットは次のとおりです。

```
KEY 0 <16 進文字列> (256 ビット AES キー )
```

例 : (top.nky)

```
Device xc7k325t;
Key 0 12345678ABCDDCBA12345678ABCDDCBA12345678ABCDDCBA12345678ABCDDCBA;
Key StartCBC 7115e9aa80085ea3ed65d26d3a8ab608;
Key HMAC d293d51c6058430262b05521f8f67279c9abce27d5fcafcf839bbe1af46713cc;
```

暗号化キーと暗号化されたビットストリームの読み込み

暗号化キーは、JTAG インターフェイスを介してのみデバイスに読み込むことができます。Vivado デバイス プログラマに NKY ファイルを入力し、サポートされているザイリンクスのプログラミング ケーブルを使用して JTAG 経由でデバイスに暗号化キーをプログラムできます。暗号化キーをプログラムする際、デバイスは特別なキー アクセス モードになります。このモードでは、暗号化キー専用 RAM とコンフィギュレーション メモリを含むすべての FPGA メモリがクリアされます。暗号化キーがプログラムされ、キー アクセス モードが終了すると、いかなる方法でもデバイスから暗号化キーを読み出すことはできず、キーを再プログラムするにはデバイス全体を初期化する必要があります。通常、このキー アクセス モードをユーザーが意識することはありません。暗号化キーは、V_{CCAUX} または V_{CCBATT} から電源が供給されるバッテリ バックアップ付き RAM (BBRAM) または eFUSE ビットのいずれかにプログラムできます。

BBRAM キーのプログラムには、Vivado Design Suite と JTAG ケーブルを使用します。

> **注記** : JTAG を介して BBRAM へ読み出し/書き込み動作を実行しようとすると、BBRAM の内容が消去され、アクセスが有効になる (キー アクセス モードへ遷移する) 前に FPGA のコンフィギュレーション全体が消去されます。

eFUSE キーのプログラムには、Vivado Design Suite と JTAG ケーブルを使用します。デバイス プログラミング サービスの詳細は、ザイリンクス販売代理店へお問い合わせください。

> **注記** : eFUSE ソリューションの場合、AES キーのインシステム プログラミング時に次の事項にも注意が必要です。
>
> - FPGA 内の電源ノイズを最小限に抑えるため、コンフィギュレーション済みデザインの FPGA は避ける、あるいはクリアしてください。
> - システムの電源ノイズを最小限に抑えるため、可能な場合にはボードレベルのシステム クロックを停止してください。

Vivado HW_Manager を使用して有効なハードウェア ターゲットへ接続した後、7 シリーズ FPGA を右クリックし、以前に選択したストレージ オプションに基づいて、BBRAM を使用する場合は [Program BBR Key]、eFUSE を使用する場合は [Program eFUSE Registers] を選択します (図 1 参照)。

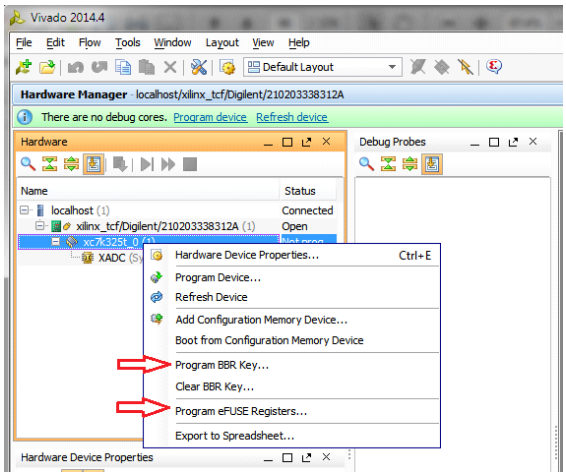

図 1 : Vivado HW Manager でキーのプログラムを選択

BBRAM キー

[Program BBR Key] を選択すると、プロジェクト ディレクトリで最近生成された NKY ファイルを参照できます。NKY ファイルを追加すると、図 2 に示すように [AES key] フィールドにキーの値が表示されます。この値を参照してキーの値をチェックし、デバイスにプログラムしようとしている正しいキーの値であることを確認できます。

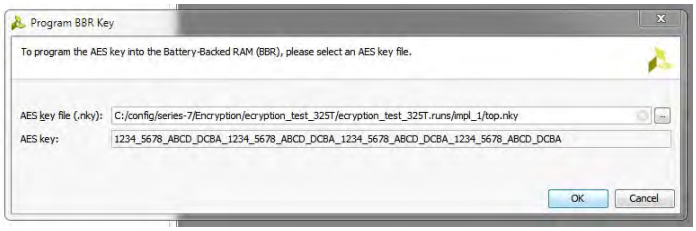

図 2 : BBRAM プログラミング GUI

JTAG を介して FPGA に NKY ファイルがすべてプログラムされると、TCL コンソールに次のように表示されます。

```
set_property ENCRYPTION.FILE
{C:/config/series-7/Encryption/ecryption_test_325T.runs/impl_1/top.nky} [get_property
PROGRAM.HW_BITSTREAM [lindex [get_hw_devices] 0]]
program_hw_devices -key {bbr} [lindex [get_hw_devices] 0]
INFO:[Labtools 27-3088] BBR Key
programmed:12345678ABCDDCBA12345678ABCDDCBA12345678ABCDDCBA12345678ABCDDCBA
INFO:[Labtools 27-3087] Key programming succeeded
INFO:[Labtools 27-3087] Key programming succeeded
```

eFUSE レジスタのプログラム

[Program eFUSE Registers] を選択すると、ウィザードが表示されて、プログラムする NKY ファイルおよび eFUSE レジスタの選択プロセスが指示されます。図 3 に、eFUSE プログラミングの GUI (AES Key Setup) を示します。

図 3 : eFUSE プログラミングの GUI - [AES Key Setup]

重要 : 7 シリーズ FPGA の場合、AES キーと FUSE_USER レジスタの下位 8 ビット [7:0] が同時にプログラムされます。したがって、AES キーをプログラムして、FUSE_USER [7:0] ビットに特定パターンを指定しない場合、これらのビットを後にプログラムすることはできません。同様に、FUSE_USER の下位ビットをプログラムして、AES キーをプログラムしない場合も後にキーをプログラムすることはできません。

推奨 : AES キーをプログラムする段階で、FUSE_USER レジスタの 32 ビットをすべてプログラムしてください。FUSE_CNTL レジスタビットの詳細は 5 ページの表 4 を参照してください。図 4 に eFUSE プログラミングの GUI (Control Register Setup) を示します。

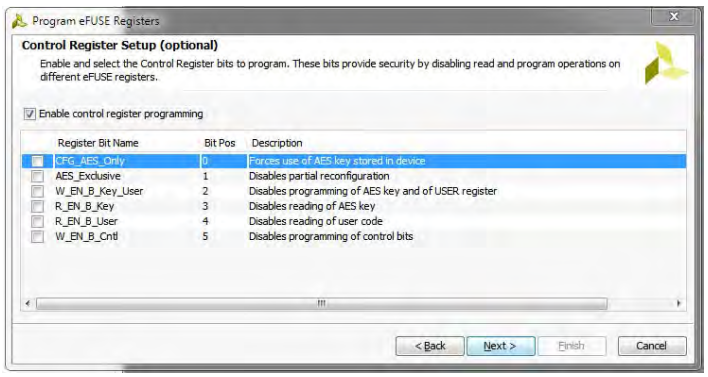

図 4 : eFUSE プログラミングの GUI - [Control Register Setup]

eFUSE レジスタをプログラムする場合の Tcl コマンドを次に示します。

- AES キーおよび FUSE_USER の全 32 ビット

```
program_hw_devices -key {efuse} -user_efuse {xxxxxxxx} [lindex [get_hw_devices] 0]
```

- FUSE_CNTL ビット

```
program_hw_devices -control_efuse {xxxxxx} [lindex [get_hw_devices] 0]
```

eFUSE レジスタのプログラミングが完了すると、[Hardware Device Properties] の [EFUSE] レジスタのドロップダウン メニューで FUSE_CNTL および FUSE_USER レジスタの値を確認できます (図 5 参照)。また、Tcl コンソールに次の Tcl コマンドを入力して確認することも可能です。

- FUSE_CNTL レジスタ

```
report_property [lindex [get_hw_device] 0] REGISTER.EFUSE.FUSE_CNTL
```

- FUSE_USER レジスタ

```
report_property [lindex [get_hw_device] 0] REGISTER.EFUSE.FUSE_USER
```

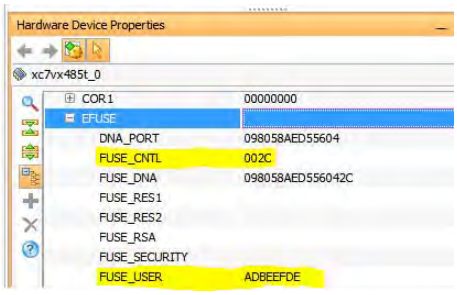

図 5 : [Hardware Device Properties] の [EFUSE] レジスタのドロップダウン メニュー

暗号化されたビットストリームの読み込み

デバイスに適切な暗号化キーをプログラムすると、暗号化ビットストリームを使用したデバイス コンフィギュレーションが可能になります。コンフィギュレーション後は、ビットストリームのセキュリティ設定にかかわらず、JTAG または SelectMAP リードバックによってコンフィギュレーション メモリを読み出すことはできません。デバイスに暗号化キーが読み込まれた状態のときに、暗号化していない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 (CFG_AES_ONLY ビットがプログラムされていない場合) するには、まず POR または PROGRAM_B をアサートしてコンフィギュレーション メモリをクリアする必要があります。この場合、暗号化キーは無視されます。また、暗号化していないビットストリームでコンフィギュレーションを行った後は、リードバックが可能です (ビットストリームのセキュリティ設定で許可されている場合のみ)。この場合でも、デバイスから暗号化キーを読み出すことはできないため、「トロイの木馬」ビットストリームを使用して 7 シリーズ FPGA の暗号化システムを無効にすることはできません。

暗号化を使用しても、コンフィギュレーションの方法にはほとんど影響はありません。7 シリーズ FPGA では、圧縮されたビットストリームと暗号化されたビットストリームの両方が作成可能です。暗号化ビットストリームは、JTAG、シリアル、SPI、BPI、SelectMAP、ICAPE2 のどのコンフィギュレーション インターフェイスでも使用できます。ただし、コンフィギュレーション方法によっては暗号化ビットストリームに若干の制約が生じたり、タイミングが変わることがあります。スレーブ SelectMAP と ICAPE2 インターフェイスでは、x8 バスでしか暗号化ビットストリームを使用できません (x16 および x32 スレーブ SelectMAP は使用できない)。マスター SelectMAP とマスター BPI インターフェイスでは x8 および x16 データ バスのどちらも暗号化ビットストリームをサポートしますが、x16 バス幅の場合、マスター CCLK 周波数が ConfigRate で設定した値の半分に低下するか、または ExtMasterCCLK_en が使用されているときは EMCCLK レートの半分に低下します。ConfigRate 周波数または外部 EMCCLK 周波数に基づいて CCLK がアップデートされる前に、DEC (AES 暗号化機能が有効) ビットが読み出されると、ビットストリームの冒頭はより低速な CCLK で開始します。

暗号化ビットストリームを使用した場合、外部コンフィギュレーション インターフェイスからパーシャル リコンフィギュレーションはできないため、デバイス全体をコンフィギュレーションする必要があります。コンフィギュレーション後にリコンフィギュレーションするには、PROGRAM_B ピンをトグルする、電源を再投入する、あるいは JPROGRAM または IPROG 命令を与える必要があります。暗号化をオンにすると、7 シリーズ FPGA ではフォールバック リコンフィギュレーションおよび IPROG リコンフィギュレーションが有効となります。また、ICAPE2 プリミティブを使用したリードバックが可能です。V_{CCBATT} または V_{CCAUX} が維持されている限り、これらのイベントによってキーがリセットされることはありません。暗号化ビットストリーム内のキーとデバイスに格納されたキーが一致しないとコンフィギュレーションがエラーとなり、INIT_B ピンが Low になって (フォールバックが有効の場合は High に戻る)、DONE ピンは Low のままになります。エラーが発生すると、Config_Status レジスタの HMAC_ERROR ビットでもわかります。

暗号化されたデザインが問題なく読み込まれたことをハードウェアで確認するには、DONE ピンが High にアサートされているかを確認するか、デザインが機能していることを視覚的に示すその他の方法 (LED、UART など) を使用して検証します。暗号化されたデザインが問題なく読み込まれたことをソフトウェアで確認するには、[Hardware Device Properties] にリストされている Config_Status レジスタを確認する方法があります。主なインジケーターは、ビット 1 (DECRYPTOR_ENABLE)、4 (EOS)、および 14 (DONE_PIN) です (図 6 参照)。

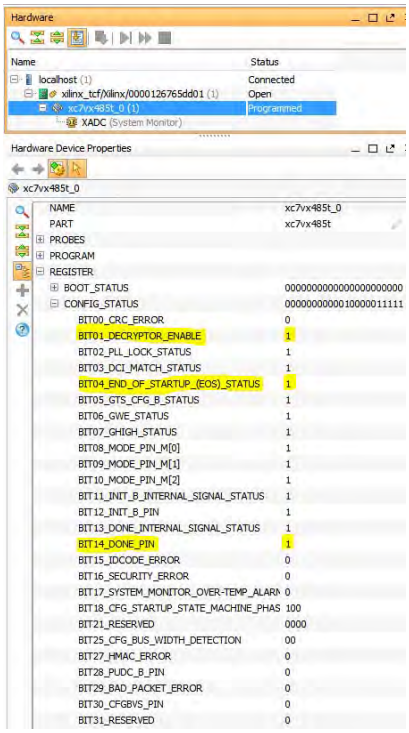

図 6 : デバイス ステータス レジスタ

ハードウェア検証

ほとんどの場合、AES キーが BBRAM または eFUSE ビットのいずれかに正しくプログラムされているかを検証することが望まれます。検証手順のチェックリストを次に示します。

1. Vivado Design Suite 2014.3.1 またはそれ以降のバージョンを使用してビットストリームを生成：暗号化されていないビットストリーム、ユーザー指定のキーで暗号化されたビットストリーム、すべて 1 のキーで暗号化されたビットストリーム、すべて 0 のキーで暗号化されたビットストリーム
2. 生成されたビットストリームを確認して、実行した暗号化を検証 : 暗号化されたビット ファイルと暗号化されていないビット ファイルの例は、14 ページの図 7 を参照してください。
3. eFUSE へのプログラミングが完了していない FPGA
   a. ハードウェアをチェック : Vivado デバイス プログラマを使用して FPGA へ接続し、JTAG を介して暗号化されていない BIT ファイルをダウンロードします。デザインは予想どおり機能しているか確認します。
   b. FPGA の復号化をテスト : すべて 0 のキーで暗号化された BIT ファイルをダウンロードします (eFUSE の場合)。
   c. 暗号化されたビットストリームのセキュリティをテスト : ユーザー指定のキーで暗号化された BIT ファイルをダウンロードします。コンフィギュレーション エラーが予想されます。
4. eFUSE キーおよびオプションをプログラム
   a. 電源を再投入して、上記テストのエラーがすべてクリアされ、FPGA がコンフィギュレーションされてない状態にする
   b. JTAG を介して AES キーをプログラム : eFUSE を使用する場合、まず最初に BBRAM を使用して手順 3b および 3c を実行し、有効性をチェックします。その後、eFUSE をプログラムして最終テストを行います。
   c. キーが読み込み不可であることを確認 : Vivado ツールを使用して、[Hardware Device] → [Property] → [Registers] → [eFUSE] → [FUSE_CNTL] でビット 3 が 1 にプログラムされていることを確認します。また、プログラムに選択したように、FUSE_CNTL のその他のビットがプログラムされていることを確認します。
5. プログラムされた eFUSE キーおよびオプションを備えた FPGA
   a. キーをテスト : ユーザー指定のキーで暗号化された BIT ファイルをダウンロードします。
   b. キーをテスト：すべて 0 のキーと関連付けられている暗号化された BIT ファイルをダウンロードします。コンフィギュレーション エラーが予想されます。
   c. キーをテスト : 暗号化されていない BIT ファイルをダウンロードします。結果は、セキュリティの設定によって異なります。

まとめ

このアプリケーション ノートでは、AES の暗号化と認証の規格について説明し、利用可能なそれぞれのキー格納オプションの長所と短所を記述しています。また、Vivado Design Suite ソフトウェアを使用して AES 暗号化キーおよび暗号化ビット ファイルを生成し、7 シリーズ FPGA にこれらのファイルをプログラムする方法について簡単に説明しています。

付録 A : 暗号化/非暗号化ビットストリーム

暗号化ビットストリームと非暗号化ビットストリームの違いを図 7 に示します。

暗号化ビットファイ

```
00000070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000090  00 00 00 BB 11 22 00 44 FF FF FF FF FF FF FF FF
000000A0  AA 99 55 66 20 00 00 00 30 00 C0 01 00 00 00 40
000000B0  30 00 A0 01 00 00 00 40 30 01 C0 01 00 00 00 00
000000C0  20 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00
000000D0  20 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00
000000E0  20 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00
000000F0  20 00 00 00 20 00 00 00 30 01 60 04 71 15 E9 AA
00000100  80 08 5E A3 ED 65 D2 6D 3A 8A B6 08 30 03 40 01
00000110  00 2B A6 58 B7 76 12 F1 90 AB 70 C8 12 9E 9B 08
00000120  34 CF 59 9A 9A D9 2F 83 81 DE D7 1B 01 19 65 10
00000130  9D DA 66 07 E7 3C 07 A5 AC A2 45 93 F4 7A F9 6E
00000140  B3 92 3A BD C7 F3 7A CF 0B A2 E7 E7 51 19 DA 8E
00000150  1E 63 E9 9B FC 6C 6E A3 A8 BE 27 85 8B B8 6E DB
00000160  7E 35 A2 DF D4 5A FA A7 57 C7 F9 54 7F C7 34 C3
00000170  03 CB 16 14 E8 0C C9 D5 AF 4B 72 9F BD 48 57 76
00000180  26 01 65 4D A3 CB 85 D1 FA 2A 40 8C 51 53 64 1F
00000190  BE C0 B5 A6 84 9B 9F 13 14 AF 16 DE BD 41 B3 36
000001A0  B3 B0 19 04 37 3F 1C 15 56 E2 E9 82 5C A6 00 41
000001B0  DD 6F 5E 3A 88 1E 2D 6A 3B 51 5C 88 D9 B0 46 33
000001C0  D2 CE B3 B5 0D B8 C2 E9 41 13 4F 6E 60 43 C4 A8
000001D0  85 AF 0A 5D 64 B5 01 85 21 CF C7 37 92 CC 23 F5
000001E0  E4 C1 15 60 F1 C6 49 F9 4A FA E9 74 2D 91 B8 3C
000001F0  EB BD 8F EC 2A C2 BF 11 0E 3C 73 0B 67 8B A7 EA
00000200  EE 73 46 97 2A 56 4C C7 18 B4 F2 29 E1 1E 38 38
```

非暗号化ビット ファイ

```
00000070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000080  FF FF FF FF 00 00 00 BB 11 22 00 44 FF FF FF FF
00000090  FF FF FF FF AA 99 55 66 20 00 00 00 30 02 20 01
000000A0  00 00 00 00 30 02 00 01 00 00 00 00 30 00 80 01
000000B0  00 00 00 00 20 00 00 00 30 00 80 01 00 00 00 07
000000C0  20 00 00 00 20 00 00 00 30 02 60 01 00 00 00 00
000000D0  30 01 20 01 02 20 3F E5 30 01 C0 01 00 00 00 00
000000E0  30 01 80 01 03 65 10 93 30 00 80 01 00 00 00 09
000000F0  20 00 00 00 30 00 C0 01 00 00 00 01 30 00 A0 01
00000100  00 00 01 01 30 00 C0 01 00 00 00 00 30 03 00 01
00000110  00 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00
00000120  20 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00
00000130  20 00 00 00 30 00 20 01 00 00 00 00 30 00 80 01
00000140  00 00 00 01 20 00 00 00 30 00 40 00 50 2B A5 20
00000150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000200  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

図 7 : 暗号化ビット ファイルと非暗号化ビット ファイル

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. 『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)
2. 『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470 : 英語版、日本語版)
3. 「BBR (バッテリー バックアップ RAM) の AES キーを使用」 (BBRAM のビデオ チュートリアル http://japan.xilinx.com/training/vivado/using-encryption-keys-with-bbram.htm

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
2015 年 4 月 15 日	1.0	初版

法的通知

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。