

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。


このアプリケーション・ノートでは、Cyclone® III LS デバイスのデザイン・セキュリティ機能について説明します。

このデザイン・セキュリティ機能は、業界標準の 256 ビット AES(Advanced Encryption Standard ; 高度暗号化規格) キー、および AES 暗号化アルゴリズム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復号化でき、デザインを下記の不正行為から保護することができます。

- コピー
- リバース・エンジニアリング
- 改ざん

次のデザイン機能により、ソリューションが安全になります。

- Cyclone III LS デバイスはコンフィギュレーション・ファイルのリードバックをサポートしません。これにより、コンフィギュレーション・ファイルが復号化された後で、それをリードバックしようとする試みは阻止されます。
- 256 ビットの AES キーを生成するには、2 つのユーザー定義の 256 ビット・シーケンスが必要です。
- この揮発性の 256 ビットは Cyclone III LS デバイ스에 格納され、外部バッテリーを取り外すとクリアされます。
- セキュリティ・モードに応じて、Cyclone III LS デバイスをコンフィギュレーションするには、同じセキュリティ・キーで暗号化されたコンフィギュレーション・ファイルを使用するか、またはボード・レベル・テストのために一般のコンフィギュレーション・ファイルを使用することができます。

 Cyclone III LS デバイスでサポートされるすべてのコンフィギュレーション手法がデザイン・セキュリティ機能をサポートするわけではありません。デザイン・セキュリティ機能をサポートするコンフィギュレーション手法について詳しくは、21 ページの「サポートされているコンフィギュレーション手法」を参照してください。

セキュリティ暗号化アルゴリズム

Cyclone III LS デバイスは、AES アルゴリズムを使用して 256 ビット AES キーによってコンフィギュレーション・データの復号化を行う、AES 復号化ブロックを備えています。暗号化されたデータを受信する前に、FPGA デバイスをコンフィギュレーションするには、256 ビットの AES キーが書き込まれる必要があります。

AES アルゴリズムは、データの暗号化と復号化を 256 ビットのブロック単位で行う共通ブロック暗号です。暗号化されたデータに対して、バイト置換、データ・ミキシング、データ・シフティング、およびキー追加を含む一連の変換が行われます。

デザイン・セキュリティ機能が使用されていない場合、AES 復号化ブロックはバイパスされます。Cyclone III LS の AES 実装は、連邦情報処理標準規格 FIPS-197 規格に準拠していることが確認されています。

-  AES アルゴリズムについて詳しくは、www.csrc.nist.gov からの「Federal Information Processing Standards Publication FIPS-197」または「AES Algorithm (Rijndael) Information」を参照してください。
-  Cyclone III LS デバイスの AES 検証について詳しくは、www.csrc.nist.gov で、NIST (標準技術局) による「Advanced Encryption Standard Algorithm Validation List」を参照してください。

揮発性キーのプログラミング手法

Cyclone III LS デバイスは、揮発性キー・ストレージを提供しています。揮発性キー・ストレージはバッテリー・バックアップを必要としますが、キーを更新することができます。表 1 に、揮発性キーの機能を示します。

表 1. 揮発性キーの機能

揮発性キーの機能	説明
鍵長	256 ビット
キーのプログラマビリティ	再プログラム可能、消去可能
外部バッテリー	必要
キーのプログラミング手法 (1)	オン・ボード (2)
デザインの保護	複製、リバース・エンジニアリング、および不正改ざんから保護 (3)

表 1 の注：

- (1) キーのプログラミングは JTAG インタフェースを介して実行されます。
- (2) オン・ボード・プログラミングとは、デバイスが別のプログラミング・システムの代わりに、ボード上でプログラムされているキー・プログラミング手法です。
- (3) デバイス・コアからの揮発性キー・クリアおよびキー・プログラム用の JTAG 命令がサポートされています。揮発性キー・クリアおよびキー・プログラムのための JTAG 命令については、「Cyclone III LS デバイス・ハンドブック Volume 1」の「IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices」の章を参照してください。

表 2 に、揮発性キーをオン・ボードでプログラミングする 2 つの方法について説明します。

表 2. キー・プログラミング方法

プログラミング方法	プログラミング・ツール
オン・ボード・プログラミング (プロトタイプ) (1)	EthernetBlaster 通信ケーブル、JTAG Technologies、ByteBlaster™ II ケーブル、USB-Blaster™ ダウンロード・ケーブル
オン・ボード・プログラミング (生産) (2)	イン・サーキット・テスト、JTAG Technologies

表 2 の注：

- (1) 初期に、特定の手法が正しく動作するかどうかを検証するために使用されます。
- (2) 量産に使用されます。

ハードウェアおよびソフトウェア要件

この項では、Cyclone III LS デザイン・セキュリティ機能のハードウェア要件およびソフトウェア要件について説明します。

ハードウェア要件


揮発性キー・プログラミングを成功させるには、デザイン・セキュリティ機能の電圧仕様に従う必要があります。表 3 に、揮発性キー・プログラミングの仕様を示します。


表 3. 揮発性キー・プログラミングの仕様

パラメータ	値
TCK 周期	「CycloneIII デバイス・ハンドブック Volume 2」の「 Cyclone III LS Device Data Sheet 」の章の「JTAG Specification」セクションを参照してください。
周囲温度	「CycloneIII デバイス・ハンドブック Volume 2」の「 Cyclone III LS Device Data Sheet 」での動作接合温度 (T) の仕様を参照してください。
電圧 (V_{CCBAT})	1.2 V (最小)、3.0 V (標準)、3.3 V (最大)

V_{CCBAT} は揮発性キー・ストレージの専用電源です。 V_{CCBAT} は、 V_{CCIO} や V_{CC} などのほかのオン・チップ電源から独立しています。 V_{CCBAT} はオン・チップ電源の状況に関係なく、揮発性レジスタに電源を供給し続けます。


V_{CCBAT} は、Cyclone III LS デバイスがパワー・オン・リセット (POR) から脱出するのに必要とされるパワーアップ電圧の 1 つです。 V_{CCBAT} が適切な電圧レベルにパワーアップされ、デバイスが POR から脱出してキー・プログラミングまたはコンフィギュレーションを開始することを確認しなければなりません。 V_{CCBAT} がキー・プログラミングの前に規定電圧に達することを保証するためには、100ms (スタンダード POR) または 12ms (ファスト POR) を待たなければなりません。

 リチウム・コイン型電池は V_{CCBAT} の電圧源として使用できます。リチウム・コイン型電池の例としては、BR1220 (-30°C ~ +80°C) および BR2477A (-40°C ~ +125°C) などがあります。

 Cyclone III LS デバイスが POR から脱出するのに必要なパワーアップ電圧について詳しくは、「CycloneIII デバイス・ハンドブック Volume 1」の「[Configuration, Design Security, and Remote System Upgrades in Cyclone III Devices](#)」の章の「POR Circuitry」の項を参照してください。

ソフトウェア要件

Cyclone III LS デバイスのデザイン・セキュリティ機能を使用するためには、Quartus® II ソフトウェア v9.0 SP2 以降を使用する必要があります。Quartus II ソフトウェアで Cyclone III LS デザイン・セキュリティ機能をイネーブルするには、ライセンス・ファイルが必要です。

 デザイン・セキュリティ機能のためのライセンス・ファイルを入手するには、www.altera.co.jp/support のアルテラ・テクニカル・サポートにお問い合わせください。

Quartus II ソフトウェアにおけるデザイン・セキュリティ機能のライセンス・ファイルのセットアップ方法

デザイン・セキュリティのライセンス・ファイルをセットアップするには、次の手順を実行してください。

1. www.altera.co.jp/support のアルテラ・テクニカル・サポートから、Cyclone III LS デザイン・セキュリティ機能をイネーブルするためのライセンス・ファイル入手します。
2. Quartus II ソフトウェアを起動します。
3. Tools メニューで、**License Setup** をクリックします。**Options** ダイアログ・ボックスで、**License Setup** オプションが表示されます。
4. **License file** 欄で、ライセンス・ファイルの位置およびファイル名を入力するか、あるいはライセンス・ファイルの位置に移動して、そのライセンス・ファイルを選択します。
5. **OK** をクリックします。

安全なコンフィギュレーション・フローの実装手順

安全なコンフィギュレーション・フローとは、デザイン・セキュリティ機能が使用されたコンフィギュレーション・フローのことを指します。安全なコンフィギュレーション・フローは通常のコンフィギュレーション・フローとは異なります。安全なコンフィギュレーション・フローを実装するには、次の手順を実行してください。

1. 暗号化キー・プログラミング・ファイルを生成して、コンフィギュレーション・データの暗号化を行います。

Quartus II ソフトウェアはユーザー定義の 256 ビット・シーケンスを用いて 256 ビット AES キーを生成します。このキーはコンフィギュレーション・ファイルの暗号化に使用されます。暗号化されたコンフィギュレーション・ファイルは、フラッシュ・メモリまたはコンフィギュレーション・デバイスなどの外部メモリに格納されます。

詳細については、5 ページの「[ステップ 1 : キー・ファイルの生成およびコンフィギュレーション・ファイルの暗号化](#)」を参照してください。

2. 256 ビット AES キーを Cyclone III LS デバイスにプログラムします。

詳細については、14 ページの「[ステップ 2 : 揮発性キーを Cyclone III LS デバイスにプログラム](#)」を参照してください。

3. Cyclone III LS デバイスをコンフィギュレーションします。

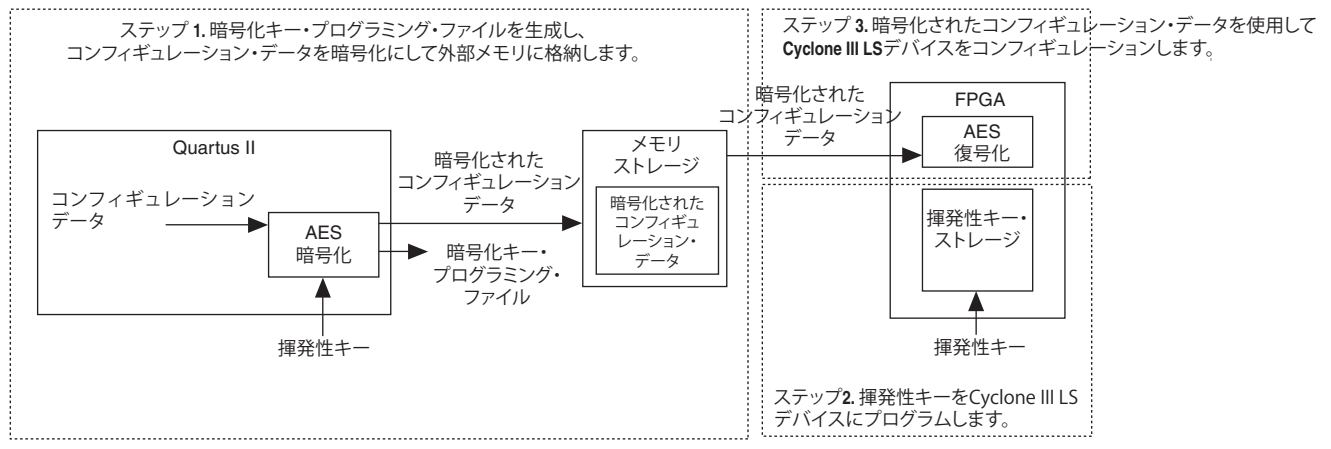
パワーアップ時、外部メモリ・ソースは暗号化されたコンフィギュレーション・ファイルを Cyclone III LS デバイスに送ります。Cyclone III LS デバイスは格納されたセキュリティ・キーを使用してファイルの復号化を実行し、そして暗号化さ

れていないデータは自身のコンフィギュレーションに使用されます。

詳細については、19 ページの「ステップ 3：暗号化されたコンフィギュレーション・ファイルによって Cyclone III LS デバイスをコンフィギュレーション」を参照してください。

図 1 に、Cyclone III LS デバイスの安全なコンフィギュレーション・フローを示します。

図 1. Cyclone III LS デバイスの安全なコンフィギュレーション・フロー



ステップ 1：キー・ファイルの生成およびコンフィギュレーション・ファイルの暗号化

Cyclone III LS デバイスのデザイン・セキュリティ機能を使用するには、Quartus II ソフトウェアでキーおよびそのキーによって暗号化されたコンフィギュレーション・ファイルを生成する必要があります。キーを生成するには、2 つのユーザー定義の 256 ビット・シーケンスを定義する必要があります。この 256 ビットの AES キーは 2 つの 256 ビットシーケンスによって生成されおり、Quartus II で生成されるコンフィギュレーション・ファイルには保存されません。そのため、このキーをほかの Cyclone III LS デバイスにコピーするのは不可能です。

 暗号化されたコンフィギュレーション・ファイルは復元機能をサポートしません。


Quartus II ソフトウェアでは、256 ビット・キーは **.ekp** ファイルとして生成されます。**.ekp** ファイルのフォーマットは、キー・プログラミングに使用されるハードウェアおよびシステムによって異なります。表 4 に、Quartus II ソフトウェアでサポートされるキー・ファイル・フォーマット、および各ファイル・フォーマットでサポートされるプログラミング・ツールを示します。

表 4. キー・ファイル・フォーマットおよびサポートされるプログラミング・ツール

キー・ファイル	プログラミング・ツール
JBC (.ekp) (1)	Quartus II ソフトウェア、EthernetBlaster 通信ケーブル、USB-Blaster ダウンロード・ケーブル、ByteBlaster II ケーブル、および ByteBlasterMV™ ダウンロード・ケーブル。
JEDEC STAPL (.jam) (2)	Quartus II ソフトウェア、EthernetBlaster 通信ケーブル、USB-Blaster ダウンロード・ケーブル、ByteBlaster II ケーブル、および ByteBlasterMV™ ダウンロード・ケーブル。 サードパーティのプログラミング・ベンダーおよび JTAG プログラマ・ベンダー。
Serial Vector Format (.svf) (2)	JTAG プログラマ・ベンダー

表 4 の注：

- (1) .ekp ファイルは、プログラミング・ファイルの変換時に Quartus II によって自動的に生成されます。
- (2) .jam および .svf ファイルを生成するには、Quartus II ソフトウェア・プログラマ内の **Create JAM, SVF, or ISC File** ダイアログ・ボックスを使用してください。

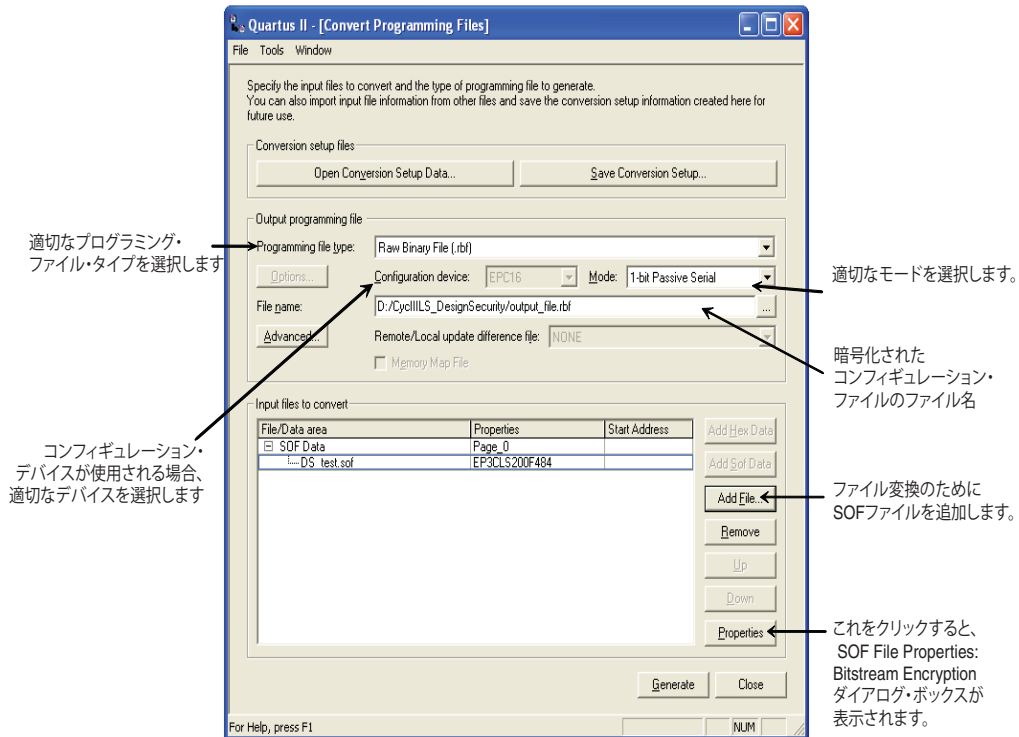
 アルテラは、キー・ファイルの機密性を保持することを推奨しています。

Quartus II ソフトウェアによるシングル・デバイス .ekp キー・ファイルの生成およびコンフィギュレーション・ファイルの暗号化

シングル・デバイス .ekp ファイルを生成し、コンフィギュレーション・ファイルを暗号化するには、次の手順を実行します。

1. 以下のオプションのいずれかを使用してデザインをコンパイルし、暗号化されていない SRAM オブジェクト・ファイル (.sof) を生成します。
 - Processing メニューで、**Start Compilation** をクリックします。
 - Processing メニューで、**Start** をポイントして、**Start Assembler** をクリックします。
2. File メニューで、**Convert Programming Files** をクリックします。**Convert Programming Files** ダイアログ・ボックスが表示されます ( 2)。

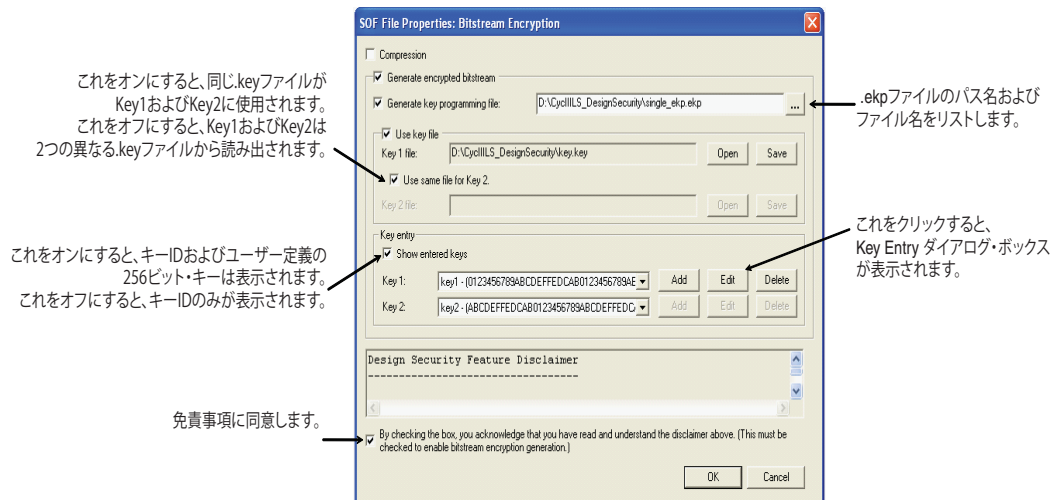
図 2. Convert Programming Files ダイアログ・ボックス



3. **Convert Programming Files** ダイアログ・ボックスで、以下の手順を実行します。
 - a. **Programming file type** リストからプログラミング・ファイル・タイプを選択します。
 - b. コンフィギュレーション・デバイスを使用する場合、**Configuration device** リストから適切なデバイスを選択します。
 - c. **Mode** リストからモードを選択します。
 - d. **File name** 欄にファイル名を入力するか、または該当するディレクトリに移動してそのファイルを選択します。
 - e. **Input files to convert** セクションで、**SOF Data** をクリックします。
 - f. **Add File** をクリックして、**Select Input File** ダイアログ・ボックスを開きます。
 - g. 暗号化されていない **.sof** ファイルに移動し、**Open** をクリックします。
 - h. **Input files to convert** セクションで、**.sof** ファイル名をクリックします。この欄がハイライトされます。
 - i. **Properties** をクリックします。**SOF Files Properties: Bitstream Encryption** ダイアログ・ボックスが表示されます (図 3)。
 - j. **SOF Files Properties: Bitstream Encryption** ダイアログ・ボックスで、**Generate encrypted bitstream** をオンにします。
 - k. **Generate key programming file** をオンにして、そして **.ekp** ファイルのパスおよびファイル名を入力するか、あるいは **<filename>.ekp** を参照して選択します。
 - l. **.key** ファイルまたは **Add** ボタンを使用して、プルダウン・リストから **key 1** および **key 2** を指定します。**key 1** および **key 2** は、**.ekp** キー・ファイルの生成およびコンフィギュレーション・ビットストリームの暗号化に使用される 2 つのユーザー定義の 256 ビット・シーケンスです。**Add** および **Edit** ボタンにより、**Key Entry** ダイアログ・ボックスは表示されます。詳細については、9 ページの「**.key** ファイルまたは **Key Entry** ダイアログ・ボックスから **Key 1** および **Key 2** を生成する方法」を参照してください。

Delete ボタンは、現在選択されているキーをプルダウン・リストから削除するのに使用されます (図 3)。
 - m. デザイン・セキュリティ機能に関する免責条項を読んでください。デザイン・セキュリティ機能の免責事項に同意する場合、確認ボックスをオンにしてください。
 - n. **OK** をクリックします。**<filename>.ekp** および暗号化されたコンフィギュレーション・ファイルは同じプロジェクト・ディレクトリに作成されます。

図 3. SOF Files Properties: Bitstream Encryption ダイアログ・ボックス



.key ファイルまたは Key Entry ダイアログ・ボックスから Key 1 および Key 2 を生成する方法

.key ファイルでは、ラインが「#」で始められない限り、各ラインは1つのユーザー定義の 256 ビット・シーケンスを表します。「#」記号は、コメントを示すために使用されます。有効なキー・ラインは次のフォーマットをしています (図 4)。

`<key identity><white space><user-defined 256-bit hexadecimal sequence>`


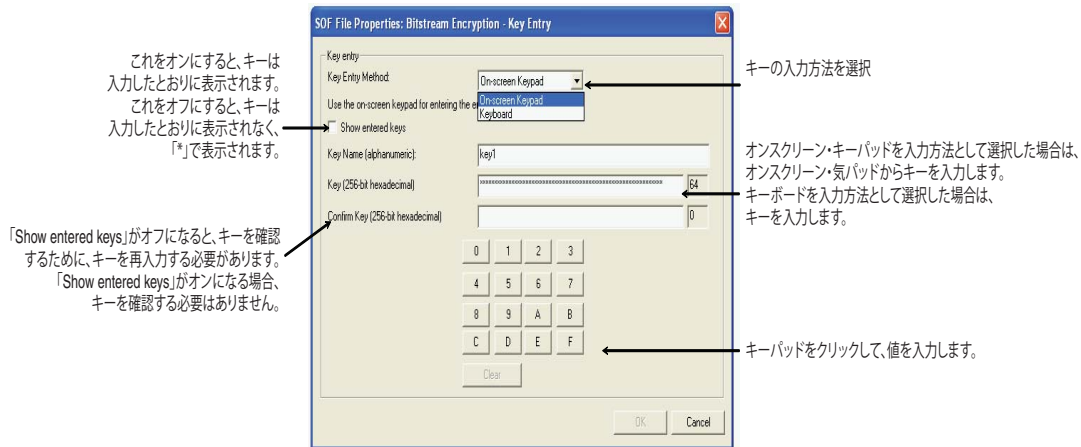
 このキー ID は、これらのキーを識別するための英数字です (キー・ファイルのエントリと同様)。


図 4. .key ファイルの例



Key entry セクション (図 3) で、プルダウン・リストから **Key 1** および **Key 2** を選択するか、あるいは **Key Entry** ダイアログ・ボックス (図 5) を開いて暗号化キーを入力します。プルダウン・リストにおけるキーは **.key** として保存できます。キーを保存して、標準の **File** ダイアログ・ボックスを表示させるには、**Use key file** セクションで対応する **Save** ボタンをクリックする必要があります。プルダウン・リストにおけるすべてのキーは選択された **.key** ファイル、あるいは作成された **.key** ファイルに保存されます。

図 5. Key Entry ダイアログ・ボックス



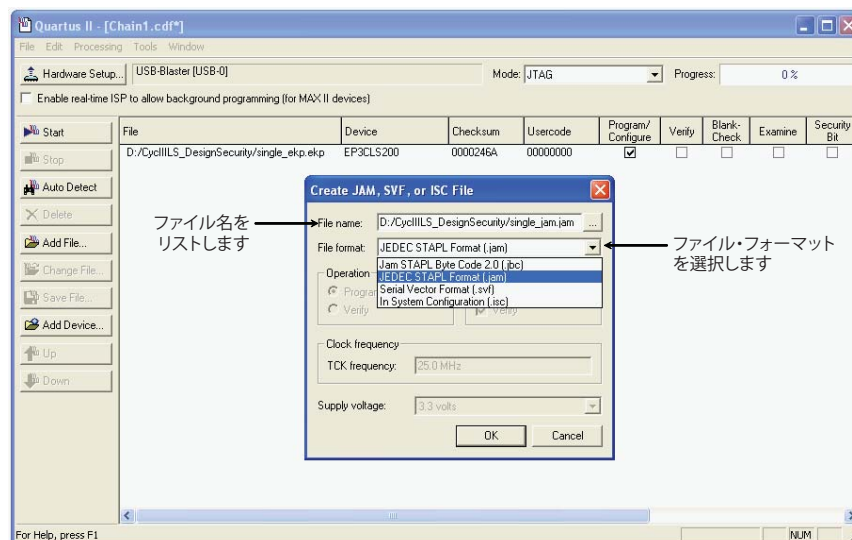
 オン・スクリーン・キーパッドが使用される際、キーボードでキーを入力すると、ポップアップが表示され、入力したキーは無視されます。

シングル・デバイスの .jam または .svf キー・ファイルの生成方法

シングル・デバイスの .jam または .svf キー・ファイルを生成するには、次の手順を実行してください。


1. Tools メニューで、**Programmer** をクリックします。**Programmer** ダイアログ・ボックスが表示されます。
2. **Mode** リストで、**JTAG** をプログラミング・モードとして選択します。
3. **Hardware Setup** をクリックします。**Hardware Setup** ダイアログ・ボックスが表示されます。
 - a. 現在選択されているハードウェア・リストから、プログラミング・ケーブルを選択します。
 - b. **Done** をクリックします。
4. **Add File** をクリックします。**Select Programmer File** ダイアログ・ボックスが表示されます。
 - a. **File name** 欄に <filename>.ekp を入力します。
 - b. **Open** をクリックします。
5. 追加した .ekp ファイルをハイライトして、そして **Program/Configure** をオンにします。
6. File メニューで、**Create/Update** をポイントして **Create JAM, SVF, or ISC File** をクリックします。**Create JAM, SVF, or ISC File** ダイアログ・ボックスが表示されます (図 6)。
7. **File format** 欄から、.ekp ファイルに必要なファイル・フォーマット (**JEDEC STAPL Format [.jam]** または **Serial Vector Format [.svf]**) を選択します。
8. **File name** 欄にファイル名を入力するか、あるいはファイルを参照して選択します。
9. **OK** をクリックして .jam または .svf ファイルを生成します。

図 6. .ekp ファイルによる .jam または .svf ファイルの作成



Quartus II ソフトウェアのコマンドライン・インタフェースによるシングル・デバイス .ekp キー・ファイルの生成およびコンフィギュレーション・ファイルの暗号化

コマンドライン・インタフェースにより、シングル・デバイスの .ekp ファイルの生成およびロウ・バイナリ・ファイル (.rbf) の暗号化が可能です。このコマンドライン・インタフェースは Quartus II ソフトウェアのコマンドライン実行コマンド `quartus_cpf` を使用します。

 `quartus_cpf` で使用可能なオプションについて詳しくは、Quartus II ソフトウェアのコマンドライン・ヘルプから `quartus_cpf -help=option` を実行してください。

シングル・デバイスの .ekp ファイルを実行するには、以下の構文またはオプションが必要です。

- `--key/-k <path to key file>:<key identity>`
 - .sof ファイル (ユーザー・デザイン)
 - .ekp ファイル (必要とされる暗号化キー・プログラミング・ファイル名)
- ```
quartus_cpf --key <keyfile>:<keyid1>:<keyid2> <input_sof_file>
<output_ekp_keyfile>
```

例 1 は、2つの異なるキー・ファイルに格納される 2 セットのキー ( `key1.key` 内の `key 1` および `key2.key` 内の `key 2` ) によって .ekp ファイルを生成するためのコマンドです。

### 例 1. 2つの異なる .key ファイル

```
quartus_cpf --key D:\CIIILS_DS\key1.key:key1 --key
D:\CIIILS_DS\key2.key:key2 D:\CIIILS_DS\test.sof
D:\CIIILS_DS\test.ekp
```

例 2 は、同じキー・ファイルに格納される 2 セットのキー ( `key12.key` 内の `key 1` および `key 2` ) によって .ekp ファイルを生成するためのコマンドです。

### 例 2. 同じ .key ファイル

```
quartus_cpf --key D:\CIIILS_DS\key12.key:key1:key2
D:\CIIILS_DS\test.sof D:\CIIILS_DS\test.ekp
```

暗号化されたシングル・デバイス .rbf ファイルを生成するには、以下の構文またはオプションが必要です。

- `-c/--convert`
  - .sof ファイル (ユーザー・デザイン)
  - .rbf ファイル (必要とされる暗号化された .rbf ファイル名)
- ```
quartus_cpf -c --key <keyfile>:<keyid1>:<keyid2>
<input_sof_file> <output_rbf_file>
```

例 3 に、.sof ファイルから暗号化された .rbf ファイルを生成するコマンドを示します。

例 3. 暗号化された .rbf ファイルの生成

```
quartus_cpf -c --key D:\CIIILS_DS\key12.key:key1:key2
D:\CIIILS_DS\test.sof D:\CIIILS_DS\test.rbf
```

Quartus II ソフトウェアによるマルチ・デバイス・キー・ファイルの生成

マルチ・デバイス **.ekp** ファイルを生成するには、次の手順を実行してください。

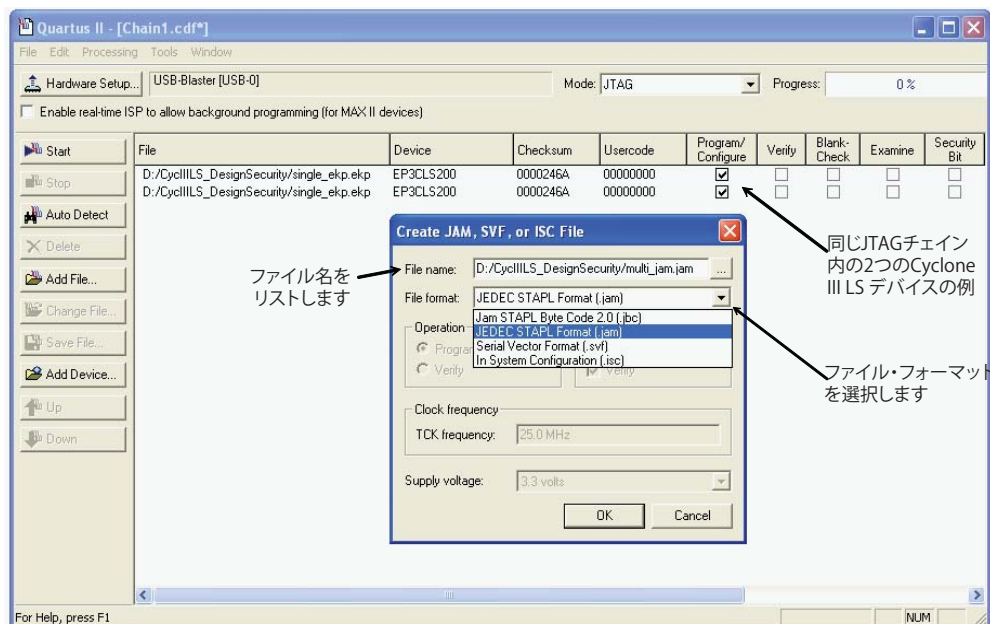
- 11 ページの「シングル・デバイスの **.jam** または **.svf** キー・ファイルの生成方法」のステップ 1～ステップ 3 を繰り返します。
2. **Add File** をクリックします。**Select Programmer File** ダイアログ・ボックスが表示されます。
 - a. シングル・デバイス **.ekp** ファイルを選択し、そして **File name** 欄に **<single_ekp>.ekp** を入力してください。
 - b. **Open** をクリックします。



同じ JTAG チェイン内におけるデバイスの正しい順序のためには、Quartus II ソフトウェア・プログラムの **Auto-Detect** オプションを使用することができます。FPGA のいずれかがキー・プログラムされる必要がない場合、Quartus II ソフトウェア・プログラマで そのデバイスを **<single_ekp>.ekp** ファイルに置き換える必要はありません。

3. 同じチェーン内の各デバイスに対してステップ 2 を繰り返します。プログラマ・ウィンドウに **.ekp** ファイルを追加する際、デバイスの順序が正しいことを確認してください。
4. 追加したすべての **.ekp** ファイルをハイライトし、**Program/Configure** をクリックします。
5. **File** メニューで、**Create/Update** をポイントして、**Create JAM, SVF, or ISC File** をクリックします。**Create JAM, SVF, or ISC File** ダイアログ・ボックスが表示されます (図 7)。
6. すべての **.ekp** ファイルに対して、**File format** 欄で必要とされるファイル・フォーマット (**.jam** または **.svf**) を選択します。
7. **File name** 欄にファイル名を入力するか、あるいはそのファイルを参照して選択します。
8. **OK** をクリックして **.jam** または **.svf** ファイルを生成します。

図 7. マルチ・デバイス・キー・ファイルの作成



ステップ 2：揮発性キーを Cyclone III LS デバイスにプログラム

揮発性キーを Cyclone III LS デバイスにプログラムする前に、FPGA が暗号化されていないコンフィギュレーション・ファイルで正しくコンフィギュレーションできることを確認してください。揮発性キーは再プログラム可能かつ消去可能なキーです。揮発性キーで Cyclone III LS デバイスをプログラムする前に、揮発性キーを保持するための外部電源が必要です。揮発性キーのプログラムが成功した Cyclone III LS デバイスは、暗号化されたコンフィギュレーション・ビットストリームおよび暗号化されていないコンフィギュレーション・ビットストリームの両方を受け入れることができます。これで、暗号化されていないコンフィギュレーション・ビットストリームをボード・レベルのテストに使用することが可能になります。

間違ったキーで暗号化されたコンフィギュレーション・ファイルで Cyclone III LS デバイスをコンフィギュレーションしようとしても、コンフィギュレーションは失敗します。これが発生した場合は、FPGA からの nSTATUS 信号は low になって、自身をリセットし続けます。

2 ページの表 2 に記載されたオン・ボード・プロトタイプ・ツールを使用して Cyclone III LS デバイスに揮発性キーをプログラムすることができます。

Quartus II ソフトウェアによる揮発性キーのプログラミング

Quartus II ソフトウェアおよびプログラミング・ケーブル (EthernetBlaster、ByteBlaster II、または USB-Blaster) を使用して揮発性キーのプログラミングを実行することができます。図 8 に示すように、プログラミング・ケーブルをプログラミング・ケーブルのヘッダに接続します。


 プログラミング・ケーブルの接続について詳しくは、[Programming Cables](#) ウェブページでの該当するプログラミング・ケーブル・ユーザーガイドを参照してください。

図 8. プログラミング・ケーブル・ヘッダ (注 1)

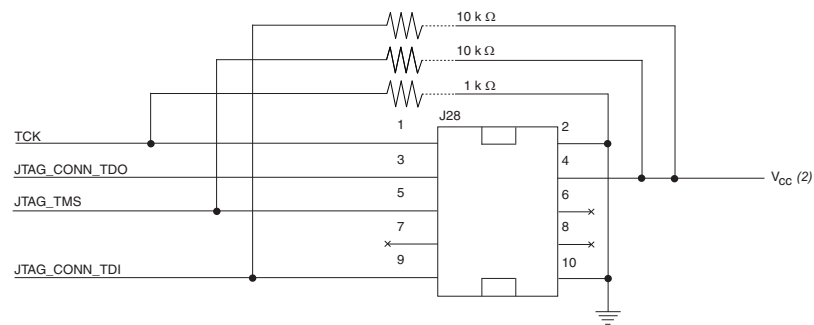


図 8 の注：

- (1) セキュリティ・キー・プログラミングにとって、EthernetBlaster、ByteBlaster II、および USB-Blaster ヘッダは同じです。
- (2) JTAG ピンが 2.5/3.0/3.3V でパワーアップされる場合、この電圧は V_{CCA} のことを指します。JTAG ピンが 1.5/1.8V の V_{CCIO} でパワーアップする場合、この電圧は V_{CCIO} のことを指します。



Quartus II ソフトウェアで EthernetBlaster を通じて揮発性キーのプログラミングを実行するには、EtherBlaster のファームウェアのバージョンをチェックする必要があります。JTAG ファームウェアのビルド番号が 101 以上であることを確認します。バージョンがビルド番号 101 の前である場合、ファームウェアの更新を実行してください (EBFW100101.tar.gz)。




ファームウェアの更新手順については、「[EthernetBlaster Communications Cable User Guide](#)」を参照してください。

Quartus II ソフトウェアによるシングル・デバイスまたはマルチ・デバイス揮発性キーのプログラミング

Quartus II ソフトウェアによってシングル・デバイス揮発性キーのプログラミングを実行するには、次の手順に従ってください。

1. Tools メニューで、**Programmer** をクリックします。プログラマ・ウィンドウが表示されます (17 ページの図 9)。
2. **Mode** リストから、**JTAG** をプログラミング・モードとして選択します (17 ページの図 9)。
3. **Hardware Setup** をクリックします。**Hardware Setup** ダイアログ・ボックスが表示されます。
 - a. 現在選択しているハードウェア・リストで、現在使用しているプログラミング・ケーブルを選択します。
 - b. **Done** をクリックします。

4. **Add File** をクリックします。**Select Programmer File** ダイアログ・ボックスが表示されます。
 - a. **.ekp** ファイルによるシングル・デバイス・キーのプログラミング
 - i. **File name** 欄に **<filename>.ekp** を入力します。
 - ii. **Open** をクリックします。
 - iii. 追加した **.ekp** ファイルをハイライトして、**Program/Configure** をクリックします (17 ページの図 9)。
 - b. **.ekp** ファイルによるマルチ・デバイス・キーのプログラミング
 - i. **File name** 欄に **<filename>.ekp** を入力します。
 - ii. **Open** をクリックします。
 - iii. 同じチェーン内のデバイスに **ステップ i ~ ステップ ii** を繰り返します。
 - iv. 追加した **.ekp** ファイルをハイライトして、**Program/Configure** をクリックします (17 ページの図 10)。

 同じ JTAG チェイン内におけるデバイスの正しい順序を保証するためには、Quartus II プログラマの **Auto-Detect** オプションを使用することができます。

 - c. **.jam** ファイルによるマルチ・デバイス・キーのプログラミング
 - i. **File name** 欄に **<filename>.jam** を入力します。
 - ii. **Open** をクリックします。
 - iii. 追加した **.jam** ファイルをハイライトして、**Program/Configure** をクリックします (18 ページの図 11)。- 5. **Start** をクリックしてキーをプログラムします。Quartus II ソフトウェアのメッセージ・ウィンドウでは、キー・プログラミング動作の成功または失敗が表示されます。

 Stratix® III、Stratix IV、および Arria® II GX デバイスと異なり、Cyclone III LS デバイスで揮発性キーのプログラミングを実行するには、**Programmer Options** ダイアログ・ボックス内の **Configure volatile design security key** をオンにする必要はありません。

図 9. .ekp ファイルによるシングル・デバイス・キーのプログラミング

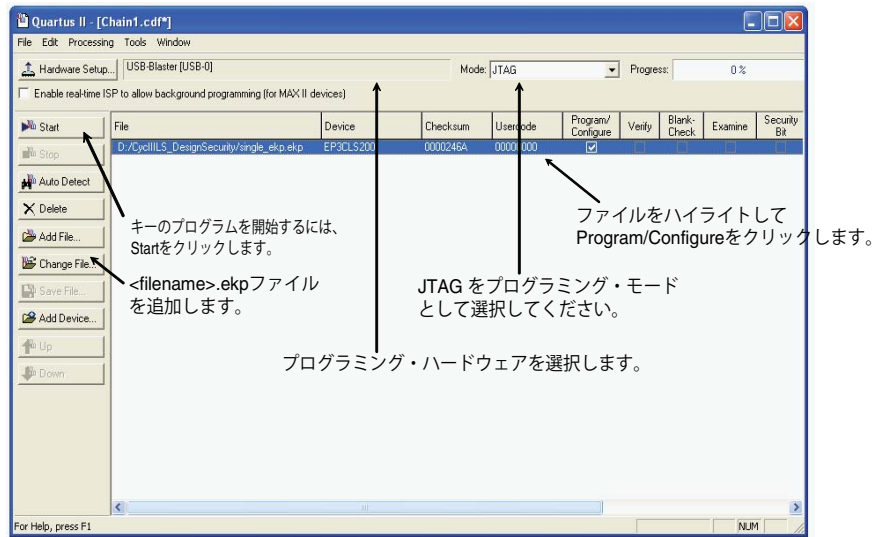


図 10. .ekp ファイルによるマルチ・デバイス・キーのプログラミング

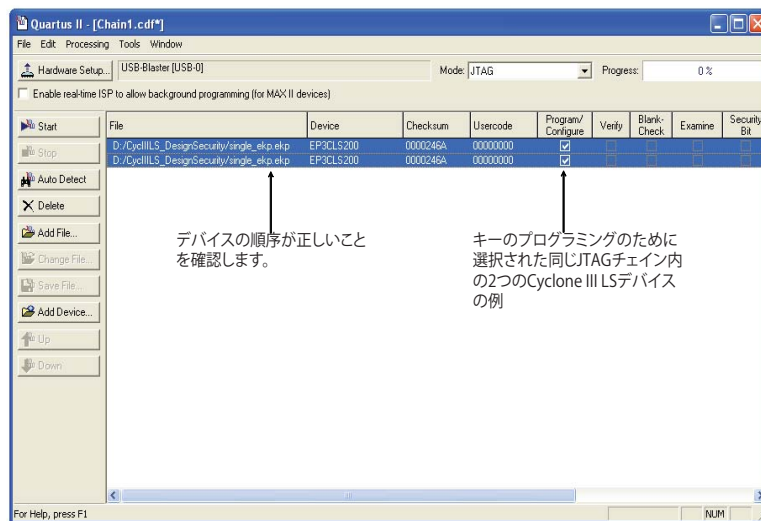
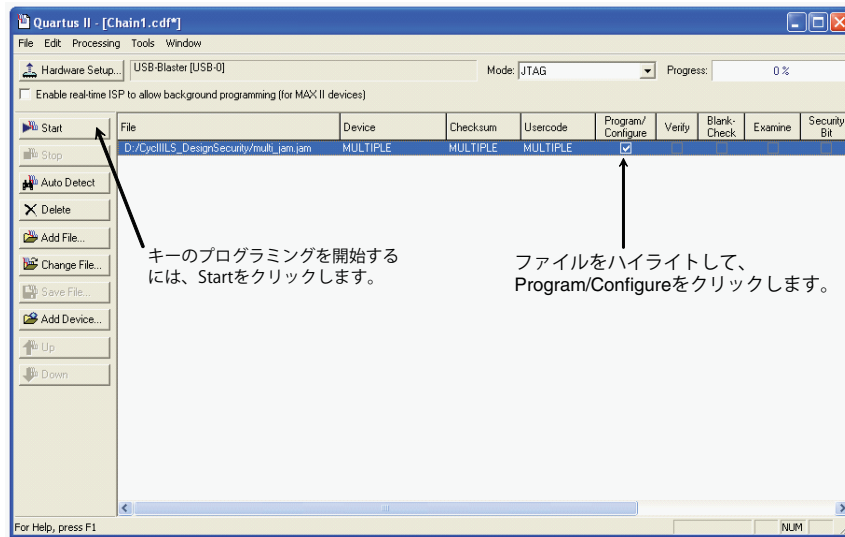


図 11. .jam ファイルによるマルチ・デバイス・キーのプログラミング




Quartus II ソフトウェアのコマンドライン・インタフェースによるシングル・デバイスまたはマルチ・デバイス揮発性キーのプログラミング


Quartus II ソフトウェアのコマンドライン・インタフェースによってシングル・デバイス揮発性キーのプログラミングを実行するには、次の手順を実行してください。

1. コマンドライン・プロンプト・ウィンドウを開きます。
2. JTAG サーバーに接続されたプログラミング・ケーブルのポート番号を確認するには、コマンドライン・プロンプトで `quartus_jli -n` を入力します。
3. 5 ページの「ステップ 1: キー・ファイルの生成およびコンフィギュレーション・ファイルの暗号化」で生成した .jam ファイルによって、下記のコマンドラインを使用して、シングル・デバイスまたはマルチ・デバイス揮発性キーのプログラミングに対して、揮発性プログラミングを実行します。

- シングル・デバイス揮発性キーのプログラミング
`quartus_jli -c<n> single_jam.jam -aKEY_CONFIGURE`
- マルチ・デバイス揮発性キーのプログラミング
`quartus_jli -c<n> multi_jam.jam -aKEY_CONFIGURE`


<n> は、-n オプションによって返されたポート番号です。

 Quartus II ソフトウェア・コマンドライン実行コマンドは、キー・プログラミング動作の成功または失敗について情報を提供します。

 `quartus_jli` コマンドラインについて詳しくは、[「AN 425: Using Command-Line Jam STAPL Solution for Device Programming」](#) の「Using the Command-Line Executable in the Quartus II Software」の項を参照してください。

JTAG Technologies による揮発性キーのプログラミング

.svf ファイルおよび JT 37xx バウンダリ・スキャン・コントローラを JT 2147 QuadPod システムと共に使用することで、デザインにセキュリティ・プログラミングを実行することができます。

 JTAG プログラミングの手順について詳しくは、JTAG Technologies のウェブサイト (www.jtag.com) を参照してください。

ステップ 3：暗号化されたコンフィギュレーション・ファイルによって Cyclone III LS デバイスをコンフィギュレーション

暗号化されたコンフィギュレーション・ファイルによって保護されている Cyclone III LS デバイスをコンフィギュレーションするには、まず暗号化されたコンフィギュレーション・データは Cyclone III LS デバイスに送信されます。そして、FPGA は以前に格納したセキュリティ・キーを用いてコンフィギュレーション・データを復号化し、暗号化されていないデータによって自身をコンフィギュレーションします。正しいセキュリティ・キーによって暗号化されたコンフィギュレーション・ファイルのみが FPGA に受け入れられ、コンフィギュレーションを成功させます。正しいセキュリティ・キーがなければ、暗号化されたファイルが盗まれても意味はありません。

セキュリティ・モードの検証

Cyclone III LS デバイスは KEY_VERIFY JTAG 命令 (表 5) をサポートし、デバイスにおける既存のセキュリティ・モードを検証することができます。**.jam** ファイルを使用してセキュリティ・モードの検証を自動化することによって揮発性キーが正しくプログラムされたかどうかをチェックすることができます。



 Cyclone III LS デバイスで利用できるセキュリティ・モードについて詳しくは、「Cyclone III デバイス・ハンドブック Volume 1」の「*Configuration, Design Security, and Remote System Upgrades in Cyclone III Devices*」の章の「Available Security Modes」の項を参照してください。

表 5. KEY_VERIFY JTAG 命令

JTAG 命令	命令コード	説明
KEY_VERIFY	00 0001 0011	TDI および TDO 間のキー検証スキャン・レジスタを接続します。

 KEY_VERIFY 命令を発行する前には、FACTORY 命令を発行して非必須な JTAG 命令のアクセスをイネーブルする必要があります。FACTORY 命令について詳しくは、「Cyclone III デバイス・ハンドブック Volume 1」の「*IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices*」の章を参照してください。

Cyclone III LS デバイス内のキー検証スキャン・レジスタは V_{CCBAT} で駆動され、256 ビット AES キーがデバイスにプログラムされたかどうかを表す 1 つの揮発性ビットのみを含みます。このビットは 256 ビット AES キーが正しくプログラムした後にセットされます。揮発性キー検証ビットの内容を読み出すことで、デバイスのセキュリティ・モードが検証できます (表 6 を参照)。

表 6. 揮発性キーの検証ビット

揮発性キーの検証ビット	セキュリティ・モード	説明
0	キーなしでの動作	この Cyclone III LS デバイスにはプログラムされたキーがないことを表します。このモードでは、暗号化されていないコンフィギュレーション・ビットストリームのみがデバイスをコンフィギュレーションできます。
1	揮発性キー	この Cyclone III LS デバイスにはキーがプログラムされてあることを表します。このモードでは、暗号化されたコンフィギュレーション・ビットストリームや暗号化されていないコンフィギュレーション・ビットストリームの両方も受け入れられます。

例 4. に、1つの Cyclone III LS デバイスのセキュリティ・モードを検証する KEY_VERIFY JTAG 命令を実行するための .jam ファイルを示します。

例 4. Cyclone III LS セキュリティ・モードを検証するための .jam ファイルの例

```
'Key Verification in JAM format
BOOLEAN verify_reg;

IRSCAN 10, $013;
WAIT 100 USEC;
DRSCAN 1, $0, CAPTURE verify_reg;

PRINT "Security Mode Verification for Single Cyclone III
LS Device ";
IF (INT(verify_reg) == 0) THEN PRINT "Security Mode: No
Key Operation";
IF (INT(verify_reg) == 1) THEN PRINT "Security Mode:
Volatile Key";
```

サポートされているコンフィギュレーション手法

デザイン・セキュリティ機能は、JTAG ベースのコンフィギュレーション方法を除く、すべてのコンフィギュレーション方法で使用できます。デザイン・セキュリティ機能は、ファースト・パッシブ・パラレル (FPP) モード (MAX[®]II デバイスまたはマイクロプロセッサおよび Flash メモリのような外部コントローラを使用する場合)、または AS および PS コンフィギュレーション手法で使用できます。

表 7 に、デザイン・セキュリティ機能をサポートする、Cyclone III LS デバイスのコンフィギュレーション手法の概要を示します。

表 7. セキュリティ・コンフィギュレーション手法

コンフィギュレーション手法	コンフィギュレーション方法	デザイン・セキュリティ
FPP	MAX II デバイス/マイクロプロセッサ、およびフラッシュ・メモリ	✓ (1)
AS	シリアル・コンフィギュレーション・デバイス	✓
PS	MAX II デバイス/マイクロプロセッサ、およびフラッシュ・メモリ	✓
	ダウンロード・ケーブル	✓ (2)
JTAG	ダウンロード・ケーブル	— (3)

表 7 の注：

- (1) ホスト・システムは 4 倍のデータ・レートの DCLK を送信する必要があります。
- (2) .sof ファイルを使用する場合、デザイン・セキュリティ機能はサポートされません。
- (3) 揮発性キーのプログラミングのみ。



デザイン・セキュリティ機能がイネーブルされる場合、復号化機能はサポートされません。ただし、デザイン・セキュリティ機能をリモート・システム・アップグレード機能と一緒に使用することはできます。



Cyclone III LS デバイス上の MSEL [3..0] ピン設定を設定して、ご使用のコンフィギュレーション手法を指定する必要があります。MSEL ピンの設定について詳しくは、「Cyclone III デバイス・ハンドブック Volume 1」の「*Configuration, Design Security, and Remote System Upgrades in Cyclone III Devices*」の「Configuration Scheme」の章を参照してください。

バウンダリ・スキャン・テスト (BST) を実行するか、または SignalTap[®] II ロジック・アナライザを用いて Cyclone III LS デバイス内の機能データを解析することができます。SignalTap II ロジック・アナライザを使用する際、最初に PS、FPP、または AS コンフィギュレーション・モードを用いて、デザインを暗号化されたコンフィギュレーション・ファイルでコンフィギュレーションする必要があります。

Quartus II ソフトウェアで SignalTap II ロジック・アナライザのウィンドウが開かれた後、デバイス・チェーンをスキャンします。これで、SignalTap II ロジック・アナライザは JTAG インタフェースを介してデータを取得する準備ができました。

- SignalTap II ロジック・アナライザで JTAG インタフェースを介してデータを取得する前には、FACTORY 命令を発行して非必須な JTAG 命令の接続をイネーブルしなければなりません。FACTORY 命令について詳しくは、「CycloneIII デバイス・ハンドブック Volume 1」の「IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices」の章を参照してください。

暗号化機能がイネーブルしたシリアル・フラッシュ・ローダのサポート

シリアル・フラッシュ・ローダ (SFL) は、シリアル・コンフィギュレーション・デバイス用のイン・システム・プログラミング・ソリューションです。AS インタフェースを介してコンフィギュレーション・デバイスを再プログラミングせずにデザインに SFL ブロックをインスタンス化することができ、シリアル・コンフィギュレーション・デバイスに格納されたデザインを更新する柔軟性が得られます。

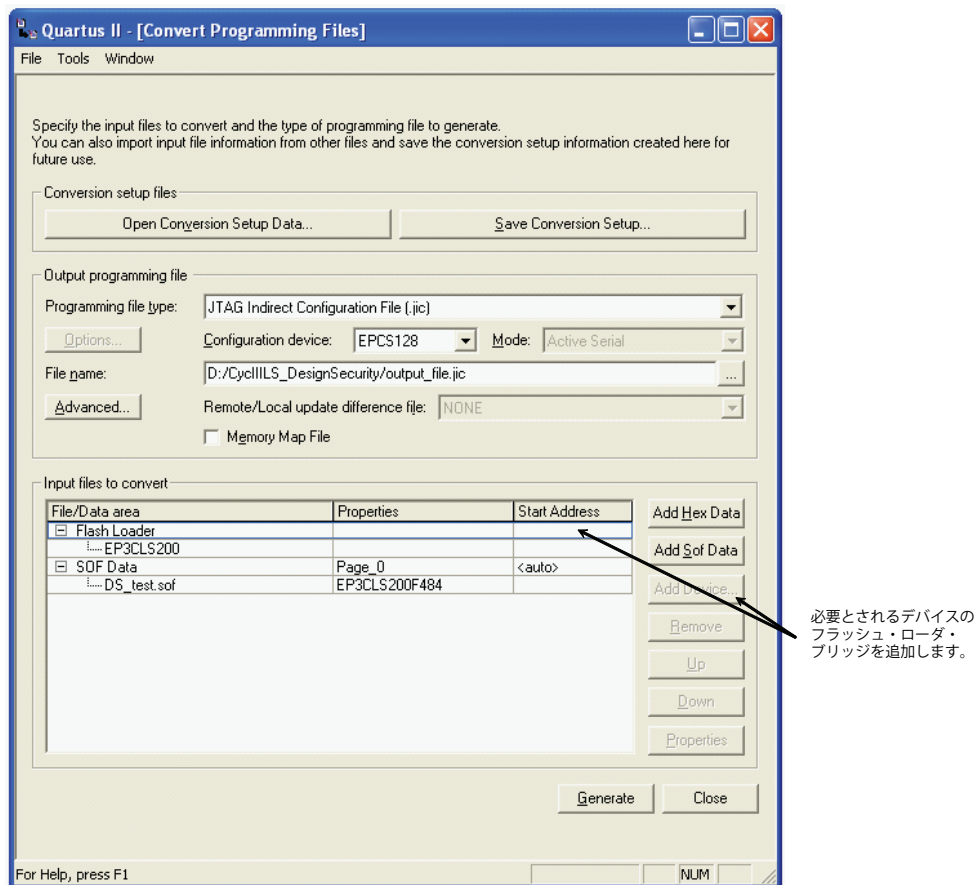
- SFL メガファンクションのインスタンス化について詳しくは、「AN 370: Using the Serial FlashLoader With the Quartus II Software」の「Instantiating SFL Megafunction in the Quartus II Software」の項を参照してください。

FPGA の JTAG インタフェースがアクセスできる限り、アプリケーションに SFL ソリューションを使用することができます。1 つの FPGA デバイス・チェーンに対して、暗号化機能がイネーブルされた SFL メガファンクションを使用するには、次の手順を実行してください。


1. Cyclone III LS デバイスのトップ・レベル・デザインに SFL メガファンクションをインスタンス化します。
2. 下記のオプションのいずれかを用いてデザインをコンパイルします。暗号化されていない .sof ファイルが生成されます。
 - Processing メニューで、**Start Compilation** をクリックします。
 - Processing メニューで、**Start** をポイントして **Start Assembler** をクリックします。

3. 次の手順に従って、**.sof** ファイルを **.jic** ファイルに変換します。
 - a. File メニューで、**Convert Programming Files** をクリックします。**Convert Programming Files** ダイアログ・ボックスが表示されます (図 12)。
 - b. **Convert Programming Files** ダイアログ・ボックスで、**the Programming file type** リストから **JTAG Indirect Configuration File (.jic)** を選択します。
 - c. **Configuration device** リストからシリアル・コンフィギュレーション・デバイスを選択します。
 - d. **File name** 欄にファイル名を入力するか、あるいはそのファイルを参照して選択します。
 - e. **Input files to convert** セクションのしたで、**SOF Data** をクリックします。
 - f. **Add File** をクリックし、**Select Input File** ダイアログ・ボックスを開きます。
 - g. 暗号化されていない **.sof** ファイルに移動し、**Open** をクリックします。
 - h. **Input files to convert** セクションのしたで、**.sof** ファイル名をクリックします。この欄がハイライトされます。6 ページの「Quartus II ソフトウェアによるシングル・デバイス .ekp キー・ファイルの生成およびコンフィギュレーション・ファイルの暗号化」でのステップ 2 に従って **.sof** ファイルを暗号化します。
 - i. **Flash Loader** をクリックして、**Add Device** をクリックします (図 12)。**Select Devices** ページが表示されます。
 - j. シリアル・コンフィギュレーション・デバイスをプログラムするのに使用されるターゲット・デバイスを選択します。**OK** をクリックします。
 - k. **Generate** をクリックして **.jic** ファイルを生成します。

図 12. .jic ファイルの作成



4. 暗号化された **.jic** ファイルによってシリアル・コンフィギュレーション・デバイスをプログラムします。

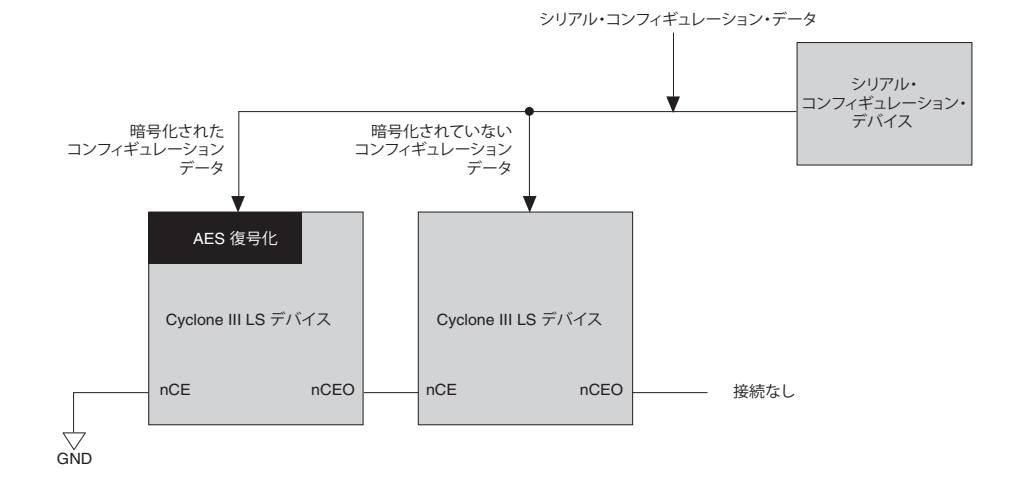
 作成された **.jic** ファイルによるシリアル・コンフィギュレーション・デバイスまたはデバイスのプログラミングについて詳しくは、「[AN 370: Using the Serial FlashLoader With the Quartus II Software](#)」の「Programming Serial Configuration Devices Using the Quartus II Programmer and .jic Files」で述べた手順を参照してください。

5. 15 ページの「[Quartus II ソフトウェアによるシングル・デバイスまたはマルチ・デバイス揮発性キーのプログラミング](#)」での手順に従って Cyclone III LS デバイスにキーをプログラムします。
6. 暗号化された Cyclone III LS デバイスは、プログラムされたシリアル・コンフィギュレーション・デバイスによってコンフィギュレーションされます。

コンフィギュレーション手法の選択時の考慮事項

図 13 に示すように、シリアル・コンフィギュレーション手法（AS または PS）では、同じコンフィギュレーション・チェーン内の暗号化されていないデータを受け入れるほかのアルテラ・デバイスなど、デザイン・セキュリティ機能を使用していないデバイスと一緒に、一連の Cyclone III LS デバイスをカスケード接続することができます。

図 13. マルチ・デバイス Cyclone III LS アクティブ・シリアル・コンフィギュレーション



FPP を使用する場合、チェーン内のすべての Cyclone III LS デバイスでは、デザイン・セキュリティ機能をイネーブルまたはディセーブルのいずれかにしておく必要があります。26 ページの「FPP コンフィギュレーション使用時の DCLK 考慮事項」で説明したと DATA と DCLK の関係のため、チェーン内の各デバイスに対してデザイン・セキュリティ機能を選択的にイネーブルすることはできません。チェーンにデザイン・セキュリティをサポートしないデバイスが含まれる場合は、シリアル・コンフィギュレーション手法を使用することが推奨されています。

ただし、FPP モードを使用する場合、DATA と DCLK の関係は、圧縮復元機能の使用時およびデザイン・セキュリティの使用時は同じです。チェーン内のすべてのデバイスが圧縮復元機能を使用する場合、各デバイスに対してデザイン・セキュリティ機能を選択的にイネーブルすることができます。

FPP コンフィギュレーション使用時の DCLK 考慮事項

Cyclone III LS デバイスへの DATA および DCLK 信号のフローを制御するために、デザイン・セキュリティ機能がイネーブルした FPP コンフィギュレーション手法には、MAX II デバイスまたはマイクロプロセッサなどの外部ホストの使用が必要です。

DCLK はコンフィギュレーション・プロセスにクロックを供給するためのクロック・ソースです。コンフィギュレーション・データは DATA [7..0] ピンで受信されます。FPP コンフィギュレーション手法で Cyclone III LS デバイスのデザイン・セキュリティ機能を使用する場合、外部ホストはデータ・レートの 4 倍の DCLK 周波数を送信できなければなりません。

4x DCLK 信号は、追加ピンの必要がなく、DCLK ピン上で送信されます。

Cyclone III LS デバイスでサポートされている最大 DCLK 周波数は 100 MHz であり、式 1 に示すような最大データ・レートを生成します。

式 1. 最大 DCLK 周波数

$$\frac{100 \text{ MHz}}{4} \times 8 \text{ bits} = 200 \text{ Mbps}$$

デザイン・セキュリティ機能を使用している場合、Cyclone III LS デバイスでは最初のコンフィギュレーション・データが DCLK の最初の立ち上がりエッジでラッチされます。後続のデータ・バイトはその前の各データ・バイトの 4 クロック・サイクル後にラッチされます。コンフィギュレーションを正しく行うには、DCLK 速度が式 1 に指定された規定周波数以下でなければなりません。DCLK には最大周期がありません。これは DCLK を無制限に停止することによってコンフィギュレーションを休止できることを意味します。

FPP コンフィギュレーション手法でデザイン・セキュリティ機能を使用していて、DCLK を停止する必要がある場合は、最終データ・バイトが Cyclone III LS デバイスにラッチされた 3 クロック・サイクル後でのみ停止できます。これで、コンフィギュレーション回路はラッチされたコンフィギュレーション・データの最終バイトを処理するのに十分なクロック・サイクルを与えます。クロックがリスタートしたら、最初の DCLK の立ち上がりエッジを送信する前に、データは DATA [7..0] ピンに存在しなければなりません。

デザイン・セキュリティ機能がイネーブルしたタイミング波形

図 14 に、MAX II デバイスまたはマイクロプロセッサを外部ホストとして使用するときの、FPP コンフィギュレーションのタイミング波形を示します。この波形は、デザイン・セキュリティ機能がイネーブルされているときのタイミングを示します。

図 14. デザイン・セキュリティ機能がイネーブルした FPP コンフィギュレーション・タイミング波形
(注 1)、(2)

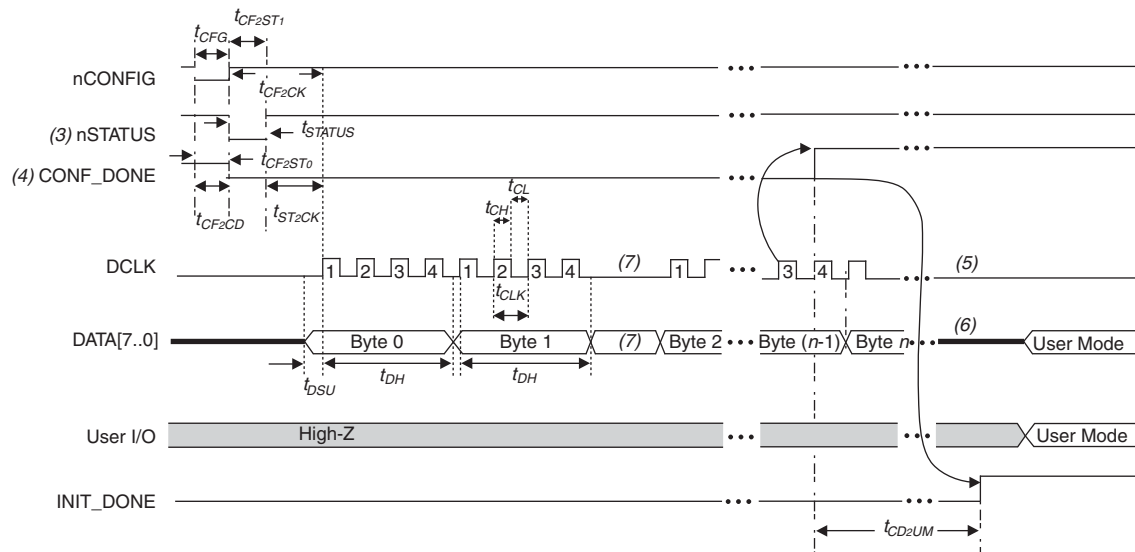


図 14 の注：

- (1) このタイミング波形は、圧縮復元機能やデザイン・セキュリティ機能が使用されているときに利用してください。
- (2) この波形の開始はデバイスがユーザー・モードにあることを示します。ユーザー・モードでは、nCONFIG、nSTATUS、および CONF_DONE はロジック High レベルにあります。nCONFIG が Low にプルダウンされると、リコンフィギュレーション・サイクルが開始します。
- (3) 電源投入後、Cyclone III LS デバイスは POR 遅延の間 nSTATUS を Low に保持します。
- (4) 電源投入後、コンフィギュレーションの実行前と実行中、CONF_DONE は Low になります。
- (5) DCLK ピンは、コンフィギュレーション後にフロートの状態のままにはなりません。DCLK は High または Low のうち都合の良いレベルにドライブしなければなりません。
- (6) DATA [7..0] はユーザ I/O ピンとして使用可能であり、このピンの状態は兼用ピンの設定によって決まります。
- (7) DCLK ピンは必要に応じて一時停止することができます。DCLK がリスタートしたら、外部ホストは最初の DCLK の立ち上がりエッジを送信する前に、DATA [7..0] ピンにデータを供給しなければなりません。

表 8 に、デザイン・セキュリティ機能がイネーブルした Cyclone III LS デバイスの FPP タイミング・パラメータを示します。

表 8. デザイン・セキュリティ機能がイネーブルした Cyclone III LS デバイスの FPP タイミング・パラメータ (注 1), (3)

シンボル	パラメータ	最小	最大	ユニット
t_{CF2CD}	nCONFIG low から CONF_DONE Low まで	—	500	ns
t_{CF2ST0}	nCONFIG low から nSTATUS low まで	—	500	ns
t_{CFG}	nCONFIG low のパルス幅	500	—	ns
t_{STATUS}	nSTATUS low のパルス幅	45	685 (2)	μ s
t_{CF2ST1}	nCONFIG high から nSTATUS high まで	—	685 (2)	μ s
t_{CF2CK}	nCONFIG high から DCLK の最初の立ち上がりエッジまで	685 (2)	—	μ s
t_{ST2CK}	nSTATUS high から DCLK の最初の立ち上がりエッジ	2	—	μ s
t_{DSU}	DCLK の立ち上がりエッジの前のデータ・セットアップ時間	5	—	ns
t_{DH}	Data hold time after rising edge on DCLK の立ち上がりエッジの後のデータ・ホールド時間	22.5	—	ns
t_{CH}	DCLK high 時間	3.2	—	ns
t_{CL}	DCLK low 時間	3.2	—	ns
t_{CLK}	DCLK 周期	7.5	—	ns
f_{MAX}	DCLK 周波数	—	100	MHz
t_{CD2UM}	CONF_DONE high からユーザー・モードまで (3)	300	650	μ s
t_{CD2CU}	CONF_DONE high から CLKUSR がイネーブルされるまで	DCLK の最大周期の 4 倍	—	—
t_{CD2UMC}	CONF_DONE high から CLKUSR オプションがオンしたユーザー・モードまで	$t_{CD2CU} + (3,192 \times \text{CLKUSR 周期})$	—	—

表 8 の注：

- (1) この情報は暫定仕様です。
- (2) この値は、ユーザーが nCONFIG または nSTATUS の Low パルス幅を拡張して、コンフィギュレーションを遅延しない場合に適用されます。
- (3) 最小値および最大値は、デバイスを起動させるためのクロック・リソースとして内部オシレータが選択された場合にのみ適用されます。

米国の輸出コントロール

一般に、Cyclone III LS デバイスの米国における輸出コントロールは、米国の ECCN3A001.a.7 または 3A991.d に分類されています。Cyclone III LS デバイスが復号化を実行しますが、その復号化機能はコンフィギュレーション・ビットストリームの保護にのみ使用されるため、デバイス輸出コントロールの分類は変更しません。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アルテラの Quartus II ソフトウェア開発ツール (V 9.0 以降) は、正式に US ECCN 5D002 c.1 に分類され、許可例外 ENC により「製品版」商品として大部分の国にエクスポートされます。エクスポートに関する質問については、opexp_imp@altera.com にお問い合わせください。

改訂履歴

表 9 に、本資料の改訂履歴を示します。

表 9. 改訂履歴

日付およびリビジョン	変更内容	概要
2009 年 9 月 v1.0	初版	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before

