

## Introduction

In the highly competitive commercial and military environments, design security is an important consideration for digital designers. As FPGAs start to play a role in larger and more critical system components, it is imperative to protect designs from copying, reverse engineering, and tampering. Stratix® II devices are the first FPGAs in the industry to address these concerns with the ability to decrypt a configuration bitstream using the advanced encryption standard (AES) algorithm, an industry standard encryption algorithm.

During device operation, Altera® FPGAs store configuration data in SRAM configuration cells. Because SRAM memory is volatile, SRAM cells must be loaded with configuration data each time the device powers up. Configuration data is sent from an external memory source, such as a flash memory or a configuration device, to the FPGA. It is possible to intercept configuration data when it is being sent from the memory source to the FPGA. The intercepted configuration data can be used to configure another FPGA.

When using the Stratix II or Stratix II GX design security feature, the security key is stored in a non-volatile location inside the Stratix II or Stratix II GX device. To successfully configure a Stratix II device with the programmed security key, you must use a configuration file that is encrypted with the same key. While the key storage is non-volatile, Stratix II and Stratix II GX devices do not require any external device, such as a back-up battery.

The design security feature is available when configuring Stratix II and Stratix II GX devices using the fast passive parallel (FPP) configuration mode with an external host (such as a MAX® II device or microprocessor), or when using active serial (AS) or passive serial (PS) configuration schemes. The design security feature is not available when you configure your Stratix II or Stratix II GX device using FPP with an enhanced configuration device, passive parallel asynchronous (PPA), or Joint Test Action Group (JTAG)-based configuration. For more details, refer to [“Supported Configuration Schemes” on page 28](#).

## Overview of the Design Security Feature

The Stratix II and Stratix II GX design security feature protects against copying, reverse engineering, and tampering. The design approach to make the solution secure includes the following:

- The security key is stored in polyfuses under layers of metals among other polyfuses. It is very difficult to determine the functionality of a particular fuse by simple visual inspection. Moreover, additional physical security is designed around the polyfuses to provide further security.
- Stratix II and Stratix II GX devices do not support configuration file readback. This prevents attempts to read back the configuration file after it is decrypted.


- Two 128-bit sequences are required to generate the 128-bit security key and are required to program the key into the Stratix II device. Your design cannot be copied by programming a 128-bit security key into another device and configuring it with an encrypted configuration file. It is virtually impossible to generate the two 128-bit sequences from the security key.
- The polyfuses used to store the security key are non-volatile and one-time programmable. No battery is required. After the Stratix II or Stratix II GX device is programmed with the key, it can only be configured with configuration files encrypted with the same key. Attempts to configure the Stratix II or Stratix II GX device with an unencrypted configuration file or a configuration file encrypted with the wrong key result in configuration failure. Therefore, tampering of your design file can be detected.


## Security Encryption Algorithm

Stratix II and Stratix II GX devices have a dedicated decryption block that uses the AES algorithm to decrypt configuration data with a 128-bit user-defined security key. Prior to receiving the encrypted data, the user-defined 128-bit security key must be written into the device.

The AES algorithm is a symmetric block cipher that encrypts and decrypts data in blocks of 128 bits. The encrypted data is subject to a series of transformations that includes byte substitutions, data mixing, data shifting, and key additions.

Stratix II and Stratix II GX devices contain an AES decryptor block that uses the AES algorithm to decrypt the configuration data prior to configuring the device. If the security feature is not used, the AES decryptor is bypassed. The Stratix II and Stratix II GX AES implementation is validated as conforming to the Federal Information Processing Standards FIPS-197.

 For more information about the AES algorithm, refer to the *Federal Information Processing Standards Publication FIPS-197* or the *AES Algorithm (Rijndael) Information* at [www.csrc.nist.gov](http://www.csrc.nist.gov).

 For more information about the Stratix II AES validation, refer to the *Advanced Encryption Standard Algorithm Validation List* published by the National Institute of Standards and Technology (NIST) at [www.csrc.nist.gov](http://www.csrc.nist.gov).

## Key Programming

Table 1 describes the four different methods for key programming.

**Table 1.** Key Programming Methods (Note 1)

Programming Procedure	Method	Programming
On-Board Programming	Prototyping	EthernetBlaster, JTAG Technologies
	Production	In-Circuit Tester, JTAG Technologies
Off-Board Programming	Prototyping	System General
	Production	System General

**Note to Table 1:**

(1) Contact [Altera Technical Support](#) for information about programming support.

Key programming uses the following definitions:

- **On-board**— procedure where the device is programmed on your board.
- **Off-board**— procedure where the device is programmed on a separate programming system.
- **Prototyping**— the method initially used to verify proper operation of a particular method.
- **Production**— method used for large volume production.

## Voltage Requirements

This section provides the voltage requirements for the Stratix II and Stratix II GX design security feature. When using this feature, a security key is stored in non-volatile memory in Stratix II and Stratix II GX devices. During user mode and configuration mode,  $V_{CCPD8}$  (3.3 V nominal) powers I/O bank 8. When programming the design security key,  $V_{CCPD8}$  (3.7 V nominal) powers the circuitry enabling the key to be programmed in non-volatile memory. The key is programmed before the device is configured and enters user mode.

During key programming, you must apply 3.7 V to  $V_{CCPD8}$ . To ensure all voltage transients are settled and all power supplies are stable, begin programming the security key at 100 ms after applying 3.7 V to  $V_{CCPD8}$ . To ensure reliable key programming, a voltage monitor on 3.7 V  $V_{CCPD8}$  can help coordinate the start of key programming. Table 2 shows the voltage regulation specifications.

**Table 2.** Voltage Regulation Specifications

Parameter	Key Programming Mode
Voltage ( $V_{CCPD8}$ )	3.7 V $\pm$ 0.1 V (1)
DC Current ( $I_{CCPD8}$ )	250 mA (maximum) (2)
Instantaneous Current ( $I_{CCPD8}$ )	500 mA (maximum) (3)
Key Programming Mode Duration	500 ms (typical)
Duration for $V_{CCPD8}$ set at 3.7 V	60 s (maximum)
$V_{CCPD8}$ Rise Time	100 $\mu$ s $\leq$ rise time $\leq$ 20 ms
TCK Period	100 $\mu$ s $\pm$ 1 $\mu$ s

**Table 2.** Voltage Regulation Specifications


Parameter	Key Programming Mode
Ambient Temperature	25°C ± 5°C


**Notes to Table 2:**


- (1) When the key is not being programmed into the Stratix II device (for example, after power-up and before configuration, during user-mode) V<sub>CCPD8</sub> must be set to 3.3 V ± 5% when Stratix II is powered up.
- (2) This is the maximum I<sub>CC</sub> from V<sub>CCPD8</sub> during the security key programming sequence. For user-mode operation after configuration, this can go as high as 1 A maximum when user I/Os toggle. This 1A is not applicable to out of system (off-board) programming. Off-board programmers must only meet the 500 mA number.
- (3) For 100 μs during fuse programming only.

Prior to key programming, V<sub>CCPD8</sub> must ramp up from 3.3 V to 3.7 V in 100 μs to 20 ms. This period of time is from when 3.7 V is applied to V<sub>CCPD8</sub> to when the voltage level has stabilized and transients have settled. Key programming must not commence prior to 3.7 V being stable on V<sub>CCPD8</sub>. For more information, refer to “[Step 2: Program the Security Key into a Stratix II or Stratix II GX Device](#)” on page 20.

For normal configuration, V<sub>CCPD</sub> voltages must ramp up from 0 V to 3.3 V in 100 μs to 100 ms. If V<sub>CCPD</sub> does not reach 3.3 V in this time period, the device will not configure successfully. If your system does not allow for a V<sub>CCPD</sub> ramp-up time of 100 ms or less, you must hold nCONFIG low until all power supplies stabilize.

 Additional operating conditions must be met to ensure proper operation. For more information about the V<sub>CCINT</sub>, V<sub>CCIO</sub>, GND, and other V<sub>CCPD</sub> specifications for Stratix II devices, refer to the *DC and Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook*. For more information about the V<sub>CCINT</sub>, V<sub>CCIO</sub>, GND, and other V<sub>CCPD</sub> specifications for Stratix II GX devices, refer to the *DC and Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

 Key programming must be performed during the typical 500 ms time period. The V<sub>CCPD8</sub> voltage remains at 3.7 V for longer than 60 seconds, Stratix II device reliability can be adversely affected.

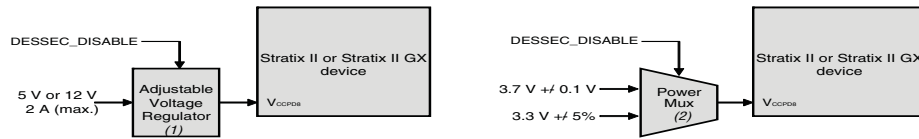
 Key programming must be performed in the allowable TCK frequency, which is up to 10 kHz.

## Providing 3.7 V to V<sub>CCPD8</sub>

Programming the security key into the device requires 3.7 V on V<sub>CCPD8</sub>, as specified in [Table 2](#). This requirement is accomplished in one of several ways, depending on the programming procedure used (for example, off-board or on-board). In situations where 3.3 V is nominally supplied to V<sub>CCPD8</sub> during user-mode operation, the key programming procedure requires a temporary disconnection of 3.3 V to temporarily connect 3.7 V to V<sub>CCPD8</sub> for the specified duration.

[Figure 1](#) shows options for power supply connection to V<sub>CCPD8</sub>, generally applicable to on-board programming environments where the device is already mounted on a board. The actual environment used determines how 3.7 V is provided. A power supply can either directly drive V<sub>CCPD8</sub> and switch between 3.7 V and 3.3 V, or a high-amperage multiplexer can directly drive V<sub>CCPD8</sub> with 3.7 V and 3.3 V as the multiplexer inputs ([Figure 2](#)).

**Figure 1.**  $V_{CCPD8}$  Power Supply Options



**Notes to Figure 1:**

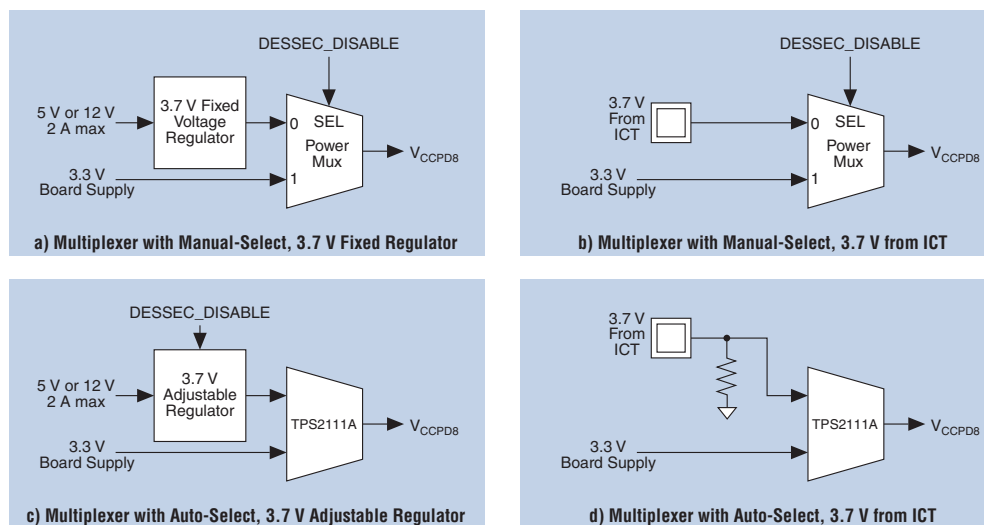
- (1) The voltage regulator is driving  $V_{CCPD8}$ .
- (2) The power multiplexer is driving  $V_{CCPD8}$ .

Altera recommends a high-amperage power multiplexer for most on-board programming cases (Figure 2). This power multiplexer is used to select whether 3.7 V or 3.3 V drives  $V_{CCPD8}$  directly. The power multiplexer can use the `DESSEC_DISABLE` control signal to determine which voltage to pass through. In this case, a dedicated voltage regulator can provide 3.7 V to one input, while the main 3.3 V rail provides the other input (A in Figure 2). In an ICT environment, 3.7 V is provided via a contact point on the board, active during key programming only (B in Figure 2). Ideally, the power multiplexer is self-selecting, passing through the higher voltage of the two inputs. In C in Figure 2, `DESSEC_DISABLE` is used to disable the 3.7 V supply if generated on-board. In D in Figure 2, the ICT disables the 3.7 V, requiring a pull-down resistor to ground the input. `DESSEC_DISABLE` is not required in this option. For more information about the solution that supports key programming via the EthernetBlaster communications cable, ICT, or an other third-party JTAG programmer, refer to “Power Multiplexer Reference Circuit” on page 6.



`DESSEC_DISABLE` is a standard, active-low, 3.3-V LVTTTL control signal. When `DESSEC_DISABLE` is low, it sets the output of the voltage regulator or power multiplexer to 3.7 V.

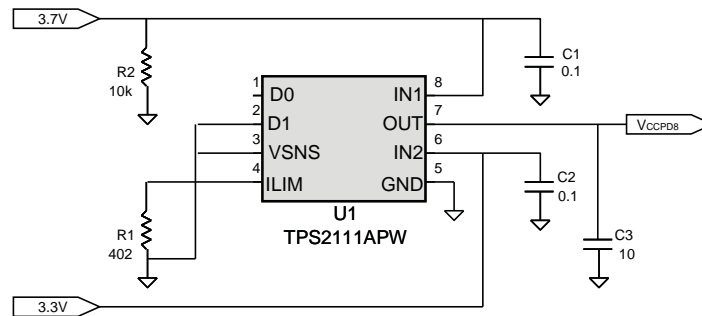
**Figure 2.** High-Amperage Power Multiplexer Options



## Power Multiplexer Reference Circuit

A reference design is developed to achieve the power requirements of key programming. It features an auto-switching power multiplexer, which is a circuit that inputs two voltages and automatically switches its output to the higher voltage supply. This meets board requirements of small size, low component count, and a low-cost solution.

**Figure 3.** Power Multiplexer Reference Circuit- Schematic *Notes (1),(2), (3), (4), (5), (6)*



**Notes to Figure 3:**

- (1) R1 = 402 W sets current limit from 0.95 A to 1.56 A.
- (2) 3.7 V and 3.3 V may require additional caps if rails are distant.
- (3) TPS2111APW is an autoswitching power multiplexer from Texas Instruments.
- (4) C3 output capacitor size controls droop at switchover. For assistance sizing C3, go to the Texas Instruments website to view the TPS2111A data sheet.
- (5) A 0.1  $\mu$ F capacitor must be near  $V_{CCPD8}$  for high frequency noise.
- (6) D0 has an internal pull-up to the higher of IN1 or IN2.

**Table 3.** Power Multiplexer Reference Circuit - Bill of Material (BOM) *Notes (1), (2), (3)*

Ref Des	Number	Description	MFR	Part Number	Size
U1	1	IC, Power Multiplexer	Texas Instruments	TPS2111APW	PW-8
R1	1	Res, 402, 1%, 1/10W	Standard (Std)	Std	603
R2	1	Res, 10k, 1%, 1/10W	Std	Std	603
C1, C2	2	Cap, 0.1 $\mu$ F, 10%, 10V, X74	Std	Std	603
C3	1	Cap, Ceramic, 10 $\mu$ F, 10%, 6.3V, X7R	Std	Std	1206

**Notes to Table 3:**

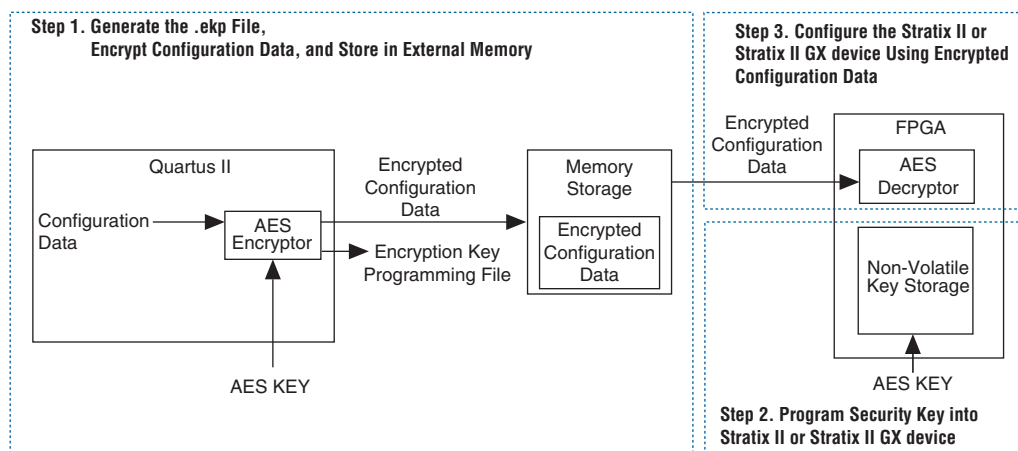
- (1) Size refers to surface mount component specifications.
- (2) Unless otherwise specified R's tolerance  $\pm 1\%$  and capacitor's tolerance  $\pm 10\%$ .
- (3) C3 output capacitor size controls droop at switchover. For assistance in sizing C3, go to the Texas Instruments website at [www.ti.com](http://www.ti.com) and view the TPS2111A data sheet.

## Providing a Secure Configuration Flow

To provide a secure configuration flow, perform the steps shown in [Figure 4](#):

1. Generate the Encryption Key Programming (.ekp) file and encrypt the configuration data. The Quartus II software uses the same security key to generate an encrypted configuration file. The encrypted configuration file is stored in an external memory, such as a flash memory or a configuration device. For more information, refer to [“Step 1: Generate the .ekp File and Encrypt Configuration Files”](#).
2. Program the non-volatile 128-bit security key into the Stratix II or Stratix II GX device. For more information, refer to [“Step 2: Program the Security Key into a Stratix II or Stratix II GX Device”](#) on page 20.
3. Configure the Stratix II or Stratix II GX device. At power-up, the external memory source sends the encrypted configuration file to the Stratix II or Stratix II GX device. The Stratix II or Stratix II GX device uses the stored security key to decrypt the file and to configure itself. For more information, refer to [“Step 3: Configure the Stratix II and Stratix II GX Devices”](#) on page 26.

**Figure 4.** Stratix II and Stratix II GX Secure Configuration Flow



### Step 1: Generate the .ekp File and Encrypt Configuration Files

To use the design security feature in Stratix II and Stratix II GX devices, you must generate an **.ekp** and encrypt your configuration files with the Quartus II software version 6.0 SP1 or later (make sure you use the same two 128-bit sequences for both). The security key is not saved into Quartus II-generated files and the actual 128-bit security key is generated from the two 128-bit sequences. This makes it impossible to copy the security key to other Stratix II or Stratix II GX devices.



Due to AES export regulations, a license file and Quartus II software patch must be obtained for Quartus II software versions 5.0 through 7.0. For the Quartus II software version 7.2 and later, a license file must be obtained. Contact [Altera Technical Support](#) for assistance.

The **.ekp** has different formats, depending on the hardware or system used for programming. There are three file formats supported in the Quartus II software version 6.0 SP1 and later:

- JBC (**.ekp**)
- JAM™ STAPL (**.jam**)
- SVF (**.svf**)



Only the **.ekp** type is generated automatically by the Quartus II software. You must create the **.jam** and **.svf** using the Quartus II software if these files are required in the key programming. The Quartus II software version 6.0 SP1 or later generates the JBC format of the **.ekp** in the same project directory.

The **.ekp** is used with the EthernetBlaster communications cable. The **.jam** is generally used with third party programming vendors and JTAG programmer vendors. The **.svf** is used with JTAG programmer vendors and in-circuit test vendors.

### How to Generate the Single-Device **.ekp** and Encrypt the Configuration File with the Quartus II Software Version 6.0 SP1

Perform the following steps to generate a single-device **.ekp** and encrypt your configuration file:

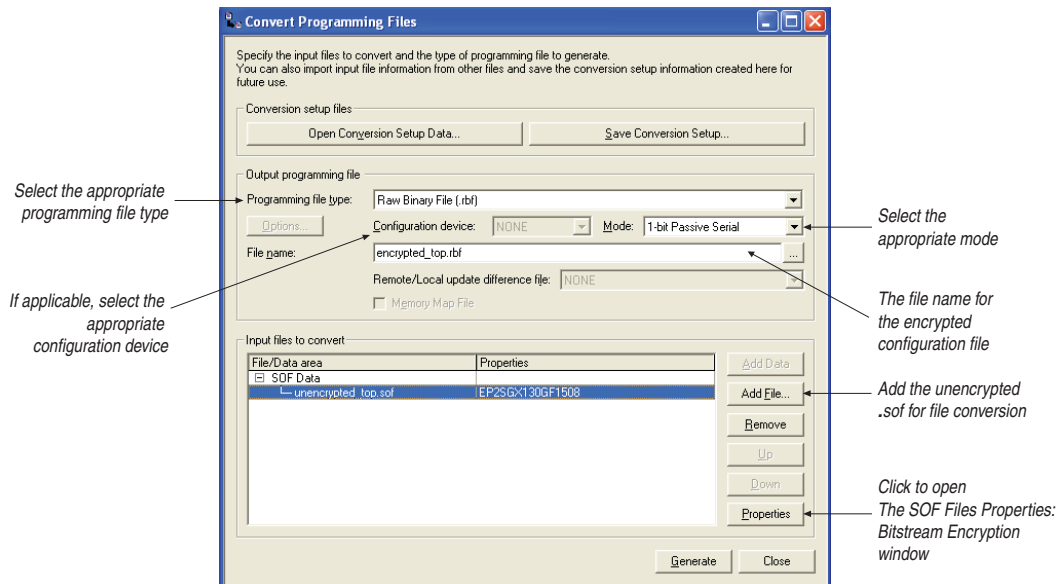
1. Obtain a license file for all versions of the Quartus II software and a Quartus II software patch for Quartus II software versions 5.0 through 7.0 to enable the Stratix II design security feature support from [Altera Technical Support](#).
2. Install the Quartus II software patch over the Quartus II software version 6.0 SP1.



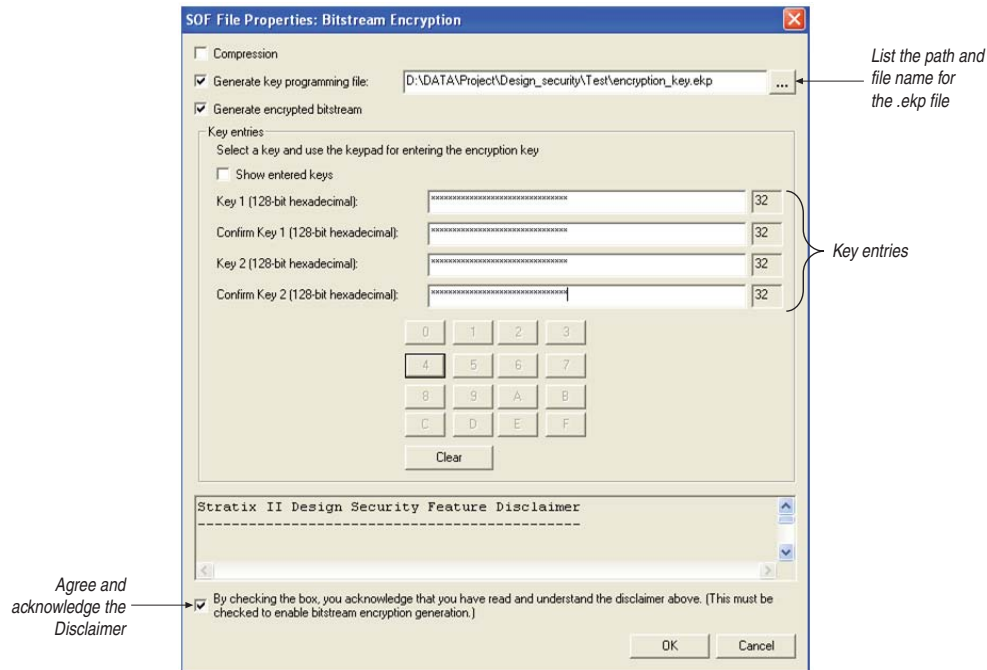
The Quartus II software version 6.1 requires a different software patch from the Quartus II software version 6.0 SP1. An updated patch can be obtained through [Altera Technical Support](#).

- a. On the Tools menu, click **License Setup**. The **Options** dialog box displays the **License Setup** options.
  - In the **License file** field, enter the location and name of the license file or browse to and select the license file.
  - Click **OK**.
3. Compile your design with one of the following options:
  - On the Processing menu, click **Start Compilation**. (period)
  - On the Processing menu, point to **Start** and click **Start Assembler**. An unencrypted SRAM Object File (**.sof**) is generated.
4. On the File menu, click **Convert Programming Files**. The **Convert Programming Files** dialog box displays ([Figure 5](#)).



**Figure 5.** Convert Programming Files Dialog Box

- a. In the **Convert Programming Files** dialog box, select the programming file type from the **Programming file type** list.
- b. If applicable, select the appropriate configuration device from the **Configuration device** list.
- c. Select the mode from the **Mode** list.
- d. Type the file name in the **File name** field or browse to and select the file.
- e. Under **Input files to convert** section, click **SOF Data**.
- f. Click **Add File**.
- g. Browse to the unencrypted **.sof** and click **Open**.
- h. In the **Input files to convert** dialog box, click on the **.sof** name. The field is highlighted.
- i. Click **Properties**. The **SOF Files Properties: Bitstream Encryption** dialog box displays (Figure 6).
- j. Turn on **Generate key programming file** and type the **.ekp** path and file name in the text area or browse to and select **<filename>.ekp**.
- k. Turn on **Generate encrypted bitstream**.

**Figure 6.** SOF File Properties: Bitstream Encryption Dialog Box

5. Under **Key entries**, turn on **Show entered keys**. This selection disables the **Confirm Key 1 (128-bit hexadecimal)** and **Confirm Key 2 (128-bit hexadecimal)** fields.
6. Click in the **Key 1 (128-bit hexadecimal)** field.
  - a. Using the hexadecimal character keypad located in the center of the dialog box, click on the alphanumeric characters that represent the first 128-bit sequence in hexadecimal format.
  - b. Because the key is entered in hexadecimal format, you must provide 32 characters. The number in the far right box displays the number of characters that have been entered. When all 32 characters have been entered, the keypad is disabled.
7. Under **Key entries**, click in the **Key 2 (128-bit hexadecimal)** field.
  - a. Using the hexadecimal character keypad located in the center of the dialog box, click on the alphanumeric characters that represent the second 128-bit sequence in hexadecimal format.



If you did not turn on **Show entered keys** in Step 5, the characters in the **Key 1** and **Key 2** fields are represented by the '\*' character. In addition, you must enter the exact set of hexadecimal characters from the **Key 1** field into the **Key Confirm 1 (128-bit hexadecimal)** field, and the exact set of hexadecimal characters from the **Key 2** field into the **Key Confirm 2 (128-bit hexadecimal)** field.

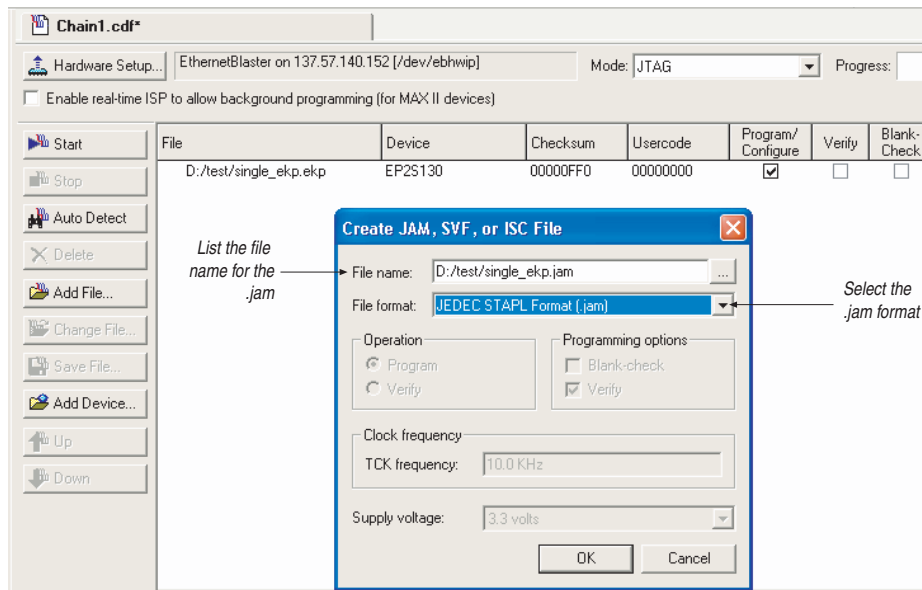
8. Read the **Stratix II Design Security Feature Disclaimer**. If you agree to and acknowledge the **Stratix II Design Security Feature Disclaimer**, turn on the check box acknowledging you have read and accept the disclaimer.
9. Click **OK**.



The **OK** button is disabled if you have not completed Steps 1–8.

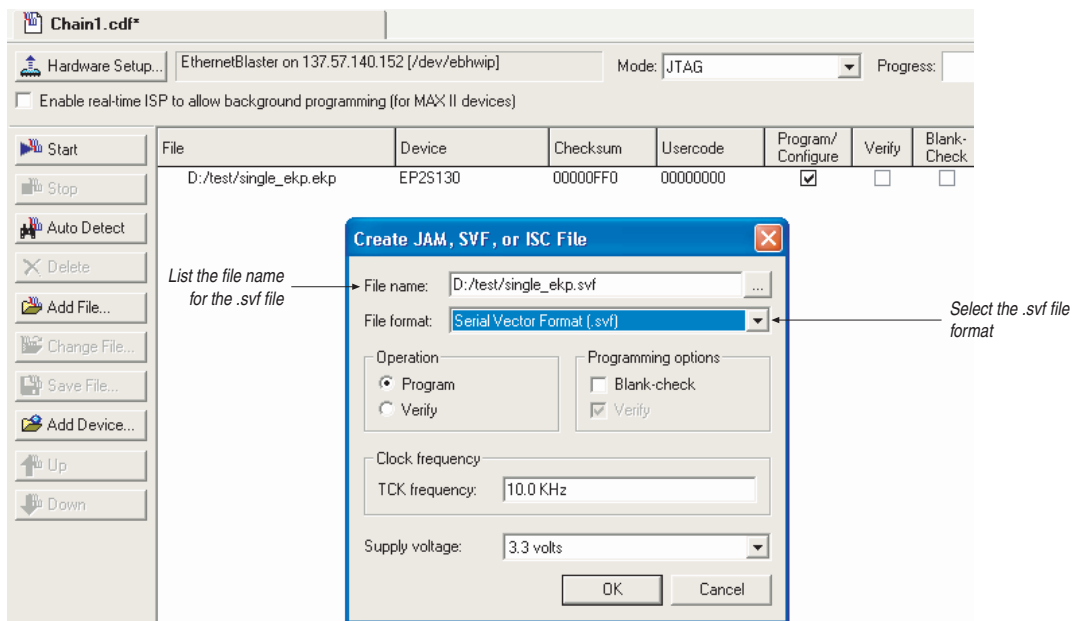
10. In the **Convert Programming Files** dialog box, click **OK**. The `<filename>.ekp` and encrypted configuration file are generated in the same project directory.
11. On the Tools menu, click **Programmer**. The **Programmer** dialog box displays.
12. In the **Mode** list, select **JTAG** as the programming mode.
13. Click **Hardware Setup**. The **Hardware Setup** dialog box is displayed.
  - a. In the currently selected hardware list, select **EthernetBlaster** as the programming hardware.
  - b. Click **Done**.
14. Click **Add File**. The **Select Programmer File** dialog box displays.
  - a. Type `<filename>.ekp` in the **File name** field.
  - b. Click **Open**.
15. Highlight the `.ekp` you added and click **Program/Configure**.
16. On the File menu, point to **Create/Update** and click **Create JAM, SVF, or ISC File**. The **Create JAM, SVF or ISC File** dialog box displays (Figure 7).
17. Select the file format required. For example, the **JEDEC STAPL Format (.jam)** for the `.ekp` in the **File format** field.
18. Type the file name in the **File name** field or browse to and select the file.
19. Click **OK** to generate the `.jam` file.

Figure 7. Create .jam from Single-Device .ekp



20. Repeat Step 17–19 to generate Serial Vector Format (.svf) of the .ekp. Use the default setting in the **Create JAM, SVF, or ISC File** dialog box when generating a SVF .ekp file (Figure 8).

Figure 8. Create .svf from Single-Device .ekp



## How to Generate the Single-Device .ekp and Encrypt the Configuration File with the Quartus II Software Version 6.1 or Later

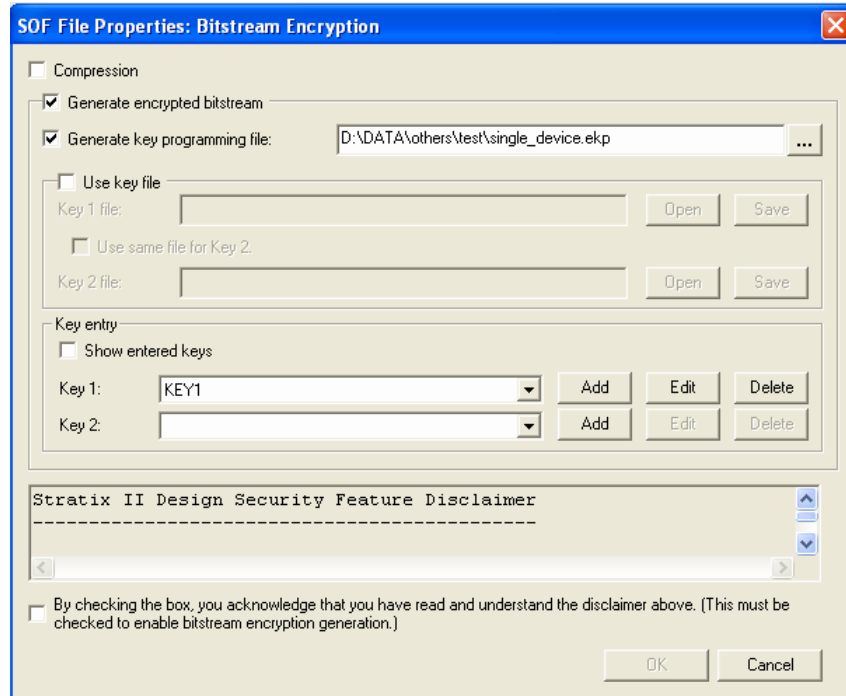
Perform the following steps to generate a single-device .ekp and encrypt your configuration file:

1. Obtain a license file for all versions of the Quartus II software and a Quartus II software patch for Quartus II software versions 5.0 through 7.0 to enable the Stratix II design security feature support from [Altera Technical Support](#).
2. Install the Quartus II software patch over the Quartus II software version 6.1 or later.



The Quartus II software version 6.1 requires a different software patch from the Quartus II software version 6.0 SP1. An updated patch can be obtained through [Altera Technical Support](#).

3. On the Tools menu, click **License Setup**. The **Options** dialog box displays the **License Setup** options.
4. In the **License file** field, enter the location and name of the license file or browse to and select the license file.
5. Click **OK**.
6. Repeat Steps 3–4 of “[How to Generate the Single-Device .ekp and Encrypt the Configuration File with the Quartus II Software Version 6.0 SP1](#)” on page 8.
  - a. Click **Properties**. The **SOF Files Properties: Bitstream Encryption** dialog box displays (as shown in [Figure 9](#)).
  - b. In the **SOF Files Properties: Bitstream Encryption** dialog box, turn on the **Generate encrypted bitstream** check box.
  - c. Turn on the **Generate key programming file** check box and type the .ekp path and file name in the text area or browse to and select *<filename>.ekp*.

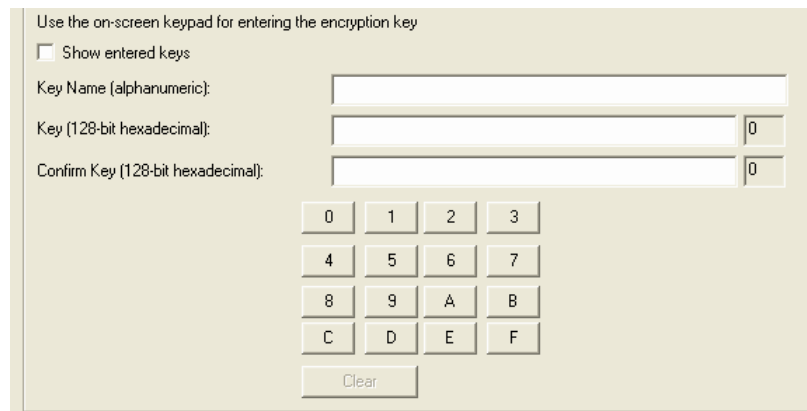
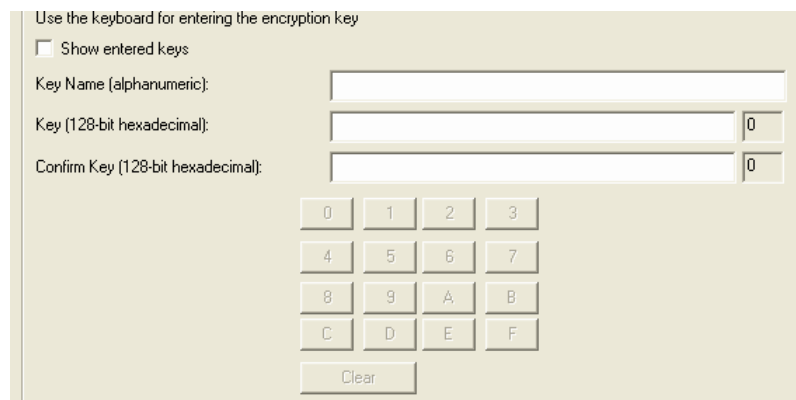
**Figure 9.** SOF File Properties Bitstream Encryption Dialog Box

7. Add the keys to the pull-down list either with a key file or using the **Add** button. The **Add** and **Edit** buttons bring up the **Key Entry** dialog box. The **Delete** button deletes the currently selected key from the drop down menu. (Figure 9)
  - a. Using the key file option allows you to specify one or two key files from which the corresponding pull-down list is populated. Use different files for Key 1 and Key 2 or use one key file for both. (Figure 10)



**Figure 13.** Key Entry Method

- i. The on-screen keypad allows you to enter the keys using the keypad shown in [Figure 14](#). Select a key and click on the on-screen keypad to enter values. You have the option of allowing the keys to be shown as they are entered . Confirmation of the keys is not required if the option is used. If you try to use the keyboard to enter keys, a pop-up notification appears and the key press will be ignored.
- ii. Enter the encryption key from the keyboard. ([Figure 15](#))

**Figure 14.** On-Screen Keypad**Figure 15.** Keyboard

8. Read the **Stratix II Design Security Feature Disclaimer**. If you agree to and acknowledge the **Stratix II Design Security Feature Disclaimer**, turn on the check box acknowledging you have read and accepted the disclaimer.





**Example 2** shows the two sets of keys stored in two different key files: key1 in **mykeys.key** and key2 in **otherkeys.key**.

#### Example 2.

---

```
quartus_cpf --key D:\SII_DS\mykeys.key:key1 --key
D:\SII_DS\otherkeys.key:key2 D:\SII_DS\test.sof D:\SII_DS\test.ekp
```

---

**Example 3** shows the two sets of keys stored in the same key file: key1 and key2 in **mykeys.key**.

#### Example 3.

---

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

---

### How to Generate the Multi-Device .ekp with the Quartus II Software Version 6.0 SP1 or Later

Perform the following steps to generate a multi-device **.ekp** and encrypt your configuration file:

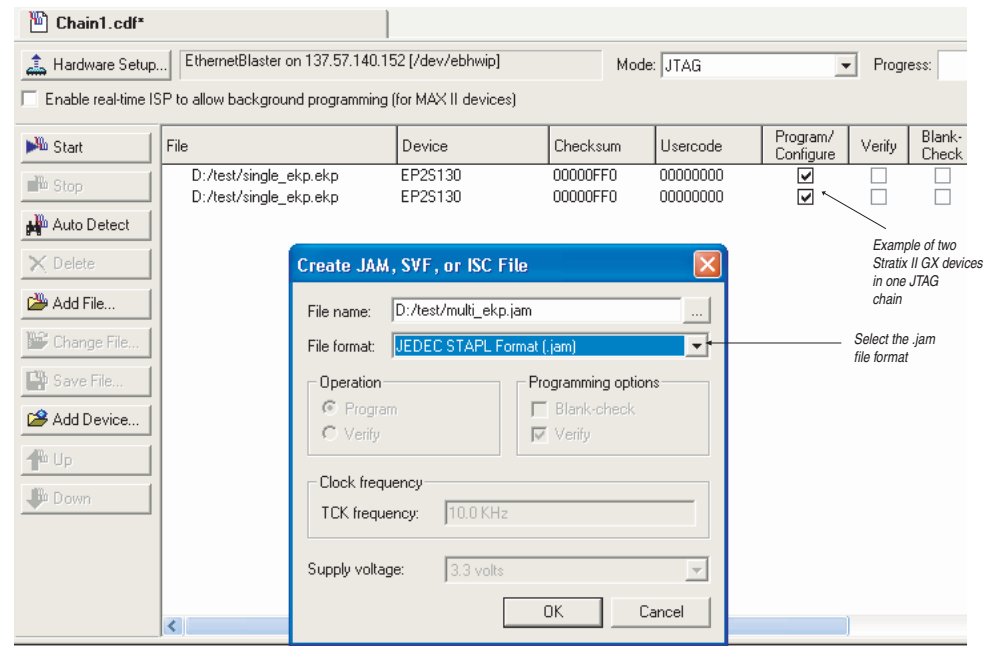
1. Start the Quartus II software.
2. Repeat Steps 10–13 of the “How to Generate the Single-Device .ekp and Encrypt the Configuration File with the Quartus II Software Version 6.0 SP1” on page 8.
3. Click **Add File**. The **Select Programmer File** dialog box displays.
  - a. Select the single-device **.ekp**, and type `<single_ekp>.ekp` in the **File name** field.
  - b. Click **Open**.



For the correct sequence of the devices in the same JTAG chain, you can use the **Auto-Detect** option in the Quartus II Programmer. If one of the devices is not key-programmed, you must not replace the device with the `<single_ekp>.ekp` in the Quartus II Programmer.

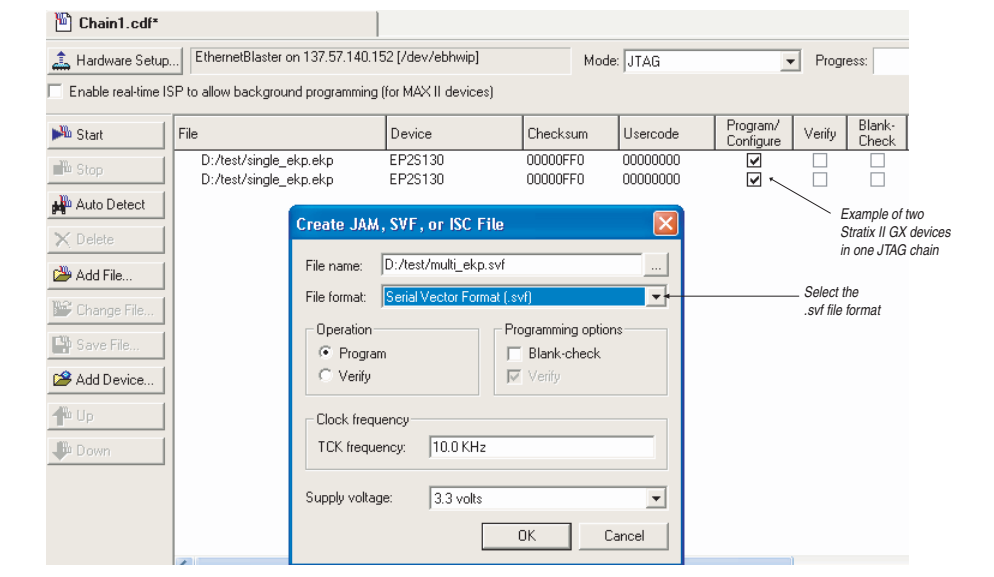
4. Repeat Step 3 for each device in the same chain. Ensure that the right device sequence is used when adding the **.ekp** to the **programmer** dialog box.
5. Highlight all **.ekp** files you added and click **Program/Configure** (Figure 17)
6. On the File menu, point to **Create/Update** and click **Create JAM, SVF, or ISC File**. The **Create JAM, SVF, or ISC File** dialog box displays. (Figure 17)
7. Select the file format required, for example, the **JEDEC STAPL Format (.jam)** for all **.ekp** files in the **File format** field.
8. Type the file name in the **File name** field or browse to and select the file.
9. Click **OK** to generate the **.jam** file.

**Figure 17.** Multi-Device Key Programming: JAM File Generation



10. Repeat Steps 7–9 to generate Serial Vector Format (.svf) for all the .ekp files. Use the default setting in the Create JAM, SVF, or ISC File dialog box when generating a SVF .ekp file (Figure 18).

**Figure 18.** Multi-Device Key Programming: SVF File Generation



## Step 2: Program the Security Key into a Stratix II or Stratix II GX Device

Before programming the security key into the Stratix II or Stratix II GX device, ensure the device is configured successfully with an unencrypted configuration file. The security key is one-time programmable through the JTAG interface. After you program the Stratix II or Stratix II GX device with the security key, you must configure it using an encrypted configuration file.

Attempting to configure a Stratix II or Stratix II GX device containing the security key with an unencrypted configuration file or configuration file encrypted with the wrong key causes configuration failure. If this occurs, the nSTATUS signal from the device pulses low and continues to reset itself.

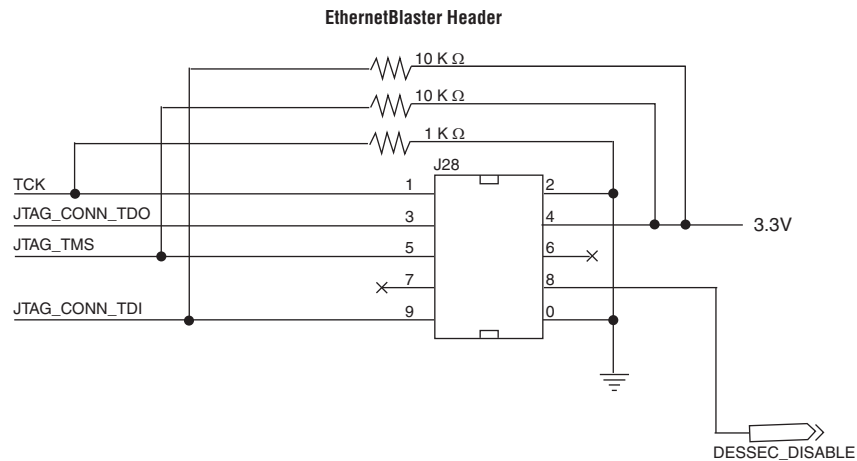
### How to Program the Security Key into the Stratix II or Stratix II GX Device

You can program the security key into the Stratix II and Stratix II GX device with on-board prototyping, volume production, and off-board prototyping and production solutions.

### Key Programming Using EthernetBlaster Communications Cable and the Quartus II Software

Connect the EthernetBlaster communications cable to the EthernetBlaster header, which are shown in [Figure 19](#). For additional information about connecting the EthernetBlaster communications cable, refer to the [EthernetBlaster Communications Cable User Guide](#).

**Figure 19.** EthernetBlaster Header *Notes (1),(2),(3)*



#### Notes to [Figure 19](#):

- (1) 1 KΩ pull-down resistor is added to the TCK while 10 KΩ pull-up resistors are added to the TMS and TDI signals for security key programming.
- (2) Pin 8 of the EthernetBlaster connects to signal DESSEC\_DISABLE of the voltage regulator circuit used to control  $V_{CCPDB}$ .
- (3) DESSEC\_DISABLE is a standard, active-low 3.3-V LVTTTL control signal. When DESSEC\_DISABLE is low, it sets the output of the voltage regulator to 3.7 V.

## How to Perform Single-Device Key Programming with the Quartus II Software Version 6.0 SP1

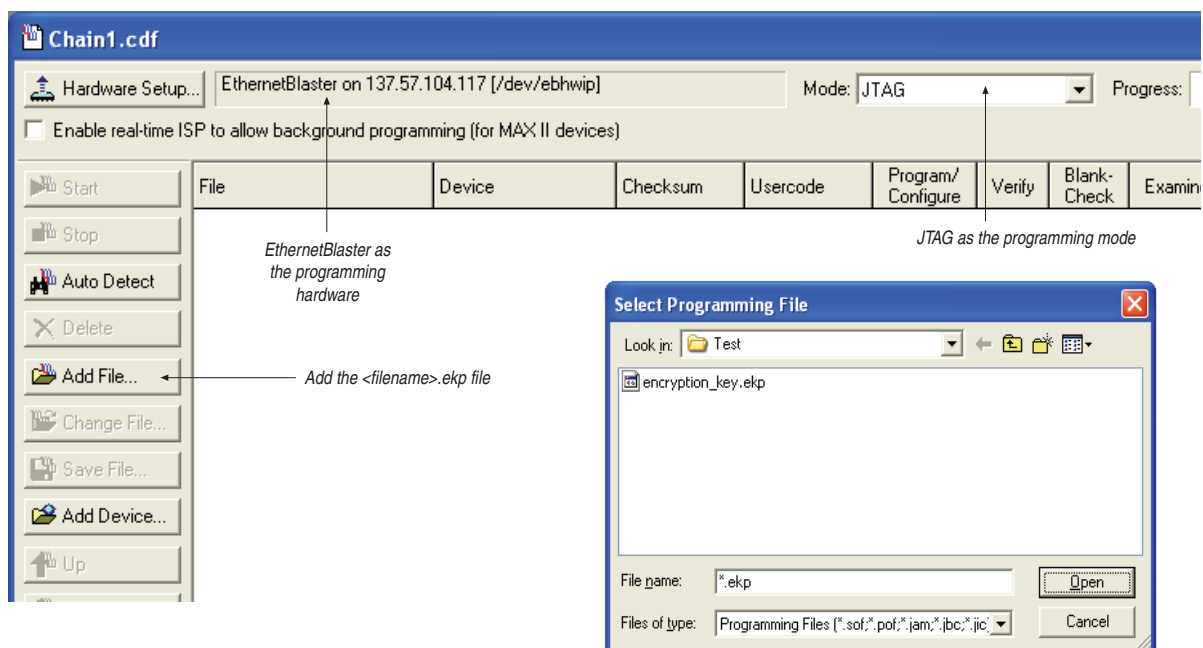
To perform single-device key programming with the Quartus II software through the EthernetBlaster, perform the following steps:

1. Check the firmware version of the EthernetBlaster. Verify that the JTAG firmware build number is 101 or greater. If the version precedes build number 101, apply the firmware upgrade.

 Apply the firmware upgrade (**EBFW100101.tar.gz**) to the EthernetBlaster unit. This updates the JTAG Firmware to Build 101. For firmware upgrade instructions, refer to the *EthernetBlaster Communications Cable User Guide*.

2. Start the Quartus II software.
3. On the Tools menu, click **Programmer**. The **Programmer** dialog box displays (Figure 20).
4. In the **Mode** list, select **JTAG** as the programming mode.

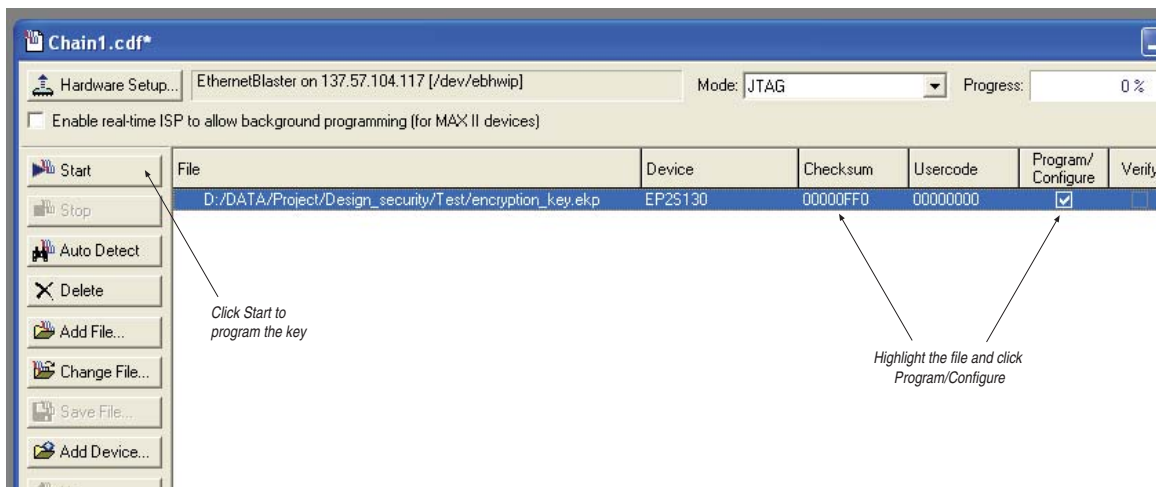
**Figure 20.** Key Programming Using EthernetBlaster and the Quartus II Software



5. Click **Hardware Setup**. The **Hardware Setup** dialog box is displayed.
  - a. In the **Currently selected hardware** list, select **EthernetBlaster** as the programming hardware.
  - b. Click **Done**.
6. Click **Add File**. The **Select Programmer File** dialog box displays.
  - a. Type **<filename>.ekp** in the **File name** field.
  - b. Click **Open**.

7. Highlight the .ekp you added and click **Program/Configure** (Figure 21).
8. Click **Start** to program the key.
9. The Quartus II software message window provides information about the success or failure of the key programming operation.

**Figure 21.** Programming the Key



### How to Perform Single-Device Key Programming with the Quartus II Software Version 6.1 or Later With GUI Interface

To perform single-device key programming with the Quartus II software version 6.1 GUI interface through the EthernetBlaster, refer to Steps 1–9 in “How to Perform Single-Device Key Programming with the Quartus II Software Version 6.0 SP1” on page 21

### How to Perform Single-Device Key Programming with the Quartus II Software Version 6.1 or Later With Command-Line Interface


To perform single-device key programming with the Quartus II software version 6.1 command-line interface through the EthernetBlaster, perform the following steps:

1. Execute Step 1 in “How to Perform Single-Device Key Programming with the Quartus II Software Version 6.0 SP1” on page 21
2. To determine the EthernetBlaster cable port number that is connected to the JTAG server, type `quartus jli -n` at the command line prompt.
3. With the `single_ekp.jam` file generated in Step 1, execute key programming to a single device with this command line:
  - `quartus jli -cn single_ekp.jam -aKEY_Program (n is the port number returned with the -n option).`
4. The Quartus II command line executable provides information about the success or failure of the key programming operation.

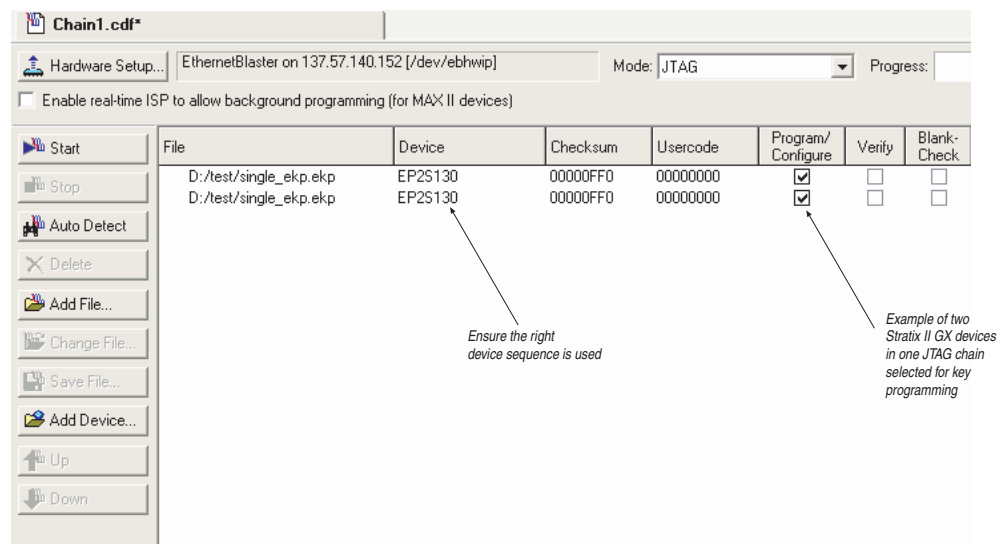
## How to Perform Multi-Device Key Programming with the Quartus II Software Version 6.0 SP1

To perform multi-device key programming with the Quartus II software through the EthernetBlaster, perform the following steps:

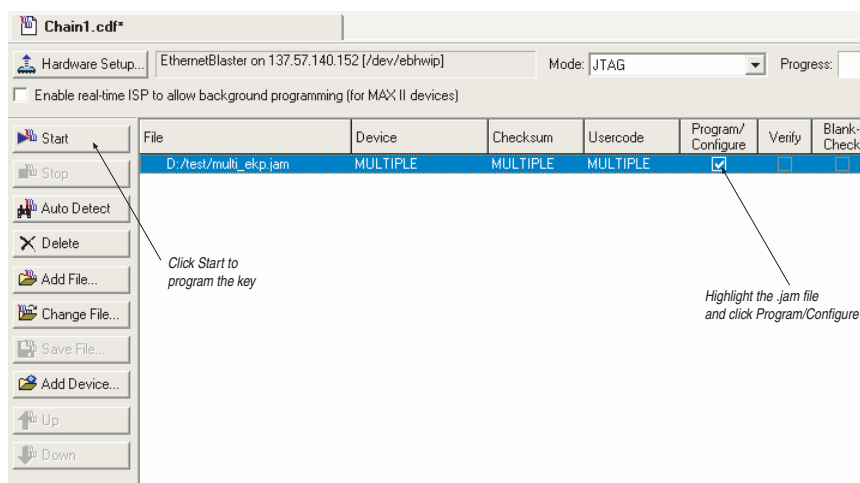
1. Start the Quartus II software.
2. Repeat Steps 3–5 in “How to Perform Single-Device Key Programming with the Quartus II Software Version 6.0 SP1” on page 21.
3. Click **Add File**. The **Select Programmer File** dialog box displays.
  - a. Programming using single-device **.ekp** files:
    - i. Type *<single\_device>.ekp* in the **File name** field.
    - ii. Click **Open**.
    - iii. Repeat Steps i–ii for the number of devices in the same chain.
    - iv. Highlight the **.ekp** files you added and click **Program/Configure** (Figure 22).

 For the correct sequence of the devices in the same JTAG chain, use the **Auto-Detect** option in the Quartus II Programmer.

**Figure 22.** Multi-Device Key Programming with **.ekp** Files



- b. Programming with multi-device **.jam** files:
      - i. Type *<multi\_device>.jam* in the **File name** field.
      - ii. Click **Open**.
      - iii. Highlight the **.jam** you added and click **Program/Configure** (Figure 23).
4. Click **Start** to program the key.

**Figure 23.** Multi-Device Key Programming with JAM File

5. The Quartus II software message window provides information about the success or failure of the key programming operation.

### How to Perform Multi-Device Key Programming with the Quartus II Software Version 6.1 or Later With GUI Interface

To perform multi-device key programming with the Quartus II software version 6.1 GUI interface through the EthernetBlaster, refer to Steps 1–5 in “[How to Perform Multi-Device Key Programming with the Quartus II Software Version 6.0 SP1](#)” on page 23.

### How to Perform Multi-Device Key Programming with the Quartus II Software Version 6.1 or Later With Command-Line Interface

To perform multi-device key programming with the Quartus II software version 6.1 command-line interface through the EthernetBlaster, perform the following steps:

1. To determine the EthernetBlaster cable port number that is connected to the JTAG server, type `quartus_jli -n` at the command line prompt.
2. With the **multi\_ekp.jam** file generated in Step 1, execute key programming to a single device with this command line:
  - `quartus_jli -cn multi_ekp.jam -aKEY_PROGRAM` (n is the port number returned with the `-n` option).
3. The Quartus II command line executable provides information about the success or failure of the key programming operation.

### Key Programming Using In-Circuit Tester (ICT)

This mode is used after the device is attached to your board. The contract manufacturer ICT fixture is used to program non-volatile memory. The ICT fixture provides one pin (probe) for 3.7 V, which is generated and controlled by the ICT fixture program.



The connection diagrams for the  $V_{CCPD8}$  power supply options and the reference design for power multiplexer requirements are shown in B and D in [Figure 2](#) and [Figure 3](#), respectively.

- Information about creating `.svf` files to support multi-device programming is described in “[How to Generate the Multi-Device .ekp with the Quartus II Software Version 6.0 SP1 or Later](#)” on page 18.

### Key Programming with JTAG Technologies

Use an `.svf` (`.ekp` in SVF format) and a JT 37xx boundary-scan controller in combination with a JT 2147 QuadPod system for design security programming. Connection diagrams for JTAG Technologies are similar to ICT for  $V_{CCPD8}$  power supply options and the power multiplexer.

- For more information about the procedures for JTAG programming, refer to the SVF Programming the Altera Stratix II Design Security Key application note on the JTAG Technologies website ([www.jtag.com](http://www.jtag.com)). Use the same steps to perform key programming with Stratix II GX devices.

- Information about creating `.svf` files to support multi-device programming is described in “[How to Generate the Multi-Device .ekp with the Quartus II Software Version 6.0 SP1 or Later](#)” on page 18.

### Key Programming with System General and other Third-Party Programming Vendors

System General T9600 programming equipment supports key programming for Stratix II and Stratix II GX devices.

The following two files are required:

- JAM STAPL file (`.ekp` in JAM format)
- Encrypted `.rbf` (contains encrypted configuration data). This `.rbf` is used to further verify the keys that are programmed into the device. It can be any design file encrypted with the same set of keys used to generate the JAM STAPL file.

Socket adapter availability for Stratix II and Stratix II GX packages is shown in [Table 4](#).

**Table 4.** Socket Availability for Stratix II and Stratix II GX Packages

Device	Package
Stratix II	F484, F672, F780, F1020, and F1508
Stratix II GX	F780, F1152, and F1508

Key programming service can be obtained from most Altera distributors. In general, refer to [Altera Technical Support](#) for updated programming support status.

- Documentation about how encryption keys are programmed into Stratix II and Stratix II GX devices with System General programming equipment is available at [www.sg.com.tw](http://www.sg.com.tw)

## Step 3: Configure the Stratix II and Stratix II GX Devices

The final step is to configure the protected Stratix II and Stratix II GX device with the encrypted configuration file.

During configuration, the encrypted configuration data is sent to the Stratix II or Stratix II GX device. Using the previously stored security key, the device decrypts the configuration data and uses the unencrypted data to configure itself. Only configuration files encrypted with the correct security key are accepted by the device for successful configuration. Without a correct security key, a stolen encrypted file is useless.

## Security Mode Verification

You can verify the security mode of Stratix II and Stratix II GX devices with the `KEY_VERIFY` JTAG instruction.

[Table 5](#) shows the security modes available in Stratix II and Stratix II GX devices.

**Table 5.** Security Modes

Security Modes	Description
Non-volatile key	Secure operation with one-time programmable (OTP) security key programmed. This mode accepts both encrypted and unencrypted configuration bitstreams.  Use the unencrypted configuration bitstream support for board-level testing only.
Non-volatile key with tamper-protection bit set (1)	Secure operation in tamper resistant mode with OTP security key programmed. Only encrypted configuration bitstreams are allowed to configure the device.  Tamper protection disables JTAG configuration with unencrypted configuration bitstream.
No-key operation	Only unencrypted configuration bitstreams are allowed to configure the device.

**Note to Table 5:**

(1) Enabling the tamper protection bit disables the test mode in Stratix II and Stratix II GX devices. This process is irreversible and prevents Altera from conducting carry-out failure analysis if the test mode is disabled. Contact [Altera Technical Support](#) to enable the tamper protection bit.

[Table 6](#) shows the instruction code of the `KEY_VERIFY` JTAG instruction.

**Table 6.** `KEY_VERIFY` JTAG Instruction

JTAG Instruction	Instruction Code	Description
<code>KEY_VERIFY</code>	00 0001 0011	This instruction connects the key verification scan registers between TDI and TDO.

Table 7 shows the contents of the key verification scan registers on Stratix II or Stratix II GX device.

**Table 7.** Key Verification Scan Registers *(Note 1)*

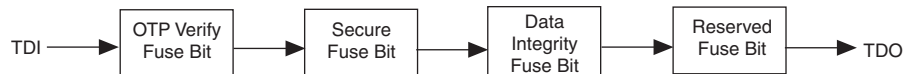
Register	Description
OTP Verify Fuse Bit	OTP Poly Fuse Key has successfully been stored. User is not allowed to enter another key.
Secure Fuse Bit	Indicates that bypass is no longer allowed except in Test Mode (for example, Test mode access is allowed). User is not allowed to enter another key.
Data Integrity Fuse Bit	Indicates that bypass and Test Mode Access is no longer allowed. User is not allowed to enter another key.
Reserved Fuse Bit	Not available.

**Note to Table 7**

(1) The key verification scan registers are active high.

Figure 24 shows the key verification scan registers in a JTAG chain when using the KEY\_VERIFY JTAG instruction.

**Figure 24.** Key Verification Scan Registers In a JTAG Chain.



Example 4 shows an example of a .jam used to verify the security mode of a single Stratix II or Stratix II GX device.

**Example 4.** Example of a .jam Used to Verify the Security Mode of a Single Stratix II or Stratix II GX Device

```

'Key Verification in JAM format
BOOLEAN verify_reg[4];

IRSCAN 10, $013;
WAIT 100 USEC;
DRSCAN 4, $0, CAPTURE verify_reg[3..0];

PRINT "Security Mode Verification for Single Stratix II or
Stratix II GX Device ";
IF (INT(verify_reg[3..0]) == 0) THEN PRINT "Security Mode: No Key";
IF (INT(verify_reg[3..0]) == 8 || INT(verify_reg[3..0]) == 12) THEN
PRINT "Security Mode: Non-Volatile Key";
IF (INT(verify_reg[3..0]) == 14 ) THEN
PRINT "Security Mode: Non-volatile Key with Tamper Protection
Bit Set";
  
```

## Supported Configuration Schemes

The design security feature is available in all configuration methods, except in PPA and JTAG. Therefore, you can use the design security feature in FPP mode (when using an external controller, such as a MAX II device or a microprocessor and a flash memory), or in AS and PS configuration schemes.

Table 8 summarizes the configuration schemes that support the design security feature.

**Table 8.** Design Security Configuration Schemes Availability

Configuration Scheme	Configuration Method	Design Security
FPP	MAX II device or microprocessor and flash memory	✓(1)
	Enhanced configuration device	—
AS	Serial configuration device	✓
PS	MAX II device or microprocessor and flash memory	✓
	Enhanced configuration device	✓
	Download cable	✓(2)
PPA	MAX II device or microprocessor and flash memory	—
JTAG	MAX II device or microprocessor and flash memory	—
	Download cable	—


**Notes to Table 8:**


- (1) In this mode, the host system must send a `DCLK` that is 4× the data rate.
- (2) The MicroBlaster™ tool is required to execute encrypted PS configuration with an `.rbf` through a ByteBlaster™ II or ByteBlaster MV download cable. For more information, refer to the [Configuration Center](#). The Quartus II software version 6.1 supports encrypted `.rbf` configuration through an Altera download cables.

The design security feature is not available in the PPA configuration scheme because the PPA scheme does not have a clock signal. When using the design security feature, a clock signal that is four times the data rate is required to process the byte-wide encrypted configuration data.

 For more information about the MAX II device and flash memory configuration method, refer to the [MAX Series Configuration Controller Using Flash Memory White Paper](#).

In addition, if your system contains a common flash interface (CFI) flash memory, you are able to use it for the device configuration as well. The MAX II parallel flash loader (PFL) feature provides an efficient method to program CFI flash memory through the JTAG interface.

 If you use the MAX II PFL and flash memory in FPP configuration with design security feature enabled, the PFL automatically generates a `DCLK` that is 4× the data rate.

 For more information about PFL, refer to [AN 386: Using the MAX II Parallel Flash Loader with the Quartus II Software](#).

In JTAG mode, the configuration data does not use the same interface that is used in the FPP, AS, and PS configuration schemes. Therefore, design security is not available in JTAG-based configurations.

You can use the design security feature with other configuration features, such as the compression and the remote system upgrade features. When compression is used with the design security feature, the configuration file is first compressed and then encrypted in the Quartus II software. During configuration, the device first decrypts and then uncompresses the configuration file.

You can perform boundary-scan testing or use the SignalTap® II Logic Analyzer to analyze functional data in the device. However, JTAG configuration is not possible after the security key is programmed into the Stratix II device.

When using the SignalTap II Logic Analyzer, you must first configure the device with an encrypted configuration file using PS, FPP, or AS configuration modes. Your design must contain at least one instance of the SignalTap II Logic Analyzer. After the device is configured with a SignalTap II Logic Analyzer instance in your design, open the SignalTap II Logic Analyzer window in the Quartus II software to scan the chain and acquire data over the JTAG.

## Serial FlashLoader Support with Encryption Enabled

Altera provides an in-system programming solution for serial configuration devices called Serial FlashLoader (SFL). The SFL megafunction is available with the Quartus II software version 6.0 SP1 or later. Instantiate the SFL block to your design and have the flexibility to update your design stored in the serial configuration device without having to reprogram the configuration device through the AS interface.

The following procedures are for the SFL megafunction with the encryption feature enabled in a single device chain.

1. Start the Quartus II software.
2. Instantiate the SFL megafunction in your device top-level design.

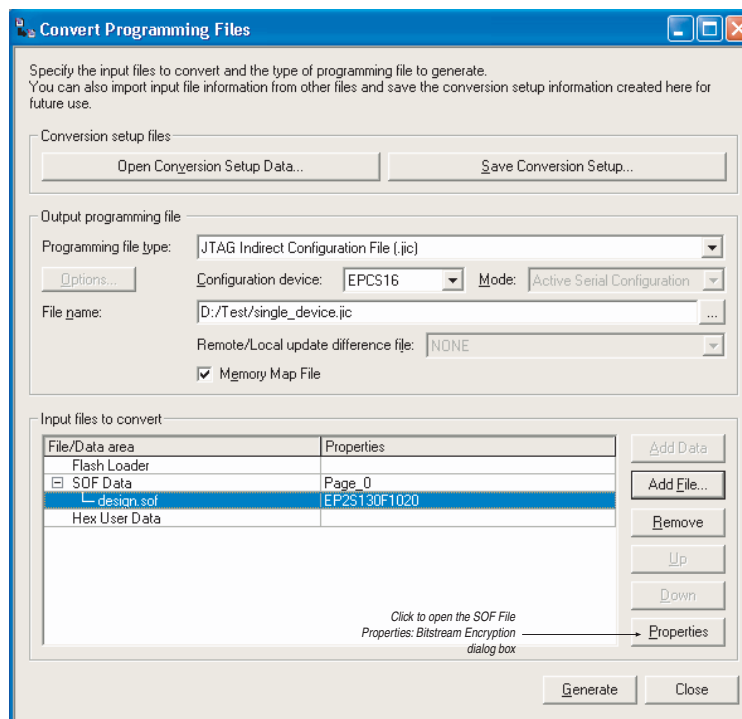


For the complete procedures, refer to the "Instantiating SFL Megafunction in the Quartus II Software" section in *AN 370: Using the Serial FlashLoader With the Quartus II Software*.


3. Compile your design with one of the following options. An unencrypted `.sof` is generated.
  - On the Processing menu, click **Start Compilation**.
  - On the Processing menu, point to **Start** and click **Start Assembler**.

4. Perform the following steps to convert a **.sof** to a **.jic**:
  - a. On the File menu, click **Convert Programming Files**. The **Convert Programming Files** dialog box appears (Figure 25).
  - b. In the **Programming file type** list, select **JTAG Indirect Configuration File (.jic)**.
  - c. In the **Configuration device** field, specify the targeted serial configuration device.
  - d. In the **File name** field, browse to the target directory and specify an output file name.
  - e. Highlight **SOF Data** in under **Input files to convert**.


**Figure 25.** JIC File Generation




- f. Click **Add File**.
- g. Select the **.sof** that you want to convert to a **.jic**.
- h. Click **OK**.
- i. Click on the **.sof** name to encrypt the **.sof**.

 Refer to the Steps 4–9 in “How to Generate the Single-Device .ekp and Encrypt the Configuration File with the Quartus II Software Version 6.1 or Later” on page 13 to encrypt the .sof.

- j. Highlight **FlashLoader** and click **Add Device**.
  - k. Click **OK**. The **Select Devices** dialog box displays.
  - l. Select the targeted device you are using to program the serial configuration device.
  - m. Click **OK**.
5. Program the serial configuration device with the encrypted **.jic**.

 For more information about programming the serial configuration device with the **.jic** you just created, add the file to the Quartus II Programmer window and refer to the steps in the "Programming Serial Configuration Devices Using the Quartus II Programmer & JIC Files" section of *AN 370: Using the Serial FlashLoader With the Quartus II Software*.

6. Program the security keys into the device.

 To program the security keys to a single device, follow the steps in “How to Perform Single-Device Key Programming with the Quartus II Software Version 6.0 SP1” on page 21.

7. Set the  $V_{CCPD8}$  to 3.3V upon the completion of key programming. The encrypted device is configured by the programmed serial configuration device.

 To program the key with a **.jam**, you must convert the **.jic** to a **.jam**.

 For more information about converting a **.jic** to a **.jam**, refer to the "Converting JIC Files to JAM Files in the Quartus II Software" section in *AN 370: Using the Serial FlashLoader With the Quartus II Software*.

## Specifying Configuration Schemes

To enable the design security feature, you must specify the configuration schemes used by setting the MSEL [ ] pin settings, which are shown in [Table 9](#).

**Table 9.** Stratix II and Stratix II GX Configuration Schemes When Using Design Security

MSEL3	MSEL2	MSEL1	MSEL0	Configuration Scheme
0	0	1	0	PS
0	1	1	0	Remote system upgrade PS (1)
1	0	0	0	Fast AS (40 MHz) (2)
1	0	0	1	Remote system upgrade fast AS (40 MHz) (2)
1	0	1	1	FPP with decompression, design security feature enabled or both(3)
1	1	0	0	Remote system upgrade FPP with decompression, design security feature enabled or both(1), (3)

**Table 9.** Stratix II and Stratix II GX Configuration Schemes When Using Design Security

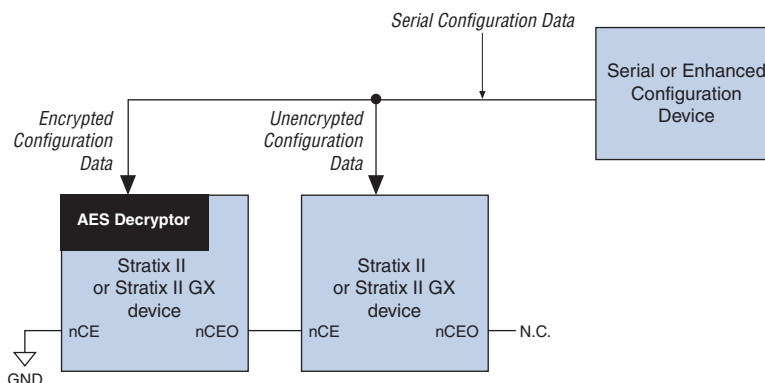
MSEL3	MSEL2	MSEL1	MSEL0	Configuration Scheme
1	1	0	1	AS (20 MHz) (2)
1	1	1	0	Remote system upgrade AS (20 MHz) (2)

**Notes to Table 9:**

- (1) These schemes require that you drive the `RUNLU` pin to specify either remote update or local update mode. For more information about remote system upgrades in Stratix II and Stratix II GX devices, refer to the *Remote System Upgrades with Stratix II & Stratix II GX Devices* chapter in the *Stratix II Device Handbook* or *Stratix II GX Device Handbook*.
- (2) Only the EPCS16 and EPCS64 devices support a `DCLK` up to 40 MHz. Other EPCS devices support a `DCLK` up to 20 MHz. For more information, refer to the *Serial Configuration Devices Data Sheet*.
- (3) These modes are only supported when using a MAX II device/microprocessor and flash memory for configuration. In these modes, the host system must output a `DCLK` that is 4× the data rate.

## Considerations When Choosing a Configuration Scheme

As shown in [Figure 26](#), in a serial configuration scheme (AS or PS), you can cascade a series of Stratix II or Stratix II GX devices that use the design security feature (accepts encrypted data) along with devices that do not use the design security feature, such as other Altera devices that accept unencrypted data in the same configuration chain.

**Figure 26.** Stratix II Series Serial Configuration

When using FPP, the design security feature must be either enabled or disabled for all Stratix II or Stratix II GX devices in the chain. The design security feature cannot be selectively enabled for individual devices in the chain because of the relationship between `DATA` and `DCLK`, which is discussed in the following section. If the chain contains devices that do not support the design security feature, you must use a serial configuration scheme.

However, the `DATA` and `DCLK` relationship in FPP mode when using the decompression feature is the same as when using the design security feature. If all devices in the chain use the decompression feature, the design security feature can be selectively enabled or disabled for each device.

### DCLK Considerations When Using the FPP Configuration Scheme

The FPP configuration scheme with the design security feature enabled requires that an external host be used, such as a MAX II device or a microprocessor to control the `DATA` and `DCLK` flow to the Stratix II and Stratix II GX devices.



The FPP configuration scheme without the design security feature enabled could use either an external host and flash memory, or it could use an enhanced configuration device.

DCLK is the clock source that is used to clock the configuration process. Configuration data is received on the DATA [7 . . 0] pins.

If you use the Stratix II or Stratix II GX design security feature with the FPP configuration scheme, the external host must be able to send a DCLK frequency that is four times the data rate.

The 4× DCLK signal does not require an additional pin and is sent on the DCLK pin. The maximum DCLK frequency supported in Stratix II devices is 100 MHz and results in the maximum data rate shown in [Equation 1](#):

**Equation 1.**

---

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

---

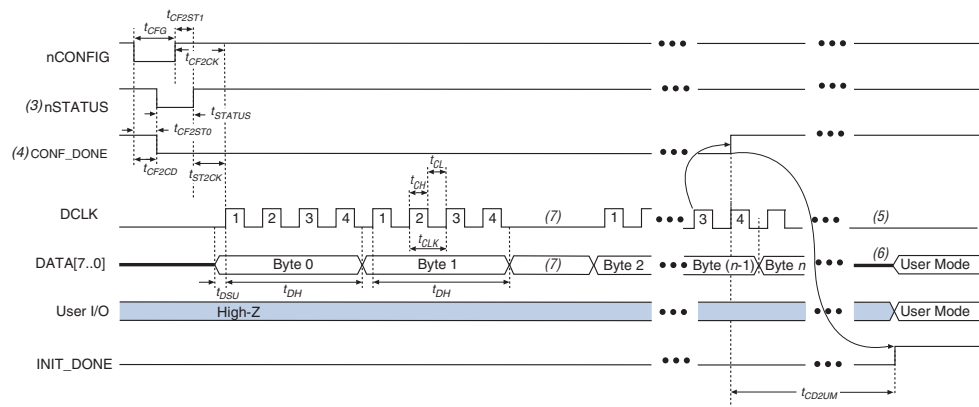
When using the design security feature, the first configuration data byte is latched into the device on the first DCLK rising edge. Subsequent data bytes are latched four clock cycles after each previous data byte (that is, data is latched in on the first, fifth, and ninth DCLK rising). The configuration clock (DCLK) speed must be below the frequency specified in [Equation 1](#) to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If you must stop DCLK while using the Stratix II or Stratix II GX design security feature with the FPP configuration scheme, it can only be stopped three clock cycles after the last data byte was latched into the Stratix II device.

Stopping DCLK three clock cycles after the last data byte was latched into the Stratix II device allows the configuration circuit enough clock cycles to process the last byte of latched configuration data. When the clock restarts, data must be present on the DATA [7 . . 0] pins prior to sending the first DCLK rising edge.

## Timing Waveform for FPP Configuration

[Figure 27](#) shows the timing waveform for FPP configuration when using a MAX II device or a microprocessor as an external host. This waveform shows the timing when the design security feature is enabled.

**Figure 27.** FPP Configuration Timing Waveform with Design Security Feature Enabled *Notes (1), (2)***Notes to Figure 27:**

- (1) This timing waveform must be used when the design security feature is used.
- (2) The beginning of this waveform shows the device in user mode. In user mode, `nCONFIG`, `nSTATUS`, and `CONF_DONE` are at logic-high levels. When `nCONFIG` is pulled low, a reconfiguration cycle begins.
- (3) Upon power-up, the Stratix II or Stratix II GX device holds `nSTATUS` low during the power-on reset (POR) delay.
- (4) Upon power-up, before and during configuration, `CONF_DONE` is low.
- (5) `DCLK` must not be left floating after configuration. It must be driven high or low, whichever is more convenient.
- (6) `DATA [7..0]` pins are available as user I/O pins after configuration and the state of these pins depends on the dual-purpose pin settings.
- (7) If required, `DCLK` can be paused. When `DCLK` restarts, the external host must provide data on the `DATA [7..0]` pins prior to sending the first `DCLK` rising edge.

## Timing Parameters with Design Security Feature Enabled

**Table 10** defines the timing parameters for Stratix II and Stratix II GX devices that provide the FPP configuration scheme with the design security feature enabled.

**Table 10.** FPP Timing Parameters for Stratix II and Stratix II GX Devices with Design Security Feature Enabled *Notes (1), (2)* (Part 1 of 2)

Symbol	Parameter	Min	Max	Unit
$t_{POR}$	POR delay	12	100	ms
$t_{CF2CD}$	<code>nCONFIG</code> low to <code>CONF_DONE</code> low	—	800	ns
$t_{CF2ST0}$	<code>nCONFIG</code> low to <code>nSTATUS</code> low	—	800	ns
$t_{CFG}$	<code>nCONFIG</code> low pulse width	2	—	$\mu$ s
$t_{STATUS}$	<code>nSTATUS</code> low pulse width	10	100 (3)	$\mu$ s
$t_{CF2ST1}$	<code>nCONFIG</code> high to <code>nSTATUS</code> high	—	100 (3)	$\mu$ s
$t_{CF2CK}$	<code>nCONFIG</code> high to first rising edge on <code>DCLK</code>	100	—	$\mu$ s

**Table 10.** FPP Timing Parameters for Stratix II and Stratix II GX Devices with Design Security Feature Enabled *Notes (1), (2)* (Part 2 of 2)

Symbol	Parameter	Min	Max	Unit
$t_{ST2CK}$	$nSTATUS$ high to first rising edge of DCLK	2	—	$\mu s$
$t_{DSU}$	Data setup time before rising edge on DCLK	5	—	ns
$t_{DH}$	Data hold time after rising edge on DCLK	30	—	ns
$t_{CH}$	DCLK high time	4	—	ns
$t_{CL}$	DCLK low time	4	—	ns
$t_{CLK}$	DCLK period	10	—	ns
$f_{MAX}$	DCLK frequency	—	100	MHz
$t_{DATA}$	Data rate	—	200	Mbps
$t_R$	Input rise time	—	40	ns
$t_F$	Input fall time	—	40	ns
$t_{CD2UM}$	CONF_DONE high to user mode (4)	20	100	$\mu s$
$t_{CD2CU}$	CONF_DONE high to CLKUSR enabled	$4 \times$ maximum DCLK period	—	—
$t_{CD2UMC}$	CONF_DONE high to user mode with CLKUSR option on	$t_{CD2CU} + (299 \times \text{CLKUSR period})$	—	—

**Notes to Table 10:**

- (1) This information is preliminary.
- (2) These timing parameters must be used when the design security feature is used.
- (3) This value is obtainable if you do not delay configuration by extending the  $nCONFIG$  or  $nSTATUS$  low pulse width.
- (4) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for initializing the device.

## US Export Controls

The US export controls for the Stratix II family of devices are generally classified under US Export Control Classification Numbers (ECCN) 3A001.a.7 or 3A991.d. Although Stratix II devices may perform decryption, the export control classification of the devices does not change as the decryption capability is only used to protect the configuration bitstream. Altera's Quartus II development tools (6.0 SP1 or later), which encrypt the configuration bitstream, are formally classified under US ECCN 5D002 c.1 and subject to export under license exception ENC as a "retail" commodity to most countries. Contact [opexp\\_imp@altera.com](mailto:opexp_imp@altera.com) with any export related questions.

## Conclusion

As design security requirements increase, FPGAs are moving from glue logic to providing critical system functions. Stratix II and Stratix II GX devices address this concern by providing built-in design security. Stratix II and Stratix II GX devices not only offer high density, fast performance, and cutting-edge features to meet your design needs, but also protect your designs against intellectual property (IP) theft and tampering of your configuration files.

## Document Revision History

Table 11 shows the revision history for this application note.

**Table 11.** Document Revision History

Date and Document Version	Changes Made	Summary of Changes
August 2009 v2.3	<ul style="list-style-type: none"> <li>■ Minor changes to <a href="#">Table 1 on page 3</a></li> <li>■ Updated “<a href="#">How to Generate the Single-Device .ekp and Encrypt the Configuration File with the Quartus II Software Version 6.0 SP1</a>” on <a href="#">page 8</a>, “<a href="#">How to Generate the Single-Device .ekp with the Quartus II Software Version 6.1 or Later With Command-Line Interface</a>” on <a href="#">page 17</a></li> <li>■ Updated <a href="#">Figure 19 on page 20</a></li> <li>■ Updated “<a href="#">Key Programming Using EthernetBlaster Communications Cable and the Quartus II Software</a>” on <a href="#">page 20</a></li> <li>■ Minor changes to “<a href="#">Key Programming with System General and other Third-Party Programming Vendors</a>” on <a href="#">page 25</a></li> <li>■ Updated <a href="#">Table 4 on page 25</a></li> <li>■ Updated <a href="#">Table 8 on page 28</a></li> <li>■ Added “<a href="#">Security Mode Verification</a>” on <a href="#">page 26</a></li> <li>■ Deleted <a href="#">Figure 26 “Flash Loader”</a>.</li> </ul>	—
July 2008 v2.2	<ul style="list-style-type: none"> <li>■ Made minor additions to “<a href="#">Voltage Requirements</a>” section.</li> <li>■ Added Quartus II support information.</li> <li>■ Made minor additions to “<a href="#">Supported Configuration Schemes</a>” section.</li> <li>■ Updated <a href="#">Figure 17</a>.</li> </ul>	—
August 2007 v2.1	Corrected <a href="#">Figure 2</a> .	—

**Table 11.** Document Revision History

Date and Document Version	Changes Made	Summary of Changes
February 2007 v2.0	<ul style="list-style-type: none"> <li>■ Made minor changes to text on page 8.</li> <li>■ Made minor changes to text on page 17.</li> <li>■ Added new section “How to Generate the Single-Device .ekp File and Encrypt the Configuration File Using the Quartus II Software Version 6.1 or Later,” on page 12.</li> <li>■ Added new section “How to Generate the Single-Device .ekp File Using the Quartus II Software Version 6.1 or Later With Command-Line Interface,” on page 17.</li> <li>■ Added new section “How to Perform Single-Device Key Programming Using the Quartus II Software Version 6.1 or Later With GUI Interface,” on page 22</li> <li>■ Added new section “How to Perform Single-Device Key Programming Using the Quartus II Software Version 6.1 or Later With Command-Line Interface,” on page 22</li> <li>■ Added new section “How to Perform Multi-Device Key Programming Using the Quartus II Software Version 6.1 or Later With GUI Interface,” on page 24</li> <li>■ Added new section “How to Perform Multi-Device Key Programming Using the Quartus II Software Version 6.1 or Later With Command-Line Interface,” on page 24</li> <li>■ Replaced Figure 6.</li> <li>■ Replaced Figure 10 and Figure 11 (previously Figure 12 and Figure 13).</li> <li>■ Added Figure 9, Figure 10, Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6.</li> </ul>	Major updates to the document including new sections and accompanying figures.
October 2006 v1.5	Update to include Stratix II GX.	—



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)  
Technical Support  
[www.altera.com/support](http://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 relying on any published information and before placing orders for products or services.

