LatticeXP2 Configuration Encryption and Security Usage Guide

April 2013 Technical Note TN1142

Introduction

Unlike a volatile FPGA, which requires an external boot-prom to store configuration data, the LatticeXP2™ devices are non-volatile and have on-chip configuration Flash. Once programmed (either by JTAG or SPI port), this data is a part of the FPGA device and can be used to self-download the SRAM portion without requiring any additional external boot prom. Hence it is inherently more secure than volatile FPGAs. Like the LatticeECP2/M, the LatticeXP2 family also offers the 128-bit Advanced Encryption Standard (AES) to protect the externally stored programming file. The user has total control over the 128-bit key and no special voltages are required to maintain the key within the FPGA. Additional security enhancement for the LatticeXP2 includes:

- A security bit for the Configuration and User Flash
- One-Time-Programmable (OTP) or Permanent Lock capability
- Flash Protect

This document explains the encryption and security features and how to take advantage of them.

Lattice only distributes these capabilities through the licensed-controlled Lattice Diamond® design software. Standalone Diamond Programmer control patches can be obtained from Lattice Sales. The control patch will insert the encryption algorithm into the Diamond software tools to enable encryption. If the Diamond design software does not have encryption enabled, the following operations cannot be performed.

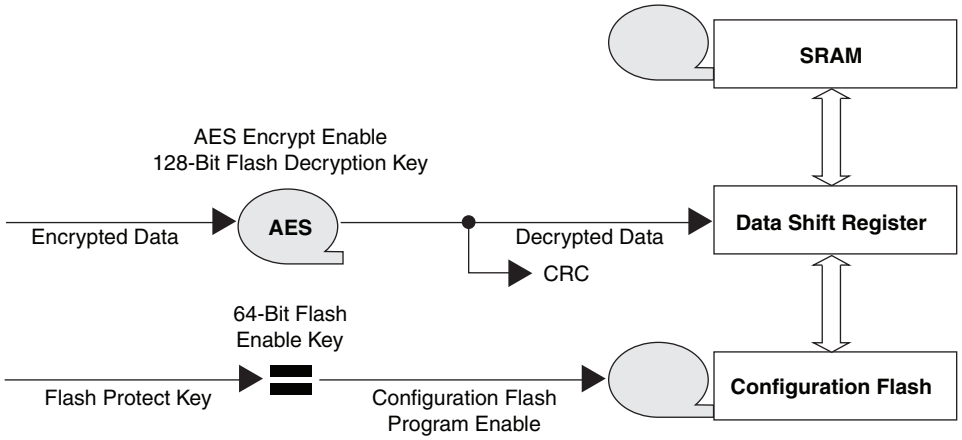
- Converting an unencrypted bitstream into an encrypted bitstream
- Writing (saving) the password-protected (encrypted) key file

If encryption is not enabled, software cannot generate the key file. Instead, the key code the user entered will be written into the chain configuration (.xcf) file.

Encryption/Decryption Flow

The LatticeXP2 supports both encrypted and non-encrypted JEDEC files. Since the non-encrypted flow is covered in TN1141, LatticeXP2 sysCONFIG™ Usage Guide, this document will concentrate on the additional steps needed for the encrypted flow. The encrypted flow adds only two steps to the normal FPGA design flow, encryption of the configuration JEDEC file and programming the encryption key into the LatticeXP2. Figure 15-1 is a block diagram describing the LatticeXP2 encryption data paths that will be used throughout this document.

Figure 15-1. Encryption Block Diagram along with Flash Protect



Encrypting the JEDEC File

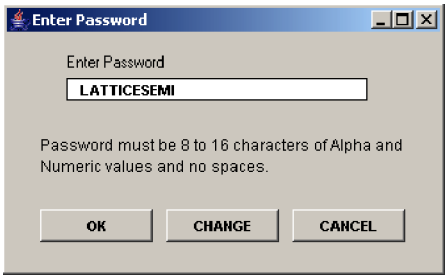
As with any other Lattice FPGA design flow, the design engineer must first create the design using the ispLEVER® design tool suite. The design is synthesized, mapped, placed and routed, and verified. Once the user is satisfied with the design, the final JEDEC file is ready for FPGA programming. This final JEDEC file is used to secure the design.

The JEDEC file can be encrypted using ispLEVER by going to the **Tools -> Security Settings** pull-down menu or by using the Universal File Writer (ispUFW), which is part of the Lattice ispVM® System tool suite.

ispLEVER Flow

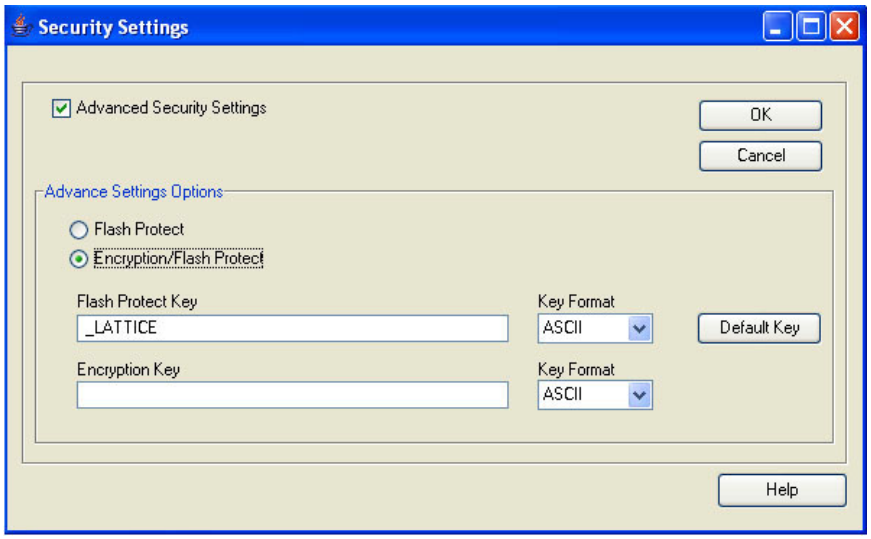
1. As mentioned above, to access the LatticeXP2 security setting GUI, go to **Project Navigator -> Tools -> Security Settings**. A password is required before entering the security features GUI section of the LatticeXP2.

Figure 15-2. Password Prompt with Default Password



2. This Password GUI prompt will automatically show the default password “LATTICESEMI”. The default password is in place for users who do not want to remember any administrative password, and especially for those who want to use the CONFIG_SECURE setting only. Users have the option of changing the password. It is the user’s responsibility to track all the keys and passwords since they will not be stored in the design files.

Figure 15-3. Security Settings



3. Once the user has selected security features, encrypted files will then be generated.

ispVM Flow

1. Start **ispUFW**. You can start ispUFW from the **Start -> Programs -> Lattice Semiconductor** menu in Windows. You will see a window that looks similar to Figure 15-4. You can also launch the ispUFW from the ispVM GUI by clicking on the UFW button on the toolbar (shown in Figure 15-5). Select **JEDEC** as the output file format, as shown in Figure 15-4.

Figure 15-4. Universal File Writer

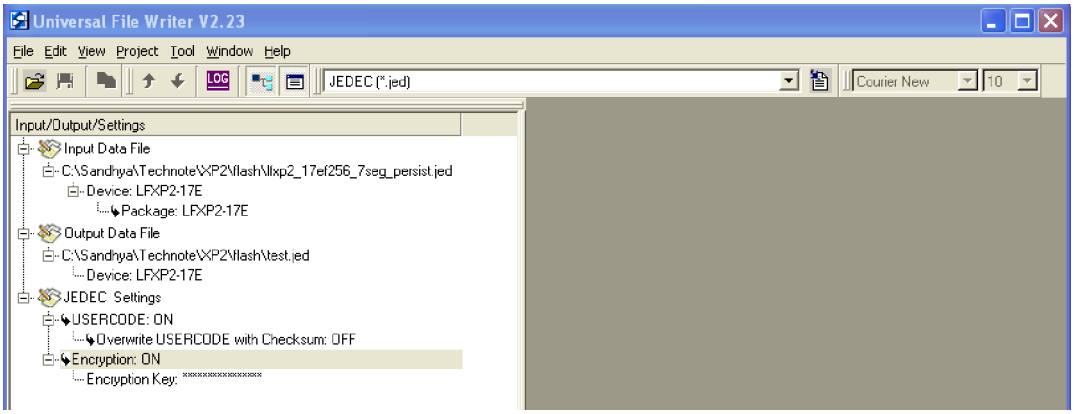
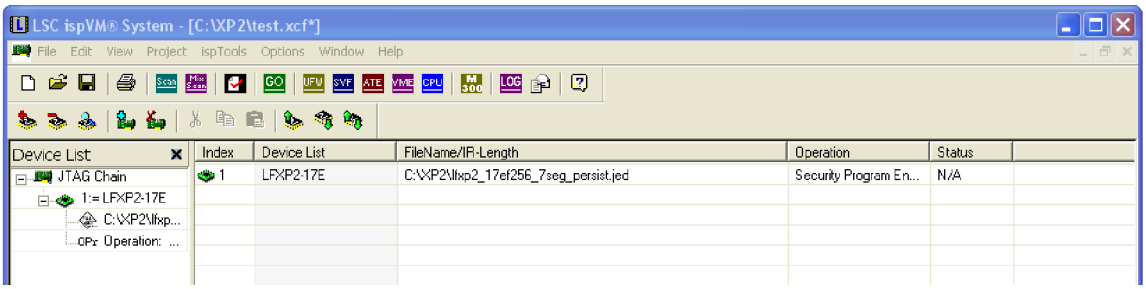
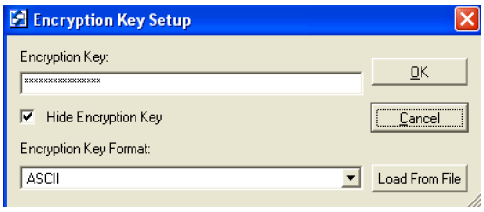


Figure 15-5. ispVM Main Window



2. Double-click on **Input Data File** and browse to the non-encrypted JEDEC file created using ispLEVER. Double-click on **Output Data File** and select an output file name. Right-click on **Encryption** and select **ON**. Right-click on **Encryption Key** and select **Edit Encryption Key**. You will see a window that looks similar to Figure 15-6.

Figure 15-6. Encryption Dialog Window



3. Enter the desired 128-bit encryption key. The key can be entered in Hexadecimal or ASCII. Hex supports 0 through F and is not case sensitive. ASCII supports all printable (ASCII codes 30 through 126) characters. You may also import the 128-bit encryption key from the <Project_Name>.bek generated by ispLEVER. To do so, click on the **Load From File** button shown in Figure 15-6. Click On **OK** to go back to the main ispUFW window.

   Note: Be sure to remember this key, as Lattice cannot recover lost keys.

4. From the ispUFW menu bar click on **Project -> Generate** or the  icon to create the encrypted JEDEC file.

5. Before the LatticeXP2 can configure with the encrypted JEDEC file, the 128-bit encryption key used to encrypt the JEDEC file must be programmed into the LatticeXP2 device.

Programming the Key into the Device

The next step is to program the 128-bit encryption key into the LatticeXP2. Note that this step is separated from JEDEC file encryption to allow flexibility in the manufacturing flow. This flow adds to design security and it allows the user to control over-building of a design. Over-building occurs when a third party builds more boards than are authorized and sells them to grey market customers. If the key is programmed at the factory, then the factory controls the number of working boards that enter the market. The LatticeXP2 will only configure from a file that has been encrypted with the same 128-bit encryption key that is programmed into the LatticeXP2.

To program the 128-bit encryption key into the LatticeXP2, proceed as follows.

1. Start ispVM. You can start ispVM from the **Start -> Programs -> Lattice Semiconductor** menu in Windows or from the ispLEVER GUI. You will see a window that looks similar to Figure 15-5.

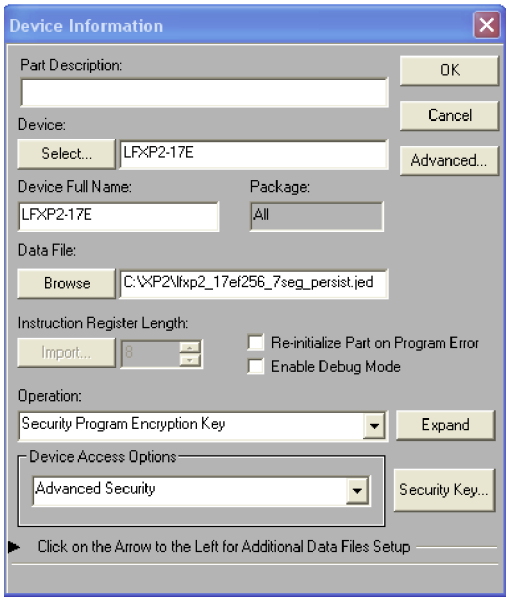
2. Attach a Lattice ispDOWNLOAD® cable from a PC to the JTAG connector wired to the LatticeXP2.

   Note: The 128-bit encryption key can only be programmed into the LatticeXP2 using the JTAG port.

3. Apply power to the board.

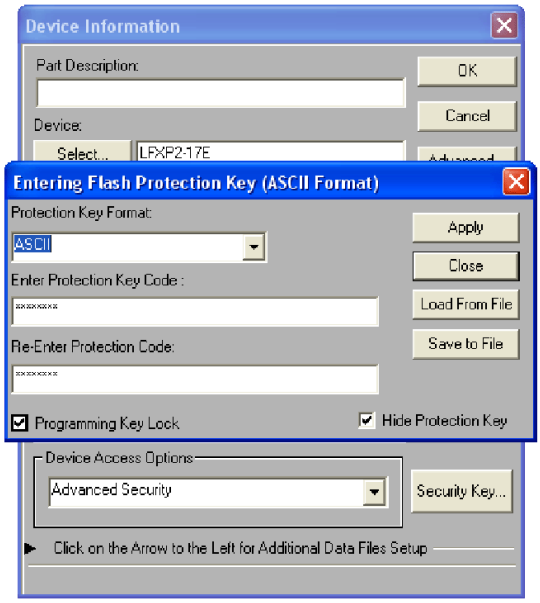
4. If the window does not show the board's JTAG chain, then proceed as follows. Otherwise, proceed to step number 5.
   a. Click the SCAN button in the toolbar to find all Lattice devices in the JTAG chain. The chain shown in Figure 15-5 has only one device, the LatticeXP2.

Figure 15-7. ispVM Device information GUI



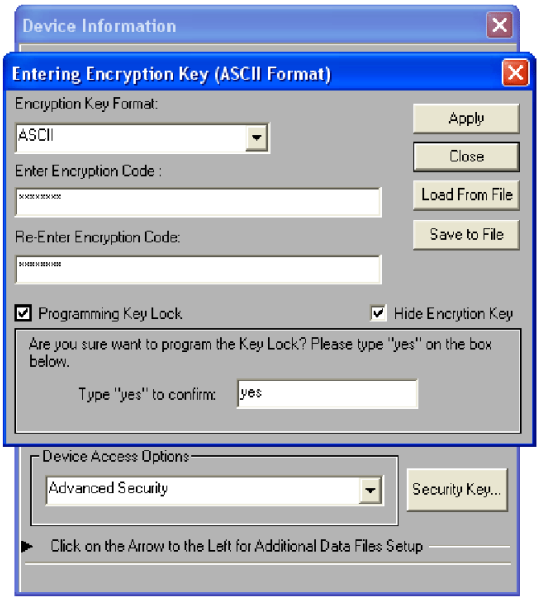
5. Double-click on the line in the chain containing the LatticeXP2. This will open the **Device Information** window (see Figure 15-7). From the **Device Access Options** drop-down box, select **Advanced Security Mode**, then click on the **Security Key** button to the right. The window will look similar to Figure 15-8.

Figure 15-8. ispVM Flash Protection Key



6. Enter the 64-bit Flash Protect key or load it from the file (<Project_Name>.key). You can save this to a file by clicking **Save to File** button. Once you click **Apply**, you will be asked to enter the 128-bit encryption as shown in Figure 15-9.

Figure 15-9. ispVM Encryption Key GUI



7. Now enter the desired 128-bit encryption key. The key can be entered in Hexadecimal or ASCII. Hex supports 0 through F and is not case sensitive. ASCII supports all printable (ASCII codes 30 through 126) characters. This key must be the same as the key used to encrypt the JEDEC file. The LatticeXP2 will only configure from an encrypted JEDEC file whose 128-bit encryption key matches the one loaded into the LatticeXP2.

   Note: Be sure to remember this key. Once the Key Lock is programmed, Lattice Semiconductor cannot read back the128-bit encryption key.

   a. The 128-bit encryption key can be saved to a file using the **Save to File** button. The 128-bit encryption key will be encrypted using an 8-character file password that the user selects. The name of the file will be **<Project_Name>.bek**. In the future, instead of entering the 128-bit key, simply click on **Load from File** and provide the file password.

8. Programming the Key Lock secures the 128-bit encryption key. When satisfied, type **Yes** to confirm, and then click **Apply**. Once the Key Lock is programmed and the device is power cycled, the 128-bit encryption key cannot be read out of the device.

9. From the main ispVM window (Figure 15-5) click on the green **GO** button on the toolbar to program the 128-bit encryption key into the LatticeXP2. When complete, the LatticeXP2 will only configure from a JEDEC file encrypted with a key that exactly matches the one just programmed.

Security Bit for the Configuration and User Flash (CONFIG_SECURE)

The CONFIG_SECURE setting is located in the GUI setting (Figure 15-3) mentioned in the previous section. After security for the device is selected, NO readback operation is supported through the sysCONFIG port or ispJTAG™ port of the general contents. This is considered the lowest level of security.

Advanced Security Settings

Selecting Advanced Security Settings will enable more security features. One-Time Programmable (OTP), Flash Protect and Encryption as shown in Figure 15-3. These settings are mutually exclusive. Selecting one or the other may nullify other fields that are not required for each particular security settings.

One-Time Programmable (OTP) or Permanent Lock

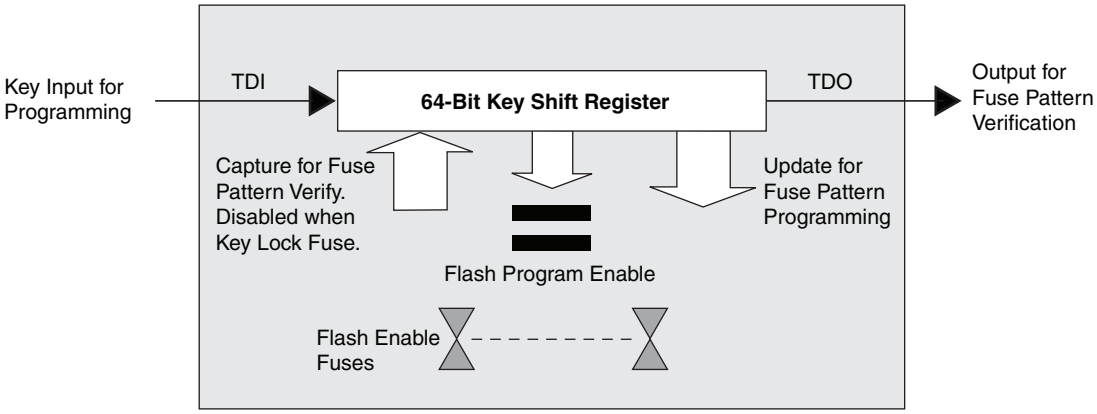
One-Time Programmable (OTP) or permanent lock is another feature that provides the highest level of security. This form of security is currently available only for LatticeXP2 devices. If OTP fuses are programmed it permanently prevents write access to the devices contents. Users must be aware before using this feature that once the OTP is programmed it is not possible to erase or reprogram the device or its security settings. **As the name implies, it is a one-time event only.** Table 15-1 specifies the behavior of the chip when security bit and the OTP bit are programmed.

Table 15-1. Security and OTP Bit Settings

Security CONFIG_SECURE	OTP Bit	Action	Re-Program	Read Back	Erase
0	0	Do nothing	Yes	Yes	Yes
0	1	Inhibit Erase or Programming	No	Yes	No
1	0	Inhibit Readback	Yes	No	Yes
1	1	Inhibit Erase, Programming or Readback	No	No	No

Flash Protect

Figure 15-10. Flash Protect



The next highest level of security for the LatticeXP2 is the 64-bit Flash Protect feature. The 64-bit Flash protect key is used to protect the embedded configuration flash from accidental or unauthorized erasure or reprogramming. This feature does not prevent the device from read back. Therefore, user is given the option to turn ON the CONFIG_SECURE feature.

The default 64-bit Flash protect key is "_LATTICE". Users can also enter their own 64-bit Flash protect key. If there is an existing 64-bit Flash protect key in the Flash Protect key file, the 64-bit Flash protect key can be imported for the Flash Protect key file (<Project_Name>.key). The ispVM GUI will display the Flash Protect key in the same format selected when the 64-bit Flash Protect key was created. Users have the option to change the 64-bit Flash protect key using the default by clicking on the **Default Key** button.

The ispVM software will automatically check the LatticeXP2 device to see if the Flash Protect feature is enabled. If it is, ispVM software will prompt the user to enter the 64-bit Flash Protect key before performing an erase or programming operation. If the 64-bit Flash Protect programmed in the device matches the 64-bit Flash Protect key entered in the ispVM GUI, the device can be erased and reprogrammed. If the keys are lost, the programmed device will be an OTP device. The re-programming of the device requires the user to enter the 64-bit Flash Protect key programmed into the device first. If it matches the 64-bit key (stored in the device), the device will enter the programming mode for erasure and re-programming of the Flash as well as the SRAM fuses.

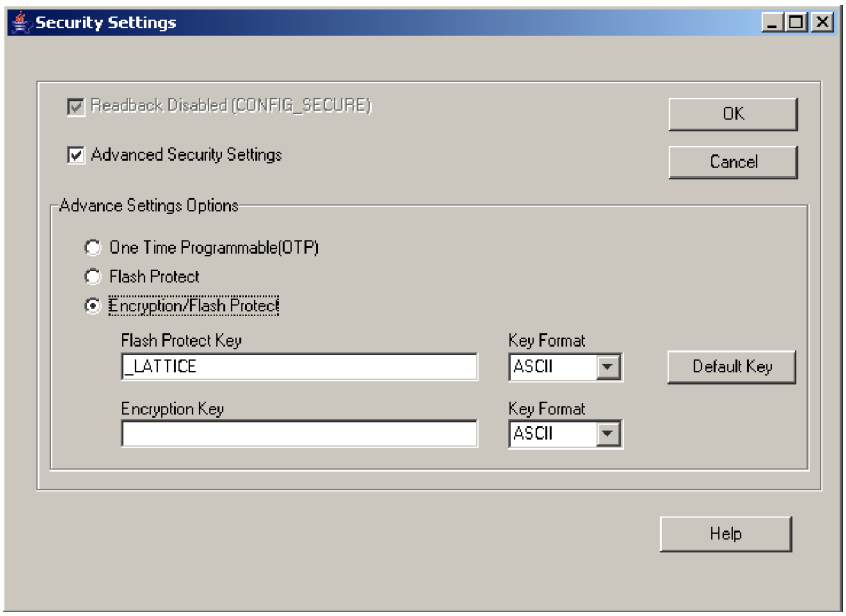
Note: The Flash Protect key cannot be the same as the Encryption Key. The Flash Protect key is 64 bits and the encryption key is 128 bits.

Changing Flash Protect

If the user decides to change the key, it can be done in the key field. The newly created Flash Protect key file (<Project_Name>.key) contains the new 64-bit Flash Protect key and is now ready to be programmed into a device.

The user can also revert back to using the default password by clicking on the **Default** button next to the Flash Protect Key. The default password is **"_LATTICE"**.

Figure 15-11. Encryption/Flash Protect Advanced Feature



Encryption

The LatticeXP2 family of devices uses the 128-bit Advanced Encryption Standard (AES) security mechanism and has a built-in AES decryption engine hardwired in the core and embedded in the device. The JEDEC file must be encrypted with the same 128-bit AES encryption key programmed into the device in order to configure it. The file is shifted into the device's JTAG port using ispVM System software. The device decrypts the JEDEC file using the 128-bit encryption key programmed into the device. The device can only be programmed if the 128-bit encryption key programmed into the device matches the 128-bit encryption key used to encrypt the JEDEC file.

Usercode in Encrypted Files

Note: The usercode is stored as a comment and will be programmed into the device's usercode.

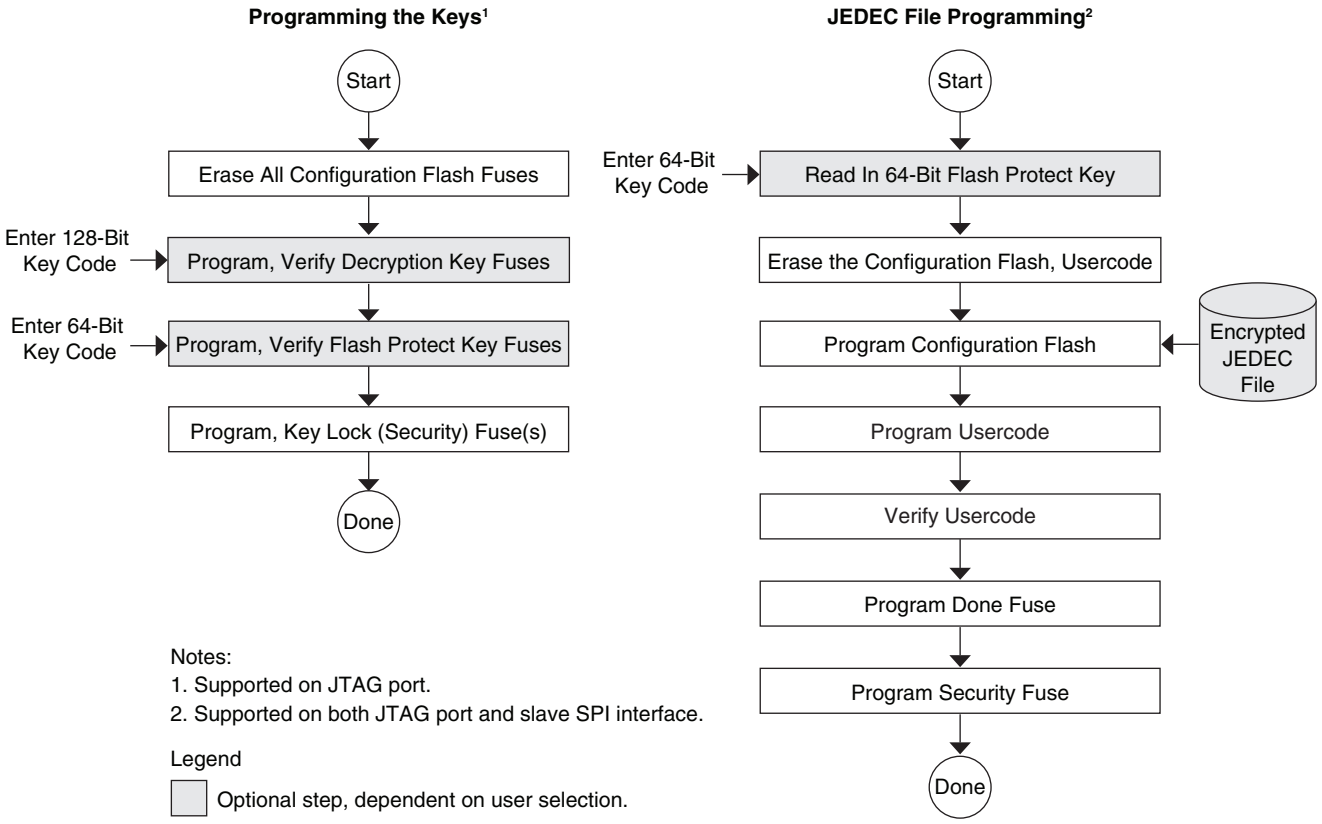
If the USERCODE is used as a custom device ID (MY_ASSP), the U field will still be used in the JEDEC file for the CRC and the value of the Custom Device ID set by the user will be part of the comment field.

The USERCODE will be available for readback regardless of the encryption setting (the USERCODE is always available for readback). Readback of the decrypted JEDEC file from a device through the JTAG port is not permitted because the security fuse is programmed when Encryption/Flash Protect is selected. Encryption will not affect the functionality of the SED

Decryption Flow

The decryption flow is a much simpler process. Start by programming the device with the 128-bit Encryption Key and the 64-bit Flash Protect Key fuses. Once the keys are programmed, the device can then be programmed with the encrypted JEDEC file.

Figure 15-12. Decryption Flow



Verifying a Configuration

If the Flash is programmed directly, the data is first decrypted and then the FPGA performs a CRC on the data. If all CRCs pass, configuration was successful. If a CRC does not pass, the Done fuse is not programmed.

References

- TN1141, LatticeXP2 sysCONFIG Usage Guide
- Federal Information Processing Standard Publication 197, Nov. 26, 2001. Advanced Encryption Standard (AES)

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2007	01.0	Initial release.
May 2008	01.1	Updated ispLEVER Security Settings screen shot.
April 2013	01.2	Updated document with new corporate logo.
		Updated Introduction content.