Lattice Sentry Root of Trust Demo Setup for MachXO3D

User Guide

FPGA-UG-02114-1.0

August 2020

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
Demo	Demonstration
JTAG	Joint Test Action Group
LED	Light-Emitting Diode
OOB	Out-of-Band
QSPI	Quad Serial Peripheral Interface
PFR	Platform Firmware Resiliency
RoT	Root of Trust
SPI	Serial Peripheral Interface
UFM	User Flash Memories
USB	Universal Serial Bus

1. Introduction

This document details the steps required to configure the Lattice Sentry Demo Board for MachXO3D™ to demonstrate the Lattice Sentry PFR solution. This document should be used in conjunction with the Lattice Sentry Root-of-Trust Reference Design for MachXO3D (FPGA-RD-02203) to run the demonstration.

1.1. Root of Trust

The MachXO3D device family is a new generation of Lattice Semiconductor Low Density PLDs with enhanced security features and on-chip dual boot flash. The enhanced bitstream security and user mode security functions enable the MachXO3D device to be used as a RoT hardware solution in a complex system. Figure 1.1 shows the MachXO3D device architecture.

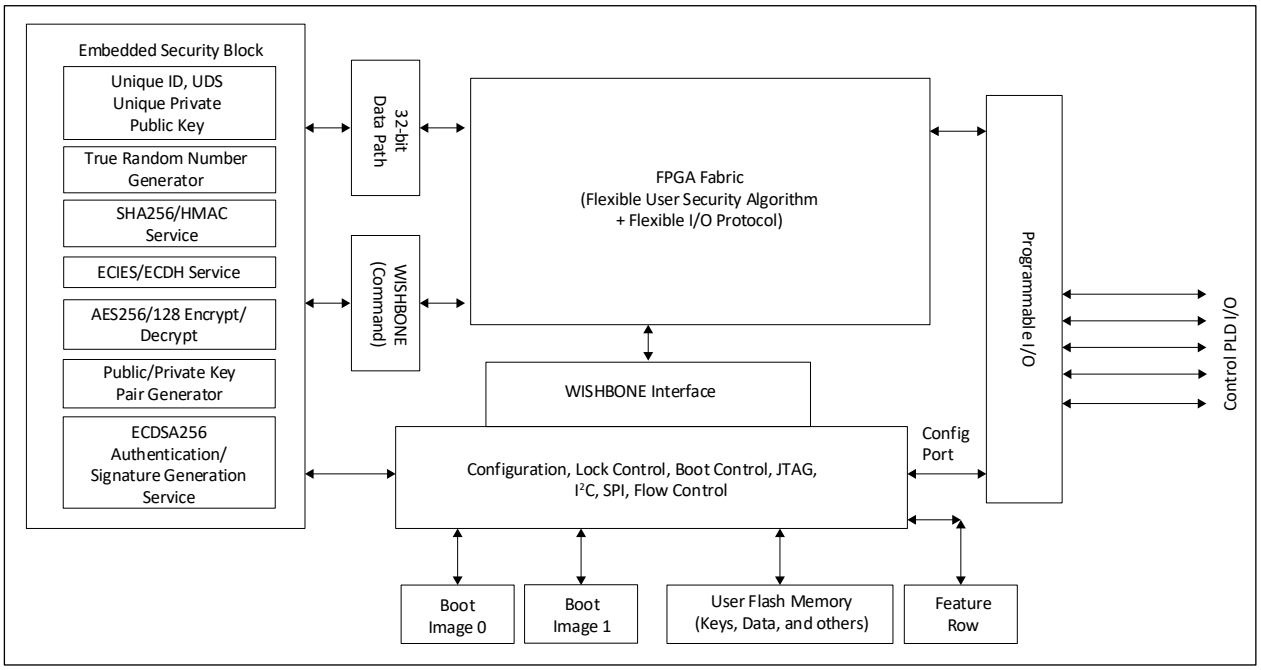

Figure 1.1. MachXO3D Device Architecture

Self-Detection

- On-chip configuration image is authenticated by immutable security engine before boot.
- Security engine uses public key stored on chip, plus the encryption support.

Self-Recovery

- Automatically switches over to the other authenticated image if authentication fails.

Self-Protection

- Prevents configuring from compromised image.
- Fabric logic controls access from programming ports.
- Lock policy: Separate access rights for each Flash store.
- Fabric communicates with external controller through out-of-band (OOB) communication interface.
- Firmware attacks (Erasing/Corrupting both images) are blocked.
- Blocks attacks from all configuration ports during in-transit.

1.2. Lattice Semiconductor PFR

Figure 1.2 shows the Lattice MachXO3D Platform Firmware Resiliency (PFR) concept. The RoT device is Lattice MachXO3D. It ensures the processor boot authentication, and detects and protects SPI flash operation. The SPI/QSPI switch bridges the processor, SPI memory, and RoT device. The switch provides the mechanism for the RoT device to terminate any unauthorized operation to SPI memory.

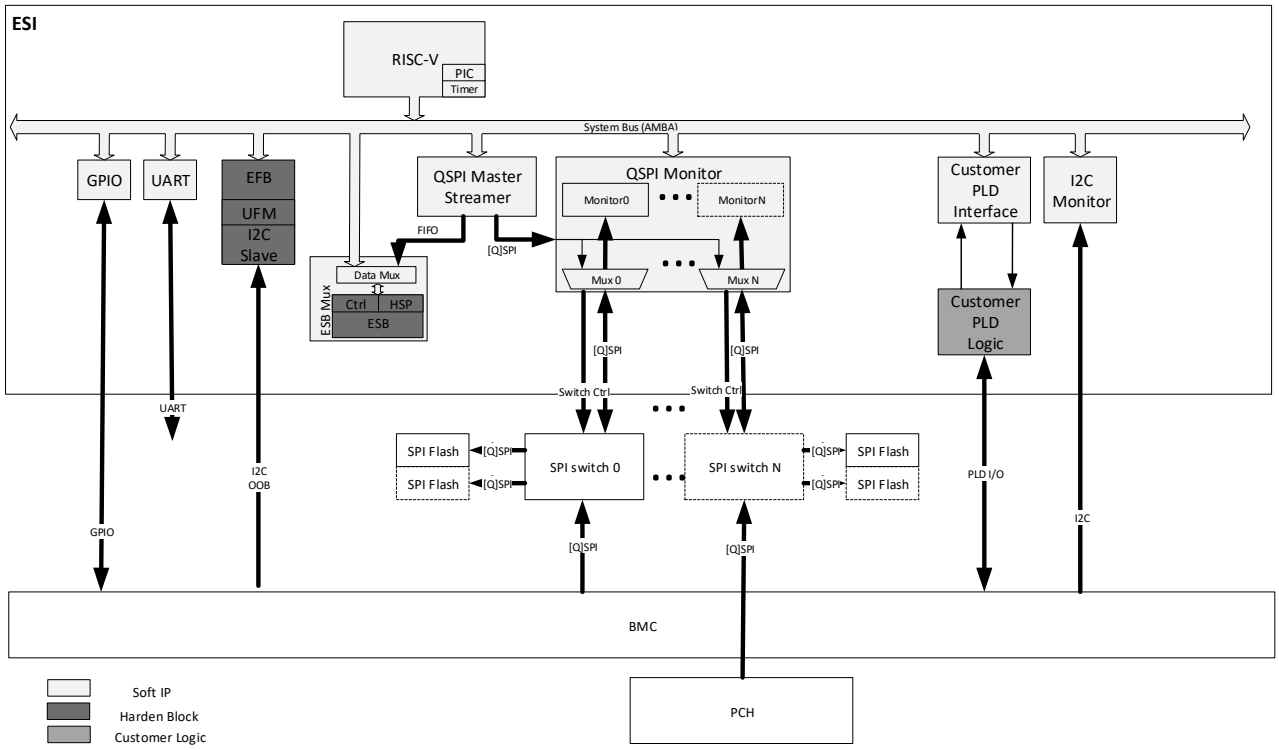

Figure 1.2. Lattice PFR System Architecture

The Lattice RoT with PFR demonstration design includes the following features:

Detection

Signature authentication for boot image – At power up, the RoT device holds the processor/controller/FPGA in reset and authenticates the image with the signature in SPI memory. Once the firmware passes authentication, the RoT device releases the reset and the processor/controller/FPGA boots from SPI memory. The feature can also be triggered with a command.

Recovery

Automatic recovery for boot image – If the firmware fails to authenticate, the RoT device copies over a valid image/signature from the backup location. The feature can also be triggered with a command.

Protection

Critical data access monitor and protection – Using defined white and black address spaces, the RoT device monitors accesses to the SPI memory for any illegal operations. Once an illegal operation is detected, the monitor terminates the access to SPI memory and reports/logs the illegal operation.

2. Lattice Sentry Root-of-Trust Demo Package

2.1. Delivery Package

Table 2.1. Delivery Package

Path	File	Description
./bitstreams	machXO3D_Sentry_PFR.jed	MachXO3D configuration file
—	manifest.jed	Manifest - UFM2 configuration file
—	manifest.mem	Manifest memory initialization file
—	ecp5_bmc.bit	BMC ECP5™ configuration bitstream
—	ecp5_bmc.bit.sig	BMC ECP5 signature file
—	ecp5_bmc.bit.sig.hex	BMC ECP5 bitstream signature
—	ecp5_pch.bit	PCH ECP5 configuration bitstream
—	ecp5_pch.bit.sig	PCH ECP5 signature file
—	ecp5_pch.bit.sig.hex	PCH ECP5 bitstream signature for programming
—	keys.txt	Private/Public key for signature generation

3. Lattice Sentry Demo Board for MachXO3D Setup

3.1. Jumper and Switch Settings

Table 3.1. Jumper and Switch Settings

Jumper	Position	Invoked Demo Resource
JP2	Installed	12 MHz clock from U1 to PCH
JP9	Installed	TCK from U1 to BMC
JP10	Installed	TCK from U1 to MachXO3D
JP19	Installed	TCK from U1 to PCH
JP20	Installed	TMS from U1 to BMC
JP21	Installed	TMS from U1 to PCH
JP22	Installed	TMS from U1 to XO3D
JP30	Installed	12 MHz clock from U2 to BMC
JP34	Installed	3.3 V supplied to BMC Bank 7, PCH Bank 1, PCH Bank 7, and MachXO3D Bank 5.
J58	1-2 Installed, 3-4 Installed	JTAG chain to invoke MachXO3D, PCH, and BMC in sequence.
J59	2-3 Installed	
J60	2-3 Installed	
SW10	1-ON, 2-ON, 3-OFF, 4-ON	Master SPI Configuration Mode
SW12	1-ON, 2-ON, 3-OFF, 4-ON	Master SPI Configuration Mode
All others	Removed	—

3.2. Setting up the Board

To set up the board:

1. Ensure that the MachXO3D PFR Demo Board jumpers are properly set. Refer to Table 3.1.
2. Connect the Mini USB cable from J6 of the MachXO3D PFR Demo Board to the host PC.
3. Connect the 12 V power supply to J11 of the MachXO3D PFR Demo Board.

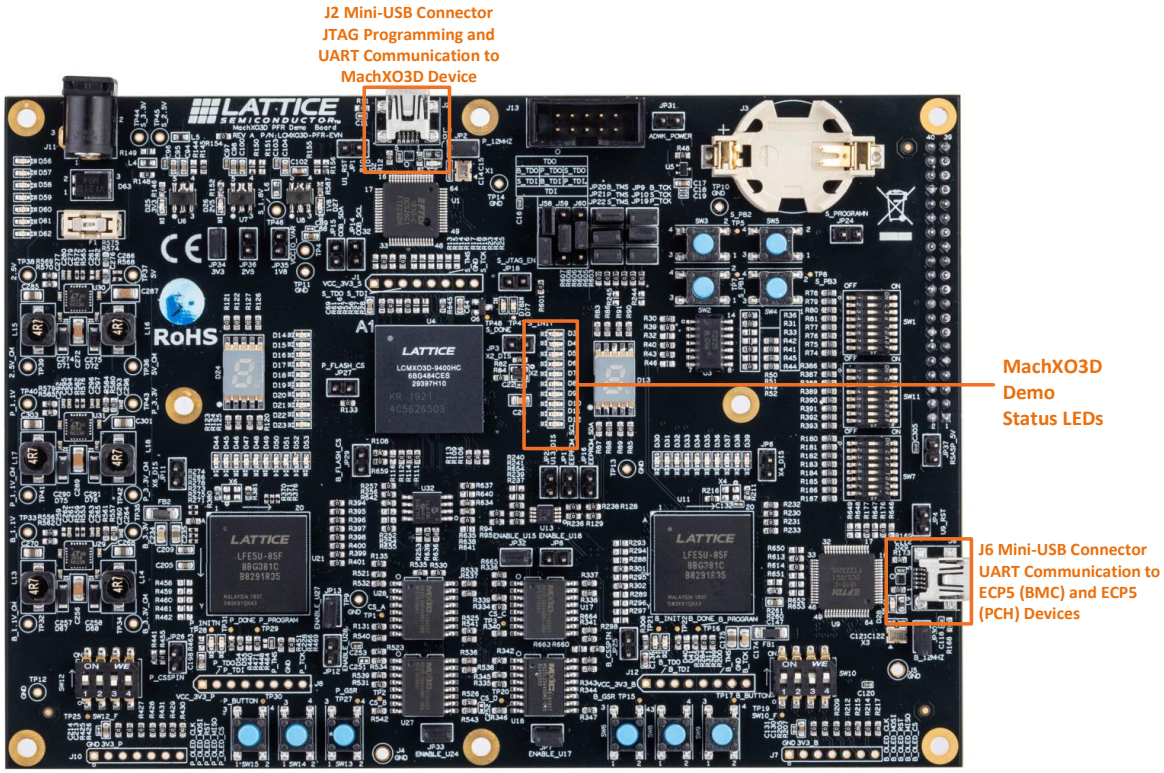

Figure 3.1. Board Setup

4. Once power is applied, the PFR Demo boots up. The MachXO3D Demo Status LEDs on the MachXO3D PFR Demo board provide the indications, as listed in Table 3.2.

Table 3.2. MachXO3D LED Indications

LED	Indication
D3	PCH ECP5 PROGn
D4	PCH Quickswitch Enabled
D5	PCH Flash A Enabled
D6	PCH Flash B Enabled
D7	BMC ECP5 PROGn
D8	BMC Quickswitch Enabled
D9	BMC Flash A Enabled
D10	BMC Flash B Enabled
D11	—
D12	SPI Exception Detected

5. Once LED D3 and D7 on the MachXO3D Development Board go off, the PCH and BMC ECP5 are released for configuration. LED D53 and D39 should start blinking after the PCH and BMC ECP5 are configured respectively.

4. Programming Demo Images

To program the devices:

1. Connect the 12 V power supply to the MachXO3D PFR Demo Board.
2. Install jumpers JP7, JP13, JP32, and JP33.

   Note: Jumpers JP7, JP8, JP12, JP13, JP32, and JP33 need to be removed before running the demo, that is, Power cycling the board or Reset).
3. Connect the Mini USB cable from the PC to connector J2.

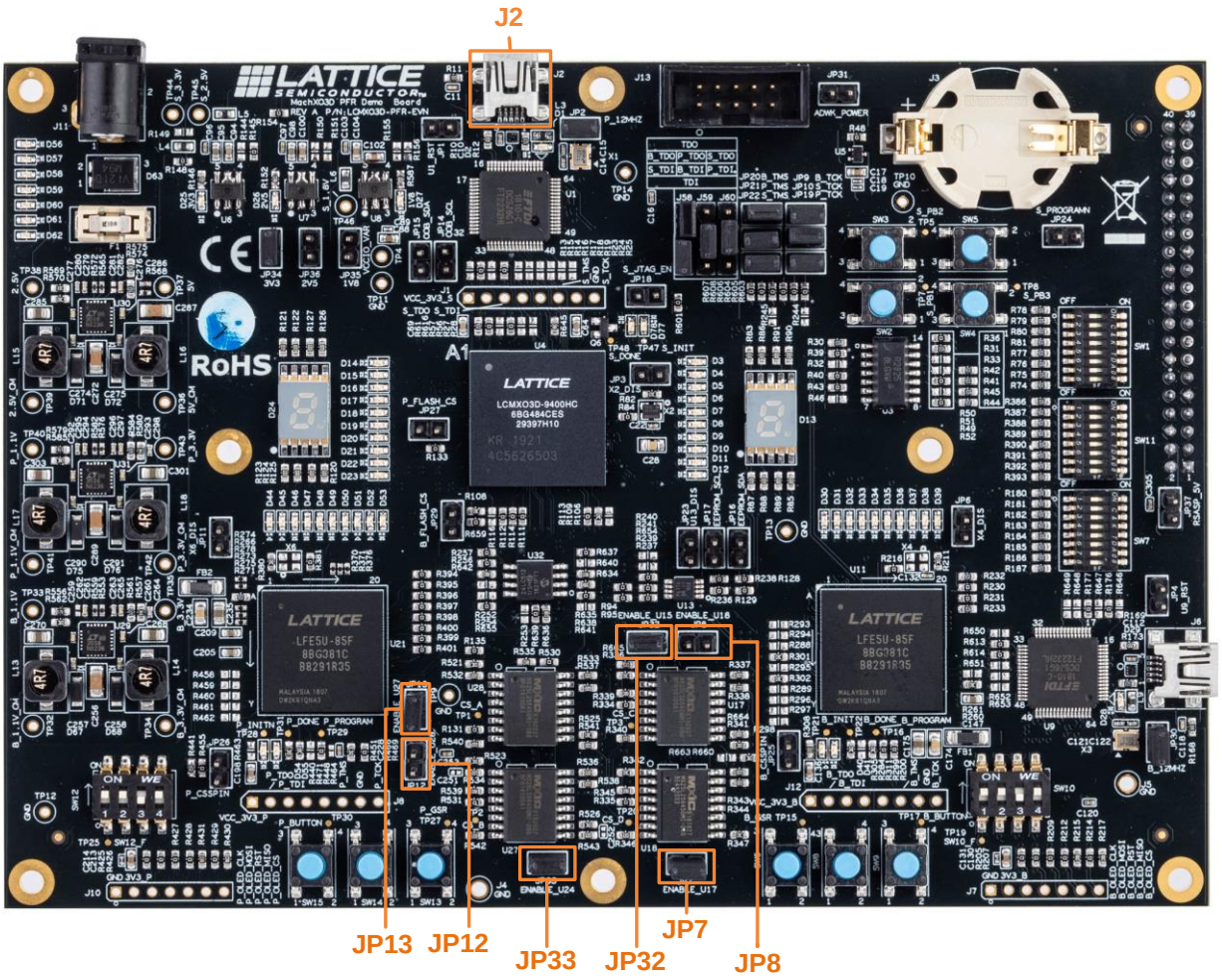

Figure 4.1. USB and Jumper Connections

4. Open Lattice Diamond Programmer (version 3.11 or later) and create a new project from JTAG scan. If the device column is highlighted in yellow, manually select the device as shown in Figure 4.2.

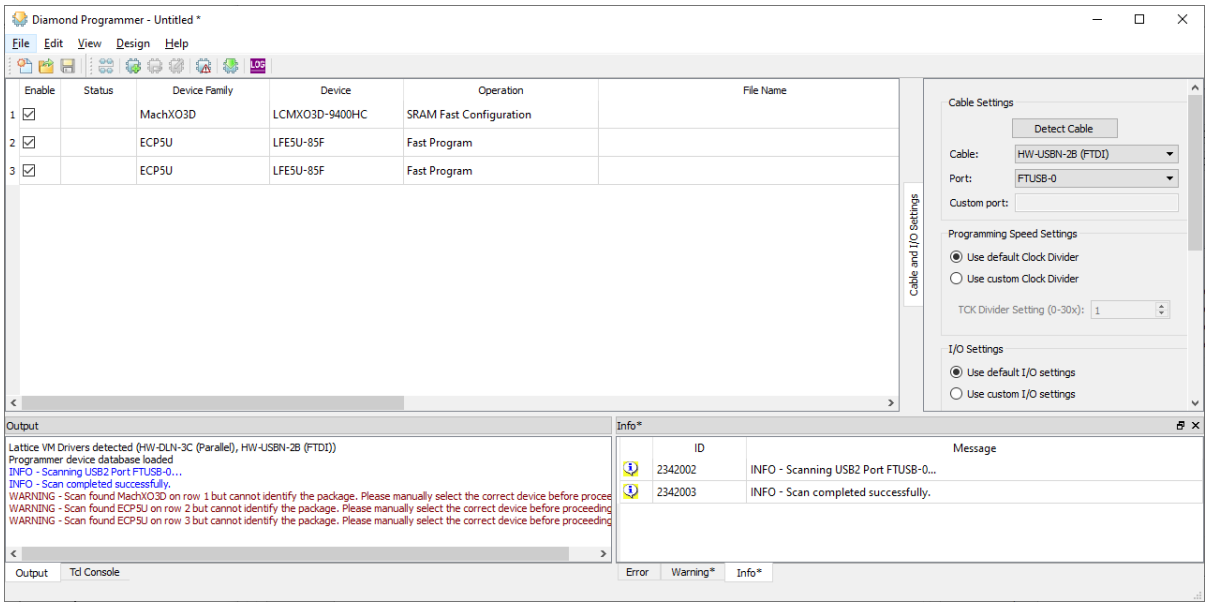

Figure 4.2. Device Selection

5. For each device, select **SRAM Erase Only** or **Erase Only** under **Operation** and click the **Program** button as shown in Figure 4.3.

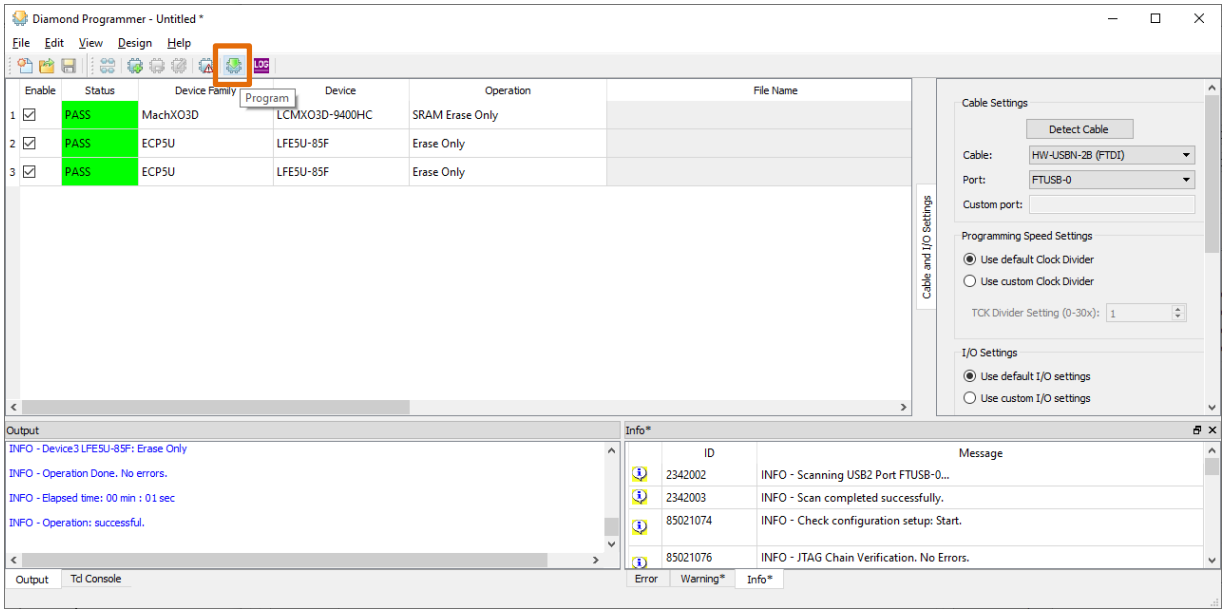

Figure 4.3. Device Operation

4.1. Programming SPI Flash for PCH ECP5

To program SPI Flash for PCH ECP5:

1. Uncheck **Enable** for Device 1 and Device 3.
2. Configure the operation for Device 2 as shown in Figure 4.4.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – SPI Flash Background Programming**
      - **Operation – SPI Flash Erase, Program, Verify**
   b. Under Programming Options, select ***<path>/bitstreams/ecp5_pch.bit*** in **Programming file**.
   c. Under **SPI Flash Options**, select the options below:
      - **Family – SPI Serial Flash**
      - **Vendor – Macronix**
      - **Device – MX25L25635E**
      - **Package – 16-Pin SOP**
   d. Under **SPI Programming**, select the options below:
      - **Data File Size – Load from File**
      - **Start Address – 0x00000000**
      - **End Address – 0x001F0000**

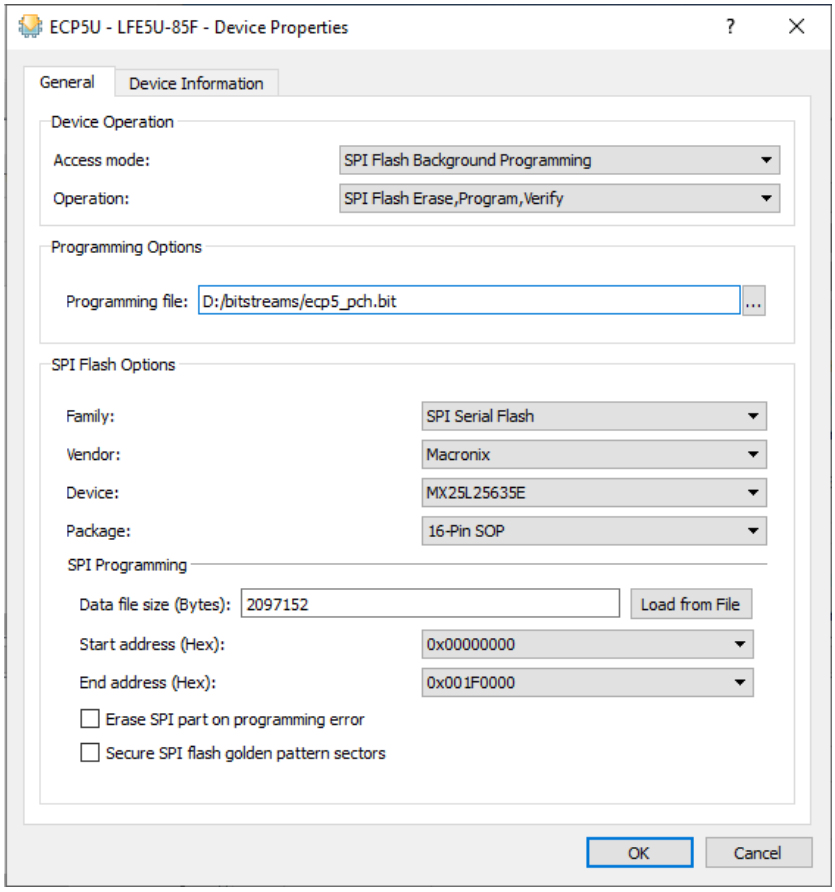

Figure 4.4. Device Operation

3. Program the Flash as shown in Figure 4.5.

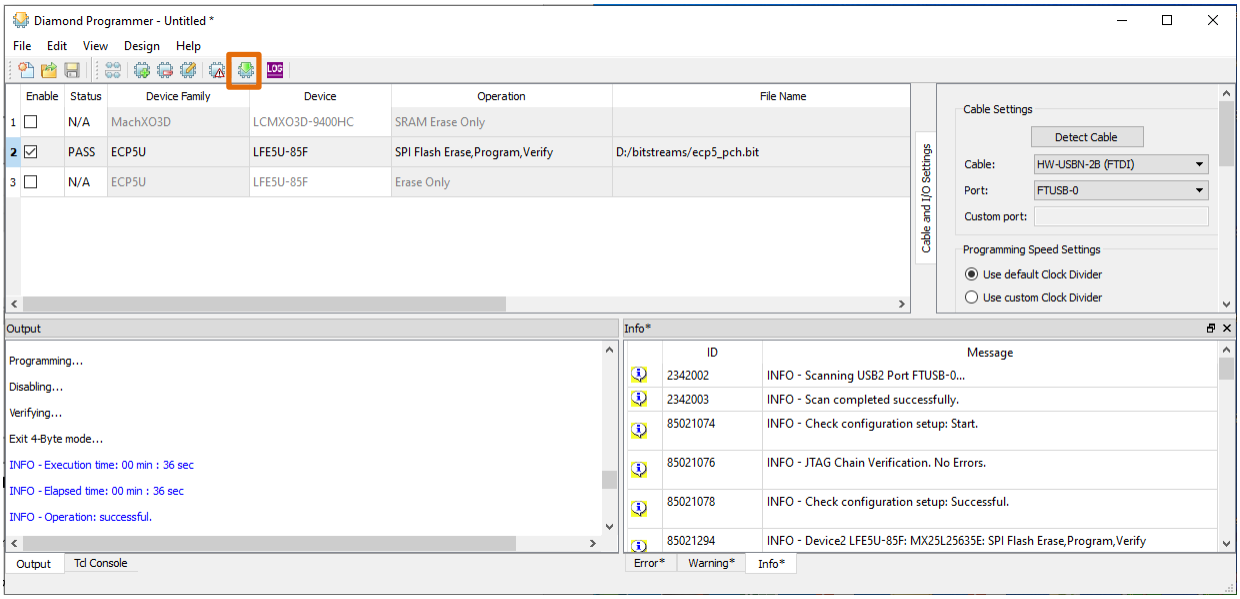

Figure 4.5. Flash Program Operation

4. Configure operation for Device 2 as shown in Figure 4.6.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – SPI Flash Background Programming**
      - **Operation – SPI Flash Erase, Program, Verify**
   b. Under **Programming Options**, select ***<path>/bitstreams/ecp5_pch.bit.sig.hex*** in **Programming file**.
   c. Under **SPI Flash Options**, select the options below:
      - **Family – SPI Serial Flash**
      - **Vendor – Macronix**
      - **Device – MX25L25635E**
      - **Package – 16-Pin SOP**
   d. Under **SPI Programming**, select the options below:
      - **Data File Size – Load from File**
      - **Start Address – 0x00200000**
      - **End Address – 0x00200000**

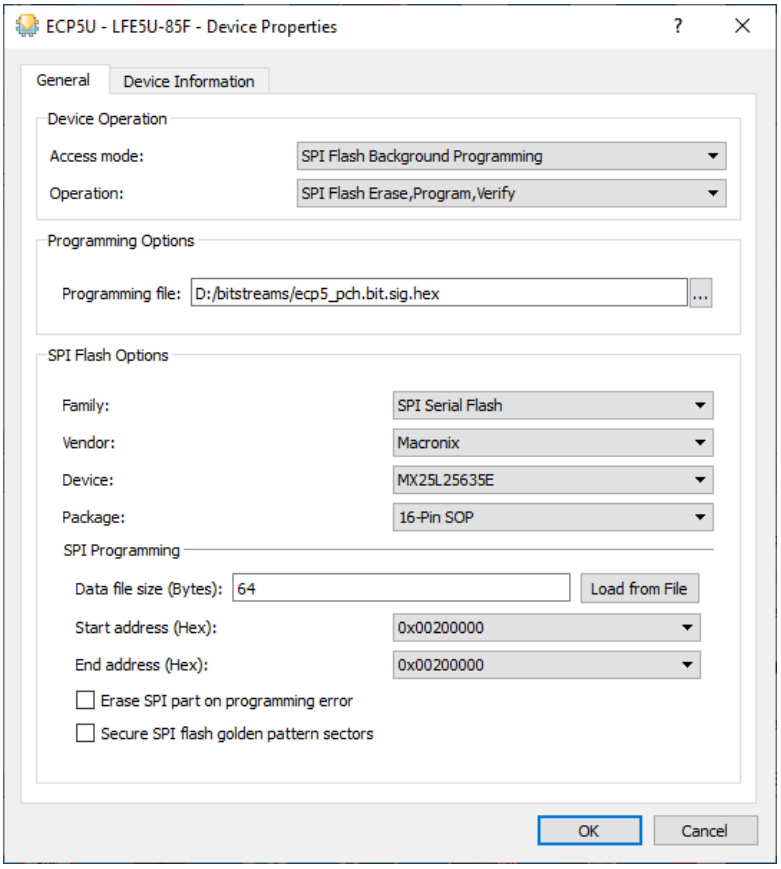

Figure 4.6. Device Operation

5. Program the Flash as shown in Figure 4.7.

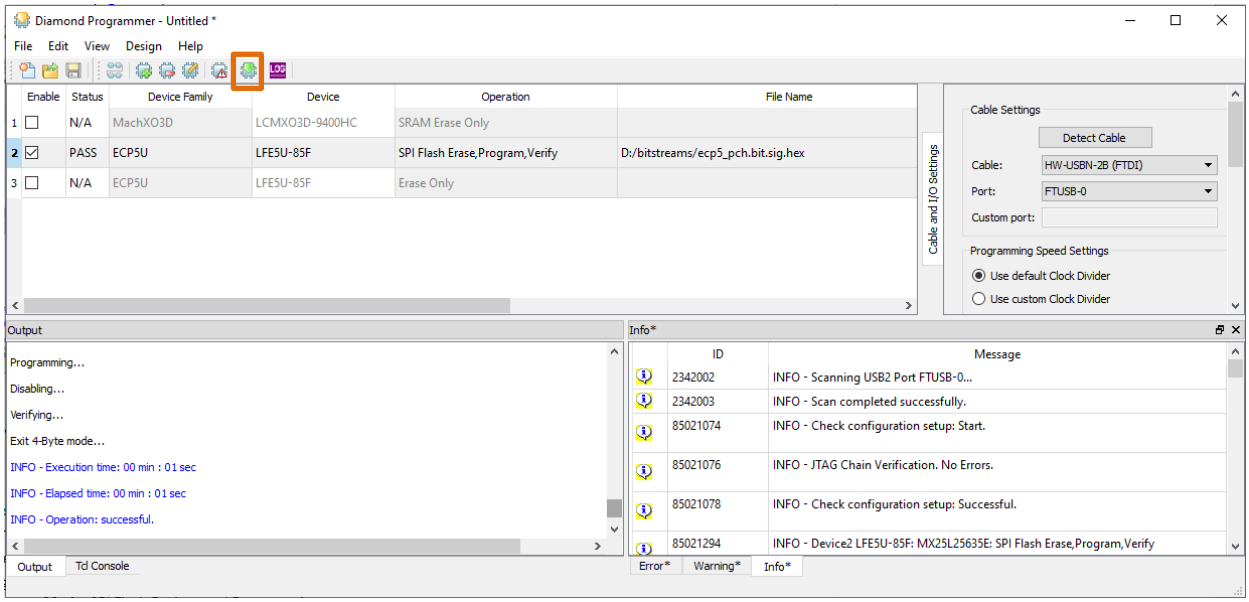

Figure 4.7. Flash Program Operation

6. Remove jumper from JP13 and install jumper on JP12. Repeat steps 2 through 5. This programs the secondary flash.

4.2. Programming SPI Flash for BMC ECP5

To program SPI Flash:

1. Deselect **Enable** for Device 2 and select **Enable** for Device 3.
2. Configure operation for Device 3 as shown in Figure 4.8.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – SPI Flash Background Programming**
      - **Operation – SPI Flash Erase, Program, Verify**
   b. Under **Programming Options**, select ***<path>/bitstreams/ecp5_bmc.bit*** in **Programming file**.
   c. Under **SPI Flash Options**, select the options below:
      - **Family – SPI Serial Flash**
      - **Vendor – Macronix**
      - **Device – MX25L25635E**
      - **Package – 16-Pin SOP**
   d. Under **SPI Programming**, select the options below:
      - **Data File Size – Load from File**
      - **Start Address – 0x00000000**
      - **End Address – 0x001F0000**

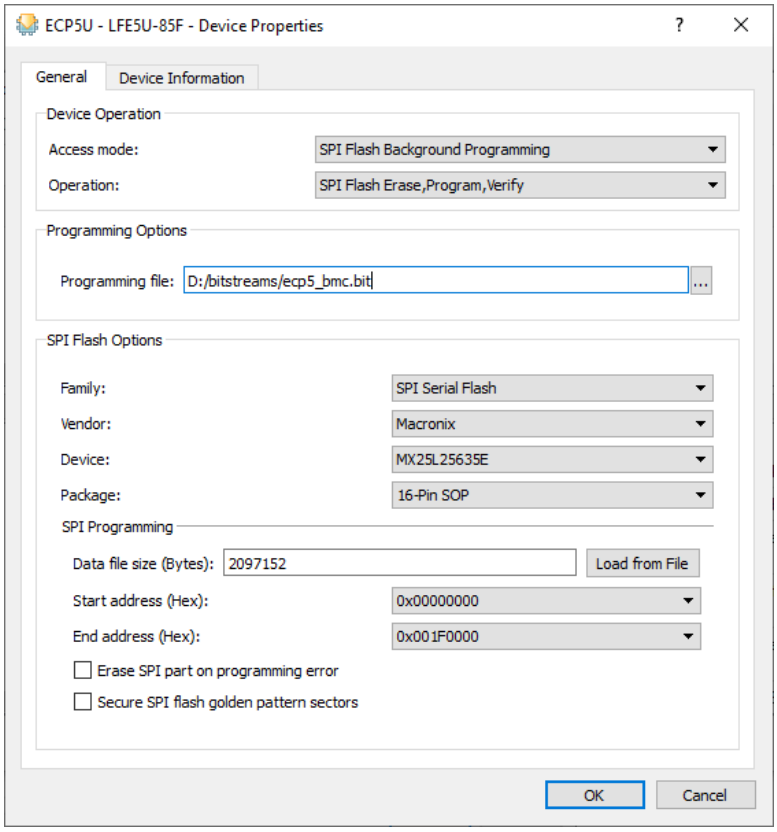

Figure 4.8. Device Operation

3. Program the Flash as shown in Figure 4.9.

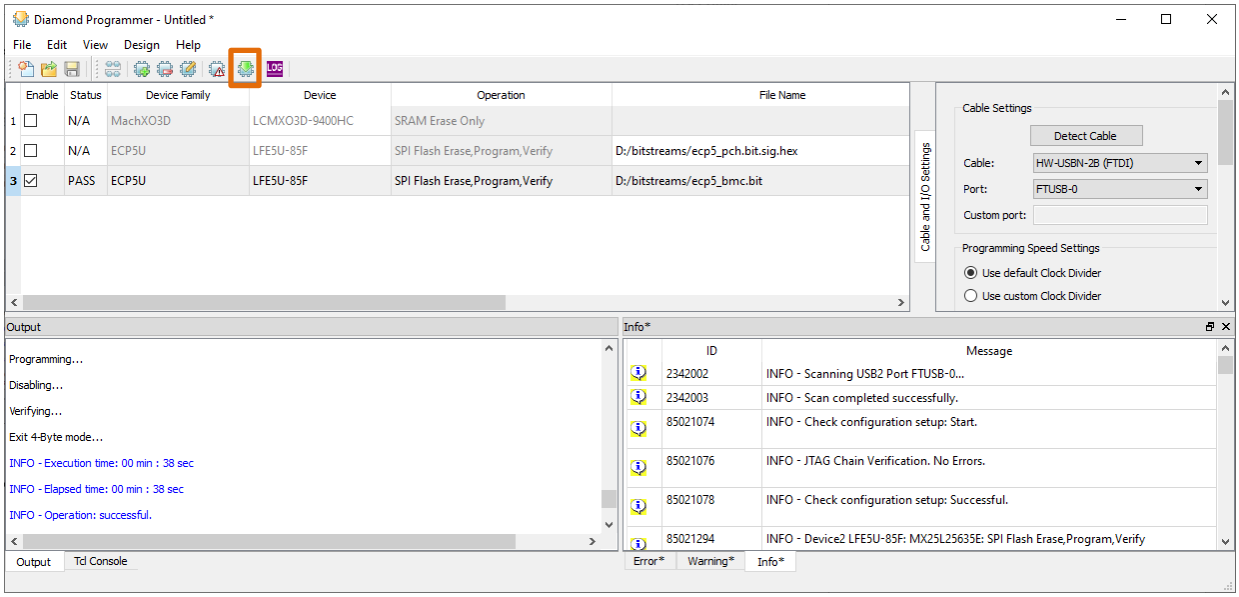

Figure 4.9. Flash Program Operation

4. Configure operation for Device 3 as shown in Figure 4.10.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – SPI Flash Background Programming**
      - **Operation – SPI Flash Erase, Program, Verify**
   b. Under **Programming Options**, select ***<path>/bitstreams/ecp5_bmc.bit.sig.hex*** in **Programming file**.
   c. Under **SPI Flash Options**, select the options below:
      - **Family – SPI Serial Flash**
      - **Vendor – Macronix**
      - **Device – MX25L25635E**
      - **Package – 16-Pin SOP**
   d. Under **SPI Programming**, select the options below:
      - **Data File Size – Load from File**
      - **Start Address – 0x00200000**
      - **End Address – 0x00200000**

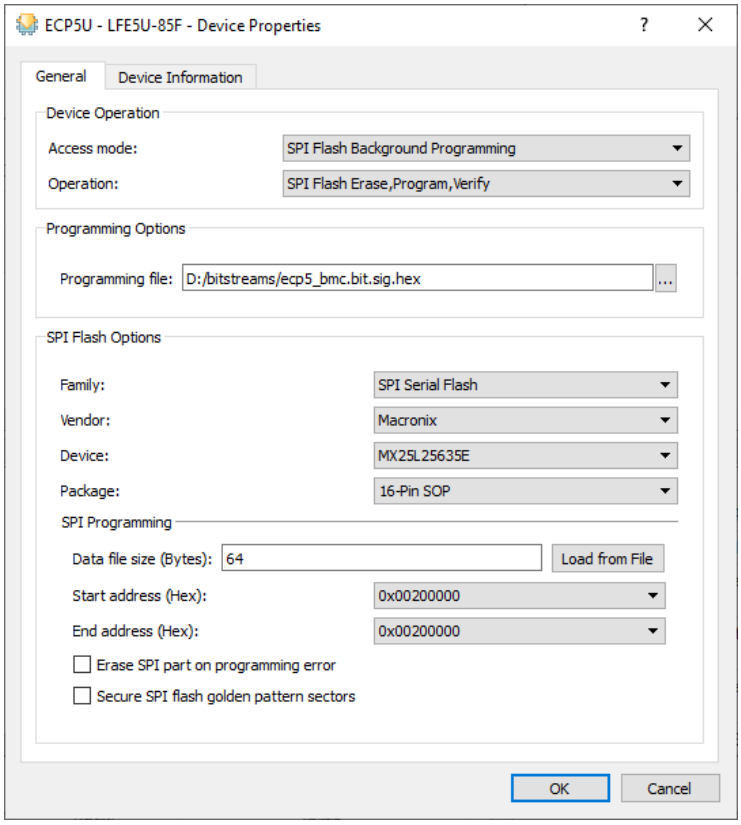

Figure 4.10. Device Operation

5. Program the Flash as shown in Figure 4.11.

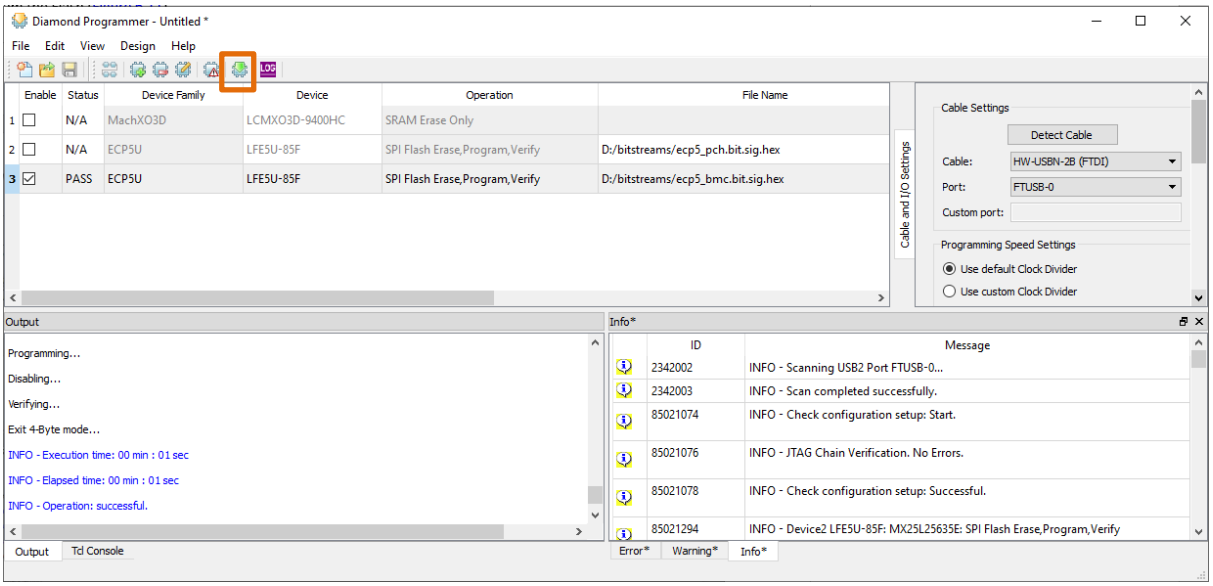

Figure 4.11. Flash Program Operation

6. Remove jumper from JP7 and install jumper on JP8. Repeat steps 2 through 5. This programs the secondary flash.

4.3. Programming MachXO3D Manifest

To program MachXO3D Manifest:

1. Remove jumpers on JP7, JP8, JP12, JP13, JP32, and JP33.
2. Select **Enable** for Device 1 and deselect **Enable** for Device 2 and Device 3.
3. Configure operation for Device 1 as shown in Figure 4.12.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – Flash Programming Mode**
      - **Port Interface – JTAG Interface**
      - **Operation – FLASH UFM Erase, Program, Verify**
   b. Select **Flash-UFM Programming Options**.
   c. Select **UFM2 Programmable file** and enter ***<path>/bitstreams/manifest.jed***.

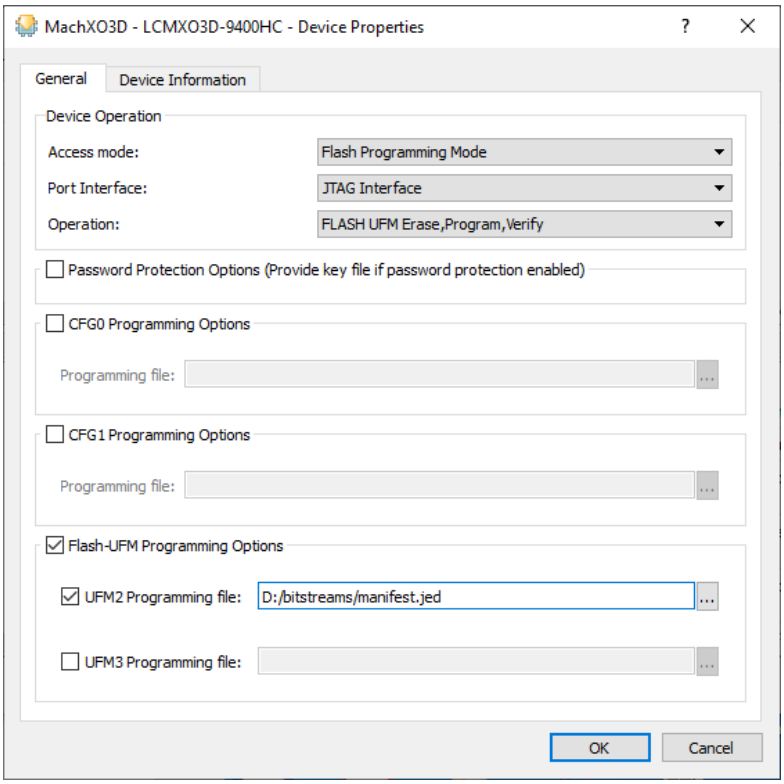

Figure 4.12. Device Operation

4. Program the MachXO3D UFM2 as shown in Figure 4.13.

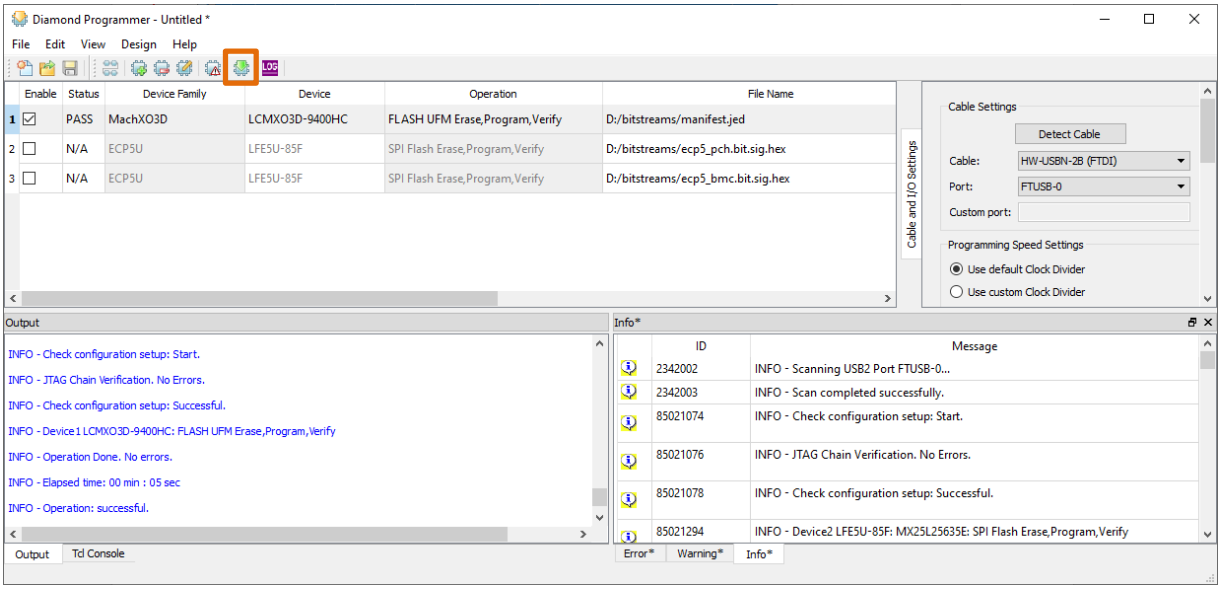

Figure 4.13. Program Operation

4.4. Programming MachXO3D Configuration

To program MachXO3D Configuration:

1. Remove jumpers on JP7, JP8, JP12, JP13, JP32, and JP33.
2. Select **Enable** for Device 1 and deselect **Enable** for Device 2 and Device 3.
3. Configure operation for Device 1 as shown in Figure 4.14.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – Flash Programming Mode**
      - **Port Interface – JTAG Interface**
      - **Operation – FLASH CFG Erase, Program, Verify**
   b. Select **CFG0 Programming Options**.
   c. In Programming file, enter ***<path>/bitstreams/machXO3D_Sentry_PFR.jed***.

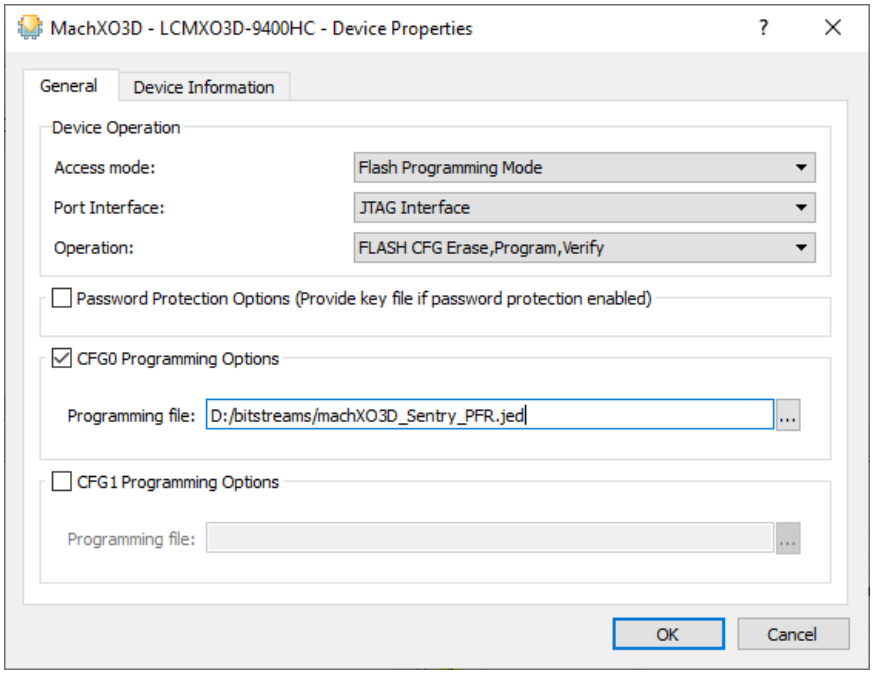

Figure 4.14. Device Operation

4. Program the MachXO3D CFG0 as shown in Figure 4.15.

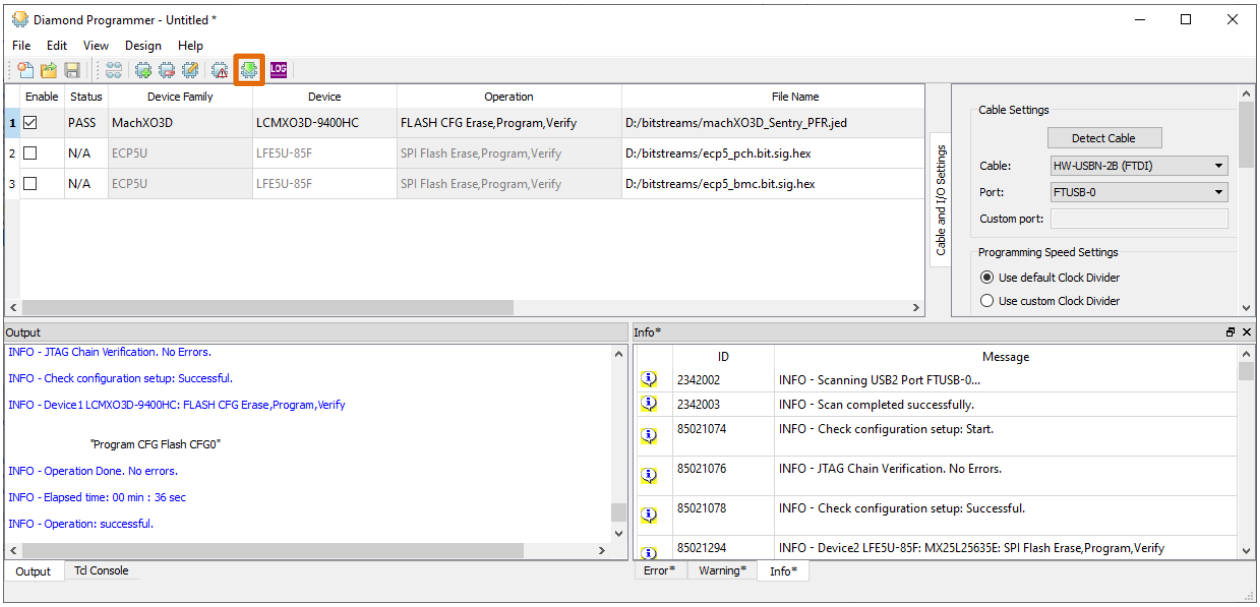

Figure 4.15. Program Operation

4.5. Erasing MachXO3D UFM3 (Log Memory)

To erase MachXO3D UFM3:

1. Remove jumpers on JP7, JP8, JP12, JP13, JP32, and JP33.
2. Select **Enable** for Device 1 and deselect **Enable** for Device 2 and Device 3.
3. Configure operation for Device 1 as shown in Figure 4.16.
   a. Under **Device Operation**, select the options below:
      - **Access Mode – Flash Programming Mode**
      - **Port Interface– JTAG Interface**
      - **Operation– FLASH UFM Erase Only**
   b. Make sure CFG0 Programming Options is not selected.
   c. Select **Flash Flash-UFM Programming Options**.
   d. Select **UFM3**.

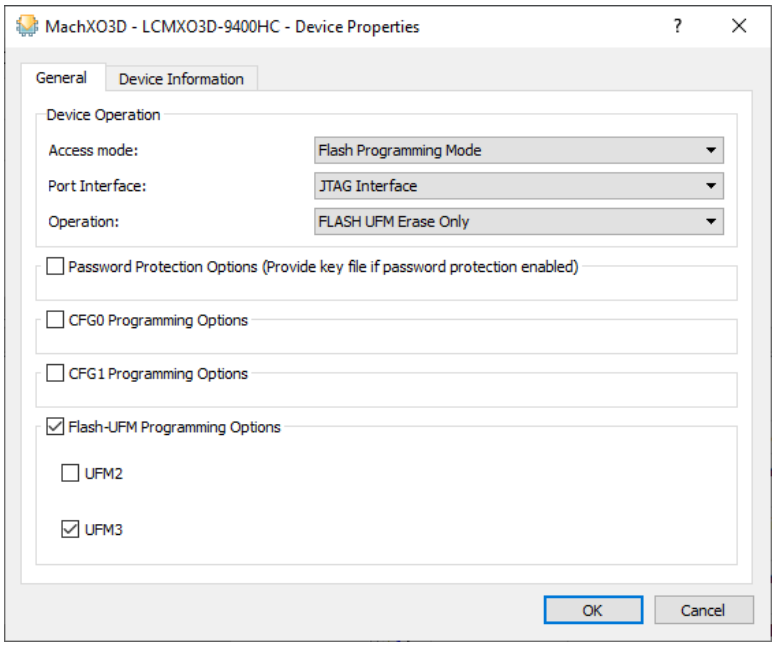

Figure 4.16. Device Operation

4. Erase MachXO3D UFM3 as shown in Figure 4.17.

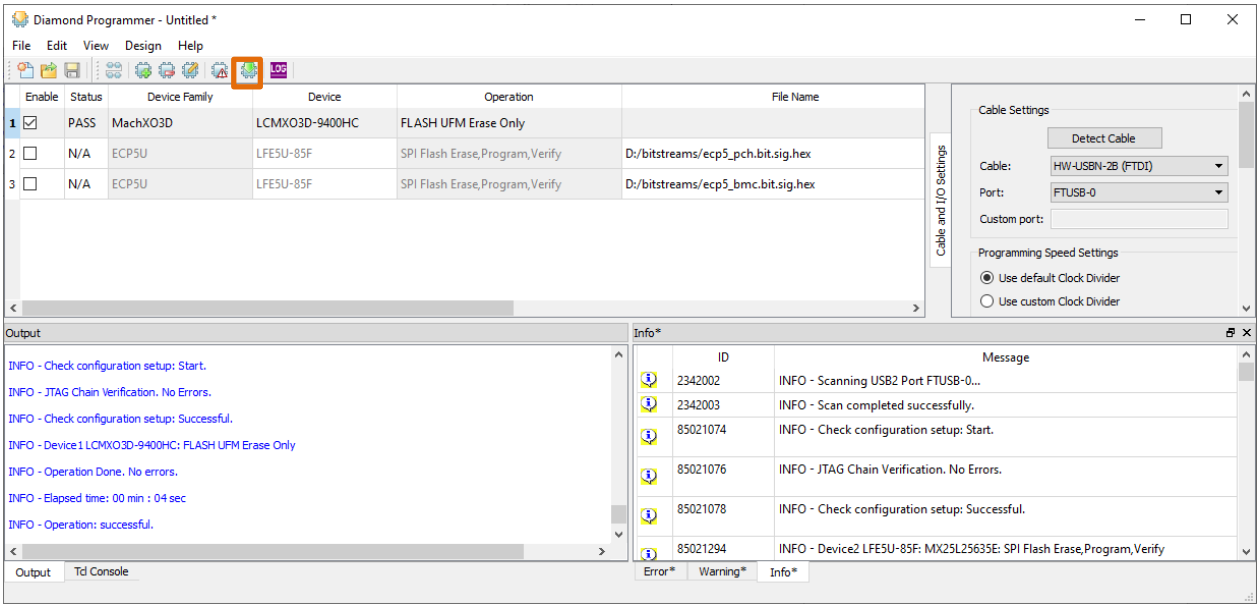

Figure 4.17. Erase Operation

5. Using the PFR Demo Tool User Interface and Running the Demo

The PFR Demo user interface is integrated into the Lattice Propel platform, refer to Lattice Sentry Root-of-Trust Reference Design for MachXO3D (FPGA-RD-02203) and follow the steps in section 7.

Table 5.1. Quick Reference Command Descriptions

Command	Supported Device	Type	Description	Option 1	Option2	Option 3	Option4	Option 5
Authenticate Image	BMC	PFR I^2C	Authenticate image with signature and public key	Image ID	1- Pri 2- Sec	—	—	—
Recover Image	BMC	PFR I^2C	Erase image/signature stored in secondary (primary) flash and copy over image and signature from primary (secondary) flash	Image ID	0- Pri -> Sec 1- Sec -> Pri	—	—	—
Recover UBoot	BMC	PFR I^2C	Erase image/signature stored in secondary (primary) flash and copy over image and signature from CFG1. Image ID must be identified as UBoot in Manifest	Image ID	0- Pri -> Sec 1- Sec -> Pri	—	—	—
Clear Log	BMC	PFR I^2C	Erase logs in UFM3	—	—	—	—	—
Update Image Info	BMC	PFR I^2C	Update Manifest with image start address and size in manifest for primary or secondary image	Image ID	1- Pri 2- Sec	Start Address	Length	
Update Signature Info	BMC	PFR I^2C	Update Manifest with signature start address in manifest for primary or secondary image	Image ID	1- Pri 2- Sec	Start Address	—	—
Update Version Info	BMC	PFR I^2C	Update Manifest with start address of version info	Image ID	Start Address	—	—	—
Update Version Threshold	BMC	PFR I^2C	Update Manifest with version threshold	Image ID	Threshold	—	—	—
Program Key	BMC	PFR I^2C	Update Manifest with image's public key	Image ID	Public Key	—	—	—
Enable SPI Filter	BMC	PFR I^2C	Turn on/off SPI command filtering for Flash monitor	Flash ID	0 – Disable 1 – Enable	—	—	—
Enable Secure I2C	BMC	PFR I^2C	Turn on/off Encrypted I^2C Communication between BMC and MachXO3D. Note: ENCRYPT_SUPPORT is required to be set in PFR software.	0 – Disable 1 – Enable	—	—	—	—
Update White/Black Space	BMC	PFR I^2C	Update manifest with individual space definition for flash monitoring. Gray Space – Read Only, White Space – Read, Prog and Erase Allowed, Black Space – Read, Prog and Erase Blocked	Flash ID	Space ID	0 – Gray Space 3 – White Space 4 – Black Space	Start Address	End Address
Select Flash	BMC	PFR I^2C	Select which flash is active	Flash ID	1 – Pri 2 – Sec	—	—	—

Command	Supported Device	Type	Description	Option 1	Option2	Option 3	Option4	Option 5
Write Manifest to Flash	BMC	PFR I^2C	Write Manifest updates to UFM2.	—	—	—	—	—
Read Time	BMC	PFR I^2C	Read time	—	—	—	—	—
Set Time	BMC	PFR I^2C	Set time in seconds	Time (32-bit)	—	—	—	—
Flash EAR Write	BMC	ECP5 SPI	Write the EAR Register	EAR Setting: 0/1	—	—	—	—
Flash Sector Erase	BMC/PCH	ECP5 SPI	Send Erase Command (0x20 or 0xD8) to SPI Flash	Flash Address	1= 4K Erase 2= 32K Erase 3= 64K Erase	—	—	—
Flash Chip Erase	BMC/PCH	ECP5 SPI	Send Chip Erase Command (0xC7) to SPI Flash	—	—	—	—	—
Flash Page Write	BMC/PCH	ECP5 SPI	Send Page Write Command (0x02) and 16 bytes of write data to the SPI Flash	Flash Address	Page Data (16 bytes)	—	—	—
Flash Quad Write	PCH	ECP5 SPI	Send Page Write Command (0x3E) and 16 bytes of write data to the SPI Flash	Flash Address	Page Data (16 bytes)	—	—	—
Flash Byte Write	BMC/PCH	ECP5 SPI	Send Page Write Command (0x02) and single byte of write data to the SPI Flash	Flash Address	Data (1 byte)	—	—	—
Flash Page Read	BMC/PCH	ECP5 SPI	Send Page Read Command (0x03) to SPI Flash and read 16 bytes of data	Flash Address	—	—	—	—
Flash Quad Read	PCH	ECP5 SPI	Send Page Read Command (0x6C) to SPI Flash and read 16 bytes of data	Flash Address	—	—	—	—

Appendix A. Adding a Manifest Manager

The manifest is used to provide system information required by PFR software to manage the authentication of image and configuration of the SPI Monitors. The Manifest Manager is used to create or edit the manifest which is stored in UFM2.

To view the manifest used for the Lattice Sentry Root-of-Trust Demo:

1. Open Lattice Propel and select **LatticeTools > Lattice Sentry Manifest Manager**.
2. Click **Open** and select *<path>\bitstreams\manifest.mem*. The Manifest Manager, as shown in Figure A.1, allows you to:
   - **Image Count** – Select Number of Images.
   - **Flash Count** – Select Number of Flash Interfaces to be monitored.
   - **I²C Filter Count** – Not supported at this time.
   - **Manifest Name** – Provide name for .mem and .jed file.
   - **Append time to filename** – Tick the box to append date/time to file name.
   - **Generate Manifest** – This creates a .mem file for UFM2 initialization and .jed to program into UFM2. The files are stored your root workspace directory.

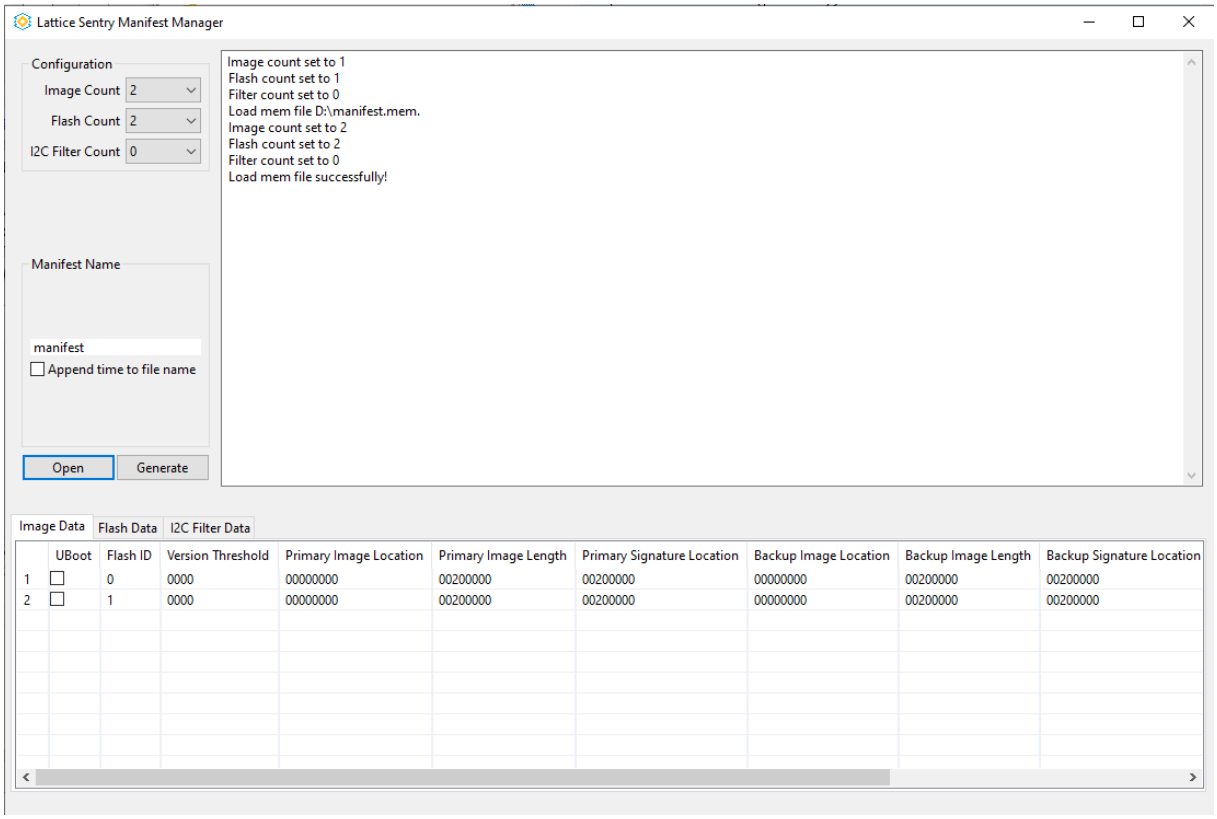

Figure A.1. Manifest Manager

The Image Data tab allows you to add specific details about each image. Table A.1 shows the details of the Image Data parameters.

Table A.1. Image Data Parameter Description

Image Data Parameter	Description
UBoot	Check to indicated golden image is stored in CFG1. If Primary and Backup fail authentication, the image stored in CFG1 is used to overwrite.
Flash ID	Indicates the flash where the image is stored
Version Threshold	16-bit version threshold, the version stored in the image fails if lower than version threshold
Primary Image Location	Location of primary image in flash
Primary Image Length	Size of primary image
Primary Signature Location	Location of primary signature in flash
Backup Image Location	Location of backup in flash
Backup Image Length	Size of backup image
Backup Signature Location	Location of backup signature in flash
Version Offset	Offset location of version in image with respect to image location
Public Key	256-byte Public key used for image authentication

The Flash Data tab allows you to add specific details about each image. Table A.2 shows the details of the Flash Data parameters.

Table A.2. Flash Data Parameter Description

Flash Data Parameter	Description
Dummy Cycles	Number of dummy cycles to be used for SPI
4-Byte Addr	Flash supports 4-byte addressing
Dual Flash	Secondary Flash used for storing backup images *(design assumes secondary flash is used)*
QSPI	Flash supports QSPI
Block Init Cmds	Check to block initialization commands to flash
Block Read Cmds 0	Check to block reads to address space 0
Allow Erase Cmds 0	Check to allow erase command to address space 0
Allow Pgm Cmds 0	Check to allow program commands to address space 0
Block Read Cmds 1	Check to block reads to address space 1
Allow Erase Cmds 1	Check to allow erase command to address space 1
Allow Pgm Cmds 1	Check to allow program commands to address space 1
Block Read Cmds 2	Check to block reads to address space 2
Allow Erase Cmds 2	Check to allow erase command to address space 2
Allow Pgm Cmds 2	Check to allow program commands to address space 2
Block Read Cmds 3	Check to block reads to address space 3
Allow Erase Cmds 3	Check to allow erase command to address space 4
Allow Pgm Cmds 3	Check to allow program commands to address space 4
Staging Area Start Addr	Staging Area start address *(currently not implemented)*
Staging Area End Addr	Staging Area end address *(currently not implemented)*
Addr Space Start 0	Space 0 start address
Addr Space End 0	Space 0 end address
Addr Space Start 1	Space 1 start address
Addr Space End 1	Space 1 end address
Addr Space Start 2	Space 2 start address
Addr Space End 2	Space 2 end address
Addr Space Start 3	Space 3 start address
Addr Space End 3	Space 3 end address

Appendix B. Creating Image Signature

For an image to be authenticated, a signature needs to be created based on a private/public key pair. The public key is stored in the manifest to prevent tampering, while the signature is stored in flash. To ensure proper authentication, the image file and size must match exactly the image used in creating the signature.

For ECP5 images, it is best to read back the image from SPI Flash. This removes any header information in the bitstream that is not loaded into the flash. The readback also gives you the exact size of the image, which is needed for the manifest.

Note: When reading back a SPI image, the Diamond Programmer reads back in 64 KB block boundaries and the end address sector is added to the image. For example, a readback from start address 0x00000000 to end address 0x001E0000 is bytes in size.

To create an image signature:

1. In Diamond, open the MachXO3D project. Note that this feature is not available with other device projects.
2. Click **Tools > Security Settings**.
3. In the pop-up **Enter Password** dialog, enter **LATTICESEMI** as shown in Figure B.1.

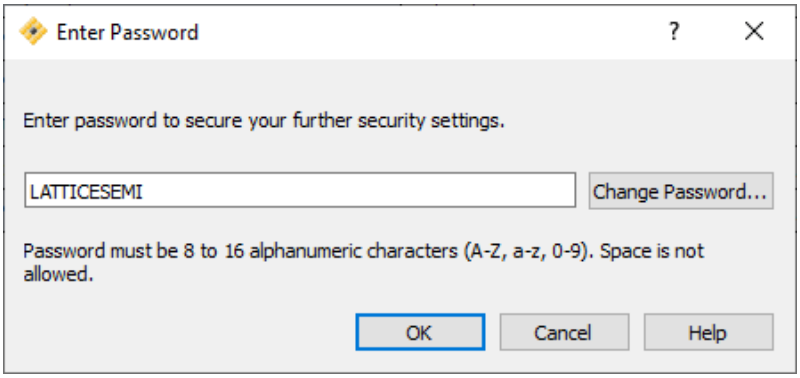

Figure B.1. Enter Password

4. Click the **Signature Authentication** tab from the pop-up **Security Settings** dialog as shown in Figure B.2.
   a. Select **Signature Generation**.
   b. In **Format**, select **Hex**.
   c. If you have a private/public key pair, enter **keys**. For example:
      - Private Key:

        48B3E3E19C81C39A63C00E39E591D3DB876773C875EF957ED0C1CE0BF3744AC3
      - Public Key:

        C72CE0EB37217D6F13787498655CAF3A9A6651CA485BBA8CF2F7B88321BE36390D8C9E3D1B95643F34CF6617CFFBFA3EB84693F893E488808FCF096AF5CCC08D

        If you are using a different key pair, you need to keep track of these keys and the public key needs to be added to the manifest.
   d. Load data file (raw binary).

      Note: Make sure the image size matches the image length entered in the manifest. If there is a size mismatch, the PFR design reads an incorrect amount of data and authentication fails.
   e. Click **Generate**.

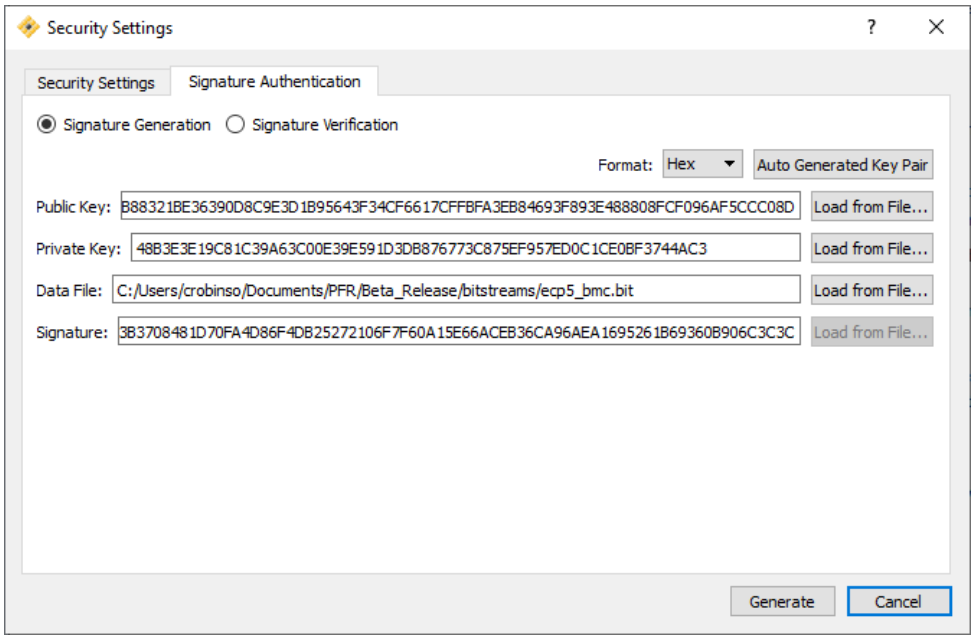

Figure B.2. Generate Signature

The following files are generated:

- *.digest (digest – test format)
- *.sig (signature – text format)
- *.sig.hex (signature – intel hex format)
- This file is used by Diamond Programmer to program the signature into flash.

References

- NIST SP 800 193 Specification
- MachXO3D Family Data Sheet (FPGA-DS-02026)
- MachXO3D Programming and Configuration Usage Guide (FPGA-TN-02069)
- Lattice Sentry Root-of-Trust Reference Design for MachXO3D (FPGA-RD-02203)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.0, August 2020

Section	Change Summary
All	Initial release

www.latticesemi.com