



MachXO3D Soft Error Detection (SED)/Soft Error Correction (SEC) Demo

User Guide

FPGA-UG-02071-1.1

November 2019

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

Acronyms in This Document	5
1. Introduction	6
1.1. Demo Design Overview	6
1.2. MachXO3D Development Board and Resources	7
2. Function Description	8
3. Demo Package	8
3.1. Hardware Requirements	8
3.2. Software Requirements	8
4. Port Assignments and Descriptions	9
5. Demo Package Directory Structure	10
6. Running the Demo	11
6.1. Programming the MachXO3D Board with the Design	11
6.2. Enabling SED	13
6.3. Loading SEI Bitstream in Background and Detecting Inserted Error	13
6.4. Background Refreshing the Image after Soft Error Detection	15
6.4.1. Providing the Original Bitstream in the Background	16
6.4.2. Transferring Original Bitstream Using REFRESH	17
6.4.3. Transferring Original Bitstream Using External PROGRAMN Pin Assertion	18
7. Rebuilding the Design	19
References	22
Lattice Semiconductor Documents	22
Technical Support	23
Revision History	24

Figures

Figure 1.1. MachXO3D Development Board.....	7
Figure 2.1. MachXO3D Development Board SED/SEC Demo Block Diagram.....	8
Figure 5.1. Demo Package Directory Structure	10
Figure 6.1. Launch Diamond Programmer	11
Figure 6.2. Open Device Properties Dialog Box	11
Figure 6.3. Set Device Properties.....	12
Figure 6.4. Start Programming Operation	12
Figure 6.5. Press SW4 to Force SED to Generate Error.....	13
Figure 6.6. Launch Diamond Programmer.....	13
Figure 6.7. Select the Device	14
Figure 6.8. Set MachXO3D Device Properties.....	14
Figure 6.9. Start XSRAM SEI Fast Program Operation	15
Figure 6.10. Press SW3 to Start SED for the Injected Error	15
Figure 6.11. Reload Design Bitstream in Background through JTAG	16
Figure 6.12. Reload Design Bitstream in Background through JTAG	16
Figure 6.13. Use XSRAM Refresh	17
Figure 6.14. Start XSRAM Refresh Programming Operation	17
Figure 6.15. Auto Background Reconfiguration.....	18
Figure 7.1. Open Design File	19
Figure 7.2. Project Opened	19
Figure 7.3. Set Global Preferences.....	20
Figure 7.4. Generate the Bitstream	21
Figure 7.5. Generate the SEI Bitstream	21

Tables

Table 4.1. Design Port Definitions	9
--	---

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
FPGA	Field Programmable Logic Array
SEC	Soft Error Correction
SED	Soft Error Detection
SEI	Soft Error Injection
SRAM	Static Random Access Memory
PLD	Programmable Logic Device

1. Introduction

An SRAM-based Programmable Logic Device (PLD) stores logic configuration data in SRAM cells. As the number and density of SRAM cells in a PLD increase, the possibility of a memory error that alters the programmed logical behaviour of the system also increases.

The Lattice Semiconductor MachXO3D™ devices have a hardware-implemented circuit called Soft Error Detection (SED) which is used to detect SRAM errors and allows them to be corrected.

The MachXO3D Soft Error Correction (SEC) is performed with background reconfiguration. All bits are rewritten during reconfiguration and the erroneous bit is replaced with the correct data. During background reconfiguration, writing the same value into configuration SRAM cells does not affect the SRAM cell output.

Soft Error Injection (SEI) is a feature of the Lattice Diamond® Programmer software. It is used to inject an error in the MachXO3D configuration SRAM.

This demo shows you all the SED, SEC, and SEI features of the MachXO3D and how the development tool is used.

1.1. Demo Design Overview

This demo design consists of two major parts:

- SED Hard Module – Performs read and error detection of SRAM content.
- Control Logic – A state machine generating control signals for driving SED module. This includes soft error indication and a function block that rotates the onboard LEDs.

The status of the demo is indicated by the onboard LEDs. The SEI bitstream is used to induce a single SRAM error into the running demo. The SEI bitstream is generated using the SEI Editor available in the Tools tab of the Lattice Diamond® software.

This demo covers the following operations:

- Programming the board with the design
- Enabling SED
- Loading the SEI bitstream in the background and detecting the inserted error
- Background refreshing of the image after the error is detected by SED

1.2. MachXO3D Development Board and Resources

Figure 1.1 shows the top side of the MachXO3D Development Board and resources used for the demo.

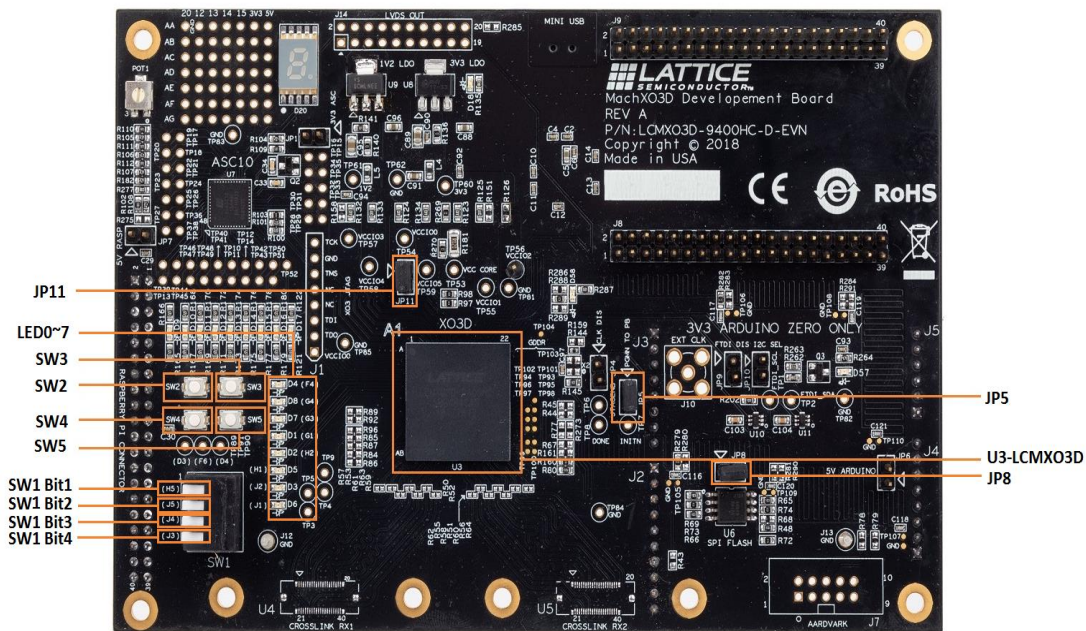


Figure 1.1. MachXO3D Development Board

- **JP11:** Set to provide an external 12 MHz clock for the design.
- **JP5:** Auto SEC when being set.
- **SW1:** A 4-bit DIP Switch; Up – 1; Down – 0; “110x” to enable SED.
- **SW2:** Active low reset input for the design.
- **SW3:** Starts SED for error detection.
- **SW4:** Forces SED to generate an error output for testing when SW1 (Bit4-1) is set to “1101”.
- **SW5:** PROGRAMN when JP5 is set.
- **JP8:** Remove jumper from JP8.
- **LED0~7:** LED0~5 start to shift when the MachXO3D device is programmed with the design. LED6 (D3) is lighted when an error is detected. LED7 (D6) blinking indicates the SED is enabled when SW1 (Bit4~1) is set to “1101”.

2. Function Description

Figure 2.1 shows the block diagram of the demo design.

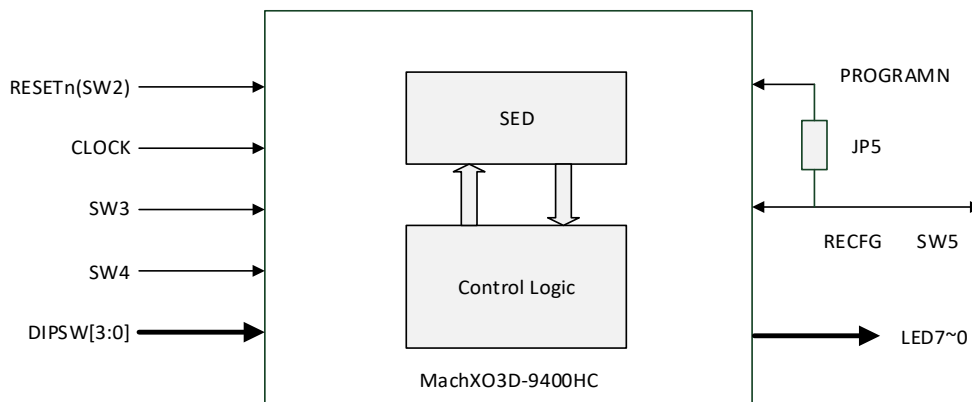


Figure 2.1. MachXO3D Development Board SED/SEC Demo Block Diagram

3. Demo Package

3.1. Hardware Requirements

To run the demo, the following hardware are required:

- PC running Windows 7 Operating System
- MachXO3D Development Board
- Mini USB cable for programming the MachXO3D device

3.2. Software Requirements

To run the demo, the following software are required:

- Lattice Diamond version 3.11 or later
- Lattice Diamond Programmer software for bitstream downloading

Note: The software programs are available at www.latticesemi.com/en/Products/DesignSoftwareAndIP.

4. Port Assignments and Descriptions

Table 4.1 provides the port assignments and definitions in this demo.

Table 4.1. Design Port Definitions

Port Name	I/O Type	Description
RESETn	Input	Asynchronous reset. Active low (SW2).
CLK	Input	External 12 MHz clock input.
DIPSW[3:0]	Input	Up – 1 Down – 0 DIPSW[3] – Bottom DIPSW[0] – Top DIPSW[3:0]='1101' enables SED detection. LED7 (D6) starts blinking.
SW3	Input	Starts SED for a detection cycle.
SW4	Input	Forces SED to generate an error output and turns ON LED6 (D3) when DIPSW[3] is Up(1).
RECFG	Output	For background SEC through reconfiguring the MachXO3D device after an error is detected. Connected with PROGRAMn and SW5, high impedance state after configuration.
LED[5:0]	Output	Circulate lighting of LED0~5 (D4, D8, D7, D1, D2, D5), indicates user logic is running.
LED6	Output	D3. Indicates an error is detected by SED.
LED7	Output	D6. When blinking, indicates the SED is enabled.

5. Demo Package Directory Structure

Figure 5.1 shows the demo package directory structure.

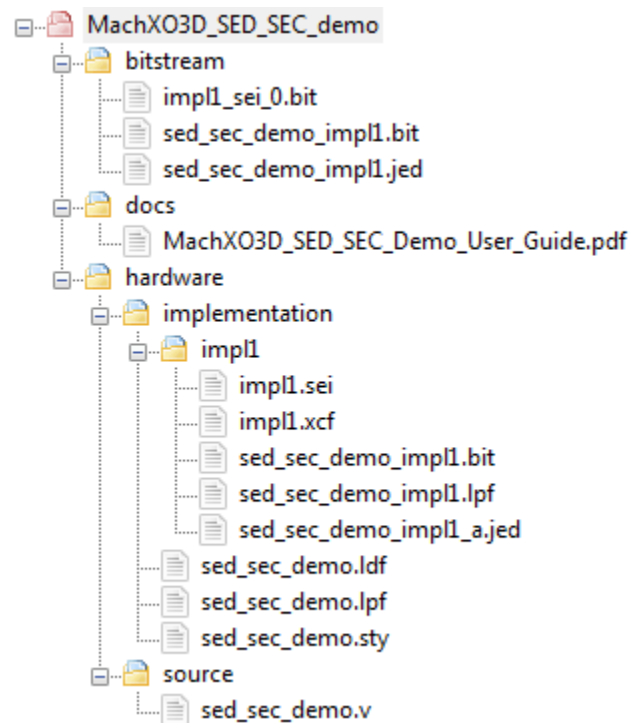


Figure 5.1. Demo Package Directory Structure

6. Running the Demo

6.1. Programming the MachXO3D Board with the Design

To load and run the SED/SEC demo design:

1. Before powering ON the board, make sure JP5 is set to 1.
2. Power ON the board by connecting it to the PC through the USB cable.
3. Launch the Diamond Programmer software (version 3.11 or later). In the **Getting Started** dialog box, select **Create a new project file from JTAG scan**.
4. Click **OK** (Figure 6.1).

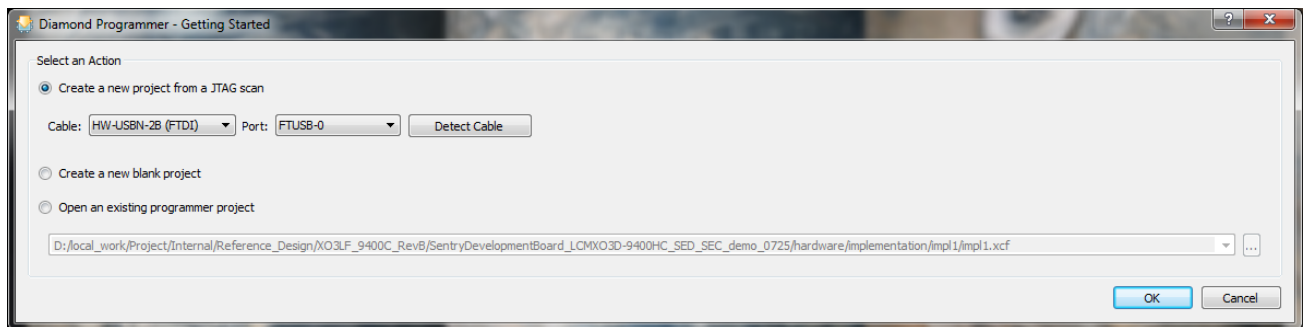


Figure 6.1. Launch Diamond Programmer

5. The LCMXO3D device is detected and displayed. Right-click on the device and select **Device Properties...** from the right-click menu (Figure 6.2).

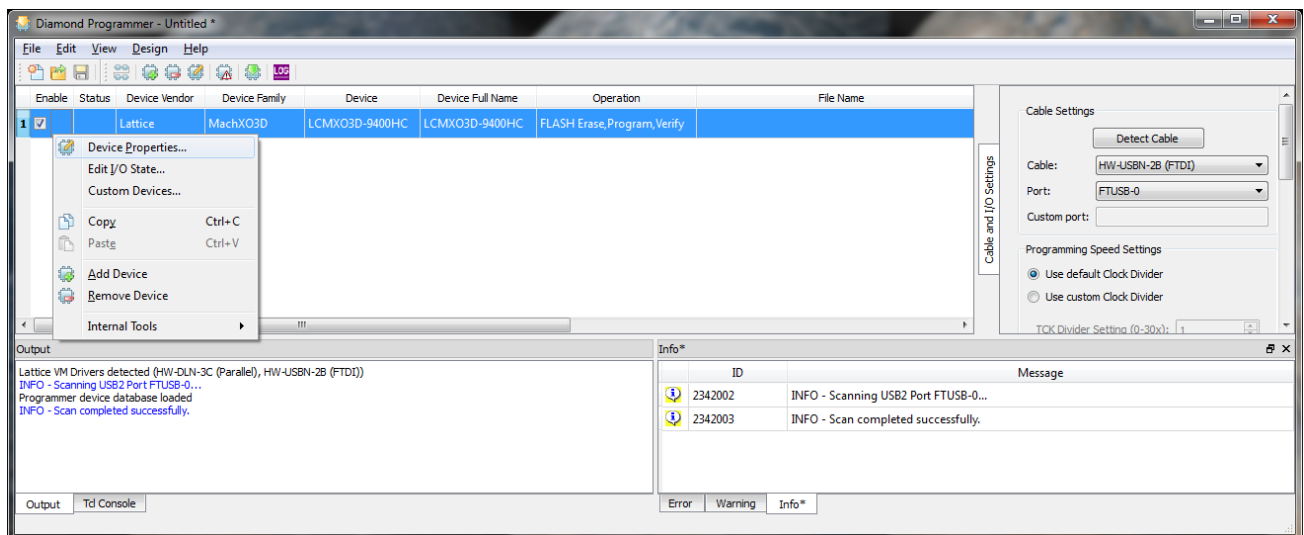


Figure 6.2. Open Device Properties Dialog Box

6. Set device properties as shown in Figure 6.3.
In the **Device Properties** dialog box, select **Flash Programming Mode** in **Access mode**, **Flash Erase, Program, Verify** in **Operation**. In the **Flash-A Programming Options** area, select the bitstream file `\bitstream\sed_sec_demo_impl1.jed` from **Programming file**.
7. Click **OK**.

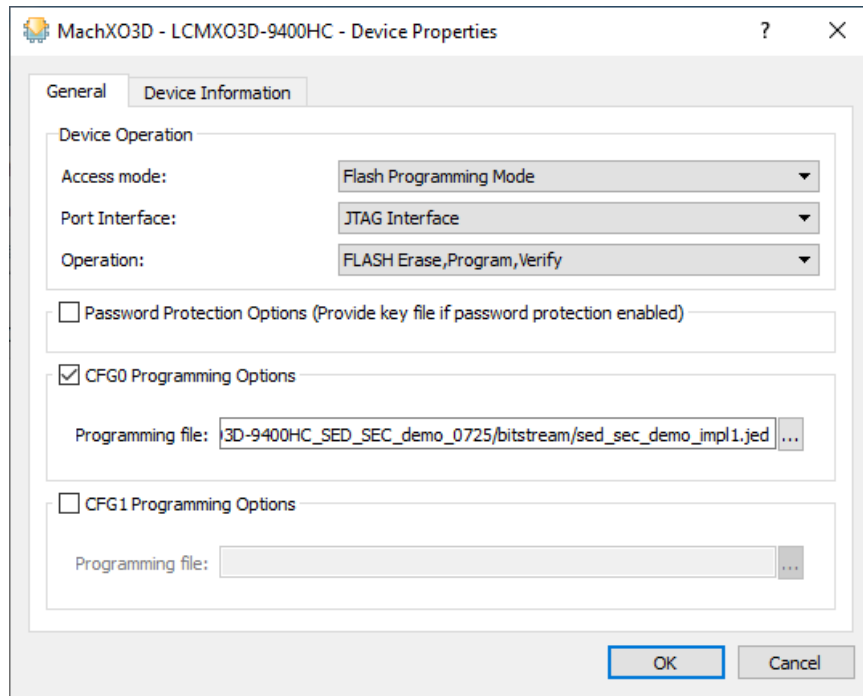


Figure 6.3. Set Device Properties

8. Click the **Program** button on the toolbar (Figure 6.4) to start programming.

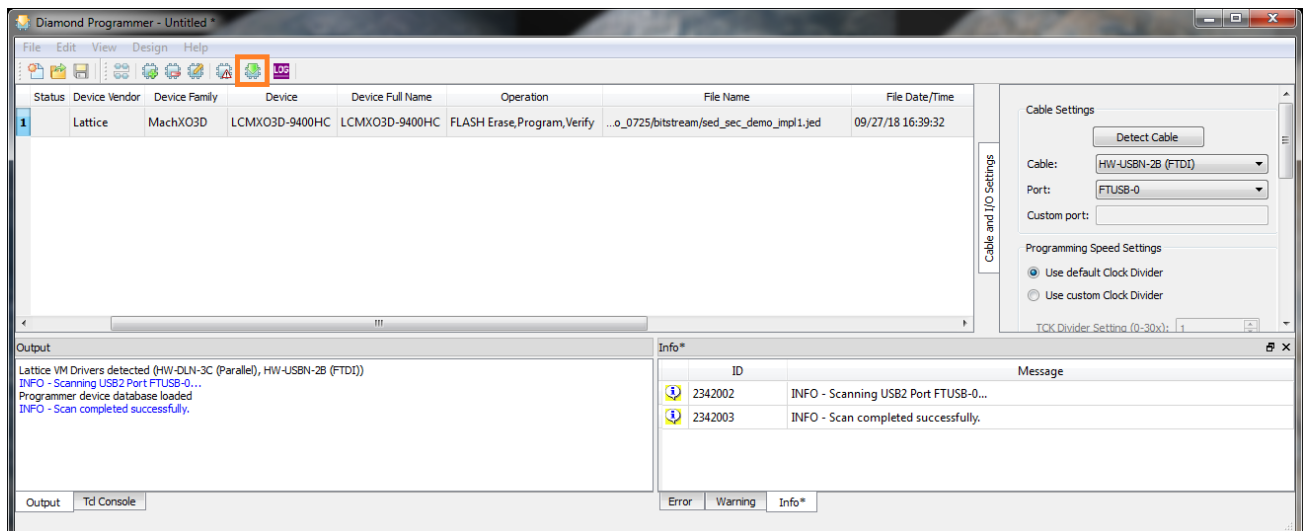


Figure 6.4. Start Programming Operation

If the programming is completed successfully, “Operation: successful” is reported in the Output panel. LEDs are circulated lightening one after another.

6.2. Enabling SED

To enable SED:

1. Make sure JP11 is set to provide the external 12 MHz clock for the design.
2. Set SW1 (DIPSW[3:0]) all bits up.
3. Push down SW1 BIT2 (DIPSW[1]) to enable SED. LED7 (D6) starts to blink. This indicates that SED is enabled.
4. Force SED to generate an error.
 - a. Set SW1-BIT1 DIPSW[0] up.
 - b. Press SW4 to generate the error.
 LED6 (D3) turns ON. This indicates that an error is generated.

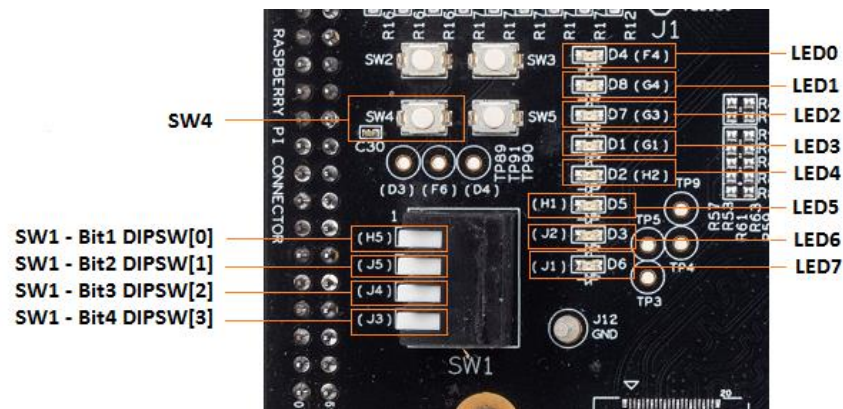


Figure 6.5. Press SW4 to Force SED to Generate Error

5. Press SW3 to start the SED detection cycle. LED6 (D3) turns OFF. This indicates that no error is injected/detected. See the [Loading SEI Bitstream in Background and Detecting Inserted Error](#) section for error injection and detection.

6.3. Loading SEI Bitstream in Background and Detecting Inserted Error

To insert an error in the SRAM array:

1. Launch the Diamond Programmer software (version 3.11 or later). In the **Getting Started** dialog box, select **Create a new project file from JTAG scan**.
2. Click **OK** (Figure 6.6).

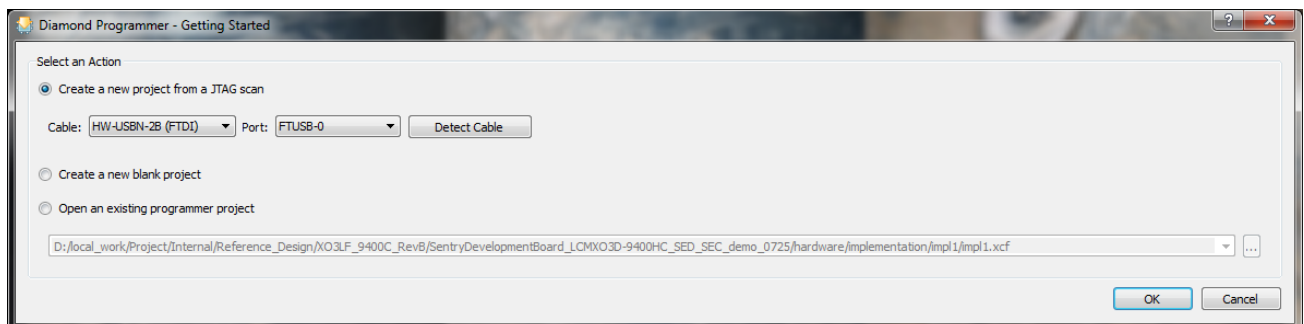


Figure 6.6. Launch Diamond Programmer

3. The LCMXO3D device is detected and displayed. Right-click on the device and select **Device Properties...** from the right-click menu (Figure 6.7).

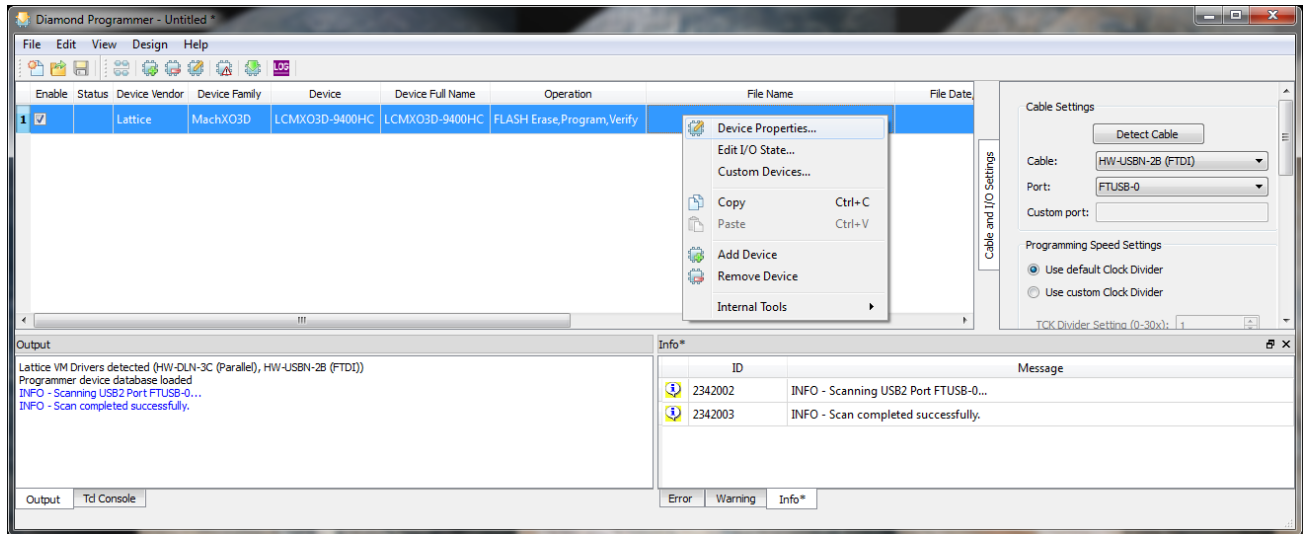


Figure 6.7. Select the Device

- Set device properties as shown in Figure 6.8.
In the **Device Properties** dialog box, select **Static RAM Cell Background Mode** from **Access mode**, **XSRAM SEI Fast Program** in **Operation**. In the **Programming Options** area, browse to select the SEI bitstream file `\bitstream\impl1_sei_0.bit` in Programming file.
- Click **OK**.

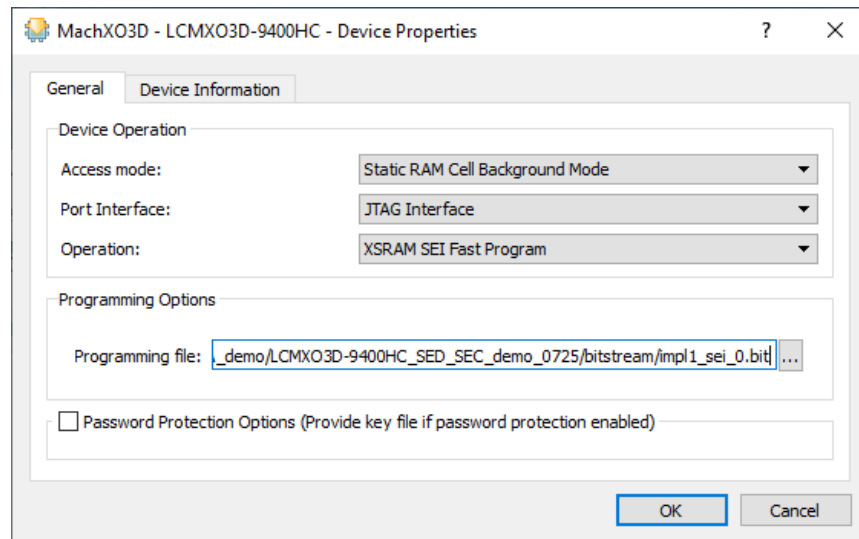


Figure 6.8. Set MachXO3D Device Properties

- Click the **Program** button on the toolbar (Figure 6.9) to initiate background programming.

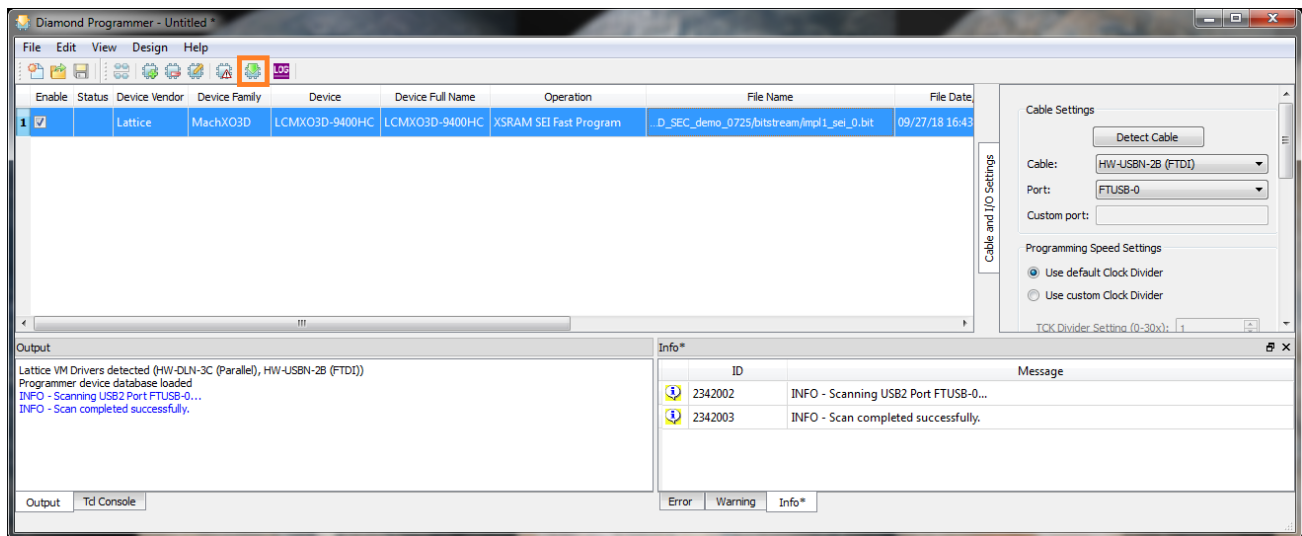


Figure 6.9. Start XSRAM SEI Fast Program Operation

7. Remove jumper from JP5.
8. With JP5 jumper removed, press SW3 again. When SW3 is pressed, LED6 (D3) turns OFF. If LED6 (D3) turns ON when SW3 is released, this indicates an error exists and SEC is not working.
9. Set JP5 and press SW3 again. LED6 (D3) does not turn ON after pressing SW3 a second time. This indicates that SEC is already working at the background by driving PROGRAMn low based on the design.

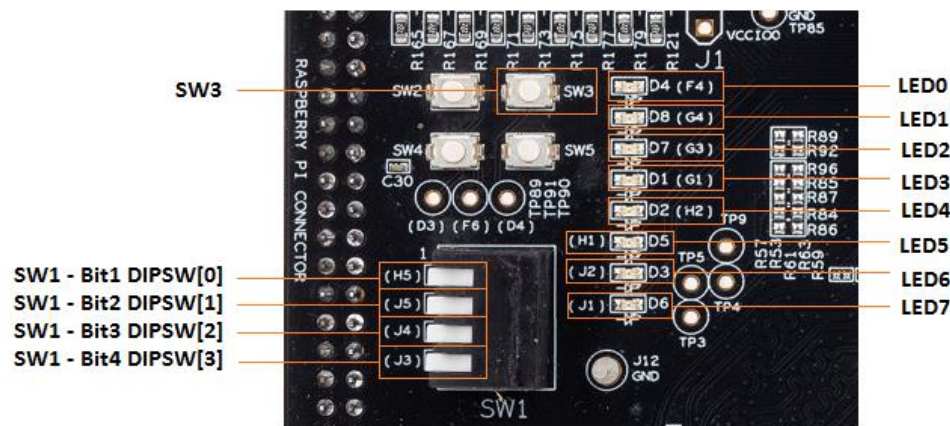


Figure 6.10. Press SW3 to Start SED for the Injected Error

6.4. Background Refreshing the Image after Soft Error Detection

There are several ways to reload the design bitstream at the background. The MachXO3D device supports the following methods of refreshing the SRAM array and correcting detected soft errors.

- Providing the original bitstream at the background through an external sysConfig™ slave port. For example, this can be done through JTAG or by using I²C/SPI embedded programming technique.
- Transferring the original bitstream from internal or external Flash memory, initiate through sysConfig REFRESH command.
- Transferring the original bitstream from internal or external Flash memory, initiate through external PROGRAMN pin assertion. This is demonstrated in steps 8 and 9 of the [Loading SEI Bitstream in Background and Detecting Inserted Error](#) section.

6.4.1. Providing the Original Bitstream in the Background

Refreshing the SRAM array with the original bitstream through the JTAG port is similar to loading the SEI bit file in the [Loading SEI Bitstream in Background and Detecting Inserted Error](#) section. The only difference is the bitstream to be used for loading.

To provide the original bitstream in the background:

1. Make sure JP5 is removed and repeat the appropriate steps in the [Loading SEI Bitstream in Background and Detecting Inserted Error](#) section to inject error again.
2. In the **Device Properties** dialog box, browse to select the design bitstream file `\bitstream\sed_sec_demo_impl1.bit` from **Programming file** in the **Programming Options** area.
3. Click **OK** (Figure 6.11).

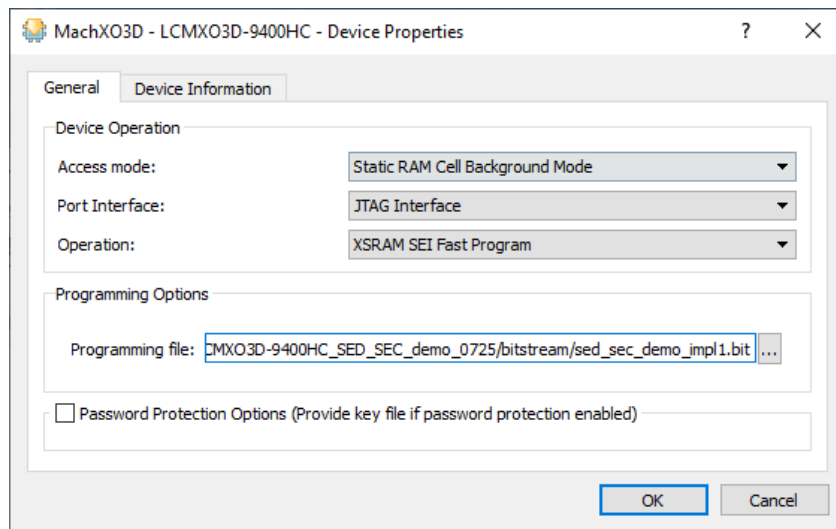


Figure 6.11. Reload Design Bitstream in Background through JTAG

4. Click the **Program** button on the toolbar, as shown in Figure 6.12, to initiate background programming of the device. The original design bitstream is loaded after the software reports “Operation: successful”.

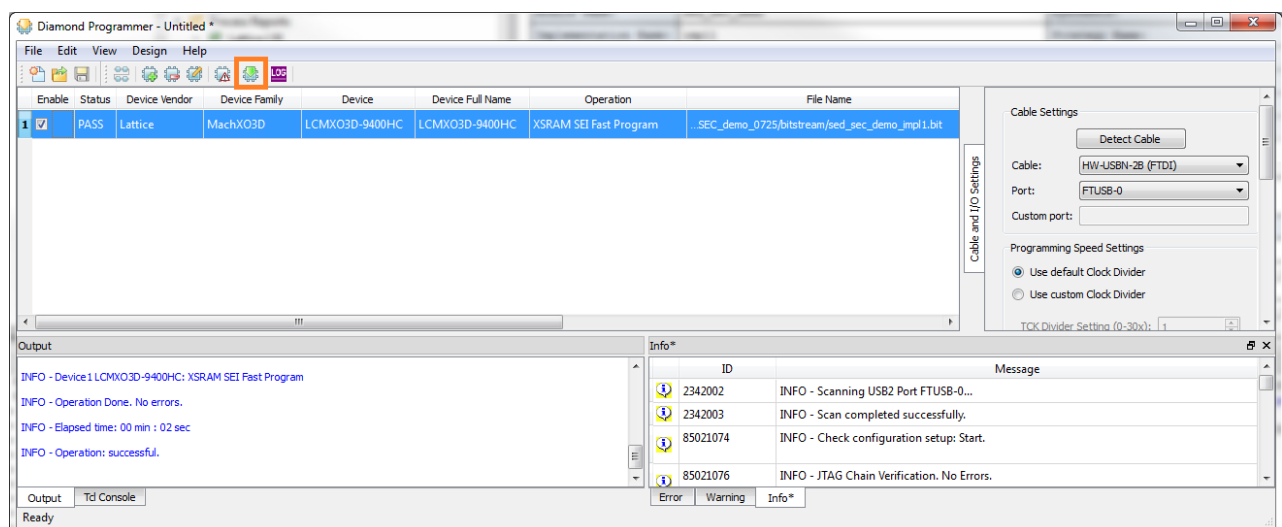


Figure 6.12. Reload Design Bitstream in Background through JTAG

5. Press SW3 to start the SED. LED6 (D3) turning OFF indicates that the injected error is corrected by background reconfiguration.

6.4.2. Transferring Original Bitstream Using REFRESH

Reloading the design bitstream at the background refreshes the SRAM array from the internal flash using the REFRESH command.

To load the original design bitstream from internal flash:

1. Make sure JP5 jumper is removed and repeat the appropriate steps in the [Loading SEI Bitstream in Background and Detecting Inserted Error](#) section to inject the error again.
2. In the **Device Properties** dialog box, select **Static RAM Cell Background Mode** from **Access mode**, **XSRAM Refresh** in **Operation**, as shown in [Figure 6.13](#).
3. Click **OK** after the command is set.

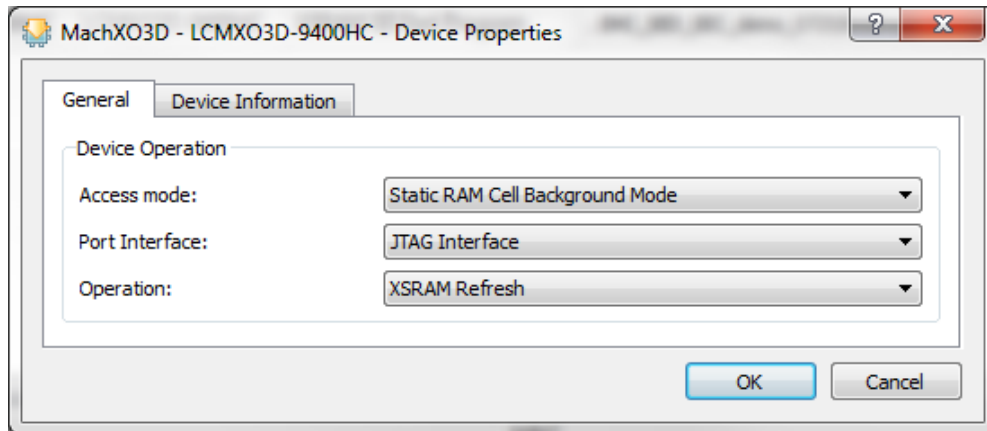


Figure 6.13. Use XSRAM Refresh

4. Click the **Program** button on the toolbar, as shown in [Figure 6.14](#) to initiate background programming of the device. The original design bitstream is loaded after the software reports “Operation: successful” while the logic is working.

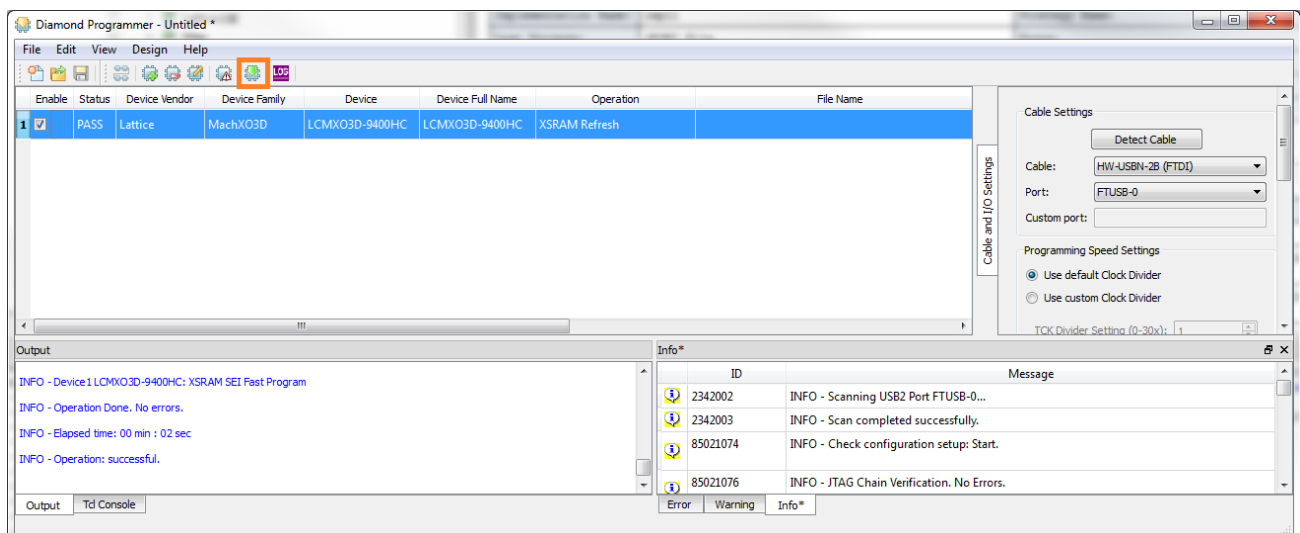


Figure 6.14. Start XSRAM Refresh Programming Operation

5. Press SW3 to start the SED. LED6 (D3) is not turned ON. This means the injected error is corrected by background refresh.

6.4.3. Transferring Original Bitstream Using External PROGRAMN Pin Assertion

To transfer the original bitstream from internal or external Flash memory by asserting external PROGRAMN pin:

1. Set JP5 by adding jumper on, as shown in [Figure 6.15](#).

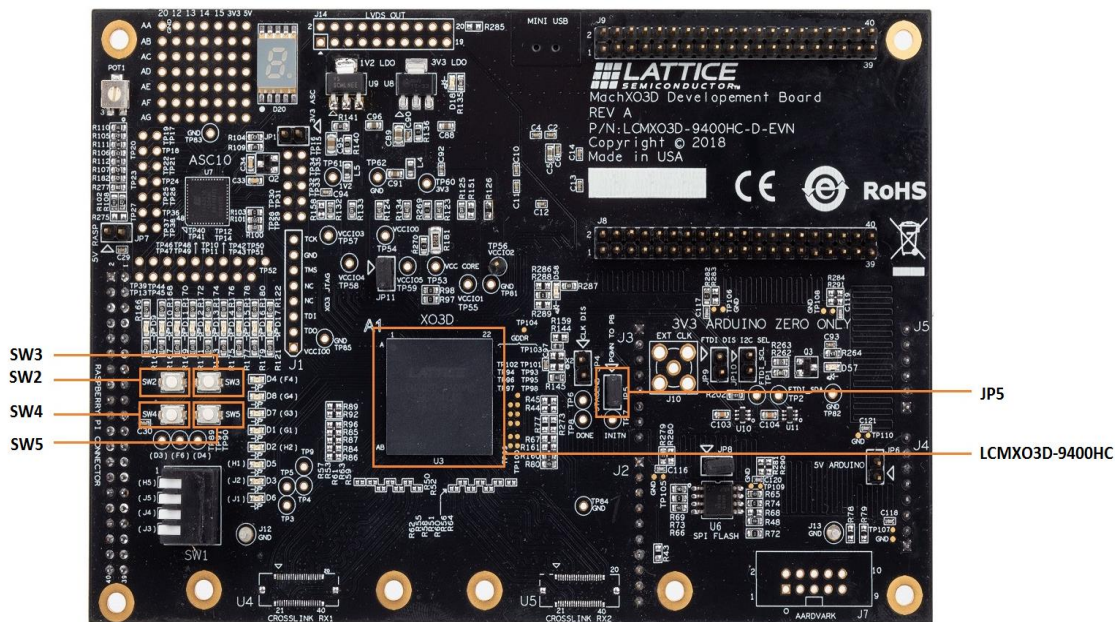


Figure 6.15. Auto Background Reconfiguration

2. After SEI operation is completed, as described in the [Loading SEI Bitstream in Background and Detecting Inserted Error](#) section, press SW3 to start SED. LED6 (D3) turns ON. This indicates that the error is injected. At the same time, the design drives PROGRAMN low if JP5 is set to reconfigure the device in the background while the logic is still working.
3. Press SW3 again to restart SED and detect no error. The LED6 (D3) is not turned ON. This indicates that the injected error is corrected by the background reconfiguration.

7. Rebuilding the Design

To rebuild the design:

1. Start the Diamond software. In the Navigator window, click **Open** under Project. Open the Lattice design file (*.ldf) located in the **./hardware/implementation** folder.

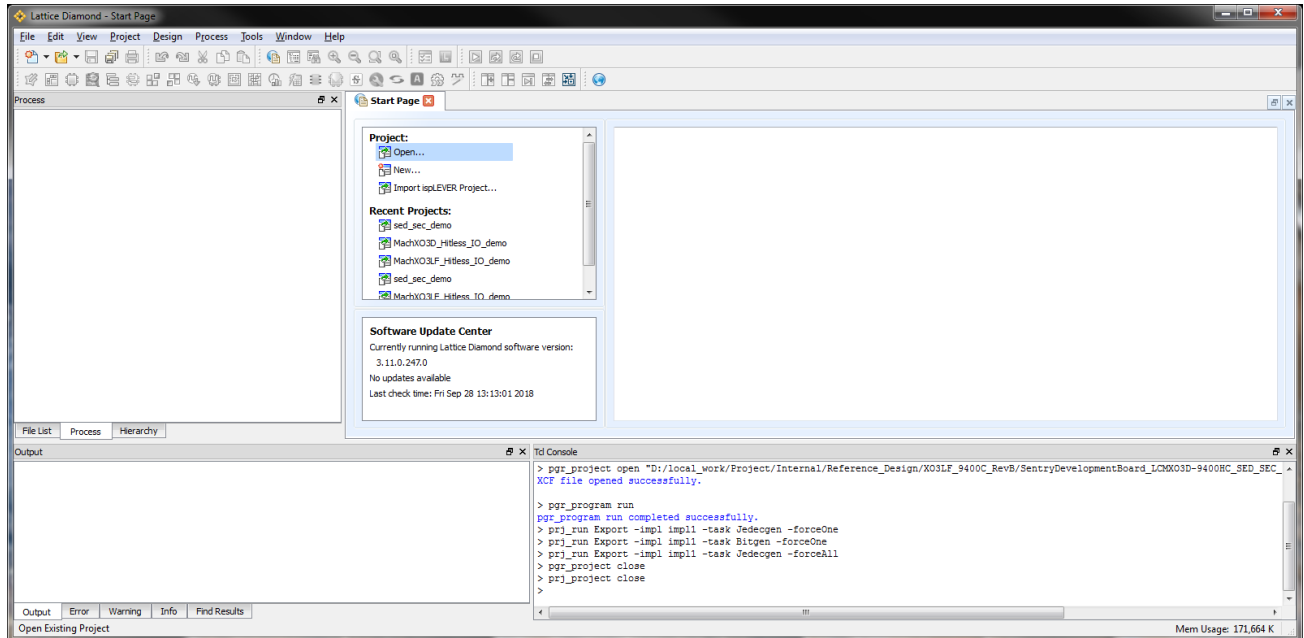


Figure 7.1. Open Design File

2. The opened project is shown in Figure 7.2.

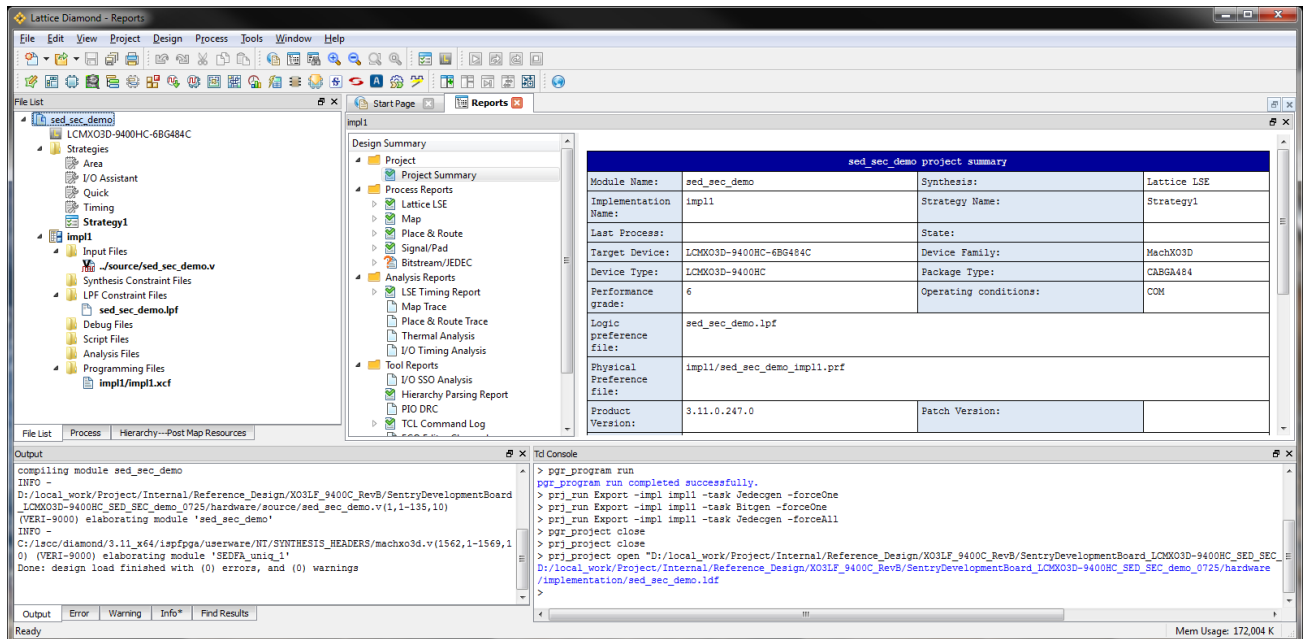


Figure 7.2. Project Opened

3. Synthesize the design to make sure there is no error. Refer to the user guide of Lattice Diamond software for details on how to perform this process.

- When you see a green check mark beside the **Synthesize Design** process in the Process view, choose **Tools > Spreadsheet View** or click the **Spreadsheet View** icon  on the toolbar to open the Spreadsheet View. In the Global Preferences tab, set **SDM_PORT** to **PROGRAMN** and **BACKGROUND_RECONFIG** to **ON**, as shown in Figure 7.3.

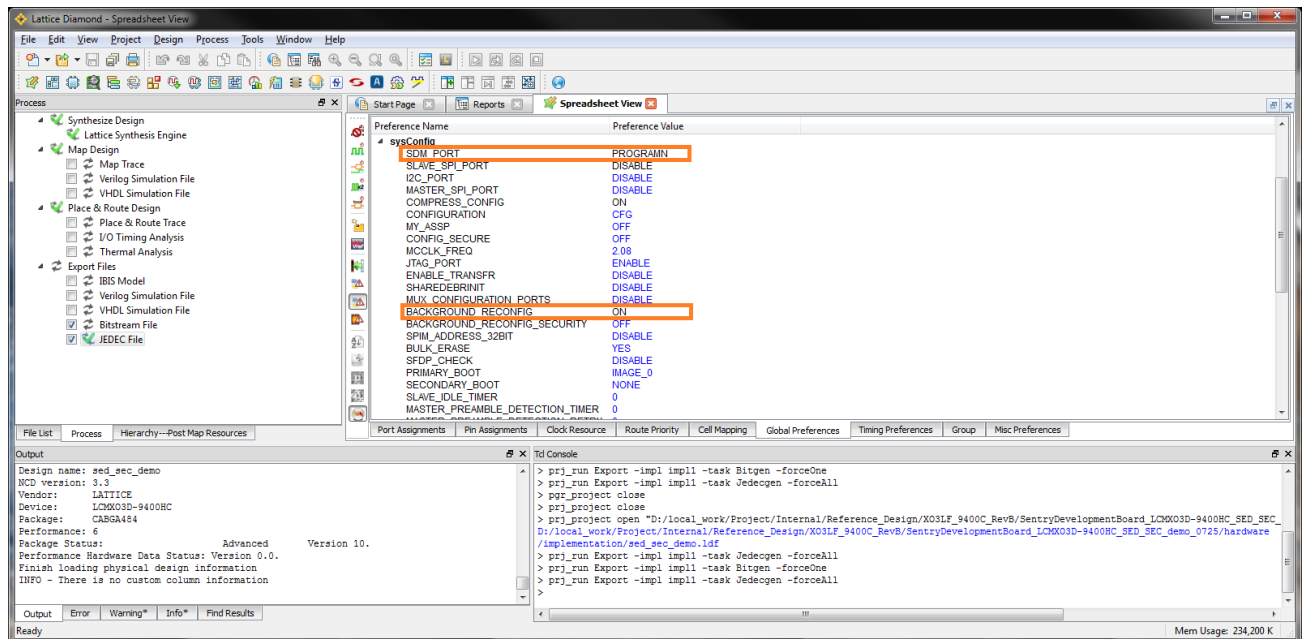


Figure 7.3. Set Global Preferences

- Click **File > Save** to save your settings.
- In the Process view, under the Export Files process, select the **Bitstream File** and **JEDEC File** checkboxes, as shown in Figure 7.4.
- Double click the **Export Files** process.
- From the menu bar, select **Process > Rerun All** to run the entire process. The bitstream file and the JEDEC file are generated.

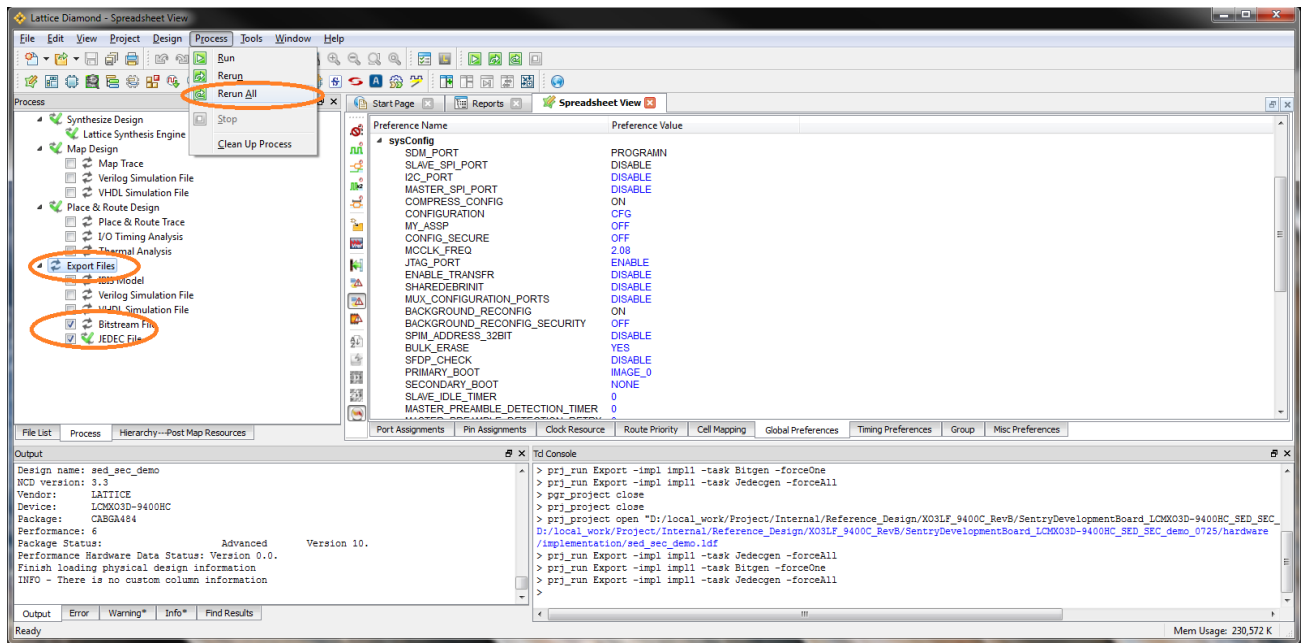


Figure 7.4. Generate the Bitstream

The SEI bitstream for error injection can be generated using the SEI Editor tool, as shown in Figure 7.5.

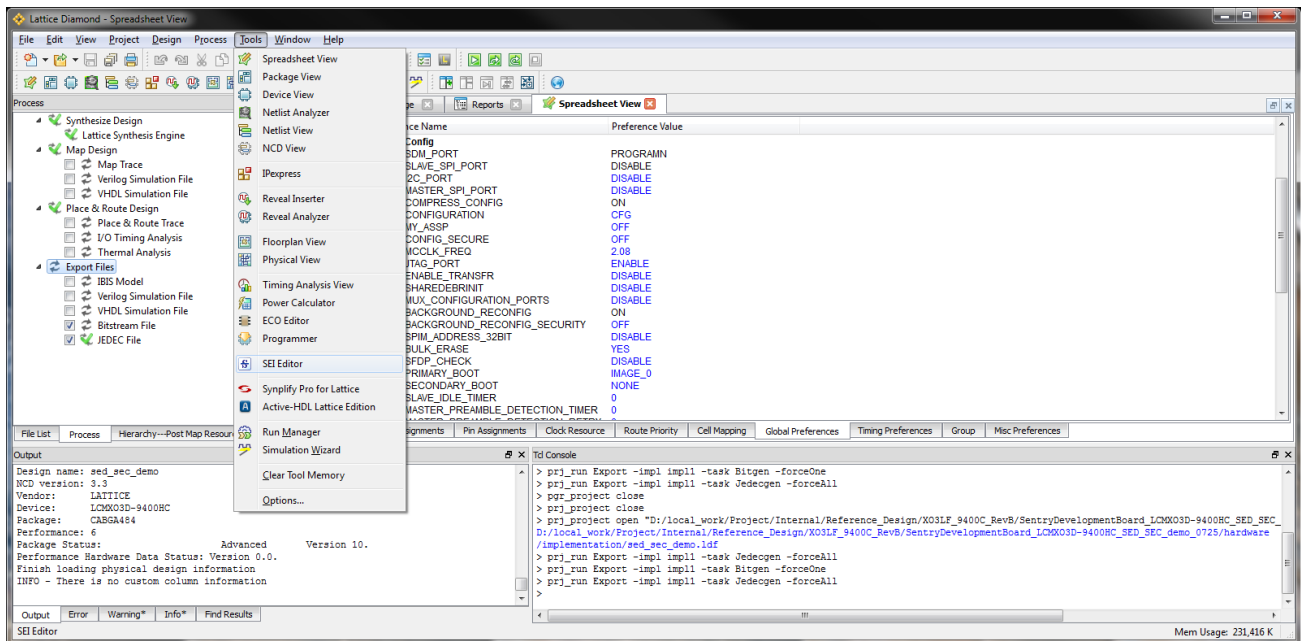


Figure 7.5. Generate the SEI Bitstream

The rebuilt design can be programmed in the MachXO3D device, as described in the Running the Demo section.

References

Lattice Semiconductor Documents

This is a list of related documents that are available from your Lattice Semiconductor sales representative.

Document	Title
FPGA-DS-02026	MachXO3D Family Data Sheet
FPGA-EB-02020	MachXO3D Development Board User Guide
TN1292	MachXO3 Soft Error Detection (SED)/Correction (SEC) Usage Guide
FPGA-TN-02069	MachXO3D Programming and Configuration Usage Guide

Technical Support

For assistance, submit a technical support case at www.latticesemi.com/techsupport.

Revision History

Revision 1.1, November 2019

Section	Change Summary
Disclaimer	Added this section.
Programming the MachXO3D Board with the Design	Updated Figure 6.3 .
Loading SEI Bitstream in Background and Detecting Inserted Error	Updated Figure 6.8 .
Background Refreshing the Image after Soft Error Detection	Updated Figure 6.11 .

Revision 1.0, December 2018

Section	Change Summary
All	Initial release.



www.latticesemi.com