MachXO3 Soft Error Detection (SED)/Correction (SEC) Usage Guide

Technical Note

FPGA-TN-02062-1.4

January 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
IP	Intellectual Property
SED	Soft Error Detection
SEC	Soft Error Correction
SEI	Soft Error Injection
EFB	Embedded Functional Block

1. Introduction

Memory errors can occur when high-energy charged particles alter the stored charge in a memory cell in an electronic circuit. The phenomenon first became an issue in DRAM, requiring error detection and correction for large memory systems in high-reliability applications. As device geometries have continued to shrink, the probability of memory errors in SRAM has become significant for some systems. Designers are using a variety of approaches to minimize the effects of memory errors on system behavior.

SRAM-based PLDs store logic configuration data in SRAM cells. As the number and density of SRAM cells in an PLD increase, the probability that a memory error will alter the programmed logical behavior of the system increases. A number of approaches have been taken to address this issue, but most involve Intellectual Property (IP) cores that the user instantiates into the logic of their design, using valuable resources and possibly affecting design performance. The MachXO3™ devices have a hardware implemented SED circuit which can be used to detect SRAM errors and allow them to be corrected.

This document describes the hardware-based Soft Error Detection (SED) approach taken by Lattice Semiconductor for MachXO3L/LF PLDs. Once soft error is detected, Lattice provides an easy way to perform the Soft Error Correction (SEC) without disturbing the functionality of the device.

Lattice also provides a tool to help the user emulate soft error impact by inserting soft error into the device. More details are provided in the MachXO3LF SEI (Soft Error Injection) section.

Table 1.1. SED/SEC/SEI Features

Device	SED	SEC	SEI
MachXO3L	Yes	No	No
MachXO3LF	Yes	Yes	Yes

2. SED Overview

The SED hardware in the MachXO3L/LF devices is part of the Embedded Functional Block (EFB) consists of an access point to the PLD's Configuration Logic, a Controller Circuit, and a 32-bit register to store the CRC for a given bitstream (see Figure 2.1). The SED hardware reads serial data from the PLD's Configuration memory and calculates a CRC. The data that is read, and the CRC that is calculated, does not include EBR memory or PFUs used as RAM. The calculated CRC is then compared with the expected CRC that was stored in the 32-bit register. If the CRC values match it indicates that there has been no configuration memory corruption, but if the values differ an error signal is generated.

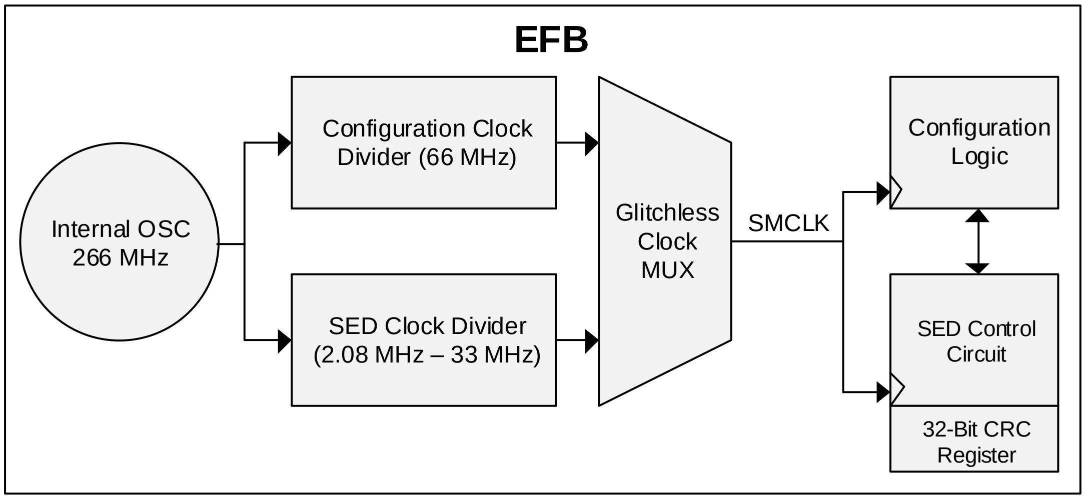


Figure 2.1. System Block Diagram

Note that the calculated CRC is based on the particular arrangement of configuration memory for a particular design. Consequently, the expected CRC results cannot be specified until after the design is placed and routed. The Lattice

Diamond® bitstream generation software analyzes the configuration of a placed and routed design and updates the 32-bit SED CRC register contents during bitstream generation.

The following sections describe the MachXO3L/LF SED implementation and flow.

3. SED Limitations

SED should only be run when once Vcc reaches the data sheet Vcc minimum recommend level. In addition, clock frequencies of greater than 33.33 MHz for the SED are not supported.

The clock (SMCLK) of the SED circuit is shared with the Configuration Logic. As a result, the SED module interacts with several EFB functions with the following results:

- If the EFB or Configuration Logic is accessed while the SED circuit is running:
  - The current SED cycle will be terminated:
    - When the SED circuit is terminated there will be a delay of two SMCLK cycles before EFB or Configuration Logic can be accessed. This is a result of the SMCLK transferring clock from the SED Clock to the Configuration Clock domain. The two SMCLK cycles are defined by the slower SED clock.
    - When the SED circuit is terminated the SEDDONE will remain low, SEDERR will remain low, and SEDINPROG resets from high to low.
    - The EFB or Configuration Logic access which interacts with the SED circuit is defined as:
      - The following commands issued through the JTAG port or WISHBONE interface:
        - LSC_REFRESH
        - ISC_ENABLE
        - ISC_ENABLE_X
        - All IEEE 1532 instructions
        - ISC_DISABLE
    - Primary I2C Configuration Logic slave address match
    - SPI Configuration Logic chip select being asserted.
- The PROGRAMN pin detection logic requires the minimal low period be longer than six SMCLK cycles. If the SED circuit is running the six SMCLK cycles are defined by the SED clock.

4. SED Operating Modes

For MachXO3L/LF devices there are two operating modes available for SED:

- Standard mode allows the design to control when the SED is run and to test the error detection operation.
- One-shot operation is used to run the SED once when the device is first configured to ensure that the configuration matches the desired configuration.

Both operations perform a single cycle which checks the CRC of all the bits in the SRAM except the EBR and RAM memory. Standard mode is activated using the SEDFA primitive while the One-Shot operation is activated using the SEDFB primitive. These primitives are described in the next section.

If an error is detected during an SED operation, the user can choose one of two corrective actions to take. One is to “Do Nothing” and the other is to initiate an on-demand user reconfiguration by pulling the PROGRAMN pin low. This can be done from another device or from an output of the MachXO3L/LF device as shown in Figure 9.1.

The PROGRAMn pin detection logic requires the minimal low period be longer than 6 SMCLK cycles. When error detection is actively enabled, the PROGRAMn pin minimal low period will be 6 SEDCLK cycles, since the active SMCLK switched to SEDCLK as mentioned previously. This will behave differently then normal operation where the SMCLK is operating at full speed. After booting, the SED function block will behave according to the new configuration programming.

4.1. Standard SED

The Standard SED operation can be used by instantiating the SEDFA primitive which is shown in Figure 4.1. The primitive port definitions are listed in Table 5.1. See the Port Descriptions section of this document for more detailed information about each of the ports.

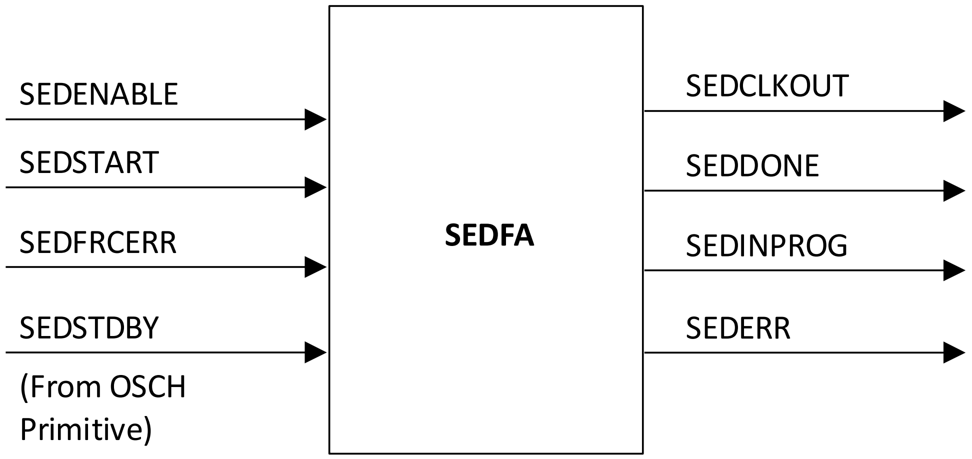


Figure 4.1. SEDFA Primitive Symbol

4.2. One-Shot SED

The One-Shot SED operation can be used by instantiating the SEDFB primitive which is shown in Figure 4.2. The definitions of the ports for the SEDFB primitive are shown in Table 5.1. See the Port Descriptions section of this document for more detailed information about each of the ports.

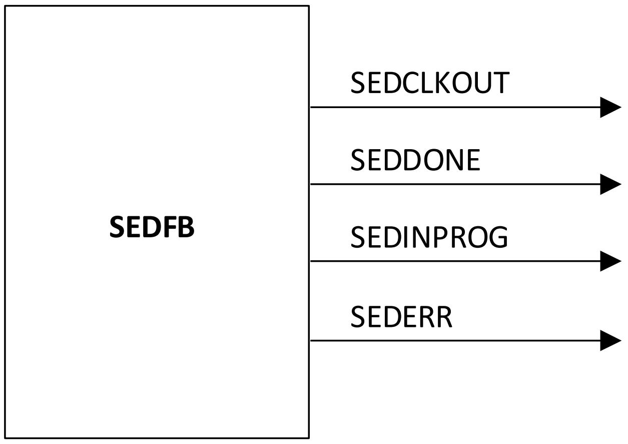


Figure 4.2. SEDFB Primitive Symbol

5. Signal Descriptions

Table 5.1. SEDFA Primitive Port Definitions

Signal Name	Direction	Active	Description
SEDENABLE	Input	High	SRAM CRC enable
SEDSTART	Input	Rising Edge	Start SRAM CRC cycle
SEDFRCERR	Input	Rising Edge	Force an SRAM CRC error flag
SEDSTDBY	Input	High	SRAM CRC disable while in standby mode
SEDCLKOUT	Output	N/A	Output clock
SEDDONE	Output	High	SRAM CRC cycle is complete
SEDINPROG	Output	High	SRAM CRC cycle is in progress
SEDERR	Output	High	SRAM CRC error flag

6. SED Clock Driver

The SED circuitry is driven by the MachXO3L/LF internal oscillator when using either the SEDFA or SEDFB primitive. The maximum frequency supported is 33.25 MHz.

The MachXO3L/LF internal oscillator can be used for several functions in the device including Configuration, SED and as an internal user clock. The frequency of the oscillator output can be set differently for each of these different uses. The settings available for the SED clock are shown in the table below. When using of the SED, the internal oscillator frequency is specified using the SED_CLK_FREQ parameter.

Table 6.1. SED Internal Oscillator Supported Frequency Settings

2.08	4.16	8.31	16.63
2.15	4.29	8.58	17.73
2.22	4.43	8.87	19.00
2.29	4.59	9.17	20.46
2.38	4.75	9.50	22.17
2.46	4.93	9.85	24.18
2.56	5.12	10.23	26.60
2.66	5.32	10.64	29.56
2.77	5.54	11.08	33.25
2.89	5.78	11.57	—
3.02	6.05	12.09	—
3.17	6.33	12.67	—
3.33	6.65	13.30	—
3.50	7.00	14.00	—
3.69	7.39	14.78	—
3.91	7.82	15.65	—

7. SED Attributes

There are three attributes that can be used with the SED primitives and these are shown in Table 7.1. Usage examples for these attributes can be found in the Sample Code section of this document. Currently the SEDFB primitive does not support the three attributes listed.

Table 7.1. SED Attributes

Attribute Name	Attribute Type	Description
SED_CLK_FREQ	String	Specifies the clock frequency when used with SEDFA primitive.
DEV_DENSITY	String	Specifies the device density for use by the simulation model.
CHECKALWAYS	String	Reserved for future use.

The SED_CLK_FREQ attribute is used to specify the clock frequency. The SEDFA primitive uses the MachXO3L/LF internal oscillator as the clock source. The available settings are those shown in Table 6.1. If a value other than those shown in the table is used the software will issue an error message and exit from the MAP process.

The DEV_DENSITY attribute is used to specify the device density for the MachXO3L/LF simulation model. If the DEV_DENSITY attribute is not specified the default value of 1300L will be used. The allowable values for the DEV_DENSITY attribute are:

"1300L", "2100L", "4300L", "6900L", "9400L", "640L_121P", "1300L_256P", "2100L_324P", "4300L_400P"

The CHECKALWAYS attribute is not supported at this time.

8. Port Descriptions

8.1. SEDENABLE

SEDENABLE is a level-sensitive signal which enables SED checking when high. When this signal is low, the SED hardware is disabled. This can be tied high in a design if desired.

8.2. SEDSTART

SEDSTART is the signal which starts the SED process. The rising edge of the SEDSTART signal will cause the SED cycle to start if SEDENABLE is high. The SEDSTART signal must remain high until the SED process has completed. If SEDSTART goes low during the SED cycle the process will be terminated without asserting SEDDONE or SEDERR.

8.3. SEDFRCERR

SEDFRCERR is used to force the SED process to return an error indication on the SEDERR signal. This is typically done to test the logic associated with the SEDERR output. The rising edge of the SEDFRCERR signal is detected by the SED hardware and latched in by the rising edge of the SED Clock Driver signal. The SEDFRCERR should be latched high while the SED is active for an error indication to be returned. The recommended use is for the user logic to drive the SEDFRCERR signal from low to high once the rising edge of the SEDINPROG signal is detected and while following the setup/hold time requirements defined in Figure 10.1.

8.4. SEDSTDBY

The SEDSTDBY port is provided on the SEDFA primitive only and must be connected to the SEDSTDBY output port on OSCH component. This signal is provided for simulation support of the STDBY function which can be used to turn off the internal oscillator. When the STDBY function turns off the internal oscillator the SEDFA block will no longer operate because its clock source has been turned off. If the user does not connect this signal on the SEDFA primitive the SED will still function the same way in the hardware but may not match the simulation results when STDBY is used.

8.5. SEDCLKOUT

SEDCLKOUT is a gated version of the SED Clock Driver signal to the SED block. SEDCLKOUT is gated by SEDENABLE. This signal can be used to synchronize the inputs to the SED block or the outputs from the SED block.

8.6. SEDDONE

SEDDONE is an output which indicates that SED checking has completed a cycle. This signal is an active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT. SEDDONE will be reset by a low SEDSTART signal.

8.7. SEDINPROG

SEDINPROG is an output which indicates that SED checking is in progress. This signal is an active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT. SEDINPROG will go high following SEDSTART going high after the delay shown in Figure 10.1.

8.8. SEDERR

SEDERR is an output which indicates that SED checking has completed a cycle with an error. This signal is an active high output from the SED hardware, clocked out on the rising edge of SEDCLKOUT. SEDERR will be reset by a low SEDSTART signal and delay as defined in Figure 10.1.

9. SED FLOW

The general SED flow once VCC reaches the data sheet Vcc minimum recommend level is as follows.

- User logic sets SEDENABLE high. This signal may be tied high if desired.
- User logic sets SEDSTART high and holds it high for the duration of the SED cycle. SEDINPROG goes high. If SEDDONE or SEDERR are already high they will be driven low.
- SED starts reading back data from the configuration SRAM.
- SED finishes checking. SEDERR is updated, SEDINPROG goes low, SEDDONE goes high and another SED cycle is started by asserting SEDSTART. When SEDSTART is asserted the SEDERR signal will be reset.
- If SEDERR is driven high it can be reset by reconfiguring the PLD.
- SEDENABLE goes low when/if the user specifies, and SED is no longer in use.

The preferred action to take when an error is detected is to reconfigure the PLD. Reconfiguration can be accomplished by driving the PROGRAMN pin low. This can be done by externally connecting a GPIO pin to PROGRAMN.

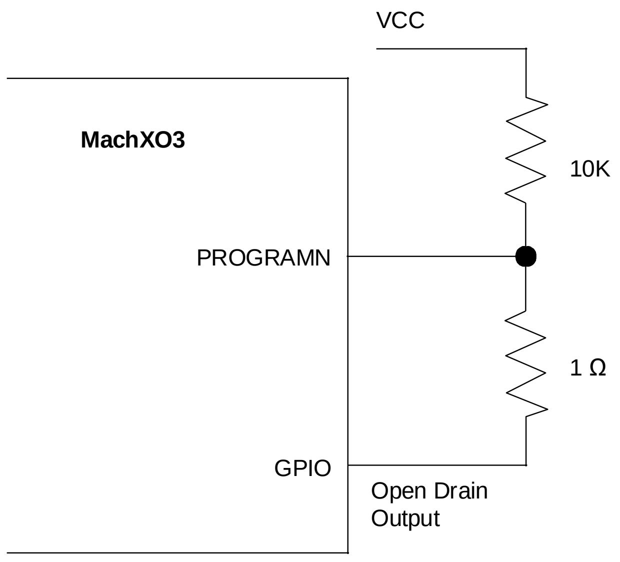


Figure 9.1. Example Schematic

Note: The 1 Ω resistor shown allows a user to recover from a bad program which always pulls the GPIO pin low in the MachXO3L/LF device. If this type of pattern is loaded into the MachXO3L/LF, the device will always be held in the re-configuration state and is not able to communicate or be erased to clear the error condition. To recover from this condition, remove the resistor and reprogram the device, then replace resistor.

10. Timing Diagram for SED Operation

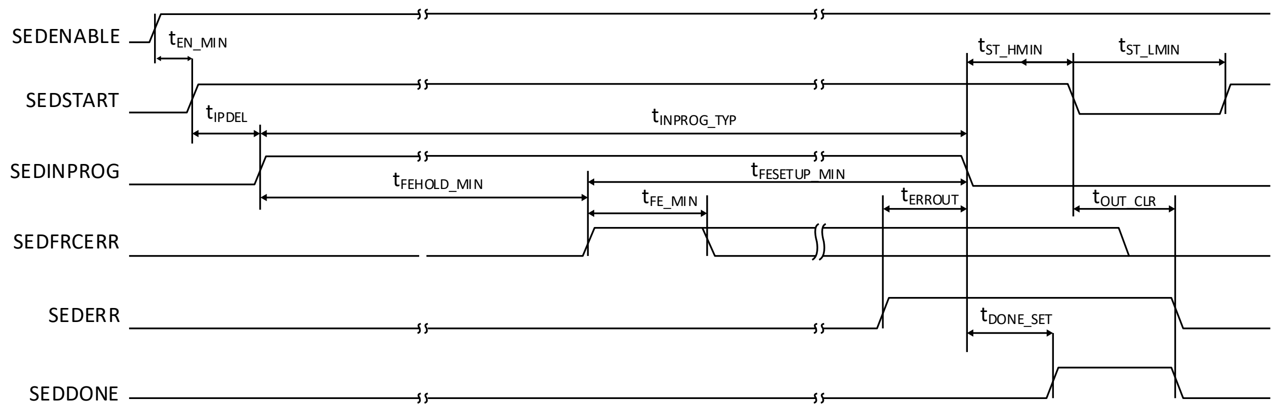


Figure 10.1. Timing Diagram for SED Operation

Where:

Parameter	**Value**	**Units**
t_{EN_MIN}	0	SEDCLK
t_{IPDEL}	2	SEDCLK
t_{FEHOLD_MIN}	92	SEDCLK
$t_{FESETUP\_MIN}$	5	SEDCLK
t_{FE_MIN}	2	SEDCLK
t_{ERROUT}	1	SEDCLK
t_{ST_HMIN}	0	SEDCLK
t_{ST_LMIN}	1	SEDCLK
t_{OUT_CLR}	2	SEDCLK
t_{DONE_SET}	0	SEDCLK

11. SED Run Time

The amount of time needed to perform an SED check depends on the density of the device and the frequency of the SED Clock Driver signal. There will also be some overhead time for calculation, but it is fairly short in comparison. An approximation of the time required can be found by using the following formula:

(Maxbits / 8) / SED Clock Driver Frequency = Time (ms)

Maxbits is in kbits and depends on the density of the PLD (see Table 11.1). The SED Clock Driver signal frequency is shown in MHz. Time is in milliseconds. The SED checking in the MachXO3L/LF device reads 8 bits (1 byte) on each SED clock cycle.

For example, for a design using a MachXO3L/LF with 4,300 look-up tables and an SED Clock Driver frequency of 7 MHz:

(972 kbits / 8) / 7.0 MHz = 17.4 ms

In this example, SED checking will take approximately 17.4 ms.

Note that the internal oscillator is used to generate the SED Clock Driver signal and its frequency can vary by ±5.5%.

Table 11.1. SED Run Time

Density*	**XO3L/LF-640 csfBGA121**	**XO3L/LF-1300**	**XO3L/LF-1300 caBGA256**	**XO3L/LF-2100**	**XO3L/LF-2000 caBGA324**	**XO3L/LF-4300**	**XO3L/LF-4300 caBGA400**	**XO3L/LF-6900**	**XO3L/LF-9400**	**Units**
	360K	**360K**	**535K**	**535K**	**972K**	**972K**	**1534K**	**1534K**	**2109K**	
33.3 MHz	1.35	1.35	2.00	2.00	3.64	3.64	5.75	5.75	7.92	ms
3.05 MHz	14.8	14.8	21.9	21.9	39.9	39.9	62.9	62.9	86.4	ms

***Note**: Density refers to the number of configuration bits in the device.

12. Sample Code

The following simple example code shows how to instantiate the SED primitive. In the example the SED is always enabled and will be run when the “sed_start” signal is driven high. The outputs of the SED hardware are available to route to the PLD output pins or for use in another module.

12.1. SED VHDL Examples

VHDL SEDFA

```
COMPONENT SEDFA
   GENERIC(
      SED_CLK_FREQ       : string    := "3.5";
      CHECKALWAYS       : string    := "DISABLED
      DEV_DENSITY        : string    := "1300L");
--"1300L", "2100L", "4300L", "6900L", "640L_121P", "1300L_256P", "2100L_324P",
"4300L_400P"

PORT(
     SEDENABLE    :     in    STD_LOGIC;
     SEDSTART     :     in    STD_LOGIC;
     SEDFRCERR    :     in    STD_LOGIC;
     SEDSTDBY     :     in    STD_LOGIC;
     SEDERR       :     out   STD_LOGIC;
     SEDDONE      :     out   STD_LOGIC;
     SEDINPROG    :     out   STD_LOGIC;
     SEDCLKOUT    :     out   STD_LOGIC);
END COMPONENT;

attribute SED_CLK_FREQ : string ;
attribute SED_CLK_FREQ of SEDinst0 : label is "13.30";
attribute CHECKALWAYS : string ;
attribute CHECKALWAYS of SEDinst0 : label is "DISABLED" ;
attribute DEV_DENSITY : string ;
attribute DEV_DENSITY of SEDinst0 : label is "1300L" ;


SEDinst0:   SEDFA
-- synthesis translate_off
      GENERIC MAP ( SED_CLK_FREQ => “13.30”;
           CHECKALWAYS => "DISABLED" ;
           DEV_DENSITY => "1300L" )

-- synthesis translate_on
PORT MAP   (SEDENABLE => '1',
         SEDSTART => sed_start,
         SEDFRCERR => '0',
         SEDSTDBY => sed_stdby,
         SEDERR => sed_err,
         SEDDONE => sed_done,
         SEDINPROG => sed_active,
         SEDCLKOUT => sed_clkout);
```

12.2. SED Verilog Examples

Verilog SEDFA

```
module SEDFA (SEDENABLE, SEDSTART, SEDFRCERR, SEDSTDBY, SEDERR, SEDDONE,
SEDINPROG, SEDCLKOUT);

input    SEDENABLE, SEDSTART, SEDFRCERR, SEDSTDBY;
output   SEDERR, SEDDONE, SEDINPROG, SEDCLKOUT;

parameter SED_CLK_FREQ = "3.5";
parameter CHECKALWAYS = "DISABLED";
parameter DEV_DENSITY = "1300L";
//"1300L","2100L","4300L","6900L","640L_121P",  "1300L_256P", "2100L-324P", and
"4300L_400P"


endmodule
```

Verilog SEDFA Primitive Instantiation

```
//  instantiate SEDFA primitive module with parameter passing to SEDFA module
SEDFA # (.SED_CLK_FREQ("4.75"), .DEV_DENSITY("1300L"))

sedfa_tst (.SEDENABLE(1'b1), .SEDSTART(sed_start), .SEDFRCERR(1'b0), .SEDSTDBY(),
          .SEDERR(sed_err),   .SEDDONE(sed_done), .SEDINPROG(sed_active),
.SEDCLK-
OUT(sed_clkout) );
```

13. MachXO3LF SEC (Soft Error Correction)

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

Soft Error Detection (SED) can be utilized in conjunction with the MachXO3LF Dual Boot feature without restriction. However, Soft Error Correction (SEC) use is limited to the primary image only. Refer to MachXO3 Programming and Configuration Usage Guide (FPGA-TN-02055) for more information on the use of the Dual Boot feature.

Upon completion of background reconfiguration, initialization of the device is bypassed, allowing the user functions to be performed during and after reconfiguration.

Soft error correction is an optional feature of the MachXO3LF device. It is used in conjunction with a sound system- wide error recovery process. When it is determined that the result of any soft error detection (SED) within the MachXO3LF user design can be contained, whether in the data or control planes, then SEC may be enabled. Refer to MachXO3 Programming and Configuration Usage Guide (FPGA-TN-02055) for more information about configuring a user design for SEC and background reconfiguration. Alternatively, if it is necessary or desired to re-initialize the user design to known state in response to a SED event, then SEC should remain disabled to allow for warm-boot recovery.

To use SEC:

1. Select or enable the BACKGROUND_RECONFIG option in Diamond Spreadsheet View when you generate the bitstream.
2. Set the synthesis option to disable distributed RAM during the synthesize process.

Once soft error is detected, either inserted or by radiation, there are multiple ways to correct it as described below.

- Provide the original bitstream in background mode via an external sysConfig slave port (for example, JTAG or I^{2}C).
- Transfer the original bitstream from internal or external Flash memory by issuing the sysConfig REFRESH command via an external sysConfig slave port.
- Transfer the original bitstream from internal or external Flash memory, initiated via external PROGRAMN pin assertion.

Two examples of how to initiate SEC are described below.

13.1. SPI Master Code

If you are using SPI Master mode to download bitstream from an external SPI flash to the MachXO3LF device, you can issue a refresh instruction or toggle the PROGRAMN pin to reload the original bitstream from the SPI flash. The DONE pin goes low during the configuration and goes back to high after the configuration.

Power cycle the device or issue offline mode instructions into the device to reinitialize the device or break the background mode.

Figure 13.1 shows the MachXO3LF Master SPI Port to SPI Flash.

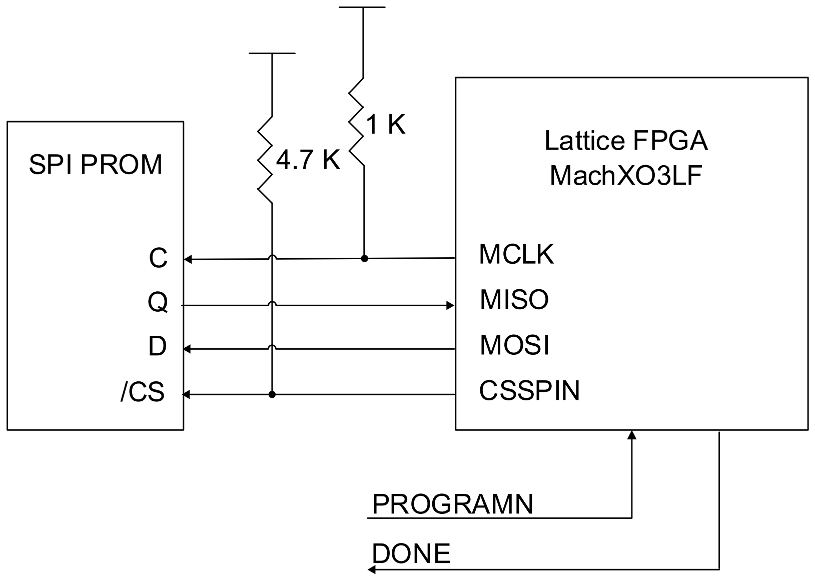


Figure 13.1. MachXO3LF Master SPI Port with SPI Flash

13.2. JTAG Mode

If the bitstream is loaded into the MachXO3LF through an external controller through the JTAG port, and you want to reprogram the device without the device function being disturbed, select the XSRAM SEI Fast Program operation in the Diamond Programmer. This creates embedded programming and loads the original bitstream into the SRAM of the device in the background.

When the MachXO3LF is in JTAG mode, reinitialize the device by reprogramming it using offline mode. For example, use the FAST PROGRAM operation.

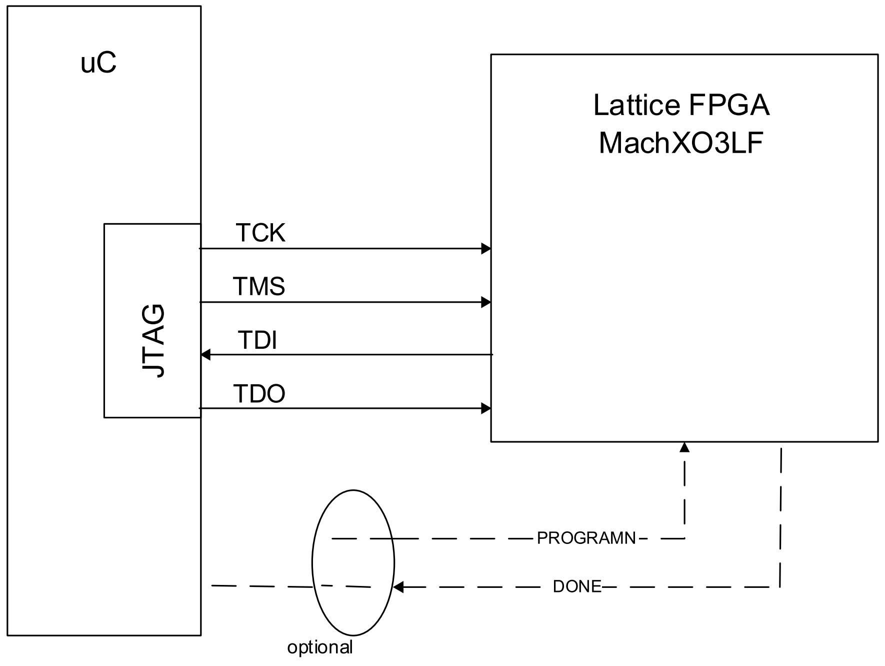


Figure 13.2. JTAG Port Connections

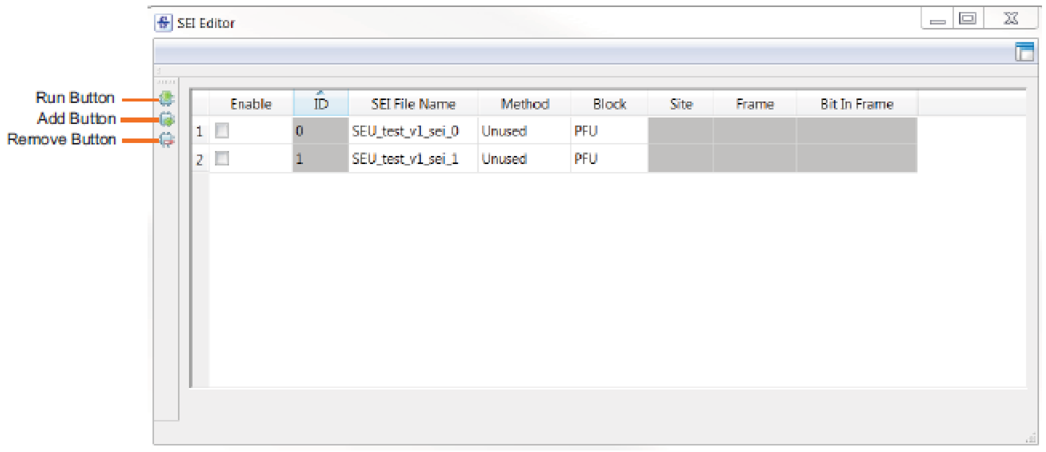
14. MachXO3LF SEI (Soft Error Injection)

The Diamond SEI tool offers an easy and economical way to emulate soft error impact to the overall system. This tool allows you to randomly generate and program one or multiple soft errors into the device in background mode without disturbing the device function.

Note: SEI is only supported when the device configuration mode is being set to slave mode.

To use the Diamond SEI tool:

1. Select or enable the BACKGROUND_RECONFIG option in Diamond spreadsheet view when you generate the bitstream.
2. Run the SEI Editor (shown in Figure 14.1) under Tools. This allows you to create one frame special bitstream that has one bit different from your original bitstream.



	Enable	ID	SEI File Name	Method	Block	Site	Frame	Bit In Frame
1	☐	0	SEU_test_v1_sei_0	Unused	PFU			
2	☐	1	SEU_test_v1_sei_1	Unused	PFU			

Figure 14.1. SEI Editor

Some of the main menus in the SEI Editor are described below:

- **Enable** – Select the checkbox of the specific bitstream to be generated when the Run button is clicked.
- **ID** – Continuous number assigned by the software.
- **File Name** – Default bitstream name. This may be changed by the user.
- **Method**
  - **Unused** – Error bit introduced is not used by customer design.
  - **Random** – Error bit introduced is random and can be used by customer design.
- **Block**
  - **Unused** – The Block can be selected among PFU, EBR or DSP.
  - **Random** – Any functional block including routing. This cannot be selected by the user.

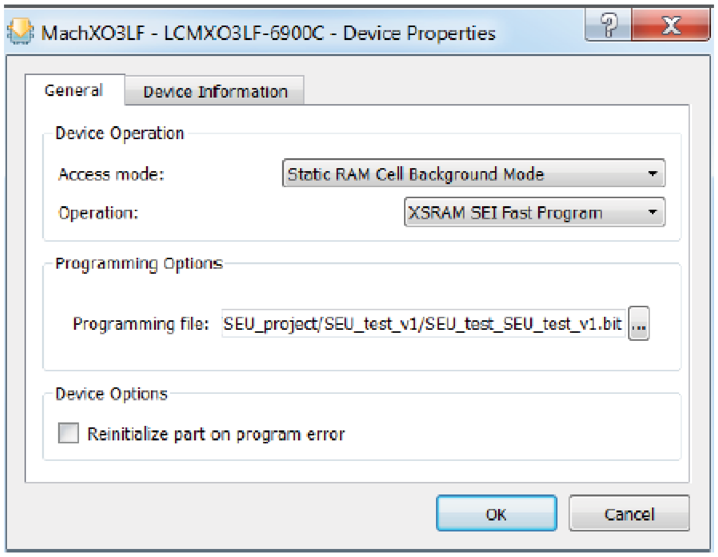
The major buttons on the SEI Editor are described below.

- **Add Button** – Click this button to add SEI files.
- **Remove Button** – Click this button to remove SEI files.
- **Run Button** – Click this button to generate SEI files.

Note: The grey area cannot be selected.

To program the one-bit-different bitstream, select the XSRAM SEI Fast Program operation in Diamond Programmer. Once it is programmed into the device, you can use the general SED routine to detect the soft error. During the SEI Fast Program, you will see the DONE pin go low during configuration and back to high after the configuration.

Note: While programming the device with soft error bitstream, SED checking will have to stop. You can deassert SEDENABLE to stop the SED checking.



Access mode: Static RAM Cell Background Mode

Operation: XSRAM SEI Fast Program

Programming file: SEU_project/SEU_test_v1/SEU_test_SEU_test_v1.bit

☐ Reinitialize part on program error

Figure 14.2. Device Properties

Technical Support Assistance

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

Revision History

Revision 1.4, January 2020

Section	Change Summary
All	• Changed document number from TN1292 to FPGA-TN-02062. • Updated document template.
Disclaimers	Added this section.

Revision 1.3, March 2017

Section	Change Summary
SED Attributes	Added 9400L to the allowable values for the DEV_DENSITY attribute.
SED Run Time	Added MachXO3L/LF-9400 values to Table 11.1. SED Run Time.
MachXO3LF SEC (Soft Error Correction)	• Added paragraph on the use of SEC and SED with MachXO3LF Dual Boot feature. • Listed ways to correct soft error. • Added lead-in sentence to introduce succeeding sections on how to initiate SEC.
JTAG Mode	Modified sentence to “This creates embedded programming and loads the original bitstream into the SRAM of the device in the background.”

Revision 1.2, April 2016

Section	Change Summary
All	Changed document title to MachXO3 Soft Error Detection (SED)/ Correction (SEC) Usage Guide.
Introduction	Added Table 1.1. SED/SEC/SEI Features.
MachXO3LF SEC (Soft Error Correction)	Added this section.
MachXO3LF SEI (Soft Error Injection)	Added this section.
Technical Support Assistance	Updated contact information.

Revision 1.1, April 2016

Section	Change Summary
All	Product name/trademark adjustment. Included MachXO3LF device.

Revision 1.0, April 2014

Section	Change Summary
All	Initial release.