MachXO2 SED Usage Guide

January 2017 **Technical Note TN1206**

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.

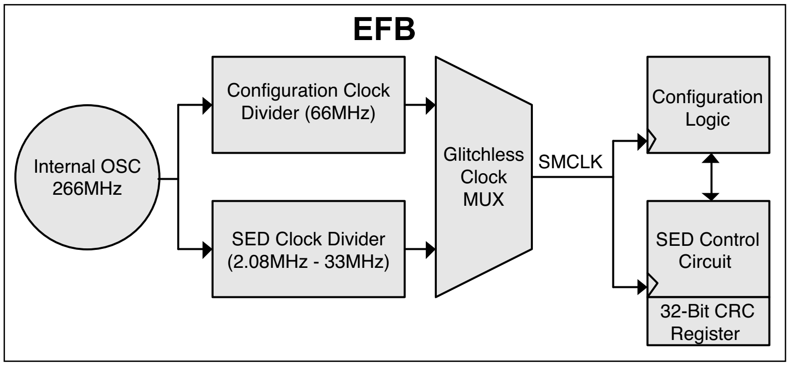
SRAM-based PLDs store logic configuration data in SRAM cells. As the number and density of SRAM cells in an PLD increase, the probability that a memory error will alter the programmed logical behavior of the system increases. A number of approaches have been taken to address this issue, but most involve Intellectual Property (IP) cores that the user instantiates into the logic of their design, using valuable resources and possibly affecting design performance. The MachXO2™ 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 SRAM CRC Error Detect (SED) approach taken by Lattice Semiconductor for MachXO2 PLDs.

SED Overview

The SED hardware in the MachXO2 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 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 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® or ispLEVER® 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 MachXO2 SED implementation and flow.

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 I^{2}C 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.

SED Operating Modes

For MachXO2 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 MachXO2 device as shown in Figure 4.

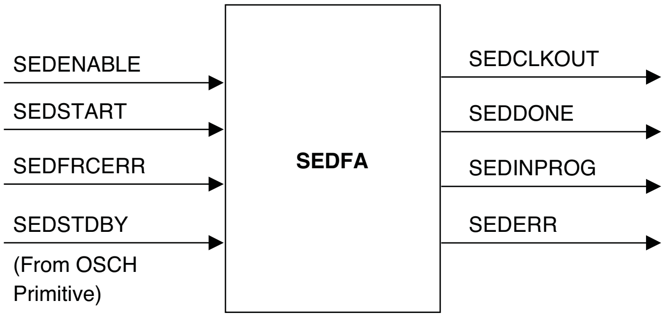
The PROGRAMn pin detection logic requires the minimal low period be longer than 6 SMCLK cycles. When error detection is actively enabled, the PROGRAMn pin minimal low period will be 6 SEDCLK cycles, since the active SMCLK switched to SEDCLK as mentioned previously. This will behave differently then normal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

Standard SED

The Standard SED operation can be used by instantiating the SEDFA primitive which is shown in Figure 2. The primitive port definitions are listed in Table 1. See the Port Descriptions section of this document for more detailed information about each of the ports.

If the Standard SED is done with the "R1" version of the MachXO2 devices, the first time the check is run a false error will be reported. If a second check is done the correct result will be reported. The "R1" versions of the MachXO2 devices have an "R1" suffix at the end of the part number, similar to LCMXO2-1200ZE-1TG144CR1. For more details on the R1 to Standard migration refer to AN8086, Designing for Migration from MachXO2-1200-R1 to Standard (Non-R1) Devices.

Figure 2. SEDFA Primitive Symbol

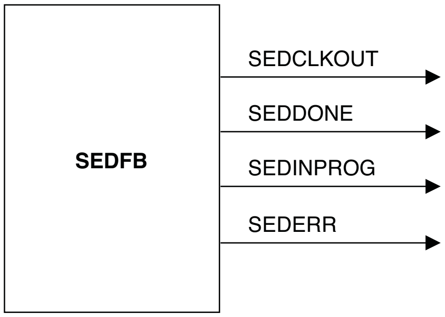


One-Shot SED

The One-Shot SED operation can be used by instantiating the SEDFB primitive which is shown in Figure 3. The definitions of the ports for the SEDFB primitive are shown in Table 1. See the Port Descriptions section of this document for more detailed information about each of the ports.

The One-Shot SED is NOT supported on the "R1" version of the MachXO2 devices because a false error will always be reported. The "R1" versions of the MachXO2 devices have an "R1" suffix at the end of the part number, similar to LCMXO2-1200ZE-1TG144CR1.

Figure 3. SEDFB Primitive Symbol



Signal Descriptions

Table 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

SED Clock Driver

The SED circuitry is driven by the MachXO2 internal oscillator when using either the SEDFA or SEDFB primitive. The maximum frequency supported is 33.25 MHz.

The MachXO2 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 2. 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	

SED Attributes

There are three attributes that can be used with the SED primitives and these are shown in Table 3. 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 3. 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 MachXO2 internal oscillator as the clock source. The available settings are those shown in Table 2. 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 MachXO2 simulation model. If the DEV_DENSITY attribute is not specified the default value of 1200L will be used. The allowable values for the DEV_DENSITY attribute are:

256L, 640L, 1200L, 2000L, 4000L, 7000L, 640U, 1200U, or 2000U

The CHECKALWAYS attribute is not supported at this time.

Port Descriptions

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.

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.

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 5.

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.

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.

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.

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 5.

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 5.

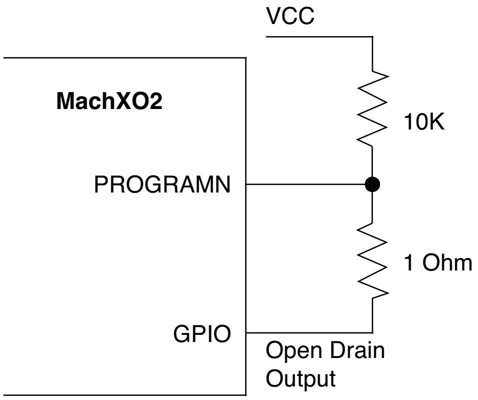
SED Flow

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

1. User logic sets SEDENABLE high. This signal may be tied high if desired.
2. 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.
3. SED starts reading back data from the configuration SRAM.
4. 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.
5. If SEDERR is driven high it can be reset by reconfiguring the PLD.
6. 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.

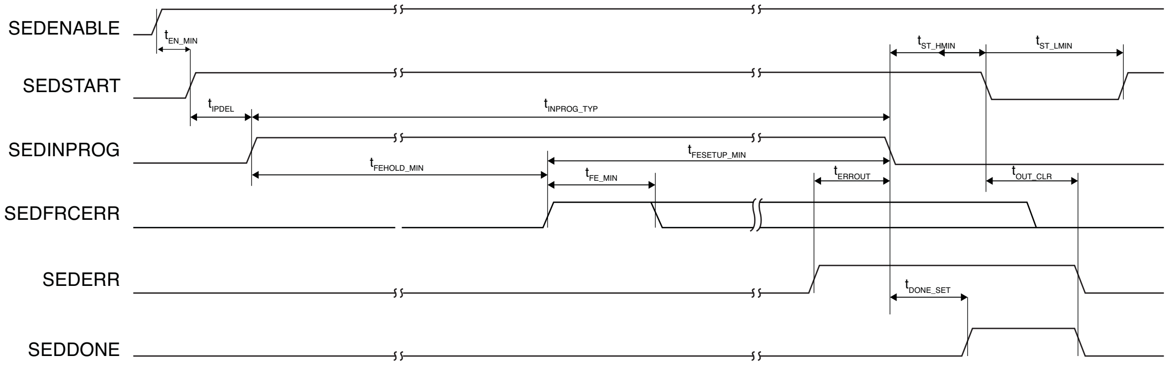
Figure 4. Example Schematic



Note: The 1 Ohm resistor shown allows a user to recover from a bad program
which always pulls the GPIO pin low in the MachXO2 device. If this type of pattern
is loaded into the MachXO2, 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.

Timing Diagram for SED Operation

Figure 5. 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

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 4). The SED Clock Driver signal frequency is shown in MHz. Time is in milliseconds. The SED checking in the MachXO2 device reads 8 bits (1 byte) on each SED clock cycle.

For example, for a design using a MachXO2 with 4,000 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 4. SED Run Time

	XO2-256	XO2-640	XO2-640U	XO2-1200	XO2-1200U	XO2-2000	XO2-2000U	XO2-4000	XO2-7000	
Density[1]	94K	191K	360K	360K	535K	535K	972K	972K	1534K	Units
33.25 MHz	0.35	0.72	1.35	1.35	2.01	2.01	3.65	3.65	5.77	ms
2.08 MHz	5.64	11.48	21.63	21.63	32.15	32.15	58.41	58.41	92.19	ms

1. Density refers to the number of configuration bits in the device.

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. Note that the SADFA primitive is part of ispLEVER 8.1 SP1 or later and Diamond 1.1 or later.

SED VHDL Examples

VHDL SEDFA

```
COMPONENT SEDFA
   GENERIC(
      SED_CLK_FREQ  : string  := "3.5";
      CHECKALWAYS  : string  := "DISABLED";
      DEV_DENSITY   : string  := "1200L");
--"256L","640L","1200L","2000L","4000L","7000L", "640U", "1200U", "2000U"

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 "1200L" ;


SEDinst0:  SEDFA
-- synthesis translate_off
      GENERIC MAP ( SED_CLK_FREQ => “13.30”;
          CHECKALWAYS => "DISABLED" ;
          DEV_DENSITY => "1200L" )
-- 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);
```

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 = "1200L";
//"256L","640L","1200L","2000L","4000L","7000L","640U", "1200U", and "2000U"


endmodule
```

Verilog SEDFA Primitive Instantiation

```
//  instantiate SEDFA primitive module with parameter passing to SEDFA module
SEDFA # (.SED_CLK_FREQ("4.75"), .DEV_DENSITY("1200L"))

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) );
```

Technical Support Assistance

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

Revision History

Date	Version	Change Summary
January 2017	2.0	Updated the SED Run Time section. Revised values in Table 4, SED Run Time.
		Updated the Technical Support Assistance section.
		Updated document template.
December 2013	01.9	Updated the body text to match Timing Diagram for SED Operation.
February 2013	01.8	Removed requirements for holding user logic in a steady state.
January 2013	01.7	Defined the EFB or Configuration Logic which interacts with the SRAM CRC Error Detection circuit.
		Added timing diagram for SED operation.
		Changed “SRAM CRC Error Detection” to “SED” throughout the document.
August 2012	01.6	Updated SRAM CRC Error Detection Limitations and SRAM CRC Error Detection Operating Modes sections.
February 2012	01.5	Updated document with new corporate logo.
July 2011	01.4	Updated Standard SRAM CRC Error Detection text section with information about migration from MachXO2-1200-R1 to Standard (N-1) devices.
June 2011	01.3	Document title changed from “MachXO2 Soft Error Detection (SED) Usage Guide” to “MachXO2 SRAM CRC Error Detection Usage Guide”.
		Clarified attribute usage for SEDFB primitive.
		Clarified run time calculation section.
May 2011	01.2	Changed “SED” to “SRAM CRC Error Detection” throughout the document.
		Added Limitations section.
		Replaced Basic SED Mode and Hardware Description sections with Operating Modes section and added description of the primitives.
		Updated supported frequencies in Table 2, SRAM CRC Error Detection Internal Oscillator Supported Frequency Settings.
		Updated Port descriptions and Flow sections.
		Corrected Run Time calculations and updated Table 4, SRAM CRC Error Detection Run Time.
		Updated Sample Code section.
January 2011	01.1	Updated for ultra-high I/O (“U”) devices.
November 2010	01.0	Initial release.