11. SEU Mitigation in Stratix IV Devices

SIV51011-3.2

This chapter describes how to use the error detection cyclical redundancy check (CRC) feature when a Stratix® IV device is in user mode and recovers from CRC errors. The purpose of the error detection CRC feature in the Stratix IV device is to detect a flip in any of the configuration random access memory (CRAM) bits in Stratix IV devices due to a soft error. With the error detection circuitry, you can continuously verify the integrity of the configuration CRAM bits.

In critical applications such as avionics, telecommunications, system control, and military applications, it is important to be able to do the following:

- Confirm that the configuration data stored in a Stratix IV device is correct
- Alert the system to the occurrence of a configuration error

The error detection feature is enhanced in the Stratix IV device family. Similar to Stratix III devices, the error detection and recovery time for single-event upset (SEU) in Stratix IV devices is reduced when compared with Stratix II devices.

For more information about test methodology for enhanced error detection in Stratix IV devices, refer to *AN 539: Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices*.

Dedicated circuitry is built into Stratix IV devices and consists of a CRC error detection feature that optionally checks for SEUs continuously and automatically.

For Stratix IV devices, the error detection CRC feature is provided in the Quartus® II software version 8.0 and onwards.

Using error detection CRC for the Stratix IV device family has no impact on fitting or performance of your device.

This chapter contains the following sections:

Error Detection Fundamentals

Error detection determines whether the data received is corrupted during transmission. To accomplish this, the transmitter uses a function to calculate a checksum value for the data and appends the checksum to the original data frame. The receiver uses the same calculation methodology to generate a checksum for the received data frame and compares the received checksum to the transmitted checksum. If the two checksum values are equal, the received data frame is correct and no data corruption occurred during transmission or storage.

The error detection CRC feature uses the same concept. When Stratix IV devices are configured successfully and are in user mode, the error detection CRC feature ensures the integrity of the configuration data.

There are two CRC error checks. One CRC error check always runs during configuration and a second optional CRC error check runs in the background in user mode. Both CRC error checks use the same CRC polynomial but different error detection implementations. For more information, refer to "Configuration Error Detection" and "User Mode Error Detection".

Configuration Error Detection

In configuration mode, a frame-based CRC is stored within the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix IV device calculates the CRC value based on the frame of data that is received and compares it against the frame CRC value in the data stream. Configuration continues until either the device detects an error or configuration is completed.

In Stratix IV devices, the CRC value is calculated during the configuration stage. A parallel CRC engine generates 16 CRC check bits per frame and then stores them in CRAM. The CRAM chain used for storing the CRC check bits is 16 bits wide and its length is equal to the number of frames in the device.

User Mode Error Detection

Stratix IV devices have built-in error detection circuitry to detect data corruption by soft errors in the CRAM cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed CRC value. Soft errors are changes in a CRAM bit state due to an ionizing particle.

The error detection capability continuously computes the CRC of the configured CRAM bits and compares it with the pre-calculated CRC. If the CRCs match, there is no error in the current configuration CRAM bits. The process of error detection continues until the device is reset (by setting `nCONFIG` low).

If you enable the **CRC error detection** option in the Quartus II software, after the device transitions into user mode, the error detection process is enabled. The internal 100 MHz configuration oscillator is divided down by a factor of two to 256 (at powers of two) to be used as the clock source during the error detection process. You must set the clock divide factor in the Quartus II software.

A single 16-bit CRC calculation is done on a per-frame basis. After it has finished the CRC calculation for a frame, the resulting 16-bit signature is `hex 0000` if there are no CRAM bit errors detected in a frame by the error detection circuitry and the output signal `CRC_ERROR` is `0`. If a CRAM bit error is detected by the circuitry within a frame in the device, the resulting signature is non-zero. This causes the CRC engine to start searching for the error bit location.

Error detection in Stratix IV devices calculates CRC check bits for each frame and pulls the `CRC_ERROR` pin high when it detects bit errors in the chip. Within a frame, it can detect all single-bit, double-bit, and three-bit errors. The probability of more than three CRAM bits being flipped by an SEU event is very low. In general, for all error patterns the probability of detection is 99.998%.

The CRC engine reports the bit location and determines the type of error for all single-bit errors and over 99.641% of double-adjacent errors. The probability of other error patterns is very low and report of the location of bit flips is not guaranteed by the CRC engine.

You can also read-out the error bit location through the JTAG and the core interface. Shift these bits out through either the `SHIFT_EDERROR_REG` JTAG instruction or the core interface before the CRC detects the next error in another frame. If the next frame also has an error, you must shift these bits out within the amount of time of one frame CRC verification. You can choose to extend this time interval by slowing down the error detection clock frequency, but this slows down the error recovery time for the SEU event. For the minimum update interval for Stratix IV devices, refer to Table 11–6 on page 11–9. If these bits are not shifted out before the next error location is found, the previous error location and error message is overwritten by the new information. The CRC circuit continues to run, and if an error is detected, you must decide whether to complete a reconfiguration or to ignore the CRC error.

The error detection logic continues to calculate the `CRC_ERROR` and 16-bit signatures for the next frame of data regardless if any error has occurred in the current frame or not. You need to monitor these signals and take the appropriate actions if a soft error occurs.

The error detection circuitry in Stratix IV devices uses a 16-bit CRC-ANSI standard (16-bit polynomial) as the CRC generator.

The computed 16-bit CRC signature for each frame is stored in the registers within the core. The total storage register size is 16 (the number of bits per frame) × the number of frames.

The Stratix IV device error detection feature does not check memory blocks and I/O buffers. Thus, the `CRC_ERROR` signal might stay solid high or low depending on the error status of the previously checked CRAM frame. The I/O buffers are not verified during error detection because these bits use flipflops as storage elements that are more resistant to soft errors when compared with CRAM cells. The support parity bits of MLAB, M9K, and M144K are used to check the contents of the memory blocks for any errors. The M144K TriMatrix memory block has a built-in error correction code block that checks and corrects the errors in the block.

For more information, refer to the *TriMatrix Embedded Memory Blocks in Stratix IV Devices* chapter.

A JTAG instruction, EDERROR_INJECT, is provided to test the capability of the error detection block. This instruction is able to change the content of the 21-bit JTAG fault injection register that is used for error injection in Stratix IV devices, enabling the testing of the error detection block.

You can only execute the EDERROR_INJECT JTAG instruction when the device is in user mode.

Table 11–1 lists the description of the EDERROR_INJECT JTAG instruction.

Table 11–1. EDERROR_INJECT JTAG Instruction

JTAG Instruction	Instruction Code	Description
EDERROR_INJECT	00 0001 0101	This instruction controls the 21-bit JTAG fault injection register, which is used for error injection.

You can create a Jam™ file (**.jam**) to automate the testing and verification process. This allows you to verify the CRC functionality in-system, on-the-fly, without having to reconfigure the device. You can then switch to the CRC circuit to check for real errors induced by an SEU.

You can introduce a single-error or double-errors adjacent to each other to the configuration memory. This provides an extra way to facilitate design verification and system fault tolerance characterization. Use the JTAG fault injection register with the EDERROR_INJECT instruction to flip the readback bits. The Stratix IV device is then forced into error test mode.

The content of the JTAG fault injection register is not loaded into the fault injection register during the processing of the last and first frame. It is only loaded at the end of this period.

You can only introduce error injection in the first data frame, but you can monitor the error information at any time. For more information about the JTAG fault injection register and fault injection register, refer to "Error Detection Registers" on page 11–7.

Table 11–2 lists how the fault injection register is implemented and describes error injection.

Table 11–2. Fault Injection Register

Bit	Bit[20..19]			Bit[18..8]	Bit[7..0]
Description	Error Type			Byte Location of the Injected Error	Error Byte Value
Content	Error Type *(1)*		Error injection type	Depicts the location of the injected error in the first data frame.	Depicts the location of the bit error and corresponds to the error injection type selection.
	Bit[20]	Bit[19]			
	0	1	Single-byte error injection		
	1	0	Double-adjacent byte error injection		
	0	0	No error injection		

Note to Table 11–2:

(1) Bit[20] and Bit[19] cannot both be set to 1 as this is not a valid selection. The error detection circuitry decodes this as no error injection.

After the test completes, Altera recommends reconfiguring the device.

Automated Single-Event Upset Detection

Stratix IV devices offer on-chip circuitry for automated checking of SEU detection. Some applications that require the device to operate error-free in high-neutron flux environments require periodic checks to ensure continued data integrity. The error detection CRC feature ensures data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Stratix IV devices, eliminating the need for external logic. The CRC_ERROR pin reports a soft error when the configuration CRAM data is corrupted. You must decide whether to reconfigure the device or to ignore the error.

Error Detection Pin Description

Depending on the type of error detection feature you choose, you must use different error detection pins to monitor the data during user mode.

CRC_ERROR Pin

Table 11–3 describes the CRC_ERROR pin.

Table 11–3. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O and open-drain	Active-high signal indicates that the error detection circuit has detected errors in the configuration CRAM bits. This pin is optional and is used when the error detection CRC circuit is enabled. When the error detection CRC circuit is disabled, it is a user I/O pin. To use the CRC_ERROR pin, you can either tie this pin to V_{CCPGM} through a 10k Ω resistor or, depending on the input voltage specification of the system receiving the signal, you can tie this pin to a different pull-up voltage.

The WYSIWYG function performs optimization on the Verilog Quartus Mapping (VQM) netlist within the Quartus II software.

For more information about the stratixiv_crcblock WYSIWYG function, refer to the *AN 539: Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices*.

For more information about the CRC_ERROR pin for Stratix IV devices, refer to *Device Pin-Outs* on the Altera website.

Error Detection Block

You can enable the Stratix IV device error detection block in the Quartus II software (refer to "Software Support" on page 11–10). This block contains the logic necessary to calculate the 16-bit CRC signature for the configuration CRAM bits in the device.

The CRC circuit continues running even if an error occurs. When a soft error occurs, the device sets the CRC_ERROR pin high. Two types of CRC detection checks the configuration bits:

- CRAM error checking ability (16-bit CRC), which occurs during user mode to be used by the CRC_ERROR pin.
  - For each frame of data, the pre-calculated 16-bit CRC enters the CRC circuit at the end of the frame data and determines whether there is an error or not.
  - If an error occurs, the search engine starts to find the location of the error.
  - The error messages are shifted out through the JTAG instruction or core interface logics while the error detection block continues running.
  - The JTAG interface reads out the 16-bit CRC result for the first frame and also shifts the 16-bit CRC bits to the 16-bit CRC storage registers for test purposes.
  - Single error, double errors, or double-errors adjacent to each other are deliberately introduced to configuration memory for testing and design verification.
- 16-bit CRC that is embedded in every configuration data frame.
  - During configuration, after a frame of data is loaded into the Stratix IV device, the pre-computed CRC is shifted into the CRC circuitry.
  - At the same time, the CRC value for the data frame shifted-in is calculated. If the pre-computed CRC and calculated CRC values do not match, nSTATUS is set low. Every data frame has a 16-bit CRC; therefore, there are many 16-bit CRC values for the whole configuration bitstream. Every device has different lengths of configuration data frame.

The "Error Detection Block" section describes the 16-bit CRC only when the device is in user mode.

Error Detection Registers

There is one set of 16-bit registers in the error detection circuitry that stores the computed CRC signature. A non-zero value on the syndrome register causes the CRC_ERROR pin to be set high.

Figure 11–1 shows the error detection circuitry, syndrome registers, and error injection block.

Figure 11–1. Error Detection Block Diagram

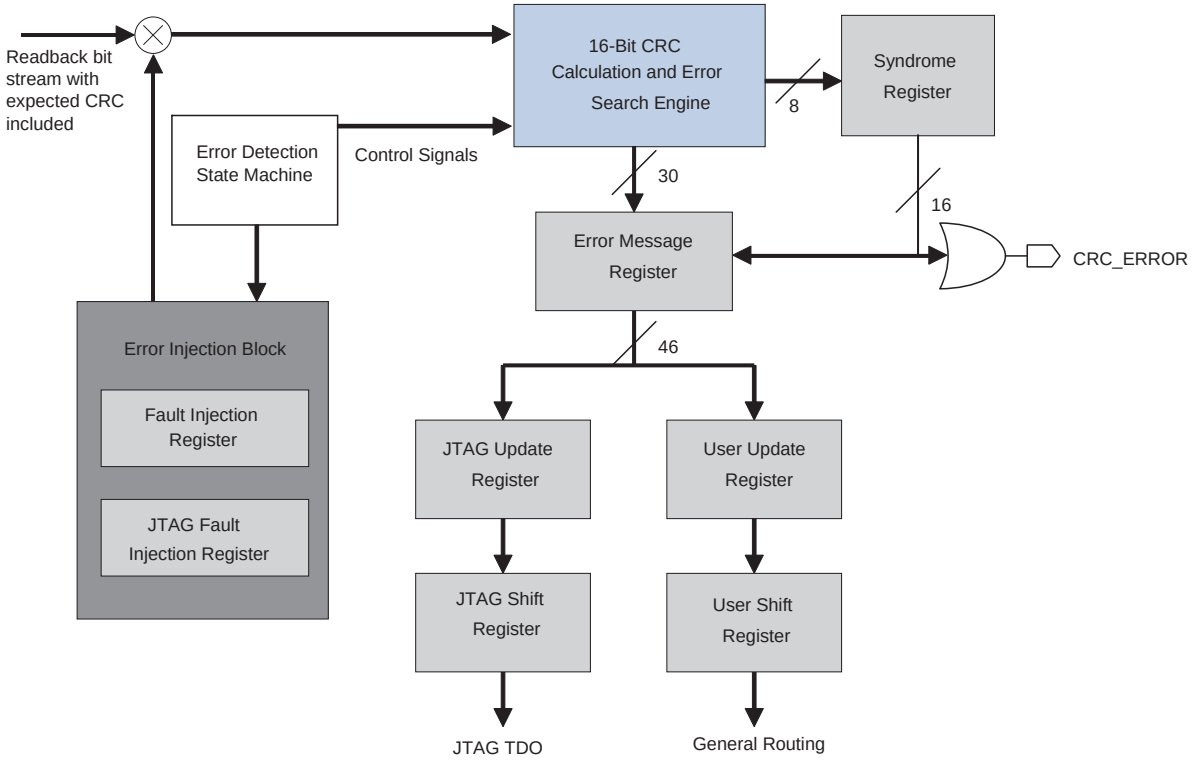


Table 11–4 lists the registers shown in Figure 11–1.

Table 11–4. Error Detection Registers (Part 1 of 2)

Register	Description
Syndrome Register	This register contains the CRC signature of the current frame through the error detection verification cycle. The CRC_ERROR signal is derived from the contents of this register.
Error Message Register	This 46-bit register contains information on the error type, location of the error, and the actual syndrome. The types of errors and location reported are single- and double-adjacent bit errors. The location bits for other types of errors are not identified by the error message register. The content of the register can be shifted out through the SHIFT_EDERROR_REG JTAG instruction or to the core through the core interface.

Table 11–4. Error Detection Registers (Part 2 of 2)

Register	Description
JTAG Update Register	This register is automatically updated with the contents of the error message register one cycle after the 46-bit register content is validated. It includes a clock enable that must be asserted prior to being sampled into the JTAG shift register. This requirement ensures that the JTAG update register is not being written into by the contents of the error message register at the same time that the JTAG shift register is reading its contents.
User Update Register	This register is automatically updated with the contents of the Error Message Register, one cycle after the 46-bit register content is validated. It includes a clock enable that must be asserted prior to being sampled into the User Shift Register. This requirement ensures that the User Update Register is not being written into by the contents of the Error Message Register at exactly the same time that the User Shift Register is reading its contents.
JTAG Shift Register	This register is accessible by the JTAG interface and allows the contents of the JTAG Update Register to be sampled and read by the JTAG instruction SHIFT_EDERROR_REG.
User Shift Register	This register is accessible by the core logic and allows the contents of the User Update Register to be sampled and read by user logic.
JTAG Fault Injection Register	This 21-bit register is fully controlled by the JTAG instruction EDERROR_INJECT. This register holds the information of the error injection that you want in the bitstream.
Fault Injection Register	The content of the JTAG Fault Injection Register is loaded into this 21-bit register when it is being updated.

Error Detection Timing

When you enable the CRC feature through the Quartus II software, the device automatically activates the CRC process after entering user mode, after configuration, and after initialization is complete.

If an error is detected within a frame, CRC_ERROR is driven high at the end of the error location search, after the error message register is updated. At the end of this cycle, the CRC_ERROR pin is pulled low for a minimum of 32 clock cycles. If the next frame contains an error, CRC_ERROR is driven high again after the error message register is overwritten by the new value. You can start to unload the error message on each rising edge of the CRC_ERROR pin. Error detection runs until the device is reset.

The error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency. Table 11–5 lists the minimum and maximum error detection frequencies based on the best performance of the internal configuration oscillator.

Table 11–5. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (n)
Stratix IV	100 MHz / 2^n	50 MHz	390 kHz	1, 2, 3, 4, 5, 6, 7, 8

You can set a lower clock frequency by specifying a division factor in the Quartus II software (refer to "Software Support" on page 11–10). The divisor is a power of two, in which *n* is between 1 and 8. The divisor ranges from 2 through 256. Refer to Equation 11–1.

Equation 11–1.

$$\text{error detection frequency} = \frac{100\ \text{MHz}}{2^n}$$

The error detection frequency reflects the frequency of the error detection process for a frame because the CRC calculation in the Stratix IV device is done on a per-frame basis.

You must monitor the error message to avoid missing information in the error message register. The error message register is updated whenever an error occurs. The minimum interval time between each update for the error message register depends on the device and the error detection clock frequency.

Table 11–6 lists the estimated minimum interval time between each update for the error message register for Stratix IV devices.

Table 11–6. Minimum Update Interval for Error Message Register ***(Note 1)***

Device	Timing Interval (µs)
EP4SGX70	13.8
EP4SGX110	13.8
EP4SGX180	19.8
EP4SGX230	19.8
EP4SGX290	21.8
EP4SGX360	21.8
EP4SGX530	26.8
EP4SE230	19.8
EP4SE360	21.8
EP4SE530	26.8
EP4SE820	33.8
EP4S40G2	19.8
EP4S40G5	26.8
EP4S100G2	19.8
EP4S100G3	26.8
EP4S100G4	26.8
EP4S100G5	26.8

Note to Table 11–6:

(1) These timing numbers are preliminary.

CRC calculation time for the error detection circuitry to check from the first until the last frame depends on the device and the error detection clock frequency.

Table 11–7 lists the estimated time for each CRC calculation with minimum and maximum clock frequencies for Stratix IV devices. The minimum CRC calculation time is calculated by using the maximum error detection frequency with a divisor factor of one, and the maximum CRC calculation time is calculated by using the minimum error detection frequency with a divisor factor of eight.

Table 11–7. CRC Calculation Time *(Note 1)*

Device	Minimum Time (ms)	Maximum Time (s)
EP4SGX70	111	30.90
EP4SGX110	111	30.90
EP4SGX180	225	62.44
EP4SGX230	225	62.44
EP4SGX290	296	82.05
EP4SGX360	296	82.05
EP4SGX530	398	110.38
EP4SE230	225	62.44
EP4SE360	296	82.05
EP4SE530	398	110.38
EP4SE820	577	160.00
EP4S40G2	225	62.44
EP4S40G5	398	110.38
EP4S100G2	225	62.44
EP4S100G3	398	110.38
EP4S100G4	398	110.38
EP4S100G5	398	110.38

Note to Table 11–7:

(1) These timing numbers are preliminary.

Software Support

The Quartus II software version 8.0 and onwards supports the error detection CRC feature for Stratix IV devices. Enabling this feature generates the CRC_ERROR output to the optional dual purpose CRC_ERROR pin.

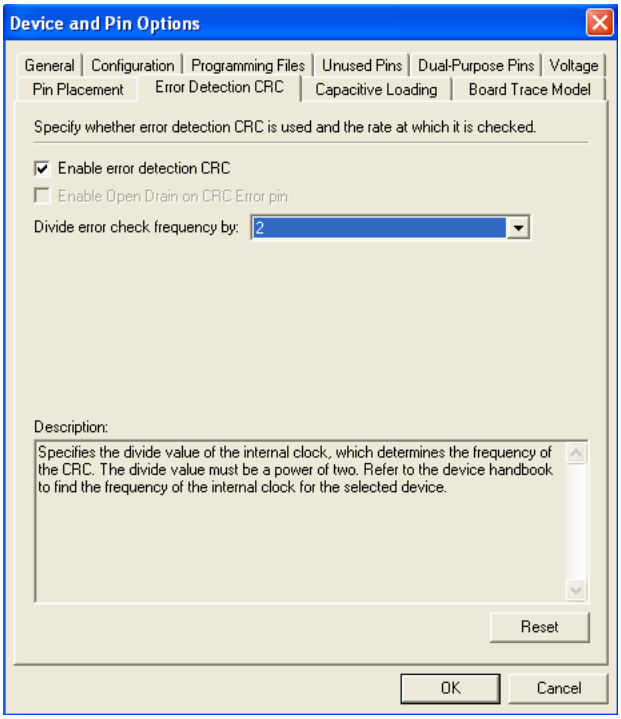
The error detection CRC feature is controlled by the **Device and Pin Options** dialog box in the Quartus II software.

To enable the error detection feature using CRC, follow these steps:

1. Open the Quartus II software and load a project using a Stratix IV device.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box is shown.
3. In the **Category** list, select **Device**. The **Device** page is shown.
4. Click **Device and Pin Options**. The **Device and Pin Options** dialog box is shown (refer to Figure 11–2).

5. In the **Device and Pin Options** dialog box, click the **Error Detection CRC** tab.
6. Turn on **Enable error detection CRC** (Figure 11–2).

Figure 11–2. Enabling the Error Detection CRC Feature in the Quartus II Software



7. In the **Divide error check frequency by** pull-down list, enter a valid divisor as listed in Table 11–5 on page 11–8.

The divide value divides the frequency of the configuration oscillator output clock that clocks the CRC circuitry.

8. Click **OK**.

Recovering From CRC Errors

The system that the Stratix IV device resides in must control device reconfiguration. After detecting an error on the CRC_ERROR pin, strobing the nCONFIG signal low directs the system to perform the reconfiguration at a time when it is safe for the system to reconfigure the device.

When the data bit is rewritten with the correct value by reconfiguring the device, the device functions correctly.

While soft errors are uncommon in Altera devices, certain high-reliability applications require a design to account for these errors.

Document Revision History

Table 11–8 lists the revision history for this chapter.

Table 11–8. Document Revision History

Date	Version	Changes
February 2011	3.2	■ Applied new template. ■ Minor Text edits.
March 2010	3.1	■ Updated Table 11–3 and Table 11–6. ■ Minor text edits.
November 2009	3.0	■ Updated Table 11–3, Table 11–5, Table 11–6, and Table 11–7. ■ Updated the "CRC_ERROR Pin" section. ■ Minor text edits.
June 2009	2.3	■ Added an introductory paragraph to increase search ability. ■ Removed the Conclusion section. ■ Minor text edits.
April 2009	2.2	■ Updated Table 11–6 and Table 11–7.
March 2009	2.1	■ Updated "Error Detection Timing" section. ■ Updated Table 11–6. ■ Added Table 11–7. ■ Removed "Critical Error Detection", "Critical Error Pin", and "Referenced Documents" sections.
November 2008	2.0	Minor text edits.
May 2008	1.0	Initial release.