

This chapter describes how to activate and use the error detection cyclic redundancy check (CRC) feature when your Stratix[®] V device is in user mode and how to recover from configuration errors caused by CRC errors. The error detection feature is enhanced in the Stratix V devices.

This chapter contains the following sections:

- “Error Detection Fundamentals” on page 10–1
- “Configuration Error Detection” on page 10–2
- “User Mode Error Detection and Correction” on page 10–2
- “Error Detection Pin Description” on page 10–5
- “Error Detection Block” on page 10–6
- “Error Detection Timing” on page 10–8
- “Software Support” on page 10–11
- “Recovering From CRC Errors” on page 10–11

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 V device is correct.
- Alert the system to the occurrence of a configuration error.



For Stratix V devices, the error detection CRC feature is provided in the Quartus[®] II software starting with version 10.0.

Using the error detection CRC feature for the Stratix V device family has no impact on fitting or performance.

Error Detection Fundamentals

Error detection determines if the data received through a medium 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 function to calculate 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 V devices are successfully configured and in user mode, the error detection CRC feature ensures the integrity of the configuration data.

Configuration Error Detection

In configuration mode, a frame-based 16-bit configuration CRC is stored in the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix V device calculates the 16-bit configuration CRC value based on the frame of data that is received and compares it against the pre-calculated 16-bit configuration CRC value in the data stream. If the 16-bit configuration CRC values do not match, `nSTATUS` is set low. Configuration continues until either the device detects an error or configuration is complete.

User Mode Error Detection and Correction

Stratix V devices offer on-chip circuitry for automated single event upset (SEU) detection. Some applications 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 V devices, eliminating the need for external logic. Stratix V devices have built-in error detection circuitry to detect data corruption by soft errors in the configuration random access memory (CRAM) cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed 32-bit error detection CRC value. Soft errors are changes in a CRAM's bit state due to an ionizing particle.

To enable the error detection process when the device transitions into user mode, turn on the **Enable Error Detection CRC_ERROR pin** option on the **Error Detection CRC** page of the **Device and Pin Options** dialog box in the Quartus II software.

The error detection capability continuously calculates the 32-bit error detection CRC value of the configured CRAM bits and compares it with the configuration-computed 32-bit error detection CRC value. The 32-bit error detection CRC value is computed during the configuration stage. The error detection circuitry generates 32 CRC check bits per frame and then stores them in the CRAM. If the 32-bit error detection CRC values 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.

A single 32-bit error detection CRC calculation is done on a per frame basis. After the error detection circuitry has finished the CRC calculation for a frame, the resulting 32-bit signature is `0x00000000`. If the error detection circuitry detects no CRAM bit errors in a frame, the output signal `CRC_ERROR` is set to low. If the circuitry detects a CRAM bit error in a frame in the device, the resulting signature is non-zero and the error detection circuitry starts searching for the error bit location.

The error detection circuitry in Stratix V 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, triple-bit, quadruple-bit, and quintuple-bit errors. The probability of more than five CRAM bits being flipped by a SEU is very low. In general, the probability of detection for all error patterns is 99.9999%.

The error detection circuitry reports the bit location and determines the type of error for single-bit errors or double-adjacent errors. The probability of other error patterns is very low and the reporting of bit location is not guaranteed.

You can also read the error bit location through the JTAG and the core interface. Before the error detection circuitry detects the next error in another frame, you must shift erroneous bits out from the error message register (EMR) with either the JTAG instruction, `SHIFT_EDERROR_REG`, or the core interface. The CRC circuitry continues to run, and if an error is detected, you must decide whether to complete the reconfiguration or to ignore the CRC error.

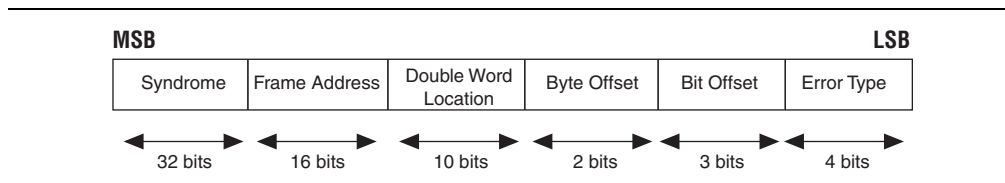
Table 10-1 lists the instruction code for the `SHIFT_EDERROR_REG` JTAG instruction.

Table 10-1. SHIFT_EDERROR_REG JTAG Instruction

JTAG Instruction	Instruction Code	Description
<code>SHIFT_EDERROR_REG</code>	00 0001 0111	The JTAG instruction connects the EMR to the JTAG pin in the error detection block between the TDI and TDO pins.

Figure 10-1 shows the content of the EMR.

Figure 10-1. Error Message Register



The type of error is identified in the first four bits of the EMR. Table 10-2 lists the error types represented in the EMR.

Table 10-2. Error Type in Error Message Register

Error Type				Description
Bit 3	Bit 2	Bit 1	Bit 0	
0	0	0	0	No CRC error.
0	0	0	1	Location of a single-bit error is identified.
0	0	1	0	Location of a double-adjacent error is identified.
1	1	1	1	There is more than one error.
Others				Reserved.



For more information about the timing requirement to shift out error information from the EMR, refer to “Error Detection Timing” on page 10-8.

The error detection circuitry continues to calculate the 32-bit error detection CRC value and 32-bit signatures for the next frame of data regardless of whether an error has occurred in the current frame or not. You must monitor the `CRC_ERROR` signal and take the appropriate actions if a CRC error occurs.

The error detection circuitry in Stratix V devices uses a 32-bit CRC-ANSI standard (32-bit polynomial) as the CRC generator. The computed 32-bit CRC signature for each frame is stored in the CRAM. The total storage size is 32 (number of bits per frame) × the number of frames.

The Stratix V device error detection CRC feature does not check memory blocks and I/O buffers. Thus, the `CRC_ERROR` signal may 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. MLAB and M20K memory blocks support parity bits that are used to check the contents of memory blocks for any error.



For more information about error detection in Stratix V memory blocks, refer to the *TriMatrix Embedded Memory Blocks in Stratix V Devices* chapter.

In Stratix V, in addition to the error detection capability, the error detection circuitry also supports error correction or internal scrubbing, which is an ability to internally correct soft errors that have been detected. This is done on a per-frame basis. If internal scrubbing is enabled, the device corrects single-bit error or double-adjacent error in the CRAM bits while the frame where the error occurred is still operational and the device is still running.

To provide testing capability of the error detection block, a JTAG instruction, `EDERROR_INJECT`, is provided. This instruction is able to change the content of the 47-bit JTAG fault injection register used for error injection in Stratix V devices, thereby enabling testing of the error detection block.



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

Table 10-3 lists the `EDERROR_INJECT` JTAG instruction.

Table 10-3. EDERROR_INJECT JTAG Instruction

JTAG Instruction	Instruction Code	Description
<code>EDERROR_INJECT</code>	00 0001 0101	This instruction controls the 47-bit JTAG fault injection register 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 and on-the-fly, without having to reconfigure the device. You can then switch to the CRC circuit to check for real errors induced by a 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 JTAG instruction to flip the readback bits. The Stratix V device is then forced into error test mode. Altera recommends reconfiguring the device after the test completes.


 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 10-7.

Table 10-4 lists how the fault injection register is implemented and describes error injection.

Table 10-4. Fault Injection Register

Description	Bit[46..43]				Error Injection Type	Bit[42..32]	Bit[31..0]
	Error Type					Byte Location of the Injected Error	Error Byte Value
	Bit[46]	Bit[45]	Bit[44]	Bit[43]			
Content	0	0	0	0	No error injection	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.
	0	0	0	1	Single error injection		
	0	0	1	0	Double-adjacent error injection		
	Others				Reserved		

Error Detection Pin Description

Table 10-5 lists the CRC_ERROR pin.

Table 10-5. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O, output, or output open-drain	<p>Active high signal indicating that the error detection circuit has detected errors in the configuration CRAM bits. This is an optional pin and is used when you enable the error detection CRC circuit. When you disable the error detection CRC circuit, it is an user I/O pin. When using the WYSIWYG function, you can route the <code>crcerror</code> port from the WYSIWYG atom to the dedicated CRC_ERROR pin or any user I/O. To route the <code>crcerror</code> port to user I/O, you must insert a D-type flipflop in between the <code>crcerror</code> port and the I/O.</p> <p>By default, the Quartus II software sets the CRC_ERROR pin as output open-drain when you enable the error detection CRC circuitry. By option, you can set this pin to be an output by turning off the Enable open-drain on CRC_ERROR pin option on the Error Detection CRC page of the Device and Pin Option dialog box in the Quartus II software. If the CRC_ERROR pin is used as an output, you must ensure that the V_{CCIO} of the bank in which the pin resides meet the input voltage specification of the system receiving the signal. Using the pin as an open-drain provides an advantage on the voltage leveling. To use the CRC_ERROR pin as an open-drain, you can tie this pin to V_{CCPGM} through a 10-kΩ resistor. Alternatively, depending on the input voltage specification of the system receiving the signal, tie this pin to a different pull-up voltage.</p>

Error Detection Block

The error detection block contains the logic necessary to calculate the 32-bit error detection CRC signature for the configuration CRAM bits in the Stratix V device.

The CRC circuit continues running even if an error occurs. When a CRC error occurs, the device sets the `CRC_ERROR` pin high. Table 10-6 lists the two types of CRC detection that check the configuration bits.

Table 10-6. Two Types of CRC Detection

User Mode CRC Error Detection	Configuration CRC Error Detection
<ul style="list-style-type: none"> ■ This is the CRAM error checking ability (32-bit error detection CRC) during user mode for use by the <code>CRC_ERROR</code> pin. ■ For each frame of data, the pre-calculated 32-bit error detection 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 finds the location of the error. ■ The error messages can be shifted out through the JTAG instruction or core interface logics while the error detection block continues running. ■ The JTAG interface reads out the 32-bit error detection CRC result for the first frame and also shifts the 32-bit error detection CRC bits to the 32-bit error detection CRC storage registers for test purposes. ■ You can deliberately introduce single error or double-adjacent error to the configuration memory for testing and design verification. 	<ul style="list-style-type: none"> ■ This is the 16-bit configuration CRC that is embedded in every configuration data frame. ■ During configuration, after a frame of data is loaded into the Stratix V device, the pre-computed configuration CRC is shifted into the CRC circuitry. ■ At the same time, the 16-bit configuration CRC value for the data frame shifted-in is calculated. If the pre-computed configuration CRC and calculated configuration CRC values do not match, <code>STATUS</code> is set low. Every data frame has a 16-bit configuration CRC; therefore, there are many 16-bit configuration CRC values for the whole configuration bitstream as there are many data frames. Every device has different lengths of the configuration data frame.

 The “Error Detection Registers” section focuses on the user mode CRC error detection.

Error Detection Registers

There is one set of 32-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 10-2 shows the error detection circuitry, syndrome registers, and error injection block.

Figure 10-2. Error Detection Circuitry, Syndrome Registers, and Error Injection Block

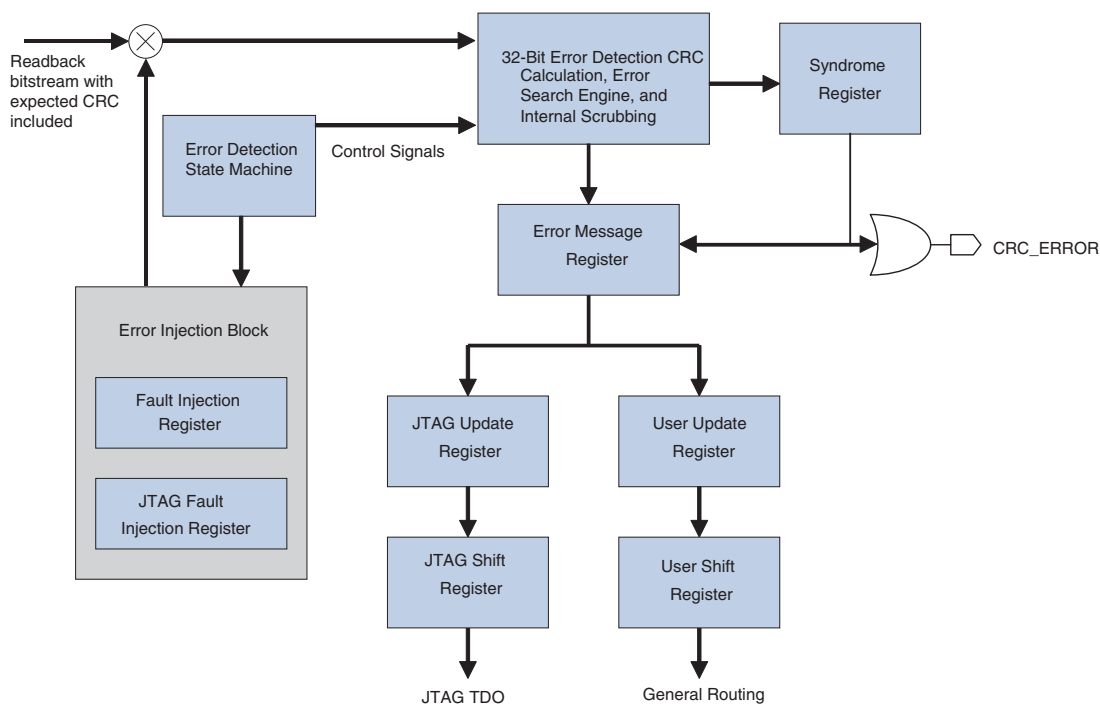


Table 10-7 lists the registers shown in Figure 10-2.

Table 10-7. Error Detection Registers (Part 1 of 2)

Register	Description
Syndrome Register	This 32-bit 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 67-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 EMR. The content of the register is shifted out through the SHIFT_EDERROR_REG JTAG instruction or to the core through the core interface.
JTAG Update Register	This 67-bit register is automatically updated with the contents of the EMR one cycle after this register content is validated. It includes a clock enable, which 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 EMR at the same time that the JTAG shift register is reading its contents.

Table 10-7. Error Detection Registers (Part 2 of 2)

Register	Description
User Update Register	This 67-bit register is automatically updated with the contents of the EMR one cycle after this register content is validated. It includes a clock enable, which 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 EMR at exactly the same time that the user shift register is reading its contents.
JTAG Shift Register	This 67-bit register is accessible by the JTAG interface and allows the contents of the JTAG update register to be sampled and read out by <code>SHIFT_EDERROR_REG</code> JTAG instruction.
User Shift Register	This 67-bit 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 47-bit register is fully controlled by the <code>EDERROR_INJECT</code> JTAG instruction. 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 47-bit register when it is updated.

Error Detection Timing

When you enable the error detection CRC feature through the Quartus II software, the device automatically activates the CRC error detection process after entering user mode.

If an error is detected within a frame, `CRC_ERROR` is driven high at the end of the error location search, after the EMR 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 EMR 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 10-8 lists the minimum and maximum error detection frequencies.

Table 10-8. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (n)
Stratix V	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 10-11). The divisor is a power of two (2), where *n* is between 1 and 8. The divisor ranges from 2 through 256 (Equation 10-1).

Equation 10-1.

$$\text{Error Detection Frequency} = \frac{100\text{MHz}}{2^n}$$



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

The EMR is updated whenever an error occurs. If the error location and message are not shifted out before the next error location is found, the previous error location and message are overwritten by the new information. To avoid this, you must shift these bits out within one frame of the CRC verification. The minimum interval time between each update for the EMR depends on the device and the error detection clock frequency. However, slowing down the error detection clock frequency slows down the error recovery time for the SEU event.

Table 10-9 lists the estimated minimum interval time between each update for the EMR for Stratix V devices.

Table 10-9. Minimum Update Interval for Error Message Register—Preliminary

Family	Device	Timing Interval (μs)
Stratix V GX	5SGXA3	2.99
	5SGXA4	2.99
	5SGXA5	3.59
	5SGXA7	3.59
	5SGXA9	4.87
	5SGXAB	4.87
	5SGXB5	3.73
	5SGXB6	3.73
Stratix V GT	5SGTC5	3.59
	5SGTC7	3.59
Stratix V GS	5SGSD3	2.43
	5SGSD4	2.43
	5SGSD5	2.99
	5SGSD6	4.21
	5SGSD8	4.21
Stratix V E	5SEE9	4.87
	5SEEB	4.87

The 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.

The minimum CRC calculation time is calculated using the maximum error detection frequency with a divisor factor of 1. The maximum CRC calculation time is calculated using the minimum error detection frequency with a divisor factor of 8. You can estimate the CRC calculation time for other valid divisors (n) for a specific device density using the Equation 10-2:

Equation 10-2.

$$\text{CRC Calculation Time} = \frac{\text{Minimum calculation time}}{2} \times 2^n$$

Table 10-10 lists the minimum and maximum estimated clock frequency time for each CRC calculation for Stratix V devices.

Table 10-10. CRC Calculation Time—Preliminary

Family	Device	Minimum Time (μ s)	Maximum Time (μ s)
Stratix V GX	5SGXA3	2.71	5.42
	5SGXA4	2.71	5.42
	5SGXA5	3.57	7.14
	5SGXA7	3.57	7.14
	5SGXA9	4.85	9.70
	5SGXAB	4.85	9.70
	5SGXB5	3.71	7.42
	5SGXB6	3.71	7.42
Stratix V GT	5SGTC5	3.57	7.14
	5SGTC7	3.57	7.14
Stratix V GS	5SGSD3	2.97	5.94
	5SGSD4	2.97	5.94
	5SGSD5	2.97	5.94
	5SGSD6	4.53	9.06
	5SGSD8	4.53	9.06
Stratix V E	5SEE9	4.85	9.70
	5SEEB	4.85	9.70

Software Support

The Quartus II software, starting with version 10.0, supports the error detection CRC feature for Stratix V devices. Enabling this feature in the **Device and Pin Options** dialog box generates the `CRC_ERROR` output to the optional dual-purpose `CRC_ERROR` pin.

To enable the error detection feature using CRC, perform the following steps:

1. Open the Quartus II software and load a project using a Stratix V device.
2. On the Assignments menu, click **Device**. The **Device** dialog box appears.
3. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
4. In the **Category** list, click **Error Detection CRC**.
5. Turn on **Enable Error Detection CRC_ERROR pin**.
6. By default, the Quartus II software sets the `CRC_ERROR` pin as output open-drain when you enable the error detection CRC circuitry. You can set this pin to be an output by turning off the **Enable open-drain on CRC_ERROR pin**.
7. In the **Divide error check frequency by** box, enter a valid divisor as shown in [Table 10-8 on page 10-8](#).
8. To enable or disable the on-chip error correction feature, turn on or turn off **Enable internal scrubbing**.
9. Click **OK**.

Recovering From CRC Errors

The system that the Stratix V 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.

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 may require a design to account for these errors.

Document Revision History

Table 10-11 lists the revision history for this chapter.

Table 10-11. Document Revision History

Date	Version	Changes
November 2011	1.3	<ul style="list-style-type: none">■ Updated Table 10-9 and Table 10-10.■ Updated “User Mode Error Detection and Correction” section.
May 2011	1.2	<ul style="list-style-type: none">■ Chapter moved to volume 2 for the 11.0 release.■ Updated Table 10-9 and Table 10-10.■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.