This chapter describes how to activate and use the error detection cyclic redundancy check (CRC) feature when your Arria® II device is in user mode and how to recover from configuration errors caused by CRC errors.

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 an Arria II device is correct.

■ Alert the system to the occurrence of a configuration error.

☞ The error detection CRC feature is provided in the Quartus® II software starting with version 9.1 for Arria II GX devices and version 10.1 for Arria II GZ devices.

Using the error detection CRC feature on Arria II devices has no impact on fitting or performance.

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

This chapter contains the following sections:

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 Arria II 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 CRC is stored in the configuration data and contains the CRC value for each data frame.

During configuration, the Arria II 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 complete.

In Arria II 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 into the configuration RAM. The configuration RAM chain used for storing CRC check bits is 16 bits wide and its length is equal to the number of frames in the device.

User Mode Error Detection

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

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

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

A single 16-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 16-bit signature is hex `0000`. If the error detection circuitry detects no configuration RAM bit errors in a frame, the output signal `CRC_ERROR` is 0. If the circuitry detects a configuration RAM 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 Arria II 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 triple-bit errors. The probability of more than three configuration RAM bits being flipped by a single event upset (SEU) is very low. In general, the probability of detection for all error patterns is 99.998%.

The error detection circuitry 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 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.

☞ For more information about the timing requirement to shift out error information from the EMR, refer to "Error Detection Timing" on page 10–7.

The error detection circuitry continues to calculate the CRC_ERROR and 16-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 Arria II 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 configuration RAM. The total storage size is 16 (number of bits per frame) × the number of frames.

The CRC_ERROR signal is asserted if the error detection circuitry verification does not match with the configuration-computed CRC value. However, the Arria II device error detection CRC feature does not check the memory blocks and I/O buffers. Therefore, the CRC_ERROR signal may stay solid high or low, depending on the error status of the previously checked configuration RAM 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 configuration RAM cells. MLAB and M9K memory blocks support parity bits that are used to check the contents of the memory blocks for any error in Arria II GX devices. In addition to MLAB and M9K memory blocks, M144K memory blocks are used to check the contents of the memory blocks for any error in Arria II GZ devices.

👣 For more information about error detection in Arria II memory blocks, refer to the *Memory Blocks in Arria II Devices* chapter.

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 21-bit JTAG fault injection register used for error injection in Arria II devices, thereby 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 10–1 lists the EDERROR_INJECT JTAG instruction for Arria II devices.

Table 10–1. EDERROR_INJECT JTAG Instruction for Arria II Devices

JTAG Instruction	Instruction Code	Description
EDERROR_INJECT	00 0001 0101	This instruction controls the 21-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.

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

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 Arria II 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 10–6.

Table 10–2 lists how the fault injection register is implemented and describes error injection for Arria II devices.

Table 10–2. Fault Injection Register for Arria II Devices

Description	Bit[20..19]		Error Injection Type	Bit[18..8] Byte Location of the Injected Error	Bit[7..0] Error Byte Value
	Error Type (1)				
	Bit[20]	Bit[19]			
Content	0	1	Single 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.
	1	0	Double-adjacent error injection		
	0	0	No error injection		

Note to Table 10–2:

(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.

Automated Single Event Upset Detection

Arria II devices offer on-chip circuitry for automated 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 Arria II devices, eliminating the need for external logic. The CRC_ERROR pin reports a CRC error when configuration RAM data is corrupted; you must decide whether to reconfigure the device or to ignore the error.

Error Detection Pin Description

Table 10–3 lists the CRC_ERROR pin description for Arria II devices.

Table 10–3. CRC_ERROR Pin Description for Arria II Devices

Pin Name	Pin Type	Description
CRC_ERROR	I/O or output open-drain	Active high signal indicating that the error detection circuit has detected errors in the configuration RAM 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 a user I/O pin. When using the WYSIWYG function, the CRC error output is a dedicated path to the CRC_ERROR pin. To use the CRC_ERROR pin, you can tie this pin to V_{CCIO} 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.

Error Detection Block

The error detection block contains the logic necessary to calculate the 16-bit error detection CRC signature for the configuration RAM bits in the Arria II 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 10–4 lists the two types of CRC detection that check the configuration bits for Arria II devices.

Table 10–4. Two Types of CRC Detection for Arria II Devices

User Mode CRC Detection	Configuration CRC Detection
■ This is the configuration RAM error checking ability (16-bit error detection CRC) during user mode for use by the CRC_ERROR pin. ■ For each frame of data, the pre-calculated 16-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 16-bit error detection CRC result for the first frame and also shifts the 16-bit error detection CRC bits to the 16-bit error detection CRC storage registers for test purposes. ■ You can deliberately introduce single error, double errors, or double-adjacent errors to the configuration memory for testing and design verification.	■ 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 Arria II device, the pre-computed configuration CRC is shifted into the CRC circuitry. ■ At the same time, the 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, nSTATUS 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 Block" section focuses on the first type, 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 10–1 shows the block diagram of the error detection circuitry, syndrome registers, and error injection block for Arria II devices.

Figure 10–1. Error Detection Circuitry, Syndrome Registers, and Error Injection Block for Arria II Devices
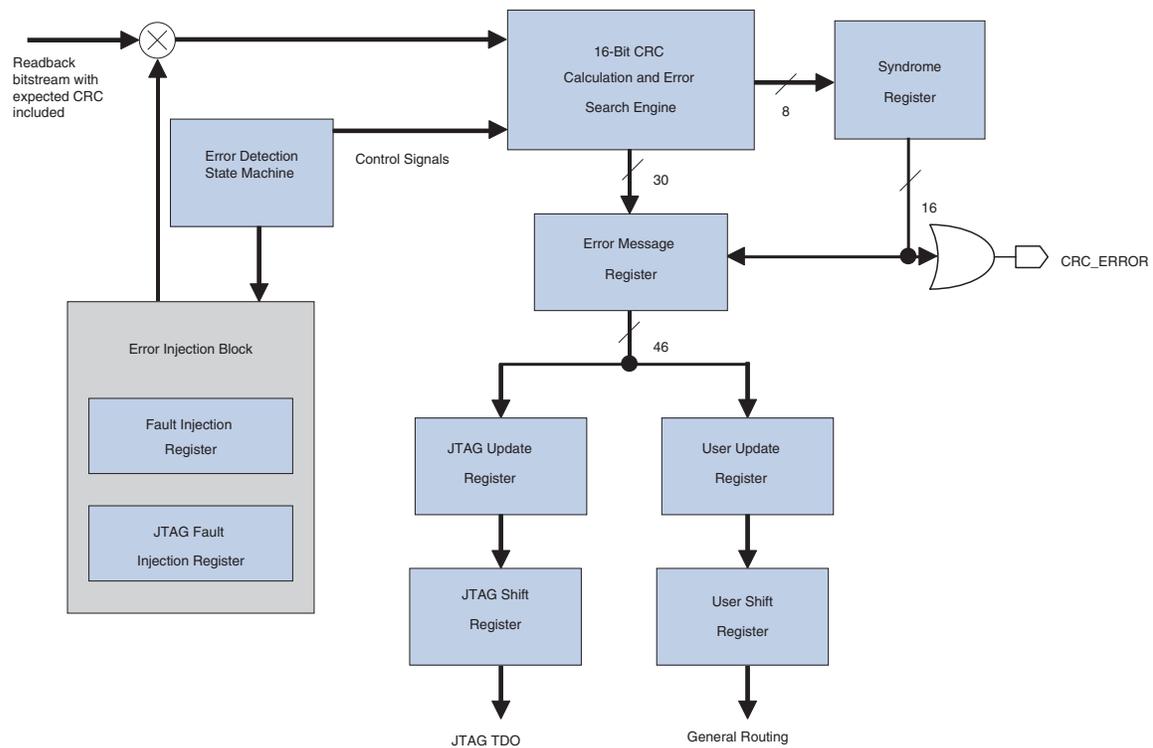
Table 10–5 lists the registers shown in Figure 10–1.

Table 10–5. Error Detection Registers for Arria II Devices

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 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 register is automatically updated with the contents of the EMR one cycle after the 46-bit 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.
User Update Register	This register is automatically updated with the contents of the EMR one cycle after the 46-bit 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 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 out by SHIFT_EDERROR_REG JTAG instruction.
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 EDERROR_INJECT 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 21-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, 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 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 10–6 lists the minimum and maximum error detection frequencies for Arria II devices.

Table 10–6. Minimum and Maximum Error Detection Frequencies for Arria II Devices

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (n)
Arria II	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 10–9). The divisor is a power of two (2), where n is between 1 and 8. The divisor ranges from 2 through 256 (refer to 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 Arria II 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 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 10–7 lists the estimated minimum interval time between each update for the EMR in Arria II devices.

Table 10–7. Minimum Update Interval for Error Message Register in Arria II Devices

Device	Timing Interval (μs)
EP2AGX45	11.04
EP2AGX65	11.04
EP2AGX95	14.88
EP2AGX125	14.88
EP2AGX190	19.64
EP2AGX260	19.64
EP2AGZ225	19.8
EP2AGZ300	21.8
EP2AGZ350	21.8

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.

Table 10–8 lists the minimum and maximum estimated clock frequency time for each CRC calculation for Arria II devices. The minimum CRC calculation time is calculated using the maximum error detection frequency with a divisor factor 1. The maximum CRC calculation time is calculated using the minimum error detection frequency with a divisor factor 8.

Table 10–8. CRC Calculation Time for Arria II Devices

Device	Minimum Time (ms)	Maximum Time (s)
EP2AGX45	73.80	20.40
EP2AGX65	73.80	20.40
EP2AGX95	125.80	34.80
EP2AGX125	125.80	34.80
EP2AGX190	216.00	59.90
EP2AGX260	216.00	59.90
EP2AGZ225	225	62.44
EP2AGZ300	296	82.05
EP2AGZ350	296	82.05

Software Support

The Quartus II software, starting with version 9.1 supports the error detection CRC feature for Arria II GX devices and starting with version 10.1 supports the error detection CRC feature for Arria II GZ 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 the CRC, follow these steps:

1. Open the Quartus II software and load a project using an Arria II 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, select **Error Detection CRC** tab.

5. Turn on **Enable Error Detection CRC**.

6. In the **Divide error check frequency by** pull-down list, enter a valid divisor as listed in Table 10–6 on page 10–7.

7. Click **OK**.

Recovering From CRC Errors

The system that the Arria II 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 10–9 lists the revision history for this chapter.

Table 10–9. Document Revision History

Date	Version	Changes
February 2014	4.3	Updated the minimum CRC calculation time for EP2AGX45, EP2AGX65, EP2AGX95, EP2AGX125, EP2AGX190, and EP2AGX260 in Table 10–8.
July 2012	4.2	Removed repeated paragraph in the "User Mode Error Detection" section.
June 2011	4.1	■ Updated "User Mode Error Detection" section. ■ Minor text edits.
December 2010	4.0	■ Updated for the Quartus II software version 10.1 release. ■ Added Arria II GZ devices information. ■ Minor text edits.
July 2010	3.0	Updated for Arria II GX v10.0 release: ■ Updated Table 10–3, Table 10–6, Table 10–7, and Table 10–8.
November 2009	2.0	Updated for Arria II GX v9.1 release: ■ Updated Table 10–7 and Table 10–8. ■ Minor text edits.
February 2009	1.0	Initial release.