SIII51015-1.7

This chapter describes how to use the error detection cyclical redundancy check (CRC) feature when a Stratix® III device is in user mode and recovers from CRC errors. The purpose of the error detection CRC feature is to detect a flip in any of the configuration CRAM bits in Stratix III devices due to a soft error. By using 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 III device is correct.

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

☞ The error detection feature has been enhanced in the Stratix III device family. In addition, the error detection and recovery time for single event upset (SEU) in Stratix III devices is reduced compared to Stratix II devices.

☞ For Stratix III devices, use of the error detection CRC feature is provided in the Quartus® II software version 6.1 and onwards.

☞ Stratix III devices only support the error detection CRC feature at 1.1 V for V_{CCL}. This feature is not supported in Stratix III devices operating at 0.9 V for V_{CCL}.

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

This section describes how to activate and use the error detection CRC feature when your Stratix III device is in user mode and describes how to recover from configuration errors caused by CRC errors.

Information about SEU is located on the Products page of the Altera® website at www.altera.com.

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

For more information, refer to the *Robust SEU Mitigation with Stratix III FPGAs White Paper*.

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

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

User Mode Error Detection

Stratix III 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's 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).

As soon as the device transitions into user mode, you can enable the error detection process if you enable the CRC error detection option. The internal 100-MHz configuration oscillator is divided down by a factor of 2 to 256 (at powers of 2) to be used as the clock source during the error detection process. Set the clock divide factor in the option setting in the Quartus II software.

A single 16-bit CRC calculation is done on a per-frame basis. Once it has finished the CRC calculation for a frame, the resulting 16-bit signature is hex `0000` if there are no detected CRAM bit errors 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 the error bit location.

Error detection in Stratix III 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 the report of the bit flips error location is not guaranteed by the CRC engine.

You can also read-out the error bit location through the Joint Test Action Group (JTAG) and the core interface. You must shift these bits out through either the JTAG instruction, `SHIFT_EDERROR_REG`, or the core interface before the CRC detects the next error in another frame. If the next frame also has an error, you have to 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. Refer to Table 15–6 on page 15–10 for the minimum update interval for Stratix III devices. 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 must monitor these signals and take the appropriate actions if a soft error occurs.

Error detection circuitry in Stratix III 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 registers within the core. The total storage register size is 16 (number of bits per frame) × the number of frames.

The Stratix III device error detection feature does not check memory blocks and I/O buffers. These memory blocks support parity bits that are used to check the contents of memory blocks for any error. The I/O buffers are not verified during error detection because these bits use flip-flops as storage elements that are more resistant to soft errors compared to CRAM cells.

The M144K TriMatrix memory block has a built-in error correction code block that checks and corrects errors in the block. However, for logic array blocks (LABs) that are used as MLAB memory blocks, they are ignored during error detection verification. Thus, the `CRC_ERROR` signal may stay solid high or low depending on the error status of the previous checked CRAM frame.

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

In order to provide testing capability of the error detection block, a JTAG instruction EDERROR_INJECT is provided. This instruction can change the content of the 21-bit JTAG fault injection register, used for error injection in Stratix III devices, hence enabling testing of the error detection block.

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

Table 15–1 lists the EDERROR_INJECT JTAG instruction.

Table 15–1. 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 only execute the EDERROR_INJECT JTAG instruction at error detection frequency 50 MHz. Refer to "Error Detection Timing" on page 15–9 for instructions about how to set the error detection frequency in the Quartus II software. For the testing of the CRC detection block with the frequency lower than 50 MHz, contact Altera Technical Support at www.altera.com/support.

You can create Jam™ files (.**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, double errors, 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 EDERROR_INJECT instruction to flip the readback bits. The Stratix III 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 the 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 15–7.

Table 15–2 lists how the fault injection register is implemented and describes error injection.

Table 15–2. Fault Injection Register and Error Injection

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 15–2:

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

☞ After the test completes, Altera recommends that you reconfigure the device.

Automated Single Event Upset Detection

Stratix III devices offer on-chip circuitry for automated checking of single-event upset 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 III devices, eliminating the need for external logic. The CRC_ERROR pin reports a soft error when configuration CRAM data is corrupted and you would have to 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 will need to use different error detection pins to monitor the data during user mode. The different error detection pins available are described in the following sections.

CRC_ERROR Pin

Table 15–3 lists the CRC_ERROR pin.

Table 15–3. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O, output, or open-drain output (optional)	Active high signal that indicates 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.
		The CRC error output, when using the WYSIWYG function, is a dedicated path to the CRC_ERROR pin. By default, the Quartus II software sets the CRC_ERROR pin as a dedicated output.
		If CRC_ERROR is used as a dedicated output, make sure V_{CCIO} of the bank where the pin resides meets the input voltage specification of the system receiving the signal. Optionally, you can set this pin to be an open-drain output by enabling the option in the Quartus II software from the **Error Detection CRC** tab of the **Device & Pin Options** dialog box.
		Using this pin as open-drain provides advantage on voltage leveling. To use this pin as open-drain, tie the pin to V_{CCPGM} through a 10-kΩ resistor. Alternatively, depending on the voltage input voltage specification of the system receiving the signal, you can tie the pull-up resistor to a different pull-up voltage.

☞ WYSIWYG is a design primitive that corresponds to device features and can be directly instantiated into your RTL design.

👣 The CRC_ERROR pin information for Stratix III devices is reported in *Device Pin-Outs* on the Literature page of the Altera website (www.altera.com).

Error Detection Block

You can enable the Stratix III device error detection block in the Quartus II software (refer to"Software Support" on page 15–11). 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 check the configuration bits:

■ The CRAM error checking ability (16-bit CRC) during user mode, for use by the CRC_ERROR pin.

- For each frame of data, the pre-calculated 16-bit CRC enters the CRC circuit right at the end of the frame data and determines whether or not there is an error.

- If an error occurs, the search engine starts to find the location of the error.

- You can shift the error messages out through the JTAG instruction or core interface logic 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.

- You can deliberately introduce single error, double errors, or double errors adjacent to each other to configuration memory for testing and design verification.

☞ The "Error Detection Registers" section focuses on the first type, the 16-bit CRC only when the device is in user mode.

■ The 16-bit CRC that is embedded in every configuration data frame.

- During configuration, after a frame of data is loaded into the Stratix III 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 the configuration data frame.

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 15–1 shows the block diagram of the error detection circuitry, syndrome registers, and error injection block.
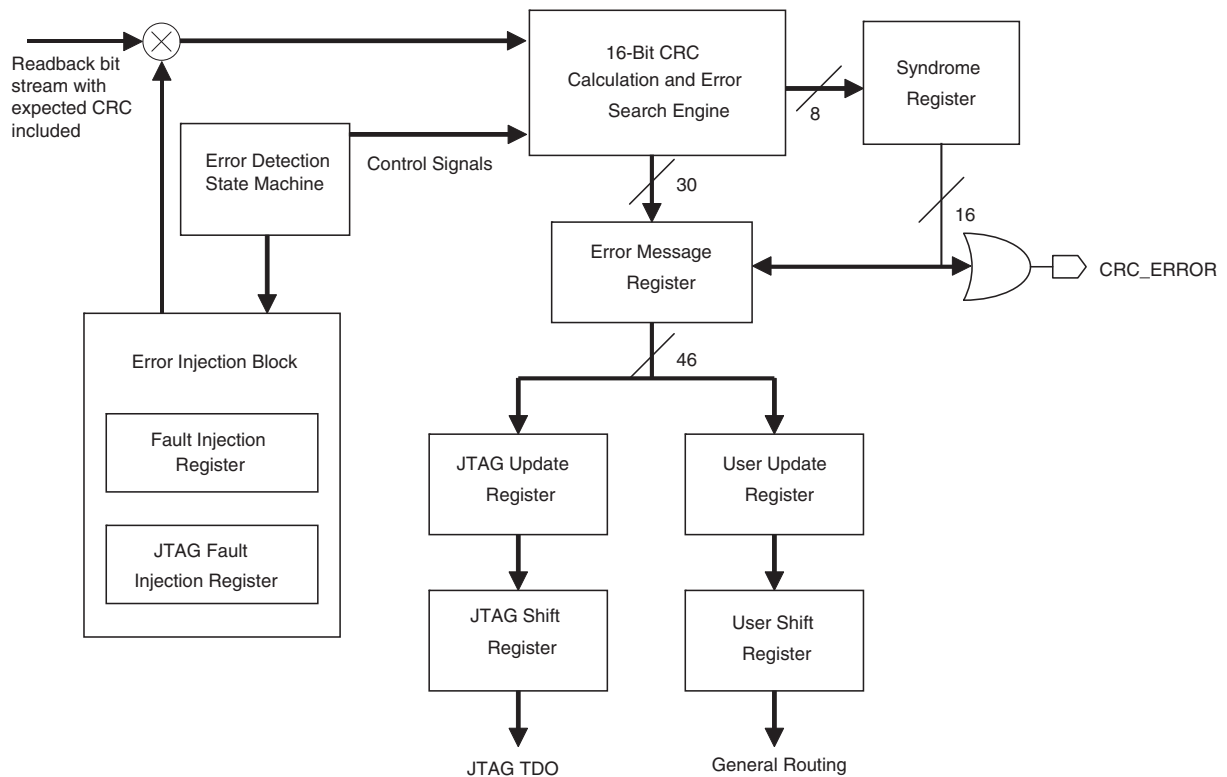
Figure 15–1. Error Detection Block Diagram

Table 15–5 lists the registers shown in Figure 15–1.

Table 15–4. 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 about 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. You can shift out the content of the register through the JTAG SHIFT_EDERROR_REG instruction or to the core through the core interface.
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 which needs to 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 exactly 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 which needs to 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.

Table 15–4. Error Detection Registers   (Part 2 of 2)

Register	Description
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 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 the 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 in this 21-bit register when it is being updated.

Error Detection Timing

When the CRC feature is enabled through the Quartus II software, the device automatically activates the CRC process upon 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, and after the Error Message Register gets updated. At the end of this cycle, the `CRC_ERROR` pin is pulled low for a minimum 32 clock cycles. If the next frame also contains an error, the `CRC_ERROR` is driven high again after the Error Message Register gets overwritten by the new value. You can start to unload the error message on each rising edge of `CRC_ERROR` pin. The error detection runs until the device is reset.

Error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency. Table 15–5 lists the minimum and maximum error detection frequencies.

Table 15–5. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Exponents (n)
Stratix III	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 15–11). 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 15–1.

Equation 15–1.

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

☞ The error detection frequency reflects the frequency of the error detection process for a frame because the CRC calculation in Stratix III devices 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 15–6 lists the estimated minimum interval time between each update for the Error Message Register for Stratix III devices.

Table 15–6. Minimum Update Interval for Error Message Register   *(Note 1)*

Device	Timing Interval (µs)
EP3SL50	9.8
EP3SL70	9.8
EP3SL110	14.8
EP3SL150	14.8
EP3SL200	19.8
EP3SE260	19.8
EP3SL340	21.8
EP3SE50	9.8
EP3SE80	14.8
EP3SE110	14.8

Note to Table 15–6:

(1)   These timing numbers are preliminary.

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 15–7 lists the estimated time for each CRC calculation with minimum and maximum clock frequencies for Stratix III devices. The minimum CRC calculation time is calculated by using the maximum error detection frequency with divisor factor 1 while the maximum CRC calculation time is calculated by using the minimum error detection frequency with divisor factor 8.

Table 15–7. CRC Calculation Time

| Device	Minimum Time (ms)	Maximum Time (s)
EP3SL50	52.00	14.36
EP3SL70	52.00	14.36
EP3SL110	110.00	30.38
EP3SL150	110.00	30.38
EP3SL200	212.00	58.72
EP3SL260	212.00	58.72
EP3SL340	270.00	74.87
EP3SE50	59.00	16.41
EP3SE80	113.00	31.28
EP3SE110	113.00	31.28

Software Support

The Quartus II software, starting with version 6.1, supports the error detection CRC feature for Stratix III 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.

Enable the error detection feature using CRC by performing the following steps:

1. Open the Quartus II software and load a project that uses a Stratix III 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 (Figure 15–2).

5. In the **Device and Pin Options** dialog box, click the **Error Detection CRC** tab.

6. Turn on **Enable error detection CRC** (Figure 15–2).

Figure 15–2. Enabling the Error Detection CRC Feature in the Quartus II Software
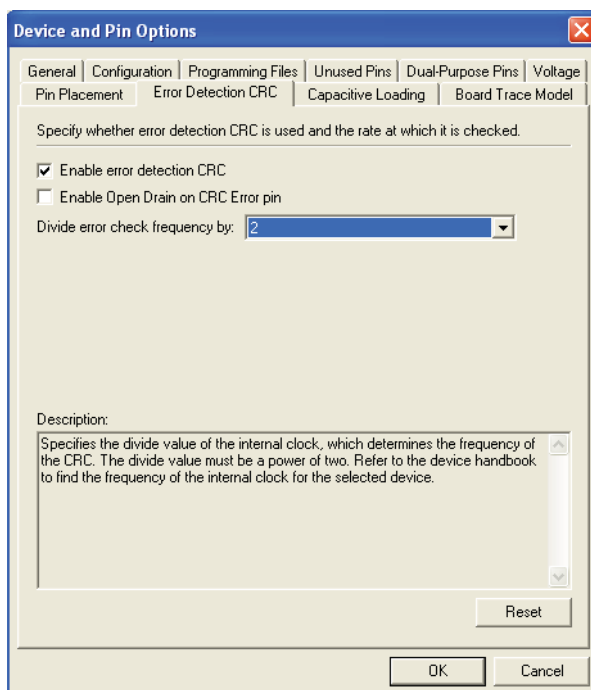
7. In the **Divide error check frequency by** box, enter a valid divisor as documented in Table 15–5 on page 15–9.

☞ 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 contains the Stratix III device must control the 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 may require a design to account for these errors.

Chapter Revision History

Table 15–8 lists the revision history for this chapter.

Table 15–8. Chapter Revision History

Date	Version	Changes Made
March 2010	1.7	Updated for the Quartus II software version 9.1 SP2 release: ■ Updated Table 15–6. ■ Minor text edits.
May 2009	1.6	Updated "User Mode Error Detection" and "CRC_ERROR Pin" sections.
February 2009	1.5	■ Updated "Error Detection Timing" section. ■ Removed "Referenced Documents", Critical Error Detection", and "CRITICAL ERROR Pin" sections.
October 2008	1.4	■ Updated "Introduction" and "Referenced Documents" sections. ■ Updated New Document Format.
May 2008	1.3	■ Updated "Configuration Error Detection", "User Mode Error Detection", and "Error Detection Timing" sections. ■ Updated Table 15–3, Table 15–6, and Table 15–7. ■ Updated Figure 15–2 and Figure 15–3.
October 2007	1.2	■ Minor edits to Table 15–3. ■ Added new section "Referenced Documents". ■ Added live links for references.
May 2007	1.1	■ Minor edits to page 2, 3, 4, and 14. ■ Updated Table 15–5.
November 2006	1.0	Initial Release.