# 14. Remote System Upgrades with Stratix II & Stratix II GX Devices

## Introduction

System designers today face difficult challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. Stratix® II and Stratix II GX FPGAs help overcome these challenges with their inherent re-programmability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduce time-to-market, and extend product life.

Stratix II and Stratix II GX FPGAs feature dedicated remote system upgrade circuitry. Soft logic (either the Nios® embedded processor or user logic) implemented in a Stratix II or Stratix II GX device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information. This dedicated remote system upgrade circuitry is unique to Stratix, Stratix II, and Stratix II GX FPGAs and helps to avoid system downtime.

Remote system upgrade is supported in all Stratix II and Stratix II GX configuration schemes: fast passive parallel (FPP), active serial (AS), passive serial (PS), and passive parallel asynchronous (PPA). Remote system upgrade can also be implemented in conjunction with advanced Stratix II and Stratix II GX features such as real-time decompression of configuration data and design security using the advanced encryption standard (AES) for secure and efficient field upgrades.

This chapter describes the functionality and implementation of the dedicated remote system upgrade circuitry. It also defines several concepts related to remote system upgrade, including factory configuration, application configuration, remote update mode, local update mode, the user watchdog timer, and page mode operation. Additionally, this chapter provides design guidelines for implementing remote system upgrade with the various supported configuration schemes.

# Functional Description

The dedicated remote system upgrade circuitry in Stratix II and Stratix II GX FPGAs manages remote configuration and provides error detection, recovery, and status information. User logic or a Nios processor implemented in the FPGA logic array provides access to the remote configuration data source and an interface to the system's configuration memory.

Stratix II and Stratix II GX FPGA's remote system upgrade process involves the following steps:

1. A Nios processor (or user logic) implemented in the FPGA logic array receives new configuration data from a remote location. The connection to the remote source is a communication protocol such as the transmission control protocol/Internet protocol (TCP/IP), peripheral component interconnect (PCI), user datagram protocol (UDP), universal asynchronous receiver/transmitter (UART), or a proprietary interface.

2. The Nios processor (or user logic) stores this new configuration data in non-volatile configuration memory. The non-volatile configuration memory can be any standard flash memory used in conjunction with an intelligent host (for example, a MAX® device or microprocessor), the serial configuration device, or the enhanced configuration device.

3. The Nios processor (or user logic) initiates a reconfiguration cycle with the new or updated configuration data.

4. The dedicated remote system upgrade circuitry detects and recovers from any error(s) that might occur during or after the reconfiguration cycle, and provides error status information to the user design.

Figure 14–1 shows the steps required for performing remote configuration updates. (The numbers in the figure below coincide with the steps above.)

*Figure 14–1. Functional Diagram of Stratix II or Stratix II GX Remote System Upgrade*
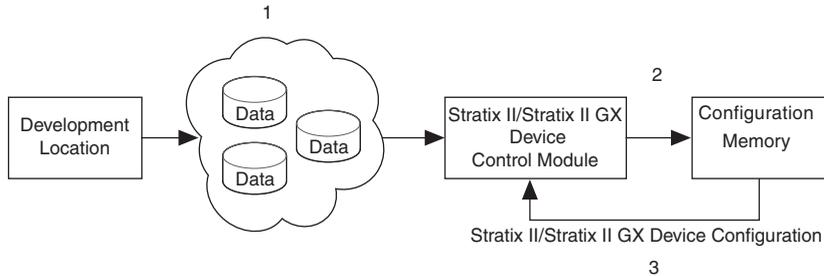


Stratix II and Stratix II GX FPGAs support remote system upgrade in the FPP, AS, PS, and PPA configuration schemes.
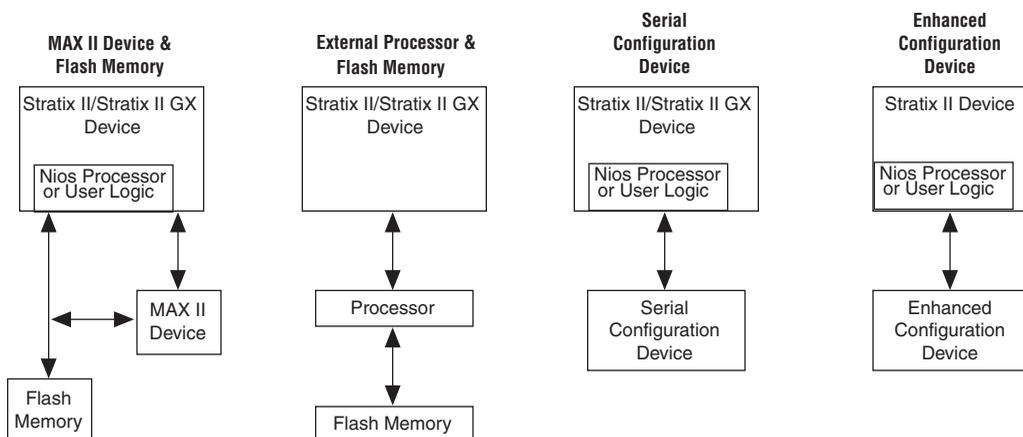
■ Serial configuration devices use the AS scheme to configure Stratix II and Stratix II GX FPGAs.
■ A MAX II device (or microprocessor and flash configuration schemes) uses FPP, PS, or PPA schemes to configure Stratix II and Stratix II GX FPGAs.
■ Enhanced configuration devices use the FPP or PS configuration schemes to configure Stratix II and Stratix II GX FPGAs.

☞ The JTAG-based configuration scheme does not support remote system upgrade.

Figure 14–2 shows the block diagrams for implementing remote system upgrade with the various Stratix II and Stratix II GX configuration schemes.

*Figure 14–2. Remote System Upgrade Block Diagrams for Various Stratix II and Stratix II GS Configuration Schemes*



☞ For the active serial configuration scheme, the remote system upgrade only supports single device configurations.

You must set the mode select pins (MSEL[3..0]) and the RUnLU pin to select the configuration scheme and remote system upgrade mode best suited for your system. Table 14–1 lists the pin settings for Stratix II and Stratix II GX FPGAs. Standard configuration mode refers to normal FPGA configuration mode with no support for remote system upgrades, and the remote system upgrade circuitry is disabled. The following sections describe the local update and remote update remote system upgrade modes.

For more information on standard configuration schemes supported in Stratix II and Stratix II GX FPGAs, see the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook* and the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*.

*Table 14–1. Stratix II and Stratix II GX Remote System Upgrade Modes*

| Configuration Scheme | MSEL[3..0] | RUnLU | Remote System Upgrade Mode |
|---|---|---|---|
| FPP | 0000 | - | Standard |
| | 0100 *(1)* | 0 | Local update |
| | 0100 *(1)* | 1 | Remote update |
| FPP with decompression and/or design security feature enabled *(2)* | 1011 | - | Standard |
| | 1100 *(1)* | 0 | Local update |
| | 1100 *(1)* | 1 | Remote update |
| Fast AS (40 MHz) *(3)* | 1000 | - | Standard |
| | 1001 | 1 | Remote update |
| AS (20 MHz) *(3)* | 1101 | - | Standard |
| | 1110 | 1 | Remote update |
| PS | 0010 | - | Standard |
| | 0110 *(1)* | 0 | Local update |
| | 0110 *(1)* | 1 | Remote update |
| PPA | 0001 | - | Standard |
| | 0101 *(1)* | 0 | Local update |
| | 0101 *(1)* | 1 | Remote update |

*Notes to Table 14–1:*

(1)    These schemes require that you drive the RUnLU pin to specify either remote update or local update mode. AS schemes only support the remote update mode.

(2)    These modes are only supported when using a MAX II device or microprocessor and flash for configuration. In these modes, the host system must output a DCLK that is 4 x the data rate.

(3)    The EPCS16 and EPCS64 serial configuration devices support a DCLK up to 40 MHz; other EPCS devices support a DCLK up to 20 MHz. See the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook* for more information.

## Configuration Image Types and Pages

When using remote system upgrade, FPGA configuration bitstreams are classified as factory configuration images or application configuration images. An image, also referred to as a configuration, is a design loaded into the FPGA that performs certain user-defined functions. Each FPGA in your system requires one factory image and one or more application

images. The factory image is a user-defined fall-back, or safe, configuration and is responsible for administering remote updates in conjunction with the dedicated circuitry. Application images implement user-defined functionality in the target FPGA.

A remote system update involves storing a new application configuration image or updating an existing one via the remote communication interface. After an application configuration image is stored or updated remotely, the user design in the FPGA initiates a reconfiguration cycle with the new image. Any errors during or after this cycle are detected by the dedicated remote system upgrade circuitry and cause the FPGA to automatically revert to the factory image. The factory image then performs error processing and recovery. While error processing functionality is limited to the factory configuration, both factory and application configurations can download and store remote updates and initiate system reconfiguration.

Stratix II and Stratix II GX FPGAs select between the different configuration images stored in the system configuration memory using the page address pins or start address registers. A page is a section of the configuration memory space that contains one configuration image for each FPGA in the system. One page stores one system configuration, regardless of the number of FPGAs in the system.

Page address pins select the configuration image within an enhanced configuration device or flash memory (MAX II device or microprocessor setup). Page start address registers are used when Stratix II and Stratix II GX FPGAs are configured in AS mode with serial configuration devices. Figure 14–3 illustrates page mode operation in Stratix II and Stratix II GX FPGAs.

*Figure 14–3. Page Mode Operation in Stratix II & Stratix II GX FPGAs*
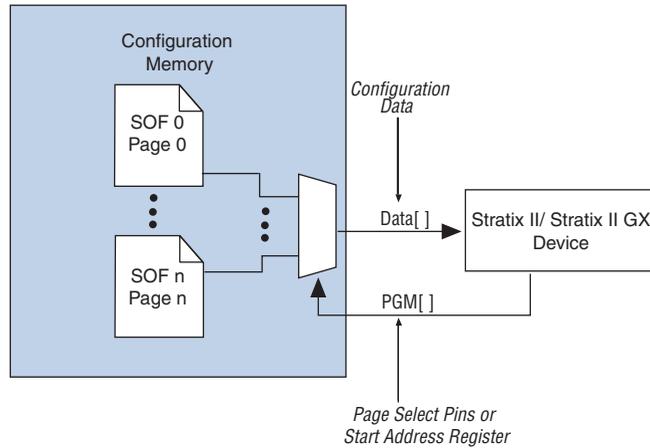


Stratix II and Stratix II GX devices drive out three page address pins, PGM[2..0], to the MAX II device or microprocessor or enhanced configuration device. These page pins select between eight configuration pages. Page zero (PGM[2..0] = 000) must contain the factory configuration, and the other seven pages are application configurations. The PGM[] pins are pointers to the start address and length of each page, and the MAX II device, microprocessor, and enhanced configuration devices perform this translation.

☞ When implementing remote system upgrade with an intelligent-host-based configuration, your MAX II device or microprocessor should emulate the page mode feature supported by the enhanced configuration device, which translates PGM pointers to a memory address in the configuration memory. Your MAX II device or microprocessor must provide a similar translation feature.

For more information about the enhanced configuration device page mode feature, refer to the Dynamic Configuration (Page Mode) Implementation section of the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

When implementing remote system upgrade with AS configuration, a dedicated 7-bit page start address register inside Stratix II and Stratix II GX FPGAs determines the start addresses for configuration pages within the serial configuration device. The PGM[6..0] registers form bits [22..16] of the 24-bit start address while the other 17 bits are

set to zero: StAdd[23..0] = {1'b0, PGM[6..0], 16'b0}. During AS configuration, Stratix II and Stratix II GX FPGAs use this 24-bit page start address to obtain configuration data from the serial configuration devices.

# Remote System Upgrade Modes

Remote system upgrade has two modes of operation: remote update mode and local update mode. The remote and local update modes allow you to determine the functionality of your system upon power up and offer different features. The RUnLU input pin selects between the remote update (logic high) and local update (logic low) modes.

## Overview

In remote update mode, Stratix II and Stratix II GX FPGAs load the factory configuration image upon power up. The user-defined factory configuration should determine which application configuration is to be loaded and trigger a reconfiguration cycle. Remote update mode allows up to eight configuration images (one factory plus seven application images) when used with the MAX II device or microprocessor and flash-based configuration or an enhanced configuration device.

When used with serial configuration devices, the remote update mode allows an application configuration to start at any flash sector boundary. This translates to a maximum of 128 pages in the EPCS64 and 32 pages in the EPCS16 device, where the minimum size of each page is 512 KBits. Additionally, the remote update mode features a user watchdog timer that can detect functional errors in an application configuration.

Local update mode is a simplified version of the remote update mode. In this mode, Stratix II and Stratix II GX FPGAs directly load the application configuration, bypassing the factory configuration. This mode is useful if your system is required to boot into user mode with minimal startup time. It is also useful during system prototyping, as it allows you to verify functionality of the application configuration.

In local update mode, a maximum of two configuration images or pages is supported: one factory configuration, located at page address PGM[2..0] = 000, and one application configuration, located at page address PGM[2..0] = 001. Because the page address of the application configuration is fixed, the local update mode does not require the factory configuration image to determine which application is to be loaded. If any errors are encountered while loading the application configuration, Stratix II and Stratix II GX FPGAs revert to the factory configuration. The user watchdog timer feature is not supported in this mode.

☞ Also, local update mode does not support AS configuration with the serial configuration devices because these devices don't support a dynamic pointer to page 001 start address location.

Table 14–2 details the differences between remote and local update modes.

*Table 14–2. Differences Between Remote and Local Update Modes*

| Features | Remote Update Mode | Local Update Mode |
|---|---|---|
| RUnLU input pin setting | 1 | 0 |
| Page selection upon power up | PGM[2..0] = 000 (Factory) | PGM[2..0] = 001 (Application) |
| Supported configurations | MAX II device or microprocessor-based configuration, serial configuration, and enhanced configuration devices (FPP, PS, AS, PPA) | MAX II device or microprocessor-based configuration and enhanced configuration devices (FPP, PS, PPA) |
| Number of pages supported | Eight pages for external host or controller based configuration; up to 128 pages (512 KBits/page) for serial configuration device | Two pages |
| User watchdog timer | Available | Disabled |
| Remote system upgrade control and status register | Read/write access allowed in factory configuration. Read access in application configuration | Only status register read access allowed in local update mode (factory and application configurations). Write access to control register is disabled |

## Remote Update Mode

When Stratix II and Stratix II GX FPGAs are first powered up in remote update mode, it loads the factory configuration located at page zero (page address pins PGM[2..0] = "000"; page registers PGM[6..0] = "0000000"). You should always store the factory configuration image for your system at page address zero. A factory configuration image is a bitstream for the FPGA(s) in your system that is programmed during production and is the fall-back image when errors occur. This image is stored in non-volatile memory and is never updated or modified using

remote access. This corresponds to PGM[2..0] = 000 of the enhanced configuration device or standard flash memory, and start address location 0x000000 in the serial configuration device.

The factory image is user designed and contains soft logic to:

■ Process any errors based on status information from the dedicated remote system upgrade circuitry
■ Communicate with the remote host and receive new application configurations, and store this new configuration data in the local non-volatile memory device
■ Determine which application configuration is to be loaded into the FPGA
■ Enable or disable the user watchdog timer and load its time-out value (optional)
■ Instruct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle

Figure 14–4 shows the transitions between the factory and application configurations in remote update mode.

*Figure 14–4. Transitions Between Configurations in Remote Update Mode*
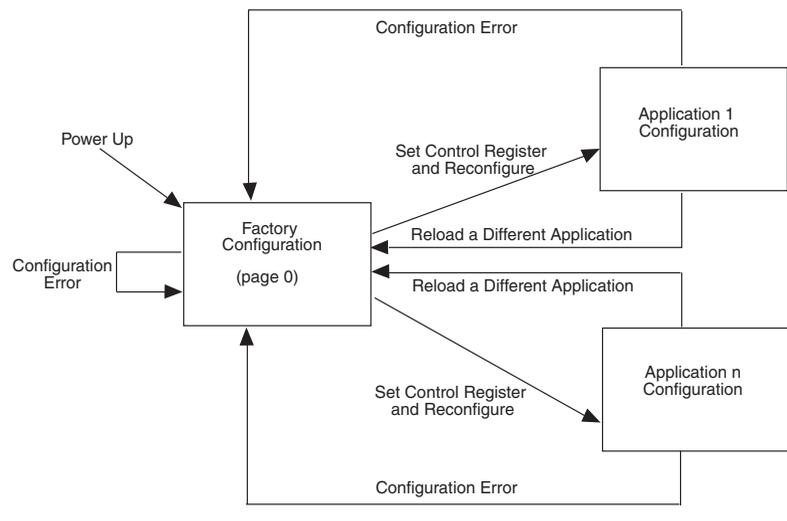
After power up or a configuration error, the factory configuration logic should write the remote system upgrade control register to specify the page address of the application configuration to be loaded. The factory configuration should also specify whether or not to enable the user watchdog timer for the application configuration and, if enabled, specify the timer setting.

The user watchdog timer ensures that the application configuration is valid and functional. After confirming the system is healthy, the user-designed application configuration should reset the timer periodically during user-mode operation of an application configuration. This timer reset logic should be a user-designed hardware and/or software health monitoring signal that indicates error-free system operation. If the user application configuration detects a functional problem or if the system hangs, the timer is not reset in time and the dedicated circuitry updates the remote system upgrade status register, triggering the device to load the factory configuration. The user watchdog timer is automatically disabled for factory configurations.

☞ Only valid application configurations designed for remote update mode include the logic to reset the timer in user mode.

☞ For more information about the user watchdog timer, see "User Watchdog Timer" on page 14–20.

If there is an error while loading the application configuration, the remote system upgrade status register is written by the Stratix II or Stratix II GX FPGA's dedicated remote system upgrade circuitry, specifying the cause of the reconfiguration. Actions that cause the remote system upgrade status register to be written are:

- nSTATUS driven low externally
- Internal CRC error
- User watchdog timer time out
- A configuration reset (logic array nCONFIG signal or external nCONFIG pin assertion)

Stratix II and Stratix II GX FPGAs automatically load the factory configuration located at page address zero. This user-designed factory configuration should read the remote system upgrade status register to determine the reason for reconfiguration. The factory configuration should then take appropriate error recovery steps and write to the remote system upgrade control register to determine the next application configuration to be loaded.

When Stratix II or Stratix II GX devices successfully load the application configuration, they enter into user mode. In user mode, the soft logic (Nios processor or state machine and the remote communication interface) assists the Stratix II or Stratix II GX device in determining when a remote system update is arriving. When a remote system update arrives, the soft logic receives the incoming data, writes it to the configuration memory device, and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and initiates system reconfiguration.

Stratix II and Stratix II GX FPGAs support the remote update mode in the AS, FPP, PS, and PPA configuration schemes. In the FPP, PS, and PPA schemes, the MAX II device, microprocessor, or enhanced configuration device should sample the PGM[2..0] outputs from the Stratix II or Stratix II GX FPGA and transmit the appropriate configuration image. In the AS scheme, the Stratix II or Stratix II GX device uses the page addresses to read configuration data out of the serial configuration device.

## Local Update Mode

Local update mode is a simplified version of the remote update mode. This feature allows systems to load an application configuration immediately upon power up without loading the factory configuration first. Local update mode does not require the factory configuration to determine which application configuration to load, because only one application configuration is allowed (at page address one (PGM [2..0] = 001). You can update this application configuration remotely. If an error occurs while loading the application configuration, the factory configuration is automatically loaded.

Upon power up or nCONFIG assertion, the dedicated remote system upgrade circuitry drives out "001" on the PGM[] pins selecting the application configuration stored in page one. If the device encounters any errors during the configuration cycle, the remote system upgrade circuitry retries configuration by driving PGM[2..0] to zero (PGM[2..0] = 000) to select the factory configuration image. The error conditions that trigger a return to the factory configuration are:

■ An internal CRC error
■ An external error signal (nSTATUS detected low)

When the remote system upgrade circuitry detects an external configuration reset (nCONFIG pulsed low) or internal configuration reset (logic array nCONFIG assertion), the device attempts to reload the application configuration from page one.

Figure 14–5 shows the transitions between configurations in local update mode.

*Figure 14–5. Transitions Between Configurations in Local Update Mode*
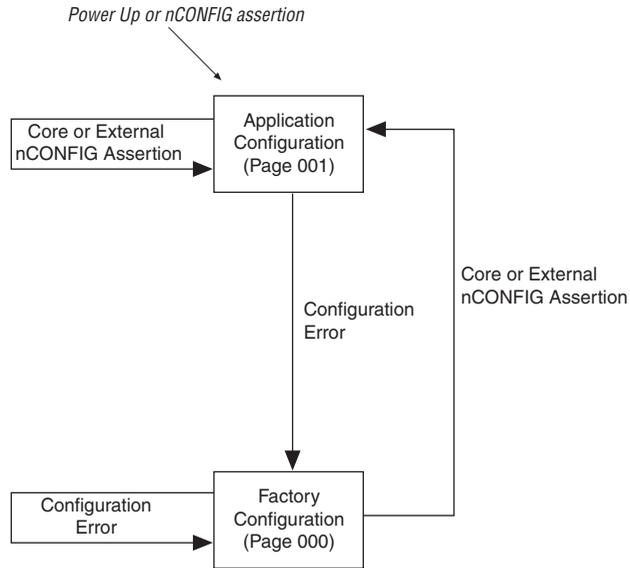


Stratix II and Stratix II GX FPGAs support local update mode in the FPP, PS, and PPA configuration schemes. In these schemes, the MAX II device, microprocessor, or enhanced configuration device should sample the PGM[2..0] outputs from the Stratix II or Stratix II GX FPGA and transmit the appropriate configuration image.

Local update mode is not supported with the AS configuration scheme, (or serial configuration device), because the Stratix II or Stratix II GX FPGA cannot determine the start address of the application configuration page upon power up. While the factory configuration is always located at memory address 0x000000, the application configuration can be located at any other sector boundary within the serial configuration device. The start address depends on the size of the factory configuration and is user selectable. Hence, only remote update mode is supported in the AS configuration scheme.

☞ Local update mode is not supported in the AS configuration scheme (with a serial configuration device).

Local update mode supports read access to the remote system upgrade status register. The factory configuration image can use this error status information to determine if a new application configuration must be downloaded from the remote source. After a remote update, the user design should assert the logic array configuration reset (nCONFIG) signal to load the new application configuration.

The device does not support write access to the remote system upgrade control register in local update mode. Write access is not required because this mode only supports one application configuration (eliminating the need to write in a page address) and does not support the user watchdog timer (eliminating the need to enable or disable the timer or specify its time-out value).

☞ The user watchdog timer is disabled in local update mode.

☞ Write access to the remote system upgrade control register is disabled in local update mode. However, the device supports read access to obtain error status information.

## Dedicated Remote System Upgrade Circuitry

This section explains the implementation of the Stratix II or Stratix II GX remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces to the user-defined factory application configurations implemented in the FPGA logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and a state machine that controls those components. Figure 14–6 shows the remote system upgrade block's data path.

*Figure 14–6. Remote System Upgrade Circuit Data Path*
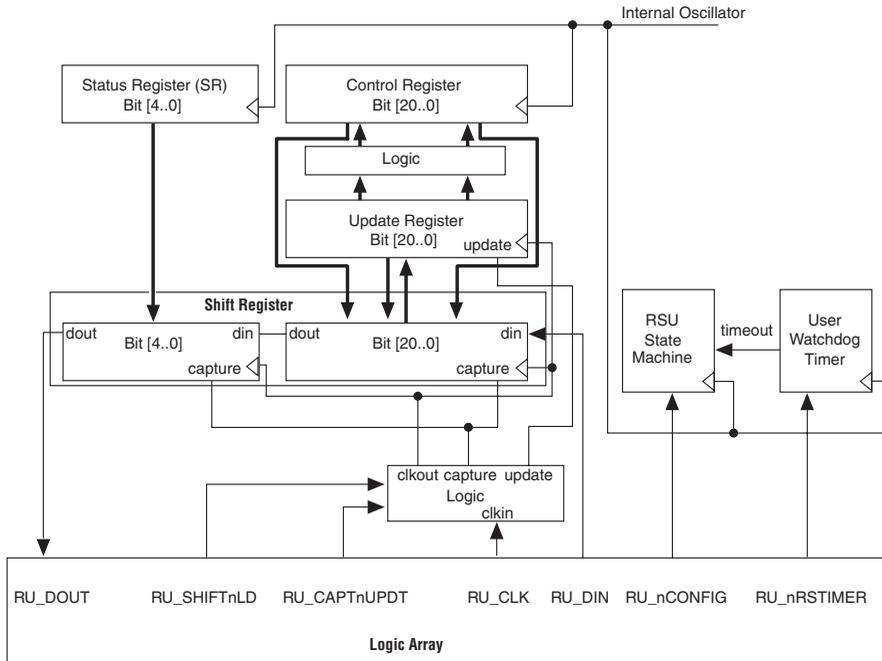


## Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the page addresses, watchdog timer settings, and status information. These registers are detailed in Table 14–3.

*Table 14–3. Remote System Upgrade Registers  (Part 1 of 2)*

| Register | Description |
|---|---|
| Shift register | This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic. Write access is enabled in remote update mode for factory configurations to allow writes to the update register. Write access is disabled in local update mode and for all application configurations in remote update mode. |
| Control register | This register contains the current page address, the user watchdog timer settings, and one bit specifying whether the current configuration is a factory configuration or an application configuration. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is initiated, the contents of the update register are written into the control register. |

*Table 14–3. Remote System Upgrade Registers  (Part 2 of 2)*

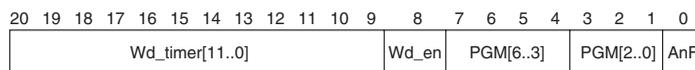| Register | Description |
|---|---|
| Update register | This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a read in a factory configuration, this register is read into the shift register. |
| Status register | This register is written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register. |

The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

*Remote System Upgrade Control Register*

The remote system upgrade control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. In remote update mode, the control register page address bits are set to all zeros (7'b0 = 0000_000) at power up in order to load the factory configuration. However, in local update mode the control register page address bits power up as (7'b1 = 0000_001) in order to select the application configuration. Additionally, the control register cannot be updated in local update mode, whereas a factory configuration in remote update mode has write access to this register.

The control register bit positions are shown in Figure 14–7 and defined in Table 14–4. In the figure, the numbers show the bit position of a setting within a register. For example, bit number 8 is the enable bit for the watchdog timer.

*Figure 14–7. Remote System Upgrade Control Register*

| 20 19 18 17 16 15 14 13 12 11 10 9 | 8 | 7 6 5 4 | 3 2 1 | 0 |
|---|---|---|---|---|
| Wd_timer[11..0] | Wd_en | PGM[6..3] | PGM[2..0] | AnF |

The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix II or Stratix II GX device is the factory configuration or an application configuration. This bit is set high at power up in local update mode, and is set low by the remote system upgrade circuitry when an error condition causes a fall-back to factory configuration. When the AnF bit is high, the control register access is limited to read operations. When the AnF bit is low, the register allows write operations and disables the watchdog timer.

☞ In remote update mode, factory configuration design should set this bit high (1'b1) when updating the contents of the update register with application page address and watchdog timer settings.

*Table 14–4. Remote System Upgrade Control Register Contents*

| Control Register Bit | Remote System Upgrade Mode | Value | Definition |
|---|---|---|---|
| AnF *(1)* | Local update<br>Remote update | 1'b1<br>1'b0 | Application not factory |
| PGM[2..0] | Local update<br>Remote update (FPP, PS, PPA) | 3'b001<br>3'b000 | Page mode select |
| | Remote update (AS) | 3'b000 | AS configuration start address (StAdd[18..16]) |
| PGM[6..3] | Local update<br>Remote update (FPP, PS, PPA) | 4'b0000<br>4'b0000 | Not used |
| | Remote update (AS) | 4'b0000 | AS configuration start address (StAdd[22..19]) |
| Wd_en | Remote update | 1'b0 | User watchdog timer enable bit |
| Wd_timer[11..0] | Remote update | 12'b000000000000 | User watchdog time-out value (most significant 12 bits of 29-bit count value: {Wd_timer[11..0], 17'b0}) |

*Note to Table 14–4:*

(1)  In remote update mode, the remote configuration block does not update the AnF bit automatically (you can update it manually). In local update mode, the remote configuration updates the AnF bit with 0 in the factory page and 1 in the application page.

### Remote System Upgrade Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition. The various trigger and error conditions include:

- CRC (cyclic redundancy check) error during application configuration
- nSTATUS assertion by an external device due to an error
- FPGA logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (nCONFIG) assertion
- User watchdog timer time out

Figure 14–8 and Table 14–5 specify the contents of the status register. The numbers in the figure show the bit positions within a 5-bit register.

*Figure 14–8. Remote System Upgrade Status Register*

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| Wd | nCONFIG | Core_nCONFIG | nSTATUS | CRC |

*Table 14–5. Remote System Upgrade Status Register Contents*

| Status Register Bit | Definition | POR Reset Value |
|---|---|---|
| CRC (from configuration) | CRC error caused reconfiguration | 1 bit '0' |
| nSTATUS | nSTATUS caused reconfiguration | 1 bit '0' |
| CORE *(1)* CORE_nCONFIG | Device logic array caused reconfiguration | 1 bit '0' |
| nCONFIG | nCONFIG caused reconfiguration | 1 bit '0' |
| Wd | Watchdog timer caused reconfiguration | 1 bit '0' |

*Note to Table 14–5:*
(1) Logic array reconfiguration forces the system to load the application configuration data into the Stratix II or Stratix II GX device. This occurs after the factory configuration specifies the appropriate application configuration page address by updating the update register.

## Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (see Table 14–3 on page 14–15). While both registers can only be updated when the FPGA is loaded with a factory configuration image, the update register writes are controlled by the user logic, and the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic should send the AnF bit (set high), the page address, and watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (RU_nCONFIG) goes high, the remote system upgrade state machine updates the control register with the contents of the update register and initiates system reconfiguration from the new application page.

In the event of an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (page zero or page one, based on mode and error condition) by setting the control register accordingly. Table 14–6 lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition but before the factory configuration is loaded.

*Table 14–6. Control Register Contents After an Error or Reconfiguration Trigger Condition*

| Reconfiguration Error/Trigger | Control Register Setting | |
|---|---|---|
| | Remote Update | Local Update |
| nCONFIG reset | All bits are 0 | PGM[6..0] = 7'b0000001<br>AnF = 1<br>All other bits are 0 |
| nSTATUS error | All bits are 0 | All bits are 0 |
| CORE triggered reconfiguration | Update register | PGM[6..0] = 7'b0000001<br>AnF = 1<br>All other bits are 0 |
| CRC error | All bits are 0 | All bits are 0 |
| Wd time out | All bits are 0 | All bits are 0 |

Read operations during factory configuration access the contents of the update register. This feature is used by the user logic to verify that the page address and watchdog timer settings were written correctly. Read operations in application configurations access the contents of the control register. This information is used by the user logic in the application configuration.

## User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the FPGA.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29-bits-wide and has a maximum count value of $2^{29}$. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is $2^{15}$ cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator. Table 14–7 specifies the operating range of the 10-MHz internal oscillator.

| *Table 14–7. 10-MHz Internal Oscillator Specifications* | | | |
|:---:|:---:|:---:|:---:|
| **Minimum** | **Typical** | **Maximum** | **Units** |
| 5 | 6.5 | 10 | MHz |

The user watchdog timer begins counting once the application configuration enters FPGA user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting RU_nRSTIMER. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (Wd) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

The user watchdog timer is not enabled during the configuration cycle of the FPGA. Errors during configuration are detected by the CRC engine. Also, the timer is disabled for factory configurations. Functional errors should not exist in the factory configuration since it is stored and validated during production and is never updated remotely.

☞ The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode.

## Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array

The dedicated remote system upgrade circuitry drives (or receives) seven signals to (or from) the FPGA logic array. The FPGA logic array uses these signals to read and write the remote system upgrade control, status, and update registers using the remote system upgrade shift register. Table 14–8 lists each of these seven signals and describes their functionality.

Except for RU_nRSTIMER and RU_CAPTnUPDT, the logic array signals are enabled for both remote and local update modes and for both factory and application configurations. RU_nRSTIMER is only valid for application configurations in remote update mode, since local update configurations and factory configurations have the user watchdog timer disabled. When RU_CAPTnUPDT is low, the device can write to the update register only for factory configurations in remote update mode, since this is the only case where the update register is written to by the user logic. When the RU_nCONFIG signal goes high, the contents of the update register are written into the control register for controlling the next configuration cycle.

*Table 14–8. Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array  (Part 1 of 3)*

| Signal Name | Signal Direction | Description |
|---|---|---|
| RU_nRSTIMER | Input to remote system upgrade block (driven by FPGA logic array) | Request from the application configuration to reset the user watchdog timer with its initial count. A falling edge of this signal triggers a reset of the user watchdog timer. |
| RU_nCONFIG | Input to remote system upgrade block (driven by FPGA logic array) | When driven low, this signal triggers the device to reconfigure.<br><br>If asserted by the factory configuration in remote update mode, the application configuration specified in the remote update control register is loaded. If requested by the application configuration in remote update mode, the factory configuration is loaded.<br><br>In the local updated mode, the application configuration is loaded whenever this signal is asserted. |

*Table 14–8. Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array  (Part 2 of 3)*

| Signal Name | Signal Direction | Description |
|---|---|---|
| RU_CLK | Input to remote system upgrade block (driven by FPGA logic array) | Clocks the remote system upgrade shift register and update register so that the contents of the status, control, and update registers can be read, and so that the contents of the update register can be loaded. The shift register latches data on the rising edge of this clock signal. |
| RU_SHIFTnLD | Input to remote system upgrade block (driven by FPGA logic array) | This pin determines if the shift register contents are shifted over during the next clock edge or loaded in/out.<br><br>When this signal is driven high (1'b1), the remote system upgrade shift register shifts data left on each rising edge of RU_CLK.<br><br>When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven low (1'b0), the remote system upgrade update register is updated with the contents of the shift register on the rising edge of RU_CLK.<br><br>When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven high (1'b1), the remote system upgrade shift register captures the status register and either the control or update register (depending on whether the current configuration is application or factory, respectively) on the rising edge of RU_CLK. |
| RU_CAPTnUPDT | Input to remote system upgrade block (driven by FPGA logic array) | This pin determines if the contents of the shift register are captured or updated on the next clock edge.<br><br>When the RU_SHIFTnLD signal is driven high (1'b1), this input signal has no function.<br><br>When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven high (1'b1), the remote system upgrade shift register captures the status register and either the control or update register (depending on whether the current configuration is application or factory, respectively) on the rising edge of RU_CLK.<br><br>When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven low (1'b0), the remote system upgrade update register is updated with the contents of the shift register on the rising edge of RU_CLK.<br><br>In local update mode, a low input on RU_CAPTnUPDT has no function, because the update register cannot be updated in this mode. |

*Table 14–8. Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array  (Part 3 of 3)*

| Signal Name | Signal Direction | Description |
|-------------|------------------|-------------|
| RU_DIN | Input to remote system upgrade block (driven by FPGA logic array) | Data to be written to the remote system upgrade shift register on the rising edge of RU_CLK. To load data into the shift register, RU_SHIFTnLD must be asserted. |
| RU_DOUT | Output from remote system upgrade block (driven to FPGA logic array) | Output data from the remote system upgrade shift register to be read by logic array logic. New data arrives on each rising edge of RU_CLK. |

## Remote System Upgrade Pin Descriptions

Table 14–9 describes the dedicated remote system upgrade configuration pins.

For descriptions of all the configuration pins, refer to the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook* and the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*.

| Table 14–9. Stratix II and Stratix II GX Remote System Upgrade Pins | | | | |
|---|---|---|---|---|
| Pin Name | User Mode | Configuration Scheme | Pin Type | Description |
| RUnLU | N/A if using remote system upgrade in FPP, PS, AS, or PPA modes. I/O if not using these modes. | Remote configuration in FPP, PS, or PPA | Input | Input that selects between remote update and local update. A logic high (1.5-V, 1.8-V, 2.5-V, 3.3-V) selects remote update, and a logic low selects local update. When not using remote update or local update configuration modes, this pin is available as a general-purpose user I/O pin. When using remote configuration in AS mode, set the RUnLU pin to high because AS does not support local update. |
| PGM[2..0] | N/A if using remote system upgrade in FPP, PS, or PPA modes. I/O if not using these modes. | Remote configuration in FPP, PS or PPA | Output | These output pins select one of eight pages in the memory (either flash or enhanced configuration device) when using remote update mode. When not using remote update or local update configuration modes, these pins are available as general-purpose user I/O pins. |

## Quartus II Software Support

Implementation in your design requires an remote system upgrade interface between the FPGA logic array and remote system upgrade circuitry. You also need to generate configuration files for production and remote programming of the system configuration memory. The Quartus® II software provides these features.

The two implementation options, altremote_update megafunction and remote system upgrade atom, are for the interface between the remote system upgrade circuitry and the FPGA logic array interface.

### altremote_update Megafunction

The altremote_update megafunction provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read/write protocol in FPGA logic. This implementation is suitable for designs that implement the factory configuration functions using a Nios processor in the FPGA.

Tables 14–10 and 14–11 describe the input and output ports available on the `altremote_update` megafunction. Table 14–12 shows the `param[2..0]` bit settings.

| *Table 14–10. Input Ports of the altremote_update Megafunction    (Part 1 of 2)* | | | |
|---|---|---|---|
| **Port Name** | **Required** | **Source** | **Description** |
| clock | Y | Logic Array | Clock input to the `altremote_update` block. All operations are performed with respects to the rising edge of this clock. |
| reset | Y | Logic Array | Asynchronous reset, which is used to initialize the remote update block. To ensure proper operation, the remote update block must be reset before first accessing the remote update block. This signal is not affected by the busy signal and will reset the remote update block even if busy is logic high. This means that if the reset signal is driven logic high during writing of a parameter, the parameter will not be properly written to the remote update block. |
| reconfig | Y | Logic Array | When driven logic high, reconfiguration of the device is initiated using the current parameter settings in the remote update block. If busy is asserted, this signal is ignored. This is to ensure all parameters are completely written before reconfiguration begins. |
| reset_timer | N | Logic Array | This signal is required if you are using the watchdog timer feature. A logic high resets the internal watchdog timer. This signal is not affected by the busy signal and can reset the timer even when the remote update block is busy. If this port is left connected, the default value is 0. |
| read_param | N | Logic Array | Once `read_param` is sampled as a logic high, the busy signal is asserted. While the parameter is being read, the busy signal remains asserted, and inputs on `param[]` are ignored. Once the busy signal is deactivated, the next parameter can be read. If this port is left unconnected, the default value is 0. |
| write_param | N | Logic Array | This signal is required if you intend on writing parameters to the remote update block. When driven logic high, the parameter specified on the `param[]` port should be written to the remote update block with the value on `data_in[]`. The number of valid bits on `data_in[]` is dependent on the parameter type. This signal is sampled on the rising edge of clock and should only be asserted for one clock cycle to prevent the parameter from being re-read on subsequent clock cycles. Once `write_param` is sampled as a logic high, the busy signal is asserted. While the parameter is being written, the busy signal remains asserted, and inputs on `param[]` and `data_in[]` are ignored. Once the busy signal is deactivated, the next parameter can be written. This signal is only valid when the `Current_Configuration` parameter is factory since parameters cannot be written in application configurations. If this port is left unconnected, the default value is 0. |

*Table 14–10. Input Ports of the altremote_update Megafunction   (Part 2 of 2)*

| Port Name | Required | Source | Description |
|---|---|---|---|
| `param[2..0]` | N | Logic Array | 3-bit bus that selects which parameter should be read or written. If this port is left unconnected, the default value is 0. |
| `data_in[11..0]` | N | Logic Array | This signal is required if you intend on writing parameters to the remote update block 12-bit bus used when writing parameters, which specifies the parameter value. The parameter value is requested using the `param[]` input and by driving the `write_param` signal logic high, at which point the busy signal goes logic high and the value of the parameter is captured from this bus. For some parameters, not all 12 bits are used, in which case only the least significant bits are used. This port is ignored if the `Current_Configuration` parameter is set to an application configuration since writing of parameters is only allowed in the factory configuration. If this port is left unconnected, the default value is 0. |

*Note to Table 14–10:*
(1)   Logic array source means that you can drive the port from internal logic or any general-purpose I/O pin.

*Table 14–11. Output Ports of the altremote_update Megafunction  (Part 1 of 2)*

| Port Name | Required | Destination | Description |
|---|---|---|---|
| `busy` | Y | Logic Array | When this signal is a logic high, the remote update block is busy either reading or writing a parameter. When the remote update block is busy, it ignores its `data_in[]`, `param[]`, and `reconfig` inputs. This signal goes high when `read_param` or `write_param` is asserted and remains asserted until the operation is complete. |
| `pgm_out[2..0]` | Y | `PGM[2..0]` pins | 3-bit bus that specifies the page pointer of the configuration data to be loaded when the device is reconfigured. This port must be connected to the `PGM[]` output pins, which should be connected to the external configuration device. |

**Table 14–11. Output Ports of the altremote_update Megafunction  (Part 2 of 2)**

| Port Name | Required | Destination | Description |
|---|---|---|---|
| `data_out[11..0]` | N | Logic Array | 12-bit bus used when reading parameters, which reads out the parameter value. The parameter value is requested using the `param[]` input and by driving the `read_param` signal logic high, at which point the busy signal goes logic high. When the busy signal goes low, the value of the parameter is driven out on this bus. The `data_out[]` port is only valid after a `read_param` has been issued and once the busy signal is deasserted. At any other time, its output values are invalid. For example, even though the `data_out[]` port may toggle during a writing of a parameter, these values are not a valid representation of what was actually written to the remote update block. For some parameters, not all 12 bits are used, in which case only the least significant bits are used. |

*Note to Table 14–11:*
(1)     Logic array destination means that you can drive the port to internal logic or any general-purpose I/O pin.

**Table 14–12. Parameter Settings for the altremote_update Megafunction    (Part 1 of 2)**

| Selected Parameter | param[2..0] Bit Setting | Width of Parameter Value | POR Reset Value | Description |
|---|---|---|---|---|
| Status Register Contents | 000 | 5 | 5 bit '0' | Specifies the reason for re-configuration, which could be caused by a CRC error during configuration, nSTATUS being pulled low due to an error, the device core caused an error, nCONFIG pulled low, or the watchdog timer timed-out. This parameter can only be read. |
| Watchdog Timeout Value | 010 | 12 | 12 bits '0' | User watchdog timer time-out value. Writing of this parameter is only allowed when in the factory configuration. |
| Watchdog Enable | 011 | 1 | 1 bit '0' | User watchdog timer enable. Writing of this parameter is only allowed when in the factory configuration |
| Page select | 100 | 3 (FPP, PS, PPA) | 3 bit '001' - Local configuration | Page mode selection. Writing of this parameter is only allowed when in the factory configuration. |
| | | | 3 bit '000' - Remote configuration | |
| | | 7 (AS) | 7 bit '0000000' - Remote configuration | |

*Table 14–12. Parameter Settings for the altremote_update Megafunction (Part 2 of 2)*

| Selected Parameter | param[2..0] Bit Setting | Width of Parameter Value | POR Reset Value | Description |
|---|---|---|---|---|
| Current configuration (AnF) | 101 | 1 | 1 bit '0' - Factory | Specifies whether the current configuration is factory or and application configuration. This parameter can only be read. |
| | | | 1 bit '1' - Application | |
| Illegal values | 001 | | | |
| | 110 | | | |
| | 111 | | | |

### Remote System Upgrade Atom

The remote system upgrade atom is a WYSIWYG atom or primitive that can be instantiated in your design. The primitive is used to access the remote system upgrade shift register, logic array reset, and watchdog timer reset signals. The ports on this primitive are the same as those listed in Table 14–8. This implementation is suitable for designs that implement the factory configuration functions using state machines (without a processor).

## System Design Guidelines

The following general guidelines are applicable when implementing remote system upgrade in Stratix II and Stratix II GX FPGAs. Guidelines for specific configuration schemes are also discussed in this section.

- After downloading a new application configuration, the soft logic implemented in the FPGA can validate the integrity of the data received over the remote communication interface. This optional step helps avoid configuration attempts with bad or incomplete configuration data. However, in the event that bad or incomplete configuration data is sent to the FPGA, it detects the data corruption using the CRC signature attached to each configuration frame.
- The auto-reconfigure on configuration error option bit is ignored when remote system upgrade is enabled in your system. This option is always enabled in remote configuration designs, allowing your system to return to the safe factory configuration in the event of an application configuration error or user watchdog timer time out.

## Remote System Upgrade With Serial Configuration Devices

Remote system upgrade support in the AS configuration scheme is similar to support in other schemes, with the following exceptions:

■ The remote system upgrade block provides the AS configuration controller inside the Stratix II or Stratix II GX FPGA with a 7-bit page start address (PGM[6..0]) instead of driving the 3-bit page mode pins (PGM[2..0]) used in FPP, PS, and PPA configuration schemes. This 7-bit address forms the 24-bit configuration start address (StAdd[23..0]). Table 14–13 illustrates the start address generation using the page address registers.

■ The configuration start address for factory configuration is always set to 24'b0.

■ PGM[2..0] pins on Stratix II devices are not used in AS configuration schemes and can not be used as regular I/O pins.

■ The Nios ASMI peripheral can be used to update configuration data within the serial configuration device.

*Table 14–13. AS Configuration Start Address Generation*

| Serial Configuration Device | Serial Configuration Device Density (MB) | Add[23] | PGM[6..0] (Add[22..16]) | Add[15..0] |
|---|---|---|---|---|
| EPCS64 | 64 | 0 | MSB[6..0] | All 0s |
| EPCS16 | 16 | 0 | 00, MSB[4..0] | All 0s |
| EPCS4 | 4 | 0 | 0000, MSB[2..0] | All 0s |

## Remote System Upgrade With a MAX II Device or Microprocessor and Flash Device

This setup requires the MAX II device or microprocessor to support page addressing. MAX II or microprocessor devices implementing remote system upgrade should emulate the enhanced configuration device page mode feature. The PGM[2..0] output pins from the Stratix II or Stratix II GX device must be sampled to determine which configuration image is to be loaded into the FPGA.

If the FPGA does not release CONF_DONE after all data has been sent, the MAX II microprocessor should reset the FPGA back to the factory image by pulsing its nSTATUS pin low.

The MAX II device or microprocessor and flash configuration can use FPP, PS, or PPA. Decompression and design security features are supported in the FPP (requires 4× DCLK) and PS modes only. Figure 14–9 shows a system block diagram for remote system upgrade with the MAX II device or microprocessor and flash.

*Figure 14–9. System Block Diagram for Remote System Upgrade With MAX II Device or Microprocessor and Flash Device*
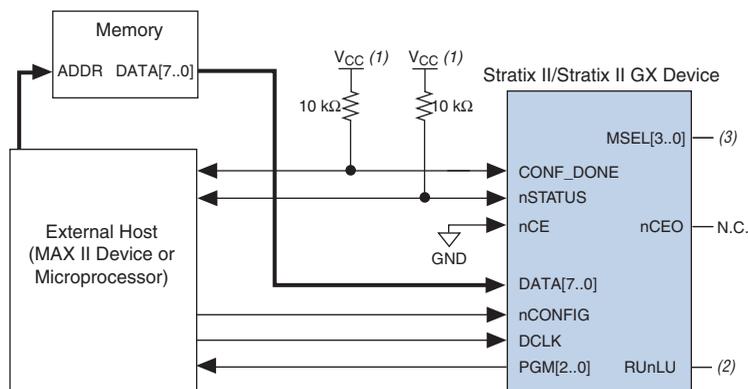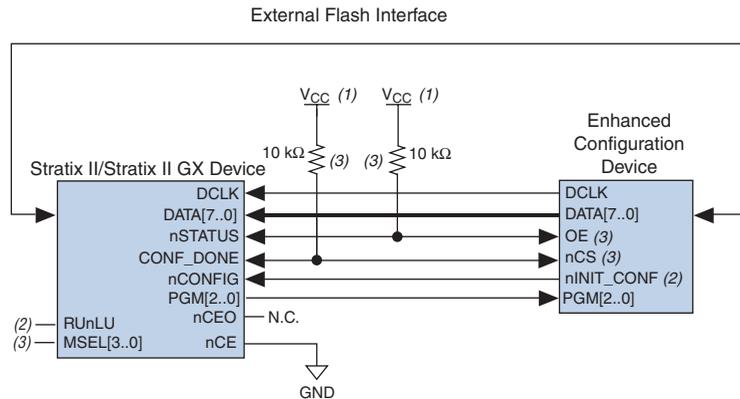


*Notes to Figure 14–9:*
(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device.
(2) Connect RUnLU to GND or $V_{CC}$ to select between remote and local update modes.
(3) Connect MSEL[3..0] to 0100 to enable remote update remote system upgrade mode.

## Remote System Upgrade with Enhanced Configuration Devices

■ Enhanced Configuration devices support remote system upgrade with FPP or PS configuration schemes. The Stratix II or Stratix II GX decompression and design security features are only supported in the PS mode. The enhanced configuration device's decompression feature is supported in both PS and FPP schemes.

■ In remote update mode, neither the factory configuration nor the application configurations should alter the enhanced configuration device's option bits or the page 000 factory configuration data. This ensures that an error during remote update can always be resolved by reverting to the factory configuration located at page 000.

■ The enhanced configuration device features an error checking mechanism to detect instances when the FPGA fails to detect the configuration preamble. In these instances, the enhanced configuration device pulses the nSTATUS signal low, and the remote system upgrade circuitry attempts to load the factory configuration. Figure 14–10 shows a system block diagram for remote system upgrade with enhanced configuration devices.

*Figure 14–10. System Block Diagram for Remote System Upgrade with Enhanced Configuration Devices*



*Notes to Figure 14–10:*
(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device.
(2) Connect RUnLU to GND or $V_{CC}$ to select between remote and local update modes.
(3) Connect MSEL[3..0] to 0100 to enable remote update remote system upgrade mode.

## Conclusion

Stratix II and Stratix II GX devices offer remote system upgrade capability, where you can upgrade a system in real-time through any network. Remote system upgrade helps to deliver feature enhancements and bug fixes without costly recalls, reduces time to market, and extends product life cycles. The dedicated remote system upgrade circuitry in Stratix II and Stratix II GX devices provides error detection, recovery, and status information to ensure reliable reconfiguration.

## Referenced Documents

This chapter references the following documents:

■ *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook*

■ *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*
■ *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* chapter in volume 2 of the *Configuration Handbook*
■ *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook*

# Document Revision History

Table 14–14 shows the revision history for this chapter.

| Table 14–14. Document Revision History | | |
|---|---|---|
| **Date and Document Version** | **Changes Made** | **Summary of Changes** |
| October 2007, v4.5 | Updated Table 14–7. | — |
| | Added the "Referenced Documents" section. | — |
| | Minor text edits. | — |
| No change | For the *Stratix II GX Device Handbook* only: Formerly chapter 13. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change. | — |
| May 2007, v4.4 | Updated note to "Functional Description" section. | — |
| | Minor text edit to "Remote System Upgrade With Serial Configuration Devices" section. | — |
| February 2007 v4.3 | Added the "Document Revision History" section to this chapter. | — |
| April 2006, v4.2 | Chapter updated as part of the *Stratix II Device Handbook* update. | — |
| No change | Formerly chapter 12. Chapter number change only due to chapter addition to Section I in February 2006; no content change. | — |
| December 2005, v4.1 | Chapter updated as part of the *Stratix II Device Handbook* update. | — |
| October 2005 v4.0 | Added chapter to the *Stratix II GX Device Handbook*. | — |