Using the Nios Development Board Configuration Controller Reference Designs

July 2006 - ver 1.1 Application Note 346

Introduction

Many modern embedded systems utilize flash memory to store processor configuration information and program data for the system processor. Increasingly, flash memory is also being used to store hardware configuration data for field programmable gate arrays (FPGAs) located in the embedded system. The use of flash memory for both processor memory and FPGA configuration data typically requires an intelligent configuration controller, particularly for systems with soft embedded processors such as the Altera® Nios II® processor solution.

Altera's Nios II development boards, Stratix II™ and Cyclone II™ editions, contain a programmable configuration controller that provides a stable platform for managing system configuration and reset in embedded systems. The Nios II development board configuration controller is implemented in an Altera EPM7256AE non-volatile MAX® 7000 device. The controller platform provides instant-on configuration control—enabling systems to be configured, or reconfigured, in a minimal amount of time.

This application note:

- Describes how to use configuration controller reference designs for systems containing Altera Stratix II or Cyclone II devices
- Provides the steps to port reference designs to your board
- Provides suggestions for modifying the reference designs to support more complex systems

For more information about the Nios II embedded processor see: *The Nios II Processor Reference Handbook* and the *Nios II Software Developer's Handbook*.

For more information about the Nios II development boards, see the: *Nios Development Board Cyclone II Edition, Reference Manual* and the *Nios Development Board Stratix II Edition, Reference Manual.*

Configuration Controller Overview

The configuration controller on the Nios development boards is designed to:

- Intelligently configure FPGAs from a flash memory device on system power-up

- Utilize a single flash memory device to store hardware configuration data as well as embedded software instructions and data
- Provide flash partitions for multiple hardware images
- Provide a mechanism for users to reset the board to a known working hardware image
- Provide an interface to a Nios II processor residing on an FPGA so that the processor can issue a reconfiguration request to the configuration controller; facilitating in-the-field upgrades

This section discusses:

- Flash images on the Nios development board, Stratix II & Cyclone II editions
- Configuration controller boot sequences for Stratix II & Cyclone II editions
- Configuration controller hardware for Stratix II & Cyclone II editions
- Configuration controller reference designs for Stratix II & Cyclone II editions

Flash Images on the Nios Development Board, Stratix II Edition

The Nios development board, Stratix II edition includes an Altera EPCS64 serial configuration device as well as an 8-bit wide Common Flash Interface (CFI) flash. The CFI flash image on the Nios development board, Stratix II edition is partitioned into three sections, see Figure 1 on page 3.

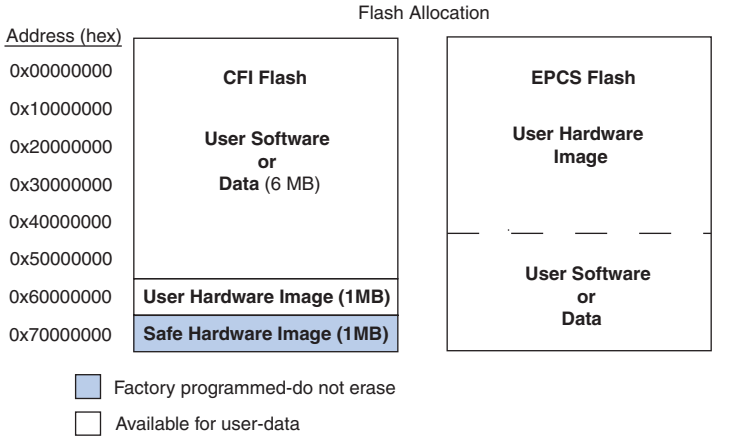
In the CFI flash:

- The first section is used for general programming or data storage.
- The other two sections—user and safe image—are used to store FPGA hardware configurations
  - The user image typically contains a hardware design that is under development or testing
  - The safe (factory programmed) image is assumed to contain a known working or default hardware image

The EPCS64 is divided into two sections with a floating section boundary. The section boundary between the User Hardware and User Software/Data sections is determined by the size of the User Hardware section.

It is the configuration controller's responsibility to load the EPCS user, CFI user or the CFI safe image into the Stratix II device. By observing the behavior illustrated in the Figure 2 flowchart, the configuration controller determines which image to load.

Figure 1. Nios Development Board Flash Partition



Configuration Controller Boot Sequence for Stratix II Devices

When the Nios development board, Stratix II edition initially powers-on, the configuration controller is idle until it detects the power-on reset. Upon detecting power-on reset, the controller attempts to load the EPCS user image into the FPGA.

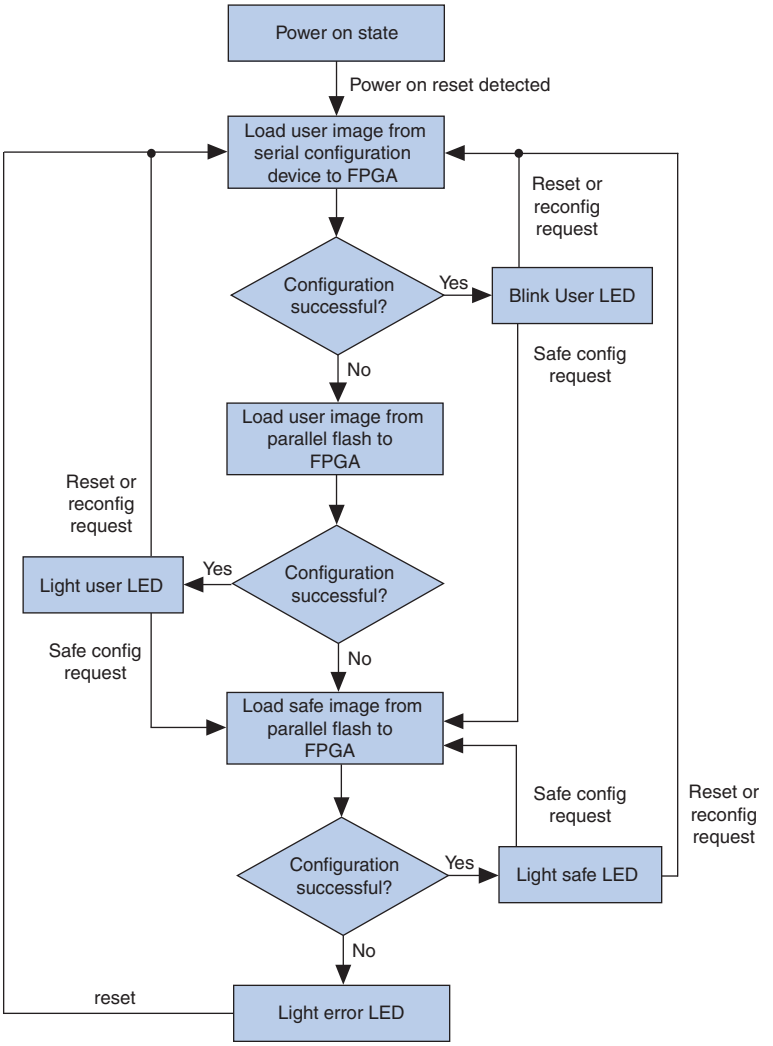
If configuration of the EPCS user flash image is successful, the controller will continually flash the **User LED**, tri-state all of its flash interface pins, and remain idle until a reset, reconfiguration, or safe image request is detected.

If configuration of the user flash image fails, the controller will attempt to program the safe flash image into the Stratix II device. If the controller can successfully program the safe flash image into the Stratix II device, it will illuminate the **Safe LED**. The controller remains idle until a reset, reconfiguration, or safe image request is detected.

If an error occurs while programming the safe image to the device, the controller will illuminate the **Error LED** and remain in an error state. The controller will only exit the error state when a reset or power-on reset request is detected.

Figure 2 shows the configuration controller boot sequence flowchart for Stratix II devices.

Figure 2. Configuration Controller Boot Sequence Flowchart



For more information on the configuration process and the flash structure of the Nios development board, Stratix II edition, refer to: *Nios Development Board Stratix II Edition, Reference Manual*.

Configuration Controller Hardware for Stratix II Devices

Stratix II FPGAs can be programmed in a number of different modes; however, when programming a Stratix II device via an external configuration controller, it is recommended to program the device in fast passive parallel (FPP) mode. When in FPP mode, the external configuration controller generates a configuration clock for the Stratix II device and presents an 8-bit configuration data word to the Stratix II device on each configuration clock cycle.

For more information on the FPP configuration capabilities of Stratix II devices refer to the *Stratix II Device Handbook, Chapter 13: Configuring Stratix II & Stratix II GX Devices*.

The configuration controller reference design included with the Nios II Embedded Design Suite (EDS) uses an EPM7256AE device to generate the configuration clock for the Stratix II device as well as generate the address and control signals necessary to pull configuration data out of flash memory. Figure 3 shows the development board configuration controller block diagram for Stratix II devices.

Figure 3. Development Board Configuration Controller for Stratix II Devices

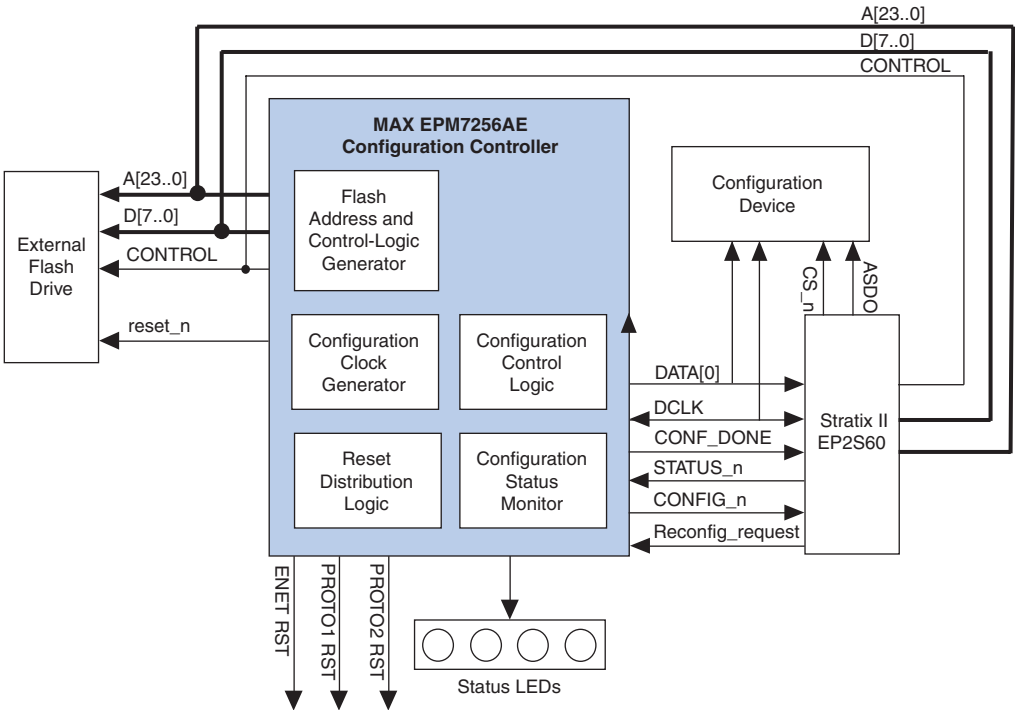


Figure 3 illustrates that the configuration controller can be conceptually viewed as five sub-blocks, which are described in this section.

Flash Address & Control Logic Generator

The flash address and control logic block generates the address signals to send to the CFI flash device. The logic will initiate configuration from either the user or factory image based on inputs from the control logic, and then increment the CFI flash address bus until the appropriate number of configuration bytes have been transferred, or an error is detected. After a successful configuration, the CFI flash address and control pins of the EPM7256AE device are tri-stated so that the flash can be accessed by the Nios II embedded processor running in the Stratix II device.

Configuration Clock Generator

The configuration clock generator drives the DCLK input signal of the Stratix II device. The clock is derived from the 50 MHz oscillator on the Stratix II development board. The reference design divides the oscillator by 16 to create a 3.125MHz configuration clock. The clock is also used to time the address and control signals of the CFI flash device.

Reset Distribution Logic

The configuration controller is also used to deliver resets to other devices on the development board. A board level reset is issued to the Ethernet device, the flash device, and the two, board-prototype headers when:

- A power-on reset occurs.
- A reconfiguration request from the processor arrives.
- The force safe button is pressed.

Configuration Control Logic

To control the configuration flow, the configuration control logic monitors the various board reset sources as well as the Stratix II device configuration status signals. This logic block drives the MSEL pins on the Stratix II device to set the configuration mode based on the flow chart shown in Figure 2 on page 4. The logic manages the loading of the user or factory images based on the status of the Stratix II devices STATUS_n and CONF_DONE signals. The configuration control logic is also responsible for initiating a new configuration if a reset_n is detected, the board's **Force Safe** button has been pressed, or the Nios II processor residing in the Stratix II device issues a Reconfiguration_request.

Configuration Status Monitor

The Nios development board, Stratix II edition contains four LEDs used to indicate the configuration status of the processor. LEDs are illuminated depending on which image is loaded into the Stratix II device or if an error has occurred during configuration.

Configuration Controller Reference Designs for Stratix II Devices

The Nios II EDS provides configuration controller reference design for the EP2S60 device. If you have installed the Nios II EDS, you will find the source code and Quartus® II project files for the reference design at:

- **\kits\nios2_60\examples\config_controller\niosII_stratixII_2s60_rohs**

Table 1 provide a description of each of the source files used in the configuration controller design.

Table 1. Stratix II Configuration Controller Files

File Name	Description
config_controller.tdf	Top-level design. Contains control logic and instantiations of the lower-level files, i.e., **address_counter.tdf**, **dclk_divider.tdf**, **reset_counter.tdf**.
address_counter.tdf	Counter used to generate address bus for flash device access.
dclk_divider.tdf	Clock divider used to generate `DCLK` from the board oscillator.
reset_counter.tdf	Used to debounce reset switch inputs.

Flash Image on the Nios Development Board, Cyclone II Edition

The flash image on the Nios development board, Cyclone II edition also contains a user and safe (factory programmed) image in CFI flash as well as a user hardware image in an EPCS64 serial configuration device.

Configuration Controller Boot Sequence for Cyclone II Devices

When the Nios development board, Cyclone II edition initially powers-on, the configuration controller is idle until it detects the power-on reset. Upon detecting power-on reset, the controller attempts to load the image stored in serial flash into the FPGA.

If configuration of the serial flash image is successful the controller continuously blinks the **User LED** on the development board, tri-states all of its flash interface pins, and remains idle until a reset or reconfiguration request is detected.

If configuration of the serial flash image fails, the controller will attempt to program the user image from the 8-bit flash into the Cyclone II device. In this case, the **User LED** will illuminate if the controller can successfully program the user image into the Cyclone II device and the controller will remain idle until a reset, reconfiguration request, or safe image reset is detected.

If configuration of the user image fails, the controller will attempt to load the safe image from the 8-bit flash into the Cyclone II device. The **Safe LED** is illuminated when the safe image is properly loaded. If an error

occurs while programming the safe image to the device, the controller will illuminate the **Error LED** and remain in an error state. The controller will only exit the error state when a reset or power-on reset is issued.

The configuration controller boot sequence flowchart for the Nios development board, Cyclone II edition is shown in Figure 2 on page 4.

For more information on the configuration process and the flash structure of the Nios development board, Cyclone II edition, refer to: *Nios Development Board Reference Manual, Cyclone II Edition*.

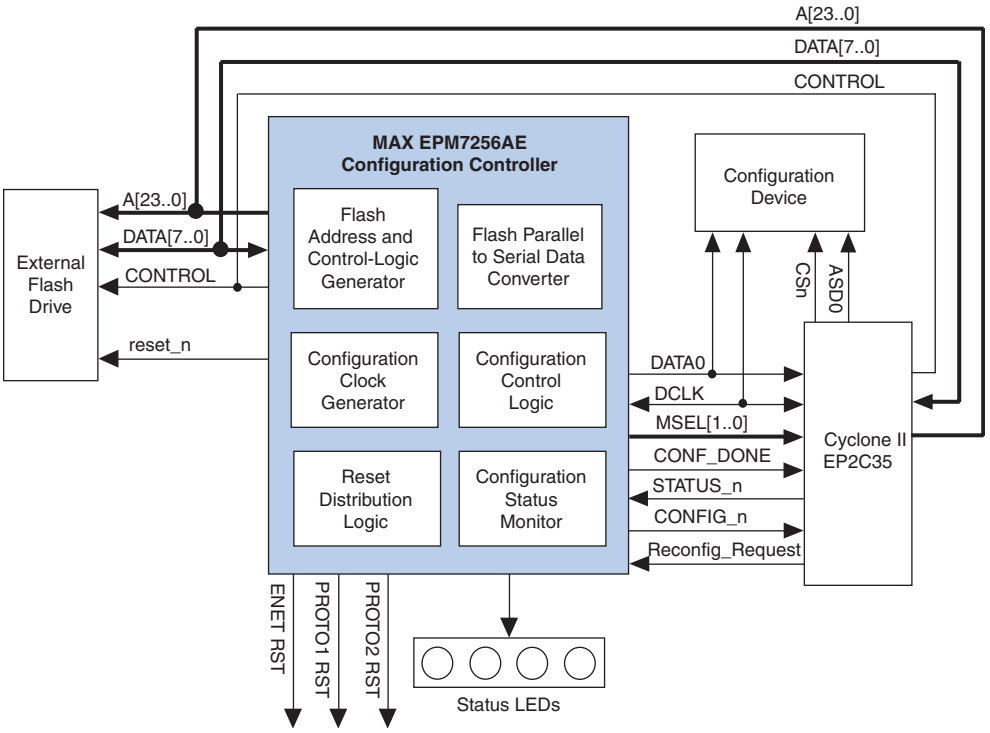
Configuration Controller Hardware for Cyclone II Devices

Cyclone II FPGAs are programmed serially via either a download cable, a serial configuration device, or in passive serial (PS) mode using an external configuration controller. The configuration controller reference design demonstrates the flexibility of using a CPLD as a configuration controller in that the same CPLD can be used as a configuration controller for two different Altera device families. In the case of the Nios development board, Cyclone II edition, the configuration controller has been modified to enable programming the Cyclone II device in PS mode using an 8-bit flash as the configuration source as well as from an Altera serial configuration device.

For more information on the configuration capabilities of Cyclone II devices refer to the *Cyclone II Device Handbook, Chapter 13: Configuring Cyclone II FPGAs*.

The configuration controller reference design included with the Nios development board, Cyclone II edition, uses a MAX 7256AE device to generate the configuration clock for the Cyclone II device as well as generate the address and control signals necessary to pull configuration data out of flash memory and convert it to serial data for the Cyclone II device. Figure 4 illustrates a block diagram of the configuration controller reference design for Cyclone II devices.

Figure 4. Development Board Configuration Controller for Cyclone II Devices



The main differences between the Cyclone II edition development board's configuration controllers and the Stratix II edition development board's configuration controllers are that the Cyclone II edition includes:

- A parallel to serial converter block
- Control logic block modifications

The parallel to serial block reads an 8-bit word from the flash image every 8 DCLK cycles and shifts the word out to the DATA0 input of the Cyclone II device on every DCLK cycle.

Configuration Controller Reference Designs for Cyclone II Devices

The Nios Development Kit, Cyclone II Edition provides configuration controller reference design for the Cyclone II EP2C35 device. If you have installed the Nios Development Kit, you will find the source code and Quartus II project files for the reference designs at:

- **\kits\nios2_60\examples\config_controller\niosII_cycloneII_2c35**

Table 2 provides a description of each of the source files used in the configuration controller design.

Table 2. Cyclone II Configuration Controller Files

File Name	Description
config_controller.tdf	Top-level design. Contains control logic and instantiations of the lower-level files, i.e., **address_counter.tdf**, **dclk_divider.tdf**, **reset_counter.tdf**, **data_bit_counter.tdf**, **shift_register.tdf**.
address_counter.tdf	Counter used to generate address bus for flash access.
dclk_divider.tdf	Clock divider used to generate `DCLK` from the board oscillator.
reset_counter.tdf	Used to debounce reset switch inputs.
data_bit_counter.tdf	Used to determine when a word should be loaded into the shift register.
shift_register.tdf	Shifts a configuration bit out to the `DATA0` line each `DCLK` cycle.

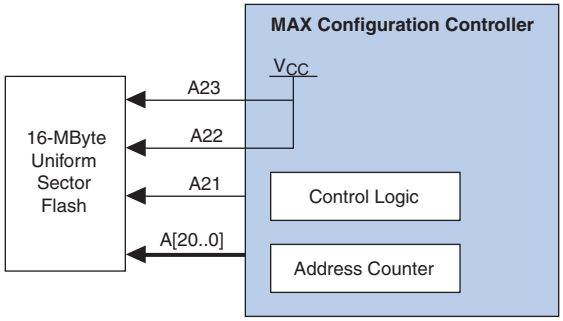
Porting the Configuration Controller Reference Designs to Other Boards

The configuration controller reference design included with Nios development kits can easily be adapted to work on other boards that contain a CPLD and an Altera FPGA. In general, the only required modification is to match the Nios development board configuration controller to the type of flash image present on the board to which you are porting.

The flash image contained on the Nios development board is an 8-bit-wide, 16-MByte flash, which requires 24 address lines. Other size flashes may require a different number of address bits. If this is the case, you need to modify the configuration controller source code such that the address bus width matches the address width of your flash.

If you are using a different size flash, it will also be necessary to modify the location of the user and safe images within the configuration controller. The configuration controller for the Stratix II development board wires the upper two address bits (i.e., bits 23 and 22) to the 16-MByte flash and to 1 (or V_{CC}), and uses address bit 21 to select between the user and safe images. See Figure 5.

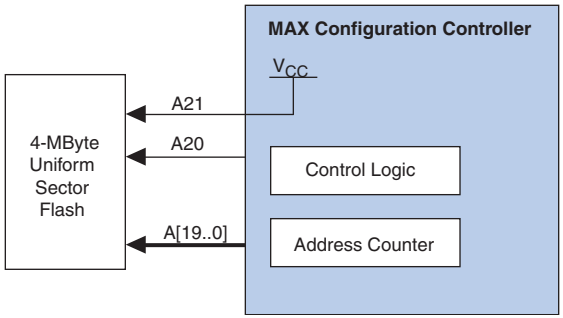
Figure 5. 8-Bit Wide, Uniform Sector Flash Connection



Note:
(1) Figure 5 assumes that the configuration target is an EP2C35 device. Other size devices may have different configuration file sizes requiring different address mapping.

If you are using a 4-MByte flash, change the address bus width from [23..0] to [21..0] and wire address bit 21 of the configuration controller address bus to 1 (or V_{CC}); then continue to use bit 20 to select between the user and safe image as indicated in Figure 6.

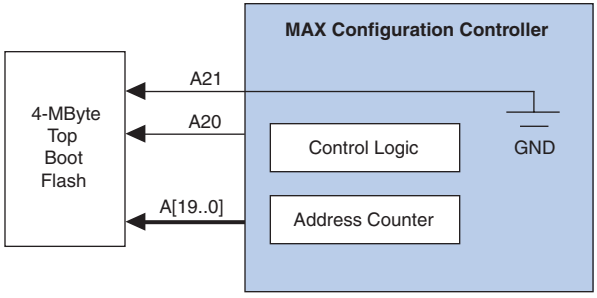
Figure 6. 8-Bit Wide, Uniform Sector Flash Connection



Note:
(1) Figure 6 assumes that the configuration target is an EP2C35 device. Other size devices may have different configuration file sizes requiring different address mapping.

If you are using a top boot sector flash, you will also need to make modifications to the source code. The flash used in the Nios development kits is a uniform sector flash; therefore, it is generally advisable to leave the lower sections of a uniform sector flash for the host processor and the upper sections for data storage. However, a top boot sector flash will generally reserve the upper sections as processor boot code. In this case, it is better to store the hardware configuration data in a lower section of the flash. This type of flash requires that the upper address bit (bit `21`) of the flash be tied to `0` (or `GND`), and bit `20` be used to select the user or safe image (see Figure 7).

Figure 7. 8-Bit-Wide, Top Boot Sector Flash Connections



Note:
(1) Figure 7 assumes that the configuration target is an EP2C35 device. Other size devices may have different configuration file sizes requiring different address mapping.

Modifying the Reference Design to Support a Variety of Systems & Configuration Requirements

In addition to accommodating different flash types, the configuration controller reference design can be modified to support many different types of systems and configuration requirements. For example a MAX device can be used to create configuration controllers that can:

- Select from multiple flash images
- Provide support for Ethernet upgrades
- Configure multiple devices

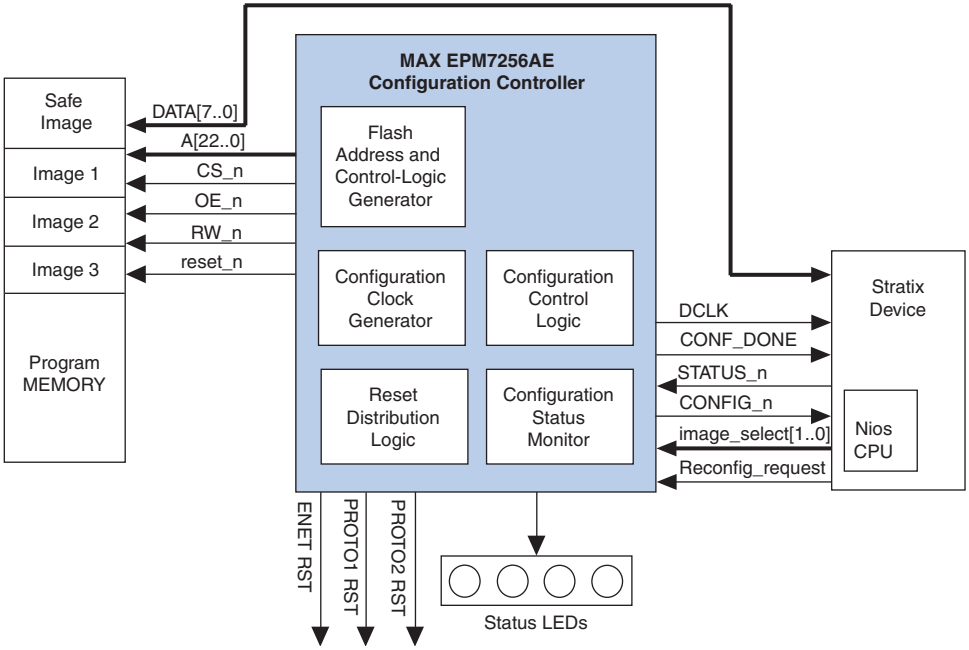
Selecting from Multiple Flash Images

The reference design that ships with Nios development kits is designed to first attempt to program the user image into the targeted FPGA. If the user image configuration attempt fails, the reference design will then attempt to program the safe image.

The concept of a priority image-selection design can easily be extended to include a number of user images that can be chosen to program the FPGA. Figure 8 shows an example configuration controller that is designed for priority image selection. The Figure 8 controller design is virtually identical to the Figure 3 controller design. However, the Figure 8 controller has a 2-bit `image_select[1..0]` bus that is used to select one of four user images stored in flash. The 2-bit `image_select[1..0]` bus is driven by an external source—such as a Nios embedded processor—running in the targeted device. To load a new image into the targeted device, the Nios processor (or external source) writes out the desired configuration image number to the `image_select[1..0]` bus and issues a reconfiguration request.

An alternative solution is to reserve a flash memory location that contains a hardware image number. In this scenario, the processor writes to the desired flash configuration number. When a reset or reconfiguration request is issued, the configuration controller reads the image number location from flash and proceeds to load the indicated image into the targeted device.

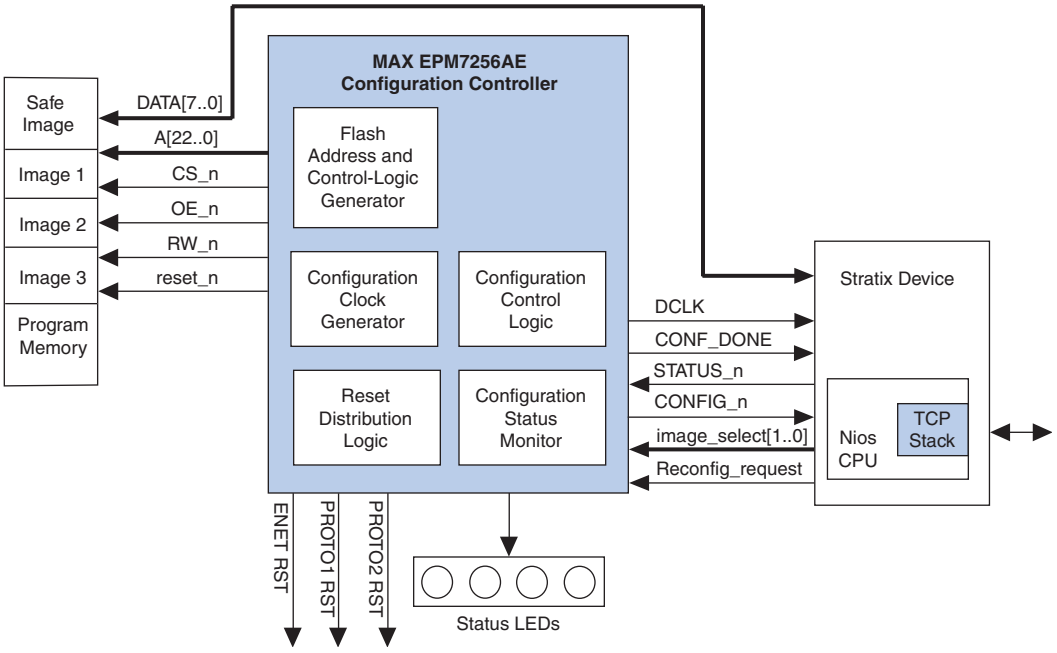
Figure 8. Image Select Capability



Upgrade Hardware Image via an Ethernet Link

The configuration controller that ships with Nios development kits supports upgrading the hardware image over an Ethernet link. The Ethernet upgrade can be achieved when a Nios embedded processor equipped with a transmission control protocol (TCP) stack is running in the targeted FPGA. In this scenario, the Nios processor can receive a programming file via an Ethernet link, and then burn the programming file into flash. After the flash file has been upgraded, the Nios processor issues a reconfiguration request, which downloads the new hardware image into the targeted device. Figure 9 shows an example system that can automatically be reconfigured via an Ethernet link.

Figure 9. Ethernet Reconfigurable System



Multiple Device Configuration

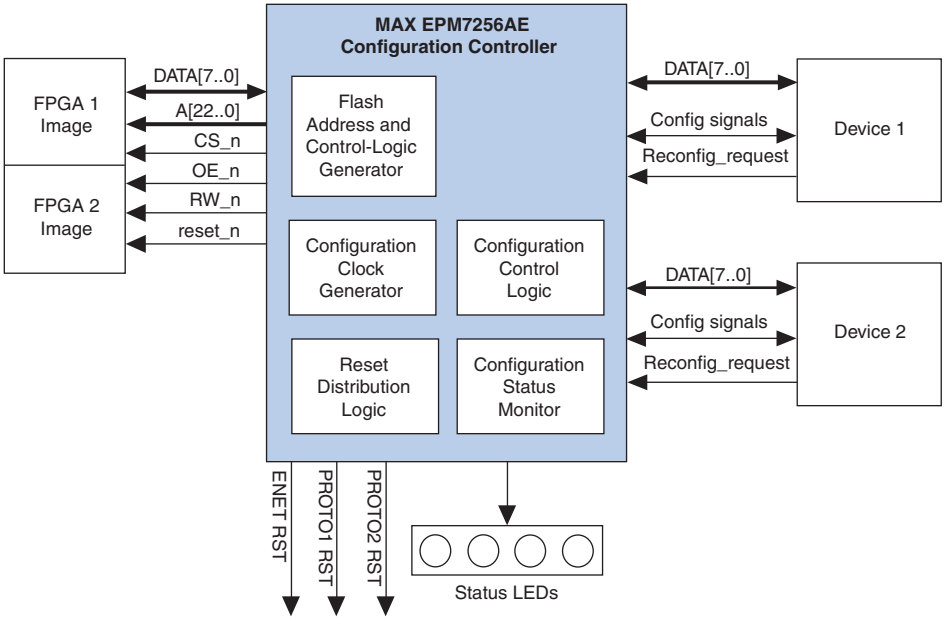
Multiple FPGAs on-board configuration is another application that demonstrates the flexibility of a CPLD-based configuration controller. There are a number of different configuration scenarios that exist when multiple FPGAs reside on the same board, including:

- Multiple identical FPGAs
- Multiple FPGAs from the same vendor with different hardware images
- Multiple FPGAs from different vendors

Figure 10 illustrates a configuration controller that could be used to configure multiple FPGAs where each FPGA requires a different configuration file. The Figure 10 controller can program different device family devices from either the same vendor or different vendors. In this

scenario, the flash memory contains two configuration images for two targeted FPGAs. The Figure 10 controller logic is designed to program one device at a time.

Figure 10. Multiple Device Configuration



Conclusion

The use of flash memory to store hardware configuration data for FPGAs located in an embedded system typically requires an intelligent configuration controller. Altera's Nios development boards, Stratix II and Cyclone II editions, contain a programmable configuration controller that provides a stable platform for managing system configuration and reset in embedded systems.

With the Nios development board configuration controller reference designs included with the Nios development kits, you can:

- Port reference designs to other boards
- Modify the reference designs to support more complex systems and configuration requirements

Because the Altera MAX family of CPLDs are ideally suited for use as board-level configuration controllers, the Nios development board's configuration controller utilizes an Altera MAX CPLD.

The programmable nature of MAX CPLDs allows the same device to be used in a variety of different systems. Because the MAX device can support both local and remote configuration updates, a MAX-based configuration controller is particularly useful in systems with embedded processors—such as the Nios embedded processor.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com