


Core Overview

The Cyclone® III Remote Update Controller core provides a method to control the Cyclone III remote update block from SOPC Builder systems. The core allows you to access all features of the ALTREMOTE_UPDATE megafunction through a simple Avalon® Memory-Mapped (Avalon-MM) slave interface. The slave interface allows Avalon-MM master peripherals, such as a Nios® II processor, to communicate with the core simply by reading and writing the registers.

The Cyclone III Remote Update Controller core is a thin Avalon interface layer on top of the ALTREMOTE_UPDATE megafunction. Every function of the core maps directly to a function of the megafunction. Altera recommends that you familiarize yourself with the ALTREMOTE_UPDATE megafunction before using the core.

 For more information about the ALTREMOTE_UPDATE megafunction, refer to the *altremote_update Megafunction User Guide*. For more information about remote system upgrade in Cyclone III devices, refer to the *Remote System Upgrade With Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*.

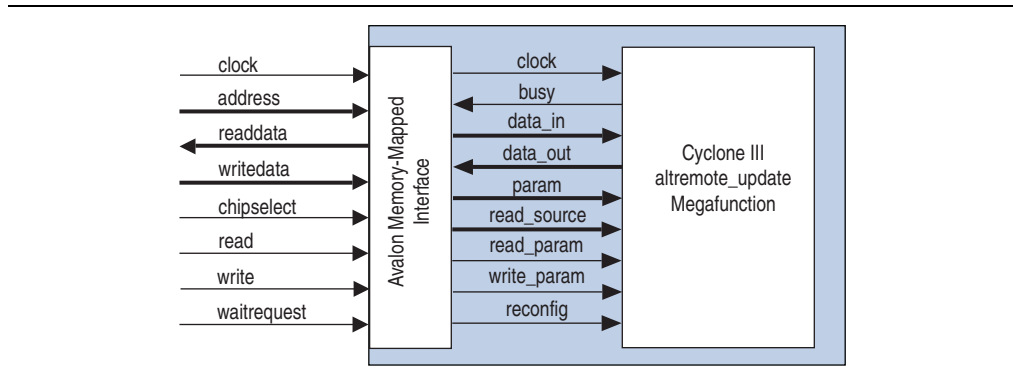
The Cyclone III Remote Update Controller core is SOPC Builder-ready and integrates easily into any SOPC Builder-generated system. This chapter contains the following sections:

- “Functional Description”
- “Device Support” on page 12-2
- “Instantiating the Core in SOPC Builder” on page 12-2

Functional Description

Figure 12-1 shows a block diagram of the Cyclone III Remote Update Controller core.

Figure 12-1. Cyclone III Remote Update Controller Core Block Diagram



Avalon-MM Slave Interface and Registers

The address bus on the core's Avalon-MM interface is 6 bits wide. The lower three bits of the address bus map directly to the `param` signal of the `ALTREMOTE_UPDATE` megafunction whereas the upper three bits map to the `read_source` signal.

Reading or writing to address offsets 0x00 – 0x1F of the Cyclone III Remote Update Controller core is equivalent to performing read or write operations to the `ALTREMOTE_UPDATE` megafunction using the `param` and `read_source` signals.

Table 12-1 shows the mapping of the 5 lowest order Remote Update Controller address bits to the `ALTREMOTE_UPDATE` megafunction signals.

Table 12-1. Avalon-MM Address Bits to Megafunction Signals Mapping

Address Bit	Megafunction Signal
address[0]	param[0]
address[1]	param[1]
address[2]	param[2]
address[3]	read_source[0]
address[4]	read_source[1]

The highest order address bit [5] is used to access a single `control/status` register. Reading or writing any address offset from 0x20 – 0x3F accesses the `control/status` register.

Table 12-2 shows the bit map of the `control/status` register.

Table 12-2. Bit Map of Control/Status Register

Bit(s)	Field	Access	Description
0	RECONFIG	RW	Set this bit to 1 to reset the FPGA and trigger reconfiguration.
1	RESET_TIMER	RW	Set this bit to 1 to reset the watchdog timer. Then, set this bit to 0 to allow the watchdog timer to continue.
2..31	Reserved		

Device Support

The Cyclone III Remote Update Controller core can only target Cyclone III device family. Both CFI flash and EPCS configuration devices are supported as non-volatile storage for configuration images.

Instantiating the Core in SOPC Builder

The Cyclone III Remote Update Controller core has no user-configurable parameters.

Software Programming Model

Software programs can operate the Cyclone III Remote Update Controller core by reading from and writing to the core's registers.



You can only reconfigure the FPGA to an application image from the factory image. Any attempt to reconfigure from an already reconfigured application image causes the FPGA to return to the factory image.

This section describes the most common types of operations using the Cyclone III Remote Update Controller core.

Setting the Configuration Offset

Before you reconfigure the FPGA, you must first specify the offset within the memory device from which you want to execute a reconfiguration. The offset is the relative address within the memory device where the configuration image is located. Write the offset value to address 0x04 of the core to set the configuration offset.

For example, if your system contains a CFI flash memory mapped at address 0x04000000, and the configuration image is located at address 0x100000 in the flash memory, the offset to set in the Cyclone III Remote Update Controller core is 0x100000.

Shifting the Configuration Offset Value

The ALTREMOTE_UPDATE megafunction requires that you provide only the 22 highest-order bits of a 24-bit address offset. To translate the address, right shift the offset by two bits. This results in a properly oriented 22-bit address offset.

If you are using a CFI flash device, you must also take into account the data width of the flash. If the data width of your flash device is 16 bits, you must provide a 16-bit address offset to the Cyclone III Remote Update Controller core. This requires an additional 1-bit right shift of the byte address offset. No translation is necessary if the data width of your flash is 8 bits.

If you are using an EPCS serial configuration device, consider the data width of the device to be 8 bits. Even though the EPCS device is a serial device, it uses byte addressing internally.

For example, an FPGA is set up to configure itself using active parallel mode from a 16-bit CFI flash memory mapped at address 0x04000000 in an SOPC Builder system, and the configuration image is located at byte offset 0x100000 within the flash memory. To derive the correct configuration offset, you must first right shift the byte offset 0x100000 by one bit to obtain the 16-bit address. Then, right shift by another two bits to obtain the highest 22 bits of the 24-bit offset. The result is a configuration offset of 0x20000 ($0x100000 \gg 3 = 0x20000$), to be written to address 0x04 of the core.

Setting up the Watchdog Timer

You can set up the watchdog timer by writing the upper 12 bits of the 29-bit timeout value to address 0x02 of the core. To reset the watchdog timer, set the RESET_TIMER bit of the control/status register to 1 and immediately set the bit to 0.



Ensure that you don't accidentally set bit 0 of the `control/status` register to 1. Otherwise, you will trigger a reconfiguration of the FPGA.



For more information on watchdog timer, refer to the *ALTREMOTE_UPDATE Megafunction User Guide*.

If you do not use the watchdog timer feature of the `ALTREMOTE_UPDATE` megafunction, it must be disabled before a reconfiguration is performed. To disable the watchdog timer, write `0x00` to address `0x03` of the core.

Triggering a Reconfiguration

You can trigger a reconfiguration once you have set the reconfiguration offset in the Cyclone III Remote Update Controller core, and you have either setup or disabled the watchdog timer. To trigger a reconfiguration, set the `RECONFIG` bit in the `control/status` register to 1. Consequently, the FPGA performs a reset and reconfigures itself from the configuration image specified.

Code Example

Example 12-1 shows a C function that can be used to operate the Cyclone III Remote Update Controller core from a processor such as Nios II.

Example 12-1. FPGA Reconfiguration Function

```
/* *****  
 * Function: CycloneIII_Reconfig  
 * Purpose: Uses the ALT_REMOTE_UPDATE megafunction to reconfigure a Cyclone III FPGA.  
 * Parameters:  
 *   remote_update_base - base address of the remote update controller  
 *   flash_base         - base address of flash device  
 *   reconfig_offset   - offset in flash from which to reconfigure  
 *   watchdog_timeout  - 29-bit watchdog timeout value  
 *   width_of_flash    - data-width of flash device  
 * Returns: 0 ( but never exits since it reconfigures the FPGA )  
 * *****/  
int CycloneIII_Reconfig( int remote_update_base,  
                        int flash_base,  
                        int reconfig_offset,  
                        int watchdog_timeout,  
                        int width_of_flash )  
{int offset_shift;  
  
    // Obtain upper 12 bits of 29-bit watchdog timeout value  
    watchdog_timeout = watchdog_timeout >> 17;  
  
    // Only enable the watchdog timer if its timeout value is greater than 0.  
    if( watchdog_timeout > 0 )  
    {  
        // Set the watchdog timeout value  
        IOWR( remote_update_base, 0x2, watchdog_timeout );  
    }  
    else  
    {  
        // Disable the watchdog timer  
        IOWR( remote_update_base, 0x3, 0 );  
    }  
  
    // Calculate how much to shift the reconfig offset location:  
    // width_of_flash == 8->offset_shift = 2.  
    // width_of_flash == 16->offset_shift = 3  
    offset_shift = (( width_of_flash / 8 ) + 1 );  
  
    // Write the offset of the desired reconfiguration image in flash  
    IOWR( remote_update_base, 0x4, reconfig_offset >> offset_shift );  
  
    // Perform the reconfiguration by setting bit 0 in the  
    // control/status register  
    IOWR( remote_update_base, 0x20, 0x1 );  
  
    return( 0 );  
}
```

Related Documentation

This chapter references the following documents:

- *altrremote_update Megafunction User Guide*
- *Remote System Upgrade With Cyclone III Devices* in volume 1 of the *Cyclone III Device Handbook*.

Document Revision History

Table 12-3 shows the revision history for this chapter.

Table 12-3. Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v9.1.0	No change from previous release.	—
March 2009 v9.0.0	No change from previous release.	—
November 2008 v8.1.0	Initial release.	—