

Altera Remote Update IP Core User Guide

2017.04.10

UG-31005



Subscribe



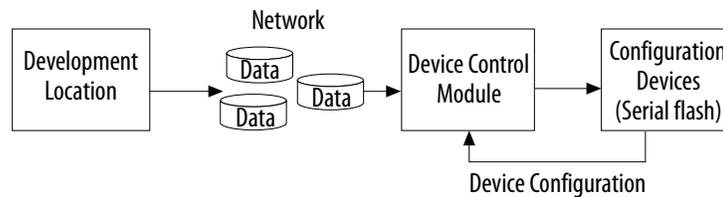
Send Feedback

The Altera Remote Update IP core implements a device reconfiguration using dedicated remote system upgrade circuitry available in supported devices. Remote system upgrade helps you deliver feature enhancements and bug fixes without recalling your product, reduces time-to-market, and extends product life. The Altera Remote Update IP core commands the configuration circuitry to start a reconfiguration cycle.

The dedicated circuitry performs error detection during and after the configuration process. When the dedicated circuitry detects errors, the circuitry facilitates system recovery by reverting back to a safe, default factory configuration image and then provides error status information.

The following figures shows a functional diagram for a typical remote system upgrade process.

Figure 1: Typical Remote System Upgrade Process



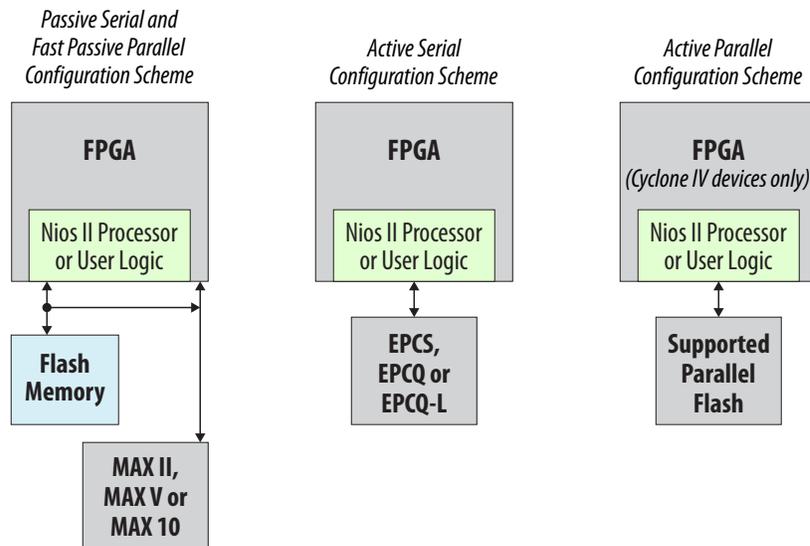
Note: Altera recommends that you use the following f_{MAX} values:

- 10 MHz— for Arria II and Stratix IV devices
- 20 MHz—for other supported devices

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2008
Registered

Figure 2: High-Level Block Diagram of Remote System Upgrade

Note: The remote system upgrade feature support for each configuration scheme varies between device family. For more information about the configuration scheme and the remote system upgrade feature, please refer to the configuration chapter of the respective device family handbook.

Related Information

- [Altera Remote Update IP Core Knowledge Base](#)
- [Configuration Support Center](#)
- [Introduction to Altera IP Cores](#)
Provides general information about all Altera FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
Create simulation scripts that do not require manual updates for software or IP version upgrades.
- [Project Management Best Practices](#)
Guidelines for efficient management and portability of your project and IP files.
- [Altera Remote Update IP Core User Guide Archives](#) on page 45
Provides a list of user guides for previous versions of the Altera Remote Update IP core.

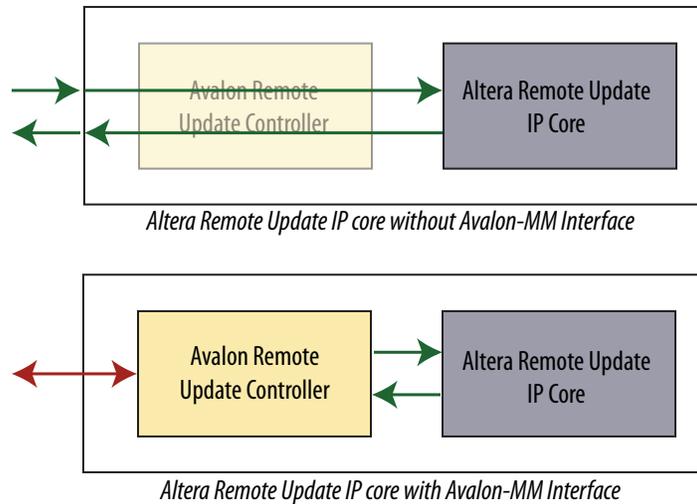
Avalon-MM in Altera Remote Update IP Core

The Avalon-MM interface is supported in the Altera Remote Update IP core. You can only use the IP core either with or without Avalon-MM interface. You can instantiate the Avalon-MM Interface by turning on the **Add support for Avalon Interface** option in Altera Remote Update IP parameter editor.

Note: The Avalon-MM support for Altera Remote Update IP core is available in Quartus II Software version 15.0 and onwards.

Figure 3: Altera Remote Update IP Core Implementation with and without Avalon-MM Interface

Figure shows the Avalon Remote update support architecture which consists of 2 components; Altera Remote Update IP core and Avalon remote update controller. If Avalon interface is enabled, the conduit interfaces of Altera Remote Update IP core will connect to conduit interface of the controller.

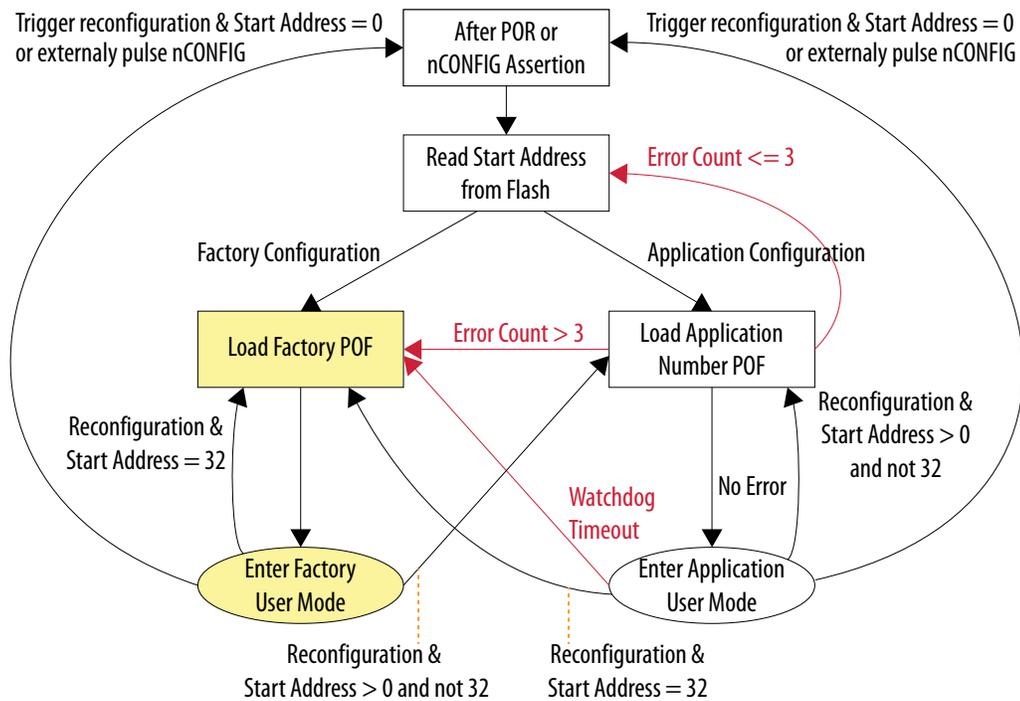


Arria 10 Devices

Remote System Configuration Mode

Remote configuration supports “Direct to application” (DTA) and “Application to Application” update. Remote configuration only supports a 4-byte address scheme so there is no support for devices with densities smaller than 128 Mbit.

Figure 4: Transitions Between Factory and Application Configurations in Remote Update Mode



When you use low-voltage quad-serial configuration (EPCQ-L) devices, the remote update mode allows a configuration space to start at any flash sector boundary. This capability allows a maximum of 512 pages in the EPCQ-L256 device and 1024 pages in the EPCQ-L512 device, in which the minimum size of each page is 512 Kbits. Additionally, the remote update mode features an optional user watchdog timer that can detect functional errors in an application configuration.

Note: When error occurs, the AS controller will load the same application configuration image for three times before reverting to factory configuration image. By that time, the total time taken exceeds 100ms and violates the PCIe boot-up time when using CvP configuration mode. If your design is sensitive to the PCIe boot-up requirement, Altera recommends that you do not use the direct-to-application feature.

Note: Altera recommends that you set a fixed start address and never update the start address during user mode. You should only overwrite an existing application configuration image when you have a new application image. This is to avoid the factory configuration image to be erased unintentionally every time you update the start address.

Remote System Configuration Components

Table 1: Remote System Configuration Components in Arria 10 Devices

Components	Details
Page mode feature	The dedicated 32-bit start address register PGM[31..0] holds the start address.

Components	Details
Factory configuration	<p>Factory configuration can be set as the default configuration setup depending on the address pointer set.</p> <p>The factory configuration loads into the device upon power-up.</p> <p>If a system encounters an error while loading application configuration data or if the device reconfigures due to <code>nCONFIG</code> assertion, the device loads the factory configuration. The remote system configuration register determines the reason for factory configuration. Based on this information, the factory configuration determines which application configuration to load.</p>
Application configuration	<p>Application configuration can be the default configuration setup depending on the address pointer set.</p> <p>The application configuration loads into the device upon power-up.</p> <p>The application configuration is the configuration data from a remote source and the data is stored in different locations or pages of the memory storage device, excluding the factory page.</p>
Watchdog timer	<p>A watchdog timer is a circuit that determines the functionality of another mechanism. The watchdog timer functions like a time delay relay that remains in the reset state while an application runs properly.</p> <p>Arria 10 devices are equipped with a built-in watchdog timer for remote system configuration to prevent a faulty application configuration from indefinitely stalling the device.</p> <p>The timer is a 29-bit counter, but you use only the upper 12 bits to set the value for the watchdog timer.</p> <p>The timer begins counting after the device goes into user mode. To ensure the application configuration is valid, you must continuously reset the watchdog <code>reset_timer</code> within a specific duration during user mode operation.</p> <p>If the application configuration does not reset the user watchdog timer before time expires, the dedicated circuitry reconfigures the device with the factory configuration and resets the user watchdog timer.</p>
Remote update sub-block	<p>The remote update sub-block manages the remote configuration feature. A remote configuration state machine controls this sub-block. This sub-block generates the control signals required to control the various configuration registers.</p>

Components	Details
Remote configuration registers	<p>The remote configuration registers keep track of page addresses and the cause of configuration errors. You can control both the update and shift registers. The status and control registers are controlled by internal logic, but are read via the shift register. The control register is 38-bits wide.</p> <p>For details about configuration registers, refer to the <i>Configuration, Design Security, and Remote System Upgrades</i> chapter in the respective device handbook.</p>

Parameter Settings

Table 2: Altera Remote Update IP Core Parameters for Arria 10 Devices

GUI Name	Values	Description
Which operation mode will you be using?	REMOTE	Specifies the configuration mode of the ALTERA REMOTE UPDATE IP core.
Which configuration device will you be using?	EPCQ-L device	Choose the configuration device you are using.
Add support for writing configuration parameters	—	Enable this if you need to write configuration parameters.
Add support for Avalon Interface	—	Enable this if you are using Avalon interface.
Enable reconfig POF checking	—	Not available as it is not required.

Ports

Table 3: Altera Remote Update IP Core Ports for Arria 10 Devices

Name	Port	Required?	Description
read_param	Input	No	<p>Read signal for the parameter specified in <code>param[]</code> input port and fed to <code>data_out[]</code> output port.</p> <p>Signal indicating the parameter specified on the <code>param[]</code> port should be read. The number of bits set on <code>data_out[]</code> depends on the parameter type. The signal is sampled at the rising clock edge. Assert the signal for only one clock cycle to prevent the parameter from being read again in a subsequent clock cycle.</p> <p>The busy signal is activated as soon as <code>read_param</code> is read as active. While the parameter is being read, the busy signal remains asserted, and <code>data_out[]</code> has invalid data. When the busy signal is deactivated and <code>data_out[]</code> has a valid data, another parameter can be read.</p>
write_param	Input	No	<p>Write signal for parameter specified in <code>param[]</code> and with value specified in <code>data_in[]</code>.</p> <p>Signal indicating parameter specified with <code>param[]</code> should be written into remote update block with the value specified in <code>data_in[]</code>. The number of bits read from <code>data_in[]</code> depends on the parameter type.</p> <p>The signal is sampled at the rising clock edge. The signal should be asserted for only one clock cycle to prevent the parameter from being rewritten on a subsequent clock cycle. The busy signal is activated as soon as <code>write_param</code> is read as being active. While the parameter is being written, the busy signal remains asserted, and input to <code>data_in[]</code> is ignored. When the busy signal is deactivated, another parameter can be written. This signal is only valid in factory configuration mode because parameters cannot be written in Application configuration mode.</p>
<code>param[]</code>	Input	No	<p>Bus that specifies which parameter need to be read or updated.</p> <p>A 3-bit bus that selects the parameter to be read or updated. If left unconnected, the default value for this port is 000.</p>

Name	Port	Required?	Description
<code>data_in[]</code>	Input	No	<p>Data input for writing parameter data into the remote update block. Input bus for parameter data.</p> <p>For some parameters, not all bits are used. In this case, the lower-order bits are used (for example, status values use bits [4:0]).</p> <p>If left unconnected, this bus defaults to 0. The port is ignored if the current configuration is the Application configuration.</p> <p>A 32-bit bus width (4-bytes addressing configuration device, for example EPCQ-L256) in the Quartus II software version 14.0 or later.</p>
<code>reconfig</code>	Input	Yes	<p>Signal indicating that reconfiguration of the part should begin using the current parameter settings. A value of 1 indicates reconfiguration should begin. This signal is ignored if the <code>busy</code> signal is asserted to ensure all parameters are completely written before reconfiguration begins.</p>
<code>reset_timer</code>	Input	No	<p>Reset signal for watchdog timer.</p> <p>Signal indicating the internal watchdog timer should be reset. Unlike other inputs, this signal is not affected by the <code>busy</code> signal and can reset the timer even when the <code>busy</code> signal is asserted.</p> <p>A falling edge of this signal triggers a reset of the user watchdog timer.</p> <p>For the timing specification of this parameter, refer to the specific device handbook.</p>
<code>clock</code>	Input	Yes	<p>Clock input to the remote update block.</p> <p>Clock input to control the machine and to drive the remote update block during the update of parameters.</p> <p>This port must be connected to a valid clock.</p>
<code>reset</code>	Input	Yes	<p>This is an active high signal. Asserting this signal high will reset the IP core.</p> <p>Asynchronous reset input to the IP core to initialize the machine to a valid state. The machine must be reset before first use, otherwise the state is not guaranteed to be valid.</p>

Name	Port	Required?	Description
busy	Output	No	<p>Busy signal that indicates when remote update block is reading or writing data.</p> <p>While this signal is asserted, the machine ignores most of its inputs and cannot be altered until the machine deasserts this signal. Therefore, changes are made only when the machine is not busy.</p> <p>This signal goes high when <code>read_param</code> or <code>write_param</code> is asserted, and remains high until the read or write operation completes.</p>
data_out[]	Output	No	<p>Data output when reading parameters.</p> <p>This bus holds read parameter data from the remote update block. The <code>param[]</code> value specifies the parameter to read. When the <code>read_param</code> signal is asserted, the parameter value is loaded and driven on this bus. Data is valid when the busy signal is deasserted.</p> <p>If left unconnected, the default value for the port is 0.</p> <p>The width of this bus is device-dependent. For the Quartus II software version 14.0 and onwards, the bus width is 32-bit—using 4-byte addressing configuration device, for example EPCQL-256.</p>
ctl_nupt	Input	Yes	<p>This port allows you to select which register to be read whenever <code>read_param</code> operation is running.</p> <ul style="list-style-type: none"> • A logic high selects the Control Register—register containing the current remote update settings such as watchdog timer settings, configuration mode (AnF), and page address. • A logic low selects the Update Register—register containing similar data as held in the Control Register, but the values are updated via <code>write_param</code> operation for use in next reconfiguration.

Parameters

Table 4: Parameter Type and Corresponding Parameter Bit Width Mapping for Arria 10 Devices

Bit	Parameter	Width	Comments
000	Reconfiguration trigger conditions (Read Only)	5	<ul style="list-style-type: none"> Bit 4—<code>wdtimer_source</code>: User watchdog timer timeout. Bit 3—<code>nconfig_source</code>: External configuration reset (<code>nCONFIG</code>) assertion. Bit 2—<code>runconfig_source</code>: Configuration reset triggered from logic array. Bit 1—<code>nstatus_source</code>: <code>nSTATUS</code> asserted by an external device as the result of an error. Bit 0—<code>crcerror_source</code>: CRC error during application configuration. <p>The POR value for all bits are 0.</p>
001	Illegal Value		
010	Watchdog Timeout Value	12	—
011	Watchdog Enable	1	—
100	Page Select	32	<p>For the Quartus II software version 14.0 and onwards:</p> <ul style="list-style-type: none"> Width of 32 when reading and writing the start address. For active serial devices using 32-bit addressing, such as EPCQL-256, <code>PGM[31..2]</code> corresponds to the upper 30 bits of the 32-bits start address. <code>PGM[1..0]</code> is read as <code>2'b0</code>.
101	Configuration Mode (AnF)	1	<p>This parameter is set to 1 in application page and is set to 0 in factory page. In remote update mode, this parameter can be read and written.</p> <p>Before loading the application page in remote update mode, Altera recommends that you set this parameter to 1. The content of the control register cannot be read properly if you fail to do so.</p>
110	Illegal Value		
111	Illegal Value		

Avalon-MM Interface

Control Status Register Signals

Table 5: Altera Remote Update IP Core Avalon-MM Control Status Register Signals for Arria 10 Devices

Name	Width	Direction	Description
clk	1	Input	Clock input
reset	1	Input	Reset input
avl_csr_address	3	Input	Address bus
avl_csr_read	1	Input	Perform a read transaction
avl_csr_write	1	Input	Perform a write transaction
avl_csr_readdata	32	Output	Read data from IP
avl_csr_readdata_valid	1	Output	Indicate when read data is valid
avl_csr_writedata	32	Input	Write data to IP
avl_csr_waitrequest	1	Output	Waitrequest signal high indicates the core is busy

Control Status Register Write Operation

To execute the write operation for the control status register, perform the following steps:

1. Asserts the `avl_csr_write` high.
2. Write a correct address of the register in the `avl_csr_address` bus. Refer to the Register Map for register information.
3. Write data into the `avl_csr_writedata` bus.

Related Information

[Register Map](#) on page 13

Control Status Register Read Operation

To execute the read operation for the control status register, perform the following steps:

1. Asserts `avl_csr_read` high.
2. Write a correct address of the register in the `avl_csr_address` bus. Refer to the Register Map for register information.
3. Wait for the `avl_csr_readdata_valid` signal to go high.
4. Retrieve read data from `avl_csr_readdata`.

Operations Example Waveforms

Figure 5: Waveform for Write Operation

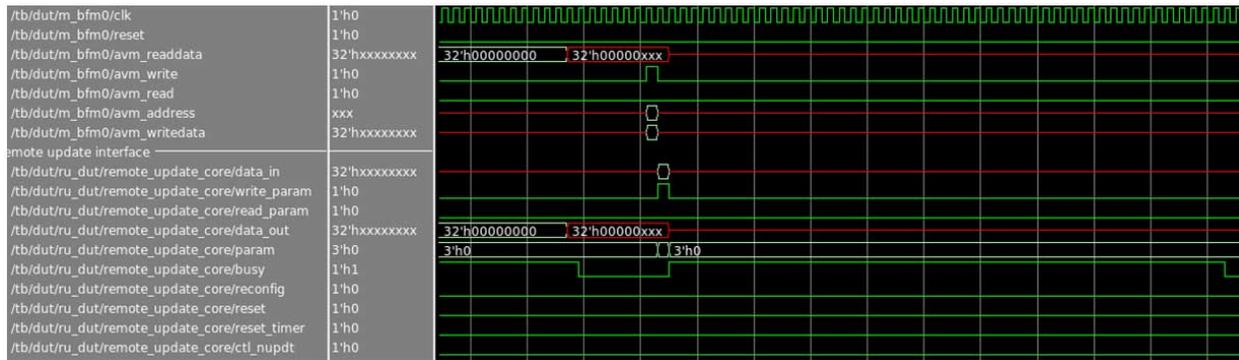


Figure 6: Waveform for Read Operation

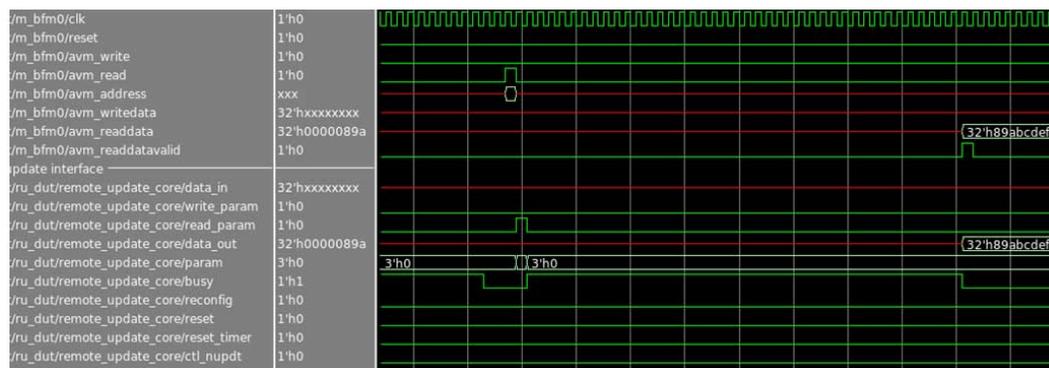


Figure 7: Waveform for RU_CTL_NUPDT Operation

The RU_CTL_NUPDT will hold the value until a new value is inserted.

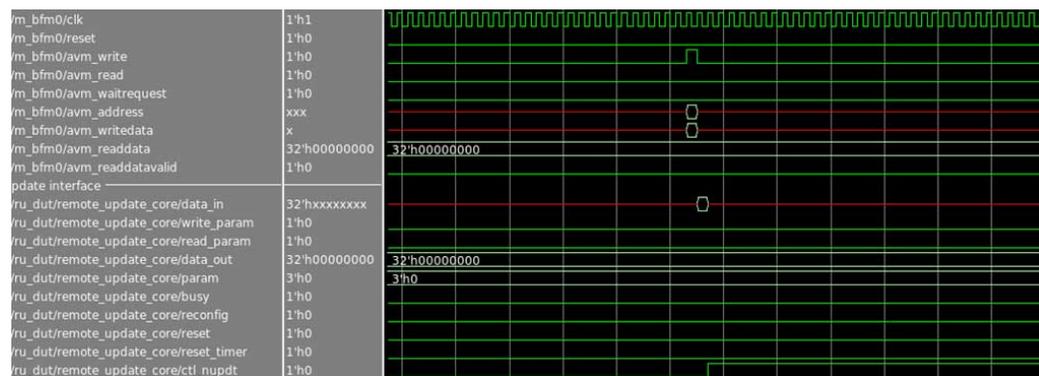
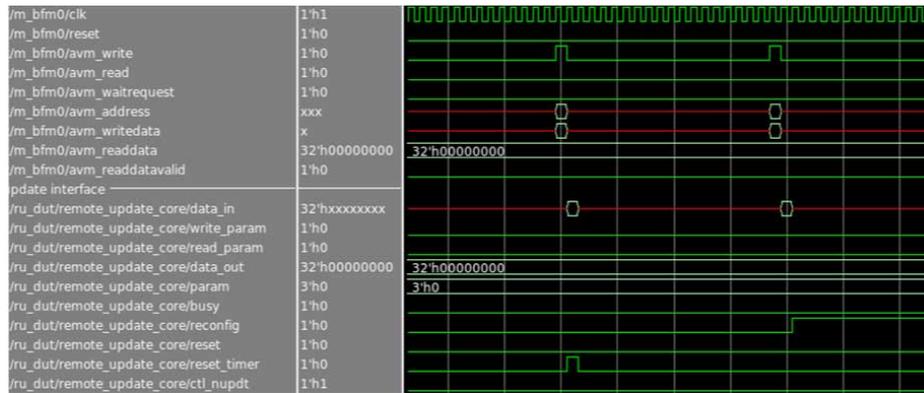


Figure 8: Waveform for Reset Timer and Reconfiguration Operation

The RU_RECONFIG will hold the value until the reconfiguration process is done.



Register Map

Table 6: Altera Remote Update IP Core Avalon-MM Register Map for Arria 10 Devices

- The IP core can read or write each field separately as each command has different parameter value.
- The default value for the registers is 0.

Register Name	Address Offset	Width	R/W	Description
RU_RECONFIG_TRIGGER_CONDITIONS	0x0	5	Read	Read configuration trigger conditions. <ul style="list-style-type: none"> • Bit 4—wdtimer_source: User watchdog timer timeout • Bit 3—nconfig_source: External configuration reset (nCONFIG) assertion. • Bit 2—runconfig_source: Configuration reset triggered from logic array • Bit 1—nstatus_source: nSTATUS asserted by an external device as the result of an error • Bit 0—crcerror_source: CRC error during application configuration.
RU_WATCHDOG_TIMEOUT	0x1	12	Read/Write	Read or write watchdog timeout value.
RU_WATCHDOG_ENABLE	0x2	1	Read/Write	Enable or disable watchdog timeout. <ul style="list-style-type: none"> • 0: Disable • 1: Enable

Register Name	Address Offset	Width	R/W	Description
RU_PAGE_SELECT	0x3	24 or 32	Read/Write	Read or write start address of the configuration image.
RU_CONFIGURATION_MODE	0x4	1	Read/Write	Write configuration mode set to 1 in application page and 0 in factory page.
RU_RESET_TIMER	0x5	1	Write	Write a value of 1 to this register to trigger reset timer of the remote update. The IP will automatically trigger a reset pulse to the reset timer pin of the remote update.
RU_RECONFIG	0x6	1	Write	Write a value of 1 to this register to trigger reconfiguration from a new image. The IP will set 1 to the <code>reconfig</code> pin of the remote update and hold this value until the process done.
RU_CTL_NUPDT	0x7	1	Write	Allow capturing of data from either Control/Update register by controlling <code>ctl_nupdt</code> . <ul style="list-style-type: none"> 0: Capture value from Update Register 1: Capture data from Control Register

Enabling Remote System Upgrade Circuitry

To enable the remote system upgrade feature, select **Active Serial** or **Configuration Device** from the Configuration scheme list in the **Configuration** page of the **Device and Pin Options** dialog box in the Quartus® Prime software.

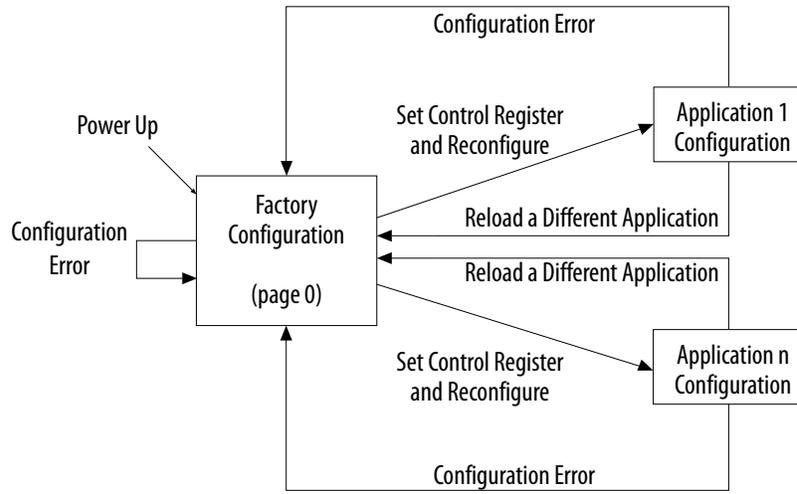
Altera-provided Altera Remote Update IP core provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read and write protocol in the device logic.

Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices

Remote System Configuration Mode

Remote Configuration Mode

Figure 9: Remote Configuration Mode



When using with serial configuration (EPCS) or quad-serial configuration (EPCQ) devices, the remote update mode allows a configuration space to start at any flash sector boundary, allowing a maximum of 128 pages in the EPCS64 device and 32 pages in the EPCS16 device, in which 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.

Remote System Configuration Components

Table 7: Remote System Configuration Components in Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices

Components	Details
Page mode feature	The dedicated 24-bit start address register <code>PGM[23..0]</code> holds the start address.

Components	Details
Factory configuration	<p>Factory configuration is the default configuration setup.</p> <p>In remote configuration mode, the factory configuration loads into the device upon power-up.</p> <p>If a system encounters an error while loading application configuration data or if the device reconfigures due to <code>nCONFIG</code> assertion, the device loads the factory configuration. The remote system configuration register determines the reason for factory configuration. Based on this information, the factory configuration determines which application configuration to load.</p>
Application configuration	<p>The application configuration is the configuration data from a remote source and the data is stored in different locations or pages of the memory storage device, excluding the factory default page.</p>
Watchdog timer	<p>A watchdog timer is a circuit that determines the functionality of another mechanism. The watchdog timer functions like a time delay relay that remains in the reset state while an application runs properly.</p> <p>Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V devices are equipped with a built-in watchdog timer for remote system configuration to prevent a faulty application configuration from indefinitely stalling the device.</p> <p>The timer is a 29-bit counter, but you use only the upper 12 bits to set the value for the watchdog timer.</p> <p>The timer begins counting after the device goes into user mode. If the application configuration does not reset the user watchdog timer before time expires, the dedicated circuitry reconfigures the device with the factory configuration and resets the user watchdog timer.</p> <p>To ensure the application configuration is valid, you must continuously reset the watchdog <code>reset_timer</code> within a specific duration during user mode operation.</p>
Remote update sub-block	<p>The remote update sub-block manages the remote configuration feature. A remote configuration state machine controls this sub-block. This sub-block generates the control signals required to control the various configuration registers.</p>

Components	Details
Remote configuration registers	<p>The remote configuration registers keep track of page addresses and the cause of configuration errors. You can control both the update and shift registers. The status and control registers are controlled by internal logic, but are read via the shift register. The control register is 38-bit wide.</p> <p>For details about configuration registers, refer to the Configuration, Design Security, and Remote System Upgrades chapter in the respective device handbook.</p>

Parameter Settings

Table 8: Altera Remote Update IP Core Parameters for Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices

GUI Name	Values	Description
Which operation mode will you be using?	REMOTE	Specifies the configuration mode.
Which configuration device will you be using?	<ul style="list-style-type: none"> EPCS device EPCQ device 	Choose the configuration device you are using.
Add support for writing configuration parameters	—	Enable this if you need to write configuration parameters.
Add support for Avalon Interface ⁽¹⁾	—	Enable this if you are using Avalon interface.
Enable reconfig POF checking	—	Allows you to enable .pof checking, which allows the remote update block to verify the existence of an application configuration image before the image is loaded. When you turn on this parameter, the Altera Remote Update IP core checks the .pof and sends the <code>reconfig</code> signal. This option is disabled by default.

The POF checking feature detects and verifies the existence of an application configuration image before the image is loaded. Loading an invalid application configuration image may lead to unexpected behaviour of the FPGA including system failure. Examples of invalid application configuration images are:

- A partially programmed application image
- A blank application image
- An application image assigned with a wrong start address

⁽¹⁾ Parameter not available in Stratix II devices.

Ports

Table 9: Altera Remote Update IP Core Ports for Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices

Name	Port	Required?	Description
read_param	Input	No	<p>Read signal for the parameter specified in <code>param[]</code> input port and fed to <code>data_out[]</code> output port.</p> <p>Signal indicating the parameter specified on the <code>param[]</code> port should be read. The number of bits set on <code>data_out[]</code> depends on the parameter type. The signal is sampled at the rising clock edge. Assert the signal for only one clock cycle to prevent the parameter from being read again in a subsequent clock cycle.</p> <p>The busy signal is activated as soon as <code>read_param</code> is read as active. While the parameter is being read, the busy signal remains asserted, and <code>data_out[]</code> has invalid data. When the busy signal is deactivated, <code>data_out[]</code> is valid, another parameter can be read.</p>
write_param	Input	No	<p>Write signal for parameter specified in <code>param[]</code> and with value specified in <code>data_in[]</code>.</p> <p>Signal indicating parameter specified with <code>param[]</code> should be written into remote update block with the value specified in <code>data_in[]</code>. The number of bits read from <code>data_in[]</code> depends on the parameter type.</p> <p>The signal is sampled at the rising clock edge. The signal should be asserted for only one clock cycle to prevent the parameter from being rewritten on a subsequent clock cycle. The busy signal is activated as soon as <code>write_param</code> is read as being active. While the parameter is being written, the busy signal remains asserted, and input to <code>data_in[]</code> is ignored. When the busy signal is deactivated, another parameter can be written. This signal is only valid in factory configuration mode because parameters cannot be written in application configuration mode.</p>

Name	Port	Required?	Description
param[]	Input	No	<p>Bus that specifies which parameter need to be read or updated.</p> <p>A 3-bit bus that selects the parameter to be read or updated. If left unconnected, the default value for this port is 000.</p> <p>For more information, refer to Parameters on page 22.</p>
data_in[]	Input	No	<p>Data input for writing parameter data into the remote update block. Input bus for parameter data.</p> <p>For some parameters, not all bits are used. In this case, the lower-order bits are used (for example, status values use bits [4:0]).</p> <p>If left unconnected, this bus defaults to 0. The port is ignored if the current configuration is the Application configuration.</p> <p>A 24-bit bus width in the Quartus II software version 13.0 or earlier. For the Quartus II software version 13.1 and onwards, the bus widths are as follow:</p> <ul style="list-style-type: none"> • 24-bit bus width—using 3-byte addressing configuration device, for example EPCS128. • 32-bit bus width—using 4-byte addressing configuration device, for example EPCQ256.
reconfig	Input	Yes	<p>Signal indicating that reconfiguration of the part should begin using the current parameter settings. A value of 1 indicates reconfiguration should begin. This signal is ignored while the busy signal is asserted to ensure all parameters are completely written before reconfiguration begins.</p>
reset_timer	Input	No	<p>Reset signal for watchdog timer.</p> <p>Signal indicating the internal watchdog timer should be reset. Unlike other inputs, this signal is not affected by the busy signal and can reset the timer even when the busy signal is asserted.</p> <p>A falling edge of this signal triggers a reset of the user watchdog timer.</p> <p>For the timing specification of this parameter, refer to the specific device handbook.</p>

Name	Port	Required?	Description
clock	Input	Yes	<p>Clock input to the remote update block.</p> <p>Clock input to control the machine and to drive the remote update block during the update of parameters.</p> <p>This port must be connected to a valid clock.</p>
reset	Input	Yes	<p>This is an active high signal. Asserting this signal high will reset the IP core.</p> <p>Asynchronous reset input to the IP core to initialize the machine to a valid state. The machine must be reset before first use, otherwise the state is not guaranteed to be valid.</p>
busy	Output	No	<p>Busy signal that indicates when remote update block is reading or writing data.</p> <p>While this signal is asserted, the machine ignores most of its inputs and cannot be altered until the machine deasserts this signal. Therefore, changes are made only when the machine is not busy.</p> <p>This signal goes high when <code>read_param</code> or <code>write_param</code> is asserted, and remains high until the read or write operation completes.</p>
data_out[]	Output	No	<p>Data output when reading parameters.</p> <p>This bus holds read parameter data from the remote update block. The <code>param[]</code> value specifies the parameter to read. When the <code>read_param</code> signal is asserted, the parameter value is loaded and driven on this bus. Data is valid when the busy signal is deasserted.</p> <p>If left unconnected, the default value for the port is 0. The width of this bus is device-dependent:</p> <p>For the Quartus II software version 13.0 or earlier, the bus widths is 24 bits. For are Quartus II software version 13.1 and onwards are as follow:</p> <ul style="list-style-type: none"> • 24-bit bus width—using 3-byte addressing configuration device, for example EPCS128. • 32-bit bus width—using 4-byte addressing configuration device, for example EPCQ256.

Name	Port	Required?	Description
asmi_busy	Input	No	<p>Input from the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>true</code>.</p> <p>A logic high on this pin indicates that the ALTASMI_PARALLEL IP core is busy processing the operation. The Altera Remote Update IP core waits for this pin to go low before initiating another operation.</p> <p>Wire this pin to the <code>asmi_busy</code> output port of the ALTASMI_PARALLEL IP core.</p>
asmi_data_valid	Input	No	<p>Input from the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>true</code>.</p> <p>A logic high on this pin indicates valid data in the <code>asmi_dataout[7..0]</code> output port of the ALTASMI_PARALLEL IP core.</p> <p>Wire this pin to the <code>asmi_data_valid</code> output port of the ALTASMI_PARALLEL IP core.</p>
asmi_dataout	Input	No	<p>Input from the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>true</code>.</p> <p>The Altera Remote Update IP core presents the address information on this pin before initiating the read operation on the ALTASMI_PARALLEL IP core.</p>
pof_error	Output	No	<p>Detects an invalid application configuration image.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>TRUE</code>.</p> <p>A logic high on this pin indicates that the Altera Remote Update IP core detects an invalid application configuration image. If asserted high, you must take corrective action by reloading a new application configuration image or specifying a different address location in the EPCS or EPCQ that contains a valid application configuration image. Wire this pin based on your system requirement.</p>

Name	Port	Required?	Description
asmi_addr	Output	No	Address signal to the <code>altasmi_parallel</code> component. Available when the <code>check_app_pof</code> parameter is set to TRUE. The Altera Remote Update IP core presents the address information on this pin before initiating the read operation on the ALTASMI_PARALLEL IP core.
asmi_read	Output	No	Read signal to the <code>altasmi_parallel</code> component. Available when the <code>check_app_pof</code> parameter is set to TRUE. A logic high on this pin initiates the read operation on the ALTASMI_PARALLEL IP core. Wire this pin to the <code>asmi_read</code> input port of the ALTASMI_PARALLEL IP core.
asmi_rden	Output	No	Read enable signal to the <code>altasmi_parallel</code> component. Available when the <code>check_app_pof</code> parameter is set to TRUE. This pin enables the read operation on the ALTASMI_PARALLEL IP core. Wire this pin to the <code>asmi_rden</code> input port of the ALTASMI_PARALLEL IP core.

Parameters

Table 10: Parameter Type and Corresponding Parameter Bit Width Mapping for Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices

Bit	Parameter	Width	Comments
000	Reconfiguration trigger conditions (Read Only)	5	<ul style="list-style-type: none"> Bit 4—<code>wdtimer_source</code>: User watchdog timer timeout. Bit 3—<code>nconfig_source</code>: External configuration reset (<code>nCONFIG</code>) assertion. Bit 2—<code>runconfig_source</code>: Configuration reset triggered from logic array. Bit 1—<code>nstatus_source</code>: <code>nSTATUS</code> asserted by an external device as the result of an error. Bit 0—<code>crcerror_source</code>: CRC error during application configuration. <p>The POR value for all bits are 0.</p>
001	Illegal Value		

Bit	Parameter	Width	Comments
010	Watchdog Timeout Value	12	—
011	Watchdog Enable	1	—
100	Page Select	24 or 32	<p>For the Quartus II software version 13.1 and onwards:</p> <ul style="list-style-type: none"> Width of 24 or 32 when reading and writing the start address. For active serial devices using 24-bit addressing, such as EPCS128 or EPCQ128, PGM[23 . . 2] corresponds to the upper 22 bits of the 24-bits start address. PGM[1 . . 0] is read as 2' b0. For active serial devices using 32-bit addressing, such as EPCQ256, PGM[31 . . 2] corresponds to the upper 30 bits of the 32-bits start address. PGM[1 . . 0] is read as 2' b0. <p>For the Quartus II software version 13.0 and earlier:</p> <ul style="list-style-type: none"> Width of 24 when reading and writing the start address. For Arria II, and Stratix IV devices, PGM[23 . . 0] form the 24-bit start address. For Arria V, Cyclone V, and Stratix V devices, if you use active serial devices using 24-bit addressing, such as EPCS128 or EPCQ128, PGM[23 . . 0] corresponds to the 24 bits of the start address. If you use active serial devices using 32-bit addressing, such as EPCQ256, PGM[23 . . 0] corresponds to the 24 MSB of the start address, thus the 32 bits start address is PGM[23 . . 0] , 8' b0.
101	Configuration Mode (AnF)	1	<p>This parameter is set to 1 in application page and is set to 0 in factory page. In remote update mode, this parameter can be read and written.</p> <p>Before loading the application page in remote update mode, Altera recommends that you set this parameter to 1. The content of the control register cannot be read properly if you fail to do so.</p>
110			Illegal Value
111			Illegal Value

Avalon-MM Interface

The Avalon-MM interface in Altera Remote Update IP core is not supported in Stratix II devices.

Control Status Register Signals

Table 11: Altera Remote Update IP Core Avalon-MM Control Status Register Signals for Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices

Name	Width	Direction	Description
clk	1	Input	Clock input
reset	1	Input	Reset input
avl_csr_address	3	Input	Address bus
avl_csr_read	1	Input	Perform a read transaction
avl_csr_write	1	Input	Perform a write transaction
avl_csr_readdata	32	Output	Read data from IP
avl_csr_readdata_valid	1	Output	Indicate when read data is valid
avl_csr_writedata	32	Input	Write data to IP
avl_csr_waitrequest	1	Output	Waitrequest signal high indicates the core is busy

Control Status Register Write Operation

To execute the write operation for control the status register, perform the following steps:

1. Asserts the `avl_csr_write` high.
2. Write a correct address of the register in the `avl_csr_address` bus. Refer to the Register Map for register information.
3. Write data into the `avl_csr_writedata` bus.

Related Information

[Register Map](#) on page 24

Control Status Register Read Operation

To execute the read operation for the control status register, perform the following steps:

1. Asserts `avl_csr_read` high.
2. Write a correct address of the register in the `avl_csr_address` bus. Refer to the Register Map for register information.
3. Wait for the `avl_csr_readdata_valid` signal to go high.
4. Retrieve read data from `avl_csr_readdata`.

Register Map

Table 12: Altera Remote Update IP Core Avalon-MM Register Map for Arria V, Cyclone V, Stratix IV, and Stratix V Devices

- The IP core can read or write each field separately as each command has different parameter value.
- The default value for the registers is 0.

Register Name	Address Offset	Width	R/W	Description
RU_RECONFIG_TRIGGER_CONDITIONS	0x0	5	Read	<p>Read configuration trigger conditions.</p> <ul style="list-style-type: none"> • Bit 4—<code>wdtimer_source</code>: Users watchdog timer timeout • Bit 3—<code>nconfig_source</code>: External configuration reset (<code>nCONFIG</code>) assertion. • Bit 2—<code>runconfig_source</code>: Configuration reset triggered from logic array • Bit 1—<code>nstatus_source</code>: <code>nSTATUS</code> asserted by an external device as the result of an error • Bit 0—<code>crcerror_source</code>: CRC error during application configuration.
RU_WATCHDOG_TIMEOUT	0x1	12	Read/Write	<p>Read or write watchdog timeout value.</p>
RU_WATCHDOG_ENABLE	0x2	1	Read/Write	<p>Enable or disable watchdog timeout.</p> <ul style="list-style-type: none"> • 0: Disable • 1: Enable
RU_PAGE_SELECT	0x3	24 or 32	Read/Write	<p>Read or write start address of configuration image.</p>
RU_CONFIGURATION_MODE	0x4	1	Read/Write	<p>Write configuration mode set to 1 in application page and 0 in factory page.</p>
RU_RESET_TIMER	0x5	1	Write	<p>Write a value of 1 to this register to trigger reset timer of the remote update. The IP will automatically trigger a reset pulse to reset timer pin of the remote update.</p>
RU_RECONFIG	0x6	1	Write	<p>Write a value of 1 to this register to trigger reconfiguration from a new image. The IP will set 1 to <code>reconfig</code> pin of the remote update and hold this value until the process done.</p>

Enabling Remote System Upgrade Circuitry

To enable the remote system upgrade feature, follow these steps:

1. Select **Active Serial x1/x4** or **Configuration Device** from the Configuration scheme list in the **Configuration** page of the **Device and Pin Options** dialog box in the Quartus Prime software.
2. Select **Remote** from the Configuration mode list in the **Configuration** page of the **Device and Pin Options** dialog box in the Quartus Prime software.

Enabling this feature automatically turns on the **Auto-restart configuration after error** option.

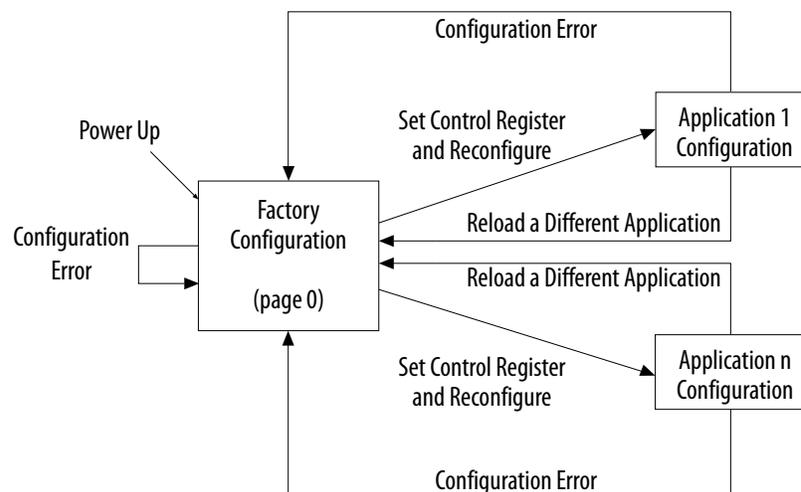
Altera Remote Update IP core provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read and write protocol in the device logic.

Cyclone IV Devices

Remote System Configuration Mode

Remote Configuration Mode

Figure 10: Remote Configuration Mode



Only Cyclone IV E devices support both the active parallel (AP) and active serial (AS) configuration scheme for remote system upgrade. Other Cyclone IV devices support only AS configuration scheme for remote system upgrade.

When using with EPCS or EPCQ devices, the remote update mode allows a configuration space to start at any flash sector boundary, allowing a maximum of 128 pages in the EPCS64 device and 32 pages in the EPCS16 device, in which 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.

Remote System Configuration Components

Table 13: Remote System Configuration Components in Cyclone IV Devices

Components	Details
Page mode feature	<p>For both AS and AP configurations, Cyclone IV devices use a 24-bit boot start address in which you set the most significant 22 bits. Cyclone IV devices do not support <code>pgmout</code> ports.</p>
Factory configuration	<p>Factory configuration is the default configuration setup.</p> <p>In remote configuration mode, the factory configuration loads into Cyclone IV devices upon power-up.</p> <p>If a system encounters an error while loading application configuration data or if the device reconfigures due to <code>nCONFIG</code> assertion, the device loads the factory configuration. The remote system configuration register determines the reason for factory configuration. Based on this information, the factory configuration determines which application configuration to load.</p> <p>Upon power-up in remote update in the AP configuration scheme, Cyclone IV devices load the default factory configuration located at the following address:</p> <pre>boot_address[23:0] = 24'h010000 = 24'b1 0000 0000 0000 0000.</pre> <p>You can change the default factory configuration address to any address using the <code>APFC_BOOT_ADDR</code> JTAG instruction. The factory image is stored in non-volatile memory and is never updated or modified using remote access. This corresponds to the default start address location <code>0x010000</code> (or the updated address if the default address is changed) in the supported parallel flash memory. Note that <code>0x010000</code> is the 16-bit word address for the AP flash memory. However, the Quartus Prime software implements 8-bit byte addressing. Therefore, the correct Quartus Prime software setting for this address is <code>0x020000</code>.</p>
Application configuration	<p>The application configuration is the configuration data from a remote source and the data is stored in different locations or pages of the memory storage device, excluding the factory default page.</p>

Components	Details
Watchdog timer	<p>A watchdog timer is a circuit that determines the functionality of another mechanism. The watchdog timer functions like a time delay relay that remains in the reset state while an application runs properly.</p> <p>The devices are equipped with a built-in watchdog timer for remote system configuration to prevent a faulty application configuration from indefinitely stalling the device.</p> <p>The timer is a 29-bit counter, but you use only the upper 12 bits to set the value for the watchdog timer.</p> <p>The timer begins counting after the device goes into user mode. If the application configuration does not reset the user watchdog timer before time expires, the dedicated circuitry reconfigures the device with the factory configuration and resets the user watchdog timer.</p> <p>To ensure the application configuration is valid, you must continuously reset the watchdog <code>reset_timer</code> within a specific duration during user mode operation.</p>
Remote update sub-block	<p>The remote update sub-block manages the remote configuration feature. A remote configuration state machine controls this sub-block. This sub-block generates the control signals required to control the various configuration registers.</p>
Remote configuration registers	<p>The remote configuration registers keep track of page addresses and the cause of configuration errors. You can control both the update and shift registers. The status and control registers are controlled by internal logic, but are read via the shift register.</p> <p>The remote system upgrade status register has additional capabilities. Three sets of registers store the status for the current application configuration and the two previous application configurations.</p> <p>For details about configuration registers, refer to the Configuration, Design Security, and Remote System Upgrades chapter in the respective device handbook.</p>

Parameter Settings

Table 14: Altera Remote Update IP core Parameters for Cyclone IV Devices

GUI Name	Legal Value in GUI	Description
Which operation mode will you be using?	REMOTE	Specifies the configuration mode of the Altera Remote Update IP core.

GUI Name	Legal Value in GUI	Description
Which configuration device will you be using?	<ul style="list-style-type: none"> EPCS device EPCQ device 	Choose the configuration device that you are using.
Add support for writing configuration parameters	—	Enable this if you need to write configuration parameters.
Enable reconfig POF checking	—	Allows you to enable .pof checking, which allows the remote update block to verify the existence of an application configuration image before the image is loaded. When you turn on this parameter, the Altera Remote Update IP core checks the .pof and sends the <code>reconfig</code> signal. This option is disabled by default.

Ports

Table 15: Altera Remote Update IP Core Ports for Cyclone IV Devices

Name	Port	Required?	Description
<code>read_param</code>	Input	No	<p>Read signal for the parameter specified in <code>param[]</code> input port and fed to <code>data_out[]</code> output port.</p> <p>Signal indicating the parameter specified on the <code>param[]</code> port should be read. The number of bits set on <code>data_out[]</code> depends on the parameter type. The signal is sampled at the rising clock edge. Assert the signal for only one clock cycle to prevent the parameter from being read again in a subsequent clock cycle.</p> <p>The busy signal is activated as soon as <code>read_param</code> is read as active. While the parameter is being read, the busy signal remains asserted, and <code>data_out[]</code> has invalid data. When the busy signal is deactivated, <code>data_out[]</code> becomes valid and another parameter can be read.</p>

Name	Port	Required?	Description
write_param	Input	No	<p>Write signal for parameter specified in <code>param[]</code> and with value specified in <code>data_in[]</code>.</p> <p>Signal indicating parameter specified with <code>param[]</code> should be written into remote update block with the value specified in <code>data_in[]</code>. The number of bits read from <code>data_in[]</code> depends on the parameter type.</p> <p>The signal is sampled at the rising clock edge. The signal should be asserted for only one clock cycle to prevent the parameter from being rewritten on a subsequent clock cycle. The busy signal is activated as soon as <code>write_param</code> is read as being active. While the parameter is being written, the busy signal remains asserted, and input to <code>data_in[]</code> is ignored. When the busy signal is deactivated, another parameter can be written. This signal is only valid in factory configuration mode because parameters cannot be written in application configuration mode.</p>
param[]	Input	No	<p>Bus that specifies which parameter need to be read or updated.</p> <p>A 3-bit bus that selects the parameter to be read or updated. If left unconnected, the default value for this port is 000.</p> <p>For more information, refer to Parameters on page 34.</p>
data_in[]	Input	No	<p>Data input for writing parameter data into the remote update block. Input bus for parameter data.</p> <p>For some parameters, not all bits are used. In this case, the lower-order bits are used (for example, status values use bits [4:0]).</p> <p>If left unconnected, this bus defaults to 0. The port is ignored if the current configuration is the application configuration.</p> <p>For the Quartus II software version 13.0 or earlier, the bus width is 22-bit. For the Quartus II software version 13.1 and onwards, the bus widths are as follow:</p> <ul style="list-style-type: none"> • 24-bit bus width—using 3-byte addressing configuration device, for example EPCS128. • 32-bit bus width—using 4-byte addressing configuration device, for example EPCQ256.

Name	Port	Required?	Description
reconfig	Input	Yes	Signal indicating that reconfiguration of the part should begin using the current parameter settings. A value of 1 indicates reconfiguration should begin. This signal is ignored while busy is asserted to ensure all parameters are completely written before reconfiguration begins.
reset_timer	Input	No	Reset signal for the watchdog timer. Signal indicating the internal watchdog timer should be reset. Unlike other inputs, this signal is not affected by the busy signal and can reset the timer even when the busy signal is asserted. A falling edge of this signal triggers a reset of the user watchdog timer. For the timing specification of this parameter, refer to the specific device handbook.
read_source	Input	Yes	Specifies whether a parameter value is read from the current or a previous state. This 2-bit port specifies the state from which a parameter value is read. This signal is valid only when the read_param signal is valid. Mapping read_source[1..0] to Selected Source is defined as follow: <ul style="list-style-type: none"> 00 - Current State Content in Status Register 01 - Previous State Register 1 Content in Status Register 10 - Previous State Register 2 Content in Status Register 11 - Value in Input Register For details, refer to the Configuration, Design Security, and Remote System Upgrades chapter in the respective device handbook.
clock	Input	Yes	Clock input to the remote update block. Clock input to control the machine and to drive the remote update block during the update of parameters. This port must be connected to a valid clock.

Name	Port	Required?	Description
reset	Input	Yes	<p>This is an active high signal. Asserting this signal high will reset the IP core.</p> <p>Asynchronous reset input to the IP core to initialize the machine to a valid state. The machine must be reset before first use, otherwise the state is not guaranteed to be valid.</p>
busy	Output	No	<p>Busy signal that indicates when remote update block is reading or writing data.</p> <p>While this signal is asserted, the machine ignores most of its inputs and cannot be altered until the machine deasserts this signal. Therefore, changes are made only when the machine is not busy.</p> <p>This signal goes high when <code>read_param</code> or <code>write_param</code> is asserted, and remains high until the read or write operation completes.</p>
data_out[]	Output	No	<p>Data output when reading parameters.</p> <p>This bus holds read parameter data from the remote update block. The <code>param[]</code> value specifies the parameter to read. When the <code>read_param</code> signal is asserted, the parameter value is loaded and driven on this bus. Data is valid when the busy signal is deasserted.</p> <p>If left unconnected, the default value for the port is 000. The width of this bus is device-dependent:</p> <p>For the Quartus II software version 13.0 or earlier, the bus width is 29-bit. For the Quartus II software version 13.1 and onwards, the bus widths are as follow:</p> <ul style="list-style-type: none"> • 29-bit bus width—using 3-byte addressing configuration device, for example EPCS128. • 32-bit bus width—using 4-byte addressing configuration device, for example EPCQ256.
asmi_busy	Input	No	<p>Input from the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>true</code>.</p> <p>A logic high on this pin indicates that the <code>ALTASMI_PARALLEL</code> IP core is busy processing the operation. The Altera Remote Update IP core waits for this pin to go low before initiating another operation.</p> <p>Wire this pin to the <code>asmi_busy</code> output port of the <code>ALTASMI_PARALLEL</code> IP core.</p>

Name	Port	Required?	Description
asmi_data_valid	Input	No	<p>Input from the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>true</code>.</p> <p>A logic high on this pin indicates valid data in the <code>asmi_dataout[7..0]</code> output port of the ALTASMI_PARALLEL IP core.</p> <p>Wire this pin to the <code>asmi_data_valid</code> output port of the ALTASMI_PARALLEL IP core.</p>
asmi_dataout	Input	No	<p>Input from the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>true</code>.</p> <p>The Altera Remote Update IP core presents the address information on this pin before initiating the read operation on the ALTASMI_PARALLEL IP core.</p>
pof_error	Output	No	<p>Detects an invalid application configuration image.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>TRUE</code>.</p> <p>A logic high on this pin indicates that the Altera Remote Update IP core detects an invalid application configuration image. If asserted high, you must take corrective action by reloading a new application configuration image or specifying a different address location in the EPCS or EPCQ that contains a valid application configuration image. Wire this pin based on your system requirement.</p>
asmi_addr	Output	No	<p>Address signal to the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to <code>TRUE</code>. The Altera Remote Update IP core presents the address information on this pin before initiating the read operation on the ALTASMI_PARALLEL IP core.</p> <p>Wire this pin to the <code>asmi_addr</code> input port of the ALTASMI_PARALLEL IP core.</p>

Name	Port	Required?	Description
asmi_read	Output	No	<p>Read signal to the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to TRUE. A logic high on this pin initiates the read operation on the ALTASMI_PARALLEL IP core.</p> <p>Wire this pin to the <code>asmi_read</code> input port of the ALTASMI_PARALLEL IP core.</p>
asmi_rden	Output	No	<p>Read enable signal to the <code>altasmi_parallel</code> component.</p> <p>Available when the <code>check_app_pof</code> parameter is set to TRUE. This pin enables the read operation on the ALTASMI_PARALLEL IP core.</p> <p>Wire this pin to the <code>asmi_rden</code> input port of the ALTASMI_PARALLEL IP core.</p>

Parameters

Table 16: Mapping to Each Parameter Type and Corresponding Parameter Bit Width for Cyclone IV Devices

Bit	Parameter	Width	Comments
000	Master State Machine Current State Mode (Read Only)	2	<p>00—Factory mode.</p> <p>01—Application mode.</p> <p>11—Application mode with the master state machine user watchdog timer enabled.</p>
001	Force early CONF_DONE (<code>cd_early</code>) check	1	—
010	Watchdog Timeout Value	12	<p>Width of 12 when writing.</p> <p>The 12 bits for writing are the upper 12 bits of the 29-bit Watchdog Timeout Value</p>
		29	Width of 29 when reading.
011	Watchdog Enable	1	—

Bit	Parameter	Width	Comments
100	Boot Address	—	<p>For the Quartus II software version 13.1 and onwards:</p> <ul style="list-style-type: none"> Width of 29 or 32 when reading the boot address. Width of 24 or 32 when writing the boot address. For active serial devices using the 24-bit addressing, such as EPCS128 or EPCQ128, <code>boot_address[23..2]</code> corresponds to the upper 22 bits of the 24-bits boot address. <code>boot_address[1..0]</code> is read as 2'b0. For active serial devices using the 32-bit addressing, such as EPCQ256, <code>boot_address[31..2]</code> corresponds to the upper 30 bits of the 32-bits boot address. <code>boot_address[1..0]</code> is read as 2'b0. <p>For the Quartus II software version 13.0 or earlier:</p> <ul style="list-style-type: none"> Width of 24 when reading the boot address. Width of 22 when writing the boot address. Writes the boot address to the upper 22 bits of the 24-bits boot address.
101	Illegal Value		
110	Force the internal oscillator as startup state machine clock (<code>osc_int</code>) option bit	1	—
111	Reconfiguration trigger conditions (Read Only)	5	<p>Bit 4 (<code>nconfig_source</code>)—external configuration reset (<code>nconfig</code>) assertion.</p> <p>Bit 3 (<code>crcerror_source</code>)—CRC error during application configuration.</p> <p>Bit 2 (<code>nstatus_source</code>)—<code>nstatus</code> asserted by an external device as the result of an error.</p> <p>Bit 1 (<code>wdtimer_source</code>)—User watchdog timer timeout.</p> <p>Bit 0 (<code>runconfig_source</code>)—Configuration reset triggered from logic array.</p>

Remote Update Operation

The operation defined in the Remote Update Operation column should only be performed in the corresponding master state machine (MSM) mode.

Table 17: Cyclone IV Devices Remote Update Operation

Note: `read_source` specifies whether a parameter value is read from the current or a previous state. For more information, refer to [Table 18](#).

read_param	write_param	read_source	param	Remote Update Operation	data_out_width (bits)	MSM Mode
1	0	[00]	[000]	Master State Machine Current State Mode (Read Only) <ul style="list-style-type: none"> • 00—Factory mode • 01—Application mode • 11—Application mode with master state machine user watchdog timer enabled 	2	Factory or Application
1	0	[00]	[100]	Read factory boot address	24	Factory
1	0	[01]	[100]	Read Past Status 1 boot address. For more information, refer to Figure 11 .	24	Factory
1	0	[01]	[111]	Read Past Status 1 reconfiguration trigger condition source. For more information, refer to Figure 11 .	5	Factory
1	0	[10]	[100]	Read Past Status 2 boot address. For more information, refer to Figure 11 .	24	Factory
1	0	[10]	[111]	Read Past Status 2 reconfiguration trigger condition source For more information, refer to Figure 11 .	5	Factory
1	0	[01]	[010]	Read current application mode watchdog value	29	Application
1	0	[01]	[011]	Read current application mode watchdog enable	1	Application
1	0	[10]	[100]	Read current application mode boot address	24	Application
0	1	[00]	[001]	Write the early <code>confdone</code> check bit. All parameters can be written in factory mode only.	1	Factory
0	1	[00]	[010]	Write the watchdog time-out value. All parameters can be written in factory mode only.	12	Factory

read_param	write_param	read_source	param	Remote Update Operation	data_out width (bits)	MSM Mode
0	1	[00]	[011]	Write the watchdog enable bit. All parameters can be written in factory mode only.	1	Factory
0	1	[00]	[100]	Write application boot address. All parameters can be written in factory mode only.	22	Factory
0	1	[00]	[110]	Write to force the internal oscillator as startup state machine clock. All parameters can be written in factory mode only.	1	Factory
1	0	[11]	[001]	Read the early <code>confdone</code> check bits	1	Factory
1	0	[11]	[010]	Read watchdog time-out value	12	Factory
1	0	[11]	[011]	Read watchdog enable bit	1	Factory
1	0	[11]	[100]	Read boot address	22	Factory
1	0	[11]	[110]	Read to check whether the internal oscillator is set as startup state machine clock	1	Factory

read_source

The following table lists the details for `read_source`. `read_source` specifies whether a parameter value is read from the current or a previous state. When you trigger the read operation, all contents in the status register or input register latched to the `data_out` node in the Altera Remote Update IP core.

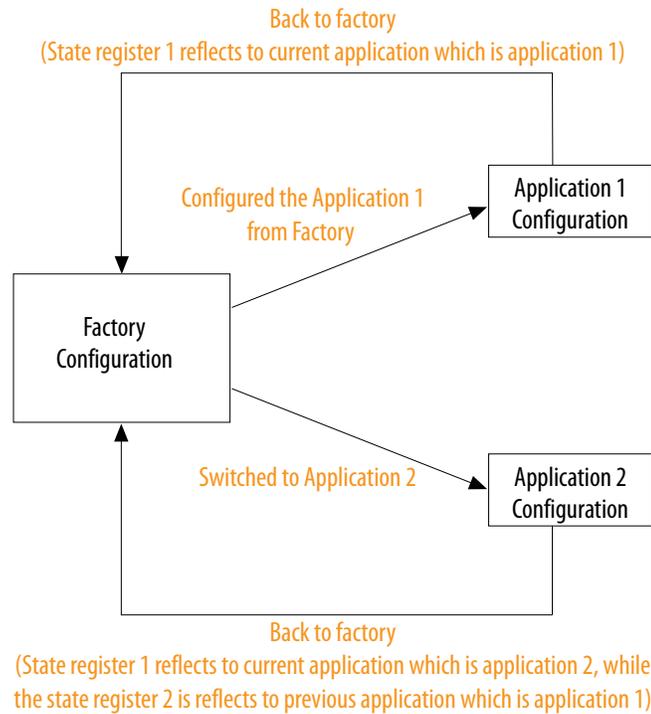
Table 18: read_source

read_source	Description
00	Current state contents in status register
01	Previous state register 1 contents in status register
10	Previous state register 2 contents in status register
11	Current contents is in input register

State Register

The previous state register 1 reflects the current application configuration and the previous state register 2 reflects the previous application configuration.

Figure 11: State Register



Avalon-MM Interface

Control Status Register Signals

Table 19: Altera Remote Update IP Core Avalon-MM Control Status Register Signals for Cyclone IV Devices

Name	Width	Direction	Description
clk	1	Input	Clock input
reset	1	Input	Reset input
avl_csr_address	3	Input	Address bus
avl_csr_read	1	Input	Perform a read transaction
avl_csr_write	1	Input	Perform a write transaction
avl_csr_readdata	32	Output	Read data from IP
avl_csr_readdata_valid	1	Output	Indicate when read data is valid
avl_csr_writedata	32	Input	Write data to IP
avl_csr_waitrequest	1	Output	Waitrequest signal high indicates the core is busy

Control Status Register Write Operation

To execute the write operation for the control status register, perform the following steps:

1. Asserts the `avl_csr_write` high.
2. Write a correct address of the register in the `avl_csr_address` bus. Refer to the Register Map for register information.
3. Write data into the `avl_csr_writedata` bus.

Related Information

[Register Map](#) on page 39

Control Status Register Read Operation

To execute the read operation for the control status register, perform the following steps:

1. Asserts `avl_csr_read` signal high.
2. Write a correct address of the register in the `avl_csr_address` bus. Refer to the Register Map for register information.
3. Wait for the `avl_csr_readdata_valid` signal to go high.
4. Retrieve read data from `avl_csr_readdata`.

Register Map

Table 20: Altera Remote Update IP Core Avalon-MM Register Map for Cyclone IV Devices

- The last two bits of an address represents the `read_source` signals.
- You have to write the correct address offset to carry `read_source` value as shown in the Read Source Mapping table.
- The IP core combines the address bus of control status register interface to the `read_source` parameter.
- The default value for the registers is 0.

Register Name	Address Offset	Width	R/W	Description
RU_MASTER_SM_CURRENT_STATE_MODE	0x0	2	Read	Read current state of the state machine 00: Factory mode 01: Application mode 11: Application mode with the master state machine user watchdog timer enabled.
RU_FORCE_EARLY_CONF_DONE	0x4	1	Read/Write	Force early <code>CONF_DONE</code>
RU_WATCHDOG_TIMEOUT	0x8	29 or 12	Read/Write	Read or write watchdog timeout value. <ul style="list-style-type: none"> • 12 bit wide when writing • 29 bit wide when reading

Register Name	Address Offset	Width	R/W	Description
RU_WATCHDOG_ENABLE	0xC	1	Read/Write	Enable or disable watchdog timeout. <ul style="list-style-type: none"> 0: Disable 1: Enable
RU_BOOT_ADDRESS	0x10	24, 29 or 32	Read/Write	<ul style="list-style-type: none"> 29 or 32 bit wide (EPCQ 32 bit addressing) when reading boot address. 24 or 32 bit wide when writing the boot address.
RU_FORCE_INTERNAL_OSC	0x14	1	Read/Write	Force the internal oscillator as startup state machine clock (osc_int) option bit
RU_RECONFIG_TRIGGER_CONDITIONS	0x18	5	Read	Read configuration trigger conditions. <ul style="list-style-type: none"> Bit 4—nconfig_source: External configuration reset (nCONFIG) assertion. Bit 3—crcerror_source: CRC error during application configuration. Bit 2—nstatus_source: nSTATUS asserted by an external device as the result of an error Bit 1—wdtimer_source: Users watchdog timer timeout Bit 0—runconfig_source: Configuration reset triggered from logic array
RU_RESET_TIMER	0x1C	1	Write	Write a value of 1 to this register to trigger reset timer of the remote update. The IP core will automatically trigger a reset pulse to reset timer pin of the remote update.
RU_RECONFIG	0x1D	1	Write	Write to this address with value of 1 to trigger reconfiguration from a new image. The IP core will set 1 to reconfig pin of the remote update and hold this value until the process done.

Read Source Mapping

Table 21: Read Source Mapping

- Table shows the address offset with their read source value.
- These combinations are used to describe all supported operations in the Control Status Register Signals.

Name	Address offset	Read source value
RU_MASTER_SM_CURRENT_STATE_MODE	0x0	00
	0x1	01
	0x2	10
	0x3	11
RU_FORCE_EARLY_CONF_DONE	0x4	00
	0x5	01
	0x6	10
	0x7	11
RU_WATCHDOG_TIMEOUT	0x8	00
	0x9	01
	0xA	10
	0xB	11
RU_WATCHDOG_ENABLE	0xC	00
	0xD	01
	0xE	10
	0xF	11
RU_BOOT_ADDRESS	0x10	00
	0x11	01
	0x12	10
	0x13	11
RU_FORCE_INTERNAL_OSC	0x14	00
	0x15	01
	0x16	10
	0x17	11

Name	Address offset	Read source value
RU_RECONFIG_TRIGGER_CONDITIONS	0x18	00
	0x19	01
	0x1A	10
	0x1B	11
RU_RESET_TIMER	0x1C	N/A
RU_RECONFIG	0x1D	N/A

Enabling Remote System Upgrade Circuitry

To enable remote update in the compiler settings of the project, perform the following steps:

1. On the **Assignments** menu, click **Device**.
2. In the **Settings** dialog box, Click **Device and Pin Options**.
3. In the **Device and Pin Options** dialog box , click the **Configuration** tab.
4. From the **Configuration Mode** list, select **Remote**.
5. Click **OK**.
6. In the **Settings** dialog box, click **OK**.

Flash Memory Programming Files

You can program the flash memory, EPCS, EPCQ, and EPCQ-L using the JTAG interface or Active Serial interface. Depending on the interface, you need to generate either a JTAG indirect configuration (`.jic`) file or a raw programming data (`.rpd`) file.

Table 22: Flash Memory Programming Files Based on Programming Interface

Programming Interface	Flash Memory Programming File Used	Description
JTAG Interface	<code>.jic</code>	The <code>.jic</code> file instantiates the Serial Flash Loader IP core in the design to form a bridge between the flash and the JTAG Interface.
Active Serial Interface	<code>.rpd</code>	Programming data is transferred directly between the flash and download cable.

To update the application image only, you can do either one of the following:

- Recompile the `.jic` file and choose new application image only in the convert programming file tool.
- Generate the `.rpd` file and program the EPCQ-L with ASMI IP or external controller.

Related Information

- [Using the Altera Serial Flash Loader IP Core with the Quartus II Software](#)
- [Altera ASMI Parallel IP Core User Guide](#)

Design Examples

Arria 10 Remote Update Design Example

This Arria 10 design example uses the Avalon-MM interface. Altera uses the following hardware and software to create the design example:

- Quartus II Version : 15.0
- Arria 10 Development Kit with 10AX115S3F45I2SGE2 FPGA Device

Follow these steps to perform the design example tasks:

1. Unzip the contents of the design example to your working directory on your PC.
2. Convert the three `.sof` files into one `.jic` by using Convert Programming File. On the **File** Menu, click **Convert Programming Files** and select the details as shown below:
 - Programming File type: **JTAG Indirect Configuration File (.jic)**.
 - Select Configuration Device: **EPCQL1024**.
 - Mode: **Active Serial**.
 - Set the file name you your desired location.
 - Flash loader: click **add device** and choose **10AX115S2E2**.
 - SOFT DATA PAGE_0: click **Add File** and select the factory image with start address set to **<auto>**.
 - SOFT DATA PAGE_1: click **Add File** and select the application image file with start address `0x2000000`. Compression is enabled for this application image file.
 - SOFT DATA PAGE_2: click **Add File** and select the application image file with start address `0x4000000`. Compression is enabled for this application image file.
 - Click **Generate**.
 - Click **OK** when the dialogue box of `.jic` file successfully generated appears.
3. Please follow the steps below to run the simple design:
 - a. After programming the `.jic` file, power cycle the board, all LED is lighted up. It indicates you are currently at factory image.
 - b. Go to system console and direct to the directory where your `FI_SysConsole_try.tcl` is located. Type source `FI_SysConsole_try.tcl`.
Only one LED is lighted up which indicates successfully go to application image 1. After the watchdog timeout, all LED will light up and go back to factory image.
Note: To go to application image 2 directly form the factory image, comment out the `write boot address` to `App1` and uncomment the `write boot address App2` in the `FI_SysConsole_try.tcl` file.
4. Setting Boot Page Selection for design with more than one SOF page:
 - a. To select the boot page, click the **Option/Boot Info** button in **Convert Programming File**.
 - b. In the **Active Serial Boot Info** window, select the page available from the **Boot from page** drop down menu. By default, the page number will be set at **page_0**.
 - c. For application to application image, change the page number to **page_1** or **page_2**.

Related Information

[Arria 10 Remote System Update with Avalon-MM Interface Design Example](#)

Cyclone V Remote Update Design Example

Altera uses the following hardware and software to create the design example:

- Quartus II Version : 13.0
- Cyclone V Development Kit with 5CEFA7F31C7ES FPGA Device

Follow these steps to perform the design example tasks:

1. Unzip the contents of the design example to your working directory on your PC.
2. In the Quartus II software, click **Open Project** in the **File** menu.
3. Compile the application image:
 - a. Browse to the folder in which you unzipped the files and open the `Application_Image.qpf`.
 - b. Click **Yes** in the message box "Do you want to overwrite the database for C:/your working directory/Application_Image.qpf created by Quatus II 64-Bit Version 13.0.a Build 232 Service Pack 1 SJ Full version?"
 - c. On the **Processing** menu, choose **Start Compilation**.
 - d. Click **OK** when the full compilation successful dialog box appears. The `Application_Image.sof` will be generated in `c:\your working directory\output_files`.
 - e. Click **close project** in the **file** menu.
4. Compile the factory image:
 - a. Browse to the folder in which you unzipped the files and open the `SVRSU.qpf`.
 - b. Click **Yes** in the message box "Do you want to overwrite the database for C:/your working directory/Application_Image.qpf created by Quatus II 64-Bit Version 13.0.a Build 232 Service Pack 1 SJ Full version?"
 - c. Choose **Start Compilation** on the **Processing** menu.
 - d. Click **OK** when the full compilation successful dialog box appears. The `Factory_Image.sof` will be generated in `c:\your working directory\output_files`.
5. On the **File** Menu, click **Convert Programming Files** and select the details as shown below:
 - Programming File type: **JTAG Indirect Configuration File (.jic)**
 - Select Configuration Device: **EPCQ 128**
 - Mode: **Active Serial x4**
 - File name: `c:/your working directory/output_file.jic`
 - Flash loader: click **add device** and choose **5CEFA7ES**
 - SOFT DATA PAGE_0: click **Add File** and select the factory image file (`SVRSU.sof`)
 - SOFT DATA PAGE_1: click **Add File** and select the Application image file (`Application_Image.sof`)
 - Click **Generate**.
 - Click **OK** when the dialogue box of .jic file successfully generated appears.
6. On the Tool Menu, click **Programmer** and follow these steps:
 - a. Make sure the board is power up and the USB Blaster is connected between computer and the board. This design example uses the USB Blaster and JTAG mode.
 - b. Click **Auto Detect**.
 - c. **Right-click** on the **5CEFA7ES** and select **change file**.
 - d. Browse to the `output_file.jic` that was generated in previous steps.
 - e. Turn on the **Program/Configure** checkbox and click **Start**.
 - f. Configuration successful indicates the FPGA is configured successfully.

Related Information

- [Cyclone V Remote System Update Design Example](#)
- [AN 603: Active Serial Remote System Upgrade Reference Design](#)
Provides more information about Arria II GX, Stratix III and Stratix IV devices reference design.

Altera Remote Update IP Core User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
16.0	Altera Remote Update IP Core User Guide
15.1	Altera Remote Update IP Core User Guide
15.0	Altera Remote Update IP Core User Guide

Document Revision History

Table 23: Document Revision History

Date	Version	Changes
April 2017	2017.04.10	<ul style="list-style-type: none"> • Updated <i>Transitions Between Factory and Application Configurations in Remote Update Mode</i> figure. • Removed redundant statement in <i>Avalon-MM in Altera Remote Update IP Core</i>. • Updated note in <i>Remote Update Operation</i> table.
October 2016	2016.10.31	<ul style="list-style-type: none"> • Updated <i>Parameters</i> table for Arria 10 devices and Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V Devices. • Added Add support for Avalon Interface parameter in <i>Parameter Settings</i> tables.
June 2016	2016.06.01	<ul style="list-style-type: none"> • Updated Convert Programming File settings in <i>Arria 10 Remote Update Design Example</i>. • Removed references to Local update mode features.

Date	Version	Changes
May 2016	2016.05.02	<ul style="list-style-type: none"> Added recommended F_{MAX} for Altera Remote Update IP core. Added missing title for <i>Cyclone IV Devices Remote Update Operation</i> table. Added note on possible PCIe timing violation when using direct-to-application. Added note on recommending user to set a fixed configuration image start address. Added <i>Enabling Remote System Upgrade</i> subsection.
December 2015	2015.12.14	Updated RU_RECONFIG_TRIGGER_CONDITIONS description for Cyclone V Avalon-MM interface register.
November 2015	2015.11.17	<ul style="list-style-type: none"> Updated Page description in Convert Programming Files setting from SOFT DATA PAGE_0 to SOFT DATA PAGE_1. Added high level configuration scheme block diagram for remote system upgrade. Added <i>Flash Memory Programming Files</i>. Added design example for Arria 10 devices.
June 2015	2015.06.15	<ul style="list-style-type: none"> Added Avalon-MM interface support for Quartus II Software version 15.0. Added table for Avalon-MM interface Control Status Registers and register map definitions. Added steps for read and write operation for Avalon-MM interface. Added example waveforms for read, write, reset timer, ctl_nupdt, and reconfiguration operation for Arria 10 devices. Updated the Device Support section to include information on device families that will be phased out from Quartus II software version 15.0.
April 2015	2015.04.07	Added design example link.
January 2015	2015.01.23	Updated Arria 10 remote system configuration mode flow diagram.
December 2014	2014.12.15	<ul style="list-style-type: none"> Updated POF checking feature description and invalid configuration image examples. Added Arria 10 device support with descriptions, ports and parameters. Replaced outdated design examples with a current application design example.

Date	Version	Changes
June 2014	2014.06.30	<ul style="list-style-type: none">• Replaced MegaWizard Plug-In Manager information with IP Catalog.• Added standard information about upgrading IP cores.• Added standard installation and licensing information.• Removed outdated device support level information. IP core device support is now available in IP Catalog and parameter editor.• Added a note to recommend users to use 20-MHz f_{MAX} for all devices.
May 2014	2014.05.13	<ul style="list-style-type: none">• Updated the Device Support section to include information on device families that will be phased out from Quartus II software version 13.1 and Quartus II software version 14.0.• Rearranged content for remote system configuration modes, remote system configuration components, parameter settings, ports, param for each group of devices. Refer to Device Support section for more information.
August 2013	2013.08.16	Added Cyclone IV devices support for Active Serial Remote Configuration Mode in Parameters, Output Ports, and Active Serial Remote Configuration Mode.

Date	Version	Changes
July 2013	2013.07.12	<ul style="list-style-type: none"> • Updated Watchdog Timer to include the watchdog <code>reset_time</code> requirement to ensure the validity of the application configuration. Listed the supported devices for the watchdog timer feature. • Updated Device Support section. • Added Active Serial Remote Configuration Mode to clarify that the active serial configuration mode is a subset of the remote configuration mode. Also clarified that this mode is only available for EPCS devices. • Added a link to the Configuration Handbook in the Remote System Configuration Modes. • Updated Remote Configuration Mode to add that Cyclone IV E devices support AP configuration scheme and included a link to the Configuration and Remote System Upgrades in Cyclone IV Devices chapter. • Updated Remote System Configuration Components to clarify that the local configuration mode does not support the user watchdog timer feature. • Included a cross-reference to the Input Port in Page Mode Feature. • Updated Parameters to update values and supported devices of the GUI parameter settings. • Updated Factory Configuration to clarify that the default factory configuration address does not apply for Cyclone V devices.

Date	Version	Changes
July 2013	2013.07.12	<ul style="list-style-type: none"> • Added Cyclone III and Cyclone IV Devices Remote Update Operation. • Updated Input Ports to include Arria V and Cyclone V support for <code>data_in[]</code> port. • Added Param[] as a standalone section. • Updated Parameter Type and Corresponding Parameter Bit Width Mapping to include Arria V and Cyclone V support for Reconfiguration trigger conditions parameter. Also updated Page Select parameter to include information for Arria V, Cyclone V, and Stratix V devices. • Updated Parameter Type and Corresponding Parameter Bit Width Mapping to update the Configuration Mode (AnF) information. • Updated Input Ports to add a link to the Configuration, Design Security, and Remote Upgrades in the Cyclone III Device Family chapter. • Updated Input Ports to clarify that a falling edge of the <code>reset_timer</code> signal triggers a reset of the user watchdog timer. • Updated Output Ports to add Arria II, Arria V, Cyclone V, Stratix IV, and Stratix V device support for 24-bit bus for <code>data_out[]</code> port. • Added Knowledge Base section. • Added Simulation to clarify that simulation capability are for Arria GX, Stratix, and Stratix II devices only.
February 2012	3.0	Add Cyclone IV support for param[] parameter.
August 2010	2.5	<p>Updated for Quartus II software v10.0, including:</p> <ul style="list-style-type: none"> • Updated the Device Family Support section. • Add Parameters table to Specifications chapter. • Added new parameters and ports to Specifications chapter. • Added new prototypes and declarations sections to Specifications chapter. • Updated design example figures and steps.
April 2009	2.4	<p>Updated for Quartus II software v9.0, including:</p> <ul style="list-style-type: none"> • Updated the section. • Added the Maximum Clock Frequency (f_{MAX}) for the supported devices. • Updated ports and parameter tables.

Date	Version	Changes
May 2007	2.3	Updated for Quartus II software v7.1, including: <ul style="list-style-type: none">• Updated to include support for Arria® GX devices.• Updated to include Cyclone® III device information.• Added Referenced Documents section.
March 2007	2.2	Updated Chapter 1 to include Cyclone III support.
December 2006	2.1	Updated Chapter 1 to include Stratix® III support.
September 2006	2.0	General update for Quartus II software version 6.0, including screenshots; added ModelSim®-Altera simulation tool section to Chapter 3.
March 2005	1.0	Initial release.