Configuration via Protocol (CvP) Implementation in Altera FPGAs

User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

UG-01101-1.1

Document last updated for Altera Complete Design Suite version: 11.1
Document publication date: January 2012

Subscribe

Contents

Chapter 1. Overview

Chapter 2. FPGA Configuration in CvP

Chapter 3. FPGA Fabric Image Update in CvP

Chapter 4. CvP Topologies

Chapter 5. Design Considerations

Chapter 6. Software Support

Additional Information

1. Overview

This user guide describes the features and implementations of the Configuration via Protocol (CvP) scheme in supported Altera® FPGA families. This user guide only shows you how to use the CvP scheme in the supported FPGA, and does not explain the PCI Express® (PCIe®) protocol.

This user guide uses the Stratix® V device in the illustrations, guidelines, and examples. Information specific to other device families will be stated accordingly.

Getting Started

Use this user guide in conjunction with an understanding of the following PCIe specifications:

- PCI Express Base 1.1 Specification
- PCI Express Base 2.1 Specification
- PCI Express Base 3.0 Specification
- PCI Express CEM 2.0 Specification

Embedded PCIe Hard IP Blocks in Altera FPGAs

Altera FPGAs contain embedded PCIe hard IP blocks that you can configure as end points or a root complex for PCIe applications. However, you cannot use the embedded PCIe hard IP block as a root complex when it is configured for CvP.

To establish the PCIe link between a root complex and an end point successfully, the end point must be functional within the time period set by the PCIe power-up and wake-up timing specification.

Autonomous PCIe Hard IP Block

After the PCIe hard IP block is configured, it is in autonomous mode. The autonomous PCIe feature allows the hard IP block in the FPGA to start link training before the fabric is configured. This feature helps the FPGA to accommodate the PCIe timing requirement in open and closed systems.

In an open system, all the electronic systems attached to the open system must obey the PCIe wake-up time requirement as defined in the PCIe specifications. In a closed system, the electronic systems attached to the closed system do not need to obey the PCIe wake-up time requirement as defined in the PCIe specifications.

If the FPGA fabric is not fully functional when the PCIe host starts accessing the PCIe hard IP block, the autonomous PCIe hard IP block responds to the PCIe host with a configuration retry status (CRS) transaction for configuration read and configuration write requests, or an unsupported request (UR) transaction for memory and I/O requests. The PCIe host will attempt to access the FPGA end point for one second after receiving the initial CRS or UR transaction before it recognizes that the FPGA is faulty. This delay allows the FPGA fabric to finish configuration. All PCIe hard IP blocks are part of the periphery image. Regardless of the CvP mode, the periphery image is always configured before the fabric configuration.

CvP Scheme

The CvP is a new FPGA configuration scheme supported in Arria® V, Cyclone® V, and Stratix V devices. It uses the autonomous PCIe hard IP block feature of the embedded PCIe hard IP block to allow the FPGA to meet the PCIe power-up timing specification. The benefit of the autonomous PCIe hard IP block is that the embedded PCIe core is operational before the FPGA fabric is configured. Conventionally, the embedded PCIe hard IP block can only be operational after the full FPGA image is fully configured.

The CvP scheme offers a solution for configuring the FPGA fabric through the PCIe link. The following are the benefits of using the CvP scheme:

- Reduced cost—the FPGA fabric configuration data is stored in the memory of the PCIe host and the external flash is used only to store the periphery configuration data. This can reduce the flash memory size required because the periphery configuration data file size is smaller when compared with the full FPGA image configuration data, which saves system cost.
- Design protection—CvP ensures the PCIe host can exclusively access the FPGA fabric image (the most important portion of the FPGA configuration data). This provides better protection for the FPGA against unauthorized design tampering or copying.
- Image update without system down time—CvP allows the FPGA fabric to be updated through the PCIe link without a host reboot or FPGA full chip reinitialization.

2. FPGA Configuration in CvP

This chapter describes the interfaces involved in the CvP setup, CvP configuration image, and types of CvP modes.

CvP Setup Interfaces

CvP setup involves two interfaces: the conventional configuration interface and the PCIe link.

Conventional Configuration Scheme

The conventional configuration interface is to connect the FPGA to a JTAG configuration interface, the configuration device for the Active Serial (AS) configuration scheme, or the configuration host for both the Fast Passive Parallel (FPP) and Passive Serial (PS) configuration schemes. This interface is used to configure the periphery image or the full image to the FPGA.

For more information about the conventional configuration scheme in Altera FPGAs, refer to the *Configuration, Design Security, and Remote System Upgrades* chapters of the respective Altera FPGA handbooks.

PCIe Link

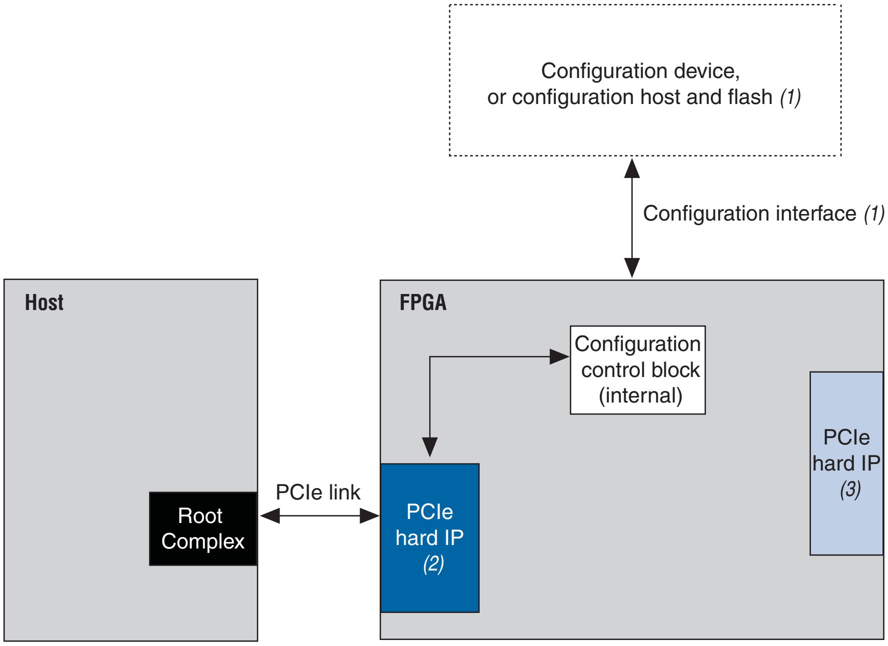
The PCIe link connects the PCIe hard IP block of the FPGA to the PCIe host. This interface is used to configure the FPGA with the fabric image. Only one designated PCIe hard IP block has the interface to the internal configuration control block, which you can use for CvP. You can also use this designated PCIe hard IP block to perform other PCIe applications after the FPGA enters user mode. The other PCIe hard IP blocks do not have the dedicated interface to the internal configuration control block; therefore you can use these other PCIe hard IP blocks for PCIe applications only and not to perform CvP.

For Arria V, Cyclone V, and Stratix V devices, the PCIe hard IP block located at the bottom left from the top view (which is GXB_L0 bank) of the FPGA has the connection to the internal configuration control block, which you can use for CvP.

For more information about the PCIe hard IP block location in Altera FPGAs, refer to the *Transceiver Architecture* chapters of the respective Altera FPGA handbooks.

Figure 2–1 shows the high-level hardware setup for FPGA configuration in the CvP scheme.

Figure 2–1. CvP Hardware Setup


Configuration device,
or configuration host and flash *(1)*
Configuration interface *(1)*
Host
FPGA
Configuration control block (internal)
PCIe hard IP *(3)*
PCIe link
Root Complex
PCIe hard IP *(2)*

Notes to Figure 2–1:

(1) The configuration interface can be any of the supported configuration schemes such as JTAG, FPP, PS, or AS. The choice of configuration device depends on the chosen configuration scheme.

(2) This PCIe hard IP block is for CvP and other PCIe applications.

(3) This PCIe hard IP blocks only support PCIe applications. You cannot use it for CvP.

CvP Configuration Image

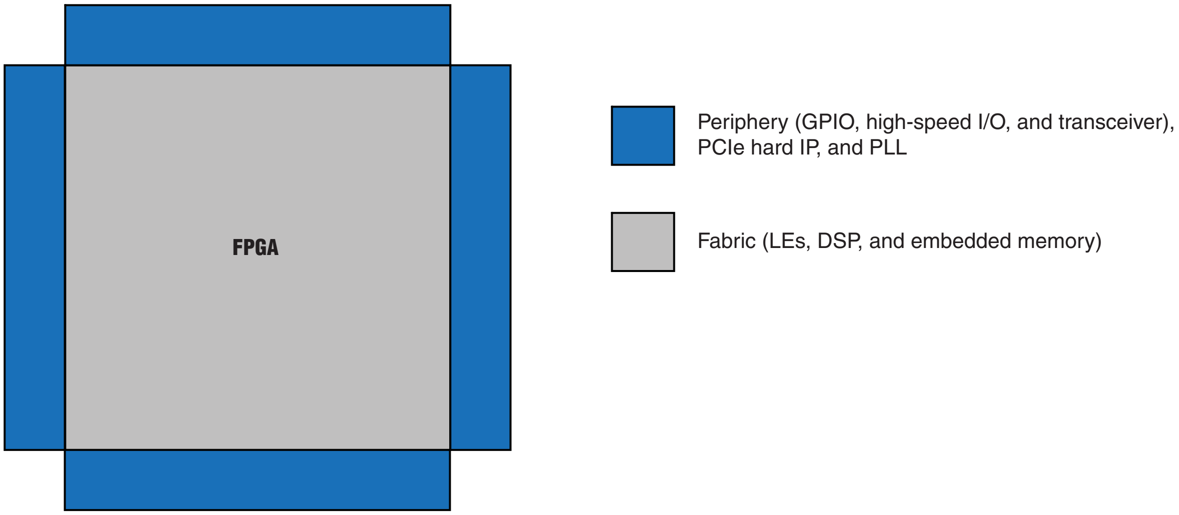
A full FPGA configuration image has two portions: the periphery image and the fabric image. The periphery image contains the configuration data for PCIe hard IP block and I/O buffer settings, which includes all user I/Os and dedicated transceiver channels. Fabric image contains the configuration data of the logic elements (LEs), digital signal processing (DSP) blocks, and embedded memory of the design.

In the conventional configuration scheme supported in previous FPGA families, the periphery image and fabric image are combined in a single configuration file.

You can generate the periphery image and fabric image in two separate configuration files for CvP scheme in a future version of the Quartus II software.

Figure 2–2 shows the periphery and fabric location of the FPGA and does not represent the transceiver and I/O bank location of the FPGA.

Figure 2–2. FPGA Periphery and Fabric


FPGA
Periphery (GPIO, high-speed I/O, and transceiver), PCIe hard IP, and PLL
Fabric (LEs, DSP, and embedded memory)

For more information about the location of the transceiver banks and I/O banks in the FPGA, refer to the *High-Speed Differential I/O Interfaces and DPA* chapters of the respective Altera FPGA handbooks.

CvP Modes

FPGA configuration in the CvP scheme supports three different modes. Depending on the CvP mode, the conventional configuration interface and the PCIe link are used to transfer different FPGA configuration data. The CvP modes are described in the following sections.

Table 2–1 lists the features of each CvP mode.

Table 2–1. CvP Mode

Feature	CvP Initialization and Update Mode (1)	CvP Update Mode (1)
PCIe Link Data Rate (2), (3)	Gen1, Gen2 (4)	Gen1, Gen2, Gen3
PCIe Link Usage	Initial FPGA fabric configuration, FPGA fabric image update, and PCIe application in user mode	FPGA fabric image update and PCIe application in user mode
FPGA Configuration Method	Periphery configuration through conventional schemes (AS, PS, FPP, and JTAG). Fabric configuration through the PCIe link	Full configuration through conventional configuration schemes (AS, PS, FPP, and JTAG)
FPGA Fabric Image Update through the PCIe Link	Supported	Supported

Notes to Table 2–1:

(1) The CvP mode is set in the Quartus II software. For more information, refer to "CvP Setting in Device and Pin Options" on page 7–1.

(2) The FPGA CvP and the PCIe application in a user design must use the same PCIe configuration settings.

(3) PCIe Gen1 and Gen2 support x1, x4, and x8 modes.

(4) PCIe Gen3 requires physical medium attachment (PMA) calibration for the PCIe link training before the link transitions to active (L0) state. The PMA calibration block is part of the fabric image, hence it is not supported in CvP Initialization and Update mode.

CvP Initialization and Update mode

There are two steps to configure the FPGA using CvP Initialization and Update mode: periphery configuration and fabric configuration. Therefore, the configuration image in CvP Initialization and Update mode is split into two different files. The first configuration file contains the periphery image (stored in the external configuration device) and the second configuration file contains the fabric image (stored in a memory that is accessible by the PCIe host).

During device power up, the periphery image is loaded from the external configuration device to the FPGA through the conventional configuration interface. The configuration bits for the PCIe hard IP blocks are part of the periphery image.

After the periphery configuration is completed, the CONF_DONE signal goes high and allows the FPGA to start the PCIe link training. When the PCIe link training completes, the PCIe link transitions to L0 state and the PCIe host begins to initiate the fabric configuration through the PCIe link. After the fabric is successfully configured, the CvP_CONFDONE pin goes high, signifying the completion of the full FPGA configuration. In CvP Initialization and Update mode, you must observe both CONF_DONE and CvP_CONFDONE to verify that the FPGA is fully configured. After the FPGA is fully configured, the FPGA enters initialization and user mode. If INIT_DONE signal is enabled, the INIT_DONE signal goes high after the initialization completes and FPGA enters user mode.

After the device enters user mode, you can use the PCIe link for normal PCIe transactions. In CvP Initialization and Update mode, you can also use the PCIe link to perform an FPGA fabric image update. To achieve this, generate multiple FPGA fabric images in the Quartus® II software that keep the existing same periphery functionality. For more information about fabric image updates in CvP Initialization and Update mode, refer to the "FPGA Fabric Image Update in CvP" on page 3–1.

CvP Update mode

In CvP Update mode, the FPGA is fully configured through a conventional configuration scheme on device power up. A single configuration file that contains the full FPGA configuration image is stored in the external configuration device and is loaded to the FPGA. The `CONF_DONE` signal goes high after the completion of the full FPGA configuration. After the FPGA is fully configured, the FPGA enters initialization and user mode. If `INIT_DONE` signal is enabled, the `INIT_DONE` signal goes high after the initialization completes and FPGA enters user mode. After the device enters user mode, the PCIe links are available for the normal PCIe application.

Besides the normal PCIe applications, you can use the PCIe link for an FPGA fabric image updates. To achieve this, you can generate multiple FPGA fabric images in the Quartus II software that keep the same periphery functionality.

For the image update in CvP Update mode, refer to the "FPGA Fabric Image Update in CvP" on page 3–1.

3. FPGA Fabric Image Update in CvP

This chapter describes the process of the FPGA fabric image update through CvP.

Fabric Image Update

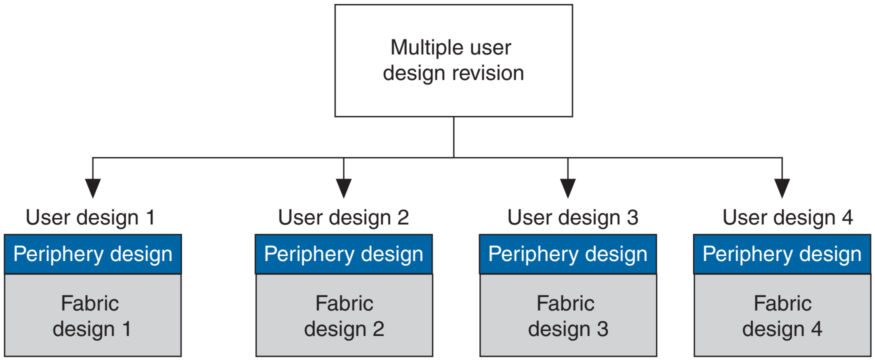
After the FPGA enters user mode, the PCIe host triggers an FPGA fabric image update through the PCIe link. FPGA fabric image update is supported in CvP Initialization and Update mode, and CvP Update mode. You cannot update the periphery image through CvP.

Altera recommends that you fix the periphery functionality for each fabric image design in CvP. After each fabric image update, the core fabric designs keep the same functionality to the periphery interface. If the periphery image and the first core image are generated with encryption, compression or both features turned on, you must ensure the encryption, compression or both features are turned on for the other fabric images that you will use for fabric image update via CvP.

You can update the PLLs and transceivers using dynamic reconfiguration. For more information about dynamic reconfiguration, refer to the *Dynamic Reconfiguration* chapters of the respective Altera FPGA handbooks.

Figure 3–1 shows an example of how you can create multiple FPGA fabric images for use with CvP Update.

Figure 3–1. Example of Multiple User Design Planning for CvP Fabric Image Update



In Figure 3–1, the periphery design is fixed from user design revision 1 to revision n. However, the fabric design changes in every user design revision. These fabric designs must have the same functionality to the periphery interface. You must ensure the system keeps the same periphery functionality for each fabric image that is created.

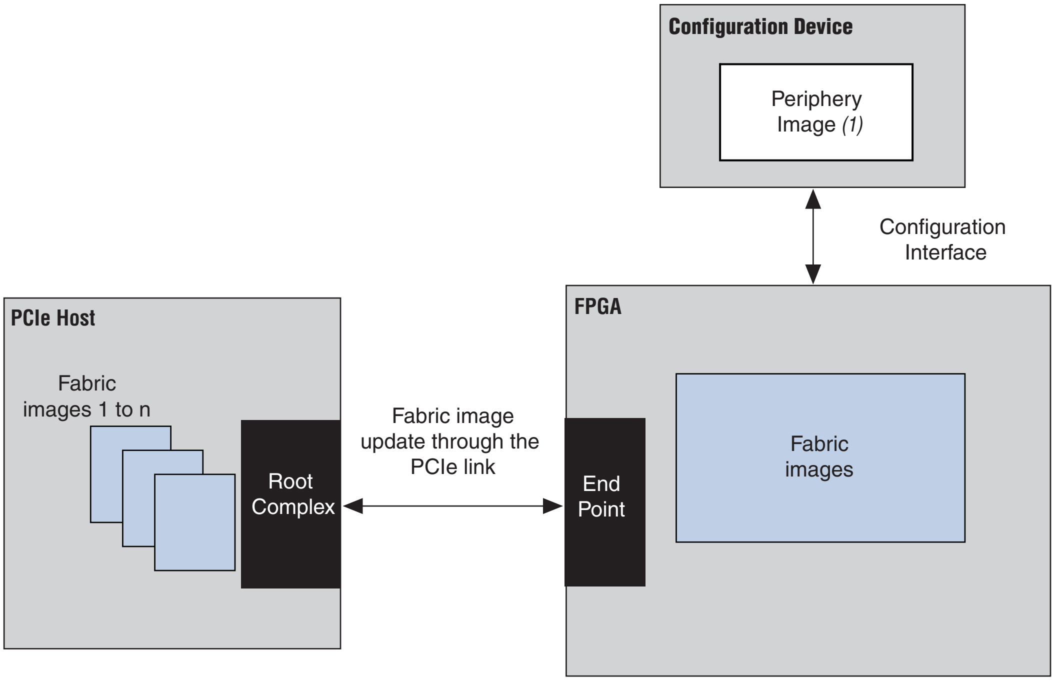
In the Quartus II software, you can generate multiple fabric images within a single design project. However, you must generate only one periphery image for that design project because the periphery image must remain the same for all the fabric images to work with CvP Update. The periphery image is stored in the configuration device and you may not modify or update the periphery image during the fabric image update. The multiple fabric images are stored in memory that is accessible by the PCIe host, as shown in Figure 3–2.

If a fabric image update attempt is triggered, the `CvP_CONFDONE` pin is pulled low, which indicates that a fabric image update has started. The FPGA fabric is reinitialized and reconfigured with the new fabric image. During the fabric image update through a PCIe link, the `nCONFIG` and `nSTATUS` pins of the FPGA are logic high. When the fabric image update completes, the `CvP_CONFDONE` pin is released high again and indicates that the FPGA has entered user mode.

☞ Support for multiple fabric images creation will be available in a future version of the Quartus II software.

Figure 3–2 shows the arrangement of the periphery image and multiple fabric images stored for fabric image update.

Figure 3–2. Periphery and Fabric Images Storage Arrangement in CvP Fabric Image Update *(1)*



Note to Figure 3–2:

(1) The periphery image remains the same for different fabric image updates. If you change the periphery image, you must reprogram the configuration device with the new periphery image.

4. CvP Topologies

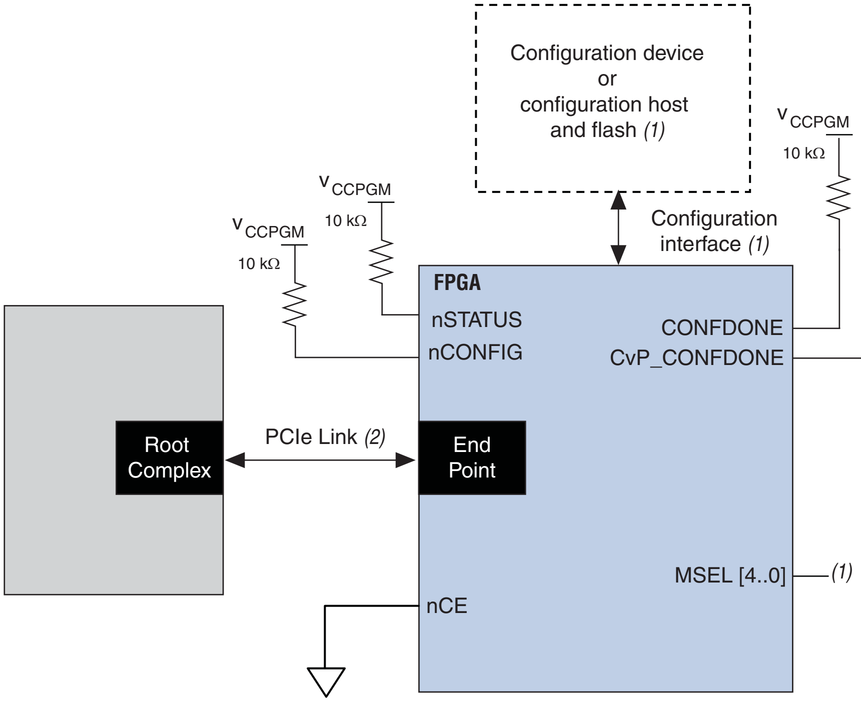
This chapter describes some of the topologies that can be used with CvP. The CvP is supported for single FPGA or multiple FPGAs configurations.

Single End Point

Use the single end point topology for a single FPGA configuration. In this topology, the PCIe link involves only one PCIe end point in the FPGA and the PCIe root complex in the host. Depending on the CvP mode you select, the conventional configuration interface is used to configure the periphery image or the full FPGA image after system power up. The PCIe link can be used for fabric image configurations or updates, in addition to the usual PCIe link application.

Figure 4–1 shows the single end point topology.

Figure 4–1. Single End Point Topology in CvP



Notes to Figure 4–1:

(1) The configuration interface can be any of the supported configuration schemes such as AS, PS, FPP, or JTAG. The `MSEL[4..0]` are used to indicate the chosen configuration scheme.

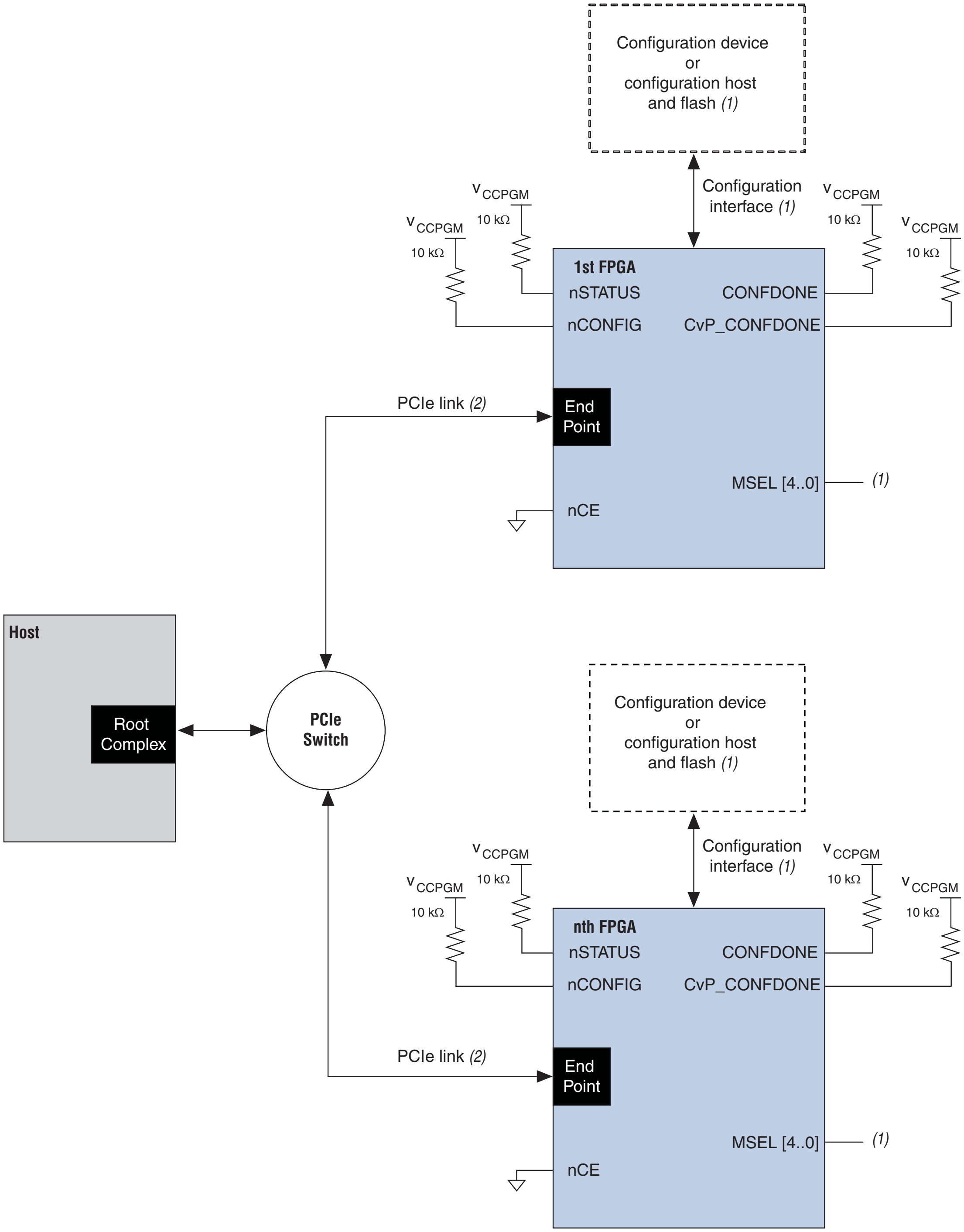
(2) The PCIe link includes the `PERST#` signal and other PCIe signals as defined in the PCIe base specification.

Multiple End Points

Use the multiple end points topology to configure multiple FPGAs. In this topology, one root complex connects to multiple FPGA end points through a PCIe switch. The PCIe switch controls the fabric image configuration through the PCIe link to the targeted FPGA end point. This topology provides flexibility for you to select the FPGA that you must configure or update through the PCIe link. There is no limitation on the number of FPGAs that can be configured using this topology. The root complex must have the ability to respond to the PCIe switch and direct the configuration data transaction to the designated end point based on the end point media access control (MAC) address reported by the PCIe switch.

Figure 4–2 shows the multiple end points topology.

Figure 4–2. Multiple End Points Topology in CvP



Notes to Figure 4–2:

(1) The configuration interface can be any of the supported configuration schemes such as AS, PS, FPP, and JTAG. The `MSEL[4..0]` signals are used to indicate the configuration scheme.

(2) The PCIe link includes the `PERST#` signal and other PCIe signals as defined in the PCIe base specification.

Mixed Chain

Use mixed chain topology to support multiple FPGA configurations. In this topology, the root complex of the host only responds to the end point in the master FPGA (the first FPGA in the chain) through the PCIe link. You can configure the slave FPGAs in the chain through a PS or FPP configuration scheme. Only the master FPGA is configured in CvP mode.

You must design user IP in the master FPGA to fetch the data from the root complex to the slave devices in the chain. The data is latched out from the master FPGA through GPIOs and latched into the slave FPGAs through the PS or FPP configuration pins such as DCLK, DATA line, or DATA bus.

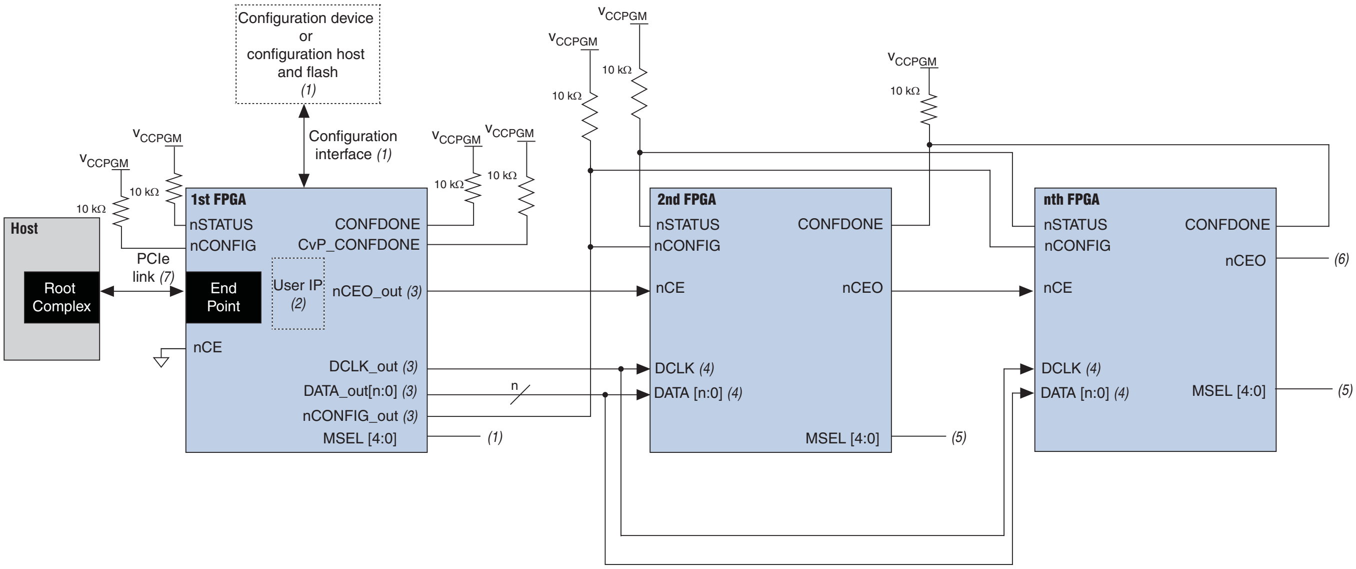
The DCLK, DATA bus, nCONFIG, nSTATUS, and CONF_DONE pins of the slave devices are tied together, allowing the slave devices to enter user mode at the same time. If there is a configuration error during the slave device configuration, the slave device chain reinitializes and reconfigures by having the nSTATUS line pulled low and released by the FPGAs again. You must ensure there is a suitable line buffering on the DCLK and DATA bus if there are more than four slave devices in the chain.

You can cascade the slave devices in two ways:

- If the slave devices are configured with different configuration files, the nCEO pin of one slave device is connected to the nCE pin of the next slave device in the chain. If the first slave device completes the configuration, the device pulls the nCEO pin that enables the next slave device. The next slave device starts its configuration and pulls the nCEO pin low to enable the subsequent device. This process continues until the last slave device in the chain is configured. You can leave the nCEO pin of the last device unconnected or use as a user I/O, as shown in Figure 4–3 with the slave devices configured in FPP mode.
- If the slave devices are configured with the same configuration file for identical user applications, the nCE pins of the slave devices are connected to the nCEO pins of the master device. The nCEO pins of the slave devices are left unconnected or used as user I/Os. In this topology, all the slave devices are configured at the same time, as shown in Figure 4–4 with the slave devices configured in FPP mode.

Figure 4–3 shows the mixed chain topology.

Figure 4–3. Mixed Chain Topology in CvP (Different Configuration Files for Slave Device Configurations)



Notes to Figure 4–3:

(1) The configuration interface can be any of the supported configuration schemes such as AS, PS, FPP, or JTAG. The `MSEL[4..0]` signals are used to indicate the configuration scheme.

(2) User IP is a user-designed IP programmed in the first FPGA to fetch data from the root complex through the PCIe link, which is sent to the slave FPGAs in the chain through a PS or FPP configuration interface.

(3) `DCLK_out` and `DATA_out` are the GPIO pins used to fetch the PS or FPP configuration data. The `nCONFIG_out` and `nCEO_out` are the GPIO pins used to connect to the `nCONFIG` and `nCE` pins of the slave device.

(4) Configuration data input to the slave FPGAs through PS or FPP configuration pins.

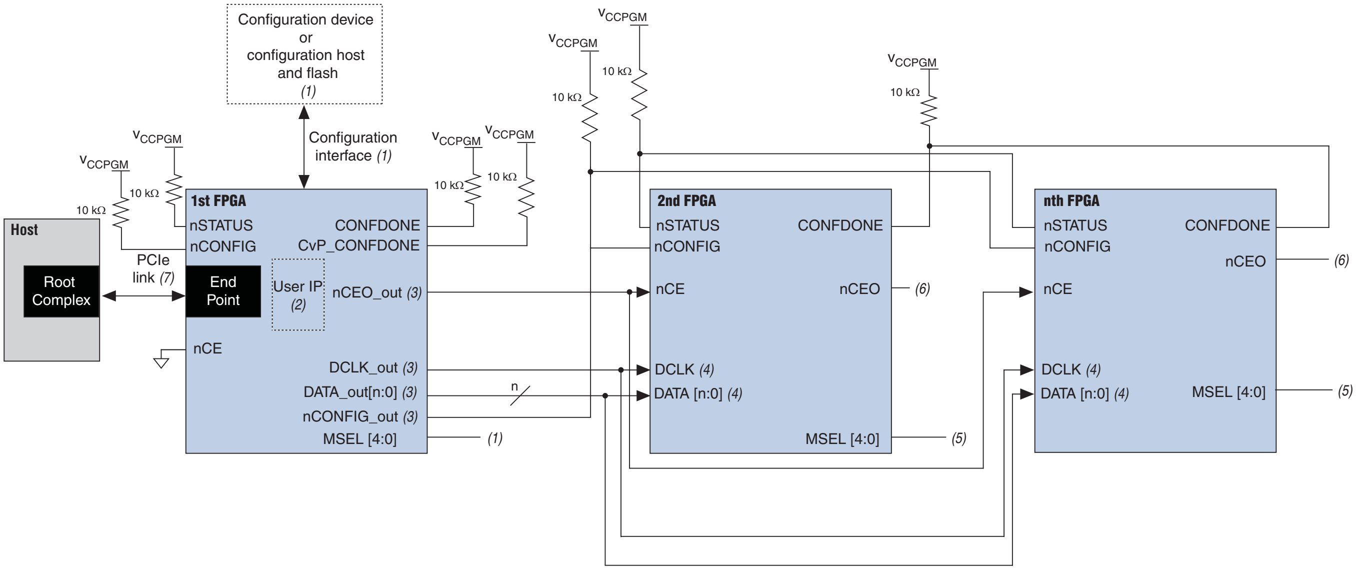
(5) Connect the `MSEL` pins to the PS or FPP scheme.

(6) You can leave the `nCEO` pin of the last FPGA in the chain unconnected or use it as a GPIO.

(7) The PCIe link includes the `PERST#` signal and other PCIe signals as defined in the PCIe base specification.

Figure 4–4 shows the mixed-chain topology in CvP with all the slave devices configured at the same time.

Figure 4–4. Mixed Chain Topology in CvP (Single Configuration File for Slave Device Configurations)



Notes to Figure 4–4:

(1) The configuration interface can be any of the supported configuration schemes such as AS, PS, FPP, or JTAG. The `MSEL[4..0]` signal and choice of configuration device depend on the chosen configuration scheme.

(2) User IP is a user-designed IP programmed in the first FPGA to fetch data from the root complex through the PCIe link, which is sent to the slave FPGAs in the chain through a PS or FPP configuration interface.

(3) `DCLK_out` and `DATA_out` are the GPIO pins used to fetch the PS or FPP configuration data. The `nCONFIG_out` and `nCEO_out` are the GPIO pins used to connect to the `nCONFIG` and `nCE` pins of the slave device.

(4) Configuration data input to the slave FPGAs through PS or FPP configuration pins.

(5) Connect the `MSEL` pins to the PS or FPP scheme.

(6) You can leave the `nCEO` pin of the last FPGA in the chain or use it as a GPIO.

(7) The PCIe link includes the `PERST#` signal and other PCIe signals as defined in the PCIe base specification.

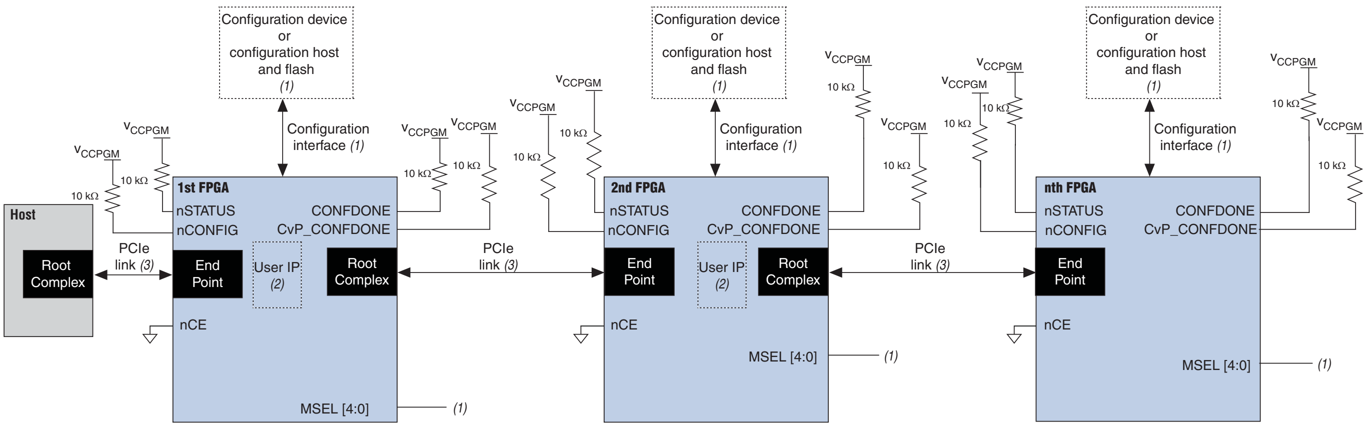
Daisy Chain

A daisy chain is a CvP topology for configuring multiple FPGAs. Configuration devices are attached to each FPGA for periphery configuration (CvP Initialization and Update mode) or full FPGA configuration (CvP Update mode) after the system powers up. The `nCE`, `nSTATUS`, `nCONFIG`, `CONFDONE`, and `CvP_CONFDONE` pins of the FPGAs are not tied together, allowing the FPGAs to receive the configuration data individually from their respective configuration device.

Each FPGA in the daisy chain has two PCIe IP cores configured as an end point and a root complex, respectively. In this topology, the root complex in the host responds to the end point of the first FPGA in the chain. The fabric image of the first FPGA is configured through the PCIe link between the host and its end point. After the configuration is completed, the first FPGA enters user mode and the user IP in the first FPGA initiates the fabric configuration of the subsequent device. The process continues until the last device in the chain is configured. You must design a user IP as part of the FPGA fabric image. The user IP routes data from the host to the FPGA root complexes. The root complex of the first FPGA is connected to the end point of the subsequent FPGA through the PCIe link. The fabric image of the second FPGA is fetched from the host and is routed through the first FPGA before it is loaded into the second FPGA through the PCIe link. This process continues until the last FPGA in the chain is configured.

Figure 4–5 shows the daisy chain topology.

Figure 4–5. Daisy Chain Topology in CvP



Notes to Figure 4–5:

(1) The configuration interface can be any of the supported configuration schemes such as AS, PS, FPP, or JTAG. The `MSEL[4..0]` signal is used to indicate the chosen configuration scheme.

(2) User IP is a user-design IP programmed in the first FPGA to data from the root complex through the PCIe link and send to the slave FPGAs in the chain through FPP configuration interface.

(3) The PCIe link includes the `PERST#` signal and other PCIe signals as defined in the PCIe base specification.

5. Design Considerations

This section discusses the hardware design considerations if your FPGA design is configured in CvP mode.

Pin Descriptions

Depending on the CvP mode chosen, configuration involves the conventional configuration interface and the PCIe link. This section covers only the CvP pins that are not included in the conventional configuration interface and the typical PCIe link.

For more information about the configuration pins involved in the conventional configuration interface, refer to the *Configuration, Design Security, and Remote System Upgrades* chapters of the respective Altera FPGA handbooks.

For more information about the typical PCIe link signals, refer to the PCI Express Base Specification and the *Transceivers* chapters of the respective Altera FPGA handbooks.

Table 5–1 lists the additional CvP pin descriptions and connections that are not listed in the *Configuration, Design Security, and Remote System Upgrades* chapters of the respective Altera FPGA handbooks and the PCI Express Base Specification.

Table 5–1. CvP Pin Descriptions and Connections

Pin Name	Pin Type	Pin Description	Pin Connection
CvP_CONFDONE	Output	Driven low during fabric configuration and released or driven high after the completion of the fabric configuration. During FPGA configuration in CvP Initialization and Update mode, you must observe this pin after the CONF_DONE pin goes high to determine if the FPGA is fully configured. During FPGA fabric image update, you must observe this pin to determine if the fabric image update is completed. If CvP mode is turned off, you can use this pin as a user I/O pin.	If this pin is set as dedicated output, the V_{CCPGM} power supply must meet the input voltage specification of the receiving side. If this pin is set as an open drain output, the pin must be connected to an external 10 kΩ-pull-up resistor to the V_{CCPGM} power supply or a different pull-up voltage that meets the input voltage specification of the receiving side. This gives an advantage on the voltage leveling.
nPERSTL0	Input	This pin is connected to the PCIe hard IP block as a dedicated fundamental reset pin for PCIe usage. If the signal is low, the transceivers and dedicated PCIe (CvP) hard IP block are in reset mode. If this signal is high, the transceivers and dedicated PCIe (CvP) hard IP block are released from the reset mode. If the PCIe hard IP block is not in use, you can use this pin as a user I/O pin.	Connect the nPERSTL0 to the PERST# pin of the PCIe slot. During CvP, this pin is powered by the V_{CCPGM} power supply. In the case that the V_{CCPGM} power supply is not at 3.3 V, you can still drive an input signal to this pin at 3.3-V volts without requiring a voltage level shifter with the following requirements to conform to the PCIe electrical standards: ■ The input signal must meet the V_{IH}/V_{IL} specification of 3.3-V **LVTTL** I/O standard. ■ The input signal must not exceed the overshoot specification for 100% operation as stated in the *Device Data Sheet* chapters of the respective Altera FPGA handbooks.

CvP Configuration Features

CvP supports configuration features such as data decompression, decryption, and partial reconfiguration.

Table 5–2 lists the configuration features supported in each CvP mode.

Table 5–2. CvP with Configuration Features

CvP Mode	Decompression (1)	Decryption (1)	Remote System Upgrade	Partial Reconfiguration (1)
CvP Initialization and Update	Supported	Supported	Not supported (2)	Supported
CvP Update Mode	Supported	Supported	Not supported (2)	Supported

Notes to Table 5–2:

(1) Feature support is not available in the Quartus II software version 11.1. This feature will be available in a future release of the Quartus II software.

(2) The system upgrade feature is supported through fabric image update through the PCIe link.

Data Compression

The data compression reduces the size of the fabric image. The periphery image is not compressed. Enabling data compression in CvP Update mode can help reduce both the external configuration memory required for full FPGA image storage and the configuration time of the full FPGA configuration.

If the data compression is enabled in CvP Initialization and Update mode, the external configuration memory required for periphery image storage remains the same but the memory size required for the fabric image in the PCIe host is reduced.

Data Encryption

Use the data encryption feature to encrypt the fabric image with the Advanced Encryption Standard (AES) algorithm to protect the data against unauthorized access. The periphery image is not encrypted by this feature. To configure the FPGA with encrypted data, a security key is pre-programmed to the FPGA and is used to decrypt the incoming bitstream.

FPGA Image Update

You can achieve the FPGA image update if you are not using any CvP modes with the normal remote system upgrade feature. FPGA fabric image update in CvP Initialization and Update mode, and CvP Update mode is performed through the PCIe link. Partial reconfiguration is supported in all the CvP modes. However, software support for this feature will only be available in a future release of the Quartus II software.

Designing CvP for an Open System

In an open system, all the electronic systems attached to the open system must obey the PCIe wake-up time requirement as defined in the PCIe specifications. This section describes the requirements for FPGA power supply ramp time, PCIe wake-up time requirement in an open system, and configuration scheme selection for CvP modes.

FPGA Power Supplies Ramp Time Requirement

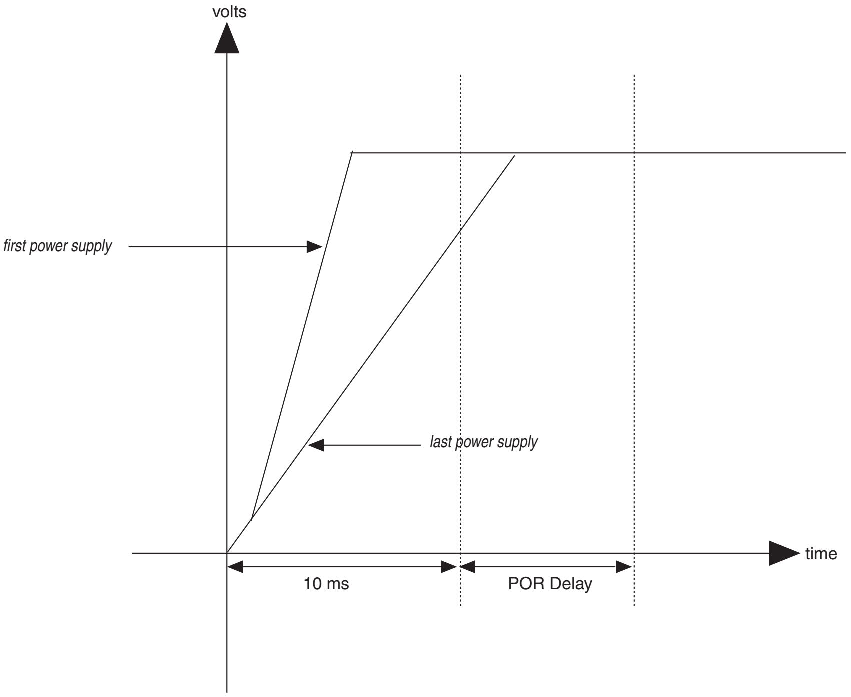
This section describes the FPGA power supplies ramp-up time requirement to accommodate a CvP in an open system.

The POR circuit generates a POR signal that keeps the device in a reset state until the power supply outputs are within the operating range. If power is supplied, a POR event occurs if the POR monitored power supplies reach the recommended operating range within the maximum power supply ramp time (t_{RAMP}).

To use CvP, the total time must be within 10 ms, from the first power supply ramp-up to the last power supply ramp-up (refer to Figure 5–1). You must set the `PORSEL` pin high for fast POR, in which the POR delay time is in the range of 4–12 ms, leaving enough time after the POR trip for the PCIe to start initialization and configuration.

Figure 5–1 shows the power supplies ramp-up time and POR.

Figure 5–1. Power Supplies Ramp-Up Time and POR



For more information about POR, refer to the *Hot Socketing and Power-On Reset* chapters of the respective Altera FPGA handbooks.

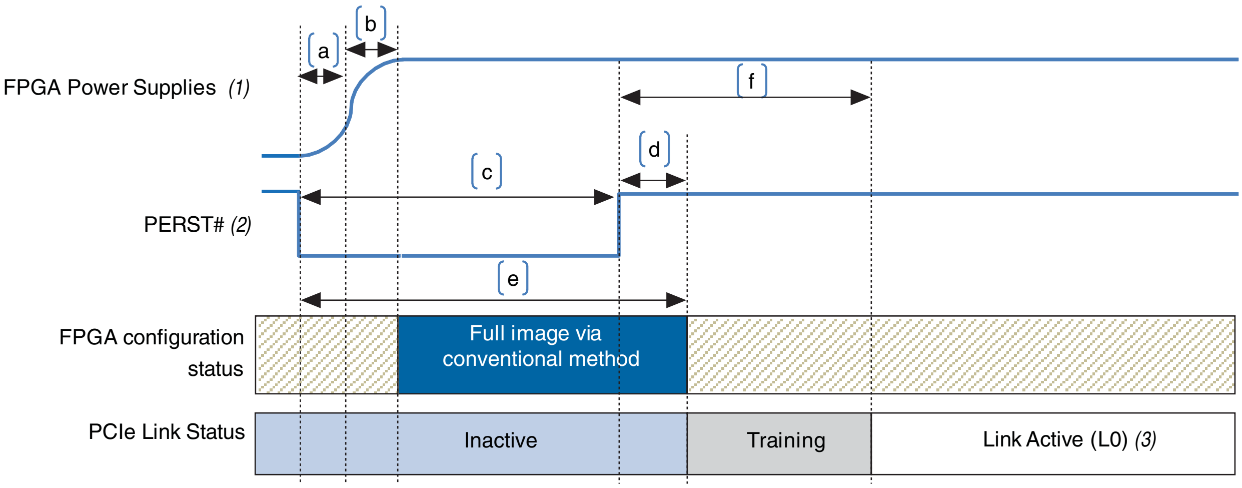
PCIe Wake-Up Time Requirement in an Open System

To accommodate the open system of PCIe applications, the PCIe link connected to the FPGA must meet the PCIe wake-up timing specification that must transition from power-on to the link active (`L0`) state within 200 ms. The overall timing from FPGA power up to the PCIe hard IP block in the FPGA is ready for link training must be within 120 ms.

For CvP Update mode, the full FPGA image is configured through the conventional configuration schemes after device power-up. To accommodate your system into an open system, the first full FPGA configuration must be within 120 ms. Therefore, you must choose the right conventional configuration scheme for your device to ensure the configuration time is able to meet the 120-ms requirement. For more information about the supported configuration scheme that you can choose to design an open system with the CvP Update mode, refer to Table 5–3.

Figure 5–2 shows the PCIe power-up sequence and the FPGA configuration timing in CvP Update mode.

Figure 5–2. PCIe Timing Sequence in CvP Update Mode


a
b
f
d
c
e
FPGA Power Supplies *(1)*
PERST# *(2)*
FPGA configuration status
Full image via conventional method
PCIe Link Status
Inactive
Training
Link Active (L0) *(3)*

Notes to Figure 5–2:

(1) To ensure successful configuration, all the POR-monitored power supplies to the FPGA must ramp up monotonically to operating range within the 10 ms ramp-up time.

(2) One of the auxiliary signals specified in the PCIe electromechanical specification. The `PERST#` signal is sent from host to the FPGA and it is used to indicate if the power supplies of the FPGA are within their specified voltage tolerance and are stable. It also initializes the FPGA state machines and other logic after power supplies are stabilized.

(3) When the PCIe link turns active, the PCIe function starts up. The PCIe link supports PCIe application in user mode for CvP Update mode, you can also use the PCIe link for fabric image update.

Table 5–3 lists the power-up sequence timing in CvP Update mode, as shown in Figure 5–2.

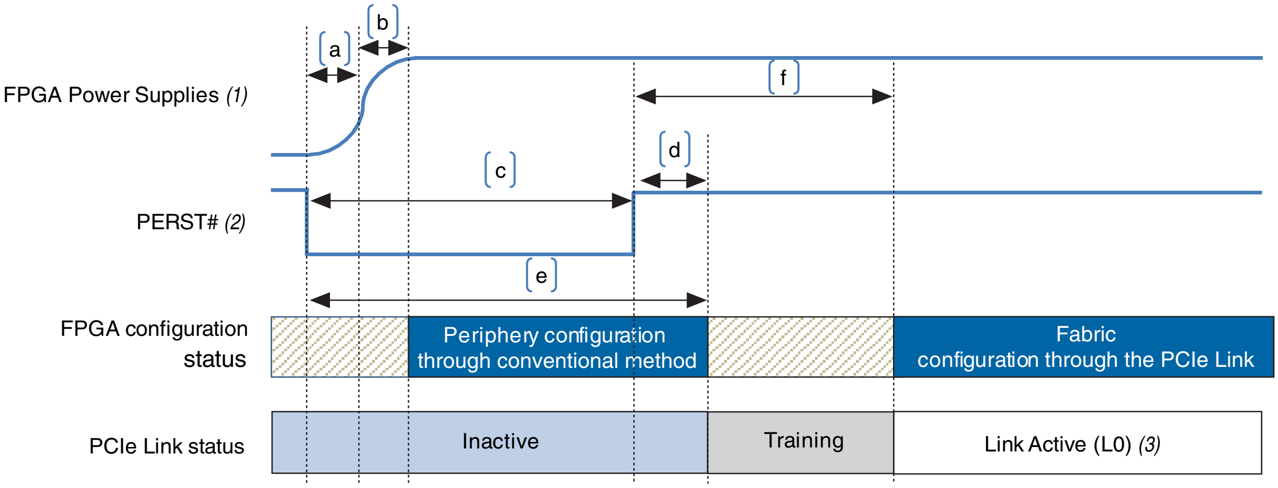
Table 5–3. Power-Up Sequence Timing in CvP Update Mode

Timing Sequence in Figure 5–2	Timing Range (ms)	Description
a	10	Maximum ramp-up time requirement for all POR-monitored power supplies in the FPGA to reach their respective operating range
b	4–12	FPGA POR delay time
c	100	Minimum `PERST#` signal active time
d	20	Minimum `PERST#` signal inactive time before the PCIe link enters training state
e	120	Maximum time from the FPGA power up to the end of the full FPGA configuration in CvP Update mode
f	100	Minimum `PERST#` signal inactive time before the PCIe link enter active state

For CvP Initialization and Update mode, the PCIe hard IP block is guaranteed to meet the 120 ms time limit because the periphery image configuration time is significantly smaller when compared with the full FPGA configuration time. When designing CvP for an open system in CvP Initialization and Update mode, you can choose any of the conventional configuration schemes for the periphery image configuration. However, the ramp-up time for the POR-monitored power supplies of the FPGA must not exceed 10 ms.

Figure 5–3 shows the PCIe power up and the FPGA configuration timing in CvP Initialization and Update mode.

Figure 5–3. PCIe Timing Sequence in CvP Initialization and Update Mode


a
b
f
FPGA Power Supplies (1)
c
d
PERST# (2)
e
FPGA configuration status
Periphery configuration through conventional method
Fabric configuration through the PCIe Link
PCIe Link status
Inactive
Training
Link Active (L0) (3)

Notes to Figure 5–3:

(1) To ensure successful configuration, all POR-monitored power supplies to the FPGA must ramp up monotonically to the operating range within the 10 ms ramp-up time.

(2) One of the auxiliary signals specified in the PCIe electromechanical specification. The `PERST#` signal is sent from the host to the FPGA and it is used to indicate if the power supplies of the FPGA are within their specified voltage tolerance and are stable. It also initializes the FPGA state machines and other logic after power supplies stabilize.

(3) After the PCIe link turns active, the PCIe function starts up. For CvP Initialization and Update mode, the PCIe link supports the FPGA fabric configuration, FPGA fabric image update, and PCIe application in user mode.

Table 5–4 lists the power-up sequence timing in CvP Initialization and Update mode, as shown in Figure 5–3.

Table 5–4. Power-Up Sequence Timing in CvP Initialization and Update mode

Timing Sequence in Figure 5–3	Timing Range (ms)	Description
a	10	Maximum ramp-up time requirement for all POR-monitored power supplies in the FPGA to reach their respective operating range
b	4-12	FPGA POR delay time
c	100	Minimum `PERST#` signal active time
d	20	Minimum `PERST#` signal inactive time before the PCIe link enters training state
e	120	Maximum time from the FPGA power up to the end of periphery configuration in CvP Initialization and Update mode
f	100	Minimum `PERST#` signal inactive time before the PCIe link enter active state

Supported Configuration Schemes for CvP Modes

Figure 5–2 on page 5–5 shows the first FPGA configuration time after device power-ups for CvP Update mode that is constrained to 120 ms to meet the 200-ms PCIe wake-up timing specification for an open system. You must ensure the chosen configuration scheme for full FPGA configuration in CvP Update mode is able to meet the configuration time requirement. If your are not using CvP mode, you must perform your own evaluation on the configuration time to meet the PCIe wake-up timing specification.

Table 5–5 lists supported configuration schemes for different CvP modes to meet the 200-ms PCIe wake-up timing specification.

Table 5–5. Supported Configuration Schemes for CvP Modes [(1), (2)] **(Part 1 of 2)**

Family	Device	CvP Initialization and Update	CvP Update
Arria V GX	5AGXA1 5AGXA3 5AGXA5 5AGXA7	(3)	FPP x8 FPP x16
	5AGXB1 5AGXB3 5AGXB5 5AGXB7	(3)	FPP x16
Arria V GT	5AGTD3 5AGTD7	(3)	FPP x16
Arria V SX	5ASXB3 5ASXB5	(3)	TBD
Arria V ST	5ASTD3 5ASTD5	(3)	TBD
Cyclone V E [(4)]	5CEA2 5CEA4 5CEA5 5CEA7 5CEA9	—	—
Cyclone V GX	5CGXC3	(3)	AS x4 FPP x8 FPP x16
	5CGXC4 5CGXC5 5CGXC7	(3)	FPP x8 FPP x16
	5CGXC9	(3)	FPP x16
Cyclone V GT	5CGTD5 5CGTD7 5CGTD9	(3)	TBD
Cyclone V SE [(4)]	5CSEA2 5CSEA4 5CSEA5 5CSEA6	—	—

Table 5–5. Supported Configuration Schemes for CvP Modes [1], [2] (Part 2 of 2)

Family	Device	CvP Initialization and Update	CvP Update
Cyclone V SX	5CSXC4 5CSXC5 5CSXC6	[3]	TBD
Cyclone V ST	5CSTD5 5CSTD6	[3]	TBD
Stratix V GX	5SGXA3 5SGXA4	[3]	FPP x16 FPP x32
	5SGXA5 5SGXA7	[3]	FPP x32
	5SGXA9 5SGXAB	[3]	—
	5SGXB5 5SGXB6	[3]	FPP x32
Stratix V GT	5SGTC5 5SGTC7	[3]	FPP x32
Stratix V GS	5SGSD2 5SGSD3	[3]	FPP x8 FPP x16 FPP x32
	5SGSD4 5SGSD5	[3]	FPP x16 FPP x32
	5SGSD6 5SGSD8	[3]	FPP x32
Stratix V E [4]	5SEE9 5SEEB	—	—

Notes to Table 5–5:

[1] Data compression and encryption feature are disabled. These features require different data to clock ratio, which causes the total configuration time to prolong and does not meet the 200-ms PCIe wake-up timing specification.

[2] The configuration schemes suggested in this table are based on the fastest DCLK frequency.

[3] For periphery configuration in CvP Initialization and Update mode, all the conventional configuration schemes such as AS, PS, FPP, or JTAG are supported.

[4] No PCIe hard IP, CvP is not supported in this family.

To design CvP in a closed system, you must perform the configuration time estimation to ensure the total FPGA configuration time—either for full FPGA configuration in CvP Update mode, or the periphery configuration in CvP Initialization and Update mode—is within the permitted time given by the PCIe host.

Use the data in Table 5–6 to estimate the file size before design compilation. Different configuration file formats, such as a hexadecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. For the different types of configuration file and file sizes, refer to the Quartus II software. However, for a specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you are using compression, the file size can vary after each compilation because the compression ratio depends on your design.

Table 5–6 lists the uncompressed raw binary file (**.rbf**) sizes for Altera FPGAs.

Table 5–6. Uncompressed .rbf Sizes for Altera FPGAs [(1)] (Part 1 of 2)

Family	Device	Configuration .rbf Size (bits)	IOCSR .rbf Size (bits)
Arria V GX	5AGXA1	69,034,936	1,380,699
	5AGXA3	69,034,936	1,380,699
	5AGXA5	101,511,640	2,030,233
	5AGXA7	101,511,640	2,030,233
	5AGXB1	138,416,696	2,768,334
	5AGXB3	138,416,696	2,768,334
	5AGXB5	185,327,416	3,706,548
	5AGXB7	185,327,416	3,706,548
Arria V GT	5AGTD3	138,416,696	2,768,334
	5AGTD7	138,416,696	2,768,334
Cyclone V E [(2)]	5CEA2	5,748,552	114,971
	5CEA4	9,534,304	190,686
	5CEA5	14,889,560	297,791
	5CEA7	19,965,752	399,315
	5CEA9	39,425,016	788,500
Cyclone V GX	5CGXC3	20,651,605	413,032
	5CGXC4	51,307,993	1,026,160
	5CGXC5	51,307,993	1,026,160
	5CGXC7	99,342,419	1,986,848
	5CGXC9	198,526,878	3,970,538
Cyclone V GT	5CGTD5	14,889,560	297,791
	5CGTD7	19,965,752	399,315
	5CGTD9	39,425,016	788,500
Stratix V GX	5SGXA3	139,255,840	1,689,600
	5SGXA4	139,255,840	1,689,600
	5SGXA5	266,599,584	2,239,436
	5SGXA7	266,599,584	2,239,436
	5SGXA9	387,394,048	2,141,440
	5SGXAB	387,394,048	2,141,440
	5SGXB5	266,798,896	2,236,416
	5SGXB6	266,798,896	2,236,416
Stratix V GT	5SGTC5	266,035,472	2,195,200
	5SGTC7	266,035,472	2,195,200

Table 5–6. Uncompressed .rbf Sizes for Altera FPGAs *(1)* **(Part 2 of 2)**

Family	Device	Configuration .rbf Size (bits)	IOCSR .rbf Size (bits)
Stratix V GS	5SGSD2	93,080,448	933,120
	5SGSD3	93,080,448	933,120
	5SGSD4	209,935,224	2,004,480
	5SGSD5	209,935,224	2,004,480
	5SGSD6	266,798,896	2,236,416
	5SGSD8	266,798,896	2,236,416
Stratix V E *(2)*	5SEE9	387,394,048	2,141,440
	5SEEB	387,394,048	2,141,440

Notes to Table 5–6:

(1) These values are preliminary.

(2) No PCIe hard IP, CvP is not supported in this family.

6. Software Support

This chapter discusses the software support for CvP mode in the Quartus II software version 11.1.

Table 6–1 lists the features for CvP support that are not enabled in the Quartus II software version 11.1, but will be available in a future release of the Quartus II software.

Table 6–1. Upcoming CvP Support

Upcoming CvP Feature	Description
Programming file generation for periphery and fabric image	Generate programming file for periphery image and fabric image.
Multiple fabric images compilation and generation	Compile multiple fabric image designs in a single project and generate multiple fabric images for fabric image update in CvP mode.
CvP with configuration features (compression, encryption, and partial reconfiguration)	Enable compression, encryption, and partial reconfiguration feature in CvP mode.

Device and Pin Options Setting

By default, CvP mode is turned off in the Quartus II software. Before compiling your design, you must set the CvP mode and the `CvP_CONFDONE` pin behavior accordingly.

Table 6–2 lists the settings you can specify in the **Device and Pin Options** dialog box to enable CvP mode in your design.

Table 6–2. CvP Settings in Device and Pin Options

Setting	Option	Description
Configuration via Protocol	**Power up and subsequent core configuration**	Enable CvP Initialization and Update mode. The FPGA periphery is configured through the conventional configuration scheme and the FPGA fabric is configured through the PCIe link. Use the PCIe link for fabric image update or PCIe applications in user mode.
	Subsequent core reconfiguration	Enable CvP Update mode. The FPGA is fully configured through the conventional configuration scheme. Use the PCIe link for the fabric image update or PCIe applications in user mode.
`CvP_CONFDONE` pin	**Enable `CvP_CONFDONE` pin**	Enable `CvP_CONFDONE`. If you turn on this option, the `CvP_CONFDONE` pin is used to indicate whether or not the fabric configuration is successfully completed. If you turn off this option, the `CvP_CONFDONE` pin functions as a normal user I/O.
	Enable open drain on `CvP_CONFDONE` pin	Enable `CvP_CONFDONE` as an open drain output pin. If you turn on this option, the `CvP_CONFDONE` pin functions as an open drain output. If you turn off this option, the `CvP_CONFDONE` is a dedicated output pin.

Hard IP for PCIe version 11.1

IP Compiler for PCIe is the parameter entry tool for the PCIe hard IP block. To use CvP in your design, you must include the hard IP compiler as part of your design. You must enable the **Select the CvP capable HIP block** check box and set the parameter settings according to the PCIe link of your system.

For complete information about the IP Compiler for PCIe, refer to the *IP Compiler for PCI Express User Guide*.

For complete information about the Arria V Hard IP Compiler for PCIe, refer to the *Arria V Hard IP for PCI Express User Guide*.

For complete information about the Cyclone V Hard IP Compiler for PCIe, refer to the *Cyclone V Hard IP for PCI Express User Guide*.

For complete information about the Stratix V Hard IP Compiler for PCIe, refer to the *Stratix V Hard IP for PCI Express User Guide*.

For example, to perform CvP applications on a Stratix V device, follow these steps:

1. Open the MegaWizard™ Plug-In Manager.
2. In the MegaWizard Plug-In Manager, select **Stratix V Hard IP for PCI Express v11.1**.
3. In the MegaWizard Parameter settings, specify the PCIe settings according to your system.
4. Turn on the **Select the CvP capable HIP block** option.
5. Click **Finish**.

Generating Periphery Image and Fabric Image

You can use the **Convert Programming Files** dialog box in the Quartus II Programmer in the Quartus II software to convert programming files for CvP. To generate the periphery image and fabric image for CvP, you must enable the **Create CvP files** check box in the **Convert Programming Files** dialog box. The Quartus II Programmer will generate the following files:

- A **.pof** for the full FPGA image
- A **.pof** for the periphery image
- A **.rbf** for the fabric image

The file extension name for periphery image is <*file_name*>.**periph.pof** and file extension name for fabric image is <*file_name*>**.core.rbf**

For more information about the settings in the **Convert Programming Files** dialog box, refer to the *Quartus II programmer* chapter.

VSEC

These Vendor-Specific Extended Capability (VSEC) registers for Configuration Space enhancement have been added to the PCIe hard IP for CvP mode. The PCIe host will write to and poll status bits in these registers to communicate with the FPGA control block. This capability structure occupies byte addresses 200h–240h in the Configuration Space, and its contents are summarized in Table 6–3.

Table 6–3. VSEC Register Address

Register Name				Byte Offset
31..24	23..16	15..8	7..0	
Next Cap Offset	Version	PCI Express Extended Capability ID		00h
VSEC Length	VSEC Rev	VSEC ID		04h
Altera Marker				08h
Reserved				0Ch
Reserved				10h
Reserved				14h
Reserved				18h
CvP Status		Reserved		1Ch
CvP Mode Control				20h
Reserved				24h
CvP Data				28h
CvP Programming Control				2Ch
Reserved (for possible CvP expansion)				30h
Uncorrectable Internal Error Status Register				34h
Uncorrectable Internal Error Mask Register				38h
Correctable Error Status Register				3Ch
Correctable Error Mask Register				40h

Altera Defined VSEC Header

Table 6–4 lists the description of the Altera defined VSEC header in the Configuration Space.

Table 6–4. Altera Defined VSEC Capability Header (Offset 00h)

Bits	Reset Value	PCIe Access Mode	Register Description
[15..0]	000Bh	RO	PCI Express Extended Capability ID—PCIe specification defined value for VSEC Capability ID
[19..16]	1h	RO	Version—PCIe specification defined value for VSEC version
[31..20]	Variable	RO	Next Capability Offset —value is the starting address of the next Capability Structure implemented, if any (per k_xxx bits)

Altera Defined Vendor-Specific Header

Table 6–5 lists the description of the Altera defined vendor-specific header in the Configuration Space.

Table 6–5. Altera defined Vendor-Specific Header (Offset 04h)

Bits	Reset Value	PCIe Access Mode	Register Description
[15..0]	1172h	RO	VSEC ID—this will be a user configurable VSEC ID, the default value will 1172h (the Altera Vendor ID) but can be changed in case of conflicts with the other user VSEC ID
[19..16]	0	RO	VSEC—user configurable VSEC revision
[31..20]	12’h44	RO	VSEC length—total length of this structure in bytes

Altera Marker

This read only register is an additional marker so that if the user uses the Quartus II software to configure the device using CvP, the Quartus II software can perform an additional check to make sure it is operating with the correct VSEC.

Table 6–6 lists the description of the Altera marker in the Configuration Space.

Table 6–6. Altera Marker (Offset 08h)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..0]	11721172h	RO	Altera Marker—an additional marker for the Quartus II software to be able to verify that this is the right structure to try to use CvP. The upper most byte now indicates a device revision number. The upper 4 bits indicates device type (`Stratix V` = 1), the lower 4 bits indicates revision (`RevA` = 1).

CvP Status

The CvP status register provides visibility to software of several control block status signals used in the programming algorithm.

Table 6–7 lists the description of the CvP status in the Configuration Space.

Table 6–7. CvP Status (Offset 1Eh)

Bits	Reset Value	PCIe Access Mode	Register Description
[15..10]	00h	RO	Reserved
[9]	Signal Value	RO	Reserved
[8]	Signal Value	RO	`PLD_CLK_IN_USE`—from clock switch module to FPGA fabric. Status bit is provided for debug.
[7]	Signal Value	RO	`CVP_CONFIG_DONE`—this signal indicates that the control block has completed the device configuration using CvP and there were no errors. This signal is mutually exclusive with `CVP_CONFIG_ERROR`. IE for a given configuration attempt either `CVP_CONFDONE` or `CVP_CONFIG_ERROR` will come on, but not both.
[6]	0	RO	Reserved
[5]	Signal Value	RO	`USERMODE`—indicates if the configurable FPGA fabric is in user mode.
[4]	Signal Value	RO	`CVP_EN`—reflects value of the control block signal, indicates if the control block is enabling CvP mode.
[3]	Signal Value	RO	`CVP_CONFIG_ERROR`—reflects the value of this signal from the control block, checked by software to determine if there was an error during configuration.
[2]	Signal Value	RO	`CVP_CONFIG_READY`—reflects the value of this signal from the control block, checked by the Quartus II software during programming algorithm.
[1]	0	RO	Indicates that CvP data is treated as compressed.
[0]	0	RO	Indicates that CvP data is treated as encrypted.

CvP Mode Control

The CvP mode control register provides some global control of the CvP operation. The bits in this register are separated from the bits toggled during programming to provide some form of protection from an accidental flipping of these global control bits that could render the hard IP non-operational.

Table 6–8 lists the description of the CvP mode control byte in the Configuration Space.

Table 6–8. CvP Mode Control (Offset 20h)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..16]	00000000h	RO	Reserved
[15..8]	00000000h	RW	CVP_NUMCLKS—specifies the number of CvP clock cycles to issue for every CvP Data Register Write. Valid values are 0x00 – 0x3F (where 0x00 corresponds to 64 clock cycles, and 0x01-0x3F corresponds to 1 to 63 clock cycles). The upper bits are not used, but are included in this field because they belong to the same byte enable.
[7..4]	0000h	RO	Reserved
[3]	0	RO	Reserved
[2]	0	RW	CVP_FULLCONFIG—request the control block to reconfigure the entire FPGA including the hard IP. Will cause the hard IP to crash the PCIe link. May be fatal to the system.
[1]	0	RW	HIP_CLK_SEL—selects between PMA and FPGA fabric clock when USER_MODE=1 and PLD_CORE_READY=1. 1 – Select internal clock from PMA, required in CVP_MODE 0 – Select clock from soft logic fabric. This setting should only be used when the fabric is configured in user mode with a configuration file that connects the correct clock, otherwise the hard IP will be inoperable. The value of this signal can only be changed when there has been no other Transaction Layer Packets (TLPs) to or from the hard IP for 10 µs. There should be no TLPs issued to the hard IP for 10 µs after this value is changed. When entering CvP mode, this bit should be set before CVP_MODE is set. When exiting CvP mode, it should be cleared after CVP_MODE is cleared. This ensures that there is no clock switching during CvP. CVP_EN == 0b forces this signal to always be 0b (this is TBD).
[0]	0	RW	CVP_MODE controls whether the hard IP is in CvP mode or normal mode. 1 – The hard IP is in CVP_MODE, the signals to the control block are active and all TLPs are routed to Configuration Space. 0 – The hard IP is in normal mode and TLPs are route to the PLD fabric normally. CVP_EN == 0b forces this signal to always be 0b.

CvP Data Register

The CvP data register is the register that the Quartus II software will write the configuration data to. Every write to this register will set the data output to the control block and generate N clock pulses (per `CVP_NUM_CLKS` in CvP Mode Register) to the control block.   Software must ensure that all bytes in the MemWr DWord are enabled. When in CvP mode, this register can also be written to by a Memory Write to any address defined by a Memory Space BAR for this device. Using Memory Writes should allow for higher throughput compared to configuration writes.

Table 6–9 lists the description of the CvP Data register in the Configuration Space.

Table 6–9. CvP Data Register (Offset 28h)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..0]	0	RW	Configuration data to be transferred to the control block to configure the device.

CvP Programming Control Register

The CvP programming control register is written to by the Quartus II software to control the programming algorithm.

Table 6–10 lists the description of the CvP programming control register in the Configuration Space.

Table 6–10. CvP Programming Control Register (Offset 2Ch)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..2]	0	RO	Reserved
[1]	0	RW	`START_XFER`—sets the CvP output to the control block to indicate the start of a transfer.
[0]	0	RW	`CVP_CONFIG`—request the control block to begin a transfer using CvP.

Uncorrectable Internal Error Status Register

The uncorrectable internal error status register is used to report the status of the internally checked errors that are uncorrectable. If these specific errors are enabled by the uncorrectable internal error mask register, the errors are forwarded as an uncorrectable internal error as defined in the PCI Express Base 2.1 Specification (or the applicable ECN to 2.0).

☞ The configuration error detected in CvP mode status bit will be present in both the uncorrectable internal error status register and the correctable internal error status register. This bit will always have the same value in both registers, writing a ‘1’ to this bit position will reset the bit in both registers. By setting either the corresponding uncorrectable internal error mask register bit or the corresponding correctable internal error mask register bit, the Quartus II software can control whether this error is reported as an uncorrectable or correctable error.

Table 6–11 lists the description of the uncorrectable internal error status register in the Configuration Space.

Table 6–11. Uncorrectable Internal Error Status Register (Offset 34h)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..12]	0	RO	Reserved
[11]	0	RW1CS	Detected RXBuf overflow condition in Posted/CPL segment. There is no room in P/CPL space for received TLP (which is not malformed)
[10]	0	RW1CS	Parity error detected on R2CSEB interface
[9]	0	RW1CS	Parity error detected on C2T interface
[8]	0	RW1CS	Parity error detected on T2C interface
[7]	0	RW1CS	Parity error detected at TxTL (TLP will be nullified)
[6]	0	RW1CS	Application detected uncorrectable internal error
[5]	0	RW1CS	`CvP_CONFIG_ERROR_LATCHED`—configuration error detected in CvP mode status (to be reported as uncorrectable). Set whenever `CVP_CONFIG_ERROR` rises while in `CvP_MODE`
[4]	0	RW1CS	Parity error detected at Tx DL LCRC generation status
[3]	0	RW1CS	Parity error detected at R2C interface in internal Configuration Space
[2]	0	RW1CS	Parity error detected at input to RX Buffer status
[1]	0	RW1CS	Retry buffer uncorrectable ECC error status
[0]	0	RW1CS	RX buffer uncorrectable ECC error status

Uncorrectable Internal Error Mask Register

The uncorrectable internal error mask register is used to control which errors are forwarded as internal uncorrectable errors. With the exception of the configuration error detected in CvP mode, all of the errors are severe and may cause the device or PCIe link to be in an inconsistent state. The configuration error detected in CvP mode may be correctable depending on the Quartus II software.

Table 6–12 lists the description of the uncorrectable internal error mask register in the Configuration Space.

Table 6–12. Uncorrectable Internal Error Mask Register (Offset 38h)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..12]	0	RO	Reserved
[11]	1	RWS	Mask for RXBuffer P/CPL segment overflow error
[10]	1	RWS	Mask for parity error detected on R2CSEB interface
[9]	1	RWS	Mask for parity error detected on C2T interface
[8]	1	RWS	Mask for parity error detected on T2C interface
[7]	1	RWS	Mask for parity error detected at TxTL (TLP will be nullified)
[6]	1	RWS	Mask for uncorrectable internal error reported by the application
[5]	0	RWS	Configuration error detected in CvP mode uncorrectable mask
[4]	1	RWS	Data parity error detected at Tx DL LCRC generation mask
[3]	1	RWS	Mask for Data parity error detected at R2C interface in internal Configuration Space
[2]	1	RWS	Data parity error detected at input to RX buffer mask
[1]	1	RWS	Retry buffer uncorrectable ECC error mask
[0]	1	RWS	RX buffer uncorrectable ECC error mask

Correctable Internal Error Status Register

The correctable internal error status register is used to report the status of the internally checked errors that are considered correctable. If these specific errors are enabled by the correctable internal error mask register, they will be forwarded as a correctable internal error as defined in the PCI Express Base 2.1 Specification (or the applicable ECN to 2.0).

☞ The configuration error detected in CvP mode status bit will be present in both the uncorrectable internal error status register and the correctable internal error status register. It will always have the same value in both registers, writing a '1' to this bit position will reset the bit in both registers. By setting either the corresponding uncorrectable internal error mask register bit or the corresponding correctable internal error mask register bit, the Quartus II software can control whether this error is reported as an uncorrectable or correctable.

Table 6–13 lists the description of the correctable internal error status register in the Configuration Space.

Table 6–13. Correctable Internal Error Status Register (Offset 3Ch)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..6]	0	RO	Reserved
[6]	0	RW1CS	Application detected a corrected Internal Error
[5]	0	RW1CS	`CVP_CONFIG_ERROR_LATCHED`—configuration detected in CvP mode (to be reported as correctable). Set whenever `CVP_CONFIG_ERROR` rises while in `CVP_MODE`.
[4..2]	0	RO	Reserved
[1]	0	RW1CS	Retry Buffer Correctable ECC error status
[0]	0	RW1CS	RX Buffer Correctable ECC error status

Correctable Internal Error Mask Register

The correctable internal error mask register is used to control which errors are forwarded as internal correctable errors.

Table 6–14 lists the description of the correctable internal error mask register in the Configuration Space.

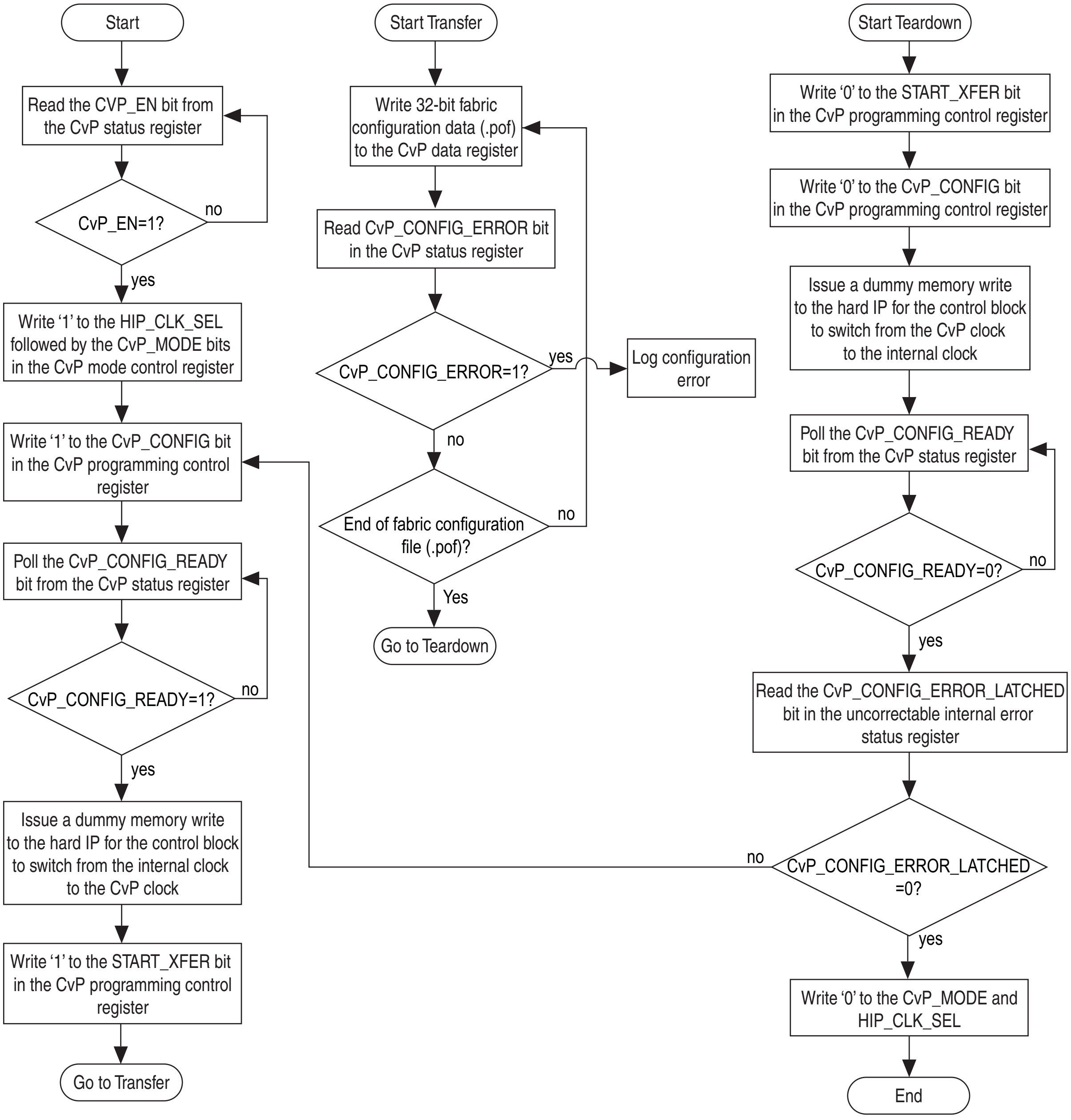
Table 6–14. Correctable Internal Error Mask Register (Offset 40h)

Bits	Reset Value	PCIe Access Mode	Register Description
[31..2]	0	RO	Reserved
[1]	0	RW	`START_XFER`—sets the CvP output to the control block to indicate the start of a transfer.
[0]	0	RW	`CVP_CONFIG`—request the control block to begin a transfer using CvP.

Procedure for Configuring Stratix V FPGAs via PCIe

Figure 6–1 assumes that the FPGA is powered up, and the control block has already configured the periphery. This will be indicated by the CVP_EN read-only bit in the CvP status register. The CvP status register and all other registers mentioned are part of the VSEC register section of the PCIe hard IP Configuration Space.

Figure 6–1. PCIe Driver in the PCIe Host Configuring the FPGA Fabric via PCIe Link


Start
Read the CVP_EN bit from the CvP status register
CvP_EN=1?
no
yes
Write '1' to the HIP_CLK_SEL followed by the CvP_MODE bits in the CvP mode control register
Write '1' to the CvP_CONFIG bit in the CvP programming control register
Poll the CvP_CONFIG_READY bit from the CvP status register
CvP_CONFIG_READY=1?
no
yes
Issue a dummy memory write to the hard IP for the control block to switch from the internal clock to the CvP clock
Write '1' to the START_XFER bit in the CvP programming control register
Go to Transfer
Start Transfer
Write 32-bit fabric configuration data (.pof) to the CvP data register
Read CvP_CONFIG_ERROR bit in the CvP status register
CvP_CONFIG_ERROR=1?
yes
Log configuration error
no
End of fabric configuration file (.pof)?
no
Yes
Go to Teardown
Start Teardown
Write '0' to the START_XFER bit in the CvP programming control register
Write '0' to the CvP_CONFIG bit in the CvP programming control register
Issue a dummy memory write to the hard IP for the control block to switch from the CvP clock to the internal clock
Poll the CvP_CONFIG_READY bit from the CvP status register
CvP_CONFIG_READY=0?
no
yes
Read the CvP_CONFIG_ERROR_LATCHED bit in the uncorrectable internal error status register
CvP_CONFIG_ERROR_LATCHED =0?
no
yes
Write '0' to the CvP_MODE and HIP_CLK_SEL
End

Additional Information

This section provides additional information about the document and Altera.

Document Revision History

The following table lists the revision history for this document.

Date	Version	Changes
January 2012	1.1	■ Added Arria V and Cyclone V devices. ■ Removed references to the CvP Off mode. ■ Updated Chapter 6, Software Support with PCIe driver and CvP programming information. ■ Added "Generating Periphery Image and Fabric Image" and "VSEC" sections.
May 2011	1.0	Initial release.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table lists the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.

Visual Cue	Meaning
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, <*file name*> and <*project name*>**.pof** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
☞	The hand points to information that requires special attention.
?	A question mark directs you to a software help system with related information.
👣	The feet direct you to another document or website with related information.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
✉	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.