

Altera's new device configuration mode—configuration via protocol (CvP)—can be used with PCI Express® to configure the core fabric of Altera's 28-nm Arria® V, Cyclone® V, and Stratix® V FPGAs. CvP can reduce product cost and board size, while simplifying the software usage model, and providing robust in-field system upgrade capability. In addition, the autonomous, embedded PCIe IP core helps ensure that designs meet PCIe power-up time requirements, irrespective of the FPGA core fabric configuration time, guaranteeing a wide range of interoperability with various PCIe-based computer platforms.

Introduction

PCIe technology has replaced PCI as the standard control plane interface between processors and the devices that they monitor. Since its introduction in 2005, FPGA designers have made PCIe one of the most widely used interfaces between FPGAs and processors. Today's FPGAs include embedded PCIe cores that serve as endpoints or root ports.

Until recently the embedded PCIe core could not begin link training and bus enumeration until the FPGA was fully configured. As FPGA configuration times increase with rising device densities, it's becoming difficult to fully configure the FPGAs within the initialization time required by the PCIe specification.

With the announcement of its 28-nm device portfolio, Altera solves this problem by allowing configuration of the PCIe hard IP separate from the FPGA core logic. This technology also allows designers to configure the core fabric of Altera Arria V, Cyclone V, and Stratix V FPGAs via PCIe. The new CvP device programming method can reduce product cost and board size, while simplifying the software usage model, and providing robust in-field system upgrade capability, as described below:

- Lower system cost—CvP can eliminate one or more parallel flash devices and possibly an external programming controller device. In addition, CvP allows designers to store FPGA programming files in a CPU memory system attached to the FPGA via a PCIe link. Using this technique, only the FPGA I/O programming and PCIe core parameters are stored in the flash device, requiring a smaller and cheaper flash device.
- Reduced FPGA resources—Stratix series devices typically require wide, data-path flash devices to store the FPGA programming file. In contrast, EPCS and EPCQ devices supported by CvP require fewer dedicated pins.
- Power savings—Low-power, temporary FPGA images can be loaded via software control based on the user application profile. This feature is useful in battery powered computers.



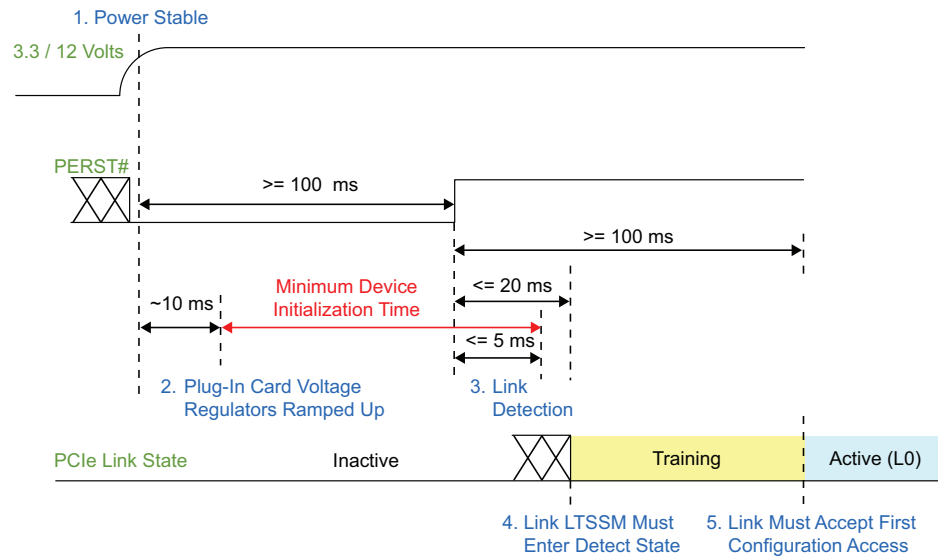
101 Innovation Drive
San Jose, CA 95134
www.altera.com

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX are Reg. U.S. Pat. & Tm. Off. and/or trademarks of Altera Corporation in the U.S. and other countries. All other trademarks and service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera 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 Altera. Altera 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.



Figure 1 provides a simplified representation of the PCIe power-up timing sequence as described in the the *PCI Express Base Specification 1.0a or 1.1 for Gen1* and *PCI Express Base Specification 2.0 for Gen2*. The minimum time allocated to device initialization and device training is 200 ms (equivalent to the difference between point 5 and point 1 in Figure 1). The minimum amount of time allocated to device initialization is depicted by the time difference between point 3 and point 2 in Figure 1, or about 95 ms.

Figure 1. PCIe Power-Up Timing Waveform



Since FPGA devices pack more logic at smaller geometries, more time is required to program large FPGA core fabric with application-specific content. The total configuration time can exceed 95 ms in large devices. When an endpoint device does not reach L0 within the time allocated to it by the PCIe specifications, the endpoint may not respond to the software configuration access transactions (point 5), and the host CPU may fail to recognize this endpoint. In that case, the host CPU may ignore the endpoint and the system operates without it.

Autonomous PCIe Hard IP

In order to circumvent this failure discovery mechanism, Altera's 28-nm FPGAs support operation of embedded PCIe cores, prior to fully configuring the FPGA core fabric. The 28-nm FPGA-embedded PCIe core always meets the PCIe power-up timing requirements by initializing the embedded PCIe cores and the device I/O ring in less than 95 ms. This separately configurable embedded PCIe IP core is referred to as "autonomous."

The following sequence defines the CvP initialization period for PCIe:

1. The embedded PCIe core is held in reset by `PERST#`, and is released shortly before point 3 to start PCIe link discovery and training.
2. The rest of the FPGA core fabric begins programming after the PCIe link completes the training phase and reaches the L0 state.

3. After the embedded PCIe endpoint core reaches the L0 state, the host operating system (OS) starts accessing the PCIe core's configuration space registers (CSR) to perform configuration write access cycles that are part of the system initialization and discovery process (point 5).
4. In case the FPGA core fabric is not fully programmed with the designer's application content (in other words, it has not yet reached "user mode"), the autonomous PCIe core responds with configuration retry status (CRS) transactions until the FPGA core fabric is fully loaded.
5. The OS correctly identifies this endpoint and attempts to poll it again until it becomes fully functional.
6. The endpoint is allowed to respond with CRS for one second before the OS determines that the endpoint is faulty. In other words, the time allocated for the FPGA core fabric programming cannot exceed one second in this device initialization mode.

FPGA Core Fabric Programming Across the PCIe Bus

The following section provides details about the use of CvP with PCIe. Designers can use CvP to load the initial FPGA core fabric image via PCIe, and then later modify that core fabric image during run-time to meet application needs. After the PCIe IP core periphery is programmed, the link trains to the corresponding PCIe operating mode.

After the PCIe link finishes training and reaches the L0 state, the host CPU can program the FPGA core fabric image via PCIe.



Only one of the embedded PCIe cores in each 28-nm FPGA is capable of performing CvP, and only when used as an endpoint.

During the FPGA core fabric configuration via PCIe, all non-serializer/deserializer (SERDES) I/O pins are held high by internal weak pull-up resistors. All other high-speed SERDES pins are essentially held in reset during CvP core fabric image loading. These I/O assignments are designed to freeze I/O operations while the FPGA core fabric configuration is updated. When CvP is enabled, the autonomous PCIe core does not respond with CRS transactions, but rather accepts and responds to PCIe configuration and data transactions in order to perform the FPGA core fabric configuration.

Table 1. 28-nm FPGA Configuration Modes

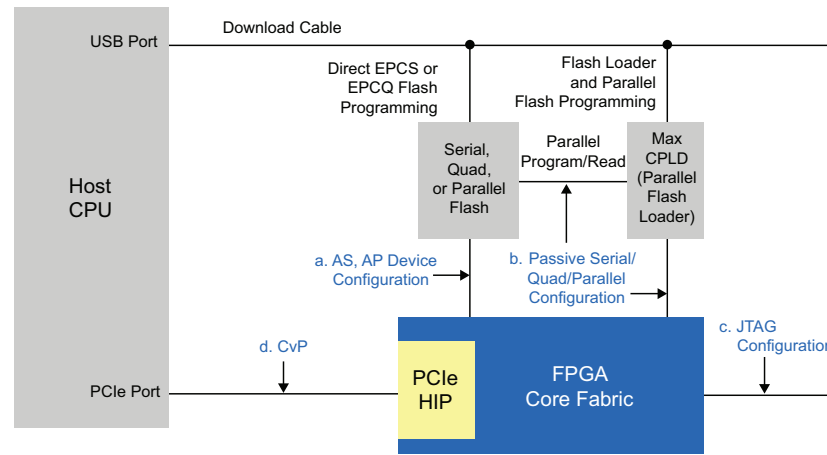
Mode	Status	Data Widths (bits)
Active serial (AS)	Existing	1
Active quad (AQ)	New	4
Passive Serial (PS)	Existing	1
8-bit fast passive parallel (FPP) using flash loader	Existing	8
4-, 16-, or 32-bit FPP using flash loader	New	4, 16, 32 ⁽¹⁾
JTAG-based	Existing	Dedicated JTAG port
CvP using PCIe	New	1, 2, 4, 8 ⁽²⁾

Notes:

- (1) Stratix V devices only.
- (2) Number of lanes in the PCIe link (Gen1 x1, x2, x4, or x8; Gen2 x1, x2, x4, or x8;)

Figure 2 provides a high-level representation of the Stratix V configuration modes and flash programming methods. To simplify the diagram, all flash modes are combined in the block diagram.

Figure 2. Device Configuration and Flash Programming Modes

**Notes:**

- (1) Four possible FPGA configuration modes: (a) Active serial (x1, x4). (b) Passive serial/parallel (x1, x4, x8, x32) using an Altera MAX® CPLD or other logic to read from flash memory and configure the FPGA. (c) JTAG configuration for debug purposes (no need for external flash memory to configure the FPGA). (d) CvP of the FPGA core fabric only. The PCIe hard IP (HIP) and I/O ring are first configured through another method.
- (2) Altera EPCS (serial) or EPCQ (quad) can be directly programmed via the download cable.
- (3) In cases where a MAX CPLD is used to program the flash memory, the MAX CPLD reads from the flash memory and configures the FPGA.

When using CvP with PCIe, the various PCIe IP core parameters, as well as the functionality of the respective high-speed transceivers (SERDES), are initially programmed through one of the existing device initialization modes shown in Table 1. Lower-cost production solutions are possible (but not mandated) through the use of an Altera serial- or quad-flash device that stores only the initialization bits associated with the I/O ring and the PCIe hard IP. Table 2 shows how two bits loaded via one of the device configuration modes (listed in Table 1) determine CvP functionality. The bits are `cvp_enabled` and `full_chip_initialization`.

Table 2. CvP Operating Modes

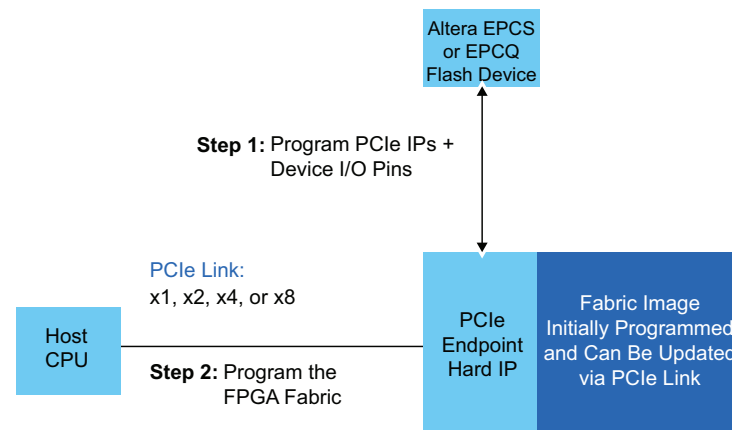
CvP Mode Number	CvP Mode Bits		FPGA Configuration Method
	CvP Enabled	Full Chip initialization	
1	0	1	CvP is off. Full chip initialization through a standard configuration mode. CvP can not be used to update the FPGA core fabric image.

Table 2. CvP Operating Modes

CvP Mode Number	CvP Mode Bits		FPGA Configuration Method
	CvP Enabled	Full Chip initialization	
2	1	0	CvP is on. Only PCIe hard IP, FPGA I/Os, and transceivers are initialized through a standard configuration mode. CvP initially configures the FPGA core fabric. CvP may also be used to update the FPGA core fabric image.
3	1	1	CvP is on. Full chip initialization through a standard configuration mode. CvP can be used to update the FPGA core fabric image.

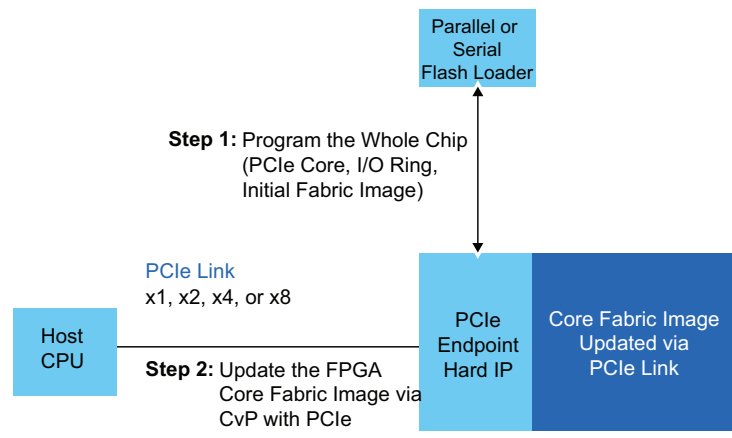
Figure 3 depicts a possible CvP operating mode with a low-cost Altera flash device, and illustrates Mode 2 of Table 2.

Figure 3. Autonomous PCIe Hard IP Core and CvP with an Altera Flash Device



Regardless of the bit settings, the embedded PCIe core is always autonomous. In other words, it wakes up and starts link training before the FPGA core fabric is configured. When `cvp_enabled` is on, CvP is enabled after the embedded PCIe core reaches the L0 state. The host CPU then can configure the FPGA core fabric image when instructed by the software application. Figure 4 depicts Mode 3 in Table 2.

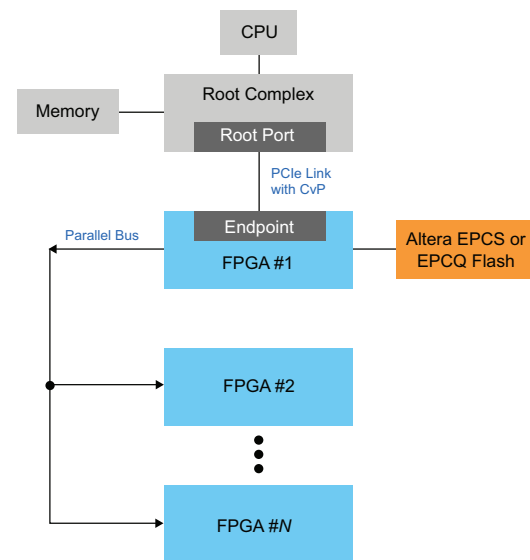
Figure 4. CvP Mode with an Altera Flash Loader



The PCIe endpoint link operating mode used for device programming is the same mode used by the FPGA target application after CvP is complete. For example, designers can program a CvP PCIe core to operate in Gen1 x4 mode. In that case, the core fabric image is loaded through a Gen1 x4 link. The user application logic that subsequently operates in the FPGA after CvP also uses a Gen1 x4 link. The I/O ring configuration (including SERDES) and the embedded PCIe CSRs content remain unchanged during and after the CvP FPGA image updates. Designs modified to replace the initial core fabric image must maintain the same I/O and PCIe core parameters and functionality used in the initial design image.

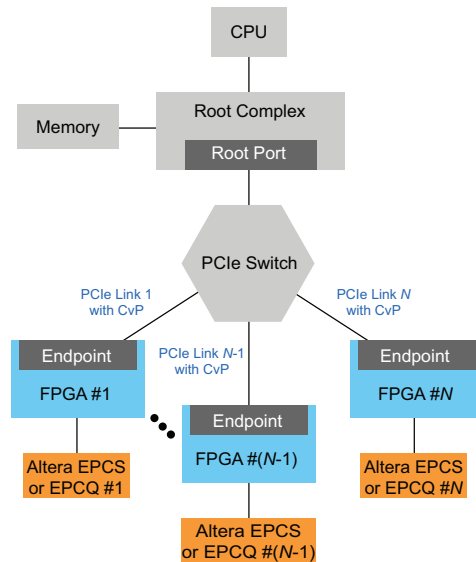
The CvP topology is not limited to the basic operating modes depicted in [Figure 3](#) and [Figure 4](#). [Figure 5](#) describes a mixed FPGA programming mode that is also feasible in Stratix V FPGAs. FPGA #1 is programmed via CvP. The embedded PCIe endpoint core and the I/O ring of FPGA #1 are programmed via an Altera serial configuration device (EPCS) or quad configuration device (EPCQ). The FPGA core fabric is programmed via CvP, similar to [Figure 3](#). Subsequent FPGA devices are programmed through the fast passive parallel mode, described in [Table 1](#). A user-designed IP core in FPGA #1 controls the programming of the other cascaded FPGAs (FPGA #2 – FPGA #N).

Figure 5. Mixed-Mode CvP Programming



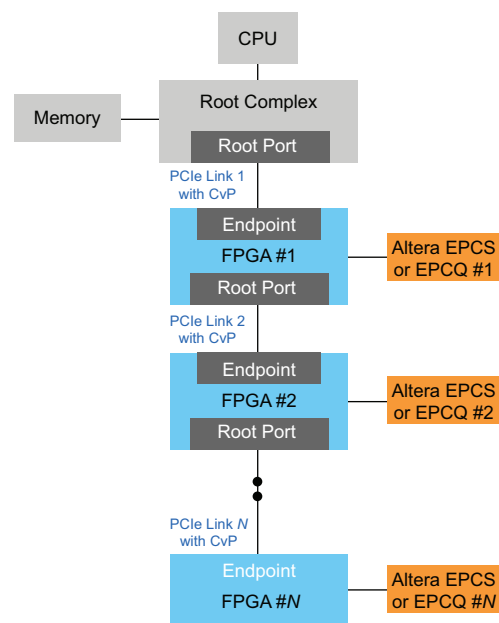
Another possible CvP topology is shown in Figure 6. Multiple FPGAs are programmed via CvP. All FPGAs interface with the root port behind a PCIe switch, thereby using the PCIe topology to program all the FPGA devices that are attached to the switch.

Figure 6. CvP with PCIe Switch



CvP support is also feasible in other PCIe topologies, such as the daisy-chain FPGA topology shown in Figure 7. In this mode, all embedded PCIe cores in the FPGAs are initially programmed by their respective Altera EPCS or EPCQ device. The FPGA core fabrics of the daisy-chained devices are programmed via CvP, and all endpoints and root ports come up in parallel. Each FPGA (except FPGA #N) has a designer-developed IP core in its core fabric that controls the programming of the next FPGA.

Figure 7. Daisy-Chain Format FPGA Programming with CvP





The FPGA designs shown in [Figure 5](#), [Figure 6](#), and [Figure 7](#) may have different user application content, and hence different configuration file images. In all cases, the programming files are initially retrieved from the host CPU memory or its file system.

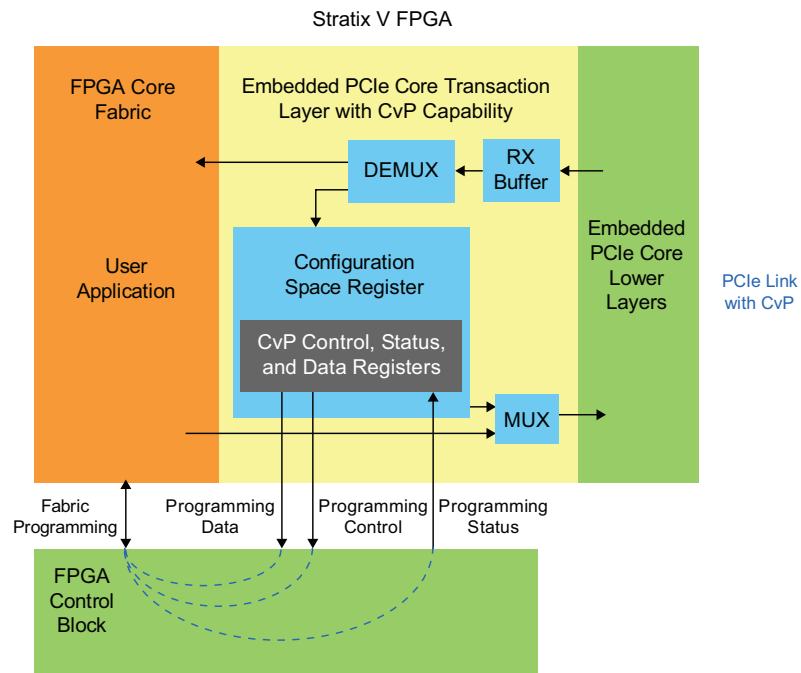
CvP Benefits

Programming the FPGA core fabric via PCIe leverages the capabilities of the autonomous PCIe core. Combined together, these features can deliver any or all of the following key benefits to 28-nm designers:

- Lower system cost—CvP can eliminate one or more parallel flash devices and possibly an external programming controller device. In addition, CvP allows designers to store FPGA programming files in a CPU memory system attached to the FPGA via a PCIe link. Using this technique, only the FPGA I/O programming and PCIe core parameters are stored in the flash device, requiring a smaller and cheaper flash device.
- Smaller board space—Parallel flash devices can be replaced by a single Altera EPCS or EPCQ flash device.
- Reduction of dedicated FPGA configuration pins—Stratix series devices typically require wide, data-path flash devices to store the FPGA programming file. In contrast, EPCS and EPCQ devices require fewer dedicated pins.
- No host-CPU downtime during core fabric updates—No need for a host-CPU stall or reboot following core fabric image updates when the FPGA operates in user mode. CvP is just another software application that the CPU can execute.
- User application image protection—Core fabric image copies are accessible only to the host CPU and can be encrypted and/or compressed.
- Simple user software model—This model uses the PCIe protocol and the user application PCIe topology to initialize single or multiple FPGAs.
- Power saving—Low-power, temporary images can be loaded via software control based on the user application profile. This feature can be useful in portable computers that are battery powered.

CvP Operation

[Figure 8](#) provides an overview of the main building blocks that support CvP and related interfaces in Altera 28-nm FPGAs.

Figure 8. CvP Main Building Blocks and Interfaces in a 28-nm FPGA


CvP operation includes the following sequence:

1. I/O pin configurations, including transceiver-block electrical and logical parameters and the embedded PCIe core functionality, are programmed by the FPGA control block.
2. The FPGA control block reads programming data from the serial, quad, or parallel flash devices.
3. The FPGA core fabric is programmed through CvP.
4. The embedded PCIe endpoint buffers the data and sends it to the control block to program the FPGA core fabric.
5. The host CPU views the CvP system as a collection of PCIe CSRs and data registers.
6. The host CPU transmits FPGA core fabric programming data to the embedded PCIe endpoint and the PCIe device passes these data onto the control block, which in turn programs the FPGA core fabric.
7. CvP software monitors the CvP status register to determine if the control block detects any errors, and reacts accordingly.
8. After CvP completion, the PCIe core switches to the functionality assigned by the application logic in the FPGA. As the FPGA operates in user mode, the host CPU software can switch the PCIe core back to the CvP mode via CSR write transactions.

In PCIe terms, CvP support is a Vendor-Specific Extended Capability (VSEC). New Altera-added registers reside in the CSR. CvP writes in these registers and poll status bits to communicate with the FPGA control block. The new set of VSEC registers includes:

- VSEC capability header.
- VSEC length, revision, and ID.
- 16-bit CvP status register—The host CPU monitors this register to know when to start/stop sending data, or when there has been a programming error that should be treated as an uncorrectable error. There are also bits to reflect whether Encryption (AES) and/or Compression (DC) are enabled.
- 32-bit CvP control register—This register provides mode and programming control. The host CPU software driver can set these bits to initiate CvP events.
- 32-bit CvP data register—This register holds the programming data coming from the embedded PCIe core receiver buffer before sending it on the control block.
- 32-bit JTAG Silicon ID—This read-only register returns the FPGA Silicon ID, which can be used by an Altera programming software to make sure it is using the correct programming file.
- 16-bit user device/board type ID—Provides a user-settable value to distinguish between the different FPGAs that need to be programmed in the PCIe topologies, such as those shown in [Figure 5](#), [Figure 6](#), and [Figure 7](#).

The host CPU programs the core fabric image using the PCIe technology's standard 32-bit memory-mapped I/O (MMIO) or configuration write transactions. As mentioned in a previous section, the I/O ring configuration (including SERDES) and the embedded PCIe CSR content remain unchanged during and after the FPGA image updates. During CvP events, all PCIe base address registers (BARs) are intercepted by the CvP. In normal operating mode, all BARs are available for application use.



PCIe core cold-reset events bring down the PCIe link, but they do not start CvP image-loading events and they do not alter the FPGA core fabric image.

Software Support

The CvP functionality is supported by Altera's Quartus® II development software, by the CvP design flow, and by design examples that demonstrate CvP operation.

Quartus II Software Support

The Quartus II software provides CvP support across all supported platforms and operating systems. The Quartus II software generates the respective programming files required to initialize the embedded PCIe cores or program the external flash devices for the CvP system initialization of the FPGAs that reside in the designer's PCIe topology. The content of the files varies based on the CvP operating modes, as illustrated in [Table 2](#). The Quartus II software also generates one or more separate FPGA core fabric programming files for each FPGA that participates in the PCIe CvP hierarchy.

For example, if a designer uses CvP to configure four FPGAs populated below a PCIe switch, similar to the topology depicted in [Figure 6](#), the Quartus II software generates two types of configuration files: raw binary files and FPGA core images.

Four raw binary files are created to program the four EPCS or EPCQ devices of this topology. Each file contains the programming information necessary to configure the embedded PCIe hard IP and I/O rings of its respective FPGA. The EPCS or EPCQ devices are programmed via one of the regular FPGA programming methods.

The Quartus II software also creates one or more FPGA fabric core images per device, depending on the number of different user application variants per FPGA. In total, there are at least four such files (one per FPGA). In theory, there is no upper limit to the number of FPGA core images per device. The FPGA core image files can use raw binary, encrypted, or compressed format. One of the images (called 'initial' image) is used to initialize the FPGA upon power-up. The initial image can be loaded via CvP or through one of the other FPGA core fabric programming methods depicted in [Table 1](#). All other FPGA core images can be used to update the FPGA core fabric through CvP. Each of the FPGAs in the PCIe topology has a unique 16-bit user device/board type ID value to help direct the host CPU to program the right device through CvP.

CvP Design Flow Support

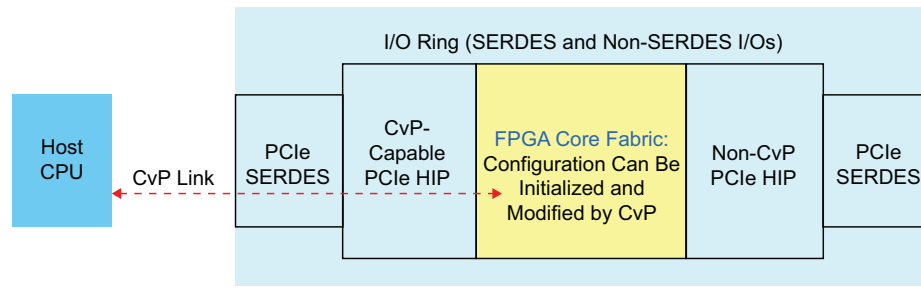
Designers who want to update the FPGA content through CvP must ensure that the I/O ring and embedded PCIe core parameters and functionality remain unchanged. The FPGA image content update through CvP can be viewed as a simple partial reconfiguration with two partitioned regions, where one region (the embedded PCIe hard IP and the I/O ring) remains unchanged, while the other region (FPGA core fabric) can be updated multiple times. This process entails ensuring that the embedded PCIe parameters remain fixed (including the PCIe hard IP that are not used for CvP), as well as maintaining the same I/O functionality.

Some timing and clocking constraints are introduced in the multiple designs that target a single FPGA to ensure migration from the initial design to the other designs. Additionally, in multi-PCIe core applications, only the CvP-capable PCIe core is guaranteed to remain operational during and after the CvP core fabric image update events.

By combining partial reconfiguration with CvP, designers can keep all other PCIe links operational during and after CvP partial core fabric image update events. Partial reconfiguration via CvP is feasible in Altera FPGAs, but it is outside the scope of this white paper.

Altera recommends a design flow that logically locks the I/O ring and the embedded PCIe hard IP cores, and allows designers to attach them to one or multiple FPGA core fabric design images subsequently loaded via the CvP. [Figure 9](#) depicts the partition between the various building blocks, which are configured just once following device power-up (I/O ring including SERDES and PCIe cores), and the FPGA core fabric, which may be updated by CvP as many times as required based on the user application code that runs on the host CPU.

Figure 9. FPGA Design Partitioning with CvP



Design Example Support

The ultimate goal of CvP designers is to initialize and periodically update the FPGA core fabric image in the target systems. From an embedded system designer's viewpoint, CvP is a software application that runs on top of a PCIe device driver that accesses CvP-capable endpoints in their native OS environment.

Altera provides a clear-text C application program design example that targets PCIe development boards under Windows 7 and Linux. The C program initializes the FPGA core fabric through CvP. Designers can use this C program as a starting point for their own code development.

The C program demonstrates the steps needed to implement the CvP algorithm. The C program design example is applicable in both CvP operating modes (Table 2, Modes 2 and 3), and can be used as a software design example for FPGA core fabric image initialization as well as subsequent core fabric image updates.

Conclusion

The 28-nm device portfolio's autonomous embedded PCIe core ensures designers that their FPGA meets the power-up time requirements of the PCIe Base, as well as the PCIe CEM specifications irrespective of the FPGA core fabric size and link-operating mode. This feature guarantees wide-range interoperability with various PCIe-based computer platforms.

Altera's 28-nm FPGA portfolio includes the major CvP feature enhancement that benefits most PCIe-based customer applications. CvP can reduce the product cost, lower the board size, simplify the software usage model, and provide a robust in-field system upgrade capability. CvP enables designers to initialize the FPGA core fabric image and later update it in run time as many times as needed by their applications.

Further Information

- *PCIe Base Specification and PCIe Card Electromechanical Specification:*
www.pcisig.com/specifications/pciexpress/base
- "PCI Express bridging options enable FPGA-based configurable computing," *Programmable Logic Designline*, Mike Alford, Gennum Corp., September 8, 2008:
www.pldesignline.com/howto/210300269

- Stratix V FPGAs: Built for Bandwidth:
www.altera.com/products/devices/stratix-fpgas/stratix-v/stxv-index.jsp
- Literature: Stratix V Devices:
www.altera.com/literature/lit-stratix-v.jsp

Acknowledgements

- Arye Ziklik, Senior Manager, Product Planning, Altera Corporation
- Mario Khalaf, Principal Investigator, Office of the CTO, Altera Corporation

