WP374 (v1.2) May 30, 2012

Partial Reconfiguration of Xilinx FPGAs Using ISE Design Suite

By: David Dye

As systems become more complex and designers are asked to do more with less, FPGA adaptability has become a critical asset. While Xilinx® FPGAs have always provided the flexibility to do on-site device reprogramming, today's tougher cost, board space, and power consumption constraints demand even more efficient design strategies.

Xilinx partial reconfiguration extends the inherent flexibility of the FPGA by allowing specific regions of the FPGA to be reprogrammed with new functionality while applications continue to run in the remainder of the device. Partial reconfiguration addresses three fundamental needs by enabling the designer to:

- Reduce cost and/or board space
- Change a design in the field
- Reduce power consumption

Reduce Cost and Board Space

The two most prevalent user problems addressed by partial reconfiguration are:

- Fitting more logic into an existing device
- Fitting a design into a smaller, less expensive device

Historically, designers have spent days, if not weeks, trying new implementation switches, reworking code, and re-engineering solutions to squeeze them into the smallest possible FPGA. Partial reconfiguration enables these designers to reduce the size of their designs by dynamically time-multiplexing portions of the available hardware resources. The ability to load functions on an as-needed basis also reduces the amount of idle logic, thereby saving additional space.

An example of this strategy is the use of partial reconfiguration within a software defined radio (SDR) system, where the user uploads a new waveform on demand to establish communication with a new channel. Any number of waveforms can be supported by a single hardware platform, requiring only unique partial bitstreams to be available for these waveforms. Established links to other channels are not disrupted by the update to another channel due to the on-the-fly characteristics of partial reconfiguration.

Increase Deployed System Flexibility

In the past, changing a design in the field required new placement and routing of the design and the delivery of a full configuration file. The engineer also had to shut the system down while making the change. In contrast, when using partial reconfiguration, the designer needs only to place and route the modified function in context with the already-verified remainder of the design, then deliver this new partial image to a system in the field.

Moreover, the engineer can dynamically insert new functions while the system is up and running, improving system up-time. Thus, mutually exclusive functions can be plugged into the same space without having to redesign the system or move to a bigger device.

Another example of the benefits of partial reconfiguration is its use within optical transport network (OTN) solutions. Like with SDR, different protocols can be supported to create a more efficient hardware system—only the protocol for a particular channel needed at any point in time is loaded in the FPGA. A deployed system cannot only handle traffic of many different types using minimal resources, but that system can be updated with the latest protocols without requiring a full redesign.

Reduce Power Consumption

Power consumption has become a primary concern for today's designers. Like size and cost, it is a metric with strict limits in most systems. However, as FPGA designs grow in size and complexity, they consume more power. While synthesis and implementation tools coupled with appropriate design techniques can help reduce power consumption, partial reconfiguration implementations can further reduce static and dynamic power.

One way to reduce static power is to simply use a smaller device. With partial reconfiguration, designers can essentially time slice the FPGA and run parts of their design independently. The design then requires a much smaller device or fewer devices because not every part of the design is needed 100% of the time.

Partial reconfiguration also has the potential to reduce operating power as well as static power. For example, many designs must be able to run at a very fast speed, but that maximum performance might only be needed a small percentage of the time. To save power, designers can use partial reconfiguration to swap out a high-performance design with a low power version of the same design—instead of designing exclusively for maximum performance. The designer can then switch back to the high-performance design when the system requires it.

Additional Advantages

The ability to time-multiplex hardware dynamically on a single FPGA offers a number of additional advantages. Partial reconfiguration:

- Provides real-time flexibility in the choice of algorithms or protocols available to an application at any given moment
- Enables the use of new techniques in design security
- Improves FPGA fault tolerance
- Accelerates configurable computing
- Reduces bitstream storage requirements

Some of these advantages are illustrated in Figure 1.

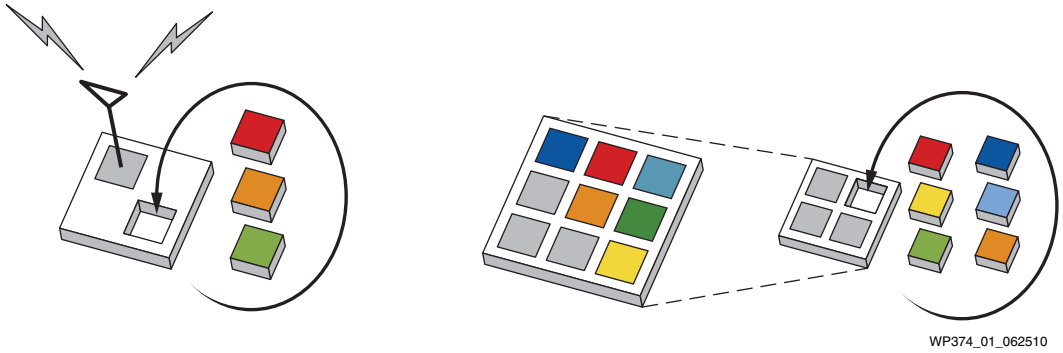

Figure 1: **Modifying Functionality and Reducing Size using Partial Reconfiguration**

The ISE® Design Suite 12 software enabled designers to target Virtex-4, Virtex-5, and Virtex-6 devices for partial reconfiguration applications. ISE Design Suite 13 software supported these devices plus introduced 7 series support. ISE Design Suite 14 completes 7 series support (non-SSI devices only) and introduces partial reconfiguration support for the Zynq-7000 Extensible Processing Platform. This Xilinx solution represents the fourth generation of software support for partial reconfiguration, and the software has evolved considerably over the past decade. Partial reconfiguration is a mature and time-tested design option.

Fine-Grained Partial Reconfiguration Support

Virtex and 7 series architectures have been engineered from the beginning to support dynamic partial reconfiguration. This includes Zynq™ EPPs, Xilinx's extensible processing platform. Each successive generation of silicon has further enhanced the features and granularity of these capabilities, and each generation sees more and more customer activity and validation using partial reconfiguration techniques. For example, Virtex-6 and 7 series architecture support encrypted partial reconfiguration via the built-in AES security features.

Virtex and 7 series architecture support reconfiguration via internal and external configuration ports, and they also offer a great deal of flexibility in how designs get mapped to the architecture. These devices feature a "non-glitching" technology where, even as the functionality of one part of the user design is reconfigured, static portions of the design—especially routing resources—can overlap this reconfigurable region and not be affected by the act of reconfiguration itself. The static routes and unchanging logic that might exist within a reconfigurable zone are simply overwritten with the exact same programming information, and no interruption of functionality occurs.

ISE Software Enables Partial Reconfiguration Design Capabilities

Implementing a partially reconfigurable FPGA design is similar to implementing multiple non-partial reconfiguration designs that share common logic. The designer designates the portions of the FPGA to be reconfigured, both in terms of the physical size of the region and the types of resources desired. Then the designer describes the different module variants that are to occupy that region. Xilinx FPGAs support reconfiguration of CLBs (flip flops, look-up tables, distributed RAM, multiplexers, etc.), block RAM, and DSP blocks, plus all associated routing resources. For some resources that are not, such as global clock management blocks and serial transceivers, the Dynamic Reconfiguration Port (DRP) allows users to change the characteristics of these components on the fly. The ISE software ensures that the resources used to construct the reconfigurable functions are completely contained within the defined physical regions and that no interference with the non-reconfiguring portion of the design occurs.

In the ISE software, the PlanAhead™ design tools manage all the details of building such a design. Multiple netlists, representing the static (non-reconfiguring) logic and each variant of the reconfigurable portions of the design, are loaded in; the designer then creates full FPGA design images with these pieces. Floorplanning, constraint entry, and design rule checks (DRCs) are all accessed through the PlanAhead software environment. Multiple passes through the place-and-route tools are used to generate the necessary bitstreams for all full and partial design images; each pass (called a configuration) represents a complete FPGA design.

Partitions—a technology introduced in the ISE 8.2i software and enhanced in subsequent releases—ensure that the logic and routing common to each of the multiple designs is absolutely identical. Once one design configuration meets all requirements, the designer can reuse the results from that implementation to create the other configurations. After all configurations are implemented, verification routines validate consistency among all the versions. All these checks combine to guarantee a safe environment when loading a partial bitstream into an operating FPGA.

This design flow is illustrated in Figure 2.

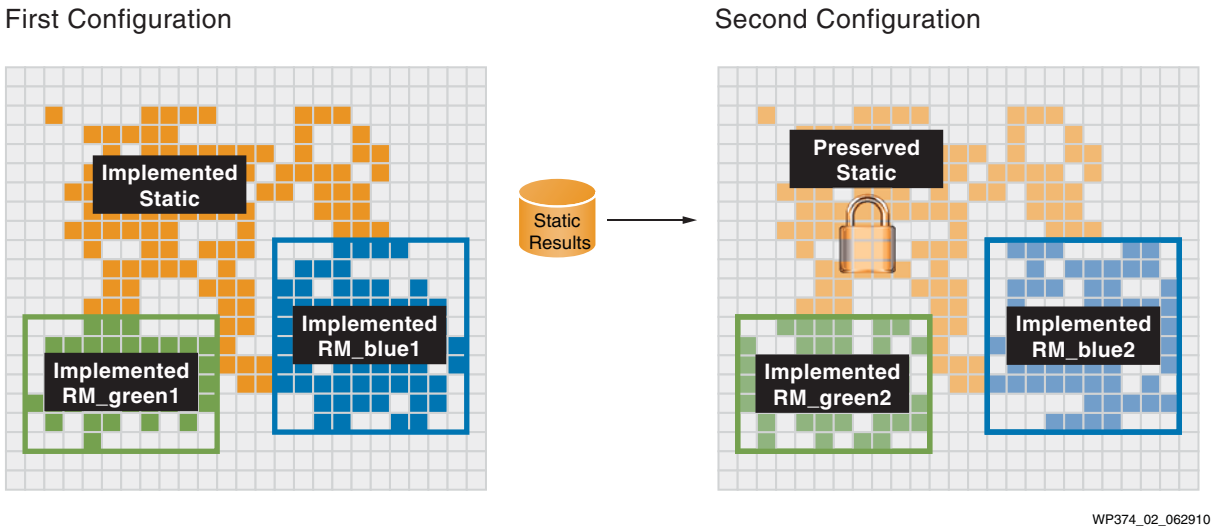

Figure 2: **Partial Reconfiguration Design Flow**

Once these multiple design images are place and routed, designers can use traditional timing analysis, simulation, and verification techniques to validate the results. Having simulated each configuration on its own to confirm that it meets all requirements, the designers can create multiple hierarchical simulation netlists and either validate them on their own or load them into a simulator to emulate the different combinations of modules that can exist during device operation.

Managing Dynamic Device Reconfiguration

To configure the FPGA, designers begin as usual by loading a full design image upon power-up. After the device is fully configured and operational, designers can use partial bit files at any time to modify the pre-defined regions while the rest of the FPGA remains fully active and uninterrupted.

Designers can choose from the following configuration ports to load the partial bitstream: Slave SelectMAP, Slave Serial, JTAG, or Internal Configuration Access Port (ICAP, an internal representation of the SelectMAP interface), which can be used with the Integrated Block for PCIe® to partially reconfigure over PCIe. Partial bitstreams are typically stored in flash memory, and bitstream delivery can be managed by a microprocessor or by routines programmed into the FPGA. System designers can manage the initiation of reconfiguration and delivery of the partial configuration image using a wide variety of techniques, two of which are shown in Figure 3.

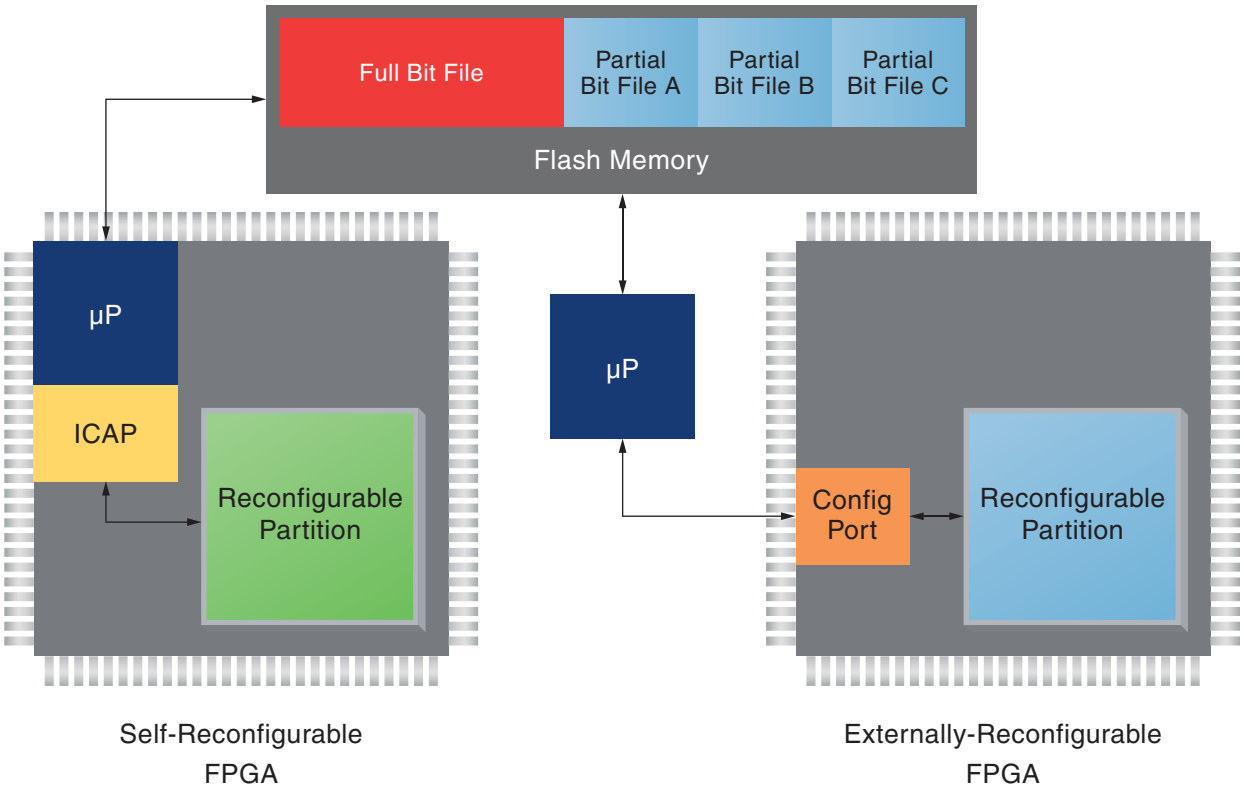

Figure 3: **Two Methods of Delivering a Partial Bit File**

Design of the self-reconfigurable FPGA on the left in Figure 3 uses a common application in which a small microprocessor is used to read the partial bitfile out of the flash and send the data to the internal configuration port (ICAP). In many cases, a simple state machine can extract the partial bitfiles from the flash and port the data to the ICAP eliminating the need for a microprocessor. If the system already has a microprocessor, a simple solution is to use the external processor to read the partial bitfile from the flash and send the data to the standard configuration port of the FPGA, shown in Figure 3 as an externally-reconfigurable FPGA.

Partial bitstreams contain all the configuration commands and data necessary for partial reconfiguration. A single FPGA configuration engine handles both full configuration and dynamic partial reconfiguration, so the exact same type of programming information is processed by the same programming mechanism. The task of loading a partial bitstream into an FPGA does not require knowledge of the physical location of the reconfigurable module. Since configuration frame addressing information is included in the partial bitstream, it cannot be sent to the wrong part of the FPGA.

Additional features and capabilities are also available for partial reconfiguration. Encrypted bitstreams can be delivered to the ICAP, taking advantage of the dedicated decryption capabilities found in Virtex-6 and 7 series devices. A solution for encrypted Virtex-5 partial bitstreams is available in XAPP887, *PRC/EPRC: Data Integrity and Security Controller for Partial Reconfiguration*. This application note provides a mechanism for checking the integrity of partial bitstreams before they are loaded into the active FPGA. This is done by inserting CRC calculations into the bitstream and

then checking them prior to delivery to the configuration port. Variations of any of these solutions can also be engineered by the designer as well.

PCIe Applications

The PCI Express specification requires ports to be ready for link training at a minimum of 100 ms after the power supply is stable. This becomes a difficult task due to the ever-increasing configuration memory size of each new generation of FPGA. One innovative approach to addressing this challenge is to leverage partial reconfiguration to split the overall configuration of a PCIe specification based system in a large FPGA into two sequential steps:

1. Initial PCIe system link configuration
2. Subsequent user application reconfiguration

It is feasible to configure only the FPGA's integrated block for PCIe and associated logic during the first stage within the 100 ms window before the fundamental reset is released. Using partial reconfiguration, the host can then configure the remainder of FPGA to apply the user application via the now-active PCIe system link as shown in Figure 4. The host can also update the user application through the PCIe bus—without losing the connection to the host, which is typical in a traditional full reconfiguration.

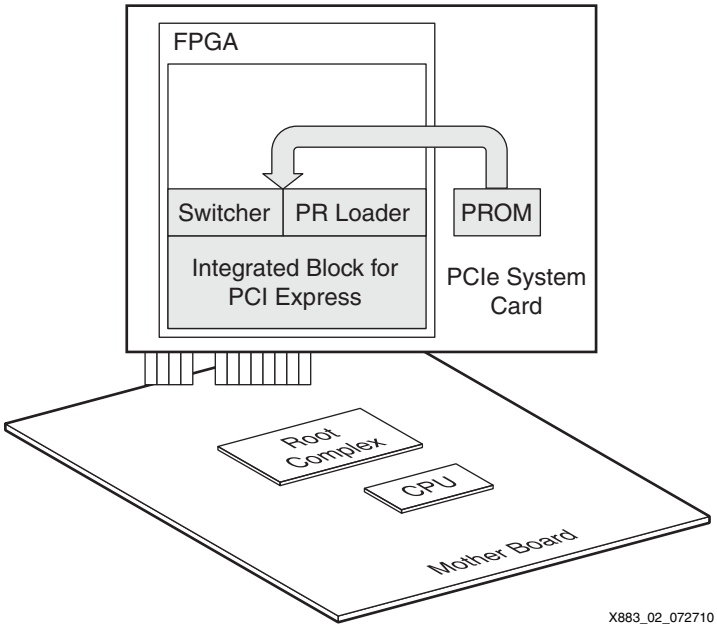

Figure 4: **Fast Configuration of a PCIe System Using Partial Reconfiguration**

This methodology not only provides a solution for faster PCIe system configuration, it enhances user application security because the bitstream is only accessible by the host and can be better encrypted. This approach also helps lower the system cost by reducing external configuration component costs and board space. This solution was developed for Virtex-6 FPGAs, as described in XAPP883: *Fast Configuration of PCI Express Technology through Partial Reconfiguration*.

Conclusion

Partial reconfiguration is a powerful solution that can dramatically extend the capabilities of Xilinx FPGAs. In addition to the potential for reducing size, weight, power, and cost, partial reconfiguration enables new types of FPGA designs that provide efficiencies unattainable with conventional design techniques.

For more information on the Xilinx Partial Reconfiguration solution, go to: http://www.xilinx.com/tools/partial-reconfiguration. This page links to all partial reconfiguration user guides, tutorials, and application notes. Additionally, a video demonstration of the PlanAhead software flow is available: https://xilinx.webex.com/xilinx/lsr.php?AT=pb&SP=EC&rID=42731112&rKey=ce8d9518cc1fda63.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
07/23/10	1.0	Initial Xilinx release.
07/06/11	1.1	Updated to include 7 series architecture and devices. Added PCIe Applications. Updated Conclusion.
05/30/12	1.2	Updated for ISE Design Suite 14 and Zynq-7000 EPPs.

Notice of Disclaimer