WP374 (v1.0) July 23, 2010

Partial Reconfiguration of Virtex FPGAs in ISE 12

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—the industry’s only dynamically reconfigurable programmable logic solution—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.

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. 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.

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 it.

First, shrinking the design to smaller or fewer parts reduces static power consumption.

Second, the designer can remove or replace power-hungry functions when those functions are not needed. Time-multiplexing the FPGA's silicon resources in this way reduces dynamic power consumption.

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.

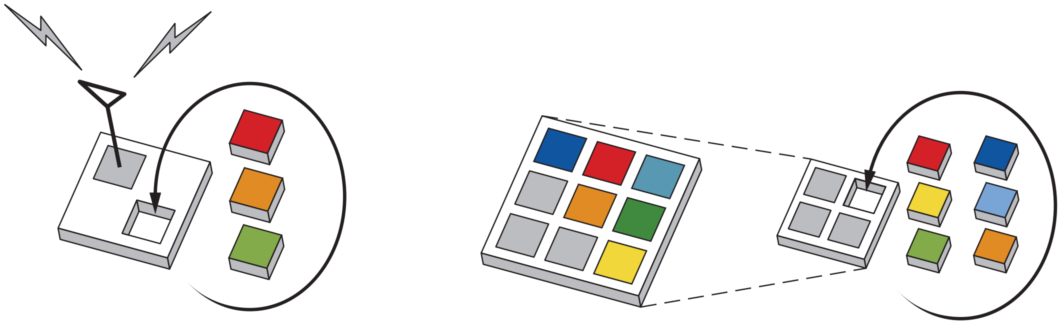

Figure 1: **Modifying Functionality and Reducing Size using Partial Reconfiguration**

The ISE® 12 software enables designers to target Virtex®-4, Virtex-5, and Virtex-6 devices for partial reconfiguration applications. This Xilinx solution represents the fourth generation of software support for partial reconfiguration; as a result, the software has evolved considerably over the past decade. Partial reconfiguration is a mature and time-tested design option.

Virtex Devices Offer Fine-Grained Partial Reconfiguration Support

Virtex architectures have been engineered from the beginning to support dynamic partial reconfiguration. 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 FPGAs support encrypted partial reconfiguration via the built-in AES security features.

Virtex devices 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. Nearly all FPGA resources are reconfigurable, including block RAM, DSP blocks, and I/O. 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 12 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 ISE 12—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.

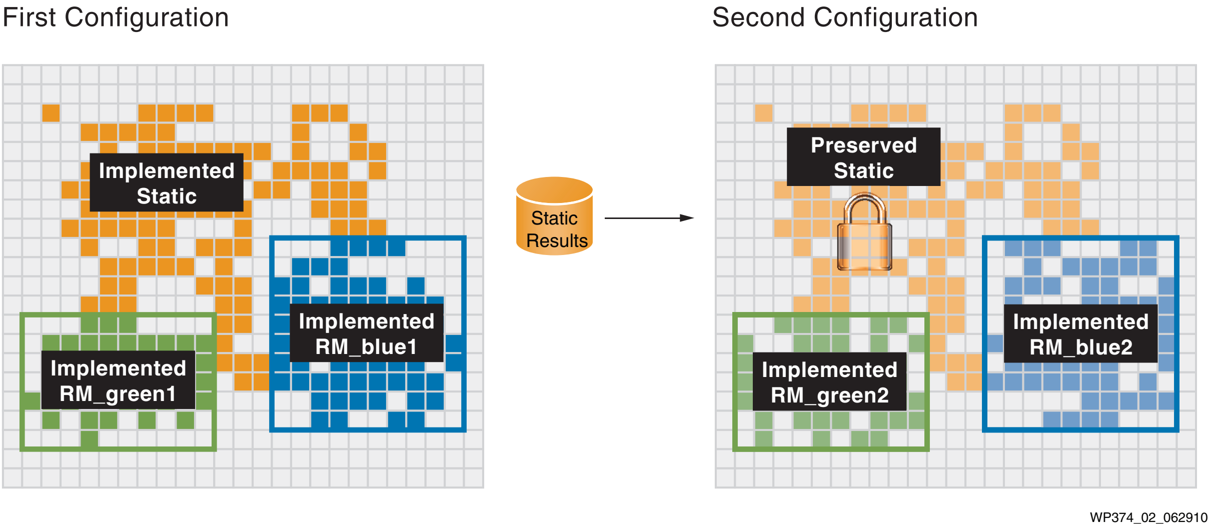

Figure 2: **Partial Reconfiguration Design Flow**

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 PCIe block 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.

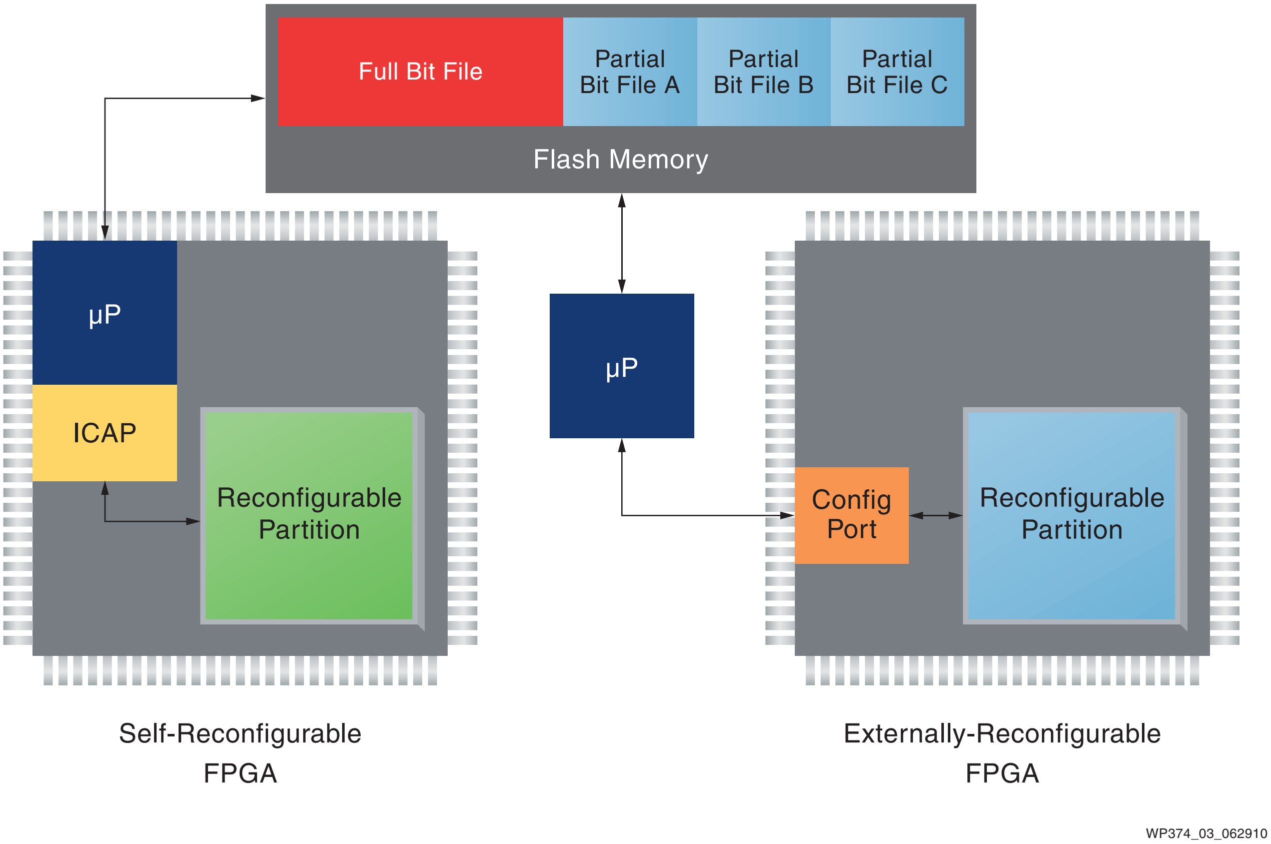

Figure 3: **Two Methods of Delivering a Partial Bit File**

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.

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.

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.

Notice of Disclaimer

The information disclosed to you hereunder (the "Information") is provided "AS-IS" with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.