13. Back-End Design Flow for HardCopy Series Devices

H51019-1.4

Introduction

This chapter discusses the back-end design flow executed by the HardCopy® Design Center when developing your HardCopy series device. The chapter is divided into two sections:

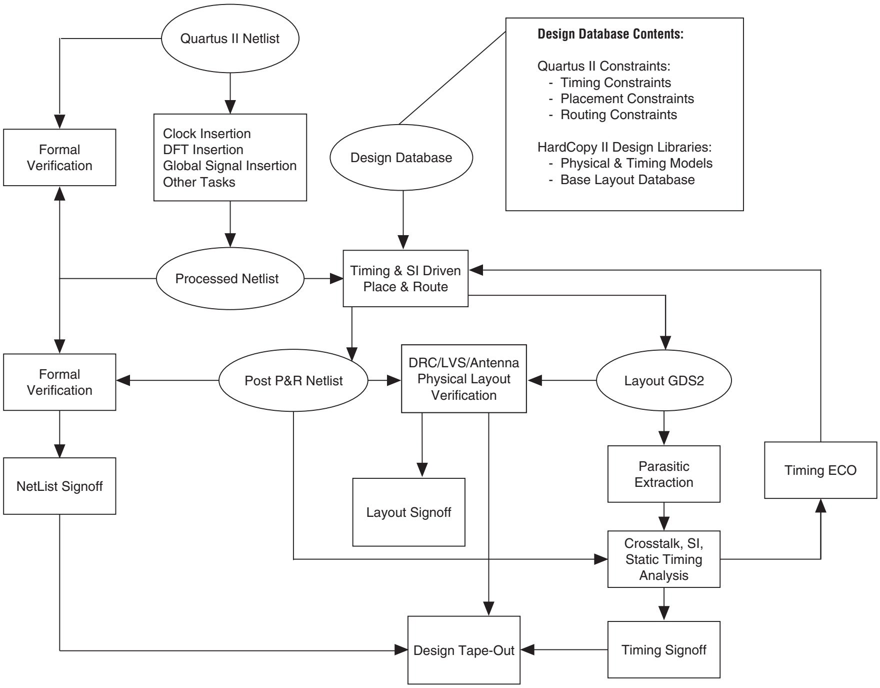
- HardCopy II Back-End Design Flow
- HardCopy Stratix® and HardCopy APEX™ Back-End Design Flow

☞ For more information on the HardCopy II, HardCopy Stratix, and HardCopy APEX families, refer to the respective sections for these families in the *HardCopy Series Handbook*.

HardCopy II Back-End Design Flow

This section outlines the back-end design process for HardCopy II devices, which occurs in several steps. Figure 13–1 illustrates these steps. The design process uses both proprietary and third-party EDA tools. The HardCopy II device design flow is different from that of previous HardCopy families (HardCopy Stratix and HardCopy APEX devices). The following sections outline these differences.

Figure 13–1. HardCopy II Back-End Design Flow



Device Netlist Generation

For HardCopy II designs, the Quartus® II software generates a complete Verilog gate-level netlist of your design. The HardCopy Design Center uses the netlist to start the migration process. HardCopy Stratix and HardCopy APEX designs use the SRAM Object file (**.sof**) to program the FPGA, as the primary starting point for generating the HardCopy device netlist.

HardCopy Stratix and HardCopy APEX designs use the **.sof** file to program the FPGA, as the primary starting point for generating the HardCopy device netlist. In addition to the Verilog gate level netlist and the **.sof** file, the Quartus II software generates additional information as part of the design database submitted to the HardCopy Design Center. This information includes timing constraints, placement constraints, global routing information, and much more. Generation of this database provides the HardCopy Design Center with the necessary information to complete the design of your HardCopy II device.

Design for Testability Insertion

The HardCopy Design Center inserts the necessary test structures into the HardCopy II Verilog netlist. These test structures include full-scan capable registers and scan chains, JTAG, and memory testing. After adding the test structures, the modified netlist is verified using third-party EDA formal verification software against the original Verilog netlist to ensure that the test structures have not broken your netlist functionality. The "Formal Verification of the Processed Netlist"section explains the formal verification process.

Clock Tree and Global Signal Insertion

Along with adding testability, the HardCopy Design Center adds an additional local layer of clock tree buffering to connect the global clock resources to the locally placed registers in the design. Global signals with high fan-out may also use dedicated Global Clock Resources built into the base layers of all HardCopy II devices. The HardCopy Design Center does local buffering.

Formal Verification of the Processed Netlist

After all design-for-testability logic, clock tree buffering, and global signal buffering are added to the processed netlist, the HardCopy Design Center uses third-party EDA formal verification software to compare the processed netlist with your submitted Verilog netlist generated by the Quartus II software. Added test structures are constrained to bypass mode during formal verification to verify that your design's intended functionality was not broken.

Timing and Signal Integrity Driven Place and Route

Placement and global signal routing is principally done in the Quartus II software before submitting the HardCopy II design to the HardCopy Design Center. Using the Quartus II software, you control the placement and timing driven placement optimization of your design. The Quartus II software also does global routing of your signal nets, and passes this information in the design database to the HardCopy Design Center to do the final routing. After submitting the design to the HardCopy Design Center, Altera® engineers use the placement and global routing information provided in the design database to do final routing and timing closure and to perform signal integrity and crosstalk analysis. This may require buffer and delay cell insertion in the design through an engineering change order (ECO). The resulting post-place and route netlist is verified again with the source netlist and the processed netlist to guarantee that functionality was not altered in the process.

Parasitic Extraction and Timing Analysis

After doing placement and routing on the design by the HardCopy Design Center, it generates the `gds2` design file and extracts the parasitic resistance and capacitance values for timing analysis. Parasitic extraction uses the physical layout of the design stored in a **.gds2** file to extract these resistance and capacitance values for all signal nets in the design. The HardCopy Design Center uses these parasitic values to calculate the path delays through the design for static timing analysis and crosstalk analysis.

Layout Verification

When the Timing Analysis reports that all timing requirements are met, the design layout goes into the final stage of verification for manufacturability. The HardCopy Design Center performs physical Design Rule Checking (DRC), antenna checking of long traces of signals in the layout, and a comparison of layout to the design netlist, commonly referred to as Layout Versus Schematic (LVS). These tasks guarantee that the layout contains the exact logic represented in the place-and-route netlist, and the physical layout conforms to 90-nm manufacturing rules.

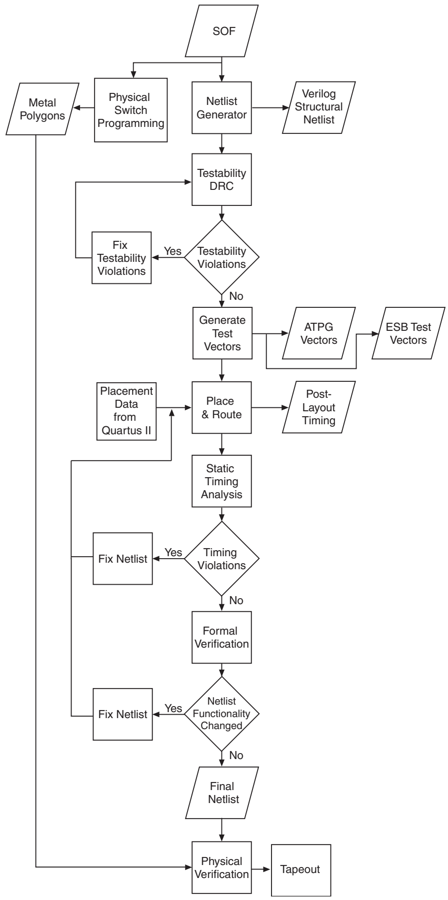
Design Signoff

The Altera HardCopy II back-end design methodology has a thorough verification and signoff process, guaranteeing your design's functionality. Signoff occurs after confirming the final place-and-route netlist functional verification, confirming layout verification for manufacturability, and the timing analysis reports meeting all requirements. After achieving all three signoff points, Altera begins the manufacturing of the HardCopy II devices.

HardCopy Stratix and HardCopy APEX Migration Flow

Design migration for HardCopy Stratix and HardCopy APEX devices occurs in several steps, outlined in this section and shown in Figure 13–2. The migration process uses both proprietary and third-party EDA tools.

Figure 13–2. HardCopy Stratix and HardCopy APEX Migration Flow Diagram



Netlist Generation

For HardCopy Stratix and HardCopy APEX designs, Altera migrates the Quartus II software-generated **.sof** file to a Verilog HDL structural netlist that describes how the following structural elements are configured in the design and how each structural element is connected to other structural elements:

- Logic element (LE)
- Phase-locked loop (PLL)
- Digital signal processing (DSP) block
- Memory block
- Input/output element (IOE)

The information that describes the structural element configuration is converted into a physical coordinate format so that metal elements can be implemented on top of the pre-defined HardCopy series device-base array. Using the **.sof** file for netlist extraction helps ensure that the HardCopy series device contains the same functional implementation that was used in the FPGA version of the design.

Testability Audit

The Design Center performs an audit for testability violations when the Verilog HDL netlist is available. This audit ensures that all built-in scan chain structures will work reliably while testing the HardCopy series devices. Certain circuit structures, such as gated clocks, gated resets, oscillators, pulse generators, or other types of asynchronous circuit structures makes the performance of scan chain structures unreliable. During the testability audit, all such circuit structures are detected and disabled when the device is put into test mode.

Placement

Beginning with version 4.2, the Quartus II software supports all HardCopy series devices. The HardCopy Timing Optimization Wizard in the Quartus II software is used for HardCopy Stratix devices and generates placement information of the design when it is mapped to the HardCopy Stratix base array. This placement information is read in and directly used by the place-and-route tool during migration to the equivalent HardCopy Stratix device.

For more information on how to use the HardCopy Timing Optimization Wizard, refer to the *Quartus II Support for HardCopy Stratix Devices* chapter. For more information on Quartus II features for HardCopy II devices, refer to the *Quartus II Support for HardCopy II Devices* chapter.

To generate placement data, the Quartus II software uses the **.sof** file to generate the netlist, as described in "Netlist Generation" on page 13–6. The netlist is then read into a place-and-route tool. The placement optimization is based on the netlist connectivity and the design's timing constraints. The placement of all IOEs is fixed. After placement is complete, the Quartus II software generates the scan chain ordering information so the scan paths can be connected.

Test Vector Generation

Memory test vectors and memory built-in self-test (BIST) circuitry ensure that all memory bits function correctly. Automatic test pattern generation (ATPG) vectors test all LE, DSP, and IOE logic. These vectors ensure that a high stuck-at-fault coverage is achieved. The target fault coverage for all HardCopy Stratix devices is near 100%.

When the testability audit is successfully completed and the scan chains have been re-ordered, the Design Center can generate memory and ATPG test vectors. When test vector generations are complete, they are simulated to verify their correctness.

Routing

Routing involves generating the physical interconnect between every element in the design. At this stage, physical design rule violations are fixed. For example, nodes with large fan-outs need to be buffered. Otherwise, these signal transition times are too slow, and the device's power consumption increases. All other types of physical design rule violations are also fixed during this stage, such as antenna violations, crosstalk violations, and metal spacing violations.

Extracted Delay Calculation

Interconnect parasitic capacitance and resistance information is generated after the routing is complete. This information is then converted into a Standard Delay File (**.sdf)** with a delay calculation tool, and timing is generated for minimum and maximum delays.

Static Timing Analysis and Timing Closure

The design timing is checked and corrected after place and route using the post-layout generated **.sdf** file. Setup time violations are corrected in two ways. First, extra buffers can be inserted to speed up slow signals. Second, if buffer insertion does not completely fix the setup violation, the placement can be re-optimized.

Setup time violations are rare in HardCopy II and HardCopy Stratix devices because the die sizes are considerably smaller than the equivalent Stratix II and Stratix devices. Statistically, the interconnect loading and distance is much smaller in HardCopy Stratix devices, so the device operates at a higher clock frequency. Hold-time violations are fixed by inserting delay elements into fast data paths.

As part of the timing analysis process, crosstalk analysis is also performed to remove any crosstalk effects that could be encountered in the device after it has been manufactured. This ensures signal integrity in the device resulting in proper functionality and satisfactory performance.

After implementing all timing violation corrections in the netlist, the place and route is updated to reflect the changes. This process is repeated until all timing violations are removed. Typically, only a single iteration is required after the initial place and route. Finally, static functional verification is tested after this stage to double-check the netlist integrity.

Formal Verification

After any change to the netlist, you must verify its integrity through static functional verification (or formal verification) techniques. These techniques show whether two versions of a design are functionally identical when certain constraints are applied. For example, after test fixes, the netlist must be logically equivalent to the netlist state before test fixes, when the test mode is disabled. This technique does not rely on any customer-supplied functional simulation vectors. Altera uses third-party formal verification software to confirm that the back-end implementation matches the netlist generated from the FPGA's **.sof** programming file.

Physical Verification

Before manufacturing the metal customization layers, the physical programming information must be verified. This stage involves cross-checking for physical design rule violations in the layout database, and also checking that the circuit was physically implemented correctly. These processes are commonly known as running design rule check and layout-versus-schematic verification.

Manufacturing

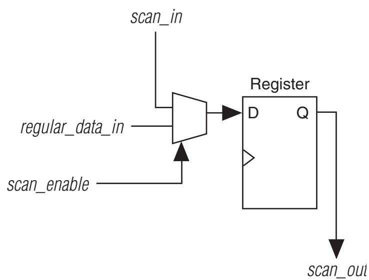
Metallization masks are created to manufacture HardCopy series devices. After manufacturing, the parts are tested using the test vectors that were developed as part of the implementation process.

Testing

HardCopy series devices are fully tested as part of the manufacturing process. Testing does not require user-specific simulation vectors, because every HardCopy series device utilizes full scan path technology. This means that every node inside the device is both controllable and observable through one or more of the package pins of the device. The scan paths, or "scan chains," are exercised through ATPG. This ensures a high-confidence level in the detection of all manufacturing defects.

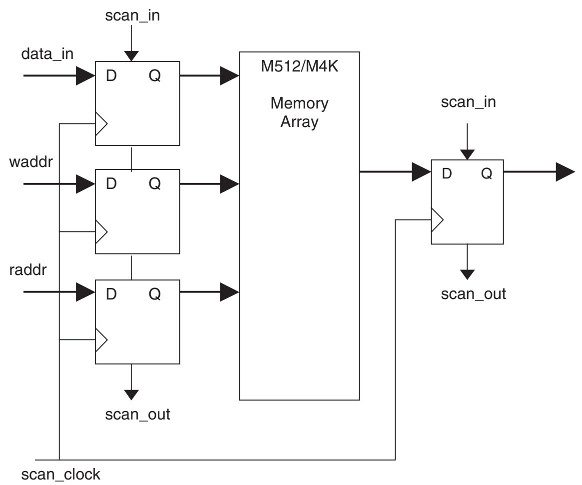
Every register in the HardCopy series device belongs to a scan chain. Scan chains are test features that exist in ASICs to ensure that there is access to all internal nodes of a design. With scan chains, defective parts can be screened out during the manufacturing process. Scan chain registers are constructed by combining the original FPGA register with a 2-to-1 multiplexer. In normal user mode, the multiplexer is transparent to the user. In scan mode, the registers in the device are connected into a long-shift register so that automatic test pattern generation vectors can be scanned into and out of the device. Several independent scan chains exist in the HardCopy series device to keep scan chain lengths short, and are run in parallel to keep tester time per device short. Figure 13–3 shows a diagram of a scan register.

Figure 13–3. HardCopy Stratix Scan Chain Circuitry



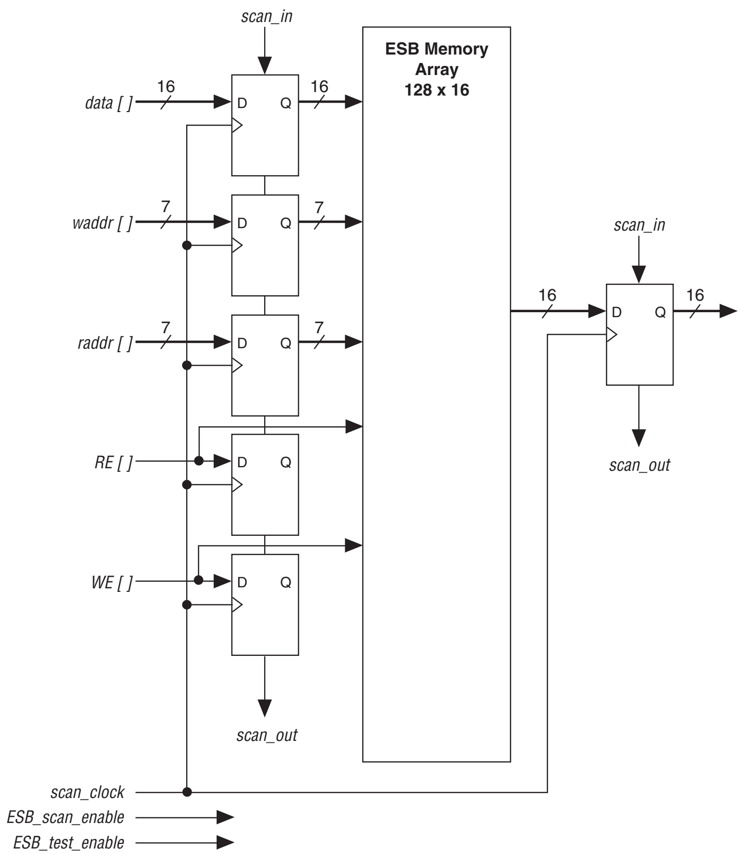
In addition to the scan circuitry (Figure 13–3), which is designed to test all LEs and IOEs, both M512 and M4K blocks (Figure 13–4) have the same scan chain structure so that all bits inside the memory array are tested for correct operation. The M512 and M4K RAM bits are tested by scanning data into the M512 and M4K blocks' `data_in`, write address (`waddr`), and read address (`raddr`) registers. After each vector has been scanned into the HardCopy Stratix device, a write enable (`WE`) pulse is generated to write the data into the M512 and M4K blocks. A read enable (`RE`) pulse is also generated to read data out of the M512 and M4K blocks. The data read back from the M512 and M4K blocks are scanned out of the device via the `data_out` registers. Figure 13–4 shows the M512 and M4K blocks' scan chain connectivity.

Figure 13–4. HardCopy Stratix M512 and M4K Block Scan Chain Connectivity



For HardCopy APEX devices, every embedded system block (ESB) contains dedicated test circuitry so that all bits inside the memory array are tested for correct operation. Access to the ESB memory is also facilitated through scan chains. The ESB also offers an ESB test mode in which the ESB is reconfigured into a 128 × 16 RAM block. In this mode, data is scanned into the ESB I/O registers and written into the ESB memory. For ESBs configured as product-term logic or ROM, the write-enable signal has no effect on the ESB memory array data. When the test mode is disabled (the default), the ESB reverts to the desired user functionality. Figure 13–5 shows the ESB test mode configuration.

Figure 13–5. HardCopy APEX ESB Test Mode Configuration



PLLs and M-RAM blocks are tested with BIST circuitry and test point additions. All test circuitry is disabled once the device is installed into the end user system so that the device then behaves in the expected normal functional mode.

Unused Resources

Unused resources in a customer design still exist in the HardCopy base. However, these resources are configured into a "parked" state. This is a state where all input pins of an unused resource are tied off to V_{CC} or GND so that the resource is in a low-power state. This is achieved using the same metal layers that are used to configure and connect all resources used in the design.

Conclusion

The HardCopy series back-end design methodology ensures that your design seamlessly migrates from your prototype FPGA to a HardCopy device. This methodology, matched with Altera's unique FPGA prototyping and migration process, provides an excellent way for you to develop your design for production.

For more information about how to start building your HardCopy series design, contact your Altera Field Applications Engineer.

For more information on HardCopy products and solutions, refer to the *HardCopy Series Handbook*.

Document Revision History

Table 13–1 shows the revision history for this chapter.

Table 13–1. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
September 2008, v1.4	Updated chapter number and revision history.	—
June 2007, v1.3	Minor text edits.	—
December 2006 v1.2	Added revision history.	—
March 2006	Formerly chapter 13; no content change.	—
October 2005 v1.1	• Graphic updates • Minor edits	—
January 2005 v1.0	Initial release of Chapter 13, Back-End Design Flow for HardCopy Series Devices.	—