1. HardCopy III Design Flow Using the Quartus II Software

HIII53001-3.3

This chapter provides recommendations for HardCopy® III development, planning, and settings considerations in the Quartus® II software. HardCopy III ASIC devices are Altera's low-cost, high-performance, and low-power ASICs with pin-outs, densities, and architecture that complement Stratix® III FPGA devices. Using the Quartus II software, you can design with one set of register transfer level (RTL) code and one IP set for both Stratix III and HardCopy III implementations. This information helps ensure that your design mapping from the Stratix III FPGA to the HardCopy III ASIC is successful.

In the Quartus II software version 10.1 SP1, both companion and compilation for the HardCopy III family are supported. When you select a HardCopy III device from the Quartus II software, the Stratix III device is compatible with the HardCopy III device in the areas of pins, I/O standards, logic, and other resources.

HardCopy III Development Flow

In the Quartus II software, two methods are available for the HardCopy III development flow: Stratix III device first flow and HardCopy III device first flow.

- Stratix III device first flow—Design the Stratix III device first for system functional verification and then create the HardCopy III companion device. Performing system verification early helps reduce overall total project development time.
- HardCopy III device first flow—Design the HardCopy III device first and then create the Stratix III companion device for system functional verification. This method more accurately predicts the maximum performance of the HardCopy III ASIC during development. If you optimize your design to maximize HardCopy III ASIC performance, but are unable to meet your performance requirements with the Stratix III FPGA, you can still map your design with decreased performance requirements for in-system verification.

Whichever design flow you choose for your HardCopy III development, both the target design and the companion device design must be in one Quartus II project.

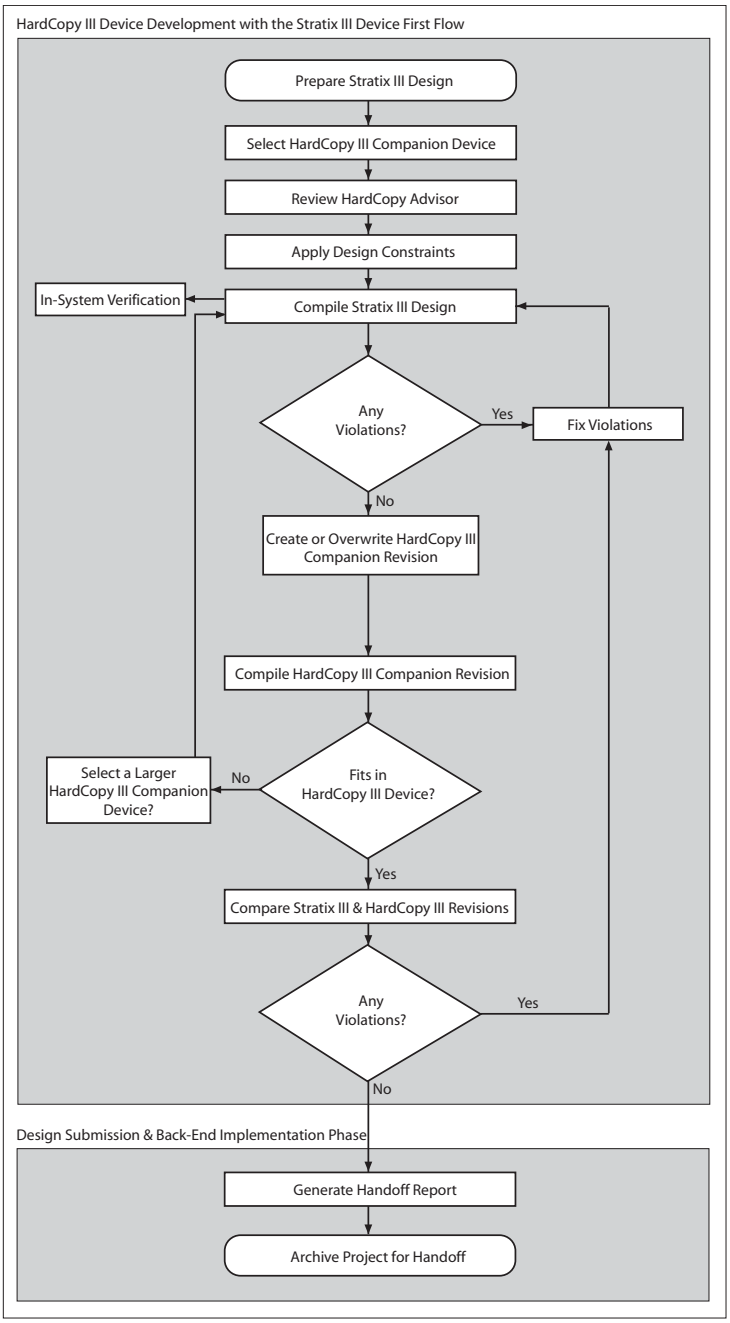
Designing with the Stratix III Device First Flow

The HardCopy III development flow beginning with the Stratix III prototype is very similar to a traditional Stratix III design flow, but requires that you perform a few additional tasks to map the design to the HardCopy III companion device:

1. Choose a Stratix III device for prototyping.
2. Specify a HardCopy III device for conversion.
3. Compile the Stratix III design.
4. Create and compile the HardCopy III companion revision.
5. Compare the HardCopy III companion revision compilation to the Stratix III device compilation.
6. Generate the handoff files and reports.
7. Archive the design and send it to Altera to start the back-end design process.

Figure 1–1 shows the development process for designing with a Stratix III device first and creating a HardCopy III companion device second.

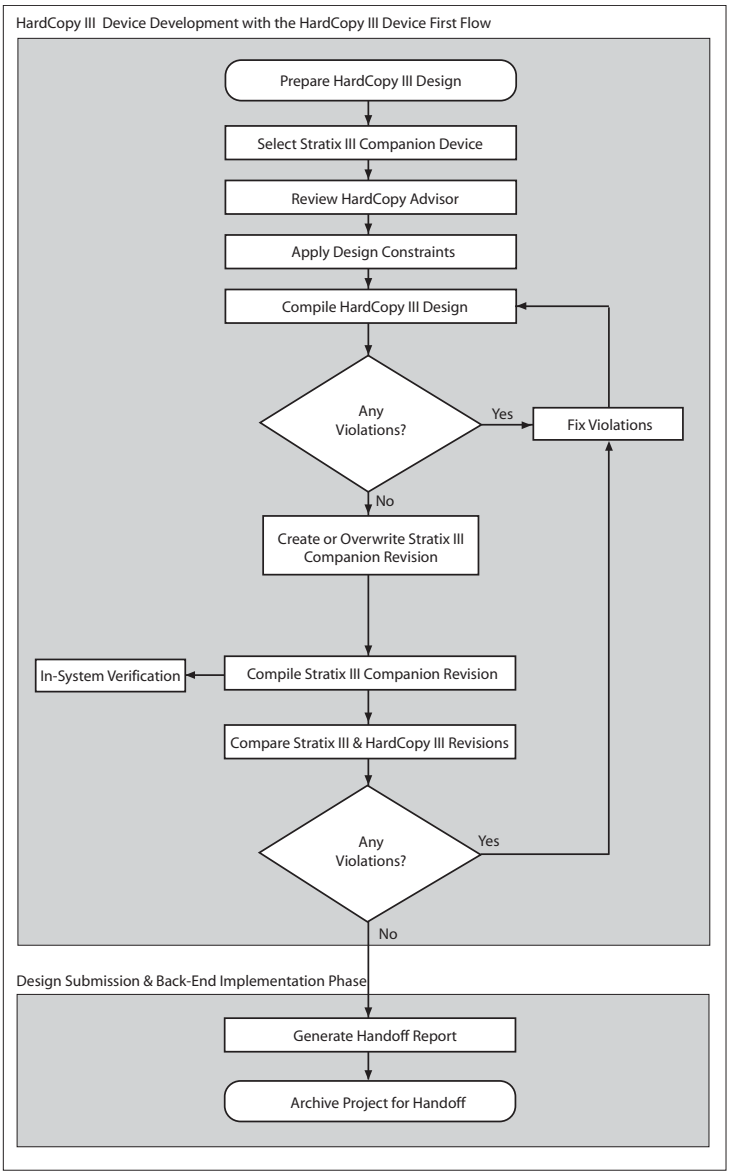
Figure 1–1. Designing with the Stratix III Device First Flow



Designing with the HardCopy III Device First Flow

Designing with the HardCopy III device first flow in the Quartus II software allows you to maximize performance in the HardCopy III device and map the design to the Stratix III prototype for in-system verification. The performance of the Stratix III prototype may be less than the HardCopy III device. For this design flow, you must select **HardCopy III** as the target device and **Stratix III** as the companion device in the **Device Settings** dialog box. The remaining tasks required to complete your design are outlined in Figure 1–2. The HardCopy Advisor adjusts its list of tasks based on the device family you start with to guide you through the development process.

Figure 1–2. Designing with the HardCopy III Device First Flow



HardCopy Advisor

The HardCopy Advisor in the Quartus II software plays an important role in HardCopy III device development. The HardCopy Advisor guides you through a sequence of recommendations, descriptions, and actions. You can track your design progress, generate the design, and complete the comparison archiving and handoff file that you send to the Altera® HardCopy Design Center.

To develop the HardCopy III design, run the HardCopy Advisor in the Quartus II software after you select the Stratix III device and the HardCopy III companion devices. To run the HardCopy Advisor, on the Project menu, point to **HardCopy Utilities** and click **HardCopy Advisor.**

FPGA and HardCopy Companion Device Planning

For both the HardCopy III device and Stratix III prototype planning, the first stage is to choose the device family, device density, speed grade, and package that best suits your design needs. Assuming a Stratix III device first flow, select the Stratix III device and HardCopy III companion device based on the best resource balance for your design requirements. Perform this task before compiling your design in a third-party synthesis tool or the Quartus II software.

For information about the features available in each device density, including logic, memory blocks, multipliers, and phase-locked loops (PLLs), as well as the various package offerings and I/O pin counts, refer to volume 1 of the HardCopy III Device Handbook.

Logic Resources

During HardCopy III device planning, determine the required logic density of the Stratix III prototype and HardCopy III companion device. Devices with more logic resources can implement larger and potentially more complex designs. Smaller devices have less logic resources available and benefit from lower power consumption. Select a device that meets your design needs with some margin, in case you want to add more logic later in the design cycle.

For information about logic resources in HardCopy III devices, refer to the *HardCopy III Device Family Overview chapter.*

I/O Pin and Package Offering

HardCopy III devices offer pin-to-pin compatibility with Stratix III prototypes, making them drop-in replacements for FPGAs. Due to this compatibility, the same system board and software developed for the FPGA prototype can be retained, enabling faster time to market for high volume production.

HardCopy III devices also offer non-socket replacement mapping for further cost reductions. For example, you can map the EP3SL110 device in the 780-pin FBGA package to the HC325 device in the 484-pin FBGA package. Because the pinout for the two packages are not the same, a separate board design is required for the Stratix III device and the HardCopy III device.

For the non-socket replacement path, be sure to select I/Os in the Stratix III device that can be mapped to the HardCopy III device. Not all I/Os in the Stratix III device are available in the HardCopy III non-socket replacement device. Check the pinout information for both the Stratix III device and the HardCopy III device to ensure that you can map successfully and be sure to select the HardCopy III companion device when designing for the Stratix III device.

When mapping a specific Stratix III device to a HardCopy III companion device, there are a number of FPGA prototype choices. Table 1–1 shows the mapping options for the Stratix III prototype and HardCopy III companion device by package.

Table 1–1. Stratix III FPGA to HardCopy III ASIC Mapping Paths

HardCopy III ASIC		Stratix III FPGA Prototype and Package *(Note 1)*												
Device	Package	EP3SL110	EP3SL150		EP3SE110		EP3SL200			EP3SE260			EP3SL340	
		F780	F780	F1152	F780	F1152	H780	F1152	F1517	H780	F1152	F1517	H1152	F1517
HC325	484-pin FineLine BGA	✓ *(2)*	✓ *(2)*	—	✓ *(2)*	—	✓ *(2)*	—	—	✓ *(2)*	—	—	✓ *(2)*	—
	780-pin FineLine BGA	✓	✓	—	✓	—	✓ *(3)*	—	—	✓ *(3)*	—	—	✓ *(2)*	—
HC335	1152-pin FineLine BGA	—	—	✓	—	✓	—	✓	—	—	✓	—	✓ *(3)*	—
	1517-pin FineLine BGA	—	—	—	—	—	—	—	✓	—	—	✓	—	✓

Notes to Table 1–1:

(1) HardCopy III device migration paths are not supported for the EP3SL50, EP3SL70, EP3SE50, and EP3SE80 Stratix III devices.

(2) This mapping is a non-socket replacement path that requires a different board design for the Stratix III device and the HardCopy III device.

(3) The Hybrid FBGA package requires additional unused board space along the edges beyond the footprint, but its footprint is compatible with the regular FBGA package.

For more information about I/O features in HardCopy III devices, refer to the *HardCopy III Device I/O Features chapter in volume 1 of the HardCopy* III *Device Handbook.*

For more information about HardCopy III device packages, refer to the *HardCopy III Device Family Overview chapter in volume 1 of the HardCopy* III *Device Handbook.*

Memory Resources

The TriMatrix memory in HardCopy III devices supports the same memory functions and features as Stratix III devices. You can configure each embedded memory block to be a single- or dual-port RAM, FIFO, ROM, or shift register using the MegaWizard™ Plug-In Manager in the Quartus II software.

HardCopy III embedded memory consists of memory logic array blocks (MLABs), M9K, and M144K memory blocks and has a one-to-one mapping from the Stratix III memory. However, the number of available memory blocks differs based on density, package, and Stratix III FPGA-to-HardCopy III ASIC mapping paths.

For more information about HardCopy III embedded memory resources, refer to the *Mapping Stratix III Device Resources with HardCopy III Devices* chapter.

While all three memory types are dedicated resources in Stratix III devices, only the M9K and M144K memory blocks are dedicated resources in the HardCopy III devices. However, the same functionality of the Stratix III MLABs can be supported in HardCopy III devices. The Quartus II software maps the Stratix III MLAB function to the appropriate HCell macro that preserves the memory function. This allows the HardCopy III core fabric to be used more efficiently, freeing up unused HCells for adaptive logic modules (ALMs) or digital signal processing (DSP) functions.

Although the memory in HardCopy III devices supports the same memory functions and features as Stratix III devices, you cannot pre-load or initialize HardCopy III memory blocks with a Memory Initialization File (**.mif**) when they are used as RAM. Unlike Stratix III devices, HardCopy III devices do not have device configuration. The memory content of HardCopy III devices are random after power-up. Therefore, you must ensure that your Stratix III design does not require a **.mif** if the memory blocks are used as RAM. However, if the HardCopy III memory block is designed as ROM, it powers up with the ROM contents.

Use the ALTMEM_INIT megafunction to initialize the RAM after power-up for HardCopy III devices. This megafunction reads from an internal ROM (inside the megafunction) or an external ROM (on-chip or off-chip) and writes to the RAM after power-up.

When using non-registered output mode for the HardCopy III MLABs, the output powers up with memory content. When using registered output mode for these memory blocks, the outputs are cleared on power-up. You must take this into consideration when designing logic that might evaluate the initial power-up values of the MLAB memory block.

For more information about memory blocks in HardCopy III devices, refer to the *TriMatrix Embedded Memory Blocks in HardCopy III Devices chapter.*

DSP Blocks Implementation

The Quartus II software uses a library of pre-characterized HCell macros to place Stratix III DSP configurations into the HardCopy III HCell-based logic fabric. Depending on the Stratix III DSP configurations, the Quartus II software partitions the DSP function into a combination of DSP HCell macros in the HardCopy III device. This optimizes the DSP function and allows the core fabric to be used more efficiently.

For more information about DSP blocks in HardCopy III devices, refer to the *DSP Blocks Implementation in HardCopy III Devices chapter.*

Clock and PLL Planning

To ensure that you map the Stratix III design to a HardCopy III design successfully, follow these guidelines when implementing your design. They can help make your design robust, ensuring it meets timing closure and achieves the performance you need.

For more information about clock scheme and PLL features in HardCopy III devices, refer to the *Clock Networks and PLLs in HardCopy III Devices chapter.*

Clock Networks and PLL Resources

You must consider the system clocking scheme, timing requirements, and fan-out requirements during clock networks and PLL resources planning. Matching PLL resources between Stratix III and HardCopy III devices is determined by the design's clocking scheme and timing requirements. For high fan-out signals, use a dedicated clock resource.

In the Quartus II software, be sure the HardCopy III companion device is selected in the device selection panel. This ensures that the PLL, other resources used, and the functions implemented in both the Stratix III and HardCopy III designs match. In addition, it ensures that the design converts successfully.

For more information about HardCopy III PLL resources, refer to the Mapping Stratix III Device Resources with HardCopy III Devices chapter.

Add PLL Reconfiguration to Altera IP Blocks

Enable PLL reconfiguration for your design if it uses PLLs. The PLL settings in HardCopy III companion devices may require different settings from the Stratix III PLLs because of different clock tree lengths and PLL compensations. By enabling PLL reconfiguration, you can adjust your PLL settings on the HardCopy III companion device after the silicon has been fabricated. This allows you to fine tune and further optimize your system performance.

For information about implementing PLL reconfiguration in Stratix III FPGAs, refer to *AN 454: Implementing PLL Reconfiguration in Stratix III Devices*.

Use Dedicated Clock Pins

During clock planning, use dedicated clock input pins for high fan-out control signals, such as asynchronous clears, presets, and clock enables for protocol signals, such as `TRDY` and `IRDY` for PCI Express (PIPE), in global or regional clock networks. These dedicated routing networks provide predictable delay and minimize skew for high fan-out signals.

Use dedicated clock pins to drive the PLL reference clock inputs, especially if the design interfaces with external memories. This minimizes the reference clock input jitter to the PLLs, providing more timing margin to make the timing closure successful. For external memory interfaces, Altera recommends using the double date rate (DDR) register in the I/O element to generate the external memory clocks.

For information about external memory interfaces, refer to the *External Memory Interfaces in HardCopy III Devices* chapter.

Quartus II Settings for HardCopy III Devices

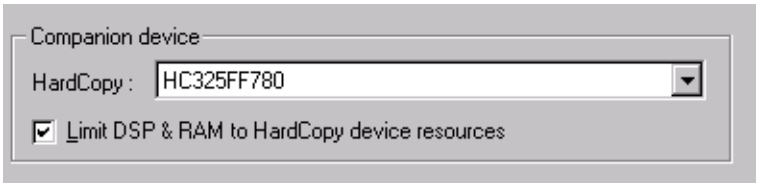
The HardCopy III development flow requires additional Quartus II settings when compared with a typical FPGA-only design flow. This is because the HardCopy III design is implemented in two devices: a Stratix III prototype and a HardCopy III companion device. You must take these settings into consideration when developing your design.

Limit DSP and RAM to HardCopy Device Resources

To maintain compatibility between the Stratix III and HardCopy III devices, your design must use resources that are common to both families. The Quartus II software turns on **Limit DSP & RAM to HardCopy device resources** by default when you select the Stratix III device and HardCopy III companion device in the Quartus II software. This prevents the Quartus II software from using resources in the Stratix III device that are not available in the HardCopy III device.

Figure 1–3 shows the appropriate setting to select in the **Companion device** section.

Figure 1–3. Limit DSP and RAM to HardCopy III Device Resources Checkbox



The Altera HardCopy Design Center requires that your final Stratix III and HardCopy III designs be compiled with the **Limit DSP & RAM to HardCopy device resources** setting turned on before submission to the Altera HardCopy Design Center for back-end implementation.

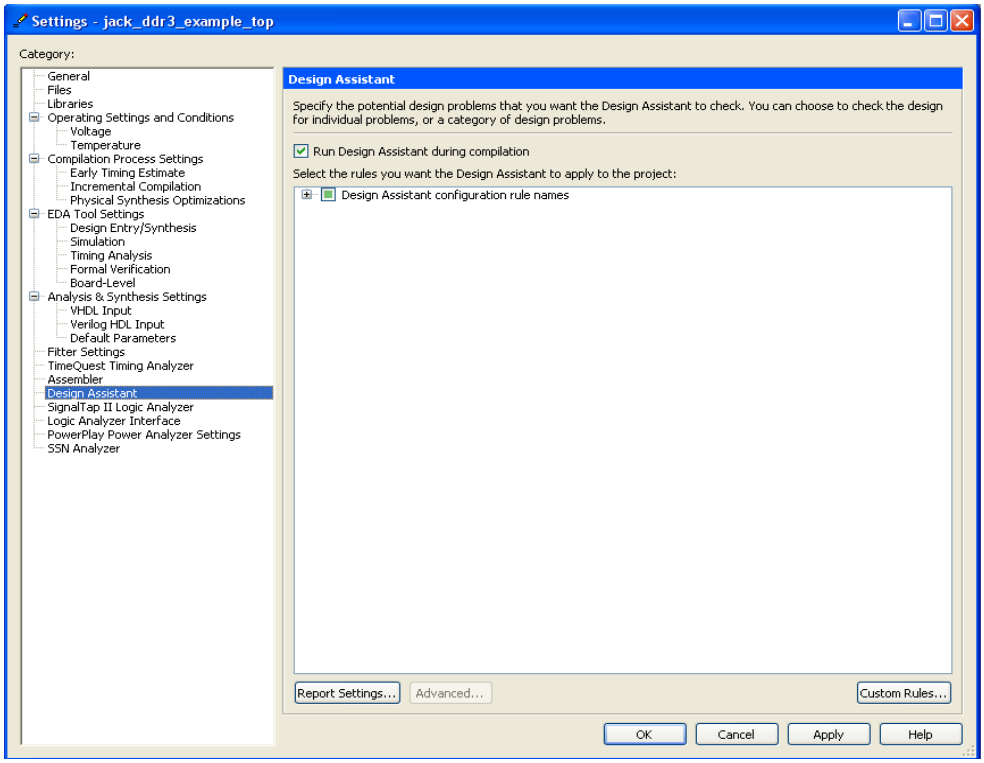
Enable Design Assistant to Run During Compile

You must use the Quartus II Design Assistant to check for design rule violations before submitting the designs to the Altera HardCopy Design Center. Additionally, you must fix all critical and high-level errors reported by the Quartus II Design Assistant. Altera recommends turning on the Design Assistant to run automatically during development.

To enable the Design Assistant to run during compilation, on the **Assignments** menu, click **Settings**. In the **Category** list, select **Design Assistant** and turn on **Run Design Assistant during compilation**.

Figure 1–4 shows the **Design Assistant**.

Figure 1–4. Enabling the Design Assistant



I/O Assignment Settings

Due to the complex rules governing the use of I/O cells and their availability for specific pins and packages, Altera recommends that I/O assignments be completed using the Pin Planner tool and the Assignment Editor in the Quartus II software. These tools ensure that all of the rules regarding each pin and I/O cell are applied correctly. The Quartus II software can export a **.Tcl** script containing all I/O assignments.

For more information about I/O location and type assignments using the Quartus II Assignment Editor and Pin Planner tools, refer to the Assignment Editor chapter in volume 2 of the *Quartus II Handbook*.

To ensure that the HardCopy III mapping is successful, you must make accurate I/O assignments that include pin locations, I/O standards, drive strengths, and capacitance loading for the design. Ensure that the I/O assignments are compatible with all selected devices. Altera recommends not leaving any I/O with an unassigned I/O assignment.

The I/O pins of a Stratix III device and a HardCopy III device are arranged in groups called modular I/O banks. When mapping between a Stratix III device and a HardCopy III device, the I/O pin location must be assigned to the available common I/O banks for both devices. Because HardCopy III devices have fewer I/O banks than Stratix III devices, the Quartus II software limits the I/O banks to only those available in HardCopy III devices.

For more information about I/O banks and pins in HardCopy III devices, refer to the *HardCopy III Device I/O Features* chapter.

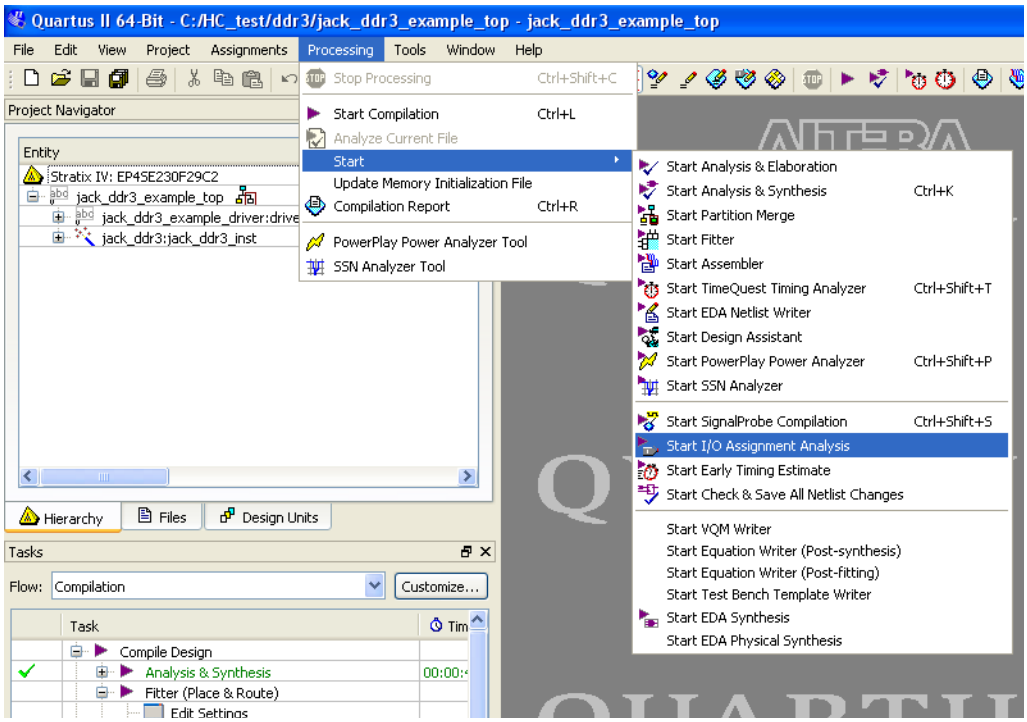
HardCopy III I/O buffers can only support the 3.0 V I/O standard with a maximum V_{CCIO} of 3.0 V. Therefore, when specifying the I/O standard for the Stratix III FPGA device with the HardCopy III companion device already selected, make sure to choose an I/O standard with a V_{CCIO} of 3.0 V or below. Selecting an I/O standard that requires a V_{CCIO} of 3.3 V results in a compilation error.

For more information about HardCopy III *I/O buffers and the standards they support, refer to the DC and Switching Characteristics chapter.*

It is essential to constrain the I/O standards for the design. If you leave the I/O with an unassigned I/O assignment, the Quartus II software assigns the I/O standard to **2.5 V** by default. This standard may not be compatible with your intended I/O standard. To check the supported I/O standards and identify incompatible I/O settings on the assigned I/Os, run the Quartus II I/O Assignment Analysis to verify the I/O settings and assignments. To run I/O Assignment Analysis, on the Processing menu, point to **Start**, then click **Start I/O Assignment Analysis**.

Figure 1–5 shows the I/O assignment analysis in the Quartus II software.

Figure 1–5. Start I/O Assignment Analysis



The default output drive strength in the Quartus II software might not be appropriate for your application. Altera recommends verifying the correct output drive strength for the design. Assigning the right output drive strength improves signal integrity while achieving timing requirements. In addition, the output capacitance loading for both the output and bidirectional pins must be set in the I/O assignment for a successful HardCopy compilation.

Physical Synthesis Optimization Settings

When you develop a HardCopy III device with the Quartus II software, you can target physical synthesis optimizations to the FPGA architecture in Stratix III-device first flow or the HardCopy architecture in the HardCopy III-first flow. The optimizations in the base revision are mapped to the companion device architecture during the mapping process and the post-fitting netlists of both devices are generated and compared. Therefore, you must have the identical physical synthesis settings for both the HardCopy III ASIC and Stratix III FPGA revisions in order to avoid revision comparison failure.

To enable Physical Synthesis Optimizations for the Stratix III FPGA revision of the design, on the **Assignments** menu, click **Settings**. In the **Settings** dialog box, in the **Category** list, expand **Fitter Settings**. These optimizations are passed into the HardCopy III companion revision for placement and timing closure. When designing with a HardCopy III device first, you can enable physical synthesis optimizations for the HardCopy III device, and these post-fit optimizations are passed to the Stratix III FPGA revision.

Beginning with the Quartus II v9.0 software, the **Physical Synthesis Optimizations** settings changed. If a HardCopy III device is set as a companion device, the **Physical Synthesis Optimization** setting in the Stratix III FPGA or HardCopy III ASIC revision supports the **Perform physical synthesis for combination logic** and **Perform register retiming** options. In addition, the effort level of physical synthesis optimization is set to **Fast** by default.

Timing Settings

For HardCopy III device development, you must use the TimeQuest Timing Analyzer. In the Quartus II software, TimeQuest Timing Analyzer is the default timing analyzer for Stratix III and HardCopy III designs. The TimeQuest Timing Analyzer guides the Quartus II Fitter and analyzes timing results during each Stratix III and HardCopy III design compilation.

For information about how to set the Quartus II Fitter to use timing-driven compilation, refer to *"Quartus II Fitter Settings" on page 1–19*.

Timing Constraints for the TimeQuest Timing Analyzer

The TimeQuest Timing Analyzer is a powerful ASIC-style timing analysis tool that validates timing in your design using industry-standard constraint, analysis, and reporting methodology. You can use the TimeQuest analyzer's GUI or command-line interface to constrain, analyze, and report results for all timing paths in your design.

Before running the TimeQuest analyzer, you must specify initial timing constraints that describe the clock characteristics, timing exceptions, signal transition arrival, and required times. You can specify timing constraints in the Synopsys Design Constraints File (**.sdc**) format using the TimeQuest analyzer GUI or the command-line interface. The Quartus II Fitter optimizes the placement of logic to meet your constraints.

The TimeQuest analyzer analyzes the timing paths in the design, calculates the propagation delay along each path, checks for timing constraint violations, and reports timing results as slack in the **Report** and **Console** panels. If the TimeQuest analyzer reports any timing violations, you can customize the reporting to view precise timing information about the specific paths, and then constrain those paths to correct the violations. When your design is free of timing violations, you can be confident that the logic will operate as intended in the target device.

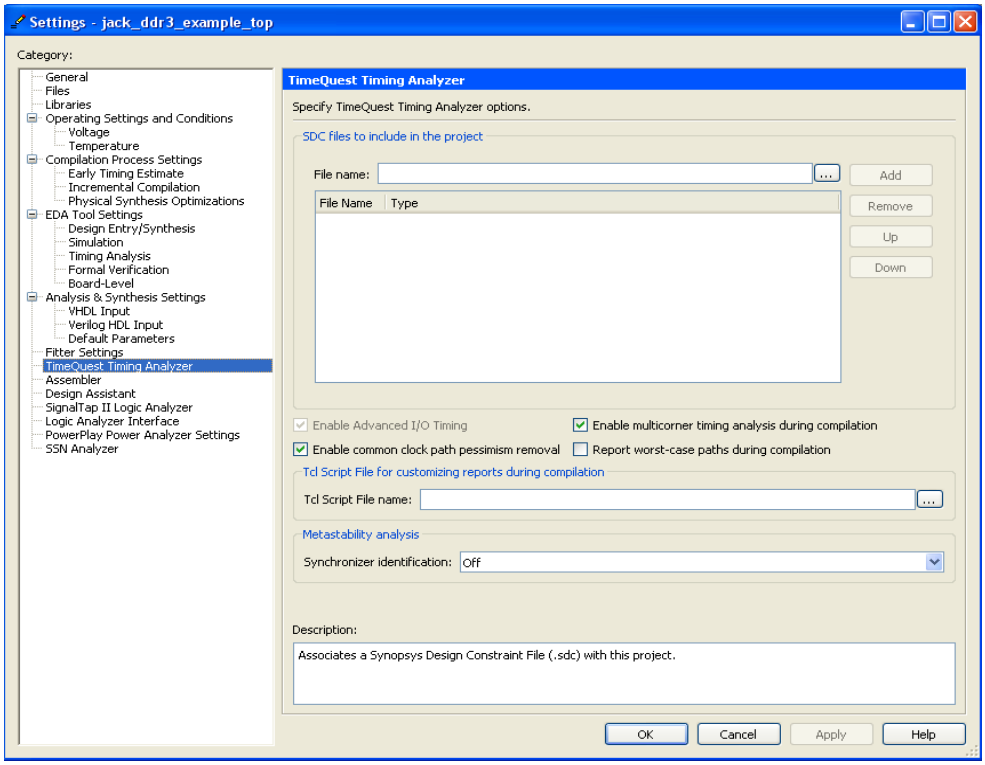
The TimeQuest analyzer is a complete static timing analysis tool that you can use as a sign-off tool for the Stratix III design. For the HardCopy III design, the Altera HardCopy Design Center uses the PrimeTime timing analyzer as the sign-off tool for back-end implementation.

For more information about how to create .sdc format timing constraints, refer to the Quartus II TimeQuest Timing Analyzer chapter in volume 3 of the *Quartus II Handbook*.

TimeQuest Multicorner Timing Analysis Setting

The Altera HardCopy Design Center requires that all HardCopy handoff files include a TimeQuest analyzer timing report for design review. In the TimeQuest analyzer timing report, you must include both fast- and slow-corner timing analysis for setup, hold, and I/O paths. To do this, enable the **Multicorner timing analysis during compilation** option on the **TimeQuest Timing Analyzer** page in the Quartus II software. This option directs the TimeQuest analyzer to analyze the design and generate slack reports for the slow and fast corners. Figure 1–6 shows the settings you must enable so that the TimeQuest analyzer generates the appropriate reports.

Figure 1–6. TimeQuest Multicorner Timing Analysis Setting



Incremental Compilation

For the HardCopy development flow, the Quartus II design software offers incremental compilation to preserve the compilation results for unchanged logic in your design. This feature dramatically reduces your design iteration time by focusing new compilations only on changed design partitions. New compilation results are then merged with the previous compilation results from unchanged design partitions.

There are two approaches of incremental compilation in the Quartus II software:

- Top-down incremental compilation
- Bottom-up incremental compilation

Bottom-up incremental compilation flow is not supported for designs targeting HardCopy ASICs.

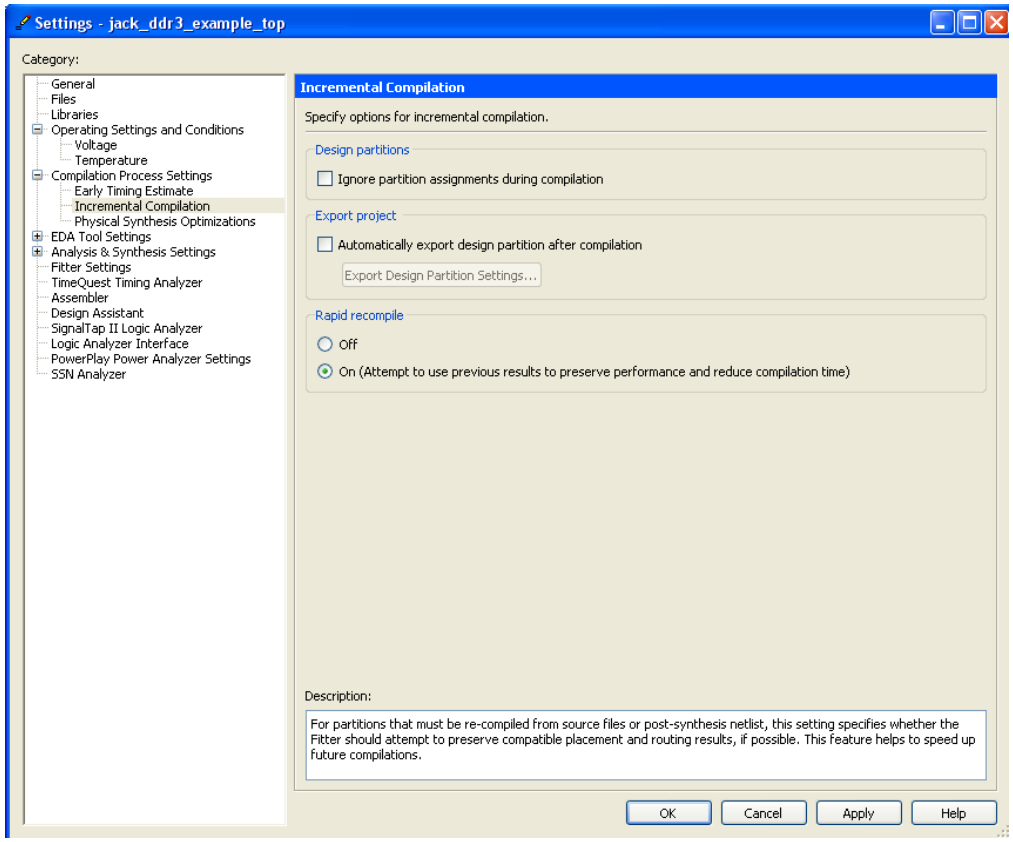
For large, high-density and high-performance designs in Stratix FPGAs and HardCopy ASICs, use top-down incremental compilation. Top-down incremental compilation facilitates team-based design environments, allowing designers to create and optimize design blocks independently. Begin planning for incremental compilation from the start of your design development. To take advantage of incremental compilation flow, split the design along any of its hierarchical boundaries into blocks called design partitions.

In the Quartus II software, the same procedures create design partitions in the HardCopy ASIC and Stratix FPGA revisions.

For more information about creating design partitions, refer to the *Quartus II Incremental Compilation for Hierarchical and Team-Based Design* chapter in volume 1 of the *Quartus II Handbook*.

In the Quartus II software, the full Incremental Compilation option is on by default, so the project is ready for you to create design partitions for incremental compilation. Figure 1–7 shows the full **Incremental Compilation** page in the Quartus II software.

Figure 1–7. Quartus II Incremental Compilation Option



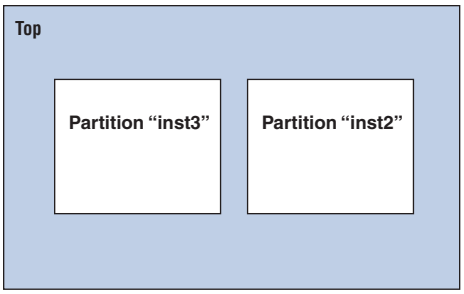
If you do not create design partitions in a design, the Quartus II software uses a flat compilation flow, and you cannot use Incremental Compilation.

Top-Down Incremental Compilation

Top-down incremental compilation is supported for the base revision for designs targeting HardCopy ASICs in both the FPGA first flow and HardCopy first flow. In the Quartus II design software, you must select the base and companion revisions before design partitions of the base revision are created.

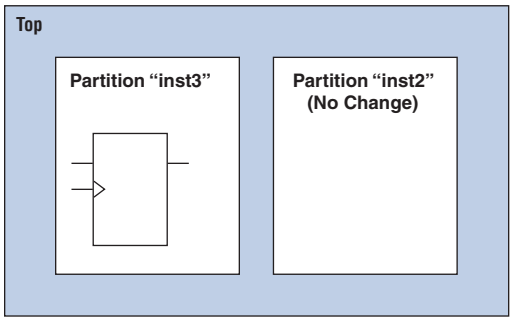
After the design partitions are created in the Quartus II design software, the base revision is compiled and the design partition assignments are mapped to the companion device. In the HardCopy development flow, you make changes only in the base revision's design and design partition assignments with the Quartus II design software. Therefore, you can perform top-down incremental compilation only in the base revision, but cannot perform incremental compilation to the companion revision. Figure 1–8 shows the design "Top" with two design partitions.

Figure 1–8. Quartus II Project with Design Partitions



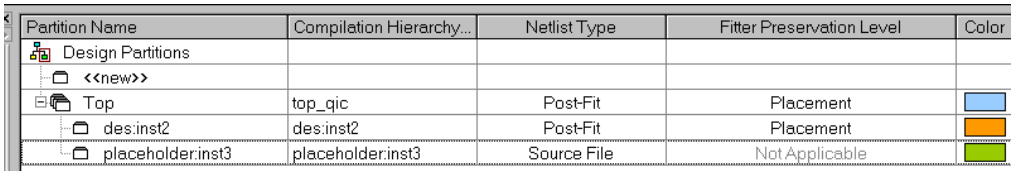
After the design partitions are created and compiled in the base family revision, you can modify the specific design partitions for additional area and performance improvement. Figure 1–9 shows that when the logic is modified in partition "inst3" the Quartus II software is ready to re-compile the individual hierarchical design partition separately, based on the preservation level in the Design Partition Window. Therefore, the optimization result of design partitions "top" and "inst2" are preserved while the partition "inst3" is re-compiled in the Quartus II design software.

Figure 1–9. Design Partition Modified



Before recompiling the design, you can set the Netlist Type in the Design Partition Window to **Source File**, **Post-Synthesis**, or **Post-Fitting** to preserve the Netlist Type of each design partition. Figure 1–10 shows the Design Partition Window with the Post-Fit preservation level for the design partitions "top" and "inst2." This allows the Quartus II design software to re-compile the design partition "inst2" from the source file, but preserves the post-fitting results of design partitions "top" and "inst3."

Figure 1–10. Design Partition Window



The Quartus II software merges the new compiled design partition "inst3" into a complete netlist for subsequent stages of the compilation flow. Design partitions "top" and "inst2" in the design do not perform incremental compilation because their logic must be preserved. Therefore, the compilation time for the overall design is reduced.

Top-Down Incremental Compilation with Empty Design Partitions

Bottom-up incremental compilation is not supported in the HardCopy development flow. During the initial stage of the design cycle, part of the design may be incomplete or developed by a different designer or IP provider. However, you can create an empty partition for this part of the design while compiling the completed partitions, and then save the results for the complete partitions while you optimize the imported part of the design.

You can often use a top-down flow with empty partitions to implement behavior similar to a bottom-up flow, as long as you do not change the global assignments between compilations. All global assignments must be the same for all compiled partitions, so the assignments can be reproduced in the companion device after mapping.

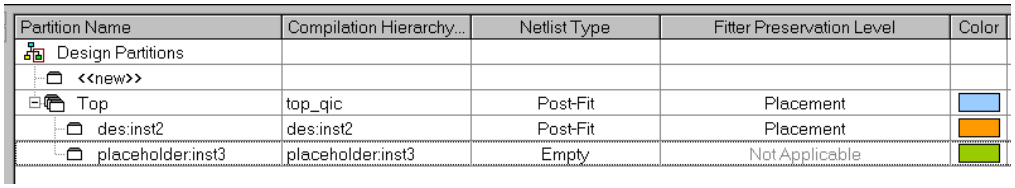
For the HardCopy development flow, create an empty partition in the base device because it cannot be created in the companion revision.

Creating an empty partition in a design is similar to creating a regular design partition in the Quartus II software. When the logic within a specific design partition is incomplete, use the following instructions to set the **Netlist Type** to **Empty**.

1. On the **Assignment** menu, click **Design Partitions Window**.
2. Double-click an entry under the **Netlist Type** column and select **Empty**.

This setting specifies that the Quartus II Compiler must use an empty placeholder netlist for the partition. Figure 1–11 shows the Empty partition setting in the Design Partition window.

Figure 1–11. Design Partition with Empty Netlist Type



When a partition Netlist Type is defined as **Empty**, virtual pins are automatically created at the boundary of the partition. This means that the software temporarily maps the I/O pins in the lower-level design entity to the internal cells instead of the pins during compilation. Any child partitions below an empty partition in the design hierarchy are also automatically treated as empty, regardless of their settings.

After the design partition with the empty Netlist Type is completed and you have defined "inst3" in the top module, the Quartus II software is ready to recompile the "inst3" design partition.

First, set the Netlist Type of the design partition "inst2" to the specific preservation target, such as **Post-Synthesis** or **Post-Fit** to preserve the performance results from the previous compilation. Before you recompile the design, ensure that you set the Netlist Type of the design partition "inst3" to **Source File** because this is new design source for the Quartus II software.

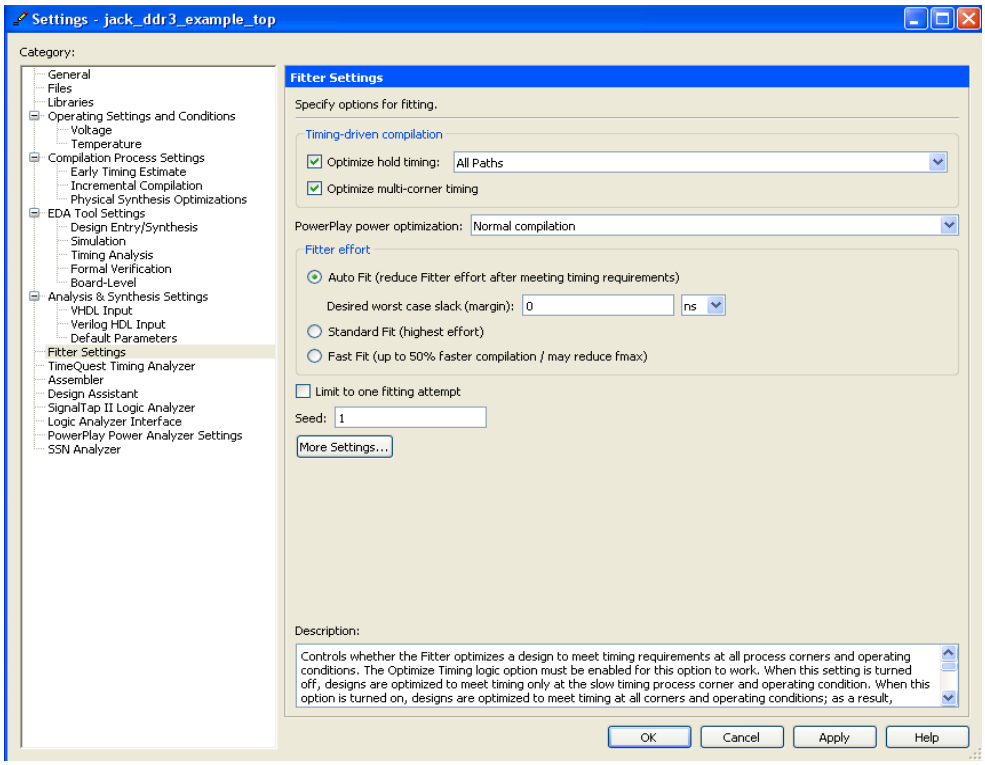
As in the traditional top-down design flow, the Quartus II software merges the new compiled design partition "inst3" into a complete netlist for the subsequent stages of the compilation flow. Therefore, the turnaround time for the design compilation is reduced.

Quartus II Fitter Settings

To make the HardCopy III device implementation more robust across process, temperature, and voltage variations, the Altera HardCopy Design Center requires that you enable **Multicorner Optimization** for the Quartus II Fitter. This setting controls whether the Fitter optimizes a design to meet timing requirements at the fast-timing process corner and operating condition, as well as at the slow-timing process corner and operating condition. The Altera HardCopy Design Center also requires that you enable the **Optimize hold timing** setting for the Quartus II Fitter. This setting allows the Fitter to optimize hold time by adding delay to the appropriate paths.

Figure 1–12 shows the **Optimize multi-corner timing** and **Optimize hold timing** settings in the **Fitter Settings** panel.

Figure 1–12. Quartus II Fitter Settings for Optimization Multicorner and Hold Time Fix

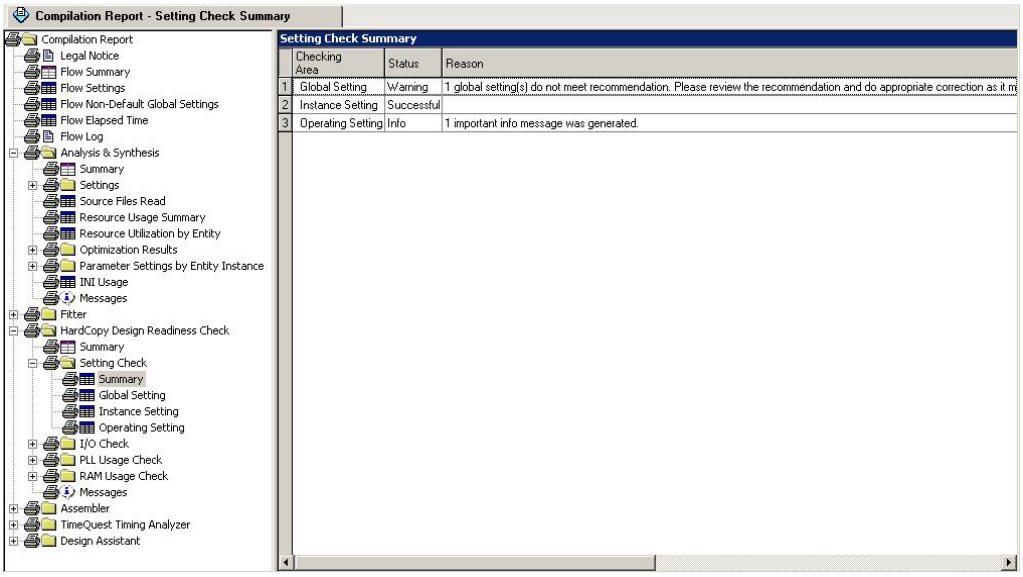


HardCopy Design Readiness Check

The HardCopy Design Readiness Check (HCDRC) feature checks issues that must be addressed before handing off the HardCopy III design to the Altera HardCopy Design Center for the back-end implementation process. In the Quartus II software, the HCDRC includes logic checks such as PLL, RAM, and setting checks (Global Setting, Instance Setting, and Operating Setting) that were previously done in the HardCopy hand-off report. Beginning with the Quartus II software version 8.0, the default setting for running HCDRC is **On**. You can run HCDRC at post-Fitter either turned on through the Quartus Settings File (**.qsf**) or GUI.

Figure 1–13 shows the HardCopy Design Readiness Check in the Quartus II software.

Figure 1–13. HardCopy Design Readiness Check in the Quartus II Software



For more information about the HardCopy Design Readiness Check, refer to the Quartus II Support for HardCopy Series Devices chapter in volume 1 of the *Quartus II Handbook*.

Timing Closure and Verification

After compiling the project for the Stratix III and HardCopy III designs, check the device used and verify that the design meets your timing requirements. Analyze the messages generated by the Quartus II software during compilation to check for any potential problems. Also verify the design functionality between the Stratix III and HardCopy III devices with the **HardCopy Companion Revision Comparison** option in the Quartus II software.

Timing Closure with the TimeQuest Timing Analyzer

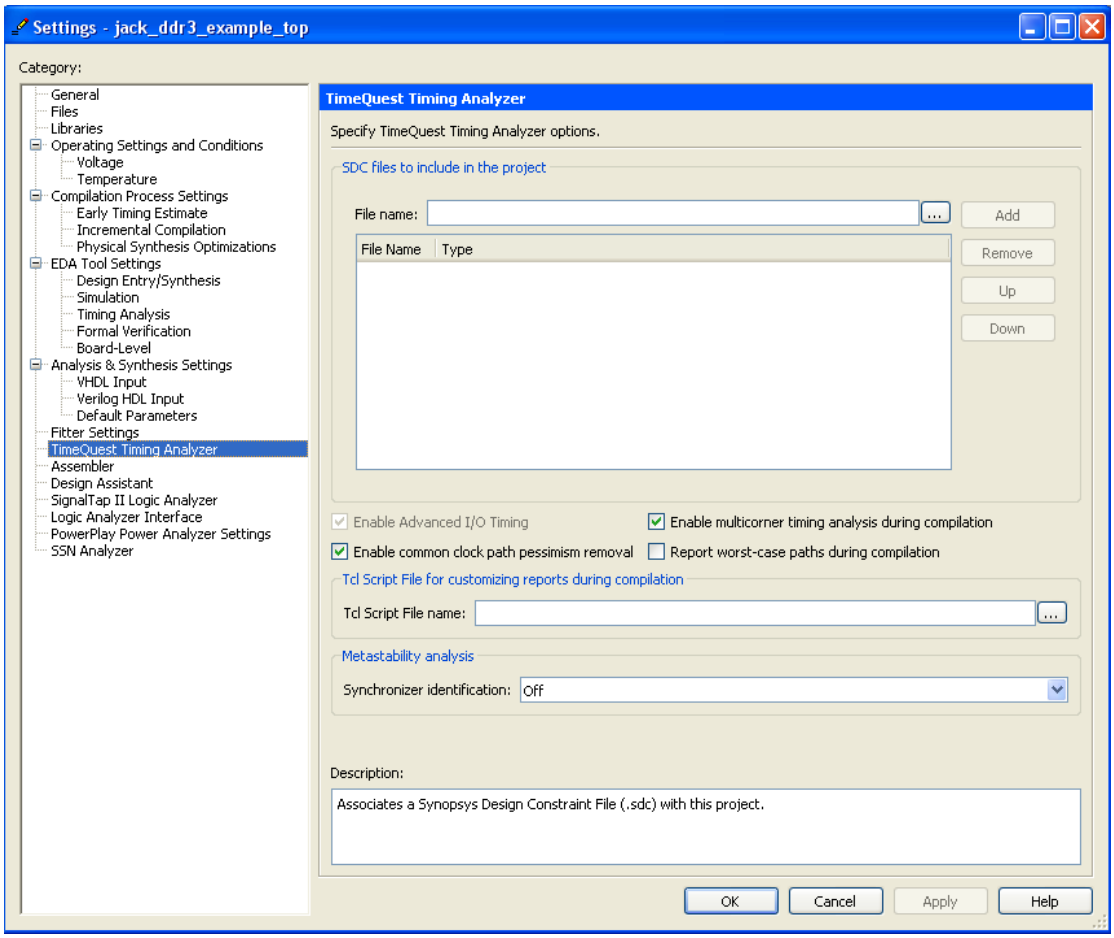
The TimeQuest Timing Analyzer is the timing analysis tool for all HardCopy III devices during the front-end design process; it is the default timing analyzer for Stratix III and HardCopy III devices in the Quartus II software.

After you specify the initial timing constraints that describe the clock characteristics, timing exceptions, and signal transition arrival and required time in the **.sdc**, the TimeQuest analyzer analyzes the timing paths in the design, calculates the propagation delay along each path, checks for timing constraint violations, and reports timing results.

From the TimeQuest analyzer settings in the Quartus II software, ensure that the TimeQuest analyzer has the **Enable multicorner timing analysis during compilation** check box selected. This setting is necessary to achieve timing closure for the HardCopy III ASIC design. By default, the TimeQuest analyzer has this setting enabled to analyze the design against best-case and worst-case operating conditions during compilation (Figure 1–14).

To direct the TimeQuest analyzer to remove the common clock path pessimism during slack computation in the Quartus II software, select the **Enable common clock path pessimism removal** option in the **TimeQuest Timing Analyzer** page (Figure 1–14).

Figure 1–14. TimeQuest Timing Analyzer Enable Multicorner Timing Analysis During Compilation and Enable Common Clock Path Pessimism Removal Options

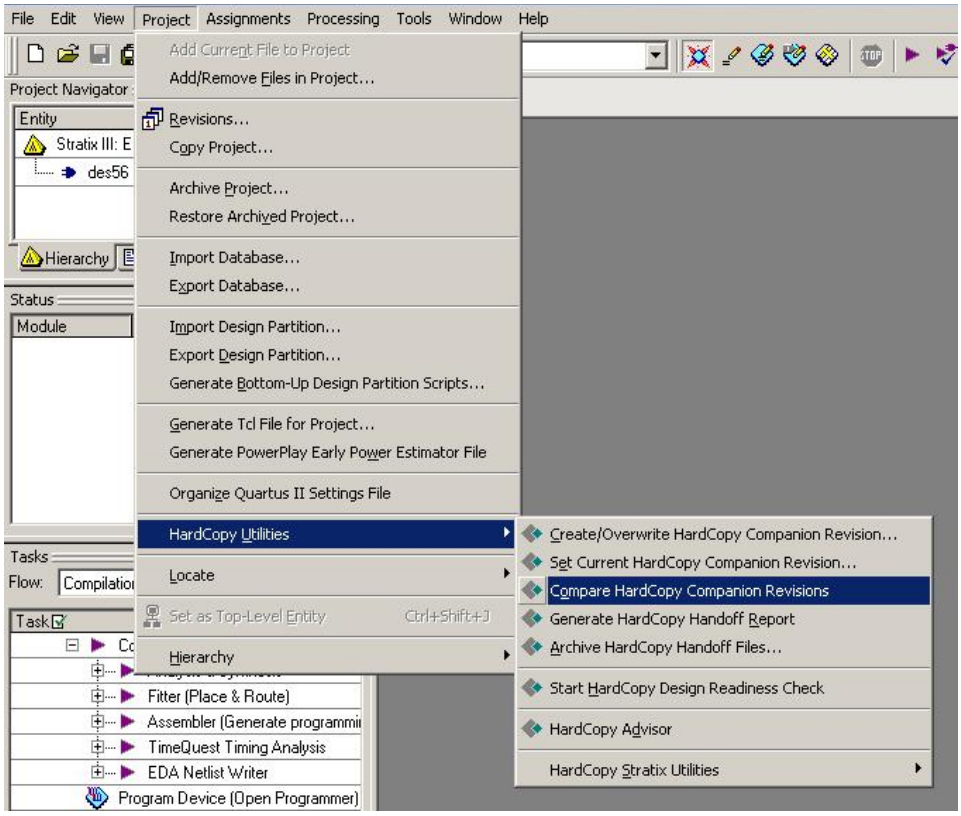


Verification

The Quartus II software uses companion revisions in a single project to promote conversion of your design from a Stratix III FPGA to a HardCopy III ASIC. This methodology allows you to design with one set of register transfer level (RTL) code to be used in both the Stratix III and HardCopy III designs, guaranteeing functional equivalency.

When making changes to your design in a companion revision, use the **Compare HardCopy Companion Revisions** feature in the Quartus II software to ensure that your Stratix III and HardCopy III designs match functionality and compilation settings. You must perform this comparison after both Stratix III and HardCopy III designs are compiled and before you hand off the design to the Altera HardCopy Design Center. Figure 1–15 shows how to navigate to the companion revisions comparison. On the Project menu, point to **HardCopy Utilities** and click **Compare HardCopy Companion Revisions**.

Figure 1–15. Compare HardCopy Companion Revisions



Engineering Change Order (ECO)

During the last stage of the design cycle, it is critical to implement a specific portion of the design, without affecting the rest of its logic. As described in the previous section, Incremental Compilation can implement and manage certain partitions of the design, and preserve the optimization results for the rest of the design. However, this becomes difficult to manage because Engineering Change Orders (ECOs) are often implemented as last-minute changes to your design.

The Quartus II software provides the Chip Planner tool and the Resource Property Editor for ECO operations to shorten the design cycle time significantly. For the HardCopy development flow, ECOs occur in the Stratix FPGA revision and you make the changes directly to the post place-and-route netlist. When you switch to the HardCopy ASIC revision, apply the same ECOs, run the timing analysis and assembler, perform a revision compare, and then run HardCopy Netlist Writer for design submission.

ECOs can be categorized in two ways for HardCopy development:

- Migrating one-to-one changes
- Migrating changes that must be implemented differently

Migrating One-to-One Changes

Some examples of migrating one-to-one changes are changes such as creating, deleting, or moving pins, changing pin or PLL properties, or changing pin connectivity (provided the source and destination of the connectivity changes are I/Os or PLLs).

To duplicate the same ECO in the Quartus II software, use the Change Manager and record all ECOs for the FPGA revision. Ensure that the same ECO operations occur on each revision for both the Stratix FPGA and HardCopy ASIC revisions to avoid a revision comparison failure.

To generate a **.tcl** script of the ECO operations in the Stratix FPGA revision and apply it to the HardCopy revision, follow these steps:

1. In the Stratix FPGA revision, open **Change Manager**.
2. On the **View** menu, click **Utility Windows** and select **Change Manager**.
3. Perform the ECO in the **Chip Planner** or **Resource Property Editor**. You will see the ECO operations in the **Change Manager**.

For the information about ECO operations in the Stratix FPGA revision, refer to the *Engineering Change Management with the Chip Planner* chapter in volume 2 of the *Quartus II HandBook*.

4. Export the ECO operations from the **Change Manager** to **Tcl Script**. On the **Change Manager**, right mouse click the entry. Click **Export**, and then click **Export All Changes AS…**
5. Save the **.tcl** script, to be used in the HardCopy revision.

In the HardCopy revision, apply the **.tcl** script to the companion revision using the following procedure.

1. Open the generated **.tcl** script from the Quartus II software or a text editor tool. Edit the line `project_open` <*project*> `- revision` <*revision*> to refer to the appropriate companion revision. Save the **.tcl** script.
2. Apply the **.tcl** script to the companion revision. On **Tools** menu, scroll the **Tcl Scripts** pull-down menu, and select **ECO Tcl** and click **Run**.

Migrating Changes that Must be Implemented Differently

Unlike migrating changes one-to-one, some changes must be implemented on the Stratix FPGA and HardCopy ASIC revisions differently. Changes affecting the logic of the design can fall into this category. Examples of these are LUTMASK changes, LC_COMB/HSADDER creation and deletion, and connectivity changes not covered in the previous section.

For a summary of suggested implementation changes, refer to the *Quartus II Support for HardCopy Series Devices* chapter in volume 1 of the *Quartus II Handbook*.

PLL settings are another example of implementing changes differently on the Stratix FPGA and the HardCopy ASIC revisions. Sometimes PLL-generated clocks must be modified to provide a higher-frequency clock in HardCopy devices to improve the performance of the HardCopy device without changing the performance of the Stratix FPGA. You must handle the modification correctly so that the HardCopy Companion Revision Comparison utility does not generate critical errors.

To set different PLL settings for the Stratix FPGA and HardCopy revisions, you must have different PLL source files. Each PLL must have the same module name, so that the same PLL in both revisions is not treated differently during the HardCopy Revision Comparison stage. You must have two PLL files with different names that reference the same module.

When starting HardCopy development with the FPGA first flow to override the file naming convention so that the same PLL module can be referenced by two different PLL files, complete the following steps:

1. Specify the PLL source file in the QSF file in the Stratix FPGA revision. For example, `set_global_assignment -name MIGRATION_DIFFERENT_SOURCE_FILE pll_fgpa.v`

   Note that when there are multiple PLL source files, you must use multiple assignments to specify the PLL source files.

2. Compile the Stratix FPGA revision in the Quartus II software.
3. After creating the HardCopy revision, modify the PLL source file manually or with the MegaWizard Plug-In Manager in order to improve the performance in the HardCopy revision.

   When the MegaWizard Plug-In Manager updates the new source file, it modifies the top-level name of the module or entity in the source file to match the name of the source file. Therefore, you must rename the module or entity after you have updated the file with the MegaWizard Plug-In Manager so that your top-level design instantiates the PLL with the newly modified PLL design file.

4. After updating the PLL source file in the HardCopy revision, verify that the QSF source file setting contains the newly modified PLL source file. For example, `set_global_assignment -name VERILOG_FILE pll_hc.v`
5. Compile the HardCopy revision in the Quartus II software.

After compilation is completed, run the **HardCopy companion Revision Comparison** utility to observe and track the changes made to the PLLs and design settings. These changes are captured as critical warnings in the revision comparison report and must be reviewed by the HardCopy Design Center before the design is accepted for mapping.

HardCopy III Handoff Process

To submit a design to the Altera HardCopy Design Center for design review and back-end implementation, generate a HardCopy III handoff report and archive the HardCopy III project.

Before you generate the HardCopy III handoff report, you must first successfully perform the following tasks:

- Compile both Stratix III and HardCopy III revisions of the design.
- Run the **Compare HardCopy Companion Revision** utility.

Archive the HardCopy III project and submit it to the Altera HardCopy Design Center for back-end implementation. This is the last step in the HardCopy III design flow. The HardCopy III archive utility creates a different Quartus II Archive File (**.qar**) than the standard Quartus II project archive utility generates. This archive contains only the data from the Quartus II project needed to implement the design in the Altera HardCopy Design Center.

To use the **Archive HardCopy Handoff Files** utility, you must perform all tasks for generating a HardCopy III handoff report.

After you generate a HardCopy III handoff report, select the handoff option. On the Project menu, point to **HardCopy Utilities** and click **Archive HardCopy Handoff Files**.

The **Archive HardCopy Handoff Files** utility archives your design, settings, results, and database files for delivery to Altera. These files are generated at the same directory level as the targeted project created with an **_hc** extension.

Document Revision History

Table 1–2 lists the revision history for this chapter.

Table 1–2. Document Revision History

Date	Version	Changes
March 2012	3.3	Updated Table 1–1: removed device HC315.
January 2011	3.2	Updated for Quartus II software version 10.1.
January 2010	3.1	■ Added “Physical Synthesis Optimization Settings” on page 1–13. ■ Added “Incremental Compilation” on page 1–15. ■ Added “Engineering Change Order (ECO)” on page 1–23.
June 2009	3.0	■ Updated Table 1–1. ■ Removed Tables 1-2 and 1-3. ■ Minor text edits.
May 2009	2.1	■ Updated “Introduction” and “I/O Pin and Package Offering.” ■ Updated Table 1–2, Table 1–4, and Table 1–6. ■ Updated Figure 1–1, Figure 1–2, Figure 1–4, Figure 1–6, Figure 1–7, and Figure 1–9. ■ Removed the Reference Documents and Conclusion sections. ■ This chapter was listed as p/n 52001 in release version 2.0.
December 2008	2.0	Minor text edits.
May 2008	1.0	Initial release.