3. Designing HardCopy Series Devices

QII51004-11.0.1

This chapter describes the flow for designing HardCopy® series devices in the Quartus® II software.

Altera® HardCopy ASICs are the lowest risk, lowest total cost ASICs. The HardCopy system development methodology offers fast time-to-market, low risk, and with the Quartus II software, you can design with one set of RTL code and one set of IP for both FPGA and ASIC implementations. This flow allows you to conduct true hardware/software co-design and completely prepare your system for production prior to ASIC design hand-off. Altera provides a turn-key process to convert your design to a HardCopy ASIC for production.

In this chapter, the term FPGA refers to a Stratix® II, Stratix III, or Stratix IV device, which is the prototype device for a HardCopy II, HardCopy III, or HardCopy IV device, respectively.

This chapter discusses the following topics:

For more information about HardCopy series devices, refer to the respective HardCopy device handbook, which is available on the Literature page of the Altera website at www.altera.com.

HardCopy Series Design Benefits

Designing with HardCopy devices offers the following substantial benefits:

- Seamless prototyping using an FPGA for at-speed system verification and system development, which reduces total project development time and cost
- Dependable conversion from an FPGA prototype to a HardCopy ASIC expands product planning options

- Unified design methodology for FPGA and HardCopy designs reduces the need for ASIC development software and two sets of intellectual property, which reduces project risk
- System development methodology delivers lowest total cost

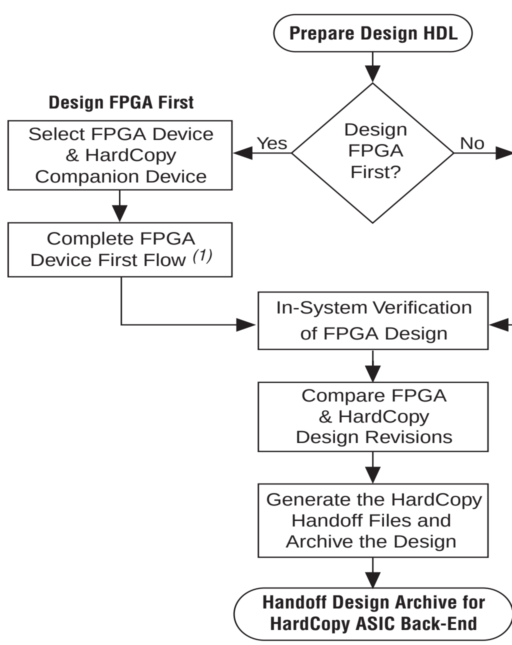
HardCopy Development Flow

In the HardCopy development flow, you design a FPGA and a HardCopy companion device together in one Quartus II project using one of the following design flows:

- FPGA first flow—Design the FPGA first for in-system verification, and then create a HardCopy companion device second. Performing system verification early helps reduce overall total project development time. The FPGA first flow is the default flow and the rest of this chapter is based on this flow.
- HardCopy first flow—Design the HardCopy device first, and then create the FPGA companion device second for in-system verification. This method more accurately predicts the maximum performance of the HardCopy device during development. If you optimize your design to maximize HardCopy performance, but cannot meet your performance requirements with the FPGA, you can still map your design with decreased performance requirements for in-system verification.

These two flows are illustrated at a high level in Figure 3–1.

Figure 3–1. HardCopy Flow in Quartus II Software



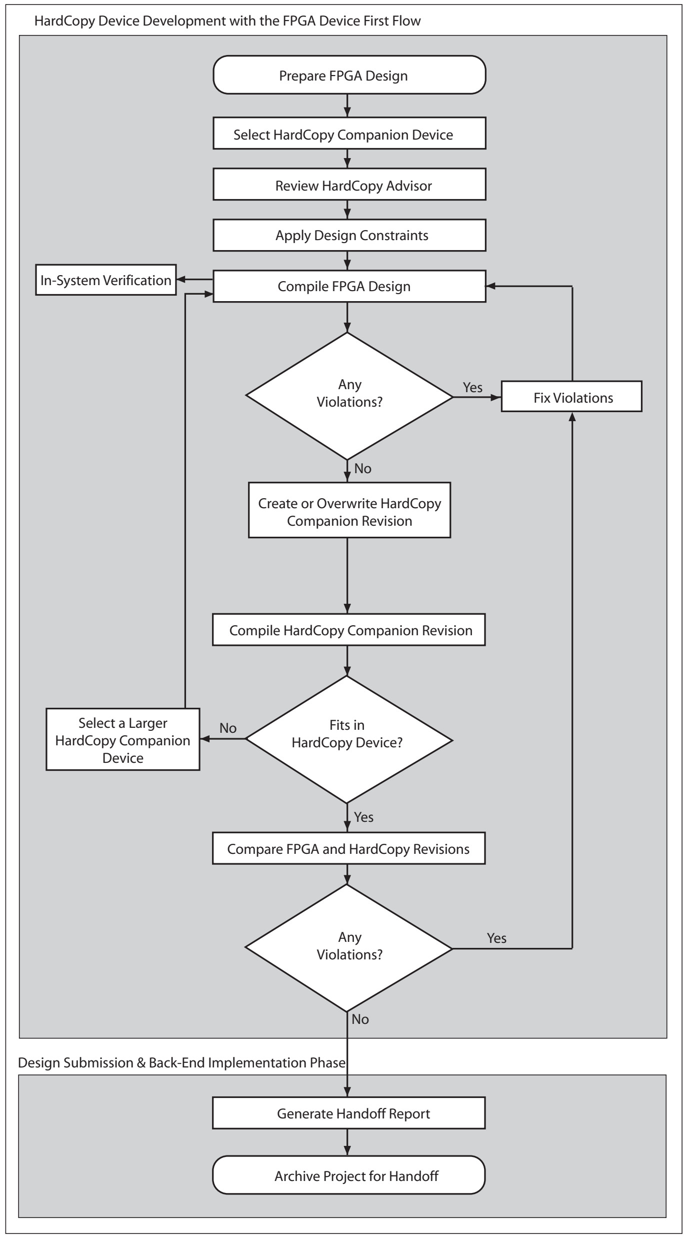
Designing the FPGA First

The FPGA first flow begins with seamless FPGA prototyping and is similar to the traditional FPGA design flow, but requires you to perform additional tasks necessary to convert the design to the HardCopy companion device within the same project. The following steps provide an overview of the tasks necessary in the FPGA first flow:

1. Specify an FPGA device for prototyping and a HardCopy companion device. Refer to "Selecting the Prototype and Companion Devices" on page 3–7 for more information.
2. Apply design constraints. Refer to "Applying Design Constraints" on page 3–10 for more information.
3. Compile the FPGA design, and then create and compile the HardCopy companion revision. Refer to "Compiling the Design and Creating Companion Revisions" on page 3–15 for more information.
4. Compare the HardCopy companion revision and FPGA device compilations. Refer to "Comparing HardCopy and FPGA Companion Revisions" on page 3–21 for more information.
5. Generate the handoff files, reports, and archive, and arrange for its submission to the Altera HardCopy Design Center for back-end implementation. Refer to "Preparing the Design for Handoff" on page 3–27 for more information.

Figure 3–2 provides an overview of the FPGA first flow.

Figure 3–2. Designing the FPGA First Flow

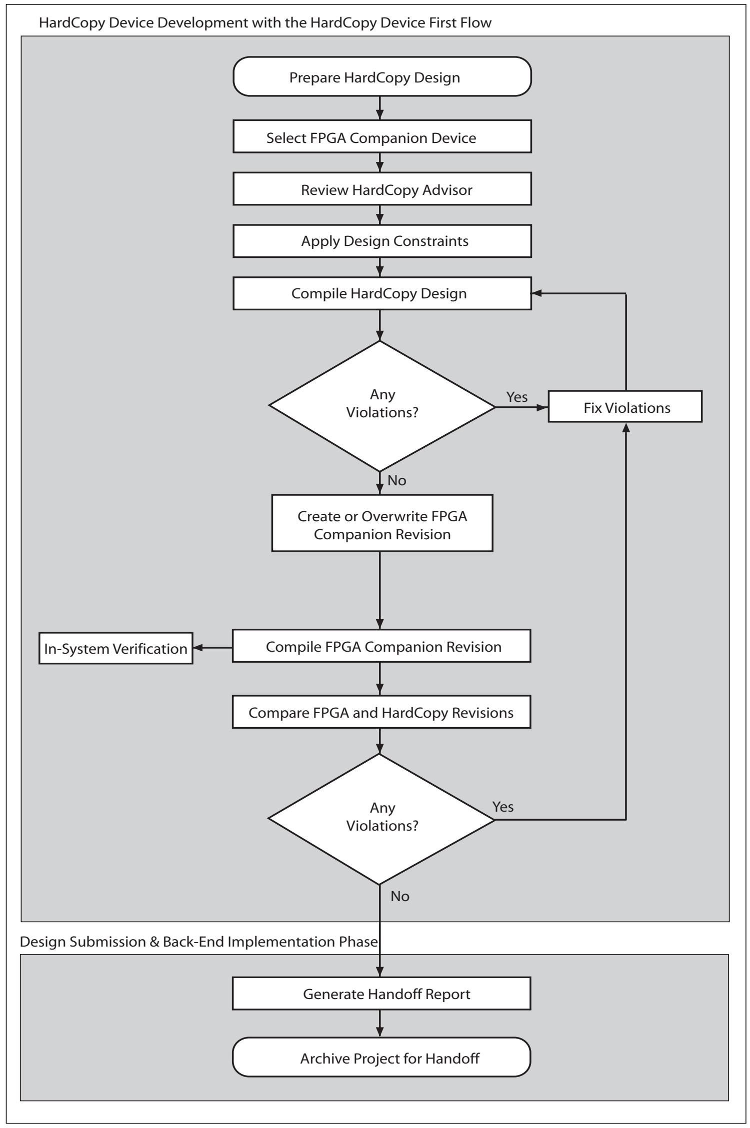


Designing the HardCopy Device First

The HardCopy device first flow in the Quartus II software allows you to maximize performance in the HardCopy device and map the design to the FPGA prototype for in-system verification. This approach is preferred if you use the HardCopy device to achieve higher performance than the FPGA prototype, because you can see your potential maximum performance in the HardCopy device immediately during development, and you can create a slower performing FPGA prototype of the design for in-system verification. This design process is similar to the FPGA first flow development flow, but you begin the design with a different initial device family instead. The remaining tasks to complete your design for both the FPGA and HardCopy devices essentially follow the same process.

The complete HardCopy first flow is shown in Figure 3–3.

Figure 3–3. Designing the HardCopy First Flow



HardCopy Advisor

The HardCopy Advisor provides an interactive list of tasks to help you through the development of your FPGA prototype and HardCopy design. The following tasks highlight the checkpoints that the HardCopy Advisor reviews, which includes the major checkpoints in the design process, but not every step in the process.

1. Select an FPGA device.
2. Select a HardCopy companion device.
3. Set up the FPGA revision.
4. Confirm FPGA junction temperature range settings (HardCopy III and HardCopy IV devices only).
5. Compile and check the FPGA design.
6. Create or overwrite the HardCopy companion revision.
7. Confirm HardCopy junction temperature range settings (HardCopy III and HardCopy IV devices only).
8. Compile and check the HardCopy companion results.
9. Compare companion revisions.
10. Generate a HardCopy Handoff report.
11. Archive handoff files and send them to Altera.

For more information about the HardCopy Advisor in the Quartus II software, refer to *About the HardCopy Advisor* in Quartus II Help.

HardCopy Utilities

The HardCopy Utilities menu contains the main functions you use to develop your HardCopy design and FPGA prototype companion revision. To access this menu in the Quartus II software, on the Project menu, click **HardCopy Utilities**. Each HardCopy Utilities menu feature is summarized in Table 3–1.

Table 3–1. HardCopy Utilities Menu Options (Part 1 of 2)

Option	Description	Applicable Design Revision	Restrictions
Create/Overwrite HardCopy Companion Revision	Creates a new companion revision or overwrites an existing companion revision for your FPGA and HardCopy design	FPGA prototype design and HardCopy companion revision	■ The **Auto device selected by the Fitter** option must be turned off ■ An FPGA device and a HardCopy companion device must be set
Set Current HardCopy Companion Revision	Specifies the companion revision to associate with the current design revision	FPGA prototype design and HardCopy companion revision	A companion revision must already exist
Compare HardCopy Companion Revisions	Compares the FPGA design revision with the HardCopy companion design revision and generates a report	FPGA prototype design and HardCopy companion revision	Both revisions must be compiled
Generate HardCopy Handoff Report	Generates a report containing important design information files and messages generated by the Quartus II Compiler	FPGA prototype design and HardCopy companion revision	■ Both revisions must be compiled ■ The **Compare HardCopy Companion Revisions** command must be successfully run

Table 3–1. HardCopy Utilities Menu Options (Part 2 of 2)

Option	Description	Applicable Design Revision	Restrictions
Archive HardCopy Handoff Files	Generates a Quartus II Archive File (**.qar**) specifically for submitting the design to the Altera HardCopy Design Center	HardCopy companion revision	■ Both revisions must be compiled ■ The **Compare HardCopy Companion Revisions** command must be successfully run ■ The **Generate HardCopy Handoff Report** command must be successfully run
Start HardCopy Design Readiness Check	Generates a report with the design's settings, I/O check, PLL, and RAM usage checks	FPGA prototype design and HardCopy companion revision	None
HardCopy Advisor	Opens the HardCopy Advisor, which guides you through the process of creating a HardCopy project	FPGA prototype design and HardCopy companion revision	None

Selecting the Prototype and Companion Devices

For both HardCopy device and FPGA prototype planning, the first stage is to choose the device family, device density, speed grade, and package that best suits your design needs. You can select the appropriate companion device based on the device that you select for prototyping.

You can use the HardCopy Device Resource Guide to help you select the appropriate companion device, as described in "HardCopy Device Resource Guide".

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 the respective HardCopy device handbook, which is available on the Literature page of the Altera website at www.altera.com.

HardCopy Device Resource Guide

The HardCopy Device Resource Guide compares the resources required to successfully compile a design with the resources available in the various HardCopy devices. The report rates each HardCopy device and each device resource according to how well it fits the design. The Quartus II software generates the HardCopy Device Resource Guide for all designs successfully compiled for FPGA devices. You can find this guide in the Fitter folder of the Compilation report. Table 3–2 describes the color codes used in the guide.

Table 3–2. HardCopy Device Resource Guide Color Legend

Color	Package Resource (1)	Device Resources
Green (High)	The design can map to the HardCopy package and has been fitted with target device migration enabled in the **HardCopy Companion Device** dialog box.	The resource quantity is within the range of the HardCopy device and the design can likely map if all other resources also fit. You still must compile the HardCopy revision to ensure the design is able to route and close timing.
Orange (Medium)	The design can map to the HardCopy package; however, the design has not been fitted with the target device migration enabled in the **HardCopy Companion Device** dialog box.	The resource quantity is within the range of the HardCopy device; however, the resource is at risk of exceeding the range for the HardCopy package. Compile your design targeting the HardCopy device as soon as possible to check if the design fits and is able to route and migrate all other resources. You might have to select a larger device.
Red (None)	The design cannot map to the HardCopy package.	The resource quantity exceeds the range of the HardCopy device. The design cannot migrate to this HardCopy device.

Note to Table 3–2:

(1) The package resource is constrained by the FPGA for which the design was compiled. Only vertical migration devices within the same package are able to migrate to HardCopy devices.

Use this report to identify potential HardCopy device candidates for your design. The HardCopy and FPGA device packages must be compatible. A logic resource usage greater than 100% or a ratio greater than 1:1 in any category indicates that the design will probably not fit in that specific HardCopy device.

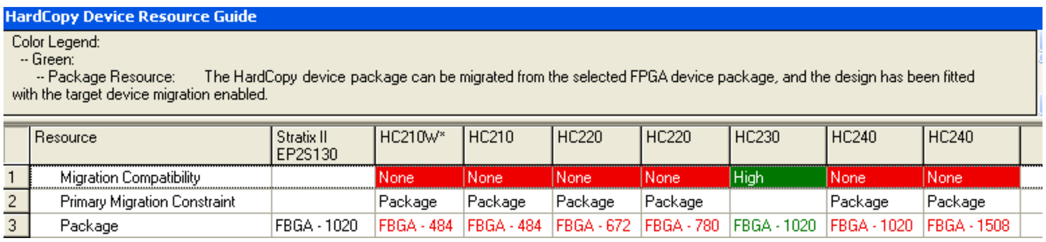
The HardCopy architecture consists of an array of fine-grained HCells, which are used to build logic equivalent to FPGA adaptive logic modules (ALMs) and digital signal processing (DSP) blocks. The DSP blocks in HardCopy devices match the functionality of the FPGA DSP blocks, though timing of these blocks is different than the FPGA DSP blocks because they are constructed of HCell macros.

Memory blocks in HardCopy devices and FPGAs are equivalent. Preliminary timing reports of the HardCopy device are available in the Quartus II software. Final timing results of the HardCopy device are provided by the Altera HardCopy Design Center after the HardCopy back-end implementation process is complete.

For more information about the HardCopy device resources, refer to the respective HardCopy series device handbook, which is available on the Literature page of the Altera website at www.altera.com.

The report example in Figure 3–4 shows the resource comparisons for a design compiled for an EP2S130F1020 device. Based on the report, the HC230F1020 device in the 1,020-pin FineLine BGA package is an appropriate HardCopy device. The EP2S180F1020 device is rated green because the device is specified as a migration target in the example. If the HC230F1020 device is not specified as a migration target during the compilation, its package and migration compatibility is rated medium (orange). The migration compatibilities of the other HardCopy devices are rated none (red), because the package types are incompatible with the FPGA device.

Figure 3–4. HardCopy Device Resource Guide with Target Migration Enabled



HardCopy Device Resource Guide

Color Legend:
-- Green:
-- Package Resource: The HardCopy device package can be migrated from the selected FPGA device package, and the design has been fitted with the target device migration enabled.

	Resource	Stratix II EP2S130	HC210W*	HC210	HC220	HC220	HC230	HC240	HC240
1	Migration Compatibility		None	None	None	None	High	None	None
2	Primary Migration Constraint		Package	Package	Package	Package		Package	Package
3	Package	FBGA - 1020	FBGA - 484	FBGA - 484	FBGA - 672	FBGA - 780	FBGA - 1020	FBGA - 1020	FBGA - 1508

Selecting the Companion Device

In the Quartus II software, you can select a HardCopy companion device to ensure compatibility between the FPGA design and the HardCopy device's resources. To select your HardCopy companion device, on the Assignments menu, click **Device** and select a companion device from the **Companion device** list in the **Device** dialog box.

Selecting a HardCopy companion device for your FPGA prototype constrains the memory blocks, DSP blocks, and pin assignments, so that your design fits into the HardCopy device resources. Pin assignments are constrained in the FPGA design revision, so that the HardCopy device selected is pin-compatible. The Quartus II software also constrains the FPGA design revision so that identical device resources are targeted in both the FPGA and the HardCopy ASIC.

Although not all FPGA ALM configurations are available in HardCopy devices, no restriction is made during synthesis of the FPGA. Unsupported configurations are converted to multiple cells for the HardCopy device.

You can also specify your HardCopy companion device using the following tool command language (Tcl) command:

```
set_global_assignment -name\
DEVICE_TECHNOLOGY_MIGRATION_LIST <HardCopy Device Part Number>
```

For example, to select the HC230F1020 device as your HardCopy companion device for the EP2S130F1020C4 FPGA, use the following the Tcl command:

```
set_global_assignment -name\
DEVICE_TECHNOLOGY_MIGRATION_LIST HC230F1020C
```

Applying Design Constraints

The HardCopy development flow requires that you plan specific steps in addition to the standard FPGA design flow, because you are developing your design for implementation in two devices: a prototype of your design in an FPGA and a companion revision in a HardCopy device for production. Additional settings and constraints in the Quartus II software are required to make the FPGA design compatible with the HardCopy device, and in some cases, you must remove certain settings in the design. This section explains the additional design constraints necessary for your design to be successful in both FPGA and HardCopy devices.

Limit DSP and RAM to HardCopy Device Resources

To maintain compatibility between the FPGA and HardCopy devices, your design must use resources that are common to both families. You must turn on the **Limit DSP & RAM to HardCopy device resources** option in the **Device** dialog box before submitting the design to the Altera HardCopy Design Center for back-end implementation. Turning on this option ensures that your design does not use resources in the FPGA device that are not available in the selected HardCopy device or vice versa.

For more information about the **Limit DSP & RAM to HardCopy device resources** option in the Quartus II software, refer to *Device Dialog Box* in Quartus II Help.

Enabling Design Assistant to Run During Compilation

You must use the Design Assistant in the Quartus II software to check all HardCopy designs 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 each compilation so that you can review the violations to determine which errors you must fix or which you can waive, iteratively.

For more information about the Design Assistant and its rules in the Quartus II software, refer to *About the Design Assistant* in Quartus II Help.

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 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 *Constraining Designs* chapter in volume 2 of the *Quartus II Handbook*.

To ensure that the HardCopy 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 assigning I/O assignments for all I/O pins. Leaving unassigned I/O assignments may result in incompatible assignments.

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

For more information about I/O banks and pins in HardCopy series devices, refer to the respective HardCopy series device handbook, which is available on the Literature page of the Altera website at www.altera.com.

HardCopy III I/O buffers support only the 3.0 V I/O standard with a maximum supply voltage (VCCIO) 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, you must choose an I/O standard with a VCCIO of 3.0 V or less. Selecting an I/O standard that requires a VCCIO of 3.3 V results in a compilation error.

For more information about HardCopy III I/O buffers, refer to the *DC and Switching Characteristics of HardCopy III Devices* chapter of the *HardCopy III Device Handbook*.

HardCopy IV I/O buffers support 3.3 V I/O standards, which you can use as transmitters or receivers in your system. The 3.3 V I/O standard can be supported by using the bank VCCIO at 3.0 V. In this method, the clamp diode (on-chip or off-chip), when enabled, can sufficiently clamp overshoot voltage to within the DC and AC input voltage specification. The clamped voltage can be expressed as the sum of the VCCIO and the diode forward voltage.

For more information about HardCopy IV I/O buffers, refer to the *DC and Switching Characteristics of HardCopy IV Devices* chapter of the *HardCopy IV Device Handbook*.

You must constrain the I/O standards for the design specifically for your HardCopy device. If you do not assign an I/O standard to an I/O pin, the Quartus II software assigns the I/O standard to **2.5 V** by default, which may not be compatible with your design. To check supported I/O standards and identify incompatible I/O settings on the assigned I/O pins, run I/O assignment analysis by pointing to **Start** on the Processing menu, and then clicking **Start I/O Assignment Analysis**. The **Start I/O Assignment Analysis** command verifies the I/O settings and assignments.

Altera recommends verifying the correct output drive strength for the design because the default value in the Quartus II software might not be appropriate for your application. 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.

Quartus II Fitter Settings

To make the HardCopy device implementation more robust across process, temperature, and voltage variations, the Altera HardCopy Design Center requires that you turn on the **Optimize multi-corner timing** option and set the **Timing-driven compilation** option to **Optimize hold timing** for the Quartus II Fitter.

The **Optimize multi-corner timing** option directs the Fitter to optimize a design to meet timing requirements at both the fast-timing and the slow-timing process corners and operating conditions. Setting the **Timing-driven compilation** option to **Optimize hold timing** allows the Fitter to optimize hold time by adding delay to the appropriate paths. You can set these Fitter options in the **Fitter Settings** page of the **Settings** dialog box.

For more information about Fitter settings in the Quartus II software, refer to *Fitter Settings Page (Settings Dialog Box)* in Quartus II Help.

Physical Synthesis Optimization

The physical synthesis optimizations performed in the FPGA device are mapped to the HardCopy companion revision for placement and timing closure. When designing with a HardCopy device first, you can enable physical synthesis optimizations for the HardCopy device. These post-fit optimizations are then passed to the FPGA revision. The optimizations in the base revision are mapped to the companion device architecture and the post-fit netlists of both devices are generated and compared. Therefore, you must have the identical physical synthesis settings for both the HardCopy ASIC and FPGA revisions in order to avoid revision comparison failure.

The **Effort level** on the **Physical Synthesis Optimizations** page of the **Settings** dialog box for HardCopy III and HardCopy IV devices must be set to **Fast** because the performance gain achieved compared to the compilation time is very limited.

For more information about setting physical synthesis optimizations for the FPGA revision of the designs in the Quartus II software, refer to *Setting up and Running the Fitter* in Quartus II Help.

Timing Settings

The TimeQuest Timing Analyzer is a complete static timing analysis tool that you use as a sign-off tool for FPGAs and HardCopy ASICs. The TimeQuest analyzer guides the Fitter and analyzes timing results after compilation and is the required timing analysis tool for all Quartus II software designs.

For more information about the TimeQuest Timing Analyzer, refer to *The Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook* and *About TimeQuest Timing Analysis* in Quartus II Help.

TimeQuest Timing Analyzer Settings

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 by turning on the **Enable multicorner timing analysis during compilation** option. This option directs the TimeQuest analyzer to analyze the design and generate slack reports for the slow and fast corners.

You must also direct the TimeQuest analyzer to remove the common clock path pessimism during slack computation by turning on the **Enable common clock path pessimism removal** option.

You can turn on these TimeQuest analyzer options in the **TimeQuest Timing Analyzer** page of the **Settings** dialog box in the Quartus II software.

For more information about TimeQuest analyzer options in the Quartus II software, refer to *TimeQuest Timing Analyzer Page (Settings Dialog Box)* in Quartus II Help.

Constraints for Clock Effect Characteristics

The `create_clock` and `create_generated_clock` commands create ideal clocks, but do not account for board effects. To account for clock effect characteristics, you can use the `set_clock_latency` and `set_clock_uncertainty` commands.

For more information about how to use these commands, refer to *The Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

You can use the `derive_clock_uncertainty` command to automatically derive the clock uncertainties in your **.sdc** file. This command is useful when you are unsure of the clock uncertainties. The calculated clock uncertainty values are based on I/O buffer, static phase errors (SPE) and jitter in the PLLs, clock networks, and core noise.

The following syntax is for the `derive_clock_uncertainty` command:

```
derive_clock_uncertainty [-h | -help] [-long_help] [-add]
\[-overwrite]
```

For more information about the `derive_clock_uncertainty` command in the Quartus II software, refer to *derive_clock_uncertainty* in Quartus II Help.

When the `derive_clock_uncertainty` command is used, a **PLLJ_PLLSPE_INFO.txt** file is automatically generated in the project directory. This file lists the names of the PLLs, as well as their jitter and SPE values in the design. This text file can be used by the HCII_DTW_CU_Calculator.

Altera strongly recommends that you use the `derive_clock_uncertainty` command in the HardCopy revision. The Altera HardCopy Design Center does not accept designs that do not have clock uncertainty constraints applied by either using the `derive_clock_uncertainty` command or the HardCopy II Clock Uncertainty Calculator, and then using the `set_clock_uncertainty` command.

For more information about how to use the HardCopy II Clock Uncertainty Calculator, refer to the *HardCopy II Clock Uncertainty Calculator User Guide*.

LogicLock Regions

LogicLock regions are flexible floorplan location constraints that help you place logic on the target device. You can use LogicLock regions in FPGA designs targeted to HardCopy devices, which are also passed onto the HardCopy companion revision.

When LogicLock regions are created in a HardCopy device, they start with width and height dimensions set to (**1,1**), and the origin coordinates for placement are at **X1_Y1** in the lower left corner of the floorplan. You must adjust the size and location of the LogicLock regions in HardCopy devices before compiling the design.

Altera recommends that you do not use floating LogicLock regions for HardCopy devices because floating LogicLock regions may affect the design's ability to meet timing closure. Additionally, you must manually size and place HardCopy device LogicLock regions in the floorplan; you cannot set the LogicLock regions to **Auto**.

For more information about using LogicLock regions, refer to the *Analyzing and Optimizing the Design Floorplan* chapter in volume 2 of the *Quartus II Handbook*.

PowerPlay Power Analyzer

You can perform initial power estimation and analysis of your HardCopy and FPGA devices using the PowerPlay Early Power Estimator. You can then use the PowerPlay Power Analyzer for a more accurate estimation of your device's power consumption.

For more information about using the PowerPlay Power Analyzer, refer to the *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*.

Incremental Compilation

The Quartus II 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.

Quartus II incremental compilation within a single Quartus II project is supported for the base family for both the FPGA first and HardCopy first flows. Exporting and importing partitions is not supported in HardCopy ASIC or FPGA device compilations when there is a migration device setting.

For more information about using Quartus II incremental compilation, refer to the *Quartus II Incremental Compilation for Hierarchical and Team-Based Design* chapter in volume 1 of the *Quartus II Handbook* and *About Incremental Compilation* in Quartus II Help.

External Memory Interfaces

The HardCopy I/O structure is equivalent to the Stratix I/O structure, providing high-performance support for existing and emerging external memory standards such as DDR, DDR2, DDR3, QDRII, QDRII+, and RLDRAM II.

A self-calibrating soft IP core (UniPHY) optimized to take advantage of HardCopy device I/Os in conjunction with the Quartus II TimeQuest Timing Analyzer, provides the total solution for the highest reliable frequency of operation across process, voltage, and temperature (PVT).

Compiling the Design and Creating Companion Revisions

After you finish applying constraints to your prototype design, you can compile your design and review the messages generated by the Quartus II software during compilation to check for any potential problems. If you do not have any violations that you must fix, you can proceed to creating or overwriting a companion revision, as described in "Creating a Companion Revision" on page 3–15. If you have violations that you must fix, you must fix the violations, recompile the design, and recheck for violations before proceeding to creating a companion revision.

Creating a Companion Revision

The Quartus II software creates specific HardCopy design revisions of the project in conjunction with the primary project revisions. These parallel design revisions for HardCopy devices are called companion revisions. You can create multiple design revisions for both the FPGA and the HardCopy device. For example, if your initial FPGA revision is named *top* and the corresponding HardCopy revision is named *top_hc*, you could create another FPGA revision, named *top_fpga*, and the corresponding HardCopy revision would be named *top_fpga_hc*.

Although you can create multiple design revisions, Altera recommends that you maintain only one FPGA revision after you create the HardCopy companion revision.

After you have successfully compiled your FPGA prototype, you can create and compile the HardCopy companion revision of your design. You can associate only one FPGA revision to one HardCopy companion revision. If you create more than one revision or companion revision, set the current companion for the revision you are working on.

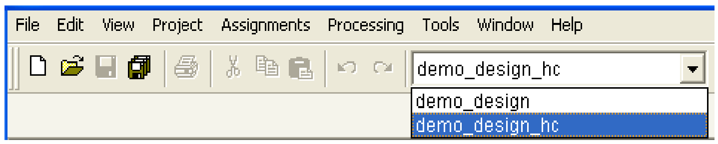
For more information about creating or setting a companion revision in the Quartus II software, refer to *Migrating a Design to a HardCopy or FPGA Device* in Quartus II Help.

Compiling the HardCopy Companion Revision

The Quartus II software contains preliminary timing models for HardCopy devices, and you can gauge the degree of performance improvement you can achieve in the HardCopy device compared to the FPGA by compiling your HardCopy design with preliminary timing information in the Quartus II software. The timing constraints for the HardCopy companion revision can be the same as the FPGA design used to create the revision. Altera verifies that the HardCopy companion device timing requirements are met in the Altera HardCopy Design Center.

After you create your HardCopy companion revision from your compiled FPGA design, select the companion revision in the Quartus II software design revision pull-down list as shown in Figure 3–5 or from the **Revisions** list, and then compile the HardCopy companion revision. After you compile your design in the Quartus II software, you can perform a comparison check of the HardCopy companion revision to the FPGA prototype revision as described in "Comparing HardCopy and FPGA Companion Revisions" on page 3–21.

Figure 3–5. Changing Current Revision



HardCopy Design Readiness Check

The HardCopy Design Readiness Check (HCDRC) is available as one of the processing steps in the default compilation of both the FPGA first and the HardCopy first flows. This feature checks issues that must be addressed prior to handing off the

HardCopy design to the Altera HardCopy Design Center for the HardCopy back-end process. The HCDRC is different from the user-driven approach in the HardCopy Advisor, in which you must manually open the advisor to check for any violations.

The checks performed in the HCDRC include settings (global, instance, and operating settings), I/O, PLL, RAM, and ALTGX checks.

Turning the HardCopy Design Readiness Check On and Off

You can turn the HCDRC on or off in the **More Compilation Process Settings** dialog box, or by using the following **.qsf** file assignments:

```
set_global_assignment -name \ FLOW_HARDCOPY_DESIGN_READINESS_CHECK ON
```

```
set_global_assignment -name \ FLOW_HARDCOPY_DESIGN_READINESS_CHECK OFF
```

Setting Check

The Setting Check report lists the results of the setting checks from the Handoff report. The Setting Check report contains the sections described below.

Summary

The Summary section displays the number of settings that do not meet recommendations. One of the following messages appears:

<*number*> global setting(s) do not meet recommendation. Please review the recommendation and do appropriate correction as it may affect the result of the migration to HardCopy.

or

<*number*> instance setting(s) do not meet recommendation. Please review the recommendation and do appropriate correction as it may affect the result of the migration to HardCopy.

Global Setting

The Global Setting section displays recommendations for global settings only. Global settings with values other than the recommended values are highlighted in red.

Instance Setting

The Instance Setting section displays recommendations for instances assignments only. Instance settings with values other than the recommended values are highlighted in red.

Operating Setting

The Operating Setting section displays checks related to the recommended operating settings for the FPGA and the HardCopy device.

This check is primarily applicable to Stratix III devices used as prototype FPGAs because HardCopy III devices only support 0.9 V core voltage, whereas Stratix III devices support both 1.1 V and 0.9 V core voltage.

The Setting Check reports also include checking for illegal assignments in the HardCopy design flow.

An example of illegal assignment checks is shown in Example 3–1.

Example 3–1. Illegal Assignment Checks

```
USE_CHECKERED_PATTERN_AS_UNINITIALIZED_RAM_CONTENT ON (1)
SIGNAL_PROBE_ENABLE ON|OFF
SIGNAL_PROBE_SOURCE ON|OFF (2)
```

Notes to Example 3–1:

(1) Refer to the section “RAM Usage Check” on page 3–19.
(2) SignalProbe is not supported in HardCopy ASICs.

I/O Check

The I/O check ensures that you have assigned location assignments for the pins, I/O standards, current strength assignments, output pin load assignments, termination assignments, and also checks for any unconnected pins.

The following message appears in the message panel during compilation when the HCDRC detects missing I/O standard assignments:

```
<number> pin(s) have no explicit I/O Standard assignments provided in the
setting file and default values are being used. Please add a specific I/O
Standard assignment for these pins.
```

Input Pin Placement for Global and Regional Clock Check

Due to the difference in the interconnect delays between the FPGA and HardCopy device, using non-primary clock inputs as clock inputs in a design can cause timing closure to be a problem when migrating the FPGA to the HardCopy device. The Input Pin Placement for Global and Regional Clock check informs you of potential problems before finalizing the pin location, so that any clock inputs can be moved to the primary clock input.

This check lists all the pins that drive the global or regional clock, but are not placed in a dedicated clock pad. All pins are required to have manual location assignments. Pins that are missing location assignments are listed in the Missing Pin Location Assignment report.

The following message appears in the message panel during compilation and also appears in the I/O Check Summary:

`<number> pin(s) drives global or regional clock, but is not placed in a dedicated clock pin position. Clock insertion delay will be different between FPGA and HardCopy companion revisions because of differences in local routing interconnect delays.`

PLL Usage Check

The PLL Usage Check report lists PLL usage requirements and violations checks.

PLL Real-Time Reconfigurable Check

This check highlights the PLLs without PLL reconfiguration. PLL reconfiguration allows fine tuning of the PLLs in the design after manufacturing. PLL elements without PLL reconfiguration are listed in a table.

The following message appears in the message panel during compilation and also appears in the Logic Check Summary:

`<number> PLL(s) don't have real time reconfiguration. It is highly recommended that each PLL to have PLL reconfiguration for designs migrating to HardCopy.`

PLL Clock Outputs Driving Multiple Clock Network Types Check

This check is derived from the Design Assistant rule check for HardCopy devices (Rule ID H102) and lists all PLL instances in the current design that have clock outputs driving multiple clock network types.

The following message appears in the message panel during compilation if the HCDRC detects this type of violation:

`Found <number> PLL(s) with clock outputs that drives multiple clock network types.`

PLL with No Compensation Mode Check

This check lists all PLLs that are in No Compensation operating mode. This setting is not recommended for a design migrating to a HardCopy device because of differences in the clock networks and the clock delays between the FPGA and HardCopy device.

The following warning message appears during compilation when a PLL is in a No Compensation mode:

`<number> PLL(s) is operating in a "No compensation" mode.`

PLL with Normal or Source Synchronous Mode Feeding Output Pin Check

When a PLL is directly feeding an output pin, it must be set to Zero Delay Buffer operating mode. If a PLL is set either in Normal Compensation mode or Source Synchronous mode, a warning message is issued during compilation.

The following warning message appears during the runtime of HC Ready:

```
<number> PLL(s) is in normal or source synchronous mode that is not fully
compensated because it feeds an output pin -- only PLLs in zero delay
buffer mode can fully compensate output pins.
```

RAM Usage Check

HardCopy series devices do not support initialized RAM blocks upon power-up. However, you can use the RAM Initializer megafunction to initialize the RAMs of a HardCopy series device in your design with the content of a ROM.

For more information about the RAM Initializer megafunction, refer to the *RAM Initializer (ALTMEM_INIT) Megafunction User Guide*.

In HardCopy series devices, RAM blocks power up uninitialized. During the RAM Usage check, the HCDRC checks for RAMs that are initialized using a Memory Initialization File (**.mif**). Any RAM with a **.mif** file is listed in a table.

The following warning message appears during compilation when the HCDRC detects this type of error:

```
<number> RAM(s) have Memory Initialization File (MIF). HardCopy devices do
not allow initialized RAM. Please ensure that no RAM is initialized by a
MIF file.
```

Initialized Memory Dependency Testing

The Assembler module of the Compiler optionally allows you to write an FPGA programming file with an initialized checkerboard pattern for memory contents of M4K memory blocks for the FPGA revision. Use this option only on a parallel copy of your compiled FPGA design that you want to test on your board. Using this option in a FPGA revision used to migrate to the HardCopy revision creates irreconcilable revision differences between the FPGA and HardCopy designs because the HardCopy handoff design cannot physically have any initialized memory content.

To create a programming file with an initialized checkerboard pattern, perform the following steps in the Quartus II software:

1. Compile your completed FPGA revision to use for prototype testing. You should eventually use this FPGA revision to create your HardCopy companion revision.
2. Create and compile the HardCopy companion revision.
3. Compare your HardCopy companion revision.
4. Generate and archive the HardCopy handoff files for your design.
5. Switch back to your FPGA revision, and on the Project menu, click **Revisions**, and then double click **<<new revision>>** in the **Revisions** table.
6. In the **Create Revision** dialog box, type a revision name in the **Revision name** box and turn on **Copy database** and **Set as current revision**. This step copies your FPGA revision and sets the new revision as the current open revision.

7. On the Assignments menu, click **Settings**, and then click **Assembler** in the **Category** list. Turn on **Use checkered pattern as uninitialized RAM content** on the **Assembler** page, or add the following line to the revision **.qsf** file:

```
set_global_assignment -name
USE_CHECKERED_PATTERN_AS_UNINITIALIZED_RAM_CONTENT ON
```

8. Run the Assembler in the FPGA revision to generate a new programming file for your FPGA.
9. Test the new programming file in your prototype environment to determine if your design has a dependency for FPGA RAM contents initialized with zeros after power-up and configuration.

Because the checkerboard pattern is used for testing, the patterns written into the RAM blocks for the new programming file may not detect all cases of zero-initialized RAM content dependencies. Some designs may detect only one bit as zero (for example, the LSB of a memory word), so this method may not detect all cases. This checkerboard pattern test will detect a case when a full RAM word line is expected as zeros at startup.

ALTGX Usage Check

Beginning in the Quartus II software version 10.0, the ALTGX Usage check performs checks on ALTGX instance usage for designs targeting Stratix IV GX and HardCopy IV GX devices.

The HCDRC checks all the ALTGX instances that are initialized in the design for connectivity with the ALTGX_RECONFIG instance. The following warning message appears, with the respective instance HSSI_CMU atom name, for ALTGX instances that do not connect to an ALTGX_RECONFIG instance:

```
ALTGX megafunctions do not have ALTGX_RECONFIG megafunctions connected.
Altera recommends connecting ALTGX_RECONFIG megafunction to each ALTGX
megafunction when migrating your designs to HardCopy devices.
```

Timing Closure and Verification

After compiling the project for the FPGA and HardCopy designs, verify that the design meets your timing requirements. Review the messages generated by the Quartus II software during compilation to check for any potential problems. Also, verify the design functionality between the FPGA and HardCopy devices with the **HardCopy Companion Revision Comparison** command as described in "Comparing HardCopy and FPGA Companion Revisions" on page 3–21.

You can also use third-party formal verification software, Cadence Encounter Conformal verification software, to run formal verification, and then compare the companion revisions. Formal verification is described in more detail in "Formal Verification of FPGA and HardCopy Revisions" on page 3–21.

Timing Closure with the TimeQuest Timing Analyzer

The TimeQuest Timing Analyzer is the timing analysis tool for all HardCopy devices during the front-end design process 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.

Comparing HardCopy and FPGA Companion Revisions

The Quartus II software uses the companion revisions in a single Quartus II project to maintain compatibility between the FPGA and HardCopy ASIC. This methodology allows you to design with one set of RTL code that is used in both the FPGA and HardCopy ASIC, guaranteeing functional equivalency.

When making changes to your design in a companion revision, use the **Compare HardCopy Companion Revisions** command to ensure that your design matches your HardCopy design functionality and compilation settings. You must perform this command after both the FPGA and HardCopy designs are compiled and before you hand off the design to the Altera HardCopy Design Center.

The Comparison Revision Summary in the Compilation report identifies where assignments were changed between revisions or if there is a change in the logic resource count due to different compilation settings.

For more information about comparing companion revisions in the Quartus II software, refer to *Migrating a Design to a HardCopy or FPGA Device* in Quartus II Help.

Formal Verification of FPGA and HardCopy Revisions

Third-party formal verification software, Cadence Encounter Conformal verification software, is used for several FPGA and HardCopy families. The formal verification flow for HardCopy ASIC designs is a two-step process. First, run formal verification on the FPGA netlist to ensure that the FPGA netlist matches the RTL. Second, use the **Compare HardCopy Revisions** command in the Quartus II software to ensure that the HardCopy implementation matches the FPGA.

Although this flow is enabled, performing formal verification is not necessary due to the one-to-one mapping of logic between the FPGA prototype and the HardCopy ASIC.

To use the Conformal software with the Quartus II software project for your FPGA design revision, you must automatically run the EDA Netlist Writer during compilation so it can generate the necessary netlist and command files required to run the Conformal software.

To automatically run the EDA Netlist Writer during the compilation of your FPGA revision, perform the following steps:

1. On the Assignments menu, click **Settings**.
2. In the **Category** list, under **EDA Tool Settings**, click **Formal Verification**, and then in the **Tool name** list, select **Conformal LEC**.

3. Compile your FPGA and HardCopy design revisions.

The Quartus II EDA Netlist Writer produces the netlist for the FPGA revision. You can compare your FPGA post-compilation netlist to your RTL source code using the scripts generated by the EDA Netlist Writer.

After compiling both the FPGA and HardCopy revisions, you can run the **Compare HardCopy Revisions** command, as described in "Comparing HardCopy and FPGA Companion Revisions" on page 3–21 to ensure that the HardCopy implementation matches the FPGA.

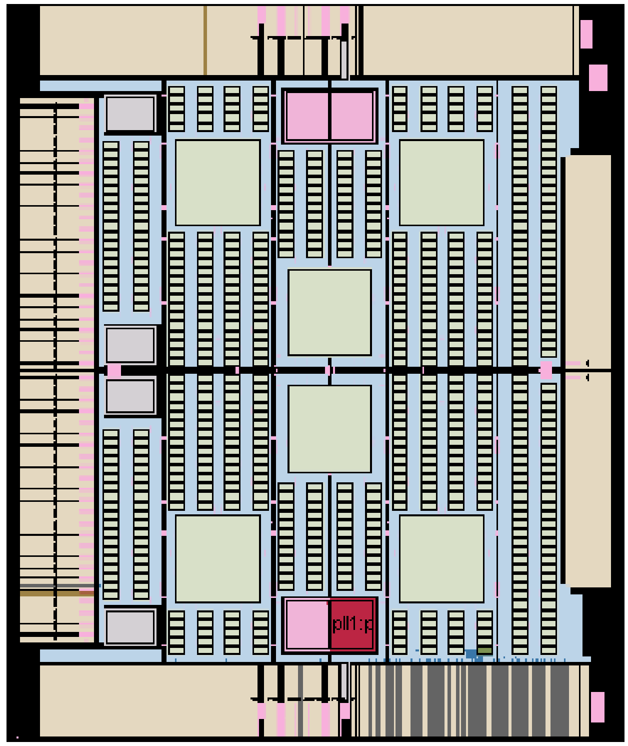
For more information about using the Cadence Encounter Conformal verification software, refer to the *Cadence Encounter Conformal Support* chapter in volume 3 of the *Quartus II Handbook*.

HardCopy Floorplan View

The Quartus II software displays the floorplan and placement of your HardCopy companion revision. This floorplan shows the preliminary placement and connectivity of all I/O pins, PLLs, memory blocks, HCell macros, and DSP HCell macros. Congestion mapping of routing connections can be viewed using the **Layers Setting** dialog box, which is available from the View menu of the Chip Planner. Congestion mapping is useful in analyzing densely packed areas of your floorplan that can reduce the peak performance of your design. The Altera HardCopy Design Center verifies final HCell macro timing and placement to guarantee that timing closure is achieved.

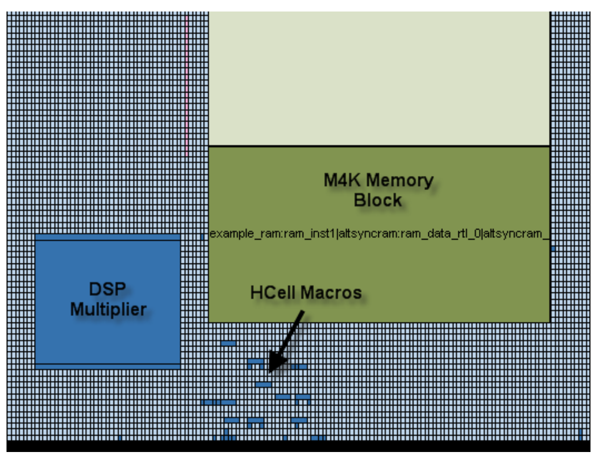
Figure 3–6 shows an example of the HC230F1020 device floorplan.

Figure 3–6. HC230F1020 Device Floorplan



In this small example design, the logic is placed near the bottom edge. You can see the placement of a DSP block constructed of HCell macros, various logic HCell macros, and an M4K memory block. A close-up view of this region is shown in Figure 3–7.

Figure 3–7. Close-Up View of Floorplan



The Altera HardCopy Design Center performs final placement and timing closure on your HardCopy design based on the timing constraints provided in the FPGA design.

For more information about the Altera HardCopy Design Center process, refer to the respective HardCopy series device handbook, which is available on the Literature page of the Altera website at www.altera.com.

Performing ECOs with Quartus II Engineering Change Management with the Chip Planner

During the last stage of the design cycle, the ability to implement a specific portion of the design without affecting the rest of its logic is critical. As described in "Incremental Compilation" on page 3–14, incremental compilation allows you to implement and manage certain partitions of the design and preserve the optimization results for the rest of the design. However, implementing changes may become difficult to manage because Engineering Change Orders (ECOs) are often implemented as last-minute changes to your design.

With the Altera Chip Planner, you can shorten the design cycle time significantly. When changes are made to your design as ECOs, you do not have to perform a full compilation in the Quartus II software. Instead, you can make changes directly to the post place-and-route netlist, generate a new programming file, test the revised design by performing a gate-level simulation and timing analysis, and then verify the change on the system. When the change has been verified on the FPGA, switch to the HardCopy revision, apply the same ECOs, run timing analysis and the Assembler, compare the revisions, and then run the HardCopy Netlist Writer for design submission.

There are three types of migration scenarios:

- One-to-one changes, which are changes that can be implemented on both FPGA and HardCopy architectures
- Changes that must be implemented differently on the two architectures to achieve the same result
- Changes that cannot be implemented on both architectures

The following sections describe the methods for migrating each type of changes.

Migrating One-to-One Changes

One-to-one changes are implemented using identical commands in both architectures, and typically include changes that affect only I/O cells or PLL cells. Some examples of one-to-one changes include 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). These types of changes can be implemented identically on both architectures.

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 FPGA and HardCopy ASIC revisions to avoid a revision comparison failure.

If such changes are exported to Tcl, directly reapplying the generated Tcl script (with a minor text edit) on the companion revision implements the appropriate changes as described in the following steps:

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

4. Export the changes from the Change Manager to Tcl, by right-clicking the entry, pointing to **Export**, and then clicking **Export All Changes As…**
5. Save the **.tcl** script, which you will use in the HardCopy revision.

In the HardCopy revision, apply the **.tcl** script to the companion revision by following these steps:

1. Open the generated Tcl script and change the `project_open` <*project*> `-revision` <*revision*> line to refer to the appropriate companion revision.
2. Save the **.tcl** script.
3. Apply the Tcl script to the companion revision. On the Tools menu, click **Tcl Scripts** and in the **Tcl Scripts** dialog box, select **ECO Tcl** and click **Run**.

Migrating Changes that Must Be Implemented Differently

Some changes must be implemented differently on the two architectures, such as changes affecting the logic of the design. Some examples include LUTMASK changes, `LC_COMB/HSADDER` creation and deletion, connectivity changes not described in the previous section, and different PLL settings for the FPGA and the HardCopy revisions.

For more information about how to use different PLL settings for the FPGA and HardCopy devices, refer to *AN 432: Using Different PLL Settings Between Stratix II and HardCopy II Devices*.

Table 3–3 summarizes the suggested implementation of various changes that must be implemented differently on the FPGA and HardCopy architectures.

Table 3–3. Implementation Suggestions for Changes that Must Be Implemented Differently

Change Type	Suggested Implementation
LUTMASK changes	Because a single FPGA atom can require multiple HardCopy atoms to implement, you may need to change multiple HardCopy atoms to implement the change, including adding or modifying connectivity.
Make/Delete LC_COMB	If you are using an FPGA `LC_COMB` in extended mode (7-LUT) or are using a SHARE chain, you must create multiple atoms to implement the same logic functions in the HardCopy device. Additionally, the placement of the `LC_COMB` cell has no meaning in the companion revision because the underlying resources are different.
Make/Delete LC_FF	Basic creation and deletion is the same on both architectures; however, similar to `LC_COMB` creation and deletion, the location of an `LC_FF` in a HardCopy and FPGA revision do not translate.
Editing logic connectivity	Because a `LCELL_COMB` atom may be required to be broken up into several HardCopy `LCELL_COMB` atoms, the source or destination ports for connectivity changes might have to be analyzed to properly implement the change in the companion revision.

Changes That Cannot Be Migrated

A small set of changes are incompatible and cannot be implemented in both architectures. The best example of this incompatibility occurs when moving logic in a design; because the logic fabric is different between the two architectures, locations in the FPGA and HardCopy device are not compatible with each other.

Overall Migration Flow

This section outlines the migration flow and the suggested procedure for implementing changes in both FPGA and HardCopy revisions to ensure a successful revision comparison so the design can be submitted to the Altera HardCopy Design Center.

Preparing the Revisions

The general process for migrating changes from the FPGA to HardCopy device or vice versa is the same and is described below:

1. Compile the design on the initial device.
2. Migrate the design from the initial device to the target device in the companion revision.
3. Compile the companion revision.
4. Run the **Compare HardCopy Companion Revisions** command. Both revisions should pass the revision comparison.

If testing identifies problems requiring ECO changes, equivalent changes can be applied to both FPGA and HardCopy revisions, as described in the following section.

Applying ECO Changes

The general process for applying equivalent changes in companion revisions is described below:

1. To make changes in one revision using the Chip Planner, and then verify and export these changes, follow these steps:
   a. Make changes using a Chip Planner tool (Chip Planner, Resource Property Editor, or Change Manager).
   b. Perform a netlist check using the **Check and Save All Netlist Changes** command.
   c. Verify correctness using timing analysis, simulation, and prototyping (FPGA only). If more changes are required, repeat steps a and b.
   d. Export change records from the Change Manager to Tcl scripts, or **.csv** or **.txt** file formats. This exported file is used to assist in making the equivalent changes in the companion revision.
2. Open the companion revision in the Quartus II software.
3. Using the exported file, manually reapply the changes using a Chip Planner tool. As stated previously, some changes can be reapplied directly to the companion revision (either manually or by applying the Tcl commands), while others require some modifications.
4. Run the **Compare HardCopy Revisions** command. The revisions should match.
5. Verify the correctness of all changes, which may require running timing analysis.

6. Run the **HardCopy Assembler** command and the **HardCopy Netlist Writer** command for design submission along with handoff files.
   - Use the following Tcl command to run the HardCopy Assembler:

     ```
     execute_module -tool asm -args "--read_settings_files=off --
     write_settings_files=off"
     ```

   - Use the following Tcl command to run the HardCopy Netlist Writer:

     ```
     execute_module -tool cdb \
     -args "--generate_hardcopy_files"\
     ```

For more information about using the Chip Planner, refer to the *Engineering Change Management with the Chip Planner* chapter in volume 2 of the *Quartus II Handbook*.

Preparing the Design for Handoff

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

Generating a HardCopy Handoff Report

The **Generate HardCopy Handoff Report** command creates the HardCopy Handoff report, which provides important information about the design for the Altera HardCopy Design Center to review.

After you generate the HardCopy Handoff report, you can archive the design using the **Archive HardCopy Handoff Files** command, which is described in "Archiving HardCopy Handoff Files".

For more information about the **Generate HardCopy Handoff Report** command in the Quartus II software, refer to *Generate HardCopy Handoff Report Command (Project Menu)* in Quartus II Help.

Archiving HardCopy Handoff Files

The last step in the HardCopy design flow is to archive the HardCopy project for submission to the Altera HardCopy Design Center for HardCopy back-end implementation. The **Archive HardCopy Handoff** command creates a unique **.qar** file, which is different than the standard file the Quartus II project archive utility generates. The HardCopy archive file contains only the necessary data from the Quartus II project required to implement the design in the Altera HardCopy Design Center.

For more information about the **Archive HardCopy Handoff Files** command in the Quartus II software, refer to *Archive HardCopy Handoff Files Dialog Box* in Quartus II Help.

Document Revision History

Table 3–4 shows the revision history for this chapter.

Table 3–4. Document Revision History

Date	Version	Changes
November 2011	11.0.1	Template update.
May 2011	11.0.0	■ Reorganized the chapter ■ Removed the "Quartus II Features for HardCopy Planning" section ■ Changed the "HardCopy Recommended Settings in the Quartus II software" section to "Applying Design Constraints" ■ Organized the "HardCopy Utilities" subsections by design flow ■ Added the "Selecting the Prototype and Companion Devices" section ■ Added the "I/O Assignments" section ■ Added the "Quartus II Fitter Settings" section ■ Added the "External Memory Interfaces" section ■ Added the "Compiling the Design and Creating Companion Revisions" section ■ Added the "Timing Closure and Verification" section ■ Added the "Preparing the Design for Handoff" section ■ Linked applicable sections to Quartus II Help
December 2010	10.1.0	■ Edited the "Timing Settings" section to remove support for the Classic Timing Analyzer ■ Changed to new document template ■ Editorial changes
July 2010	10.0.0	■ Added the "ALTGX Usage Check" section ■ Updated the "LogicLock Regions" section for updated companion revision support ■ Updated the "Incremental Compilation" section for updated companion revision support ■ Linked sections throughout the chapter to Quartus II Help ■ Removed the "Referenced Documents" section
November 2009	9.1.0	■ Removed HardCopy Stratix legacy support information ■ Updated the "Physical Synthesis Optimization" section ■ Updated the "Quartus II Software Features Supported for HardCopy Designs" section ■ Updated the "Referenced Documents" section ■ Updated the tables and figures for HardCopy Series devices
March 2009	9.0.0	■ Updated the "RAM Usage Check" section ■ Updated the "Referenced Documents" section
November 2008	8.1.0	■ Added HardCopy IV E support information ■ Added notes for Initialized Memory Dependency testing ■ Changed page size to 8.5" × 11"
May 2008	8.0.0	■ Updated the "RAM Usage Check" section ■ Updated "Referenced Documents"

For previous versions of the *Quartus II Handbook*, refer to the Quartus II Handbook Archive.

Take an online survey to provide feedback about this handbook chapter.