

## Introduction

Advanced design techniques using Altera® HardCopy® Stratix® devices can yield tremendous performance improvements over the design implemented in a Stratix FPGA device. After you verify your Stratix FPGA design in system operation and are ready to migrate to a HardCopy Stratix device, additional device performance is possible through the migration. This chapter focuses on Quartus® II software advanced design techniques that apply to both Stratix FPGA devices and HardCopy Stratix devices. Use these techniques to increase your maximum clock frequency, improve input and output pin timing, and improve timing closure in HardCopy Stratix designs.



Every design is different. The techniques described in this chapter may not apply to every design, and may not yield the same level of improvement.

This document discusses the following topics:

- Planning Stratix FPGA design for HardCopy Stratix design conversion
- Using LogicLock™ regions in HardCopy Stratix designs
- Using Design Space Explorer (DSE) on HardCopy Stratix designs
- Design performance improvement example

## Background Information

To understand the Quartus II software and device architecture, and to use the advanced design techniques described in this chapter, Altera recommends reading the *HardCopy Series Handbook* and the following chapters in the *Quartus II Software Handbook*:

- *Design Recommendations for Altera Devices and the Quartus II Design Assistant*
- *Design Optimization for Altera Devices*
- *Design Space Explorer*
- *Analyzing and Optimizing the Design Floorplan*
- *Netlist Optimizations and Physical Synthesis*

## Planning Stratix FPGA Design for HardCopy Stratix Design Conversion

In order to achieve greater performance improvement in your HardCopy Stratix device, additional Quartus II software constraints and placement techniques in the `HARDCOPY_FPGA_PROTOTYPE` design project may be necessary. This does not mean changing the source hardware description language (HDL) code or functionality, but providing additional constraints in the Quartus II software that specifically impact HardCopy Stratix timing optimization.

Planning ahead for migration to the HardCopy design, while still modifying the `HARDCOPY_FPGA_PROTOTYPE` design, can improve design performance results. You must anticipate how portions of your FPGA design are placed and connected in the HardCopy device floorplan. The HardCopy device floorplan is smaller than the FPGA device floorplan, allowing use of the customized metal routing in HardCopy Stratix devices.

### Partitioning Your Design

Partitioning your design into functional blocks is essential in multi-million gate designs. With a HardCopy Stratix device, you can implement approximately one million ASIC gates of logic. Therefore, Altera recommends hierarchical-design partitioning based on system functions.

When using a hierarchical- or incremental-design methodology, you must consider how your design is partitioned to achieve good results. Altera recommends the following practices for partitioning designs as documented in the *Design Recommendations for Altera Devices* chapter in volume 1 of the *Quartus II Development Software Handbook*:

- Partition your design at functional boundaries.
- Minimize the I/O connections between different partitions.
- Register all inputs and outputs of each block. This makes logic synchronous and avoids glitches and any delay penalty on signals that cross between partitions. Registering I/O pins typically eliminates the need to specify timing requirements for signals that connect between different blocks.
- Do not use glue logic or connection logic between hierarchical blocks. When you preserve hierarchy boundaries, glue logic is not merged with hierarchical blocks. Your synthesis software may optimize glue logic separately, which can degrade synthesis results and is not efficient when used with the LogicLock design methodology.
- Logic is not synthesized or optimized across partition boundaries. Any constant values (for example, signals set to GND), are not propagated across partitions.

- Do not use tri-state signals or bidirectional ports on hierarchical boundaries. If you use tri-state boundaries in a lower-level block, synthesis pushes the tri-state signals through the hierarchy to the top-level. This takes advantage of the tri-state drivers on the output pins of the Altera device. Since this requires optimizing through hierarchies, lower-level boundary tri-state signals are not supported with a block-level design methodology.
- Limit clocks to one per block. Partitioning your design into clock domains makes synthesis and timing analysis easier.
- Place state machines in separate blocks to speed optimization and provide greater encoding control.
- Separate timing-critical functions from non-timing-critical functions.
- Limit the critical timing path to one hierarchical block. Group the logic from several design blocks to ensure the critical path resides in one block.

These guidelines apply to all Altera device architectures including HardCopy Stratix devices. Partitioning functional boundaries to have all outputs immediately registered is crucial to using LogicLock regions effectively in HardCopy devices. With registered outputs, you allow the signals to leave a function block at the start of the clock period. This gives the signals more set-up time to reach their endpoints in the clock period. In large designs that are partitioned into multiple function blocks, the block-to-block interconnects are often the limiting factor for  $f_{MAX}$  performance. Registered outputs give the Quartus II Fitter the optimal place-and-route flexibility for interconnects between major function blocks.

### Physical Synthesis Optimization

All physical synthesis settings in the Quartus II software can be used in the `HARDCOPY_FPGA_PROTOTYPE` design. These settings are found in the **Physical Synthesis Optimizations** section of the **Fitter Settings** dialog box (Assignments menu) and include the following settings:

- Physical synthesis for combinational logic
- Register duplication
- Register retiming

These settings can improve FPGA performance while developing the `HARDCOPY_FPGA_PROTOTYPE`. All modifications are passed along into the HardCopy Stratix project when you run the HardCopy Timing Optimization wizard. After running the HardCopy Timing Optimization wizard and subsequently opening the HardCopy project in the Quartus II software, these physical synthesis optimizations are disabled. No further modifications to the netlist are made.

Altera recommends physical synthesis optimizations for the `HARDCOPY_FPGA_PROTOTYPE`. The work done in the prototype enhances performance in the HardCopy Stratix device after migration. Duplicating combinational logic and registers can increase area utilization, which limits placement flexibility when designs exceed 95% logic element (LE) utilization. However, duplicating combinational logic and registers can help with performance by allowing critical paths to be duplicated when their endpoints must reach different areas of the device floorplan.



For more information on netlist and design optimization, refer to *Area Optimization and Timing Closure* in volume 2 of the *Quartus II Development Software Handbook*.

## Using LogicLock Regions in HardCopy Stratix Designs

Create LogicLock regions in the `HARDCOPY_FPGA_PROTOTYPE` project and migrate the regions into the HardCopy Stratix optimization project using the Quartus II software. LogicLock regions can provide significant benefits in design performance by carefully isolating critical blocks of logic, including:

- MegaCore® IP functions
- I/O interfaces
- Reset or other critical logic feeding global clock lines
- Partitioned function blocks

You must compile your design initially without LogicLock regions present and review the timing analysis reports to determine if additional constraints or LogicLock regions are necessary. This process allows you to determine which function blocks or data paths require LogicLock regions.

Create LogicLock regions in the `HARDCOPY_FPGA_PROTOTYPE` design project in the Quartus II software. This transfers the LogicLock regions to the HardCopy design project after the HardCopy Timing Optimization Wizard is run. Although the Quartus II software transfers the contents of the LogicLock region, the area, location, and soft boundary settings revert to their default settings in the HardCopy project immediately after the HardCopy Timing Optimization Wizard is run.

If you are using LogicLock regions, Altera recommends you use the **Migration Only** setting in the HardCopy Timing Optimization Wizard to create the HardCopy design project. You should not compile your design automatically using the **Full Compilation** or **Migrate and Compile** options in the wizard. Open the HardCopy design project and verify that the LogicLock region properties meet your desired settings before compiling the HardCopy optimization project. LogicLock soft regions are

turned on by default in the HardCopy Stratix design. While this does allow the Fitter to place all logic in your design with fewer restrictions, it is not optimal for performance improvement in the HardCopy Stratix design.

### Recommended LogicLock Settings for HardCopy Stratix Designs

Altera recommends the following LogicLock region settings for the `HARDCOPY_FPGA_PROTOTYPE`:

- Turn on **Reserve Unused Logic**
- Turn off **Soft Region**
- Select either **Auto** or **Fixed** as the **Size** (design-dependent)
- Select either **Floating** or **Locked** as the **Location** (design-dependent)

When using the **Reserve Unused Logic** setting in a design with high resource utilization (> 95% LE utilization), and a large number of LogicLock regions, the design may not fit in the device. Turning off **Reserve Unused Logic** in less critical LogicLock regions can help Fitter placement. The LEs allowed to float in placement and be packed into unused LEs of LogicLock regions may not be placed optimally after migration to the HardCopy Stratix device since they are merged with other LogicLock regions.

After running the HardCopy Timing Optimization Wizard, the LogicLock region properties are reset to their default conditions. This allows a successful and immediate placement of your design in the Quartus II software. You can further refine the LogicLock region properties for additional benefits.

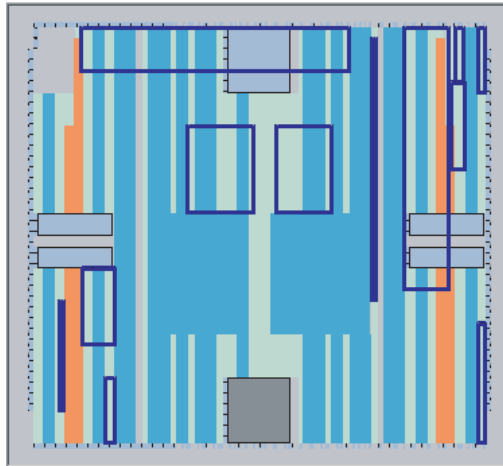
Altera recommends using the following properties for LogicLock regions in the HardCopy design project:

- Turn off **Soft Region**
- Select either **Auto** or **Fixed** as the **Size** after you are satisfied with the placement and timing result of a LogicLock region in a successful HardCopy Stratix compilation
- Select either **Floating** or **Locked** as the **Location** after you are satisfied with the placement and timing results
- **Reserve Unused Logic** is not applicable in the HardCopy Stratix device placement because logic array block (LAB) contents can not be changed after the HardCopy Timing Optimization Wizard is run

An example of a well partitioned design using LogicLock regions effectively for some portions of the design is shown in [Figure 6-1](#). Only the most critical logic functions required are placed in LogicLock regions in order to achieve the desired performance in the HardCopy Stratix

device. The dark blue rectangles shown in [Figure 6–1](#) are the user-assigned LogicLock regions that have fixed locations. In this example, the design needed to be constrained by LogicLock regions first inside the `HARDCOPY_FPGA_PROTOTYPE` with **Reserve Unused Logic** turned off in **Properties** in LogicLock regions. This selection allows the Quartus II software to isolate and compact the logic of these blocks in the `HARDCOPY_FPGA_PROTOTYPE` such that the placement is tightly controlled in the HardCopy Stratix device.

**Figure 6–1. A Well Partitioned Design**



In the example shown in [Figure 6–1](#), once suitable locations were identified for LogicLock regions, the LogicLock region properties were changed from floating to locked. The Quartus II software can then reproduce their placement in subsequent compilations, while focusing attention on fixing other portions of the design.

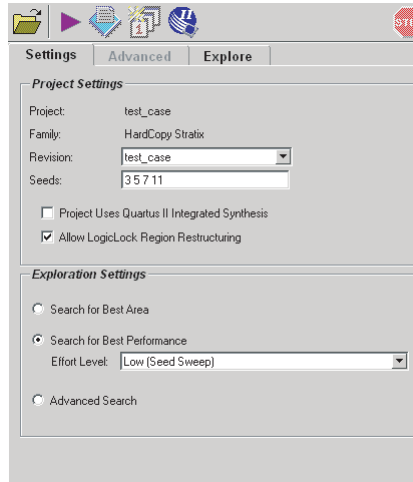
## Using Design Space Explorer for HardCopy Stratix Designs

The DSE feature in the Quartus II software allows you to evaluate various compilation settings to achieve the best results for your FPGA designs. DSE can also be used in the HardCopy Stratix project after running the HardCopy Timing Optimization wizard.

Only some of the DSE settings affect HardCopy Stratix designs because HDL synthesis and physical optimization have been completed on the FPGA. No logic restructuring can occur after using the HardCopy Timing Optimization wizard. When you compile your design, the placement of LABs is optimized in the HardCopy Stratix device. To access the DSE GUI

in your open project in the Quartus II software, select **Launch Design Space Explorer** (Tools menu). An example of the DSE GUI and DSE Settings window for the HardCopy Stratix device is shown in [Figure 6–2](#).

**Figure 6–2. DSE Settings Window in the DSE GUI**



## Recommended DSE Settings for HardCopy Stratix Designs

The HardCopy Stratix design does not require all advanced settings or effort-level settings in DSE. Altera recommends using the following settings in DSE for HardCopy Stratix designs:

- In the **Settings** tab ([Figure 6–2](#)), make the following selections:
  - Under **Project Settings**, enter several seed numbers in the **Seeds** box. Each seed number requires one full compile of the HardCopy Stratix project.
  - Under **Project Settings**, select **Allow LogicLock Region Restructuring**.
  - Under **Exploration Settings**, select **Search for Best Performance**, and select **Low (Seed Sweep)** from the **Effort Level** menu.
- Turn on **Archive all Compilations** (Options menu).

After running DSE with the seed sweep setting, view the results and identify which seed settings produced the best compilation results. Use the archive of the identified seed, or merge the compilation settings and seed number from the DSE archived project into your primary HardCopy Stratix project.

## Performance Improvement Example

With the design used for the performance improvement example in this section, the designer was seeking performance improvement on an HC1S30F780 design for an intellectual property (IP) core consisting of approximately 5200 LEs, 75,000 bits of memory, and two digital signal processing (DSP) multiplier accumulators (MACs). The final application needed to fit in a reserved portion of the HC1S30 device floorplan, so the entire block of IP was initially bounded in a single LogicLock region. The IP block was evaluated as a stand-alone block.

### Initial Design Example Settings

The default settings in the Quartus II software version 4.2 were used, with the following initial constraints added:

- The device was set to the target Stratix FPGA device which is the prototype for the HC1S30F780 device:  

```
set_global_assignment -name DEVICE  
EP1S30F780C6_HARDCOPY_FPGA_PROTOTYPE
```
- A LogicLock region was created for the block to bound it in the reserved region.
- The LogicLock region properties were set to **Auto Size** and **Floating Location**, and **Reserve Unused Logic** was turned on:  

```
set_global_assignment -name LL_STATE FLOATING  
set_global_assignment -name LL_AUTO_SIZE ON  
set_global_assignment -name LL_RESERVED OFF  
set_global_assignment -name LL_SOFT OFF
```
- Virtual I/O pins were used for the ports of the core since this core does not interface to pins in the parent design, and the I/O pins were placed outside the LogicLock region and are represented as registers in LEs.

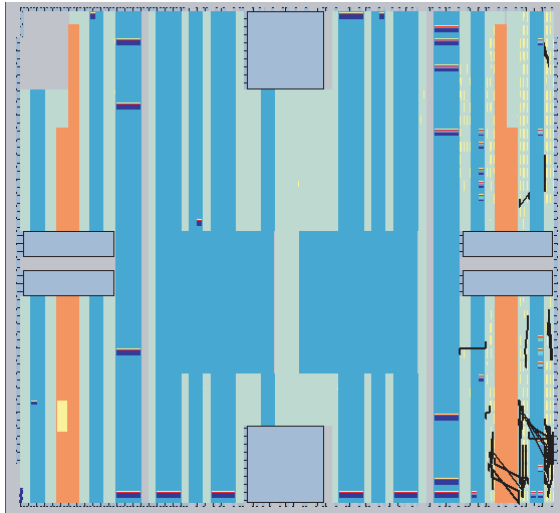
The initial compilation results yielded 65.30-MHz  $f_{MAX}$  in the FPGA. The block was constrained through virtual I/O pins and a LogicLock region to keep the logic from spreading throughout the floorplan.

The initial compile-relevant statistics for this example are provided in [Table 6-1](#).

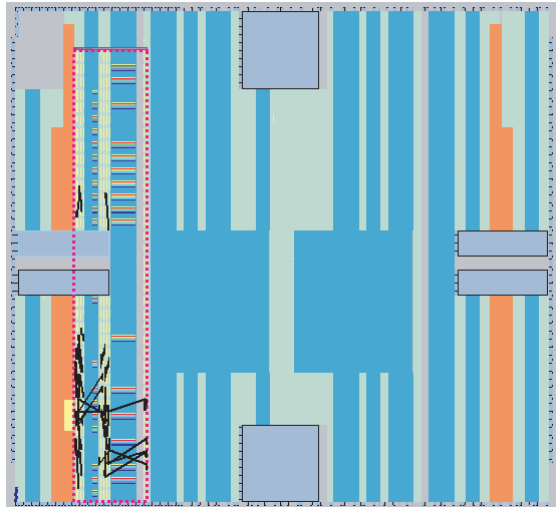
**Table 6-1. Initial Compilation Statistics**

Result Type	Results
$f_{MAX}$	65.30 MHz
Total logic elements (LEs)	5,187/32,470 (15%)
Total LABs	564/3,247 (17%)
M512 blocks	20/295 (6%)
M4K blocks	16/171 (9%)
M-RAM blocks	0/2 (0%)
Total memory bits	74,752/2,137,536 (3%)
Total RAM block bits	85,248/2,137,536 (3%)
DSP block 9-bit elements	2/96 (2%)

The design project was migrated to the HardCopy device using the HardCopy Timing Optimization wizard and was compiled. The default settings of the LogicLock region in a HardCopy Stratix project in the Quartus II software have the **Soft Region** option turned on. With this setting, the HardCopy Stratix compilation yields an  $f_{MAX}$  of 66.48 MHz, mainly due to the Fitter placement being scattered in an open design ([Figure 6-3](#)). Because the **Soft Region** is set to on, the LogicLock region is not bounded. This is not an optimal placement in the HardCopy Stratix design and is not the best possible performance.

**Figure 6–3. HardCopy Stratix Device Floorplan with Soft Region On**

To keep the LogicLock region contents bounded in the final placement in the HardCopy Stratix device floorplan, turn off the **Soft Region** option. After turning off the **Soft Region** option and compiling the HardCopy Stratix design, the result is an  $f_{MAX}$  of 88.14 MHz—a gain of 33% over the Stratix FPGA device performance. The bounded placement in the LogicLock region helps to achieve performance improvement in well-partitioned design blocks by taking advantage of the smaller die size and custom metal routing interconnect of the HardCopy Stratix device. The floorplan of the bounded LogicLock region is visible in [Figure 6–4](#). In this figure, you can see the difference in disabling the Soft Region setting in the HardCopy Stratix design.

**Figure 6–4. HardCopy Stratix Device Floorplan with Soft Region Off**

## Using Analysis and Synthesis Settings for Performance Improvement

After establishing the baseline for improvement for this design of 65.30 MHz FPGA/88.14 MHz HardCopy, you can gain additional performance improvement in the Stratix FPGA and HardCopy Stratix devices using the available features in the Quartus II software.

Changing the **Analysis & Synthesis Effort** from **Balanced** to **Speed** yields additional benefit in performance, but at the cost of additional LE resources. The Tcl command for this assignment is as follows:

```
set_global_assignment -name  
STRATIX_OPTIMIZATION_TECHNIQUE SPEED
```

The relevant compilation results of the FPGA are provided in [Table 6-2](#).

<b>Result Type</b>	<b>Results</b>
$f_{MAX}$	68.88 MHz
Total logic elements	5,508/32,470 (16%)
Total LABs	598/3,247 (18%)
M512 blocks	20/295 (6%)
M4K blocks	16/171 (9%)
M-RAM blocks	0/2 (0%)
Total memory bits	74,752/2,137,536 (3%)
Total RAM block bits	85,248/2,137,536 (3%)
DSP block 9-bit elements	2/96 (2%)

Increasing the LE resources by 6% only yielded an additional 3 MHz in performance in the FPGA, without using additional settings. However, after migrating this design to the HardCopy Stratix design and compiling it, the performance did not improve over the previous HardCopy Stratix design compile, and was slightly worse in performance at 87.34 MHz. This shows that the Quartus II software synthesis was very effective with the **Synthesis Effort Level** set to **Balanced**, and there was only marginal improvement in the FPGA when this option was set to **Speed**.

The next settings activated in this example were the **Synthesis Netlist Optimizations** shown below in Tcl format for WYSIWYG synthesis remapping and gate-level retiming after synthesis mapping:

```
set_global_assignment -name
ADV_NETLIST_OPT_SYNTH_WYSIWYG_REMAP ON

set_global_assignment -name
ADV_NETLIST_OPT_SYNTH_GATE_RETIME ON
```

Making these settings in the FPGA while leaving **Analysis & Synthesis Effort** set to **Speed** yielded some additional improvement in the FPGA as shown in [Table 6–3](#).

**Table 6–3. Results of Analysis & Synthesis Effort Set to Speed**

Result Type	Results
$f_{MAX}$	70.28 MHz
Total logic elements	5,515/32,470 (16%)
Total LABs	597/3,247 (18%)

The WYSIWYG resynthesis added a minimal increase in LEs over the speed setting, and the design performance improved by 2 MHz in the FPGA. Using the HardCopy Timing Optimization wizard to migrate the design to HardCopy and subsequently compiling the HardCopy Stratix design, we find that performance is not improved beyond previous compiles, with an  $f_{MAX}$  of 86.58 MHz.

The Quartus II software automatically optimizes state machines and restructures multiplexers when these settings are set to **Auto** in the **Analysis & Synthesis** settings. Changing these options from **Auto** usually does not yield performance improvement.

For example, changing the multiplexer restructuring and state machine processing settings from both set to **Auto**, to **On** and **One-Hot**, respectively, actually hurt performance, not allowing the Quartus II software to determine the optimization on a case-by-case basis. With these settings, the FPGA compiled to an  $f_{MAX}$  of 65.99 MHz, and the HardCopy Stratix design only performed at 83.77 MHz. For this design example, it is better to leave these settings to **Auto** as seen in the Tcl assignments in the [“Using Fitter Assignments and Physical Synthesis Optimizations for Performance Improvement”](#) section, and allow the Quartus II software to determine when to use these features.

### Using Fitter Assignments and Physical Synthesis Optimizations for Performance Improvement

After exploring the Analysis & Synthesis optimization settings in the Quartus II software, you can use the Fitter Settings and Physical Synthesis Optimization features to gain further performance improvement in your Stratix FPGA and HardCopy Stratix devices. In this design example, multiplexer and state machine restructuring settings have been set to **Auto**, and the **Synthesis Optimization Technique** is set

for **Speed**. The **Fitter effort** is set to **Standard Fit (highest effort)**. The next features enabled are the **Physical Synthesis Optimizations** as seen in the Tcl assignments below and in [Figure 6-5](#):

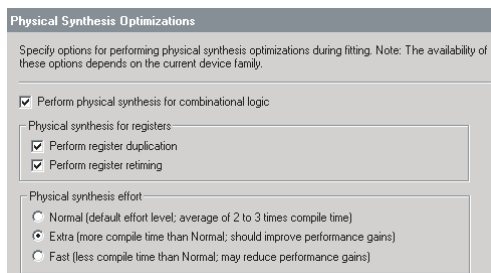
```
set_global_assignment -name
PHYSICAL_SYNTHESIS_COMBO_LOGIC ON

set_global_assignment -name
PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION ON

set_global_assignment -name
PHYSICAL_SYNTHESIS_REGISTER_RETIMING ON

set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT
EXTRA
```

**Figure 6-5. Physical Synthesis Optimization Settings**



The compiled design shows a performance increase in the FPGA, running at an  $f_{MAX}$  of 74.34 MHz, requiring additional LE resources as a result of the physical synthesis and logic duplication. In this example, you can see how performance can be increased in the Stratix FPGA device at the expense of additional LE resources, as this design’s LE resources grew almost 12% over the beginning compilation. The compiled FPGA design’s statistics are provided in [Table 6-4](#).

**Table 6-4. Compiled FPGA Design Statistics**

Result Type	Results
$f_{MAX}$	74.34 MHz
Total logic elements	5,781/32,470 (17%)
Total LABs	610/3,247 (18%)

Running the HardCopy Timing Optimization wizard on this design and compiling the HardCopy Stratix project yields an  $f_{MAX}$  of 92.01 MHz, a 24% improvement over the FPGA timing.

## Design Space Explorer

The available Fitter Settings produce an additional performance improvement. The DSE feature is used on the Stratix FPGA device to run through the various seeds in the design and select the best seed point to use for future compiles. This can often yield additional performance benefits as the Quartus II software further refines placement of the LEs and performs clustering of associated logic together.

For this design example, DSE was run with high effort (physical synthesis) and multiple placement seeds. Table 6–5 shows the DSE results. The base compile matches the fifth compile in the DSE variations, showing that the work already done on the design before DSE was optimal. The FPGA project was optimized before running DSE.

Compile Point	Clock Period: CLK	Logic Cells
Base (Best)	13.451 ns (74.34 MHz)	5,781
1	13.954 ns	5,703
2	13.712 ns	6,447
3	14.615 ns	5,777
4	13.911 ns	5,742
5	13.451 ns	5,781
6	14.838 ns	5,407
7	14.177 ns	5,751
8	14.479 ns	5,827
9	14.863 ns	5,596
10	14.662 ns	5,605
11	14.250 ns	5,710
12	14.016 ns	5,708
13	13.840 ns	5,802
14	13.681 ns	5,788
15	14.829 ns	5,644

Additional correlation is seen inside the `<project>.dse.rpt` file, showing the summary of assignments used for each compile inside the Quartus II software. The base compile settings and the fifth compile settings show good correlation, as shown in Table 6–6. The `MUX_RESTRUCTURE` setting did not have any effect on the design performance. This may be due to an already efficient HDL coding for multiplexer structures, requiring no optimization.

**Table 6–6. Base Compile and Fifth Compile Correlation**

Setting	New Value	Base Value
PHYSICAL_SYNTHESIS_REGISTER_RETIMING	ON	ON
SEED	1	1
STATE_MACHINE_PROCESSING	AUTO	AUTO
MUX_RESTRUCTURE	OFF	AUTO
PHYSICAL_SYNTHESIS_COMBO_LOGIC	ON	ON
FITTER_EFFORT	STANDARD FIT	STANDARD FIT
AUTO_PACKED_REGISTERS_STRATIX	NORMAL	NORMAL
PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION	ON	ON
ADV_NETLIST_OPT_SYNTH_GATE_RETIME	ON	ON
STRATIX_OPTIMIZATION_TECHNIQUE	SPEED	SPEED
PHYSICAL_SYNTHESIS_EFFORT	EXTRA	EXTRA

The information presented in Table 6–6 confirms that the FPGA Prototype device has been optimized as much as possible without manual floorplan adjustments.

### *Design Space Explorer for HardCopy Stratix Devices*

Migrating this compiled design to the HardCopy Stratix project and compiling the HardCopy Stratix design optimization, results in a design performance of 92.01 MHz. The next task is to run DSE on the HardCopy Stratix project using **Low Effort (Seed Sweep)** in the **Exploration Settings**, and entering a range of seed numbers with which to compile the project.

The results of the DSE run with the **Seed Sweep** option are summarized in [Table 6–7](#).

**Table 6–7. DSE Results Run with Seed Sweep**

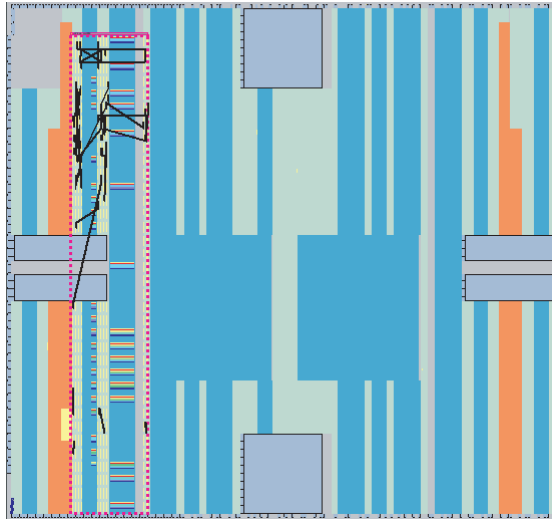
Compile Point	Clock Period: CLK
Base (Best)	10.868 ns
1	11.710 ns
2	11.040 ns
3	10.790 ns
4	10.945 ns
5	11.154 ns
6	11.707 ns
7	11.648 ns
8	11.476 ns
9	11.423 ns
10	11.449 ns

The results in [Table 6–7](#) illustrate how the **Seed Sweep** option in DSE provides additional improvement in the HardCopy Stratix design, even after DSE has been run on the Stratix FPGA project. In this example, compile point 3 using seed value = 4 turns out to be slightly beneficial over other seeds in the Fitter Placement. The HardCopy Stratix device has an  $f_{MAX}$  of 92.71 MHz.

## Back-Annotation and Location Assignment Adjustments

Another technique available for improving performance in the HardCopy Stratix design is manually adjusting placement and back-annotating location assignments from the placement results. These techniques should be one of the last steps taken for design optimization of HardCopy Stratix devices.

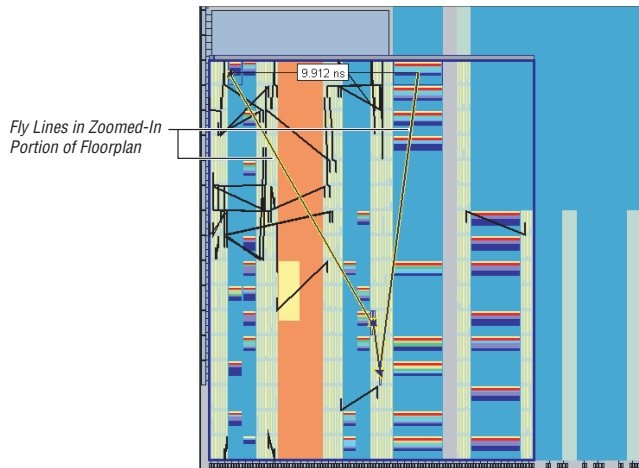
Observing the floorplan of the 92.71 MHz compile ([Figure 6–6](#)), the placement of the LogicLock region is stretched vertically, and additional improvement is possible if the aspect ratio of the LogicLock region is defined, and placement in it is refined.

**Figure 6–6. Vertically Stretched LogicLock Region**

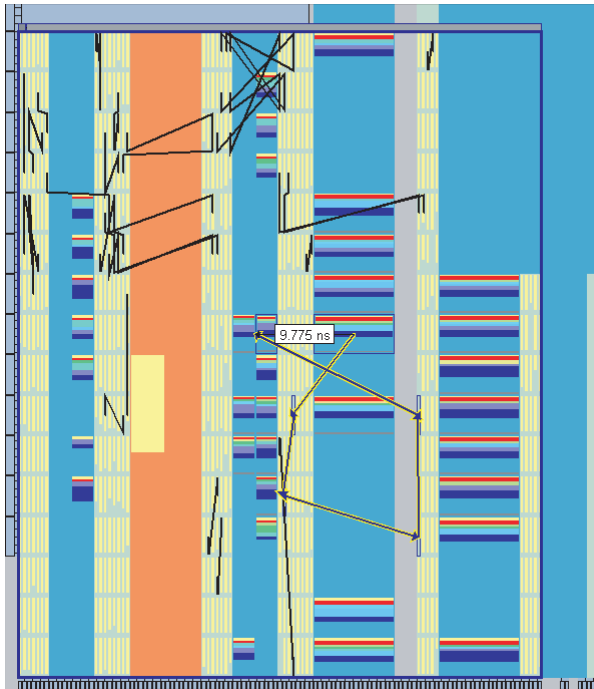
This floorplan would be better optimized if the LogicLock region had a more square shape, helping the paths that go from memory-to-memory, by containing the M4K and M512 memory blocks in a smaller space, and allowing LAB placement to be adjusted by the Fitter. In the HardCopy Stratix device, signals are routed between LABs, DSP blocks, and memory blocks using the customized metal layers. The reconfigurable routing tracks in the Stratix FPGA device limit the routing paths and delays between elements in the HardCopy Stratix device. This flexibility allows for aspect ratio changes in LogicLock regions, so the raw distance between points becomes the critical factor, and not the usage of available routing resources in the FPGA.

For the final placement optimization in this example, the LogicLock region was fixed in a square region that encompassed two columns of M4K blocks, four columns of M512 blocks, two columns of DSP blocks, and enough LABs to fit the remaining resources required. After compiling the design with these new LogicLock assignments, the performance increased to 93.46 MHz in the HardCopy Stratix device. The critical path and LogicLock region location can be seen in the zoomed-in area of the floorplan (Figure 6–7).

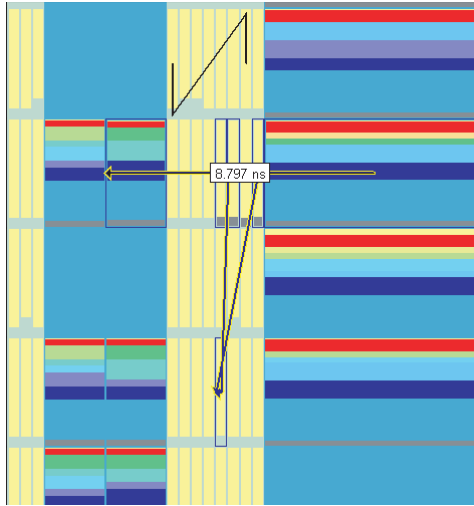
You can see in Figure 6–7 that the critical path shown is from an M4K block to an M512 block through several levels of logic. The placement of the memory blocks can be optimized manually, since the LogicLock region contains more memory blocks than necessary.

**Figure 6–7. Critical Path and LogicLock Region**

Using the critical path “fly lines” as a guide for placement optimization, manual location assignments were made for some of the M512 and M4K instances used in the design. The resulting compile improved the  $f_{MAX}$  to 94.67 MHz. The new critical path (Figure 6–8) shows how placement of all path elements are confined to a much smaller area. As a result, the routing distances and delays are smaller through the path.

**Figure 6–8. New Critical Path**

Examining this new critical path placement, you can see that there is room for further performance improvement through additional location assignments. The current slowest path is 9.775 ns of delay. Manually moving the LABs in this critical path and placing them between the M4K and M512 endpoints, and subsequently recompiling, shows improved results not only for this path, but for several other paths, as this path contained a major timing bottleneck. The critical path between this start and endpoint was reduced to 8.797 ns (Figure 6–9). However, the entire design only improved to 100.30 MHz because other paths are now the slowest paths in the design. This illustrates that fixing one major bottleneck path can raise the entire design performance since one high fanout node can affect multiple timing paths, as was the case in this example.

**Figure 6–9. Improved Results**

In summary, this design example started with 65.30 MHz in the Stratix FPGA device, and was improved to 74.34 MHz. It was then taken from the Stratix FPGA device compile and improved to 100.30 MHz in the HardCopy Stratix design, for a performance improvement of 35%.

## Conclusion

Using performance-optimization techniques specifically for HardCopy Stratix devices can achieve significant performance improvement over the Stratix FPGA prototype device. Many of these changes must be incorporated up-front in the `HARDCOPY_FPGA_PROTOTYPE` so that your design is properly prepared for performance improvement after running the HardCopy Timing Optimization wizard.

The example discussed in this chapter demonstrates the process for performance improvement and various features in the Quartus II software available for use when optimizing your Stratix FPGA prototype and HardCopy Stratix device. It also demonstrates the importance of planning ahead for the HardCopy Stratix design implementation while continuing to work in the `HARDCOPY_FPGA_PROTOTYPE` design if you are going to seek performance improvement in the HardCopy Stratix device.

## Document Revision History

Table 6–8 shows the revision history for this chapter.

<b>Date and Document Version</b>	<b>Changes Made</b>	<b>Summary of Changes</b>
September 2008 v1.4	Updated chapter number and metadata.	—
June 2007 v1.3	<ul style="list-style-type: none"><li>• Updated the “Background Information” section.</li><li>• Completed minor typographical updates.</li></ul>	—
December 2006 v1.2	Updated revision history.	—
March 2006	Formerly chapter 21; no content change.	—
October 2005 v1.1	<ul style="list-style-type: none"><li>• Updated graphics</li><li>• Minor edits</li></ul>	—
July 2005 v1.0	Initial release of Chapter 21, Design Guidelines for HardCopy Stratix Performance Improvement.	—