

Introduction

The Quartus® II software offers physical synthesis optimizations to optimize your design beyond the optimization performed in the normal course of the Quartus II compilation flow. The effect of these options depends on the structure of your design. Physical synthesis optimizations can help improve the performance of your design, regardless of the synthesis tool used.

Netlist optimization options work with your design's atom netlist, which describes a design in terms of Altera®-specific primitives. An atom netlist file can take the form of an Electronic Design Interchange Format (.edf) file or a Verilog Quartus Mapping (.vqm) file generated by a third-party synthesis tool, or a netlist used internally by the Quartus II software. Physical synthesis optimizations are applied at different stages of the Quartus II compilation flow, either during synthesis, fitting, or both.

This chapter explains how the physical synthesis optimizations in the Quartus II software can modify your design's netlist and help improve your quality of results. This chapter also provides information about preserving your compilation results through back-annotation and writing out a new netlist, and provides guidelines for applying the various options.



Because the node names for primitives in the design can change when you use the physical synthesis option, evaluate whether your design flow requires fixed node names. If you use a verification flow that might require fixed node names, such as the SignalTap® II Embedded Logic Analyzer, formal verification, or the LogicLock based optimization flow (for legacy devices), you must turn off the synthesis netlist optimization and physical synthesis options.

WYSIWYG Primitive Resynthesis

If you use a third-party tool to synthesize your design, use the **Perform WYSIWYG primitive resynthesis** option to apply optimizations to the synthesized netlist.

The **Perform WYSIWYG primitive resynthesis** option directs the Quartus II software to un-map the logic elements (LEs) in an atom netlist to logic gates, and then re-map the gates back to Altera-specific primitives. Third-party synthesis tools generate an atom netlist file that specifies Altera-specific primitives. Atom netlist files can take the form of either an .edf or .vqm file generated by the third-party synthesis tool. When you turn on this option, the Quartus II software can work on different techniques specific to the device architecture during the re-mapping process. This feature re-maps the design using the **Optimization Technique** specified for your project (**Speed, Area, or Balanced**).

This option has no effect if you are using Quartus II integrated synthesis to synthesize your design.

To turn on this option, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.

2. In the **Category** list, select **Analysis and Synthesis Settings**. The **Analysis & Synthesis Settings** page appears.
3. Select **Perform WYSIWYG Primitive Resynthesis**. Turn on the setting and click **OK**.

If you want to perform WYSIWYG resynthesis on only a portion of your design, you can use the Assignment Editor to assign the **Perform WYSIWYG primitive resynthesis** logic option to a lower-level entity in your design. This option can be used with Arria® II GX, Arria GX, HardCopy® series, Stratix® series, Cyclone® series, MAX® II, or APEX™ series device families.

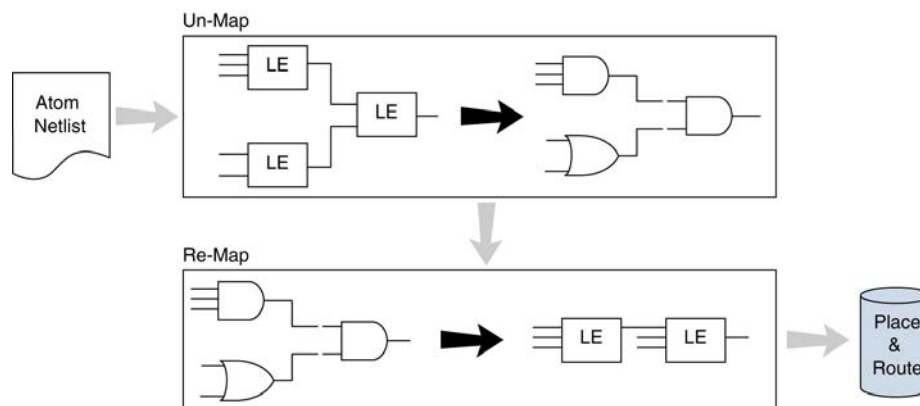
The results of the remapping are dependent on the **Optimization Technique** chosen for your project. To select the **Optimization Technique**, perform the following steps:

1. In the **Category** list, select **Analysis & Synthesis Settings**. The **Analysis & Synthesis Settings** page appears.
2. Under **Optimization Technique**, select from **Speed**, **Area**, or **Balanced** to specify how the Quartus II technology mapper optimizes the design. The **Balanced** setting is the default for many Altera device families; this setting optimizes the timing critical parts of the design for speed and the rest for area.
3. Click **OK**.

Refer to the *Quartus II Integrated Synthesis* chapter in volume 1 of the *Quartus II Handbook* for details on the Optimization Technique option.

Figure 13-1 shows the Quartus II software flow for this feature.

Figure 13-1. WYSIWYG Primitive Resynthesis




The **Perform WYSIWYG primitive resynthesis** option is not applicable if you are using Quartus II integrated synthesis to synthesize your design because the design does not contain Altera primitives.


The **Perform WYSIWYG primitive resynthesis** option unmaps and remaps only logic cells, also referred to as LCELL or LE primitives, and regular I/O primitives (which may contain registers). Double data rate (DDR) I/O primitives, memory primitives, digital signal processing (DSP) primitives, and logic cells in carry/cascade chains are not touched. Logic specified in an encrypted **.vqm** file or an **.edf** file, such as third-party intellectual property (IP), is not touched.

Turning on this option can cause drastic changes to the node names in the `.vqm` file or `.edf` file from your third-party synthesis tool, because the primitives in the atom netlist are broken apart and then remapped within the Quartus II software. Registers can be minimized away and duplicates removed, but registers that are not removed have the same name after remapping.

Any nodes or entities that have the **Netlist Optimizations** logic option set to **Never Allow** are not affected during WYSIWYG primitive resynthesis. To apply this logic option, on the Assignments menu, click **Assignment Editor**. This option disables WYSIWYG resynthesis for parts of your design.

 Primitive node names are specified during synthesis. When netlist optimizations are applied, node names might change because primitives are created and removed. HDL attributes applied to preserve logic in third-party synthesis tools cannot be maintained because those attributes are not written into the atom netlist read by the Quartus II software.

If you use the Quartus II software to synthesize, you can use the **Preserve Register (preserve)** and **Keep Combinational Logic (keep)** attributes to maintain certain nodes in the design.


 For more information about using these attributes during synthesis in the Quartus II software, refer to the *Quartus II Integrated Synthesis* chapter in volume 1 of the *Quartus II Handbook*.

Performing Physical Synthesis Optimizations

Traditionally, the Quartus II design flow has involved separate steps of synthesis and fitting. The synthesis step optimizes the logical structure of a circuit for area, speed, or both. The Fitter then places and routes the logic cells to ensure critical portions of logic are close together and use the fastest possible routing resources. While you are using this push-button flow, the synthesis stage is unable to anticipate the routing delays seen in the Fitter. Because routing delays are a significant part of the typical critical path delay, performing synthesis operations with physical delay knowledge allows the tool to target its timing-driven optimizations at these parts of the design. This tight integration of the fitting and synthesis processes is known as physical synthesis.

The following sections describe the physical synthesis optimizations available in the Quartus II software, and how they can help improve your performance results. Physical synthesis optimization options can be used with Arria II GX, Arria GX, the Stratix and Cyclone series device families, and HardCopy ASIC series.

If you are migrating your design to a HardCopy II device, you can target physical synthesis optimizations to the FPGA architecture in the FPGA-first flow or to the HardCopy II architecture in the HardCopy-first flow. The optimizations are mapped to the other device architecture during the migration process.

 You cannot target optimizations to optimize for both device architectures individually because doing so would result in a different post-fitting netlist for each device.

 For more information about using physical synthesis with HardCopy devices, refer to the *Quartus II Support of HardCopy Series Devices* chapter in volume 1 of the *Quartus II Handbook*.

You can choose the physical synthesis optimization options you want for your design during synthesis and fitting in the **Physical Synthesis Optimizations** page. The settings include optimizations for improving performance and fitting in the selected device.

You can also set the effort level for physical synthesis optimizations. Normally, physical synthesis optimizations increase the compilation time. Choose the **Fast** option if you want to reduce the impact on compilation time. When you choose this effort level, the Quartus II software performs limited register retiming operations during fitting. The **Extra** option runs additional algorithms to get the best possible results, with maximum impact on the compilation time.

To optimize the performance, the following options are available:

- **Perform physical synthesis for combinational logic**
- **Perform register retiming**
- **Perform automatic asynchronous signal pipelining**
- **Perform register duplication**

To optimize for better fitting, you can choose from the following options:

- **Perform physical synthesis for combinational logic**
- **Perform logic to memory mapping**

To view and modify the physical synthesis optimization options, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, click the “+” icon to expand **Compilation Process Settings** and select **Physical Synthesis Optimizations**. The **Physical Synthesis Optimizations** page appears.
3. Specify the options for performing physical synthesis optimizations.

Some physical synthesis options affect only registered logic and some options affect only combinational logic. Select options based on whether you want to keep the registers intact or not. For example, if your verification flow involves formal verification, you might have to keep the registers intact.

You can control the effect of physical synthesis with the **Effort Level** option. The default selection is **Normal**. The **Extra** effort setting uses extra compilation time to try to achieve extra circuit performance, while the **Fast** effort setting uses less compilation time than **Normal** but might not achieve the same gains.

All Physical Synthesis optimizations write results to the **Netlist Optimizations** report. To access this report, on the Processing menu, click **Compilation Report**. In the **Compilation Report** list, click the “+” icon to expand **Fitter** and select **Netlist Optimizations**. This report provides a list of atom netlist files that were modified, created, and deleted during physical synthesis. Similarly, physical synthesis optimizations performed during synthesis write results to the synthesis report. To access this report, on the Processing menu, click **Compilation Report**. In the **Compilation Report** list, select **Analysis & Synthesis**.

Nodes or entities that have the **Netlist Optimizations** logic option set to **Never Allow** are not affected by the physical synthesis algorithms. To access this logic option, on the Assignments menu, click **Assignment Editor**. Use this option to disable physical synthesis optimizations for parts of your design.

Automatic Asynchronous Signal Pipelining

The **Perform automatic asynchronous signal pipelining** option on the **Physical Synthesis Optimizations** page in the **Compilation Process Settings** section of the **Settings** dialog box allows the Quartus II Fitter to perform automatic insertion of pipeline stages for asynchronous clear and asynchronous load signals during fitting when these signals negatively affect performance. You can use this option if asynchronous control signal recovery and removal times are not achieving their requirements.

This option improves performance for designs in which asynchronous signals in very fast clock domains cannot be distributed across the chip fast enough due to long global network delays. This optimization performs automatic pipelining of these signals, while attempting to minimize the total number of registers inserted.



The **Perform automatic asynchronous signal pipelining** option adds registers to nets driving the asynchronous clear or asynchronous load ports of registers. This adds register delays (adds latency) to the reset, adding the same number of register delays for each destination using the reset. Thus, changing the behavior of the signal in the design. Use this only if adding latency to reset signals does not violate any design requirements. This option also prevents the promotion of signals to global routing resources.

The Quartus II software performs automatic asynchronous signal pipelining only if **Enable Recovery/Removal analysis** is enabled. If you use the TimeQuest Timing Analyzer, **Enable Recovery/Removal analysis** is turned on by default. Pipelining is allowed only on asynchronous signals that have the following properties:

- The asynchronous signal is synchronized to a clock (a synchronization register drives the signal)
- The asynchronous signal fans-out only to asynchronous control ports of registers

To use **Enable Recovery/Removal analysis** with the Classic Timing Analyzer, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, click the “+” icon to expand **Timing Analysis Setting** and select **Classic Timing Analyzer Settings**. The **Classic Timing Analyzer Settings** page appears.
3. Click **More Settings**. The **More Timing Settings** dialog box appears.
4. In the **Name** list, select **Enable Recovery/Removal analysis**. In the **Setting** list, select **On**.
5. Click **OK**.
6. Click **OK**.

The Quartus II software does not perform automatic asynchronous signal pipelining on asynchronous signals that have the **Netlist Optimization** logic option set to **Never Allow**.

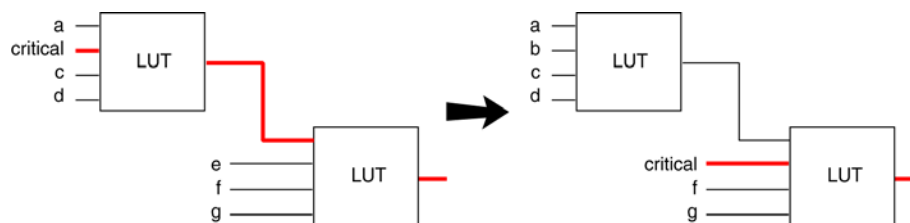
Physical Synthesis for Combinational Logic

To optimize the design and reduce delay along the critical paths, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, click the “+” icon to expand **Compilation Process Settings** and select **Physical Synthesis Optimizations**. The **Physical Synthesis Optimizations** page appears.
3. Turn on **Perform physical synthesis for combinational logic** by checking the appropriate box.

The software can accomplish this type of optimization by swapping the look-up table (LUT) ports within LEs so that the critical path has fewer layers through which to travel. See [Figure 13-2](#) for an example. This option also allows the duplication of LUTs to enable further optimizations on the critical path.

Figure 13-2. Physical Synthesis for Combinational Logic



In the first case, the critical input feeds through the first LUT to the second LUT. The Quartus II software swaps the critical input to the first LUT with an input feeding the second LUT. This reduces the number of LUTs contained in the critical path. The synthesis information for each LUT is altered to maintain design functionality.

The **Physical synthesis for combinational logic** option affects only combinational logic in the form of LUTs. These transformations might occur during the synthesis stage or the Fitter stage during compilation. The registers contained in the affected logic cells are not modified. Inputs into memory blocks, DSP blocks, and I/O elements (IOEs) are not swapped.

The Quartus II software does not perform combinational optimization on logic cells that have the following properties:

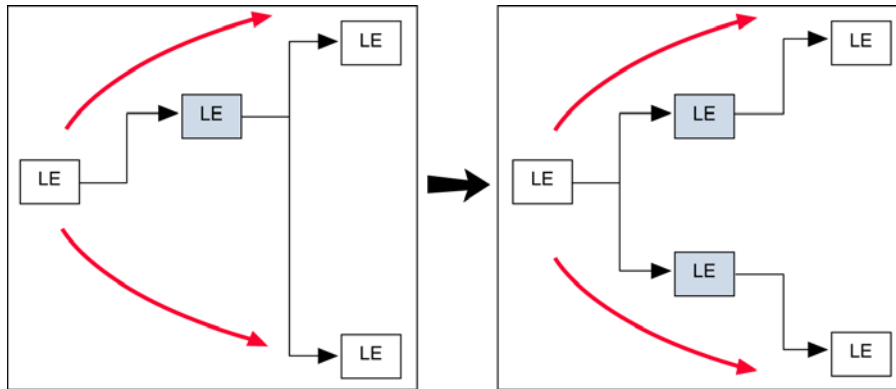
- Are part of a chain
- Drive global signals
- Are constrained to a single logic array block (LAB) location
- Have the **Netlist Optimizations** option set to **Never Allow**

If you consider logic cells with any of these conditions for physical synthesis, you can override these rules by setting the **Netlist Optimizations** logic option to **Always Allow** on a given set of nodes.

Physical Synthesis for Registers—Register Duplication


The **Perform register duplication** option on the **Physical Synthesis Optimizations** page in the **Compilation Process Settings** section of the **Settings** dialog box allows the Quartus II Fitter to duplicate registers based on Fitter placement information. You can also duplicate combinational logic when this option is enabled. A logic cell that fans out to multiple locations can be duplicated to reduce the delay of one path without degrading the delay of another. The new logic cell can be placed closer to critical logic without affecting the other fan-out paths of the original logic cell. [Figure 13-3](#) shows an example of register duplication.

Figure 13-3. Register Duplication



The Quartus II software does not perform register duplication on logic cells that have the following properties:

- Are part of a chain
- Contain registers that drive asynchronous control signals on another register
- Contain registers that drive the clock of another register
- Contain registers that drive global signals
- Contain registers that are constrained to a single LAB location
- Contain registers that are driven by input pins without a t_{SU} constraint
- Contain registers that are driven by a register in another clock domain
- Are considered virtual I/O pins
- Have the **Netlist Optimizations** option set to **Never Allow**

 For more information about virtual I/O pins, refer to the *Analyzing and Optimizing the Design Floorplan* chapter in volume 2 of the *Quartus II Handbook*.

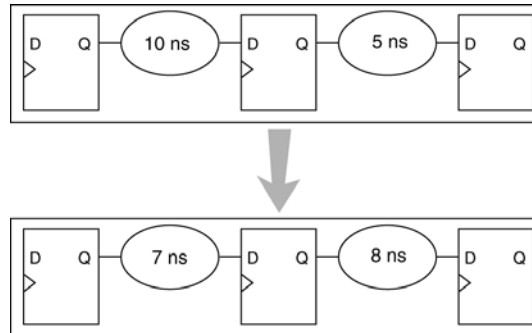
If you want to consider logic cells that meet any of these conditions for physical synthesis, you can override these rules by setting the **Netlist Optimizations** logic option to **Always Allow** on a given set of nodes.

Physical Synthesis for Registers—Register Retiming

The Register Retiming option enables the movement of registers across combinational logic, allowing the Quartus II software to trade off the delay between timing critical paths and non-critical paths. Register Retiming can be done during Quartus II integrated synthesis or during the Fitter stages of design compilation.

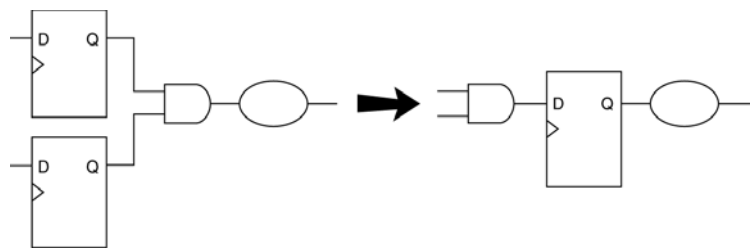
Figure 13-4 shows an example of register retiming in which the 10-ns critical delay is reduced by moving the register relative to the combinational logic.

Figure 13-4. Register Retiming Diagram



Retiming can create multiple registers at the input of a combinational block from a register at the output of a combinational block. In this case, the new registers have the same clock and clock enable. The asynchronous control signals and power-up level are derived from previous registers to provide equivalent functionality. Retiming can also combine multiple registers at the input of a combinational block to a single register (Figure 13-5).

Figure 13-5. Combining Registers with Register Retiming



To move registers across combinational logic to balance timing, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, click the “+” icon to expand **Compilation Process Settings** and select **Physical Synthesis Optimizations**. The **Physical Synthesis Optimizations** page appears.
3. Specify your preferred option under **Physical synthesis for performance and Effort level**.
4. Click **OK**.

If you want to prevent register movement during register retiming, you can set the **Netlist Optimizations** logic option to **Never Allow**. This option can be applied either to individual registers or entities in the design using the Assignment Editor.

In digital circuits, synchronization registers are instantiated on cross clock domain paths to reduce the possibility of metastability. The Quartus II software detects such synchronization registers and they are not moved, even if register retiming is turned on.

The following sets of registers are not moved during register retiming:


- Both registers in a direct connection from input pin-to-register-to-register if both registers have the same clock and the first register does not fan-out to anywhere else. These registers are considered synchronization registers.
- Both registers in a direct connection from register-to-register if both registers have the same clock, the first register does not fan-out to anywhere else, and the first register is fed by another register in a different clock domain (directly or through combinational logic). These registers are considered synchronization registers.

By default, the Quartus II software assumes that a synchronization register chain consists of a set of two registers. If your design has synchronization register chains that are longer than two, you must indicate the number of registers in your synchronization chains, so that those are not affected by when register retiming is enabled. To do this, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, select **Analysis & Synthesis Settings**. The **Analysis & Synthesis Setting** page appears.
3. Click **More Settings**. The **More Analysis & Synthesis Settings** dialog box appears.
4. In the **Name** list, select **Synchronization Register Chain Length** and modify the setting to match the synchronization register length used in your design. If you set a value of 1 for the **Synchronization Register Chain Length**, it means that any registers connected to the first register in a register-to-register connection can be moved during retiming. A value of $n > 1$ means that any registers in a sequence of length 1, 2, ... n are not moved during register retiming.

In general, the Quartus II software does not perform register retiming on logic cells that have the following properties:

- Are part of a cascade chain
- Contain registers that drive asynchronous control signals on another register
- Contain registers that drive the clock of another register
- Contain registers that drive a register in another clock domain
- Contain registers that are driven by a register in another clock domain
- Contain registers that are constrained to a single LAB location
- Contain registers that are connected to SERDES
- Are considered virtual I/O pins
- Registers that have the **Netlist Optimizations** logic option set to **Never Allow**


 For more information about virtual I/O pins, refer to the *Analyzing and Optimizing the Design Floorplan* chapter in volume 2 of the *Quartus II Handbook*.

If you want to consider logic cells that meet any of these conditions for physical synthesis, you can override these rules by setting the **Netlist Optimizations** logic option to **Always Allow** on a given set of registers.

Preserving Your Physical Synthesis Results

The Quartus II software generates the same results on every compilation for the same source code and settings on a given system. Therefore, it is typically not necessary to take steps to preserve your results from compilation to compilation. When changes are made to the source code or to the settings, you usually get the best results by allowing the software to compile without using previous compilation results or location assignments. In some cases, if you avoid performing analysis and synthesis or `quartus_map`, and run the Fitter or another desired Quartus II executable instead, you can skip the synthesis stage of the compilation.

When you use the Quartus II incremental compilation flow, you can preserve synthesis results for a particular partition of your design by choosing a netlist type of post-synthesis. If you want to preserve fitting results between compilation runs, choose a netlist type of post-fit during incremental compilation. The rest of this section is relevant only for those designs using older devices that do not support incremental compilation.

 For information about the incremental compilation design methodology, refer to the *Quartus II Incremental Compilation for Hierarchical and Team Based Design* chapter in volume 1 of the *Quartus II Handbook*.

You can preserve the resulting nodes from physical synthesis in older devices that do not support incremental compilation. Preserving the nodes might be required if you use the LogicLock flow to back-annotate placement and/or import one design into another. Use this only for older devices such as the APEX or ACEX family. If you use older devices (such as the APEX or ACEX series) with LogicLock as an optimization strategy, you might have to use this option to preserve the nodes resulting from netlist optimizations. For newer devices, use incremental compilation to preserve results. When using the incremental compilation flow, you do not have to use this option even if you have LogicLock regions in the design.

To preserve the nodes from Quartus II physical synthesis optimization options for older devices that do not support incremental compilation, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
2. In the **Category** list, select **Compilation Process Settings**. The **Compilation Process Settings** page appears.
3. Turn on **Save a node-level netlist of the entire design into a persistent source file**. This setting is not available for Cyclone III, Stratix III, and newer devices.
4. Click **OK**.

The **Save a node-level netlist of the entire design into a persistent source file** option saves your final results as an atom-based netlist in **.vqm** file format. By default, the Quartus II software places the **.vqm** file in the **atom_netlists** directory under the current project directory. To create a different **.vqm** file using different Quartus II settings, in the **Compilation Process Settings** page, change the **File name** setting.

If you use synthesis netlist optimizations (and not physical synthesis optimizations), generating a **.vqm** file is optional. To lock down the location of all logic and device resources in the design with or without a Quartus II-generated **.vqm** file, on the Assignments menu, click **Back-Annotate Assignments** and specify the desired options. You should use back-annotated location assignments unless the design has been finalized. Making any changes to the design invalidates your back-annotated location assignments. If you require changes later, use the new source HDL code as your input files, and remove the back-annotated assignments corresponding to the old code or netlist.

If you create a **.vqm** file to recompile the design, use the new **.vqm** file as the input source file and turn off the synthesis netlist optimizations for the new compilation.

If you use the physical synthesis optimizations and want to lock down the location of all LEs and other device resources in the design with the **Back-Annotate Assignments** command, a **.vqm** file netlist is required. The **.vqm** file preserves the changes that were made to your original netlist. Because the physical synthesis optimizations depend on the placement of the nodes in the design, back-annotating the placement changes the results from physical synthesis. Changing the results means that node names are different, and your back-annotated locations are no longer valid.

You should not use a Quartus II-generated **.vqm** file or back-annotated location assignments with physical synthesis optimizations unless the design has been finalized. Making any changes to the design invalidates your physical synthesis results and back-annotated location assignments. If you require changes later, use the new source HDL code as your input files, and remove the back-annotated assignments corresponding to the Quartus II-generated **.vqm** file.

To back-annotate logic locations for a design that was compiled with physical synthesis optimizations, first create a **.vqm** file. When recompiling the design with the hard logic location assignments, use the new **.vqm** file as the input source file and turn off the physical synthesis optimizations for the new compilation.

If you are importing a **.vqm** file and back-annotated locations into another project that has any **Netlist Optimizations** turned on, it is important to apply the **Never Allow** constraint to make sure node names don't change; otherwise, the back-annotated location or LogicLock assignments are invalid.



For newer devices, such as the Stratix, Cyclone, or Arria GX series, use incremental compilation to preserve compilation results instead of using logic back-annotation.

Physical Synthesis Options for Fitting

The Quartus II software enables you to use the physical synthesis option to improve the fitting results. To access these settings, perform the following steps:

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.

- In the **Category** list, click the “+” icon to expand **Compilation Process Settings** and select **Physical Synthesis Optimizations**. The **Physical Synthesis Optimizations** page appears.
- Under **Optimize for fitting** (physical synthesis for density), there are two physical synthesis options available to improve fitting your design in the target device: **Physical synthesis for combinational logic** and **Perform logic to memory mapping** (Table 13-1).

Table 13-1. Physical Synthesis Optimizations Options

Option	Function
Physical Synthesis for Combinational Logic	When you select this option, the Fitter detects duplicate combinational logic and optimizes combinational logic to improve the fit.
Perform Logic to Memory Mapping	When you select this option, the Fitter can remap registers and combinational logic in your design into unused memory blocks and achieves a fit.

Applying Netlist Optimization Options

Netlist optimizations options can have various effects on different designs. Designs that are well coded or have already been restructured to balance critical path delays might not see a noticeable difference in performance.

To obtain optimal results when using netlist optimization options, you might have to vary the options applied to find the best results. By default, all options are off. Turning on additional options leads to the largest effect on the node names in the design. Take this into consideration if you are using a LogicLock-based optimization strategy (for older devices such as APEX and ACEX), or verification flow, such as the SignalTap II Embedded Logic Analyzer or formal verification that requires fixed or known node names. Applying all of the physical synthesis options at the **Extra** effort level generally produces the best results for those options, but adds significantly to the compilation time. You can also use the **Physical synthesis effort level** options to decrease the compilation time.

Typically, the WYSIWYG primitive resynthesis does not add much compilation time relative to the overall design compilation time.

To find the best results, you can use the Quartus II Design Space Explorer (DSE) to apply various sets of netlist optimization options.




For more information about using DSE, refer to the *Design Space Explorer* chapter in volume 2 of the *Quartus II Handbook*.

Scripting Support

You can run procedures and make settings described in this chapter in a Tcl script. You can also run some procedures at a command prompt. For detailed information about scripting command options, refer to the Quartus II Command-Line and Tcl API Help browser. To run the Help browser, type the following command at the command prompt:

```
quartus_sh --qhhelp ←
```

 The *Scripting Reference Manual* includes the same information in PDF form. For more information about Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*. Refer to the *Quartus II Settings File Reference Manual* for information about all settings and constraints in the Quartus II software. For more information about command-line scripting, refer to the *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*.

You can specify many of the options described in this section either on an instance, or at a global level, or both.

Use the following Tcl command to make a global assignment:

```
set_global_assignment -name <QSF variable name> <value> ←
```

Use the following Tcl command to make an instance assignment:

```
set_instance_assignment -name <QSF variable name> <value> \  
-to <instance name> ←
```

Synthesis Netlist Optimizations

Table 13-2 lists the Quartus II Settings File (.qsf) variable names and applicable values for the settings discussed in “WYSIWYG Primitive Resynthesis” on page 13-1. The .qsf file variable name is used in the Tcl assignment to make the setting along with the appropriate value. The **Type** column indicates whether the setting is supported as a global setting, an instance setting, or both.

Table 13-2. Synthesis Netlist Optimizations and Associated Settings

Setting Name	Quartus II Settings File Variable Name	Values	Type
Perform WYSIWYG Primitive Resynthesis	ADV_NETLIST_OPT_SYNTH_WYSIWYG_REMAP	ON, OFF	Global, Instance
Optimization Technique	<Device Family Name>_OPTIMIZATION_TECHNIQUE	AREA, SPEED, BALANCED	Global, Instance
Power-Up Don't Care	ALLOW_POWER_UP_DONT_CARE	ON, OFF	Global
Save a node-level netlist into a persistent source file	LOGICLOCK_INCREMENTAL_COMPILE_ASSIGNMENT	ON, OFF	Global
	LOGICLOCK_INCREMENTAL_COMPILE_FILE	<filename>	
Allow Netlist Optimizations	ADV_NETLIST_OPT_ALLOWED	"ALWAYS ALLOW", DEFAULT, "NEVER ALLOW"	Instance

Physical Synthesis Optimizations

Table 13-3 lists the .qsf file variable name and applicable values for the settings discussed in “Performing Physical Synthesis Optimizations” on page 13-3. The .qsf file variable name is used in the Tcl assignment to make the setting, along with the appropriate value. The **Type** column indicates whether the setting is supported as a global setting, an instance setting, or both.

Table 13-3. Physical Synthesis Optimizations and Associated Settings

Setting Name	Quartus II Settings File Variable Name	Values	Type
Physical Synthesis for Combinational Logic	PHYSICAL_SYNTHESIS_COMBO_LOGIC	ON, OFF	Global
Automatic Asynchronous Signal Pipelining	PHYSICAL_SYNTHESIS_ASYNCHRONOUS_SIGNAL_PIPELINING	ON, OFF	Global
Perform Register Duplication	PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION	ON, OFF	Global
Perform Register Retiming	PHYSICAL_SYNTHESIS_REGISTER_RETIMING	ON, OFF	Global
Power-Up Don't Care	ALLOW_POWER_UP_DONT_CARE	ON, OFF	Global, Instance
Power-Up Level	POWER_UP_LEVEL	HIGH, LOW	Instance
Allow Netlist Optimizations	ADV_NETLIST_OPT_ALLOWED	"ALWAYS ALLOW", DEFAULT, "NEVER ALLOW"	Instance
Save a node-level netlist into a persistent source file	LOGICLOCK_INCREMENTAL_COMPILE_ASSIGNMENT	ON, OFF	Global
	LOGICLOCK_INCREMENTAL_COMPILE_FILE	<filename>	

Incremental Compilation

For information about scripting and command line usage for incremental compilation as mentioned in [“Preserving Your Physical Synthesis Results”](#) on page 13-10, refer to the *Quartus II Incremental Compilation for Hierarchical and Team-Based Design* chapter in volume 1 of the *Quartus II Handbook*.

Back-Annotating Assignments

You can use the `logiclock_back_annotate` Tcl command to back-annotate resources in your design. This command can back-annotate resources in LogicLock regions, and resources in designs without LogicLock regions.

For more information about back-annotating assignments, refer to [“Preserving Your Physical Synthesis Results”](#) on page 13-10.

The following Tcl command back-annotates all registers in your design:

```
logiclock_back_annotate -resource_filter "REGISTER"
```

The `logiclock_back_annotate` command is in the `backannotate` package.

Conclusion

Physical synthesis optimizations restructure and optimize your design netlist. Taking advantage of these Quartus II netlist optimizations can help improve your quality of results.

Referenced Documents

This chapter references the following documents:


- *Analyzing and Optimizing the Design Floorplan* chapter in volume 2 of the *Quartus II Handbook*
- *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*
- *Design Space Explorer* chapter in volume 2 of the *Quartus II Handbook*
- *Quartus II Incremental Compilation for Hierarchical and Team-Based Design* chapter in volume 1 of the *Quartus II Handbook*
- *Quartus II Integrated Synthesis* chapter in volume 1 of the *Quartus II Handbook*
- *Quartus II Settings File Reference Manual*
- *Quartus II Support for HardCopy Series Devices* chapter in volume 1 of the *Quartus II Handbook*
- *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*

Document Revision History

Table 13-4 shows the revision history for this chapter.

Table 13-4. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
March 2009 v9.0.0	<ul style="list-style-type: none"> ■ Was chapter 11 in the 8.1.0 release. ■ Updated the “Physical Synthesis for Registers—Register Retiming” and “Physical Synthesis Options for Fitting” ■ Updated “Performing Physical Synthesis Optimizations” ■ Deleted Gate-Level Register Retiming section. ■ Updated the referenced documents 	Updated GUI references and procedure steps, and document structure for the Quartus II software 9.0 release.
November 2008 v8.1.0	Changed to 8½” × 11” page size. No change to content.	Updated for the Quartus II 8.1 software release.
May 2008 v8.0.0	<p>Updated “Physical Synthesis Optimizations for Performance on page 11-9</p> <p>Added Physical Synthesis Options for Fitting on page 11-16</p>	Updated for Quartus II 8.0 version.

 For previous versions of the *Quartus II Handbook*, refer to the [Quartus II Handbook Archive](#).

