Contents

About this User Guide
Revision History ............................................................................................................... v
Referenced Documents ........................................................................................................ v
How to Contact Altera ........................................................................................................ vi
Typographic Conventions ................................................................................................ vi

Chapter 1. About this Megafunction
Device Family Support ...................................................................................................... 1–1
Introduction .................................................................................................................... 1–1
Features .......................................................................................................................... 1–1
General Description ....................................................................................................... 1–2
   Common Applications ................................................................................................. 1–3
      Clock Source Selection ............................................................................................ 1–3
      Clock Power-Down ................................................................................................. 1–3
   Global Clock Control Block ....................................................................................... 1–3
   Regional Clock Control Block ................................................................................. 1–4
   External PLL Output Clock Control Block ............................................................... 1–5
   Clock Enable Signals ............................................................................................... 1–6
   Clock Enable Timing ............................................................................................... 1–7
Resource Utilization and Performance ........................................................................... 1–7

Chapter 2. Getting Started
System and Software Requirements ................................................................................... 2–1
MegaWizard Plug-In Manager Customization ................................................................. 2–1
Using the MegaWizard Plug-In Manager ......................................................................... 2–2
Inferring Megafunctions from HDL Code .......................................................................... 2–8
Instantiating Megafunctions in HDL Code or Schematic Designs .................................... 2–8
   Generating a Netlist for EDA Tool Use ...................................................................... 2–9
   Using the Port and Parameter Definitions .................................................................. 2–9
Identifying a Megafunction after Compilation ................................................................. 2–9
Simulation ..................................................................................................................... 2–10
   Quartus II Simulation .............................................................................................. 2–10
   EDA Simulation ...................................................................................................... 2–10
SignalTap II Embedded Logic Analyzer .......................................................................... 2–11
Design Example: Clock Control Block ............................................................................ 2–11
   Design Files ........................................................................................................... 2–11
   Example .................................................................................................................. 2–11
   Generate a clkctrl_gclk Block ............................................................................. 2–11
   Implement the clkctrl_gclk Top .......................................................................... 2–17
   Functional Results—Simulate the clkctrl_gclk_top Design Using the Quartus II Software .......................................................................................................................................................... 2–20
Contents

Functional Results—Simulate the clkctrl_gclk_top Design Using ModelSim-Altera Tool .................................................. 2–21
Conclusion ........................................................................................................................................................................... 2–22

Chapter 3. Specifications
Ports and Parameters ......................................................................................................................................................... 3–1
Connectivity Restrictions .................................................................................................................................................. 3–3
  General Restrictions .................................................................................................................................................. 3–3
  Stratix II Restrictions ................................................................................................................................................ 3–4
About this User Guide

Revision History

The following table shows the revision history for this user guide.

<table>
<thead>
<tr>
<th>Date and Document Version</th>
<th>Changes Made</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 2007 v2.3</td>
<td>Updated for Quartus® II software version 7.1, including:</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>● Added information on Cyclone III and Arria GX device support</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>● Added Referenced Documents section</td>
<td>—</td>
</tr>
<tr>
<td>March 2007 v2.2</td>
<td>Added Cyclone® III device to list of supported devices.</td>
<td>Updated for Quartus II version 7.0 by adding support for Cyclone III device.</td>
</tr>
<tr>
<td>December 2006 v2.1</td>
<td>Updated device family support to include Stratix III.</td>
<td>—</td>
</tr>
<tr>
<td>October 2006 v2.0</td>
<td>● Updated for Quartus II version 6.0, including</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>● Screen shots</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>● Adding ModelSim section in Chapter 2</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>● General edit</td>
<td>—</td>
</tr>
<tr>
<td>September 2004 v1.0</td>
<td>Initial release</td>
<td>—</td>
</tr>
</tbody>
</table>

Referenced Documents

This user guide references the following documents:

- Recommended HDL Coding Styles chapter in volume 1 of the Quartus II Handbook
- Quartus II Integrated Synthesis chapter in volume 1 of the Quartus II Handbook
- Design Debugging Using the SignalTap II Embedded Logic Analyzer chapter in volume 3 of the Quartus II Handbook
How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

<table>
<thead>
<tr>
<th>Information Type</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical support</td>
<td><a href="http://www.altera.com/mysupport/">www.altera.com/mysupport/</a></td>
</tr>
<tr>
<td>Technical training</td>
<td><a href="http://www.altera.com/training/">www.altera.com/training/</a> <a href="mailto:custrain@altera.com">custrain@altera.com</a></td>
</tr>
<tr>
<td>Product literature</td>
<td><a href="http://www.altera.com/literature/">www.altera.com/literature/</a></td>
</tr>
<tr>
<td>Altera literature services</td>
<td><a href="mailto:literature@altera.com">literature@altera.com</a> (1)</td>
</tr>
<tr>
<td>FTP site</td>
<td>ftp.altera.com</td>
</tr>
</tbody>
</table>

Note to table:
(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown below.

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold Type with Initial Capital Letters</td>
<td>Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: <strong>Save As</strong> dialog box.</td>
</tr>
<tr>
<td>bold type</td>
<td>External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: <em>fMAX</em>, \qdesigns directory, d: drive, chiptrip.gdf file.</td>
</tr>
<tr>
<td>Italic Type with Initial Capital Letters</td>
<td>Document titles are shown in italic type with initial capital letters. Example: <strong>AN 75: High-Speed Board Design.</strong></td>
</tr>
<tr>
<td>italic type</td>
<td>Internal timing parameters and variables are shown in italic type. Examples: <em>tpA</em>, <em>n + 1</em>.</td>
</tr>
<tr>
<td>Variable names are enclosed in angle brackets (&lt; &gt;) and shown in italic type. Example: <em>&lt;file name&gt;</em>.</td>
<td></td>
</tr>
<tr>
<td>initial capital letters</td>
<td>Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the <strong>Options</strong> menu.</td>
</tr>
<tr>
<td>“Subheading Title”</td>
<td>References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: “Typographic Conventions.”</td>
</tr>
<tr>
<td>Courier type</td>
<td>Signal and port names are shown in lowercase Courier type. Examples: <em>d&lt;al, tdi, input</em>. Active-low signals are denoted by suffix <em>n</em>, e.g., <em>resetn</em>.</td>
</tr>
<tr>
<td>Anything that must be typed exactly as it appears is shown in Courier type. For example: <em>c:\qdesigna\tutorial\chiptrip.gdf</em>. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the <strong>AHDL</strong> keyword <strong>SUBDESIGN</strong>), as well as logic function names (e.g., <strong>TRI</strong>), are shown in Courier.</td>
<td></td>
</tr>
</tbody>
</table>
### About this User Guide

<table>
<thead>
<tr>
<th>Visual Cue</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1., 2., 3., and a., b., c., etc.</td>
<td>Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.</td>
</tr>
<tr>
<td>● ● ●</td>
<td>Bullets are used in a list of items when the sequence of the items is not important.</td>
</tr>
<tr>
<td>✔</td>
<td>The checkmark indicates a procedure that consists of one step only.</td>
</tr>
<tr>
<td>▼</td>
<td>The hand points to information that requires special attention.</td>
</tr>
<tr>
<td><img src="image" alt="CAUTION" /></td>
<td>A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.</td>
</tr>
<tr>
<td><img src="image" alt="WARNING" /></td>
<td>A warning calls attention to a condition or possible situation that can cause injury to the user.</td>
</tr>
<tr>
<td>❳</td>
<td>The angled arrow indicates you should press the Enter key.</td>
</tr>
<tr>
<td>【】</td>
<td>The feet direct you to more information about a particular topic.</td>
</tr>
</tbody>
</table>
Typographic Conventions
Chapter 1. About this Megafunction

Device Family Support

The altclkctrl megafunction supports the following target Altera® device families:

- Arria™ GX
- Stratix® III
- Stratix II
- Stratix II GX
- Cyclone® III
- Cyclone II
- HardCopy® II

Introduction

As design complexities increase, use of vendor-specific intellectual property (IP) blocks has become a common design methodology. Altera provides parameterizable megafunctions that are optimized for Altera device architectures. Using megafunctions instead of coding your own logic saves valuable design time. Additionally, the Altera-provided functions may offer more efficient logic synthesis and device implementation. You can scale the megafunction’s size by simply setting parameters.

Features

The altclkctrl megafunction implements a basic clock control block and offers additional features, which include:

- Supporting specification of the operation mode of the clock control block
- Supporting specification of the number of input clock sources
- Providing an active high clock enable control input
General Description

The `altclkctrl` megafunction is a clock control function provided by the Quartus® II MegaWizard® Plug-In Manager. The `altclkctrl` megafunction allows you to easily configure the clock control block in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices.

A clock control block is a dynamic clock buffer that allows you to enable and disable the clock network and dynamically switch between multiple sources to drive the clock network. Table 1–1 shows the clock control block and the devices that support it.

<table>
<thead>
<tr>
<th>Clock Control Block</th>
<th>Arria GX</th>
<th>Stratix III</th>
<th>Stratix II</th>
<th>Stratix II GX</th>
<th>Cyclone III</th>
<th>Cyclone II</th>
<th>HardCopy II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Clock Network</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Dual Regional Clock Network</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>Regional Clock Network</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>Dedicated External Clock Out Path</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>For Periphery Clock</td>
<td>—</td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

The global clock network allows a clock signal (or other global signal) to reach all parts of the chip with a similar amount of skew. Similarly, the regional clock network allows a signal to reach one quadrant of the chip (though half of the chip can be reached by driving two quadrants). The external clock-out path represents the clock path from the outputs of the PLL to the dedicated `PLL_OUT` pins. Figure 1–1 shows an `altclkctrl` megafunction block diagram.

**Figure 1–1. altclkctrl Block Diagram**

![Diagram of altclkctrl Block Diagram](image-url)
Common Applications

Use the altclkctrl megafunction to control the clock control block in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices. Each global and regional clock network has its own clock control block. The clock control block has the following two functions:

- Clock source selection (dynamic selection of global clocks)
- Clock power-down (dynamic clock enable or disable)

Clock Source Selection

When using the clock control block, you can select the dynamic clock source that drives the global clock network. However, only certain combinations of signal sources are supported, as described in “Global Clock Control Block” on page 1–3. You cannot select clock sources dynamically to drive the regional clock networks and the dedicated external clock-out path.

Clock Power-Down

The dynamic clock enable or disable feature allows internal logic control the clock network. When a clock network is powered down, all the logic fed by that clock network is not toggling, thereby reducing the overall power consumption of the device.

Global Clock Control Block

When a clock control block is configured to drive a global clock network, you can select the clock source statically, or you can control the selection dynamically by using internal logic to drive the multiplexer select inputs. When selecting the clock source statically, the clock source can be set to any of the inputs; for example, you can use the dedicated CLK pin, internal logic, and PLL outputs.

When selecting the clock source dynamically, you can select two PLL outputs (such as c0 or c1), a combination of clock pins, or PLL outputs. Figure 1–2 shows a clock control block and the possible sources that can drive the global clock network of a Stratix II device.
Regional Clock Control Block

When the clock control block is configured to drive a regional clock network, the clock source selection can be controlled only statically. Any of the inputs to the clock select multiplexer can be set as the clock source. Figure 1–3 shows a clock control block configured to drive a regional clock network in a Stratix II device.

Notes to Figure 1–2:
(1) These clock select signals can be dynamically controlled through internal logic only when the device is operating in user mode.
(2) These clock select signals can be set only through a configuration file and cannot be dynamically controlled during user-mode operation.
About this Megafunction

Figure 1–3. Regional Clock Control Block in Stratix II Devices

Notes to Figure 1–3:
(1) These clock select signals can be controlled only through a configuration file and cannot be dynamically controlled during user-mode operation.
(2) Only the \textit{CLKn} pins on the top and bottom of the device feed to regional clock control blocks.

The unused global and regional clock networks are powered down automatically in the configuration file generated by the Quartus II software. The dynamic clock enable feature allows the internal logic to control the power for the GCLK and RCLK networks. You can enable or disable the clock network with the altclkctrl megafunction.

External PLL Output Clock Control Block

When the clock control block is configured to drive the dedicated external clock out, the clock source selection can only be controlled statically. The PLL outputs can be set only as the clock source. Figure 1–4 shows a clock control block configured to drive a dedicated external clock out for a Stratix II device.
General Description

**Figure 1–4. External PLL Output Clock Control Block in Stratix II Devices**

![Diagram of PLL Output Clock Control Block]

**Notes to Figure 1–4:**

1. These clock select signals can be set only through the configuration file and cannot be dynamically controlled during user-mode operation.
2. The clock control block feeds to a multiplexer within the PLL_OUT pin’s I/O element (IOE). The PLL_OUT pin is a dual-purpose pin. Therefore, this multiplexer selects either an internal signal or the output of the clock control block.

**Clock Enable Signals**

In Stratix II devices, the clock enable signals are supported at the clock network level. This allows you to enable or disable the GCLK and RCLK networks, or the PLL_OUT pins, which is useful for applications that require a low power or sleep mode. Figure 1–5 shows how the ena clock enable signal is implemented.
About this Megafunction

**Figure 1-5. Clock Enable Implementation in Stratix II Devices**

Clock Enable Timing

Figure 1-6 shows a functional timing waveform example for a clock-output enable. Clock enable is synchronous with the falling edge of the input clock.

**Figure 1-6. Clock Enable Timing**

Resource Utilization and Performance

The altclkctrl megafunction is implemented with the dedicated clock control block. Each global clock, regional clock, and PLL external output clock has its own clock control block. Resource utilization and performance can be obtained from the MegaWizard and Compilation Results.

For more information about the clock control blocks, refer to the Clock Network and PLL chapter in the relevant device handbook.
Chapter 2. Getting Started

System and Software Requirements

The instructions in this section require the following software:

- For operating system support information, refer to:
  
  www.altera.com/support/software/os_support/oss-index.html

- Quartus® II software version 7.1 or later

MegaWizard Plug-In Manager Customization

You can use the MegaWizard® Plug-In Manager to specify the altclkctrl megafunction features and parameters for each clock control buffer in your design.

Start the MegaWizard Plug-In Manager in one of the following ways:

- On the Tools menu, click MegaWizard Plug-In Manager.
- When working in the Block Editor, from the Edit menu, click Insert Symbol as Block, or right-click in the Block Editor, point to Insert, and click Symbol as Block. In the Symbol dialog box, click MegaWizard Plug-In Manager.
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt: qmegawiz
Using the MegaWizard Plug-In Manager

This section provides descriptions for the options available in the altclkctrl megafunction wizard.

Figure 2–1 shows page 1 of the altclkctrl megafunction wizard. Select Create a new megafunction variation and click Next.

Figure 2–1. MegaWizard Plug-In Manager —altclkctrl (page 1)

Page 2a (Figure 2–2) is where you indicate which features or settings apply to the megafunction variation you are creating.
Table 2–1 describes the features or settings that appear on page 2a.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which device family will you be using?</td>
<td>Specify the device family for the clock control block.</td>
</tr>
<tr>
<td>Which type of output file do you want to create?</td>
<td>Choose between AHDL, VHDL, or VHDL Verilog.</td>
</tr>
<tr>
<td>What name do you want for the output file?</td>
<td>Specify the name of your output file.</td>
</tr>
</tbody>
</table>

Page 3 of the MegaWizard is where you specify the operation mode, number of clock inputs, the clock enable option, and whether to include an ena port in your megafunction. Figure 2–3 shows page 3 of the altclkctrl megafunction wizard.
Using the MegaWizard Plug-In Manager

Figure 2–3. MegaWizard Plug-In Manager—altclkctrl (page 3)
Table 2–2 shows which features or settings to apply to the clock control block. Use this table and hardware descriptions of the clock control block feature to determine appropriate settings for your design.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currently selected device family</td>
<td>Specify the device family for the clock control block.</td>
</tr>
</tbody>
</table>
| Clock Buffers                     | Options include:  
  * **Auto**—Allows the compiles to pick the best clock buffer, while other values restrict usage to only the given clock buffer.  
  * **For global clock**—Allows a clock signal to reach all parts of the chip with the same amount of skew; input port clkselect can be used to switch between the four clock inputs.  
  * **For dual regional clock - half chip**—Allows a clock signal to reach half of the chip by using two regional clocks to drive two quadrants; only one clock input is accepted. This option is not supported in Cyclone III and Cyclone II devices.  
  * **For regional clock - quarter chip**—Allows a clock signal to reach a quadrant of the chip; only one clock input is accepted. This option is not supported in Cyclone III and Cyclone II devices.  
  * **For external path**—Represents the clock path from the outputs of the PLL to the dedicated clock output pins; only one clock output is accepted.  
  * **For periphery clock**—Represents the clock path to a periphery clock (PCLK) network. PCLK networks are a collection of individual clock networks driven from the periphery of the Stratix III device. Clock outputs from the dynamic phase alignment (DPA) block, horizontal I/O pins, and internal logic can drive the PCLK networks. This option is supported in Stratix III devices only.  

  You can change the number of clock inputs only if you choose **Auto** or **For global clock**. |
| How many clock inputs would you like? | This option allows you to specify the number of input clock sources for the clock control block. You can specify as many as four clock inputs, depending on the device. Some connection restrictions may apply. Refer to the relevant device handbook for details. |
| Create 'ena' port to enable or disable the clock network driven by this buffer | This option allows you to select whether you want to create an active high clock enable signal to enable or disable the clock network. |
| How do you want to register the "ena" port? | This option allows you to register the ena port. If enabled, values are Falling edge of input clock, and Double register with input clock. This option is supported in Stratix III devices only. For additional information, refer to the Clock Networks and PLLs in Stratix III Devices chapter in the Stratix III Device Handbook. |
Starting on page 3 of the altclkctrl wizard, you can generate a sample simulation waveform or launch the Quartus II Help by selecting the Generate Sample Waveforms or Quartus II Megafunction Reference options from the Documentation button.

Page 4 of the MegaWizard lists the simulation model files required to properly simulate the generated design files (Figure 2–4). On this page you can enable the Quartus II software to generate a synthesis area timing estimation netlist for the megafuction to be used by third-party tools.

Figure 2–4. MegaWizard Plug-In Manager—altclkctrl [page 4]
Page 5 of the altclkctrl wizard shows a summary of the files generated by
the Megafunction Plug-In Manager (Figure 2–5). The gray check marks
indicate those files generated automatically, and the red check marks are
optional files you select.

Figure 2–5. MegaWizard Plug-In Manager—altclkctrl (page 5), Summary
Inferring Megafunctions from HDL Code

Synthesis tools, including the Quartus II integrated synthesis, recognize certain types of HDL code and automatically infer the appropriate megafunction for an aspect of particular design when a megafunction will provide optimal results for that design. In other words, the Quartus II software uses the Altera® megafunction code when compiling your design, even though you may not specifically instantiate the megafunction. The Quartus II software infers megafunctions because they are optimized for Altera devices, so the area, performance, or both may be better than generic HDL code. Additionally, you must use megafunctions to access certain Altera architecture-specific features (such as memory, DSP blocks, and shift registers) that generally provide improved performance compared with basic logic elements.

Refer to the Recommended HDL Coding Styles chapter in volume 1 of the Quartus II Handbook for specific information about your particular megafunction.

Instantiating Megafunctions in HDL Code or Schematic Designs

When you use the MegaWizard Plug-In Manager to customize and parameterize a megafunction, it creates a set of output files that allow you to instantiate the customized function in your design. Depending on the language you choose in the MegaWizard Plug-In Manager, the MegaWizard instantiates the megafunction with the correct parameter values and generates a megafunction variation file (wrapper file) in Verilog HDL (.v), VHDL (.vhd), or AHDL (.tdf), along with other supporting files.

The MegaWizard Plug-In Manager provides options to create the following files:

- A sample instantiation template for the language of the variation file (_inst.v, _inst.vhd, or _inst.tdf)
- Component Declaration File (.cmp) that can be used in VHDL Design Files
- ADHDL Include File (.inc) that can be used in Text Design Files (.tdf)
- Quartus II Block Symbol File (.bsf) that can be used in schematic designs
- Verilog HDL module declaration file that can be used when instantiating the megafunction as a black box in a third-party synthesis tool (_bb.v)

For more information about the MegaWizard-generated files, refer to Quartus II Help or to the Recommended HDL Coding Styles chapter in volume 1 of the Quartus II Handbook.
Generating a Netlist for EDA Tool Use

If you use a third-party EDA synthesis tool, you can instantiate the megafunction variation file as a black box for synthesis. Use the VHDL component declaration or Verilog module declaration black box file to define the function in your synthesis tool, and then include the megafunction variation file in your Quartus II project.

If you enable the option to generate a synthesis area and timing estimation netlist in the MegaWizard Plug-In Manager, the MegaWizard generates an additional netlist file (_syn.v). The netlist file is a representation of the customized logic used in the Quartus II software. The file provides the connectivity of the architectural elements in the megafunction but may not represent true functionality. This information enables certain third-party synthesis tools to better report area and timing estimates. In addition, synthesis tools can use the timing information to focus timing-driven optimizations and improve the quality of results.

For more information about using megafunctions in your third-party synthesis tool, refer to the appropriate chapter in the Synthesis section in volume 1 of the Quartus II Handbook.

Using the Port and Parameter Definitions

Instead of the MegaWizard Plug-In Manager, you can instantiate the megafunction directly in your Verilog HDL, VHDL, or AHDL code by calling the megafunction and setting its parameters as you would any other module, component, or subdesign.

Altera strongly recommends that you use the MegaWizard Plug-In Manager for complex megafunctions. The MegaWizard Plug-In Manager ensures that you set all megafunction parameters properly.

Refer to Chapter 3, Specifications for a list of the megafunction ports and parameters.

Identifying a Megafuncti on after Compilation

During compilation with the Quartus II software, analysis and elaboration is performed to build the structure of your design. You can locate your megafunction in the Project Navigator window by expanding the compilation hierarchy and locating the megafunction by its name.

Similarly, to search for node names within the megafunction (using the Node Finder), in the Look in box, browse to locate the megafunction in the Hierarchy box.
Simulation

The Quartus II Simulation tool provides an easy-to-use, integrated solution for performing simulations. The following sections describe the simulation options.

Quartus II Simulation

With the Quartus II Simulator, you can perform two types of simulations: functional and timing. A functional simulation in the Quartus II program allows you to verify the logical operation of your design without taking into consideration the timing delays in the FPGA. This simulation is performed using only your RTL code. When performing a functional simulation, you add only signals that exist before synthesis. You can find these signals with the Registers: pre-synthesis, Design Entry, or Pin filters in the Node Finder. The top-level ports of megafunctions are found using these three filters.

In contrast, timing simulation in the Quartus II software verifies the operation of your design with annotated timing information. This simulation is performed using the post place-and-route netlist. When performing a timing simulation, you add only signals that exist after place and route. These signals are found with the Post-Compilation filter of the Node Finder. During synthesis and place and route, the names of your RTL signals will change. Therefore, it might be difficult to find signals from your megafunction instantiation in the Post-Compilation filter. However, if you want to preserve the names of your signals during the synthesis and place-and-route stages, you must use the synthesis attributes keep or preserve. These are Verilog and VHDL synthesis attributes that direct analysis and synthesis to keep a particular wire, register, or node intact. You can use these synthesis attributes to keep a combinational logic node so you can observe the node during simulation.

More information about these attributes is available in the Quartus II Integrated Synthesis chapter in volume 1 of the Quartus II Handbook.

EDA Simulation

Depending on the third-party simulation tool you are using, refer to the appropriate chapter in the Simulation section in volume 3 of the Quartus II Handbook. The Quartus II Handbook chapters describe how to perform functional and gate-level timing simulations that include the megafunctions, with details on the necessary files and the directories where these files are located.
Getting Started

SignalTap II Embedded Logic Analyzer

The SignalTap® II embedded logic analyzer provides a non-intrusive method of debugging all Altera megafuntions in your design. With the SignalTap II embedded logic analyzer, you can capture and analyze data samples for the top-level ports of the Altera megafuntions in your design while your system is running at full speed.

To monitor signals from your Altera megafuntions, you must first configure the SignalTap II embedded logic analyzer in the Quartus II software, and then include the analyzer as part of your Quartus II project. The Quartus II software seamlessly embeds the analyzer along with your design in the selected device.

For more information about using the SignalTap II embedded logic analyzer, refer to the Design Debugging Using the SignalTap II Embedded Logic Analyzer chapter in volume 3 of the Quartus II Handbook.

Design Example: Clock Control Block

This chapter presents a design example that uses the altclkctrl megafuntion to select clock signals from PLL outputs and dedicated clock pins. This example uses the MegaWizard Plug-In Manager in the Quartus II software. As you progress through the wizard, each page is described in detail. When you are finished with this example, you can incorporate it into your overall project.

Design Files

The example design files are available with this user guide in the Quartus II Project section and in the User Guides section of the Altera website (www.altera.com).

Example

In this example, you perform the following:

- Generate a clkctrl_gclk block using the altclkctrl megafuntion in the MegaWizard Plug-In Manager
- Implement the clkctrl_gclk_top project by assigning the EP2S15F484 device to the project and compiling the project
- Simulate the clkctrl_gclk_top design

Generate a clkctrl_gclk Block

1. Unzip altclkctrl_DesignExample.zip to any working directory on your PC.
2. Open the project file \clkctrl\clkctrl_gclk_top.qar.
Design Example: Clock Control Block

3. Open the top-level file \clkctrl\clkctrl_gclk_top.bdf. This is an incomplete file that you complete as part of this example.

4. Double-click on a blank area in the Block Design File (.bdf).

5. In the Symbol window, click MegaWizard Plug-In Manager.

6. On page 1, select Create a new custom megafunction variation.

7. Click Next. Page 2a appears.

8. In the Which device family will you be using? list, select Stratix II.

9. Expand the I/O folder and select ALTCLKCTRL (Figure 2–6).

10. Under Which type of output file do you want to create?, select AHDL.
11. Name the output file `clkctrl_gclk`.

12. Click Next. Page 3 appears (Figure 2–7).

**Figure 2–7. MegaWizard Plug In Manager (page 3)**

13. Under the heading **Altclkctrl**, select **For global clock** from the drop-down menu.

14. In the **How many clock inputs would you like?** list, select 4 from the drop-down menu.
Design Example: Clock Control Block

15. Turn on Create ‘ena’ port to enable or disable the clock network driven by this buffer.

16. Click Next. Page 4 appears (Figure 2–8).

Figure 2–8. MegaWizard Plug In Manager (page 4)

17. Turn on Generate a netlist for synthesis area and timing estimation to enable the Quartus II software to generate a synthesis area timing estimation netlist for the megafunction to be used by third-party tools.

18. Click Next. Page 5 appears (Figure 2–9).
19. Select all available output files.

20. Click **Finish**. The `clkctrl_gclk` module is built.
Design Example: Clock Control Block

21. In the clkctrl_gclk_top.qpf file, in the Symbol window (Figure 2–10), click OK.

Figure 2–10. The Symbol Window

22. Move the mouse to place the clkctrl_gclk symbol between the input/output ports in the clkctrl_gclk_top.bdf. Click to place the symbol. You have now completed the design file, as shown in Figure 2–11.
23. On the File menu, click **Save**.

**Implement the clkctrl_gclk Top**

This section describes how to implement the clock control top design, assign the EP2S15F484C3 device to the project, and compile the project.

1. In the Quartus II software, on the Assignments menu, click **Settings**. The **Settings** dialog box appears (Figure 2–12).
2. In the **Category** list, select **Device**.

### Figure 2–12. Settings Dialog Box

3. In the **Family** list, select **Stratix II**.

4. Under **Available Devices**, select **EP2S15F484C3**.

5. Click **OK**.

6. To compile the design on the **Processing** menu, click **Start Compilation** or click the **Compilation** button on the toolbar.

7. When the **Full compilation was successful** message appears, click **OK**. (This message box might indicate some warnings; this is unimportant to the design.)
8. In the Compilation Report, expand the Fitter folder and click the Summary icon to view the resource usage (Figure 2–13).

Figure 2–13. Fitter Summary

9. To view the fMAX in the Compilation Report, expand Timing Analyzer and click Summary (Figure 2–14).

Figure 2–14. Timing Analyzer Summary
Design Example: Clock Control Block

Functional Results—Simulate the clkctrl_gclk_top Design Using the Quartus II Software

Next, simulate the design to verify the results. Set up the Quartus II Simulator by performing the following steps:

1. On the Processing menu, click **Generate Functional Simulation Netlist**.

2. When **Functional Simulation Netlist was successful** appears, click **OK**.

3. On the Assignments menu, click **Settings**.

4. In the **Category** list, select the **Simulator Settings**.

5. Under **Simulation mode**, select **Functional** and browse to the input vector waveform file (**clkctrl_gclk_top.vwf**).

6. Click **OK**.

7. On the Processing menu, click **Start Simulation**.

8. When the simulation completes, a **Simulator was successful** message appears. Click **OK**.

9. In the Simulation Report window, view the simulation output waveforms and verify the results. **Figure 2–15** shows the expected simulation results.

---

**Figure 2–15. Simulation Waveforms Using the Quartus II Software**
Functional Results—Simulate the clkctrl_gclk_top Design Using ModelSim-Altera Tool

Simulate the design in the ModelSim tool to compare the results of both simulators.

The following steps assume that you are familiar with using the ModelSim-Altera tool before trying out the design example. If you are unfamiliar with ModelSim-Altera, refer to www.altera.com/support/software/products/modelsim/mod-modelsim.html, a support page for ModelSim-Altera. There are links to topics such as installation, usage, and troubleshooting.

Set up the ModelSim-Altera simulator by performing the following steps:

1. Unzip the altclkctrl_msim.zip file to your PC.
2. Browse to the folder in which you unzipped the files and open the clkctrl_gclk_top.do file in a text editor.
3. In line 1 of the clkctrl_gclk_top.do file, replace <insert_directory_path_here> with the directory path of the appropriate library files. For example, C:/Modeltech_ae/altera/verilog/stratixii
4. On the File menu, click Save.
5. Start ModelSim-Altera.
7. Select the folder in which you unzipped the files. Click OK.
8. On the Tools menu, click Execute Macro.
9. Select the clkctrl_gclk_top.do file and click Open. This is a ModelSim script file that automates all necessary simulation settings.
10. Verify the results by looking at the Waveform Viewer window.

You may need to rearrange signals, remove redundant signals, and change the radix to suit the results in the Quartus II Simulator. Figure 2–16 shows the expected simulation results in ModelSim.
Conclusion

The Quartus II software provides parameterizable megafunctions ranging from simple arithmetic units, such as adders and counters, to advanced phase-locked loop (PLL) blocks, multipliers, and memory structures. These megafunctions are performance-optimized for Altera devices and therefore provide more efficient logic synthesis and device implementation because they automate the coding process and save valuable design time. Altera recommends using these functions during design implementation so you can consistently meet your design goals.
In this section, Figure 3–1 shows the ports and parameters for the altclkctrl megafunction. Table 3–1 describes the input ports, Table 3–2 describes the output ports, and Table 3–3 describes the altclkctrl megafunction parameters.

Refer to the latest version of the Quartus® II Help for the most current information about the ports and parameters for this megafunction.

The parameter details are only relevant for users who by-pass the MegaWizard® Plug-In Manager interface and use the megafunction as a directly parameterized instantiation in their design. The details of these parameters are hidden from MegaWizard Plug-In Manager interface users.

Figure 3–1. altclkctrl Port and Parameter Description Symbol
Table 3–1 describes the input ports of the altclkctrl megafuction.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>clkselect[]</td>
<td>No</td>
<td>Dynamically selects which clock source drives the clock network that is driven by this clock control block. The signal selects one of the four clock inputs in which a value of: ● 00 selects inclk[0] ● 01 selects inclk[1] ● 10 selects inclk[2] ● 11 selects inclk[3]</td>
<td>A 2-bit input port (clkselect[1:0]). This signal should only be connected if you want to dynamically select between multiple clock inputs. This signal defaults to GND if left unconnected. If this signal is connected, only the Global Clock network can be driven by this clock control block.</td>
</tr>
<tr>
<td>ena</td>
<td>No</td>
<td>Enables or disables the entire clock network driven by this clock control block.</td>
<td>If left unconnected, this signal defaults to Vcc (enabled). However, unused clock networks (no signals are the clock) will be disabled. The ena port cannot be used for the external clock output path in Cyclone® II devices.</td>
</tr>
<tr>
<td>inclk</td>
<td>Yes</td>
<td>Drive the clock network.</td>
<td>If more than one inclk[] signal is specified, you select between them with the clkselect[1:0] signal. The inclk[0] signal must be connected, while the other clock inputs should only be connected if you want to switch clock sources. Clock pins, clock outputs from the PLL, and core signals can drive the inclk[] input. However, only certain combinations of signal sources are supported. If more than one inclk[0] is connected, only the Global Clock network can be driven by this clock control block. In other words, the Dual Regional clock, Regional Clock and External Path network cannot be used.</td>
</tr>
</tbody>
</table>

Table 3–2 describes the output ports of the altclkctrl megafuction.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outclk</td>
<td>Yes</td>
<td>The clock output that drives the clock network associated with this clock control block. This parameter can drive either the Regional Clock network or Global Clock network as determined by the compiler.</td>
</tr>
</tbody>
</table>
Specifications

Table 3–3 describes the parameters for the altclkctrl megafunction.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCK_TYPE</td>
<td>String</td>
<td>Yes</td>
<td>Specifies the operation mode for the clock buffer. This parameter can specify auto-selected clocks, global clocks, regional clocks,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>external clock outputs, dual-regional clocks, or periphery clocks. Valid values are AUTO, Global Clock, Regional Clock, External Clock Output,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dual-Regional Clock, and Periphery Clock. If omitted, the default value is AUTO. When the CLOCK_TYPE parameter has the value EXTCLK, the clkselect[]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and ena ports are unavailable. The AUTO setting allows the compiler to pick the best clock control block to use, and other values restrict usage to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>specified clock network only. The Dual-Regional Clock type supports the use of two quadrants to reach half of the chip.</td>
</tr>
<tr>
<td>ENA_REGISTER_MODE</td>
<td>String</td>
<td>No</td>
<td>Specifies the register mode for the ena port. If this option is enabled, you can specify the value as FALLING_EDGE or DOUBLE_REGISTER. If not</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>enabled, the value is NONE. This parameter is available in Stratix® III devices only.</td>
</tr>
<tr>
<td>LPM_TYPE</td>
<td>String</td>
<td>No</td>
<td>Identifies the library of parameterized modules (LPM) entity name in VHDL Design Files.</td>
</tr>
</tbody>
</table>

Connectivity Restrictions

The following section describes the restrictions associated with the signal sources that can drive the inclk[] input.

General Restrictions

- The inclk[] ports that you use must be consistent with the clkselect[] ports that you use. For example, if the clkselect[0] ports are set to 0, inclk[0] must be used. If clkselect[0] is fed by anything other than 0 or 1 and clkselect[1] is set to 1, you must use either inclk[0] or inclk[1].
- If the clkselect[] ports are set to anything other than 0, only pins or PLL clock outputs may feed the inclk[] ports. In addition, pins must feed only inclk[0] or inclk[1], while PLL clock outputs must feed only inclk[2] or inclk[3].
- If the clock control block feeds any inclk[] port of another clock control block, both must be able to be reduced to a single clock control block of equivalent functionality.
Connectivity Restrictions

**Stratix II Restrictions**

- When connecting dedicated clock input pins to `inclk0x` and `inclk1x`, you must connect the low order CLK pin to `inclk0x` and the high order CLK pin to `inclk1x`. For example, if you are connecting CLK14p and CLK15p to a clock control block, CLK14p must connect to `inclk0x`, and CLK15p must connect to `inclk1x`. The Quartus II software has the ability in most cases to swap ports as required, but this can help solve errors that you may encounter in the Quartus II fitter.

- When connecting PLL output ports to `inclk2x` and `inclk3x`, you must connect the low order PLL port to `inclk2x` and the high order PLL port to `inclk3x`. For example, if you are connecting PLL output ports C0 and C1 to a clock control block, C0 must connect to `inclk2x`, and C1 must connect to `inclk3x`. Quartus II has the ability in most cases to swap ports as required, but this can help solve errors that you may encounter in the Quartus II fitter.

- Refer to the Clocking section of the PLL chapter in the handbook of the device family you are using for more information about valid connectivity configurations. Each global and regional resource has a clock control block, so you can use the information shown in the handbook to determine the valid dedicated clock pin and PLL output port resources which can feed any global or regional resource through a clock control block.

Table 3–4 summarizes which ports are usable by the different clock control block types. If a port is not available for a particular clock network, the clock control block placement is restricted to only the clock network types that support that port.

<table>
<thead>
<tr>
<th>Port</th>
<th>Global</th>
<th>Regional</th>
<th>External Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>inclk[3:0]</code></td>
<td>Has all four ports</td>
<td>Has only <code>inclk[0]</code></td>
<td>Has only <code>inclk[0]</code></td>
</tr>
<tr>
<td><code>clkselect[1:0]</code></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><code>ena</code></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>outclk</code></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

As shown, all clock buffer types support the clock-enable feature, while only the global clock network supports the dynamic clock source selection feature.

If the CLOCK_TYPE value is AUTO, the Quartus II software selects the clock control block type that meets all the requirements. For example, if you use the dynamic clock source selection and the clock control block
feeds a pin, a global clock control block is used. Similarly, if the clock control block feeds a pin that cannot be reached by the PLL using external clock output path (in other words, not a dedicated clock path), a global clock network is used. In this case, a warning message about the “irregular” path to the pin is issued.