Clock Control Block (ALTCLKCTRL) Megafunction

User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

UG-MF9604-3.1

Document last updated for Altera Complete Design Suite version: 13.0
Document publication date: February 2014

Feedback Subscribe

Contents

1. About this Megafunction

The clock control block (ALTCLKCTRL) megafunction is a clock control function provided by the Quartus® II MegaWizard™ Plug-In Manager to easily configure the clock control block in supported devices.

The common applications of using this megafunction are as follows:

- Dynamic 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 3–2. You cannot select clock sources dynamically to drive the regional clock networks and the dedicated external clock-out path.
- Dynamic power-down of a clock network—The dynamic clock enable or disable feature allows internal logic to power-down the clock network. When a clock network is powered-down, all the logic fed by that clock network is not toggling, thus the overall power consumption of the device is reduced.

Features

The ALTCLKCTRL megafunction provides the following additional features:

- Supports specification of operation mode of the clock control block
- Supports specification of the number of input clock sources
- Provides an active high clock enable control input

Device Support

The ALTCLKCTRL megafunction is available for the following devices:

- Arria® II GX
- Arria II GZ
- Arria V
- Arria V GZ
- Arria 10
- Cyclone® III
- Cyclone III LS
- Cyclone IV E
- Cyclone IV GX
- Cyclone V
- Stratix® III
- Stratix IV
- Stratix V

2. Parameter Settings

This section describes the parameter settings for the ALTCLKCTRL megafunction.You can parameterize the megafunction using the MegaWizard Plug-In Manager or the command-line interface (CLI). Altera recommends that you configure the megafunctions using the MegaWizard Plug-In Manager.

This user guide assumes that you are familiar with megafunctions and how to create them. If you are unfamiliar with Altera® megafunctions, refer to the *Introduction to Megafunctions User Guide*.

MegaWizard Parameter Settings

Table 2–1 provides descriptions of the options available on the individual pages of the ALTCLKCTRL MegaWizard Plug-In Manager.

Table 2–1. ALTCLKCTRL MegaWizard Plug-In Manager Page Options and Description (Part 1 of 4)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
1	Which action do you want to perform?	You can select from the following options: **Create a new custom megafunction variation**, **Edit an existing custom megafunction variation**, or **Copy an existing custom megafunction variation**.
2a	Select a megafunction from the list below	Select **ALTCLKCTRL** from the **I/O** category.
	Which device family will you be using?	Specify the device family that you want to use.
	Which type of output file do you want to create?	You can choose AHDL(**.tdf**), VHDL(**.vhd**), or Verilog HDL (**.v**) as the output file type.
	What name do you want for the output file?	Specify the name of the output file.
	Return to this page for another create operation	Turn on this option if you want to return to this page to create multiple megafunctions.
3	Currently selected device family	Specifies the device family you chose on page 2a.
	Match project/default	Turn on this option to ensure that the device selected matches the device family that is chosen in the previous page.

Table 2–1. ALTCLKCTRL MegaWizard Plug-In Manager Page Options and Description (Part 2 of 4)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
	How do you want to use the ALTCLKCTRL?	Specify the ALTCLKCTRL buffering mode. You can select from the following modes: **Auto** (1) —Allows the Compiler to pick the best clock buffer to use. **For global clock**—Allows a clock signal to reach all parts of the chip with the same amount of skew; you can select input port `clkselect` to switch between the four clock inputs. **For dual regional clock—half chip** (1), (6)—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. **For regional clock —quarter chip** (1), (6)—Allows a clock signal to reach a quadrant of the chip; only one clock input is accepted. **For regional clock**—This mode is available for Arria 10 devices only. Allows a clock to reach a region covering six interface tiles vertically (two at the edge of the device), and the entire chip horizontally. **For external path** (6)—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** (1)—Allows a clock signal to reach a quadrant or an octant of the chip depending on the device; only one clock input is accepted. For tile-based architectures like Arria 10 devices, a periphery clock will reach a region aligned with the source interface tile, and half the chip horizontally.
	How many clock inputs would you like? (2)	Specify the number of input clock sources for the clock control block. You can specify up to four clock inputs.
	Create ‘ena’ port to enable or disable the clock network driven by this buffer (1), (3)	Turn on this option if you want to create an active high clock enable signal to enable or disable the clock network.

Table 2–1. ALTCLKCTRL MegaWizard Plug-In Manager Page Options and Description (Part 3 of 4)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
	How do you want to register the 'ena' port?	Turn on this option to choose the register mode for cthe `ena` port. The available register modes are: **Falling edge of input clock**—the `clkout` pin is the output of the first falling edge driven register. **Double register with input clock**—the `clkout` pin is the output of the second falling edge driven register. **Not registered**—the `clkout` pin is set to one of the `clkin` input (after the multiplexer). This option is available after you turn on the **Create 'ena' port to enable or disable the clock network driven by this buffer** option. This option is available in Stratix III, Stratix IV, and Stratix V devices only.
3	Ensure glitch-free switchover implementation	Turn on this option to implement a glitch-free switchover when you use multiple clock inputs. You must ensure the clock that is currently selected is running before switching to another source. If the selected clock is not running, the glitch-free switchover implementation will not be able to switch to the new clock source. By default, the `clkselect` port is set to 00. A clock must be applied to `inclk0x` for the values on the `clkselect` ports to be read.
4	Generate netlist	Turn on this option if you want to generate a netlist for your third-party EDA synthesis tool to estimate the timing and resource usage of the megafunction. If you turn on this option, a netlist file (**_syn.v**) is generated. This file is a representation of the customized logic used in the Quartus II software and provides connectivity of the architectural elements in the megafunction but may not represent true functionality.

Table 2–1. ALTCLKCTRL MegaWizard Plug-In Manager Page Options and Description (Part 4 of 4)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
6	Summary Page	Specify the types of files to be generated. Only the files marked with red check marks are optional. Choose from the following types of files: ■ Variation file (4) ■ AHDL Include file (<*function name*>**.inc**) ■ VHDL component declaration file (<*function name*>.**cmp**) ■ Quartus II symbol file (<*function name*>**.bsf)** ■ Instantiation template file (<*function name*>**_inst.v**) ■ Verilog HDL black box file (<*function name*>**_bb.v)** ■ Synthesis area and timing estimation netlist (**_syn.v**) (5) For more information about the wizard-generated files, refer to Quartus II Help or to the *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*.

Notes to Table 2–1:

(1) This option is not supported in Cyclone III devices.
(2) You can change the number of clock inputs only if you choose the **Auto** or **For global clock** options.
(3) Not supported if you choose the **For periphery clock** option.
(4) The Variation file contains wrapper code in the language you specified on page 2a and is automatically generated.
(5) The synthesis area and timing estimation netlist file (**_syn.v**) is automatically generated if the **Generate netlist** option on page 4 is turned on.
(6) This mode is not supported in Arria 10 devices.

Command Line Interface Parameters

Expert users can choose to instantiate and parameterize the megafunction through the command-line interface using the clear box generator command. This method requires you to have command-line scripting knowledge.

For more information about using the clear box generator, refer to the *Introduction to Megafunctions User Guide*.

Table 2–2 lists the parameters for the ALTCLKCTRL megafunction.

Table 2–2. ALTCLKCTRL Megafunction Parameters

Parameter Name	Type	Required	Comments
clock_type	String	Yes	This parameter specifies the operation mode. The values are: **Value** / **Signal Selection** AUTO — Auto-selected clock (default value) GCLK — Global clock LCLK — Regional clock EXTCLK — External clock SIDE_CLK — Dual-regional clock The clkselect and ena ports are unavailable if the clock_type parameter is set to **EXTCLK**.
ena_register_mode	String	No	Register mode for the ena port. Values are **NONE**, **FALLING_EDGE**, and **DOUBLE_REGISTER**. Only available in Arria II GX, Arria II GZ, Arria V, Arria V GZ, Arria 10, Cyclone V, Stratix III, Stratix IV, and Stratix V devices.
lpm_hint	String	No	Allows you to specify Altera-specific parameters in VHDL Design Files (**.vhd**). The default value is **UNUSED**.
lpm_type	String	No	Identifies the library of parameterized modules (LPM) entity name in VHDL Design Files (**.vhd**).
intended_device_family	String	No	Used for modeling and behavioral simulation purposes. Create the ALTCLKCTRL megafunction with the MegaWizard Plug-In Manager to get the value for this parameter.
implement_in_les	String	No	Specifies if you want the clock control unit to be implemented using logic elements (LEs). Values are "**ON**" or "**OFF**". The default setting is "**OFF**".
number_of_clocks	Integer	Yes	Specifies the number of global-type clock inputs. Values are numeric type (1 to 4). For other clock types, only one clock input is accepted.
use_glitch_free_switch_over_implementation	String	No	Specifies if you want to implement a glitch-free switchover when you use multiple clock inputs. Values are "**ON**" and "**OFF**". If omitted, the default setting is "**OFF**".
width_select	Integer	Yes	Specifies the width of the clock select when you use multiple clock inputs. The clock select inputs dynamically selects the clock source that drives the clock network. The values are **1** or **2**. If omitted, the default value is **1**, which means 1-bit width.

3. Functional Description

This chapter describes the functional description and the design examples of the ALTCLKCTRL megafunction. This section also includes the prototype, component declarations, and the ports descriptions of the ALTCLKCTRL megafunction. You can use the ports to customize the ALTCLKCTRL megafunction according to your application.

Clock Control Block

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 3–1 shows the clock control block and the devices that support it.

Table 3–1. Clock Buffers that Drive the Clock Control Block

Clock Control Block	Arria 10	Arria V	Arria II GX	Stratix V	Stratix IV	Stratix III	Cyclone V	Cyclone IV	Cyclone III
Global Clock Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Dual Regional Clock Network	—	✓	✓	✓	✓	✓	✓	—	—
Regional Clock Network	✓	✓	✓	✓	✓	✓	✓	—	—
Dedicated External Clock Out Path	✓	✓	✓	✓	✓	✓	✓	✓	✓
For Periphery Clock	✓	✓	✓	✓	✓	✓	✓	—	—

The following table describes the clock control block.

Table 3–2. Clock Control Block

Clock Control Block	Description
Global Clock Network	Allows a clock signal (or other global signals) to reach all parts of the chip with a similar amount of skew.
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). For Arria 10 devices, the regional clock network drives a “sliding window” of SCLK regions corresponding to six interface tiles high.
External Clock-Out Path	Represents the clock path from the outputs of the phase-locked loop (PLL) to the dedicated PLL_OUT pins. The ALTCLKCTRL megafunction also provides glitch-free implementation for multiple clock input signals.

You must ensure the clock that is currently selected is running before switching to another source. If the selected clock is not running, the glitch-free switchover implementation will not be able to switch to the new clock source.

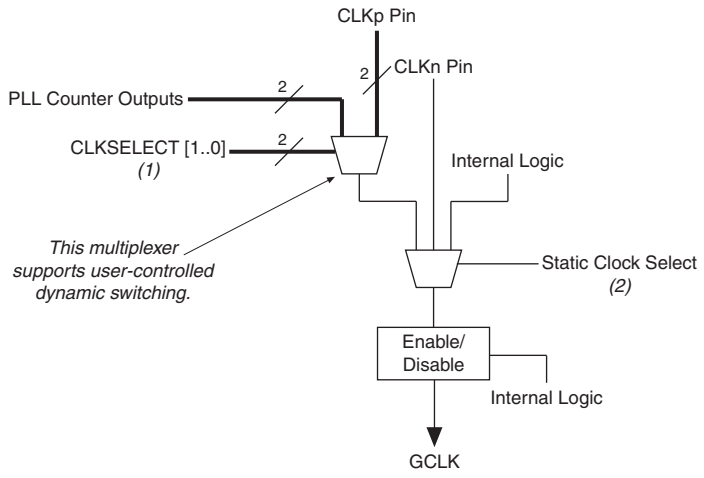
Global Clock Control Block

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 multiplexer selector inputs. When selecting the clock source statically, you can set the clock source 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 3–1 shows a clock control block and the possible sources that can drive the global clock network.

Figure 3–1. Global Clock Control Block



Notes to Figure 3–1:

(1) You can dynamically control these clock select signals through internal logic only when the device is operating in user mode.

(2) You can only set these clock select signals through a configuration file and cannot be dynamically controlled during user-mode operation.

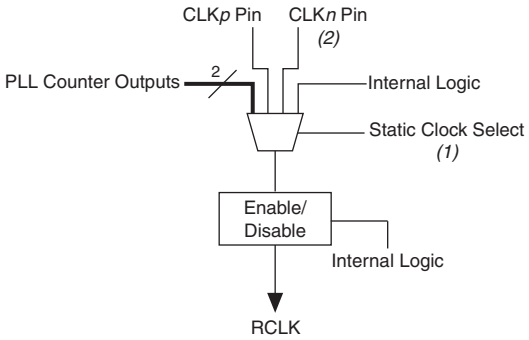
For more information about global clock control block or network in a specific device, refer to the respective device handbook.

Regional Clock Control Block

When the clock control block is configured to drive a regional clock network, you can only control the clock source selection statically. You can set any inputs to the clock select multiplexer as the clock source.

Figure 3–2 shows a clock control block configured to drive a regional clock network.

Figure 3–2. Regional Clock Control Block



Notes to Figure 3–2:

(1) You can only control these clock select signals through a configuration file and cannot be dynamically controlled during user-mode operation.

(2) Only the `CLKn` pins on the top and bottom of the device feed to the 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.

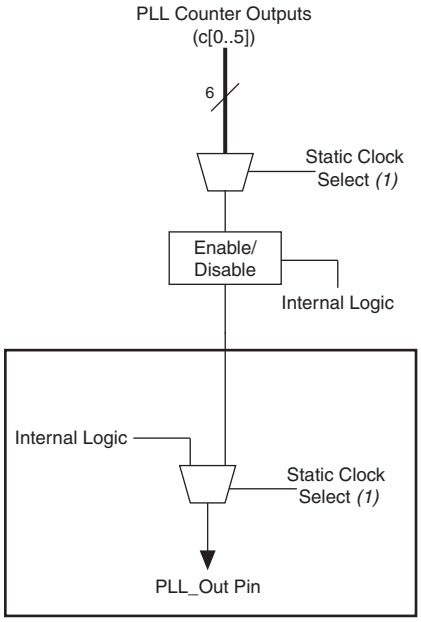
For more information about regional clock control block or network in a specific device, refer to the respective device handbook.

External PLL Output Clock Control Block

When the clock control block is configured to drive the dedicated external clock out, you can only control the clock source selection statically. You can only set the PLL outputs as the clock source.

Figure 3–3 shows a clock control block configured to drive a dedicated external clock out.

Figure 3–3. External PLL Output Clock Control Block *(1)*



Notes to Figure 3–3:

(1) 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.

(2) You can only set these clock select signals through the configuration file and cannot be dynamically controlled during user-mode operation.

For more information about external PLL output clock control block or network in a specific device, refer to the respective device handbook.

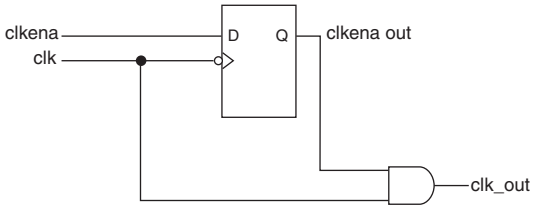
Clock Enable Signals

Single Register Clock Enable Circuit

In Cyclone III and Cyclone IV 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 low power or sleep mode.

Figure 3–4 shows how the ena clock enable signal is implemented.

Figure 3–4. Clock Enable Implementation in Cyclone III and Cyclone IV Devices



Single register is applicable for Cyclone III and Cyclone IV devices only.

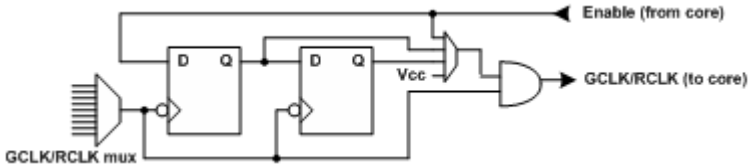
For more information about clock enable signals in a specific device, refer to the respective device handbook.

Double Register Clock Enable Circuit

The double register clock enable circuit in Arria V, Arria 10, Cyclone V, Stratix III and Stratix IV devices helps with asynchronous enable/disable of the clock network, and avoid metastability issues. If the enable signal can toggle at any time, it's possible that if the enable toggles at the same instant as the falling clock edge, the register can get "stuck" in a state between `0` and `1` for some time, before resolving. Having two registers on the path acts as a synchronization chain and reduces the probability of getting stuck in this state.

Figure 3–5 shows the double register clock enable circuit.

Figure 3–5. Double Register Clock Enable Circuit

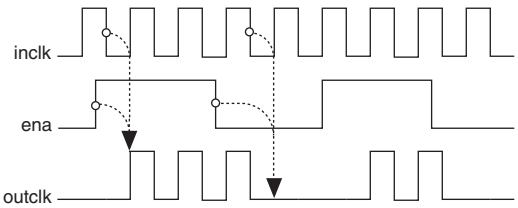


For more information about metastability issues, refer to *Managing Metastability with the Quartus II Software* chapter of the Quartus II Handbook.

Clock Enable Timing

Figure 3–6 shows a functional timing waveform example for clock-output enable. Clock enable is synchronous with the falling edge of the input clock.

Figure 3–6. Clock Enable Timing



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.

- When you are using multiple input sources, the `inclk[]` ports can only be driven by the dedicated clock input pins and the PLL clock outputs. Dedicated clock input pins must feed only `inclk[0]` and `inclk[1]`, while the PLL clock outputs must feed only `inclk[2]` and `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.
- When you are using the glitch free switchover feature, the clock you are switching from must be active. If it is not active, the switchover circuit will not be able to transition from the clock you originally selected.

ALTCLKCTRL Megafunction Ports

Table 3–3 and Table 3–4 lists the input and output ports for the ALTCLKCTRL megafunction.

Input Ports

Table 3–3. ALTCLKCTRL Megafunction Input Ports

Port Name	Required	Description	Comments

`clkselect[]`	No	Input that dynamically selects the clock source to drive the clock network that is driven by the clock buffer.	Input port `[1 DOWNTO 0]` wide.
			If omitted, the default is **GND**.
			If this signal is connected, only the global clock network can be driven by this clock control block.
			Binary Value / **Signal Selection**
			`00` — `inclk[0]`
			`01` — `inclk[1]`
			`10` — `inclk[2]`
			`11` — `inclk[3]`

`ena`	No	Clock enable of the clock buffer	If omitted, the default value is V_{CC}. This option cannot be used for periphery clock network path in Stratix® III and Stratix IV devices.
`inclk[]`	Yes	Clock input of the clock buffer	Input port `[3 DOWNTO 0]` wide. You can specify up to four clock inputs, `inclk[3:0]`. Clock pins, clock outputs from the PLL, and core signals can drive the `inclk[]` port. Multiple clock inputs are only supported for the global and auto-selected clock networks.

Output Ports

Table 3–4. ALTCLKCTRL Megafunction Output Ports

Port Name	Required	Description	Comments
`outclk`	Yes	Output of the clock buffer.	—

Prototypes and Component Declarations

This section describes the prototypes and component declarations of the ALTCLKCTRL megafunction.

Verilog HDL Prototype

You can locate the following Verilog HDL prototype in the Verilog Design File (**.v**) **altera_mf.v** in the <*Quartus II installation directory*>**\eda\synthesis** directory.

```
module  altclkctrl
#(  parameter   clock_type = "AUTO",
    parameter   intended_device_family = "unused",
    parameter   ena_register_mode = "falling edge",
    parameter   implement_in_les = "OFF",
    parameter   number_of_clocks = 4,
    parameter   use_glitch_free_switch_over_implementation = "OFF",
    parameter   width_clkselect = 2,
    parameter   lpm_type = "altclkctrl",
    parameter   lpm_hint = "unused")
(   input   wire    [width_clkselect-1:0]   clkselect,
    input   wire    ena,
    input   wire    [number_of_clocks-1:0]  inclk,
    output  wire    outclk)/* synthesis syn_black_box=1 */;
endmodule //altclkctrl
```

VHDL Component Declaration

The following VHDL component declaration is located in the VHDL Design File **(.vhd) altera_mf_components.vhd** in the <*Quartus II installation directory*>**\libraries\vhdl\altera_mf** directory.

```
component altclkctrl
    generic (
        clock_type:string := "AUTO";
        intended_device_family:string := "unused";
        ena_register_mode:string := "falling edge";
        implement_in_les:string := "OFF";
        number_of_clocks:natural := 4;
        use_glitch_free_switch_over_implementation:string := "OFF";
        width_clkselect:natural := 2;
        lpm_hint:string := "UNUSED";
        lpm_type:string := "altclkctrl"
    );
    port(
```

```
      clkselect:in std_logic_vector(width_clkselect-1 downto 0) :=
(others => '0');
      ena :  in std_logic := '1';
      inclk: in std_logic_vector(number_of_clocks-1 downto 0) :=
(others => '0');
      outclk:out std_logic
   );
end component;
```

VHDL LIBRARY-USE Declaration

The VHDL LIBRARY-USE declaration is not required if you use the VHDL component declaration.

```
LIBRARY altera_mf;
USE altera_mf.altera_mf_components.all;
```

This chapter provides additional information about the document and Altera.

Document Revision History

The following table lists the revision history for this document.

Date	Version	Changes
February 2014	3.1	■ Updated the"ALTCLKCTRL Megafunction Parameters" on page 2–5 to add the information for the **How do you want to register the 'ena' port?** and **How do you want to use the ALTCLKCTRL?** parameter settings. ■ Updated Table 3–1 on page 3–1 to include Arria 10 and Stratix V devices information. Also added a row for Large Periphery Clocks. ■ Added "Double Register Clock Enable Circuit" on page 3–5. ■ Removed "Stratix II Devices Restrictions", "Design Example: Global Clock Buffer", and "Functional Simulation in the ModelSim-Altera Software" because beginning from Quartus II software version 13.1, Stratix II devices are no longer supported.
February 2012	3.0	■ Updated information for switchover usage. ■ Added a note about assigning clock type through assignment editor.
September 2010	2.5	Updated ports and parameters

Date	Version	Changes
December 2008	2.4	■ Updated the following sections: ■ "Device Family Support" section ■ "Introduction" section ■ "General Description" section ■ "Design Example: Global Clock Buffer" section ■ "Functional Simulation in the ModelSim-Altera Software" section ■ "This chapter describes the prototype, component declaration, ports, and parameters of the ALTCLKCTRL megafunction. These ports and parameters are available to customize the ALTCLKCTRL megafunction according to your application." section ■ "How to Contact Altera" section ■ Removed the following sections: ■ "Resource Utilization & Performance" section ■ "Software and System Requirements" section ■ "Inferring Megafunctions from HDL Code" section ■ "Instantiating Megafunctions in HDL Code or Schematic Designs" section ■ "Identifying a Megafunction after Compilation" section ■ "SignalTap II Embedded Logic Analyzer" section ■ Removed all screenshots in the "MegaWizard Plug-In Manager Page Descriptions" section ■ Reorganized the "MegaWizard Plug-In Manager Page Descriptions" section into table format. ■ Renamed "About this User Guide" section to "Additional Information" and moved the section to the end of the user guide.
May 2007	2.3	Updated for Quartus II software version 7.1, including: ■ Added information on Cyclone® III and Arria® GX device support ■ Added Referenced Documents section
March 2007	2.2	Added Cyclone III device to list of supported devices.
December 2006	2.1	Updated device family support to include Stratix® III devices.
October 2006	2.0	Updated for Quartus II version 6.0, including ■ Screen shots ■ ModelSim section in Chapter 2
September 2004	1.0	Initial release

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com

Contact (1)	Contact Method	Address
Product literature	Website	www.altera.com/literature
Nontechnical support (general)	Email	nacomp@altera.com
(software licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>***.pof** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	The multimedia icon directs you to a related multimedia presentation.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.

Visual Cue	Meaning
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.