Altera Double Data Rate Megafunctions

User Guide

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

Quartus II Version: 2.2
Document Version: 1.0
Document Date: May 2003

About this User Guide

This user guide provides comprehensive information about the Altera® `altddio_in`, `altddio_out`, and `altddio_bidir` megafunctions.

Table 1 shows the user guide revision history.

Table 1. User Guide Revision History

Date	Description
May 2003	First release.

How to Find Information

- The Adobe Acrobat Find feature allows you to search the contents of a PDF file. Click on the binoculars icon in the top toolbar to open the Find dialog box.
- Bookmarks serve as an additional table of contents.
- Thumbnail icons, which provide miniature previews of each page, provide a link to the pages.
- Numerous links, shown in green text, allow you to jump to related information.

How to Contact Altera

For the most up-to-date information about Altera products, go to the Altera world-wide web site at http://www.altera.com.

For technical support on this product, go to http://www.altera.com/mysupport. For additional information about Altera products, consult the sources shown in Table 2.

Table 2. How to Contact Altera

Information Type	USA & Canada	All Other Locations
Technical support	http://www.altera.com/mysupport/	http://www.altera.com/mysupport/
	(800) 800-EPLD (3753) (7:00 a.m. to 5:00 p.m. Pacific Time)	(408) 544-7000 *(1)* (7:00 a.m. to 5:00 p.m. Pacific Time)
Product literature	http://www.altera.com	http://www.altera.com
Altera literature services	lit_req@altera.com *(1)*	lit_req@altera.com *(1)*
Non-technical customer service	(800) 767-3753	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time)
FTP site	ftp.altera.com	ftp.altera.com

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

Typographic Conventions

The *Altera Double Data Rate Megafunctions User Guide* uses the typographic conventions shown in Table 3.

Table 3. Conventions

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c.,...	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.
■	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Contents

General Description

The Altera double data rate (DDR) megafunctions (`altddio_in`, `altddio_out`, and `altddio_bidir`) allow you to configure the DDR I/O registers in Stratix, Stratix GX, Cyclone, APEX II, and Mercury devices. You can also use the megafunctions to implement DDR registers in the logic elements (LEs). In Cyclone devices, the megafunctions automatically implements the DDR registers in the LEs closest to the pin. In Mercury devices, DDR input and outputs are implemented in the I/O element (IOE), and bidirectional DDR uses a combination of IOEs and LEs. The DDR megafunctions help you interface with applications that use DDR signaling, including memory interfaces such as DDR SDRAM, DDR FCRAM , Reduced Latency DRAM I and II (RLDRAM I and RLDRAM II), and QDR SRAM, as well as high-speed applications.

The `altddio_in` megafunction implements the interface for DDR inputs. The `altddio_out` megafunction implements the interface for DDR outputs. The `altddio_bidir` megafunction implements the interface for bidirectional DDR inputs and outputs.

For more information on DDR circuitry, see

- *AN 212: Implementing Double Data Rate I/O Signaling in Stratix & Stratix GX Devices*.
- *AN 256: Implementing Double Data Rate I/O Signaling in Cyclone Devices*

This user guide contains `altddio_in` and `altddio_out` megafunction design examples in the "Design Example: Verilog HDL & VHDL DDR I/O Megafunctions" section. The example design files are available in the Design Example section of the Altera web site at **http://www.altera.com**.

Common Applications

DDR registers are used to capture and/or send data at twice the rate of the clock or data strobe to interface with a memory device or other high-speed interface applications where the data is clocked at both edges of the clock. This feature is used to interface with DDR SDRAM, FCRAM, RLDRAM I and II and QDR SRAM memory devices. You can also use the DDR I/O registers as a SERDES bypass mechanism in LVDS applications.

Features

Table 4 summaries key features of the altddio megafunction. The port/parameter column provides a convienent cross-reference to the detailed explanation of ports and parameters in Tables 5 to 13 in the "Ports & Parameters" chapter.

Table 4. altddio Megafunction Features

Feature	Port/Parameter	Description
Asynchronous clear	aclr	This option adds an asynchronous clear to the megafunction. The aset and aclr ports cannot be connected at the same time.
Asynchronous set	aset	This option adds an asynchronous set to the megafunction. The aset and aclr ports cannot be connected at the same time.
Power-up high	POWER_UP_HIGH	When both the aclr and aset ports are not used, the POWER_UP_HIGH parameter is available to specify the out ports power-up. Values are ON and OFF. The default is OFF.
Delay switch-on by half a clock cycle	EXTEND_OE_DISABLE	This option specifies whether the second OE register should be used in altddio_out or altddio_bidir. When this parameter is on, the output pin is held at high-impedance for an extra half a clock cycle after the OE port goes high. Values are ON, OFF, and UNUSED. The default is UNUSED. This option does not apply to Mercury devices.

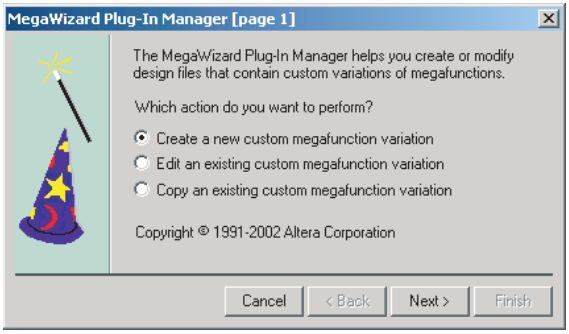
MegaWizard Page Description

The Quartus II software allows you to easily and quickly instantiate megafunctions using the MegaWizard® Plug-In Manager. To implement a megafunction, follow the below steps:

1. Launch the MegaWizard Plug-In Manager by choosing **MegaWizard Plug-In Manager** (Tools menu) in the Quartus II software.

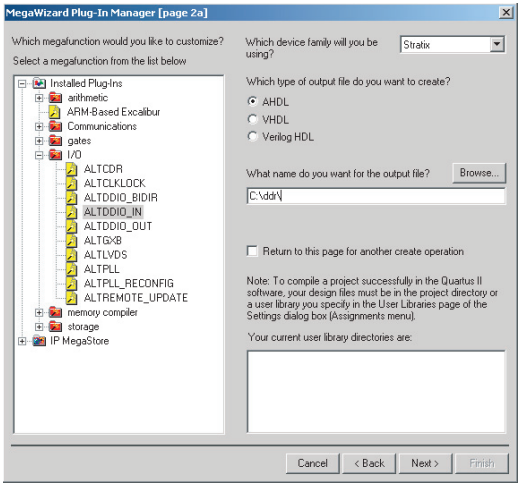
2. Select **Create a new custom megafunction variation** and click **Next**. See Figure 1.

Figure 1. Create a New Megafunction Variation



3. Click the + icon next to I/O to expand the I/O megafunction list.
4. Choose a DDR I/O megafunction under **I/O**. See Figure 2.

Figure 2. Select a DDR I/O Megafunction



5. Select an output file type and enter the desired name of the megafunction. You can choose AHDL (**.tdf**), VHDL (**.vhd**), or Verilog HDL (**.v**) as the output file type. Along with these HDL files, the MegaWizard plug-in manager creates an include file (**.inc**), a VHDL Component Declaration File (**.cmp**) and a Block Symbol File (**.bsf**).

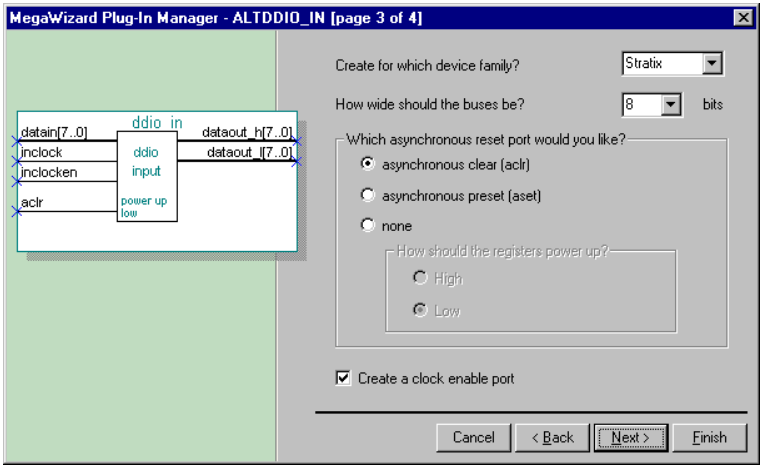
All Stratix, Stratix GX, and APEX II IOEs have the six registers and a latch that are required to implement DDR I/O. Only the number of I/O pins available per Stratix or Stratix GX device limits the data bus width. Cyclone devices implement the DDR interface in LEs. Mercury IOEs only have three registers, but the OE register is a multi-purpose register that can be used as a second input or output register, enabling it to implement the DDR inputs and outputs in the IOE. For bidirectional DDR in Mercury devices, the IOE implements the DDR output, and the LE implements the DDR input.

The following sections describe the options that are available for the DDR I/O megafunction.

altddio_in Configuration

The `altddio_in` function provides customizable parameters for device family, data bus width, type of reset, and the clock enable option. Figure 3 shows the `altddio_in` MegaWizard Plug-In.

Figure 3. altddio_in Megafunction Configuration Panel



You have the option of using the asynchronous clear (`aclr`) or the sychronous clear (`aset`) as the asynchronous reset. If you do not use either clear option, you need to specify whether the registers should power up high or low.

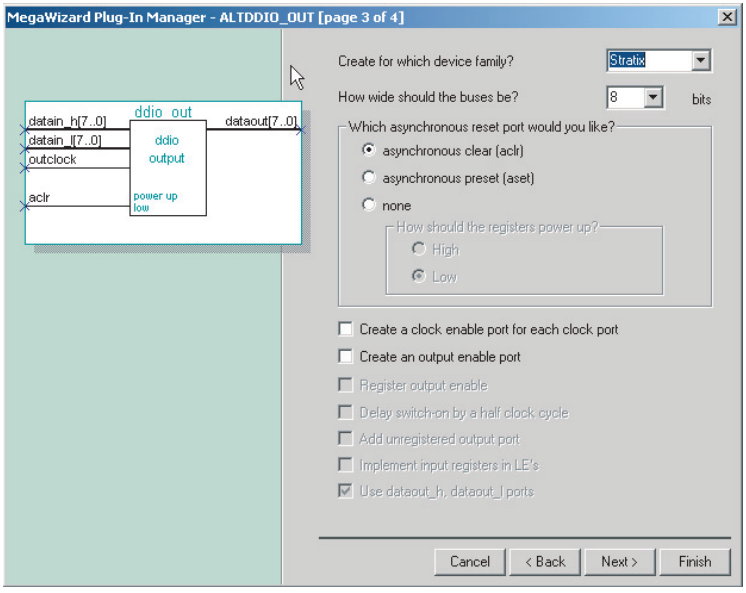
You can add a clock enable port to control when data is clocked in. This signal prevents data from being passed through.

For more information regarding the ports for this megafunction see "altddio_in" on page 21.

altddio_out Configuration

The `altddio_out` function provides customizable parameters for device family, data bus width, and type of reset. Other available options include a clock enable port, an output enable port with the option to register the port, and extending the tri-state output for a half clock cycle. Figure 4 shows the `altddio_out` MegaWizard Plug-In.

Figure 4. altddio_out Megafunction Configuration Panel



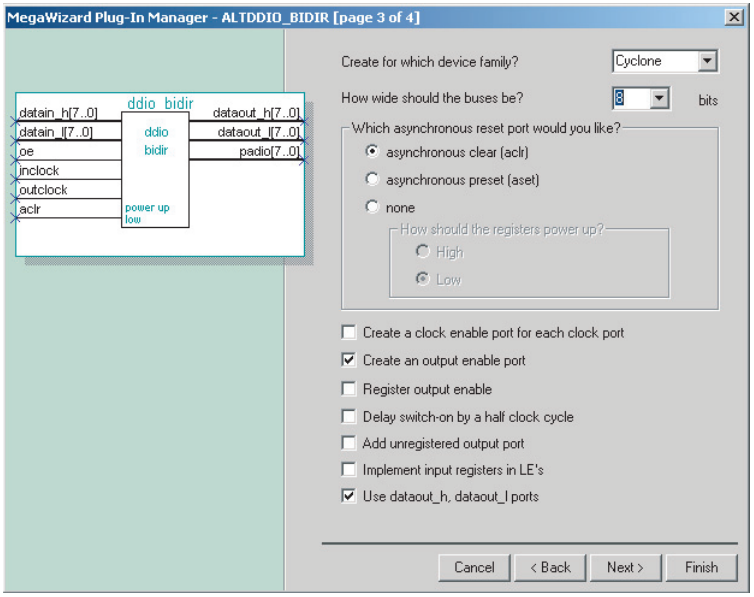
You can add an output enable port to the megafunction. When you enable this port, you have the option to register the output enable port, Aoe, and to delay the signal by a half clock cycle. The "Delay switch-on by a half clock cycle" option enables the second output enable register, Boe, as pictured in Figure 3 on page 12.

You have the option of using the asynchronous clear (`aclr`)or the sychronous clear (`aset`) as the asynchronous reset. If you do not use either clear option, you need to specify whether the registers should power up high or low.

altddio_bidir Configuration

The `altddio_bidir` megafunction combines `altddio_in` and `altddio_out` into a single megafunction, which instantiates bidirectional DDR ports. Figure 5 shows the `altddio_bidir` MegaWizard Plug-In.

Figure 5. altddio_bidir Megafunction Configuration Panel



The options for `altddio_bidir` are the same as `altddio_out` with the following additions:

- An option for an unregistered data port, `combout`, is included. The `combout` port sends data to the core bypassing the DDR I/O input registers. You can also disable the dataout ports. You need to have the `dataout_h` and `dataout_l` enabled or both or the `combout` ports for bidirectional operation.
- The input path of the `altddio_bidir` megafunction can be implemented in logic elements.
- Register and delay `OE` by a half clock cycle if you use the `OE` port. The function will act similarly to the `altddio_out`. This does not apply to Mercury devices.

For more information about the `altddio_bidir` megafunction, see "altddio_bidir" on page 26.

Conversion to Hardware

Figures 6 through 9 shows the DDR IOE structure for Stratix, Stratix GX, APEX II, and Mercury devices.

Figure 6. Stratix & Stratix GX IOE Structure

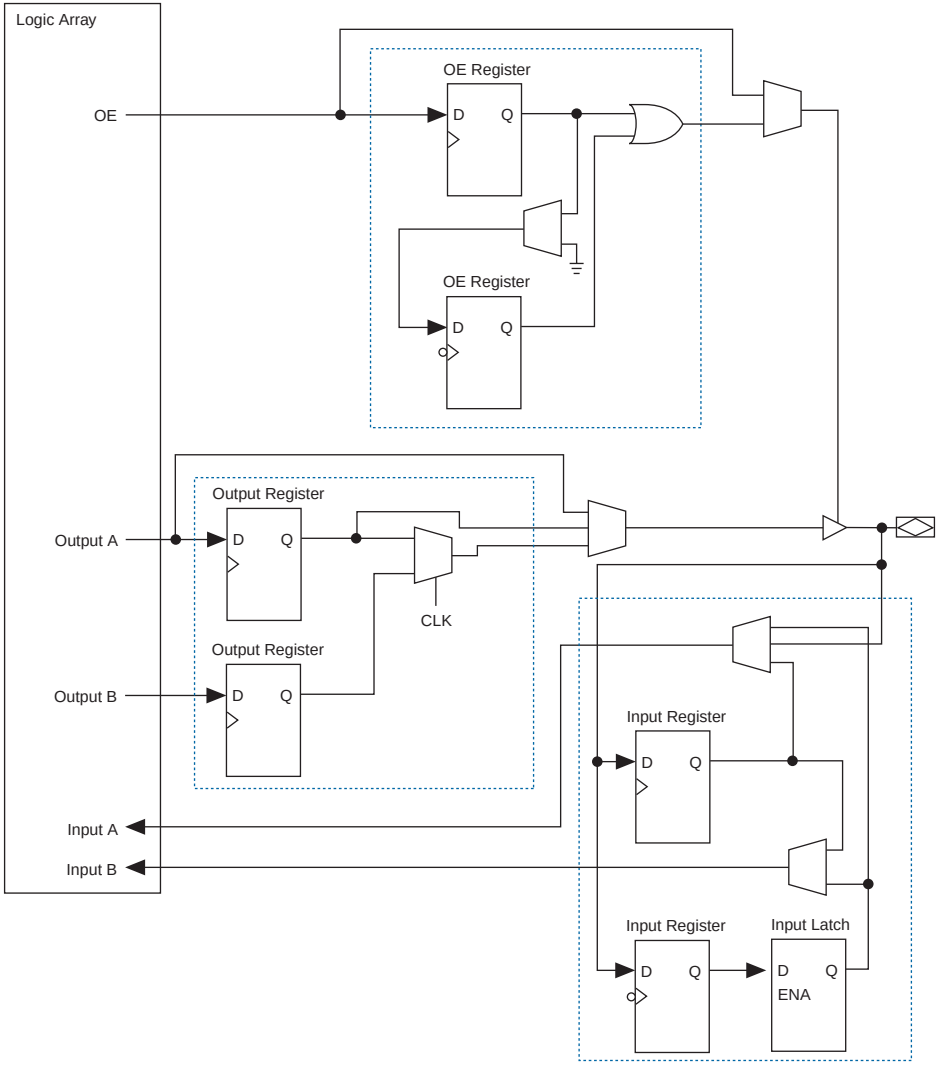


Figure 7. APEX II IOE Structure

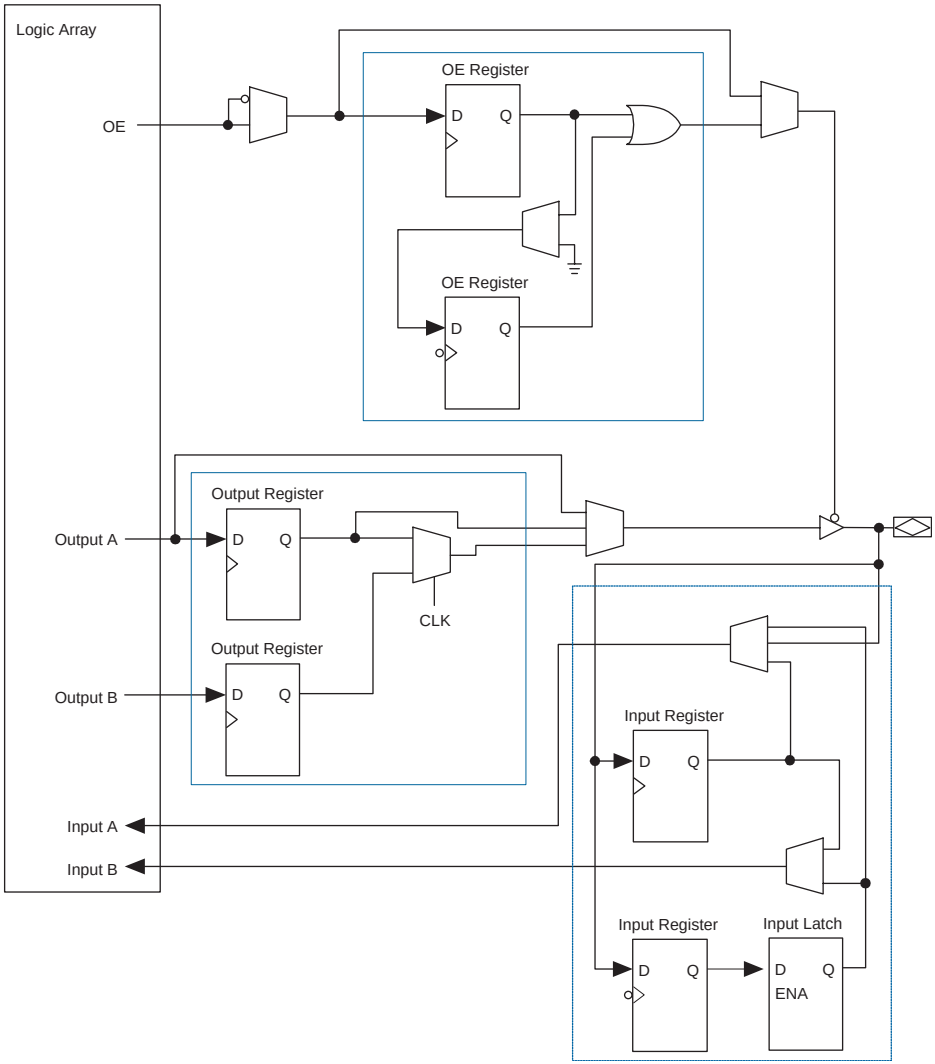


Figure 8. Mercury IOE Configured for Input

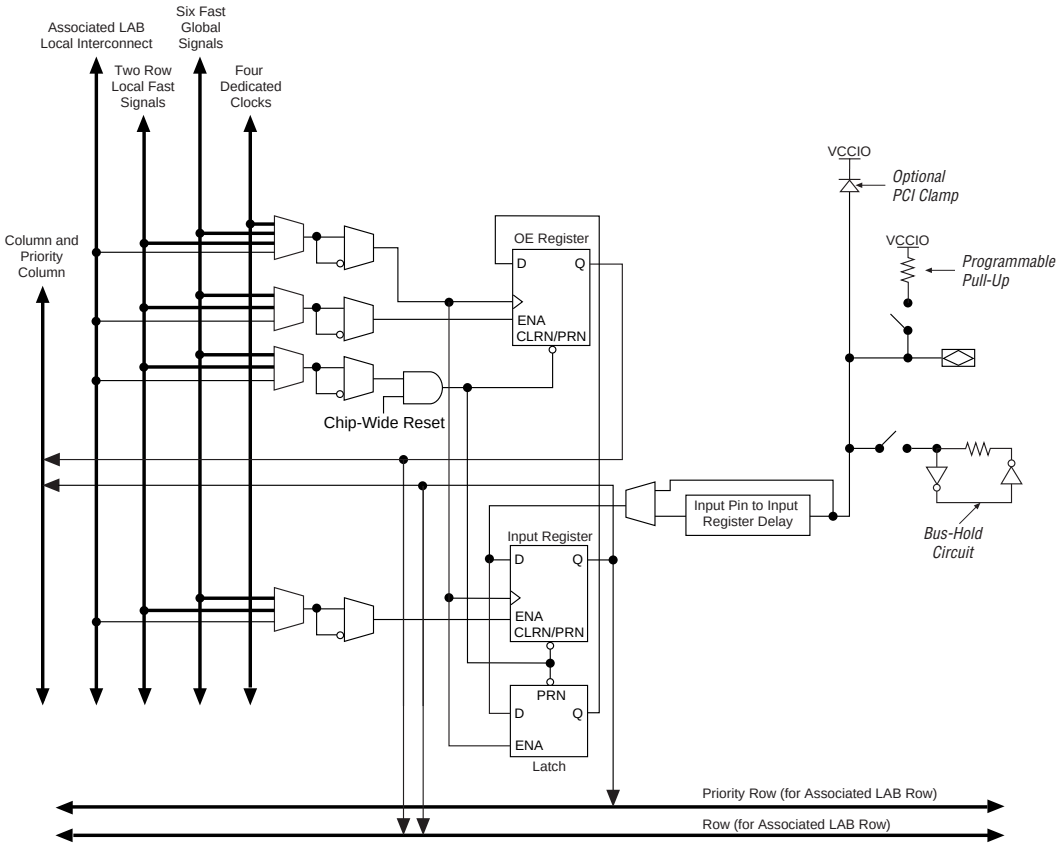
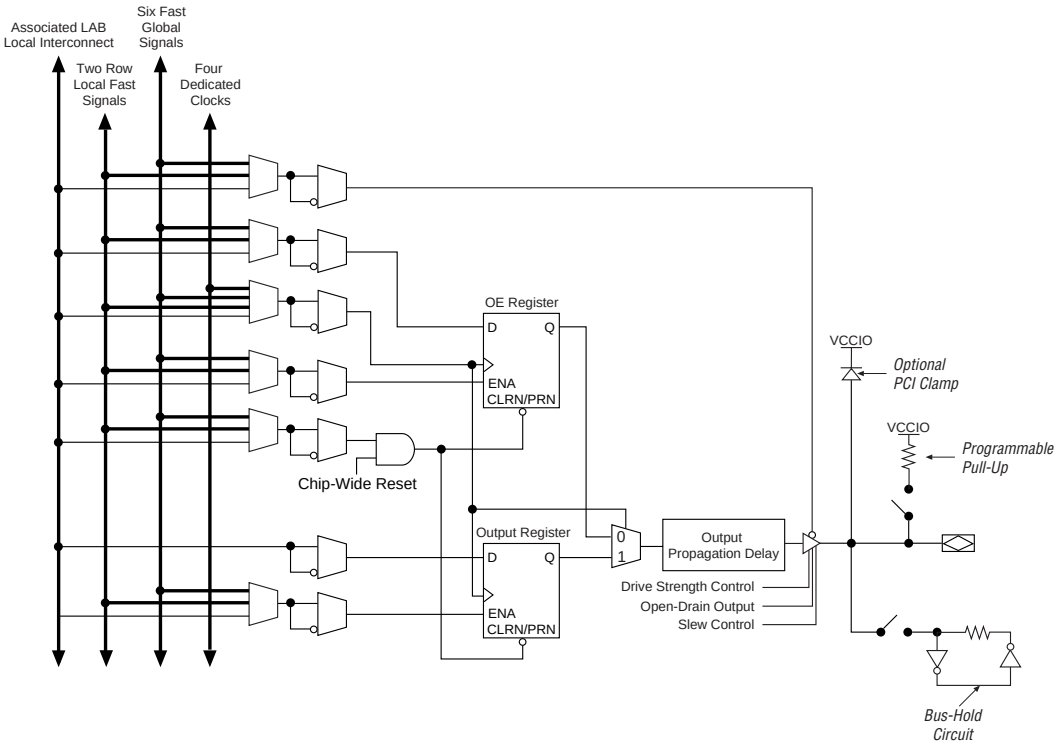


Figure 9. Mercury IOE Configured for Output



Software Requirements

The `altddio_in`, `altddio_out`, and `altddio_bidir` megafunctions are provided with the Quartus® II software version 1.1 and higher. You need to install the Quartus II software to use this megafunction.

Overview

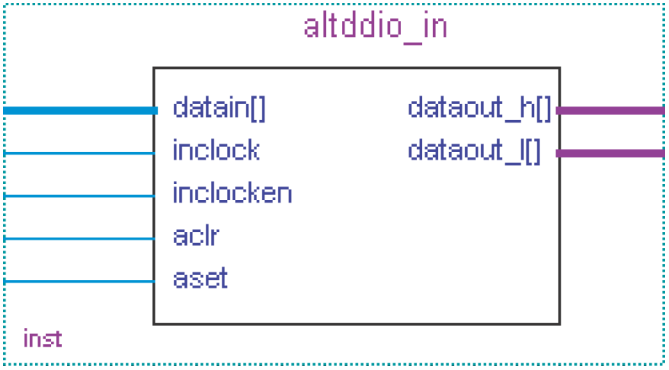
This section provides information on the `altddio_in`, `altddio_out`, and `altddio_bidir` megafunction ports and parameters.

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

altddio_in

Figure 10 shows the ports and parameters for the `altddio_in` megafunction.

Figure 10. altddio_in Port & Parameter Description



Tables 5, 6, and 7 show the port names and parameters for the `altddio_in` megafunction. The options listed in these tables are valid for Stratix and Stratix GX devices. Other ports and parameters are available if you select a different device family.

Table 5. altddio_in Input Ports

Name	Required	Description	Comments
datain[]	Yes	DDR input data port.	Input port WIDTH wide. The datain port should be directly fed from an input pin in the top level design.
inclock	Yes	Clock signal to sample the DDR input.	The datain port is sampled on each clock edge of the inclock signal.
inclocken	No	Clock enable for the data clock.	
aclr	No	Asynchronous clear input.	The aclr port and the aset port cannot be connected at the same time.
aset	No	Asynchronous preset input.	The aclr port and the aset port cannot be connected at the same time.

Table 6. altddio_in Output Ports

Name	Required	Description
dataout_h[]	Yes	Data sampled from the datain[] port at the rising edge of the inclock signal.
dataout_l[]	Yes	Data sampled from the datain[] port at the falling edge of the inclock signal.

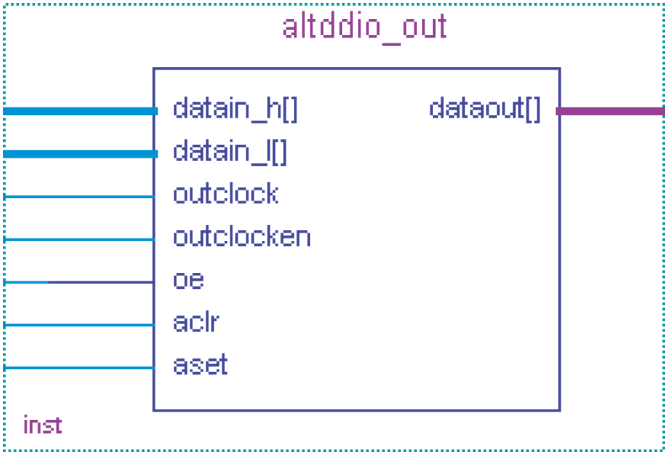
Table 7. altddio_in Parameters

Parameter	Type	Required	Description
WIDTH	Integer	Yes	Width of the datain, dataout_h, and dataout_l ports
POWER_UP_HIGH	String	No	When both the aset and aclr ports are unused, the POWER_UP_HIGH parameter can specify the power-up state of the output ports. Values are ON and OFF. The default setting is OFF.
INTENDED_DEVICE_FAMILY	String	No	This parameter is used for modeling and behavioral simulation. Create the altddio_in megafunction with the MegaWizard® Plug-in Manager to calculate the value for this parameter.

altddio_out

Figure 11 shows the ports and parameters for the altddio_out megafunction.

Figure 11. altddio_out Port & Parameter Description



Tables 8, 9, and 10 show the port names and parameters for the altddio_out megafunction. The options listed in these tables are valid for Stratix and Stratix GX devices. Other ports and parameters are available if you select a different device family.

Table 8. altddio_out Input Ports

Name	Required	Description	Comments
datain_h[]	Yes	Input data, which is output on the high level of the outclock port.	Input port WIDTH wide.
datain_l[]	Yes	Input data, which is output on the low level of the outclock port.	Input port WIDTH wide.
outclock	Yes	Clock signal to register the data output.	The dataout port outputs the DDR data on each level of the outclock signal.
outclocken	No	Clock enable for the outclock port.	
aclr	No	Asynchronous clear input.	The aclr port and the aset port cannot be connected at the same time.
aset	No	Asynchronous set input.	The aclr port and the aset port cannot be connected at the same time.
oe	No	Output enable for the dataout port.	Active-high signal. You can add an inverter if you need an active-low oe

Table 9. altddio_out Output Port

Name	Required	Description	Comments
dataout[]	Yes	DDR output data port.	Output port WIDTH wide. The dataout port should directly feed an output pin in the top-level design.

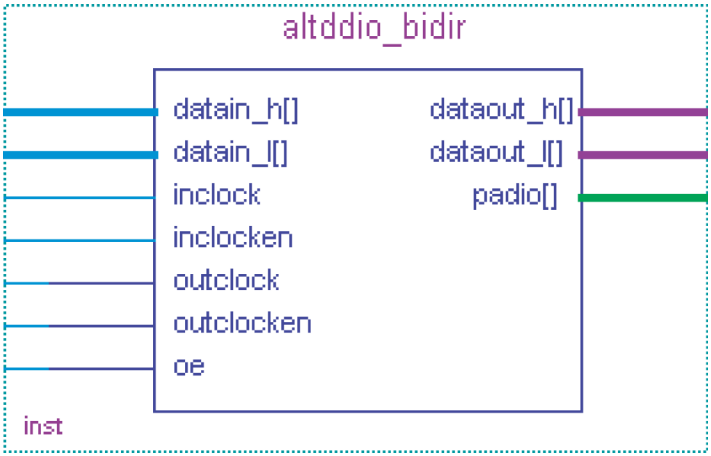
Table 10. altddio_out Parameters

Parameter	Type	Required	Comments
WIDTH	Integer	Yes	This parameter sets the width of the datain_h, datain_l, and dataout ports.
POWER_UP_HIGH	String	No	If both the aset and aclr ports are unused, the POWER_UP_HIGH parameter is available to specify the power-up state of the output ports. Values are ON and OFF. The default setting is OFF.
INTENDED_DEVICE_FAMILY	String	No	This parameter is used for modeling and behavioral simulation. Create the altddio_out megafunction with the MegaWizard Plug-in Manager to calculate the value for this parameter.
OE_REG	String	No	This specifies whether the oe port is registered. Values are REGISTERED, UNREGISTERED, and UNUSED. The default setting is UNUSED.
EXTEND_OE_DISABLE	String	No	This specifies whether the second oe register should be used. When the second oe register is used, the output pin is held at high impedance an extra half clock cycle after the oe port is disabled. Values are ON, OFF, and UNUSED. The default setting is UNUSED. This option is used to implement DDR memory interfaces.

altddio_bidir

Figure 12 shows the ports and parameters for the `altddio_bidir` megafunction.

Figure 12. altddio_bidir Port & Parameter Description



Tables 11, 12, and 13 show the port names and parameters for the altddio_bidir megafunction. The options listed in these tables are valid when targeting Stratix and Stratix GX devices. Other ports and parameters are available if you select a different device family.

Table 11. altddio_bidir Input Ports

Name	Required	Description	Comments
datain_h[]	Yes	Input data to be output to the padio port at the falling edge of the outclock port.	Input port WIDTH wide.
datain_l[]	Yes	Input data to be output to the padio port at the rising edge of the outclock port.	Input port WIDTH wide.
inclock	Yes	Clock signal to sample the DDR input.	The padio port is sampled on each clock edge of the inclock signal.
inclocken	No	Clock enable for the inclock port.	
outclock	Yes	Clock signal to register the data output.	The padio port outputs the DDR data on each level of the outclock signal.
outclocken	No	Clock enable for the outclock port.	
aclr	No	Asynchronous clear input.	The aclr port and the aset port cannot be connected at the same time.
aset	No	Asynchronous set input.	The aclr port and the aset port cannot be connected at the same time.
oe	No	Output enable for the bidirectional padio port.	If oe is not selected, then the padio port is an output port. This signal is active-high. Add an inverter if you need an active-low oe. If oe is not used, it will act as altddio_out.

Table 12. altddio_bidir Output Ports

Name	Required	Description	Comments
dataout_h[]	No	Data sampled from the padio port at the rising edge of the inclock signal.	You need to have either combout or dataout_h and dataout_l or both enabled for bidirectional operation.
dataout_l[]	No	Data sampled from the padio port at the falling edge of the inclock signal.	You need to have either combout or dataout_h and dataout_l or both enabled for bidirectional operation.
combout	No	Combinatorial data from the padio port to the logic array.	You need to have either combout or dataout_h and dataout_l or both enabled for bidirectional operation..

Table 13. altddio_bidir Bidirectional Port

Name	Required	Description	Comments
padio	Yes	Birdirectional DDR port that should directly feed a bidirectional pin in the top-level design.	The DDR data is transmitted and received on this bidirectional port.

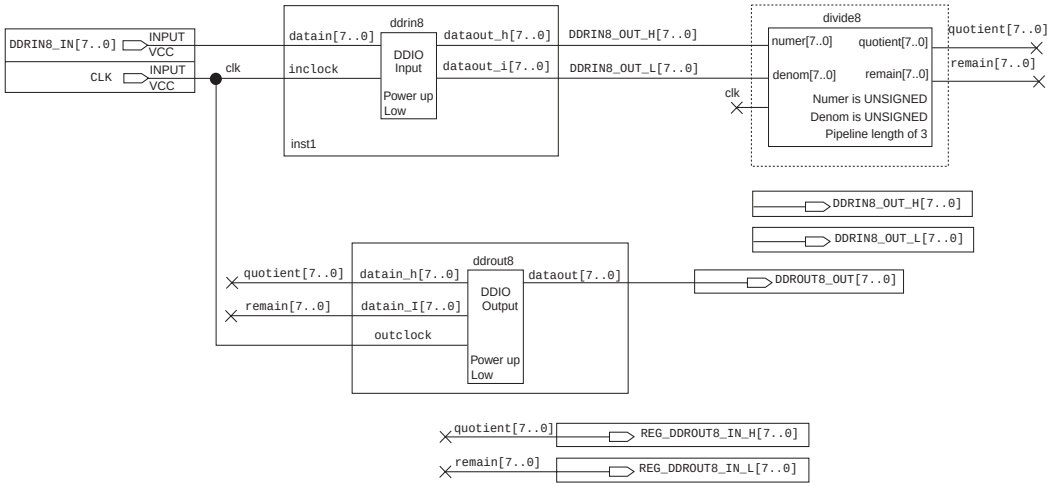
Design Example: Verilog HDL & VHDL DDR I/O Megafunctions

Introduction

This section describes how to implement DDR I/O megafunctions in a design. Figure 13 shows a simple implementation of the `altddio_in` and `altddio_out` megafunctions.

Altera provides design files (in Verilog HDL and VHDL) for the sample designs described in this section. You can download the design files from the Altera® website at **www.altera.com**. The files provided with this application note implement the designs shown in Figure 13 and Figure 15 in both Verilog HDL and VHDL. These files show how to instantiate the DDR I/O megafunctions in Verilog HDL and VHDL.

Figure 13. Sample Design using the altddio_in & altddio_out Megafunctions

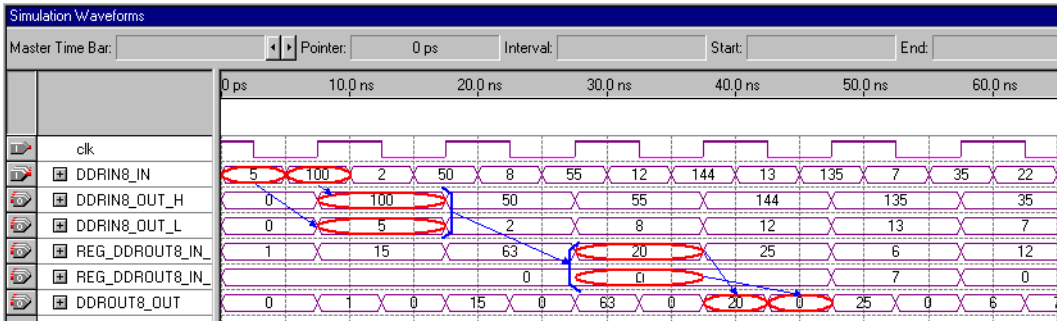


In this design, data is received at double the clock rate through pins DDRIN8_IN[7..0] of the DDRIN8 megafunction. The data output from this megafunction is fed to a simple divider circuit. The DDRIN8_OUT_H[7..0] signals are the numerator and the DDRIN8_OUT_L[7..0] signals are the denominator. The equation below describes the function of the sample design in Figure 13.

DDRIN8_OUT_H[7..0]/DDRIN8_OUT_L[7..0] = quotient[7..0] R remain[7..0]

These sets of signals (DDRIN8_OUT_H[7..0] and DDRIN8_OUT_L[7..0]) are passed into the divide8 function where the quotient and remainder are calculated. The divider calculates the quotient and remainder through a three-stage pipeline. The quotient and remainder are then fed via signals quotient[7..0] and remain[7..0] into the DDROUT8 module. The DDROUT8 module then drives the data out through pins DDROUT8_OUT[7..0] at double the data rate. Figure 14 shows the functional waveform for the sample design.

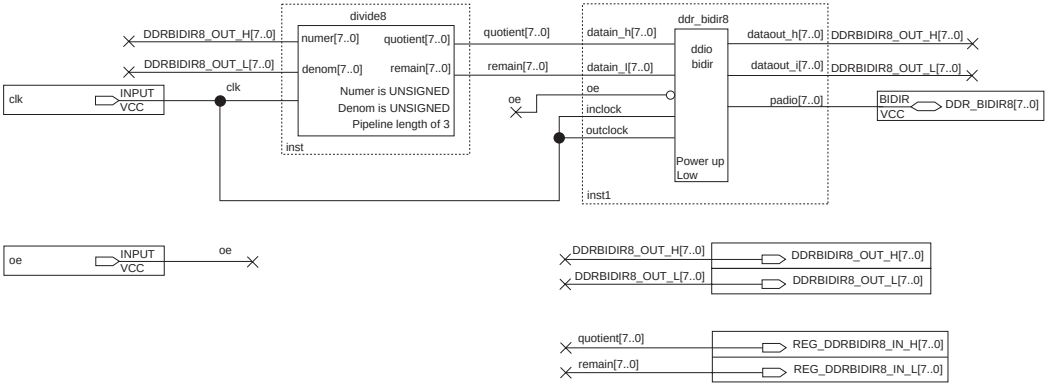
Figure 14. Timing Results for Sample Design Using altddio_in & altddio_out



1. The numerator (100) and denominator (5) are captured at 200 Mbps through pin `DDRIN8_IN`.
2. On the rising edge of `clk` at 7.5 ns, the numerator (100) drives onto the signal `DDRIN8_OUT_H` and the denominator (5) drives onto the signal `DDRIN8_OUT_L`.
3. At 27.5 ns, the quotient (20) and the remainder (0) are calculated and driven onto signals `REG_DDROUT8_IN_H` and `REG_DDROUT8_IN_L`.
4. The high level of `clk`, starting at 37.5 ns, selects the quotient (20) to drive the `DDROUT8_OUT` pin, and the low level of `clk` selects the remainder (0) to drive the same pin.
5. The waveform contains calculations for two more sets of numbers. The latency (7.5 ns to 37.5 ns) exists because of a three-stage pipeline in the divider.

Figure 15 shows a simple implementation of the `ddr_bidir8` megafunction.

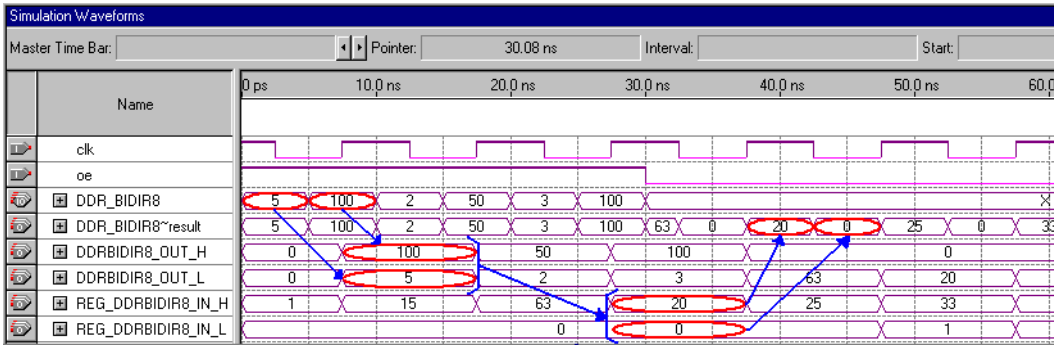
Figure 15. Sample Design Using the altddio_bidir Megafunction



This design implements the same divider example as shown in Figure 15, but instead the functionality of `altddio_in` and `altddio_out` are implemented in a single megafunction `altddio_bidir`. The bidirectional pins `DDR_BIDIR8[7..0]` receive data at double the clock rate.

The `DDRBIDIR8_OUT_H[7..0]` signals are the numerator and the `DDRBIDIR8_OUT_L[7..0]` signals are the denominator. These two sets of signals are passed into `divide8` where the quotient and remainder are calculated. The divider calculates the quotient and remainder through a three-stage pipeline. The quotient and remainder are then fed via signals `quotient[7..0]` and `remain[7..0]` back into the `altddio_bidir` megafunction. The `altddio_bidir` megafunction then drives the data out through pins `DDR_BIDIR8[7..0]` at double the data rate. Figure 16 shows the functional waveform for the sample design.

Figure 16. Timing Results for a Sample Design Using the altddio_bidir Megafunction



In Figure 16, three sets of numerators and denominators are brought in through the bidirectional pin DDR_BIDIR8. After three sets of data are brought in, the oe signal enables the answers to be driven out on the same bidirectional pin DDR_BIDIR8.

The flow of the first set of data is as follows:

1. The numerator (100) and denominator (5) are captured at 200 Mbps through pin DDRBIDIR8.
2. On the rising edge of clk at 7.5 ns, the numerator (100) drives onto the signal DDRBIDIR8_OUT_H and the denominator (5) drives onto the signal DDRIN8_OUT_L.
3. At 27.5 ns, the quotient (20) and the remainder (0) are calculated and driven to signals REG_DDRBIDIR8_IN_H and REG_DDRBIDIR8_IN_L.
4. At 30 ns, the oe signal goes low, allowing the calculated quotient and remainder to be driven out on the bidirectional pin.
5. The high level of clk starting at 37.5 ns, selects the quotient (20) to drive out the DDROUT8_OUT pin and the low level of clk selects the remainder (0) to drive out on the same pin.

Two more sets of numbers show the flow of the design. To allow the data to be driven out of the bidirectional pin in the simulation, make sure the input signal part of the bidirectional pin is set to a weak unknown, which allows the simulation to overwrite the value at the specific time interval. The Quartus II software creates an additional signal to emulate the output part of the bidirectional pin. This signal is named <*pin name*>~result. A three-stage pipeline causes latency (7.5 ns to 37.5 ns) in the divider.