LogiCORE™ IP SelectIO™ Interface Wizard v1.4

Getting Started Guide

UG700 (v1.4) July 23, 2010

XILINX®

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice.

XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.

© 2009-2010 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/16/09	1.1	Initial Xilinx release.
12/02/09	1.2	Updated tool versions. Added Phase Detector Interface Ports to Table 3-2. Added "Phase Detector" in Chapter 4. Replaced the following: IODELAY with IODELAY2; ISERDES with ISERDES2; OSERDES with OSERDES2; IDDR with IDDR2; ODDR with ODDR2.
04/19/10	1.3	Updated Wizard and tool versions. Miscellaneous edits for clarification.
07/23/10	1.4	Revised LogiCORE IP Facts table's format and content. Added support for Virtex-6 FGPAs. Added Chapter 5, "Generating the Core for a Virtex-6 FPGA."

Table of Contents

Revision History 2

Preface: About This Guide

Guide Contents 5
Additional Resources 5
Conventions 5

Chapter 1: Introduction

About the Core 9
Recommended Design Experience 9
Related Xilinx Documents 9
Additional Core Resources 10
Technical Support 10
Feedback 10

Chapter 2: Installation and Licensing

Supported Tools and System Requirements 11
  Windows 11
  Linux 11
Before You Begin 12
Installing the Wizard 12
Verifying Your Installation 12

Chapter 3: Core Architecture

Clock Buffering and Manipulation 13
Datapath 14

Chapter 4: Generating the Core for a Spartan-6 FPGA

Data Bus Setup 19
Data Bus Setup 2 20
Data Delay 21
Clock Setup 22
Clock Delay 23

Chapter 5: Generating the Core for a Virtex-6 FPGA

Data Bus Setup 25
Data Bus Setup 2 26
Data Delay 27
Clock Setup 28
Clock Delay 29

Chapter 6: Detailed Example Design

Directory and File Structure 31
Directory and File Contents 32
Implementation Scripts 34
Simulation Scripts 34
Example Design. 35
Demonstration Test Bench 36

About This Guide

The guide provides information about the Xilinx® LogiCORE™ IP SelectIO™ Interface Wizard core. This guide walks you through using the graphical user interface (GUI) and explains how to use the provided example design and test bench. The SelectIO Interface Wizard v1.4 supports Spartan®-6 and Virtex®-6 devices.

Guide Contents

This guide contains the following chapters:

- Preface, "About this Guide" introduces the organization and purpose of this guide, a list of additional resources, and the conventions used in this document.
- Chapter 1, "Introduction" describes the core and related information, including recommended design experience, additional resources, technical support, and submitting feedback to Xilinx.
- Chapter 2, "Installation and Licensing" provides instructions for installing the LogiCORE IP SelectIO Interface Wizard in the Xilinx® CORE Generator™ tool.
- Chapter 3, "Core Architecture" describes the generated I/O circuit, including the datapath and clock generators.
- Chapter 4, "Generating the Core for a Spartan-6 FPGA" provides complete information about the GUI, including detailed information about entering parameters to generate the desired I/O circuit.
- Chapter 5, "Generating the Core for a Virtex-6 FPGA" describes the GUI and follows the same flow required to set up the I/O circuit for Virtex-6 devices.
- Chapter 6, "Detailed Example Design" describes the provided example design and test bench.

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/support/documentation/index.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support/mysupport.htm.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *User Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Dark Shading	Items that are not supported or reserved	This feature is not supported
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** `[`*`option_name`*`]` *`design_name`*
Braces { }	A list of items from which you must choose one or more	**`lowpwr ={on|off}`**
Vertical bar |	Separates items in a list of choices	**`lowpwr ={on|off}`**
Angle brackets < >	User-defined variable or in code samples	<directory name>

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis . . .	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*
Notations	The prefix '0x' or the suffix 'h' indicate hexadecimal notation	A read of address 0x00112975 returned 45524943h.
	An '_n' means the signal is active low	**`usr_teof_n`** is active low.

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Chapter 1

Introduction

This chapter introduces the LogiCORE™ IP SelectIO™ Interface Wizard core and provides related information, including recommended design experience, additional resources, technical support, and submitting feedback to Xilinx. The SelectIO Interface Wizard core generates source code to implement an I/O circuit matched to your requirements and is designed to support both Verilog and VHDL design environments. In addition, the example design delivered with the core is provided in both Verilog and VHDL.

About the Core

The SelectIO Interface Wizard 1.4 core is a Xilinx® CORE Generator™ IP core supporting the SelectIO Interface in Spartan®-6 and Virtex®-6 FPGAs included in the latest IP Update on the Xilinx IP Center.

Recommended Design Experience

The SelectIO Interface Wizard is designed to be used by those will some level of experience with Xilinx I/Os. It is the Xilinx recommended method to create your I/O circuit. Advanced users can modify the provided source code. Although the SelectIO Interface Wizard provides a fully verified solution, understanding the Xilinx I/O primitives will aid users in making design trade-off decisions.

Related Xilinx Documents

UG361: *Virtex-6 FPGA SelectIO Resources User Guide*

UG362: *Virtex-6 FPGA Clocking Resources User Guide*

UG381: *Spartan-6 FPGA SelectIO Resources User Guide*

UG382: *Spartan-6 FPGA Clocking Resources User Guide*

DS709: *LogiCORE IP Clocking Wizard Data Sheet*

UG521: *LogiCORE IP Clocking Wizard Getting Started Guide*

ISE® documentation at http://www.xilinx.com/ise

Additional Core Resources

For detailed information and updates about the SelectIO Interface Wizard core, see the following documents, located on the Architecture Wizards product page:

- DS746: *LogiCORE IP SelectIO Interface Wizard v1.4 Data Sheet*
- SelectIO Interface Wizard Release Notes
- UG700: *LogiCORE IP SelectIO Interface Wizard v1.4 Getting Started Guide*

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team with expertise using the SelectIO Interface Wizard core.

Xilinx will provide technical support for use of this product as described in the *LogiCORE IP SelectIO Interface Wizard v1.4 Getting Started Guide*. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.

Feedback

Xilinx welcomes comments and suggestions about the SelectIO Interface Wizard core and the accompanying documentation.

SelectIO Interface Wizard

For comments or suggestions about the SelectIO Interface Wizard core, please submit a WebCase from www.xilinx.com/support/clearexpress/websupport.htm. Be sure to include the following information:

- Product name
- Core version number
- Explanation of your comments

Document

For comments or suggestions about the SelectIO Interface Wizard core, please submit a WebCase from www.xilinx.com/support/clearexpress/websupport.htm. Be sure to include the following information:

- Document title
- Document number
- Page number(s) to which your comments refer
- Explanation of your comments

Installation and Licensing

This chapter provides instructions for installing the LogiCORE™ IP SelectIO™ Interface Wizard in the Xilinx® CORE Generator™ tool. It is not necessary to obtain a license to use the Wizard.

Supported Tools and System Requirements

Operating Systems

Windows

- Windows XP Professional 32-bit/64-bit
- Windows Vista Business 32-bit/64-bit

Linux

- Red Hat Enterprise Linux WS v4.0 32-bit/64-bit
- Red Hat Enterprise Desktop v5.0 32-bit/64-bit (with Workstation Option)
- SUSE Linux Enterprise (SLE) desktop and server v10.1 32-bit/64-bit

Tools

- ISE® 12.2 software
- Mentor Graphics ModelSim 6.5c and above
- Cadence Incisive Enterprise Simulator (IES) 9.2 and above
- Synopsys VCS and VCS MX 2009.12 and above
- Synopsys Synplify Pro D-2009.12

Check the release notes for the required Service Pack; ISE Service Packs can be downloaded from www.xilinx.com/xlnx/xil_sw_updates_home.jsp?update=sp.

Before You Begin

Before installing the Wizard, you must have a MySupport account and the ISE 12.2 software installed on your system. If you already have an account and have the software installed, go to "Installing the Wizard", otherwise do the following:

1. Click **Login** at the top of the Xilinx home page then follow the onscreen instructions to create a MySupport account.
2. Install the ISE 12.2 software.

   For the software installation instructions, see the ISE Design Suite Release Notes and Installation Guide available in ISE software Documentation.

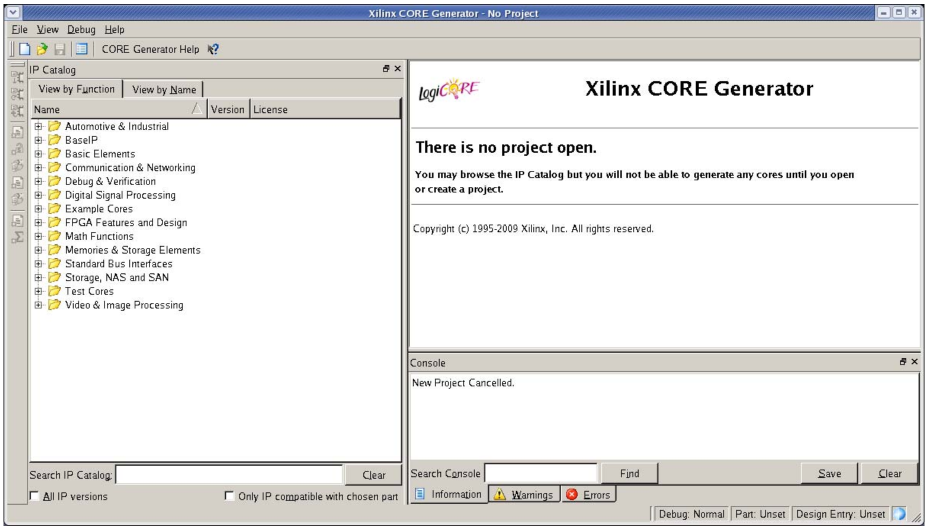
Installing the Wizard

The LogiCORE IP SelectIO Interface Wizard is included with the ISE 12.2 software. See *ISE CORE Generator IP Updates - Installation Instructions* for details about the installation of ISE 12.2 software.

Verifying Your Installation

Use the following procedure to verify that you have successfully installed the LogiCORE IP SelectIO Interface Wizard in the CORE Generator tool.

1. Start the CORE Generator tool.
2. The IP core functional categories appear at the left side of the window, as shown in Figure 2-1.


Xilinx CORE Generator - No Project
File View Debug Help
CORE Generator Help
IP Catalog
View by Function
View by Name
Name
Version
License
Automotive & Industrial
BaseIP
Basic Elements
Communication & Networking
Debug & Verification
Digital Signal Processing
Example Cores
FPGA Features and Design
Math Functions
Memories & Storage Elements
Standard Bus Interfaces
Storage, NAS and SAN
Test Cores
Video & Image Processing
Search IP Catalog:
Clear
All IP versions
Only IP compatible with chosen part
LogiCORE
Xilinx CORE Generator
Console
Search Console
Find
Save
Clear
Information
Warnings
Errors
Debug: Normal
Part: Unset
Design Entry: Unset

There is no project open.

You may browse the IP Catalog but you will not be able to generate any cores until you open or create a project.

Copyright (c) 1995-2009 Xilinx, Inc. All rights reserved.

New Project Cancelled.

Figure 2-1: **CORE Generator Tool Window**

3. Click to expand or collapse the view of individual functional categories, or click the **View by Name** tab at the top of the list to see an alphabetical list of all cores in all categories.
4. Determine if the installation was successful by verifying that SelectIO Interface Wizard v1.4 appears at the following location in the Functional Categories list:
   `/FPGA Features and Design/IO Interfaces`

Chapter 3

Core Architecture

The SelectIO™ Interface Wizard provides source HDL that implements an I/O circuit for an input, output or bidirectional bus, including the buffer, any delay elements, serialization, registers, and the I/O clock driver. The circuit is designed in two major components: clock buffering and manipulation; and datapath, which is implemented per pin.

Clock Buffering and Manipulation

The wizard supports the use of a BUFG, BUFIO, BUFIO2, or BUFPLL for clocking the I/O logic. An example circuit illustrating a BUFIO2 primitive with input data is illustrated in Figure 3-1.

Insertion delay can be added for the input clock (except in the case of a BUFPLL, which is driven from a PLL_BASE in fabric).

For serialization or deserialization of the datapath, the slower divided fabric clock is created and/or aligned to the input clock on behalf of the user (except in the case of a BUFG, which does not support serialized/deserialized data).

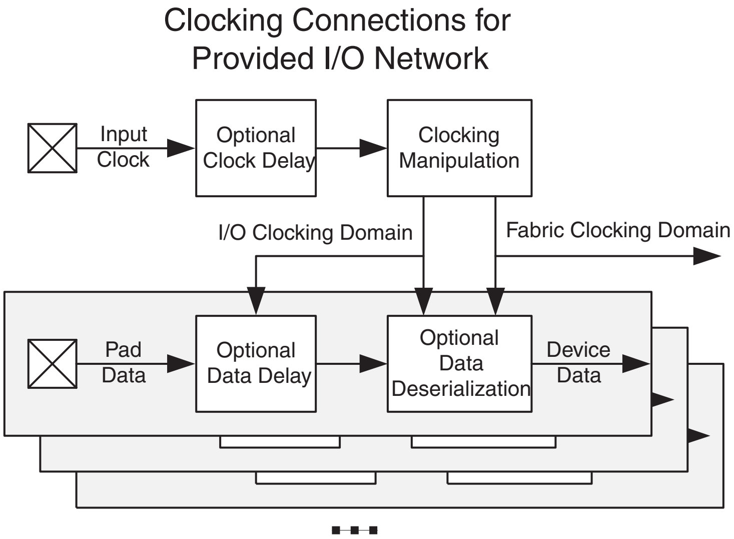

Clocking Connections for
Provided I/O Network
Input Clock
Optional Clock Delay
Clocking Manipulation
I/O Clocking Domain
Fabric Clocking Domain
Pad Data
Optional Data Delay
Optional Data Deserialization
Device Data

Figure 3-1: **Provided I/O Circuit**

Datapath

The wizard assists the user in instantiating and configuring the components within the I/O interconnect block.

The user can choose to:

- Use or bypass the delay insertion functionality
- Use serialization/deserialization through use of Input SERDES or Output SERDES
- Register double data-rate data
- Use the I/O registers for single rate data
- Drive directly into the fabric

The dataflow graph for an input bus is shown in Figure 3-2. For an output bus, the components will be similar, but the data will flow in the other direction. For a bidirectional bus, there will be both an input and output path, although there is only one IODELAY2 or IODELAYE1 element.

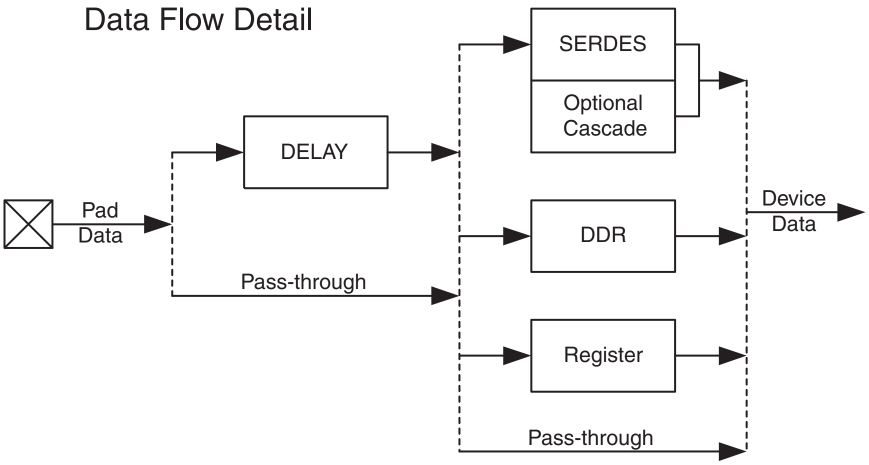

Data Flow Detail
SERDES
Optional Cascade
DELAY
Pad Data
Pass-through
DDR
Register
Pass-through
Device Data

Figure 3-2: **Flow in the I/O Input Datapath**

I/O Signals

Table 3-2 describes the input and output ports provided by the I/O circuit. All ports are optional, although there will be at least one input clock, one signal tied to a pin connection, and one signal tied to a device connection. Availability of the ports is controlled by user-selected parameters. For example, when a variable delay is selected, the delay programming ports are exposed to the user.

Table 3-1 provides a list of resources for specific I/O interconnect and clock primitives.

Table 3-1: **SelectIO and Clock Resources**

I/O Primitives	Document
Interconnect	*Spartan-6 FPGA SelectIO Resources User Guide* *Virtex-6 FPGA SelectIO Resources User Guide*
Clock	*Virtex-6 FPGA Clocking Resources User Guide* *Spartan-6 FPGA Clocking Resources User Guide*

Table 3-2 defines the I/O circuit input and output ports.

Table 3-2: **I/O Circuit Input and Output Port Descriptions**

Port	I/O	Description
Clock Ports (1)		
CLK_IN	Input	Clock in: Single-ended input clock. Available when a single-ended clock is selected.
CLK_IN_P	Input	Clock in Positive and Negative. Available when a differential clock source is selected.
CLK_IN_N		
CLK_OUT	Output	Clock out: Buffered and/or delayed output clock to connect to fabric. Available when no serialization is selected, and the clock primitive is not a BUFPLL or MMCM.
CLK_DIV_IN	Input	Clock divided in: Input clock for serialization in the I/O Logic. Available when serialization is chosen, and the clock primitive is BUFPLL/MMCM.
CLK_DIV_OUT	Output	Clock divided out: Buffered and divided output clock to connect to fabric. Available when serialization is selected, and the clock primitive is a BUFIO2 or BUFIO.
Reset Ports		
CLK_RESET	Input	Clock reset: Reset connected to clocking elements in the circuit.
IO_RESET	Input	I/O reset: Reset connected to all other elements in the circuit.
Pin Data Bus Ports		
DATA_IN_FROM_PINS	Input	Data in from pins: Single-ended input bus on the side of the pins.

Table 3-2: **I/O Circuit Input and Output Port Descriptions *(Cont'd)***

Port	I/O	Description
DATA_IN_FROM_PINS_P	Input	Data in from pins positive and negative: Differential input bus on the side of the pins.
DATA_IN_FROM_PINS_N		
DATA_OUT_TO_PINS	Output	Data out to pins: Single-ended output bus on the side of the pins.
DATA_OUT_TO_PINS_P	Output	Data out to pins positive and negative: Differential output bus on the side of the pins.
DATA_OUT_TO_PINS_N		
DATA_TO_AND_FROM_PINS	Input/ Output	Data to and from pins: Single-ended bidirectional data bus on the side of the pins
DATA_TO_AND_FROM_PINS_P	Input/ Output	Data to and from pins positive and negative: Differential bidirectional data bus on the side of the pins.
DATA_TO_AND_FROM_PINS_N		
Device Data Bus Ports		
DATA_IN_TO_DEVICE	Output	Data in to device: Input bus on the side of the device.
DATA_OUT_FROM_DEVICE	Input	Data out from device: Output bus on the side of the device.
Control and Status Ports		
BITSLIP	Input	Bit slip: Enable bit slip functionality on input data. Available on a input datapath and when enabled. For Virtex-6 based designs, this functionality is present for ISERDESE1 in NETWORKING mode.
TRAIN	Input	Train: Enable the training pattern. The train function is a means of specifying a fixed output pattern that can be used to calibrate the receiver of the signal. This port allows the FPGA logic to control whether the output is the fixed training pattern or the output data from the pins. Available on a output datapath and when enabled.
TRISTATE_OUTPUT	Input	3-state Output: Disables the output path. This signal is synchronized with the input data. Available with a bidirectional datapath.
LOCKED_IN	Input	Locked In: The input clock generator has locked. Available with a BUFPLL. Connect to locked indicator from the PLL in the fabric.
LOCKED_OUT	Output	Locked Out: The BUFPLL has locked. Use this signal as the PLL locked indicator.
Variable Delay Ports		
DELAY_BUSY	Output	Delay busy: The variable delay circuity is still busy- don't change current state.

Table 3-2: **I/O Circuit Input and Output Port Descriptions *(Cont'd)***

Port	I/O	Description
DELAY_CLK	Input	Delay clock: The clock used to control the variable delay circuitry. Most designs will have this connected to the divided/buffered clock for the I/O logic.
DELAY_DATA_CAL	Input	Delay data calibrate: Trigger calibration on the delay for the datapath.
DELAY_DATA_CE	Input	Delay data clock enable: Enable a delay change event for the datapath. In case of Virtex-6 family devices, this pin is provided for each of the IODELAYE1 components.
DELAY_DATA_INC	Input	Delay data increment: Controls whether the delay is incremented (when asserted) or decremented (when deasserted) when the delay clock is enabled. In case of Virtex-6 family devices, this pin is provided for each of the IODELAYE1 components.
DELAY_RESET	Input	IODELAYE1 reset signal: Controls the loading of initial delay value
DELAY_TAP_IN [4:0]	Input	IODELAYE1 tap in signal: Counter value from FPGA logic for dynamically loadable tap value (CNTVALUEIN). This is provided for each of the IODELAYE1 components.
DELAY_TAP_OUT[4:0]	Output	IODELAYE1 tap out signal: Counter value going to FPGA logic for monitoring tap value (CNTVALUEOUT). This is provided for each of the IODELAYE1 components.

Notes:

1. Only a single-ended or differential input clock is required. For a BUFG or BUFIO2, this comes from a pin. For a BUFPLL, this comes from fabric.

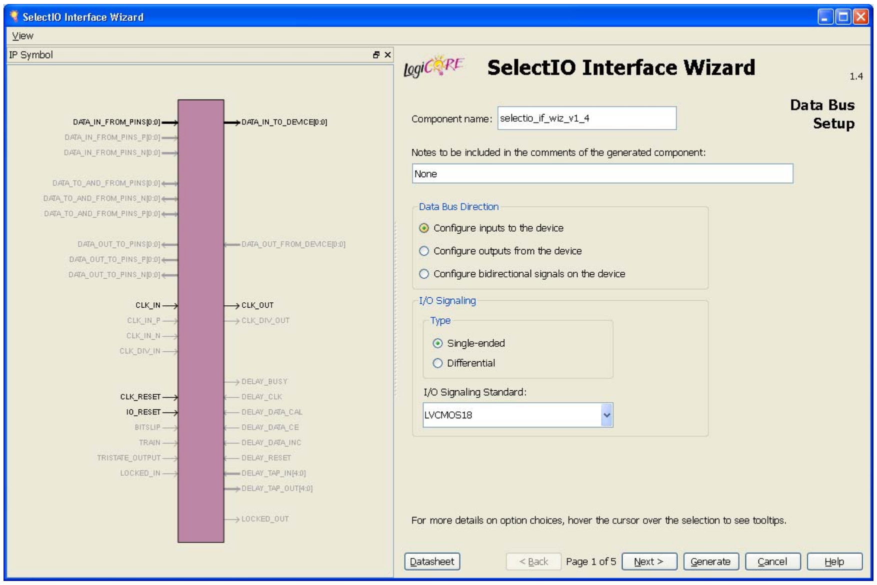
Chapter 4

Generating the Core for a Spartan-6 FPGA

This chapter describes the GUI and follows the same flow required to set up the I/O circuit. Tool tips are available in the GUI for most features; simply place your mouse over the relevant text, and additional information is provided in a pop-up dialog.

Data Bus Setup

Page 1 of the GUI (Figure 4-1) allows you to set up some general features for the data bus.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[0:0]
DATA_OUT_FROM_DEVICE[0:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Data Bus Setup
Datasheet
< Back
Page 1 of 5
Next >
Generate
Cancel
Help

Component name: selectio_if_wiz_v1_4

Notes to be included in the comments of the generated component:

None

Data Bus Direction

- ● Configure inputs to the device
- ○ Configure outputs from the device
- ○ Configure bidirectional signals on the device

I/O Signaling

Type

- ● Single-ended
- ○ Differential

I/O Signaling Standard: LVCMOS18

For more details on option choices, hover the cursor over the selection to see tooltips.

Figure 4-1: **Main Screen- Data Bus Setup - Page 1**

Component Name

User selectable component name and notes are available. Component names must not contain any reserved words in Verilog or VHDL. Notes must not include any spaces; instead use an underscore.

Data Bus Direction

The direction of the bus can be chosen here. Only choose bidirectional if you need your bus to be bidirectional: selecting it will cause restrictions later on in the configuration process.

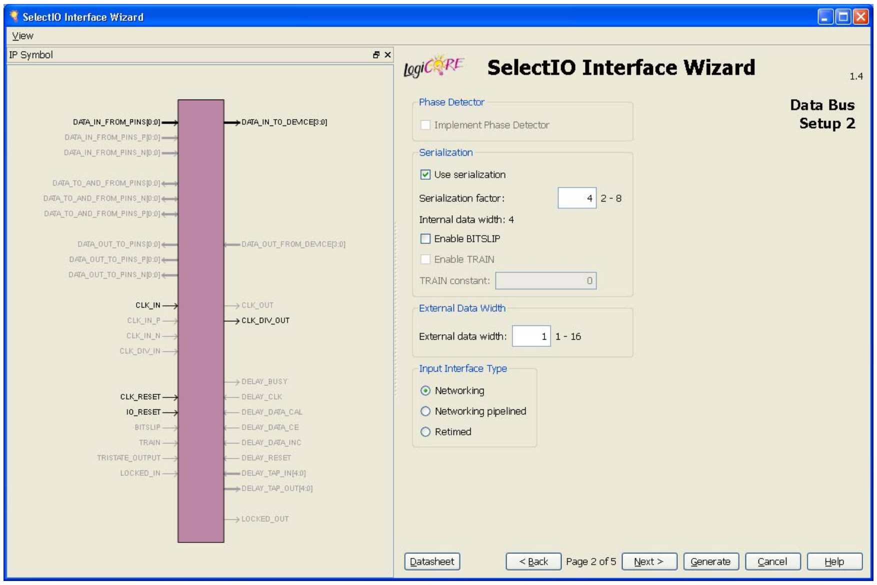
I/O Signaling

Choose whether your bus is single-ended or differential. Single-ended signals with a serialization factor of 4 or less will occupy half of an I/O pair. Single-ended signals with a serialization factor of 5 or more will occupy an entire I/O pair. Differential signals will be created as I/O pairs.

All I/O signaling standards are shown for the I/O signaling type that has been selected. This value will appear in the generated HDL code.

Data Bus Setup 2

Page 2 of the GUI (Figure 4-2) allows you to specify the configuration for the I/O interconnect datapath.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[3:0]
DATA_OUT_FROM_DEVICE[3:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Data Bus Setup 2
Datasheet
< Back
Page 2 of 5
Next >
Generate
Cancel
Help

Phase Detector

☐ Implement Phase Detector

Serialization

☑ Use serialization

Serialization factor: 4 2 - 8

Internal data width: 4

☐ Enable BITSLIP

☐ Enable TRAIN

TRAIN constant: 0

External Data Width

External data width: 1 1 - 16

Input Interface Type

● Networking

○ Networking pipelined

○ Retimed

Figure 4-2: **Data Bus Setup - Page 2**

Phase Detector

If Implement Phase Detector is selected, the IODELAY2 and ISERDES2 will be configured for phase detector. The bus on the device side will increase by the serialization factor. Selecting this feature will instantiate a reference logic that would implement the Phase detector functionality. Selecting this option makes the wizard skip the Data Delay and Clock Delay pages. The Phase Detector box is available only when I/Os are configured as Inputs with differential I/O standard.

Serialization

If Use Serialization is selected, an ISERDES2 and/or OSERDES2 will be instantiated for the user. The bus on the device side will increase by the serialization factor. All data is collected by timeslice, then concatenated from right to left. For example, assume that the output data bus is 8-bits wide, with a serialization factor of 4. If the data is presented on the pins as: 00, 01, 02, and 03, the data presented to the device will be 03020100.

If a serialization factor of 5-8 is selected, two SERDES blocks per I/O will be instantiated for a user because each SERDES is capable of a maximum serialization of 4:1. Even if a single-ended bus was chosen, the entire I/O pair is now occupied. Once serialization is selected, BITSLIP and TRAIN can be chosen depending on the presence of an input or output datapath. The TRAIN constant will be configured in the source code.

If the Phase Detector interface is chosen, then both ISERDES2 are instantiated.

External Data Width

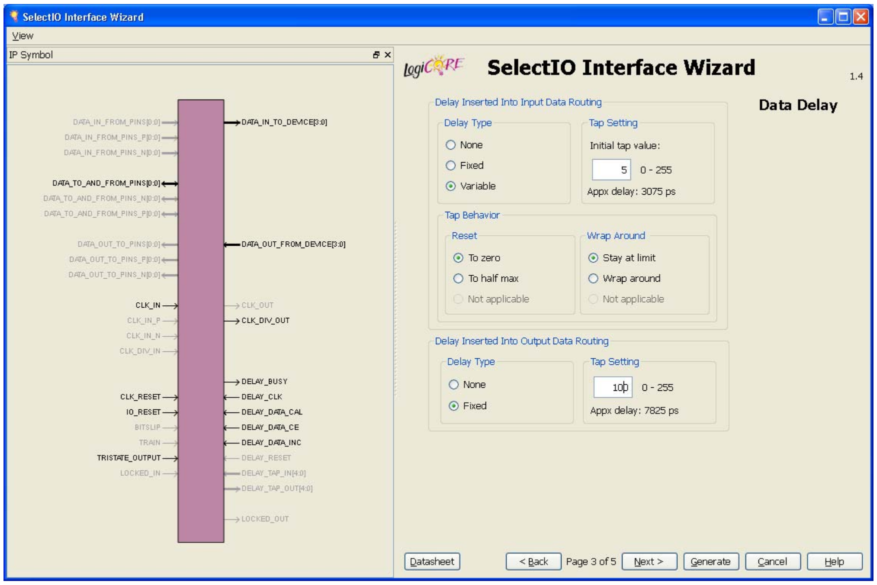
You can configure the number of bits on the system side, and this will automatically be set up on the device side. Note that differential signals will occupy two pins for each data bit.

Input Interface Type

If serialization is chosen, the interface type can be configured to set to specify the timing of the data on the device side.

Data Delay

Page 3 of the GUI (Figure 4-3) allows you to specify the type of delay for the data bus.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[3:0]
DATA_OUT_FROM_DEVICE[3:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Data Delay
Datasheet
< Back
Page 3 of 5
Next >
Generate
Cancel
Help

Delay Inserted Into Input Data Routing

Delay Type
○ None
○ Fixed
● Variable

Tap Setting
Initial tap value: 5 0 - 255
Appx delay: 3075 ps

Tap Behavior

Reset
● To zero
○ To half max
○ Not applicable

Wrap Around
● Stay at limit
○ Wrap around
○ Not applicable

Delay Inserted Into Output Data Routing

Delay Type
○ None
● Fixed

Tap Setting
100 0 - 255
Appx delay: 7825 ps

Figure 4-3: **Data Delay - Page 3**

Delay Type

An IODELAY2 will be instantiated if Fixed or Variable delay is chosen. Generally, if data is delayed with Fixed or Variable, delay will also be desired for the clock, given the high amount of insertion delay for the IODELAY2 primitive. The wizard will suggest delay settings for the clock based on the Data Delay settings. The tool will instantiate two IODELAY2 components when phase detector interface is chosen.

Tap Setting

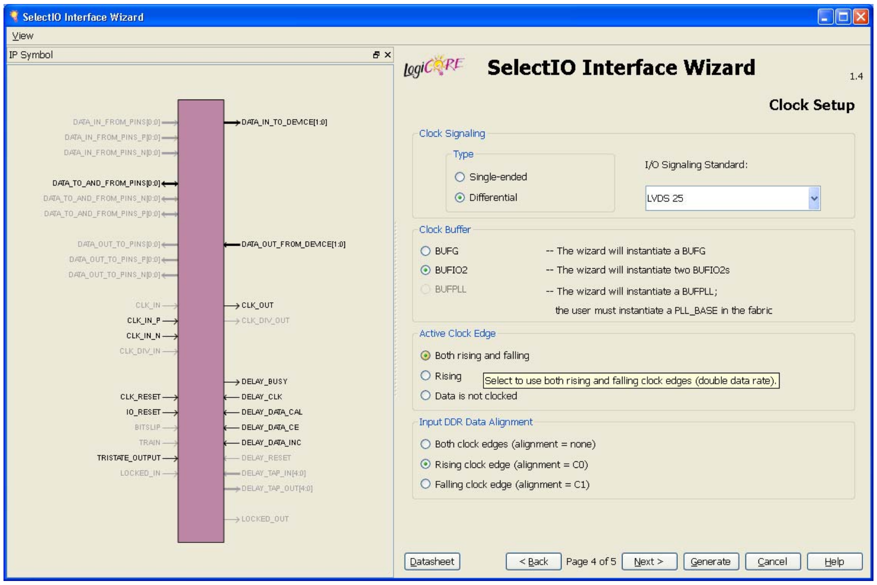
If a delay is chosen, the tap value can be specified. A typical insertion delay for the value specified is show under the Tap value box. For a variable delay, the tap value is for the initial programming. The GUI also shows the approximate delay value for tap setting. The user is expected to refer to the trace report for exact values.

Tap Behavior

If a Variable delay is chosen, the user can specify the behavior during a reset/calibration sequence, and the behavior in the event the user attempts to go past the final value in the counter.

Clock Setup

Page 4 of the GUI (Figure 4-4) allows you to configure the behavior of the clock.



SelectIO Interface Wizard 1.4

Clock Setup

Clock Signaling

Type

○ Single-ended

● Differential

I/O Signaling Standard: LVDS 25

Clock Buffer

○ BUFG -- The wizard will instantiate a BUFG

● BUFIO2 -- The wizard will instantiate two BUFIO2s

○ BUFPLL -- The wizard will instantiate a BUFPLL; the user must instantiate a PLL_BASE in the fabric

Active Clock Edge

● Both rising and falling

○ Rising

○ Data is not clocked

Select to use both rising and falling clock edges (double data rate).

Input DDR Data Alignment

○ Both clock edges (alignment = none)

● Rising clock edge (alignment = C0)

○ Falling clock edge (alignment = C1)

Page 4 of 5

SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[1:0]
DATA_OUT_FROM_DEVICE[1:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
Datasheet
< Back
Next >
Generate
Cancel
Help

Figure 4-4: **Clock Setup - Page 4**

Clock Signaling

You can specify the signaling type and standard for the input clock. The I/O signaling standard will be embedded in the provided HDL source. Using double-data rate (DDR) data places some restrictions on clocks. Either a BUFIO2 with Differential signaling, or a BUFG with Single-ended or Differential signaling must be used in this case.

Clock Buffer

If your clock comes from a pin, you should leave the input buffer as a BUFIO2 for the most flexible functionality. In the event your clock comes from fabric, you will want to choose a BUFPLL, but you will need to be sure to instantiate a PLL_BASE in fabric to drive the BUFPLL. See the LogiCORE IP Clocking Wizard core and the *LogiCORE IP Clocking Wizard Getting Started Guide* for assistance with PLL_BASE instantiation and configuration.

Active Clock Edge

If using DDR data, select Both Rising And Falling. If the data requires an asynchronous delay only, select Data Is Not Clocked. In all other circumstances, leave it at the default value of Rising. If a topology is not available, you will not be able to select it. See the *Spartan-6 FPGA SelectIO Resources User Guide* and the *Spartan-6 FPGA Clocking Resources User Guide* for more information on clocking requirements.

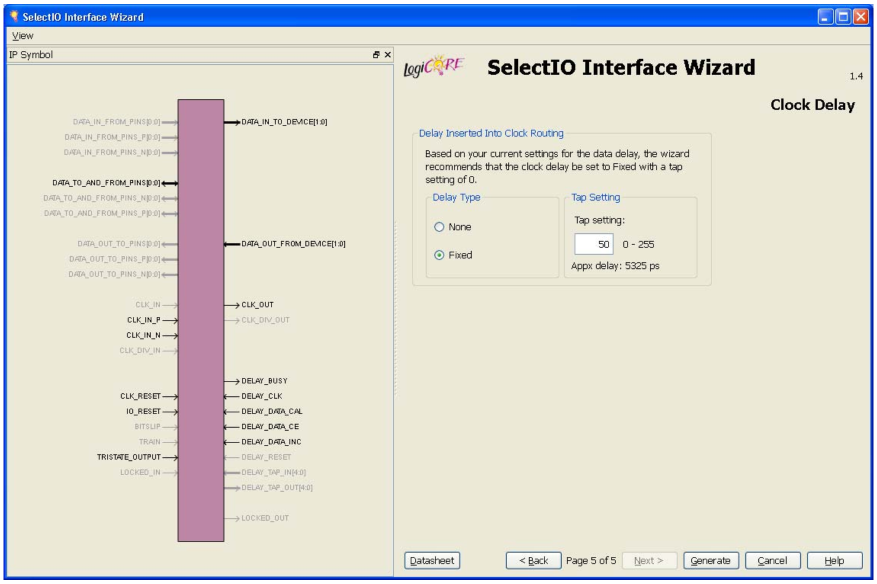
DDR Data Alignment

If serialization is not chosen, but DDR data is chosen, the ODDR2 and IDDR2 primitives can be configured to align data to the rising, falling, or both edges of the input clock. Note that the internal data width will double, and that data will be grouped by timeslice just as is it for serialization.

Clock Delay

Page 5 of the GUI (Figure 4-5) allows you to specify the type of delay for the clock bus. Please see "Data Delay" for more information on the meanings of the fields within the clock delay configuration page. When using the Phase detector feature, the clock delay is set to FIXED with a tap value of 0.

- If there is no delay in the datapath, there generally should not be any delay in the clock path, choose None.
- If there is delay in the datapath, you'll generally want to match the insertion delay in the clock path, choose Fixed with a tap value of 0.



SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[1:0]
DATA_OUT_FROM_DEVICE[1:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Clock Delay
Datasheet
< Back
Page 5 of 5
Next >
Generate
Cancel
Help

Delay Inserted Into Clock Routing

Based on your current settings for the data delay, the wizard recommends that the clock delay be set to Fixed with a tap setting of 0.

Delay Type

○ None
● Fixed

Tap Setting

Tap setting: 50 0 - 255
Appx delay: 5325 ps

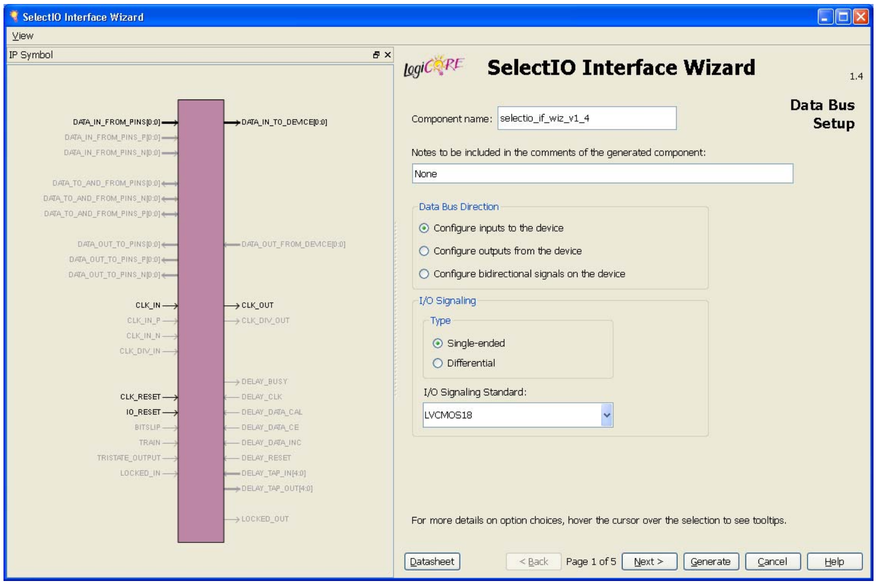
Figure 4-5: **Clock Delay - Page 5**

Generating the Core for a Virtex-6 FPGA

This chapter describes the GUI and follows the same flow required to set up the I/O circuit for Virtex®-6 devices. Tool tips are available in the GUI for most features; simply place your mouse over the relevant text, and additional information is provided in a pop-up dialog.

Data Bus Setup

Page 1 of the GUI (Figure 5-1) allows you to set up some general features for the data bus.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[0:0]
DATA_OUT_FROM_DEVICE[0:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Data Bus Setup
Datasheet
< Back
Page 1 of 5
Next >
Generate
Cancel
Help

Component name: selectio_if_wiz_v1_4

Notes to be included in the comments of the generated component:
None

Data Bus Direction
● Configure inputs to the device
○ Configure outputs from the device
○ Configure bidirectional signals on the device

I/O Signaling
Type
● Single-ended
○ Differential

I/O Signaling Standard:
LVCMOS18

For more details on option choices, hover the cursor over the selection to see tooltips.

Figure 5-1: **Main Screen- Data Bus Setup - Page 1**

Component Name

User selectable component name and notes are available. Component names must not contain any reserved words in Verilog or VHDL. Notes must not include any spaces; instead use an underscore.

Data Bus Direction

The direction of the bus can be chosen here. Only choose bidirectional if you need your bus to be bidirectional: selecting it will cause restrictions later on in the configuration process.

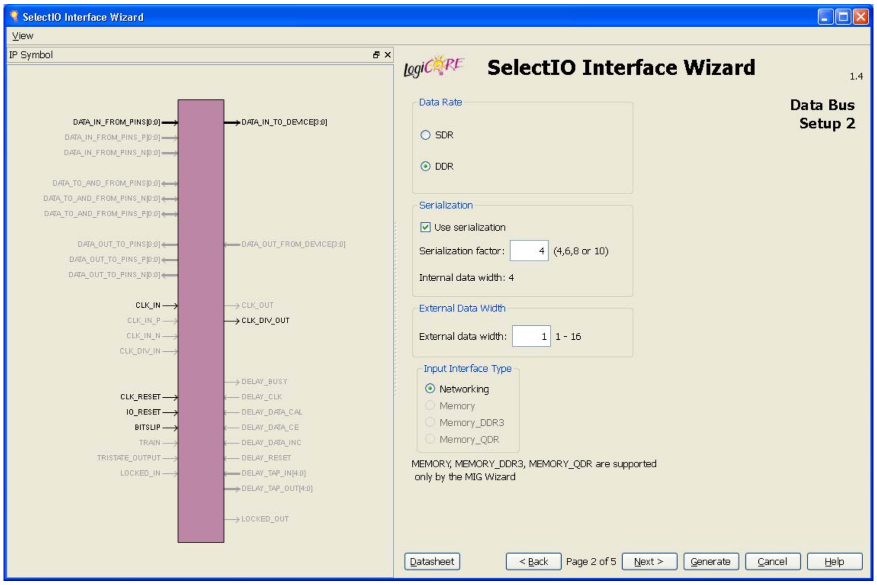
I/O Signaling

Choose whether your bus is single-ended or differential. Single-ended signals with a serialization factor of 6 or less will occupy half of an I/O pair. Single-ended signals with a serialization factor of 7 or more will occupy an entire I/O pair. Differential signals will be created as I/O pairs.

All I/O signaling standards are shown for the I/O signaling type that has been selected. This value will appear in the generated HDL code.

Data Bus Setup 2

Page 2 of the GUI (Figure 5-2) allows you to specify the configuration for the I/O interconnect datapath.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[3:0]
DATA_OUT_FROM_DEVICE[3:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Data Bus Setup 2
Datasheet
< Back
Page 2 of 5
Next >
Generate
Cancel
Help

Data Rate
○ SDR
● DDR

Serialization
☑ Use serialization
Serialization factor: 4 (4,6,8 or 10)
Internal data width: 4

External Data Width
External data width: 1 1 - 16

Input Interface Type
● Networking
○ Memory
○ Memory_DDR3
○ Memory_QDR

MEMORY, MEMORY_DDR3, MEMORY_QDR are supported only by the MIG Wizard

Figure 5-2: **Data Bus Setup - Page 2**

Data Rate

The user selects "SDR" if the data is clocked on rising edge. If the incoming or outgoing data is clocked on both the edges, then the user should select "DDR".

The selection of Data Rate affects the serialization factor limits. These are displayed dynamically on the page.

Serialization

If Use Serialization is selected, an ISERDESE1 and/or OSERDESE1 will be instantiated for the user. The bus on the device side will increase by the serialization factor. All data is collected by timeslice, then concatenated from right to left. For example, assume that the output data bus is 8-bits wide, with a serialization factor of 4. If the data is presented on the pins as: 00, 01, 02, and 03, the data presented to the device will be 03020100.

If a serialization factor of 7-10 is selected, two SERDES blocks per I/O will be instantiated for a user because each SERDES is capable of a maximum serialization of 6:1. Even if a single-ended bus was chosen, the entire I/O pair is now occupied. When the Data Rate is "SDR", the possible values for serialization factor are 2-8. When Data Rate is "DDR", the serialization factor can be set to 4, 6, 8, or 10.

External Data Width

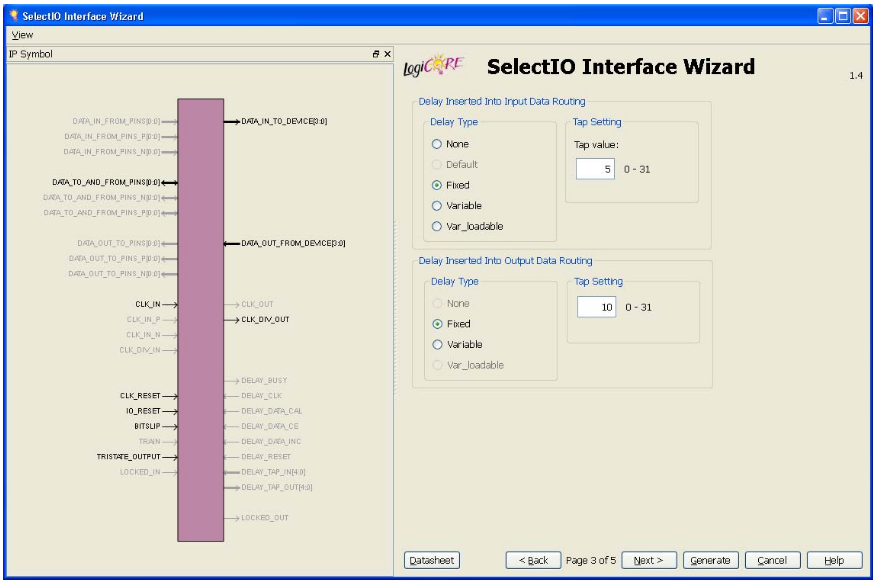
You can configure the number of bits on the system side, and this will automatically be set up on the device side. Note that differential signals will occupy two pins for each data bit.

Input Interface Type

If serialization is chosen, the interface type can be configured to set to specify the timing of the data on the device side. The SelectIO Interface Wizard only supports NETWORKING type of Input Interface. For other interfaces such as MEMORY, MEMORY_QDR and MEMORY_DDR3 user should refer to the MIG tool. Bitslip functionality is always enabled for NETWOKRING mode. User should tie this pin to logic 0 if not required.

Data Delay

Page 3 of the GUI (Figure 5-3) allows you to specify the type of delay for the data bus.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[3:0]
DATA_OUT_FROM_DEVICE[3:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Datasheet
< Back
Page 3 of 5
Next >
Generate
Cancel
Help

Delay Inserted Into Input Data Routing

Delay Type
○ None
○ Default
● Fixed
○ Variable
○ Var_loadable

Tap Setting
Tap value: 5 0 - 31

Delay Inserted Into Output Data Routing

Delay Type
○ None
● Fixed
○ Variable
○ Var_loadable

Tap Setting
10 0 - 31

Figure 5-3: **Data Delay - Page 3**

Delay Type

An IODELAYE1 will be instantiated if Default, Fixed, Variable or Var_loadable delay is chosen. Generally, if data is delayed with Fixed or Variable, delay will also be desired for the clock, given the high amount of insertion delay for the IODELAYE1 primitive.For a bidirectional bus, only specific combinations of Input and Ouput delay are possible.

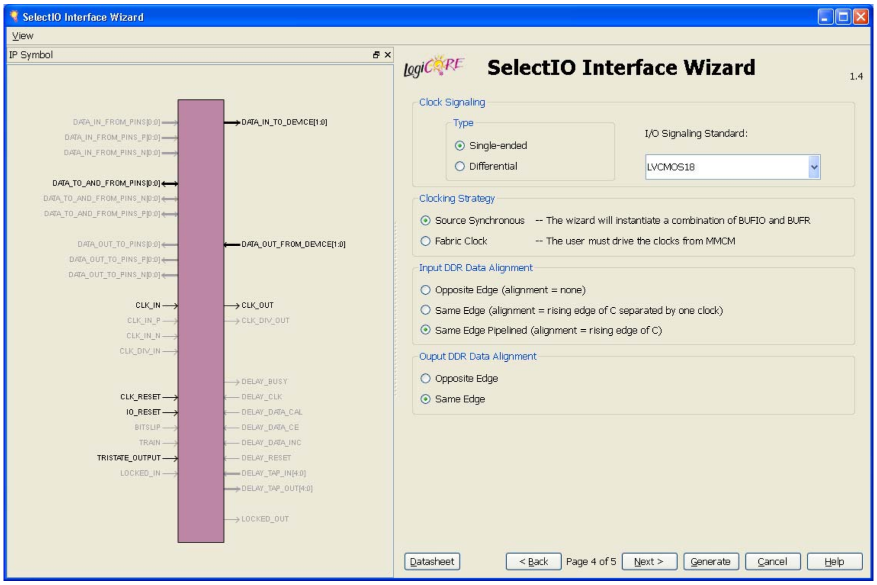
Selecting Variable or Var_loadable option will enable the user to control each IODELAYE1 element individually. This means that the control signals of each IODELAYE1 element (for example, CE, INC, CNTVALUEIN, CNTVALUEOUT) would be accessible to the user. If the user wishes to control all IODELAYE1s in same way and at same time, then the signals such as CE, INC and CNTVALUEIN can be driven together from a common logic.

Tap Setting

If delay type chosen is FIXED, VARIABLE, the tap value can be specified. The allowed value for tap is 0-31.

Clock Setup

Page 4 of the GUI (Figure 5-4) allows you to configure the behavior of the clock.


SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[1:0]
DATA_OUT_FROM_DEVICE[1:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Datasheet
< Back
Page 4 of 5
Next >
Generate
Cancel
Help

Clock Signaling

Type

● Single-ended

○ Differential

I/O Signaling Standard: LVCMOS18

Clocking Strategy

● Source Synchronous -- The wizard will instantiate a combination of BUFIO and BUFR

○ Fabric Clock -- The user must drive the clocks from MMCM

Input DDR Data Alignment

○ Opposite Edge (alignment = none)

○ Same Edge (alignment = rising edge of C separated by one clock)

● Same Edge Pipelined (alignment = rising edge of C)

Ouput DDR Data Alignment

○ Opposite Edge

● Same Edge

Figure 5-4: **Clock Setup - Page 4**

Clock Signaling

You can specify the signaling type and standard for the input clock. The I/O signaling standard will be embedded in the provided HDL source. For any design that is being configured, the clock I/O signalling standard will be same as that of bus I/O standard.

Clocking Strategy

If your clock comes from a pin, you should leave the input buffer as "Source Synchronous" for the most flexible functionality. Selecting this option will instantiate the necessary circuitry of BUFIO and BUFR and configure the same.

In the event your clock comes from fabric, you will want to choose "Fabric Clock", but you will need to be sure to instantiate a MMCM in the fabric to drive the clocks. Selecting the "Fabric Clock" option will override the “Clock Signaling” section. See the LogiCORE IP Clocking Wizard core and the *LogiCORE IP Clocking Wizard Getting Started Guide* for assistance with MMCM instantiation and configuration.

Input and Output DDR Data Alignment

If serialization is not chosen, but DDR data is chosen, the ODDR and IDDR primitives can be configured to align data to the rising, falling, or both edges of the input clock. Note that the internal data width will double, and that data rate will be grouped by timeslice just as is it for serialization.

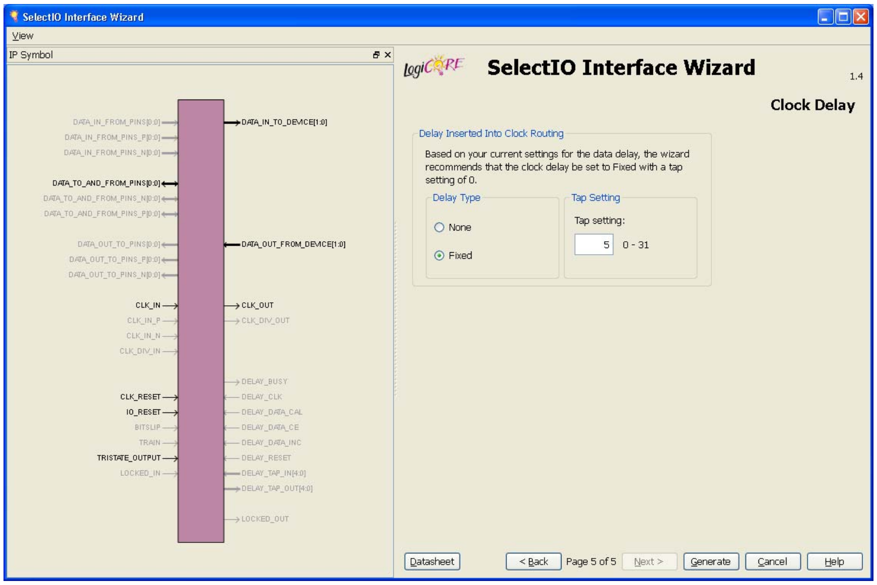
The Input DDR Data Alignment option is available when bus direction is input or bidirectional.

The Output DDR Data Alignment option is available when bus direction is output or bidirectional.

Clock Delay

Page 5 of the GUI (Figure 5-5, page 29) allows you to specify the type of delay for the clock bus. See "Data Delay" for the definitions of the clock delay configuration fields.

- If there is no delay in the datapath, there generally should not be any delay in the clock path, choose None.
- If there is delay in the datapath, you'll generally want to match the insertion delay in the clock path, choose Fixed with a tap value of 0.



SelectIO Interface Wizard
View
IP Symbol
DATA_IN_FROM_PINS[0:0]
DATA_IN_FROM_PINS_P[0:0]
DATA_IN_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS[0:0]
DATA_TO_AND_FROM_PINS_N[0:0]
DATA_TO_AND_FROM_PINS_P[0:0]
DATA_OUT_TO_PINS[0:0]
DATA_OUT_TO_PINS_P[0:0]
DATA_OUT_TO_PINS_N[0:0]
CLK_IN
CLK_IN_P
CLK_IN_N
CLK_DIV_IN
CLK_RESET
IO_RESET
BITSLIP
TRAIN
TRISTATE_OUTPUT
LOCKED_IN
DATA_IN_TO_DEVICE[1:0]
DATA_OUT_FROM_DEVICE[1:0]
CLK_OUT
CLK_DIV_OUT
DELAY_BUSY
DELAY_CLK
DELAY_DATA_CAL
DELAY_DATA_CE
DELAY_DATA_INC
DELAY_RESET
DELAY_TAP_IN[4:0]
DELAY_TAP_OUT[4:0]
LOCKED_OUT
LogiCORE
SelectIO Interface Wizard
1.4
Datasheet
< Back
Page 5 of 5
Next >
Generate
Cancel
Help

Clock Delay

Delay Inserted Into Clock Routing

Based on your current settings for the data delay, the wizard recommends that the clock delay be set to Fixed with a tap setting of 0.

Delay Type

○ None

● Fixed

Tap Setting

Tap setting: 5 0 - 31

Figure 5-5: **Clock Delay - Page 5**

Chapter 6

Detailed Example Design

This chapter provides detailed information about the example design, including a description of files and the directory structure generated by the Xilinx® CORE Generator™ tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench.

Directory and File Structure

- **<project directory>**
  Top-level project directory; name is user-defined
  - <project directory>/<component name>
    Core release notes file
    - <component name>/doc
      Product documentation
    - <component name>/example design
      Verilog and VHDL design files
    - <component name>/implement
      Implementation script files
      - implement/results
        Results directory, created after implementation scripts are run, and contains implement script results
    - <component name>/simulation
      Simulation scripts
      - simulation/functional
        Functional simulation files

Directory and File Contents

The SelectIO™ Interface Wizard core directories and their associated files are defined below.

<project directory>

The <project directory> contains all the CORE Generator tool project files.

Table 6-1: **Project Directory**

Name	Description
`<project_dir>`	
`<component_name>.v[hd]`	Verilog or VHDL source code.
`phase_detector.v[hd]`	Verilog or VHDL source code of phase detector implementation logic.
`<component_name>.xco`	CORE Generator tool project-specific option file; can be used as an input to the CORE Generator tool.
`<component_name>_flist.txt`	List of files delivered with the core.
`<component_name>.{veo|vho}`	VHDL or Verilog instantiation template.
`<component_name>.ise`	Files used to incorporate the core into an ISE® software project.

Back to Top

<project directory>/<component name>

The <component name> directory contains the readme file provided with the core, which may include last-minute changes and updates.

Table 6-2: **Component Name Directory**

Name	Description
`<project_dir>/<component_name>`	
`selectio_wizard_readme.txt`	SelectIO Interface Wizard readme file.

Back to Top

<component name>/example design

The example design directory contains the example design files provided with the core.

Table 6-3: **Example Design Directory**

Name	Description
`<project_dir>/<component_name>/example_design`	
`<component_name>_exdes.v[hd]`	Implementable Verilog or VHDL example design, connecting the I/O circuit to a RAM.

Back to Top

<component name>/doc

The doc directory contains the PDF documentation provided with the core.

Table 6-4: **Doc Directory**

Name	Description
`<project_dir>/<component_name>/doc`	
`selectio_wiz_ds746.pdf`	*LogiCORE IP SelectIO Interface Wizard v1.4 Data Sheet*
`selectio_wiz_gsg700.pdf`	*LogiCORE IP SelectIO Interface Wizard v1.4 Getting Started Guide*

Back to Top

<component name>/implement

The implement directory contains the core implementation script files.

Table 6-5: **Implement Directory**

Name	Description
`<project_dir>/<component_name>/implement`	
Scripts and projects to implement the example design	

Back to Top

implement/results

The results directory is created by the implement script, after which the implement script results are placed in the results directory.

Table 6-6: **Results Directory**

Name	Description
`<project_dir>/<component_name>/implement/results`	
Implement script result files.	

Back to Top

<component name>/simulation

The simulation directory contains the simulation test bench for the example design.

Table 6-7: **Simulation Directory**

Name	Description
`<project_dir>/<component_name>/simulation`	
`<component_name>_tb.v[hd]`	Demonstration test bench.

Back to Top

simulation/functional

The functional directory contains functional simulation scripts provided with the core.

Table 6-8: **Functional Directory**

Name	Description
`<project_dir>/<component_name>/simulation/functional`	
Contains simulation scripts and waveform formats.	

Back to Top

Implementation Scripts

The implementation script is either a shell script or batch file that processes the example design through the Xilinx tool flow. It is located at:

UNIX

```
<project_dir>/<component_name>/implement/implement.sh
```

Windows

```
<project_dir>/<component_name>/implement/implement.bat
```

The implement script performs the following steps:

- Synthesizes the HDL example design files using XST
- Runs Ngdbuild to consolidate the core netlist and the example design netlist into the NGD file containing the entire design
- Maps the design to the target technology
- Place-and-routes the design on the target device
- Performs static timing analysis on the routed design using Timing Analyzer (TRCE)
- Generates a bitstream
- Enables Netgen to run on the routed design to generate a VHDL or Verilog netlist (as appropriate for the Design Entry project setting) and timing information in the form of SDF files

The Xilinx tool flow generates several output and report files. These are saved in the following directory which is created by the implement script:

```
<project_dir>/<component_name>/implement/results
```

Simulation Scripts

Functional Simulation

The test scripts are a ModelSim, IUS, VCS, or ISIM macro that automate the simulation of the test bench. They are available from the following location:

```
<project_dir>/<component_name>/simulation/functional/
```

The test script performs the following tasks:

- Compiles the structural UniSim simulation model

- Compiles HDL Example Design source code
- Compiles the demonstration test bench
- Starts a simulation of the test bench
- Opens a Wave window and adds signals of interest
- Runs the simulation to completion

Example Design

Top Level Example Design

The following files describe the top-level example design for the SelectIO Interface Wizard core.

VHDL

```
project_dir>/<component_name>/example_design/<component_name>_exdes.vh
d
```

Verilog

```
project_dir>/<component_name>/example_design/<component_name>_exdes.v
```

The top-level example design adds a RAM on the user side of the I/O. In the event that a BUFPLL is chosen as the clocking primitive, it also instantiates and configures the required PLL_BASE. This allows the entire design to be synthesized and implemented in a target device to provide post place-and-route gate-level simulation.

Customizing the Example Design

The example design features parameters such as RAM size. By changing the parameters, the RAM can be increased or decreased in size. Initialization values can also be added to the RAM in the example design, rather than requiring that the RAM be loaded with values during operation.

Demonstration Test Bench

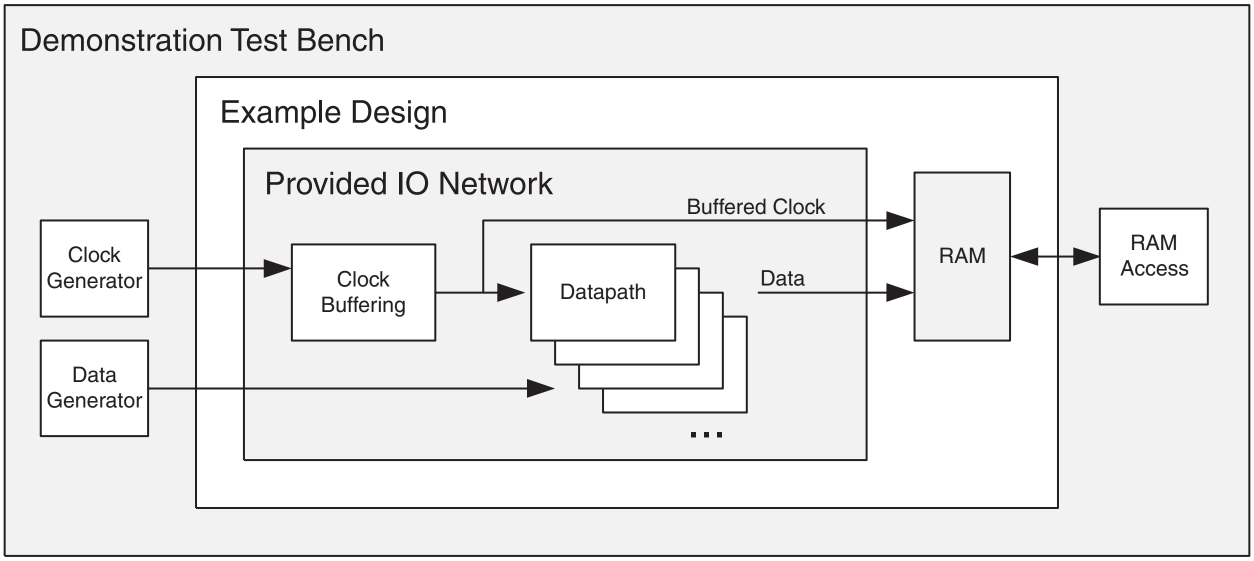

Demonstration Test Bench
Example Design
Provided IO Network
Buffered Clock
Clock Generator
Clock Buffering
Datapath
Data
RAM
RAM Access
Data Generator
...

Figure 6-1: **Demonstration Test Bench for the SelectIO Interface Wizard Core and Example Design**

The following files describe the demonstration test bench.

VHDL

```
project_dir>/<component_name>/simulation/<component_name>_tb.vhd
```

Verilog

```
project_dir>/<component_name>/simulation/<component_name>_tb.v
```

The demonstration test bench is a simple VHDL or Verilog program to exercise the example design and the core.

The demonstration test bench performs the following tasks:

- Generates input clock signals
- Applies a reset to the example design
- For an output only design, the example design memory is pre-loaded with data
- For a design with an input path, data is driven onto the pins and received in the memory
- For a design with bidirectional ports, the I/O input logic is reconfigured to be outputs
- For a design with an output path, data read from the memory and driven out on the pins