SelectIO Interface Wizard v5.1

LogiCORE IP Product Guide

Vivado Design Suite

PG070 April 6, 2016

XILINX
ALL PROGRAMMABLE™

Table of Contents

Introduction

The LogiCORE™ IP SelectIO™ Interface Wizard simplifies the integration of SelectIO technology into system designs for supported devices. The SelectIO Wizard creates an VHDL/Verilog HDL wrapper file that instantiates and configures I/O logic such as Input SERDES, Output SERDES and DELAY blocks to customer requirements. Additionally, it instantiates and configures the desired I/O clock primitive, connecting it to the instantiated I/O logic.

Features

- Supports input, output or bidirectional buses, and data buses up to 16 bits wide
- Creates clock circuitry required to drive I/O logic
- Optional data serialization support for each FPGA family
- Optional data and/or clock delay insertion
- Single and double data rates
- Predefined templates support multiple data bus standards: Chip-to-Chip, Camera receiver, Camera transmitter, digital visual interface (DVI) receiver, DVI transmitter and serial gigabit media independent interface (SGMII)
- Output from the SelectIO Wizard can be imported into the I/O planning project for further I/O attribute modifications
- Provides synthesizable example design and demonstration test bench to help with integration

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Zynq®-7000, Artix®-7, Virtex®-7, Kintex®-7
Supported User Interfaces	Native
Resource Utilization	Not Applicable
Provided with Core	
Design Files	Verilog
Example Design	Verilog
Test Bench	Verilog
Constraints File	Xilinx Design Constraints (XDC)
Simulation Model	None
Supported S/W Driver	N/A
Tested Design Flows(2)	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide
Synthesis	Synplify PRO Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

The SelectIO™ Interface Wizard provides source HDL that implements an I/O circuit for an input, output or bidirectional bus, including the buffer, any required delay elements, ISERDES and OSERDES elements, 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.

Applications

This solution is useful for multi-FPGA systems, like ASIC prototyping using FPGAs where serialization is required to accommodate thousands of signals on multiplexed I/Os in a single FPGA.

Licensing and Ordering Information

This Xilinx LogiCORE™ IP module is provided at no additional cost with the Xilinx® Vivado® Design Suite tool under the terms of the Xilinx End User License. Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information about pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

The SelectIO™ Interface Wizard provides source HDL that implements an I/O circuit for an input, output or bidirectional bus, including the buffer, any required delay elements, ISERDES and OSERDES elements, registers, and the I/O clock driver. The circuit is designed in two major components:

- Clock buffering and manipulation
- Datapath (implemented per-pin)

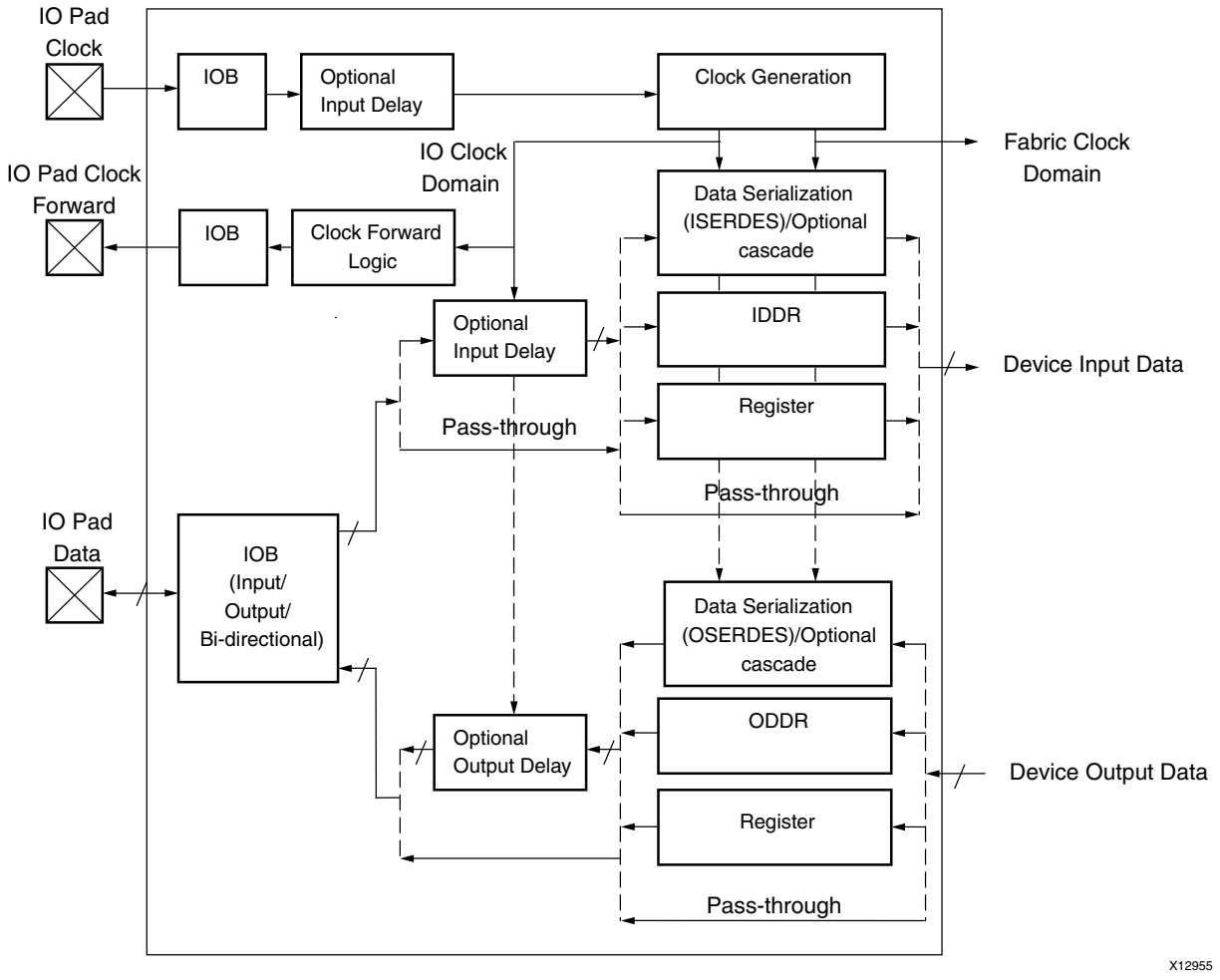

Figure 2-1: **SelectIO Interface Wizard Block Diagram**

Standards

SelectIO Wizard supports the following I/O standards.

- Single Ended signal: HSTL_I, HSTL_II, HSTL_III, HSTL_I_18, HSTL_II_18, HSTL_III_18, HSTL_I_12, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, LVCMOS12, SSTL15, SSTL18_I, SSTL18_II
- Differential signal: DIFF HSTL I, DIFF HSTL I 18, DIFF HSTL II, DIFF HSTL II 18, DIFF SSTL15, DIFF SSTL18 I, DIFF SSTL18 II, LVDS25, TMDS_33, MINI_LVDS_25, PPDS_25, BLVDS_25, LVDS, RSDS_25

Performance

The SelectIO Wizard can be configured for high performance depending on the selection and type of I/O standard and primitives.

Maximum Frequencies

For more details about frequencies, see the appropriate FPGA data sheet:

- *Virtex-7 FPGAs Data Sheet: DC and Switching Characteristics* (DS183) [Ref 2]
- *Kintex-7 FPGAs Data Sheet: DC and Switching Characteristics* (DS182) [Ref 3]
- *Artix-7 FPGAs Data Sheet: DC and Switching Characteristics* (DS181) [Ref 4]

Resource Utilization

Not Applicable — direct instantiation of primitives.

Port Descriptions

Table 2-1 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.

Table 2-1 defines the I/O circuit input and output ports.

Table 2-1: **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		
clock_enable	Input	Clock Enable: Input connected to the SelectIO primitives.
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 mixed-mode clock manager (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 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 BUFIO.
ref_clock	Input	Reference clock for IDELAYCTRL. Must come from BUFG.
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. For proper functionality, io_reset has to be deasserted when the clocks to SERDES are stable. Due to this requirement, io_reset must be deasserted after some cycles of clk_reset deassertion. For 8:1 serialization, sixteen cycles delay of I/O clock between clk_reset and io_reset can be used.
sync_reset	Input	Sync reset: Reset is connected to input double data rate (IDDR) when IDDR reset type is set to SYNC.
delay_reset	Input	Active-High synchronous reset for input delay
Pin Data Bus Ports		
data_in_from_pins	Input	Data in from pins: Single-ended input bus on the side of the pins.
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		

Table 2-1: **I/O Circuit Input and Output Port Descriptions *(Cont'd)***

Port	I/O	Description
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[n-1:0]	Input	Bit slip: Enable bit slip functionality on input data. Available on a input datapath and when enabled. This functionality is present for ISERDES in NETWORKING mode. 'n' indicates the datawidth.
tristate_output	Input	3-state Output: Disables the output path. This signal is synchronized with the input data. Available with a bidirectional datapath.
Variable Delay Ports		
delay_data_ce	Input	Delay data clock enable: Enable a delay change event for the datapath. This pin is provided for each of the IODELAYE2 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. This pin is provided for each of the IODELAYE2 components.
delay_tap_in [4:0]	Input	IODELAYE2 tap in signal: Counter value from FPGA logic for dynamically loadable tap value (CNTVALUEIN). This is provided for each of the IODELAYE2 components.
delay_tap_out[4:0]	Output	IODELAYE2 tap out signal: Counter value going to FPGA logic for monitoring tap value (CNTVALUEOUT). This is provided for each of the IODELAYE2 components.
delay_locked	Output	Locked signal from IDELAYCTRL

Notes:

1. Only a single-ended or differential input clock is required.

For details on I/O Interconnect and Clock primitives, see *7 Series FPGAs SelectIO Resources User Guide* (UG471) [Ref 12] and *7 Series Clocking Resources User Guide* (UG472) [Ref 13].

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 5]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 10]

Customizing and Generating the Core

This section includes information about using the Vivado Design Suite software to customize and generate the core.

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7].

Note: Figures in this chapter are illustrations of the Vivado Integrated Design Environment (IDE.) This layout might vary from the current version.

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

Data Bus Setup

Tab 1 of the IDE (Figure 3-1) sets up some general features for the data bus.

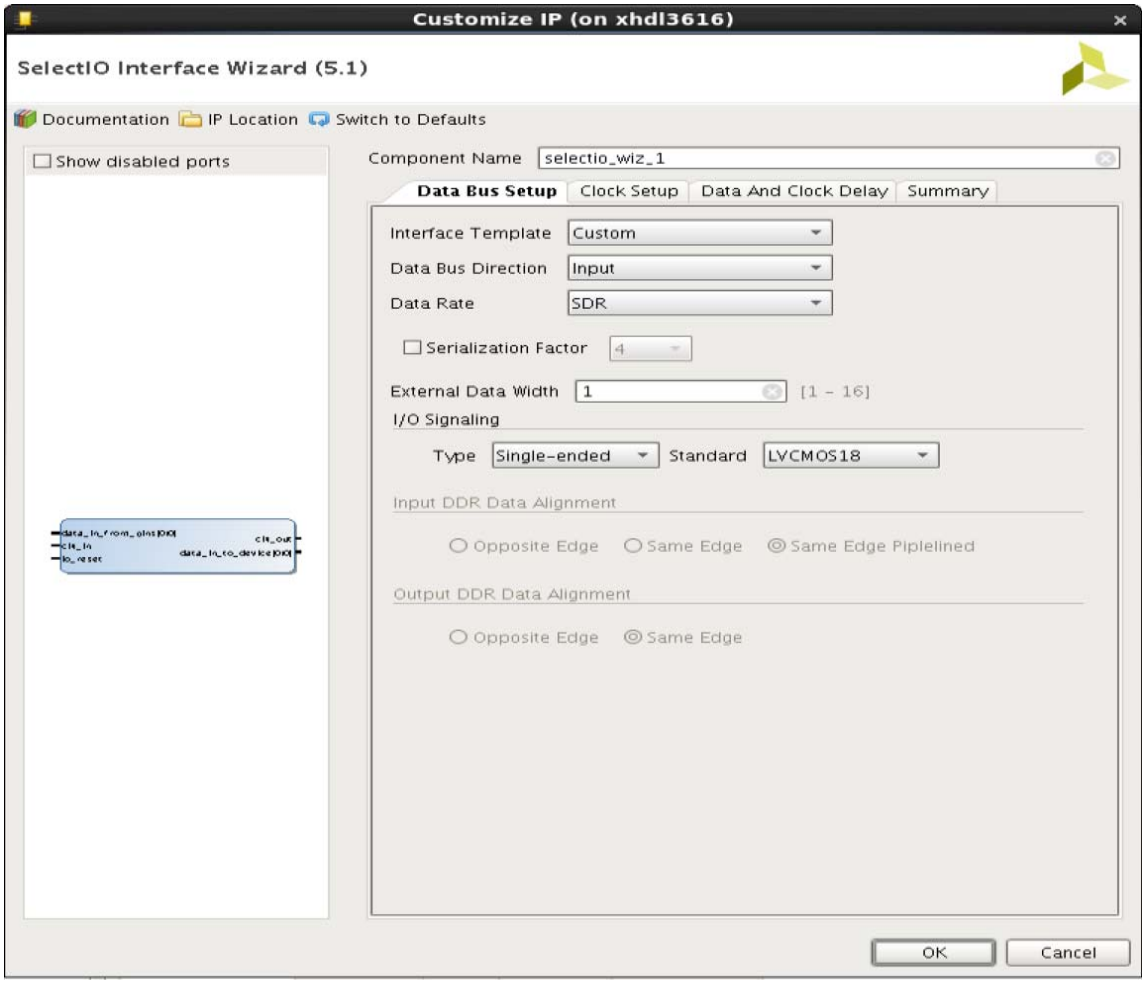

Figure 3-1: **Data Bus Setup - Tab 1**

Component Name

User selectable component name is available. Component names must not contain any reserved words in Verilog or VHDL.

Note: VHDL is only an instantiation template; the source files are Verilog.

Interface Template

SelectIO Wizard has some pre-configured I/O interfaces. Choosing one of these interfaces from the drop-down menu automatically sets the necessary parameters such as data bus direction, I/O signaling, and serialization factor.

The SelectIO Wizard supports SGMII, DVI receiver, DVI transmitter, Camera link receiver data bus format, Camera link transmitter and Chip-to-Chip interface. The SelectIO Interface Wizard only configures the data pins for all the interfaces mentioned above.

The listed options vary based on the device family selected.

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 causes restrictions later on in the configuration process.

The SelectIO Wizard core supports Input, Output, Bidirectional and Separate I/O buses.

The **Separate Inputs and Outputs** option creates independent input and output pins. Other configurable settings such as **Serialization Factor, External Data Width**, and delays are common to both inputs and outputs. For example, if **Separate Inputs and Outputs** is chosen and the **Serialization Factor** is set to 5, then this serialization factor of 5 applies to both Input SERDES as well as Output SERDES.

Data Rate

Select **SDR** if the data is clocked on the rising edge. If the incoming or outgoing data is clocked on both the edges, select **DDR**.

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

Serialization Factor

If **Serialization Factor** is selected, an ISERDESE2 and/or OSERDESE2 will be instantiated depending on the device selected.

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 10 or 14 is selected, two SERDES blocks per I/O will be instantiated because each SERDES is capable of a maximum serialization of 8:1. 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 the serialization factor are 2-8. When Data Rate is **DDR**, the serialization factor can be set to 4, 6, 8, 10, or 14.

If **Serialization Factor** 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 the NETWORKING type of Input Interface. For other interfaces such as MEMORY, MEMORY_QDR and MEMORY_DDR3, refer to the Memory Interface Generator (MIG) tool. Bitslip functionality is always enabled for NETWORKING mode. Tie this pin to logic 0 if not required.

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.

I/O Signaling

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

Input and Output DDR Data Alignment

Select this option if you want the SelectIO wizard to generate a clock forwarding logic. This option is only available when the bus direction is Output, Bidirectional or Separate Inputs and Outputs. By default the I/O settings of data are applied to the forwarded clock. You can update these settings by choosing an internal clock in the clocking strategy and selecting the **Config Clk Fwd** as shown in Figure 3-2. This allows the forwarded clock to be placed into a different bank than the bank selected for data.

- If **Opposite Edge** is selected, Output Q1 is present at the rising edge of the clock; Output Q2 is present at the falling edge of the clock.
- If **Same Edge** is selected, Output Q1 is present at the rising edge of current cycle; Output Q2 is present at the rising edge of next cycle.
- If **Same Edge Pipelined** is selected, Output pairs Q1 and Q2 are presented at the same time on the rising edge of the clock.

The **Input DDR Data Alignment** option is available when the bus direction is input or bidirectional. The **Output DDR Data Alignment** option is available when the bus direction is output or bidirectional.

Clock Setup

Tab 2 of the Vivado IDE allows you to configure the behavior of the clock. This section includes examples of External Clock and Internal Clock strategies.

External Clock

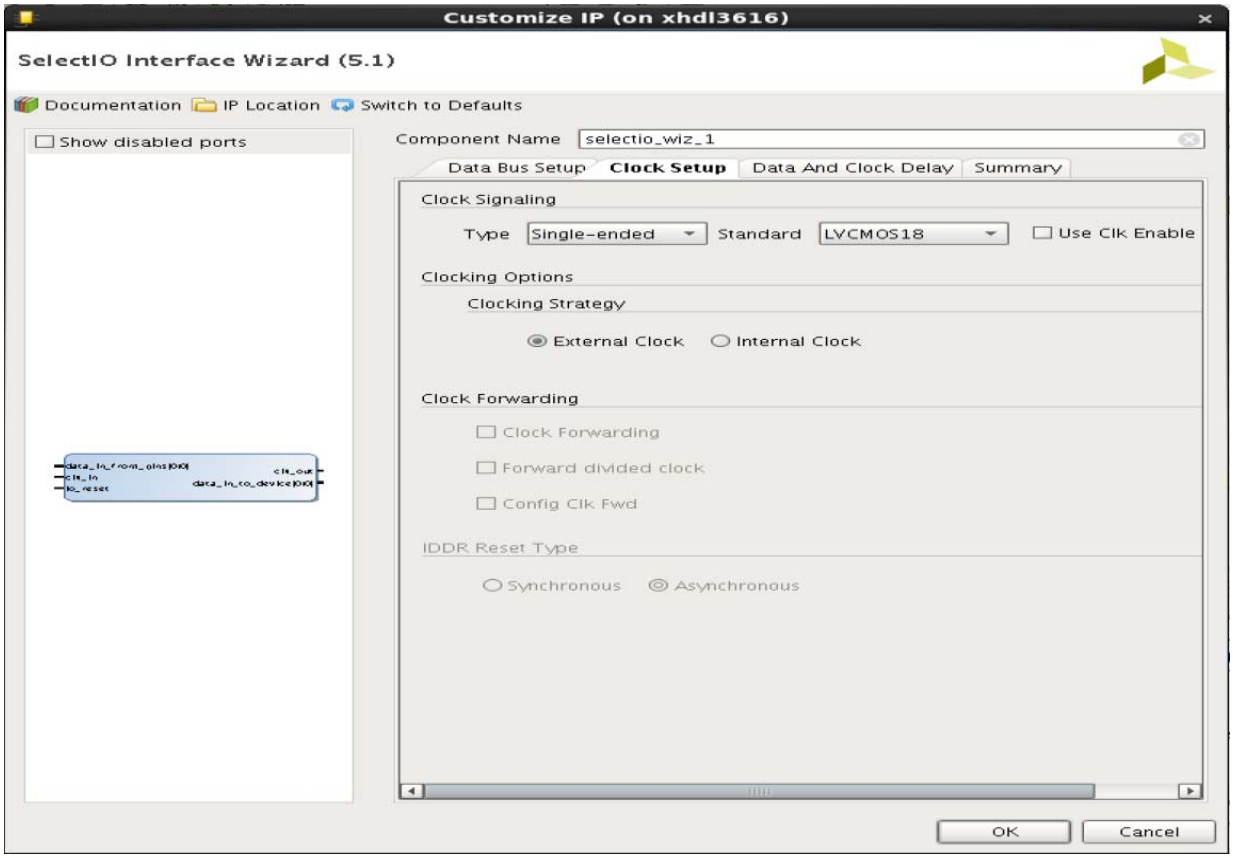

Figure 3-2: **Clock Setup with Clk Forwarding and Config Clk FWD**

If any delay is set on the output data path, the same delay is assigned to the forwarded clock so that the data and clock remain in sync.

Clock Forwarding

If clock forwarding is selected, then clock forwarding logic is added.

This option is only available when the bus direction is Output, Bidirectional or Separate Inputs and Outputs. By default the I/O settings of data are applied to the forwarded clock. You can update these settings by choosing an internal clock in clocking strategy and selecting **Config Clk Fwd**. This allows the forwarded clock to be placed into a different bank than the bank selected for data. Forward divided clock allows to forward frame clock if set, otherwise bit clock is forwarded

IDDR Reset Type

Input, Bidirectional, and Separate Bus Designs: The **IDDR Reset Type** is a new parameter with two IDDR reset type options: DDR Data and Serialization. When DDR is selected, and serialization is not, you can select the type of reset for the IDDR primitive. The default value is **Asynchronous**. However, if you select **Synchronous** for the reset, it should be synchronized with the clock driving the IDDR primitive.

Internal Clock

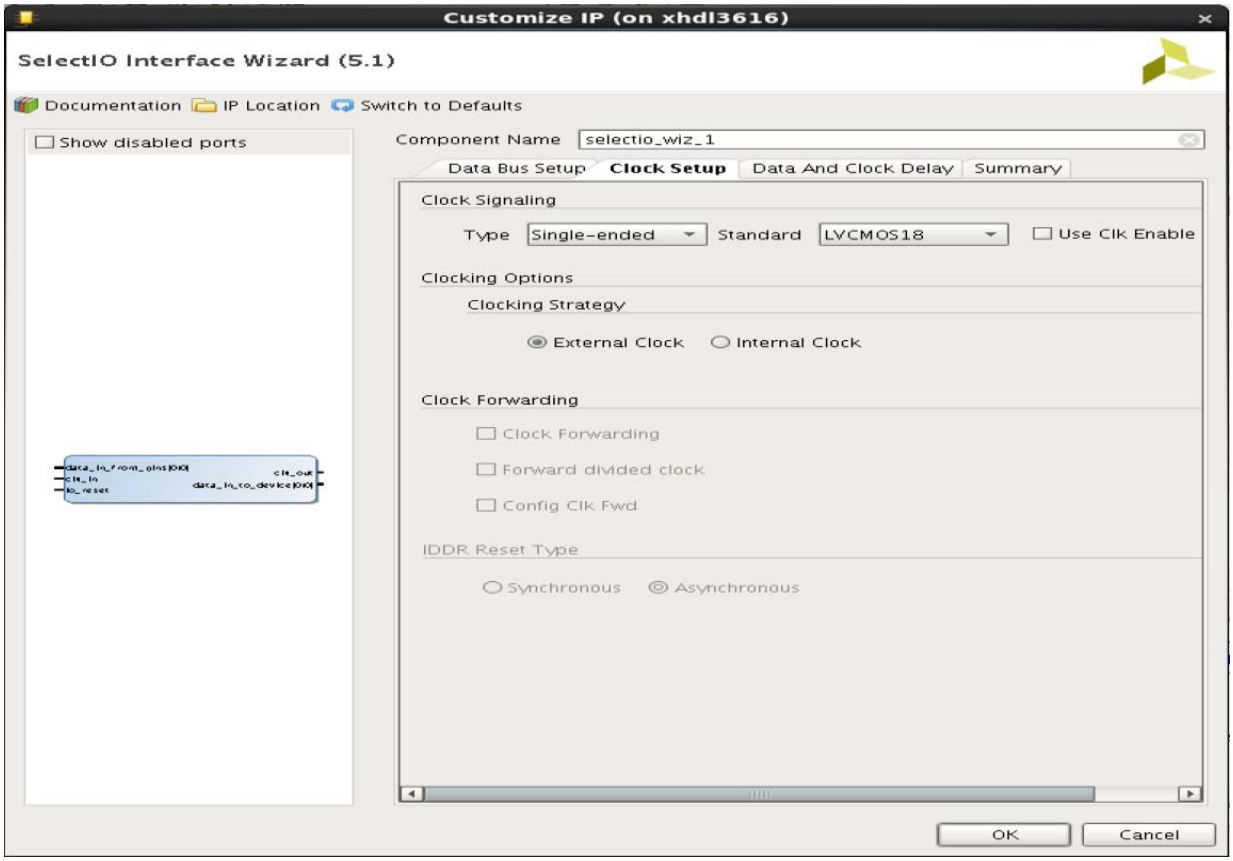

Figure 3-3: **Clock Setup - Tab 2**

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 signaling standard will be same as that of bus I/O standard.

The **Use Clock Enable** option provides a clock enable signal at the core top-level block for the SelectIO primitives. If this option is not selected, all clocks are enabled by default.

Clocking Options (Clocking Strategy)

If your clock comes from a pin, you should leave the input buffer as **External Clock** 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 the fabric, you will want to choose **Internal Clock**, but you need to be sure to instantiate an MMCM in the fabric to drive the clocks. Selecting the **Internal Clock** option overrides the **Clock Signaling** section. See the LogiCORE IP Clocking Wizard core and the *Clocking Wizard LogiCORE IP Product Guide* (PG065) [Ref 8], for assistance with MMCM instantiation and configuration.

If **Serialization Factor** 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 the data rate will be grouped by timeslice just as is it for serialization.

Data and Clock Delay

Tab 3 of the Vivado IDE (Figure 3-4) allows you to specify the type of delay for the data and clock.

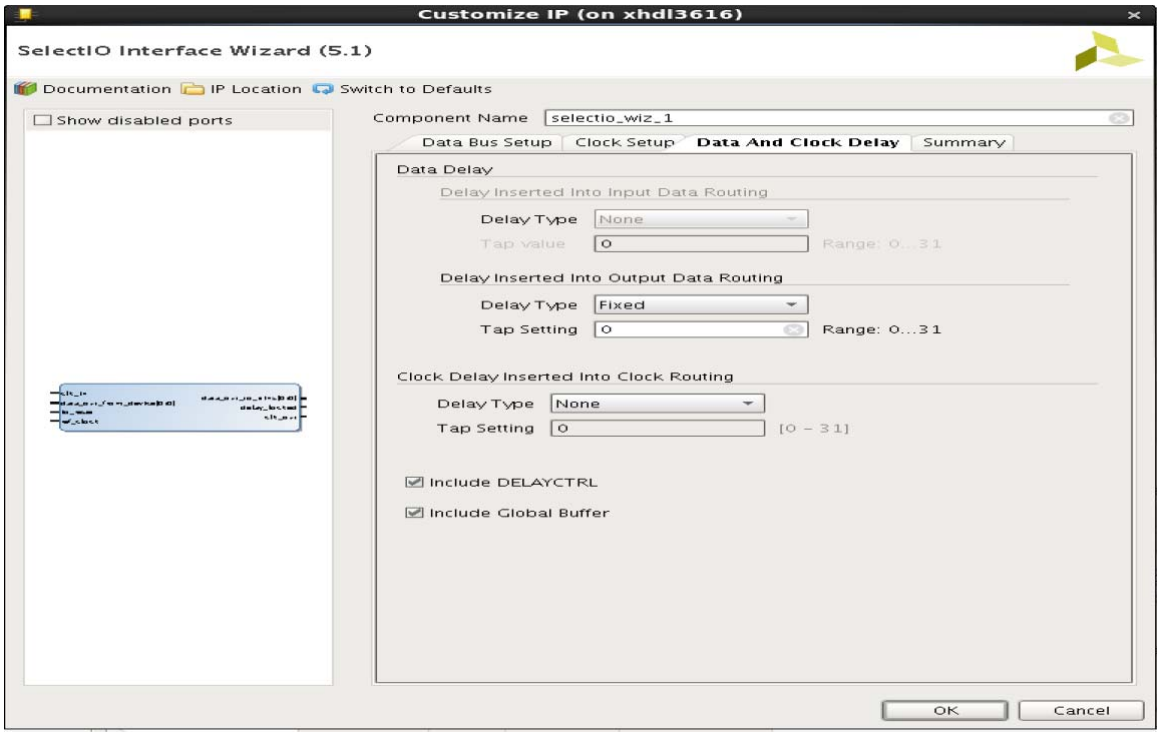

Figure 3-4: **Data and Clock Delay - Tab 3**

Delay Type

An IODELAYE2 is instantiated if **Default**, **Fixed**, **Variable** or **Variable loadable** delay is chosen. Generally, if data is delayed with **Fixed** or **Variable**, a clock delay is also required, given the high amount of insertion delay for the IODELAYE2 primitive. For a bidirectional bus, only specific combinations of Input and Output delay are possible.

Selecting the **Variable** or **Variable loadable** option allows control of each IODELAYE2 element individually. This means that the control signals of each IODELAYE2 element (for example, CE, INC, CNTVALUEIN, CNTVALUEOUT) are accessible. If you want to control all IODELAYE2s in the same way and at the same time, 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.

Include DELAYCTRL

Applicable only for **FIXED/VARIABLE** delays. If selected, **Include IODELAYCTRL** is instantiated in the design.

Included Global Buffer

If selected, BUFG is instantiated in the design. When **Include DELAYCTRL** is not selected, BUFG is not enabled for selection.

Summary Page

The summary tab lists the selected key parameters, such as the number of data I/Os, bus direction, serialization factor, buffers used, and the bus I/O standard.

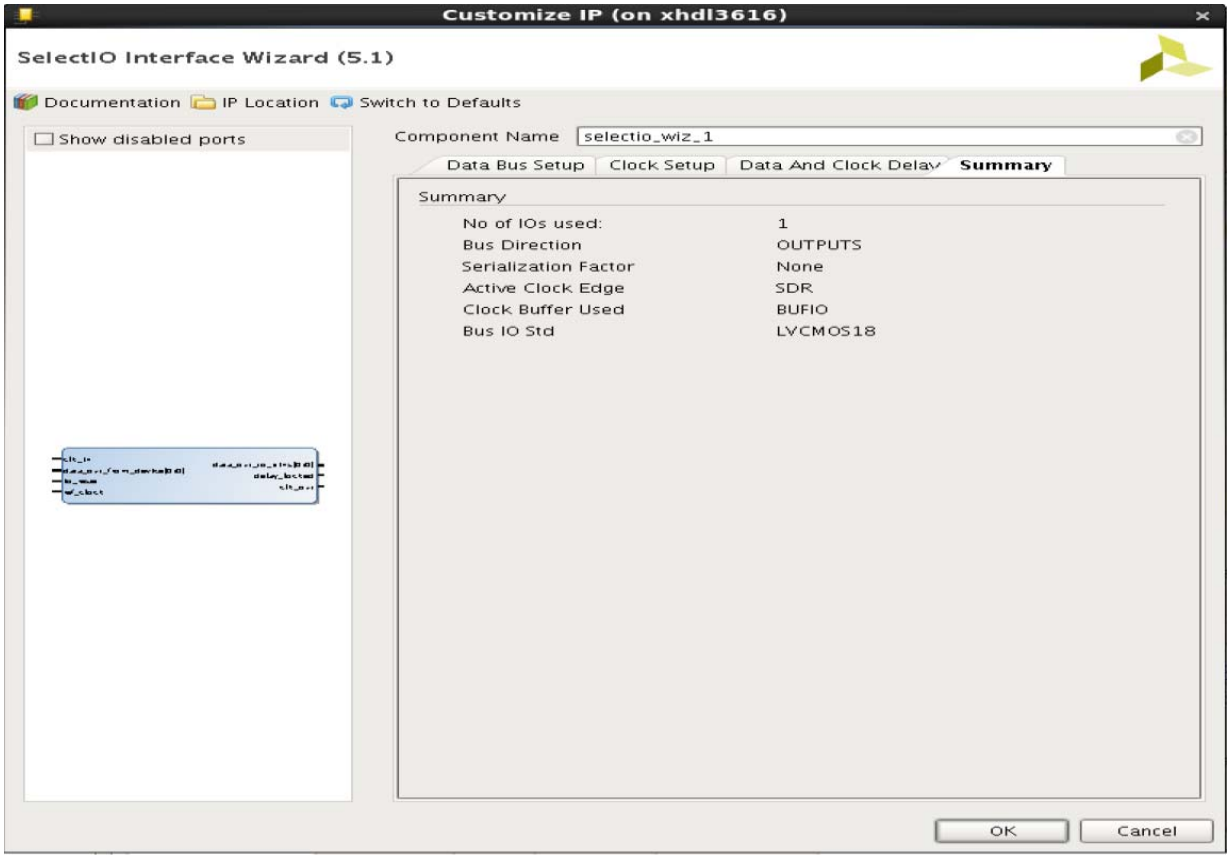

Figure 3-5: **Summary - Tab 4**

Generating the Core

After the desired configuration parameters have been selected, you can generate the SelectIO Wizard Interface core. To do so, click **Generate** that is located at the bottom of the Summary page.

User Parameters

Table 3-1 shows the relationship between the fields in the Vivado IDE and the User Parameters (which can be viewed in the Tcl Console).

Table 3-1: **Vivado IDE Parameter to User Parameter Relationship**

Vivado IDE Parameter/Value[1]	User Parameter/Value[1]	Default Value
Interface Template	USE_TEMPLATE	Custom
Data Bus Direction	BUS_DIR	Input
Data Rate	SELIO_ACTIVE_EDGE	SDR
Serialization Factor	SERIALIZATION_FACTOR	4
Serialization Factor	USE_SERIALIZATION	FALSE
External Data Width	SYSTEM_DATA_WIDTH	1
Type	BUS_SIG_TYPE	Single-ended
Standard	BUS_IO_STD	LVCMOS18
Input DDR Data Alignment	SELIO_DDR_ALIGNMENT	Same Edge Pipelined
Output DDR Data Alignment	SELIO_ODDR_ALIGNMENT	Same Edge
Type	CLK_SIG_TYPE	Single-ended
Standard	CLK_IO_STD	HSTL1
Use Clk Enable	CLK_EN	FALSE
Clocking Strategy	SELIO_CLK_BUF	BUFIO
Clock Forwarding	CLK_FWD	FALSE
Forward divided clock	CLK_FWD_SER	FALSE
Config Clk Fwd	CONFIG_CLK_FWD	FALSE
IDDR Reset Type	IDDR_RST_TYPE	Asynchronous
Delay Type (Data)	SELIO_BUS_IN_DELAY	None
Tap Value (Data)	SELIO_BUS_IN_TAP	0
Delay Type (Data, Output)	SELIO_BUS_OUT_DELAY	None
Tap Setting (Datam Output)	SELIO_BUS_OUT_TAP	0
Delay Type (Clk)	CLK_DELAY	None
Tap Setting	CLK_TAP	0
Include DELAYCTRL	INCLUDE_IDELAYCTRL	TRUE
Include Global Buffer	INCLUDE_IDELAYCTRL_BUFG	TRUE

Notes:

1. Parameter values are listed in the table where the Vivado IDE parameter value differs from the user parameter value.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Constraining the Core

This section contains any applicable constraints for the core.

Required Constraints

For a single-ended clock, use the following constraint:

```
create_clock -period 10 [get_ports clk_in]
set_input_jitter [get_clocks -of_objects [get_ports clk_in] 0.1
```

For a differential clock, use the following constraint:

```
create_clock -period 10 [get_ports clk_in_p]
set_input_jitter [get_clocks -of_objects [get_ports clk_in_p] 0.1
```

Device, Package, and Speed Grade Selections

Supports all Zynq®-7000 and 7 series devices, packages, and speed grades.

Clock Frequencies

Default input clock frequency set is 100 MHz. Modify this frequency as required.

Clock Management

Refer to the example design HDL for clock generation for the SelectIO Interface Wizard.

Clock Placement

There are no placement constraints for the SelectIO Interface Wizard.

Banking

There are no banking constraints for the SelectIO Interface Wizard.

Transceiver Placement

There are no transceiver constraints for the SelectIO Interface Wizard.

I/O Standard and Placement

The I/O Standard setting is available in the Vivado IDE.

Simulation

For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 10].

IMPORTANT: *For cores targeting 7 series or Zynq®-7000 devices, UNIFAST libraries are not supported. Xilinx IP is tested and qualified with UNISIM libraries only.*

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Designing with the Core

This chapter includes guidelines and additional information to make designing with the core easier.

Clock Buffering and Manipulation

The SelectIO Wizard supports the use of a BUFG or BUFIO2 for clocking the I/O logic. An example circuit illustrating a BUFIO2 primitive with input data is illustrated in Figure 4-1. Insertion delay can be added for the input clock.

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

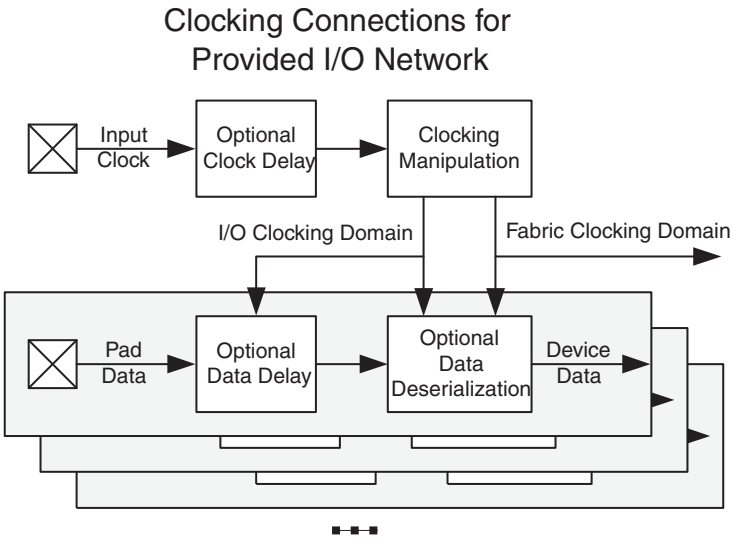

Figure 4-1: **Provided I/O Circuit**

Datapath

The SelectIO Wizard assists in instantiating and configuring the components within the I/O interconnect block.

You 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 data flow graph for an input bus is shown in Figure 4-2. For an output bus, the components are similar, but the data flows in the other direction. For a bidirectional bus, there is both an input and output path, although there is only one IODELAY2 or IODELAYE1 element.

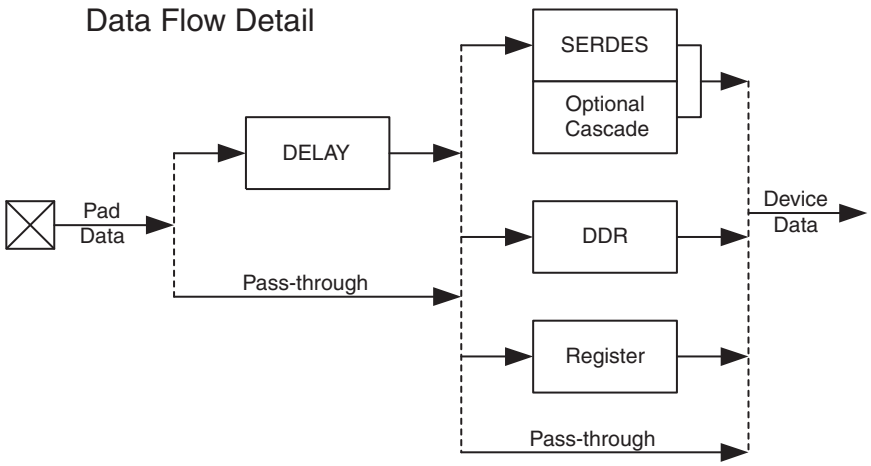

Figure 4-2: **Flow in the I/O Input Datapath**

Clocking

If the clock comes from a pin, leave the input buffer as **Source Synchronous** for the most flexible functionality. Selecting this option instantiates the necessary circuitry of BUFIO and BUFR.

When the clock comes from the fabric, choose **Fabric Clock**; ensure that you also instantiate a MMCME2 in the fabric to drive the clocks. Selecting the **Fabric Clock** option overrides the **Clock Signaling** section. See the example design for reference.

By default, the input clock is set to 100 MHz. If clock forwarding is selected then clock forwarding logic is added.

Resets

Active-High `IO_RESET` is provided to reset the I/O blocks. Active-High `CLK_RESET` is provided to reset the clocking logic.

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 Vivado® Design Suite, the purpose and contents of the provided scripts, and the contents of the example HDL wrappers.

In the IP Catalog project, clicking **Open IP Example Design** in Vivado IDE or typing the open_example_project command (`get_ips <component_name>`) in the TCL console invokes a separate example design project. In this new project `<component_name>_exdes` is the top module for synthesis, and `<component_name>_tb` is the top module for simulation. The implementation or simulation of the example design can be run from the example project.

Top-Level Example Design

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

```
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/example_design/
<component_name>_exdes.v
```

The top-level example design implements a loop-back strategy to verify the I/O logic implementation. The example design generates data that is loop-backed to itself through the device under test (DUT). The data received is verified, and a status signal is generated accordingly. The example design instantiates a MMCME2 to generate various clocks. The entire design is synthesized and implemented in a target device.

Note: The example design instantiates *BUFG/IODELAY_CTRL if **Include DELAYCTRL** is not selected in the Vivado IDE during IP customization.

Simulation

SelectIO Wizard includes a simulation test bench for the example design. You can launch the simulation in the example design project to verify the functionality of the core.

For more details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Test Bench

This chapter contains information about the test bench provided in the Vivado® Design Suite.

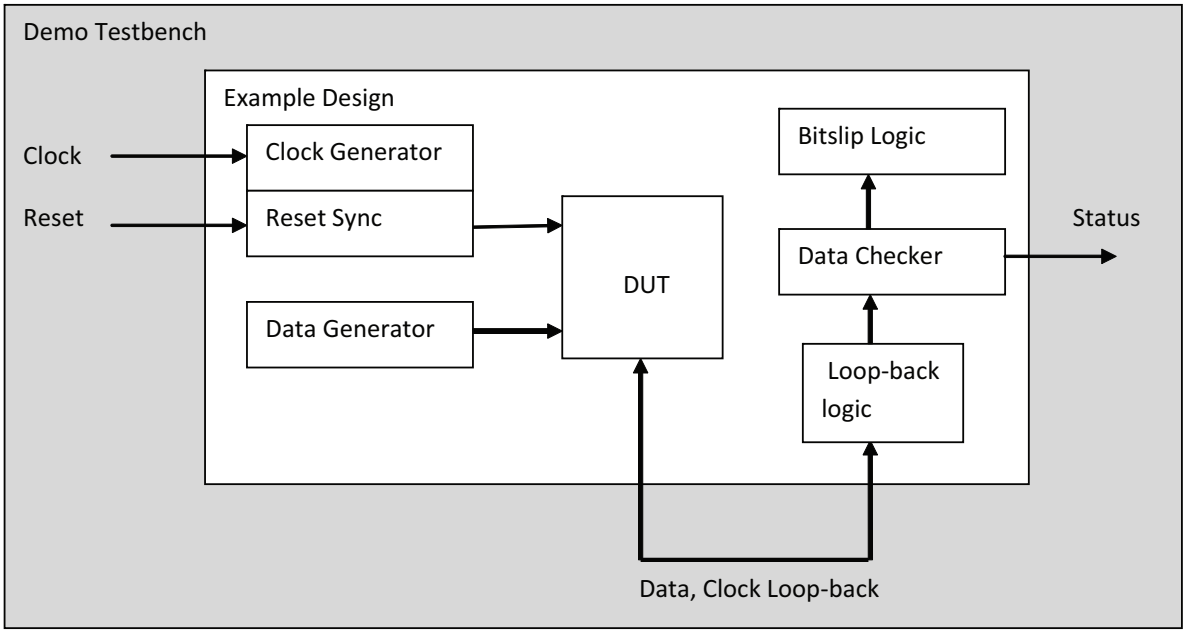

Figure 6-1: **Demonstration Test Bench for the SelectIO Interface Wizard Core and Example Design**

The following file describes the demonstration test bench.

```
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/simulation/
<component_name>_tb.v
```

The demonstration test bench is a simple 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 any type of Bus I/O direction, the example design uses a loop-back architecture. If the design generated is for input direction, the example design will have an output logic to drive the data and vice-versa. The loop-back connection is done in the test bench.
- The example design has a bitslip logic that generates the required amount of bitslip pulses for ISERDES to get the right data. When the ISERDESs are locked, the design then starts checking for the output of the ISERDES.

Appendix A

Verification, Compliance, and Interoperability

This appendix includes information about how the IP is tested.

Simulation

Verified with all the supported simulators.

Hardware Testing

Hardware testing is performed for all the features on the Kintex®-7 KC705 Evaluation Kit using the example design.

Migrating and Upgrading

This appendix contains information about migrating a design from ISE® to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impacts to user logic are included.

Migrating to the Vivado Design Suite

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 9].

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Parameter Changes

No parameter changes.

Port Changes

No parameter changes.

Other Changes

Forwarded divided clock option added in Vivado IDE.

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools.

Finding Help on Xilinx.com

To help in the design and debug process when using the SelectIO Wizard, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the SelectIO Wizard. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core are listed below, and can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as:

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the SelectIO Wizard

AR: 54649

Technical Support

Xilinx provides technical support in the Xilinx Support web page for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx.

The Vivado logic analyzer is used to interact with the logic debug IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 11].

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The debugging tool is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the debugging tool for debugging the specific problems.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `locked` port.
- If your outputs go to 0, check your licensing.
- Data Settings: Ensure that the data settings are correct for the SERDES/DDR in the Data Bus setup.
- Clock Settings: Check that the clock signaling and strategy are correct in the clock setup. For source synchronous designs, use clock forwarding for the Output Bus.
- Delay Settings:
  - If the design does not work in post-place and route timing simulation, set different delay values on Input/Output delay logic after making delay type to "FIXED."
  - For a runtime delay value update, set the delay type to "VARIABLE" or "VAR_LOAD" and create a state machine to generate the delay control signals to match the input and output data pattern.
  - For timing closure, set the SLEW RATE and DRIVE STRENGTH on the forwarded clock and output/input data through IO Planner or XDC constraints.

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

References

1. *7 Series FPGAs Overview* (DS180)
2. *Virtex-7 FPGAs Data Sheet: DC and Switching Characteristics* (DS183)
3. *Kintex-7 FPGAs Data Sheet: DC and Switching Characteristics* (DS182)
4. *Artix-7 FPGAs Data Sheet: DC and Switching Characteristics* (DS181)
5. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
6. *Vivado Design Suite User Guide: Designing with IP* (UG896)
7. *Vivado Design Suite User Guide: Getting Started* (UG910)
8. *Clocking Wizard LogiCORE IP Product Guide* (PG065)
9. *ISE to Vivado Design Suite Migration Guide* (UG911)
10. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
11. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
12. *7 Series FPGAs SelectIO Resources User Guide* (UG471)
13. *7 Series FPGAs Clocking Resources User Guide* (UG472)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/06/2016	5.1	• Added Forward divided clock option to Vivado IDE and User Parameters. • Updated Figures 3-1 through 3-5. • Added information for Input and Output DDR Alignment section. • Added Clock Forwarding section.
09/30/2015	5.1	• Added Include DELAYCTRL and BUFG menu items.
04/01/2015	5.1	• Added User Parameters section. • Updated Simulation section.
04/02/2014	5.1	Added additional hardware debugging details in the Hardware Debug section.
10/02/2013	5.1	• Updated doc version number to match core version number. • Updated for clock enable logic and IOSTANDARD configuration for forwarded clock.
03/20/2013	2.0	• Updated core to v5.0. • Added support for Zynq®-7000 devices.
12/18/2012	1.1	• Updated GUI details in “Customizing and Generating the Core.” • Updated Vivado Design Suite to v2012.4. No other documentation changes.
07/25/2012	1.0	Initial Xilinx release as a product guide. Replaces DS746, *LogiCORE IP SelectIO Interface Wizard Data Sheet* and GSG700, *LogiCORE IP SelectIO Interface Wizard Getting Started Guide*.

Please Read: Important Legal Notices