# Table of Contents

**IP Facts**

**Chapter 1: Overview**
- Applications .................................................. 5
- Licensing and Ordering Information .................................. 5

**Chapter 2: Product Specification**
- Standards Compliance .................................................. 7
- Performance .................................................. 7
- Port Descriptions .................................................. 7

**Chapter 3: Designing with the Core**
- Clock Buffering and Manipulation ........................................ 10
- Datapath .................................................. 11
- Clocking .................................................. 11
- Resets .................................................. 12

**Chapter 4: Customizing and Generating the Core**
- GUI .................................................. 13
- Output Generation .................................................. 20

**Chapter 5: Constraining the Core**
- Required Constraints .................................................. 21
- Device, Package, and Speed Grade Selections ............................................... 21
- Clock Frequencies .................................................. 21
- Clock Management .................................................. 21
- Clock Placement .................................................. 22
- Banking .................................................. 22
- Transceiver Placement .................................................. 22
- I/O Standard and Placement .................................................. 22

**Chapter 6: Detailed Example Design**
- Directory and File Contents .................................................. 23
Example Design ................................................................. 26
Demonstration Test Bench .................................................. 27
Simulation ................................................................. 28

Appendix A: Verification, Compliance, and Interoperability
   Simulation ................................................................. 29
   Hardware Testing .................................................. 29

Appendix B: Migrating
   Parameter Changes in the XCI File .................................. 30
   Port Changes ........................................................... 30
   Functionality Changes .............................................. 30

Appendix C: Additional Resources
   Xilinx Resources .......................................................... 31
   References .............................................................. 31
   Technical Support .................................................... 31
   Revision History .......................................................... 32
   Notice of Disclaimer ................................................... 32
Introduction

The LogiCORE™ IP SelectIO™ Interface Wizard simplifies the integration of SelectIO technology into system designs for 7 series devices. The Wizard creates an HDL file (Verilog or VHDL) 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 busses, and data busses 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 rate data
- Predefined templates support multiple data bus standards: Chip-to-Chip, Camera receiver, Camera transmitter, DVI receiver, DVI transmitter and SGMII
- Output from the wizard can be imported into the PlanAhead™ design tool for further I/O attribute modifications
- Provides synthesizable example design and demonstration test bench to help with integration

LogiCORE IP Facts Table

<table>
<thead>
<tr>
<th>Core Specifics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Device Family (1)</td>
<td>Artix™-7, Virtex®-7, Kintex™-7</td>
</tr>
<tr>
<td>Supported User Interfaces</td>
<td>Native</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Provided with Core</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Files</td>
<td>Verilog and VHDL</td>
</tr>
<tr>
<td>Example Design</td>
<td>Verilog and VHDL</td>
</tr>
<tr>
<td>Test Bench</td>
<td>Verilog and VHDL</td>
</tr>
<tr>
<td>Constraints File</td>
<td>XDC (Xilinx Design Constraints)</td>
</tr>
<tr>
<td>Simulation Model</td>
<td>None</td>
</tr>
<tr>
<td>Supported S/W Driver</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Tested Design Flows (2)

<table>
<thead>
<tr>
<th>Design Entry</th>
<th>Vivado™ Design Suite 2012.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>Mentor Graphics ModelSim, Cadence Incisive Enterprise Simulator (IES), Synopsys VCS and VCS MX, XSim</td>
</tr>
<tr>
<td>Synthesis</td>
<td>Synplify PRO, Vivado Synthesis</td>
</tr>
</tbody>
</table>

Support

Provided by Xilinx @ www.xilinx.com/support

Notes:
1. For a complete listing of supported devices, see the release notes for this core.
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 IO's 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.
Chapter 2

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, and datapath, which is implemented per-pin.

Figure 2-1: SelectIO Interface Wizard Block Diagram
Standards Compliance

SelectIO Wizard supports following IO 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
- For 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 IO standard and primitives.

Maximum Frequencies

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

- DS183, *Virtex-7 FPGAs Data Sheet: DC and Switching Characteristics*
- DS182, *Kintex-7 FPGAs Data Sheet: DC and Switching Characteristics*
- DS181, *Artix-7 FPGAs Data Sheet: DC and Switching Characteristics*

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 to the user.

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

<table>
<thead>
<tr>
<th>Port</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clock Ports</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLK_IN</td>
<td>Input</td>
<td>Clock in: Single-ended input clock. Available when a single-ended clock is selected.</td>
</tr>
<tr>
<td>CLK_IN_P</td>
<td>Input</td>
<td>Clock in Positive and Negative. Available when a differential clock source is selected.</td>
</tr>
<tr>
<td>CLK_IN_N</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>CLK_OUT</td>
<td>Output</td>
<td>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.</td>
</tr>
<tr>
<td>CLK_DIV_IN</td>
<td>Input</td>
<td>Clock divided in: Input clock for serialization in the I/O Logic. Available when serialization is chosen, and the clock primitive is BUFPLL/MMCM.</td>
</tr>
<tr>
<td>CLK_DIV_OUT</td>
<td>Output</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Reset Ports</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLK_RESET</td>
<td>Input</td>
<td>Clock reset: Reset connected to clocking elements in the circuit.</td>
</tr>
<tr>
<td>IO_RESET</td>
<td>Input</td>
<td>I/O reset: Reset connected to all other elements in the circuit.</td>
</tr>
<tr>
<td>SYNC_RESET</td>
<td>Input</td>
<td>Sync reset: Reset is connected to IDDR when IDDR reset type is set to SYNC.</td>
</tr>
<tr>
<td><strong>Pin Data Bus Ports</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA_IN_FROM_PINS</td>
<td>Input</td>
<td>Data in from pins: Single-ended input bus on the side of the pins.</td>
</tr>
<tr>
<td>DATA_IN_FROM_PINS_P</td>
<td>Input</td>
<td>Data in from pins positive and negative: Differential input bus on the side of the pins.</td>
</tr>
<tr>
<td>DATA_IN_FROM_PINS_N</td>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>DATA_OUT_TO_PINS</td>
<td>Output</td>
<td>Data out to pins: Single-ended output bus on the side of the pins.</td>
</tr>
<tr>
<td>DATA_OUT_TO_PINS_P</td>
<td>Output</td>
<td>Data out to pins positive and negative: Differential output bus on the side of the pins.</td>
</tr>
<tr>
<td>DATA_OUT_TO_PINS_N</td>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>DATA_TO_AND_FROM_PINS</td>
<td>Input/Output</td>
<td>Data to and from pins: Single-ended bidirectional data bus on the side of the pins.</td>
</tr>
<tr>
<td>DATA_TO_AND_FROM_PINS_P</td>
<td>Input/Output</td>
<td>Data to and from pins positive and negative: Differential bidirectional data bus on the side of the pins.</td>
</tr>
<tr>
<td>DATA_TO_AND_FROM_PINS_N</td>
<td>Input/Output</td>
<td></td>
</tr>
<tr>
<td><strong>Device Data Bus Ports</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA_IN_TO_DEVICE</td>
<td>Output</td>
<td>Data in to device: Input bus on the side of the device.</td>
</tr>
<tr>
<td>DATA_OUT_FROMDEVICE</td>
<td>Input</td>
<td>Data out from device: Output bus on the side of the device.</td>
</tr>
<tr>
<td><strong>Control and Status Ports</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BITSLIP</td>
<td>Input</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 2-2 provides a list of resources for specific I/O interconnect and clock primitives.

Table 2-2:  SelectIO and Clock Resources

<table>
<thead>
<tr>
<th>I/O Primitives</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interconnect</td>
<td>7 Series FPGAs SelectIO Resources User Guide</td>
</tr>
<tr>
<td>Clock</td>
<td>7 Series Clocking Resources User Guide</td>
</tr>
</tbody>
</table>
Designing with the Core

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

Clock Buffering and Manipulation

The 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 3-1.

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 on behalf of the user (except in the case of a BUFG, which does not support serialized/deserialized data).

Figure 3-1: Provided I/O Circuit
Datapath

The 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 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.

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 and configure the same.
In the event the clock comes from fabric, choose Fabric Clock. However, be sure to 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 100 MHz input clock is set.

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

---

**Resets**

Active High `IO_RESET` is provided to reset the IO blocks. Active High `CLK_RESET` is provided for resetting the clocking logic.
Chapter 4

Customizing and Generating the Core

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

GUI

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

Tab 1 of the GUI (Figure 4-1) sets up some general features for the data bus.
GUI

Component Name

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

Interface Template

SelectIO Wizard has some pre-configured IO interfaces. Choosing one of these interfaces from the drop-down menu will automatically set the necessary parameters such data bus direction, I/O signalling, serialization factor etc.

Currently the wizard supports SGMII, DVI receiver, DVI transmitter, Camera link receiver data bus format, Camera link transmitter and Chip-to-Chip interface. SelectIO Interface Wizard would only configure 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 will cause restrictions later on in the configuration process.
SelectIO Wizard supports Inputs, Outputs, Bidirectional and Separate IO buses.

Separate Inputs and Outputs create independent Inputs and Outputs pins. Other configurable settings such as Serialization factor, 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” would apply to both Input SERDES as well as Output SERDES.

**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.

**Clock Forwarding**

Select this option if you want the SelectIO wizard to generate a clock forwarding logic. This option is only available when bus direction is Output, Bidirectional or Separate Inputs and Outputs.

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.

**Serialization**

If Use Serialization is selected, an ISERDESE2 and/or OSERDESE2 will be instantiated for the user 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 for the user 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 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 NETWORKING mode. User should 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.

**IDDR Reset**

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 ASYNC. However, if you select SYNC for the reset, it should be synchronized with the clock driving the IDDR primitive.

**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.

**Clock Setup**

Tab 2 of the GUI (Figure 4-2) allows you to configure the behavior of the clock.
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 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.

**Data and Clock Delay**

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

![Data and Clock Delay - Tab 3](image)

**Delay Type**

An IODELAYE2 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 IODELAYE2 primitive. For a bidirectional bus, only specific combinations of Input and Output delay are possible.

Selecting Variable or Var_loadable option will enable the user to control each IODELAYE2 element individually. This means that the control signals of each IODELAYE2 element (for example, CE, INC, CNTVALUEIN, CNTVALUEOUT) would be accessible to the user. If the user wishes to control all IODELAYE2s 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 Delay

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.

Summary Page

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

Generating the Core

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

See Directory and File Contents in Chapter 6 for details about output files.
Chapter 5

Constraining the Core

This chapter contain any applicable constraints for the core.

Required Constraints

For a single ended clock, use the following constraint:

```tcl
create_clock -name CLK_IN -period 10 [get_ports CLK_IN]
set_input_jitter CLK_IN 0.1
```

For a differential clock, use the following constraint:

```tcl
create_clock -name CLK_IN_P -period 10 [get_ports CLK_IN_P]
set_input_jitter CLK_IN_P 0.1
```

Device, Package, and Speed Grade Selections

Supports all 7 series devices, packages and speed grades.

Clock Frequencies

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

Clock Management

Refer to 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 GUI.
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 Vivado™ tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench.

In the IP Catalog project, clicking Open IP Example Design in GUI 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.

Directory and File Contents

- `<project_name>/project_name.srs/sources_1/ip/`  
  Top-level project directory; name is user-defined
- `<project_name>/project_name.srs/sources_1/ip/<component name>`  
  Core release notes file
- `<component name>/example design`  
  Verilog and VHDL design files
- `<component name>/implement`  
  Create this directory to run the timing simulation
- `implement/results`  
  Copy implementation results into this directory to run the timing simulation using scripts
- `<component name>/simulation`  
  Simulation scripts
- `simulation/functional`  
  Functional simulation files
- `simulation/timing`  
  Timing simulation files

The SelectIO Interface Wizard core directories and their associated files are defined below.
<project_name>/<project_name>.srcs/sources_1/ip/

The <project directory> contains all the Vivado tool project files.

**Table 6-1: Project Directory**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project_dir&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;component_name&gt;.v[hd]</td>
<td>Verilog or VHDL top level wrapper.</td>
</tr>
<tr>
<td>&lt;component_name&gt;_selectio_wiz.v[hd]</td>
<td>Verilog or VHDL source code.</td>
</tr>
<tr>
<td>&lt;component_name&gt;.xci</td>
<td>Vivado tools project-specific option file; can be used as an input to the Vivado tools.</td>
</tr>
<tr>
<td>&lt;component_name&gt;_flist.txt</td>
<td>List of files delivered with the core.</td>
</tr>
<tr>
<td>&lt;component_name&gt;.{veo</td>
<td>vho}</td>
</tr>
<tr>
<td>&lt;component_name&gt;.xdc</td>
<td>Constraint file for core.</td>
</tr>
</tbody>
</table>

**<project_name>/<project_name>.srcs/sources_1/ip/ <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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project_name&gt;/&lt;project_name&gt;.srcs/sources_1/ip/&lt;component_name&gt;</td>
<td></td>
</tr>
<tr>
<td>selectio_wiz_v4_2_readme.txt</td>
<td>SelectIO Interface Wizard readme file.</td>
</tr>
</tbody>
</table>

**<component name>/example design**

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;component_name&gt;/example_design</td>
<td></td>
</tr>
<tr>
<td>&lt;component_name&gt;_exdes.v[hd]</td>
<td>Implementable Verilog or VHDL example design.</td>
</tr>
<tr>
<td>&lt;component_name&gt;_exdes.xdc</td>
<td>Constraint file for example design.</td>
</tr>
</tbody>
</table>
<component name>/implement
This directory must be created by the user for the timing simulation.

implement/results
This directory must be created by the user for the timing simulation.

<component name>/simulation
The simulation directory contains the simulation test bench for the example design.

Table 6-4: Simulation Directory

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;component name&gt;/simulation</td>
<td></td>
</tr>
<tr>
<td>&lt;component name&gt;_tb.v[hd]</td>
<td>Demonstration test bench.</td>
</tr>
</tbody>
</table>

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

Table 6-5: Functional Directory

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;component name&gt;/simulation/functional</td>
<td>Contains simulation scripts and waveform formats.</td>
</tr>
</tbody>
</table>

simulation/timing
The timing directory contains the timing simulation scripts provided with the core.

Table 6-6: Timing Directory

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;component name&gt;/simulation/timing</td>
<td>Contains the timing scripts and waveform.</td>
</tr>
</tbody>
</table>
Example Design

Top Level Example Design

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

VHDL

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

Verilog

```
<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 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.
Demonstration Test Bench

The following files describe the demonstration test bench.

**VHDL**

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

**Verilog**

```<project_name>/<project_name>.srcs/sources_1/ip/<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 any type of Bus I/O direction, the example design uses a loop-back architecture. If the design generated is for input direction, then 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. Once the ISERDESs are locked, the design then starts checking for the output of the ISERDES.

Simulation

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

```
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/simulation/
  functional/
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/simulation/timing/
```

The test script performs the following tasks:

• Compiles the structural UniSim/SimPrim simulation model
• Compiles Example Design source code or netlist
• Compiles the demonstration test bench
• Starts a simulation of the test bench
• Runs the simulation to completion
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 Kintex-7 KC705 Evaluation Kit using the example design.
Appendix B

Migrating

This appendix describes migrating from older versions of the IP to the current IP release.

For details about migrating from the ISE Design Suite to the Vivado Design Suite, see UG911, *Vivado Design Suite Migration Methodology Guide*.

Parameter Changes in the XCI File

No parameter changes occurred in this release.

Port Changes

No port changes occurred in this release.

Functionality Changes

No functionality changes occurred in this release.
Appendix C

Additional Resources

**Xilinx Resources**

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:


For a glossary of technical terms used in Xilinx documentation, see:

[www.xilinx.com/company/terms.htm](http://www.xilinx.com/company/terms.htm).

**References**

1. [DS180]: 7 Series FPGAs Overview

**Technical Support**

Xilinx provides technical support at [www.xilinx.com/support](http://www.xilinx.com/support) for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

See the IP Release Notes Guide ([XTP025](http://www.xilinx.com)) for more information on this core. For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for the core being used. The following information is listed for each version of the core:

- New Features
- Resolved Issues
- Known Issues
Revision History

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/25/2012</td>
<td>1.0</td>
<td>Initial Xilinx release as a product guide. Replaces DS746, LogiCORE IP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SelectIO Interface Wizard Data Sheet and GSG700, LogiCORE IP SelectIO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interface Wizard Getting Started Guide.</td>
</tr>
<tr>
<td>12/18/2012</td>
<td>1.1</td>
<td>Updated GUI details in Chapter 4, Customizing and Generating the Core.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Vivado Design Suite to v2012.4. No other documentation changes.</td>
</tr>
</tbody>
</table>

Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available “AS IS” and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at [http://www.xilinx.com/warranty.htm](http://www.xilinx.com/warranty.htm); IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: [http://www.xilinx.com/warranty.htm#critapps](http://www.xilinx.com/warranty.htm#critapps).

© Copyright 2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, 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.