Virtex-6 FPGA SelectIO Resources

User Guide

UG361 (v1.6) November 7, 2014

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/24/2009	1.0	Initial Xilinx release.
11/02/2009	1.1	Added user guide reference for the Virtex-6 HXT FPGAs to the preface. Removed some DCI cascading limitations on page 19. Updated V_{CCO} maximum specification references to 2.625V in Table 1-7 and Table 1-20 through Table 1-23. Updated V_{REF} parameter name to match JEDEC specifications in Table 1-22, Table 1-25, and Table 1-26. Renamed the LOWPOWER attribute to IBUF_LOW_PWR. Revised Figure 3-1, page 130 and Figure 3-2, page 131 including removing DYNOCLKSEL and adding OCLKB. Removed DYN_OCLK_INV_EN attribute from Figure 3-2 and updated the Dynamic Clock Inversions discussion. Updated discussion in OSERDES Clocking Methods.
01/18/2010	1.2	Revised Figure 1-1 and Figure 1-3. Added to the DCI cascade guidelines on page 19. Updated Figure 1-76, Figure 1-77, Table 1-28, Figure 1-78, Figure 1-79, Figure 1-80, and Figure 1-81. Removed the previous Figure 1-81: Differential SSTL (1.5V) with DCI Bidirectional Termination since it is not a valid implementation. In Chapter 2, clarified that IODELAYE1 is a 31-tap delay element in Table 2-6, Figure 2-15, and on page 98, page 101, page 103, and page 108. Updated the default value for HIGH_PERFORMANCE_MODE Table 2-6, page 102. Removed OCLKB from Figure 3-2 and Table 3-1.

Date	Version	Revision
08/16/2010	1.3	Added the V_{REF} section. Updated guidelines when using DCI cascading on page 19. Updated the DCI in Virtex-6 Device I/O Standards rules. Removed SSTL_15 from heading in Figure 1-15. Updated REFCLK_FREQUENCY in Table 2-6. Added OVERSAMPLE value to INTERFACE_TYPE in Table 3-2. Updated MEMORY Interface Type discussion. Added OVERSAMPLE Interface Type discussion and Figure 3-10. Updated Figure 3-15 by removing BITSLIP_ENABLE = TRUE. Clarified descriptions in Table 3-5.
06/21/2013	1.4	Updated third bullet after Figure 1-5. Removed DIFF_SSTL15 standard from Table 1-1. Updated PULLUP/PULLDOWN/KEEPER for IBUF, OBUFT, and IOBUF, page 34. Updated Table 1-7. Replaced separate DIFF_HSTL and DIFF_SSTL buffer symbols with one OBUFDS buffer symbol in Figure 1-34, Figure 1-35, Figure 1-38, Figure 1-39, Figure 1-40, Figure 1-41, Figure 1-45, Figure 1-46, Figure 1-49, Figure 1-50, Figure 1-51, Figure 1-52, Figure 1-57, Figure 1-58, Figure 1-61, Figure 1-62, Figure 1-63, Figure 1-64, Figure 1-67, Figure 1-68, Figure 1-71, Figure 1-72, Figure 1-73, Figure 1-74, Figure 1-78, Figure 1-79, and Figure 1-80. Changed SSTL_15 to SSTL15 in SSTL (Stub-Series Terminated Logic). In Internal V_{REF}, changed 1.08V to 0.90V. Added sentence about referring to timing analyzer in IBUF_LOW_PWR Attribute. Added paragraph about terminations after Figure 1-26. Added V_{REF} input to right side of Figure 1-32. Added parenthetical sentence about series resistance to SSTL2_II, SSTL18_II, SSTL15 and SSTL2_II_DCI, SSTL18_II_DCI, SSTL15_DCI. In Table 1-33, changed output termination for SSTL15_DCI and DIFF_SSTL15_DCI standards from split to N/R and updated note 1. Added programmable input inversion to Introduction. Updated Combinatorial Input Path and IODELAYE1 Ports. Added VAR_LOADABLE to and updated fourth paragraph of IDELAY_TYPE Attribute. Updated last sentence of IDELAY_VALUE Attribute and ODELAY_VALUE Attribute. Updated third paragraph of ODELAY_TYPE Attribute. Removed description of FALSE setting from HIGH_PERFORMANCE_MODE Attribute. In Table 2-12, added BUFG to source column for ODELAY and Bidirectional Delay modes. Updated sentence before Figure 2-9. Updated Stability after an Increment/Decrement Operation. Updated first paragraph of IDELAYCTRL Overview. Updated second and fourth bullets of OLOGIC Resources. Updated Reset Input - RST. Added INIT_Q and SRVAL_Q attributes to Table 3-2. Added third bullet to NETWORKING Interface Type and MEMORY Interface Type. Updated CLK and OCLK labels in Figure 3-10. Updated Bitslip Operation, Figure 3-16, and Reset Input - RST. Added third bullet to DEFAULT Interface Type Method. Updated Latency column in Table 3-11.
03/21/2014	1.5	Updated disclaimer and copyright. Updated Step 6d and added Step 7 in DCI in Virtex-6 Device I/O Standards. Updated DIFF_SSTL2_II, DIFF_SSTL18_II, DIFF_SSTL15, DIFF_SSTL2_II_DCI, DIFF_SSTL18_II_DCI, DIFF_SSTL15_DCI, and SSTL2_II_T_DCI, SSTL18_II_T_DCI, SSTL15_T_DCI. Updated Control Value Out - CNTVALUEOUT. Updated OFB description in Table 3-1. Updated Figure 3-3. Added IOBDELAY attribute to Table 3-2. Added IOBDELAY Attribute and OFB_USED_Attribute. Updated High-Speed Clock for Strobe-Based Memory Interfaces - OCLK. Updated Output Feedback from OSERDESE1 - OFB. Removed Using D and DDLY in the ISERDESE1, Output Feedback and CLKPERF, and ISERDESE1 Feedback from OSERDESE1. Added additional information to IDELAYCTRL Usage and Design Guidelines and to Reset Input - RST.

Date	Version	Revision
11/07/2014	1.6	Clarified OSERDES support for both DDR3 and DDR2 in Table 3-5 including revised descriptions for OCBEXTEND, CLKPERF, CLKPERFDELAY, WC, and ODV. Updated ODELAY_USED in Table 3-6.

Table of Contents

Revision History 2

Preface: About This Guide

Guide Contents 11
Additional Documentation 11
Additional Resources 12

Chapter 1: SelectIO Resource

I/O Tile Overview 13
SelectIO Resources Introduction 14
SelectIO Resources General Guidelines 15
Virtex-6 FPGA I/O Bank Rules 15
Output Drive Source Voltage (V_{CCO}) Pins 15
V_{REF} 16
Pins 16
Internal V_{REF} 16
Virtex-6 FPGA Digitally Controlled Impedance (DCI) 16
Introduction 16
DCI Cascading 17
Xilinx DCI 19
Controlled Impedance Driver (Source Termination) 20
Controlled Impedance Driver with Half Impedance (Source Termination) 20
Input Termination to VCCO (Single Termination) 21
Input Termination to VCCO/2 (Split Termination) 21
Driver with Termination to V_{CCO}/2 (Split Termination) 23
DCI in Virtex-6 Device I/O Standards 24
DCI Usage Examples 26
Virtex-6 FPGA SelectIO Primitives 28
IBUF and IBUFG 28
OBUF 29
OBUFT 29
IOBUF 29
IBUFDS and IBUFGDS 30
OBUFDS 30
OBUFTDS 30
IOBUFDS 31
IBUFDS_DIFF_OUT 31
IOBUFDS_DIFF_OUT 31
Virtex-6 FPGA SelectIO Attributes/Constraints 32
Location Constraints 32
IOSTANDARD Attribute 32
IBUF_LOW_PWR Attribute 33
Output Slew Rate Attributes 33
Output Drive Strength Attributes 33
PULLUP/PULLDOWN/KEEPER for IBUF, OBUFT, and IOBUF 34

Differential Termination Attribute 34
Virtex-6 FPGA I/O Resource VHDL/Verilog Examples 34
Specific Guidelines for I/O Supported Standards 35
LVCMOS (Low Voltage Complementary Metal Oxide Semiconductor) 35
LVDCI (Low Voltage Digitally Controlled Impedance) 37
LVDCI_DV2 38
HSLVDCI (High-Speed Low Voltage Digitally Controlled Impedance) 39
HSTL (High-Speed Transceiver Logic) 40
HSTL_ I, HSTL_ III, HSTL_ I_18, HSTL_ III_18, HSTL_I_12 40
HSTL_ I_DCI, HSTL_ III_DCI, HSTL_ I_DCI_18, HSTL_ III_DCI_18 40
HSTL_ II and HSTL_ II_18 40
HSTL_ II_DCI and HSTL_ II_DCI_18 40
HSTL_ II_T_DCI and HSTL_ II_T_DCI_18 40
DIFF_HSTL_ II and DIFF_HSTL_II_18 40
DIFF_HSTL_II_DCI and DIFF_HSTL_II_DCI_18 41
DIFF_HSTL_II_T_DCI and DIFF_HSTL_II_T_DCI_18 41
DIFF_HSTL_I and DIFF_HSTL_I_18 41
DIFF_HSTL_I_DCI and DIFF_HSTL_I_DCI_18 41
HSTL Class I 41
Differential HSTL Class I 42
HSTL Class II 44
Differential HSTL Class II 46
HSTL Class III 49
HSTL_II_T_DCI (1.5V) Split-Thevenin Termination 50
HSTL Class I (1.8V) 50
Differential HSTL Class I (1.8V) 51
HSTL Class II (1.8V) 53
Differential HSTL Class II (1.8V) 55
HSTL Class III (1.8V) 57
HSTL_II_T_DCI_18 (1.8V) Split-Thevenin Termination 58
HSTL Class I (1.2V) 59
SSTL (Stub-Series Terminated Logic) 59
SSTL2_I, SSTL18_I 60
SSTL2_I_DCI, SSTL18_I_DCI 60
SSTL2_II, SSTL18_II, SSTL15 60
SSTL2_II_DCI, SSTL18_II_DCI, SSTL15_DCI 60
DIFF_SSTL2_I, DIFF_SSTL18_I 60
DIFF_SSTL2_I_DCI, DIFF_SSTL18_I_DCI 60
DIFF_SSTL2_II, DIFF_SSTL18_II, DIFF_SSTL15 60
DIFF_SSTL2_II_DCI, DIFF_SSTL18_II_DCI, DIFF_SSTL15_DCI 60
SSTL2_II_T_DCI, SSTL18_II_T_DCI, SSTL15_T_DCI 60
SSTL2 Class I (2.5V) 61
Differential SSTL2 Class I (2.5V) 62
SSTL2 Class II (2.5V) 64
Differential SSTL2 Class II (2.5V) 66
SSTL2_II_T_DCI (2.5V) Split-Thevenin Termination 69
SSTL18 Class I (1.8V) 69
Differential SSTL Class I (1.8V) 70
SSTL18 Class II (1.8V) 72
Differential SSTL Class II (1.8V) 75
SSTL18_II_T_DCI (1.8V) Split-Thevenin Termination 77
SSTL15 (1.5V) 78
Differential SSTL (1.5V) 80

SSTL15_T_DCI (1.5V) Split-Thevenin Termination 82
Differential Termination: DIFF_TERM Attribute 82
LVDS and Extended LVDS (Low Voltage Differential Signaling) 82
Transmitter Termination 83
Receiver Termination 83
HyperTransport™ Protocol (HT) 84
Reduced Swing Differential Signaling (RSDS) 84
BLVDS (Bus LVDS) 85
Differential LVPECL (Low-Voltage Positive Emitter-Coupled Logic) 85
LVPECL Transceiver Termination 85
Rules for Combining I/O Standards in the Same Bank 86

Chapter 2: SelectIO Logic Resources

Introduction 91
ILOGIC Resources 91
Combinatorial Input Path 92
Input DDR Overview (IDDR) 93
OPPOSITE_EDGE Mode 93
SAME_EDGE Mode 93
SAME_EDGE_PIPELINED Mode 94
Input DDR Primitive (IDDR) 94
IDDR VHDL and Verilog Templates 95
ILOGIC Timing Models 96
ILOGIC Timing Characteristics 96
ILOGIC Timing Characteristics, DDR 96
Input/Output Delay Element (IODELAYE1) 98
IODELAYE1 Primitive 99
IODELAYE1 Ports 100
IODELAYE1 Attributes 102
IODELAYE1 Modes 104
IODELAYE1 Timing 108
Stability after an Increment/Decrement Operation 109
IODELAYE1 VHDL and Verilog Instantiation Template 110
IODELAYE1 Turnaround Time Usage Model 110
IDELAYCTRL Overview 114
IDELAYCTRL Primitive 115
IDELAYCTRL Ports 115
IDELAYCTRL Timing 116
IDELAYCTRL Locations 116
IDELAYCTRL Usage and Design Guidelines 117
OLOGIC Resources 120
Combinatorial Output Data and 3-State Control Path 121
Output DDR Overview (ODDR) 121
OPPOSITE_EDGE Mode 122
SAME_EDGE Mode 122
Clock Forwarding 122
Output DDR Primitive (ODDR) 123
ODDR VHDL and Verilog Templates 124
OLOGIC Timing Models 124
Timing Characteristics 124

Chapter 3: Advanced SelectIO Logic Resources

Introduction . . . 129

Input Serial-to-Parallel Logic Resources (ISERDES) . . . 129

ISERDES Primitive (ISERDESE1) . . . 131

ISERDESE1 Ports . . . 132

Registered Outputs – Q1 to Q6 . . . 132

Combinatorial Output – O . . . 133

Bitslip Operation - BITSLIP . . . 133

Clock Enable Inputs - CE1 and CE2 . . . 133

High-Speed Clock Input - CLK . . . 134

High-Speed Clock Input - CLKB . . . 134

Divided Clock Input - CLKDIV . . . 134

Serial Input Data from IOB - D . . . 134

Serial Input Data from IODELAYE1 - DDLY . . . 134

Serial Input Data from OSERDESE1 - OFB . . . 134

High-Speed Clock for Strobe-Based Memory Interfaces - OCLK . . . 134

Reset Input - RST . . . 135

ISERDESE1 Attributes . . . 136

DATA_RATE Attribute . . . 137

DATA_WIDTH Attribute . . . 137

INTERFACE_TYPE Attribute . . . 137

IOBDELAY Attribute . . . 139

NUM_CE Attribute . . . 140

OFB_USED_Attribute . . . 140

SERDES_MODE Attribute . . . 140

ISERDESE1 Clocking Methods . . . 141

NETWORKING Interface Type . . . 141

MEMORY Interface Type . . . 141

MEMORY_QDR Interface Type . . . 142

OVERSAMPLE Interface Type . . . 142

MEMORY_DDR3 Interface Type . . . 144

ISERDES Width Expansion . . . 144

Guidelines for Expanding the Serial-to-Parallel Converter Bit Width . . . 144

ISERDES Latencies . . . 145

Dynamic Clock Inversions . . . 145

ISERDES Timing Model and Parameters . . . 145

Timing Characteristics . . . 146

Reset Input Timing . . . 146

ISERDESE1 VHDL and Verilog Instantiation Template . . . 147

BITSLIP Submodule . . . 148

Bitslip Operation . . . 148

Bitslip Timing Model and Parameters . . . 150

Output Parallel-to-Serial Logic Resources (OSERDES) . . . 152

Data Parallel-to-Serial Converter . . . 152

3-State Parallel-to-Serial Conversion . . . 153

OSERDES Primitive . . . 153

OSERDES Ports . . . 154

Data Path Output - OQ . . . 155

Output Feedback from OSERDESE1 - OFB . . . 155

3-state Control Output - TQ . . . 155

3-state Control Output - TFB . . . 155

High-Speed Clock Input - CLK . . . 155

Divided Clock Input - CLKDIV . . . 155

Performance Clock from MMCM - CLKPERF 155
Performance Clock from MMCM Delayed with IODELAYE1 - CLKPERFDELAY . . . 156
Parallel Data Inputs - D1 to D6 156
Reset Input - RST 156
Output Data Clock Enable - OCE 157
Write Command - WC 157
ODELAY Value - ODV 157
Output Circular Buffer Extend - OCBEXTEND 158
3-state Signal Clock Enable - TCE 158
Parallel 3-state Inputs - T1 to T4 158
OSERDES Attributes 158
DATA_RATE_OQ Attribute 159
DATA_RATE_TQ Attribute 159
DATA_WIDTH Attribute 159
SERDES_MODE Attribute 159
TRISTATE_WIDTH Attribute 159
ODELAY_USED Attribute 159
INTERFACE_TYPE Attribute 160
OSERDES Clocking Methods 160
DEFAULT Interface Type Method 160
MEMORY_DDR3 Interface Type Method 161
OSERDES Width Expansion 161
Guidelines for Expanding the Parallel-to-Serial Converter Bit Width 162
OSERDES Latencies 163
DEFAULT Interface Type Latencies 163
OSERDES Timing Model and Parameters 163
Timing Characteristics of 2:1 SDR Serialization 164
Timing Characteristics of 8:1 DDR Serialization 165
Timing Characteristics of 4:1 DDR 3-State Controller Serialization 166
Reset Output Timing 167
OSERDES VHDL and Verilog Instantiation Templates 169

About This Guide

This document describes Virtex®-6 FPGA SelectIO™ technology.

Guide Contents

This manual contains the following chapters:

- Chapter 1, SelectIO Resource
- Chapter 2, SelectIO Logic Resources
- Chapter 3, Advanced SelectIO Logic Resources

Additional Documentation

The following documents are also available for download at:

http://www.xilinx.com/support/documentation/virtex-6.htm

- Virtex-6 Family Overview

  The features and product selection of the Virtex-6 family are outlined in this overview.

- Virtex-6 FPGA Data Sheet: DC and Switching Characteristics

  This data sheet contains the DC and Switching Characteristic specifications for the Virtex-6 family.

- Virtex-6 FPGA Packaging and Pinout Specifications

  This specification includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.

- Virtex-6 FPGA Configuration Guide

  This all-encompassing configuration guide includes chapters on configuration interfaces (serial and SelectMAP), bitstream encryption, boundary-scan and JTAG configuration, reconfiguration techniques, and readback through the SelectMAP and JTAG interfaces.

- Virtex-6 FPGA Clocking Resources User Guide

  This guide describes the clocking resources available in all Virtex-6 devices, including the MMCM and PLLs.

- Virtex-6 FPGA Configurable Logic Blocks User Guide

  This guide describes the capabilities of the configurable logic blocks (CLBs) available in all Virtex-6 devices.

- Virtex-6 FPGA Memory Resources User Guide

  The functionality of the block RAM and FIFO are described in this user guide.

- Virtex-6 FPGA GTH Transceivers User Guide

  This guide describes the GTH transceivers available in all Virtex-6 HXT FPGAs except the XC6VHX250T and the XC6VHX380T in the FF1154 package.

- Virtex-6 FPGA GTX Transceivers User Guide

  This guide describes the GTX transceivers available in all Virtex-6 FPGAs except the XC6VLX760.

- Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC User Guide

  This guide describes the dedicated Tri-Mode Ethernet Media Access Controller available in all Virtex-6 FPGAs except the XC6VLX760.

- Virtex-6 FPGA DSP48E1 Slice User Guide

  This guide describes the architecture of the DSP48E1 slice in Virtex-6 FPGAs and provides configuration examples.

- Virtex-6 FPGA System Monitor User Guide

  The System Monitor functionality available in all Virtex-6 devices is outlined in this guide.

- Virtex-6 FPGA PCB Design Guide

  This guide provides information on PCB design for Virtex-6 devices, with a focus on strategies for making design decisions at the PCB and interface level.

Additional Resources

To search the database of silicon and software questions and answers or to create a technical support case in WebCase, see the Xilinx website at:

http://www.xilinx.com/support

Chapter 1

SelectIO Resource

I/O Tile Overview

Input/output characteristics and logic resources are covered in three consecutive chapters.

Chapter 1, SelectIO Resource describes the electrical behavior of the output drivers and input receivers, and gives detailed examples of many standard interfaces. Chapter 2, SelectIO Logic Resources describes the input and output data registers and their Double-Data-Rate (DDR) operation, and the programmable input delay (IDELAY). Chapter 3, Advanced SelectIO Logic Resources describes the data serializer/deserializer (SERDES).

An I/O tile contains two IOBs, two ILOGICs, two OLOGICs, and two IODELAYs. Figure 1-1 shows a Virtex-6 FPGA I/O tile.

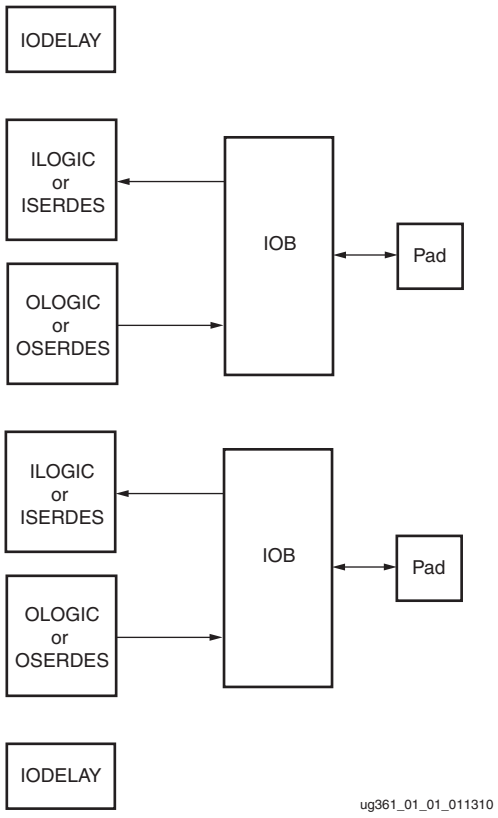


Figure 1-1: **Virtex-6 FPGA I/O Tile**

SelectIO Resources Introduction

All Virtex-6 FPGAs have configurable high-performance SelectIO™ drivers and receivers, supporting a wide variety of standard interfaces. The robust feature set includes programmable control of output strength and slew rate, on-chip termination using Digitally Controlled Impedance (DCI), and the ability to internally generate a reference voltage (INTERNAL_VREF).

Each IOB contains both input, output, and 3-state SelectIO drivers. These drivers can be configured to various I/O standards. Differential I/O uses the two IOBs grouped together in one tile.

- Single-ended I/O standards (LVCMOS, HSTL, and SSTL)
- Differential I/O standards (LVDS, HT, LVPECL, BLVDS, Differential HSTL and SSTL)
- Differential and V_{REF} dependent inputs are powered by V_{CCAUX}

Each Virtex-6 FPGA I/O tile contains two IOBs, and also two ILOGIC blocks and two OLOGIC blocks, as described in Chapter 2, SelectIO Logic Resources

Figure 1-2 shows the basic IOB and its connections to the internal logic and the device Pad.

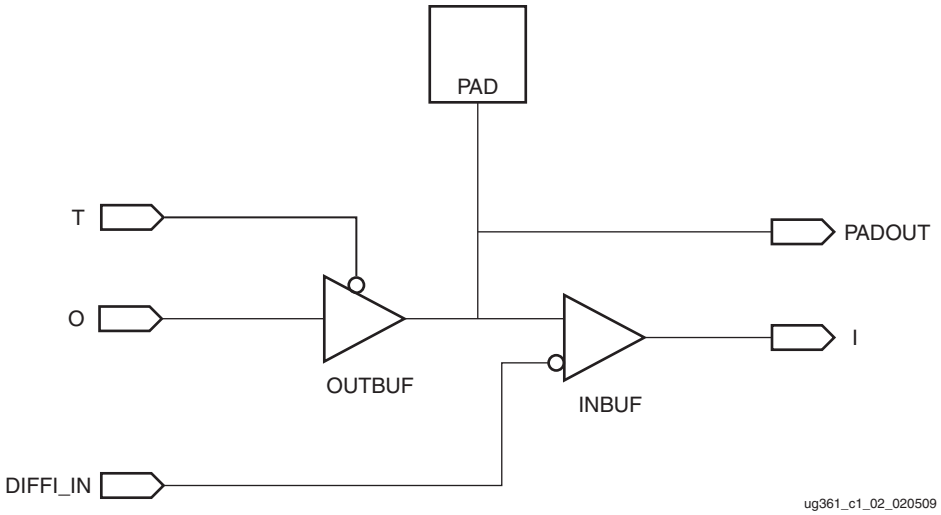


Figure 1-2: **Basic IOB Diagram**

Each IOB has a direct connection to an ILOGIC/OLOGIC pair containing the input and output logic resources for data and 3-state control for the IOB. Both ILOGIC and OLOGIC can be configured as ISERDES and OSERDES, respectively, as described in Chapter 3, Advanced SelectIO Logic Resources

SelectIO Resources General Guidelines

This section summarizes the general guidelines to be considered when designing with the SelectIO resources in Virtex-6 FPGAs.

Virtex-6 FPGA I/O Bank Rules

In Virtex-6 devices, with some exceptions in the center column, an I/O bank consists of 40 IOBs. The number of banks depends upon the device size. In the *Virtex-6 Family Overview* the total number of I/O banks is listed by device type. The XC6VLX130T has 15 usable I/O banks. Figure 1-3 is an example of a columnar floorplan showing the XC6VLX130T I/O banks.

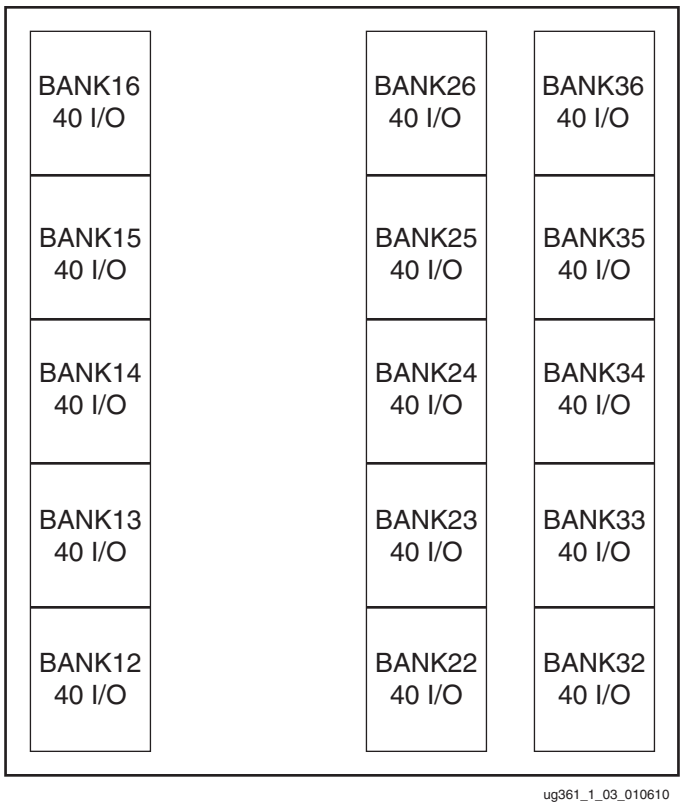


Figure 1-3: **Virtex-6 FPGA XC6VLX130T I/O Banks**

Output Drive Source Voltage (V_{CCO}) Pins

Many of the low-voltage I/O standards supported by Virtex-6 devices require a different output drive voltage (V_{CCO}). As a result, each device often supports multiple output drive source voltages.

Output buffers within a given V_{CCO} bank must share the same output drive source voltage. The following input buffers use the V_{CCO} voltage: LVCMOS, LVDCI and other DCI standards. Additionally, the V_{CCAUX} supply must turn on before the V_{CCO} supply, and always remain at a voltage greater than, or equal to, the V_{CCO} supply.

V_{REF}

Pins

Single-ended I/O standards with a differential amplifier input buffer require an input reference voltage (V_{REF}). V_{REF} can be an external input into Virtex-6 devices. When using a single-ended I/O standard that requires a differential amplifier input buffer, within each I/O bank, one of every 20 I/O pins is automatically configured as a V_{REF} input.

Internal V_{REF}

The V_{REF} can also be generated internally by the FPGA removing the need for an externally generated V_{REF} supply and freeing the V_{REF} pin for standard usage. This internally generated V_{REF} is sourced from the V_{CCAUX}. Each bank has a single V_{REF} plane and each bank can only be set to a single V_{REF} value. When using internal V_{REF}, the V_{REF} pins are available for use.

The constraint INTERNAL_VREF is assigned to one bank at time.

Example 1: Bank 24 using HSTL_II (1.5V) requires the use of the following constraint:

```
INTERNAL_VREF_BANK24 = 0.75;
```

Example 2: Bank 25 using HSTL_II_18 (1.8V) requires the use of the following constraint. According to the JEDEC standard, the nominal or typical V_{REF} value is 0.90V.

```
INTERNAL_VREF_BANK25 = 0.90;
```

The rules for using INTERNAL_VREF are:

- One value of V_{REF} can be set for the bank.
- INTERNAL_VREF can only be set to the nominal value of a given IOSTANDARD.
- Valid setting of INTERNAL_VREF are:
  - 0.60
  - 0.75
  - 0.90
  - 1.1
  - 1.25

V_{REF} pins can be used as normal I/O. The rules for combining I/O standards in the same bank also apply for INTERNAL_VREF. Table 1-33, page 87 includes a listing of V_{REF} values for various I/O standards.

Virtex-6 FPGA Digitally Controlled Impedance (DCI)

Introduction

As FPGAs get bigger and system clock speeds get faster, PC board design and manufacturing becomes more difficult. With ever faster edge rates, maintaining signal integrity becomes a critical issue. PC board traces must be properly terminated to avoid reflections or ringing.

To terminate a trace, resistors are traditionally added to make the output and/or input match the impedance of the receiver or driver to the impedance of the trace. However, due to increased device I/Os, adding resistors close to the device pins increases the board area and component count, and can in some cases be physically impossible. To address these

issues and to achieve better signal integrity, Xilinx developed the Digitally Controlled Impedance (DCI) technology.

DCI adjusts the output impedance or input termination to accurately match the characteristic impedance of the transmission line. DCI actively adjusts the impedance of the I/O to equal an external reference resistance. This compensates for changes in I/O impedance due to process variation. It also continuously adjusts the impedance of the I/O to compensate for variations of temperature and supply voltage fluctuations.

In the case of controlled impedance drivers, DCI controls the driver impedance to match two reference resistors, or optionally, to match half the value of these reference resistors. DCI eliminates the need for external series termination resistors.

DCI provides the parallel or series termination for transmitters or receivers. This eliminates the need for termination resistors on the board, reduces board routing difficulties and component count, and improves signal integrity by eliminating stub reflection. Stub reflection occurs when termination resistors are located too far from the end of the transmission line. With DCI, the termination resistors are as close as possible to the output driver or the input buffer, thus, eliminating stub reflections.

DCI Cascading

Previously, using DCI I/O standards in a bank required connecting external reference resistors to the VRN and VRP pins in that same bank. The VRN/VRP pins provide a reference voltage used by internal DCI circuitry to adjust the I/O output impedance to match the external reference resistors. As shown in Figure 1-4, a digital control bus is internally distributed throughout the bank to control the impedance of each I/O.

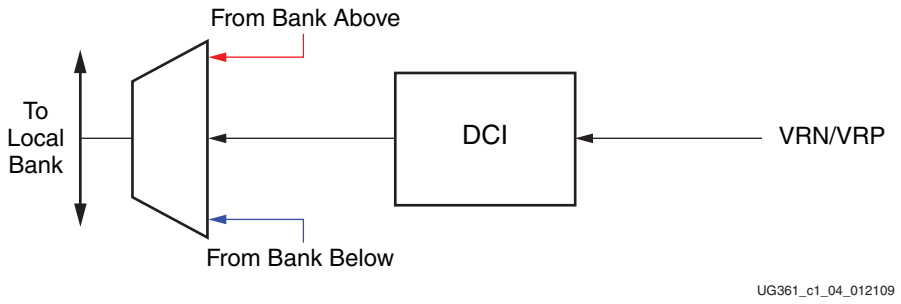


Figure 1-4: **DCI Use within a Bank**

The Virtex-6 FPGA banks using DCI I/O standards now have the option of deriving the DCI impedance values from another DCI bank. With DCI cascading, one bank (the master bank) must have its VRN/VRP pins connected to external reference resistors. Other banks in the same column (slave banks) can use DCI standards with the same impedance as the master bank, without connecting the VRN/VRP pins on these banks to external resistors. DCI impedance control in cascaded banks is received from the master bank.

When using DCI cascading, the DCI control circuitry in the master bank creates and routes DCI control to the cascaded banks in daisy-chain style. Only the master bank's VRN/VRP pins are required when using DCI cascading.

Also, when using DCI cascading, only one set of VRN/VRP pins provides the DCI reference voltage for multiple banks. DCI cascading:

- Reduces overall power, since fewer voltage references are required
- Frees up VRN/VRP pins on slave banks for general customer use

Figure 1-5 shows DCI cascading support over multiple banks. Bank B is the master bank.

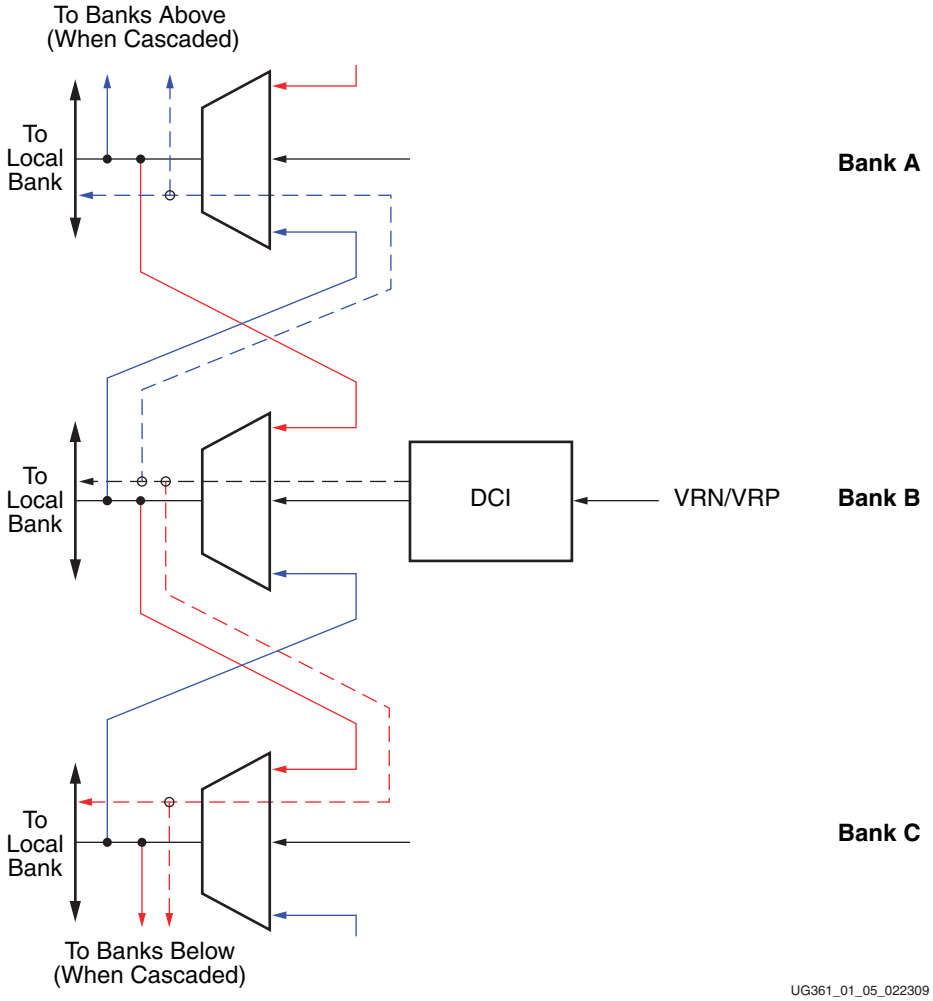


Figure 1-5: **DCI Cascading Supported Over Multiple Banks**

The guidelines when using DCI cascading are as follows:

- The master and slave banks must all reside on the same column (left, center, or right) on the device.
- Master and slave banks must have the same V_{CCO} and V_{REF} (if applicable) voltage.
- DCI cascading is allowed through banks not using DCI (pass-through banks).
- DCI I/O banking compatibility rules must be satisfied across all master and slave banks (for example, only one DCI I/O standard using single termination type is allowed across all master and slave banks). DCI I/O standard compatibility is not constrained to one bank when DCI cascading is implemented; it extends across all master and slave banks.
- DCI cascading can span the entire column as long as the above guidelines are met.
- Locate adjacent banks. Bank location information is best determined from partgen generated package files (`partgen -v XC6VLX130TFF1156`). The resulting package file with a .pkg extension contains XY I/O location information. The X designator indicates I/Os in the same column. The Y designator indicates the position of an I/O within a specific bank. The bank number is also shown. Consecutive Y locations across bank boundaries show adjacent banks.
- DCI cascade is enabled by using the DCI_CASCADE constraint described in the constraints guide.

Xilinx DCI

DCI uses two multi-purpose reference pins in each bank to control the impedance of the driver or the parallel termination value for all of the I/Os of that bank. The N reference pin (VRN) must be pulled up to V_{CCO} by a reference resistor, and the P reference pin (VRP) must be pulled down to ground by another reference resistor. The value of each reference resistor should be equal to the characteristic impedance of the PC board traces, or should be twice that value. See Driver with Termination to V_{CCO}/2 (Split Termination), page 23.

When a DCI I/O standard is used on a particular bank, the two multi-purpose reference pins cannot be used as regular I/Os. However, if DCI I/O standards are not used in the bank, these pins are available as regular I/O pins. The *Virtex-6 FPGA Packaging and Pinout Specifications* gives detailed pin descriptions.

DCI adjusts the impedance of the I/O by selectively turning transistors in the I/Os on or off. The impedance is adjusted to match the external reference resistors. The impedance adjustment process has two phases. The first phase compensates for process variations by controlling the larger transistors in the I/Os. It occurs during the device startup sequence. The second phase maintains the impedance in response to temperature and supply voltage changes by controlling the smaller transistors in the I/Os. It begins immediately after the first phase and continues indefinitely, even while the device is operating. By default, the DONE pin does not go High until the first phase of the impedance adjustment process is complete.

The coarse impedance calibration during the first phase of impedance adjustment can be invoked after configuration by instantiating the DCIRESET primitive. By toggling the RST input to the DCIRESET primitive while the device is operating, the DCI state machine is reset and both phases of impedance adjustment proceed in succession. All I/Os using DCI will be unavailable until the LOCKED output from the DCIRESET block is asserted.

This functionality is useful in applications where the temperature and/or supply voltage changes significantly from device power-up to the nominal operating condition. Once at

the nominal operating temperature and voltage, performing the first phase of impedance adjustment allows optimal headroom for the second phase of impedance adjustment.

For controlled impedance output drivers, the impedance can be adjusted either to match the reference resistors or half the resistance of the reference resistors. For on-chip termination, the termination is always adjusted to match the reference resistors.

DCI can configure output drivers to be the following types:

1. Controlled Impedance Driver (Source Termination)
2. Controlled Impedance Driver with Half Impedance (Source Termination)

It can also configure inputs to have the following types of on-chip terminations:

1. Input termination to V_{CCO} (Single Termination)
2. Input termination to $V_{CCO}/2$ (Split Termination, Thevenin equivalent)

For bidirectional operation, both ends of the line can be DCI-terminated regardless of direction:

1. Driver with termination to V_{CCO} (Single Termination)
2. Driver with termination to $V_{CCO}/2$ (Split Termination, Thevenin equivalent)

Alternatively, bidirectional point-to-point lines can use controlled-impedance drivers (with 3-state buffers) on both ends.

Controlled Impedance Driver (Source Termination)

Some I/O standards, such as LVCMOS, must have a drive impedance matching the characteristic impedance of the driven line. DCI can provide controlled impedance output drivers to eliminate reflections without an external source termination. The impedance is set by the external reference resistors with resistance equal to the trace impedance.

The DCI I/O standards supporting the controlled impedance driver are: LVDCI_15, LVDCI_18, LVDCI_25, HSLVDCI_15, HSLVDCI_18, and HSLVDCI_25. Figure 1-6 illustrates a controlled impedance driver in a Virtex-6 device.

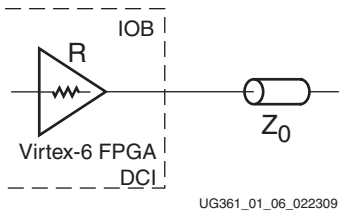


Figure 1-6: **Controlled Impedance Driver**

Controlled Impedance Driver with Half Impedance (Source Termination)

DCI also provides drivers with one half of the impedance of the reference resistors. This doubling of the reference resistor value reduces the static power consumption through these resistors by a factor of half. The DCI I/O standards supporting controlled impedance drivers with half-impedance are LVDCI_DV2_15, LVDCI_DV2_18, and LVDCI_DV2_25.

Figure 1-7 illustrates a controlled driver with half impedance inside a Virtex-6 device. The reference resistors R must be $2 \times Z_0$ in order to match the impedance of Z_0.

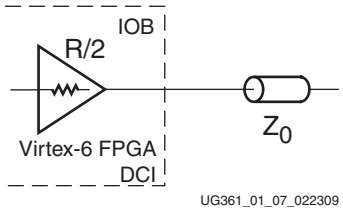


Figure 1-7: **Controlled Impedance Driver with Half Impedance**

Input Termination to V_{CCO} (Single Termination)

Some I/O standards require an input termination to V_{CCO} (see Figure 1-8).

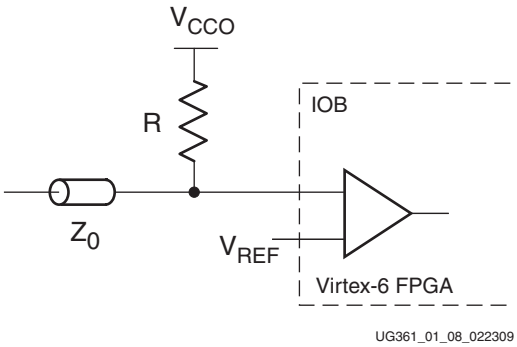


Figure 1-8: **Input Termination to V_{CCO} without DCI**

DCI can also provide input termination to V_{CCO} using single termination. The termination resistance is set by the reference resistors. The HSTL standard is controlled by 50 Ω reference resistors. The DCI I/O standards supporting single termination are: HSTL_III_DCI and HSTL_III_DCI_18.

Figure 1-9 illustrates DCI single termination inside a Virtex-6 device.

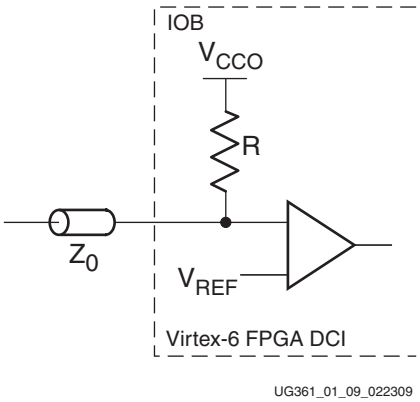


Figure 1-9: **Input Termination Using DCI Single Termination**

Input Termination to $V_{CCO}/2$ (Split Termination)

Some I/O standards (e.g., HSTL Class I and II) require an input termination voltage of $V_{CCO}/2$ (see Figure 1-10).

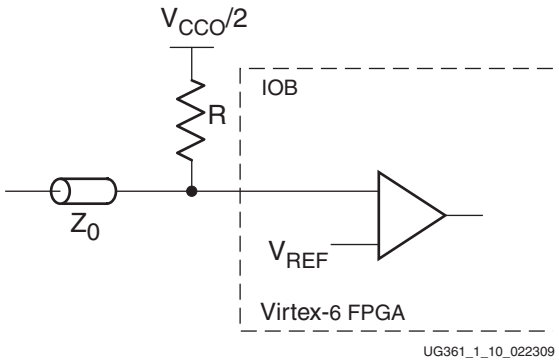


Figure 1-10: **Input Termination to $V_{CCO}/2$ without DCI**

This is equivalent to having a split termination composed of two resistors. One terminates to V_{CCO}, the other to ground. The resistor values are 2R. DCI provides termination to $V_{CCO}/2$ using split termination. The termination resistance is set by the external reference resistors, i.e., the resistors to V_{CCO} and ground are each twice the reference resistor value. Both HSTL and SSTL standards need 50 Ω external reference resistors. The DCI input standards supporting split termination are shown in Table 1-1.

Table 1-1: **DCI Input Standards Supporting Split Termination**

HSTL_I_DCI	DIFF_HSTL_I_DCI	SSTL2_I_DCI	DIFF_SSTL2_I_DCI
HSTL_I_DCI_18	DIFF_HSTL_I_DCI_18	SSTL2_II_DCI	DIFF_SSTL2_II_DCI
HSTL_II_DCI	DIFF_HSTL_II_DCI	SSTL18_I_DCI	DIFF_SSTL18_I_DCI
HSTL_II_DCI_18	DIFF_HSTL_II_DCI_18	SSTL18_II_DCI	DIFF_SSTL18_II_DCI
HSTL_II_T_DCI		SSTL15_DCI	DIFF_SSTL15_DCI
HSTL_II_T_DCI_18		SSTL2_II_T_DCI	DIFF_SSTL15_T_DCI
		SSTL18_II_T_DCI	
		SSTL15_T_DCI	

Figure 1-11 illustrates split termination inside a Virtex-6 device.

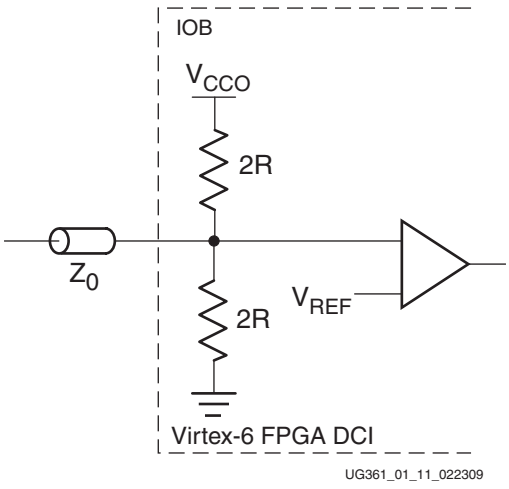


Figure 1-11: **Input Termination to $V_{CCO}/2$ Using DCI Split Termination**

Driver with Termination to $V_{CCO}/2$ (Split Termination)

Some I/O standards, such as HSTL Class II, require an output termination to $V_{CCO}/2$ (see Figure 1-12).

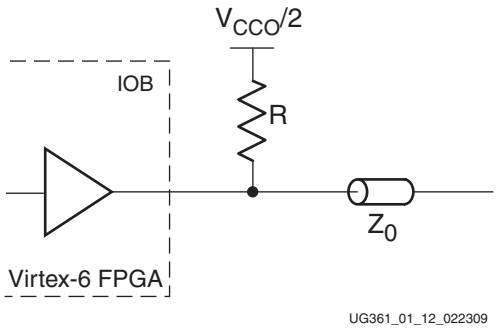


Figure 1-12: **Driver with Termination to $V_{CCO}/2$ without DCI**

DCI can provide output termination to $V_{CCO}/2$ using split termination. DCI only controls the impedance of the termination, but not the driver. Both HSTL and SSTL standards need 50Ω external reference resistors. The DCI output standards supporting drivers with split termination are shown in Table 1-2.

Table 1-2: **DCI Output Standards Supporting Split Termination**

HSTL_II_DCI	DIFF_HSTL_II_DCI	SSTL2_II_DCI	DIFF_SSTL2_II_DCI
HSTL_II_DCI_18	DIFF_HSTL_II_DCI_18	SSTL18_II_DCI	DIFF_SSTL18_II_DCI

Figure 1-13 illustrates a driver with split termination inside a Virtex-6 device.

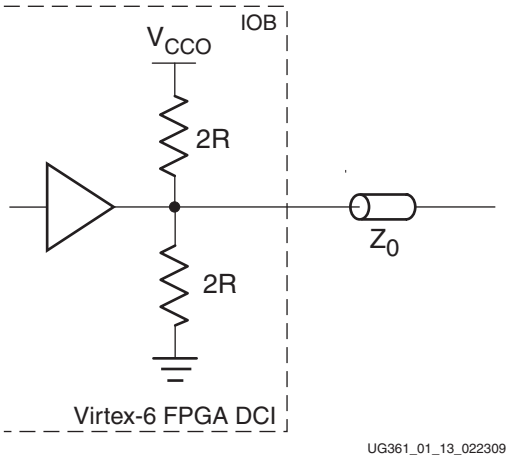


Figure 1-13: **Driver with Termination to V_{CCO}/2 Using DCI Split Termination**

DCI in Virtex-6 Device I/O Standards

DCI works with single-ended I/O standards. DCI supports the standards shown in Table 1-3.

Table 1-3: **Virtex-6 Device DCI I/O Standards**

LVDCI	HSTL_I_DCI	DIFF_HSTL_I_DCI	HSTL_III_DCI	SSTL2_I_DCI	DIFF_SSTL2_I_DCI
HSLVDCI	HSTL_I_DCI_18	DIFF_HSTL_I_DCI_18	HSTL_III_DCI_18	SSTL2_II_DCI	DIFF_SSTL2_II_DCI
LVDCI_DV2	HSTL_II_DCI	DIFF_HSTL_II_DCI		SSTL18_I_DCI	DIFF_SSTL18_I_DCI
	HSTL_II_DCI_18	DIFF_HSTL_II_DCI_18		SSTL18_II_DCI	DIFF_SSTL18_II_DCI
	HSTL_II_T_DCI			SSTL15_DCI	DIFF_SSTL15_DCI
	HSTL_II_T_DCI_18			SSTL2_II_T_DCI	DIFF_SSTL15_T_DCI
				SSTL18_II_T_DCI	
				SSTL15_T_DCI	

To correctly use DCI in a Virtex-6 device, users must follow the following rules:

1. V_{CCO} pins must be connected to the appropriate V_{CCO} voltage based on the IOSTANDARDs in that bank.
2. Correct DCI I/O buffers must be used in the software either by using IOSTANDARD attributes or instantiations in the HDL code.

3. Some DCI standards require connecting the external reference resistors to the multipurpose pins (VRN and VRP) in the bank. Where this is required, these two multipurpose pins cannot be used as general-purpose I/O. Refer to the Virtex-6 FPGA pinout tables for the specific pin locations. Pin VRN must be pulled up to V_{CCO} by its reference resistor. Pin VRP must be pulled down to ground by its reference resistor.

   Some DCI standards do not require connecting the external reference resistors to the VRP/VRN pins. When these DCI-based I/O standards are the only ones in a bank, the the VRP and VRN pins in that bank *can be used* as general-purpose I/O.

   - DCI outputs that do not require reference resistors on VRP/VRN:

     HSTL_I_DCI
     HSTL_III_DCI
     HSTL_I_DCI_18
     HSTL_III_DCI_18
     SSTL2_I_DCI
     SSTL18_I_DCI
     SSTL15_DCI

   - DCI inputs that do not require reference resistors on VRP/VRN:

     LVDCI_15
     LVDCI_18
     LVDCI_25
     LVDCI_DV2_15
     LVDCI_DV2_18
     LVDCI_DV2_25

4. The value of the external reference resistors should be selected to give the desired output impedance. If using HSTL_DCI or SSTL_DCI I/O standards, then the external reference resistors should be 50 Ω.
5. The values of the reference resistors must be within the supported range (20 Ω – 100 Ω).
6. Follow the DCI I/O banking rules:
   a. V_{REF} must be compatible for all of the inputs in the same bank.
   b. V_{CCO} must be compatible for all of the inputs and outputs in the same bank.
   c. No more than one DCI I/O standard using single termination type is allowed per bank.
   d. No more than one DCI I/O standard using split termination type is allowed per bank. One exception is DIFF_HSTL_T_DCI and DIFF_HSTL_II_T_DCI, which are equivalent.
   e. Single termination and split termination, controlled impedance driver, and controlled impedance driver with half impedance can co-exist in the same bank.
7. If the dual-purpose configuration pins D0-D31 are used as I/O after configuration, and one or more of them are set to DCI input or output, then the configuration option DCIUpdateMode must be set to Continuous to ensure proper operation after configuration.

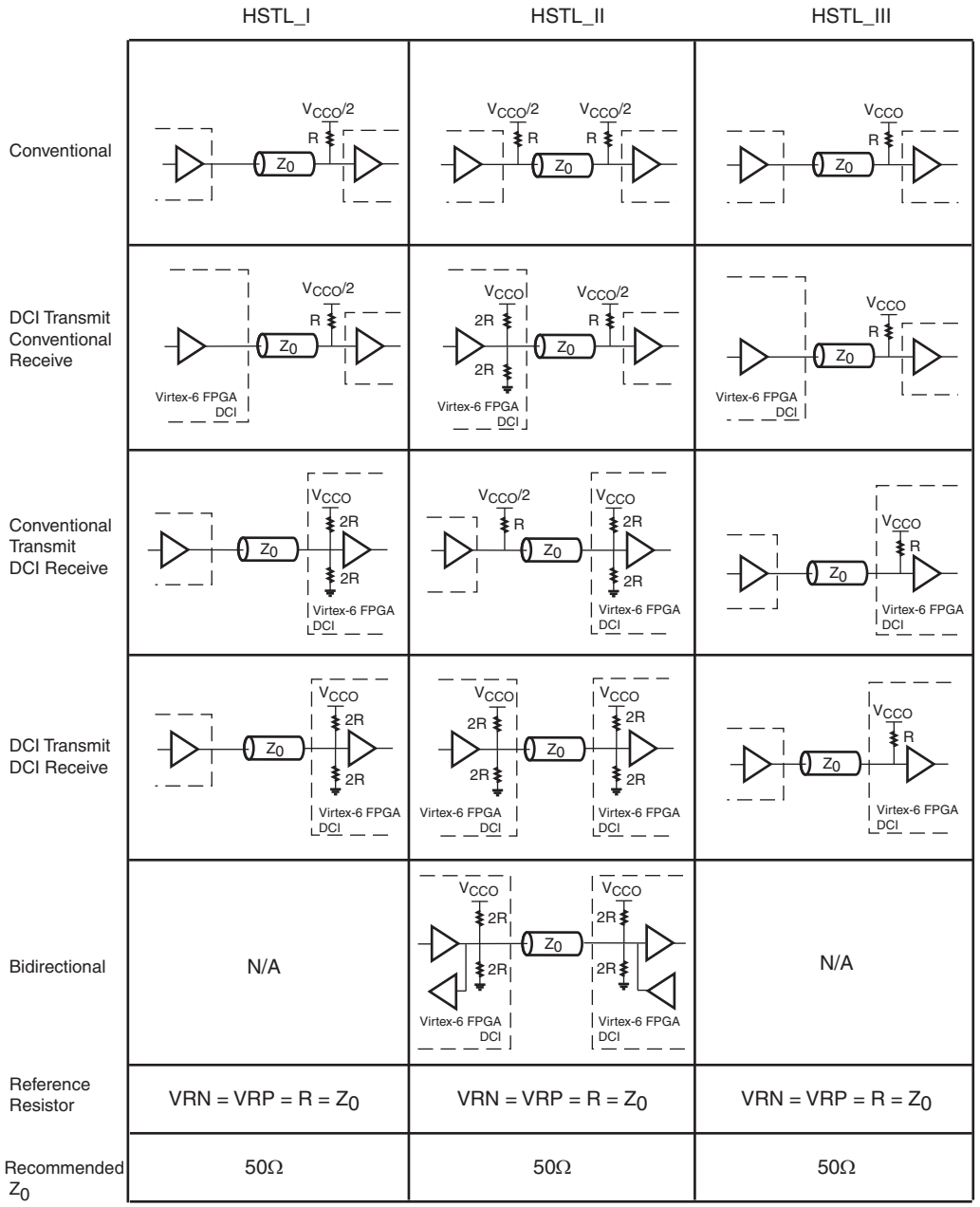
The behavior of a DCI 3-state outputs is as follows:

If a LVDCI or LVDCI_DV2 driver is in 3-state, the driver is 3-stated. If a driver with single or split termination is in 3-state, the driver is 3-stated but the termination resistor remains.

The following section lists actions that must be taken for each DCI I/O standard.

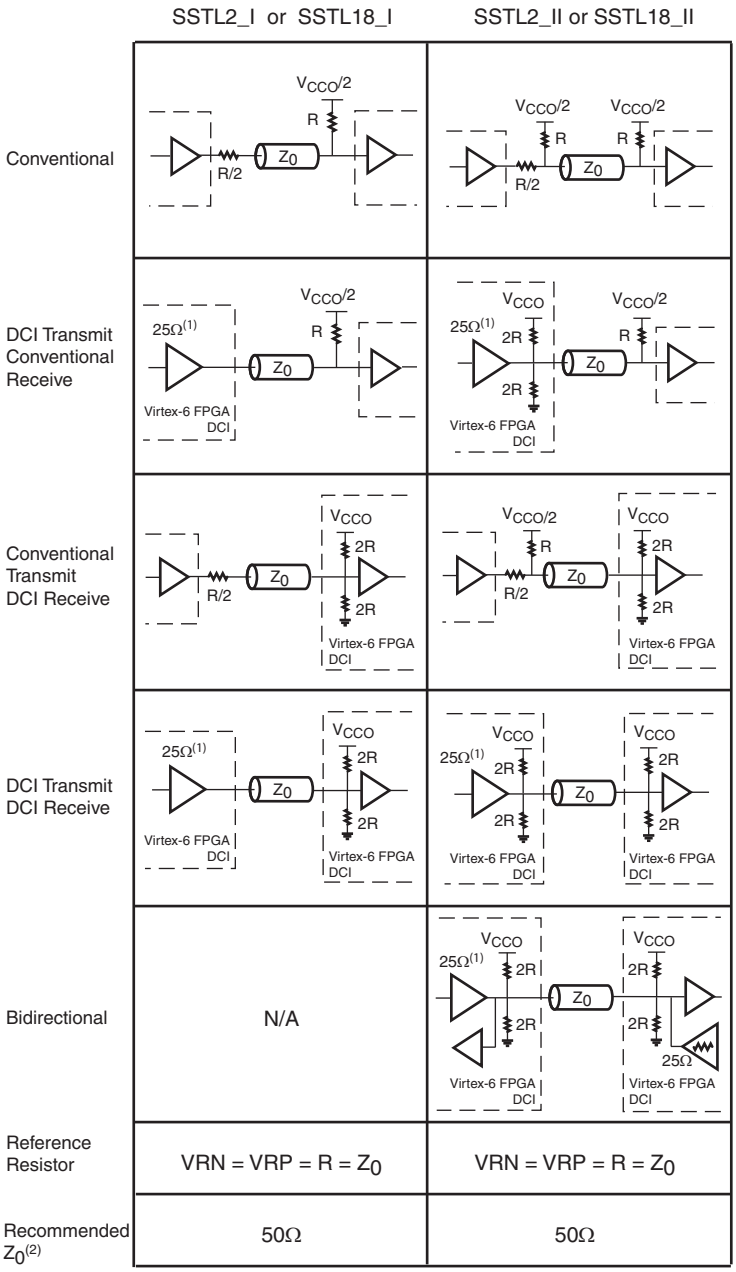
DCI Usage Examples

- Figure 1-14 provides examples illustrating the use of the HSTL_I_DCI, HSTL_II_DCI, and HSTL_III_DCI I/O standards.
- Figure 1-15 provides examples illustrating the use of the SSTL2_I_DCI and SSTL2_II_DCI I/O standards.



Notes:
1. Z_0 is the recommended PCB trace impedance.

Figure 1-14: **HSTL DCI Usage Examples**



Notes:
1. The SSTL-compatible 25Ω or 20Ω series resistor is accounted for in the DCI buffer, and it is not DCI controlled.
2. Z_0 is the recommended PCB trace impedance.

ug361_c1_15_062810

Figure 1-15: **SSTL DCI Usage Examples**

Virtex-6 FPGA SelectIO Primitives

The Xilinx software library includes an extensive list of primitives to support a variety of I/O standards available in the Virtex-6 FPGA I/O primitives. The following are five generic primitive names representing most of the available single-ended I/O standards.

- IBUF (input buffer)
- IBUFG (clock input buffer)
- OBUF (output buffer)
- OBUFT (3-state output buffer)
- IOBUF (input/output buffer)

These seven generic primitive names represent most of the available differential I/O standards:

- IBUFDS (input buffer)
- IBUFGDS (clock input buffer)
- OBUFDS (output buffer)
- OBUFTDS (3-state output buffer)
- IOBUFDS (input/output buffer)
- IBUFDS_DIFF_OUT (input buffer)
- IOBUFDS_DIFF_OUT (input/output buffer)

IBUF and IBUFG

Signals used as inputs to Virtex-6 devices must use an input buffer (IBUF). The generic Virtex-6 FPGA IBUF primitive is shown in Figure 1-16.

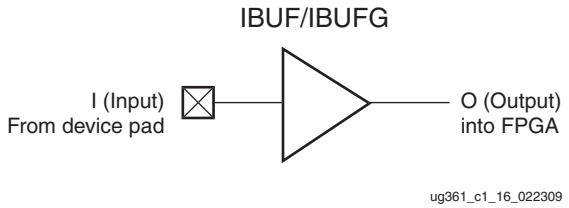


Figure 1-16: **Input Buffer (IBUF/IBUFG) Primitives**

The IBUF and IBUFG primitives are the same. IBUFGs are used when an input buffer is used as a clock input. In the Xilinx software tools, an IBUFG is automatically placed at clock input sites.

OBUF

An output buffer (OBUF) must be used to drive signals from Virtex-6 devices to external output pads. A generic Virtex-6 FPGA OBUF primitive is shown in Figure 1-17.

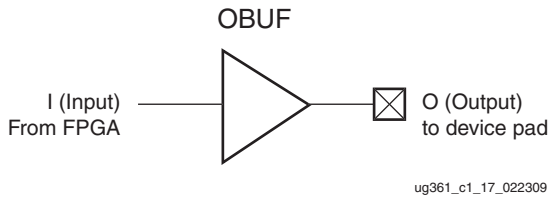


Figure 1-17: **Output Buffer (OBUF) Primitive**

OBUFT

The generic 3-state output buffer OBUFT, shown in Figure 1-18, typically implements 3-state outputs or bidirectional I/O.

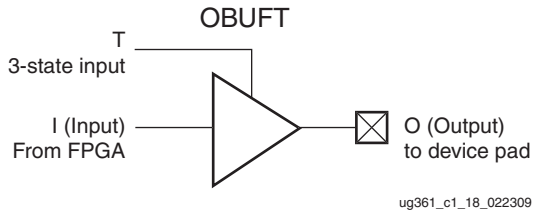


Figure 1-18: **3-State Output Buffer (OBUFT) Primitive**

IOBUF

The IOBUF primitive is needed when bidirectional signals require both an input buffer and a 3-state output buffer with an active High 3-state pin. Figure 1-19 shows a generic Virtex-6 FPGA IOBUF.

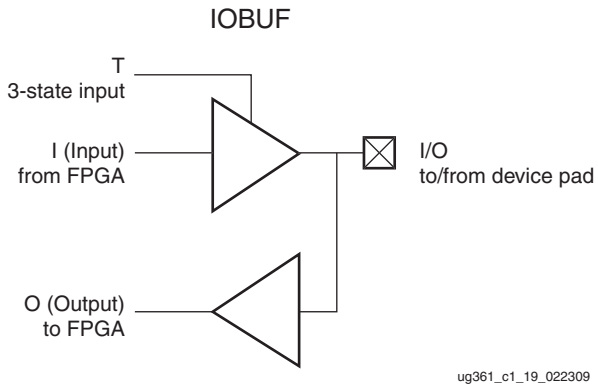


Figure 1-19: **Input/Output Buffer (IOBUF) Primitive**

IBUFDS and IBUFGDS

The usage and rules corresponding to the differential primitives are similar to the single-ended SelectIO primitives. Differential SelectIO primitives have two pins to and from the device pads to show the P and N channel pins in a differential pair. N channel pins have a "B" suffix.

Figure 1-20 shows the differential input buffer primitive.

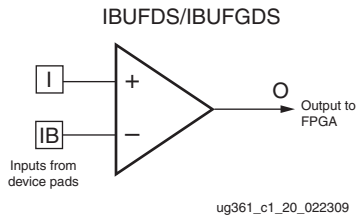


Figure 1-20: **Differential Input Buffer Primitive (IBUFDS/IBUFGDS)**

OBUFDS

Figure 1-21 shows the differential output buffer primitive.

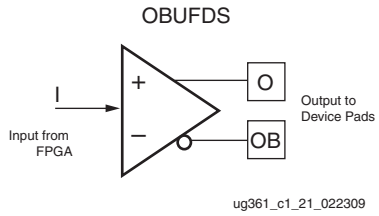


Figure 1-21: **Differential Output Buffer Primitive (OBUFDS)**

OBUFTDS

Figure 1-22 shows the differential 3-state output buffer primitive.

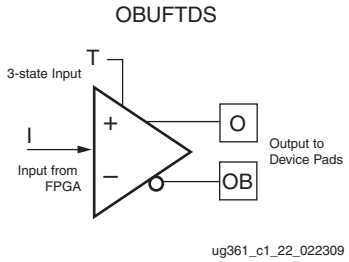


Figure 1-22: **Differential 3-state Output Buffer Primitive (OBUFTDS)**

IOBUFDS

Figure 1-23 shows the differential input/output buffer primitive.

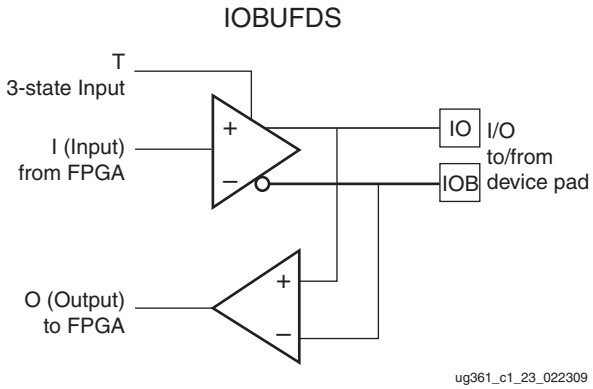


Figure 1-23: **Differential Input/Output Buffer Primitive (IOBUFDS)**

IBUFDS_DIFF_OUT

Figure 1-24 shows the differential input buffer primitive with a complementary output (OB). This primitive is only recommended for use by experienced Xilinx designers.

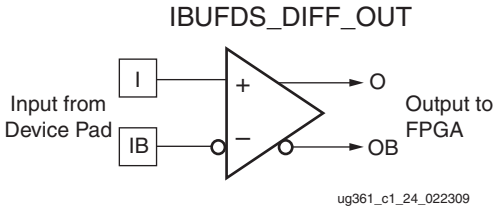


Figure 1-24: **Differential Input Buffer Primitive (IBUFDS_DIFF_OUT)**

IOBUFDS_DIFF_OUT

Figure 1-25 shows the differential input/output buffer primitive with a complimentarily output (OB). This primitive is only recommended for use by experienced Xilinx designers with DDR2 and DDR3 applications.

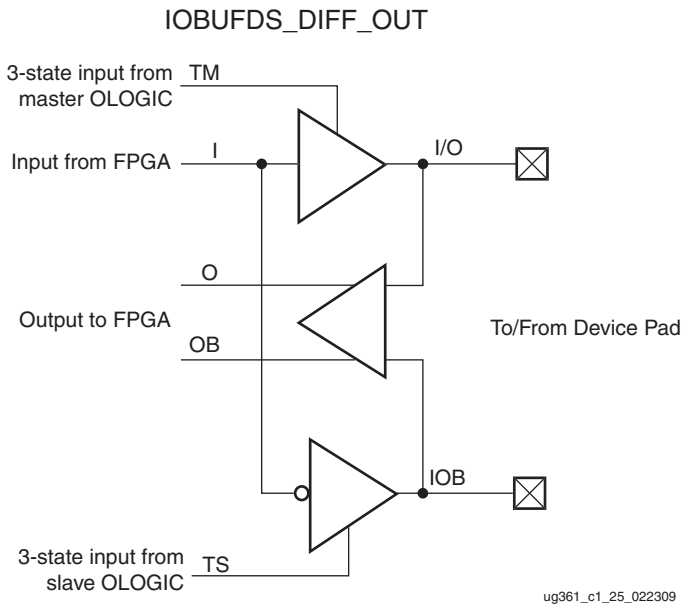


Figure 1-25: **Differential Input/Output Buffer Primitive (IOBUFDS_DIFF_OUT)**

Virtex-6 FPGA SelectIO Attributes/Constraints

Access to some Virtex-6 FPGA I/O resource features (e.g., location constraints, input delay, output drive strength, and slew rate) is available through the attributes/constraints associated with these features. For more information a Constraints Guide is available on the Xilinx web site with syntax examples and VHDL/Verilog reference code. This guide is available inside the Software Manuals at:
http://www.support.xilinx.com/support/software_manuals.htm

Location Constraints

The location constraint (LOC) must be used to specify the I/O location of an instantiated I/O primitive. The possible values for the location constraint are all the external port identifiers (e.g., A8, M5, AM6, etc.). These values are device and package size dependent.

The LOC attribute uses the following syntax in the UCF file:

```
INST <I/O_BUFFER_INSTANTIATION_NAME> LOC =
"<EXTERNAL_PORT_IDENTIFIER>";
```

Example:

```
INST MY_IO LOC=R7;
```

IOSTANDARD Attribute

The IOSTANDARD attribute is available to choose the values for an I/O standard for all I/O buffers. The supported I/O standards are listed in Table 1-33. The IOSTANDARD attribute uses the following syntax in the UCF file:

```
INST <I/O_BUFFER_INSTANTIATION_NAME> IOSTANDARD="<IOSTANDARD VALUE>";
```

The IOSTANDARD default for single-ended I/O is LVCMOS25, for differential I/Os the default is LVDS_25.

IBUF_LOW_PWR Attribute

The IBUF_LOW_PWR attribute is available for the following inputs:

- IBUF (LVDS)
- All V_{REF}-based inputs

The attribute is turned on by default. The IBUF_LOW_PWR attribute uses the following syntax in the UCF file:

```
INST <I/O_BUFFER_INSTANTIATION_NAME> IBUF_LOW_PWR=[TRUE|FALSE];
```

Refer to the timing analyzer to determine how performance might be affected by setting this attribute to TRUE.

Output Slew Rate Attributes

A variety of attribute values provide the option of choosing the desired slew rate for single-ended I/O output buffers. For LVCMOS output buffers (OBUF, OBUFT, and IOBUF), the desired slew rate can be specified with the SLEW attribute.

The allowed values for the SLEW attribute are:

- SLEW = SLOW (Default)
- SLEW = FAST

The SLEW attribute uses the following syntax in the UCF file:

```
INST <I/O_BUFFER_INSTANTIATION_NAME> SLEW = "<SLEW_VALUE>";
```

By the default, the slew rate for each output buffer is set to SLOW. This is the default used to minimize the power bus transients when switching non-critical signals.

Output Drive Strength Attributes

For LVCMOS output buffers (OBUF, OBUFT, and IOBUF), the desired drive strength (in mA) can be specified with the DRIVE attribute.

The allowed values for the DRIVE attribute are:

- DRIVE = 2
- DRIVE = 4
- DRIVE = 6
- DRIVE = 8
- DRIVE = 12 (Default)
- DRIVE = 16
- DRIVE = 24

LVCMOS12 only supports the 2, 4, 6, 8 mA DRIVE settings. LVCMOS15 and LVCMOS18 only support the 2, 4, 6, 8, 12, and 16 mA DRIVE settings.

The DRIVE attribute uses the following syntax in the UCF file:

```
INST <I/O_BUFFER_INSTANTIATION_NAME> DRIVE = "<DRIVE_VALUE>";
```

PULLUP/PULLDOWN/KEEPER for IBUF, OBUFT, and IOBUF

When using 3-state output (OBUFT) or bidirectional (IOBUF) buffers, the output can have a weak pull-up resistor, a weak pull-down resistor, or a weak "keeper" circuit. For input (IBUF) buffers, the input can also have a weak pull-up resistor, a weak pull-down resistor, or a weak "keeper" circuit. This feature can be invoked by adding the following possible constraint values to the relevant net of the buffers:

- PULLUP
- PULLDOWN
- KEEPER

Differential Termination Attribute

The differential termination (DIFF_TERM) attribute is designed for the Virtex-6 FPGA supported differential input I/O standards. It is used to turn the built-in, 100Ω, differential termination on or off.

The allowed values for the DIFF_TERM attribute are:

- TRUE
- FALSE (Default)

To specify the DIFF_TERM attribute, set the appropriate value in the generic map (VHDL) or inline parameter (Verilog) of the instantiated IBUFDS, IBUFGDS, IBUFDS_DIFF_OUT, or IOBUFDS_DIFF_OUT primitives. Refer to the ISE language templates or the Virtex-6 FPGA HDL Libraries Guide for the proper syntax for instantiating these primitives and setting the DIFF_TERM attribute.

Virtex-6 FPGA I/O Resource VHDL/Verilog Examples

The VHDL and Verilog example syntaxes to declare a standard for Virtex-6 FPGA I/O resources are found in the Virtex-6 FPGA Libraries Guide.

Specific Guidelines for I/O Supported Standards

The following sections provide an overview of the I/O standards supported by all Virtex-6 devices.

While most Virtex-6 FPGA I/O supported standards specify a range of allowed voltages, this chapter records typical voltage values only. Detailed information on each specification can be found on the Electronic Industry Alliance JEDEC web site at http://www.jedec.org.

LVCMOS (Low Voltage Complementary Metal Oxide Semiconductor)

LVCMOS is a widely used switching standard implemented in CMOS transistors. This standard is defined by JEDEC (JESD 8-5). The LVCMOS standards supported in Virtex-6 FPGAs are: LVCMOS12, LVCMOS15, LVCMOS18, and LVCMOS25.

Sample circuits illustrating both unidirectional and bidirectional LVCMOS termination techniques are shown in Figure 1-26 and Figure 1-27. These two diagrams show examples of source-series and parallel terminated topologies. Figure 1-26 shows a unidirectional source-series terminated topology.

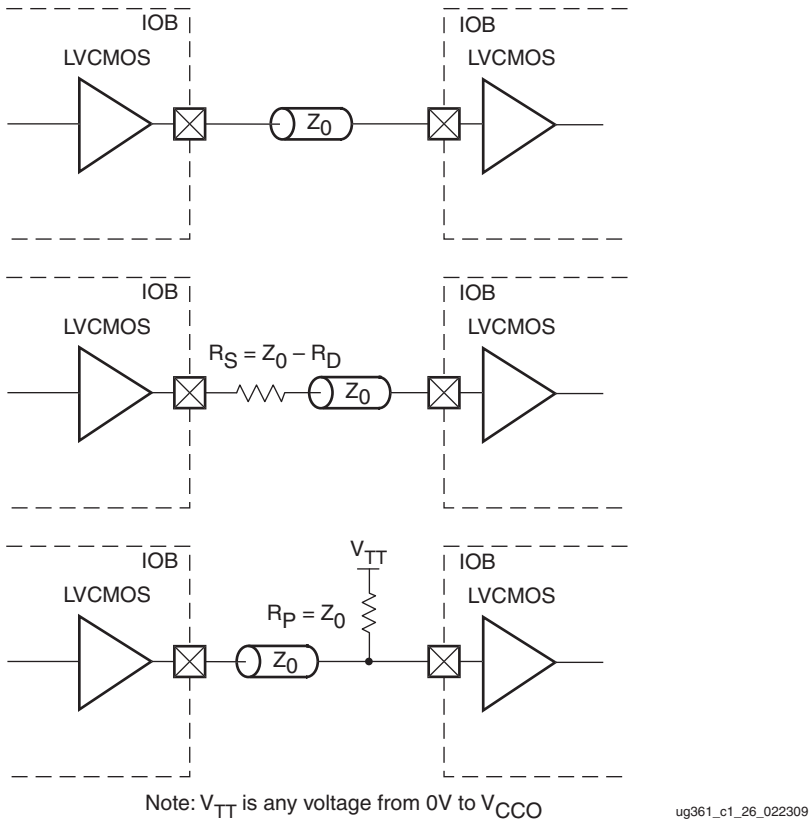


Figure 1-26: **LVCMOS Unidirectional Termination**

The unterminated and series-terminated examples are generally recommended for drive strengths up to 8 mA because overshoot might be less of a concern. Parallel termination should be considered for drive strengths higher than 8 mA because the possibility of overshoot becomes more prevalent. When using parallel termination, a V_{TT} value (typically $V_{CCO}/2$) should be chosen that still allows the driver to meet V_{IH}.

Figure 1-27 shows a bi-directional, parallel-terminated topology.

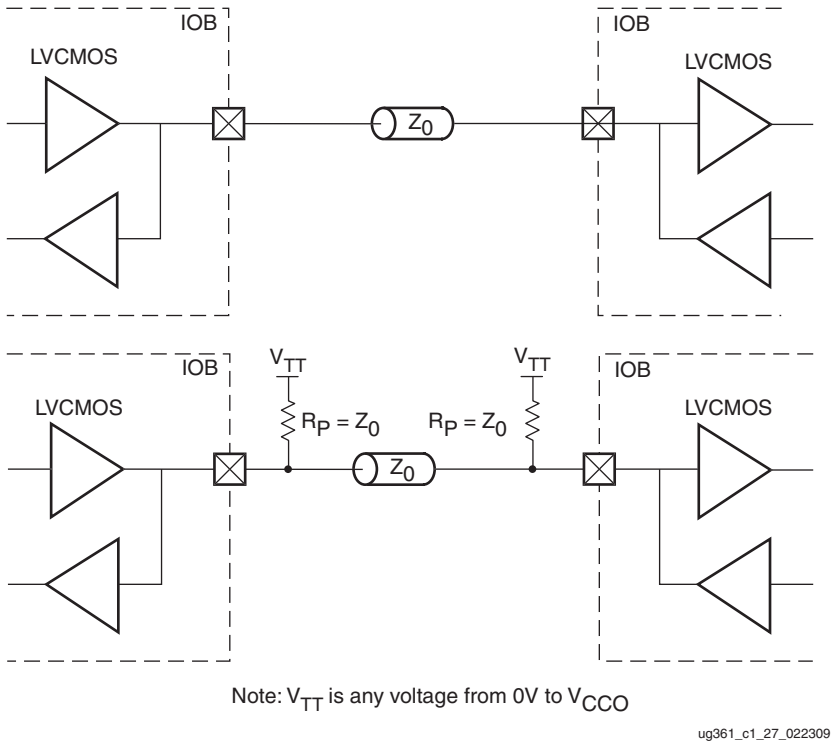


Figure 1-27: **LVCMOS Bidirectional Termination**

Table 1-4 details the allowed attributes that can be applied to the LVCMOS25 I/O standard.

Table 1-4: **Allowed Attributes for the LVCMOS25 I/O Standard**

Attributes	Primitives		
	IBUF/IBUFG	OBUF/OBUFT	IOBUF
IOSTANDARD	LVCMOS25	LVCMOS25	LVCMOS25
DRIVE	UNUSED	2, 4, 6, 8, 12, 16, 24	2, 4, 6, 8, 12, 16, 24
SLEW	UNUSED	{FAST, SLOW}	{FAST, SLOW}

Table 1-5 details the allowed attributes that can be applied to the LVCMOS18 and LVCMOS15 I/O standards.

Table 1-5: **Allowed Attributes for the LVCMOS18 and LVCMOS15 I/O Standard**

Attributes	Primitives		
	IBUF/IBUFG	OBUF/OBUFT	IOBUF
IOSTANDARD	LVCMOS18 LVCMOS15	LVCMOS18 LVCMOS15	LVCMOS18 LVCMOS15
DRIVE	UNUSED	2, 4, 6, 8, 12, 16	2, 4, 6, 8, 12, 16
SLEW	UNUSED	{FAST, SLOW}	{FAST, SLOW}

Table 1-6 details the allowed attributes that can be applied to the LVCMOS12 I/O standard.

Table 1-6: **Allowed Attributes for the LVCMOS12 I/O Standard**

Attributes	Primitives		
	IBUF/IBUFG	OBUF/OBUFT	IOBUF
IOSTANDARD	LVCMOS12	LVCMOS12	LVCMOS12
DRIVE	UNUSED	2, 4, 6, 8	2, 4, 6, 8
SLEW	UNUSED	{FAST, SLOW}	{FAST, SLOW}

LVDCI (Low Voltage Digitally Controlled Impedance)

Using these I/O buffers configures the outputs as controlled impedance drivers. The receiver of LVDCI is identical to a LVCMOS receiver. Some I/O standards, such as LVCMOS, must have a drive impedance that matches the characteristic impedance of the driven line. Virtex-6 devices provide a controlled impedance output driver to provide series termination without external source termination resistors. The impedance is set by the common external reference resistors, with resistance equal to the trace characteristic impedance, Z_0.

Sample circuits illustrating both unidirectional and bidirectional termination techniques for a controlled impedance driver are shown in Figure 1-28 and Figure 1-29. The DCI I/O standards supporting a controlled impedance driver are: LVDCI_15, LVDCI_18, and LVDCI_25.

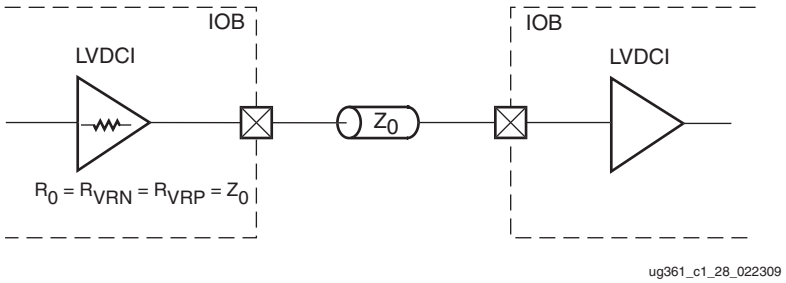


Figure 1-28: **Controlled Impedance Driver with Unidirectional Termination**

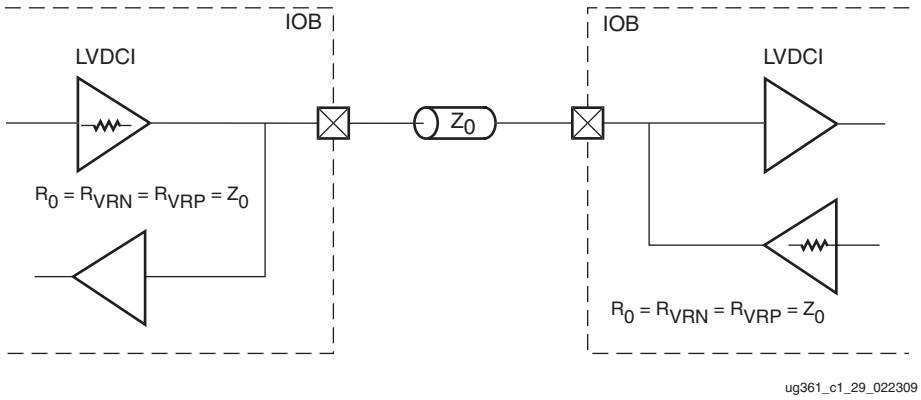


Figure 1-29: **Controlled Impedance Driver with Bidirectional Termination**

LVDCI_DV2

A controlled impedance driver with half impedance (source termination) can also provide drivers with one half of the impedance of the reference resistors. This allows reference resistors to be twice as large, thus reducing static power consumption through VRN/VRP. The I/O standards supporting a controlled impedance driver with half impedance are: LVDCI_DV2_15, LVDCI_DV2_18, and LVDCI_DV2_25. Figure 1-30 and Figure 1-31 illustrate a controlled driver with half impedance unidirectional and bidirectional termination.

To match the drive impedance to Z_0 when using a driver with half impedance, the reference resistor R must be twice Z_0.

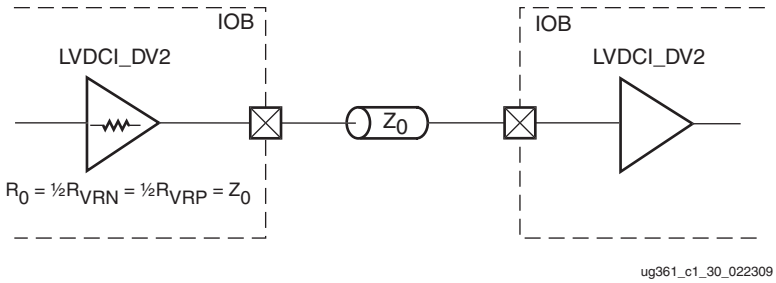


Figure 1-30: **Controlled Impedance Driver with Half Impedance Unidirectional Termination**

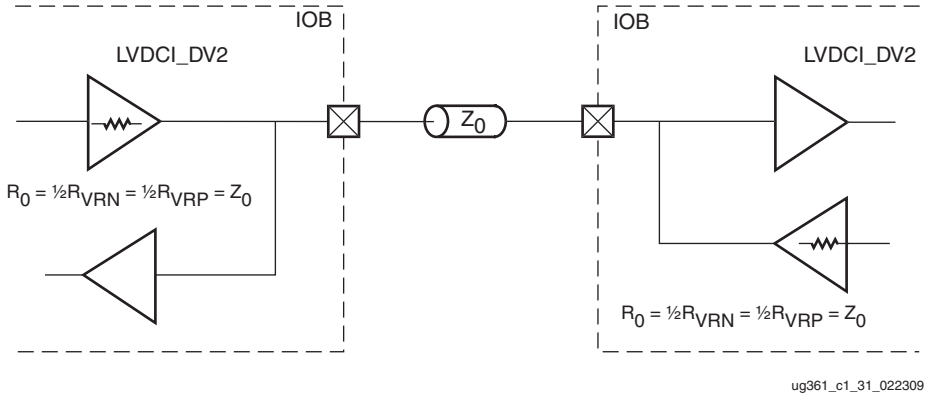


Figure 1-31: **Controlled Impedance Driver with Half Impedance Bidirectional Termination**

There are no drive strength settings for LVDCI drivers. When the driver impedance is one-half of the VRN/VRP reference resistors, it is indicated by the addition of DV2 to the attribute name.

Table 1-7 lists the LVCMOS, LVDCI, and LVDCI_DV2 voltage specifications.

Table 1-7: **LVCMOS, LVDCI, and LVDCI_DV2 DC Voltage Specifications at Various Voltage References**

Standard	+2.5V			+1.8V			+1.5V			+1.2V(2)		
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
V_{CCO} [V]	2.3	2.5	2.625	1.7	1.8	1.9	1.4	1.5	1.6	1.1	1.2	1.3
V_{IH} [V]	1.7	–	V_{CCO}+0.3	65% V_{CCO}	–	V_{CCO}+0.3	65% V_{CCO}	–	V_{CCO}+0.3	65% V_{CCO}	–	V_{CCO}+0.3
V_{IL} [V]	–0.3	–	0.7	–0.3	–	35% V_{CCO}	–0.3	–	35% V_{CCO}	0.3	–	35% V_{CCO}
V_{OH} [V]	V_{CCO}–0.4	–	–	V_{CCO}–0.45	–	–	75% V_{CCO}	–	–	75% V_{CCO}	–	–
V_{OL} [V]	–	–	0.4	–	–	0.45	–	–	25% V_{CCO}	–	–	25% V_{CCO}
I_{IN} [µA]	–	–	± 5	–	–	± 5	–	–	± 10	–	–	± 10

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Only LVCMOS is supported at + 1.2V with valid DRIVE attributes of 2, 4, 6, 8.

HSLVDCI (High-Speed Low Voltage Digitally Controlled Impedance)

The HSLVDCI standard is intended for bidirectional use. The driver is identical to LVDCI, while the input is identical to HSTL and SSTL. By using a V_{REF}-referenced input, HSLVDCI allows greater input sensitivity at the receiver than when using a single-ended LVCMOS-type receiver.

A sample circuit illustrating bidirectional termination techniques for an HSLVDCI controlled impedance driver is shown in Figure 1-32. The DCI I/O standards supporting a controlled impedance driver with a V_{REF} referenced input are: HSLVDCI_15, HSLVDCI_18, and HSLVDCI_25.

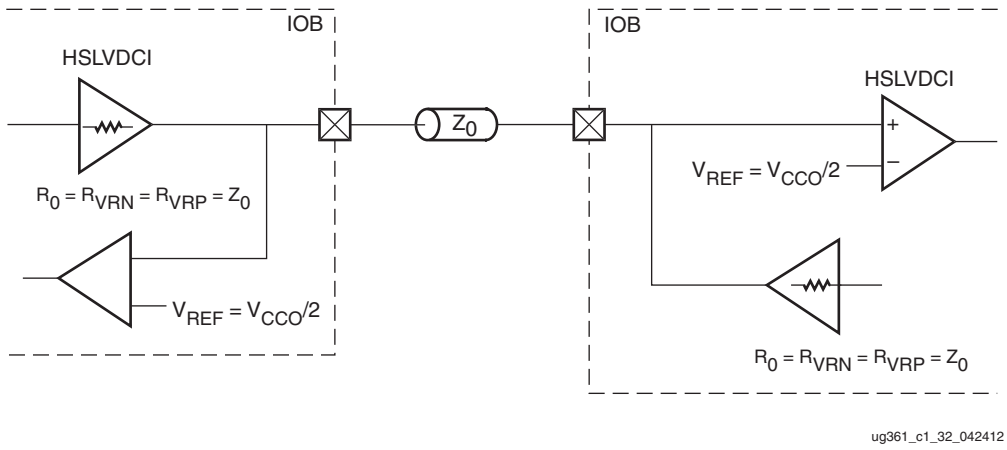


Figure 1-32: **HSLVDCI Controlled Impedance Driver with Bidirectional Termination**

For output DC voltage specifications, refer to the LVDCI V_{OH} and V_{OL} entries in Table 1-7 LVCMOS, LVDCI, and LVDCI_DV2 DC Voltage Specifications at Various Voltage References. Table 1-8 lists the input DC voltage specifications when using HSLVDCI. Valid values of V_{CCO} are 1.5V, 1.8V, and 2.5V. Select V_{REF} to provide the optimum noise margin in specific use conditions.

Table 1-8: **HSLVDCI Input DC Voltage Specifications**

Standard	Min	Typ	Max
V_{REF}	–	$V_{CCO}/2$	–
V_{IH}	$V_{REF} + 0.1$	–	–
V_{IL}	–	–	$V_{REF} - 0.1$

HSTL (High-Speed Transceiver Logic)

The High-Speed Transceiver Logic (HSTL) standard is a general purpose high-speed bus standard sponsored by IBM (EIA/JESD8-6). The 1.5V and 1.8V have four variations or classes. To support clocking high speed memory interfaces, a differential version of this standard was added. Virtex-6 FPGA I/O supports all four classes for 1.5V and 1.8V and the differential versions of classes I and II. These differential versions of the standard require a differential amplifier input buffer and a push-pull output buffer.

HSTL_ I, HSTL_ III, HSTL_ I_18, HSTL_ III_18, HSTL_I_12

HSTL_I uses $V_{CCO}/2$ as a parallel termination voltage (V_{TT}). HSTL_III uses V_{CCO} as a parallel termination voltage (V_{TT}). HSTL_I and HSTL_III are intended to be used in unidirectional links.

HSTL_ I_DCI, HSTL_ III_DCI, HSTL_ I_DCI_18, HSTL_ III_DCI_18

HSTL_I_DCI provides on-chip split thevenin termination powered from V_{CCO}, creating an equivalent parallel termination voltage (V_{TT}) of $V_{CCO}/2$. HSTL_III_DCI provides on-chip single termination powered from V_{CCO}. HSTL_I_DCI and HSTL_III_DCI are intended to be used in unidirectional links.

HSTL_ II and HSTL_ II_18

HSTL_II uses $V_{CCO}/2$ as a parallel termination voltage (V_{TT}). HSTL_II is intended to be used in bidirectional links.

HSTL_ II_DCI and HSTL_ II_DCI_18

HSTL_II_DCI provides on-chip split thevenin termination powered from V_{CCO}, creating an equivalent termination voltage of $V_{CCO}/2$. HSTL_II_DCI is intended to be used in bidirectional links.

HSTL_ II_T_DCI and HSTL_ II_T_DCI_18

HSTL_ II_T_DCI and HSTL_ II_T_DCI_18 provide on-chip split-thevenin termination powered from V_{CCO} that creates an equivalent termination voltage of $V_{CCO}/2$ when these standards are 3-stated. When not 3-stated, these two standards do not have termination.

DIFF_HSTL_ II and DIFF_HSTL_II_18

Differential HSTL class II pairs complimentary single-ended HSTL_II type drivers with a differential receiver. Differential HSTL class II is intended to be used in bidirectional links. Differential HSTL can also be used for differential clock and DQS signals in memory interface designs.

DIFF_HSTL_II_DCI and DIFF_HSTL_II_DCI_18

Differential HSTL class II pairs complimentary single-ended HSTL_II type drivers with a differential receiver, including on-chip differential split-thevenin termination. Differential HSTL class II is intended to be used in bidirectional links. Differential HSTL can also be used for differential clock and DQS signals in memory interface designs.

DIFF_HSTL_II_T_DCI and DIFF_HSTL_II_T_DCI_18

These standards are almost the same as the DIFF_HSTL_II_DCI and DIFF_HSTL_II_DCI_18 standards except that the termination is only present when the driver is 3-stated.

DIFF_HSTL_I and DIFF_HSTL_I_18

Differential HSTL class I pairs complimentary single-ended HSTL_I type drivers with a differential receiver. Differential HSTL class I is intended to be used in unidirectional links.

DIFF_HSTL_I_DCI and DIFF_HSTL_I_DCI_18

Differential HSTL class I pairs complimentary single-ended HSTL_I type drivers with a differential receiver, including on-chip differential split-thevenin termination. Differential HSTL class I is intended to be used in unidirectional links.

HSTL Class I

Figure 1-33 shows a sample circuit illustrating a valid termination technique for HSTL Class I.

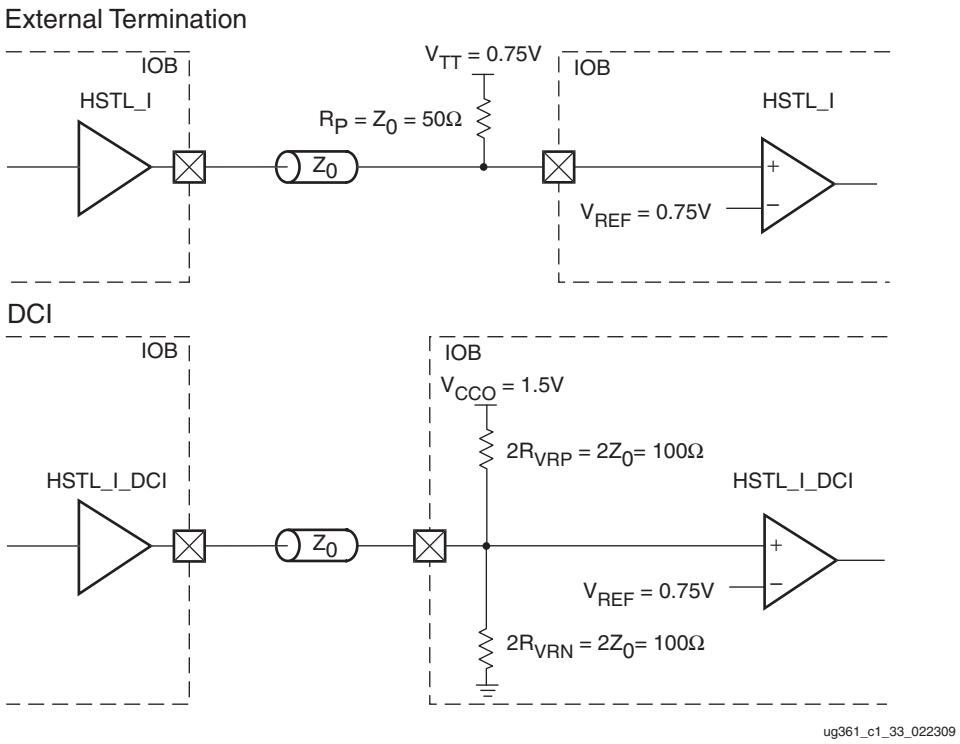


Figure 1-33: **HSTL Class I Termination**

Table 1-9 lists the HSTL Class I DC voltage specifications.

Table 1-9: **HSTL Class I DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.40	1.50	1.60
V_{REF} (2)	0.68	0.75	0.90
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IH}	$V_{REF} + 0.1$	–	–
V_{IL}	–	–	$V_{REF} - 0.1$
V_{OH}	$V_{CCO} - 0.4$	–	–
V_{OL}	–	–	0.4
I_{OH} at V_{OH} (mA)(1)	–8	–	–
I_{OL} at V_{OL} (mA)(1)	8	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."

Differential HSTL Class I

Figure 1-34 shows a sample circuit illustrating a valid termination technique for differential HSTL Class I (1.5V) with unidirectional termination.

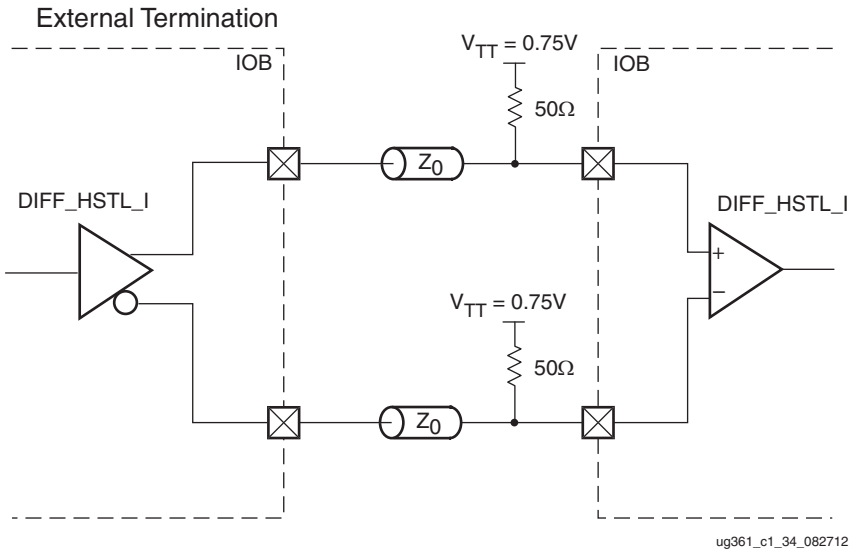


Figure 1-34: **Differential HSTL (1.5V) Class I Unidirectional Termination**

Figure 1-35 shows a sample circuit illustrating a valid termination technique for differential HSTL Class I (1.5V) with unidirectional DCI termination.

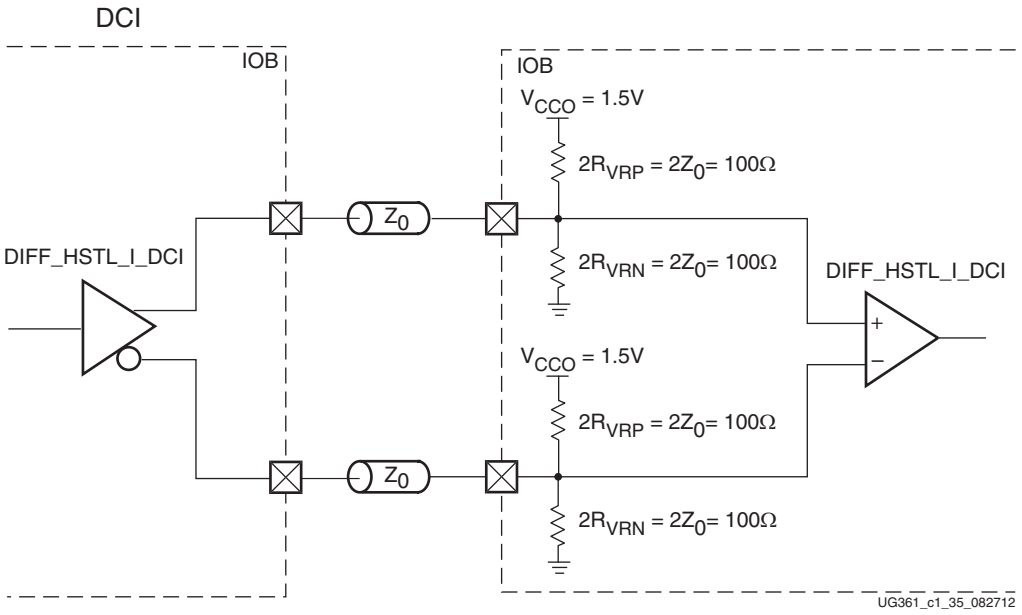


Figure 1-35: **Differential HSTL (1.5V) Class I DCI Unidirectional Termination**

Table 1-10 lists the differential HSTL Class I DC voltage specifications.

Table 1-10: **Differential HSTL Class I DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.40	1.50	1.60
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)	–0.30	–	V_{CCO} + 0.30
V_{DIFF} (DC)	0.20	–	V_{CCO} + 0.60
V_{CM} (DC)(1)	0.68	–	0.90
V_{DIFF} (AC)	0.40	–	V_{CCO} + 0.60
V_X (Crossover)(2)	0.68	–	0.90

Notes:

1. Common mode voltage: $V_{CM} = V_P - ((V_P - V_N)/2)$
2. Crossover point: V_X where $V_P - V_N = 0$ (AC coupled)

HSTL Class II

Figure 1-36 shows a sample circuit illustrating a valid termination technique for HSTL Class II (1.5V) with unidirectional termination.

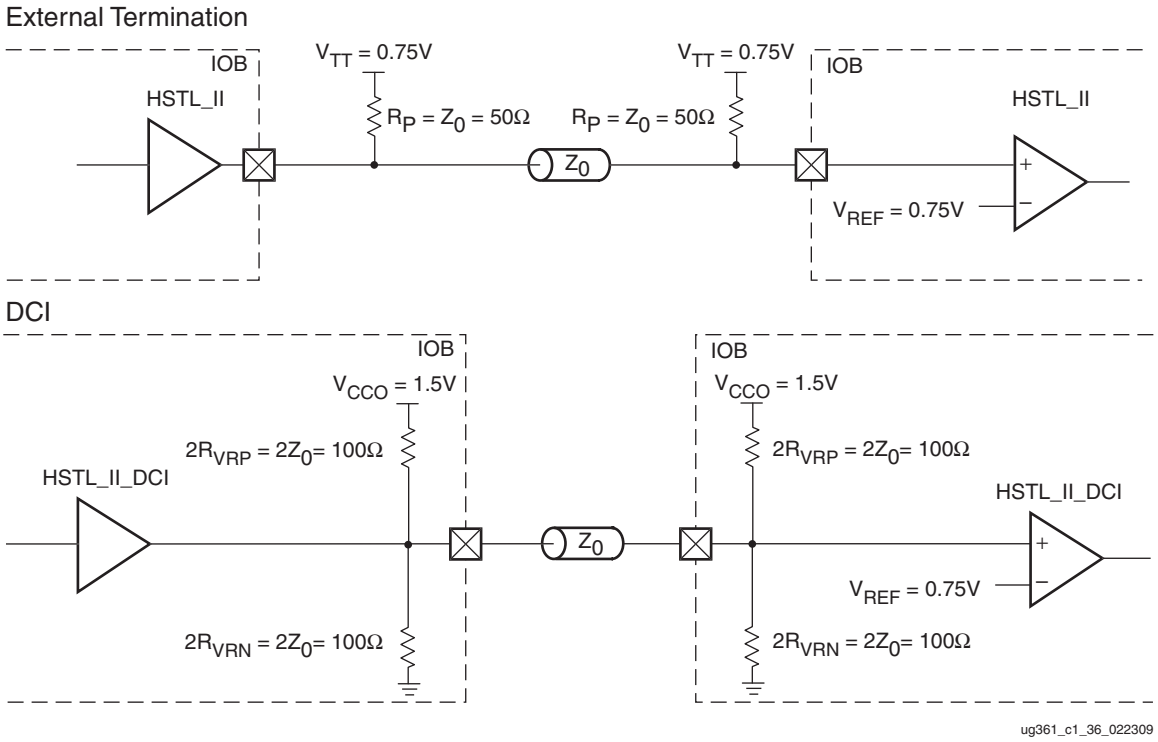


Figure 1-36: **HSTL (1.5V) Class II Unidirectional Termination**

Figure 1-37 shows a sample circuit illustrating a valid termination technique for HSTL Class II (1.5V) with bidirectional termination.

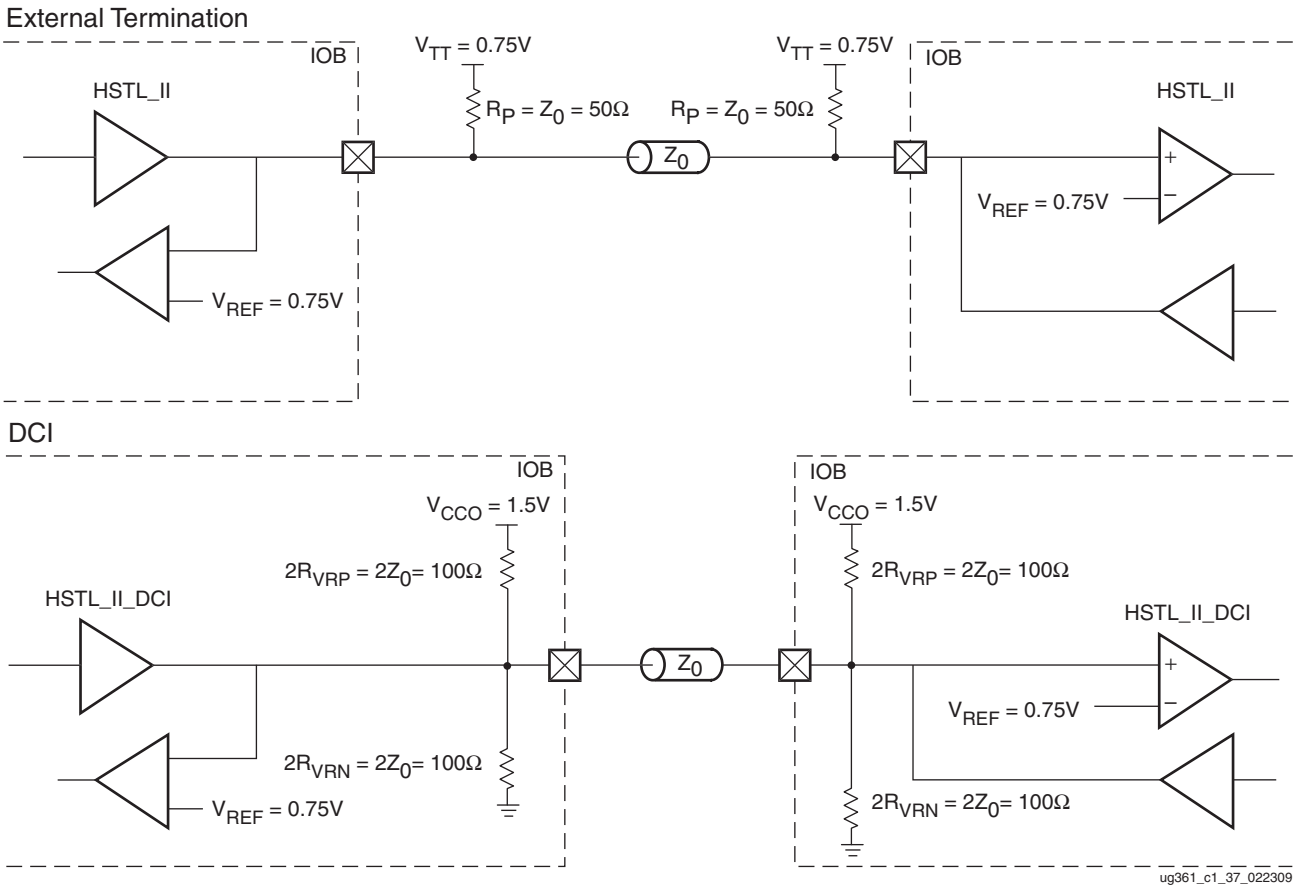


Figure 1-37: **HSTL (1.5V) Class II Bidirectional Termination**

Table 1-11 lists the HSTL (1.5V) Class II DC voltage specifications.

Table 1-11: **HSTL (1.5V) Class II DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.40	1.50	1.60
V_{REF} (2)	0.68	0.75	0.90
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IH}	$V_{REF} + 0.1$	–	–
V_{IL}	–	–	$V_{REF} - 0.1$
V_{OH}	$V_{CCO} - 0.4$	–	–
V_{OL}	–	–	0.4
I_{OH} at V_{OH} (mA)(1)	–16	–	–

Table 1-11: **HSTL (1.5V) Class II DC Voltage Specifications *(Cont'd)***

Parameter	Min	Typ	Max
I_{OL} at V_{OL} (mA)(1) (3)	16	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."
3. HSTL_II_T_DCI has a weaker driver than HSTL_II_DCI.

Differential HSTL Class II

Figure 1-38 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.5V) with unidirectional termination.

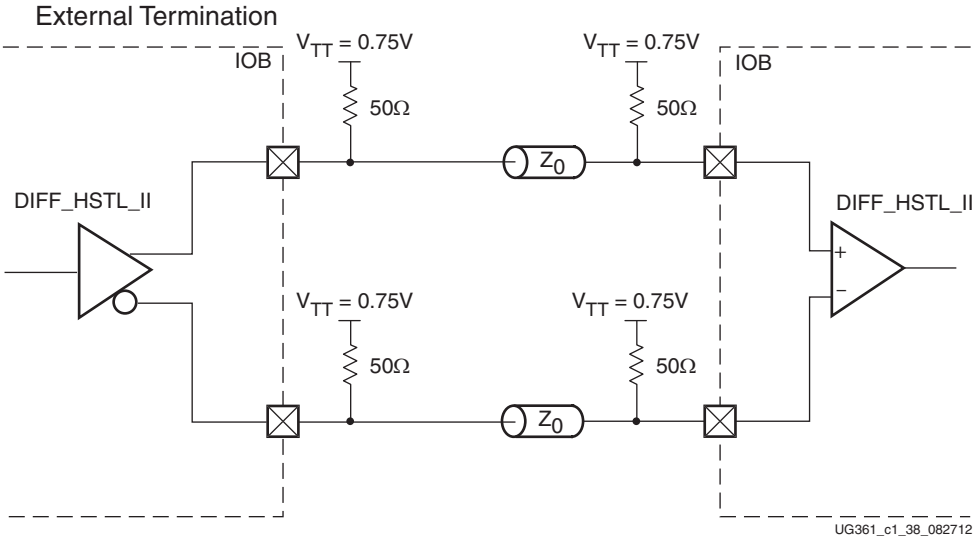


Figure 1-38: **Differential HSTL (1.5V) Class II Unidirectional Termination**

Figure 1-39 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.5V) with unidirectional DCI termination.

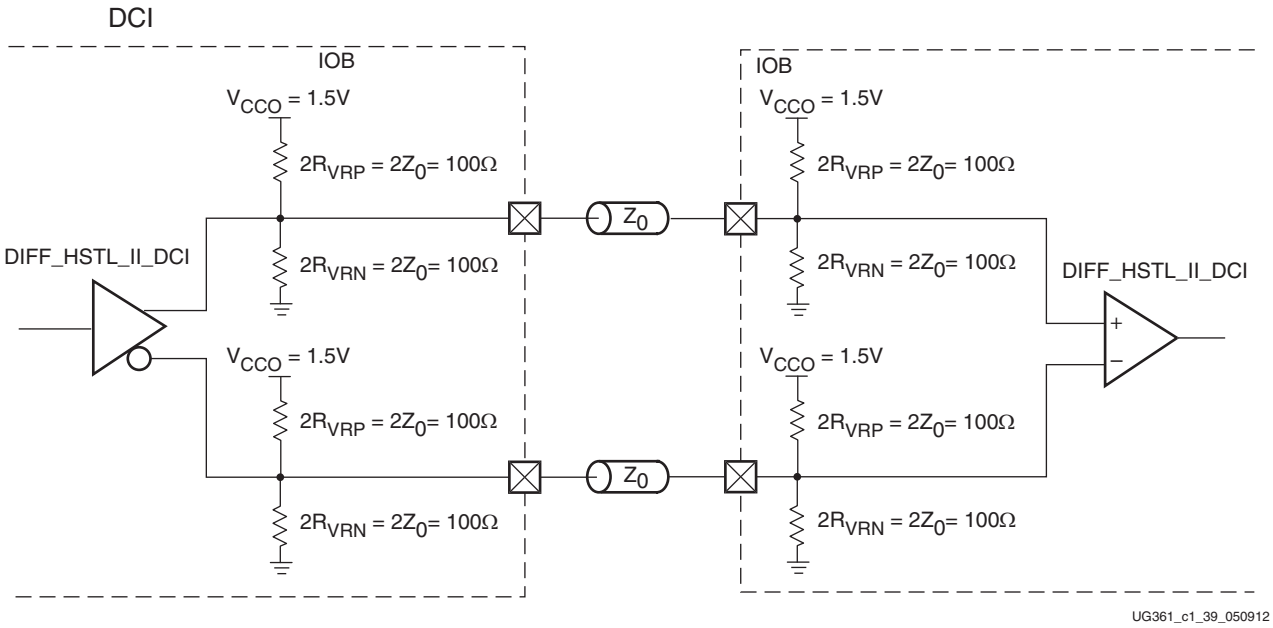


Figure 1-39: **Differential HSTL (1.5V) Class II DCI Unidirectional Termination**

Figure 1-40 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.5V) with bidirectional termination.

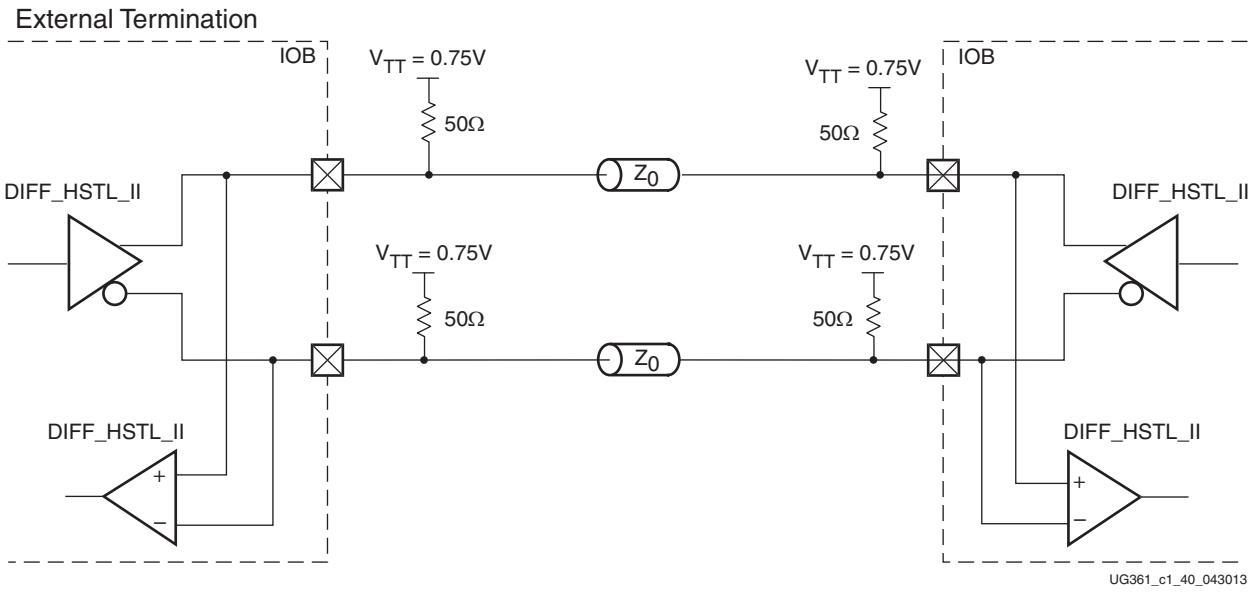


Figure 1-40: **Differential HSTL (1.5V) Class II Bidirectional Termination**

Figure 1-41 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.5V) with bidirectional DCI termination.

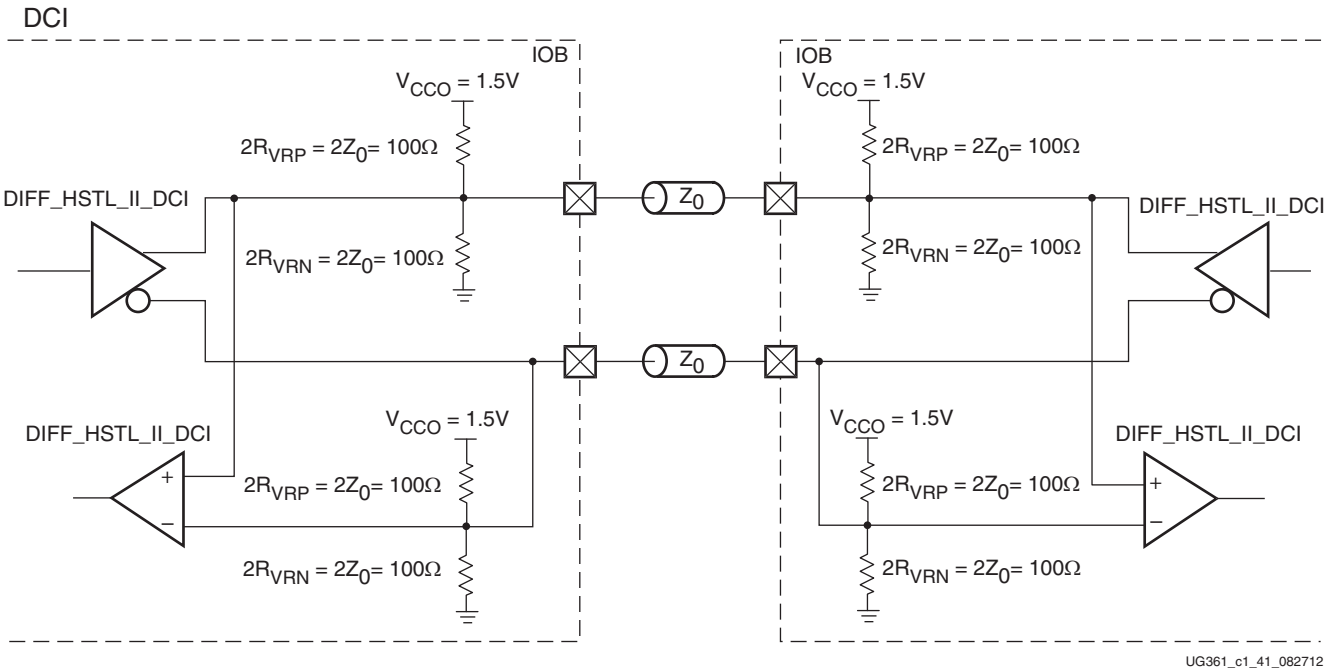


Figure 1-41: **Differential HSTL (1.5V) Class II DCI Bidirectional Termination**

Table 1-12 lists the differential HSTL Class II DC voltage specifications.

Table 1-12: **Differential HSTL Class II DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.40	1.50	1.60
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)	–0.30	–	V_{CCO} + 0.30
V_{DIFF} (DC)	0.20	–	V_{CCO} + 0.60
V_{CM} (DC)(1)	0.68	–	0.90
V_{DIFF} (AC)	0.40	–	V_{CCO} + 0.60
V_X (Crossover)(2)	0.68	–	0.90

Notes:

1. Common mode voltage: $V_{CM} = V_P - ((V_P - V_N)/2)$
2. Crossover point: V_X where $V_P - V_N = 0$ (AC coupled)

HSTL Class III

Figure 1-42 shows a sample circuit illustrating a valid termination technique for HSTL Class III.

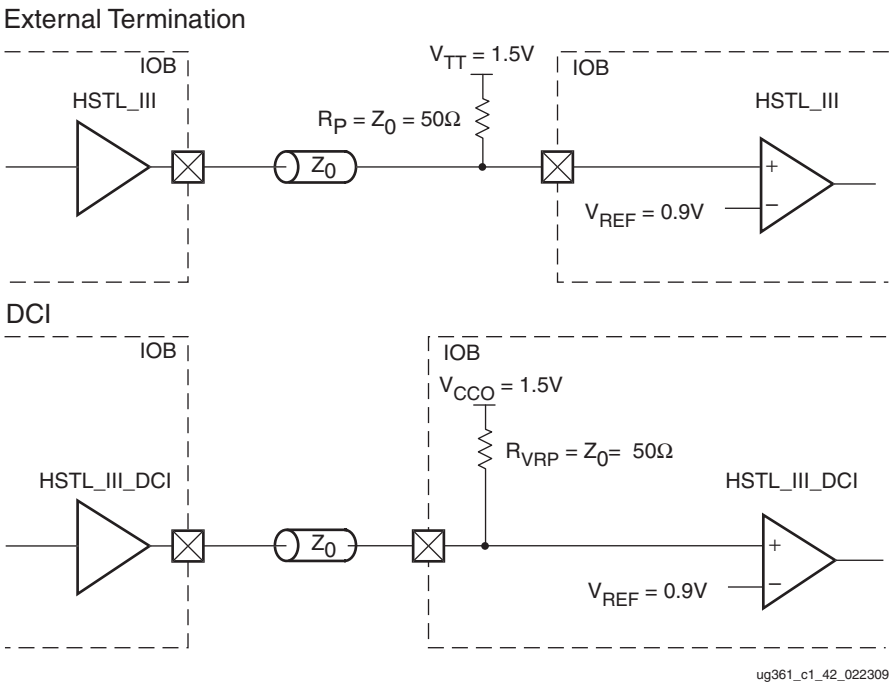


Figure 1-42: **HSTL Class III Termination**

Table 1-13 lists the HSTL Class III DC voltage specifications.

Table 1-13: **HSTL Class III DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.40	1.50	1.60
V_{REF} (2)	–	0.90	–
V_{TT}	–	V_{CCO}	–
V_{IH}	V_{REF} + 0.1	–	–
V_{IL}	–	–	V_{REF} – 0.1
V_{OH}	V_{CCO} – 0.4	–	–
V_{OL}	–	–	0.4
I_{OH} at V_{OH} (mA)(1)	–8	–	–
I_{OL} at V_{OL} (mA)(1)	24	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."

HSTL_II_T_DCI (1.5V) Split-Thevenin Termination

Figure 1-43 shows a sample circuit illustrating a valid termination technique for HSTL_II_T_DCI (1.5V) with on-chip split-thevenin termination. In this bidirectional case, when 3-stated, the termination is invoked on the receiver and not on the driver.

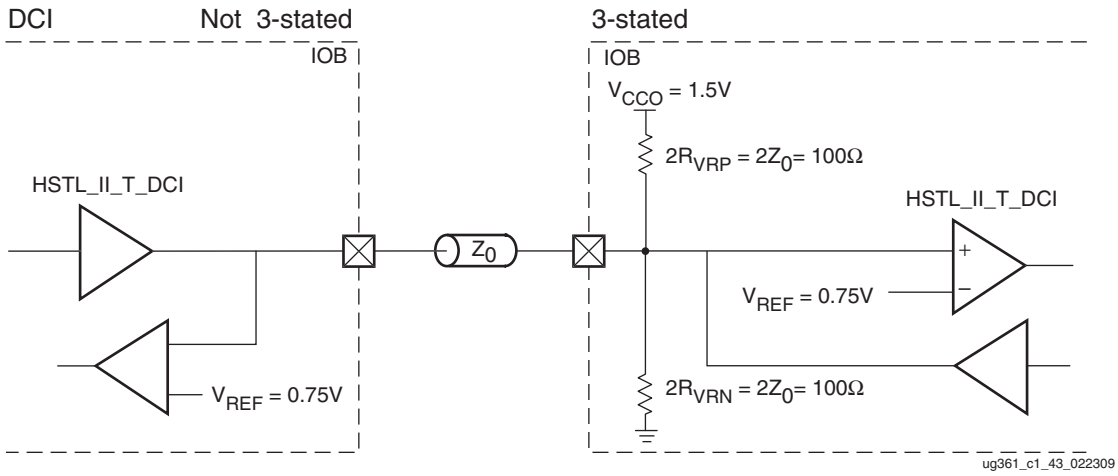


Figure 1-43: **HSTL_II_T_DCI (1.5V) Split-Thevenin Termination**

HSTL Class I (1.8V)

Figure 1-44 shows a sample circuit illustrating a valid termination technique for HSTL Class I (1.8V).

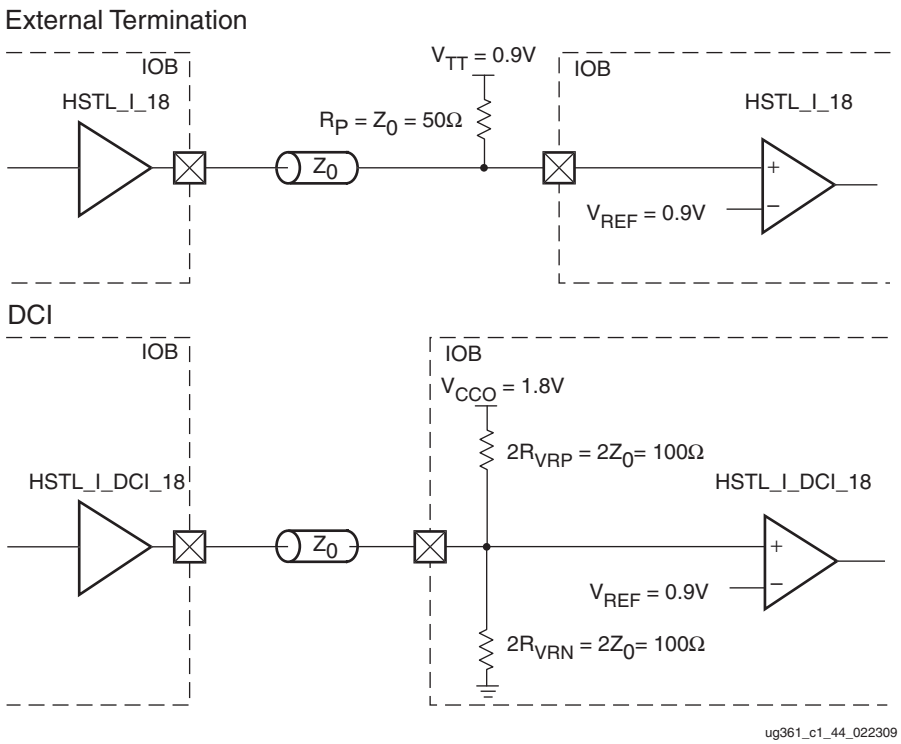


Figure 1-44: **HSTL Class I (1.8V) Termination**

Table 1-14 lists the HSTL Class I (1.8V) DC voltage specifications.

Table 1-14: **HSTL Class I (1.8V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{REF} (2)	0.83	0.9	1.08
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IH}	$V_{REF} + 0.1$	–	–
V_{IL}	–	–	$V_{REF} - 0.1$
V_{OH}	$V_{CCO} - 0.4$	–	–
V_{OL}	–	–	0.4
I_{OH} at V_{OH} (mA)(1)	–8	–	–
I_{OL} at V_{OL} (mA)(1)	8	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."

Differential HSTL Class I (1.8V)

Figure 1-45 shows a sample circuit illustrating a valid termination technique for differential HSTL Class I (1.8V) with unidirectional termination.

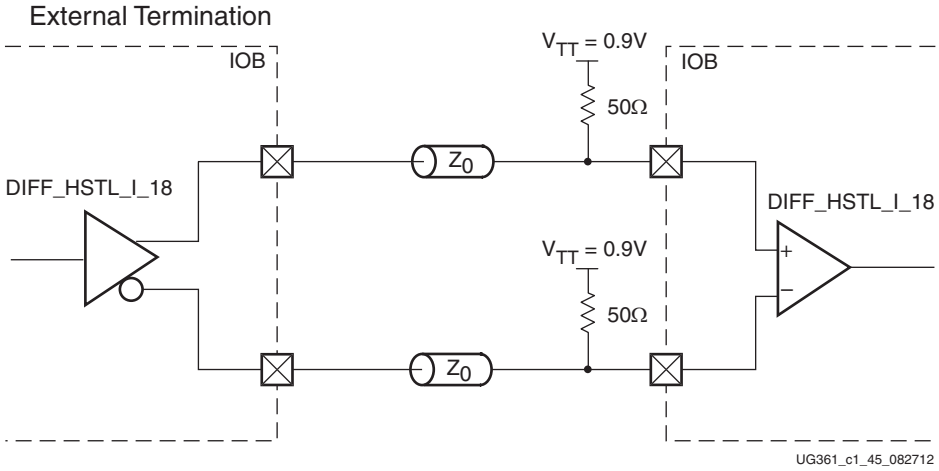


Figure 1-45: **Differential HSTL (1.8V) Class I Unidirectional Termination**

Figure 1-46 shows a sample circuit illustrating a valid termination technique for differential HSTL Class I (1.8V) with unidirectional DCI termination.

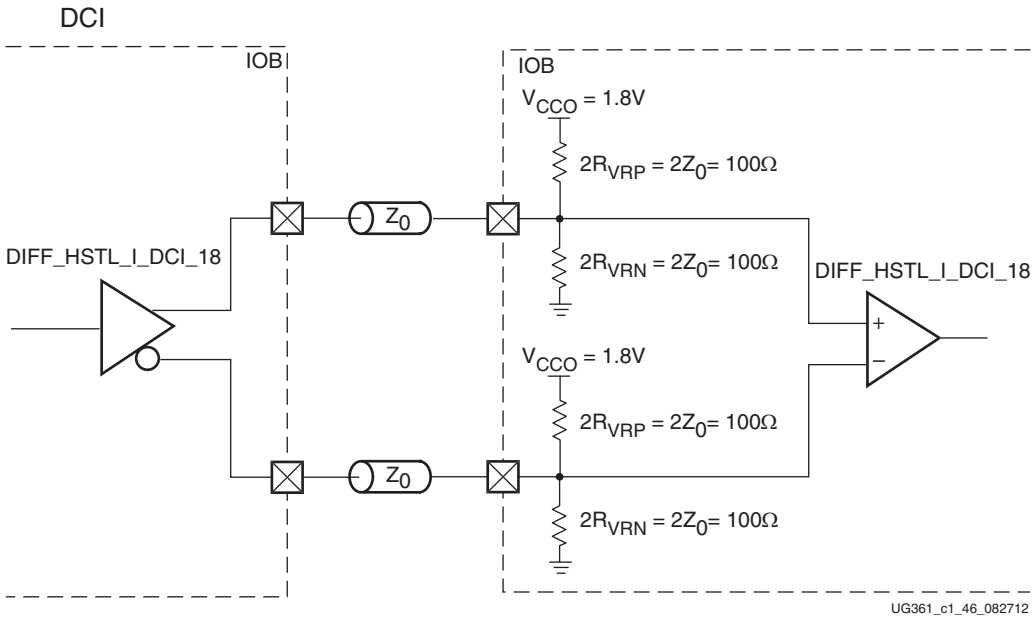


Figure 1-46: **Differential HSTL (1.8V) Class I DCI Unidirectional Termination**

Table 1-15 lists the differential HSTL Class I (1.8V) DC voltage specifications.

Table 1-15: **Differential HSTL Class I (1.8V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)	–0.30	–	V_{CCO} + 0.30
V_{DIFF} (DC)	0.20	–	V_{CCO} + 0.60
V_{CM} (DC)(1)	0.83	–	1.08
V_{DIFF} (AC)	0.40	–	V_{CCO} + 0.60
V_X (Crossover)(2)	0.83	–	1.08

Notes:

1. Common mode voltage: $V_{CM} = V_P - ((V_P - V_N)/2)$
2. Crossover point: V_X where $V_P - V_N = 0$ (AC coupled)

HSTL Class II (1.8V)

Figure 1-47 shows a sample circuit illustrating a valid termination technique for HSTL Class II (1.8V) with unidirectional termination.

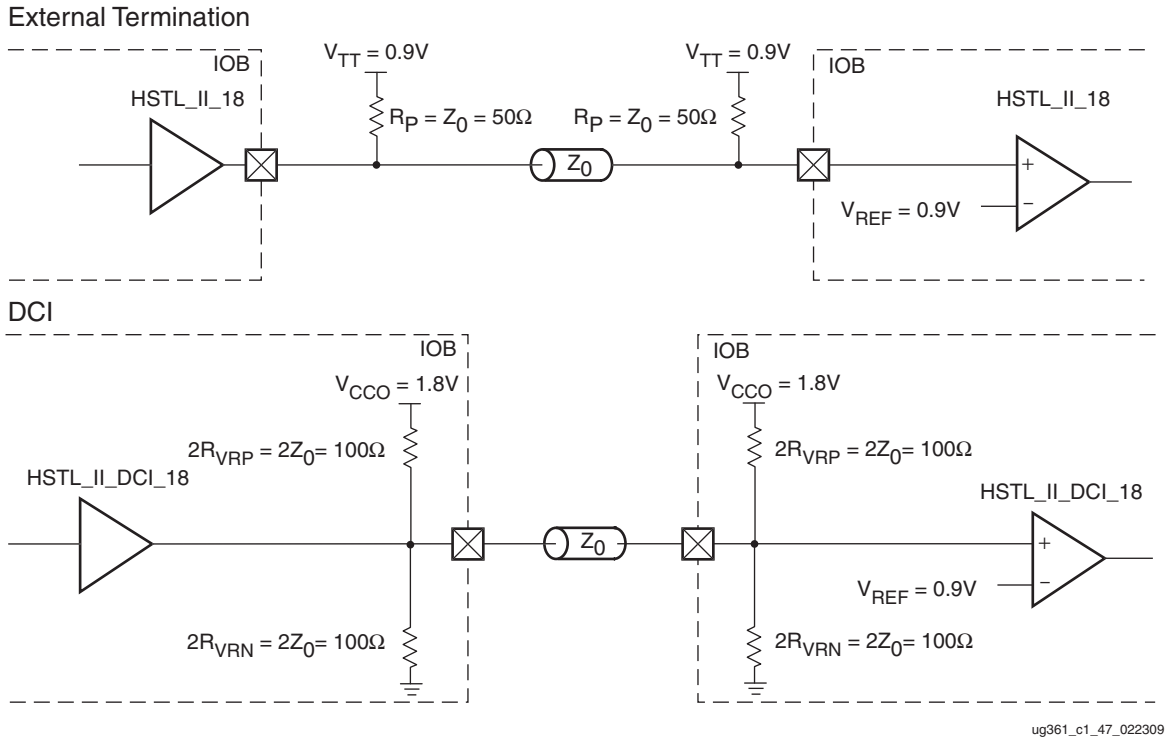


Figure 1-47: **HSTL Class II (1.8V) with Unidirectional Termination**

Figure 1-48 shows a sample circuit illustrating a valid termination technique for HSTL Class II (1.8V) with bidirectional termination.

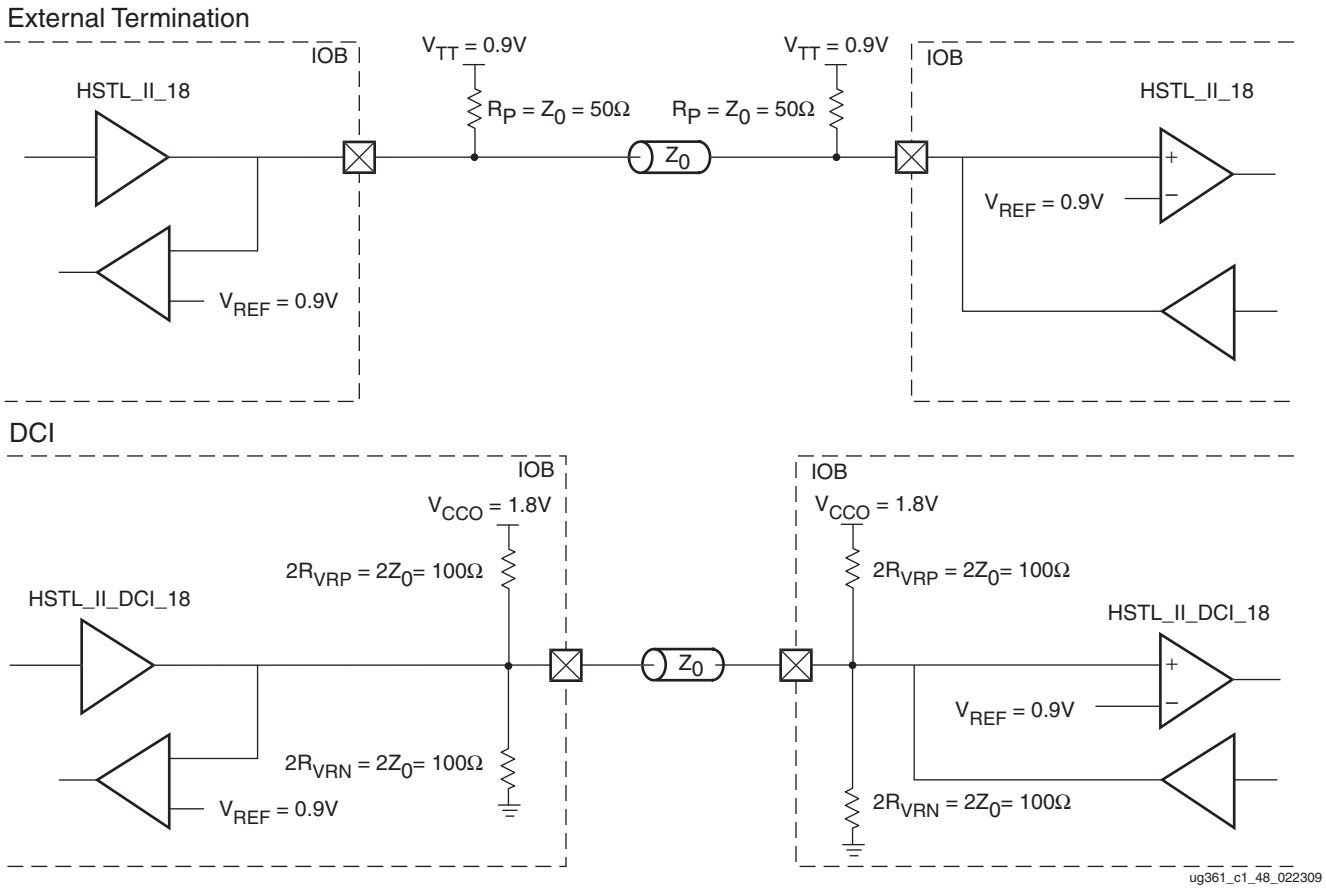


Figure 1-48: **HSTL Class II (1.8V) with Bidirectional Termination**

Table 1-16 lists the HSTL Class II (1.8V) DC voltage specifications.

Table 1-16: **HSTL Class II (1.8V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{REF} (2)	–	0.9	–
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IH}	$V_{REF} + 0.1$	–	–
V_{IL}	–	–	$V_{REF} - 0.1$
V_{OH}	$V_{CCO} - 0.4$	–	–
V_{OL}	–	–	0.4
I_{OH} at V_{OH} (mA)(1)	–16	–	–
I_{OL} at V_{OL} (mA)(1)	16	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."

Differential HSTL Class II (1.8V)

Figure 1-49 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.8V) with unidirectional termination.

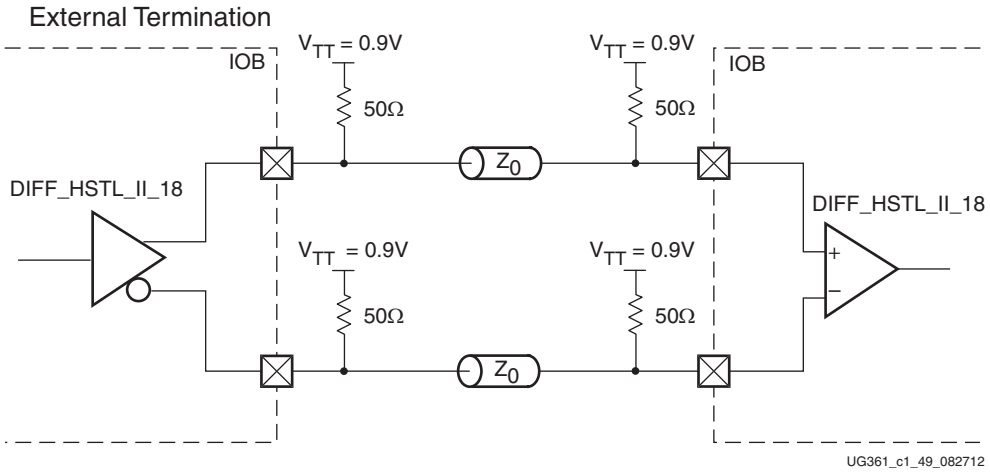


Figure 1-49: **Differential HSTL (1.8V) Class II Unidirectional Termination**

Figure 1-50 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.8V) with unidirectional DCI termination.

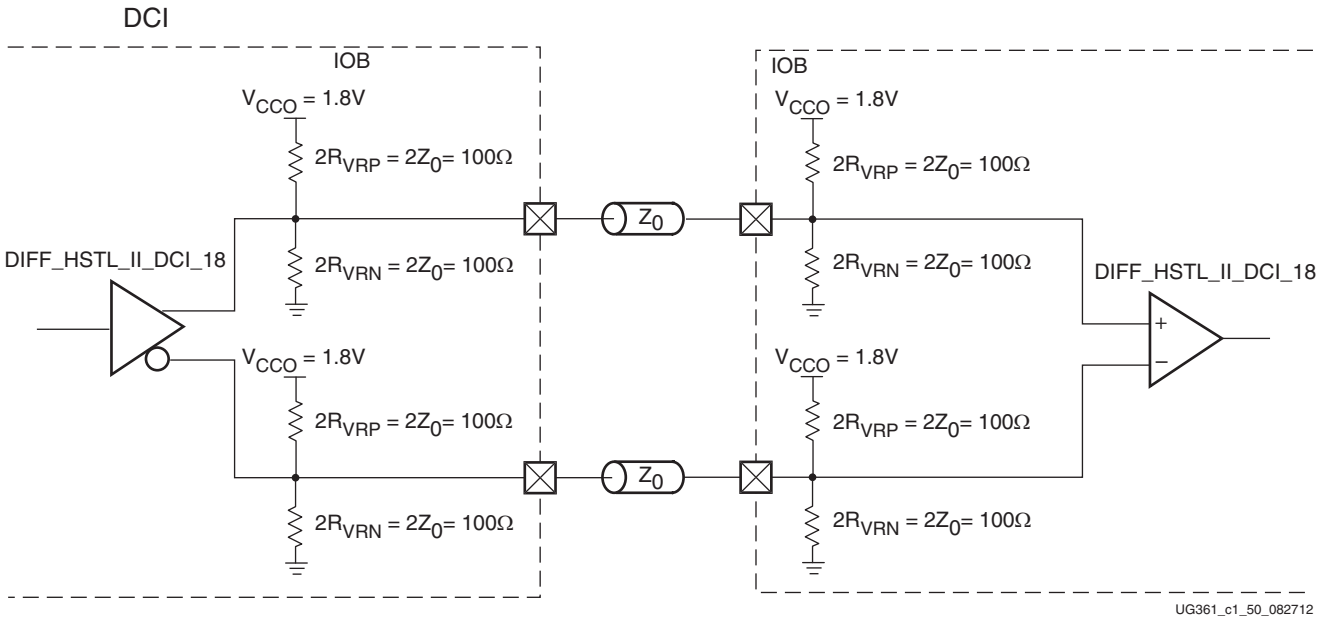


Figure 1-50: **Differential HSTL (1.8V) Class II DCI Unidirectional Termination**

Figure 1-51 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.8V) with bidirectional termination.

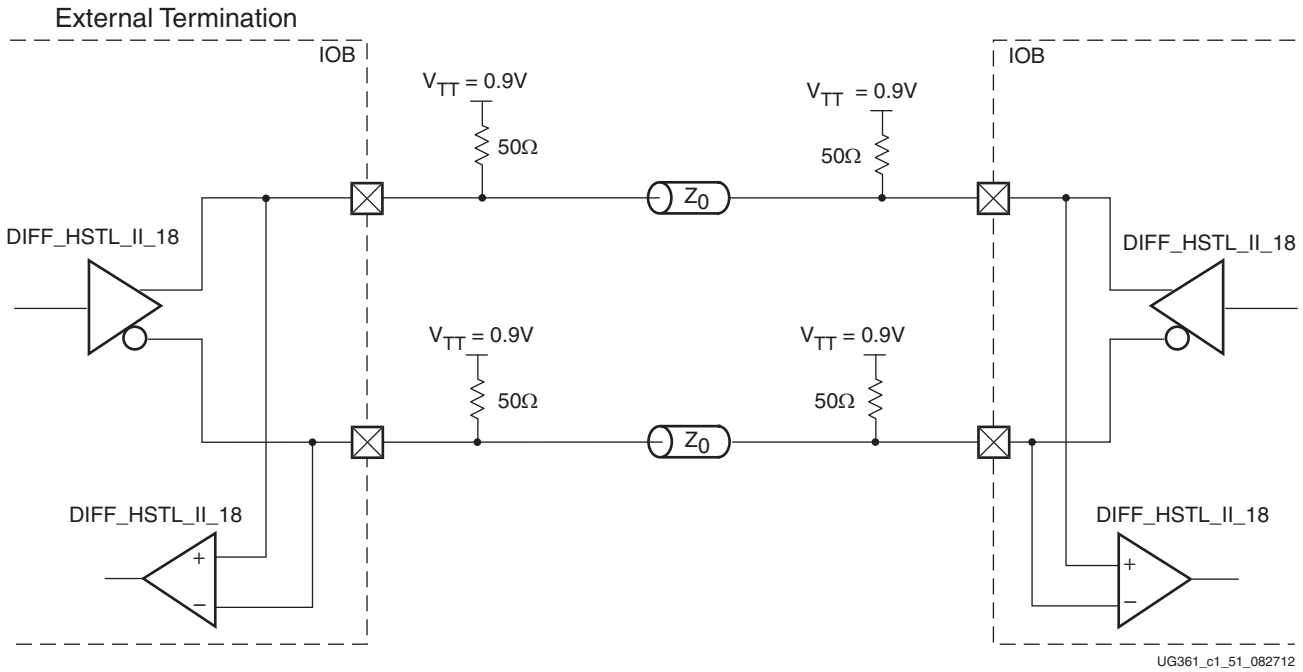


Figure 1-51: **Differential HSTL (1.8V) Class II Bidirectional Termination**

Figure 1-52 shows a sample circuit illustrating a valid termination technique for differential HSTL Class II (1.8V) with bidirectional DCI termination.

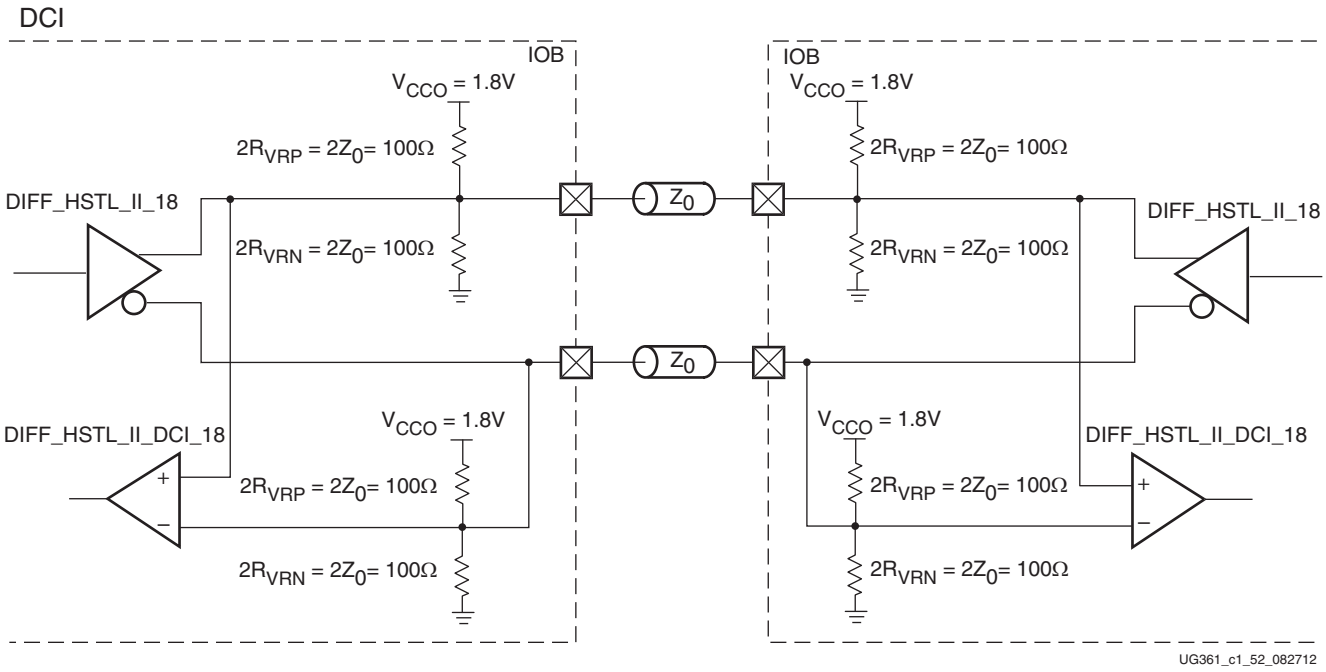


Figure 1-52: **Differential HSTL (1.8V) Class II DCI Bidirectional Termination**

Table 1-17 lists the differential HSTL Class II (1.8V) DC voltage specifications.

Table 1-17: **Differential HSTL Class II (1.8V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)	–0.30	–	V_{CCO} + 0.30
V_{DIFF} (DC)	0.20	–	V_{CCO} + 0.60
V_{CM} (DC)(1)	0.83	–	1.08
V_{DIFF} (AC)	0.40	–	V_{CCO} + 0.60
V_X (Crossover)(2)	0.83	–	1.08

Notes:

1. Common mode voltage: $V_{CM} = V_P - ((V_P - V_N)/2)$
2. Crossover point: V_X where $V_P - V_N = 0$ (AC coupled)

HSTL Class III (1.8V)

Figure 1-53 shows a sample circuit illustrating a valid termination technique for HSTL Class III (1.8V).

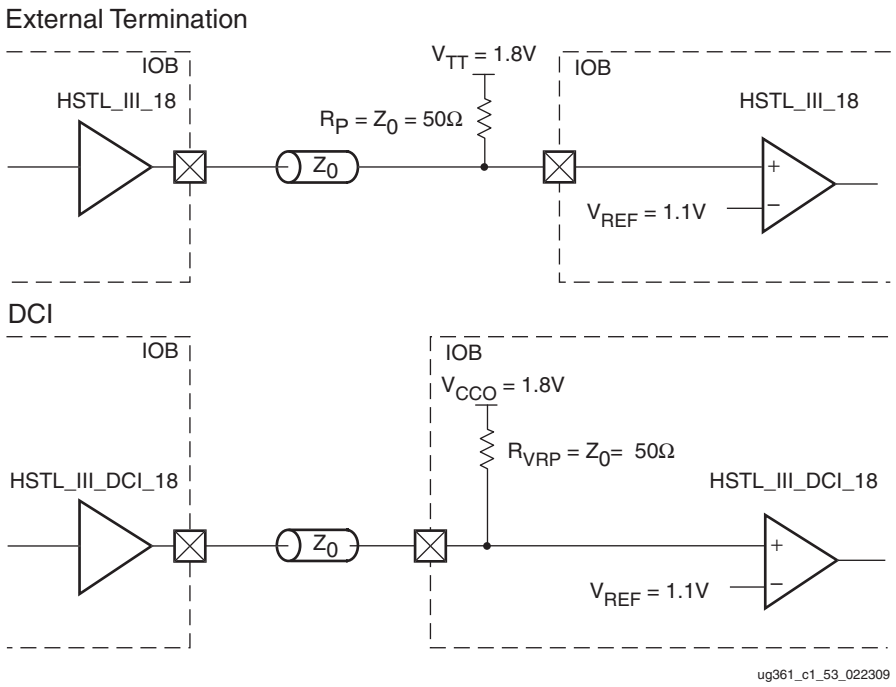


Figure 1-53: **HSTL Class III (1.8V) Termination**

Table 1-18 lists the HSTL Class III (1.8V) DC voltage specifications.

Table 1-18: **HSTL Class III (1.8V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{REF} (2)	–	1.1	–
V_{TT}	–	V_{CCO}	–
V_{IH}	V_{REF} + 0.1	–	–
V_{IL}	–	–	V_{REF} – 0.1
V_{OH}	V_{CCO} – 0.4	–	–
V_{OL}	–	–	0.4
I_{OH} at V_{OH} (mA)(1)	–8	–	–
I_{OL} at V_{OL} (mA)(1)	24	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."

HSTL_II_T_DCI_18 (1.8V) Split-Thevenin Termination

Figure 1-54 shows a sample circuit illustrating a valid termination technique for HSTL_II_T_DCI_18 (1.8V) with on-chip split-thevenin termination. In this bidirectional case, when 3-stated, the termination is invoked on the receiver and not on the driver.

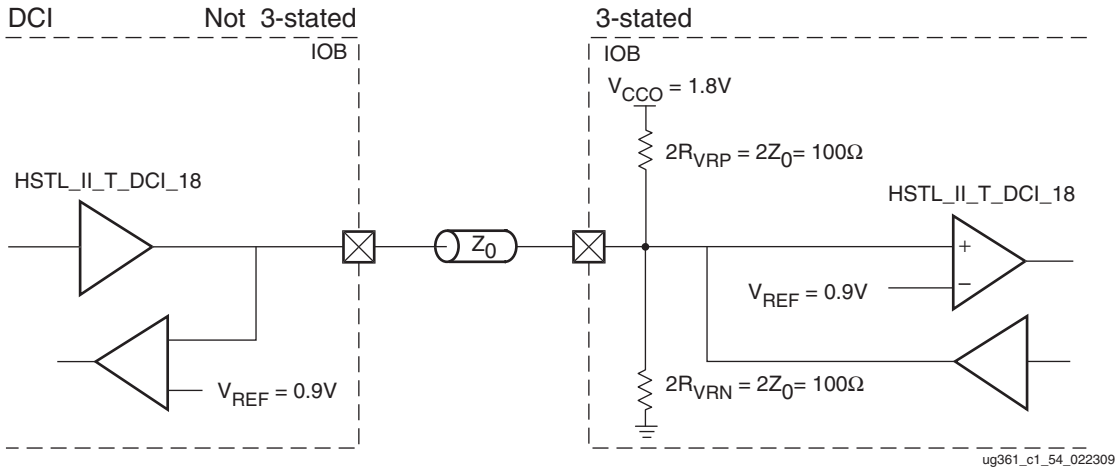


Figure 1-54: **HSTL_II_T_DCI_18 Split-Thevenin Termination**

HSTL Class I (1.2V)

Figure 1-55 shows a sample circuit illustrating a valid termination technique for HSTL Class I (1.2V). It is used for unidirectional links.

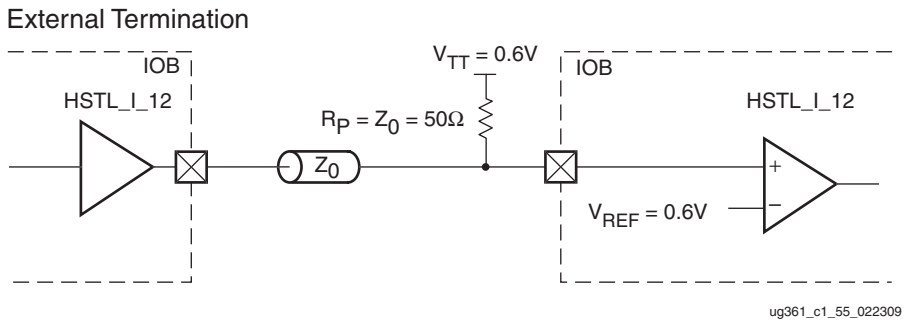


Figure 1-55: **HSTL Class I (1.2V) Termination**

Table 1-19 lists the HSTL Class I (1.2V) DC voltage specifications.

Table 1-19: **HSTL Class I (1.2V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.14	1.2	1.26
V_{REF} (2)	$V_{CCO} \times 0.48$	0.6	$V_{CCO} \times 0.52$
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IH}	$V_{REF} + 0.08$	–	–
V_{IL}	–	–	$V_{REF} - 0.08$
V_{OH}	$V_{CCO} - .0.315$	–	–
V_{OL}	–	–	0.315
I_{OH} at V_{OH} (mA)(1)	–6.3	–	–
I_{OL} at V_{OL} (mA)(1)	6.3	–	–

Notes:

1. V_{OL} and V_{OH} for lower drive currents are sample tested.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."

SSTL (Stub-Series Terminated Logic)

The Stub-Series Terminated Logic (SSTL) for 2.5V (SSTL2) and 1.8V (SSTL18) standards are for general purpose memory buses. SSTL2 is defined by the JEDEC standard JESD8-9B and SSTL18 is defined by the JEDEC standard JESD8-15. The SSTL2 standard has two classes; Class I is for unidirectional and class II is for bidirectional signaling. Virtex-6 FPGA I/O supports both standards for single-ended signaling and differential signaling. This standard requires a differential amplifier input buffer and a push-pull output buffer.

SSTL2_I, SSTL18_I

Class I signaling uses V_{TT} ($V_{CCO}/2$) as a parallel termination voltage to a 50 Ω resistor at the receiver. A series resistor (25 Ω at 2.5V, 20 Ω at 1.8V) must be connected to the transmitter output.

SSTL2_I_DCI, SSTL18_I_DCI

The DCI transmitter provides the internal series resistance (25 Ω at 2.5V, 20 Ω at 1.8V). The DCI receiver has an internal split thevenin termination powered from V_{CCO} creating an equivalent V_{TT} voltage and termination impedance.

SSTL2_II, SSTL18_II, SSTL15

Class II signaling uses V_{TT} ($V_{CCO}/2$) as a parallel termination voltage to a 50 Ω resistor at the receiver and transmitter respectively. A series resistor (25 Ω at 2.5V, 20 Ω at 1.8V) must be connected to the transmitter output for a unidirectional link. (No series resistance is required for 1.5V.) For a bidirectional link, 25 Ω series resistors must be connected to the transmitters of the transceivers.

SSTL2_II_DCI, SSTL18_II_DCI, SSTL15_DCI

The DCI circuits have a split thevenin termination powered from V_{CCO} and an internal series resistor (25 Ω at 2.5V, 20 Ω at 1.8V). For a unidirectional link the internal series resistance is supplied only for the transmitter. (No series resistance is required for 1.5V.) A bidirectional link has the internal series resistor for both transmitters.

DIFF_SSTL2_I, DIFF_SSTL18_I

Differential SSTL 2.5V and 1.8V Class I pairs complementary single-ended SSTL_I type drivers with a differential receiver.

DIFF_SSTL2_I_DCI, DIFF_SSTL18_I_DCI

Differential SSTL 2.5V and 1.8V Class I pairs complementary single-ended SSTL_II type drivers with a differential receiver, including on-chip differential split thevenin termination.

DIFF_SSTL2_II, DIFF_SSTL18_II, DIFF_SSTL15

Differential SSTL 2.5V/1.8V Class II and differential SSTL 1.5V pairs complementary single-ended SSTL_II/SSTL type drivers with a differential receiver. For a bidirectional link, a 25 Ω series resistor must be connected to both transmitters.

DIFF_SSTL2_II_DCI, DIFF_SSTL18_II_DCI, DIFF_SSTL15_DCI

Differential SSTL 2.5V/1.8V Class II and differential SSTL 1.5V pairs complementary single-ended SSTL_II/SSTL type drivers with a differential receiver, including on-chip differential termination. DCI can be used for unidirectional and bidirectional links.

SSTL2_II_T_DCI, SSTL18_II_T_DCI, SSTL15_T_DCI

SSTL2_II_T_DCI, SSTL18_II_T_DCI, and SSTL15_T_DCI provide on-chip split thevenin termination powered from V_{CCO} that creates an equivalent termination voltage of $V_{CCO}/2$ when these standards are 3-stated. When not 3-stated, these two standards do not have

parallel termination but when invoked they have an internal series resistor (25 Ω at 2.5V and 20 Ω at 1.8V.)

SSTL2 Class I (2.5V)

Figure 1-56 shows a sample circuit illustrating a valid termination technique for SSTL2 Class I.

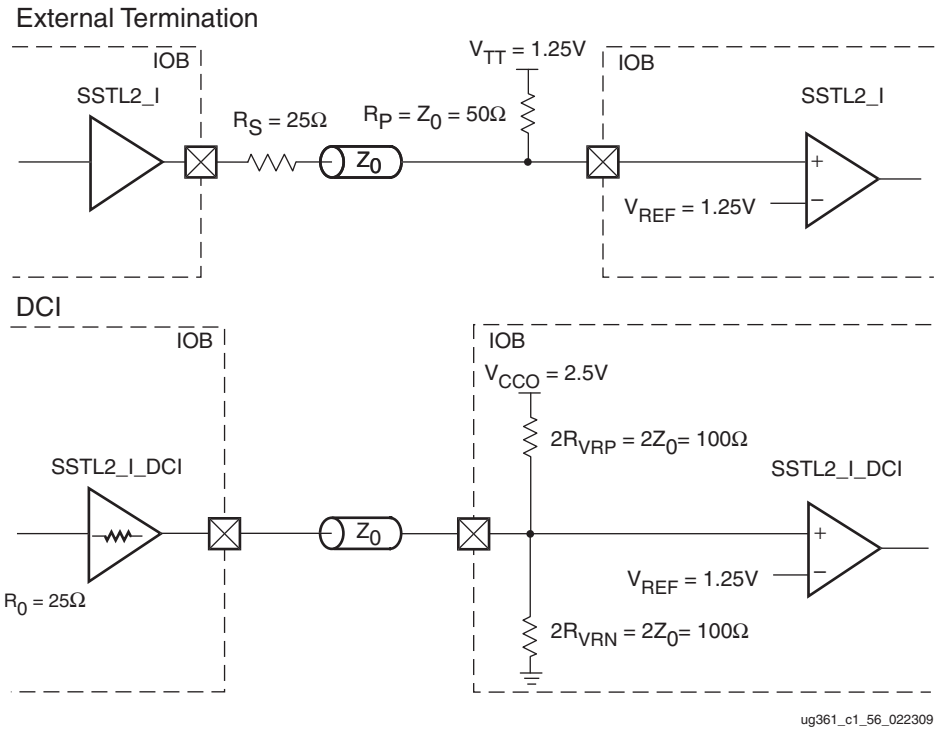


Figure 1-56: **SSTL2 Class I Termination**

Table 1-20 lists the SSTL2 DC voltage specifications for Class I.

Table 1-20: **SSTL2 DC Voltage Specifications Class I**

Parameter	Min	Typ	Max
V_{CCO}	2.3	2.5	2.625
$V_{REF} = 0.5 \times V_{CCO}$	1.13	1.25	1.38
$V_{TT} = V_{REF} + N$(1)	1.09	1.25	1.42
$V_{IH} \geq V_{REF} + 0.15$	1.28	1.4	$V_{CCO} + 0.3$(2)
$V_{IL} \leq V_{REF} - 0.15$	–0.3(3)	1.1	1.23
$V_{OH} \geq V_{REF} + 0.61$	1.74	1.84	1.94
$V_{OL} \leq V_{REF} - 0.61$(4)	0.56	0.66	0.76
I_{OH} at V_{OH} (mA)	–8.1	–	–

Table 1-20: **SSTL2 DC Voltage Specifications Class I *(Cont'd)***

Parameter	Min	Typ	Max
I_{OL} at V_{OL} (mA)	8.1	–	–

Notes:

1. N must be greater than or equal to –0.04 and less than or equal to 0.04.
2. V_{IH} maximum is V_{CCO} +0.3.
3. V_{IL} minimum does not conform to the formula.
4. Because SSTL2_I_DCI uses a controlled-impedance driver, V_{OH} and V_{OL} are different.

Differential SSTL2 Class I (2.5V)

Figure 1-57 shows a sample circuit illustrating a valid termination technique for differential SSTL2 Class I (2.5V) with unidirectional termination.

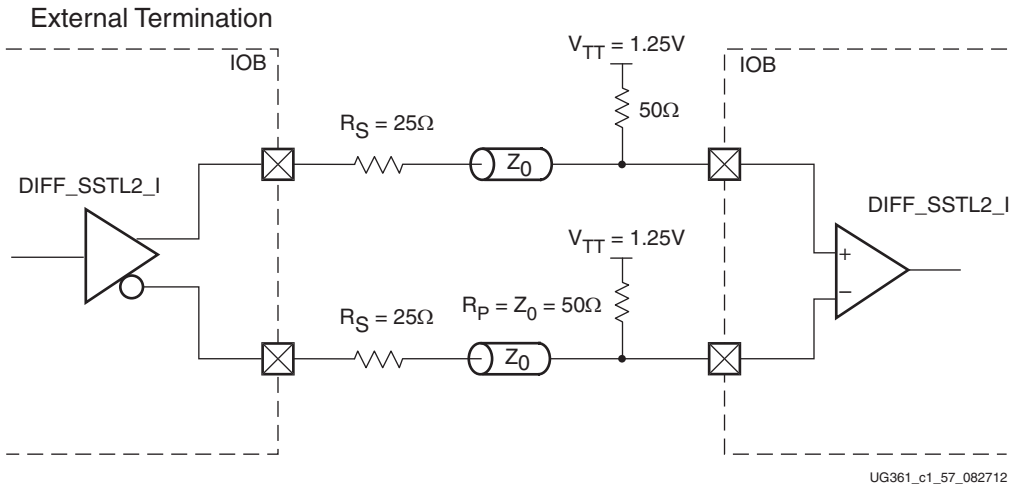


Figure 1-57: **Differential SSTL2 Class I Unidirectional Termination**

Figure 1-58 shows a sample circuit illustrating a valid termination technique for differential SSTL2 Class I (2.5V) with unidirectional DCI termination.

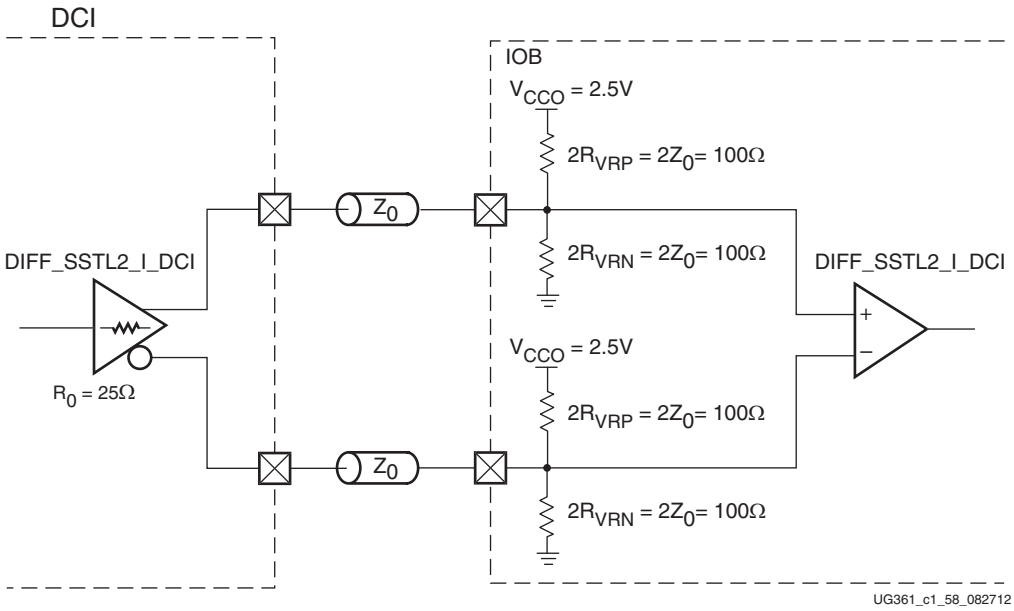


Figure 1-58: **Differential SSTL2 (2.5V) Class I Unidirectional DCI Termination**

Table 1-21 lists the differential SSTL2 Class I DC voltage specifications.

Table 1-21: **Differential SSTL2 Class I DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	2.3	2.5	2.625
Input Parameters			
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)(1)	–0.30	–	V_{CCO} + 0.30
V_{ID} (DC)(2)	0.3	–	V_{CCO} + 0.60
V_{ID} (AC)	0.62	–	V_{CCO} + 0.60
V_{IX} (AC)(3)	0.95	–	1.55
Output Parameters			
V_{OX} (AC)(4)	1.0	–	1.5

Notes:

1. V_{IN} (DC) specifies the allowable DC excursion of each differential input.
2. V_{ID} (DC) specifies the input differential voltage required for switching.
3. V_{IX} (AC) indicates the voltage where the differential input signals must cross.
4. V_{OX} (AC) indicates the voltage where the differential output signals must cross.

SSTL2 Class II (2.5V)

Figure 1-59 shows a sample circuit illustrating a valid unidirectional termination technique for SSTL2 Class II.

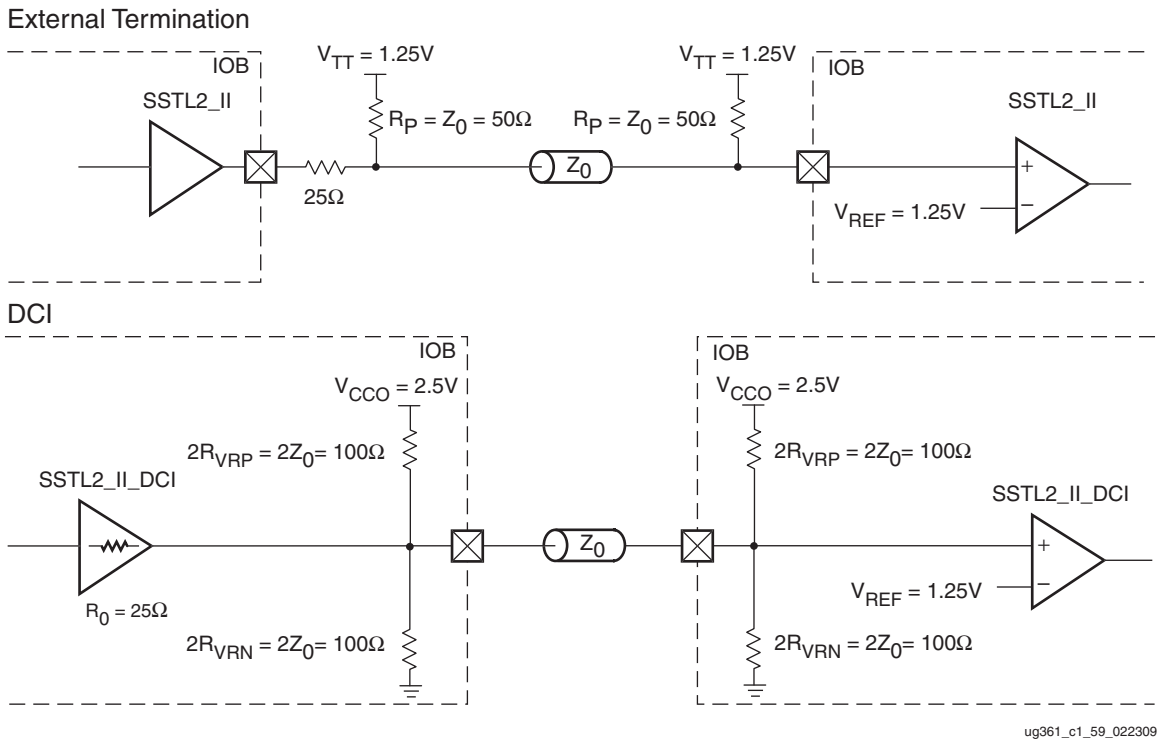


Figure 1-59: **SSTL2 Class II with Unidirectional Termination**

Figure 1-60 shows a sample circuit illustrating a valid bidirectional termination technique for SSTL2 Class II.

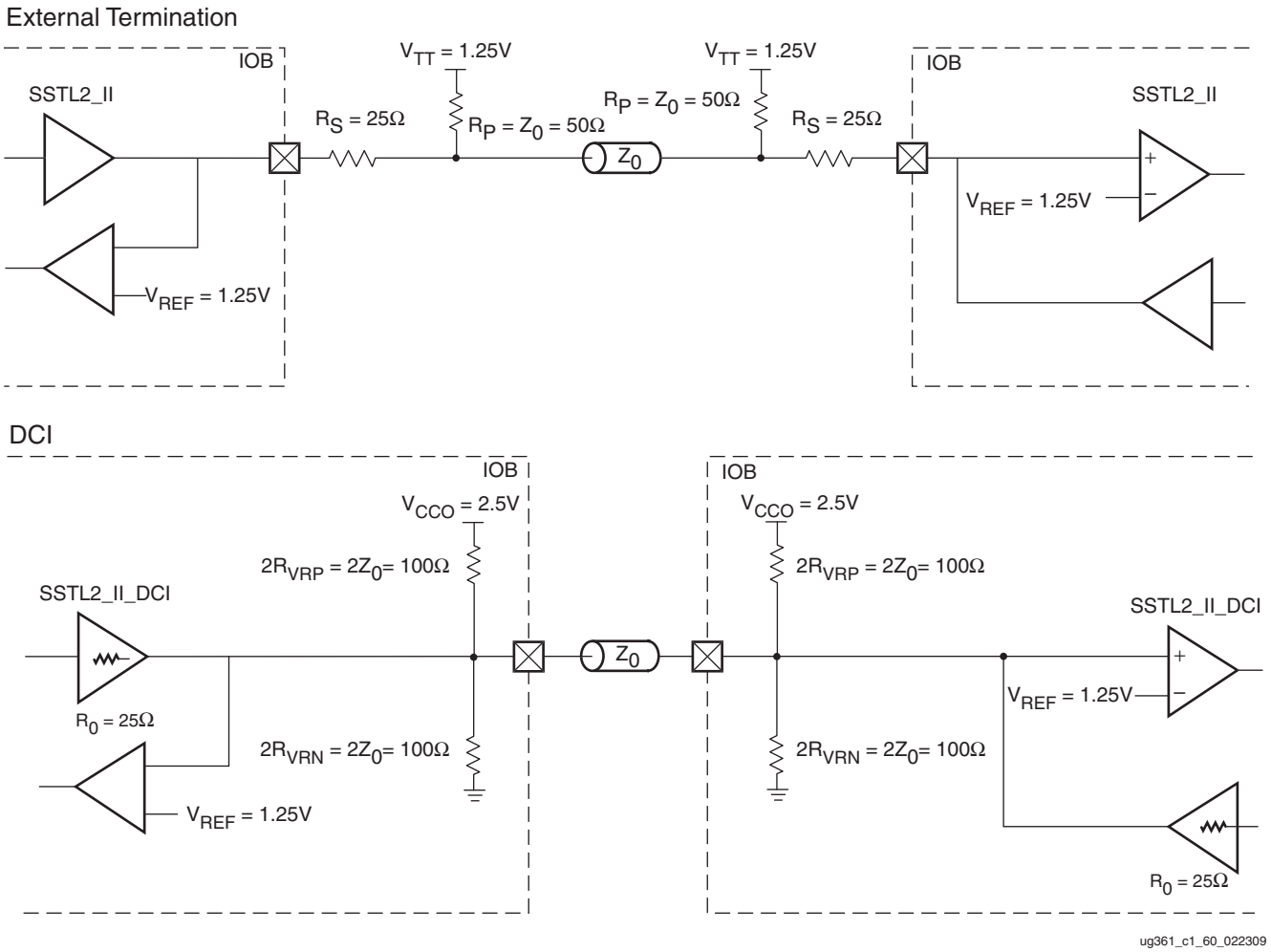


Figure 1-60: **SSTL2 Class II with Bidirectional Termination**

Table 1-22 lists the SSTL2 DC voltage specifications for Class II.

Table 1-22: **SSTL2 DC Voltage Specifications Class II**

Parameter	Min	Typ	Max
V_{CCO}	2.3	2.5	2.625
V_{REF}	1.13	1.25	1.38
$V_{TT} = V_{REF} + N^{(1)}$	1.09	1.25	1.42
$V_{IH} \geq V_{REF} + 0.15$	1.28	1.40	$V_{CCO} + 0.3^{(2)}$
$V_{IL} \leq V_{REF} - 0.15$	$-0.3^{(3)}$	1.1	1.27
$V_{OH} \geq V_{REF} + 0.81$	1.93	2.03	2.13
$V_{OL} \leq V_{REF} - 0.81^{(4)}$	0.36	0.46	0.55
I_{OH} at V_{OH} (mA)	–16.2	–	–

Table 1-22: **SSTL2 DC Voltage Specifications Class II *(Cont'd)***

Parameter	Min	Typ	Max
I_{OL} at V_{OL} (mA)	16.2	–	–

Notes:

1. N must be greater than or equal to –0.04 and less than or equal to 0.04.
2. V_{IH} maximum is V_{CCO} +0.3.
3. V_{IL} minimum does not conform to the formula.
4. Because SSTL2_I_DCI uses a controlled-impedance driver, V_{OH} and V_{OL} are different.

Differential SSTL2 Class II (2.5V)

Figure 1-61 shows a sample circuit illustrating a valid termination technique for differential SSTL2 Class II (2.5V) with unidirectional termination.

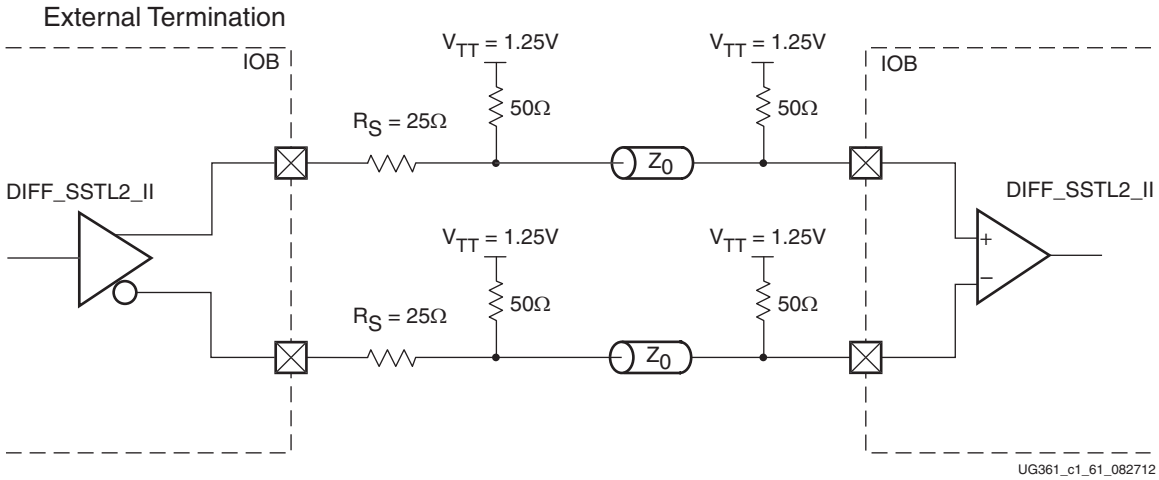


Figure 1-61: **Differential SSTL2 Class II Unidirectional Termination**

Figure 1-62 shows a sample circuit illustrating a valid termination technique for differential SSTL2 Class II (2.5V) with unidirectional DCI termination.

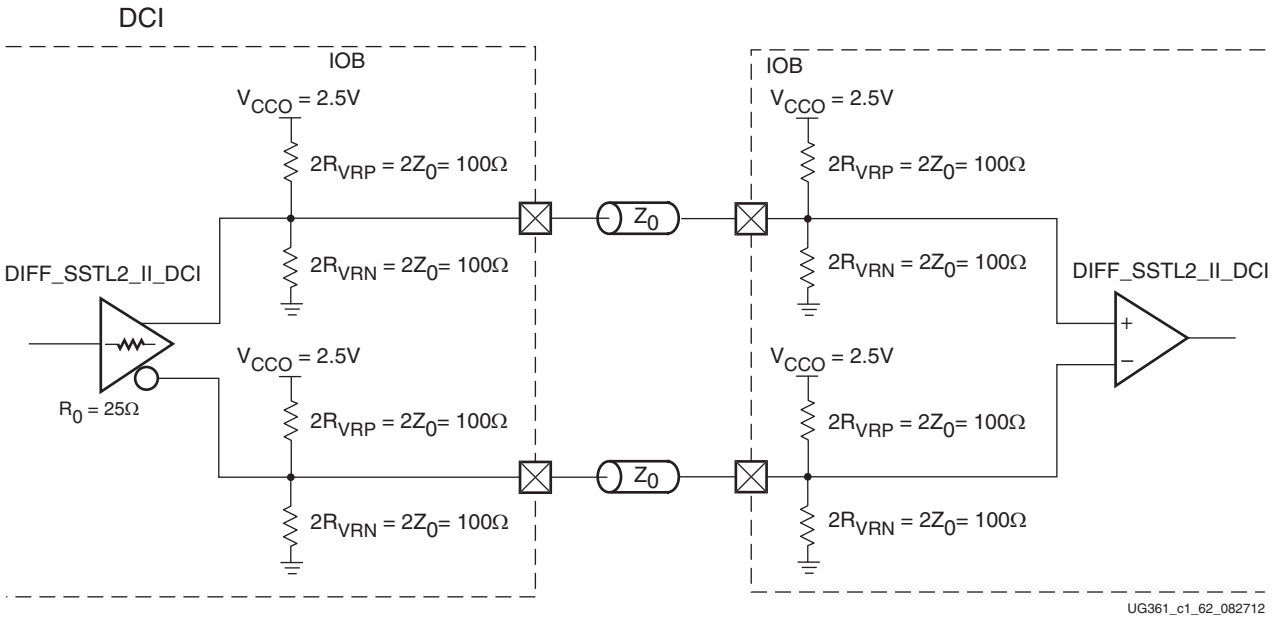


Figure 1-62: **Differential SSTL2 (2.5V) Class II Unidirectional DCI Termination**

Figure 1-63 shows a sample circuit illustrating a valid termination technique for differential SSTL2 Class II (2.5V) with bidirectional termination.

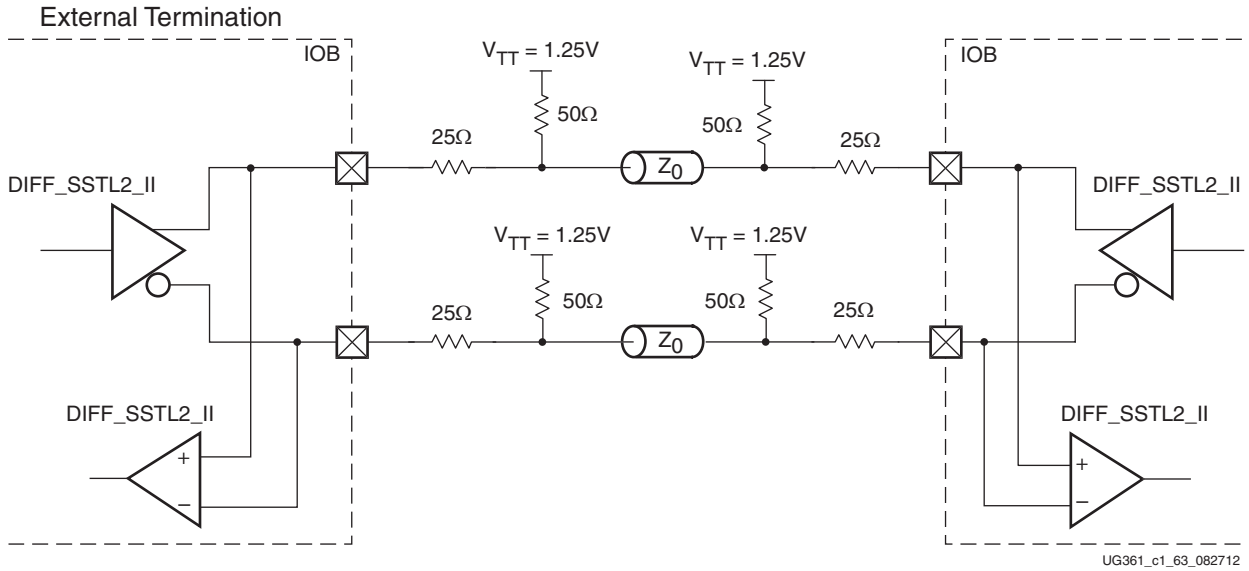


Figure 1-63: **Differential SSTL2 (2.5V) Class II with Bidirectional Termination**

Figure 1-64 shows a sample circuit illustrating a valid termination technique for differential SSTL2 Class II (2.5V) with bidirectional DCI termination.

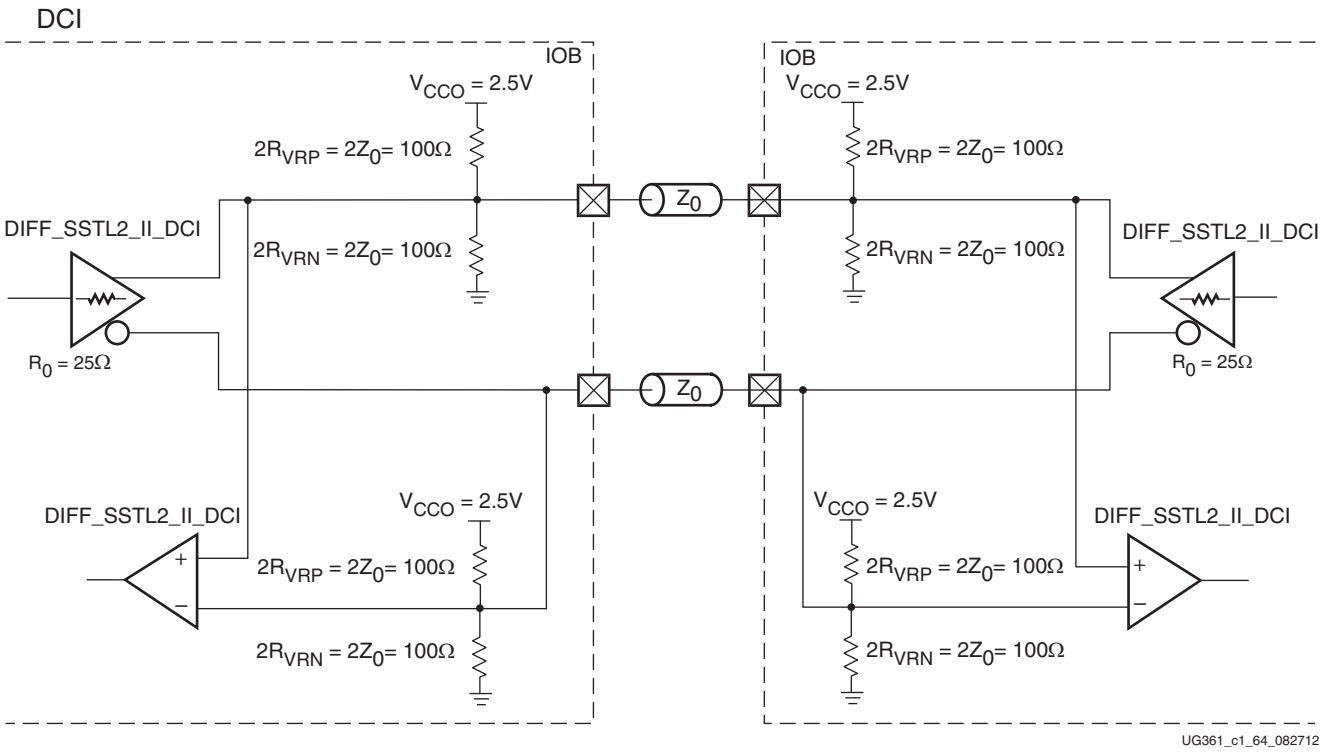


Figure 1-64: **Differential SSTL2 (2.5V) Class II with DCI Bidirectional Termination**

Table 1-23 lists the differential SSTL2 Class II DC voltage specifications.

Table 1-23: **Differential SSTL2 Class II DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	2.3	2.5	2.625
Input Parameters			
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)(1)	–0.30	–	V_{CCO} + 0.30
V_{ID} (DC)(2)	0.3	–	V_{CCO} + 0.60
V_{ID} (AC)	0.62	–	V_{CCO} + 0.60
V_{IX} (AC)(3)	0.95	–	1.55
Output Parameters			
V_{OX} (AC)(4)	1.0	–	1.5

Notes:

1. V_{IN} (DC) specifies the allowable DC excursion of each differential input.
2. V_{ID} (DC) specifies the input differential voltage required for switching.
3. V_{IX} (AC) indicates the voltage where the differential input signals must cross.
4. V_{OX} (AC) indicates the voltage where the differential output signals must cross.

SSTL2_II_T_DCI (2.5V) Split-Thevenin Termination

Figure 1-65 shows a sample circuit illustrating a valid termination technique for SSTL2_II_T_DCI (2.5V) with on-chip split-thevenin termination. In this bidirectional I/O standard, when 3-stated, the termination is invoked on the receiver and not on the driver.

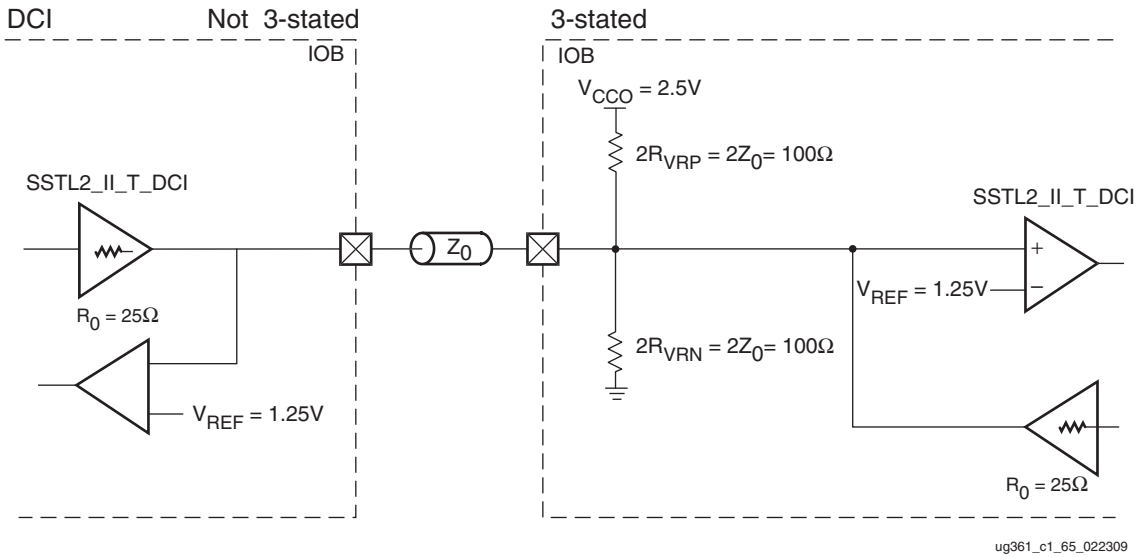


Figure 1-65: **SSTL2_II_T_DCI (2.5V) Split-Thevenin Termination**

SSTL18 Class I (1.8V)

Figure 1-66 shows a sample circuit illustrating a valid termination technique for SSTL Class I (1.8V).

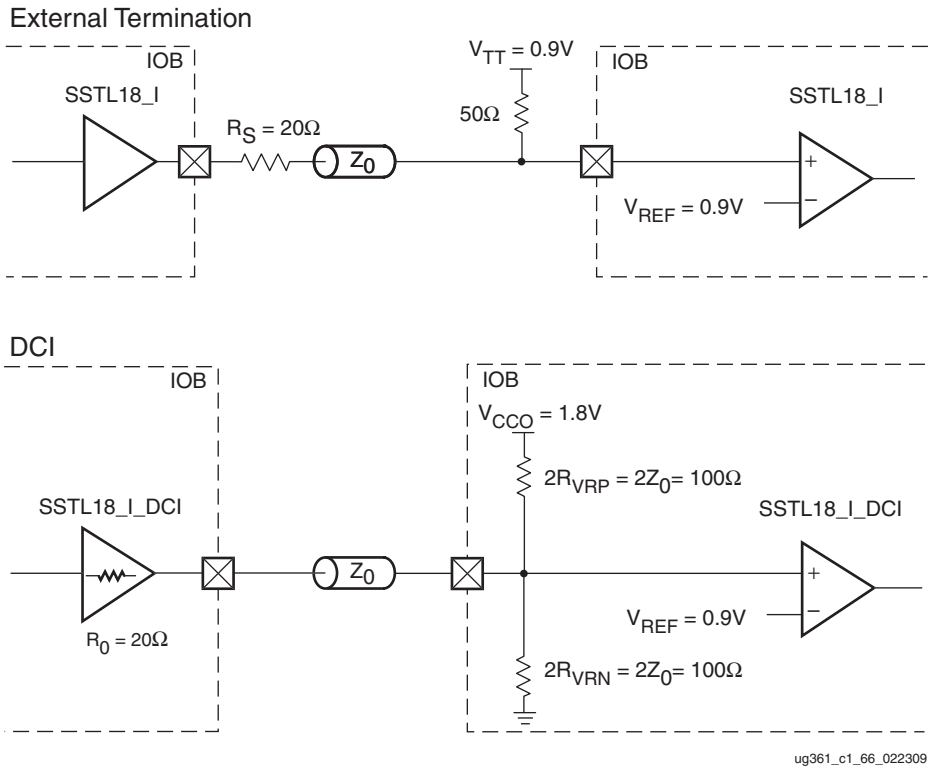


Figure 1-66: **SSTL18 (1.8V) Class I Termination**

Differential SSTL Class I (1.8V)

Figure 1-67 shows a sample circuit illustrating a valid termination technique for differential SSTL Class I (1.8V) with unidirectional termination.

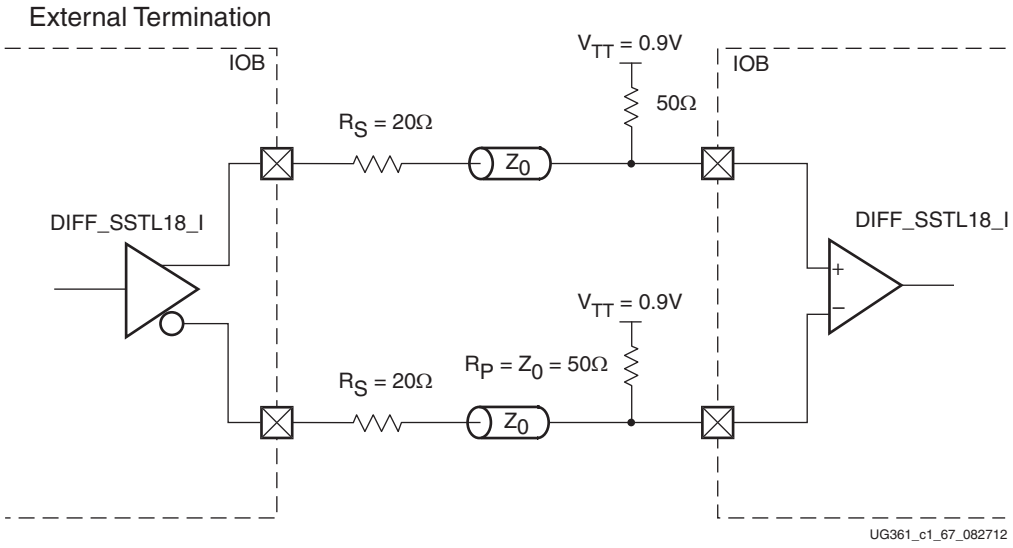


Figure 1-67: **Differential SSTL (1.8V) Class I Unidirectional Termination**

Figure 1-68 shows a sample circuit illustrating a valid termination technique for differential SSTL Class I (1.8V) with unidirectional DCI termination.

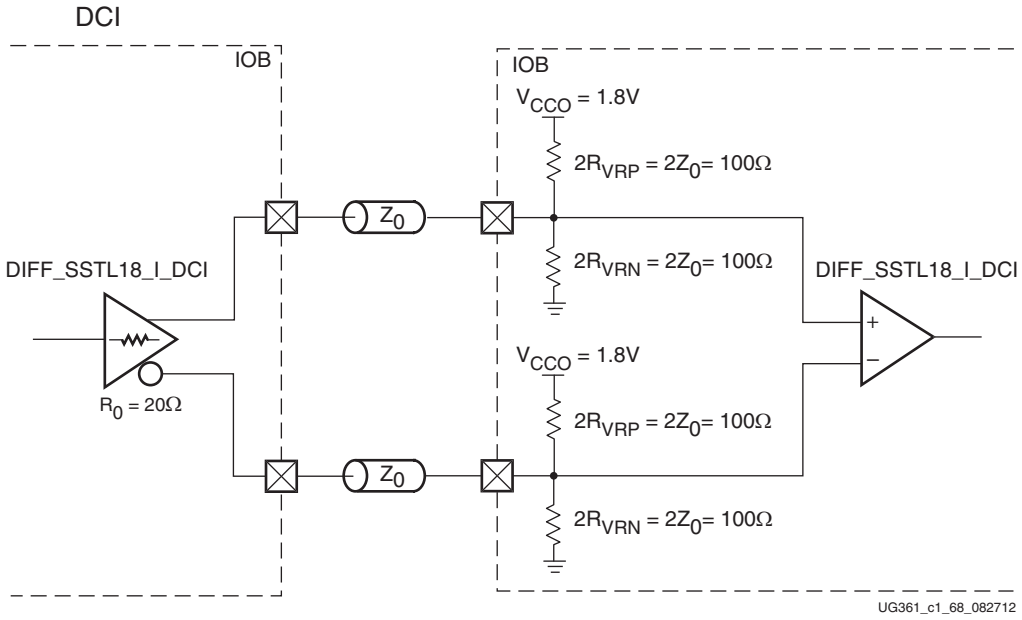


Figure 1-68: **Differential SSTL (1.8V) Class I Unidirectional DCI Termination**

Table 1-24 lists the differential SSTL (1.8V) Class I DC voltage specifications.

Table 1-24: **Differential SSTL (1.8V) Class I and Class II DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
Input Parameters			
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)(1)	–0.30	–	V_{CCO} + 0.30
V_{ID} (DC)(3)	0.25	–	V_{CCO} + 0.60
V_{ID} (AC)	0.50	–	V_{CCO} + 0.60
V_{IX} (AC)(4)	0.675	–	1.125
Output Parameters			
V_{OX} (AC)(5)	0.725	–	1.075

Notes:

1. V_{IN} (DC) specifies the allowable DC excursion of each differential input.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."
3. V_{ID} (DC) specifies the input differential voltage required for switching.
4. V_{IX} (AC) indicates the voltage where the differential input signals must cross.
5. V_{OX} (AC) indicates the voltage where the differential output signals must cross.

SSTL18 Class II (1.8V)

Figure 1-69 shows a sample circuit illustrating a valid unidirectional termination technique for SSTL Class II (1.8V).

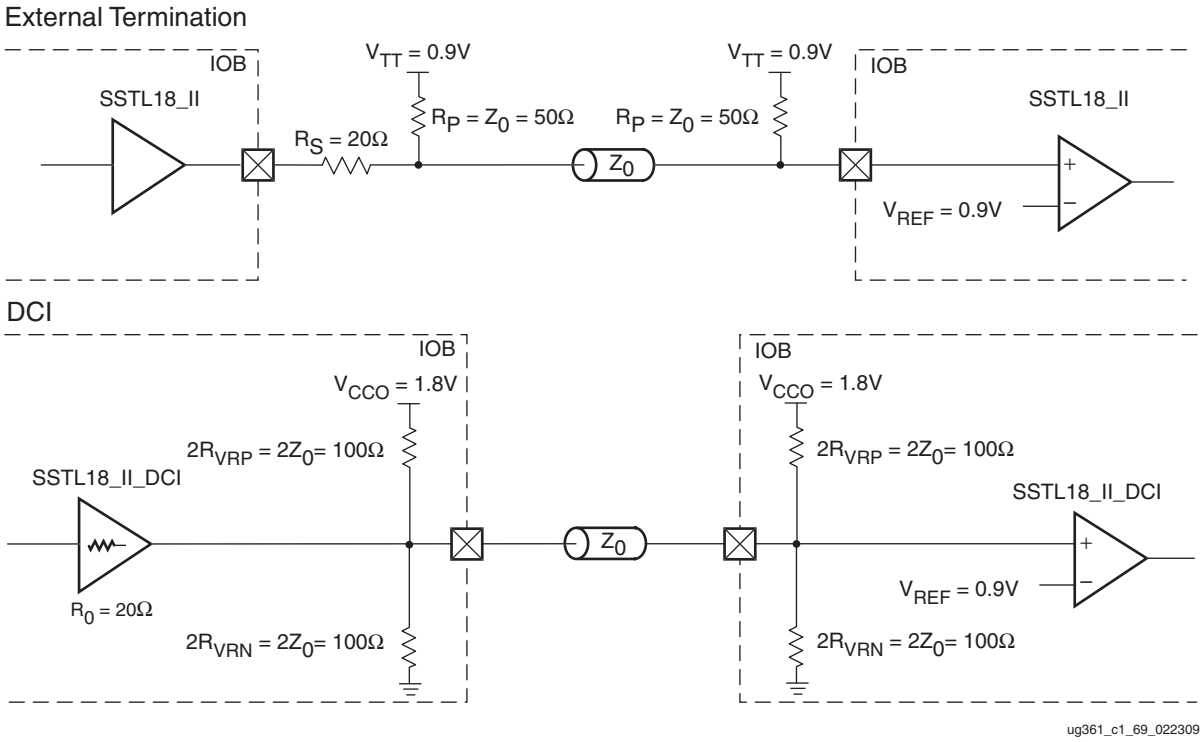


Figure 1-69: **SSTL18 (1.8V) Class II Unidirectional Termination**

Figure 1-70 shows a sample circuit illustrating a valid bidirectional termination technique for SSTL (1.8V) Class II.

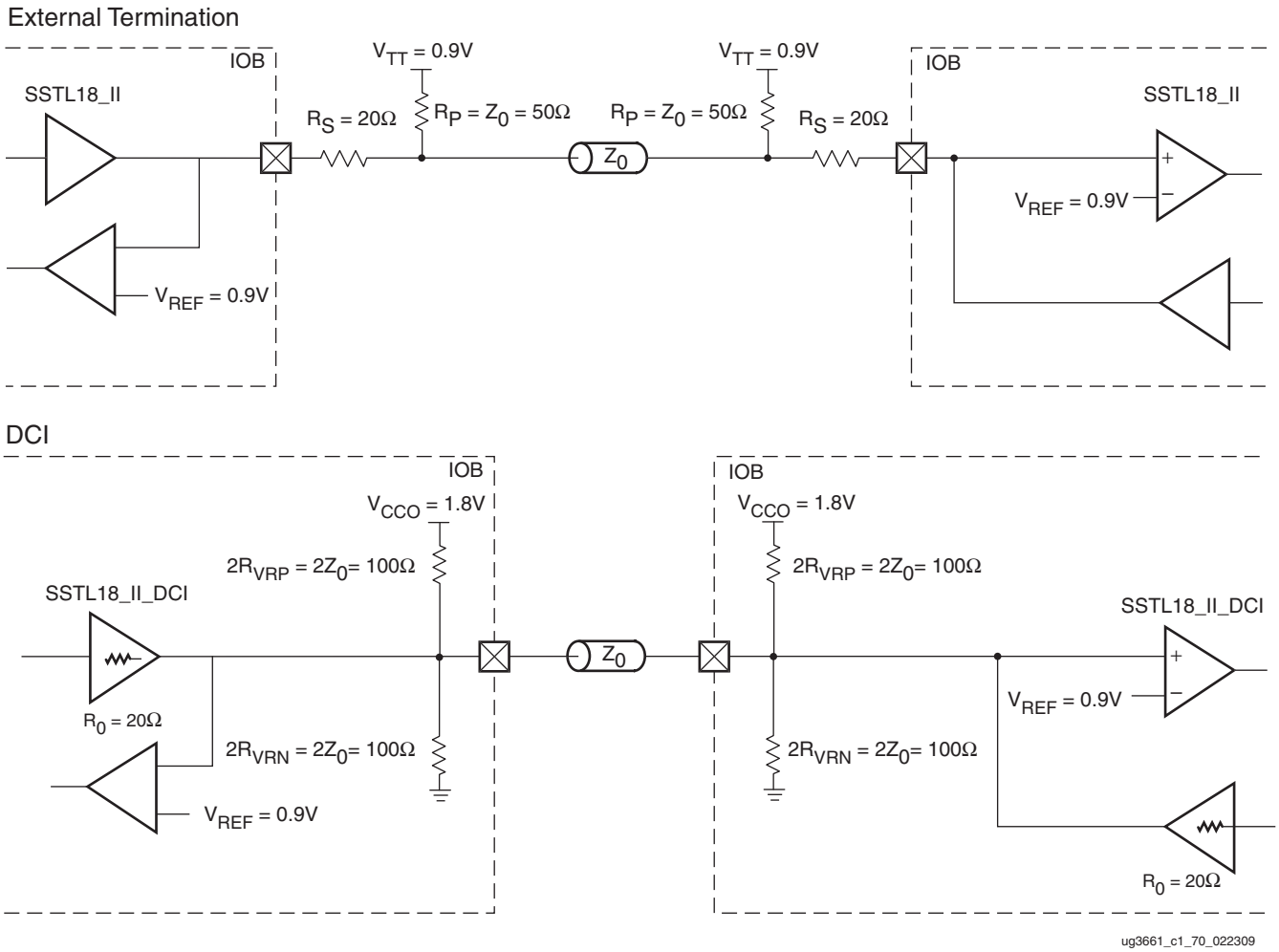


Figure 1-70: **SSTL (1.8V) Class II Termination**

Table 1-25 and Table 1-26 lists the SSTL (1.8V) DC voltage specifications for Class I and Class II respectively.

Table 1-25: **SSTL (1.8V) DC Voltage Specifications Class I**

Parameter	Class I		
	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{REF}	0.833	0.9	0.969
$V_{TT} = V_{REF} + N^{(1)}$	0.793	0.9	1.009
$V_{IH} \geq V_{REF} + 0.125$	0.958	–	$V_{CCO} + 0.3^{(2)}$
$V_{IL} \leq V_{REF} - 0.125$	$-0.3^{(3)}$	–	0.844
$V_{OH} \geq V_{TT} + 0.47^{(4)}$	1.263	–	–
$V_{OL} \leq V_{TT} - 0.47^{(4)}$	–	–	0.539
I_{OH} at V_{OH} (mA)	–6.7	–	–
I_{OL} at V_{OL} (mA)	6.7	–	–

Notes:

1. N must be greater than or equal to –0.04 and less than or equal to 0.04.
2. V_{IH} maximum is V_{CCO} +0.3.
3. V_{IL} minimum does not conform to the formula.
4. Because SSTL_I_DCI uses a controlled-impedance driver, V_{OH} and V_{OL} are different.

Table 1-26: **SSTL (1.8V) DC Voltage Specifications Class II**

Parameter	Class II		
	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
V_{REF}	0.833	0.9	0.969
$V_{TT} = V_{REF} + N^{(1)}$	0.793	0.9	1.009
$V_{IH} \geq V_{REF} + 0.125$	0.958	–	$V_{CCO} + 0.3^{(2)}$
$V_{IL} \leq V_{REF} - 0.125$	$-0.3^{(3)}$	–	0.844
$V_{OH} \geq V_{TT} + 0.603^{(4)}$	1.396	–	–
$V_{OL} \leq V_{TT} - 0.603^{(4)}$	–	–	0.406
I_{OH} at V_{OH} (mA)	–13.4	–	–
I_{OL} at V_{OL} (mA)	13.4	–	–

Notes:

1. N must be greater than or equal to –0.04 and less than or equal to 0.04.
2. V_{IH} maximum is V_{CCO} +0.3.
3. V_{IL} minimum does not conform to the formula.
4. Because SSTL_I_DCI uses a controlled-impedance driver, V_{OH} and V_{OL} are different.

Differential SSTL Class II (1.8V)

Figure 1-71 shows a sample circuit illustrating a valid termination technique for differential SSTL Class II (1.8V) with unidirectional termination.

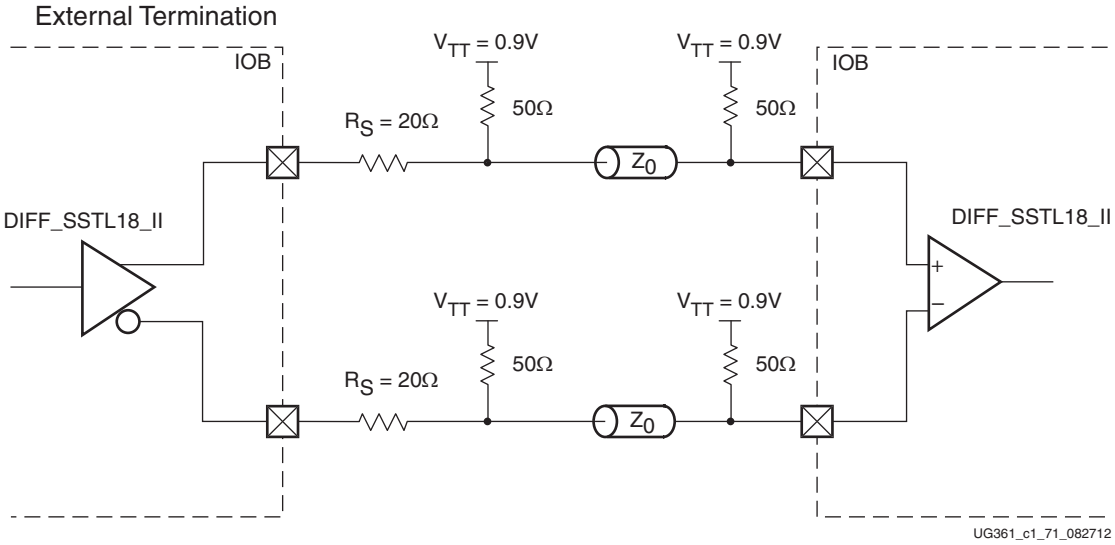


Figure 1-71: **Differential SSTL (1.8V) Class II Unidirectional Termination**

Figure 1-72 shows a sample circuit illustrating a valid termination technique for differential SSTL Class II (1.8V) with unidirectional DCI termination.

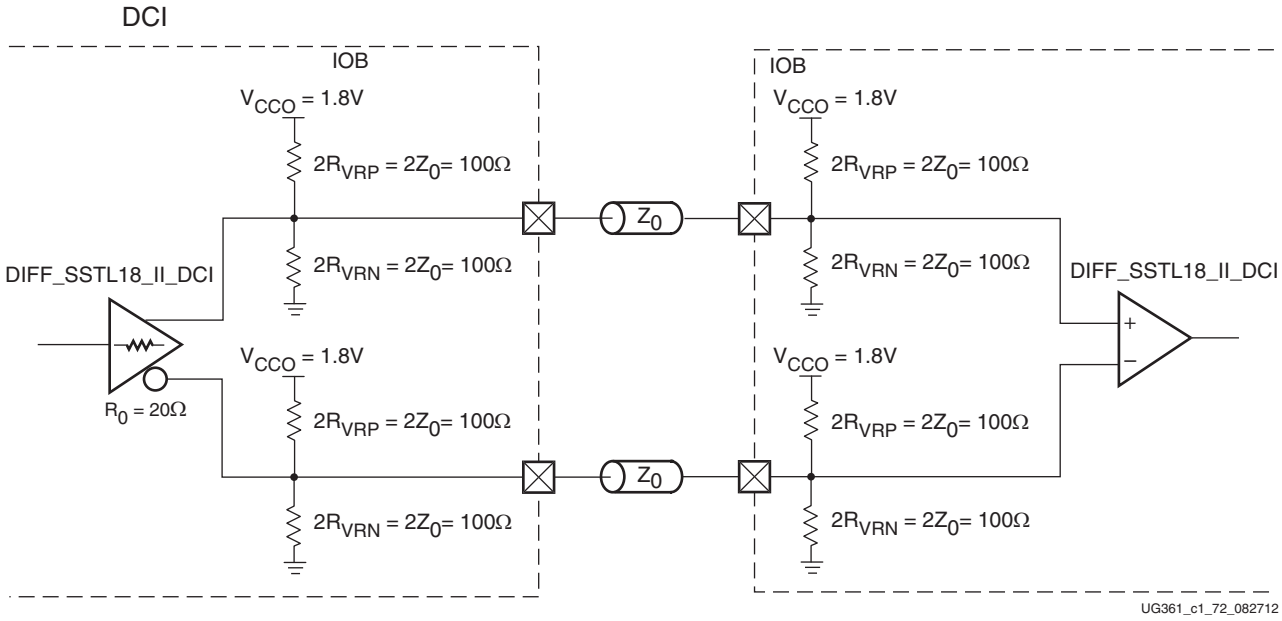


Figure 1-72: **Differential SSTL (1.8V) Class II Unidirectional DCI Termination**

Figure 1-73 shows a sample circuit illustrating a valid termination technique for differential SSTL Class II (1.8V) with bidirectional termination.

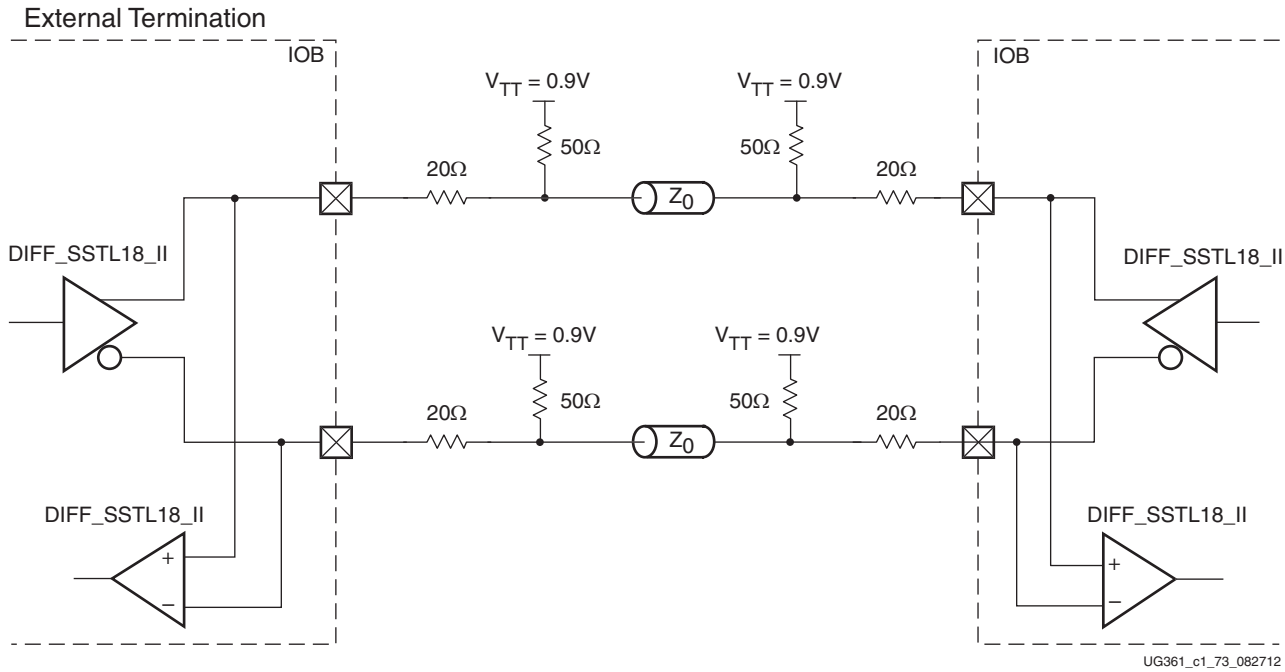


Figure 1-73: **Differential SSTL (1.8V) Class II with Bidirectional Termination**

Figure 1-74 shows a sample circuit illustrating a valid termination technique for differential SSTL Class II (1.8V) with bidirectional DCI termination.

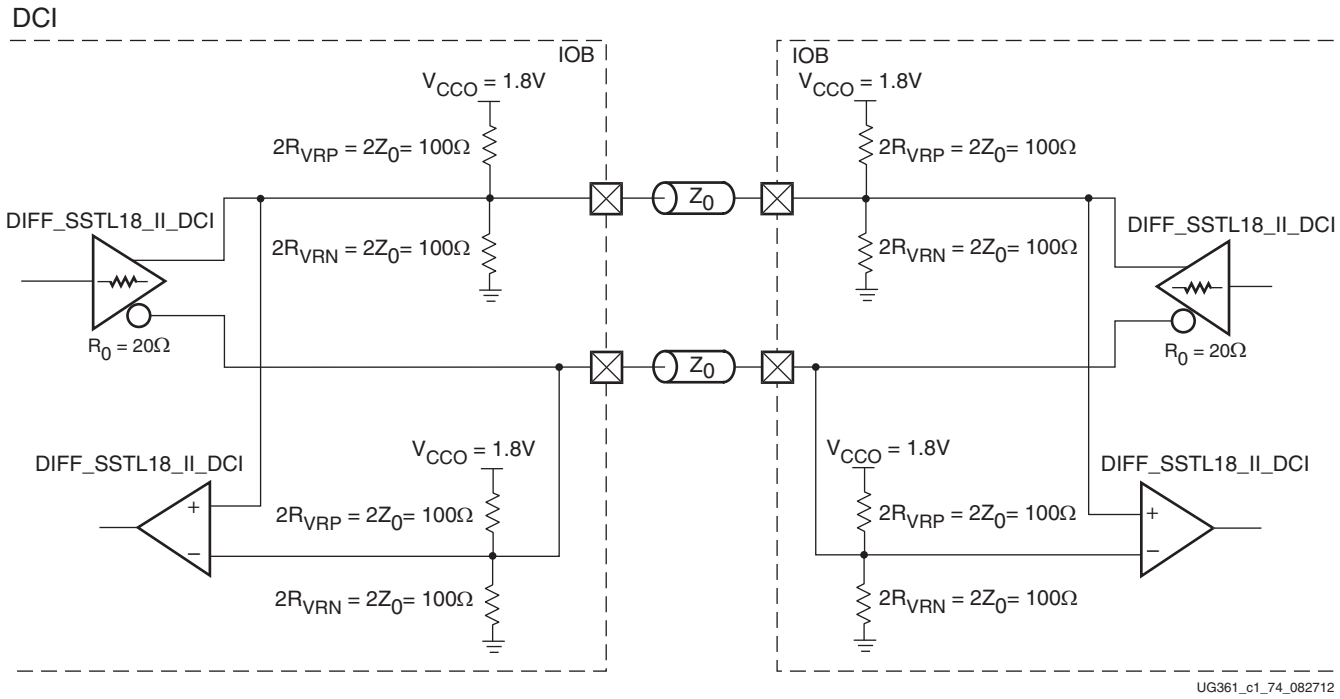


Figure 1-74: **Differential SSTL (1.8V) Class II with DCI Bidirectional Termination**

Table 1-27 lists the differential SSTL (1.8V) Class II DC voltage specifications.

Table 1-27: **Differential SSTL (1.8V) Class II DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.7	1.8	1.9
Input Parameters			
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)(1)	–0.30	–	V_{CCO} + 0.30
V_{ID} (DC)(3)	0.25	–	V_{CCO} + 0.60
V_{ID} (AC)	0.50	–	V_{CCO} + 0.60
V_{IX} (AC)(4)	0.675	–	1.125
Output Parameters			
V_{OX} (AC)(5)	0.725	–	1.075

Notes:

1. V_{IN} (DC) specifies the allowable DC excursion of each differential input.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."
3. V_{ID} (DC) specifies the input differential voltage required for switching.
4. V_{IX} (AC) indicates the voltage where the differential input signals must cross.
5. V_{OX} (AC) indicates the voltage where the differential output signals must cross.

SSTL18_II_T_DCI (1.8V) Split-Thevenin Termination

Figure 1-75 shows a sample circuit illustrating a valid termination technique for SSTL18_II_T_DCI (1.8V) with on-chip split-thevenin termination. In this bidirectional I/O standard, when 3-stated, the termination is invoked on the receiver and not on the driver.

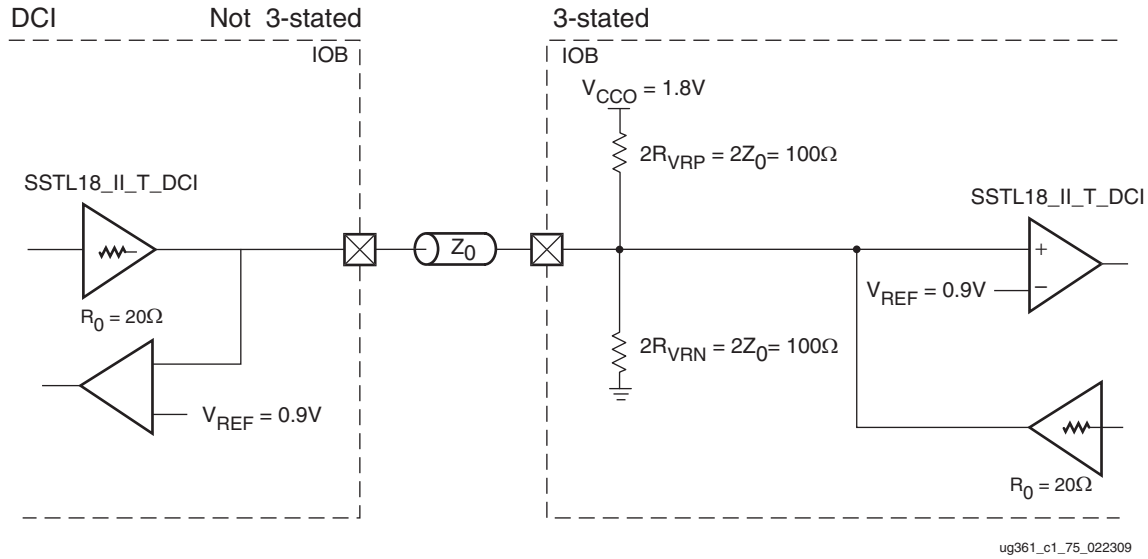


Figure 1-75: **SSTL18_II_T_DCI (1.8V) Split-Thevenin Termination**

SSTL15 (1.5V)

Figure 1-76 shows a sample circuit illustrating a valid unidirectional termination technique for SSTL (1.5V).

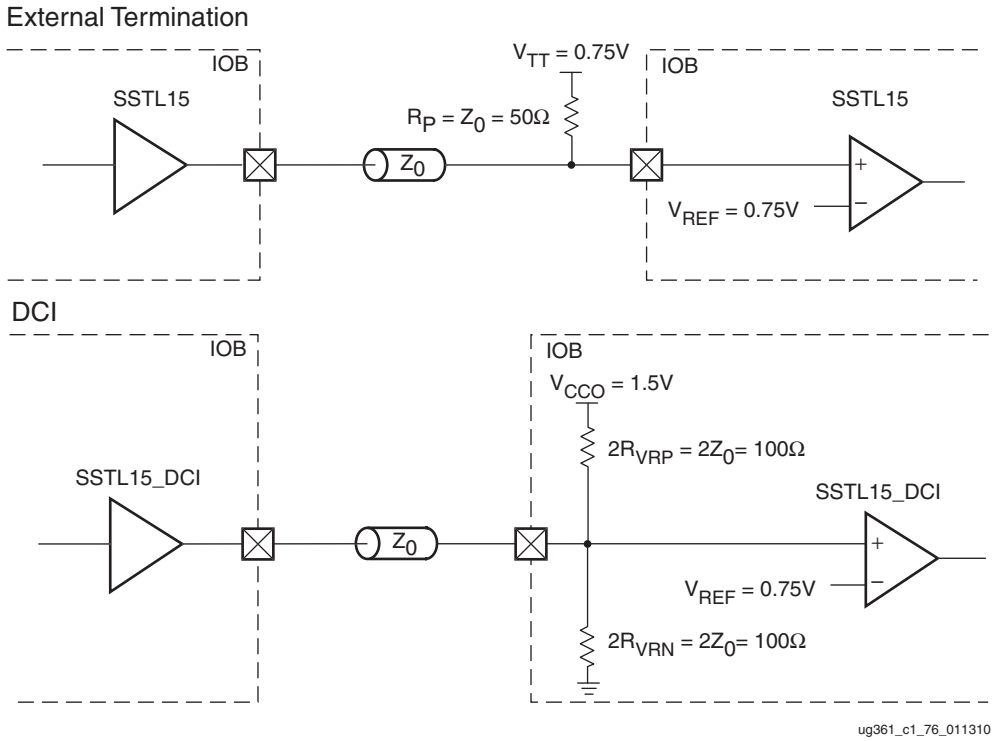


Figure 1-76: **SSTL (1.5V) Unidirectional Termination**

Figure 1-77 shows a sample circuit illustrating a valid bidirectional termination technique for SSTL (1.5V).

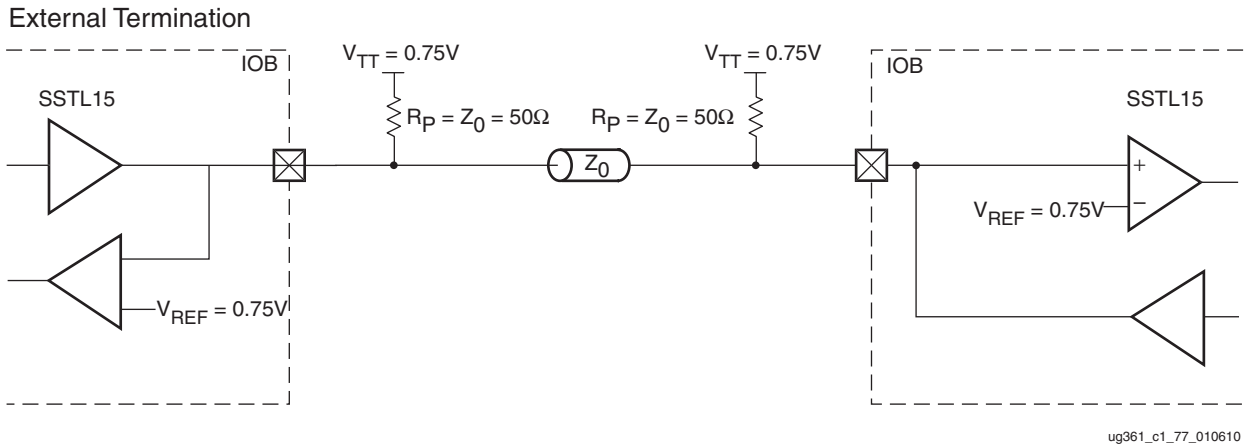


Figure 1-77: **SSTL (1.5V) Bidirectional Termination**

Table 1-28 lists the SSTL (1.5V) DC voltage specifications.

Table 1-28: **SSTL (1.5V) DC Voltage Specifications**

Parameter	Min	Typ	Max	Units
V_{CCO}	1.425	1.5	1.575	V
V_{REF}	V_{CCO} x 0.49	V_{CCO} x 0.5	V_{CCO} x 0.51	V
V_{TT}	V_{CCO} x 0.5			V
V_{IH} (DC)	V_{REF} + 0.1	–	–	V
V_{IL} (DC)	–	–	V_{REF} – 0.1	V
V_{OH}	V_{TT} + (V_{CCO} x 0.1)	–	V_{CCO} x 0.8	V
V_{OL}	V_{CCO} x 0.2	–	V_{TT} – (V_{CCO} x 0.1)	V
I_{OH}(1)	–	14.3	–	mA
I_{OL}(1)	–	14.3	–	mA

Notes:

1. I_{OH} and I_{OL} measured for a pad at 0.5 x V_{CCO}. Minimum level for a 50Ω driver.

Differential SSTL (1.5V)

Figure 1-78 shows a sample circuit illustrating a valid termination technique for differential SSTL (1.5V) with unidirectional termination.

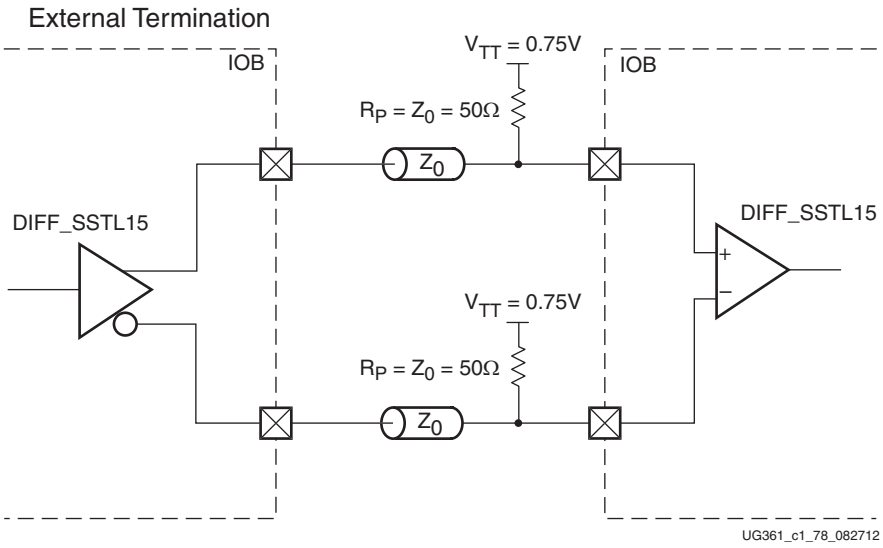


Figure 1-78: **Differential SSTL (1.5V) Unidirectional Termination**

Figure 1-79 shows a sample circuit illustrating a valid termination technique for differential SSTL (1.5V) with unidirectional DCI termination.

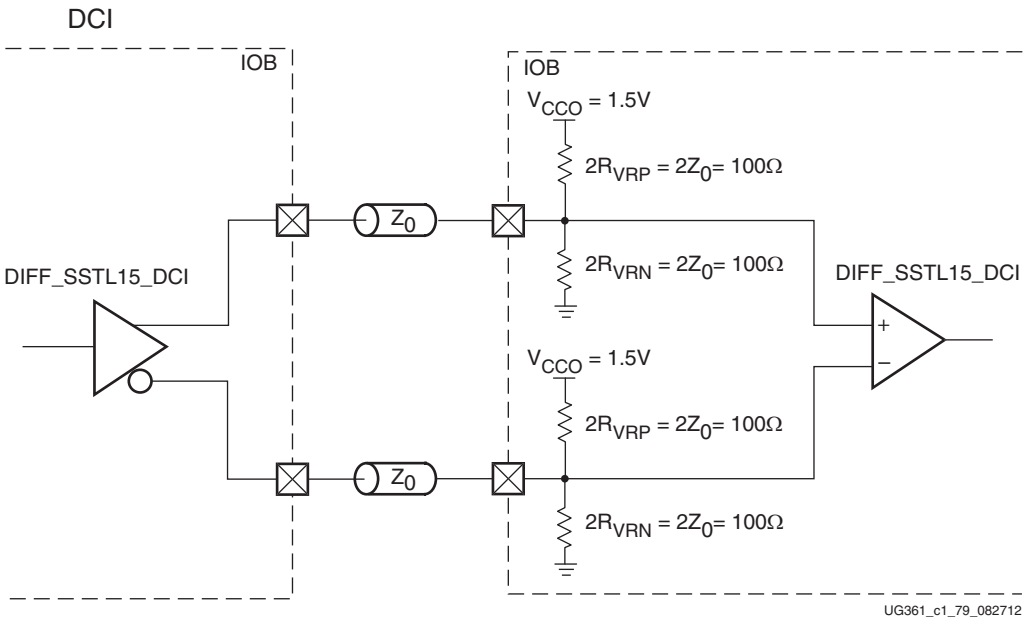


Figure 1-79: **Differential SSTL (1.5V) Unidirectional DCI Termination**

Figure 1-80 shows a sample circuit illustrating a valid termination technique for differential SSTL (1.5V) with bidirectional termination.

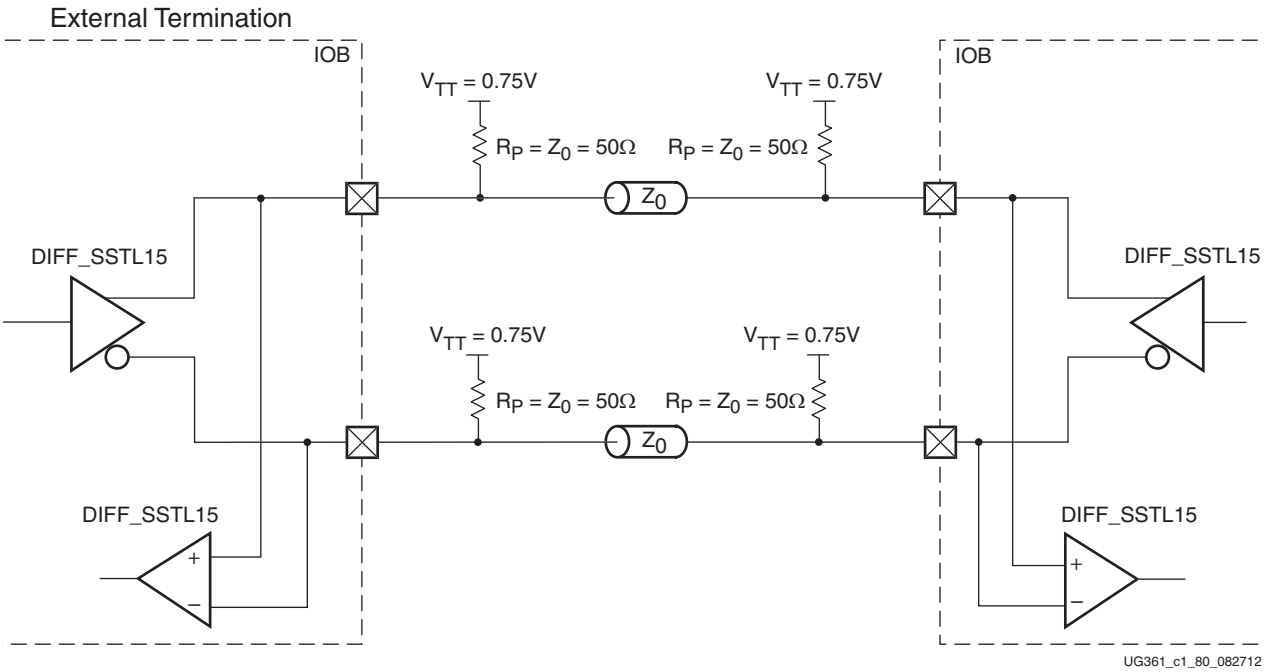


Figure 1-80: **Differential SSTL (1.5V) with Bidirectional Termination**

Table 1-29 lists the differential SSTL (1.5V) DC voltage specifications.

Table 1-29: **Differential SSTL (1.5V) DC Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	1.425	1.5	1.575
V_{TT}	–	$V_{CCO} \times 0.5$	–
V_{IN} (DC)(1)	–0.3	–	V_{CCO} + 0.3
V_{ID} (DC)(3)	0.2	–	V_{CCO} + 0.4
V_{ID} (AC)	0.4	–	V_{CCO} + 0.4
V_{IX} (AC)(4)	V_{TT} – 0.175	–	V_{TT}+ 0.175
V_{OX} (AC)(5)	V_{TT} – 0.15	–	V_{TT} + 0.15

Notes:

1. V_{IN} (DC) specifies the allowable DC excursion of each differential input.
2. Per EIA/JESD8-6, "The value of V_{REF} is to be selected by the user to provide optimum noise margin in the use conditions specified by the user."
3. V_{ID} (DC) specifies the input differential voltage required for switching.
4. V_{IX} (AC) indicates the voltage where the differential input signals must cross.
5. V_{OX} (AC) indicates the voltage where the differential output signals must cross.

SSTL15_T_DCI (1.5V) Split-Thevenin Termination

Figure 1-81 shows a sample circuit illustrating a valid termination technique for SSTL15_T_DCI (1.5V) with on-chip split-thevenin termination. In this bidirectional I/O standard, when 3-stated, the termination is invoked on the receiver and not on the driver.

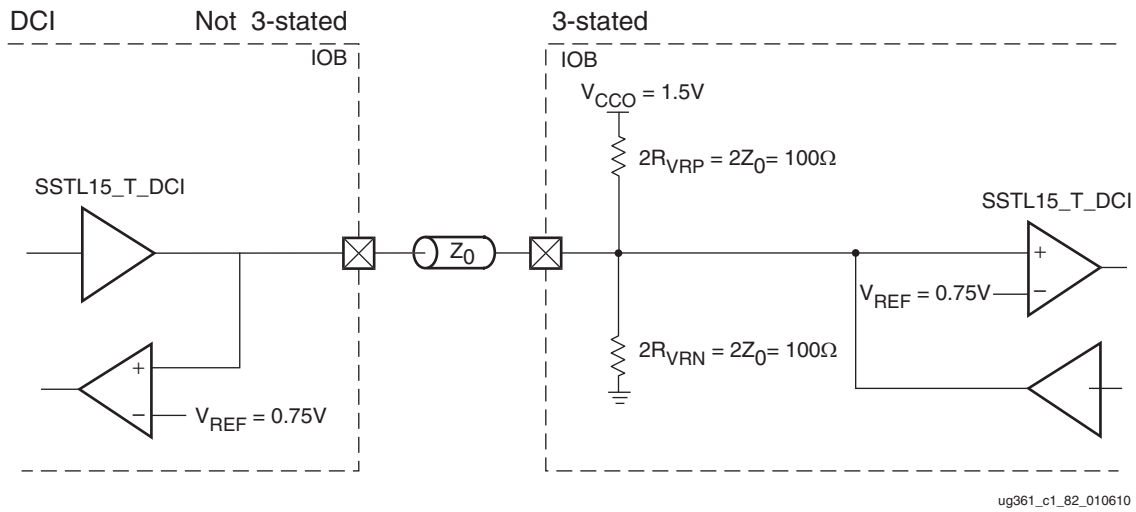


Figure 1-81: **SSTL15_T_DCI (1.5V) Split-Thevenin Termination**

Differential Termination: DIFF_TERM Attribute

Virtex-6 FPGA IOBs provide a 100Ω differential termination across the input differential receiver terminals. This attribute is used in conjunction with LVDS_25, LVDSEXT_25, HT_25, and RSDS_25.

The on-chip input differential termination in Virtex-6 devices provides major advantages over the external resistor by removing the stub at the receiver completely and therefore greatly improving signal integrity:

- Consumes less power than DCI termination
- Does not use VRP/VRN pins (DCI)

The V_{CCO} of the I/O bank must be connected to 2.5V ±5% to provide 100Ω of effective differential termination. DIFF_TERM is only available for inputs and can *only* be used with a bank voltage of V_{CCO} = 2.5V. The Differential Termination Attribute (DIFF_TERM) section outlines using this feature.

LVDS and Extended LVDS (Low Voltage Differential Signaling)

Low Voltage Differential Signaling (LVDS) is a very popular and powerful high-speed interface in many system applications. With the use of an LVDS current-mode driver in the IOBs, the need for external source termination in point-to-point applications is eliminated, and with the choice of an extended mode, Virtex-6 devices provide the most flexible solution for doing an LVDS design in an FPGA.

Extended LVDS provides a higher drive capability and voltage swing (350 - 750 mV), making it ideal for long-distance or cable LVDS links. The output AC characteristics of the LVDS extended mode driver are not within the EIA/TIA specifications. The LVDS extended mode driver is intended for situations requiring higher drive capabilities to produce an LVDS signal within the EIA/TIA specification at the receiver.

Transmitter Termination

The Virtex-6 FPGA LVDS transmitter does not require any external termination. Table 1-30 lists the allowed attributes corresponding to the Virtex-6 FPGA LVDS current-mode drivers. Virtex-6 FPGA LVDS current-mode drivers are a true current source and produce the proper (EIA/TIA compliant) LVDS signal.

Receiver Termination

Figure 1-82 is an example of differential termination for an LVDS receiver on a board with 50Ω transmission lines.

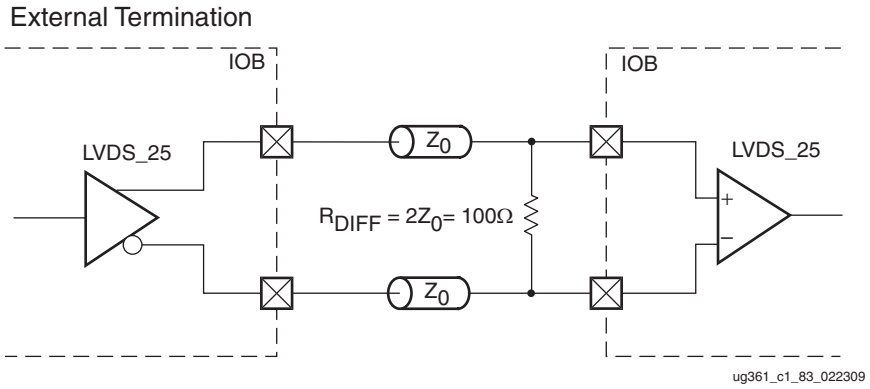


Figure 1-82: **LVDS_25 Receiver Termination**

Figure 1-83 is an example of a differential termination for an LVDS receiver on a board with 50Ω transmission lines.

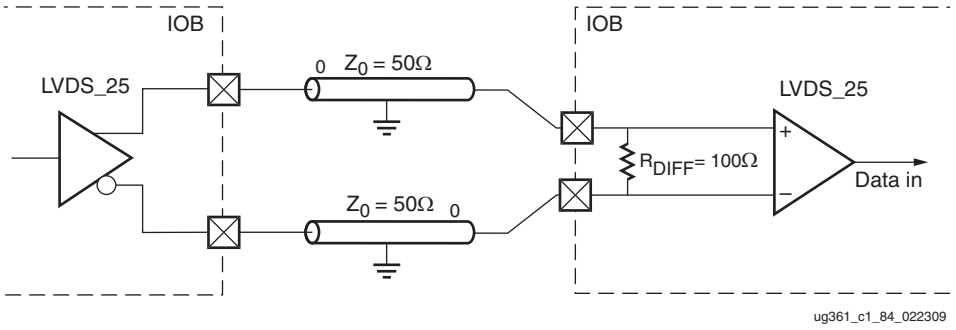


Figure 1-83: **LVDS_25 With DIFF_TERM Receiver Termination**

Table 1-30 lists the available Virtex-6 FPGA LVDS I/O standards and attributes supported.

Table 1-30: **Allowed Attributes of the LVDS I/O Standard**

Attributes	Primitives	
	IBUFDS/IBUFGDS	OBUFDS/OBUFTDS
IOSTANDARD	LVDS_25, LVDSEXT_25	
DIFF_TERM	TRUE, FALSE	N/A

HyperTransport™ Protocol (HT)

The HyperTransport protocol (HT) also known as Lightning Data Transport (LDT), is a low-voltage standard for high speed interfaces. Its differential signaling based interface is very similar to LVDS. Virtex-6 FPGA IOBs are equipped with HT buffers. Table 1-31 summarizes all the possible HT I/O standards and attributes supported.

Table 1-31: **Allowed Attributes of the HT I/O Standard**

Attributes	Primitives	
	IBUFDS/IBUFGDS	OBUFDS/OBUFTDS
IOSTANDARD	HT_25	
DIFF_TERM	TRUE, FALSE	N/A

Reduced Swing Differential Signaling (RSDS)

Reduced Swing Differential Signaling (RSDS) is similar to an LVDS high-speed interface using differential signaling. RSDS has a similar implementation to LVDS in Virtex-6 devices and is only intended for point-to-point applications. Table 1-32 summarizes all the possible RSDS I/O standards and attributes supported.

Table 1-32: **Allowed Attributes of the RSDS I/O Standard**

Attributes	Primitives	
	IBUFDS/IBUFGDS	OBUFDS/OBUFTDS
IOSTANDARD	RSDS_25	
DIFF_TERM	TRUE, FALSE	N/A

BLVDS (Bus LVDS)

Since LVDS is intended for point-to-point applications, BLVDS is not an EIA/TIA standard implementation and requires careful adaptation of I/O and PCB layout design rules. The primitive supplied in the software library for bidirectional LVDS does not use the Virtex-6 FPGA LVDS current-mode driver, instead, it uses complementary single-ended differential drivers. Therefore, source termination is required. Figure 1-84 shows the BLVDS transmitter termination.

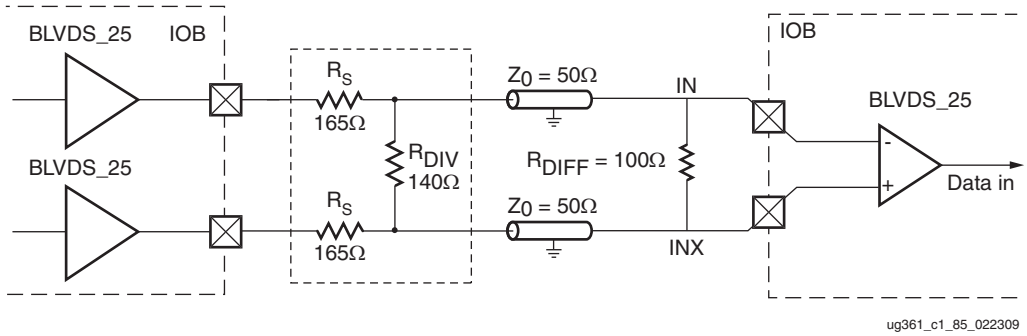


Figure 1-84: **BLVDS Transmitter Termination**

Differential LVPECL (Low-Voltage Positive Emitter-Coupled Logic)

LVPECL is a very popular and powerful high-speed interface in many system applications. Virtex-6 FPGA I/Os are designed to comply with the EIA/TIA electrical specifications for 2.5V LVPECL to make system and board design easier.

LVPECL Transceiver Termination

The Virtex-6 FPGA LVPECL transmitter and receiver requires the termination shown in Figure 1-85, illustrating a Virtex-6 FPGA LVPECL transmitter and receiver on a board with 50 Ω transmission lines. The LVPECL driver is composed of two LVCMOS drivers that form a compliant LVPECL output when combined with the three resistor output termination circuit.

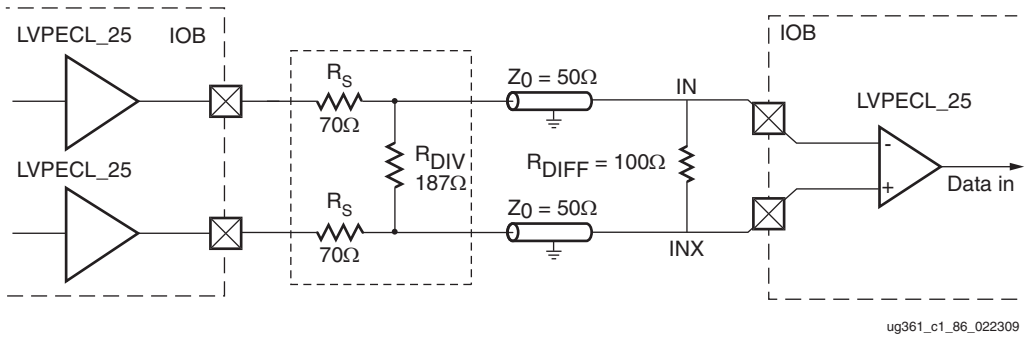


Figure 1-85: **LVPECL Transmitter Termination**

Rules for Combining I/O Standards in the Same Bank

The following rules must be obeyed to combine different input, output, and bidirectional standards in the same bank:

1. **Combining output standards only.** Output standards with the same output V_{CCO} requirement can be combined in the same bank.

   Compatible example:

   SSTL2_I and LVDCI_25 outputs

   Incompatible example:

   SSTL2_I (output V_{CCO} = 2.5V) and
   LVCMOS18 (output V_{CCO} = 1.8V) outputs

2. **Combining input standards only.** Input standards with the same V_{CCO} and V_{REF} requirements can be combined in the same bank.

   Compatible example:

   LVCMOS15 and HSTL_II inputs

   Incompatible example:

   LVCMOS15 (input V_{CCO} = 1.5V) and
   LVCMOS18 (input V_{CCO} = 1.8V) inputs

   Incompatible example:

   HSTL_I_DCI_18 (V_{REF} = 0.9V) and
   HSTL_I_DCI (V_{REF} = 0.75V) inputs

3. **Combining input standards and output standards.** Input standards and output standards with the same V_{CCO} requirement can be combined in the same bank.

   Compatible example:

   LVDS_25 output and HSTL_I input

   Incompatible example:

   LVDS_25 output (output V_{CCO} = 2.5V) and
   HSTL_I_DCI_18 input (input V_{CCO} = 1.8V)

4. **Combining bidirectional standards with input or output standards.** When combining bidirectional I/O with other standards, make sure the bidirectional standard can meet the first three rules.

5. **Additional rules for combining DCI I/O standards.**

   a. No more than one Single Termination type (input or output) is allowed in the same bank.

      Incompatible example:

      HSTL_III_DCI_18 input and HSTL_III_DCI input

   b. No more than one Split Termination type (input or output) is allowed in the same bank.

      Incompatible example:

      HSTL_I_DCI input and HSTL_II_DCI input

The implementation tools enforce these design rules.

Table 1-33, summarizes the Virtex-6 FPGA supported I/O standards.

Table 1-33: **I/O Compatibility**

I/O Standard	V_{CCO}		V_{REF}	Termination Type	
	Output	Input	Input	Output	Input
LVDS_25	2.5	Note (1)	N/R	N/R	N/R
LVDSEXT_25			N/R	N/R	N/R
HT_25			N/R	N/R	N/R
RSDS_25(3)			N/R	N/R	N/R
BLVDS_25			N/R	N/R	N/R
LVPECL_25			N/R	N/R	N/R
SSTL2_I			1.25	N/R	N/R
SSTL2_II			1.25	N/R	N/R
DIFF_SSTL2_I			N/R	N/R	N/R
DIFF_SSTL2_II			N/R	N/R	N/R
LVCMOS25		2.5	N/R	N/R	N/R
LVDCI_25			N/R	Series	N/R
HSLVDCI_25			$V_{CCO}/2$	Series	N/R
LVDCI_DV2_25			N/R	Series	N/R
SSTL2_I_DCI			1.25	N/R	Split
SSTL2_II_DCI			1.25	Split	Split
SSTL2_II_T_DCI			1.25	N/R	Split
DIFF_SSTL2_I_DCI			N/R	N/R	Split
DIFF_SSTL2_II_DCI			N/R	Split	Split
DIFF_SSTL2_II_T_DCI			N/R	N/R	Split

Table 1-33: **I/O Compatibility *(Cont'd)***

I/O Standard	V_{CCO}		V_{REF}	Termination Type	
	Output	Input	Input	Output	Input
HSTL_III_18	1.8	Note (1)	1.08	N/R	N/R
HSTL_I_18			0.9	N/R	N/R
HSTL_II_18			0.9	N/R	N/R
DIFF_HSTL_I_18			N/R	N/R	N/R
DIFF_HSTL_II_18			N/R	N/R	N/R
SSTL18_I			0.9	N/R	N/R
SSTL18_II			0.9	N/R	N/R
DIFF_SSTL18_I			N/R	N/R	N/R
DIFF_SSTL18_II			N/R	N/R	N/R
LVCMOS18		1.8	N/R	N/R	N/R
LVDCI_18			N/R	Series	N/R
HSLVDCI_18			$V_{CCO}/2$	Series	N/R
LVDCI_DV2_18			N/R	Series	N/R
HSTL_III_DCI_18			1.08	N/R	Single
HSTL_I_DCI_18			0.9	N/R	Split
HSTL_II_DCI_18			0.9	Split	Split
HSTL_II_T_DCI_18			0.9	N/R	Split
DIFF_HSTL_I_DCI_18			N/R	N/R	Split
DIFF_HSTL_II_DCI_18			N/R	Split	Split
DIFF_HSTL_II_T_DCI_18			N/R	N/R	Split
SSTL18_I_DCI			0.9	N/R	Split
SSTL18_II_DCI			0.9	Split	Split
SSTL18_II_T_DCI			0.9	N/R	Split
DIFF_SSTL18_I_DCI			N/R	N/R	Split
DIFF_SSTL18_II_DCI			N/R	Split	Split
DIFF_SSTL18_II_T_DCI			N/R	N/R	Split

Table 1-33: **I/O Compatibility *(Cont'd)***

I/O Standard	V_{CCO}		V_{REF}	Termination Type	
	Output	Input	Input	Output	Input
HSTL_III	1.5	Note (1)	0.9	N/R	N/R
HSTL_I			0.75	N/R	N/R
HSTL_II			0.75	N/R	N/R
DIFF_HSTL_I			N/R	N/R	N/R
DIFF_HSTL_II			N/R	N/R	N/R
LVCMOS15		1.5	N/R	N/R	N/R
LVDCI_15			N/R	Series	N/R
HSLVDCI_15			$V_{CCO}/2$	Series	N/R
LVDCI_DV2_15			N/R	Series	N/R
HSTL_III_DCI			0.9	N/R	Single
HSTL_I_DCI			0.75	N/R	Split
HSTL_II_DCI			0.75	Split	Split
HSTL_II_T_DCI			0.75	N/R	Split
DIFF_HSTL_I_DCI			N/R	N/R	Split
DIFF_HSTL_II_DCI			N/R	Split	Split
DIFF_HSTL_II_T_DCI			N/R	N/R	Split
SSTL15			0.75	N/R	N/R
SSTL15_DCI			0.75	N/R	Split
SSTL15_T_DCI			0.75	N/R	Split
DIFF_SSTL15			N/R	N/R	N/R
DIFF_SSTL15_DCI			N/R	N/R	Split
DIFF_SSTL15_T_DCI			N/R	N/R	Split
LVCMOS12	1.2	1.2	N/R	N/R	N/R
HSTL_I_12			0.6	N/R	N/R

Notes:

1. Differential inputs and inputs using V_{REF} are powered from V_{CCAUX}.
2. N/R = no requirement.
3. RSDS_25 has the same DC specifications as LVDS_25. All information pertaining to LVDS_25 is applicable to RSDS_25.
4. I/O standard is selected using the IOSTANDARD attribute.

XILINX®

Chapter 2

SelectIO Logic Resources

Introduction

This chapter describes the logic directly behind the I/O drivers and receivers covered in Chapter 1, SelectIO Resource.

Virtex-6 FPGAs contain all of the basic I/O logic resources from Virtex-5 FPGAs. These resources include the following with programmable input inversion:

- Combinatorial input/output
- 3-state output control with programmable input inversion
- Registered input/output
- Registered 3-state output control with programmable input inversion
- Double-Data-Rate (DDR) input/output
- DDR output 3-state control
- IODELAYE1 provides users control of an adjustable, fine-resolution delay element
- SAME_EDGE output DDR mode
- SAME_EDGE and SAME_EDGE_PIPELINED input DDR mode

ILOGIC Resources

The ILOGIC block shown in Figure 2-1.

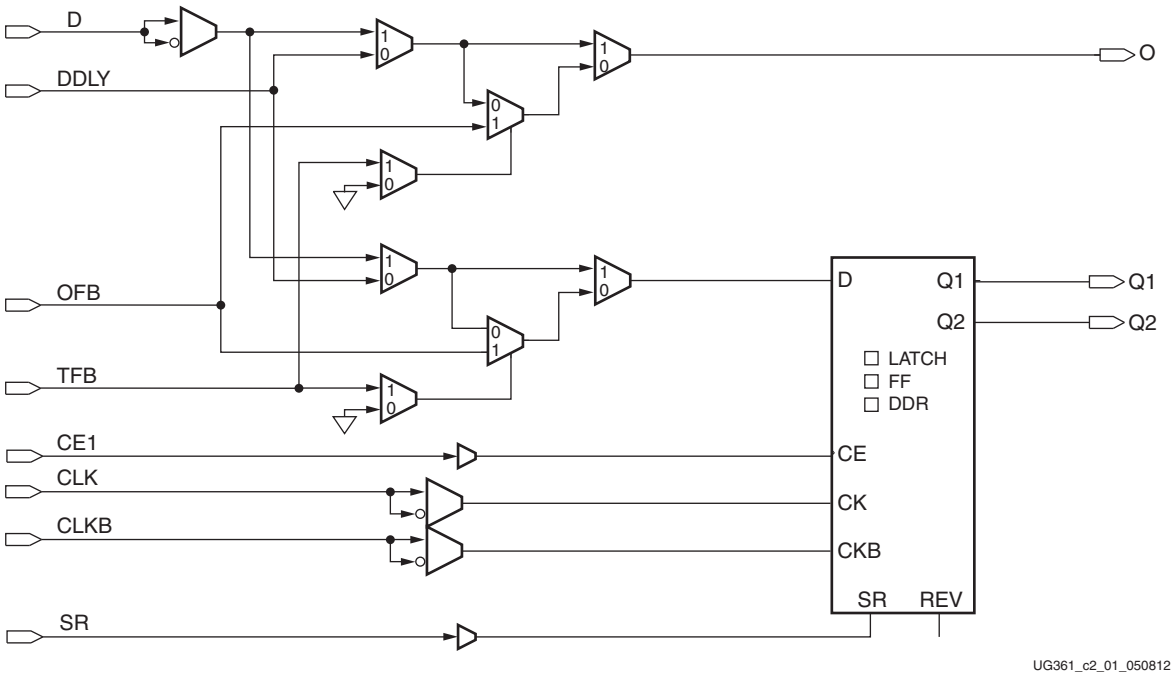


Figure 2-1: **ILOGIC Block Diagram**

ILOGIC can support the following operations:

- Edge-triggered D-type flip-flop
- IDDR mode (OPPOSITE_EDGE or SAME_EDGE or SAME_EDGE_PIPELINED). See Input DDR Overview (IDDR), page 93 for further discussion on input DDR.
- Level sensitive latch
- Asynchronous/combinatorial

All ILOGIC block registers have a common clock enable signal (CE1) that is active High by default. If left unconnected, the clock enable pin for any storage element defaults to the active state.

All ILOGIC block registers have a common synchronous or asynchronous set and reset (SR signal). The set/reset input pin, SR forces the storage element into the state specified by the SRVAL attributes. The reset condition predominates over the set condition.

The SRVAL attributes can be set individually for each storage element in the ILOGIC block, but the choice of synchronous or asynchronous set/reset (SRTYPE) can not be set individually for each storage element in the ILOGIC block.

The following sections discuss the various resources within the ILOGIC blocks. All connections between the ILOGIC resources are managed in Xilinx software.

Combinatorial Input Path

The combinatorial input path is used to create a direct connection from the input driver to the FPGA logic. The data input (D) can be inverted in the ILOGIC. This path is used by software automatically when:

1. There is a direct (unregistered) connection from input data to logic resources in the FPGA logic.

2. An inverter used on the combinatorial input path can be absorbed into the ILOGIC.
3. The "pack I/O register/latches into IOBs" is set to OFF.

Input DDR Overview (IDDR)

Virtex-6 devices have dedicated registers in the ILOGIC to implement input double-data-rate (DDR) registers. This feature is used by instantiating the IDDR primitive.

There is only one clock input to the IDDR primitive. Falling edge data is clocked by a locally inverted version of the input clock. All clocks feeding into the I/O tile are fully multiplexed, i.e., there is no clock sharing between ILOGIC and OLOGIC blocks. The IDDR primitive supports the following modes of operation:

- OPPOSITE_EDGE mode
- SAME_EDGE mode
- SAME_EDGE_PIPELINED mode

The SAME_EDGE and SAME_EDGE_PIPELINED modes are the same as for the Virtex-5 architecture. These modes allow designers to transfer falling edge data to the rising edge domain within the ILOGIC block, saving CLB and clock resources, and increasing performance. These modes are implemented using the DDR_CLK_EDGE attribute. The following sections describe each of the modes in detail.

OPPOSITE_EDGE Mode

A traditional input DDR solution, or OPPOSITE_EDGE mode, is accomplished via a single input in the ILOGIC. The data is presented to the FPGA logic via the output Q1 on the rising edge of the clock and via the output Q2 on the falling edge of the clock. This structure is similar to the Virtex-5 FPGA implementation. Figure 2-2 shows the timing diagram of the input DDR using the OPPOSITE_EDGE mode.

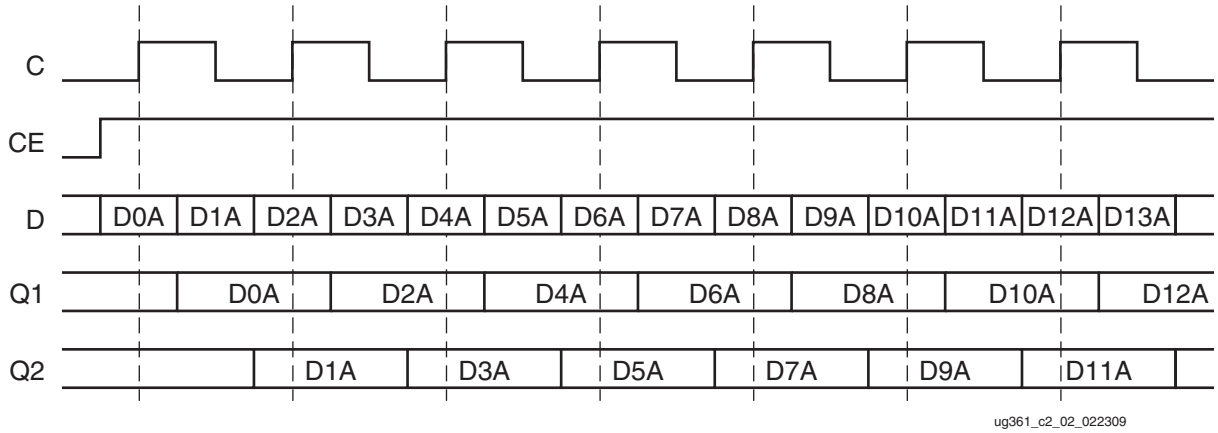


Figure 2-2: **Input DDR Timing in OPPOSITE_EDGE Mode**

SAME_EDGE Mode

In the SAME_EDGE mode, the data is presented into the FPGA logic on the same clock edge. However, the data pair to be separated by one clock cycle. This structure is similar to the Virtex-5 FPGA implementation.

Figure 2-3 shows the timing diagram of the input DDR using SAME_EDGE mode. In the timing diagram, the output pairs Q1 and Q2 are no longer (0) and (1). Instead, the first pair

presented is pair Q1 and Q2 (0) and (don't care) respectively, followed by pair (1) and (2) on the next clock cycle.

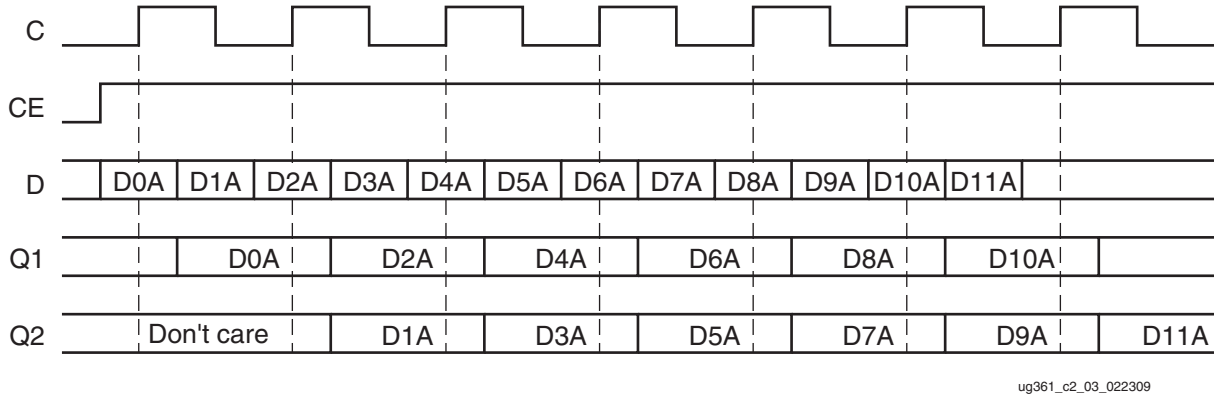


Figure 2-3: **Input DDR Timing in SAME_EDGE Mode**

SAME_EDGE_PIPELINED Mode

In the SAME_EDGE_PIPELINED mode, the data is presented into the FPGA logic on the same clock edge.

Unlike the SAME_EDGE mode, the data pair is not separated by one clock cycle. However, an additional clock latency is required to remove the separated effect of the SAME_EDGE mode. Figure 2-4 shows the timing diagram of the input DDR using the SAME_EDGE_PIPELINED mode. The output pairs Q1 and Q2 are presented to the FPGA logic at the same time.

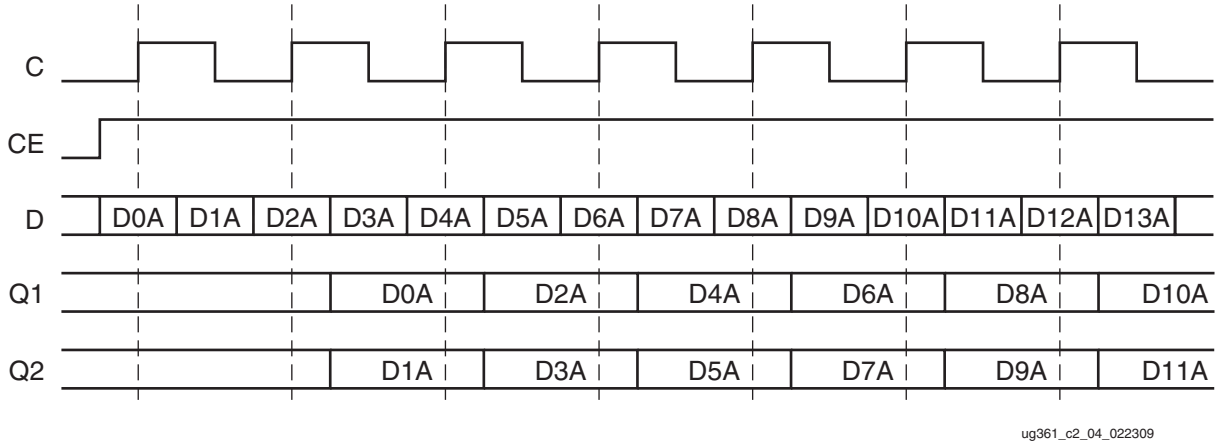


Figure 2-4: **Input DDR Timing in SAME_EDGE_PIPELINED Mode**

Input DDR Primitive (IDDR)

Figure 2-5 shows the block diagram of the IDDR primitive. Set and Reset are not supported at the same time. Table 2-1 lists the IDDR port signals. Table 2-2 describes the various attributes available and default values for the IDDR primitive.

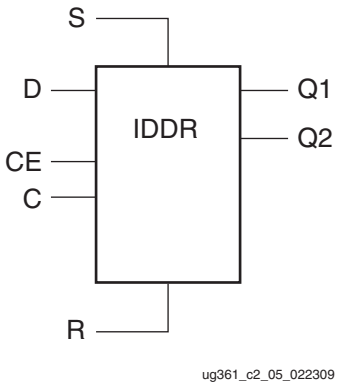


Figure 2-5: **IDDR Primitive Block Diagram**

Table 2-1: **IDDR Port Signals**

Port Name	Function	Description
Q1 and Q2	Data outputs	IDDR register outputs.
C	Clock input port	The C pin represents the clock input pin.
CE	Clock enable port	The enable pin affects the loading of data into the DDR flip-flop. When Low, clock transitions are ignored and new data is not loaded into the DDR flip-flop. CE must be High to load new data into the DDR flip-flop.
D	Data input (DDR)	IDDR register input from IOB.
R	Reset	Synchronous/Asynchronous reset pin. Reset is asserted High. Not supported when using Set.
S	Set	Synchronous/Asynchronous set pin. Set is asserted High. Not supported when using Reset.

Table 2-2: **IDDR Attributes**

Attribute Name	Description	Possible Values
DDR_CLK_EDGE	Sets the IDDR mode of operation with respect to clock edge	OPPOSITE_EDGE (default), SAME_EDGE, SAME_EDGE_PIPELINED
INIT_Q1	Sets the initial value for Q1 port	0 (default), 1
INIT_Q2	Sets the initial value for Q2 port	0 (default), 1
SRTYPE	Set/Reset type with respect to clock (C)	ASYNC (default), SYNC

IDDR VHDL and Verilog Templates

The Libraries Guide includes templates for instantiation of the IDDR primitive in VHDL and Verilog.

ILOGIC Timing Models

This section describes the timing associated with the various resources within the ILOGIC block.

ILOGIC Timing Characteristics

Figure 2-6 illustrates ILOGIC register timing. When IDELAY is used, T_{IDOCK} is replaced by T_{IDOCKD}.

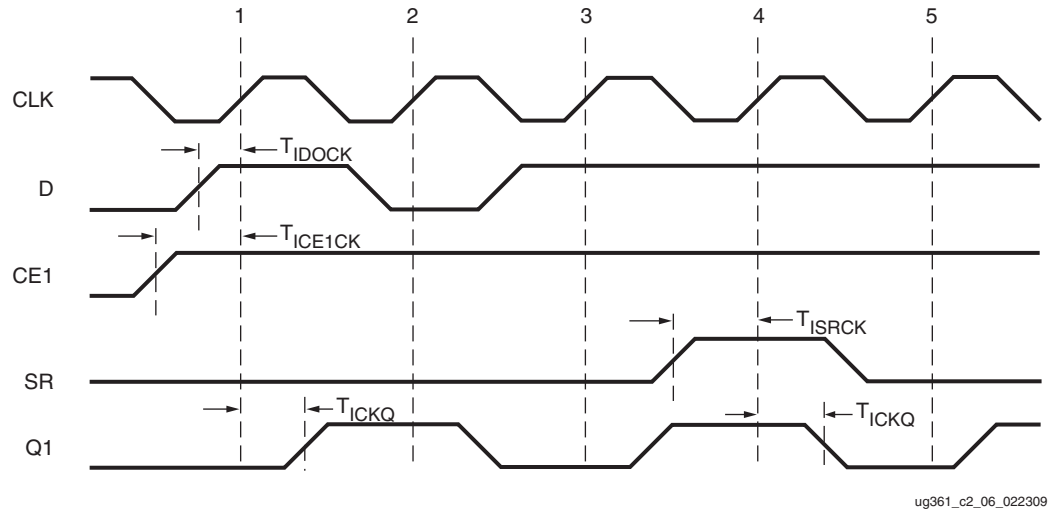


Figure 2-6: **ILOGIC Input Register Timing Characteristics**

Clock Event 1

- At time T_{ICE1CK} before Clock Event 1, the input clock enable signal becomes valid-high at the CE1 input of the input register, enabling the input register for incoming data.
- At time T_{IDOCK} before Clock Event 1, the input signal becomes valid-high at the D input of the input register and is reflected on the Q1 output of the input register at time T_{ICKQ} after Clock Event 1.

Clock Event 4

- At time T_{ISRCK} before Clock Event 4, the SR signal (configured as synchronous reset in this case) becomes valid-high resetting the input register and reflected at the Q1 output of the IOB at time T_{ICKQ} after Clock Event 4.

ILOGIC Timing Characteristics, DDR

Figure 2-7 illustrates the ILOGIC in IDDR mode timing characteristics. When IDELAY is used, T_{IDOCK} is replaced by T_{IDOCKD}. The example shown uses IDDR in OPPOSITE_EDGE mode. For other modes, add the appropriate latencies as shown in Figure 2-4, page 94.

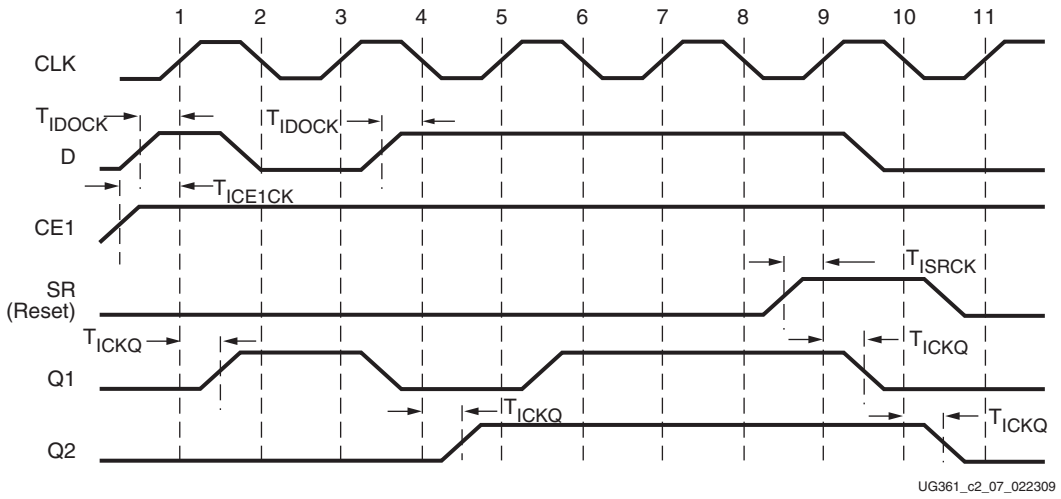


Figure 2-7: **ILOGIC in IDDR Mode Timing Characteristics (OPPOSITE_EDGE Mode)**

Clock Event 1

- At time T_{ICE1CK} before Clock Event 1, the input clock enable signal becomes valid-high at the CE1 input of both of the DDR input registers, enabling them for incoming data. Since the CE1 and D signals are common to both DDR registers, care must be taken to toggle these signals between the rising edges and falling edges of CLK as well as meeting the register setup-time relative to both clocks.
- At time T_{IDOCK} before Clock Event 1 (rising edge of CLK), the input signal becomes valid-high at the D input of both registers and is reflected on the Q1 output of input-register 1 at time T_{ICKQ} after Clock Event 1.

Clock Event 2

- At time T_{IDOCK} before Clock Event 2 (falling edge of CLK), the input signal becomes valid-low at the D input of both registers and is reflected on the Q2 output of input-register 2 at time T_{ICKQ} after Clock Event 2 (no change in this case).

Clock Event 9

- At time T_{ISRCK} before Clock Event 9, the SR signal (configured as synchronous reset in this case) becomes valid-high resetting Q1 at time T_{ICKQ} after Clock Event 9, and Q2 at time T_{ICKQ} after Clock Event 10.

Table 2-3 describes the function and control signals of the ILOGIC switching characteristics in the *Virtex-6 FPGA Data Sheet*.

Table 2-3: **ILOGIC Switching Characteristics**

Symbol	Description
Setup/Hold	
T_{ICE1CK}/T_{ICKCE1}	CE1 pin Setup/Hold with respect to CLK
T_{ISRCK}/T_{ICKSR}	SR pin Setup/Hold with respect to CLK
T_{IDOCK}/T_{IOCKD}	D pin Setup/Hold with respect to CLK
Combinatorial	
T_{IDI}	D pin to O pin propagation delay, no Delay
Sequential Delays	
T_{IDLO}	D pin to Q1 pin using flip-flop as a latch without Delay
T_{ICKQ}	CLK to Q outputs
T_{ICE1Q}	CE1 pin to Q1 using flip-flop as a latch, propagation delay
T_{RQ}	SR pin to OQ/TQ out

Note: The DDLY timing diagrams and parameters are identical to the D timing diagrams and parameters.

Input/Output Delay Element (IODELAYE1)

Every I/O block contains a programmable absolute delay element called IODELAYE1. The IODELAYE1 can be connected to an ILOGIC/ISERDES or OLOGIC/OSERDES block or both. IODELAYE1 is a 31-tap, wraparound, delay element with a calibrated tap resolution. Refer to the *Virtex-6 FPGA Data Sheet* for delay values. It can be applied to the combinatorial input path, registered input path, combinatorial output path, or registered output path. It can also be accessed directly from the FPGA logic. IODELAYE1 allows incoming signals to be delayed on an individual basis. The tap delay resolution is varied by selecting an IDELAYCTRL reference clock from the range specified in the *Virtex-6 FPGA Data Sheet*. The IODELAYE1 resource can function as IDELAY, ODELAY, or bidirectional delay.

IODELAYE1 Primitive

Figure 2-8 shows the IODELAYE1 primitive.

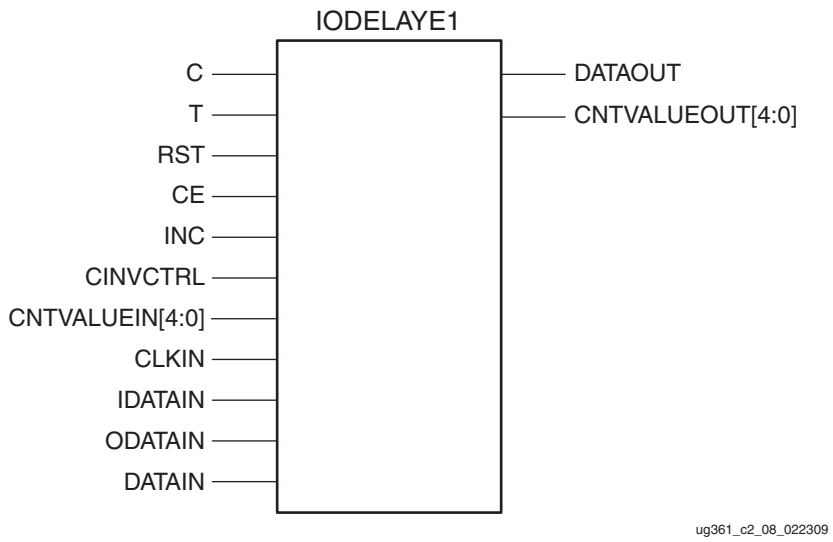


Figure 2-8: **IODELAYE1 Primitive**

Table 2-4 lists the available ports in the IODELAYE1 primitive.

Table 2-4: **IODELAYE1 Primitive Ports**

Port Name	Direction	Width	Function
C	Input	1	Clock input used in VARIABLE or VAR_LOADABLE mode
T	Input	1	3-state input control port. This port determines dynamically if IODELAYE1 is used as IDELAY or ODELAY.
RST	Input	1	Reset the IODELAYE1 element to the pre-programmed value in VARIABLE mode. In VAR_LOADABLE mode, it loads the value of CNTVALUEIN.
CE	Input	1	Enable increment/decrement function
INC	Input	1	Increment/decrement number of tap delays
CINVCTRL	Input	1	Dynamically inverts the clock (C) polarity
CNTVALUEIN	Input	5	Counter value from FPGA logic for dynamically loadable tap value
CLKIN	Input	1	Clock Access into the IODELAYE1 (from the I/O CLKMUX)
IDATAIN	Input	1	Data input for IODELAYE1 from the IOB
DATAIN	Input	1	Data input for IODELAYE1 from the FPGA logic
ODATAIN	Input	1	Data input for IODELAYE1 from the OSERDES/OLOGIC
DATAOUT	Output	1	Delayed data from one of four data input ports (IDATAIN, ODATAIN, DATAIN, and CLKIN)
CNTVALUEOUT	Output	5	Counter value going to FPGA logic for monitoring tap value

IODELAYE1 Ports

Data Input from the IOB - IDATAIN

The IDATAIN input is driven by its associated IOB. In IDELAY mode the data can be driven to either an ILOGIC/ISERDES block, directly into the FPGA logic, or to both through the DATAOUT port with a delay set by the IDELAY_VALUE.

Data Input from the FPGA OLOGIC/OSERDES - ODATAIN

The ODATAIN input is driven by OLOGIC/OSERDES. In ODELAY mode, the ODATAIN drives the DATAOUT port which is connected to an IOB with a delay set by the ODELAY_VALUE.

Data Input from the FPGA Logic - DATAIN

The DATAIN input is directly driven from the FPGA logic providing a logic accessible delay line. The data is driven back into the FPGA logic through the DATAOUT port with a delay set by the IDELAY_VALUE. DATAIN can be locally inverted. The data cannot be driven to an IOB.

Clock Input from Clock Buffer - CLKIN

The CLKIN input is driven from clock buffers (BUFIO, BUFG, or BUFR). The clock is driven back into the FPGA logic through the DATAOUT port with a delay set by the IDELAY_VALUE.

Data Output - DATAOUT

Delayed data from the three data input ports. DATAOUT connects to the FPGA logic (IDELAY mode), IOB (ODELAY mode) or both (bidirectional delay mode). If used in the bidirectional delay mode, the T port dynamically switches between the IDATAIN and ODATAIN paths providing an alternating input/output delay based on the direction indicated by the 3-state signal T from the OLOGIC block.

3-state Input - T

This is the 3-state input control port. For bidirectional operation, the T pin signal also controls the T pin of the OBUFT.

Clock Input - C

All control inputs to IODELAYE1 primitive (RST, CE, and INC) are synchronous to the clock input (C). A clock must be connected to this port when IODELAYE1 is configured in VARIABLE or VAR_LOADABLE mode. C can be locally inverted, and must be supplied by a global or regional clock buffer. This clock should be connected to the same clock in the SelectIO logic resources (when using ISERDES and OSERDES, C is connected to CLKDIV).

Module Reset - RST

When in VARIABLE mode, the IODELAYE1 reset signal, RST, resets the delay element to a value set by the IDELAY_VALUE or ODELAY_VALUE attribute. If these attributes are not specified, a value of zero is assumed. The RST signal is an active-High reset and is synchronous to the input clock signal (C).

When in VAR_LOADABLE mode, the IODELAYE1 reset signal, RST, resets the delay element to a value set by the CNTVALUEIN. The value present at CNTVALUEIN[4:0] will be the new tap value. As a results of this functionality the IDELAY_VALUE and

ODELAY_VALUE attribute is ignored. The reset signal, RST, can thus be seen as a IODELAY load signal.

C Pin Polarity Switch - CINVCTRL

The CINVCTRL pin is used for dynamically switching the polarity of C pin. This is for use in applications when glitches are not an issue. When switching the polarity, do not use IDELAY control pins for two clock cycles.

Control Value In - CNTVALUEIN

The CNTVALUEIN pins are used for dynamically switching the value of the delay element.

Control Value Out - CNTVALUEOUT

The CNTVALUEOUT pins are used for reporting the dynamically switching value of the delay element.

The control pins are summarized in Table 2-5.

Table 2-5: **Control Pin Descriptions**

Pin	Type	Value	Description
INC	Input	1	Increment/decrement number of tap delays
CE	Input	1	Enable increment/decrement function
RST	Input	1	Reset delay element to pre-programmed value. If no value programmed, reset to 0 Reset delay element to pre-programmed value in VARIABLE mode. Reset delay element to the CNTVALUEIN value in VAR_LOADABLE mode.

Increment/Decrement Signals - CE, INC

The increment/decrement is controlled by the enable signal (CE). This interface is only available when either IDELAY, ODELAY, or both are in VARIABLE or VAR_LOADABLE mode.

As long as CE remains High, IDELAY will increment or decrement by $T_{IDELAYRESOLUTION}$ every clock (C) cycle. The state of INC determines whether IDELAY will increment or decrement; INC = 1 increments, INC = 0 decrements, synchronously to the clock (C). If CE is Low the delay through IDELAY will not change regardless of the state of INC.

When CE goes High, the increment/decrement operation begins on the next positive clock cycle. When CE goes Low, the increment/decrement operation ceases on the next positive clock cycle.

IODELAYE1 is a wrap-around programmable delay element. When the end of the delay element is reached (tap 31) a subsequent increment function will return to tap 0. The same applies to the decrement function: decrementing below zero moves to tap 31.

IODELAYE1 Attributes

Table 2-6 summarizes the IODELAYE1 attributes.

Table 2-6: **IODELAYE1 Attribute Summary**

Attribute	Value	Default Value	Description
IDELAY_TYPE	String: DEFAULT, FIXED, VARIABLE, or VAR_LOADABLE	DEFAULT	Sets the type of tap delay line. DEFAULT delay guarantees zero hold times. FIXED delay sets a static delay value. VAR_LOADABLE dynamically loads tap values. VARIABLE delay dynamically adjusts the delay value.
ODELAY_TYPE	String: FIXED, VARIABLE, or VAR_LOADABLE	FIXED	Sets the type of tap delay line. FIXED delay sets a static delay value. VAR_LOADABLE dynamically loads tap values. VARIABLE delay dynamically adjusts the delay value.
DELAY_SRC	String: I, CLKIN, DATAIN, IO, O	I	I: IODELAYE1 chain input is IDATAIN O: IODELAYE1 chain input is ODATAIN IO: IODELAYE1 chain input is IDATAIN and ODATAIN (controlled by T) DATAIN: IODELAYE1 chain input is DATAIN CLKIN: IODELAYE1 chain input is CLKIN
IDELAY_VALUE	Integer: 0 to 31	0	Specifies the fixed number of delay taps in fixed mode or the initial starting number of taps in VARIABLE mode (input path). When IDELAY_TYPE is set to VAR_LOADABLE mode, this value is ignored.
ODELAY_VALUE	Integer: 0 to 31	0	Specifies the fixed number of delay taps in fixed mode or the initial starting number of taps in VARIABLE mode (output path). When IDELAY_TYPE is set to VAR_LOADABLE mode, this value is ignored.
HIGH_PERFORMANCE_MODE	Boolean: FALSE or TRUE	FALSE	When TRUE, this attribute reduces the output jitter. The difference in power consumption is quantified in the Xilinx Power Estimator tool.
SIGNAL_PATTERN	String: DATA, CLOCK	DATA	Causes the timing analyzer to account for the appropriate amount of delay-chain jitter in the data or clock path.
REFCLK_FREQUENCY	Real: 190–210 or 290 to 310	200	Sets the tap value (in MHz) used by the timing analyzer for static timing analysis. The range of 290.0 to 310.0 is not available in all speed grades. See DS152: *Virtex-6 FPGA Data Sheet*.
CINVCTRL_SEL	Boolean: FALSE or TRUE	FALSE	Enables the CINVCTRL_SEL pin to dynamically switch the polarity of the C pin.

IDELAY_TYPE Attribute

The IDELAY_TYPE attribute sets the type of delay used. The attribute values are DEFAULT, FIXED, VARIABLE, and VAR_LOADABLE. When set to DEFAULT, the zero-hold time delay element is selected. The DEFAULT delay element is used to guarantee non-positive hold times when global clocks are used without MMCMs to capture data (pin-to-pin parameters). The DEFAULT delay must be the only delay mode used on a per bank basis and can not be mixed with other delay modes in the same bank.

When the IDELAY_TYPE attribute is set to FIXED, the tap-delay value is fixed at the number of taps determined by the IDELAY_VALUE attribute setting. This value is preset and cannot be changed after configuration.

When the IDELAY_TYPE attribute is set to VARIABLE, the variable tap delay element is selected. The tap delay can be incremented by setting CE = 1 and INC = 1, or decremented by CE = 1 and INC = 0. The increment/decrement operation is synchronous to C, the input clock signal.

When the IDELAY_TYPE attribute is set to VAR_LOADABLE, the variable tap delay element can be changed and dynamically loaded. The tap delay can also be incremented by setting CE = 1 and INC = 1, or decremented by CE = 1 and INC = 0. In VARIABLE mode, the increment/decrement operation is synchronous to C. The RST pin loads the value presented on CNTVALUEIN. This allows the tap value to be dynamically set.

Refer to Table 2-12 for valid combinations of attribute settings.

IDELAY_VALUE Attribute

The IDELAY_VALUE attribute specifies the initial number of tap delays. The possible values are any integer from 0 to 31. The default value is zero. The value of the tap delay reverts to IDELAY_VALUE when the tap delay is reset. In VARIABLE mode this attribute determines the initial setting of the delay line. In VAR_LOADABLE mode, this attribute is not used, and therefore the initial value of the delay line is always zero.

ODELAY_VALUE Attribute

The ODELAY_VALUE attribute specifies tap delays. The possible values are any integer from 0 to 31. The default value is zero. The value of the tap delay reverts to ODELAY_VALUE when the tap delay is reset. In VAR_LOADABLE mode, this attribute is not used, and therefore the initial value of the delay line is always zero.

ODELAY_TYPE Attribute

When set to FIXED, the tap-delay value is fixed at the number of taps determined by the ODELAY_VALUE attribute setting. This value is preset and cannot be changed after configuration.

When set to VARIABLE, the variable tap delay element is selected. The tap delay can be incremented by setting CE = 1 and INC = 1, or decremented by CE = 1 and INC = 0. The increment/decrement operation is synchronous to C, the input clock signal.

When set to VAR_LOADABLE, the variable tap delay element can be changed and dynamically loaded. The tap delay can also be incremented by setting CE = 1 and INC = 1, or decremented by CE = 1 and INC = 0. The increment/decrement operation is synchronous to C. The RST pin in this mode loads the value presented on CNTVALUEIN. This allows the tap value to be dynamically set. When in this mode the ODELAY_VALUE is ignored.

Refer to Table 2-12 for valid combinations of attribute settings.

HIGH_PERFORMANCE_MODE Attribute

When TRUE, this attribute reduces the output jitter. This reduction results in a slight increase in power dissipation from the IODELAYE1 element.

SIGNAL_PATTERN Attribute

Clock and data signals have different electrical profiles and therefore accumulate different amounts of jitter in the IODELAYE1 chain. By setting the SIGNAL_PATTERN attribute, the user enables timing analyzer to account for jitter appropriately when calculating timing. A clock signal is periodic in nature and does not have long sequences of consecutive ones or zeroes, while data is random in nature and can have long and short sequences of ones and zeroes.

IODELAYE1 Modes

When used as IDELAY, the data input comes from either IBUF or the FPGA logic and the output goes to ILOGIC/ISERDES. There are four modes of operation available:

- Zero-hold time delay mode (IDELAY_TYPE = DEFAULT)

  This mode of operation allows backward compatibility for designs using the zero-hold time delay feature in Virtex-5 devices. This delay element is used to provide non-positive hold times when global clocks are used without MMCMs to capture data (pin-to-pin parameters). When used in this mode, the IDELAYCTRL primitive does not need to be instantiated.

  The DEFAULT delay can not be used with the IODELAYE1s in any other configuration in the same bank.

- Fixed delay mode (IDELAY_TYPE = FIXED)

  In the fixed delay mode, the delay value is preset at configuration to the tap number determined by the attribute IDELAY_VALUE. Once configured, this value cannot be changed. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details.

- Variable delay mode (IDELAY_TYPE = VARIABLE)

  In the variable delay mode, the delay value can be changed after configuration by manipulating the control signals CE and INC. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details. The control pins are described in Table 2-7.

Table 2-7: **Control Pin when IDELAY_TYPE = VARIABLE**

C	RST	CE	INC	TAP Setting
0	x	x	x	No Change
1	1	x	x	IDELAY_VALUE
1	0	0	x	No Change
1	0	1	1	Current Value +1
1	0	1	0	Current Value –1
1	0	0	0	No Change

- Loadable variable delay mode (IDELAY_TYPE = VAR_LOADABLE)

In addition to having the same functionality of (IDELAY_TYPE = VARIABLE) in this mode the IDELAY tap can be loaded via the 5-input bits CNTVALUEIN<4:0> from the FPGA logic. When RST is pulsed the value present at CNTVALUEIN<4:0> will be the new tap value. As a results of this functionality the IDELAY_VALUE attribute is ignored. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details. The control pins are described in Table 2-8.

Table 2-8: **Control Pin when IDELAY_TYPE = VAR_LOADABLE**

C	RST	CE	INC	CNTVALUEIN	CNTVALUEOUT	TAP Setting
0	x	x	x	x	No Change	No Change
1	1	x	x	CNTVALUEIN	CNTVALUEIN	CNTVALUEIN
1	0	0	x	x	No Change	No Change
1	0	1	1	x	Current Value +1	Current Value +1
1	0	1	0	x	Current Value –1	Current Value –1
1	0	0	0	0	No Change	No Change

When used as ODELAY, the data input comes from OLOGIC/OSERDES and the data output goes to OBUF. There are three modes of operation available:

- Fixed delay output mode (ODELAY_TYPE = FIXED)

  In the fixed delay output mode, the delay value is preset at configuration to the tap number determined by the attribute ODELAY_VALUE. Once configured, this value cannot be changed. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details.

- Variable delay mode (ODELAY_TYPE = VARIABLE)

  In the variable delay mode, the delay value can be changed after configuration by manipulating the control signals CE and INC. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details. The control pins are described in Table 2-9.

Table 2-9: **Control Pin when ODELAY_TYPE = VARIABLE**

C	RST	CE	INC	TAP Setting
0	x	x	x	No Change
1	1	x	x	ODELAY_VALUE
1	0	0	x	No Change
1	0	1	1	Current Value +1
1	0	1	0	Current Value –1
1	0	0	0	No Change

- Loadable variable delay mode (ODELAY_TYPE = VAR_LOADABLE)

  In addition to having the same functionality of (ODELAY_TYPE = VARIABLE) in this mode the ODELAY tap can be loaded via the 5-input bits CNTVALUEIN<4:0> from the FPGA logic. When RST is pulsed the value present at CNTVALUEIN<4:0> will be the new tap value. As a results of this functionality the ODELAY_VALUE attribute is

ignored. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details. The control pins are described in Table 2-10.

Table 2-10: **Control Pin when ODELAY_TYPE = VAR_LOADABLE**

C	RST	CE	INC	CNTVALUEIN	CNTVALUEOUT	TAP Setting
0	x	x	x	x	No Change	No Change
1	1	x	x	CNTVALUEIN	CNTVALUEIN	CNTVALUEIN
1	0	0	x	x	No Change	No Change
1	0	1	1	x	Current Value +1	Current Value +1
1	0	1	0	x	Current Value –1	Current Value –1
1	0	0	0	0	No Change	No Change

When used as bidirectional delay, the IOB is configured in bidirectional mode. IODELAYE1 alternately delays the data on the input path and output path. There are four modes of operation:

- Fixed IDELAY (IDELAY_TYPE = FIXED) and fixed ODELAY (ODELAY_TYPE = FIXED) mode

  In this mode, both the values for IDELAY and ODELAY are preset at configuration and are determined by the IDELAY_VALUE and ODELAY_VALUE attributes. Once configured, this value cannot be changed. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details.

- Variable IDELAY (IDELAY_TYPE = VARIABLE) and fixed ODELAY (ODELAY_TYPE = FIXED) mode

  In this mode, only the IDELAY value can be dynamically changed after configuration by manipulating the control signals CE and INC. The logic level of the T pin in the IODELAYE1 primitive dynamically determines if the block is in IDELAY or ODELAY mode. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details.

- Variable ODELAY (ODELAY_TYPE = VARIABLE) and fixed IDELAY (IDELAY_TYPE = FIXED) mode

  In this mode, only the ODELAY value can be dynamically changed after configuration by manipulating the control signals CE and INC. The logic level of the T pin in the IODELAYE1 primitive dynamically determines if the block is in IDELAY or ODELAY mode. When used in this mode, the IDELAYCTRL primitive must be instantiated. See IDELAYCTRL Usage and Design Guidelines for more details.

- Loadable variable IDELAY (IDELAY_TYPE = VAR_LOADABLE) and loadable variable ODELAY (ODELAY_TYPE = VAR_LOADABLE)

  In addition to having the same functionality of (ODELAY_TYPE = VARIABLE) and (IDELAY_TYPE = VARIABLE), in this mode the IDELAY tap and ODELAY tap can be loaded via the 5-input bits CNTVALUEIN<4:0> from the FPGA logic. When RST is pulsed the value present at CNTVALUEIN<4:0> will be the new tap value. As a results of this functionality the ODELAY_VALUE attribute is ignored. The control pins are described in Table 2-11.

Table 2-11: **Control Pin when IDELAY_TYPE = VAR_LOADABLE and ODELAY_TYPE = VAR_LOADABLE**

C	T	RST	CE	INC	CNTVALUEIN	CNTVALUEOUT	TAP Setting	DATAOUT
0	0	x	x	x	x	No Change	No Change	ODATAIN
1	0	1	x	x	CNTVALUEIN	CNTVALUEIN	CNTVALUEIN	ODATAIN
1	0	0	0	x	x	No Change	No Change	ODATAIN
1	0	0	1	1	x	Current Value +1	Current Value +1	ODATAIN
1	0	0	1	0	x	Current Value –1	Current Value –1	ODATAIN
1	0	0	0	0	0	No Change	No Change	ODATAIN
0	1	x	x	x	x	No Change	No Change	IDATAIN
1	1	1	x	x	CNTVALUEIN	CNTVALUEIN	CNTVALUEIN	IDATAIN
1	1	0	0	x	x	No Change	No Change	IDATAIN
1	1	0	1	1	x	Current Value +1	Current Value +1	IDATAIN
1	1	0	1	0	x	Current Value –1	Current Value –1	IDATAIN
1	1	0	0	0	0	No Change	No Change	IDATAIN

Table 2-12 lists the supported IODELAYE1 configurations.

Table 2-12: **IODELAYE1 Configurations Supported**

IODELAYE1 Modes	Direction of IODELAYE1	Input Pin Used in the IODELAYE1 Element	Source	Destination	Supported Delay Modes
IDELAY Unidirectional Input Delay	I	IDATAIN	IBUF	ILOGIC, ISERDES, or Fabric	DEFAULT, FIXED,VARIABLE, or VAR_LOADABLE
		CLKIN	BUFIO, BUFG, BUFR		FIXED, VARIABLE, or VAR_LOADABLE
		DATAIN	Fabric		FIXED, VARIABLE, or VAR_LOADABLE
ODELAY Unidirectional Output Delay	O	ODATAIN	BUFG, OLOGIC, OSERDES (OFB or Q)	OBUF or CLKPERFDELAY	FIXED, VARIABLE, or VAR_LOADABLE
Bidirectional Delay	I (when T = 1)	IDATAIN	IBUF	ILOGIC, ISERDES, or Fabric	I = FIXED, O = FIXED I = VARIABLE, O = FIXED I = FIXED, O = VARIABLE I = VAR_LOADABLE O = VAR_LOADABLE
	O (when T = 0)	ODATAIN	BUFG, OLOGIC, OSERDES (OFB or Q)	OBUF or CLKPERFDELAY	

IODELAYE1 Timing

Table 2-13 shows the IODELAYE1 switching characteristics.

Table 2-13: **IODELAYE1 Switching Characteristics**

Symbol	Description
$T_{IDELAYRESOLUTION}$	IDELAY tap resolution
T_{ICECK}/T_{ICKCE}	CE pin Setup/Hold with respect to C
T_{IINCCK}/T_{ICKINC}	INC pin Setup/Hold with respect to C
T_{IRSTCK}/T_{ICKRST}	RST pin Setup/Hold with respect to C

Figure 2-9 shows an IODELAYE1 (IDELAY_TYPE = VARIABLE, IDELAY_VALUE = 0, and DELAY_SRC = I) timing diagram. It is assumed that IDELAY_VALUE = 0.

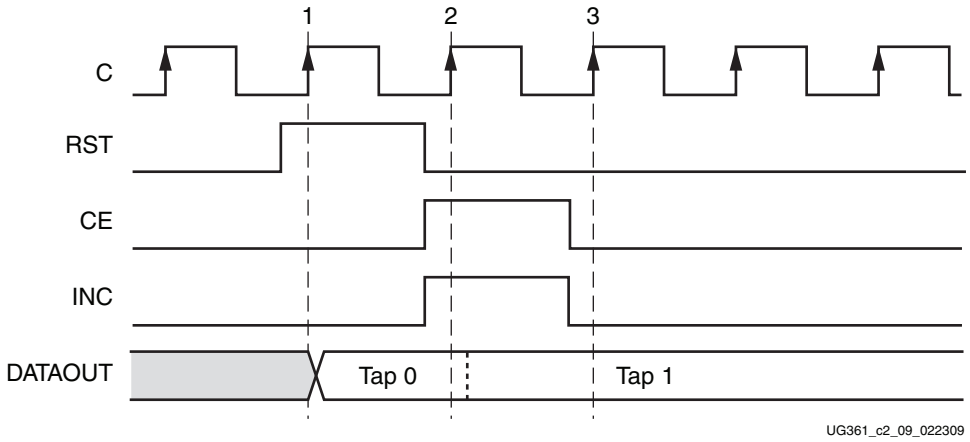


Figure 2-9: **IDELAY Timing Diagram**

Clock Event 1

On the rising edge of C, a reset is detected, causing the output DATAOUT to select tap 0 as the output from the 31-tap chain.

Clock Event 2

A pulse on CE and INC is captured on the rising edge of C. This indicates an increment operation. The output changes without glitches from tap 0 to tap 1. See Stability after an Increment/Decrement Operation.

Clock Event 3

CE and INC are no longer asserted, thus completing the increment operation. The output remains at tap 1 indefinitely until there is further activity on the RST, CE, or INC pins.

Figure 2-10 shows an IODELAYE1 timing diagram.

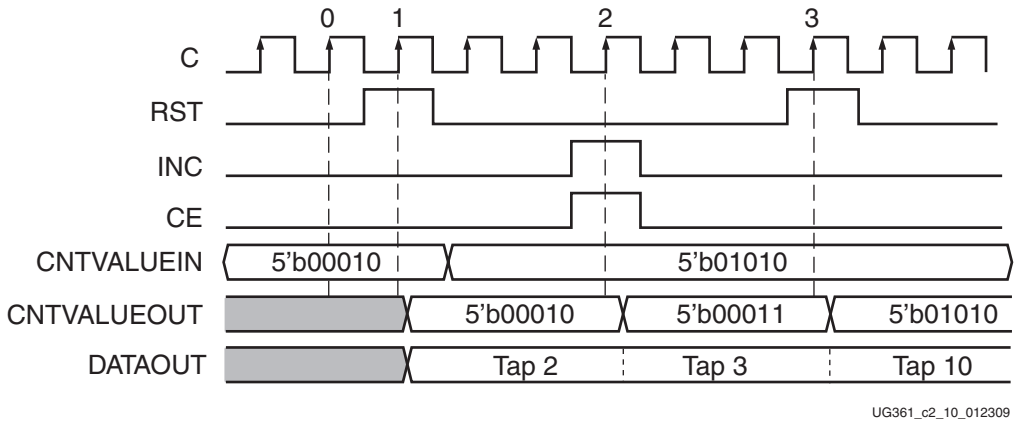


Figure 2-10: **IODELAYE1 in VAR_LOADABLE Timing Diagram**

Clock Event 0

Before RST is pulsed the tap setting and CNTVALUEOUT are at an unknown value.

Clock Event 1

On the rising edge of C, RST is detected causing the output DATAOUT to be equal to the CNTINVALUE, and changing the tap setting to tap 2. The CNTVALUEOUT is updated to represent the new tap value.

Clock Event 2

A pulse on CE and INC are captured on the rising edge of C. This indicates an increment operation. The output changes without glitches from tap 2 to tap 3. The CNTVALUEOUT is updated to represent the new tap value.

Clock Event 3

On the rising edge of C, a RST is detected causing the output DATAOUT to be equal to the CNTINVALUE. The CNTVALUEOUT shows the value of the tap setting. The output will remain at tap 10 indefinitely until there is further activity on the RST, CE, or INC pins.

Stability after an Increment/Decrement Operation

Figure 2-9 shows the delay line changing from tap 0 to tap1 in response to an INC and CE command. Clearly, when the data value at tap 0 is different from the data value at tap 1, the output must change state. However, when the data values at tap 0 and tap 1 are the same (e.g., both 0 or both 1), then the transition from tap 0 to tap 1 causes no glitch or disruption on the output. This concept can be better comprehended by imagining the receiver data signal passing through the IODELAYE1 tap chain. If tap 0 and tap 1 are both near the center of the receiver data eye, then the data sampled at tap 0 will be no different than the data sampled at tap 1. In this case, the transition from tap 0 to tap 1 causes no change to the output. To ensure that this is the case, the increment/decrement operation of IODELAYE1 is designed to be glitchless.

The user can therefore dynamically adjust the IODELAYE1 tap setting in real-time while live user data is passing through the IODELAYE1 element. The adjustments do not disrupt the live user data as long as the current delay line value is near the center of the received data eye.

The glitchless behavior also applies when an IODELAYE1 element is used in the path of a clock signal. Adjusting the tap setting does not cause a glitch or disruption on the output as long as the delay line value is not near the edges seen in the received clock signal. In this case, the tap setting of the IODELAYE1 element in the clock path can be adjusted without disrupting any clock management elements or state machines that could be running on that clock.

IODELAYE1 VHDL and Verilog Instantiation Template

VHDL and Verilog instantiation templates are available in the Libraries Guide for all primitives and submodules.

In VHDL, each template has a component declaration section and an architecture section. Each part of the template should be inserted within the VHDL design file. The port map of the architecture section should include the design signals names.

IODELAYE1 Turnaround Time Usage Model

When using IODELAYE1 in bidirectional mode, the turnaround time needs to be considered. Figure 2-11 shows a simplified block diagram of the IODELAYE1 in the Virtex-6 FPGA IOB that applies to one use of the bidirectional IODELAYE1 functionality.

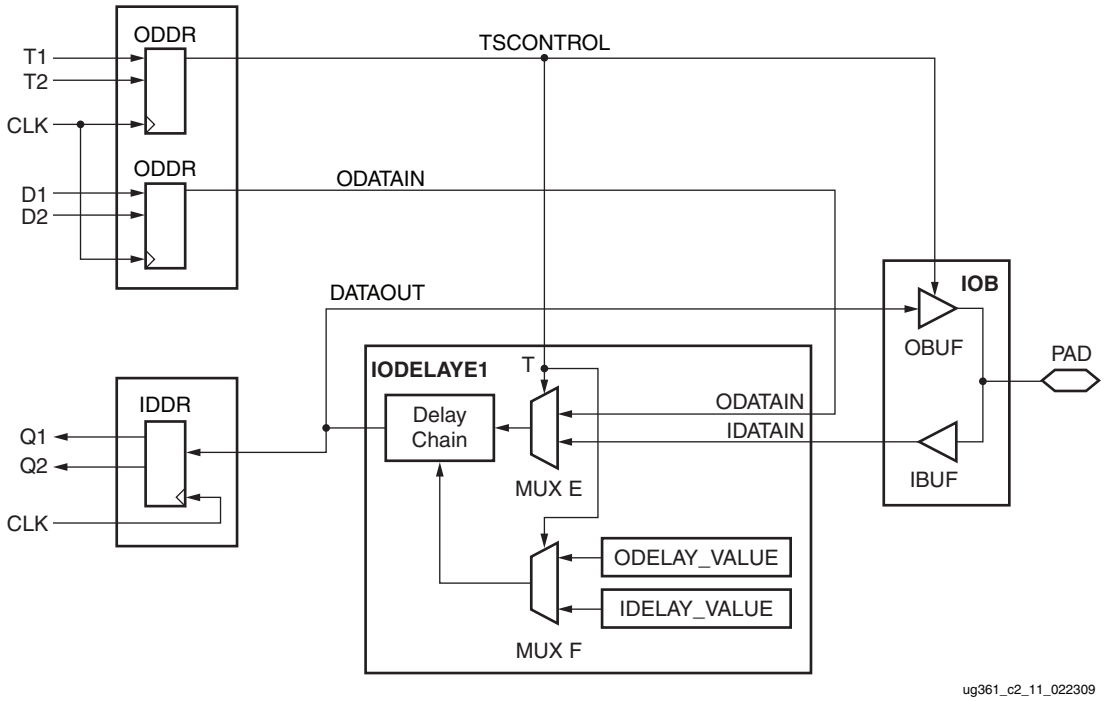


Figure 2-11: **Basic Sections of Blocks Related to IODELAYE1 Turnaround with Pertinent Paths Shown**

When DELAY_SRC = IO, MUXE and MUXF dynamically selects ODATAIN or IDATAIN and ODELAY_VALUE or IDELAY_VALUE inside the IODELAYE1 block.

Two cases that use the bidirectional IODELAYE1 functionality are important for a given I/O pin. The first case uses bidirectional IODELAYE1 when the I/O is an output being switched to an input. Figure 2-12 shows the IOB and IODELAYE1 moving toward the input mode as set by the TSCONTROL net coming from the ODDR flip-flop. This controls

the selection of MUXes E and F for the IOB input path and IDELAY_VALUE respectively. Additionally, the OBUF is 3-stated.

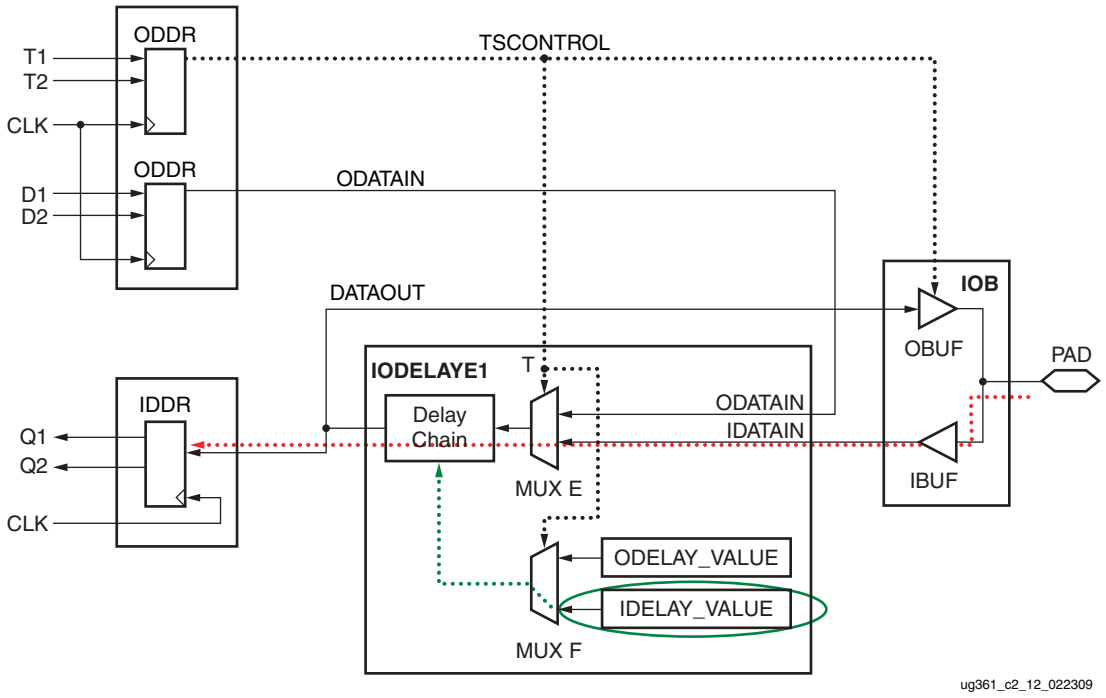


Figure 2-12: **IODELAYE1 and IOB in Input Mode when 3-state is Disabled**

The timing diagram in Figure 2-13 shows the relevant signal timing for the case when the I/O is an output switching to an input using 3-state control. The switching characteristics shown in the diagram are specified in the *Virtex-6 FPGA Data Sheet*.

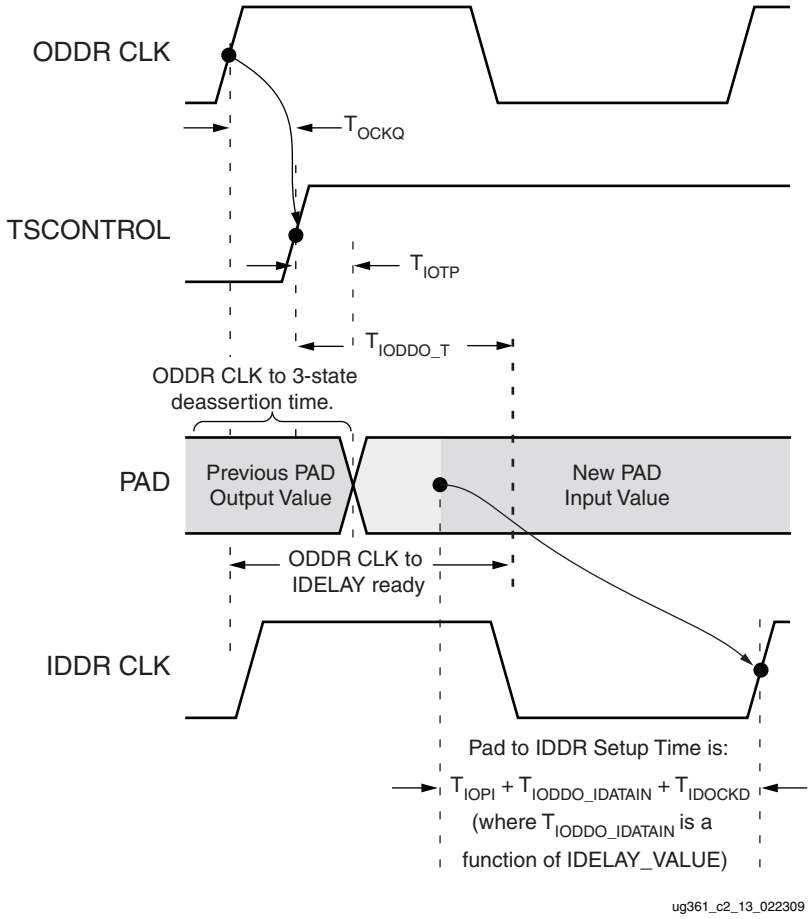


Figure 2-13: **Relevant Timing Signals to Examine IODELAYE1 Timing when the IOB Switches From an Output to an Input**

The activities of the OBUFT pin are controlled by the propagation and state of the TSCONTROL signal from the ODDR flip-flop. The 3-state control data receipt on the OBUF and IDDR flip-flop from a PAD are in parallel with each other, depending on the IDELAY_VALUE setting the final value at the IDDR flip-flop input in response to a clock edge is valid before or after the pad is driven from the 3-state control. After the 3-state control propagates through to the PAD and the IODELAYE1 has been switched to an input, the IDDR setup time is the sole determiner of timing based on the IDELAY_VALUE and other timing parameters defined in the Xilinx speed specification and represented in the ISE tools.

The second case uses bidirectional IODELAYE1 when the I/O is an input switching to an output. Figure 2-14 shows the IOB and IODELAYE1 moving toward the output mode as set by the 3-state TSCONTROL signal coming from the ODDR T flip-flop. This controls the selection of MUXes E and F for the output path and ODELAY_VALUE respectively. Additionally, the OBUF changes to not being 3-stated and starts to drive the PAD.

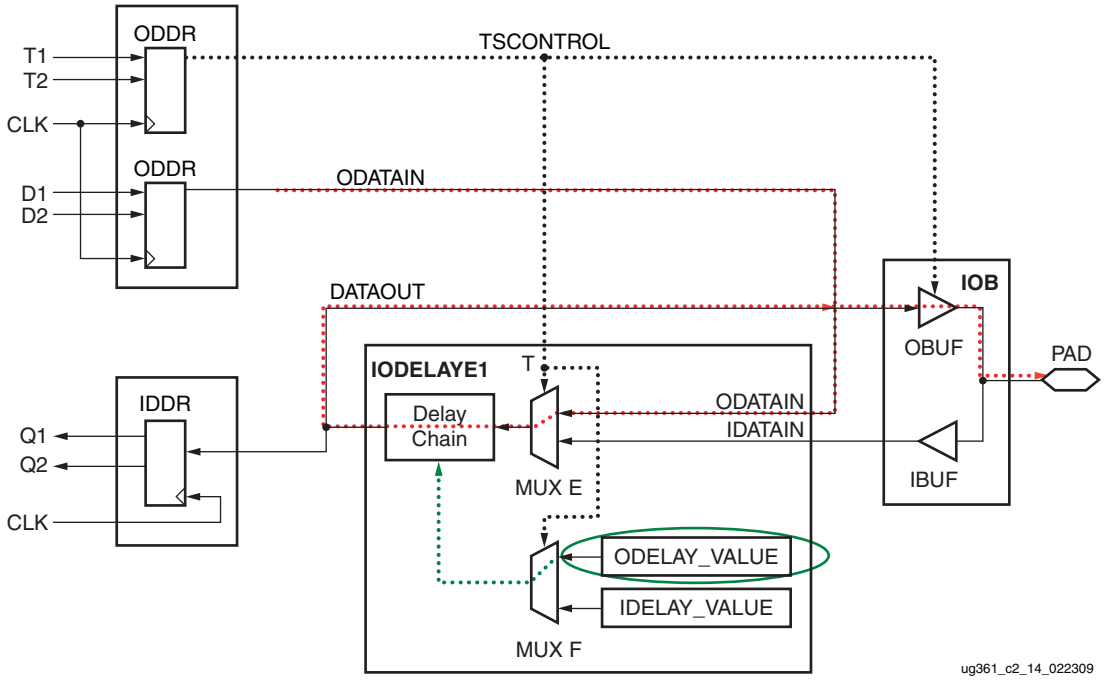


Figure 2-14: **IODELAYE1 and IOB in Output Mode when 3-state is Enabled**

The timing diagram in Figure 2-15 shows the relevant signal timing for the case where the I/O switches from input to an output using 3-state control. The switching characteristics shown in the diagram are specified in the *Virtex-6 FPGA Data Sheet*.

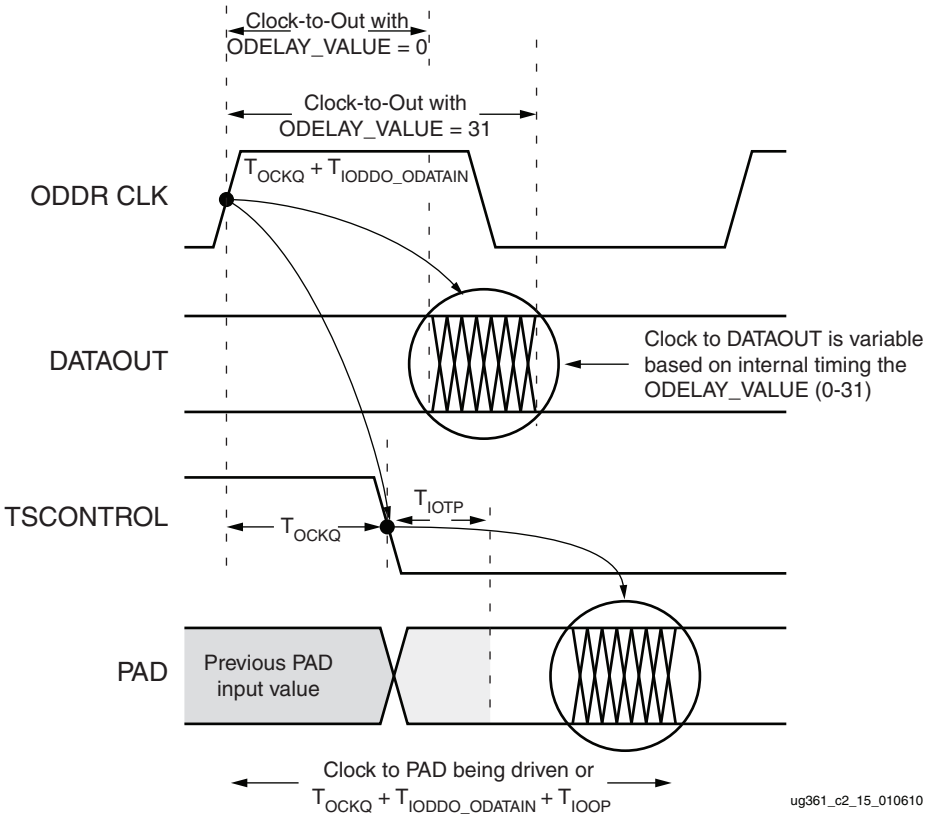


Figure 2-15: **Relevant Timing Signals used to Examine IODELAYE1 Timing when an IOB Changes from an Input to an Output**

3-state control activities on the OBUF of the IOB and ODDR flip-flop to PAD timing are in parallel with each other, depending on the ODELAY_VALUE setting the final output value in response to a clock edge at the ODDR CLK pin is valid before or after the pad is driven from the 3-state control. After the 3-state control propagates through to the PAD and the IODELAYE1 is turned around, the clock-to-output time of the ODDR flip-flop through the IODELAYE1 element (with the ODELAY_VALUE setting) solely determines the clock-to-output time to the pad.

IDELAYCTRL Overview

If the IODELAYE1 primitive is instantiated, the IDELAYCTRL component must also be instantiated. The IDELAYCTRL module continuously calibrates the individual delay elements (IODELAYE1) in its region (see Figure 2-18, page 117), to reduce the effects of process, voltage, and temperature variations. The IDELAYCTRL module calibrates IODELAYE1 using the user supplied REFCLK.

IDELAYCTRL Primitive

Figure 2-16 shows the IDELAYCTRL primitive.

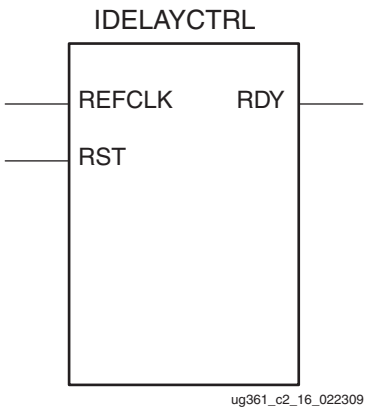


Figure 2-16: **IDELAYCTRL Primitive**

IDELAYCTRL Ports

RST - Reset

The reset input pin (RST) is an active-High asynchronous reset. IDELAYCTRL must be reset after configuration (and the REFCLK signal has stabilized) to ensure proper IODELAYE1 operation. A reset pulse width $T_{IDELAYCTRL\_RPW}$ is required. IDELAYCTRL must be reset after configuration.

REFCLK - Reference Clock

The reference clock (REFCLK) provides a time reference to IDELAYCTRL to calibrate all IODELAYE1 modules in the same region. This clock must be driven by a global clock buffer (BUFGCTRL). REFCLK must be $F_{IDELAYCTRL\_REF}$ ± the specified ppm tolerance (IDELAYCTRL_REF_PRECISION) to guarantee a specified IODELAYE1 resolution ($T_{IDELAYRESOLUTION}$). REFCLK can be supplied directly from a user-supplied source or the MMCM and must be routed on a global clock buffer.

RDY - Ready

The ready (RDY) signal indicates when the IODELAYE1 modules in the specific region are calibrated. The RDY signal is deasserted if REFCLK is held High or Low for one clock period or more. If RDY is deasserted Low, the IDELAYCTRL module must be reset. The implementation tools allow RDY to be unconnected/ignored. Figure 2-17 illustrates the timing relationship between RDY and RST.

IDELAYCTRL Timing

Table 2-14 shows the IDELAYCTRL switching characteristics.

Table 2-14: **IDELAYCTRL Switching Characteristics**

Symbol	Description
$F_{IDELAYCTRL\_REF}$	REFCLK frequency
IDELAYCTRL_REF_PRECISION	REFCLK precision
$T_{IDELAYCTRLCO\_RDY}$	Reset/Startup to Ready for IDELAYCTRL

As shown in Figure 2-17, the Virtex-6 FPGA RST is an edge-triggered signal.

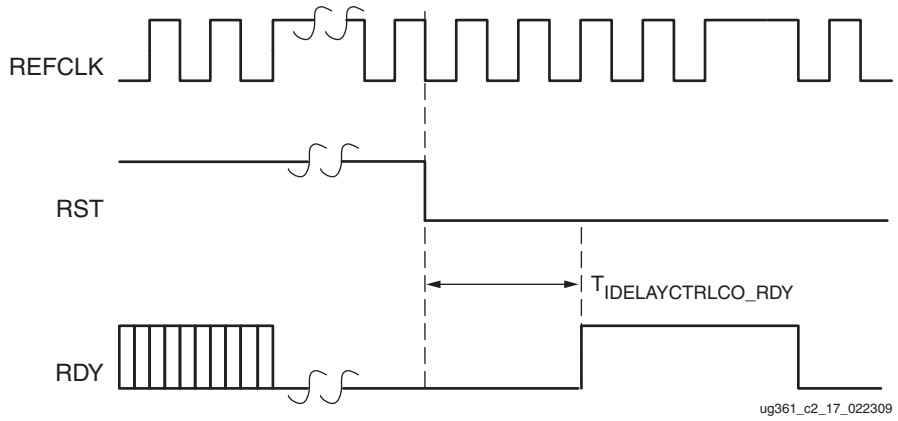


Figure 2-17: **Timing Relationship Between RST and RDY**

IDELAYCTRL Locations

IDELAYCTRL modules exist in every I/O column in every clock region. An IDELAYCTRL module calibrates all the IDELAY modules within its clock region. See the *Virtex-6 FPGA Clocking User Guide* for the definition of a clock region.

Figure 2-18 illustrates the relative locations of the IDELAYCTRL modules.

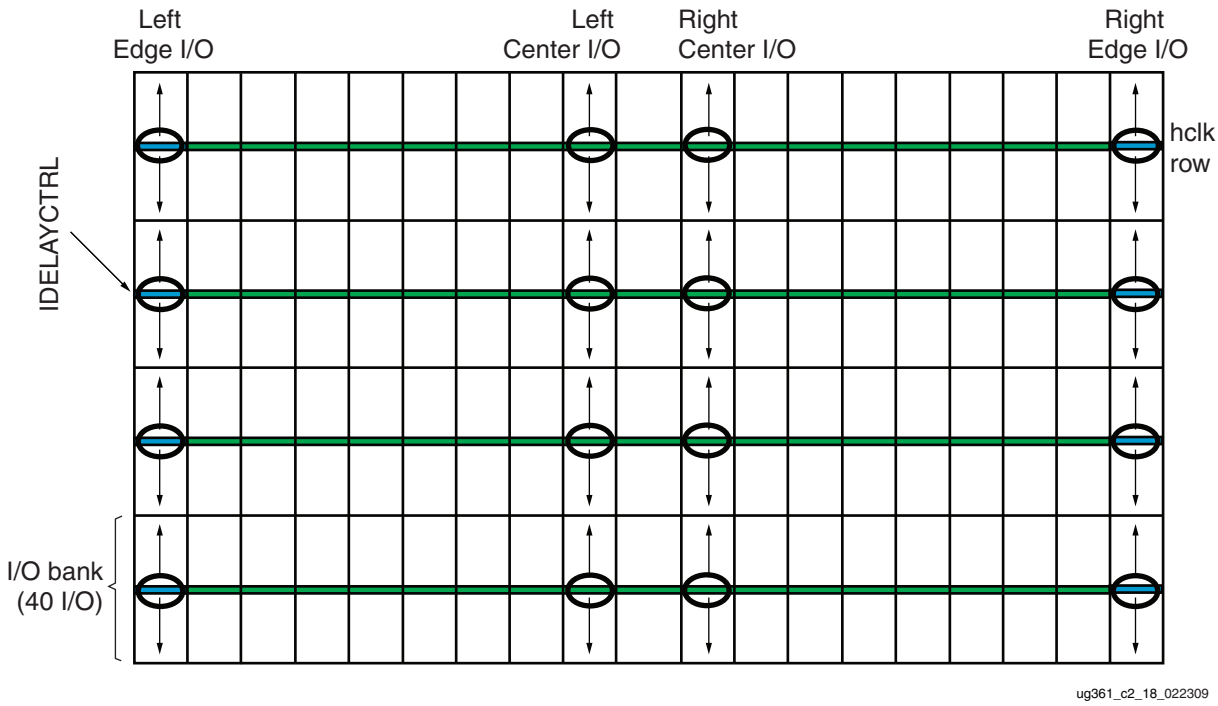


Figure 2-18: **Relative Locations of IDELAYCTRL Modules**

IDELAYCTRL Usage and Design Guidelines

This section includes how to use the IDELAYCTRL modules, design guidelines, and recommended usage in Virtex-6 devices.

Instantiating IDELAYCTRL Without LOC Constraints

When instantiating IDELAYCTRL without LOC constraints, instantiate only *one* instance of IDELAYCTRL in the HDL design code. The implementation tools auto-replicate IDELAYCTRL instances throughout the entire device. When IDELAYCTRL instances are replicated to clock regions but not used, the extra instances are trimmed out of the design automatically by the software.

The signals connected to the RST and REFCLK input ports of the instantiated IDELAYCTRL instance are connected to the corresponding input ports of the replicated IDELAYCTRL instances. There are two special cases:

- When the RDY port is ignored, the RDY signals of all the replacement IDELAYCTRL instances are left unconnected.
- The VHDL and Verilog use models for instantiating an IDELAYCTRL primitive without LOC constraints leaving the RDY output port unconnected are provided in the Libraries Guide. The resulting circuitry after instantiating the IDELAYCTRL components is illustrated in Figure 2-19.

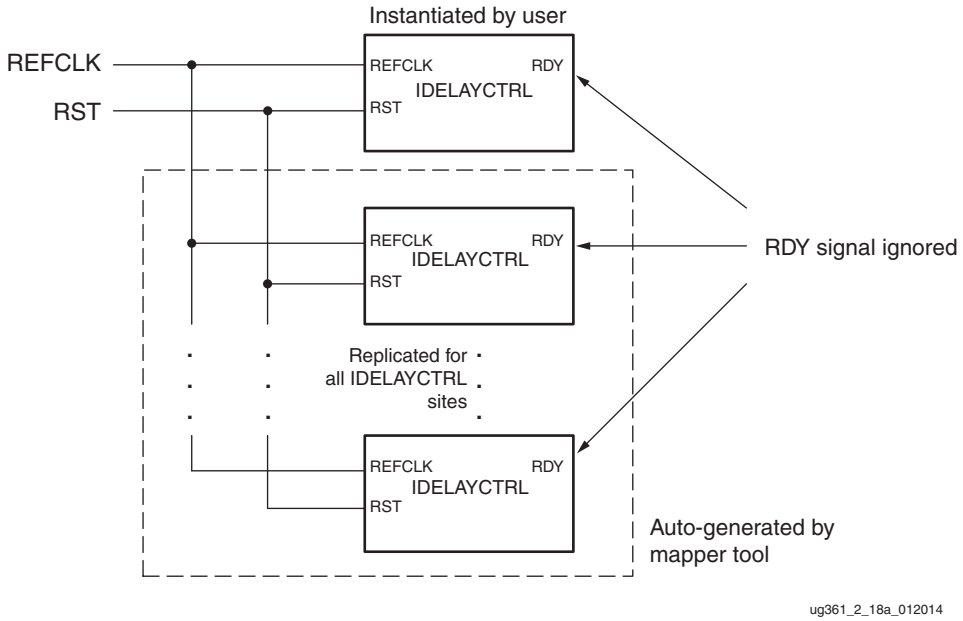


Figure 2-19: **Instantiate IDELAYCTRL Without LOC Constraints - RDY Unconnected**

- When the RDY port is connected, an AND gate of width equal to the number of clock regions is instantiated and the RDY output ports from the instantiated and replicated IDELAYCTRL instances are connected to the inputs of the AND gate. The tools assign the signal name connected to the RDY port of the instantiated IDELAYCTRL instance to the output of the AND gate. The VHDL and Verilog use models for instantiating an IDELAYCTRL primitive without LOC constraints with the RDY port connected are provided in the Libraries Guide. The resulting circuitry after instantiating the IDELAYCTRL components is illustrated in Figure 2-20.

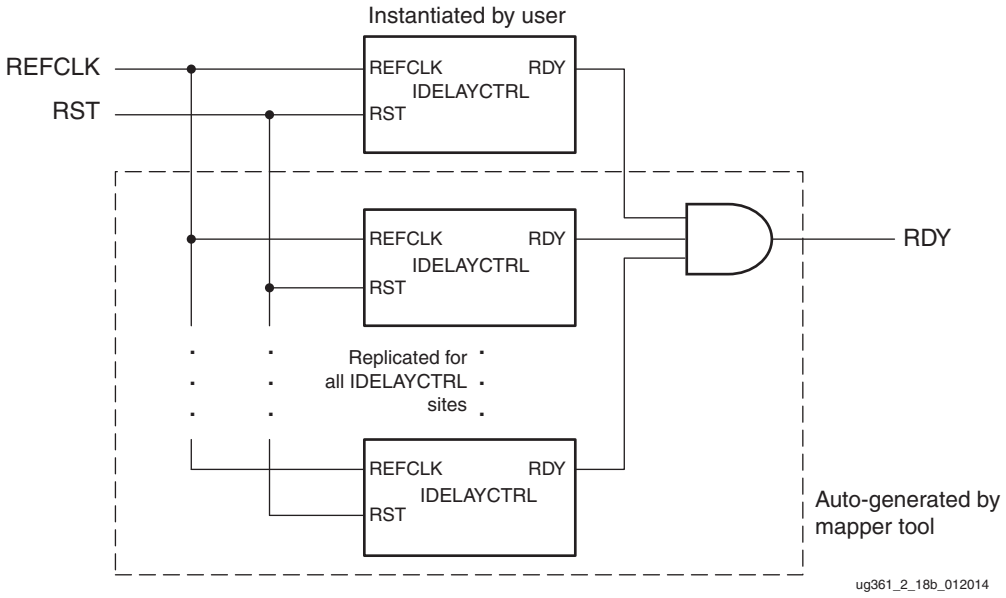


Figure 2-20: **Instantiate IDELAYCTRL Without LOC Constraints - RDY Connected**

Instantiating IDELAYCTRL with LOC Constraints

The most efficient way to use the IDELAYCTRL module is to define and lock down the placement of every IDELAYCTRL instance used in a design. This is done by instantiating the IDELAYCTRL instances with LOC constraints. The user must define and lock placement of all ISERDES and IDELAY components using the delay element. (IDELAY_TYPE attribute set to FIXED or VARIABLE).

Once completed, IDELAYCTRL sites can be chosen and LOC constraints assigned. Xilinx strongly recommends using IDELAYCTRL with a LOC constraint. When not using an IDELAY (with IDELAY_TYPE in FIXED or VARIABLE mode), do not assign a LOC constraint to the IDELAYCTRL for that clock region.

Location Constraints

Each IDELAYCTRL module has XY location coordinates (IDELAYCTRL_XnYm, where X:row_n, Y:column_m). To constrain placement, IDELAYCTRL instances can have LOC properties attached to them. The naming convention for IDELAYCTRL placement coordinates is different from the convention used in naming CLB locations. This allows LOC properties to transfer easily from array to array. See the Constraints User Guide for information on applying LOC constraints to specific components.

Instantiating IDELAYCTRL With and Without LOC Constraints

In some cases, an IDELAYCTRL module with a LOC constraint is instantiated and an IDELAYCTRL module without a LOC constraint is also instantiated. When an IP core is instantiated with a non-location constrained IDELAYCTRL module and an IDELAYCTRL module with a LOC constraint is instantiated for another part of the design, the implementation tools perform the following:

- Instantiate the LOC IDELAYCTRL instances as described in Instantiating IDELAYCTRL with LOC Constraints.
- Replicate the non-location constrained IDELAYCTRL instance to populate with an IDELAYCTRL instance in every clock region without a location constrained IDELAYCTRL instance in place.
- Connect the signals connected to the RST and REFCLK input ports of the non-location constrained IDELAYCTRL instance to the corresponding input ports of the replicated IDELAYCTRL instances.
- If the RDY port of the non-location constrained IDELAYCTRL instance is ignored, then all the RDY signals of the replicated IDELAYCTRL instances are also ignored.
- If the RDY port of the non-location constrained IDELAYCTRL instance is connected, then the RDY port of the non-location constrained instance plus the RDY ports of the replicated instances are connected to an auto-generated AND gate. The implementation tools assign the signal name connected to the RDY port of the non-location constrained instance to the output of the AND gate.
- All the ports of the location constrained instances (RST, REFCLK, and RDY) are independent from each other and from the replicated instances.

The VHDL and Verilog use models for instantiating a mixed usage model are provided in the Libraries Guide. In the example, a non-location constrained IDELAYCTRL instance is instantiated with the RDY signal connected. This discussion is also valid when the RDY signal is ignored. The circuitry that results from instantiating the IDELAYCTRL components is illustrated in Figure 2-21.

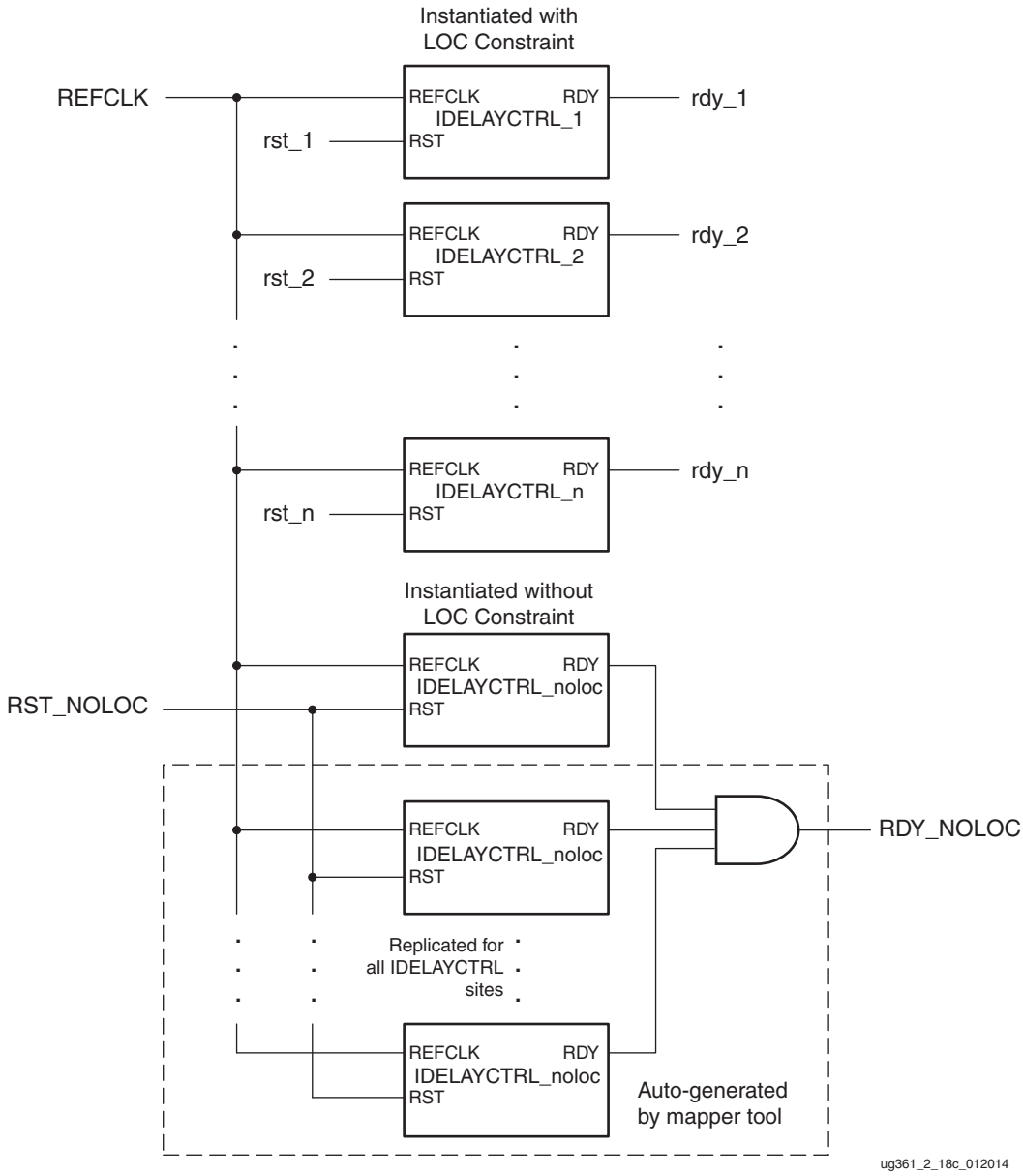


Figure 2-21: **Instantiate IDELAYCTRL Without LOC Constraints**

OLOGIC Resources

OLOGIC consists of two major blocks, one to configure the output data path and the other to configure the 3-state control path. These two blocks have a common clock (CLK) but different enable signals, OCE and TCE. Both have asynchronous and synchronous set and reset (SR signal) controlled by an independent SRVAL attribute.

The output and the 3-state paths can be independently configured in one of the following modes.

- Edge triggered D type flip-flop

- DDR mode as ODDR component (SAME_EDGE or OPPOSITE_EDGE)
- Level sensitive latch
- Asynchronous/combinatorial output path

Figure 2-22 illustrates the various logic resources in the OLOGIC block.

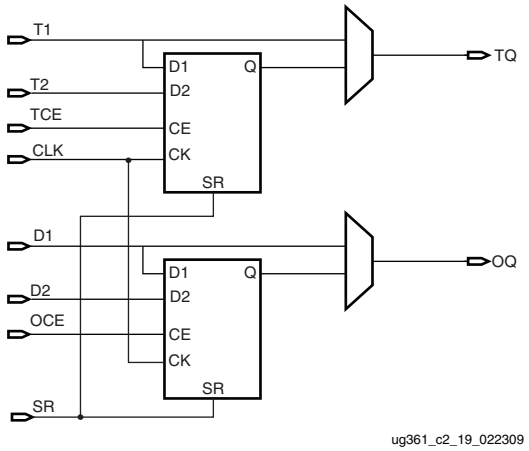


Figure 2-22: **OLOGIC Block Diagram**

This section of the documentation discusses the various features available using the OLOGIC resources. All connections between the OLOGIC resources are managed in Xilinx software.

Combinatorial Output Data and 3-State Control Path

The combinatorial output paths create a direct connection from the FPGA logic to the output driver or output driver control. These paths is used when:

1. There is direct (unregistered) connection from logic resources in the FPGA logic to the output data or 3-state control.
2. The "pack I/O register/latches into IOBs" is set to OFF.

Output DDR Overview (ODDR)

Virtex-6 devices have dedicated registers in the OLOGIC to implement output DDR registers. This feature is accessed when instantiating the ODDR primitive. DDR multiplexing is automatic when using OLOGIC. No manual control of the mux-select is needed. This control is generated from the clock.

There is only one clock input to the ODDR primitive. Falling edge data is clocked by a locally inverted version of the input clock. All clocks feeding into the I/O tile are fully multiplexed, i.e., there is no clock sharing between ILOGIC or OLOGIC blocks. The ODDR primitive supports the following modes of operation:

- OPPOSITE_EDGE mode
- SAME_EDGE mode

The SAME_EDGE mode is the same as for the Virtex-5 architecture. This mode allows designers to present both data inputs to the ODDR primitive on the rising-edge of the ODDR clock, saving CLB and clock resources, and increasing performance. This mode is

implemented using the DDR_CLK_EDGE attribute. It is supported for 3-state control as well. The following sections describe each of the modes in detail.

OPPOSITE_EDGE Mode

In OPPOSITE_EDGE mode, both the edges of the clock (CLK) are used to capture the data from the FPGA logic at twice the throughput. This structure is similar to the Virtex-5 FPGA implementation. Both outputs are presented to the data input or 3-state control input of the IOB. The timing diagram of the output DDR using the OPPPOSITE_EDGE mode is shown in Figure 2-23.

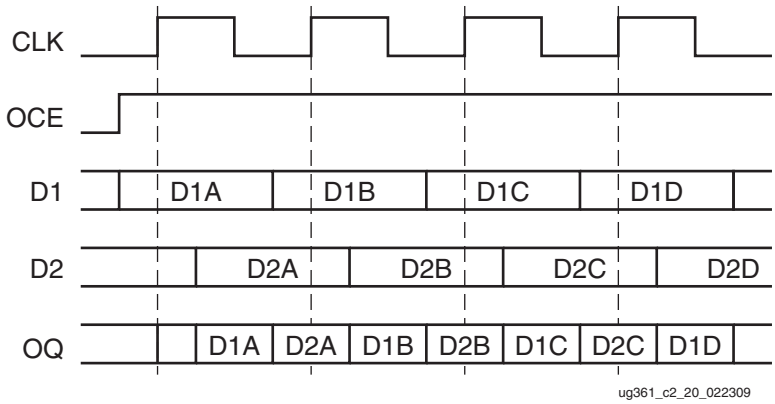


Figure 2-23: **Output DDR Timing in OPPOSITE_EDGE Mode**

SAME_EDGE Mode

In SAME_EDGE mode, data can be presented to the IOB on the same clock edge. Presenting the data to the IOB on the same clock edge avoids setup time violations and allows the user to perform higher DDR frequency with minimal register to register delay, as opposed to using the CLB registers. Figure 2-24 shows the timing diagram of the output DDR using the SAME_EDGE mode.

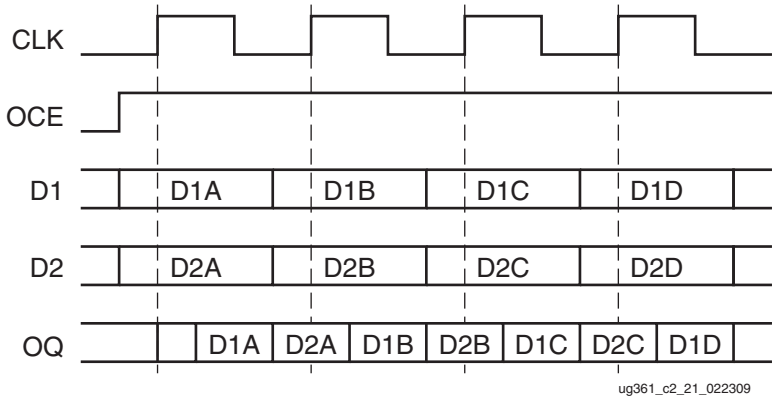


Figure 2-24: **Output DDR Timing in SAME_EDGE Mode**

Clock Forwarding

Output DDR can forward a copy of the clock to the output. This is useful for propagating a clock and DDR data with identical delays, and for multiple clock generation, where every

clock load has a unique clock driver. This is accomplished by tying the D1 input of the ODDR primitive High, and the D2 input Low. Xilinx recommends using this scheme to forward clocks from the FPGA logic to the output pins.

Output DDR Primitive (ODDR)

Figure 2-25 shows the ODDR primitive block diagram. Set and Reset are not supported at the same time. Table 2-15 lists the ODDR port signals. Table 2-16 describes the various attributes available and default values for the ODDR primitive.

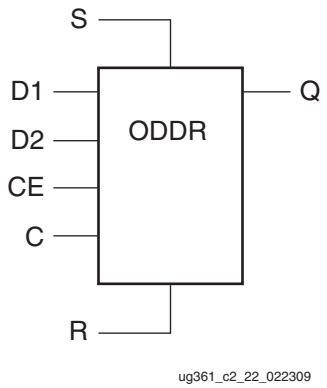


Figure 2-25: **ODDR Primitive Block Diagram**

Table 2-15: **ODDR Port Signals**

Port Name	Function	Description
Q	Data output (DDR)	ODDR register output.
C	Clock input port	The CLK pin represents the clock input pin.
CE	Clock enable port	CE represents the clock enable pin. When asserted Low, this port disables the output clock on port Q.
D1 and D2	Data inputs	ODDR register inputs.
R	Reset	Synchronous/Asynchronous reset pin. Reset is asserted High. Not supported when using Set.
S	Set	Synchronous/Asynchronous set pin. Set is asserted High. Not supported when using Reset.

Table 2-16: **ODDR Attributes**

Attribute Name	Description	Possible Values
DDR_CLK_EDGE	Sets the ODDR mode of operation with respect to clock edge	OPPOSITE_EDGE (default), SAME_EDGE
INIT	Sets the initial value for Q port	0 (default), 1
SRTYPE	Set/Reset type with respect to clock (C)	ASYNC, SYNC (default)

ODDR VHDL and Verilog Templates

The Libraries Guide includes templates for instantiation of the ODDR module in VHDL and Verilog.

OLOGIC Timing Models

This section discusses all timing models associated with the OLOGIC block. Table 2-17 describes the function and control signals of the OLOGIC switching characteristics in the *Virtex-6 FPGA Data Sheet*.

Table 2-17: **OLOGIC Switching Characteristics**

Symbol	Description
Setup/Hold	
T_{ODCK}/T_{OCKD}	D1/D2 pins Setup/Hold with respect to CLK
T_{OOCECK}/T_{OCKOCE}	OCE pin Setup/Hold with respect to CLK
T_{OSRCK}/T_{OCKSR}	SR pin Setup/Hold with respect to CLK
T_{OTCK}/T_{OCKT}	T1/T2 pins Setup/Hold with respect to CLK
T_{OTCECK}/T_{OCKTCE}	TCE pin Setup/Hold with respect to CLK
Clock to Out	
T_{OCKQ}	CLK to OQ/TQ out
T_{RQ}	SR pin to OQ/TQ out

Timing Characteristics

Figure 2-26 illustrates the OLOGIC output register timing.

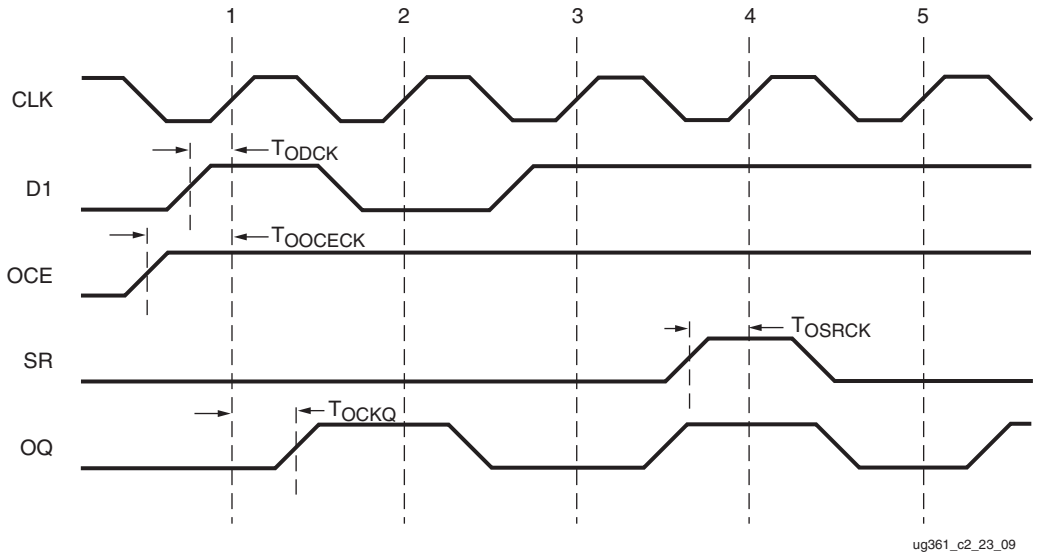


Figure 2-26: **OLOGIC Output Register Timing Characteristics**

Clock Event 1

- At time T_{OOCECK} before Clock Event 1, the output clock enable signal becomes valid-high at the OCE input of the output register, enabling the output register for incoming data.
- At time T_{ODCK} before Clock Event 1, the output signal becomes valid-high at the D1 input of the output register and is reflected at the OQ output at time T_{OCKQ} after Clock Event 1.

Clock Event 4

At time T_{OSRCK} before Clock Event 4, the SR signal (configured as synchronous reset in this case) becomes valid-high, resetting the output register and reflected at the OQ output at time T_{RQ} after Clock Event 4.

Figure 2-27 illustrates the OLOGIC ODDR register timing.

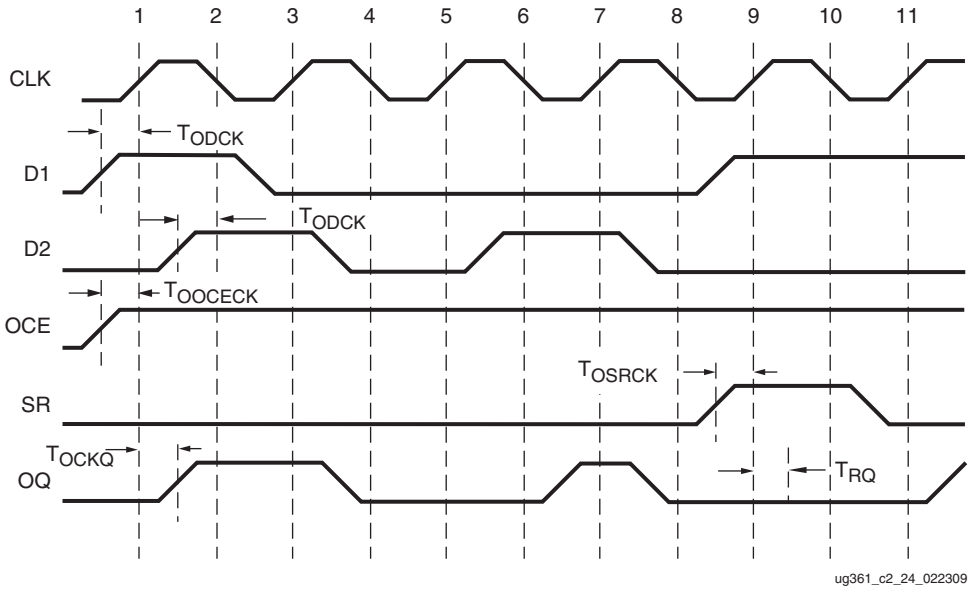


Figure 2-27: **OLOGIC ODDR Register Timing Characteristics**

Clock Event 1

- At time T_{OOCECK} before Clock Event 1, the ODDR clock enable signal becomes valid-High at the OCE input of the ODDR, enabling ODDR for incoming data. Care must be taken to toggle the OCE signal of the ODDR register between the rising edges and falling edges of CLK as well as meeting the register setup-time relative to both clock edges.
- At time T_{ODCK} before Clock Event 1 (rising edge of CLK), the data signal D1 becomes valid-high at the D1 input of ODDR register and is reflected on the OQ output at time T_{OCKQ} after Clock Event 1.

Clock Event 2

- At time T_{ODCK} before Clock Event 2 (falling edge of CLK), the data signal D2 becomes valid-high at the D2 input of ODDR register and is reflected on the OQ output at time T_{OCKQ} after Clock Event 2 (no change at the OQ output in this case).

Clock Event 9

At time T_{OSRCK} before Clock Event 9 (rising edge of CLK), the SR signal (configured as synchronous reset in this case) becomes valid-high resetting ODDR register, reflected at the OQ output at time T_{RQ} after Clock Event 9 (no change at the OQ output in this case) and resetting ODDR register, reflected at the OQ output at time T_{RQ} after Clock Event 10 (no change at the OQ output in this case).

Figure 2-28 illustrates the OLOGIC 3-state register timing.

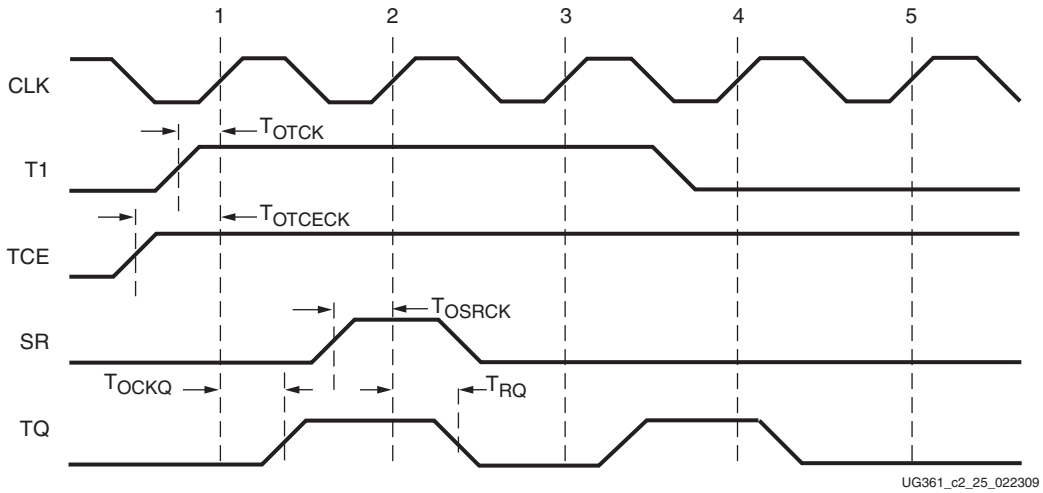


Figure 2-28: **OLOGIC 3-State Register Timing Characteristics**

Clock Event 1

- At time T_{OTCECK} before Clock Event 1, the 3-state clock enable signal becomes valid-high at the TCE input of the 3-state register, enabling the 3-state register for incoming data.
- At time T_{OTCK} before Clock Event 1 the 3-state signal becomes valid-high at the T input of the 3-state register, returning the pad to high-impedance at time T_{OCKQ} after Clock Event 1.

Clock Event 2

- At time T_{OSRCK} before Clock Event 2, the SR signal (configured as synchronous reset in this case) becomes valid-high, resetting the 3-state register at time T_{RQ} after Clock Event 2.

Figure 2-29 illustrates IOB DDR 3-state register timing. This example is shown using DDR in opposite edge mode. For other modes add the appropriate latencies as shown in Figure 2-4, page 94.

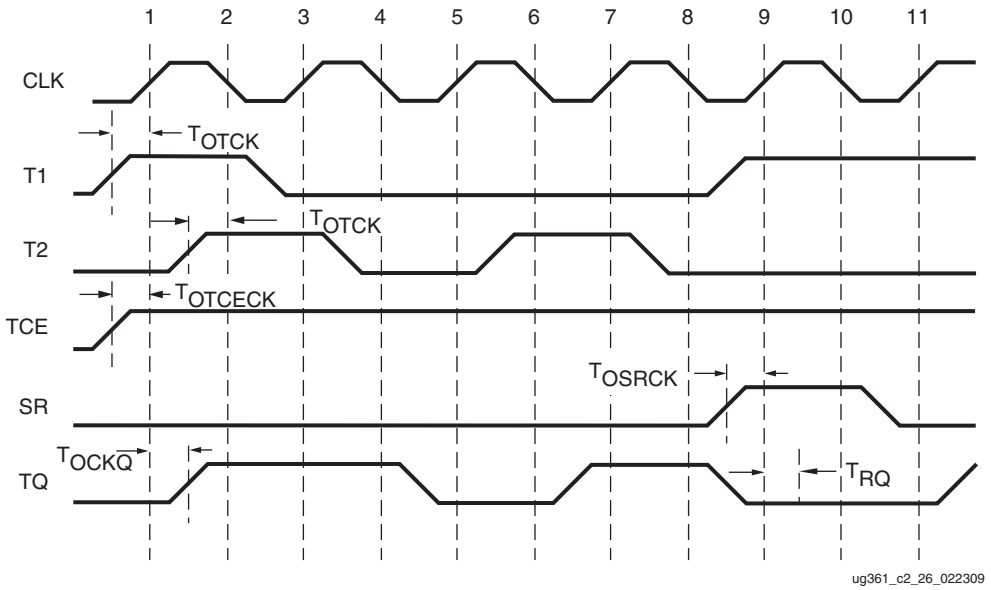


Figure 2-29: **OLOGIC ODDR 3-State Register Timing Characteristics**

Clock Event 1

- At time T_{OTCECK} before Clock Event 1, the 3-state clock enable signal becomes valid-High at the TCE input of the 3-state ODDR register, enabling them for incoming data. Care must be taken to toggle the TCE signal of the 3-state ODDR between the rising edges and falling edges of CLK as well as meeting the register setup-time relative to both clock edges.
- At time T_{OTCK} before Clock Event 1 (rising edge of CLK), the 3-state signal T1 becomes valid-high at the T1 input of 3-state register and is reflected on the TQ output at time T_{OCKQ} after Clock Event 1.

Clock Event 2

- At time T_{OTCK} before Clock Event 2 (falling edge of CLK), the 3-state signal T2 becomes valid-high at the T2 input of 3-state register and is reflected on the TQ output at time T_{OCKQ} after Clock Event 2 (no change at the TQ output in this case).

Clock Event 9

- At time T_{OSRCK} before Clock Event 9 (rising edge of CLK), the SR signal (configured as synchronous reset in this case) becomes valid-high resetting 3-state Register, reflected at the TQ output at time T_{RQ} after Clock Event 9 (no change at the TQ output in this case) and resetting 3-state Register, reflected at the TQ output at time T_{RQ} after Clock Event 10 (no change at the TQ output in this case).

XILINX®

Chapter 3

Advanced SelectIO Logic Resources

Introduction

The I/O functionality in Virtex-6 FPGAs is described in Chapter 1 through Chapter 3 of this user guide.

- Chapter 1 covers the electrical characteristics of input receivers and output drivers, and their compliance with many industry standards.
- Chapter 2 describes the register structures dedicated for sending and receiving SDR or DDR data.
- This chapter covers additional resources:
  - Input serial-to-parallel converters (ISERDES) and output parallel-to-serial converters (OSERDES) support very fast I/O data rates, and allow the internal logic to run up to 10 times slower than the I/O.
  - The Bitslip submodule can re-align data to word boundaries, detected with the help of a training pattern.

Input Serial-to-Parallel Logic Resources (ISERDES)

The ISERDES in Virtex-6 FPGAs is a dedicated serial-to-parallel converter with specific clocking and logic features designed to facilitate the implementation of high-speed source-synchronous applications. The ISERDES avoids the additional timing complexities encountered when designing deserializers in the FPGA fabric.

ISERDES features include:

- Dedicated Deserializer/Serial-to-Parallel Converter

  The ISERDES deserializer enables high-speed data transfer without requiring the FPGA fabric to match the input data frequency. This converter supports both single data rate (SDR) and double data rate (DDR) modes. In SDR mode, the serial-to-parallel converter creates a 2-, 3-, 4-, 5-, 6-, 7-, or 8-bit wide parallel word. In DDR mode, the serial-to-parallel converter creates a 4-, 6-, 8-, or 10-bit-wide parallel word.

- Bitslip Submodule

  The Bitslip submodule allows designers to reorder the sequence of the parallel data stream going into the FPGA fabric. This can be used for training source-synchronous interfaces that include a training pattern.

- Dedicated Support for Strobe-based Memory Interfaces

  ISERDES contains dedicated circuitry (including the OCLK input pin) to handle the strobe-to-FPGA clock domain crossover entirely within the ISERDES block. This allows for higher performance and a simplified implementation.

- Dedicated Support for Networking Interfaces
- Dedicated Support for DDR3 Interfaces
- Dedicated Support for QDR Interfaces

Figure 3-1 shows the block diagram of the ISERDES, highlighting all the major components and features of the block including the optional inverters.

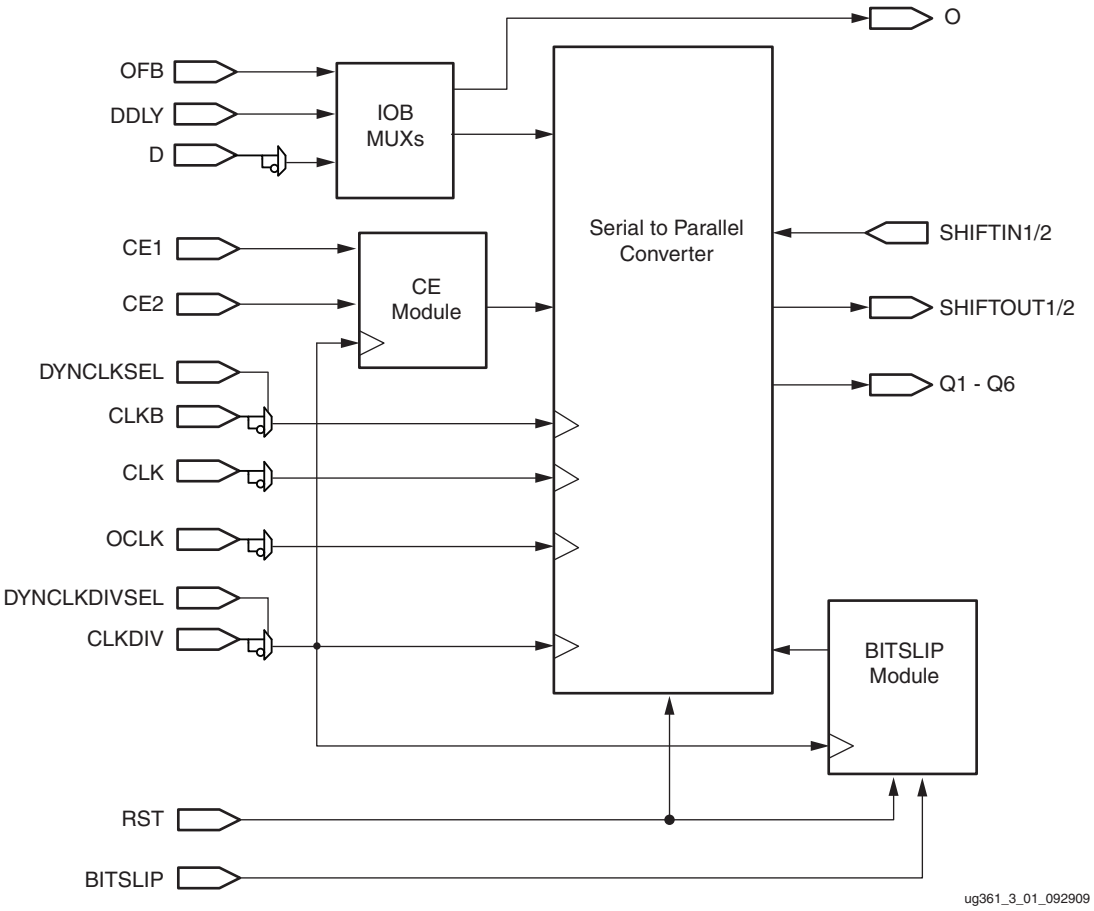


Figure 3-1: **ISERDES Block Diagram**

ISERDES Primitive (ISERDESE1)

The ISERDES primitive in Virtex-6 devices (shown in Figure 3-2) is ISERDESE1.

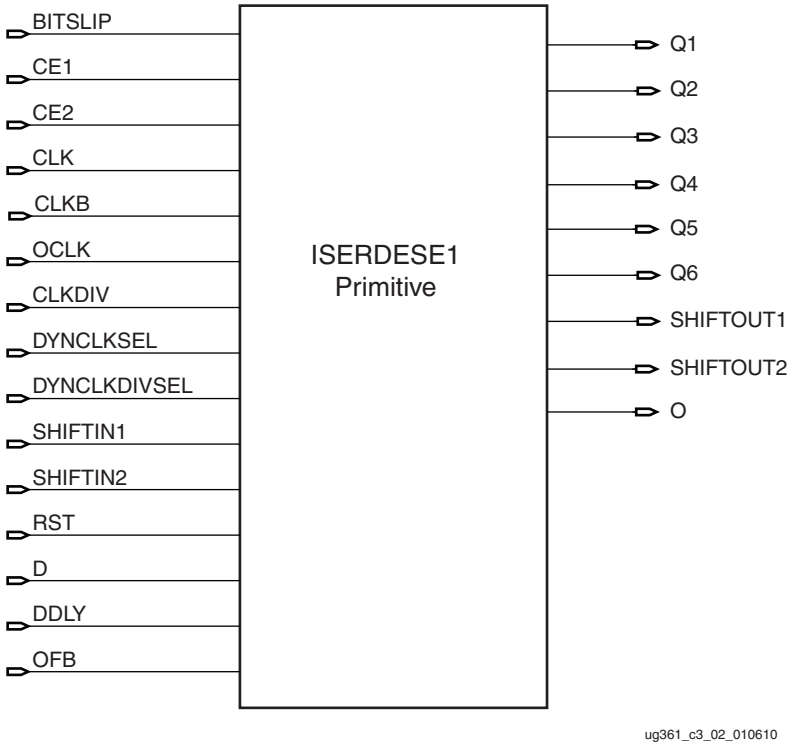


Figure 3-2: **ISERDESE1 Primitive**

Table 3-1 lists the available ports in the ISERDESE1 primitive.

Table 3-1: **ISERDESE1 Port List and Definitions**

Port Name	Type	Width	Description
Q1 – Q6	Output	1 (each)	Registered outputs. See Registered Outputs – Q1 to Q6.
O	Output	1	Combintorial output. See Combinatorial Output – O.
SHIFTOUT1	Output	1	Carry out for data width expansion. Connect to SHIFTIN1 of slave IOB. See ISERDES Width Expansion.
SHIFTOUT2	Output	1	Carry out for data width expansion. Connect to SHIFTIN2 of slave IOB. See ISERDES Width Expansion.
D	Input	1	Serial input data from IOB. See Serial Input Data from IOB - D.
DDLY	Input	1	Serial input data from IODELAY. See Serial Input Data from IODELAYE1 - DDLY.
CLK	Input	1	High-speed clock input. Clocks serial input data stream. See High-Speed Clock Input - CLK.
CLKB	Input	1	Second High speed clock input only for MEMORY_QDR mode. Always connect to inverted CLK unless in MEMORY_QDR mode. See MEMORY_QDR Interface Type.
CE1, CE2	Input	1 (each)	Clock enable inputs. See Clock Enable Inputs - CE1 and CE2.

Table 3-1: **ISERDESE1 Port List and Definitions *(Cont'd)***

Port Name	Type	Width	Description
RST	Input	1	Active High reset. See Reset Input - RST.
CLKDIV	Input	1	Divided clock input. Clocks delay element, deserialized data, Bitslip submodule, and CE unit. See Divided Clock Input - CLKDIV.
OCLK	Input	1	High-speed clock input for memory applications. See High-Speed Clock for Strobe-Based Memory Interfaces - OCLK
BITSLIP	Input	1	Invokes the Bitslip operation. See Bitslip Operation - BITSLIP.
SHIFTIN1	Input	1	Carry input for data width expansion. Connect to SHIFTOUT1 of master IOB. See ISERDES Width Expansion.
SHIFTIN2	Input	1	Carry input for data width expansion. Connect to SHIFTOUT2 of master IOB. See ISERDES Width Expansion.
OFB	Input	1	Feedback Path from the OLOGIC/OSERDES output. This is referred to as *digital loopback* because the loopback signal from the OLOGIC/OSERDES does not use any output or input buffer.
DYNCLKDIVSEL	Input	1	Dynamically select CLKDIV inversion. See Dynamic Clock Inversions.
DYNCLKSEL	Input	1	Dynamically select CLK and CLKB inversion. See Dynamic Clock Inversions.

ISERDESE1 Ports

Registered Outputs – Q1 to Q6

The output ports Q1 to Q6 are the registered outputs of the ISERDESE1 module. One ISERDESE1 block can support up to six bits (i.e., a 1:6 deserialization). Bit widths greater than six (up to 10) can be supported. See ISERDES Width Expansion. The first data bit received appears on the highest order Q output.

The bit ordering at the input of an OSERDES is the opposite of the bit ordering at the output of an ISERDESE1 block, as shown in Figure 3-3. For example, the least significant bit A of the word FEDCBA is placed at the D1 input of an OSERDESE1, but the same bit A emerges from the ISERDESE1 block at the Q6 output. In other words, D1 is the least significant input to the OSERDESE1, while Q6 is the least significant output of the ISERDESE1 block. When width expansion is used, D1 of the master OSERDESE1 is the least significant input, while Q4 of the slave ISERDESE1 block is the least significant output.

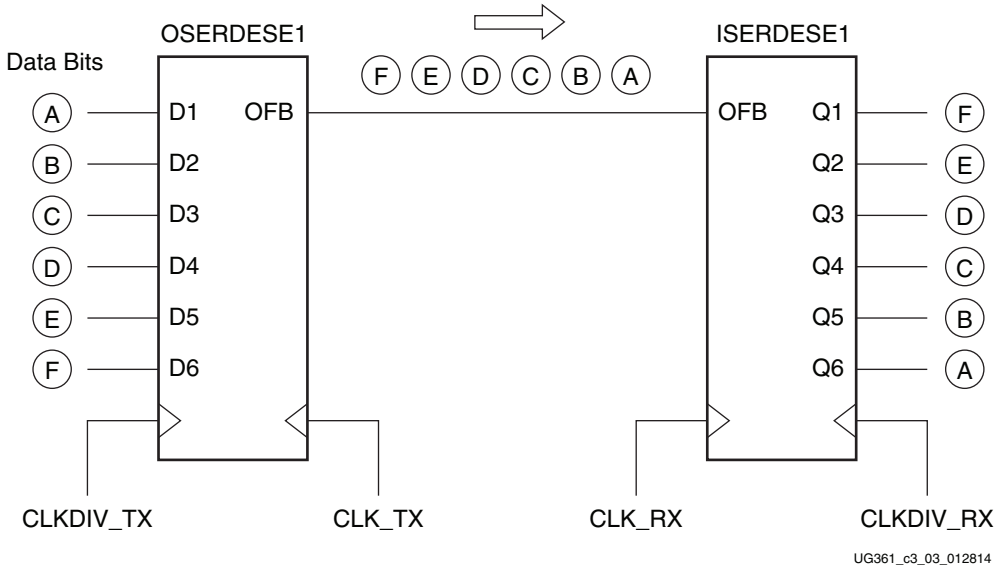


Figure 3-3: **Bit Ordering on Q1–Q6 Outputs of ISERDESE1 Ports**

Combinatorial Output – O

The combinatorial output port (O) is an unregistered output of the ISERDESE1 module. This output can come directly from the data input (D), or from the data input (DDLY) via the IODELAYE1.

Bitslip Operation - BITSLIP

The BITSLIP pin performs a Bitslip operation synchronous to CLKDIV when asserted (active High). Subsequently, the data seen on the Q1 to Q6 output ports will shift, as in a barrel-shifter operation, one position every time Bitslip is invoked (DDR operation is different from SDR). See BITSLIP Submodule for more details.

Clock Enable Inputs - CE1 and CE2

Each ISERDESE1 block contains an input clock enable module (Figure 3-4).

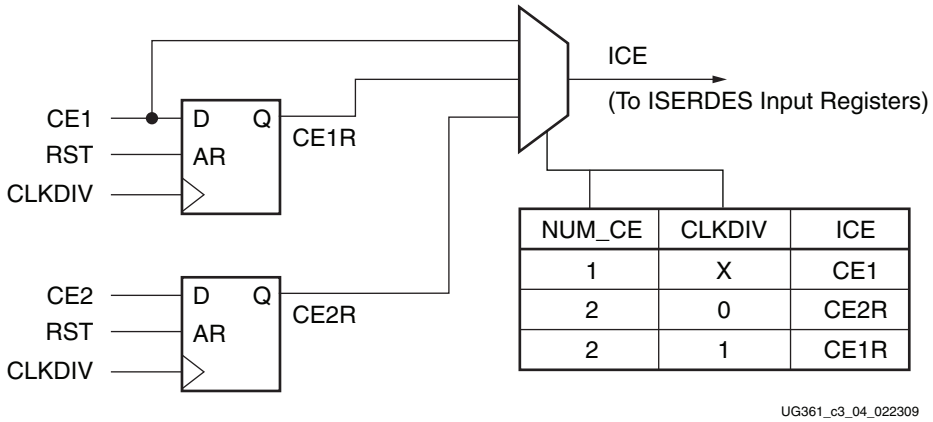


Figure 3-4: **Input Clock Enable Module**

When NUM_CE = 1, the CE2 input is not used, and the CE1 input is an active High clock enable connected directly to the input registers in the ISERDESE1. When NUM_CE = 2, the CE1 and CE2 inputs are both used, with CE1 enabling the ISERDESE1 for ½ of a CLKDIV cycle, and CE2 enabling the ISERDESE1 for the other ½. The internal clock enable signal ICE shown in Figure 3-4 is derived from the CE1 and CE2 inputs. ICE drives the clock enable inputs of registers FF0, FF1, FF2, and FF3 shown in Figure 3-5, page 138. The remaining registers in Figure 3-5, page 138 do not have clock enable inputs.

The clock enable module functions as a 2:1 serial-to-parallel converter, clocked by CLKDIV. The clock enable module is needed specifically for bidirectional memory interfaces when ISERDESE1 is configured for 1:4 deserialization in DDR mode. When the attribute NUM_CE = 2, the clock enable module is enabled and both CE1 and CE2 ports are available. When NUM_CE = 1, only CE1 is available and functions as a regular clock enable.

High-Speed Clock Input - CLK

The high-speed clock input (CLK) is used to clock in the input serial data stream.

High-Speed Clock Input - CLKB

The high-speed secondary clock input (CLKB) is used to clock in the input serial data stream. In any mode other than MEMORY_QDR, connect CLKB to an inverted version of CLK. In MEMORY_QDR mode CLKB should be connected to a unique, phase shifted clock. See ISERDESE1 Clocking Methods.

Divided Clock Input - CLKDIV

The divided clock input (CLKDIV) is typically a divided version of CLK (depending on the width of the implemented deserialization). It drives the output of the serial-to-parallel converter, the Bitslip submodule, and the CE module.

Serial Input Data from IOB - D

The serial input data port (D) is the serial (high-speed) data input port of the ISERDESE1. This port works in conjunction only with the Virtex-6 FPGA IOB resource. See IOBDELAY Attribute.

Serial Input Data from IODELAYE1 - DDLY

The serial input data port (DDLY) is the serial (high-speed) data input port of the ISERDESE1. This port works in conjunction only with the Virtex-6 FPGA IODELAY resource. See IOBDELAY Attribute.

Serial Input Data from OSERDESE1 - OFB

The serial input data port (OFB) is the serial (high-speed) data input port of the ISERDESE1. This port works in conjunction only with the Virtex-6 FPGA OSERDESE1 port OFB.

High-Speed Clock for Strobe-Based Memory Interfaces - OCLK

The OCLK clock input synchronizes data transfer in strobe-based memory interfaces. The OCLK clock is only used when INTERFACE_TYPE is set to MEMORY.

The OCLK clock input is used to transfer strobe-based memory data onto a free-running clock domain. OCLK is a free-running FPGA clock at the same frequency as the strobe on the CLK input. The domain transfer from CLK to OCLK is shown in the Figure 3-5 block diagram. The timing of the domain transfer is set by the user by adjusting the delay of the strobe signal to the CLK input (e.g., using IDELAY). Examples of setting the timing of this domain transfer are given in the Memory Interface Generator (MIG). When INTERFACE_TYPE is NETWORKING, this port is unused. The OCLK of the ISERDES_NODELAY shares the same routing as the CLK port of the OSERDES.

Reset Input - RST

When asserted, the reset input causes the outputs of most internal ISERDES data flip-flops in the CLK and CLKDIV domains to be driven Low asynchronously. The exceptions are the first four flip-flops in the input structure whose value after reset is selectable via attributes on the component.

RST must be deasserted synchronously with CLKDIV. The deassertion of reset releases most flip-flops in the ISERDESE1 except for some internal control logic that retimes this reset deassertion to the first rising edge of CLKDIV followed by a consecutive rising edge of CLK.

- After the release of reset, serial data is loaded in a serial-to-parallel input register at the CLK rate.
- The first rising CLKDIV edge, after releasing reset, releases the first stage of an internal retiming circuit. The CLKDIV edge data is available at the Q[6:1] pins of the ISERDESE1.
- After the release of reset and after a first rising CLKDIV edge, the first rising CLK edge releases a state machine that generates a clock pulse to load the captured and parallelized data to a temporary register. Data from this temporary register is passed to the outputs Q[6:1} at the CLKDIV rate. The timing of this is determined by the DATA_WIDTH attribute.

As a result, after releasing reset, the ISERDESE1 can capture several bits and these bits are passed to the ISERDESE1 outputs with CLKDIV. Consequently, the first output value of an ISERDESE1 can be ignored. The reset timing of multiple ISERDESE1 ports is shown in Figure 3-13, page 147.

Every ISERDESE1 in a multiple bit input structure should be driven by the same reset signal, asserted asynchronously, and deasserted synchronously to CLKDIV. The best practices for ensuring that all ISERDESE1 components rise out of reset simultaneously are:

- Wait until all MMCM/PLL used in the design are locked.
  - Reset should only be deasserted when it is known that CLK and CLKDIV are stable and present.
- Wait until all IDELAYCTRL components, when used, show RDY high.
- Release reset synchronously to, and delayed by, a couple of CLKDIV cycles.
  - A reset pulse should be a minimum of two CLKDIV cycles and a maximum of 32 CLKDIV cycles.
- Use on the reset net one flip-flop clocked by CLKDIV per ISERDESE1/OSERDESE1 pair.
  - Put the flip-flop in the FPGA logic in front of the ISERDESE1/OSERDESE1 pair.
  - Put a timing constraint on the flip-flop to ISERDESE1/OSERDESE1 of one CLKDIV period or less.

The reset timing of multiple OSERDESE1 ports is shown in Figure 3-24, page 167.

ISERDESE1 Attributes

Table 3-2 summarizes all the applicable ISERDESE1 attributes. A detailed description of each attribute follows the table. For more information on applying these attributes in UCF, VHDL, or Verilog code, refer to the Xilinx ISE Software Manual.

Table 3-2: **ISERDESE1 Attributes**

Attribute Name	Description	Value	Default Value
DATA_RATE	Enables incoming data stream to be processed as SDR or DDR data. See DATA_RATE Attribute.	String: SDR or DDR	DDR
DATA_WIDTH	Defines the width of the serial-to-parallel converter. The legal value depends on the DATA_RATE attribute (SDR or DDR). See DATA_WIDTH Attribute.	Integer: 2, 3, 4, 5, 6, 7, 8, or 10. If DATA_RATE = DDR, value is limited to 4, 6, 8, or 10. If DATA_RATE = SDR, value is limited to 2, 3, 4, 5, 6, 7, or 8.	4
DYN_CLKDIV_INV_EN	Enables DYNCLKDIVINVSEL inversion when TRUE and disables HDL inversions on CLKDIV pin. See Dynamic Clock Inversions.	Boolean: TRUE or FALSE	FALSE
DYN_CLK_INV_EN	Enables DYNCLKINVSEL inversion when TRUE and disables HDL inversions on CLK and CLKB pins. See Dynamic Clock Inversions.	Boolean: TRUE or FALSE	FALSE
INTERFACE_TYPE	Chooses the ISERDESE1 use model. See INTERFACE_TYPE Attribute.	String: MEMORY, MEMORY_DDR3, MEMORY_QDR, OVERSAMPLE, or NETWORKING	MEMORY
IOBDELAY	Defines how an IODELAY is connected and used together with the ISERDES. See IOBDELAY Attribute.	String: NONE, IBUF, IFD, or BOTH	NONE
NUM_CE	Defines the number of clock enables. See NUM_CE Attribute.	Integer: 1 or 2	2
OFB_USED	Enables the path from the OLOGIC, OSERDES OFB pin to the ISERDES OFB pin. Disables the use of the D input pin.	Boolean: TRUE or FALSE	FALSE
SERDES_MODE	Defines whether the ISERDESE1 module is a master or slave when using width expansion. See SERDES_MODE Attribute.	String: MASTER or SLAVE	MASTER
INIT_Q1	Sets initial value for first sample register	0 or 1	0
INIT_Q2	Sets initial value for second sample register	0 or 1	0
INIT_Q3	Sets initial value for third sample register	0 or 1	0
INIT_Q4	Sets initial value for fourth sample register	0 or 1	0

Table 3-2: **ISERDESE1 Attributes *(Cont'd)***

Attribute Name	Description	Value	Default Value
SRVAL_Q1	Sets value after reset of first sample register	0 or 1	1
SRVAL_Q2	Sets value after reset of second sample register	0 or 1	1
SRVAL_Q3	Sets value after reset of third sample register	0 or 1	1
SRVAL_Q4	Sets value after reset of fourth sample register	0 or 1	1

DATA_RATE Attribute

The DATA_RATE attribute defines whether the incoming data stream is processed as single data rate (SDR) or double data rate (DDR). The allowed values for this attribute are SDR and DDR. The default value is DDR.

DATA_WIDTH Attribute

The DATA_WIDTH attribute defines the parallel data output width of the serial-to-parallel converter. The possible values for this attribute depend on the INTERFACE_TYPE and DATA_RATE attributes. See Table 3-3 for supported data widths.

Table 3-3: **Supported Data Widths**

INTERFACE_TYPE	DATA_RATE	Supported Data Widths
NETWORKING	SDR	2, 3, 4, 5, 6, 7, 8
	DDR	4, 6, 8, 10
MEMORY MEMORY_DDR3 MEMORY_QDR	SDR	None
	DDR	4

When the DATA_WIDTH is set to widths larger than six, a pair of ISERDESE1 must be configured into a master-slave configuration. See ISERDES Width Expansion. Width expansion is not allowed in memory mode.

INTERFACE_TYPE Attribute

The INTERFACE_TYPE attribute determines whether the ISERDESE1 is configured in memory or networking mode. The allowed values for this attribute are MEMORY, MEMORY_DDR3, MEMORY_QDR, OVERSAMPLE, or NETWORKING. The default mode is MEMORY.

Figure 3-5 illustrates the ISERDESE1 internal connections when in MEMORY mode.

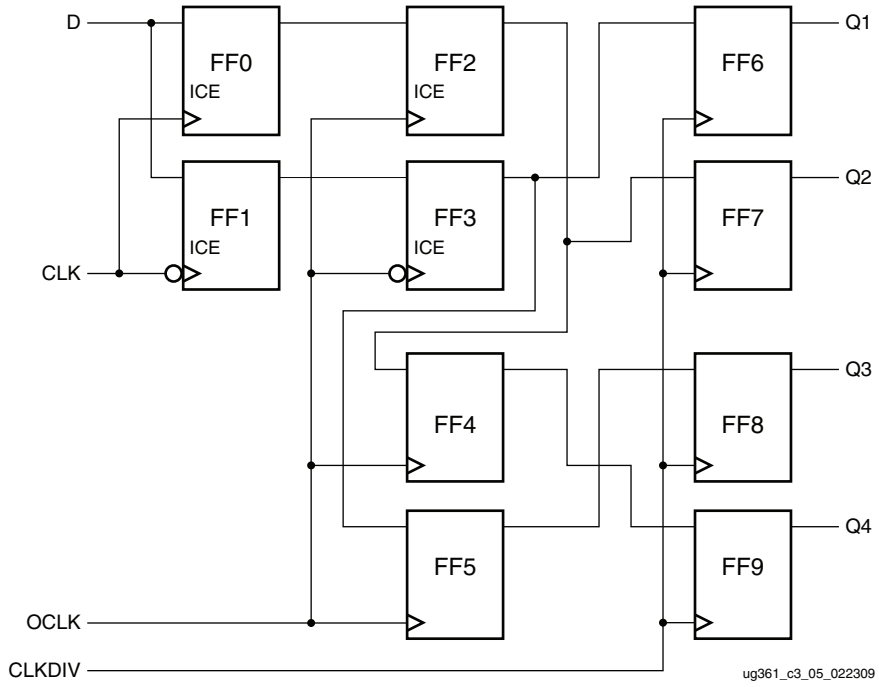


Figure 3-5: **Internal Connections of ISERDESE1 When in MEMORY Mode**

IOBDELAY Attribute

The IOBDELAY attribute defines how the output of an IOBDELAY is used in the ISERDESE1 (see Figure 3-6).

- When an IOBDELAY is used with an ISERDESE1 component, the output of the IOBDELAY must be connected to the ISERDESE1.DDLY pin.
- When IOBDELAY is set to NONE, the IOBDELAY is not used as input delay (IDELAY). The ISERDESE1.D pin must be used.
- When IOBDELAY is set to IBUF, the IOBDELAY used as IDELAY is passing through the ISERDESE1 as a combinatorial signal. The ISERDES does not use the IDELAY and the delayed data or clock is sent out via the ISERDESE1.O output pin.
- When IOBDELAY is set to IFD, the IOBDELAY used as IDELAY connects to the input serial-to-parallel register of the ISREDESE1. The outputs of the ISERDESE1 are the Q1 to Q6 pins.
- When the IOBDELAY is set to BOTH, the output from the IOBDELAY used as IDELAY passes through the ISERDESE1 as a combinatorial signal (output ISERDESE1.O) and, at the same time, it is connected to the input serial-to-parallel register of the ISERDESE1 (the outputs are ISERDESE1.Q1-Q6).

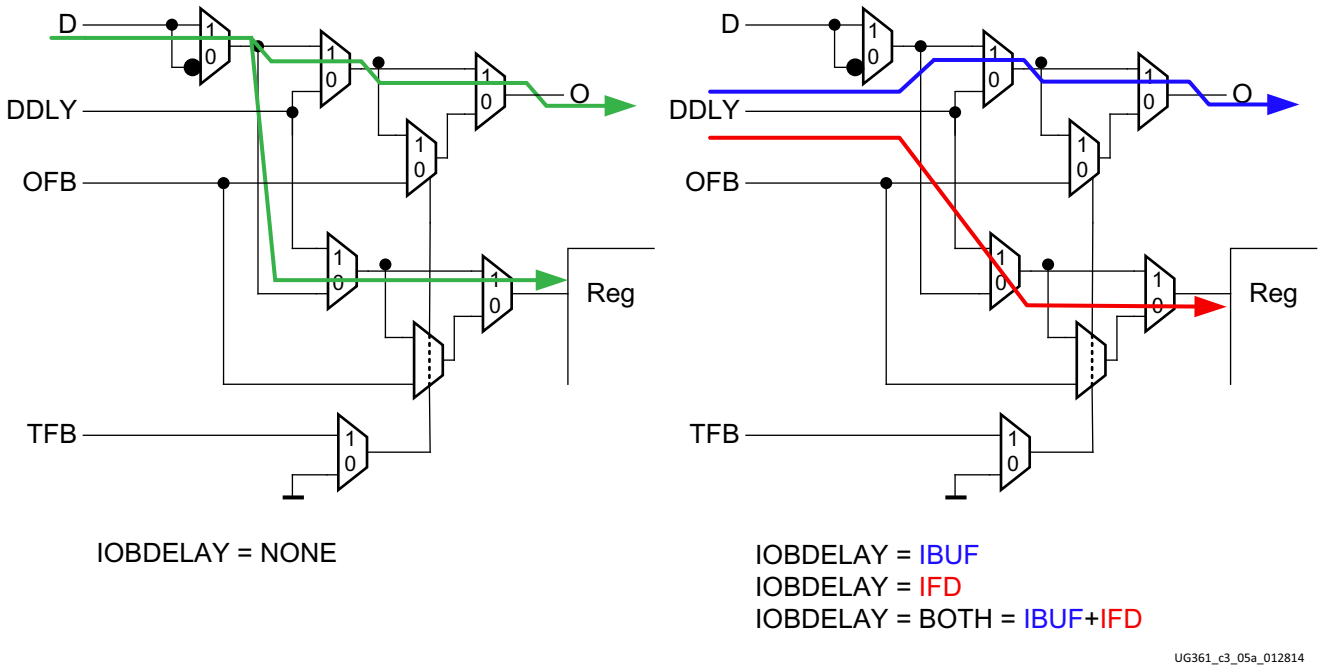


Figure 3-6: **IOBDELAY Configurations for ISERDESE1**

NUM_CE Attribute

The NUM_CE attribute defines the number of clock enables (CE1 and CE2) used. The possible values are 1 and 2 (default = 2).

OFB_USED_Attribute

OFB_USED set to TRUE enables the OFB port in the ISERDESE1 and OSERDESE1 and allows a route from the OSERDESE1.OFB output to the ISERDESE1.OFB input. Data can now be transmitted from the OSERDESE1 back to the ISERDESE1 (see Figure 3-7). This feature is enabled when the OSERDESE1 and ISERDESE1 have the same DATA_RATE and DATA_WIDTH setting for the feedback. When using the ISERDESE1 and OSERDESE1 in width expansion mode only, connect the master OSERDESE1 to the master ISERDESE1. When using the ISERDESE1 as a feedback port, it cannot be used as an input for external data.

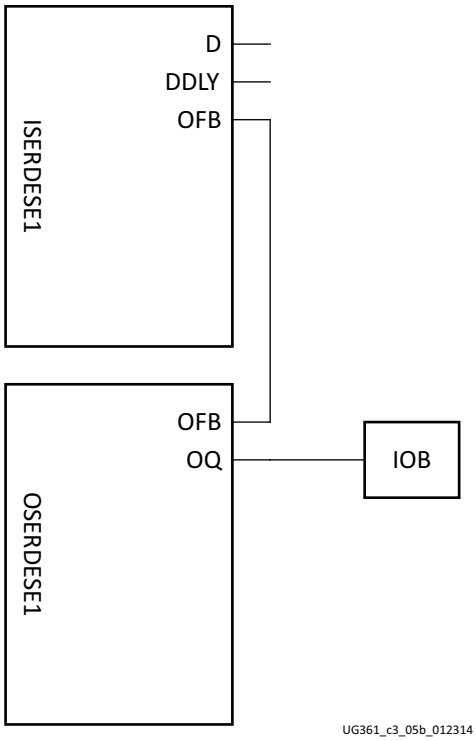


Figure 3-7: **OSERDESE1 Looped Back to ISERDESE1 via OFB Port**

SERDES_MODE Attribute

The SERDES_MODE attribute defines whether the ISERDESE1 module is a master or slave when using width expansion. The possible values are MASTER and SLAVE. The default value is MASTER. See ISERDES Width Expansion.

ISERDESE1 Clocking Methods

NETWORKING Interface Type

The phase relationship of CLK and CLKDIV is important in the serial-to-parallel conversion process. CLK and CLKDIV are (ideally) phase-aligned within a tolerance. There are several clocking arrangements within the FPGA to help the design meet the phase relationship requirements of CLK and CLKDIV.

The CLK and CLKDIV inputs must be nominally phase-aligned. For example, if CLK and CLKDIV in Figure 3-8 were inverted by the designer at the ISERDES inputs, then although the clocking arrangement is an allowed BUFIO/BUFR configuration, the clocks would still be out of phase. This also prohibits using DYNCLKINVSEL and DYNCLKDIVINVSEL.

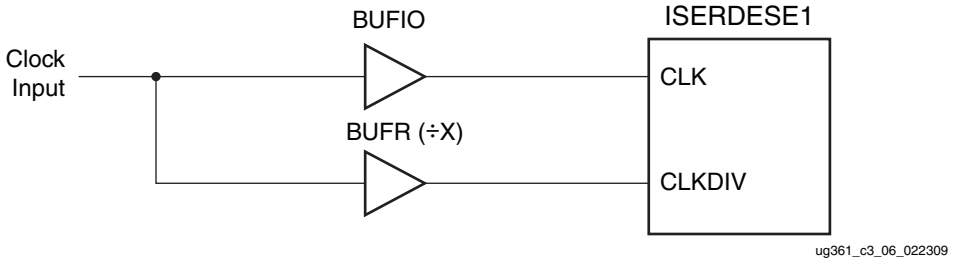


Figure 3-8: **Clocking Arrangement Using BUFIO and BUFR**

The only valid clocking arrangements for the ISERDESE1 block using the networking interface type are:

- CLK driven by BUFIO, CLKDIV driven by BUFR
- CLK driven by MMCM, CLKDIV driven by CLKOUT[0:6] of same MMCM
- CLK driven by BUFG, CLKDIV driven by another BUFG

When using a MMCM to drive the CLK and CLKDIV of the ISERDESE1, the buffer types suppling the ISERDESE1 can not be mixed. For example, if CLK is driven by a BUFG, then CLKDIV must be driven by a BUFG as well.

MEMORY Interface Type

The only valid clocking arrangements for the ISERDESE1 block using the memory interface type are:

- CLK driven by BUFIO, OCLK driven by BUFIO, or CLKDIV driven by BUFR
- CLK driven by MMCM, OCLK driven by MMCM, or CLKDIV driven by CLKOUT[0:6] of same MMCM
- CLK driven by BUFG, CLKDIV driven by another BUFG

The OCLK and CLKDIV inputs must be nominally phase-aligned. No phase relationship between CLK and OCLK is expected. Calibration must be performed for reliable data transfer from CLK to OCLK domain. High-Speed Clock for Strobe-Based Memory Interfaces - OCLK gives further information about transferring data between CLK and OCLK.

MEMORY_QDR Interface Type

The only valid clocking arrangements for the ISERDESE1 block using the MEMORY_QDR interface type are:

- CLK driven by BUFIO
- CLKB driven by BUFIO in same region as CLK
- CLKDIV driven from BUFR from same source as CLK

The clocking arrangement shown in Figure 3-9 is the only valid clocking scheme for using the ISERDESE1 in MEMORY_QDR mode. This INTERFACE_TYPE attribute setting is only supported when using the MIG tool. The QDR memory forwards a clock (CQ) and clock bar (CQ#) with the data (Q0). To maintain this phase relationship, the dedicated BUFIO resources must be used. An example of this clocking method is given in Figure 3-9.

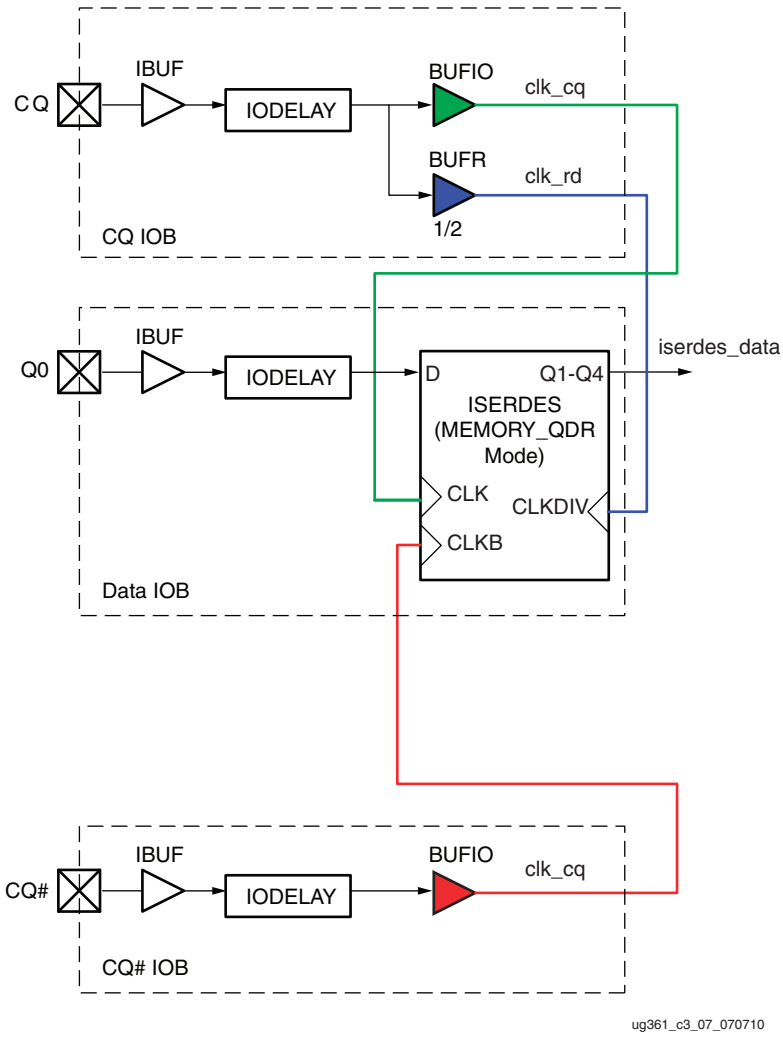


Figure 3-9: **Clocking Method for ISERDESE1 in MEMORY_QDR Mode**

OVERSAMPLE Interface Type

The OVERSAMPLE mode is used to capture two phases of DDR data. Figure 3-10 shows a more detailed logical representation of the ISERDESE1 and how data is captured on both

the rising and falling edge of CLK and OCLK. As shown in Figure 3-10, there must be a 90°offset phase relationship between CLK and OCLK as the data is captured on both CLK and OCLK but is clocked out of the ISERDESE1 on the CLK domain. CLKDIV is not used in this mode. The only valid clocking arrangements for the OVERSAMPLE interface type are:

- CLK and CLKB are driven by a BUFIO. OCLKB is driven by a BUFIO that is phase shifted by 90°. The BUFIOs are driven from a single MMCM.
- CLK and CLKB are driven by a BUFG. OCLKB is driven by a BUFG that is phase shifted by 90°. The BUFGs are driven from a single MMCM.

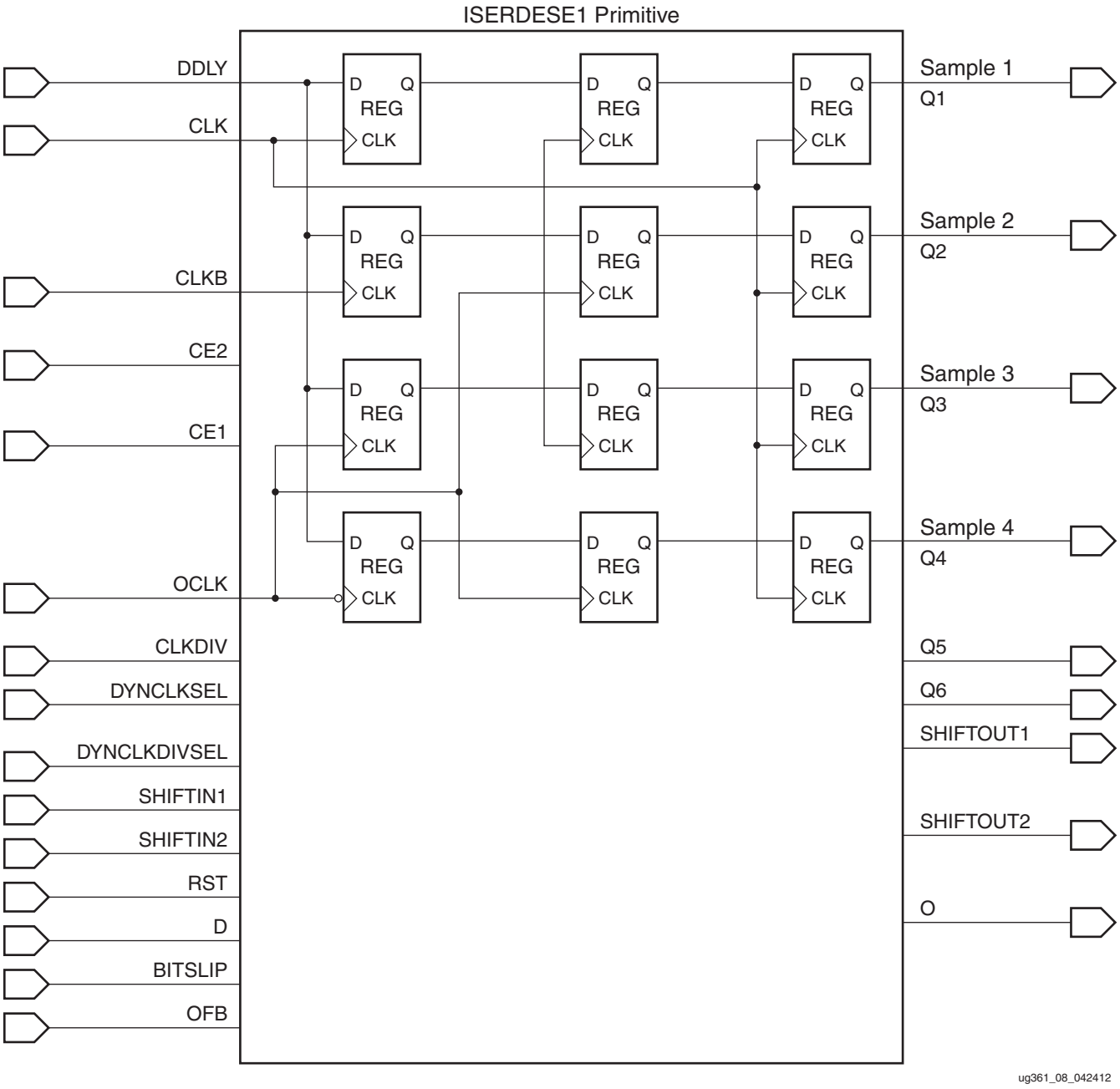


Figure 3-10: **Logical View of ISERDESE1 Primitive in Oversample Mode**

MEMORY_DDR3 Interface Type

The MEMORY_DDR3 mode has a complex clocking structure as a result of the DDR3 memory requirements. This INTERFACE_TYPE attribute setting is only supported when using the MIG tool. The only valid clocking arrangements for the ISERDESE1 block using the MEMORY_DDR3 interface type are:

- CLK driven by BUFIO or BUFG MMCM combination
- CLK driven by MMCM, CLKDIV driven by CLKOUT[0:6] of same MMCM

ISERDES Width Expansion

Two ISERDES modules are used to build a serial-to-parallel converter larger than 1:6. In every I/O tile there are two ISERDES modules; one master and one slave. By connecting the SHIFTOUT ports of the master ISERDES to the SHIFTIN ports of the slave ISERDES the serial-to-parallel converter can be expanded to up to 1:10 (DDR) and 1:8 (SDR).

Figure 3-11 illustrates a block diagram of a 1:10 DDR serial-to-parallel converter using the master and slave ISERDES modules. Ports Q3 - Q6 are used for the last four bits of the parallel interface on the slave ISERDES.

For a differential input, the master ISERDES must be on the positive side of the differential input pair. When the input is not differential, the input buffer associated with the slave ISERDES is not available and can not be used.

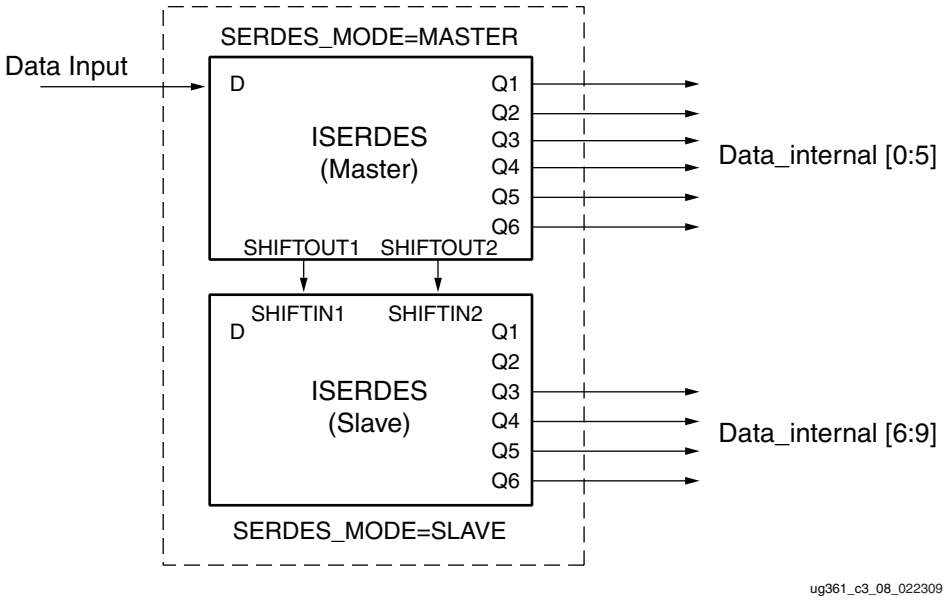


Figure 3-11: **Block Diagram of ISERDES Width Expansion**

Guidelines for Expanding the Serial-to-Parallel Converter Bit Width

1. Both ISERDES modules must be adjacent master and slave pairs. Both ISERDES modules must be in NETWORKING mode because width expansion is not available in MEMORY mode.
2. Set the SERDES_MODE attribute for the master ISERDES to MASTER and the slave ISERDES to SLAVE. See SERDES_MODE Attribute.

3. The user must connect the SHIFTIN ports of the SLAVE to the SHIFTOUT ports of the MASTER.
4. The SLAVE only uses the ports Q3 to Q6 as an output.
5. DATA_WIDTH applies to both MASTER and SLAVE in Figure 3-11.

ISERDES Latencies

When the ISERDES interface type is MEMORY, the latency through the OCLK stage is one CLKDIV cycle. However, the total latency through the ISERDES depends on the phase relationship between the CLK and the OCLK clock inputs. When the ISERDES interface type is NETWORKING, the latency is two CLKDIV cycles. See Figure 3-16, page 150 and Figure 3-17, page 151 for a visualization of latency in networking mode. The extra CLKDIV cycle of latency in networking mode (compared to memory mode) is due to the Bitslip submodule.

The latency in MEMORY_QDR and MEMORY_DDR3 is two CLKDIV cycles.

Dynamic Clock Inversions

The dynamic clock inversion pins DYNCLKSEL and DYNCLKDIVSEL when used in conjunction with DYN_CLK_SEL_EN and DYN_CLKDIV_SEL_EN respectively can enable the user to dynamically switch the polarity of the respective clock source. This operation causes the clock going into ISERDESE1 to switch asynchronously and will likely cause the ISERDESE1 to produce erroneous data until the ISERDESE1 is reset. This operation can only be supported in MEMORY_QDR and MEMORY_DDR3 mode.

ISERDES Timing Model and Parameters

Table 3-4 describes the function and control signals of the ISERDES switching characteristics in the *Virtex-6 FPGA Data Sheet*.

Table 3-4: **ISERDES Switching Characteristics**

Symbol	Description
Setup/Hold for Control Lines	
$T_{ISCCK\_BITSLIP}$/ $T_{ISCKC\_BITSLIP}$	BITSLIP pin Setup/Hold with respect to CLKDIV
T_{ISCCK_CE} /T_{ISCKC_CE}	CE pin Setup/Hold with respect to CLK (for CE1)
T_{ISCCK_CE} /T_{ISCKC_CE}	CE pin Setup/Hold with respect to CLKDIV (for CE2)
Setup/Hold for Data Lines	
T_{ISDCK_D} / T_{ISCKD_D}	D pin Setup/Hold with respect to CLK
T_{ISDCK_DDR} / T_{ISCKD_DDR}	D pin Setup/Hold with respect to CLK at DDR mode
Sequential Delay	
T_{ISCKO_Q}	CLKDIV to Out at Q pins

Timing Characteristics

Figure 3-12 illustrates an ISERDES timing diagram for the input data to the ISERDES. The timing parameter names change for different modes (SDR/DDR). However, the names do not change when a different bus input width, including when two ISERDES are cascaded together to form 10 bits. In DDR mode, the data input (D) switches at every CLK edge (rising and falling).

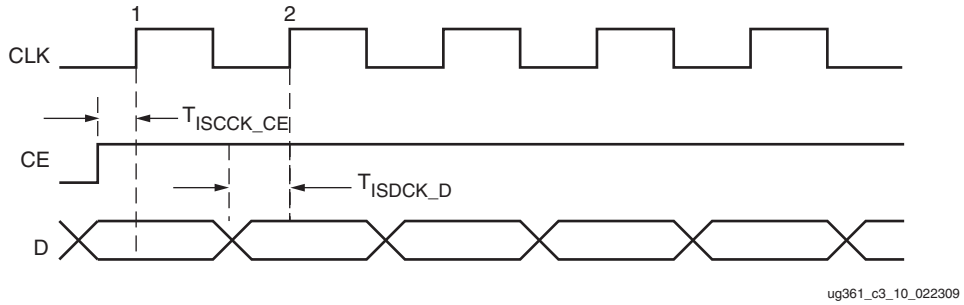


Figure 3-12: **ISERDES Input Data Timing Diagram**

Clock Event 1

- At time T_{ISCCK_CE}, before Clock Event 1, the clock enable signal becomes valid-High and the ISERDES can sample data.

Clock Event 2

- At time T_{ISDCK_D}, before Clock Event 2, the input data pin (D) becomes valid and is sampled at the next positive clock edge.

Reset Input Timing

Clock Event 1

As shown in Figure 3-13, the reset pulse is generated on the rising edge of CLKDIV. Because the pulse must take two different routes to get to ISERDES0 and ISERDES1, there are different propagation delays for both paths. The difference in propagation delay is emphasized. The path to ISERDES0 is very long and the path to ISERDES1 is very short, such that each ISERDES receives the reset pulse in a different CLK cycle. The internal resets for both CLK and CLKDIV are reset asynchronously when the RST input is asserted.

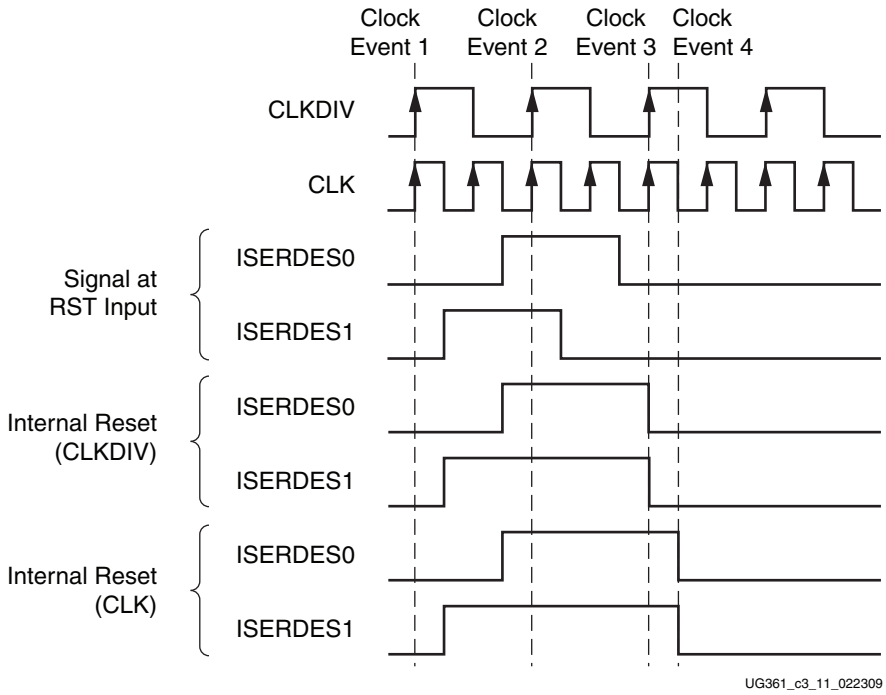


Figure 3-13: **Two ISERDES Coming Out of Reset Synchronously with One Another**

Clock Event 2

The reset pulse is deasserted on the rising edge of CLKDIV. The difference in propagation delay between the two ISERDES causes the RST input to come out of reset on two different CLK cycles. Without internal retiming, ISERDES1 finishes reset one CLK cycle before ISERDES0 and both ISERDES are asynchronous.

Clock Event 3

The release of the reset signal at the RST input is retimed internally to CLKDIV. This synchronizes ISERDES0 and ISERDES1.

Clock Event 4

The release of the reset signal at the RST input is retimed internally to CLK.

ISERDESE1 VHDL and Verilog Instantiation Template

VHDL and Verilog instantiation templates are available in the Libraries Guide for all primitives and submodules.

In VHDL, each template has a component declaration section and an architecture section.

Each part of the template should be inserted within the VHDL design file. The port map of the architecture section should include the design signal names.

BITSLIP Submodule

All ISERDES blocks in Virtex-6 devices contain a Bitslip submodule. This submodule is used for word-alignment purposes in source-synchronous networking-type applications. Bitslip reorders the parallel data in the ISERDES block, allowing every combination of a repeating serial pattern received by the deserializer to be presented to the FPGA fabric. This repeating serial pattern is typically called a training pattern (training patterns are supported by many networking and telecom standards).

Bitslip Operation

By asserting the Bitslip pin of the ISERDES block, the incoming serial data stream is reordered at the parallel side. This operation is repeated until the training pattern is seen. The tables in Figure 3-14 illustrate the effects of a Bitslip operation in SDR and DDR mode. (Bit 8 of an input ISERDESE2 is the first bit received.) For illustrative purposes the data width is eight. The Bitslip operation is synchronous to CLKDIV. In SDR mode, every Bitslip operation causes the output pattern to shift left by one. In DDR mode, every Bitslip operation causes the output pattern to alternate between a shift right by one and shift left by three. In this example, on the eighth Bitslip operation, the output pattern reverts to the initial pattern. This assumes that serial data is an eight bit repeating pattern. Although the repeating pattern seems to show that Bitslip is a barrel shifting operation, this is not the case. A Bitslip operation adds one bit to the input data stream and loses the nth bit.This causes the operation on repetitive patterns to show up as a barrel shifter.

Bitslip Operation in SDR Mode

Bitslip Operations Executed	Output Pattern (8:1)
Initial	10010011
1	00100111
2	01001110
3	10011100
4	00111001
5	01110010
6	11100100
7	11001001

Bitslip Operation in DDR Mode

Bitslip Operations Executed	Output Pattern (8:1)
Initial	00100111
1	10010011
2	10011100
3	01001110
4	01110010
5	00111001
6	11001001
7	11100100

ug361_c3_12_022309

Figure 3-14: **Bitslip Operation Examples**

Figure 3-15 illustrates the ISERDES configured in 1:8 SDR NETWORKING mode. Two ISERDES modules are in a master-slave configuration for a data width of eight.

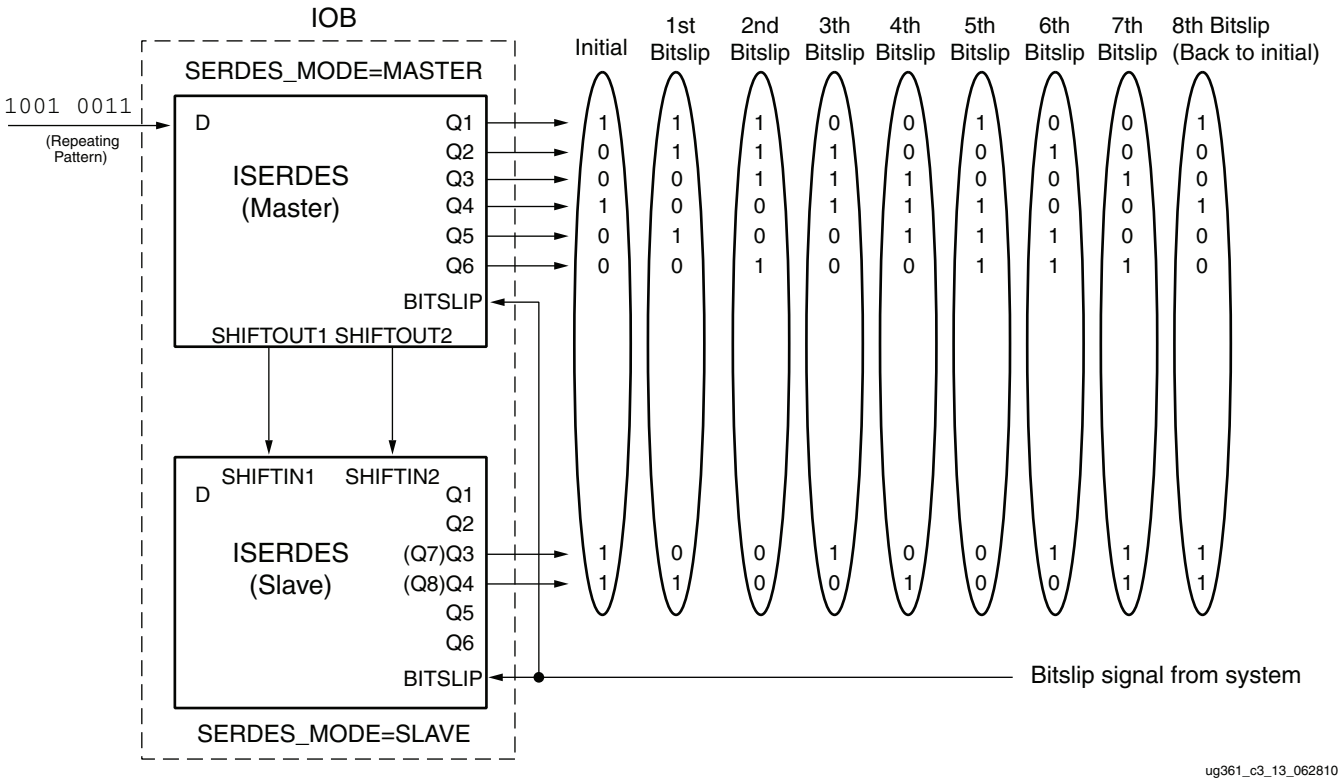


Figure 3-15: **Circuit Diagram for Bitslip Configuration in 1:8 SDR Mode**

Guidelines for Using the Bitslip Submodule

In NETWORKING mode the Bitslip submodule is available. In all other modes, the module is not available.

To invoke a Bitslip operation, the BITSLIP port must be asserted High for one CLKDIV cycle. In SDR mode, Bitslip cannot be asserted for two consecutive CLKDIV cycles; Bitslip must be deasserted for at least one CLKDIV cycle between two Bitslip assertions. In both SDR and DDR mode, the total latency from when the ISERDES captures the asserted Bitslip input to when the "bit-slipped" ISERDES outputs Q1–Q6 are sampled into the FPGA logic by CLKDIV is two CLKDIV cycles.

Bitslip Timing Model and Parameters

This section discusses the timing models associated with the Bitslip controller in a 1:4 DDR configuration. Data (D) is a repeating, 4-bit training pattern ABCD. ABCD could appear at the parallel outputs Q1–Q4 of the ISERDES in four possible ways: ABCD, BCDA, CDAB, and DABC. Only one of these four alignments of the parallel word makes sense to the user's downstream logic that reads the data from the Q1–Q4 outputs of the ISERDES. In this case, ABCD is assumed to be the word alignment that makes sense. Asserting Bitslip allows the user to see all possible configurations of ABCD and then choose the expected alignment (ABCD). Figure 3-16 shows the timing of two Bitslip operations and the corresponding re-alignments of the ISERDES parallel outputs Q1–Q4.

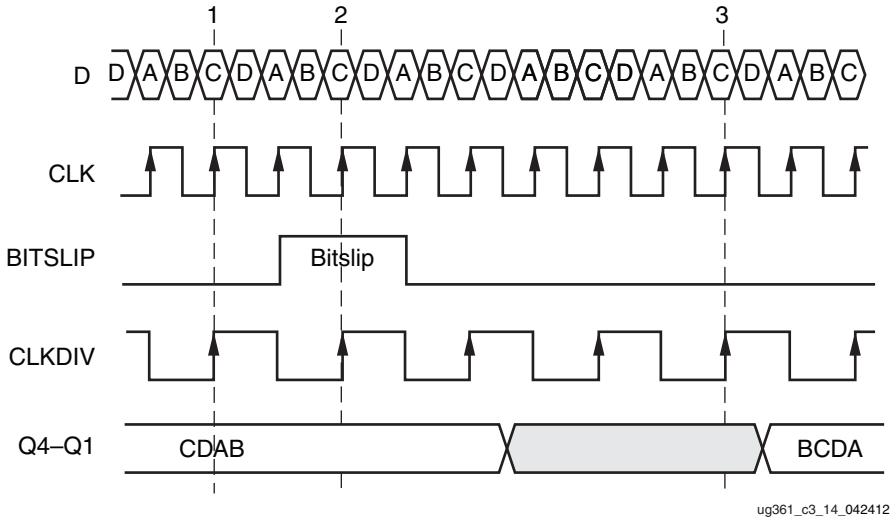


Figure 3-16: **Bitslip Timing Diagram**

Clock Event 1

The entire first word CDAB has been sampled into the input side registers of the ISERDES. The Bitslip pin is not asserted; the word propagates through the ISERDES without any realignment.

Clock Event 2

The second word CDAB has been sampled into the input side registers of the ISERDES. The Bitslip pin is asserted, which causes the Bitslip controller to shift all bits internally by one bit to the right.

Clock Event 3

The third word CDAB has been sampled into the input side registers of the ISERDES. The Bitslip pin is asserted for a second time, which causes the Bitslip controller to shift all bits internally by three bits to the left.

On this same edge of CLKDIV, the first word sampled is presented to Q1–Q4 without any realignment. The actual bits from the input stream that appear at the Q1–Q4 outputs during this cycle are shown in A of Figure 3-17.

Figure 3-17: **Bits from Data Input Stream (D) of Figure 3-16**

Clock Event 4

The first two bits of the fourth word CD have been sampled into the input side registers of the ISERDES. On this same edge of CLKDIV, the second word sampled is presented to Q1–Q4 with one bit shifted to the right. The actual bits from the input stream that appear at the Q1–Q4 outputs during this cycle are shown in B of Figure 3-17.

The realigned bits on Q1–Q4 are sampled into the FPGA logic on the CLKDIV domain. The total latency from when the ISERDES captures the asserted Bitslip input to when the realigned ISERDES outputs Q1–Q4 are sampled by CLKDIV is two CLKDIV cycles.

Clock Event 5

The third word sampled is presented to Q1–Q4 with three bits shifted to the left. The actual bits from the input stream that appear at the Q1–Q4 outputs during this cycle are shown in C of Figure 3-17.

Output Parallel-to-Serial Logic Resources (OSERDES)

The OSERDES in Virtex-6 devices is a dedicated parallel-to-serial converter with specific clocking and logic resources designed to facilitate the implementation of high-speed source-synchronous interfaces. Every OSERDES module includes a dedicated serializer for data and 3-state control. Both data and 3-state serializers can be configured in SDR and DDR mode. Data serialization can be up to 6:1 (10:1 if using OSERDES Width Expansion). 3-state serialization can be up to 4:1. There is a dedicated DDR3 mode to support high-speed memory applications.

Figure 3-18 shows a block diagram of the OSERDES, highlighting all the major components and features of the block.

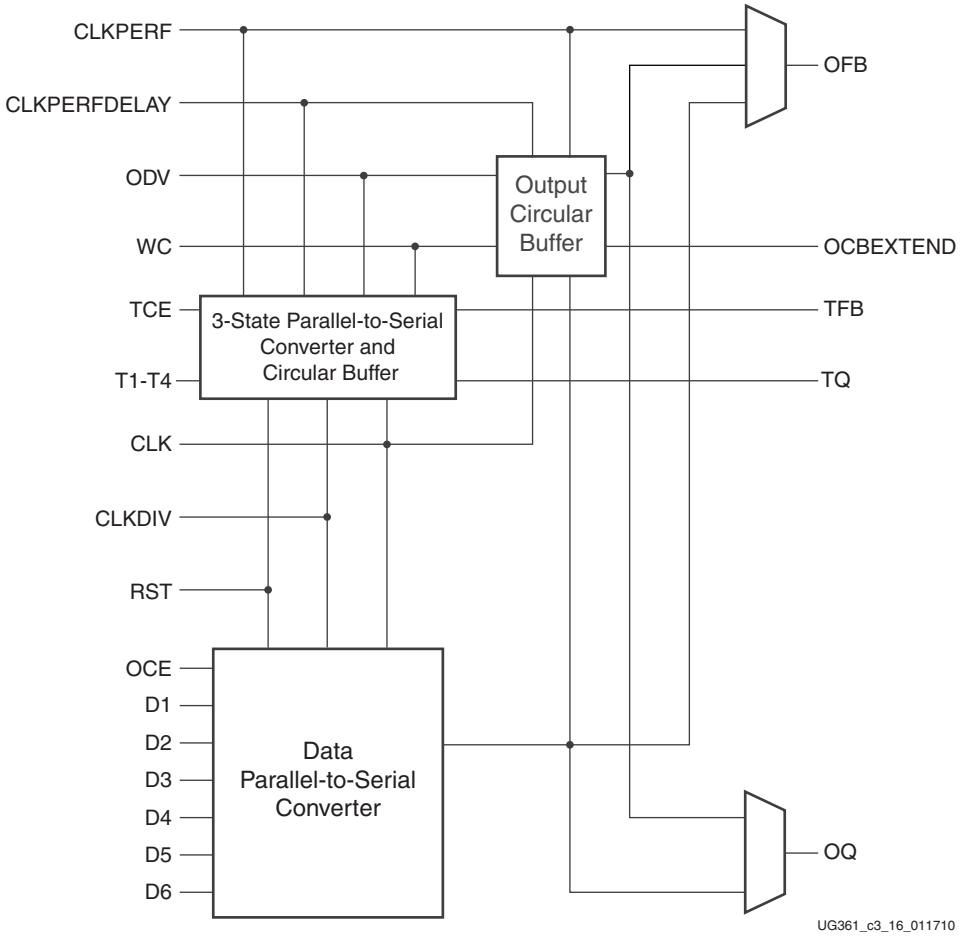


Figure 3-18: **OSERDES Block Diagram**

Data Parallel-to-Serial Converter

The data parallel-to-serial converter in one OSERDES blocks receives two to six bits of parallel data from the fabric (10:1 if using OSERDES Width Expansion), serializes the data, and presents it to the IOB via the OQ outputs. Parallel data is serialized from lowest order data input pin to highest (i.e., data on the D1 input pin is the first bit transmitted at the OQ pins). The data parallel-to-serial converter is available in two modes: single-data rate (SDR) and double-data rate (DDR).

The OSERDES uses two clocks, CLK and CLKDIV, for data rate conversion. CLK is the high-speed serial clock, CLKDIV is the divided parallel clock. CLK and CLKDIV must be phase aligned. See OSERDES Clocking Methods.

Prior to use, a reset must be applied to the OSERDES. The OSERDES contains an internal counter that controls dataflow. Failure to synchronize the reset deassertion with the CLKDIV will produce an unexpected output.

3-State Parallel-to-Serial Conversion

In addition to parallel-to-serial conversion of data, an OSERDES module also contains a parallel-to-serial converter for 3-state control of the IOB. Unlike data conversion, the 3-state converter can only serialize up to four bits of parallel 3-state signals. The 3-state converter cannot be cascaded.

OSERDES Primitive

The OSERDES primitive is shown in Figure 3-19.

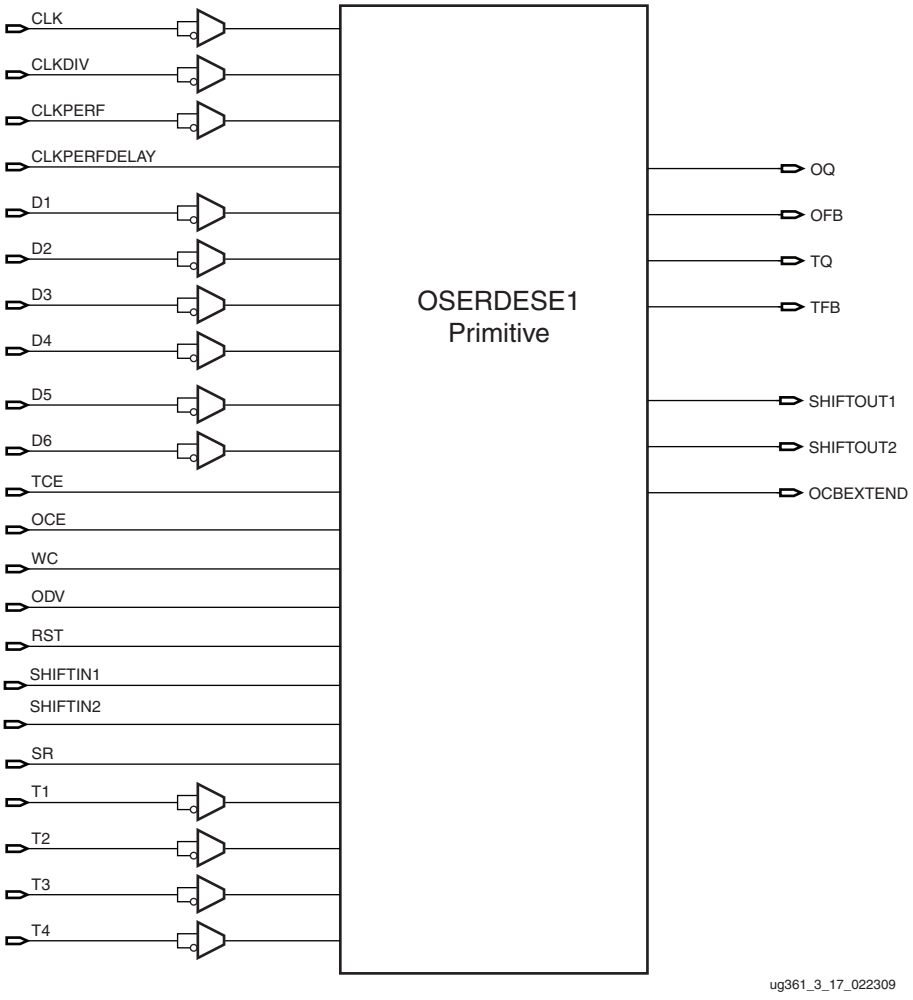


Figure 3-19: **OSERDESE1 Primitive**

OSERDES Ports

Table 3-5 lists the available ports in the OSERDES primitive.

Table 3-5: **OSERDES Port List and Definitions**

Port Name	Type	Width	Description
OQ	Output	1	Data path output to IOB only. See Data Path Output - OQ.
OFB	Output	1	Data path output feedback to ISERDESE1, or route through for CLKPERF to IODELAYE1, or connection to IODELAYE1. See OSERDES Latencies.
TQ	Output	1	3-state control output to IOB. See 3-state Control Output - TQ.
TFB	Output		3-state control output to IODELAYE1. See 3-state Control Output - TFB
SHIFTOUT1	Output	1	Carry output for data width expansion. Connect to SHIFTOUT1 of slave OSERDES. See OSERDES Width Expansion.
SHIFTOUT2	Output	1	Carry output for data width expansion. Connect to SHIFTOUT2 of slave OSERDES. See OSERDES Width Expansion.
OCBEXTEND	Output	1	Used in DDR mode (for DDR3 or DDR2), with the MEMORY_DDR3 attribute, to signal that the output circular buffer has extended the latency to match the CLK to the CLKPERF or CLKPERFDELAY. See Output Circular Buffer Extend - OCBEXTEND.
CLK	Input	1	High-speed clock input. See High-Speed Clock Input - CLK.
CLKDIV	Input	1	Divided clock input. Clocks delay element, deserialized data, Bitslip submodule, and CE unit. See Divided Clock Input - CLKDIV.
CLKPERF	Input	1	Only used in DDR mode (for DDR3 or DDR2), with the MEMORY_DDR3 attribute, for a dedicated high-speed clock from MMCM.
CLKPERFDELAY	Input	1	Only used in DDR mode (for DDR3 or DDR2), with the MEMORY_DDR3 attribute, for a dedicated high-speed clock from MMCM after being delayed through IODELAYE1.
D1 to D6	Input	1 (each)	Parallel data inputs. See Parallel Data Inputs - D1 to D6.
TCE	Input	1	3-state clock enable. See 3-state Signal Clock Enable - TCE.
OCE	Input	1	Output data clock enable. See Output Data Clock Enable - OCE.
WC	Input	1	Used in DDR mode (for DDR3 or DDR2), with the MEMORY_DDR3 attribute, when switching from transmitting data to receiving data to reset the output circular buffer.
ODV	Input	1	Used in DDR mode (for DDR3 or DDR2), with the MEMORY_DDR3 attribute, when the ODELAY delay value is greater than 180° of the fast clock period.
RST	Input	1	Active High reset.
SHIFTIN1	Input	1	Carry input for data width expansion. Connect to SHIFTIN1 of master OSERDES. See OSERDES Width Expansion.
SHIFTIN2	Input	1	Carry input for data width expansion. Connect to SHIFTIN2 of master OSERDES. See OSERDES Width Expansion.
T1 to T4	Input	1 (each)	Parallel 3-state inputs. See Parallel 3-state Inputs - T1 to T4.

Data Path Output - OQ

The OQ port is the data output port of the OSERDES module. Data at the input port D1 will appear first at OQ. This port connects the output of the data parallel-to-serial converter to the data input of the IOB. This port can not drive the IODELAYE1; the OFB pin must be used.

Output Feedback from OSERDESE1 - OFB

The output feedback port (OFB) is the serial (high-speed) data output port of the OSERDESE1 or the bypassed version of the CLKPERF. When the attribute ODELAYUSED is set to 0, the OFB port can be used to send out serial data to the ISERDESE1.

The OSERDESE1 pin OFB has three functions:

- As a feedback path to the ISERDESE1 OFB pin.
- As a connection to the IODELAYE1. The output of the OSERDESE1 can be routed though the OFB pin and then delayed through the IODELAYE1.
- To delay the high-performance input CLKPERF path through the IODELAYE1. When the ODELAY_VALUE attribute is set to 1 and the OSERDESE1 is in MEMORY_DDR3 mode, the CLKPERF input can be routed out through the OFB pin to the IODELAYE1 and back to the CLKPERFDELAY input.

When the attribute ODELAYUSED is set to 1 and the OSERDESE1 is in MEMORY_DDR3 mode, the OFB port can be used to link the high-performance clock input (CLKPERF) to the IODELAYE1.

3-state Control Output - TQ

This port is the 3-state control output of the OSERDES module. When used, this port connects the output of the 3-state parallel-to-serial converter to the control/3-state input of the IOB.

3-state Control Output - TFB

This port is the 3-state control output of the OSERDES module sent to the IODELAY. When used, this port connects the output of the 3-state parallel-to-serial converter to the control/3-state input of the IODELAY.

High-Speed Clock Input - CLK

This high speed clock input drives the serial side of the parallel-to-serial converters.

Divided Clock Input - CLKDIV

This divided high-speed clock input drives the parallel side of the parallel-to-serial converters. This clock is the divided version of the clock connected to the CLK port.

Performance Clock from MMCM - CLKPERF

This port (CLKPERF) is part of a dedicated path that provides a high-performance clock from the MMCM to the OSERDESE1. CLKPERF can be used in MEMORY_DDR3 mode for DDR3 and DDR2 applications. See OSERDES Clocking Methods.

Performance Clock from MMCM Delayed with IODELAYE1 - CLKPERFDELAY

This port (CLKPERFDELAY) is part of a dedicated path that provides a high-performance clock from the MMCM delayed with the IODELAYE1 to OSERDESE1. CLKPERFDELAY can be used in MEMORY_DDR3 mode for DDR3 and DDR2 applications. See OSERDES Clocking Methods. When the IODELAYE1 is not being used to delay the CLKPERF, connect CLKPERFDELAY to the same source as CLKPERF.

Parallel Data Inputs - D1 to D6

All incoming parallel data enters the OSERDES module through ports D1 to D6. These ports are connected to the FPGA fabric, and can be configured from two to six bits (i.e., a 6:1 serialization). Bit widths greater than six (up to 10) can be supported by using a second OSERDES in SLAVE mode. See OSERDES Width Expansion. Refer to Figure 3-3, page 133 for bit ordering at the inputs and output of the OSERDES along with the corresponding bit order of the ISERDESE1.

Reset Input - RST

When asserted, the reset input causes the outputs of most OSERDES internal data flip-flops in the CLK and CLKDIV domains to be driven low asynchronously. The exceptions are the four flip-flops in the output parallel-to-serial structure whose RESET value is selectable via attributes on the component. RST must be deasserted synchronously with CLKDIV.

The deassertion of reset releases most flip-flops in the OSERDESE1 except for some internal control logic that retimes this deassertion to the first rising edge of CLKDIV followed by the consecutive following rising edge of CLK.

- For the first rising CLKDIV edge after releasing reset, data at the data inputs is loaded in the OSERDESE1 and the first stage of the reset circuit is released.
- For the first rising CLK edge after releasing reset, data is shifted out at the OQ pin of the OSERDESE1 and the second stage of the reset circuit is released.
- After the release of the reset, and a first rising CLKDIV edge and a consecutive rising CLK edge, a state machine is engaged. This state machine generates pulses to load the data from the OSERDES input register into the output parallel-to-serial register. The moment of the load is defined by the DATA_WIDTH attribute.

As a result, after the release of the reset, the OSERDESE1 can output several bits, mostly zero, before meaningful data is output.

Every OSERDESE1 in a multiple bit output structure should be driven by the same reset signal, asserted asynchronously, and deasserted synchronously to CLKDIV. Best practices for ensuring that all OSERDESE1 components rise out of reset simultaneously are:

- Wait until all MMCM/PLL used in the design are locked.
  - Reset should only be deasserted when it is known that CLK and CLKDIV are stable and present.
- Wait until all IDELAYCTRL components, when used, show RDY high.
- Release reset synchronously to, and delayed by, a couple of CLKDIV cycles.
  - A reset pulse should be at minimum two CLKDIV cycles and at a maximum 32 CLKDIV cycles.

- On this reset net, use one flip-flop clocked by CLKDIV per ISERDESE1/OSERDESE1 pair.
  - Put this flip-flop in the FPGA fabric in front of the ISERDESE1/OSERDESE1 pair.
  - Put a timing constraint on the flip-flop to ISERDESE1/OSERDESE1 of one CLKDIV period or less.

The reset timing of multiple OSERDESE1 ports is shown in Figure 3-24, page 167.

To ensure that data flows out of all OSERDESE1 blocks in a multiple bit output structure:

- Place a register in front of the OSERDESE1 inputs.
- Clock the register by the CLKDIV clock of the OSERDESE1.
- Use the same reset signal for the register as for the OSERDESE1.

When the OSERDESE1 reset is released:

1. The first rising CLKDIV edge:
   - Loads all zeroes from the front register into the OSERDESE1.
   - Loads new application data in the front register.
   - Releases part of the internal OSERDESE1 control state machine.
2. The first rising CLK edge after the first rising CLKDIV edge:
   - Serial data is clocked out of the OSERDESE1.
   - The internal OSERDESE1 control state machine is released.
3. At the next rising CLK edges:
   - Data is serially clocked out of the OSERDESE1.
   - The internal control state machine is clocked.
   - The state machine generates a load signal at DATA_WIDTH boundary to load data from the parallel input register into the serial output register.
4. The next rising CLKDIV edge:
   - Loads new data into the OSERDESE1 input.
   - Loads new application data in the front register.
5. Return to 3.

Output Data Clock Enable - OCE

OCE is an active High clock enable for the data path.

Write Command - WC

The WC port is a part of the dedicated logic for the MEMORY_DDR3 mode. The write command is issued when switching from writing to reading data. WC is only available in MEMORY_DDR3 mode for DDR3 and DDR2 applications. When not using MEMORY_DDR3 mode, connect this port to GND.

ODELAY Value - ODV

The ODV port is a part of the dedicated logic for the MEMORY_DDR3 mode. The ODV is asserted High by the user when CLKPERFDELAY delay through the IODELAYE1 is greater than half of the period. ODV is only available in MEMORY_DDR3 mode for DDR3 and DDR2 applications. When not using MEMORY_DDR3 mode, connect this port to GND.

Output Circular Buffer Extend - OCBEXTEND

The OCBEXTEND port is a part of the dedicated logic for the MEMORY_DDR3 mode. The port is used to signal that the output circular buffer has extended the latency of the OSERDESE1.

3-state Signal Clock Enable - TCE

TCE is an active High clock enable for the 3-state control path.

Parallel 3-state Inputs - T1 to T4

All parallel 3-state signals enter the OSERDES module through ports T1 to T4. The ports are connected to the FPGA fabric, and can be configured as one, two, or four bits.

OSERDES Attributes

The Table 3-6 lists and describes the various attributes that are available for the OSERDES primitive. The table includes the default values.

Table 3-6: **OSERDES Attribute Summary**

Attribute	Description	Value	Default Value
DATA_RATE_OQ	Defines whether data (OQ) changes at every clock edge or every positive clock edge with respect to CLK.	String: SDR or DDR	DDR
DATA_RATE_TQ	Defines whether the 3-state (TQ) changes at every clock edge, every positive clock edge with respect to clock, or is set to buffer configuration.	String: BUF, SDR, or DDR	DDR
DATA_WIDTH	Defines the parallel-to-serial data converter width. This value also depends on the DATA_RATE_OQ value.	Integer: 2, 3, 4, 5, 6, 7, 8, or 10 See OSERDESE1 Attributes (Table 3-8) for valid combinations	4
SERDES_MODE	Defines whether the OSERDES module is a master or slave when using width expansion.	String: MASTER or SLAVE	MASTER
TRISTATE_WIDTH	Defines the parallel to serial 3-state converter width.	Integer: 1 or 4 See OSERDESE1 Attributes (Table 3-8) for valid combinations	4
ODELAY_USED	Helps to set the output circular buffer in the correct mode when using ODELAY. Only used when DDR mode is used (MEMORY_DDR3 attribute) for DDR2 and DDR3. Set ODELAY_USED to 0 for all other modes, even when ODELAY is used in the design.	Integer: 0 or 1	0
INTERFACE_TYPE	Chooses OSERDESE1 use model.	String: DEFAULT or MEMORY_DDR3	DEFAULT

DATA_RATE_OQ Attribute

The DATA_RATE_OQ attribute defines whether data is processed as single data rate (SDR) or double data rate (DDR). The allowed values for this attribute are SDR and DDR. The default value is DDR.

DATA_RATE_TQ Attribute

The DATA_RATE_TQ attribute defines whether 3-state control is to be processed as single data rate (SDR) or double data rate (DDR). The allowed values for this attribute are SDR and DDR. The default value is DDR.

DATA_WIDTH Attribute

The DATA_WIDTH attribute defines the parallel data input width of the parallel-to-serial converter. The possible values for this attribute depend on the DATA_RATE_OQ attribute. When DATA_RATE_OQ is set to SDR, the possible values for the DATA_WIDTH attribute are 2, 3, 4, 5, 6, 7, and 8. When DATA_RATE_OQ is set to DDR, the possible values for the DATA_WIDTH attribute are 4, 6, 8, and 10.

When the DATA_WIDTH is set to widths larger than six, a pair of OSERDES must be configured into a master-slave configuration. See OSERDES Width Expansion.

SERDES_MODE Attribute

The SERDES_MODE attribute defines whether the OSERDES module is a master or slave when using width expansion. The possible values are MASTER and SLAVE. The default value is MASTER. See OSERDES Width Expansion.

TRISTATE_WIDTH Attribute

The TRISTATE_WIDTH attribute defines the parallel 3-state input width of the 3-state control parallel-to-serial converter. The possible values for this attribute depend on the DATA_RATE_TQ attribute. When DATA_RATE_TQ is set to SDR or BUF, the TRISTATE_WIDTH attribute can only be set to 1. When DATA_RATE_TQ is set to DDR, the possible values for the TRISTATE_WIDTH attribute is 4.

TRISTATE_WIDTH cannot be set to widths larger than 4. When a DATA_WIDTH is larger than four, set the TRISTATE_WIDTH to 1.

ODELAY_USED Attribute

The ODELAY_USED attribute is only for DDR mode (DDR3 and DDR2). This attribute helps to set the output circular buffer in the correct mode when using ODELAY. Set ODELAY_USED to 0 for all other modes, even when ODELAY is used in the design.

INTERFACE_TYPE Attribute

The INTERFACE_TYPE attribute defines the mode of the OSERDESE1. The DEFAULT mode has the same functionality as it did in Virtex-5 FPGA designs. Table 3-7 shows the supported data width and clocking ratio of CLK and CLKDIV. This attribute supports the various clocking and data rates shown in Table 3-7.

Table 3-7: **CLK/CLKDIV Relationship of the Data Parallel-to-Serial Converter**

Input Data Width Output in SDR Mode	Input Data Width Output in DDR Mode	CLK	CLKDIV
2	4	2X	X
3	6	3X	X
4	8	4X	X
5	10	5X	X
6	–	6X	X
7	–	7X	X
8	–	8X	X

The MEMORY_DDR3 mode is used to allow the new features for the dedicated support of DDR3 and DDR2 applications. MEMORY_DDR3 mode is only supported with the MIG tool. Table 3-8 shows the valid setting and combinations of using the OSERDESE1.

Table 3-8: **OSERDESE1 Attribute Combinations**

INTERFACE_TYPE	DATA_RATE_OQ	DATA_RATE_TQ	DATA_WIDTH	TRISTATE_WIDTH
DEFAULT	SDR	SDR	1, 2, 3, 4, 5, 6, 7, 8	1
	DDR	DDR	4	4
		SDR	2, 6, 8, 10	1
MEMORY_DDR3	DDR	DDR	4	4

Notes:

1. ODELAY_USED is only set to 1 when in MEMORY_DDR3 mode when using the IODELAYE1.

OSERDES Clocking Methods

DEFAULT Interface Type Method

The phase relationship of CLK and CLKDIV is important in the parallel-to-serial conversion process. CLK and CLKDIV are (ideally) phase-aligned within a tolerance.

There are several clocking arrangements within the FPGA to help the design meet the phase relationship requirements of CLK and CLKDIV. The only valid clocking arrangements for the OSERDES are:

- CLK driven by BUFIO, CLKDIV driven by BUFR
- CLK and CLKDIV driven by CLKOUT[0:6] of same MMCM
- CLK driven by BUFG, CLKDIV driven by another BUFG

When using a MMCM to drive the CLK and CLKDIV of the OSERDESE1 the buffer types suppling the OSERDESE1 can not be mixed. For example, if CLK is driven by a BUFG, CLKDIV must be driven by a BUFG as well.

MEMORY_DDR3 Interface Type Method

In MEMORY_DDR3 there are four clock inputs to the OSERDESE1 (CLK, CLKDIV, CLKPERF, and CLKPERFDELAY). Ideally, CLK and CLKDIV are phase aligned coming from the MMCM via the BUFG (the same as the DEFAULT mode). In the DDR3 applications, the data must have a very small transmission loss and be able to delay each of the data bits. Hence, the CLKPERF and CLKPERFDELAY is added to the OSERDES. CLKPERF is the direct connection from the MMCM referenced as the high-performance path in the *Virtex-6 FPGA Clocking User Guide*. CLKPERFDELAY is delayed through the dedicated IODELAYE1 for each OSERDESE1 allowing the data offset.

- CLK and CLKDIV driven by CLKOUT[0:6] of same MMCM
- CLKPERF is driven by the MMCM's high-performance path
- CLKPERFDELAY is driven by a delayed version of CLKPERF via IODELAYE1 or tied to CLKPERF

Special care must be taken to set these clocks and phase alignment correctly. This mode is only supported through the MIG tool.

OSERDES Width Expansion

Only when the interface type is set to DEFAULT can two OSERDES modules be used to build a parallel-to-serial converter larger than 6:1. In every I/O tile there are two OSERDES modules; one master and one slave. By connecting the SHIFTIN ports of the master OSERDES to the SHIFTOUT ports of the slave OSERDES, the parallel-to-serial converter can be expanded to up to 10:1(DDR) and 8:1 (SDR). For a differential output, the master OSERDES must be on the positive side of the differential output pair. When the output is not differential, the output buffer associated with the slave OSERDES is not available and can not be used.

When using the OSERDES with width expansion, complementary single-ended standards (e.g., DIFF_HSTL and DIFF_SSTL) cannot be used. This is because both OLOGIC blocks in an I/O tile are used by the complementary single-ended standards to transmit both legs of the signal, leaving no OLOGIC blocks available for width expansion.

Figure 3-20 illustrates a block diagram of a 10:1 DDR parallel-to-serial converter using the master and slave OSERDES modules. Ports Q3-Q6 are used for the last four bits of the parallel interface on the slave OSERDES (LSB to MSB).

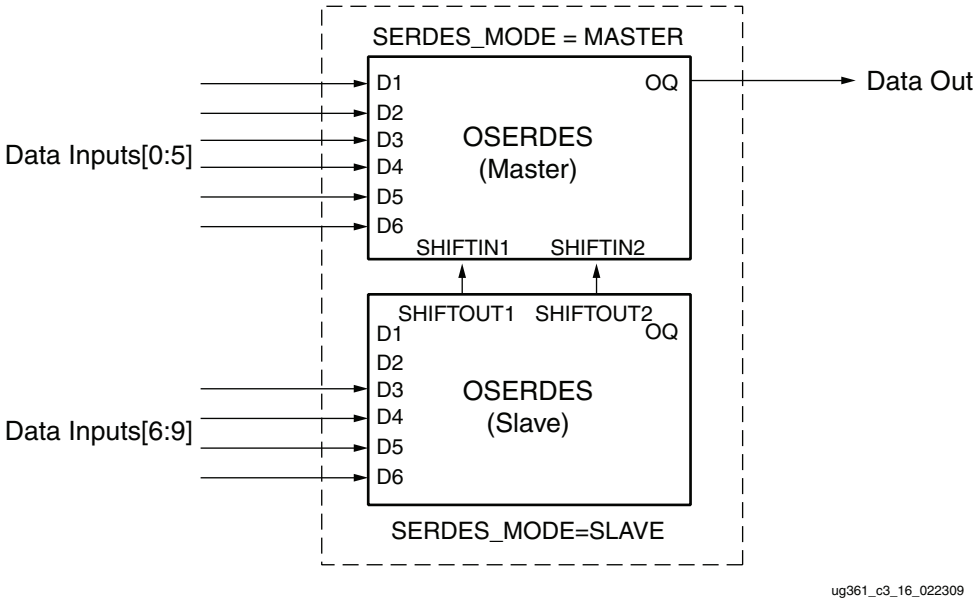


Figure 3-20: **Block Diagram of OSERDES Width Expansion**

Table 3-9 lists the data width availability for SDR and DDR mode.

Table 3-9: **OSERDES SDR/DDR Data Width Availability**

SDR Data Widths	2, 3, 4, 5, 6, 7, 8
DDR Data Widths	4, 6, 8, 10

Guidelines for Expanding the Parallel-to-Serial Converter Bit Width

1. Both the OSERDES modules must be adjacent master and slave pairs.
2. Set the SERDES_MODE attribute for the master OSERDES to MASTER and the slave OSERDES to SLAVE. See SERDES_MODE Attribute.
3. The user must connect the SHIFTIN ports of the MASTER to the SHIFTOUT ports of the SLAVE.
4. The SLAVE only uses the ports D3 to D6 as an input.
5. DATA_WIDTH for Master and Slave are equal. See DATA_WIDTH Attribute.
6. The attribute INTERFACE_TYPE is set to DEFAULT.

The slave inputs used for data widths requiring width expansion are listed in Table 3-10.

Table 3-10: **Slave Inputs Used for Data Width Expansion**

Data Width	Slave Inputs Used
7	D3
8	D3–D4
10	D3–D6

OSERDES Latencies

DEFAULT Interface Type Latencies

The input to output latencies of OSERDES blocks depend on the DATA_RATE and DATA_WIDTH attributes. Latency is defined as a period of time between the following two events: (a) when the rising edge of CLKDIV clocks the data at inputs D1–D6 into the OSERDES, and (b) when the first bit of the serial stream appears at OQ. Table 3-11 summarizes the various OSERDES latency values.

Table 3-11: **OSERDES Latencies**

DATA_RATE	DATA_WIDTH	Latency
SDR	2:1	1 CLK cycle
	3:1	2 CLK cycles
	4:1	3 CLK cycles
	5:1	4 CLK cycles
	6:1	5 CLK cycles
	7:1	6 CLK cycles
	8:1	7 CLK cycles
DDR	4:1	2 CLK cycle
	6:1	3 CLK cycles
	8:1	4 CLK cycles
	10:1	5 CLK cycles

OSERDES Timing Model and Parameters

This section discusses all timing models associated with the OSERDES primitive. Table 3-12 describes the function and control signals of the OSERDES switching characteristics in the *Virtex-6 FPGA Data Sheet*.

Table 3-12: **OSERDES Switching Characteristics**

Symbol	Description
Setup/Hold	
T_{OSDCK_D}/T_{OSCKD_D}	D input Setup/Hold with respect to CLKDIV
T_{OSDCK_T}/T_{OSCKD_T}	T input Setup/Hold with respect to CLK
T_{OSDCK_T}/T_{OSCKD_T}	T input Setup/Hold with respect to CLKDIV
$T_{OSCCK\_OCE}/T_{OSCKC\_OCE}$	OCE input Setup/Hold with respect to CLK
$T_{OSCCK\_TCE}/T_{OSCKC\_TCE}$	TCE input Setup/Hold with respect to CLK
Sequential Delays	
T_{OSCKO_OQ}	Clock to Out from CLK to OQ
T_{OSCKO_TQ}	Clock to Out from CLK to TQ

Table 3-12: **OSERDES Switching Characteristics *(Cont'd)***

Symbol	Description
Combinatorial	
T_{OSCO_OQ}	Asynchronous Reset to OQ
T_{OSCO_TQ}	Asynchronous Reset to TQ

Timing Characteristics of 2:1 SDR Serialization

In Figure 3-21, the timing of a 2:1 SDR data serialization is illustrated.

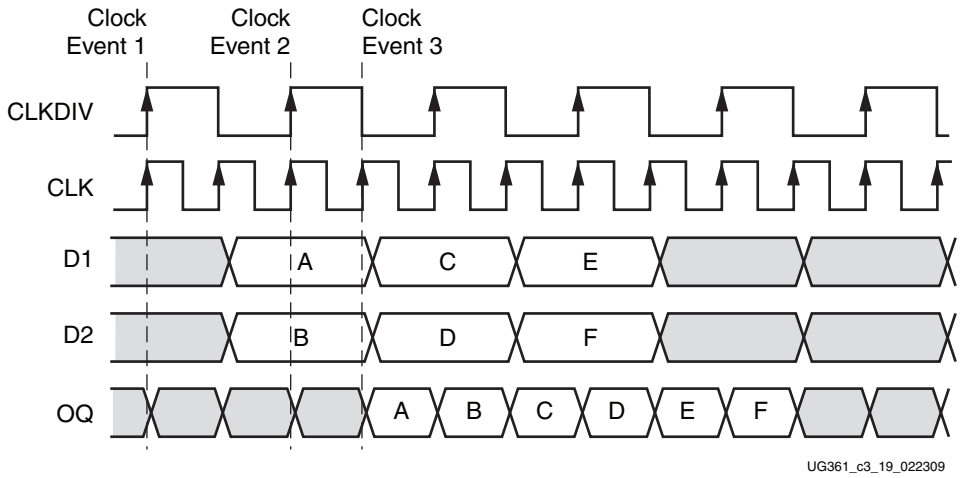


Figure 3-21: **OSERDES Data Flow and Latency in 2:1 SDR Mode**

Clock Event 1

On the rising edge of CLKDIV, the word *AB* is driven from the FPGA logic to the D1 and D2 inputs of the OSERDES (after some propagation delay).

Clock Event 2

On the rising edge of CLKDIV, the word *AB* is sampled into the OSERDES from the D1 and D2 inputs.

Clock Event 3

The data bit A appears at OQ one CLK cycle after *AB* is sampled into the OSERDES. This latency is consistent with the Table 3-11 listing of a 2:1 SDR mode OSERDES latency of one CLK cycle.

Timing Characteristics of 8:1 DDR Serialization

Figure 3-22 illustrates the timing of an 8:1 DDR data serialization. In contrast to the 2:1 SDR example, a second OSERDES is required to achieve an 8:1 serialization. The two OSERDES are connected and configured using the methods described in OSERDES Width Expansion. Six of the eight bits are connected to D1–D6 of the master OSERDES while the remaining two bits are connected to D3–D4 of the slave OSERDES.

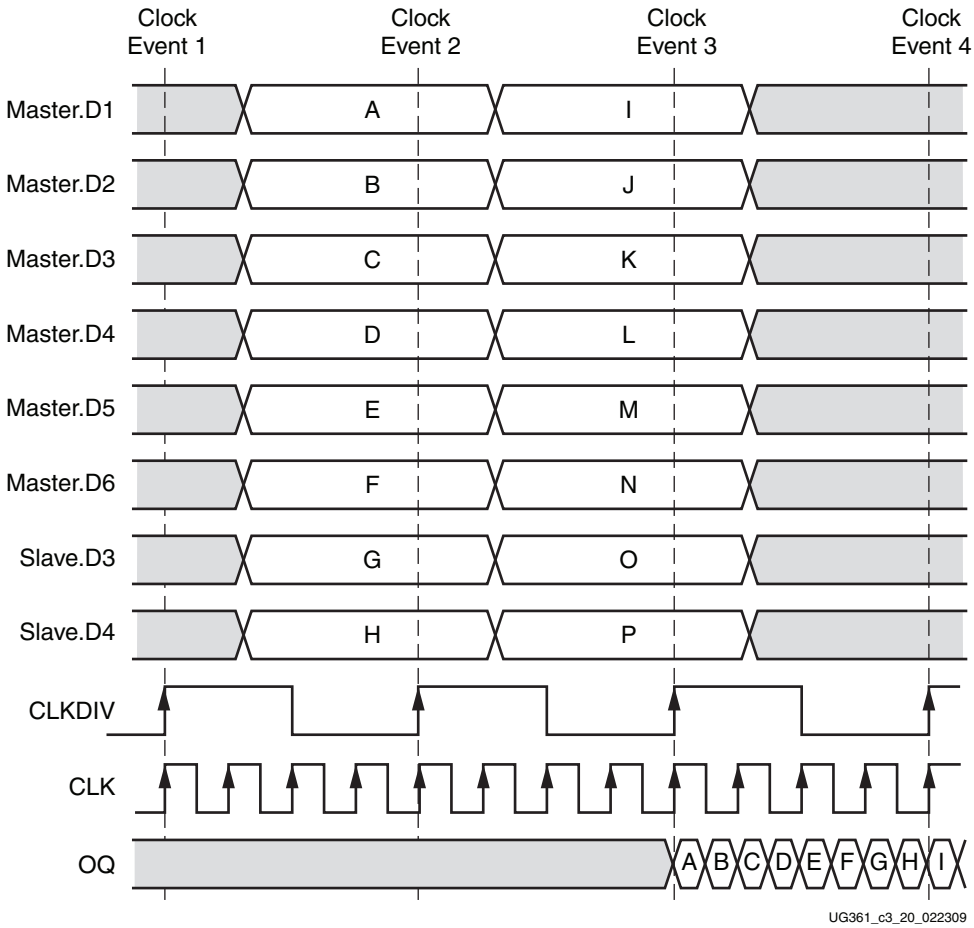


Figure 3-22: **OSERDES Data Flow and Latency in 8:1 DDR Mode**

Clock Event 1

On the rising edge of CLKDIV, the word *ABCDEFGH* is driven from the FPGA logic to the D1–D6 inputs of the master OSERDES and D3–D4 of the slave OSERDES (after some propagation delay).

Clock Event 2

On the rising edge of CLKDIV, the word *ABCDEFGH* is sampled into the master and slave OSERDES from the D1–D6 and D3–D4 inputs, respectively.

Clock Event 3

The data bit *A* appears at OQ four CLK cycles after *ABCDEFGH* is sampled into the OSERDES. This latency is consistent with the Table 3-11 listing of a 8:1 DDR mode OSERDES latency of four CLK cycles.

The second word *IJKLMNOP* is sampled into the master and slave OSERDES from the D1–D6 and D3–D4 inputs, respectively.

Clock Event 4

Between Clock Events 3 and 4, the entire word *ABCDEFGH* is transmitted serially on OQ, a total of eight bits transmitted in one CLKDIV cycle.

The data bit *I* appears at OQ four CLK cycles after *IJKLMNOP* is sampled into the OSERDES. This latency is consistent with the Table 3-11 listing of a 8:1 DDR mode OSERDES latency of four CLK cycles.

Timing Characteristics of 4:1 DDR 3-State Controller Serialization

The operation of a 3-State Controller is illustrated in Figure 3-23. The example is a 4:1 DDR case shown in a bidirectional system where the IOB must be frequently 3-stated.

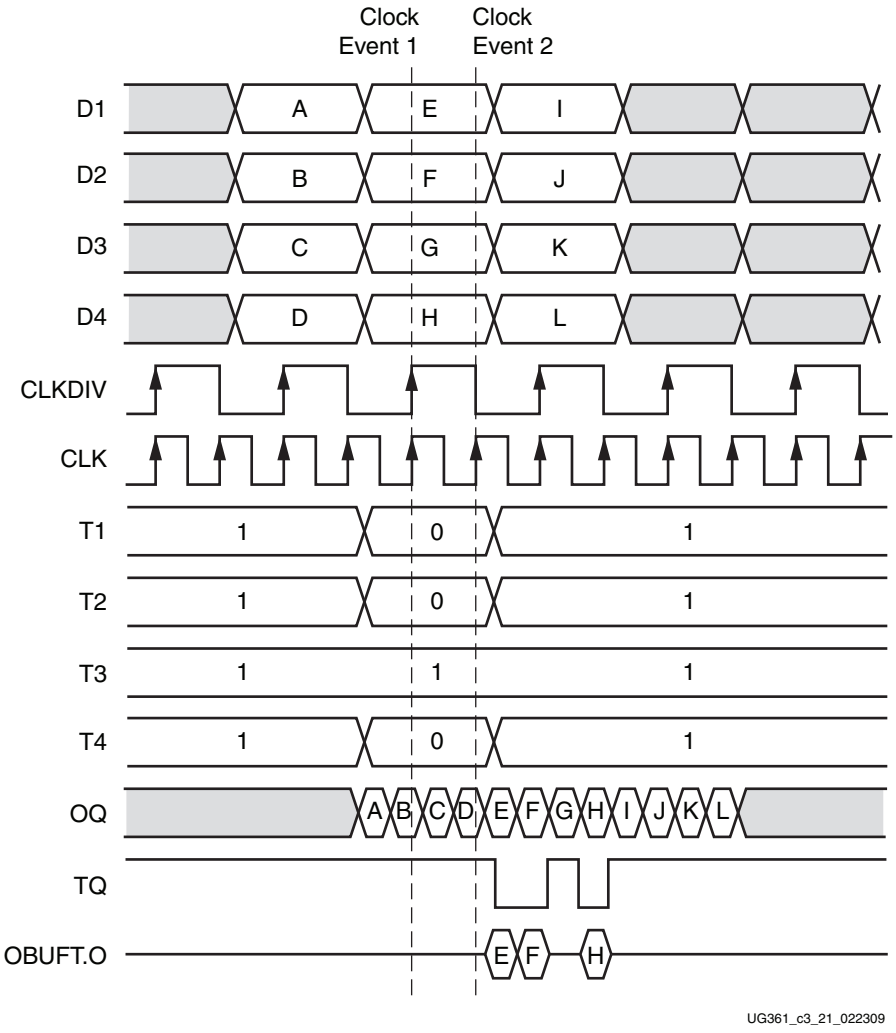


Figure 3-23: **OSERDES Data Flow and Latency in 4:1 DDR Mode**

Clock Event 1

T1, T2, and T4 are driven Low to release the 3-state condition. The serialization paths of T1–T4 and D1–D4 in the OSERDES are identical (including latency), such that the bits *EFGH* are always aligned with the `0010` presented at the T1–T4 pins during Clock Event 1.

Clock Event 2

The data bit *E* appears at OQ one CLK cycle after *EFGH* is sampled into the OSERDES. This latency is consistent with the Table 3-11 listing of a 4:1 DDR mode OSERDES latency of one CLK cycle.

The 3-state bit 0 at T1 during Clock Event 1 appears at TQ one CLK cycle after `0010` is sampled into the OSERDES 3-state block. This latency is consistent with the Table 3-11 listing of a 4:1 DDR mode OSERDES latency of one CLK cycle.

Reset Output Timing

The reset must be applied to the OSERDESE1 regardless of the mode of operation. Figure 3-24 shows the reset behavior when INTERFACE_TYPE is set to DEFAULT.

Clock Event 1

A reset pulse is generated on the rising edge of CLKDIV. Because the pulse must take two different routes to get to OSERDES0 and OSERDES1, there are different propagation delays for both paths. The difference in propagation delay is emphasized in Figure 3-24. The path to OSERDES0 is very long and the path to OSERDES1 is very short, such that each OSERDES receives the reset pulse in a different CLK cycle. The internal resets for both CLK and CLKDIV go into reset asynchronously when the RST input is asserted.

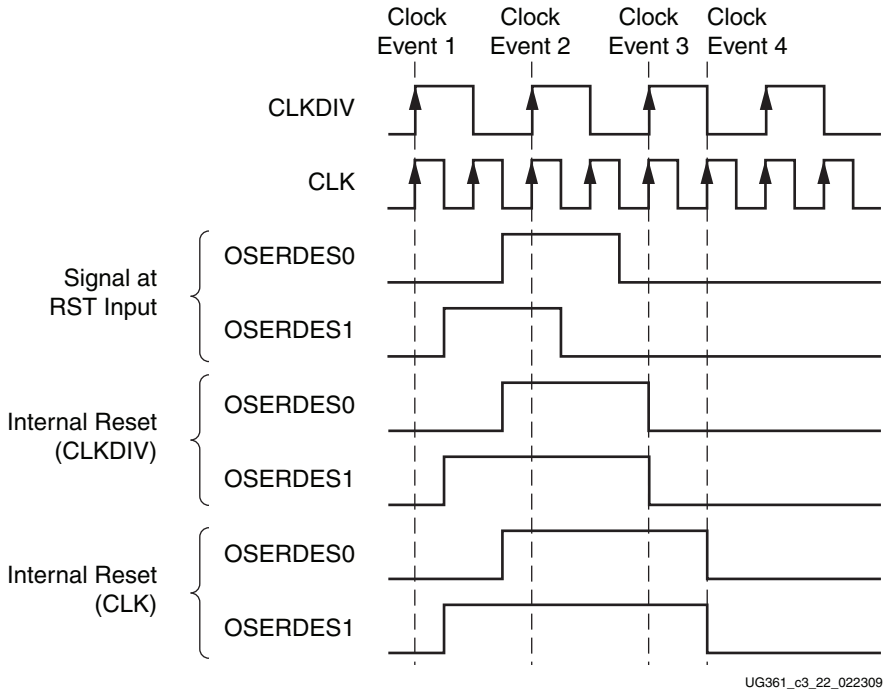


Figure 3-24: **Two OSERDESE1 with INTERFACE_TYPE set to DEFAULT Coming Out of Reset Synchronously with One Another**

Clock Event 2

The reset pulse is deasserted on the rising edge of CLKDIV. The difference in propagation delay between the two OSERDES causes the RST input to come out of reset on two different CLK cycles. Without internal retiming, OSERDES1 finishes reset one CLK cycle before OSERDES0 and both OSERDES are asynchronous.

Clock Event 3

The release of the reset signal at the RST input is retimed internally to CLKDIV. This synchronizes OSERDES0 and OSERDES1.

Clock Event 4

The release of the reset signal at the RST input is retimed internally to CLK.

Figure 3-25 shows the reset behavior when INTERFACE_TYPE is set to MEMORY_DDR3.

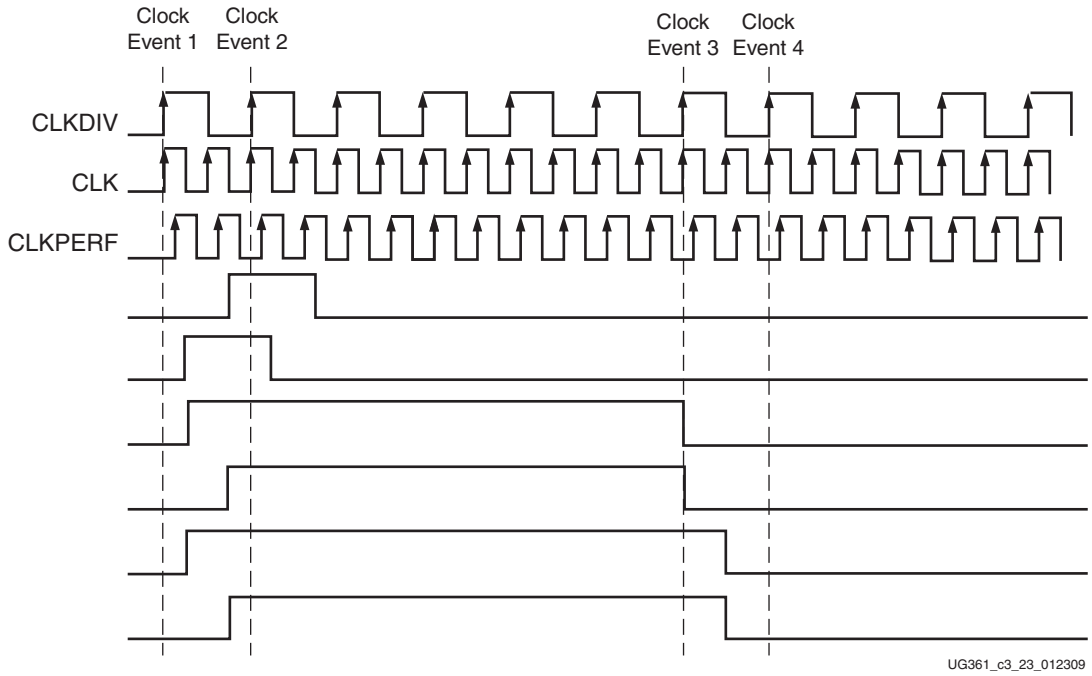


Figure 3-25: **Two OSERDESE1 with INTERFACE_TYPE set to MEMORY_DDR3 Coming Out of Reset Synchronously with One Another**

Clock Event 1

A reset pulse is generated on the rising edge of CLKDIV. Because the pulse must take two different routes to get to OSERDES0 and OSERDES1, there are different propagation delays for both paths. The difference in propagation delay is emphasized in Figure 3-25. The path to OSERDES0 is very long and the path to OSERDES1 is very short, so that each OSERDES receives the reset pulse in a different CLK cycle. The internal resets for both CLK and CLKDIV are reset asynchronously when the RST input is asserted.

Clock Event 2

The reset pulse is deasserted on the rising edge of CLKDIV. The difference in propagation delay between the two OSERDES causes the RST input to come out of reset on two

different CLK cycles. Without internal retiming, OSERDES1 finishes reset one CLK cycle before OSERDES0 and both OSERDES are asynchronous.

Clock Event 3

The release of the reset signal at the RST input is retimed internally to six CLKDIV after Clock Event 2. This synchronizes OSERDES0 and OSERDES1 and allows for the output circular buffer to be retimed.

Clock Event 4

The release of the reset signal at the RST input is retimed internally to CLK.

OSERDES VHDL and Verilog Instantiation Templates

The Libraries Guide includes instantiation templates of the OSERDESE1 module in VHDL and Verilog.