Vivado Design Suite User Guide

I/O and Clock Planning

UG899 (v2012.3) October 30, 2012

This document applies to the following software versions: Vivado Design Suite 2012.3 and 2012.4

XILINX®

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

© Copyright 2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
08/08/12	2012.2	Initial Xilinx release.
10/30/12	2012.3	Revised Setting I/O Port Direction. Revised Viewing SSN Results. Revised Sorting DRC Violations. Made typographical edits.

XILINX®

Table of Contents

Revision History 2

Chapter 1: Introduction

Overview 5
I/O Planning Stages 5
I/O Planning View Layout 8

Chapter 2: I/O Pin Planning Methodology

Overview 9
I/O Planning Project Flow 10
RTL Project Flow 13

Chapter 3: I/O Pin Planning Features

Using the I/O Planning View Layout 16
Viewing Device Resources 18
Defining Alternate Compatible Parts 23
Setting Device Configuration Modes 24
Defining and Configuring I/O Ports 25
Disabling or Enabling Interactive DRCs 35
Placing I/O Ports 36
Placing Clock Logic 42
Validating I/O and Clock Logic Placement 46
Migrating to an RTL Design 49
Exporting I/O Pin and Package Data 50
Exporting IBIS Models 51
Running SSN Analysis 52

Appendix A: I/O Port Lists in CSV File Format

CSV File 56
Differential Pairs in the CSV File 58

Appendix B: I/O Port DRC Rule Descriptions

DRCs 59

Appendix C: Additional Resources

Xilinx Resources **62**
Solution Centers **62**
References **62**

Introduction

Overview

I/O pin planning is a process that includes printed circuit board (PCB) designers, field programmable gate array (FPGA) designers, and system designers; each with their own specific set of concerns and requirements. Often, designers are hindered by a non-optimal pinout that causes further delays when trying to meet timing and signal integrity requirements.

By considering the data flow from PCB to FPGA die, you can achieve optimal pinout configurations quickly, thus reducing internal and external trace lengths as well as routing congestion. This chapter provides an overview of the I/O planning process using the graphical user interface (GUI) known as the Vivado™ Integrated Design Environment (IDE).

I/O Planning Stages

The Vivado IDE facilitates I/O planning at different stages of the design process. As the design progresses, more information becomes available, enabling more complex rule checking as the design is synthesized and implemented. The type of work you can do in each step of the design process varies. For example, early in the process, some data is missing; therefore, analysis is an estimate only. Later in the process, additional data is available; therefore, analysis is more accurate.

In the Vivado tools, you can start I/O planning with an empty project, move to register transfer level (RTL) source files and synthesized netlist, and finally work in an implemented design. Working with an implemented design is the final validation of an I/O pin and clock configuration. Proper clock resource validation requires full implementation of all clocks.

Proper I/O assignment can depend on how the clocks are configured; assigning I/Os and clock logic often go together. For the I/O placement design rule checks (DRCs) to account for clocks, you must have a synthesized design. Whenever possible, it is best to perform I/O assignment with a synthesized design.

Following is an overview of the I/O planning stages:

1. **Create an I/O planning project.**

   You can create an empty project to enable early device exploration and I/O port configuration. This allows for early pinout definition to eliminate pinout-related changes that typically happen late in the design cycle. When creating the project, you can create I/O ports manually or import them from comma separated values (CSV) files or Xilinx® design constraints (XDC) files. After the project is created, you can:

   - Export device and I/O port assignments for use later in the design process.
   - Migrate an I/O planning project to an RTL project when the port definitions and pin assignments are resolved.

2. **Elaborate and check RTL source files.**

   You can perform I/O planning in an RTL project. In an elaborated design, the tool provides basic DRCs.

RECOMMENDED: *To check clock logic, validation with a synthesized design is recommended.*

3. **Synthesize the design.**

   You can perform I/O planning after synthesis in the synthesized design. Because all clocks are determined and the Vivado IDE has visibility into all clocks, a more thorough validation is performed. Whenever possible, perform I/O assignment using a synthesized design.

4. **Implement the design and conduct final I/O validation.**

   The design must be fully implemented to ensure a legal I/O pinout. Examine the implementation reports for I/O and clock-related messages.

Table 1-1 shows the features supported at each I/O planning stage.

Table 1-1: **I/O Planning Stages and Features**

Feature	I/O Planning Project	RTL Design	Synthesized Design	Implemented Design
Read Ports from CSV and XDC Files	Supported	N/A	N/A	N/A
Create or Delete Ports	Supported	N/A	N/A	N/A
Migrate to RTL Project	Supported	N/A	N/A	N/A
Set Part Compatibility	Supported	Supported	Supported	Supported
Set Configuration Modes	Supported	Supported	Supported	Supported
Base I/O Standard DRC	Supported	Supported	Supported	Supported
SSN Analysis	Supported	Supported	Supported	Supported
Clock-Aware Placement and Clock-Aware DRC	N/A	N/A	Supported	Supported
Final Sign-Off DRC	N/A	N/A	N/A	Supported

For detailed information on these stages, see Chapter 2, I/O Pin Planning Methodology. For more information on working with projects, such as creating an empty I/O planning project or elaborating an RTL design, see the *Vivado Design Suite User Guide: System-Level Design Entry (UG895)*.

I/O Planning View Layout

In the Vivado IDE, the I/O Planning view layout provides an interface to:

- Create, import, and configure the initial list of I/O ports early in the design flow.
- Perform final verification of the pinout at the end of the design flow.
- Group related ports into interfaces, then assign them to package pins.
- Use fully automatic pin placement or semi-automated interactive modes for controlled I/O port assignment.
- View the relationship of the physical package pins and banks with their corresponding I/O die pads.
- Make intelligent decisions to optimize the connectivity between the PCB and the FPGA device.
- Analyze the design and device I/O requirements.
- Define an I/O pinout configuration or *pinout* that satisfies the requirements of both PCB and FPGA designs.

For detailed information on the I/O Planning view layout, see Chapter 3, I/O Pin Planning Features.

Chapter 2

I/O Pin Planning Methodology

Overview

Using the Vivado™ IDE, you can work on I/O pin planning at any stage in the design flow. Following are the most commonly used methods, which are covered in this chapter:

- **I/O Planning Project Flow**: In this flow, you do not have a netlist, and you are working on initial I/O and board planning.
- **RTL Project Flow**: In this flow, you have an RTL design, and you are targeting a new Xilinx® FPGA.

I/O Planning Project Flow

Figure 2-1 shows the I/O planning flow using an I/O planning project.

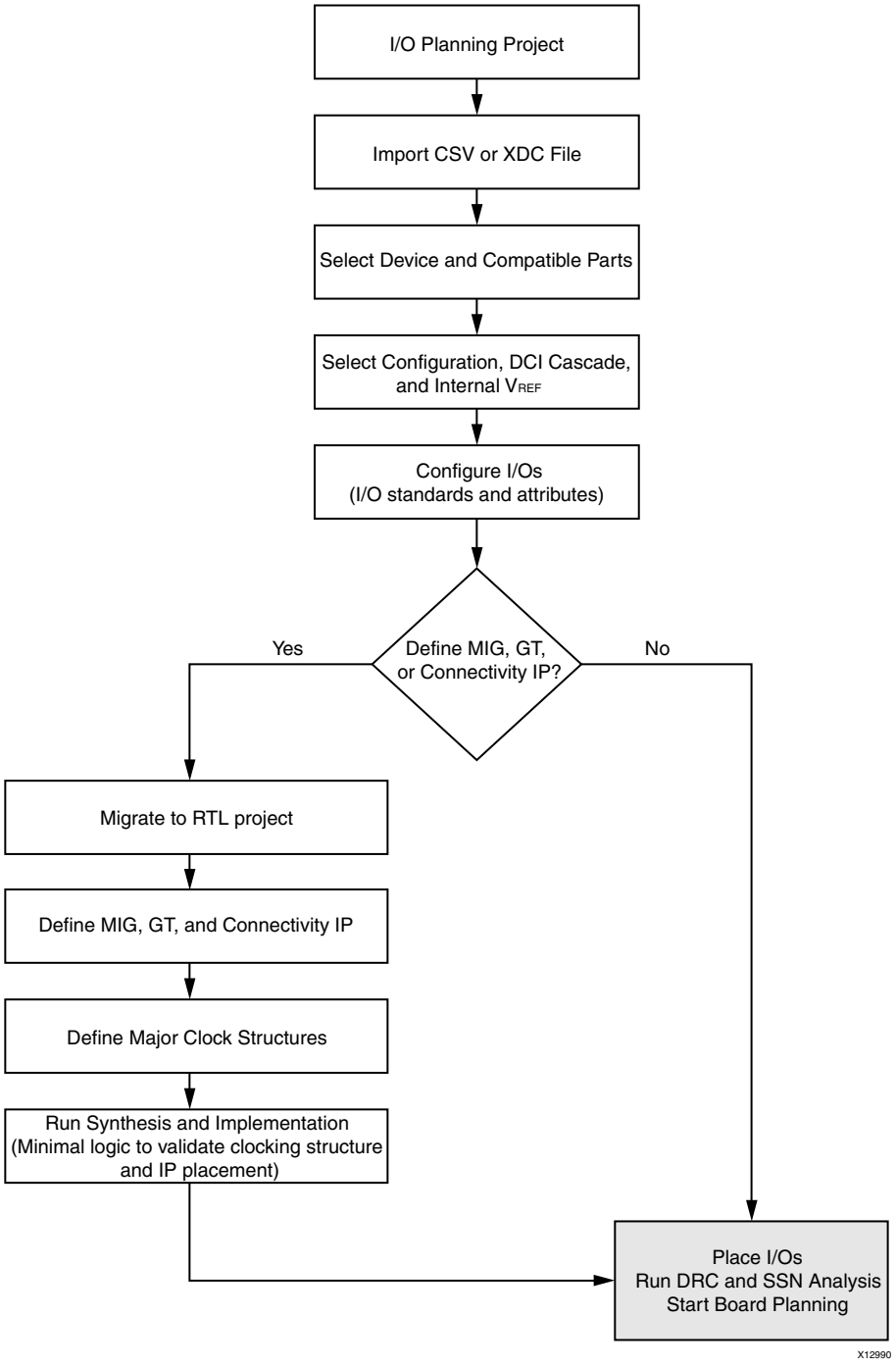

Figure 2-1: **I/O Planning Project Flow**

Following are the steps in the I/O planning project flow:

1. **Create an I/O planning project using the New Project wizard.**

   For information on creating an I/O planning project, see the *Vivado Design Suite User Guide: System-Level Design Entry (UG895)*.

2. **Import CSV or XDC files.**

   For information, see Defining and Configuring I/O Ports in Chapter 3.

3. **Select the device.**

   Determine the device size based on resource needs. Select the package based on PCB requirements, such as critical routes to memories. For designs using stacked silicon interconnect (SSI) technology, see the *Large FPGA Methodology Guide (UG782)*.

   Note: In addition to the selected device, you can also identify alternate compatible parts, as described in Defining Alternate Compatible Parts in Chapter 3.

4. **Select configuration, digitally controlled impedance (DCI) cascade, and internal voltage reference (V_{REF}).**

   For information, see the following sections in Chapter 3, I/O Pin Planning Features:

   - Setting Device Configuration Modes
   - Configuring DCI_CASCADE Constraints

5. **Configure I/Os.**

   For information, see Configuring I/O Ports in Chapter 3. Also, see the following resources available from the Xilinx website or Xilinx Documentation Navigator:

   - *FPGA Packaging and Pinout Specifications*: Provide information on the packaging and pinout specifications for specific device families.
   - *FPGA SelectIO™ Resources User Guides*: Provide information on banking rules. For example, some I/O standards can be combined within a single bank and some cannot.

6. **Optionally, migrate to an RTL project.**

   You can migrate the I/O port assignments made in the I/O planning project to an RTL project. For more information, see Migrating to an RTL Design in Chapter 3.

From the RTL project, you can:

a. **Define MIG, GT, and connectivity IP.**

You can use the IP Catalog to define memory interface generator (MIG), gigabit transceiver (GT), and connectivity intellectual property (IP). For more information on working with IP, see the *Vivado Design Suite User Guide: Designing with IP (UG896)*.

Note: Some IP, such as Ethernet IP and PCI Express® (PCIe) technology IP, has specific pinout requirements. In addition, high speed memory interfaces have specific pinout requirements driven by clocking and skew needs.

IMPORTANT: *After generating MIG IP, you must add the resulting XDC file to your project. In some cases, generating GT and connectivity IP also results in an XDC file that you must add to your project.*

b. **Define major clock structures.**

For information, see Placing Clock Logic in Chapter 3. You can also use the IP Catalog to generate mixed-mode clock manager (MMCM) or phase-locked loop (PLL) modules to define clock connections. For information on achieving timing closure, see the *Vivado Design Suite User Guide: Design Analysis and Closure Techniques (UG906)*.

c. **Run synthesis and implementation.**

For information, see the *Vivado Design Suite User Guide: Synthesis (UG901)* and *Vivado Design Suite User Guide: Implementation (UG904)*.

7. **Place I/Os, run DRC and SSN analysis, and start board planning.**

After placing I/Os, run DRC and simultaneous switching noise (SSN) analysis. For information, see the following sections in Chapter 3, I/O Pin Planning Features:

- Disabling or Enabling Interactive DRCs
- Running SSN Analysis

The following are considerations when board planning:

- For board-level validation, perform signal integrity analysis using I/O buffer information specification (IBIS) or HSPICE models. For information, see Exporting IBIS Models in Chapter 3.
- To optimize the pinout within the context of the entire board, import the FPGA into third-party products, such as Cadence Allegro FPGA System Planner or Mentor Graphics I/O Designer.

RTL Project Flow

Figure 2-2 shows the I/O planning flow using an RTL project.

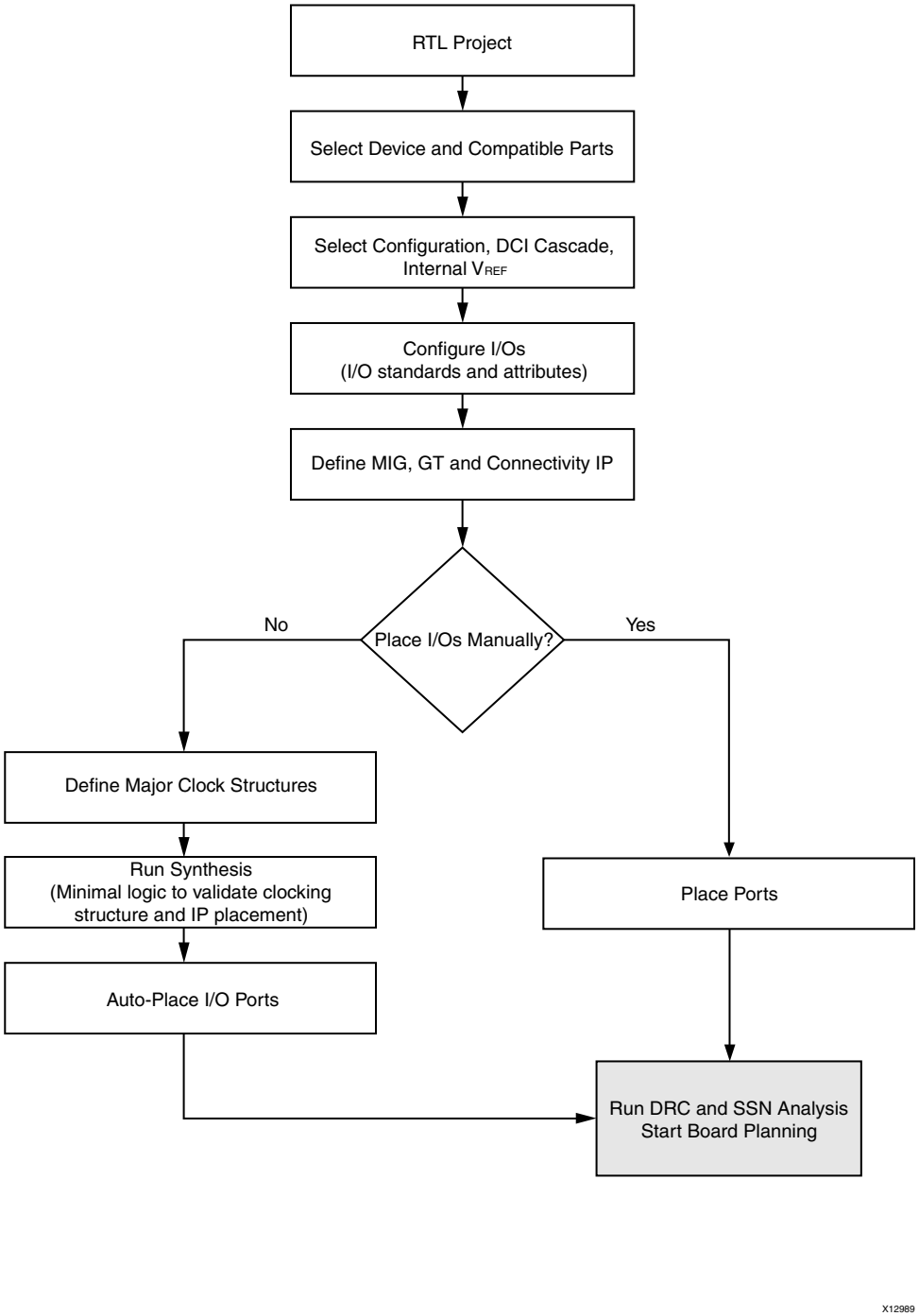

Figure 2-2: **RTL Project Flow**

Following are the steps in the RTL project flow:

1. **Create an RTL project using the New Project wizard.**

   For information on creating an RTL project, see the *Vivado Design Suite User Guide: System-Level Design Entry (UG895)*.

2. **Select the device.**

   Determine the device size based on resource needs. Select the package based on PCB requirements, such as critical routes to memories. For designs using SSI technology, see the *Large FPGA Methodology Guide (UG782)*.

   Note: In addition to the selected device, you can also identify alternate compatible parts, as described in Defining Alternate Compatible Parts in Chapter 3.

3. **Select configuration, digitally controlled impedance (DCI) cascade, and internal voltage reference (V_{REF}).**

   For information, see the following sections in Chapter 3, I/O Pin Planning Features:

   - Setting Device Configuration Modes
   - Configuring DCI_CASCADE Constraints

4. **Configure I/Os.**

   For information, see Configuring I/O Ports in Chapter 3. Also, see the following resources available from the Xilinx website or Xilinx Documentation Navigator:

   - *FPGA Packaging and Pinout Specifications*: Provide information on the packaging and pinout specifications for specific device families.
   - *FPGA SelectIO Resources User Guides*: Provide information on banking rules. For example, some I/O standards can be combined within a single bank and some cannot.

5. **Define MIG, GT, and connectivity IP.**

   You can use the IP Catalog to define MIG, GT, and connectivity IP. For more information on working with IP, see the *Vivado Design Suite User Guide: Designing with IP (UG896)*.

   Note: Some IP, such as Ethernet IP and PCI Express® (PCIe) technology IP, has specific pinout requirements. In addition, high speed memory interfaces have specific pinout requirements driven by clocking and skew needs.

IMPORTANT: *After generating MIG IP, you must add the resulting XDC file to your project. In some cases, generating GT and connectivity IP also results in an XDC file that you must add to your project.*

6. **Optionally, automatically place I/Os**.

   You can automatically place I/Os as follows:

   a. **Define major clock structures.**

      For information, see Placing Clock Logic in Chapter 3. You can also use the IP Catalog to generate MMCM or PLL modules to define clock connections. For information on achieving timing closure, see the *Vivado Design Suite User Guide: Design Analysis and Closure Techniques (UG906)*.

   b. **Run synthesis.**

      For information, see the *Vivado Design Suite User Guide: Synthesis (UG901)*.

   c. **Auto-place ports.**

      For information, see Automatically Placing I/O Ports in Chapter 3.

7. **Optionally, manually place I/Os.**

   For information, see Placing I/O Ports in Chapter 3.

8. **Place I/Os, run DRC and SSN analysis, and start board planning.**

   After placing I/Os, run DRC and SSN analysis. For information, see the following sections in Chapter 3, I/O Pin Planning Features:

   - Disabling or Enabling Interactive DRCs
   - Running SSN Analysis

   The following are considerations when board planning:

   - For board-level validation, perform signal integrity analysis using IBIS or HSPICE models. For information, see Exporting IBIS Models in Chapter 3.
   - To optimize the pinout within the context of the entire board, import the FPGA into third-party products, such as Cadence Allegro FPGA System Planner or Mentor Graphics I/O Designer.

I/O Pin Planning Features

Using the I/O Planning View Layout

In the Vivado™ IDE, you can use the I/O Planning view layout on elaborated, synthesized, and implemented designs. The view layout uses both Device and Package windows. Additional I/O information appears in the following windows: Clock Resources, Clock Regions, Package Pins, I/O Ports, and Properties windows.

Note: For more information on the windows in the Vivado IDE, see the *Vivado Design Suite User Guide: Using the Vivado IDE (UG893)*.

To launch the I/O Planning view layout, use any of the following methods:

- Select **Layout > I/O Planning**.
- From the Layout selector, select **I/O Planning**.
- Using the New Project wizard, create a new **I/O Planning Project**.

  Note: For more information on creating an I/O planning project, see the *Vivado Design Suite User Guide: System-Level Design Entry (UG895)*.

Figure 3-1 shows the I/O Planning view layout.

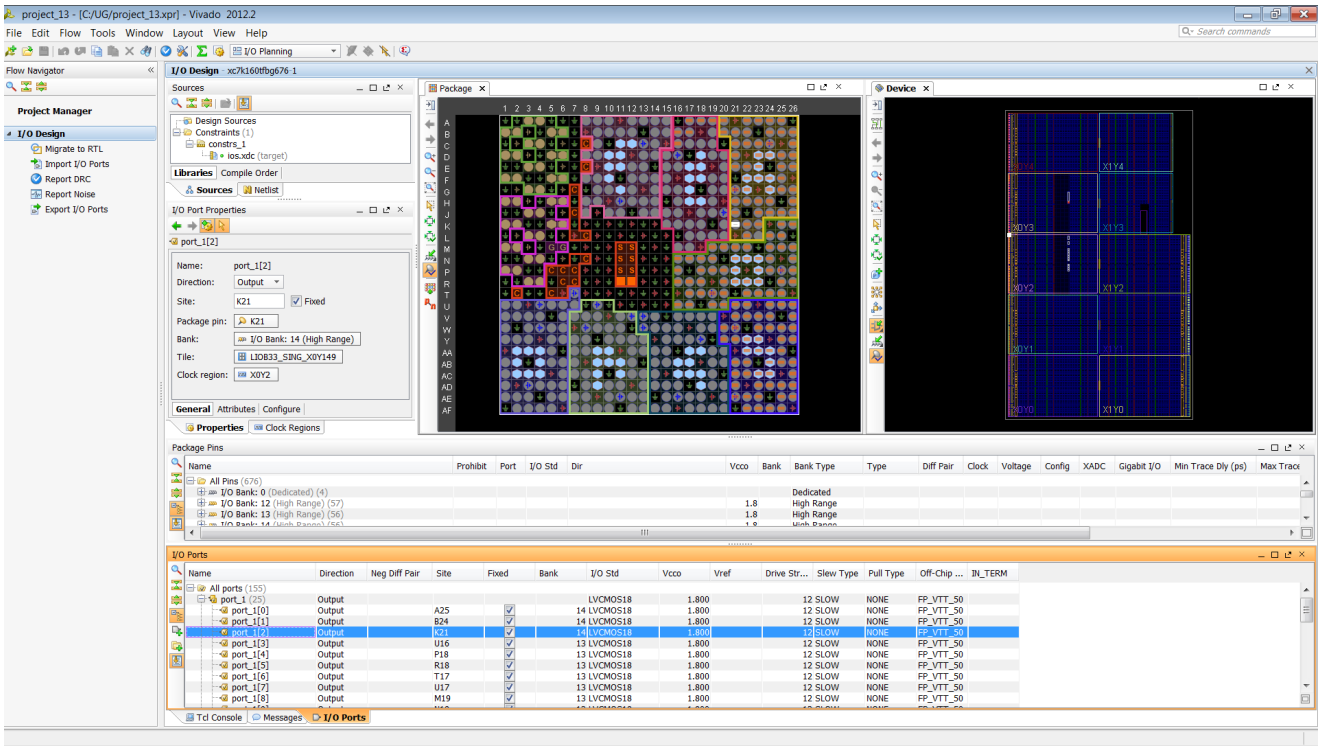

Figure 3-1: **I/O Planning View Layout**

Viewing Device Resources

The Device and Package windows show a graphical representation of the device and placed logic resources. When a logic object or device site is selected in these windows, information appears in the Properties window. The following sections describe these windows in detail.

TIP: *To search for specific objects or device sites, use the **Edit > Find** command. The Find dialog box includes many searchable object types and robust filtering capabilities to search the device or design for specific objects. You can select objects directly from the Find Results window.*

Properties

The Properties window shows properties for a selected object. The title bar changes to reflect the type of object selected. In most cases, the Properties window includes various views to show different information about the object. For example, in Figure 3-2, the Properties window shows the I/O Port Properties and includes the General, Attributes, and Configure views. To show the Properties window, select **Window > Properties**.

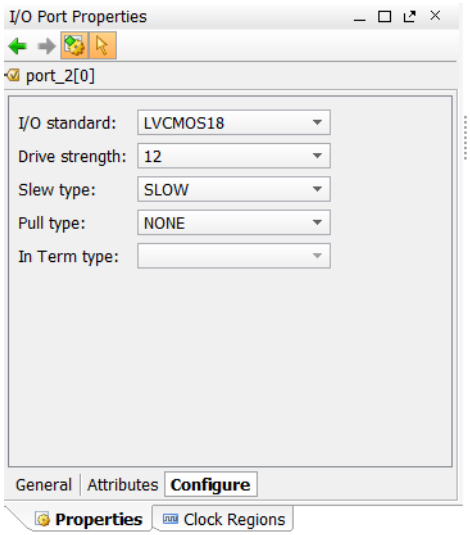

Figure 3-2: **I/O Port Properties**

Clock Region Resources and Statistics

The Clock Regions window allows easy selection of the clock regions. When you select a clock region in the Clock Regions window, the related I/O banks and regional clock resources are highlighted in the Package and Device windows, as shown in Figure 3-3.

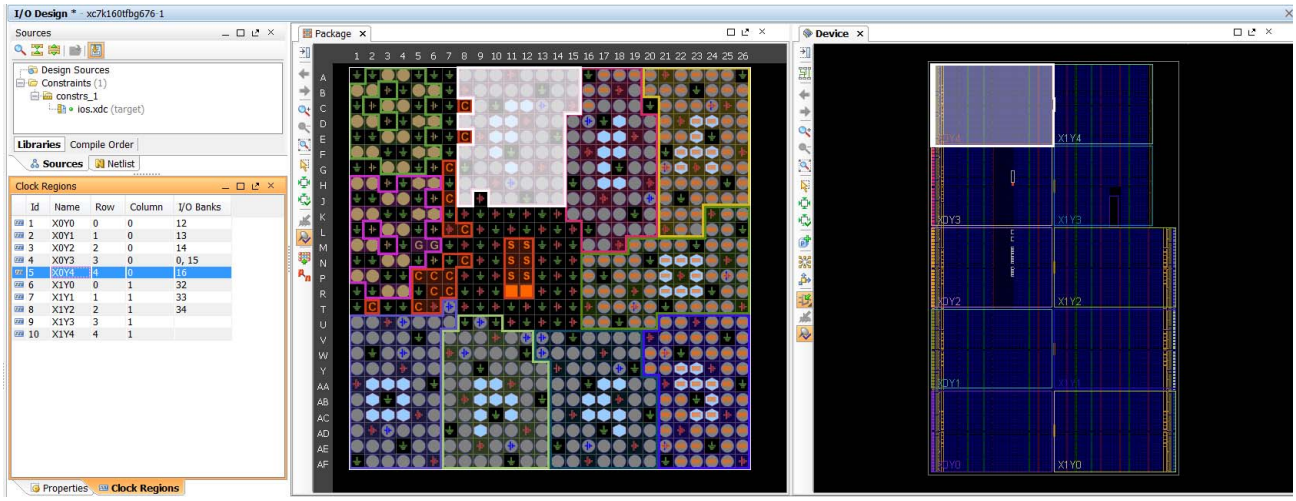

Figure 3-3: **Clock Regions Window**

When clock regions are highlighted, you can click the **Properties** tab to view the properties for the selected clock region. In the Clock Region Properties window, you can:

- Select the **Statistics** view to display the resource statistics available within the clock region as well as the logic content of the selected clock region.
- Select the **Resources** view to locate device clock resources for logic assignment, as shown in Figure 3-4.

Note: When you select an object in the Clock Regions Properties window, the object is cross-selected in another open window, such as the Device window.

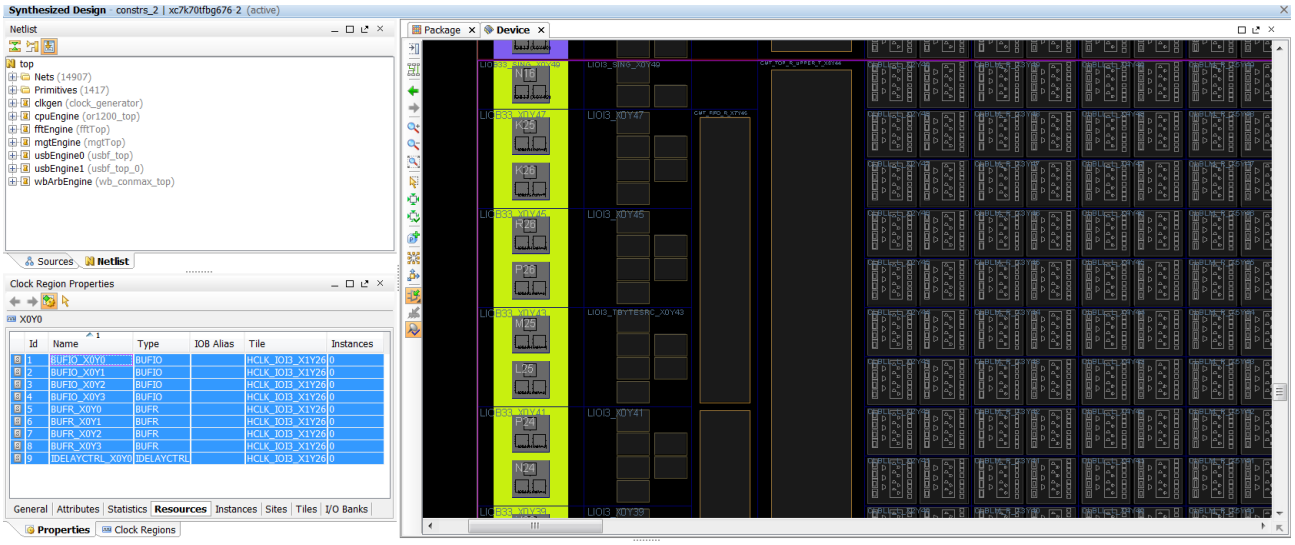

Figure 3-4: **Clock Region Properties—Resources View**

The Clock Resources window also provides a view of available clock resources to aid in planning and placing elements of global and regional clock trees. For information, see Using the Clock Resources Window.

I/O Bank Resources

You can select I/O resources in any of the I/O planning windows and their corresponding data is highlighted in all other windows as shown in Figure 3-3. This provides a visual indication of the relationship between the physical package and the internal die.

To access information about a specific I/O bank:

1. In the Package Pins window, select an I/O bank.
2. In the I/O Bank Properties window (Figure 3-5), click the views at the bottom of the window to see the different types of information available.

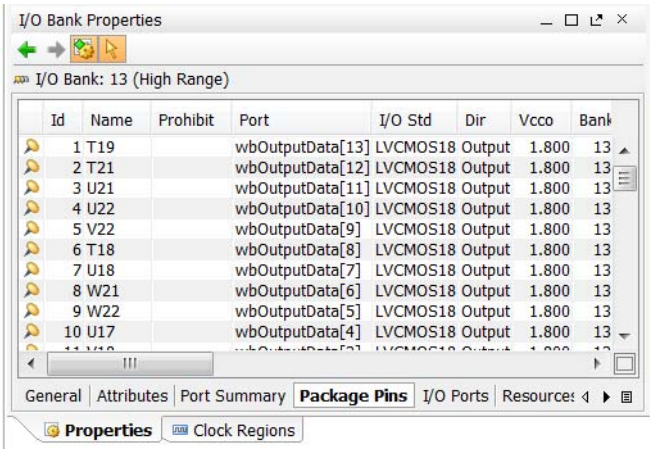

Figure 3-5: **I/O Bank Properties**

Multifunction Pins

The Package Pins window (Figure 3-6) contains several data types that display in columns similar to a spreadsheet. In this window, you can:

- Flatten, filter, and sort data.
- Move, hide, and configure columns to display and compare the various multifunction pins.
- Directly edit certain cells by entering text or selecting a value from drop-down menus.

The Package Pins window includes the following information:

- Type column that identifies multifunction pin types.
- Config column that shows the pin definition of the multifunction pin after you set the device configuration mode.

  Note: Many device configuration modes use multifunction pins. For more information, see Setting Device Configuration Modes.
- Other columns that contain information about logic or configuration modes involving multifunction type pins.
- Information that identifies conflicting multifunction pins for designs that contain GTs, memory controllers, or PCI™ logic.

In the Package window, the following symbols indicate the function of multifunction pins:

- Clock capable pins display as a hexagon shape
- V_{REF} pins display as a small power icon

Figure 3-6: **Package Pins Window**

Defining Alternate Compatible Parts

You can select compatible devices for the design to allow you to retarget the design to alternate Xilinx® FPGAs if necessary. This feature checks that I/O pin assignments are defined to work across selected alternate devices. Compatible Xilinx devices are selected in the same package as the current target part to preserve as much of the I/O assignment as possible.

To define an alternate compatible part:

1. Select **Tools > I/O Planning > Set Part Compatibility**.
2. In the Set Part Compatibility dialog box (Figure 3-7), select the alternate parts, and click **OK**.

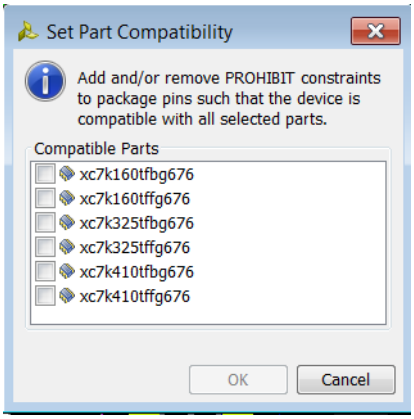

Figure 3-7: **Set Part Compatibility Dialog Box**

The Vivado IDE identifies the pins that are common to all selected alternate parts, and assigns `PROHIBIT` constraints to pins that are not common to all devices. The number of pins available for placement might be reduced when you select additional alternate parts.

In addition, the Vivado IDE automatically prohibits signals from being assigned to any unbonded pins in the selected alternate devices. A dialog box displays the number of prohibited package pins. You can view prohibits in the Package, Package Pins, and Device windows.

Setting Device Configuration Modes

To set device configuration modes as well as view and print information about the modes:

1. Select **Tools > I/O Planning > Set Configuration Modes**.
2. In the Set Configuration Modes dialog box (Figure 3-8), do the following, and click **OK**.
   - Select a configuration mode to view information, including a configuration diagram.
   - Click **Print** to print the configuration diagram.
   - Select the configuration modes.

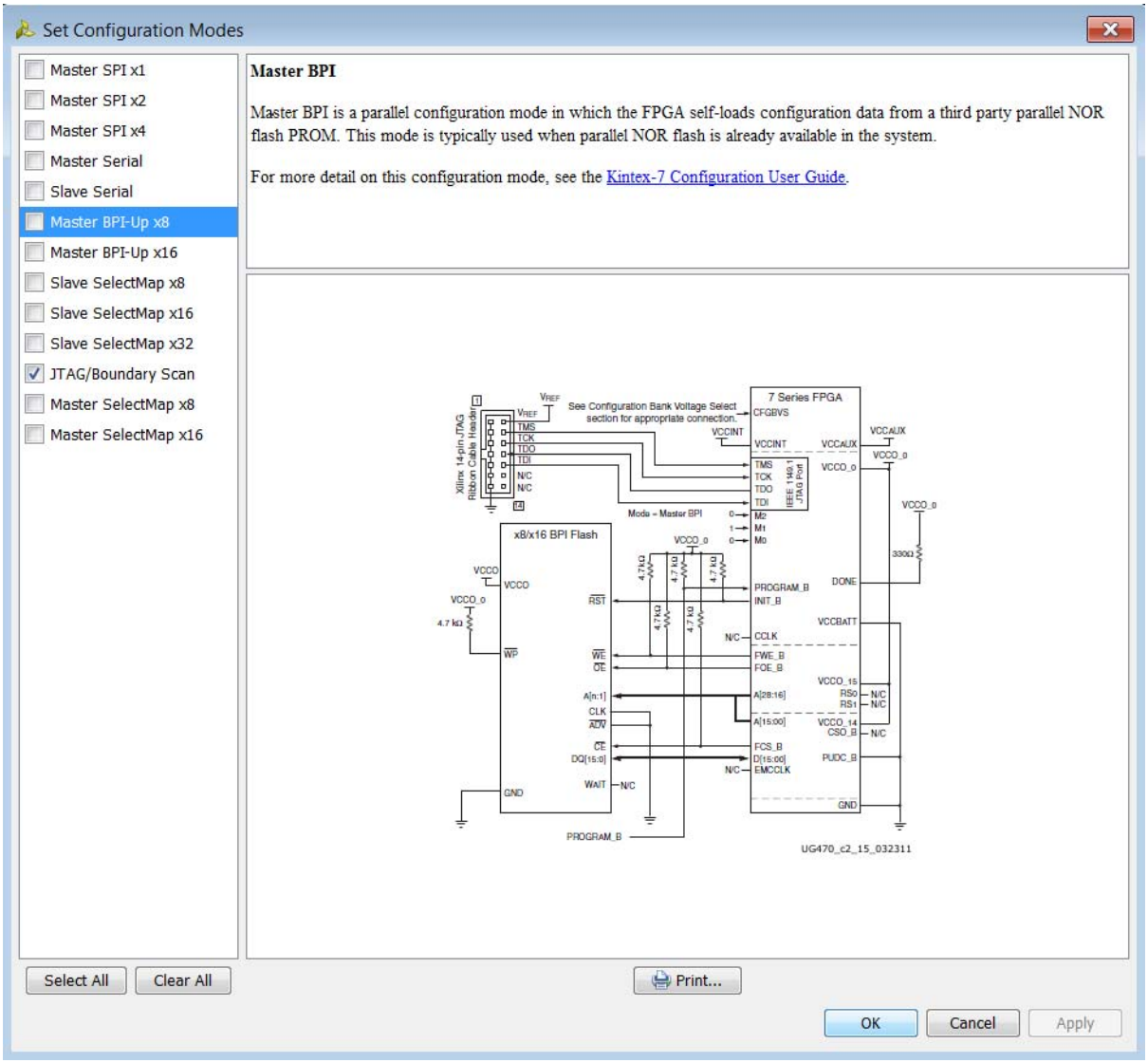

Figure 3-8: **Set Configuration Modes Dialog Box**

When you set the configuration modes, the associated I/O pins display in the Config column of the Package Pins window. The Vivado IDE prompts you to automatically sort the I/O pins using the Config column header.

Note: For more information on analyzing how the configuration modes might conflict with other multifunction pins, see Multifunction Pins.

Defining and Configuring I/O Ports

You can use the Vivado IDE to import, create, and configure I/O ports as described in the following sections.

Importing I/O Ports

Depending on the project type, you can use the following methods to import I/O ports:

- **I/O Planning Project**: You can import XDC and CSV files into an empty I/O planning project when you create the project or later using the file import capability. For details, see Importing a CSV File and Importing an XDC File.
- **RTL Project**: Use RTL files or headers to create an RTL project for I/O pin planning, then add more complete RTL source files to the project later as the design progresses. When you create an RTL-based or synthesized netlist-based project, the I/O Ports window automatically populates with the I/O ports defined in the design.
- **Migrate from I/O Planning Project to RTL Project**: You can convert an I/O planning project to an RTL project, turning the I/O ports into a top-level Verilog or VHDL module definition for the design. For more information, see Migrating to an RTL Design.

Importing a CSV File

You can import a CSV file to populate the I/O Ports window within the I/O Planning layout view. You can then assign these I/O ports to physical package pins to define the device pin configuration.

To import an I/O ports list from a CSV file:

1. Select **File > Import > Import I/O Ports**.
2. In the Import I/O Ports dialog box (Figure 3-9), select **CSV File**, and browse to select the file to import.

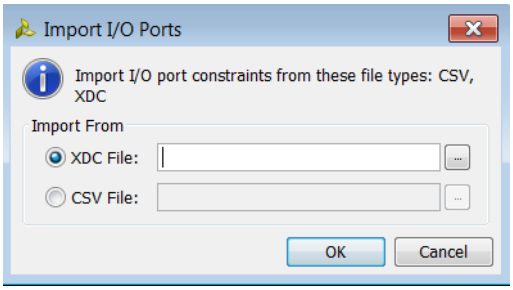

Figure 3-9: **Import I/O Ports Dialog Box**

Figure 3-10 shows the CSV file format. CSV is a standard file format used by FPGA and board designers to exchange information about device pins and pinout. The Vivado IDE requires a specific CSV file format for importing I/O pin-related data, as described in Appendix A, I/O Port Lists in CSV File Format.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
#Top: ios Floorplan: io_xc7k160tffg676-1 Part: xc7k160tffg676-1															
#Build: Vivado v2012.2															
IO Bank	Pin Numbe	IOB Alias	Site Type	Min Trace	Max Trace	Trace Le	Prohibit	Interface	Signal Nam	Direction	DiffPair Ty	DiffPair Sig	IO Standard	Drive (mA)	Slew Rate
14	A25	IOB_X0Y147	IO_L1N_T0_D01_DIN_14	191.467	193.391				port_1[0]	OUT			LVCMOS18	12	SLOW
14	B24	IOB_X0Y148	IO_L1P_T0_D00_MOSI_1	180.881	182.699				port_1[1]	OUT			LVCMOS18	12	SLOW
14	K21	IOB_X0Y149	IO_0_14	73.73	74.471				port_1[2]	OUT			LVCMOS18	12	SLOW
13	U16	IOB_X0Y50	IO_25_13	73.041	73.775				port_1[3]	OUT			LVCMOS18	12	SLOW
13	P18	IOB_X0Y51	IO_L24N_T3_13	73.003	73.737				port_1[4]	OUT			LVCMOS18	12	SLOW
13	R18	IOB_X0Y52	IO_L24P_T3_13	63.796	64.437				port_1[5]	OUT			LVCMOS18	12	SLOW
13	T17	IOB_X0Y53	IO_L23N_T3_13	73.336	74.073				port_1[6]	OUT			LVCMOS18	12	SLOW
13	U17	IOB_X0Y54	IO_L23P_T3_13	78.405	79.193				port_1[7]	OUT			LVCMOS18	12	SLOW
13	M19	IOB_X0Y55	IO_L22N_T3_13	100.104	101.11				port_1[8]	OUT			LVCMOS18	12	SLOW
13	N18	IOB_X0Y56	IO_L22P_T3_13	86.656	87.527				port_1[9]	OUT			LVCMOS18	12	SLOW
13	R17	IOB_X0Y57	IO_L21N_T3_DQS_13	66.334	67.001				port_1[10]	OUT			LVCMOS18	12	SLOW
13	R16	IOB_X0Y58	IO_L21P_T3_DQS_13	66.536	67.205				port_1[11]	OUT			LVCMOS18	12	SLOW
13	N17	IOB_X0Y59	IO_L20N_T3_13	65.725	66.385				port_1[12]	OUT			LVCMOS18	12	SLOW
13	P16	IOB_X0Y60	IO_L20P_T3_13	53.671	54.21				port_1[13]	OUT			LVCMOS18	12	SLOW
13	T19	IOB_X0Y61	IO_L19N_T3_VREF_13	53.96	54.503				port_1[14]	OUT			LVCMOS18	12	SLOW

Figure 3-10: **I/O Port List in CSV File Format**

You can define differential pairs in the CSV file in several ways. For example, the Vivado IDE recognizes differential pairs directly defined with DiffPair Signal and DiffPair Type attributes. In addition, the Vivado IDE can infer diff pairs when only one port of the pair is defined in the CSV file or two named nets imply a differential pair. For more information, see Differential Pairs in the CSV File in Appendix A.

When inferring differential pairs, the Vivado IDE displays a prompt to confirm the assignment of the pairs, as shown in Figure 3-11.

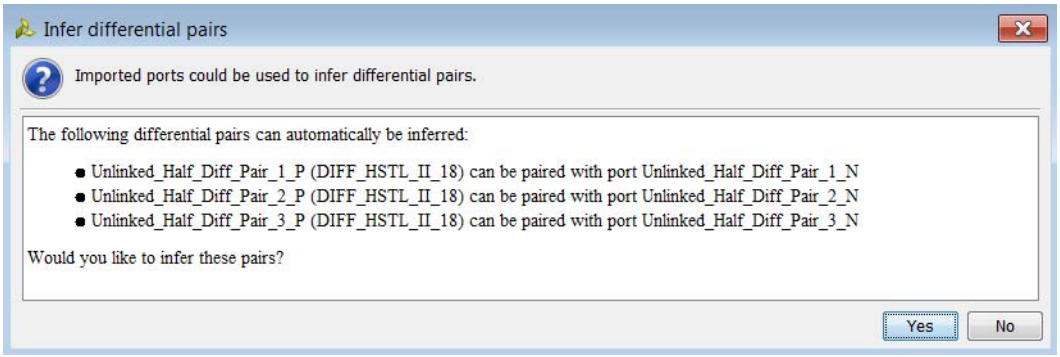

Figure 3-11: **Infer Differential Pairs Dialog Box**

CSV files can also contain additional information not recognized by the Vivado IDE. If unrecognized information is found in the imported CSV file, the information is displayed in user columns of the Package Pins window for your review and use. To modify or define values in the user CSV fields, select the **Set User Column Values** from the popup menu in the Package Pins window.

Note: For information on exporting a CSV file, see Exporting I/O Pin and Package Data.

Importing an XDC File

To import I/O port definitions from an XDC file:

1. Select **File > Import > Import I/O Ports**.
2. In the Import I/O Ports dialog box (Figure 3-9), select **XDC File**, and browse to select the file to import.

Because the XDC format does not define port direction, the direction is undefined. To define the I/O port direction, select **Set Direction** from the popup menu in the I/O Ports window. You can also directly modify the direction of a specific I/O port in the I/O Ports window. For more information, see Setting I/O Port Direction.

Creating I/O Ports

You can manually define new ports in an I/O planning project. Refer to Xilinx device documentation for information regarding voltage capabilities of the device.

To create I/O ports:

1. In the I/O Ports window, select **Create I/O Ports** from the popup menu.
2. In the Create I/O Ports dialog box (Figure 3-12), edit the following options, and click **OK**:
   - **Name**: Enter the port or bus name to create.
   - **Direction**: Select the port direction.
   - **Diff Pair**: Define differential pair signals or buses. This creates two ports and adds _P and _N to the specified name.
   - **Create Bus**: Enter a bus range for bus creation.
   - **I/O Standard**: Select the I/O standard constraint.
   - **Drive Strength**: Select the drive strength value.
   - **Slew Type**: Select the slew type value.
   - **Pull Type**: Select the pull type value.
   - **In Term Type**: Define the parallel termination attributes of the input signal.

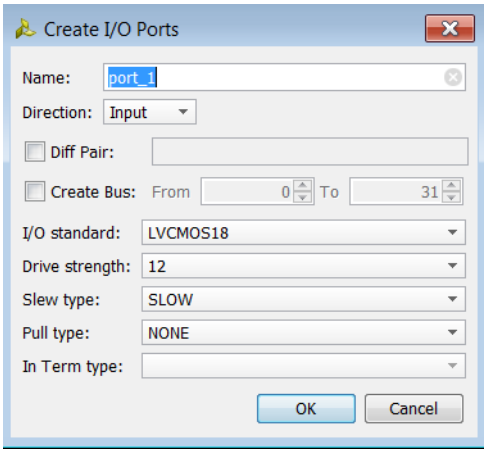

Figure 3-12: **Create I/O Ports Dialog Box**

Configuring I/O Ports

You can configure one or more I/O ports to define I/O standard, drive strength, slew type, pull type, and in term. This is useful to configure ports that were imported from CSV or XDC files without the appropriate characteristics. Refer to Xilinx device documentation for information regarding voltage capabilities of the device.

To configure a port or a group of ports:

1. In the I/O Ports window, select the ports.
2. Select **Configure I/O Ports** from the popup menu.
3. In the Configure Ports dialog box (Figure 3-13), edit the following options, and click **OK**:
   - **I/O Standard**: Select the I/O standard constraint. The tool does not check the I/O standard when it is assigned. You can assign any I/O standard to any port, but this might result in errors when running DRCs.
   - **Drive Strength**: Select the drive strength value.
   - **Slew Type**: Select the slew type value.
   - **Pull Type**: Select the pull type value.
   - **In Term Type**: Define the parallel termination attributes of the input signal.

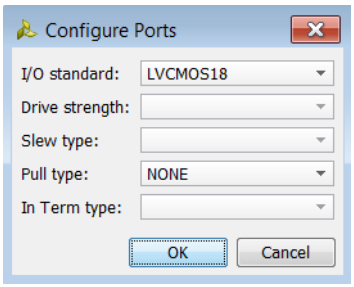

Figure 3-13: **Configure Ports Dialog Box**

Setting I/O Port Direction

To set I/O port direction, use any of the following methods:

- For I/O planning projects only, in the Direction (Dir) column of the I/O Ports window, click a port and change the direction using the drop-down menu.
- For I/O planning projects only, click a port in the I/O Ports window, and change the direction in the I/O Port Properties window.
- For I/O planning projects only, select the I/O ports, buses, or interfaces to be configured, and select **Set Direction** from the popup menu in the I/O Ports window.
- For RTL projects only, define the port direction in the RTL source.

Making and Splitting Differential Pairs

To define a differential pin pair in an I/O planning project:

1. Select any two I/O ports, and select **Make Diff Pair** in the popup menu of the I/O Ports window.

IMPORTANT: *The Make Diff Pair option is not available in RTL projects. In RTL projects, differential ports must be defined in the source code using appropriate I/O buffer instantiations.*

In the Make I/O Diff Pair dialog box (Figure 3-14), the two I/O Ports display with Positive End and Negative End assignments made by the tool.

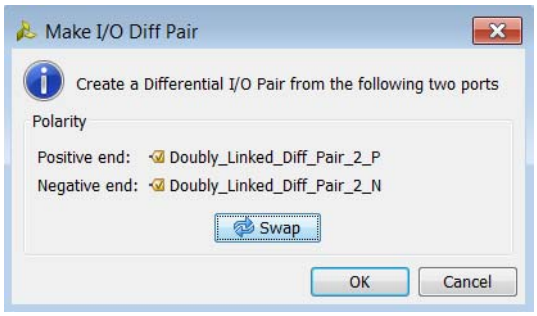

Figure 3-14: **Make I/O Diff Pair Dialog Box**

2. To reverse the Positive End and Negative End signals, click **Swap**, and click **OK**.

TIP: *Use the **Split Diff Pair** command from the popup menu to separate a diff pair into two ports.*

Configuring DCI_CASCADE Constraints

The DCI_CASCADE constraint links two or more adjacent I/O banks together for DCI reference voltage purposes. The I/O bank with the DCI reference voltage is the *master*. All other I/O banks in the cascade are *slaves*. All banks in a cascade must be in the same I/O column of the device. The following sections describe how to set the constraint.

Note: You can configure the DCI_CASCADE constraint for Xilinx 7 series devices. For more information on this constraint, see the *Constraints Guide (UG625)*.

Creating a DCI_CASCADE Constraint

To create a DCI_CASCADE constraint:

1. In the Package Pins window or Package window, select the I/O banks to configure, and select **Create a DCI Cascade** from the popup menu.

   In the DCI Cascade Editor (Figure 3-15), the selected I/O banks appear.

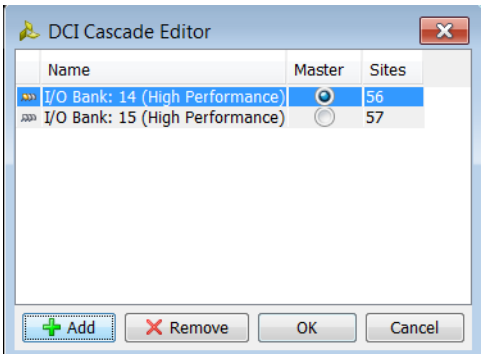

Figure 3-15: **DCI Cascade Editor**

2. To include additional I/O banks, click **Add**.

   Select I/O banks from the same column on the device. The tool does not check this interactively when you create the DCI_CASCADE constraint. However, the tool can check for errors when you run DRCs.

3. Select an I/O bank to be the **Master**, and click **OK**.

The DCI cascades are displayed in the Physical Constraints window (Figure 3-16). If you select a **DCI Cascade** in the Physical Constraints window, the I/O banks are highlighted in associated windows, such as the Package and Device windows.

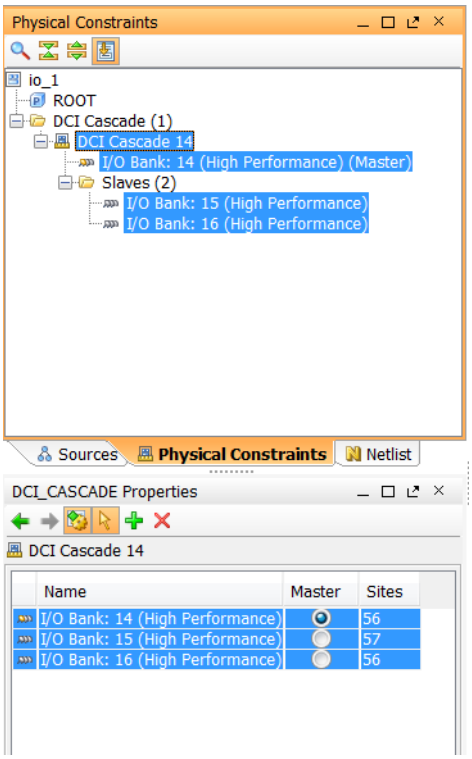

Figure 3-16: **Physical Constraints Window**

Modifying or Removing DCI Cascade Constraints

To modify DCI cascades, select the **DCI Cascade** in the Physical Constraints window, and update the settings in the DCI_CASCADE Properties window as follows:

- To change the Master, select a different I/O bank and assign it as the **Master**.
- To remove I/O banks from the DCI cascade, select the I/O banks in the DCI_CASCADE Properties window, and click the **Delete I/O Banks** toolbar button .
- To include additional I/O banks in the DCI cascade, select the I/O banks in the DCI_CASCADE Properties window, and click the **Add I/O Banks** toolbar button . In the Select I/O Banks dialog box, you can select new I/O banks. The newly-selected I/O banks also highlight in the other windows.
- To save all changes in the DCI_CASCADE Properties window, click **Apply**.

TIP: *To remove DCI Cascade constraints, right-click the constraint in the Physical Constraints window, and select* ***Delete****.*

Prohibiting I/O Pins and I/O Banks

The I/O Planning view layout provides an interface to selectively prohibit port placement onto individual I/O pins, groups of I/O pins, or I/O banks. You can select and prohibit pins in the Device, Package, and Package Pins windows.

To prohibit I/O pins or I/O banks:

1. Select the I/O pins or I/O banks in the Device, Package, or Package Pins window.
2. Select **Set Prohibit** from the popup menu.

Prohibited pins are indicated by a:

- Slashed circle in the Device window and Package window (Figure 3-17)
- Check mark in the Prohibit column of the Package Pins window

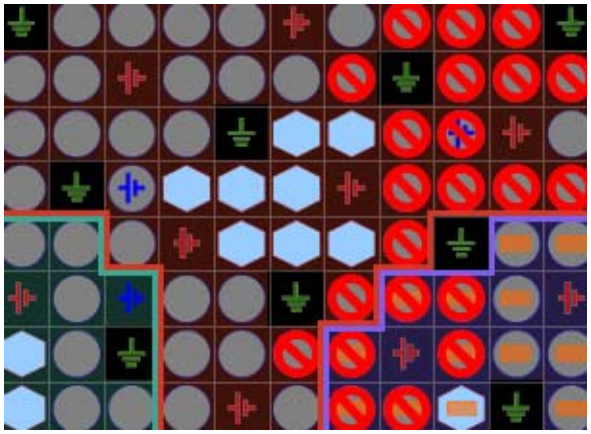

Figure 3-17: **Prohibits on Package Pins**

Creating I/O Port Interfaces

To group multiple ports or buses together, you can create an interface. This aids in pin assignment by treating all of the interface ports as one group. Assigning all of the pins simultaneously helps condense and isolate the interface for clock region or PCB routing. This also makes it easier to visualize and manage the signals associated with a particular logic interface.

To create an interface:

1. In the I/O Ports window, select the signals to group together.
2. From the popup menu, select **Create I/O Port Interface**.
3. In the Create I/O Port Interface dialog box (Figure 3-18), enter a name for the interface, adjust assignment selection, and click **OK**.

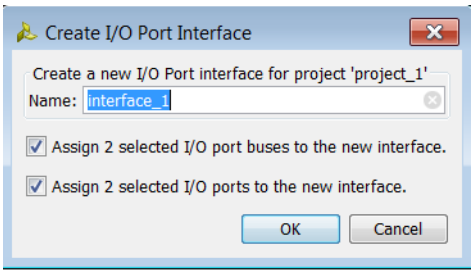

Figure 3-18: **Create I/O Port Interface Dialog Box**

The interfaces appear as expandable folders in the I/O Ports window (Figure 3-19).

Figure 3-19: **I/O Port Interfaces in I/O Ports Window**

TIP: *To delete interfaces, select an interface, and select* ***Delete*** *from the popup menu, or press the* ***Delete*** *key.*

Adding I/O Ports to an Interface

To add I/O ports to an interface, do either of the following in the I/O Ports window:

- Select the I/O ports, and drag them into the interface folder.
- Right-click a port or bus, and select **Assign to Interface**. In the Select I/O Port Interface dialog box, select the target interface.

Removing I/O Ports from an Interface

To remove I/O ports, do the following in the I/O Ports window:

1. Right-click a port.
2. From the popup menu, click **Unassign from Interface**.

Disabling or Enabling Interactive DRCs

During I/O planning, the Vivado IDE checks to ensure a legal pinout. However, complete sign-off DRCs are only run during Vivado implementation. Therefore, you need to run your design through Vivado implementation to ensure final legal pinouts.

The interactive I/O placement routines check common error cases during pin placement. You can toggle this capability on and off using either of the following methods:

- In the Device window or Package window, click the **Autocheck I/O Placement** toolbar button .
- Select **Tools > Options**. In the General Options, enable the **Automatically enforce legal I/O placement** option.

When you enable automatic checking, the tool does not allow placement of I/O ports on pins that cause a design issue. In **Place I/O Ports Sequentially** mode, if you attempt to place an I/O Port on a problematic pin, a tooltip appears that describes why the I/O port cannot be placed. The interactive DRCs are enabled by default.

IMPORTANT: *Many of these checks can run only when a synthesized netlist of the full design is loaded.*

The interactive I/O placement rules include:

- **Prohibiting**:
  - Placement on noise-sensitive pins associated with GTs or on I/O package pins that are potentially noise-sensitive.
  - I/O standard violations.
- **Ensuring**:
  - I/O standards are not used in banks that do not support them.
  - Banks do not have incompatible V_{CC} ports assigned.
  - Banks that need V_{REF} ports have free V_{REF} pins.
  - Proper assignment of global clocks and regional clocks (only with an imported netlist and XDC file).
  - Differential I/O ports are set to the proper sense pin.
  - No output pins are placed on input-only pins.

RECOMMENDED: *It is recommended that you begin your I/O port placement with DRCs enabled. For more information on I/O-related DRCs, see Appendix B, I/O Port DRC Rule Descriptions.*

Placing I/O Ports

The I/O Planning view layout provides several ways to assign I/O ports to package pins. You can select individual I/O ports, groups of I/O ports, or interfaces in the I/O Ports window, and assign them to package pins in the Package window or to I/O pads in the Device window.

In the Package window, you can:

- Drag and drop ports to package pins.
- View port placement and constraints.
- Move the cursor over the pins to show the I/O pin coordinates on the top and left sides of the window.
- Hold the cursor over a pin to show a tooltip that displays the pin information.

Note: Additional I/O pin and bank information displays in the status bar located at the bottom of the Vivado IDE.

Placing I/O Ports Sequentially

To place I/O ports sequentially:

1. In the I/O Ports window, select an individual I/O port, a group of I/O ports, or interfaces.
2. Use one of the following commands:
   - In the I/O Ports window, select **Place I/O Ports Sequentially** from the popup menu.
   - In either the Package window or the Device window, click the **Place Ports** toolbar button, and select **Place I/O Ports Sequentially** .

   The first I/O port in the group is attached to the cursor when you move it over a package pin or I/O pad. A tooltip displays the I/O port and package pin names.
3. To assign an I/O port, click a pin or a pad.

   If you select more I/O ports, the command is continued. The cursor drags the next I/O ports and so on until all of the I/O ports are placed, or you press **Esc**.

TIP: *The Vivado IDE assigns ports in the order that they appear in the I/O Ports window. You can adjust the assignment order by applying sorting techniques in the I/O Ports window prior to assignment.*

Figure 3-20 shows I/O ports placed sequentially.

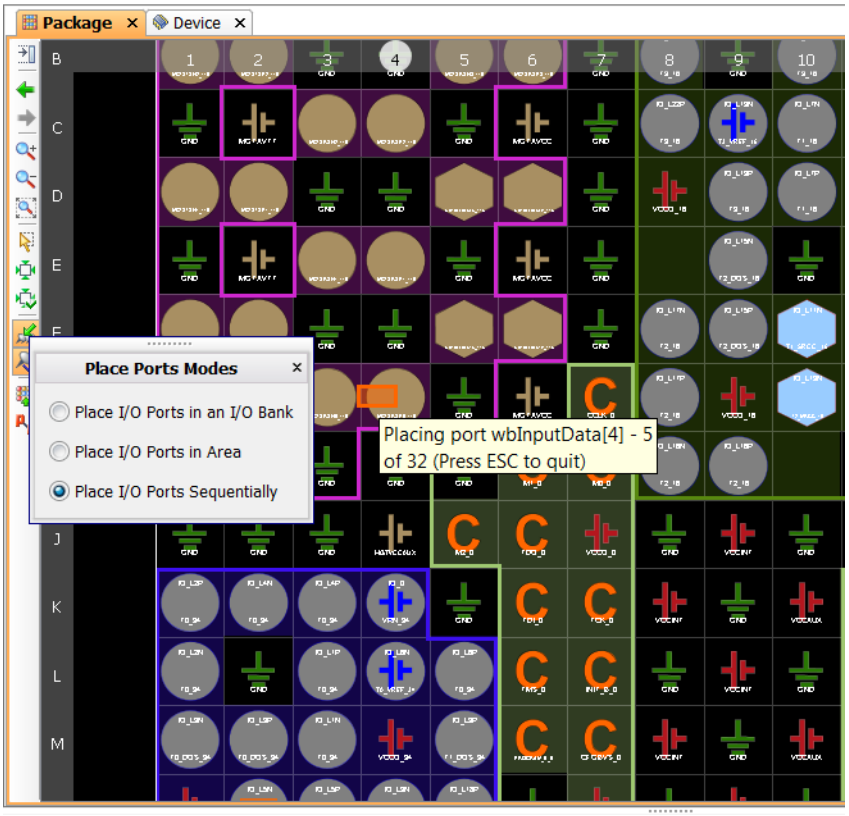

Figure 3-20: **I/O Ports Placed Sequentially**

Placing I/O Ports into I/O Banks

To place I/O ports into I/O banks:

1. In the I/O Ports window, select an individual I/O port, a group of I/O ports, or interfaces.
2. Use one of the following commands:
   - In the I/O Ports window, select **Place I/O Ports in an I/O Bank** from the popup menu.
   - In either the Package window or the Device window, click the **Place Ports** toolbar button, and select **Place I/O Ports in an I/O Bank** .

   The group of I/O ports is attached to the cursor when it is dragged over a package pin or I/O pad. A tooltip shows the number of pins that can be placed in the selected I/O bank.

3. Click a pin or pad to assign the selected I/O ports.

   If more I/O ports are selected than fit in the I/O bank, the Vivado IDE places as many as possible in the selected I/O bank, then lets you select another I/O bank into which to place the remaining ports. The cursor drags the remaining I/O ports to the next selected I/O bank and so on until all of the I/O ports are placed, or you press **Esc**.

TIP: *The Vivado IDE assigns ports in the order that they appear in the I/O Ports window. You can adjust the assignment order by applying sorting techniques in the I/O Ports window prior to assignment.*

Port assignment to device resources is also driven from the initial selection from the I/O bank. Selecting a pin at one end of an I/O bank results in a continuous bus assignment across the I/O bank.

The Vivado IDE also keeps track of PCB routing concerns for buses. Pin ordering during assignment attempts to keep the bus bits vectored within the assignment area. You can customize assignment patterns to address other bus routing concerns.

Figure 3-21 shows I/O ports placed in an I/O bank.

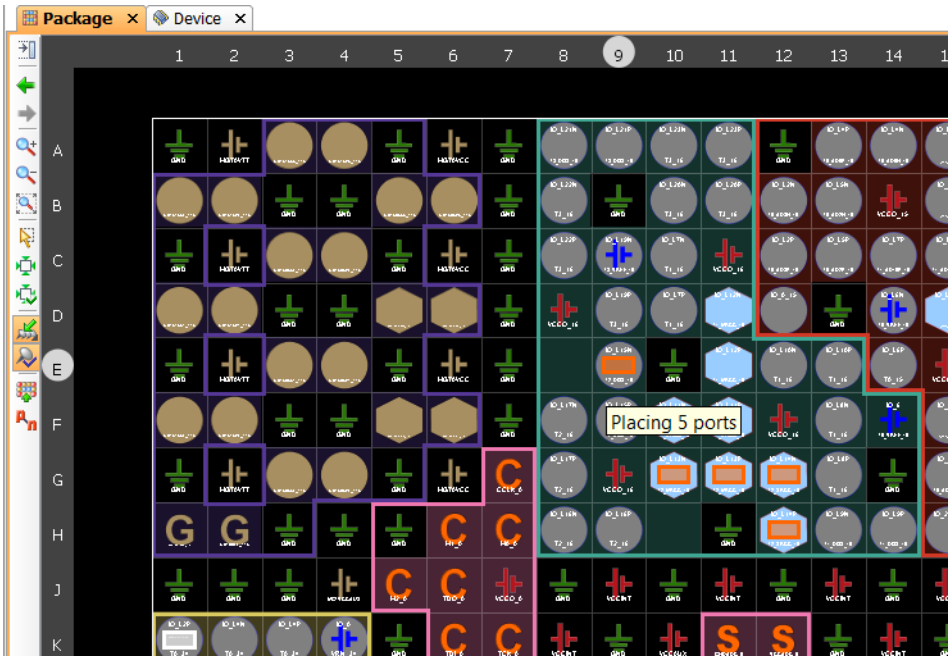

Figure 3-21: **I/O Ports Placed in I/O Banks**

Placing I/O Ports in a Defined Area

To place I/O Ports into a defined area:

1. In the I/O Ports window, select individual I/O ports, groups of I/O ports, or interfaces.
2. Use one of the following commands:
   - In the I/O Ports window, select **Place I/O Ports in Area** from the popup menu.
   - In either the Package window or the Device window, click the **Place Ports** toolbar button, and select **Place I/O Ports in Area** .

   The cursor turns into a cross symbol, which indicates that you can define a rectangle for port placement.
3. In either the Package window or the Device window, draw a rectangle to define the assignment area.

   If you select more I/O Ports than fit in the defined area, the command is continued. The cursor continues to display as a cross to draw another area to place the remaining I/O ports until all of the I/O ports are placed, or you press **Esc**.

TIP: *The Vivado IDE assigns ports in the order that they appear in the I/O Ports window. You can adjust the assignment order by applying sorting techniques in the I/O Ports window prior to assignment.*

The direction in which you draw the rectangle dictates the I/O ports assignment order. I/O ports are assigned from the inside pin of the first rectangle coordinate selected. Creative definition of the area rectangles can provide useful pinout configurations from a PCB routing perspective.

Figure 3-22 shows I/O ports placed in an area.

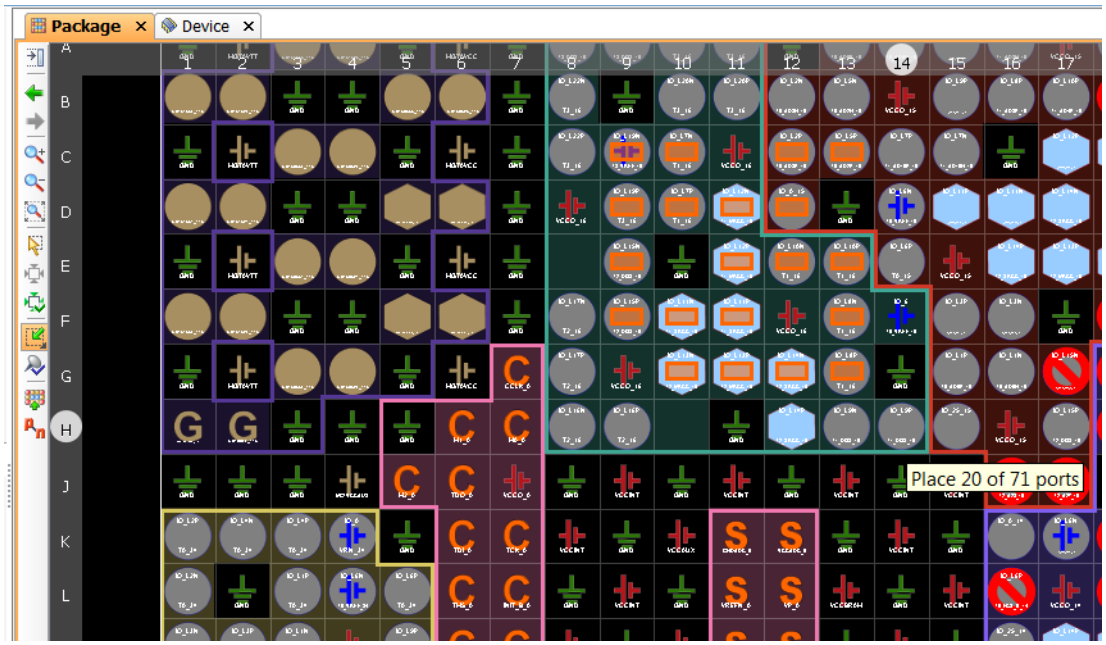

Figure 3-22: **I/O Ports Placed in an Area**

Swapping Previously Placed I/O Ports

To swap the location of two placed I/O ports that are already assigned:

1. Select two I/O ports from any of the available windows.
2. Select **Swap Locations** from the popup menu.

IMPORTANT: *If you are working in an implemented design and you swap two ports that are not yet fixed, swapping the ports fixes the ports and writes constraints to the XDC file.*

Moving Previously Placed I/O Ports

To move a port or groups of ports that are already assigned, select the port or group of ports, and drag them from one location to another. When you move a group of ports from one I/O bank to another, the Vivado IDE automatically finds suitable locations for the selected ports.

Note: This is similar to using the **Place I/O Ports in an I/O Bank** command.

Automatically Placing I/O Ports

You can automatically assign I/O ports to package pins. The Vivado IDE obeys I/O standard and differential pair rules and places global clock pins appropriately.

To automatically assign I/O ports:

1. In the I/O Ports window, select the I/O ports to place.
2. Select **Tools > I/O Planning > Auto-place I/O Ports**.

   Note: Alternatively, you can select **Auto-place I/O Ports** from the popup menu in the I/O Ports window.
3. In the Autoplace I/O Ports wizard (Figure 3-23), select the group of I/O ports to place, and click **Next**.

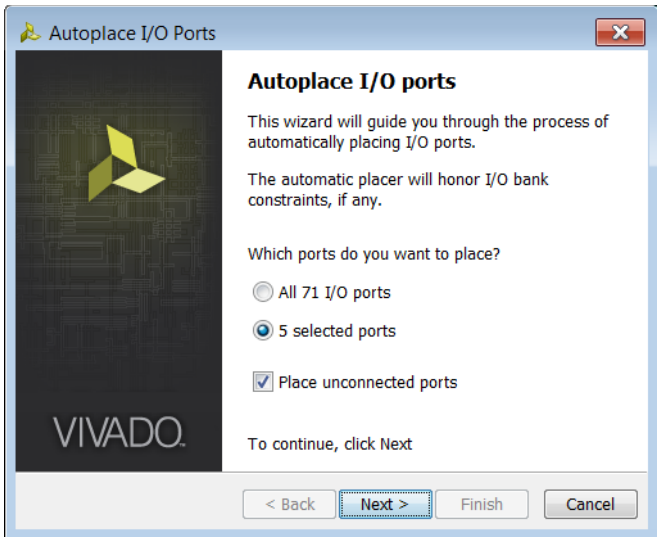

Figure 3-23: **Autoplace I/O Ports Wizard**

4. If you selected I/O ports that are already assigned to package pins, select an option in the Placed I/O Ports page (Figure 3-24), and click **Next**.

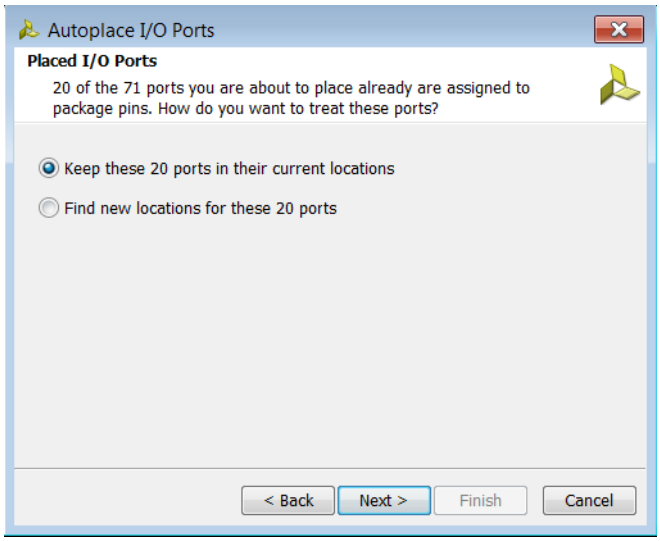

Figure 3-24: **Autoplace I/O Ports Wizard—Placed I/O Ports Page**

5. In the Summary page, click **Finish**.

Placing Gigabit Transceiver I/O Ports

To better manage GTs, the I/O planning windows group the two related I/O diff pairs and the GT logic object automatically during selection, placement, and moving. The GT objects are selected as one object and move together, which prohibits illegal assignment of the GT resources.

If the interactive DRCs are enabled, the noise sensitive I/O pins surrounding the GTXs are prohibited automatically during port placement. For more information, see Disabling or Enabling Interactive DRCs.

Removing I/O Placement Constraints

To remove placement constraints, select the placed logic, and select **Unplace** from the popup menu.

Placing Clock Logic

You can manually place global and regional clock-related logic, such as BUFGCTRLs, MMCMs, BUFRs, and IDELAYCTRLs, using the Clock Resources window as described in Using the Clock Resources Window. You can also manually place clock logic in the Device window. Appropriate logic sites are displayed in the Device window for all device-specific resources.

Locating Logic Instances

To locate logic instances for placing onto the device:

1. Select **Edit > Find**.
2. In the Find dialog box, specify **Instances** in the Find field, and define the criteria to locate the specific logic instance or instances.
3. From the Find Results window, drag logic instances onto the Clock Resources window or the Device window to assign to the appropriate device resource.

TIP: *You can also locate physical resources on the device, such as global clock buffers, for placing logic instances. Specify **Sites** in the Find field, and define the criteria as needed. Select a result in the Find Results window to highlight the physical device resource in the Clock Resources window or Device window.*

Using the Clock Resources Window

The Clock Resources window shows the relationship and interactions between regional and global clocking resources: BUFRs, BUFIOs, BUFGs, MMCMs and GTs. The Clock Resources window shows a simplified view of the device resources but maintains proper relative positioning between these resources.

Note: Most of the details of the FPGA device shown in the Device window are not shown in the Clock Resources window.

Figure 3-25 shows the Clock Resources window for a Kintex™-7 K70T device, which indicates:

- The device has eight clock regions arranged in a 4x2 matrix, numbered from X0Y0 in the lower-left of the device to X1Y3 in the upper-right.
- Each of these clock regions also has I/O banks containing clock-capable pins (CCIOs), BUFIOs, and BUFRs that display in the Clock Resources window for each clock region.
- The device itself is divided into a top "half" containing four clock regions, and a bottom "half" containing four clock regions.
- The BUFGs for managing global clocks on the device are in the center column of the device.

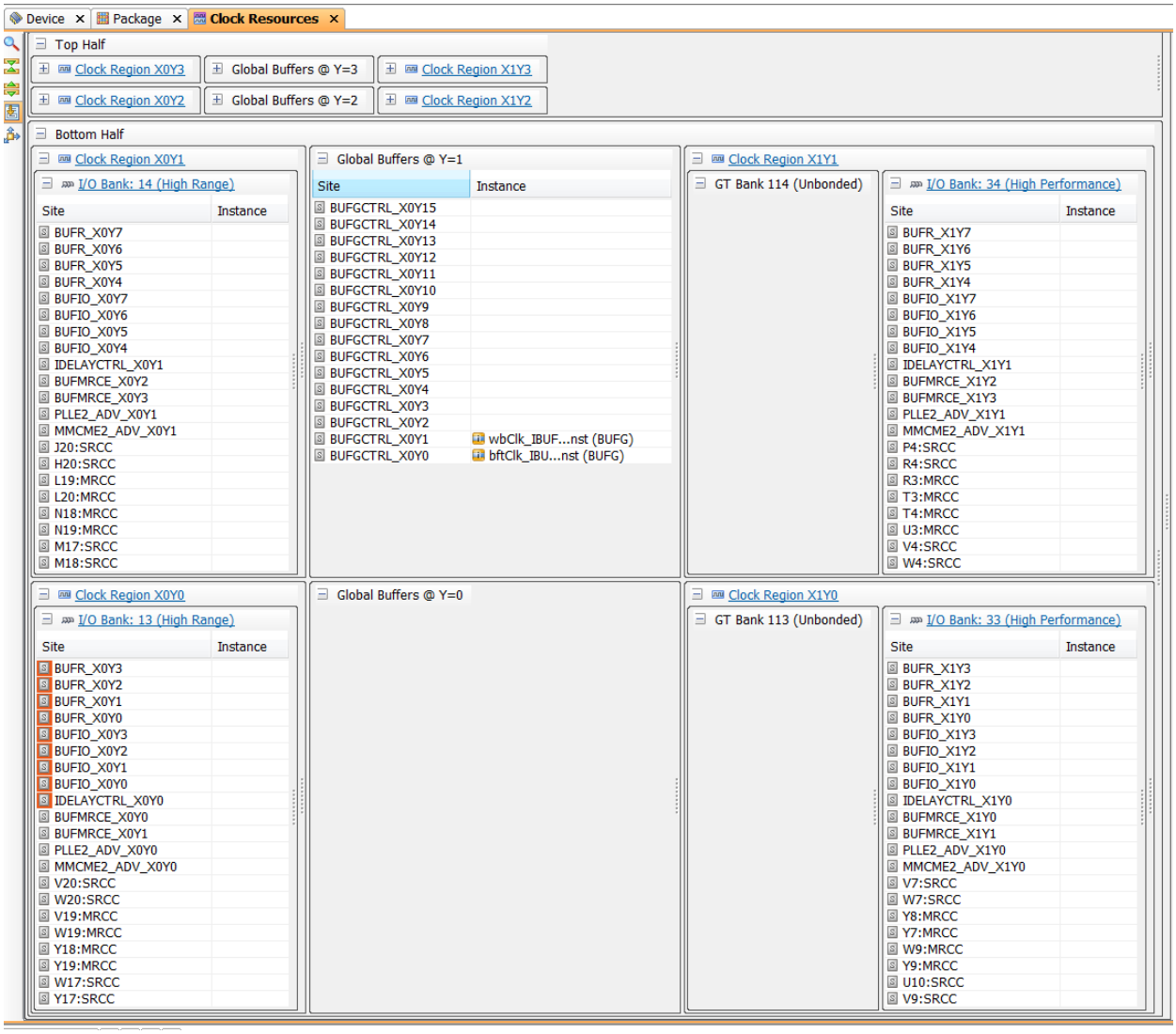

Figure 3-25: **Clock Resources Window**

Collapsing and Expanding Levels

To expand or collapse levels to display only the information of interest:

- Click the expand (+) and collapse (-) buttons to expand or collapse portions of the tree.
- In the local toolbar, use the **Expand All** and **Collapse All** buttons to expand or collapse the entire tree.

Cross-Selecting Objects

To cross-select objects between windows, do either of the following in the Clock Resources window:

- Click the name of a specific clock region or I/O bank.

  Use this method to locate a specific object on the device, on the package, or in the netlist.

- Click the **Automatically Scroll to Selected Object** toolbar button to scroll to an object that is selected in another window.

  Use this method to locate a specific resource on the device in the Clock Resources window.

TIP: *You can toggle off automatic scrolling to prevent the Vivado IDE from changing the displayed resources every time an object is selected in a different window.*

Placing Design Instances

The Clock Resources window displays two columns: Site and Instance. These columns report both the device resource and the design instance to which it is assigned.

To place design instances:

1. In the Find Results, Schematic, Netlist, or I/O Ports windows, select logic instances from the design to place into the device resources.
2. Drag the instances onto the Instance column of the appropriate device resource in the Clock Resources window.

   As you drag the instance in the Clock Resources window, tooltips indicate sites where the instance cannot be placed with a slashed circle symbol and sites where the instance can be placed with a rectangle.

IMPORTANT: *When you place instances from the design, the Vivado IDE enforces specific rules and limitations regarding global and regional clock tree structures. For specific information on these rules and limitations, refer to the FPGA Clocking Resources User Guide for the targeted device.*

Examining Clock Connections

To examine the clock connections graphically, right-click the site in the Clock Resources window, and select any of the following commands from the popup menu:

- **Select Drivers**
- **Select Loads**
- **Show Clock Connections**

Placing Clock Logic in the Device Window

To place clock logic manually:

1. In the Device window, zoom to locate the appropriate device site to place the logic.
2. Select the **Instance Drag & Drop Modes** toolbar button, and select **Create Site Constraint Mode** .
3. Select the logic instance to place from the Find Results, Schematic, Netlist, or I/O Ports window, and drag it onto the appropriate device resource in the Device window.

Validating I/O and Clock Logic Placement

This section describes the required steps for running I/O port and clock-related DRCs and viewing DRC violations.

Running I/O Port and Clock Logic Related DRCs

To select and run individual rules:

1. Select **Tools > Report DRC**.

   Note: Alternatively, you can click **Report DRC** from the Flow Navigator.

2. In the Report DRC dialog box (Figure 3-26), enter a name in the Results Name field.

   Entering a unique name makes it easier to identify the results for a particular run during debug in the DRC Violations browser. The output file name matches the entered name.

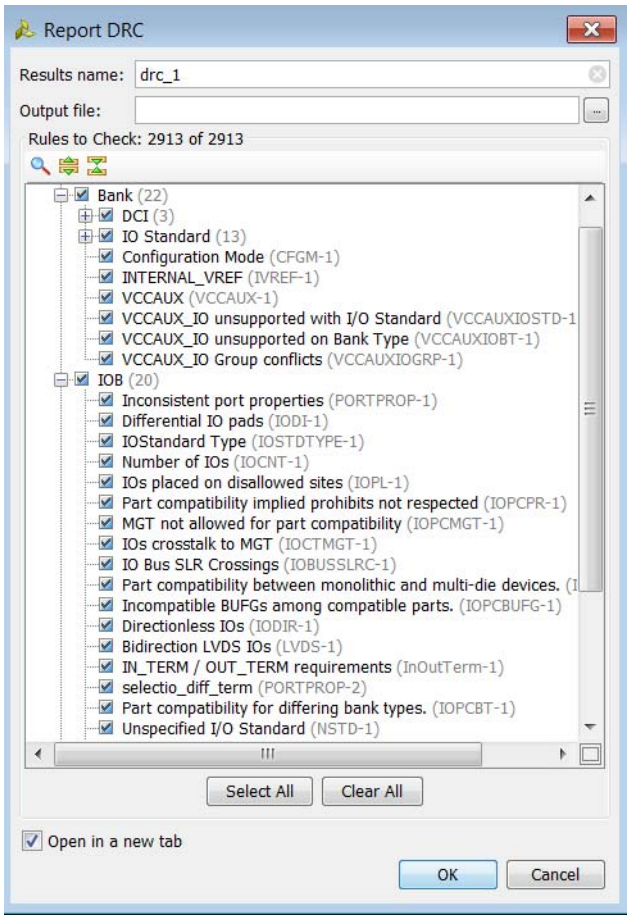

Figure 3-26: **Report DRC Dialog Box**

3. In the Rules to Check field, use the checkboxes to select the design rules to check for each design object. In this field, you can:
   - Expand the hierarchy using the **Expand All** toolbar button, or click the expand (+) button next to each category or design object.
   - Click **All Rules** to run a complete DRC, which runs all rules for all design objects.
   - Click a design object to run all DRCs for that object.
   - Click individual DRCs.

   Note: For information about each rule, see Appendix B, I/O Port DRC Rule Descriptions.
4. Click **OK** to run the selected DRCs.

Viewing DRC Violations

If violations are found, the DRC window opens, as shown in Figure 3-27. The DRC window shows the rule violations, grouped under the various rule categories defined in the Report DRC dialog box. The rule violations are also categorized by severity and are color coded for quick review as follows:

- **Informational only**: Provides general status and feedback on design processing.
- **Warning**: Indicates that design results might be sub-optimal, because constraints or specifications might not be applied as intended.
- **Critical warning**: Indicates that certain user input or constraints will not be applied or do not adhere to best practices. It is highly recommended that you examine these issues and make changes.
- **Error**: Indicates an issue that renders design results unusable and cannot be resolved without your intervention. The design flow stops.

TIP: *You can toggle the **Hide Warnings and Informational Messages** toolbar button to turn off warnings and informational messages to see only the errors reported.*

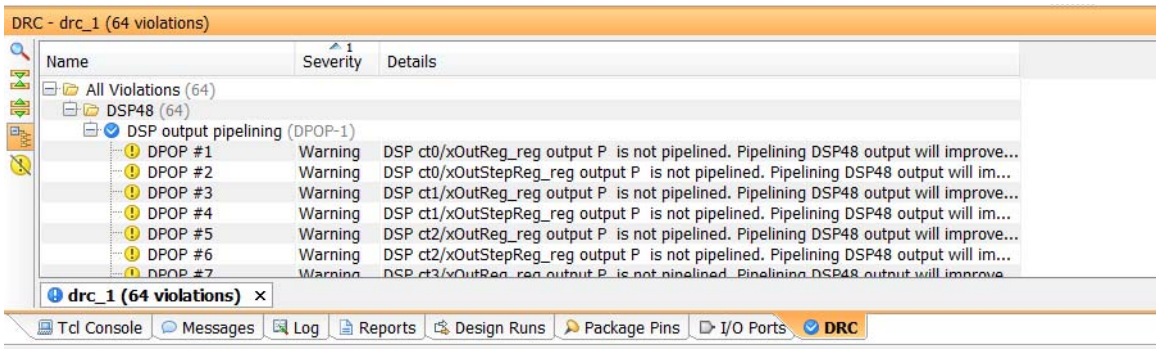

Figure 3-27: **DRC Violations**

Sorting DRC Violations

To sort DRC violations by severity, click the Severity column header as follows:

- Click the column header to sort in an increasing order.
- Click the column header again to sort in a decreasing order.

Note: For more information, see Using Data Table Windows in the *Vivado Design Suite User Guide: Using the Vivado IDE (UG893)*.

Viewing DRC Violation Properties

In the DRC window, right-click a violation message, and select **Violations Properties** from the popup menu. In the Violation Properties window, click the following views:

- **General**: Provides high-level information about the DRC rule violation, including type, severity, and description.
- **Details**: Provides information about the design elements that violate the rule. In addition, this view might include links to specific design objects that violate the DRC. Click these links to view the design object in the RTL Netlist window, Device window, Schematic window, or source RTL file.

Migrating to an RTL Design

After the I/O ports are defined and placed onto the package pins, you can migrate the I/O planning project to an RTL project. The port definitions are used to create a top module for the RTL design in either Verilog or VHDL, as specified. Differential pair buffers are added to the top module, and bus definitions are also included in the RTL. The project attributes are changed to reflect the RTL project type.

IMPORTANT: *After migration, the RTL project cannot be converted back into an I/O planning project.*

To convert the project:

1. Select **File > Migrate to RTL**.

   Note: Alternatively, you can select **Migrate to RTL** from the Flow Navigator.

2. In the Migrate to RTL dialog box (Figure 3-28), set the following options, and click **OK**.
   - **Top RTL file**: Specify the Verilog (.v extension) or VHDL (.vhd extension) file to create for the top module of the design. The HDL file will include the module definition with port definitions, direction, and width for bus pins.
   - **Netlist format**: Specify Verilog or VHDL format for the top module.
   - **Write diff buffers**: Write the diff pair buffers as part of the top module definition. This preserves any differential pairs defined in the I/O planning project.

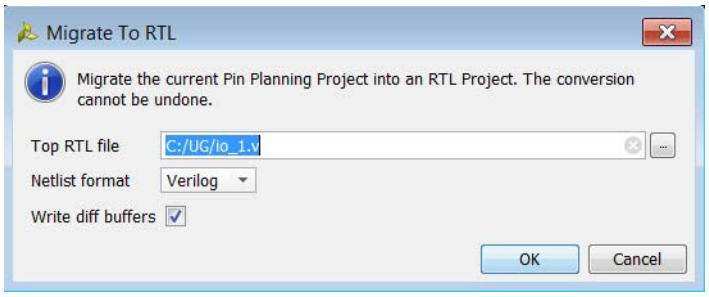

Figure 3-28: **Migrate to RTL Dialog Box**

After the I/O planning project is converted to an RTL project, you can begin adding sources to the project and working on your design. For more information, see the *Vivado Design Suite User Guide: System-Level Design Entry (UG895)*.

Exporting I/O Pin and Package Data

You can export I/O pin and package pin information for the following purposes:

- **I/O Pin Information**: You can export the I/O port list to a file for use in RTL coding or PCB schematic symbol creation.
- **Package Pin Information**: When working with an elaborated, synthesized, or implemented design, you can export the device package pin information to a CSV file. The package pin section of the exported list is a good starting point for defining I/O port definitions in a spreadsheet format. The exported information includes information about all of the package pins in the device as well as design-specific I/O port assignments and their configuration. Added columns and user-defined values are preserved and exported to the output file. For information on the exported CSV file format, see Defining and Configuring I/O Ports.

To export the I/O ports list information:

1. Select **File > Export > Export I/O Ports**.
2. In the Export I/O Ports dialog box, specify the type of I/O port to generate and the path, and click **OK**.

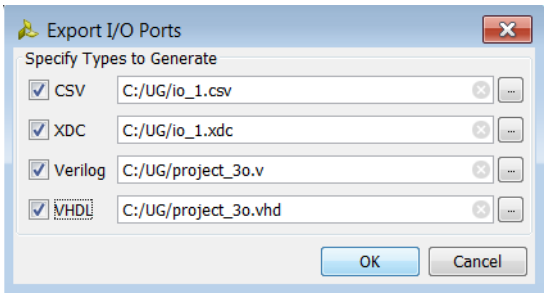

Figure 3-29: **Export I/O Ports Dialog Box**

Exporting IBIS Models

To better understand the signal integrity at the system level, PCB designers often need to simulate the design with IBIS models. Designers must consider signal integrity issues such crosstalk, ground bounce, and SSN. IBIS models help characterize current-voltage (I-V) curves and parasitic information of the packaged device.

TIP: *You can download generic IBIS models from the Downloads page of the Xilinx website.*

From the Vivado IDE, you can generate IBIS models from the design and per-pin package data. The Vivado IDE uses the netlist and implementation details from the design, and combines that information with the available per-pin parasitic package information to create a custom IBIS model for the design.

With an elaborated, synthesized, or implemented design open, you can export an IBIS file for use in analyzing the design as follows:

1. Select **File > Export > Export IBIS Model**.
2. In the Export IBIS Model dialog box (Figure 3-30), set the following options, and click **OK**:
   - **Output File:** Specify the filename and path for the output IBIS file.
   - **Include all models**: Include all available I/O buffer models for this device. By default, only buffer models used in the design are included.
   - **Disable per pin modeling**: Disable inclusion of the per-pin modeling of the package. This is the path from the die pad of the device to the pin of the package. With per-pin modeling disabled, the package is reduced to a single RLC transmission line model applied to all pins and defined in the [Package] section of the IBIS file.

- **Maximum length of signal names**: Truncate signal names to the specified limit:
  - **40**: Truncate signal names to 40 characters, which is supported by IBIS version 4.2 as the default.
  - **20**: Truncate signal names to 20 characters.
  - **Unlimited**: Do not truncate signal names.
- **Updated generic IBIS model file**: Optionally, provide an IBIS model file for the device. This is used to override the IBIS models found in the installation under the parts directory.

IMPORTANT: *The IBIS model file is required for devices that do not have IBIS models included with software installation.*

- **Updated parasitic package data file**: Optionally, provide a parasitic package file (.pkg extension) file to use for per-pin extractions. This is used to override the parasitic package file found in the installation hierarchy under the parts directory.

IMPORTANT: *The parasitic package file is required for devices that do not have IBIS models included with software installation.*

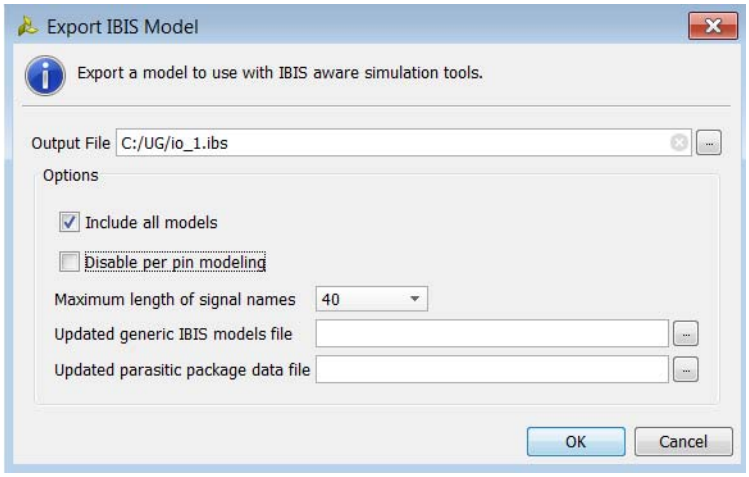

Figure 3-30: **Export IBIS Model Dialog Box**

Running SSN Analysis

The Vivado IDE provides analysis of the switching noise levels associated with the I/O of different devices. SSN analysis provides estimates of the disruption that simultaneously switching outputs can cause on other output ports in the I/O bank. SSN analysis incorporates I/O bank-specific electrical characteristics into the prediction to better model package effects on SSN.

I/Os are grouped into separate isolated I/O banks, each with its own unique power distribution networks, and each with unique responses to switching activity. Because power distribution networks within a packaged FPGA have different responses to noise, it is important to understand the I/O standards and number of I/Os in a design as well as the response of the device power system to this switching.

Xilinx characterizes all banks through three-dimensional extraction and simulation. This information is incorporated into SSN analysis. SSN analysis uses the expected switching profile of a device to predict how the switching affects the power network of the system and in turn, how other outputs in the I/O bank are affected.

IMPORTANT: *SSN analysis is the most accurate method available for predicting how output switching affects interface noise margins. The calculation and results are based on a range of variables. These estimates are intended to identify potential noise-related issues in your design and are not intended as final design sign-off criteria.*

To run SSN analysis:

1. Select **Tools > Report Noise**.

   Note: Alternatively, you can select **Report Noise** from the Flow Navigator.

2. In the Run SSN Analysis dialog box (Figure 3-31), set the following options, and click **OK**.
   - **Results Name**: Enter a name to identify the results in the Noise window.
   - **Export to File**: Export the analysis to an external report file. Enter an output file name, or browse and select a location. Specify the output format for the file as either **CSV** or **HTML**.

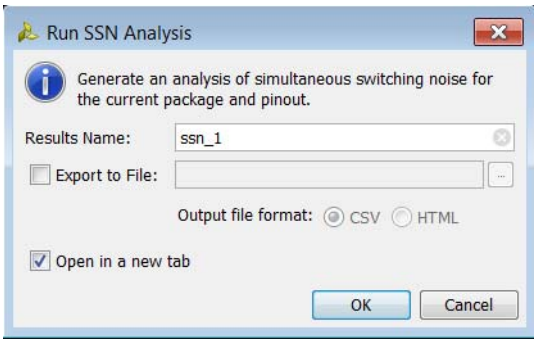

Figure 3-31: **Run SSN Analysis Dialog Box**

Viewing SSN Results

After the analysis is complete, the Noise window (Figure 3-32) opens.

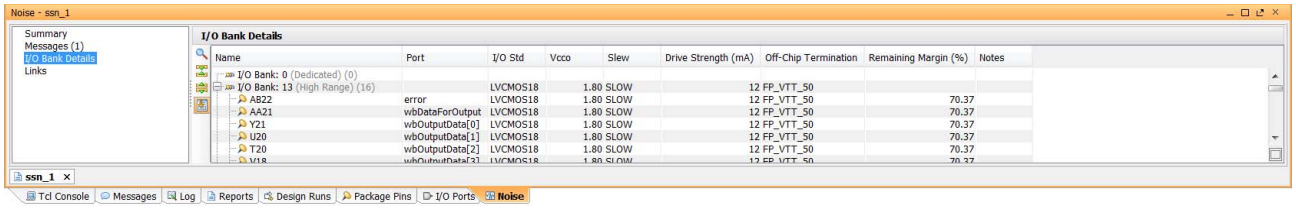

Figure 3-32: **Noise Window**

Click the links on the left side of the window to view different information about the SSN analysis. For example, click **I/O Bank Details** to view the following information:

- **Name**: Displays the I/O banks available in the device. Each I/O bank has pin icons indicating how full the bank is. A check mark indicates a passing result, and a red circle indicates a failure.
- **Port**: Displays the name of the user I/O in the FPGA design.
- **I/O Std, V_{CCO}, Slew, Drive Strength**: Displays the appropriate values for the port or bank.
- **Off-Chip Termination**: Displays the default terminations for each I/O standard, if one exists. Displays either **None** or a short description of the expected or defined off-chip termination style. For example, `FP_VTT_50` describes a far-end parallel 50 Ω termination to VTT termination style. The full list of termination styles is available in the *FPGA SelectIO™ Resources User Guide* for the targeted device.

  For LVTTL (at 2mA, 4mA, 6mA, and 8mA) no termination is assumed. However, for LVTTL (at 12mA, 16mA, and 24mA) a far-end parallel termination of 50 ohms to VTT is assumed. As a result of this termination, the available noise margin is less for signals with drive strength of 12mA, or more, when compared to 2mA to 8mA. Xilinx 7 series FPGA devices use this assumption.

  To change the settings, use either of the following methods:

  - Use the CSV file import feature described in Importing a CSV File.
  - In the I/O Ports table, select an item from the pull-down menu.
- **Remaining Margin %**: Displays the amount of noise margin that is left over after accounting for all SSN in the bank.
- **Notes**: Displays information about the I/O bank or groups.

IMPORTANT: *The SSN results are relative to the state of the design when the SSN analysis is run. It is not a dynamic report.*

Improving SSN Results

To improve SSN results when a violation occurs:

- Use I/O standards that have a lower SSN impact for the failing group. Changing to a lower drive strength, a parallel-terminated DCI I/O standard, or a lower class of driver can improve SSN; for example, changing the SSTL Class II to an SSTL Class I.
- Spread the failing pins across multiple banks. This reduces the number of aggressive outputs on the power system of one bank.
- Spread the failing groups across multiple synchronous phases.

Viewing I/O Bank Properties in SSN Results

You can select an I/O bank in the Noise window to display information about the I/O ports, pins, and groups assigned to the I/O bank in the I/O Bank Properties window. In the I/O Bank Properties window, you can:

- Select the **General** view to view information about the number and types of ports assigned to the I/O bank.
- Select the **Package Pins** or **I/O Ports** view to view the detailed information about the pins or ports within the bank, as shown in Figure 3-33.

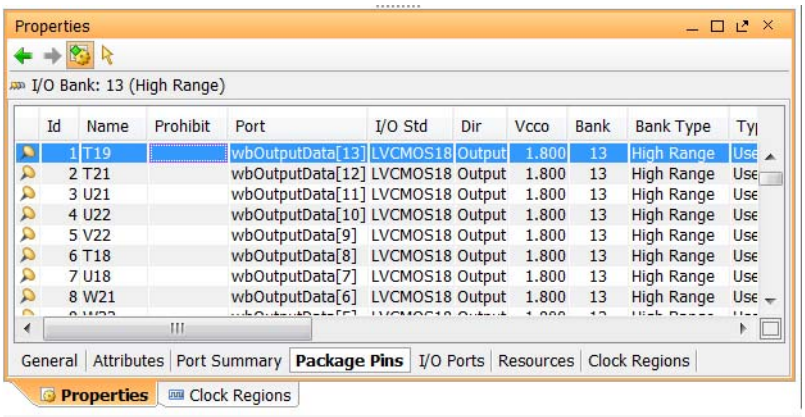

Figure 3-33: **I/O Bank Properties with Package Pins**

Appendix A

I/O Port Lists in CSV File Format

CSV File

A CSV file is a standard file format used by FPGA and board designers to exchange information about device pins and pinout. For more information, see Importing a CSV File and Exporting I/O Pin and Package Data in Chapter 3.

The CSV columns are:

- **I/O Bank**: The I/O Bank in which the pin is located. The tool fills in this field for all pins in the device. Values are a number or blank. This is not required in the input CSV file.
- **Pin Number**: The name (or location) of the package pin. The tool writes this out for all pins in the device. This is not required in the input file. If used for input, it is used to define placement. Values are legal pins in the device.
- **IOB Alias**: An alternate part name for the package pin. This field is specified by the tool and is unused if specified in the input CSV file.
- **Site Type**: The pin name from the device data sheet. This field is specified by the tool and is unused if specified in the input CSV file.
- **Min/Max Trace Delay (ps)**: The distance between the pad site of the die and the ball on the package, in picoseconds. This is specified by the tool to help the board engineer match trace delays. The Trace Delay fields are in the output file only. They are not expected in the input file.
- **Trace Length (um)**: Specifies the length of the internal trace between the package pin and the die pad.
- **Prohibit**: Certain sites can be prohibited for many reasons to prevent user I/O from being added to the site. For example, sites can be prohibited to:
  - Prohibit ease board layout issues.
  - Reduce crosstalk between signals.
  - Ensure that a pinout works between multiple FPGAs in the same package.

  Note: In the XDC file, this is represented by a PROHIBIT property.

- **Interface**: An optional user-specified grouping for an arbitrary set of user I/O. For example, this field provides a means to specify a relationship for the data, address, and enable signals for a memory interface. Values are a text string or blank.
- **Signal Name**: The name of the user I/O in the FPGA design. Values are a string or blank for an unassigned Package Pin.
- **Direction**: The direction of the signal. Values are IN, OUT, INOUT, or blank when a user I/O is not assigned to the site.
- **DiffPair Type**: This value instructs the software about which pin is the N side of a differential pair, and which pin is the P side. This is used for differential signals only. The tool uses this column instead of a naming convention to determine which pin is the N side of the pair, and which pin is the P side. Values are P, N, or blank when a user I/O is not assigned to the site.
- **DiffPair Signal**: Specifies the name of the other pin in the differential pair. Values are the name of the user I/O or blank when unused.
- **I/O Standard**: I/O standard for a specific user I/O. When this field is blank for a user I/O, the tool uses the appropriate device defaults. Values are a legal I/O standard for the user I/O in the device or blank.
- **Drive**: Drive strength of the I/O standard for a specific user I/O. Not all I/O standards accept a drive strength. If this field is blank, the tool uses the default. Values are a number or blank.
- **Slew Rate**: Slew rate of the I/O standard for a specific user I/O. Not all I/O standards accept a slew rate. If this field is blank, the tool uses the default. Values are FAST and SLOW.
- **Pull Type**: Specifies the pull type for the selected port. When using tristate 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 have either a weak pull-up resistor or a weak pull-down resistor.
- **Phase**: Specifies the phase of an I/O relative to the phase of other I/O in the bank in cases of a synchronous phase offset.
- **Board Signal**: The name of the signal coming into the I/O from the board-level design.
- **Board Voltage**: Specifies the voltage level of the signal coming into the I/O from the board-level design.
- **BUFIO2_REGION**: Defines the BUFIO2 clocking region a port is related to.
- **IN_TERM/OUT_TERM**: Defines the optional IN_TERM or OUT_TERM driver impedance attributes. This is most commonly left blank and is not yet supported for production devices. Using this terminal definition overrides the SLEW and DRIVE STRENGTH attributes and is not supported in SSN calculations.
- **OFFCHIP_TERM**: Specifies the external board level termination of the I/O. This is used for SSN calculations. If the field is left blank, the tool uses the expected terminations in

the SSN calculations and shows this expected termination by default in the SSN report and I/O Ports table.

Note: For information on the expected terminations as well as the corresponding shortened names that display in the tool, see the *FPGA SelectIO™ Resources User Guide* for the targeted device.

IMPORTANT: *When reading the CSV file, the Vivado™ tools retain undefined column values, reporting them as user-defined columns in the I/O Ports window.*

Differential Pairs in the CSV File

There are several properties used to define differential pairs in the CSV file:

- Signal Name
- DiffPair Signal
- DiffPair Type
- I/O Standard

The other values in the CSV file are used to validate a diff pair, to ensure they are fully compatible, but they are not used to define the pair. You can define differential pairs in the CSV file in the following ways:

- **Diff Pair**: This is a direct definition of the two signals which make up a differential pair. Two port entries each have DiffPair Signal values linking to the Signal Name of the other and have complementary DiffPair Type values, one N and one P. The tool checks to ensure that the other attributes such as I/O Standard are compatible when forming the diff pair.
- **Single-link Diff Pair**: Two port entries with complementary DiffPair Type values (one N, one P), but only one port has a DiffPair Signal linking to the other Signal Name. The tool creates the differential pair if all other attributes are compatible.
- **Single Port Diff Pair**: A single port entry with a differential I/O Standard, a DiffPair Type value, and a DiffPair Signal that does not otherwise appear in the CSV file. The tool creates the opposite side of the differential pair (the N or P side) with all properties matching those on the original port.
- **Inferred Diff Pair**: Two ports entries with differential pair I/O standards (for example, DIFF_HSTL, DIFF_SSTL) and Signal Names that infer the N and P side. The tool infers a differential pair if all other attributes are compatible.

Appendix B

I/O Port DRC Rule Descriptions

DRCs

The I/O port DRCs available in the Vivado™ tools are not an exhaustive set of I/O-related DRCs. See the device documentation for more information on I/O ports and clock region specifications.

IOB DRCs

Table B-1 lists the input/output block (IOB) DRCs, including abbreviation, intent, and severity.

Table B-1: **IOB DRCs**

Rule Name	Rule Abbreviation	Rule Intent	Severity
Port properties	PORTPROP	Checks for inconsistencies within the properties of a port.	Error
Differential I/O pads	IODI	Checks that differential I/O P and N signals are defined in a dedicated differential pair.	Error
I/O standard type	IOSTDTYPE	Ensures that a diff pair I/O standard is applied to diff pair pins only.	Warning
Number of I/Os	IOCNT	Indicates whether more I/O Ports are defined than there are pins in the target device.	Warning
I/Os placement	IOPL	I/Os placement on disallowed site. Included in the IOPL are: • IOPR, which checks that a port is not placed on a prohibited pin. • IOLVDSCC, which checks that differential output standards are not used on low-capacitance sites that cannot support them.	Error

Table B-1: **IOB DRCs *(Cont'd)***

Rule Name	Rule Abbreviation	Rule Intent	Severity
I/O part compatibility for bank type	IOPCBT	This message occurs in 7 series designs with multiple compatible parts when the use of an I/O standard is legal in the bank it is placed in on one part but not in legal in the corresponding bank of one of the compatible parts. This can occur when the bank on one part is a high performance (HP) I/O bank, but the corresponding bank on a compatible part is a high range (HR) I/O bank. The set of legal I/O standards is not the same for these two bank types. To resolve this issue: • Place the ports on an I/O bank that supports the I/O standard on all the compatible parts. • Change to an I/O standard that is legal in both HP and HR banks. • Change to different compatible parts.	Warning
Prohibit not specified for part compatibility	IOPCPR	For designs that use part compatibility, checks that if any package pin does not exist on at least one compatible part, it is marked as "prohibit" and nothing is placed on it.	Error
MGT not allowed for part compatibility	IOPCMGT	Indicates whether part compatibility is used with two parts that have different serial transceiver supply voltages, thereby disallowing the use of serial transceivers.	Warning
I/O crosstalk to MGT	IOCTMGT	Checks for possible crosstalk problems between I/Os and serial transceivers.	Warning

I/O bus SLR crossings	IOBUSSLRC	The following message occurs when bits of the same bus are placed on different Super Logic Regions:
		`Bus port <BUSPORT[LO:HI]> spans more than one Super Logic Region (SLR).`
		`Bits placed in SLR <SLR1>: 0-3`
		`Bits placed in SLR <SLR2>: 4-7`
		This is not recommended as it makes routing and timing closure more difficult.
		Eliminate the warning by moving all related bus ports into the same SLR.
			Warning

Part compatibility	IOPCSLR	Checks for part compatibility between monolithic and multi-die devices.	Information
IOB clock sharing	IOCS	IOB sites are divided into pairs for the purpose of sharing clock routing resources. These pairs are normally LVDS pairs as well. In some cases there could be routing issues based on how the flops are packed inside the IOB. To resolve this issue flops need to be assigned to specific BEL.	Warning
IOB set reset sharing	IOSR	IOB site has input, output, and tristate registers; all of these registers share same set/reset signal. Cannot pack registers with different reset signals.	Error

Bank I/O Standard DRCs

Table B-2 lists the bank I/O standard DRCs, including abbreviation, intent, and severity.

Table B-2: **Bank I/O Standard DRCs**

Rule Name	Rule Abbreviation	Rule Intent	Severity
Bank I/O standard V_{CC}	BIVC	IOSTANDARD-based V_{OUT} voltage compatibility check for I/Os in that bank.	Error
Bank I/O standard support	BIVB	Checks that the I/O standard is supported in the I/O bank.	Error
Bank I/O standard termination	BIVT	IOSTANDARD-based DCI termination voltage compatibility check for I/Os in that bank.	Error
Bank I/O standard V_{REF}	BIVR	IOSTANDARD-based V_{REF} voltage compatibility check for I/Os in that bank.	Error
Bank I/O standard V_{REF} utilization	BIVRU	Checks for an available V_{REF} site in I/O banks which implement I/O standards requiring a V_{REF}.	Warning
Bank I/O standard	BIIVRC	Checks for a conflict between the I/O standards of a bank and an INTERNAL_VREF constraint on the bank. Standards in a bank cannot require a V_{REF} voltage that differs from that specified by an INTERNAL_VREF constraint for the bank.	Warning
Bank I/O simultaneous switching output limits	BISLIM	Checks the I/O placed within an I/O bank against SSN output.	Error
Bank I/O standard V_{RN}/V_{RP} occupied	DCIP	There are dedicated V_{RP} and V_{RN} I/O sites in I/O banks that can also be used as regular I/Os. If a DCI I/O standard is used in that bank, leave these I/Os unoccupied.	Error
Inconsistent diff pair I/O standards	DIFFISTD	Checks that the terminals of a differential pair have the same I/O standard.	Error
Inconsistent diff pair I/O standards	DIFFISTDDrv	Checks that the terminals of a differential pair have the same drive.	Error
Inconsistent diff pair I/O standards	DIFFISTDSlew	Checks that the terminals of a differential pair have the same slew.	Error
V_{CCAUX} voltage requirement	VCCAUX1	Warns of any requirements on for LVCMOS25.	Warning
V_{CCAUX} voltage requirement	VCCAUX2	Warns of any requirements on LVPECL_33 and TMDS_33.	Error

Additional Resources

Xilinx Resources

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

www.xilinx.com/support

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

www.xilinx.com/company/terms.htm

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

- *Vivado™ Design Suite 2012.3 Documentation* (www.xilinx.com/support/documentation/dt_vivado_vivado2012-3.htm)
- *Large FPGA Methodology Guide* (UG782)
- *Constraints Guide* (UG625)
- *Xilinx Downloads Page* (www.xilinx.com/support/download)