

Vivado Design Suite

Tutorial

I/O and Clock Planning

UG935 (v 2012.3) October 16, 2012





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
10/16/12	2012.3	Initial Xilinx release.

Table of Contents

Overview	4
Software Requirements.....	5
Hardware Requirements.....	5
Tutorial Design Description.....	5
Tutorial Design Files	5
Lab 1: Pre-Synthesis I/O Planning.....	6
Step 1: Creating a Project and Exploring the Windows	6
Step 2: Examining Device I/O Resources	10
Step 3: Prohibiting Pins from I/O Assignment.....	12
Step 4: Creating and Configuring I/O Ports	13
Step 5: Importing an I/O Port List	14
Step 6: Exporting the Device and I/O Pin Assignments.....	15
Step 7: Migrating the I/O Planning Project to an RTL Project.....	16
Lab 2: Post-Synthesis I/O Planning	17
Step 1: Opening the Synthesized Netlist-Based Project.....	17
Step 2: Examining the I/O Ports in the Design	17
Step 3: Configuring I/Os and Setting I/O Standards.....	18
Step 4: Creating I/O Port Interfaces.....	19
Step 5: Viewing Multi-Function Package Pins	21
Step 6: Setting Device Configuration Modes.....	23
Step 7: Defining Alternate Compatible Devices.....	24
Step 8: Placing I/O Ports	25
Step 9: Placing Clock Logic	29
Step 10: Using the Schematic to Trace Clock Logic	30
Step 11: Exploring the Clock Resources Window.....	32
Step 12: Placing the MMCM Instance.....	32
Step 13: Running Design Rule Checks.....	33
Step 14: Running Simultaneous Switching Noise Analysis.....	34
Step 15: Updating the Constraint Files with Interactive Assignments.....	35
Conclusion.....	36

I/O Planning Tutorial

Overview

This tutorial introduces the I/O planning capabilities of the Xilinx® Vivado™ Design Suite for field programmable gate array (FPGA) devices. The objective of this tutorial is to familiarize you with the I/O planning process using the graphical user interface (GUI) known as the Vivado Integrated Design Environment (IDE). In the Vivado IDE, you can begin I/O planning by creating, importing, and configuring the initial list of I/O ports. You can also use various I/O planning capabilities, such as creating and assigning I/O ports to physical package pins or grouping related ports into interfaces for assigning to physical package pins.

You can perform I/O planning at any stage in the design flow. For example, you can begin the I/O assignment process before a register-transfer level (RTL) or synthesized netlist is available using an I/O Planning project. You can also perform I/O planning prior to synthesis by opening the elaborated RTL design. For the most comprehensive set of features and design rule checks (DRCs), you can perform I/O planning after synthesis by opening the synthesized design.

This tutorial includes two labs that you can perform independently:

- **Lab 1:** This lab briefly describes the I/O planning capabilities prior to synthesis before you have a synthesized netlist or RTL sources with I/O ports defined. For example, you can perform I/O exploration and assignment with an I/O Planning project even before the design source files are available. You can import a comma separated value (CSV) format file for I/O planning or export it for use in printed circuit board (PCB) schematic symbol or hardware description language (HDL) header generation.
- **Lab 2:** This lab describes I/O planning capabilities after synthesis. For example, you can use the automatic placement command or the semi-automatic interactive modes to control I/O port assignment. You can also use the I/O Planning layout view to see the relationship of the physical package pins and banks with corresponding I/O die pads. This enables you to make intelligent decisions to optimize the connectivity between the PCB and the FPGA device.

Note: Most of the I/O planning features are described in Lab 2. However, many of the features are available prior to running synthesis as well. Not all commands or command options are covered in this tutorial.

Software Requirements

This tutorial requires that the Vivado Design Suite 2012.3 release or later is installed.

Hardware Requirements

Xilinx recommends a minimum of 2 GB of RAM when using the Vivado tools.

Tutorial Design Description

The sample design used throughout this tutorial consists of a small design called `bft`. The design targets an `xc7k70t` device. A small design is used to allow the tutorial to be run with minimal hardware requirements, to enable timely completion of the tutorial, and to minimize the data size.

Tutorial Design Files

Copy the files from the installation area:

```
<Xilinx_install_area>/Vivado/<version>/examples/Vivado_Tutorial.zip
```

Extract the zip file contents into any write-accessible location. The unzipped `Vivado_Tutorial` data directory is referred to in this tutorial as the `<Extract_Dir>`.

Note: The tutorial sample design data is modified while performing this tutorial. A new copy of the original `Vivado_Tutorial` data is required each time you run the tutorial.

Lab 1: Pre-Synthesis I/O Planning


This lab covers pre-synthesis techniques to begin I/O planning before an RTL or synthesized netlist is available. Most of the features are also available for an elaborated RTL, synthesized, or implemented design. This lab also covers how to use the I/O Planning view layout in the Vivado IDE, which displays windows applicable to placing I/O ports and clock logic.



TIP: You can open the I/O Planning view layout without a design to analyze device resources.

Step 1: Creating a Project and Exploring the Windows

You can create an I/O Planning project as follows:

1. Open the Vivado IDE:
 - **Windows:** Double-click the Vivado 2012.3 shortcut icon , or select **Start > All Programs > Xilinx Design Tools > ISE Design Suite 14.3 > Vivado 2012.3 > Vivado**.
 - **Linux:** Go to the `<Extract_Dir>/Vivado_Tutorial/Tutorial_Created_Data` directory and type: `vivado`
2. In the Getting Started page, select **Create New Project**.
3. In the New Project Wizard, click **Next** to confirm the project creation.
4. In the Project Name page, set the following options, and click **Next**:
 - Type the project name: `project_pinout`
 - Enter the project location:
`<Extract_Dir>/Vivado_Tutorial/Tutorial_Created_Data`

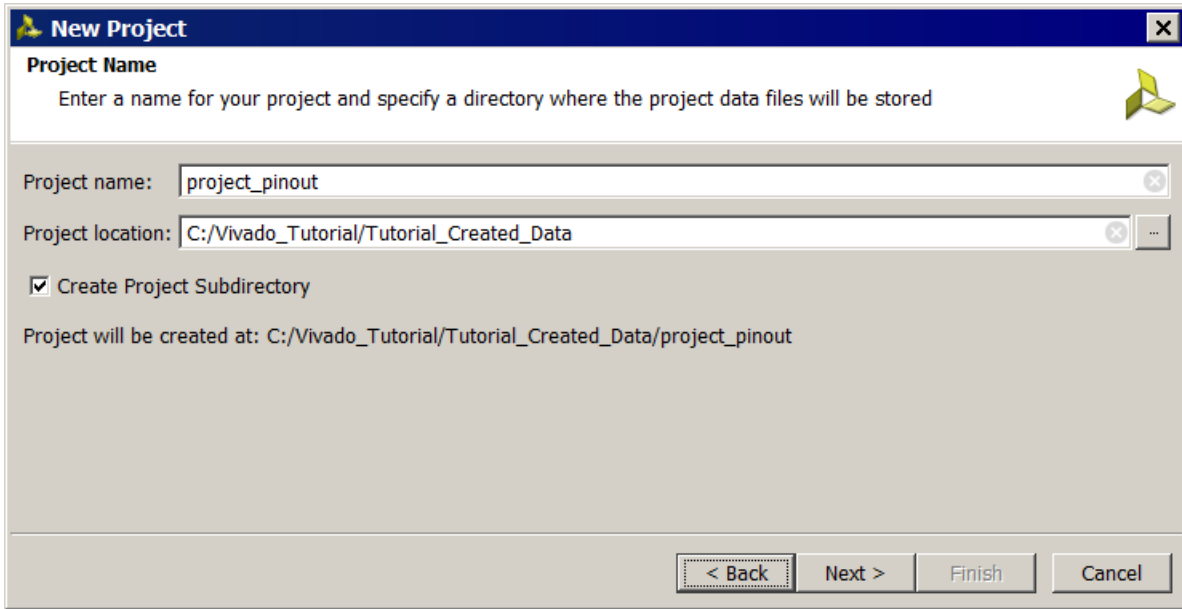


Figure 1: New Project Wizard—Project Name Page

5. In the Project Type page, select **I/O Planning Project**, and click **Next**.

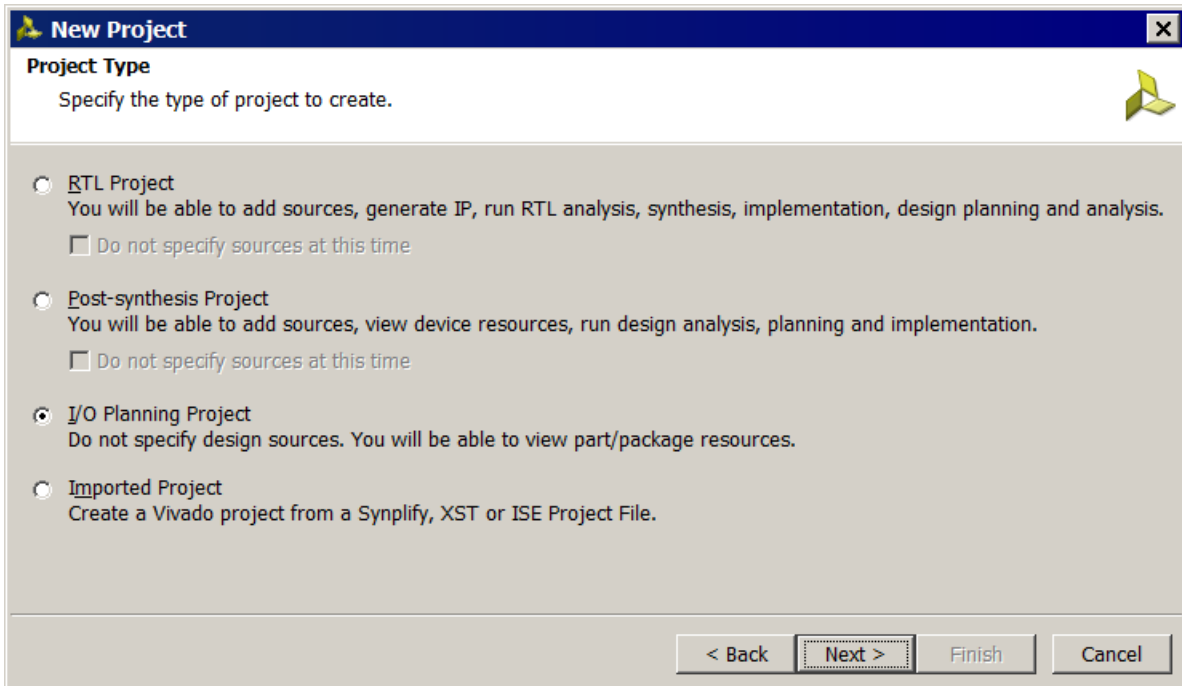


Figure 2: New Project Wizard—Project Type Page

6. In the Import Ports page, select **Do not import I/O ports at this time**, and click **Next**.

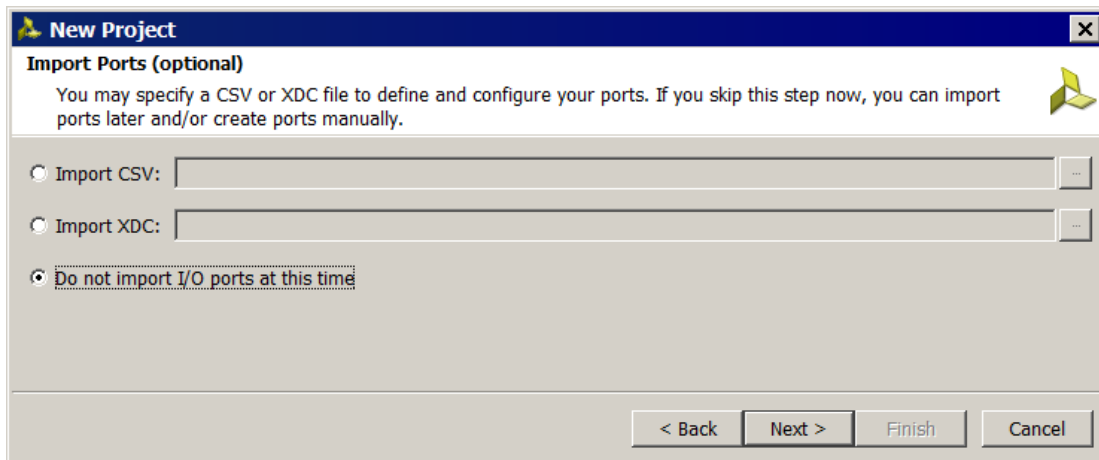


Figure 3: New Project Wizard—Import Ports Page

7. In the Default Part page, set the following options, and click **Next**.

- In the Filter section, click the **Family** pull down menu and select **Kintex™-7**.
- In the Search field, type **70T**.
- Select the **xc7k70tfbg676-2** device.

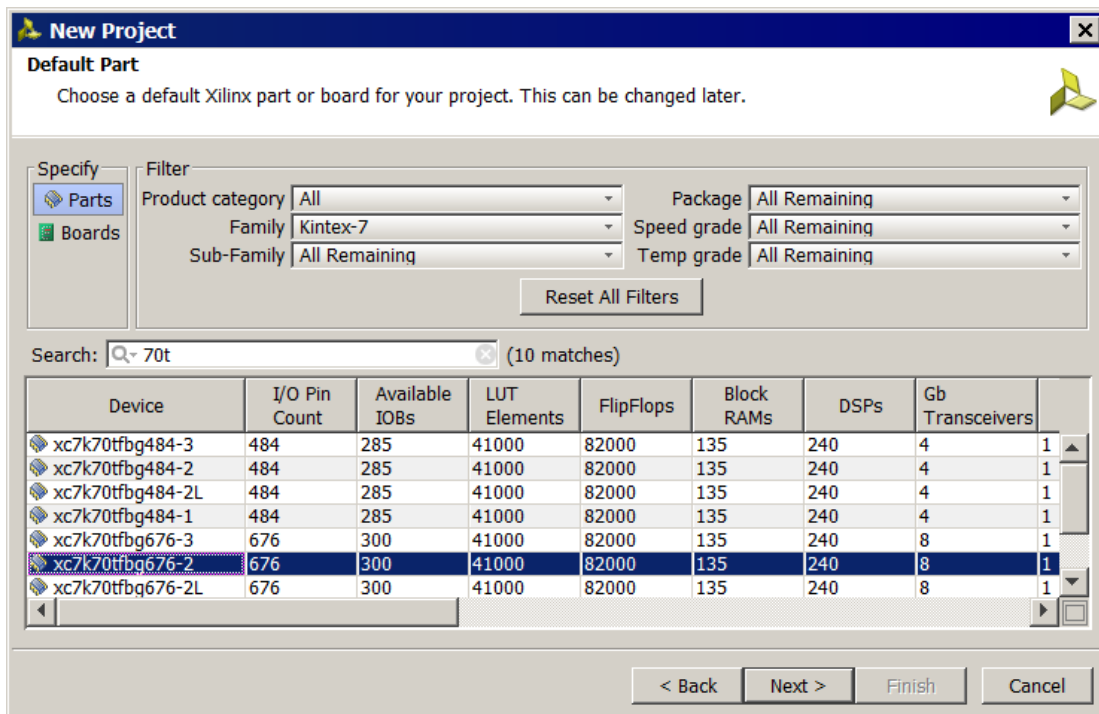


Figure 4: New Project Wizard—Default Part Page

8. Click **Finish** to create the project.

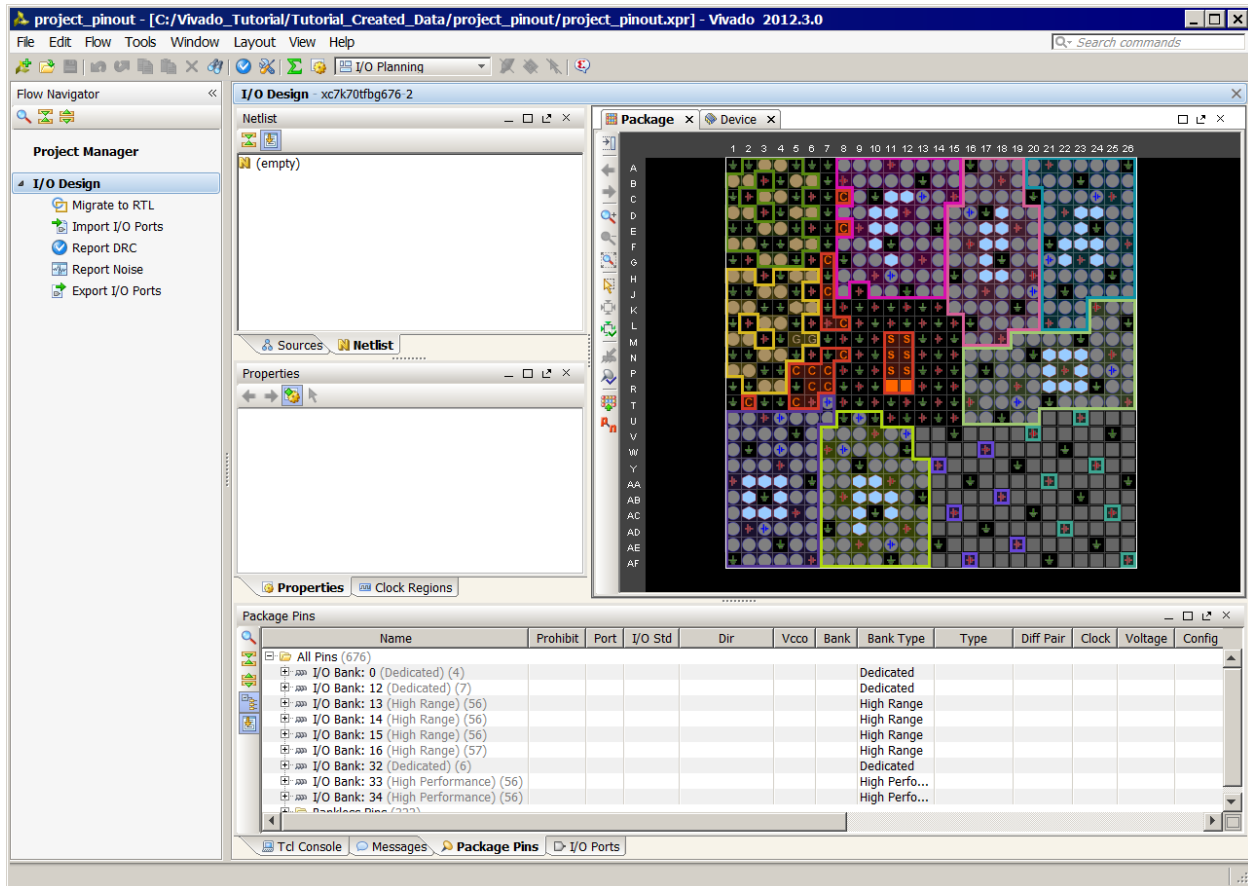


Figure 5: I/O Planning View Layout

9. Explore the various windows in the I/O Planning view layout. Many windows are empty because I/O ports are not yet defined.
10. Right-click on either the Device or Package window tab, and select **New Vertical Group**.
11. Notice that the Device and Package windows are displayed side by side. Being able to visualize the I/O bank locations both internally on the die and externally on the package helps you plan for an optimal I/O port assignment.


Step 2: Examining Device I/O Resources

The I/O Planning view layout lets you explore various device resources. The different windows graphically display and cross-select the location of various I/O, clock, and logic objects to help you make I/O and device-related design decisions. The Package Pins window and I/O Bank Properties window provide some of the I/O related information typically found in the device data sheets.

In the I/O Planning view layout, you can select several I/O banks to show the package-to-die relationship, view I/O bank properties, and select and expand an I/O bank to view package pin specifications as follows:

1. In the Package Pins window, select an I/O bank such as **I/O Bank 33**.
2. Notice that **I/O Bank 33** is selected in the Device and Package windows.
3. To select a bank in the Package window, double-click any pin in the I/O bank.

The first click selects the pin, and the second click selects the I/O bank in which that the pin is located.

Note: Alternatively, you can highlight the I/O bank in the Package window by clicking the **Package View Layers** toolbar button . In the Package View Options, expand the **I/O Banks**, right-click an I/O bank, and select **Select Objects**. You can also use the Package View Options to display specific multi-function pins (such as V_{REF}), adjust the look of the Package window, highlight specific bank types, or hide transceiver banks.

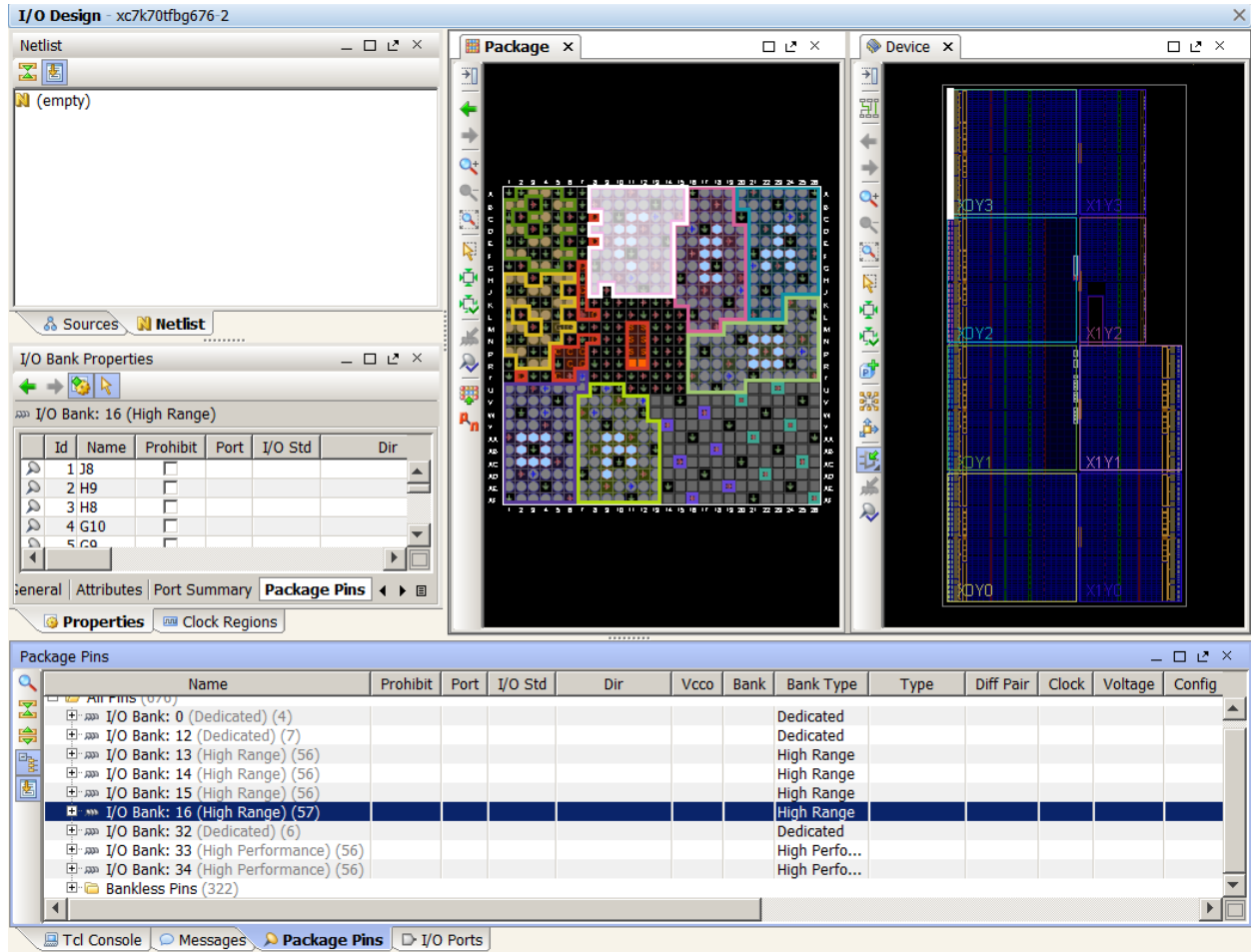





Figure 6: Cross-Selected I/Os and I/O Banks


- In the Package Pins window, expand the selected **I/O Bank** to display the package pin information for each pin in the I/O bank. Scroll to the right to view the internal package trace min and max delays. These are the routing delays between the pin on the package and the pad on the die.
- Scroll down the list and select any **I/O Bank**.
- In the I/O Bank Properties window, select the **General** view.
- Review the I/O count and voltages. This information is populated as I/O ports are assigned to the I/O bank. This allows you to search for compatible I/O banks to place the remaining I/O Ports.
- Select the various tabs in the I/O Bank Properties window.
- In the Package Pins window title bar, click the **Maximize** button .
- In the Package Pins window, click the **Expand All** button .


11. Scroll to the right to view the pin information in the table.
12. In the Package Pins window, deselect the **Group by I/O Bank** toolbar button  to expand and flatten the list.

Step 3: Prohibiting Pins from I/O Assignment

You can prohibit I/O package pins from having I/O ports assigned to them as follows:

1. In the Package Pins window, double-click the **Voltage** column header, and scroll to the top of the list to locate the **VREF** values.
2. Use the **Shift** key to select all **VREF** Voltage pins.
3. Right-click and select **Set Prohibit**.
4. In the Package Pins window title bar, click the Restore button .

The Package window displays prohibited pins, which are indicated by a slashed circle .

5. In the main toolbar, click **Unselect All** .
6. In the Package window, zoom in to an area to view the prohibited pins, as shown in the following figure. To zoom, click and drag the cursor from the upper left to the lower right of the zoom area.

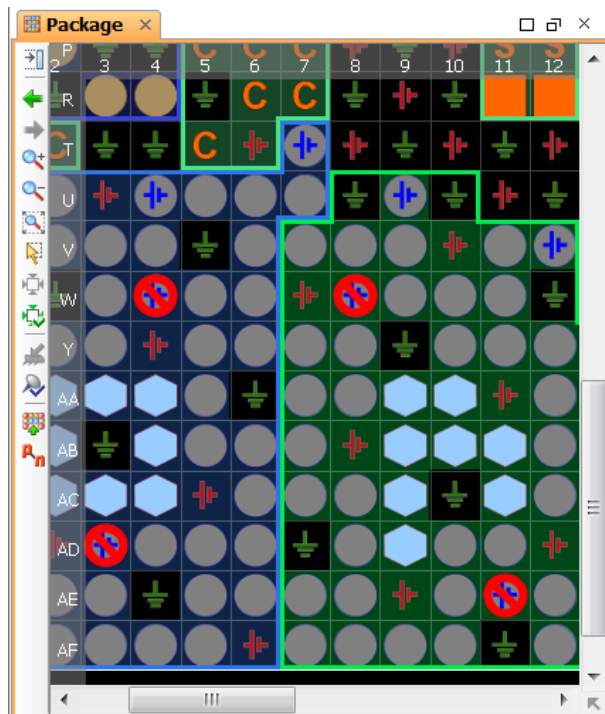


Figure 7: Prohibited V_{REF} Package Pins

7. To zoom fit the Package window, click and drag the cursor from the lower right to the upper left.

Step 4: Creating and Configuring I/O Ports

You can create and configure a new I/O bus port called **mybus** as follows:

1. In the I/O Ports window tab, right-click and select **Create I/O Ports**.
2. In the Create I/O Ports dialog box, set the following options, and click **OK**.
 - In the Name field, type: **mybus**.
 - Enable **Create Bus**.

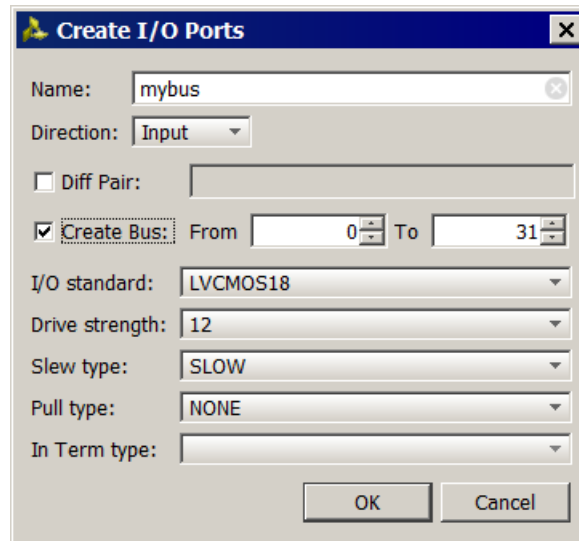


Figure 8: Create I/O Ports Dialog Box



TIP: The **Configure I/O Ports** command opens a similar dialog box that enables you to configure existing I/O Ports.

The new I/O ports display in the I/O Ports window.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip ...	IN_TERM
mybus (32)													
mybus[0]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[1]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[2]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[3]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[4]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[5]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[6]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[7]	Input					LVC MOS18	1.800				NONE	NONE	
mybus[8]	Input					LVC MOS18	1.800				NONE	NONE	

Figure 9: I/O Ports Window with Newly Added I/O Ports

3. Select **Edit > Undo** to remove the recently added **mybus** I/O ports.

Step 5: Importing an I/O Port List

Using the Vivado IDE, you can import several file formats to begin the I/O planning process. You can import CSV, Xilinx Design Constraints (XDC), or RTL format files and perform I/O pin exploration and assignments. This is an alternative to creating I/O ports interactively, which was covered in the previous step.



IMPORTANT: Use care with early input methods of I/O pin planning. Without a synthesized netlist, the I/O ports placement and DRC routines do not consider clocks, clock relationships, or gigabit transceiver (GT) logic in their calculations. When possible, perform I/O pin assignment after importing a synthesized netlist. Legal I/O pinouts are guaranteed only after the design has run through the implementation and after DRCs for I/O and clock placement are run without error.

1. In Windows Explorer, open the following I/O ports CSV file:


```
<Extract_Dir>/Vivado_Tutorial/Sources/IO_Ports_import.csv
```
2. Examine the I/O ports spreadsheet format and content, and exit without saving.
3. In the Vivado IDE Flow Navigator, select **Import I/O Ports**.
4. In the Import I/O Ports dialog box, select **CSV File**, browse to select the following file, and click **OK**:

```
<Extract_Dir>/Vivado_Tutorial/Sources/IO_Ports_import.csv
```

The Device and Package views display the assigned ports, and the I/O Ports window is now populated with the imported I/O ports, as shown in the following figure. The buses are grouped together and are expandable.



CAUTION! If you plan to import a CSV file, do this before defining ports with the Create I/O Ports command, because the port definitions are overwritten when you use this command.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip ...	IN_TERM
All ports (135)													
DataIn_pad_0_i (8)	Input					LVCMOS18	1.800				NONE	NONE	
DataIn_pad_1_i (8)	Input					LVCMOS18	1.800				NONE	NONE	
DataOut_pad_0_o (8)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
DataOut_pad_1_o (8)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
LineState_pad_0_i (2)	Input					LVCMOS18	1.800				NONE	NONE	
LineState_pad_1_i (2)	Input					LVCMOS18	1.800				NONE	NONE	
OpMode_pad_0_o (2)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
OpMode_pad_1_o (2)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
or1200_pm_out (4)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
RXP_IN (16)	Input	RXN_IN				DIFF_HSTL_I1_18					NONE	NONE	NONE
TXP_OUT (16)	Output	TXN_OUT				DIFF_HSTL_I1_18	1.800		SLOW		NONE	NP_VTT_...	
VControl_pad_0_o (4)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
VControl_pad_1_o (4)	Output					LVCMOS18	1.800		12 SLOW		NONE	FP_VTT_50	
VStatus_pad_0_i (8)	Input					LVCMOS18	1.800				NONE	NONE	
VStatus_pad_1_i (8)	Input					LVCMOS18	1.800				NONE	NONE	
Scalar ports (35)													

Figure 10: I/O Bus Ports Grouped by Bus

Step 6: Exporting the Device and I/O Pin Assignments

You can export the I/O port assignments to XDC, CSV, VHDL, or Verilog format files. This is useful for creating HDL headers and PCB schematic symbols. The CSV format output file contains package information for all pins, which you can use to begin I/O port assignments.

1. Select **File > Export > Export I/O Ports**.
2. In the Export I/O Ports dialog box, select **CSV, XDC, Verilog, and VHDL**.

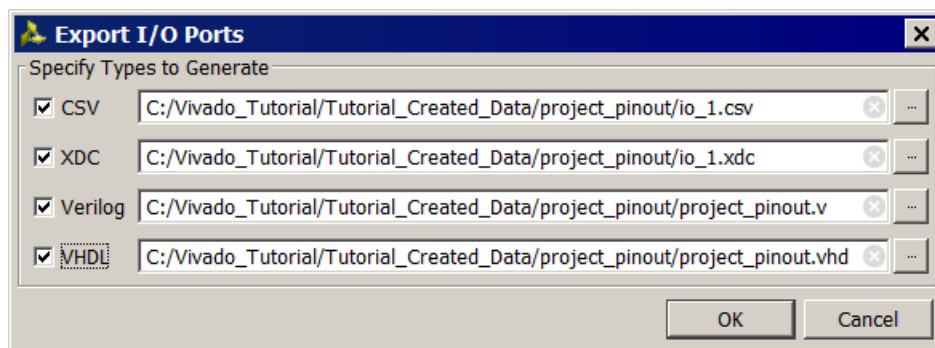


Figure 11: Export I/O Ports Dialog Box

3. Click **OK** to accept the default file names and locations.
4. In an Explorer window, browse to and open the exported files located in:

<Extract_Dir>/Vivado_Tutorial/Tutorial_Created_Data/project_pinout

If defined, the interface group names are included in the CSV spreadsheet. PCB designers can use this spreadsheet to create interface-specific schematic symbols. See Lab 2 for information on creating I/O port interfaces, placing I/O ports, and running DRCs.

Step 7: Migrating the I/O Planning Project to an RTL Project

You can migrate the I/O port assignments made in I/O planning projects to an RTL project. This enables you to add sources and to run the design flow through implementation. The I/O port assignments and names are translated into an RTL header and XDC source files in an RTL project.



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

1. In the Flow Navigator, select **Migrate to RTL**.

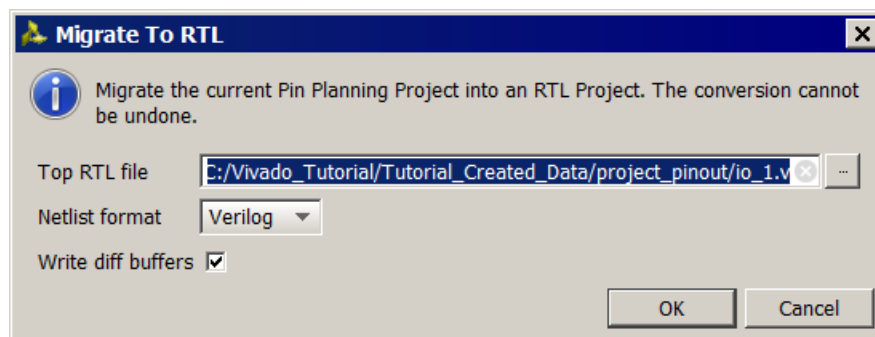


Figure 12: Migrate to RTL Dialog Box

2. In the Migrate to RTL dialog box, click **OK** to accept the default file names and locations.
3. Notice the RTL Project is now displayed. The Sources window contains the newly created source files.
4. Double-click **io_1.v** and **ios_1.xdc** to view the files in the Vivado IDE Text Editor.
5. Select **File > Close Project**.
6. In the Confirm Close Project dialog box, click **OK**.

Lab 2: Post-Synthesis I/O Planning

This lab covers I/O planning with a synthesized design, including the different ways to analyze, group, and place the I/O ports onto package pins or the I/O die pads. For example, it covers how to control I/O port placement using the semi-automatic placement modes. This lab also covers how to run simultaneous switching noise (SSN) analysis and DRCs.

Step 1: Opening the Synthesized Netlist-Based Project

You can open the post-synthesis project as follows:

1. Select **File > Open Project**.
2. In the Open Project dialog box, browse to select the following project file:

```
<Extract_Dir>/Vivado_Tutorial/Projects/project_cpu_netlist/project_cpu_netlist.xpr
```




Note: Alternatively, select **Open Example Project > CPU (Synthesized)** from the Getting Started page. Use **File > Save Project As** to save a local writable copy of the project at `<Extract_Dir>/Vivado_Tutorial/Tutorial_Created_Data`

3. In the Sources window, right-click the **constr_1** folder, and select **Make Active**.
4. Expand the Constraints **constr_1** folder, and double-click the **top.xdc** file.
5. Notice this XDC file only contains timing constraints and no I/Os yet.
6. Close the **top.xdc** file.
7. In the Flow Navigator, select **Open Synthesized Design** to open the synthesized design.
Note: Alternatively, you can select **Flow > Open Synthesized Design** from the main menu.
8. In the Layout Selector located next to the main toolbar, notice that the **I/O Planning** view layout is automatically selected.

The I/O Planning view layout shows windows related to I/O planning, such as the Device and Package windows.

Step 2: Examining the I/O Ports in the Design

You can examine the I/O ports in the I/O Ports window as follows:

1. In the I/O Ports window title bar, click the **Maximize** button .
2. Click the **Expand All**  toolbar button.
3. Click the **Group by Interface and Bus**  toolbar button to disable this feature.
4. Scroll down the list of buses and signals.

The I/O ports now display as a flat list rather than grouped by bus. The Neg Diff Pair fields are populated for some of the buses indicating that they are differential pair buses.

Id	Name	Direction	Interface	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
1	DataIn_pad_0_i[0]	Input						default (LVCMOS18)	1.800				NONE	NONE	
2	DataIn_pad_0_i[1]	Input						default (LVCMOS18)	1.800				NONE	NONE	
3	DataIn_pad_0_i[2]	Input						default (LVCMOS18)	1.800				NONE	NONE	
4	DataIn_pad_0_i[3]	Input						default (LVCMOS18)	1.800				NONE	NONE	
5	DataIn_pad_0_i[4]	Input						default (LVCMOS18)	1.800				NONE	NONE	
6	DataIn_pad_0_i[5]	Input						default (LVCMOS18)	1.800				NONE	NONE	
7	DataIn_pad_0_i[6]	Input						default (LVCMOS18)	1.800				NONE	NONE	
8	DataIn_pad_0_i[7]	Input						default (LVCMOS18)	1.800				NONE	NONE	
9	DataIn_pad_1_i[0]	Input						default (LVCMOS18)	1.800				NONE	NONE	
10	DataIn_pad_1_i[1]	Input						default (LVCMOS18)	1.800				NONE	NONE	
11	DataIn_pad_1_i[2]	Input						default (LVCMOS18)	1.800				NONE	NONE	
12	DataIn_pad_1_i[3]	Input						default (LVCMOS18)	1.800				NONE	NONE	
13	DataIn_pad_1_i[4]	Input						default (LVCMOS18)	1.800				NONE	NONE	
14	DataIn_pad_1_i[5]	Input						default (LVCMOS18)	1.800				NONE	NONE	
15	DataIn_pad_1_i[6]	Input						default (LVCMOS18)	1.800				NONE	NONE	
16	DataIn_pad_1_i[7]	Input						default (LVCMOS18)	1.800				NONE	NONE	
17	DataOut_pad_0_o[0]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
18	DataOut_pad_0_o[1]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
19	DataOut_pad_0_o[2]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
20	DataOut_pad_0_o[3]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
21	DataOut_pad_0_o[4]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
22	DataOut_pad_0_o[5]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
23	DataOut_pad_0_o[6]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
24	DataOut_pad_0_o[7]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
25	DataOut_pad_1_o[0]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
26	DataOut_pad_1_o[1]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
27	DataOut_pad_1_o[2]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
28	DataOut_pad_1_o[3]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	
29	DataOut_pad_1_o[4]	Output						default (LVCMOS18)	1.800		12 SLOW		NONE	FP_VTT_50	

Figure 13: I/O Standard and Diff Pair Requirements

Step 3: Configuring I/Os and Setting I/O Standards

You can use the Vivado IDE to interactively sort and select I/O ports to assign the proper I/O standard, drive strength, slew type, pull type and input termination constraints.



IMPORTANT: For 7 series devices, all I/O ports must have explicit values for the `PACKAGE_PIN` and `IOSTANDARD` constraints to generate a bitstream file. The word `default` is displayed in red to indicate that these values must be applied manually. This is because 7 series devices have low and high voltage I/O banks, and you must apply extra care when assigning I/O standards.

1. In the I/O Ports window, click the **Neg Diff Pair** column header to sort by diff pair port type.
2. Scroll to the top of the list, and select the first port.
Note: At the top of the list, there are *no* Neg Diff Pair ports.
3. Use the **Shift** key, and scroll to select all I/O ports that are *not* Neg Diff Pair ports.

Id	Name	Direction	Interface	Neg Dif...	Site	Fixed	Bank	I/O Std	Vcco
79	VStatus_pad_0_i[1]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
80	VStatus_pad_0_i[2]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
81	VStatus_pad_0_i[3]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
82	VStatus_pad_0_i[4]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
83	VStatus_pad_0_i[5]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
84	VStatus_pad_0_i[6]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
85	VStatus_pad_0_i[7]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
86	VStatus_pad_1_i[0]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
87	VStatus_pad_1_i[1]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
88	VStatus_pad_1_i[2]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
89	VStatus_pad_1_i[3]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
90	VStatus_pad_1_i[4]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
91	VStatus_pad_1_i[5]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
92	VStatus_pad_1_i[6]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
93	VStatus_pad_1_i[7]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800
94	XcvSelect_pad_0_o	Output				<input type="checkbox"/>		default (LVCMOS18)	1.800
95	XcvSelect_pad_1_o	Output				<input type="checkbox"/>		default (LVCMOS18)	1.800
96	RXP_IN[0]	Input		RXN_IN[0]		<input type="checkbox"/>			
97	RXP_IN[1]	Input		RXN_IN[1]		<input type="checkbox"/>			
98	RXP_IN[2]	Input		RXN_IN[2]		<input type="checkbox"/>			
99	RXP_IN[3]	Input		RXN_IN[3]		<input type="checkbox"/>			
100	RXP_IN[4]	Input		RXN_IN[4]		<input type="checkbox"/>			
101	RXP_IN[5]	Input		RXN_IN[5]		<input type="checkbox"/>			
102	RXP_IN[6]	Input		RXN_IN[6]		<input type="checkbox"/>			
103	RXP_IN[7]	Input		RXN_IN[7]		<input type="checkbox"/>			

Figure 14: Selected Single Ended Ports


- Right-click and select **Configure I/O Ports**.
- In the Configure Ports dialog box, ensure that the I/O Standard option is set to the default of **LVCMOS18**, and click **OK**.

Note: The options in this dialog box enable you to set I/O configuration constraints. For this lab, the options are left at their default settings.
- Notice the I/O Std column entries are now set to LVCMOS18. The Neg Diff Pair signals are associated with GT pins, so an I/O standard is not applicable. After these ports are placed on GT pins, these fields will be empty.

Step 4: Creating I/O Port Interfaces

It can be beneficial to group I/O ports associated with various I/O interfaces. The I/O Planning view layout lets you define groups of pins, buses, or other interfaces together as an *interface*. You can use interfaces to help with I/O port management and with generating interface-specific PCB schematic symbols. Using interfaces also forces the various **Place I/O Ports** commands to group the entire interface together on the device where possible.

The design used in this tutorial has two universal serial bus (USB) interfaces, each containing many I/O ports. The I/O port names are differentiated by `_0_` and `_1_`. You can create interfaces for all signals in `USB0` and `USB1` as follows:

1. In the I/O Ports window, click the **Show Search** toolbar button .
2. In the Search field, type `_0_`.

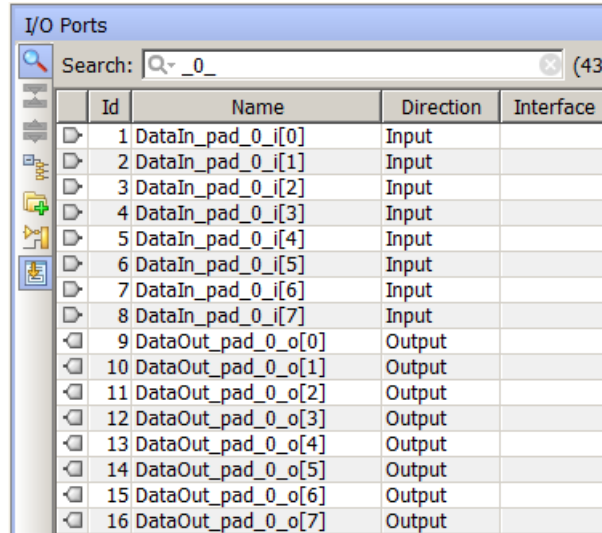


Figure 15: USB_0_ Related Ports

3. Select one of the ports in the filtered list.
4. Press **Ctrl+A** to select all ports in the filtered list.
5. Right-click and select **Create I/O Port Interface**.
6. In the Create I/O Port Interface dialog box, type **USB0** in the Name field, and click **OK**.

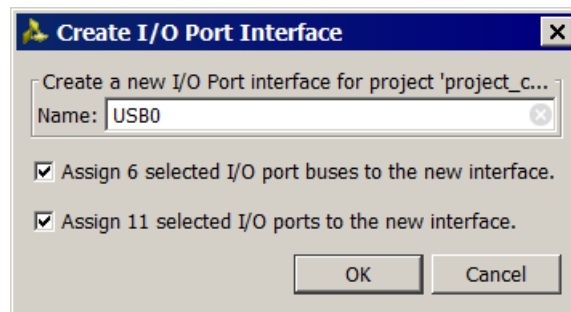






Figure 16: Create I/O Port Interface Dialog Box

7. In the Search field, change `_0_` to `_1_` and follow the same steps to create a **USB1** I/O port interface.

8. Click the **Show Search** toolbar button  to remove the Search filter.
9. Click the **Group by Interface and Bus**  and the **Collapse All**  toolbar buttons.
The I/O ports list is condensed with all of the USB related-ports in interface groups.
10. Expand the **Scalar ports** folder to view the clocks resets and other ports.
11. In the I/O Ports window title bar, click the **Restore** button .

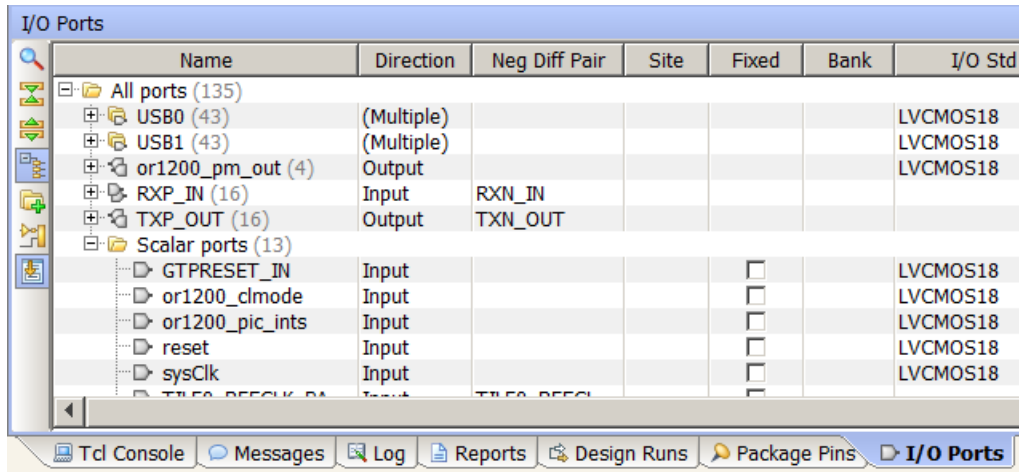





Figure 17: I/O Port Interface Groups and Scalar Ports

Step 5: Viewing Multi-Function Package Pins

Some Xilinx devices have a set of package pins that you can use for multiple purposes depending on the design configuration. These are referred to as *multi-function pins*. For example, you might use multi-function pins for the mode in which you intend to configure the device or to use memory controllers or a peripheral component interface (PCI). You can examine the Package Pins window to ensure that no conflicts exist. View the package pins data and multi-functional pins as follows:

1. In the Package Pins window title bar, click the **Maximize** button .
2. Click **Expand All**  toolbar button.
3. Scroll down and to the right to examine the pins information such as Bank Type, Clock, Voltage, Config, and Site Type. The information displayed in the Package Pins window is dynamically updated as I/O ports are placed in the design.
4. In the Package Pins window, click to unselect **Group by I/O Bank**  toolbar button.

The package pins now display as a flat list rather than in I/O bank groups.




5. Click the **Type** column header to sort based on the Type field.
6. Scroll to view the multi-function pins.

Id	Name	Prohibit	Port	I/O Std	Dir	Vcco	Bank	Bank Type	Type	Diff Pair	Clock	V
183	V25							NONE	GND			
184	W2							NONE	GND			
185	W12							NONE	GND			
186	W22							NONE	GND			
187	Y9							NONE	GND			
188	Y19							NONE	GND			
189	P25	<input type="checkbox"/>					13	High Range	Multi-function	L6N		VREF
190	P23	<input type="checkbox"/>					13	High Range	Multi-function	L11P	SRCC	
191	N23	<input type="checkbox"/>					13	High Range	Multi-function	L11N	SRCC	
192	N21	<input type="checkbox"/>					13	High Range	Multi-function	L12P	MRCC	
193	N22	<input type="checkbox"/>					13	High Range	Multi-function	L12N	MRCC	
194	R21	<input type="checkbox"/>					13	High Range	Multi-function	L13P	MRCC	
195	P21	<input type="checkbox"/>					13	High Range	Multi-function	L13N	MRCC	
196	R22	<input type="checkbox"/>					13	High Range	Multi-function	L14P	SRCC	
197	R23	<input type="checkbox"/>					13	High Range	Multi-function	L14N	SRCC	
198	T19	<input type="checkbox"/>					13	High Range	Multi-function	L19N		VREF
199	B24	<input type="checkbox"/>					14	High Range	Multi-function	L1P		
200	A25	<input type="checkbox"/>					14	High Range	Multi-function	L1N		
201	B22	<input type="checkbox"/>					14	High Range	Multi-function	L2P		
202	A22	<input type="checkbox"/>					14	High Range	Multi-function	L2N		
203	B25	<input type="checkbox"/>					14	High Range	Multi-function	L3P		
204	B26	<input type="checkbox"/>					14	High Range	Multi-function	L3N		
205	A23	<input type="checkbox"/>					14	High Range	Multi-function	L4P		
206	A24	<input type="checkbox"/>					14	High Range	Multi-function	L4N		

Figure 18: Multi-Function Pins

7. Examine the following columns:
 - Config (device configuration pins)
 - XADC
 - Gigabit I/O

These logic objects can impact I/O assignment because many of them rely on multi-function pins and have fixed I/O requirements. If the design used in this tutorial contained these logic objects, this table would be filled out accordingly, allowing you to examine multi-function pins.

8. In the Package Pins window, click the **Group by I/O Bank** toolbar button .
9. Click the **Collapse All**  toolbar button to return the tree table display to the default display structure.
10. In the Package Pins window title bar, click the **Restore** button .

Note: The Vivado IDE has several data table windows with search and filtering capabilities. See the Using Data Table Windows section in the *Vivado Design Suite User Guide: Using the Vivado IDE (UG893)*.

Step 6: Setting Device Configuration Modes

In the Vivado IDE, you can set one or more device configuration options. In addition, some configuration modes might have an impact on multi-function I/O pins. The related pins display this information in the Config column of the Package Pins window.

1. Select **Tools > I/O Planning > Set Configuration Modes**.
2. In the Set Configuration Modes dialog box, select one or two of the modes to view the descriptions, schematics, and related data sheets.

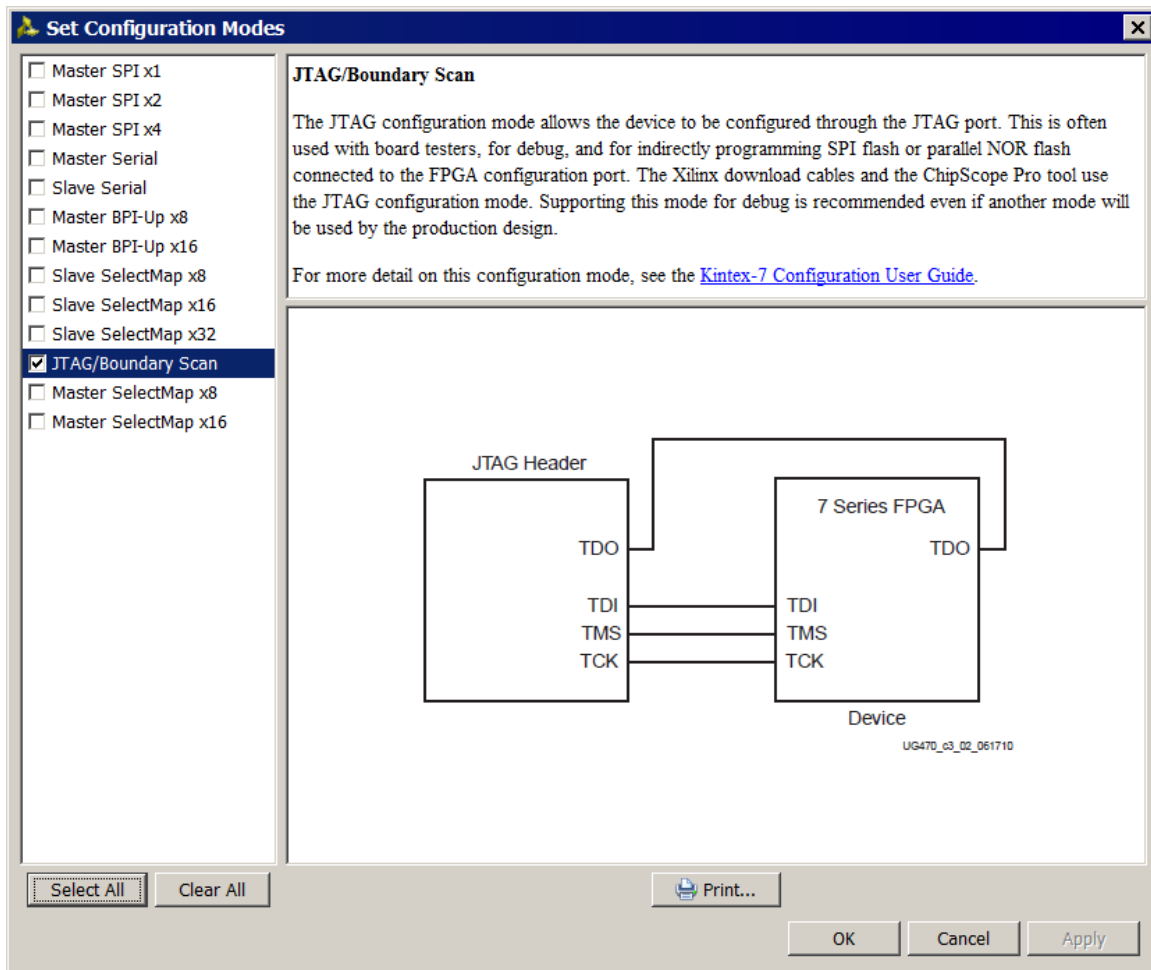


Figure 19: Set Device Configuration Modes Dialog Box

3. Leave the mode set to **JTAG/Boundary Scan** and click **Apply**.
4. Click **OK** to set prohibits on the configuration pins and to sort the Package Pins window by Config pins.

5. After you set the configuration mode, the pins associated with the mode are displayed at the top of the Package Pins window. Examine the pins for potential multi-function pin conflicts.

Step 7: Defining Alternate Compatible Devices

During the FPGA design process, you can change the target device when a design decision calls for a larger or different type. The Vivado IDE lets you define alternate compatible devices up-front so I/O assignments can work across the selected set of devices. This capability is typically limited to devices that use a common package. This ensures that the I/O pinouts work across the selected set of devices.

1. Select **Tools > I/O Planning > Set Part Compatibility**.
2. In the Set Part Compatibility dialog box, select the **xc7k160tfbg676** device, and click **OK**.

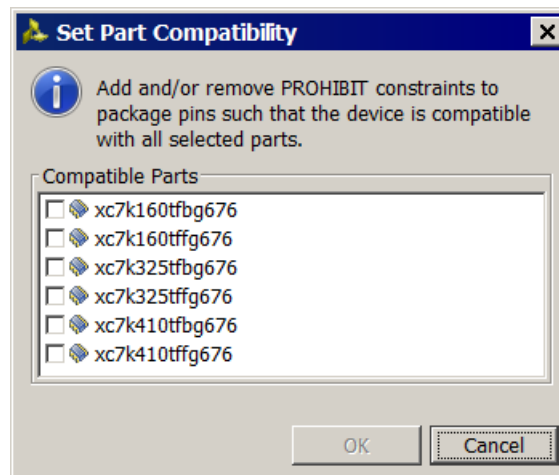


Figure 20: Set Part Compatibility Dialog Box

3. In the confirmation dialog box, click **OK** to indicate that no prohibits were placed.
Prohibits are assigned based on the most restrictive parts. In this example you are targeting the smallest device for this package, so no prohibits are placed.


Step 8: Placing I/O Ports

The Vivado IDE provides several ways to place the I/O ports onto either package pins or I/O die pads. The automatic placement command attempts to place the entire selected group of I/O ports, adhering to I/O bank rules while grouping buses and interfaces together. For more control over I/O port placement, you can drag the selected I/O ports into the Package or Device windows using one of the following semi-automatic placement modes covered in this step:

- **Place I/O Ports in an I/O Bank**
- **Place I/O Ports in an Area**
- **Place I/O Ports Sequentially**

Note: By default, the Vivado IDE uses interactive DRCs during I/O placement.

Placing the USB0 Port Interface Using Place Ports in an I/O Bank

1. In the I/O Ports window, select the **USB0** interface.
2. In the Package window, click the **Place Ports** toolbar button  and select **Place I/O Ports in an I/O Bank**.

As you drag the cursor over the package pins in the Package view, a tooltip shows the number of pins to be placed. The status bar at the bottom of the Vivado IDE also displays information about the objects, including I/O banks and package pins.

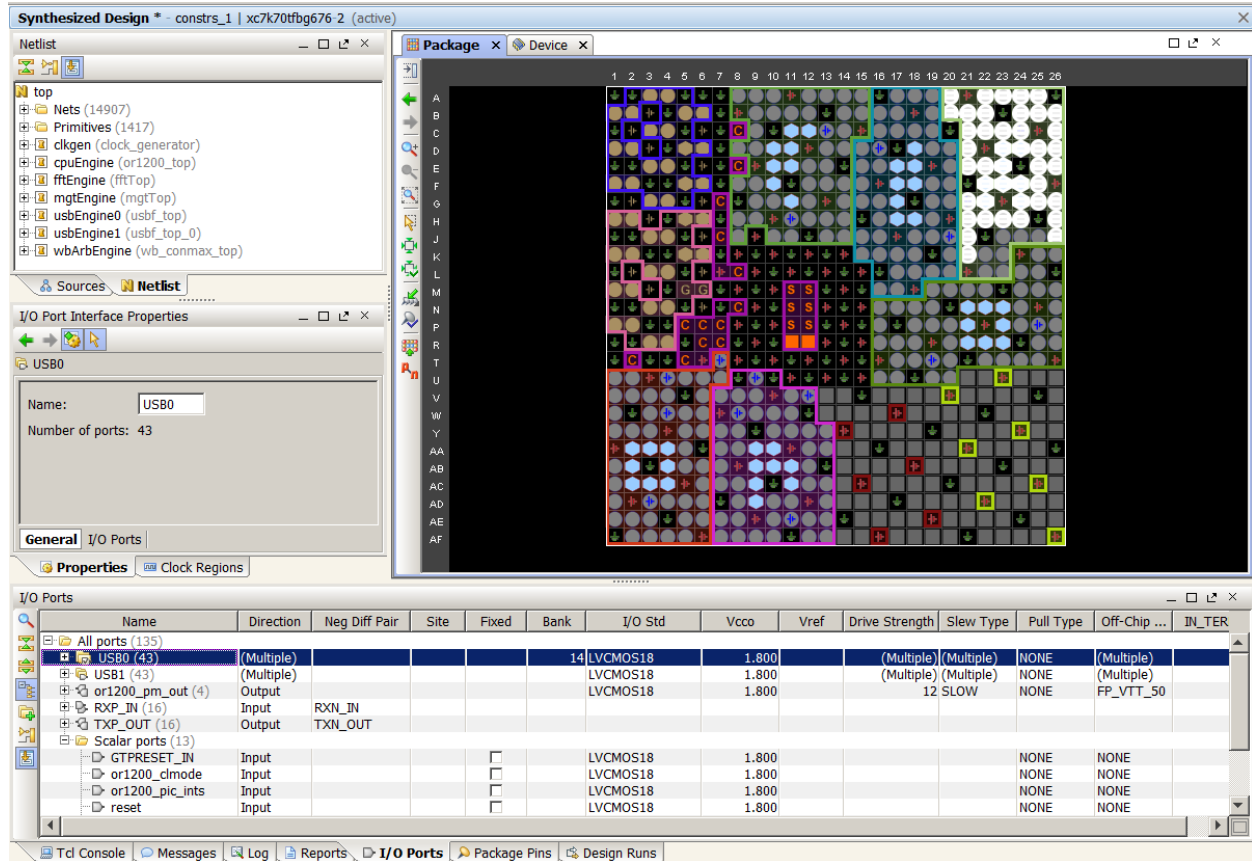



Figure 21: I/O Ports Placed in an I/O Bank

3. On the top right side of the package, click **I/O Bank 14** to drop the I/O ports.

The I/O ports are assigned in the order they appear in the I/O Ports window. Assignment locations are vectored out from the initial pin selected.

Placing the USB1 I/O Port Interface Using Place Ports in Area

1. In the Device window, zoom in to the upper half of the device.
2. In the I/O Ports window, select the **USB1** interface.
3. In the Device window, click the **Place Ports** toolbar button  and select **Place I/O Ports in Area**.

The cursor displays a cross indicating that you can draw a rectangle.

4. Draw a rectangle starting at the bottom of the first I/O bank in the top half of the device, and drag it up and to the right until all I/O ports are placed in the rectangle within the top clock region.

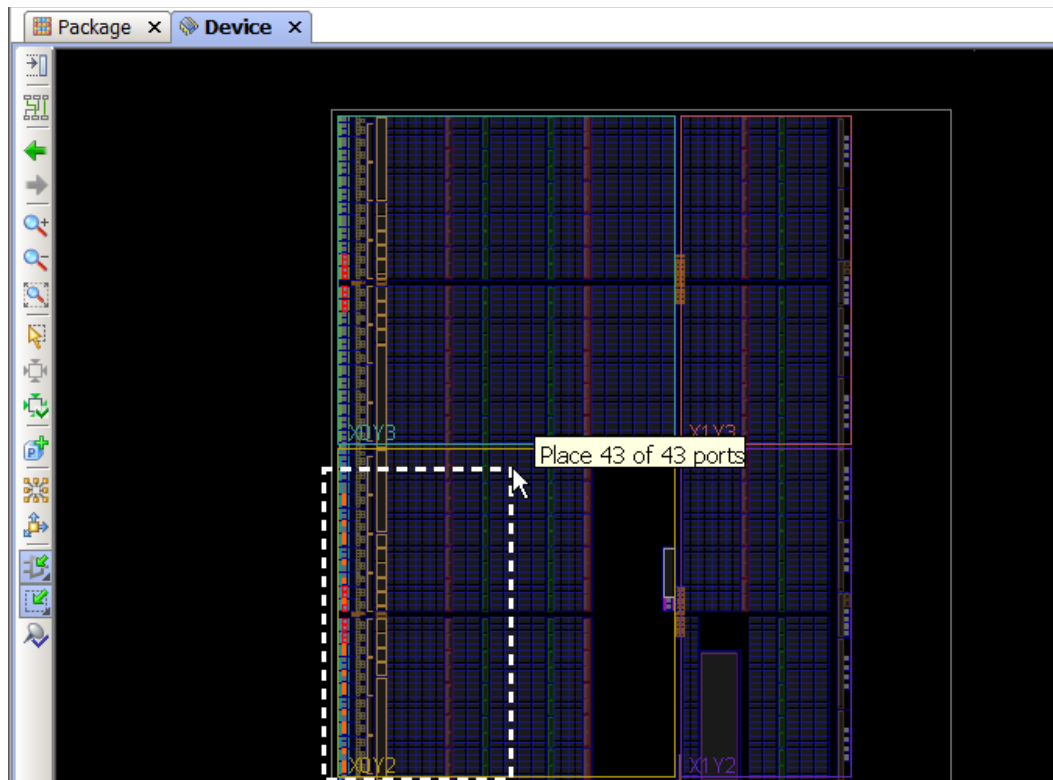



Figure 22: USB1 I/O Ports Placed in an Area

Placing the RXP_IN Differential Pair Bus

1. Zoom in on the upper left corner of the Package window to show the GT pins.
2. In the I/O Ports window, select the **RXP_IN** bus.
3. In the Package window, click the **Place Ports** toolbar button  and select **Place I/O Ports Sequentially**.
4. Drag and click to place the first diff pair I/O port into one of the GT I/O banks on a designated pin. The GT banks are the two banks located in the upper left.

Both diff pairs associated with the GTs are placed on legal sites. If you try to select a site that is not legal, a tooltip indicates why it is not a legal site.



TIP: You can manually enter a pin location in the Site field in the I/O Port Properties window.

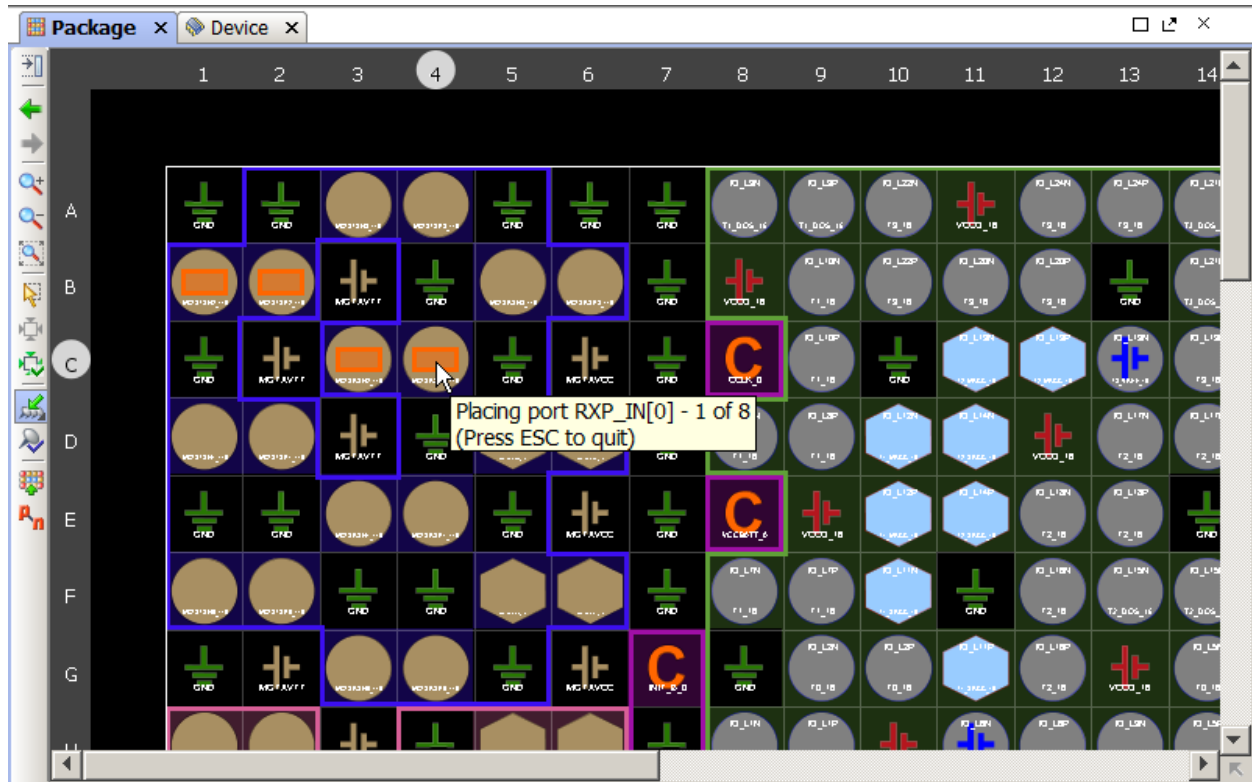



Figure 23: Diff Pair I/O Bus Ports Placed Sequentially

5. Place all **8 RXP_IN** GT port groups in the two upper left I/O banks.
 Tooltips show legal pin selections to help you place the I/O banks.
Note: GT logic objects are automatically grouped by the Vivado IDE to ensure proper behavior when I/O ports are placed or moved. Both sets of diff pair I/O ports as well as the GT itself are all placed and moved as a group.
6. In the I/O Ports view, expand the **Scalar ports** folder, and use the **Shift** key to select the 4 GT reference clock ports called **TILE0-3**.
7. In the Package window, click the Place Ports toolbar button , and select **Place I/O Ports Sequentially**.
8. Drag and click to place the first GT reference clock I/O port into one of the hexagon shaped pins within the GT I/O banks. The GT banks are the two banks located in the upper left.
9. Place all four GT reference clock ports.

Automatically Placing the Remaining Ports

1. Select **Tools > I/O Planning > Auto-Place I/O Ports**.
2. In the Auto-Place Ports wizard, click **Next**.

3. In the Placed I/O Ports page, click **Next** to keep the current locations as assigned.
4. In the Summary page, click **Finish**.
5. Click **OK** to confirm that all I/O ports were placed.

Step 9: Placing Clock Logic

The Vivado IDE lets you place critical clock or I/O-related logic. After a synthesized netlist is imported, you can explore clocks and clock relationships and use this information to lock down these logic objects onto specific device sites. The Vivado IDE automatically groups some logic, such as GTs and their associated I/O pin pairs. This makes selection and placement of GTs and other related logic less prone to errors. To search for global clock logic in the design:

1. Select **Edit > Find**.

Note: Alternatively, you can click the **Find** toolbar button .

2. In the Find dialog box, adjust the selection filters to match the following figure, and click **OK**.

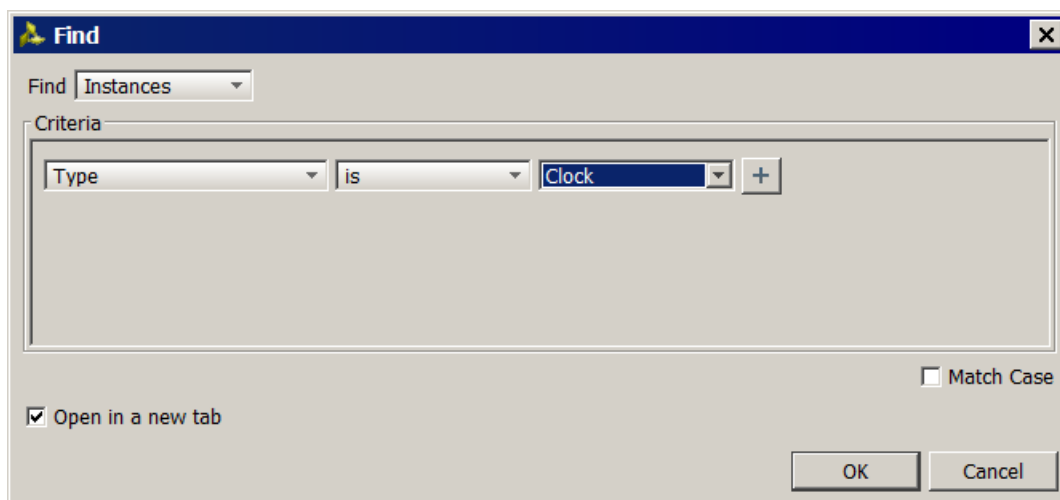


Figure 24: Find Dialog Box to Search for Clock Logic

3. In the Find Results window, scroll down the list of objects, and observe the following:

- BUFG
- IBUFG
- MMCME2_ADV

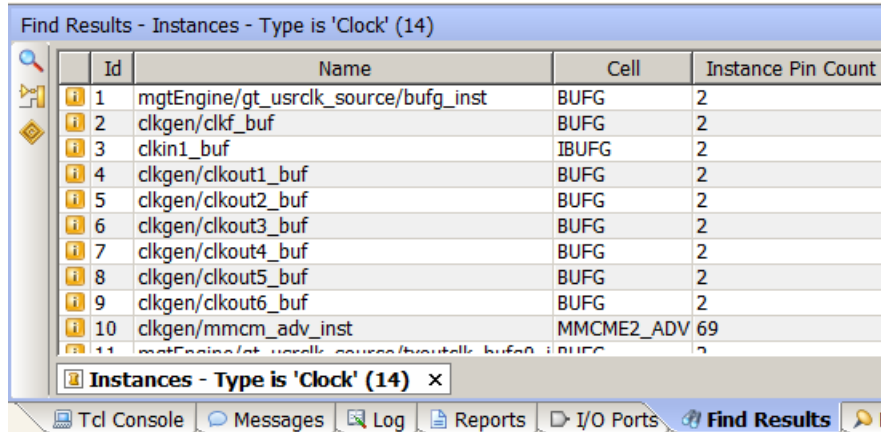


Figure 25: Clock Objects in the Find Results Window

Step 10: Using the Schematic to Trace Clock Logic

You can use the Schematic window to expand and explore any logic in the design. You can also apply placement constraints from the Schematic window.

1. In the Find Results window, select the first **MMCME2_ADV** cell.

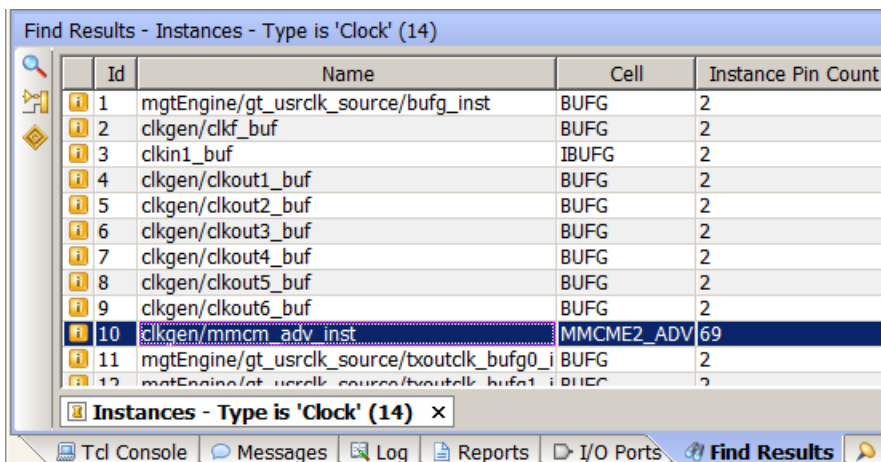




Figure 26: Clock Logic to Trace in the Schematic

2. In the Find Results window, click the **Schematic** toolbar button .

3. In the Schematic window, double-click the **CLKIN1** input port on the upper left of the mixed mode clock manager (MMCM) module.
4. Double-click the **CLKFBIN** input pin on the MMCM module.
5. Double-click the five MMCM output pins, **CLKOUT1-5**, to expand the BUFs.
6. In the Schematic window, click the **Regenerate Schematic** toolbar button  to clean up the connections.

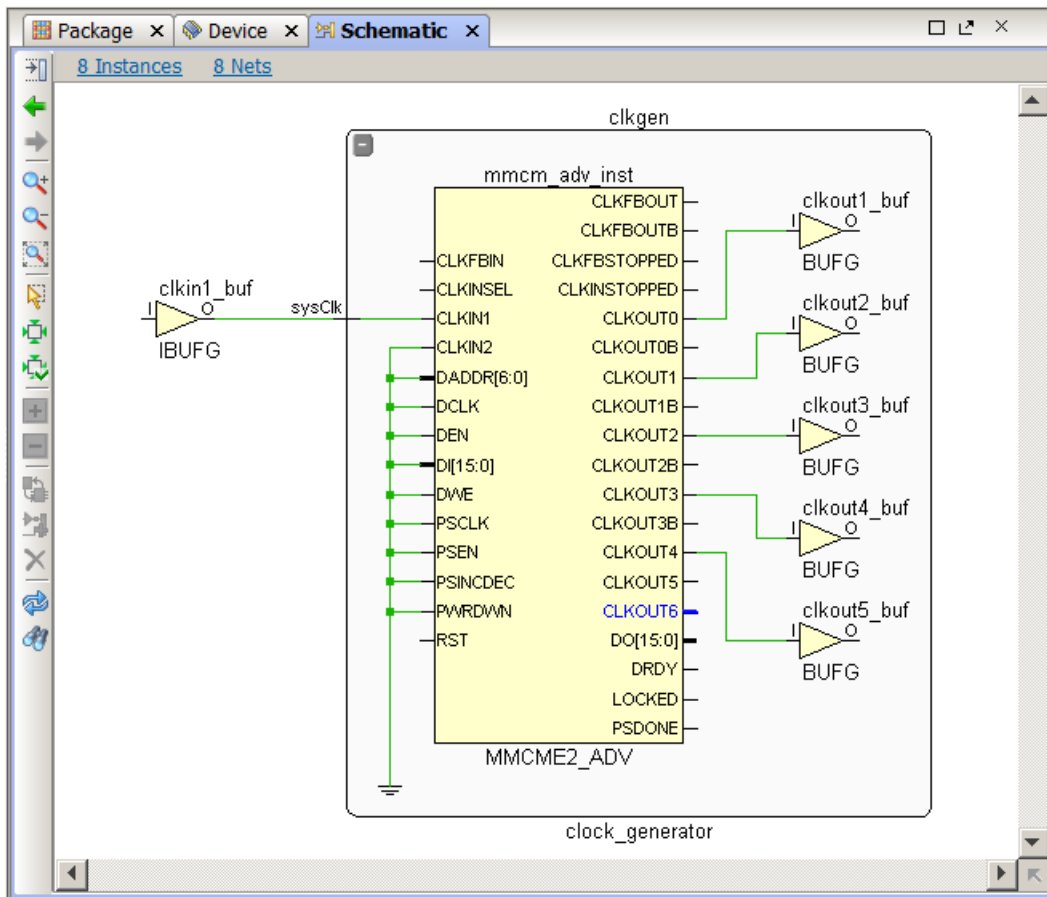


Figure 27: Clock Logic Connectivity

In the Schematic window, you can:

- Expand and explore logic.
- Select or highlight logic in the Schematic window to cross-select or highlight it in all other windows.
- Drag logic from the Schematic window to the Device, Package, or Clock Resources windows to place the logic.

7. Close the Schematic window.

Step 11: Exploring the Clock Resources Window

The Clock Resources window shows the available clock resources to aid you in planning and placing elements of global and regional clock trees.

1. To display the Clock Resources window, select **Window > Clock Resources**.
2. In the Clock Resources window title bar, click the **Maximize** button .
3. In the Clock Resources window, notice that the clock regions, I/O banks, and various device resources display in their relative location as found on the device.
4. Expand or collapse sections in the Clock Resources window to hide or display the resources.

Placed logic is displayed under the Instance columns. Placed clock resources, such as the GTXE2s and their associated Diff Pair ports, are displayed in the chart along with BUFs and MMCMs.

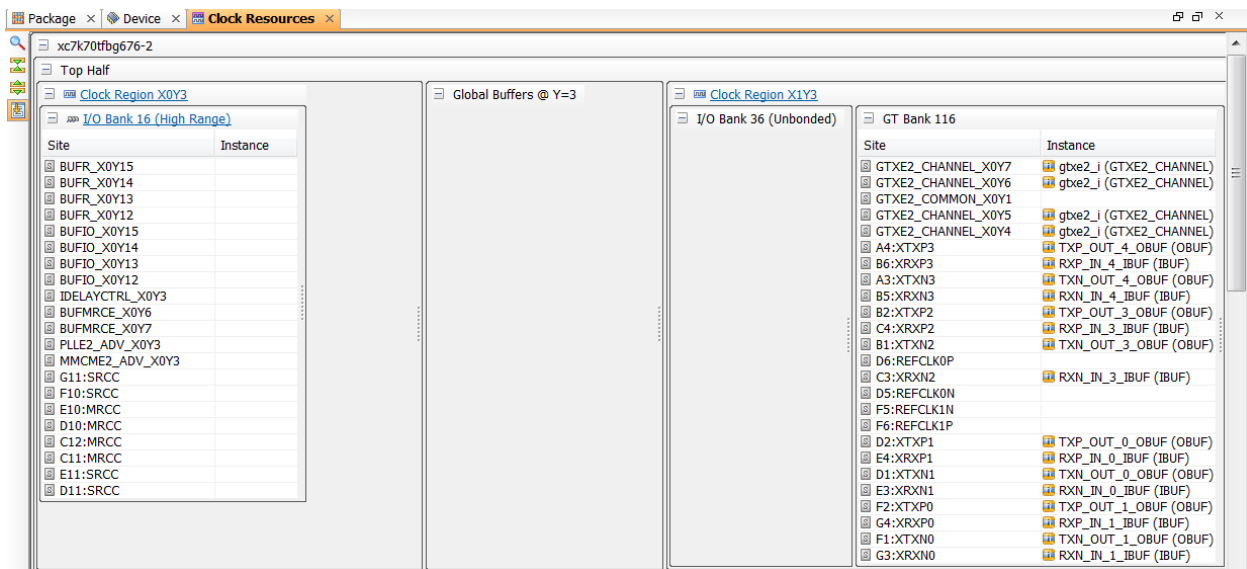


Figure 28: Clock Resources in the Clock Resources Window

Step 12: Placing the MMCM Instance

From the Clock Resources window, you can place the MMCM instance as follows:

1. In the Find Results window, select the MMCME2_ADV instance and drag it into the Clock Resources window in the Instance column next to one of the MMCME2_ADV Sites.
2. Notice how you can place clock and related I/O logic in the Clock Resources window.

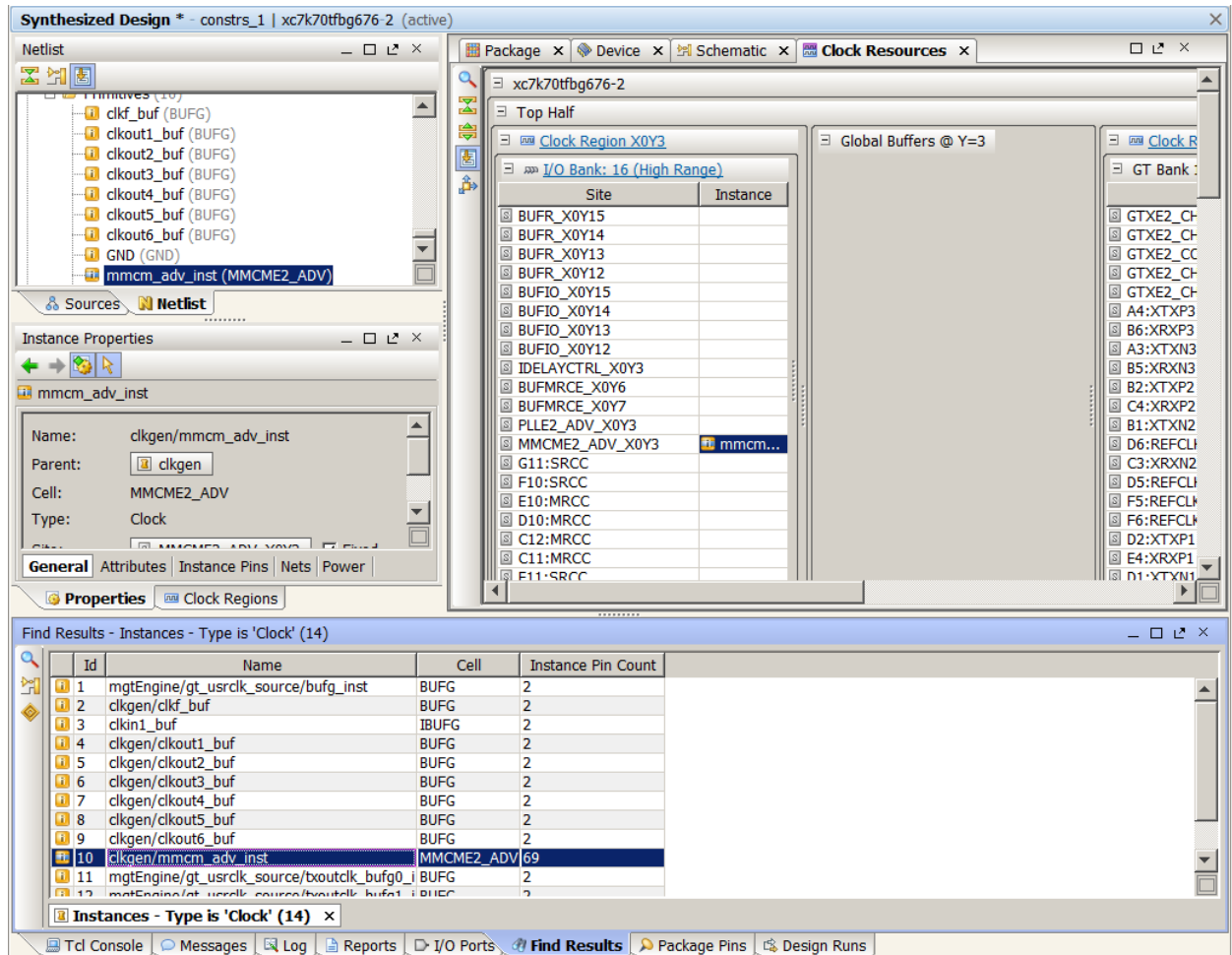



Figure 29: Placed MMCM_ADV Logic Instance

3. Close the Find Results window.
4. In the Clock Resources window title bar, click the **Restore** button .
5. Click the Device window tab.

Step 13: Running Design Rule Checks

The Vivado IDE has an extensive set of I/O-related DRCs to ensure that I/O ports are assigned correctly. You can explore and resolve any violations interactively.

1. In the Flow Navigator under the Synthesized Design section, click **Report DRC**.
2. In the Report DRC dialog box, deselect the **Placer**, **Netlist**, **Floorplan**, **DSP48**, **RAMB**, and **Implementation** rule categories.

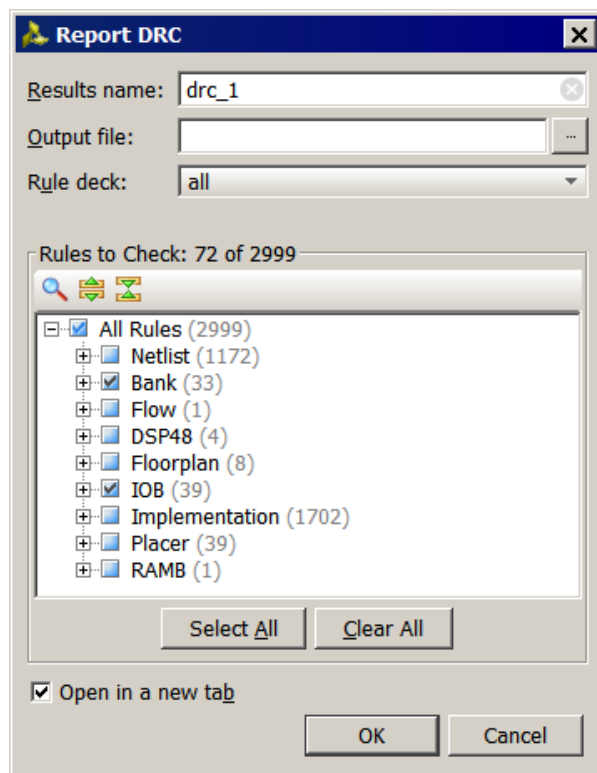


Figure 30: Report DRC Dialog Box

3. Expand the selected rules to examine the rule types, and click **OK**.
4. Notice that no violations were found.
5. In the No Violations Found dialog box, click **OK**.

Step 14: Running Simultaneous Switching Noise Analysis

You can perform SSN analysis to help identify potential signal integrity issues as follows:

1. In the Flow Navigator under the Synthesized Design section, click **Report Noise**.
2. In the Run SSN Analysis dialog box, leave the options at the default values, and click **OK**.

The Noise window opens.

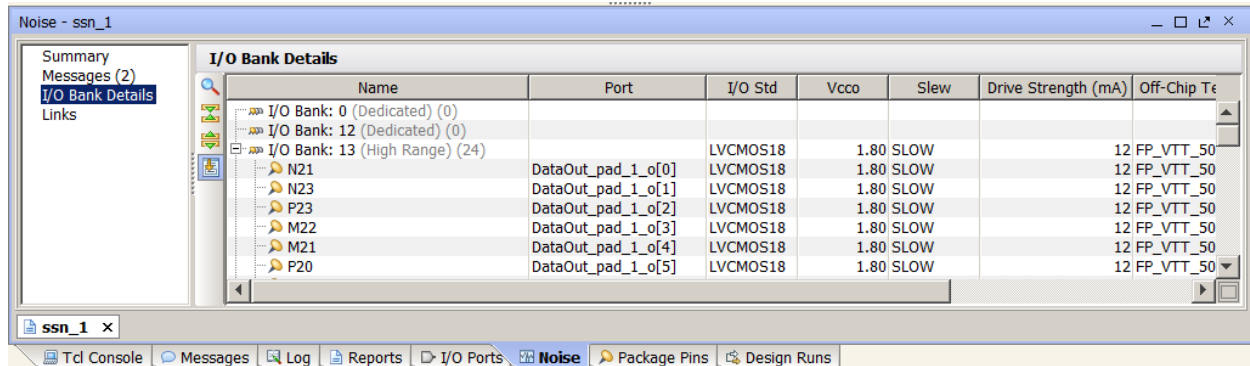




Figure 31: Noise Results Window

3. In the Noise window title bar, click the **Maximize** button .
4. Scroll down, and expand the list of I/Os.
5. In the left pane, click **Summary**, and examine the information.
6. Click **Messages** and **Links** to examine the information.
7. In the Noise window title bar, click the Restore button .



TIP: You can also perform simultaneous switching noise (SSN) analysis to help identify potential signal integrity concerns.

Step 15: Updating the Constraint Files with Interactive Assignments

The Vivado IDE is an interactive constraint assignment environment. During this tutorial, you made numerous modifications to the physical constraints in the design. These changes are currently stored in memory. You can save changes back to the project constraint files as follows:

1. Select **File > Save Constraints**.
2. In the Sources window, double-click the `top.xdc` file under the constraint folder **constr_1**.
3. Notice the new physical constraints applied.
4. Close the `top.xdc` file.
5. Select **File > Exit**.
6. Click **OK** to close the Vivado IDE.

Conclusion

In this tutorial, you accomplished the following:

- Used the I/O Planning view layout to explore device resources and define alternate compatible devices for the design.
- Imported, created, and configured I/O ports.
- Created interfaces by grouping the related I/O ports.
- Used the semi-automatic placement modes to assign critical I/O ports to package pins.
- Used automatic placement to assign the remaining I/O ports.
- Exported and examined the I/O ports list, which can be used for HDL header or PCB schematic symbol generation.
- Opened a netlist-based project and placed GTXE, MMCM_ADV, and BUFG objects using logic connectivity as a guide for correct placement.
- Ran DRCs and SSN analysis to validate legal I/O placement.
- Updated the constraint files with the interactive assignments.