

Vivado Design Suite

Tutorial

I/O and Clock Planning

UG935 (v2014.2) June 4, 2014





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 Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; 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 such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

©Copyright 2012-2014 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, Vivado, ISE, Kintex, Spartan, UltraScale, 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
06/04/2014	2014.2	Validated with release.
04/02/2014	2014.1	Validated with release.

Table of Contents

Revision History.....	2
Table of Contents.....	3
I/O Planning Tutorial.....	5
Overview	5
Tutorial Design Description	6
Hardware and Software Requirements	6
Preparing the Tutorial Design Files.....	7
Lab 1: Pre-Synthesis I/O Planning.....	8
Step 1: Creating a Project.....	8
Step 2: Examining Device I/O Resources.....	13
Step 3: Prohibiting Pins from I/O Assignment	14
Step 4: Creating and Configuring I/O Ports	15
Step 5: Importing an I/O Port List	17
Step 6: Exporting the Device and I/O Pin Assignments.....	18
Step 7: Migrating the I/O Planning Project to an RTL Project.....	18
Conclusion.....	20
Lab 2: Post-Synthesis I/O Planning	21
Step 1: Opening the Synthesized Netlist.....	21
Opening the Vivado IDE:.....	21
Opening the Project.....	21
Step 2: Configuring I/Os and Setting I/O Standards.....	23
Step 3: Creating I/O Port Interfaces.....	24
Step 4: Viewing Multi-Function Package Pins	26
Step 5: Setting Device Configuration Modes.....	27
Step 6: Defining Alternate Compatible Parts	28

Step 7: Placing I/O Ports 29

- Placing I/O Ports in an I/O Bank..... 29
- Placing I/O Ports in an Area..... 31
- Placing I/O Ports Sequentially..... 32
- Placing I/O Ports Automatically..... 33

Step 8: Placing Clock Logic 34

Step 9: Using the Schematic to Trace Clock Logic..... 36

Step 10: Exploring the Clock Resources Window 38

Step 11: Placing the MMCM Instance..... 39

Step 12: Running Design Rule Checks..... 40

Step 13: Running Simultaneous Switching Noise Analysis..... 41

Step 14: Updating the Constraint Files with Interactive Assignments..... 42

Conclusion..... 42



I/O Planning Tutorial



IMPORTANT: This tutorial requires the use of the Kintex[®]-7 family of devices. You will need to update your Vivado tools installation if you do not have this device family installed. Refer to the Vivado Design Suite User Guide: Release Notes, Installation, and Licensing ([UG973](#)) for more information on Adding Design Tools or Devices.

Overview

This tutorial introduces the I/O planning capabilities of the Xilinx[®] Vivado[®] Design Suite for 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 configuring I/O ports, grouping related ports into interfaces, and assigning ports to physical package pins.



VIDEO: You can also learn more about the I/O planning capabilities of the Vivado Design Suite by viewing the quick take video at <http://www.xilinx.com/training/vivado/io-planning-overview.htm>.

You can perform I/O planning at any stage in the design flow. For example, you can begin the I/O assignment process from the top-level of a register-transfer level (RTL) design, or even before an RTL design or synthesized netlist are available. For the most comprehensive set of features and design rule checks (DRCs), you should 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) file for I/O planning or export one for use in printed circuit board (PCB) design, schematic symbol generation, or hardware description language (HDL) module definition.
- **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 informed 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.

Tutorial Design Description

The sample design used throughout this tutorial consists of a small design targeting 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.

Hardware and Software Requirements

This tutorial requires that the 2014.2 Vivado Design Suite software release or later is installed. The following partial list describes the operating systems that the Vivado Design Suite supports on x86 and x86-64 processor architectures:

Microsoft Windows Support:

- Windows 8.1 Professional (32-bit and 64-bit), English/Japanese
- Windows 7 and 7 SP1 Professional (32-bit and 64-bit), English/Japanese

Linux Support:

- Red Hat Enterprise Workstation 6.4 and 6.5 (32-bit and 64-bit)
- SUSE Linux Enterprise 11 (32-bit and 64-bit)
- Cent OS 6.4 and 6.5 (64-bit)

Refer to the *Vivado Design Suite User Guide: Release Notes, Installation, and Licensing* ([UG973](#)) for a complete list and description of the system and software requirements.

Preparing the Tutorial Design Files

You can find the files for this tutorial in the examples directory of the Vivado Design Suite software installation, at the following location:

- **<Vivado_install_area>/Vivado/<version>/examples/Vivado_Tutorial.zip**

1. **Extract** the **zip file** contents from the software installation into any write-accessible location.

The location of the extracted `Vivado_Tutorial` directory is referred to as the `<Extract_Dir>` in this Tutorial.

You can also extract the provided zip file, at any time to restore the files to their starting condition.

Note: You will modify the tutorial design data while working through this tutorial. You should use a new copy of the original `Vivado_Tutorial` directory each time you start this tutorial.

Lab 1: Pre-Synthesis I/O Planning

This lab covers techniques for pre-synthesis I/O planning before an RTL or synthesized netlist is available. Most of the features discussed here are also available for use in an elaborated RTL, synthesized, or implemented design.



TIP: You can open a new I/O Planning project, without a design source, to analyze the available device resources on the target part.

This lab also discusses features of the I/O Planning view layout in the Vivado IDE, which displays windows applicable to placing I/O ports and clock logic.

Step 1: Creating a Project

Open the Vivado IDE:

- On Linux,
2. Change to the directory where the lab materials are stored:

```
cd <Extract_Dir>/Vivado_Tutorial
```

3. Launch the Vivado IDE: **vivado**

- On Windows,

1. Launch the Vivado Design Suite IDE:

```
Start > All Programs > Xilinx Design Tools > Vivado 2014.x > Vivado 2014.x1
```

2. As an alternative, click the **Vivado 2014.x** Desktop icon to start the Vivado IDE.

The Vivado IDE Getting Started page contains links to open or create projects and to view documentation.

¹ Your Vivado Design Suite installation may be called something different than **Xilinx Design Tools** on the **Start** menu.

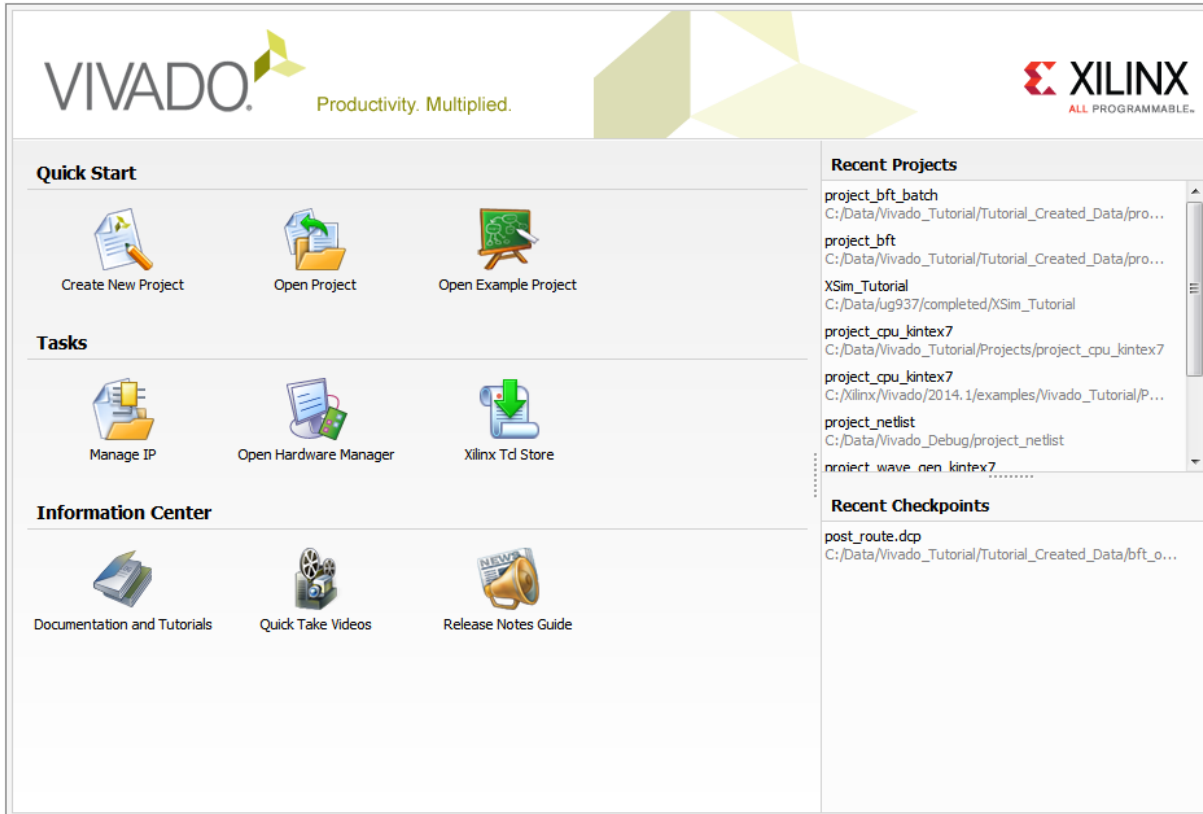


Figure 1: Vivado IDE Getting Started Page

3. In the Getting Started page, select **Create New Project**.
4. In the New Project Wizard, click **Next** to confirm the project creation.
5. 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`

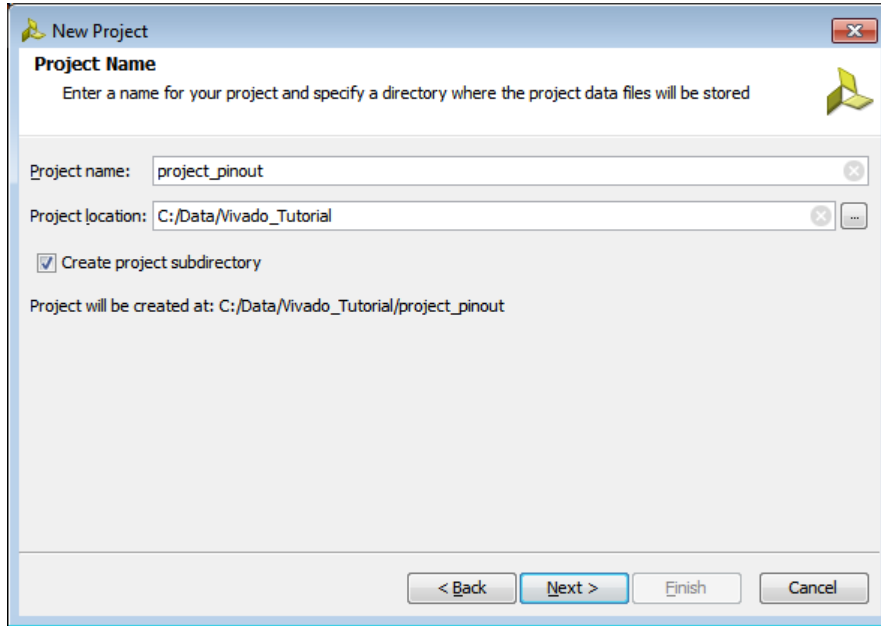


Figure 2: New Project Wizard—Project Name Page

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

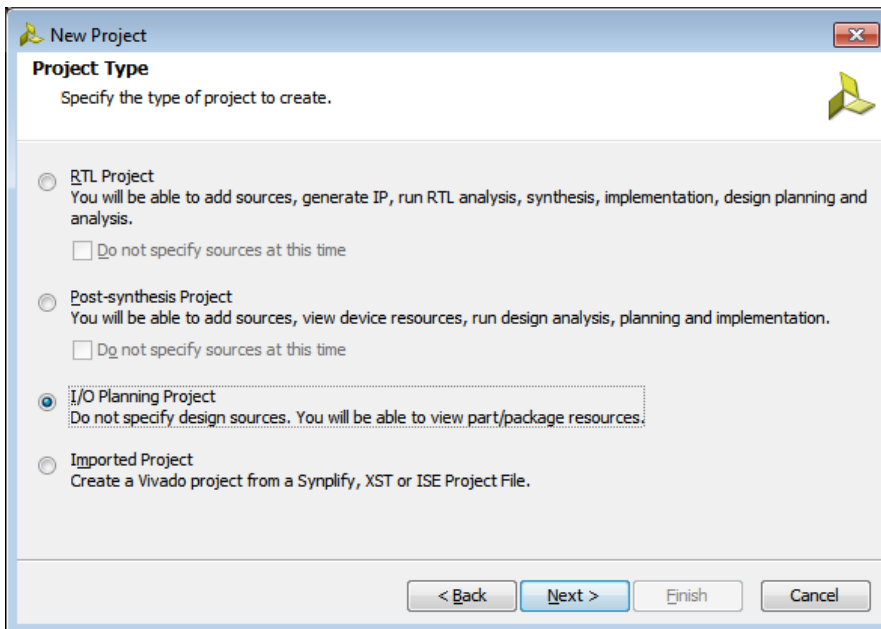


Figure 3: New Project Wizard—Project Type Page

7. In the Import Ports page, select **Do not import I/O ports at this time**, and click **Next**.
8. 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.

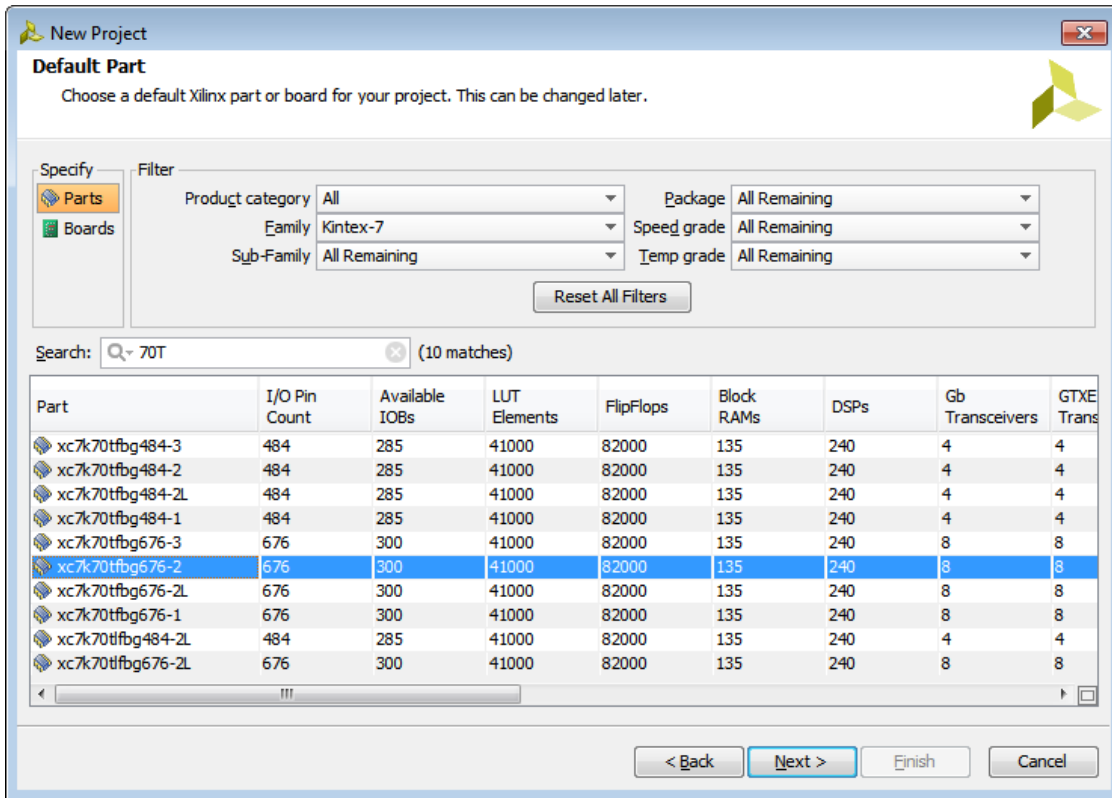


Figure 4: New Project Wizard—Default Part Page

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

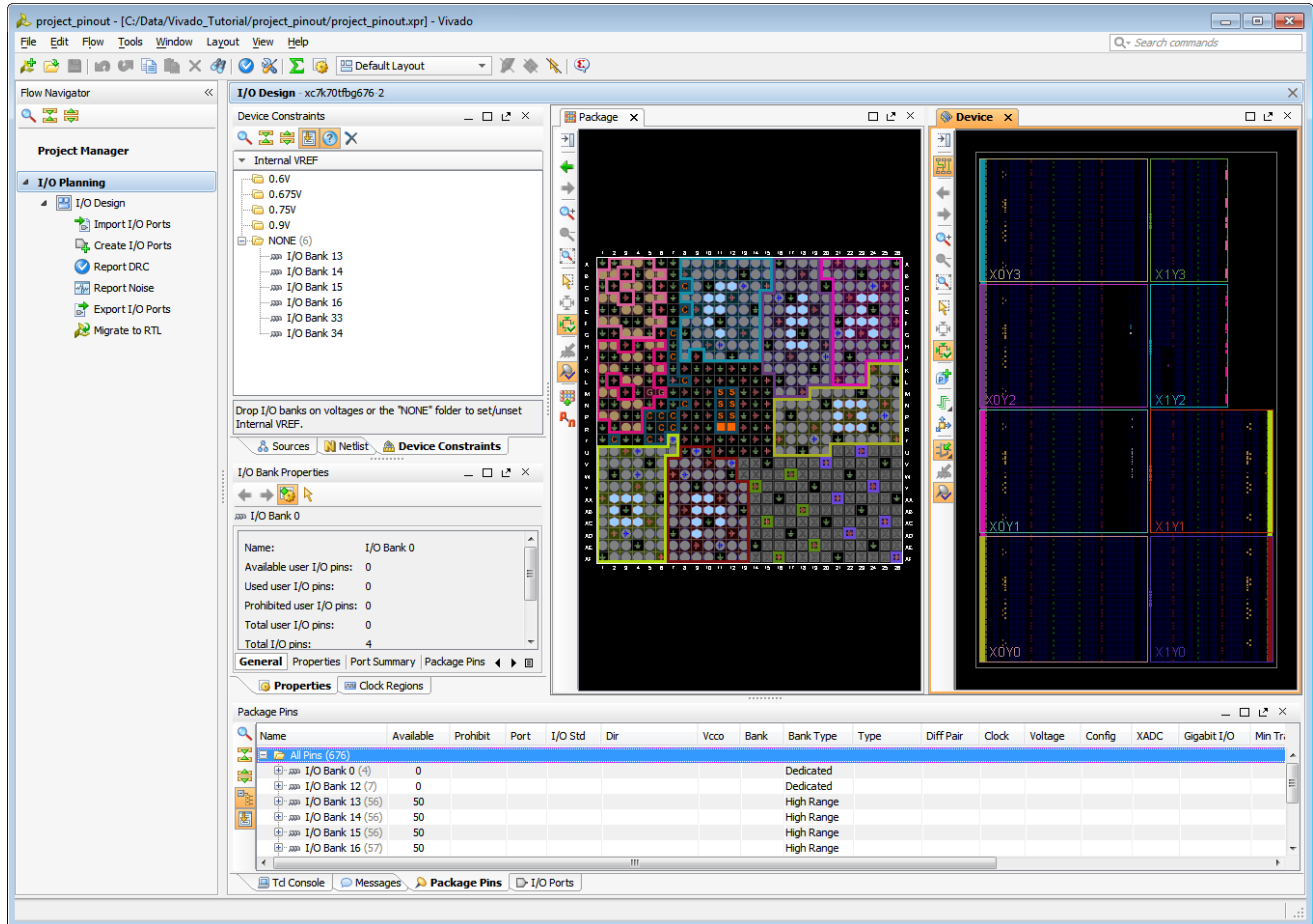


Figure 5: I/O Planning View Layout

- Explore the various windows in the I/O Planning view layout. Many windows are empty because I/O ports are not yet defined.
- Right-click on either the Device or Package window tab, and select **New Vertical Group**.

Notice that the Device and Package windows are displayed side by side as shown in [Figure 5](#). 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 I/O Bank 33**.

Notice that I/O Bank 33 is selected in the Device and Package windows. 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.



TIP: You can also highlight an I/O bank in the Package window by clicking the Package View Layers toolbar button. In the Package Options, expand the I/O Banks, right-click an I/O bank, and select Select Objects.

You can also use the Package 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.

2. 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.

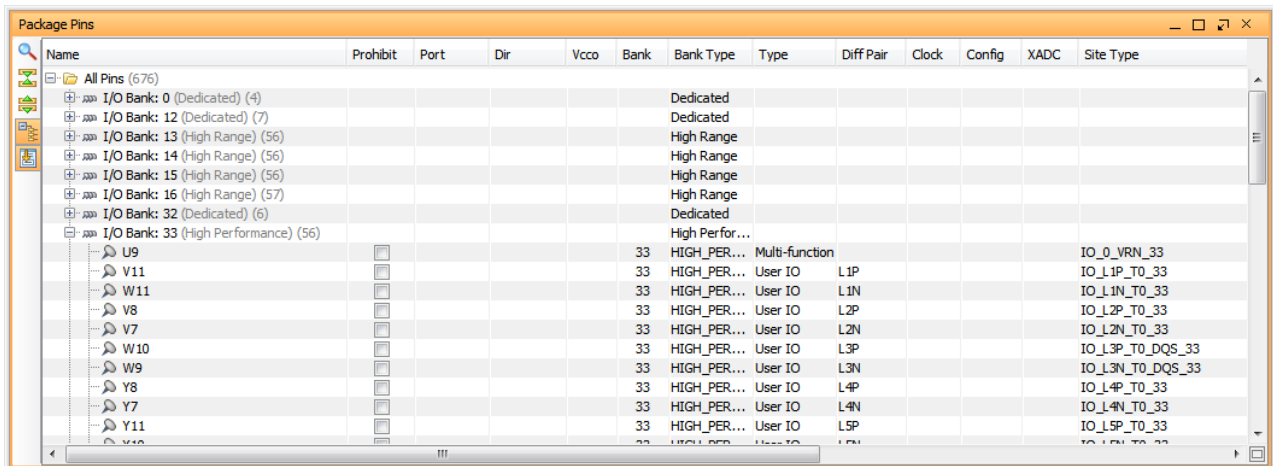


Figure 6: Package Pins Window

- In the I/O Bank Properties window, select the **General** view.

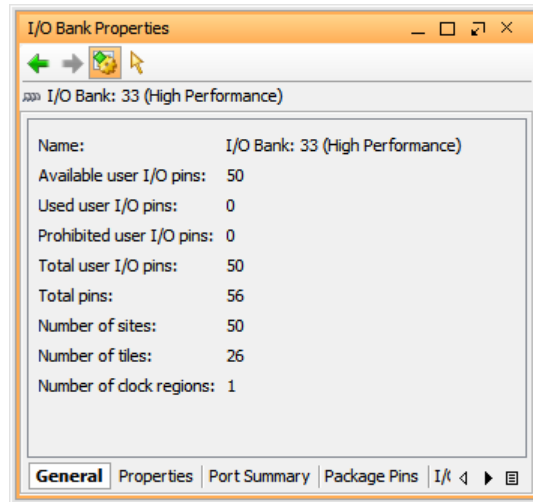







Figure 7: I/O Bank Properties


Review the available and used I/O count information. This is populated as I/O ports are assigned to the I/O bank. The I/O Bank Properties window allows you to identify 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** toolbar button .
- 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:

- In the Package Pins window, double-click the **Voltage** column header, and scroll to the top of the list to locate the **VREF** values.
- Select** the first **VREF** pin, and hold the **Shift** key while selecting the last to select all **VREF** Voltage pins.
- Right-click and select **Set Prohibit**.
- In the Package Pins window title bar, click the **Restore** button .
- In the main toolbar, click **Unselect All** .

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

6. In the Package window, zoom in to an area to view the prohibited pins, as shown in [Figure 8](#).
To zoom, click and drag the cursor from the upper left to the lower right of the zoom area. To zoom fit the Package window, click and drag the cursor from the lower right to the upper left.

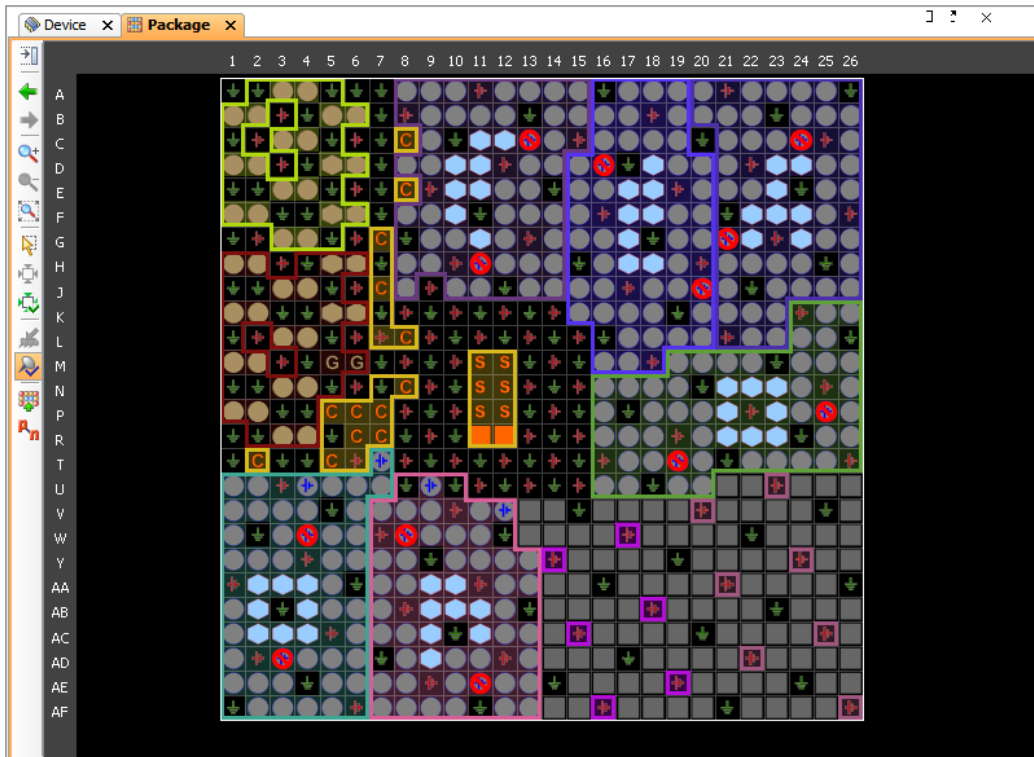


Figure 8: Prohibited V_{REF} Package Pins

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**.
3. Click **OK**.

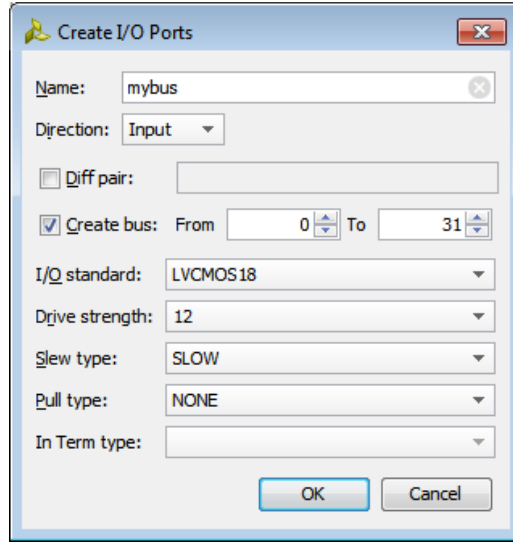


Figure 9: Create I/O Ports Dialog Box



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

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

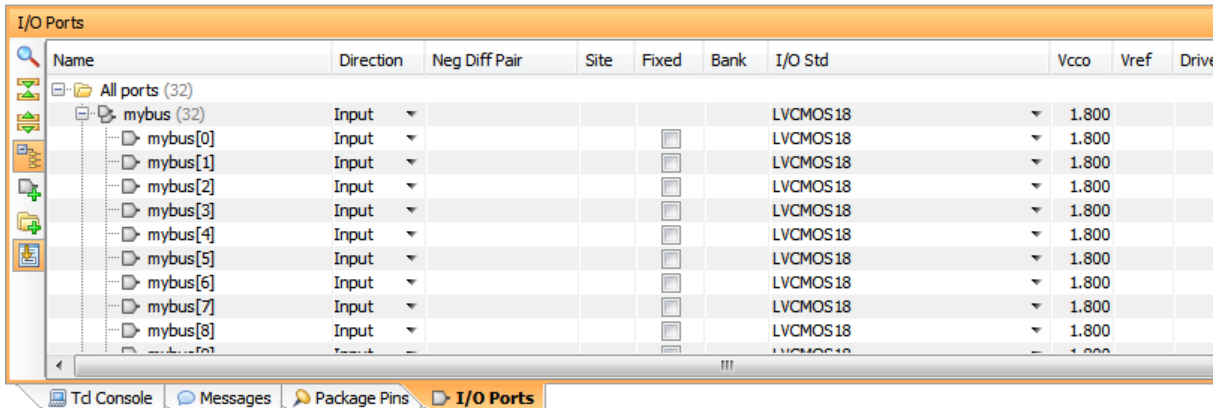


Figure 10: I/O Ports Window

4. 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, a bitstream has been generated, 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

Examine the I/O ports spreadsheet format and content, and exit without saving.

2. In the Vivado IDE Flow Navigator, select **Import I/O Ports**.
3. 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 [Figure 11](#). The buses are grouped together and are expandable.



CAUTION! Existing port definitions are overwritten when you import a CSV file, so import the file before defining new ports with the Create I/O Ports command.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	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_II_18	0.000				NONE	NONE	NONE
TXP_OUT (16)	Output		TXN_OUT			DIFF_HSTL_II_18	1.800			SLOW	NONE	NP_VTT_50_FP_V...	
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 11: 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.

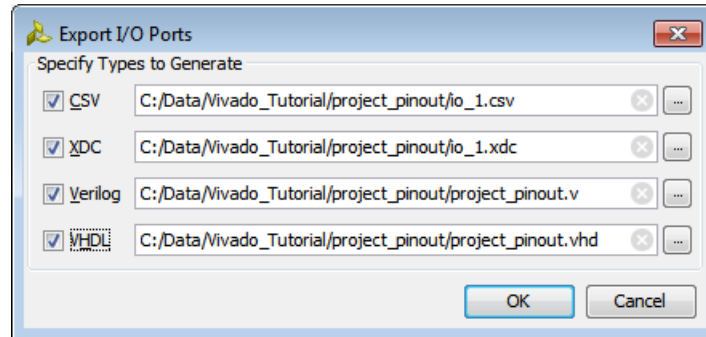


Figure 12: Export I/O Ports Dialog Box

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/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.*

Before migrating the I/O Planning project to an RTL project, you should save the current design constraints.

1. From the main menu, select the **File > Save Constraints** command.

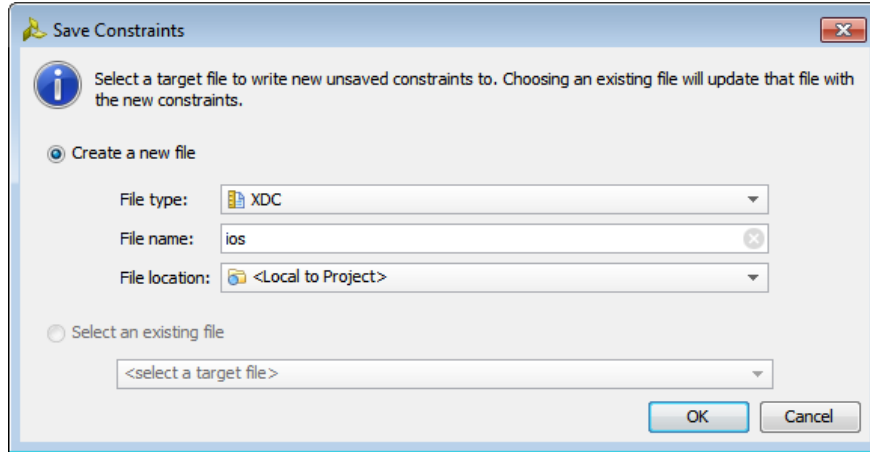


Figure 13: Save Constraints prior to RTL Migration

2. Specify **Create a new file**, with the following settings:

- File type: **XDC**
- File name: **ios.xdc**
- File location: **Local to Project**

3. Click **OK** to close the form.

The design constraints from the prior steps, including the pin prohibits and the I/O port definitions are saved to the specified `ios.xdc` constraints file, and added to the constraints set in the current project.

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

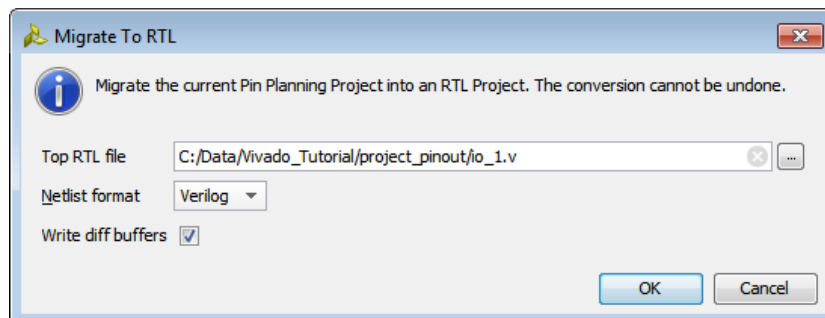


Figure 14: Migrate to RTL Dialog Box

- In the Migrate to RTL dialog box, click **OK** to accept the default file names and locations. Notice the RTL Project is now displayed. The Sources window contains the newly created source files.

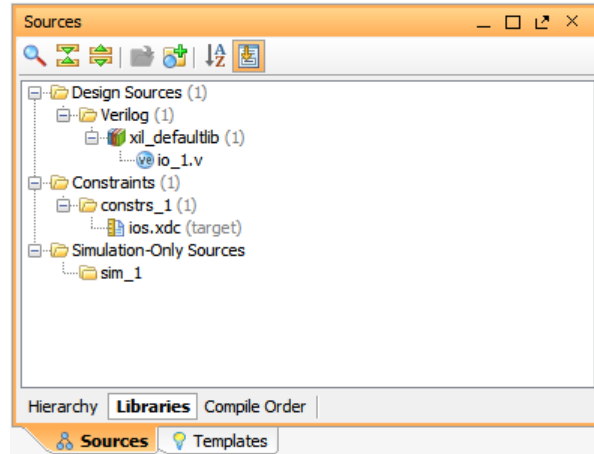


Figure 15: RTL Project Sources

- Double-click **io_1.v** and **ios.xdc** to open the files in the Vivado IDE Text Editor.
- Select **File > Close Project**.
- In the **Confirm** Close Project dialog box, click **OK**.

Conclusion

In this lab, you have created an I/O Planning project, and examined various aspects of the tools available within the Vivado IDE. You have examined the Package Pins and I/O Ports windows, as well as the Package window. You have created bus ports, and imported port definitions from a CSV file. Finally, you have converted your I/O Planning project into an RTL project to let you move from port placement to RTL design.

This concludes Lab #1. You can proceed at this time to Lab #2, or use the following steps to exit the tool:

- Click **File > Exit** to exit the Vivado Design Suite.

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

If you exited the Vivado IDE at the end of Lab #1, you will need to begin by opening the Vivado IDE. Otherwise, you can skip ahead to [Opening the Project](#).

Opening the Vivado IDE:

- On Linux,
 1. Change to the directory where the lab materials are stored:
cd <Extract_Dir>/Vivado_Tutorial
 2. Launch the Vivado IDE: **vivado**
- On Windows,
 1. Launch the Vivado Design Suite IDE:
Start > All Programs > Xilinx Design Tools > Vivado 2014.x > Vivado 2014.x²
 2. As an alternative, click the **Vivado 2014.x** Desktop icon to start the Vivado IDE.
The Vivado IDE Getting Started page opens.

Opening the Project

You can open the post-synthesis project as follows:

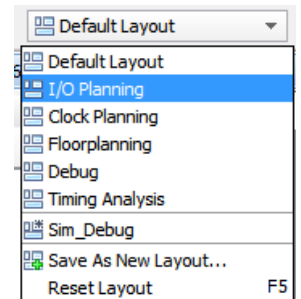
1. Select **File > Open Project**.
2. In the **Open Project** dialog box, browse to **select** the following project file:
`<Extract_Dir>/Vivado_Tutorial/Projects/project_cpu_netlist_kintex7/
project_cpu_netlist_kintex7.xpr`

² Your Vivado Design Suite installation may be called something different than **Xilinx Design Tools** on the **Start** menu.






TIP: Alternatively, select **Open Example Project > 7-Series Kintex > CPU (Synthesized)** from the Getting Started page. Use **File > Save Project As** to save a local writeable copy of the project at `<Extract_Dir>/Vivado_Tutorial`

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.
8. In the **Layout Selector** located next to the main toolbar, select the **I/O Planning** view layout.



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

9. **Examine** the I/O ports in the **I/O Ports** window.
10. In the I/O Ports window title bar, click the **Maximize** button .
11. In the I/O Ports window, click the **Expand All** toolbar button .
12. Click the **Group by Interface and Bus** toolbar button  to disable this feature.

The I/O ports now display as a flat list rather than grouped by bus.

Id	Name	Direction	Interface	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination
1	DataIn_pad_0_j[0]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
2	DataIn_pad_0_j[1]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
3	DataIn_pad_0_j[2]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
4	DataIn_pad_0_j[3]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
5	DataIn_pad_0_j[4]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
6	DataIn_pad_0_j[5]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
7	DataIn_pad_0_j[6]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
8	DataIn_pad_0_j[7]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
9	DataIn_pad_1_j[0]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
10	DataIn_pad_1_j[1]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
11	DataIn_pad_1_j[2]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
12	DataIn_pad_1_j[3]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
13	DataIn_pad_1_j[4]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
14	DataIn_pad_1_j[5]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
15	DataIn_pad_1_j[6]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE
16	DataIn_pad_1_j[7]	Input				<input type="checkbox"/>		default (LVCMOS18)	1.800				NONE	NONE

Figure 16: I/O Ports window

Step 2: 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: Because 7 series and UltraScale devices have low and high voltage I/O banks, all I/O ports must have explicit values specified for the `IOSTANDARD` and `PACKAGE_PIN` properties in order to generate a bitstream file for the design. The word *default* is displayed in red to indicate that these values must be assigned.

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. Select the first I/O port, hold the **Shift** key, and scroll down to select the last I/O port that is *NOT* a Neg Diff Pair port.

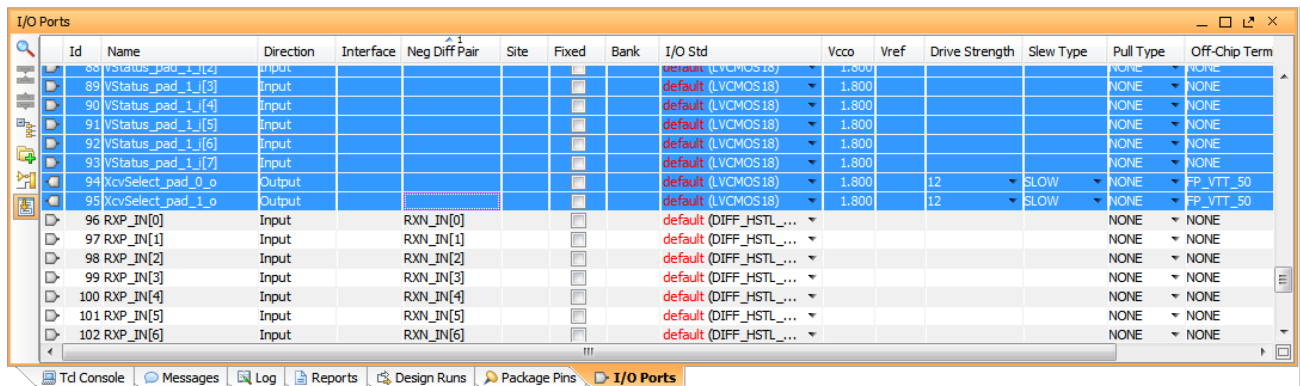


Figure 17: Single Ended Ports Selected

4. Right-click and select **Configure I/O Ports**.
5. 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, the I/O Std column will be empty for those ports.

6. Click the **Unselect All** button  in the toolbar menu.

Step 3: 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_`.

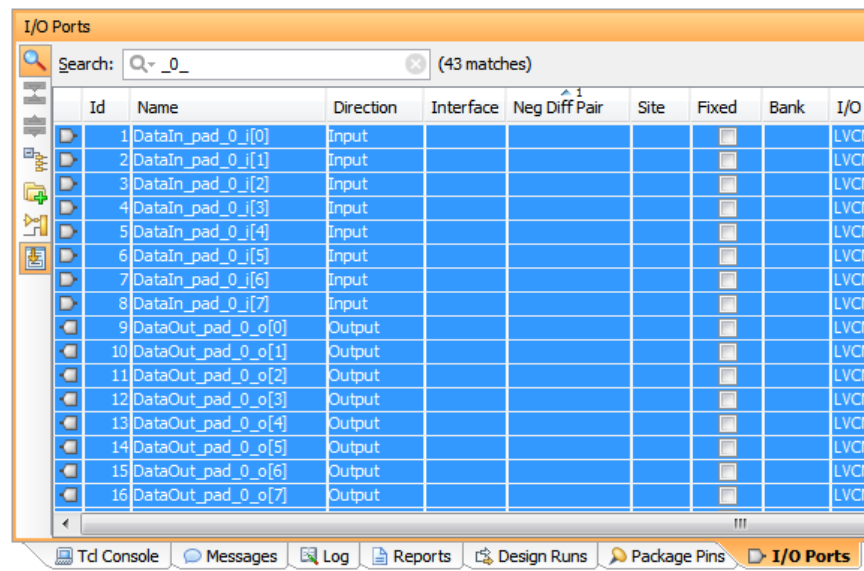


Figure 18: USB_0_ Related Ports

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

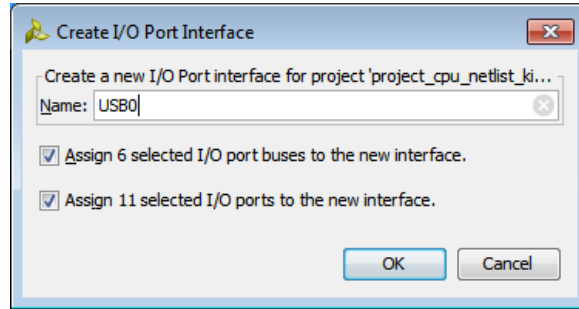








Figure 19: Create I/O Port Interface dialog box

6. Click the **Unselect All** button  in the toolbar menu.
7. In the Search field, change **_0_** to **_1_** and follow the same steps to create a **USB1** interface.
8. Click the **Unselect All** button  in the toolbar menu.
9. Click the **Show Search** toolbar button  to remove the Search filter.
10. Click to enable the **Group by Interface and Bus**  button.
11. Click **Collapse All**. 

The I/O Ports list is condensed with all of the USB related-ports in interface groups.

12. Expand the **Scalar ports** folder to view the clocks resets and other ports.
13. In the I/O Ports window title bar, click the **Restore** button 

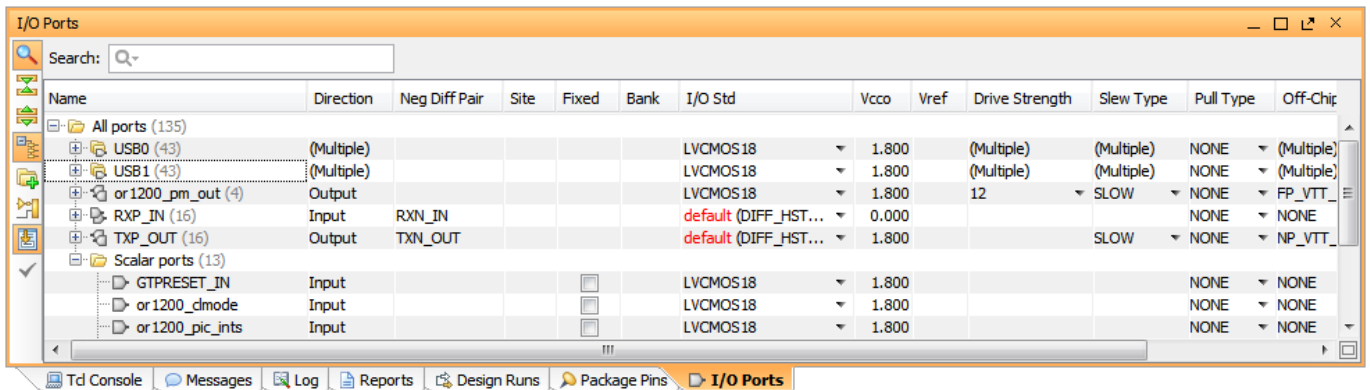



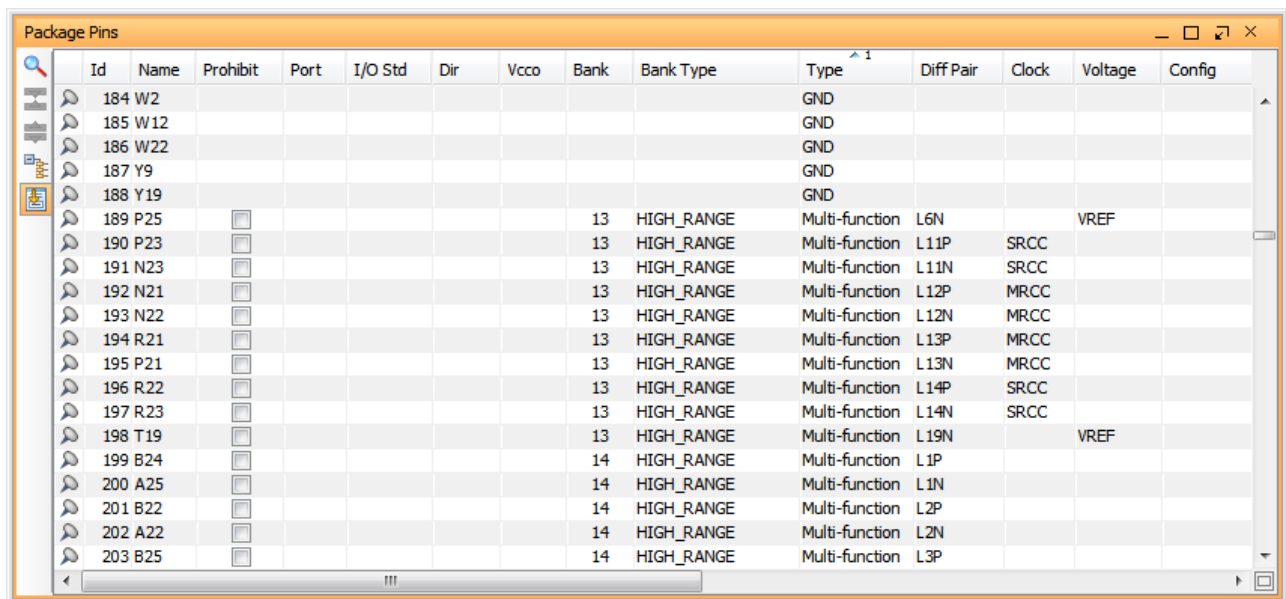


Figure 20: I/O Port Interface Groups and Scalar Ports

Step 4: 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 and select the multi-function pins.



Id	Name	Prohibit	Port	I/O Std	Dir	Vcco	Bank	Bank Type	Type	Diff Pair	Clock	Voltage	Config
184	W2								GND				
185	W12								GND				
186	W22								GND				
187	Y9								GND				
188	Y19								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			

Figure 21: Multi-Function Pins

7. **Examine** the **Config** (device configuration pins), **XADC**, and **Gigabit I/O** columns.
 These elements 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 table format windows which let you search and sort data by columns. See the “Using Data Table Windows” section in the *Vivado Design Suite User Guide: Using the Vivado IDE* ([UG893](#)).

Step 5: Setting Device Configuration Modes

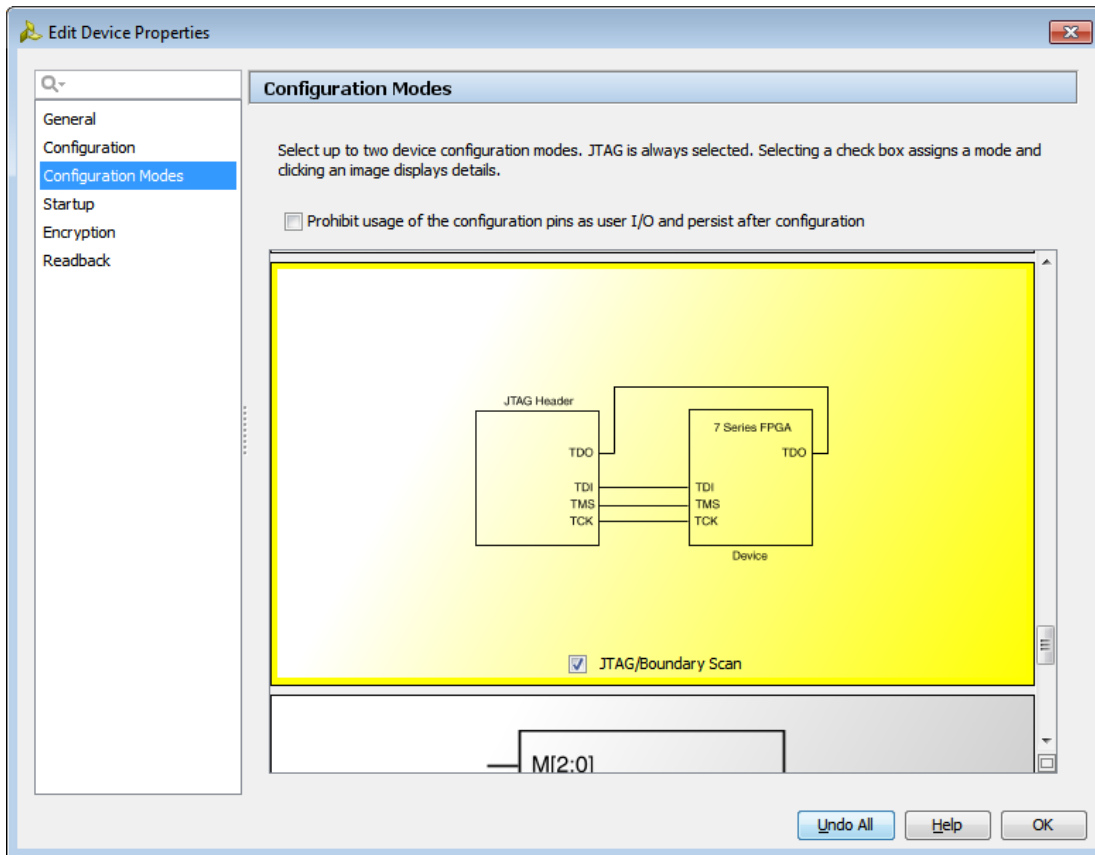


Figure 22: Edit Device Properties – Configuration Modes

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 > Edit Device Properties > Configuration Modes**.
2. In the **Edit Device Properties > Configuration Modes** dialog box, seen in [Figure 22](#), select one or two of the **modes** to view the descriptions, schematics, and related data sheets.
3. Leave the mode set to **JTAG/Boundary Scan** and click **OK**.
4. If prompted, click **OK** to **set prohibits** on the configuration pins and to **sort** the Package Pins window by Config pins.

Step 6: Defining Alternate Compatible Parts

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.
3. Click **OK**.

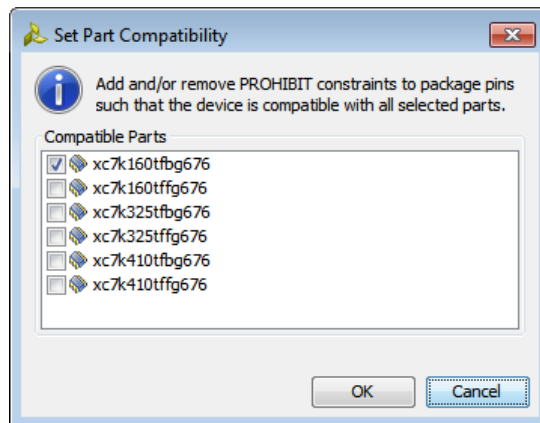


Figure 23: Set Part Compatibility Dialog Box

4. In the confirmation dialog box, click **OK** to indicate that no prohibit constraints were placed. Prohibits are assigned to pins based on the most restrictive parts. In this example you are currently targeting the smallest device for this package, so no prohibited pins are added or removed.

Step 7: Placing I/O Ports


The Vivado IDE provides several ways to place the I/O ports onto either package pins in the Package window, or I/O die pads in the Device window. 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 semi-automatic placement modes. This step covers the following methods:

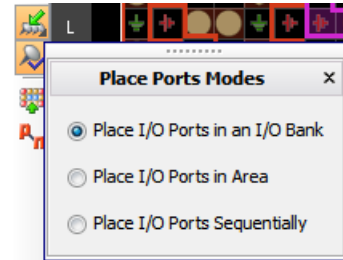
- [Placing I/O Ports in an I/O Bank](#)
- [Placing I/O Ports in an Area](#)
- [Placing I/O Ports Sequentially](#)
- [Placing I/O Ports Automatically](#)



TIP: By default, the Vivado IDE uses interactive DRCs during I/O placement. The `report_drc` command should still be used to validate interactively placed I/O Ports.

Placing I/O 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.

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.

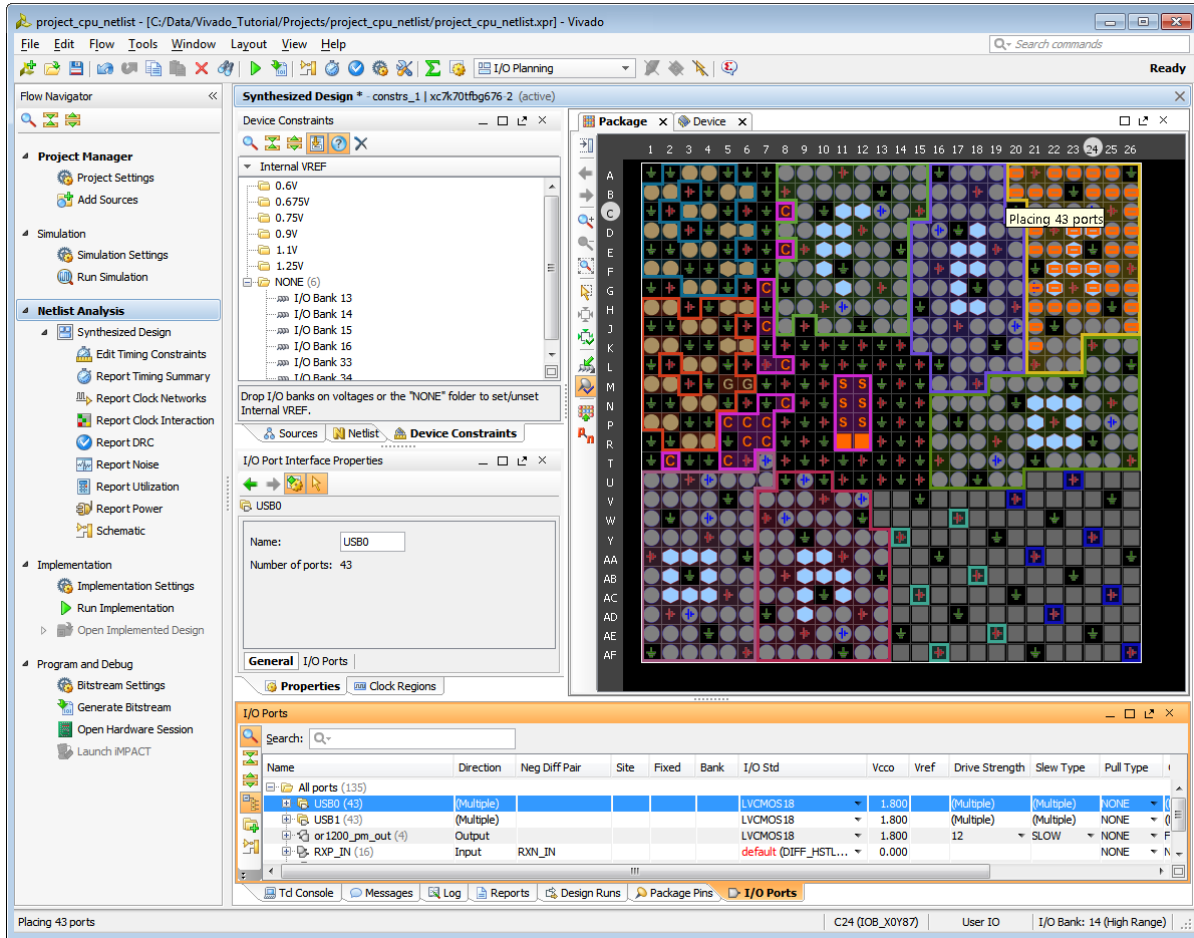

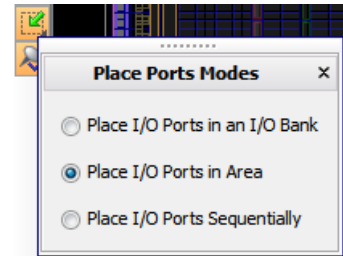


Figure 24: I/O Ports Placed in an I/O Bank

Placing I/O Ports in an 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.

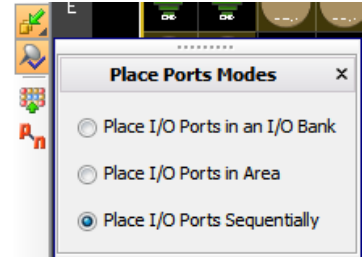
4. Draw a rectangle starting at the bottom of the first I/O bank in the top half of the device, and drag it up and to the right until all I/O ports are placed in the rectangle within the top clock region.



Figure 25: USB1 I/O Ports Placed in an Area

Placing I/O Ports Sequentially

1. In the I/O Ports window, select the **RXP_IN** bus.
2. **Zoom in** on the **upper left** corner of the Package window to show the GT pins.
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 located in the upper left. Both diff pairs associated with the GTs are placed on legal sites. If you select a site that is not legal, a tooltip indicates why it is not a legal site.

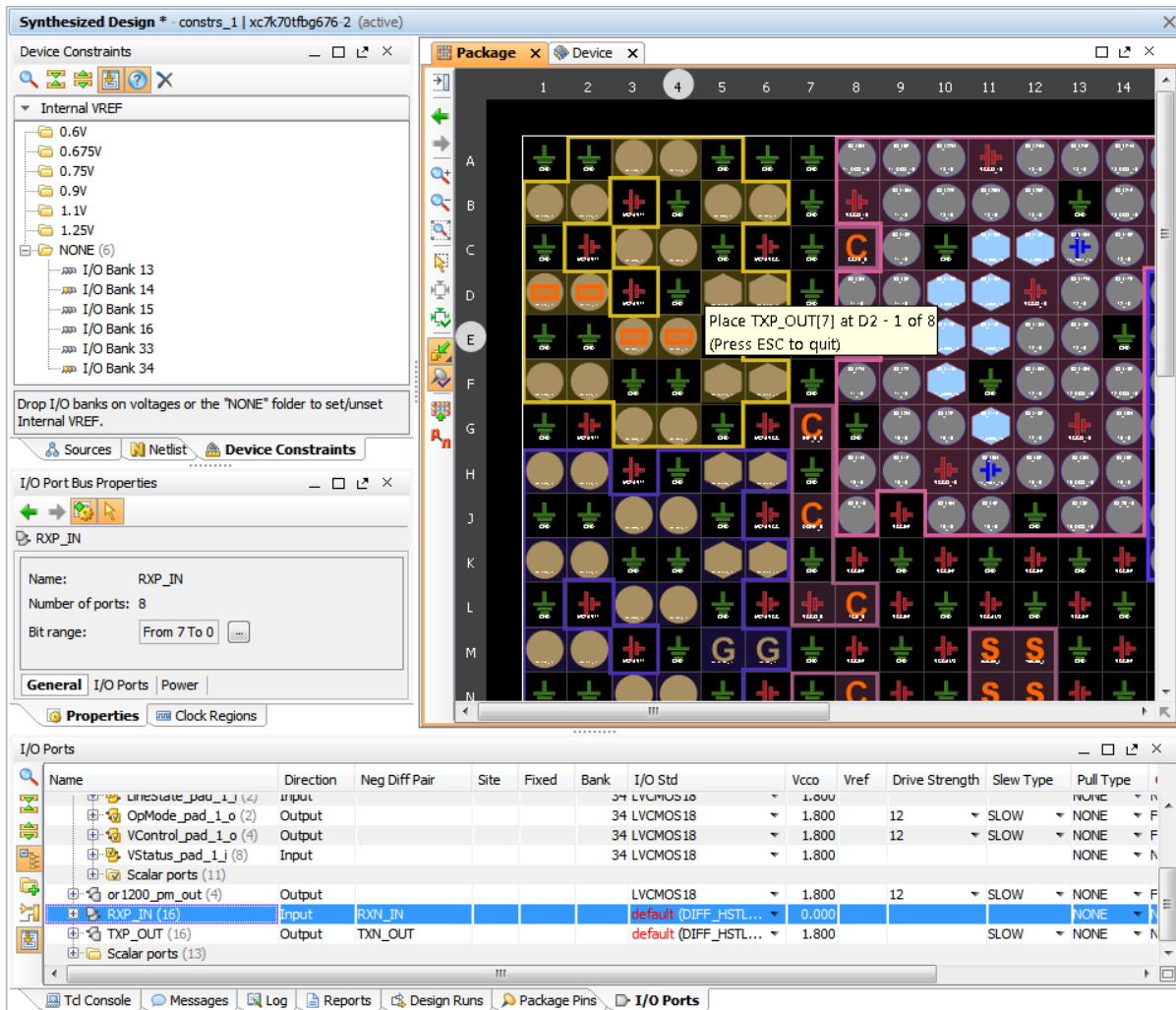




Figure 26: Diff Pair I/O Bus Ports Placed Sequentially

- 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.



TIP: GT logic objects are automatically grouped by the Vivado IDE to ensure proper behavior when I/O ports are placed or moved. Both transmitter and receiver diff pair ports, as well as the GT, are all placed and moved as a group.

- In the I/O Ports view, expand the **Scalar ports** folder, and use the **Shift** key to select the 4 GT reference clock ports, **TILE0_REFCLK_PAD_P_IN** to **TILE3_REFCLK_PAD_P_IN**.
- In the Package window, click the Place Ports toolbar button , and select **Place I/O Ports Sequentially**.
- 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, located in the upper left of the Package window.
- Place all** four GT reference clock ports.
- Click the **Unselect All** button  in the toolbar menu.

Placing I/O Ports Automatically

The Vivado Design Suite can also automatically place I/O ports onto the device, while adhering to various design rules, signal types, IOSTANDARDS, and grouping interfaces.

- Select **Tools > I/O Planning > Auto-Place I/O Ports**.

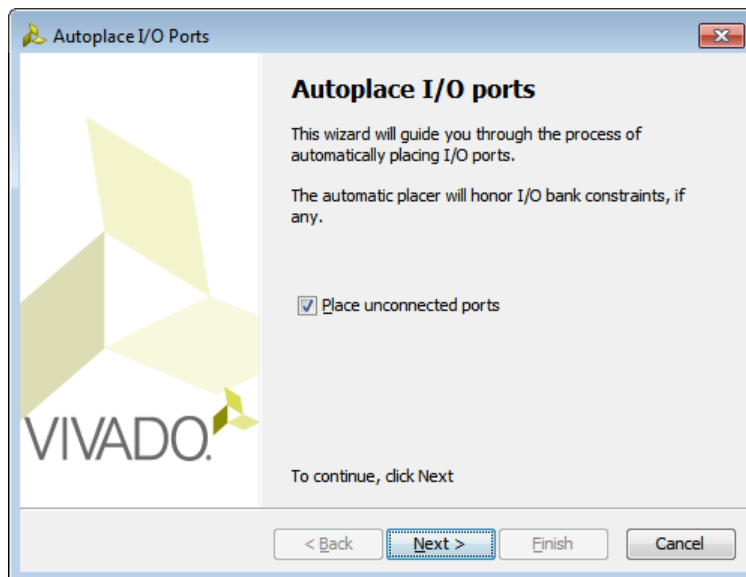


Figure 27: Autoplace I/O Ports

If you have any ports currently selected, you will be prompted to place the selected ports, or all ports in the design.

2. As seen in [Figure 27](#), check the **Place unconnected ports** checkbox, and click **Next**.
3. In the Placed I/O Ports dialog box, select **Keep these 126 ports in their current locations**, and click **Next**.

Note: 126 is the number of ports placed so far. Your number may vary.

4. In the Summary page, click **Finish** to autoplace the remaining ports.
5. Click **OK** to confirm that all I/O ports were placed.
6. **Select any port** in the I/O Ports window and use **Ctrl-A** to select all I/O Ports.
7. **Right-click** and use the **Fix Ports** command to lock all I/O Ports in place.

Step 8: 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.

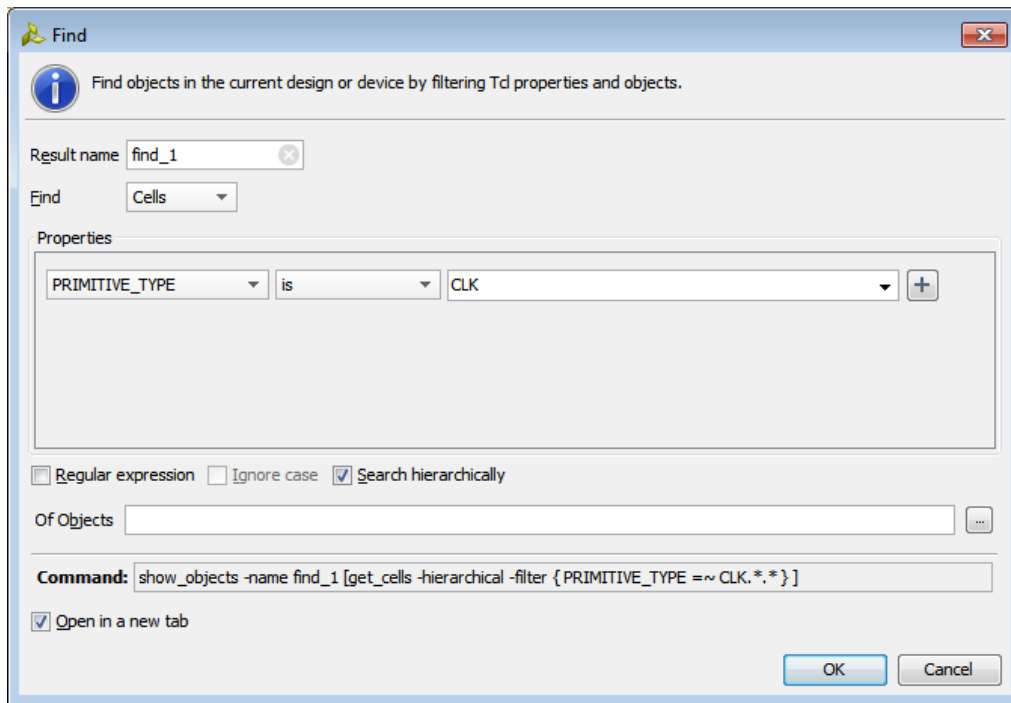


Figure 28: Find Dialog Box to Search for Clock Logic

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, set the following as seen in [Figure 28](#):

- Find: **Cells**
- **Primitive Type**
- **is**
- **CLK**

3. Click **OK**.

4. In the **Find Results** window, scroll down the list of **Cells**, and observe the following:

- MMCME2_ADV
- BUFG

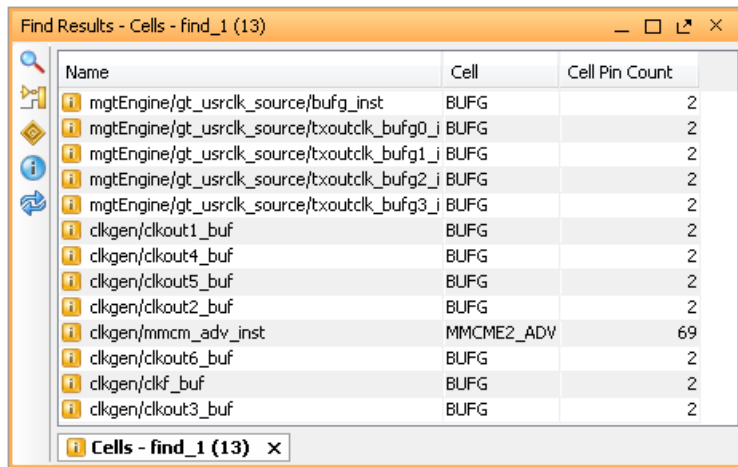



Figure 29: Clock Objects in the Find Results Window

Step 9: 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 **MMCME2_ADV** cell.
2. Click the **Schematic** toolbar button , or press the **F4** key.
3. In the Schematic window, **double-click** the **CLKIN1** input pin on the upper left of the mixed mode clock manager (MMCM) module.

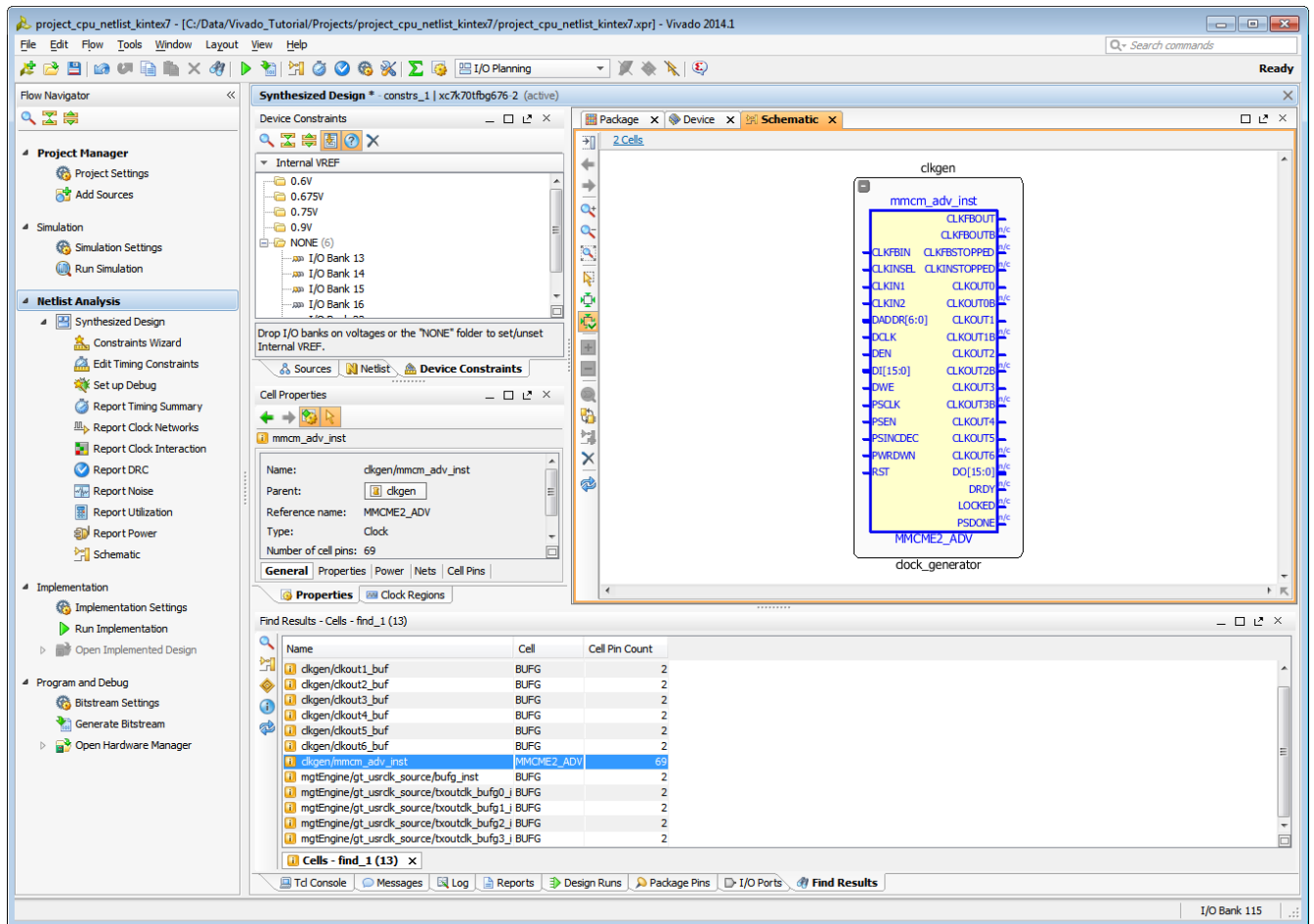


Figure 30: Tracing Clock Logic in the Schematic

Step 10: 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 .

Notice that the clock regions, I/O banks, and various device resources display in their relative location as found on the device.

3. **Expand** or **collapse** sections in the Clock Resources window to hide or display the resources.

Placed clock logic is displayed under the **Cell** columns. Placed clock resources, such the GTXE2s and their associated Diff Pair ports, are also displayed, as well as BUFGRs and MMCMs.

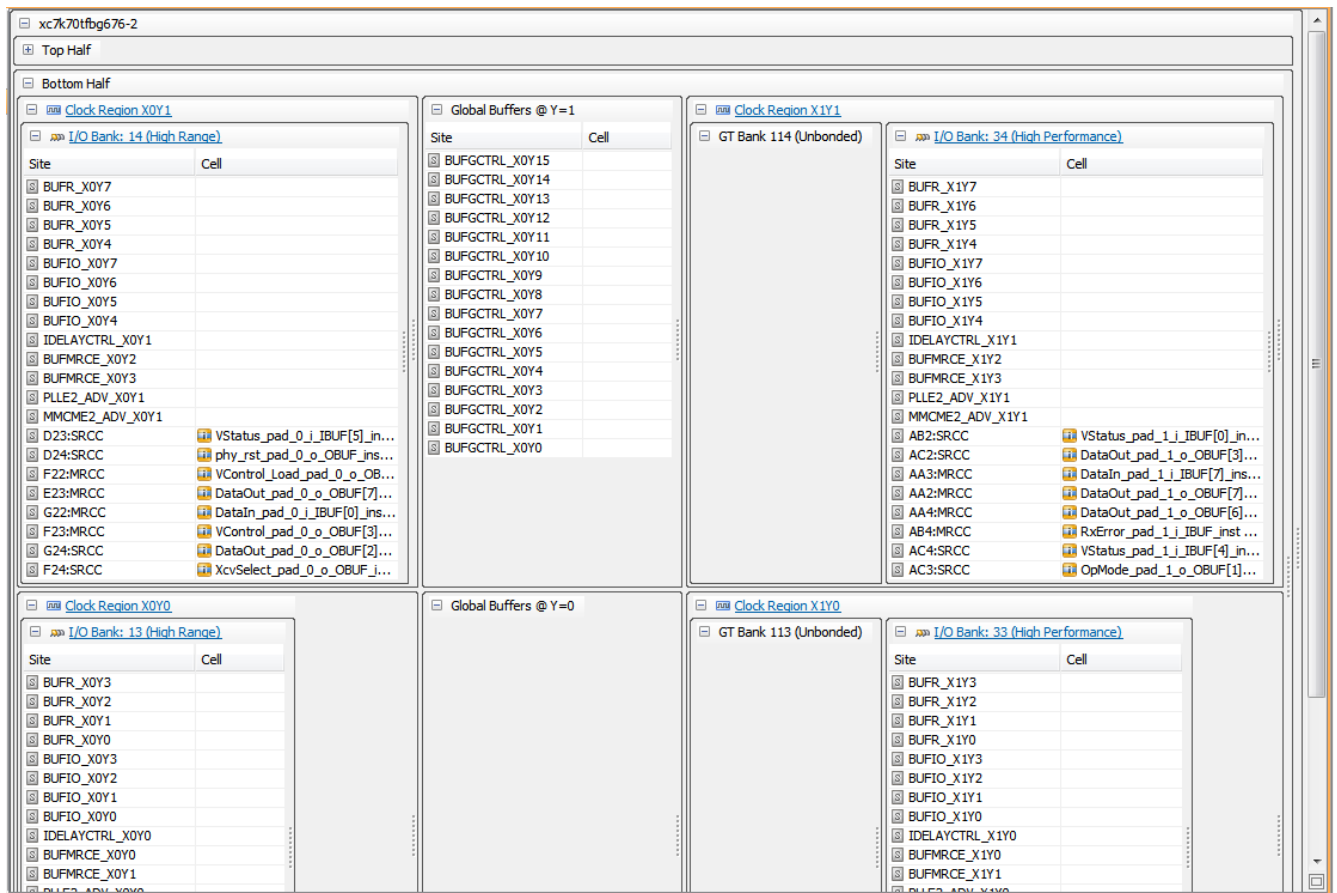


Figure 32: Clock Resources Window

Step 11: Placing the MMCM Instance

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

1. In the Find Results window, **select** the **MMCME2_ADV** cell and drag it into the Clock Resources window in the **Cell** column next to one of the MMCME2_ADV Sites.

Notice how you can place clock and related I/O logic in the Clock Resources window.

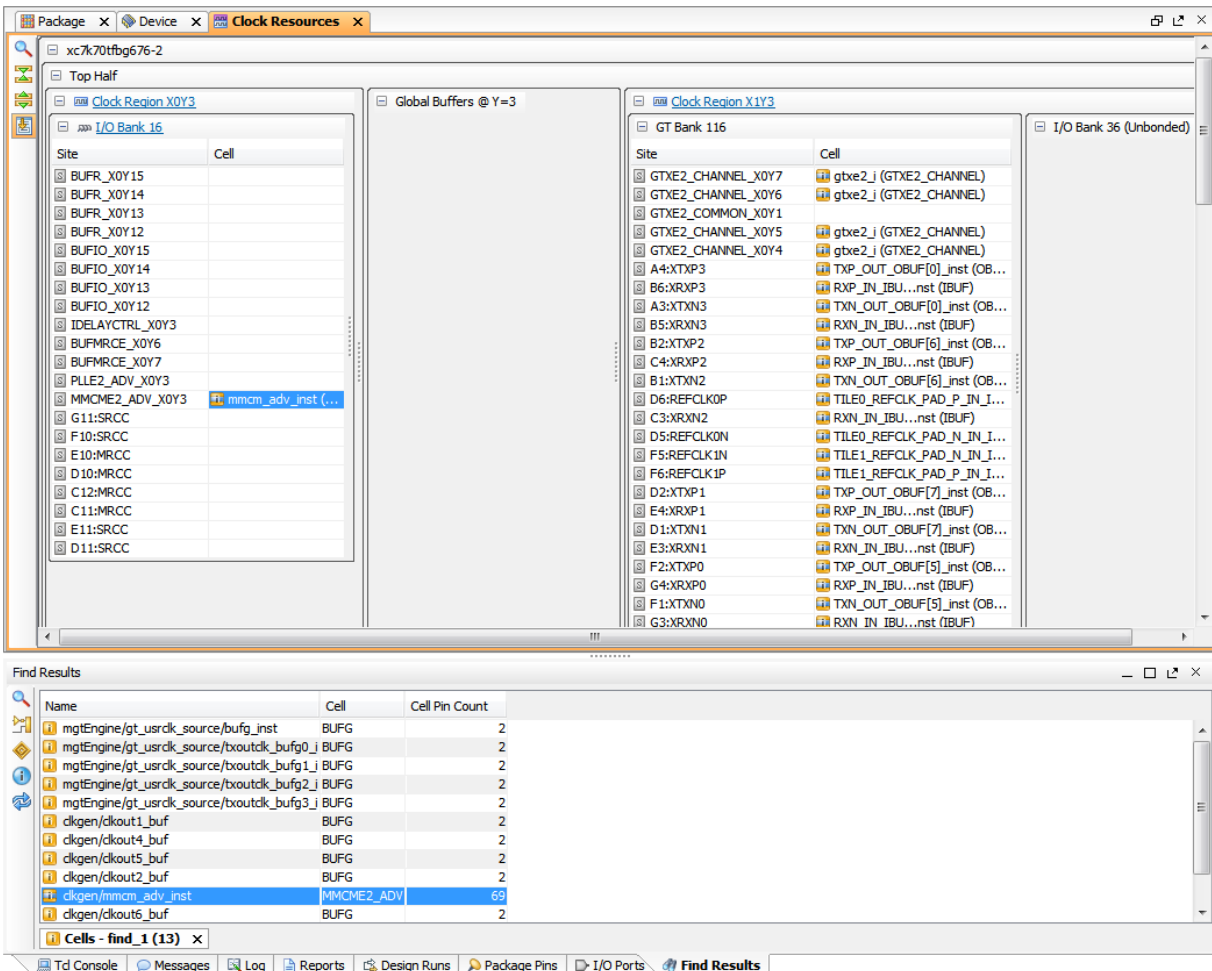


Figure 33: Placing Clock Logic

2. **Close** the **Find Results** window.
3. In the Clock Resources window title bar, click the **Restore** button.
4. Click the **Device** window tab, and you should see the placed MMCM cell.
5. **Click** the **Unselect All** button on the main toolbar menu.

Step 12: 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, click Clear All and then select the **Pin Planning** and **Clocking** rule categories.

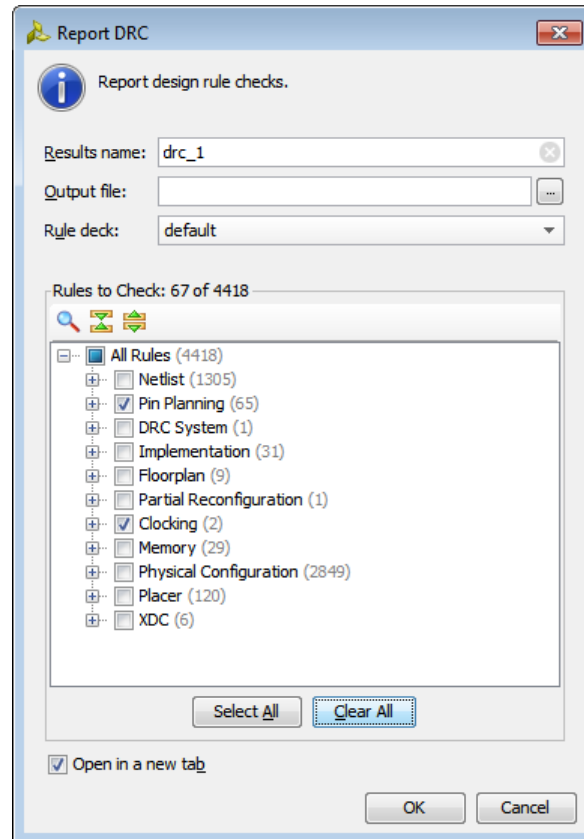


Figure 34: Report DRC Dialog Box

3. **Expand** the **selected rules** to examine the rule types, and click **OK** to run Report DRC.
4. **Examine** any errors, warnings, or messages returned by the DRC report., or **click OK** to accept that there were no problems reported.



TIP: You may encounter the PLCK-23 error related to the sub-optimal placement of the MMCM and the clock-capable IO pin it is connected to. For the purposes of this Tutorial, you can ignore the error, or move the MMCM into Clock Region X1Y1 to correct the error.

Step 13: Running Simultaneous Switching Noise Analysis

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

1. In the Flow Navigator, under the Synthesized Design section, click **Report Noise**.

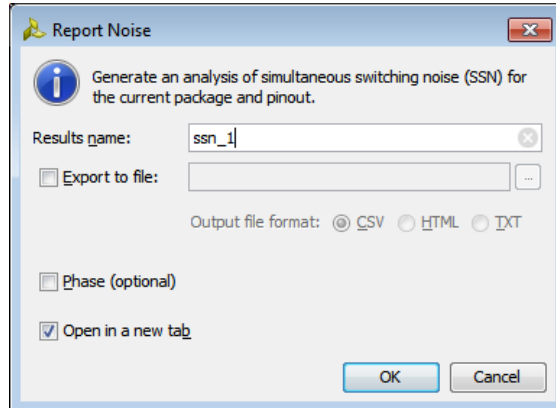


Figure 35: Run SSN Analysis

2. In the **Run SSN Analysis** dialog box, leave the default options selected and click **OK**.

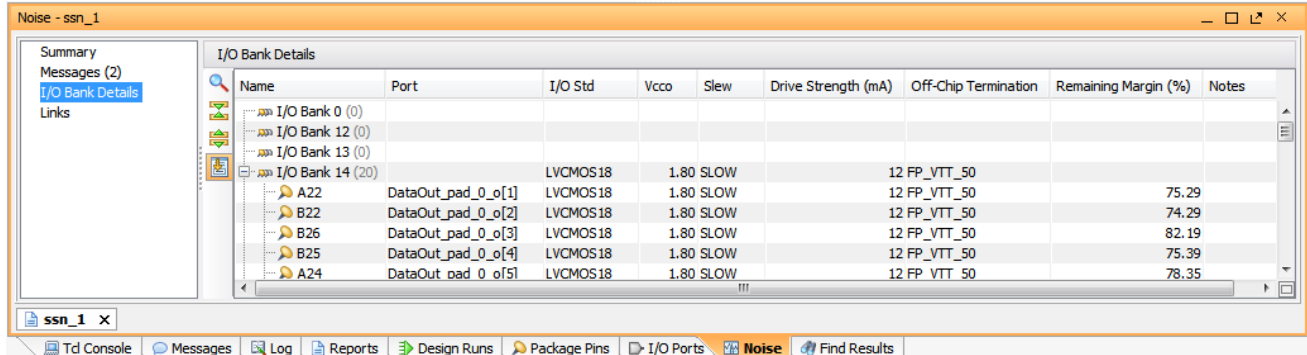


Figure 36: Noise Results Window

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**, **Messages**, and **Links**, to examine the reported information.
6. In the Noise window title bar, click the **Restore** button.

Step 14: 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. To save the changes you made, you must write the constraints into the active constraint set, and the target constraint file. For more information on working with constraints, see the *Vivado Design Suite User Guide: Using Constraints* ([UG903](#)).

You can write the constraints to the target constraint file as follows:

1. Select **File > Save Constraints**.

Note: You can also simply click the Save Constraints button in the main toolbar menu. 

2. In the Sources window, **double-click** the `top.xdc` file under the constraint folder `constr_1`.
3. Notice the new physical constraints written to the XDC file.
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.