I/O Pin Planning Tutorial

PlanAhead Design Tool

UG674 (v 14.1) May 8, 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, 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



Table of Contents

Software Requirements	4
Hardware Requirements	5
Tutorial Design Description	5
Locating Tutorial Design Files	5
Part 1: Using I/O Pin Planning Projects (pre-synthesis)	6
Step 1: Creating a New Project and Exploring the Views Step 2: Examining Device I/O Resources Step 3: Prohibiting Pins from I/O Assignment Step 4: Creating and Configuring I/O Ports	10 12
Step 5: Importing an I/O Port List Step 6: Exporting the Device and I/O Pin Assignments Step 7: Migrating the I/O Planning Project to an RTL Project	15 16
Part 2: I/O Planning Features using a Synthesized Design	
Step 8: Opening the Synthesized Netlist-Based Project Step 9: Examining the I/O Ports in the Design Step 10: Configuring I/Os and Setting I/O Standards	19
Step 11: Creating I/O Port Interfaces	20
Step 12: Viewing Multi-function Package Pins Step 13: Setting Device Configuration Modes	24
Step 14: Defining Alternate Compatible Devices Step 15: Placing I/O Ports	25
Step 16: Placing Clock Logic Step 17: Using the Schematic to Trace Clock Logic	
Step 18: Exploring the Clock Resources View Step 19: Placing the MCM Instance	
Step 20: Running Design Rule Checks - "DRC" Step 21: Running Simultaneous Switching Noise Analysis - "SSN"	
Step 22: Updating the Constraint Files with Interactive Assignments Conclusion	



I/O Pin Planning Tutorial

This tutorial introduces the Xilinx[®] PlanAhead[™] software capabilities and benefits when performing I/O pin assignment for FPGA devices. It describes the procedure for creating and assigning I/O ports to physical package pins. The I/O Planning view environment enables you to create, import, and configure the initial list of I/O ports. You can group the related ports into Interfaces and then assign them to package pins.

The objective of this tutorial is to familiarize you with the I/O pin planning process using the I/O Planning functionality in the PlanAhead tool. There are two parts to this tutorial that can be performed independently. The first part briefly describes I/O planning capabilities prior to having a synthesized netlist or RTL Sources with I/O ports defined. The second part describes I/O planning functionality after synthesis. Most of the PlanAhead I/O planning features are described in Part2. However, many of them are available at prior to running synthesis as well.

The capabilities include semi-automated interactive modes to allow controlled I/O port assignment. Fully automatic pin placement is also available for some device architectures. The I/O Planning view environment shows the relationship of the physical package pins and banks with their corresponding I/O die pads. Intelligent decisions can be made to optimize the connectivity between the PCB and the FGPA device.

You can perform I/O pin assignment at various stages of the design cycle. 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 PCB schematic symbol or Hardware Description Language (HDL) header generation.

The PlanAhead tool also enables you to I/O pin plan in the elaborated Register Transfer Level (RTL) design or in the synthesized netlist design. The PlanAhead tool performs more comprehensive I/O and clocking DRCs when using a netlist design.

Not all commands or command options are covered in this tutorial. This tutorial uses the features contained in the PlanAhead tool, which is bundled as a part of ISE® Design Suite version 14.1.

Software Requirements

The PlanAhead tool is installed with ISE Design Suite software. Before starting the tutorial, be sure that the PlanAhead tool is operational, and that the tutorial design data is installed.



For installation instructions and information, see the *ISE Design Suite: Installation and Licensing Guide (UG798)* at <u>http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/iil.pdf</u>.

Hardware Requirements

Xilinx recommends a minimum of 2 GB of RAM when using the PlanAhead tool on larger devices. For this tutorial, a smaller xc7k70t design is used, and the number of designs open at one time is limited. Although 1 GB is sufficient, it can impact performance.

Tutorial Design Description

The small sample design used in this tutorial includes:

- A RISC processor CPU core
- A pseudo FFT
- Four gigabit transceivers (GTs)
- Two USB interfaces

The design targets an xc7k70t device. A small design is used to:

- Allow the tutorial to be run with minimal hardware requirements
- Enable timely completion of the tutorials
- Minimize data size

Locating Tutorial Design Files

Copy the files from the ISE software installation area:

/ISE_DS/PlanAhead/testcases/PlanAhead_Tutorial.zip

Extract the zip file contents into any write-accessible location.

The unzipped PlanAhead_Tutorial data directory is referred to in this tutorial as the <Extract_Dir>.

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

Part 1: Using I/O Pin Planning Projects (pre-synthesis)

The PlanAhead tool provides an I/O Pin Planning view layout that displays views more applicable to placing I/O Ports and clock logic. You can open the I/O planning layout without a design to analyze device resources.

These first few I/O Pin Planning steps involve pre-synthesis techniques to begin I/O Planning early prior to a synthesized netlist. Most of the features are also available for an Elaborated RTL, Synthesized or Implemented Design. I/O Pin Planning can be performed at any stage of the design process. Post-synthesis I/O Planning enables a much more robust set of DRCs and clock placement capabilities.

Step 1: Creating a New Project and Exploring the Views

- 1. Open the PlanAhead tool and create the project_pinout I/O Pin Planning project.
- On Windows, double-click the Xilinx PlanAhead 14.1 Desktop icon, or select: Start > All Programs > Xilinx Design Tools > ISE Design Suite 14.1 > PlanAhead > PlanAhead.
- 3. On Linux, go to

<Extract_Dir>/PlanAhead_Tutorial/Tutorial_Created_Data directory and type planAhead.

- 4. In the Getting Started page, select **Create New Project**.
- 5. Click **Next** to confirm the project creation and to display the Project Name dialog box.

💽 New Project	Decomentation 🗮	
Project Name Enter a name	e for your project and specify a directory where the project data files will be stored	3
Project name:	project_pinout	3
Project location:	C:\14.1_Tutorials\PlanAhead_Tutorial\Tutorial_Created_Data	
🔽 Create Proje	ct Subdirectory	
Project will be cr	eated at: C:\\Tutorial_Created_Data\project_pinout	
	< Back Next > Finish Cancel	

Figure 1: New Project Name and Location Dialog Box

www.xilinx.com



- 6. Type the Project name, **project_pinout**.
- 7. Enter the Project location:

< *Extract_Dir*>/PlanAhead_Tutorial/Tutorial_Created_Data.

- 8. Click **Next** to open the Design Source dialog box.
- 9. Select I/O Planning Project.

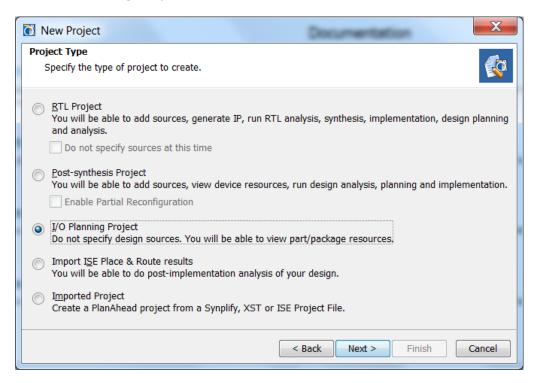


Figure 2: Specifying an I/O Pin Planning Project

- 10. Click **Next** to open the Import Ports dialog box.
- 11. Select **Do not import I/O ports at this time**.



New Project	Documentation	X
Import Ports (o	ptional)	
	cify a CSV, XDC, or UCF file to define and configure your ports. If you skip this step import ports later and/or create ports manually.	
O Import CSV:		
Import UCF:		
	t I/O ports at this time	
	< Back Next > Finish	Cancel

Figure 3: Import Ports from Existing UCF or CSV Files

- 12. Click **Next** to open the Default Part selector dialog box.
- 13. In the Filter section, click the **Family** pull down menu and select **Kintex-7**. Notice the list is filtered to show those devices.
- 14. In the Search field, type **70T**. Notice the 70T devices.



Default Par Choose a	-	nx part or b	oard for your p	roject. This c	an be change	d later.		£a
		· .		·				
Specify	Filter							
Parts	Product Cat	tegory All		Ψ.	Package	All Rema	ining	-
Boards	l l	Family Kint	tex-7	*	Speed Grade	All Rema	ining	+
Budius	Sub-I	Family All	Remaining	Ψ	Temp Grade	All Rema	ining	-
				Reset All	Filtore			
				Reset All	Filters			
Search: Q	, 70t			🛛 🛛 (10 ma	tches)			
		I/O Pin	Available	LUT		Block		Gb
Device		Count	IOBs	Elements	FlipFlops	RAMs	DSPs	Trans
xc7k70tfb	q484-1	484	285	41000	82000	135	240	4
xc7k70tfb	xc7k70tfbg484-1 xc7k70tfbg484-2 xc7k70tfbg484-2L xc7k70tfbg484-2L xc7k70tfbg484-3		285	41000	82000	135	240	4
xc7k70tfb	g484-2L	484	285	41000	82000	135	240	4
xc7k70tfb	g484-3	484	285	41000	82000	135	240	4
xc7k70tfb	g676-1	676	300	41000	82000	135	240	8
🔷 xc7k70tfb	9676-2	676	300	41000	82000	135	240	8
xc7k70tfb		676	300	41000	82000	135	240	8
vc7k70tfb	g676-3	676	300	41000	82000	135	240	8
	<u> </u>	484	285	41000	82000	135	240	4
xc7k70tlf	bg676-2L	676	300	41000	82000	135	240	8
xc7k70tlf	111							
xc7k70tlf xc7k70tlf	111							

Figure 4: Selecting a Family and Default Part

- 15. Select the **xc7k70tfbg676-2** device and click **Next**.
- 16. Click **Finish** to create the project.



project_pinout - [C:\14.1_Tuto	orials\PlanAhead_Tutorial\Tutorial_Created_E	ata\proje	t_pino	ut\projec	t_pinout.ppr]	- PlanAhe	ead 14.	1							X
File Edit Flow Tools Wind	low Layout View Help											Q- Sear	ch comma	ands	
💕 😂 🗏 🖍 💷 🗙 🏘 🔮 💈	🐒 🔀 🥵 🖽 I/O Planning 🛛 🔹 🗶 🔖	K 🖏													
Flow Navigator «	I/O Design - xc7k70tfbg676-2														×
🔍 🖾 🖨) 🔠 P	ackage	× 🔷 Device	×								00	×
Project Manager	🛣 🛃 🕅 (empty)					A		56789101	12 13 14 15 16	17 18 19 20 21 22 2		L.			
₄ I/O Design			→ +			C D	FF								
🔄 Migrate to RTL			Q+			E						, I -			
诸 Import I/O Ports			0-			G					tot	3			
Report DRC			٩			4 . F									
Report Noise			: 🔯			K L									
📑 Export I/O Ports	10 Ports SRC kaise 10 Ports 10 P														
			- 😳			PR					+ As 222 ≥ 25 30 - □ □ ×				
			*			т u						1			
						Ŵ									
:	A Sources in Netlist Properties Clock Regions Package Pins Name Prohibit Port J/O Std Dir Vcco Bank Bank Type Diff Pair Clock Voltage Config														
A Sources N Hetlist Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Properties Propertie	18 10														
														- 0 8	×
		Prohibit	Port	I/O Std	Dir	Vcco	Bank	Bank Type	Туре	Diff Pair	Clock	Voltage	Config	XADC	
	I/O Bank 13 (High Bange) (56)														
	I/O Bank 14 (High Range) (56)							High Range							Ξ
								High Range							
	I/O Bank 10 (High Range) (57)	With Clock Regions Image: Clock Regions Image: Clock Re													
	I/O Bank 33 (High Performance) (56			Pan Image I											
	I/O Bank 34 (High Performance) (56							High Perfo							-
		-1		1											
	E Tcl Console Messages Package	Pins 🗋 🗅 I/	O Ports												

Figure 5: I/O Pin Planning Environment

- 17. Explore the various views in the I/O Planning layout. Many are empty because I/O Ports are not yet defined.
- 18. Right-click on either the Device or Package view tabs to select **New Vertical Group**.

Notice that the Device and Package views are now both displayed. 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 PlanAhead tool I/O pin planning environment lets you explore various device resources.

The different views 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 view and I/O Bank Properties view provide some I/O related information typically found in the device data sheets.

Next, you:

- Select several I/O banks to show the package-to-die relationship
- View I/O bank properties
- Select and expand the I/O Bank 14 to view package pin specifications

I/O Pin Planning Tutorial UG674 (v 14.1) May 8, 2012



Examining I/O Banks

1. In the Package Pins view, select an I/O Bank such as **I/O Bank 33**.

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

2. To select a bank in the Package view, double-click any pin from 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 a part of.

e Edit Flow Tools W	indow Layout View Help								Q+ Searc	h comma	nds
	🕺 ∑ 👩 🔚 1/0 Planning	· # * * 9									
Prov Newlogstor I/O Design xc7k/0thp676.2 Project Manager I/O Design xc7k/0thp676.2 Import Vo Ports Import Vo Ports Propert Noise Import Vo Ports Import Vo Ports Import Vo Ports											
Project Manager // O Design @ Migrate to RTL @ Migrate to RTL @ Import 10 Ports @ Report Noise @ Export I/O Ports Wo Bank Properties - U/O Bank 30 (High Performance) Id Mame Prohibit Port 1/O Std Dir Vt											
Yroject Manager Yl/O Design Migrate to RTL Migrate to RTL Migrate to RTL Report Noise Report Noise Propert V/O Ports		- 0 0 ×	2 Package	×		o a ×	Device ×				00
Images I/O Design Images Images <t< td=""><td></td><td></td><td>1</td><td></td><td></td><td></td><td><u>+0</u></td><td></td><td></td><td></td><td></td></t<>			1				<u>+0</u>				
	a) (empty)		*: 200	- Balanta Balanta Ba			4-				
			→ 6 1.568				→				
			A				04			Y3	
			* • •		16 2 (C.		0				
[™] Report I/O Ports [™] Export I/O Ports [™] Monoreties [™] I/O Bank Properties [™] I/O Bank Sources [™] I/O Bank Properties [™] I/O Bank 33 (High Performance)		1									
	- 0 8 ×					R.	1000				
The traction of the					111.3	1001	X				
	# 1/O Bank 33 (High Performance)			Billionality	Torier	0000	22				
							3				
		I/O Std Dir Ve	2 3 3 3 3				**	X0Y1	xt	YI	
		0	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				p.				
		+ 🗆	P _n A second			AD AD	19.			- 6	
	General Port Summary Package	Pins I/O Port 4 + III					1				
							8	D YU		YO	
											- 0 6
		Prohibit	Port 1/0 Std	Dir Vcc	Bank	Bank Type	Type	Diff Pair	Clock \	/oltage	Config
Properties Clock Regions Package Pins	T = I/O Bank 32 (Dedicated)	(6)									
	mance) (56)				High Perfo						
			110	V	RN						
	L2P										
		1									
			1		33						

Figure 6: Cross Highlighting I/Os and I/O Banks

An alternate method is to click the **Package View Layers** button ¹, located in the Package View toolbar. Expand the I/O Banks, select any I/O Bank, right click it and choose **Select Objects**.

The Layer Control can also be used to display specific Multi-Function Pins, such as Vref, adjust the look of the Package view, highlight specific bank types, or hide transceiver banks.

The selected I/O bank location is highlighted in the Package view.

- 3. Expand the selected **I/O Bank** in the Package Pins view to display the package pin information for each pin in the I/O Bank. The internal package trace min and max delays are shown also (scroll the view to the right to see them). These are the routing delays between the pin on the package and the pad on the die.
- 4. Scroll down the list and select any I/O Bank.



- 5. Select the **General** tab in the I/O Bank Properties view.
- 6. 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.
- 7. Select the various tabs in the I/O Bank Properties view.
- 8. Click the **Maximize** button \square in the Package Pins view banner.

The Package Pins view is maximized.

- 9. Select the **Expand All** button 🚔 in the Package Pins view.
- 10. Scroll to the right and view the pin information in the table.
- 11. Unselect the **Group by I/O Bank** button in the Package Pins view 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. In the following sequence, you will sort the Package Pins view by Voltage to select all VREF I/O pins, then use the Set Prohibits popup command to prohibit placement on those pins.

- 1. Click the Voltage column header twice 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 view header, click the Restore button \square .
- 5. The Package Pins view is restored. The Package view now displays prohibited pins.
- 6. In the main toolbar, click Unselect All 📐
- 7. Zoom in to an area of the Package view to view the Prohibited pins marked with red Ss, as shown in the following figure. To zoom, click and drag a rectangle in the Package view starting with a click at the upper left of the zoom area and drag to the lower right zoom area.

E XILINX.

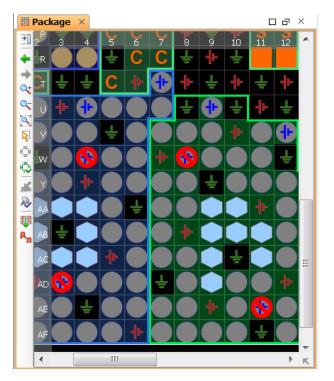


Figure 7: Examining Prohibited VREF Package Pins

8. Zoom Fit the Package view. Click and drag the cursor from the lower right to the upper left in a diagonal motion.

Step 4: Creating and Configuring I/O Ports

In this step you will create and configuring a new I/O bus port called **mybus**.

- 1. Click the **I/O Ports** view tab.
- 2. Right-click in the I/O Ports view and select **Create I/O Ports**.

The Create I/O Ports dialog box opens.

	Create I/O Ports	
	Name: mybus	8
	Direction: Input 👻	
l	Diff Pair:	
	Create Bus: From	0 🖨 To 31 🖨
	I/O Standard:	LVCMOS18 -
	Drive Strength:	default 🔻
	Slew Type:	SLOW (default) 🔻
	Pull Type:	NONE 🔻
	Input Termination Type:	NONE (default)
		OK Cancel

Figure 8: Create I/O Ports

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

- 3. Type **mybus** in the Name field.
- 4. Click the checkbox for **Create Bus**.
- 5. Select the default I/O Standard **LVCMOS18** from the drop down menu.
- 6. Review the other options and click **OK**.

The new I/O Ports display in the I/O Ports view.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip T	IN_TERM
🖃 🗁 All port	s (32)												
🖻 📴 myt	ous (32) Input					LVCMOS18	1.8		12	SLOW	NONE	NONE	
- D- 1	mybus[0] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
	mybus[1] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
- D- 1	mybus[2] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
	mybus[3] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
	mybus[4] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
	mybus[5] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
D- 1	mybus[6] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
- D-	mybus[7] Input					LVCMOS18	1.8			SLOW	NONE	NONE	
- D-1	mybus[8] Input					LVCMOS18	1.8		12	SLOW	NONE	NONE	
	mubuel01 Toput					LVCMOS18	1.8			SLOW	NONE	NONE	

Figure 9: Displaying Newly Added I/O Ports

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

www.xilinx.com



Step 5: Importing an I/O Port List

The PlanAhead tool can import a variety of file formats to begin the I/O pin planning process. You can import CSV, UCF, or RTL format files and perform I/O pin exploration and assignments. You can also create I/O Ports interactively, which was covered in the last step.

Use care with early input methods for I/O pin planning. Without a synthesized netlist, the I/O Ports placement and DRC routines do not take clocks, clock relationships, or GT logic into account 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 ISE implementation tools, and after DRCs for I/O and clock placement are run without error.

Importing and Examining the CSV Format I/O Port List

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

<Extract_Dir>/PlanAhead_Tutorial/Sources/IO_Ports_import.csv

- 2. Examine the I/O ports spreadsheet format and content, and exit without saving.
- 3. In the PlanAhead tool environment, select **Import I/O Ports** from the Flow Navigator, located on the left side.
- 4. Select CSV File and browse to select:

<Extract_Dir>/PlanAhead_Tutorial/Sources/IO_Ports_import.csv

5. Click **OK**.

The Device and Package views display the assigned Ports, and the I/O Ports view is now populated with the imported I/O Ports, as shown in the following figure. If you are going to import a CSV file, do this before defining ports with the Create I/O Ports command because they will be overwritten.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TE	R
- 🗁 All ports (135)														
🕀 📴 DataIn_pad_0_i (8)	Input					LVCMOS18	1.8		12 SLOW			NONE	NONE	
🖶 📴 DataIn_pad_1_i (8)	Input					LVCMOS18	1.8		12	2 SLOW		NONE	NONE	
DataOut_pad_0_o (8)	Output					LVCMOS18	1.8		12 SLOW			FP_VTT_50	NONE	
DataOut_pad_1_o (8)	Output					LVCMOS18	1.8		12 SLOW			FP_VTT_50	NONE	
LineState_pad_0_i (2)	Input					LVCMOS18	1.8		12	2 SLOW		NONE	NONE	
LineState_pad_1_i (2)	Input					LVCMOS18	1.8		12 SLOW			NONE	NONE	
OpMode_pad_0_o (2)	Output					LVCMOS18	1.8	1.8 12 SLOW 1.8 12 SLOW		2 SLOW		FP_VTT_50	NONE	
OpMode_pad_1_o (2)	Output					LVCMOS18	1.8 12 SLOW 1.8 12 SLOW		2 SLOW		FP_VTT_50	NONE		
Image: Second	Output					LVCMOS18	1.8 12 SLOV 1.8 12 SLOV		2 SLOW		FP_VTT_50	NONE		
RXP_IN (16)	Input	RXN_IN				DIFF_HSTL		1.8		SLOW		NONE	NONE	
TXP_OUT (16)	Output	TXN_OUT				DIFF_HSTL	1.8			SLOW		NP_VTT_50_FP_VTT	NONE	
VControl_pad_0_0 (4)	Output					LVCMOS18	1.8		12	2 SLOW		FP_VTT_50	NONE	

Figure 10: I/O Bus Ports are Grouped by Bus

The buses are grouped together and are expandable.



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

You can export the I/O Port assignments to UCF, 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 can be used to begin I/O Port assignments.

Exporting the I/O Ports List Using the Export I/O Ports Command

- 1. Select File > Export > Export I/O Ports.
- 2. Select CSV, UCF, Verilog, and VHDL in the Export I/O Ports dialog box.

Export	I/O Ports
Specify Ty	pes to Generate
CSV	C:\14.1_Tutorials\PlanAhead_Tutorial\Tutorial_Created_Data\project_pinout\io_1.csv
UCF	C:\14.1_Tutorials\PlanAhead_Tutorial\Tutorial_Created_Data\project_pinout\io_1.ucf
Verilo	C:\14.1_Tutorials\PlanAhead_Tutorial\Tutorial_Created_Data\project_pinout\project_pinout.v
	C:\14.1_Tutorials\PlanAhead_Tutorial\Tutorial_Created_Data\project_pinout\project_pinout.vhd 💿 🛄
	OK Cancel

Figure 11: Exporting I/O Ports to a CSV Spreadsheet and UCF File

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

< Extract_Dir>/PlanAhead_Tutorial/Tutorial_Created_Data/project_pinout

If defined, the Interface group names are included in the spreadsheet. Printed circuit board designers can use this spreadsheet to create Interface-specific schematic symbols. Creating I/O Port Interfaces is covered in an upcoming step.

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 additional Sources to be added and a path through to implementation. The I/O Port assignments and names will be translated into an RTL header and UCF Source files in an RTL Project.

Note: Once you migrate an I/O Planning Project to an RTL Project, you cannot go back to the original I/O Project.

1. Select **Migrate to RTL** in the Flow Navigator.

The Migrate to RTL dialog box appears.

I/O Pin Planning Tutorial UG674 (v 14.1) May 8, 2012



Migrate To RTL	
Migrate the current Pin Planning Project into an RTL Project. The conversion cannot be undone.	
Top RTL file inAhead_Tutorial\Tutorial_Created_Data\project_pinout\io_1.v Netlist format Verilog	
Write diff buffers 🔽	
OK Cancel	

Figure 12: Migrating an I/O Planning Project to an RTL Project

- 2. Click **OK** to accept the default file names and locations.
- 3. Notice the RTL Project is now displayed. The Sources view contains the newly created source files.

Closing the Newly Created RTL Project

- 4. Select File > Close Project.
- 5. Click **OK** in the **Confirm Close Project** dialog box.

Part 2: I/O Planning Features using a Synthesized Design

Many of the features presented in this tutorial are available in several places during the design flow. As Part 1 describes, an I/O Planning Project can begin the I/O assignment process well before any RTL or synthesized netlist is available. In RTL Projects, I/O Planning can also be performed pre-synthesis by opening the elaborated RTL design. The most comprehensive set of features and DRCs are available after synthesis by opening the synthesized design.

The I/O Planning features provide several ways to analyze, group and place the I/O ports onto package pins or the I/O die pads.

For control over I/O port placement, you can drag the selected I/O Ports into the Package or Device views interactively using one of the following semi-automatic placement modes:

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

In addition, the I/O Planning features enables you to toggle DRCs on and off during I/O placement.

Step 8: Opening the Synthesized Netlist-Based Project

- 1. Click the **Open Project** link in the Getting Started view, or select **File > Open Project**.
- 2. Browse to select the following project file:

<Extract_Dir>/PlanAhead_Tutorial/Projects/project_cpu_netlist/project_c pu_netlist.ppr

Alternately, select **Open Example Project > CPU (Synthesized)** from the Getting Started page. If needed, use **File > Save Project As...** to save a local writeable copy of the project at <*Extract_Dir*>/PlanAhead_Tutorial/Tutorial_Created_Data

- 3. In the Sources view, expand the Constraints **constr_1** folder and double-click on the **top.ucf** file.
- 4. Notice this UCF file only contains timing constraints and no I/Os yet.
- 5. Close the top.ucf file.
- 6. Select the **constr_1** folder and right-click to select **Make Active**.
- Select **Open Synthesized Design** in the Flow Navigator to open the synthesized design.
 Alternately, you can select **Flow > Open Synthesized Design** from the main menu.
- 8. From the Layout pull-down located in the main toolbar, select **I/O Planning**.
- 9. The I/O Planning view layout displays.



Step 9: Examining the I/O Ports in the Design

- 10. In the I/O Ports view banner, click the **Maximize View** button \Box .
- 11. Click **Expand All** 🚔 in the I/O Ports view.
- 12. In the I/O Ports view, click to unselect **Group by Interface and Bus** 🖹
- 13. Scroll down the list of buses and signals.

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 Terminati	ог
	► 43	or1200_pic_ints	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE	
0	44	or1200_pm_out[0]	Output						default (LVCMOS18)	1.8		12	SLOW	NONE	FP_VTT_50	
-0	45	or1200_pm_out[1]	Output						default (LVCMOS18)	1.8		12	SLOW	NONE	FP_VTT_50	
R	46	or1200_pm_out[2]	Output						default (LVCMOS18)	1.8		12	SLOW	NONE	FP_VTT_50	
-0	47	or1200_pm_out[3]	Output						default (LVCMOS18)	1.8		12	SLOW	NONE	FP_VTT_50	
0	48	phy_rst_pad_0_o	Output						default (LVCMOS18)	1.8		12	SLOW	NONE	FP_VTT_50	
-0	49	phy_rst_pad_1_o	Output						default (LVCMOS18)	1.8		12	SLOW	NONE	FP_VTT_50	
	E 50	l reset	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE	
	F 51	RxActive_pad_0_i	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE	
	E 52	RxActive_pad_1_i	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE	
	E 53	RxError_pad_0_i	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE	
	► 54	RxError_pad_1_i	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE	
	F 55	RXP_IN[0]	Input		RXN_IN[0]											
	► 56	RXP_IN[1]	Input		RXN_IN[1]											
	F 57	RXP_IN[2]	Input		RXN_IN[2]											
	F 58	RXP_IN[3]	Input		RXN_IN[3]											
	► 59	RXP_IN[4]	Input		RXN_IN[4]											

[📃] Tcl Console 🗋 🗩 Messages 🛛 💐 Compilation 🖉 🔓 Reports 🖉 🕸 Design Runs 💭 Package Pins 💭 I/O Ports

Figure 13: Examining I/O Standard and Diff Pair Requirements

The Neg Diff Pair fields are populated for some of the buses indicating that they are differential pair buses.

Step 10: Configuring I/Os and Setting I/O Standards

PlanAhead can be used 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.

Note: For 7-Series devices, all I/O Ports must have explicit values for the IOSTANDARD constraint in order to generate a bitstream file. The word "default" is displayed in red to indicate that these values must be applied manually. The reason for this is that 7-Series devices have low and high voltage I/O Banks and extra care must be applied when assigning I/O standards.

Setting the I/O Standard Constraints

- 1. Click on the **Neg Diff Pair** column header to sort by diff pair port type.
- 2. Scroll to the top of the list and you should see no Neg Diff Pair ports, select the first port.
- 3. Use the Shift key and scroll to select all I/O Ports that are <u>not</u> diff pair ports.

	Id	Name		Interface	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination
i D	82	VStatus_pad_0_i[4]	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE
I D	83	VStatus_pad_0_i[5]	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE
	84	VStatus_pad_0_i[6]	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE
	85	VStatus_pad_0_i[7]	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE
P D-	86	VStatus_pad_1_i[0]	Input						default (LVCMOS18)	1.8		12	SLOW	NONE	NONE
	87	VStatus_pad_1_i[1]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
		VStatus_pad_1_i[2]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
	89	VStatus_pad_1_i[3]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
D	90	VStatus_pad_1_i[4]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
D	91	VStatus_pad_1_i[5]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
Þ	92	VStatus_pad_1_i[6]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
Þ	93	VStatus_pad_1_i[7]	Input						default (LVCMO518)	1.8		12	SLOW	NONE	NONE
		XcvSelect_pad_0_o	Output						default (LVCMO518)	1.8		12	SLOW	NONE	FP_VTT_50
	- 95	XcvSelect_pad_1_o	Output						default (LVCMO518)	1.8		12	SLOW	NONE	FP_VTT_50
	96	RXP_IN[0]	Input		RXN_IN[0]										
		RXP_IN[1]	Input		RXN_IN[1]										
D .	08	DVD TNE21	Toput		DVN TNF21										

📃 🔚 Tcl Console 🗋 🗭 Messages 🛛 🗟 Compilation 🖉 Reports 🖉 Design Runs 💭 Package Pins 🕞 🗗 I/O Ports

Figure 14: All Single Ended Ports Selected

4. Right-click and select **Configure I/O Ports**.

Notice the various drop-down menus to set I/O Configuration constraints. The I/O standard field displays the default value.

5. Click in the **I/O Standard** drop-down menu to scroll to select **LVCMOS18** and click **OK**.

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. Once these Ports are placed on GT pins, these fields will be empty.

Step 11: Creating I/O Port Interfaces

It can be beneficial to group I/O Ports associated with various I/O interfaces. The I/O Planning layout lets you define groups of pins, buses or other interfaces together as an "Interface." This ability helps with I/O Port management and with generating interface-specific PCB schematic symbols. It also forces the I/O Port automatic placement command to group the entire interface together on the device (where possible).

Creating Interfaces for Similar I/O Port Groups

The design used in this tutorial has two USB interfaces, each containing many I/O ports. The I/O port names are differentiated by $_0$ and $_1$. You will create Interfaces for all signals in USB0 and USB1.

- 1. Click the **Show Search** button $\stackrel{\frown}{\longrightarrow}$ in the I/O Ports view.
- 2. Type **_0**_ in the Search field.

Se	arch:	Q*_0_		(43 match
	Id	Name	Direction	Interface
	1	DataIn_pad_0_i[0]	Input	
	2	DataIn_pad_0_i[1]	Input	
	3	DataIn_pad_0_i[2]	Input	
	4	DataIn_pad_0_i[3]	Input	
	5	DataIn_pad_0_i[4]	Input	
	6	DataIn_pad_0_i[5]	Input	
	7	DataIn_pad_0_i[6]	Input	
	8	DataIn_pad_0_i[7]	Input	
-77	9	DataOut_pad_0_o[0]	Output	
-27	10	DataOut_pad_0_o[1]	Output	
-77	11	DataOut_pad_0_o[2]	Output	
-27	12	DataOut_pad_0_o[3]	Output	
-	13	DataOut nad 0 o[4]	Output	

Figure 15: Selecting 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.
- Right-click and select Create I/O Port Interface....
 The Create I/O Port Interface dialog box opens.
- 6. Type **USB0** in the Name field.



Figure 16: Create I/O Port Interface

- 7. Click **OK**.
- 8. In the Search field, change **_0** to **_1** and follow the same steps to create a USB1 I/O Port Interface.
- 9. Click the **Show Search** button \bigcirc to remove the Search filter.
- 10. Click the **Group by Interface and Bus** and the **Collapse All** to buttons.

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



- 11. Expand the **Scalar ports** folder to view the clocks resets and other ports.
- 12. Click the **Restore** button $\stackrel{\square}{=}$ in the view banner.

The I/O Ports view is restored to the original location, shown in the following figure.

	Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip T	IN_TERM	
	🖃 🗁 All ports (135)														
	🖻 🔂 USBO (43)	(Multiple)					LVCMOS18	1.8		1:	2 SLOW	NONE	(Multiple)		
	🗄 🔂 USB1 (43)	(Multiple)					LVCMO518	1.8		1	2 SLOW	NONE	(Multiple)		
	🗄 🖓 or1200_pm	Output					LVCMOS18	1.8		1:	2 SLOW	NONE	FP_VTT_50		
	😟 🔂 RXP_IN (16)	Input	RXN_IN												
	😟 🖓 TXP_OUT (16)	Output	TXN_OUT												
	🖻 🗁 Scalar ports (13)													
	- D GTPRESE	Input					LVCMOS18	1.8		1:	2 SLOW	NONE	NONE		
1	- D or1200_c	Input					LVCMO518	1.8		1:	2 SLOW	NONE	NONE		
I	- D or1200_p	Input					LVCMO518	1.8		1:	2 SLOW	NONE	NONE		
I	- D reset	Input					LVCMOS18	1.8		1:	2 SLOW	NONE	NONE		
1	- D sysClk	Input					LVCMOS18	1.8		1:	2 SLOW	NONE	NONE		
1	TILEO_RE	Input	TILE0_REFCL												
I	- D TILE1_RE	Input	TILE1_REFCL												
I	TILE2_RE	Input	TILE2_REFCL												
1	TILE3_RE	Input	TILE3_REFCL												

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

Step 12: Viewing Multi-function Package Pins

Some Xilinx devices have a set of package pins that can be used for a variety of purposes depending on the design configuration. These are referred to as Multi-function pins. The mode in which you intend to configure the device or the use of memory controllers or a PCI interface can require use of some of these pins. The Package Pins view can be examined to ensure that no conflicts exist. In this step you will view the package pins data and multi-functional pins.

- 1. Click the **Package Pins** view tab.
- 2. In the Package Pins view banner, click the **Maximize View** button \Box .
- 3. Click **Expand All** 🚔 in the Package Pins view
- 4. 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 view is dynamically updated as I/O Ports are placed in the design.
- 5. In the Package Pins view, click to unselect **Group by I/O Bank** 🖹

The Package Pins now display as a flat list rather than grouped by I/O Bank.

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

	Id	Name	Prohibit	Port	I/O Std	Dir	Vcco	Bank	Bank Type	Type 1	Diff Pair	Clock	Voltage	Confic
1		VVZZ	TTOTIDIC	TOIL	40.50	Uli	veco	DUIK	NONE	GNU	Diri Taii	CIUCK	voltage	conng
ã									NONE	GND				
a	188								NONE	GND				
ã		P25						13	High Range	Multi-function	L6N		VREF	
ã		P23						13		Multi-function		SRCC	VINEI	
ã		N23						13	High Range	Multi-function		SRCC		
ã		N21						13		Multi-function		MRCC		
ã		N22						13	High Range	Multi-function		MRCC		
a		R21						13	High Range	Multi-function		MRCC		
ã		P21						13	High Range	Multi-function		MRCC		
ã		R22						13	High Range	Multi-function		SRCC		
a		R23						13	High Range	Multi-function		SRCC		
a		T19						13		Multi-function		Sillee	VREF	
a		B24						14	High Range	Multi-function			VINE!	
a		A25						14	High Range	Multi-function				
a		B22						14	High Range	Multi-function				
a		A22						14	High Range	Multi-function				
a		B25						14	High Range	Multi-function				PUD
a		B26						14	High Range	Multi-function				
à		A23						14	High Range	Multi-function				
a		A24						14	High Range	Multi-function				
a		D26						14	High Range	Multi-function				
Ĩ	_									111				

Figure 18: Viewing Multi-Function Pins

- 8. Examine the following columns:
 - Device Configuration pins (Config)
 - XADC
 - Gigabit I/O

These logic objects can impact I/O assignment because many of them rely on multifunction 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.

Restore the Package Pins view.

9. In the Package Pins view, click the **Group by I/O Bank** button 🔧.

10. Click **Collapse All** $\stackrel{\scriptstyle{\scriptstyle{\sim}}}{=}$ to return the tree table display to the default display structure.

Note: The PlanAhead tool has several tree table style views. There are search and filtering capabilities available in these views. See "Using Tree Table Style Views" in the *Using the Viewing Environment* chapter of the *PlanAhead User Guide (UG632)* for more information.

www.xilinx.com



Step 13: Setting Device Configuration Modes

In the PlanAhead tool you can set one or more device configuration options. Some configuration modes can have an impact on multi-function I/O pins also. The related pins display this information in the **Config** column of the Package Pins view.

1. Select **Tools > I/O Planning > Set Configuration Modes...**. The Set Configuration Modes dialog box opens.

Master SPI x1	JTAG/Boundary Scan
Master SPI X1 Master SPI X2 Master SPI X4 Master Serial Slave Serial Master BPI-Up X8 Master BPI-Up X16 Slave SelectMap X8 Slave SelectMap X16 Slave SelectMap X32 JTAG/Boundary Scan Master SelectMap X8 Master SelectMap X16	The JTAG configuration mode allows the device to be configured through the JTAG port. This is often used with board testers, for debug, and for indirectly programming SPI flash or parallel NOR flash connected to the FPGA configuration port. The Xilinx download cables and the ChipScope Pro tool use the JTAG configuration mode. Supporting this mode for debug is recommended even if another mode will be used by the production design. For more detail on this configuration mode, see the <u>Kintex-7 Configuration User Guide</u> . JTAG Header JTAG Header TDO TDI TMS TCK Userte_et_re_et_reter to the State of the set of the s
Select All Clear All	Print

Figure 19: Selecting Device Configuration Modes

2. In the **Set Configuration Modes** dialog box, select one or two of the other modes to view the descriptions, schematics, and related data sheets.

3. Leave it set on JTAG/Boundary Scan and click Cancel.

Setting a Configuration Mode results in the pins associated with it to be displayed at the top of the Package Pins view allowing you to examine potential multi-function pin conflicts.



Step 14: 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 PlanAhead tool 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 step ensures that the I/O pinouts work across the selected set of devices.

1. Select Tools > I/O Planning > Set Part Compatibility....

The Set Part Compatibility dialog box opens.

Set Part Compatibil	ity 🗾 📉
	ve PROHIBIT constraints to n that the device is compatible parts.
Compatible Parts	
🔲 🔷 xc7k160tfbg676	
📃 xc7k160tffg676	
📃 🔷 xc7k325tfbg676	
xc7k325tffg676	
📃 🔷 xc7k410tfbg676	
📃 🧼 xc7k410tffg676	
	OK Cancel

Figure 20: Defining Compatible Parts

- 2. Select the top **xc7k160tfbg676** device.
- 3. Click **OK**.
- 4. In the confirmation dialog box, click **OK** to indicate that no Prohibits were placed.

The 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 15: Placing I/O Ports

The PlanAhead tool provides several ways to place the I/O Ports onto either package pins or I/O die pads. The automatic placement command tries to place the entire selected group of I/O Ports, adhering to I/O bank rules while grouping buses and Interfaces together.

By default, the PlanAhead tool uses interactive Design Rule Checks (DRCs) during I/O placement.



For more control over I/O port placement, you can drag the selected I/O Ports into the Package or Device views using one of the following semi-automatic placement modes:

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

Placing the USBO Port Interface using Place Ports in an I/O Bank

- 1. In the I/O Ports view, select the **USB0 Interface**.
- 2. In the Package view, click to expand the **Place Ports** button

3. Select Place I/O Ports in an I/O Bank.

As you drag the cursor over the Package Pins, the assignment pattern displays the number of pins to be placed shows in the tooltip.

The Information bar at the bottom of the PlanAhead tool displays information about the objects being dragged over, including I/O Banks and Package Pins.

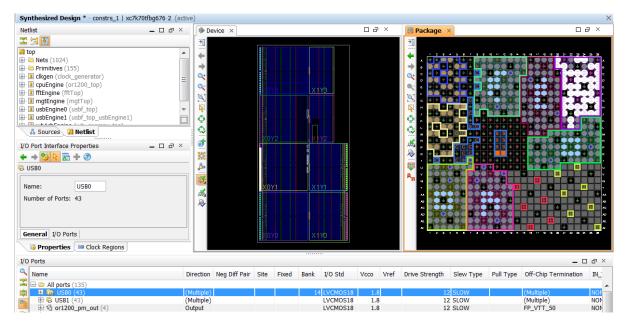


Figure 21: Place I/O Ports in an I/O Bank

4. Click **I/O Bank 14** on the top, right side of the package to drop the I/O Ports.

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



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

- 5. In the Device view, zoom in to the upper half of the device.
- 6. In the I/O Ports view, select the **USB1 Interface**.
- 7. In the Device view, click to expand the **Place Ports** button <u></u>
- 8. Select Place I/O Ports in Area.

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

9. 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.

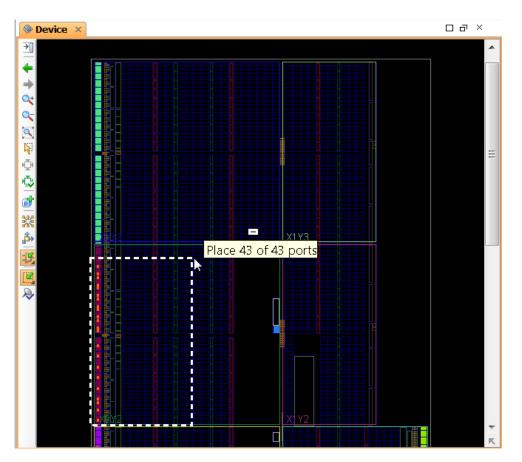


Figure 22: Placing USB1 I/O Ports in an Area

Placing the RXP_IN Differential Pair Bus

- 10. Select the **RXP_IN** bus in the I/O Ports view.
- 11. In the Device view, click to expand the **Place Ports** button $\overset{(1)}{\Longrightarrow}$.

I/O Pin Planning Tutorial UG674 (v 14.1) May 8, 2012 www.xilinx.com



12. Select Place I/O Ports Sequentially.

13. Drag and click to place the first diff pair I/O Port into one of the GT I/O Banks on a designated pin. (Hint: the GT banks are the two upper left ones.)

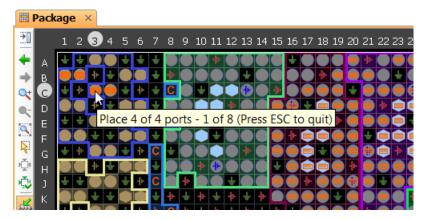


Figure 23: Placing Diff Pair I/O Bus Ports Sequentially

Both diff pairs associated with the GTs were placed on legal sites. You might see a tool tip indicating that the selected site is not legal and stating why it is not legal.

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

After placing the first set of GT diff pair pins, PlanAhead queues up the next group of pins to place.

14. Place all 8 RXP_IN GT Port groups in the two upper left I/O Banks. The cursor Tooltips will help guide you to legal pin selections to place the I/O Banks.

Note: GT logic objects are automatically grouped by PlanAhead 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.

Removing the Split Workspace View for the Device and Package Views

Now that the I/O Ports are placed, the Package view is no longer needed to share the Workspace view. The split view can be removed easily. To remove the split view:

- 15. Right-click on the Package or Device view tab to select Move to Previous Tab Group.
- 16. Select the Device view tab to bring it to the front.
- 17. Adjust the view size, and Zoom fit the view, if needed.



Step 16: Placing Clock Logic

The PlanAhead tool lets you place critical clock or I/O related logic. After a synthesized netlist is imported, clocks and clock relationships can be explored and used to lock down these logic objects onto specific device sites. The PlanAhead tool 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.

Searching for the Global Clock Logic in the Design

1. Click the Find button do r select **Edit > Find**.

The Find dialog box opens.

2. Adjust the selection filters to match the following figure (Type is Clock).

Find	12			×
Find Instances	•			
Туре	▼ is	✓ Clock	- +	
				Match Case
📝 Open in a new tab			ОК	Cancel
			ОК	Cancel

Figure 24: Using Find to Search for Clock Logic

3. Click **OK**.

The Find Results view opens.



iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		Id	Name	Cell	Instance Pin
Image: 2clkgen/clkf_bufBUFG2Image: 2clkin1_bufIBUFG2Image: 2clkgen/clkout1_bufBUFG2Image: 2clkgen/clkout2_bufBUFG2Image: 2clkgen/clkout3_bufBUFG2Image: 2clkgen/clkout3_bufBUFG2Image: 2clkgen/clkout4_bufBUFG2Image: 2clkgen/clkout5_bufBUFG2Image: 3clkgen/clkout6_bufBUFG2Image: 3clkgen/mmcm_adv_instMMCME2_ADV 69Image: 311mgtEngine/gt_usrclk_source/txoutclk_bufg0_i BUFG2	i	1	mgtEngine/gt_usrclk_source/bufg_inst	BUFG	2
ii 4 clkgen/clkout1_buf BUFG 2 ii 5 clkgen/clkout2_buf BUFG 2 ii 6 clkgen/clkout3_buf BUFG 2 ii 7 clkgen/clkout4_buf BUFG 2 ii 8 clkgen/clkout5_buf BUFG 2 ii 9 clkgen/clkout6_buf BUFG 2 ii 10 clkgen/mmcm_adv_inst MMCME2_ADV 69 ii 11 mgtEngine/gt_usrclk_source/txoutclk_bufg0_i BUFG 2	i	2		BUFG	2
Image: Constraint of the structure Source Source<	i	3	clkin1_buf	IBUFG	2
10 6 clkgen/clkout3_buf BUFG 2 10 7 clkgen/clkout4_buf BUFG 2 11 8 clkgen/clkout5_buf BUFG 2 11 mgtEngine/gt_usrclk_source/txoutclk_bufg0_i BUFG 2	i	4	clkgen/clkout1_buf	BUFG	2
Image: 7 clkgen/clkout4_buf BUFG 2 Image: 8 clkgen/clkout5_buf BUFG 2 Image: 9 clkgen/clkout6_buf BUFG 2 Image: 10 clkgen/mmcm_adv_inst MMCME2_ADV 69 Image: 11 mgtEngine/gt_usrclk_source/txoutclk_bufg0_i BUFG 2	i	5	clkgen/clkout2_buf	BUFG	2
¹ ⁸ ¹ ¹ ⁸ ¹	1	6	clkgen/clkout3_buf	BUFG	2
Image: 0.1 gradient of the state of the	i	7	clkgen/clkout4_buf	BUFG	2
Image: 10 clkgen/mmcm_adv_inst MMCME2_ADV 69 Image: 11 mgtEngine/gt_usrclk_source/txoutclk_bufg0_i BUFG 2	i	8	clkgen/clkout5_buf	BUFG	2
11 mgtEngine/gt_usrclk_source/txoutclk_bufg0_i BUFG 2	i	9	clkgen/clkout6_buf	BUFG	2
	i	10	clkgen/mmcm_adv_inst	MMCME2_ADV	69
12 matEnsing/at uscalle sources/troutable hufat i DUEC 2	i	11	mgtEngine/gt_usrclk_source/txoutclk_bufg0	_i BUFG	2
		10	matEnsing (at usralle source /broutable bufat	; DUEC	2

Figure 25: Viewing the Clock Objects Found

- 4. Scroll down the list of objects, and observe the following:
 - BUFG
 - MMCME2_ADV

Step 17: Using the Schematic to Trace Clock Logic

The Schematic view can be used to expand and explore any logic in the design. Placement constraints can be applied from the Schematic view.

1. In the Find Result view, select the first **MMCME2_ADV** Cell.

	Id	Name	Cell	Instance Pin
	2	clkgen/clkf_buf	BUFG	2
1	3	clkin1_buf	IBUFG	2
i	4	clkgen/clkout1_buf	BUFG	2
i	5	clkgen/clkout2_buf	BUFG	2
i	6	clkgen/clkout3_buf	BUFG	2
i	7	clkgen/clkout4_buf	BUFG	2
i	8	clkgen/clkout5_buf	BUFG	2
i	9	clkgen/clkout6_buf	BUFG	2
i	10	clkgen/mmcm_adv_inst	MMCME2_ADV	69
i	11	mgtEngine/gt_usrdk_source/txoutdk_bufg0_i	BUFG	2
	17	matEngine/at_usrdk_source/tyoutdk_bufa1_i	RUEG	2
1	Inst	ances - Type is 'Clock' (14) $~ imes$		

Figure 26: Selecting Clock Logic to Trace in the Schematic

2. In the Find Results view, click the **Schematic** button \square .

I/O Pin Planning Tutorial UG674 (v 14.1) May 8, 2012



- 3. In the Schematic view, double-click the **CLKIN1** input port on the upper left of MCM module
- 4. Double-click to expand the **CLKFBIN** input pin on the MCM module.
- 5. Double-click on the 5 MCM output pins **CLKOUT1-5** to expand the BUFGs.
- 6. Clean up the connections by clicking the **Regenerate Schematic** button in the Schematic view

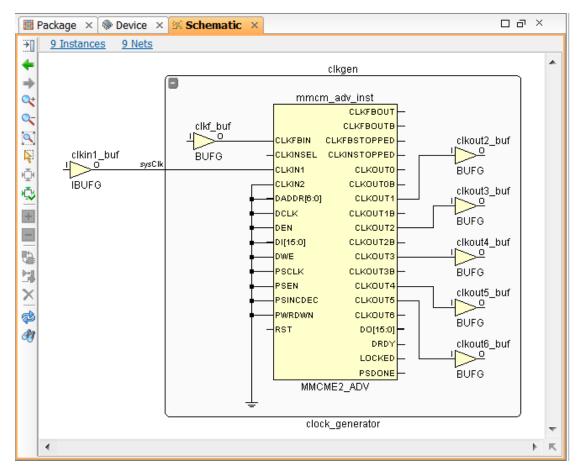


Figure 27: Exploring Clock Logic Connectivity

You can easily expand and explore logic in the Schematic view. Select or highlight logic in the Schematic view to cross-select or highlights it in all other views.

You can also drag logic directly out of the Schematic for placement into the Device, Package or Clock Resources view.

7. Close the Schematic view tab.



Step 18: Exploring the Clock Resources View

- 1. In the Workspace, select the **Clock Resources** view tab.
- 2. If no view tab exists, select **Window > Clock Resources** to display the view.
- 3. Click the **Maximize Workspace** button \Box in the view banner to display the view full screen.
- 4. Scroll around and examine the Clock Resources view.

xc7k70tfbg676-2					
Top Half					
Clock Region X0Y3		☐ Global Buffers @ Y=3	📔 📼 <u>Clock Region X1Y3</u>		
🗏 🔊 I/O Bank 16 (High	Range)		∃ I/O Bank 36 (Unbonded)	GT Bank 116	
Site	Instance			Site	Instance
BUFR X0Y15				GTXE2 CHANNEL X0Y7	gtxe2_i (GTXE2_CHANNEL
BUFR_X0Y14				GTXE2_CHANNEL_X0Y6	gtxe2_i (GTXE2_CHANNEL
BUFR_X0Y13				GTXE2_COMMON_X0Y1	
BUFR_X0Y12				GTXE2_CHANNEL_X0Y5	gtxe2_i (GTXE2_CHANNEL)
BUFIO_X0Y15				GTXE2_CHANNEL_X0Y4	gtxe2_i (GTXE2_CHANNEL)
BUFIO_X0Y14				A4:XTXP3	TXP_OUT_4_OBUF (OBUF)
BUFIO_X0Y13				B6:XRXP3	RXP_IN_4_IBUF (IBUF)
BUFIO_X0Y12				A3:XTXN3	TXN_OUT_4_OBUF (OBUF)
IDELAYCTRL_X0Y3				B5:XRXN3	RXN_IN_4_IBUF (IBUF)
BUFMRCE_X0Y6				B2:XTXP2	TXP_OUT_3_OBUF (OBUF)
BUFMRCE_X0Y7				C4:XRXP2	RXP_IN_3_IBUF (IBUF)
PLLE2_ADV_X0Y3				B1:XTXN2	TXN_OUT_3_OBUF (OBUF)
MMCME2_ADV_X0Y3				D6:REFCLK0P	
G11:SRCC				C3:XRXN2	RXN_IN_3_IBUF (IBUF)
F10:SRCC				D5:REFCLK0N	
E10:MRCC				S F5:REFCLK1N	
D10:MRCC				F6:REFCLK1P	
C12:MRCC				D2:XTXP1	TXP_OUT_0_OBUF (OBUF)
C11:MRCC				E4:XRXP1	RXP_IN_0_IBUF (IBUF)
E11:SRCC				D1:XTXN1	TXN_OUT_0_OBUF (OBUF)
D11:SRCC				E3:XRXN1	RXN_IN_0_IBUF (IBUF)
				F2:XTXP0	TXP_OUT_1_OBUF (OBUF)
				G4:XRXP0	RXP_IN_1_IBUF (IBUF)
				F1:XTXN0	TXN_OUT_1_OBUF (OBUF)
				G3:XRXN0	RXN_IN_1_IBUF (IBUF)

Figure 28: Viewing Clock Resources in the Clock Resources View

The Clock Regions, I/O Banks, and various device resources display in their relative location as found on the device.

You can expand or collapse sections of the Clock Resources to hide or display the resources as needed. Logic that is placed is displayed under the Instance columns.

Placed clock resources such the GTXE2s and their associated Diff Pair ports are displayed in the chart along with BUFGs and MMCMs

Step 19: Placing the MCM Instance

- 1. Click the **Find Results** view tab on the bottom of the screen to display the view.
- 2. Select the MMCME2_ADV instance and drag it into the Clock Resources view on the Instance field next to one of the MMCME2_ADV Sites.

Note: If the instance does not drag out of the Find Results view, click the Netlist view tab and drag to place it from there.



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

Netlist 🗕 🗆	ъ×	🔠 Package 🛛 🛞 Devi	ice 🗙 🚟 Clock Resources 🗙
🔀 🔄 🗧		S BUFR_X0Y10	
XST_GND (GND)			
	-		
Clock_generator)		BUFIO X0Y11	
⊞		BUFIO_X0Y10	
Primitives (10)		BUFIO X0Y9	
clkf_buf (BUFG)		BUFIO_X0Y9	
		. IDELAYCTRL_X	01/2
		BUFMRCE_X0Y	
		BUFMRCE_X0Y	
clkout5 buf (BUFG)		PLLE2_ADV_X0	
		MMCME2_ADV	_X0Y2 III mmcm_adv_inst (MMCME2 III VStatus_pad_tvi_4_IBUF (IB
mmcm_adv_inst (MMCME2_ADV)		G17:SRCC	
XST_GND (GND)		F18:SRCC	VStatus_pad_1_i_3_IBUF (IB
XST_VCC (VCC)		F17:MRCC	VStatus_pad_1_i_2_IBUF (IB
E couEngine (or1200 top)	-	E17:MRCC	VStatus_pad_1_i_1_IBUF (IB
		E18:MRCC	VStatus_pad_1_i_0_IBUF (IB
		D18:MRCC	VControl_pad_1_o_3_OBUF
🖧 Sources 🕅 Netlist		•	III
Find Results - Instances - Type is 'Clock' (14)			
Find Results - Instances - Type is 'Clock' (14)	Cell	Instance Pins	
Find Results - Instances - Type is 'Clock' (14)	Cell	Instance Pins	
Find Results - Instances - Type is 'Clock' (14)			
Find Results - Instances - Type is 'Clock' (14) Id Name Id	BUFG	Instance Pins 2	
Find Results - Instances - Type is 'Clock' (14) Id	BUFG BUFG BUFG	Instance Pins 2 2 2 2	
Id Name Id Name Id Cikgen/cikout1_but Id Cikgen/cikout1_but Id Cikgen/cikout1_but Id Cikgen/cikout1_but Id Cikgen/cikout1_but Id Cikgen/cikout1_but Id Cikgen/cikout2_but Id Cikgen/cikout3_but Id Cikgen/cikout4_but	BUFG BUFG BUFG BUFG	Instance Pins 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id Cikgen/cikout1_but Cikgen/cikout2_buf Cikgen/cikout3_buf Cikgen/cikout4_buf Cikgen/cikout5_buf Cikgen/cikout5_buf Cikgen/cikout5_buf	BUFG BUFG BUFG BUFG BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id Cikgen/cikout1_but I cikgen/cikout2_buf I cikgen/cikout3_buf I cikgen/cikout4_buf I s cikgen/cikout5_buf I s cikgen/cikout6_buf I s cikgen/cikout6_buf	BUFG BUFG BUFG BUFG BUFG BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name A Cikgen/cikout1_but C Cikgen/cikout2_buf C Cikgen/cikout3_buf S Cikgen/cikout5_buf S Cikgen/cikout5_buf C Cikgen/cikout6_buf C Cikgen	BUFG BUFG BUFG BUFG BUFG MMCM	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id Name Id 4 clkgen/clkout1_but Id 5 clkgen/clkout2_buf Id 6 clkgen/clkout3_buf Id 7 clkgen/clkout5_buf Id 9 clkgen/clkout6_buf Id 10 clkgen/mmcm_adv_inst Id 11	BUFG BUFG BUFG BUFG BUFG BUFG i BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id Name Id 4 clkgen/clkout1_but Id 5 clkgen/clkout2_buf Id 6 clkgen/clkout3_buf Id 7 clkgen/clkout4_buf Id 8 clkgen/clkout5_buf Id 9 clkgen/clkout5_buf Id 10 clkgen/mmcm_adv_inst Id 11 mgtEngine/gt_usrclk_source/txoutclk_bufg0 Id 12 mgtEngine/gt_usrclk_source/txoutclk_bufg1	BUFG BUFG BUFG BUFG BUFG BUFG _i BUFG _i BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id Variable	BUFG BUFG BUFG BUFG BUFG BUFG i BUFG i BUFG i BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id A cikgen/cikout1_but Id 5 cikgen/cikout2_buf Id 6 cikgen/cikout3_buf Id 7 cikgen/cikout5_buf Id 8 cikgen/cikout5_buf Id 9 cikgen/cikout5_buf Id 10 cikgen/mmcm_adv_inst I1 mgtEngine/gt_usrclk_source/txoutclk_bufg1 I1 mgtEngine/gt_usrclk_source/txoutclk_bufg2 I1 mgtEngine/gt_usrclk_source/txoutclk_bufg1 I1 mgtEngine/gt_usrclk_source/txoutclk_bufg2 I1 mgtEngine/gt_usrclk_source/txoutclk_bufg2	BUFG BUFG BUFG BUFG BUFG BUFG i BUFG i BUFG i BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id Variable	BUFG BUFG BUFG BUFG BUFG BUFG i BUFG i BUFG i BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Find Results - Instances - Type is 'Clock' (14) Id Name Id A cikgen/clkout1_but Id 5 clkgen/clkout2_buf Id 6 clkgen/clkout3_buf Id 7 clkgen/clkout4_buf Id 8 clkgen/clkout5_buf Id 9 clkgen/clkout6_buf Id 10 clkgen/mmcm_adv_inst Id 11 mgtEngine/gt_usrclk_source/txoutclk_bufg0 Id 13 mgtEngine/gt_usrclk_source/txoutclk_bufg2 Id 14 mgtEngine/gt_usrclk_source/txoutclk_bufg3 Instances - Type is 'Clock' (14) ×	BUFG BUFG BUFG BUFG BUFG i BUFG i BUFG i BUFG i BUFG	Instance Pins 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	

Figure 29: Displaying Placed MMCM_ADV Logic Instance

- 3. Close the Find Results view.
- 4. In the Clock Resources view, click the **Restore Workspace** button [□] in the view banner to return the view layout.
- 5. Click the **Device** view tab in the Workspace.

Step 20: Running Design Rule Checks - "DRC"

The PlanAhead tool has an extensive set of I/O related DRCs to be sure that I/O Ports are assigned accordingly. You can explore and resolve any violations interactively.



Reporting the I/O Related DRC Checks

- 1. Click **Report DRC** in the Flow Navigator.
- 2. Deselect the **Placer, Netlist**, **Floorplan**, **DSP48**, and **RAMB** rule categories.

C Run DRC
Results Name: drc_1 Output File: Rules to Check: 38 of 1220 Image: Second Stress Stres
Select All Clear All
Open in a new tab OK Cancel
UK Cancer

Figure 30: Run I/O Related DRCs

- 3. Expand the selected rules to examine the rule types.
- 4. Click **OK**.

The DRC violations view now opens.

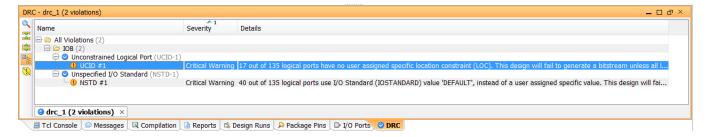


Figure 31: Reporting DRC Violations

The Violations indicate that I have several I/O Ports missing LOC and IOSTANDARD constraints. The design will fail to create a bitstream file as is. We could use the techniques describe in this tutorial to place and configure the remaining I/O Ports.



5. Select one of the Violations and examine the Violation Properties view

Notice that links in the Violations Properties view allowing you to select the offending logic objects.

6. Close the DRC view.

Step 21: Running Simultaneous Switching Noise Analysis - "SSN"

Simultaneous Switching Noise (SSN) analysis can also be performed to help identify potential signal integrity concerns.

- 1. In the Flow Navigator, click **Report Noise**.
- 2. Click **OK** in the Run SSN Analysis dialog box.

The Noise report view opens.

→ P25 Dataout_pad0_(1) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P25 Dataout_pad0_(2) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P25 Dataout_pad0_(3) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P25 Dataout_pad0_(4) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P25 Dataout_pad0_(5) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P25 Dataout_pad0_(6) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P25 Dataout_pad0_(7) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P26 OpMode_pad0_(7) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P21 VControl_pad_0_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → P21	Port I/O Std Vcco Slew Drive Strength ((mA) Off-Chip Termination Remaining Margin (%) Note:
S Image: VD Bank 12 (Declicated) (0) VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 VD Bank 12 (Melph Range) (20) VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A P25 DataOut_pad_0_0[1] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A P25 DataOut_pad_0_0[2] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A C25 DataOut_pad_0_0[3] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A C25 DataOut_pad_0_0[3] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A C25 DataOut_pad_0_0[3] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A C25 DataOut_pad_0_0[3] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A D25 DataOut_pad_0_0[1] VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A D26 SuspendM_pad_0_0_0 VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A D26 SuspendM_pad_0_0_0 VVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 A D26 SuspendM_pad_0_0_0 VVCMOS18 1.80 SLOW 12 PP_VTT_50 <td< td=""><td>/O Bank 0 (Dedicated) (0)</td><td></td></td<>	/O Bank 0 (Dedicated) (0)	
Image 1/0 Bank 13 (High Range) (0) LVCMOS18 1.80 SLOW 12 multiple Image 1/0 Bank 13 (High Range) (20) LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.011 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.0121 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.0131 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.0131 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.0101 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.0101 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.011 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.011 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.00 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 Image 1/2 P VTT_50 DataOut_pad_0.00<		
□ □ 10 EVXMOS18 1.80 SLOW 12 PutTs 50 62.2 □ □ Data0ut pad 0_0[0] UVXMOS18 1.80 SLOW 12 PuTTs 50 62.2 □ □ Data0ut pad 0_0[1] UVXMOS18 1.80 SLOW 12 PuTTs 50 62.2 □ Data0ut pad 0_0[2] LVXMOS18 1.80 SLOW 12 PuTTs 50 62.2 □ G25 Data0ut pad 0_0[3] LVXMOS18 1.80 SLOW 12 PuTT 50 62.2 □ G26 Data0ut pad 0_0[3] LVXMOS18 1.80 SLOW 12 PuTT 50 62.2 □ G26 Data0ut pad 0_0[3] LVXMOS18 1.80 SLOW 12 PuTT 50 62.2 □ Data0ut pad 0_0[1] LVXMOS18 1.80 SLOW 12 PuTT 50 62.2 □ Data0ut pad 0_0[1] LVXMOS18 1.80 SLOW 12 PuTT 50 62.2 □		
P H26 DataOut_pad_0_0[0] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P E25 DataOut_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P E26 DataOut_pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P E26 DataOut_pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G26 DataOut_pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G26 DataOut_pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G26 DataOut_pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G24 DataOut_pad_0_0[7] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G24 DataOut_pad_0_0[7] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G25 DataOut_pad_0_0[7] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G24 DataOut_pad_0_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 P G25		12 multiple
→ P25 DataOut_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G25 DataOut_pad_0_o[2] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G25 DataOut_pad_0_o[3] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G26 DataOut_pad_0_o[3] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G25 DataOut_pad_0_o[3] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G24 DataOut_pad_0_o[3] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G24 DataOut_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G24 DataOut_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G24 OpMode_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G26 Suspend/_pad_0_o LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → A20 TermSel_pad_0_o LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D21		12 FP VTT 50 62,96
→ E26 Datout_pod_0.0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.5 > G25 Datout_pd_0.0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.5 > G26 Datout_pd_0.0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.5 > G26 Datout_pd_0.0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > D25 Datout_pd_0.0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > G24 Datout_pd_0.0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > D25 Datout_pd_0.0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > D26 SuspendM_pd_0.0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > D26 SuspendM_pd_0.0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > A20 TermSel_pd_0.0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > A20 TermSel_pd_0.0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 > D21 VCont		
→ G25 DataOut_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G25 DataOut_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G25 DataOut_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G25 DataOut_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G24 DataOut_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G24 DataOut_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G20 OpMode_pad_0.gl DLVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → G26 Suspend_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D26 Suspend_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D21 VControl_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D21 VControl_pad_0.gl LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D21 VContr		
→ G26 Datout_pad_0_o[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D25 Datout_pad_0_o[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D25 Datout_pad_0_o[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D25 Datout_pad_0_o[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D25 Datout_pad_0_o[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → B20 Ophode_pad_0_o[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D26 SuspendM_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D26 SuspendM_od_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D26 SuspendM_od_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D21 VControl_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D21 VControl_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.2 → D21 VCon		
→ E25 DataOut_pad_0_o[3] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D25 DataOut_pad_0_o[3] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → G24 DataOut_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → E22 OpMode_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → B20 OpMode_pad_0_o[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D26 SuspendM_pad_0_o LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D26 SuspendM_pad_0_o LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D26 SuspendM_pad_0_o LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D21 VControl_pad_0_o LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → H24 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D26 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62.2 → D21		
→ G24 DatoUt_pad_0_0[7] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → E22 OpMode_pad_0_0[0] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → B20 OpMode_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A20 TermSel_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 VControl_pad_0_o0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H24 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H21 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D25 VCon		12 FP VTT 50 62.96
→ G24 Dat50ut_pad_0_o[7] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → E22 OpMode_pad_0_o[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → B20 OpMode_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 SuspendM_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A20 TermSel pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A20 TermSel pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C22 TxValid_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 Vcontrol_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H21 Vcontrol_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 Vcontrol_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H21 Vcontrol_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 Vcontrol_	D25 DataOut pad 0 o[6] LVCMOS18 1.80 SLOW	12 FP VTT 50 62.96
→ E22 OpMode_pad_0_o[0] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 SuspendM_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 VControl_Load_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 VControl_Load_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D24 VControl_Load_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H24 VControl_Load_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H24 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H24 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 Xorsletcpad_0_0_0_1 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23		12 FP VTT 50 62.96
→ D26 Suspendik_pad_0_0 LVCMOS18 1.40 SLOW 12 PP_VTT_50 62. → A20 TermSel.pad_0_0 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → C22 TVXalid_pad_0_0 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → D21 VControl_Load_pad_0_0 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → H24 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → H24 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → H21 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → A23 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → A23 VControl_pad_0_0_3 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → A23 VControl_pad_0_0_0_0 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → C23 phy_rst_pad_0_0_0 LVCMOS18 1.80 SLOW 12 PP_VTT_50 62. → C23		
→ A20 TermSel_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C22 TxValid_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D21 VControl_pad_0_o[] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H24 VControl_pad_0_o[] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → H21 VControl_pad_0_o[] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → B21 VControl_pad_0_o[] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 VControl_pad_0_o[] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 VControl_pad_0_o[] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 XcxSelect_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 XcxSelect_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 phy_rst_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 phy_rst_		
→ C22 TxValid_pid_0_0_0 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D21 VControl_pid_0_d0_0_0 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → H24 VControl_pid_0_d0_1 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → H21 VControl_pid_0_d11 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → G21 VControl_pid_0_d21 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D24 VControl_pid_0_d21 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D24 VControl_pid_0_d21 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D23 VControl_pid_0_0_1 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D23 pty_rst_pid_0_0 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D23 pty_rst_pid_0_0 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D24 pty_rst_pid_0_0 LVCMOS18 1.80 SLOW 12 P ² VTT_50 62. → D2	D26 SuspendM_pad_0_o LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ D21 VControl_Load_Dad_0_0_UVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → P44 VControl_pad_0_0[0] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → P44 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → P41 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → G21 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 VControl_pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 Xcr/Select_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 Xcr/Select_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 Xcr/Select_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → M20 Bank 15 (High Range) (20) LVCMOS18 1.80 SLOW 12 multiple 62.	A20 TermSel_pad_0_o LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ PL24 VControl_pad_0_0[0] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → PL21 VControl_pad_0_0[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → G21 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 VControl_pad_0_0[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 VControl_pad_0_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C23 phy_rst_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C3 phy_rst_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C3 phy_rst_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.	C22 TxValid_pad_0_o LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ H21 VControl_pad_0_o[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → G21 VControl_pad_0_o[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → D26 VControl_pad_0_o[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → A23 XcrSelect_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → A23 XcrSelect_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → C23 phy_rst_pad_0_0 LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → Z03 Bank 15 (High Range) (20) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62:	D21 VControl_Load_pad_0_o_LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ G21 VControl_pad_0_o[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → D26 VControl_pad_0_o[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A23 XXx5elect_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C23 XXx5elect_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C23 phy_rst_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → C3 phy_rst_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.	H24 VControl_pad_0_o[0] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ 126 VContro[pad_0_0[3] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → A23 XcxSelect_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → C23 phy_rst_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62: → m 1/O Bank 15 (High Range) (20) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62:	H21 VControl_pad_0_o[1] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
A 23 XorSelect_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → A 23 phy_rst_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. → a 1/0 Bank 15 (High Range) (20) LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.	G21 VControl_pad_0_o[2] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
C23 phy_rst_pad_0_o LVCMOS18 1.80 SLOW 12 FP_VTT_50 62. Image: Model and Marge (20) LVCMOS18 1.80 SLOW 12 multiple 62.	26 VControl_pad_0_0[3] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
TVO Bank 15 (High Range) (20) LVCMOS18 1.80 SLOW 12 multiple	A23 XcvSelect_pad_0_o LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
	C23 phy_rst_pad_0_o LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ L19 DataOut_pad 1_0[0] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.	/O Bank 15 (High Range) (20) LVCMOS18 1.80 SLOW	12 multiple
	L19 DataOut_pad_1_o[0] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ J19 DataOut_pad_1_o[1] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.	J19 DataOut_pad_1_o[1] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
→ 118 DataOut_pad_1_o[2] LVCMOS18 1.80 SLOW 12 FP_VTT_50 62.	J18 DataOut_pad_1_o[2] LVCMOS18 1.80 SLOW	12 FP_VTT_50 62.96
	III	

Figure 32: Noise Results View

Examining the SSN Results View

- 3. Maximize the Noise view and
- 4. Scroll down and expand the list of I/O Banks.
- 5. Select **Summary** in the upper left categories list and examine the information.
- 6. Select **Messages** and **Links** to examine the information.
- 7. Restore the Noise View.

Simultaneous Switching Noise (SSN) analysis can also be performed to help identify potential signal integrity concerns.



Step 22: Updating the Constraint Files with Interactive Assignments

PlanAhead is an interactive constraint assignment environment. Through the course of performing this Tutorial, we have made numerous modifications to the physical constraints in the design. These changes are currently stored in memory. Use the Save Design command to write the changes back to the project constraint files. The Save Design As command can be used to create a new Constraints Set containing all of the currently assigned constraints.

Save the Design

- 1. Select **File > Save Design**.
- In the Sources view double-click the **top.ucf** file under Constraint folder constr_1.
 Notice the new physical constraints applied.
- 3. Close the top.ucf file.

Close the PlanAhead Tool

- 4. Select **File > Exit**.
- 5. Click **OK** to close PlanAhead.

Conclusion

In this tutorial, you:

- Used the I/O pin planning environment 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 together.
- Used the semi-automatic placement modes to assign critical I/O Ports to package pins. Placement of the remaining I/O Ports was done using automatic placement.
- 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 Noise Analysis to validate legal I/O placement.
- Updated the constraint files with the interactive assignments.