

IP Integrator and IP Catalog



Objectives

- > **After completing this module, you will be able to:**
 - >> Achieve greater design productivity using Vivado IP Integrator
 - >> Rapidly create and reuse subsystem level IP with Vivado and IP Integrator
 - >> Describe the IP Packager features
 - >> Differentiate between the free and evaluation base IP available in IP Catalog
 - >> Use the Clocking Wizard to configure and add clocking resources to the design

Outline

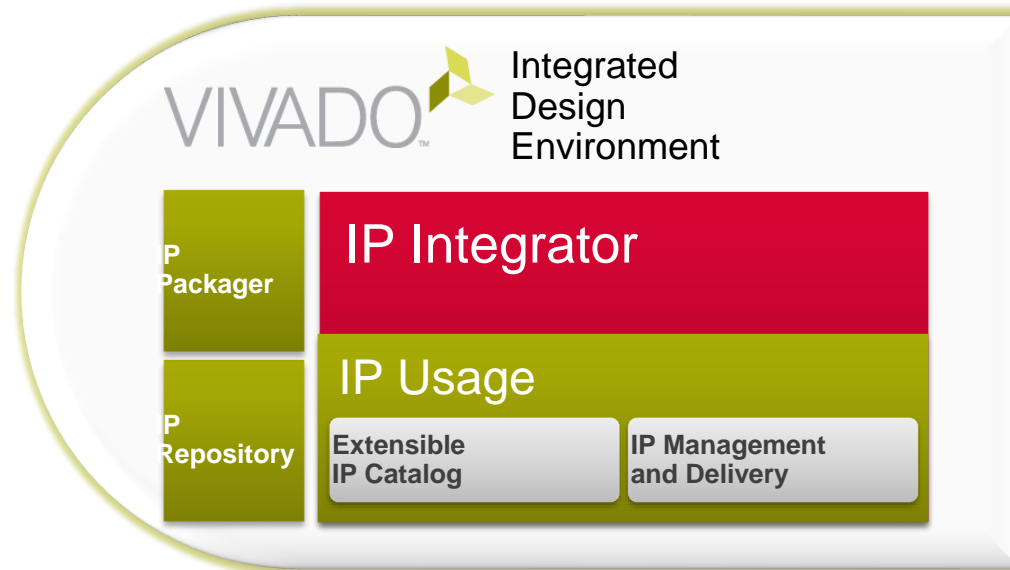
- > *IP Integrator*
- > IP Packager
- > IP Catalog
- > Clocking Wizard
- > Summary

Challenges of Designing with Complex IP

- > **Modern IP contains multiple complex interfaces**
 - >> Many signals, complex port mappings in RTL
 - >> Connectivity in overall design not apparent
- > **Modern IP is highly parameterizable**
 - >> Designer must maintain consistency between interconnected IP blocks
 - >> Designer must understand interactions between blocks
- > **Subsystems containing multiple IP blocks are often difficult to capture and reuse**
 - >> How to best capture IP for reuse
- > **To help designers be more productive when integrating IP, Xilinx introduced the IP Integrator in the Vivado Design Suite**

What is IP Integrator?

- > A hierarchical IP integration tool for processor based and non-processor based systems
- > A graphical and scriptable IP configuration and connection environment



Benefits of Vivado IP Integrator

> Interface Level Connectivity

- >> Connect complex interfaces between IP in a single step
- >> Users can create custom interface definitions

> Integration and Reuse of IP

- >> Rapid creation of complex IP by packaging the contents of a diagram

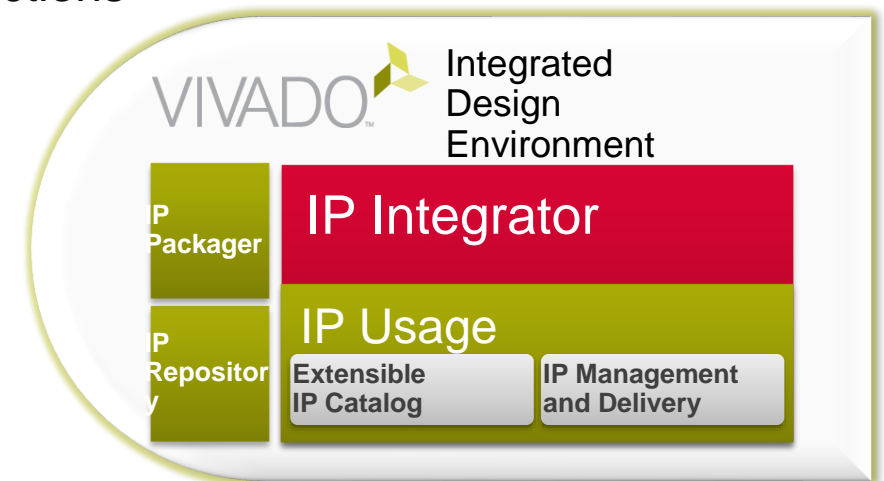
> Automatic Generation of HDL

- >> Instantiates all IP in a diagram and makes all the interconnections

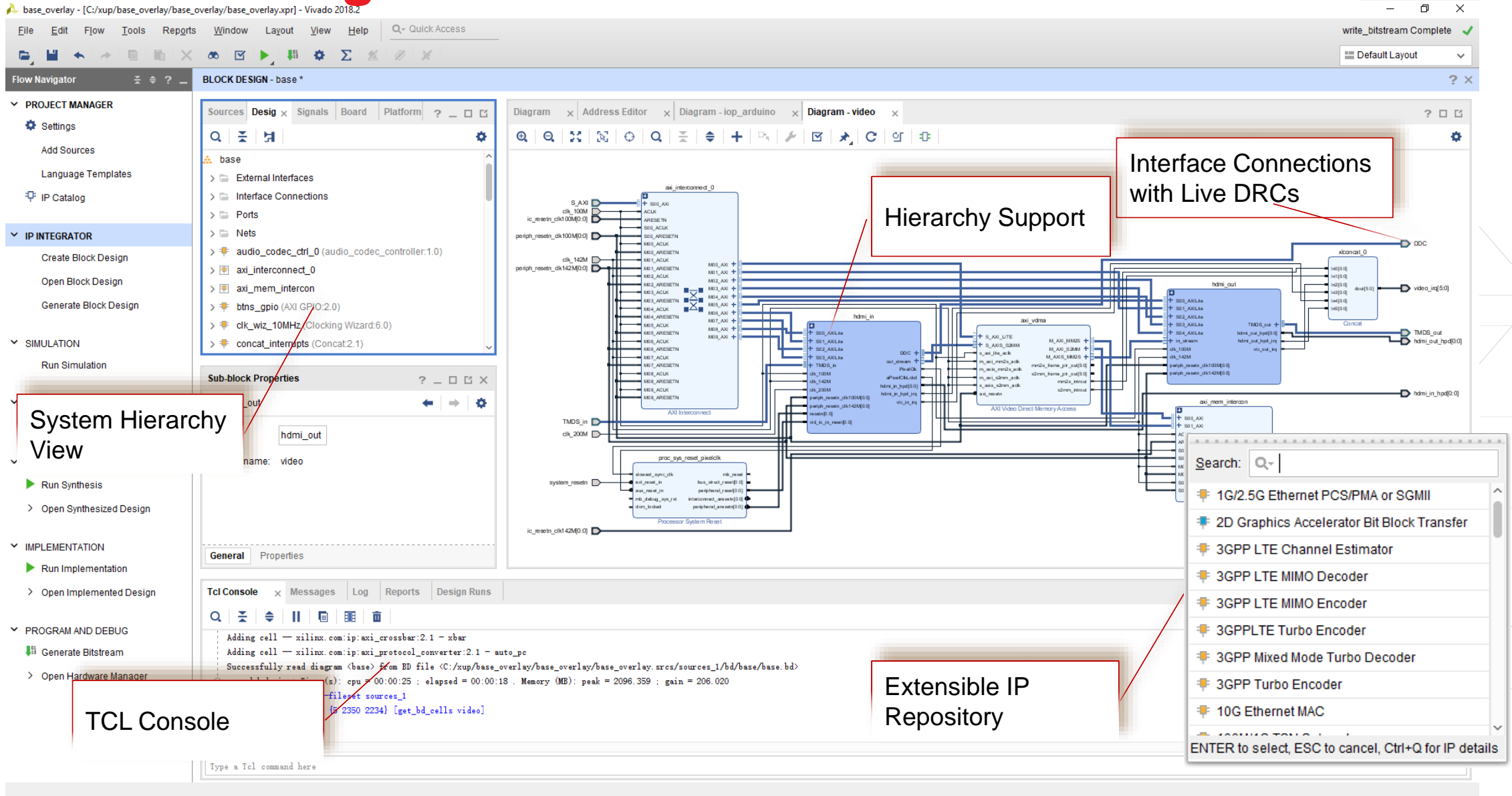
> Take advantage of IP metadata

- >> Propagate correct parameters to connected IP
- >> Recognition of unique data types

> Processor based system support

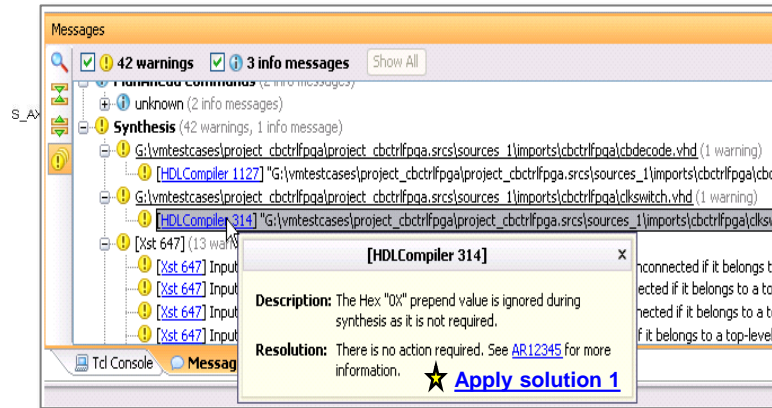


Vivado IP Integrator User Interface

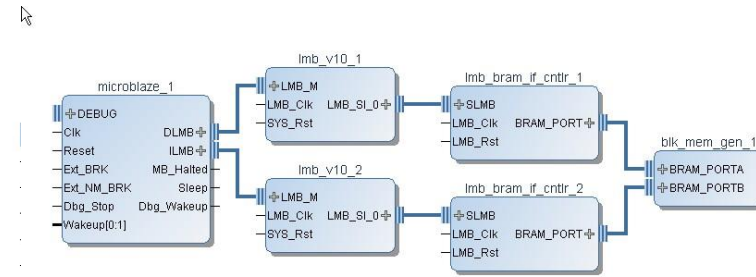


Intelligent IP Integration: Correct by Construction Design

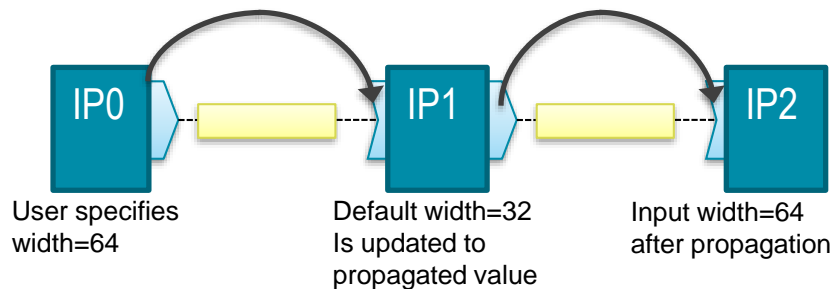
Real-time DRCs



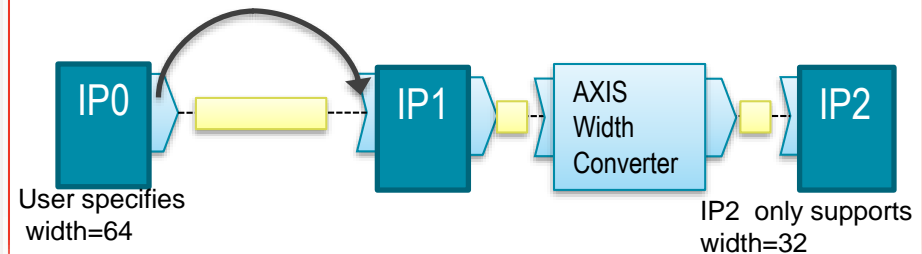
Auto Connection



Parameter Propagation



Parameter Resolution

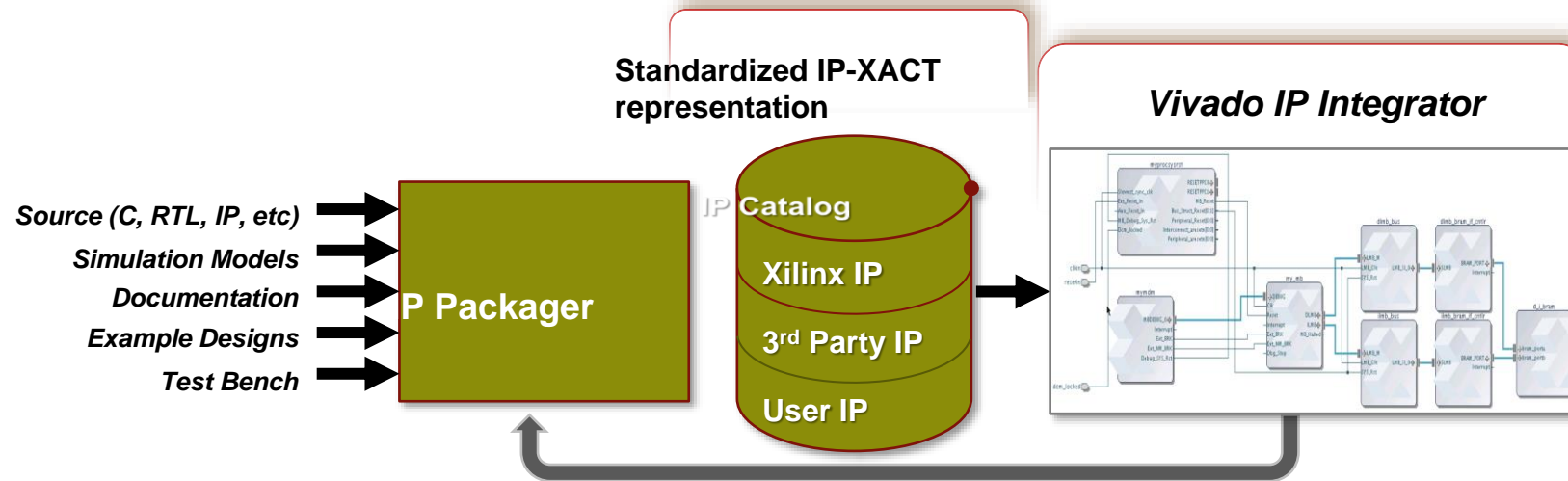


IP Packager



Reusing Your IP

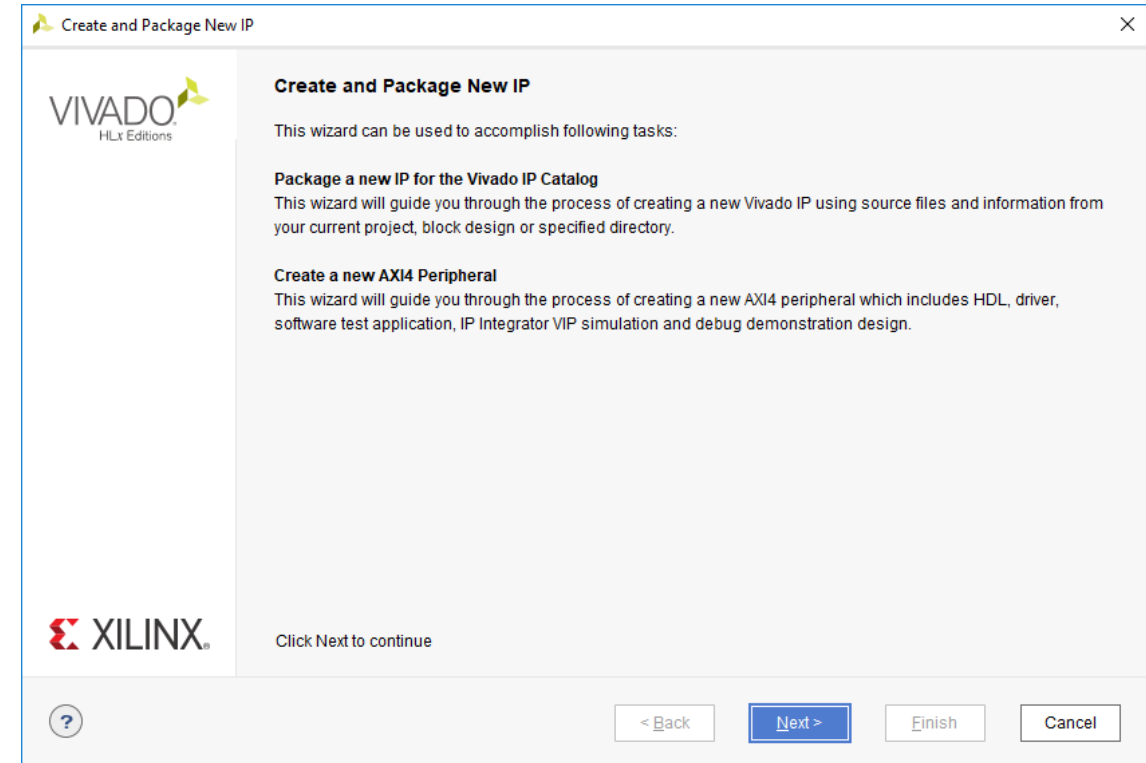
- > IP from many sources can be packaged and made available in Vivado
- > All IP available in the Vivado IP Catalog can be used to create IP Integrator designs
- > Any IP Integrator diagram can be quickly packaged as a single complex IP



Capture Your IP Using the Vivado IP Packager

> Wizard-based flow

- >> Automates generation of IP-XACT IPs
- >> Many pieces of meta-data automatically inferred
- >> Users can add additional meta-data



Customizing IP for Reuse in IP Packager

Packaging Steps

- ✓ Identification
- ✓ Compatibility
- ✓ File Groups
- Customization Parameters
- ✓ Ports and Interfaces
- Addressing and Memory
- ✓ Customization GUI
- ✓ Review and Package

Identification

Vendor:

Library:

Name:

Version:

Display name:

Description:

Vendor display name:

Company url:

Root directory:

Xml file name:

Categories

+ - ↑ ↓

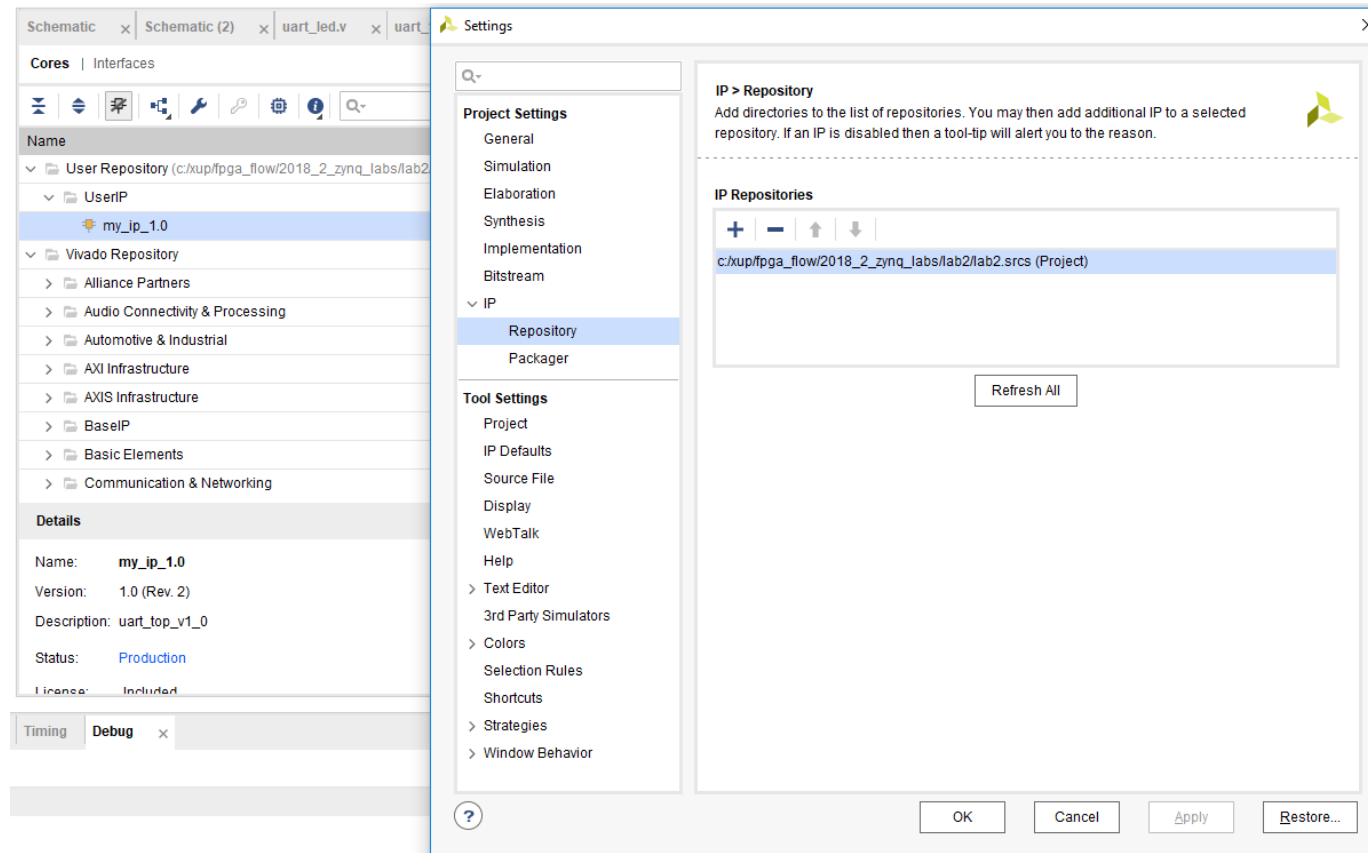
/UserIP

Select Options

Add, Edit or change defaults

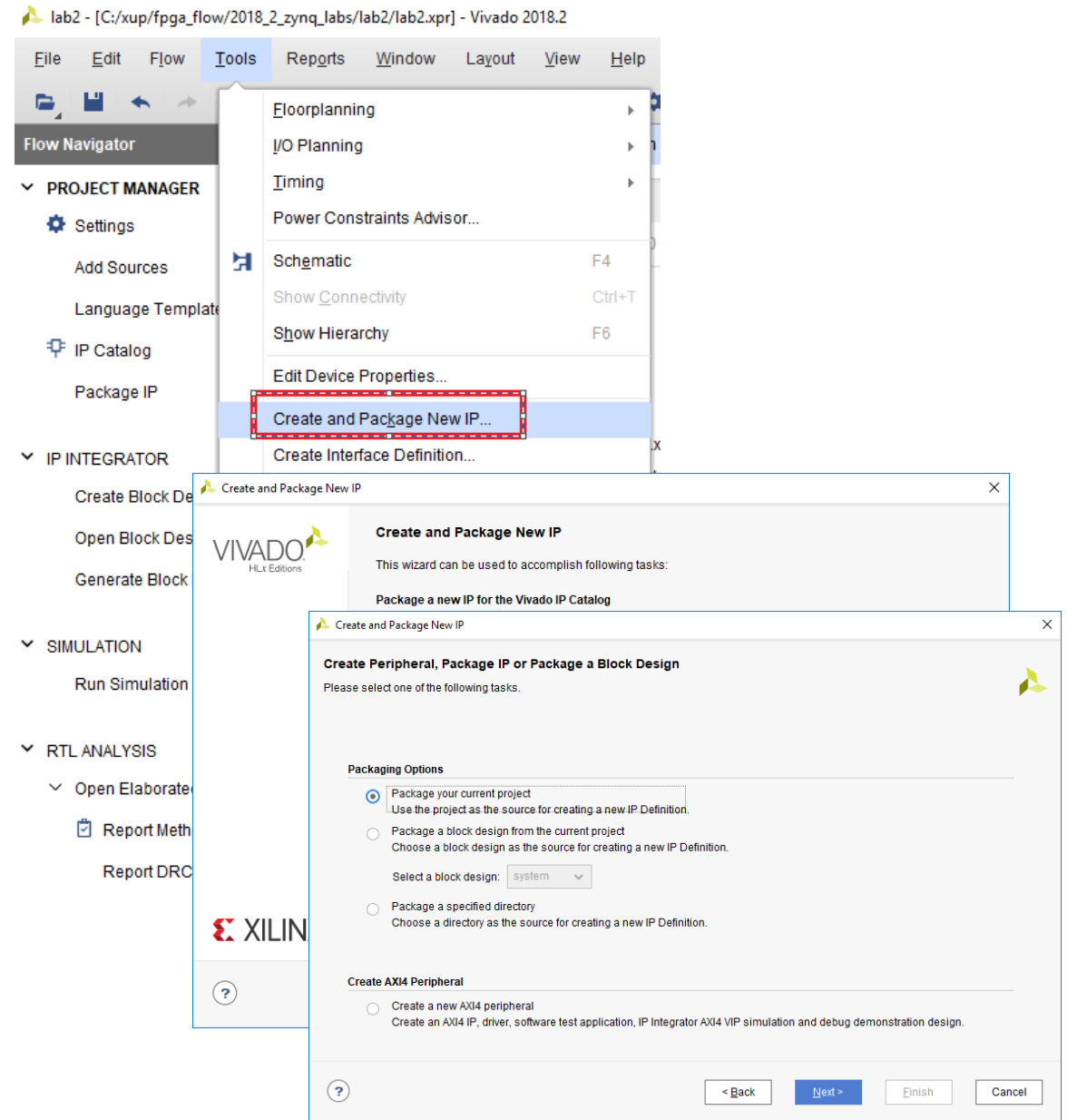
Using and Reusing Packaged IP

- > The Vivado IP Catalog can be extended by adding additional IP Repositories. Third party IP, your custom IP, and Xilinx provided IP are displayed in the identical manner



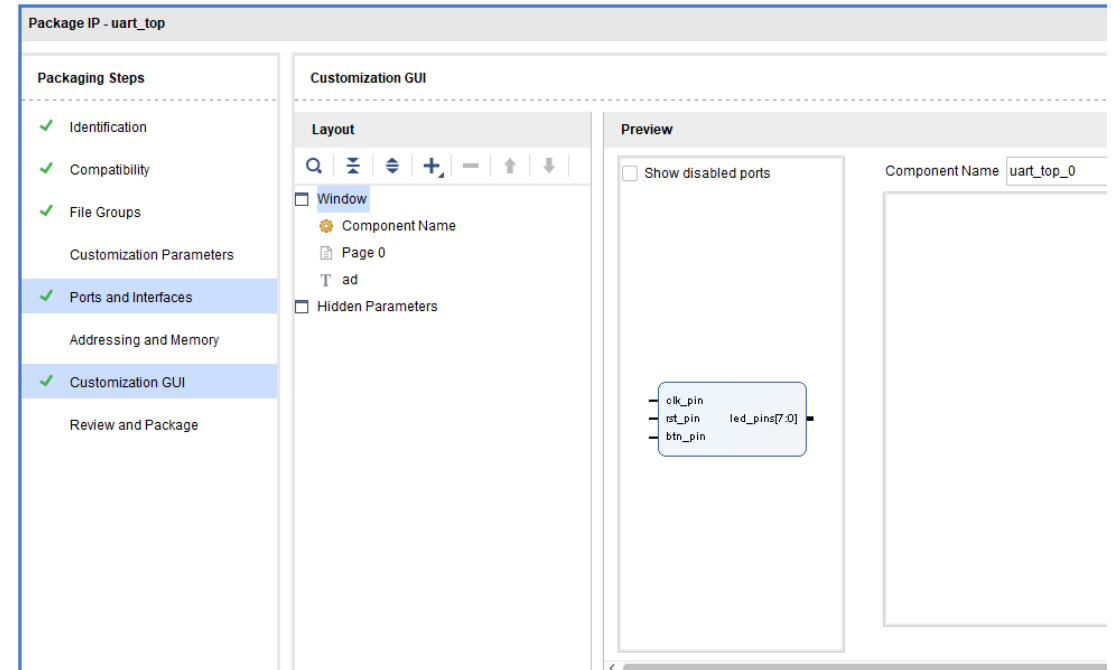
IP Packager

- > The IP Packager allows a core to be packaged and included in the IP Catalog, or for distribution
- > Complete set of files include
 - >> Source code, Constraints, Test Benches (simulation files), documentation
- > IP Packager can be run from Vivado on the current project, or on a specified directory



IP Packager

- > Automatically analyse project/files to determine parameters
- > Initial Summary
- > Identifies
 - >> Files
 - >> Parameters
 - >> Ports
 - >> Interfaces
- > Creates GUI Layout for IPI



IP Packager

> Modify configuration

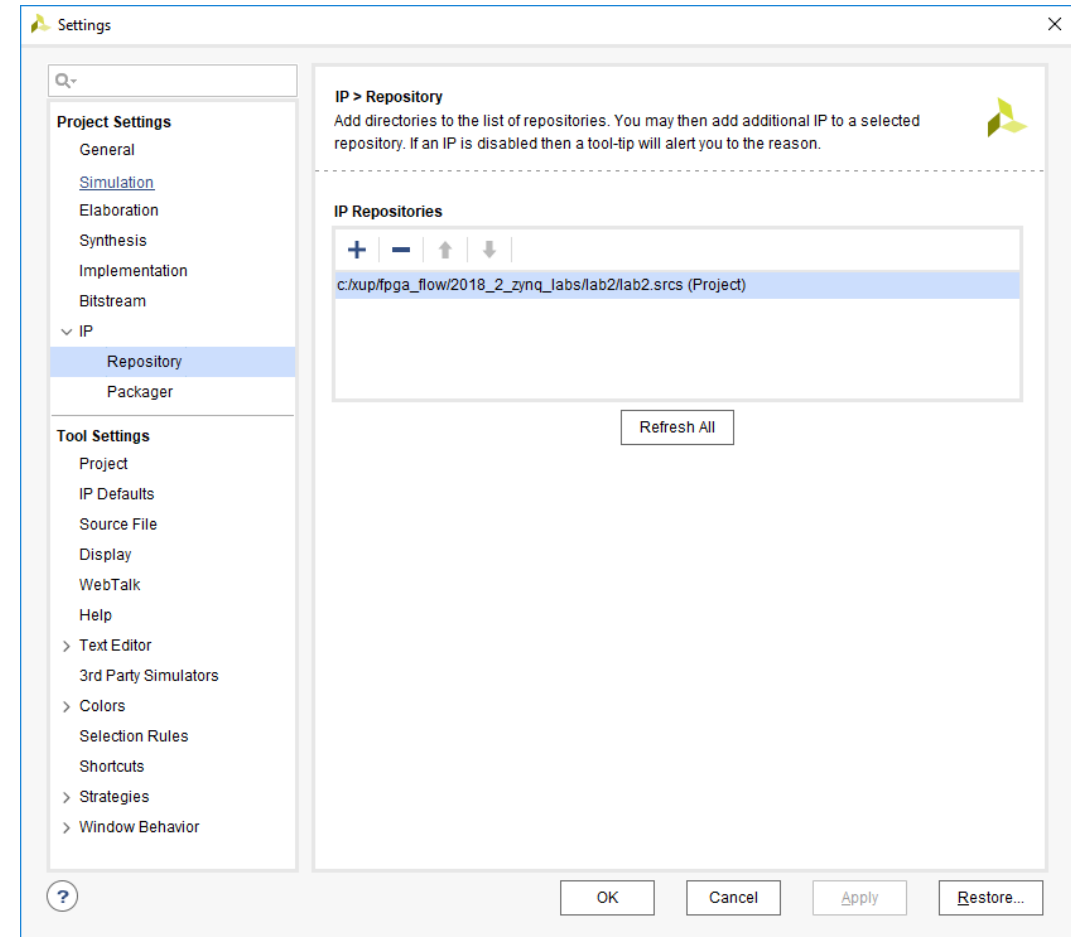
- >> Properties
- >> Compatibility
- >> Files
- >> Custom parameters
- >> Ports
- >> Interfaces
- >> Address and Memory
- >> IP and security

Package IP - uart_top

| Packaging Steps | Identification |
|--------------------------|-------------------------------------------------------------------------------|
| ✓ Identification | Vendor: xilinx.com |
| ✓ Compatibility | Library: user |
| ✓ File Groups | Name: uart_top |
| Customization Parameters | Version: 1.0 |
| ✓ Ports and Interfaces | Display name: uart_top_v1_0 |
| Addressing and Memory | Description: uart_top_v1_0 |
| ✓ Customization GUI | Vendor display name: |
| Review and Package | Company url: |
| | Root directory: c:/xup/fpga_flow/2018_2_zynq_labs/lab2/lab2.srcs |
| | Xml file name: c:/xup/fpga_flow/2018_2_zynq_labs/lab2/lab2.srcs/component.xml |
| | Categories |
| | + - ↑ ↓ |
| | /UserIP |

IP Repository

- > Creates component.xml file for the IP
- > Specify the directory in the repository
- > Displays IP in the repository

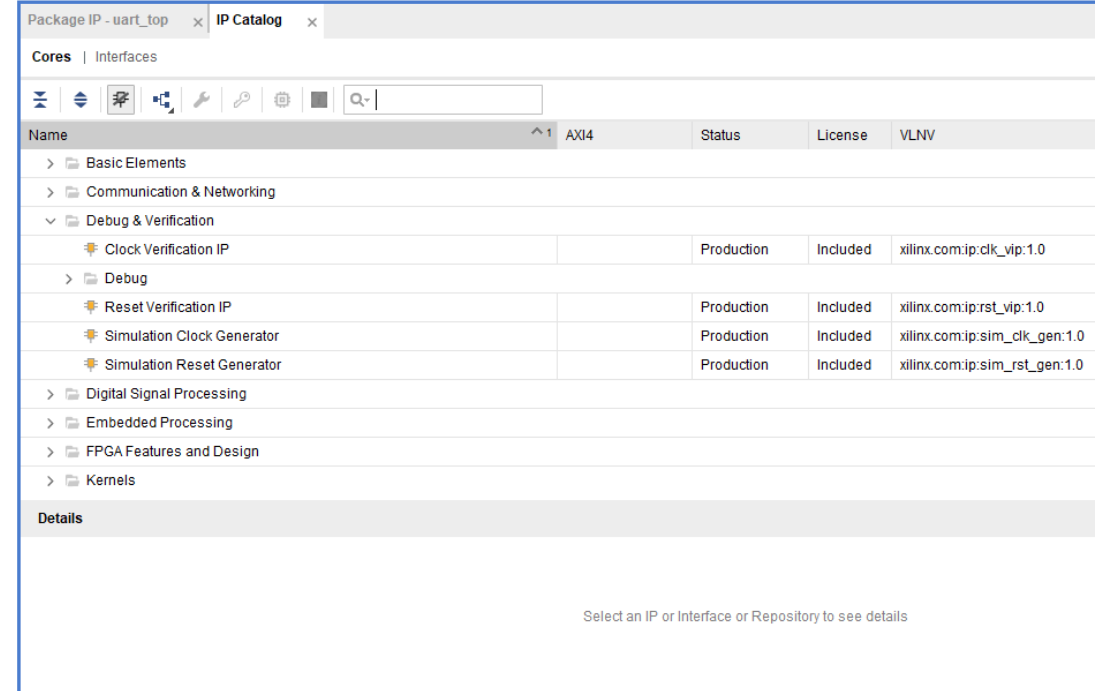


IP Catalog



IP Catalog

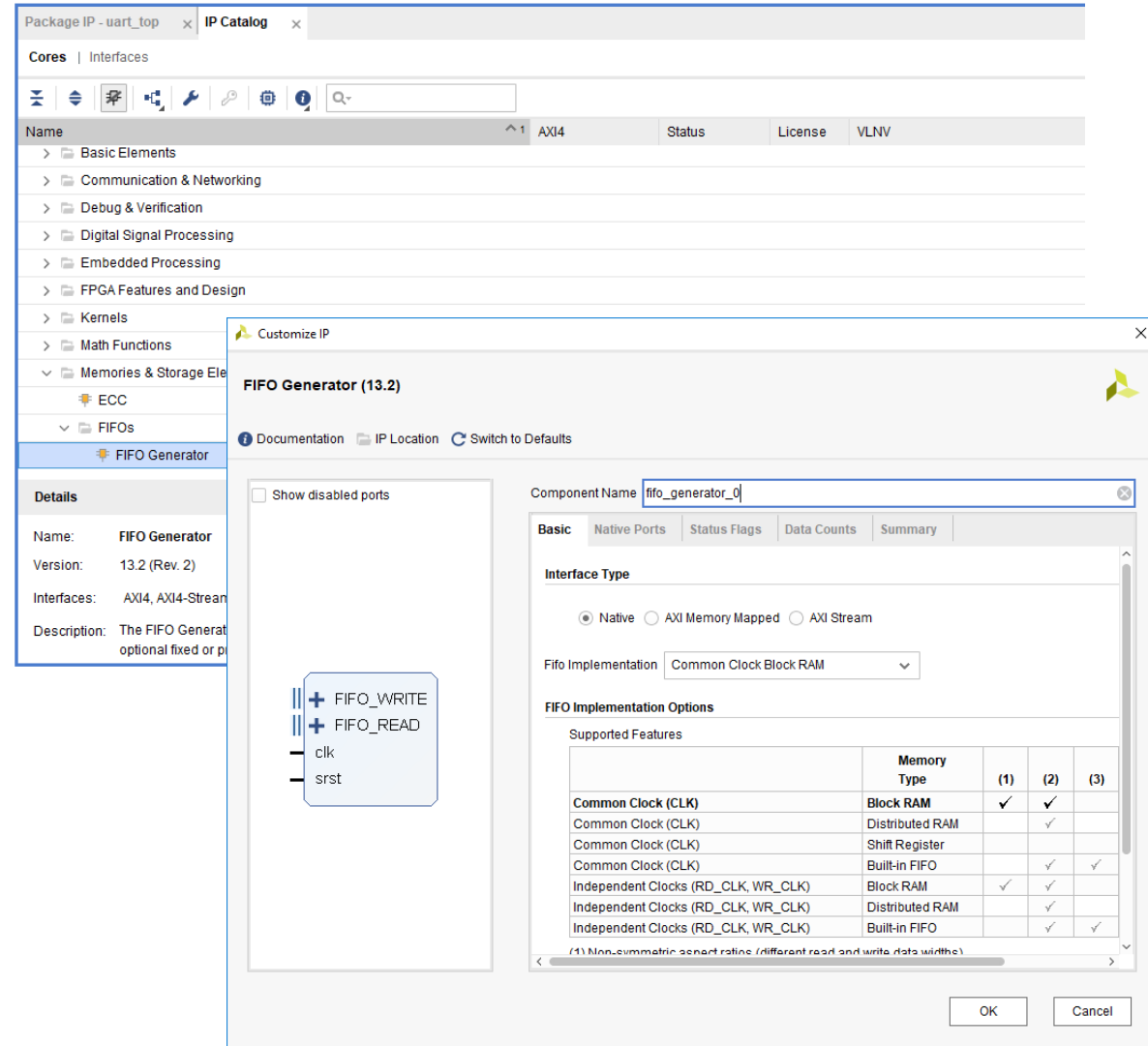
- > The IP Catalog contains a collection of IP that can be used to assemble a system
- > Supported by IPI
- > Facilitates quick system construction
- > Each IP block has its own configuration parameters
- > Most of the IP are free, some require licenses
- > Stored as source code in the install directory
 - >> Always synthesized with the latest tools
 - >> Some proprietary source code is encrypted



Vivado IP Catalog

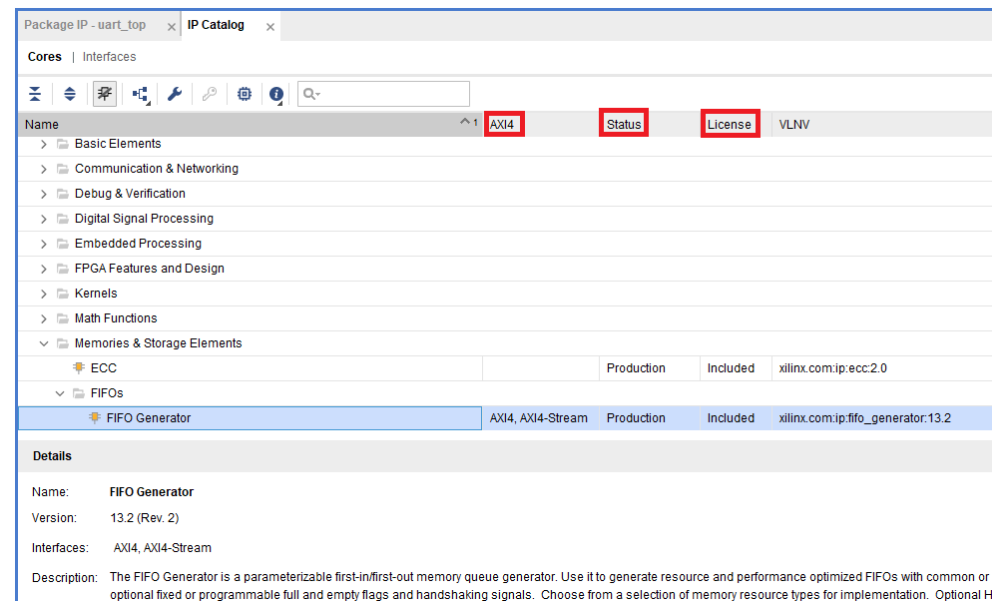
> Integrated IP Support

- >> Instant access to IP customization
- >> Vivado IP GUI look and feel
- >> Support for Vivado synthesis and implementation
- >> Selectable IP output products
- >> Full Tcl support



Peripheral and Non-Peripheral Type IP

- > Many IP in the catalog are peripheral type
 - >> Peripheral type are indicated by AXI4, AXI4-Stream
- > Non-peripheral IP listing do not have entry in the AXI4 column
- > Status column indicates if the IP is of production or pre-production category
- > License column indicates if the IP is free (Included) or costs money (Purchase)



The screenshot shows the Xilinx IP Catalog interface. The 'Cores' tab is selected, and the 'FIFO Generator' IP is highlighted. The table below shows the details of the selected IP.

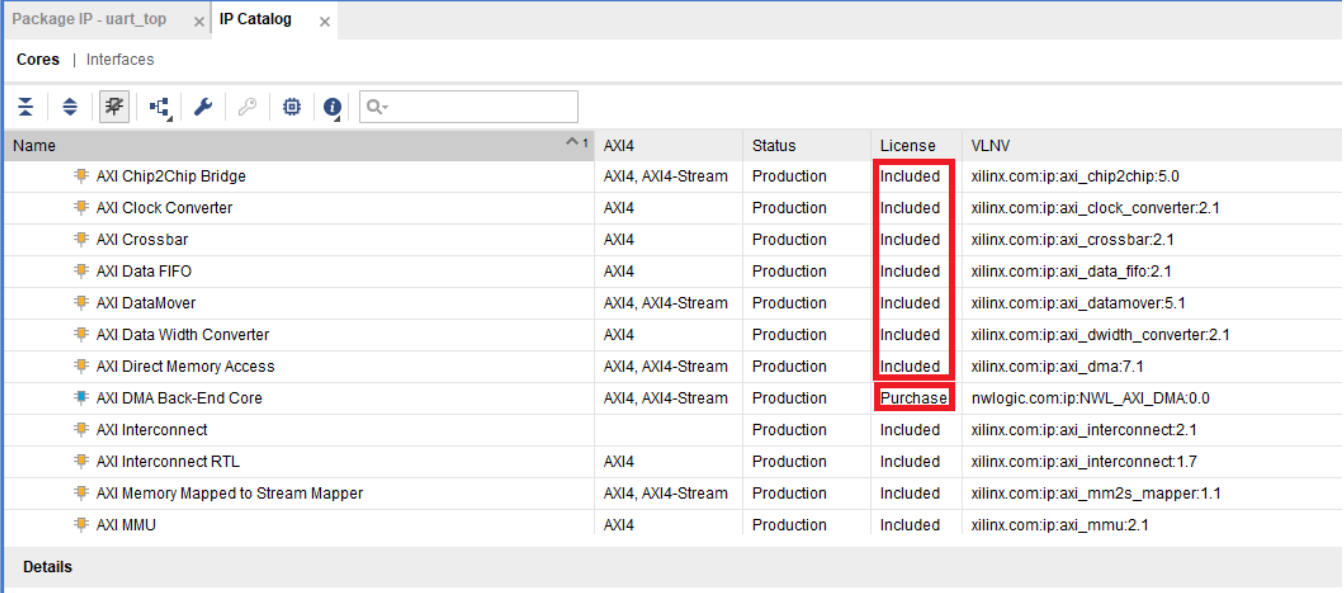
| Name | AXI4 | Status | License | VLNV |
|-----------------------------|-------------------|------------|----------|-----------------------------------|
| Basic Elements | | | | |
| Communication & Networking | | | | |
| Debug & Verification | | | | |
| Digital Signal Processing | | | | |
| Embedded Processing | | | | |
| FPGA Features and Design | | | | |
| Kernels | | | | |
| Math Functions | | | | |
| Memories & Storage Elements | | | | |
| ECC | | Production | Included | xilinx.com:ip:ecc:2.0 |
| FIFOs | | | | |
| FIFO Generator | AXI4, AXI4-Stream | Production | Included | xilinx.com:ip:fifo_generator:13.2 |

Details

Name: FIFO Generator
Version: 13.2 (Rev. 2)
Interfaces: AXI4, AXI4-Stream
Description: The FIFO Generator is a parameterizable first-in-first-out memory queue generator. Use it to generate resource and performance optimized FIFOs with common or optional fixed or programmable full and empty flags and handshaking signals. Choose from a selection of memory resource types for implementation. Optional Hs

IP Peripherals Included as Source (Free)

- > Bus and bridge controllers
- > Debug cores
- > DMA and Timers
- > Inter-processor communication
- > External peripheral controller Memory and memory controller
- > High-speed and low-speed communication peripherals
- > Other cores

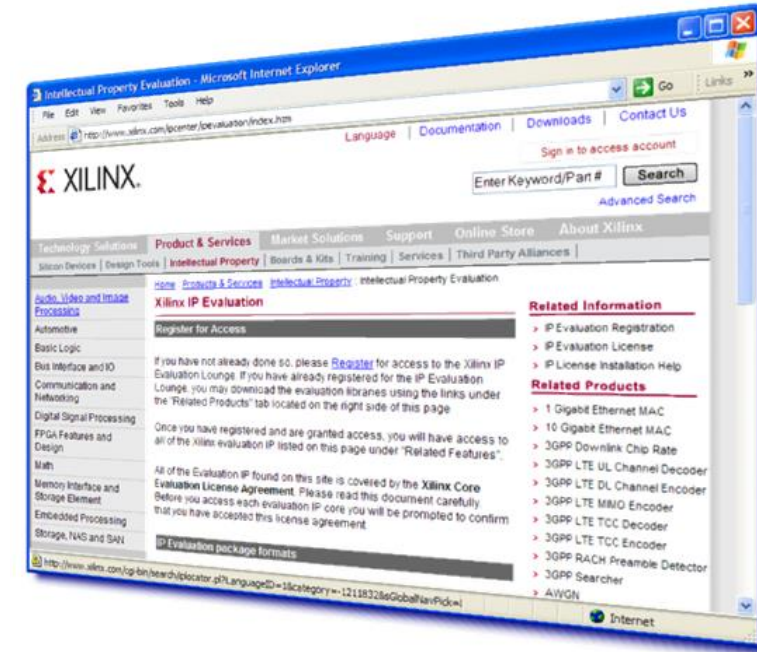


| Package IP - uart_top x IP Catalog x | | | | |
|--------------------------------------|-------------------|------------|----------|----------------------------------------|
| Cores Interfaces | | | | |
| Name | AXI4 | Status | License | VLNV |
| AXI Chip2Chip Bridge | AXI4, AXI4-Stream | Production | Included | xilinx.com:ip:axi_chip2chip:5.0 |
| AXI Clock Converter | AXI4 | Production | Included | xilinx.com:ip:axi_clock_converter:2.1 |
| AXI Crossbar | AXI4 | Production | Included | xilinx.com:ip:axi_crossbar:2.1 |
| AXI Data FIFO | AXI4 | Production | Included | xilinx.com:ip:axi_data_fifo:2.1 |
| AXI DataMover | AXI4, AXI4-Stream | Production | Included | xilinx.com:ip:axi_datamover:5.1 |
| AXI Data Width Converter | AXI4 | Production | Included | xilinx.com:ip:axi_dwidth_converter:2.1 |
| AXI Direct Memory Access | AXI4, AXI4-Stream | Production | Included | xilinx.com:ip:axi_dma:7.1 |
| AXI DMA Back-End Core | AXI4, AXI4-Stream | Production | Purchase | nwllogic.com:ip:NWL_AXI_DMA:0.0 |
| AXI Interconnect | | Production | Included | xilinx.com:ip:axi_interconnect:2.1 |
| AXI Interconnect RTL | AXI4 | Production | Included | xilinx.com:ip:axi_interconnect:1.7 |
| AXI Memory Mapped to Stream Mapper | AXI4, AXI4-Stream | Production | Included | xilinx.com:ip:axi_mm2s_mapper:1.1 |
| AXI MMU | AXI4 | Production | Included | xilinx.com:ip:axi_mmu:2.1 |
| Details | | | | |

IP Cores Included as Evaluation

- > AXI CAN controller
- > AXI USB2 device
- > Video IP
- > Telecoms/ Wireless IP

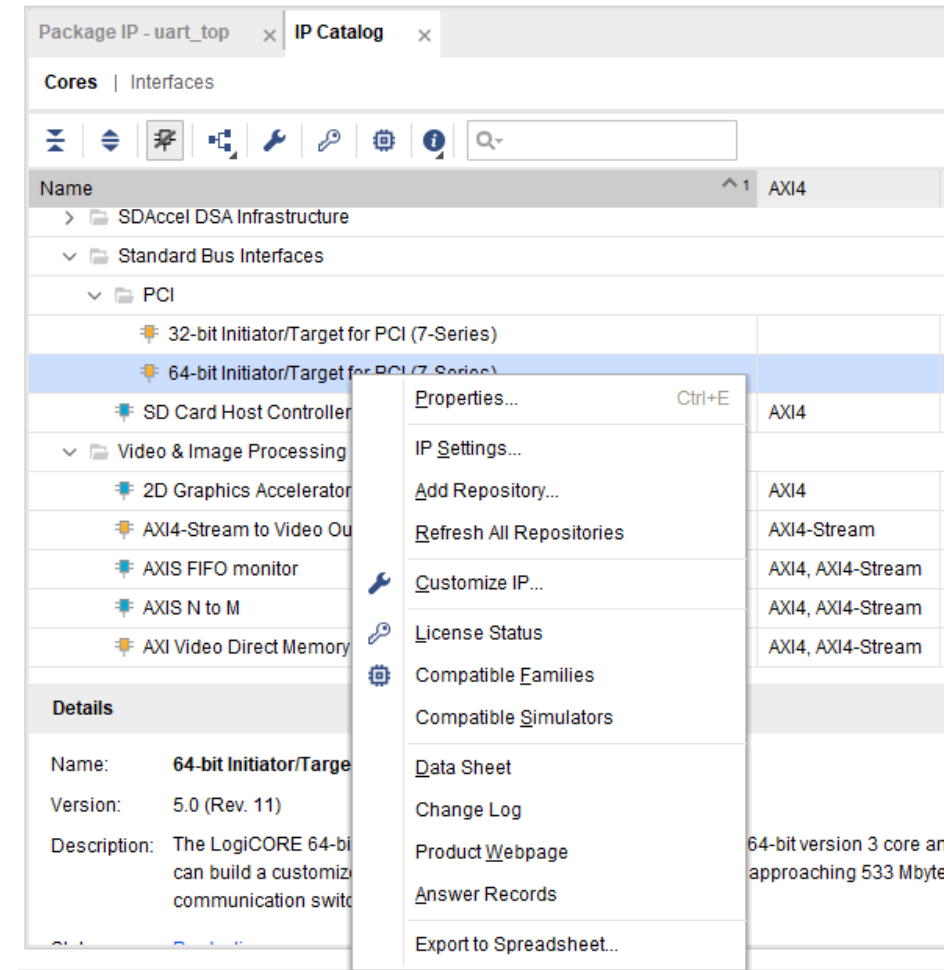
| Package IP - uart_top x IP Catalog x | | | | |
|--------------------------------------------|-------------------|------------|----------|--------------------------------------------|
| Cores Interfaces | | | | |
| Name | | | | |
| > SDAccel DSA Infrastructure | AXI4 | Status | License | VLNV |
| > Standard Bus Interfaces | | | | |
| > PCI | | | | |
| SD Card Host Controller | AXI4 | Production | Purchase | logicbricks.com:logicbricks:logisdhc:0.0 |
| > Video & Image Processing | | | | |
| 2D Graphics Accelerator Bit Block Transfer | AXI4 | Production | Purchase | logicbricks.com:logicbricks:logibitbtt:0.0 |
| AXI4-Stream to Video Out | AXI4-Stream | Production | Included | xilinx.com:ip:v_axi4s_vid_out:4.0 |
| AXIS FIFO monitor | AXI4, AXI4-Stream | Production | Purchase | omnitek.tv:ip:omni_axis_fifo_mon:0.0 |
| AXIS N to M | AXI4, AXI4-Stream | Production | Purchase | omnitek.tv:ip:omni_axis_n_to_m:0.0 |
| AXI Video Direct Memory Access | AXI4, AXI4-Stream | Production | Included | xilinx.com:ip:axi_vdma:6.3 |
| Bitmap 2.5D Graphics Accelerator | AXI4 | Production | Purchase | logicbricks.com:logicbricks:logibmp:0.0 |
| Chroma Down-Sampler | AXI4, AXI4-Stream | Production | Purchase | omnitek.tv:ip:omni_chroma_down:0.0 |



Xilinx developed, delivered, and supported Evaluation IP installs with a 90-day evaluation license

IP Cores

- > Right click to customize
- > Determine compatibility
- > Product Guide (datasheet) > Document Navigator
- > Change Log
- > Product Webpage
- > Answer record
- > Export complete IP Catalog to excel

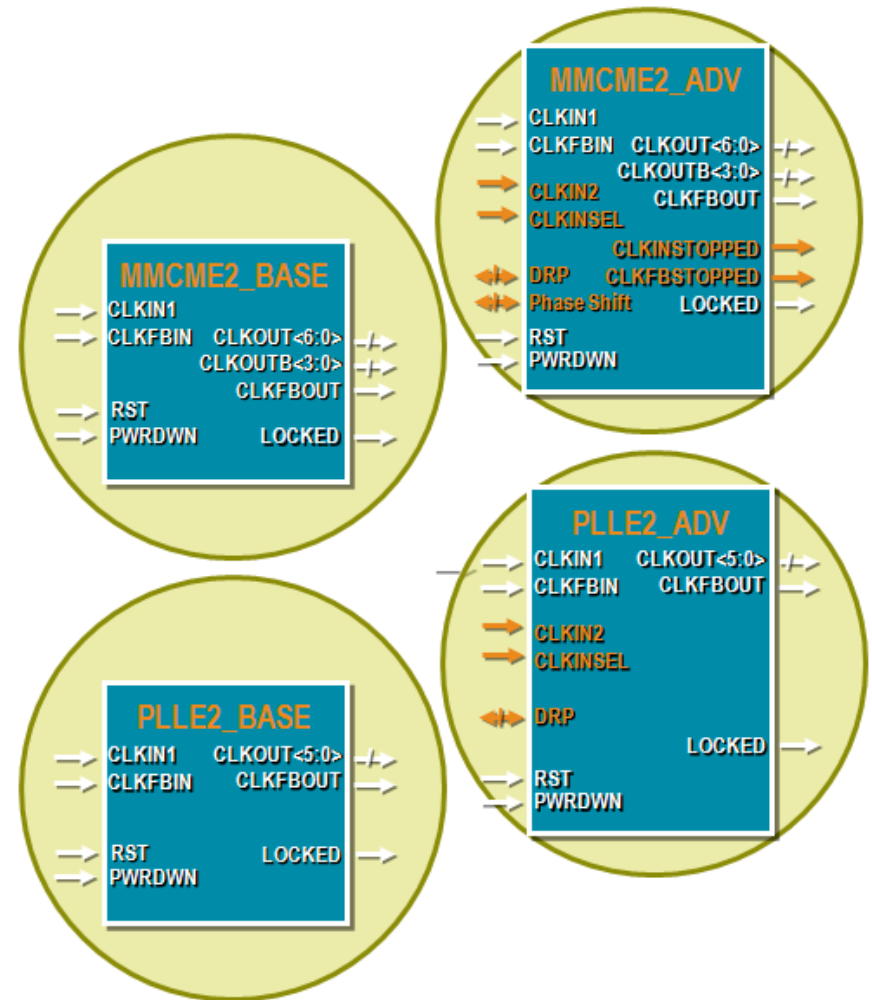


Clocking Wizard



Clocking Resources: MMCM and PLL

- > Up to 24 CMTs per device
- > One MMCM and one PLL per CMT
- > Two software primitives (instantiation)
 - >> *_BASE has only the basic ports
 - >> *_ADV provides access to all ports
- > PLL is primarily intended for use with the I/O phaser for high-speed memory controllers
- > The MMCM is the primary clock resource for user clocks



Inference

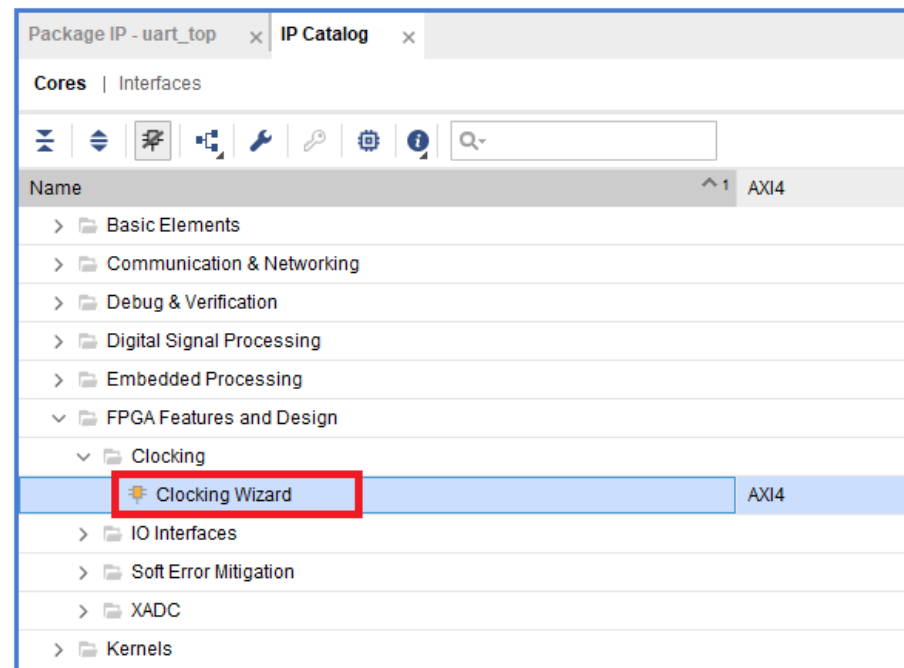
- > **Clock networks are represented by nets in your RTL design**
 - >> The mapping of an RTL net to a clock network is managed by using the appropriate clock buffer to generate that net
- > **Certain resources can be inferred**
 - >> A primary input net (with or without an IBUF instantiated) will be mapped to a global clock if it drives the clock inputs of clocked resources
 - The BUFG will be inferred
 - >> BUFH drivers will be inferred whenever a global clock (driven by a BUFG) is required in a clock region
 - BUFHs for each region required will be inferred
- > **BUFIO, BUFR, and BUFMR cannot be inferred**
 - >> Instantiating these buffers tells the tools that you want to use the corresponding clock networks
- > **PLLs and MMCMs cannot be inferred**

Instantiation

- > **All clocking resources can be directly instantiated in your RTL code**
 - >> Simulation models exist for all resources
 - >> Refer to the Library Guide for HDL Designs
 - >> Use the Language Templates (💡) tab
- > **PLLs and MMCMs have many inputs and outputs, as well as many attributes**
 - >> Optimal dividers for obtaining the desired characteristics may be hard to derive
 - >> The Clocking Wizard via the IP Catalog
 - Only *_ADV available

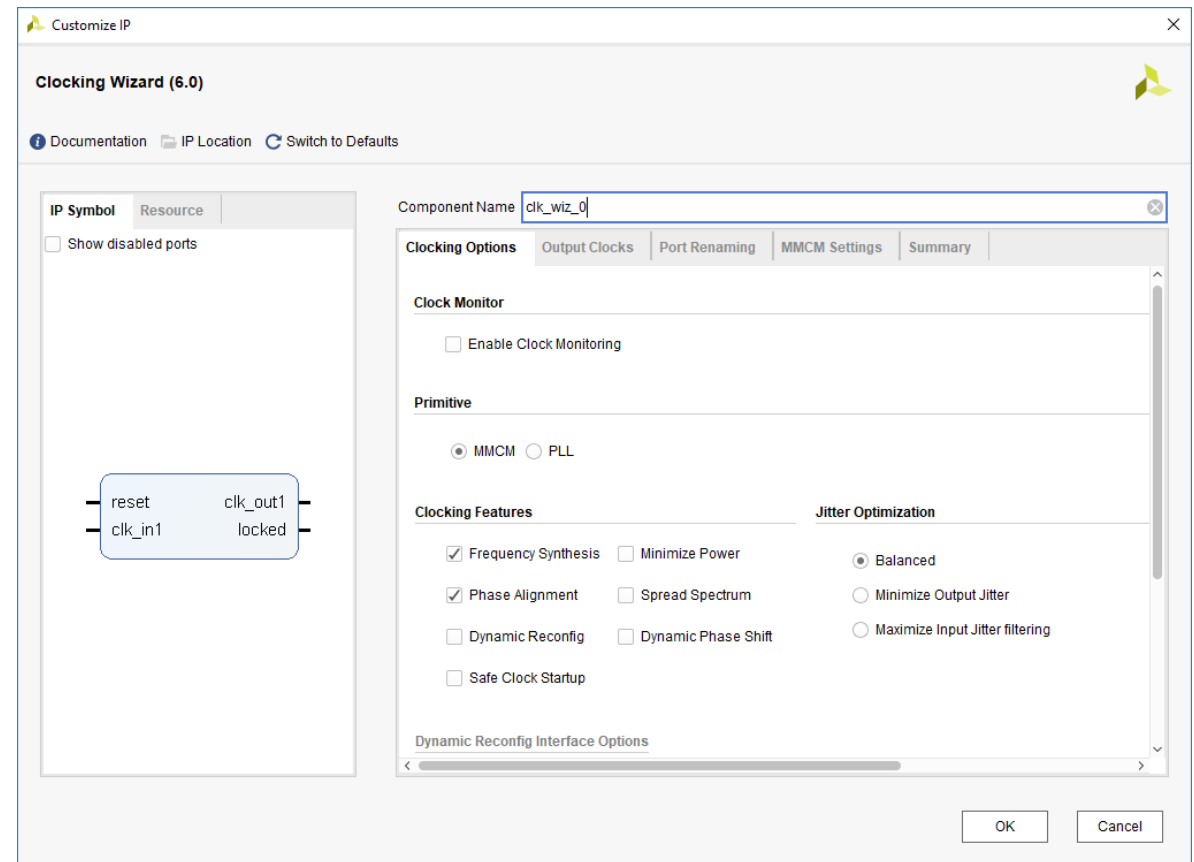
Invoking Clocking Wizard

- > Click on the IP Catalog
- > Expand FPGA Features and Design > Clocking
- > Double-click on Clocking Wizard
- > The Clocking Wizard walks you through the generation of complete clocking subsystems



The Clocking Wizard: Clocking Options

- > **Select Primitives to be used**
 - >> MMCME2_ADV
 - >> PLLE2_ADV
- > **Specify the primary input frequency and source type**
 - >> Optionally, select and specify secondary input clock
- > **Select clocking features**
 - >> Frequency synthesis
 - >> Phase alignment
 - >> Dynamic phase shift
 - >> ...



The Clocking Wizard: Output Clocks

- > Select the desired number of output clocks
- > Set the desired output frequencies
- > Select optional ports

Customize IP

Clocking Wizard (6.0)

Documentation IP Location Switch to Defaults

Component Name: clk_wiz_0

Output Clocks

The phase is calculated relative to the active input clock.

| Output Clock | Port Name | Output Freq (MHz) | | Phase (degrees) | | R |
|----------------------------------------------|-----------|-------------------|---------|-----------------|--------|---|
| | | Requested | Actual | Requested | Actual | |
| <input checked="" type="checkbox"/> clk_out1 | clk_out1 | 100.000 | 100.000 | 0.000 | 0.000 | 5 |
| <input type="checkbox"/> clk_out2 | clk_out2 | 100.000 | N/A | 0.000 | N/A | 5 |
| <input type="checkbox"/> clk_out3 | clk_out3 | 100.000 | N/A | 0.000 | N/A | 5 |
| <input type="checkbox"/> clk_out4 | clk_out4 | 100.000 | N/A | 0.000 | N/A | 5 |
| <input type="checkbox"/> clk_out5 | clk_out5 | 100.000 | N/A | 0.000 | N/A | 5 |
| <input type="checkbox"/> clk_out6 | clk_out6 | 100.000 | N/A | 0.000 | N/A | 5 |
| <input type="checkbox"/> clk_out7 | clk_out7 | 100.000 | N/A | 0.000 | N/A | 5 |

☐ USE CLOCK SEQUENCING

Clocking Feedback

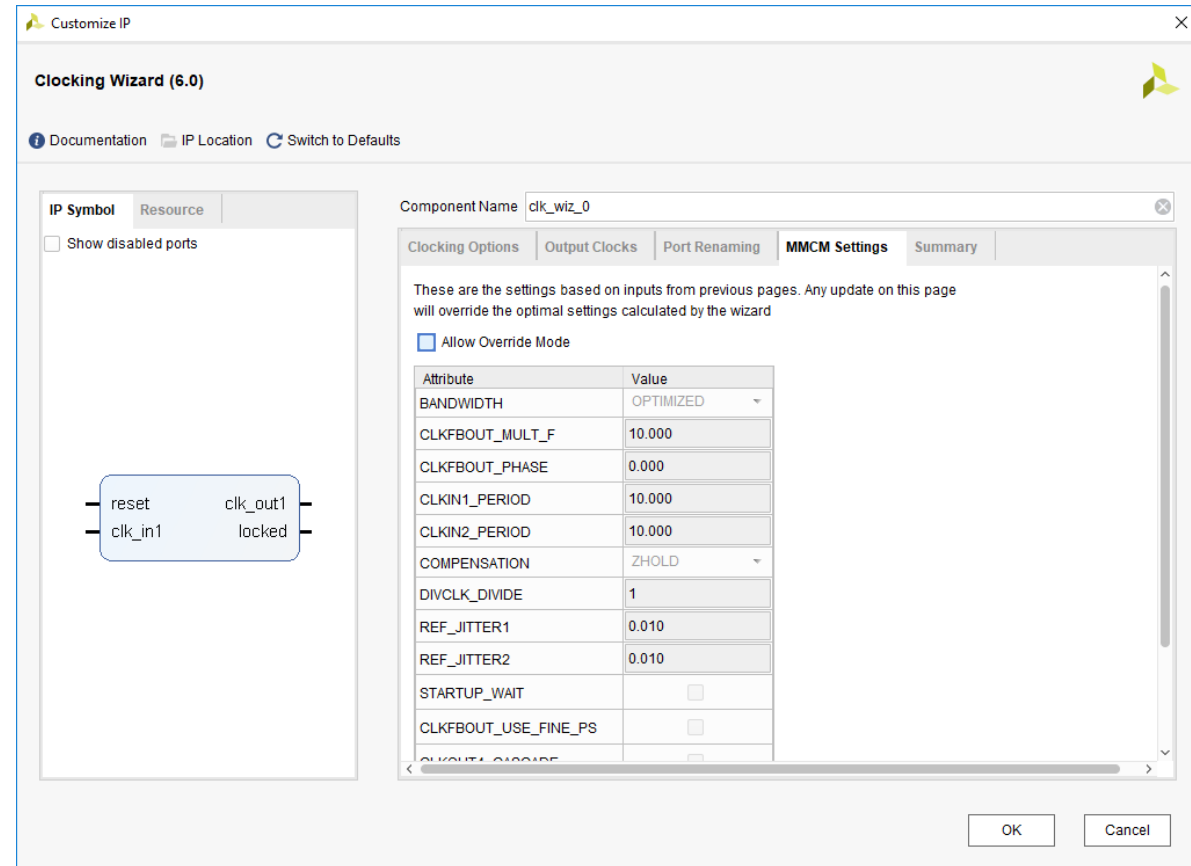
| Output Clock | Sequence Number | Source | Signaling |
|--------------|-----------------|--------|-----------|
| clk_out1 | 1 | | |

☒ Automatic Control On-Chip ☐ Single

OK Cancel

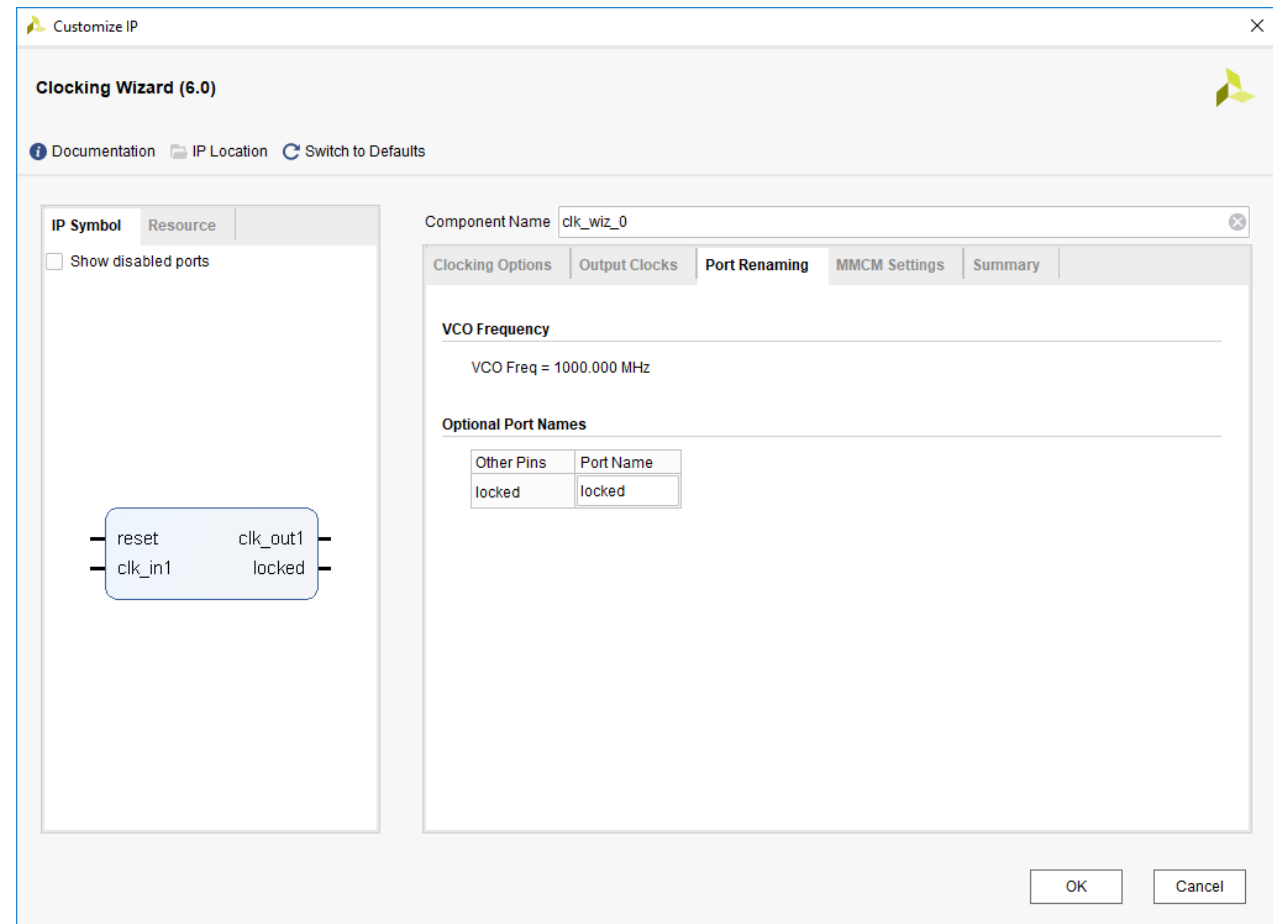
The Clocking Wizard: MMCM Settings

- > Allows overriding of wizard settings
- > Sets the phase, period, jitter...



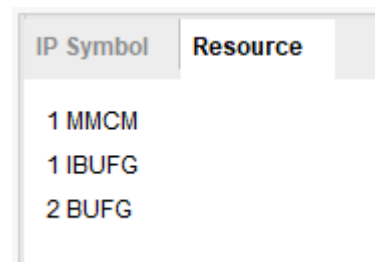
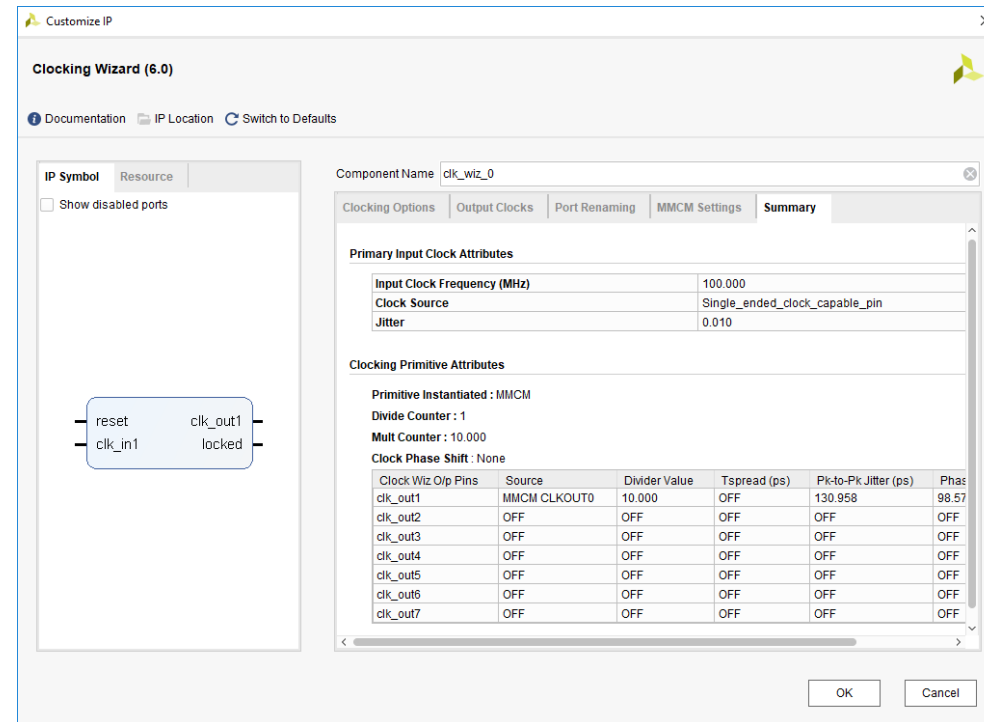
The Clocking Wizard: Port Renaming

- > Change input/output port names
- > Change optional port names



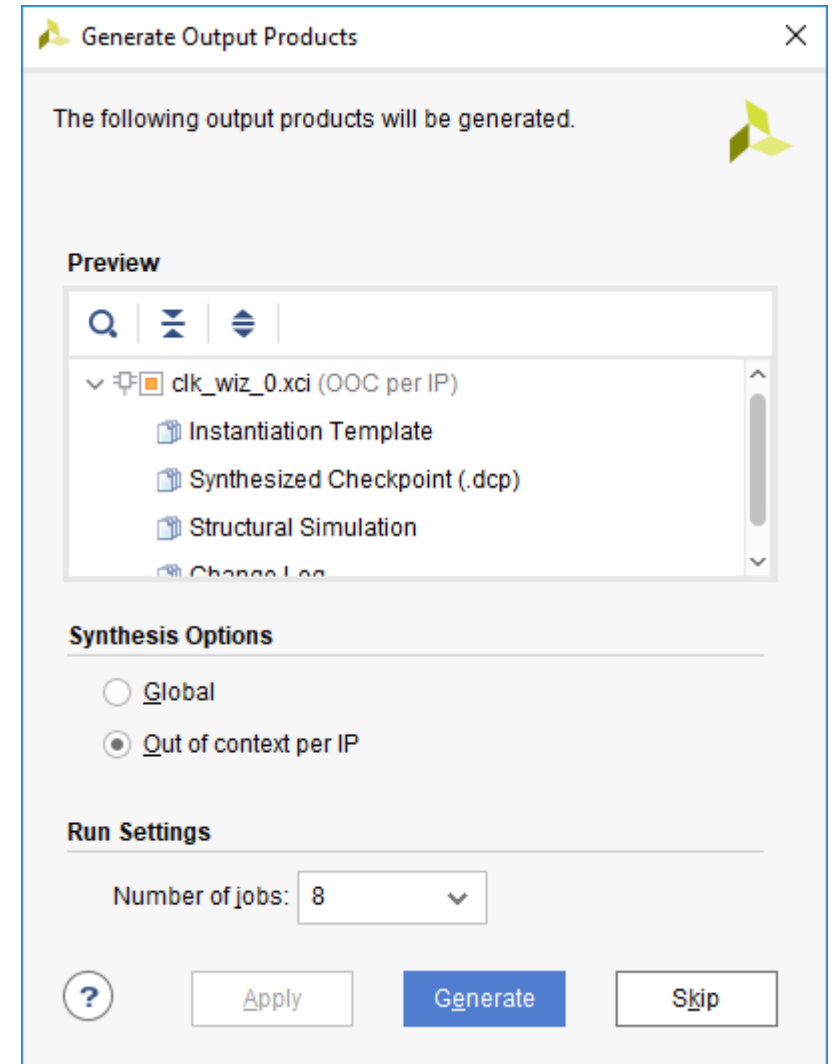
The Clocking Wizard: Summary

- > Shows the input, output frequencies
- > Other attributes depending on the selections made
- > The Resource tab on the left provides summary of type and number of resources used



Generating Output Products

- > Generate Instantiation Template
- > Synthesis model
- > Simulation model
- > Implementation related files
- > Examples on how to use the product
- > Optionally, synthesized design checkpoint so it can be used in the implementation without re-synthesizing in the project



Summary



Summary

- > **Vivado is an IP Centric design environment, allowing multiple ways to add/manage IP**
 - >> Automatic RTL generation
 - >> Rapid assembly, packaging, and reuse of IP subsystems
 - >> Processor based system design support
- > **Vivado IP Integrator provides unique time to market advantages when designing with complex IP**
- > **The IP Packagers provides flexibility of IP reuse and repackaging**
- > **The IP Catalog provides access to extensive IP resources**
 - >> Simple IP to highly complex IP
 - >> Grouped according to functionality
- > **The Clocking Wizard enables the configuration of complex clocking resources**