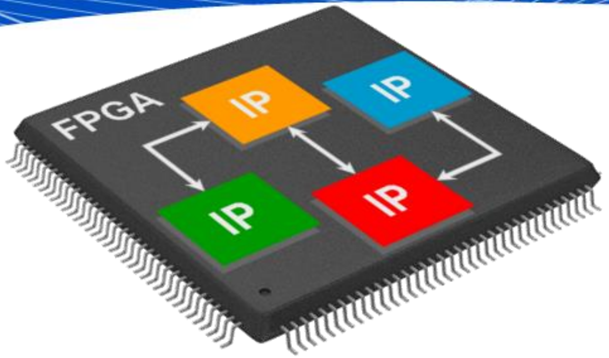
POWER & ENERGY CONVERSION LABORATORY

Lecture 05
Intellectual Property (IP) Cores



Electrical Engineering Department
Pontificia Universidad Católica de Chile
peclab.ing.uc.cl

PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IP Cores

What is an IP CORE?

- An intellectual property core (IP core) is a functional block of logic or data used to make a field-programmable gate array (FPGA) or application-specific integrated circuit for a product.
- The IP of one party may be licensed by others for use in their own ICs and semiconductors.
- Most SOC chips incorporate a standard microprocessor and standardized functionalities, accommodating design reuse across multiple ICs by multiple vendors on a licensing basis.
- Most IP cores are developed using hardware description languages (HDLs), like VHSIC HDL, Verilog or SystemVerilog.

IP Cores

What is an IP CORE?

- **1. Hard IP core**
  - It is a physical manifestation of the IP's design.
  - The IP owner usually offers the hard IP core as a layout design mapped to a process technology.
  - A hard core is not portable or flexible.
  - It has a fixed location in the FPGA and cannot be ported to other FPGAs or customized for different process technologies.
  - The advantage of the hard IP core is that it reduces the need for code maintenance. It also minimizes timing violations, fosters high performance and functionality, and provides a low-cost IP core option since it is included in the FPGA.
  - This makes it best used for plug-and-play applications.

- **2. Firm IP core**
  - Or semihard IP core, can be configured and modified for different applications.
  - It provides greater flexibility to place the module in the FPGA and interconnect it with other modules.
  - Once the firm component is instantiated at the top level, it can be moved around within the FPGA to satisfy performance and timing requirements.
  - More flexible than Hard IP Core.
  - It offers limited portability compared to a soft IP core.
  - Modifications to the source code are not possible, and it may cause some timing or performance issues when reused.

IP Cores

What is an IP CORE?

- **3. Soft IP core**
  - Most flexible type of IP core since it can be customized to map to any process technology and reused for a wide range of applications.
  - It can exist as a modifiable netlist, which is a list of the logic gates and associated interconnections making up the IC.
  - With a soft IP core, the licensee gets the source code with the license. This enables it to modify the IP to suit its application and easily integrate it with its modules. It can also reuse the core for many types of FPGAs.
  - One of the main drawbacks of a soft IP core is its cost. Since the vendor provides the modifiable source code, the soft core tends to be more expensive than a hard or firm core.
  - It may also have to put in extra effort customizing its applications. In addition, the application's performance may vary and not match its requirements.

IP Cores

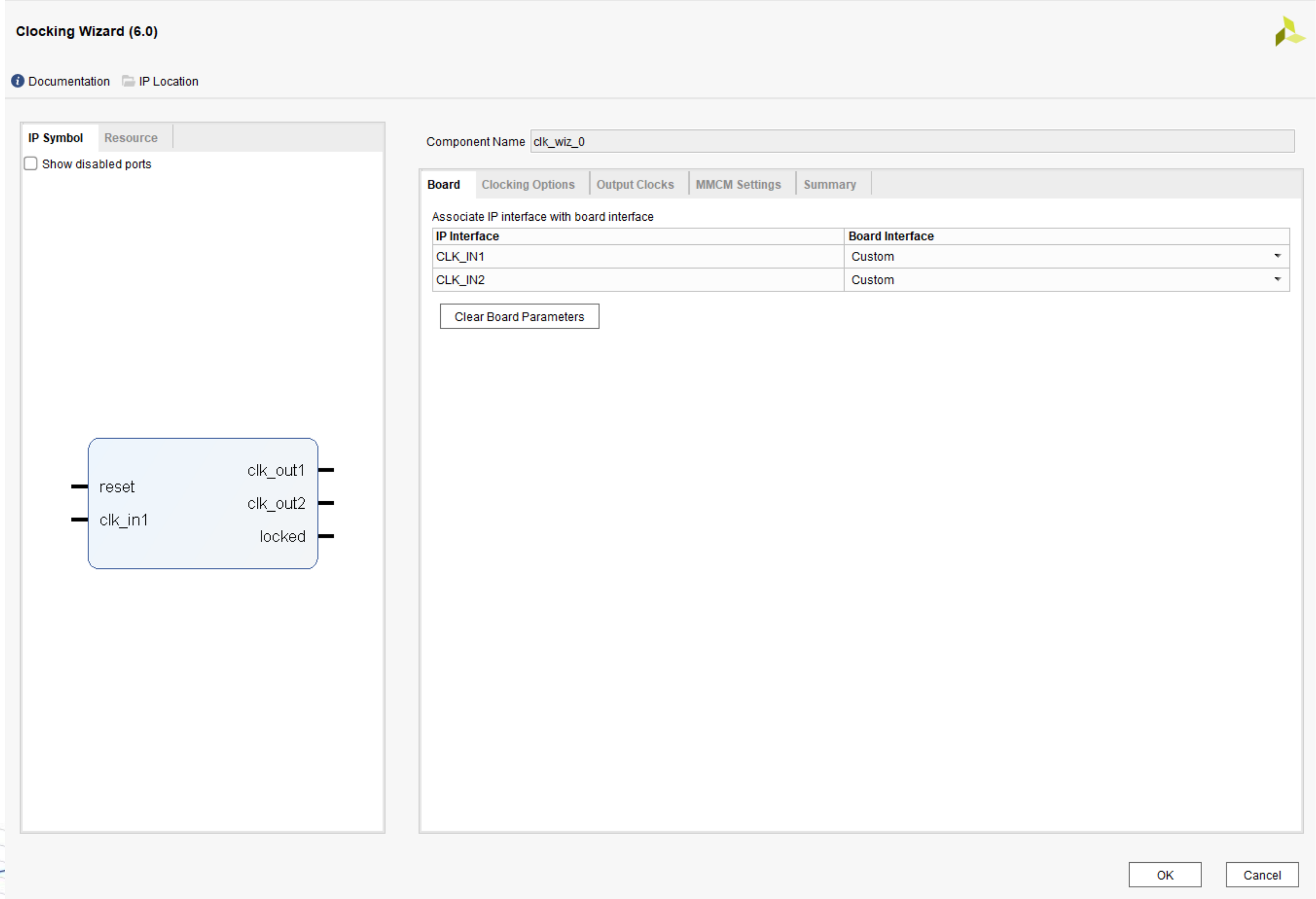
List of useful IP Cores in Xilinx Catalog

Classification regarding the use, and level of understanding, in this course:

Fundamental	Uso Intensivo	Uso frecuente	Uso Opcional
CLOCK WIZARD	ATG	CONCAT	AXI QUAD SPI
SYSTEM RESET	AXI GPIO	SLICE	AXI I2C
ZYNQ	AXI TIMER	BINARY COUNTER	MDM (DEBUG)
AXI Interconnect	VIO		
INT Controller	ILA		
MICROBLAZE	BRAM		

IP Cores

Clock Wizard



☐ Show disabled ports

Component Name clk_wiz_0

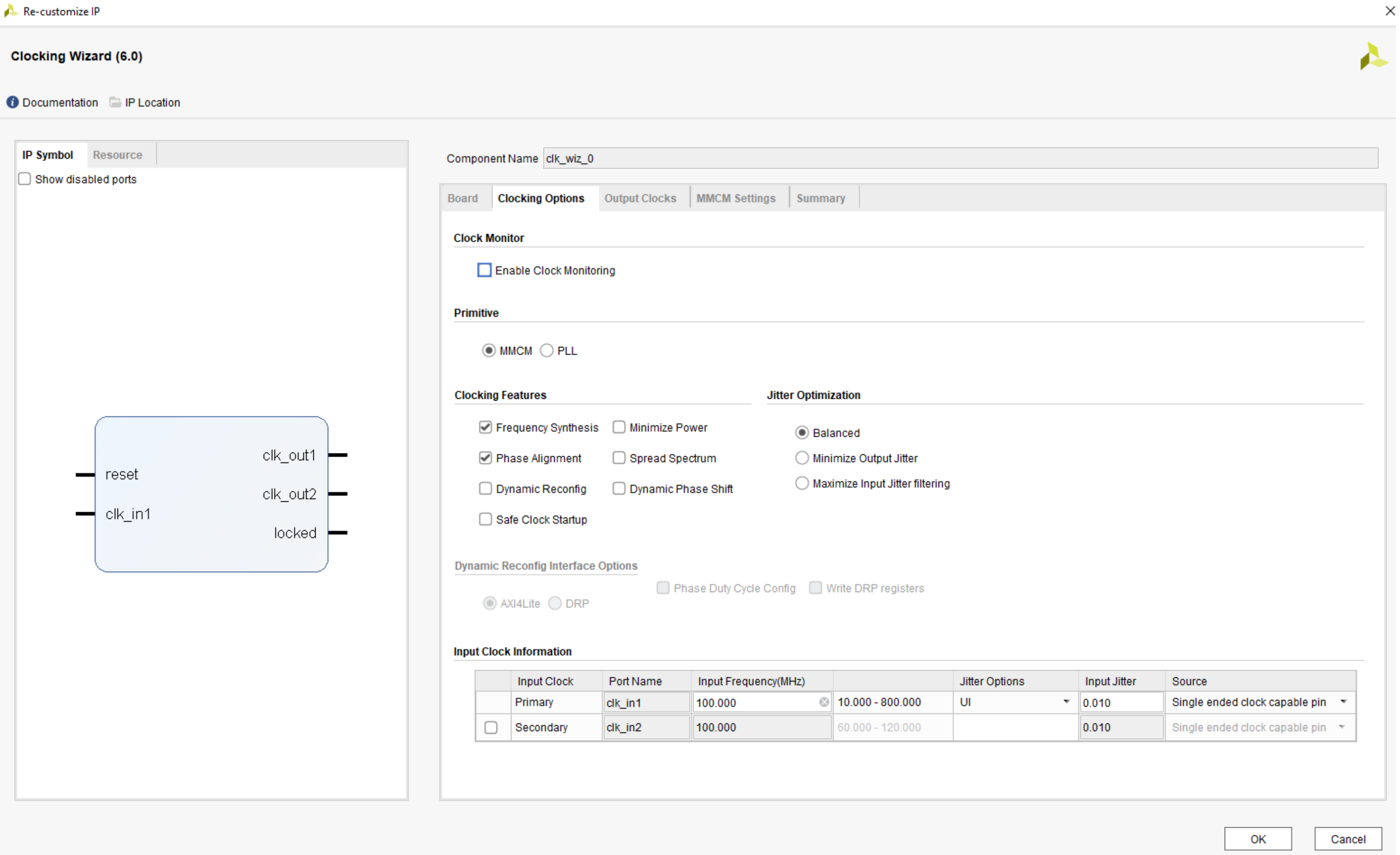
Associate IP interface with board interface

IP Interface	Board Interface
CLK_IN1	Custom
CLK_IN2	Custom

- Useful to create different output clocks.
- Accetp upt to two input clocks and 7 output clocks.

IP Cores

Clock Wizard



Re-customize IP

Clocking Wizard (6.0)

Documentation IP Location

IP Symbol | Resource

☐ Show disabled ports

Component Name clk_wiz_0

Board | Clocking Options | Output Clocks | MMCM Settings | Summary

Clock Monitor

☐ Enable Clock Monitoring

Primitive

● MMCM ○ PLL

Clocking Features

☑ Frequency Synthesis ☐ Minimize Power
☑ Phase Alignment ☐ Spread Spectrum
☐ Dynamic Reconfig ☐ Dynamic Phase Shift
☐ Safe Clock Startup

Jitter Optimization

● Balanced
○ Minimize Output Jitter
○ Maximize Input Jitter filtering

Dynamic Reconfig Interface Options

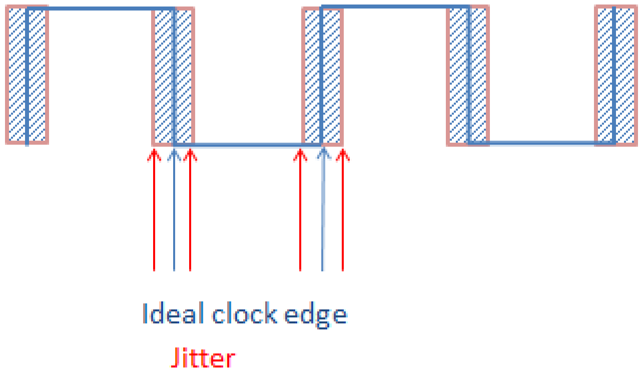
● AXI4Lite ○ DRP ☐ Phase Duty Cycle Config ☐ Write DRP registers

Input Clock Information

	Input Clock	Port Name	Input Frequency(MHz)		Jitter Options	Input Jitter	Source
	Primary	clk_in1	100.000	10.000 - 800.000	UI	0.010	Single ended clock capable pin
☐	Secondary	clk_in2	100.000	60.000 - 120.000		0.010	Single ended clock capable pin

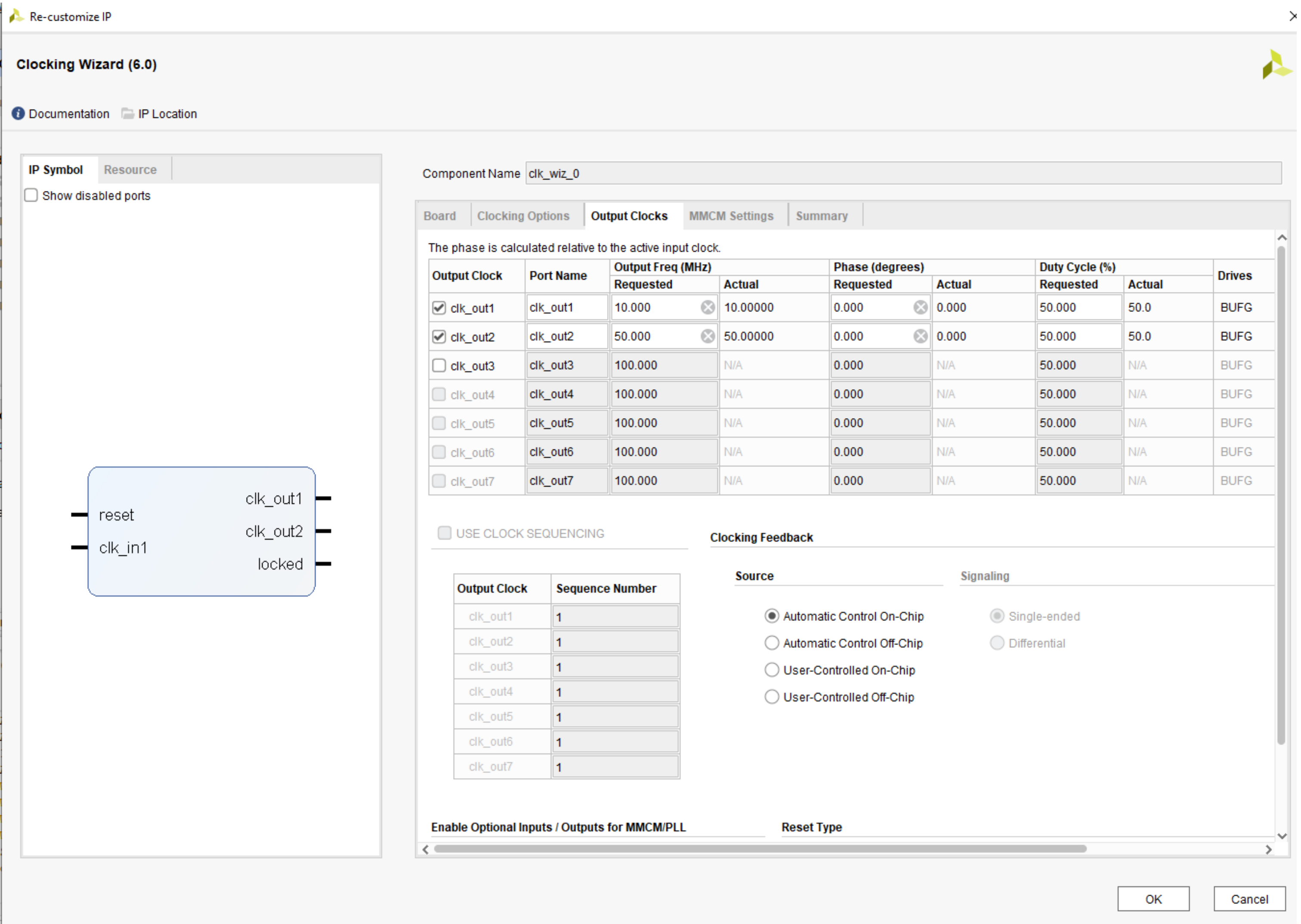
OK Cancel

- **MMCM**: Mixed-mode clock manager.
- **PLL**: Phase-lock-loop
- Both primitives are almost same. MMCM is a PLL with capability of phase adjustment.
- **Frequency Synthesis**: Allow different output frequencies.
- **Phase alignment**: Same phase for input and output clocks.
- **Dynamic**: Allow to set clock and phase dynamically through AXI.
- **Jitter**:



IP Cores

Clock Wizard



Re-customize IP

Clocking Wizard (6.0)

Documentation   IP Location

☐ Show disabled ports

Component Name clk_wiz_0

The phase is calculated relative to the active input clock.

Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)		Duty Cycle (%)		Drives
		Requested	Actual	Requested	Actual	Requested	Actual	
☑ clk_out1	clk_out1	10.000	10.00000	0.000	0.000	50.000	50.0	BUFG
☑ clk_out2	clk_out2	50.000	50.00000	0.000	0.000	50.000	50.0	BUFG
☐ clk_out3	clk_out3	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
☐ clk_out4	clk_out4	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
☐ clk_out5	clk_out5	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
☐ clk_out6	clk_out6	100.000	N/A	0.000	N/A	50.000	N/A	BUFG
☐ clk_out7	clk_out7	100.000	N/A	0.000	N/A	50.000	N/A	BUFG

☐ USE CLOCK SEQUENCING

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Clocking Feedback

Source
- ● Automatic Control On-Chip
- ○ Automatic Control Off-Chip
- ○ User-Controlled On-Chip
- ○ User-Controlled Off-Chip

Signaling
- ● Single-ended
- ○ Differential

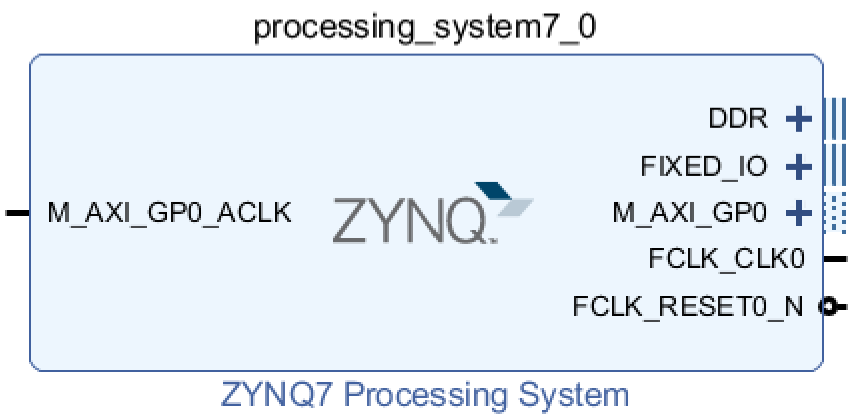
Enable Optional Inputs / Outputs for MMCM/PLL

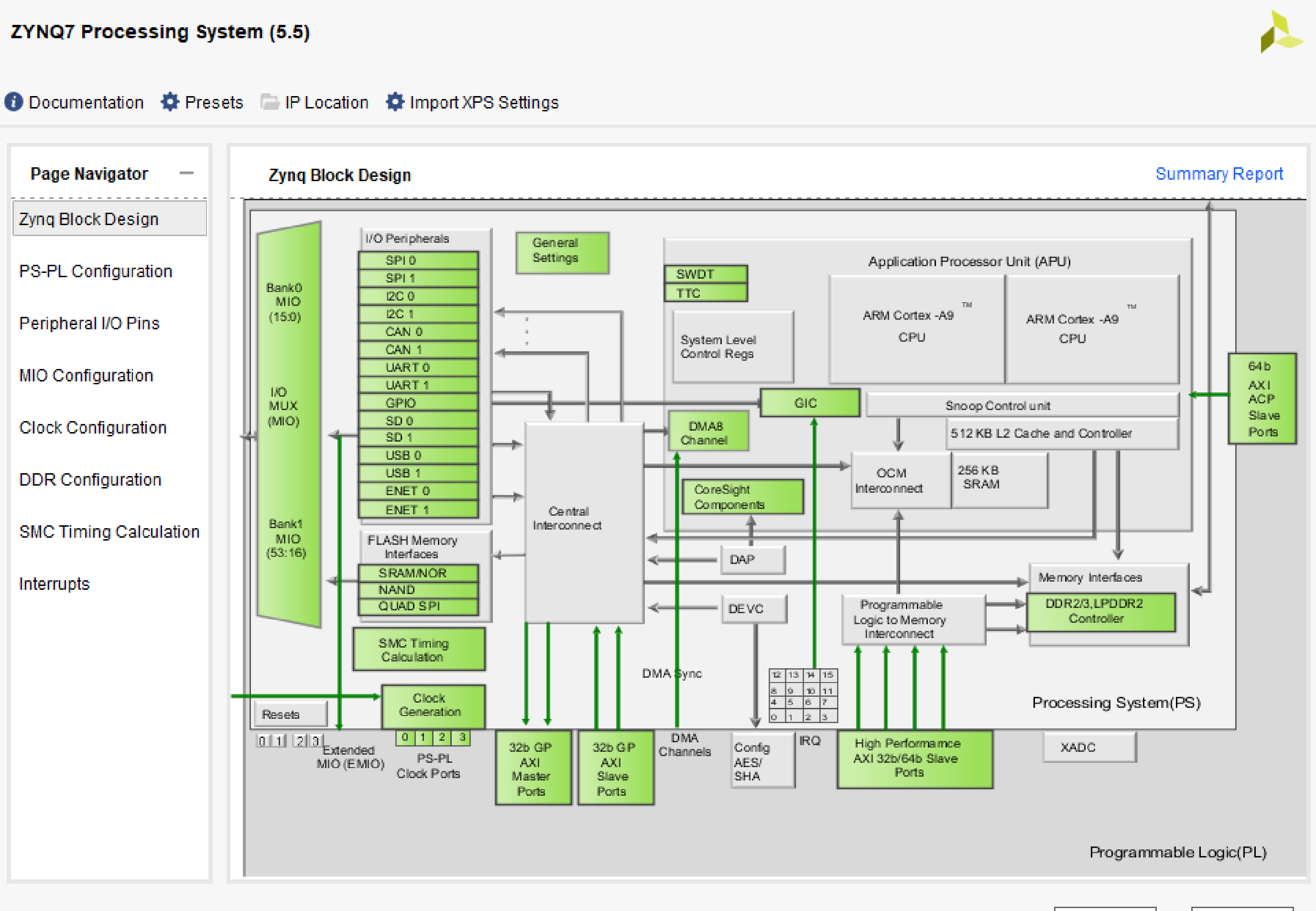
Reset Type

- **Here** we can configure our output clocks.
- We can configure duty cycle.

IP Cores

ZYNQ

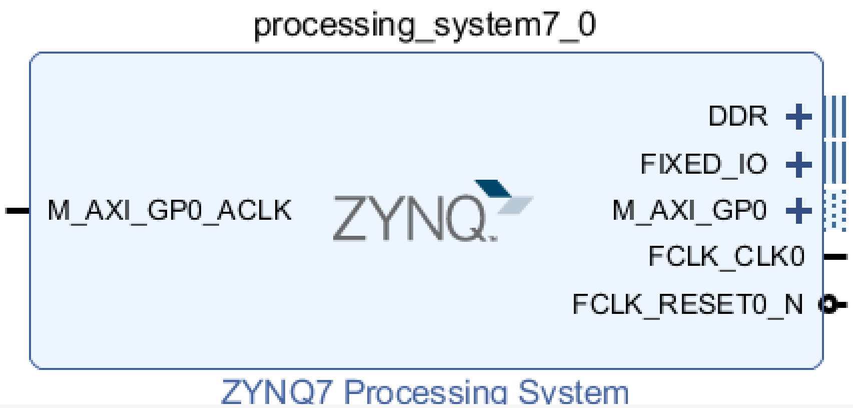


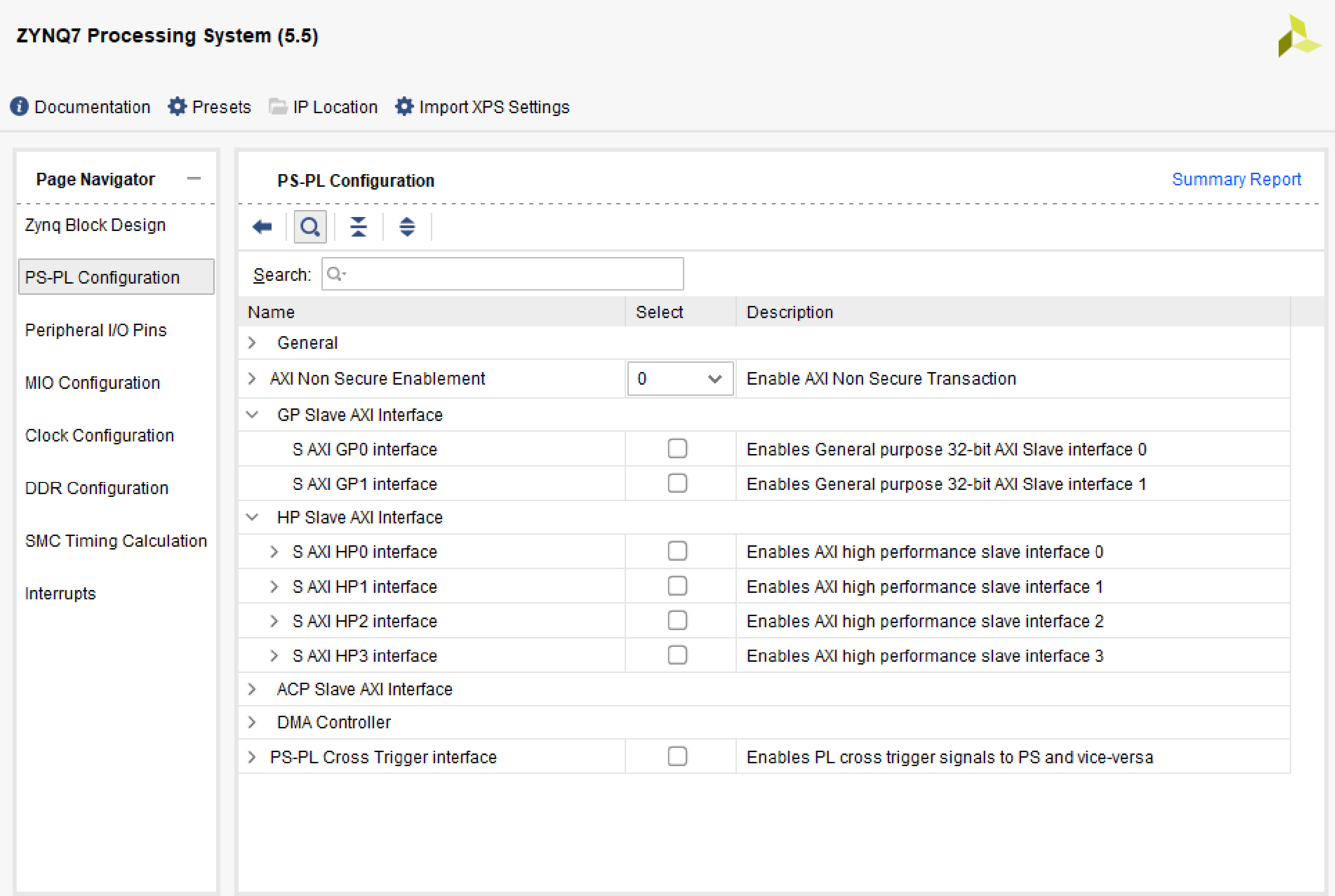


- One of the most important IP Cores
- It enables the configuration of the PS.
- It has 8 layers of configuration.
- Easy to configure just by clicking.
- Green blocks are configurable. However, it contains all PS and PL-PS communication channels.

IP Cores

ZYNQ





ZYNQ7 Processing System (5.5)

Page Navigator
- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

PS-PL Configuration

Search:

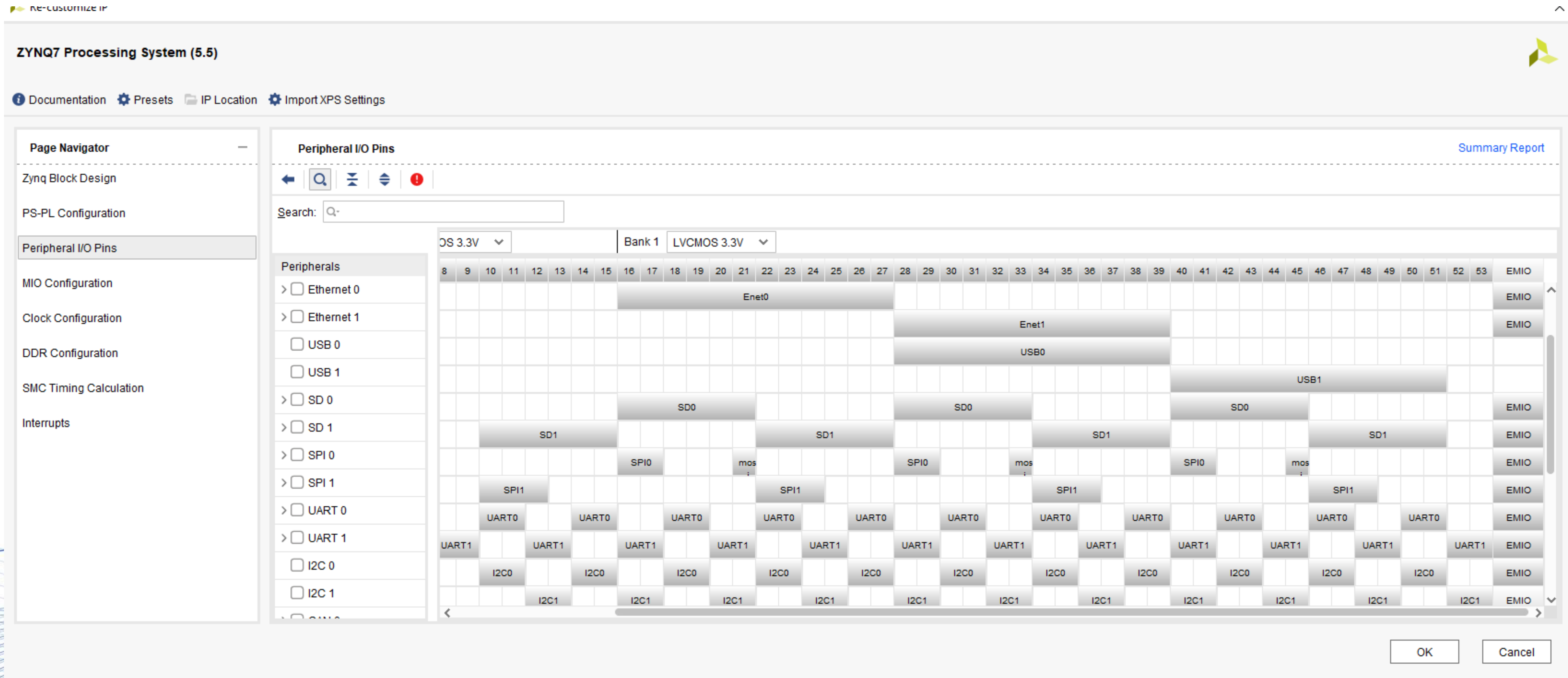
Name	Select	Description
› General		
› AXI Non Secure Enablement	0	Enable AXI Non Secure Transaction
⌄ GP Slave AXI Interface		
S AXI GP0 interface	□	Enables General purpose 32-bit AXI Slave interface 0
S AXI GP1 interface	□	Enables General purpose 32-bit AXI Slave interface 1
⌄ HP Slave AXI Interface		
› S AXI HP0 interface	□	Enables AXI high performance slave interface 0
› S AXI HP1 interface	□	Enables AXI high performance slave interface 1
› S AXI HP2 interface	□	Enables AXI high performance slave interface 2
› S AXI HP3 interface	□	Enables AXI high performance slave interface 3
› ACP Slave AXI Interface		
› DMA Controller		
› PS-PL Cross Trigger interface	□	Enables PL cross trigger signals to PS and vice-versa

- PS-PL Configuration allow us to enable the communication channels between PS and PL.
  - GP Master (Non secure enablement)
  - GP Slave
  - HP Slave
  - ACP Slave
  - General: EMIO
  - ACP

IP Cores

ZYNQ

- Peripherals I/o Pins can connect all peripherals to certain pins of the chip.
- This must match with the pins that the ZYBO Z7 is routed
- Peripheral can be also routed to EMIO. It means to PL.



Re-customize IP

ZYNQ7 Processing System (5.5)

Page Navigator

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

Peripheral I/O Pins

Summary Report

Search:

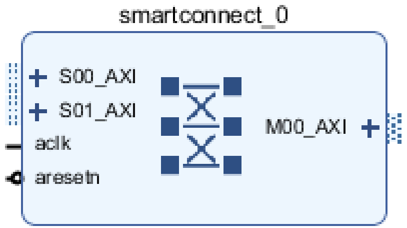
OS 3.3V

Bank 1 LVCMOS 3.3V

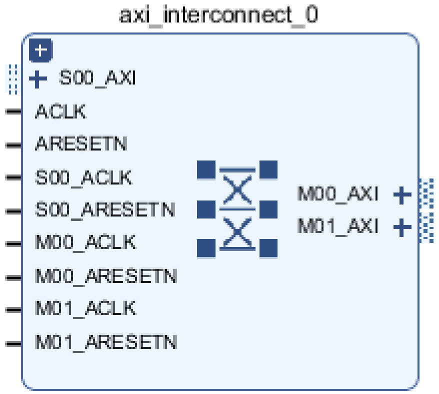
Peripherals	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	EMIO
□ Ethernet 0									Enet0																																						EMIO
□ Ethernet 1																					Enet1																										EMIO
□ USB 0																					USB0																										
□ USB 1																																	USB1														
□ SD 0									SD0												SD0												SD0														EMIO
□ SD 1			SD1												SD1												SD1												SD1								EMIO
□ SPI 0									SPI0					mos							SPI0					mos							SPI0					mos									EMIO
□ SPI 1			SPI1												SPI1												SPI1												SPI1								EMIO
□ UART 0			UART0				UART0				UART0				UART0				UART0				UART0				UART0				UART0				UART0				UART0				UART0				EMIO
□ UART 1	UART1				UART1				UART1				UART1				UART1				UART1				UART1				UART1				UART1				UART1				UART1				UART1		EMIO
□ I2C 0			I2C0				I2C0				I2C0				I2C0				I2C0				I2C0				I2C0				I2C0				I2C0				I2C0				I2C0				EMIO
□ I2C 1					I2C1				I2C1				I2C1				I2C1				I2C1				I2C1				I2C1				I2C1				I2C1				I2C1				I2C1		EMIO

IP Cores

AXI Interconnect



AXI SmartConnect

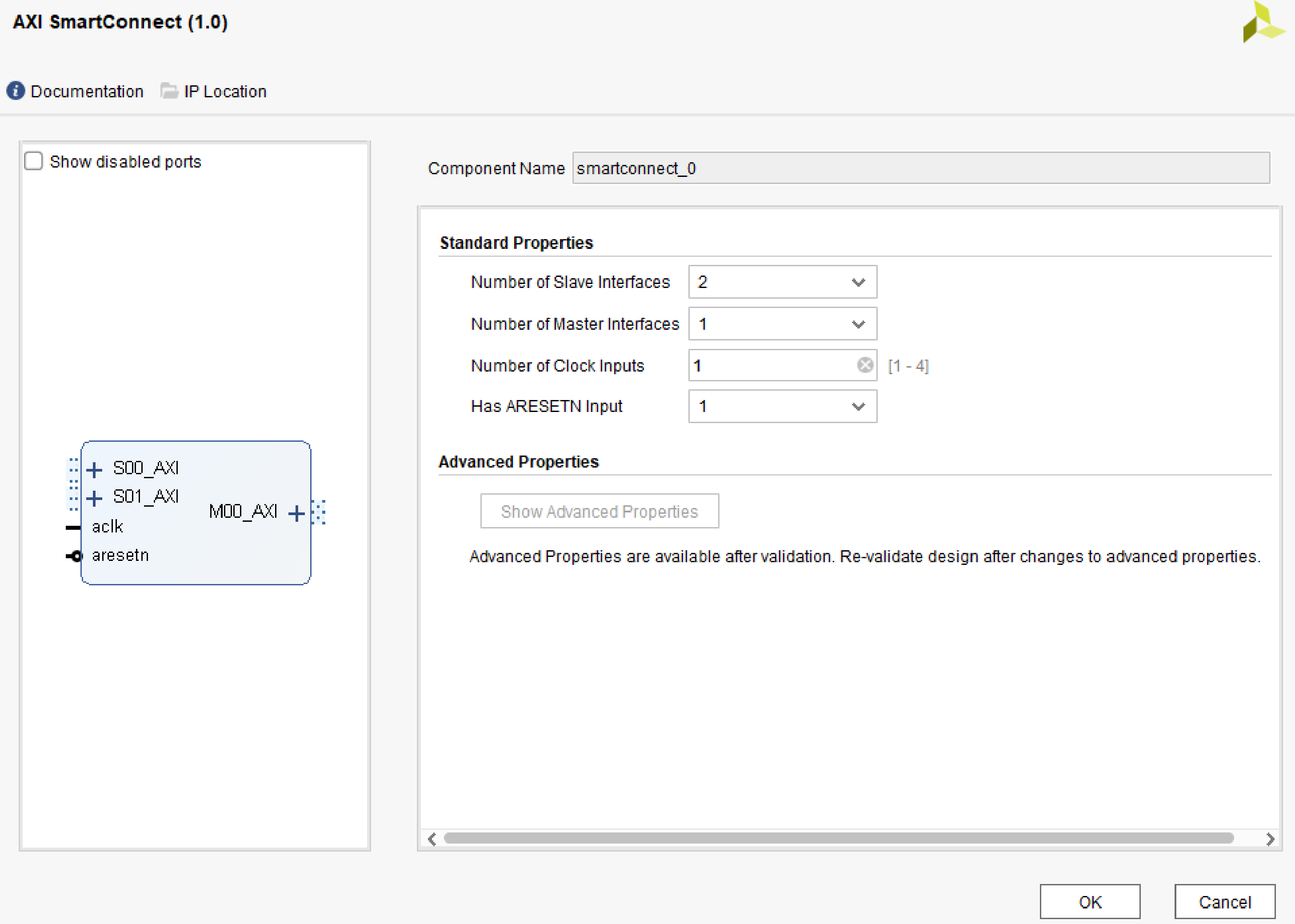


AXI Interconnect

- AXI Interconnect and Smart interconnect has the same function. It is an Ip core which plays the role of a “node” of different master and slaves AXI channels.
- Smart Interconnect is a newer version of AXI interconnect. We should use it in new designs.
- The block Works automatically , user do no have to configure anything.

IP Cores

AXI Interconnect



AXI SmartConnect (1.0)

Documentation   IP Location

□ Show disabled ports

Component Name: smartconnect_0

Standard Properties

Number of Slave Interfaces: 2

Number of Master Interfaces: 1

Number of Clock Inputs: 1 [1 - 4]

Has ARESETN Input: 1

Advanced Properties

Show Advanced Properties

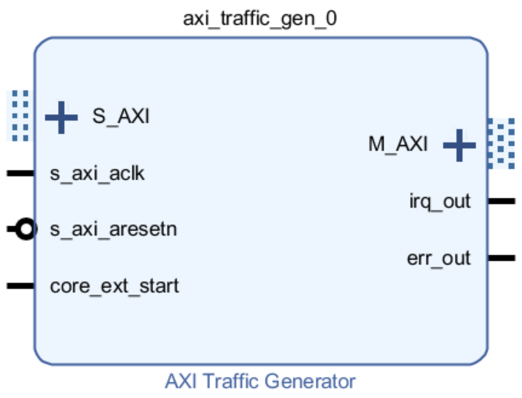
Advanced Properties are available after validation. Re-validate design after changes to advanced properties.

OK   Cancel

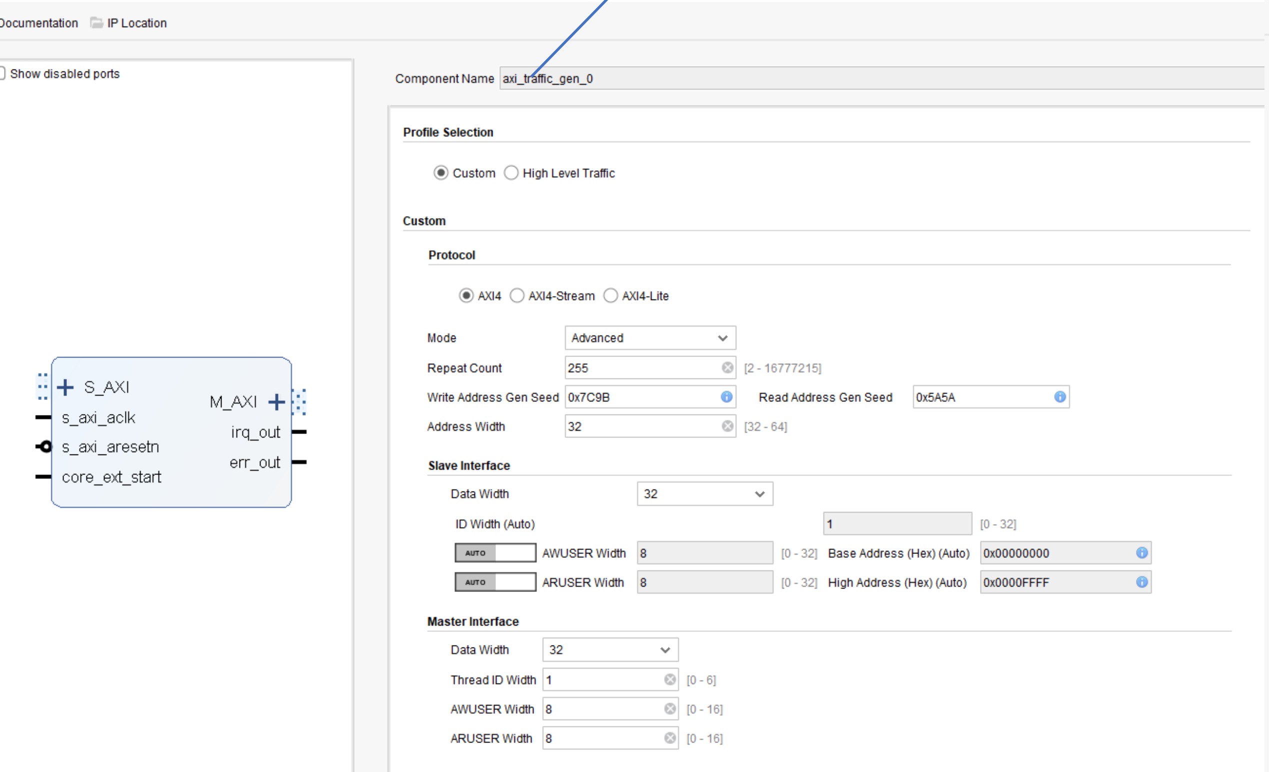
- We can easily configure number of slave and master interfaces
- We can also configure more than one clock, in case different AXI channels work at different clocks.

IP Cores

ATG



- This is a very well studied IP Core in previous lecture.
- Its main purpose it to easily generate AXI traffic in any of its versions:
  - AXI-Lite
  - AXI-Full
  - AXI-Steam
- It can be also configured for high-traffic data, such as: ethernet, PCIe, USB.



□ Show disabled ports

Component Name axi_traffic_gen_0

Profile Selection

● Custom ○ High Level Traffic

Custom

Protocol

● AXI4 ○ AXI4-Stream ○ AXI4-Lite

Mode: Advanced

Repeat Count: 255 [2 - 16777215]

Write Address Gen Seed: 0x7C9B Read Address Gen Seed: 0x5A5A

Address Width: 32 [32 - 64]

Slave Interface

Data Width: 32

ID Width (Auto): 1 [0 - 32]

AUTO AWUSER Width: 8 [0 - 32] Base Address (Hex) (Auto): 0x00000000

AUTO ARUSER Width: 8 [0 - 32] High Address (Hex) (Auto): 0x0000FFFF

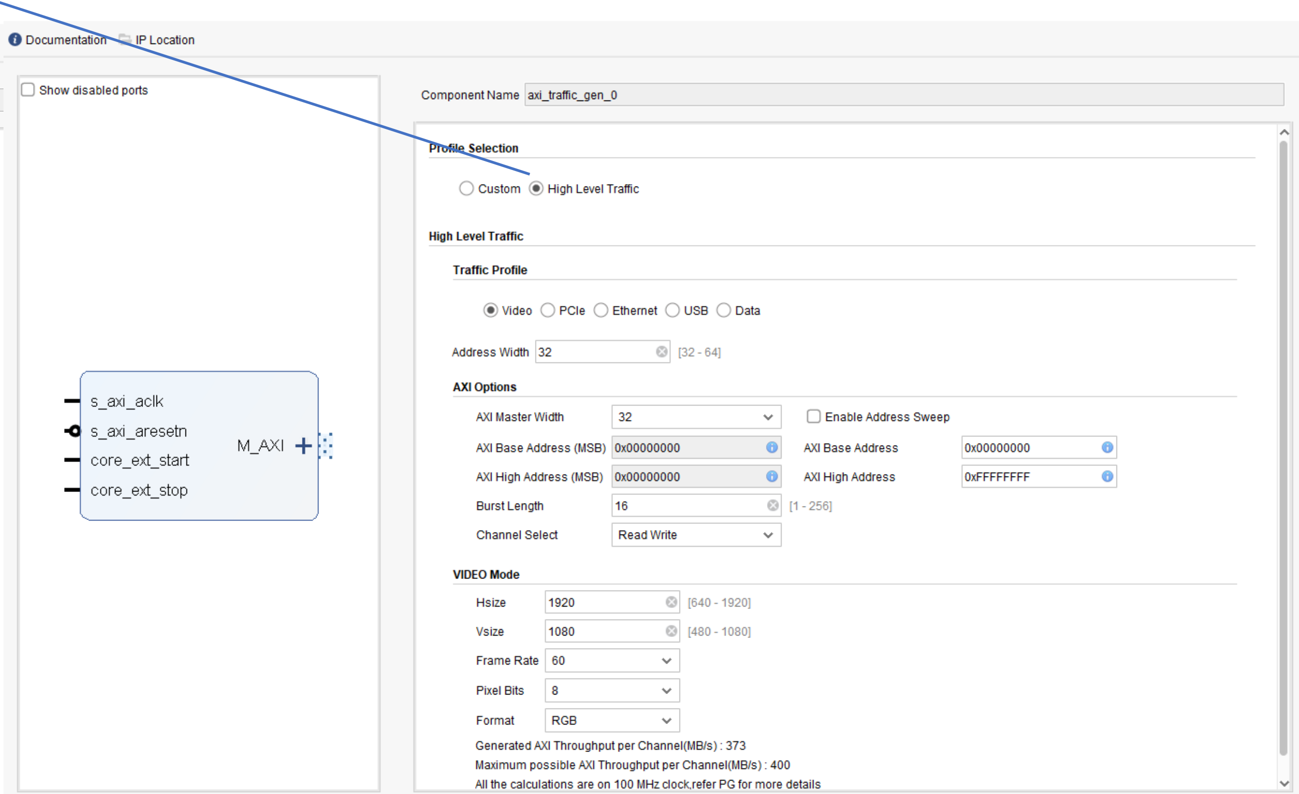
Master Interface

Data Width: 32

Thread ID Width: 1 [0 - 6]

AWUSER Width: 8 [0 - 16]

ARUSER Width: 8 [0 - 16]



□ Show disabled ports

Component Name axi_traffic_gen_0

Profile Selection

○ Custom ● High Level Traffic

High Level Traffic

Traffic Profile

● Video ○ PCIe ○ Ethernet ○ USB ○ Data

Address Width: 32 [32 - 64]

AXI Options

AXI Master Width: 32 □ Enable Address Sweep

AXI Base Address (MSB): 0x00000000 AXI Base Address: 0x00000000

AXI High Address (MSB): 0x00000000 AXI High Address: 0xFFFFFFFF

Burst Length: 16 [1 - 256]

Channel Select: Read Write

VIDEO Mode

Hsize: 1920 [640 - 1920]

Vsize: 1080 [480 - 1080]

Frame Rate: 60

Pixel Bits: 8

Format: RGB

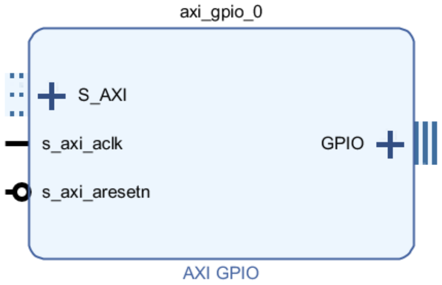
Generated AXI Throughput per Channel(MB/s) : 373

Maximum possible AXI Throughput per Channel(MB/s) : 400

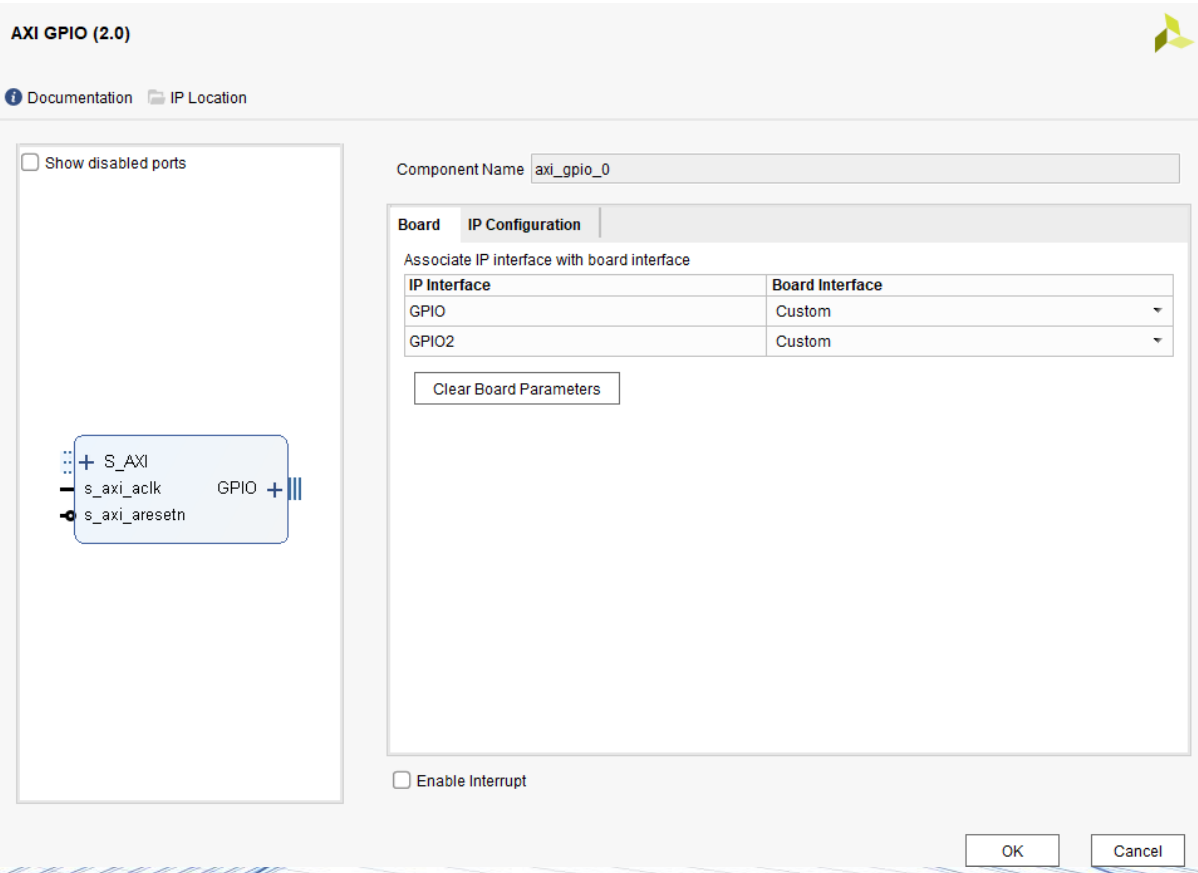
All the calculations are on 100 MHz clock,refer PG for more details

IP Cores

AXI GPIO



- It is an Ip Core that has AXI Lite protocol as input.
- It provides two channels of GPIO of up to 32 bits each.
- I/O signals can be configured as input or outputs.
- The ports are configured dynamically for input or output by enabling or disabling the 3-state buffer.
- The channels can be configured to generate an interrupt when a transition on any of their inputs occurs



AXI GPIO (2.0)

Documentation · IP Location

□ Show disabled ports

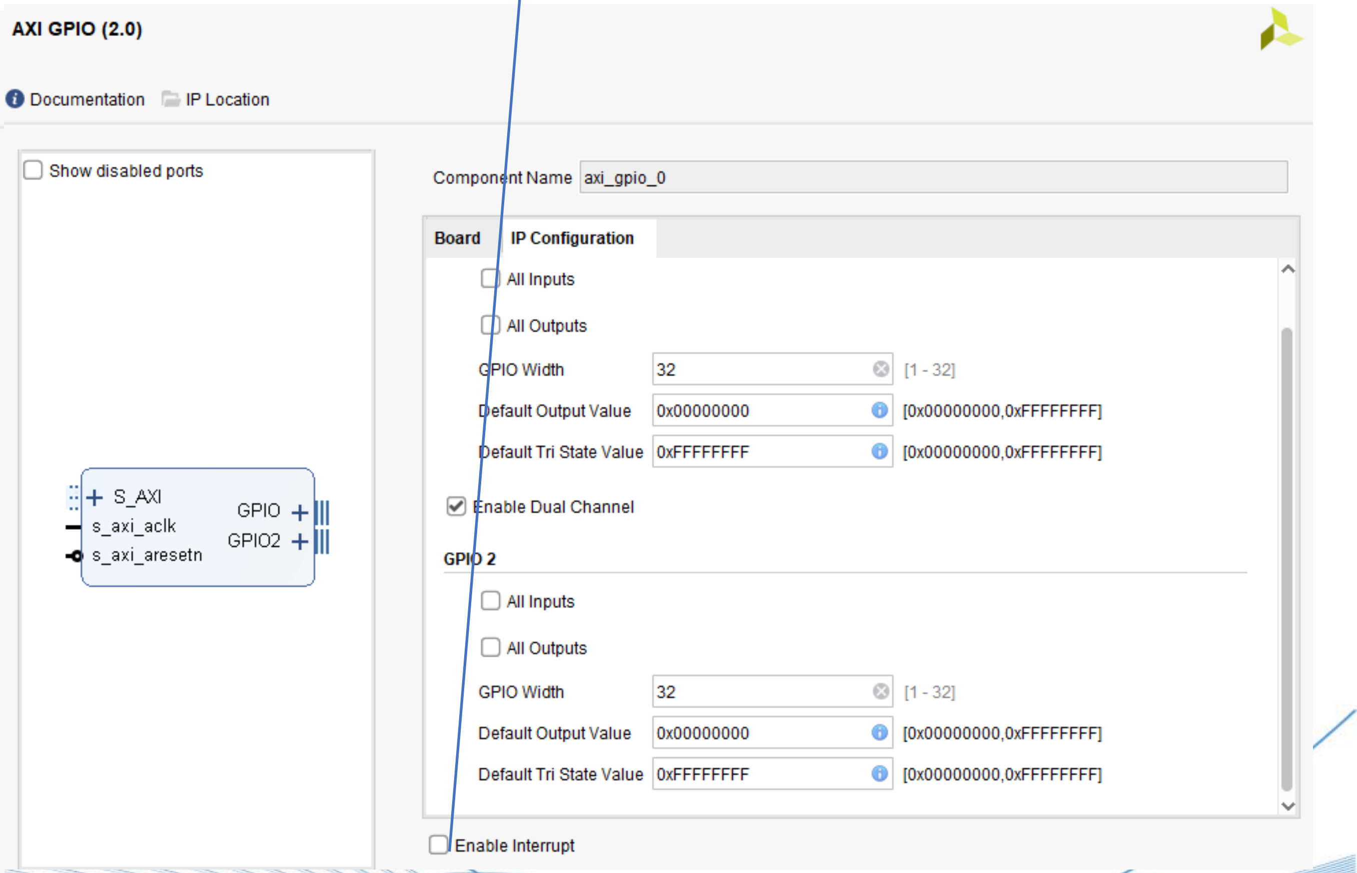
Component Name: axi_gpio_0

Board | IP Configuration

Associate IP interface with board interface

IP Interface	Board Interface
GPIO	Custom
GPIO2	Custom

□ Enable Interrupt



AXI GPIO (2.0)

Documentation · IP Location

□ Show disabled ports

Component Name: axi_gpio_0

Board | IP Configuration

□ All Inputs

□ All Outputs

GPIO Width	32	[1 - 32]
Default Output Value	0x00000000	[0x00000000,0xFFFFFFFF]
Default Tri State Value	0xFFFFFFFF	[0x00000000,0xFFFFFFFF]

☑ Enable Dual Channel

GPIO 2

□ All Inputs

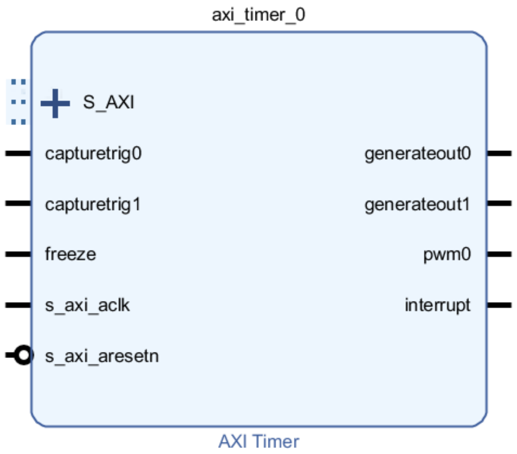
□ All Outputs

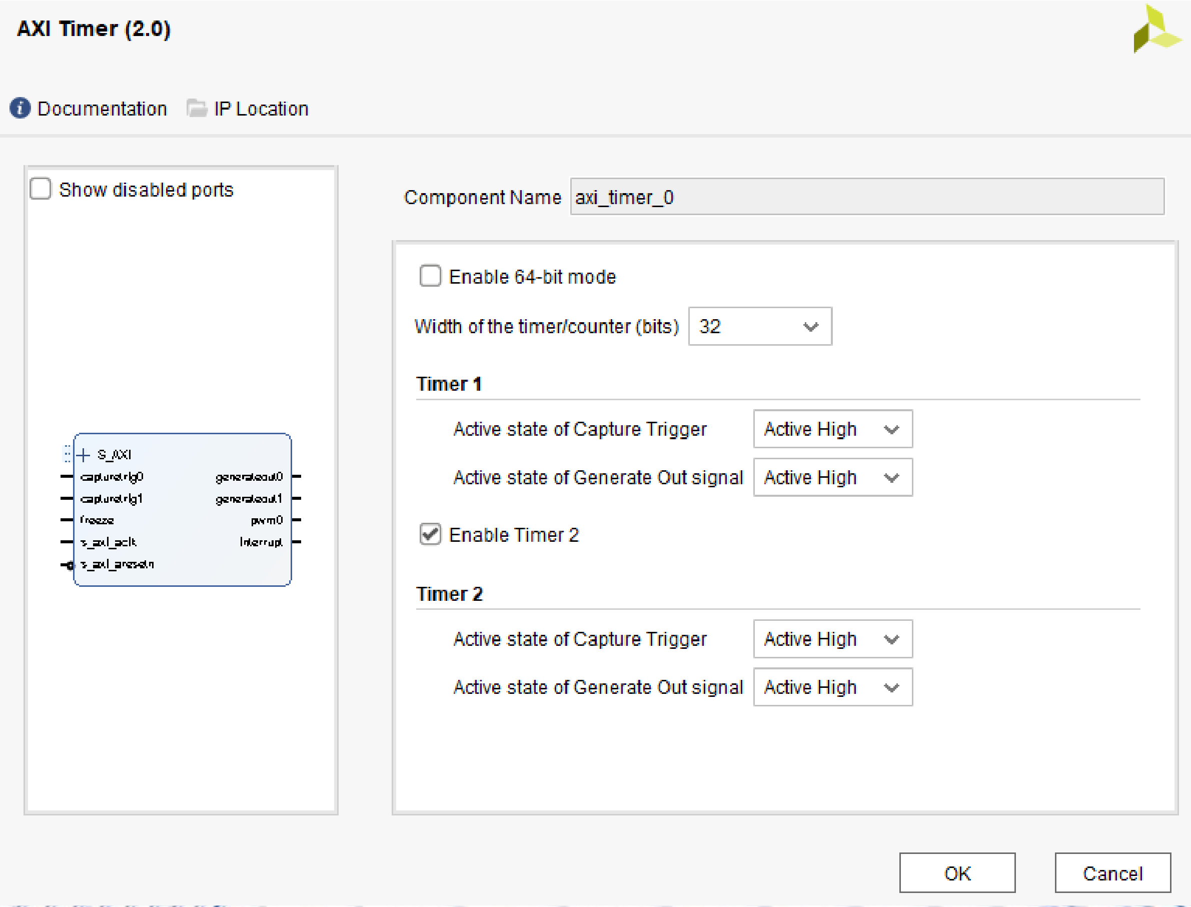
GPIO Width	32	[1 - 32]
Default Output Value	0x00000000	[0x00000000,0xFFFFFFFF]
Default Tri State Value	0xFFFFFFFF	[0x00000000,0xFFFFFFFF]

□ Enable Interrupt

IP Cores

AXI TIMER





AXI Timer (2.0)

□ Show disabled ports

Component Name: axi_timer_0

□ Enable 64-bit mode

Width of the timer/counter (bits): 32

Timer 1

Active state of Capture Trigger: Active High

Active state of Generate Out signal: Active High

☑ Enable Timer 2

Timer 2

Active state of Capture Trigger: Active High

Active state of Generate Out signal: Active High

- Very important and useful IP Core.
- Can be configured for up to two timers.
- It is configured through a AXI-Lite port
- It has a counter output for each time. “generateout0/1”
- It has also a PWM output signal
- It has also an interrupt signal when the counter is completed.
- It can be configured in 4 modes:
  - Generate Mode
  - Capture Mode
  - PWM Mode
  - Cascade Mode

IP Cores

AXI TIMER

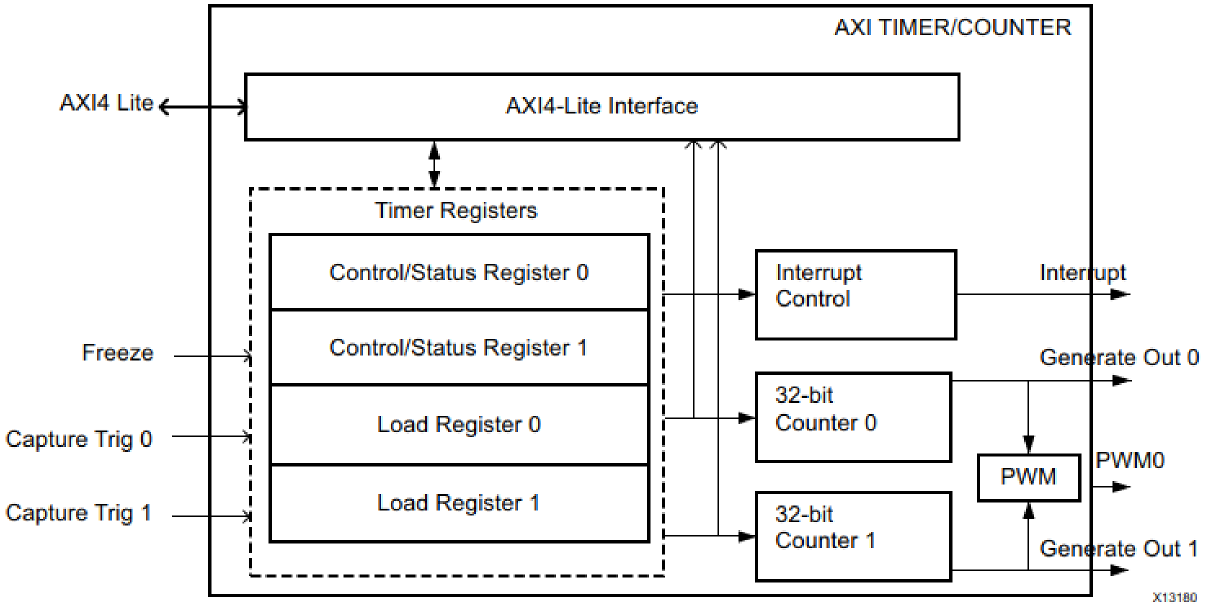
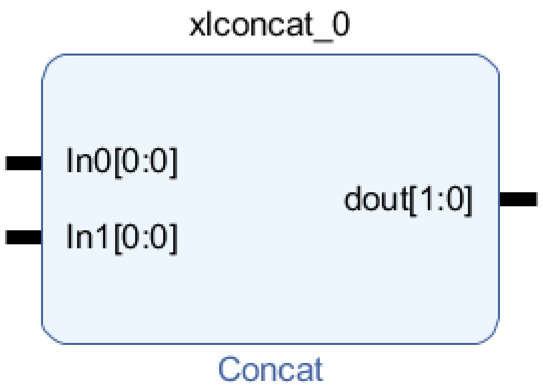
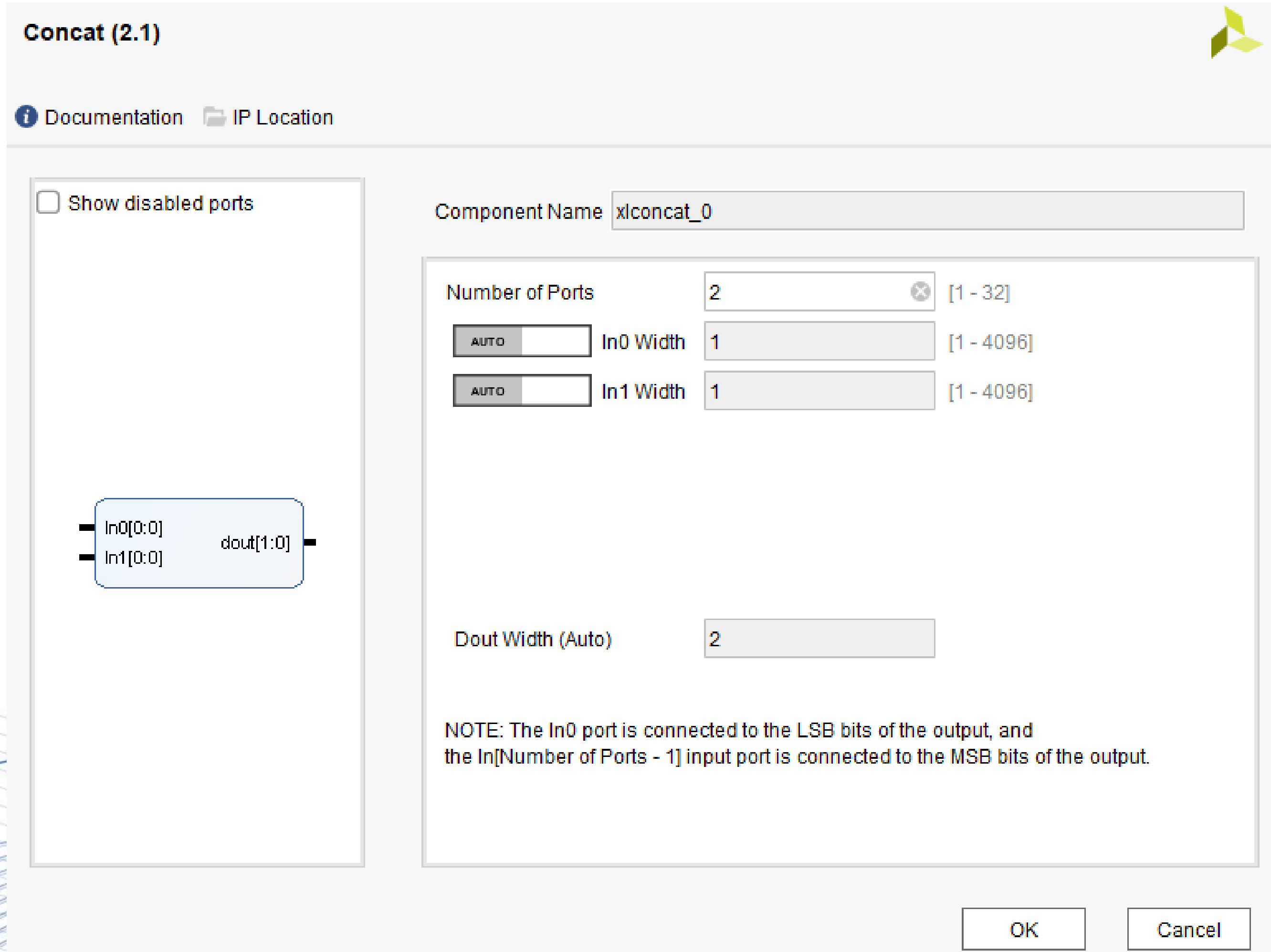


Figure 1-1: **Block Diagram of AXI Timer**

- Load register hold either the initial value for
- the counter for event generation or a capture value, depending on the mode of the timer.
- Generate Mode: load register is loaded into the counter. Finish the count and generate an interrupt (if enabled).
- Capture Mode: The counter is compared with capture and bit TINT is set when counter reach capture.
- PWM Mode: Both timers are used to create a PWM signal Timer 0 define period and Timer 1 set the high time for PWM0.
- Cascade Mode: Both 32 bit timers are used together to operate as 64bit timer.

IP Cores
CONCAT




Concat (2.1)

□ Show disabled ports

Component Name: xlconcat_0

Number of Ports: 2 [1 - 32]

AUTO In0 Width: 1 [1 - 4096]

AUTO In1 Width: 1 [1 - 4096]

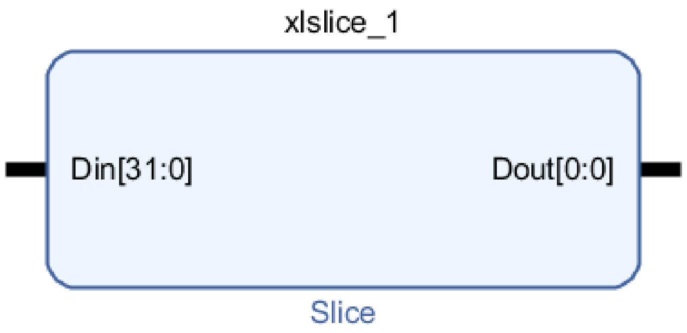
Dout Width (Auto): 2

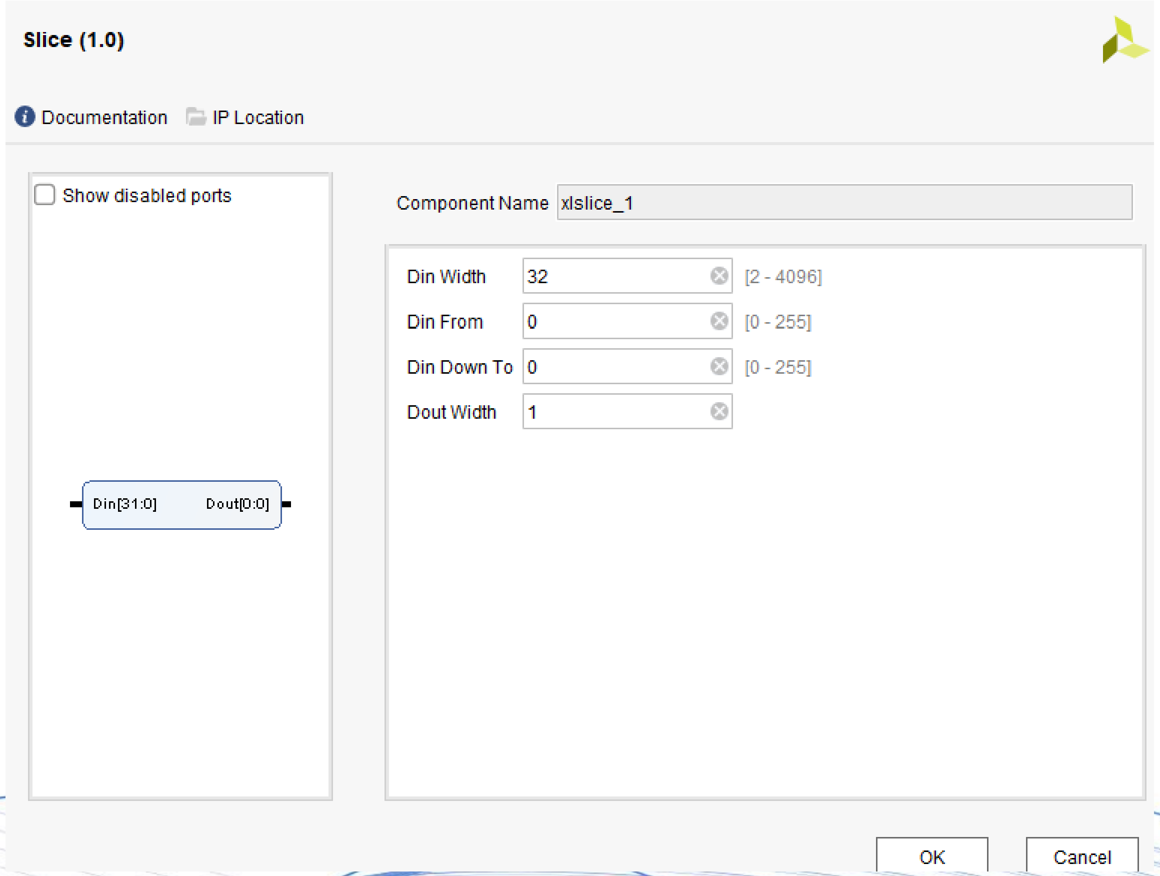
NOTE: The In0 port is connected to the LSB bits of the output, and
the In[Number of Ports - 1] input port is connected to the MSB bits of the output.

- Useful Ip Core to concatenate words
- Dout is same width as In0 width plus In2 width, plus…. In32.
- Up to 32 channels.

IP Cores

SLICE





Slice (1.0)

Documentation | IP Location

□ Show disabled ports

Component Name: xlslice_1

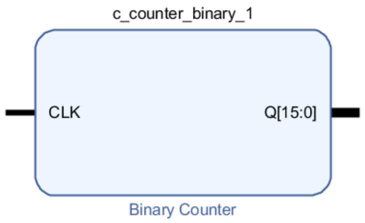
Din Width	32	[2 - 4096]
Din From	0	[0 - 255]
Din Down To	0	[0 - 255]
Dout Width	1	

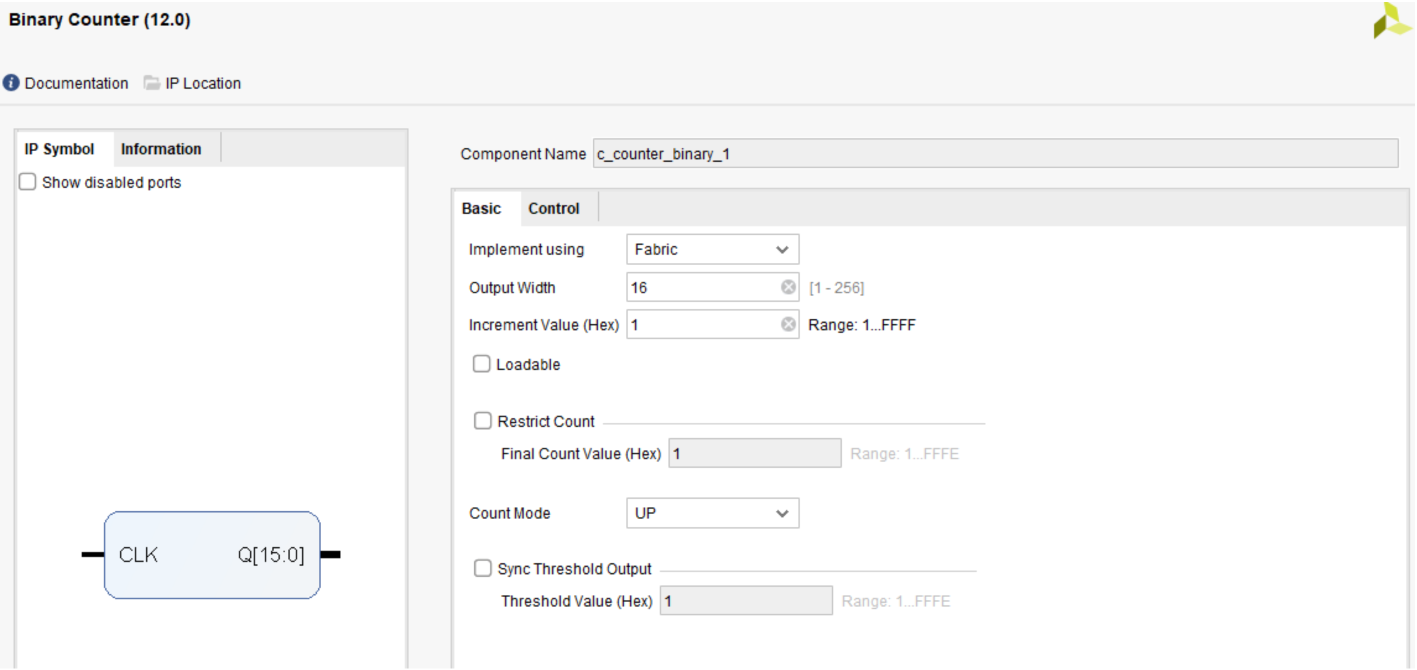
- Very useful Ip Core to select bits from a Word
- Bits can be selected as user define.
- From define the MSB and down to up to the LSB.

[1]

IP Cores

BINARY COUNTER





Binary Counter (12.0)

□ Show disabled ports

Component Name c_counter_binary_1

Implement using Fabric

Output Width 16 [1 - 256]

Increment Value (Hex) 1 Range: 1...FFFF

□ Loadable

□ Restrict Count

Final Count Value (Hex) 1 Range: 1...FFFE

Count Mode UP

□ Sync Threshold Output

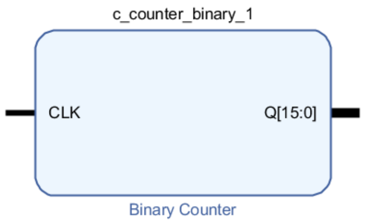
Threshold Value (Hex) 1 Range: 1...FFFE

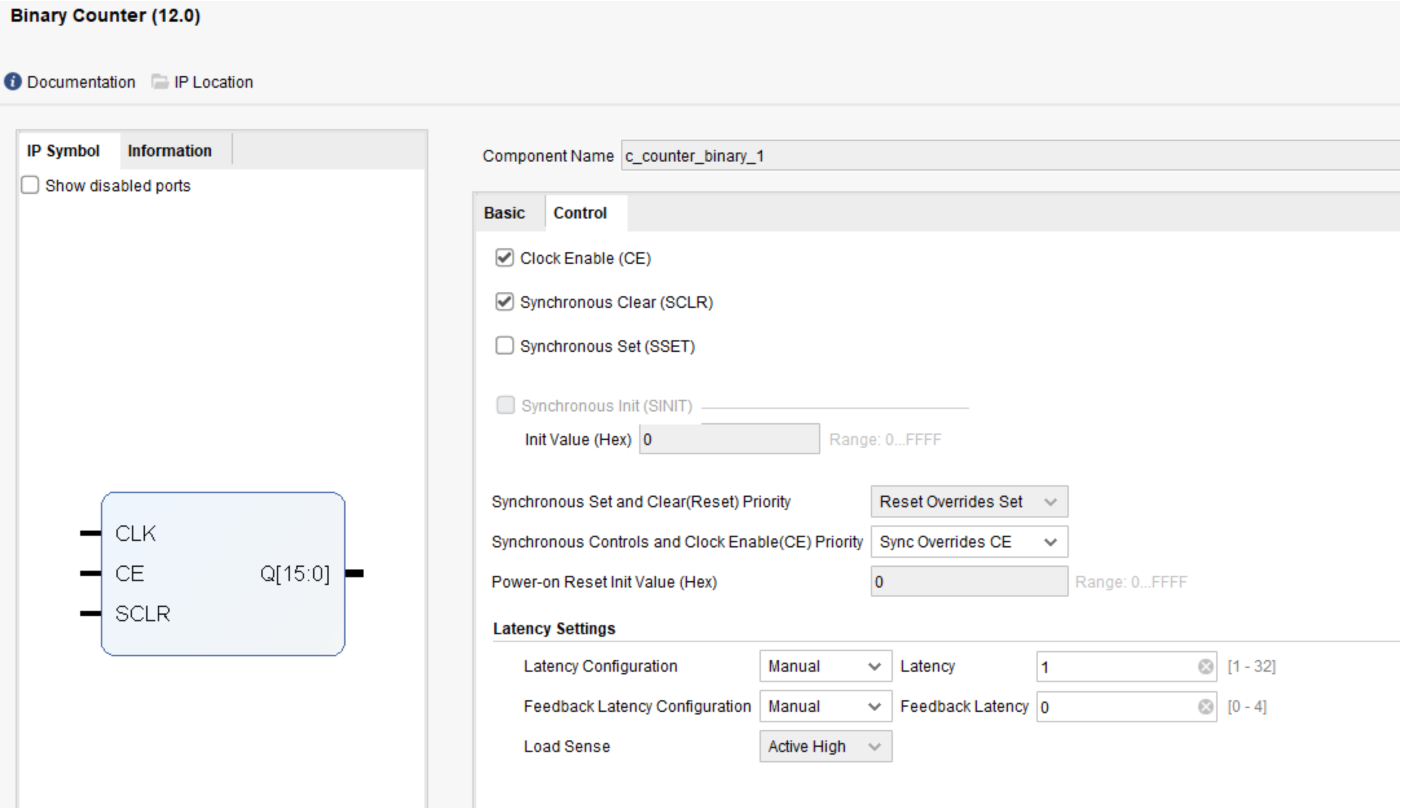
- A useful IP Core to count.
- Output width can be configurable upt to 256bit
- Increment value is configurable.
- Up or down count mode can be implemented

[1]

IP Cores

BINARY COUNTER





Binary Counter (12.0)

Documentation   IP Location

□ Show disabled ports

Component Name c_counter_binary_1

Basic | Control

☑ Clock Enable (CE)

☑ Synchronous Clear (SCLR)

□ Synchronous Set (SSET)

□ Synchronous Init (SINIT)

Init Value (Hex) 0 Range: 0...FFFF

Synchronous Set and Clear(Reset) Priority: Reset Overrides Set

Synchronous Controls and Clock Enable(CE) Priority: Sync Overrides CE

Power-on Reset Init Value (Hex): 0 Range: 0...FFFF

Latency Settings

Latency Configuration: Manual   Latency: 1 [1 - 32]

Feedback Latency Configuration: Manual   Feedback Latency: 0 [0 - 4]

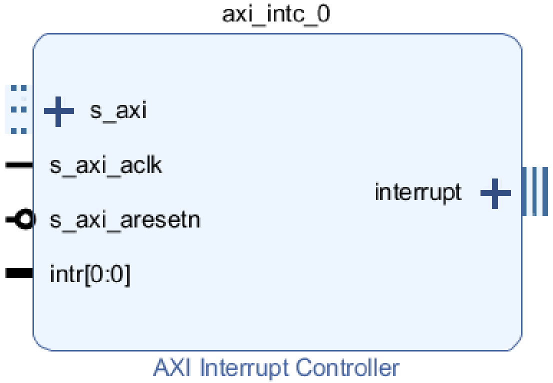
Load Sense: Active High

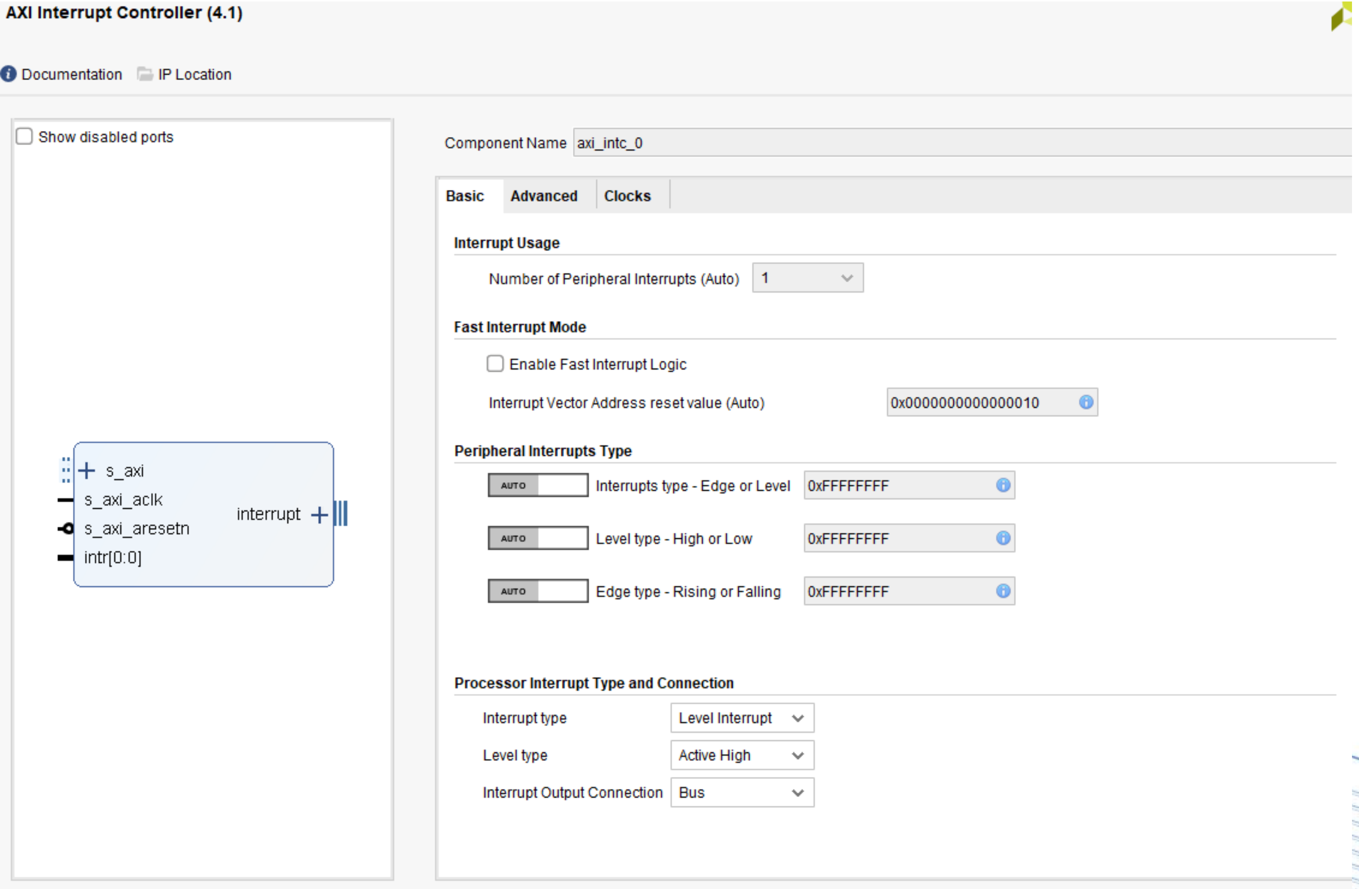
- Clock Enable allows external control
- Synchronous Clear enable clear the count.
- Synchronous Set: Forces output to High.

[1]

IP Cores

AXI Interrupt Controller




AXI Interrupt Controller (4.1)

Documentation   IP Location

□ Show disabled ports

Component Name axi_intc_0

Basic | Advanced | Clocks

Interrupt Usage

Number of Peripheral Interrupts (Auto) 1

Fast Interrupt Mode

□ Enable Fast Interrupt Logic

Interrupt Vector Address reset value (Auto) 0x0000000000000010

Peripheral Interrupts Type

AUTO Interrupts type - Edge or Level 0xFFFFFFFF

AUTO Level type - High or Low 0xFFFFFFFF

AUTO Edge type - Rising or Falling 0xFFFFFFFF

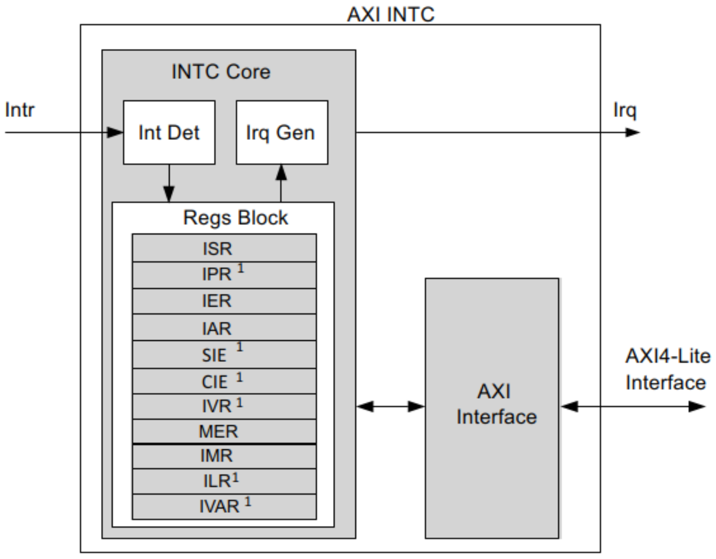
Processor Interrupt Type and Connection

Interrupt type Level Interrupt

Level type Active High

Interrupt Output Connection Bus

- This Ip Core is very important to handle interrupts.
- Receives multiple interrupt inputs from peripheral devices and merges them into an interrupt output to the system processor.
- It acknowledge and priories different interrupts from different peripherals.
- Registers are access by AXI-Lite

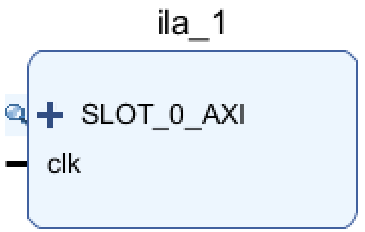


1. Registers are OPTIONAL

Figure 1-1: **AXI INTC Core Block Diagram**

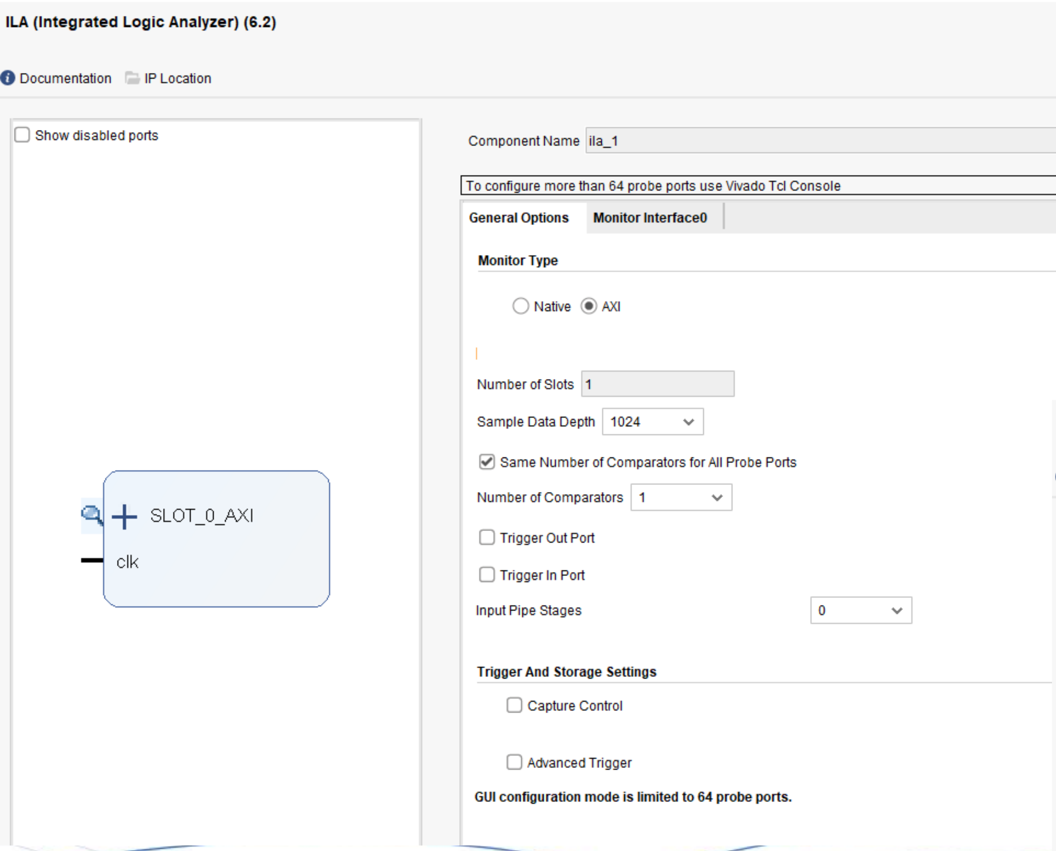
IP Cores

Integrated Logic Analyzer (ILA)



ILA (Integrated Logic Analyzer)

- This IP Core is very useful to debug our logic design.
- We can make the simile with an oscilloscope.
- It can be configured in two main modes:
  - Native: to check standard signals
  - AXI: to check AXI Transactions.
- Sample Data Depth: Number of samples
- We can choose between all AXI types to monitor.



ILA (Integrated Logic Analyzer) (6.2)

Documentation   IP Location

□ Show disabled ports

Component Name ila_1

To configure more than 64 probe ports use Vivado Tcl Console

General Options | Monitor Interface0

Monitor Type

○ Native ● AXI

Number of Slots 1

Sample Data Depth 1024

☑ Same Number of Comparators for All Probe Ports

Number of Comparators 1

□ Trigger Out Port

□ Trigger In Port

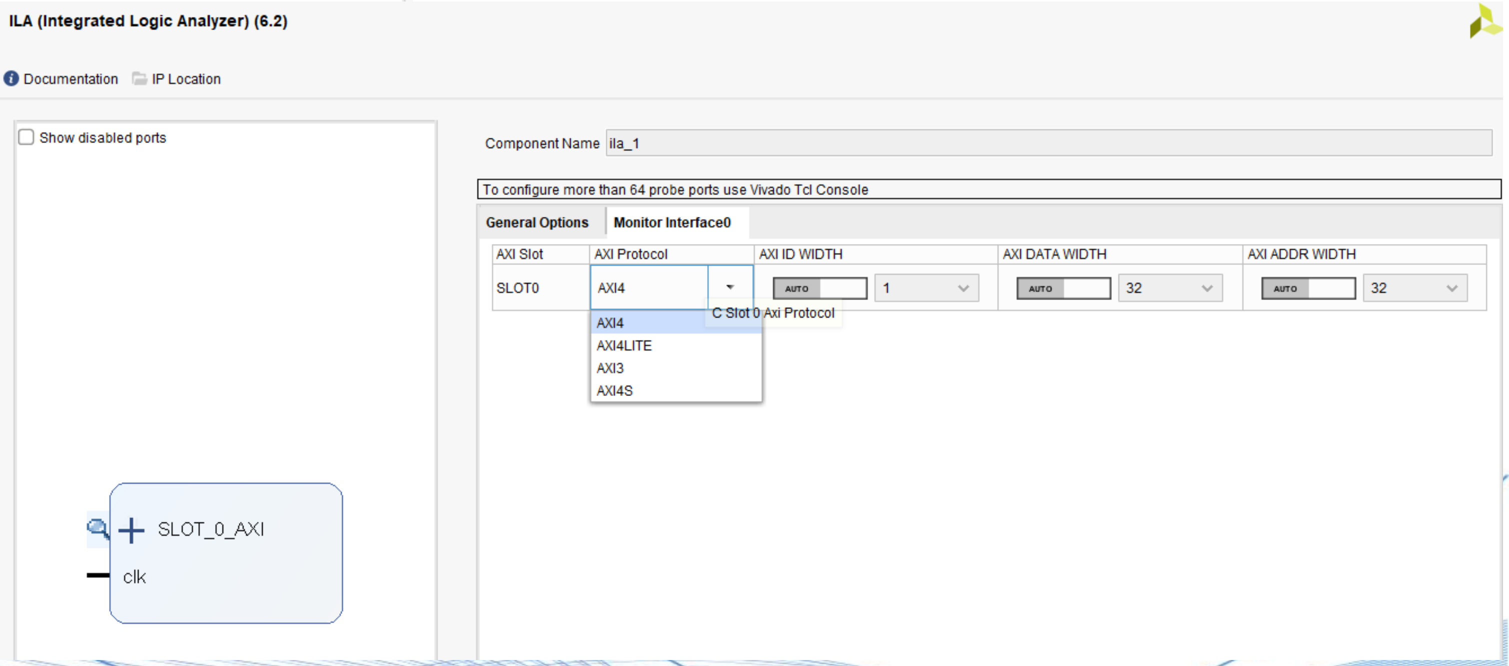
Input Pipe Stages 0

Trigger And Storage Settings

□ Capture Control

□ Advanced Trigger

GUI configuration mode is limited to 64 probe ports.



ILA (Integrated Logic Analyzer) (6.2)

Documentation   IP Location

□ Show disabled ports

Component Name ila_1

To configure more than 64 probe ports use Vivado Tcl Console

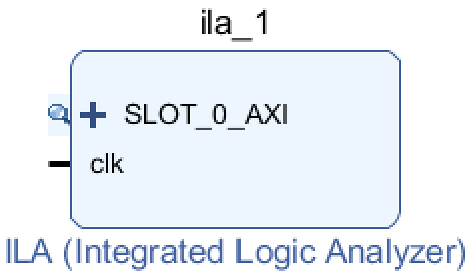
General Options | Monitor Interface0

AXI Slot	AXI Protocol	AXI ID WIDTH	AXI DATA WIDTH	AXI ADDR WIDTH
SLOT0	AXI4	AUTO 1	AUTO 32	AUTO 32

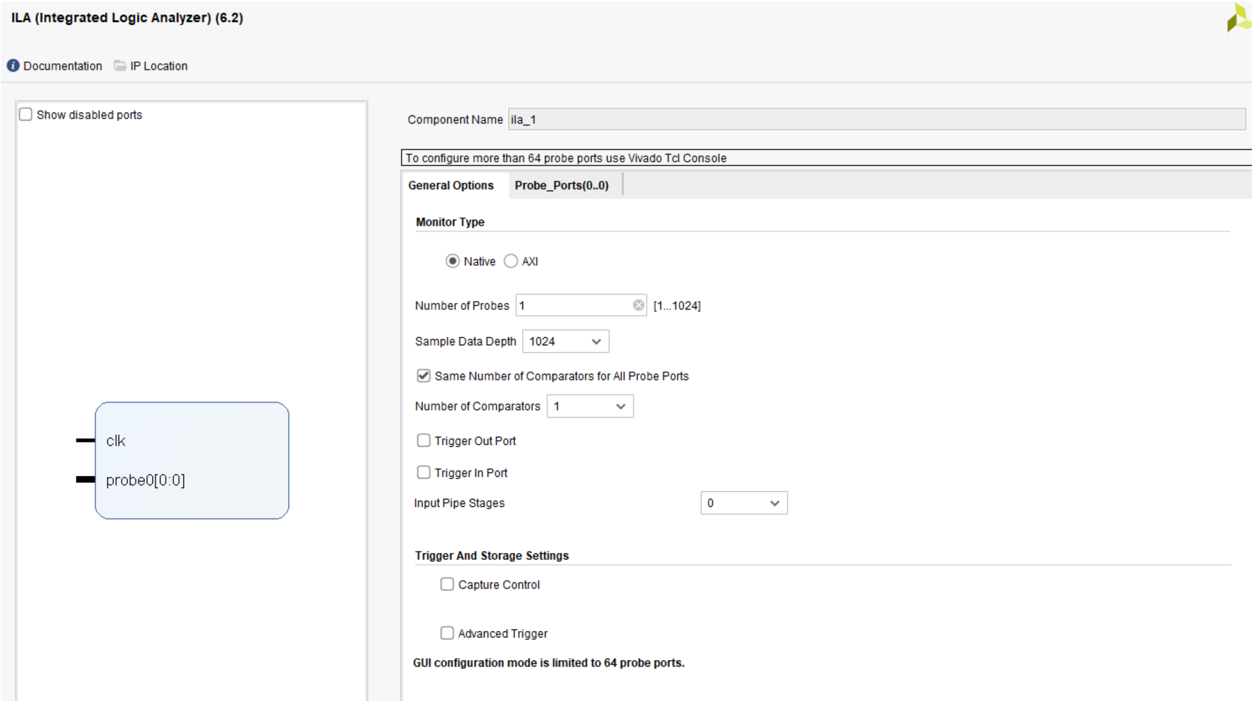
C Slot 0 Axi Protocol

IP Cores

Integrated Logic Analyzer (ILA)



- Up to 1024 probes.
- Depth up to 131Mbs
- TRIG_IN: It triggers ILA
- TRIG_OUT: Useful to trigger others ILA, activated by trigger _in or internal conditions.
- Probe with can be configured.
- Comparators: Numbers of possibles comparators for triggers.



ILA (Integrated Logic Analyzer) (6.2)

Documentation IP Location

□ Show disabled ports

Component Name ila_1

To configure more than 64 probe ports use Vivado Tcl Console

General Options | Probe_Ports(0..0)

Monitor Type

● Native ○ AXI

Number of Probes 1 [1...1024]

Sample Data Depth 1024

☑ Same Number of Comparators for All Probe Ports

Number of Comparators 1

□ Trigger Out Port

□ Trigger In Port

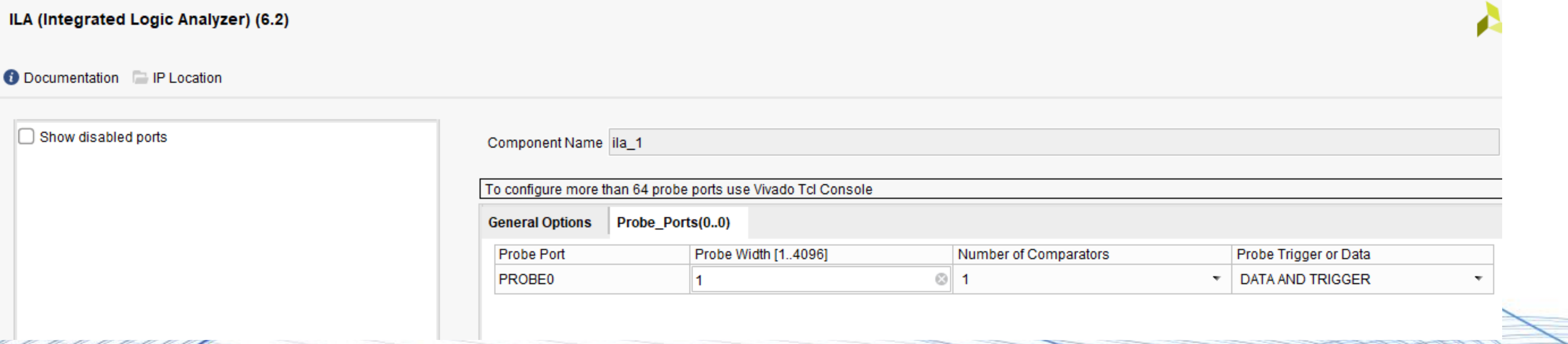
Input Pipe Stages 0

Trigger And Storage Settings

□ Capture Control

□ Advanced Trigger

GUI configuration mode is limited to 64 probe ports.



ILA (Integrated Logic Analyzer) (6.2)

Documentation IP Location

□ Show disabled ports

Component Name ila_1

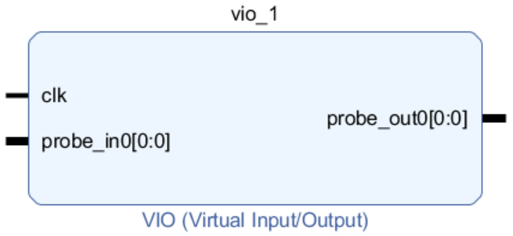
To configure more than 64 probe ports use Vivado Tcl Console

General Options | Probe_Ports(0..0)

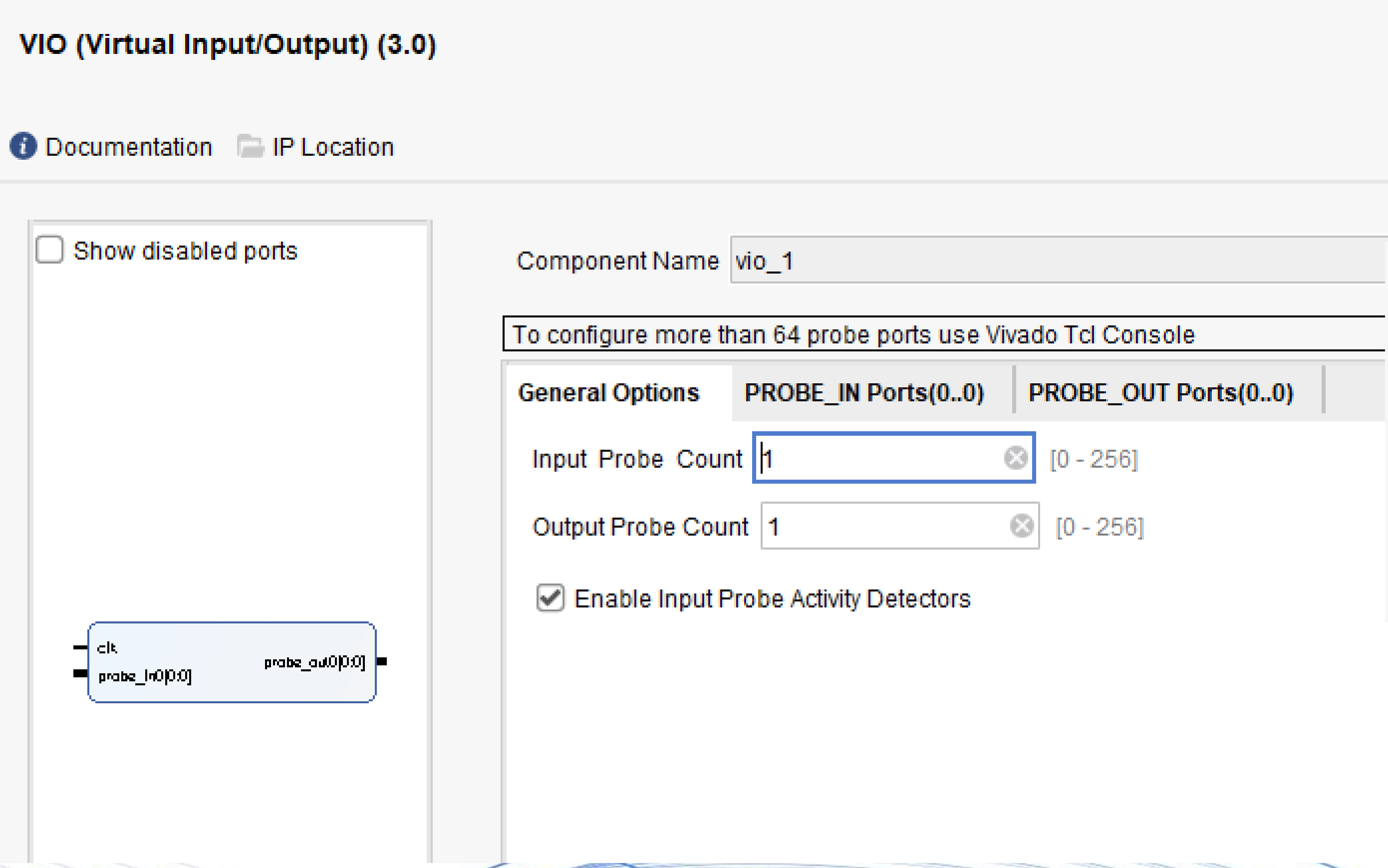
Probe Port	Probe Width [1..4096]	Number of Comparators	Probe Trigger or Data
PROBE0	1	1	DATA AND TRIGGER

IP Cores

Virtual Input/Output - VIO



- Similar to ILA , but instead of showing signals, it shows its numerical value.
- It enables the use of output probes.
- Output probes can be used to activate signals in real time.



VIO (Virtual Input/Output) (3.0)

☐ Show disabled ports

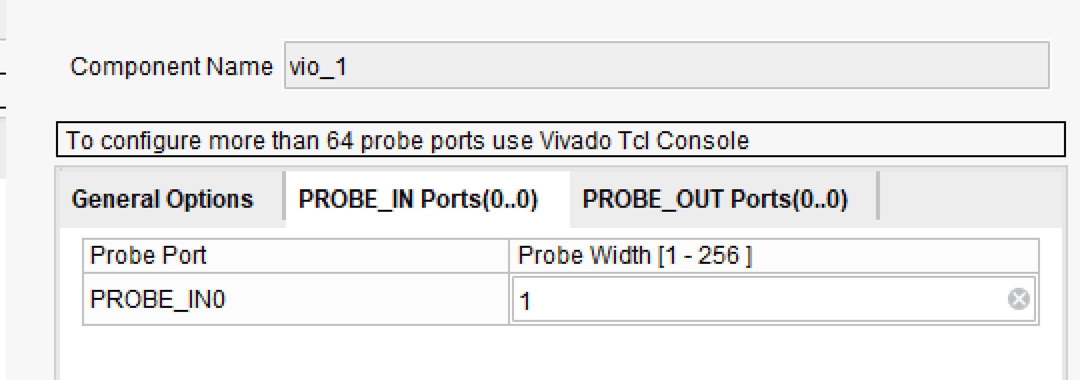
Component Name vio_1

To configure more than 64 probe ports use Vivado Tcl Console

Input Probe Count 1 [0 - 256]

Output Probe Count 1 [0 - 256]

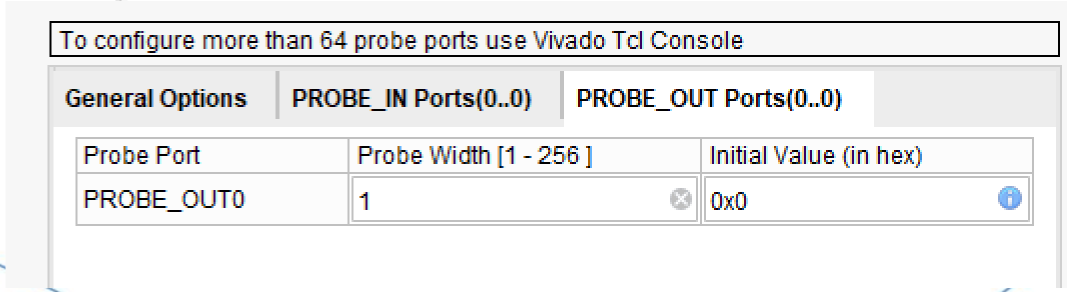
☑ Enable Input Probe Activity Detectors



Component Name vio_1

To configure more than 64 probe ports use Vivado Tcl Console

Probe Port	Probe Width [1 - 256]
PROBE_IN0	1

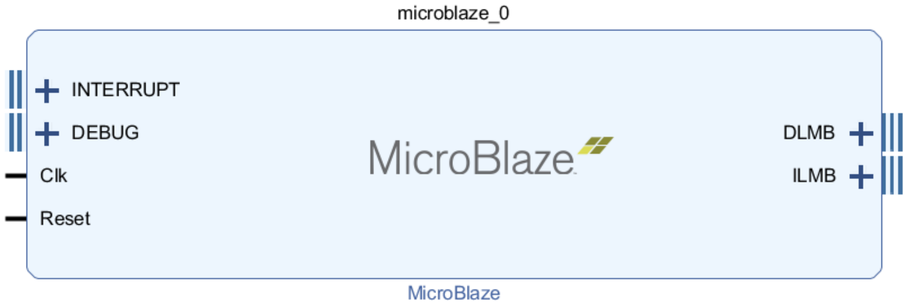


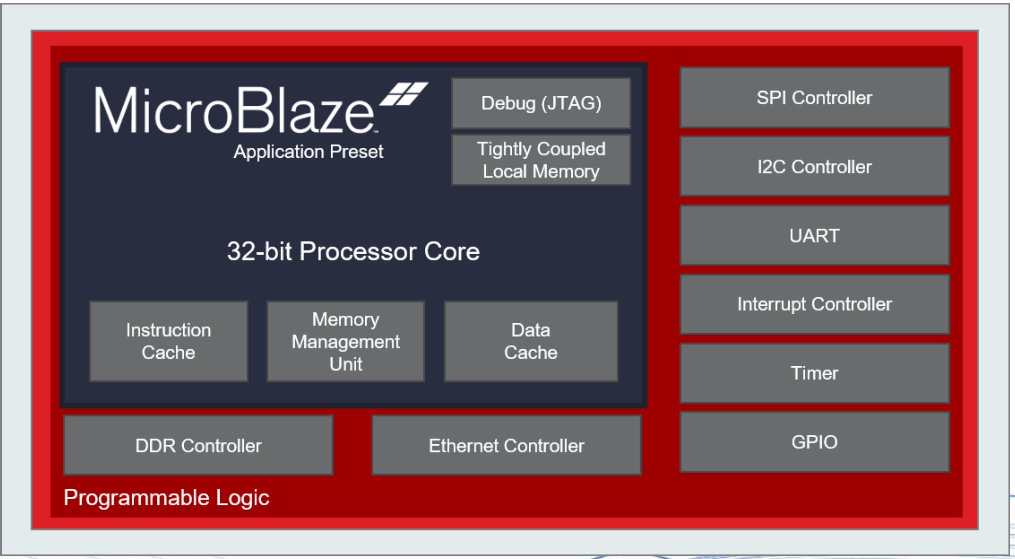
To configure more than 64 probe ports use Vivado Tcl Console

Probe Port	Probe Width [1 - 256]	Initial Value (in hex)
PROBE_OUT0	1	0x0

IP Cores

Microblaze





MicroBlaze Application Preset

Debug (JTAG)

Tightly Coupled Local Memory

32-bit Processor Core

Instruction Cache

Memory Management Unit

Data Cache

SPI Controller

I2C Controller

UART

Interrupt Controller

Timer

GPIO

DDR Controller

Ethernet Controller

Programmable Logic

- It is usually a softcore processor.
- It can be implemented in FPGA or PL part of a SoC
- It can be also found as hardcore processor (dedicated silicon) in MPSoC and RFSoc.
- It can be configured as 32bit or 64bit.
- 32-bit instruction word with three operands and two addressing modes
- Default 32-bit address bus, extensible to 64 bits
- It is very configurable on its internal capabilities and external connections.

peclab

POWER & ENERGY CONVERSION LABORATORY

Electrical Engineering Department

Pontificia Universidad Católica de Chile

peclab.ing.uc.cl