Extendiendo el Sistema Embebido dentro de la PL

Zynq
Vivado 2018.1

Objetivos

Al completar este módulo, el alumno será capaz de:

- Identificar la IP provista como parte de Vivado
- Describir cómo agregar hardware a un proyecto de Vivado existente
- Explicar cómo es agregada la IP para ampliar la funcionalidad del sistema de procesamiento

Temario

- ***Catálogo IP***
- Directorio de IP
- Archivos de dispositivos IP
- Interfaces GP
- Agregando IP para extender PS dentro de la PL
- Generación del bitstream
- Resumen

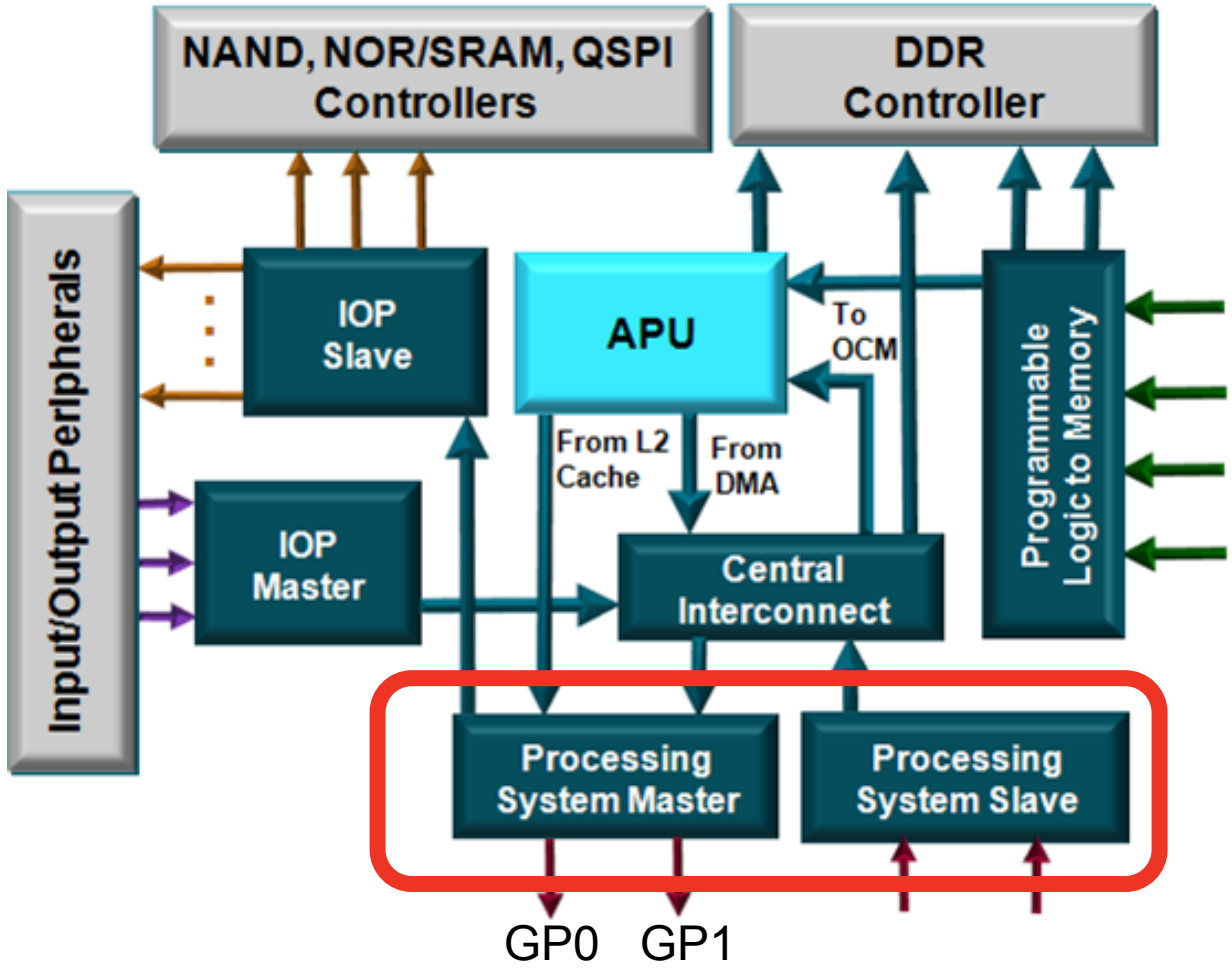
El PS y la PL

- **The Zynq-7000 AP SoC architecture consists of two major sections**
  - PS: Processing system
    - Dual ARM Cortex-A9 processor based
    - Multiple peripherals
    - Hard silicon core
  - **PL: Programmable logic**
    - Shares the same 7 series programmable logic as
      - Artix™-based devices: Z-7010, Z-7015 and Z-7020 (high-range I/O banks only)
      - Kintex™-based devices: Z-7030, Z-7035, Z-7045, and Z-7100 (mix of high-range and high-performance I/O banks)

- **This section focuses on the PL**

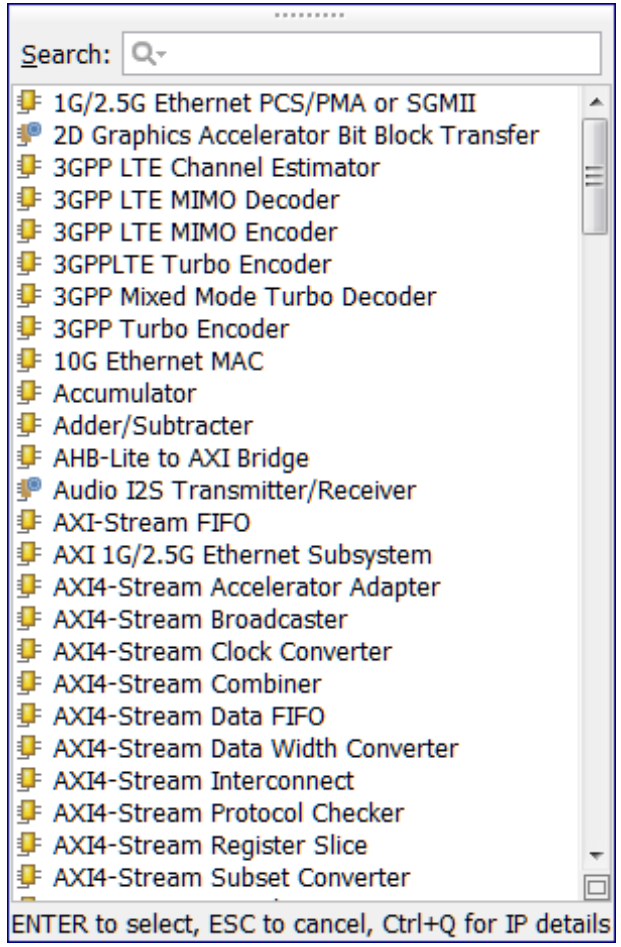
Comunicándose con la PL

- **Processing system master**
  - Two ports from the processing system to programmable logic
  - Connects the CPU block to common peripherals through the central interconnect
- **Processing system slave**
  - Two ports from programmable logic to the processing system
- **Slave PL peripherals address range**
  - 4000_0000 and 7FFF_FFFF (connected to GP0) and
  - 8000_0000 and BFFF_FFFF (connected to GP1)



Catálogo IP

- **The IP Catalog contain a collection of IP that can be used to assemble the (embedded) system**
- **Supported by IPI**
- **Facilitates quick system construction**
- **Each IP block has its own configuration parameters**
- **Most of the IP is free, some require licenses**
- **Stored as source code in the install directory**
  - Always synthesized with the latest tools
  - Some proprietary source code is encrypted
- **Peripherals in the PS are always present and can be dynamically enabled or disabled through PS Configuration wizard**



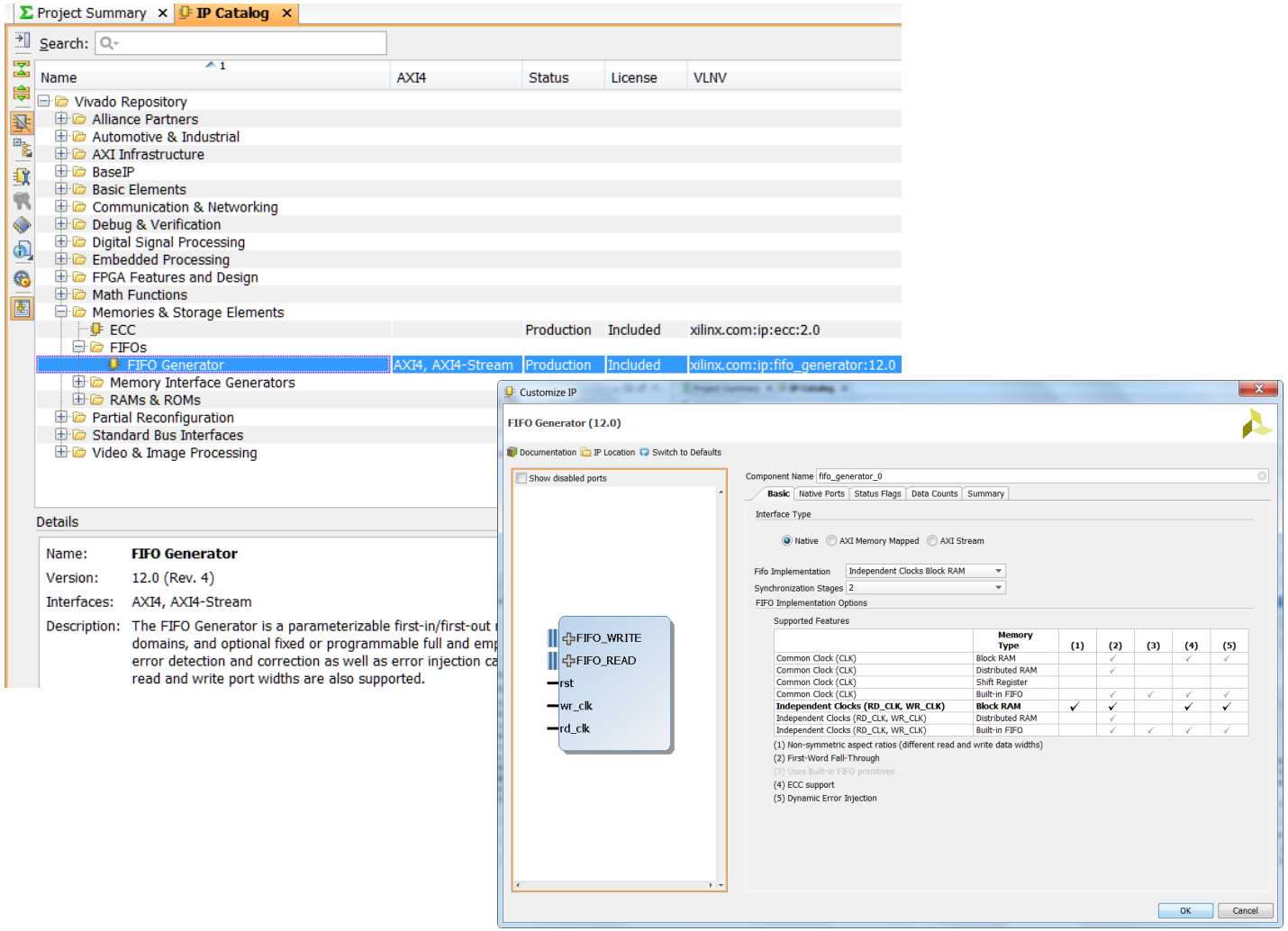
Periféricos IP
Incluidos como fuente (Free)

- **Controladores de bus y bridge**
  - AXI to AXI connector
  - Local Memory Bus (LMB)
  - AXI Chip to Chip
  - AHB-Lite to AXI
  - AXI4-Lite to APB
  - AXI4 to AHB-Lite
- **Debug cores**
  - Integrated Logic Analyzer
- **DMA and Timers**
  - Watchdog, fixed interval
- **Inter-processor communication**
- **External peripheral controller Memory and memory controller**
- **High-speed and low-speed communication peripherals**
  - AXI 10/100 Ethernet MAC controller
  - Hard-core tri-mode Ethernet MAC
  - AXI IIC
  - AXI SPI
  - AXI UART
- **Other cores**
  - System monitor
  - Xilinx Analog-to-Digital Converter (XADC)
  - Clock generator, System reset module
  - interrupt controller
  - Traffic Generator, Performance monitor

Catálogo de IP en Vivado

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



Cores IP incluidos como evaluación

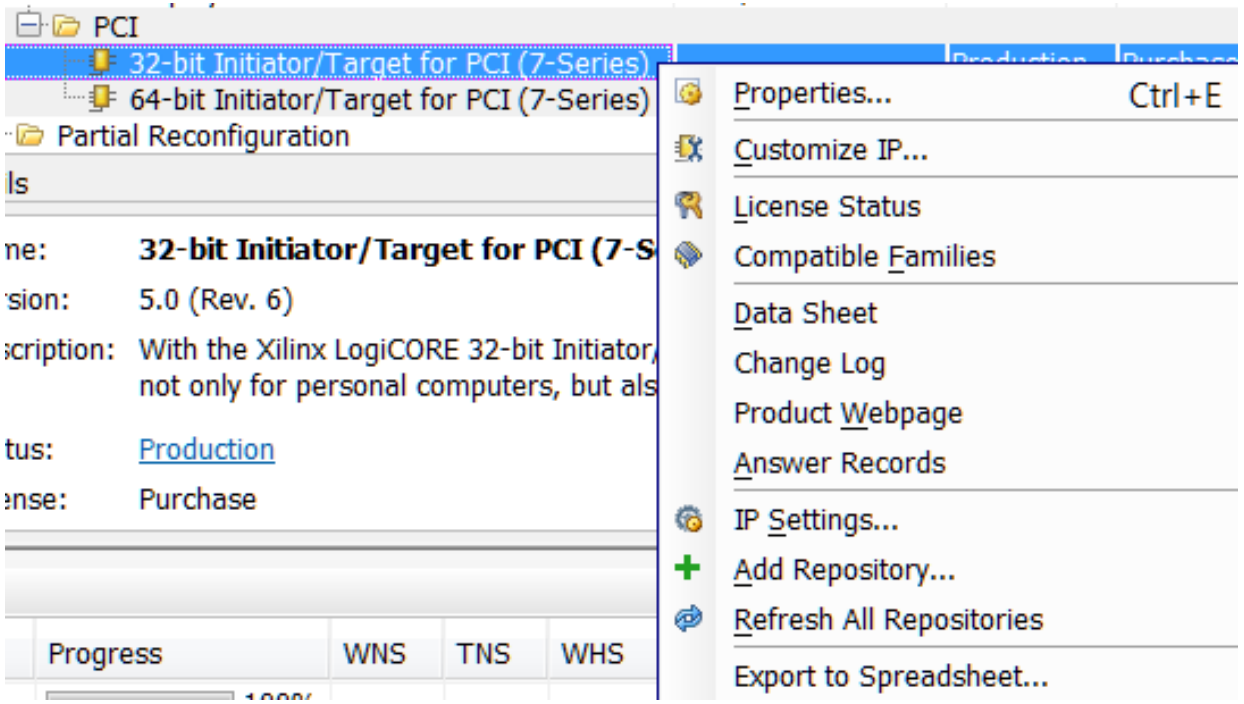
- AXI CAN controller
- AXI USB2 device
- Video IP
- Telecoms/ Wireless IP

Standard Bus Interfaces				
S/PDIF	AXI4, AXI4-Stream	Production	Purchase	xilinx.com:ip:spdif:2.0
SD Card Host Controller	AXI4	Production	Purchase	logicbricks.com:logicbricks:logisdhc:0.0
RapidIO				
Serial RapidIO Gen2	AXI4, AXI4-Stream	Production	Purchase	xilinx.com:ip:srio_gen2:4.0
PCI Express				
AXI Memory Mapped To PCI Express	AXI4	Production	Included	xilinx.com:ip:axi_pcie:2.6
7 Series Integrated Block for PCI Express	AXI4-Stream	Production	Included	xilinx.com:ip:pcie_7x:3.1
DisplayPort				
DisplayPort	AXI4, AXI4-Stream	Pre-Produ...	Purchase	xilinx.com:ip:displayport:6.0
PCI				
32-bit Initiator/Target for PCI (7-Series)		Production	Purchase	xilinx.com:ip:pci32:5.0
64-bit Initiator/Target for PCI (7-Series)		Production	Purchase	xilinx.com:ip:pci64:5.0

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

Cores IP

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



Información de un IP Core

Hoja de datos provista para cada core (para acceder hacer click-derecho sobre el core en el catálogo IP

- **El tamaño de cada core está disponible en la hoja de datos**
- **Por ejemplo, la hoja de datos del axi_timer_v2_00_a contiene la siguiente tabla:**

Table 2-2: **Performance and Resource Utilization: Artix-7 FPGA (XC7A355TDIE) and Zynq-7000 Devices**

Parameter Values		Device Resources		
Width of Timer/Counter	Enable Timer2	Slices	Flip-Flops	LUTs
8	False	49	53	96
16	False	61	69	120
32	False	84	101	181
8	True	50	74	123
16	True	74	106	161
32	True	97	170	256

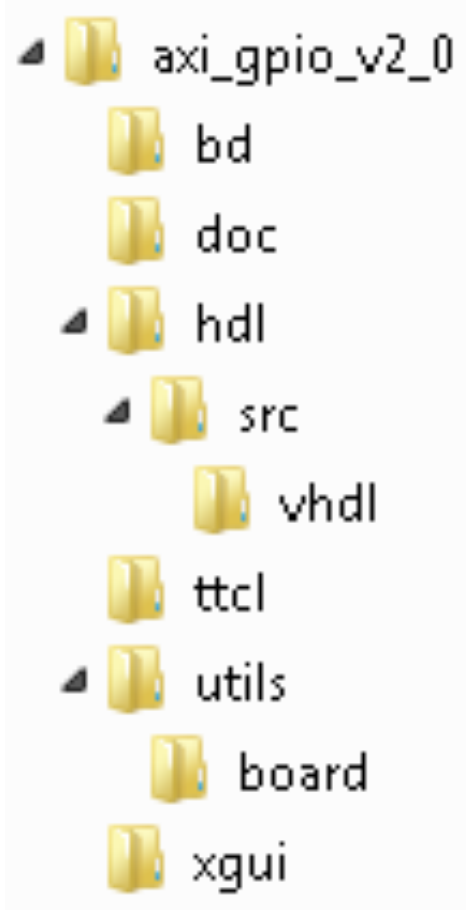
Temario

- Catálogo IP
- ***Directorio de IP***
- Archivos de dispositivos IP
- Interfaces GP
- Agregando IP para extender PS dentro de la PL
- Generación del bitstream
- Resumen

Almacenamiento de periféricos

Los periféricos de usuario pueden ser ubicados en el directorio del proyecto o en un repositorio de perifpericos

- **Directorio de los cores de IP (ubicado en el directorio del proyecto)**
  - {component}.xml
  - Directorio MyProcessorIPLib (definido por el usuario)
    - Directorio del repositorio, listado usando la pestaña **Project → Project Options → Device and Repository Search**
  - *%XILINX_INSTALL%\Vivado\2015.X\data\ip*



Temario

- Catálogo IP
- Directorio de IP
- ***Archivos de dispositivos IP***
- Interfaces GP
- Agregando IP para extender PS dentro de la PL
- Generación del bitstream
- Resumen

Archivos de los cores de IP

- **component.xml**
  - Formato XML
  - Carpeta Top level
  - Provee la descripción de puertos, parámetros y opciones para las IP
  - Links a los archivos fuente
- **xgui folder**
  - Archivo .tcl para la GUI IPI

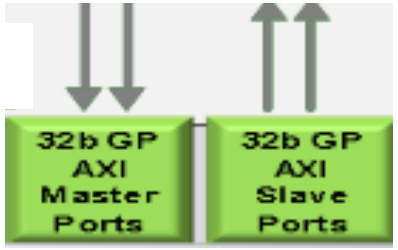
```
<?xml version="1.0" encoding="UTF-8"?>
<spirit:component xmlns:xilinx="http://www.xilinx.com" xmlns:spirit="http://www.spiritconsor
tium.org/XMLSchema/SPIRIT/1685-2009" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <spirit:vendor>xilinx.com</spirit:vendor>
  <spirit:library>XUP</spirit:library>
  <spirit:name>led_ip</spirit:name>
  <spirit:version>1.0</spirit:version>
  <spirit:busInterfaces>
    <spirit:busInterface>
      <spirit:name>S_AXI</spirit:name>
      <spirit:busType spirit:vendor="xilinx.com" spirit:library="interface" spirit:name="axi
mm" spirit:version="1.0"/>
      <spirit:abstractionType spirit:vendor="xilinx.com" spirit:library="interface" spirit:n
ame="aximm_rtl" spirit:version="1.0"/>
      <spirit:slave>
        <spirit:memoryMapRef spirit:memoryMapRef="S_AXI"/>
      </spirit:slave>
      <spirit:portMaps>
        <spirit:portMap>
          <spirit:logicalPort>
            <spirit:name>AWADDR</spirit:name>
          </spirit:logicalPort>
          <spirit:physicalPort>
            <spirit:name>s_axi_awaddr</spirit:name>
          </spirit:physicalPort>
        </spirit:portMap>
```

Temario

- Catálogo IP
- Directorio de IP
- Archivos de dispositivos IP
- ***Interfaces GP***
- Agregando IP para extender PS dentro de la PL
- Generación del bitstream
- Resumen

Puertos de GP

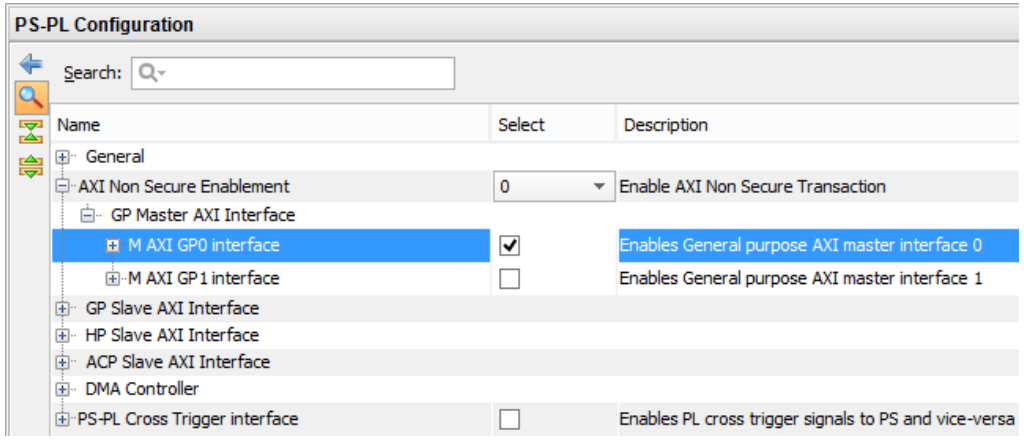
- **Por defecto, los puertos de GP maestro y esclavo están deshabilitados**

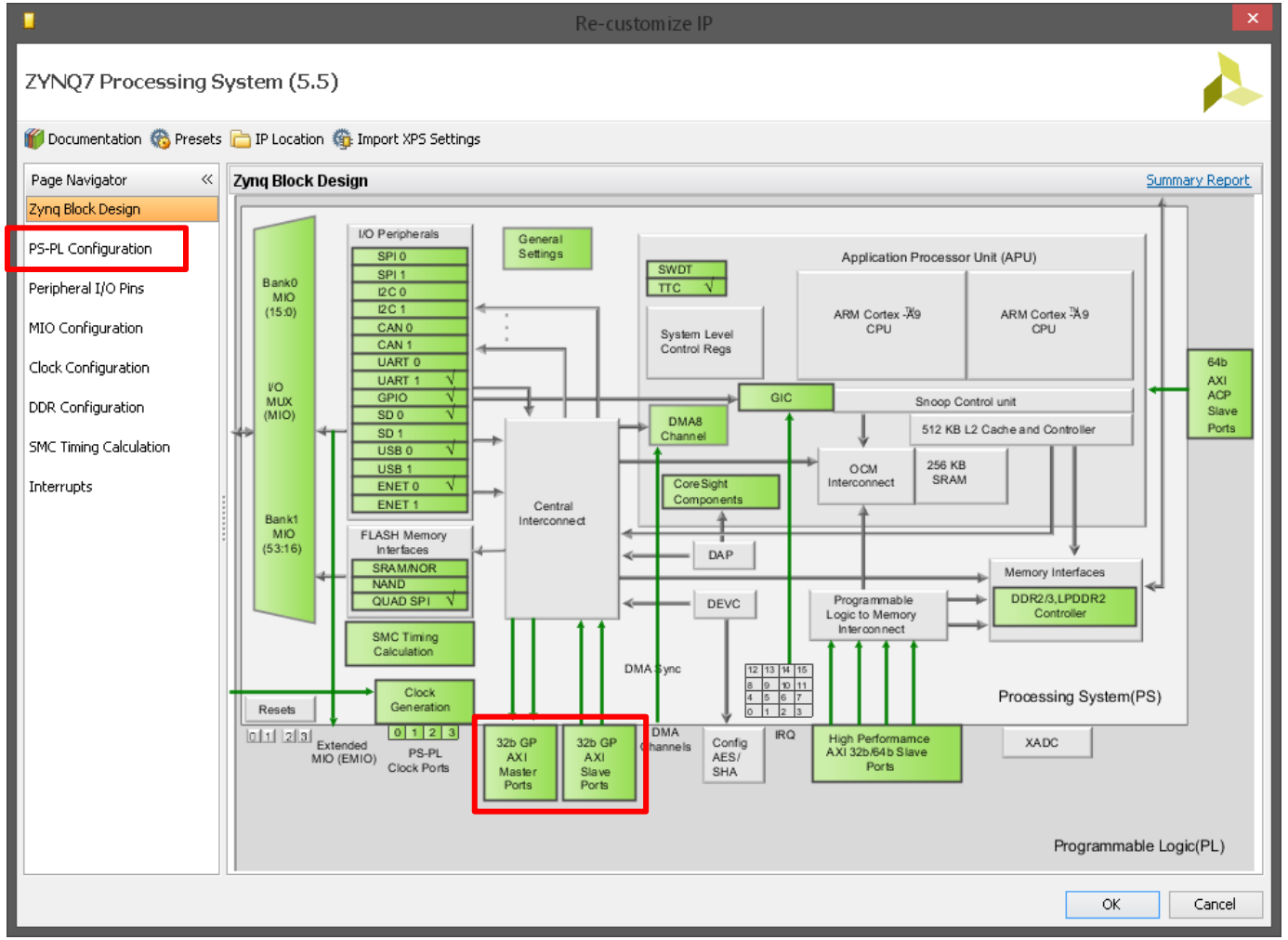


- **Habilitar los puertos GP maestro y/o esclavo depediendo si un periférico esclavo o maestro será agregado en la PL**
- **El bloque axi_interconnect es requerido para conectar la IP a un puerto con diferentes protocolos**
  - Conversión automática de protocolos
  - Puede ser automáticamente agregado cuando se usa Block Automation en IPI (IP Integrator)

Configurando los puertos GP

- **Clickear sobre el menu o sobre los bloques GP verdes para configurar**



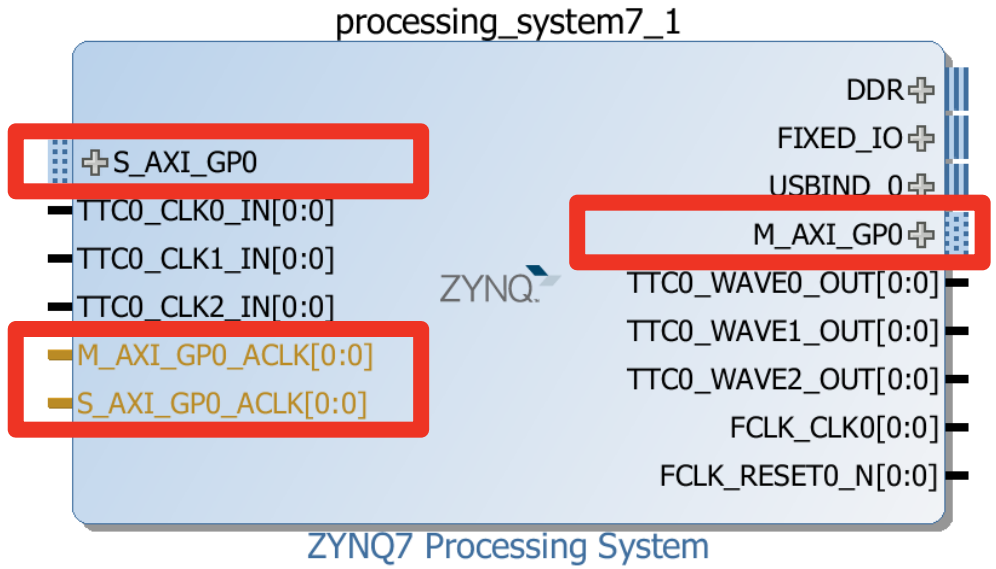


Temario

- Catálogo IP
- Directorio de IP
- Archivos de dispositivos IP
- Interfaces GP
- ***Agregando IP para extender PS dentro de la PL***
- Generación del bitstream
- Resumen

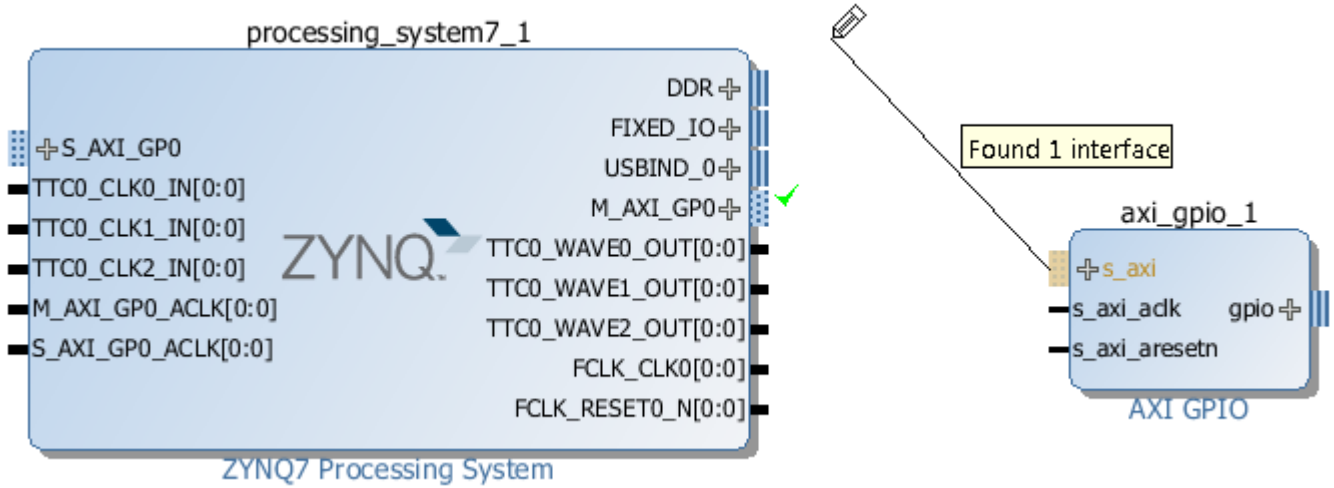
Agregar IP en la PL

- **Configurar los puertos GP ports desde PS Customization GUI**
- **Configurción PS-PL**
  - Ej: Habilitar M_AXI_GP0/1 o S_AXI_GP0/1
- **Los puertos serán entonces habilitados en el diagrama en bloques Zynq**
- **Conectar la IP agregada al puerto apropiado**
- **Asignar una dirección al IP agregado, si no está mapeado**
- **Si es necesario, sonfigurar la IP**
- **Si es necesario, estblecer las conexiones externas, if needed**
  - Agregar puertos/interfaces externas si la IP agregada interactúa con dispositivos externos



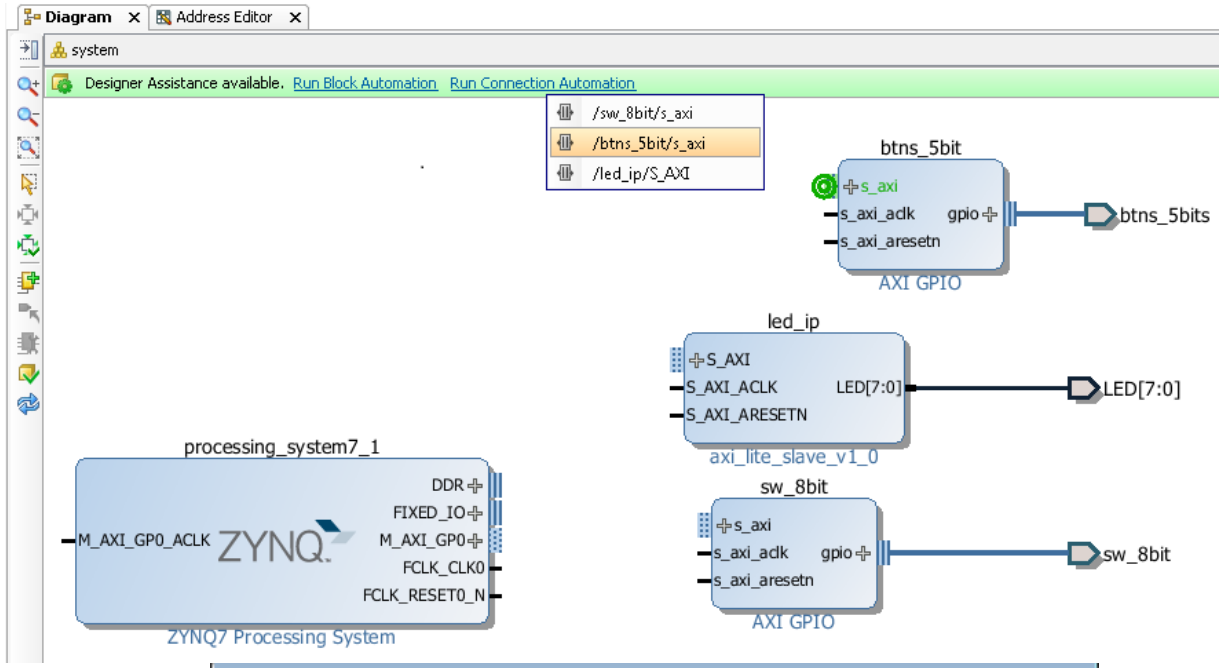
Connecting IP

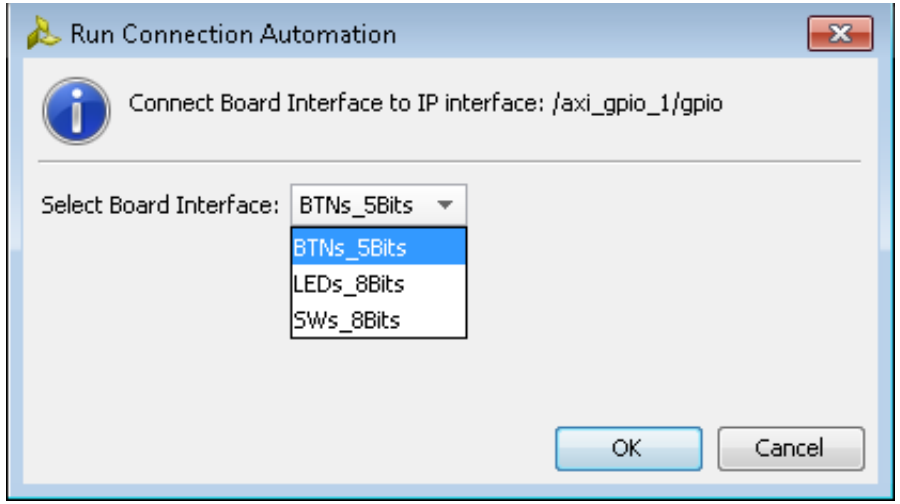
- **Agregar IP desdeel catálogo de IP**
- **Clickear y arrastrar para encontrar las conexiones**
- **Las conexiones válidas son resaltadas**
- **Asistencia al diseñador, automatización de conexión**
  - Si hay disponible soporte para la placa, la IP puede ser conectada a pines externos
- **O creación y conexión manual de puertos (externos)**



Asistente al diseñador; Automatización de bloques, Automatización de conexión

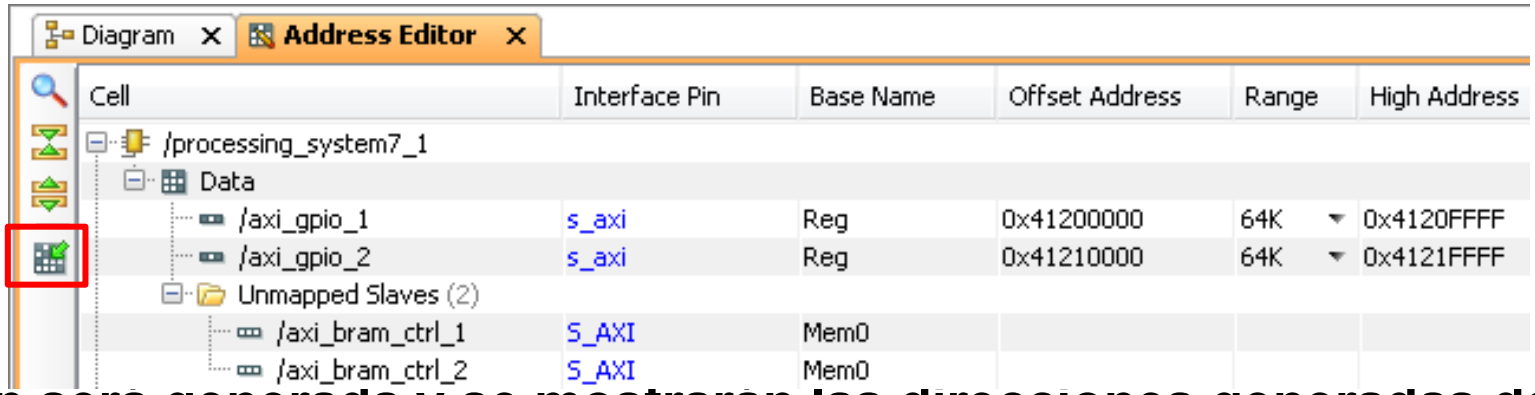
- Bloque, Conexión
- Puede automáticamente conectar bloques IP
- Inserta automáticamente bloques requeridos
- Ej: Agregado de BRAM; la automatización inertará y conectará el controlador BRAM y la lógica de reset
- Si hay disponible soporte para la placa (Board support), la IP puede ser conectada automáticamente a los puertos top level





Asignación de Direcciones

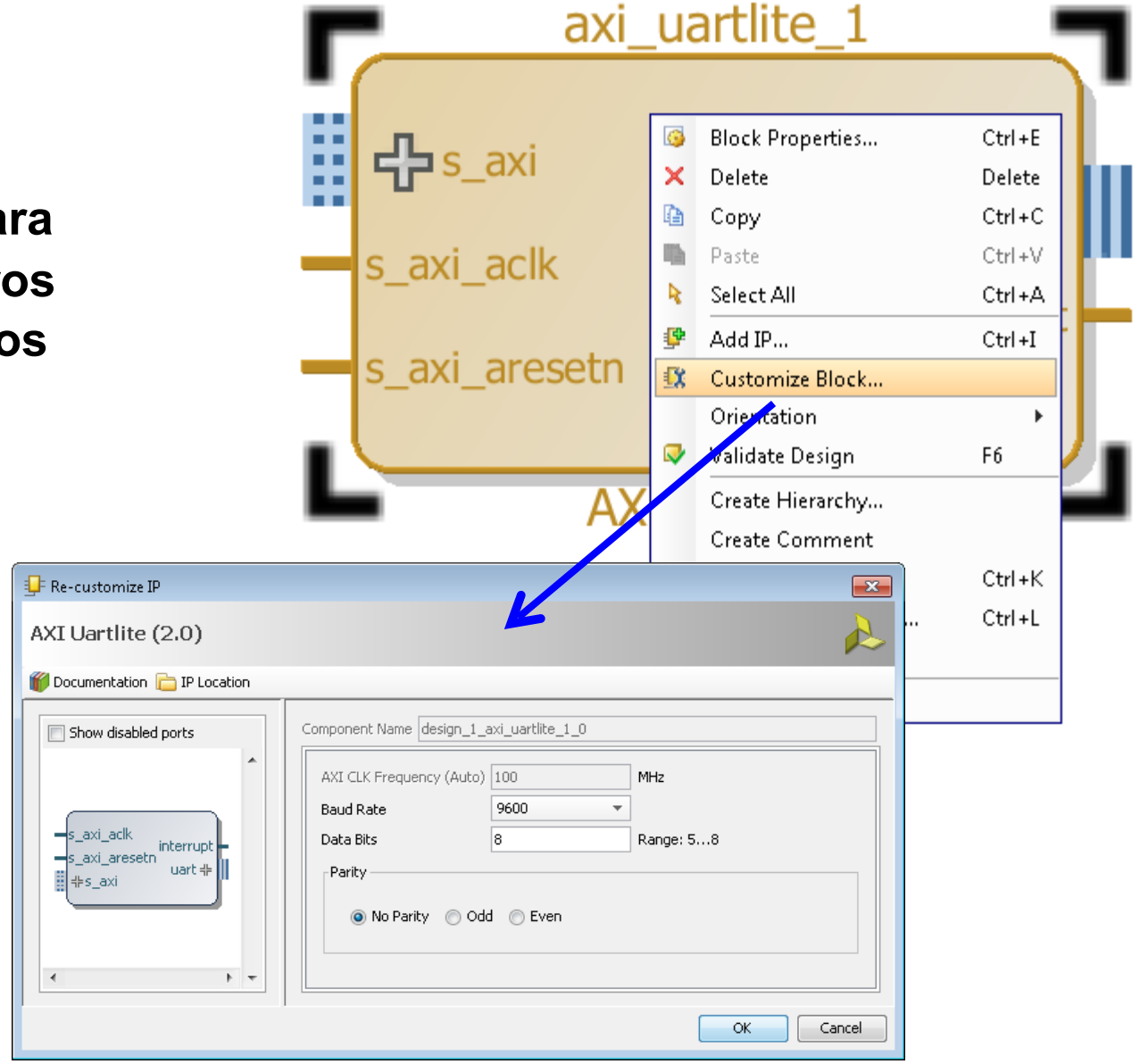
- Los periféricos en el Zynq™ AP SoC PS tiene direcciones fijas y no aparecen en el mapa de direcciones cuando una IP es agregada al sistema
- Para los periféricos del PS Clickear sobre el botón *Auto Assign Addresses*



- La dirección sera generada y se mostraran las direcciones generadas de la IP agregada
- Las direcciones fijas de los periféricos configurados del PS

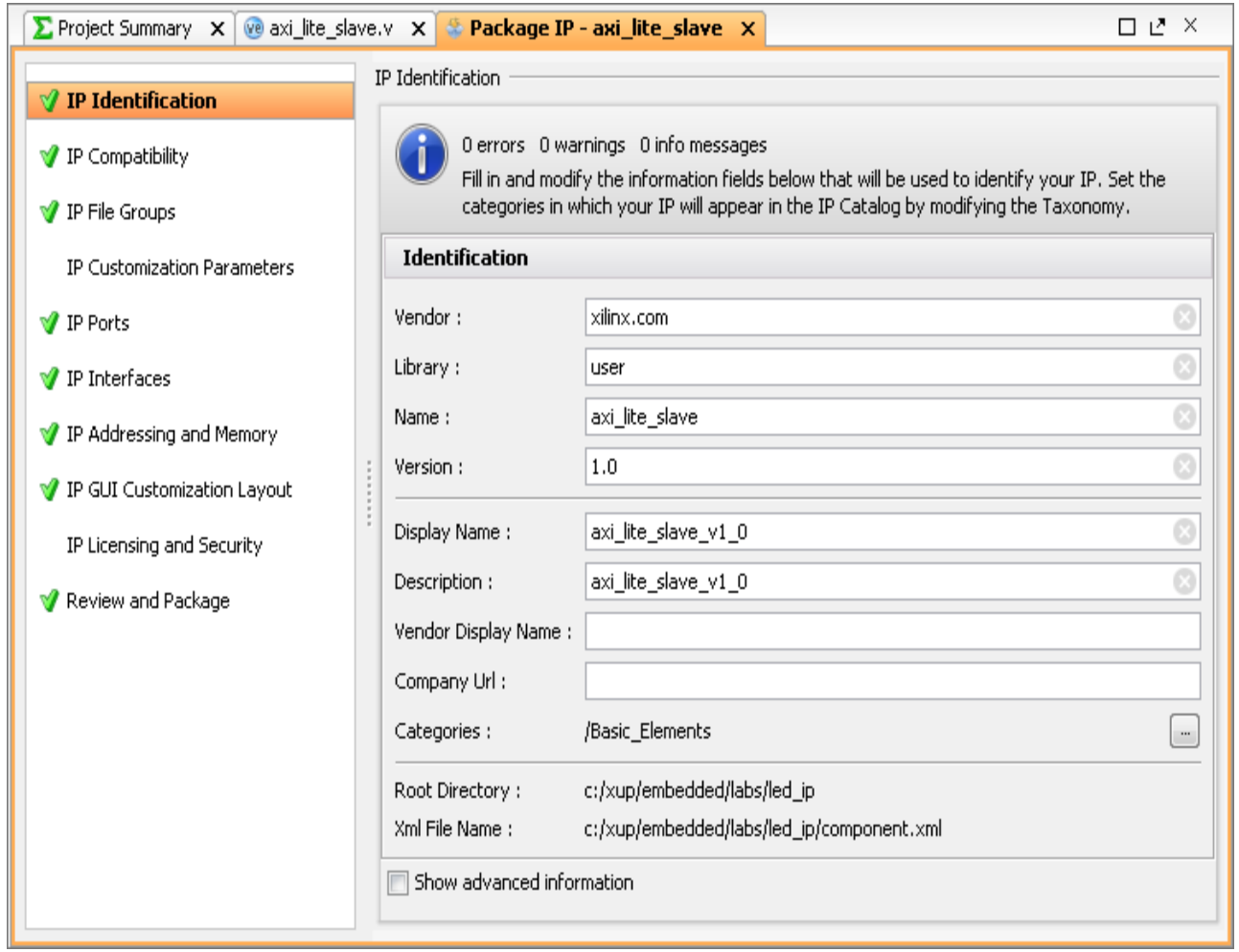
Parametrizar instancias de IP

- **Doble-click o click derecho sobre la instancia y seleccionar *Customize Block* para abrir el cuadro de diálogo de los parámetros configurables (referirse a las hojas de datos de ser necesario)**
- **Son mostrados los valores por defecto**
  - Personalizar los parámetros que se requieran



Ampliando el catálogo IP

- **IP Packager**
  - Empaquetado dentro del formato IP Integrator
- **Especificar el repositorio (local/global)**
- **La IP puede entonces ser usada en IP Integrator**
- **Luego veremos más sobre IP Packager**



Temario

- Catálogo IP
- Directorio de IP
- Archivos de dispositivos IP
- Interfaces GP
- Agregando IP para extender PS dentro de la PL
- ***Generación del bitstream***
- Resumen

- **Después de haber definido el sistema de hardware, el siguiente paso es crear la netlist de hardware si el sistema de hardware tiene lógica en la PL**
- **Se debe generar un wrapper de HDL para el diagrama en bloques**
  - Se puede agregar lógica adicional al HDL, o el sistema del procesador puede ser usado como un sub-bloque en un diseño HDL
- **El diseño y el diagrama en bloques deben estar abiertos antes de que la síntesis y la implementación puedan llevarse a cabo**
- **Si el sistema tiene hardware en la PL, se debe generar el bitstream**
- **La PL (FPGA) debe ser programada antes de que la aplicación pueda ser descargada y ejecutada**

Temario

- Catálogo IP
- Directorio de IP
- Archivos de dispositivos IP
- Interfaces GP
- Agregando IP para extender PS dentro de la PL
- Generación del bitstream
- ***Resumen***

- **La funcionalidad del PS puede ser ampliada por medio de la instanciación de periféricos en la PL**
- **Agregar IP en la PL involucra**
  - Habilitar interface(s) en el PS
  - Seleccionar la IP del catálogo IP y configurarla para la funcionalidad deseada
  - Conectar la (PL) IP al PS usando IP Integrator
  - Asignar dirección
  - Conectar los puertos de la IP a puertos de otros perifpericos y/o a pines externos
- **Se necesita un Wrapper HDL para el *IP Integrator Block***
- **Se debe generar el bitstream cuando la PL tiene alguna IP**
- **La FPGA debe ser programada con el bitstream de hardware generado antes de que una aplicación pueda correr**