

# Introducción a Zynq con ZYBO



DEPARTAMENTO  
DE COMPUTACION  
Facultad de Ciencias Exactas y Naturales - UBA

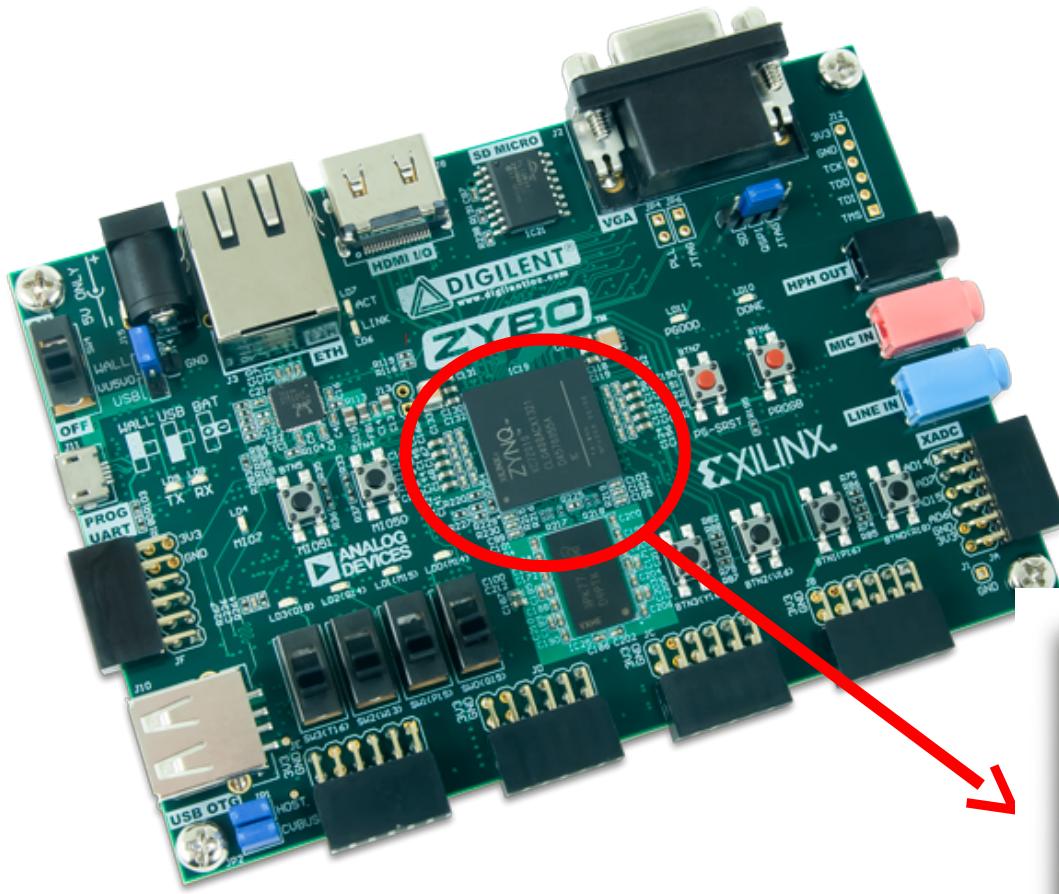


**SASE 2015**  
**Patricia Borensztein**  
**y Miguel García**

# Zynq y Zybo

- Zynq es la nueva generación de **All-Programmable System-on-Chip** de Xilinx
- ZYBO es el nombre de la placa de desarrollo que integra uno de los componentes mas sencillos de la familia Zynq, el Z7010 **AP Soc**

# ZYBO y Zynq



# Zynq-7000: Product of the Year

- 2011: It was named the SoC Product of the year by CMP Media (now called UBM, the publisher of EE Times and EDN),
- 2011: It won the IET Innovation Award as well as the Embedded System Product of the Year Award from Elektra (the European Electronics Industry)
- 2012: Electronic Products Magazine named the Zynq-7000 its Product of the Year.
- 2012: The Microprocessor Report gave the device its Analyst Choice Award.

# Familia Zynq-7000

Zynq-7000 All Programmable SoC												
	Device Name	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100				
	Part Number	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100				
Processing System	Processor Core	Dual ARM® Cortex™-A9 MPCore™ with CoreSight™										
	Processor Extensions	NEON™ & Single / Double Precision Floating Point for each processor										
	Maximum Frequency	667 MHz (-1); 766 MHz (-2); 866 MHz (-3)		667 MHz (-1); 800 MHz (-2); 1 GHz (-3)		667 MHz (-1) 800 MHz (-2)						
	L1 Cache	32 KB Instruction, 32 KB Data per processor										
	L2 Cache	512 KB										
	On-Chip Memory	256 KB										
	External Memory Support <sup>(1)</sup>	DDR3, DDR3L, DDR2, LPDDR2										
	External Static Memory Support <sup>(1)</sup>	2x Quad-SPI, NAND, NOR										
	DMA Channels	8 (4 dedicated to Programmable Logic)										
	Peripherals <sup>(1)</sup>	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO										
Programmable Logic	Peripherals w/ built-in DMA <sup>(1)</sup>	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO										
	Security <sup>(2)</sup>	RSA Authentication, and AES and SHA 256-bit Decryption and Authentication for Secure Boot										
	Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)		2x AXI 32b Master 2x AXI 32-bit Slave									
			4x AXI 64-bit/32-bit Memory									
			AXI 64-bit ACP									
			16 Interrupts									

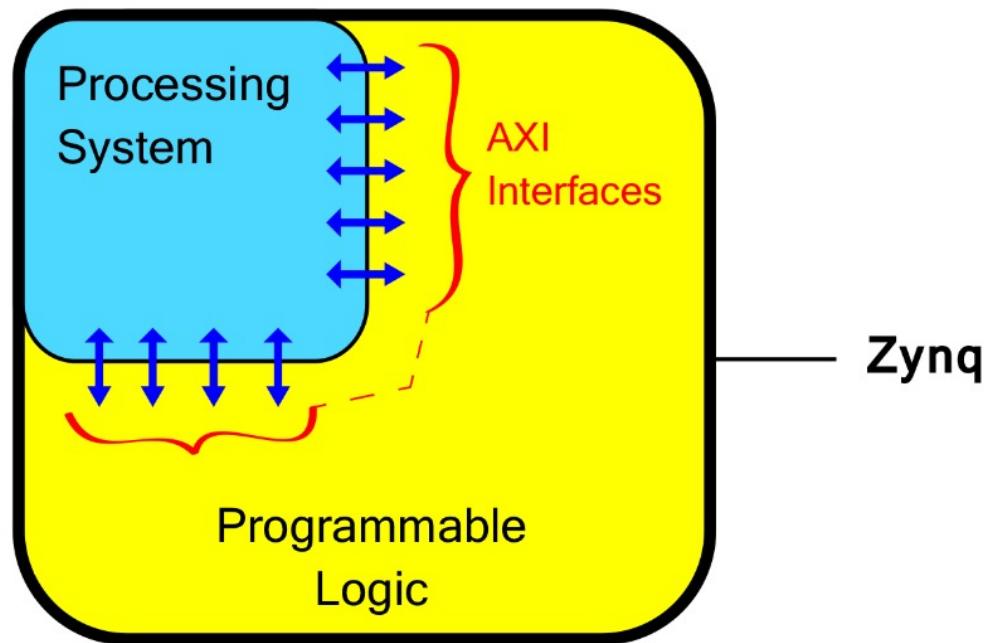
# Familia Zynq-7000

Zynq-7000 All Programmable SoC								
	Device Name	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
	Part Number	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100
Programmable Logic	Xilinx 7 Series Programmable Logic Equivalent	Artix®-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Kintex®-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA
	Programmable Logic Cells (Approximate ASIC Gates) <sup>(3)</sup>	28K Logic Cells (~430K)	74K Logic Cells (~1.1M)	85K Logic Cells (~1.3M)	125K Logic Cells (~1.9M)	275K Logic Cells (~4.1M)	350K Logic Cells (~5.2M)	444K Logic Cells (~6.6M)
	Look-Up Tables (LUTs)	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Extensible Block RAM (# 36 Kb Blocks)	240 KB (60)	380 KB (95)	560 KB (140)	1,060 KB (265)	2,000 KB (500)	2,180 KB (545)	3,020 KB (755)
	Programmable DSP Slices (18x25 MACCs)	80	160	220	400	900	900	2,020
	Peak DSP Performance (Symmetric FIR)	100 GMACs	200 GMACs	276 GMACs	593 GMACs	1,334 GMACs	1,334 GMACs	2,622 GMACs
	PCI Express® (Root Complex or Endpoint) <sup>(4)</sup>	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs						
Security <sup>(2)</sup>		AES and SHA 256b for Boot Code and Programmable Logic Configuration, Decryption, and Authentication						

**Notes:**

1. Restrictions apply for CLG225 package. Refer to the [UG585, Zynq-7000 AP SoC Technical Reference Manual](#) (TRM) for details.
2. Security is shared by the Processing System and the Programmable Logic.
3. Equivalent ASIC gate count is dependent on the function implemented. The assumption is 1 Logic Cell = ~15 ASIC Gates.
4. Refer to [PG054, 7 Series FPGAs Integrated Block for PCI Express](#) for PCI Express support in specific devices.

# Arquitectura Zynq



*Figure 1.2: A simplified model of the Zynq architecture*

*¿En que se diferencia una virtex5 con power pc de una Zynq? (pregunta Bob)*

*The difference is 2 technology generations, (65nm to 28nm), and about 6 years of technical progress.*

*Seriously: read the first data sheet for each family -- all the features are listed, one by one.*

*Austin Lesea  
Principal Engineer  
Xilinx San Jose*

# Arquitectura de un SoC

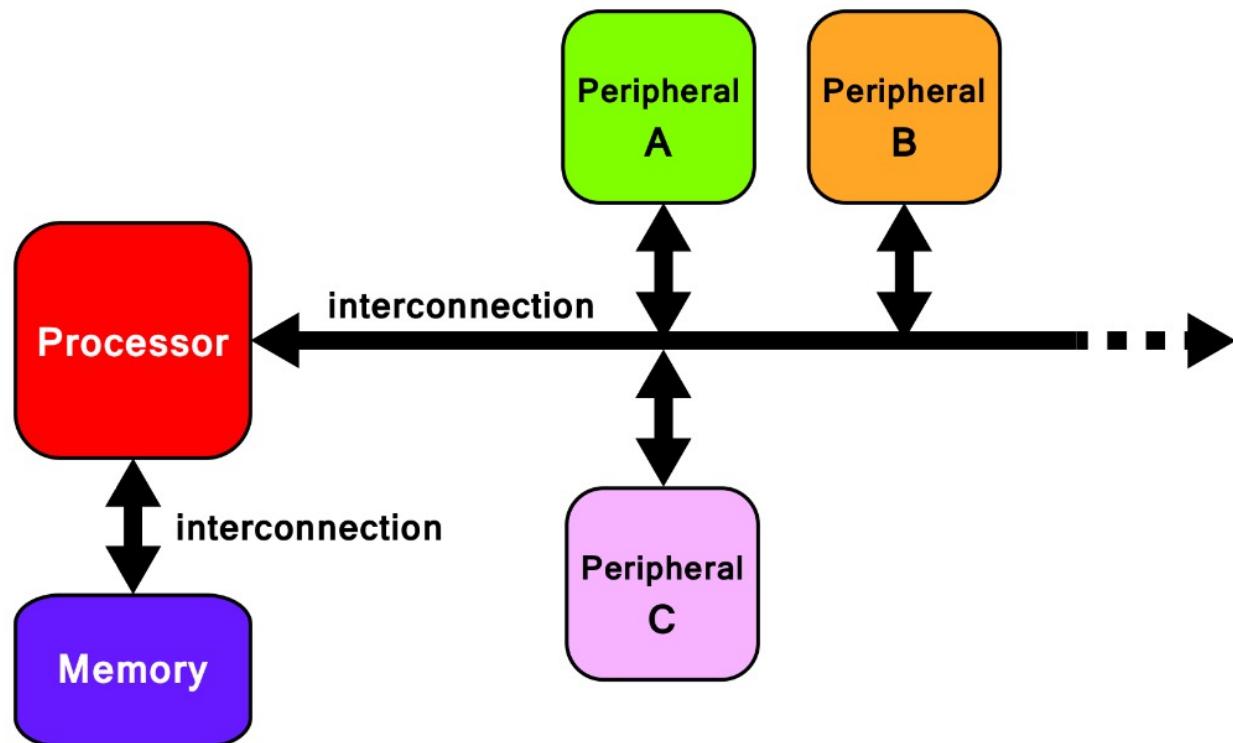


Figure 1.3: The hardware system architecture of an embedded SoC (simplified)

# Arquitectura Zynq

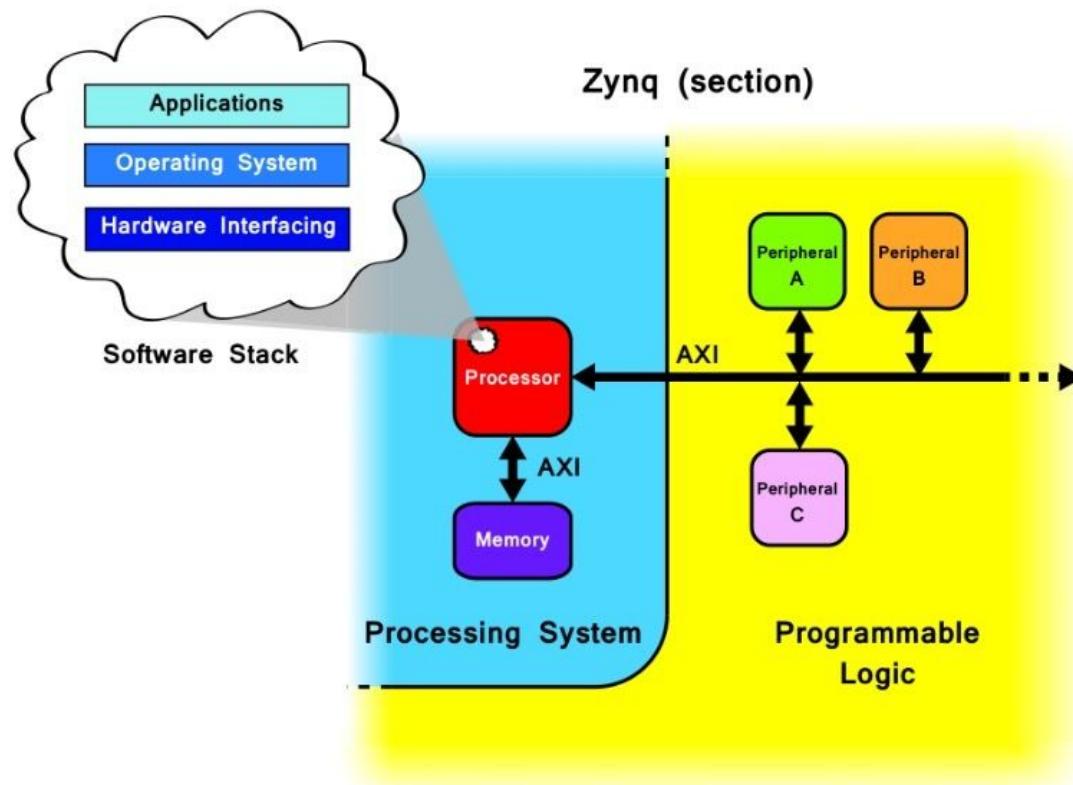


Figure 1.4: Relationship of the software system, hardware system, and Zynq architecture

# Arquitectura Zynq

- PS:
  - Todos los Zynq comparten la misma arquitectura básica construída alrededor del dual ARM Cortex-A9.

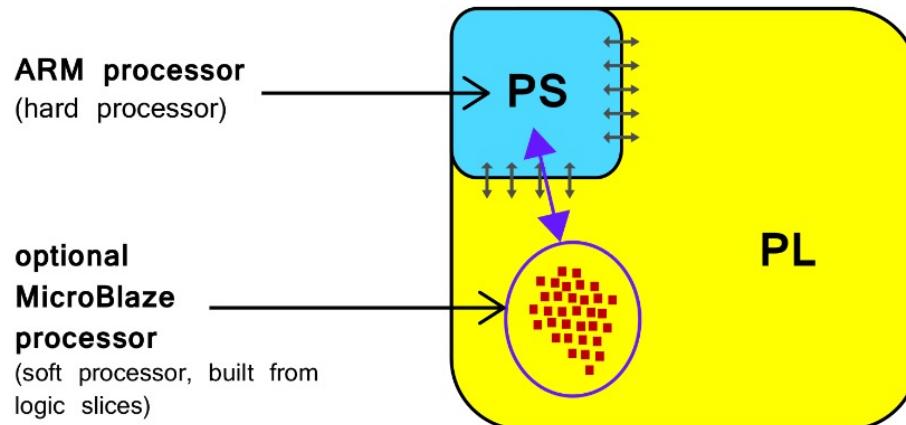
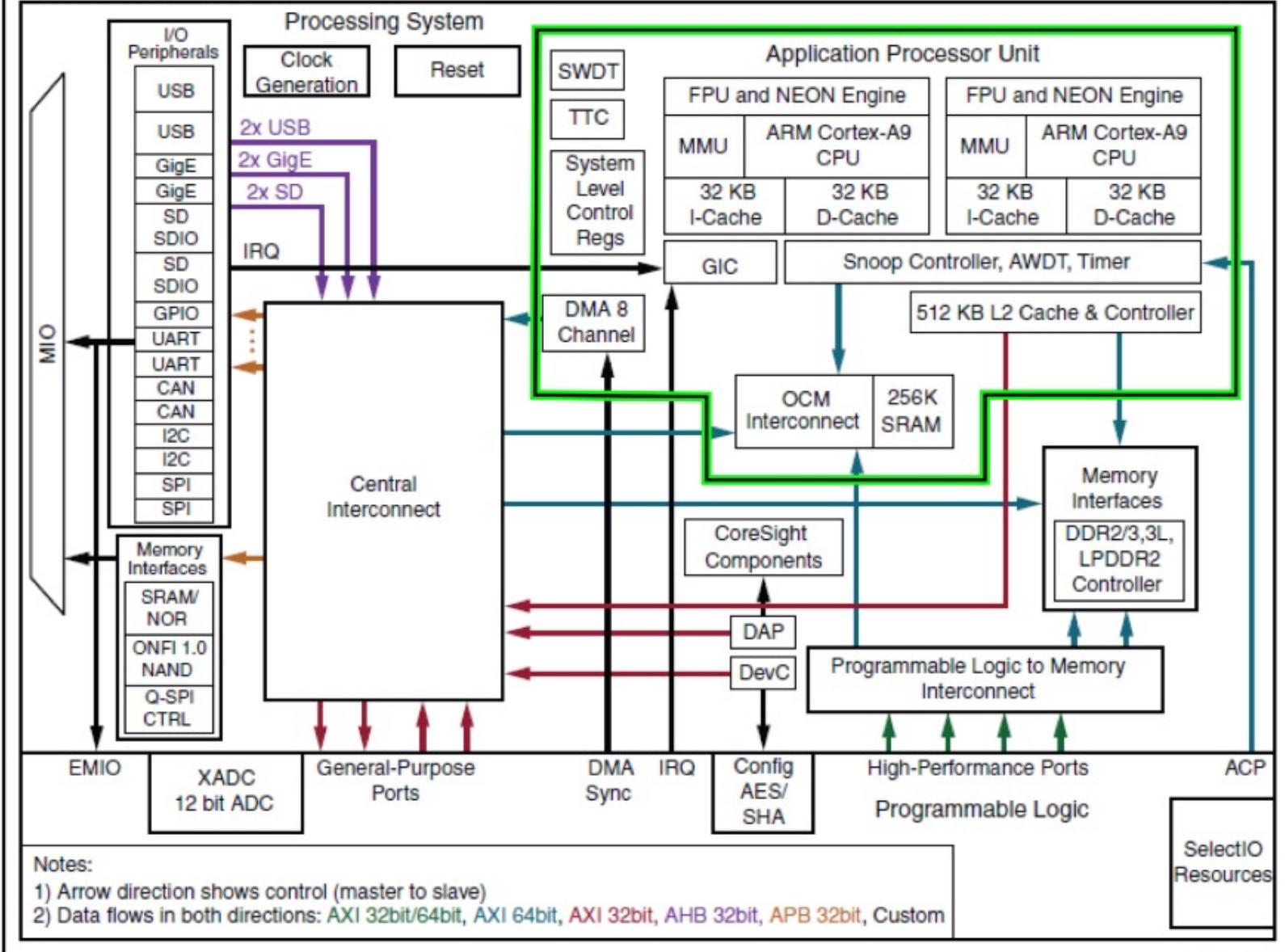


Figure 2.1: Locations of hard (ARM Cortex-A9) and soft (MicroBlaze) processors on a Zynq device

# Arquitectura Zynq

- PS: no sólo contiene el ARM, sino también integra otros recursos
  - APU: Application Processing Unit (APU)
  - Interfaces con Periféricos de E/S
  - Interfaces con Memoria
  - Interconexiones

# Zynq-7000 AP SoC



# Arquitectura Zynq: APU

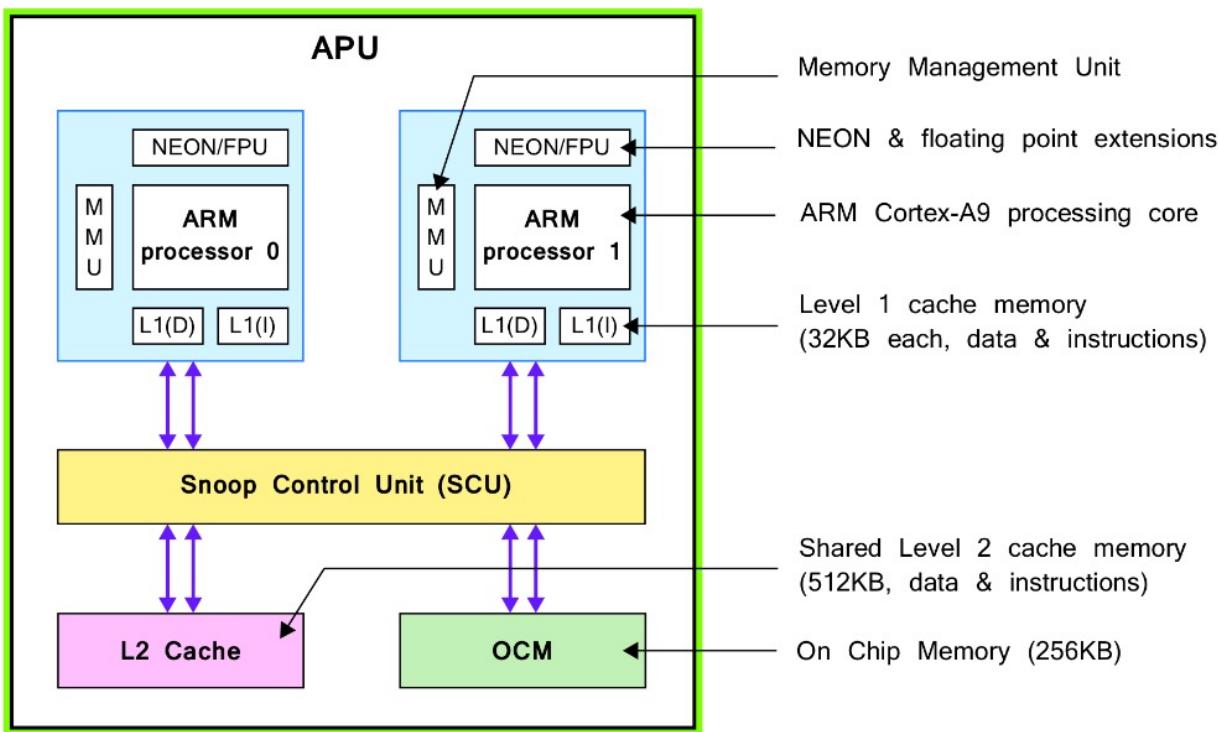


Figure 2.3: Block diagram of the application processing unit (simplified)

# PS: APU

- ARM puede operar hasta 1 GHz, dependiendo del dispositivo
- L1 separadas en instrucciones y datos de 32KB
- L2 compartida por los dos ARM de 512KB
- OCP: On Chip Memory de 256KB, accesible por PL y por CPU
- ACP: Accelerator coherence port interface para accesos a memoria desde PL a CPU.
- SCU: Snoop Control Unit , ademas de mantener coherencia entre las memorias, tiene un rol tambien en la interface entre PS y PL
- DMA
- Timers
- Controlador de Interrupciones

# PS: APU

- Soporte para Programación: SDK de Xilinx
- Compilador soporta ARM y Thumb instruction sets (16 y 32 bits)
- NEON provee instrucciones SIMD
- Unidad FPE: Floating Point Extensions

# PS: Memory Interfaces

- Controladores de memoria dinámica:
  - Pueden accederse por PL y CPU
  - 4 ports AXI slaves: uno para ARM, 2 para PL y otro para otro master del sistema
- Controladores de memoria estática externa

# PS: Periféricos I/O

- 2 Ethernet MAC
- 2 USB
- 2 CAN
- 2 SD/SDIO
- 2 SPI ports
- 2 UARTS
- 2 I2C interfaces
- Todos ellos se comunican con los dispositivos externos mediante un multiplexor MIO a través del cual comparten 54 pines

# M|O

Table 2.1: List of I/O Peripheral Interfaces

I/O Interface	Description
SPI (x2)	Serial Peripheral Interface [10] <i>De facto standard for serial communications based on a 4-pin interface. Can be used either in master or slave mode.</i>
I2C (x2)	I <sup>2</sup> C bus [14] <i>Compliant with the I2C bus specification, version 2. Supports master and slave modes.</i>
CAN (x2)	Controller Area Network <i>Bus interface controller compliant with ISO 118980-1, CAN 2.0A and CAN 2.0B standards.</i>
UART (x2)	Universal Asynchronous Receiver Transmitter <i>Low rate data modem interface for serial communication. Often used for Terminal connections to a host PC.</i>
GPIO	General Purpose Input/Output <i>There are 4 banks GPIO, each of 32 bits.</i>
SD (x2)	<i>For interfacing with SD card memory.</i>
USB (x2)	Universal Serial Bus <i>Compliant with USB 2.0, and can be used as a host, device, or flexibly (“on-the-go” or OTG mode, meaning that it can switch between host and device modes).</i>
GigE (x2)	Ethernet <i>Ethernet MAC peripheral, supporting 10Mbps, 100Mbps and 1Gbps modes.</i>

# PS: Interconexiones

- Todos estos bloques, APU, Interfaces con Memoria, y los IOP se conectan entre si y con PL a través de una conexión AMBA AXI
- Esta interconexión es no bloqueante y soporta a la vez varias comunicaciones simultáneas

# Zynq: Interfaces

- Entre PS y PL, y dentro de PS: mediante interconexiones AXI
- Entre PS y componentes externos
  - MIO: Multiplexed Input/Output (54 pines): permite mapear flexiblemente cualquiera de las interfaces standard
  - EMIO: Extended MIO : no son directas entre PS y conexiones externas, pasa a través de recursos del PL y comparte recursos con éste.

# Arquitectura Zynq

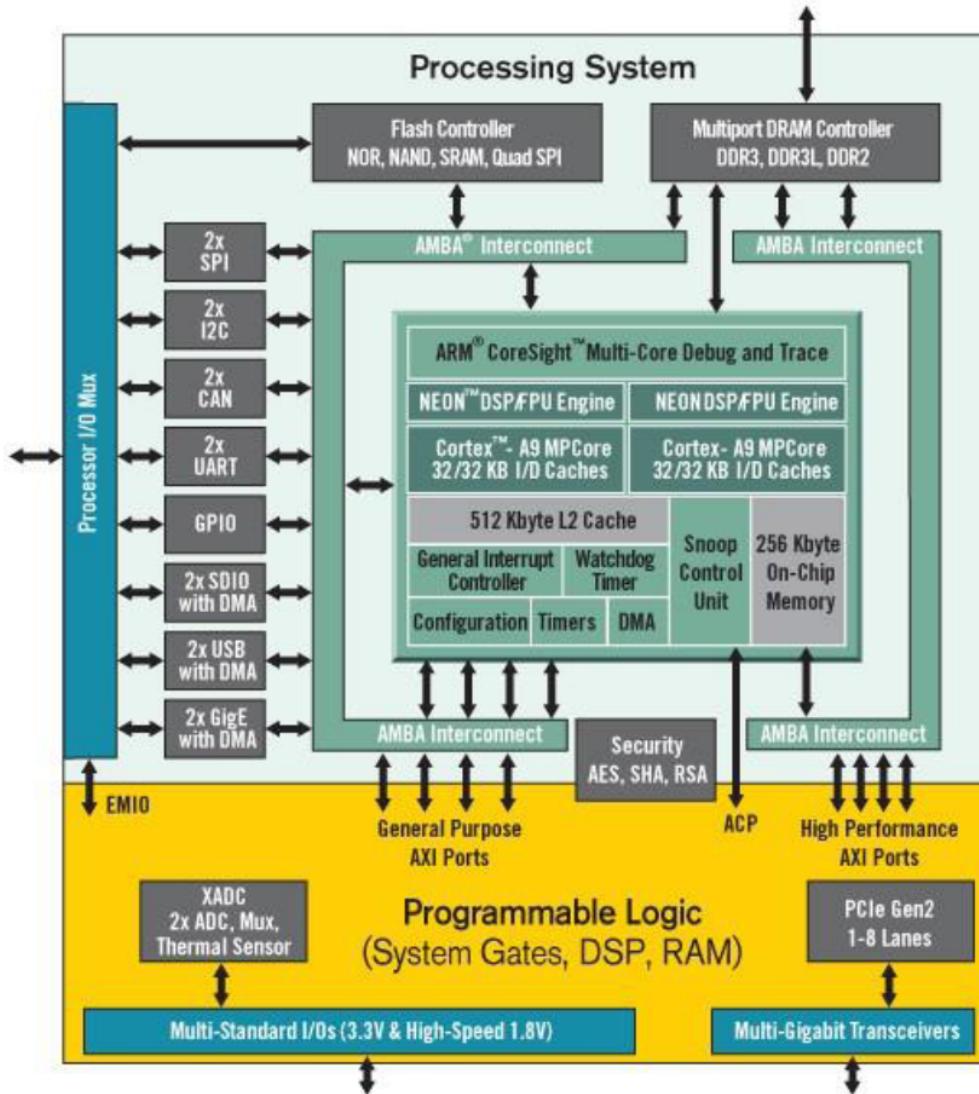


Figure 3. Zynq AP SoC architecture.

# Arquitectura Zynq: PL

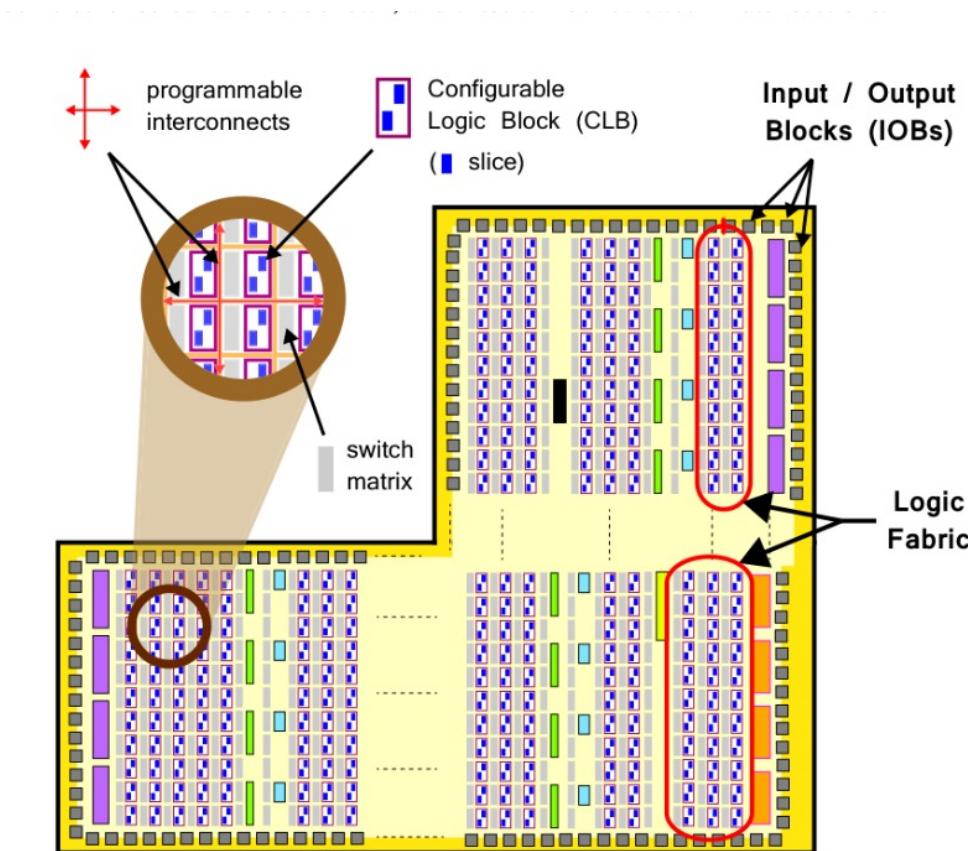
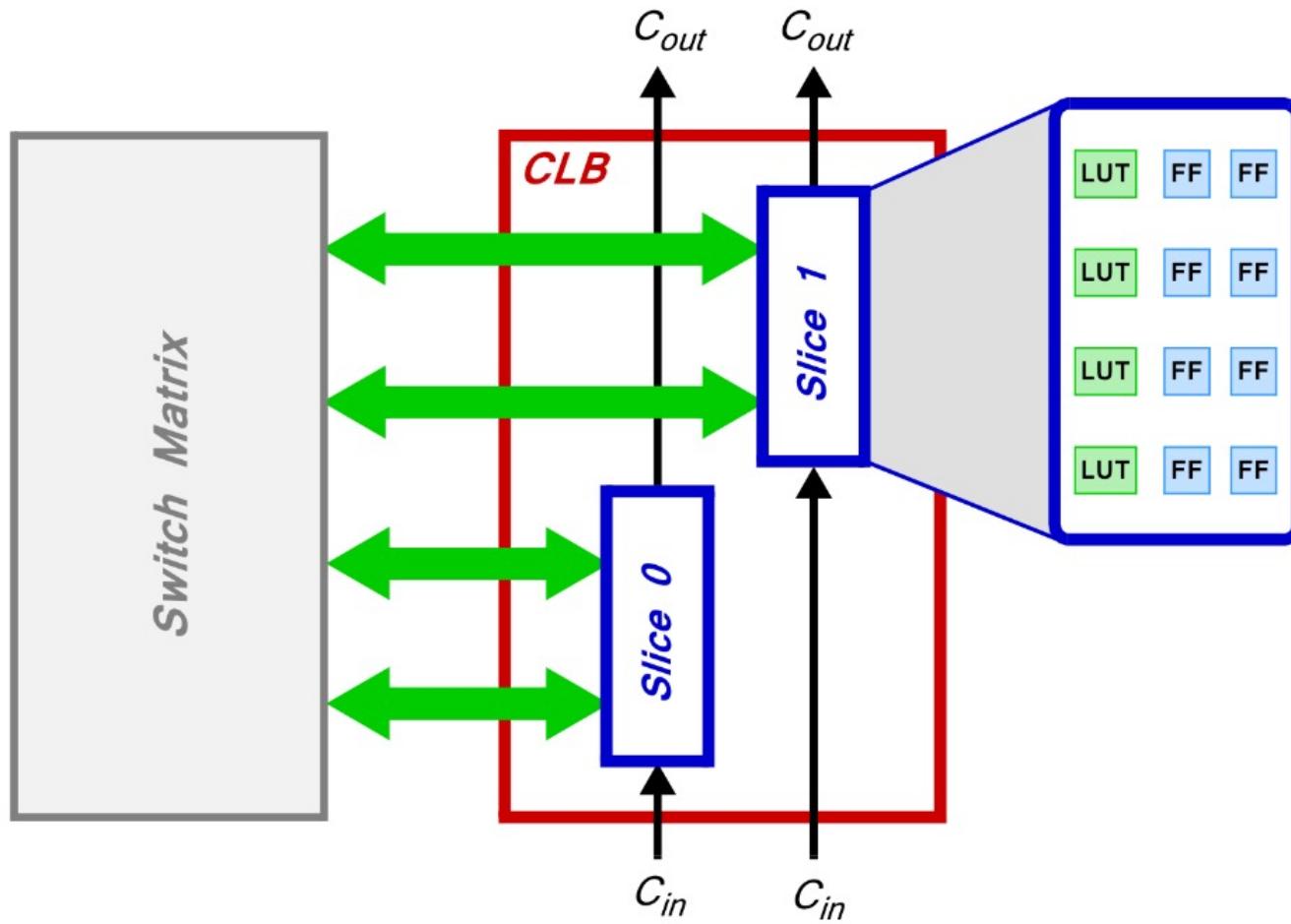


Figure 2.5: The logic fabric and its constituent elements

# CLB



# Familia Zynq-7000

Table 2.3: Zynq-7000 family members

	Z-7010	Z-7015	Z-7020	Z-7030	Z-7045	Z-7100	
Processor	Dual core ARM Cortex-A9 with NEON and FPU extensions						
Max. processor clock frequency	866MHz			1GHz			
Programmable Logic	Artix-7			Kintex-7			
No. of FlipFlops	35,200	96,400	106,400	157,200	437,200	554,800	
No. of 6-input LUTs	17,600	46,200	53,200	78,600	218,600	277,400	
No. of 36Kb Block RAMs	60	95	140	265	545	755	
No. of DSP48 slices (18x25 bit)	80	160	220	400	900	2020	
No. of SelectIO Input/Output Blocks <sup>a</sup>	HR: 100 HP: 0	HR: 150 HP: 0	HR: 200 HP: 0	HR: 100 HP: 150	HR: 212 HP: 150	HR: 250 HP: 150	
No. of PCI Express Blocks	-	4	-	4	8	8	
No. of serial transceivers	-	4	-	4	8 or 16 <sup>b</sup>	16	
Serial transceivers maximum rate	-	6.25Gbps	-	6.6Gbps / 12.5Gbps <sup>c</sup>	6.6Gbps / 12.5Gbps <sup>b</sup>	10.3Gbps	

a. Depends on the package; maximum numbers are shown here. HR = High Range, HP = High Performance.

b. Depends on package chosen.

c. Depends on the speed grade of the device.

# Input/Output

- SelectIO Resources:
  - Organizados en bancos de 50 IOBs, cada uno de ellos contiene un pad con la conexión física al exterior
  - Pueden ser
    - HP (High Performance) : para “interfacear” con memoria o otros chips con requerimientos de velocidad
    - o High Range (HR): soporta amplia gama de dispositivos
- Otras interfaces (según dispositivos)
  - GTX Transceivers, bloques IP hard embebidos para altas velocidades

# Input/Output

- Conversores analógicos digitales: XADC
- Clocks: el PL recibe 4 señales de reloj del PS, y tiene la capacidad de generar y distribuir su propias señales de reloj independientemente del PS
- Programación y Debug: puertos JTAG

# Interfaces e Interconexiones

- Interface: conexión punto a punto para pasar datos, direcciones y señales de control entre master y slaves del sistema
- Interconexión: Maneja el tráfico y lo dirige hacia las interfaces.

# Interfaces PS-PL

- Conjunto de 9 interfaces basadas en AXI, (standard para interfaces de dispositivos y bloques IP)

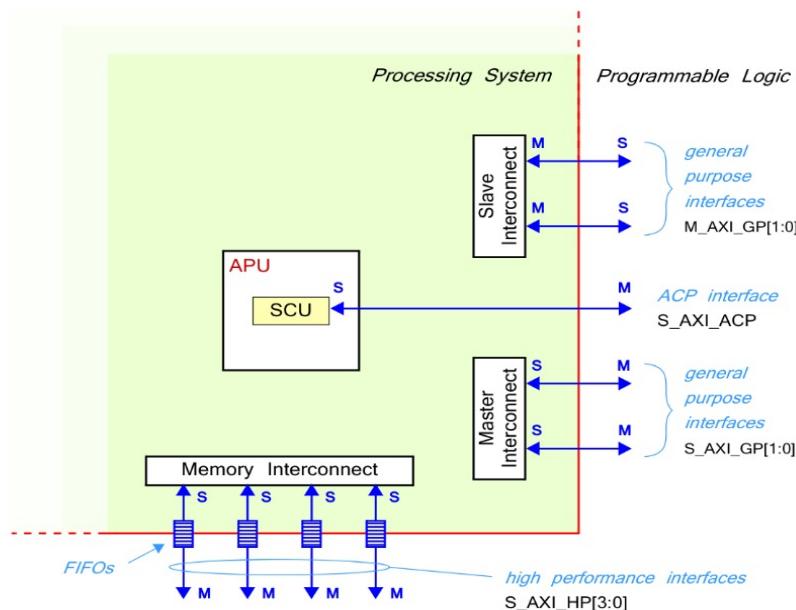


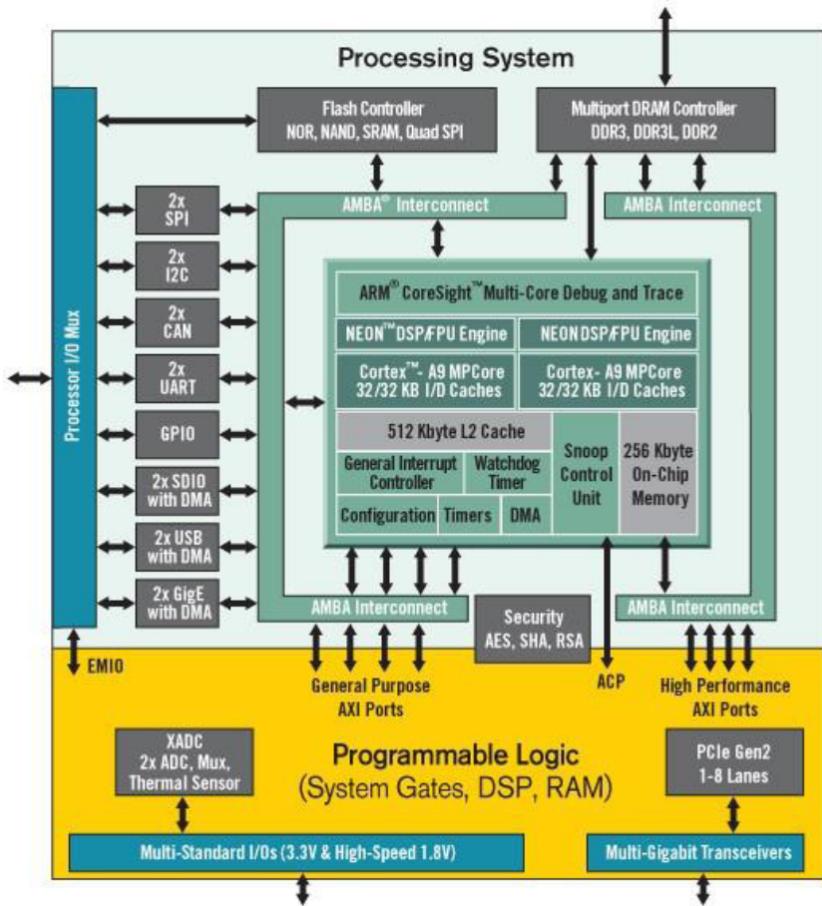
Figure 2.9: The structure of AXI interconnects and interfaces connecting the PS and PL

# Interfaces PS-PL

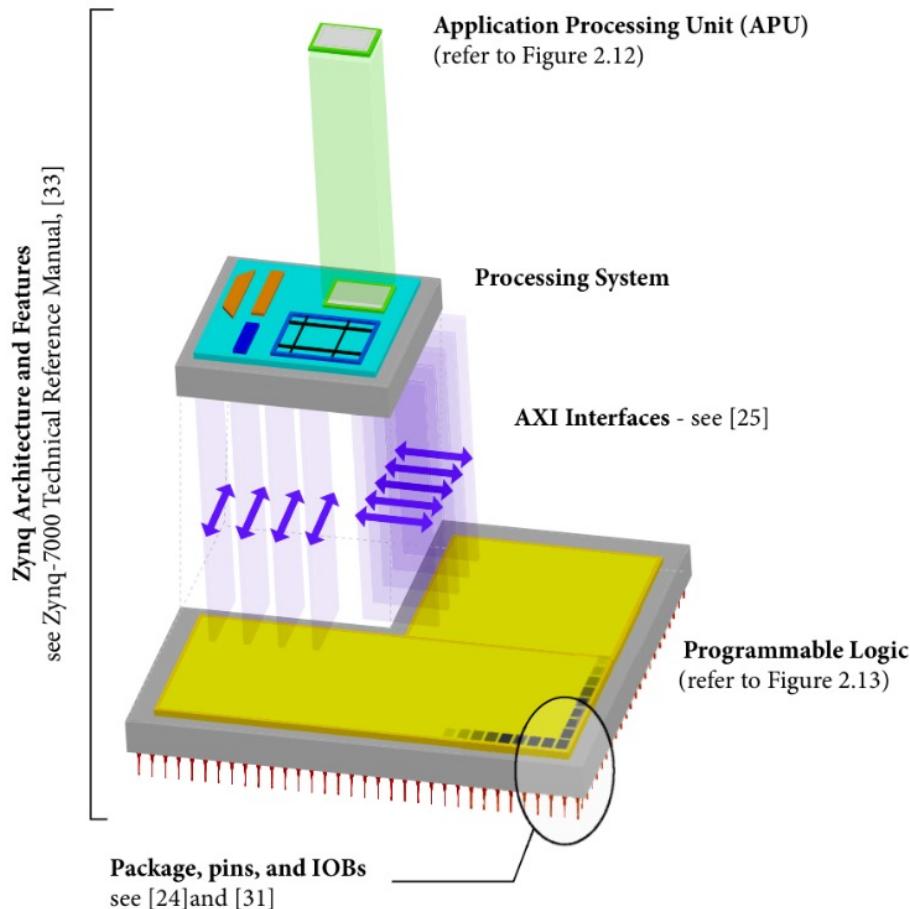
Interface Name	Interface Description	Master	Slave
M_AXI_GP0	General Purpose (AXI_GP)	PS	PL
M_AXI_GP1		PS	PL
S_AXI_GP0	General Purpose (AXI_GP)	PL	PS
S_AXI_GP1		PL	PS
S_AXI_ACP	Accelerator Coherency Port (ACP), cache coherent transaction	PL	PS
S_AXI_HP0	High Performance Ports (AXI_HP) with read/write FIFOs.  (Note that AXI_HP interfaces are sometimes referred to as AXI Fifo Interfaces, or AFIs).	PL	PS
S_AXI_HP1		PL	PS
S_AXI_HP2		PL	PS
S_AXI_HP3		PL	PS

# AXI

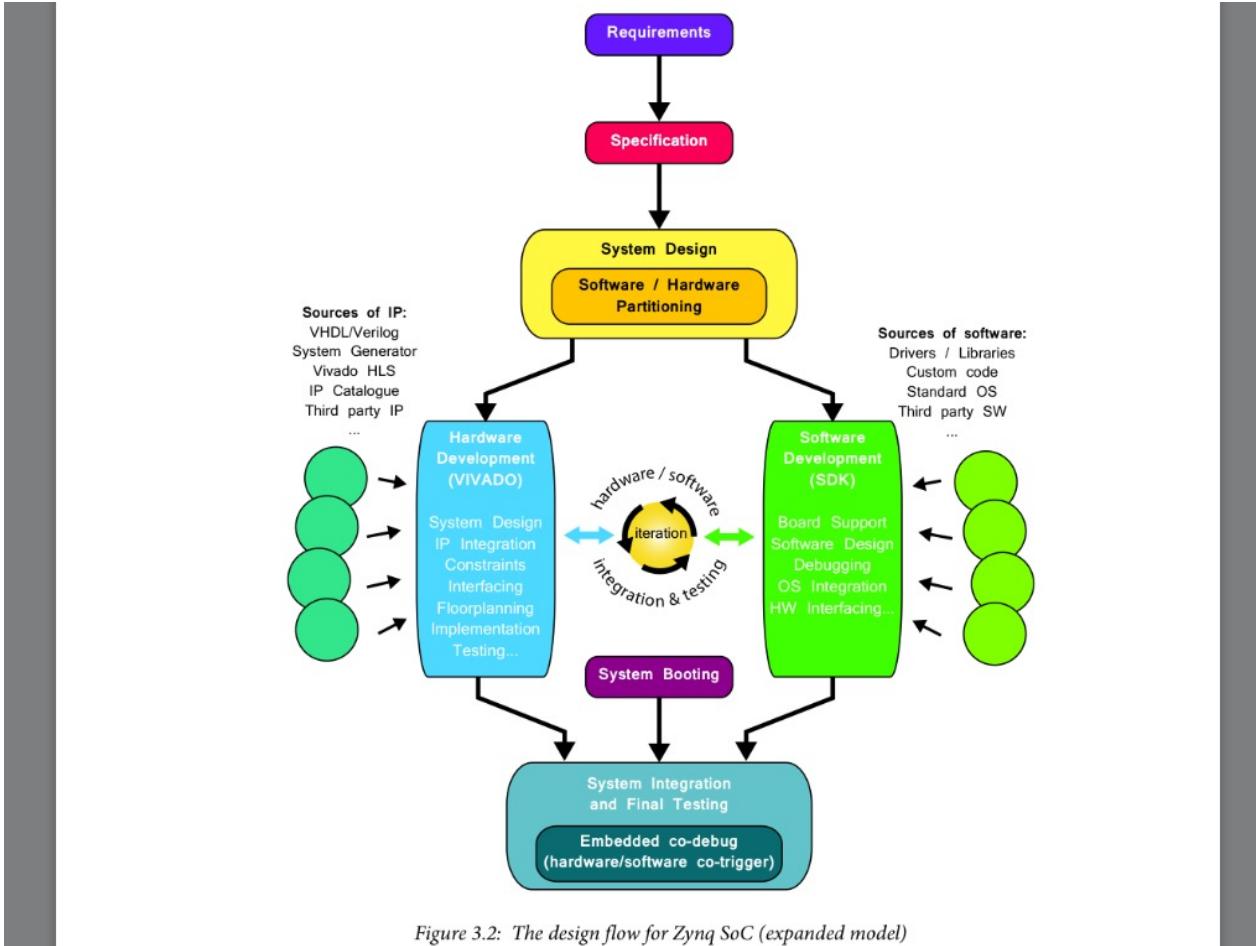
Dentro del PS, Interfaces AXI se usan para conectar los núcleos ARM con memoria caché, SCU, OCM, y en general con otras interconexiones.



# Arquitectura Zynq



# Diseño de SoCs con Zynq: Vivado Design Suite



# VIVADO IDE: Diseño del Hardware

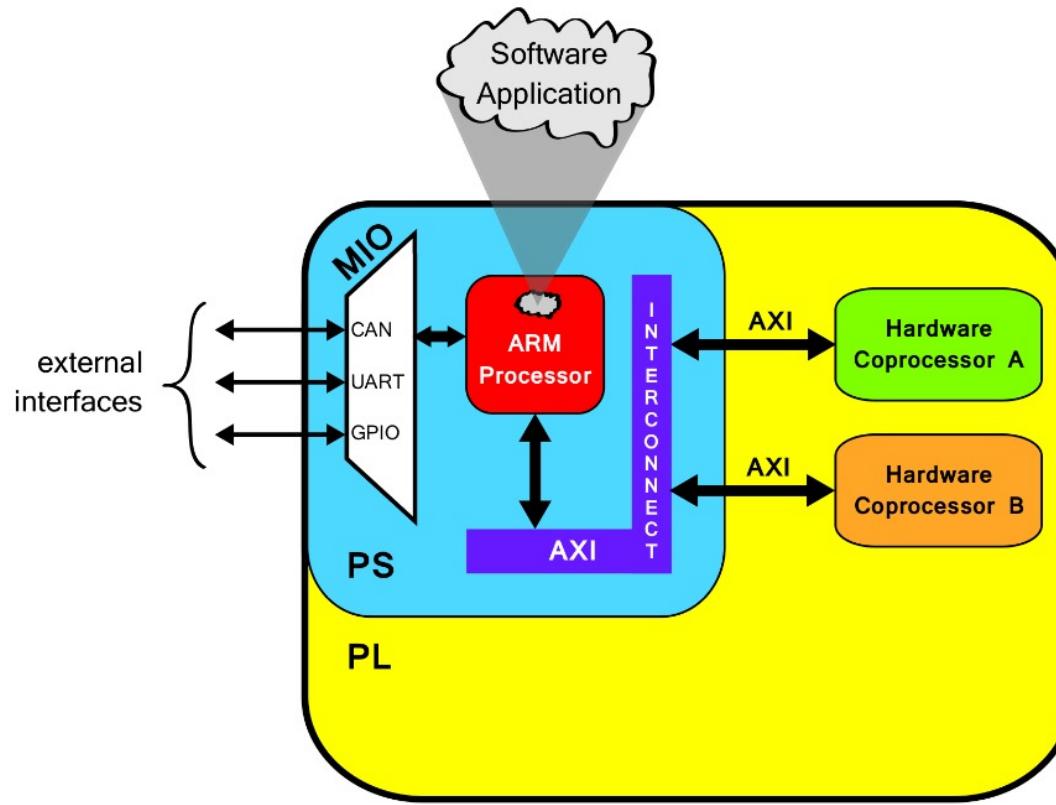


Figure 3.3: Conceptual diagram of an example hardware system with MIO

# VIVADO IDE: IP Integrator

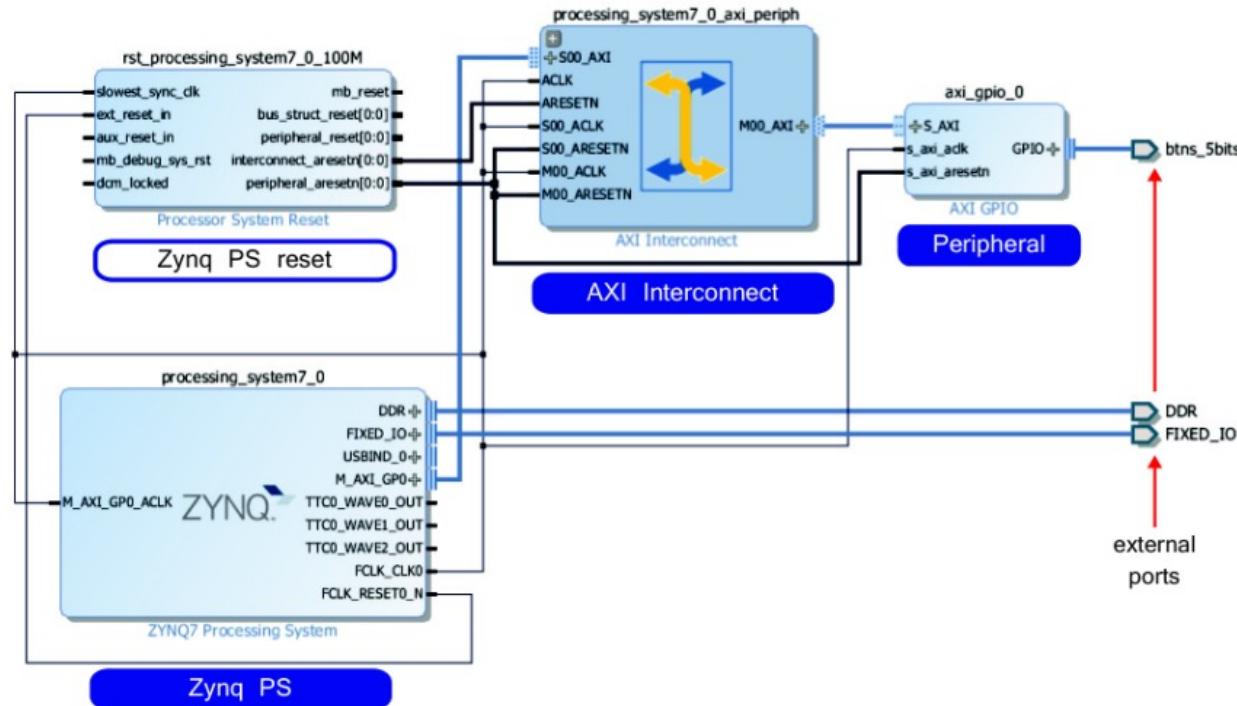


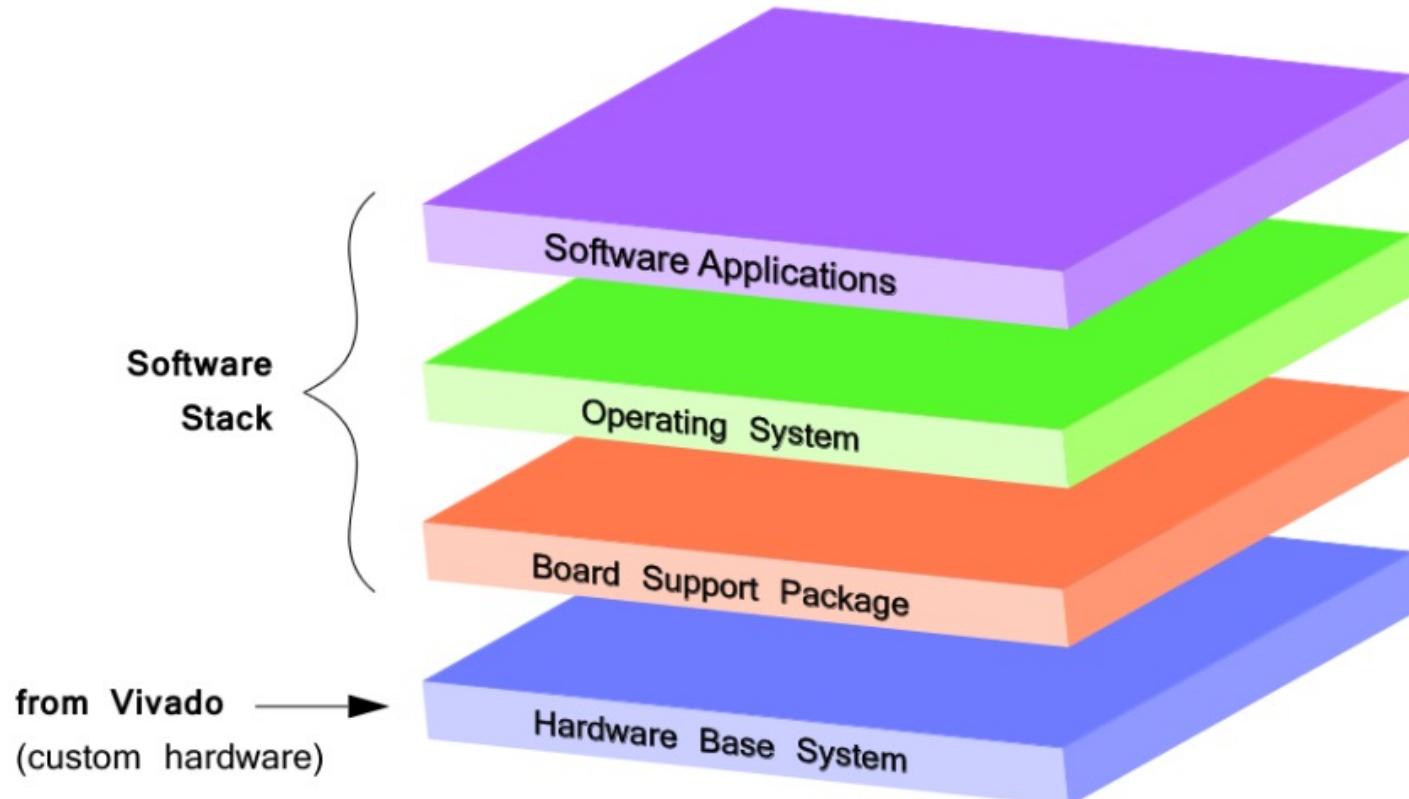
Figure 3.4: An example IP Integrator block diagram

# Vivado IP Integrator

PL:

- Creación de IP’s siguiendo el formato standard de la industria IP-XACT
- IP’s provenientes de librerías de Xilinx
- Métodos (automatizados) para integrar los IP’s al sistema
- Uso de HLS (High Level Synthesis) para crear componentes desde niveles de descripción mas altos

# VIDADO SDK



*Figure 3.5: Hardware and software layers of a Zynq design*

# SDK: Board Support Package

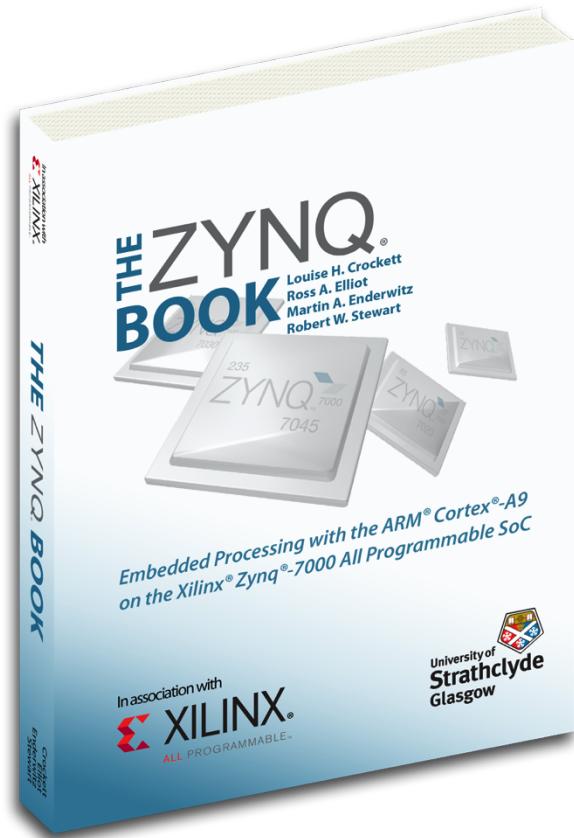
- Se elije el SO: "light", RTOS,Linux, "baremetal"(cuando se comunica directo con el hardware) etc
- Incluye device drivers para todos los dispositivos del sistema base (el que se exporta del VIVADO IDE)
- Debe refrescarse si se introducen cambios en el Base System (hardware)
- La aplicación corre arriba del BSP y SO

# SDK: herramientas de testing y debugging y profiling

- Los diseños que usan tanto PL como PS contienen dos partes:
  - El archivo .bit que programa la parte PL
  - El archivo .elf para el PS
- Herramientas para co-debugging
- Herramientas para debugging del software en el hardware

# The Zynq Book

- <http://www.zynqbook.com/>



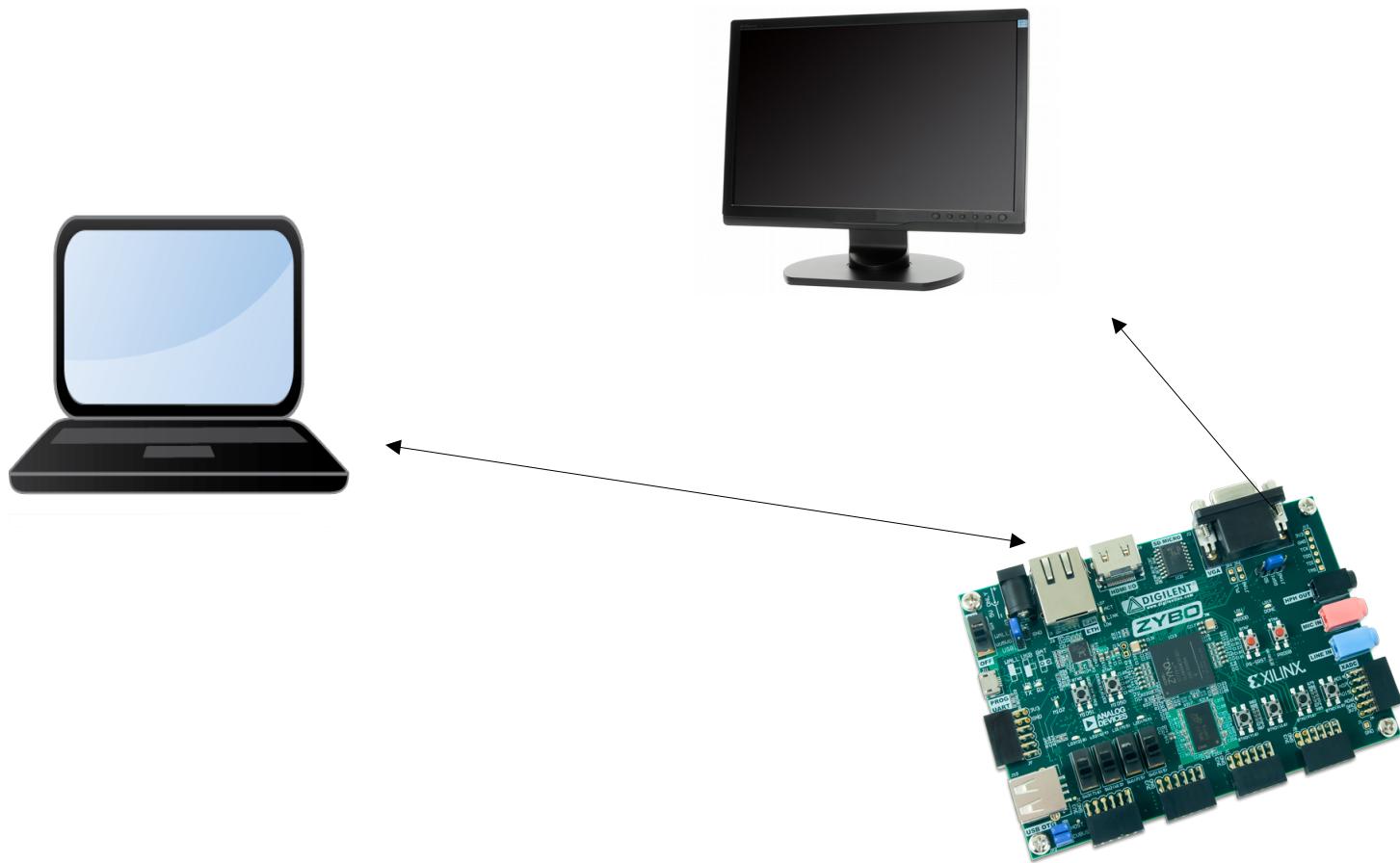
# Tutorial

Construcción de un sistema de  
procesamiento de video utilizando  
Zybo y Vivado

# Qué vamos a hacer ?

- Construir un sistema que recibe video desde una computadora, lo procesa y lo muestra en un monitor VGA.
  - Ethernet y VGA en Zybo
  - Zynq: Integración PS ↔ PL
  - Uso de la Suite Vivado
  - High Level Synthesis

# El Sistema a Construir



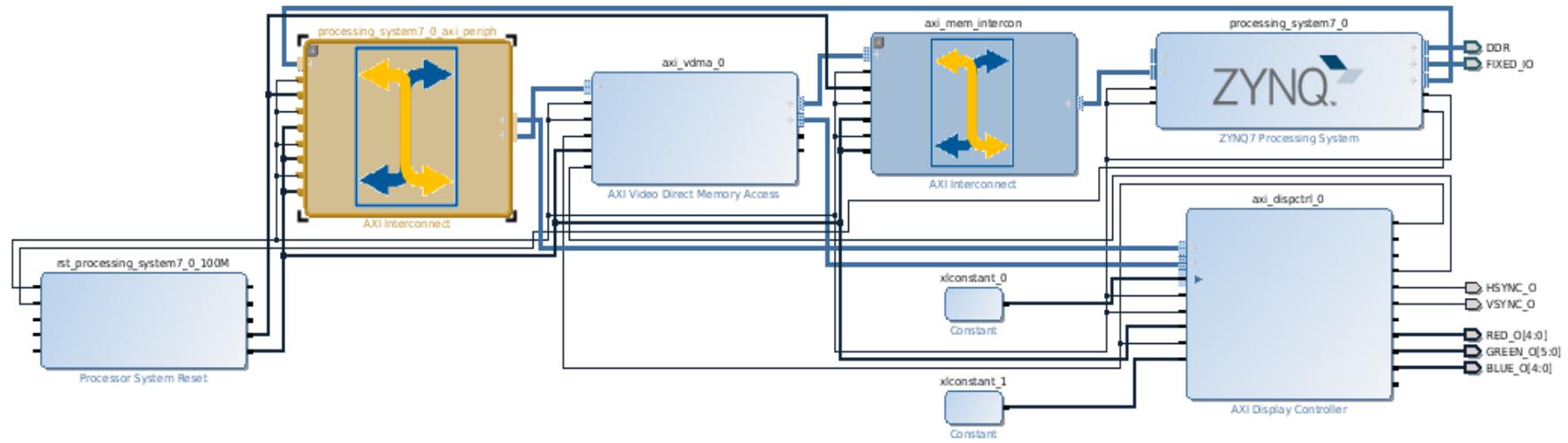
# Herramientas

- Suite **Vivado**:
  - Vivado IP Integrator
  - Vivado SDK
  - Vivado HLS (High Level Synthesis)
- Placa de desarrollo Digilent **Zybo**
- Computadora servidor de video (Escrito en Python)

# Paso a Paso

- 1.Crear la plataforma de hardware en Vivado IP Integrator
- 2.Programar el software en Vivado SDK
- 3.Test de la aplicación
- 4.Optimización mediante Vivado HLS

# ■ Crear la plataforma de hardware



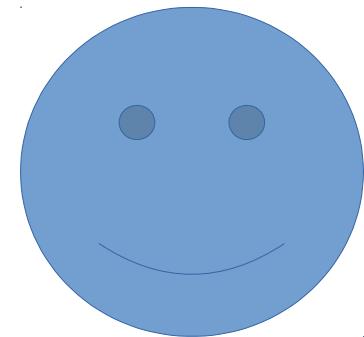
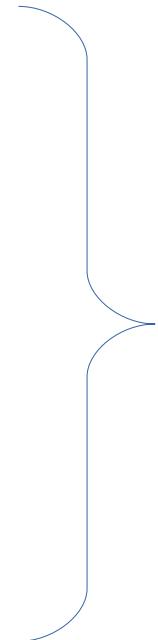
- Configurar Zynq
  - Integrar otros IPs
  - Interfaces externas
- } Generar Bitstream

# Programar el software

- Configurar Board Support Package:
    - Drivers para los IPs
    - Soporte TCP/IP
  - Programar la aplicación:
    - Obtener video vía TCP
    - Procesar el video
    - Mostrar video en monitor
- 
- Archivo ELF

# Test de la Aplicación

- Servidor (PC):
  - Configurar Red
  - Correr el servidor en Python
- Cliente (**ZYBO**):
  - Programar FPGA con
  - Bitstream
  - UART
  - Correr programa (ELF)



# Optimización (Vivado HLS)

- Convertir el algoritmo de procesamiento en “hardware”, vamos a crear un nuevo IP
  - Traducción C → HDL
  - Definir interfaces
  - Medir utilización de recursos
- Integrar el nuevo IP a la plataforma
- Modificar la aplicación para usar el nuevo IP



IP



Solución procesando video en hardware

# Muchas Gracias



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA



*Gracias a Fernando Martinez Vallina  
por la gestión de licencias Vivado  
XUP.*

Patricia Borensztein <[patricia@dc.uba.ar](mailto:patricia@dc.uba.ar)>  
Miguel García <[miguel.garcia@gmail.com](mailto:miguel.garcia@gmail.com)>