IEE 2463
Sistemas Electrónicos Programables
Lecture: Zynq and AXI Overview

Architecture of Programmable Systems

POWER & ENERGY CONVERSION LABORATORY

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

PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

peclab
POWER & ENERGY CONVERSION LABORATORY

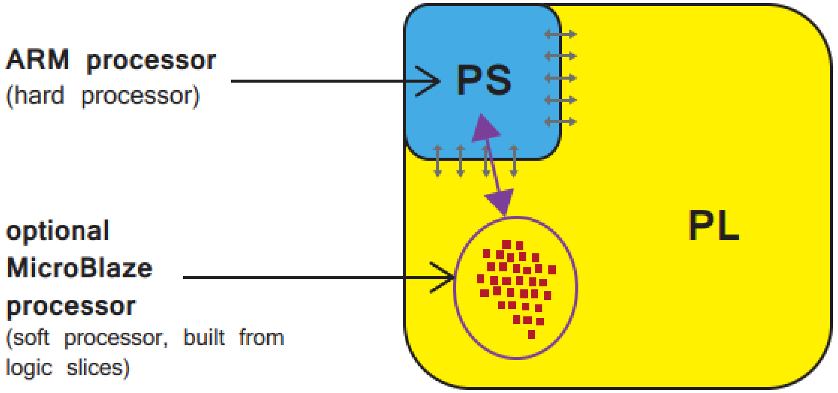
PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Zynq 7000 Device
Overview

This section provides an brief overview of the Xilinx Zynq Device

SoC Architecture and On-Chip Interconnect Standards

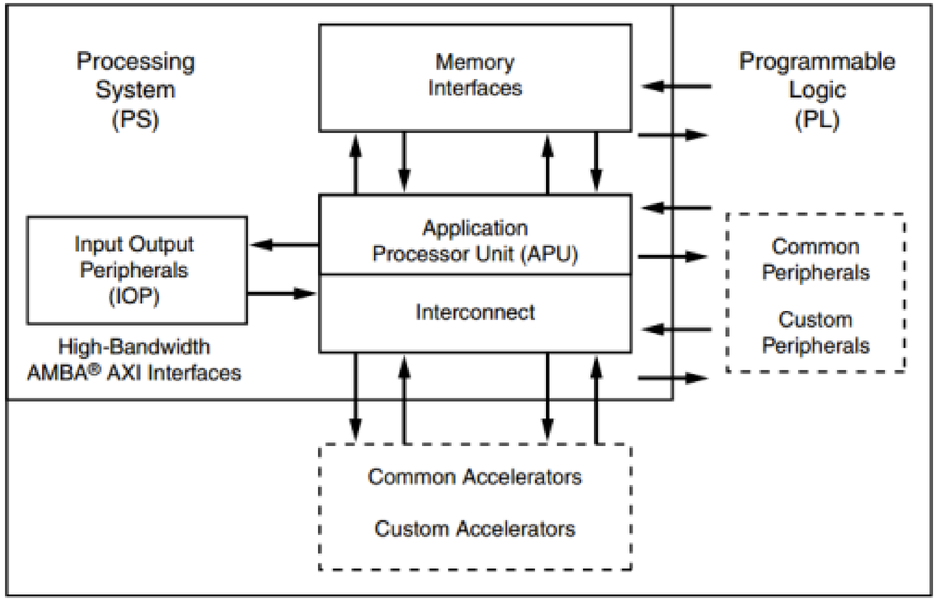
Internal Architecture



- Hard Processor: Dual-core ARM Cortex A9 processor
- Soft Processor: Xilinx Microblaze is built into the PL.

(Note: you can even create your own processor!)

- Several soft processors can be used within PL, which can coordinate specific low-level functions. Releasing tasks of the main ARM processor.
- PS and PL are communicated through an internal protocol named ***AXI.***

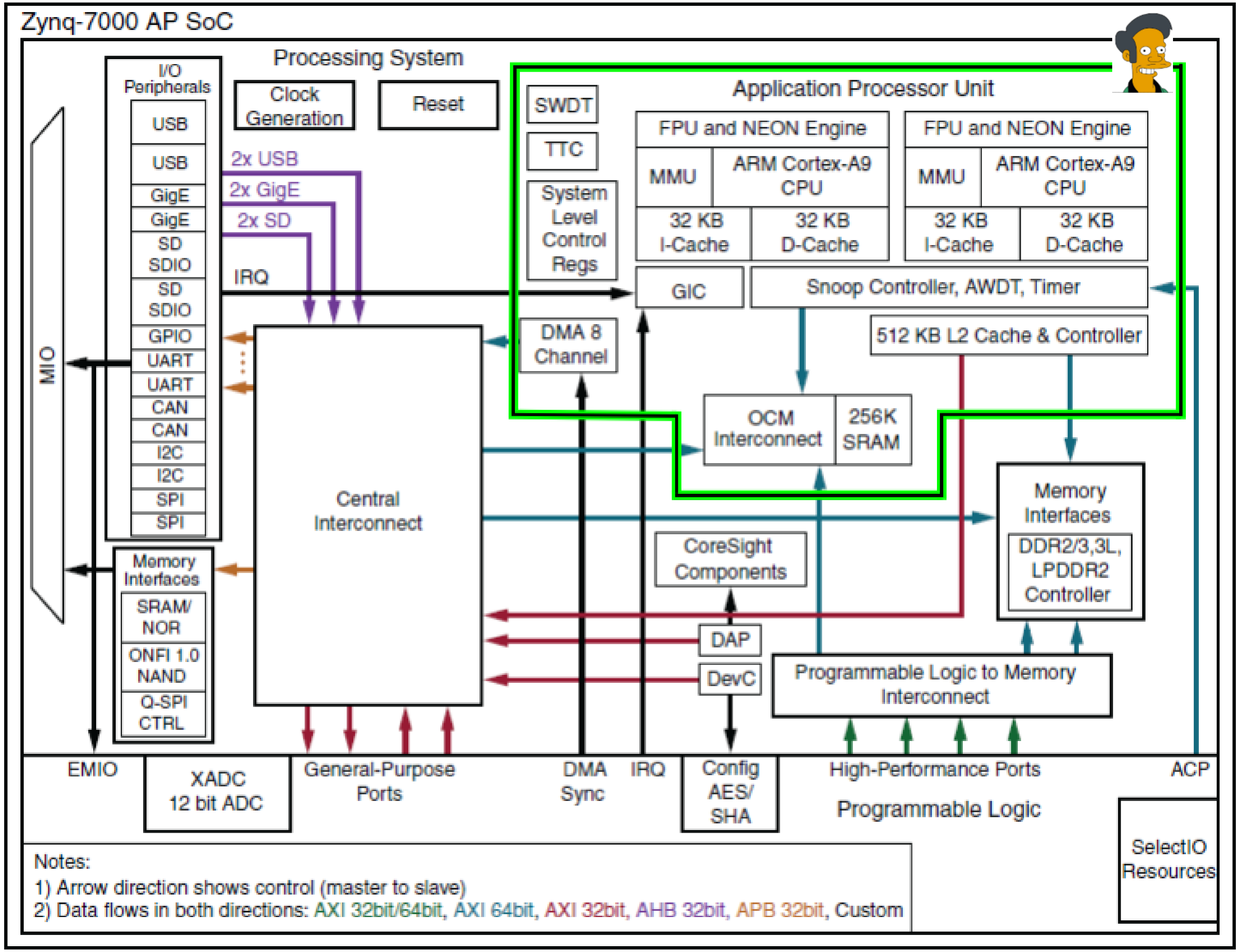


High-level
Block Diagram

Basic Structure of Zynq 7000 Devices.
Reference manual here!

SoC Architecture and On-Chip Interconnect Standards

Internal Architecture



Notes:
1) Arrow direction shows control (master to slave)
2) Data flows in both directions: AXI 32bit/64bit, AXI 64bit, AXI 32bit, AHB 32bit, APB 32bit, Custom

DS190_01_030713
© Xilinx

Application Proccesing Unit (APU): Contains the ARM and other processing resources and peripheral. Another term for referring to a Microprocessor (or part of it).

PS: Mainly contains APU, Memory interfaces, I/O peripherals and connection to PL (central interconnect).

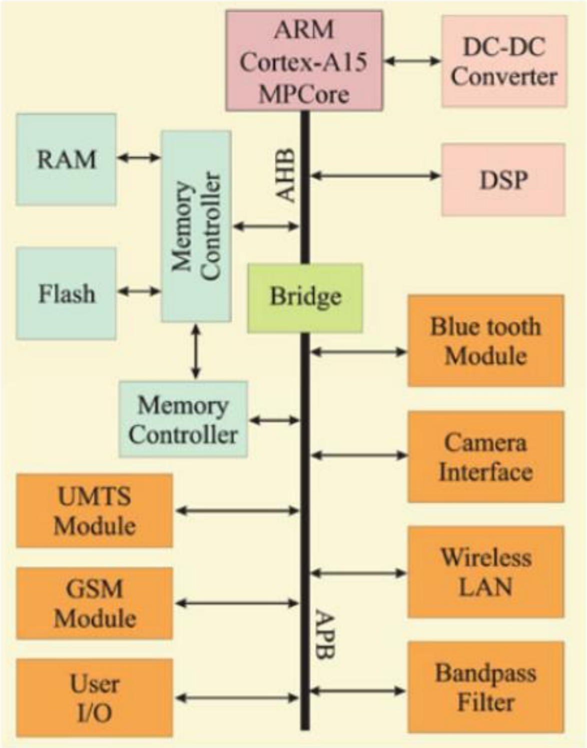
PL: Besides logic gates, contains 12bits XADC for analog to digital conversion and connection to PS (ACP, HP ports).

There is a need of internal buses architecture to interconnect all components of a SoC, some are very fast and other rather slow. **How to do that?(See next slide)**

Technical Reference Manual Zynq-SoC here.

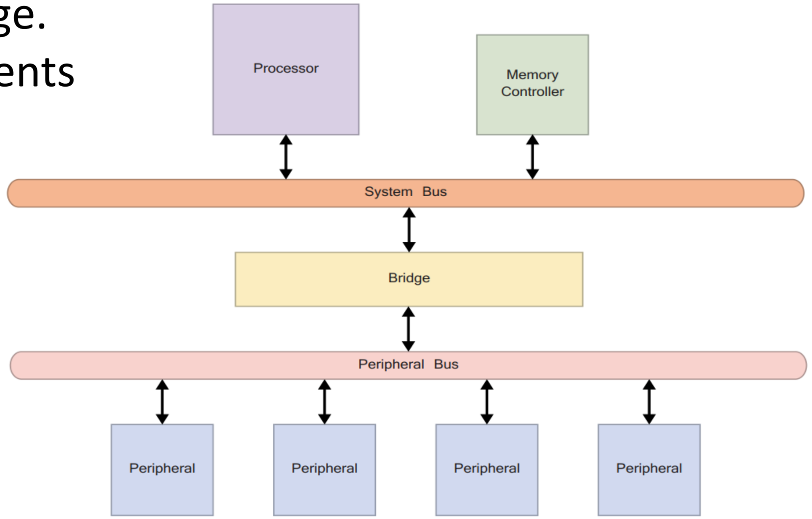
SoC Architecture and On-Chip Interconnect Standards

Advanced Microcontroller Bus Architecture (AMBA)



AMBA-based mobile phone SoC

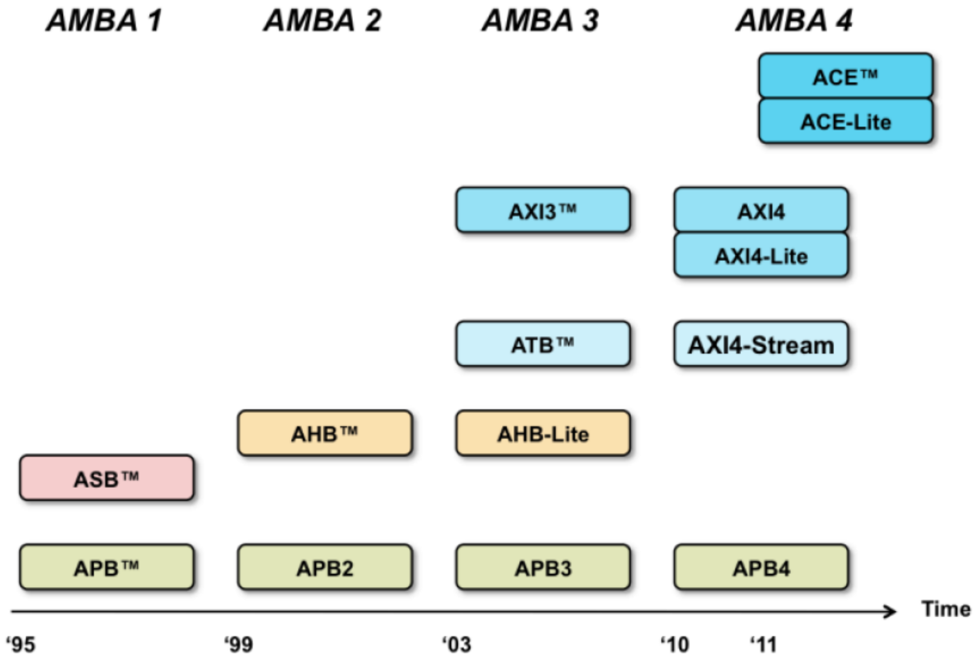
- Introduced by ARM in 1996 to interconnect several IPs.
- It uses 2-layer hierarchical bus topology. High and low performance (HP and LP) buses interfaced by a bridge.
- HP Bus: Processor cores, memories and HP components
- LP Bus: Ethernet, USB, UART,etc.
- AMBA consist of:
  - Masters: Connected to HP bus, allow to initiate transfer of data.
  - Slaves: Connected to LP bus
  - Bus Arbiter: Allow only one master to initiate a transfer art the same time.
  - Central decoder: Provides selects signals for peripheral based on addresses.
- Communication is based on “transactions”. One transaction const of:
  - Securing the bus, commencing operation and completing operation (read/write).



AMBA-based interconnection (generic representation)

Advanced Microcontroller Bus Architecture (AMBA)

- Buses in AMBA version 1.0:
  - Advanced System Bus (ASB) - HP Bus
  - Advanced Peripheral Bus (APB) – LP Bus
  - APB bridge act as cross-over and handshake between ASB and APB.
  - ASB and APB support 8, 16 and 32 bits data width.
- AMBA version 2.0
  - Introduces Advanced High-Performance Bus (AHB)
  - Allow burst data transfer and Split transactions-
  - AHB allow 32, 64 and 128 bit data width
  - Incorporates protection scheme to distinguish between priviledged and non-priv. access mode.
- AMBA version 3.0
  - Advanced eXtensible Interface (AXI3).
  - Provides point-to-point interconnect between master and slave. (no bus sharing!)
  - Enables higher bandwidth and lower latency design.

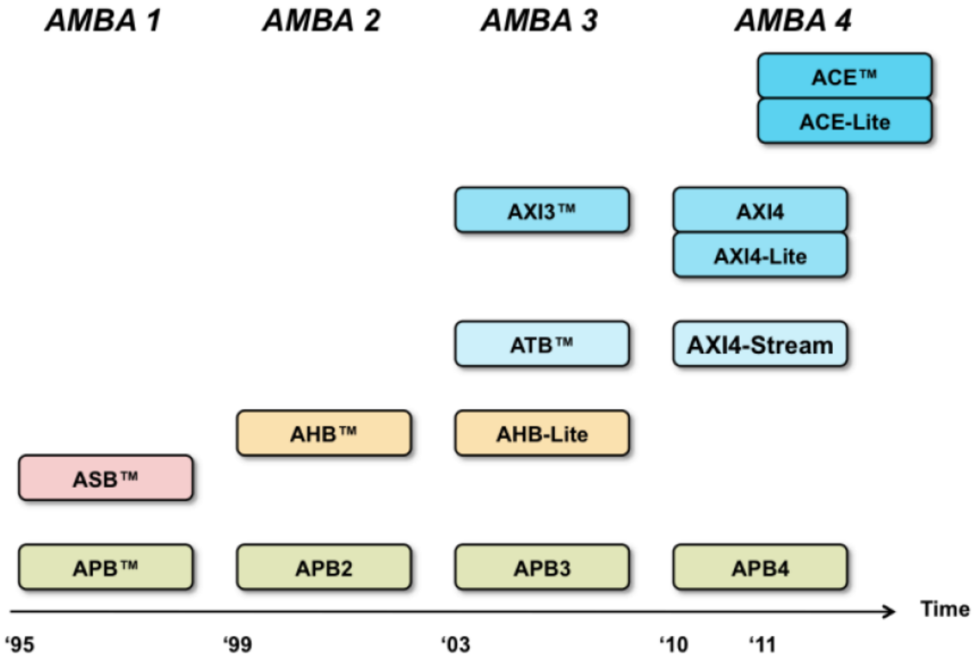


AMBA Evolution

SoC Architecture and On-Chip Interconnect Standards

Advanced Microcontroller Bus Architecture (AMBA)

- AMBA version 3.0
  - Introduced Advanced Trace Bus (ATB) and ATB-Lite
  - ATB is part of CoreSight. An on-ship debug to IP verification. (ATB-Lite; only one master)
- AMBA version 4.0
  - AXI4 is AXI3 updated.
  - AXI-Stream: allows unlimited burst of data.
  - AXI-Lite: simplified versión of AXI, allow one master.
  - Advanced Coherency Extension (ACE):
    - Is a hardware management of cache memory among multiple cores.
    - No need of maintain coherency in software.
    - Save processor and bus usage.
  - ACE-Lite: Only one-way coherency for components without DMA(we see this later).



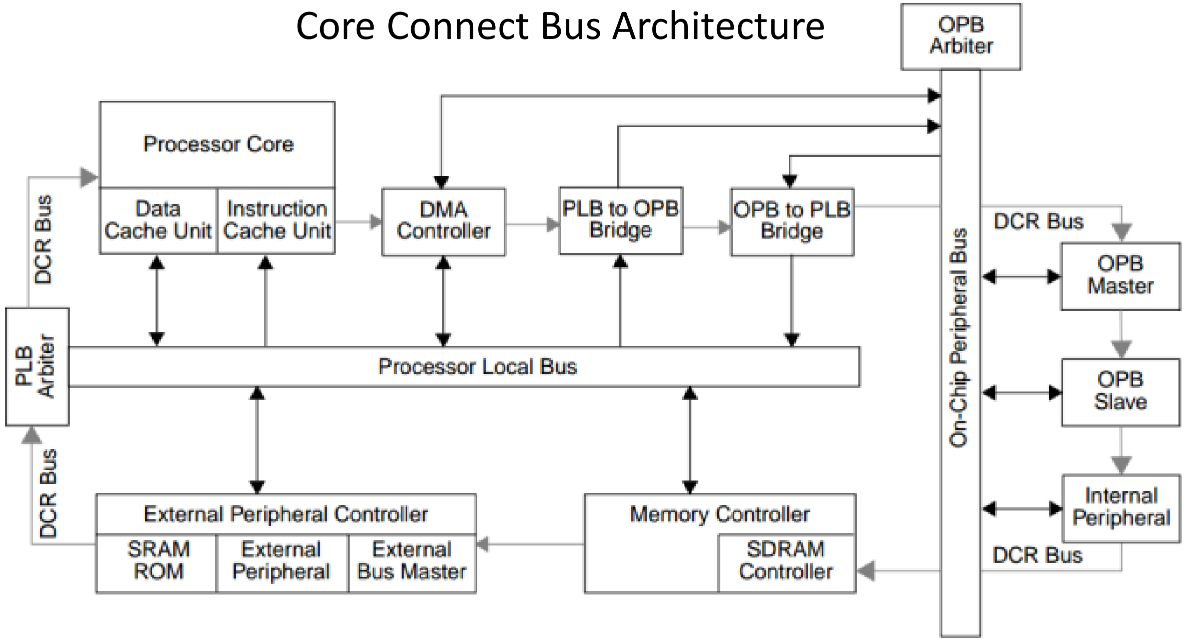
AMBA Evolution

SoC Architecture and On-Chip Interconnect Standards

Core Connect

- Similar to AMBA, CoreConnect is a bus architecture for SoCs proposed by **IBM**
- It is composed of three buses:
  - Processor Local Bus (PLB)
  - On-Chip Peripheral Bus (OPB)
  - Data Control Register Bus (DCR)
- PLB: High bandwidth low latency, 16,32,64 bits data width (extensible to 256bits). Processor, DMA, memories, support up tp 16 master and unlimited slaves.
- Masters and slaves connected to PLB have separate buses for adress, data read and data write.
- Support burst transfer, Split transaction and address pipelining.
- OPB uses 32bit data width, separate address and data bus, a synchronous operation, multiple masters.
- Bridges are used to communicate PLB with OPB.

Core Connect Bus Architecture

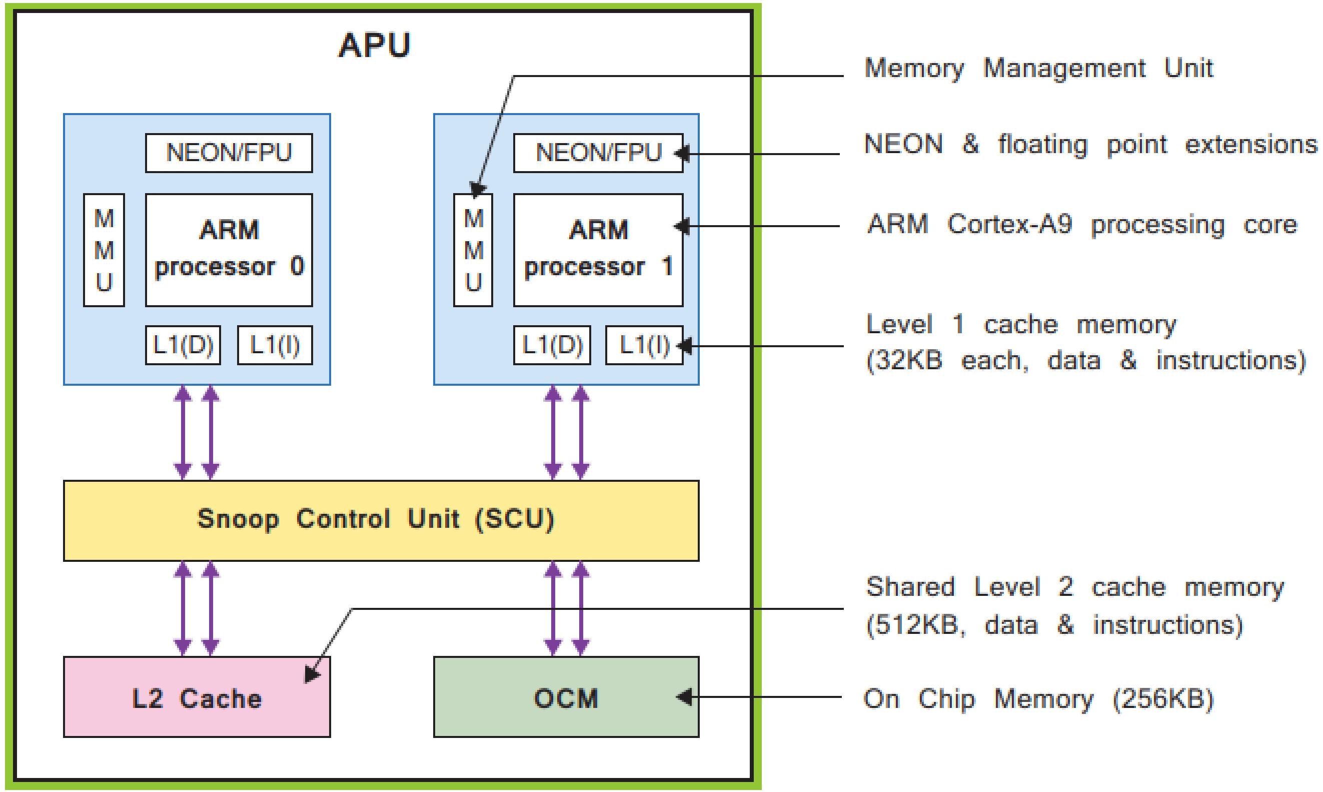


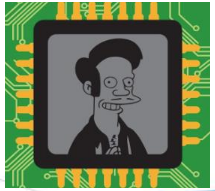
- DCR provides status and configuration data. It is 10bit adress bus and 32bit dará bus.

Note: CoreConnect finds its use inPowerPC400 family, for example, PowerPC 440 incorporates CoreConnect bus standard to interface components inside a PowerPC. (Xbox, Apple, intend, Motorola, etc..read more)

SoC Application Processing Unit (APU)

PS- (APU)


APU
NEON/FPU
M M U
ARM processor 0
L1(D)
L1(I)
NEON/FPU
M M U
ARM processor 1
L1(D)
L1(I)
Snoop Control Unit (SCU)
L2 Cache
OCM
Memory Management Unit
NEON & floating point extensions
ARM Cortex-A9 processing core
Level 1 cache memory (32KB each, data & instructions)
Shared Level 2 cache memory (512KB, data & instructions)
On Chip Memory (256KB)



Each APU contains two blocks of:

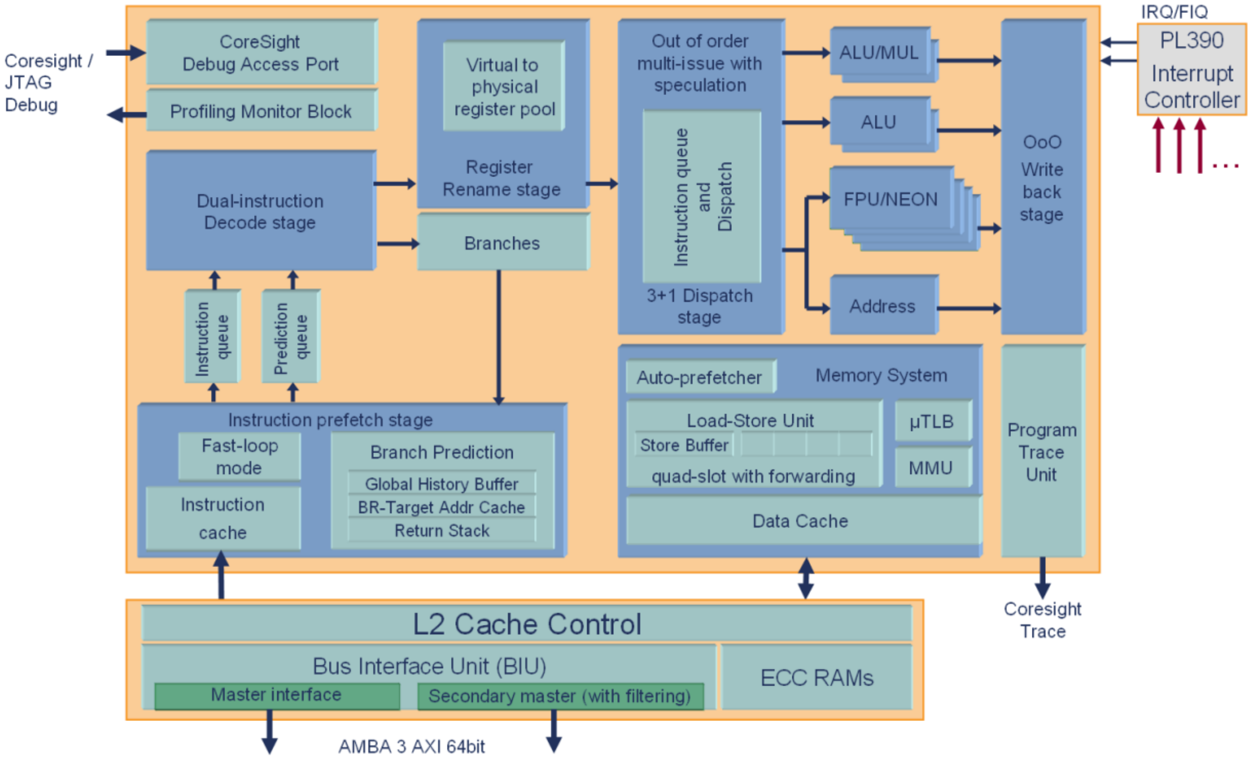
- **ARM** processor (its set of instructions)
- Floating Point Unit (**FPU**) and **NEON** Media Processing Engine (**MPE**). (computation units known as **co-processors**)
- Memory Management Unit (**MMU**)
- Level 1 Cache Memory for Instructions (**L1(I)**)
- Level 1 Cache Memory for Data (**L1(D))**

Also:

- **OCM**: Additional memory for data.
- Level 2 Cache Memory (**L2**) (shared)
- SCU: Responsible for:
  - Maintaining memory coherency between **L1(D)** and **L2.**
  - Control access to L2 requested arbitrarily by ARM0 and ARM1.
  - Control transaction of date between PS and PL via Accelerator Coherency Port **(ACP).**
- **Timers** and **Interrupts controller.** Read this link for its details.

SoC Application Processing Unit (APU)

PS-Extra Note About ARM

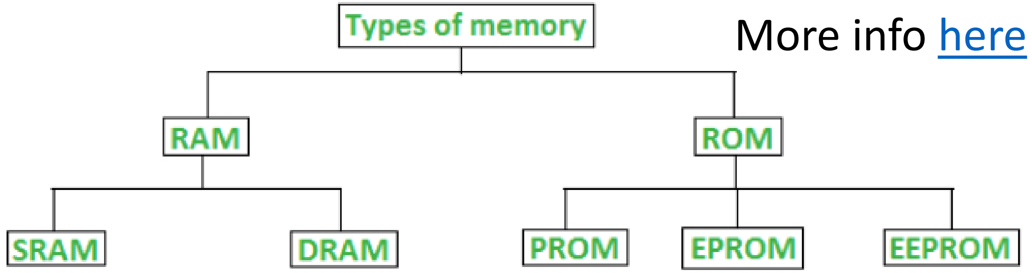


Cortex A-9 microarchitecture structure

- ARM's business model is to license Original Equipment Manufacturers (OEMs), such as Xilinx, to utilize ARM processor IP within the devices they develop (in this case, Zynq).
- Zynq has ARM Cortex A9. Architecture ARM v7.
- OEMS can configure special features for its ARM. For instance: size of **L1**, number of cores or inclusion of optional extensions such as NEON and FPU.
- ARM documentation find at ARM website describes in detail the APU within a Zynq.
- Zynq-7000 uses the r3p0 revision of ARM Cortex-A9, which is based on **ARMv7-A.** (important when checking the ARM manual).
- ARM is RISC technology. Small and simple set of instructions and low energy consumption. The best processor for mobile and table applications.
- ARM Cortex can be model A (application); R(real time), M(microcontroller) or X (custom). See here

SoC Application Processing Unit (APU)

Type of Memories



More info here

1. Random Access Memory (RAM) –

•It is also called as *read write memory* or the *main memory* or the *primary memory*.

•The programs and data that the CPU requires during execution of a program are stored in this memory.

•It is a volatile memory as the data loses when the power is turned off.

•RAM is further classified into two types- ***SRAM*** *(Static Random Access Memory)* and ***DRAM*** *(Dynamic Random Access Memory)*.

2. Read Only Memory (ROM) –

•Stores crucial information essential to operate the system, like the program essential to boot the computer.

•It is not volatile.

•Always retains its data.

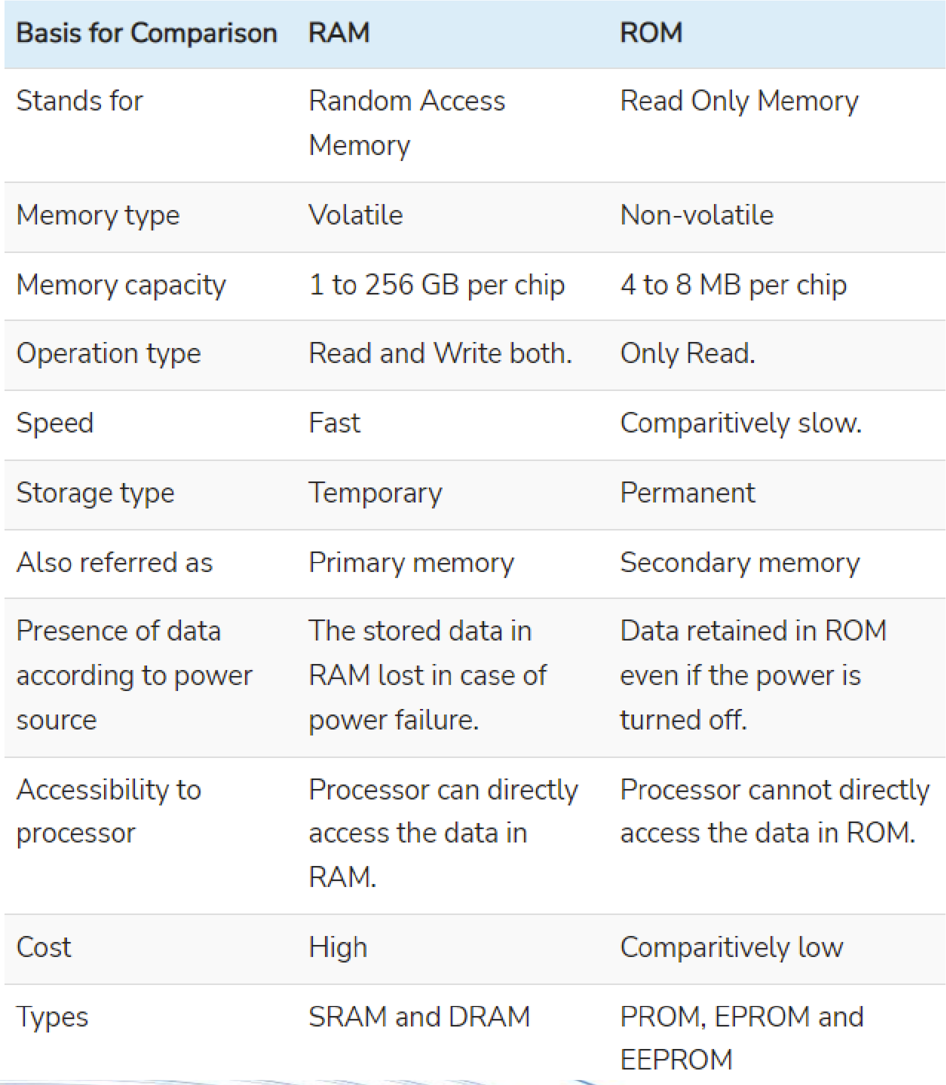
•Used in embedded systems or where the programming needs no change.

•Used in calculators and peripheral devices.

•ROM is further classified into 3 types- ***PROM***, ***EPROM***, and ***EEPROM***.

SoC Application Processing Unit (APU)

Type of Memories



Basis for Comparison	RAM	ROM
Stands for	Random Access Memory	Read Only Memory
Memory type	Volatile	Non-volatile
Memory capacity	1 to 256 GB per chip	4 to 8 MB per chip
Operation type	Read and Write both.	Only Read.
Speed	Fast	Comparitively slow.
Storage type	Temporary	Permanent
Also referred as	Primary memory	Secondary memory
Presence of data according to power source	The stored data in RAM lost in case of power failure.	Data retained in ROM even if the power is turned off.
Accessibility to processor	Processor can directly access the data in RAM.	Processor cannot directly access the data in ROM.
Cost	High	Comparitively low
Types	SRAM and DRAM	PROM, EPROM and EEPROM

info

SoC Application Processing Unit (APU)

Type of Memories

Types of Read Only Memory (ROM) –

1.PROM (Programmable read-only memory) – It can be programmed by user. Once programmed, the data and instructions in it cannot be changed.

2.EPROM (Erasable Programmable read only memory) – It can be reprogrammed. To erase data from it, expose it to ultraviolet light. To reprogram it, erase all the previous data.

3.EEPROM (Electrically erasable programmable read only memory) – The data can be erased by applying electric field, no need of ultraviolet light. We can erase only portions of the chip. Byte-wise erasable, NOR Type Memory.

•**Flash:** Special type of EEPROM. Block-wise erasable, constantly rewritten, suitable for large amounts of data. Based on logic gates. Flash uses NAND-type memory.

.

Types of Read Only Memory (RAM) –

Dynamic RAM (DRAM): Composed of millions of 1-bit memory cells. These are composed of a transistor and a capacitor. The information is held as capacitor charge. To avoid undesirable discharge, DRAM is constantly Reading and rewriting the memory to avoid fade.

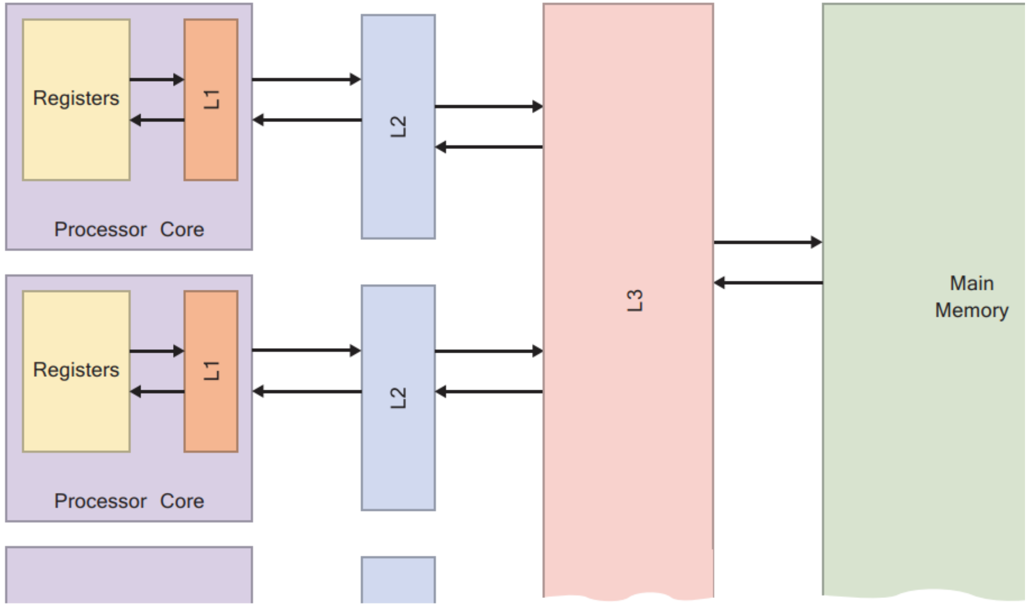
A DDR memory is a Synchronous DRAM (**SDRAM**), synchronized with the clock speed that the microprocessor is optimized for.

Static RAM (SRAM): Uses latches to avoid discharge (4 to 6 latches per bit). Require more space and is more expensive. Does not require refresh and therefore it is much faster than DRAM. Only used in high-speed low-capacity memory. Uses more energy.

More info here

SoC Application Processing Unit (APU)

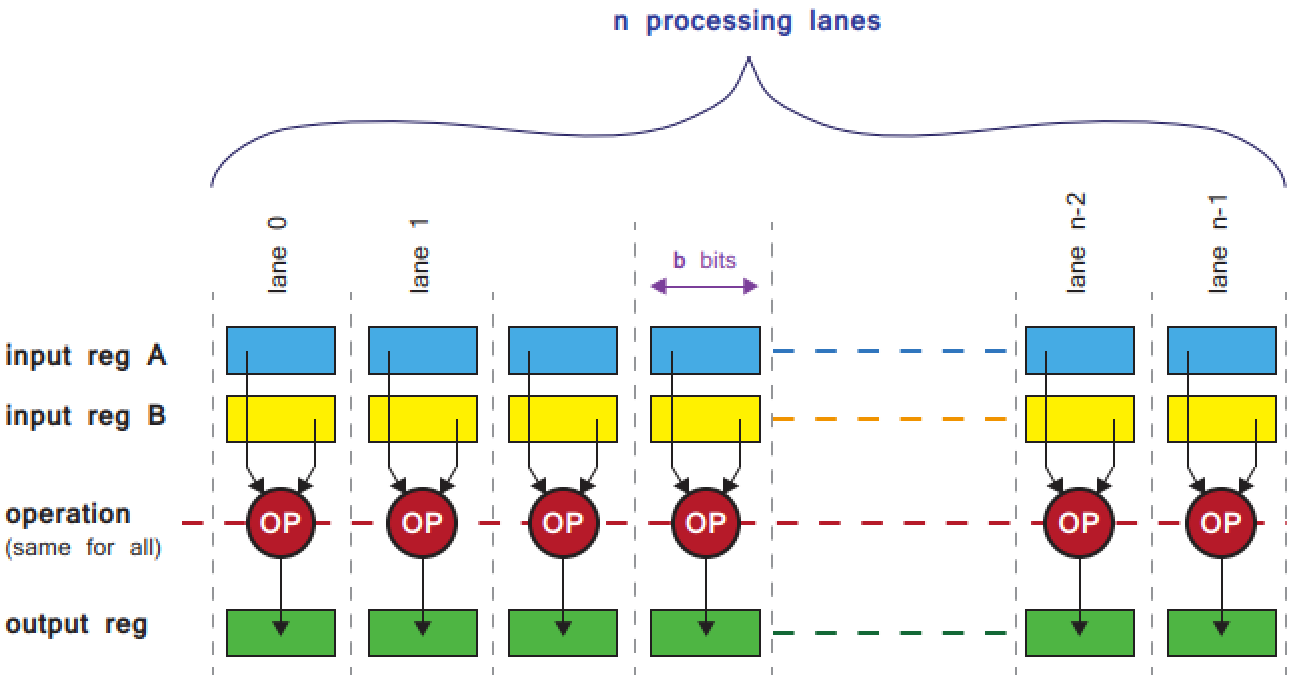
Cache Memory



- Level 1 (**L1**) Cache:
  - Smallest cache (8 to128kB).
  - Built into the fabric of processor core.
  - SRAM type (faster and more expensive vs DRAM)
  - Typ. Divided into Data and Address.
  - Store local copy of frequently accessed data and instructions
- Level 2 (**L2**) Cache:
  - Size of 256 to 1024 kB
  - External to processing core, but extremally close to it.
  - Slower access compared to L1, but still very fast.
  - Unified in one section.
  - DRAM type.
  - L2 constantly read the main memory and store data to then transfer it to L1.
- Level 3 (**L3**) Cache:
  - Largest form of cache (2MB and upwards)
  - Always shared by all processors.
  - DRAM Type

SoC Application Processing Unit (APU)

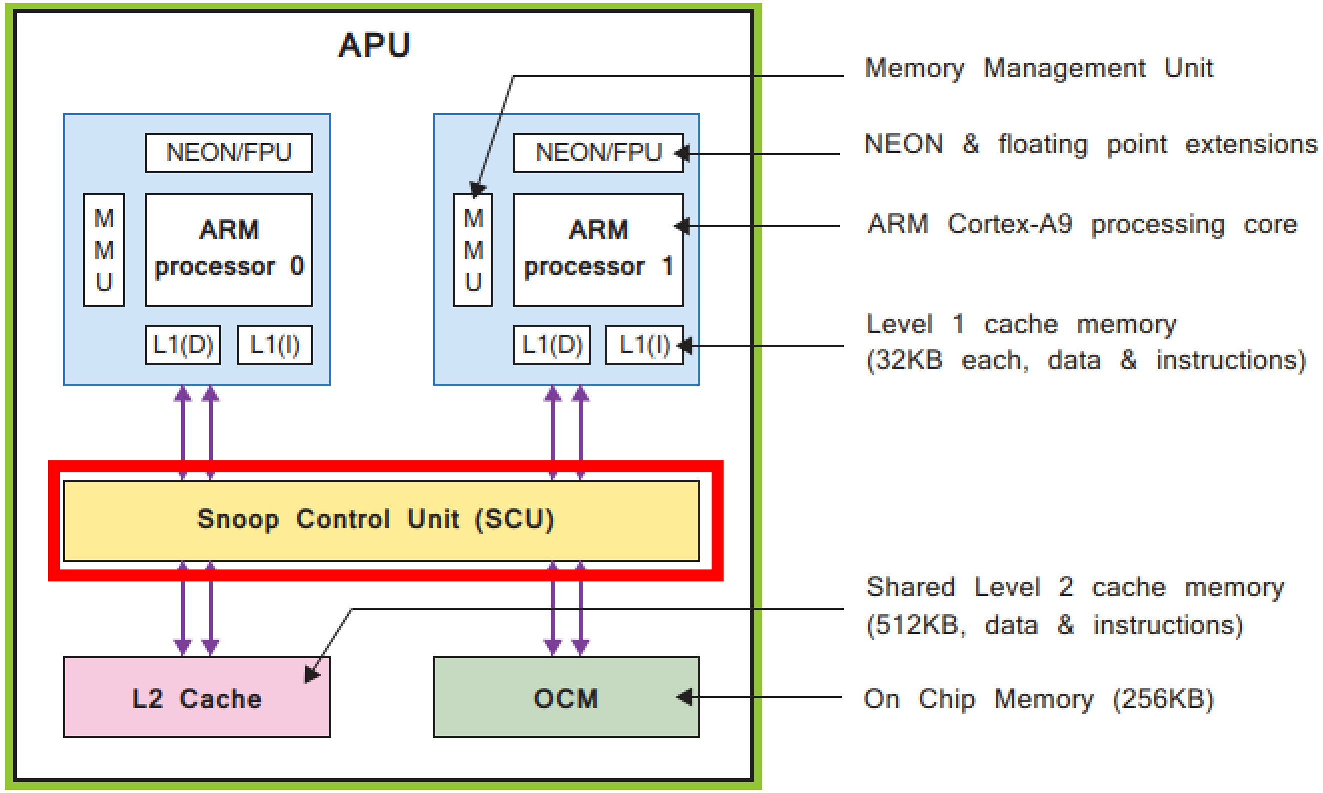
PS-SIMD *NEON* MPE



- NEON engine is a computational unit additional to the ARM that make “n” parallel operation “OP” (same) for two registers A and B (“b” bits vector) saved as “n” “b” bits output vector. This is known as Single Instruction Multiple Data (**SIMD**) data processing.
- Very useful for video processing or implementing parallel calculations such as Fast Fourier Transform (FFT) or Finite Impulse Response Filters (FIR).
- It possess an **own set of instructions**. Additional to the ARM instructions. These can be also compiled by C compiler. See here
- Supports int, unsigned int, single and half precision floating point. Double precision floating point is not supported.
- The FPU doe not support SIMD but can handle double precision floating point numbers. In compliance to IEEE-754.

SoC Application Processing Unit (APU)

Snoop Control Unit (SCU)



APU
NEON/FPU
MMU
ARM processor 0
L1(D)
L1(I)
NEON/FPU
MMU
ARM processor 1
L1(D)
L1(I)
Snoop Control Unit (SCU)
L2 Cache
OCM
Memory Management Unit
NEON & floating point extensions
ARM Cortex-A9 processing core
Level 1 cache memory (32KB each, data & instructions)
Shared Level 2 cache memory (512KB, data & instructions)
On Chip Memory (256KB)

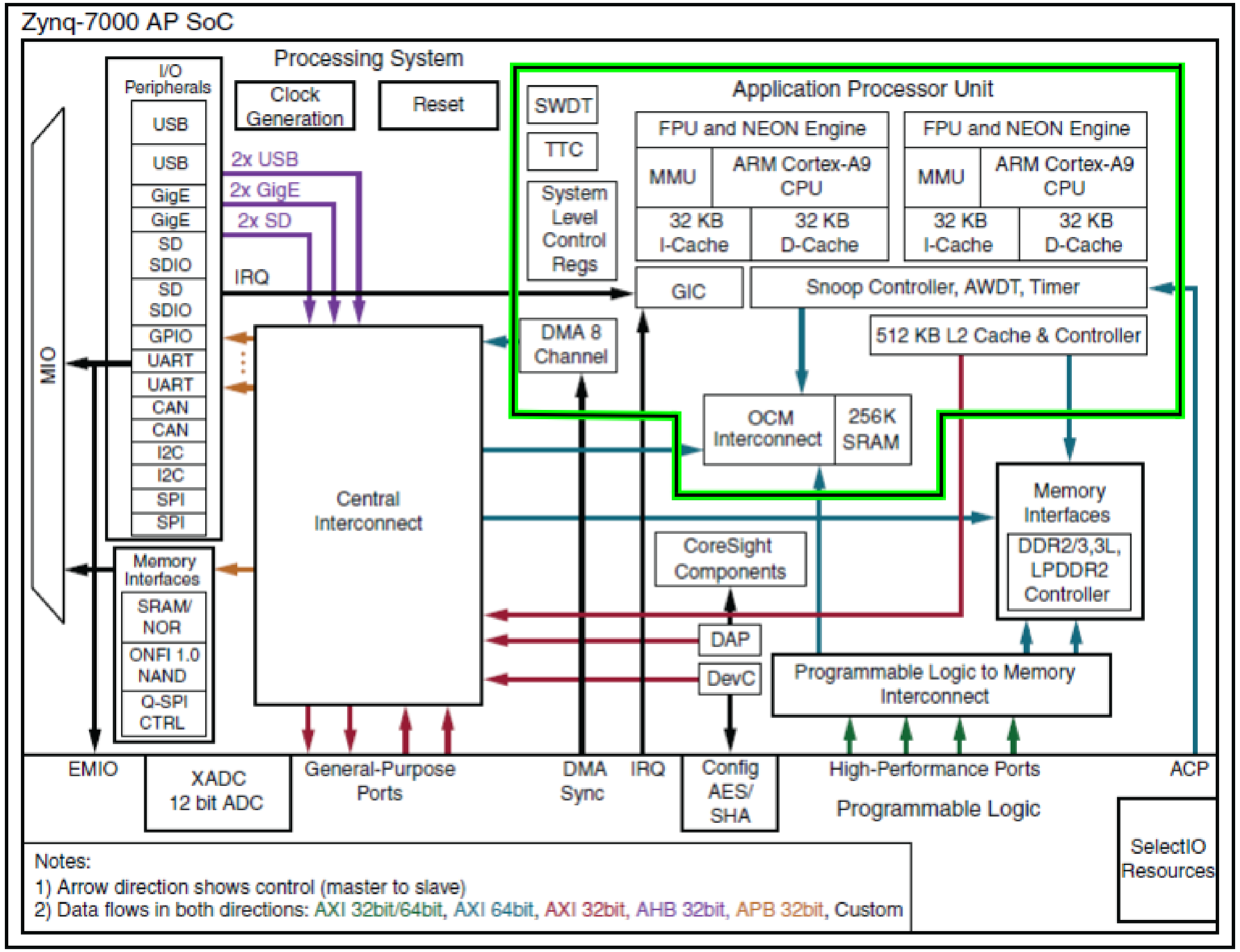
- SCU forms a bridge between ARM cores and L2 Cache and OCM (SRAM).
- SCU also serve as bridge with PL through ACP (see next slides)
- Its main task is managing consistency/coherency of data across shared cache resources.
- **Initiates and control access to L2, arbitrating request between two cores and PL.**

SoC Application Processing Unit (APU)

PS-External Interfaces

Visit here for detailed info about zynq 7000.

- SPI(x2): Serial Peripheral Interface
- I2C:
- CAN(x2): Controller Area Network
- UART(x2): Universal Asynchronous Receiver Transmmitter
- GPIO: General Purpose Input/Output
- SD(x2): For interfacing SD memory card
- USD(x2): Universal Serial Bus
- GigE(x2): Ethernet MAC peripheral 10Mbps, 100Mbps and 1Gbps
- MIO: Multiplexer Input/Output: 54 pins for flexible connection between PS and PL.
- Extended MIO: **EMIO** for more than 54 pins.



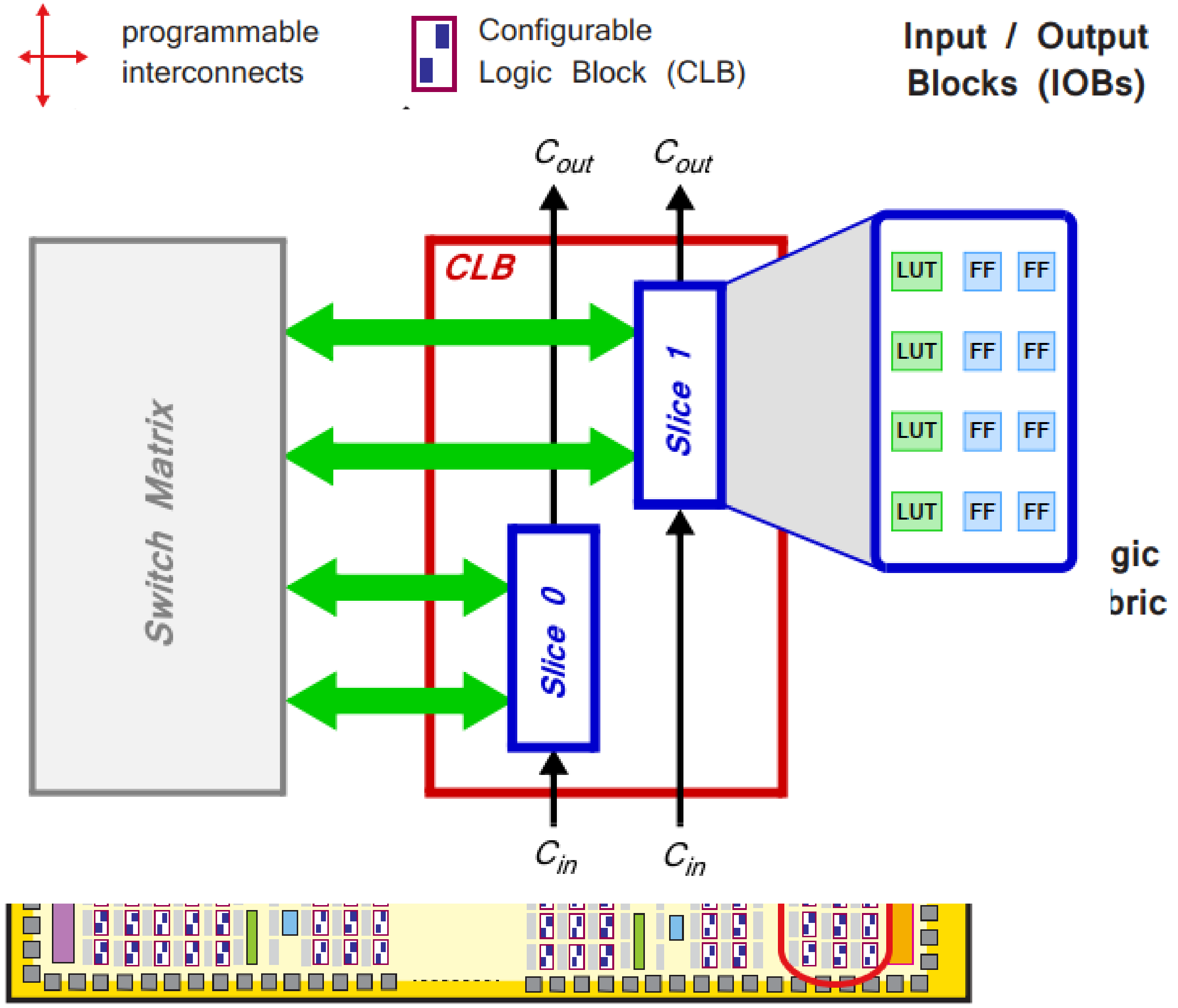
DS190_01_030713

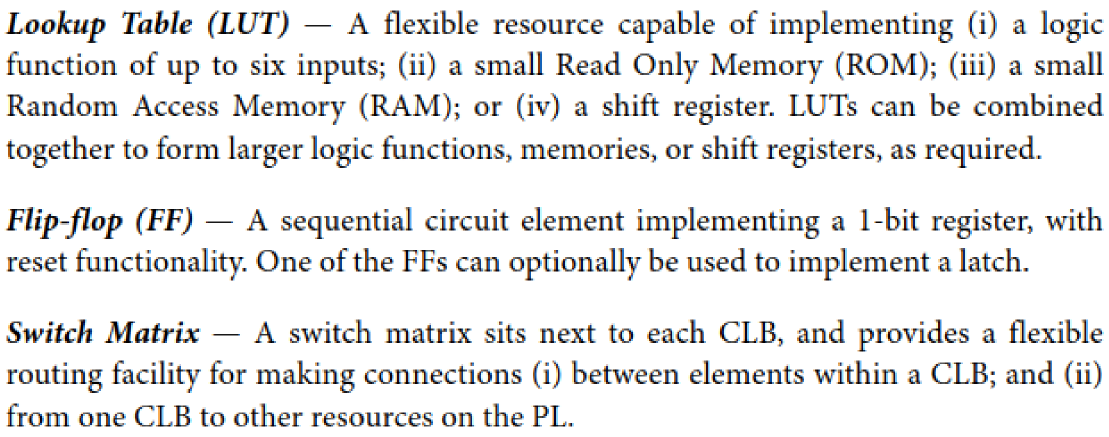
© Xilinx

System on Chip (SoC)

The PL Part

- Predominantly composed of general-purpose logic fabric which are composed of ***slices*** and ***CLBs.***
- *Input/Output Blocks:* provide a **1-bit interface** between PL logic resources and the physical device “pads” for external circuitry.
  - 50 IOBs generate a I/O *Bank.*
  - High Performance (HP) I/O Bank: high-speed interfaces for memory or other chips.
  - High Range (HR) I/O Bank: general interface for variety of IO standards.
- *Slice:* Sub-unit within a CLB contain **LUT and FLIPFLOPS.**




Lookup Table (LUT) — A flexible resource capable of implementing (i) a logic function of up to six inputs; (ii) a small Read Only Memory (ROM); (iii) a small Random Access Memory (RAM); or (iv) a shift register. LUTs can be combined together to form larger logic functions, memories, or shift registers, as required.

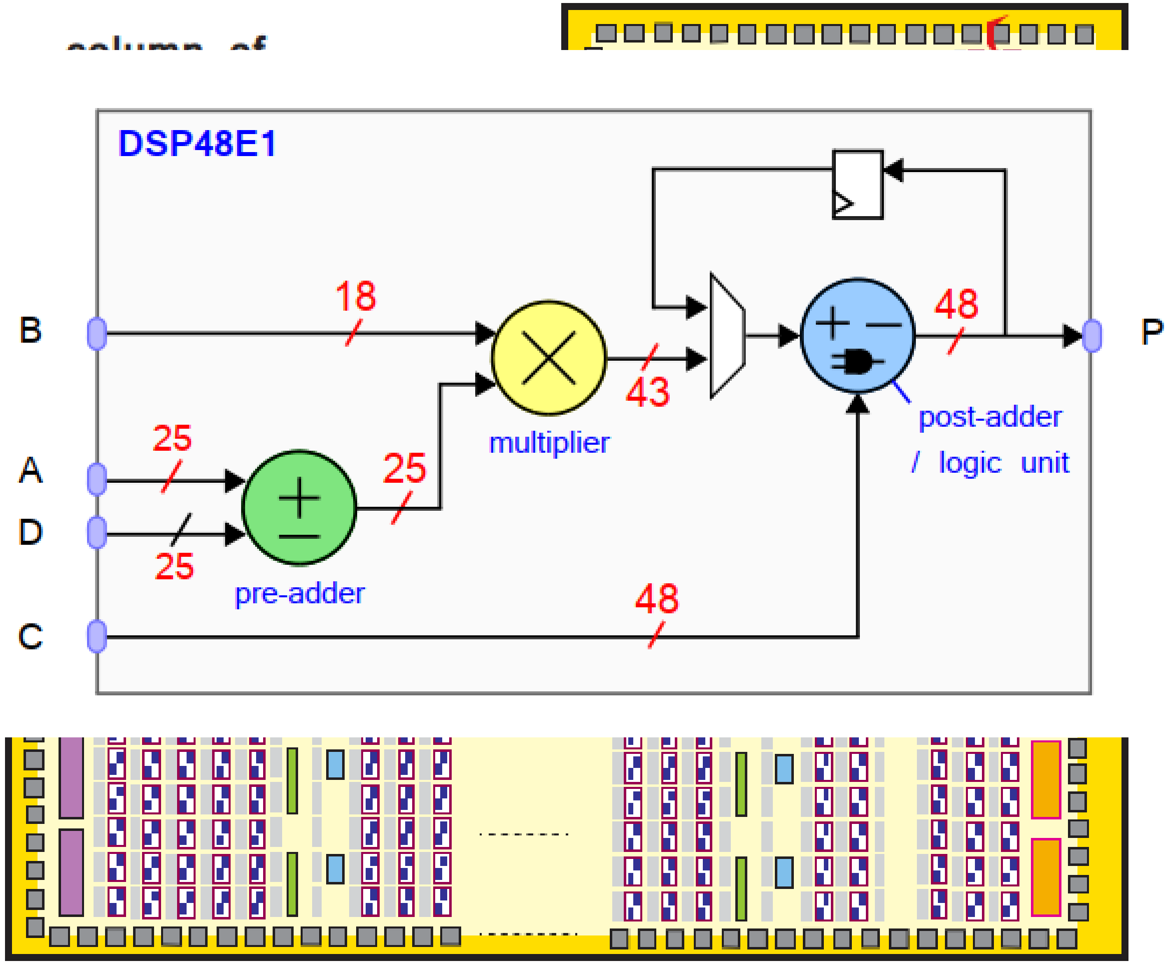
Flip-flop (FF) — A sequential circuit element implementing a 1-bit register, with reset functionality. One of the FFs can optionally be used to implement a latch.

Switch Matrix — A switch matrix sits next to each CLB, and provides a flexible routing facility for making connections (i) between elements within a CLB; and (ii) from one CLB to other resources on the PL.

System on Chip (SoC)

The PL Part-Special Resources

- **Block RAM**: This block can act as RAM (36Kb), ROM or FIFO Buffer. :
  - Block RAM is optimized to act as memory compared to use LUTs. It has less delays due to connections and uses less space.
  - Block RAM can be clocked at highest clock frequency supported by the device.
- **DSP48E1 Slices:** Specially suited for high-speed arithmetic operations with medium to long wordlenghts.
  - Dedicate silicon resources.
  - Composed of pre-adder/subtractor, multiplier and post-adder/substractor with logic unit.
  - Example of execution: P=(A+D)*B-C.
  - Capable of SIMD processing.
  - Post-adder can be used in "logic mode" to perform bit-wise NOT,AND,OR, NAND, NOR,XOR and XNOR.
  - Can be combined for complex operation.

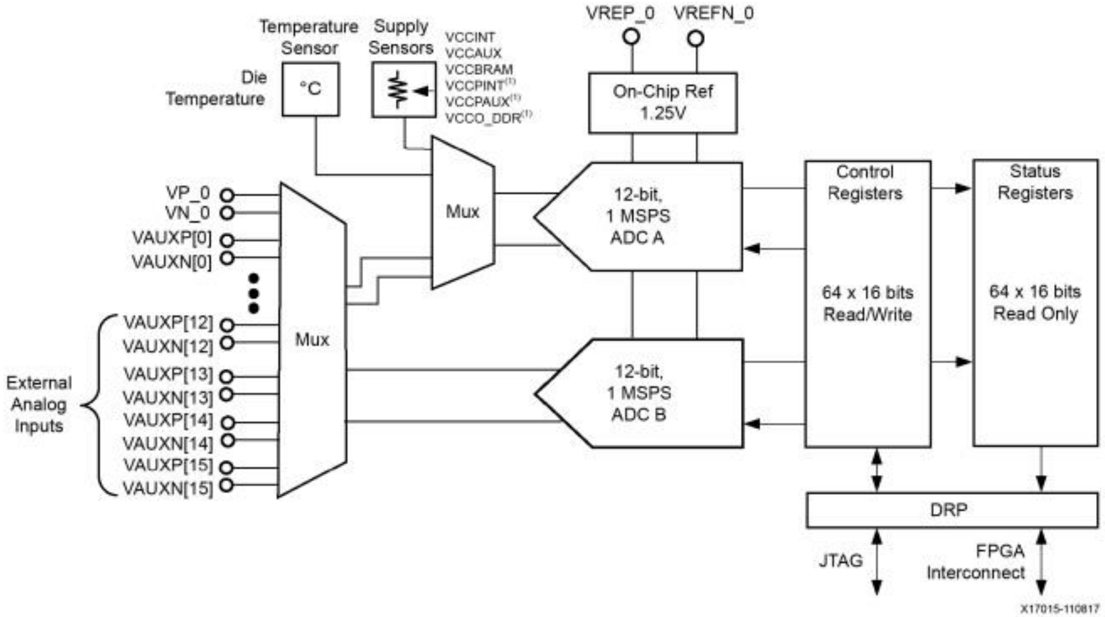


SIMD (del inglés ***S**ingle **I**nstruction, **M**ultiple **D**ata*, en español: "una instrucción, múltiples datos") es una técnica empleada para conseguir paralelismo a nivel de datos.

System on Chip (SoC)

The PL Part-Special Resources XADC

- Analog to Digital Conversion: PL has a hard IP XADC block.
  - 2x 12bit ADCs
  - Sampling at 1Msps
  - Up to 17 external analog channels.
  - XADC controlled via PS-XADC in APU.



In addition:

- **Clocks:** PL receives 4 separate clock inputs from PS. Also, it can generate its own clock.

- **Programming and Debug:** A set of JTAG ports are available. This is used for programming and debugging.

System on Chip (SoC)

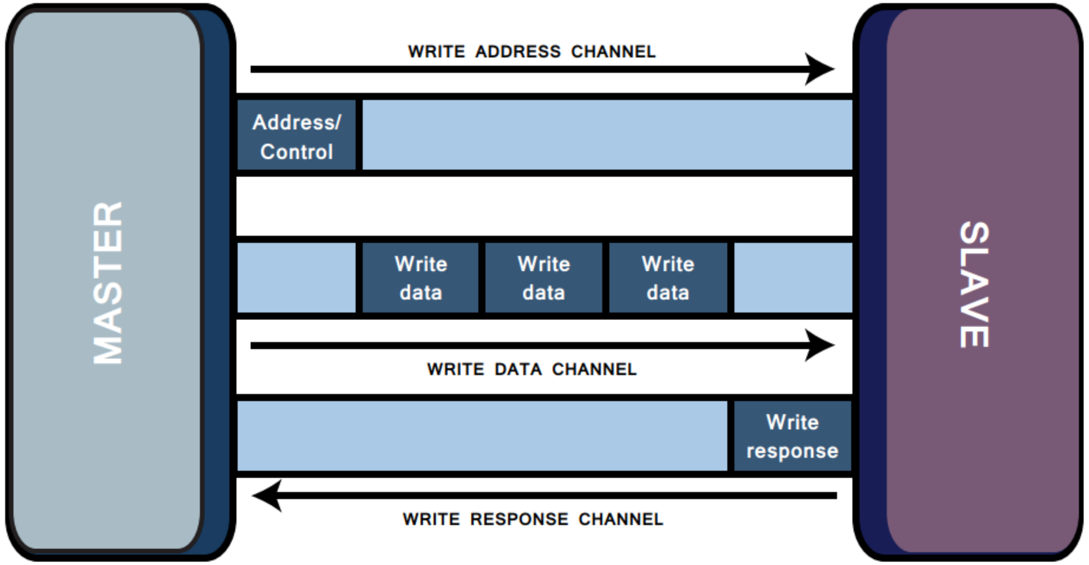
PS-PL Interfaces – AXI

- A key to explode the best of a SoC is the **communication between PL and PS**. **AXI** or (EMIO) can be used.

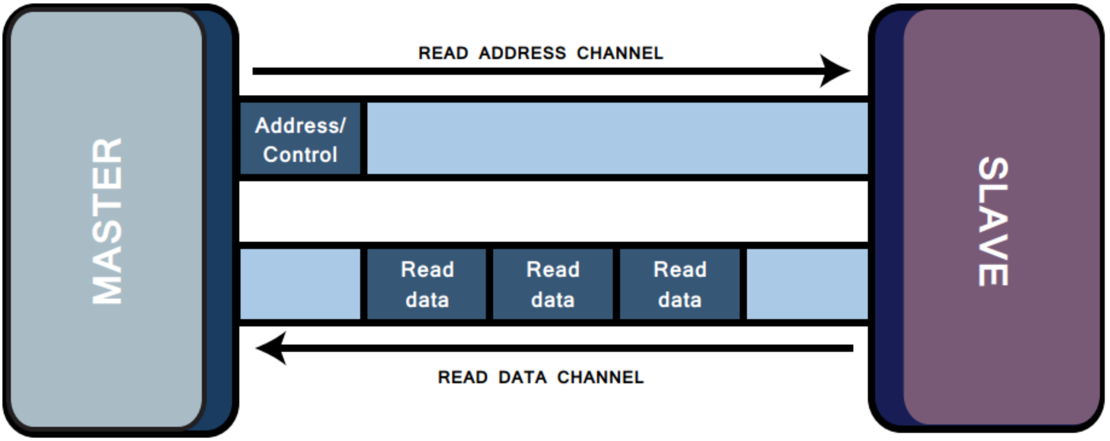
- **Advanced eXtensible Interface (AXI)** is a specialized bridge between PL and PS.
  - Current version is AXI4. Developed since 1996.
  - AXI4 is part of the ARM AMBA 3.0 standard.
  - Many IP block produced by third parties are based on this standard.
  - Known as “the facto standard for on-chip communication”
  - Exist 3 different AXI4 protocols, for different requirements:
    - **AXI4:** Intended for memory-mapped links, providing highest performance. Send one address followed by a burst of 256 words.
    - **AXI4-Lite:** Simplified version. Also memory-mapped but send one data per connection (no burst).
    - **AXI4-Stream:** Intended for high-speed streaming data. Burst data of unrestricted size. No memory-mapped based and no address mechanism. The data is sent directly between source and destination.
  - Memory-mapped meaning: A memory mapped protocol means that an address of a memory space is needed. The single data (AXI4-Lite) is written in that memory address. In multiple data transfer (AXI4) first data is saved on this address and the following addresses are calculated by the slave.

System on Chip (SoC)

AXI Communication

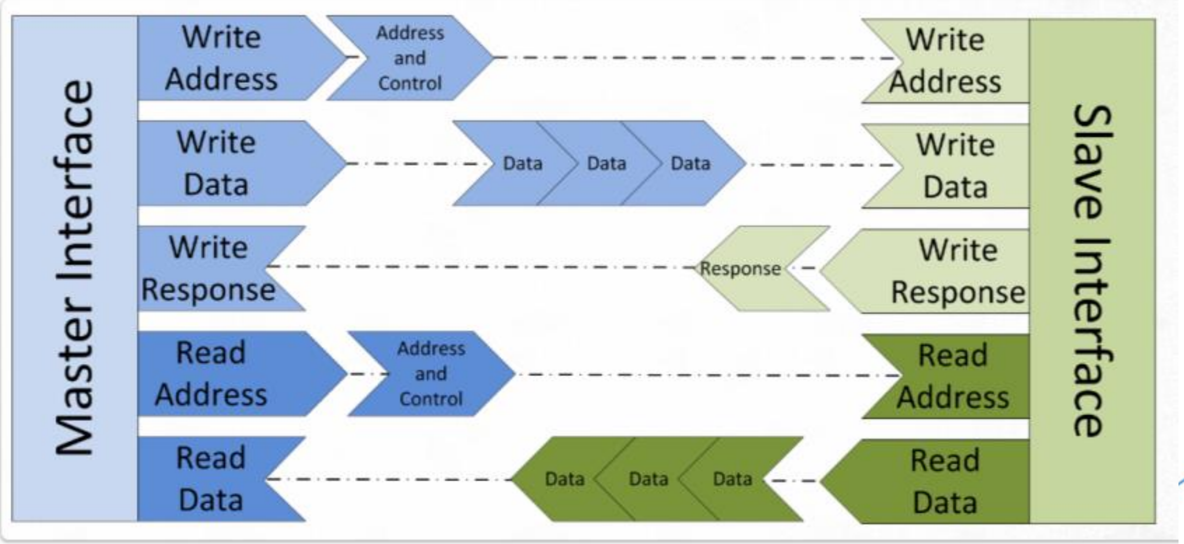


AXI4 write channel architecture



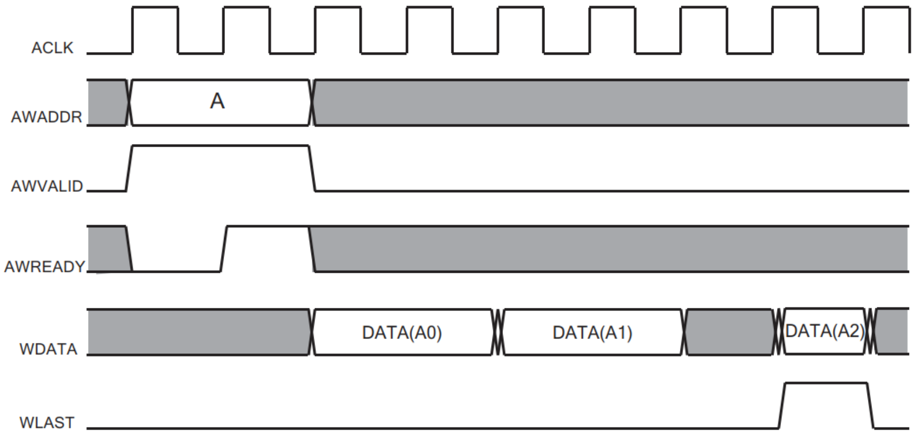
AXI4 read channel architecture

- One Channel is for **Adress/Control**
- One channel y for **Write** or **Read**
- AXI Master reads or write data from/to slave through independent buses.
- Write transaction has a response channel for completion of a write transaction.
- Through **Control** AXI can mainly configure:
  - Number of data (1 to 16) per burst.
  - Burst data size 8-1024bits
  - Wrapping, incrementing and non-incrementing burst.



System on Chip (SoC)

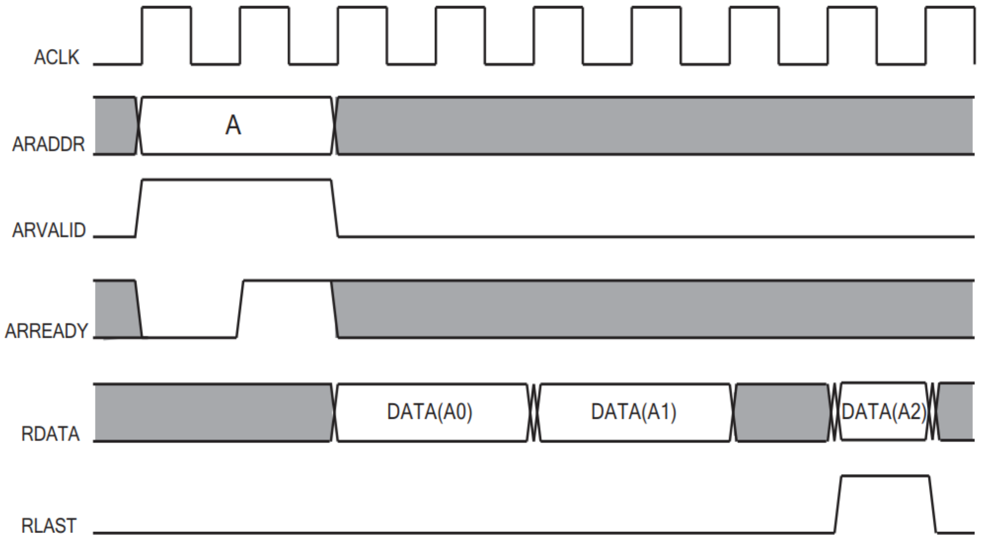
AXI Communication-Example



AXI4 Write burst transaction

Simplified Write Burst Process:

1. Transactions begins sending address and control signals via signal **AWADDR**. (AW:Address write)
2. The **AWVALID** signal confirm that it is a valid address.
3. Then **AWREADY** is high to indicate that the data write can begins.
4. Master sends blocks of data (Data(A0)-Data(A2)) through **WDATA** signals.
5. **WLAST** is high when last data is sent.



AXI4 Read burst transaction

Simplified Read Burst Process:

1. Transaction begins when master send address and control information through **ARADDR.** (AR: Address read)
2. The **ARVALID** signal is high if the address is valid.
3. One address is valid **ARREADY** is high, and master is ready to read.
4. The **RDATA** signals carries the data blocks being read. (DATA(A0)-DATA(A2))
5. The **RLAST** signal indicates that the final data block was read.

System on Chip (SoC)

PS-PL Interfaces-AXI

AXI-Interconnect: A switch that manages the data traffic between AXI interfaces. The PS has several AXI-interconnects, some to interface PL and some for internal AXI-interface. Through this, several masters and slaves can be connected.

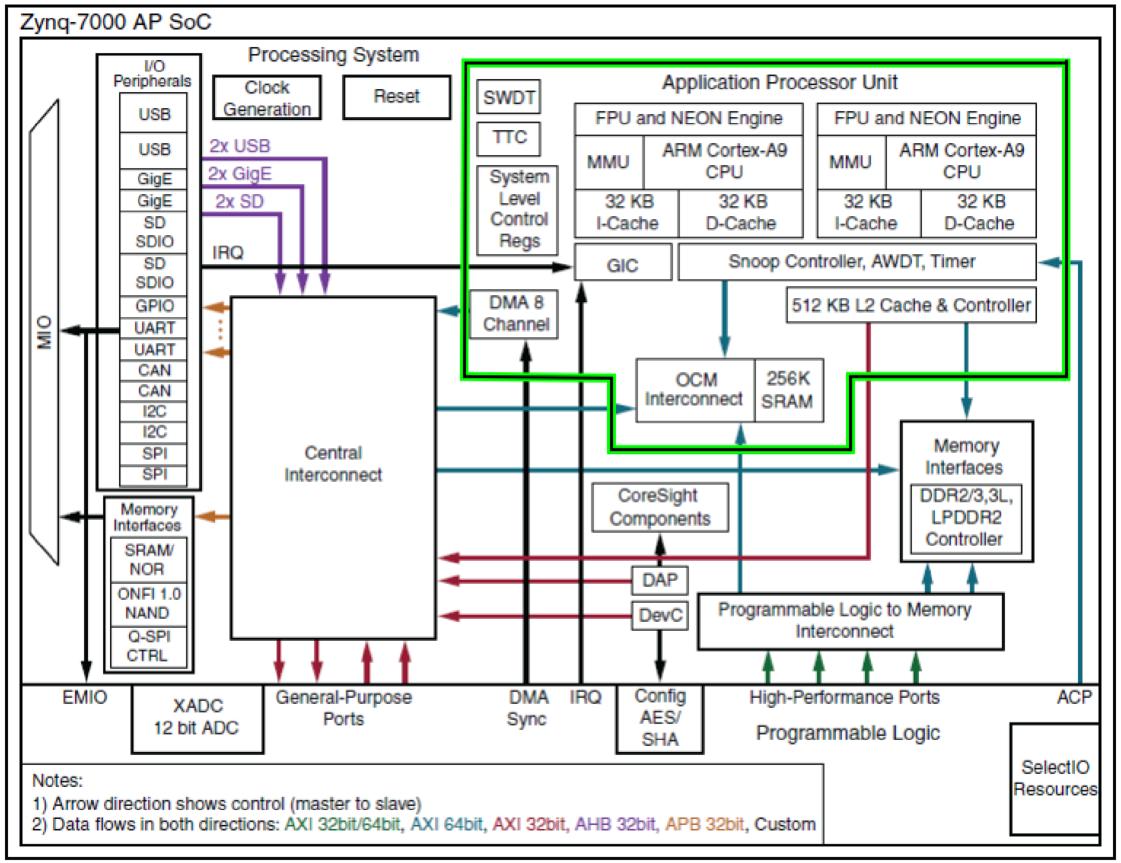
AXI-Interface: Connection lines for transferring data, address and hand shaking signals between **Masters and Slaves.**

ACP is an exception that directly goes to SCU, avoiding AXI-Interconnect. Direct Master-slave interconnection.

Central Interconnect contains all AXI-Interconnects PS peripheral and connection to PL. Details here.

APU, ARM and PS posses several AXI-interface within themselves. (see color lines in the picture of the right)

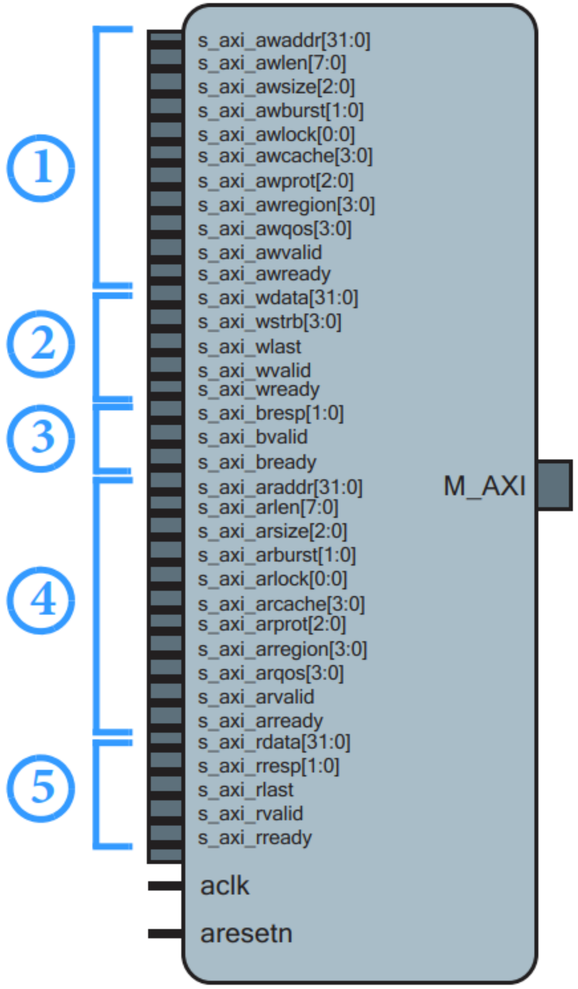
Out-of-order Transaction: AXI protocol through **Interconnect** gives a tag to each Transaction. Transactions can be executed in the order they were issued, or out-of-order. E.g. slaves with faster response time are prioritized. Useful to improve performance.



DS190_01_030713
© Xilinx

System on Chip (SoC)

AXI Communication-Example

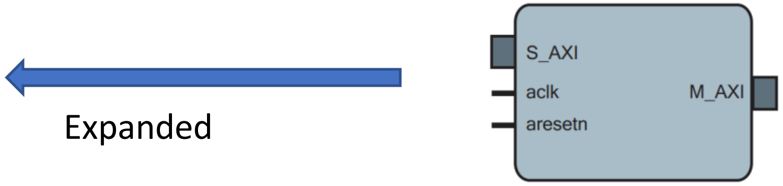


Vivado provides a simple graphical interface to integrate AXI4-

The example shows a FIFO block that features slave AXI BUS (S_AXI) and master M_AXI.

Expanding the block, all signals are shown, divided into 4 groups:

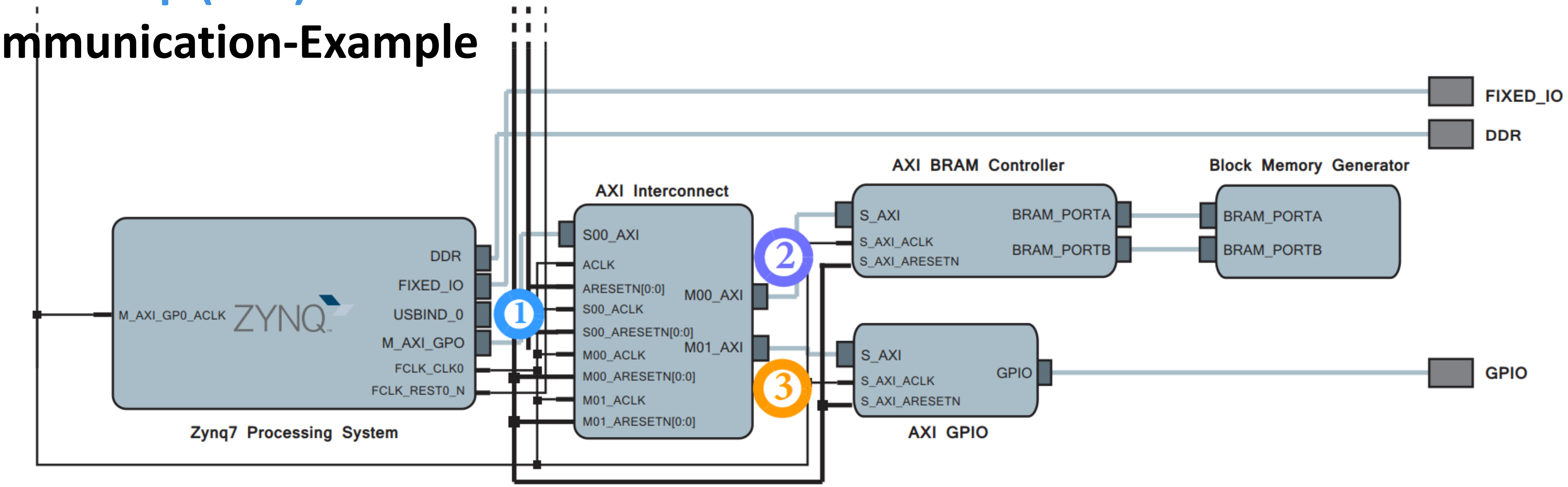
1. **Write Adress Channel**
2. **Write Data Channel**
3. **Read Adress Channel**
4. **Read Data Channel**



AXI4 data FIFO Vivado IP Core

System on Chip (SoC)

AXI Communication-Example



- This example interconnect two slave devices: AXI GPIO (pins GPIO) and AXI BRAM controller (Access to RAM).
- The Processing system acts as master for the *AXI Interconnect* block. Its M_AXI_GPO Is connected to S00_AXI. It also provides the clock to the interconnect master and slave ports through *S00_ACLK, M00_ACLK, M01_ACLK.*
- Same clock (S00_ACLK) is used for the two AXI devices (S_AXI_ACLK).
- All reset signals come from a processor rest block (not shown).
- AXI interconnect has two outputs master channels. (its signals were showed before).

System on Chip (SoC)

PS-PL Interfaces-AXI

General Purpose AXI (x4):

- 32bits data bus.
- Does not include buffering.
- PS is master of two of them, and PL of other two.

Accelerator Coherency Port(x1):

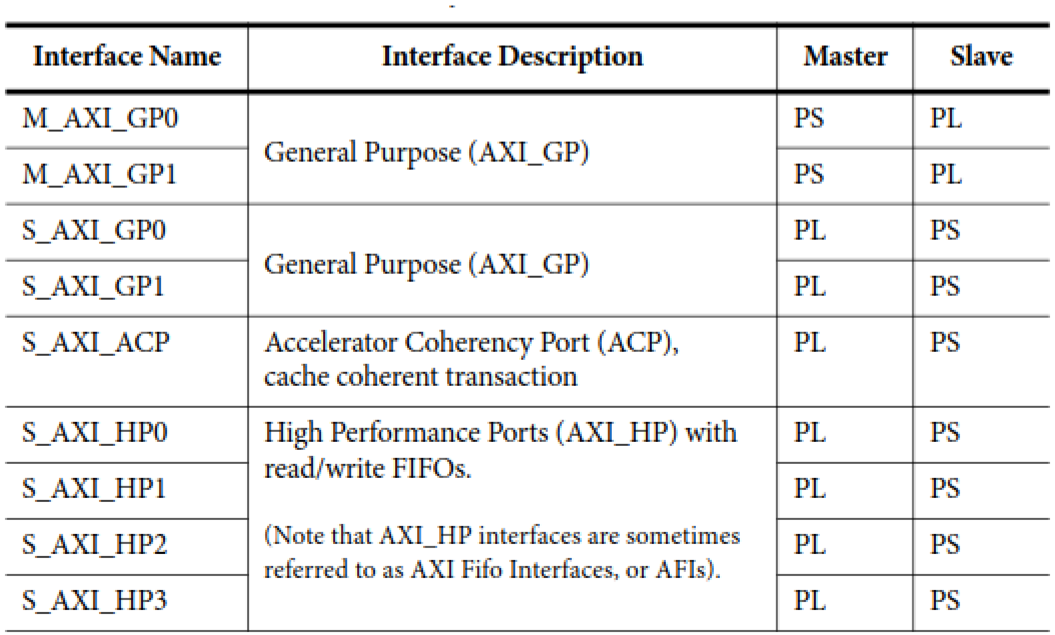
- 64bits bus.
- Direct connection between PL and SCU.
- PL is master.
- Port used for achieving coherency between APU caches and elements in PL.

High Performance Port(x4):

Uses FIFO buffers for “bursty” read and write.

Data width is 32bits or 64bits.

Support high-rate communication between PL and Memories of PS.



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

This table shows the PS AXI interfaces of the previously shown AXI-Interconnects. “M” stands for PS acting as Master. “S” for slave.

System on Chip (SoC)

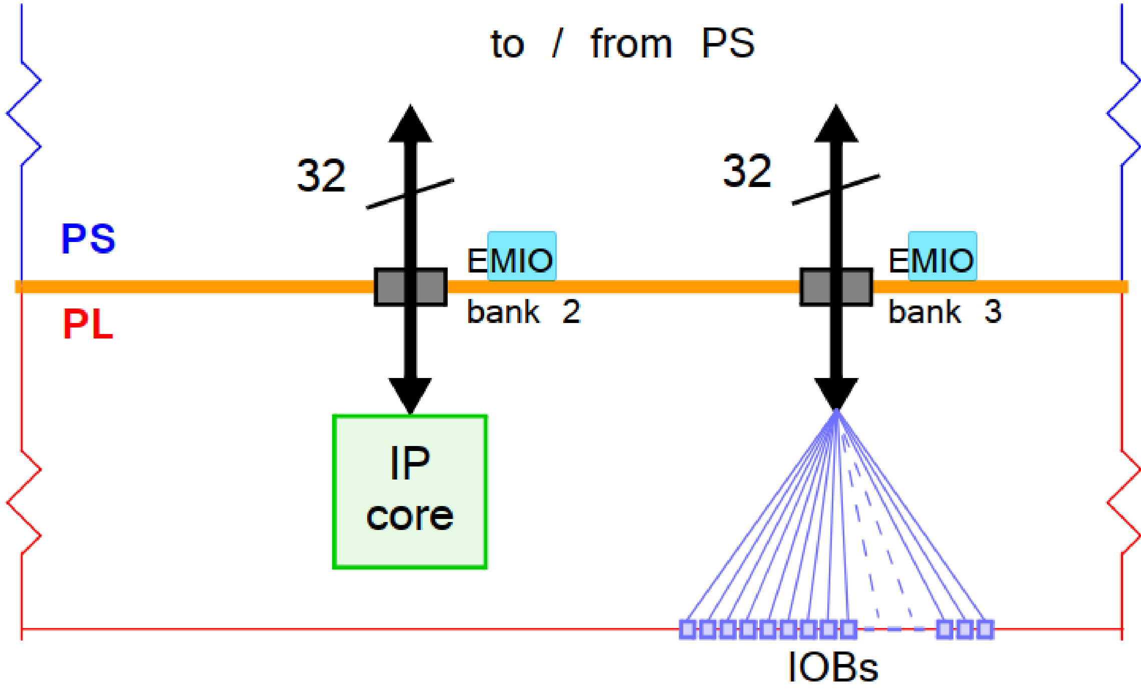
PS-PL Interfaces- EMIO

Extended Multiplexer Input/Output (EMIO)

- Enables signal transfer between PS with PL.
- Connection are arranged in two banks of 32bits.
- It is achieved by simple set of wires.
- Used to route some PS peripheral (UART,I2C,etc) to an IP-core the PL or to external pins of PL.

Other Signals that cross PS-PL boundaries:

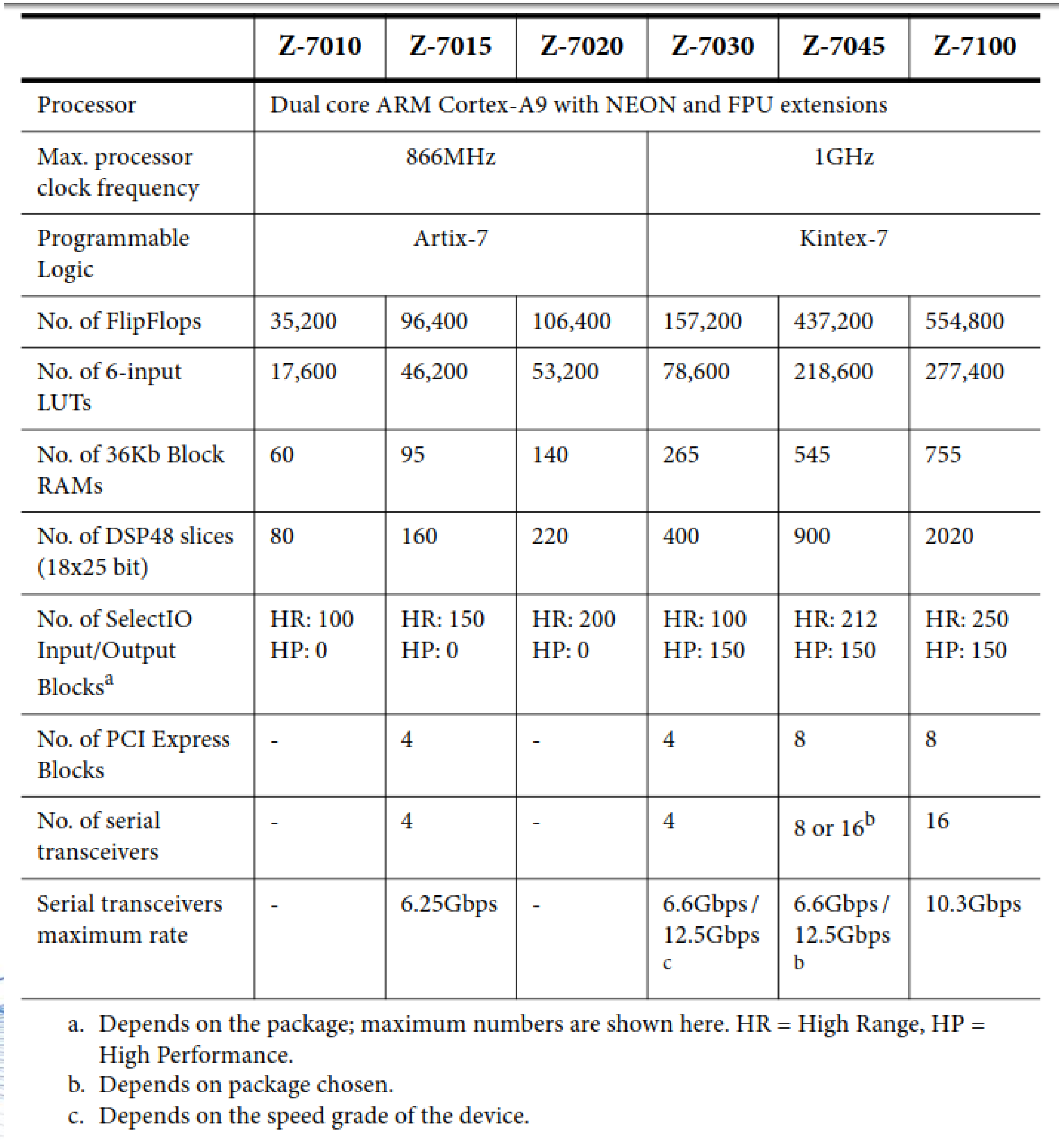
- Watchdog timers
- Reset signals
- DMA interfacing signals. (we will check this later).



What is a multiplexer?

System on Chip (SoC)

Now we can understand the Zynq Family Specifications!



	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[a]	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[b]	16
Serial transceivers maximum rate	-	6.25Gbps	-	6.6Gbps / 12.5Gbps c	6.6Gbps / 12.5Gbps b	10.3Gbps

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

b. Depends on package chosen.

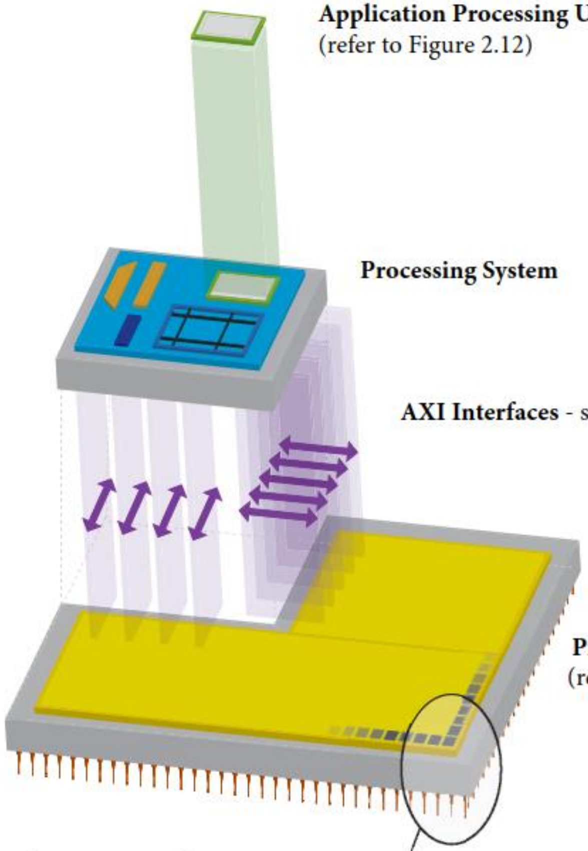
c. Depends on the speed grade of the device.

System on Chip (SoC)

Overview Summary

Zynq Architecture and Features

see Zynq-7000 Technical Reference Manual, [33]



Application Processing Unit (APU) (refer to Figure 2.12)

Processing System

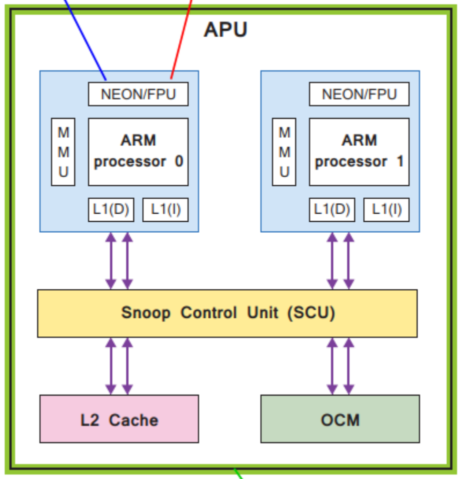
AXI Interfaces - see [25]

Programmable Logic (refer to Figure 2.13)

Package, pins, and IOBs see [24]and [31]

EON Processing Unit - see [9] & [15]

Floating Point Unit - see [7]



ARM Cortex-A9 MPCore- see [8]

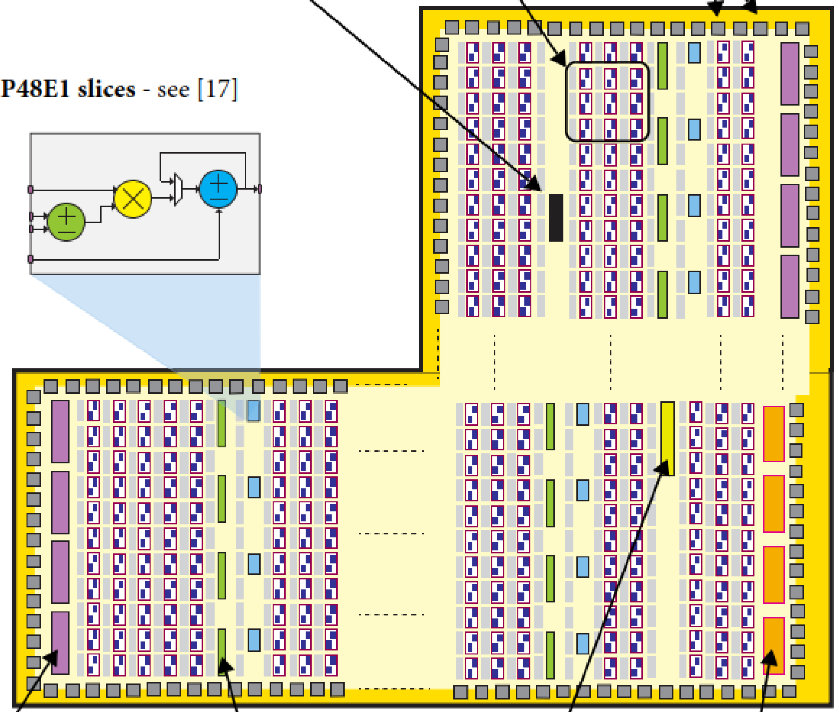
ARM architecture - see [5]

Configurable Logic Blocks see [20]

Input/Output Blocks see [24]

XADC block - see [18]

DSP48E1 slices - see [17]



clock tiles - see [19]

Block RAM - see [23]

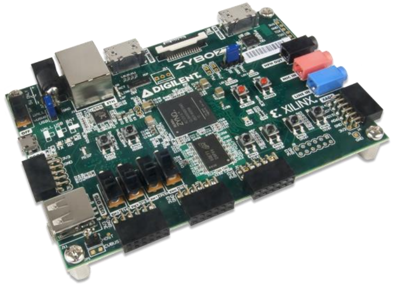
GTX transceivers see [21]

PCI Express block see [22]

More info see here

End of Lecture 01 Architecture of Programmable Systems

Next Lecture – C Programming



peclab

POWER & ENERGY CONVERSION LABORATORY

Electrical Engineering Department

Pontificia Universidad Católica de Chile

peclab.ing.uc.cl