ZYNQ Architecture

November 15, 2020

Learning Objectives

1. Evolution of ZNYQ Architecture
   - 1.1 Accelerators
   - 1.2 FPGA vs CPU
   - 1.3 Microblaze
   - 1.4 ZYNQ
2. ZYNQ Architecture
   - 2.1 Block Diagram of ZYNQ Architecture
   - 2.2 Different Modules in ZYNQ Architecture
   - 2.3 ZYNQ AXI Interfaces
3. Operating Systems
   - 3.1 Bare Metal
   - 3.2 Petalinux

1.Evolution of ZYNQ Architecture

ZYNQ is advanced SoC architecture it evolves by solving the problems in previous architectures FPGA without CPU and Microblaze(CPU on FPGA).

1.1 Accelerators

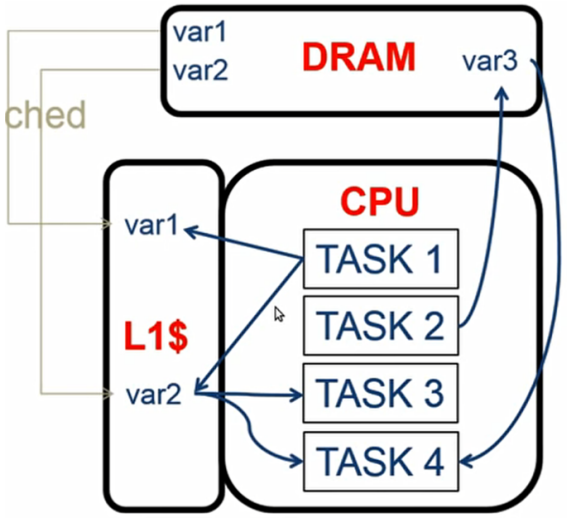

Figure 1: Traditional CPU

Traditional CPU contains DRAM,Cache,Processing Unit.This could performs low computation tasks quickly but for high computations it takes more time to complete.And main function of CPU is control the parts in the system this would effect if it performs high computation tasks.

1.1 Accelerators

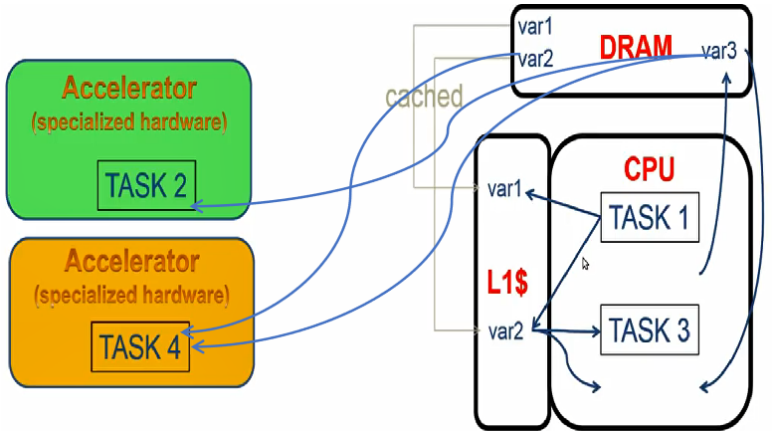

Figure 2: CPU with Accelerators

Accelerators are extra components in CPU which are faster,more power efficient and gives better performance per watt.So CPU gives high computation tasks to accelerators they perform those then return results to CPU.

1.1 Accelerators

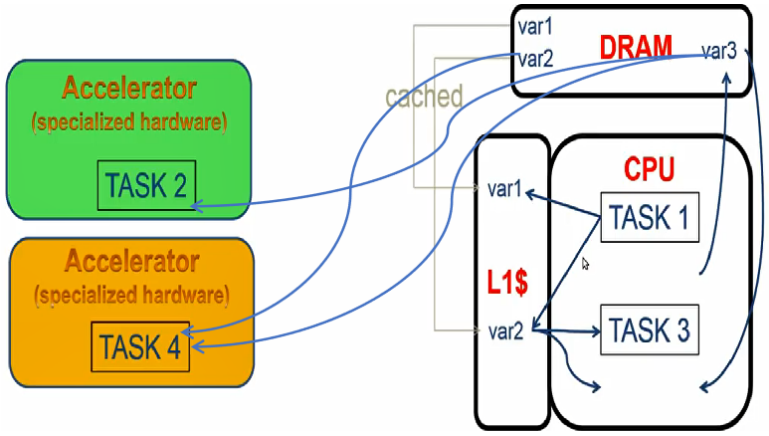

Figure 2: CPU with Accelerators

But promblem here is getting variables to perform tasks.Variables could get in two ways:

1)from DRAM:In DRAM updated variables are not there.So variables should flused to it from CPU before doing tasks in DRAM this may reduce performance CPU.

2)from Cache: Directly cannot get variables from cache. Accelerator Coherency Port(ACP) allows to perform coherent access to CPU memory.

1.2 FPGA vs CPU

Field Programmable Gate Array(FPGA):

FPGAs are capable of processing large size of inputs and return output.Massively parallel algorithms are acceptable.Capable of performing high I/O applications.But not capable of performing huge control flow condtions.

Central Processing Unit(CPU):

CPUs are highly capable of performing large control flow contditions.But CPU is not flexible as much as FPGA.

So there is trade off between flexibility and Control flow execution.To get both designed a architecture CPU on FPGA

1.3 MicroBlaze

Figure 3: MicroBlaze

MicroBlaze is microprocessor core designed by Xillinx.It has CPU on FPGA.CPU is designed on FPGA so it occupies area on FPGA which could be useful to other tasks.And also performance of this CPU is as much good as individual CPU.

1.4 CPU IP core on FPGA

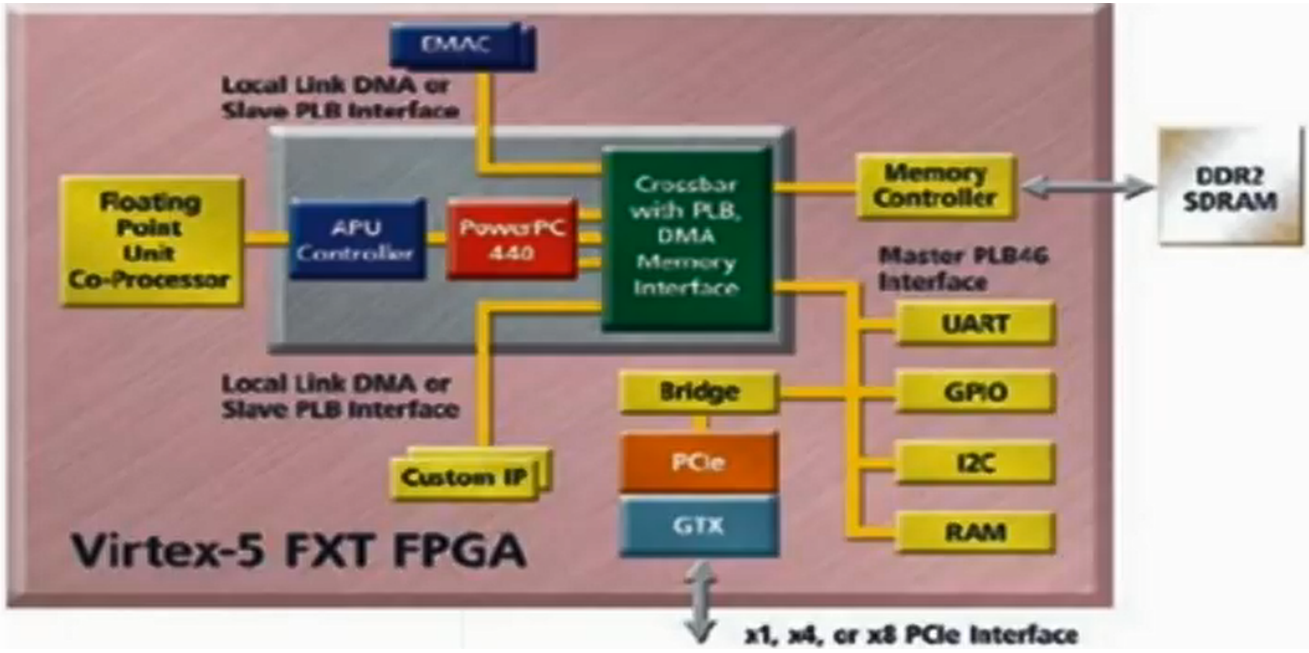

Figure 4: Virtex-5

ZYNQ Architecture introduced CPU IP core on FPGA.This has high performance and low power consumption.Let us discuss this architecture elaborately.

2.ZYNQ Architecture

The Zynq-7000 SoC has the two major functional blocks:

1.Processing System

(a) Application Processing Unit

(b) Memory Interfaces

(c) I/O Pheripherals

(d) Interconnect

2.Programmable Logic

(a) Configurable Logic Blocks(CLBs)

(b) DSP Slices

(c) Analog to Digital Converter

(d) Clock Management Tiles

2.1 Block Diagram of ZYNQ Architecture

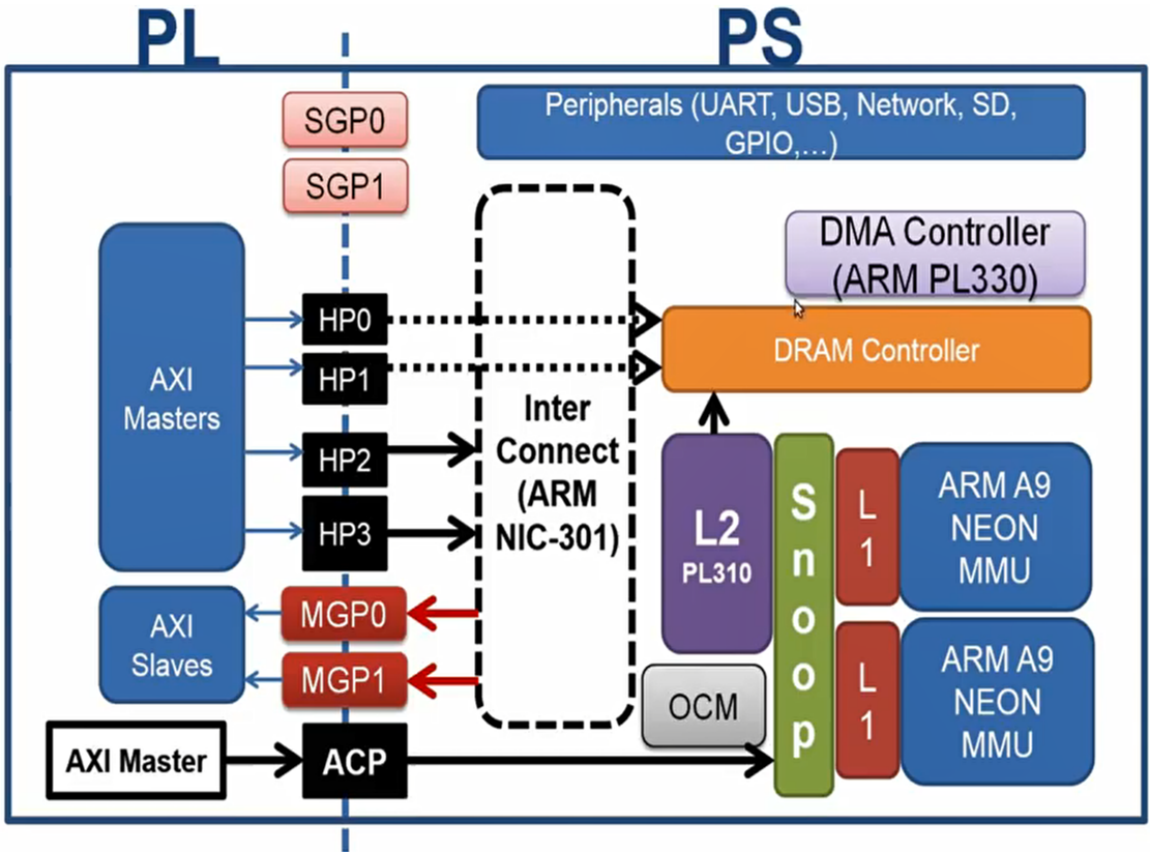

Figure 5: ZYNQ Architecture

Now let us discuss purpose of each module in the architecture.Then it is easy to understand architecture working.

2.2 Different Modules in ZYNQ Architecture

Modules in Processing System(PS):

1.Dual ARM Cortex-A9 MP Core CPUs

(i) These are ARM Cortex-A9 processors with NEON co-processors connected in an MP configuration sharing a 512 KB L2 cache.

(ii) Can execute 32-bit ARM instructions and 16-bit or 32-bit thumb instructions.

(iii)ARM architecture supports multiple operating modes including supervisor, system, and user modes to provide different levels of protection at the application level.

(iv) Accesses from this Processor can target the OCM, DDR, PL, I/O slaves,ACP or registers within the PS sub-blocks.So this could access and control any module in architecture that is called as heart of architecture.

2.L1 and L2 Caches

(i) There are two L1 caches each belongs each CPU core.These support 32-bit instruction and 32-bit data.Coherence data from Cortex-A9 cores could get from these caches through Snoop Control Unit.

(ii) The shareable cache L2 connected to DRAM,SCU,Central Interconnect share data from CPU and vice-versa.

3.Snoop Control Unit(SCU)

All accesses from the dual Cortex-A9 MP system go through the SCU and all accesses from any other master that requires coherency with the Cortex-A9 MP system also need to be routed through the SCU using the ACP Port.

4.DRAM Controller

The DRAM controller is also known as Double Data Rate(DDR) RAM Controller.It consists three major blocks: an AXI memory port interface (DDRI),a core controller with transaction scheduler(DDRC),and a a controller with digital PHY (DDRP).

(i) The DDRI block interfaces with four 64-bit synchronous AXI interfaces to serve multiple AXI masters simultaneously.

(ii) The DDRC contains two 32-entry content addressable memories (CAMs) to perform DDR data service scheduling to maximize DDR memory efficiency.

(iii) The PHY processes read/write requests from the controller and translates them into specific signals within the timing constraints of the target DDR memory.

2.2 Different Modules in ZYNQ Architecture

5.DMA Controller

Direct Memory Access(DMA) Controller perform data transfer to/system memories and PL pheripherals.

6.Pheripherals

(i) Universal Asynchronus Receiver Transmitter(UART)

(ii) General Purpose I/O (GPIO)

(iii) Universal Standard Bus(USB)controllers

(iv) SD card slot

(v) Gigabit Ethernet Controller

(vi) CAN Controller

(vii) I2C Controllers

7.Central Interconnect

The interconnect is the major one for data communication in the SoC of this architecture.It connects to

(i) L2 Cache Controller

(ii) Snoop Controller

(iii) Master and Slave interconnect to pheripherals

(iv) On chip Memory(OCM)

(v) Memory(DDR Controller)

(vi) AXI GP,general purpose ports

(vii) AXI HP,high performance slave port for AXI Master in PL

8.On Chip Memory(OCM)

The on-chip memory (OCM) module contains 256 KB of RAM and 128 KB of ROM (BootROM).

2.2 Different Modules in ZYNQ Architecture

Modules in Programmable Logic(PL):

1. Digital Signal Processors
2. Analog to Digital Converters(ADC)
3. Configuration Logic Blocks(CLB)
4. 32Kb Block RAM
5. Clock Managers
6. Low Power Gigabit Transceivers

Working of overall PL is similar to FPGA.

2.3 System Address

Address Range	CPUs and ACP	AXI_HP	Other Bus Masters[1]	Notes
0000_0000 to 0003_FFFF[2]	OCM	OCM	OCM	Address not filtered by SCU and OCM is mapped low
	DDR	OCM	OCM	Address filtered by SCU and OCM is mapped low
	DDR			Address filtered by SCU and OCM is not mapped low
				Address not filtered by SCU and OCM is not mapped low
0004_0000 to 0007_FFFF	DDR			Address filtered by SCU
				Address not filtered by SCU
0008_0000 to 000F_FFFF	DDR	DDR	DDR	Address filtered by SCU
		DDR	DDR	Address not filtered by SCU[3]
0010_0000 to 3FFF_FFFF	DDR	DDR	DDR	Accessible to all interconnect masters
4000_0000 to 7FFF_FFFF	PL		PL	General Purpose Port #0 to the PL, M_AXI_GP0
8000_0000 to BFFF_FFFF	PL		PL	General Purpose Port #1 to the PL, M_AXI_GP1
E000_0000 to E02F_FFFF	IOP		IOP	I/O Peripheral registers, see Table 4-6
E100_0000 to E5FF_FFFF	SMC		SMC	SMC Memories, see Table 4-5
F800_0000 to F800_0BFF	SLCR		SLCR	SLCR registers, see Table 4-3
F800_1000 to F880_FFFF	PS		PS	PS System registers, see Table 4-7
F890_0000 to F8F0_2FFF	CPU			CPU Private registers, see Table 4-4
FC00_0000 to FDFF_FFFF[4]	Quad-SPI		Quad-SPI	Quad-SPI linear address for linear mode
FFFC_0000 to FFFF_FFFF[2]	OCM	OCM	OCM	OCM is mapped high
				OCM is not mapped high

Figure 6: System Address

CPU in this architecture is memory mapped so every module has unique range of address to access.

2.3 ZYNQ AXI Interfaces

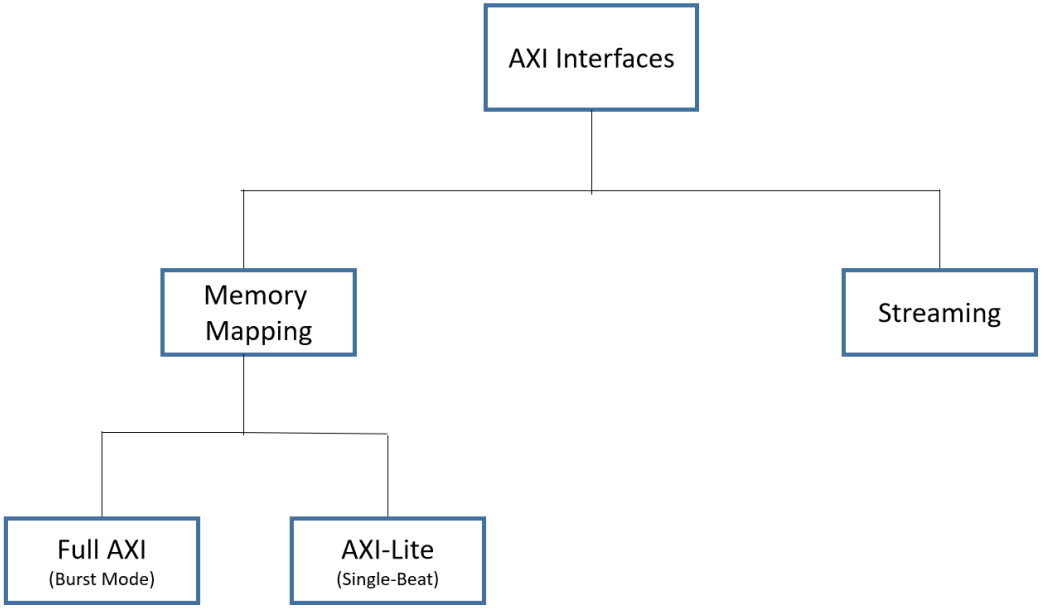

Figure 7: AXI Interfaces

Transaction: Transferring data from one point to another on chip.

AXI Master: Initiates the transaction for read or write.

AXI Slave: Respond to the transaction master read or write from slave.

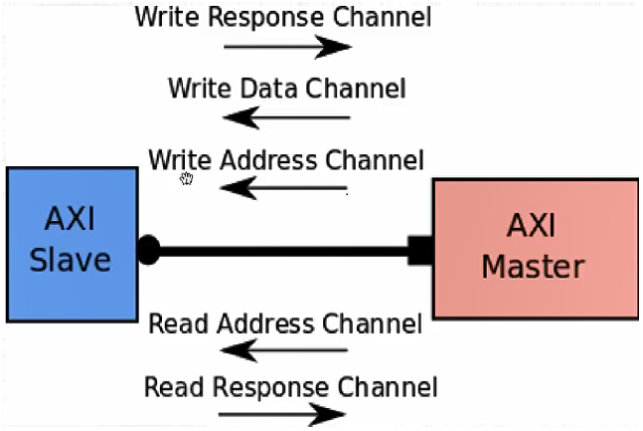

Figure 8: AXI Memory Map Interfacing

AXI Memory Map Interfacing:

AXI memory map interfacing have five types of channels.Write address,write data,write response channels are required to write data in slave by master.Read address,read response channels are required to read data from slave by master. This interfacing requires addresses.

2.3 ZYNQ AXI Interfaces

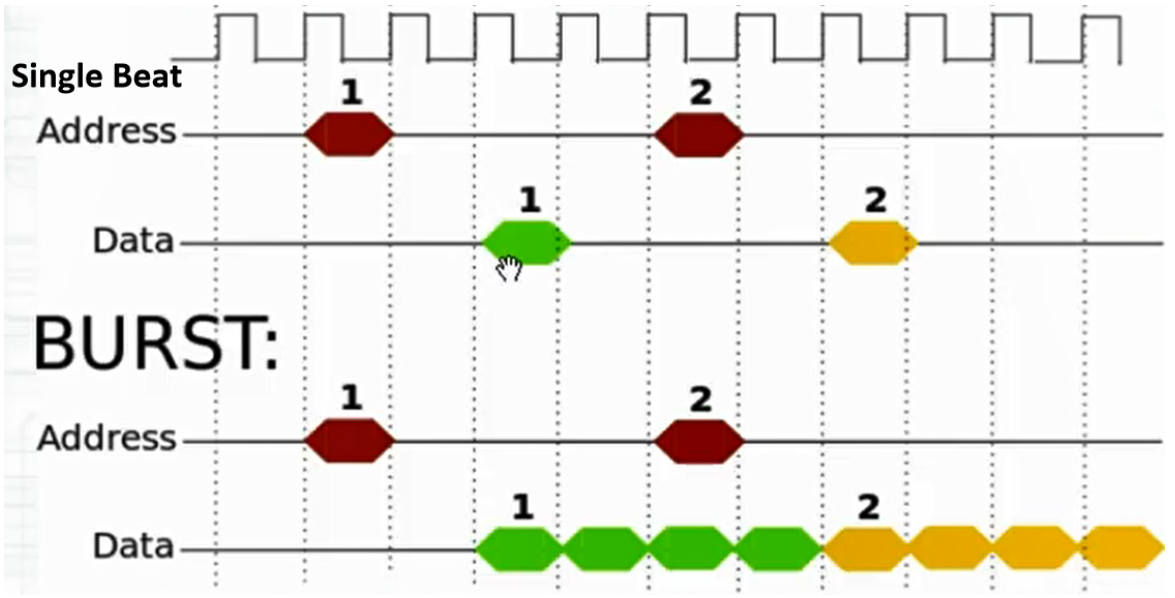

Figure 9: AXI Lite vs Full AXI

AXI Lite: Transactions capable of sending only one chunk of data for each transaction.

AXI Full: Transactions capable of sending multiple chunks of data for each transaction.

Figure 10: AXI Stream Interfacing

AXI Stream Interfacing:

In AXI stram interfacing flow of signals only in one direction since it doesn't have any address to send to data there is only one destination.So there is one channel which is only capable of writing data.

3.Operating Systems

Operating System makes user to control hardware with software.This could be done in some ways:
1.Bare Metal
2.Petalinux

Bare Metal Programming

(i) Bare-metal refers to a software system without an operating system. This software system typically does not need many features (such as networking) that are provided by an operating system.

(ii) Bare metal programming , the code directly runs in the hardware.Bare metal programming mostly involves two programming languages, C and Assembly.

(iii) The main advantage of coding embedded systems bare metal is its execution speed,simplicity, reduced power consumption.

Petalinux

(i) PetaLinux provides an easy way to develop user applications for Zynq and MicroBlaze Linux systems, including building, installing, and debugging.

(ii) Petalinux is advanced programming to control embedded systems.Although this takes more memory space than bare metal programming but it performs more user applications.