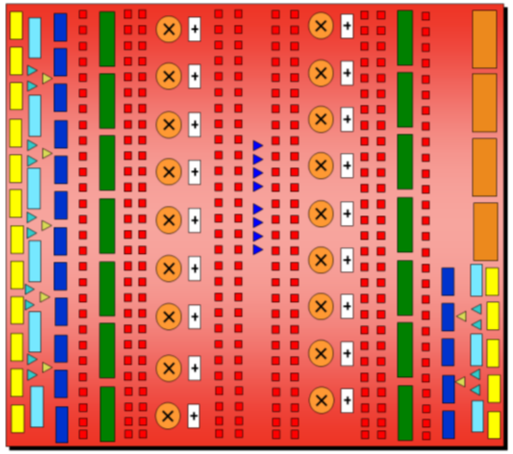
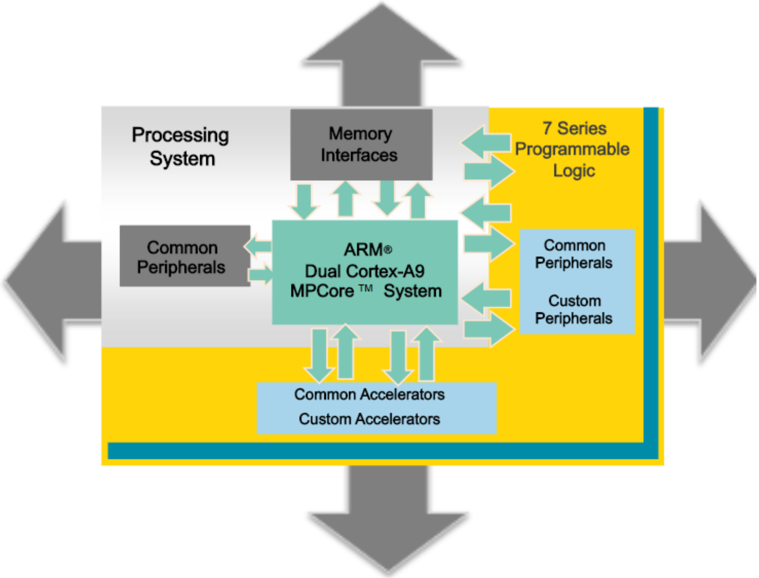
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

A2A

Algorithms to Architecture

ECE 270: Embedded Logic Design





Processing System

Memory Interfaces

7 Series Programmable Logic

Common Peripherals

ARM® Dual Cortex-A9 MPCore™ System

Common Peripherals

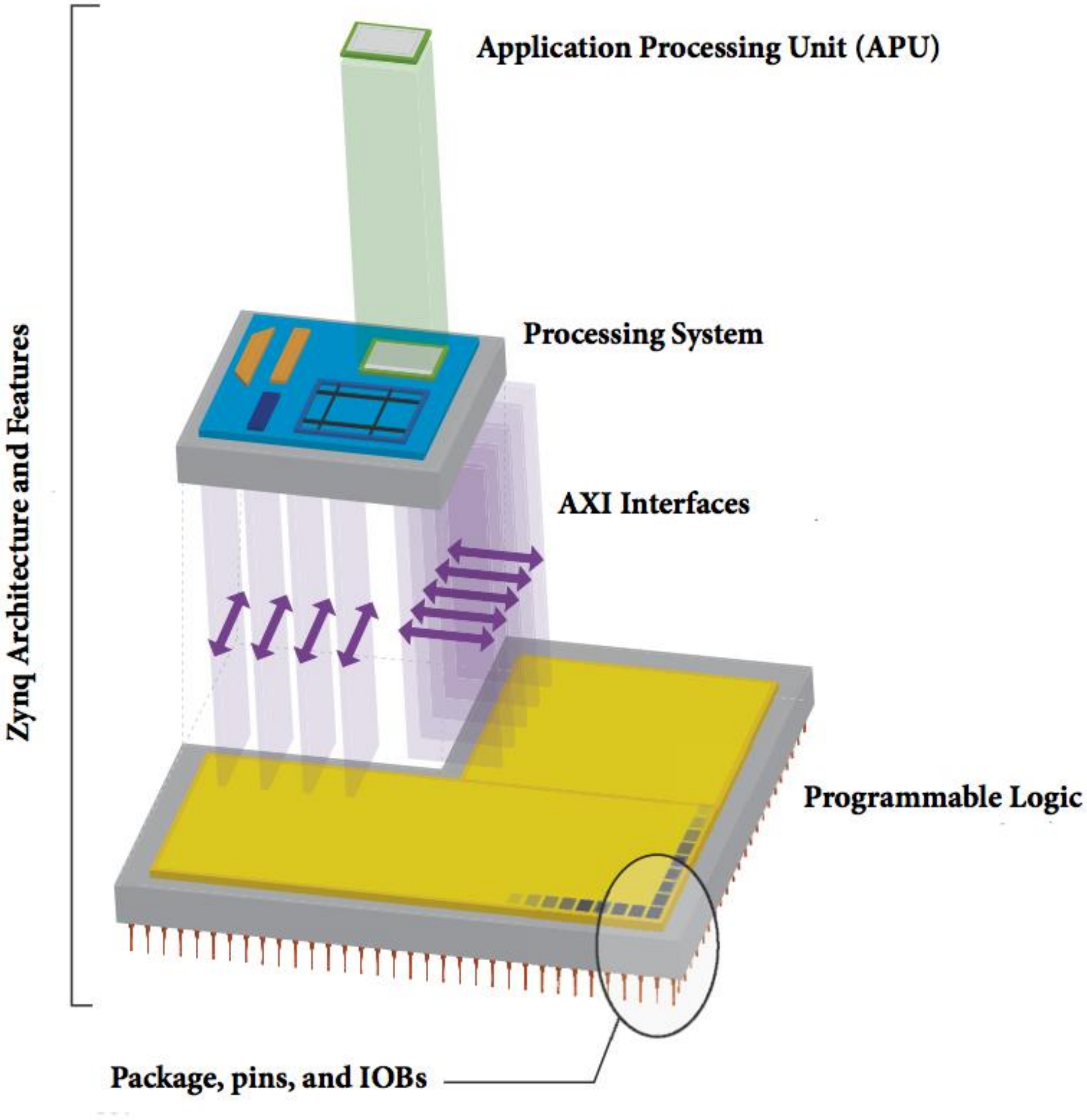
Custom Peripherals

Common Accelerators

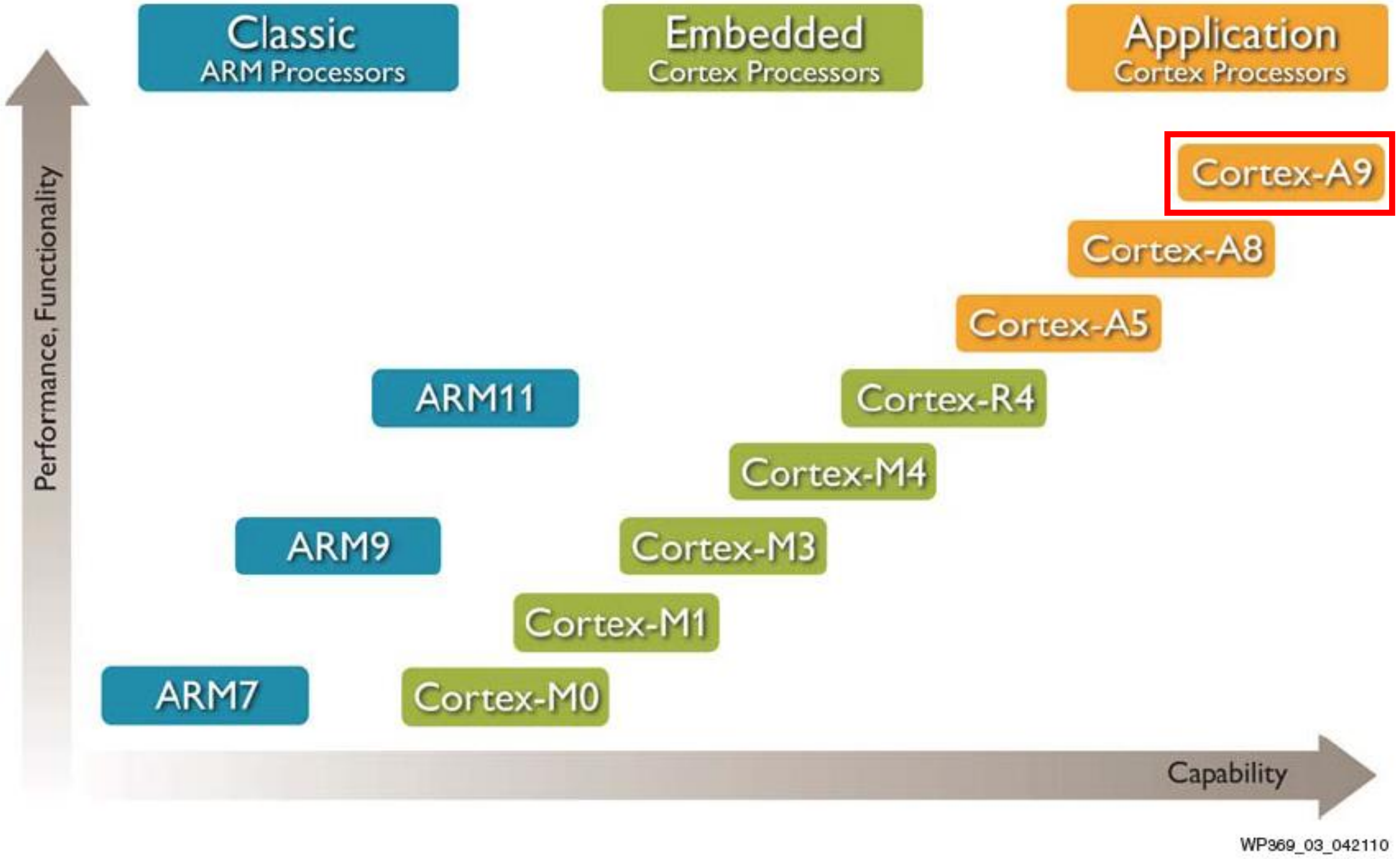
Custom Accelerators

Zynq

- Not an ordinary FPGA
- Not an ordinary microprocessor
- Unique blend of two technologies with powerful interconnect
- In Zynq, the ARM Cortex-A9 is an application grade processor, capable of running full operating systems such as Linux, while the programmable logic is based on Xilinx 7-series FPGA architecture



ARM Processor Roadmap



ARM

Samsung GALAXY S III

Apple A5:

iPhone 4S iPad 2

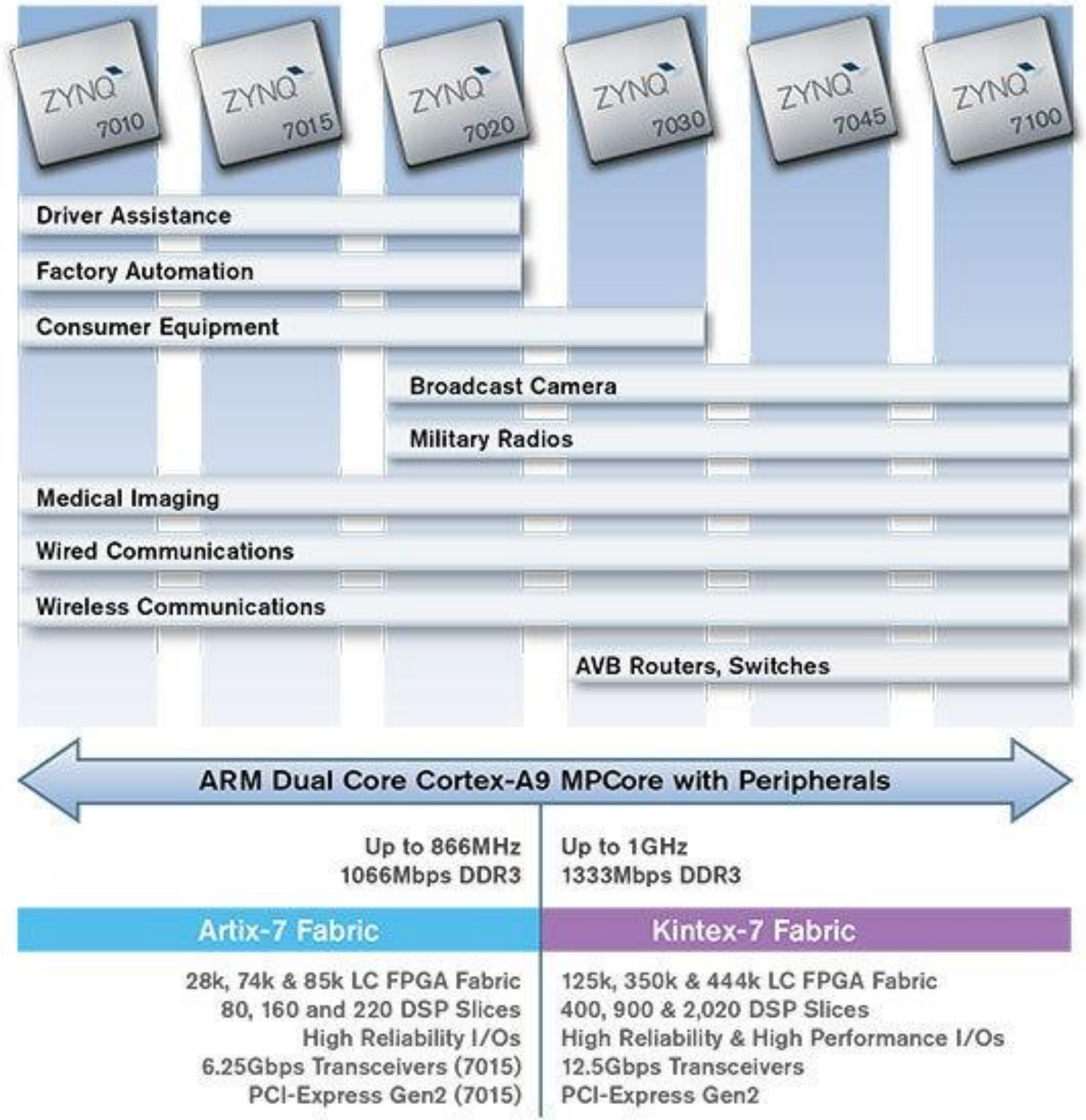
PSVITA

PlayStation Vita

MOTOROLA XOOM™

Zynq

- All programmable SoC (APSoC)
- Offers software and hardware and IO Programmability on single SoC
- Is there any link between Zynq and Zinc?



Up to 866MHz
1066Mbps DDR3

Up to 1GHz
1333Mbps DDR3

Artix-7 Fabric

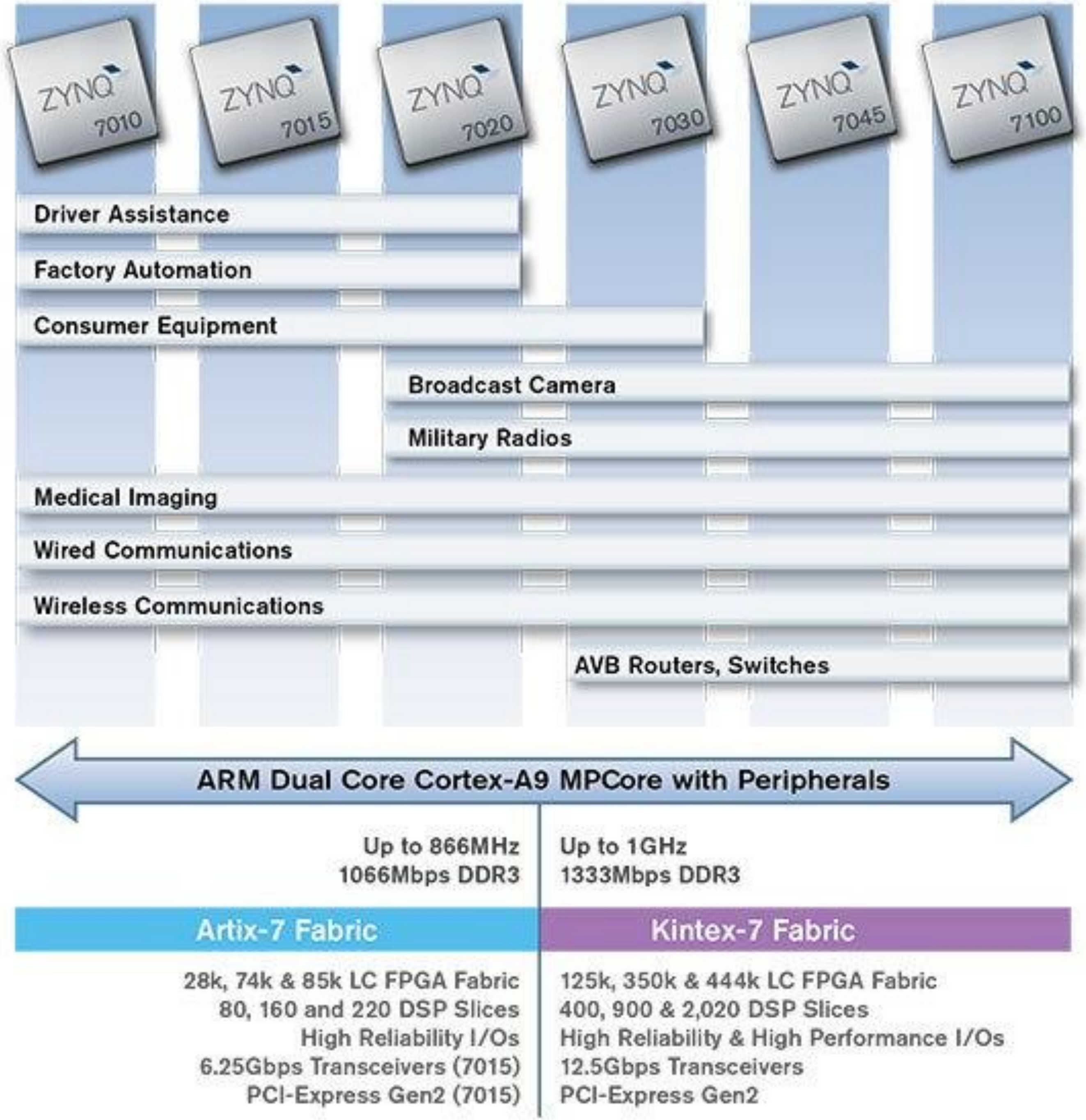
28k, 74k & 85k LC FPGA Fabric
80, 160 and 220 DSP Slices
High Reliability I/Os
6.25Gbps Transceivers (7015)
PCI-Express Gen2 (7015)

Kintex-7 Fabric

125k, 350k & 444k LC FPGA Fabric
400, 900 & 2,020 DSP Slices
High Reliability & High Performance I/Os
12.5Gbps Transceivers
PCI-Express Gen2

Zynq

- All programmable SoC (APSoC)
- Offers software and hardware and IO Programmability on single SoC
- Is there any link between Zynq and Zinc?
- Zynq devices are intended to be flexible and form a compelling platform for a wide variety of applications, just as the metal zinc can be mixed with various other metals to form alloys with differing desirable properties.



Driver Assistance

Factory Automation

Consumer Equipment

Broadcast Camera

Military Radios

Medical Imaging

Wired Communications

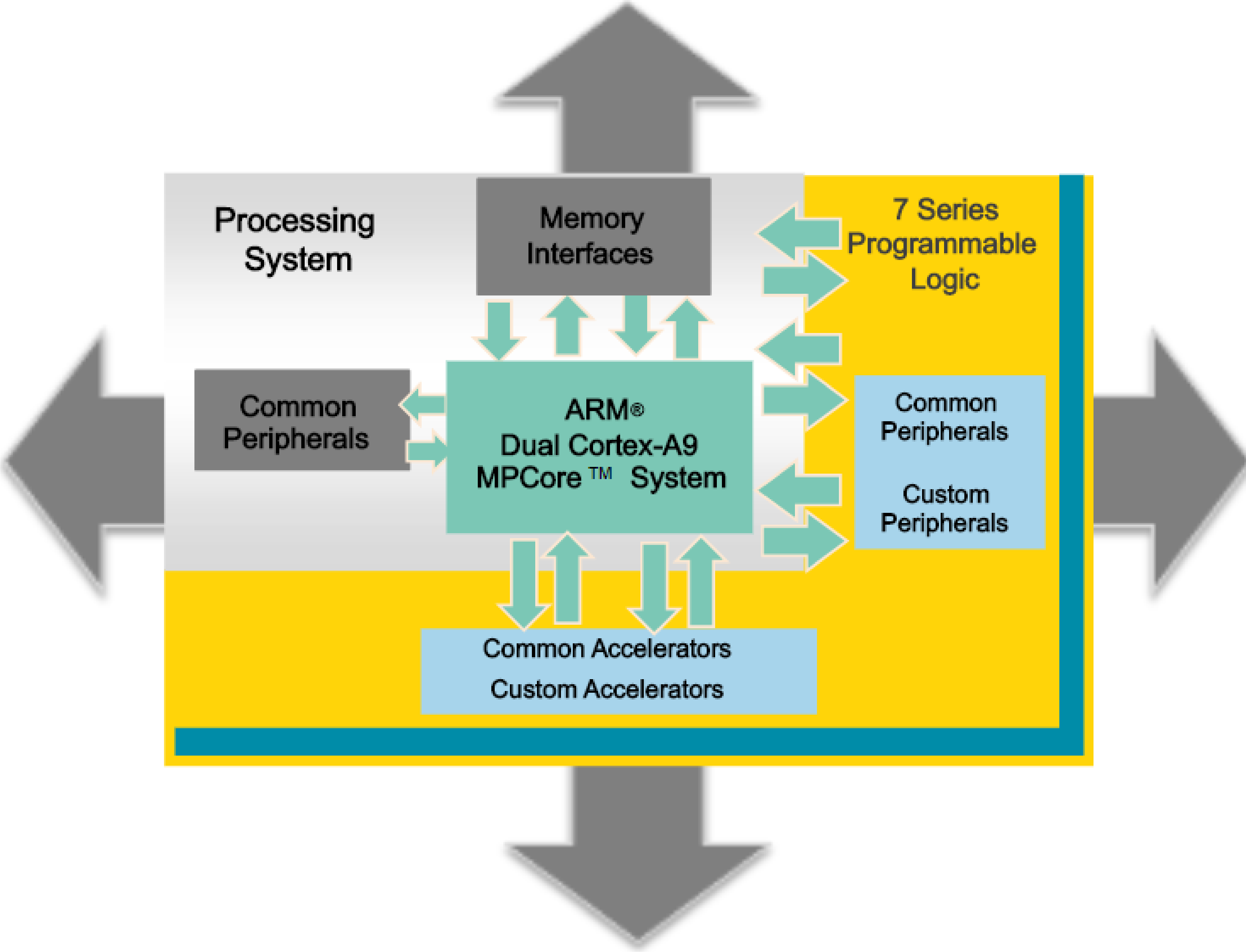
Wireless Communications

AVB Routers, Switches

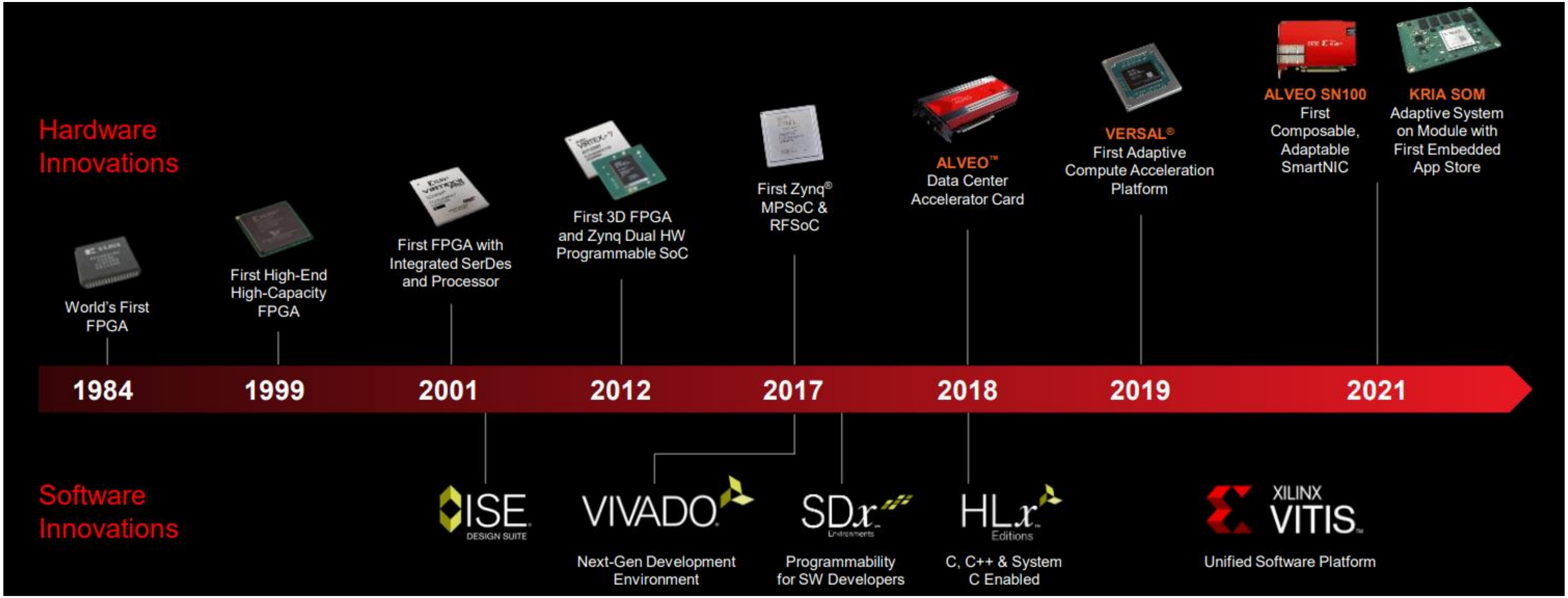
ARM Dual Core Cortex-A9 MPCore with Peripherals

Up to 866MHz 1066Mbps DDR3	Up to 1GHz 1333Mbps DDR3
Artix-7 Fabric	Kintex-7 Fabric
28k, 74k & 85k LC FPGA Fabric 80, 160 and 220 DSP Slices High Reliability I/Os 6.25Gbps Transceivers (7015) PCI-Express Gen2 (7015)	125k, 350k & 444k LC FPGA Fabric 400, 900 & 2,020 DSP Slices High Reliability & High Performance I/Os 12.5Gbps Transceivers PCI-Express Gen2

Zynq EPP


Processing System
Memory Interfaces
7 Series Programmable Logic
Common Peripherals
ARM® Dual Cortex-A9 MPCore ™ System
Common Peripherals
Custom Peripherals
Common Accelerators
Custom Accelerators

Zynq Evolution



Hardware Innovations

1984 – World's First FPGA

1999 – First High-End High-Capacity FPGA

2001 – First FPGA with Integrated SerDes and Processor

2012 – First 3D FPGA and Zynq Dual HW Programmable SoC

2017 – First Zynq® MPSoC & RFSoC

2018 – ALVEO™ Data Center Accelerator Card

2019 – VERSAL® First Adaptive Compute Acceleration Platform

2021 – ALVEO SN100 First Composable, Adaptable SmartNIC

2021 – KRIA SOM Adaptive System on Module with First Embedded App Store

Software Innovations

ISE DESIGN SUITE

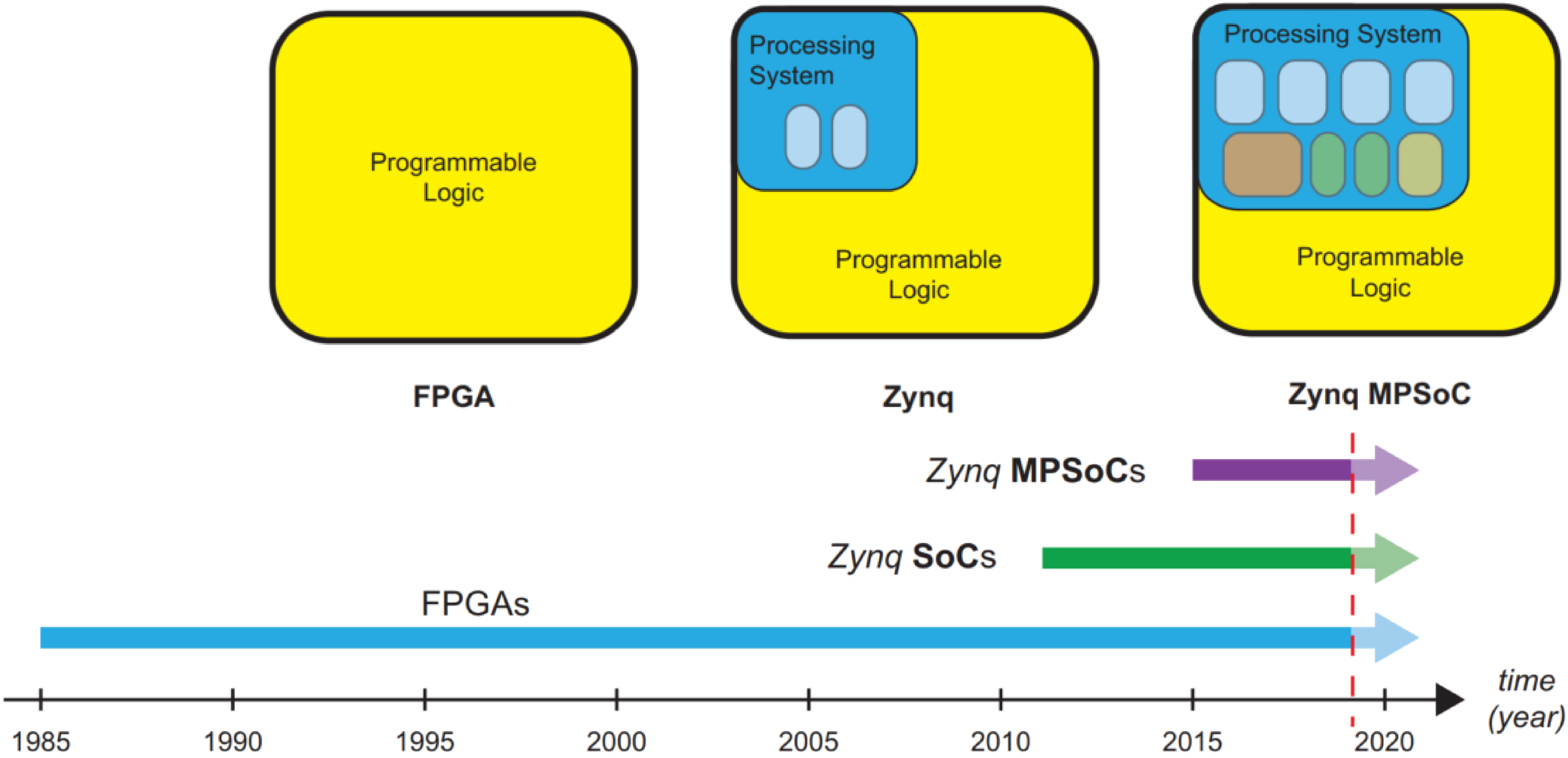
VIVADO – Next-Gen Development Environment

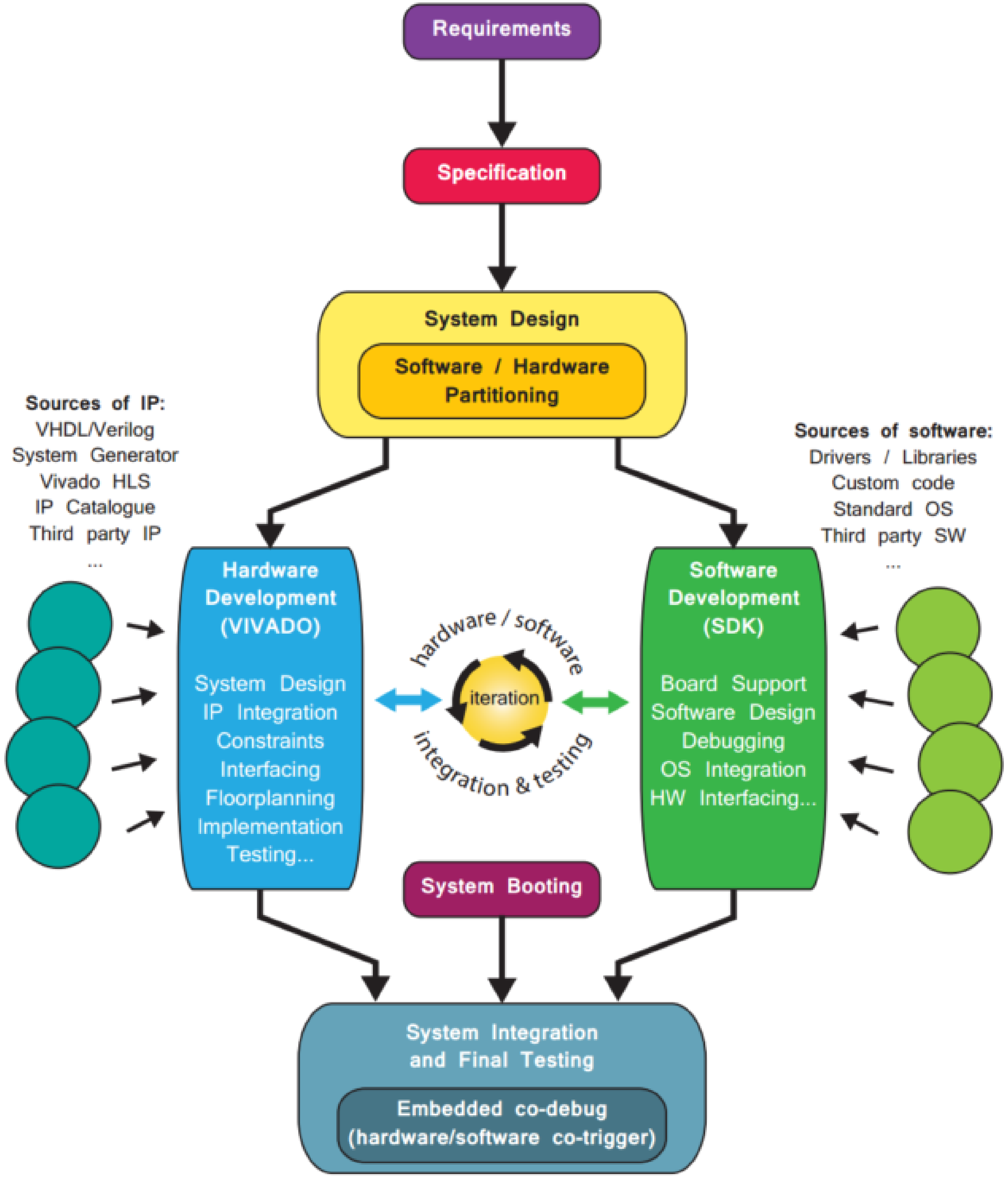
SDx Environments – Programmability for SW Developers

HLx Editions – C, C++ & System C Enabled

XILINX VITIS – Unified Software Platform

Zynq Evolution





Requirements

Specification

System Design

Software / Hardware Partitioning

Sources of IP:
VHDL/Verilog
System Generator
Vivado HLS
IP Catalogue
Third party IP
...

Hardware Development (VIVADO)

System Design
IP Integration
Constraints
Interfacing
Floorplanning
Implementation
Testing...

Sources of software:
Drivers / Libraries
Custom code
Standard OS
Third party SW
...

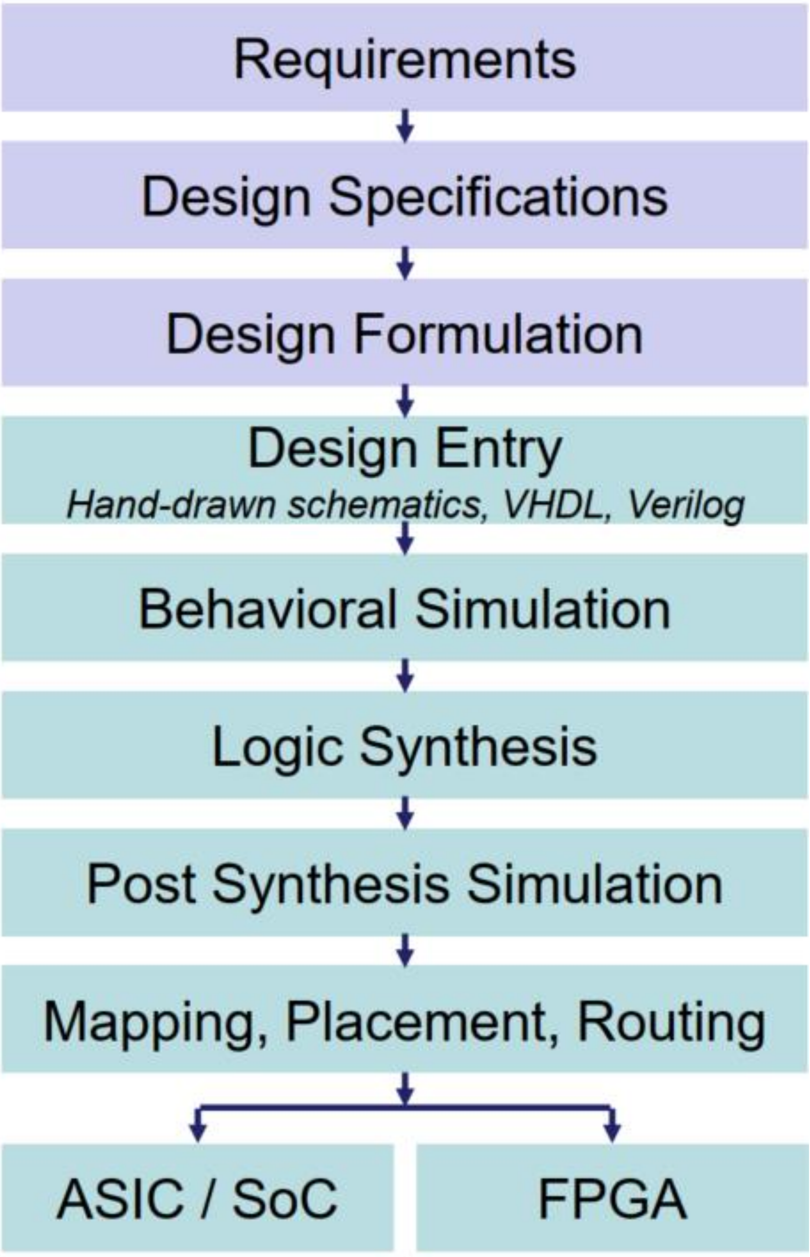
Software Development (SDK)

Board Support
Software Design
Debugging
OS Integration
HW Interfacing...

System Booting

System Integration and Final Testing

Embedded co-debug (hardware/software co-trigger)



Requirements

Design Specifications

Design Formulation

Design Entry
Hand-drawn schematics, VHDL, Verilog

Behavioral Simulation

Logic Synthesis

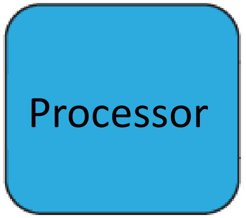
Post Synthesis Simulation

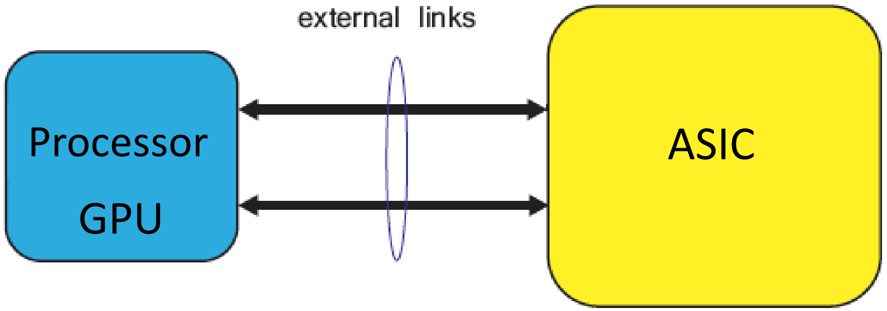
Mapping, Placement, Routing

ASIC / SoC

FPGA

SoC Architectures

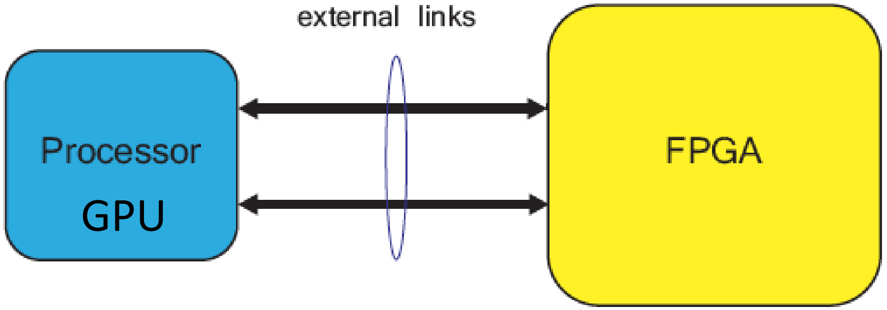

Processor


external links

Processor

GPU

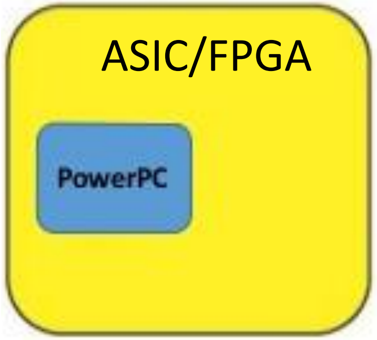
ASIC


external links

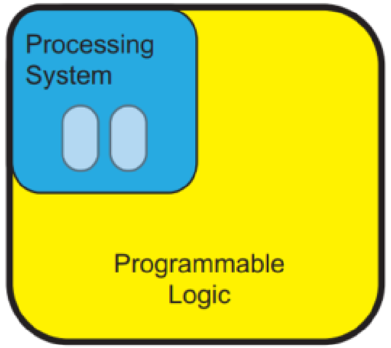
Processor

GPU

FPGA

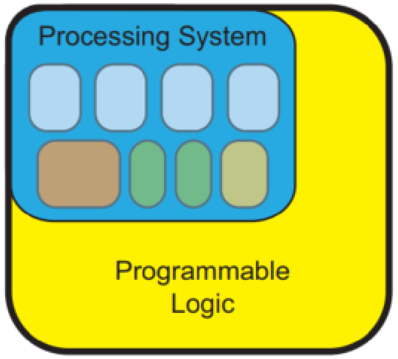

ASIC/FPGA

PowerPC


Processing System

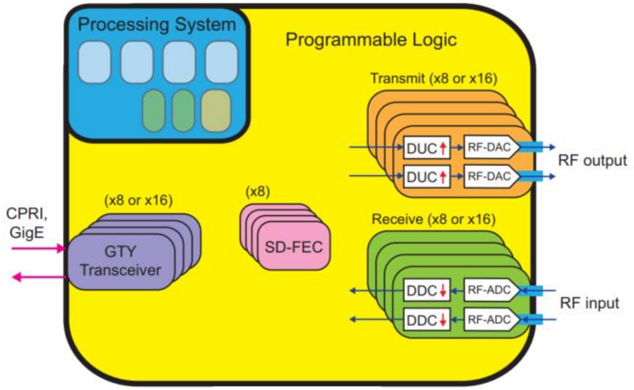
Programmable Logic

APSOC

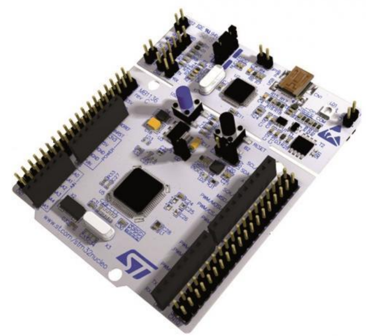

Processing System

Programmable Logic

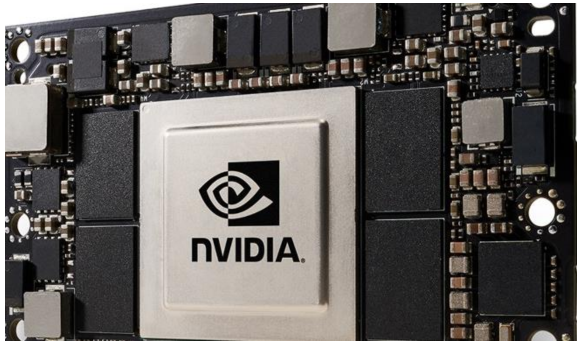
MPSoC



RFSOC

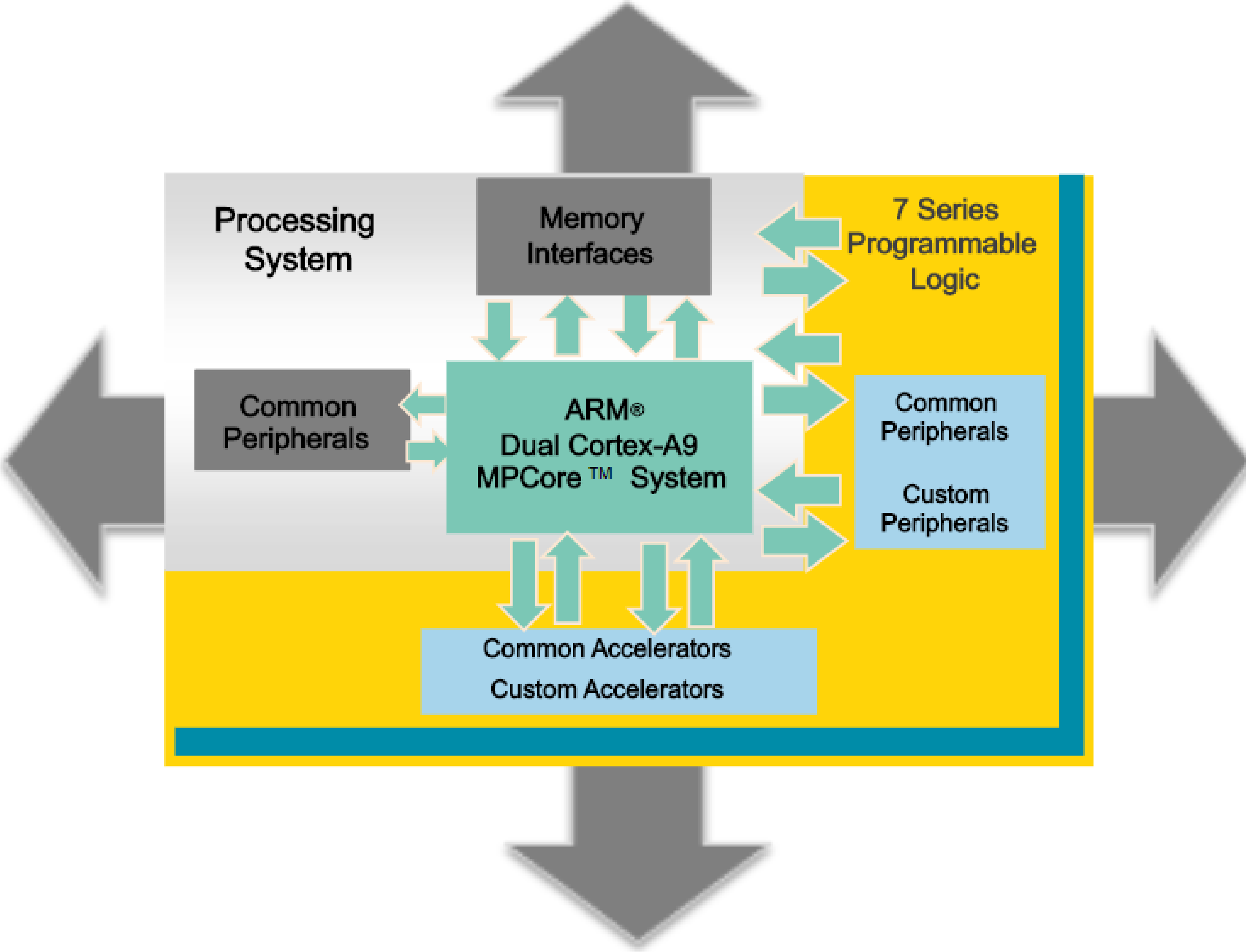


Heterogenous All Programmable SoC

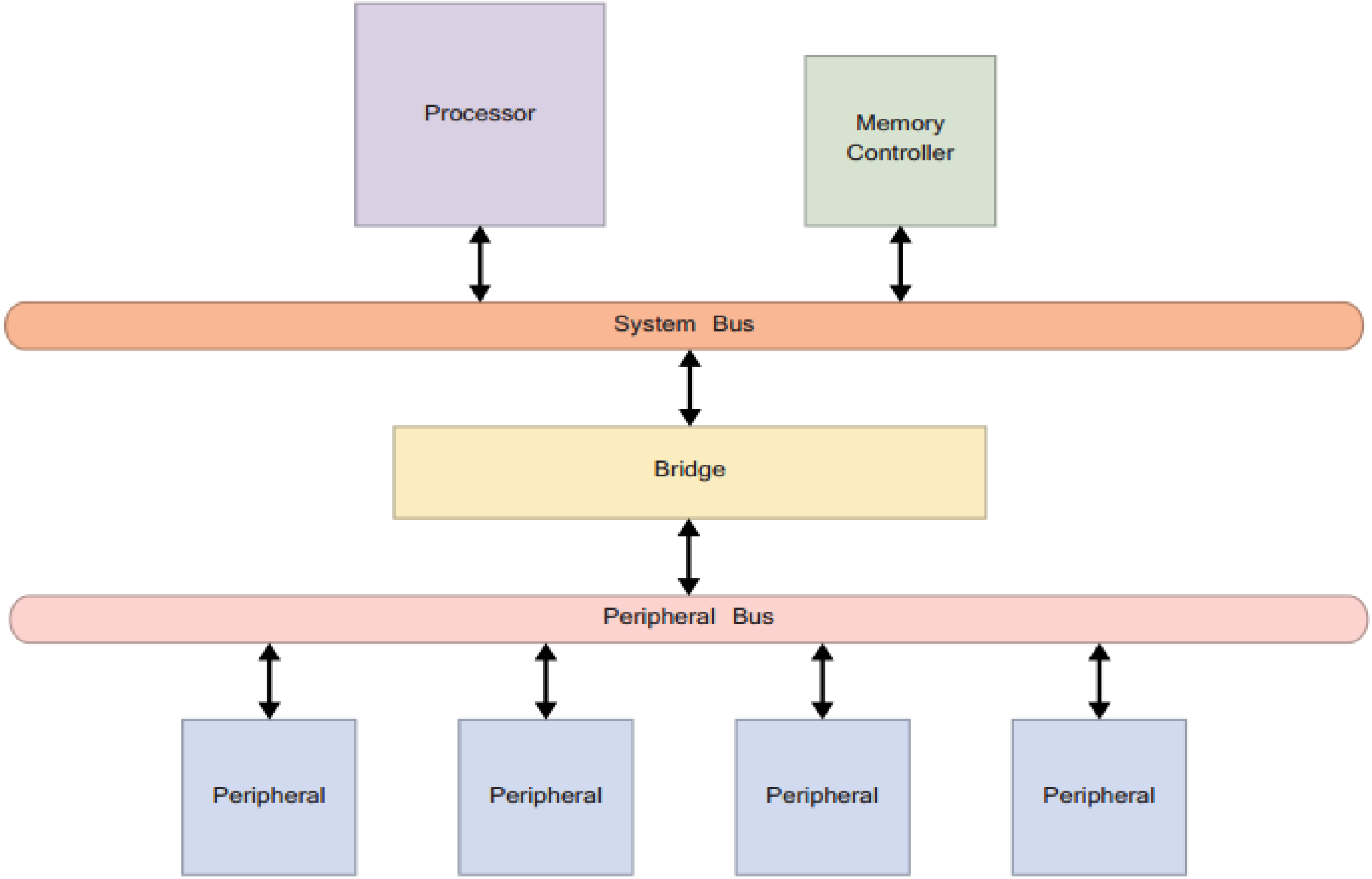


Zynq Book by Xilinx

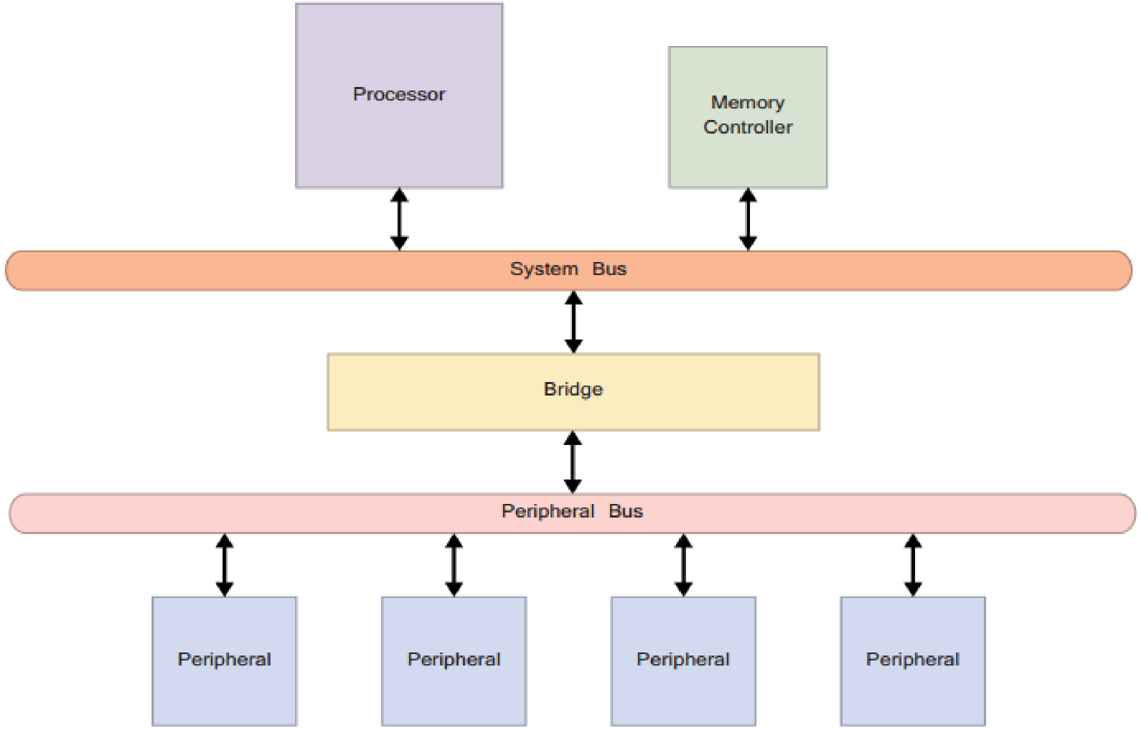
Zynq EPP


Processing System
Memory Interfaces
7 Series Programmable Logic
Common Peripherals
ARM® Dual Cortex-A9 MPCore ™ System
Common Peripherals
Custom Peripherals
Common Accelerators
Custom Accelerators

Embedded System

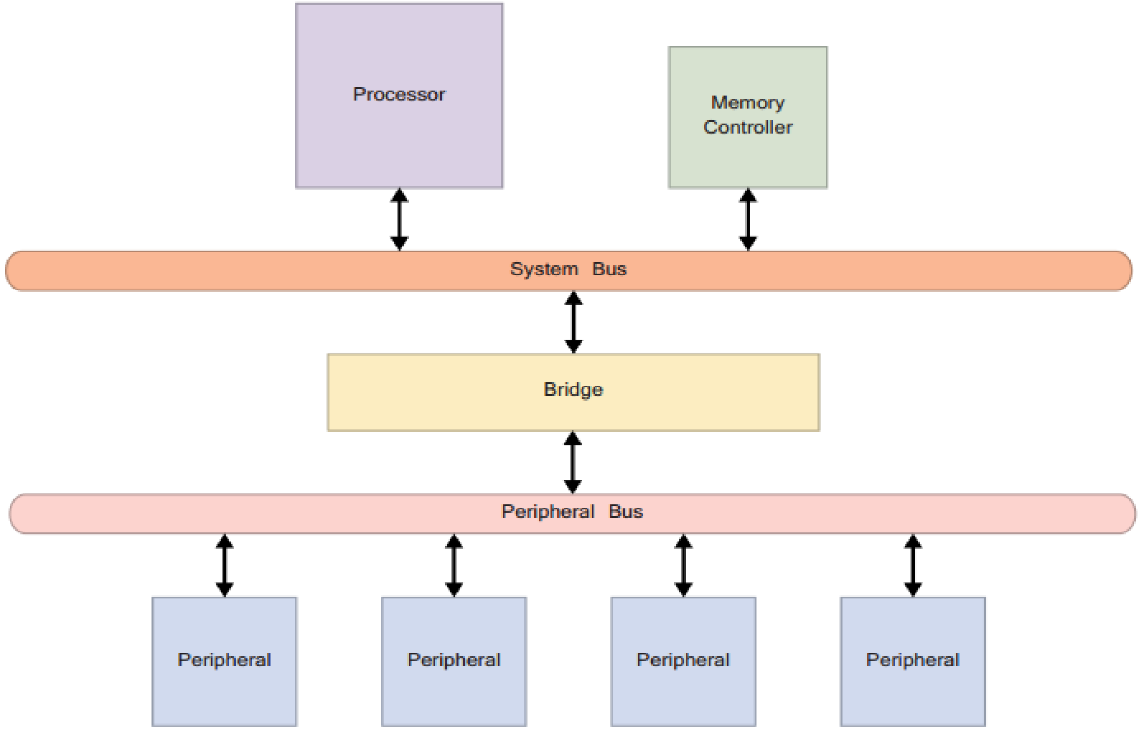


Embedded System



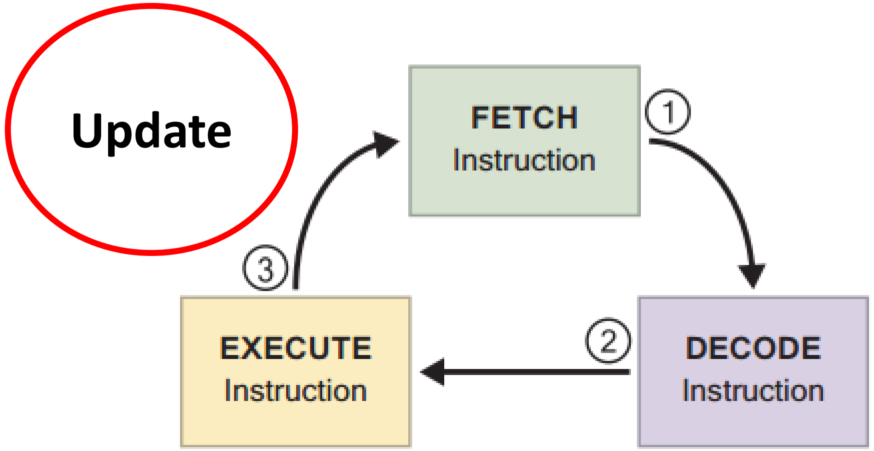
- **Processor** — This is the ‘brain' of the system. It is programmed to perform the tasks specific to the application of the embedded system.

Embedded System



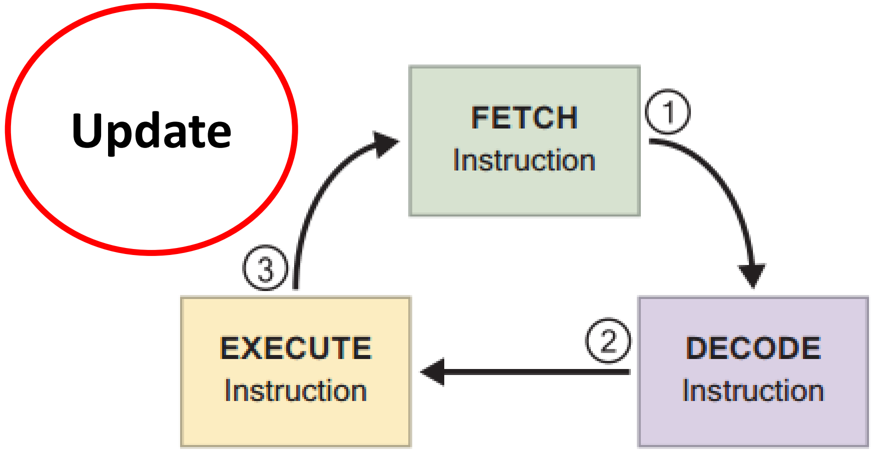
- **Memory Controller** — Memory controllers manage the reading and writing of data to and from main memory in an embedded system. Provides an interface between the system memory and all other parts of the system.

Execution Cycles



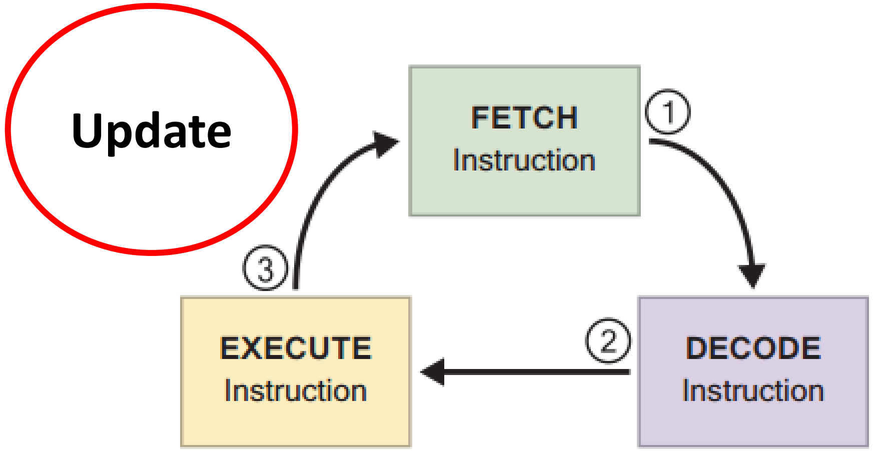
- In order for a program that is stored in memory to be executed by the processor it must go through an *instruction execution cycle*.
- This is the process by which a system retrieves an instruction from memory, determines the required actions for that instruction, and executes those actions.
- Also referred to as the *fetch-and-execute* cycle or the *fetch-decode-execute cycle*

Execution Cycles



- To run a program, the program code is copied from secondary storage into the primary memory.
- The CPU’s program counter indicates the memory location where the first command in the program has been saved and execution starts.
- A program counter keeps track of the memory address of the command to be executed next.
- In a program, each instruction occupies a space in the main memory. The program counter saves the address of each instruction and instructs the CPU in what sequence they should be executed.

Execution Cycles



- In plain CPUs the fetch execute cycle is carried out progressively, each instruction is being handled before the succeeding one is initiated.
- In modern CPUs the fetch execute cycle is executed simultaneously, in parallel via an instruction pipeline, which means the next command starts being executed before the previous command has completed.

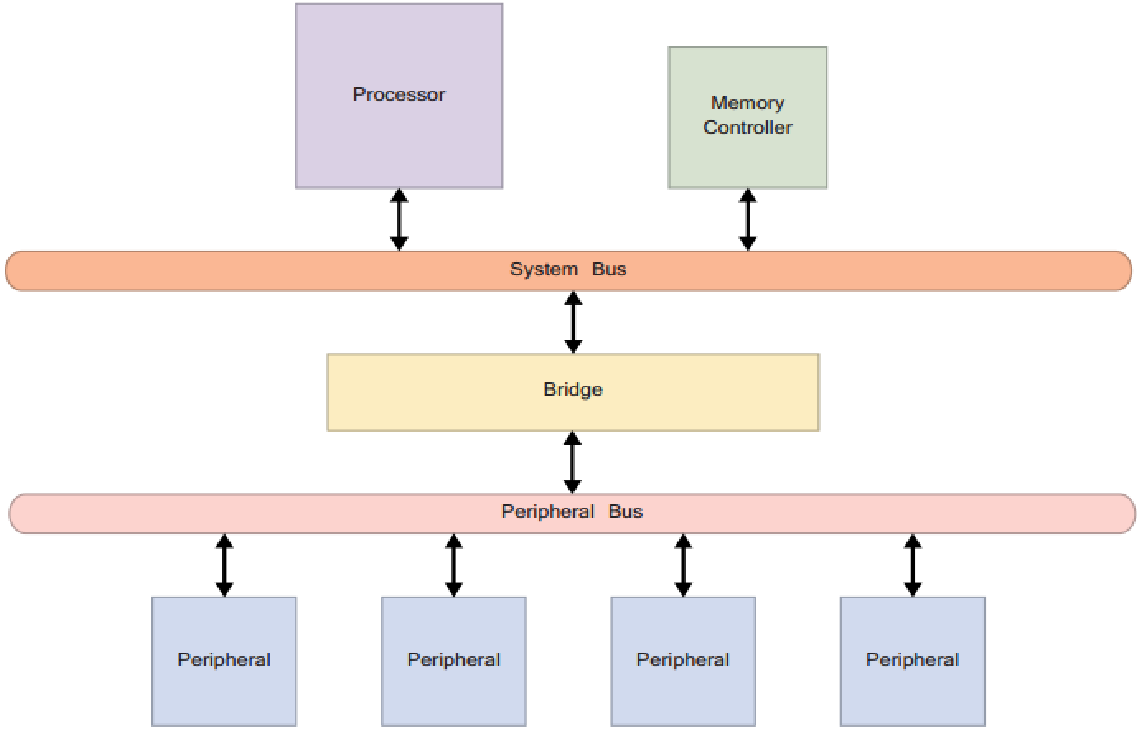
Cycle

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Fetch	A					B					C				
Decode		A					B					C			
Execute			A					B					C		
Memory				A					B					C	
Write					A					B					C

Cycle

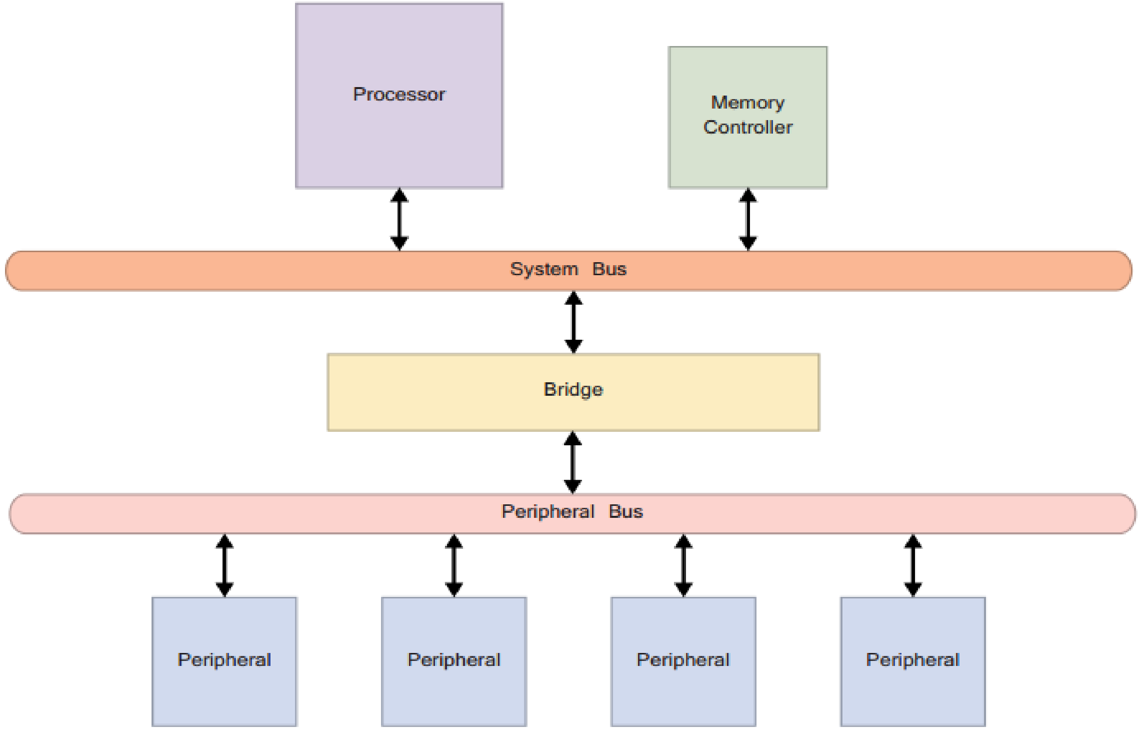
	1	2	3	4	5	6	7
Fetch	A	B	C				
Decode		A	B	C			
Execute			A	B	C		
Memory				A	B	C	
Write					A	B	C

Embedded System



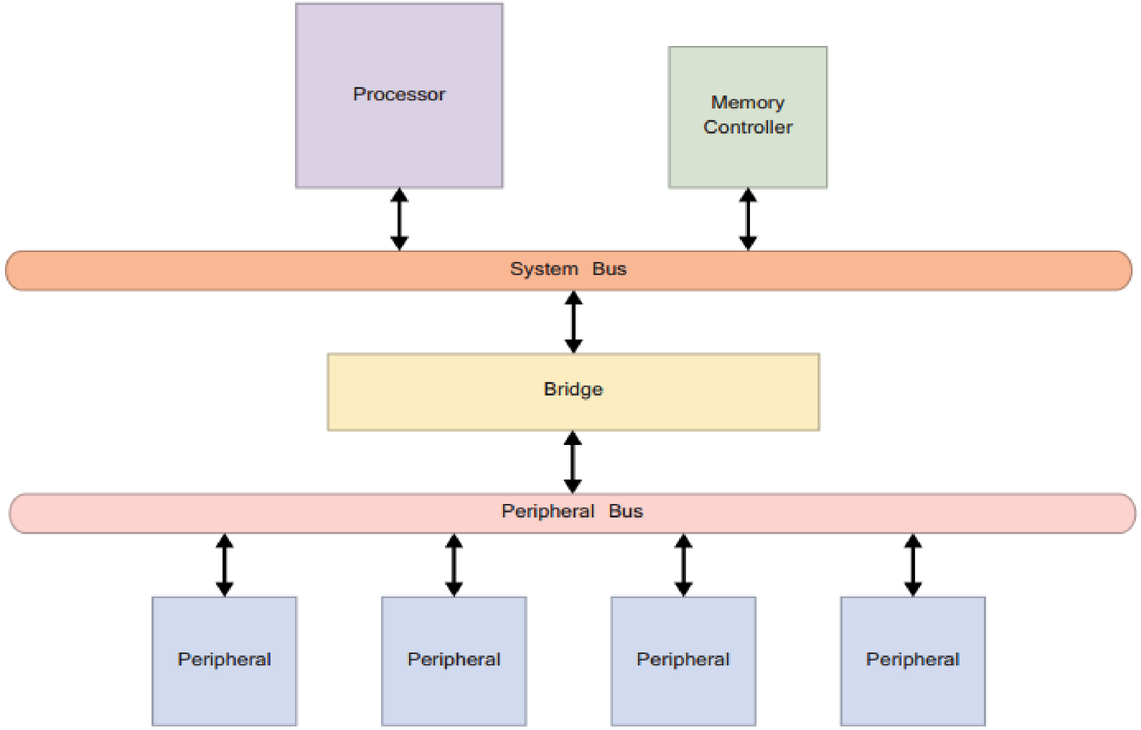
- **Peripherals** — These are the components around the central processing unit.
- Peripherals can be implemented as individual integrated circuits, be contained on-chip with the processor, or may reside in an area of programmable logic such as an FPGA.

Embedded System



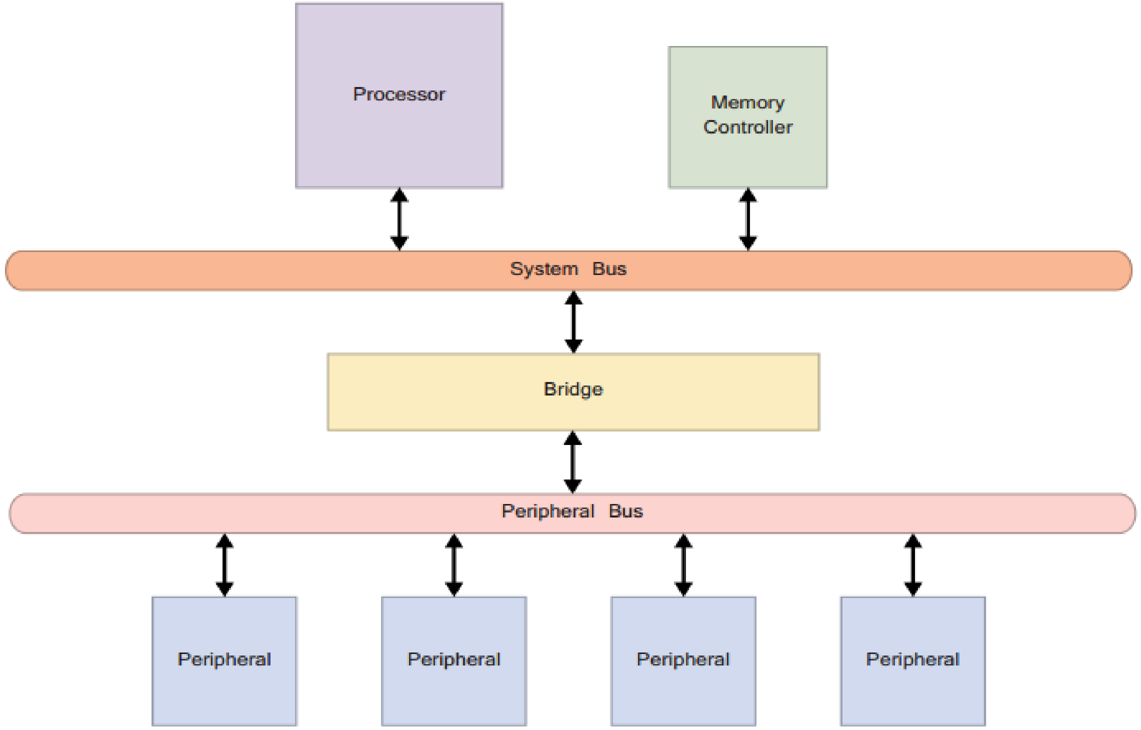
- **System Bus** — In an embedded system with multiple buses, the system bus connects the processor, memory controller and other high-speed devices together.
- **Peripheral Bus** — Allows devices on the peripheral bus to communicate with each other, even when a high-priority processor-memory transaction is taking place on the system bus.

Embedded System



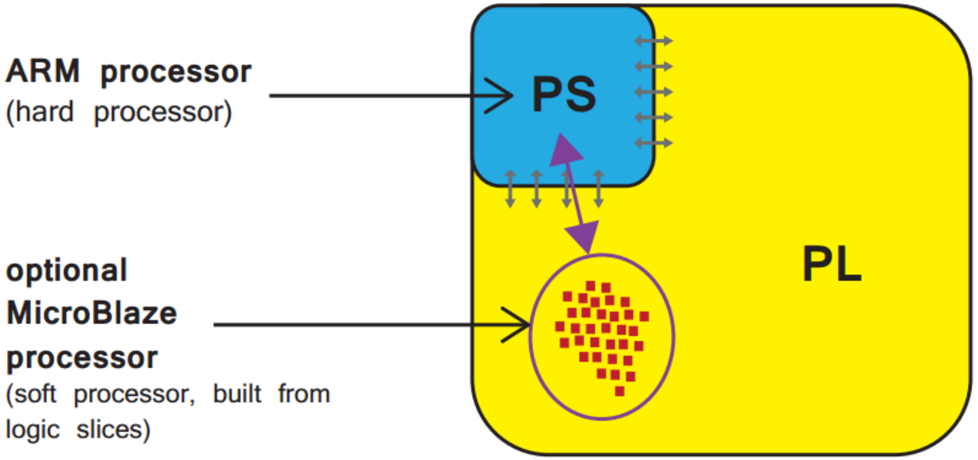
- A **coprocessor** is a processing core that supplements the functionality of the primary processor, and is optimised for a single, specific task.
- By off-loading computations from the main processor to one or more co-processing units, the overall system performance can be accelerated.

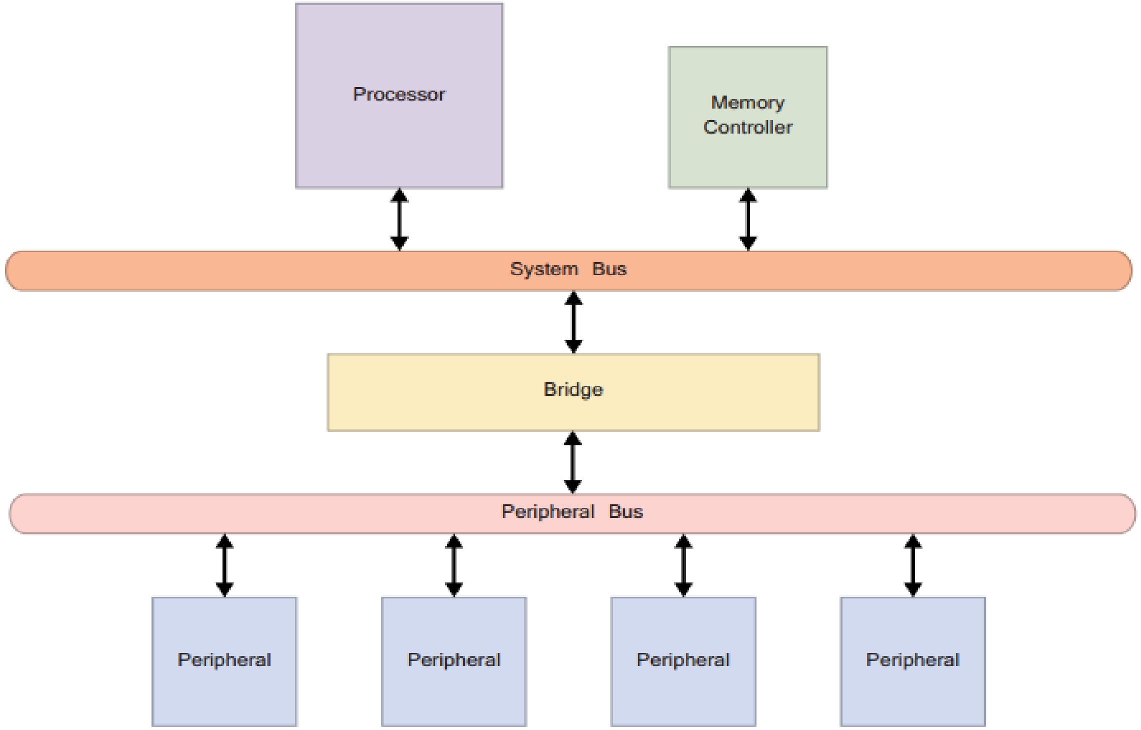
Embedded System



- Whereas the main processor may be used for a variety of different tasks, co-processors are generally used to perform dedicated tasks.
- Examples :
  - High-speed arithmetic
  - Image and video processing
  - Digital signal processing
  - Data encryption

Co-processors


ARM processor
(hard processor)
PS
PL
optional MicroBlaze processor
(soft processor, built from logic slices)



- With reference to Zynq SoC EPP, the programmable logic (PL: FPGA) provides a perfect platform to create co-processing cores due to the ability to perform parallel execution.
- **Other co-processors:** FPE, NEON, GPU

Convolution/Filter Task

$w_o = 1, w_1 = 2, w_2 = 3$

$x_0 = 1 \; then \; y_0 = \; x_0 w_0 = 1$

$x_1 = 2 \; then \; y_1 = x_0 w_1 + x_1 w_0 = 4$

$x_2 = 1 \; then \; y_2 = x_0 w_2 + x_1 w_1 + x_2 w_0 = 8$

$x_3 = 2 \; then \; y_3 = x_1 w_2 + x_2 w_1 + x_3 w_0 = 10$

- Parallel computing process by nature
- N number of taps/weights/filter coefficients
- N multiplications should happen in parallel

Viewed as an Equation

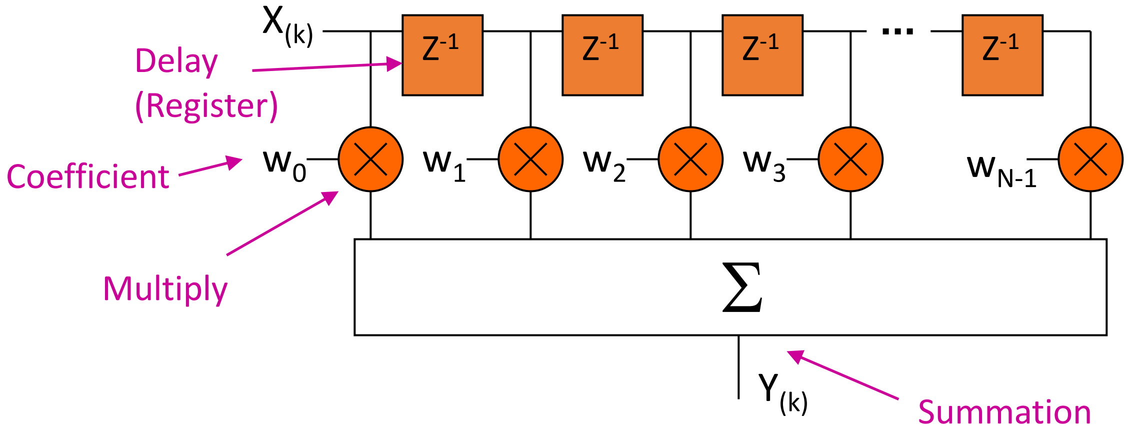
Coefficients

$$\mathbf{Y_{(k)} = \sum_{i=0}^{i=N-1} w_i . X_{(k-i)}}$$

Accumulate N times

Multiply

Viewed as a Diagram



Sequential vs. Parallel DSP Processing

SIMD:
Single
Instruction
Multiple
data

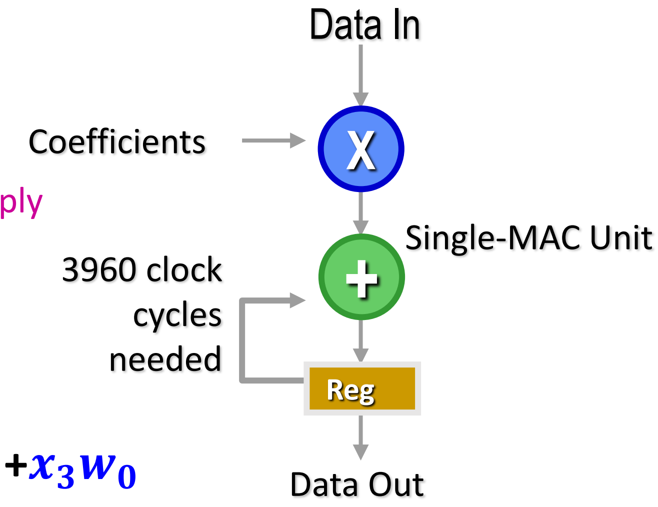
Standard DSP Processor – Sequential (Generic DSP)

$$Y_{(k)} = \sum_{i=0}^{i=N-1} w_i . X_{(k-i)}$$

Coefficients

Multiply

Accumulate
N times

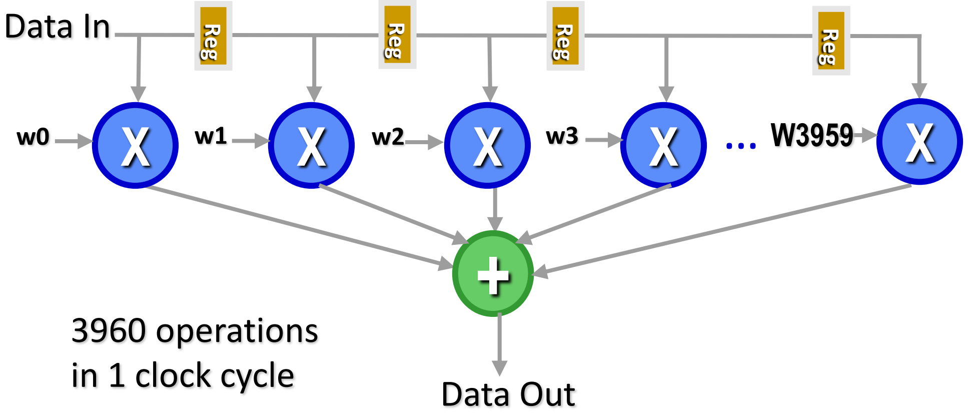


3960 clock
cycles
needed

$$y_3 = x_1 w_2 + x_2 w_1 + x_3 w_0$$

$\frac{\text{1.2 GHz}}{\text{3960 clock cycles}}$ = 303 KSPS

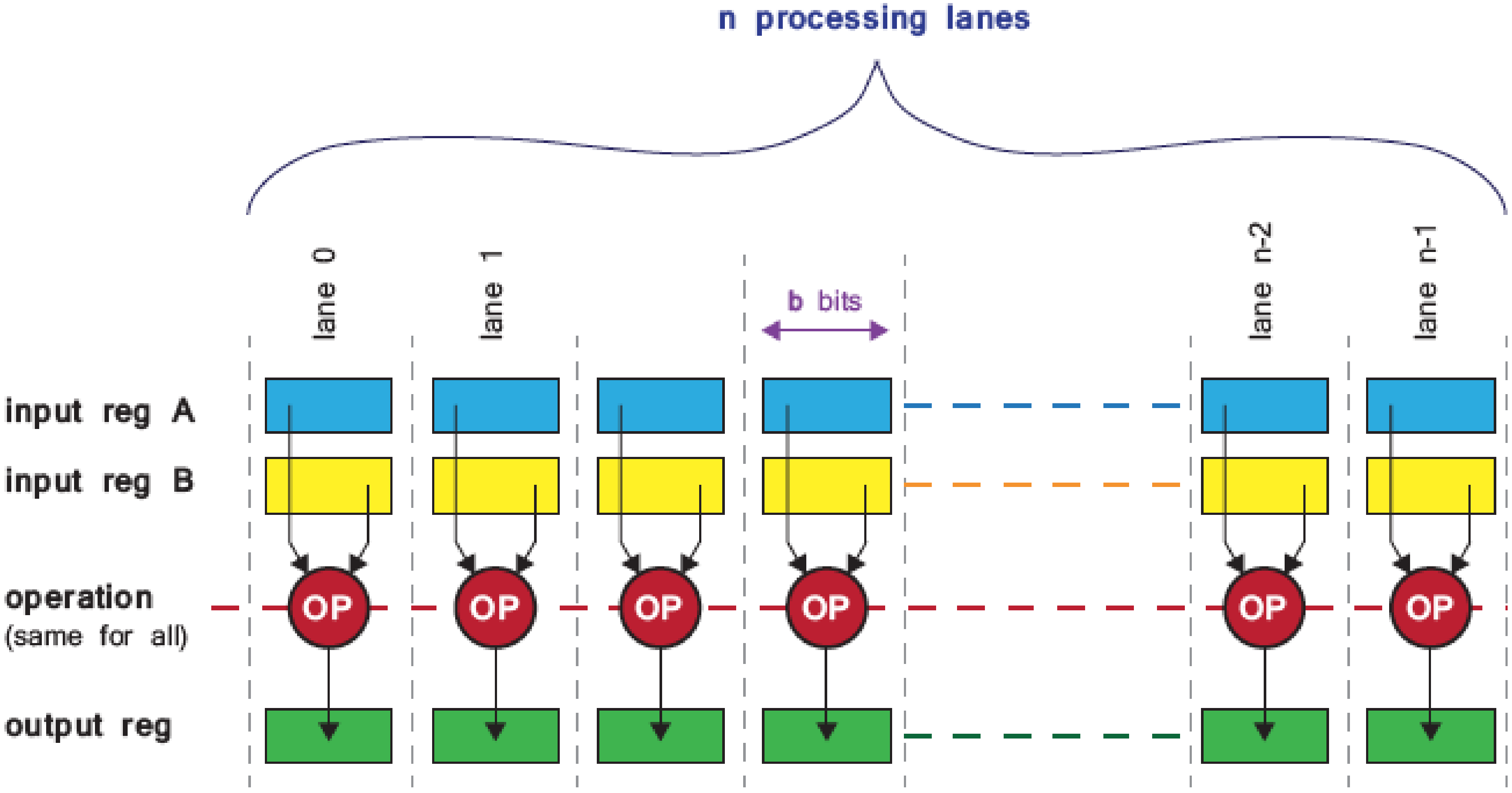
FPGA - Fully Parallel Implementation (7 Series FPGA)



3960 operations
in 1 clock cycle

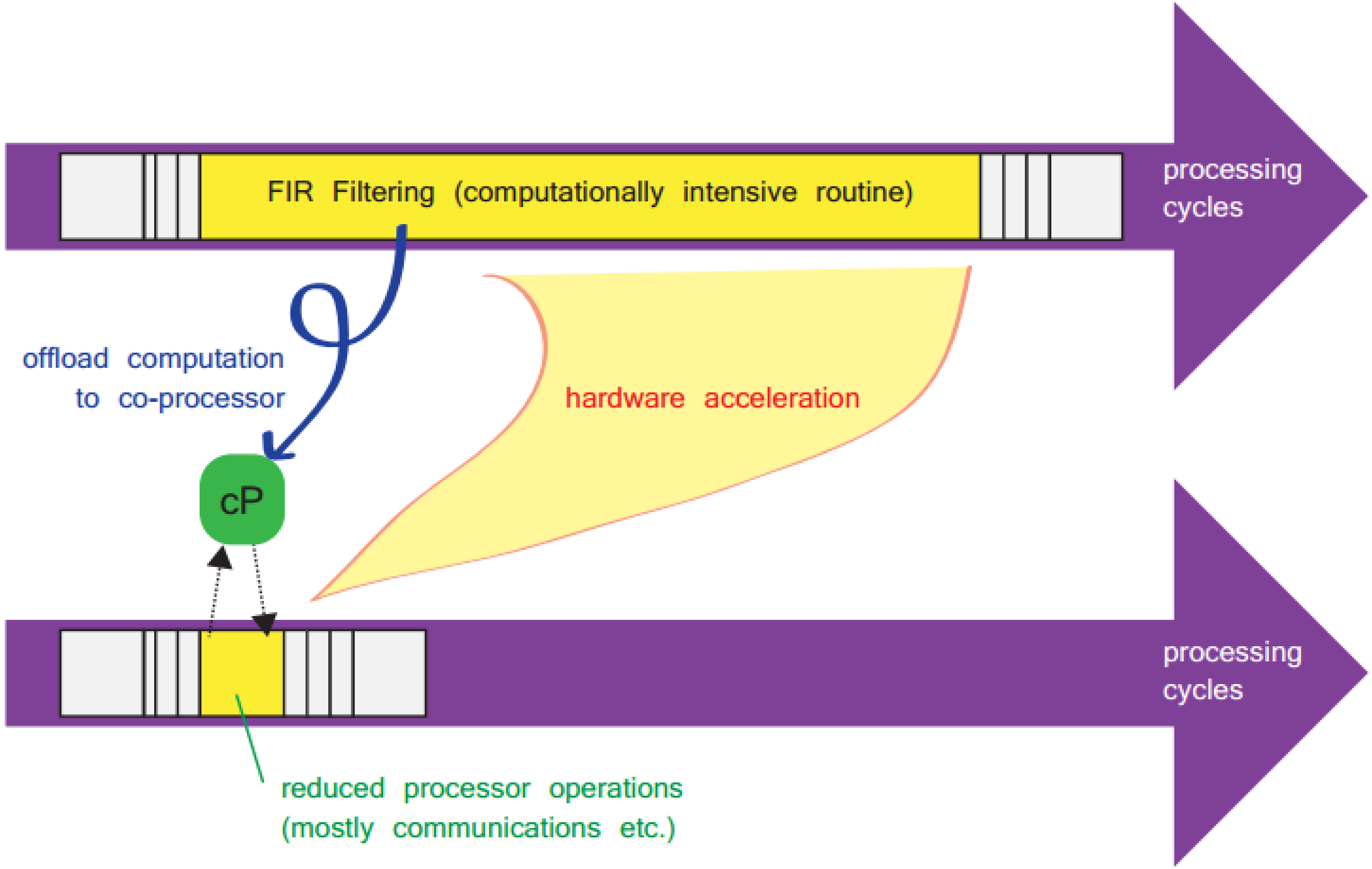
$\frac{\text{600 MHz}}{\text{1 clock cycle}}$ = 600 MSPS

SIMD (Single Instruction Multiple Data) Processing in the NEON



Source: The Zynq Book

SIMD (Single Instruction Multiple Data) Processing in the NEON


FIR Filtering (computationally intensive routine)
processing cycles
offload computation to co-processor
hardware acceleration
cP
processing cycles
reduced processor operations (mostly communications etc.)

- This means that complex tasks that would require a large number of sequential CPU clock cycles to compute can be executed far quicker using the dedicated coprocessor.