Some background from you....
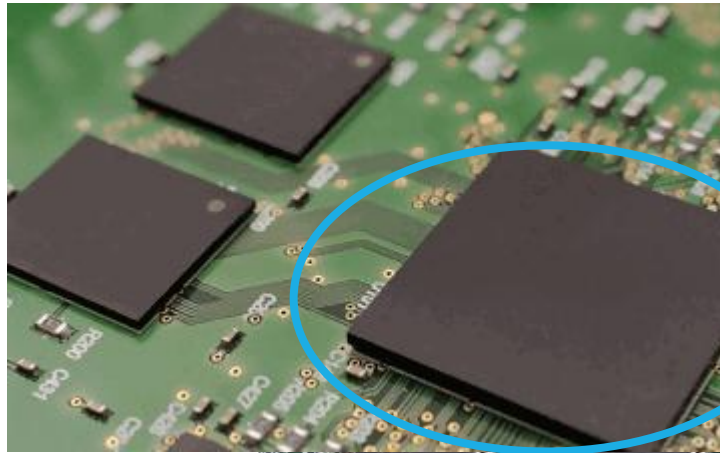
Who knows about VHDL/Verilog?

Who knows about FPGA?

Who knows about SoC?
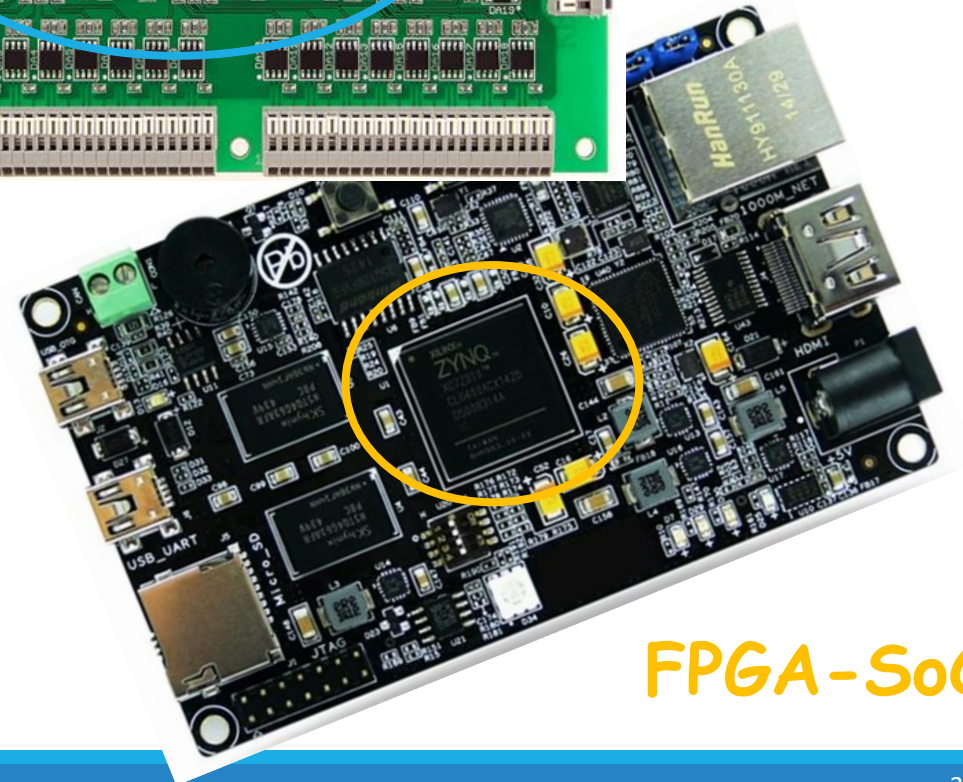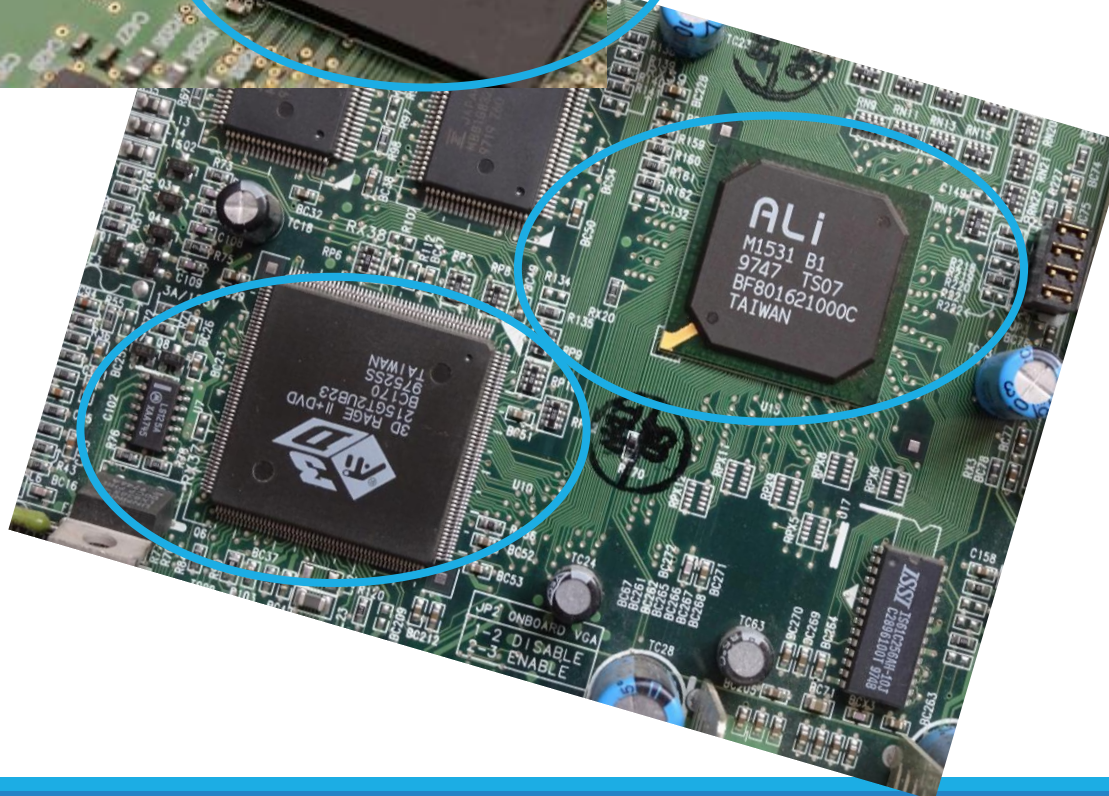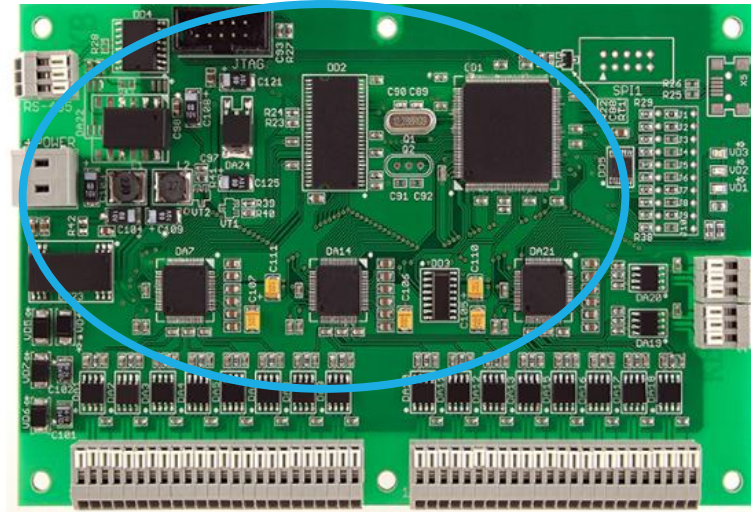
Who knows about ..... ?

Who knows about ..... ?

Who knows about 'C'?

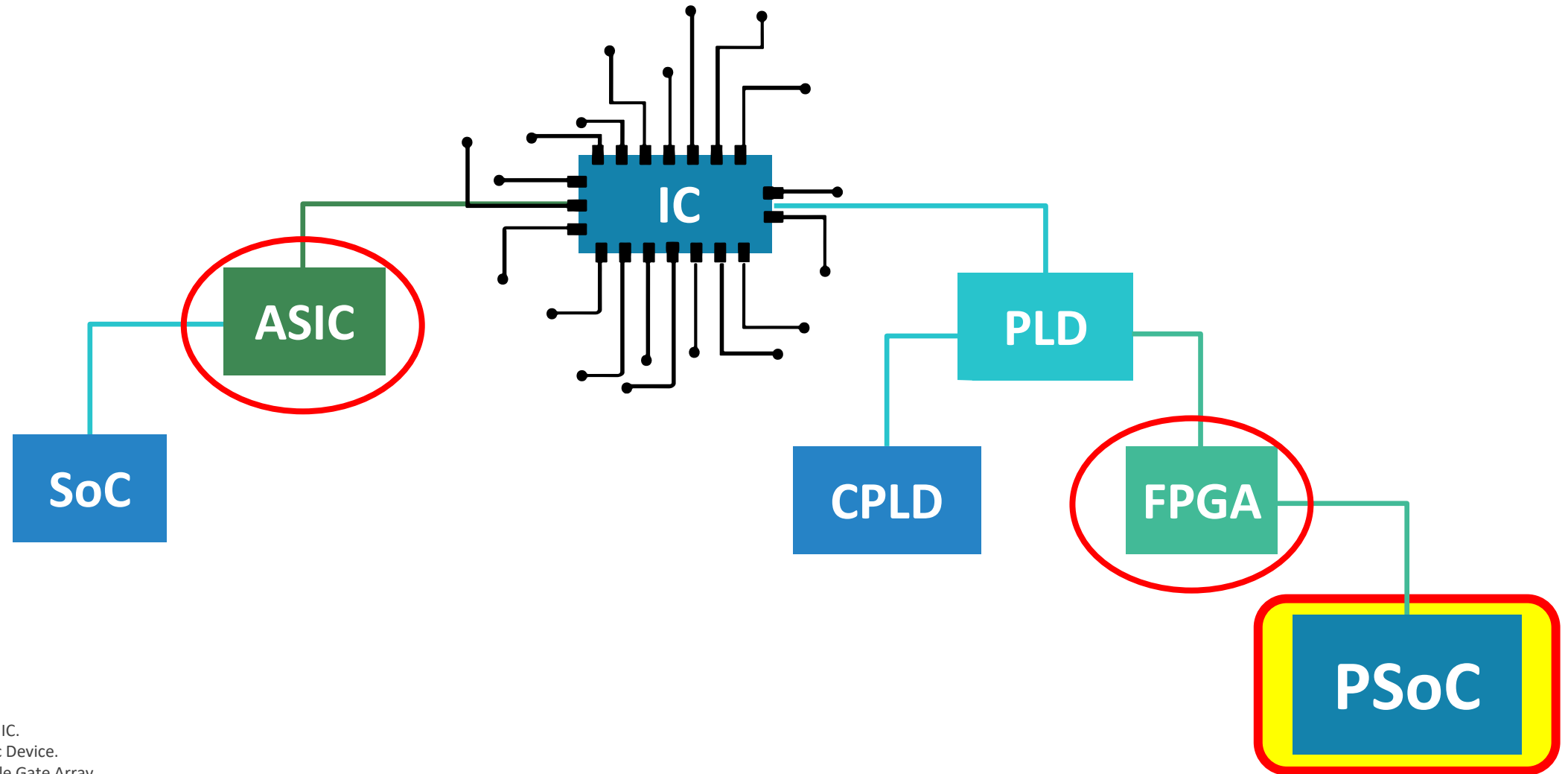# Integrated Circuits - High Level View

ASIC

FPGA-SoC

Integrated Circuit App Clasification

ASIC: application specific IC.
PLD: Programmable Logic Device.
FPGA: Field Programmable Gate Array
SoC: System on a Chip.
PSoC: Programmable System on a Chip.

What is an ASIC ?

An **application-specific integrated circuit** (**ASIC**) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use.

Modern **ASICs** often include entire microprocessors, memory blocks, interface blocks and other large building blocks. Such an **ASIC** is often termed a <u>**SoC**</u> (System-on-a-Chip).

Designers of digital ASICs often use a **Hardware Description Language** (HDL), such as **Verilog** or **VHDL**, to describe the functionality of **ASICs**.

What is an *FPGA* ?

F   P   G   A

Field   **Programmable**   **Gate**   **Array**
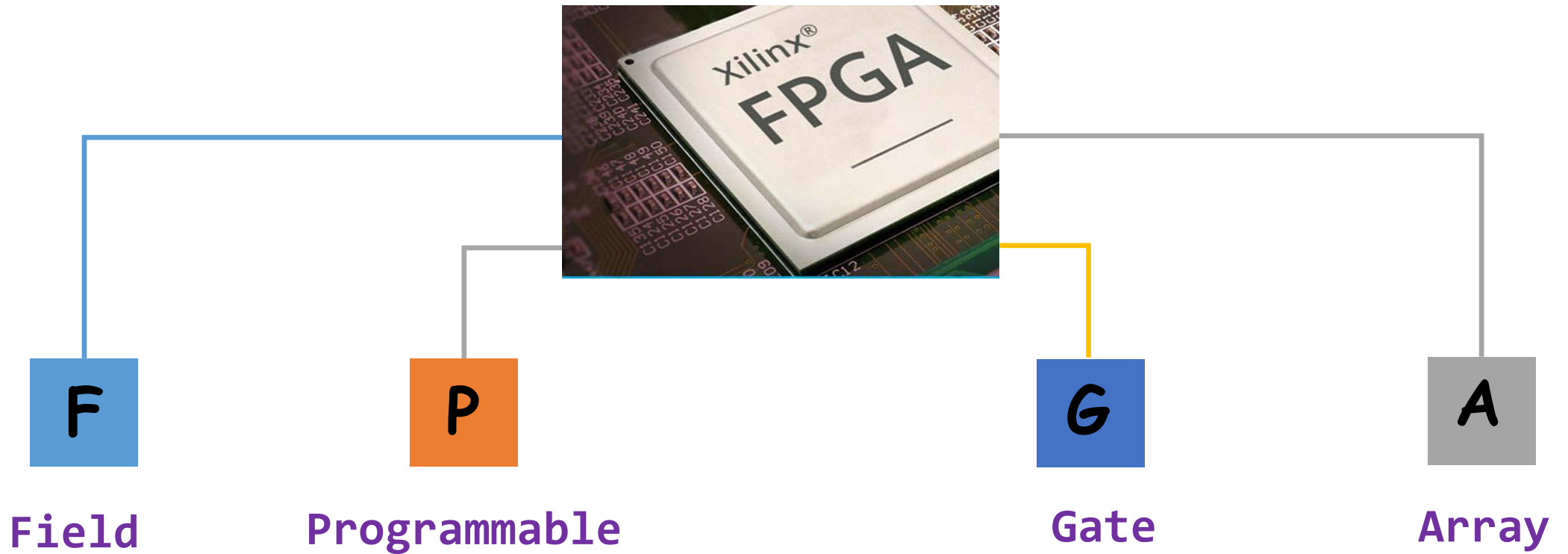
What is an FPGA ?

A **field-programmable gate array** (**FPGA**) is an integrated circuit (IC) _designed to be configured_ by a customer or a designer after manufacturing (that is the reason of the term _field programmable_)

FPGA are not made to be application-specific IC as opposed to **ASICs**

FPGA configuration is generally specified using an HDL language
(either Verilog or VHDL)

FPGA ~ Lego Bricks
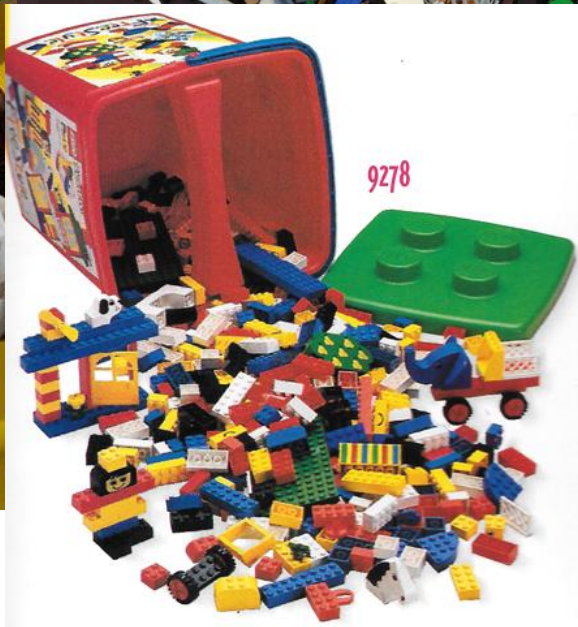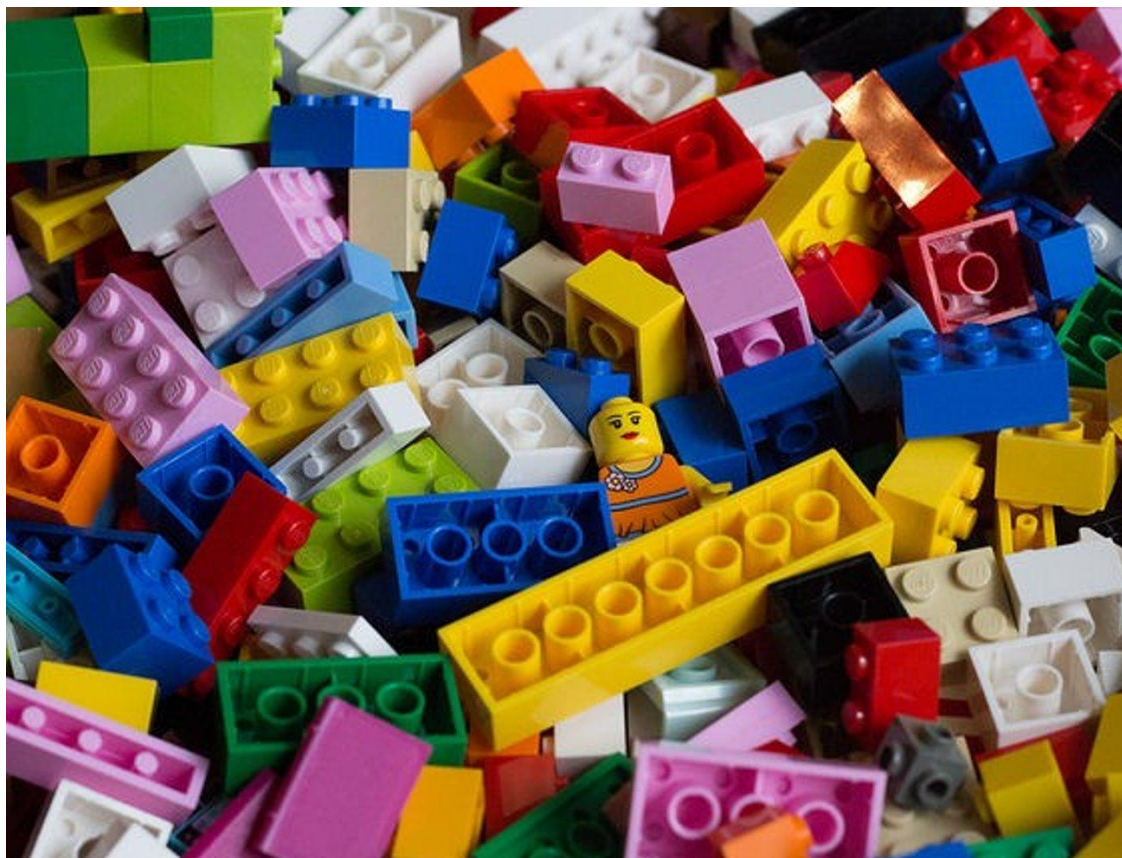
FPGA Basic Architectural View

RAM Memory RAM

DSP Block

Clock Manager

I/O Blocks

Logic + Fip-Flops

Communication Interfaces
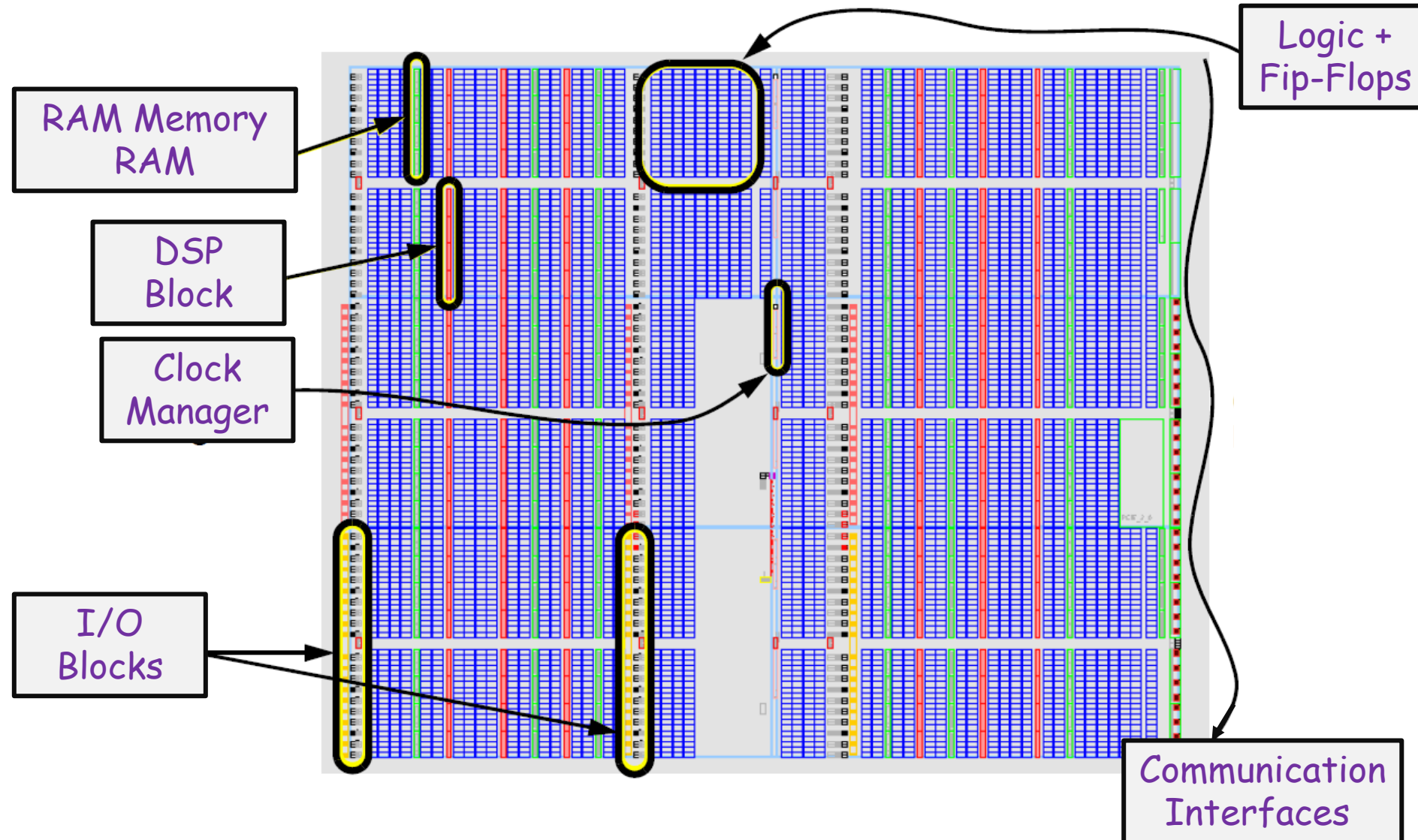
What is a SoC ?

A **S**ystem-**o**n-a-**C**hip (SoC) is an integrated circuit that integrates most or all components of a computer or other electronic system.

A **SoC** usually includes a Central Processing Unit (CPU), Memories, I/O interfaces, Digital Signal Processing (DSP) blocks, digital-analog mixed signals components, etc., all on a single IC.

System-on-a-Chip (SoC)

System on a Board

clocks

decisions

$f_1(x)$ function

interfaces

memory

arithmetic

$f_2(x)$ function

logic
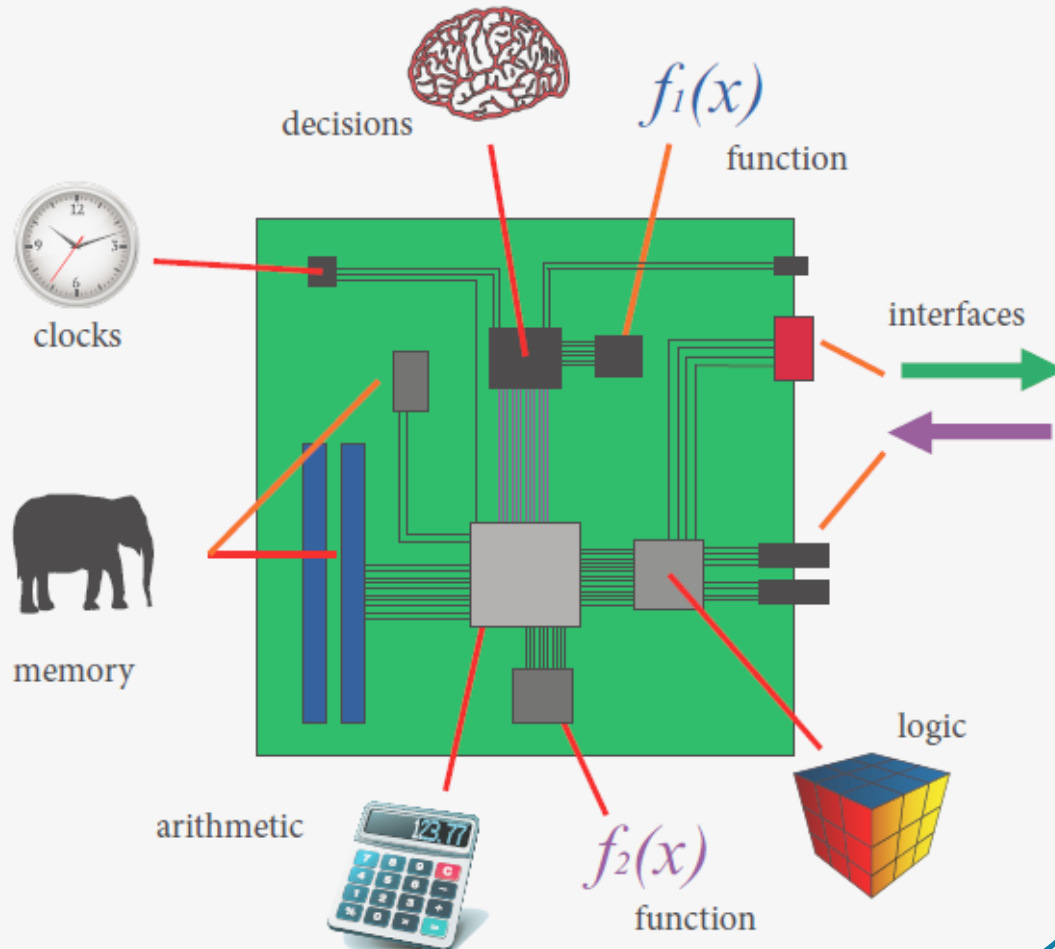
Figure from the "The Zynq Book"

What is an *Embedded System* ?

An **embedded system** is a **complete computing system** that is designed to perform **a specific function or set of functions** within a larger device.

Key Features:

✓ Combines **hardware and software**.

✓ Often runs a **real-time operating system (RTOS)** or bare metal code.

✓ Built for **specific tasks**, not general-purpose computing.

✓ Examples:

    ✓ Microwave control board

    ✓ Car engine control unit (ECU)

Components:

❑ Processor or SoC

❑ Memory (RAM, ROM/Flash)

❑ Input/output interfaces

❑ Software (firmware)

A *SIMPLE* View of an Embedded System
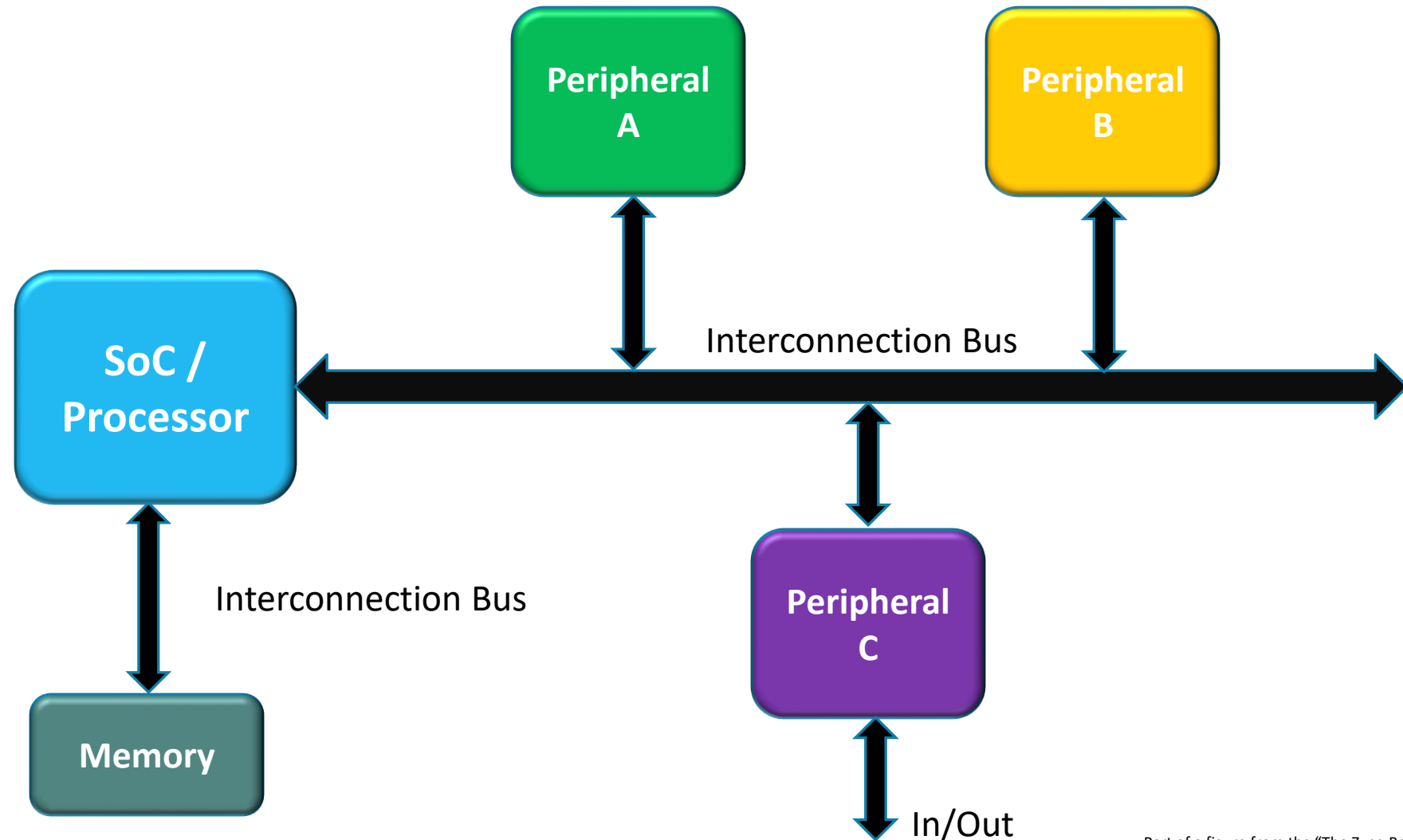
Part of a figure from the "The Zynq Book"

PSoC: Software System-Hardware System

Zynq - AMD PSoC

Applications
Operating System
Hardware Interfacing

Software Stack

Peripherals

Processor

Memory

AXI

Processing System

Peripheral A
Peripheral B

AXI

Peripheral C

Programmable Logic

VHDL/Verilog

Third-Party IPs

HLS
(High Level Synthesis)

HDL Coder
(MatLab)

Core Generator
(Simulink)

System Verilog

Rising the Abstraction Level
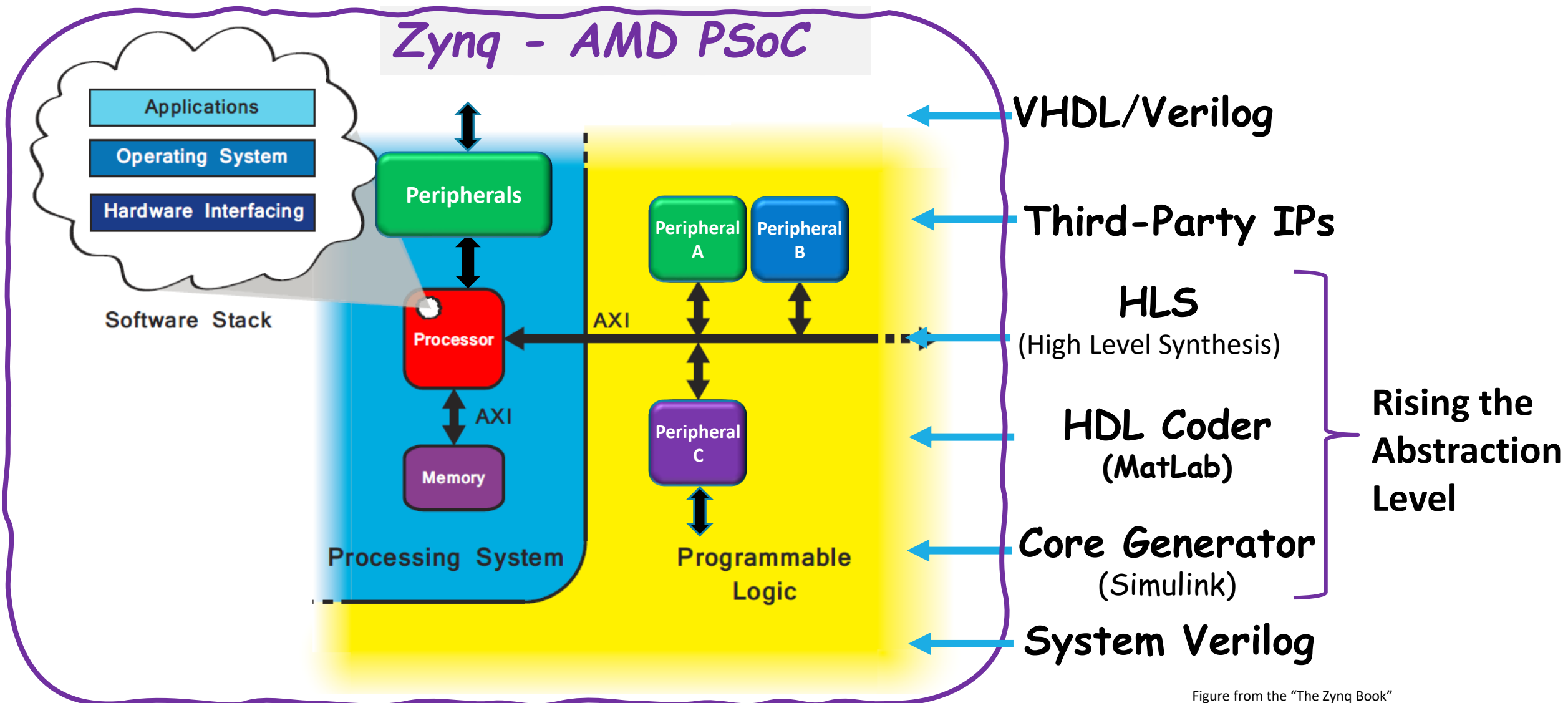
Figure from the "The Zynq Book"

Embedded System

Embedded System

PSoC
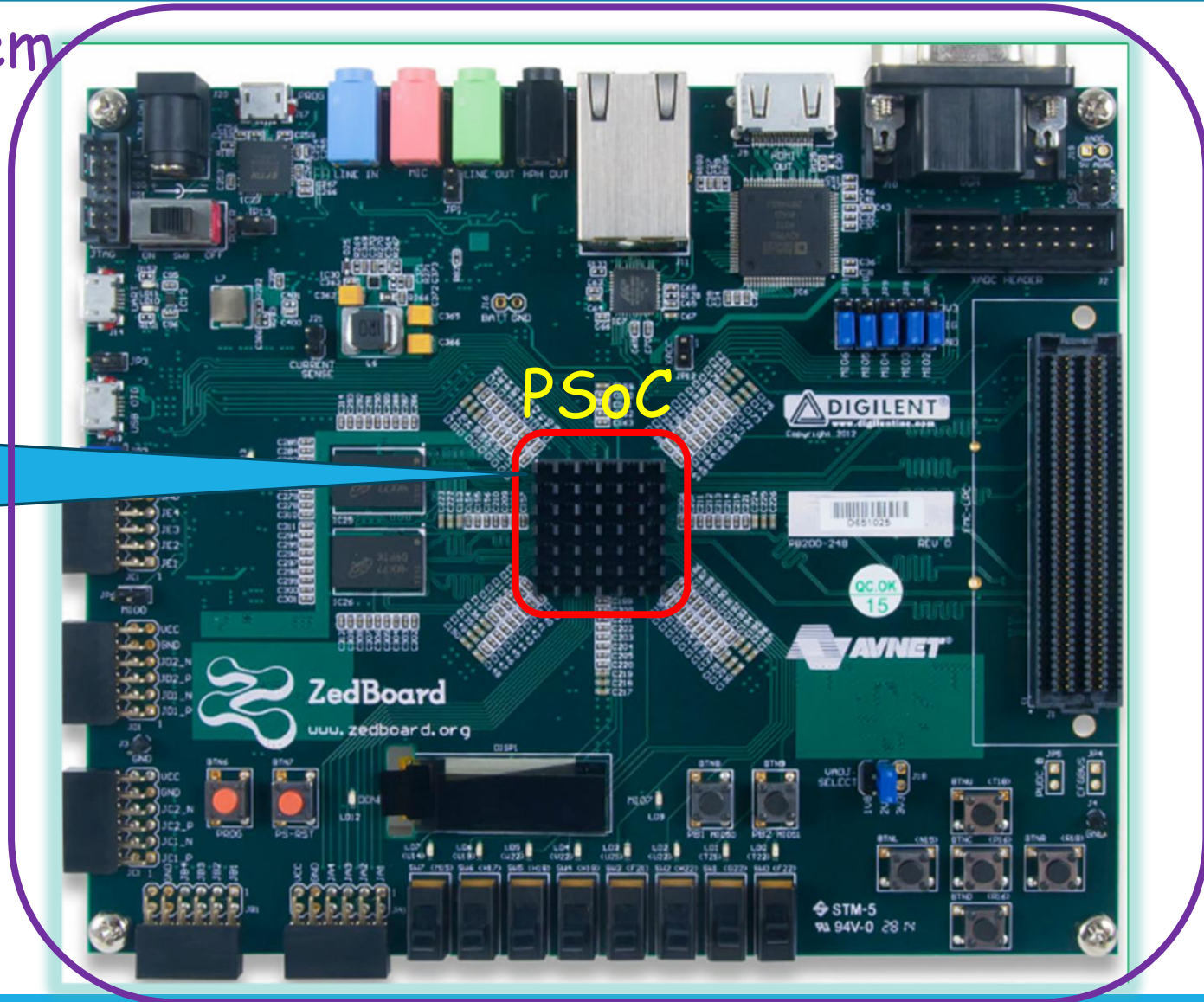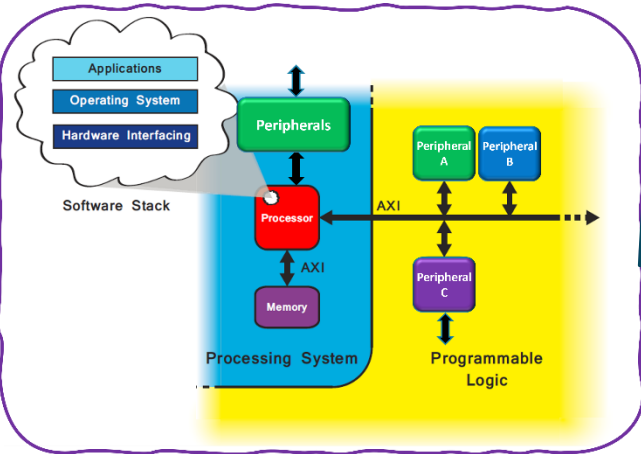
Embedded System & SoC

Feature	Embedded System	System on a Chip (SoC)
Concept	A functional system for a specific task	A chip integrating components of a system
Form	Can include multiple Chips (including a SoC chip)	Usually a single chip
Includes	Software + Hardware	Mainly hardware
Scope	Broader (includes SoC, memories, I/O, etc.	Narrower (hardware platform)
Usage	In cars, phones, medical devices	In phones, tablets, embedded systems

ASIC-SoC vs *PSoC*

ASIC SoC

- Development Time
- Cost
- Lack of flexibility

- Great performance
- Tiny size
- Very large amount of logic
- Power Efficient
- Support analog and mixed signal designs

PSoC

- Great flexibility
- Fast time-to-market
- Upgrade-ability in the field
- Availability of IP cores
- Cheap and easy to use development tools
- Lower performance
- Power hungry

SmartFusion2 (Microchip)

Stratix (Intel)

Zynq/Ultra Scale (Xilinx-AMD)

ASIC SoC – PSoC

ASIC

FPGA + ASIC = PSoC

A Simple View of the *AMD Zynq PSoC*

PS
-Operating System
-Bare Metal (C code)
-Software iterations
-HS memory

PL
-High speed logic
-Parallel processing
-Flexible logic

Zynq

PS

Peripheral

Peripheral

.
.
.

Peripheral

ARM

ARM

AXI

AXI

AXI

Peripheral IP Core

Peripheral IP Core

PL
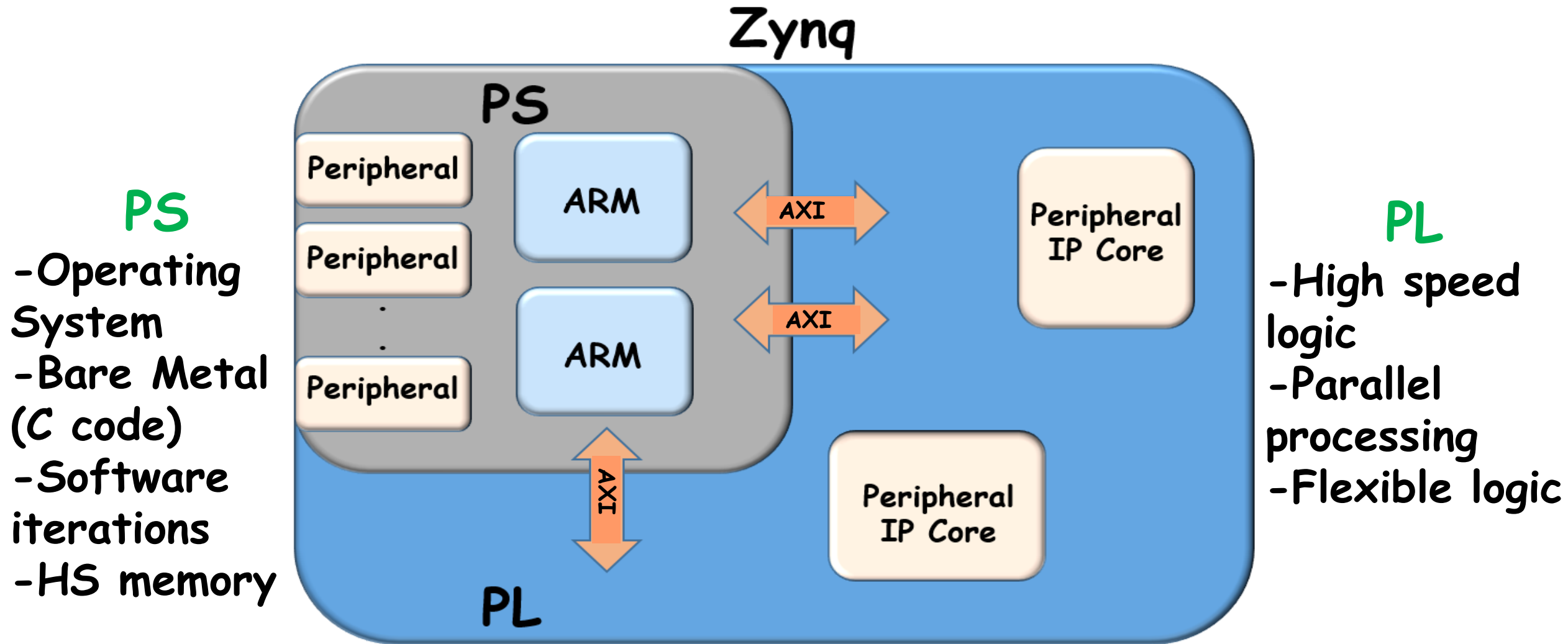
Programmable System on Chip (PSoC) - Zynq

A PSoC family integrates in a single chip the software programmability of an ARM®-based processor with the hardware configurability of an FPGA
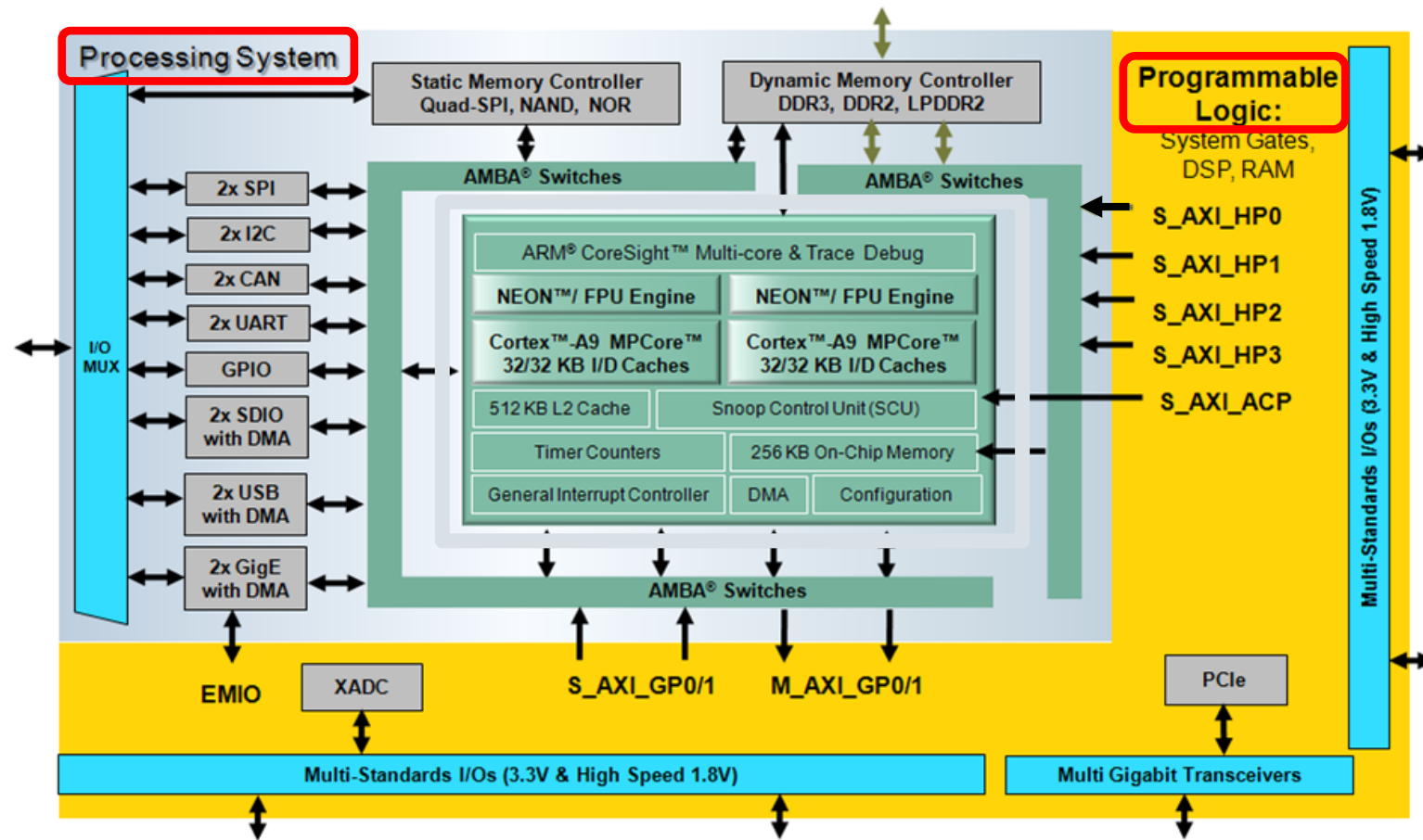
PSoC Design Methodology

- Requirements
- Specifications
- System Design — Software/Hardware Partitioning
- IP Cores
- Software Modules
- Hardware Development & Simulation
- Software Development & Simulation
- Placement & Timing Constraints
- Operating Systems
- System Integration and Debug
- Vivado IP Integrator
- Software Development Vitis
- VIVADO
- XILINX VITIS

Hardware and Software Layers in a SoPC

.elf

Software Stack
(Vitis)

.bit

Custom
Hardware-FPGA
(Vivado)

Software Applications

Operating System

Board Support Package

Hardware Base System

Export
to
Vitis

Part of figure from the "The Zynq Book"

Hardware and Software Layers in a SoPC

Software Stack (Vitis)

Custom app, 'C' code to access the Hw

OS (ex. FreeRTOS) or Bare Metal

BSP: set the lower-level drivers and functions to access the Hw from the Sw

Hardware Configuration designed in **Vivado IP Integrator**

Software Applications

Operating System

Board Support Package

Hardware Base System

Export to Vitis

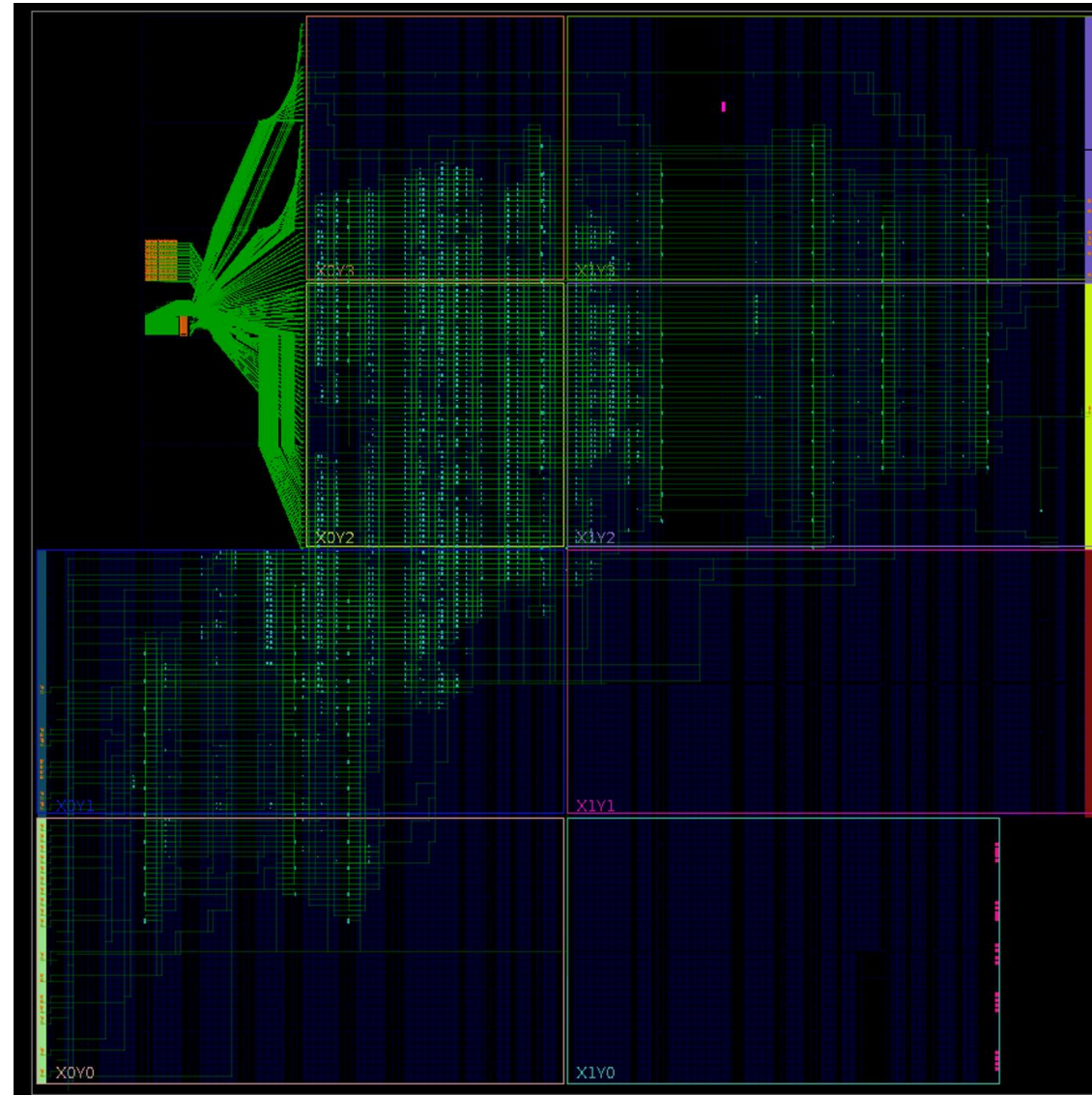**Hardware Design (Vivado)**
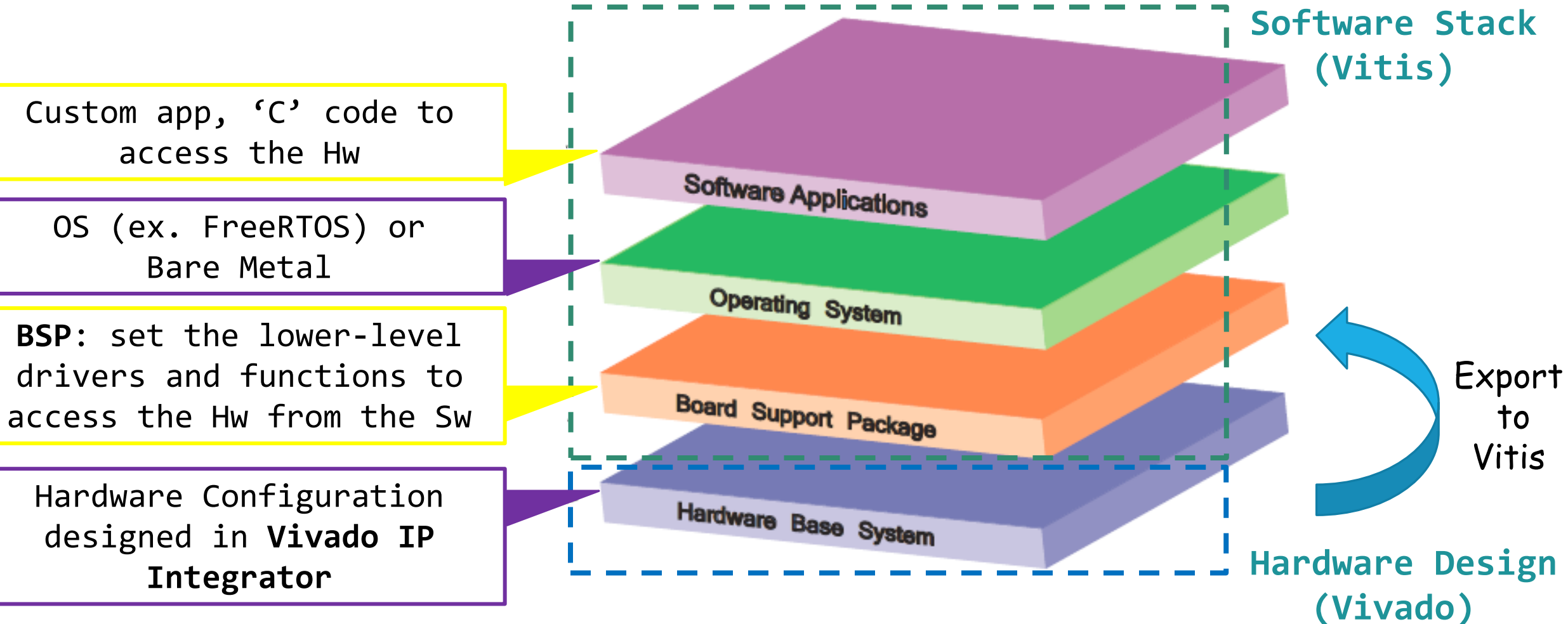
Part of figure from the "The Zynq Book"

Zynq Block Design in Vivado – PS + PL
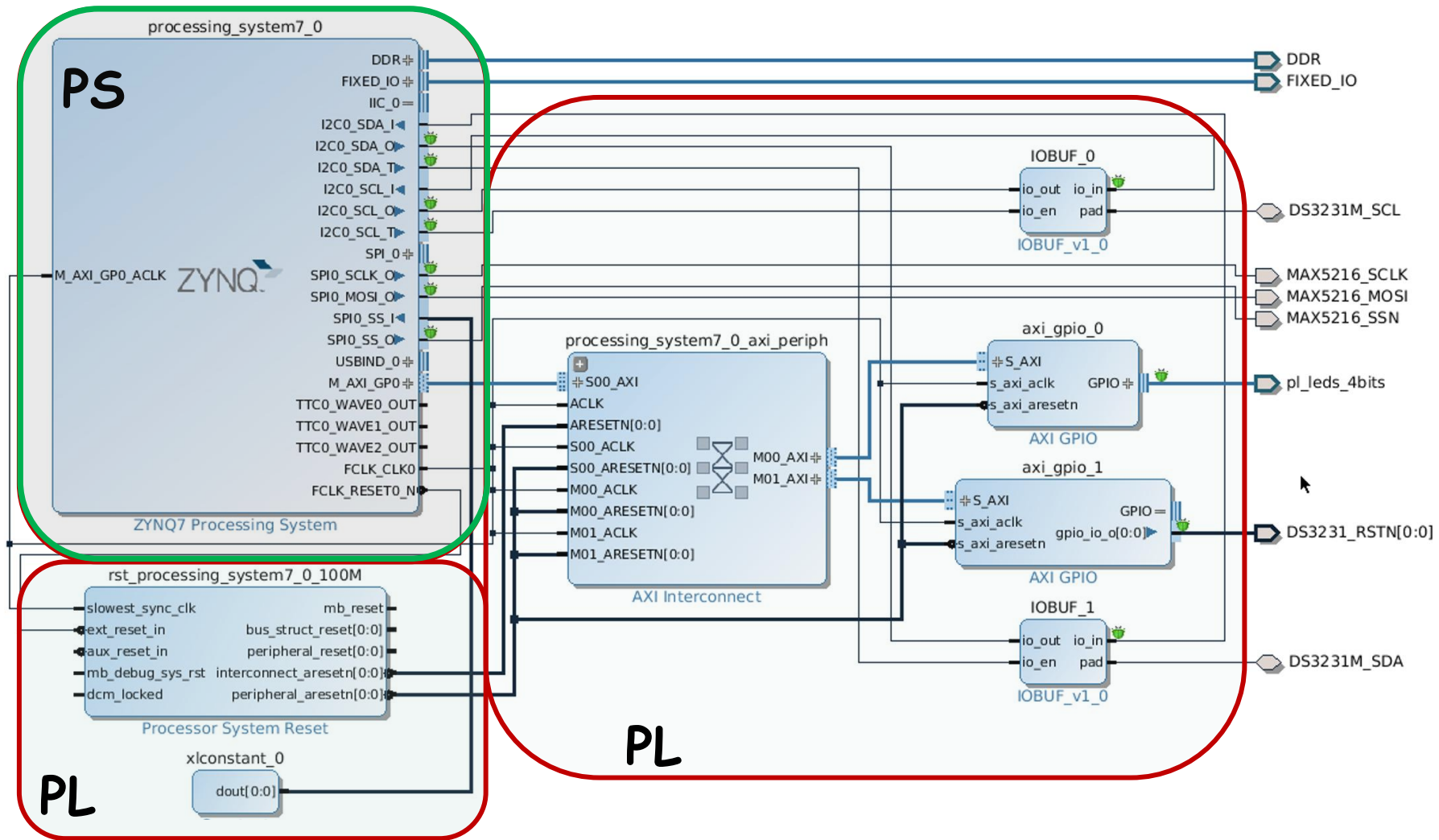
Internal Zynq View – System Placed & Routed

FPGAs Soft Processors

Sources:

o HDL Code

o Netlist

o Placed & Routed netlist

Common Soft-processors:

o RISC-V

o LEON

o 8051

o OpenRISC

When to use it:

o Cost-sensitive applications

o App where the processor is just a support

o Processor configurability and upgradeability

FPGA Hard Processors (SoPC)

√ **Xilinx**:

 √ ARM Cortex A9-dual in Zynq devices

 √ Cortex A-53 in MPSoC (Zynq Uktrascale+)

√ **Intel-Altera**

 √ Cortex A9 dual core in Arria V SoC and Cyclone V SoC

 √ Cortex A53 in Stratix 10 SoC using 14nm Intel process

√ **Microchip (former Actel)**

 √ Smart Fusion uses a Cortex M3 (at 100Mhz) and programmable analog

FPGAs vs Processors

FPGA	Processor
Perform multiple instruction at once. Execution is done in parallel/concurrently. Hence, minimize the latency and maximizes the throughput	Performs only one instruction at a time, because the execution is sequential.
Provides ultra-high memory bandwidth. Dedicated DDR memory blocks for Rd/Wr.	Limited memory bandwidth.
Provides constant latency for each iteration.	Latency depends on the operating system load, and sometimes, on the compilation options.
A microcontroller or a microprocessor can be implemented within an FPGA.	It is not possible to implement an FPGA in a processor.
Very high data processing throughput.	Lower data processing throughput.
It could be expensive.	Usually is cheap.
User-configurable logic, dedicated DSP blocks.	Fixed arithmetic engines.
User configurable I/O ports – Multiple I/O standards.	Fixed, dedicated I/O ports.
Compute intensive algorithms. Massive parallel operations. High data rate computation.	Decision making. Complex Analysis. Block-oriented tasks.

Zynq SoPC *ZedBoard* – Board to be used in the Labs