Del Algoritmo al Hardware:
Aprendizaje Automático en Sistemas Embebidos

From Algorithm to Hardware: Machine Learning in Embedded Systems

1 al 11 de Abril, 2025. Universidad Nacional de Mar del Plata - Mar del Plata - Argentina.

System-On-Chip based on FPGA: Architecture and workflow

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -

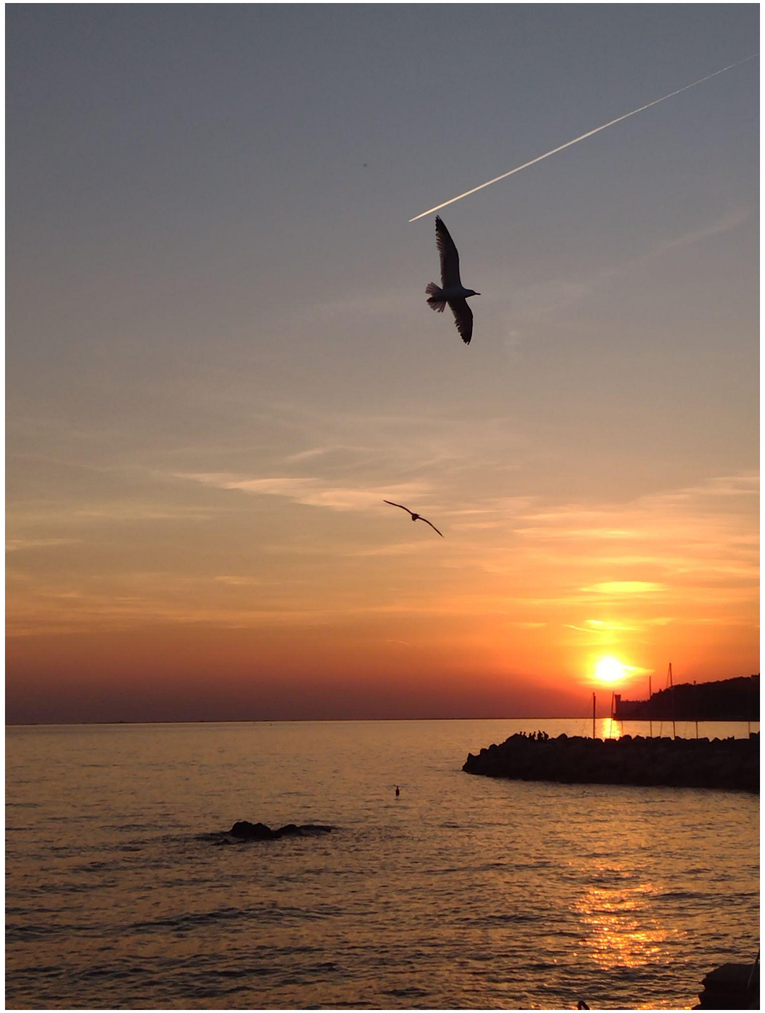
Universidad Nacional de Mar del Plata

Fundación Williams

ICTP The Abdus Salam International Centre for Theoretical Physics

Outline

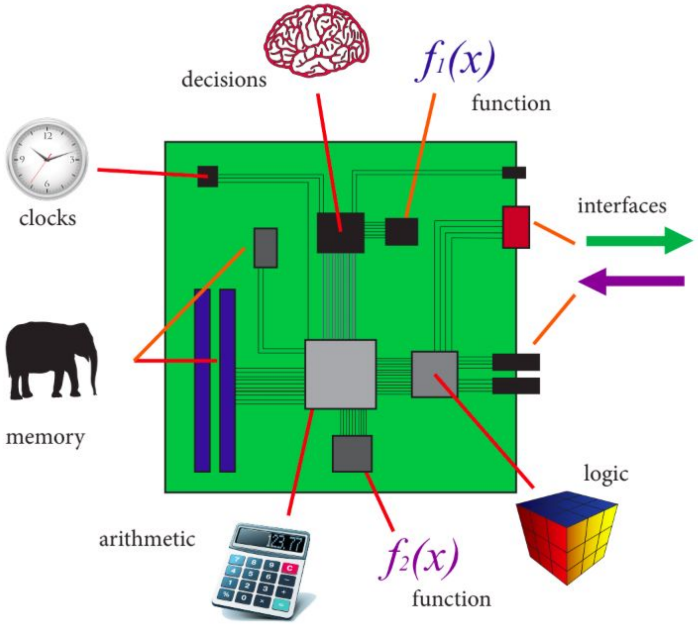
- Introduction
- Architecture of an SoC FPGA.
- Development Workflow in SoC FPGA.
- Development Tools - IP Integrator.
- Demo: Binary classifier - Vivado IP Integrator and Vitis.



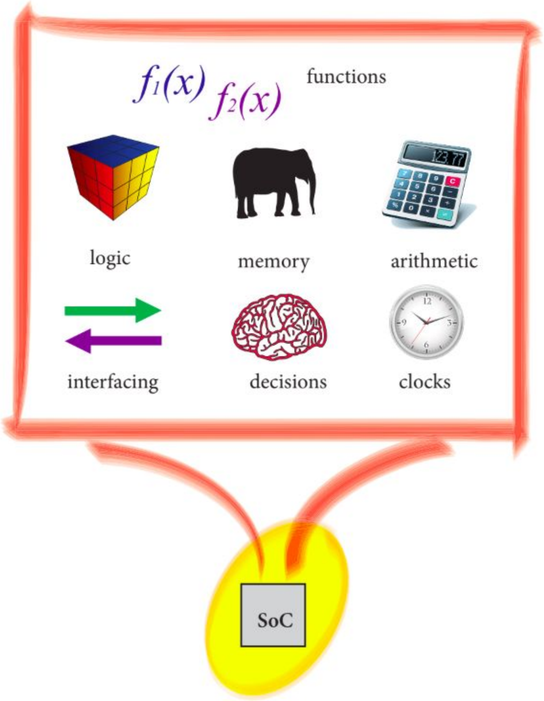
Introduction

Introduction

- What is an SoC?



System-on-a-board



System-on-chip

Image from The Zynq book http://www.zynqbook.com/.

Introduction

- The hardware system architecture of an embedded SoC

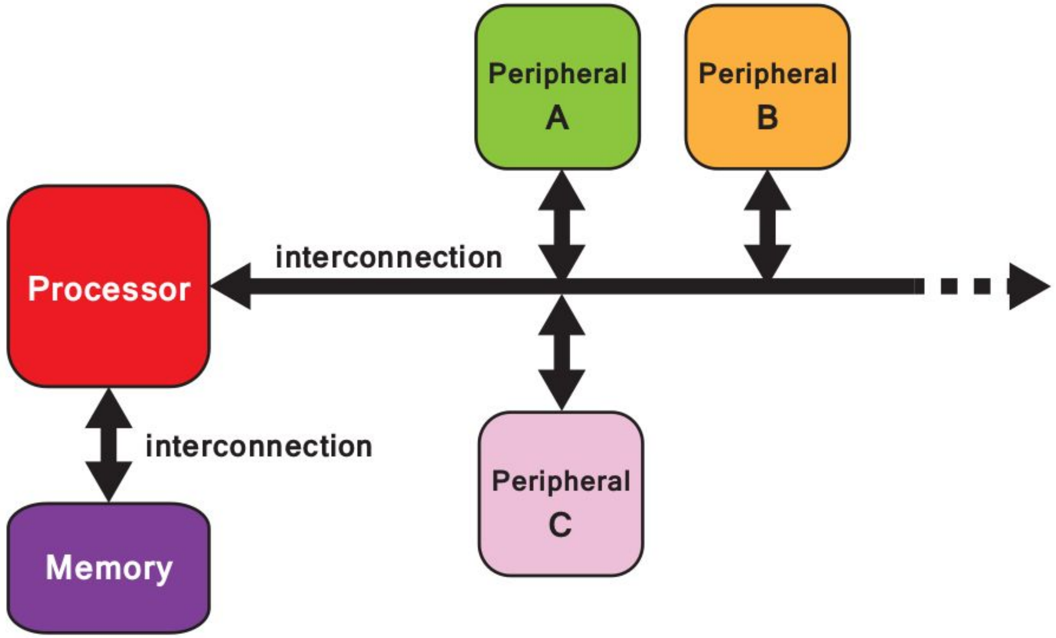


Image from The Zynq book http://www.zynqbook.com/.

Introduction

- SoC FPGAs combine programmable logic (FPGA) with high-performance processors in a single chip.

Introduction

- SoC FPGAs combine programmable logic (FPGA) with high-performance processors in a single chip.
- They offer flexibility, efficiency, and real-time processing capabilities.

Introduction

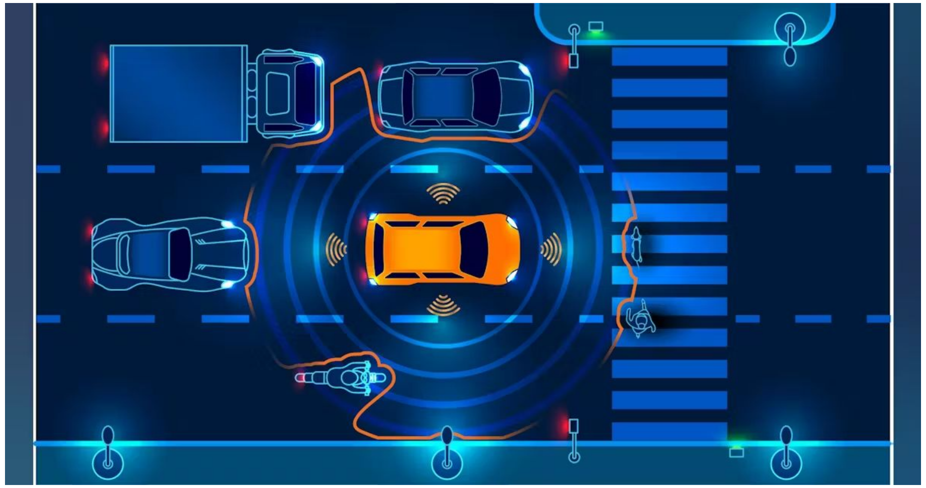
- SoC FPGAs combine programmable logic (FPGA) with high-performance processors in a single chip.

- They offer flexibility, efficiency, and real-time processing capabilities.

- Bridges the gap between flexibility of CPUs and performance of dedicated hardware.
  - The CPU handles general-purpose tasks.
  - The FPGA accelerates specific tasks.
  - The FPGA can be reconfigured, making it more **flexible** than an ASIC but **faster** than a CPU.

Introduction

- Typical industry applications:
  - Automotive.
  - Telecommunications.
  - Robotics.
  - Industrial Automation.
  - Aerospace.
  - Defense.
  - Medical.
  - AI.

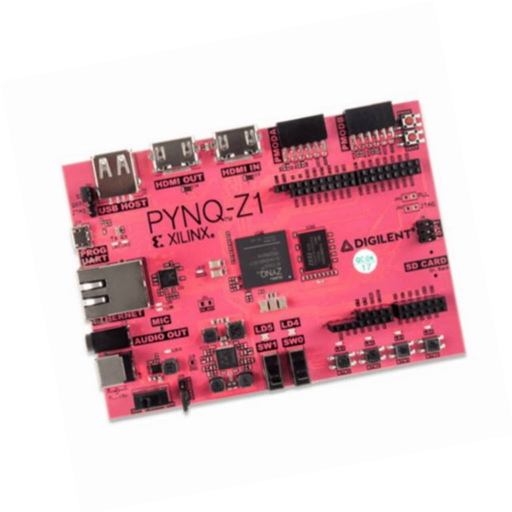


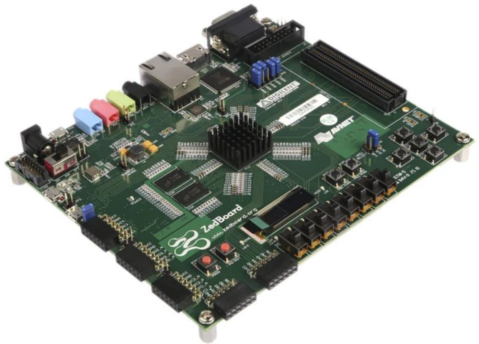
https://www.electronicdesign.com/technologies/embedded/digital-ics/fpga/article/55240420/lattice-semiconductor-automotive-and-functional-safety-fpgas-offer-programmable-protection

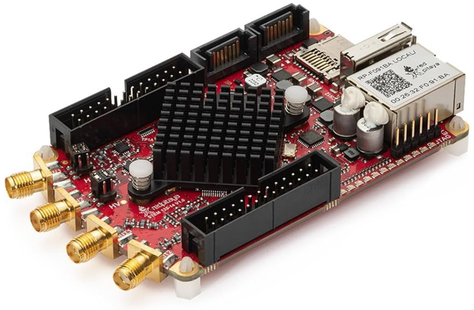
Architecture of an SoC FPGA

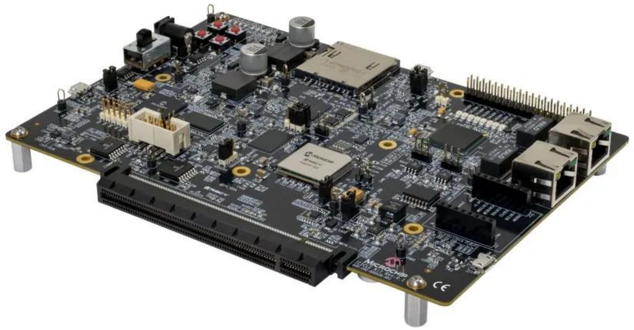
Architecture of an SoC FPGA

In this section: SoC FPGA architecture, its main components, how they interact, and why their integration is crucial for embedded applications.









Architecture of an SoC FPGA

Overview of an SoC

- An SoC FPGA is a device that combines two key elements in a single chip:

Architecture of an SoC FPGA

Overview of an SoC

- An SoC FPGA is a device that combines two key elements in a single chip:
    - An **embedded processor** (CPU), typically based on ARM (Cortex-A9, Cortex-R5, Cortex-A53, etc.).

Architecture of an SoC FPGA

Overview of an SoC

- An SoC FPGA is a device that combines two key elements in a single chip:
    - An **embedded processor** (CPU), typically based on ARM (Cortex-A9, Cortex-R5, Cortex-A53, etc.).
    - A **programmable logic fabric** (FPGA), allowing for customized hardware implementations.

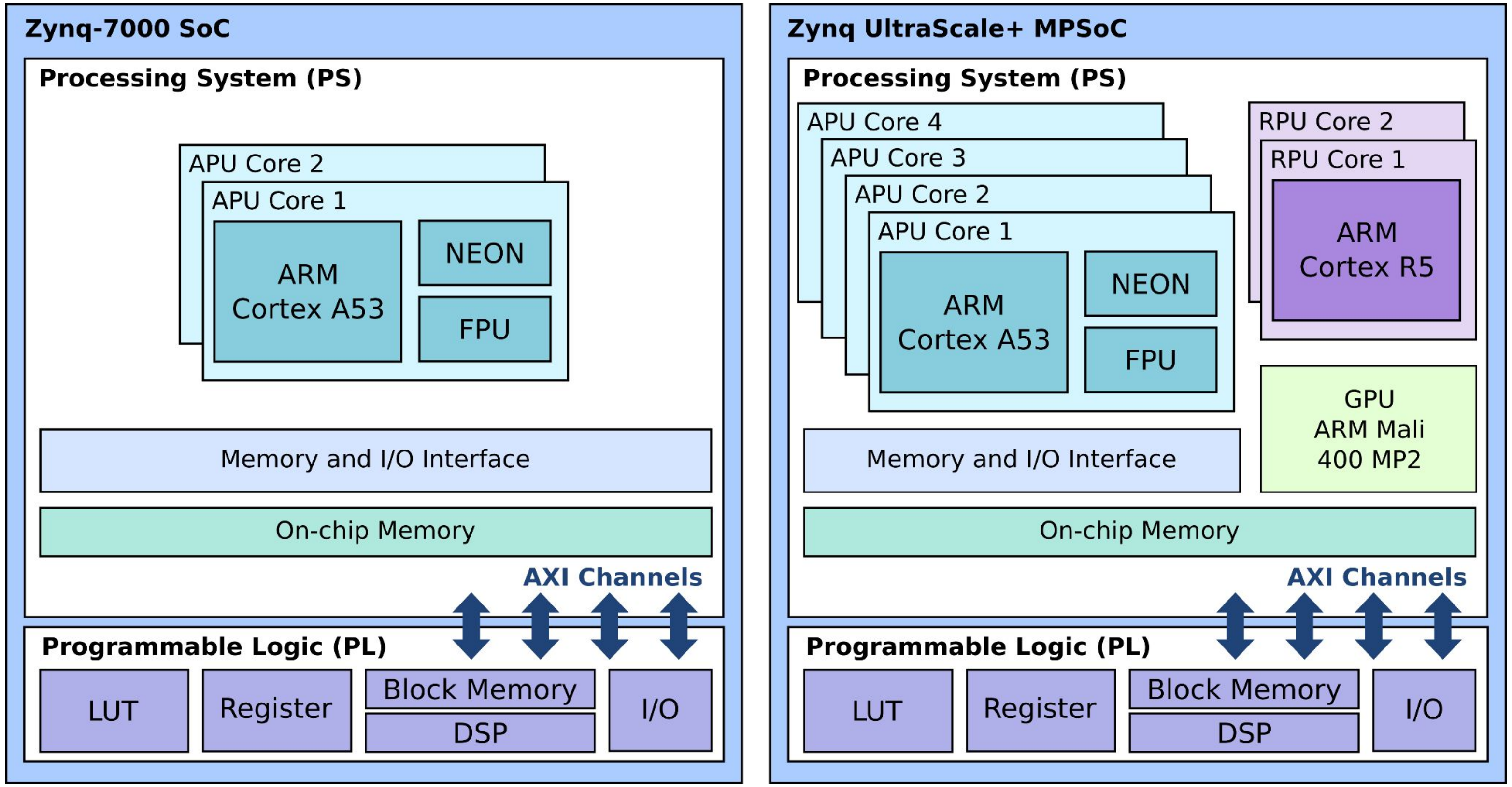
Architecture of an SoC FPGA

Overview of an SoC

- An SoC FPGA is a device that combines two key elements in a single chip:
  - An **embedded processor** (CPU), typically based on ARM (Cortex-A9, Cortex-R5, Cortex-A53, etc.).
  - A **programmable logic fabric** (FPGA), allowing for customized hardware implementations.
- An **interconnect system** (AXI - Advanced eXtensible Interface) enables efficient communication between the processor and the FPGA.

Architecture of an SoC FPGA

Overview of an SoC



Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Embedded processor**
  - Based on ARM
  - Runs embedded Linux, FreeRTOS, or bare-metal applications.
  - Interacts with the FPGA side.

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Embedded processor**

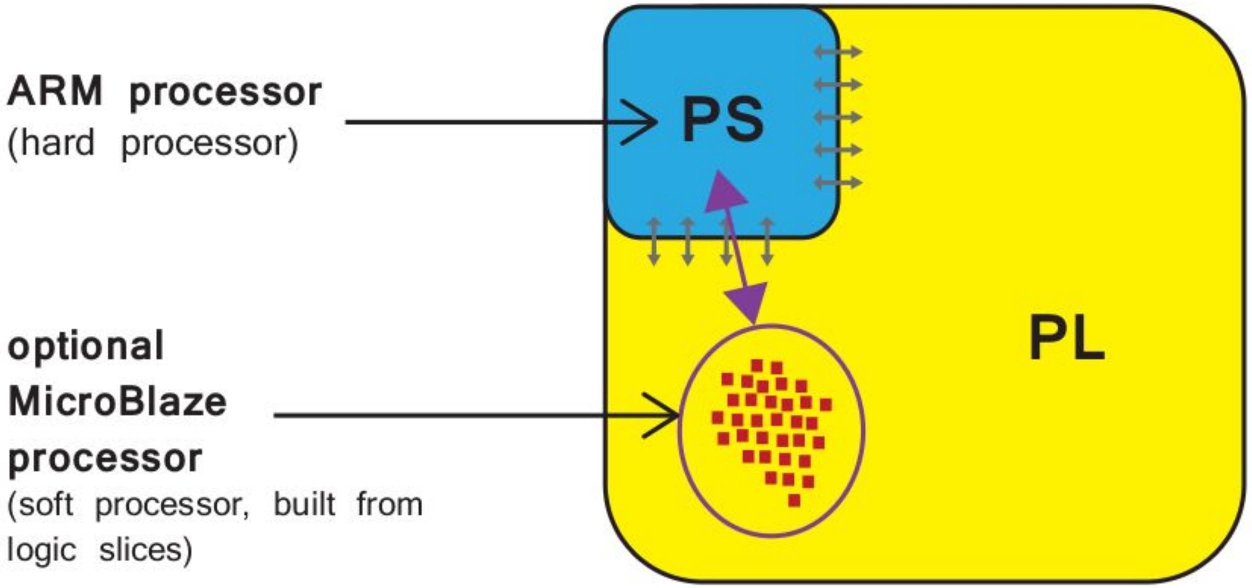


Image from The Zynq book http://www.zynqbook.com/.

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Embedded processor**

“Zynq processing system encompasses not just the ARM processor, but a set of associated processing resources forming an Application Processing Unit (APU), and further peripheral interfaces, cache memory, memory interfaces, interconnect, and clock generation circuitry”

The Zynq book http://www.zynqbook.com/.

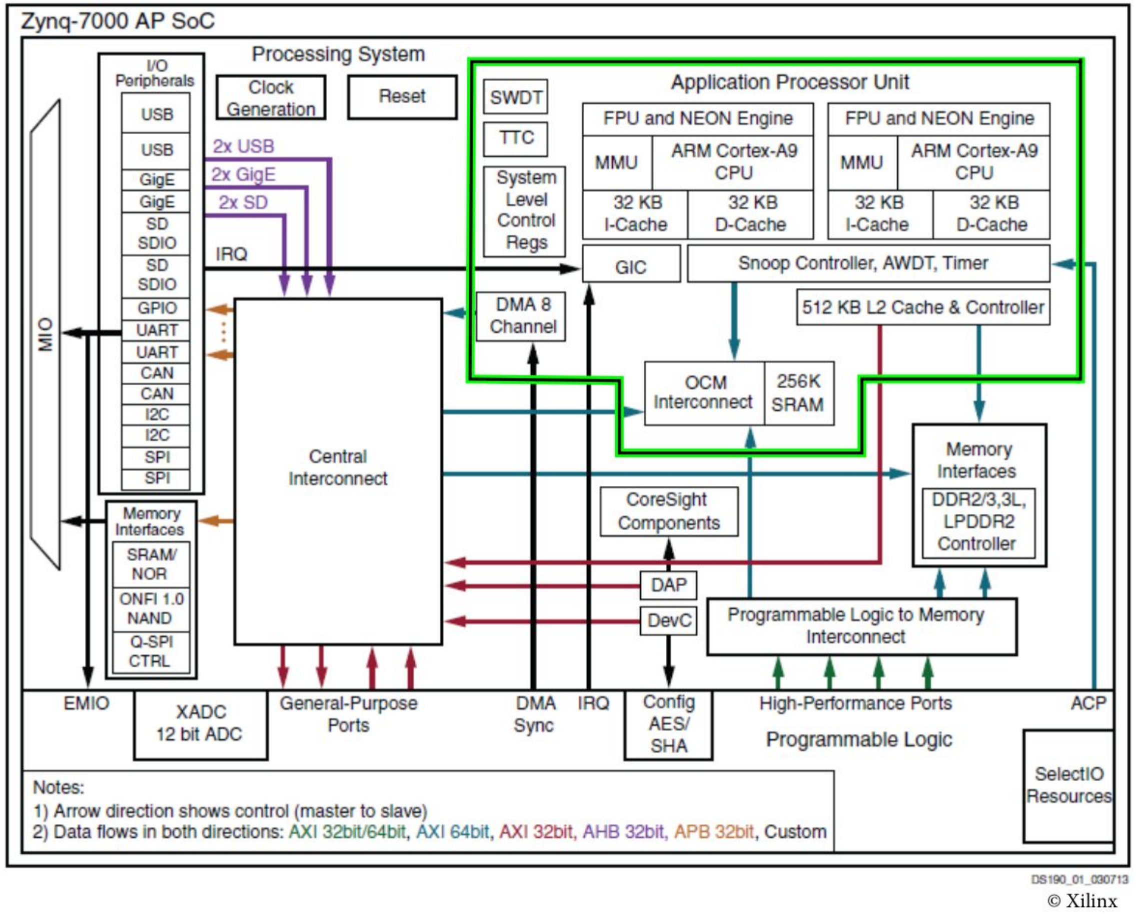


Image from The Zynq book http://www.zynqbook.com/.

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Programmable Logic**
  - Hardware accelerators, parallel processing, DSPs, and custom interfaces.
  - VHDL/Verilog or high-level tools.
  - High-performance and parallelism.

Architecture of an SoC FPGA

Main Components of an SoC FPGA

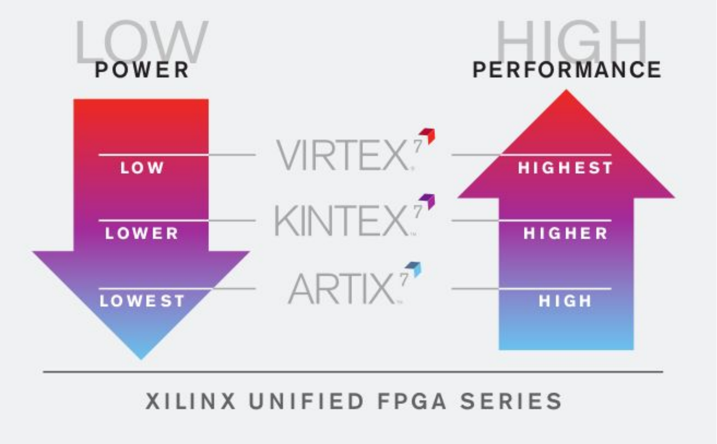
- **Programmable Logic**
  - Zynq shares the same 7 series programmable logic as
    - Artix™-based devices: Z-7010, Z-7015, and Z-7020 (high-range I/O banks only)
    - Kintex™-based devices: Z-7030, Z-7035, Z-7045, and Z-7100 (mix of high-range and high-performance I/O banks)

Architecture of an SoC FPGA

7 SERIES FPGA FAMILY COMPARISON

MAXIMUM CAPABILITY	ARTIX-7 FPGAS	KINTEX-7 FPGAS	VIRTEX-7 FPGAS
Logic Cells	215K	478K	1,955K
Block RAM	13Mb	34Mb	68Mb
DSP Slices	740	1,920	3,600
Peak DSP Performance (symmetric FIR)	930 GMACS	2,845 GMACS	5,335 GMACS
Transceiver Count	16	32	96
Peak Transceiver Speed	6.6 Gbps	12.5 Gbps	13.1 / 28.05 Gbps
Peak Serial Bandwidth (full duplex)	211Gbps	800Gbps	2,784 Gbps
PCI Express® Interface	x4 Gen2	x8 Gen2	x8 Gen2 / x8 Gen3*
Memory Interface	1,066 Mbps	1,866 Mbps	1,866 Mbps
I/O Pins	500	500	1,200
I/O Voltage	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V	1.2V, 1.35V, 1.5V, 1.8V, 2.5V, 3.3V*
Packaging Options	Low-cost wire bond	Low-cost lidless flip-chip and high-performance flip-chip	Highest performance flip-chip; stacked silicon interconnect technology based
Target Application Examples	• Portable/handheld ultrasound • 3D cameras and camcorders • D-SLR still cameras • Software defined radio • 3D TV • Portable eReaders • Automotive Infotainment • Multifunction printers • Video surveillance	• Wireless LTE infrastructure • 10G PON OLT line card • LED backlit and 3D video displays • Video-over-IP bridge • Cellular radio • Medical Imaging • Avionics imaging • Set top boxes • Motor control	• 400G and 100G line cards • 300G Interlaken bridge • Terabit switch fabric • 100G OTN • MUXPONDER • RADAR • ASIC emulation • High-performance computing • Test and measurement

*Refer to the 7 Series Product Overview for device details such as soft vs. hard Gen3 interface, and> 2.5V/3.3V support.



Source: https://www.xilinx.com/publications/prod_mktg/7-Series-Product-Brief.pdf

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **CPU-FPGA Interconnect**
    - The processor and FPGA communicate through **high-speed buses**, mainly AXI.
    - AXI stands for Advanced eXtensible Interface.

    - Interconnects:
        - AXI4: Memory-mapped links. Large memory transfers - Data burst transfer of up to 256 data words.
        - AXI4-Lite: Memory-mapped. Low-bandwidth, used for control registers.
        - AXI4-Stream: For high-speed streaming data.

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Memory storage**
  - DDR: Shared memory between CPU and FPGA.
  - BRAM (Block RAM): Fast internal FPGA memory for quick access.
  - Linux-based SoC FPGA:
    - the OS runs in DDR.
    - data processing core in the FPGA uses BRAM as a high-speed cache.

Architecture of an SoC FPGA

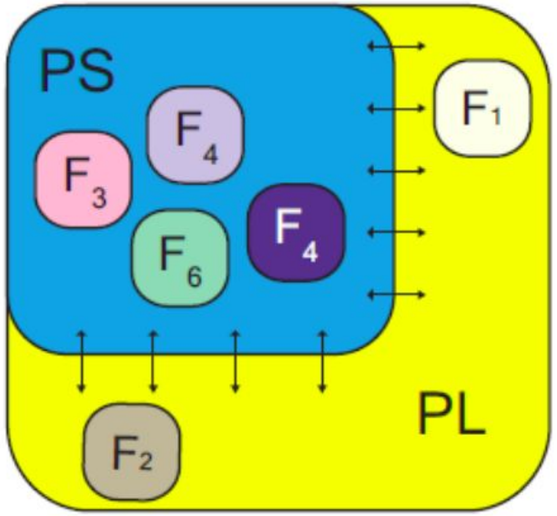
Main Components of an SoC FPGA

- **Peripherals and I/O**
  - GPIOs.
  - Networking: Ethernet.
  - Communication interfaces: UART, SPI, I2C, CAN, USB, PCIe.

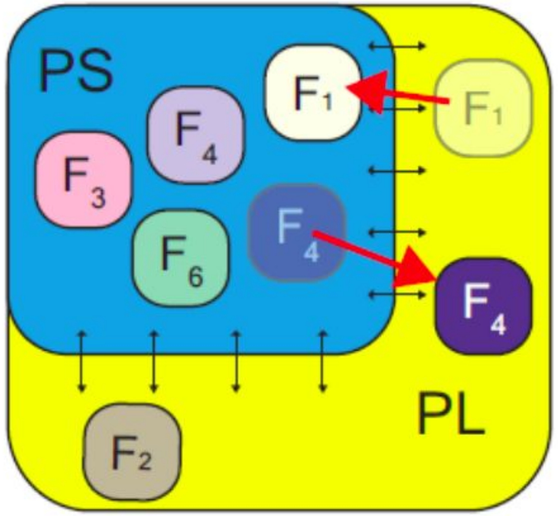
Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Hardware/Software Co-Design**



HW/SW partitioning #1



HW/SW partitioning #2
(F_4 moved to PL, F_1 moved to PS)

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Hardware/Software Co-Design**
  - It aims to exploit the inherent features of different technologies, deciding **which part of the algorithm should be implemented with sequential instructions and which part in the hardware.**

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Hardware/Software Co-Design**
    - It aims to exploit the inherent features of different technologies, deciding **which part of the algorithm should be implemented with sequential instructions and which part in the hardware.**
    - Usually, a **profiling** of the algorithm helps to determine which part is suitable to accelerate. Typically, the most expensive section of the code, in terms of runtime, is good candidate for hardware acceleration.

Architecture of an SoC FPGA

Main Components of an SoC FPGA

- **Hardware/Software Co-Design**

    - It aims to exploit the inherent features of different technologies, deciding **which part of the algorithm should be implemented with sequential instructions and which part in the hardware.**

    - Usually, a **profiling** of the algorithm helps to determine which part is suitable to accelerate. Typically, the most expensive section of the code, in terms of runtime, is good candidate for hardware acceleration.

    - Regarding **communication overhead**, its complexity should be minimized between both technologies (that is, between the processor and the FPGA).

Architecture of an SoC FPGA

Overview of an SoC FPGA

- The AMD Zynq™ 7000 SoC processing system (PS) integrated with a highly flexible and high-performance programmable logic (PL) section, all on a single system-on-a-chip (SoC).

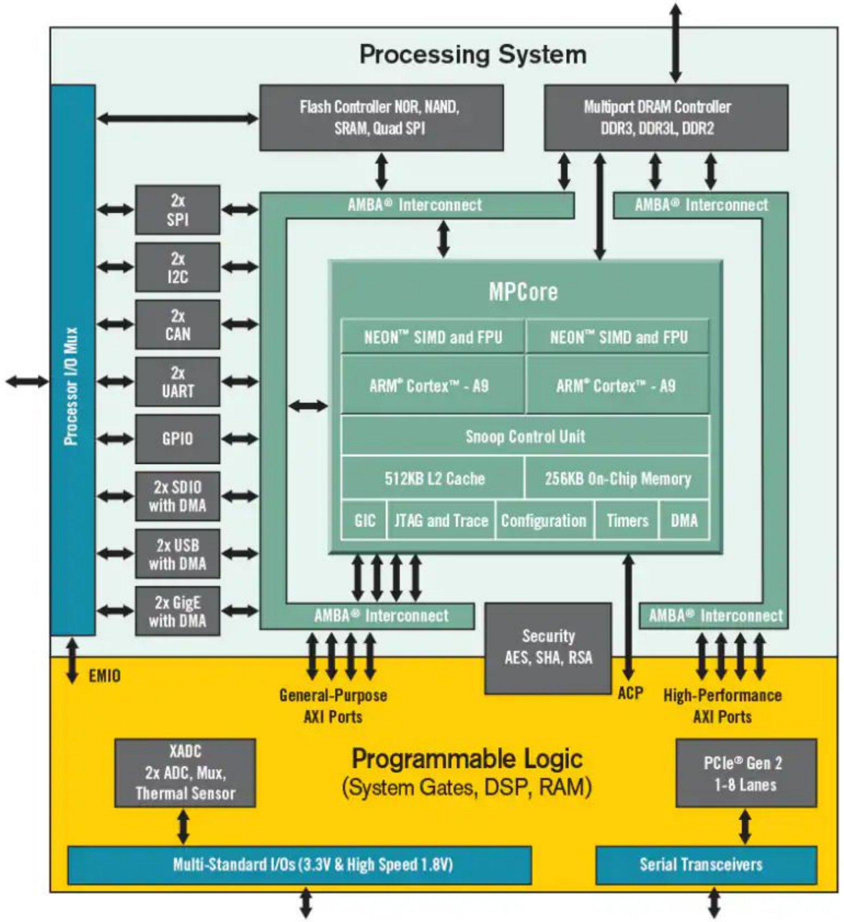


Image from https://www.mouser.co.uk/new/xilinx/xilinx-zynq-7000-socs/

Architecture of an SoC FPGA

Overview of an SoC FPGA

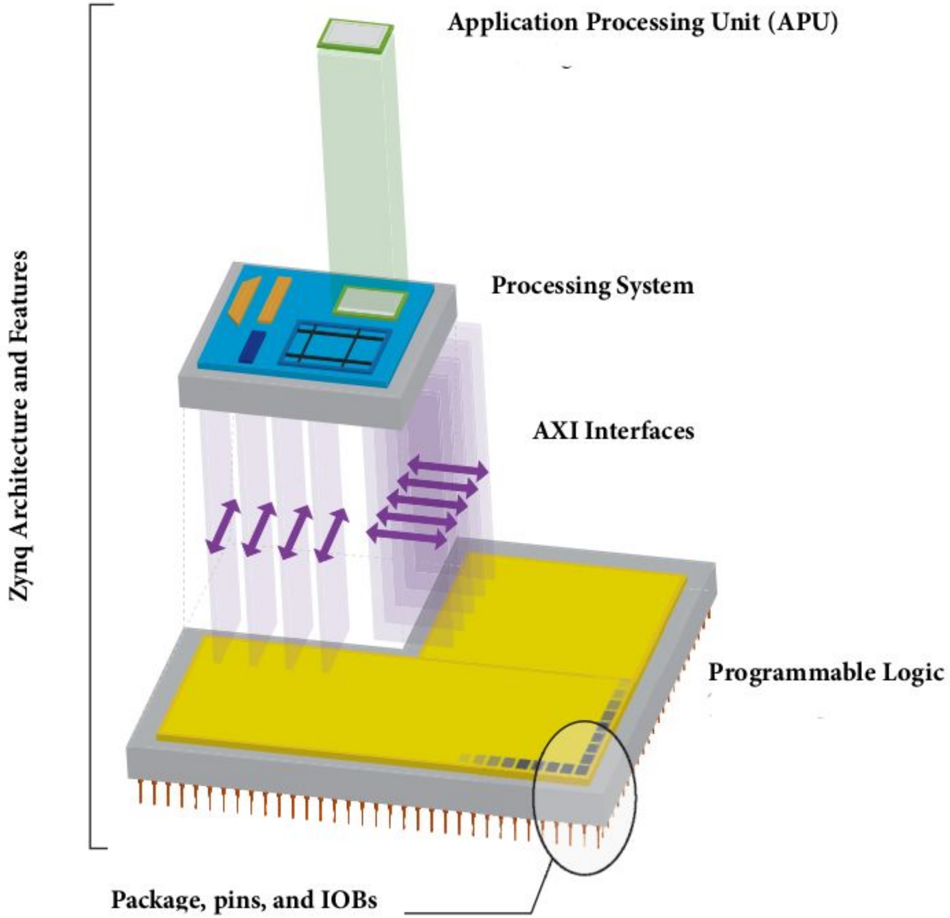


Image from The Zynq book http://www.zynqbook.com/.

Architecture of an SoC FPGA

Overview of an SoC FPGA

“**Embedded systems** that combine a **processing unit with FPGA-based designs** can be inherently complex.”

Source: Zynq-7000 SoC Embedded Design Tutorial (UG1165)

Architecture of an SoC FPGA

Overview of an SoC FPGA

“**Embedded systems** that combine a **processing unit with FPGA-based designs** can be inherently complex.”

“Both the hardware and software components are substantial projects on their own, each requiring development and optimization. “

Source: Zynq-7000 SoC Embedded Design Tutorial (UG1165)

Architecture of an SoC FPGA

Overview of an SoC FPGA

“Embedded systems that combine a **processing unit with FPGA-based designs** can be inherently complex.”

“Both the hardware and software components are substantial projects on their own, each requiring development and optimization. “

“Integrating these two elements into a cohesive, fully functional system introduces **additional challenges, including synchronization, communication, and performance optimization**.”

Source: Zynq-7000 SoC Embedded Design Tutorial (UG1165)

Development Workflow in SoC FPGA

Development Workflow in SoC FPGA

- **Objective of this section:**
  - Explain the **development workflow from design to deployment**.

Development Workflow in SoC FPGA

- **Key Stages:**
  - Define Problem and Select Hardware.
  - Hardware Design (FPGA).
  - Software Development (Embedded CPU).
  - Hardware-Software Integration.
  - Testing and Debugging.
  - Implementation and Deployment.

Development Workflow in SoC FPGA

- **Defining the problem and selecting hardware**
  - Processing needs (CPU vs FPGA workload balance).
  - Peripheral interfaces (USB, PCIe, Ethernet, etc.).
  - Latency and real-time constraints.
  - Power consumption and efficiency.

Development Workflow in SoC FPGA

- **Hardware Design (FPGA)**
  - Develop logic using VHDL/Verilog or HLS.
  - Use IP cores and predefined blocks.
  - Simulate and verify FPGA design.

Development Workflow in SoC FPGA

- **Software Development (Embedded CPU)**
    - Select an OS (Linux, FreeRTOS) or Bare-Metal.
    - Develop device drivers for FPGA peripherals.
    - Write applications and firmware for the CPU.

Development Workflow in SoC FPGA

- **Hardware-Software Integration**
    - Ensure CPU-FPGA communication.
    - Interrupt handling for real-time processing.

Development Workflow in SoC FPGA

- **Testing & Debugging**
  - Simulation.
  - Hardware Debugging (JTAG, ILA).
  - Profiling and Performance Optimization.

Development Workflow in SoC FPGA

- **Implementation and Deployment**
  - Reconfigure the FPGA.
  - Load embedded application into the CPU.
  - Deploy on target hardware.
  - Iterative process.

Development Workflow in SoC FPGA

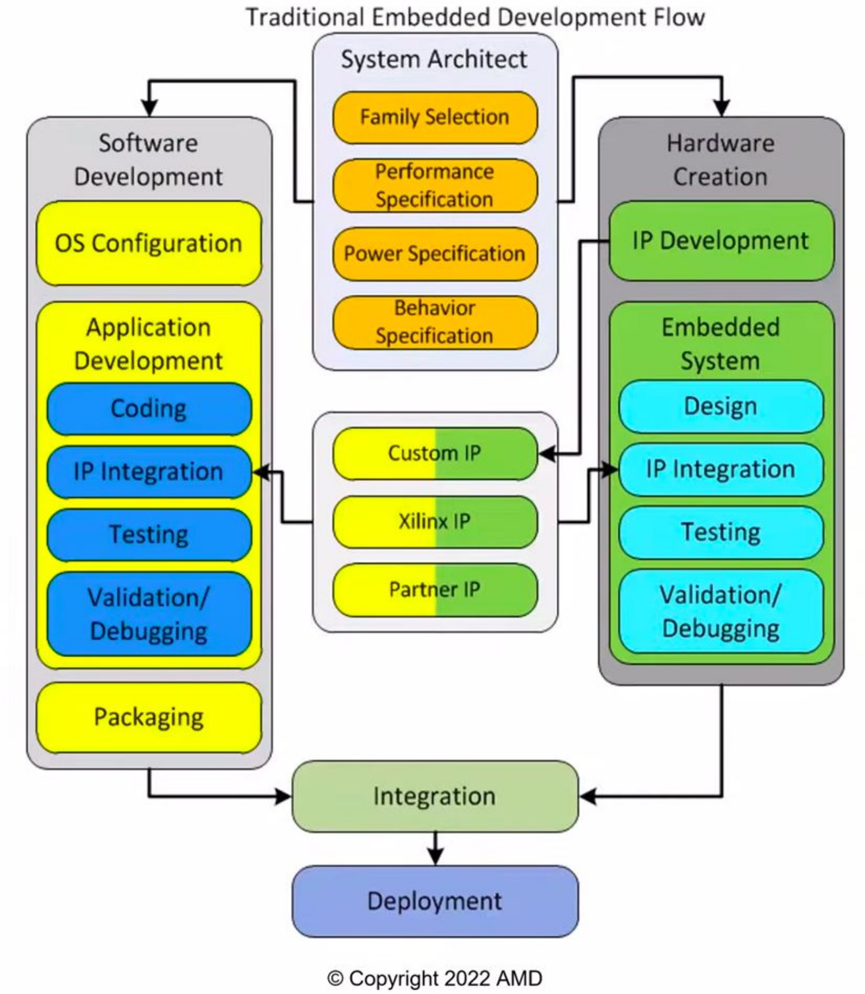


Image from
https://www.mouser.co.uk/new/xilinx/xilinx-zynq-7000-socs/

Development Workflow in SoC FPGA

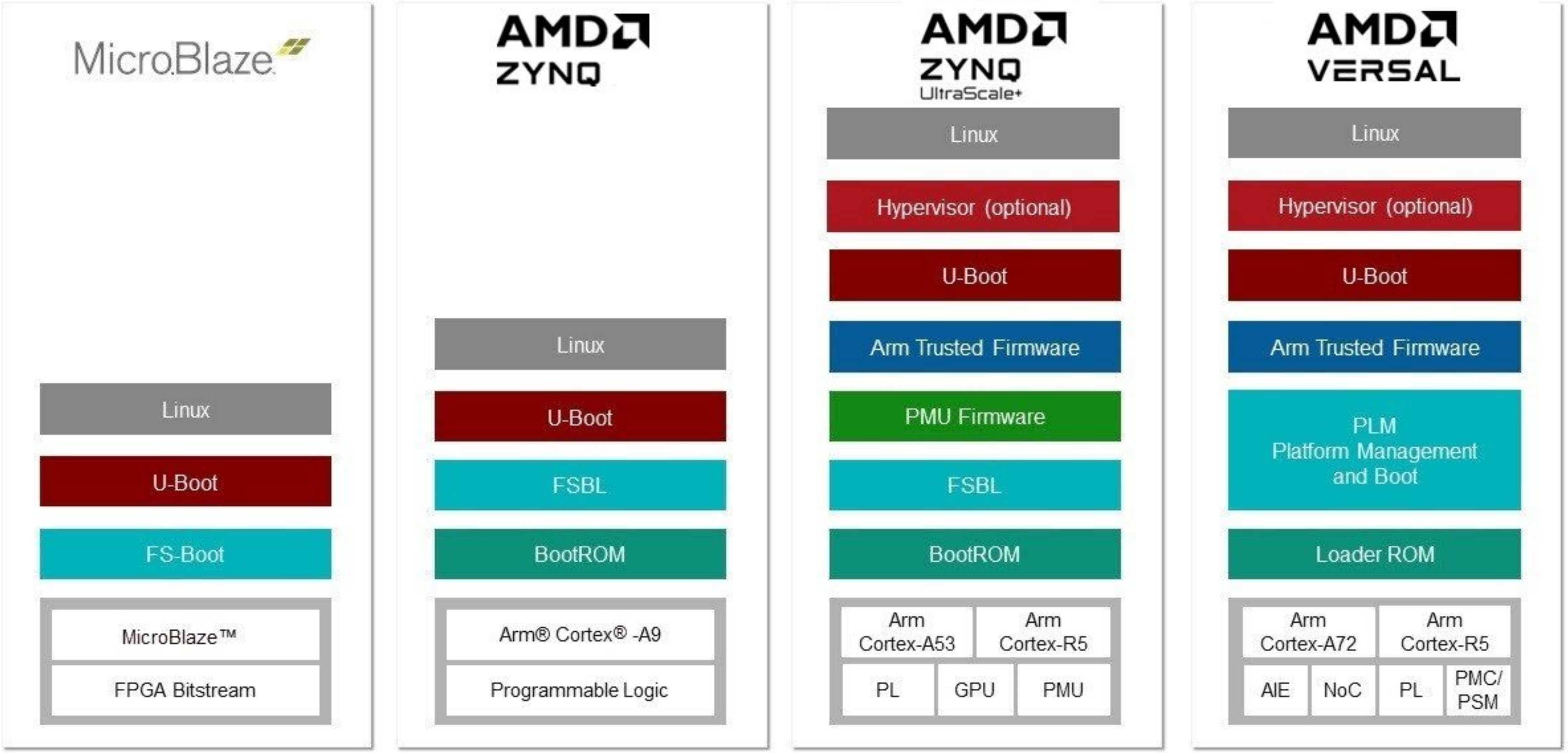
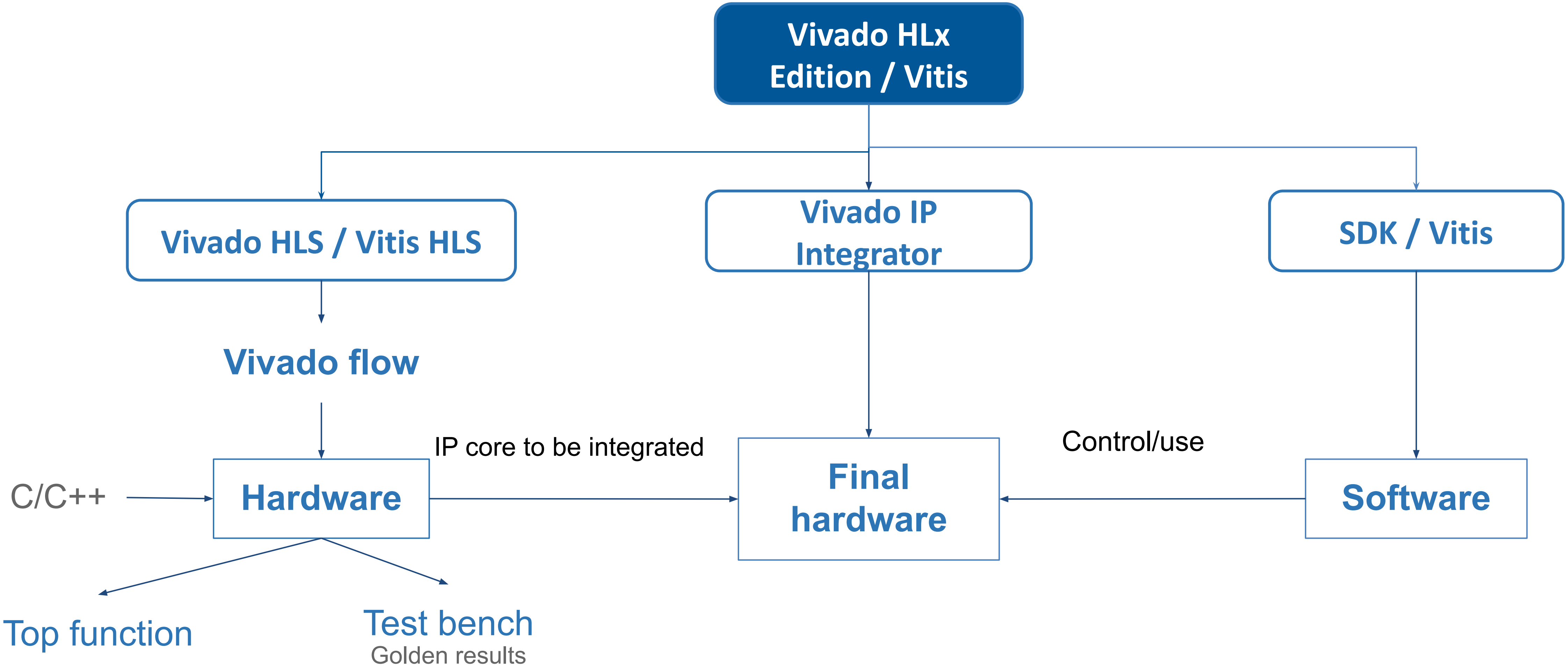


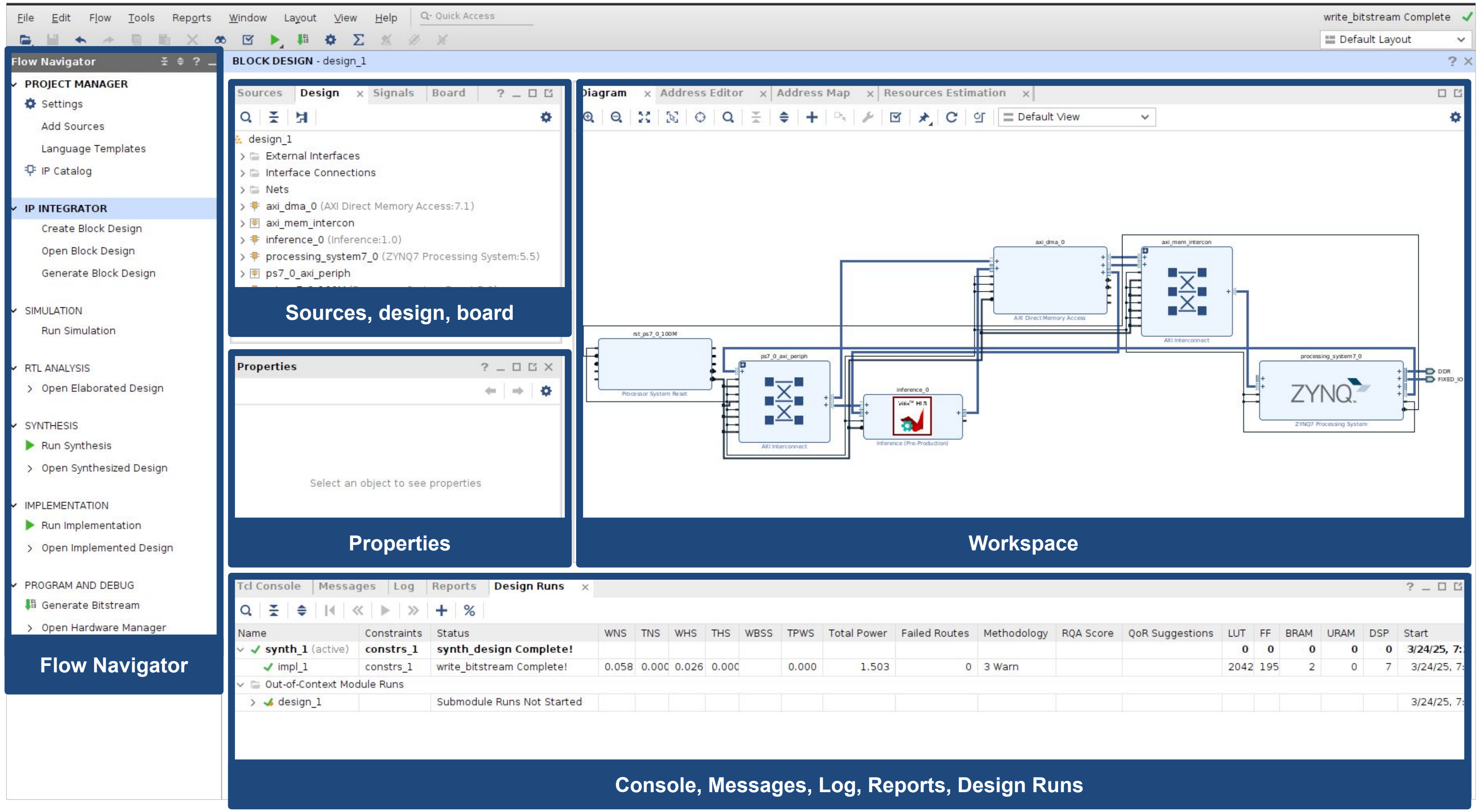
Image from https://xilinx.github.io/Embedded-Design-Tutorials/docs/2023.1/build/html/index.html

Development Tools

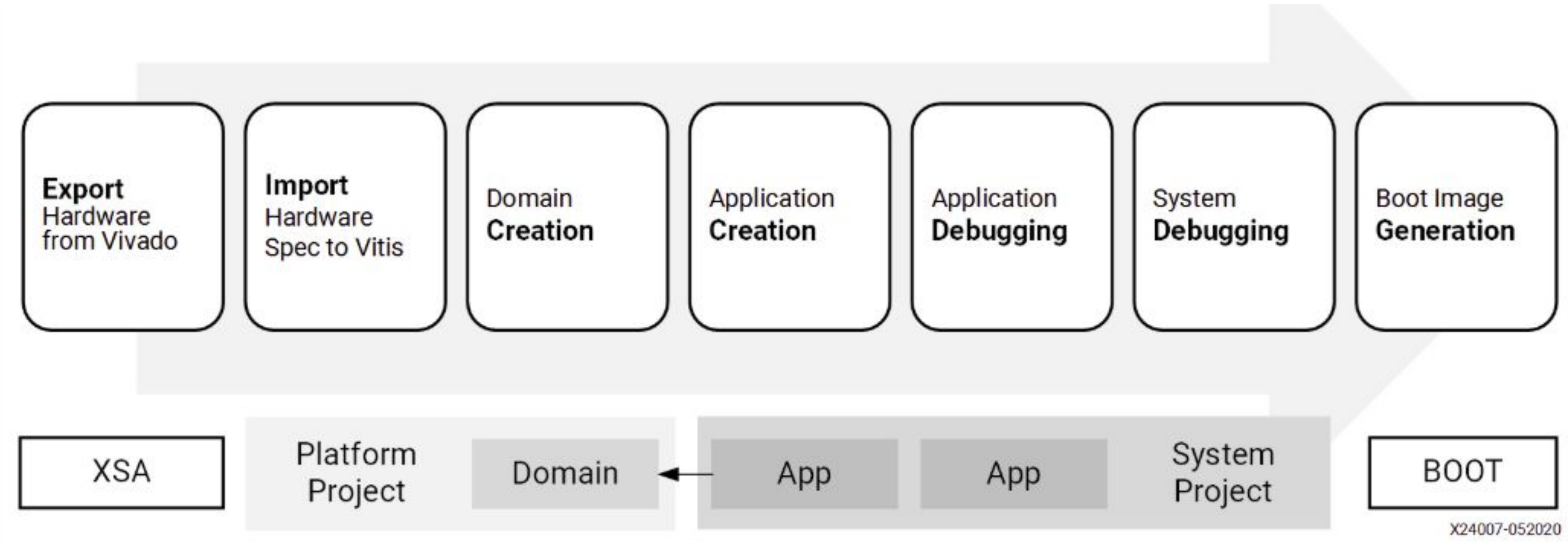
Development Tools



Development Tools - IP Integrator



Development Tools - Vitis



Demo:
Binary classifier - Vivado IP Integrator and Vitis

Thank you!

Romina Soledad Molina, Ph.D.
MLab-STI, ICTP

Mar del Plata, Argentina - 2025 -