

Lecture 5

Introduction to Zynq

Required Reading

The ZYNQ Book

- *Chapter 1: Introduction*
- *Chapter 2: The Zynq Device (“What is it?)*
- *Chapter 5: Applications and Opportunities (“What can I do with it?”)*

Xilinx Educational Video

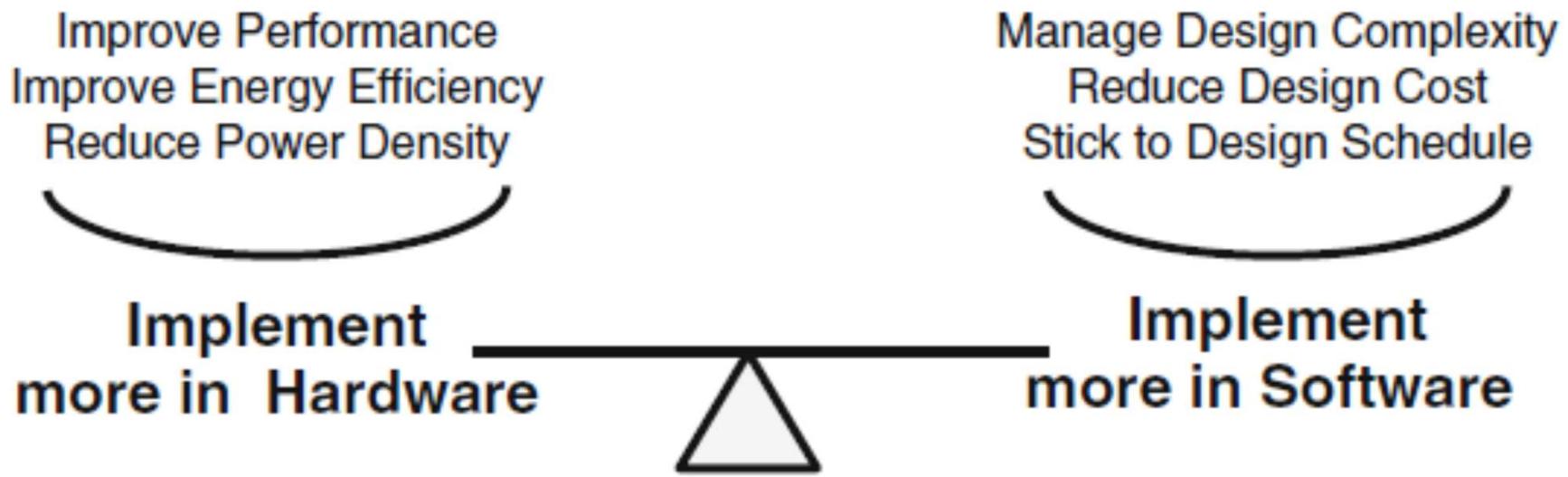
- *Why Zynq?*
<http://www.xilinx.com/training/zynq/why-zynq.htm>

What is Software/Hardware Co-design?

*Integrated design of systems that consist
of hardware and software components*

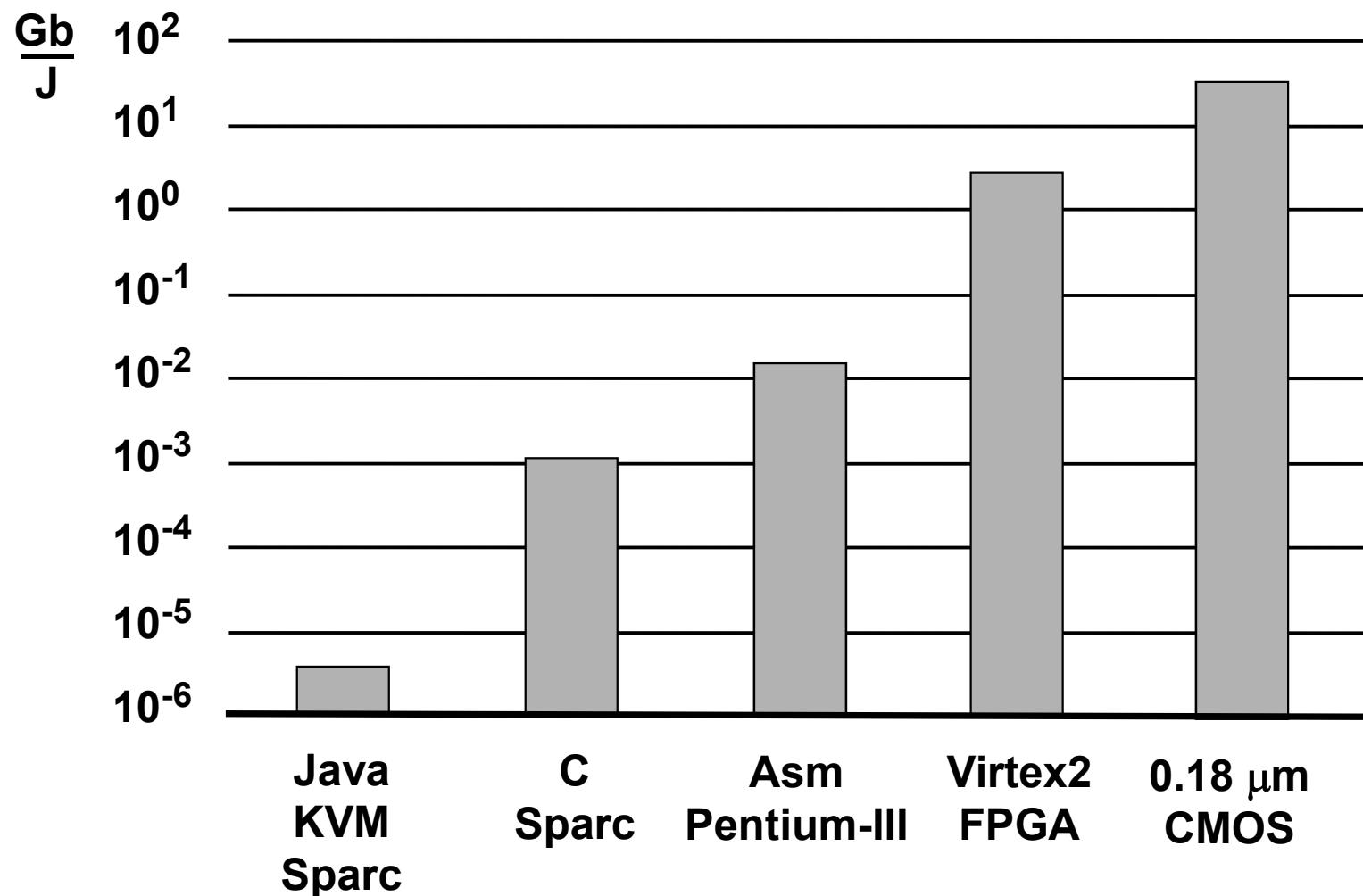
- Analysis of HW/SW boundaries and interfaces
- Evaluation of design alternatives

Software vs. Hardware Trade-offs



Source: A Practical Introduction to Hardware/Software Codesign

Energy Efficiency of AES Implementations on Various Platforms



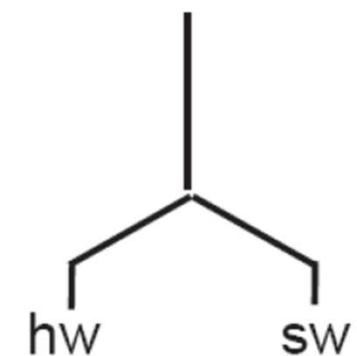
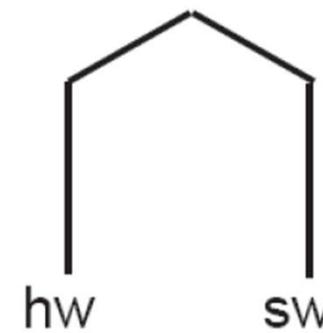
Source: A Practical Introduction to Hardware/Software Codesign

Why Co-design?

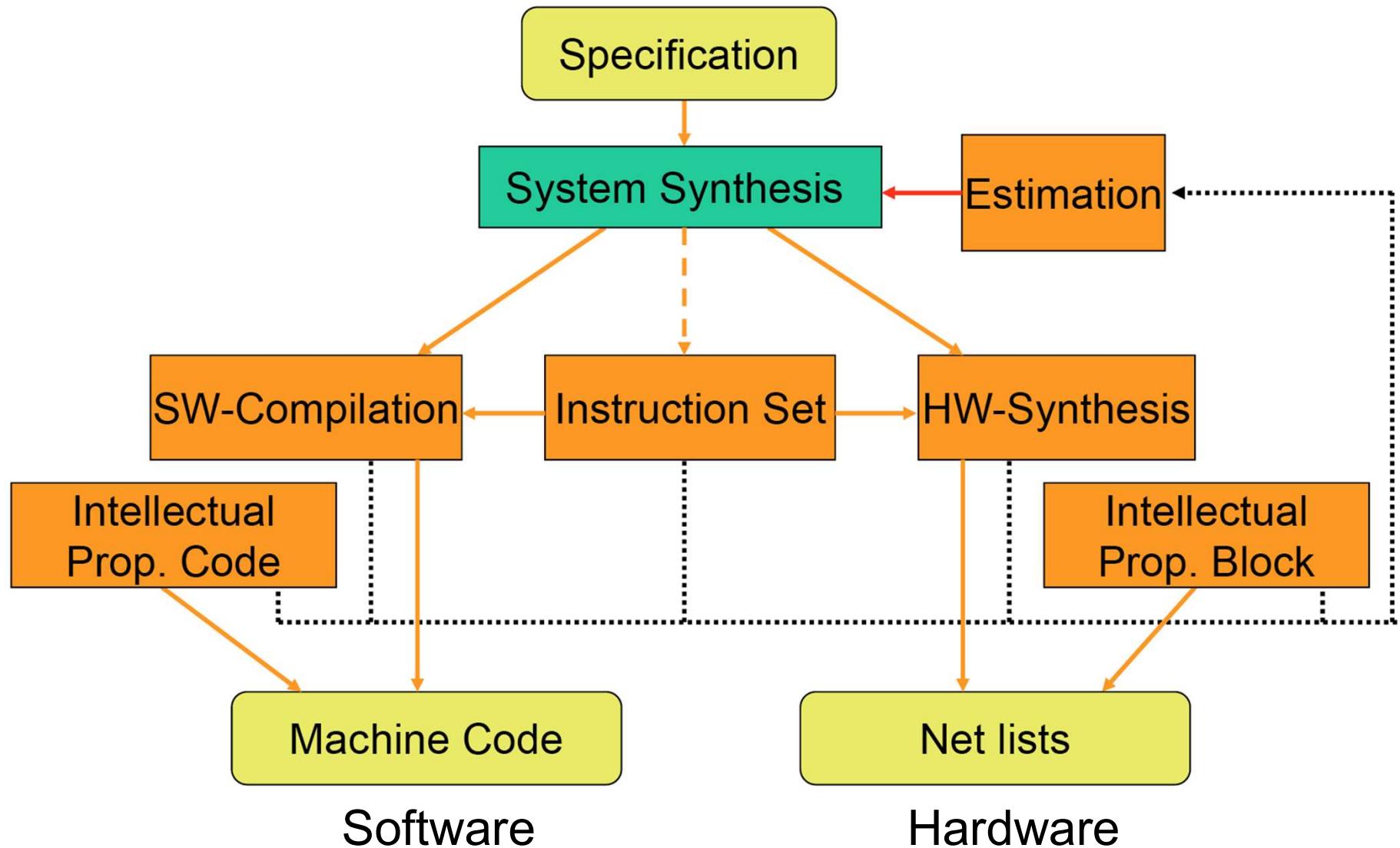
classic design



co-design

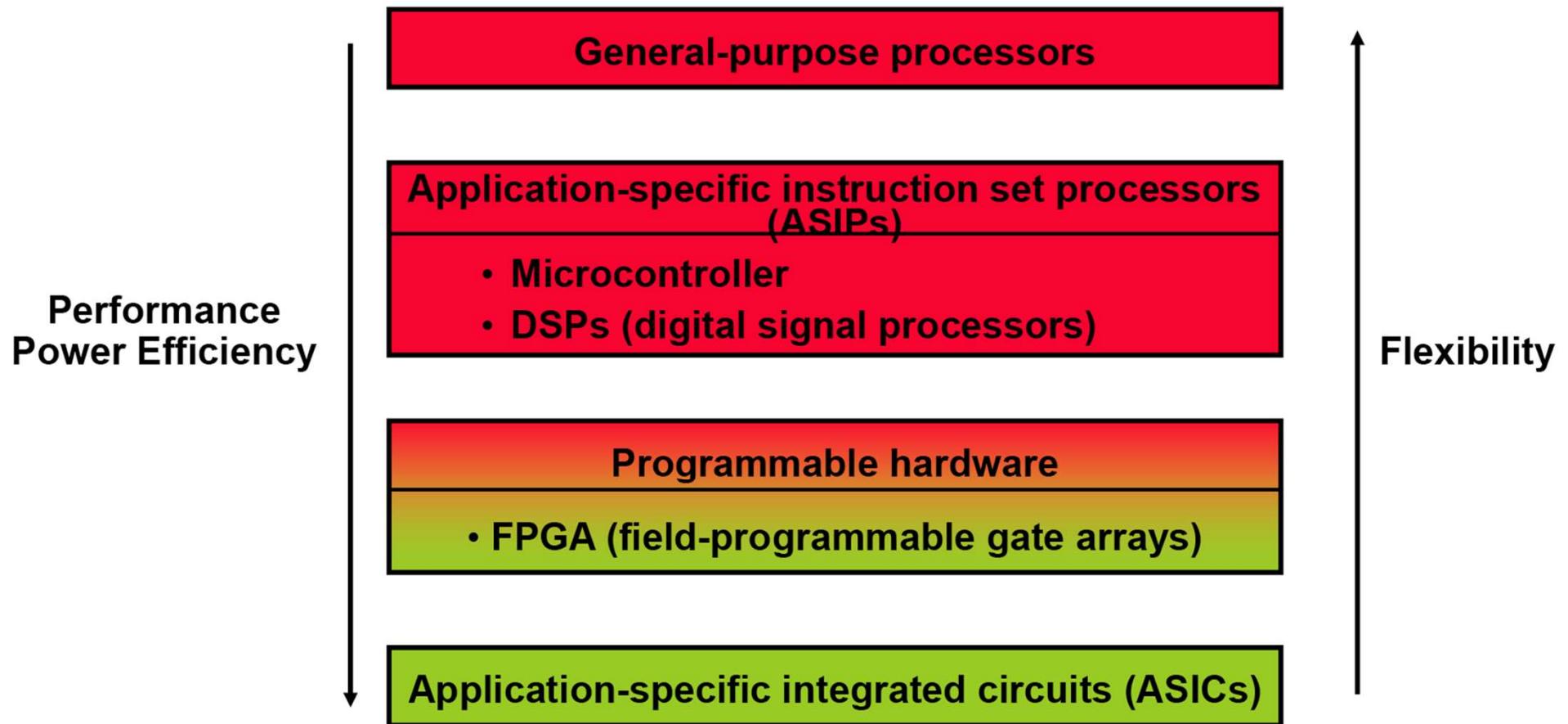


System Design Flow



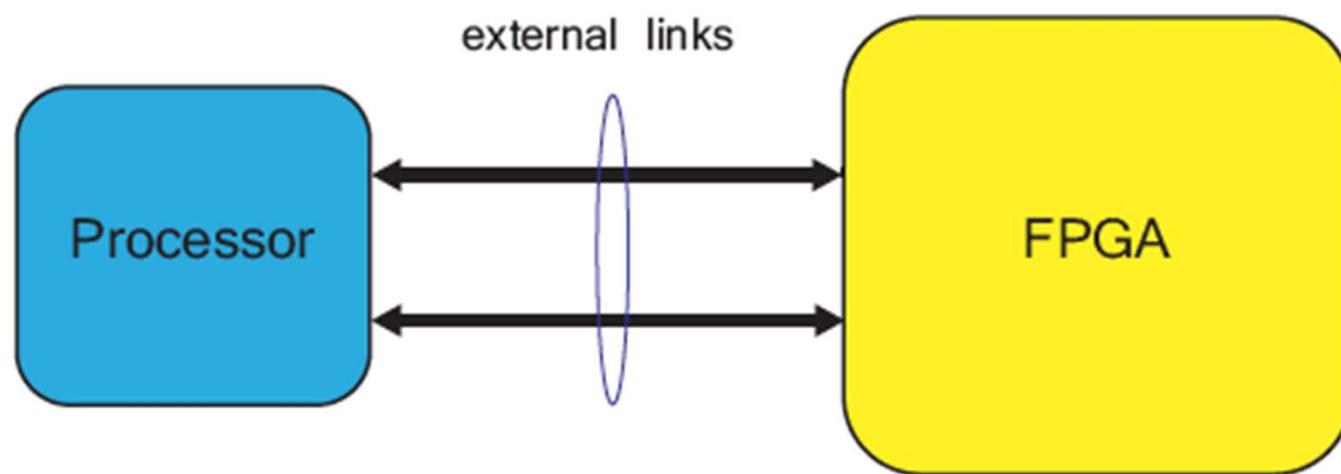
Source: ETHZ, Prof. Lothar Thiele

Implementation Alternatives



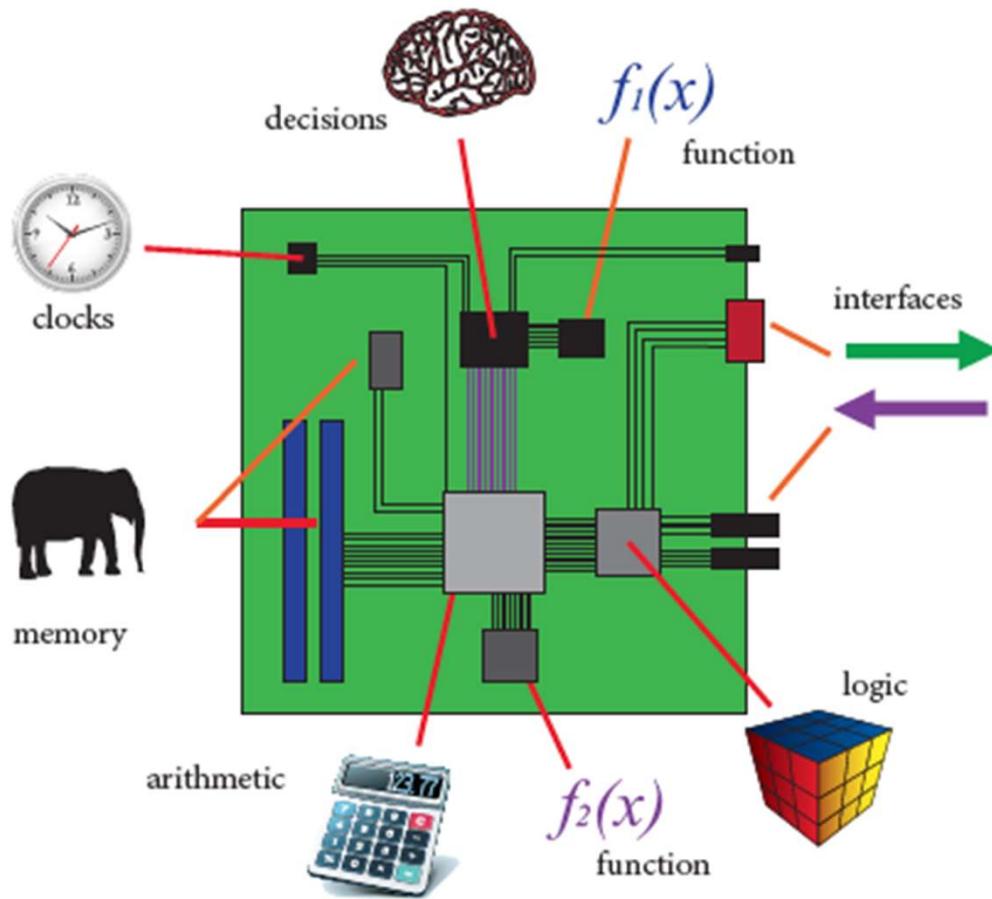
Source: ETHZ, Prof. Lothar Thiele

Traditional Discrete Component Architecture



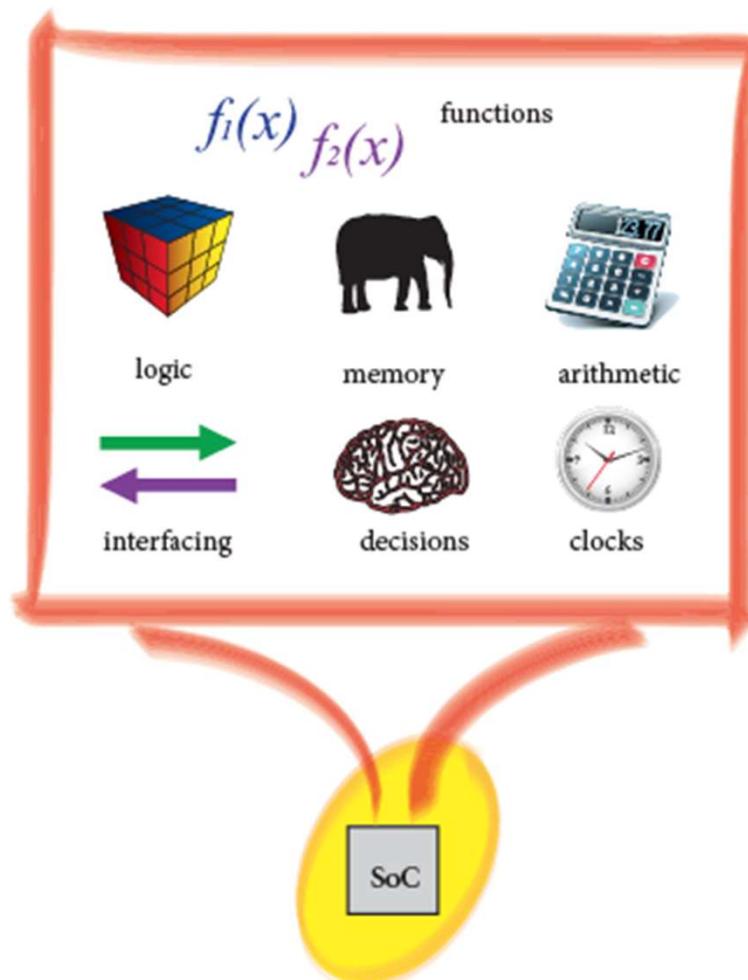
Source: The Zynq Book

System-on-a-Board



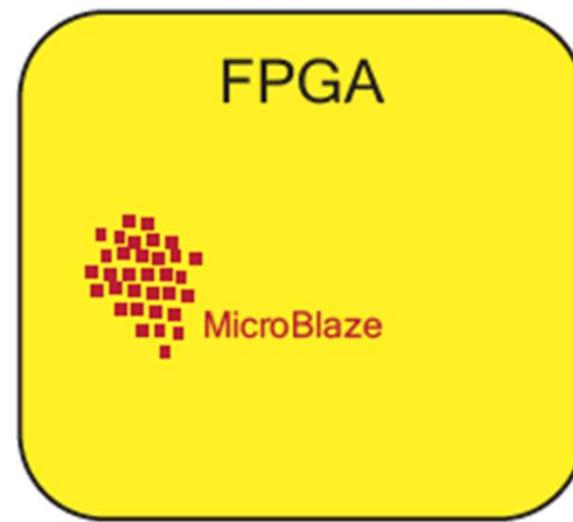
Source: The Zynq Book

System-on-Chip (SoC)



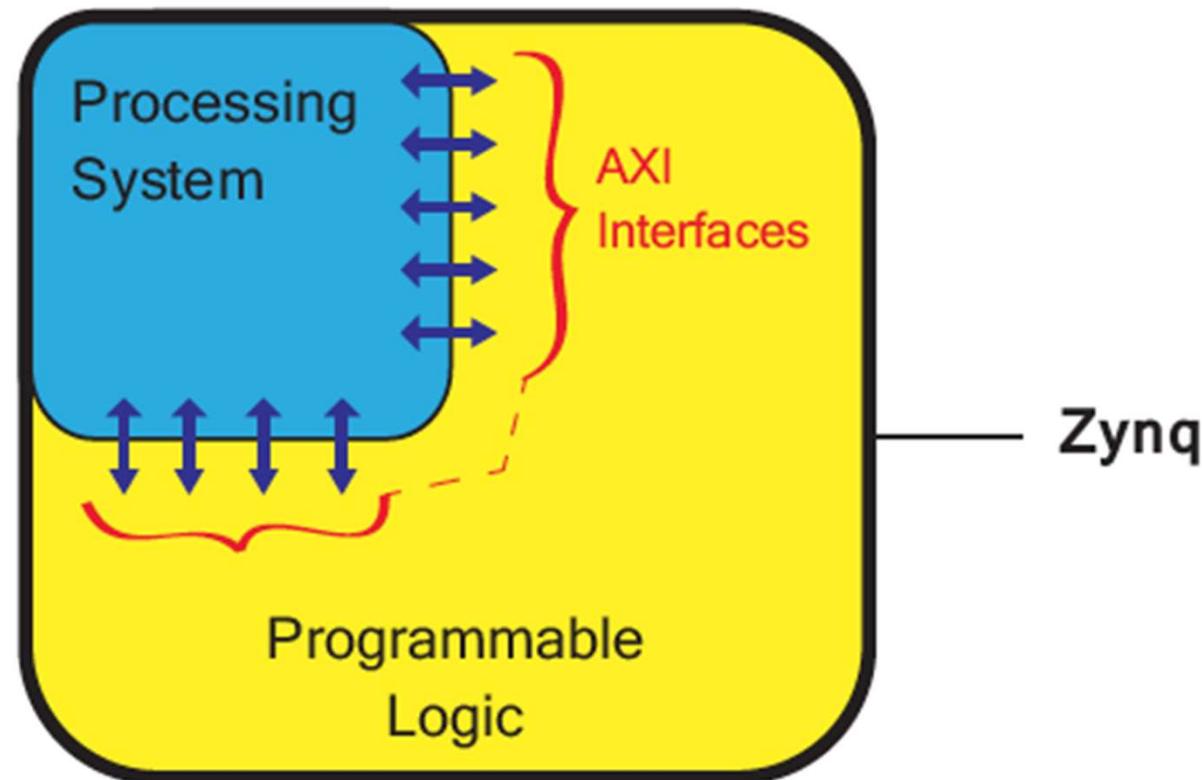
Source: The Zynq Book

FPGA with Soft Processor Core



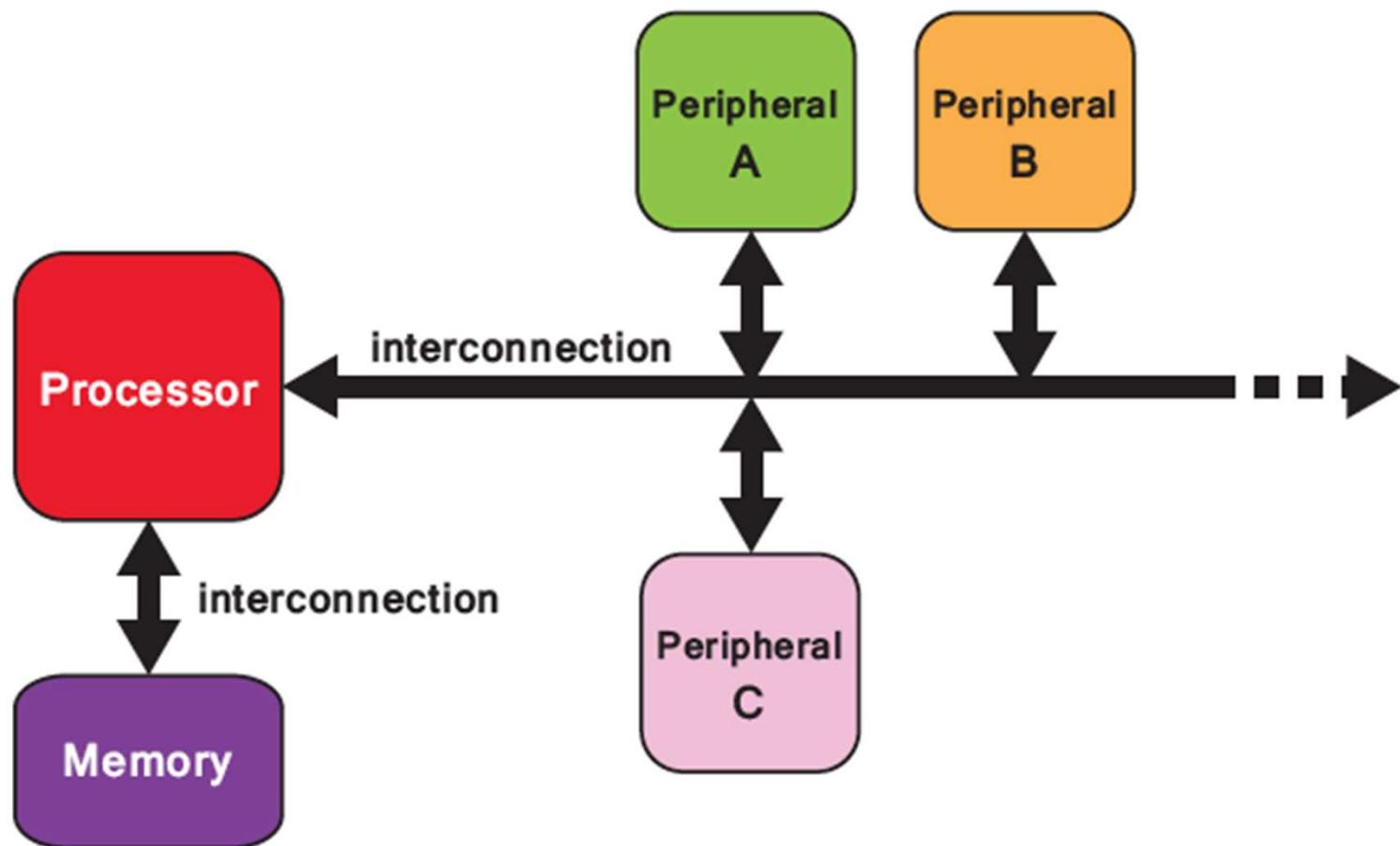
Source: The Zynq Book

A Simplified Model of the Zynq Architecture



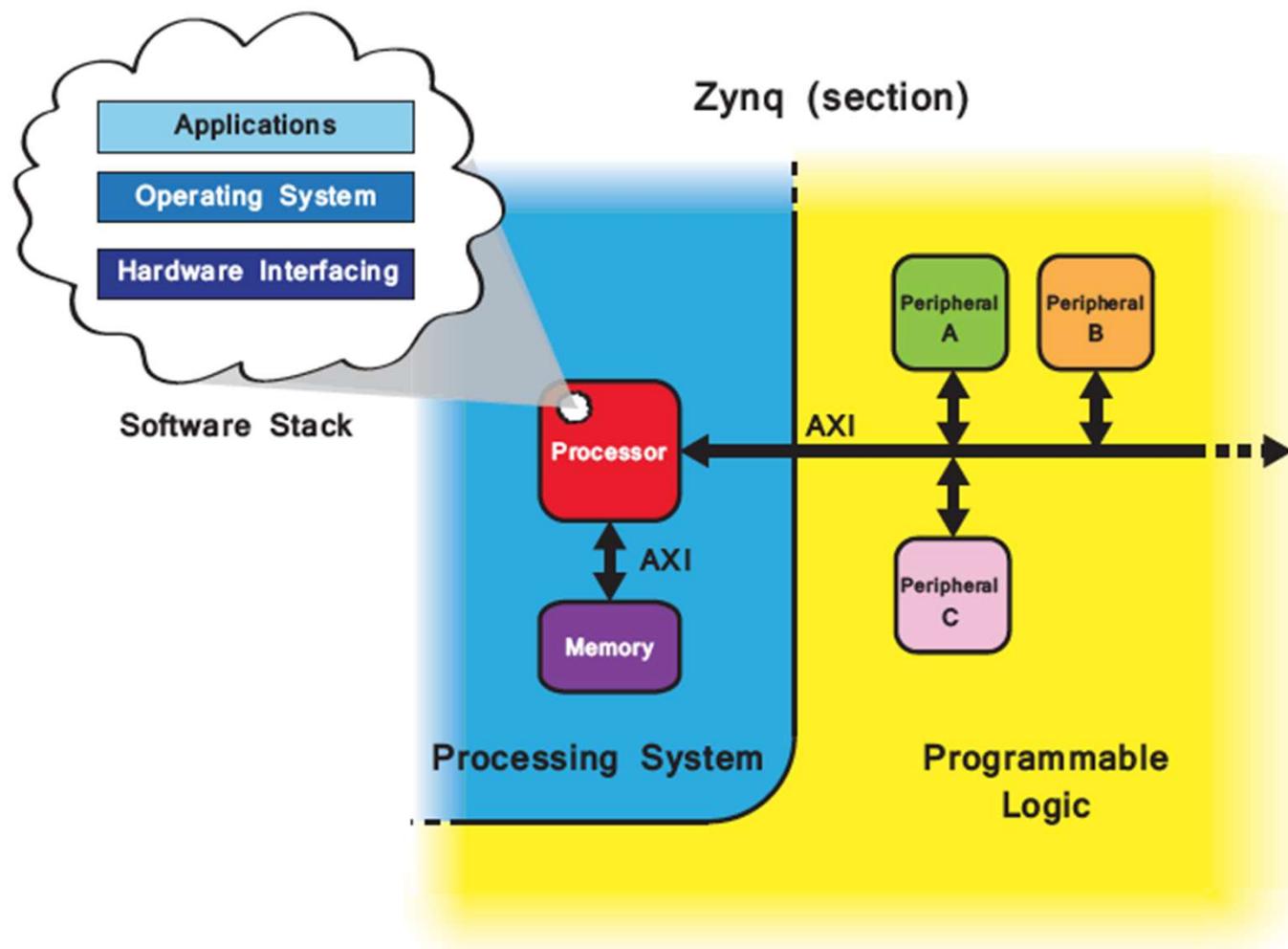
Source: The Zynq Book

Simplified Hardware Architecture of an Embedded SoC



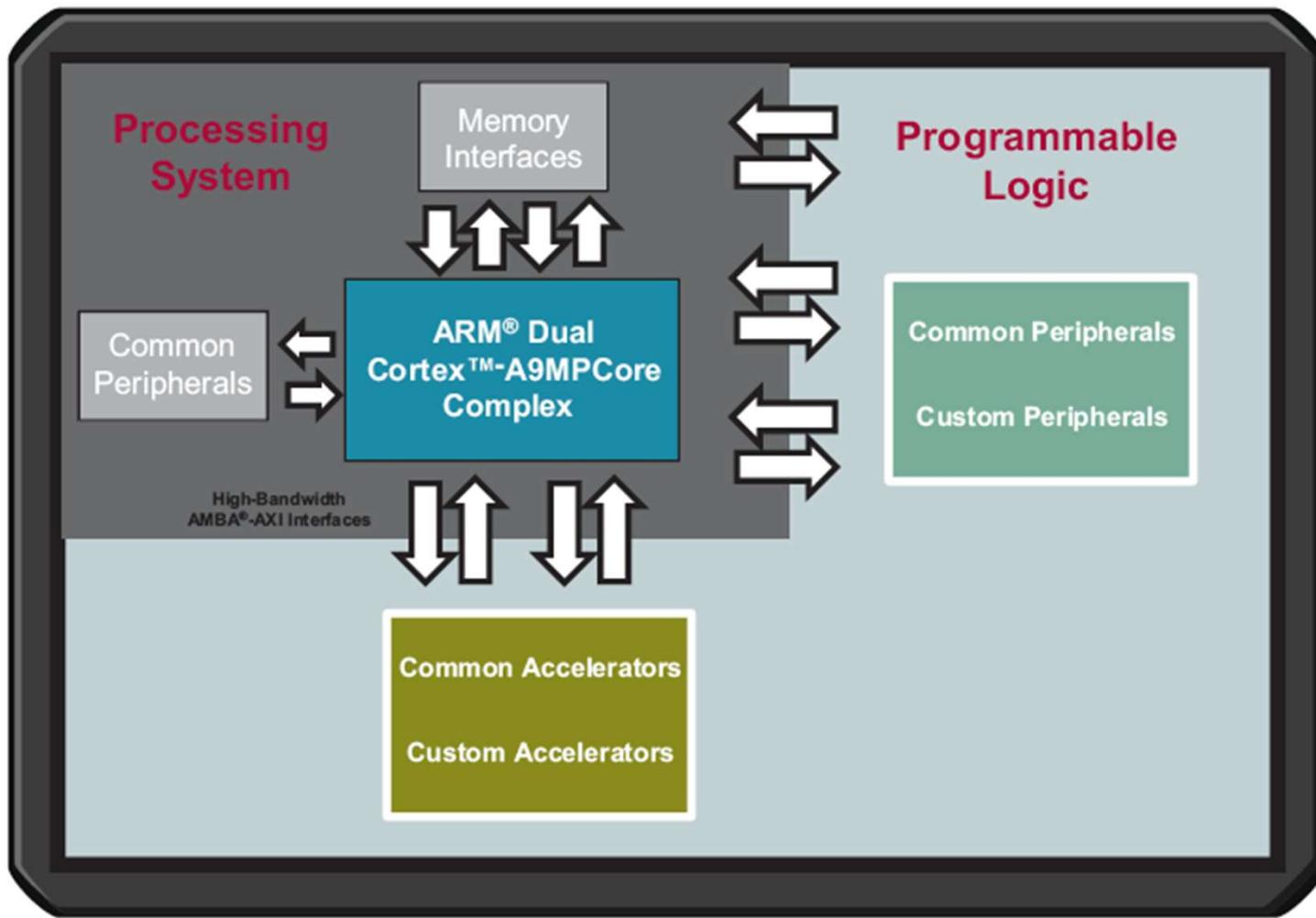
Source: The Zynq Book

Mapping of an Embedded SoC Hardware Architecture to Zynq



Source: The Zynq Book

Mapping of an Embedded SoC Hardware Architecture to Zynq



WP369_04_042310

Source: Xilinx White Paper: Extensible Processing Platform

Comparison with Alternative Solutions

	ASIC	ASSP	2 Chip Solution	Zynq
Performance	+	+	□	+
Power	+	+	□	+
Unit Cost	+	+	□	□
Total Cost of Ownership	□	+	+	+
Risk	□	+	+	+
Time to Market	□	+	+	+
Flexibility	□	□	+	+
Scalability	□	□	+	+

□ positive, □ negative, □ neutral

Source: Xilinx Video Tutorials

Zynq Highlights

➤ Complete ARM®-based Processing System

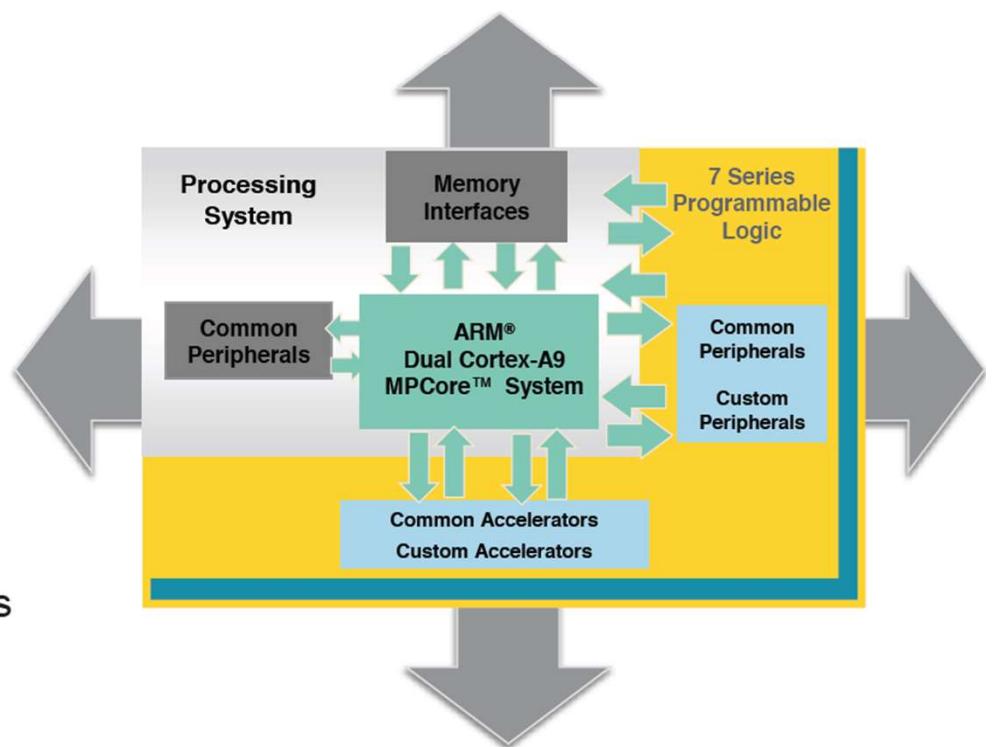
- Dual ARM Cortex™-A9 MPCore™, processor centric
- Integrated memory controllers & peripherals
- Fully autonomous to the Programmable Logic

➤ Tightly Integrated Programmable Logic

- Used to extend Processing System
- High performance ARM AXI interfaces
- Scalable density and performance

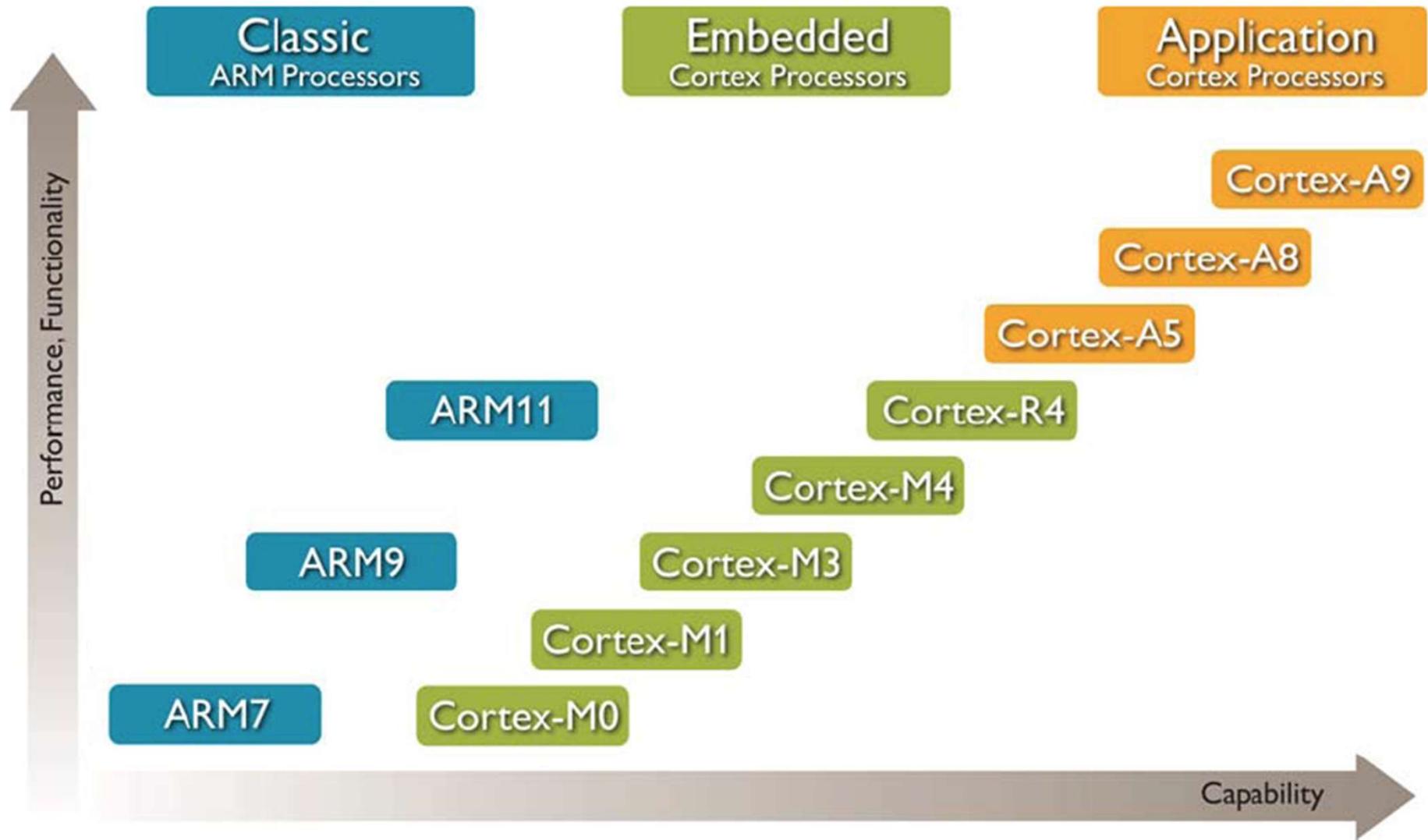
➤ Flexible Array of I/O

- Wide range of external multi-standard I/O
- High performance integrated serial transceivers
- Analog-to-Digital Converter inputs



Source: Xilinx Video Tutorials

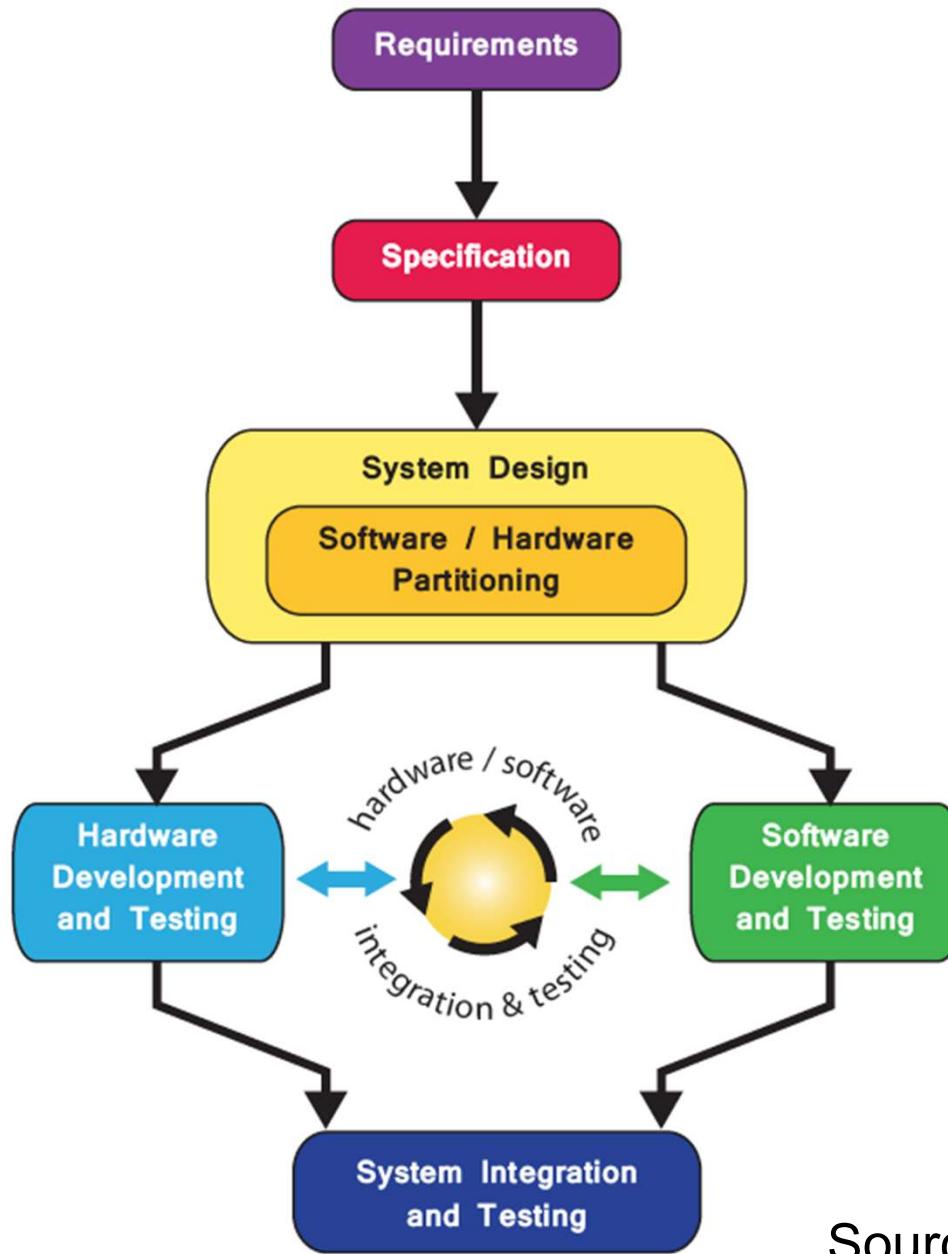
ARM Processor Roadmap



WP369_03_042110

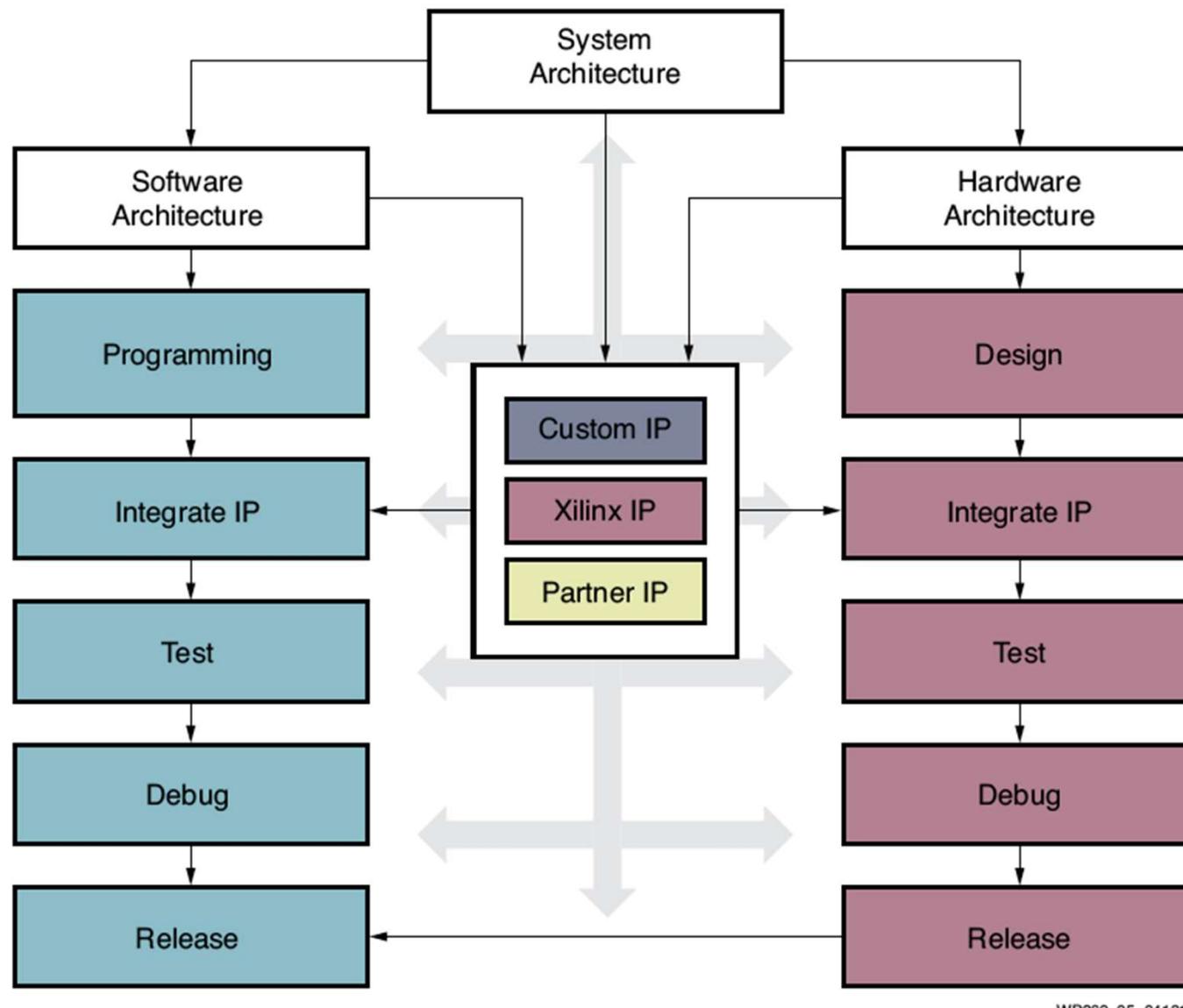
Source: Xilinx White Paper: Extensible Processing Platform

Basic Design Flow for Zynq SoC



Source: The Zynq Book

Design Flow for Zynq SoC



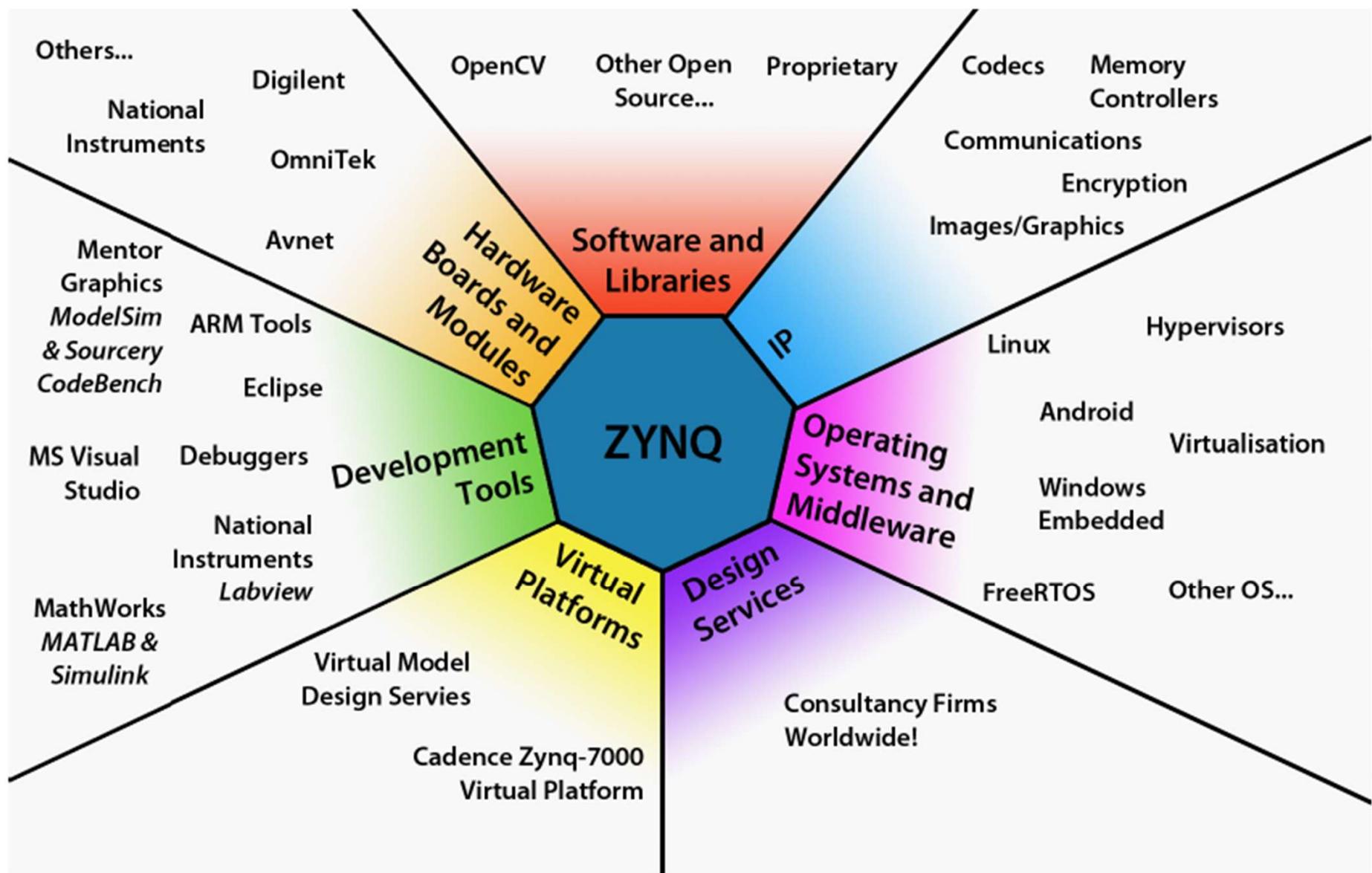
WP369_05_041810

Source: Xilinx White Paper: Extensible Processing Platform

Zynq SoC Ecosystem



Zynq SoC Ecosystem



Source: The Zynq Book

Alternative Solutions

Xilinx Zynq

Zynq-7000 All Programmable SoCs with Cortex-A9 MPCore



Altera Arria V & Cyclone V

Hard processor system (HPS) with Cortex-A9 MPCore

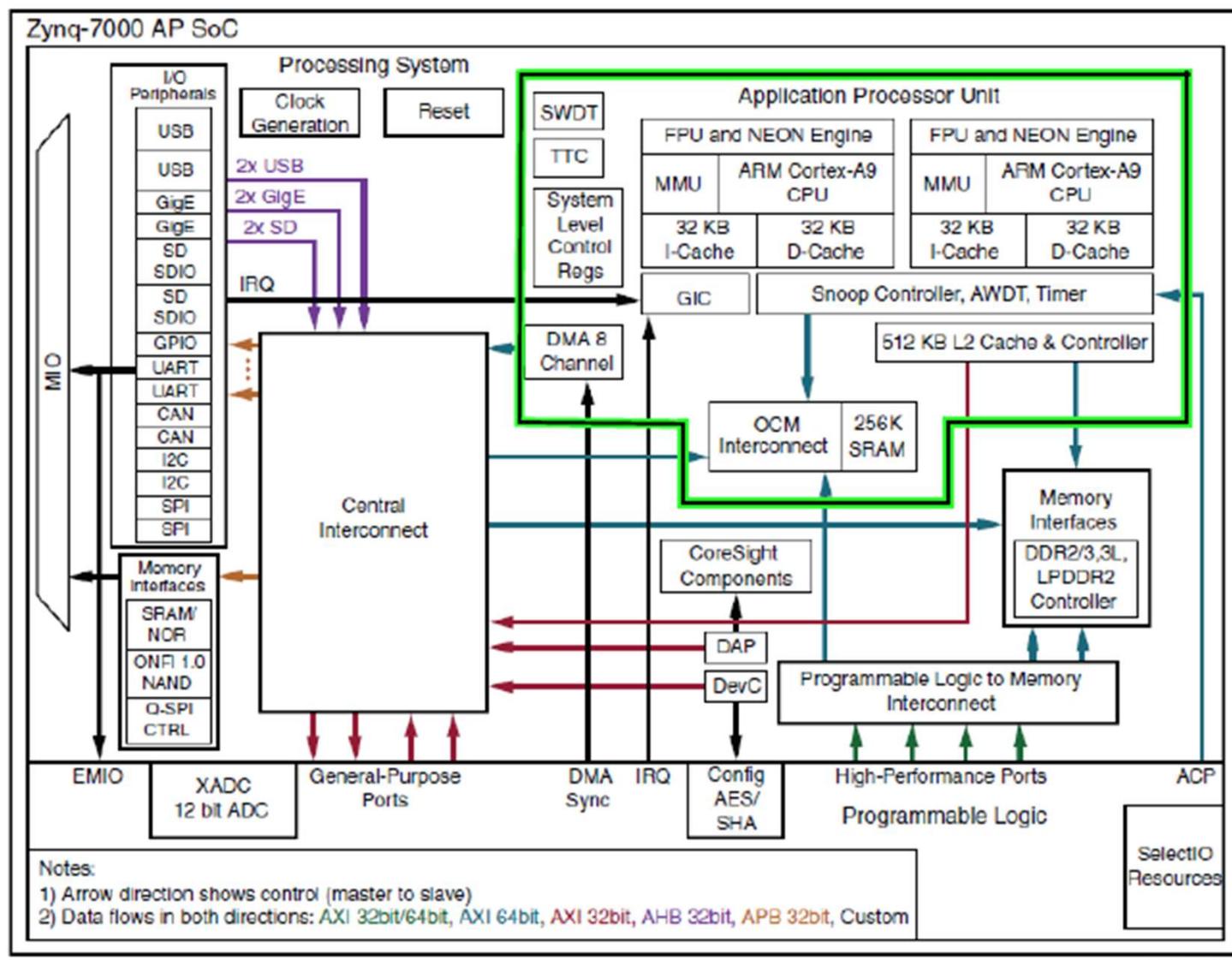


Microsemi Smartfusion2

Cortex M3

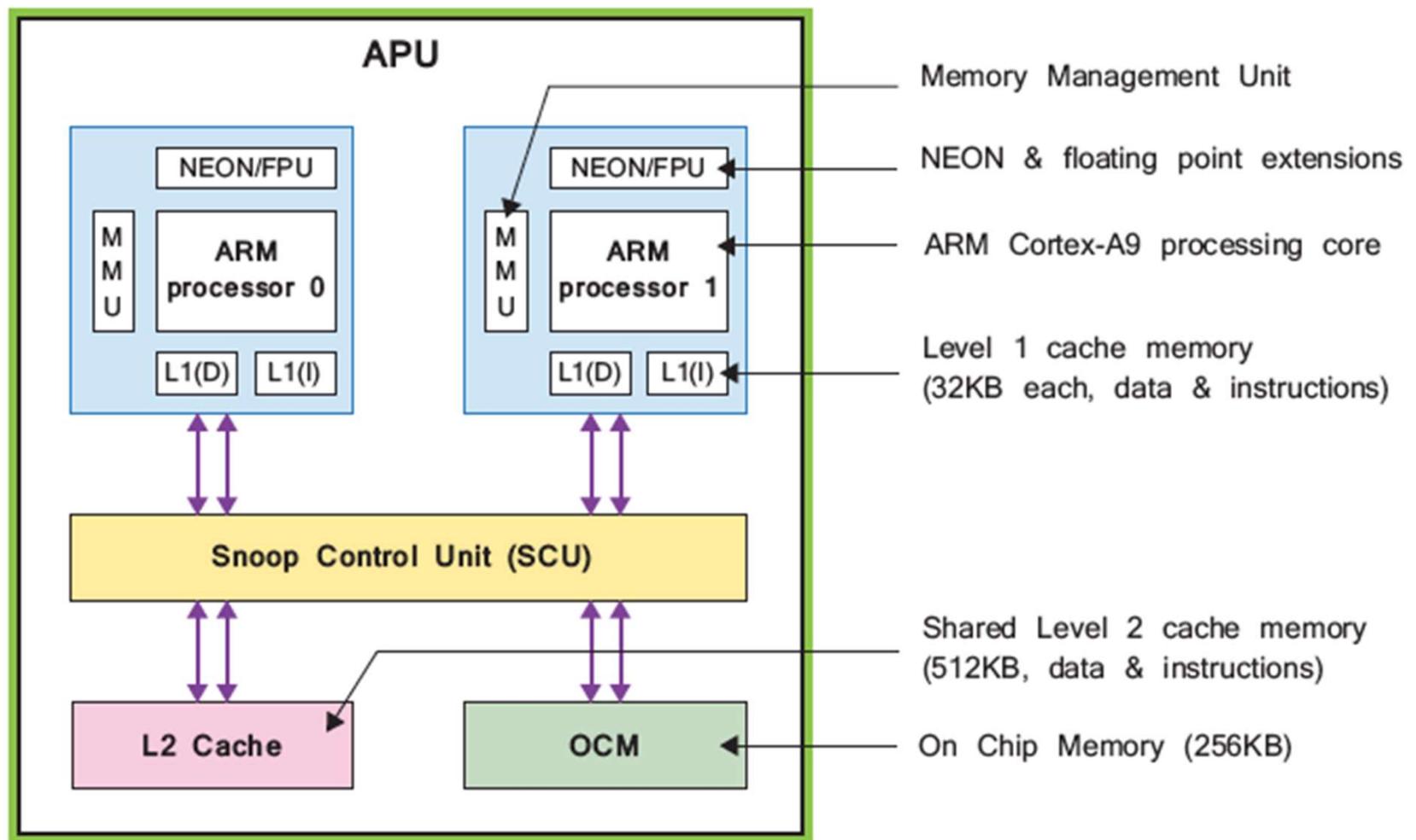


The Zynq Processing System



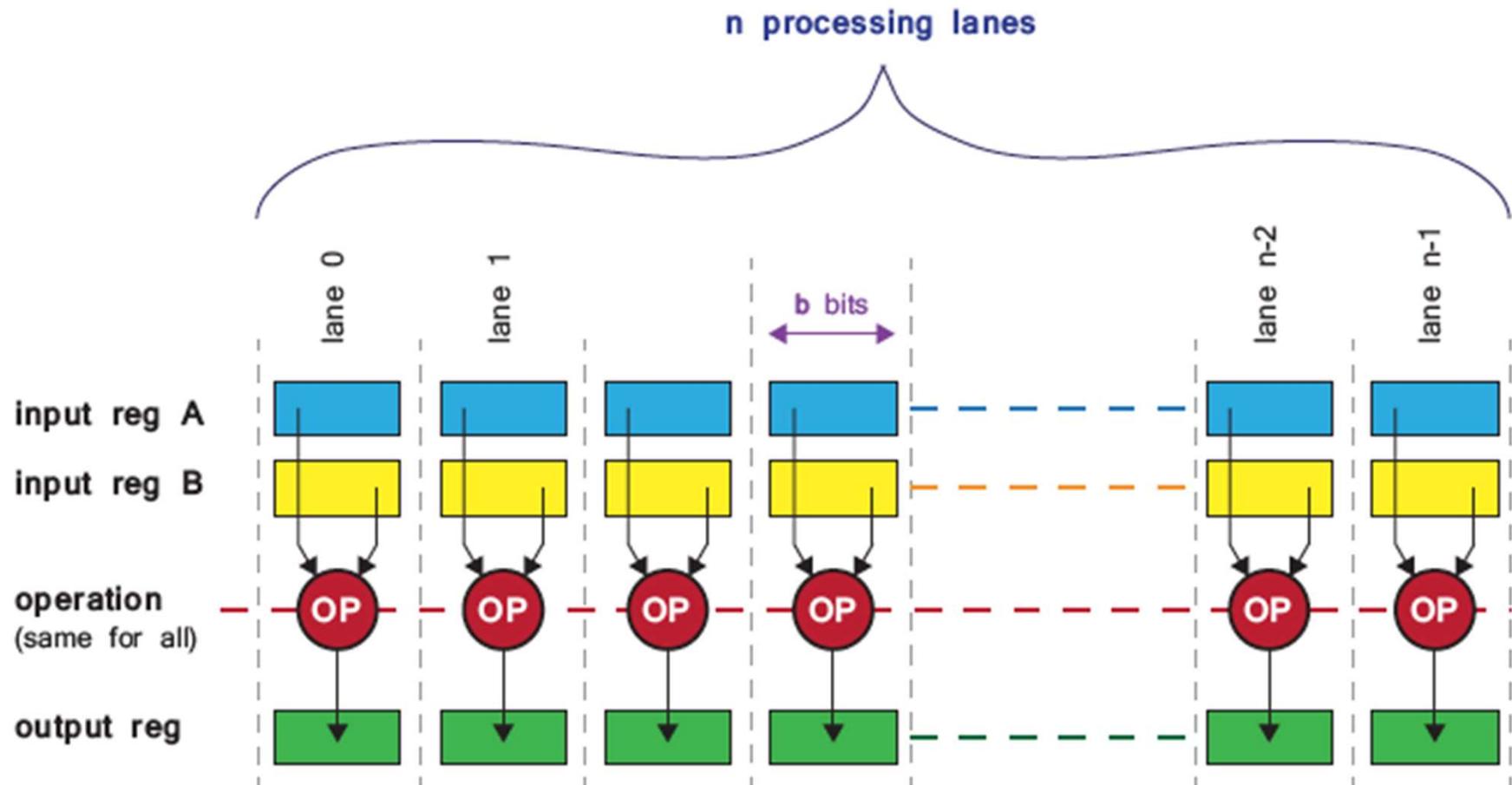
Source: The Zynq Book

Simplified Block Diagram of the Application Processing Unit (APU)



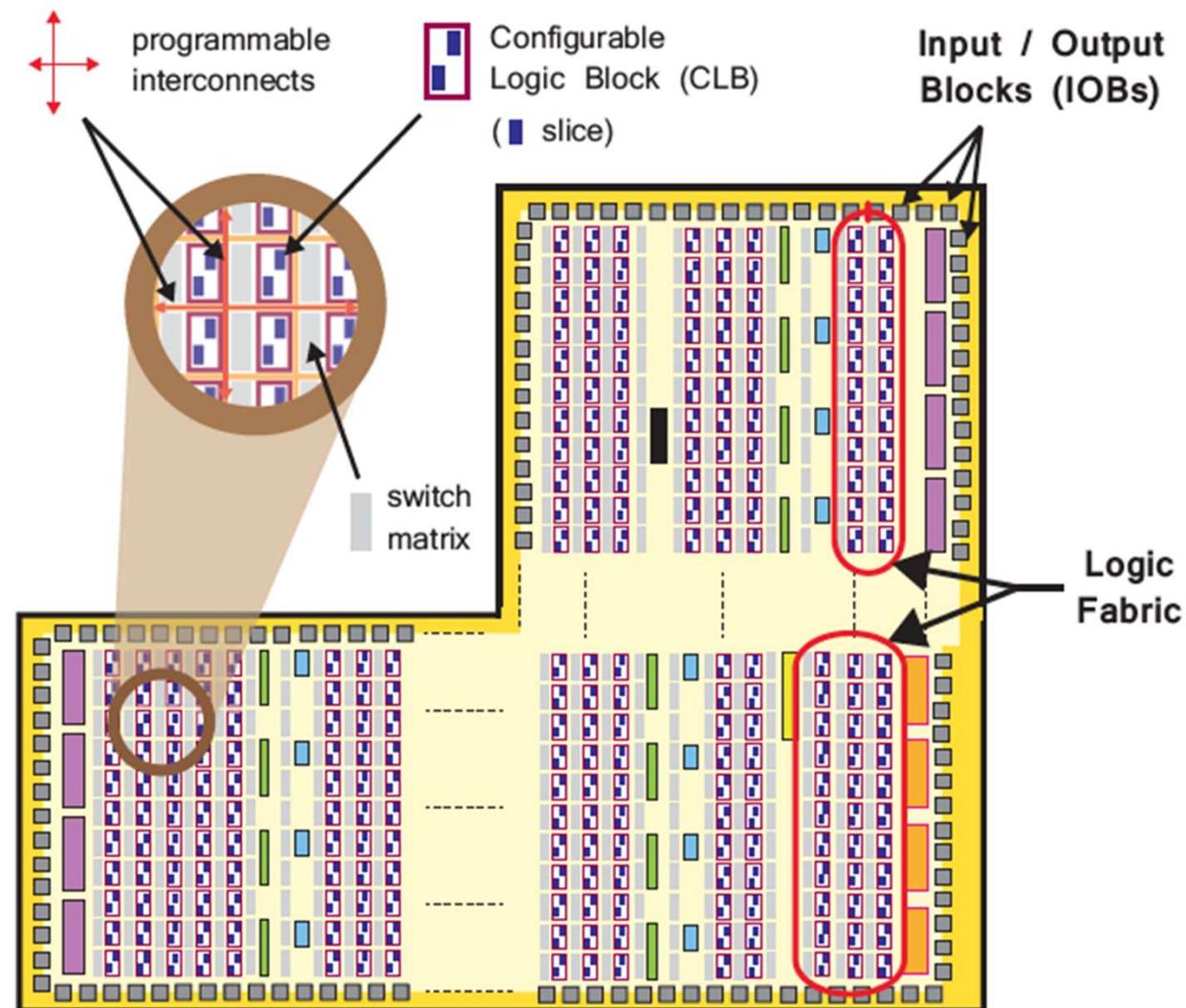
Source: The Zynq Book

SIMD (Single Instruction Multiple Data) Processing in the NEON Media Processing Engine (MPE)



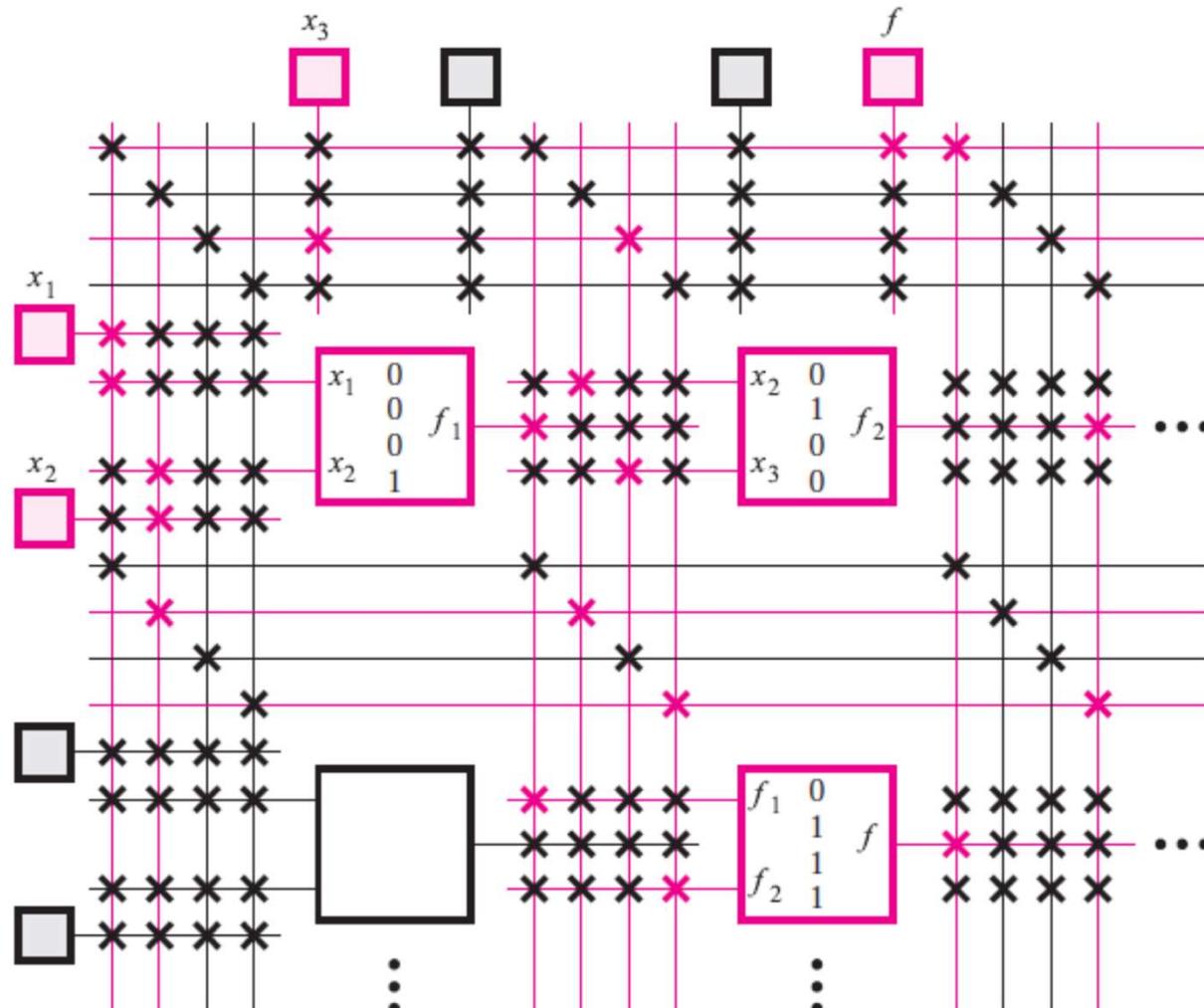
Source: The Zynq Book

Programmable Logic (PL) CLBs and IOBs



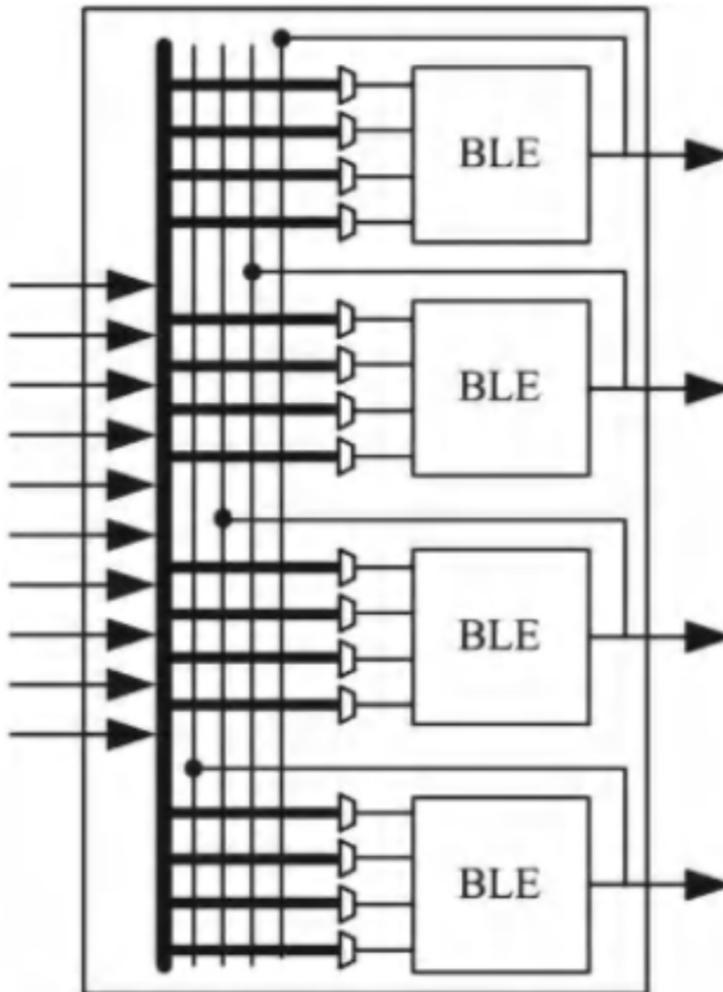
Source: The Zynq Book

Example of Programmable Logic



$$f = f_1 + f_2 = x_1 x_2 + \bar{x}_2 x_3$$

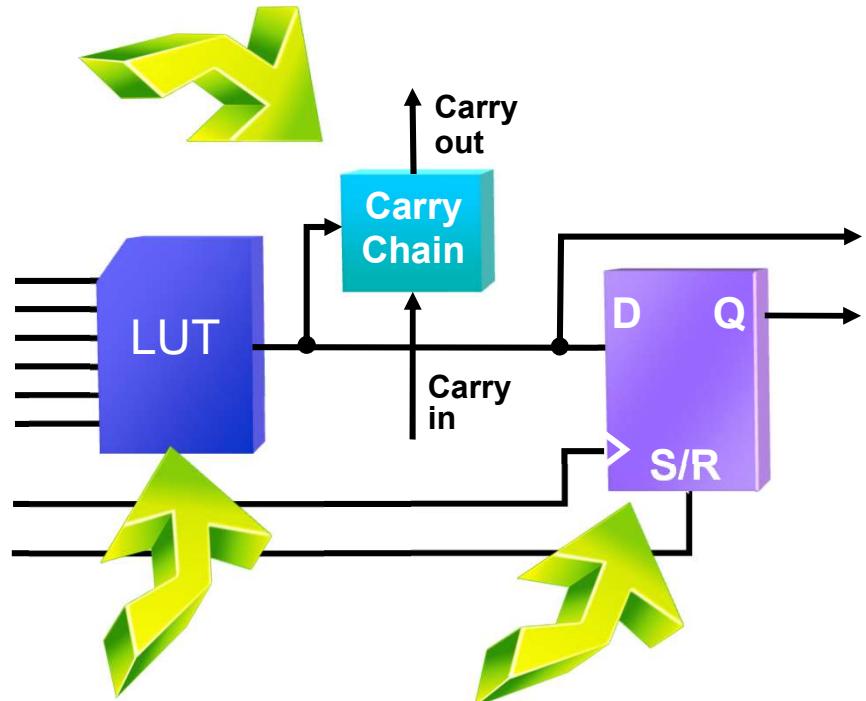
Configurable Logic Blocks



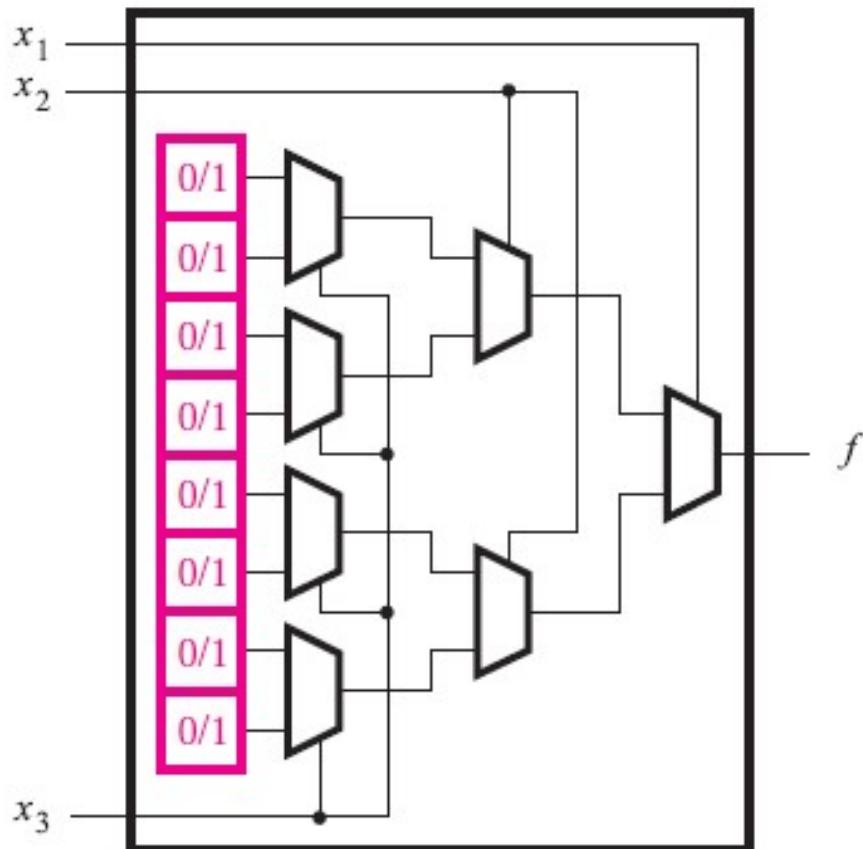
A configurable logic block (CLB) having four BLEs

Xilinx BLEs: Logic Cells

- Logic cells include
 - Combinatorial logic, arithmetic logic, and a register
- Combinatorial logic is implemented using Look-Up Tables (LUTs)
- Register can function as latches, JK, SR, D, and T-type flip-flops
- Arithmetic logic is a dedicated carry chain for implementing fast arithmetic operations



LUT: Lookup Table

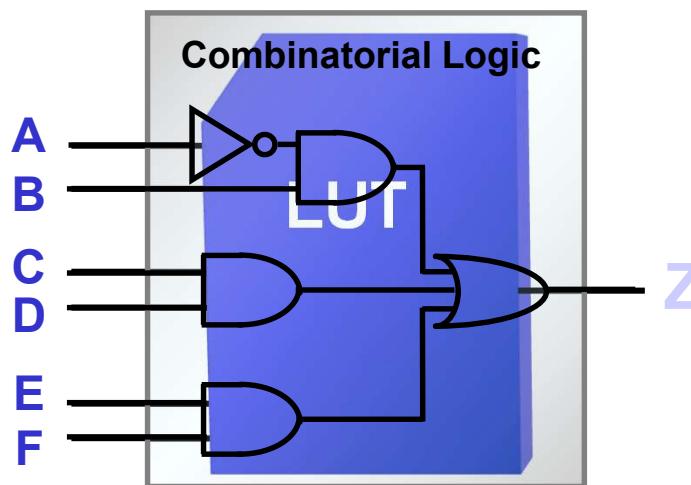


- Used to implement a small logic function
- Composed of:
 - storage cells store values that produce the output of the logic function f
 - Multiplexers select the content of one of the storage cells as the output of the LUT
- LUT's size is defined by the number of inputs

Combinatorial Logic

A	B	C	D	E	F	Z
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	1
0	0	0	1	0	0	1
0	0	0	1	0	1	
1	
0	0	1	1	0	0	0
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	0	1	1	1	1	1

- LUTs function as a Memory

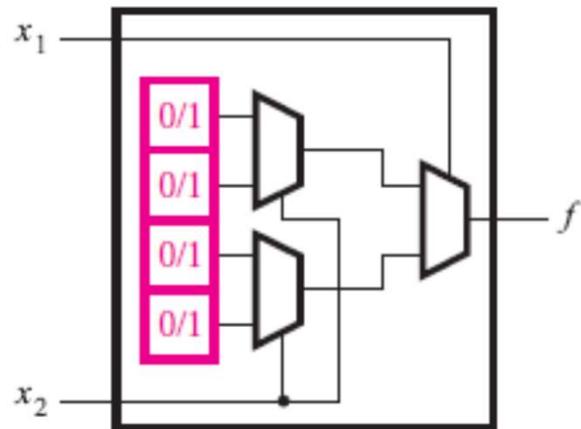


They generate
the output
value...

for a given set
of inputs

- Constant delay through a LUT
- Limited by the number of inputs and outputs, not by complexity

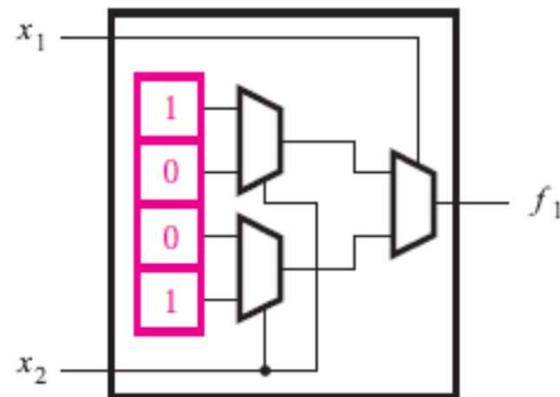
LUT: A Simple Example



(a) Circuit for a two-input LUT

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

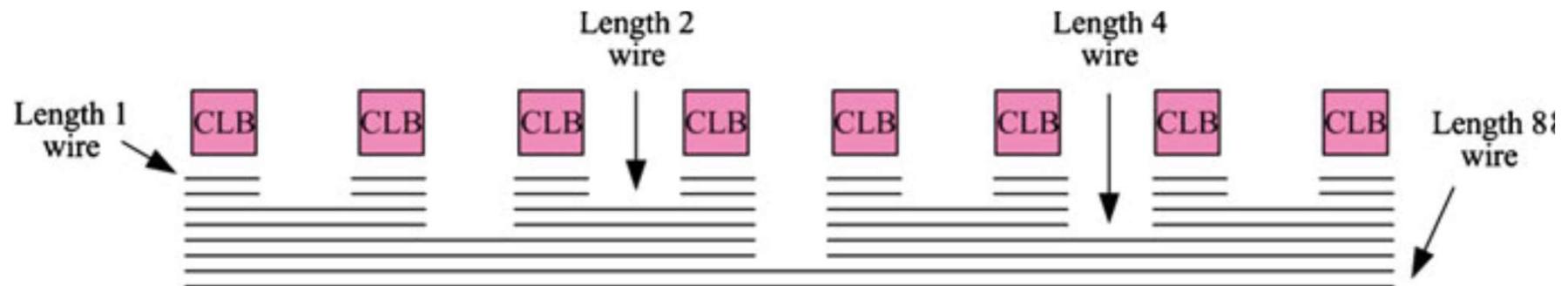
(b) $f_1 = \overline{x}_1\overline{x}_2 + x_1x_2$



(c) Storage cell contents in the LUT

Routing Network Architecture

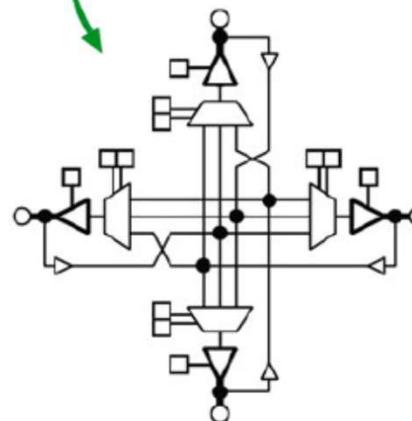
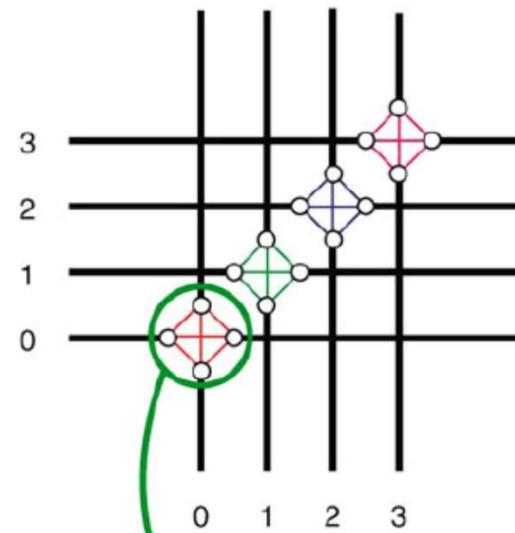
- Multi-length wires are created to balance flexibility, area and delay of the routing network
 - Longer wire segments:
 - Span multiple blocks and require fewer switches, thereby reducing routing area and delay
 - But also decrease routing flexibility, which reduces the probability to route a hardware circuit successfully



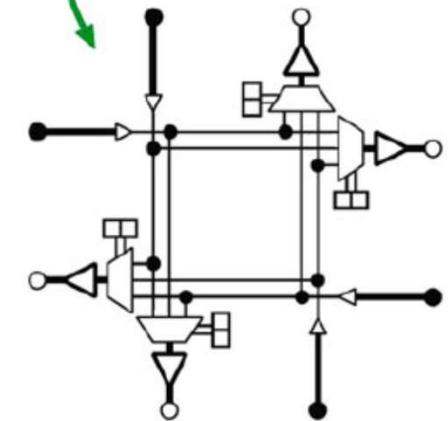
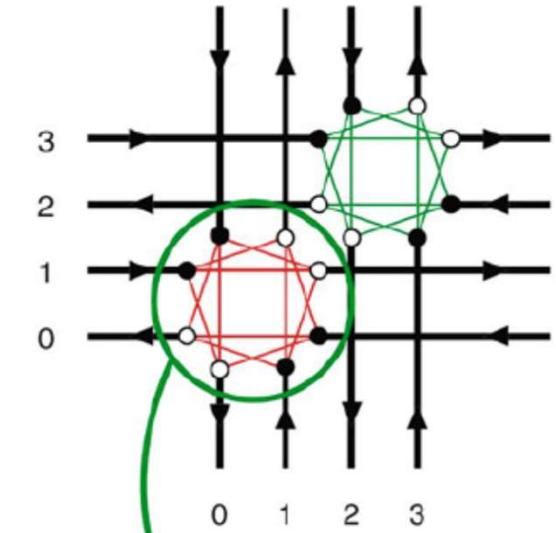
Routing Network Architecture

- Routing tracks can be **bidirectional** or **unidirectional**
 - Channel width of unidirectional wiring must be in multiples of 2

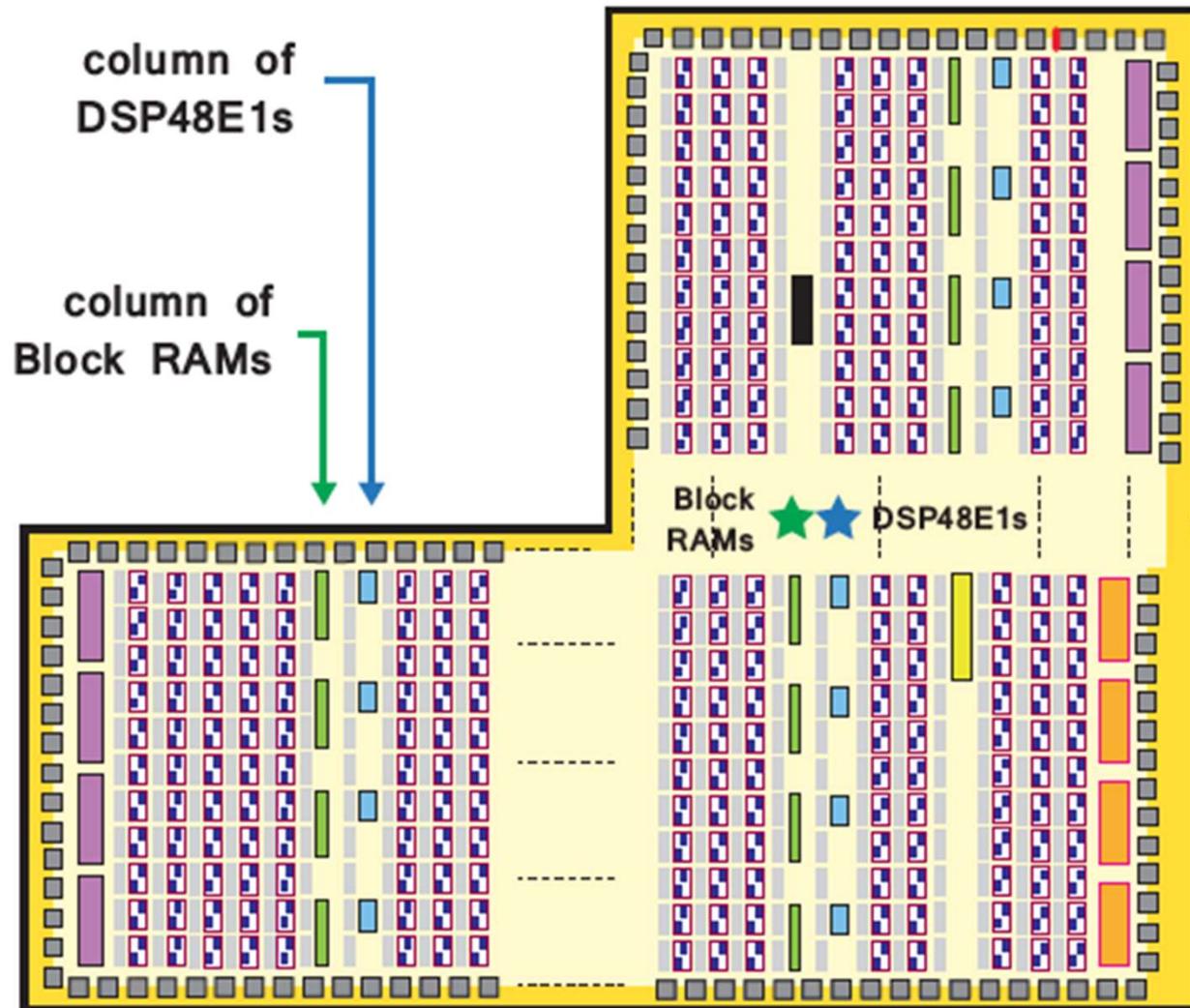
(a) bidirectional



(b) directional

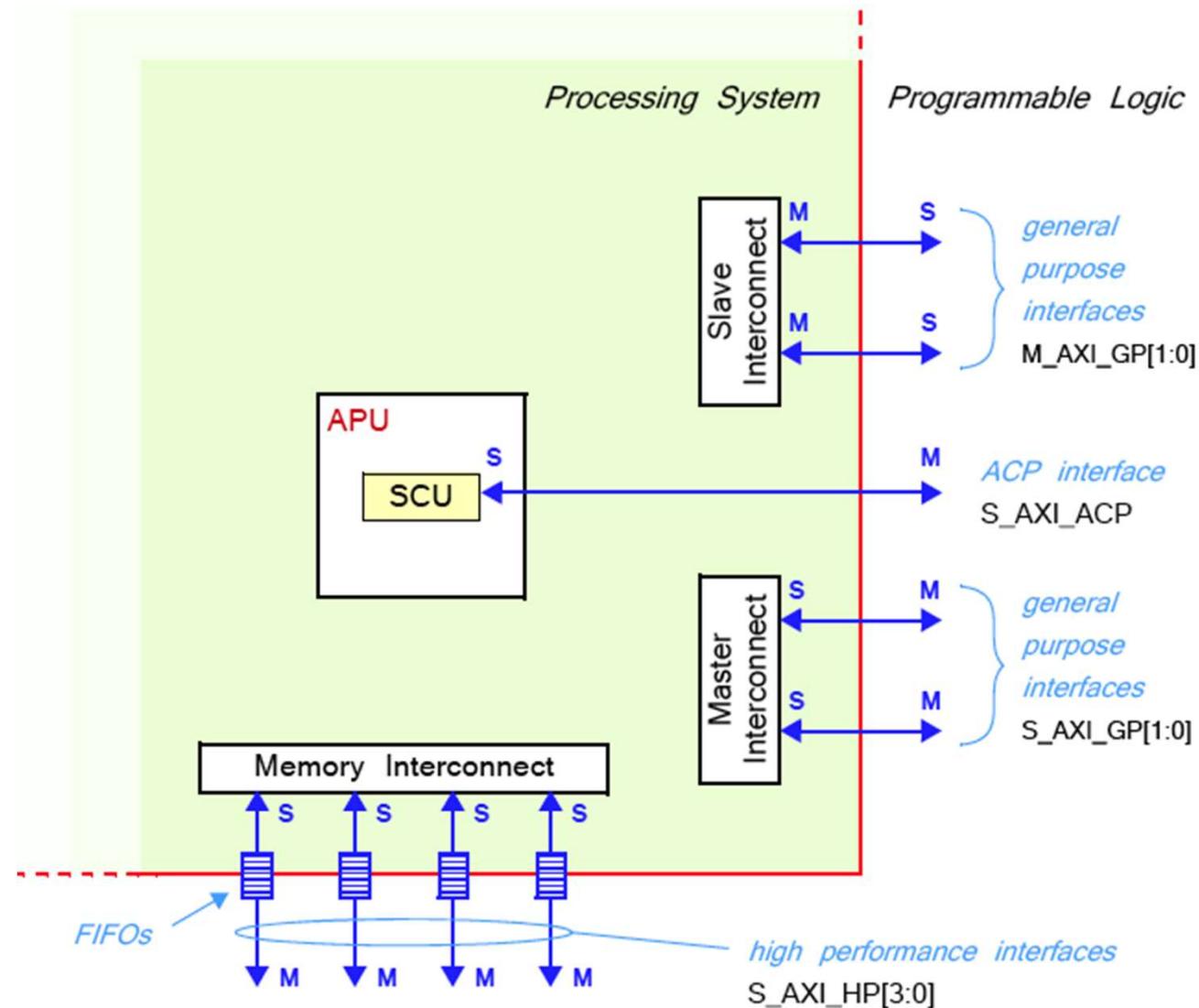


Programmable Logic (PL) BRAMs and DSP units



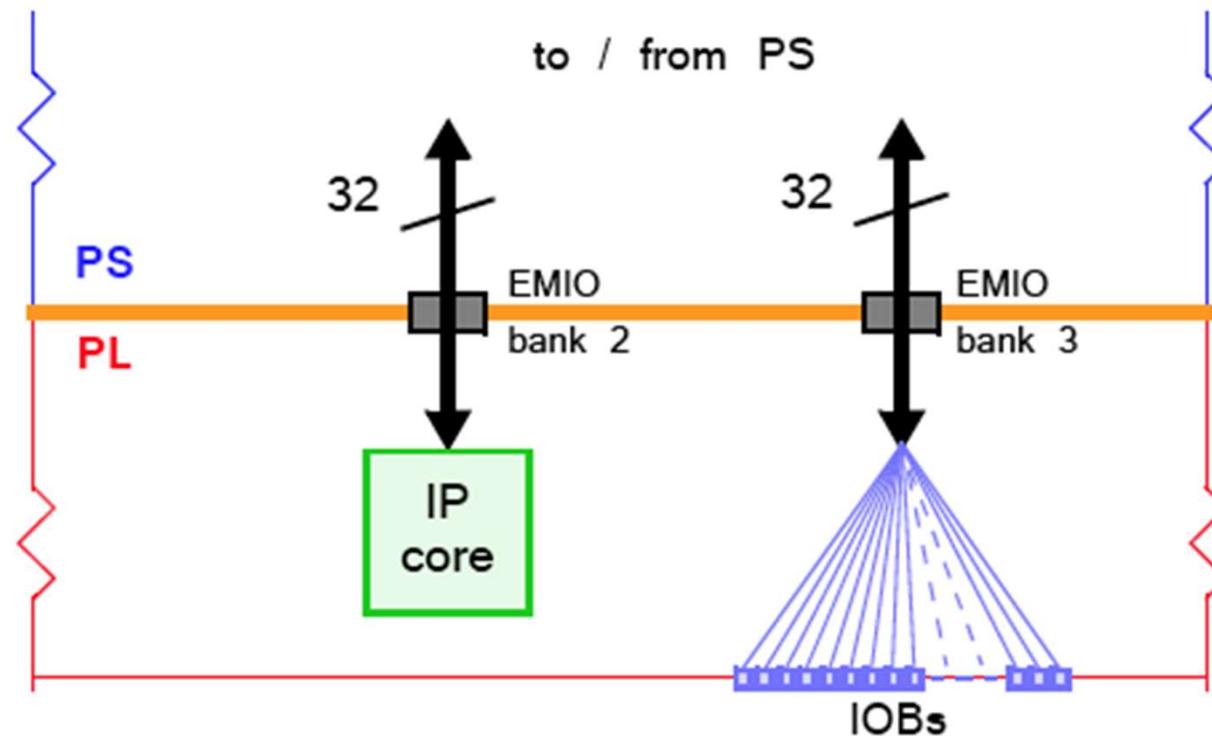
Source: The Zynq Book

AXI Interconnects and Interfaces



Source: The Zynq Book

Using Extended Multiplexed Input/Output (EMIO) to Interface Between PS and PL



Source: The Zynq Book

Automotive Applications

- ▶ Functions by embedded processing:
 - ABS: Anti-lock braking systems
 - ESP: Electronic stability control
 - Airbags
 - Efficient automatic gearboxes
 - Theft prevention with smart keys
 - Blind-angle alert systems
 - ... etc ...
- ▶ Multiple networks
 - Body, engine, telematics, media, safety
- ▶ Multiple processors
 - Up to 100
 - 8-bit – door locks, lights, etc.
 - 16-bit – most functions
 - 32-bit – engine control, airbags
 - Processing where the action is
 - Sensors and actuators distributed all over the vehicle
 - Networked together



Automotive Applications

Lane and Road Sign Recognition



© Xilinx



© Xilinx

Source: The Zynq Book

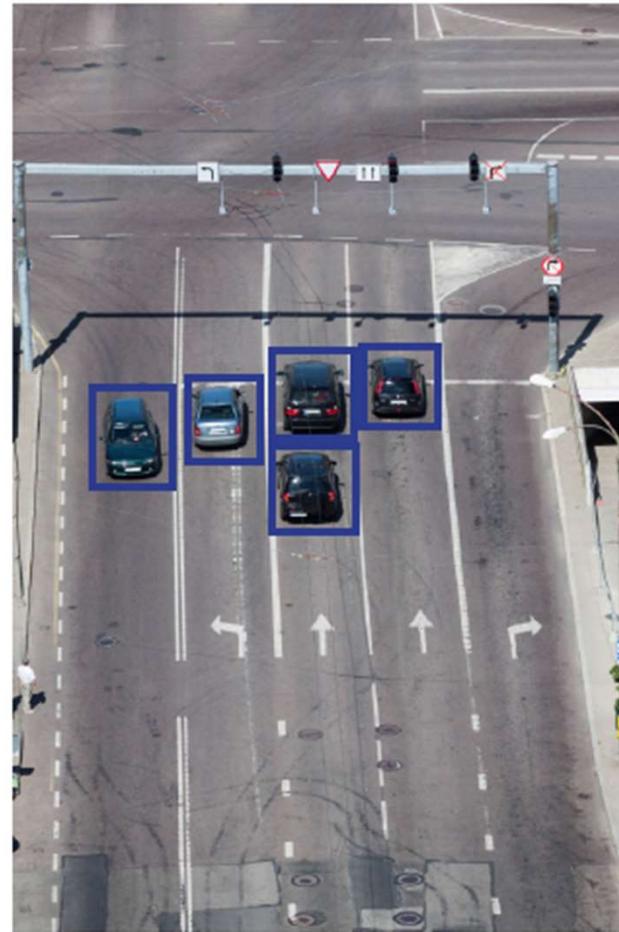
Computer Vision

Detection of Cars at a Junction

Original Image

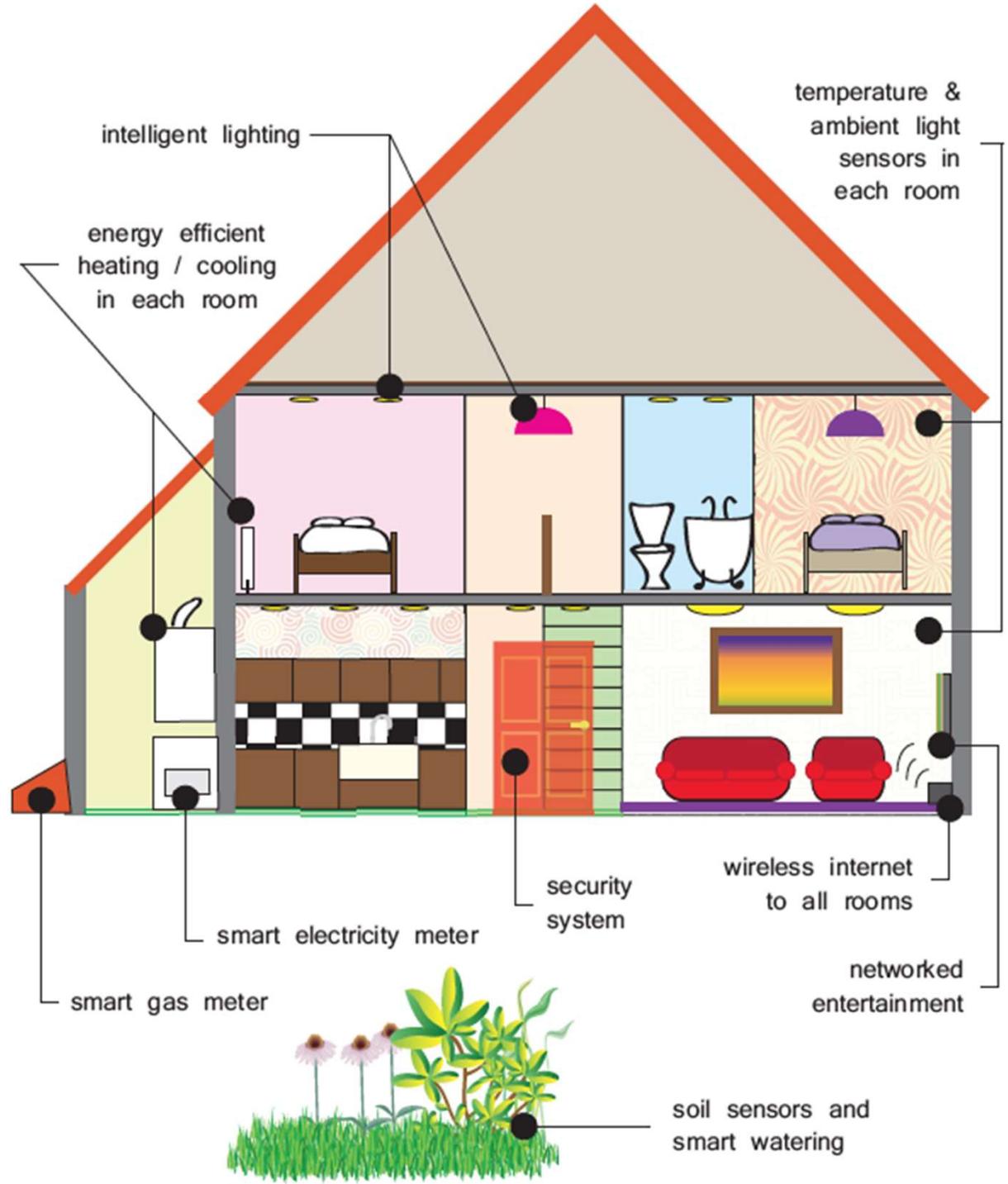


Cars Detected



Source: The Zynq Book

Smart Home

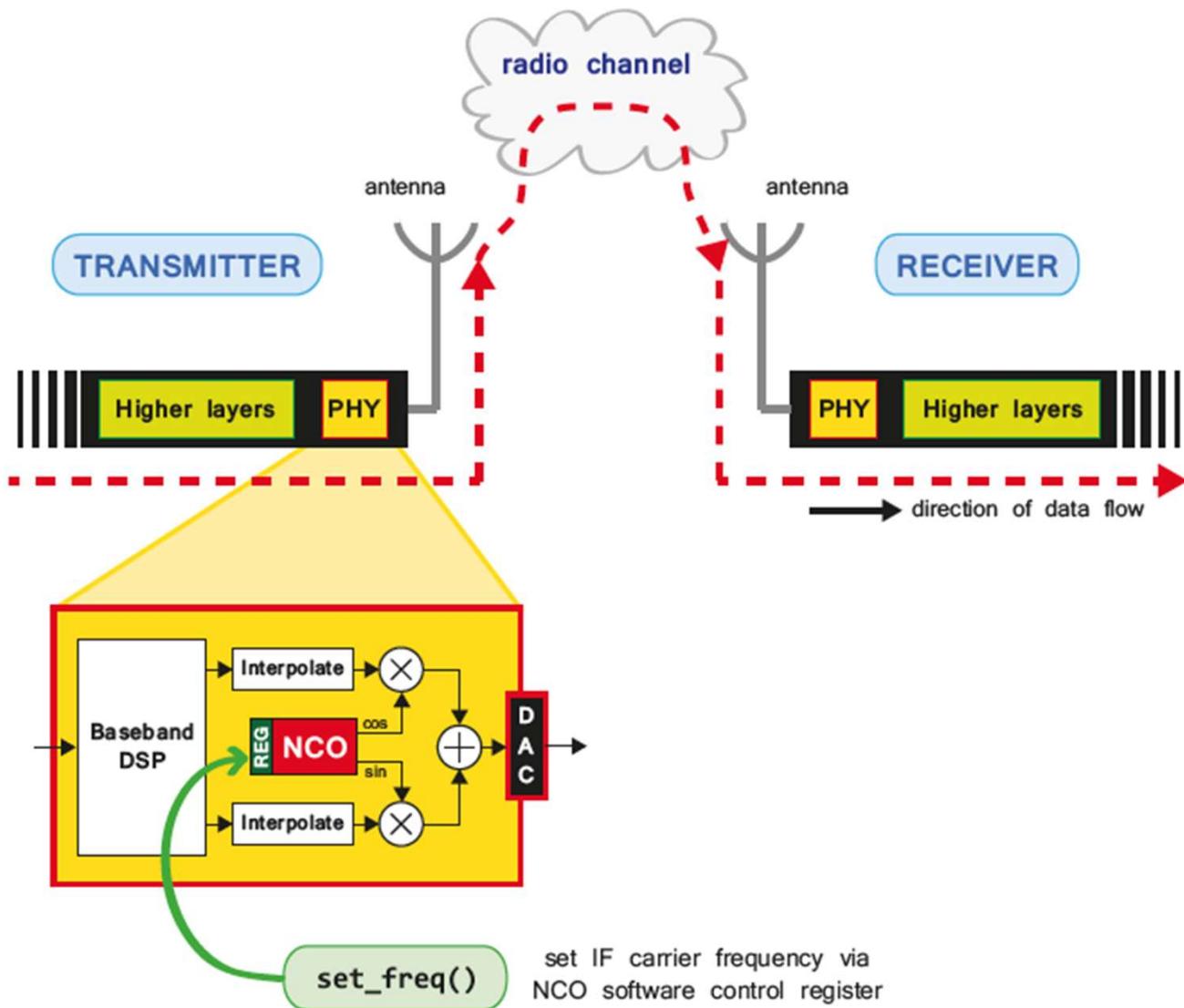


Source: The Zynq Book

Software Defined Radio (SDR)

- a radio which can be reconfigured while in operation
- all of the physical layer functions are software defined
- used initially in military applications (JTRS 1998), recently entering the commercial arena
- can support multiple radio standards (for cellular networks [2G, 3G, 4G], WiFi, Bluetooth, GPS reception, etc.)
- May be used in smartphones, tablets, e-readers, TVs, cars, transportation, emergency services, etc.)

Software Defined Radio (SDR)



Source: The Zynq Book

Software Defined Radio (SDR)

- The Physical Layer (PHY) – the part of radio directly adjacent to the Radio Frequency (RF) circuitry and air interface
- Computationally intensive, implementing high-speed filters, modulation, coding, DSP algorithms, support for ADC and DAC
- Most complex computations implemented in hardware (with parameters set from software)
- Less complex computations can be performed in either hardware or software

Cognitive Radio

- an intelligent radio that can be programmed and configured dynamically
- its transceiver is designed to use the best (under-used) wireless channels in its vicinity
- automatically detects available channels in wireless spectrum, and changes its transmission or reception parameters accordingly
- allows more concurrent wireless communications in a given spectrum band at one location
- a form of dynamic spectrum management

Communication Systems



© Xilinx



© Xilinx



© Xilinx

Wireless
Basestation

Satellite
Groundstation

Wired Network
Switches

Source: The Zynq Book

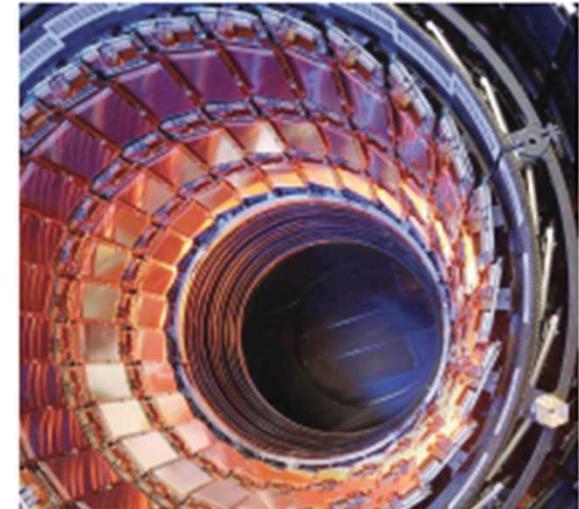
Control and Instrumentation Systems



© Xilinx



© Xilinx



© Xilinx

Industrial
Control
Room

Wind
Turbines

High Energy
Physics
Experiment

Source: The Zynq Book

Medical Applications



© Xilinx



© Xilinx

MRI Scanning

Robot Assisted Surgery

Source: The Zynq Book

Choice Among Various Implementation Platforms

	Total System Cost	Flexibility	Differentiation	Time-to-Market	Cost of Derivatives	Risk
Zynq SoC	Low + best value	Most flexible: HW and SW programmable + programmable I/O	Highest degree of programmability, HW/SW co-design	Fastest for integrated HW & SW differentiation	Lowest due to HW & SW programmability	Predictably low risk
ASSP + FPGA	Higher than Zynq SoC (system dependent)	Highly flexible but ASSP I/O limited compared to Zynq SoC	HW and SW programmable, ASSP-dependent	Fastest if ASSP requires HW differentiation	Low to high depending on FPGA vendor	Low to high depending on FPGA vendor
ASSP	Lowest if SW-only programmability is sufficient	Good but SW-programmable only	Limited to SW programmable only - easy cloning	Fastest if SW-only differentiation required	Lowest if SW-only derivatives needed	Can be Lowest if SW-only programmability is sufficient
ASIC	High to prohibitive	Once manufactured only limited SW flexibility	Best HW differentiation but limited SW differentiation	Lowest & riskiest	Highest	Terrible (respins)

Source: Xcell Journal, no. 88, Q3 2014

Advantages of Zynq

Lowest NRE, Best Risk Mitigation	Greatest Flexibility & Differentiation	Streamlined Productivity & Fast TTM	Lowest Cost of Derivatives & Highest Profitability
<ul style="list-style-type: none">✓ Already manufactured silicon✓ Negligible development & design tool costs✓ Xilinx IP library + third-party IP✓ Extensive development boards	<ul style="list-style-type: none">✓ All Programmable HW, SW & I/O✓ Anytime field programmable✓ Partial reconfiguration✓ System Secure (encryption)	<ul style="list-style-type: none">✓ Instant HW/SW co-development✓ All Programmable Abstractions (C, C++, OpenCV, OpenCL, HDL, model-based entry)✓ Vivado Design Suite, Vivado HLS, IP Integrator & UltraFast Methodology✓ Broad and open OS & IDE support (Open-source Linux & Android, FreeRTOS, Windows Embedded, Wind River, Green Hills, & many others)	<ul style="list-style-type: none">✓ IP standardized on ARM AMBA AXI4✓ Reuse precertified code (ISO, FCC, etc.)✓ Reuse & refine code & testbenches✓ Volume silicon, power circuitry, PCBs & IP licensing

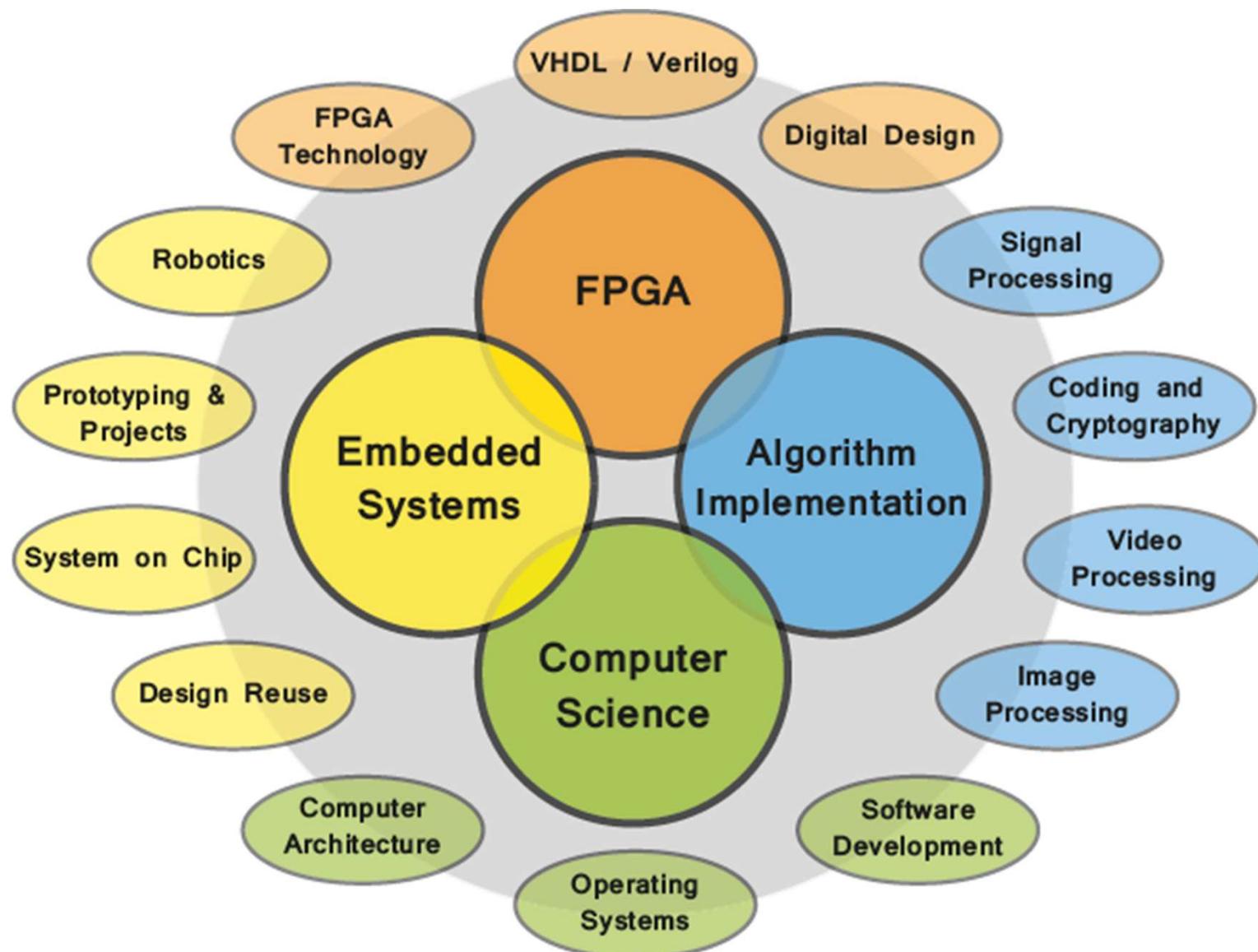
Source: Xcell Journal, no. 88, Q3 2014

Comparison of the Development Time & Cost

	28nm ASIC (IBS Data)			Zynq SoC (Xilinx Estimates)		
	%	Approximate Engineering Months	Total Cost (\$M)	%	Approximate Engineering Months	Total Cost (\$M)
Hardware						
IP qualification	26	704	11.8	20	240	4.0
Architecture	8	209	4.2	45	100	2.1
Verification	53	1431	28.9	35	160	3.0
Physical design	13	350	6.9	0	0	0
Subtotal hardware (design engineering resources)	100	2694	51.8	100	500	9.1
Software		4296	59.8		720	10.0
Prototype cost (\$M)	2.1			1.0		
Validation of prototypes		815	16.6		140	2.8
Total		7805	130.3		1360	22.9

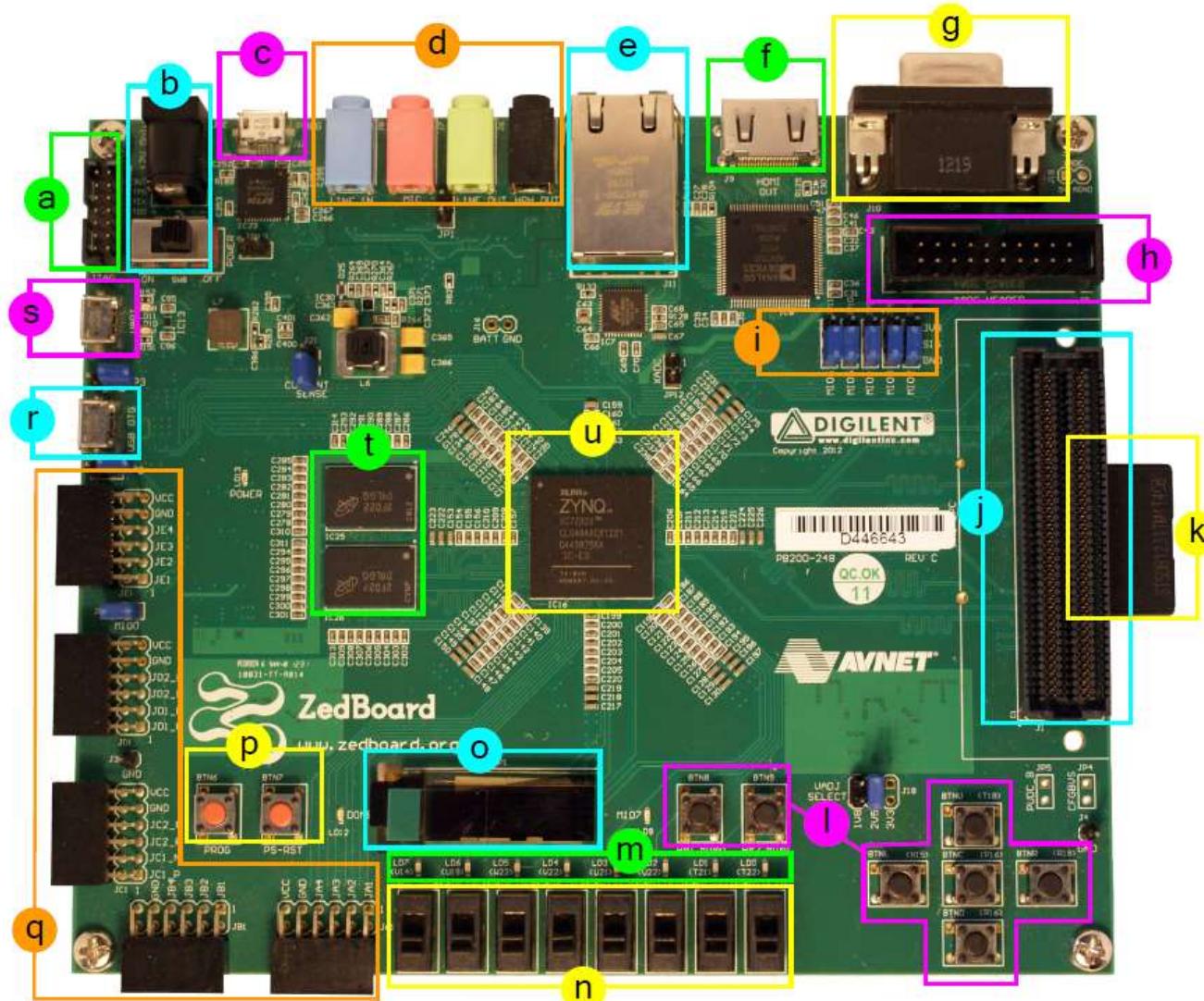
Source: Xcell Journal, no. 88, Q3 2014

Academic Subjects to which Zynq is Relevant



Source: The Zynq Book

The ZedBoard Development Board



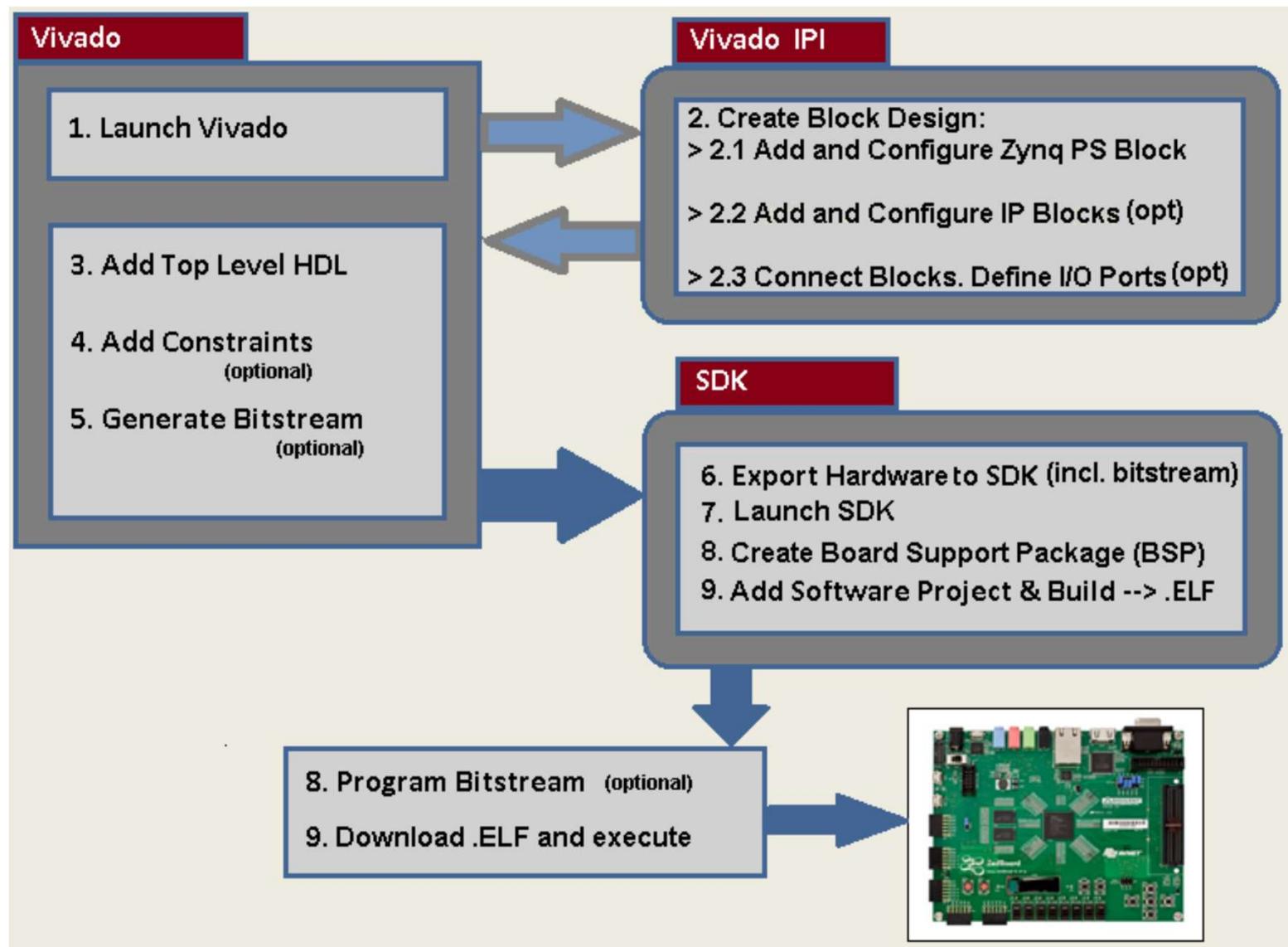
Source: ZedBoard Reference Manual

ZedBoard Components

- | | | | | | |
|----------|------------------------|----------|-----------------------|----------|---------------------------|
| a | Xilinx JTAG connector | h | XADC header port | o | OLED display |
| b | Power input and switch | i | Configuration jumpers | p | Prog & reset push buttons |
| c | USB-JTAG (programming) | j | FMC connector | q | 5 x Pmod connector ports |
| d | Audio ports | k | SD card (underside) | r | USB-OTG peripheral port |
| e | Ethernet port | l | User push buttons | s | USB-UART port |
| f | HDMI port (output) | m | LEDs | t | DDR3 memory |
| g | VGA port | n | Switches | u | Zynq device (+ heatsink) |

Source: ZedBoard Reference Manual

Vivado Design Flow for Zynq



Source: ZedBoard Reference Manual