Inclusive Language Commitment

- *Arm is committed to making the language we use inclusive, meaningful, and respectful. Our goal is to remove and replace non-inclusive language from our vocabulary to reflect our values and represent our global ecosystem.*
- *Arm is working actively with our partners, standards bodies, and the wider ecosystem to adopt a consistent approach to the use of inclusive language and to eradicate and replace offensive terms. We recognise that this will take time. This course may contain references to non-inclusive language; it will be updated with newer terms as those terms are agreed and ratified with the wider community. We recognise that some of you will be accustomed to using the previous terms and may not immediately recognise their replacements. Please refer to the following examples:*
  - *When introducing the AMBA AXI Protocols, we will use the term 'Manager' instead of 'Master' and 'Subordinate' instead of 'Slave'.*
  - *When introducing the architecture, we will use the term 'Requester' instead of 'Master' and 'Completer' instead of 'Slave'.*
- *Contact us at education@arm.com with questions or comments about this course. You can also report non-inclusive and offensive terminology usage in Arm content at terms@arm.com.*

Introduction to Arm-based System on Chip Design

Learning Outcomes

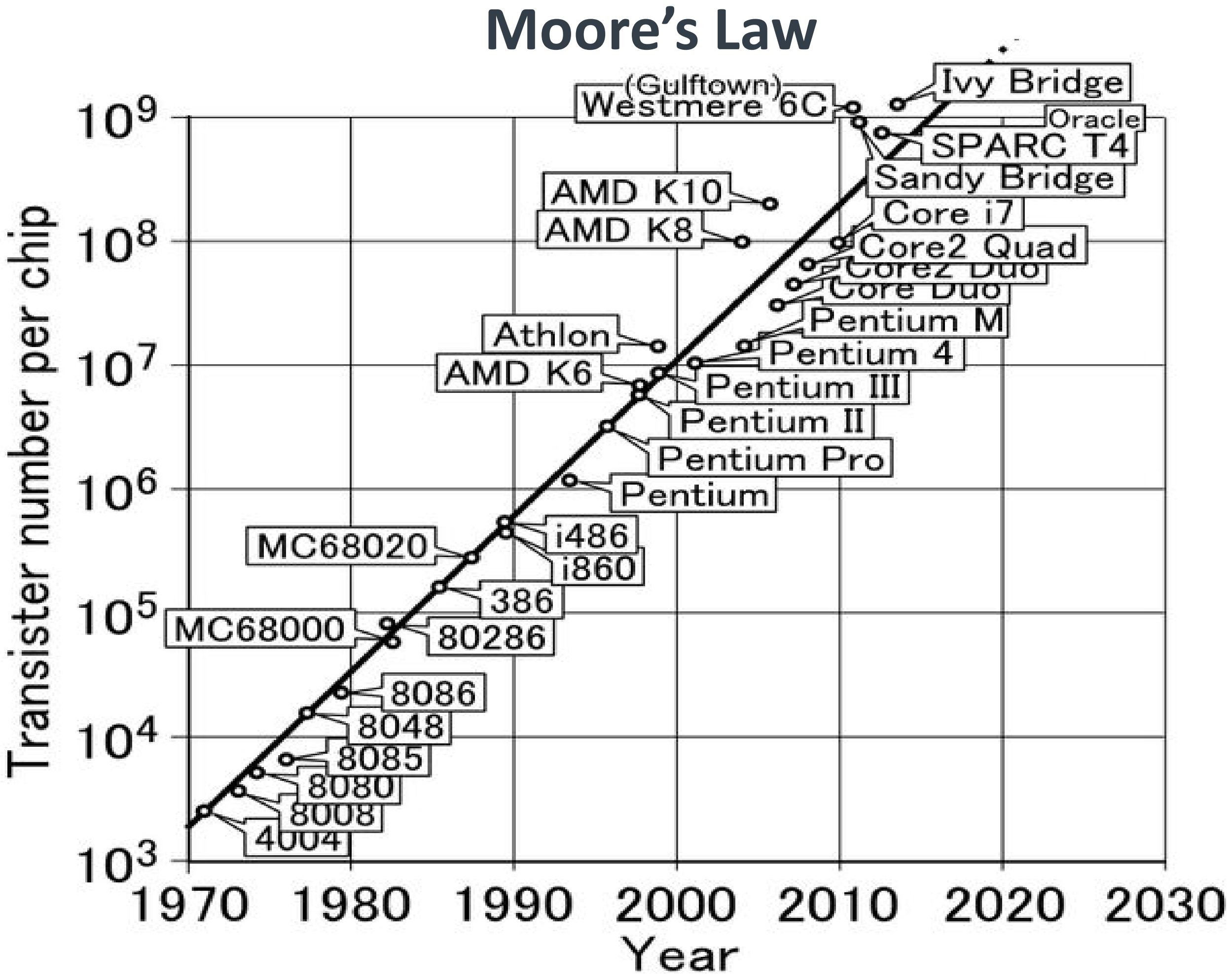
At the end of this module, you will be able to:

- Explain the motivations for the development of a System on Chip (SoC).
- Define what an SoC is and its characteristics.
- Outline the advantages and limitations of SoCs.
- Describe the main steps in an SoC design flow.
- Define what a Programmable SoC (PSoC) and its characteristics.

Why the SoC Design Concept Developed

- We are living in a post-PC era, with:
  - Smartphones and tablets
  - The Internet of Things, wearable computing, and cyber-physical systems
  - Industry 4.0
- The silicon transistor is still at the heart of this revolution.
- The primary metrics of silicon chips have changed: from clock-frequency to cost, form-factor, and power.
- On-chip integration of functional hardware is now more important than ever.
- How and why have we reached this point?

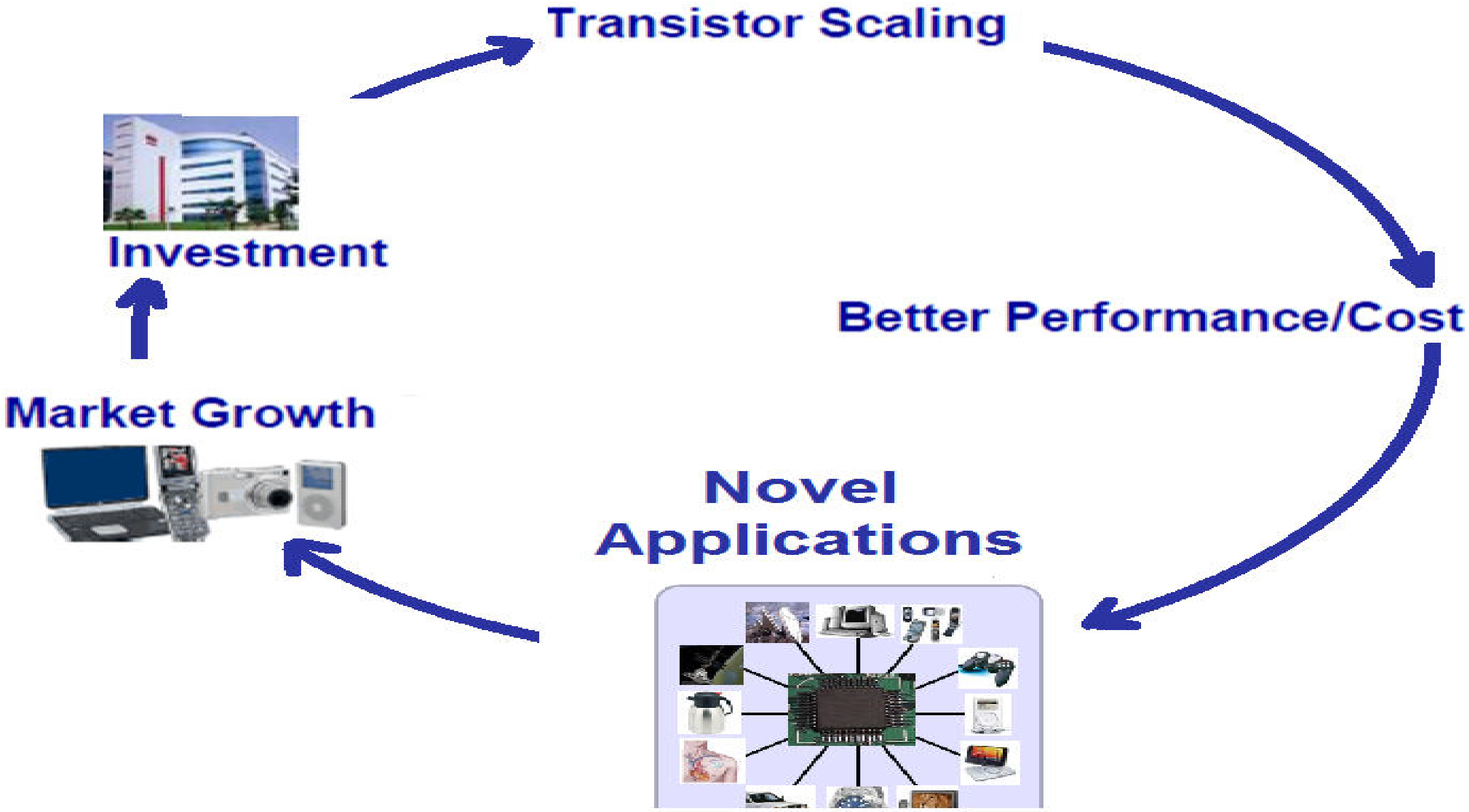
Moore’s Law



Prediction of Moore’s Law(*)

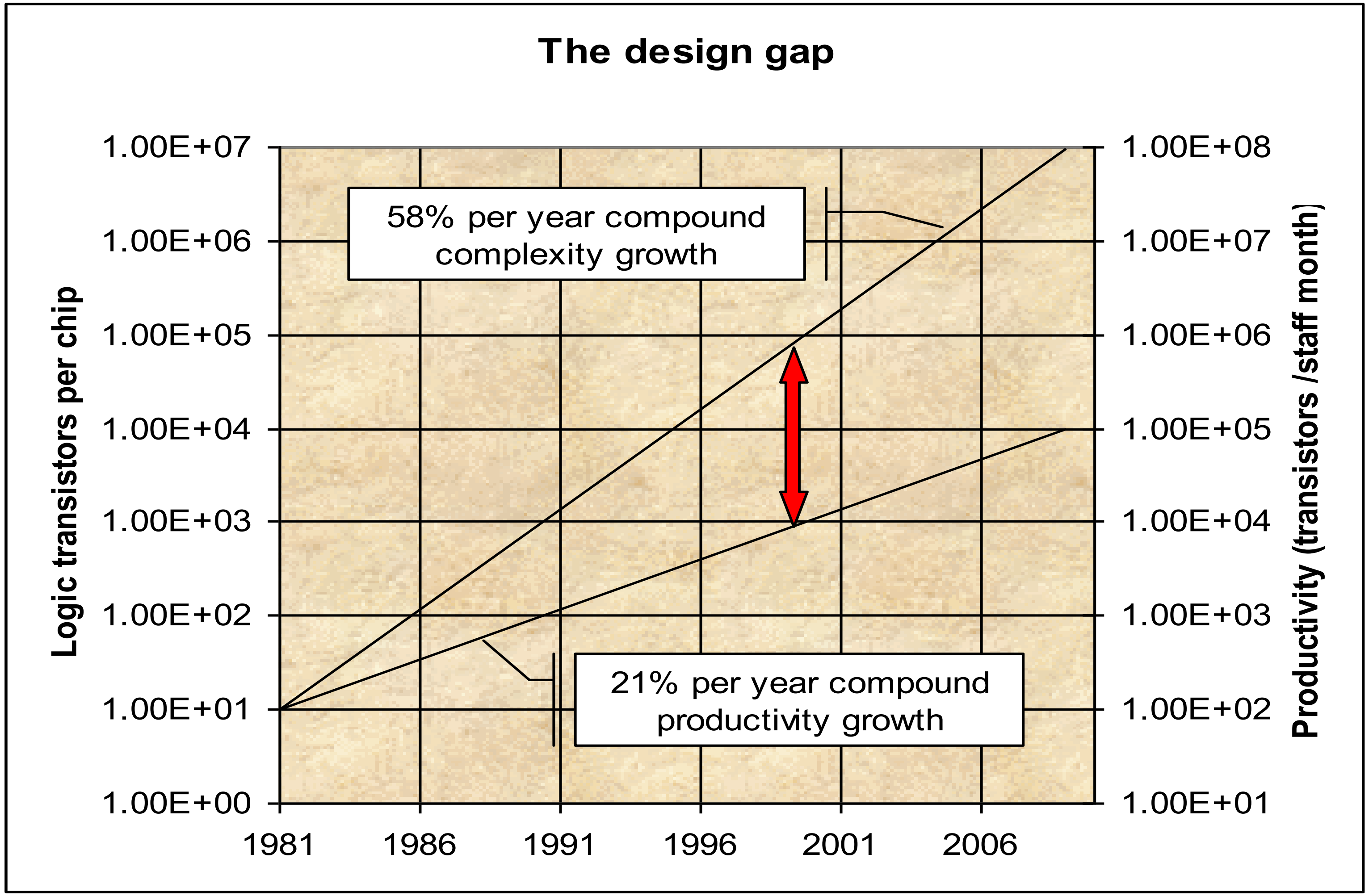
(*) Data are based on international semiconductor technology road map (http://www.itrs.net/)

Why Scaling?



The virtuous circle of the semiconductor industry

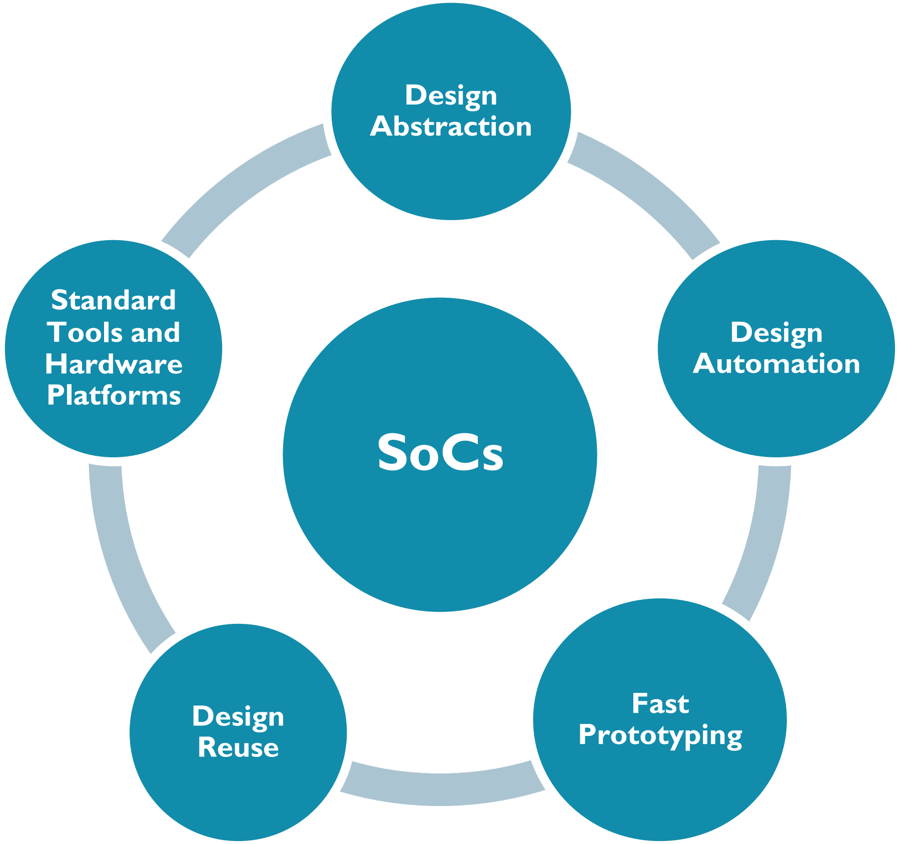
The Design Productivity Gap



Complexity outpaces design productivity

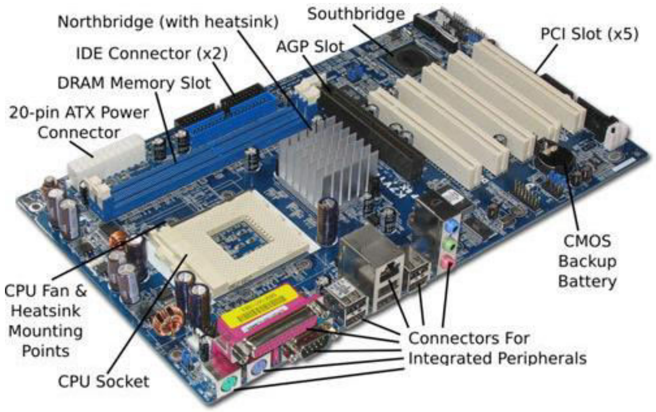
Bridging the Design Productivity Gap

- Several strategies exist to reduce the design productivity gap exist, namely:

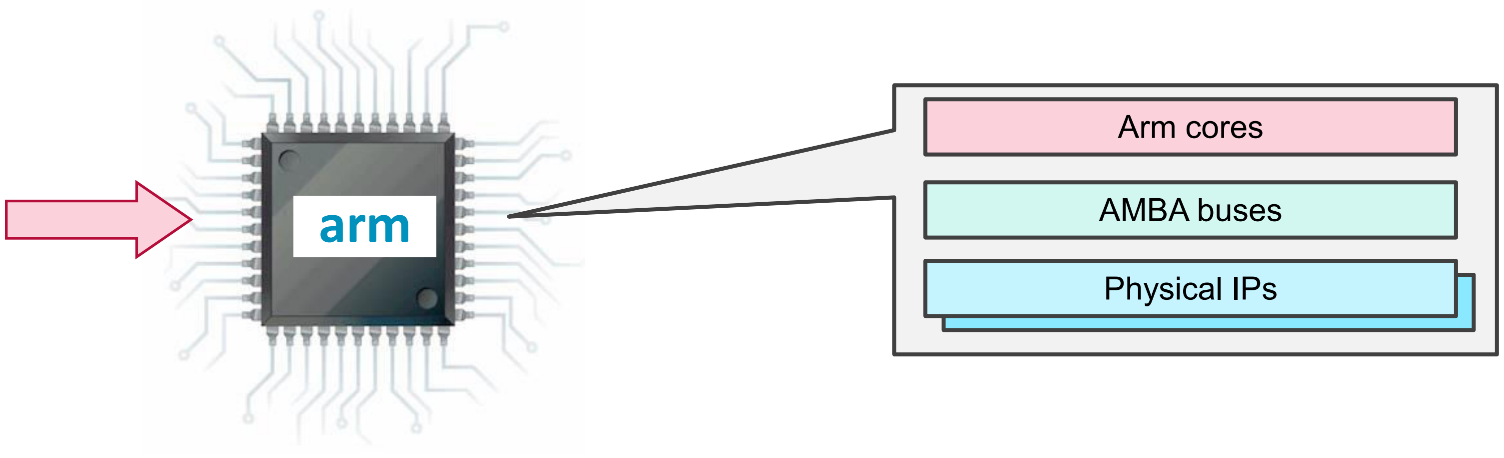


What Is an SoC?

- An SoC is an integrated circuit that packages basic computing components into a single chip.
- An SoC may have most or even all of the components to power a computer.



Motherboard of a PC

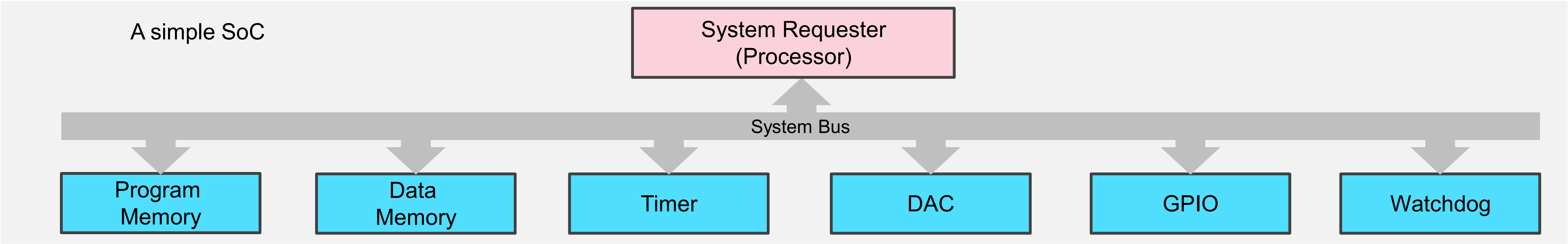


SoC

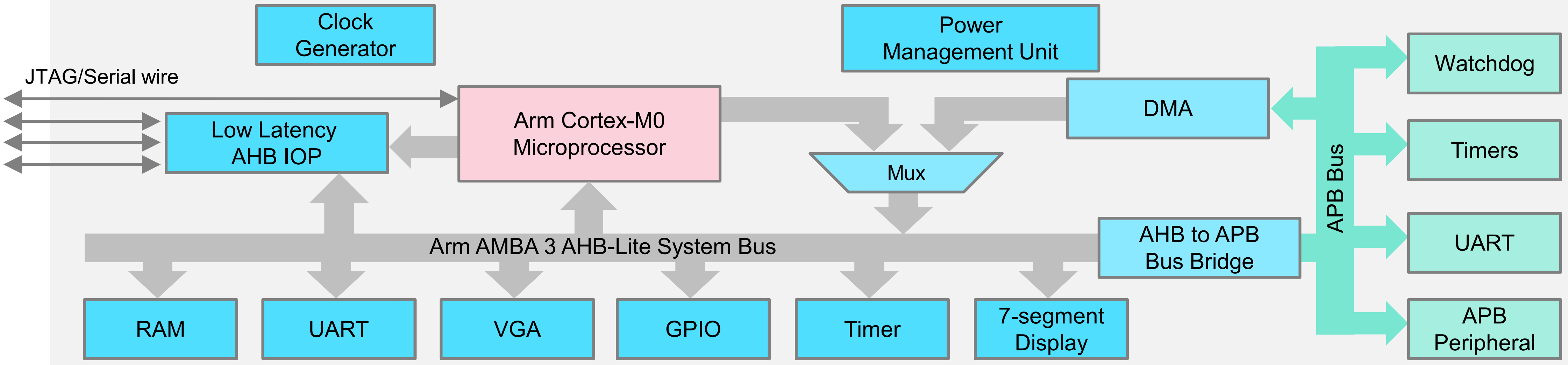
Picture source: http://thecustomizewindows.com/, http://www.adafruit.com/

What Is Inside an SoC?

- The basic components of an SoC include:
  - A system Requester, such as a microprocessor or DSP
  - System peripherals, such as memory blocks, timers, and external digital/analog interfaces
  - A system bus that connects Requester and peripherals using a specific bus protocol
- More sophisticated s are integrated in modern SoCs, such as multicores, DSPs, GPUs, and multiple buses connected by bus bridges.

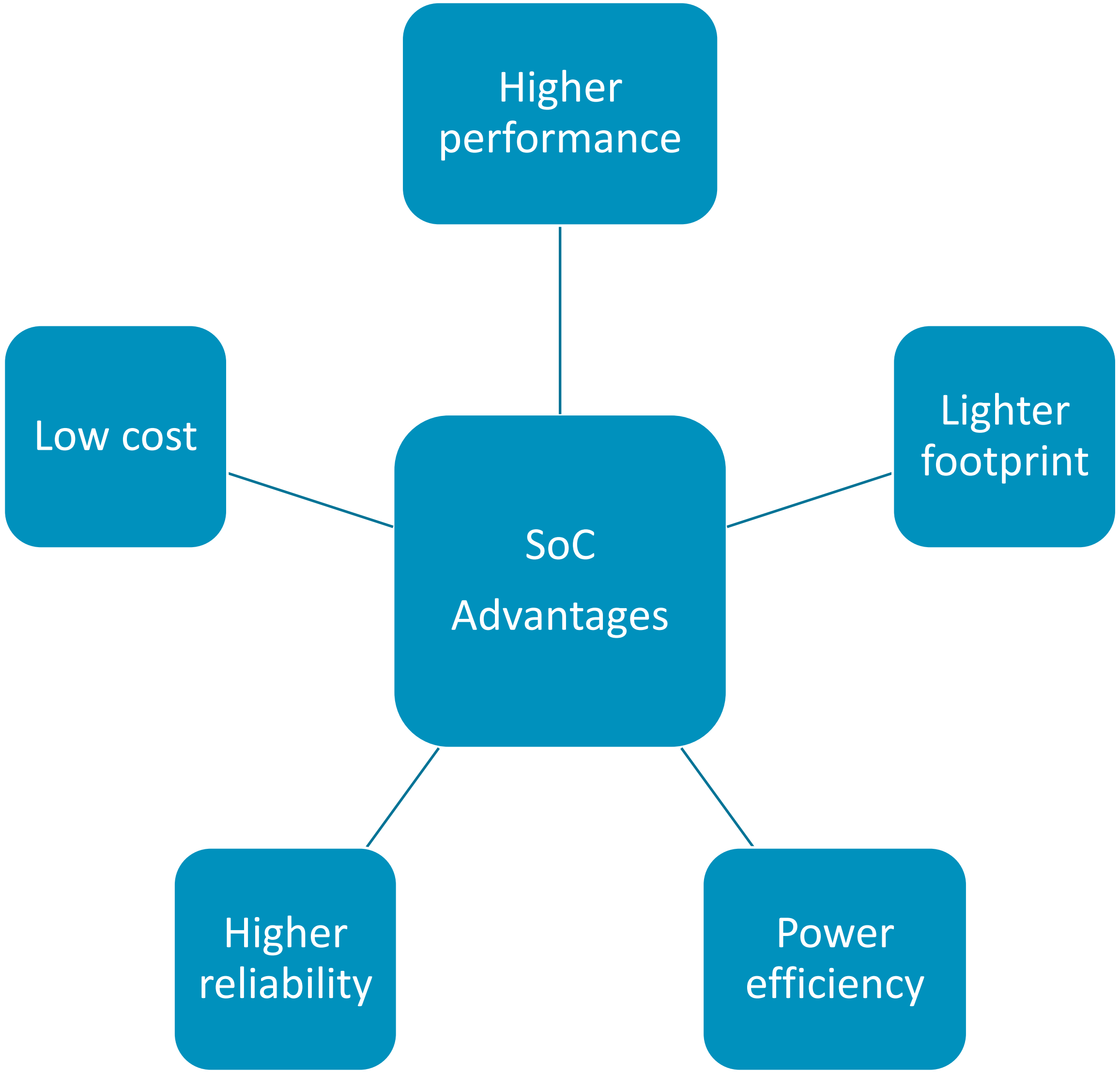


Example Arm-based SoC

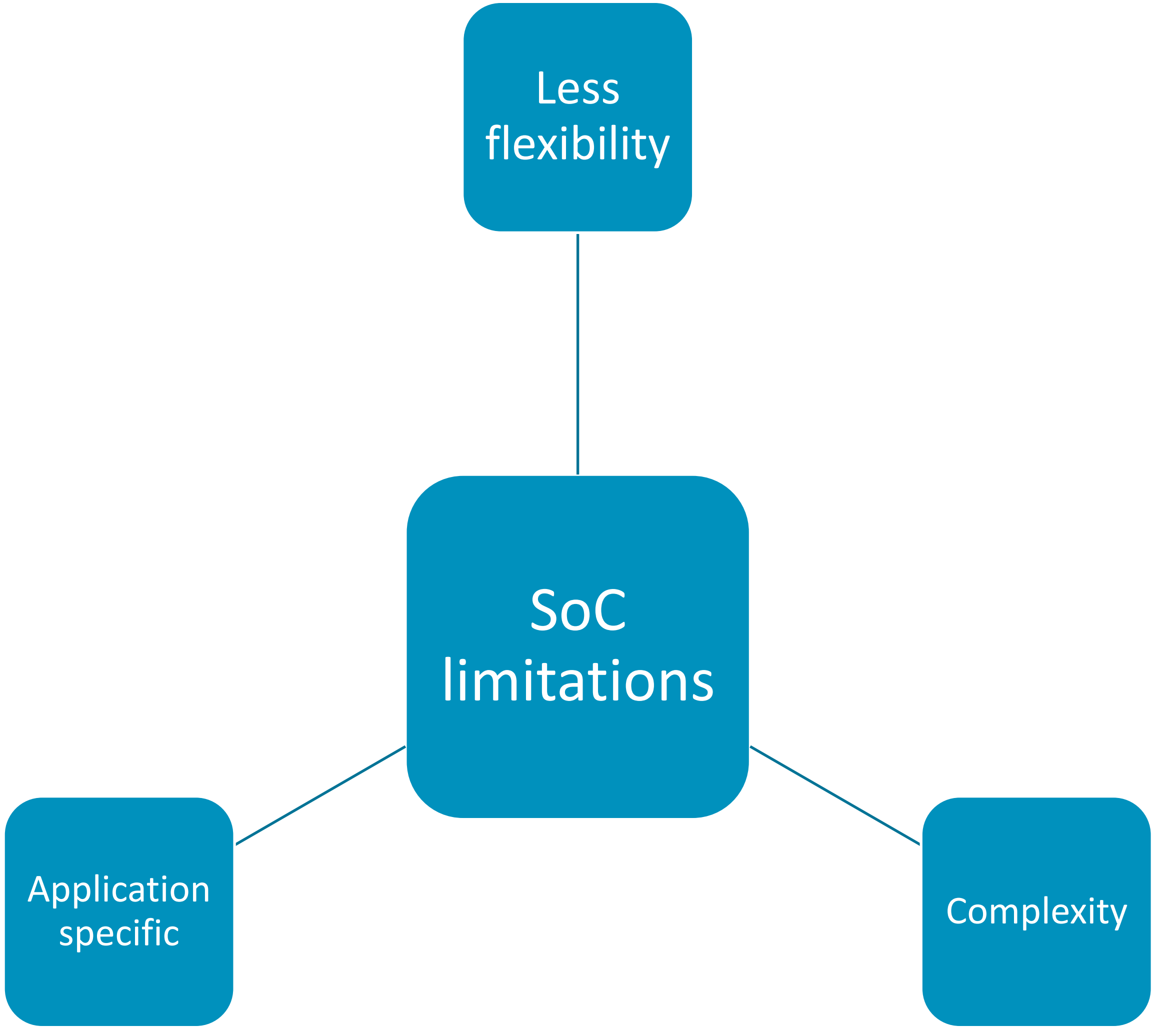


An example of an Arm-based SoC

Advantages of SoCs



Limitations of SoCs



SoC v Microcontroller v Processor

SoC

- Can have a single or multiple processor cores
- Has larger memory blocks, a variety of IOs, and other peripherals
- Integrated with more powerful blocks, e.g., GPU, DSP
- Capable of running OSs
- Mainly used for advanced applications (e.g., smartphones, tablets).

CPU

- Is a single processor core
- Used for general purposes
- It needs to be supported with memories and IOs

MCU

- Typically has a single processor core
- Has memory blocks, basic IOs, and other basic peripherals
- Mainly used for basic control purposes, such as embedded applications

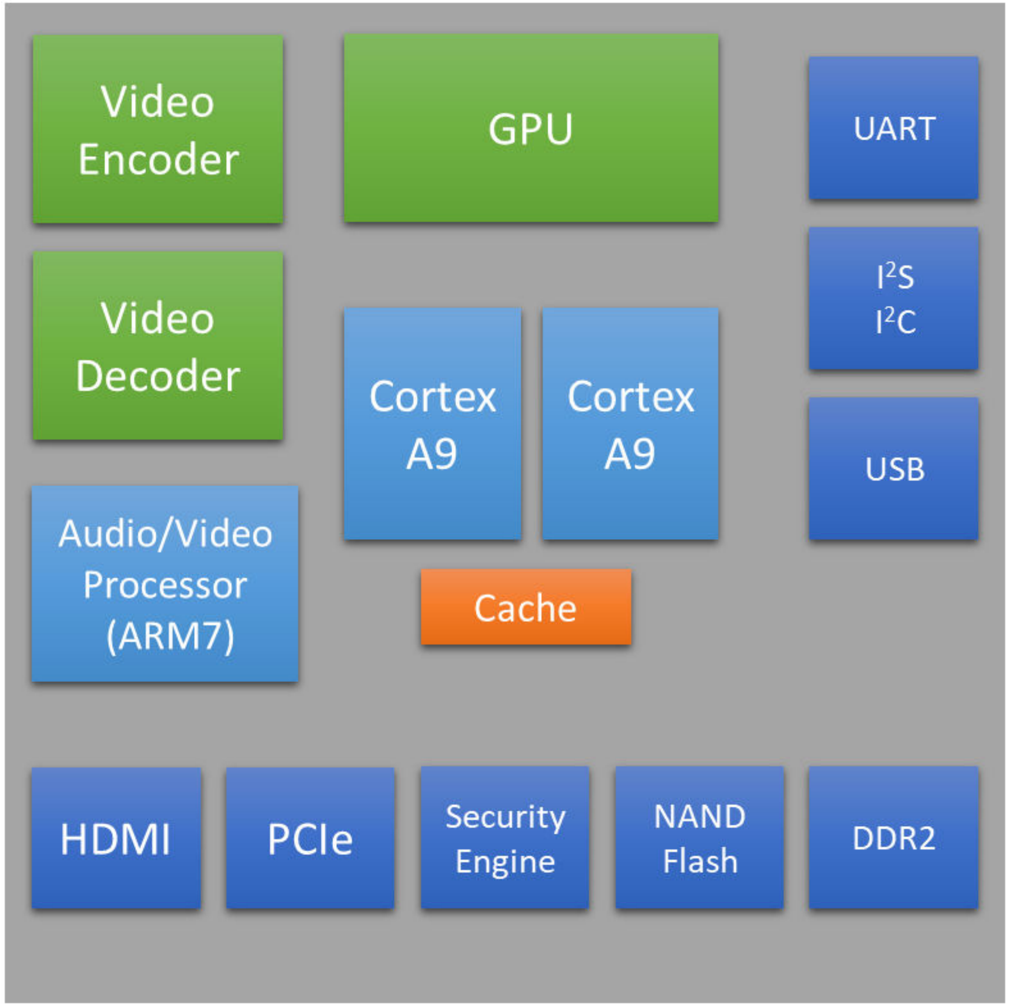
Commercialized SoCs

- Benefiting from its power efficiency, SoCs have been widely used in mobile devices, such as smartphones, tablets, and digital cameras.
- A number of SoCs have been developed by a large ecosystem of design companies:
  - Snapdragon by Qualcomm
  - Tegra by Nvidia
  - OMAP by Texas Instruments

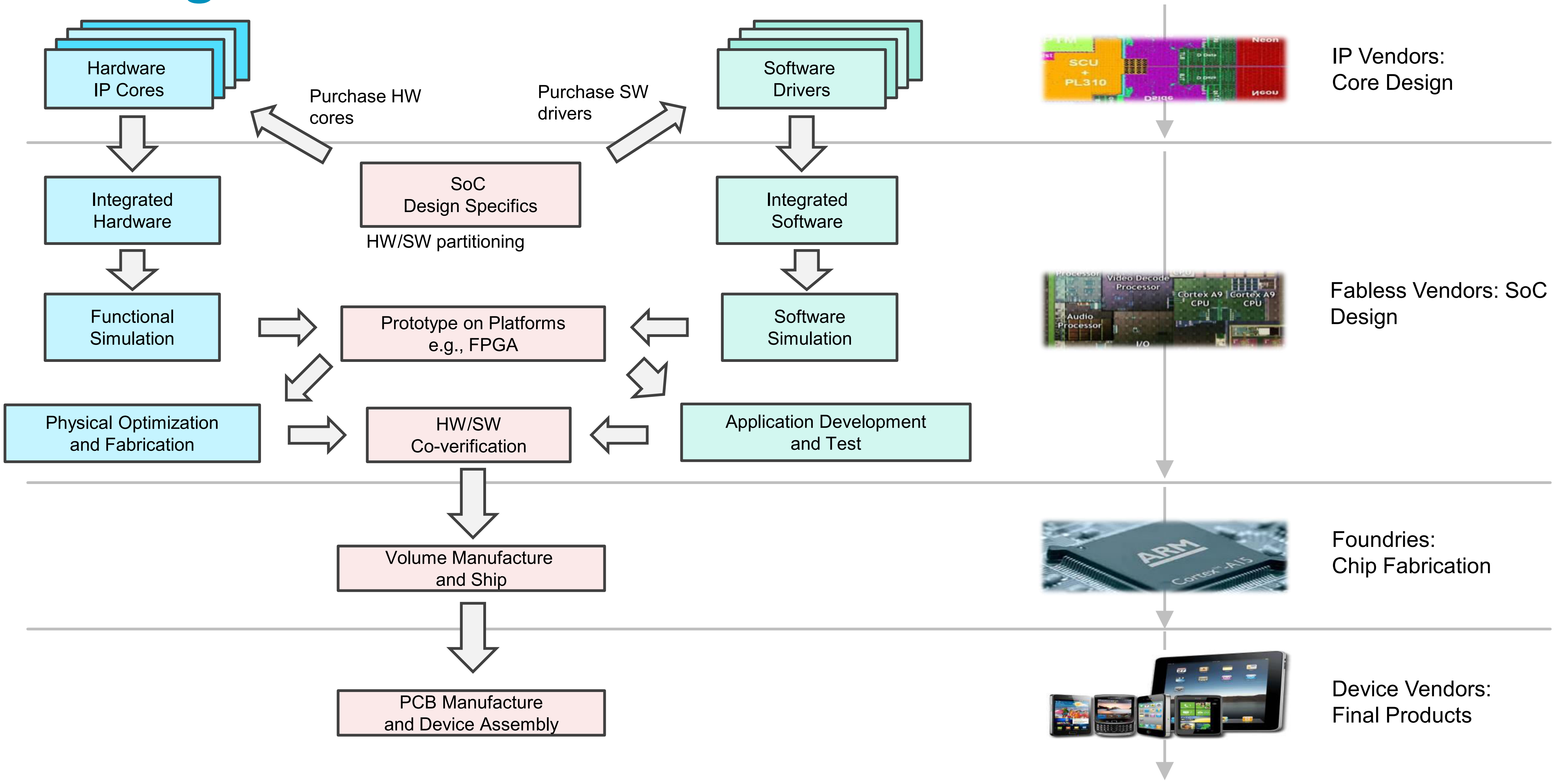
Most mobile SoCs use Arm-based microprocessors since they deliver high performance with less power consumption.

SoC Example: NVIDIA Tegra 2

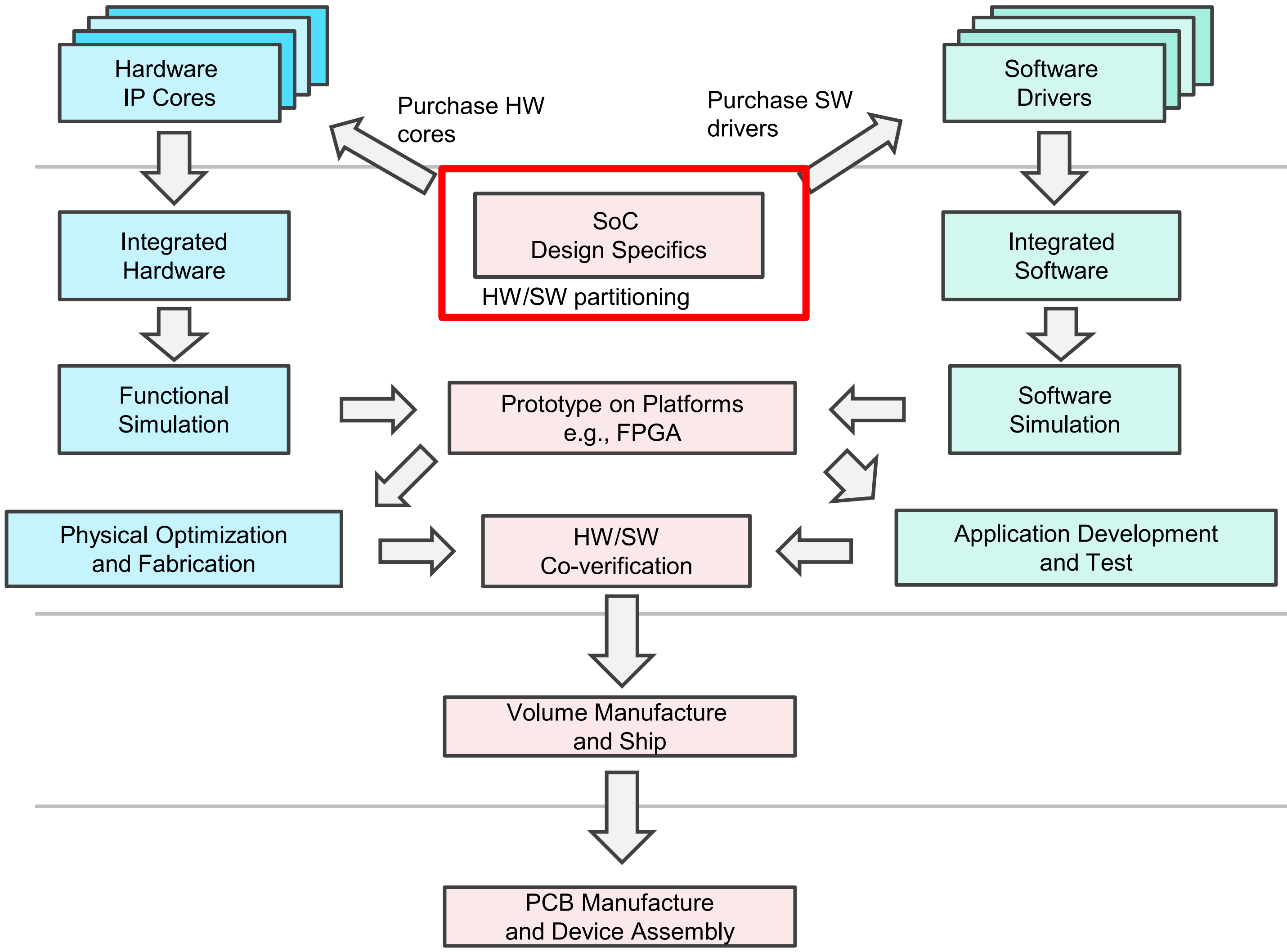
Designer	NVIDIA
Year	2010
Processor	Arm Cortex-A9 (dual-core)
Frequency	Up to 1.2 GHz
Memory	1 GB 667 MHz LP-DDR2
Graphics	ULP GeForce
Process	40 nm
Package	12 × 12 mm (Package on Package)
Used in tablets	Acer Iconia Tab A500 Asus Eee Pad Transformer Motorola Xoom Motorola Xoom Family Edition Samsung Galaxy Tab 10.1 Toshiba Thrive

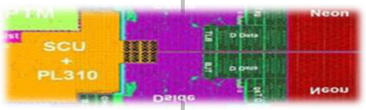


SoC Design Flow



SoC Design Flow





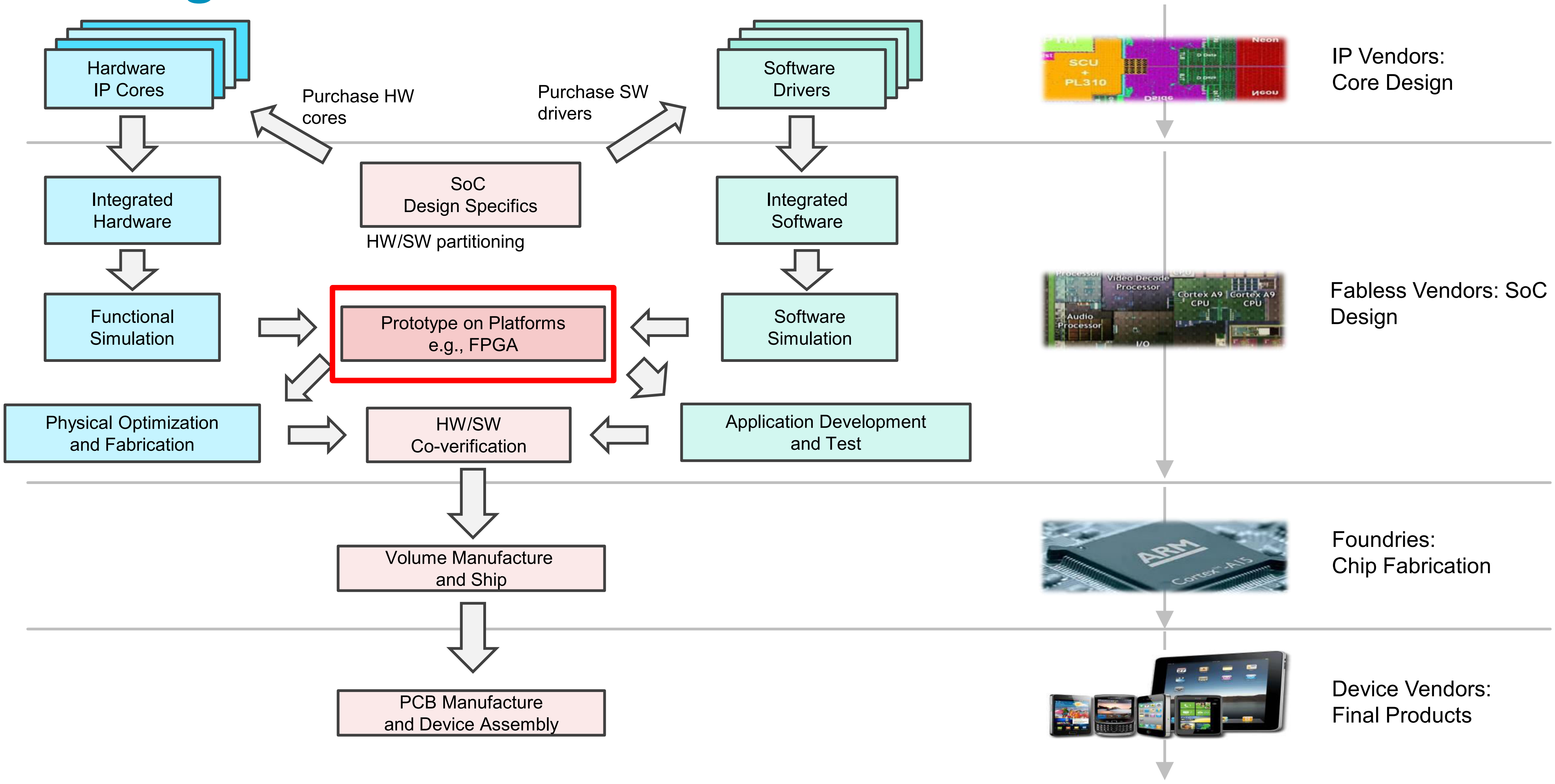
IP Vendors:
Core Design

Fabless Vendors: SoC
Design

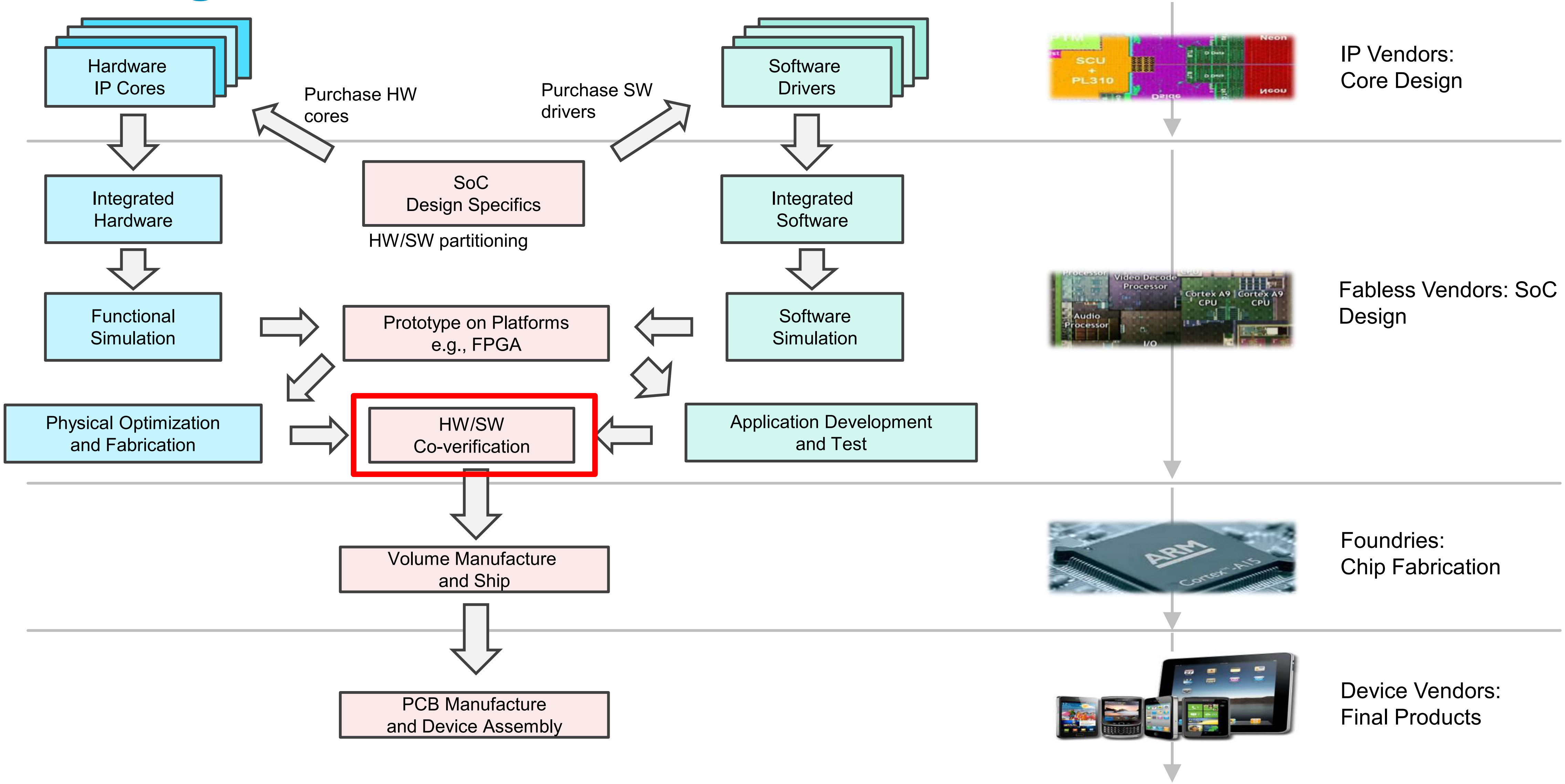
Foundries:
Chip Fabrication

Device Vendors:
Final Products

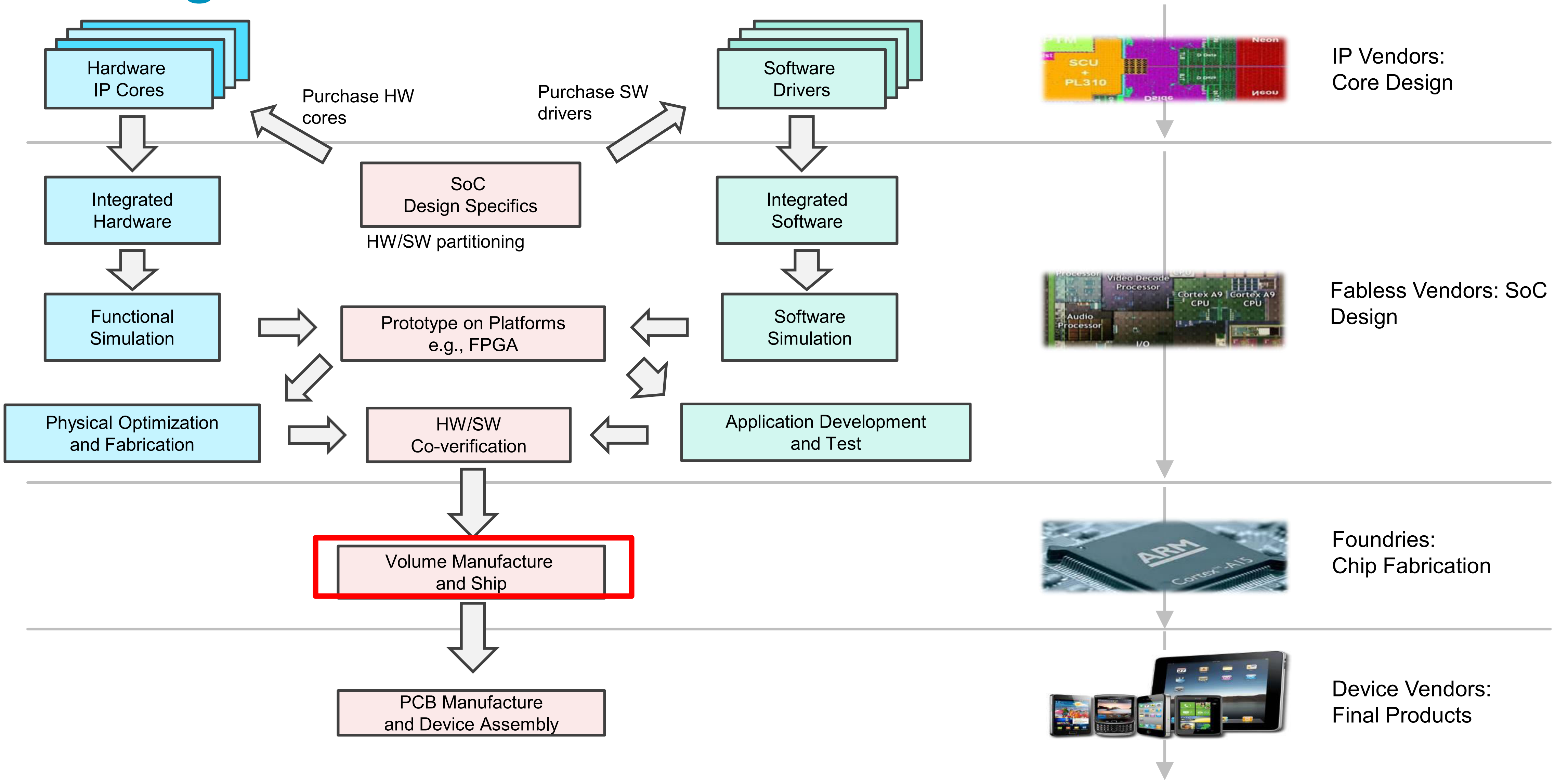
SoC Design Flow



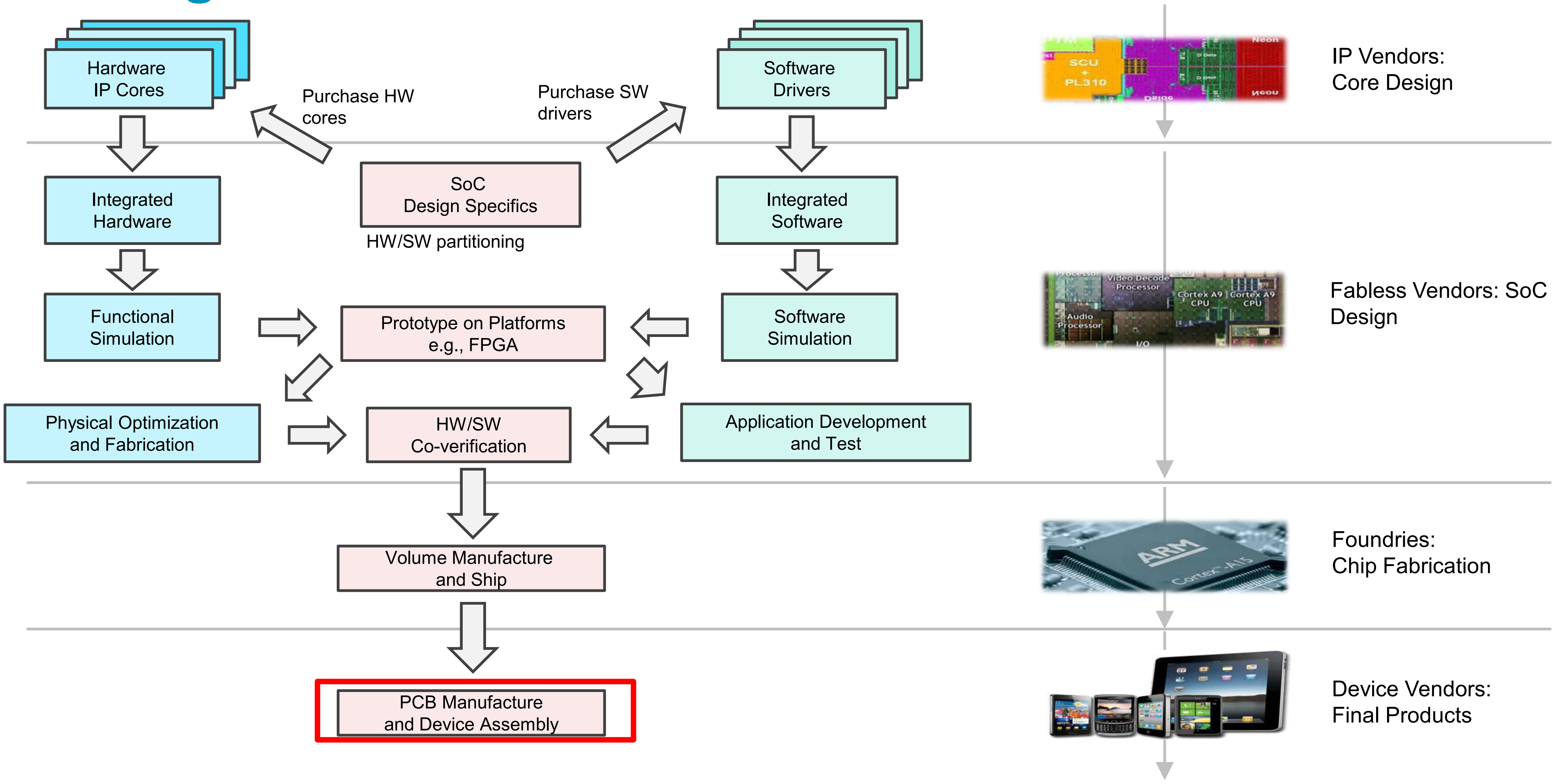
SoC Design Flow



SoC Design Flow



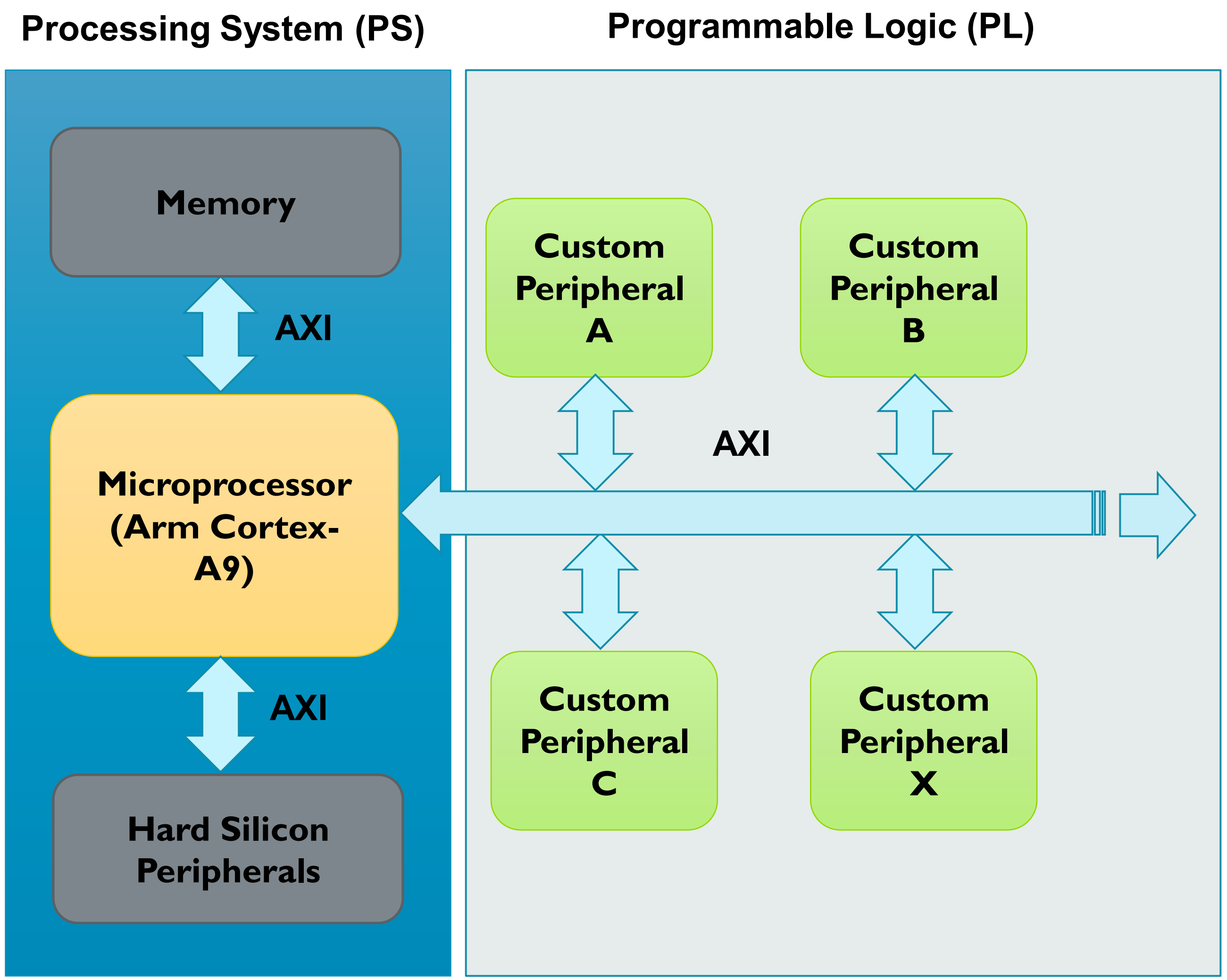
SoC Design Flow



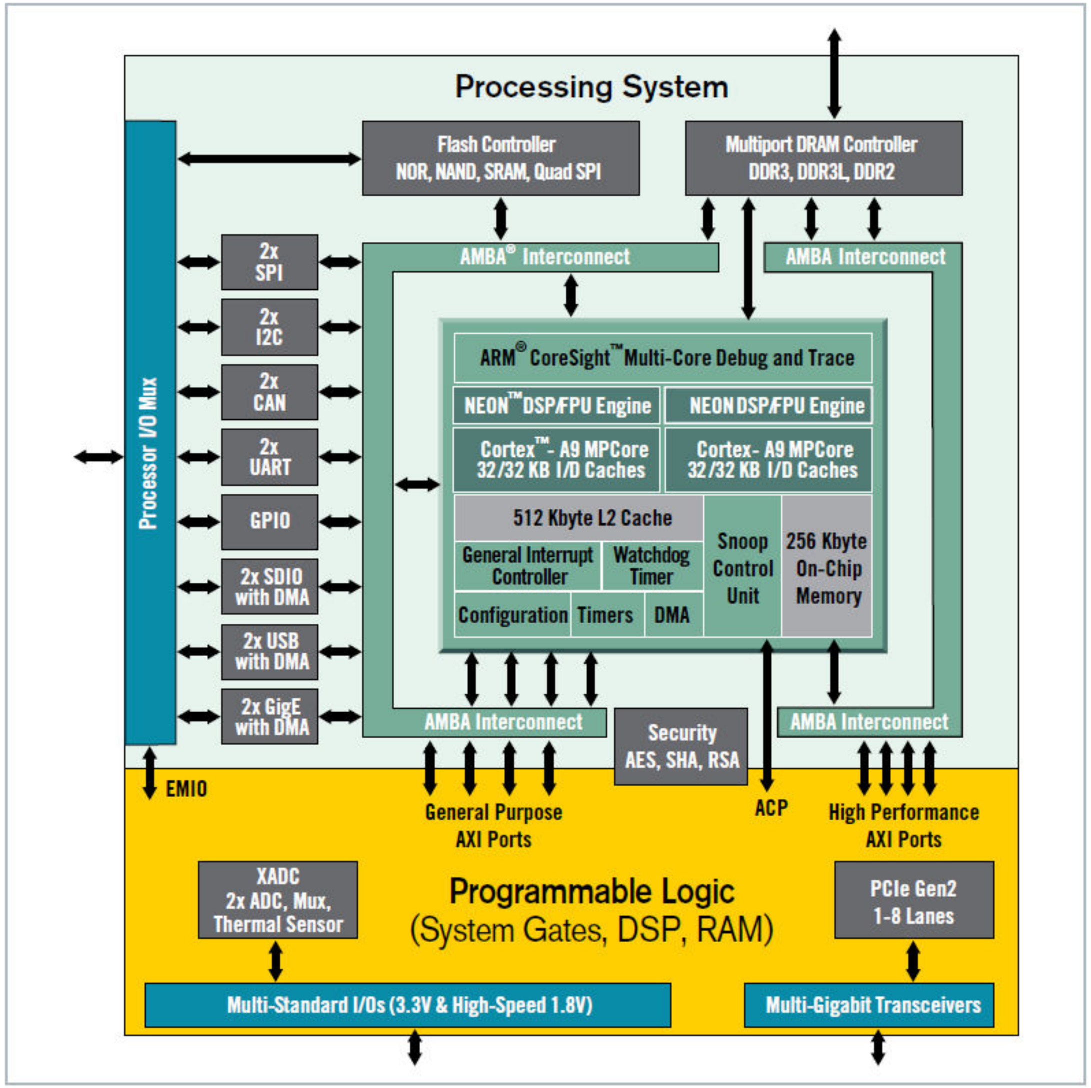
Programmable SoC

- SoCs can be prototyped and tested on FPGAs.
- Two options:
  - Use soft cores to embed a processor in the logic fabric (soft processor) and use device interconnect resources to implement a bus that communicates with other custom design blocks.
  - Use modern programmable SoCs (PSoCs; e.g., Xilinx Zynq), which include hard processors (e.g., Arm) connected to peripherals and to the logic fabric through a bus (e.g., AXI bus).
- PSoCs can overcome ASIC SoC limitations in some application areas by providing:
  - Flexibility for upgrading and functionality modification
  - Faster time-to-market for low to medium production volumes

Architecture of a PSoC



Example: Xilinx Zynq-7000

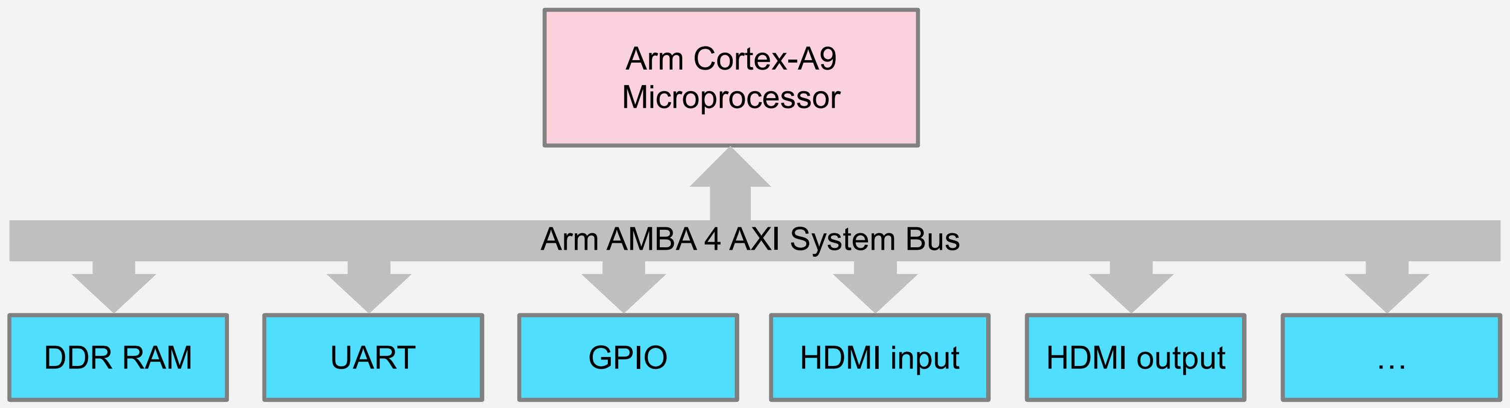


Picture source: http://www.xilinx.com/publications/prod_mktg/zynq-7000-generation-ahead-backgrounder.pdf

- Dual-core Arm Cortex-A9 processor
- On-chip memory
- Memory interfaces
- Integrated peripherals: timers, USB, UART, I2C, SPI
- AXI buses and AXI ports
- Programmable logic

Design a Simple Arm-based SoC

- Design Arm-based SoCs and prototype them onto a Zynq chip.
- The SoCs will consist of:
  - An Arm Cortex-A9 microprocessor
  - An AXI bus
  - Different customer-made physical IPs



A simplified Arm-based SoC