System-on-Chip *SoC*

Dongbing Gu

School of Computer Science and Electronic Engineering
University of Essex
UK

Spring 2017

Outline

1. WHAT
2. WHY
3. HOW
4. Platform based Design

Section 1

WHAT

What is System-on-Chip (SoC)

People A: The VLSI manufacturing technology advances have made possible to put millions of transistors on a single chip. It enables designers to move everything from the board onto the chip eventually.

People B: SoC is a high performance microprocessor, since you can program and give instructions to the microprocessor to do whatever you want to do.

People C: SoC is the efforts to integrate heterogeneous or different types of silicon IPs on to the same chip, like memory, microprocessor, ASIC logics, and analogue circuitry.

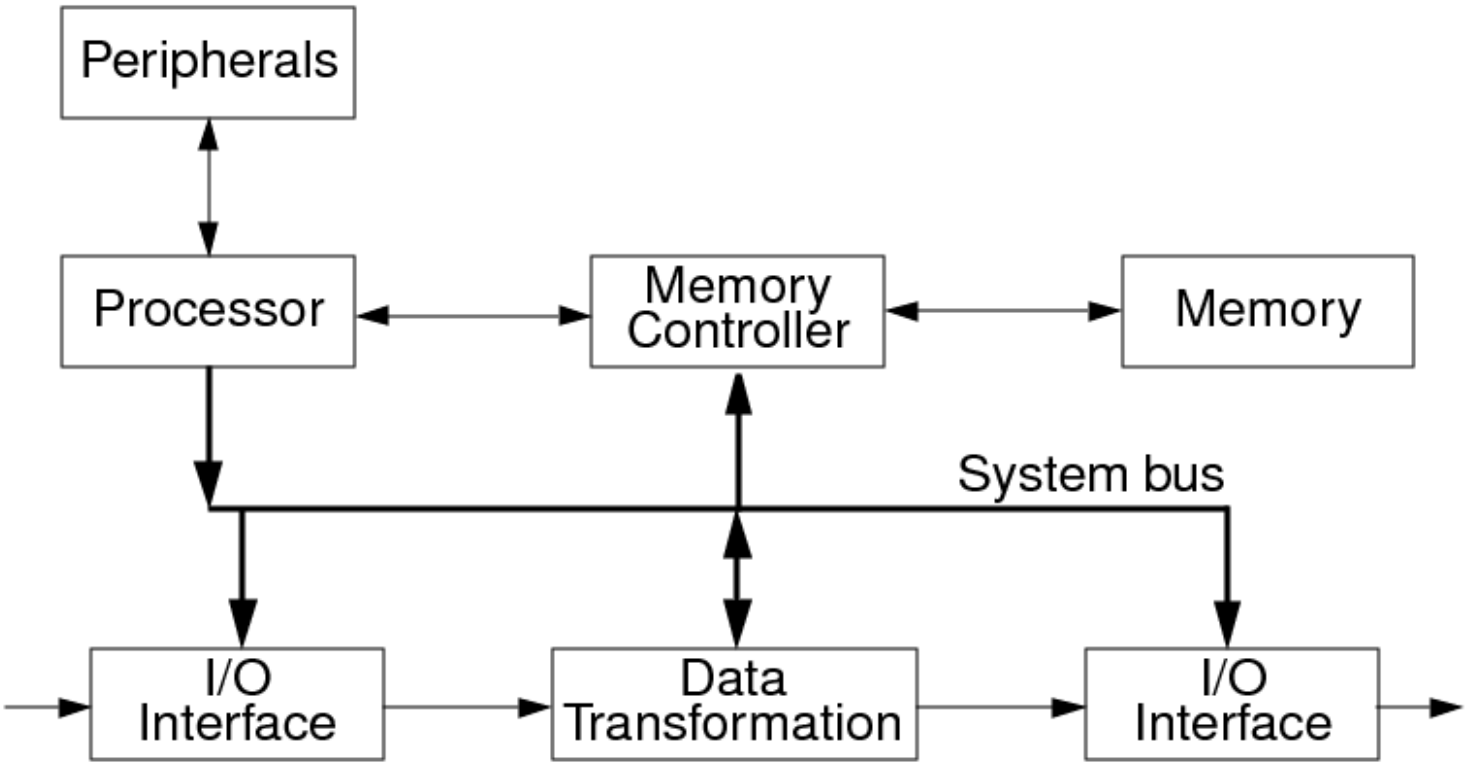
(All of the above are partially right, but not very accurate.)

What is System-on-Chip (SoC)

- SoC not only chip, but more on "system". SoC = Chip + Software + Integration
- The SoC chip includes embedded processor, ASIC logics and analogue circuitry, embedded memory.
- The SoC software includes OS, compiler, simulator, firmware, driver, protocol stack, integrated development environment (debugger, linker, In-Circuit Emulator (ICE)), application interface (C/C++, assembly)
- The SoC integration includes the whole system solution, manufacture consultant, technical supporting

Typical SoC

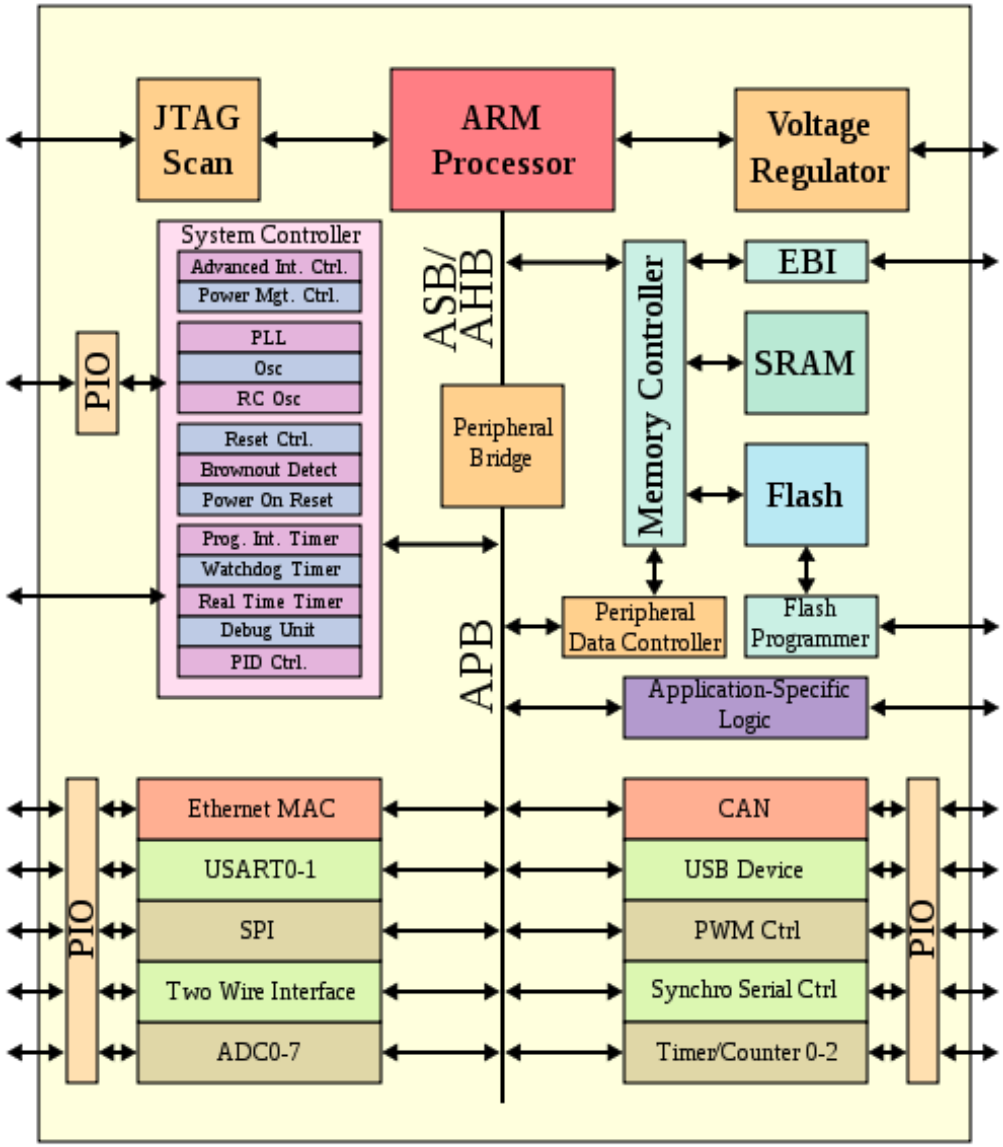
- A generic design:



Typical SoC

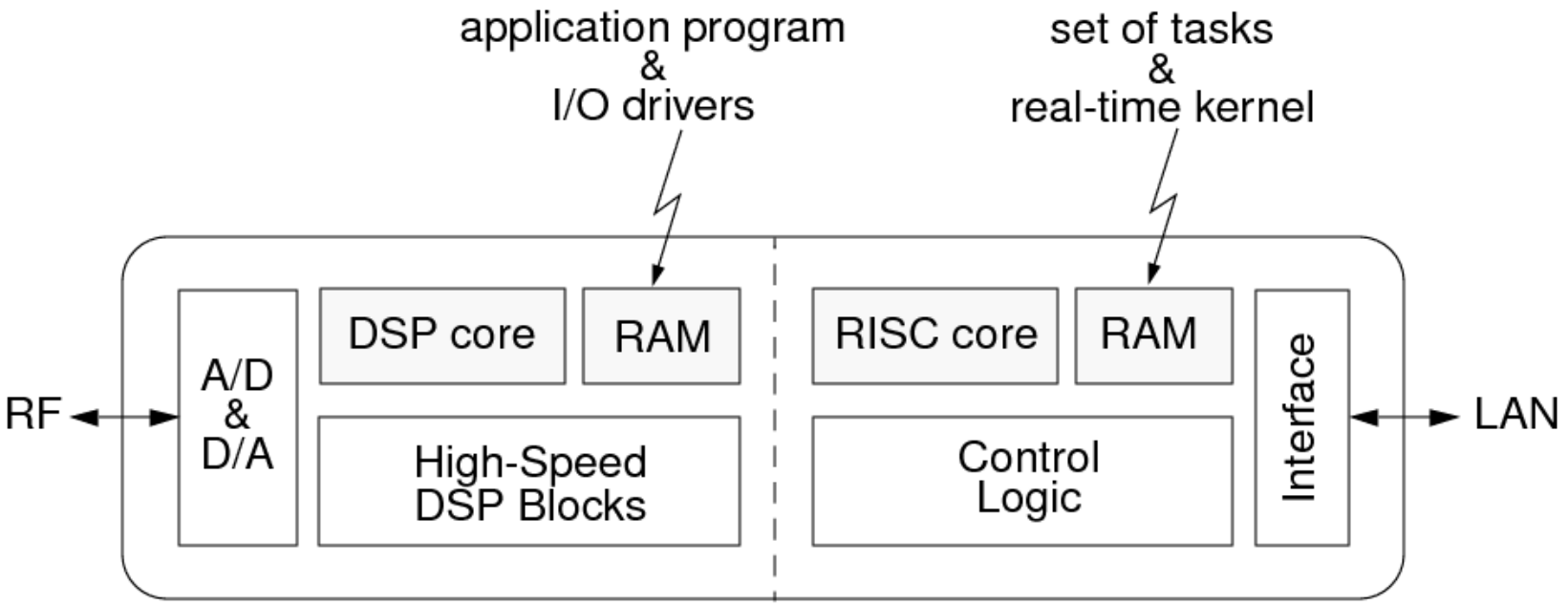
- A microporcessor and its memory subsystems
  - 8-bit or 64-bit RISC.
  - the memory system can be single or multi-levelled.
- A datapath with interfaces to the external system
  - the external interfaces can be bus drivers, Ethernet interfaces.
  - ADC or DAC.
- Blocks performing transformations on data received from the external system.
  - can be implemented as ASIC and/or DSP cores.
- Interface logic to peripherals

ARM Cortex M3



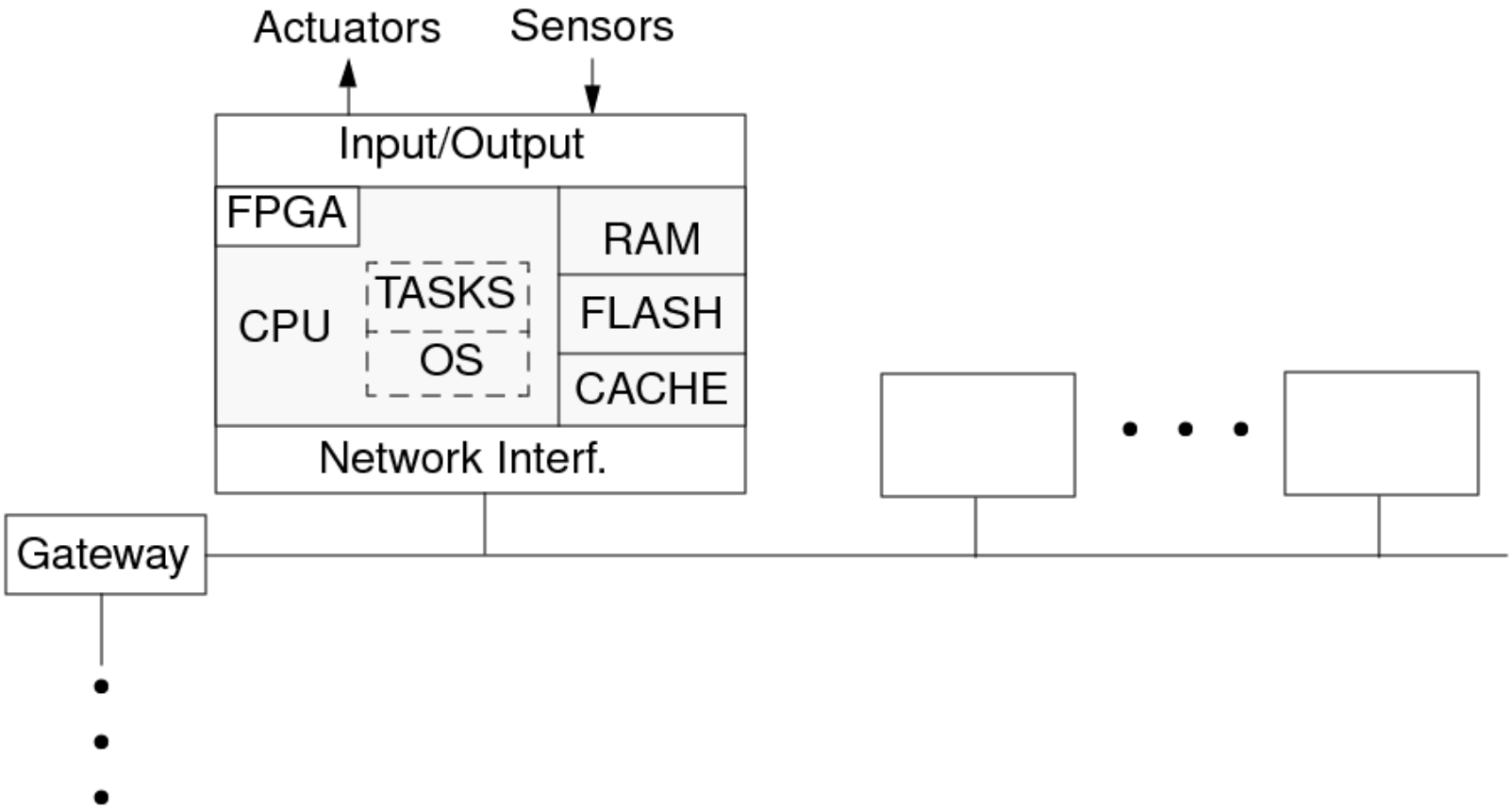
Typical SoC

- A typical structure for a wireless application:

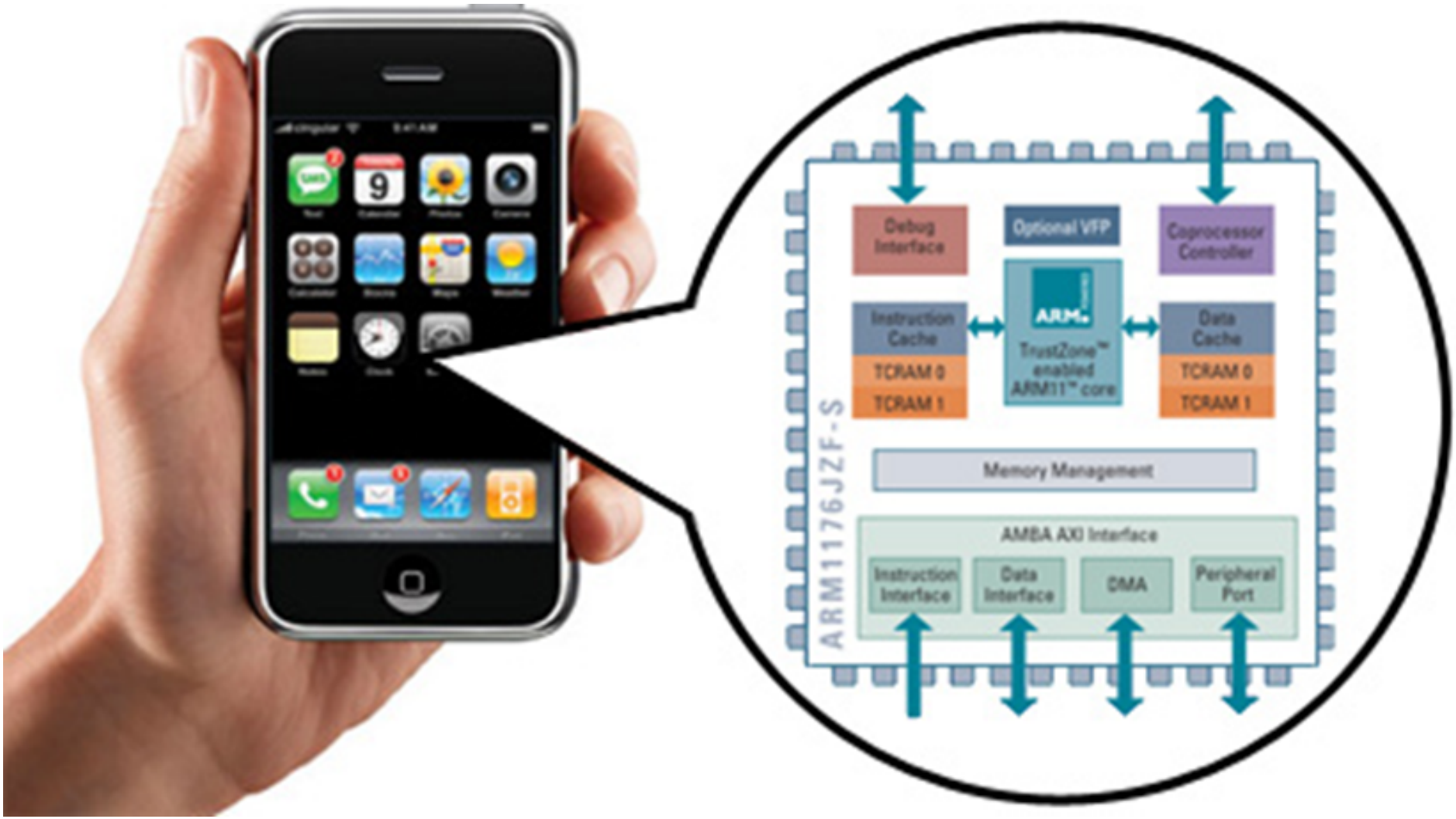


Typical SoC

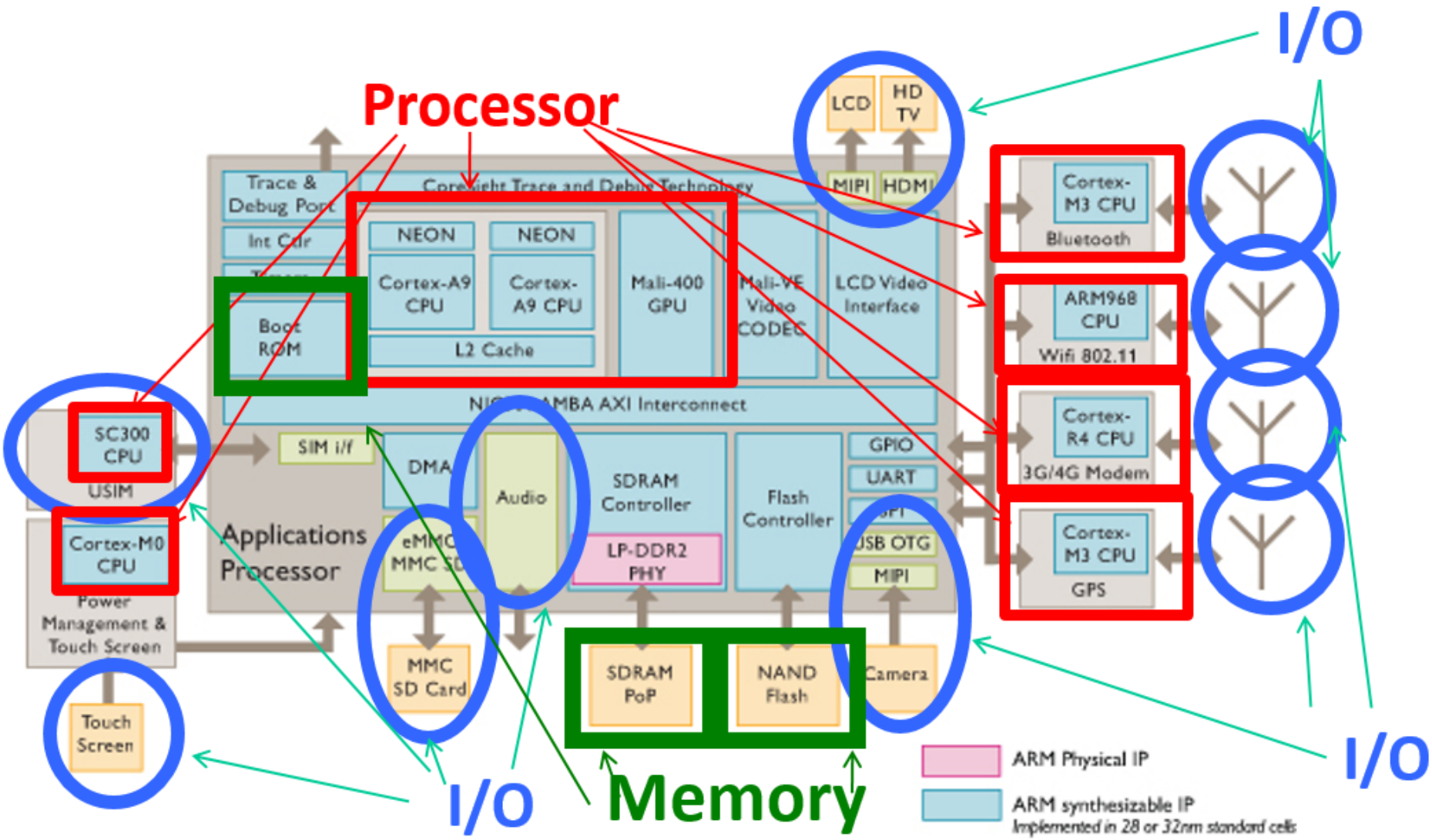
- A typical structure for an automotive application:



Typical SoC


Debug Interface
Optional VFP
Coprocessor Controller
Instruction Cache
TCRAM 0
TCRAM 1
ARM
TrustZone™ enabled ARM11™ core
Data Cache
TCRAM 0
TCRAM 1
Memory Management
AMBA AXI Interface
Instruction Interface
Data Interface
DMA
Peripheral Port
ARM1176JZF-S

Typical SoC


Processor
I/O
Memory
I/O
I/O
LCD
HD TV
MIPI
HDMI
Trace & Debug Port
CoreSight Trace and Debug Technology
Int Ctlr
NEON
NEON
Cortex-A9 CPU
Cortex-A9 CPU
L2 Cache
Mali-400 GPU
Mali-VE Video CODEC
LCD Video Interface
Boot ROM
NIC-301 AMBA AXI Interconnect
SC300 CPU
USIM
SIM i/f
DMA
Audio
SDRAM Controller
LP-DDR2 PHY
Flash Controller
GPIO
UART
USB OTG
MIPI
Cortex-M0 CPU
Power Management & Touch Screen
Applications Processor
eMMC MMC SD
MMC SD Card
SDRAM PoP
NAND Flash
Camera
Touch Screen
Cortex-M3 CPU
Bluetooth
ARM968 CPU
Wifi 802.11
Cortex-R4 CPU
3G/4G Modem
Cortex-M3 CPU
GPS
ARM Physical IP
ARM synthesizable IP
Implemented in 28 or 32nm standard cells

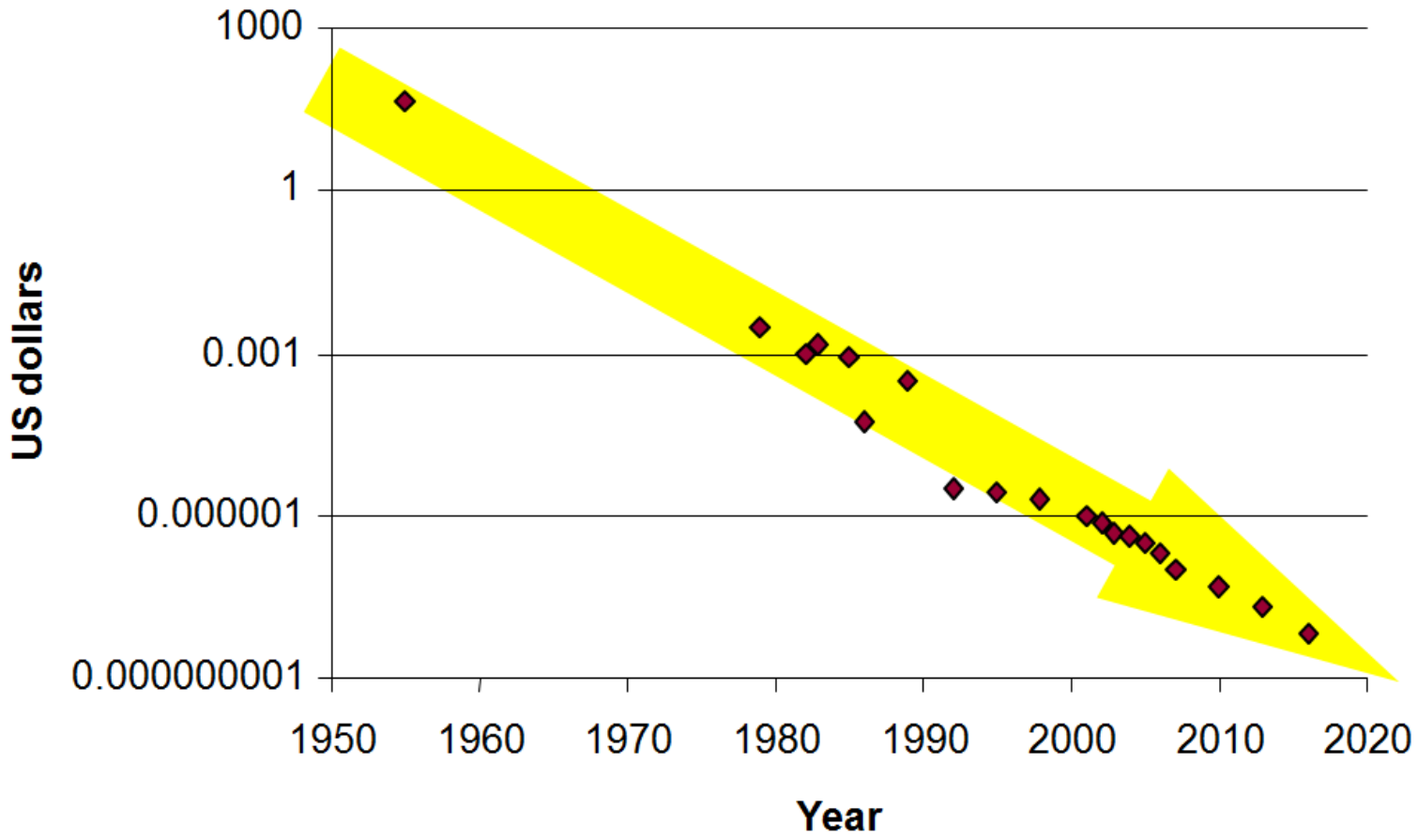
Section 2

WHY

Why do we need SoC

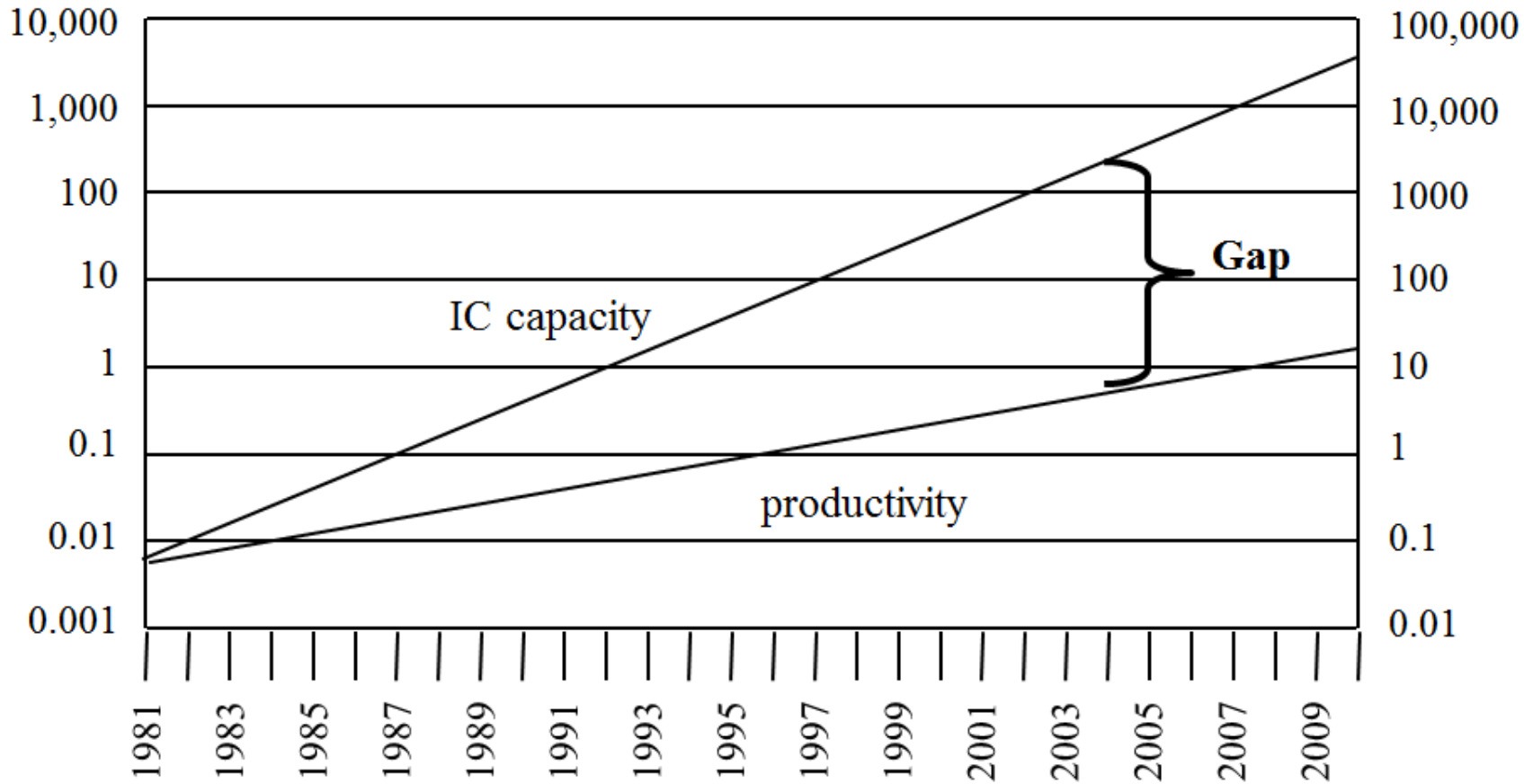
- Now SoC became technologically possible.
- VLSI technology advances:
  - today's chip can contains 100M transistors.
  - transistor gate lengths are now in term of nano meters.
  - approximately every 18 months the number of transistors on a chip doubles (Moore's law).
- The consequences:
  - components connected on a Printed Circuit Board can now be integrated onto single chip.
  - the development of System-On-Chip design .

Cost of a transistor



Productivity Gap

- There is an increasing productivity gap between IC hardware and the product produced by the skilled engineer.



Benefits

- Higher performance (fast data transfer compared to multi-chip design)
- Low power (multi-chip designs need additional drivers and bus interface for inter-chip and inter-board connection)
- Reduced size.
- Low cost.
- Increased reliability.
- Potential of innovative architectures.

Problems with SoC design

- Heterogeneous components
  - Analogue-digital
  - Processors - DSPs, ASIC, Programmable logic.
  - Different memory types (EEPROM, ROM, RAM, DRAM)
  - ...
- This creates problems with
  - Design: expertise in very difference design styles and areas
  - Technology: several components have to be incorporated onto a single system.

Section 3

HOW

Solution is Design Re-use

- Overcome complexity and verification issues by designing Intellectual Property (IP) to be re-usable.
- Done on such a scale that a new industry has been developed.
- Design activity is split into two groups:
  - IP Authors - producers
  - IP Integrators - consumers
- IP Authors produce fully verified IP libraries, thus making overall verification task more manageable.
- IP Integrators select, evaluate, integrate IP from multiple vendors - IP integrated onto Integration Platform designed with specific application in mind.

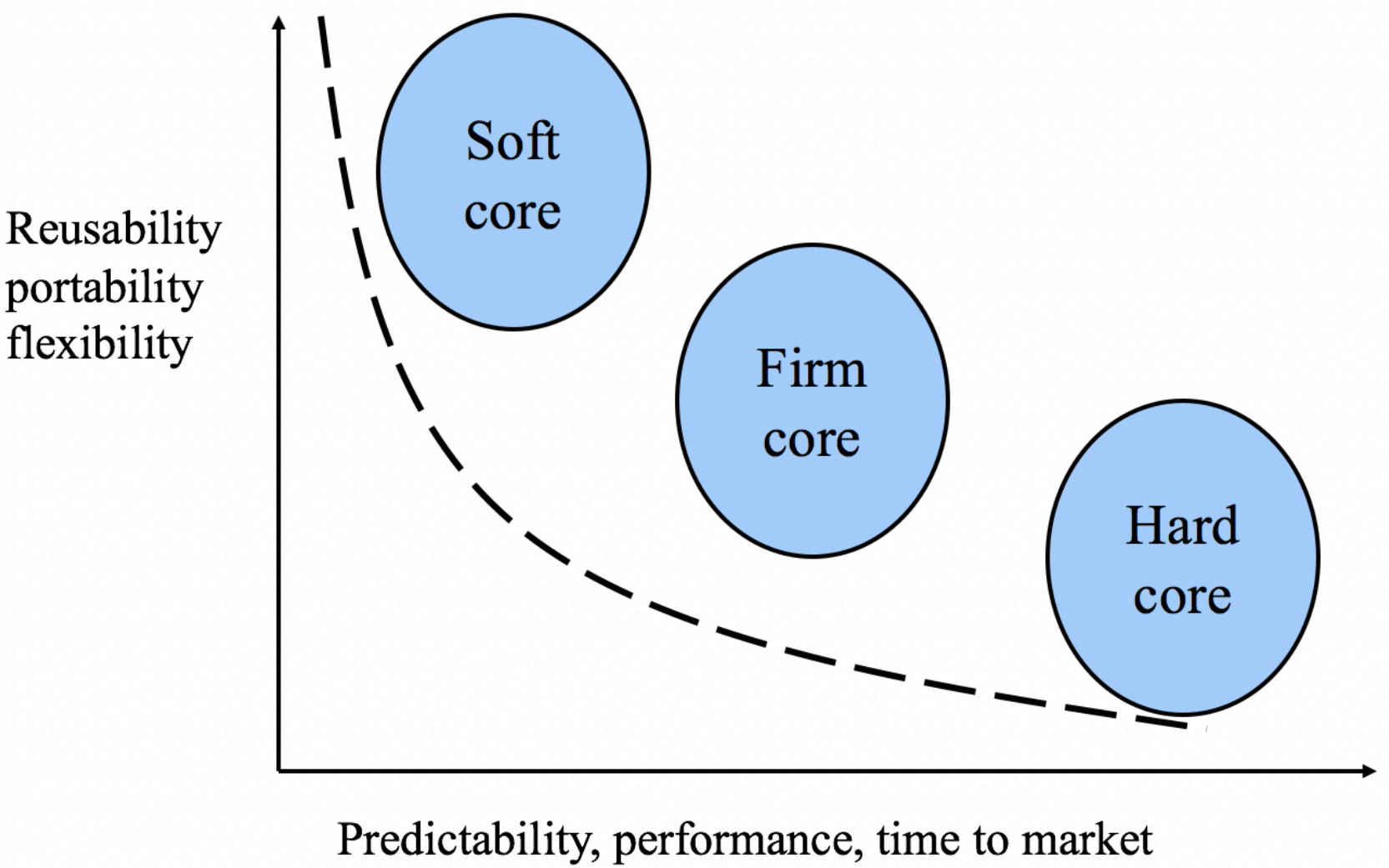
System on Chip cores

- Make ASIC designs more standardised by reusing segments of previously manufactured chips.
- These segments are known as “blocks”, “macros”, “cores” or “cells”.
- The blocks can either be developed in-house or licensed from an IP company.
- Cores are the basic building blocks.

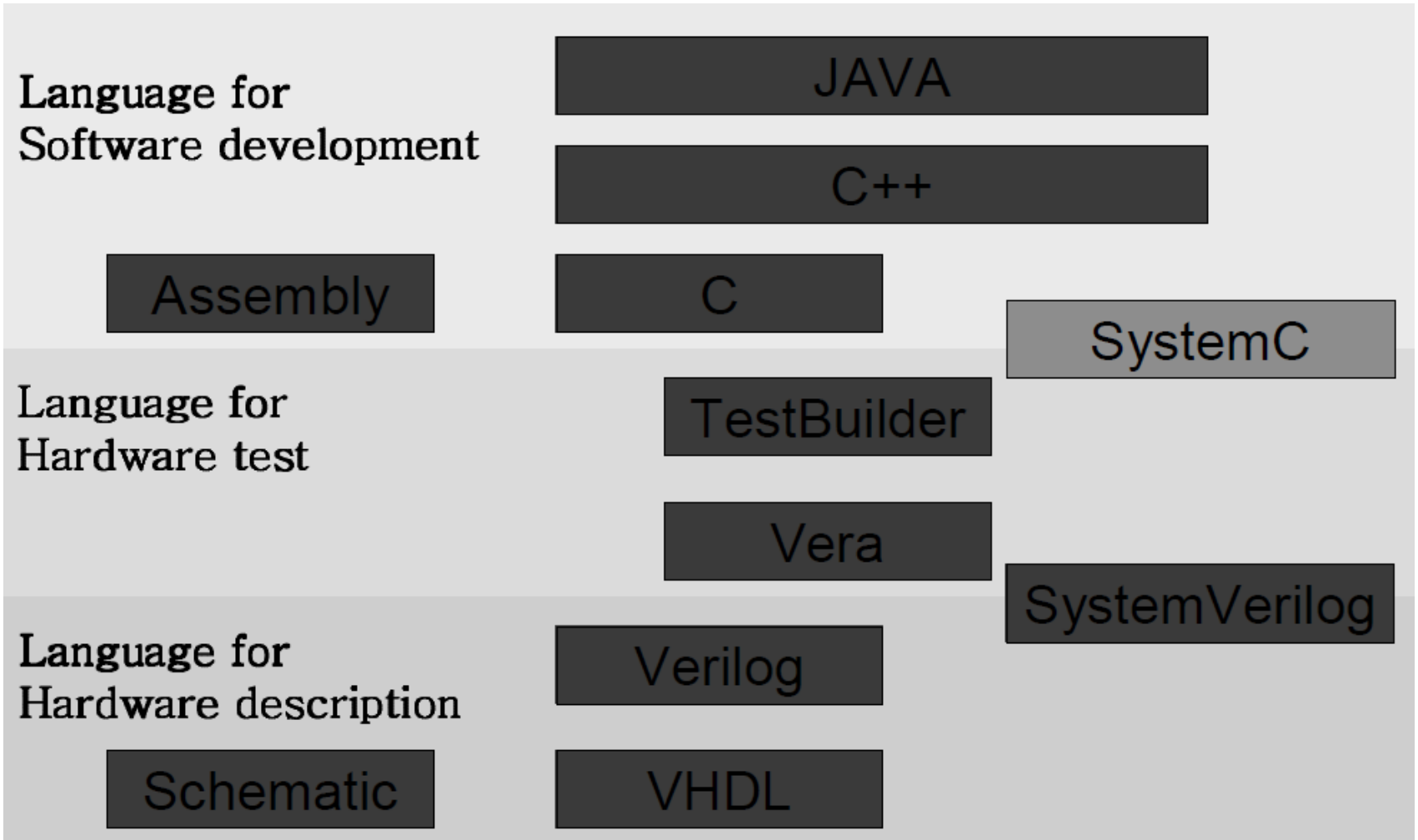
System on Chip cores

- Soft Macro
  - Reusable synthesizable RTL or netlist of generic library elements.
  - User of the core is responsible for the implementation and layout.
- Firm Macro
  - Structurally and topologically optimised for performance and area through floor planning and placement.
  - Exist as synthesized code or as a netlist of generic library elements.
- Hard Macro
  - Reusable blocks optimised for performance, power, size and mapped to a specific process technology.
  - Exist as fully placed and routed netlist and as a fixed layout.

System on Chip cores



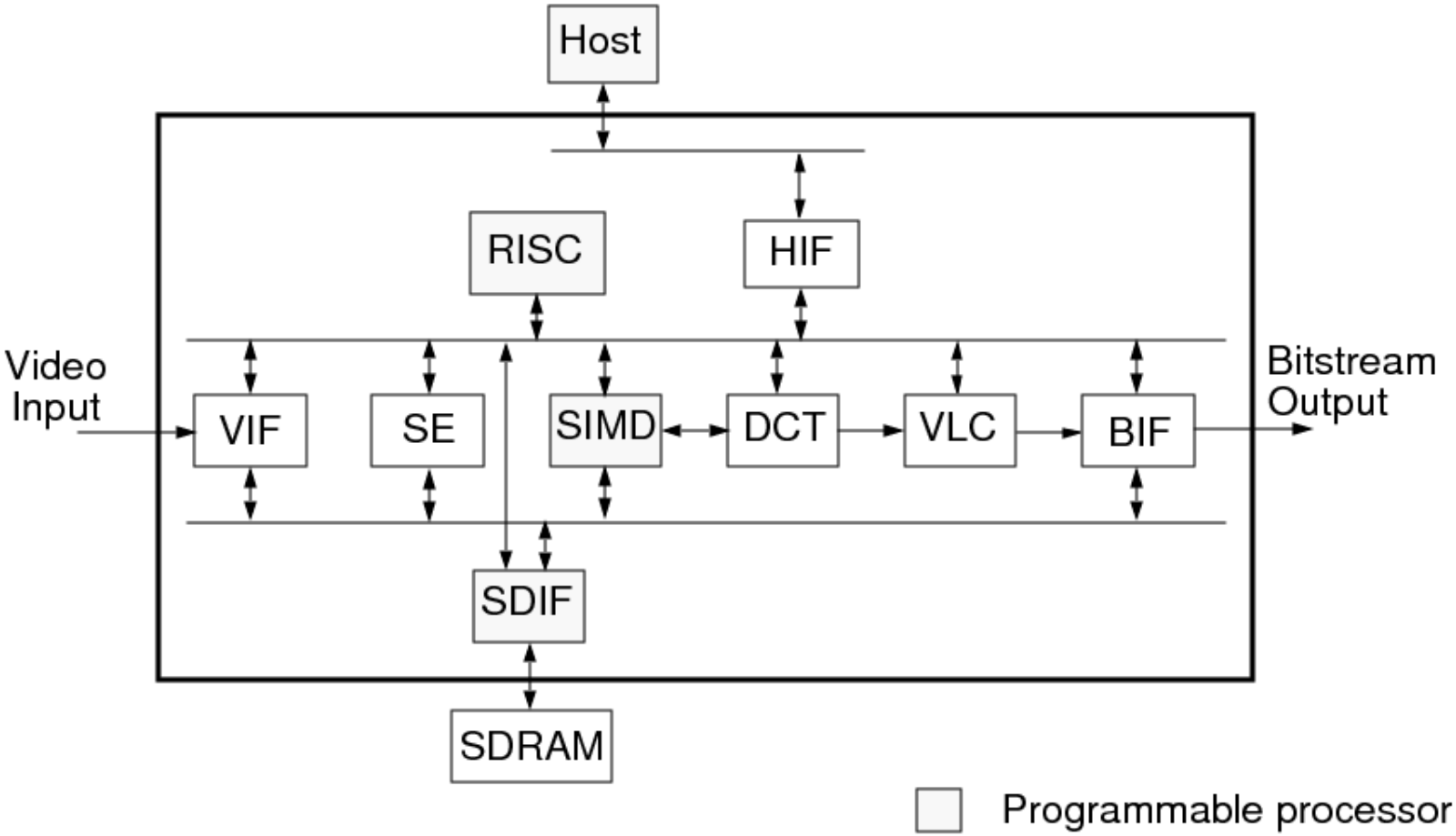
Language Evolution for SoC Design



SoC for Video Ecoding (MPEG-2 Encoding)

- The encoder receives video input and outputs a bitstream.
- ME (motion estimation) + MC(motion compensation): find out where a macroblock has moved to when comparing the current picture with a reference.
- DCT: discrete cosine transform
- VLC: variable length coding
- BIF: bitstream interface
- VIF: video interface
- HIF: host interface
- SDIF: SDRAM interface

SoC for Video Ecoding (MPEG-2 Encoding)



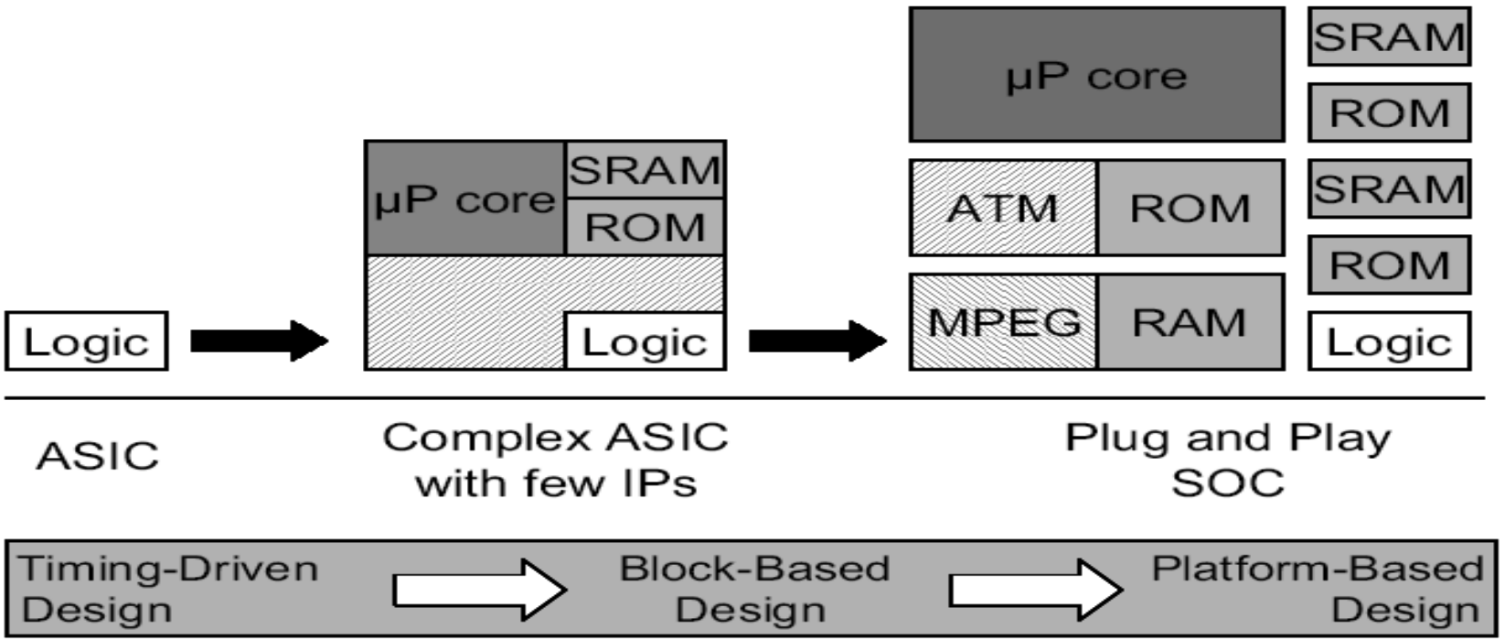
SoC for Video Ecoding (MPEG-2 Encoding)

- Functions which require flexibility and programmability have been mapped on programmable processors:
- A RISC processor controls the overall encoding process
- ME+MC are partitioned between a dedicated hardware (search engine SE) and a programmable SIMD (single instruction multiple data) machine.
- Pictures are stored on an external SDRAM; buffering and data flow to and from SDRAM are controlled by a programmable device.
- RISC, HIF, VIF, DCT, VLC are reused cores.
- Some features: 9.8 by 9.8 mm^2, 5M transistors, 81MHz clock frequency.

Section 4

Platform based Design

Platform based Design



- In TDD, re-use in ASIC design is of cell level libraries.
- In BBD, re-use in hierarchical design is of major IP blocks.
- In PBD, re-use is of collections of IP blocks organised into HW-SW architecture.

Hardware IP Integration

- Block-based design
  - No standard interfaces.
  - Not good for SoC design due to extra efforts to design proprietary interface logic as well as increased complexity to verify system integration.
- Platform-based design
  - Standard interface depending on platform bus system.
  - Plug-n-Play IP
  - Abundant silicon approved IP resources for popular platforms.

Platform based Design

- It's probably not a good idea to develop an all-purpose platform for IP integration, since different applications requires different CPU/DSP powers, memories, IOs, etc.
- Props:
  - Shorten time to market.
  - Easy to locate problems in a complicated system where major components are already pre-verified.
  - Easy to extend functions from a basic design.
  - Some popular platforms are already there for use.
- Cons:
  - Predefined platform restricts design flexibility.
  - Hardware design converges. Main differentiation is provided in software.

Buses for SoC Platforms

- Embedded processor cores need to communicate with memory and I/O devices.
- Want to be able to add various IP blocks seamlessly.
- Communication is typically done using **on-chip buses** - shared communication link between IP blocks
- Two main types of buses:
  - Relatively short CPU-memory bus for high-speed access.
  - Relatively long I/O bus for various I/O devices in the system operating at different speeds, typically slower than CPU memory operations.
  - Could combine buses to reduce overhead, but speed differences usually require separate buses with a connecting bridge.

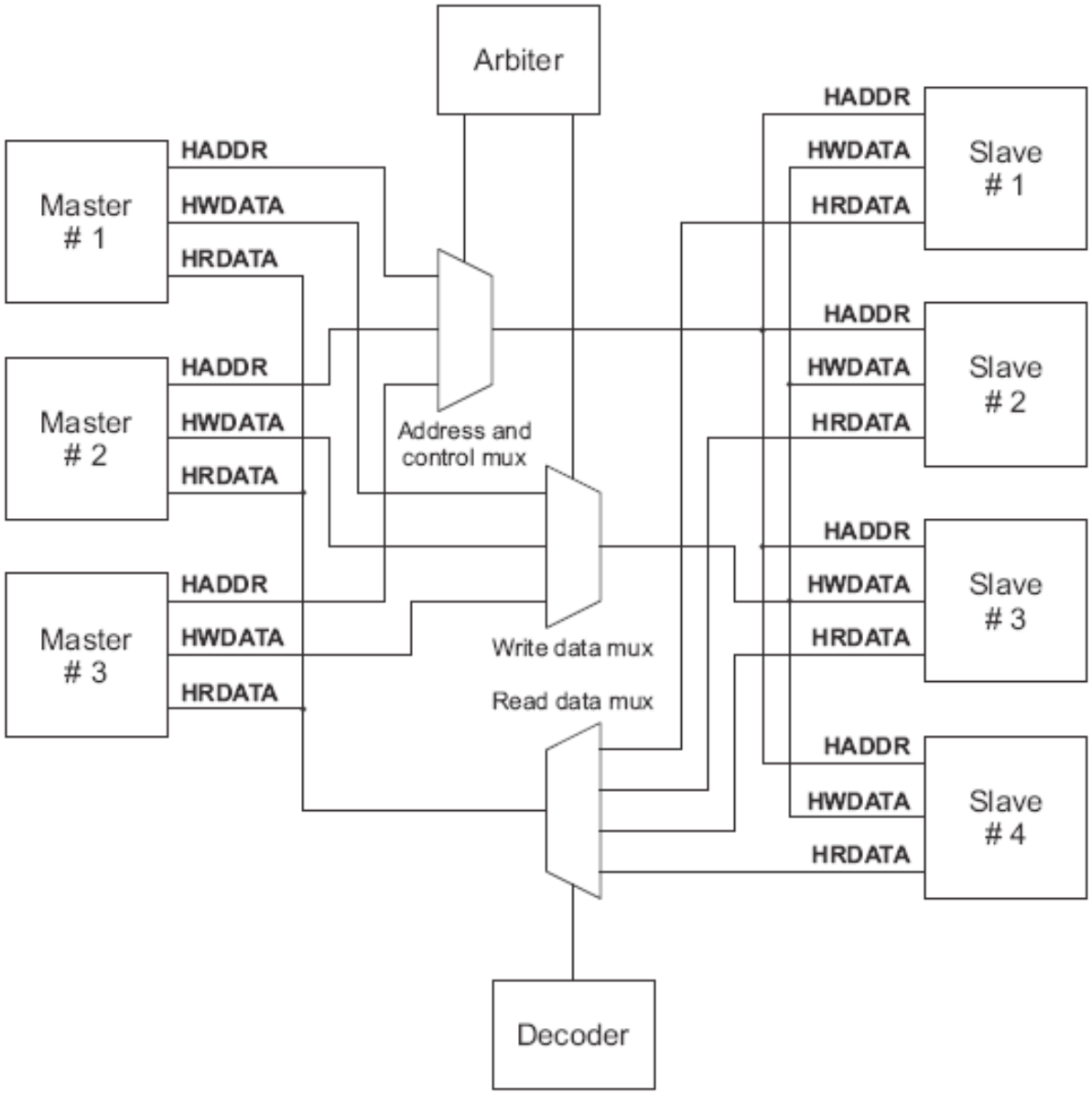
Buses for SoC Platforms

- Have to standardise buses to allow relatively quick connections of IP blocks in an SoC
- Some common buses are:
  - AMBA(ARM)
  - CoreConnect (IBM)
  - OCP-IP (VSIA Standard)
  - SoC-it (MIPS)

AMBA

- AMBA: Advanced Microcontroller Bus Architecture
- Created by ARM to enable standardised interfaces to their embedded processors
- Actually three standards: APB, AHB, and AXI
- Very commonly used for commercial IP cores

AMBA AHB


Arbiter
Master # 1
Master # 2
Master # 3
HADDR
HWDATA
HRDATA
Address and control mux
Write data mux
Read data mux
Slave # 1
Slave # 2
Slave # 3
Slave # 4
Decoder

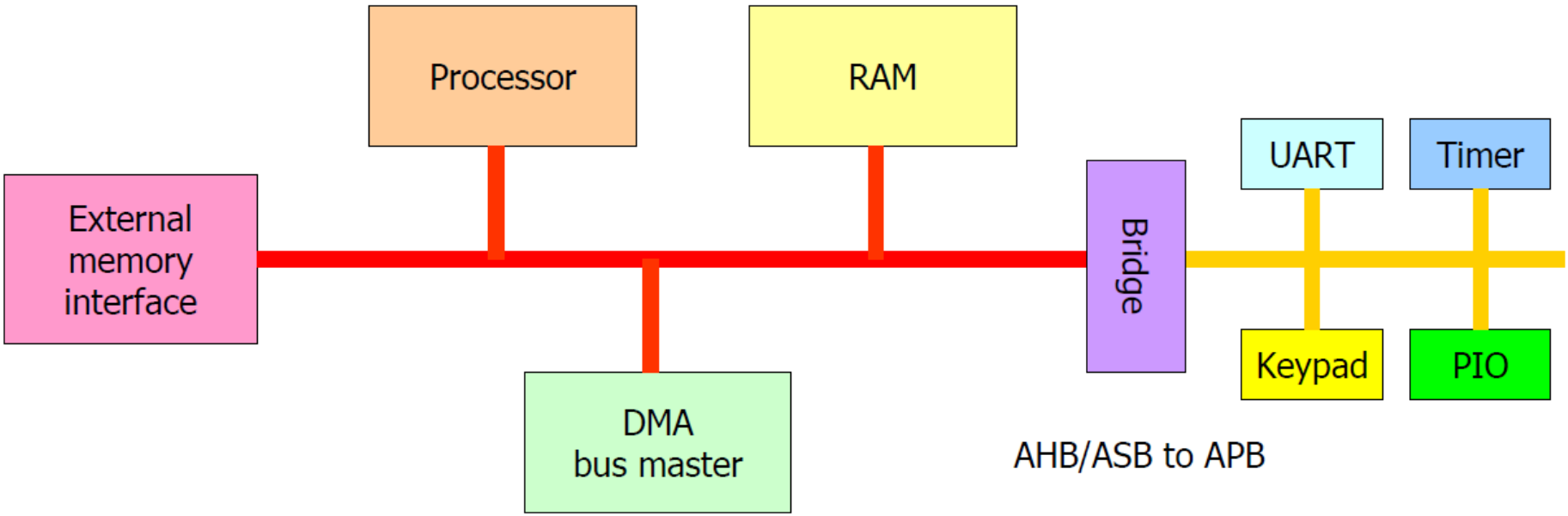
AMBA AHB

- AHB master is able to initiate read and write operations.
- AHB slave responds to a read or write operation within a given address range.
- AHB arbiter ensures that only one bus master at a time is allowed to initiate data transfers.
- AHB decoder is used to decode the address of each transfer and provide a select signal for the involved slave.

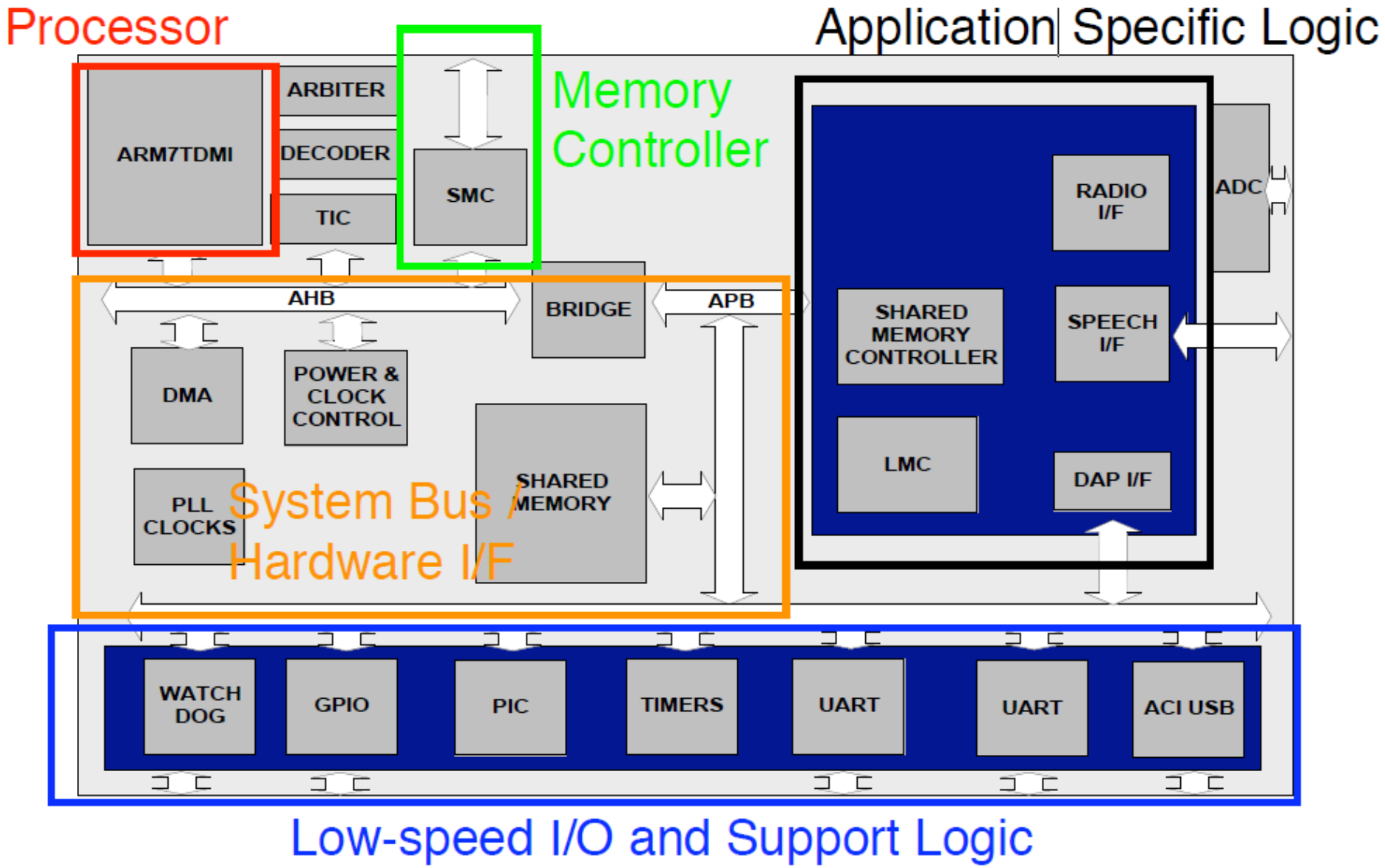
AMBA APB

- The APB bus is a low-cost bus used to connect low-bandwidth peripherals of an SoC.
- The APB is connected to a system-bus (such as AHB and AXI) using a bridge.
- The bridge allows masters on the AHB to address the APB system as a single slave.
- The bridge is the only Bus Master on the APB.
- The bridge decodes the address and generates the peripheral select.
- The bridge drives the data from/to the APB.

AMBA APB


Processor
RAM
External memory interface
Bridge
UART
Timer
Keypad
PIO
DMA bus master
AHB/ASB to APB

Bluetooth SoC



Networks-on-Chip *NoC*

- Number embedded processors in SoC rising:

ST: recordable DVD	5
Hughes: set-top box	7
Agere: Wireless base station	8
ST: HDTV platform	8
Latest mobile handsets	10
NEC: Image processor	128
In-house NPU (network processing unit)	150

Networks-on-Chip *NoC*

- The SoCs of the future will:
  - have 100s of hardware blocks,
  - have billions of transistors,
  - have multiple processors,
  - have large wire-to-gate delay ratios,
  - handle large amounts of high-speed data,
  - need to support plug-and-play IP blocks
- We need NoCs for these SoCs.
- VLSI Implementation.

AMBA AXI

- We want our NoCs to be "plug-and-play" so industry standardisation is key.
- The standard should be universal enough to address many different needs.
- AMBA AXI is an example of an attempt at this.
  - **It defines the interface and leave the interconnect up to the designers**.
  - Specific bus implementation is no longer required.

AMBA AXI

- Interface divided into 5 channels.
  - Write Address
  - Write Data
  - Write Response
  - Read Address
  - Read Data/Response
- Each channel is independent and use two way flow control.

Networks-on-Chip

- Network Protocol
  - circuit-switched : plan a connection before communication starts
  - packet-switched : issues packets which compete for network resources.
- Network Topology
  - Fully connected
  - Mesh
  - Tree
  - ...
- There is lots of research here