SMILIES

reSilient coMputer archItectures and LIfE Sciences

Politecnico di Torino

Department of Control and Computer Engineering

DAUIN

INTRODUCTION TO COMPUTER ORGANIZATION

STEFANO DI CARLO

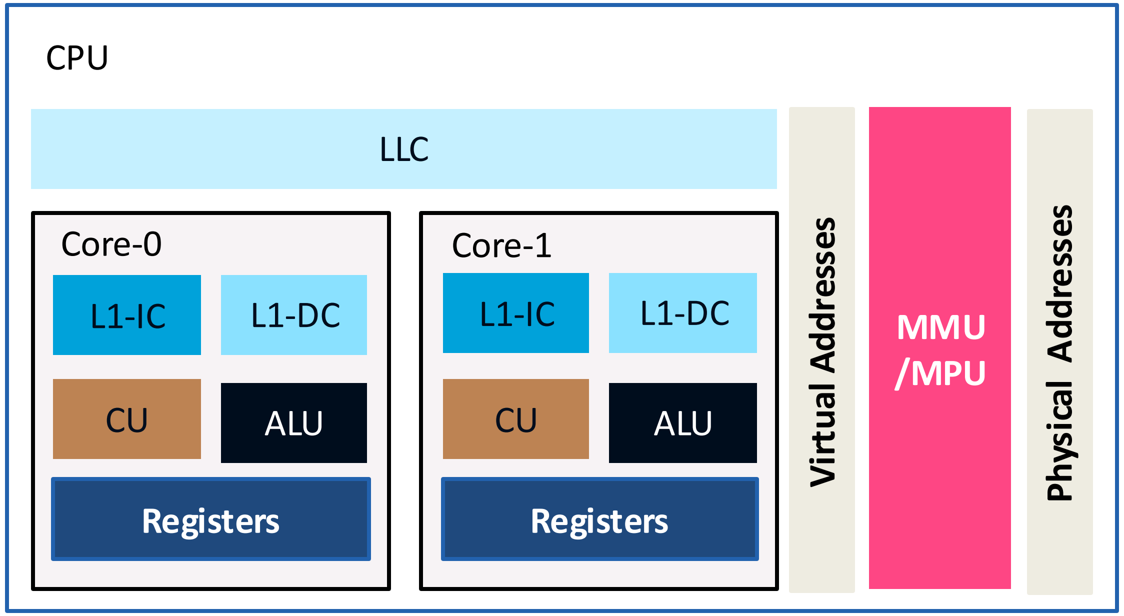
INTRODUCTION

- Just as buildings, each computer has a visible structure, referred to as its **architecture**.
- In computer science and engineering **computer architecture** is the practical art of **selecting and interconnecting hardware components** to create computers that meet functional, performance and cost goals.
- The functional blocks in a computer are of four types:

    1. Central Processing Unit
    2. Memory
    3. Input Unit
    4. Output Unit

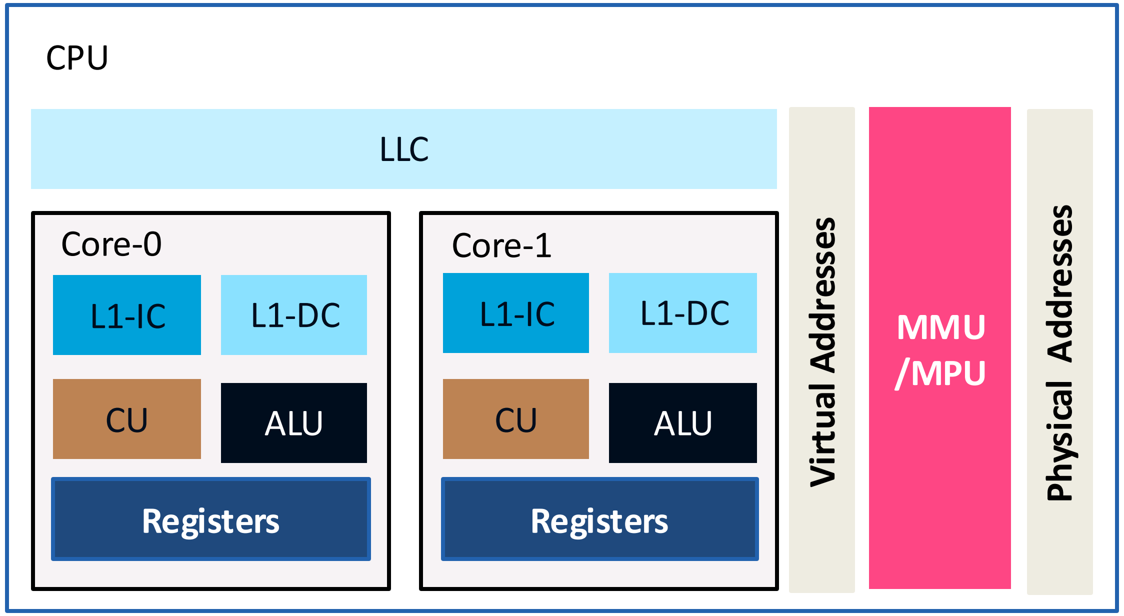
Central Processing Unit (CPU)

INSIDE THE CPU



- **Core —** A processing unit within a microprocessor capable of executing instructions independently.
- **Registers —** A small, fast storage location within the CPU that holds data or addresses used by the processor.
- **Control Unit (CU) —** The component of a CPU that manages the execution of instructions by directing data flow and operation sequencing.
- **Arithmetic Logic Unit (ALU) —** The CPU component that performs arithmetic and logical operations on data.
- **L1 Instruction Cache (L1-IC) —** A small, fast cache that stores the most frequently accessed instructions for quick retrieval.
- **L1 Data Cache (L1-DC) —** A small, fast cache that stores frequently accessed data close to the core for quick access.

INSIDE THE CPU



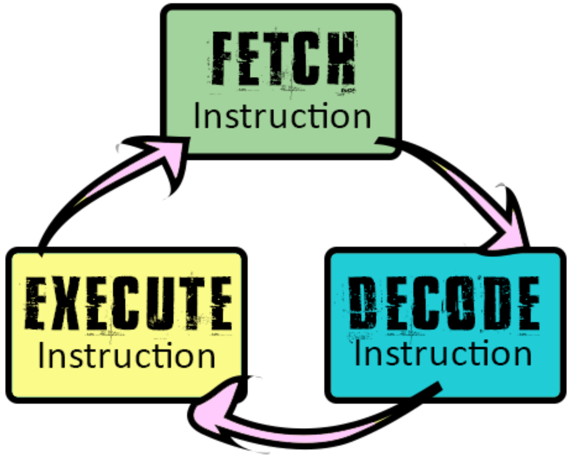
- **Last Level Cache (LLC) —** The largest and slowest cache level, typically shared by all cores, providing a buffer before accessing main memory.
- **Memory Management Unit (MMU) —** A hardware component that handles virtual-to-physical address translation and memory protection.
- **Memory Protection Unit (MPU) —** A hardware feature that enforces access control and protection on memory regions within the microprocessor.
- **Virtual/Physical Address —** An address generated by the CPU used by programs to access memory, translated by the MMU into a physical address.

INSTRUCTION CYCLE

mmm!!!!
Interesting!!!!
mmm!!!

How a
computer
works!!!!

- A computer program consists of both instructions and data
- The program is stored in the memory
- To execute the program, the instructions must be fetched from memory, decoded and then executed
- This process continues forever



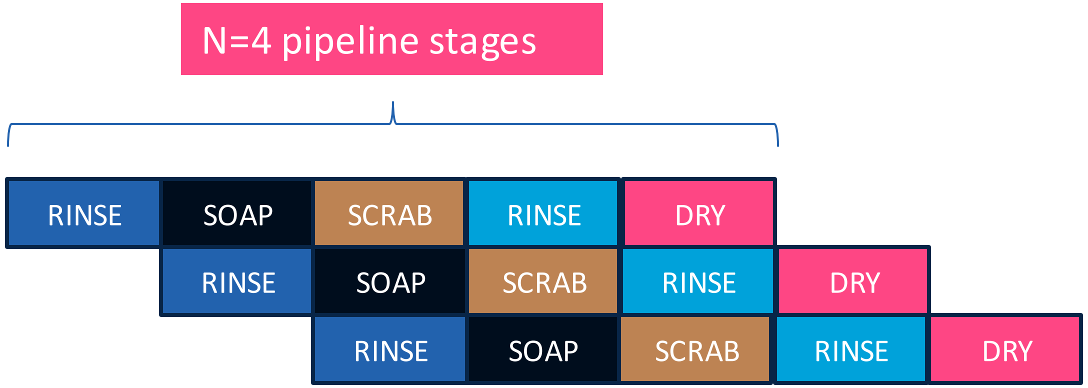
FETCH
Instruction

DECODE
Instruction

EXECUTE
Instruction

PIPELINING – OVERVIEW

- **Pipelining** is the mechanism used by modern processors to execute instructions:
    - It enables fetching an instruction, while other instructions are being decoded and executed **simultaneously.**
    - It allows the memory system and processor to work **continuously**..
- Pipelining plays an important role in increasing the efficiency of data processing.

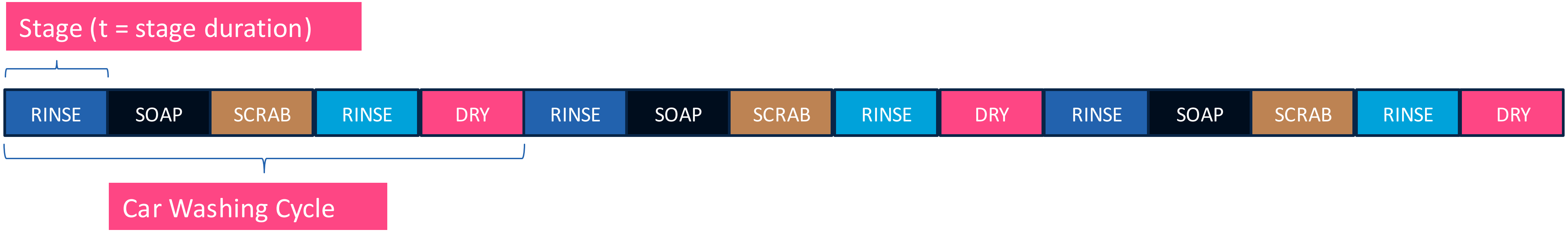


N=4 pipeline stages

RINSE SOAP SCRAB RINSE DRY

RINSE SOAP SCRAB RINSE DRY

RINSE SOAP SCRAB RINSE DRY



Stage (t = stage duration)

RINSE SOAP SCRAB RINSE DRY RINSE SOAP SCRAB RINSE DRY RINSE SOAP SCRAB RINSE DRY

Car Washing Cycle

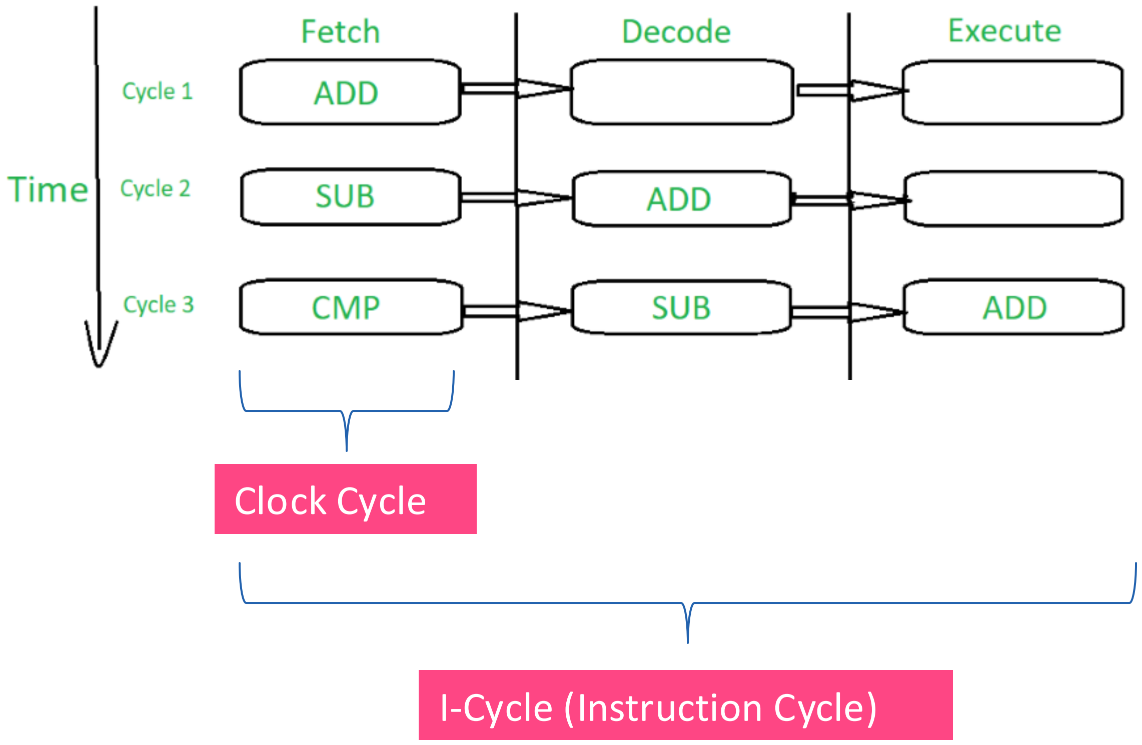
PIPELINING — OVERVIEW

- Latency (Execution Time for One Car):
  - $L = N \cdot t$ — non-pipelined
  - $L = N \cdot t$ — pipelined
- Throughput (Cars per Unit Time):
  - $T = \frac{1}{N \cdot t}$ — non-pipelined
  - When the pipeline is full: $T = \frac{1}{t}$ — pipelined
- Total Time for *m* Cars:
  - $T_{total} = m \cdot n \cdot t$ — non-pipelined.
  - Filling the pipeline takes $n \cdot t$ time.
  - After the pipeline is full, each subsequent element takes t time to process, so processing $m - 1$ additional elements takes $(m-1) \cdot t$
  - $T_{total} = n \cdot t + (m-1) \cdot t = (n + m - 1) \cdot t$

PIPELINING OF INSTRUCTIONS

An example of a generic 3 stages pipeline

[Taken from https://www.geeksforgeeks.org/pipelining-in-arm//]



	Fetch	Decode	Execute
Cycle 1	ADD		
Cycle 2	SUB	ADD	
Cycle 3	CMP	SUB	ADD

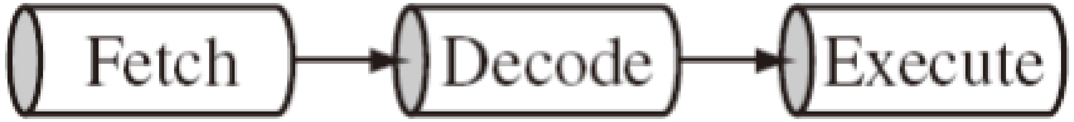
Clock Cycle

I-Cycle (Instruction Cycle)

- Pipeline stages:
  - **Fetch** loads an instruction from memory.
  - **Decode** identifies the instruction to be executed.
  - **Execute** processes the instruction and writes the result back (to registers or memory).
- By overlapping the three stages for different instructions, the speed of execution is increases.
- The pipelining allows the core to execute an instruction every cycle, which results in increased throughput.

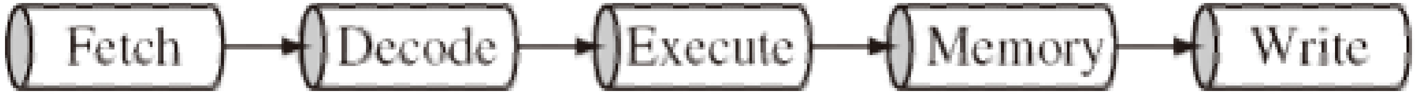
ARM-7 — 3 STAGE PIPELINE

- Pipeline stages:
  - **Fetch** — loads an instruction from memory.
  - **Decode** — decodes the instruction to be executed.
  - **Execute** — executes the instruction and writes the result back (to registers or memory).



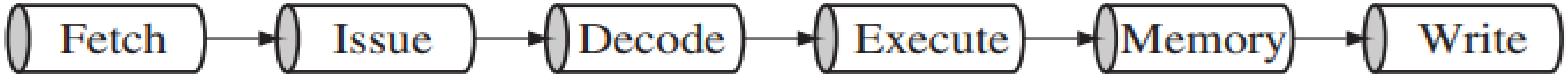
- Simplest pipeline in the ARM family
- Instruction-Cycle: 3 clock cycles.
- Advantages:
  - **Reduced Complexity:** the more is the number of stages, the more complex is the hardware required to handle the execution
  - **Lower Power** consumption: reduced complexity in general leads to reduced power consumption
  - **Reduced Latency:** latency is directly proportional to the number of stages of the pipeline ($L = N \cdot t$)

ARM-9 — 5 STAGE PIPELINE



- Pipelining in ARM 9 is similar to ARM 7 but with 5 stages.
- It takes 5 cycles to complete the process.
  - **Fetch** — loads an instruction from memory.
  - **Decode** — decodes the instruction to be executed..
  - **Execute (ALU)** — executes the instruction .
  - **Memory** — reads any registers needed by the instruction (Loads/Stores).
  - **Write** — writes back the result of the operation.
- Because of an increase in stages and efficiency, the throughput is 10%-13% higher than ARM-7.

ARM-10 – 6 STAGE PIPELINE

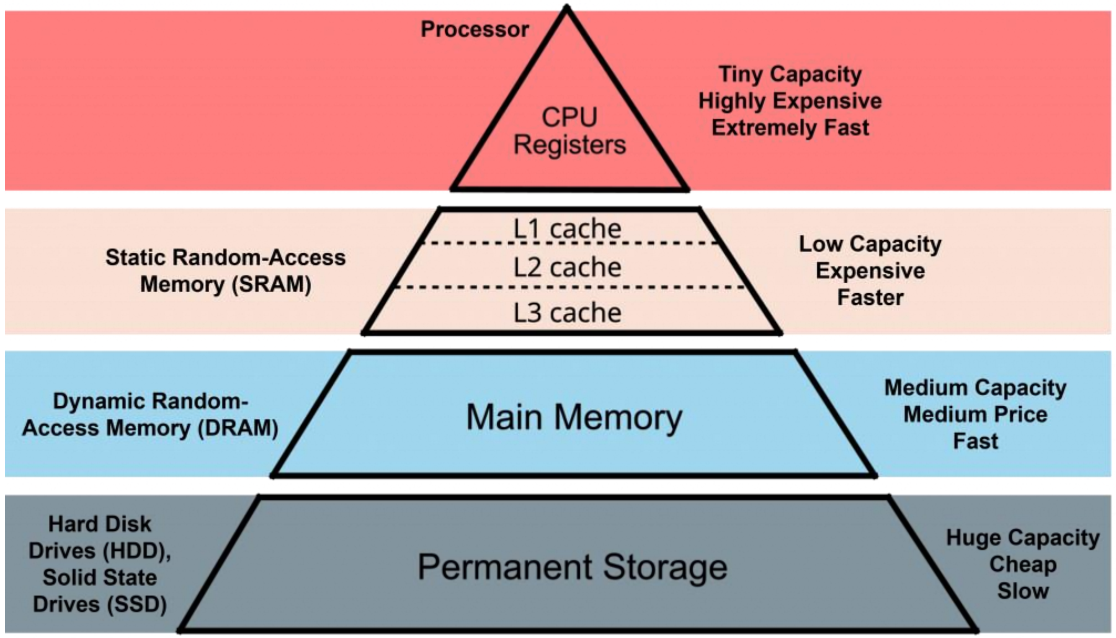


- Additional **ISSUE stage** added.
  - Issue stage checks if the instruction is ready to be decoded in the current stage or not.
  - If the instruction is not ready it allows **out-of-order execution** by allowing the next instruction in the pipeline to start processing in the available time gap.
- **Branch Prediction Mechanism** has been introduced to improve throughput.
- Reduces processor stalls by **resolving the Hazards**.
- Throughput is almost double of ARM 7 but latency is compromised (Trade-off)

Memory

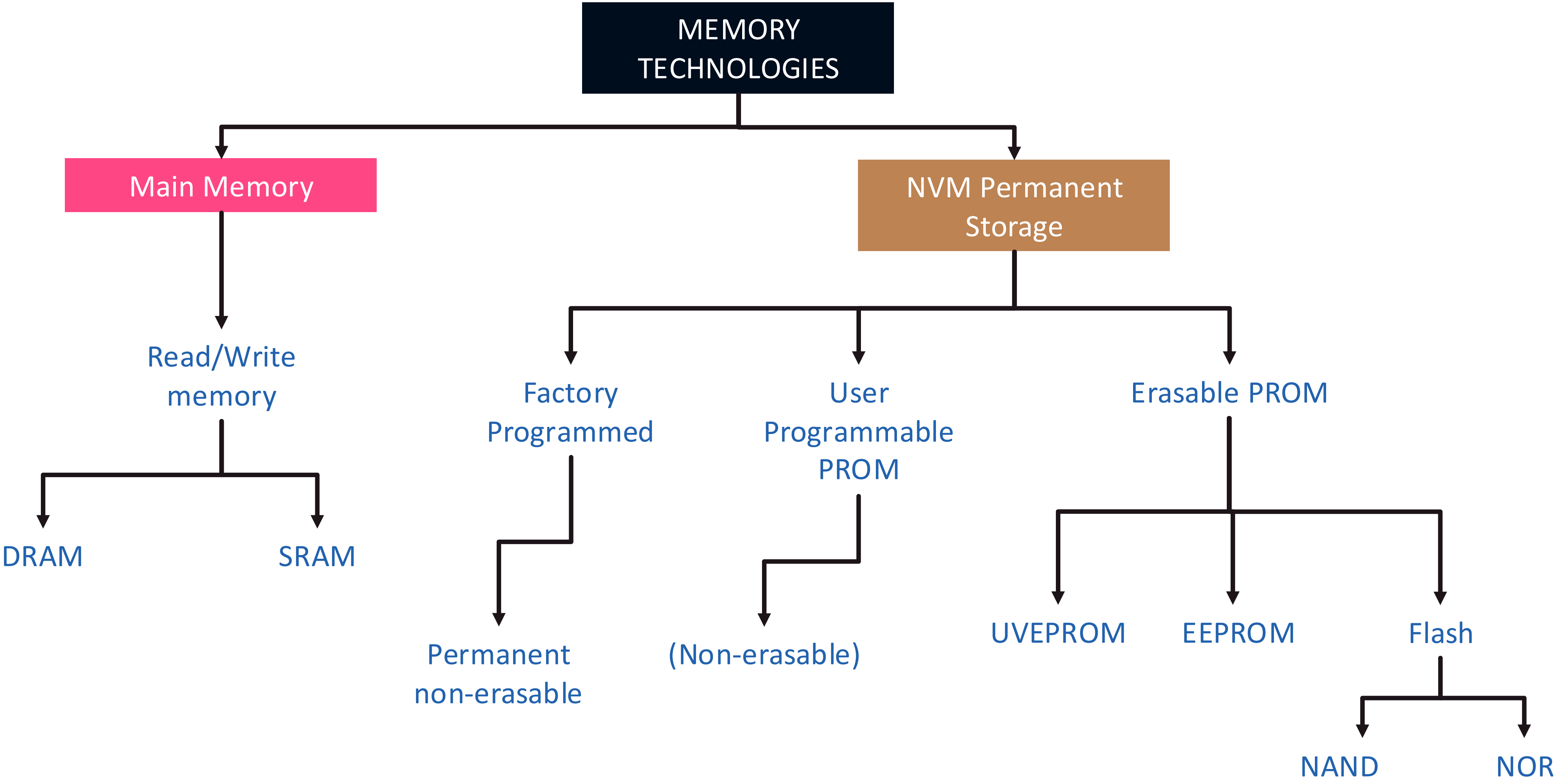
MEMORY

- Memory is the computer's electronic scratchpad or local store in computer terminology.
- Used for temporary storage of calculations, data, and other work in progress.
- Tree main types:
  - **CPU Memory:** volatile memory embedded in the CPU (registers and cache)
  - **Main Memory:** volatile memory usually referred to as Random Access Memory (RAM).
  - **Permanent Storage:** is a large Non-Volatile Memory (NVM) used to store programs and data.



[Taken from https://spear-itn.eu/memory-hierarchy-how-does-computer-memory-work/]

MEMORY SEMICONDUCTORS TECHNOLOGIES



RANDOM ACCESS MEMORY (RAM)

- The processor directly stores and retrieves information from it.
- Memory is organized into locations. Each memory location is identified by a unique address. The access time is same for all locations.
- It is volatile: when turned off, everything in RAM disappears.
- Two types:
  - **Dynamic Random Access Memory (DRAM)**: retains the content of any location only for a few milliseconds. Within that period, each location must be written again with the same content. This is known as refreshing.
  - **Static Random Access Memory (SRAM):** preserves the content of all the locations if the power supply is present. Typically used to implement cache memories.

NON—VOLATILE MEMORIES

- Data stored in NVM cannot be modified, or can be modified only slowly or with difficulty, so it is mainly used to distribute.
- The instructions in NVM are built into the electronic circuits of the chip which is called firmware.
- Random access in nature.

TYPES OF NVM

- Programmable read-only memory (**PROM**), or one-time programmable ROM can be written to or programmed via a special device called a PROM programmer.
- Erasable programmable read-only memory (**EPROM**) can be erased by exposure to strong ultraviolet light then rewritten with a process that again needs higher than usual voltage applied.
- Electrically erasable programmable read-only memory (**EEPROM**) is based on a similar semiconductor structure to EPROM, but allows its entire contents (or selected banks) to be electrically erased, then rewritten electrically, so that they need not be removed from the computer

FLASH MEMORY

- Modern type of EEPROM invented in 1984.
- Random access memories and are non-volatile.
- Use one transistor per memory cell and come in capacities ranging from 1 MB to 32 GB by the year 2007.
- The read time is much smaller (tens of nanoseconds) compared write time (tens of microseconds)
- Two technologies:
  - **NAND Flash** uses NAND logic gates to store data, allowing for high-density memory cells arranged in a grid, making it suitable for devices like USB drives and SSDs.
  - **NOR Flash** utilizes NOR logic gates, providing faster read speeds and random-access capabilities, commonly used in applications requiring quick data retrieval, such as firmware storage in embedded systems.

CACHE MEMORY

- High speed memory kept in between processor and RAM to increase the data execution speed.
- Kept near to the processor.
- Major reason for incorporating cache in the system is that the CPU is much faster than the DRAM and needs a place to store information that can be accessed quickly.
- Cache fetches the frequently used data from the DRAM and buffers (stores) it for further processor usage.

DIFFERENT LEVELS OF CACHE

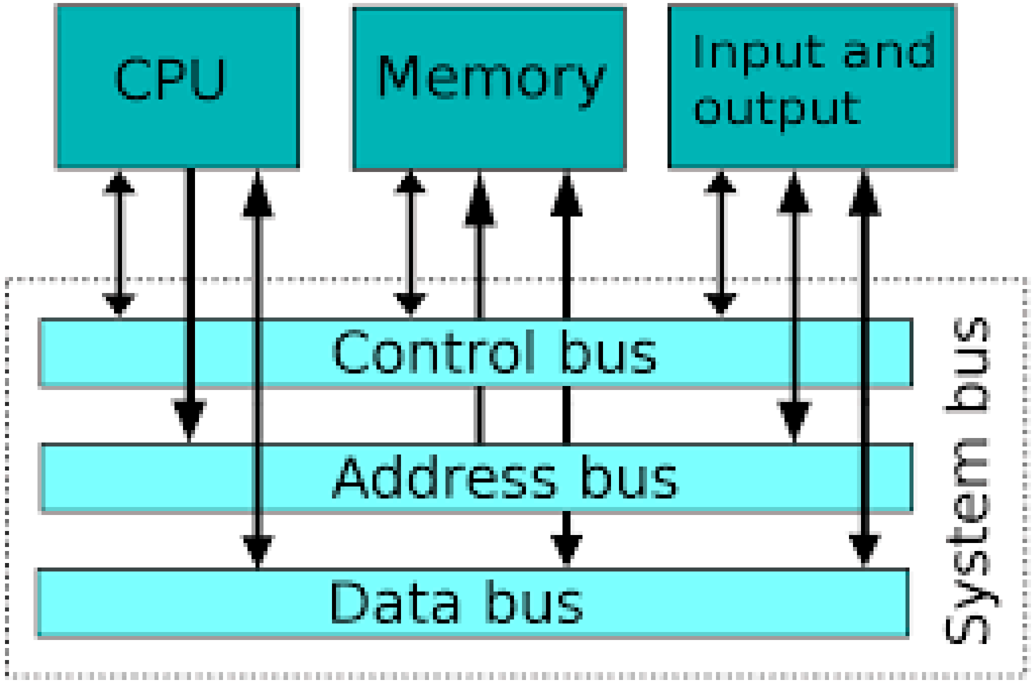
- L1-cache is the fastest cache, and it usually comes within the processor chip itself. L1 cache typically ranges in size from 8KB to 64KB and uses the high-speed SRAM instead of the slower and cheaper DRAM used for main memory.
- L2 cache comes between L1 and RAM and is bigger than the primary cache.
- L3 cache is not found nowadays as its function is replaced by L2 cache. L3 caches are found on the motherboard rather than the processor. It is kept between RAM and L2 cache.

PROCESSOR SPEED

- Speed of a computer system is determined by several factors, clock speed of the processor and the speed and size of the data bus.
- Clock speed is the rate at which the processor processes information and this is measured in millions of cycles per second (Megahertz)
- The more the number of hertz, the faster is the processing speed
- The larger the bus width and the faster the bus speed, the greater the amount of data can travel on it in each amount of time.

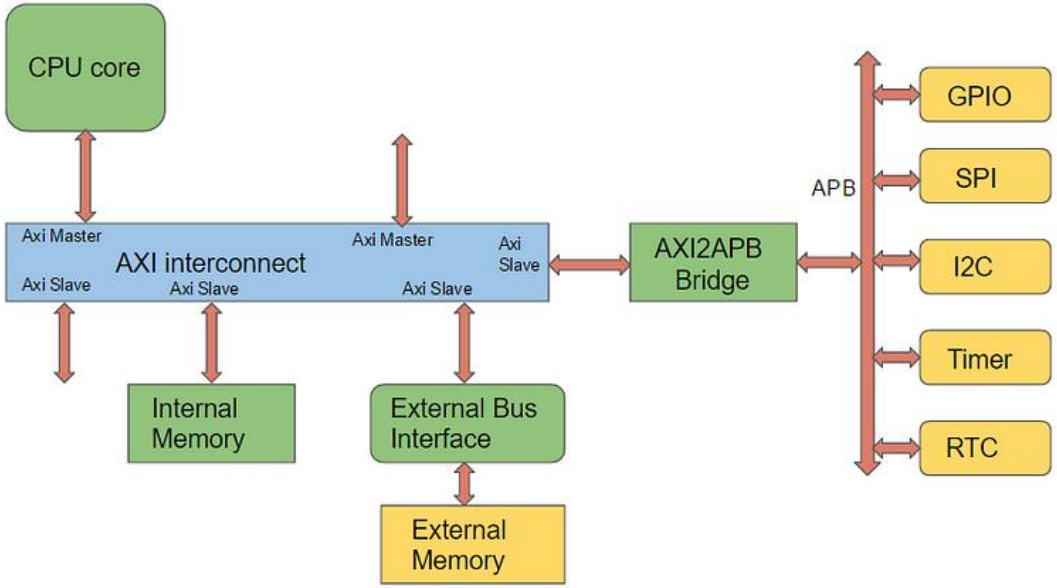
Input/Output

COMMUNICATION INSIDE A COMPUTER



[Taken from https://en.wikipedia.org/wiki/System_bus/]

THE EMBEDDED SYSTEMS BUS IN REAL



[Taken from https://anysilicon.com/understanding-amba-bus-architecture-protocols/]

- The **Advanced Micro controller Bus Architecture (AMBA)** bus protocols is a set of interconnect specifications from ARM that standardizes on chip communication mechanisms between various functional blocks (or IP) for building high performance SOC designs.
- With increasing number of functional blocks (IP) integrating into SOC designs, the shared bus protocols (AHB/ASB) started hitting limitations sooner and in 2003 , the new revision of AMBA 3 introduced a **point-to-point connectivity protocol — AXI (Advanced Extensible Interface)**.

SMILIES

reSilient coMputer archItectures and LIfE Sciences

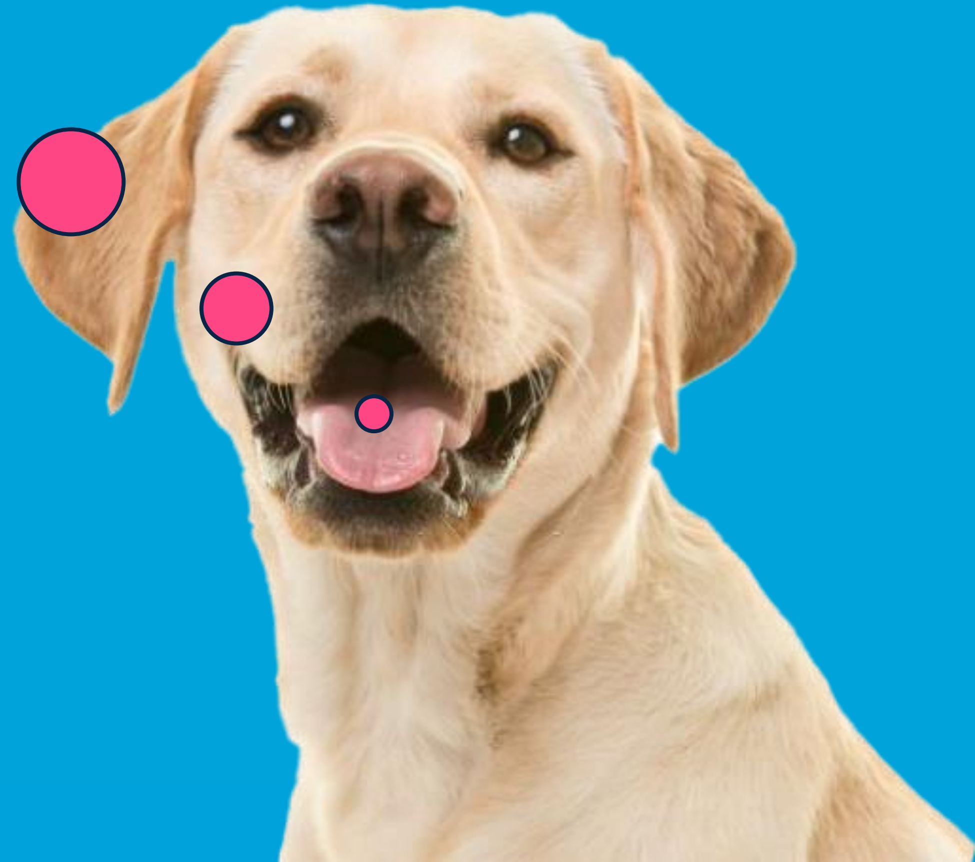
Politecnico di Torino

1859

Department of Control and Computer Engineering

DAUIN

QUESTIONS?



THANK YOU!