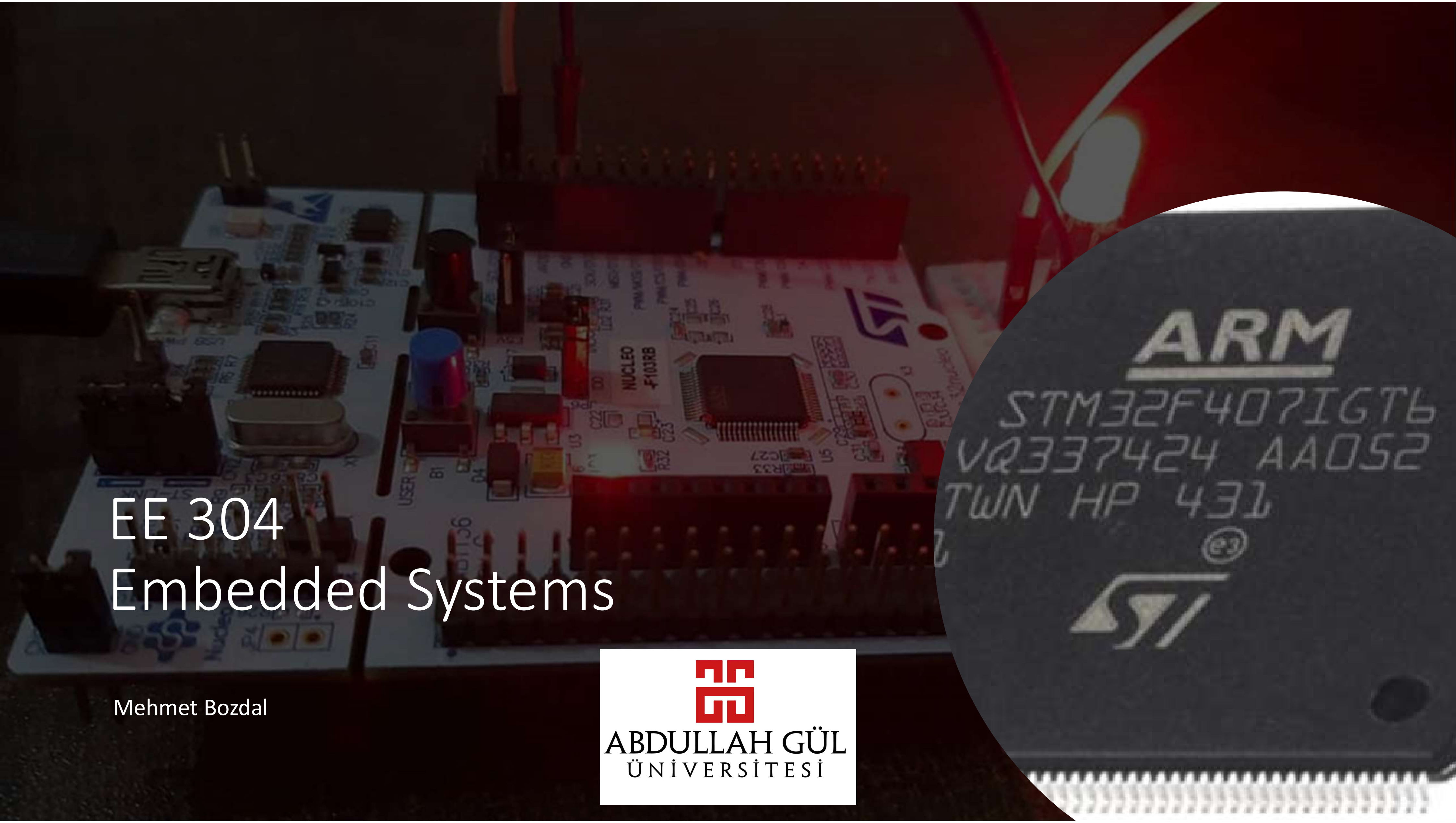


EE 304
Embedded Systems

Mehmet Bozdal

ABDULLAH GÜL
ÜNİVERSİTESİ

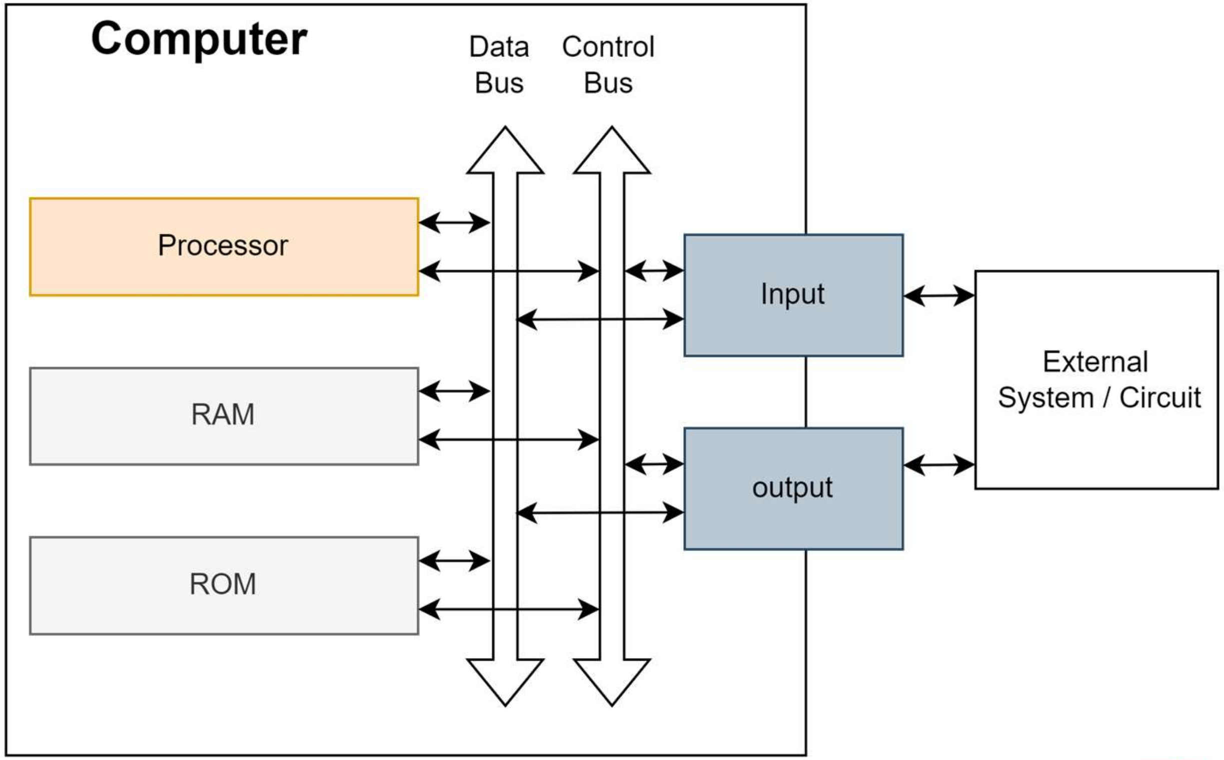
Agenda: Review of Computer Architecture

- Central Processing Unit (CPU)
  - Arithmetic Logic Unit (ALU)
  - Control Unit
  - Registers
- Memory
  - Role and Types
- Instruction Set Architecture (ISA)
  - Defining the Interface

Additional Resource: [1] Arm Education: Introduction to Microprocessors (edx)

The Basic Components of a Computer

- Central Processing Unit (CPU)
- Memory: Cash, Random Access Memory(RAM), storage devices
- Input - Output Devices



What is CPU?

Executes instructions stored in memory, controlling the overall operation of the computer.

1.Arithmetic Logic Unit (ALU)

- Performs arithmetic operations (addition, subtraction, etc.) and logical operations (AND, OR, NOT, etc.).

2. Control Unit

- Manages the flow of data and instructions within the CPU and to other parts of the computer.

3. Registers

- Small, high-speed storage locations used for temporary data storage during processing.

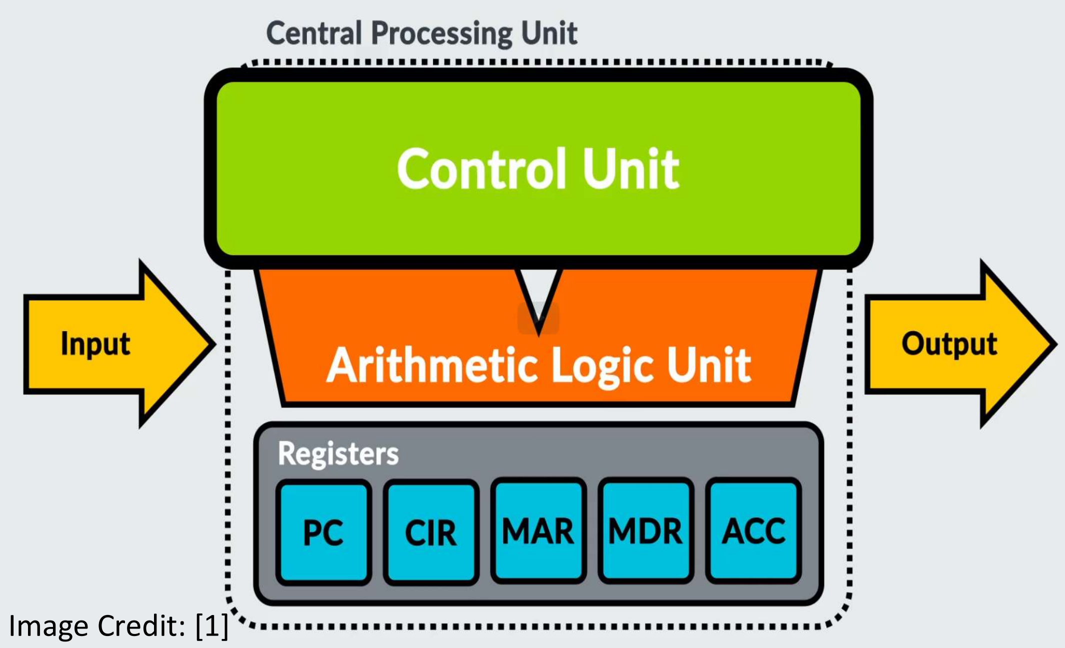


Image Credit: [1]

Memory and Buses

1. RAM (Random Access Memory)

- RAM is volatile memory that stores data and instructions for the CPU.
- It provides high-speed access for temporary data storage.
- Data is lost when the computer is powered off.

2. ROM (Read-Only Memory)

- ROM is non-volatile memory that stores firmware and permanent instructions.
- It retains data even when the computer is powered off.
- Typically contains the computer's startup instructions.

3. Buses

- Buses are data highways that connect different computer components.
- Address Bus: Carries memory addresses for data retrieval.
- Data Bus: Transfers data between the CPU and memory.
- Control Bus: Manages control signals for data flow and operations.

The System

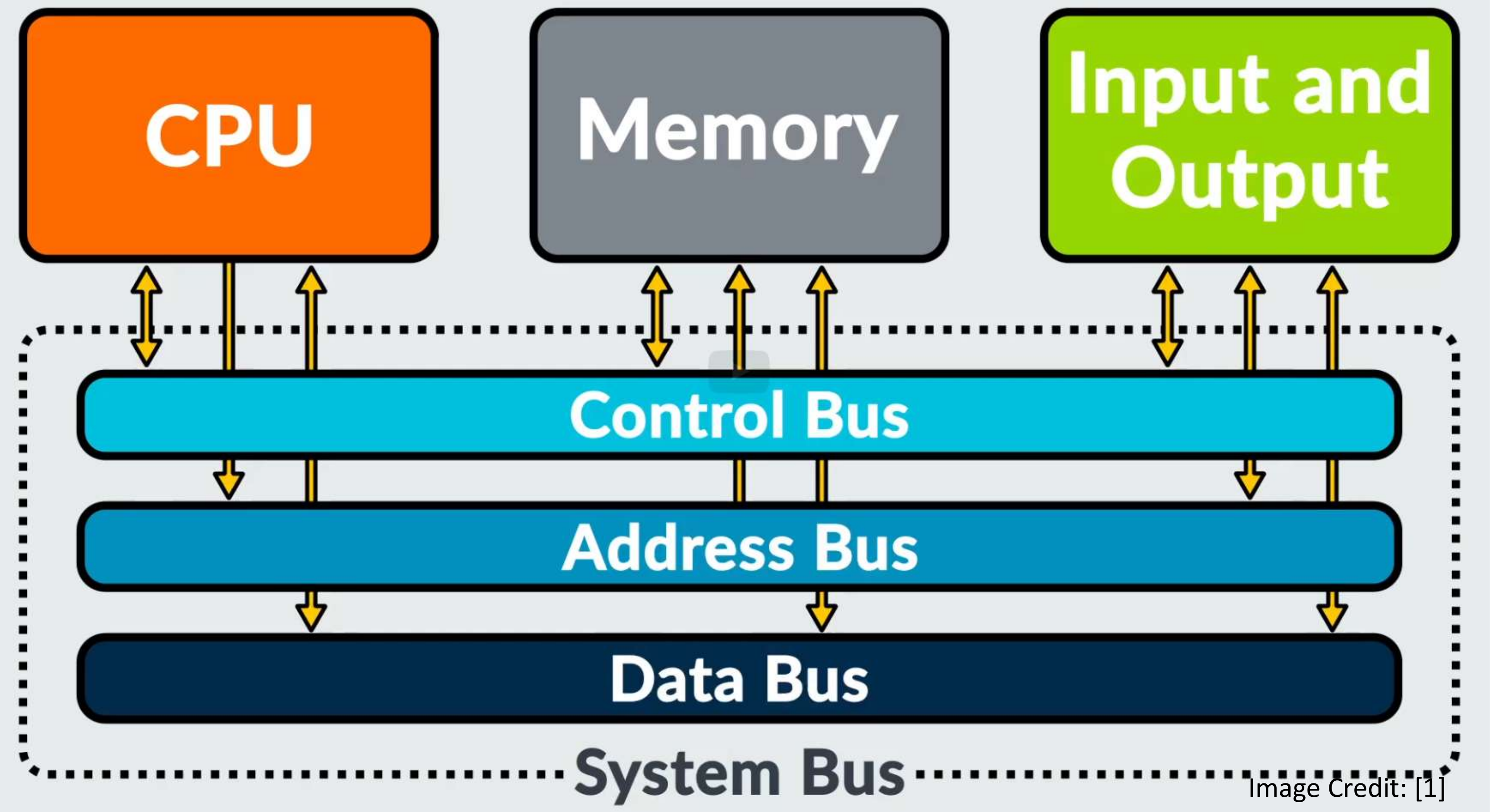


Image Credit: [1]

STM32F107

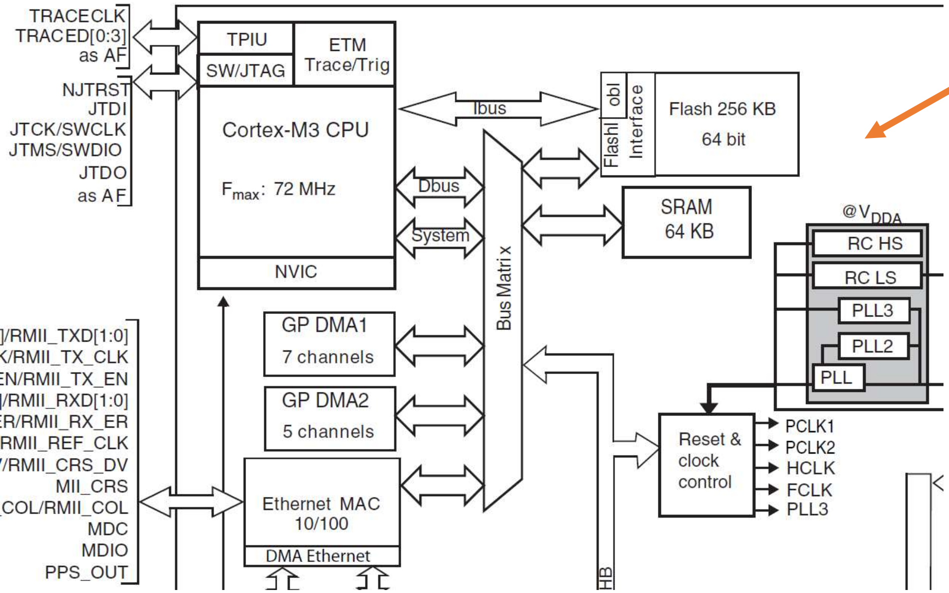
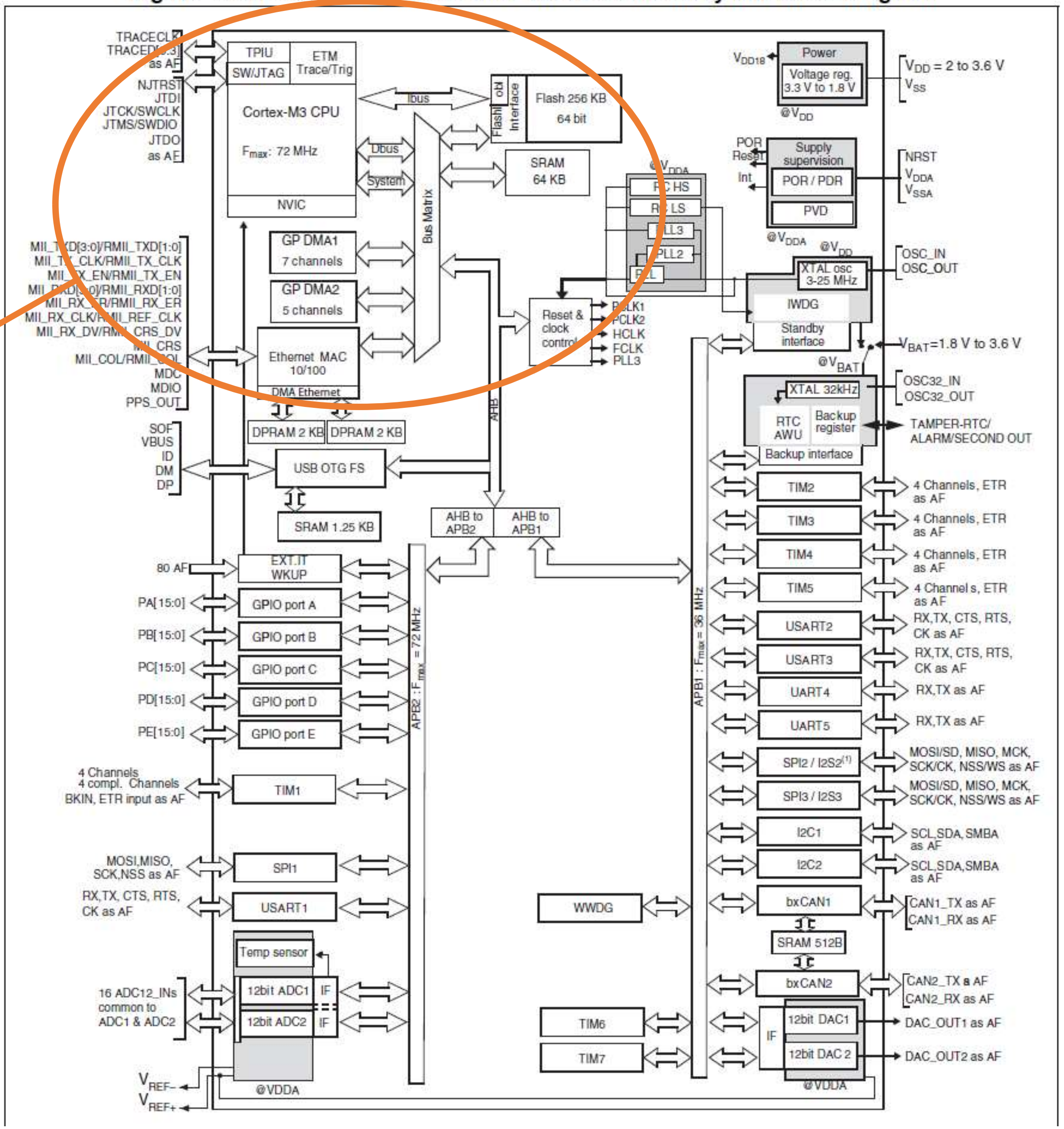


Figure 1. STM32F105xx and STM32F107xx connectivity line block diagram



D-Bus, I-Bus, and S-Bus

1. D-Bus (Data Bus)

- The Data Bus is a communication pathway that carries data between the CPU and memory (RAM).
- It allows bidirectional data transfer, ensuring the CPU can read from and write to memory.
- Data Bus width (e.g., 8-bit, 16-bit, 32-bit) determines the amount of data transferred at once.

2. I-Bus (Instruction Bus)

- The Instruction Bus is responsible for carrying instructions from memory to the CPU.
- It's a unidirectional bus, transmitting instructions from memory to the CPU's Instruction Register.
- It ensures that the CPU fetches and executes instructions in the correct order.

3. S-Bus (System Bus)

- The System Bus is a comprehensive communication network connecting various computer components.
- It includes the Data Bus, Address Bus, and Control Bus.
- The S-Bus facilitates communication between the CPU, memory, input/output devices, and other subsystems.

4. AHB (Advanced High-Performance Bus)

- AHB is an advanced bus architecture used for high-speed data transfer.
- Connects high-speed peripherals and memory.
- Optimized for efficient data transmission.

5. APB (Advanced Peripheral Bus)

- APB connects lower-speed peripherals to the CPU.
- Designed for slower devices, reducing power consumption.
- Supports various peripheral interfaces.

More: ARM Cortex M Bus Protocols & Bus Interfaces

Fetch-Decode-Execute Cycle

1. Fetch (Instruction Fetch)

- The CPU fetches the next instruction from memory.
- The program counter (PC) keeps track of the memory address of the next instruction.
- The fetched instruction is stored in the instruction register (IR).

2. Decode (Instruction Decode)

- The CPU's control unit decodes the instruction in the IR.
- Decoding determines the operation to be performed and the operands involved.
- This phase prepares the CPU for the upcoming execution.

3. Execute (Instruction Execution)

- The CPU executes the decoded instruction.
- It performs the specified operation, which may involve data manipulation, arithmetic, or logical operations.
- The result is stored in registers or memory as needed.

4. Repeat

- The cycle repeats, advancing the program counter to fetch and execute the next instruction.
- This process continues until the program's end.

More: [2] - The machine instruction cycle

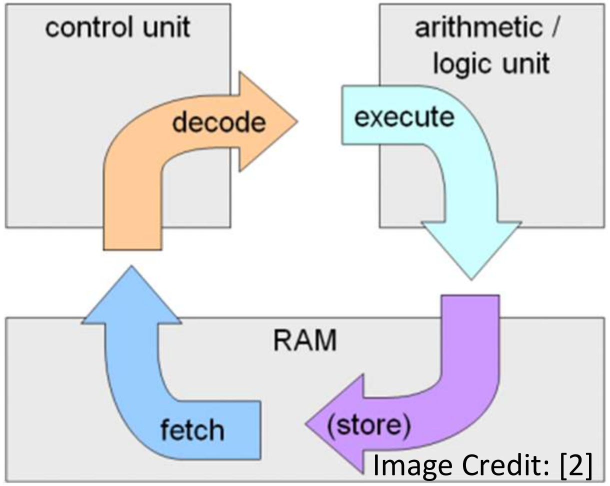
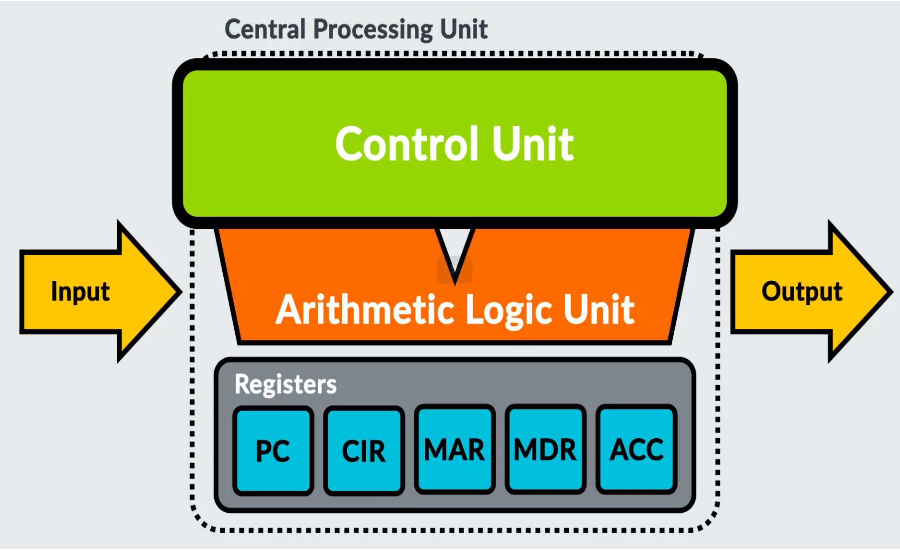
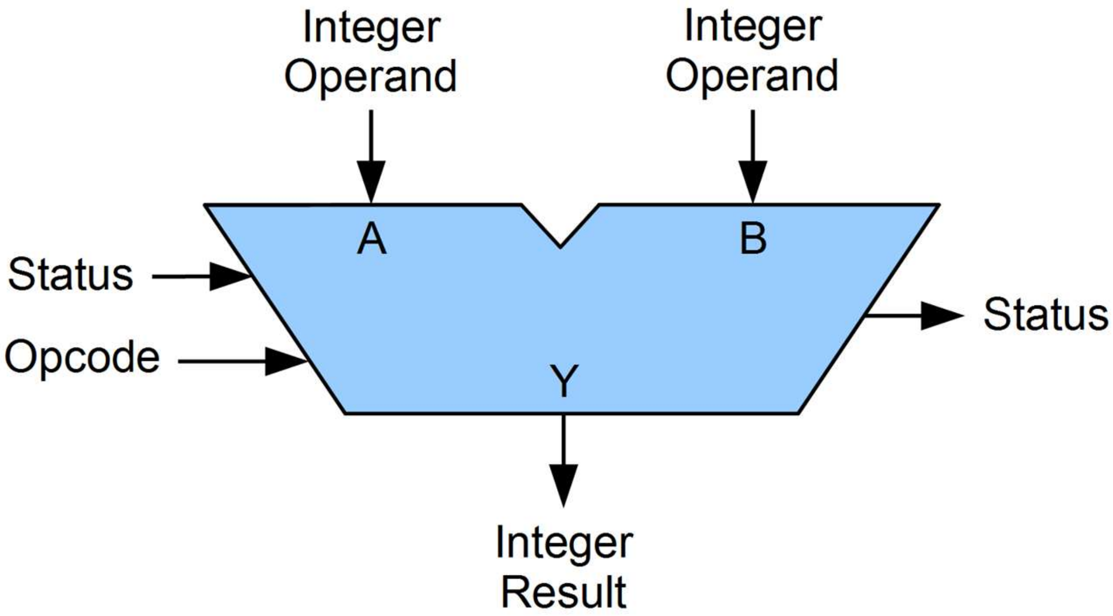


Image Credit: [2]

The ALU and Decode Unit: Input-Process-Output

ADD R0, #1 ;R0=R0+1





Pipeline

		t1	t2	t3	t4	t5	t6	t7	t8	t9
Normal	**User 1**	Wash	Dry	Iron						
	User 2				Wash	Dry	Iron			
	User 3							Wash	Dry	Iron

		t1	t2	t3	t4	t5
Pipelined	**User 1**	Wash	Dry	Iron		
	User 2		Wash	Dry	Iron	
	User 3			Wash	Dry	Iron

Instruction Set Architecture (ISA)

ISA is part of the abstract model of a computer that defines how the CPU is controlled by the software [3]. It is architecture-specific (e.g., x86, Arm, MIPS).

- **Word Size:** Determines the number of bits processed at once (e.g., 32-bit or 64-bit).
- **Memory Address Modes:** Specify how the CPU accesses memory (e.g., direct or indirect).
- **Processor Registers:** ISA defines the set of registers available within the microprocessor.
- **Data Types:** Formats for representing data (e.g., integers, floating-point).

[3]: What Is an Instruction Set Architecture?

Types of Instruction Set Architectures

RISC (Reduced Instruction Set Computer):

- Uses a smaller set of simple and efficient instructions.
- Prefers faster, single-cycle operations.
- Common in modern CPUs for better performance.

CISC (Complex Instruction Set Computer):

- Provides a larger, more diverse set of instructions.
- Supports multi-cycle and complex operations.
- Balances coding simplicity with hardware complexity.

Choosing the Right Architecture

- The choice of instruction set architecture affects CPU design and performance.
- Different architectures cater to various computing needs.
- Selection depends on factors like efficiency, power usage, and application requirements.

Processor Instructions and CPU Architectures

Processors may implement instructions differently.

```
ADD   R0, #1  ;R0=R0+1
```

Single-Cycle:

- Executes instructions in a single instruction cycle.
- Emphasizes efficiency and speed.

Multi-Cycle:

- Executes instructions in multiple stages.
- Can involve fetching, processing, and storing data.
- Offers flexibility but may take longer.

The choice of instruction design impacts CPU efficiency.

RISC vs. CISC

- **RISC (Reduced Instruction Set Computer) :**
  - Prioritizes single-cycle, simple instructions.
  - Designed for efficient, streamlined processing.

- **CISC (Complex Instruction Set Computer):**
  - Allows for more complex, often multi-cycle instructions.
  - Balances simplicity of coding with development complexity.

Multiplying Two Numbers in Memory

Let's say we want to find the product of two numbers - one stored in location 2:3 and another stored in location 5:2 - and then store the product back in the location 2:3 [4].

CISC

```
MULT 2:3, 5:2
```

RISC

```
LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A
```

More: [4] - RISC vs CISC (Crystal Chen, Greg Novick, and Kirk Shimano)

CISC VS RISC

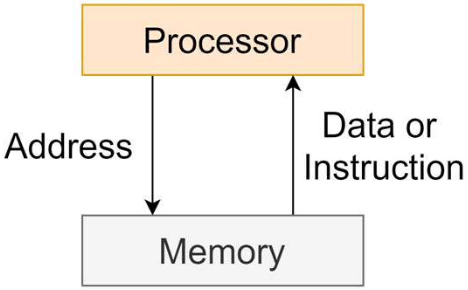
CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

Table taken from [3].

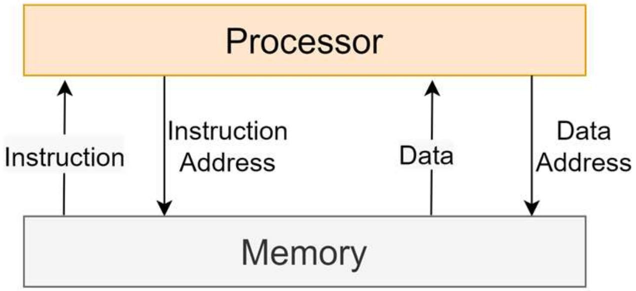
Computer Architecture

- There are two types of computer architecture: **Von Neumann** architecture and **Harvard** architecture.
- The Cortex M processor has a Harvard architecture with separate code and data busses.

Von Neumann



Harvard



Von Neumann vs Harvard

Von Neumann	Harvard
Same physical memory address is used for instructions and data.	Separate physical memory address is used for instructions and data.
It has only one set of data bus and address bus shared by the instruction memory and the data memory	It has a dedicated set of data bus and address bus for the instruction memory and the data memory.
The speeds of the processors are significantly lower.	It is faster than Von Neumann. An instruction is can be executed in a single cycle.
It is cheaper in cost.	It is costly than Von Neumann Architecture.
CPU can not access instructions and read/write at the same time.	CPU can access instructions and read/write at the same time.

How do computers read code?

- https://www.youtube.com/watch?v=QXjU9qTsYCc&ab_channel=FrameofEssence

Q&A