TASK 11

Overview of Microcontroller Unit (MCU)

- **Microcontroller definition**

A microcontroller (MCU) is a small computer on a single integrated circuit that is designed to control specific tasks within electronic systems. It combines the functions of a central processing unit (CPU), memory, and input/output interfaces, all on a single chip.

A microcontroller includes:

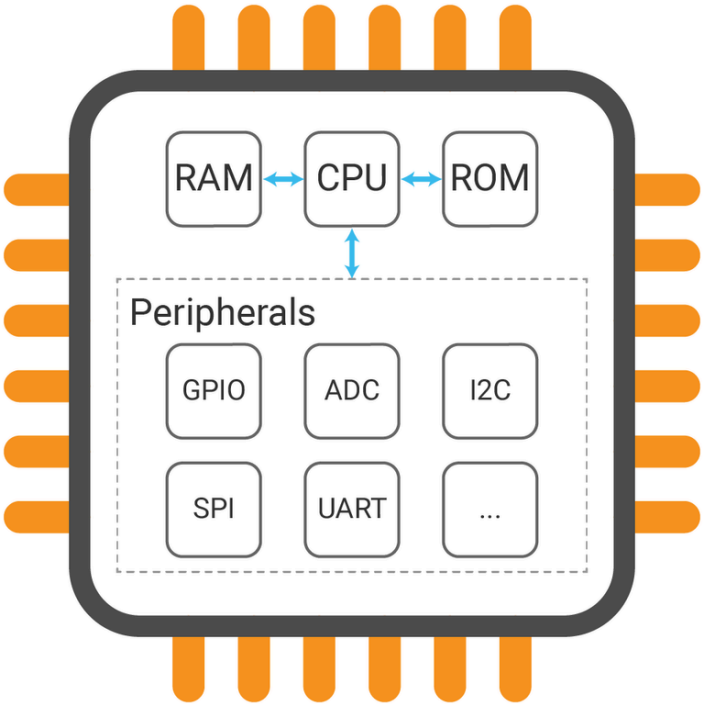
- A processor that runs instructions.
- Memory to store programs and data.
- Input/Output ports to connect with buttons, sensors, screens, etc.
- Communication ports (to connect with other devices).

peripherals on MCU

Peripheral: A device that aids the microprocessor in accomplishing a given job.

The common peripherals on many MCUs are GPIO, Timer, PWM, ADC, UART, SPI, I2C... For example, GPIO allows the MCU to read and write digital signals and I2C, SPI or UART allows it to communicate with other devices. By using peripherals, the MCU can perform a wide range of work.

MCUs often have reserved memory space for peripherals, called **peripheral register**, which is used to store the configuration and operating parameters for the peripherals.

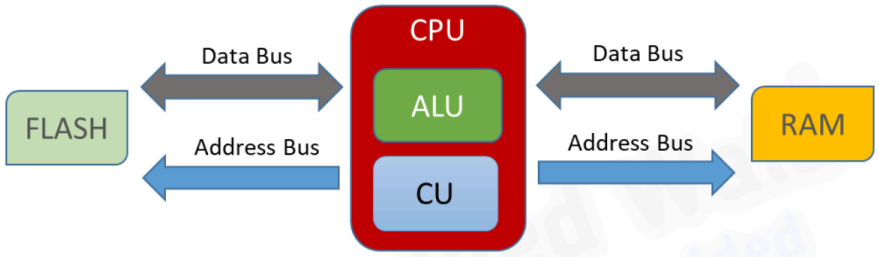


With the memory map technique, each peripheral is assigned a unique address in one unified memory space. **The CPU can write and read data to and from its register to access the specified peripheral**. In this way, you can use and control the peripherals as needed. Normally, the clock (the operating frequency of the peripheral) and power (whether the peripheral is active or in a low-power state) of each peripheral are required to be configured. **Memory mapping** will be discussed in detailed latter on.

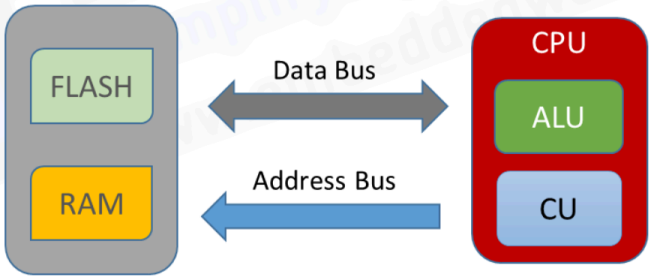
- **MCU Architecture**
  INTERNAL ARCHITECTURE :

• All MCs use one of two basic design models: Harvard Architecture and von-Neumann architecture.

• They represent two different ways of exchanging data between CPU and memory.



Harvard Architecture



Von-Neumann Architecture

- **Von Neumann Architecture → Program and data share the same memory and bus (simpler, but may cause bottlenecks).**

- **Harvard Architecture → Separate memories and buses for program and data (higher performance, used in many MCUs like ARM Cortex).**

Modern MCUs are typically modified Harvard, allowing simultaneous instruction fetch and data access.

- **MCU Clock System**

An MCU (microcontroller) clock system provides synchronized, timed signals to all internal components, enabling the microcontroller to operate and perform tasks at specific

frequencies

🔹 The system clock acts like the heartbeat of the MCU, providing timing signals for its operation.

🔹 It can be generated in two main ways:

1. *Crystal Oscillator* – Uses a quartz crystal that vibrates when electricity is applied.

✅ High precision and stability.

❌ More expensive.

2. *RC Oscillator* – Uses a resistor and capacitor charging/discharging cycle.

✅ Cheaper and simpler.

❌ Less accurate and less consistent across devices.

- **MCU Memory Mapping**

In a Microcontroller Unit (MCU), all memory and peripherals are laid out in a single address space. The CPU “sees” everything—Flash, SRAM, peripherals just as addresses. This is called memory-mapped architecture.

1. Main Memory Regions in MCU

- Flash memory (non-volatile) : Stores the program code (firmware) and sometimes constant data.
- SRAM (volatile) : Stores variables, stack, and heap .
- Peripheral registers (memory-mapped I/O) : Each GPIO, UART, SPI, Timer, etc.has control registers at fixed addresses.

2. How Do We Know These Ranges? → **TRM**

The Technical Reference Manual (TRM) or Datasheet of the MCU defines the exact address ranges.

- **MCU Bus Interfaces**

The bus system connects CPU, memory, and peripherals

- Data bus → transfers data.
- Address bus → specifies memory/peripheral location.
- Control bus → carries control signals (read/write, enable, etc.).

MCUs may have multiple buses (e.g., instruction bus, data bus, system bus) to reduce bottlenecks and improve performance.

- **Advanced Microcontroller Bus Architecture (AMBA)**

Developed by ARM, AMBA standardizes MCU bus communication.

- AHB (Advanced High-performance Bus) → high-speed CPU, DMA, memory.
- APB (Advanced Peripheral Bus) → simple, low-power peripherals (UART, GPIO).
- AXI (Advanced eXtensible Interface) → high-end, parallel transactions (used in Cortex-A, SoCs).

This layered bus system balances *performance, cost, and power efficiency.