10th Task in Embedded Systems

◈ Embedded System:

- **what is Embedded System?**

  -It's dedicated computer system designed to perform one or a few specific tasks, usually within a larger device.

  -It consists of (microcontroller-software-real time- input/output devices).

- **Embedded System Characteristics:**

  -single functioned

  -tightly constrained

  -Memory

  -microprocessor based

  -Reactive and real time

- **Embedded Systems: Advantages and Disadvantages:**

  Advantages:

  Cost-effective: Low production cost.

  Energy-efficient: Consumes minimal power.

  High performance: Optimized for specific tasks.

  Customizable: Can be tailored to specific needs.

  Disadvantages:

  Resource constraints: Limited memory and processing power.

  Complex development: Requires significant design effort.

  Scalability issues: Difficult to upgrade or expand.

  Debugging challenges: Hard to troubleshoot due to hardware dependence.

- **Embedded Systems Applications:**

  -Automotive(Cruise control, ABS, airbag systems)

  -Networking(Routers, switches)

  -Finance & Retail(ATMs, POS terminals)

  -Home Appliances(Smart thermostats, washing machines)

  -Healthcare (Pacemakers, fitness trackers)

  -Consumer Electronics(Smartphones, smart TVs)

◈ Microprocessor:

- **What is Microprocessor?**

  -An integrated circuit that performs (Arithmetic operations- Logical operations- Control functions).

  -Also called the **CPU** (Central Processing Unit)

  -Acts as the "brain" of computing devices

  -Executes program instructions

  -Fundamental component in microcontrollers

- **Processor Building Blocks:**

  1. ALU (Arithmetic Logic Unit): Performs all arithmetic and logical operations .

  2. Control Unit (CU): Directs how data moves and how instructions are executed.

  3. Registers: Small, fast storage used for immediate data processing.

  4. Internal Buses: Connect internal components to transfer data:

  -Data Bus: Transfers data

  -Address Bus: Transfers memory addresses

  -Control Bus: Transfers control signals

- **CPU Registers:**

  -Fastest memory type in a CPU

  -Essential for processor operations

  General Purpose Registers

  -Numbered R1-Rn (typically 8-32 registers)

  -Used for temporary data storage

  Special Purpose Registers:

  -Program Counter (PC): Holds address of next instruction

  -Accumulator (ACC): Stores frequently used data

  -Instruction Register (IR) :Holds current instruction

  -Status Register:Contains processor state flags

  -MAR/MDR:Handles memory read/write operations

- **Instruction Life Cycle:**

  Fetch: CPU reads an instruction from memory (address held in the Program Counter).
  Decode: The Control Unit interprets the instruction and identifies what to do.
  Execute: The operation is performed (ALU or memory operation).

- **Instruction Set Architecture:**

  It defines: What instructions a processor can execute - How data is accessed and manipulated - Types of operands and operations.

  Types of Instructions:

  - **Data Transfer:** MOV, LOAD, STORE
  - **Arithmetic/Logic:** ADD, SUB, AND
  - **Control Flow:** JMP, CALL, RET
  - **I/O Instructions:** IN, OUT

  Examples:

  - RISC (Reduced Instruction Set Computer): ARM, MIPS

  - CISC (Complex Instruction Set Computer): x86

◈ Memory Types in Embedded Systems:

RAM (Random Access Memory):

- Volatile (erased on power off)
- Stores temporary variables and runtime data

ROM (Read Only Memory):

- Non-volatile
- Stores permanent program code

EEPROM / Flash:

- Rewritable non-volatile memory
- Used to store firmware or user settings

Cache Memory:

- Very fast memory close to the CPU
- Stores frequently accessed data/instructions

◈ Introduction to computer Architecture:

- **System Architecture Types:**

  Von Neumann Architecture:

  - Single memory for both data and program

  - Simpler but slower due to bottleneck

  - **Harvard Architecture:**

    - Separate memories for data and code

    - Faster because instructions and data can be fetched simultaneously

- **Timing:**

  -Controlled by a clock signal

  -Each instruction takes a number of clock cycles

  -Precise timing is essential for real-time systems, where delay or jitter can cause failure (airbags, medical devices)