MCU Clock System & Interrupts

1. MCU Clock System: -

The MCU clock system provides timing signals to coordinate all operations in the microcontroller.
It can be sourced from:

- **Internal RC Oscillator: Cost-effective but less accurate.**
- **External Crystal Oscillator: More precise for timing-critical applications.**
- **PLL (Phase Locked Loop): Multiplies frequency for higher performance.**

Clock configuration impacts speed, power consumption, and peripheral operation.

2. Interrupt Fundamentals & Architecture: -

Interrupts are signals that temporarily stop the CPU's current execution to handle important events.
Types of interruptions:

- **Hardware Interrupts: Triggered by peripherals (e.g., timers, GPIO, communication modules).**
- **Software Interrupts: Triggered by instructions in the code.**

Interrupt architecture includes an interrupted vector table mapping each interrupt source to its service routine.

3. Interrupt Handling & Startup Process:-

When an interrupt occurs:

- **CPU finishes the current instruction.**
- **Program counter (PC) is saved.**
- **CPU jumps to the Interrupt Service Routine (ISR) address from the vector table.**
- **ISR executes and then returns to the previous task using a "return from interrupt" instruction.**

Startup process involves configuring interrupt priorities, enabling global and specific interrupt sources.

4. Software Mechanisms for Managing Interrupts:-

- **Interrupt Enable/Disable: Control whether interrupts are processed.**
- **Prioritization: Assigns importance levels to different interrupts.**
- **Nested Interrupts: Allow higher-priority interrupts to preempt lower-priority ones.**
- **Debouncing: Prevents false triggers from noisy signals.**
- **ISR Design Best Practices:**
  - *** Keep ISR short to avoid delays.**
  - ***Avoid heavy processing in ISR; defer work to main loop or tasks.**