Chapter 5: Interrupt Operations

An interrupt is a signal sent by a peripheral device or an internal component to the processor, requesting immediate attention. Interrupts enable the microprocessor to handle real-time tasks efficiently by allowing it to pause its current execution and service the interrupting device before resuming normal operations. Interrupts improve system performance by eliminating the need for continuous polling, where the processor repeatedly checks device status, thereby wasting processing power.

Interrupts can be classified into two main types: hardware interrupts and software interrupts. Hardware interrupts are triggered by external devices such as keyboards, timers, and communication ports. The 8085 microprocessor has five hardware interrupts: TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. The TRAP interrupt has the highest priority and is non-maskable, meaning it cannot be disabled. It is used for critical operations such as handling power failures. The RST 7.5, RST 6.5, and RST 5.5 interrupts have decreasing priority levels and can be enabled or disabled through software instructions. The INTR interrupt is the lowest priority interrupt and requires an external controller to determine the ISR address.

Software interrupts are triggered by program instructions and are used for system calls and debugging. The 8085 microprocessor provides eight software interrupts, known as RST instructions (RST 0 to RST 7), which allow the execution of predefined routines stored in memory.

When an interrupt occurs, the microprocessor follows a specific sequence to handle it. First, the processor completes the execution of the current instruction before acknowledging the interrupt. It then saves the contents of the program counter (PC) and status registers to preserve the execution state. The processor then transfers control to the corresponding interrupt service routine (ISR), which is a predefined set of instructions that handle the interrupt request. After executing the ISR, the processor restores the previously saved register values and resumes execution of the interrupted program.

Interrupt handling can be implemented using vectored and non-vectored interrupts. Vectored interrupts have predefined memory addresses for their corresponding ISRs, allowing for faster execution. Non-vectored interrupts require additional processing to determine the ISR address. Maskable interrupts can be enabled or disabled through software, while non-maskable interrupts (NMIs) have the highest priority and cannot be disabled.

The 8259 Programmable Interrupt Controller (PIC) is used to manage multiple interrupts efficiently. It prioritizes interrupt requests and sends them to the processor based on predefined priority levels. The PIC enables nested interrupt processing, where higher-priority interrupts can interrupt lower-priority ones.

Interrupts are essential in real-time computing, embedded systems, and multitasking operating systems, enabling efficient handling of multiple concurrent tasks. Modern microprocessors implement advanced interrupt handling techniques, including interrupt vector tables, nested interrupts, and priority-based scheduling, to ensure system responsiveness and stability.