

3.5 Interrupts

An interrupt is an external or internal event that disturbs the microcontroller to inform it that a device needs its service. The program which is associated with the interrupt is called the interrupt service routine (ISR) or interrupt handler. Upon receiving the interrupt signal the Microcontroller, finish current instruction and saves the PC on stack. Jumps to a fixed location in memory depending on type of interrupt Starts to execute the interrupt service routine until RETI (return from interrupt) Upon executing the RETI the microcontroller returns to the place where it was interrupted. Get pop PC from stack.

The 8051 microcontroller has FIVE interrupts in addition to Reset. They are

- Timer 0 overflow Interrupt
- Timer 1 overflow Interrupt
- External Interrupt 0(INT0)
- External Interrupt 1(INT1)
- Serial Port events (buffer full, buffer empty, etc) Interrupt

Each interrupt has a specific place in code memory where program execution (interrupt service routine) begins.

- External Interrupt 0 : 0003 H
- Timer 0 overflow : 000B H
- External Interrupt 1 : 0013 H
- Timer 1 overflow : 001B H
- Serial Interrupt : 0023 H

Upon reset all Interrupts are disabled & do not respond to the Microcontroller. These interrupts must be enabled by software in order for the Microcontroller to respond to them. This is done by an 8-bit register called Interrupt Enable Register (IE).

Interrupt Enable Register :

EA	—	ET2	ES	ET1	EX1	ET0	EX0
-----------	---	------------	-----------	------------	------------	------------	------------

- EA : Global enable/disable. To enable the interrupts this bit must be set High.
- --- : Undefined-reserved for future use.
- ET2 : Enable /disable Timer 2 overflow interrupt.
- ES : Enable/disable Serial port interrupt.
- ET1 : Enable /disable Timer 1 overflow interrupt.
- EX1 : Enable/disable External interrupt1.
- ET0 : Enable /disable Timer 0 overflow interrupt.
- EX0 : Enable/disable External interrupt0

Upon reset the interrupts have the following priority.(Top to down). The interrupt with the highest PRIORITY gets serviced first.

1. External interrupt 0 (INT0)
2. Timer interrupt0 (TF0)
3. External interrupt 1 (INT1)
4. Timer interrupt1 (TF1)
5. Serial communication (RI+TI)

Priority can also be set to “high” or “low” by 8-bit IP register.- Interrupt priority register

—	—	PT2	PS	PT1	PX1	PT0	PX0
---	---	------------	-----------	------------	------------	------------	------------

IP.7: reserved

IP.6: reserved

IP.5: Timer 2 interrupt priority bit (8052 only)

IP.4: Serial port interrupt priority bit

IP.3: Timer 1 interrupt priority bit

IP.2: External interrupt 1 priority bit

IP.1: Timser 0 interrupt priority bit

IP.0: External interrupt 0 priority bit