INTERRUPTS OF 8051

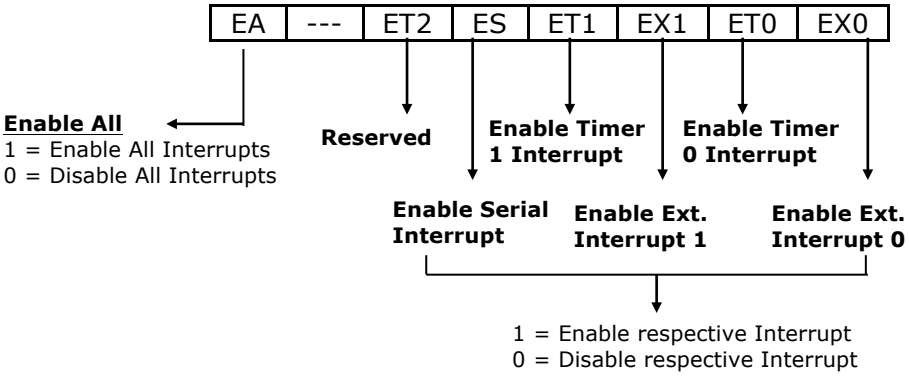
8051 supports **5** interrupts.
2 External Interrupts are on the following pins

$\overline{\text{INT1}}$

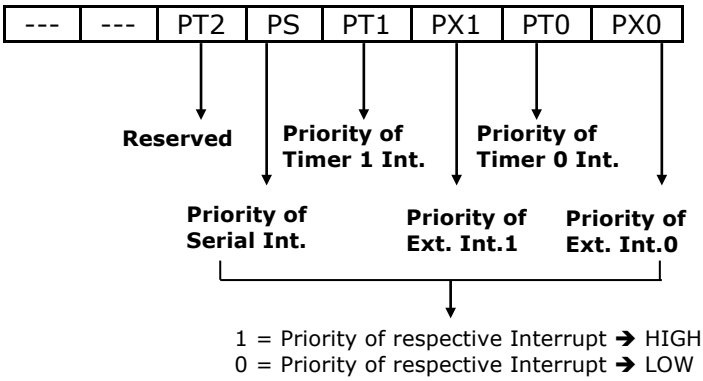
$\overline{\text{INT0}}$

2 Internal Timer interrupts are:
Timer 1 Overflow Interrupt
Timer 0 Overflow Interrupt
1 Serial Port Interrupt (Common for RI or TI)
All interrupts are **vectored** i.e. they cause the program to execute an ISR from a pre-determined address in the Program Memory. #Please refer Bharat Sir's Lecture Notes for this ...
Interrupts are controlled mainly by **IE** and **IP** SFR's and also by some bits of **TCON** SFR.

IE - Interrupt Enable (SFR) [Bit-Addressable As IE.7 to IE.0]



IP - Interrupt Priority (SFR) [Bit-Addressable As IP.7 to IP.0]



Timer Overflow Interrupts (TF1 and TF0)

When any of the 2 **Timers overflow**, their respective bit **TFX** (TF1 or TF0) is **set** in **TCON** SFR.
If Timer Interrupts are enabled then the **timer interrupt occurs**.
The **TFX** bits are **cleared** when their respective **ISR** is executed.

Serial Port Interrupt (RI or TI)

While receiving serial data, when a **complete byte** is **received** the **RI** (receive interrupt) bit is set in the **SCON.**
During transmission, when a **complete byte** is **transmitted** the **TI** (transmit interrupt) bit is set in the **SCON**.
ANY of these events can cause the **Serial Interrupt** (provided Serial Interrupt is enabled).
The **RI/TI** bit is **not cleared** automatically on **executing** the **ISR**. The program should **explicitly clear** this bit to allow further Serial Interrupts.

External Interrupts ($\overline{\text{INT1}}$ and $\overline{\text{INT0}}$)

Pins $\overline{\text{INT1}}$ and $\overline{\text{INT0}}$ are inputs for external interrupts.

These interrupts can be -ve **edge** or low-**level triggered** depending upon the **IT0 and IT1** bit in **TCON** SFR. (ITX = 1 ➔ -ve edge triggered)
Whew any of these interrupts occur the respective bits **TE1** or **IE0** are **set** in the **TCON** SFR.
If External Interrupts are enabled then the **ISR** is **executed** from the respective address.

Interrupt Sequence

The following sequence is executed to service an interrupt:
Address of next instruction of the main program i.e. **PC** is **Pushed** into the **Stack**.
All **interrupts** are **disabled**, by making EA bit in IE SFR ← 0.
Program Control is shifted to the **Vector Address** (location) of the **ISR**.
The **ISR begins**.

Returning Sequence

RETI instruction denotes the **end** of the **ISR**.
It causes the processor to **POP** the contents of the Stack Top into the **PC.**
It also re-enables interrupts by making EA bit in IE SFR ← 1.
The **main program resumes**.

Interrupt Priorities

8051 has only **two priority levels** for the interrupts: **Low** and **High**.
Interrupt priorities are set using the **IP** SFR.
As the name suggests, a high priority interrupt can interrupt a low priority interrupt.
It two or more interrupts at the same level occur simultaneously then priorities are decided as follows:

INTERRUPT	PRIORITY	VECTOR ADDRESS
$\overline{\textbf{INT0}}$	1	00**03**H
TF0	2	00**0B**H
$\overline{\textbf{INT1}}$	3	00**13**H
TF1	4	00**1B**H
Serial (RI or TI)	5	00**23**H

DIAGRAM FOR INTERRUPTS... OPTIONAL
DRAW ONLY IF ASKED