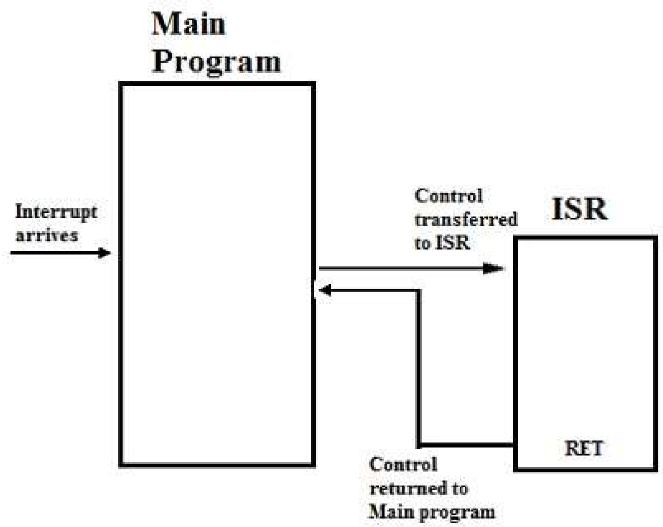
Hardware Interrupts

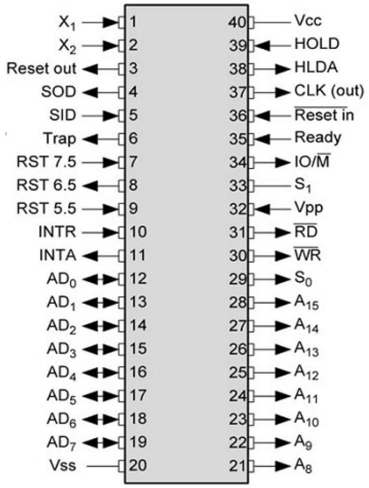
In the microprocessor based system the interrupts are used for data transfer between the peripheral devices and the microprocessor.



A small program or a routine that when executed services the corresponding interrupting source is called as an ISR.

The Interrupt Vector Table (IVT) is located between 00-FFH

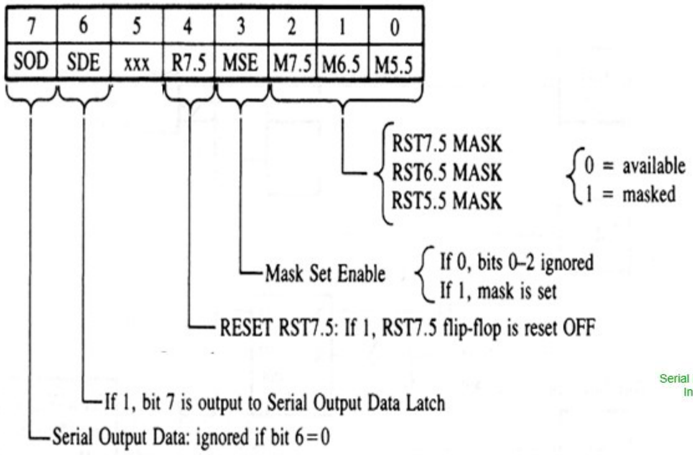
Q. So what is the max. no of interrupts?



1. A peripheral sends interrupt requests to the Microprocessor.
2. After accepting the interrupts Microprocessor send the INTA (active low) signal to the peripheral.
3. The vectored address of particular interrupt is stored in program counter.
4. The processor executes an interrupt service routine (ISR) addressed in program counter.

1.TRAP	0024H	NMI	Edge & Level
2.RST7.5	003CH	MI	Edge
3.RST6.5	0034H	MI	Level
4.RST5.5	002CH	MI	Level
5.INTR			

Priority



7	6	5	4	3	2	1	0
SOD	SDE	xxx	R7.5	MSE	M7.5	M6.5	M5.5

RST7.5 MASK
RST6.5 MASK
RST5.5 MASK

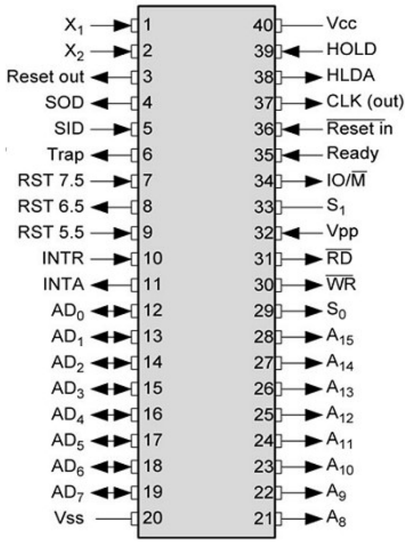
0 = available
1 = masked

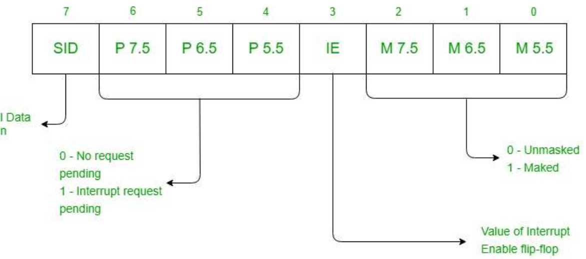
Mask Set Enable
If 0, bits 0–2 ignored
If 1, mask is set

RESET RST7.5: If 1, RST7.5 flip-flop is reset OFF

If 1, bit 7 is output to Serial Output Data Latch

Serial Output Data: ignored if bit 6=0





7	6	5	4	3	2	1	0
SID	P 7.5	P 6.5	P 5.5	IE	M 7.5	M 6.5	M 5.5

Serial Data In

0 - No request pending
1 - Interrupt request pending

0 - Unmasked
1 - Maked

Value of Interrupt Enable flip-flop

```
EI
MVI A, <B>
SIM
```

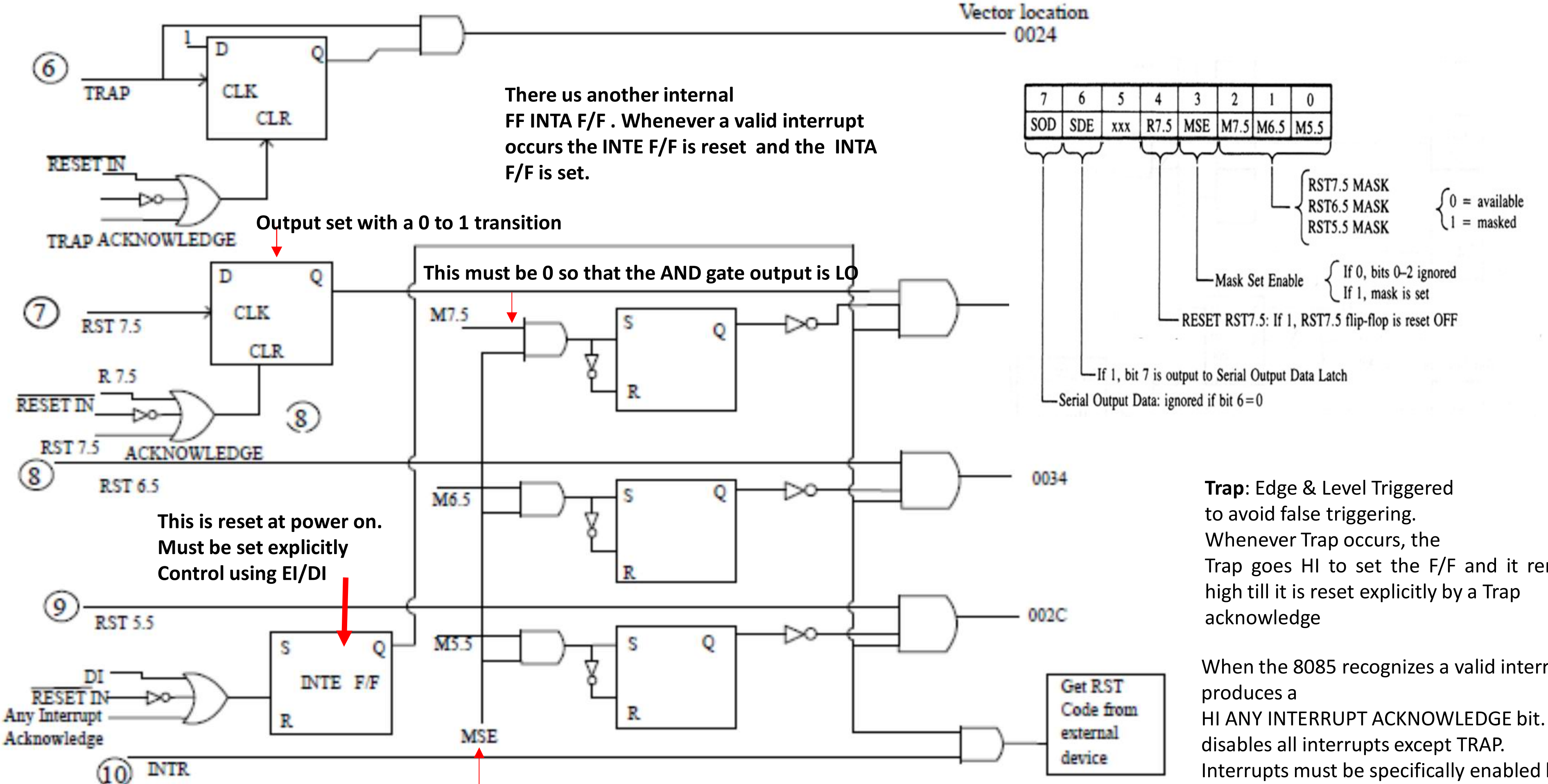
Enable Interrupt (EI) – The interrupt enable flip-flop is set and all interrupts are enabled following the execution of next instruction followed by EI. No flags are affected. After a system reset, the interrupt enable flip-flop is reset, thus disabling the interrupts. This instruction is necessary to enable the interrupts again (except TRAP).

Disable Interrupt (DI) – This instruction is used to reset the value of enable flip-flop hence disabling all the interrupts. No flags are affected by this instruction.

Set Interrupt Mask (SIM) – It is used to implement the hardware interrupts (RST 7.5, RST 6.5, RST 5.5) by setting various bits to form masks or generate output data via the Serial Output Data (SOD) line. First the required value is loaded in accumulator then SIM will take the bit pattern from it.

Read Interrupt Mask (RIM) – This instruction is used to read the status of the hardware interrupts (RST 7.5, RST 6.5, RST 5.5) by loading into the A register a byte which defines the condition of the mask bits for the interrupts. It also reads the condition of SID (Serial Input Data) bit on the microprocessor.

Interrupt Structure of 8085A



7	6	5	4	3	2	1	0
SOD	SDE	xxx	R7.5	MSE	M7.5	M6.5	M5.5

There us another internal FF INTA F/F . Whenever a valid interrupt occurs the INTE F/F is reset and the INTA F/F is set.

Output set with a 0 to 1 transition

This must be 0 so that the AND gate output is LO

This is reset at power on. Must be set explicitly Control using EI/DI

This must be 1 so that the AND gate is enabled

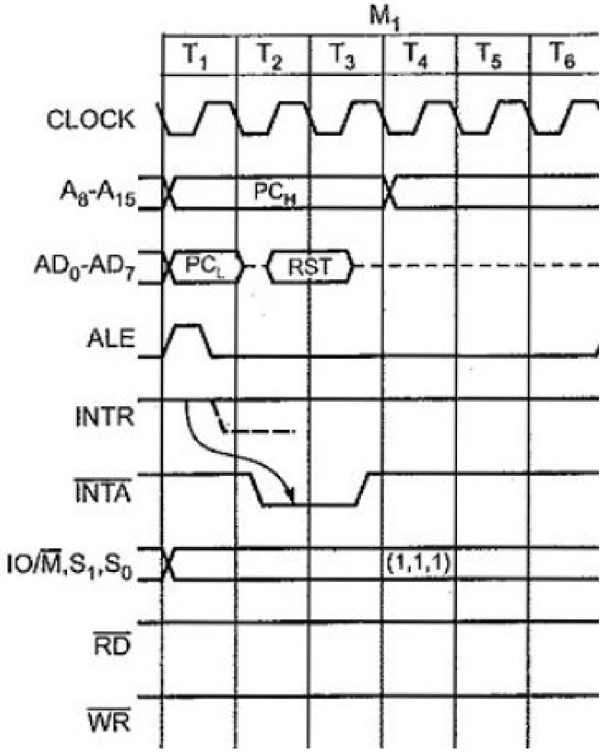
Trap: Edge & Level Triggered to avoid false triggering. Whenever Trap occurs, the Trap goes HI to set the F/F and it remains high till it is reset explicitly by a Trap acknowledge

When the 8085 recognizes a valid interrupt it produces a HI ANY INTERRUPT ACKNOWLEDGE bit. This disables all interrupts except TRAP. Interrupts must be specifically enabled by EI at the end of a ISR before RET

Interrupts are sensed by the 8085 one cycle before the end of execution of an instruction. The longest instruction takes 18 clock cycles. So interrupts must be there for 17 clock cycles at least

Software Interrupts: Restart F Instructions

Mnemonics, Operand	Opcode(in HEX)	In Binary	Bytes	Target Address (n*8)
RST 0	C7	1100 0111	1	0000H
RST 1	CF	1100 1111	1	0008H
RST 2	D7	1101 0111	1	0010H
RST 3	DF	1101 1111	1	0018H
RST 4	E7	1110 0111	1	0020H
RST 5	EF	1110 1111	1	0028H
RST 6	F7	1111 0111	1	0030H
RST 7	FF	1111 1111	1	0038H



The RST instruction has only one interrupt acknowledge cycle of 6 T-states.

Whenever INTR is HI:

T1: The first T state of all the machine cycles involving data transfer is for the demultiplexing of AD0-AD7. INTA remains HI

T2:T3: RST Code is received

T4:T6 : The instruction is decoded and depending on the Instruction further machine cycles are executed

RST n = CALL n*8

For example, the advantage of RST 2 is that it is only 1 Byte, whereas CALL 0010H is 3-Byte long. Thus RST instructions are useful for branching to frequently used subroutines.

Q. Why 8 Bytes? (Hint : CALL 0010H = PUSH PC + JMP 4050H)

Stack will be taken up in next lecture