INTERRUPT

It means to stop the execution of the current program.

Types of interrupts : There are two types of interrupts -

i) Software Interrupts

ii) Hardware Interrupts

I) **Software Interrupts** : Software interrupt are present in the form of instructions.

II) **Hardware interrupts** : Hardware interrupts are present in the physical form.

Difference Between Software Interrupt & Hardware Interrupts

Software Interrupts	Hardware Interrupts
Position of software interrupts are fixed	Position of hardware interrupts are not fixed it may occur anywhere during the execution of the program.

Software interrupt

- In case of software interrupts, the cause of the interrupt is the execution of the instruction.
- The 8085 microprocessor has eight instructions. These eight instructions are RSTn, where(n=0 to7).
- Such interrupts are called as software interrupts.
- They allow microprocessor to transfer program control from the main program to the subroutine program.
- After completing the subroutine program. the program control returns back to the main program.

Formate of RSTn Instruction OPCODE is as follows

D7	D6	D5	D4	D3	D2	D1	D0
1	1	N_2	N_1	N_0	1	1	1

- The difference between two successive location is only 8 bytes. Hence jump instruction must be stored into corresponding location to transfer microprocessor's control to user defined ISR address.
- Software interrupts are not used to handle asynchronous events. They are used to call software routines like single step, break points etc.

Vector location for software interrupt

Interrupt	n*8	vector address	opcode
RST0	0*8 = 0	0000 H	C7 H
RST1	1*8=8	0008 H	CF H
RST2	2*8=16	0010 H	D7 H
RST3	3*8=24	0018 H	DF H
RST4	4*8=32	0020 H	E7 H
RST5	5*8=40	0028 H	EF H
RST6	6*8=48	0030 H	F7 H
RST7	7*8=56	0038 H	FF H

HARDWARE INTERRUPTS

- HARDWARE INTERRUPTS - There are 5 types of Hardware Interrupts TRAP, RST 7.5, RST 6.5, RST 5.5, INTR.

  -These interrupts are present in the form of physical form.

  - These interrupts do not have a fixed position

- **Maskable interrupt** − In this type of interrupt, we can disable the interrupt by writing some instructions into the program. **For example** RST7.5, RST6.5, RST5.5.

- **Non-Maskable interrupt** − In this type of interrupt, we cannot disable the interrupt by writing some instructions into the program. **For example** TRAP.

- **Vector interrupts** − In this type of interrupt, the interrupt address is known to the processor. **For example** RST7.5, RST6.5, RST5.5, TRAP.

- **Non-Vector interrupts** − In this type of interrupt, the interrupt address is not known to the processor so, the interrupt address needs to be sent externally by the device to perform interrupts.**For example** INTR.

TRAP : It is a non-maskable interrupt, having the highest priority among all interrupts. By default, it is enabled until it gets acknowledged.
RST 7.5 : It is a maskable interrupt, having the second highest priority among all interrupts.
RST 6.5 : It is a maskable interrupt, having the third highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to the 0034H address.
RST 5.5 : It is a maskable interruption. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to the 002CH address.
INTR : It is a maskable interrupt, having the lowest priority among all interrupts.

Interrupt type	Trigger	Priority	Maskable	Vector address
TRAP	Edge and Level	1st (Highest)	No	0024 H
RST 7.5	Edge	2nd	Yes	003CH
RST 6.5	Level	3rd	Yes	0034H
RST 5.5	Level	4th	Yes	002CH
INTR	Level	5th (Lowest)	Yes	

Instructions To Control Interrupt

1)EI :- Enable Interrupt .It is a single byte instruction that is opcode only. When microprocessor execute EI instruction SR flip-flop (also known as a interrupt flip-flop) is set, i.e. Q =1 .Therefore all the interrupts are enabled.

2)DI :- Disable Interrupt. It is also a single byte instruction that is opcode only. When microprocessor execute DI instruction SR flip-flop is resetted i.e.Q = 0. Therefore all the interrupts are disabled except TRAP.

3)SIM :- Set Interrupt Mask. It is a single byte instruction that is opcode only. It is a dual purpose instruction. it is used -

i)To send serial data out on a SOD pin.

ii)To mask/ unmask RST 7.5, RST 6.5 and RST 5.5

before giving the SIM instruction, register A is to be loaded with the information as follows -

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SOD	SDE	X	Reset RST 7.5	MSE	M7.5	M6.5	M5.5

-D_7= SOD
If D_7=1, a logic high signal is transferred by the SOD pin.
If D_7=0, a logic low signal is transferred by the SOD pin.

-D_6 =SDE (Serial Data Enable)
if D_6 = 1 then only the serial data transfer is possible as per the D_7 bit.
And if it is zero then there is no serial data transfer.

-D_5 = Don't Care.

-D_4 = Reset RST 7.5 .
if 1 D_4=RST 7.5 is resetted i.e. it will clear the D flip-flop.

-D_3 = MSE (Mask Set Enable).
if D_3 =1, masking of RST 7.5 RST 6.5 RST 5.5 is possible as per the D_2, D_1 , D_0 .
if MSE = 0, masking is not possible.

-D_2 = RST 7.5 is masked .
if D_2 =0, RST 7.5 is unmasked.
In case of the D_1 = 1, RST 6.5 is masked.
If D_1 = 0, RST 6.5 is unmasked.
D_0 is same as D_1

e.g. :- Write the instruction to enable RST 6.5 and disable RST 7.5 and RST 5.5.

0	0	0	0	1	1	0	1

MVI A,0DH
SIM
EI

4) RIM :- Read Interrupt Mask.

It is also a single byte instruction that is a opcode only. It is used -

i) for the status of interrupt.

ii)for serial data bit on SID pin .

After the execution of RIM instruction register A give the information as follows-

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SID	I_7	I_6	I_5	IE	M7.5	M6.5	M5.5

-D_7=SID (Serial Input Data)

If D_7 = 1, then there is logic high signal on SID pin.

If D_6 = 0, then there is logic low signal on SID pin.

-D_6 =I_7

If D_6 = 1,then there is a interrupt on RST 7.5 pin that is RST 7.5 is pending.

If D_6= 0,then RST 7.5 is not pending.

-D_5 =I_6

If it is 1 then RST 6.5 is pending.

If it is zero then RST 6.5 is not pending.

-D_4=I_5

If it is equal to 1 Then RST 5.5 is pending.

If it is equal to zero then RST 5.5 is not pending.

-D_3=I_4 (Interrupt enable)

if D_3=1,then all the interrupts are enabled that means SR flip flop in set mode.

If D_3=0,then interrupts are disabled.

-D_2= M7.5

if D_2=1,then RST 7.5 is masked.

if D_2=0,then RST 7.5 is unmasked.

-D_1=M 6.5

If D_1=1,then RST 6.5 is masked. D_1

If D_1 =0,then RST 6.5 is unmasked.

D_0=M 5.5 same as D_2.

5) INTR:- INTR is a non vector interrupt.it is having its own external hardware circuit. When the interrupt is received on INTR pin, after the execution of current instruction. Microprocessor will generate a low pulse on INTA’ pin and it will enter in the interrupt acknowledgement cycle.