Real Time System

Third Level

Lecture Fourteen

Interrupt

RealTime Systems.
Dr. Osama Abbas Hussein

Goals
Up-on completing this lecture, the student should be able to:

1- Understand how interrupts are used in RTs
2- Interrupt types in CPU

Interrupt is a signal send by an external device to the processor to perform a particular task or work (Interrupt is a process where an external device can get the attention of the microprocessor, the process starts from the I/O device).

When a peripheral is ready for data transfer, it interrupts the processor by sending an appropriate signal to the interrupt pin of the processor. If the processor accepts the interrupt then the processor suspends its current activity and executes an interrupt service routine to complete the data transfer between the peripheral and processor. After executing the interrupt service routine the processor resumes its current activity.

Interrupts increase processor system efficiency by letting I/O device request CPU only when that device needs immediate attention.

- When a microprocessor is interrupted, it stops executing its current program and calls a special routine which "services" the interrupt.
- The event that causes the interruption is called **Interrupt**
- The special routine executed to service the interrupt is called **ISR - Interrupt Service Routine.**

Classification of Interrupts:-

Interrupts can be classified into the following:

- **Hardware Interrupt** an interrupt caused by an "External Signal "
- **Software Interrupt** an interrupt caused by "Special Instruction"
- **Maskable Interrupts** (Can be delayed or rejected)
- **Non-Maskable Interrupts** (Cannot be delayed or rejected)
- **Vectored** the address of the service routine is fixed. **An interrupt vector is a pointer to where the ISR is stored in memory.**
- **Non-vectored** the address of the service routine needs to be supplied externally by the device.

Software Interrupts of 8085:-

The software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, if software interrupt

instruction is encountered, then the processor executes an interrupt service routine (ISR).

When the instruction is executed, the processor executes an interrupt service routine stored in the vector address of the software interrupt instruction. The 8085 has eight software interrupts from RST 0 to RST 7. The vector address for these interrupts can be calculated as follows:-

Interrupt number * 8 = vector address

For RST 5 * 8 = 40 = 28H

Vector address for interrupt RST 5 is 0028H

Interrupt	Vector address
RST 0 RST 1	0000_H 0008_H
RST 2 RST 3	0010_H 0018_H
RST 4 RST 5	0020_H 0028_H
RST 6 RST 7	0030_H 0038_H

The software interrupt instructions are included at the appropriate (or required) place in the main program. When the processor encounters the software instruction, it pushes the content of PC (Program Counter) to stack. Then loads the Vector address in PC and starts executing the Interrupt Service Routine (ISR) stored in this vector address. At the end of ISR, a return instruction - RET will be placed. When the RET instruction is executed, the processor POP the content of stack to PC. Hence the processor control returns to the main program after servicing the interrupt. **All software interrupts of 8085 are vectored interrupts. The software interrupts cannot be masked and they cannot be disabled.**

Hardware Interrupts of 8085:-

The hardware interrupts are initiated by an external device by placing an appropriate signal at the interrupt pin of the processor. If the interrupt is accepted, then the processor executes an interrupt service routine (ISR).

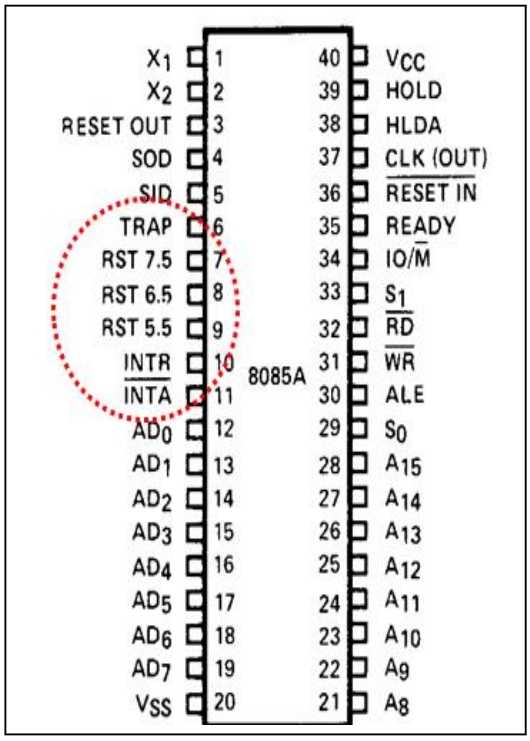
If the processor finds a valid interrupt signal and if the interrupt is unmasked and enabled, then the processor accepts the interrupt. The acceptance of the interrupt is acknowledged by sending an INTA signal to the interrupted device. The processor saves the content of PC (program Counter) in stack and then loads the vector address of the interrupt in PC. (If the interrupt is non-vectored, then the interrupting device has to supply the address of ISR when it receives INTA signal). It starts executing ISR in this address. At the end of ISR, a return instruction, RET will be placed. When the processor executes the RET instruction, it POP the content of top of stack to PC. Thus the processor control returns to main program after servicing interrupt.

The 8085 has five hardware interrupts

(1) TRAP (2) RST 7.5 (3) RST 6.5 (4) RST 5.5 (5) INTR

Interrupt	Vector address
RST 7.5	$003C_H$
RST 6.5	0034_H
RST 5.5	$002C_H$
TRAP	0024_H

Interrupt Name	Maskable	Vectored
INTR	Yes	No
RST 5.5	Yes	Yes
RST 6.5	Yes	Yes
RST 7.5	Yes	Yes
TRAP	No	Yes



Interrupt Name	Calculation	Vector Address
INTR	--	--
TRAP (RST 4.5)	4.5x8=36	0024H
RST 5.5	5.5x8=44	002CH
RST 6.5	6.5x8=52	0034H
RST 7.5	7.5x8=60	003CH

8085 Interrupts Summary:-

Interrupt Name	Triggering Method	Priority	Maskable	Masking Method	Vector Address
TRAP RST 4.5	Edge & Level Sensitive	1st Highest	No	None	0024H
RST 7.5	Edge Sensitive	2nd	Yes	DI / EI SIM	003CH
RST 6.5	Level Sensitive	3rd	Yes	DI / EI SIM	0034H
RST 5.5	Level Sensitive	4th	Yes	DI / EI SIM	002CH
INTR	Level Sensitive	5th Lowest	Yes	Pin (INTR & INTA)	--

TRAP:-

- This interrupt is a non-maskable interrupt. It is unaffected by any mask or interrupt enable.
- TRAP bas the highest priority and vectored interrupt.
- TRAP interrupt is edge and level triggered.
- The signal, which overrides the TRAP, is HOLD signal. (i.e., If the processor receives HOLD and TRAP at the same time then HOLD is recognized first and then TRAP is recognized).

There are two ways to clear TRAP interrupt.

1. By resetting microprocessor (External signal)
2. By giving a high TRAP ACKNOWLEDGE (Internal signal)

RTS 7.5:-

The RST 7.5 interrupt is a maskable interrupt. It has the second highest priority. It is edge sensitive. Maskable interrupt. It is disabled by,

1. DI instruction
2. System or processor reset.

RTS 6.5, 5.5:-

The RST 6.5 and RST 5.5 both are level triggered. Maskable interrupt. The RST 6.5 has the third priority whereas RST 5.5 has the fourth priority. It is disabled by,

1. DI, SIM instruction
2. System or processor reset.

INTR:-

INTR is a maskable interrupt. It is disabled by,

1. DI, SIM instruction
2. System or processor reset.

Non- vectored interrupt. After receiving INTA (active low) signal, it has to supply the address of ISR. It has lowest priority. It is a level sensitive interrupts.

The following sequence of events occurs when INTR signal goes high:-

- The 8085 checks the status of INTR signal during execution of each instruction.
- If INTR signal is high, then 8085 complete its current instruction and sends active low interrupt acknowledge signal, if the interrupt is enabled.
- On receiving the instruction, the 8085 save the address of next instruction on stack and execute received instruction.

Vectored / Non-Vectored and Maskable / Non-Maskable Interrupts:-

In **vectored interrupts**, the processor automatically branches to the specific address in response to an interrupt. But in **non-vectored interrupts** the interrupted device should give the address of the interrupt service routine (ISR). In vectored interrupts, the manufacturer fixes the address of the ISR to which the program control is to be transferred.

The hardware interrupts are classified into **maskable** and **non-maskable** interrupts.

• TRAP is non-maskable interrupt

• INTR, RST 7.5, RST 6.5 and RST 5.5 are maskable interrupt.

Masking is preventing the interrupt from disturbing the main program. When an interrupt is masked the processor will not accept the interrupt signal. The interrupts can be masked by moving an appropriate data (or code) to accumulator and then executing SIM instruction. (SIM - Set Interrupt Mask). The status of maskable interrupts can be read into accumulator by executing RIM instruction (RIM - Read

Interrupt Mask). All the hardware interrupts, except TRAP are disabled, when the processor is reset. They can also be disabled by executing Dl instruction. (Dl-Disable Interrupt).

When an interrupt is disabled, it will not be accepted by the processor. (i.e., INTR, RST 5.5, RST 6.5 and RST 7.5 are disabled by DI instruction and upon hardware reset). To enable (to allow) the disabled interrupt, the processor has to execute El instruction (El-Enable Interrupt).

Summary:-

An interrupt is considered to be an emergency signal that may be serviced.

- **The Microprocessor may respond to it as soon as possible.**

What happens when MP is interrupted?

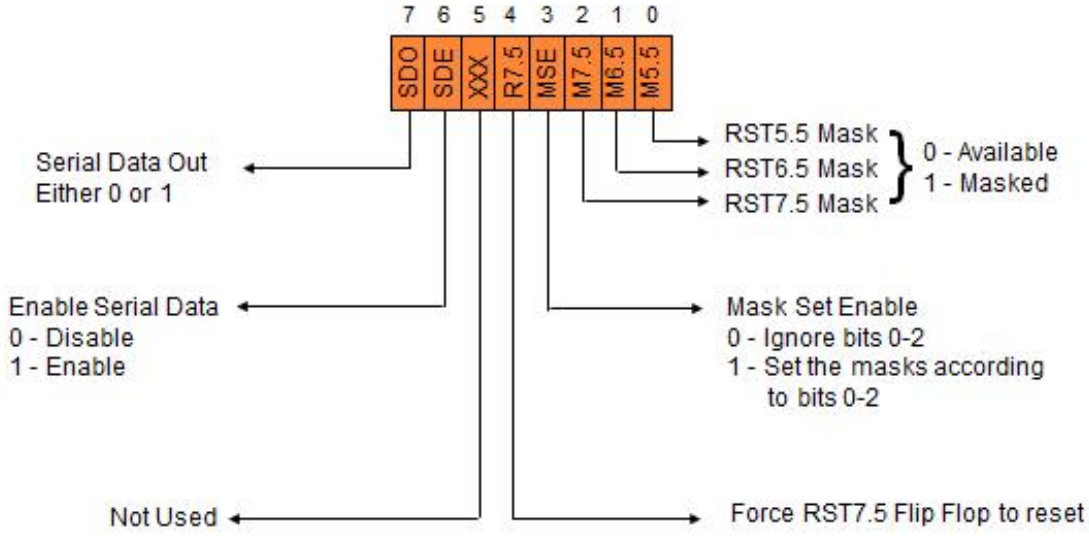
- When the Microprocessor receives an interrupt signal, it suspends the currently executing program and jumps to an Interrupt Service Routine (ISR) to respond to the incoming interrupt.
  - Each interrupt will most probably have its own ISR.
- Responding to an interrupt may be immediate or delayed depending on whether the interrupt is maskable or non-maskable.
- There are two ways of redirecting the execution to the ISR depending on whether the interrupt is vectored or non-vectored.
  - Vectored: The address of the subroutine is already known to the Microprocessor
  - Non Vectored: The device will have to supply the address of the subroutine to the Microprocessor
- When a device interrupts, it actually wants the MP to give a service which is equivalent to asking the MP to call a subroutine. This subroutine is called ISR (Interrupt Service Routine)
- The 'EI' instruction is a one byte instruction and is used to **Enable** the non-maskable interrupts.
- The 'DI' instruction is a one byte instruction and is used to **Disable** the non-maskable interrupts.
- The 8085 has a single Non-Maskable interrupt.

- The non-maskable interrupt is not affected by the value of the Interrupt Enable flip flop.
- The 8085 has 5 interrupt inputs.
  - The INTR input is the only non-vectored interrupt.
  - INTR is maskable using the EI/DI instruction pair.
  - RST 5.5, RST 6.5, RST 7.5 are all automatically vectored.
  - RST 5.5, RST 6.5, and RST 7.5 are all maskable.
  - TRAP is the only non-maskable interrupt in the 8085
  - TRAP is also automatically vectored

What are the Process steps of 8085 Maskable/Vectored Interrupt?

1. The interrupt process should be enabled using the EI instruction.
2. The 8085 checks for an interrupt during the execution of every instruction.
3. If there is an interrupt, and if the interrupt is enabled using the interrupt mask, the microprocessor will complete the executing instruction, and reset the interrupt flip flop.
4. The microprocessor then executes a call instruction that sends the execution to the appropriate location in the interrupt vector table.
5. When the microprocessor executes the call instruction, it saves the address of the next instruction on the stack.
6. The microprocessor jumps to the specific service routine.
7. The service routine must include the instruction EI to re-enable the interrupt process.
8. At the end of the service routine, the RET instruction returns the execution to where the program was interrupted.

SIM – SERIAL INTERRUPT MASK



- Example

  MSE Mask Set Enable

  RST 6.5 Mask

  RST 5.5 & 7.2 Unmask

  RST FF Don't Reset

  Serial Data Igonered

SDO	SDE	XXX	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	1	0	1	0

Contents of accumulator are: 0AH

```
EI          ; Enable interrupts including INTR
MVI A, 0A   ; Prepare the mask to enable RST 7.5, and 5.5, disable 6.5
SIM         ; Apply the settings RST masks
```

- Example

  MSE Mask Set Disable

  RST FF Reset

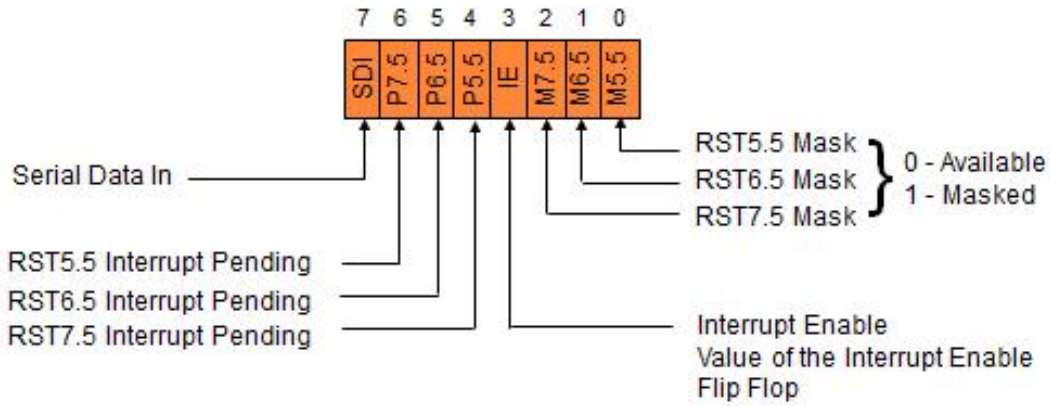
  Serial Data Enable

  Serial Data output is 0

SDO	SDE	XXX	R7.5	MSE	M7.5	M6.5	M5.5
0	1	0	1	0	1	0	0

Contents of accumulator are: 54H

RIM – READ INTERRUPT MASK



Set – 1
Reset - 0

- Example

  Interrupt Enable

  RST 5.5 & 6.5 Masked

  RST 7.5 Pending

  Serial Input Data is 0

SID	P7.5	P6.5	P5.5	IE	M7.5	M6.5	M5.5
0	1	0	0	1	0	1	1

Contents of accumulator are: 4BH

Summary:
1- Without interrupts, CPU operation will be impossible to replicate
2- Interrupts can be triggered in H.W or S.W.

Questions:
1- What are mask-able vs non-mask-able interrupts ?
2- Enumerate and explain Interrupt types.