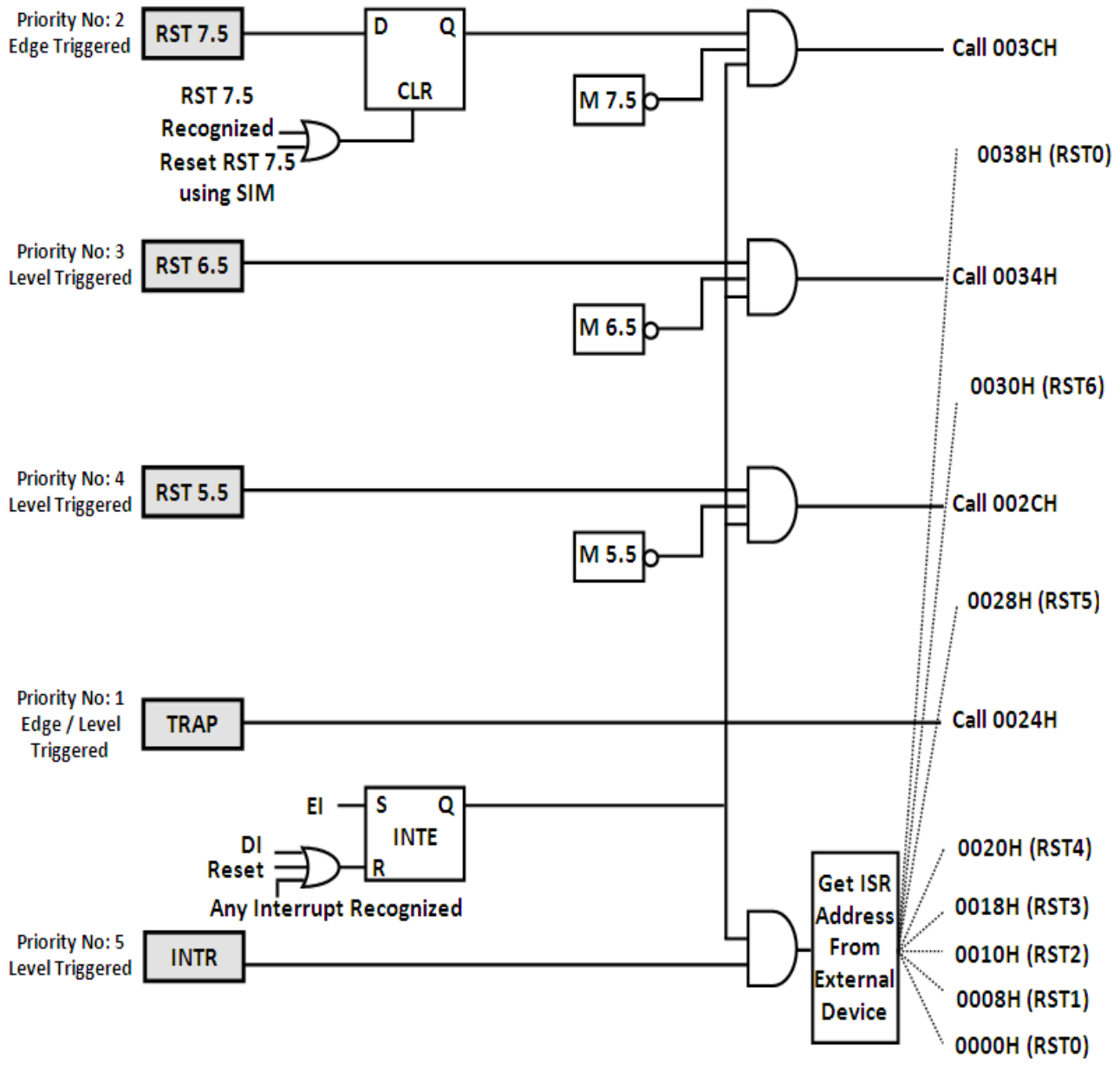
8085 INTERRUPTS

- An interrupt is a **special condition** that makes the µP execute an ISR.
- **µP services** the interrupt **by executing** a subroutine called as the **Interrupt Service Routine**.
- The **µP checks** for interrupts **during every instruction**.
- When an **interrupt occurs**, the **µP first finishes the current instruction**.
- It then **Pushes** the address of the next instruction **(contents of PC) on the STACK**.
- It **resets** the **INTE flip-flop** so that **no more interrupts are recognized**.
- Thereafter the program control transfers to the **address of** the Interrupt Service Routine (**ISR**) and the **µP** thus **executes the ISR**.

	SOFTWARE INTERRUPTS	HARDWARE INTERRUPTS
1	They are caused by writing an instruction.	They occur as signals on external pins.
2	8085 has 8 software interrupt instructions called RST N, where N can be any value from 0…7. Hence we have RST 0 … RST 7.	8085 has 5 hardware interrupt pins: TRAP RST 7.5 RST 6.5 RST 5.5 INTR
3	Software interrupts cannot be masked or disabled.	All Hardware interrupts can be disabled except TRAP.
4	All Software interrupts have the same priority.	Hardware interrupts have the following priority order: TRAP … 1st (Highest) RST 7.5 … 2nd RST 6.5 … 3rd RST 5.5 … 4th INTR … 5th(Lowest)
5	All Software interrupts are Vectored	All Hardware interrupts are Vectored except INTR.
6	The Vector addresses are as follows: RST 0 … 0000 H RST 1 … 0008 H RST 2 … 0010 H RST 3 … 0018 H RST 4 … 0020 H RST 5 … 0028 H RST 6 … 0030 H RST 7 … 0038 H	The Vector addresses are as follows: TRAP … 0024 H RST 7.5 … 003C H RST 6.5 … 0034 H RST 5.5 … 002C H INTR … obtain ISR address from device

INTERRUPT STRUCTURE OF 8085



Software Interrupts:
Interrupts that are **initiated by writing an Instruction** (software) are called as software interrupts.
8085 has 8 software interrupts:
RSTn where n = 0,1,2, … 7 i.e. RST0, RST1 … upto RST7.
– This instruction causes a **service routine** to be Called from the **address (n*8)**.
– Hence, if RST1 occurs then the program control moves to location 0008 (1*8 = 0008).
The respective addresses for software interrupts are given below.

S/W Interrupt	ISR Address
RST0	00**0**0H
RST1	00**0**8H
RST2	00**1**0H
RST3	00**1**8H
RST4	00**2**0H
RST5	00**2**8H
RST6	00**3**0H
RST7	00**3**8H

Hardware Interrupts:
Interrupts that are initiated through a hardware pin are called as hardware interrupts.
8085 supports the following hardware interrupts:
- TRAP
- RST 7.5
- RST 6.5
- RST 5.5
- INTR

Vectored Interrupts:
- Interrupts that **have** a **FIXED Address** for their ISR are called as Vectored Interrupts.
- **Eg: TRAP** is a vectored interrupt. Its vector address is 0024H.

Non-Vectored Interrupts:
- Interrupts that **have** a **Variable Address** for their ISR are called as Non-Vectored Interrupts.
- **Eg: INTR** is a Non-Vectored Interrupt.

Methods of preventing an interrupt from occurring.
- **MASK Individual Bits** through **SIM** Instruction
- **Disable all** Interrupts through **DI** Instruction

MASKING:
- We can prevent an interrupt from occurring by MASKING its individual bit through **SIM Instruction**.
- If an interrupt is masked it will not be serviced.
- One of the **main advantages** of masking as opposed to disabling interrupts is that by masking we can **selectively disable a particular interrupt** while keeping other interrupts active**,** whereas through DI instruction all interrupts are disabled.
- ONLY **RST 7.5, RST 6.5 and RST 5.5** can be masked by this method.

DISABLING INTERRUPTS:

- Interrupts can be disabled through the **DI** Instruction.
- This instruction resets the INTE Flip Flop and hence none of the interrupts can occur (**Except TRAP**).
- I.e. INTE F/F ← 0.
- Once disabled, these interrupts can be **re-enabled** through **EI** instruction, which sets the INTE Flip Flop.
- I.e. INTE F/F ← 1.

Hardware Interrupts (In detail)

TRAP

- TRAP has the **highest priority.**
- It is **Edge as well as Level triggered** hence the signal must go High and also Remain high for some time for it to be recognized. This prevents any noise signal from being accepted.
- It is a Non-Maskable Interrupt i.e. it can **neither be masked nor be disabled.**
- It is a vectored interrupt and has a **vector address of 0024H.**

RST 7.5

- RST 7.5 has the **priority lower than TRAP.**
- It is **Edge triggered.**
- It is a **Maskable Interrupt** i.e. it can be masked through the **SIM Instruction**.
- It can also be disabled though the **DI Instruction**.
- It is a vectored interrupt and has a **vector address of 003CH.**
- RST 7.5 can also be **reset** through the **R 7.5** bit in the **SIM** Instruction irrespective of whether it is Masked or not.

RST 6.5

- RST 6.5 has the **priority lower than RST 7.5.**
- It is **Level triggered.**
- It is a **Maskable Interrupt** i.e. it can be masked through the **SIM Instruction**.
- It can also be disabled though the **DI Instruction**.
- It is a vectored interrupt and has a **vector address of 0034H.**

RST 5.5

- RST 5.5 has the **priority lower than RST 6.5.**
- It is **Level triggered.**
- It is a **Maskable Interrupt** i.e. it can be masked through the **SIM Instruction**.
- It can also be disabled though the **DI Instruction**.
- It is a vectored interrupt and has a **vector address of 002CH.**

INTR

- INTR has the **priority lower than RST 5.5.**
- It is **Level triggered.**
- It can only be disabled though the **DI Instruction**.
- **It cannot be masked through the SIM Instruction**.
- It is a **Non-Vectored** interrupt.

Response to INTR:

- When INTR occurs the µP, in response, issues the **first INTA** cycle.
- The **External Hardware sends** an **opcode**, which can be of **RSTn** Instruction **or** of **CALL** instruction.

a) **If** opcode of **RSTn** is sent by the external hardware
   - The µP calculates the address of the ISR as **n*8.**

b) **If** opcode of **CALL** is sent:
   - As Call is a **3-Byte Instruction** the µP send **2 more INTA** signals
   - In response to the **2nd and the 3rd INTA** cycle the external hardware returns the **lower and the higher byte of the address** of the ISR respectively.

This is how the address is determined when INTR occurs.

INTA : (Interrupt Acknowledge)

- This is an **active low acknowledge** signal going out of 8085.
- This signal is **given in response to** an interrupt on **INTR ONLY**.
- After the **first INTA** is given, the interrupting **peripheral sends an opcode**.
- **If** the **Opcode** is **of RSTn**, then the **ISR address is calculated as n × 8.**
- **If** the **Opcode** is **of CALL**, then **two more INTA signals** are given and, the lower byte, and then the higher byte of the ISR address are sent by the peripheral.
- *#Please refer Bharat Sir's video at www.BharatAcharyaEducation.com for understanding this..*

EI and DI Instructions

INTE F/F : (Interrupt Enable Flip Flop)

- This flip-flop decides if interrupts are enabled in the µP i.e. **if it is set, all interrupts are enabled**.
- It is **set by** the **EI Instruction**.
- It is **reset** in the following 3 ways:

i. **If µP is reset**.
- If **DI instruction** is executed.
- If **any other interrupt is recognized** by the µP. In this case the INTE F/F is later set in the ISR by EI.

☺ In case of doubts, contact Bharat Sir: - 98204 08217.

The INTE F/F affects all interrupts **EXCEPT TRAP**, as it cannot be disabled.

Note

On reset by default interrupts are disabled.
If we don't write EI in our program hardware interrupts will not be serviced except TRAP.

Interrupt	Priority	Triggering	Maskable by SIM	Disabled by DI	Vectored	Vector Address
TRAP	**1**	**Edge / Level**	**No**	**No**	Yes	0024 H
RST 7.5	2	**Edge**	Yes	Yes	Yes	003C H
RST 6.5	3	Level	Yes	Yes	Yes	0034 H
RST 5.5	4	Level	Yes	Yes	Yes	002C H
INTR	5	Level	**No**	Yes	**No**	**Get ISR Address from External Hardware**

Special Note:

If you are learning this by piracy, then you are not my student. You are simply a thief! #PoorUpbringing