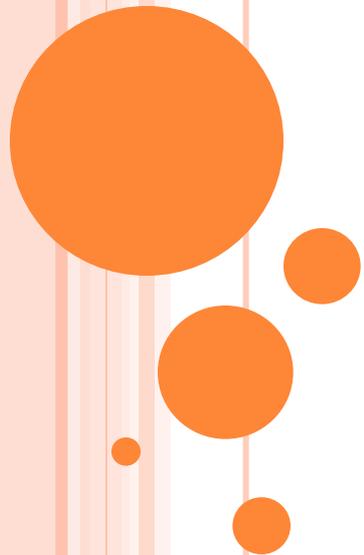


# **INTERRUPTS (CONTINUED) LECTURE 6**



# The 8085 Maskable/Vectored Interrupts

- The 8085 has 4 Masked/Vectored interrupt inputs.
  - RST 5.5, RST 6.5, RST 7.5
    - They are all **maskable**.
    - They are **automatically vectored** according to the following table:

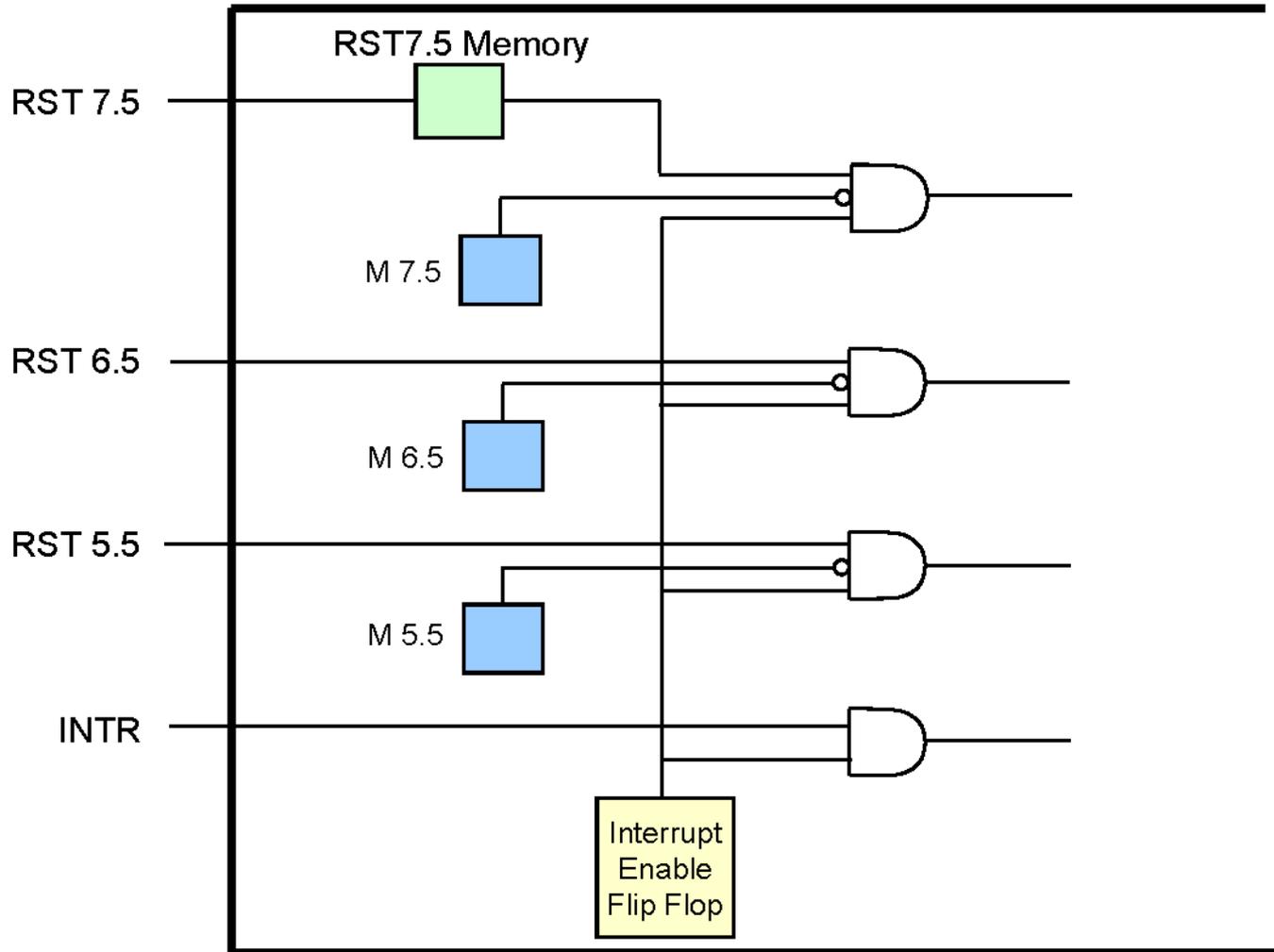
Interrupt	Vector
RST 5.5	002CH
RST 6.5	0034H
RST 7.5	003CH

- The vectors for these interrupt fall in between the vectors for the RST instructions. That's why they have names like RST 5.5 (RST 5 and a half).

# Masking RST 5.5, RST 6.5 and RST 7.5

- These three interrupts are masked at two levels:
  - Through the Interrupt Enable flip flop and the EI/DI instructions.
    - The Interrupt Enable flip flop controls the whole maskable interrupt process.
  - Through individual mask flip flops that control the availability of the individual interrupts.
    - These flip flops control the interrupts individually.

# Maskable Interrupts



# The 8085 Maskable/Vectored Interrupt Process

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.

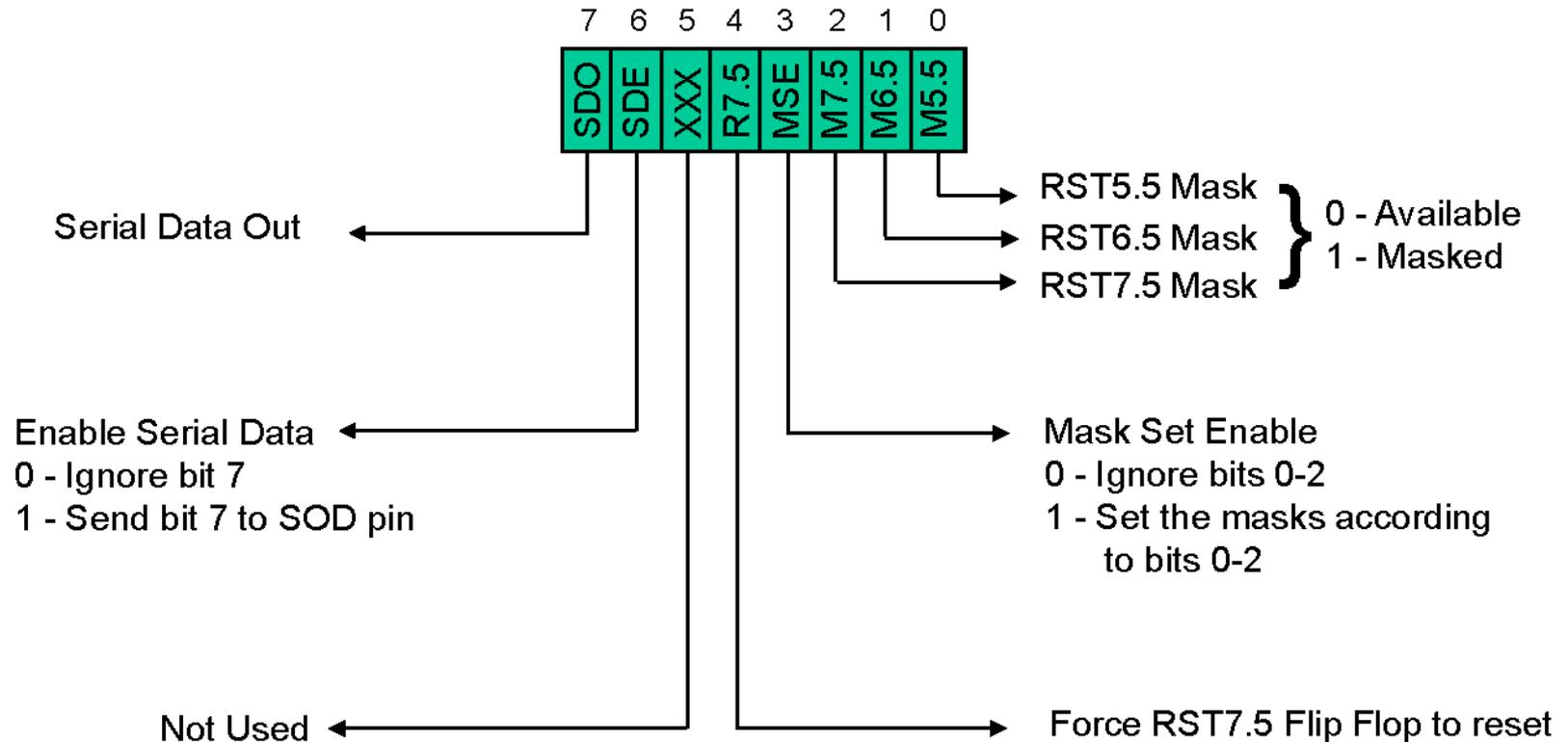
# The 8085 Maskable/Vectored Interrupt Process

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**.

# Manipulating the Masks

- The Interrupt Enable flip flop is manipulated using the EI/DI instructions.
- The individual **masks** for RST 5.5, RST 6.5 and RST 7.5 are manipulated using the **SIM** instruction.
  - This instruction takes the bit pattern in the Accumulator and applies it to the interrupt mask enabling and disabling the specific interrupts.

# How SIM Interprets the Accumulator



# SIM and the Interrupt Mask

- Bit 0 is the **mask** for RST 5.5, bit 1 is the **mask** for RST 6.5 and bit 2 is the **mask** for RST 7.5.
  - If the mask bit is 0, the interrupt is **available**.
  - If the mask bit is 1, the interrupt is **masked**.
- Bit 3 (Mask Set Enable - MSE) is an **enable for setting the mask**.
  - If it is set to 0 the mask is **ignored** and the old settings remain.
  - If it is set to 1, the new settings are **applied**.
  - The SIM instruction is used for multiple purposes and not only for setting interrupt masks.
    - It is also used to control functionality such as **Serial Data Transmission**.
    - Therefore, bit 3 is necessary to tell the microprocessor whether or not the interrupt masks should be modified

# SIM and the Interrupt Mask

- The RST 7.5 interrupt is the **only** 8085 interrupt that has **memory**.
  - If a signal on RST7.5 arrives while it is masked, a flip flop will remember the signal.
  - When RST7.5 is unmasked, the microprocessor will be interrupted **even if the device has removed the interrupt signal**.
  - This flip flop will be **automatically reset** when the microprocessor **responds to an RST 7.5 interrupt**.
- Bit 4 of the accumulator in the SIM instruction allows **explicitly resetting** the RST 7.5 memory even if the microprocessor did not respond to it.

# SIM and the Interrupt Mask

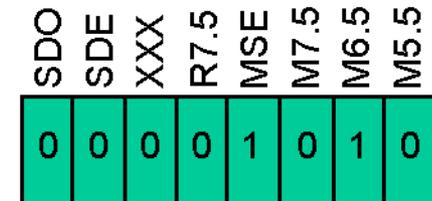
- The SIM instruction can also be used to perform serial data transmission out of the 8085's SOD pin.
  - One bit at a time can be sent out serially over the SOD pin.
- Bit 6 is used to tell the microprocessor whether or not to perform serial data transmission
  - If 0, then do not perform serial data transmission
  - If 1, then do.
- The value to be sent out on SOD has to be placed in bit 7 of the accumulator.
- Bit 5 is not used by the SIM instruction

# Using the SIM Instruction to Modify the Interrupt Masks

- Example: Set the interrupt masks so that RST5.5 is enabled, RST6.5 is masked, and RST7.5 is enabled.

– First, determine the contents of the accumulator

- |                             |           |
|-----------------------------|-----------|
| - Enable 5.5                | bit 0 = 0 |
| - Disable 6.5               | bit 1 = 1 |
| - Enable 7.5                | bit 2 = 0 |
| - Allow setting the masks   | bit 3 = 1 |
| - Don't reset the flip flop | bit 4 = 0 |
| - Bit 5 is not used         | bit 5 = 0 |
| - Don't use serial data     | bit 6 = 0 |
| - Serial data is ignored    | bit 7 = 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

# Triggering Levels

- RST 7.5 is **positive edge sensitive**.
  - When a positive edge appears on the RST7.5 line, a logic 1 is **stored** in the flip-flop as a “**pending**” interrupt.
  - Since the value has been stored in the flip flop, the line **does not have to be high** when the microprocessor checks for the interrupt to be recognized.
  - The line must **go to zero and back to one** before a new interrupt is recognized.
- RST 6.5 and RST 5.5 are **level sensitive**.
  - The interrupting signal **must remain present until the microprocessor checks for interrupts**.