1

SIM.

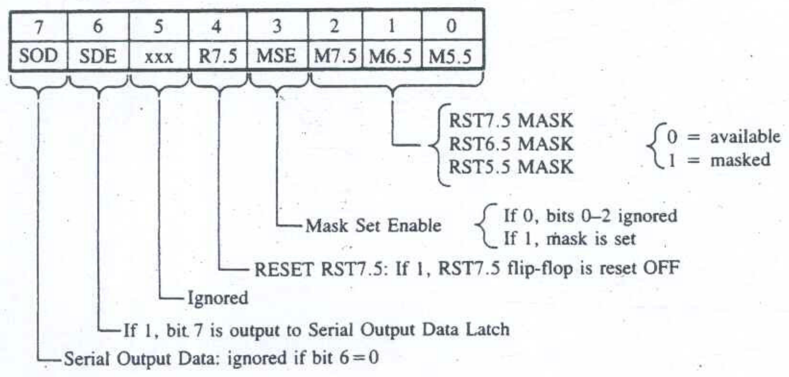

7 6 5 4 3 2 1 0
SOD SDE xxx R7.5 MSE M7.5 M6.5 M5.5
RST7.5 MASK
RST6.5 MASK
RST5.5 MASK
0 = available
1 = masked
Mask Set Enable
If 0, bits 0–2 ignored
If 1, mask is set
RESET RST7.5: If 1, RST7.5 flip-flop is reset OFF
Ignored
If 1, bit 7 is output to Serial Output Data Latch
Serial Output Data: ignored if bit 6 = 0

FIGURE 12.6
Interpretation of the Accumulator Bit Pattern for the SIM Instruction
SOURCE: Intel Corporation, *Assembly Language Programming Manual* (Santa Clara, Calif.: Author, 1979), pp. 3–59.

Instruction SIM: Set Interrupt Mask. This is a 1-byte instruction and can be used for three different functions (Figure 12.6).

- ☑ One function is to set mask for RST 7.5, 6.5, and 5.5 interrupts. This instruction reads the content of the accumulator and enables or disables the interrupts according to the content of the accumulator. Bit D_3 is a control bit and should = 1 for bits D_0, D_1, and D_2 to be effective. Logic 0 on D_0, D_1, and D_2 will enable the corresponding interrupts, and logic 1 will disable the interrupts.
- ☐ The second function is to reset RST 7.5 flip-flop (Figure 12.6). Bit D_4 is additional control for RST 7.5. If $D_4 = 1$, RST 7.5 is reset. This is used to override (or ignore) RST 7.5 without servicing it.
- ☐ The third function is to implement serial I/O (discussed in Chapter 16). Bits D_7 and D_6 of the accumulator are used for serial I/O and do not affect the interrupts. Bit $D_6 = 1$ enables the serial I/O and bit D_7 is used to transmit (output) bits.

Here we are concerned with RST 7.5, 6.5, and 5.5 interrupts and not with serial I/O.

The mnemonic SIM is confusing. The wording—Set Interrupt Mask—implies that the instruction masks the interrupts. However, the instruction must be executed in order to use the interrupts. The process required to enable these interrupts can be likened to a switchboard controlling three telephone extensions in a company. Let us assume these phone extensions are assigned to the president (RST 7.5), the vice president (RST 6.5), and the manager (RST 5.5), in that priority, and are monitored by their receptionist according to the instructions given. The protocols of placing a telephone call to one of the executives and of interrupting the microprocessor using RST 7.5, 6.5, and 5.5 can be compared as follows:

Placing a Telephone Call	**Interrupting the 8085 (Figure 12.6)**
1. The switchboard is functional and all telephone lines are open.	1. The interrupt process is enabled. The instruction EI sets the Interrupt Enable flip-flop, and one of the inputs to the AND gates is set to logic 1 (Figure 12.5). These AND gates activate the program transfer to various vectored locations.
2. All executives leave instructions on the receptionist's desk as to whether they wish to receive any phone calls.	2. An appropriate bit pattern is loaded into the accumulator.
3. The receptionist reads the instructions.	3. If bit $D_3 = 1$, the respective interrupts are enabled according to bits D_2–D_0.
4. The receptionist is on duty and sends calls through for whoever is available.	4. RST 7.5, 6.5, and 5.5 are being monitored.
5. The receptionist is busy typing. Phone calls can be received directly according to previous instructions.	5. If bit $D_3 = 0$, bits D_2–D_0 have no effect on previous conditions.
6. No calls for the president now. Call back later.	6. Bit $D_4 = 1$; this resets RST 7.5.

This analogy can be extended to the interrupt INTR, which is viewed as one telephone line shared by eight engineers with a switchboard operator (external hardware) who rings the appropriate extension.

The entire interrupt process (except TRAP) is disabled by resetting the Interrupt Enable flip-flop (Figure 12.5). The flip-flop can be reset in one of the three ways:

- ☐ Instruction DI
- ☐ System Reset
- ☐ Recognition of an Interrupt Request

Figure 12.5 shows that these three signals are ORed and the output of the OR gate is used to reset the flip-flop.

TRIGGERING LEVELS

These interrupts are sensitive to different types of triggering as listed below:

- ☐ **RST 7.5** This is positive-edge sensitive and can be triggered with a short pulse. The request is stored internally by the D flip-flop (Figure 12.5) until the microprocessor responds to the request or until it is cleared by Reset or by bit D_4 in the SIM instruction.
- ☐ **RST 6.5** and **RST 5.5** These interrupts are level-sensitive, meaning that the triggering level should be on until the microprocessor completes the execution of the current instruction. If the microprocessor is unable to respond to these requests immediately, they should be stored or held by external hardware.

Example 12.1

Enable all the interrupts in an 8085 system.

Instructions

```
EI          ;Enable interrupts
MVI A,08H   ;Load bit pattern to enable RST 7.5, 6.5, and 5.5
SIM         ;Enable RST 7.5, 6.5, and 5.5
```

Bit $D_3 = 1$ in the accumulator makes the instruction SIM functional, and bits D_2, D_1, and $D_0 = 0$ enable the interrupts 7.5, 6.5, and 5.5.

Example 12.2

Reset the 7.5 interrupt from Example 12.1.

Instructions

```
MVI A,18H   ;Set D4 = 1
SIM         ;Reset 7.5 interrupt flip-flop
```

PENDING INTERRUPTS

Because there are several interrupt lines, when one interrupt request is being served, other interrupt requests may occur and remain pending. The 8085 has an additional instruction called RIM (Read Interrupt Mask) to sense these pending interrupts.

Instruction RIM: Read Interrupt Mask. This is a 1-byte instruction that can be used for the following functions.

- ☐ To read interrupt masks. This instruction loads the accumulator with 8 bits indicating the current status of the interrupt masks (Figure 12.7).
- ☐ To identify pending interrupts. Bits D_4, D_5, and D_6 (Figure 12.7) identify the pending interrupts.
- ☐ To receive serial data. Bit D_7 (Figure 12.7) is used to receive serial data.

As discussed in last lecture ~~[illegible]~~ ↓

#

Example 12.3

Assuming the microprocessor is completing an RST 7.5 interrupt request, check to see if RST 6.5 is pending. If it is pending, enable RST 6.5 without affecting any other interrupts; otherwise, return to the main program.

Instructions

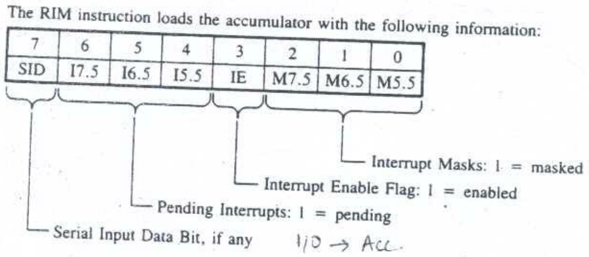
```
RIM         ;Read interrupt mask
MOV B,A     ;Save mask information
ANI 20H     ;Check whether RST 6.5 is pending
```

4

```
            JNZ NEXT
            EI
            RET           ;RST 6.5 is not pending, return to main program
NEXT:       MOV A,B       ;Get bit pattern; RST 6.5 is pending
            ANI 0DH       ;Enables RST 6.5 by setting D1 = 0
            ORI 08H       ;Enable SIM by setting D3 = 1
            SIM
            JMP SERV      ;Jump to service routine for RST 6.5
```

The instruction RIM checks for a pending interrupt. Instruction ANI 20H masks all the bits except D_5 to check pending RST 6.5. If $D_5 = 0$, the program control is transferred to the main program. $D_5 = 1$ indicates that RST 6.5 is pending. Instruction ANI 0DH sets $D_1 = 0$ (RST 6.5 bit for SIM), instruction ORI sets $D_3 = 1$ (this is necessary for SIM to be effective), and instruction SIM enables RST 6.5 without affecting any other interrupts. The JMP instruction transfers the program to the service routine (SERV) written for RST 6.5.

The RIM instruction loads the accumulator with the following information:



7
6
5
4
3
2
1
0
SID
I7.5
I6.5
I5.5
IE
M7.5
M6.5
M5.5
Interrupt Masks: 1 = masked
Interrupt Enable Flag: 1 = enabled
Pending Interrupts: 1 = pending
Serial Input Data Bit, if any

I/O → Acc.

FIGURE 12.7
Interpretation of the Accumulator Bit Pattern for the RIM Instruction
SOURCE: Intel Corporation, *Assembly Language Programming Manual* (Santa Clara, Calif.: Author, 1979), pp. 3–49.

~~12.2.3 Illustration: Interrupt-Driven Clock~~

~~PROBLEM STATEMENT~~

~~Design a 1-minute timer using a 60 Hz power line as an interrupting source. The output ports should display minutes and seconds in BCD. At the end of the minute, the output ports should continue displaying one minute and zero seconds.~~

~~HARDWARE DESCRIPTION~~

~~This 1-minute timer is designed with a 60 Hz AC line. The circuit (Figure 12.8) uses a step-down transformer, the 74121 monostable multivibrator, and interrupt pin RST 6.5. After the interrupt, program control is transferred to memory location 0034H in the monitor program.~~

~~The AC line with 60 Hz frequency has a period of 16.6 ms; that means it can provide a pulse every sixtieth of a second with 8.3 ms pulse width, which is too long for the interrupt. The interrupt flip-flop is enabled again within 6 μs in the timer service routine;~~