Examples of Assembly Language Programs

6.1 INTRODUCTION

To learn assembly language programming the beginner should write simple programs given in this chapter and try to execute them on Intel 8085 based microprocessor kit. The memory addresses given in the program are for a particular microprocessor kit. These addresses can be changed to suit the microprocessor kit available in the laboratory. Before writing assembly language program, one should learn some important Intel 8085 instructions such as MOV, MVI, ADD, SUB, LXI, LDA, INX, INR, HLT etc. described in chapter 4. While learning programs, one should gradually pick up new instructions which have not been used earlier.

6.2 SIMPLE EXAMPLES

The use of some important instructions are described below with very simple examples.

The memory addresses given in these examples are for Vinytics' kit.

Example 1. Object: Place 05 in register B.

PROGRAM

Memory address	*Machine codes*	*Mnemonics*	*Operands*	*Comments*
FC00	06,05	MVI	B, 05	Get 05 in register B.
FC02	76	HLT		Stop

The instruction MVI B, 05 moves 05 to register B. HLT halts the program. A program is fed to the microprocessor kit in machine codes. The machine code for the instruction MVI B, 05 is 06, 05. The 1st byte of the instruction is 06 which is the machine code for the instruction MVI B. The second byte of the instruction, 05 is the data which is to be moved to register B. The code for HLT is 76. The machine codes for a program are entered in the memory. In the above program the memory addresses from FC00 to FC02 have been used. The machine code 06 is entered in the memory location FC00 H; 05 in FC01 H and 76 in FC02 H. This program can be executed on a microprocessor kit and the register B can be examined. After the execution of the program the register B will contain 05. The memory address can be changed to suit the microprocessor kit available in the laboratory. The address and data used for a microprocessor based system are in hexadecimal system. The symbol H after a digit denotes that it is in hexadecimal system.

Example 2

Object: Get 05 in register A; then move it to register B.

PROGRAM

Memory address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
FC00	3E, 05	MVI	A, 05	Get 05 in register A.
FC02	47	MOV	B,A	Transfer 05 from register A to B.
FC03	76	HLT		Stop.

Note: Microprocessors are designed to process hexadecimal numbers. Hence, data and address given in an assembly language program are hexadecimal numbers.

The instruction MVI A, 05 will move 05 to register A. In the code from it is written as 3E, 05. The 1st byte of the instruction is 3E. This code is for MVI A. The second byte 05 is the data which is to be placed in A. The instruction MOV B, A transfers the content of register A to register B. After the execution of the above program regiser B will contain 05. The program can be executed on a microprocessor kit and the register B can be examined.

Example 3

Object: Load the content of the memory location FC50 H directly to the accumulator, then transfer it to regiser B. The content of the memory location FC50 H is 05.

PROGRAM

Memory address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
FC00	3A, 50, FC	LDA	FC50	Get the content of the memory location FC50 H into accumulator.
FC03	47	MOV	B,A	Move the content of register A to B.
FC04	76	HLT	Halt.	

DATA

FC50 — 05

The instruction LDA loads the accumulator directly with the content of the memory location specified in the instruction. Thus the instruction LDA FC50 H will load the accumulator with the content of the memory location FC50 H. The instruction MOV B, A will move the content of the accumulator to B. The content of the memory location FC50 H is 05. It is fed to the microprocessor kit as data. After the execution of the above program the register B will contain 05.

Example 4

Object: Move the content of the memory location FC50 H to register C. The content of the memory location FC50 H is 08.

PROGRAM

Memory address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
FC00	21, 50, FC	LXI	H,FC50	Get the memory address FC50 H in H-L pair.
FC03	4E	MOV	C,M	Move the content of the memory location, whose address is in the H-L pair, to register C.
FC04	76	HLT		Halt.

DATA

FC50 — 08

The instruction LXI H, FC50 H will place FC50 H in register pair H-L. FC50 H is the address of the memory location from where the data is to be transferred to register C. In the code form the instruction is written as 21, 50, FC. The 1st byte of the instruction is 21. It is the machine code for the instruction LXI H. The operand is FC50 which is to be placed in H-L pair. The 2nd byte of the instruction is 50. It is 8 LSBs of the operand. The 3rd byte is FC which is 8 MSBs of the operand. In the code form the LSB is written first then the MSB. Due to this reason the operand FC50 has been written in the code form as 50, FC. The instruction MOV C, M transfers the content of the memory location, whose address is in H-L pair, to register C. The execution of the previous instruction has placed FC50

H in H-L pair. Therefore, MOV C, M will move the content of FC50 H to register C. After the execution of the above program the regiseter C will contain 08.

Example 5

Object : Place the content of the memory location FC50 H in register B and that of FC51 H in register C. The contents of FC50 and FC51 H are 11 H and 12 H respectively.

PROGRAM

Memory address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
FC00	21, 50, FC	LXI	H, FC50 H	Get FC50 H in H-L pair.
FC03	46	MOV	B, M	Move the content of FC50 to B.
FC04	23	INX	H	Increment H-L pair by one.
FC05	4E	MOV	C, M	Move the content of FC51 to C.
FC06	76	HLT		Halt.

DATA

FC50 — 11 H
FC51 — 12 H.

The instruction LXI H, FC50 H will place FC50 H in H-L pair. FC50 H is the address of the memory location which contains 11 H. MOV B, M will move the content of FC50 H to register B. The instruction INX H increases the content of H-L pair by 1. The execution of the instruction INX H will increase the content of H-L pair from FC50 H to FC51 H. MOV C, M will move the content of FC51 H to register C. Thus, after the execution of the above program, the register B and C will contain 11 H and 12 H respectively.

Example 6

Object : Place 05 in the accumulator. Increment it by one and store the result in the memory location FC50 H.

PROGRAM

Memory address	Machine Codes	*Mnemonics*	*Operands*	*Comments*
FC00	3E,05	MVI	A,05	Get 05 in the accumulator.
FC02	3C	INR	A	Increment the content of accumulator by one.
FC03	32,50,FC	STA	FC50 H	Store result in FC50 H.
FC06	76	HLT		Halt

The instruction MVI A, 05 moves 05 to the accumulator. INR A increases the content of the accumulator from 05 to 06. STA FC50 stores the content of the accumulator in the memory location FC50 H. After the execution of the above program the memory location FC50 H will contain 06.

A number of important and useful assembly language programs are given in the subsequent subsections. Memory addresses used for them are for Professional's kits/Vinytics' kits.

6.3. ADDITION OF TWO 8-BIT NUMBERS; SUM 8-BITS

Problem:

Add 49 H and 56 H.

The 1st number 49 H is in the memory location 2501 H.
The 2nd number 56 H is in the memory location 2502 H.
The result is to be stored in the memory location 2503 H.
Numbers are represented in hexadecimal system.

PROGRAM

Memory address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
2000	21, 01, 25	LXI	H, 2501 H	Get address of 1st number in H-L pair.
2003	7E	MOV	A,M	1st number in accumulator.
2004	23	INX	H	Increment content of H-L pair.
2005	86	ADD	M	Add 1st and 2nd numbers.
2006	32, 03,25	STA	2503 H	Store sum in 2503 H.
2009	76	HLT		Stop

DATA
2501 — 49 H
2502 — 56 H
The sum is stored in the memory location 2503 H.

Result
2503 — 9F H.

2501 H is the address of memory location for the 1st number. 2501 is placed in H-L pair by the instruction LXI H, 2501 H. The next instruction is MOV A, M which moves the content of the memory location addressed by H-L pair to the accumulator. In this case H-L pair contains 2501 H and, therefore, the content of the memory location 2501 H is moved to the accumulator. Thus the 1st number 49 H has been moved to the accumulator. The instruction INX H increases the content of H-L pair by one. Previously, the content of H-L pair was 2501 H. After the execution of INX H it becomes 2502 H. ADD M adds the contents of the accumulator and the content of the memory location addressed by H-L pair. The content of 2502 H is the 2nd number 56 H. So 56 H is added to 49 H. Sum resides in the accumulator. The instruction STA 2503 H stores the sum in the memory location 2503 H. The instruction HLT ends the program.

6.4. 8-BIT SUBTRACTION*

Example 1

	49 H	1st number
	– 32 H	2nd number
Result	17 H	

The 1st number 49 H is in the memory location 2501 H.
The 2nd number 32 H is in the memory location 2502 H.
The result is to be stored in the memory location 2503 H.

PROGRAM

Memory address	Machine Codes	*Mnemonics*	*Operands*	*Comments*
2000	21, 01, 25	LXI	H, 2501 H	Get address of 1st number in H-L pair.
2003	7E	MOV	A, M	1st number in accumulator.
2004	23	INX	H	Content of H-L pair increases from 2501 to 2502 H.
2005	96	SUB	M	1st number - 2nd number.
2006	23	INX	H	Content of H-L pair becomes 2503 H.
2007	77	MOV	M, A	Store result in 2503 H.
2008	76	HLT		Halt

Example 1

DATA

2501 — 49 H

2502 — 32 H

Result is stored in the Memory location 2503 H

2503 — 17 H

Example 2

DATA

2501 — F8 H

2502 — 9B H

Result

2503 — 5D H

The 1st number is in the memory location 2501 H. 2501 is placed in H-L pair by the execution of the instruction LXI H, 2501 H. The instruction MOV A, M moves the content of the memory location addressed by H-L pair to the accumulator. Thus the 1st number 49 H (Example 1) which is in 2501 H is placed in the accumulator. INX H increases the content of H-L pair from 2501 to 2502 H. The instruction SUB M subtracts the content of the memory location addressed by H-L pair from the accumulator. The 2nd number which is in the memory location 2502 H is subtracted from the 1st number which is in the accumulator. The result resides in the accumulator. The instruction INX H increases the content of H-L pair from 2502 to 2503 H. The instruction MOV M, A transfers the content of the accumulator to the memory location addressed by H-L pair. So the result which is in the accumulator is sotred in the memory location 2503 H. The instruction HLT ends the program.

6.9. FIND ONE'S COMPLEMENT OF AN 8-BIT NUMBER

Example 1. Find one's complement of 96 H. The number in the binary form is represented as follows :

96 H = 1001 0110
(9) (6)

One's complement = 0110 1001 = 69 H.
(6) (9)

To obtain one's complement of a number its 0 bits are replaced by 1 and 1 by 0.

The number is placed in the memory location 2501 H.

The result is stored in the memory location 2502 H.

PROGRAM

Address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
2000	3A, 01, 25	LDA	C100H 2501 H	Get data in accumulator.
2003	2F	CMA		Take its complement.
2004	32, 02, 25	STA	C102H 2502 H	Store result in 2502 H.
2007	76	HLT		Halt

Example 1

DATA

2501 — 96 H

Result

2502 — 69 H

Example 2

DATA

2501 — E4 H

Result

2502 — 1B H

The instruction LDA 2501 H transfers the number from memory location 2501 H to the accumulator. CMA takes complement of the number. STA 2502 H stores the result in the memory location 2502 H. HLT ends the program.

6.10. FIND ONE'S COMPLEMENT OF A 16-BIT NUMBER

Example 1. Find one's complement of 5485 H.

The number in the binary form can be represented as follows :

5485 = 0101 0100 1000 0101
(5) (4) (8) (5)

One's complement = 1010 1011 0111 1010 = AB7A H
(A) (B) (7) (A)

The number is in the memory locations 2501 H and 2502 H.

The result is to be stroed in the memory locations 2503 H and 2504 H.

PROGRAM

Address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
2000	21, 01, 25	LXI	H C100H H, 2501 H	Address of LSBs of the number.

```
2003    7E          MOV    A, M       8 LSBs of the number in
                                      accumulator.
2004    2F          CMA               Complement of 8 LSBs of the
                           C103H      number.
2005    32, 03, 25  STA    2503 H     Store 8 LSBs of result.
2008    23          INX    H          Address of 8 MSBs of the
                                      number.
2009    7E          MOV    A, M       8 MSBs of the number in
                                      accumulator.
200A    2F          CMA               Complement of 8 MSBs of the
                           C104'      number.
200B    32, 04, 25  STA    2504       Store 8 MSBs of the result.
200E    76          HLT               Halt
```

Example 1

DATA

2501 — 85 H, LSBs of the number.

2502 — 54 H, MSBs of the number.

Result

2503 — 7A H, LSBs of the result.

2504 — AB H, MSBs of the result.

Example 2

DATA

2501 — 7E H, LSBs of the number.

2502 — 89 H, MSBs of the number

Result

2503 — 81 H, LSBs of the result.

2504 — 76 H, MSBs of the result.

The 8 LSBs of the number are in the memory location 2501 H. The address 2501 is placed in H-L pair. The 8 LSBs of the number are transferred from 2501 H to the accumulator. The instruction CMA takes one's complement of 8 LSBs. The 8 LSBs of the result are stored in the memory location 2503 H. The address of 8 MSBs of the number is 2502 H, and it is placed in H-L pair. The 8 MSBs of the number are transferred from 2502 H to the accumulator. The instruction CMA takes one's complement of 8 MSBs. The 8 MSBs of the result are stored in memory location 2504 H.

6.11 FIND TWO'S COMPLEMENT OF AN 8-BIT NUMBER

Example 1. Find two's complement of 96.

$$
\begin{array}{lllll}
 & 96 = & 1001 & 0110 & \\
 & & (9) & (6) & \\
\text{1's complement} & = & 0110 & 1001 & = 69 \\
 & & (6) & (9) & \\
 & + & 0000 & 0001 & \\
\hline
\text{2's complement} & = & 0110 & 1010 & = 6\text{A} \\
 & & (6) & (\text{A}) &
\end{array}
$$

Two's complement of a number is obtained by adding 1 to the 1's complement of the number.
The number is placed in the memory location 2501 H.
The result is to be stored in the memory location 2502 H.

PROGRAM

Address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
2000	3A, 01, 25	LDA	C100H 2501 H	Get data in accumulator.
2003	2F	CMA		Take its 1's complement.
2004	3C	INR	A C102 H	Take 2's complement.
2005	32, 02, 25	STA	2502 H	Store result in 2502 H.
2008	76	HLT		Stop

Example 1

DATA

2501 — 96 H

Result

2502 — 6A H

Example 2

DATA

2501 — E4 H

Result

2502 — 1C H

6.12. FIND TWO'S COMPLEMENT OF A 16-BIT NUMBER

Example 1. Find two's complement of 5B8C.

```
                      5B8C = 0101  1011  1000  1100
                             (5)   (B)   (8)   (C)
      1's complement       = 1010  0100  0111  0011 = A473
                             (A)   (4)   (7)   (3)
                           + 0000  0000  0000  0001
      ---------------------------------------------------
      2's complement       = 1010  0100  0111  0100 = A474
```

Two's complement of a number is obtained by adding 1 to the one's complement of the numbe
The number is stored in the memory location 2501 and 2502 H.
The result is to be stored in the memory location 2503 and 2504 H.

PROGRAM

Address	*Machine Codes*	*Label*	*Mnemonics*	*Operands*	*Comments*
2000	21, 01, 25		LXI	C100H H, 2501 H	Address of 8 LSBs of the number.
2003	06, 00		MVI	B, 00	Use register B to store carry.
2005	7E		MOV	A, M	8 LSBs in accumulator.
2006	2F		CMA		1's complement of 8 LSBs of the number.
2007	C6, 01		ADI	01	2's complement of 8 LSBs of the number.
2009	32, 03, 25		STA	C103H 2503 H	Store 8 LSBs of the result.
200C	D2,10, 20		JNC	GO	
200F	04		INR	B	Store carry.
2010	23	GO	INX	H	Address of 8 MSBs of the number.

```
2011    7E              MOV     A, M            8 MSBs in accumulator.
2012    2F              CMA                     1's complement of 8 MSBs of the
                                                number.
2013    80              ADD     B (104H)        Add carry.
2014    32, 04, 25      STA     2504 H          Store 8 MSBs of the result.
2017    76              HLT                     Stop.
```

Example 1

DATA

2501 — 8C, LSBs of the number.

2502 — 5B, MSBs of the number.

Result

2503 — 74, LSBs of the result.

2504 — A4, MSBs of the result.

2's complement of the number is A474.

Example 2. Find 2's complement of 5B00.

```
                    5B00 = 0101   1011   0000   0000
                            (5)    (B)    (0)    (0)
  1's complement         = 1010   0100   1111   1111
                            (A)    (4)    (F)    (F)
                         + 0000   0000   0000   0001
  ------------------------------------------------------
  2's complement         = 1010   0101   0000   0000
                            (A)    (5)    (0)    (0)
```

2's complement of the number is A500.

For the problem given above the data and results are as follows :

DATA

2501 — 00, LSBs of the number.

2502 — 5B, MSBs of the number.

Result

2503 — 00, LSBs of the result.

2504 — A5, MSBs of the result.

The 8 LSBs of the number are in the memory location 2501 H. They are taken first and 1's complement is obtained. To obtain 2's complement 1 is added to 1's complement. 8 LSBs of 2's complement are stored in 2503 H. The carry resulting from the addition of 1 to 1's complement is stored in register B. After this 8 MSBs of the number are taken and 1's complement is obtained. The carry is added to it. The 8 MSBs of the result are stored in 2504 H. In case of no carry the program jumps from JNC GO to INX H and the content of register B is not incremented. It remains zero. The addition of zero to 1's complement of 8 MSBs does not affect the result.

* Interrupts of Intel 8085:-

- The Intel 8085 has five interrupt inputs namely TRAP, RST 7.5, RST6.5, RST 5.5 and INTR. The TRAP has the highest priority, followed by RST 7.5, RST 6.5 and RST 5.5. The INTR has the lowest priority. when interrupts are to be used they are enabled by software using the instruction EI (Enable Interrupt). in the main program. Fig. shows the schematic diagram of interrupts of Intel 8085 The instruction EI sets the interrupt enable Flip-Flop to enable the interrupts. The use of the instruction EI enables all the interrupts. The instruction DI (Disable interrupt) is used to disable interrupts. When microprocessor perfoming a particular task then the using DI instruction we can prevent the occurrence of interrupts in µP.

- The DI instruction resets the interrupt enable Flip Flop and disables all the interrupts except nonmaskable interrupt TRAP. using system RESET we can also resets the Interrupt Enable Flip-Flop.

- When an interrupt line goes high processor completes its current instruction and saves program counter on the stack. It also resets Interrupt Enable Flip-Flop before taking up ISS (Interrupt service subroutine) so that the occurence of further interrupts by other devices is prevented during the execution of ISS. All the interrupts except TRAP are disabled by resetting the Interrupt Enable Flip-Flop.

→ The resetting of this Flip-Flop can be done in one of the three ways:

1) By software using instruction

2) system reset

3) By recognition of an interrupt request.

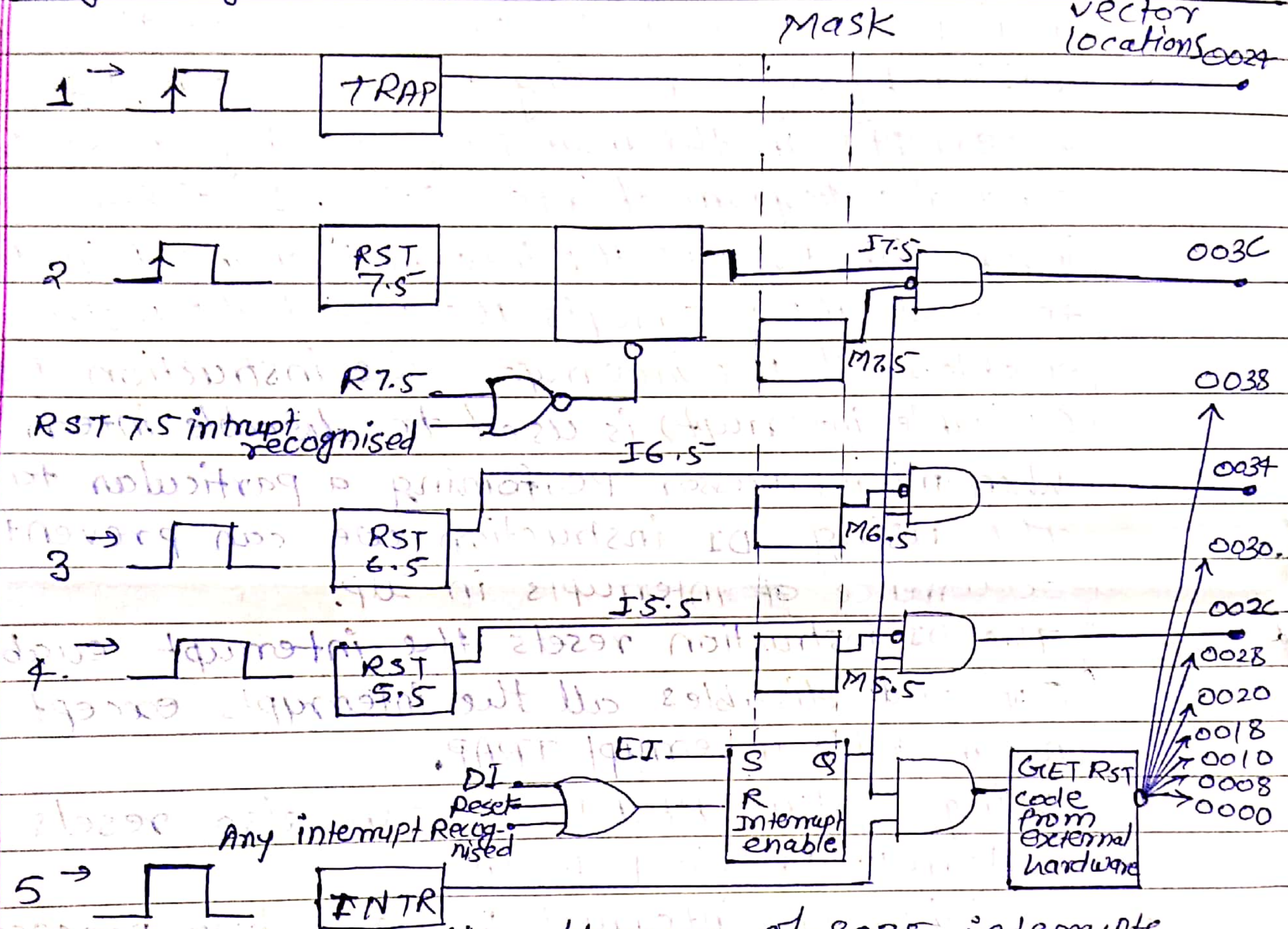

fig. Schematic diagram of 8085 interrupts

→ Before the program returns back from ISS to the main program all the interrupts are to be enabled again. This is done using instruction EI in ISS before using the instruction RET.

There are two types of interrupts in intel 8085 up

① Non maskable interrupt

② Maskable interrupt

→ TRAP is a nonmaskable interrupt. It need not be enabled. It cannot be disabled. It is not accessible to user. It is used for emergency situation such as power failure and euergy shut-off.

→ RST 7.5, RST.6.5, aud RST 5.5 are maskable interrupts. at many times the programmer may like to prevent the occurrence of a few of several ~~†~~ interrupts while up is performing certain tasks. This is done by masking off those interrupts which are required to occur when certain task is being performed. These interrupts which can be masked off (i.e. made ineffective) are called maskable interrupts. Masking is done by software.

* Hardware aud Software Interrupts :-

→ Interrupts caused by I/O devices are called hardware interrupt.

→ The normal operation of a up can also be interrupted by abnormal internal conditions or special instructions. such an interrupt is called a software interrupt.
RST n instructions of the 8085 are used for software interrupt. when RSTn instruction is inserted in a program, the program is executed upto the point where RSTn has been inserted.

→ The internal abnormal or unusal conditions which prevent the normal processing sequence of a up are also called exceptions.
for example, divide by zero will cause an

exception. Intel literatures do not use the term exception, whereas Motorola literatures use the term exception.
Intel includes exception in software interrupt.

→ When several I/o device are connected to INTR interrupt line, an external hardware is used to interface I/O devices. The external hardware circuit generates RST n codes to implement the multiple interrupt scheme.

* Interrupts call - locations (Software interrupt call location) :-
When an interrupt occurs the program is transferred to a specific memory location. Then the monitor transfers the program from the specific memory location to a memory location in RAM, from where the user can write the program for interrupt service sub-routine.

Interrupt	call -location in Hex
TRAP	0024
RST 7.5	003C
RST 6.5	0034
RST 5.5	002C

An interrupt for which hardware automatically transfers the program to a specific memory location is known as vectored interrupt.

INTR call locations (Hardware interrupt call location) :-

There are 8 numbers of call-locations for INTR interrupt. Table shows CALL-locations, RST n instructions and corresponding hex-code. For INTR external hardware is used to transfer program to specific call-location. The hardware circuit generates RST codes for this purpose. When INTR is high, the µp saves the contents of the program counter on the stack and then sends an interrupt acknowledge signal, INTA to the external hardware. In response to INTA the external hardware generates RST n code. when µp receives this code, it tranfer program to the corresponding call-location. upto 8 number of I/O devices can be connected to INTR through an external hardware.

RSTn	Hex-code	call-locations
RST 0	C7	0000
RST 1	CF	0008
RST 2	D7	0010
RST 3	DF	0018
RST 4	E7	0020
RST 5	EF	0028
RST 6	F7	0030
RST 7	FF	0038