6.13. ADDRESSING MODES

etc.

To perform any operation using microprocessor, we have to give the corresponding instruction to the microprocessor. In each instruction, user has to give following information :

(*i*) Operation to be performed

(*ii*) Address of source of data

(*iii*) Address of destination of result

The Method by which address of source of data or address of destination of result is given in the instruction is called addressing mode.

There are five types of addressing modes in microprocessor 8085.

1. Immediate addressing mode : If 8/16 bit data required for executing the instruction is given directly along with the instruction, then such instructions are called immediate addressing mode instructions.

Mostly the instructions with letter I (Immediate) fall under this category along with those which can be directly loaded with 16 bit number.

The length of instructions is two or three bytes.

Instruction Format

(*a*) 8 bit data

First Byte	*Second Byte*
Opcode	Data

(*b*) 16 bit data

First Byte	*Second Byte*	*Third Byte*
Opcode	Data (lower)	Data (Higher)

e.g.

I. MVI C, 25 H

25 ⟵ 25 H (8)

C

This instruction moves immediately 25 H to register C.

II. LXI B, 2500 H

25 | 00 ⟵ 2500 H

B C

This instruction loads immediately 2500 H to B – C pair.

2. Register Addressing Mode : If 8/16 bit data required for executing the instruction is present in Register/Register pair and name of the register/Register pair is given along the register the instructions are called register addressing mode instructions.

As the data is in CPU *i.e.* in register/register pair, it results in faster execution due to less number of T-states.

The length of instruction is one byte.

Instruction Format

First Byte
OPCODE r_d , r_s

where

$r_d \rightarrow$ Destination register

$r_s \rightarrow$ Source register

e.g.

I. MOV B, A

25H ⟵ 25H

B A

This instruction moves the contents of register A to register B. Suppose register A contains 25 H

II. ADD B

This instruction adds the contents of register B with Accumulator contents. After execution, result is also placed in Accumulator.

Suppose register B contains 02 H and accumulator contains 06 H.

3. Direct Addressing Mode : If 8/16 bit data required for executing the instruction is present in memory and 16 bit address of this memory location is given along the instruction or we can also say that the address of data is directly specified with instruction. Now this address may be address of source of data or address of destination of result.

Byte 2 and 3 contains the address of data.

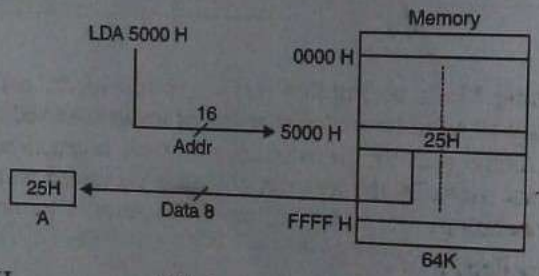
The length of the instruction is three bytes.

First Byte	*Second Byte*	*Third Byte*
Opcode	Lower byte of Addr	Higher Byte of Addr

e.g.

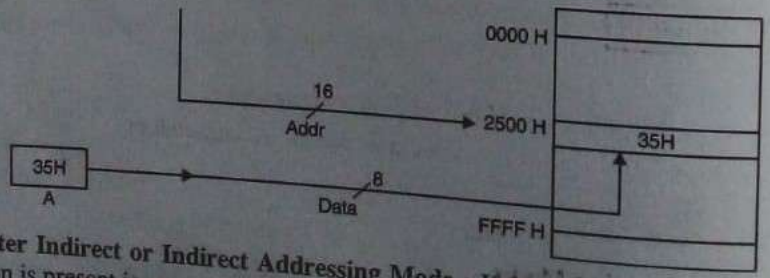
I. LDA 5000 H

This instruction loads the contents of specified memory location 5000 H in Accumulator.



II. STA 2500 H

This instruction stores the contents of Accumulator in specified memory location 2500 H.



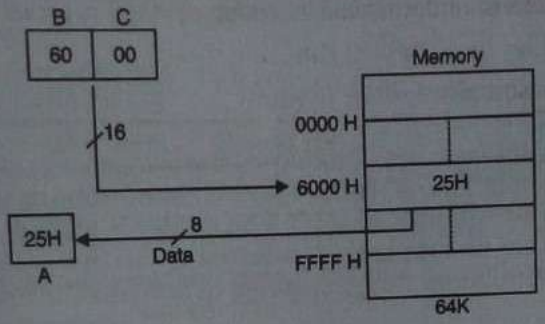
4. Register Indirect or Indirect Addressing Mode : If 8/16 bit data required for executing the instruction is present in memory location, the 16 bit address of this memory location is present in 16 bit Register pair and name of this 16 bit Register pair is given along with instruction. We can also say that the memory address is specified indirectly by the contents of register pair. Actually in this type address of address is specified with the instructions.

The length of instructions is one byte.

First Byte
Opcode

e.g.

I. LDAX B — *This instruction load the contents of memory location whose address is in the register pair BC, into Accumulator.*



II. MOV M, A

III. STAX r_p

5. Implied Addressing Mode or Implicit A.M. : In this mode, data is not specified. If address of source of data as well as address of destination of result are fixed, then there is no need to give any data/operand along with the instructions and such instructions are called Implicit addressing mode instructions. Actually the location of data is contained within the opcode itself. Generally, in this mode, operations are performed on the contents of accumulator.

The length of the instructions is one byte.

Instruction Format

First Byte
Opcode

e.g.

CMA (Complement Accumulator)
DAA (Decimal Adjust Accumulator)
RAL
RAR etc.

Summary of Addressing Modes

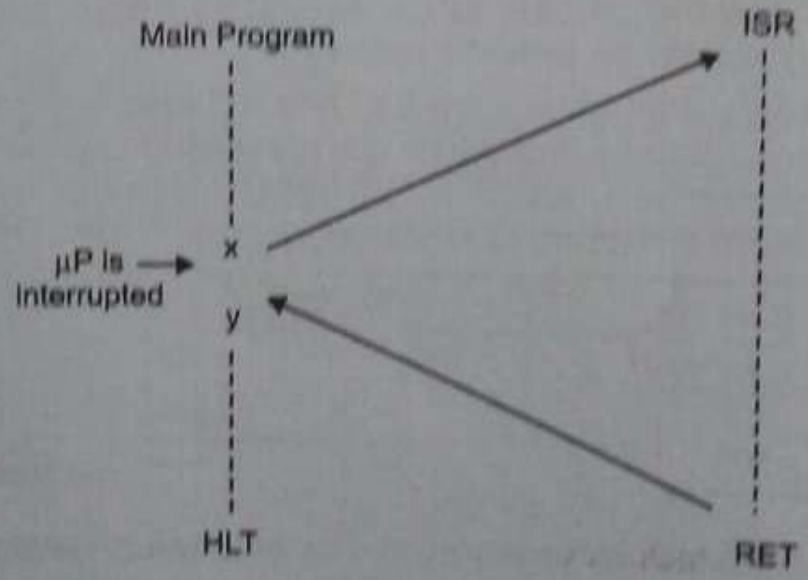
1. Immediate Addressing Mode : INSTRUCTION · DATA
2. Register Addressing Mode : INSTRUCTION · R/RP ; [Data] R/RP
3. Direct Addressing Mode : INSTRUCTION · ADDRESS
4. Indirect Addressing Mode : INSTRUCTION R/RP ; [ADDR] R/RP
5. Implied Addressing Mode : INSTRUCTION

6.5. INTERRUPTS AND ITS TYPES

An interrupt is a subroutine call initiated by an external device through hardware (*Hardware interrupt*) or μP itself (*Software interrupt*). Actually it is the process of temporary transfer of control of μP from one program (*main program*) to other (*subroutine*) which is causing the interrupt.

Interrupt is a process of data transfer whereby an external device informs the μP about its readiness for communication and requests its attention. External device may interrupt the process at any time without reference to a system clock : *i.e.* Interrupt request is an asynchronous event that can occur at any time during execution of the main program.

In response to an interrupt request, μP performs the following steps :



(*i*) The μP completes the current instruction execution '*x*'.

(*ii*) μP saves the address of next instruction *y* from PC into stack (*i.e.* PUSH PC) *i.e.* return address.

(*iii*) Now μP transfer its control to Interrupt Service Routine (ISR).

(*iv*) When μP executes 'RET' instruction in the ISR (RET *is always the last instruction in* ISR) it returns back to next instruction *y* of main program.

So whenever μP is interrupted, the μP executes ISR.

There are two types of interrupts :

(*a*) Hardware Interrupts (*b*) Software Interrupts

6.5.1. Hardware Interrupts

If μP is interrupted by applying signal on hardware interrupt pin, then it is called hardware interrupt.

There are five hardware interrupts :

Interrupt	Priority	Vector Location
TRAP	1	0024 H
RST 7·5	2	003C H
RST 6·5	3	0034 H
RST 5·5	4	002C H
INTR	5	.....

The first four are automatically vectored (*transferred*) to specific memory locations on memory page 00 H and do not require any external hardware. The necessary hardware is already provided

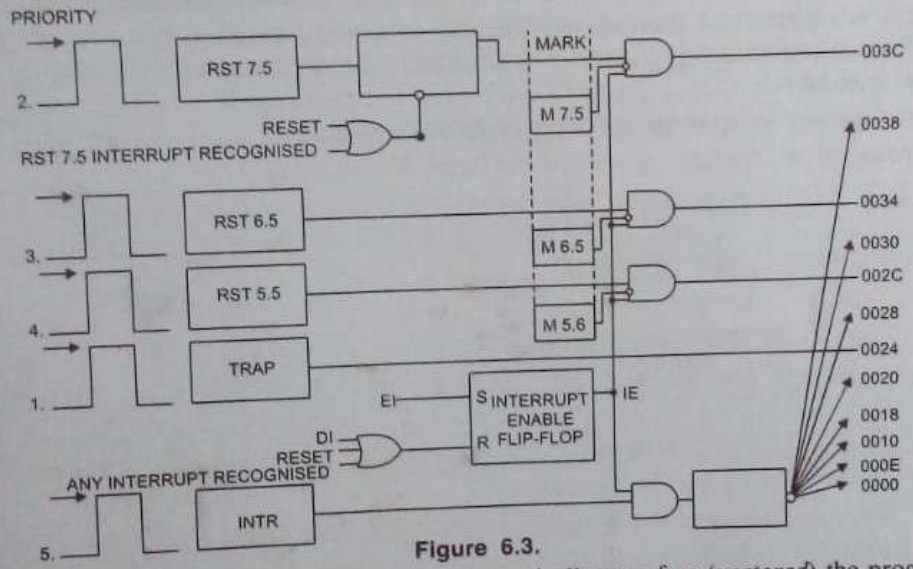


Figure 6.3.

within μP. So interrupts for which hardware automatically transfers (*vectored*) the program to a specific memory location is known as vectored interrupts.

(a) TRAP : Trap is a non-maskable interrupt also known as NMI. As the name indicates, NON MASKABLE, It cannot be masked (*disable*).

TRAP has the highest priority. It is level & edge sensitive *i.e.* I/P signal at this pin should go from low to high and stay high to be acknowledged. This avoids false triggering caused by noise & transients.

We can see from Figure 6.4, the positive edge of TRAP signal sets the D flip flop. However due to the AND gate, we also require a continuous high level TRAP input.

Trap flip flop can be cleared in two ways :

(*i*) By resetting the μP *i.e.* giving low signal on $\overline{\text{RESET IN}}$ pin.

(*ii*) By giving a high TRAP ACKNOWLEDGE.

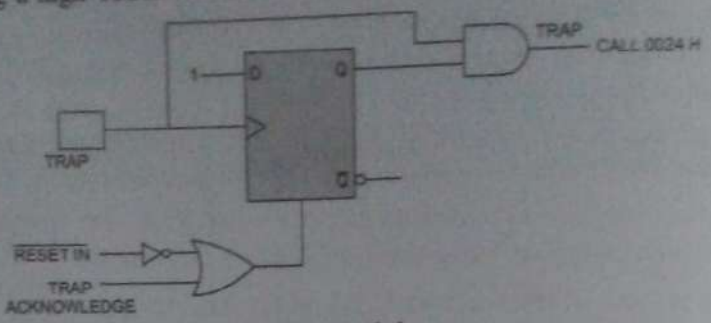


Figure 6.4.

After recognizing a TRAP I/P, µP sends a high TRAP ACKNOWLEDGE bit to TRAP flip flop & clears it.

When TRAP interrupt is triggered, µP first completes its current instruction, then pushes the address of next instruction *i.e.* return address onto the stack. The PC is then loaded with fixed vector location address 0024 H and now ISR starts from this address. (Starting address $8 \times 4{\cdot}5 = 0024$ H).

TRAP interrupt is generally used for critical events such as power failure & emergency shut off.

(*b*) **RST 7·5 :** It has the second highest priority. It is +ve edge triggered and can be triggered with a short pulse. This request is stored internally by the D flip flop as shown in Figure 6.2 until it is cleared by reset or SIM instruction.

RST 6·5 can be cleared by two ways :

(*i*) Giving a high R 6·5 bit using SIM instruction.

(*ii*) By internally generated high RST 6·5 ACKNOWLEDGE. This occurs automatically as follows :

On recognizing a RST 7·5 interrupt, µP sends a high RST 7·5 ACKNOWLEDGE bit to second D flip flop which clears it for future use.

RST 7·5 vectors to a memory location 003C H RST 7·5 is maskable.

(*c*) **RST 6·5 & RST 5·5 :** Both these are level sensitive *i.e.* triggering level should remain high until the µP completes the execution of current instruction.

Both are maskable and can be enabled using SIM instruction.

Both have lower priority than TRAP & RST 7·5.

The vector addresses of RST 6·5 & RST 5·5 are 0034 H and 002C H respectively.

Note : Vector addresses of TRAP, RST 7·5, RST 6·5 and RST 5·5 are easy to remember.

These can be calculated by multiplying the specified number by 8 and converting it to Hexadecimal (dividing by 16).

For example, RST 5·5

$$\text{Vector address} = 5{\cdot}5 \times 8 = 44 \text{ decimal} = 002C \text{ H}$$

One can remember the vector address of TRAP by considering it as RST 4·5.

$\therefore$ Vector address $= 4{\cdot}5 \times 8 = 36{\cdot}0 = 0024$ H

(*d*) **INTR :** INTR is an abbreviation for interrupt request. It has the lowest priority. The following steps takes place if INTR occurs :

1. When µP executes a program, it checks the INTR during execution of each instruction more precisely it checks in last T-state of last machine cycle.
2. If INTR is high, the µP completes its current instruction and sends an active low signal $\overline{\text{INTA}}$ (interrupt acknowledgment).
3. $\overline{\text{INTA}}$ is used to insert an instruction through additional hardware using 8 bit tristate buffers.
4. On receiving this instruction, the µP saves the address of next instruction (*i.e.* return address) onto the stack and performs CALL instruction.
5. The µP performs the ISR until it counters RET instruction.
6. The service routine should include EI to enable the interrupt again.

The µP checks the INTR one cycle before the end of instruction cycle. In 8085, the CALL instruction requires 18 T-states, hence INTR should be high for atleast 17·5 T-states. With a clock frequency of 3 MHz the I/P pulse to INTR should be 5·8 µs long.

INTR is a maskable interrupt.

The starting addresses of INTR are variable, it depends upon the instruction which is transferred by external hardware to µP. This hardware ckt. generates RSTn codes for this purpose. There are 8 numbers of CALL locations for INTR *i.e.* In RSTn, *n* is from 0 to 7, RST 0 to RST 8.

SUMMARY OF HARDWARE INTERRUPTS

S.No.	Interrupt pin name	Priority	Signal	Type	Control	ISR Addr
1.	TRAP	1st	⎍	Non Maskable	NIL	$8 \times 4{\cdot}5 = 0024$ H
2.	RST 7·5	2nd	⎍	Maskable	$Q_i = 1$ M 7·5 = 0 R 7·5 = 0	$8 \times 7{\cdot}5 = 003C$ H
3.	RST 6·5	3rd	⎍	Maskable	$Q_i = 1$ M 6·5 = 0	$8 \times 6{\cdot}5 = 0034$ H
4.	RST 5·5	4th	⎍	Maskable	$Q_i = 1$ M 5·5 = 0	$8 \times 5{\cdot}5 = 002C$ H
5.	INTR	5th	⎍	Maskable	$Q_i = 1$	Variable depends upon instructions transferred by external hardware to µP. (RSTn)