Lecture-18

INSTRUCTION SET

Classification of instruction set:

The 74 instructions available in the 8085A can be divided into five groups depending on their function.

1. Data transfer group: Instruction that more data between registers, between register and memory location and I/O transfer.
2. Arithmetic group: Instructions that add subtract increment or decrement data in the register.
3. Logic group: Instruction that carry out logic operation, such an AND, OR, EX-OR compare between and data in the accumulator & a register compliment and rotate data in the accumulator.
4. Branch group: Instruction that change .the execution sequence of a program, such as conditional and unconditional jump instruction and subroutine call and return instruction.
5. Stack machine control group: Instruction for maintaining the stack and internal control flags.

DATA TRANSFER GROUP:

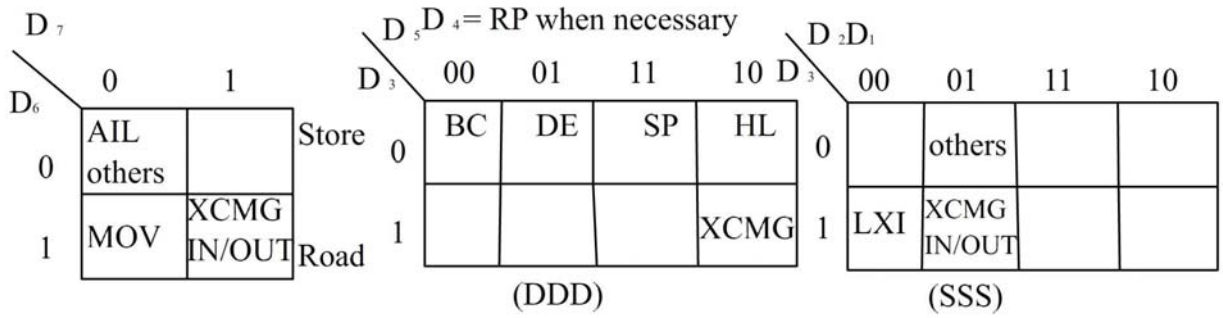
It consists of 15 basic operation and 86 variations. The basic operation involved is DATA transfer between two register of a microprocessor system. One of the register is always located in the μP itself the other may be located in one of the following

1) An I/o device
2) Memory
3) The microprocessor

Registers located in the microprocessor are referred to as internal registers. (A, A, C, D, E, H, L, SP, PC) and those in ROM ,RWM, or I/O are referred to as external registers therefore, this group includes transfer of data from reg to mg, Reg to memory, memory to reg. Reg (a) to output devices, or from input device to reg (A).

The register from which data in transfers is the source register and the register to which data is transferred in the destination register. A transfer involves copying the contents of the source reg into the destination register; the contents of the source register and not attend. Each data transfer instruction identifies the source register and the destination register. Identification of one or both of these register may be implied by the instruction in memories or may be explicit. Internal registers are frequently implied; whereas the external registers are usually identified by an explicit address that is part of the instruction.

The operation code format pr shown on fig.



MOV r_1, r_2: This is an ALP statement, MOV is the mnemonic for move. r_2 is the source register. r_1 is the destination register in the offer and fields the meaning of the instruction is move the contents of the register r_2 into r_1. The content of r_2 is not destroyed .content of $r_1$os destroyed and new value form r_2 takes it place. The macro RTL implemented is

$$(r_1) \leftarrow (r_2)$$

This is a single byte instruction at memory location N. The code of will be

0 1 D D D S S S	N

It has 49 variations (7x7) seven for SSS and seven for DDD. It is register addressing mode. How the execution of this instruction takes place. During the T_1 state of the first M/C up to (OFMC), the contents of the program contents are placed on the address bus (A_{15} ...A_8) and address data bus (AD_1- AD_0). Since address may be low or high, it is customary to use the double sided waveforms, the high byte of the program counter (PCM) goes to the address bus and the low byte (PCM) to the address data bus.

The ALE signal initially goes high then midway through the T_1 state, ALE goes low 2 ties this falling edge that latches the address in to the external latch. Also IO / N go low near the beginning of the T_1 state there enables the peripheral chips for a memory operation rather than an I/O operation.

During the T_2 state, the program counter is incremented. The address disappears from the address –data bus at the beginning of the T_2 state. This is
Necessary because as instruction fetch is in needed. The dashed line an AD_1-AD_0. Wave from means that the data on the bus is invalid. Towards the end of the T_2 state, the OP code appears on the address – data bus the precise time when OP code appears depends on the memory access time; the length of the buses and other factors.

At the beginning of the T_2 state $\overline{RD}$ goes low and stays low until the middle of the T_1 state during the T_3 state, the OP code on the AD bus is copied into the instruction reg. During the T_4 state, the instruction is decided. The pl now knows that the instruction is MOV r_1, r_2 21 is represented as MOV r_1, r_2=1 and is executed.

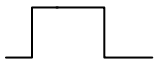
In fact the execution does not take place instantaneously, when the instruction is decoded, the contents of register 2 are copied states T_1,T_2 the contents of the temporary register are copied into register 1 this completes the
Execution of the MOV instruction since (in these two states) during MC-2 up on only the internal bus is used. Add r bus and add ^{d}ata bus

are not used, therefore there buses can be used for some other purpose.

One way to save processing time is by starting the next instruction up to during the second M/C of the move instruction. This is called fetch executed overlap. During T_1 of this m/c, the contents of the program counter are out pattern to Add_r bus & AD bus during T_2, state the next OP code is transferred to AD bus form memory and simultaneously MOV r_1, r_2 instruction is implied. Hence, during the MC-2 of the move instruction cycle, MC–1 of the next cycle is operative.

Thus the instruction cycle for MOV r_1,r_2 takes only L_1 clock states (& not six) after the executions PC point to next address.

MC- 1.

OFMC T_1 : AD,- AD ← (PCL) , A_{15} – $A_{5=}$← (PCH), ALE = 

IO/$\overline{M}$ = 0 T_2 : $\overline{RD}$ = 0, (P'C) ← (PC) + 1, AD_1- AD_0 ← M(AB)

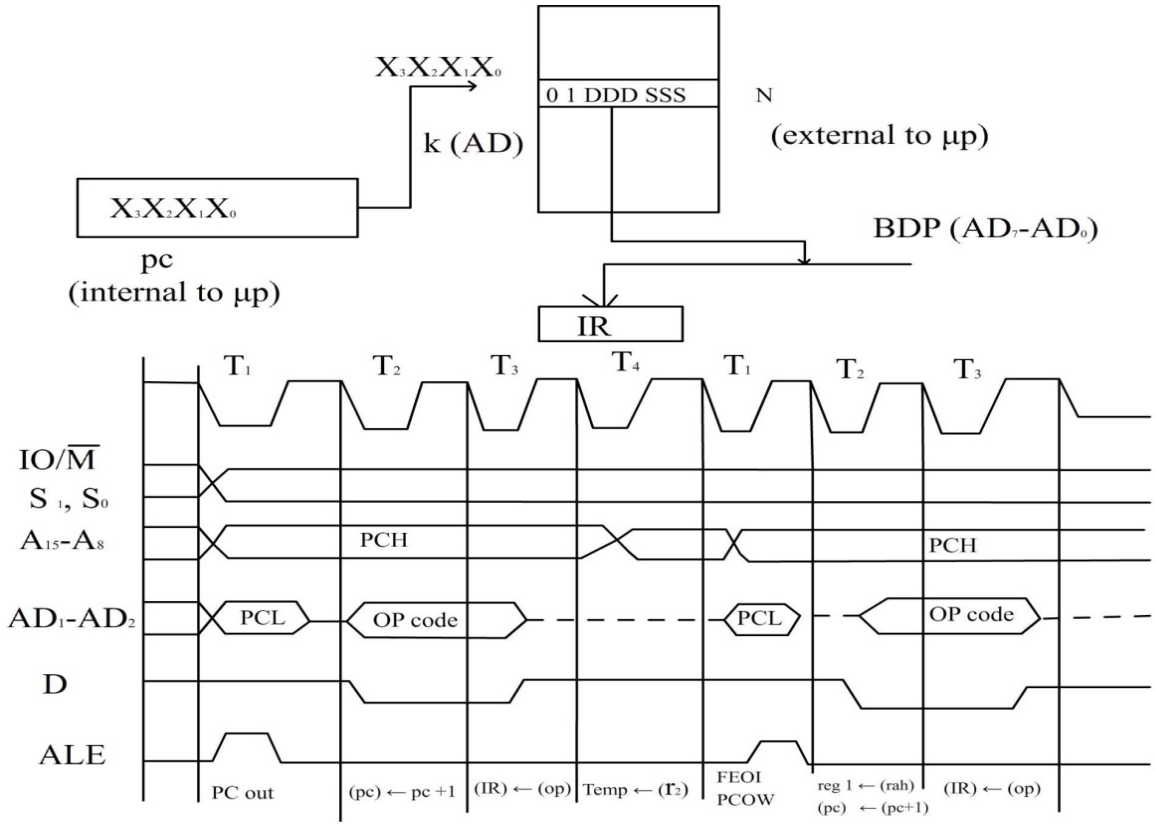
S_1 = 1 T_3 : $\overline{RD}$ = 1, ↑ , (IR) ← (AD_2- AD_0)

S_0 = 1 T_4 : MOV r_1, r_2= 1, ---------- approximately or (Z) ← (r_2)

Mc – 2

FEO T_1 : AD_7- AD_0 ← (PCL) , A_{15} – $A_{5=}$← (PCH), ALE

(r_1) ← (z) T_2 :: $\overline{RD}$ = 0, (PC) ← (PC) + 1, AD_1- AD_0 ← M(AB)



This Operation Require Only Single M/C Cycle of OFMC & needs only four state.

MC -1 OFMC

T_1	T_2	T_3	T_4

INSTRUCTION CYCLE

Now after execution of this instruction pointer goes to T_1 state & PC points to next address.

MOV r, M: This is an ALP statement, the meaning of this instruction is the content of the memory location whose address is available into (H,L) pair should be moved into the internal general purpose register r. The macro RTL implemented.

$$r \leftarrow M(H,L)$$

M is the source, v is the destination. It is a single byte instruction at memory location N. The operation code is,

01 DDD 110	N

It has seven variations. DDD cannot be 110 because memory to memory transfer is not allowed.

The micro RTL follows:

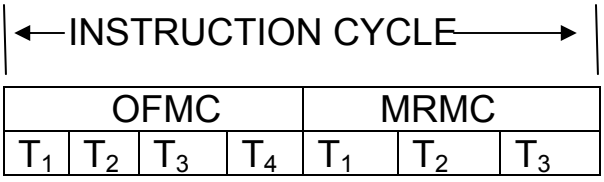
MC – 1
OFMC
IO / $\overline{M}$ = 0
S1 = 1
S0 = 1

T1: $AD_7 - AD_0 \leftarrow (PCL),\ A_{15} - A_8 \leftarrow (PCH),\ ALE =$
T2: $\overline{RD} = 0, (PC) \leftarrow (PC) + 1,\quad AD_7 - AD_0 \leftarrow M\ (AB)$
T3: $\overline{RD} = 1, \uparrow, (IR) \leftarrow (AD_7 - AD_0)$
T4: MOV r, M = 1

C – 2
OFMC
IO / $\overline{M}$ = 0
S1 = 1
S0 = 0

T1: $AD_7 - AD_0 \leftarrow (L),\ A_{15} - A_8 \leftarrow (H),\ ALE =$
T2: $\overline{RD} = 0,\ AD_7 - AD_0 \leftarrow M\ (AB)$
T3: $\overline{RD} = 1, \uparrow, (R) \leftarrow (AD_7 - AD_0)$

This Operation requires only two m/c cycle OFMC & MRMC and total 7 states. It requires 3.5μ sec using 2 MHz clock



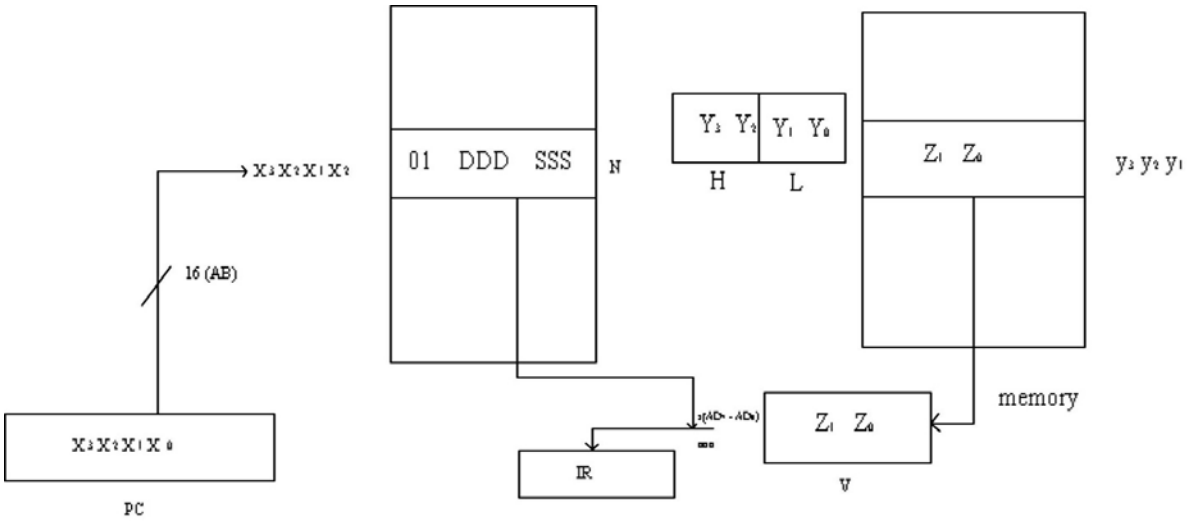


Fig.7

The addressing mode is register indirect addressing mode fig 7 illustrates the operation cycle.

MOV M, r This is an ALP statement. The meaning of the instruction is then content of the internal general purpose register r should be move into the memory location whose address is available is

$$M(H, L) \leftarrow (r)$$

It is a single byte instruction. The operation code is

01 110 SSS	N

It has seven variations SSS cannot be 110.

The micro RTL flow is

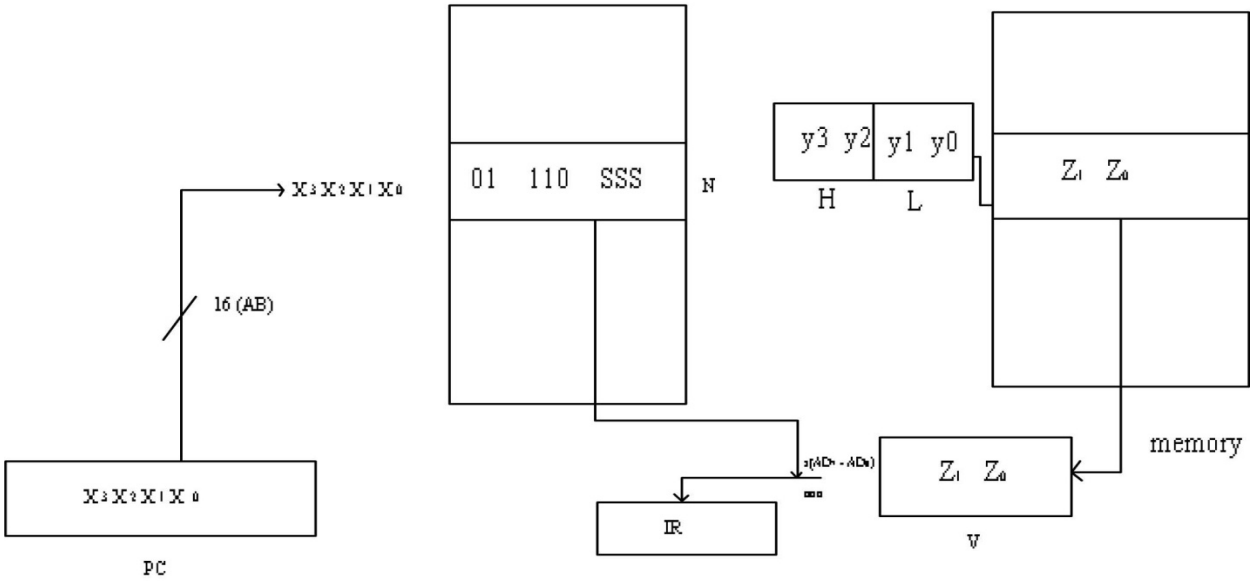
$MC-1$	$T1: AD_7 - AD_0 \leftarrow (PCL),\ A_{15} - A_8 \leftarrow (PCH),\ ALE =$
OFMC	$T2: \overline{RD} = 0,\ (PC) \leftarrow (PC) + 1,\quad AD_7 - AD_0 \leftarrow M(AB)$
$IO/\overline{M} = 0$	$T3: \overline{RD} = 1, \uparrow, (IR) \leftarrow (AD_7 - AD_0)$
$S1 = 1$	$T4: MOV\ MV = 1$
$S0 = 1$	

$MC-2$	
MWRMC	$T1: AD_7 - AD_0 \leftarrow (L),\ A_{15} - A_8 \leftarrow (H),\ ALE =$
$IO/\overline{M} = 0$	$T2: \overline{WR} = 0,\quad AD_7 - AD_0 \leftarrow M\ (v)$
$S1 = 0$	$T3: \overline{WR} = 1, \uparrow, M(AB) \leftarrow (AD_7 - AD_0)$
$S0 = 1$	

The Operation requires only Two m/c cycles- OFMC & MWRMC and total no of 7 states. It requires 3.5 μp using 2 MHZ internal clocks. It is register indirect addressing mode.

INSTRUCTION CYCLE

OFMC				MWRMC		
T_1	T_2	T_3	T_4	T_1	T_2	T_3



MVI r, DATA:

This is an ALP statement DATA is the symbolic name given to 8 bit data which is immediately available as second byte of the instruction. Therefore, the source of data is the 2nd byte instruction itself. Therefore, it is immediately addressing mode MVF is the mnemonic for move immediate v is the destination register. It is a 2 byte instruction at the memory location N & N+1. The opcode is,

00 DDD 110	N
<B2>	N+1

The meaning of the instruction is 8 bit data immediately available as a 2nd byte of the instruction should be loaded into the destination register. Whose code is DDD. It has 7 variations

$$(v) \leftarrow < B_2 >$$

The micro RTL flow is

$MC-1$

OFMC

$IO/\overline{M}=0$

$S1=1$

$S0=1$

$T1: AD_7-AD_0 \leftarrow (PCL),\ A_{15}-A_8 \leftarrow (PCH),\ ALE=$

$T2: \overline{RD}=0,\ (PC) \leftarrow (PC)+1,\quad AD_7-AD_0 \leftarrow M(AB)$

$T3: \overline{RD}=1, \uparrow, (IR) \leftarrow (AD_7-AD_0)$

$T4: MMIV=1$

$MC-2$

MRMC

$IO/\overline{M}=0$

$S1=1$

$S0=0$

$T1: AD_7-AD_0 \leftarrow (PCL),\ A_{15}-A_8 \leftarrow (PCH),\ ALE=$

$T2: \overline{RD}=0,\ (PC) \leftarrow (PC)+1,\quad AD_7-AD_0 \leftarrow M(AB)$

$T3: \overline{RD}=1, \uparrow, (v) \leftarrow (AD_7-AD_0)$

It require only two m/c cycles OFMC & MRMC and 7 states. It requires 3.5μ using 2 MHz internal clock.

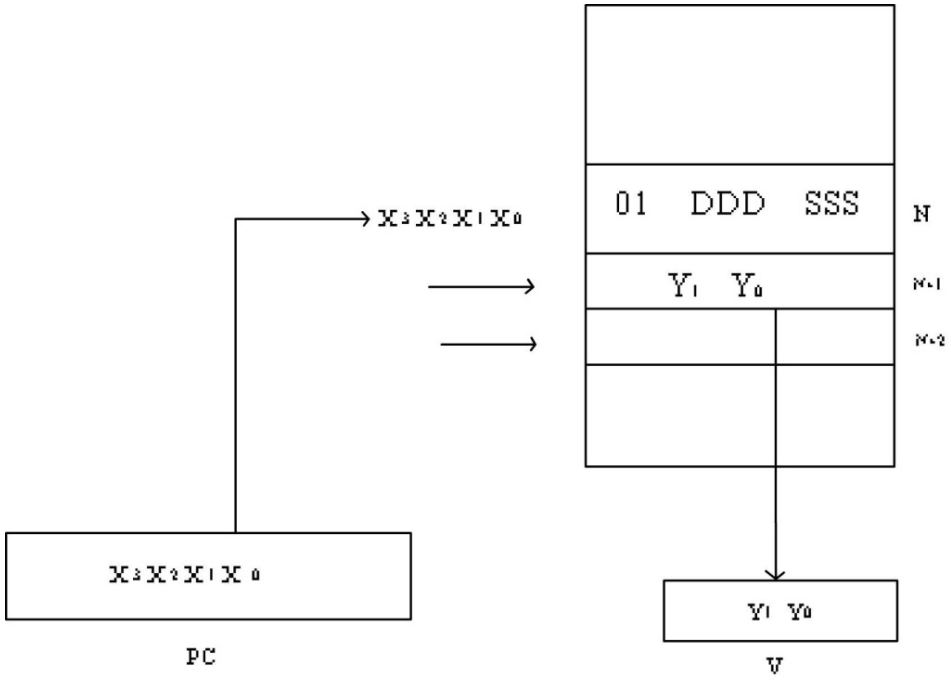


Figure. 9