MODULE 2 :- Addressing Modes, Instruction classification, Instruction Format, Data transfer operation, Arithmetic operations, Logical Operations, Stack and subroutine operation, Looping, Counting and indexing operations. (1)

INSTRUCTION FORMAT :-

- An instruction is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions, called the instruction set, determines what functions the microprocessor can perform.
- Each instruction consists of two parts. The first part is called opcode and the second is called as operand.
- The opcode part of an instruction specifies the operation to be performed. The operand can be specified in a no of ways. It includes
  (i) 8 bit/16 bit General purpose Register.
  (ii) A memory location.
  (iii) A 8 bit port address /16 bit memory address
  (iv) Implicit operand : The operand is not specified, instead it is assumed in register.

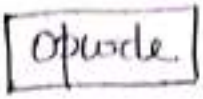
Depending upon the no of bytes or bit patterns required to specify an operation and the data, the instructions are of 3 types:

- One byte instructions
- Two byte instructions
- Three byte instructions

1 BYTE INSTRUCTIONS

- A 1 byte instruction includes the opcode and the operand in the 8 bit only i.e one byte.
- The one byte instructions specify the operation to be performed and who is going to perform it. These instructions require one (single) memory location.

FORMAT



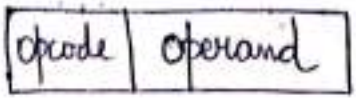
EXAMPLE

1) MOV B, C
- This instruction moves the contents of register C in register B.
- This instruction has an opcode 41H and is a one byte instruction. This instruction copies the content of register C in B register.

2 BYTES INSTRUCTIONS

- The 2 bytes instruction uses first byte to specify the operation ie opcode and second byte to specify the operand.
- These instructions requires two successive memory location in the memory.
- First byte stored is opcode and second byte stored is data ie operand or address.

FORMAT



EXAMPLE

MVI B, 57H
- This instruction moves the data 57H in register B.
- The first byte of this instruction is the opcode for instruction MVI B, and the second byte is the data that is to be moved to register B.

3 BYTE INSTRUCTIONS

The 3 bytes instruction uses first byte to specify the operation ie opcode, second and third bytes are used to specify the operand.

first byte stored is opcode, second byte stored is lower order 8 bits of 16 bits operand or address and third byte stored is higher order 8 bits of 16 bits operand or address.

FORMAT

opcode	operand	operand

EXAMPLE

STA 5600 H

- This instruction moves the contents of accumulator to the memory location 5600H.
- The first byte of this instruction is opcode for instruction STA 5600H and the second and third bytes are the address 5600H

OPCODE FORMATS

- The microprocessor 8085 is an 8 bit microprocessor and has 8 bit opcodes. Each instruction has a unique opcode.
- The opcode contains information regarding the operation, type of operation to be performed, registers to be used, flags. The opcode is fixed for each instruction.

Notations used in object code or OPCODE are:

Notations	Meaning
ddd	Destination register(s)
sss	Source register ddd = sss 111 = A register 000 = B register 001 = C register 010 = D register 011 = E register 100 = H register 101 = L register
nnn	Restart no 000 to 111

Notations	Meaning
yy yyyy	An 8 bit binary data A 16 bit binary data unit
x	Register pair 0 = BC 1 = DE
xx	Register pair 00 = BC 01 = DE 10 = HL 11 = SP (PUSH / POP) PSW
PPQQ	A 16 bit memory address

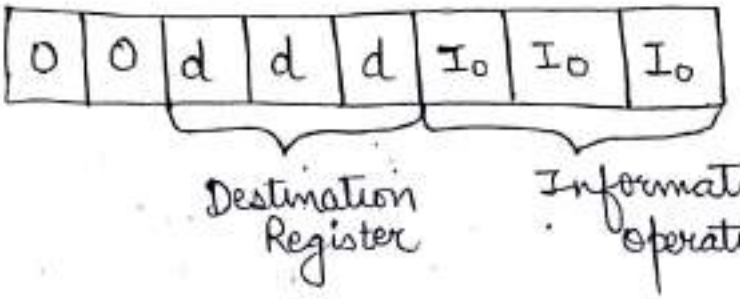
INFORMATION OPERATIONS			
I_0	I_0	I_0	Operation
0	0	0	Read / set interrupt mask
0	0	1	Immediate operation Rp
0	1	0	Load / store
0	1	1	Increment / Decrement Rp
1	0	0	Increment Single Register
1	0	1	Decrement Single Register
1	1	0	Immediate operation on Single Register
1	1	1	Register Shifting / Miscellaneous

ARITHMETIC LOGIC UNIT OPERATION			
A_L	A_L	A_L	Operation
0	0	0	Add
0	0	1	Add with carry
0	1	0	SUB
0	1	1	SUB with borrow
1	0	0	Logical AND
1	0	1	X-OR
1	1	0	Logical OR.
1	1	1	Compare

BRANCH CONDITION			
CB	CB	CB	Operation
0	0	0	JNZ
0	0	1	JZ
0	1	0	JNC
0	1	1	JC
1	0	0	JPO
1	0	1	JPE
1	1	0	JP
1	1	1	JM

BRANCH OPERATION			
B_0	B_0	B_0	Operation
0	0	0	Conditional return
0	0	1	Simple return / Miscellaneous
0	1	0	Conditional jump
0	1	1	Unconditional jump / Miscellaneous
1	0	0	Conditional CALL
1	0	1	Simple CALL / Miscellaneous
1	1	0	Special A/L operations
1	1	1	Special unconditional jump.

For all the data transfer instructions except MOV instruction format, the opcode is:



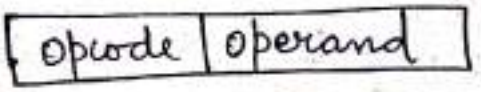
ADDRESSING MODES :-

The different methods (modes) to select (address) the operands are called the Addressing Modes. For 8085, they are:

- Immediate addressing
- Register addressing
- Direct addressing
- Indirect Addressing
- Implied Addressing

IMMEDIATE ADDRESSING MODE

- In immediate addressing mode the data (8/16 bit) is specified in the instruction itself.
- The immediate addressing instructions are either 2 byte or 3 byte long. In 2 byte instruction first byte is OPCODE and second byte is the 8 bit data. In 3 byte instructions first byte is OPCODE; second and third bytes are the 16 bit data.
- The instructions containing the letter "I" indicate immediate addressing mode.



INSTRUCTION

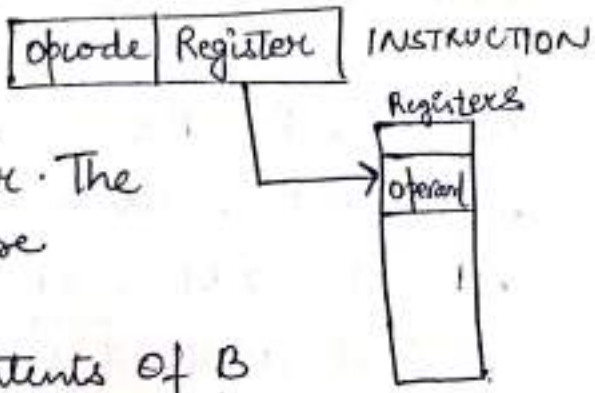
EXAMPLE

(i) MVI A, A0 H:
This instruction transfers immediate data (A0 H) to A register.

(ii) LXI H, C200H :
This instruction transfers 16 bit immediate data C200 to HL register pair. Lower order data (00H) to L register and higher order data (C2H) to H register.

REGISTER ADDRESSING MODE

In register addressing mode the source and destination operands are general purpose registers. The register addressing instructions are generally of 1 byte i.e OPCODE only. The OPCODE specifies the operation and registers to be used to perform the operation.



EXAMPLE

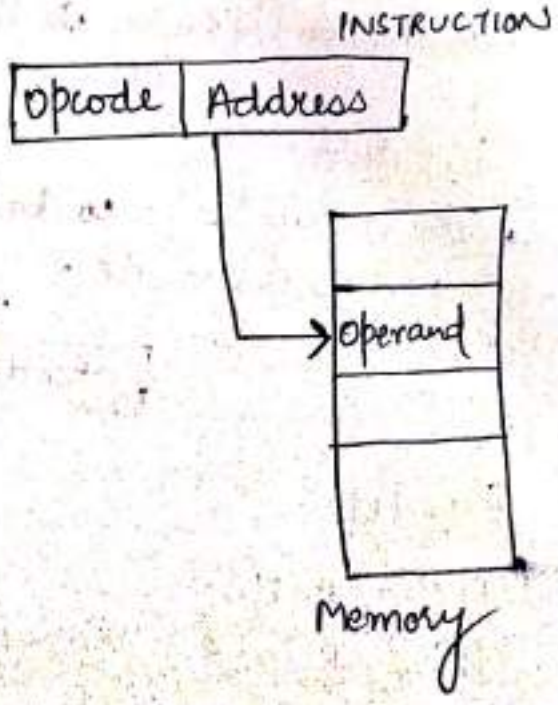
(i) MOV D, B : This instruction copies the contents of register B to the D register. The source and destination operands are both registers.

(ii) ADD B : This instruction adds the contents of B register and A register. The data is present in both B and A registers. The result is stored in the accumulator.

(iii) PCHL : This instruction will transfer the contents of register pair HL to the PC.

DIRECT ADDRESSING MODE

In direct addressing mode the 16 bit address of the operand is given within the instruction itself. The instructions in the direct addressing mode are 3 byte instructions. First byte is a OPCODE, second is lower order address byte and third is higher order address byte.
For I/O instruction that use direct addressing mode are 2 byte as the address of I/O is one byte.



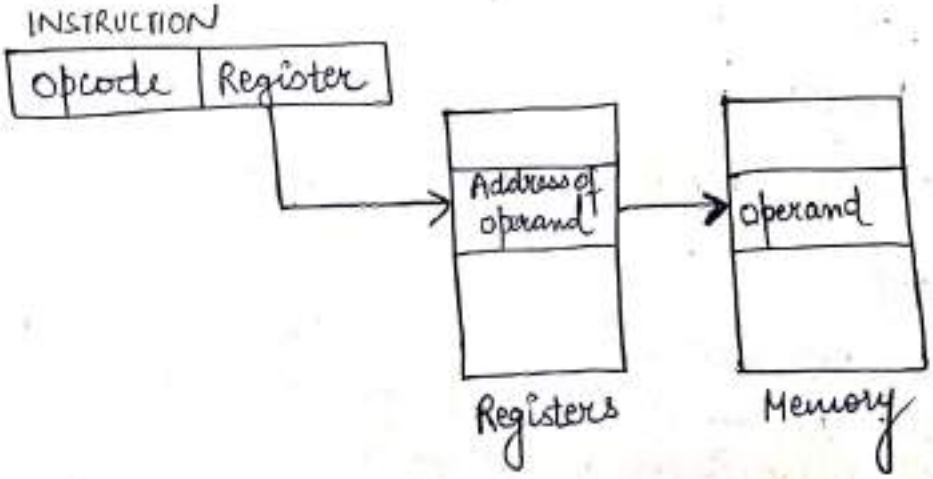
EXAMPLE

(i) LDA C200H : Load accumulator directly from memory location. In this instruction the contents of C200 memory location are transferred to accumulator.

(ii) STA C200H : Store accumulator directly to memory location. In this instruction the contents of accumulator are stored at memory location C200H.

INDIRECT ADDRESSING MODE

In indirect addressing mode the instructions reference the memory through a register pair i.e the memory address where the operand is located is specified by the contents of a register pair.



EXAMPLE

(i) MOV A,M : In this case M is a memory pointer specifying HL register pair where the address is stored. The contents of HL pair are used as address and the contents of that memory location are transferred to accumulator.

(ii) LDAX B : In this case BC register pair is used as address and the contents of the memory location specified by BC register pair are copied to accumulator.

IMPLIED ADDRESSING OR IMPLICIT ADDRESSING MODE

- The implied mode of addressing doesnot require any operand.

- The data is specified within the opcode itself. Generally, the implied addressing mode instruction is a 1 byte instruction
- The data is supposed to be present generally in accumulator.

EXAMPLE

(i) RAL: Rotate accumulator left, it operates on the data in accumulator only. So whenever RAL is used it is implied that the data to be operated on is available in accumulator only.

(ii) CMC: Complement carry flag.

TIMING DIAGRAM

We draw graphical presentation of steps with respect to time i.e clock. The graphical representation is called as the "Timing diagram".

INSTRUCTION CYCLE, MACHINE CYCLE AND TSTATES

INSTRUCTION CYCLE

- The CPU fetches one instruction from the memory at a time and execute it. The essential steps required by CPU to fetch and execute an instruction is called as instruction cycle. An instruction cycle consists of a fetch cycle and an execute cycle.
In a fetch cycle the CPU fetches the opcode of the instruction from the memory. The steps required to acquire the data from the memory and complete the operation that is specified in the instruction is constituted by the execute cycle.
- The total time required to execute an instruction is

$$IC = FC + EC$$

IC: Time required for instruction cycle.
FC: time required for fetch cycle.
EC: time required for execution cycle.

MACHINE CYCLE

- The time required by the microprocessor to complete the operation of accessing memory or I/O device is called a machine cycle. Operations like opcode fetch, memory read; memory write, I/O read, I/O write are performed in machine cycle.
- An instruction cycle thus consists of several machines cycles.
- The microprocessor 8085A consists of seven different types of machine cycles.

MACHINE CYCLE	STATUS			CONTROL		
	$IO/\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	INTA
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read	1	1	0	0	1	1
I/O Write	1	0	1	1	0	1
INTR Acknowledge	1	1	1	1	1	0
Bus Idle	0	0	0	1	1	1

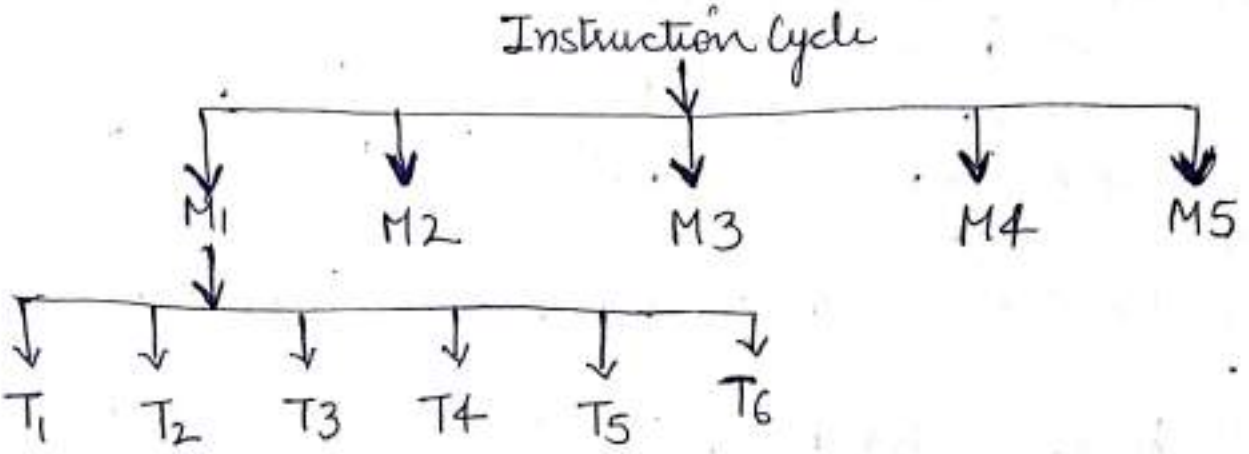
T STATE

Microprocessor performs an operation in a specific time period i.e specific clock cycles. Each clock cycle is called as T state.

The number of T states required to perform an operation is called machine cycles.

Machine cycle consists of 3 to 6 T states and 1T state is of one clock period. An instruction cycle consists of 1 to 5 machine cycles.

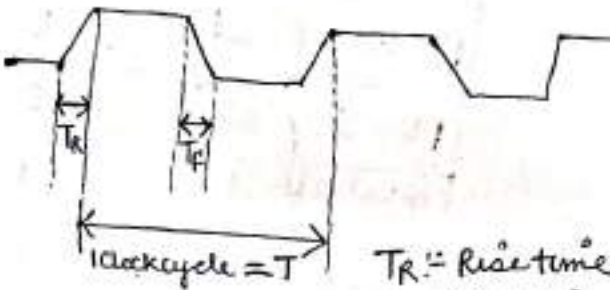
Relation b/w instruction cycle, machine cycle and T State



REPRESENTATION OF SIGNALS IN TIMING DIAGRAM

CLOCK SIGNAL

The microprocessor operates with reference to clock signals provided to it. At pins X_1 and X_2 we provide clock generator and this frequency is divided by 2 and used by the microprocessor. This frequency is called as operating frequency.

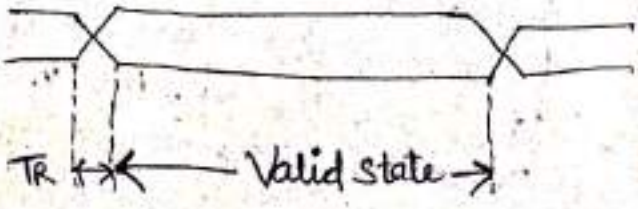


SINGLE LINE SIGNAL

The status of single line will be either LOW or HIGH i.e any one state logic 0 or logic 1. But the change from one state to another is not possible in zero time. It requires some time to go from LOW to HIGH or from HIGH to LOW.

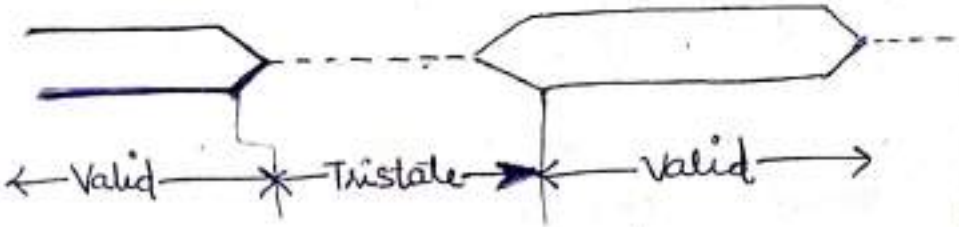
MULTIPLE LINE SIGNAL

These signals are also called as bus signals. Instead of having each line using single line signals, these are grouped and shown in the form of block.



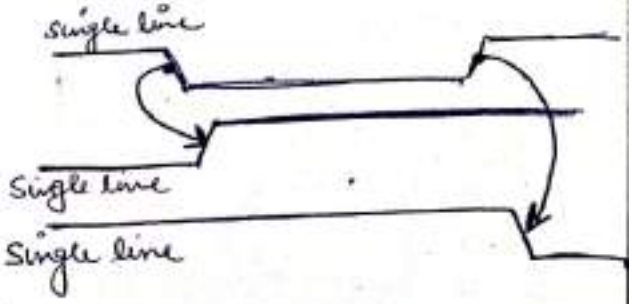
MULTIPLE LINES WITH HIGH IMPEDANCE

If multiple lines are having 3 states viz logic 0, logic 1 and tristate



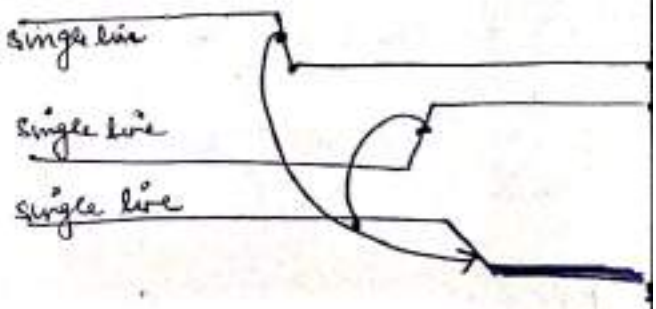
SINGLE LINE AND SINGLE EFFECT SIGNAL

Because of single line the other signal also changes. A change in one single line causes change in other signal.



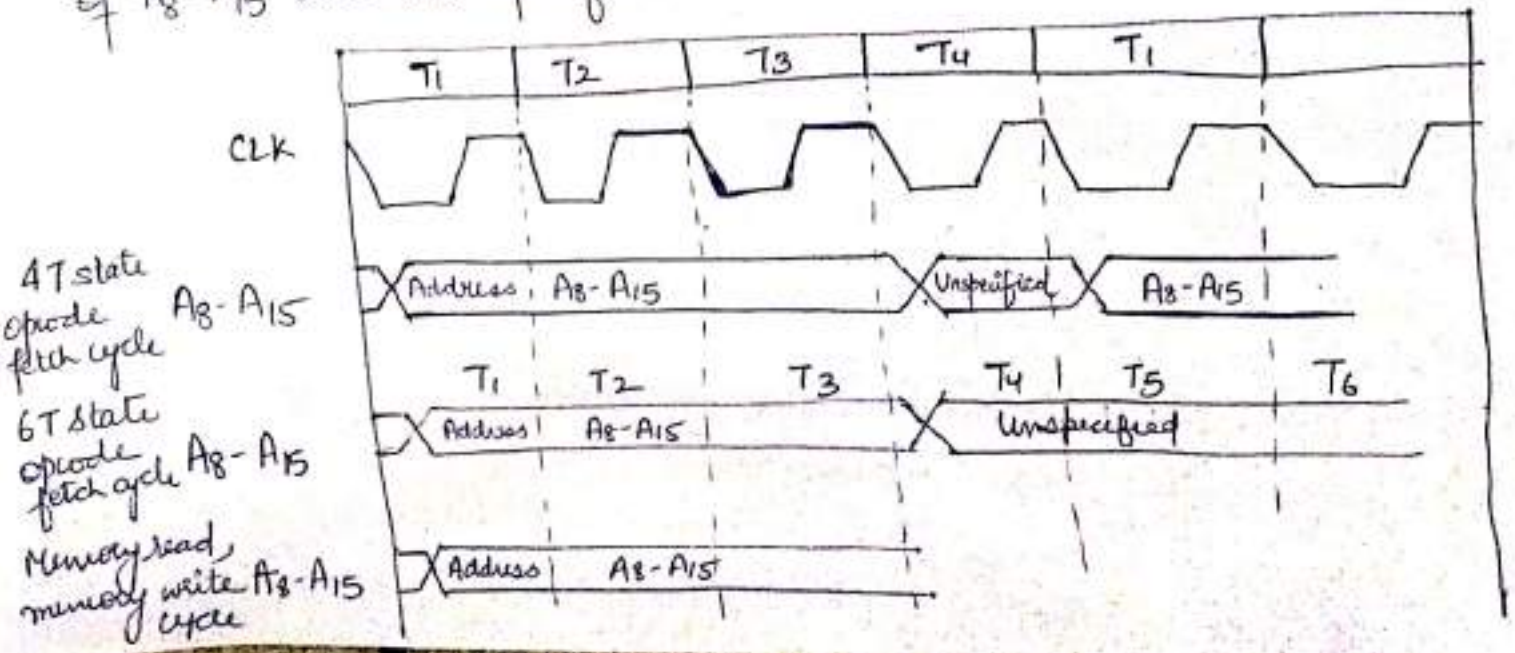
MULTIPLE LINES AND SINGLE EFFECT SIGNAL

If a change in multiple lines causes change in single signal.



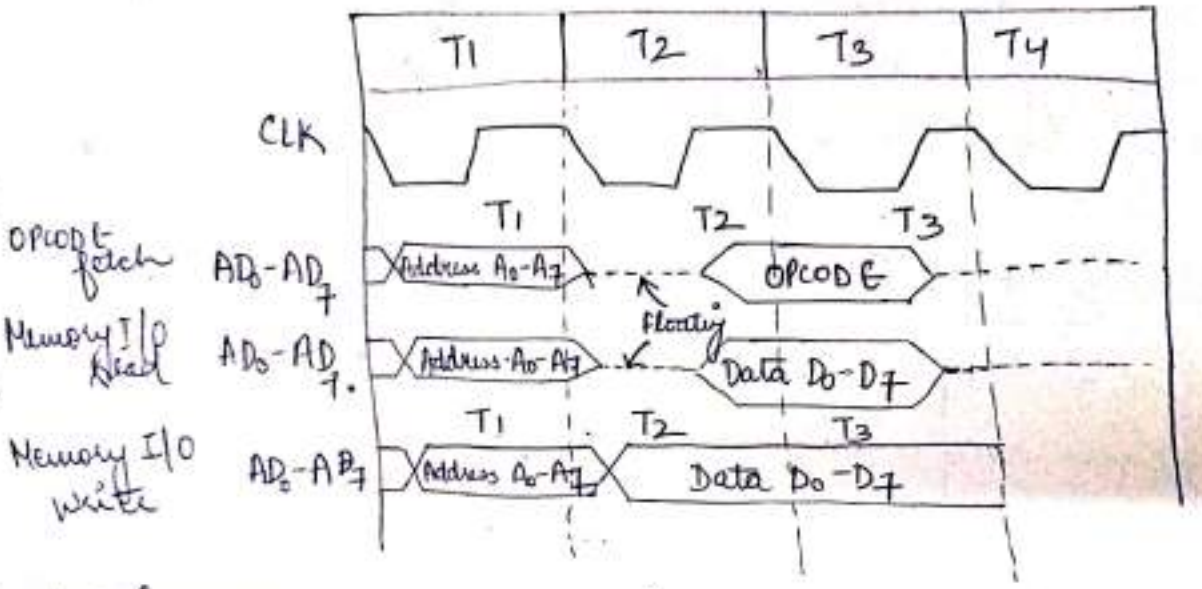
$A_8 - A_{15}$ (HIGHER ORDER ADDRESS)

The higher order address is available on $A_8 - A_{15}$ lines during T_1, T_2 and T_3 state of each machine cycle except bus idle cycle. During T_4, T_5 and T_6 state of OPCODE fetch the contents of $A_8 - A_{15}$ are not specified



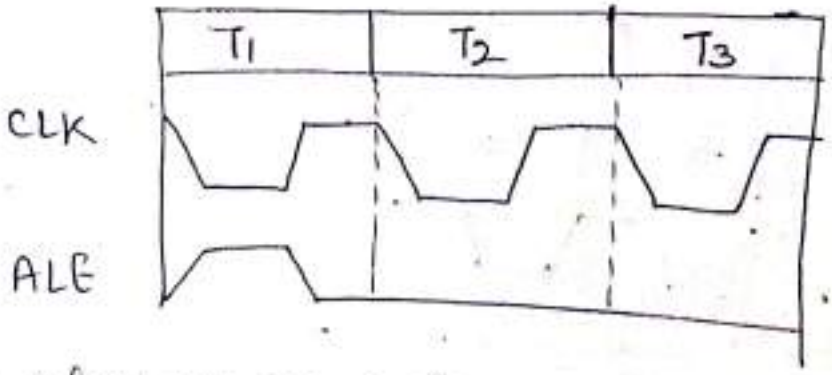
LOWER ORDER ADDRESS / DATA

The lower order address is available on $AD_0 - AD_7$ lines during T_1 state of each machine cycle.
During memory read operation, data from memory or I/O device is transferred on $AD_0 - AD_7$ lines during T_2.
During memory write operation, data from microprocessor is transferred on $AD_0 - AD_7$ lines as soon as the address $AD_0 - AD_7$ is removed.



ALE (ADDRESS LATCH ENABLE)

The ALE signal is available during state T_1 of each m/c cycle.
The ALE is a pulse signal, used to latch the address from $AD_0 - AD_7$

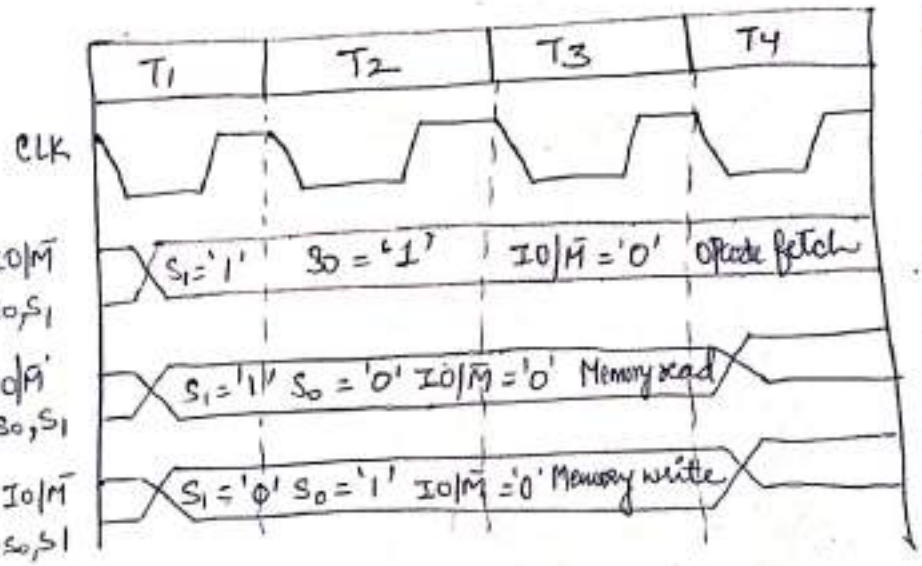


IO/$\overline{M}$, S_0, S_1 (STATUS SIGNALS)

The status signals IO/$\overline{M}$, S_0, S_1 are available during state T_1 to last T state of each machine cycle.

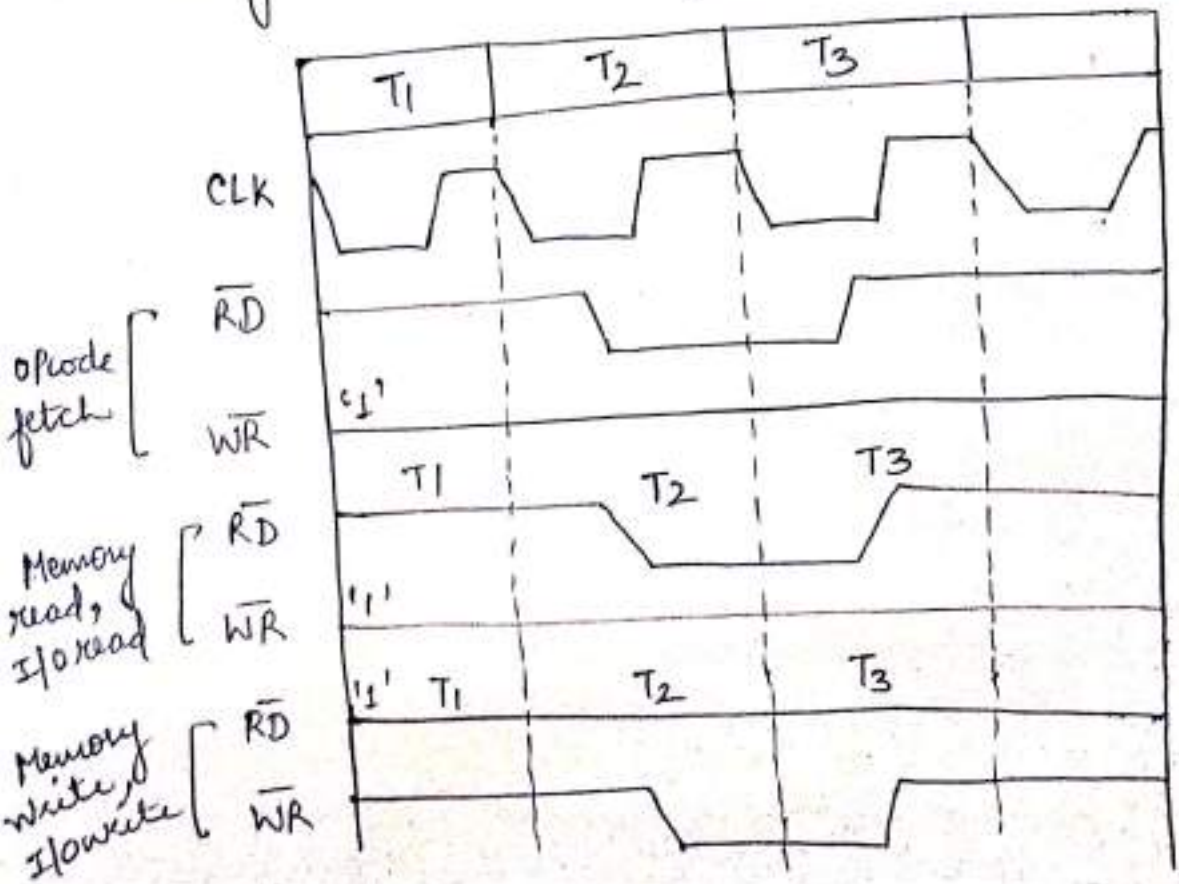
Machine Cycle	IO/$\bar{M}$	S_1	S_0
Opcode fetch	0	1	1
Memory read	0	1	0
Memory write	0	0	1
I/O Read	1	1	0
I/O write	1	0	1
Interrupt acknowledge	1	1	1
Bus ideal	0	0	0

Opcode fetch: IO/$\bar{M}$, S_0, S_1
Memory read: IO/$\bar{M}$, S_0, S_1
Memory write: IO/$\bar{M}$, S_0, S_1



READ AND WRITE CONTROL SIGNALS

- The control signals decide the operation to be performed:
  (i) For memory read, opcode fetch, I/O read the $\overline{RD}$ signal is activated during T_2 and T_3 state of machine cycle.
  (ii) for memory write, I/O write the $\overline{WR}$ signal is activated during T_2 and T_3 state of machine cycle.



CLASSIFICATION OF THE INSTRUCTION SET

The instructions are generally classified into the functional categories as follows:

i) Data transfer group
ii) Arithmetic group
iii) Logical group
iv) Branching group
v) Stack and Machine control group.

Notations used in Instructions

Notations	Meaning
M	Memory location pointed by HL register pair.
r	8 bit register
Rp	16 bit register
R_S	Source Register
R_D	Destination Register
addr	16 bit address
data	8 bit data
data 16 bit	16 bit data