Unit 1: Fundamental of Microprocessor (5 Hrs.)

BCA – 2nd Semester

Prime College

Rolisha Sthapit

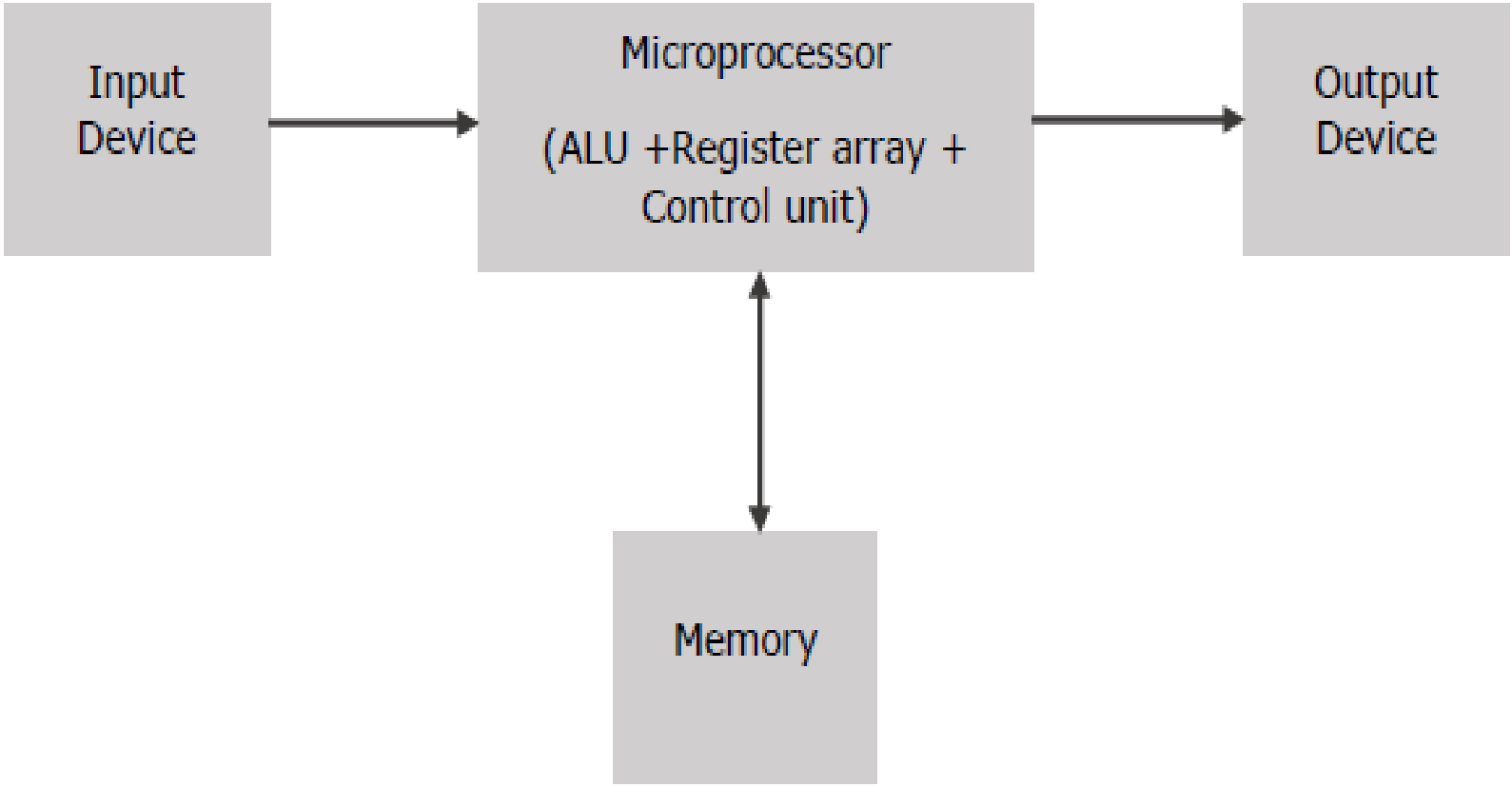
- Introduction to Microprocessors, Microprocessor system with bus organization, Microprocessor architecture and operation, 8085 Microprocessor and its operation, 8085 instruction cycle, machine cycle, T states, Addressing modes in 8085, Introduction to 8086.

What is Microprocessor?

- Microprocessor is a controlling unit of a micro-computer, fabricated on a small chip capable of performing ALU (Arithmetic Logical Unit) operations and communicating with the other devices connected to it.
- A microprocessor is a multipurpose programmable, clock driven, register based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input, processes data according to those instructions and provide results as output.
- In other words, a microprocessor is a clock driven semiconductor device consisting of electronic circuits.

- Microprocessor is capable of performing various computing functions and making decisions to change the sequence of program execution.
- Microprocessor consists of an ALU, register array, and a control unit. ALU performs arithmetical and logical operations on the data received from the memory or an input device. Register array consists of registers identified by letters like B, C, D, E, H, L and accumulator (register in which intermediate arithmetic and logic results are stored.). The control unit controls the flow of data and instructions within the computer.

Block Diagram of a Basic Microcomputer



Terms Used

CPU: Central processing unit which consists of ALU and control unit.

Microprocessor: Single chip containing all units of CPU.

Microcomputer: Computer having microprocessor as CPU.

Microcontroller: Single chip consisting of MPU, memory, I/O and interfacing circuit.

MPU: Micro processing unit – complete processing unit with the necessary control signals.

How does a Microprocessor Work?

- The microprocessor follows a sequence: Fetch, Decode, and then Execute.
- Initially, the instructions are stored in the memory in a sequential order.
- The microprocessor fetches those instructions from the memory, then decodes it and executes those instructions till STOP instruction is reached.
- Later, it sends the result in binary to the output port.
- Between these processes, the register stores the temporarily data and ALU performs the computing functions.

Features of a Microprocessor

a) Cost-effective − The microprocessor chips are available at low prices and results its low cost.

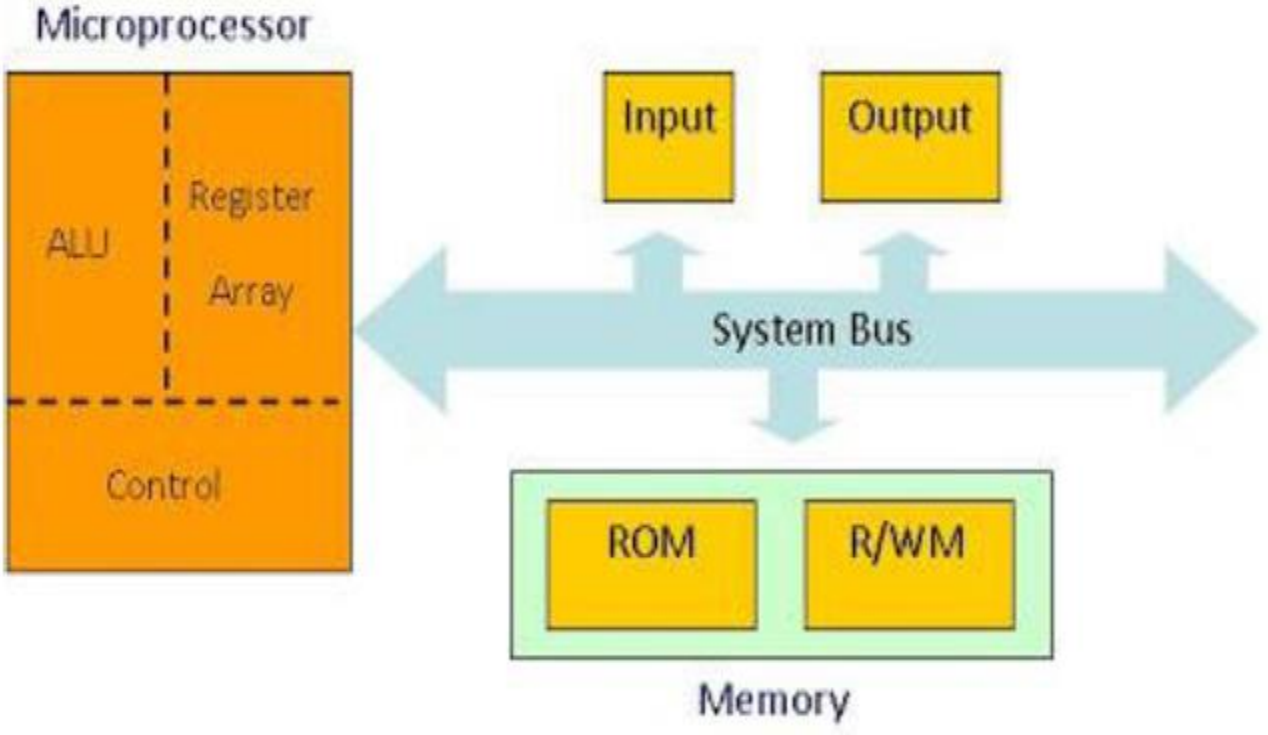
b) Size − The microprocessor is of small size chip, hence is portable.

c) Low Power Consumption − Microprocessors are manufactured by using metal oxide semiconductor technology, which has low power consumption.

d) Versatility − The microprocessors are versatile as we can use the same chip in a number of applications by configuring the software program.

e) Reliability − The failure rate of microprocessors is very low, hence it is reliable.

Microprocessor Architecture & Operation



a) Arithmetic /Logic Unit:

ALU performs arithmetic and logical operations on the data received from the memory or an input device. The arithmetic operations like addition and subtraction and logical operations like AND, OR and XOR.

b) Register Array:

The registers are primarily used to store data temporarily during the execution of a program and are accessible to the user through instruction. Register array consists of registers identified by letters like B, C, D, E, H, L, IX and accumulator (register in which intermediate arithmetic and logic results are stored.).

c) Control Unit:

It provides the necessary timing and control signals to all the operations in the microcomputer. It controls the flow of data between the microprocessor and memory and peripherals.

d) Input:

The input section transfers data and instructions in binary from the outside world to the microprocessor. It includes such devices as a keyboard, switches, a scanner, and an analog-to-digital converter.

f) Output :

The output section transfers data from the microprocessor to such output devices as LED, CRT, printer, magnetic tape, or another computer.

g) Memory:

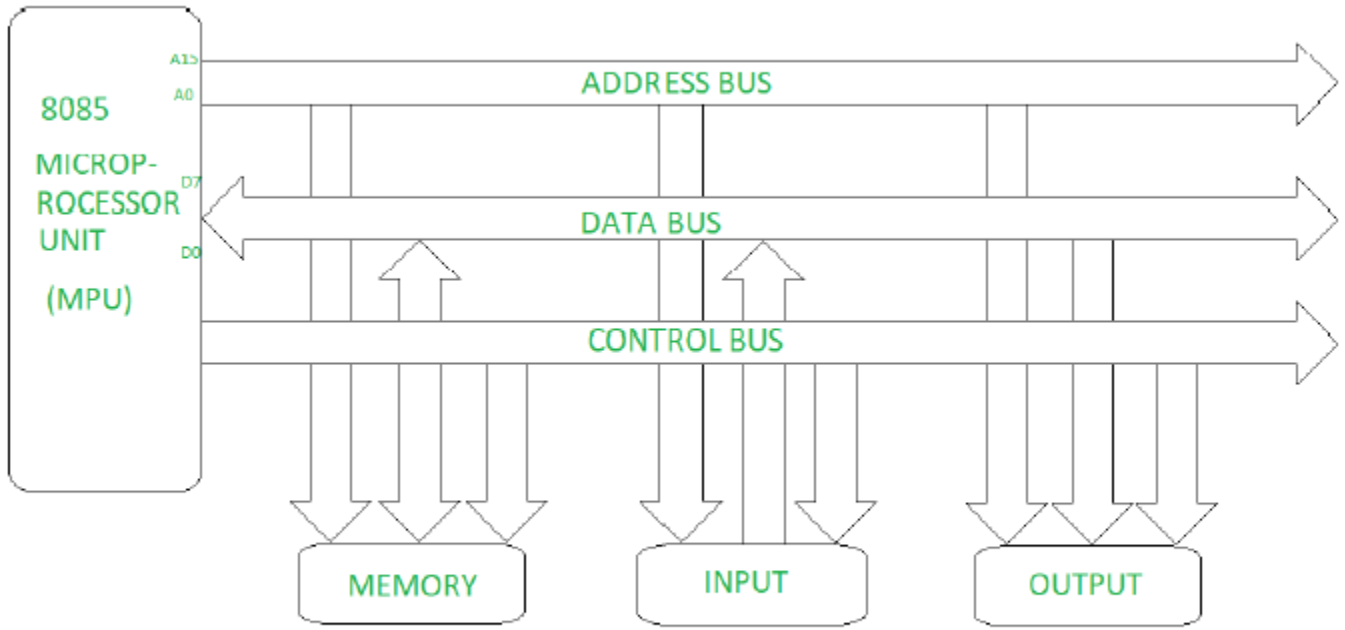
It stores such binary information as instructions and data, and provides that information to the microprocessor. To execute programs, the microprocessor reads instructions and data from memory and performs the computing operations in its ALU section. Results are either transferred to the output section for display or stored in memory for later use.

h) System bus:

It is a communication path between the microprocessor and peripherals. The microprocessor communicates with only one peripheral at a time. The timing is provided by the control unit of the microprocessor.

Bus organization of 8085 microprocessor

Bus is a group of conducting wires which carries information, all the peripherals are connected to microprocessor through bus.



Bus organization system of 8085 Microprocessor

There are three types of buses.

a) Address bus :

- It is a group of conducting wires which carries address only.
- Address bus is unidirectional because data flow in one direction, from microprocessor to memory or from microprocessor to Input/output devices (That is, Out of Microprocessor).
- Length of address bus of 8085 microprocessor is 16 bit.

b) Data bus :

- It is a group of conducting wires which carries Data only.
- Data bus is bidirectional because data flow in both directions, from microprocessor to memory or Input/output devices and from memory or Input/output devices to microprocessor.
- Length of data bus of 8085 microprocessor is 8 bit.

c) Control bus –

It is a group of wires, which is used to generate timing and control signals to control all the associated peripherals, microprocessor uses control bus to process data, that is what to do with selected memory location. Some control signals are:

- Memory read
- Memory write
- I/O read
- I/O Write

Applications

- The microprocessor is used in personal computers (PCs).
- The microprocessor is used in LASER printers for good speed and making automatic photo copies.
- The microprocessors are used in modems, telephone, digital telephone sets, and also in air reservation systems and railway reservation systems.
- The microprocessor is used in medical instrument to measure temperature and blood pressure.
- It is also used in mobile phones and television.
- It is used in calculators and game machine.
- It is used in accounting system and data acquisition system.
- It is used in military applications.
- It is also used in traffic light control.
- Microprocessor is used in home appliances such as microwave ovens, washing machine etc.

Microprocessor Vs Microcontroller

Microprocessor	Microcontroller
CPU is stand alone, RAM,ROM, I/O & timer are separate.	CPU, RAM,ROM, I/O & timer all are on single chip.
Designer can decide amount of RAM,ROM, & I/O ports.	Fixed amount of on-chip RAM,ROM, & I/O ports.
High processing power	Low processing power
High power consumption	Low power consumption
Typically 32/64 bit	8/16 bit
General purpose	Single purpose(control oriented)
Less reliable	Highly reliable
Eg.- 8086,8085	8051

8085 Microprocessor & its Operation

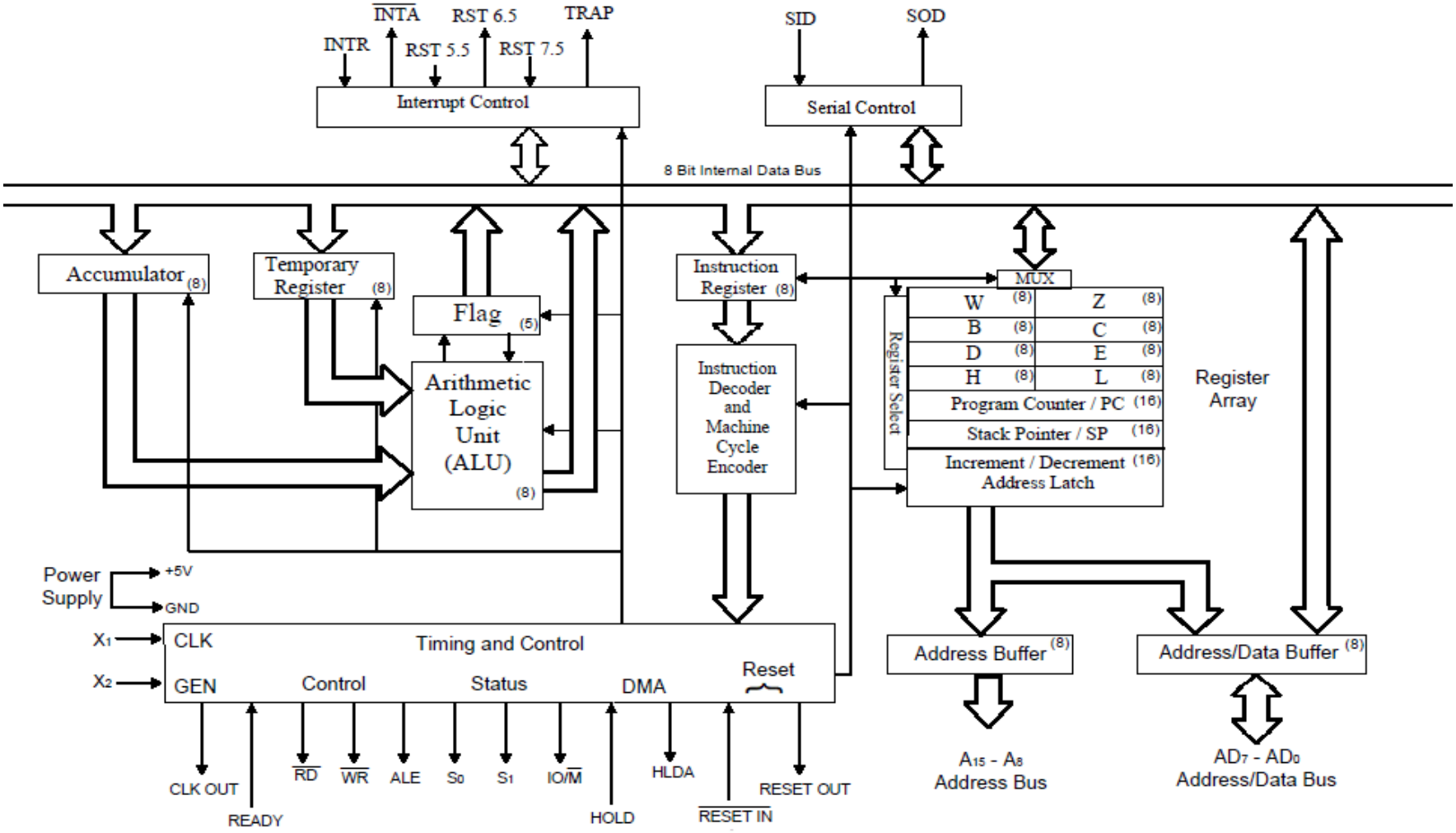
- 8085 is pronounced as "eighty-eighty-five" microprocessor.
- It is an 8-bit microprocessor designed by Intel in 1977 using NMOS technology (**N-type metal-oxide-semiconductor logic**)
- It is a complete 8 bit parallel central processing unit.
- The main components of 8085 are array of registers, the ALU, encoder/decoder, and timing and control circuits linked by an internal data bus.

It has the following configuration –

- 8-bit data bus
- 16-bit address bus, which can address up to 64KB
- A 16-bit program counter
- A 16-bit stack pointer
- Six 8-bit registers arranged in pairs: BC, DE, HL
- Requires +5V supply to operate at 3.2 MHZ single phase clock

- It is used in washing machines, microwave ovens, mobile phones, etc

8085 Microprocessor Architecture & Functional Units



8085 consists of the following functional units −

1) Accumulator

It is an 8-bit register used to perform arithmetic, logical, I/O & LOAD/STORE operations. It is connected to internal data bus & ALU.

2) Arithmetic and logic unit

As the name suggests, it performs arithmetic and logical operations like Addition, Subtraction, AND, OR, etc. on 8-bit data.

3) General purpose register

There are 6 general purpose registers in 8085 processor, i.e. B, C, D, E, H & L. Each register can hold 8-bit data. These registers can work in pair to hold 16-bit data and their pairing combination is like B-C, D-E & H-L.

4) Program counter

It is a 16-bit register used to store the memory address location of the next instruction to be executed. Microprocessor increments the program whenever an instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

5) Stack pointer

It is also a 16-bit register works like stack, which is always incremented/decremented by 2 during push & pop operations.

6) Temporary register

It is an 8-bit register, which holds the temporary data of arithmetic and logical operations.

7) Flag register

It is an 8-bit register having five 1-bit flip-flops, which holds either 0 or 1 depending upon the result stored in the accumulator.

These are the set of 5 flip-flops –

- Sign (S)
- Zero (Z)
- Auxiliary Carry (AC)
- Parity (P)
- Carry (C)

Its bit position is shown in the following table –

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

a) Sign Flag (S) – After any operation if result is negative sign flag becomes set, i.e. If result is positive sign flag becomes reset i.e. 0.

Example:

MVI A 30 (load 30H in register A)

MVI B 40 (load 40H in register B)

SUB B (A = A – B)

These set of instructions will set the sign flag to 1 as 30 – 40 is a negative number.

MVI A 40 (load 40H in register A)

MVI B 30 (load 30H in register B)

SUB B (A = A – B)

These set of instructions will reset the sign flag to 0 as 40 – 30 is a positive number.

b) Zero Flag (Z) – After any arithmetical or logical operation if the result is 0 (00)H, the zero flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.

Example:

MVI A 10 (load 10H in register A)

SUB A (A = A – A)

These set of instructions will set the zero flag to 1 as 10H – 10H is 00H

c) Auxiliary Carry Flag (AC) – If intermediate carry is generated this flag is set to 1, otherwise it is reset to 0.

•Example:

MOV A 2B (load 2BH in register A)

MOV B 39 (load 39H in register B)

ADD B (A = A + B)

These set of instructions will set the auxiliary carry flag to 1, as on adding 2B and 39, addition of lower order nibbles B and 9 will generate a carry.

d) **Parity Flag (P) –** If after any arithmetic or logical operation the result has even parity, an even number of 1 bits, the parity register becomes set i.e. 1, otherwise it becomes reset.

1-accumulator has even number of 1 bits

0-accumulator has odd parity

e) **Carry Flag (CY) –** Carry is generated when performing n bit operations and the result is more than n bits, then this flag becomes set i.e. 1, otherwise it becomes reset i.e. 0.

During subtraction (A-B), if A>B it becomes reset and if (A<B) it becomes set.

Carry flag is also called borrow flag.

8) Instruction register and decoder

It is an 8-bit register. When an instruction is fetched from memory then it is stored in the Instruction register. Instruction decoder decodes the information present in the Instruction register.

9) Timing and control unit

It provides timing and control signal to the microprocessor to perform operations. Following are the timing and control signals-

- Control Signals: READY, RD', WR'
- Status Signals: S0, S1, IO/M'
- DMA Signals: HOLD, HLDA
- RESET Signals: RESET IN, RESET OUT

[S0 and S1 are used to observe the state, HOLD is used by peripheral device to request permission for accessing the system bus, HLDA is used to grant or deny access to its system bus, IO/M' is to distinguish between I/O and memory port.

10) Interrupt control

As the name suggests it controls the interrupts during a process. When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program.

There are 5 interrupt signals in 8085 microprocessor: INTR, RST 7.5, RST 6.5, RST 5.5, TRAP. When microprocessor receives interrupt signals, it sends an acknowledgement (INTA) to the peripheral which is requesting for its service.

Maskable and Non-Maskable Interrupts

 Maskable Interrupts are those which can be disabled or ignored by the microprocessor. INTR, RST 7.5, RST 6.5, RST 5.5 are maskable interrupts in 8085 microprocessor.

 Non-Maskable Interrupts are those which cannot be disabled or ignored by microprocessor. TRAP is a non-maskable interrupt.

11) Serial Input/output control

It controls the serial data communication by using these two instructions: SID (Serial input data) and SOD (Serial output data).

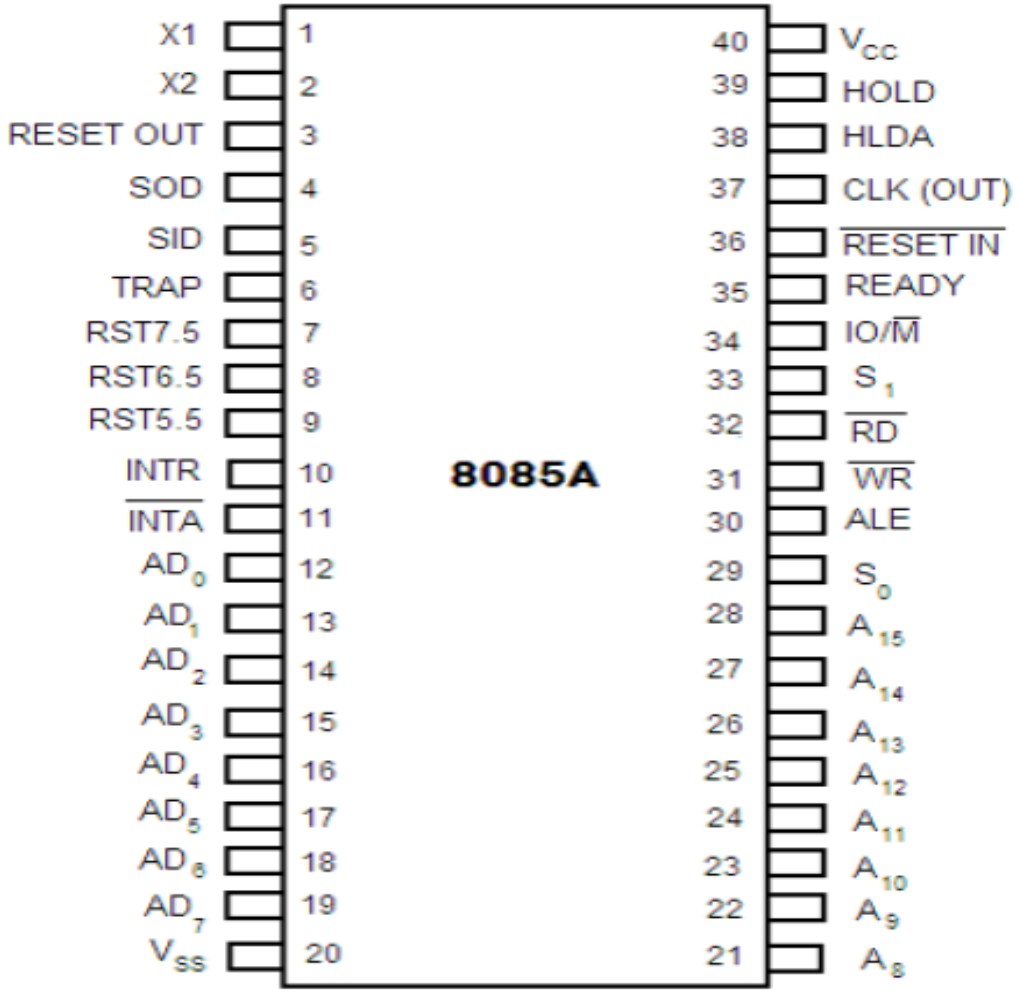
12) Address buffer and address-data buffer

The content stored in the stack pointer and program counter is loaded into the address buffer and address-data buffer to communicate with the CPU. The memory and I/O chips are connected to these buses; the CPU can exchange the desired data with the memory and I/O chips.

13) Address bus and data bus

Data bus carries the data to be stored. It is bidirectional, whereas address bus carries the location to where it should be stored and it is unidirectional. It is used to transfer the data & Address I/O devices.

8085 Pin Configuration



The pins of a 8085 microprocessor can be classified into seven groups:

1) Address bus

A15-A8, it carries the most significant 8-bits of memory/IO address.

2) Data bus

AD7-AD0, it carries the least significant 8-bit address and data bus.

3) Control and status signals

These signals are used to identify the nature of operation. There are 3 control signal and 3 status signals.

Three control signals are RD, WR & ALE.

a) **RD** − This signal indicates that the selected IO or memory device is to be read and is ready for accepting data available on the data bus.

b) WR − This signal indicates that the data on the data bus is to be written into a selected memory or IO location.

c) ALE (Address Latch Enable) − It is a positive going pulse generated when a new operation is started by the microprocessor. When the pulse goes high, it indicates address. When the pulse goes down it indicates data.

Three status signals are IO/M, S0 & S1.

IO/M

This signal is used to differentiate between IO and Memory operations, i.e. when it is high indicates IO operation and when it is low then it indicates memory operation.

S1 & S0

These signals are used to identify the type of current operation.

IO/M (Active Low)	S1	S2	Data Bus Status (Output)
0	0	0	Halt
0	0	1	Memory WRITE
0	1	0	Memory READ
1	0	1	IO WRITE
1	1	0	IO READ
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

4) Power supply

There are 2 power supply signals: VCC & VSS. VCC indicates +5v power supply and VSS indicates ground signal.

5) Clock signals

There are 3 clock signals, i.e. X1, X2, CLK OUT.

- **X1, X2** − A crystal (RC, LC N/W) is connected at these two pins and is used to set frequency of the internal clock generator. This frequency is internally divided by 2.
- **CLK OUT** − This signal is used as the system clock for devices connected with the microprocessor.

6) Interrupts & externally initiated signals

Interrupts are the signals generated by external devices to request the microprocessor to perform a task. There are 5 interrupt signals, i.e. TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR.

7) Serial I/O signals

There are 2 serial signals, i.e. SID (Serial input data line) and SOD (Serial output data line) and these signals are used for serial communication.

Addressing Modes in 8085

To perform any operation, we have to give the corresponding instructions to the microprocessor. In each instruction, programmer has to specify 3 things:

1. Operation to be performed.
2. Address of source of data.
3. Address of destination of result.

The method by which the address of source of data or the address of destination of result is given in the instruction is called Addressing Modes. The term addressing mode refers to the way in which the operand of the instruction is specified.

Intel 8085 uses the following addressing modes:

1) Direct Addressing Mode
2) Register Addressing Mode
3) Register Indirect Addressing Mode
4) Immediate Addressing Mode
5) Implicit Addressing Mode

Direct Addressing Mode

In direct addressing mode, the data to be operated is available inside a memory location and that memory location is directly specified as an operand. The operand is directly available in the instruction itself.

Example:

LDA 2050 (load the contents of memory location into accumulator A)

Register Addressing Mode

In register addressing mode, the data to be operated is available inside the register(s) and register(s) is(are) operands. Therefore, the operation is performed within various registers of the microprocessor.

Examples:

MOV A, B (move the contents of register B to register A)

ADD B (add contents of registers A and B and store the result in register A)

INR A (increment the contents of register A by one)

Register Indirect Addressing Mode

In register indirect addressing mode, the data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair.

Example:

MOV A, M (move the contents of the memory location pointed by the H-L pair to the accumulator)

Immediate Addressing Mode

In immediate addressing mode the source operand is always data. If the data is 8-bit, then the instruction will be of 2 bytes, if the data is of 16-bit then the instruction will be of 3 bytes.

Example:

MVI B,45 (move the data 45H immediately to register B)

Implied/Implicit Addressing Mode

In implied/implicit addressing mode the operand is hidden and the data to be operated is available in the instruction itself.

Examples:

RRC (rotate accumulator A right by one bit)

RLC (rotate accumulator A left by one bit)

Instruction cycle, Machine Cycle & T State in 8085 microprocessor

Time required to execute and fetch an entire instruction is **called instruction cycle.**

It consists:

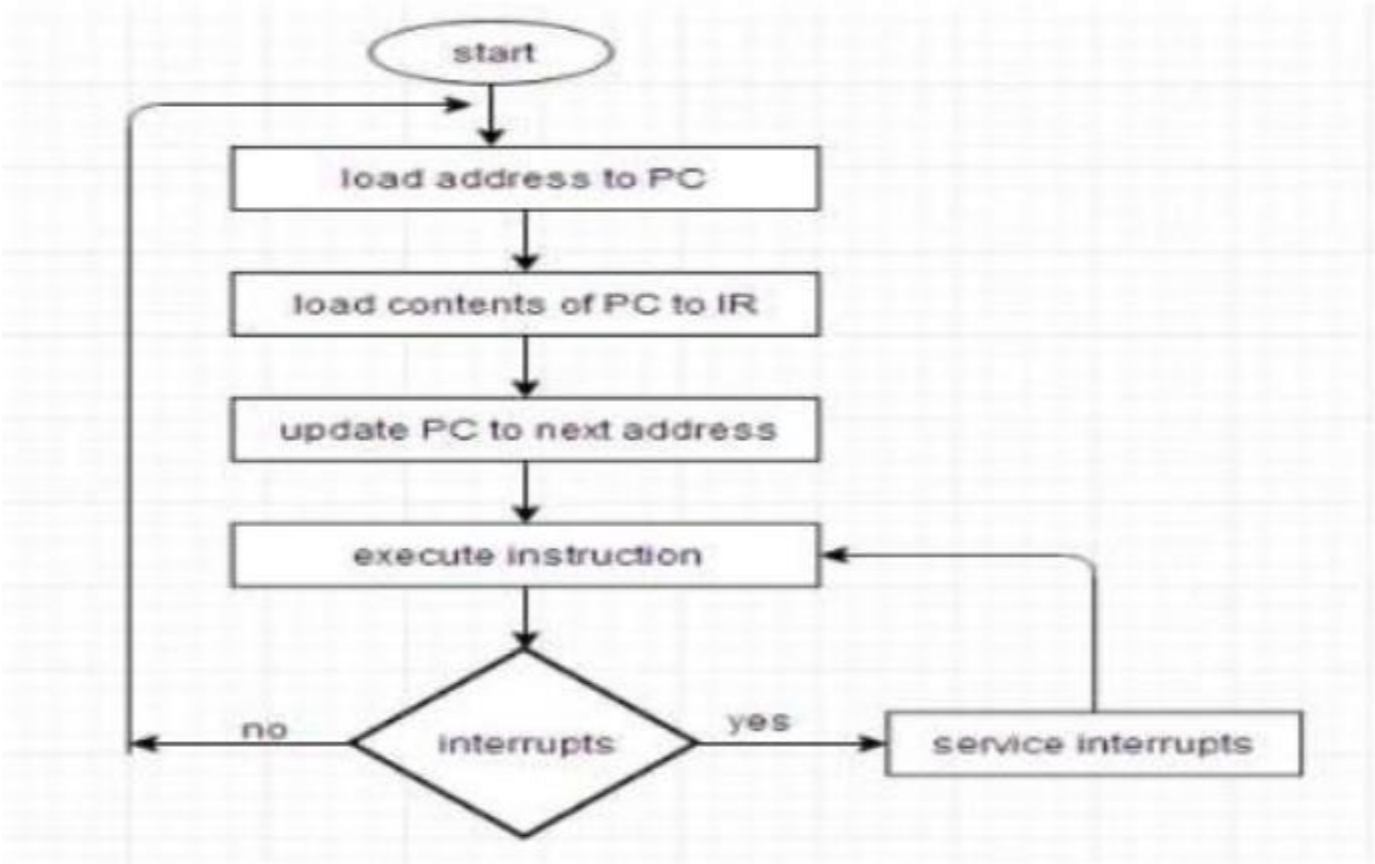
a) **Fetch cycle** – The next instruction is fetched by the address stored in program counter (PC) and then stored in the instruction register.

b) **Decode instruction** – Decoder interprets the encoded instruction from instruction register.

c) **Reading effective address** – The address given in instruction is read from main memory and required data is fetched. The effective address depends on direct addressing mode or indirect addressing mode.

d) **Execution cycle** – consists memory read (MR), memory write (MW), input output read (IOR) and input output write (IOW).

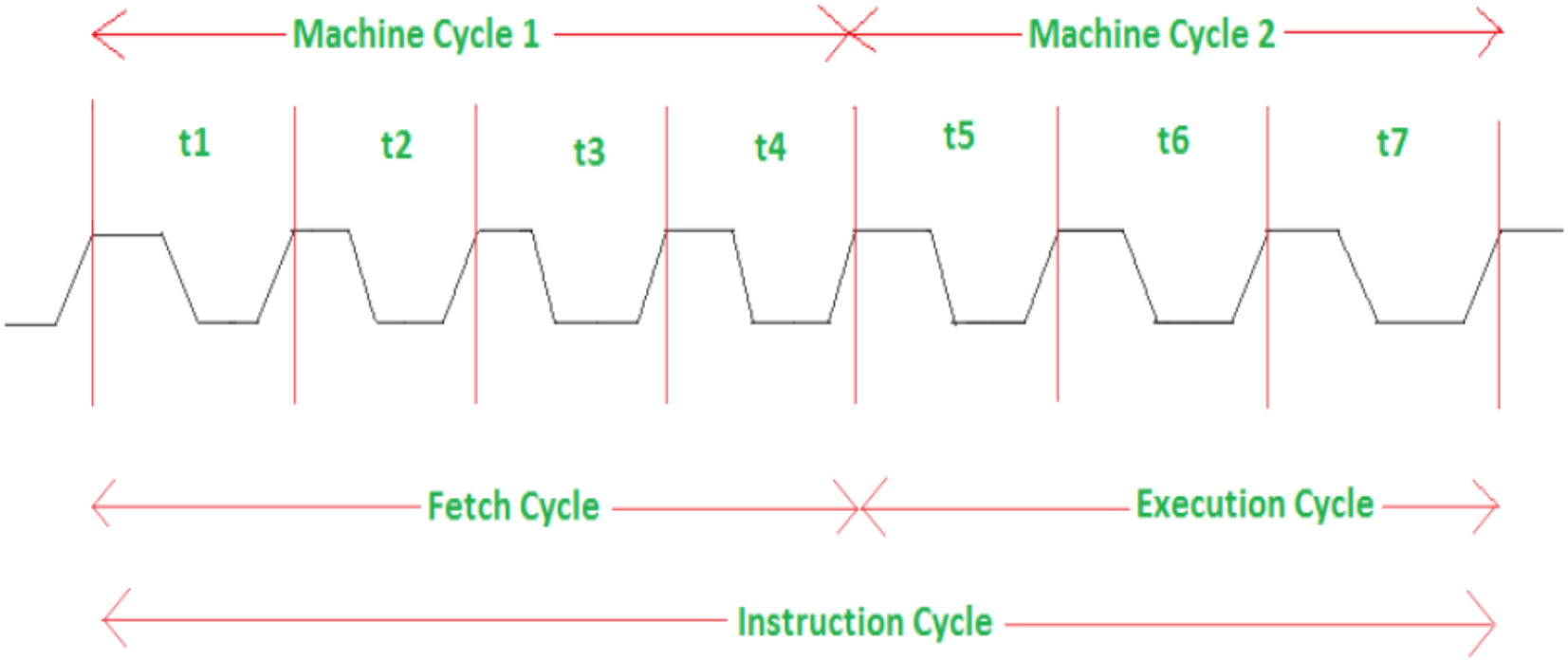
General Flowchart of Instruction Cycle



Upon the completion of step 4, the control goes back to step 1 to fetch, decode, and execute the next instruction. This process continues indefinitely unless a HALT instruction is encountered.

The time required by the microprocessor to complete an operation of accessing memory or input/output devices is called machine cycle. Following are the different types of machine cycle:

- Opcode fetch, which takes 4 t-states
- Memory Read, which takes 3 t-states
- Memory Write, which takes 3 t-states
- I/O Read, which takes 3 t-states
- I/O Write, which takes 3 t-states


Machine Cycle 1
Machine Cycle 2
t1
t2
t3
t4
t5
t6
t7
Fetch Cycle
Execution Cycle
Instruction Cycle

- The time required by the microprocessor to complete an operation of accessing memory or input/output devices is called **machine cycle**.
- One time period of frequency of microprocessor is called **t-state.** A t-state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse.
- Fetch cycle takes four t-states and execution cycle takes three t-states.

Introduction to 8086

- 8086 Microprocessor is an enhanced version of 8085Microprocessor that was designed by Intel.
- It is a 16-bit Microprocessor having 20 address lines and16 data lines that provides up to 1MB storage.
- It consists of powerful instruction set, which provides operations like multiplication and division easily.
- It supports two modes of operation, i.e. Maximum mode and Minimum mode. Maximum mode is suitable for system having multiple processors and Minimum mode is suitable for system having a single processor.

Features of 8086

a) It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.

b) It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.

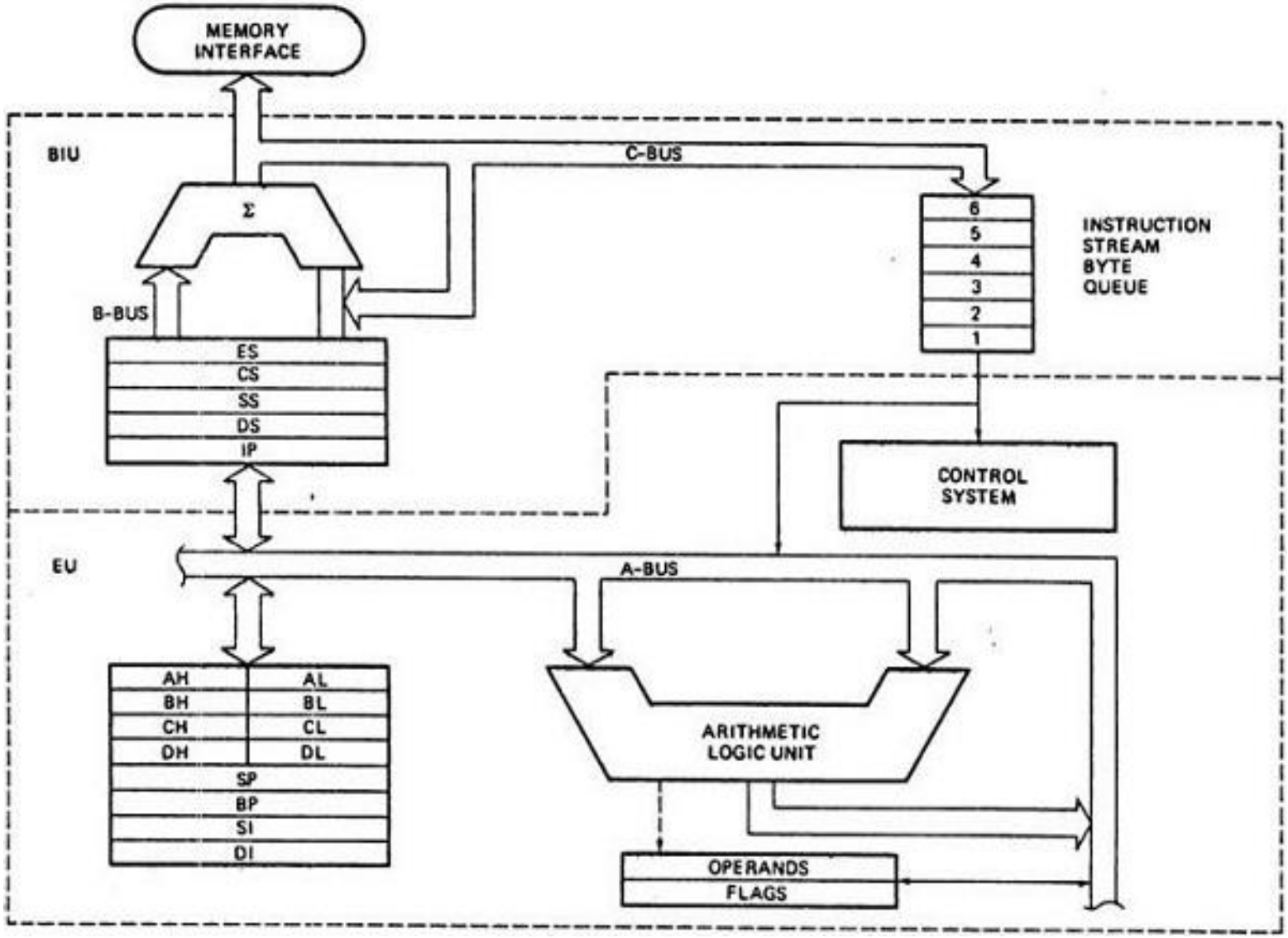
c) It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance. **Fetch stage** can prefetched up to 6 bytes of instructions and stores them in the queue. **Execute stage** executes these instructions.

d) It consists of 29,000 transistors.

Comparison

8085 Microprocessor	8086 Microprocessor
It is an 8 bit microprocessor.	It is a 16 bit microprocessor.
It has 16 bit address line.	It has 20 bit address line.
It has 8- bit data bus.	It has 16- bit data bus.
The memory capacity is 64 KB.	The memory capacity is 1 MB.
Clock speed of this microprocessor is 3 MHz.	Clock speed of this microprocessor varies between 5, 8 and 10 MHz for different versions.
It has 5 flags.	It has 9 flags.
8085 microprocessor does not support memory segmentation.	8086 microprocessor supports memory segmentation.
It does not support pipelining.	It supports pipelining.
It is accumulator based processor.	It is general purpose register based processor.
It has no minimum or maximum mode.	It has minimum and maximum modes.
In 8085, only one processor is used.	In 8086, more than one processor is used. Additional external processor can also be employed.

Internal Architecture of 8086



- The 8086 CPU is divided into two independent functional parts : BIU (Bus Interface Unit) and EU (Execution Unit)
- Dividing the work between these units speeds up the processing.
- The BIU send out address, fetches instructions from memory, reads data from ports and memory, and writes data to ports and memory.
- In other words BIU handles all transfers of data and addresses on the buses for the execution unit.
- The EU of the 8086 tells the BIU where to fetch the instructions and data from, decodes instructions and executes instructions.

THE EXECUTION UNIT

- The EU contains control circuitry which directs internal operations.
- Decoder in EU translates instructions fetched from memory into a series of actions which the EU carries out.
- The EU has a 16 bit ALU which can add subtract, AND, OR, increment, decrement, complement or shift binary numbers.

1. GENERAL PURPOSE REGISTERS

• The EU has eight general purpose registers, labeled AH, AL, BH, BL, CH, CL, DH and DL.

• These registers can be used individually for temporary storage of 8 bit data.

• The AL register is also called accumulator

• It has some features that the other general purpose registers do not have.

• Certain pairs of these general purpose registers can be used together to store 16 bit words.

The acceptable register pairs are AH and AL,BH and BL,CH and CL,DH and DL

• The AH-AL pair is referred to as the AX register, the BH-BL pair is referred to as the BX register, the CH-CL pair is referred to as the CX register, and the DH-DL pair is referred to as the DX register.

- AX = Accumulator Register
- BX = Base Register
- CX = Count Register
- DX = Data Register

2. FLAG REGISTER

• A Flag is a flip-flop which indicates some condition produced by the execution of an instruction or controls certain operations of the EU.

• A 16 bit flag register in the EU contains 9 active flags.

• Figure below shows the location of the nine flags in the flag register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

U: Undefined

Figure: 8086 Flag Register Format

- U = UNDEFINED
- CONDITIONAL FLAGS
    - CF = CARRY FLAG [Set by Carry out of MSB]
    - PF = PARITY FLAG [Set if Result has even parity]
    - AF= AUXILIARY CARRY FLAG FOR BCD
    - ZF = ZERO FLAG [Set if Result is 0]
    - SF = SIGN FLAG [MSB of Result]
    - OF = OVERFLOW FLAG
- CONTROL FLAG
    - TF = SINGLE STEP TRAPFLAG
    - IF = INTERRUPT ENABLE FLAG
    - DF = STRING DIRECTION FLAG

- The six conditional flags in this group are the CF,PF,AF,ZF,SF and OF
- The three remaining flags in the Flag Register are used to control certain operations of the processor.
- The six conditional flags are set or reset by the EU on the basis of the result of some arithmetic or logic operation.
- The Control Flags are deliberately set or reset with specific instructions you put in your program.
- The three control flags are the TF,IF and DF.
- Trap Flag is used for single stepping through a program. If this flag is set, the processor enters the single step execution.
- The Interrupt Flag is used to allow or prohibit the interruption of a program. If this flag is set, the markable interrupts are recognized by the CPU, otherwise they are ignored.
- The Direction Flag is used with string instructions. If this flag bit is 0,the string is processed from the lowest to the highest address i.e., auto incrementing mode. Otherwise, the string is processed from highest address to lowest address, i.e., auto decrementing mode.

3. POINTER REGISTERS

• The 16 bit Pointer Registers are IP,SP and BP respectively

• SP and BP are located in EU whereas IP is located in BIU

3.1 STACK POINTER (SP)

• The 16 bit SP Register provides an offset value, which when associated with the SS register (SS:SP)

3.2 BASE POINTER (BP)

• The 16 bit BP facilitates referencing parameters, which are data and addresses that a program passes via the stack.

• The processor combines the addresses in SS with the offset in BP.

• BP can also be combined with DI and SI as a base register for special addressing.

Note: Both SP and BP are the offsets for Stack Register (SS). The address calculated when BP is taken as the offset gives the starting address of the stack. The address when SP is taken as the offset denotes the memory location where the top of the stack lies

4. INDEX REGISTERS

• The 16 bit Index Registers are SI and DI

4.1 SOURCE INDEX (SI) REGISTER

• The 16 bit Source Index Register is required for some string handling operations

• SI is associated with the DS Register.

4.2 DESTINATION INDEX (DI) REGISTER

• The 16 bit Destination Index Register is also required for some string operations.

• In this context, DI is associated with the ES register.

THE BUS INTERFACE UNIT

1. SEGMENT REGISTERS

1.1 CS REGISTER (Code Segment)

• It contains the starting address of a program's code segment.

• This segment address plus an offset value in the IP register indicates the address of an instruction to be fetched for execution

• For normal programming purpose, you need not directly reference this register.

1.2 DS REGISTER (Data Segment)

• It contains the starting address of a program's data segment

• Instruction uses this address to locate data.

• This address plus an offset value in an instruction causes a reference to a specific byte location in the data segment.

1.3 SS REGISTER (Stack Segment)

- Permits the implementation of a stack in memory
- It stores the starting address of a program's stack segment the SS register.
- This segment address plus an offset value in the Stack Pointer (SP) register indicates the current word in the stack being addressed.

1.4 ES REGISTER (Extra Segment)

- It is used by some string operations to handle memory addressing.
- ES Register is associated with the DI Register.

2. INSTRUCTION POINTER (IP)

- The 16 bit IP Register contains the offset address of the next instruction that is to execute.
- IP is associated with CS register as (CS:IP)
- For each instruction that executes, the processor changes the offset value in IP so that IP in effect directs each step of execution.

3. THE QUEUE

• While the EU is decoding an instruction or executing an instruction which does not require use of the buses, the BIU fetches up to six instructions bytes for the following instructions.

• The BIU Stores prefetched bytes in First in First out register set called a queue.

When the EU is ready for its next instruction, it simply reads the instruction bytes for the instruction from the queue in the BIU.

• This is much faster than sending out an address to the system memory and waiting for memory to send back the next instruction bytes or bytes.

• Fetching the next instruction while the current instruction executes is called pipelining

Pin Configuration of 8086/Signal Diagram

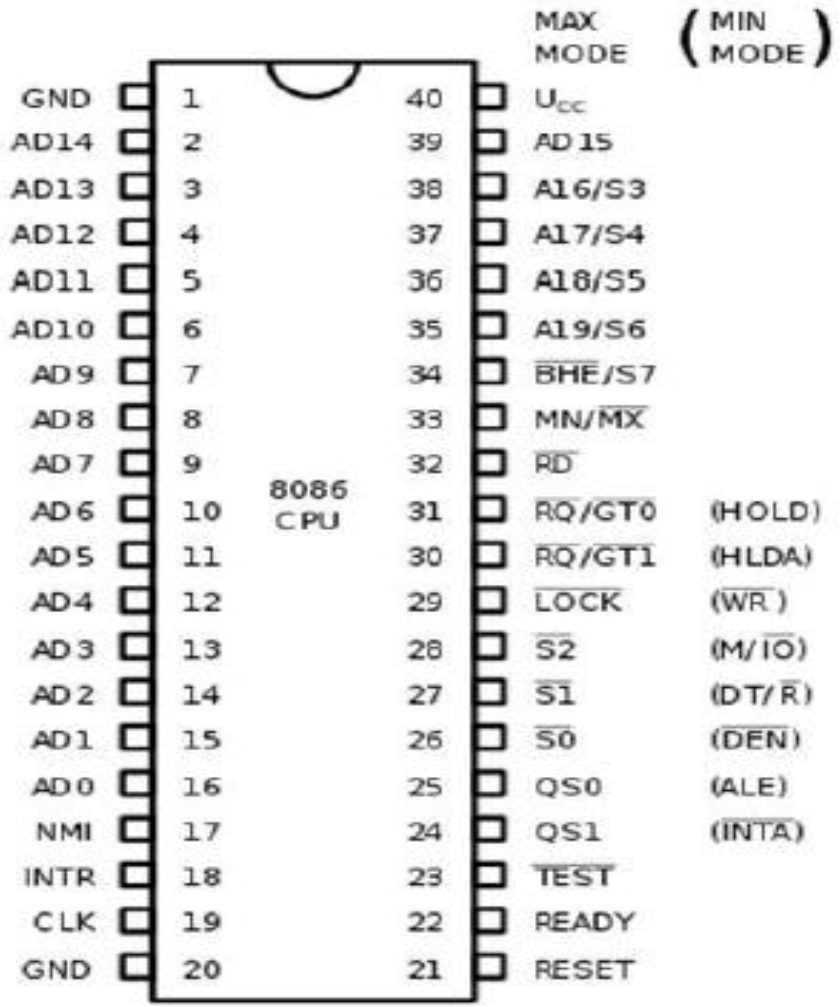


Fig: Pin Diagram of 8086

- **AD0-AD15 (Address Data Bus):** Bidirectional address/data lines. These are low order address bus. When these lines are used to transmit memory address the symbol A is used instead of AD for example A0-A15.
- **A16 - A19 (Output):** High order address lines. These are multiplexed with status signals.
- **A16/S3, A17/S4:** A16 and A17 are multiplexed with segment identifier signals S3 and S4.
- **A18/S5:** A18 is multiplexed with interrupt status S5.
- **A19/S6**: A19 is multiplexed with status signal S6.

- **BHE/S7 (Output):** Bus High Enable/Status. During T1, it is low. It enables the data onto the most significant half of data bus, D8-D15. 8-bit device connected to upper half of the data bus use BHE signal. It is multiplexed with status signal S7. S7 signal is available during T3 and T4.

• **RD (Read):** For read operation. It is an output signal. It is active when LOW.

• **Ready (Input):** The addressed memory or I/O sends acknowledgement through this pin. When HIGH it denotes that the peripheral is ready to transfer data.

• **RESET (Input):** System reset.

• **CLK (input):** Clock 5, 8 or 10 MHz.

• **INTR:** Interrupt Request.

- **NMI (Input):** Non-maskable interrupt request.
- **TEST (Input):** Wait for test control. When LOW the microprocessor continues execution otherwise waits.
- **VCC:** Power supply +5V dc.
- **GND:** Ground.

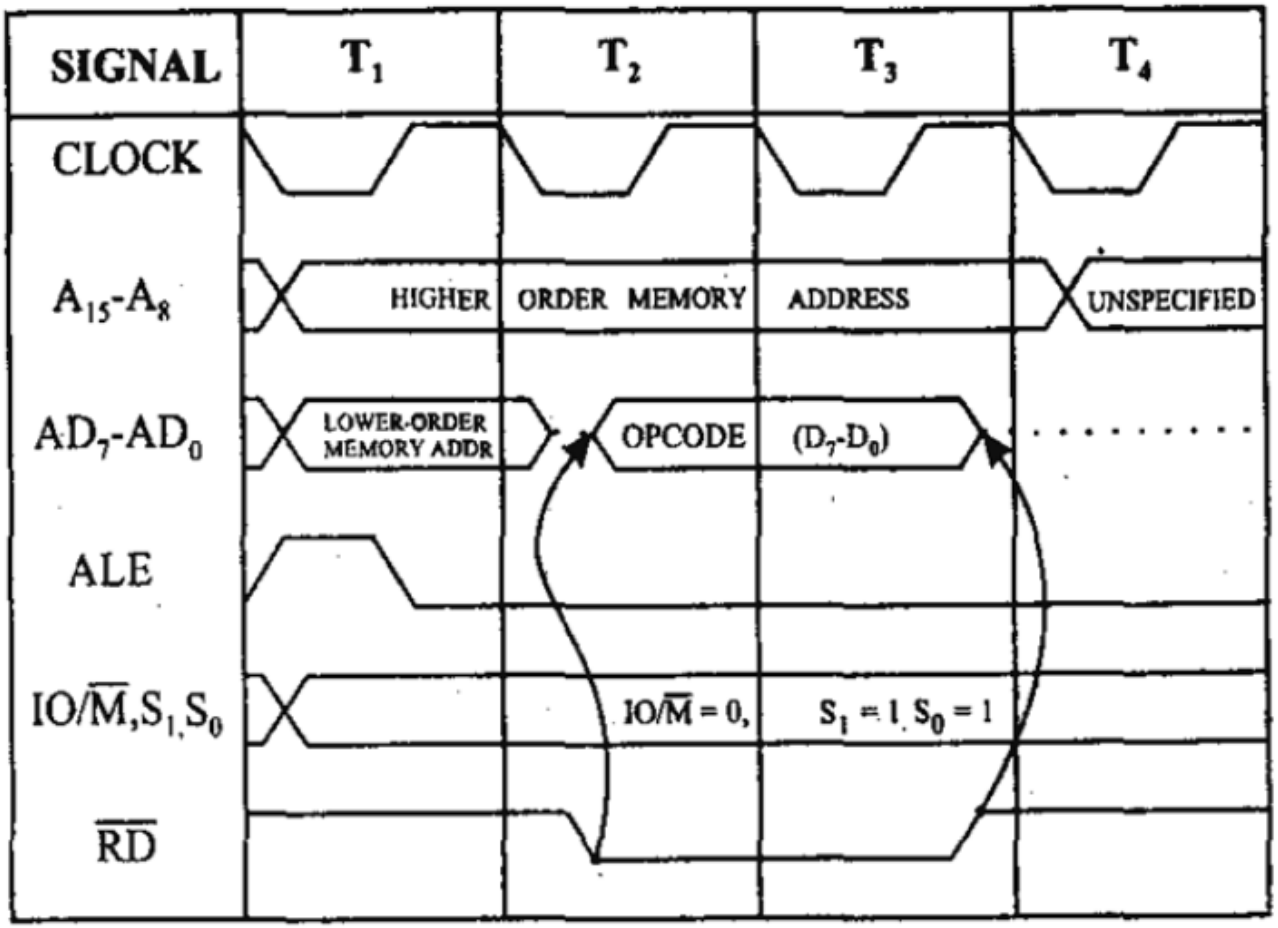
Fetch & Execute Operation: Timing Diagram

- The graphical representation of status of various signals involved during a machine cycle with respect to time is called timing diagram.
- This gives basic idea of what is happening in the system when the instruction is getting fetched and executed, at what instant which signal is getting activated.
- The signals involved during machine cycle are CLK, A15 – A8, AD7 – AD0, IO/M(bar), RD(bar), WR(bar) and S1,S0.

S_0.

IO/M(bar)	S_1	S_0	Operation
0	0	0	Halt
0	0	1	Memory write
0	1	0	Memory read
0	1	1	Op-code fetch
1	0	1	IO write
1	1	0	IO read
1	1	1	Interrupt acknowledge

Timing diagram for op-code fetch cycle



The op-code fetch timing diagram can be explained as below:

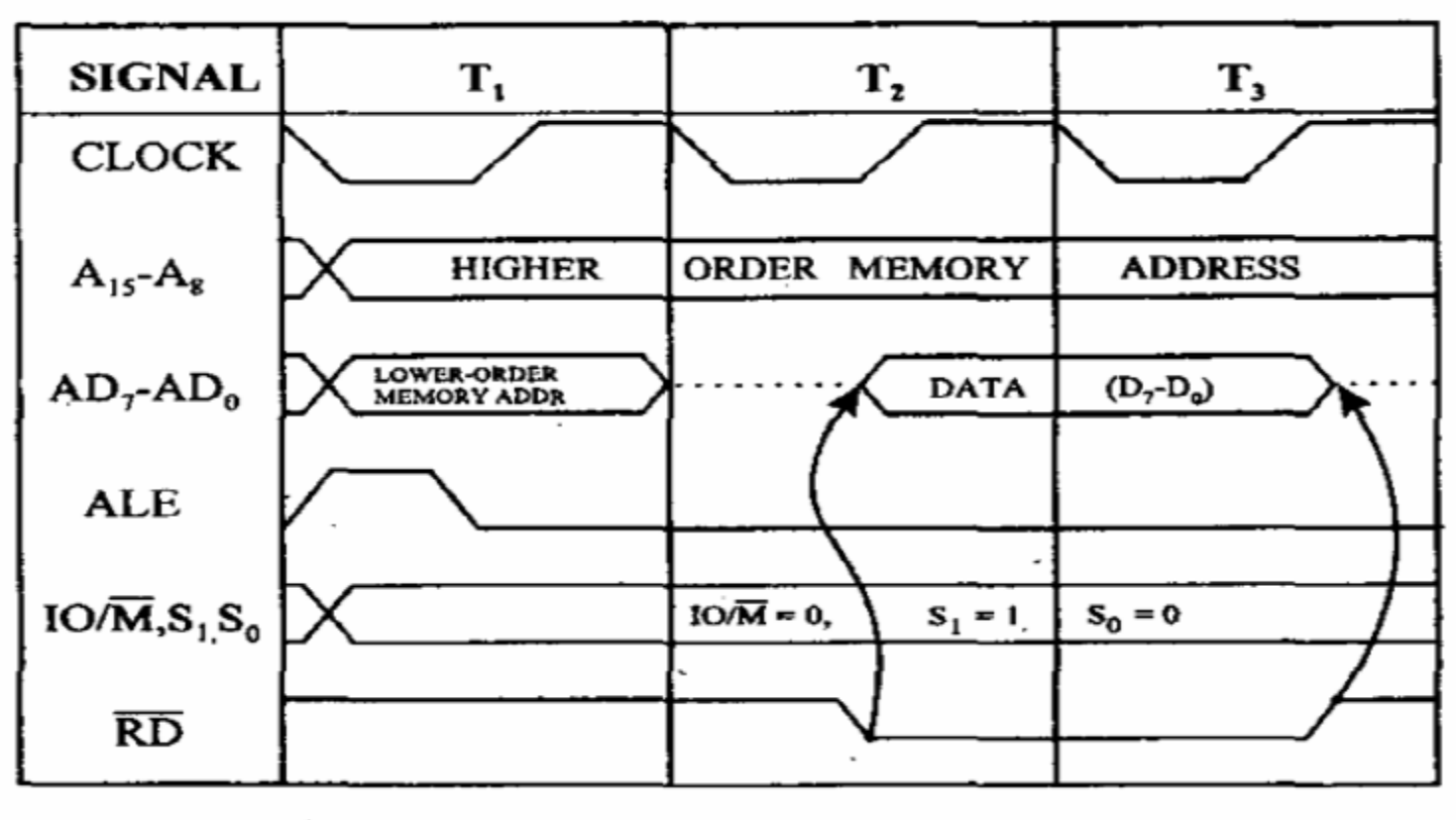
i) The MP places the 16-bit memory address from the program counter on address bus. At time period T1, the higher order memory address is placed on the address lines A15 – A8. When ALE is high, the lower address is placed on the bus AD7 – AD0. The status signal IO/M(bar) goes low indicating the memory operation and two status signals S1 = 1, S0 = 1 to indicate op-code fetch operation.

ii) At time period T2, the MP sends RD(bar) control line to enable the memory read. When memory is enabled with RD(bar) signal, the op-code value from the addressed memory location is placed on the data bus with ALE low.

iii) The op-code value is reached at processor register during T3 time period. When data (op-code value) is arrived, the RD(bar) signal goes high. It causes the bus to go into high impedance state.

iv) The op-code byte is placed in instruction decoder of MP and the op-code is decoded and executed. This happens during time period T4.

Timing diagram for memory read cycle



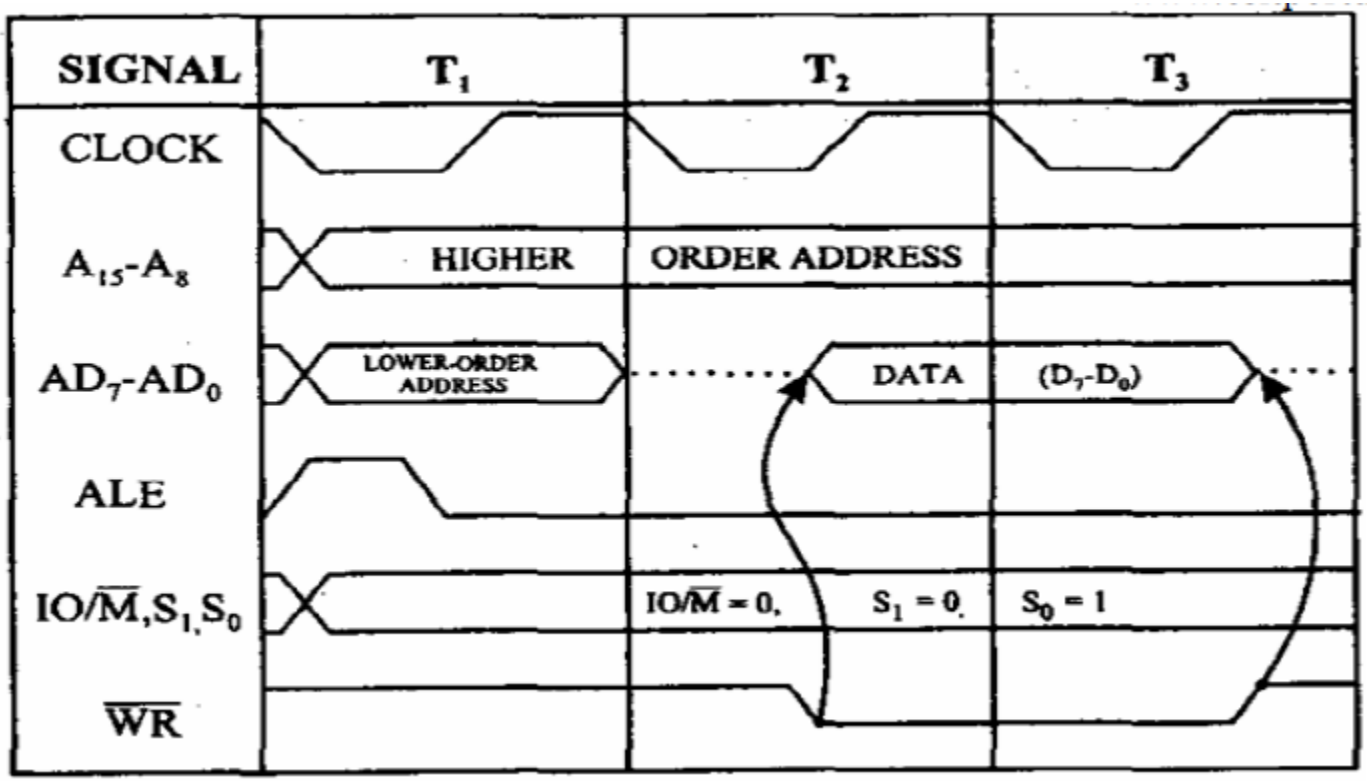
The memory read timing diagram can be explained as below:

i) The MP places the 16-bit memory address from the program counter on address bus. At time period T1, the higher order memory address is placed on the address lines A15 – A8. When ALE is high, the lower address is placed on the bus AD7 – AD0. The status signal IO/M(bar) goes low indicating the memory operation and two status signals S1 = 1, S0 = 0 to indicate memory read operation.

ii) At time period T2, the MP sends RD(bar) control line to enable the memory read. When memory is enabled with RD(bar) signal, the data from the addressed memory location is placed on the data bus with ALE low.

iii) The data is reached at processor register during T3 state. When data is arrived, the RD(bar) signal goes high. It causes the bus to go into high impedance state.

Timing diagram for memory write cycle



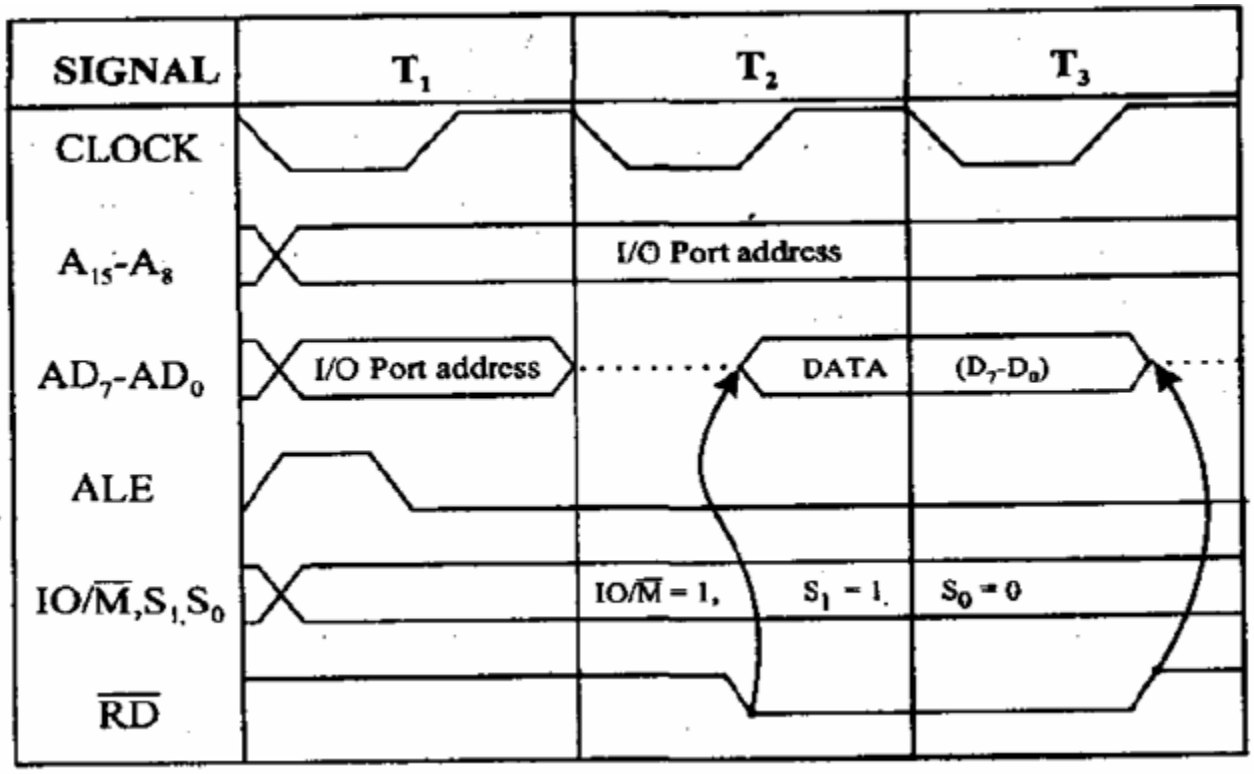
The memory write timing diagram can be explained as below:

i) The MP places the 16-bit memory address from the program counter on address bus. At time period T1, the higher order memory address is placed on the address lines A15 – A8. When ALE is high, the lower address is placed on the bus AD7 – AD0. The status signal IO/M(bar) goes low indicating the memory operation and two status signals S1 = 0, S0 = 1 to indicate memory write operation.

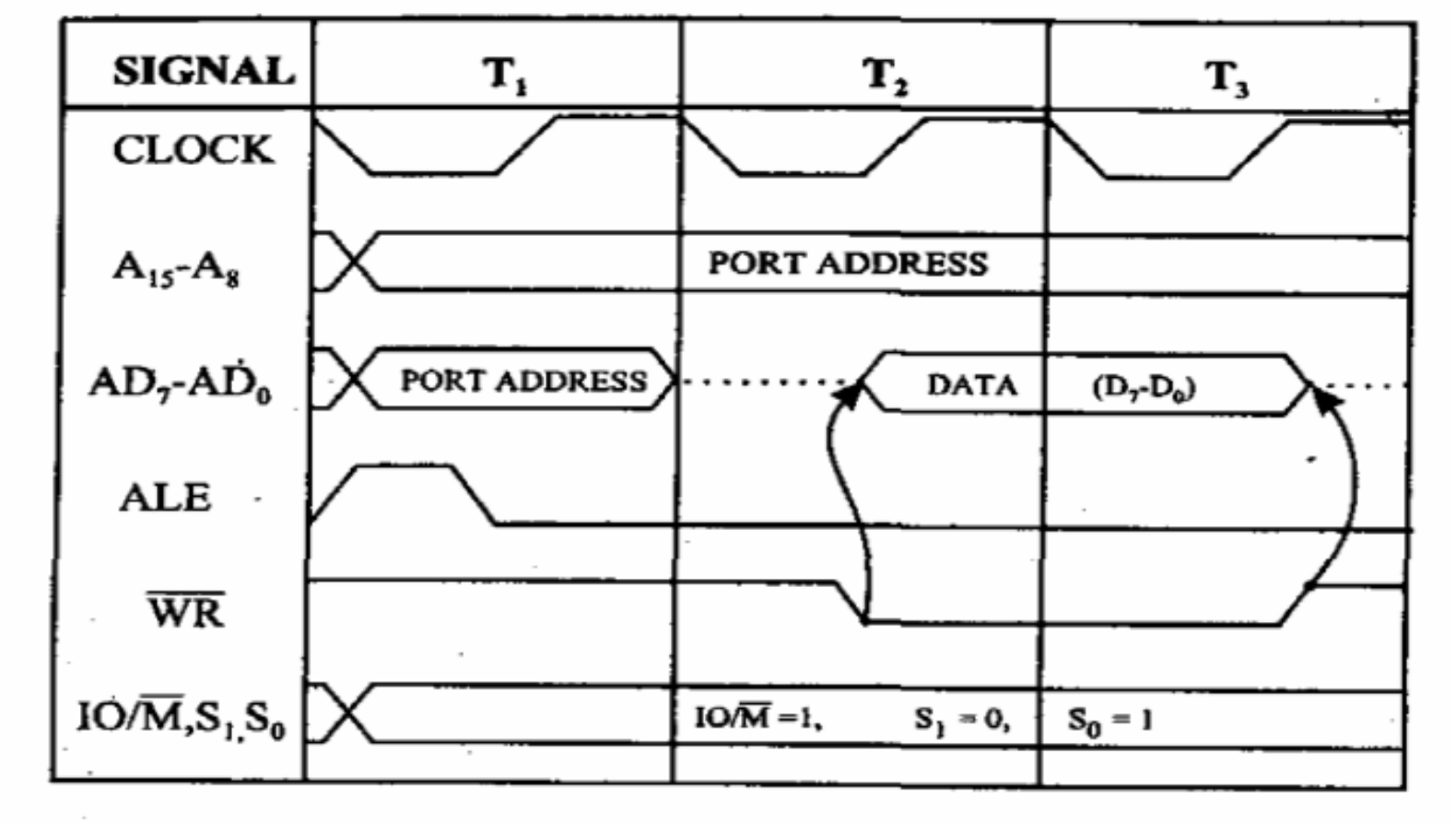
ii) At time period T2, the MP sends WR(bar) control line to enable the memory write. When memory is enabled with WR(bar) signal, the data from the processor is placed on the addressed location with ALE low.

iii) The data is reached at memory location during T3 state. When data is reached, the WR(bar) signal goes high. It causes the bus to go into high impedance state.

Timing diagram for IO read cycle



Timing diagram for IO write cycle



Timing Diagram of MOV Eg: MOV A,B

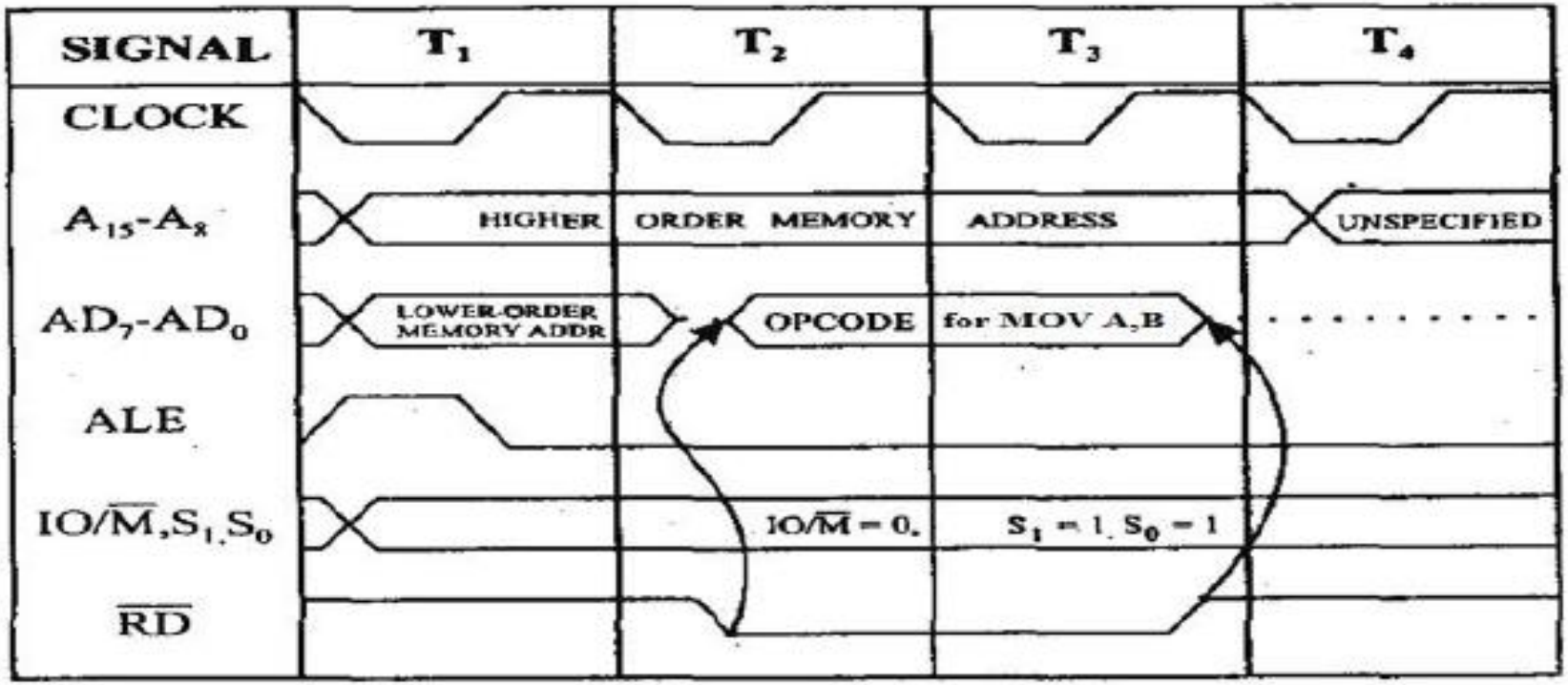
- The instruction MOV A ,B is of 1 byte; therefore the complete instruction will be stored in a single memory address.
- For example: 2000: **MOV A, B**

In Opcode fetch (t1-t4 T states):

- 00 – lower bit of address where opcode is stored, i.e., 00
- 20 – higher bit of address where opcode is stored, i.e., 20.
- ALE – provides signal for multiplexed address and data bus. Only in t1 it used as address bus to fetch lower bit of address otherwise it will be used as data bus.
- RD (low active) – signal is 1 in t1 & t4 as no data is read by microprocessor. Signal is 0 in t2 & t3 because here the data is read by microprocessor.
- WR (low active) – signal is 1 throughout, no data is written by microprocessor.
- IO/M (low active) – signal is 1 in throughout because the operation is performing on memory.
- S0 and S1 – both are 1 in case of opcode fetching.

Timing Diagram of MOV

Eg: MOV A,B

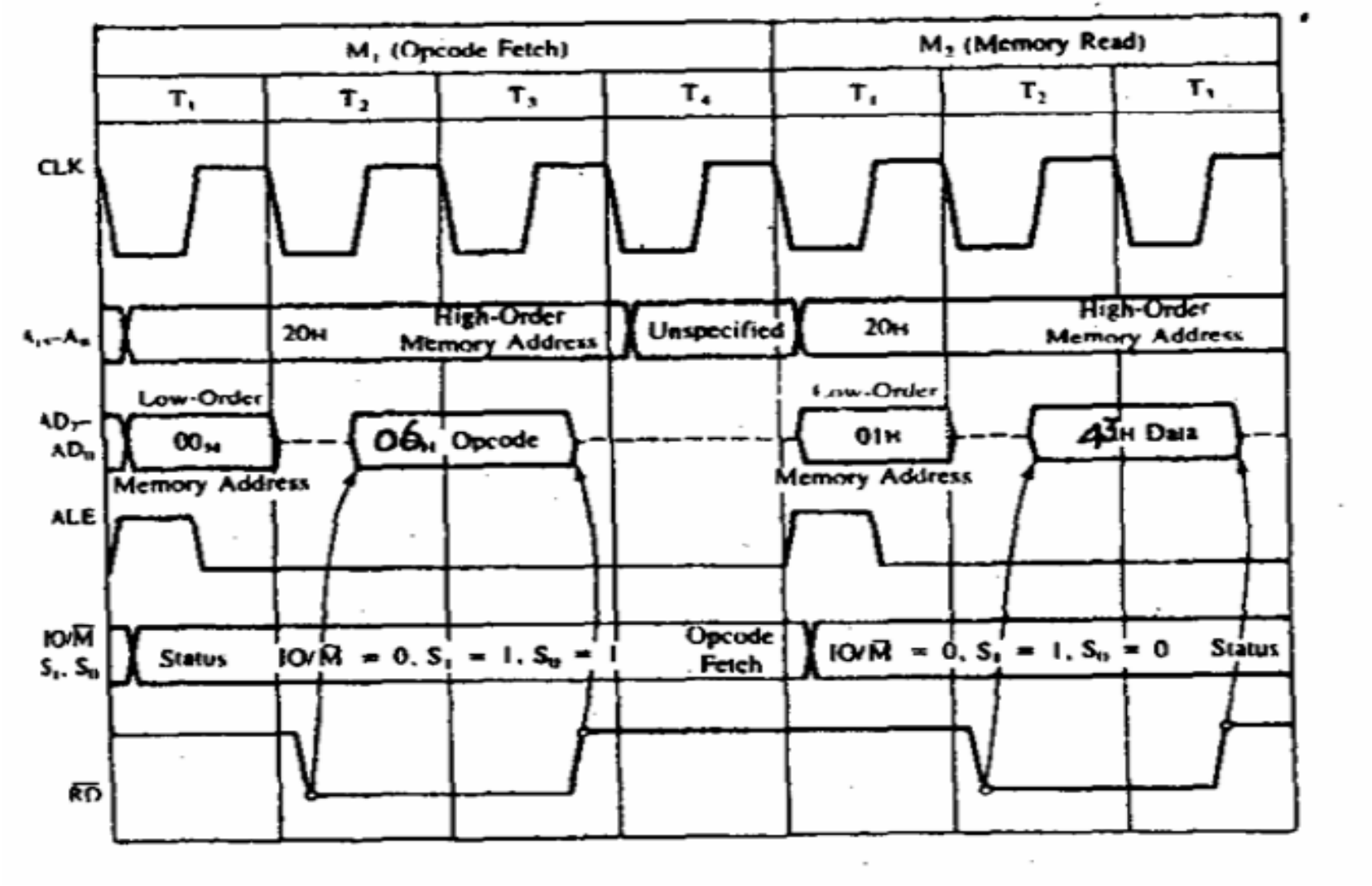


Timing Diagram of MVI Eg: MVI B, 43H

Timing diagram for MVI B, 43H

- Fetching the Op-code 06H from the memory 2000H. (OF machine cycle)
- Read (move) the data 43H from memory 2001H. (memory read)

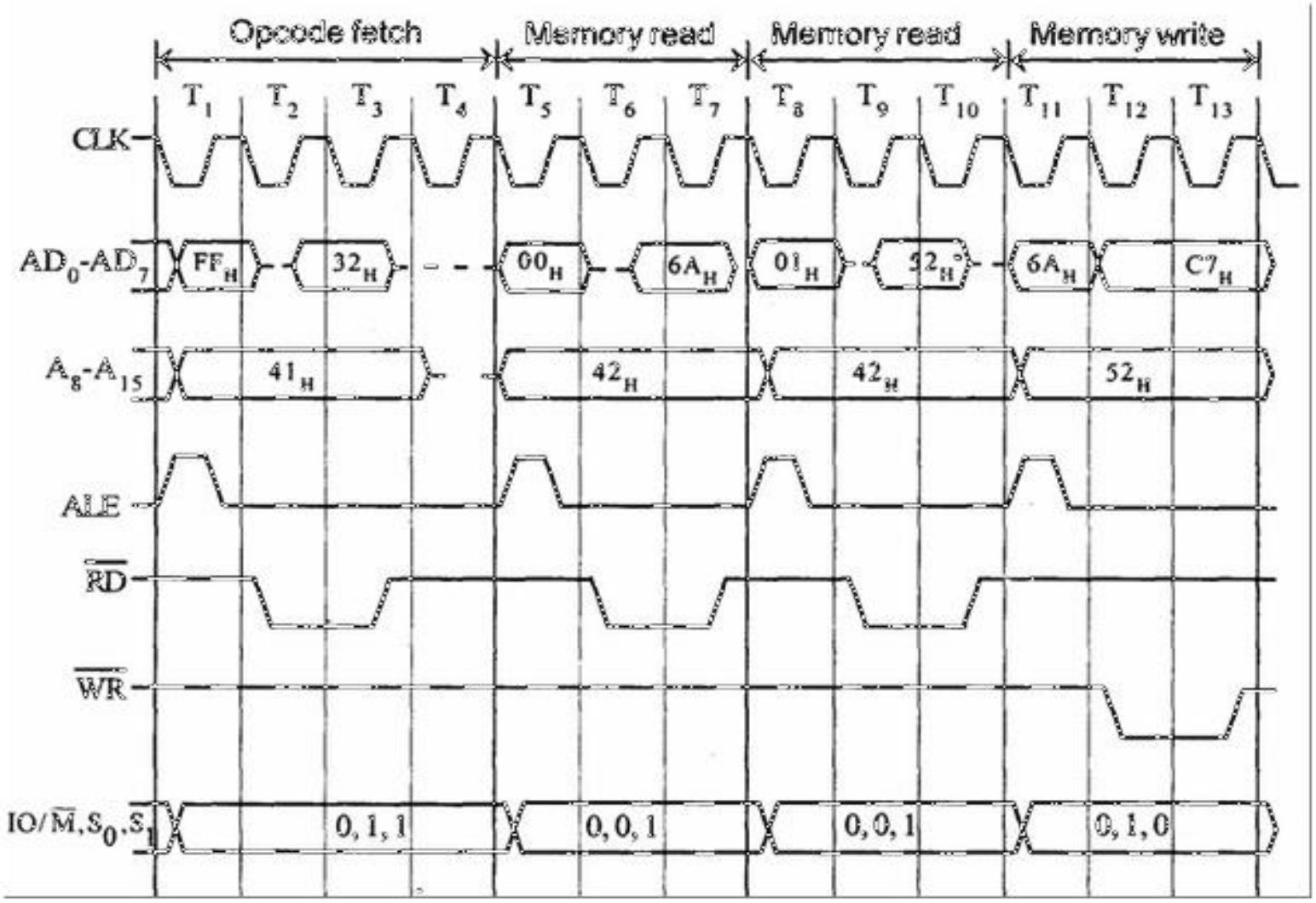
Address	Mnemonics	Op code
2000	MVI B, 43_H	06_H
2001		43_H


M1 (Opcode Fetch)
M2 (Memory Read)
T1
T2
T3
T4
T1
T2
T3
CLK
A15–A8
20H
High-Order Memory Address
Unspecified
20H
High-Order Memory Address
Low-Order
AD7–AD0
00H
06H Opcode
01H
43H Data
Memory Address
Memory Address
ALE
IO/M
S1, S0
Status
IO/M = 0, S1 = 1, S0 = 1
Opcode Fetch
IO/M = 0, S1 = 1, S0 = 0
Status
RD

Timing diagram for STA 526AH.

- STA means Store Accumulator -The contents of the accumulator is stored in the specified address(526A).
- The opcode of the STA instruction is said to be 32H. It is fetched from the memory 41FFH(see fig). - OF machine cycle
- Then the lower order memory address is read(6A). - Memory Read Machine Cycle
- Read the higher order memory address (52).- Memory Read Machine Cycle
- The combination of both the addresses are considered and the content from accumulator is written in 526A. - Memory Write Machine Cycle
- Assume the memory address for the instruction and let the content of accumulator is C7H. So, C7H from accumulator is now stored in 526A.

Address	Mnemonics	Opcode
41FF	STA $526A_H$	32_H
4200		$6A_H$
4201		52_H

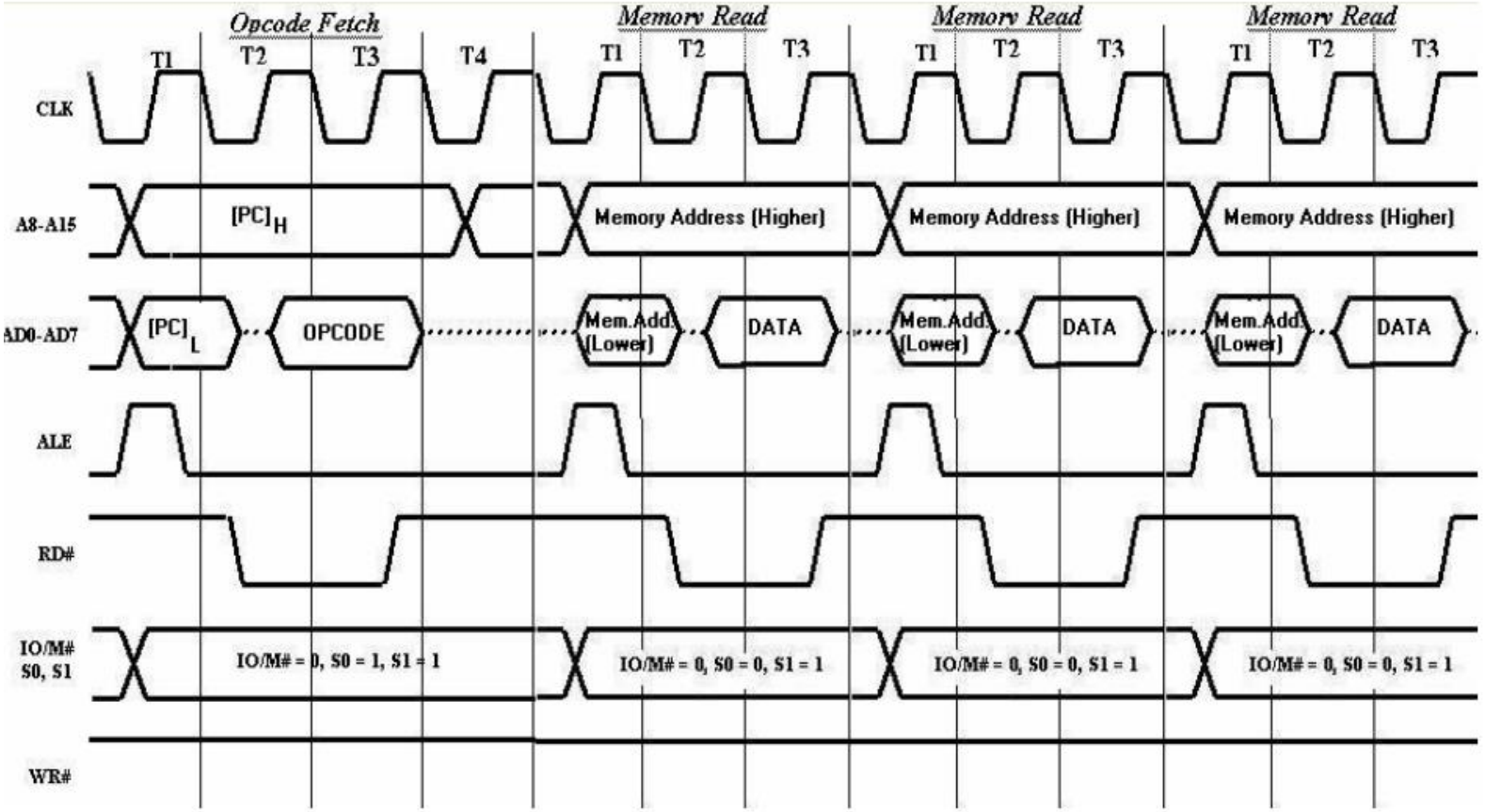

Opcode fetch
Memory read
Memory read
Memory write
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13
CLK
AD0-AD7
FFH 32H 00H 6AH 01H 52H 6AH C7H
A8-A15
41H 42H 42H 52H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1 0, 0, 1 0, 0, 1 0, 1, 0

Timing diagram for LDA 4050H

- Let us consider LDA 4050H as an example instruction of this type. It is a 3-Byte instruction. The initial content of memory address 4050H is ABH. Initially Accumulator content is CDH. As after execution A will be initialized with value ABH. Memory location 4050H will still remain with the content ABH. The results of execution of this instruction is as below −

	Before	After
(4050)	ABH	ABH
A	CDH	ABH

Address	Hex Codes	Mnemonic	Comment
2008	3A	LDA 4050H	A <- Content of the memory location 4050H
2009	50		Low order Byte of the address
200A	40		High order Byte of the address


Opcode Fetch
Memory Read
Memory Read
Memory Read
T1
T2
T3
T4
CLK
A8-A15
[PC]H
Memory Address (Higher)
AD0-AD7
[PC]L
OPCODE
Mem.Add\ (Lower)
DATA
ALE
RD#
IO/M#
S0, S1
IO/M# = 0, S0 = 1, S1 = 1
IO/M# = 0, S0 = 0, S1 = 1
WR#

Timing Diagram of IN

Timing diagram for IN C0H

→ Fetching the Op-code DBH from the memory 4125H.

→ Read the port address C0H from 4126H.

→ Read the content of port C0H and send it to the accumulator.

→ Let the content of port is 5EH.

Address	Mnemonics	Op code
4125	IN $C0_H$	DB_H
4126		$C0_H$

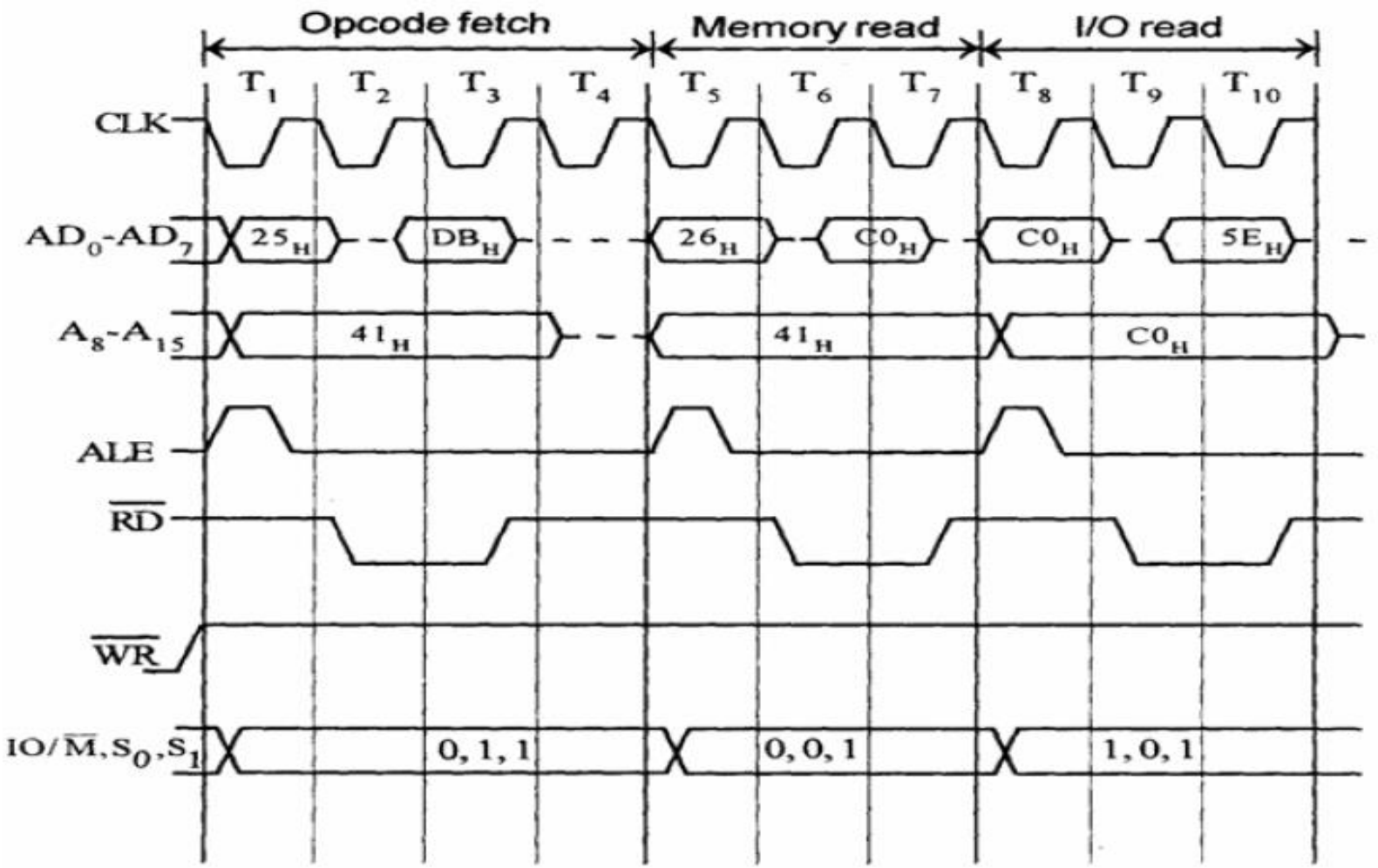
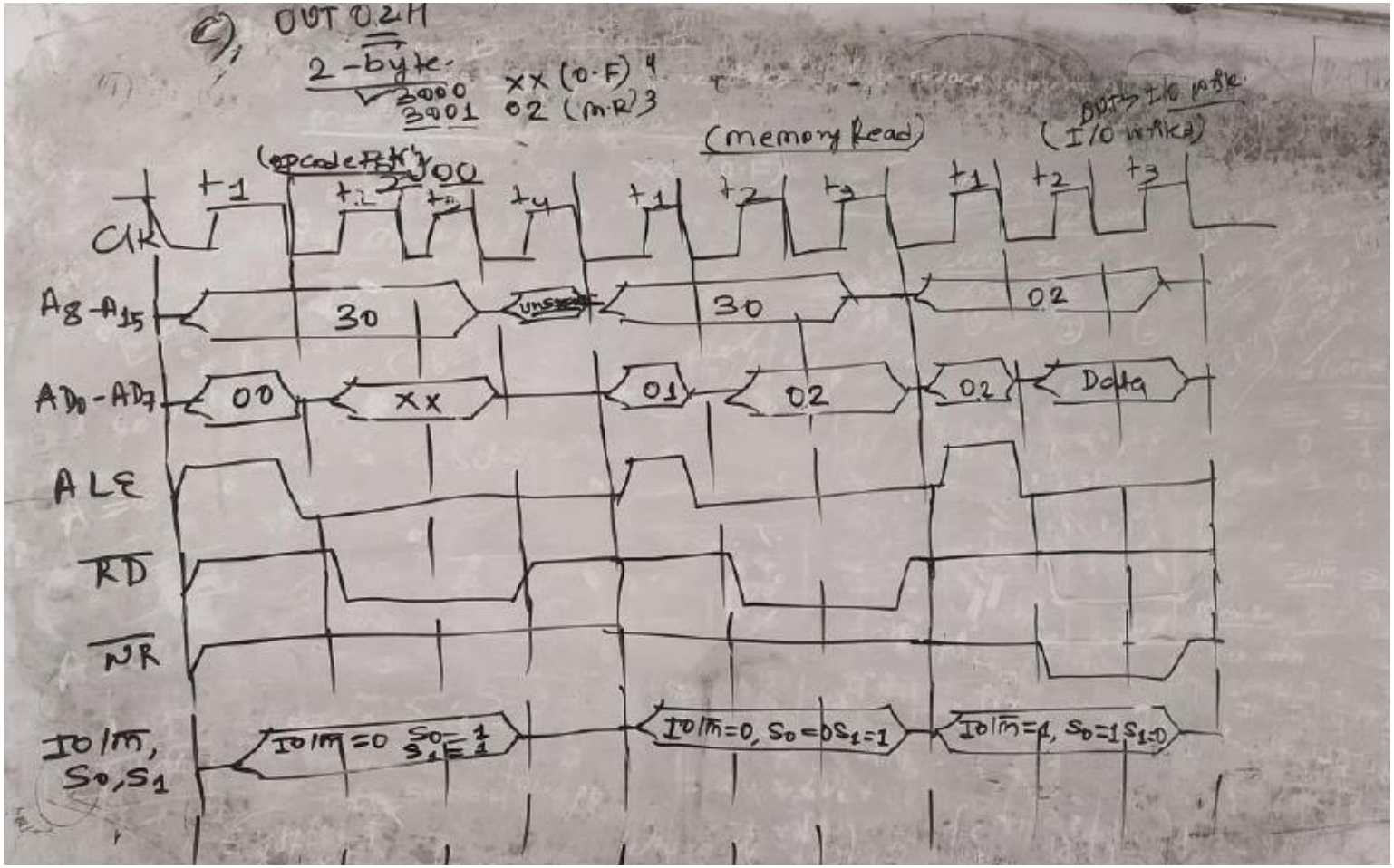
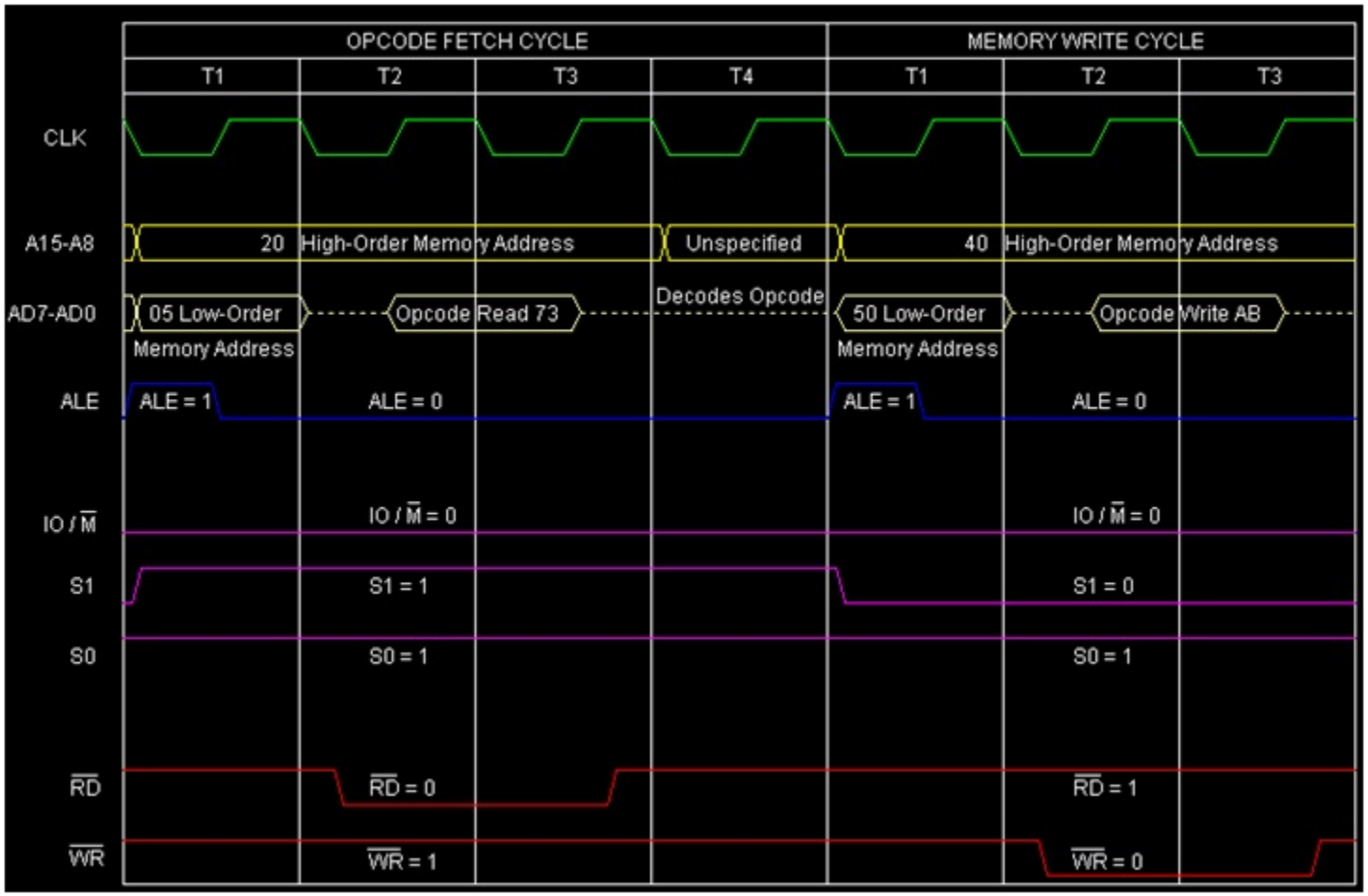


Fig: Timing Diagram for IN C0H

Timing Diagram of OUT



Timing Diagram of MOV M, E



Timing Diagram of MOV E, M