

## Fundamental of Microprocessor

### Introduction to microprocessors:

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 and processes data according to those instructions and provide result as output.

Microprocessor is programmable means it can be instructed to perform given task within its capability. A programmer can select appropriate instructions and ask the microprocessor to perform tasks on a given set of data. These instructions are entered or stored in storage called memory, which can be read by the microprocessor.

A microprocessor incorporates the functions of a computer's central processing unit (CPU) on a single integrated chip (IC).

Each MP communicates and operates in the binary number 0 and 1 called bits.

Each MP has fixed sets of instruction in the form of binary pattern called a machine language.

A typical microprocessor consists of arithmetic and logic unit (ALU) in association with control unit to process the instruction execution. Almost all the microprocessors are based on the principle of store program concept.

A typical programmable machine can be represented with four components: Microprocessor, memory, input and output. The physical components of this system are called hardware. A set of instructions written for the microprocessor to perform a task is called a Program, and a group of program is called Software.

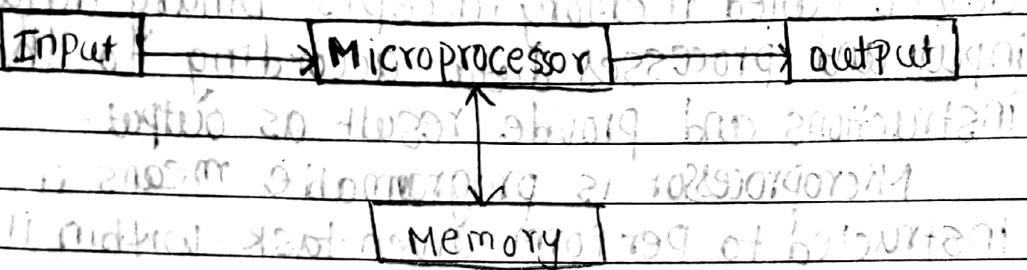


Fig: Typical programmable machine

Each MP recognizes and processes a group of bits called the word and microprocessors are classified according to their word length such as 8-bit MP with 8-bit word and 32-bit microprocessor with 32-bit word etc.

### Application of MP:

The application of MP are sorted as follows:

#### 1. Test instruments:

MPs are widely used in devices such as signal generators, oscilloscopes, counters, digital multimeters, X-ray analyzers, blood group analyzers, baby incubator, frequency synthesizers, data acquisition systems, spectrum analyzers etc.

#### 2. Communications:

Communication today required tens of thousands of

circuits to be managed. Data should be received checked for errors and further analysis should also be performed. The speed at which the microprocessor can take decisions and compute errors is truly substantial.

### 3. Computer:

The microprocessor is a central processing unit (CPU) of the microcomputers. It can perform arithmetic and logic functions as well as control function. The control unit of microprocessor sends signals to input output units, memory, ALU and arrange the sequence of their controlling operation operations.

### 4. Industries:

The microprocessor is widely used in data monitoring systems, smart cameras for quality control, automatic weighing, batching systems, assembly machine control, torque certification systems, machine tool controller etc.

### 5. Embedded System:

Used in microcontrollers.

### 6. Scientific and Engineering research.

Basic Block diagram of computer:

Traditionally, the computer is represented with four components such as memory, input, output and central processing unit (CPU) which

Box 4 consists of arithmetic logic unit (ALU) and

control unit (CU).

ALU performs arithmetic and logical operations.

CU controls ALU and I/O devices.

CPU

ALU

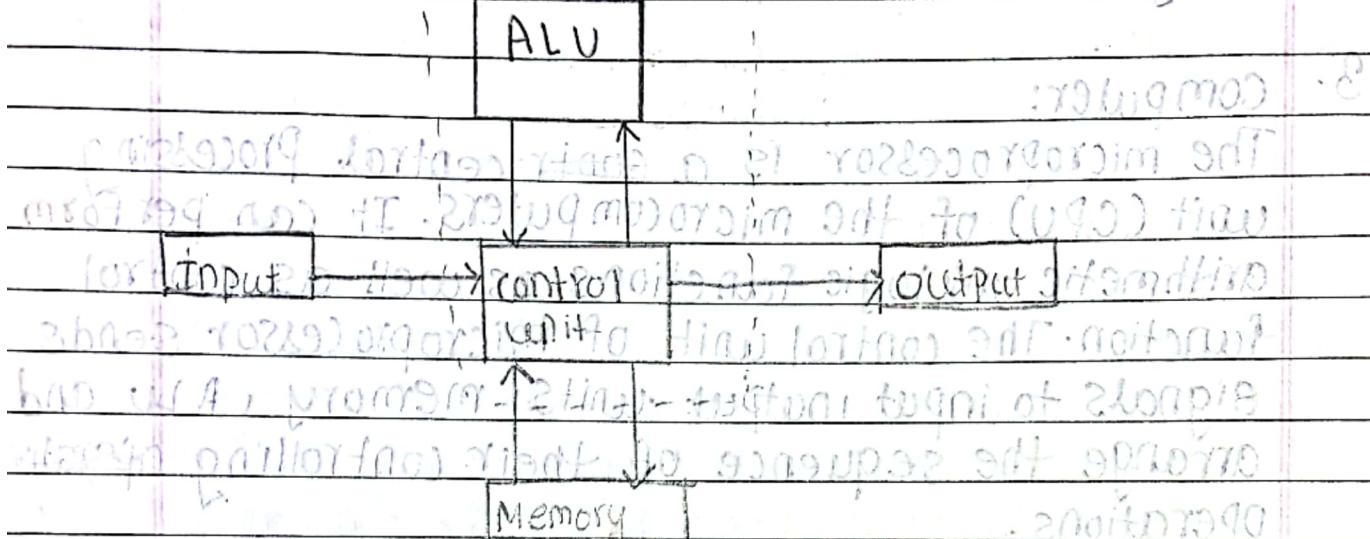


Fig: Traditional Block diagram of Computer I

The CPU contains various registers to store data, the ALU to perform arithmetic and logical operations, instruction decoders, counters and control lines.

The CPU reads instructions from memory and perform the tasks specified. It communicates with input / output (I/O) devices either to accept or to send data, the I/O devices is known as peripherals.

Later on around late 1960's, traditional block diagram can be replaced with computer having microprocessor as CPU which is known as microcomputer.

Here, CPU was designed using integrated circuit technology (IC's) which provided the possibility to

built the CPU on a single chip.

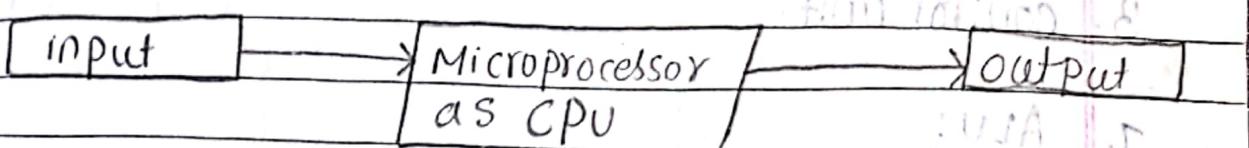


fig: Block diagram of a computer with microprocessor as CPU.

Later, on semiconductor fabrication technology became more advanced, manufacturers were able to place not only MPU (Microprocessing unit) (complete processing unit with necessary control signals) but also memory and I/O interfacing circuits on a single chip known as microcontroller, which also includes additional devices such as A/D converter, serial I/O, timer etc.

### Microprocessor's Components:

Microprocessor consists of three segments/parts given below:

1. ALU

2. Register Array
3. Control unit.

### 1. ALU:

This area of microprocessor perform various functions on data. The ALU performs arithmetic operation like addition, subtraction and logical operation like AND, OR, X-OR etc.

### 2. Register Array:

This area of microprocessor consist of various registers identified by B, C, D, E, H, L. These registers are used to temporary store the data during the execution of a program.

### 3. Control signal unit:

This area provides the timing and control signal to all the operations in the microcomputer. It controls the flow of data between the microprocessor memory and peripheral.

## Microprocessor Systems with bus organization:

Microprocessor based system includes three components. Microprocessor, input / output and memory (read only and read / write). These components are organized around a common communication path called a bus.

Ques 2) What is Microprocessor? What are its functions?

Ans: Microprocessor is a central processing unit which performs arithmetic and logical operations.

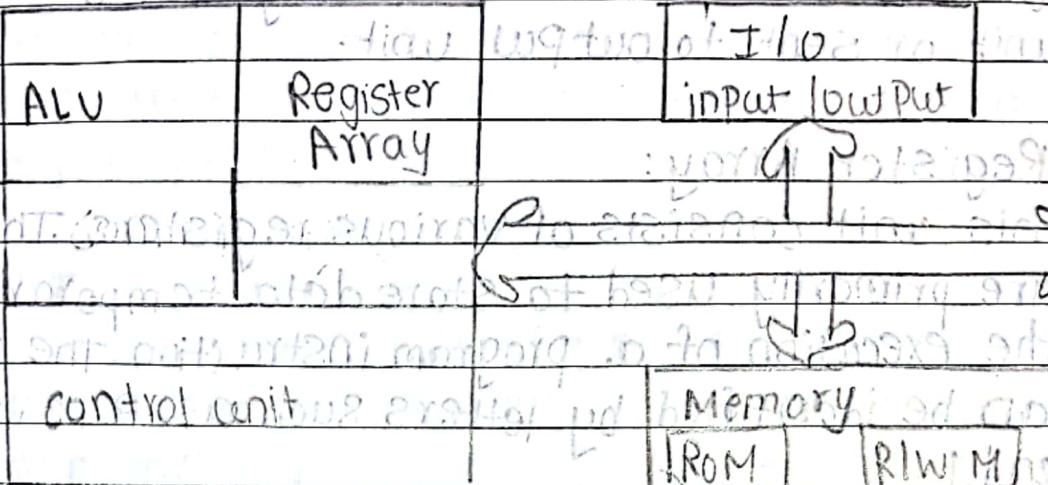


fig: HP based System with bus architecture

Ques 3) Explain what is meant by cache memory of a computer system.

### 1. Microprocessor:

It is a clock driven semiconductor device consisting of electronic logic circuits manufactured by using either a large scale integration (LSI) or very large scale integration (VLSI) technique. It is capable of performing various computing functions and making decisions to change the sequence of program execution. It can be divided into three segments:

(a) Arithmetic logic unit (ALU)

(b) Register Array

(c) Control unit

#### (a) ALU:

This unit executes all arithmetic and logical operations such as addition, subtraction and AND, OR and XOR respectively as specified by instruction set and produces output. The result of arithmetic

(addition, subtraction) and logical operations (AND, OR, XOR) are stored in the register or in memory unit or sent to output unit.

### (b) Register Array:

This unit consists of various registers. The registers are primarily used to store data temporarily during the execution of a program instruction. The registers can be identified by letters such as B, C, D, E, H and L.

### (c) Control unit:

It provides the necessary timing and control signals to all the operations in the microprocessor. It

controls the flow of data between the CPU and memory and peripheral units.

### Memory:

Memory stores binary information such as instructions and data and provides that information to the CPU whenever necessary. To execute programs, the CPU reads instructions and data from memory and

performs the computing operations in its ALU. Its

Results are either transferred to the output section for display or stored in memory for later use.

Memory has two sections:

### (a) Read only memory (ROM):

It is used to store programs that do not need alteration, alterations and can only be read.

### (b) Read / write memory (RAM):

It is also known as user memory which is used to store user programs and data. The information stored in this memory can be easily read and altered.

### 3. Input / output:

It communicates with the outside world using two devices input and output which are also known as peripherals.

The input device such as keyboard, switches and analog to digital converter transfer binary information from outside world to the UP.

The output device transfer data from the UP to the outside world. They include the devices such as LED, CRT, digital to analog converter, Printer etc.

### 4. System Bus:

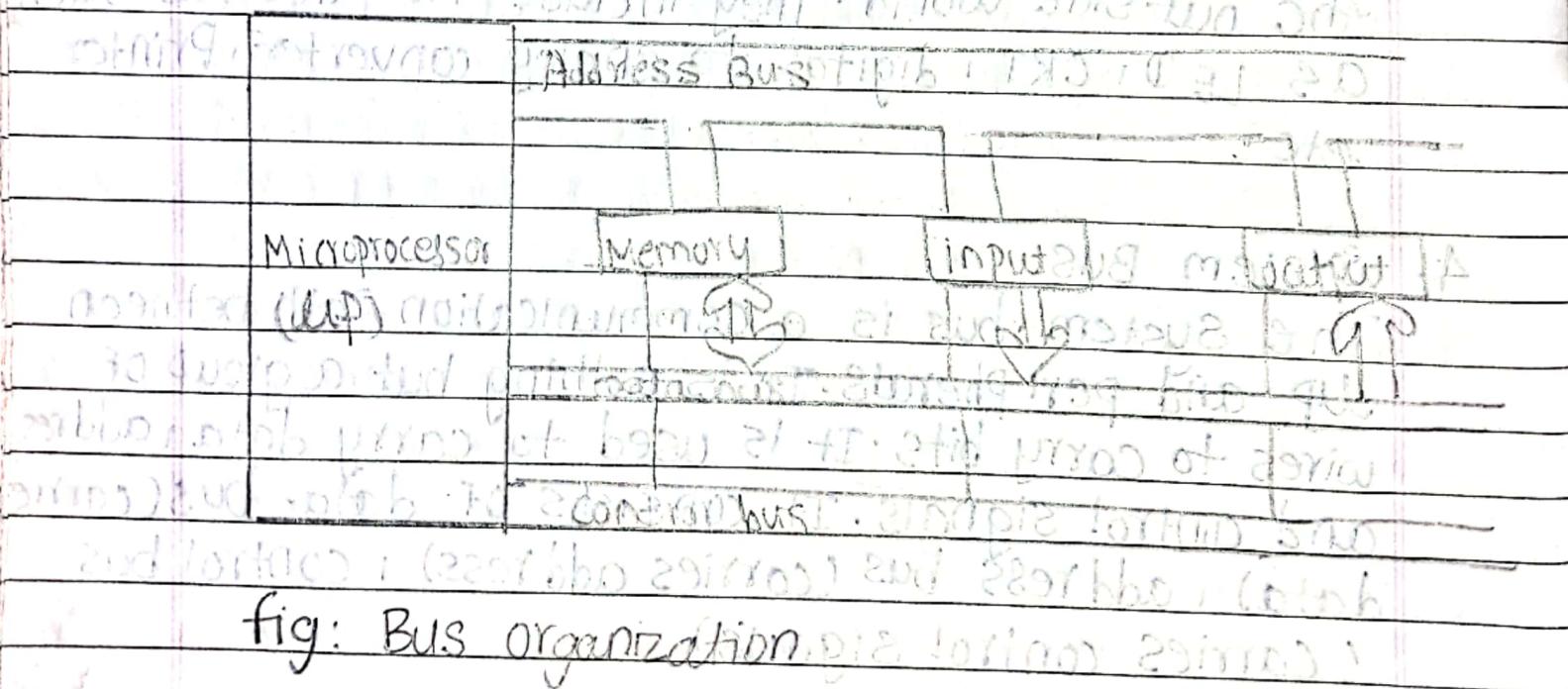
The System bus is a communication path between UP and peripherals. It is nothing but a group of wires to carry bits. It is used to carry data, address and control signals. It consists of data bus (carries data), address bus (carries address), control bus (carries control signals).

### Bus organization:

Bus is a common channel through which bits from any sources can be transferred to the destination. A typical digital computer has many register and paths must be provided to transfer instruction.

From one register to another. The number of wires will be excessive if separate lines are used between each register and all other registers in the system.

A more efficient scheme for transferring information between registers in a multiple registers configuration is a common bus system. A bus's structure consists of a set of common lines, one for each bit of a register through which binary information is transferred one at a time. Control signals determine which register is selected by the bus during each particular register transfer.



A very easy way of constructing a common bus system is with multiplexers. The multiplexers select the source register whose binary information is then placed on the bus. A system bus consists of about 50 to 100 of

Separate lines each assigned a particular meaning or function. The lines can be classified into three functional groups; data, address and control lines. In addition, there may be power distribution lines as well.

- \* The data lines provide a path for moving data between system modules. These lines are collectively called data bus.
- \* The address lines are used to designate the source/destination of data on data bus.
- \* The control lines are used to control the access to and the use of the data and address lines. Control signals both command and timing signals that indicate the validity of data and address information. Command signals specify operation read/write, I/O-read/write, bus-request/grant, clock, reset, interrupt request/acknowledge etc.

### System Bus:

The System bus is a communication path between the CPU and peripherals. It is nothing but a group of wires which are used to carry data, addresses and control signals in form of bits.

All the peripherals and memory share the same bus however the CPU communicates with only one peripheral at a time.

The various types of system bus are :-

1. Address bus
2. Data bus
3. Control bus

### 1. Address bus:

The bus over which the CPU sends out the address of memory location is called as address bus.

The address bus carries the address of the memory location to write or read from.

The address bus may consist of 16, 20, 24 or 32 parallel signal lines. It is also used to send port address on the address bus when CPU reads data from or writes data to a port.

The address bus is a group of 16 lines generally identified as A<sub>0</sub> to A<sub>15</sub>. It is unidirectional i.e. bits flow in one direction from the MPU (Microprocessing unit) to peripheral devices. If there are n address lines, then it can directly address 2<sup>n</sup> memory locations.

For eg; if n = 16, then  $2^{16} = 65536$  memory location it can address most 8 bit chips including the 8085 are 16 have 16 address lines whereas pentium processor has 32 address lines.

### 2. Data Bus:

The data bus consists of 8, 16 or 32 parallel lines. It is a bidirectional bus i.e. data flow in both direction between the MPU and memory/Peripherals. It can read data from memory

or write data to the memory so, it is bit bidirectional)

The lower half of the system bus works as address bus as well as data bus and is expressed as AD<sub>n</sub> where n = 0, 1, ..., 7.

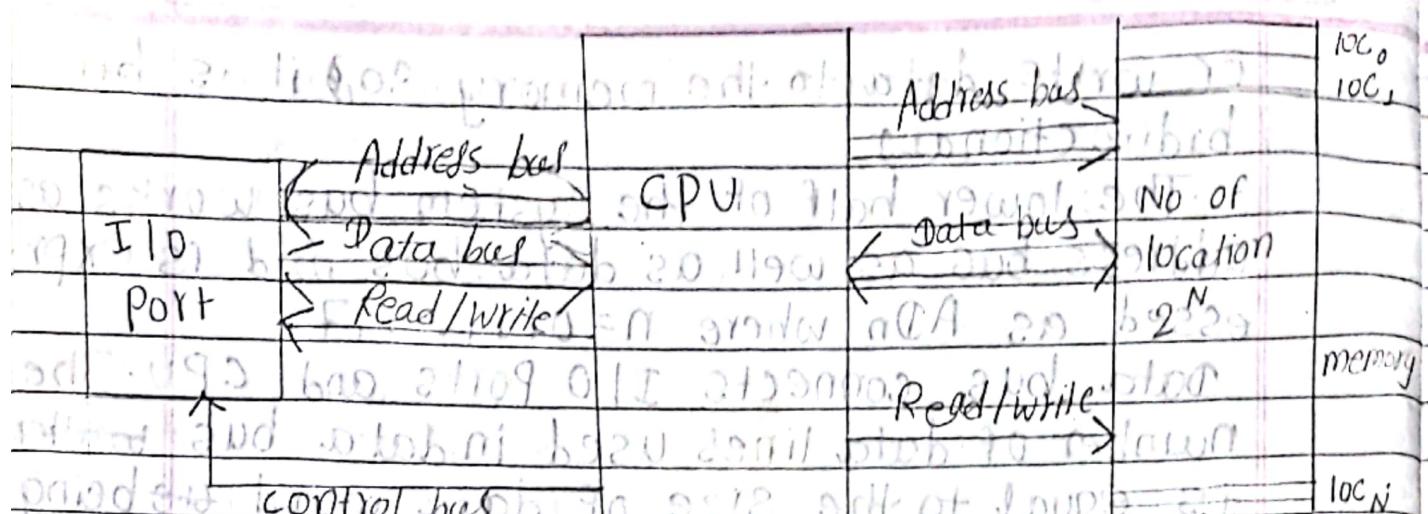
Data bus connects I/O Ports and CPU. The number of data lines used in data bus is equal to the size of data word being written or read. Note that many devices in the system will connect their outputs to the same data bus but only one device will have its outputs enabled at a time. So, there is no collision of data, actually.

### 3. Control Bus: a collection of control signals and timing signals

The control bus is used in sending control signals to the memory and I/O devices. The control bus is comprised of various signal lines that carry synchronization signals.

Control bus is the collection of individual lines that carry the control signals for every operation the MPU performs. The control signals transmit both command and timing information between the system modules. The timing signals indicate the validity of data and address information while the command signals specify the operation to be performed.

The control bus consists of 4-10 parallel signals lines. Control bus signals are memory read, write, I/O read, I/O write.



### Stored Program concept:

The simplest way to organize a computer is to have one processor, register and instruction code format with two parts OP-code and address/operand. The Memory address tells the control where to find an operand in memory. This operand is read from memory and used as data to be operated on together with the data stored in the processor register. Instructions are stored in one section of same memory. It is called stored program concept.

The tasks of entering and altering the programs for the ENIAC (Electronic Numerical Integrator AND Computer) was extremely tedious. The programming concept could be facilitated if the program could be represented in a form suitable for storing in memory alongside the data. Then a computer could get

its instruction by reading from the memory and program could be set or altered by setting the values of a portion of memory. This approach is known as stored program concept.

### Von Neumann Architecture

It's a theoretical design based on the concept of stored-program computers where program data and instruction data are stored in the same memory.

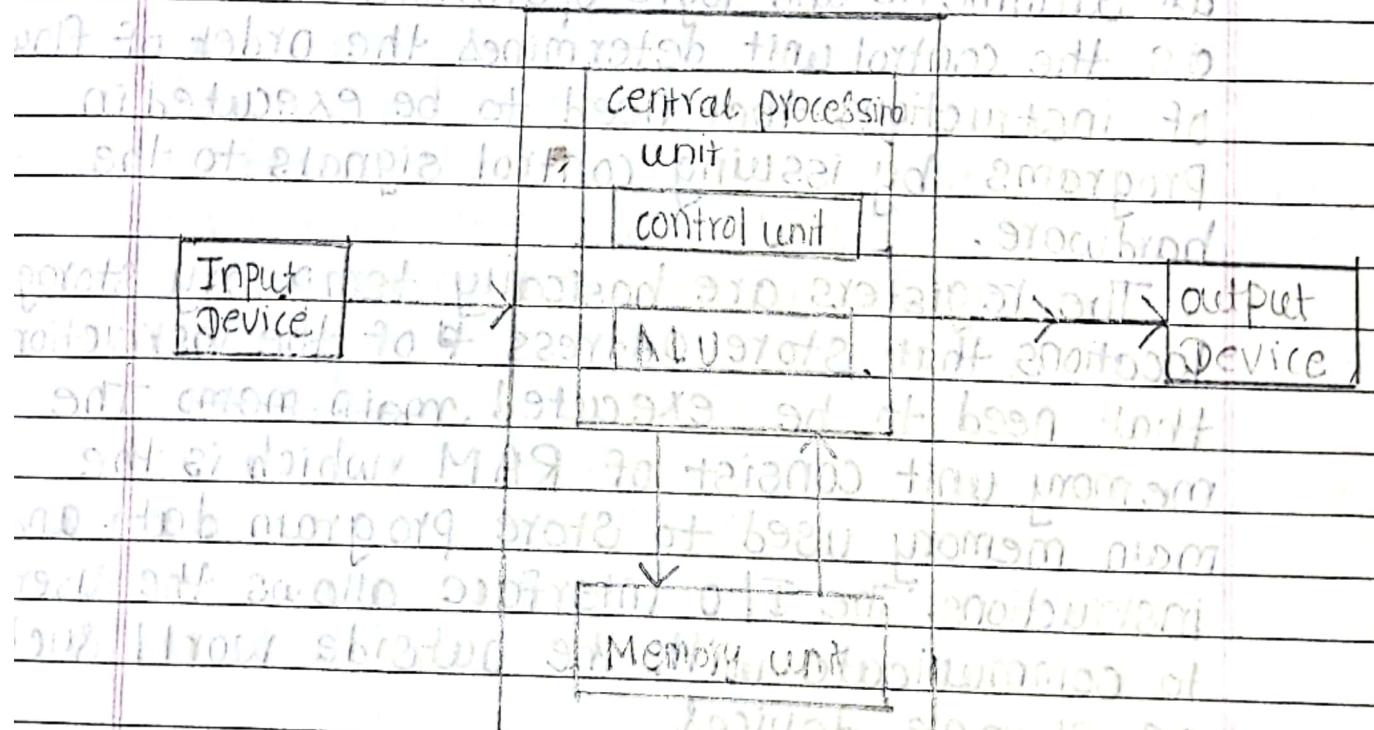


Fig: Block diagram of Von Neumann Architecture

The architecture was designed by the renowned mathematician and physicist John von Neumann in 1945.

The idea behind the Von Neumann architecture

ture is the ability to store instructions in the memory along with the data on which instructions operate. In short, von Neumann architecture refers to a general framework that a computer's hardware, programming and data should follow.

The von Neumann architecture consists of three distinct components: a central processing unit (CPU), memory unit, and Input/Output (I/O) interfaces. The CPU is the heart of the computer system that consists of three main components: the ALU, the control unit (CU), and registers.

The ALU is responsible for carrying out all arithmetic and logic operations on data, whereas the control unit determines the order of flow of instructions that need to be executed in programs by issuing control signals to the hardware.

The registers are basically temporary storage locations that store address # of the instructions that need to be executed. The memory unit consists of RAM, which is the main memory used to store program data and instructions. The I/O interface allows the users to communicate with the outside world such as storage devices.

### Harvard Architecture:

It is a computer architecture with physically separate storage and signal pathways for program data and instructions. Unlike von Neumann architecture which employs a single

bus to both fetch instructions from memory and transfer data from one part of a computer to another. Harvard architecture has separate memory space for data and instruction.

Both (Van Neumann and Harvard architecture) the concepts are similar except the way they access memories. The idea behind the Harvard architecture is to split the memory into two parts - one for data and another for programs. The term was based on the original Harvard Mark I relay based computer which employed a system that would allow both data transfers and instructions fetched to be performed at the same time.

Real world computer designs are actually based on modified Harvard architecture and are commonly used in microcontrollers and digital signal processing.

Instruction memory

Fig: Block diagram of Harvard architecture.

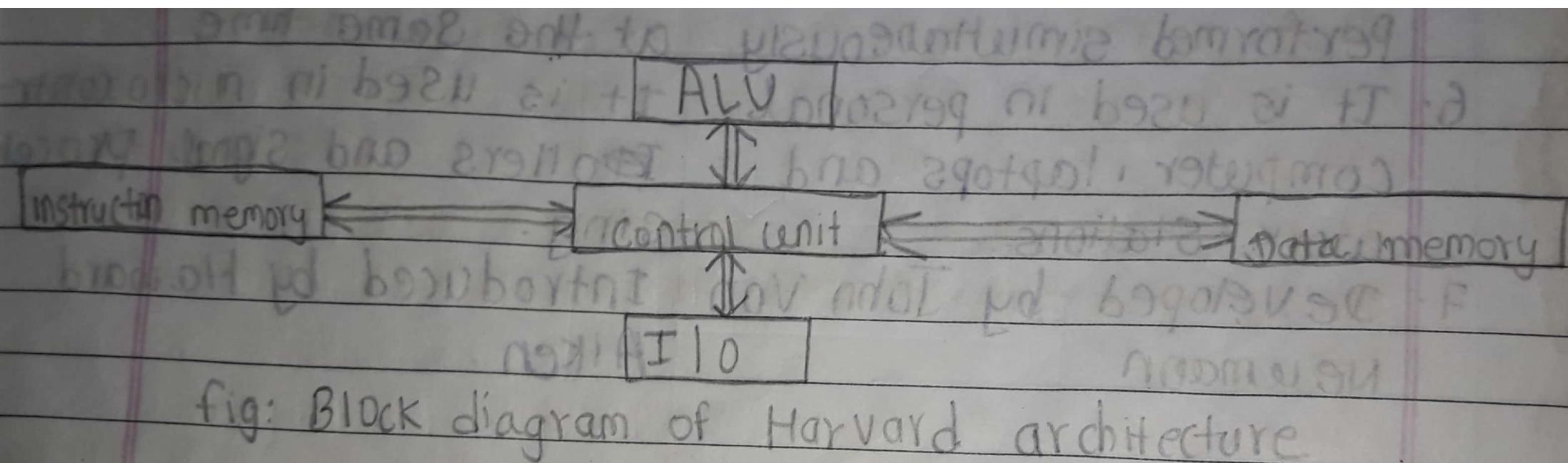
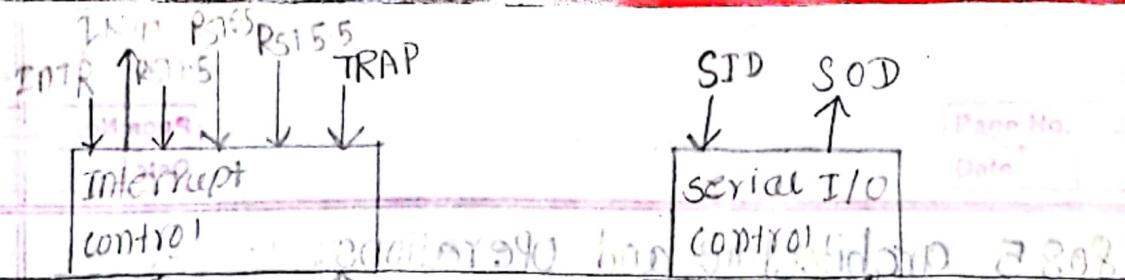


fig: BLOCK diagram of Harvard architecture

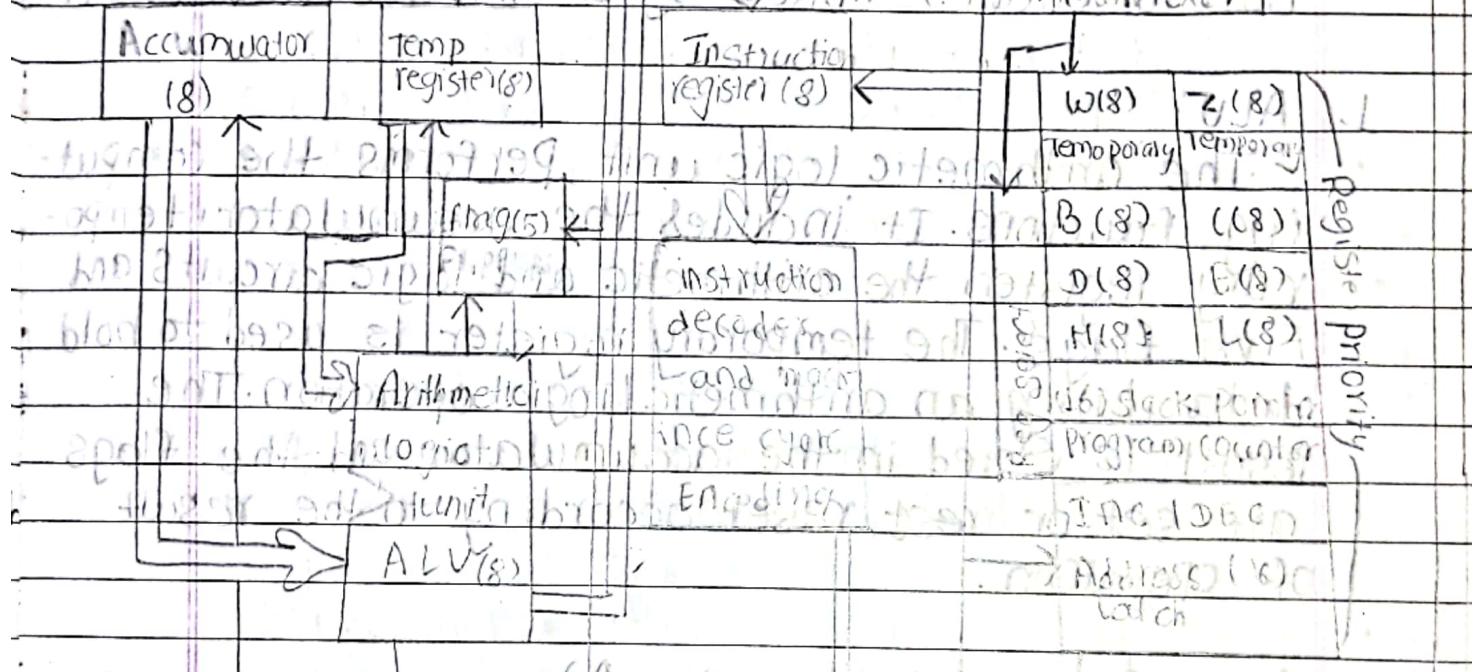
## Von Neumann vs Harvard Architecture

Von-Neumann	Harvard
1. It is a theoretical design based on the stored program computer concept.	It is a modern computer architecture based on the Harvard Mark I computer modes.
2. It uses same physical memory address for instructions and data.	It uses separate memory addresses for instruction and data.
3. Processor needs two clock cycles to execute an instruction.	Processor needs one cycle to complete an instruction.
4. Similar control unit design and development of one is more complicated which is cheaper and faster adds to the development cost.	Control unit for two buses adds to the development cost.
5. Data transfers and instructions fetches can not be performed simultaneously.	Data transfer and instructions fetches can be performed at the same time.
6. It is used in personal computer, laptops and work-stations.	It is used in micro controllers and signal processing.
7. Developed by John Von Neumann	Introduced by Howard Aiken.



internal bus connects to Address bus A8085 and Control bus C8085  
to other microprocessor, memory, peripherals, drives, etc.  
and 8-bit internal data bus to ALU A8085

bus priority logic receives address bus A8085 and control bus C8085  
and outputs to multiplexer



Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

Address bus A8085 connects to RAM, ROM, and I/O ports.

fig: 8085 block diagram

## 8085 architecture and operations:

The Intel 8085 A is a complete 8-bit parallel Central Processing unit. The main components of 8085A are array of registers, the arithmetic logic unit, the encoder / decoder and timing and control circuits linked by an internal data bus.

### 1. ALU:

The arithmetic logic unit performs the computing functions. It includes the accumulator, temporary register, the arithmetic and logic circuits and Five flags. The temporary register is used to hold data during an arithmetic logic operation. The result is stored in the accumulator and the flags are set or reset according to the result of operation.

### 2. Accumulator (register A)

It is an 8-bit register which is a part of ALU and stores the immediate result of arithmetic and logic operation. When data is read from input Port, it is first moved to accumulator and when data is sent to output port, it must be first placed in accumulator.

### 3. Temporary registers (W & Z)

They are 8-bit registers not accessible to the programmers. During Program execution 8085A places the data into it for a brief Period.

4. Instruction register and decoder:

It is an 8-bit register not accessible to the programmer. When an instruction is fetched from memory, it is loaded into the instruction register. The decoder decodes the instruction and establishes the sequence of events to follow.

5. Register Array (B, C, D, E, H, L) (16 PRs)

It is a 8-bit register accessible to the programmers. Data can be stored upon it during program execution. These can be used individually as 8-bit registers or in pair BC, DE and HL as 16 bit registers. Their contents may be incremented or decremented and combined logically with the content of the accumulator.

Stack Pointer (SP)

SP is a 16-bit register that holds the memory address. It points to a memory location in R/W memory called Stack. It holds the address of the beginning of the stack.

Program Counter (PC):

PC is a 16-bit register that holds the memory address. The program counter contains the address of the next instruction to be executed. When a byte is fetched, the PC is incremented by one to point to the next memory location.

6. Flag register:

Register consists of five flip-flops each holding

### Parity (P):

If the result of the last operation has even number of 1's (even parity), its status will be 1 otherwise 0.

Checking the status of different states separately is known as flag register and each register flip-flop are called flags. The flag register contains five flags namely: zero flag (Z), sign flag (S), carry flag (CY), parity flag (P) and auxiliary carry flag (AC).

The different flags are described as:

#### \* Carry (CY):

If the last operation generates a carry, its status will be 1 otherwise 0.

#### \* zero (Z):

If the result of last operation is zero, its status will be 1 otherwise 0.

#### \* Parity (P) Sign (S):

If the most significant bit (MSB) of the result of the last operation is 1 (negative) then its status will be 1 otherwise 0.

#### \* Auxiliary carry (AC):

If the last operation generate a carry from the lower half word, its status will be 1 otherwise 0.

The bit position of flags are shown below:

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
S	Z	-	AC	-	P	-	(Y)

7. Timing and control unit:

This unit synchronizes all the microprocessor operations with the clock and generates the control signals necessary for communication between the microprocessor and peripherals.

RD, WR, IO $\bar{M}$ , ALE etc. are some control signals.

### 8. Interrupt Control:

Interrupt control allows the microprocessor to respond to the external request made by the peripherals. The 8085 has five interrupt inputs namely: INTR, RST 5.5, RST 6.5, RST 7.5 and TRAP.

### 9. Serial I/O Controls:

It is used to control I/O operations with peripherals.

- Two serial I/O control signals (SID and SOD) are used to implement the serial data transmission.

#### (a) SID (Serial input data):

It is given by an input port to the microprocessor to indicate that it is ready for input operation.

#### (b) SOD (Serial output data):

It is given by a processor to the output device to make it ready for displaying the data.

8085 Pin configuration: w/ latch

X <sub>1</sub>	1	10000000000000000000000000000000	Vccpin (latch)
X <sub>2</sub>	2	00000000000000000000000000000000	HOLD
RESETOUT	3	00000000000000000000000000000000	CHLDATW, R
SOD	4	00000000000000000000000000000000	CLKout)
SID	5	00000000000000000000000000000000	RESETIN
TRAP	6	00000000000000000000000000000000	READYIN
RST 7.5	7	00000000000000000000000000000000	IO1 of bus
RST 6.5	8	00000000000000000000000000000000	SET - 2072d9
RST 5.5	9	00000000000000000000000000000000	RD1 : R/W
INTR	10	00000000000000000000000000000000	WR
INTA	11	00000000000000000000000000000000	ALE
AD <sub>0</sub>	12	00000000000000000000000000000000	0IE lines
AD <sub>1</sub>	13	00000000000000000000000000000000	Ad bus at TI
AD <sub>2</sub>	14	00000000000000000000000000000000	Ad bus out
AD <sub>3</sub>	15	00000000000000000000000000000000	Ad bus 073
AD <sub>4</sub>	16	00000000000000000000000000000000	Ad bus 22m port
AD <sub>5</sub>	17	00000000000000000000000000000000	Ad bus 104
AD <sub>6</sub>	18	00000000000000000000000000000000	Ad bus 21
AD <sub>7</sub>	19	00000000000000000000000000000000	Ad bus 702/200
UND	20	00000000000000000000000000000000	Ad bus 300

fig: 8085: (pin configuration) 2072) 5.5

Intel 8085 A contains 40 pins as shown in figure  
which has

- # 8 unidirectional address pins (A<sub>8</sub> to A<sub>15</sub>)
- # 8 bidirectional multiplexed address / data pins  
(AD<sub>0</sub> to AD<sub>7</sub>)
- # 11 control output pins.

- # 11 control input pins.
- # 2 Power Supply Pins (+5V and ground).

Pin 1 and 2 : X<sub>1</sub> and X<sub>2</sub>

It maintains the internal frequency of 8085 microprocessor. X<sub>1</sub> and X<sub>2</sub> can be crystal, RLC network or LC network.

Pin 3: RESET OUT

This signal is used to reset all the connected devices when the microprocessor is set reset.

Pin 4: SOD

It is a data line for serial output.

Pin 5: SID

It is a data line for serial input.

Pin 6 - 10: TRAP, RST 7.5, RST 6.5, RST 5.5, INTR

These are interrupt Pins which is used to interrupt the microprocessor to perform a task. TRAP has highest priority and INTR has a lowest priority.

Pin 11: INTA

It is an interrupt acknowledgement signal. The microprocessor sends an interrupt ACK after INTR is received.

Pin 12 - 19: AD<sub>0</sub> - AD<sub>7</sub>

These are multiplexed set of lines used to carry

the lower 8 bit address as well as data. The CPU may read or write out data through these lines.

Pin 20: V<sub>IND</sub>

This pin is used to ground (V<sub>SS</sub> or V)

Pin 21-28: A<sub>8</sub>-A<sub>15</sub>

These are address pins | bus which carries the most significant (higher order) 8 bits of the memory address

Pin 29,33: S<sub>0</sub>, S<sub>1</sub>

These pins are status signals and indicates the type of operation performed.

S <sub>0</sub>	S <sub>1</sub>	Operation
----------------	----------------	-----------

0	0	HALT
---	---	------

0	1	READ
---	---	------

1	0	WRITE
---	---	-------

1	1	FETCH AND SHIFT
---	---	-----------------

pin 30: ALE

This signal helps to capture the lower order address presented on the multiplexed address / data bus. When ALE signal is high, AD<sub>0</sub>-AD<sub>7</sub> is used as address bus A<sub>0</sub>-A<sub>7</sub>.

Pin 31: WR

This is an active low signal. A low indicates a write operation being performed into the selected memory or I/O device.

### Pin 32: RD

This is also an active low pin. When it goes low, the selected memory or I/O device is read.

### Pin 34: IOM

This pin is used to distinguish whether the address is for memory or I/O. When this pin is high the operation is performed between I/O and CPU.

When the pin is low, the operation is performed between memory and CPU.

### Pin 35: READY

This signal indicates that the device (peripheral) is ready to send or receive data. If READY is high, the device (peripheral) is ready. If it is low then the CPU waits till it goes high.

### Pin 36: RESETIN

This is an active low pin. This signal is used to reset the microprocessor by setting the program counter to zero.

### Pin 37: CLKOUT

This is used as the system clock for devices interfaced with the CPU.

### Pin 38: HLDA

This is a signal for hold acknowledgement. It indicates that the HOLD request has been received. After the removal of a HOLD request the HLDA goes low.

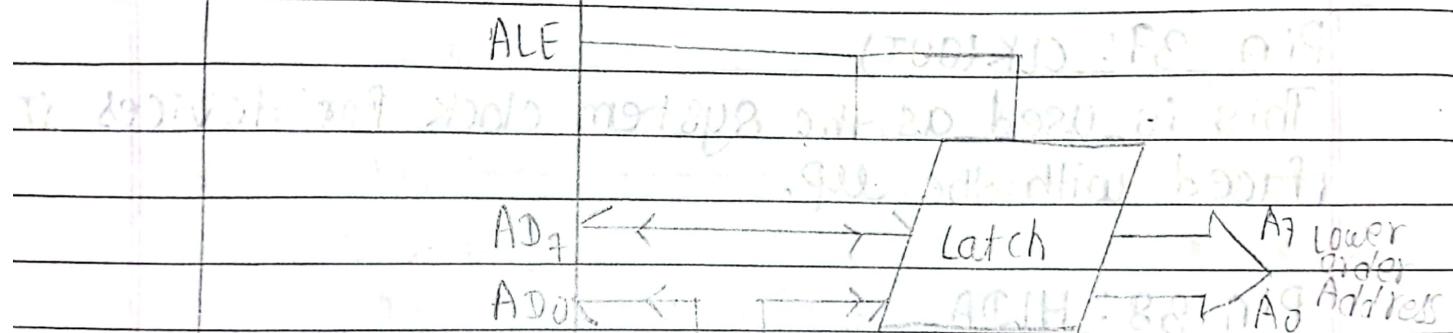
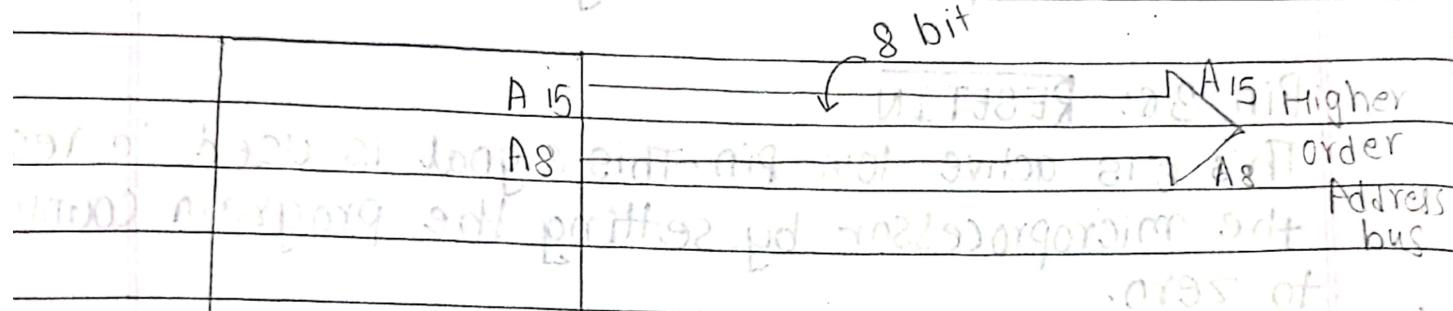
### Pin 39: HOLD

This signal indicates that another device is requesting the use of buses. Having received a HOLD request, the uP stops the use of buses or as soon as the current instruction is completed. The processor regains the bus after the removal of the HOLD signal.

### Pin 40: Vcc (+5V)

This pin provides +5V of electric current to the uP.

## # Multiplexing & Demultiplexing of Address / Data bus



The combination of bus for logic in a 16 bit bus  
can be used and example of Latch logic is given  
with 16 bit bus in diagram.

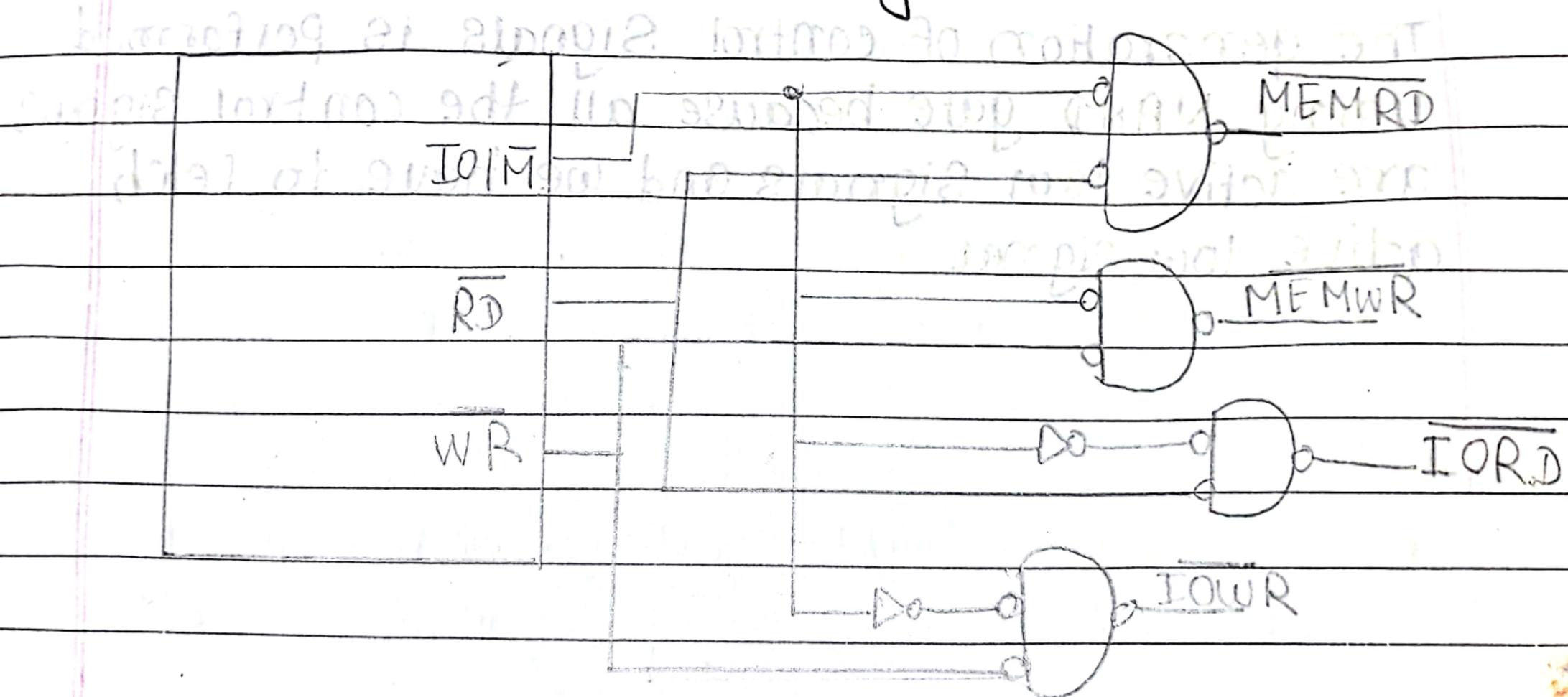
fig: multiplexing & demultiplexing of Address & data bus.

The address bus has 8 signal lines A<sub>8</sub> - A<sub>15</sub> which are unidirectional. The other 8 address bits are multiplexed with the 8 data bits. So, the bits AD<sub>0</sub> - AD<sub>7</sub> are bidirectional and serve as address bus i.e. A<sub>0</sub> - A<sub>7</sub> and data bus D<sub>0</sub> - D<sub>7</sub> at the same time. During the execution of the instruction, these lines carry the address bits during the carry part, then during the late parts of the execution, they carry the 8 bit data bits. In order to separate the address from the data, latch is used to save the value before the function of the bits changes.

The higher order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period.

When ALE (Address latch enable) is 1, it activates the latch circuit then takes the input from AD<sub>0</sub> - AD<sub>7</sub> and produces output at address A<sub>7</sub> - A<sub>0</sub>. When ALE goes AD<sub>7</sub> - AD<sub>0</sub> lines can be used as bidirectional data lines D<sub>0</sub> - D<sub>7</sub>.

## Veneration of control signals:



The 8085 CPU has signals given by I<sub>O/M</sub>, RD and WR. When the 8085 wants to read from and write into memory, it activates I<sub>O/M</sub>, RD and WR Signals.

By using three signals (lines), the CPU generates four control signals i.e. MEMRD, MEMWR, I<sub>ORD</sub> and I<sub>OWR</sub>.

The status of I<sub>O/M</sub>, RD and WR signals during memory and I/O read and write operation is given by

Memory	$I_0 \mid \bar{I}_M$	$\bar{R}D$	$RD$	$\bar{W}R$	$WR$	Operation
initial state	0 1 0 0 0 0 0	1	0	0	0	8085 CPU reads data from memory (memory address 210 to 219)
initial state	0 1 0 0 0 0 0	0	1	1	0	8085 CPU writes data to memory (address 210 to 219)
initial state	0 1 0 0 0 0 0	1	1	0	1	8085 CPU reads data from T10A device (TIA ready)
initial state	0 1 0 0 0 0 0	1	0	0	0	8085 CPU writes data to T10A device.

: Alphanumeric keyboard : 90 characters

The generation of control Signals is performed using NAND gate because all the control signals are active low signals and we have to fetch active low signal.

## Instruction:

An instruction is a binary pattern designed inside a CPU to perform a specific function. The entire group of instructions are called the instruction set. Each instruction has two parts:

One is the task to be performed called the operation code (opcode) and the other is data to be operated on, called the operand. Operand may include 8 bit or 16 bit data and internal register a memory location or an 8 (or 16 bit) address.

In some instructions, the operand is implicit.

The 8085 instruction set is classified into

3 groups according to word or byte size:

1. 1-byte instructions

2. 2-byte instructions

3. 3/1 byte instructions

1. 1-byte instructions:

A 1 byte instructions include opcode and operand in the same byte. Operands are internal register and are coded into the instruction. For e.g. MOV A,B

ADD C, ICMA, CMP, DAR, ANA, ADC, IDT, DCR, JNR etc are 1 byte instructions.

2. 2-byte instructions:

In a 2 byte instruction, the first byte specifies the opcode and the second byte specifies the operand.

Eg: ADI 50H; Add 8 bit data (50H) to the content of accumulator.

\* MVI A, 32H

### 3. 3-byte instructions:

In a 3 byte instruction, the first byte specifies the opcode and the following two bytes specify the 16 bit data or 16 bit address.

e.g.: LX+ B, 4050 H

1 byte 2 byte

\* LDA 2001 H

1 byte 2 byte

Bus Type:

#### 1. Synchronous Bus:

In a synchronous bus, the occurrence of the events on the bus is determined by a clock. The clock transmit a regular sequence of 0's and 1's of equal duration. All the events start at beginning of the clock cycle.

1. Here the CPU issues a START signal to indicate the presence of address and control information on the bus.

2. Then it issues the memory read signal and place the memory address on the address bus.

3. The addressed memory module recognizes the address and after a delay of one clock cycle it places the data and acknowledgement signal on the buses.

In Synchronous bus, all devices are tied to a fixed rate and hence the system cannot take

advantage of device performance but it is  
easy to implement.

Time taken in memory don't vary due to burst mode so it is easy to implement.

clock

T<sub>1</sub> T<sub>2</sub> T<sub>3</sub> T<sub>4</sub> T<sub>5</sub> T<sub>6</sub> T<sub>7</sub> T<sub>8</sub> T<sub>9</sub> T<sub>10</sub> T<sub>11</sub> T<sub>12</sub> T<sub>13</sub> T<sub>14</sub> T<sub>15</sub> T<sub>16</sub> T<sub>17</sub> T<sub>18</sub> T<sub>19</sub> T<sub>20</sub> T<sub>21</sub>

and add no along with

Address to Read to RAM add address of (2118M)

and add no along with last no. and

add no along with system program for adding add. &  
data start of 2118M

Address

Data bus

Acknowledge signal

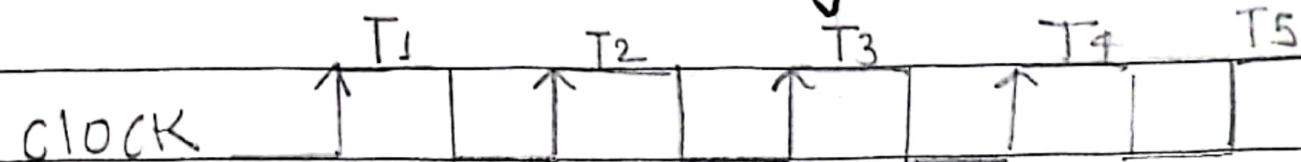
fig: Synchronous Read operation

## ~~An~~ Asynchronous Bus

In an asynchronous bus, the timing is maintained in such way that occurrence of one event on the bus follows and depends on the occurrence of previous event.

1. Here the CPU places Memory Read (control) and address signals on the bus.
2. Then it issues master synchronous signal (MSYNC) to indicate the presence of valid address and control signals on the bus.
3. The addressed memory module responds with the data and the slave synchronous signal (SSYNC).

T<sub>1</sub>T<sub>2</sub>T<sub>3</sub>T<sub>4</sub>T<sub>5</sub>



Read

Address

MSYNC

Data bus

SSYNC

fig: Asynchronous Read operation.

## Instruction cycle:

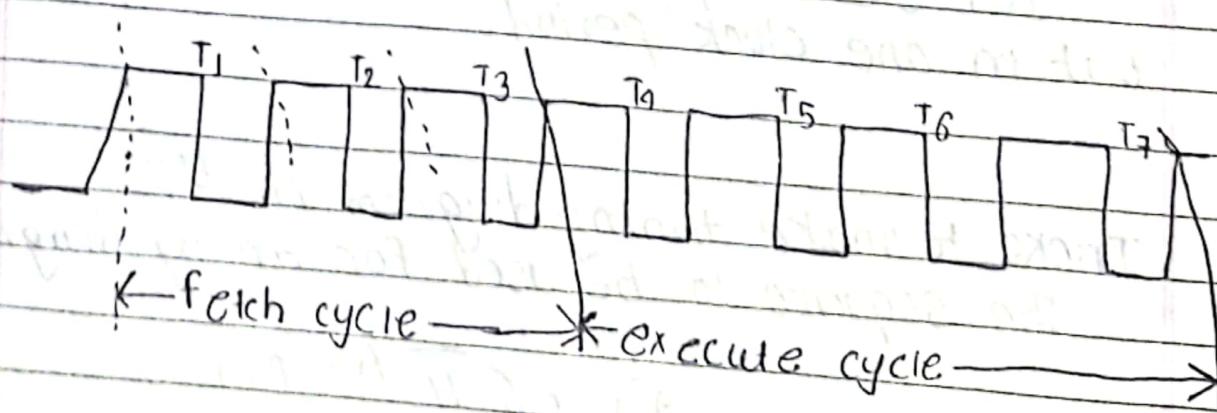
It is the total process of fetching an instruction from the memory, decoding the instruction and then finally executing the instruction.

The memory necessary steps that the CPU carries out to fetch an instruction and necessary data from the memory and to execute it constitute an instruction cycle. Moreover, it is defined as the time required to complete the execution of an instruction.

An instruction cycle consists of fetch cycle and execute cycle. In fetch cycle, CPU fetches opcode from the memory. The necessary steps which are carried out to get data if any from the memory and to perform the specific operation specified in an instruction constitute an execute cycle. The total time required to execute an instruction is given by

$$T_C = F_C + E_C$$

The 8085 consists of 1-6 machine cycle or operations.



## Machine cycle:

It is defined as the time required to complete one operation of accessing memory, input output or acknowledging an external request. This cycle may consist of 3 to 6 T - States.

## T - States:

It is defined as one sub division of the operation performed in one clock period. These sub division are internal states synchronized with system clock and each T - States precisely equal to one clock period.

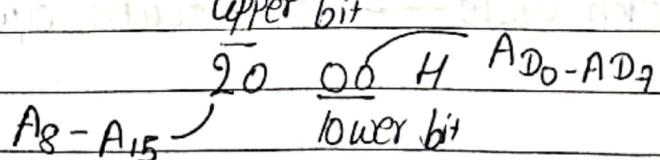
## Timing diagram:

It is the graphical representation of execution time taken by the processor for any instruction. It represents specific instruction cycle in terms of machine cycles and T - States.

1 T - State represents the operation carried out in one clock period.

Tricks to make timing diagram in UP.

The sequence to be used for timing diagram



1. A<sub>15</sub> - A<sub>8</sub>

2.  $AD_7 - AD_0$
3. ALE
4.  $IO/M$
5.  $\overline{RD}$  (low active)
6.  $\overline{WR}$  (low active)

### 1. OPCode fetch machine cycle:

The first operation in any instruction is OP-code fetch. The microprocessor needs to get (fetch) this machine code from the memory register where it is stored before the microprocessor can begin to execute the instruction.

**Step 1:** Microprocessor places the 16 bit memory address from program counter on the address bus. At  $T_1$ , high order address ( $9_0$ ) is placed at  $A_8 - A_{15}$  and lower order address ( $0_5$ ) is placed at  $AD_0 - AD_7$ . ALE signal goes high.  $IO/M$  goes low and both S<sub>1</sub> and S<sub>0</sub> goes high for UP OPCode fetch.

**Step 2:** The control unit sends the control signal RD to enable the memory chip and active during  $T_2$  and  $T_3$ .

**Step 3:** The byte from memory location is placed on the data bus, that is  $4F$  into  $D_0 - D_7$  and RD goes high impedance.

**Step 4:** The instruction  $4FH$  is decoded and content of accumulator will be copied into register C during clock  $T_4$  cycle  $T_4$ .

Signal	$T_1$	$T_2$	$T_3$	$T_4$
clock				
$A_{15} - A_3$	X Higher order memory		Address (A1)	X Unspecified
$A_{D7} - A_{D0}$	X lower order memory address	opcode (D7-D0)		X -----
$I_{O/M}, S_1, S_0$	X -----	X -----	X -----	X -----
$R_D$				

Verbalized ALU timing diagram for opcode fetch machine.  
Read & write at same time and write back.

location is placed on the data bus with ALE low.

3. The data is reached at processor register during  $T_3$  state when data is arrived the  $\overline{RD}$  signal goes high. It causes the bus to go into high impedance state.

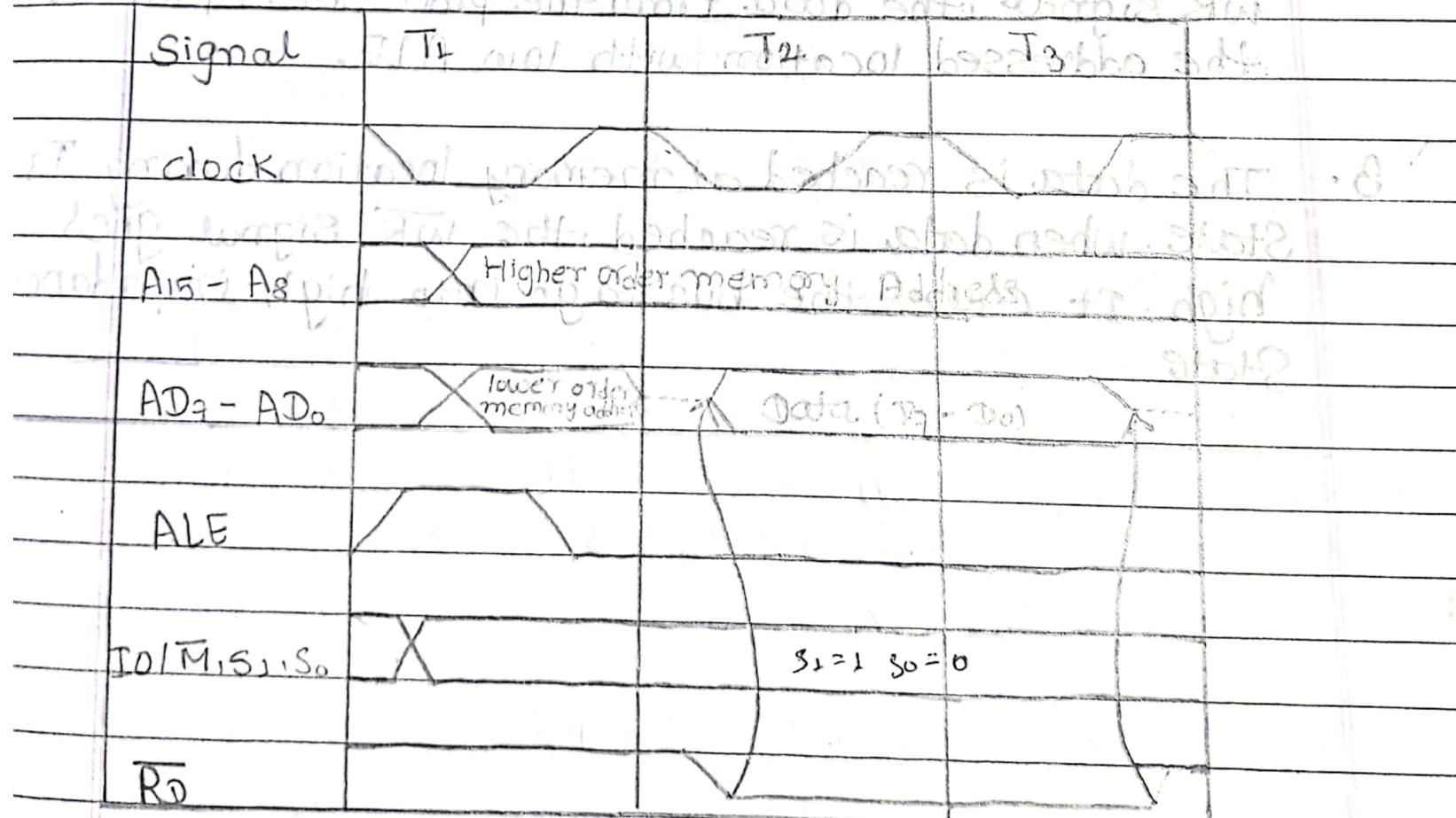


fig: Timing diagram of memory Read cycle

### 3. Memory write cycle:

The memory write cycle timing diagram can be explained as below.

1. The CPU places the 16-bit memory address from the program counter on address bus. At time period  $T_1$ , the higher order memory address is placed on the address lines  $A_{15} - A_8$ . When ALE is high the lower address is placed on the bus  $A_{D7} - A_{D0}$ . The status signal  $IOL\bar{M}$  goes low indicating the memory operation and two

Status signals  $S_1 = 0$ ,  $S_0 = 1$  to indicate memory write operation.

At time  $T_2$ , the CPU sends WR control line to enable the memory write. When memory is enabled with WR signal, the data from the processor is placed on the addressed location with low ALE.

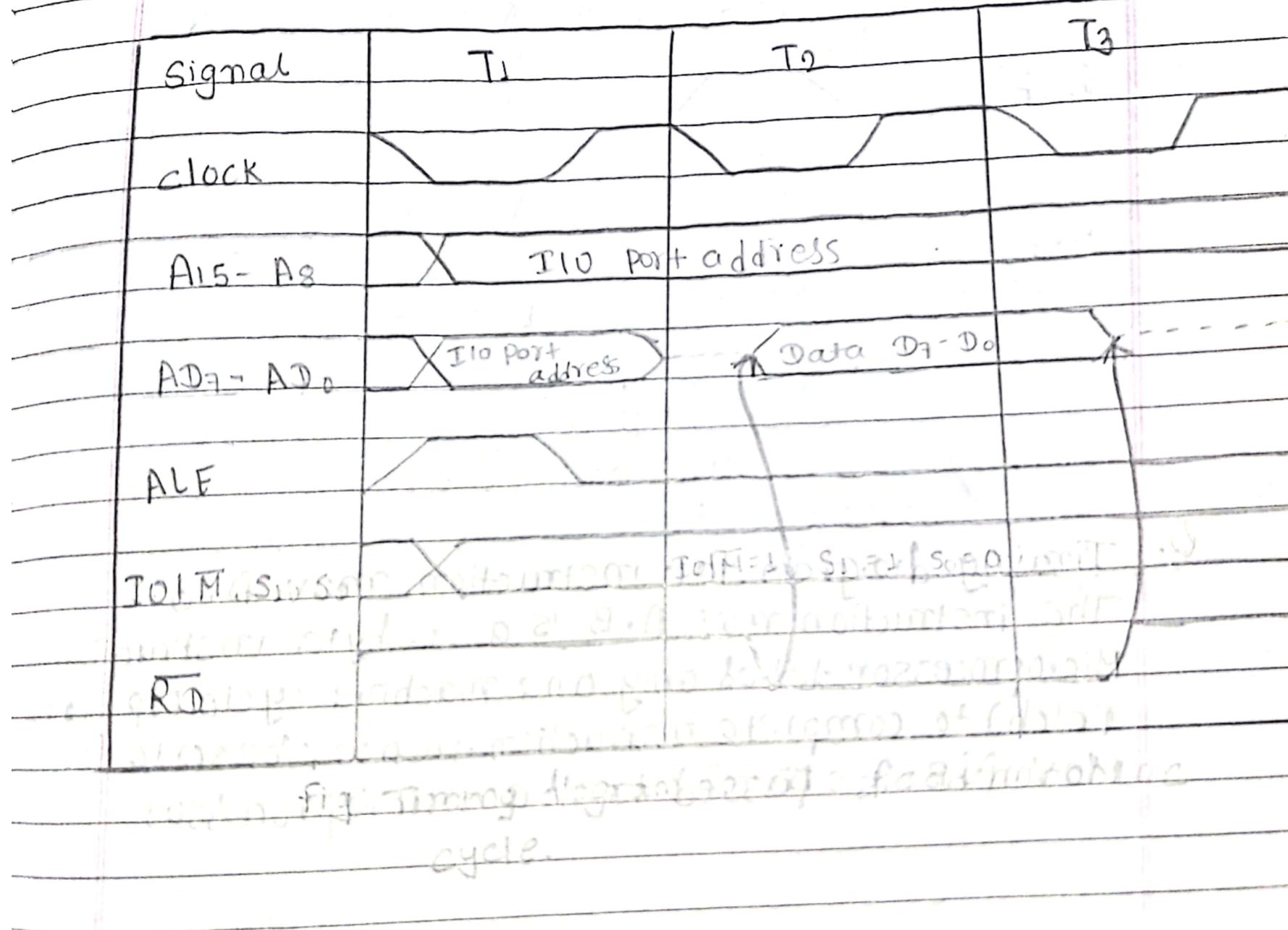
The data is reached at memory location during  $T_3$  State. When data is reached, the WR signal goes high. It causes the bus to go into high impedance state.

Signal	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>
clock			
A <sub>15</sub> - A <sub>8</sub>	X	Higher order Address	
A <sub>D<sub>7</sub></sub> - A <sub>D<sub>0</sub></sub>	X	lower order Address	Data (D <sub>7</sub> - D <sub>0</sub> )
ALE			
I <sub>O/M</sub> , S <sub>1</sub> , S <sub>0</sub>	X		
WR			

Fig: Timing diagram for memory write machine  
 Address & control signals: A<sub>15</sub> - A<sub>0</sub>, ALE, I<sub>O/M</sub>, S<sub>1</sub>, S<sub>0</sub>, WR  
 Data: D<sub>7</sub> - D<sub>0</sub>  
 Clock: clock  
 Address: A<sub>15</sub> - A<sub>8</sub> (Higher order address), A<sub>D<sub>7</sub></sub> - A<sub>D<sub>0</sub></sub> (lower order address)  
 Control: I<sub>O/M</sub> (Memory write), S<sub>1</sub> (Byte enable), S<sub>0</sub> (Write strobe), WR (Write enable)

Note: The I<sub>O</sub> read and I<sub>O</sub> write machine cycle are similar to the memory read and memory write machine cycles respectively except that the I<sub>O</sub>IM signal is high for I<sub>O</sub> read and I<sub>O</sub> write machine cycles. High I<sub>O</sub>IM signal indicates that it is an I<sub>O</sub> operator.

#### A. I/O Read cycle.



## 5. I/O write cycle

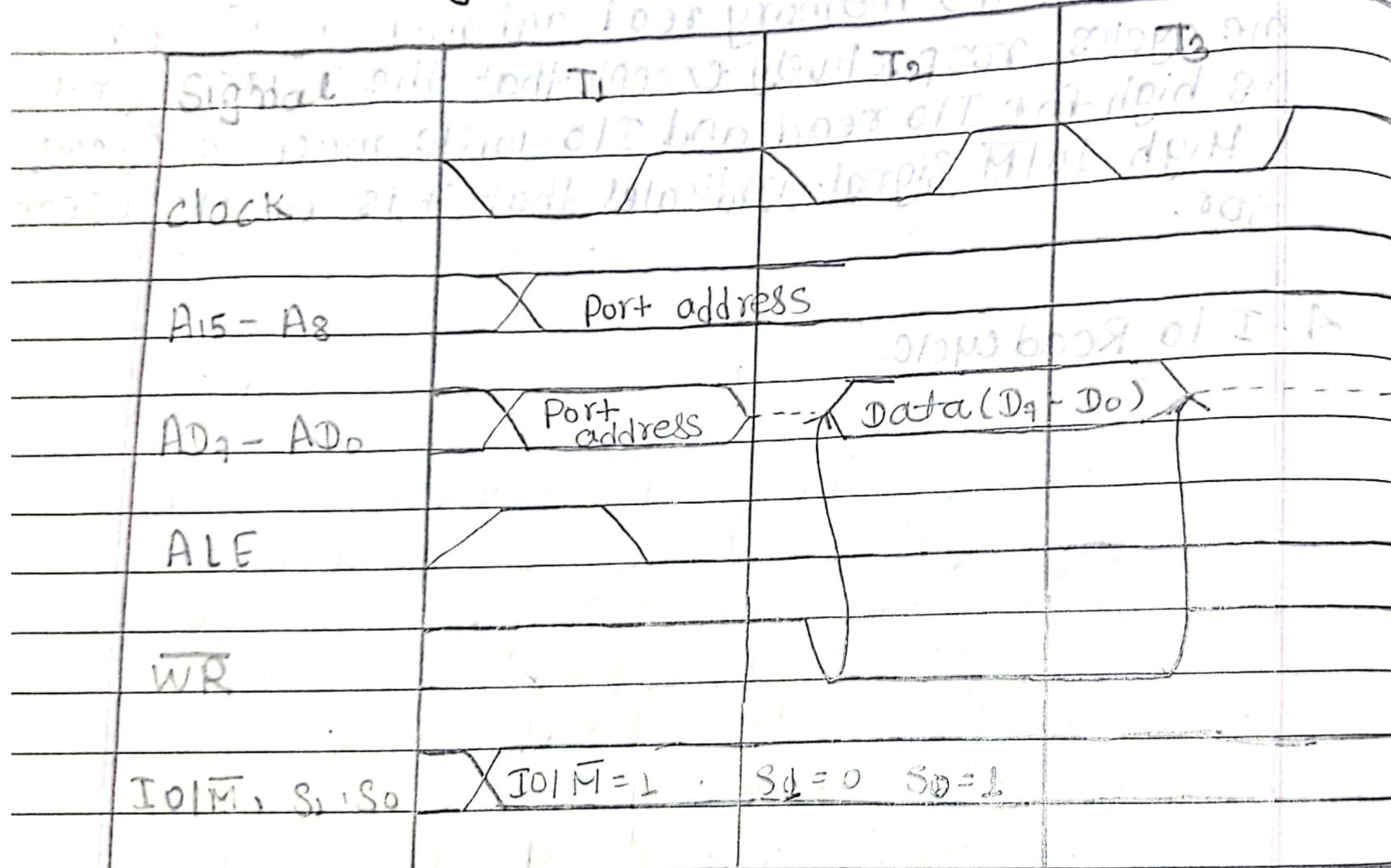


fig: Timing diagram for I/O  
write machine cycle

Timing diagram for instruction mov A,B:

The instruction mov A,B is a 1-byte instruction.

Microprocessor takes only one machine cycle (op-code fetch) to complete instruction. Hence, hexcode for MOV A,B is passed to the microprocessor.

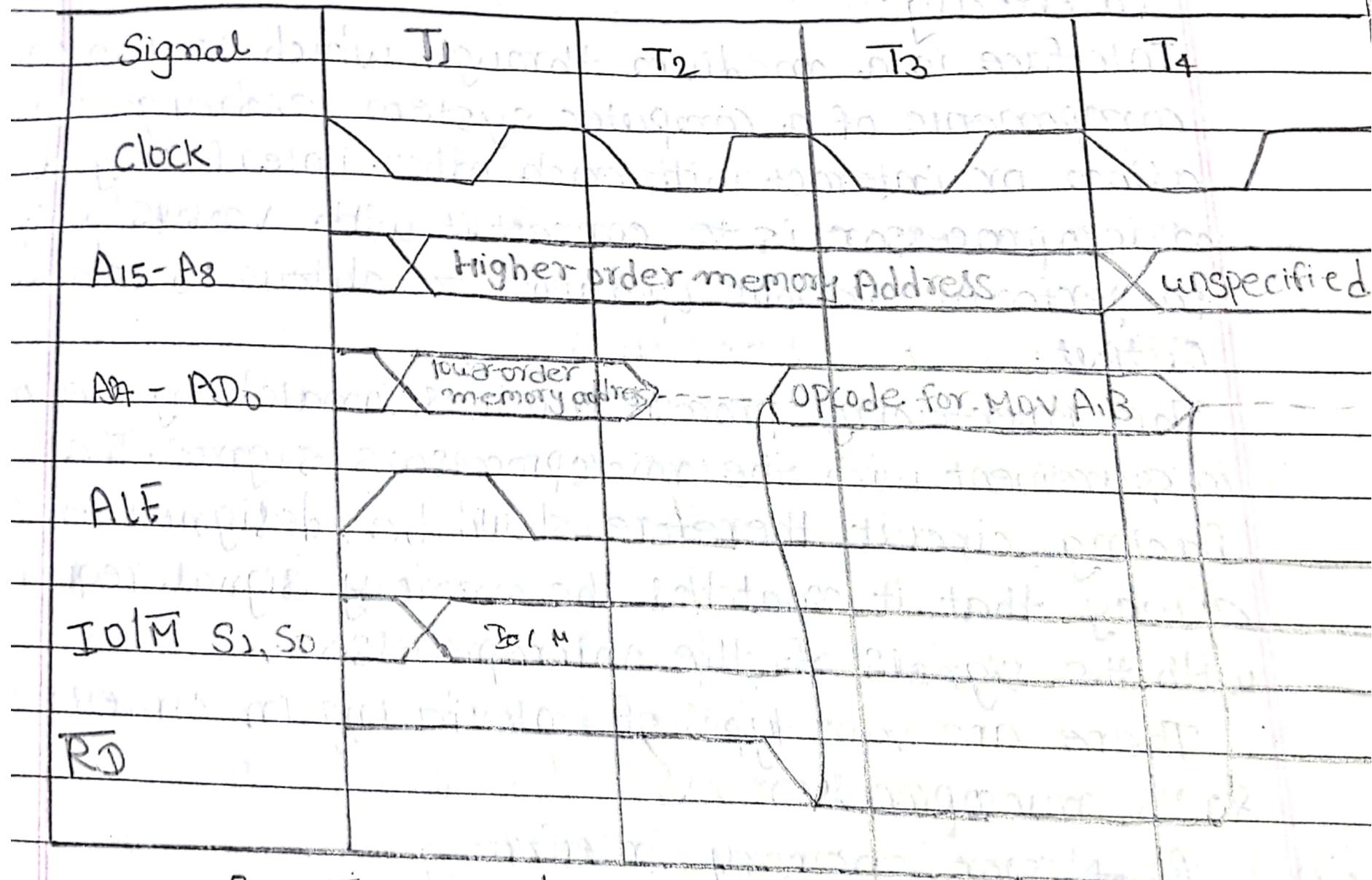


fig: Timing diagram for instruction Mov A,B