

Date: _____

Page: _____

Name : Arpan Adhikari

Roll No. : 231309

MALP, Assignment

BE Comp. (semester - 3)

~~check at 11/11/2021~~
~~11/11/2021~~
~~09/11/2021~~

- Q1) Briefly discuss the evolution of microprocessor. Also mention the important features of different generations.



The evolution of microprocessors shows how they have become faster, smarter, and more efficient over time.

1st Generation : (1970s)

- Example : Intel 4004 (1971), 4-bit processor
- Features : * very basic and slow (worked at kilohertz speed)
* could only handle simple tasks, like calculators.
* limited data processing capability.

2nd Generation : Late 1970s

- 8-bit processors, making them faster than the first generation.
- Used in early personal computers and simple control systems.
- Introduced new technology, leading to more complex operations.
- Example : Intel 8080, Zilog Z80

3rd Generation (1980s) :

- 16-bit processors allowed for more data processing and greater memory capacity.
- Used in first IBM PCs and Apple Macintosh computers.
- Enabled ~~multitasking~~ ^{complex tasks} and the growth of personal computers, workstations, and servers.
- Example: Intel Pentium Series
- Example: Intel 8086, Motorola 68000

4th Generation (1990s) :

- 32-bit processors allowed for even faster performance.
- Introduced advanced features like memory caches, making data access faster.
- Enabled multitasking and the growth of personal computers, workstations, and servers.
- Example: Intel Pentium series

5th Generation (2000s - Present) :

- 64-bit processors improved performance by handling larger amounts of data and memory.
- Multicore processors (multiple processing units) allowed computers to handle many tasks at once, making them faster and more efficient.
- Energy efficient designs, supporting things like virtualization

Date: _____

Page: _____

and integrated graphics.

- Example: Intel Core series, AMD Ryzen

6th Generation (Emerging):

- Focus on AI, ML and tasks like data processing at the edge
- Chips are now smaller, faster & more energy efficient
- Used in cutting edge technologies like self-driving cars, smart devices and AI systems.

(Q2) Differentiate between:

I) Von Neuman and Harvard Architecture

Parameters	V.N.A.	H.A.
------------	--------	------

i) Definition

Von Neuman Architecture is an ancient type of computer architecture that follows the concept of stored program computer.

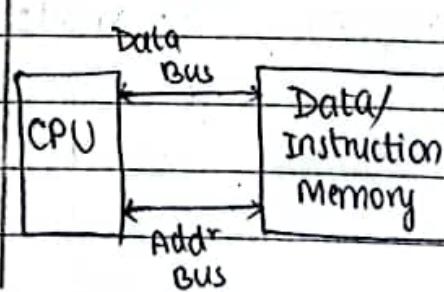
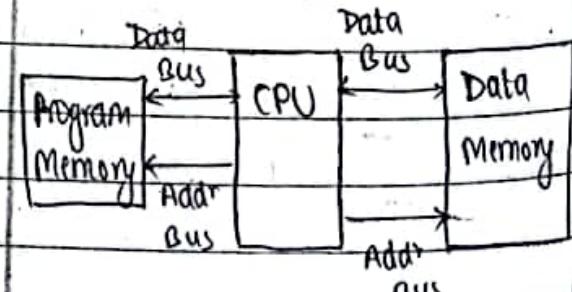
Harvard Architecture is a modern type of computer architecture that follows the concept of relay-based model.

Date: _____

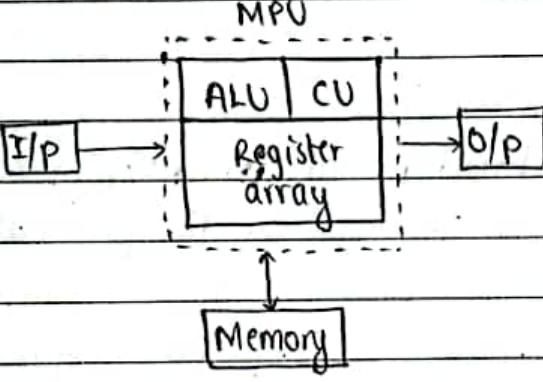
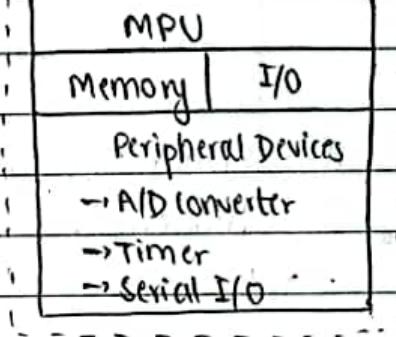
Page: _____

VNA

HA

	VNA	HA
2. Physical Address	It uses one single physical address, for accessing & storing both data & instructions.	It uses two separate physical addresses, for accessing and storing both data & instructions.
3. Buses	One common single path helps in transfer of both data and instructions.	It uses separate buses for the transfer of data and instructions.
4. No. of cycle	It requires two clock cycles for executing a single instruction.	It executes any instructions using only one clock cycle.
5. Cost	It is comparatively cheaper than H.A.	It is comparatively more expensive than V.N.A.
6. Access to CPU	The CPU is not able to read/write data & access instructions at the same time.	CPU can easily read/write data as well as access instructions at same time.
7. Block Diagram	 <pre> graph LR CPU[CPU] <-- Data Bus --> DI[Data/ Instruction Memory] CPU <-- Address Bus --> DI </pre>	 <pre> graph LR PM[Program Memory] <-- Data Bus --> CPU PM <-- Address Bus --> CPU CPU <-- Data Bus --> DM[Data Memory] CPU <-- Address Bus --> DM </pre>

2) Microprocessor and Microcontroller

Parameter	Microprocessor	Microcontroller
1. Definition	Microprocessor is a silicon chip which includes ALU, register circuit and control circuit.	Microcontroller is a silicon chip which includes microprocessor, memory, I/O and other peripheral devices.
2. Block Diagram	 <pre> graph LR IIP[I/P] --> ALUCU[ALU CU] Memory --> ALUCU RA[Register array] --> ALUCU ALUCU --> OIP[O/P] </pre>	 <pre> graph LR IIP[I/P] --> MPU[MPU] MPU --> OIP[O/P] subgraph MPU Memory PD[Peripheral Devices] PD --> AID[A/D converter] PD --> T[Timer] PD --> SI[Serial I/O] end </pre>
3. Usage	Normally used for general purpose computers as CPU.	Normally used for specific purposes (embedded system) e.g. Traffic light controller
4. Performance Speed	The performance speed i.e. clock speed of MP is higher ranging frequency from Megahertz to Gigahertz.	The performance speed of μ-controller is relatively slower than that of MP, with clock speed from 3 to 33 MHz.

Date: _____
 Page: _____

5. Size & Cost	Addition of external memory and I/O makes this system bulkier & much more expensive.	It has fined memory & all peripherals are embedded together, on a single chip, so are not bulkier & are cheaper than MP.
6. Example	e.g. INTEL 8085, INTEL 8086, Motorola, Intel core i7 etc.	e.g. AT 89C51, AT mega 32, AT 89S52 etc.

(Q3) Enlist the greatest breakthrough in microprocessor so that modern microprocessor are available for personal computer.



The development of modern microprocessors for personal computers has been shaped by several groundbreaking advancements. Some greatest breakthrough are:

1. Integrated circuits (1958)

Allowed many transistors to be packed on a single chip, making processors smaller and faster.

2. First Microprocessor (1971)

Intel 4004 introduced the idea of a small general purpose CPU.

Date: _____

Page: _____

3. 8-bit Processors (1974):
Enabled early personal computers like the Altair 8800.
4. 16-bit Processors (1978):
Intel 8086 created the x86 architecture, now standard in PCs.
5. 32-bit Processors (1985):
Allowed multitasking and more memory, supporting modern OS.
6. RISC Architecture (1980s):
Simplified processor design for speed and efficiency, used in many devices.
7. Multi-core Processors (2000s):
Multiple cores in one chip improved speed and multitasking.
8. 64-bit Processors (2003):
Enabled PCs to use more memory and handle larger tasks.
9. Integrated GPUs (2000):
Combined graphics and processing on one chip, saving space and power.
10. 3D Transistors (2011):
Improved speed and energy efficiency with advanced transistor designs.
11. AI-optimized chips (2020s):
Designed for modern tasks like AI and machine learning.

(Q4) Draw and explain the architecture of micro computer architecture.

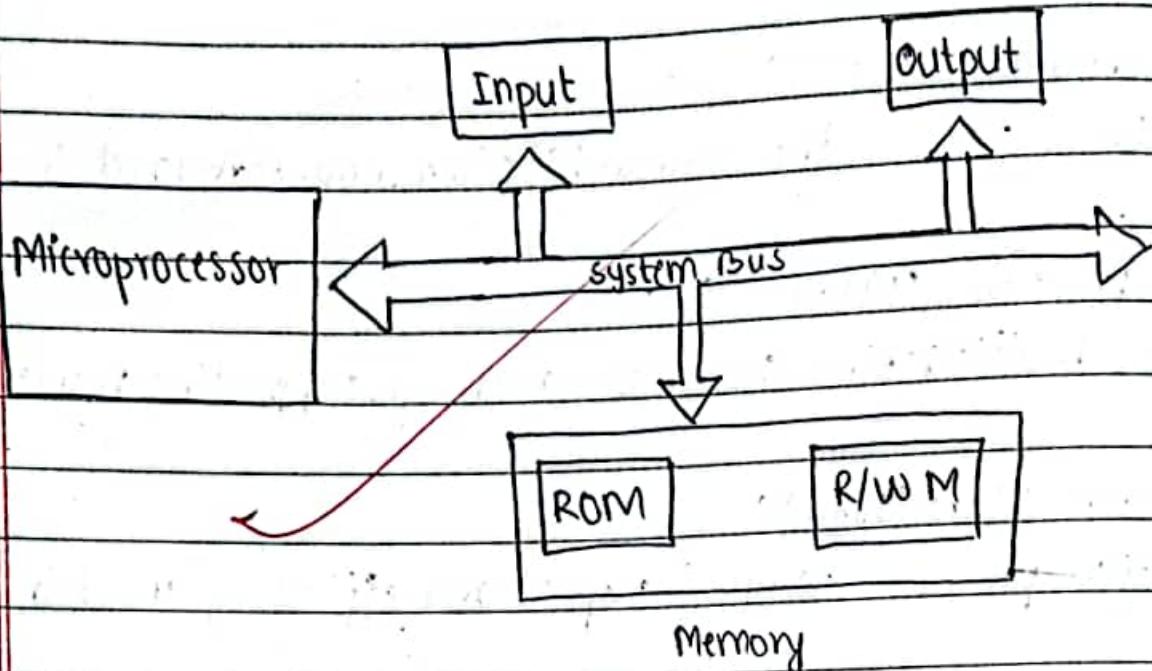


Fig. Block diagram of microcomputer

1. Micro IT includes four components : microprocessor, input, output and memory.

1. Microprocessor

A microprocessor is a general purpose processing unit built into one chip. It is the heart of microcomputer that

- executes instructions of the program and processes data.
- is responsible for performing all arithmetic and logical operations.
- controls overall system operation.

The microprocessor can be divided into three segments :

Arithmetic/Logic Unit (ALU), Register Unit & Control Unit.

Date: _____

Page: _____

2. Input Device

Used to transfer data binary from the outside world to the μp. It includes devices such as keyboards, analog-to-digital converters.

3. Output Device

Used to transfer data from the μp to output devices such as light emitting diodes (LEDs), printers.

4. Memory

Used to store binary information such as instructions and data, and provides information to μp whenever necessary.

The memory has two types:

- * ROM : This type can store program single time and read it many times. Programs stored in ROM can only be read, they cannot be altered.
- * R/W M or RAM : It is used to store user programs and data. The information stored in this memory can be read and altered easily.

5. System Bus

The system bus is a communication path between the μp and peripherals. It is a group of wire that carries bits.

(Q5) Explain about working principle of microprocessor with suitable example.

→

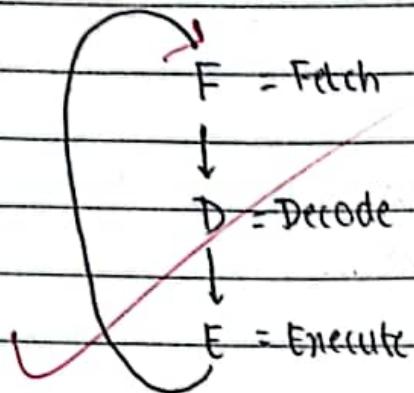


Fig. FDE cycle

The microprocessor operates using Fetch-Decode-Execute cycle, where it processes instructions stored in memory:

1. Fetch

The microprocessor retrieves the next instructions from memory using the Program Counter (PC), which holds the address of the instruction. The instruction is then loaded into the Instruction Register (IR), and the PC is incremented to point to the next instruction.

2. Decode

The control unit decodes the fetched instruction to determine the type of operation. It identifies the operands (data) and their locations (in registers or memory).

3. Execute

This instruction is executed by the appropriate microprocessor unit, such as the Arithmetic Logic Unit (ALU) for mathematical operations or data transfer units for memory access. Results are stored in a register, memory, or output device as needed.

~~Example: Adding Two Numbers~~

1. The ADD instruction is fetched from memory.
2. The instruction is decoded to identify the operation (addition) and operands (5 and 7).
3. The ALU adds the values $5+7=12$, & result is stored in register or memory.

Q6) Explain the concept of bus system in a computer system with suitable diagram.

→

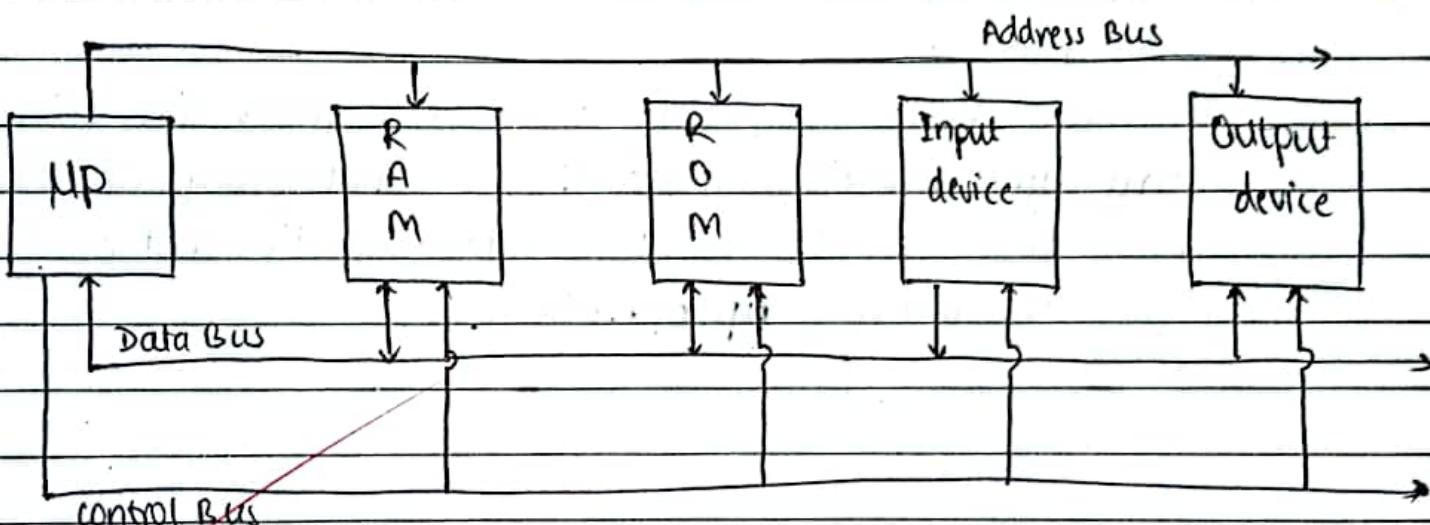


Fig. Bus Systems in Computer system

1. Data Bus

The databus provides a path for monitoring data between system modules. It consists of number of separate lines generally 8, 16, 32 or 64. The no. of lines is referred to as the width of data bus. It is bidirectional bus for data transfer to and from CPU. For e.g. In 8085 CPU, databus is of 8-bit.

2. Address Bus

It is unidirectional bus from CPU to peripherals. Address bus is used to transfer address from microprocessor to peripherals. It consists of no. of separate lines that are used to designate the source or destination of the data on data bus. For e.g. A microprocessor with 16-bit address bus can access upto 2^{16} diff. addresses.

3. Control Bus

The control bus is composed of various signals lines that carry synchronized signals. These are not group of lines like address or data bus, but individual line that provides a pulse to indicate CPU operations.

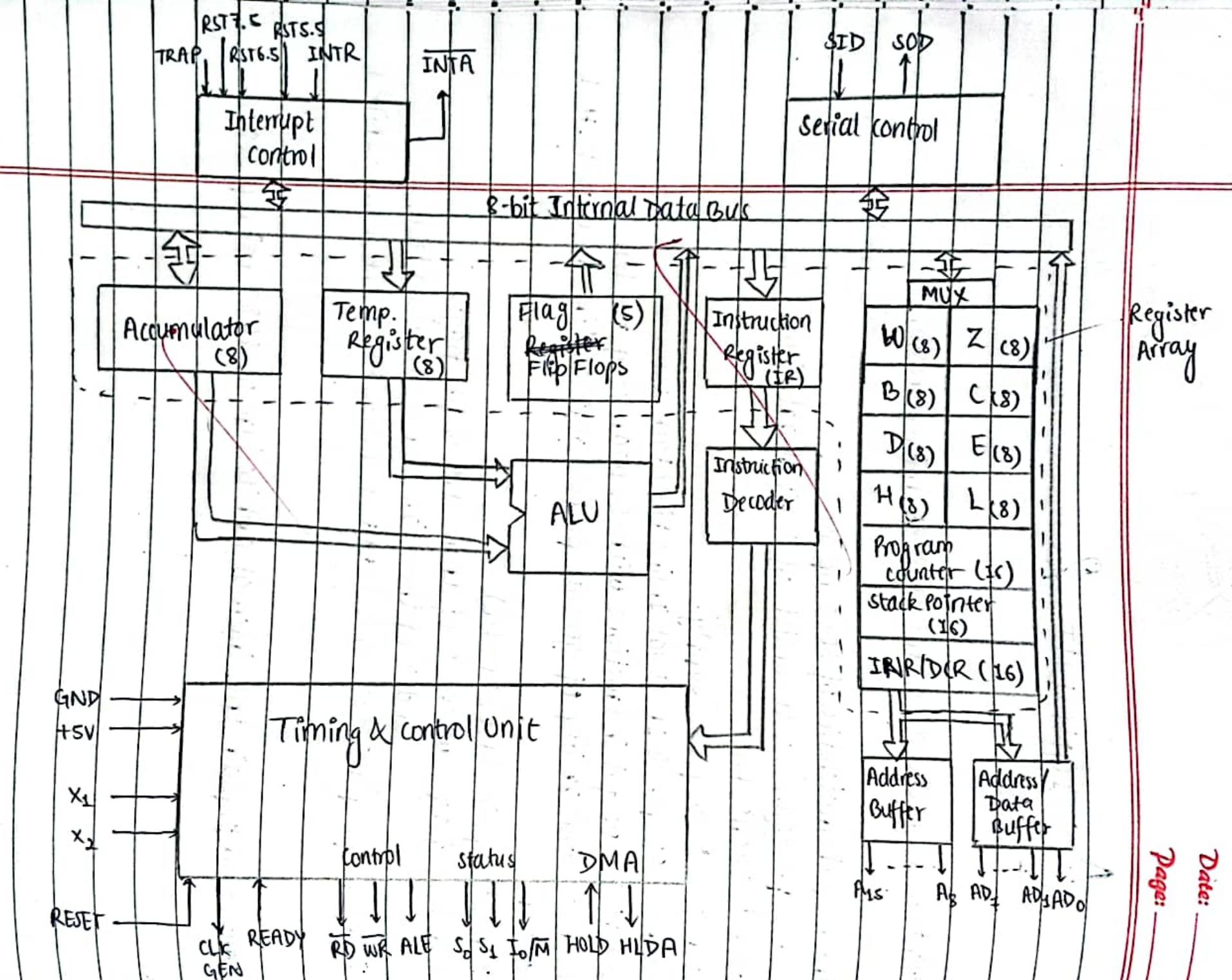
ASSIGNMENT - 2

I) Mention key characteristics of 8085 microprocessor.

The key characteristics of 8085 μp are:

1. 8-bit processor i.e. It consists of 8-bit of data bus.
2. 16-bit address bus
3. It needs +5V power supply.
4. It has 40 pin DIP (Dual Inline Pins/ Package).
5. It has 74 useful instructions with 5 addressing modes.
6. It has availability of general purpose registers like B,C,D,E,H,L and A.
7. 3-register pairs (BC, DE, HL)
8. 8-bit of flags where 5 of them are useful flags.
and 3 of them are don't care conditions.
9. 5 interrupt pins available (TRAP, RST 7.5, RST 6.5, RST 5.5, INTR)

Q2) Draw and explain the internal architecture of 8085 microprocessor.



Date: _____
Page: _____

The internal architecture of 8085 microprocessor consists of following components:

1. ALU

It performs the computing function which include:

- Accumulator (A) : 8 bit special purpose register used in arithmetic, logic, load and store operations
- Temporary register : 8 bit register used to hold data temporarily during program execution & is not accessible to programmer
- It consists of arithmetic and logical circuits.
- 5 flag flip-flop registers are also associated.
- The bit position reserve for different flags in flag register array is shown below:

S	Z	X	AC	X	P	X	CY	
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	

Here, X = Don't care conditions

S = Sign flag (For +ve number, S=0
and For -ve number, S=1)

Z = Zero flag (If result = 0, Z=1 (set)
If result ≠ 0, Z=0 (reset))

AC = Auxiliary Carry flag (set when carry from lower nibble)

P = Parity flag (P=1 (set) for even parity &

P=0 (reset) for odd parity)

CY = Carry flag (CY = 1 (set) and CY = 0 (reset))

2. Register Array

- 8085 has 8 and 16 bit registers.
- 8 bit registers are B, C, H, L, accumulator and flag.
- There are two 16 bit registers:

Program Counter (PC)

Stack Pointer (SP)

- 8 bit general purpose registers can be used as 8-bit or 16-bit register pair.

3. Timing & control Unit

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

4. Instruction Register and Decoder

When instruction is fetched from memory, it is loaded in the IR. Then the decoder decodes the instruction and establish a sequence of events to follow. The instruction register is not programmable and cannot access through any instructions.

5. Interrupt control

8085 has 5 interrupt signals i.e. TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR.

INTA is Interrupt Acknowledgement center. Valid interrupt may occur for at least 160 nsec.

6. serial I/O O/P control

- It has two pins i.e. SID (serial Input Data), SOD
- Data transfer is controlled through two instructions.
 - (i) SIM (Set Interrupt Mask) → Used for SOD
 - (ii) RIM (Read Interrupt Mask) → Used for SID

(Q3) Define instruction. Explain the types of instructions on the basis of byte size with suitable example.

→ An instruction is a command given to a computer processor to perform a specific operation. It is a part of the machine code or assembly language that dictates tasks such as arithmetic operations, data transfer, or logical operations.

Types:

(i) 1-Byte Instructions

- They occupy a single bit (8-bits) in the memory.
- Usually perform simple operations like register-to-register or accumulator-based operations.
- May not include an operand; the operation may inherently affect specific registers or flags.
- Example (8085):

HLT // Terminates the program

(iii) 2-Byte Instructions

- They occupy two bytes (16 bits) in memory.
 - Include one opcode and one immediate operand or address.
 - The first byte specifies the operation, and the second byte contains the operand.
- E.g. MVI A, 05H //Move data 05H to register A

(iv) 3-Byte Instructions

- They occupy 3 bytes (24 bits) in memory.
 - Include an opcode and 16-bit address or data
 - The first byte is the opcode, and the next two bytes represent the lower and higher bits of the address or data.
- E.g. LDA 2050H //Load value from memory address 2050H to accumulator

Q4) Explain instructions sets available in 8085 microprocessor with suitable examples.



Following are the instructions sets available in 8085 microprocessor:

1. Data transfer Instructions

These move data between registers, memory and I/O.

Examples:

* MOV A,B //copy data from B to A

* MVI A, 05H // copy 05H data into A
 * LDA 2000H // Loads data from address 2000H into accumulator

~~2. Arithmetic Instructions~~

These perform mathematical operations.

Examples:

* ADD B // Adds content of B with accumulator and store the result in A.
 * SUB C // subtracts content of C from accumulator
 * INR A // Increment the value of register A by 1.

~~3. Logical Instructions~~

These perform logical operations.

Examples:

* ANA R // Logically 'AND' the content of specified register with the content of A and stores the final result in A.
 * XRI 8-bitdata // Exclusive ORing the content of A and immediate data and stores in A.

~~4. Branch Instructions~~

These alter the program flow.

Example:

* TZ 16-bit address.
 // Jump to specific memory address if O flag is set or result is zero.

* JC 16-bitaddr

//Jump to specific memory address if carry flag is set or carry occur.

S. Control Instructions

These control the program execution.

Example :

* HLT //Terminate the program

* NOP //No operation

(Q5) Define addressing mode. Explain different types of addressing modes available in 8085 microprocessor.

The various ways in which the processor can access data during program execution are referred to as its addressing modes.

The different types of addressing modes are :

1. Immediate Addressing Mode

- The data is present in the instruction itself.
- Operands can be of 8 or 16 bits in size.

For e.g.

* MVI A, 32H ; loads the 8-bit data (i.e. 32H) immediately into accumulator.

* ADI 20H ; Adds the content of A and 20H immediately.

Date: _____

Page: _____

2. Direct Addressing Mode

In this addressing mode, the effective address of memory for I/O port/device is defined in the instructions.

For e.g.:

* LDA 2050H ; loads the content of memory location 2050H into accumulator register.

3. Register Addressing Mode

In this, registers are used to define data i.e. both source and destination.

For e.g.:

MOV A,B // Copies/move the content of register B to accumulator

4. Register Indirect Addressing Mode

In this addressing mode, the register pair are used as memory address of data.

For e.g.:

LDAX B // Loads the data byte into A from the memory specified by the address in register pair BC.

5. Implied/Hidden Addressing Mode

In this addressing mode, the operand for the instruction is hidden and it is normally the accumulator.

For e.g.:

CMA // complements the value of accumulator

- Q6) Define T-state. Draw well labelled timing diagram of following instructions.

~~T states are the basic clock cycles in a microprocessor representing the smallest unit of time for its operations; like fetching, decoding or executing instruction.~~

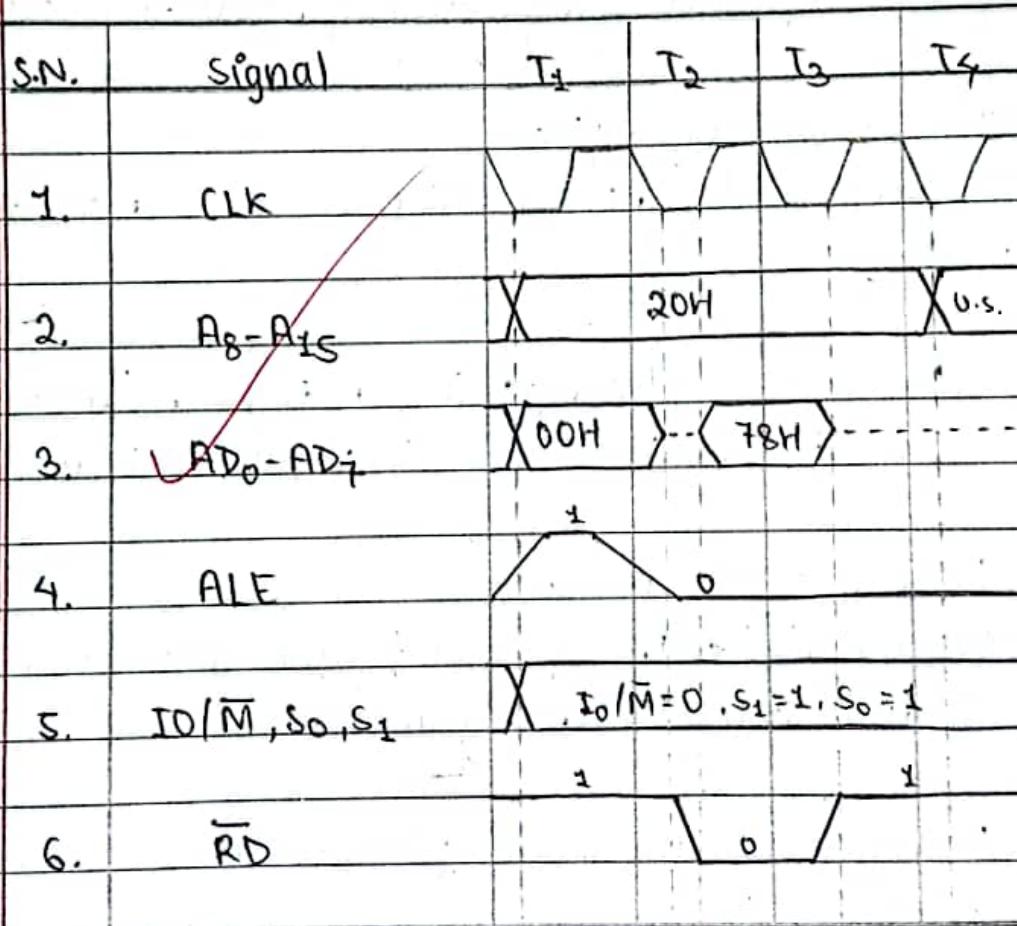
A. MOV C,D

Assume memory address as,

2000H	78H
-------	-----

MOV C,D instruction takes 0.F. as machine cycle.

So, Total No. of T-states = 4T



Date: _____

Page: _____

B MVI, C, AB H

Assume memory address as,

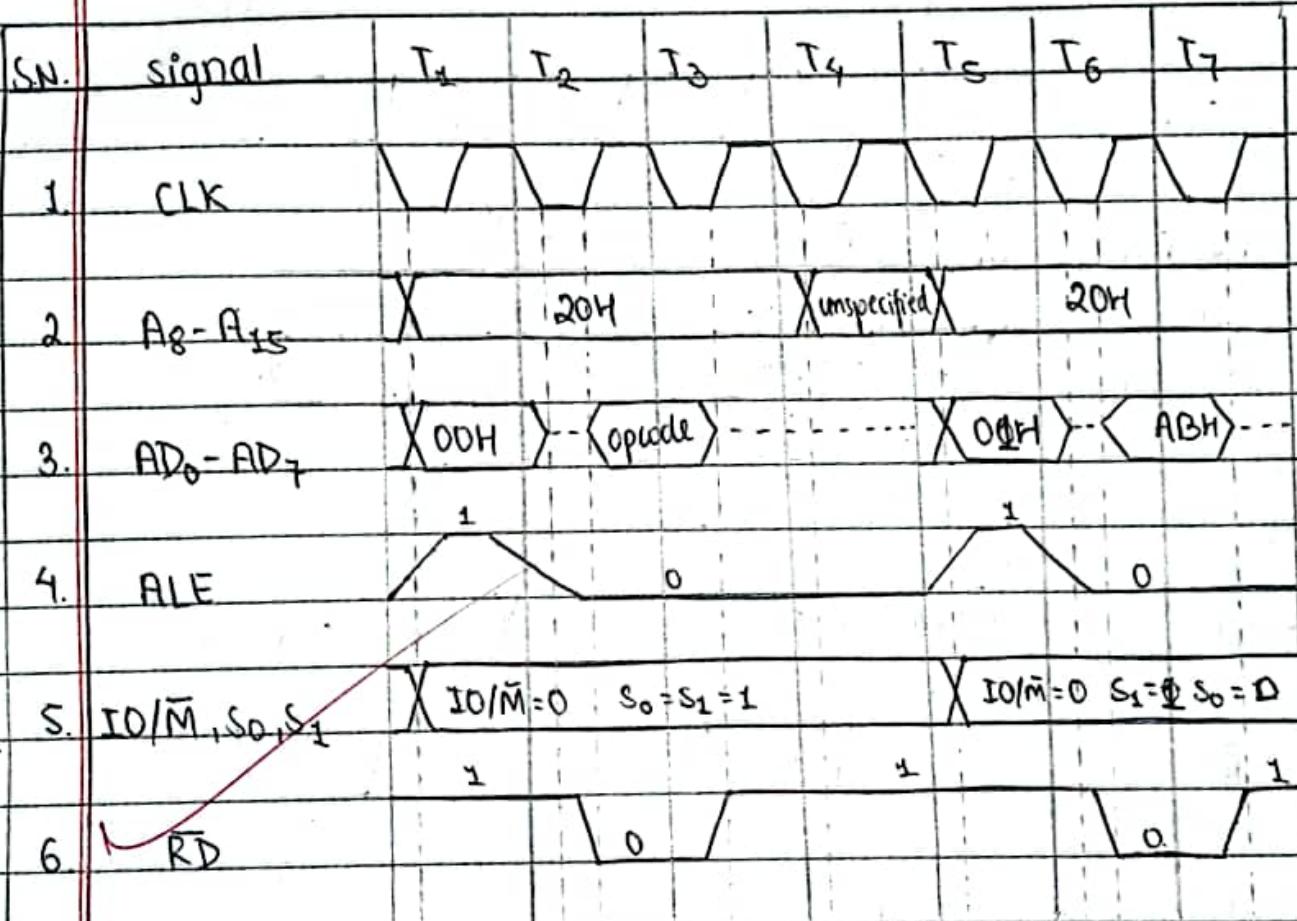
2000H	300p-wd
2001H	ABH

MVI instruction takes O.F. and M.R as machine cycles.

So, Total no. of T-states = 7T

; opcode-fetch (O.F.) = 4T

Memory Read (M.R.) = 3T



Date: _____

Page: _____

C. MOV A,M

Assume memory address as, 2000H op-code

MOV A,M instruction takes O.F. and M.R. as machine cycles.

∴ OPCODE fetch (O.F.) = 4T

Memory Read (M.R.) = 3T

So, Total no. of T-states = 7T

S.N.	signal	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇
1.	CLK	↓	↑	↓	↑	↓	↑	↓
2.	A ₈ -A ₁₅	X	20H		X U.S. X	20H		
3.	AD ₀ -AD ₇	X 00H	→ {opcode}	---	X 00H	→ {opcode}	---	
4.	ALE	1	0		1	0		
5.	I _O /M, S ₀ , S ₁	X I _O /M=0 S ₀ =S ₁ =1			X I _O /M=0 S ₀ =0 S ₁ =1			
6.	RD		0	1	0	1	0	

where, U.S. = unspecified

Date: _____

Page: _____

D. STA 0000H

We know, STA is a 3-byte instruction.

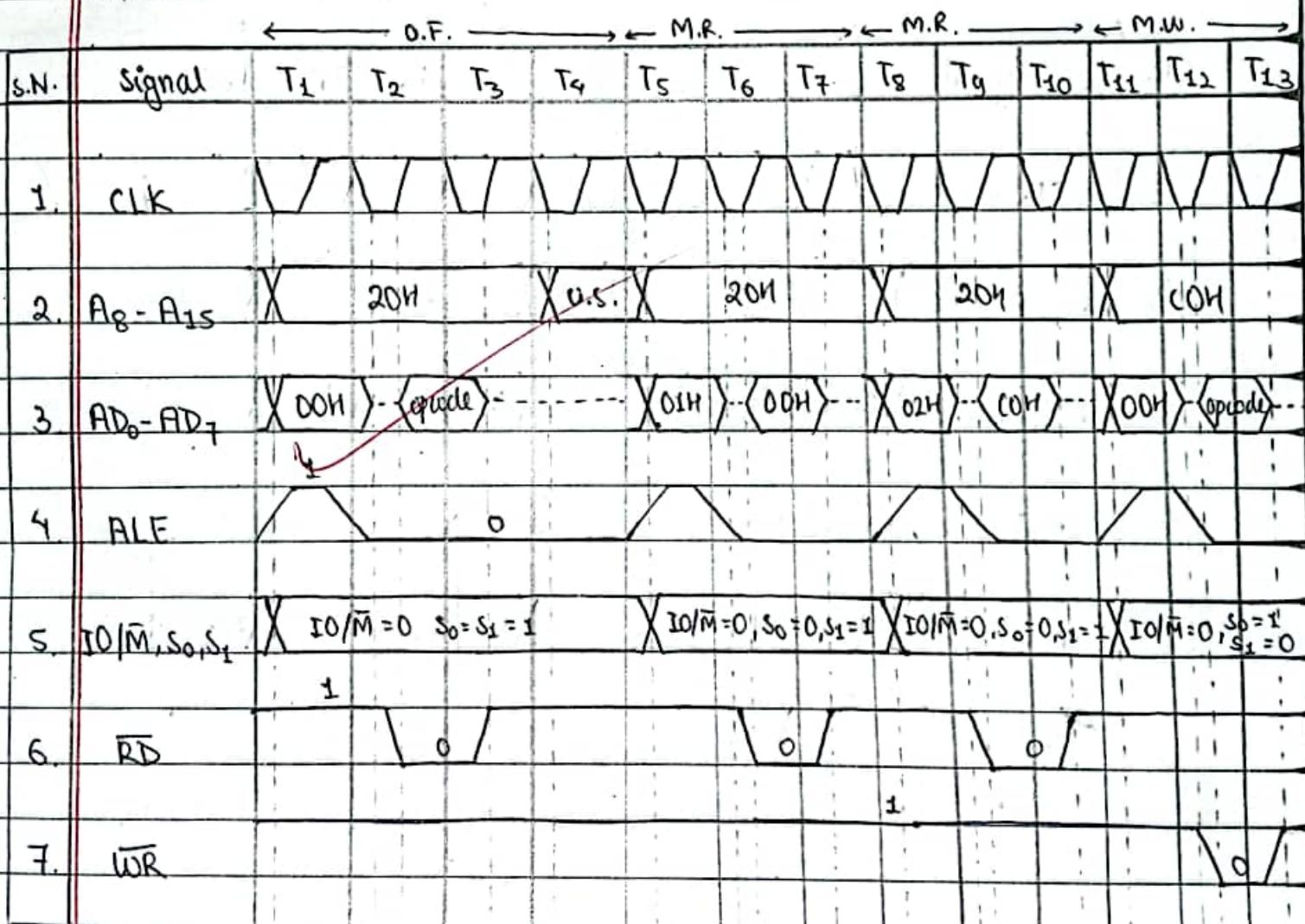
Assume memory address as, Address Hex-Code :

2000H	op-code
2001H	00H (Lower byte)
2002H	COH (Upper)

STA consists of 4-machine cycles : OPCODE fetch (4T)

* 2 Memory Read (2x3T = 6T), M.W. (3T)

$$\text{So, Total T-states} = 4T + 6T + 3T = 13T$$



Date: _____
 Page: _____

E. LXI B, 2050H

We know, LXI is a 3-byte instruction.

Assume memory address c1s, Address Hex-code

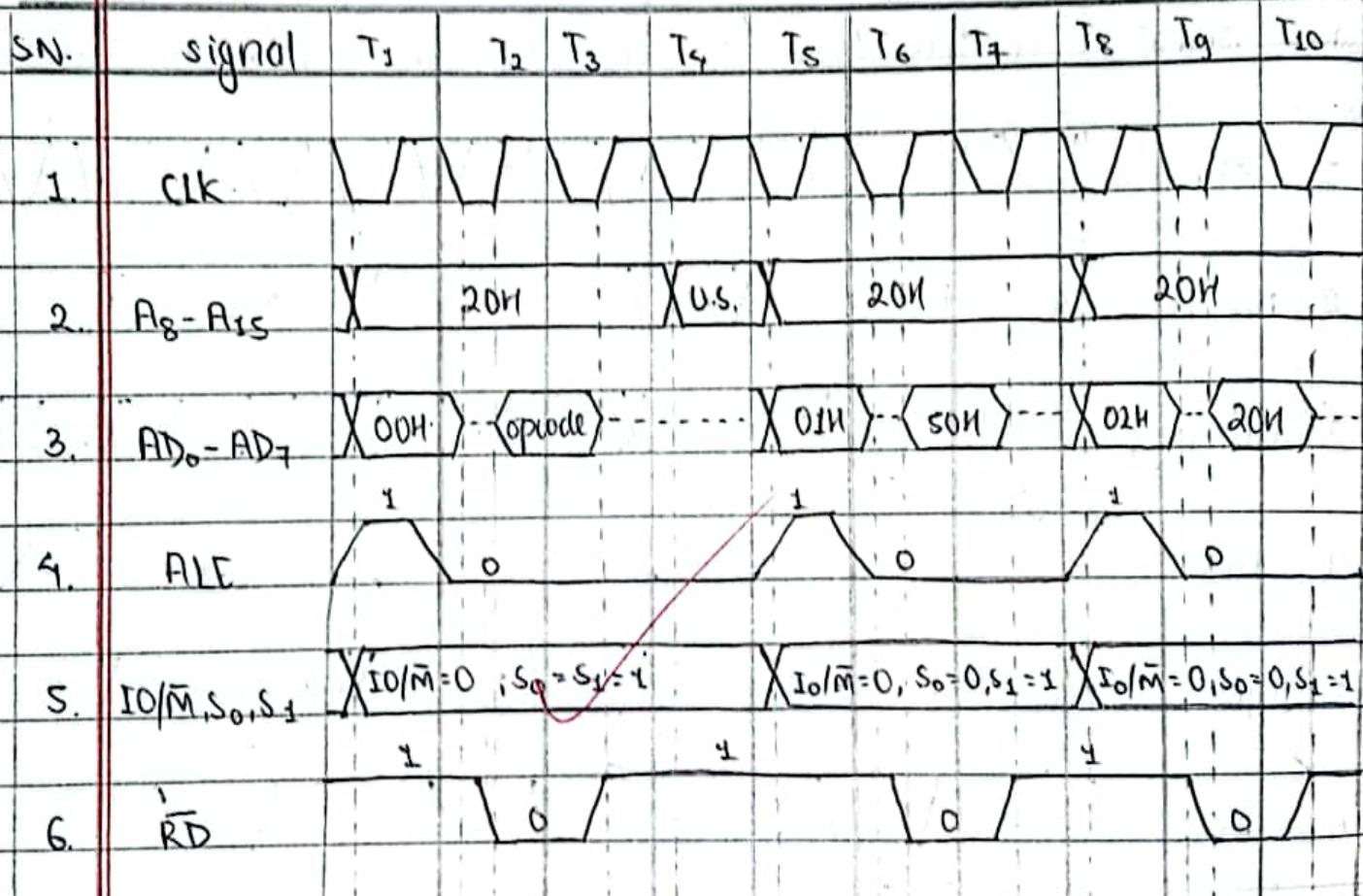
2000H	opcode
2001H	50H
2002H	20H

LXI consists of 3 machine cycles: opcode fetch (4T)

: 2 memory Read (2x3T = 6T)

∴ Total no. of T-states = 10T

← D.F. → ← M.R. → ← M.R. →



Date: _____

Page: _____

E. IN 34H

We know, IN is a 2-byte instruction.

Assume memory address as, Address Hex-code

2000H opcode

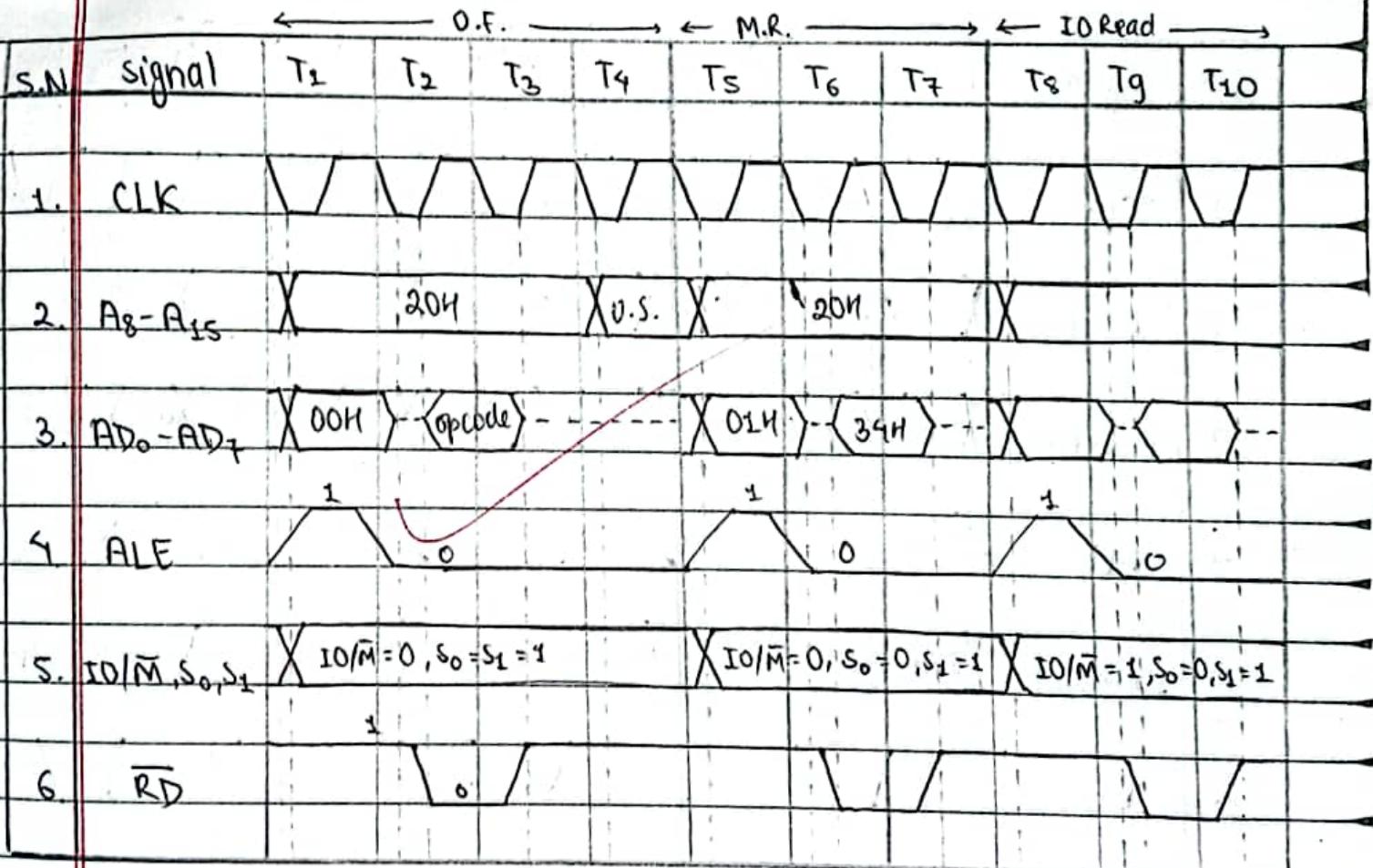
2001H 34H

IN consists of 3 machine cycles : opcode fetch (4T)

Memory Read (3T)

IO Read (3T)

∴ Total no. of T-states = 10T



G. LDA E050H

We know, LDA E050H is a 3-byte instruction.

Assume address as,

Address Hex-Code

2000H

Opcode

2001H

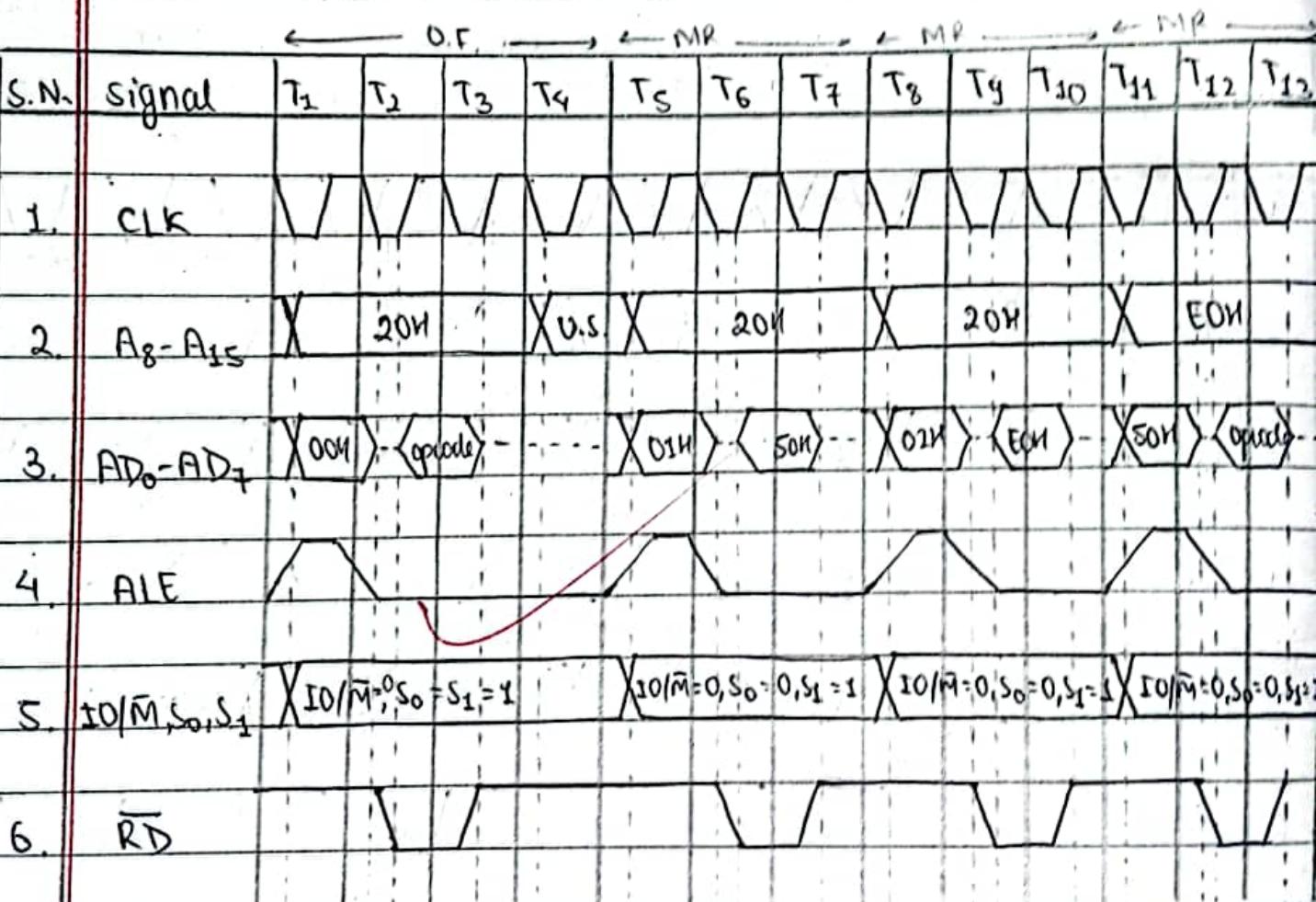
50H

2002H

EOH

LDA consists of 4 machine cycles : Opcode fetch (4T)
3 Memory Read (9T)

$$\therefore \text{Total T-states} = 13T$$



H) ACI 48H

We know, ACI 48H is a 2-byte instruction.

Assume Address as,

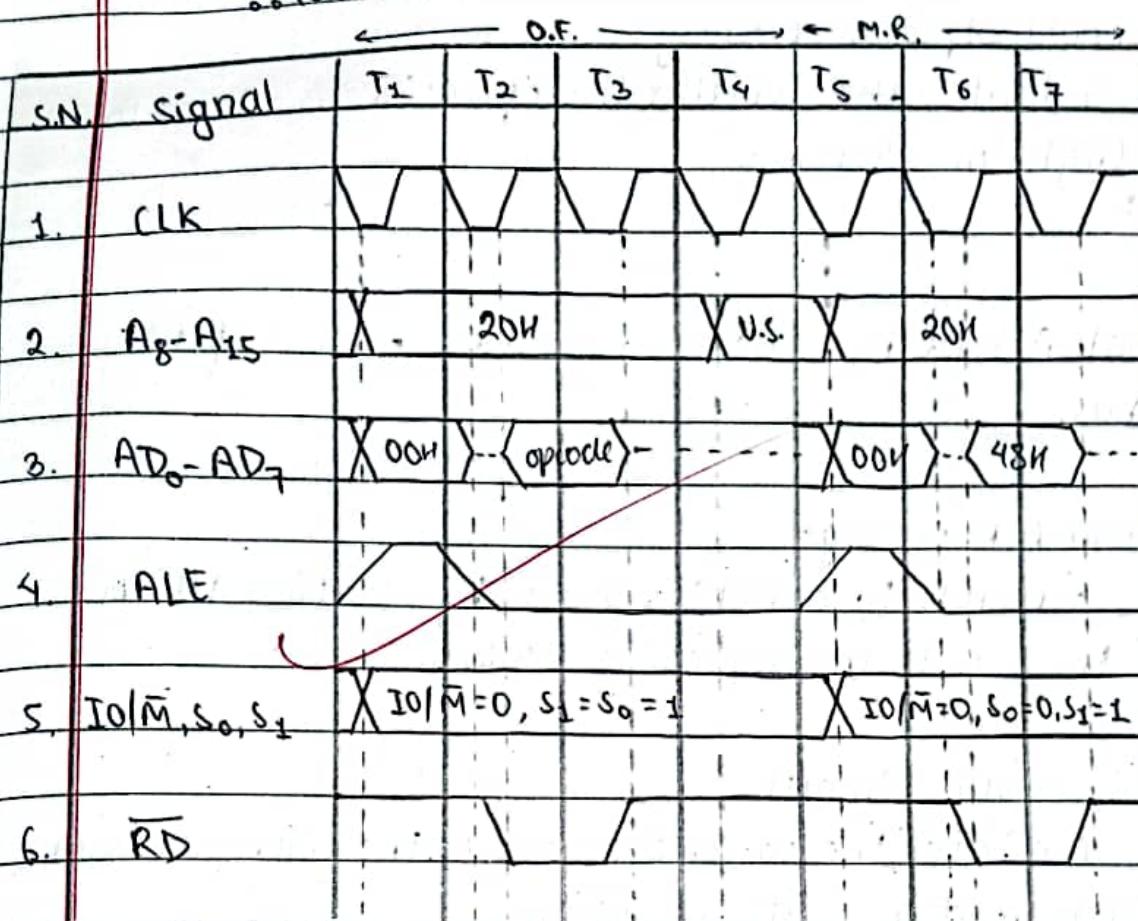
Address Hexcode

2000H opcode

2001H 48H

ACI instruction takes opcode fetch (4T) and Memory Read (3T).

$$\therefore \text{Total T-states} = 7T$$



(Q7) Explain the function of following 8085 pins.

A. ALE (Address latch Enable)

It activates during the first clock cycle to enable latching the lower byte of the address onto the address bus. This separates the address and data bus.

B. IO/M (Input/Output or Memory)

This pin indicates whether the operation is related to memory (high) or I/O (low).

C. HOLD

This signal is sent by a device requesting control of the system bus.

D. HLDA (Hold Acknowledge)

The 8085 responds to a HOLD request by asserting HLDA, granting the bus to the requesting device.

E. S₀ and S₁ (status signals)

These pins indicate the current status of the microprocessor, such as fetch, read or write operations.

(Q8) Write an ALP to in 8085 to add the content of memory location 2050H and 2051H. Store the result in memory locations 3050H and carry in 3051H.

LDA 2050H //Load content from 2050H addr
MOV B,A //Move content from A to B
LDA 2051H // Load content from 2051H addr
ADD B //Add A and B
MVI C,00H // Put empty value in Register C

ADD B //Add A and B
JNC loop //Jump to loop when no carry
INR C //Increment Content of C by 1

loop: STA 2050H //store value of A to 2050H addr
MOV A,C //Move content from C to A
STA 2051H //store content of A to 2051H addr
HLT //Haults the program

Q9) Write an ALP in 8085 to swap the content of two memory locations A050H and B050H.

→

LDA A050H

MOV B,A

LDA B050H

STA A050H

MOV A,B

STA B050H

HLT

Date: _____

Page: _____

- Q10) write an ALP to add two 16 bit numbers AB22H and CD44H and store the result in memory location starting from 3000H.



LXI B,AB22H //Loads the 16-bit no. AB22H into BC register pair
LXI H,CD44H //Loads the 16-bit CD44H to HL pair
MOV A,C //Moves lower byte of AB22H to A (i.e. 22H)
ADD L //Add lower byte of CD44H with A.
(i.e. 44H) (i.e. 22H)
STA 3000H //store result (lower byte) at 3000H
MOV A,B //Move ~~higher~~^{higher} byte of AB22H (i.e. ABH)
ADC H //Add content of A with higher byte of CD44H with carry
STA 3001H //store higher bits result in 3001
HLT //Ends program

- Q11) write an ALP in 8085 to multiply 06H and 05H and store the final result in memory location 2000H.



MVI B, 06H
MVI C, 05H
MVI A, 00H

Date: _____

Page: _____

loop: ADD B

DCR C

JNZ loop

STA 2000H

HLT

(Q12) Write an ALP in 8085 to check whether number stored in memory location C000H is even or odd, and store it in memory location D000H if it is odd otherwise store it at memory location E000H.

→

LDA C000H

RAR

JNC loop

MVI A,01H

STA E000H

HLT

loop: MVI A,00H

STA D000H

HLT

(13) Write an ALP in 8085 to find the greatest of three numbers stored in memory location from 2000H and store the final result in memory location 3000H.

LDA 2000H

MOV C,A

LDA 2001H

MOV B,A

LDA 2002H

CMP B

JNC loop1

MOV A,B

loop1 : CMP C

JNC loop2

MOV A,C

loop2 : STA 2050H

HLT

Q14) Write an ALP in 8085 to find the smallest of three numbers stored in memory location starting from 4000H and store the final result in memory location 4000H.

LDA 4000H

MOV C,A

LDA 4001H

MOV B,A

LDA 4002H

CMP B

JC loop1

MOV A,B

loop1: CMP C

JC loop2

MOV A,C

loop2: STA 4000H

HLT

RAL

cy

Date: _____

Page: _____

- (Q15) Write an ALP in 8085 to count number of 1's in the given byte of data. Data byte is 55H.

→

MVI A, 55H // Load data byte 55H to Accumulator
MVI B, 00H // Clear Register B (counter for no. of 1's)
MVI C, 08H // Loads register C with 8 (Number of bits in a byte)

loop: RAL // Rotate accumulator left through carry
JNC skip // If carry is 0, skip incrementing counter
INR B // If carry is 1, increment register B (counter)

skip: DCR C // Decrement C
JNZ loop // Repeat loop for 8 bits

MOV A,B // Store the result (number of 1's in memory location specified by HL pair).
STA 2050H.
HLT // Terminates program

(Q16) Write an ALP in 8085 to count number of 0s in data byte ABH.

MVI A, ABH	// Load data byte ABH into Accumulator (A)
MVI B, 00H	// Clear register B (counter for number of 0s)
MVI C, 08H	// Load register C with 8 (number of bits in a byte)

loop:

RAL	// Rotate accumulator left through carry
JNCZ Skip	// If carry is 1, skip incrementing counter
INR B	// If carry is 0, increment register B (counter)

skip:

DCR C	// Decrement counter C
JNZ loop	// Repeat the loop for all 8 bits

MOV A, B	// Store the result (number of 0s in memory location
STA 2050H	specified by HL pair)
HLT	// Terminates the program

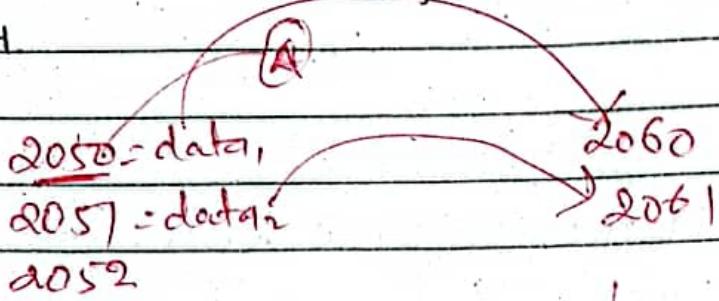
(Q17) Write an ALP in 8085 to check whether the data byte 45H is positive or negative. Store it in 3000H if it is positive otherwise in 4000H.

MVI A, 45H	// Load data to accumulator
RAL	// Rotate accumulator left with carry

JNC loop // Jump if no carry
 STA 4000H // Store content of accumulator for negative
 (carry = 1)
 HLT // Terminates program

loop: STA 3000H // store content of A for positive
 HLT // Terminates program

(Q18) write an ALP in 8085 to transfer 10 data bytes starting from 2050H into memory locations starting from 2060H.



LXI D, 2060H MVI B, 0A H

LXI H, 2050

MOV A, M

STAX D

DCR B

JN2 LOOP

LXI H, 2050H

LXI D, 2060H

MVI B, 0AH

loop: MOV A,M

STAX D

INX H

INX D

DCR B

JNZ loop

HLT

(Q19) Write an ALP in 8085 to perform following expression.

$$1^2 + 2^2 + 3^2 + \dots + n^2, \text{ where } n = 1, 2, \dots, 9$$

MVI A, 00H

MVI B, 09H

loop: MOV C, B

loop1: ADD B

DCR C

JNZ loop1

DCR B

JNZ loop

STA 2050H

HLT