Mahesh Singh Madai
017BSCIT022 (St. Xavier's College)

Model Set

1. Draw logical diagram of 8086 microprocessor and explain its segmented memory structure

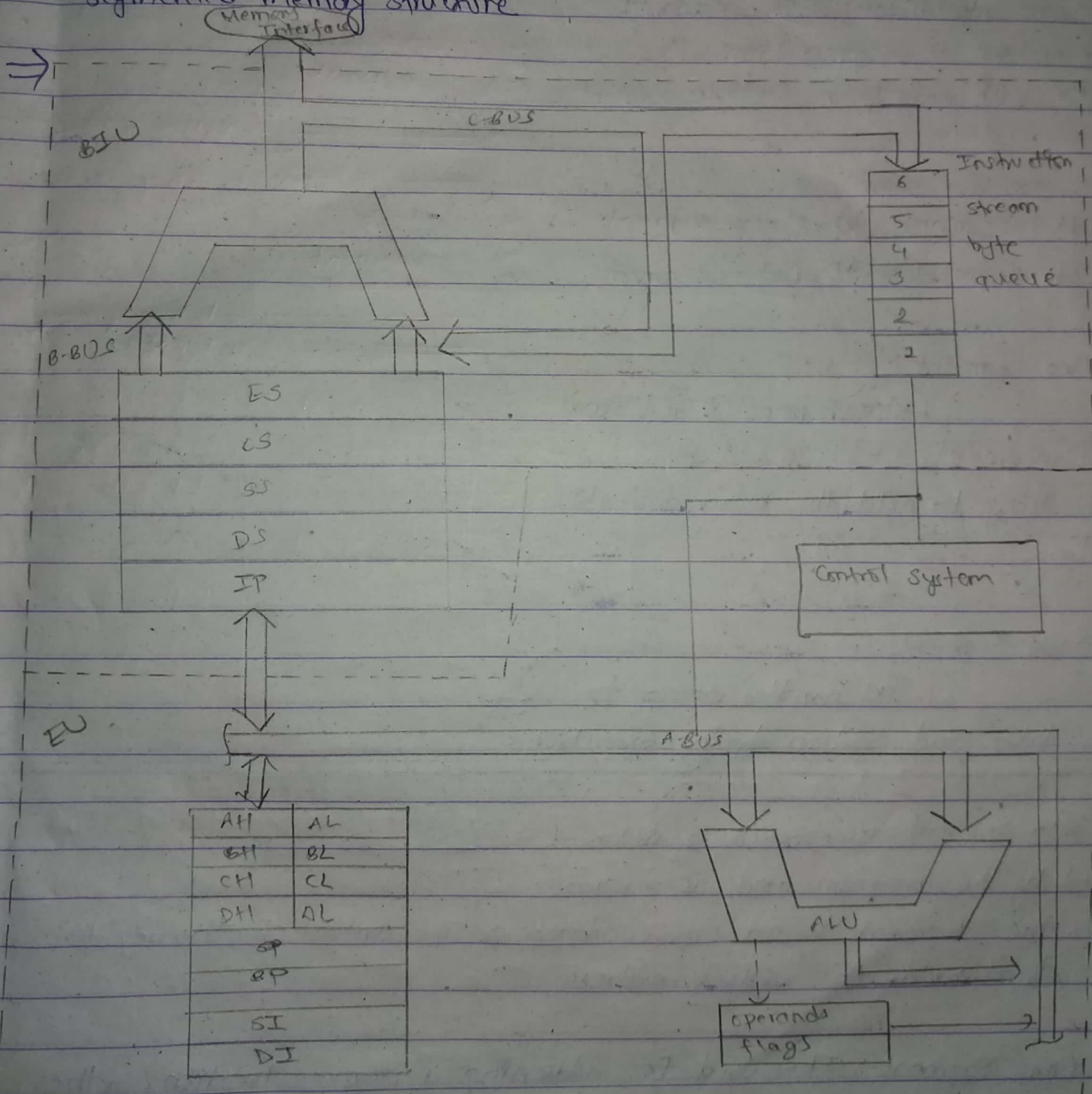


fig: Block Diagram of 8086

Segmented Memory of 8086.

The BIU sends out 20-bit address so it can address any of 2^{20} bytes in memory. But at a given time, it works with only four 64kbyte segment

Here, are four segments:
a) Extra Segment b) stack segment
c) Code Segment d) Data Segment

a) Extra Segment:-

Extra Segment is additional data segment, which is used by the string to hold the extra destination data.

b) Stack Segment:

It handles memory to store data and address during execution.

c) Data Segment: It consists of data used by the program and is accessed in the segment by an offset address or the content of other register that holds the offset address.

d) Code Segment: It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.

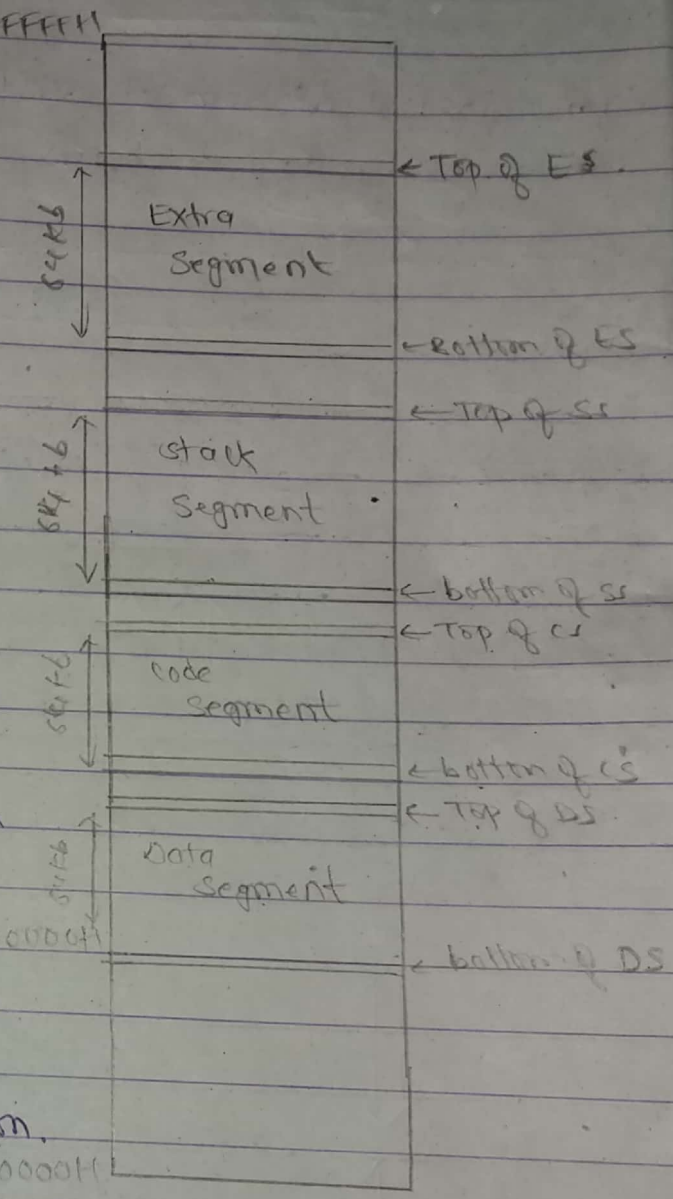


Fig: Memory segmentation in 8086

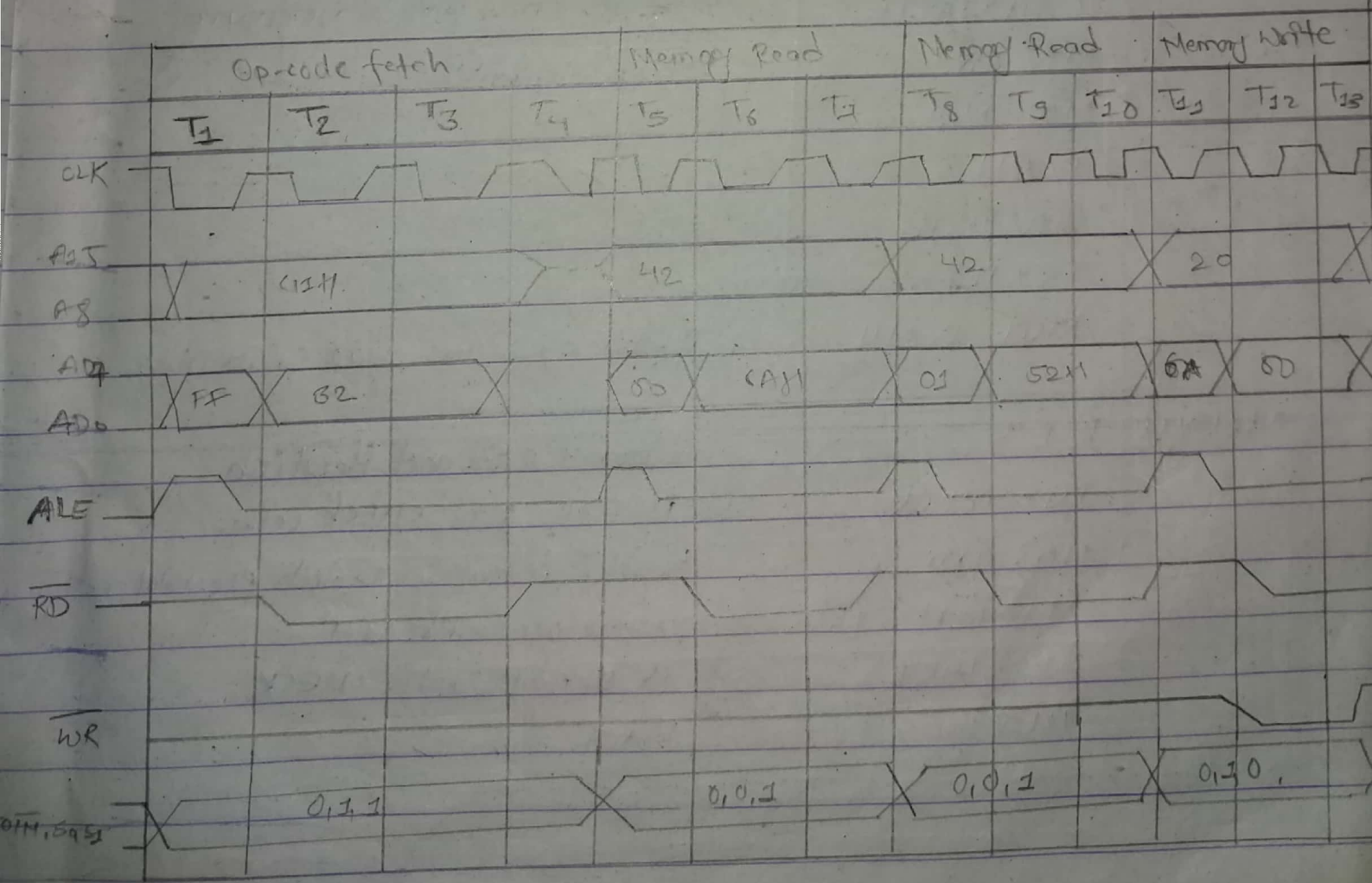
2. What is machine cycle and instruction cycle? Draw a timing diagram of STA 2000h memory instruction.

Ans Machine cycle: It is defined as the time required to complete one operation of accessing memory i/p, o/p or acknowledge and external request.

Instruction cycle:

It is defined as the total time required to execute an instruction is called instruction cycle.

Timing Diagram of STA 2000H :-



3. Write an Assembly Language program to sort an array in ascending order using 8bit microprocessor

⇒

Label	Instruction	Comment
START:	LXI H, 2040H	Load size of array
	MVI D, 00H	clear D register
	MOV C, M	counter = C
	DCR C	decrement of C by 1
	INX H.	
CHECK:	MOV A, M	list in A.
	INX H	
	CMP M	compare accumulator with next element
	JC NEXTBYTE	If acc < no., jump to NEXTBYTE else
	MOV B, M	Swap elements
	MOV M, A	
	DCX H	
	MOV M, B	
	INX H.	
	MVI D, 01H	if exchange occurs, swap 01 in D reg
NEXTBYTE	DCR C	Decrement C for next iteration
	JNZ CHECK	If C>0, goto CHECK else,
	MOV A, D	transfer contents of D into Accumulator
	CPI 01H	compare acc. with 01H
	JZ START	if D=01H, goto CHECK
	HLT	HALT.

OBSERVATION:

Size

05H

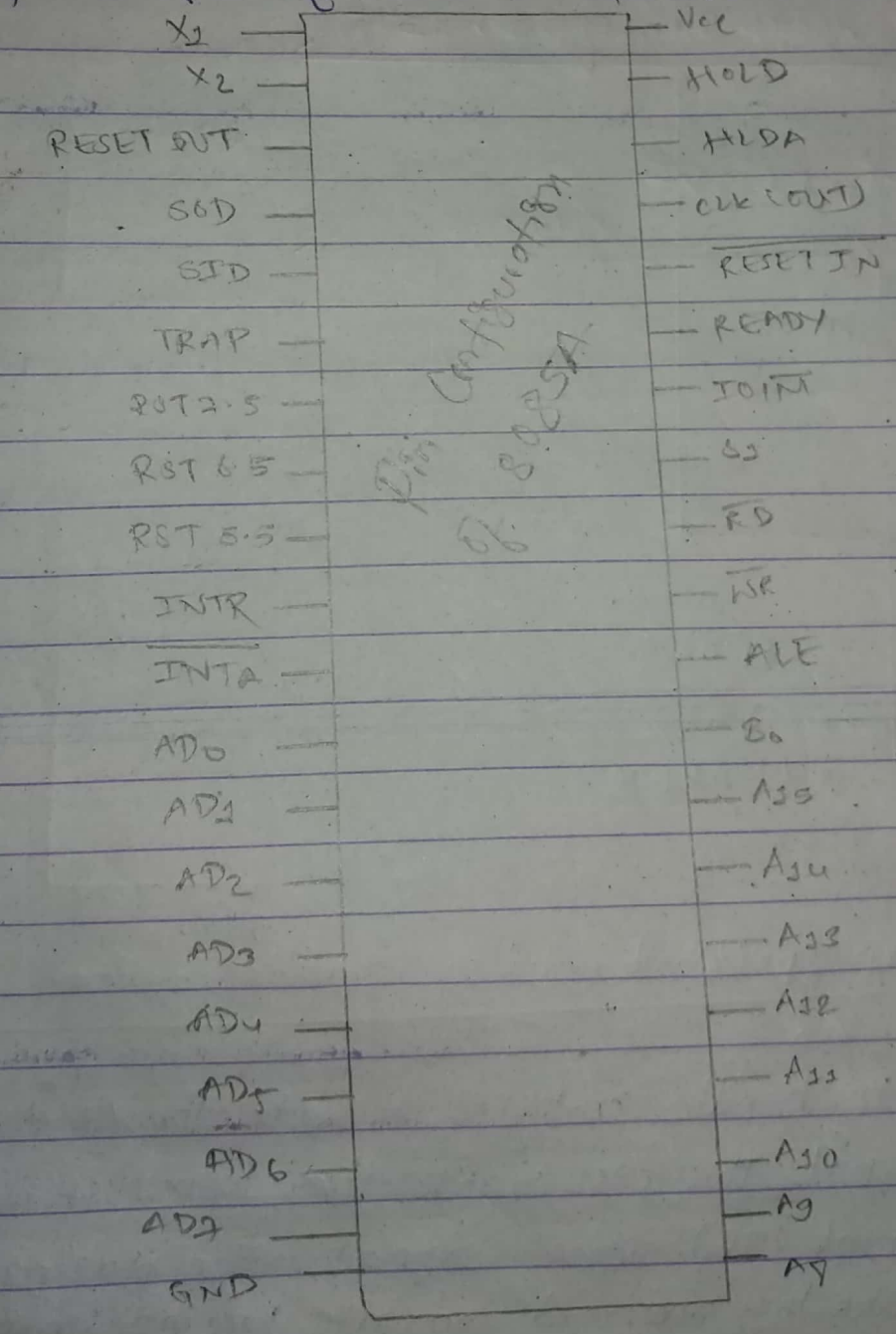
List

Input	35H	10H	02H	02H	F0H

List

Output	02H	10H	21H	35H	F0H

4. Draw a pin diagram of 8085 microprocessor with labelling.



6. What is DMA? Explain the sequence of events that occur during DMA operation?

DMA is a Direct Memory Access which is defined as the data transfer technique in which peripherals manage the buses for direct interaction with main memory without involving the CPU.

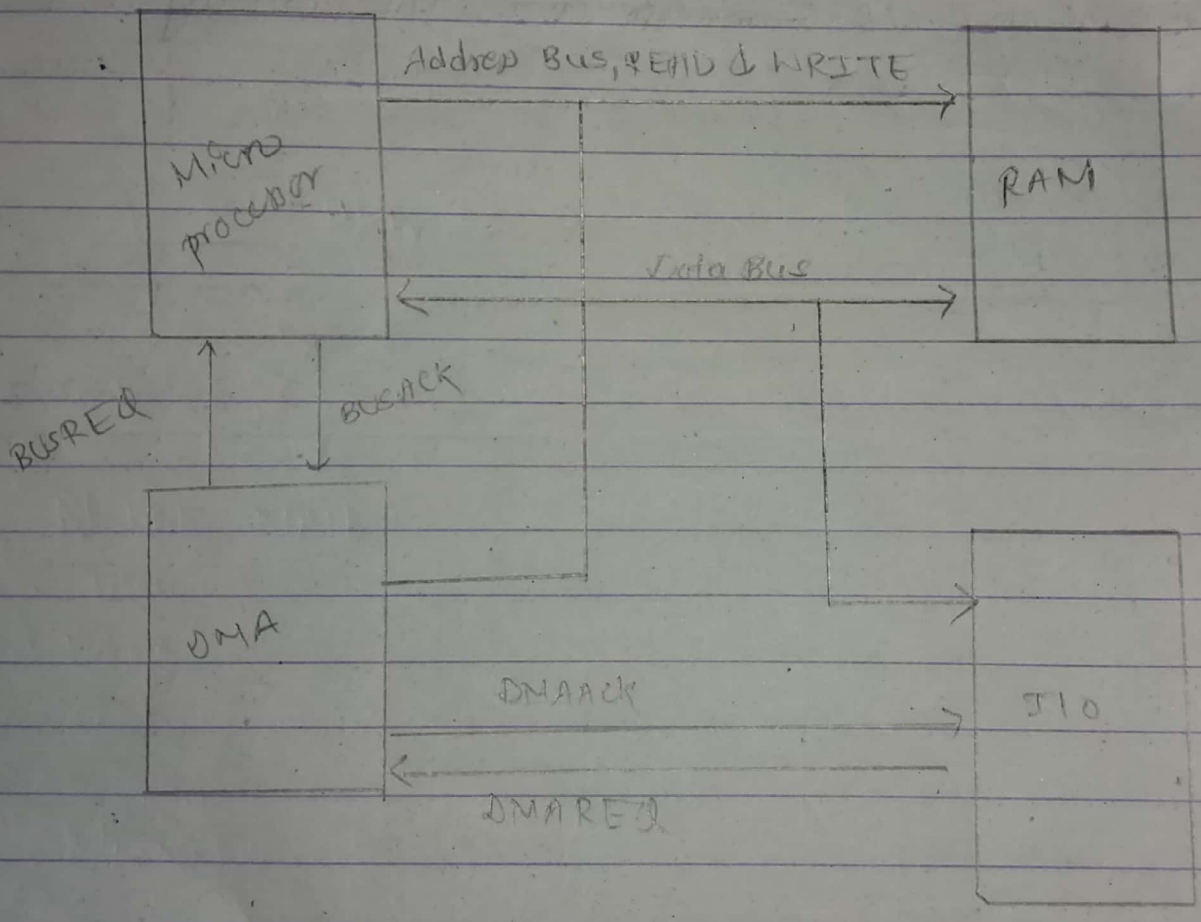


fig: Illustration of DMA transfer in computer System

DMA request CPU to handle control of buses to the DMA using Bus Request (BR) signal. The CPU grants the control of buses to DMA using Bus Grant (BG) signal signal after placing the address bus, data bus and read and write lines into impedance state

7. What is addressing mode? Explain different addressing modes in 8085 microprocessor

Ans The various way of specifying the operands are called addressing modes. The operands maybe the source only, destination only or both of them

In 8085, there are 5 addressing modes:

1) Direct Addressing Mode:

The instruction using this mode specifies the effective address as part of instruction. The instruction size is either 2-bytes or 3-bytes with first byte op-code followed by 1 or 2 bytes of address of data.

eg: LDA 9500H

IN 80H

2) Register Direct Addressing Mode:

This mode specifies register or register pair that contains the data.

eg:

MOV A,B

3) Register Indirect Addressing Mode:

In this mode, the address part of the instruction specifies the memory whose contents are the address of the operand. So, here address of the address is called

eg. MOV A,M

STAX B

4) Immediate Addressing Mode:

In this mode the operand position is the immediate data. For 8bit data, instructions is 2 byte and for 16 bit data, instruction size is 3 bytes.

eg: MVI A,32H
LXI B, 4567H.

5) Implied or Inherent Addesing Mode:

The instruction of this mode donot have operands.

eg. NOP. HLT, EI, DI

8. WAP to reverse a given string using 16-bit microprocessor.

⇒

```
.MODEL SMALL
.STACK
.DATA
.CODE

MAIN PROC

    MOV AX, @DATA
    MOV DS, AX

    MOV CX, 0
```

```
READ_CHAR:
        MOV AH, 01H
        INT 21H
        CMP AL, 0DH.

        JE END_OF_LINE

        PUSH AX
        INC CX
        JMP READ_CHAR

END_OF_LINE:
        POP DX
        MOV AH, 02H
        INT 21H
        LOOP END_OF_LINE

        MOV AX, 4C00H
        INT 21H

MAIN ENDP
END MAIN
```

9. Explain Memory Interfacing in 8085 microprocessor along with appropriate diagram.

AW

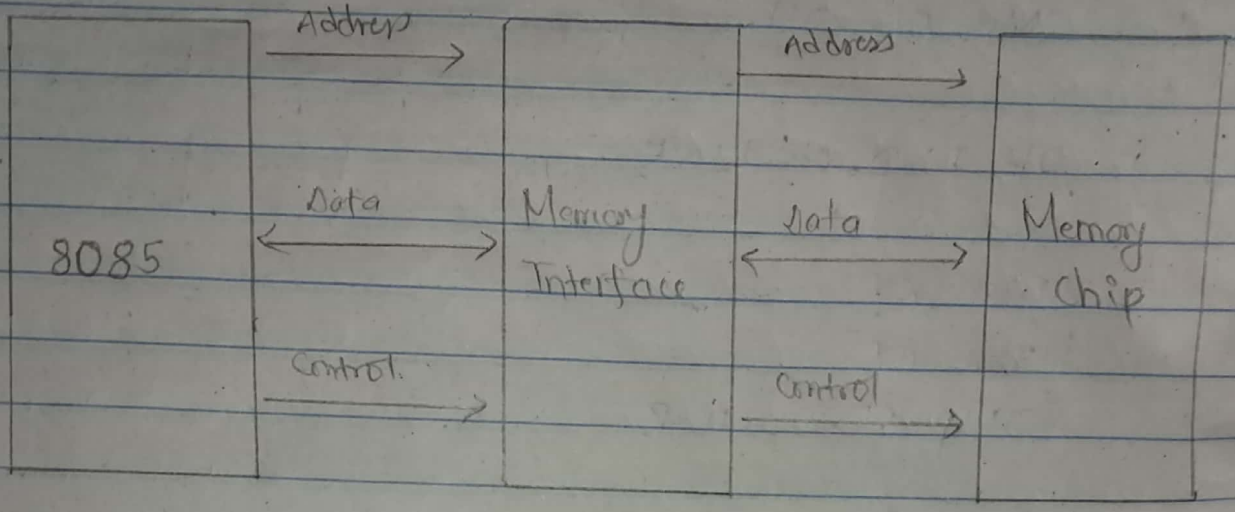


fig: 8085 interfacing with memory chips

The interface process involves designing a circuit that will match the memory requirements with the microprocessor signal.

Memory has certain signal requirements to read from and write into memory. Similarly, microprocessor initiates the set of signals when it wants to read from and write into memory.

- 8085 has 16 Address lines. Hence a maximum of 64Kb of memory locations can be interfaced with it. The memory address space of 8085 takes values of 0000H to FFFFH. It initiates IO/$\overline{M}$, $\overline{RD}$, $\overline{WR}$ signal. Similarly, each memory chip has signals such as CS, OE or $\overline{RD}$ and $\overline{WR}$ associated with it.

10. What are different modes of operation in 80286 microprocessor? Explain in brief.

Ans The operating modes of 80286 are: Real mode and private virtual mode

a) Real Mode:

In Real addressing mode, only 1 MB of physical memory is addressed using lines $A_0 - A_{19}$. While addressing, the content of segment registers are used as segment base addresses. In this mode, the first 1 kbyte of memory starting from address 0000 H to 003FFH is reserved for interrupt vector table.

When 80286 is reset, it always starts the execution in real address mode. In real address mode, it initialises the IP and other registers of 80286.

b) Protected Virtual Address Mode (PVAM):

In 80286, the concept of virtual memory was introduced and it could provide 1 GB of virtual memory per task. The complete virtual memory is mapped onto the 16 MB physical memory. Whenever the portion of a program is required for execution, it is fetched from the secondary memory and placed in the physical memory. And also, results maybe saved back on secondary memory

For addressing: 80286 uses 16-bit content of a segment register as a selector to a address of a descriptor block stored in the physical me

11) Interrupt base I/O is efficient compared to polled I/O." Justify.

Ans In polled I/O, CPU constantly checks device status whether it needs CPU's attention. The CPU continuously test each and every device attached to it for detecting whether any device needs CPU attention. Here, CPU has to wait and check the service need for devices and this wastes a lot of time (CPU cycles). Polling becomes inefficient when CPU rarely finds a device ready for service.

Whereas, in interrupt I/O, interrupt handler serves the device. It is a hardware mechanism that enables CPU to detect that a device needs its attention. Here, CPU is distributed only when a device needs serves, which saves the CPU cycles.

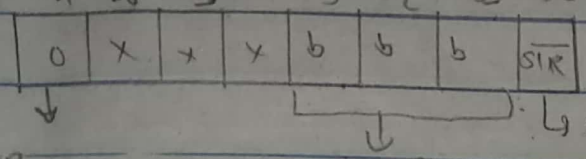
So, in the cases when CPU doesnot find a service to do, polling becomes inefficient but interrupt based I/O is efficient.

12) Write Short Notes on:

a) BSR Mode:

BSR stands for Bit Set Reset Mode. It is a port C bit set/reset mode. The individual bit of port C can be set or reset by writing control word in the control register.

The control word format of BSR mode is shown in the figure below:



BSR mode

port c bit select

↳ port c set/reset
1 = set
0 = reset

b	b	b	
0	0	0	Bit 0
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7

- The pin of port c is selected using bit select bits [b b b] and set or reset is decided by bit S/R̄.
- The BSR mode affects only one bit at a time. To set any bit of port C, bit pattern is loaded in control registers.
- If a BSR mode is selected, it will not affect IO mode.

a) Macro Assembler:-

An assember that can perform macro substitutions and expansion is the macro assembler. The programmer can define a macro that consists of several statements and then use the macro name later in the program, thus avoiding having to rewrite the statements. The macro begins with %macro directive and ends with %endmacro directives.

For example:

A macro called swap exchanges the values of two variables. After defining swap, the programmer can then insert an instruction such as swap a,b in the assembly language program.

Q.NO.5

MVI A, AAH

→ copies immediate value 'AAH' into the accumulator.

i.e. A = AAH

MOV B, A

→ copies content of accumulator to the reg-B.

i.e

A = AAH

B = AAH

RRC

→ Rotate accumulator right with carry.

i.e

A = 55H

B = AAH

XRA B

→ XOR operation between A and B.

A = 55H → 0101 0101

B = AAH → (XOR) 1010 1010

1111 1111

Then,

A = FFH

B = AAH.

OUT PORT 1

→ PORT 1 outputs FFH.

HLT →

Halts the program.