Microprocessor

Course Title: Microprocessor
Course No: CSC162
Nature of the Course: Theory + Lab
Semester: II

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

Course Description: This course contains fundamental concepts of Microprocessor operations, basic I/O interfaces and Interrupts operations.

Course Objectives: The course objective is to introduce the operation, programming and application of microprocessor.

Course Details

Unit 1: Introduction (4 Hrs.)

- Definition of microprocessor and its application
- Evolution of microprocessor, Von Neumann and Harvard architecture
- Components of microprocessor
  - Microprocessor: Arithmetic and Logic Unit (ALU), Control Unit (CU), Registers
  - Memory
  - Input / Output
- System Bus: Data , Address and Control Bus
- Microprocessor with Bus Organization

Unit 2: Basic Computer Architecture (7 Hrs.)

- 8085 Microprocessor Architecture and Operations
  - Address, Data And Control Buses
  - Internal Data Operation and Registers
  - Externally Initiated Operations
  - Addressing Modes
  - Memory and Memory Operations
  - Flag and Flag Register
  - 8085 Pin Diagram and Functions
  - Multiplexing and De-multiplexing of address/data bus
  - Generation Of Control Signals
- 8086 Microprocessor
  - Logical Block Diagram
  - Segment Registers,
  - Memory Segmentation
  - Bus Interface Unit and Execution Unit
  - Pipelining

Unit 3: Instruction Cycle (3 Hrs.)

- Instruction Cycle, Machine Cycle and T-states
  - Machine Cycle of 8085 Microprocessor: op-code fetch, memory read, memory write, I/O read, I/O write, interrupt
  - Fetch and Execute Operation, Timing Diagram
  - Timing Diagram of MOV, MVI, IN, OUT, LDA, STA
  - Memory Interfacing and Generation of Chip Select Signal

Unit 4: Assembly Language Programming **(10 Hrs.)**

- Programming with Intel 8085 Microprocessor
- Instruction and Data Format
- Mnemonics and Operands
- Instruction Sets
- Data Transfer:- MOV, IN, OUT, STA;LDA, LXI, LDAX, STAX, XCHG
- Arithmetic and Logic:- ADD, SUB, INR, DCR, AND, OR, XOR, CMP, RLC, RRC, RAL, RAR
- Branching:- JMP, JNZ, JZ, JNC, JC, CALL
- Stack:- PUSH, POP
- Multiplication and Division
- Simple Sequence Programs, Branching, Looping
- Array(Sorting) and Table Processing
- Decimal to BCD Conversion

❑ Programming with Intel 8086 microprocessor

- Macro Assembler
- Assembling and Linking
- Assembler Directives, Comments
- Instructions: LEA, MUL, DIV, LOOP, AAA, DAA
- INT 21H Functions
- 01H, 02H, 09H, 0AH, 4CH
- INT 10H Functions (Introduction Only)
- 00H, 01H, 02H, 06H, 07H, 08H, 09H, 0AH
- Simple String and Character Manipulation Programs
- Debugging

Unit 5: Basic I/O, Memory R/W and Interrupt Operations **(6 Hrs.)**

- Memory mapped I/O, I/O Mapped I/O and Hybrid I/O
- Direct Memory Access (DMA)
- Introduction, Advantage and Application
- 8237 DMA Controller and Interfacing

❑ Interrupt

- 8085 Interrupt Pins and Priority
- Maskable and Non-maskable Interrupts
- RST Instructions
- Vector and Polled Interrupt

❑ 8259 Interrupt Controller

- Block Diagram and Explanation
- Priority Modes and Additional Features

Unit 6: Input/ Output Interfaces **(6 Hrs.)**

❑ Parallel Communication – Introduction and Applications

❑ Serial Communication

- Introduction and Applications
- Introduction to Programmable Communication Interface 8251

- Basic Concept of Synchronous and Asynchronous Modes

- Simple I/O, Strobe I/O, Single handshake I/O, Double handshake I/O
- 8255 A and it's Working
  - Block Diagram
  - Modes of Operation
  - Control Word
- RS-232 – Introduction, Pin Configuration (9 pin and 25 pin) and function of each pin, Interconnection between DTE-DTE and DTE-DCE

Unit 7: Advanced Microprocessors **(9 Hrs.)**

- 80286: Architecture (Block Diagram) , Registers, (Real/Protected mode), Privilege Levels, Descriptor Cache, Memory Access in GDT and LDT, Multitasking, Addressing Modes, Flag Register
- 80386: Architecture (Block Diagram), Register organization, Memory Access in Protected Mode, Paging (Up to LA to PA)

Laboratory Works:

The laboratory work includes Assembly language programming using 8085/8086/8088 trainer kit. The programming should include: Arithmetic operation, base conversion, conditional branching etc. The lab work list may include following concepts:

1. Assembly language program using 8085 microprocessor kit and 8085 microprocessor simulator.
2. Use of all types of instructions and addressing modes.
3. Program including basic arithmetical, logical, looping, bitwise and branching.
4. Assembly language programming using 8086 microprocessor emulator, using any types of Assembler, including the different functions of 21H.

Text Books:

1. Ramesh S.Gaonkar, Microprocessor Architecture, Programming, and Applications with 8085, Prentice Hall

Reference Books:

1. A.P.Malvino and J.A.Brown, Digital Computer Electronics, 3rd Edition, Tata McGraw Hill D.V.Hall, Microprocessors and Interfacing-Programming and Hardware, McGraw Hill
2. 8000 to 8085 Introduction to 8085 Microprocessor for Engineers and Scientists, A.K.Gosh, Prentice Hall

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Bachelor Level/ First Year/ Second Semester/ Science
Microprocessor (CSC 162)
Time: 3 hours.

Full Marks: 60
Pass Marks: 24

MODEL QUESTIONS-ANWERS

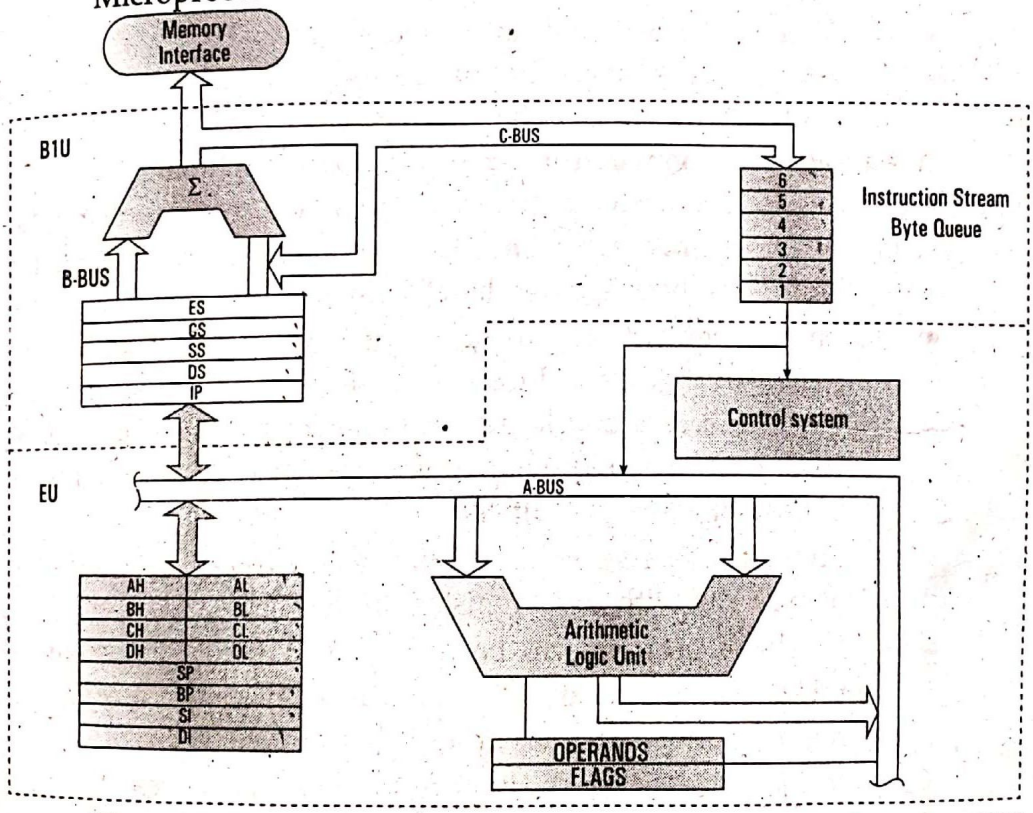
Candidates are required to give their answers in their own words as for as practicable.
The figures in the margin indicate full marks.

Group A (Long Answer Question Section)

Attempt any TWO questions. **(2x10=20)**

1. **Draw logical block diagram of 8086 microprocessor and explain its segmented memorystructure.**

Ans: The following diagram depicts the architecture of an 8086 Microprocessor:



8086 Microprocessor is divided into two functional units, i.e., EU (Execution Unit) and BIU (Bus Interface Unit).

Segmented Memory

- The 8086 BIU sends out 20-bit address so it can address any of 2^{20} or 1,048,576 bytes in memory.
- However at any given time the 8086 works with only four 65536 bytes (64 Kbyte) segment within this 1,048,576 byte (1 Mbyte) Range

- Four segments are : Code Segment, Stack Segment, Data Segment and Extra Segment
- Four segment registers in BIU are used to hold the upper 16 bits of the starting address of 4 memory segments that the 8086 is working with at a particular time.
- The 4 segment registers are code segment register (CS), stack segment register (SS), data segment register (DS) and the extra segment register (ES).
- For small programs which do not need all 64 Kbytes in each segment can overlap.
- For example, the code segment holds the upper 16 bits of the starting address for the segment from which the BIU is currently fetching instruction code bytes.
- The BIU always inserts zero for the lowest 4 bits of the 20-bit starting address.
- If the code segment register contains 348A H then the code segment will start at address 348A0 H
- A 64 Kbytes segment can be located anywhere within the 1 Mbyte address space, but the segment will always start at an address with zeros in the lowest 4 bits

Advantages of Segmentation [Segment: Offset Scheme]

Intel designed the 8086 family devices to access memory using the segment: offset approach rather than accessing memory directly with 20 bit. The advantages are listed below.

- The segment: offset scheme requires only a 16-bit number to represent the base address for a segment and only a 16 bit offset to access any location in a segment. This means that 8086 has to manipulate and store only 16-bit quantities instead of 20-bit quantities.
- This makes easier interface with 8 and 16-bit wide memory boards and with 16 bit registers in the 8086.
- It allows programs to be relocated in memory system. A relocatable program is one that can be placed in any area of memory and executed without change.
- It allows programs written to function in the real mode to operate in protected mode.
- Segmentation also makes easy to keep user's program and data separate from one another and segmentation makes it easy to switch from one user's program to another user's program.

Disadvantage of Segment: Offset Approach

- The segment: offset scheme introduces complexity in hardware and software design.

Different Segment Offset Combination

SEGMENT	OFFSET	SPECIAL PURPOSE
CS [CODE SEGMENT]	IP [INSTRUCTION POINTER]	INSTRUCTION
SS [STACK SEGMENT]	SP [STACK POINTER] BP [BASE POINTER]	ADDRESS
DS [DATA SEGMENT]	BX [BASE REGISTER] DI [DESTINATION INDEX	STACK
	SI [SOURCE INDEX] 8 BIT NUMBER 16 BIT NUMBER	ADDRESS
ES [EXTRA SEGMENT]		

2. What is machine cycle and instruction cycle? Draw a timing diagram for STA 2000hmemory instruction. (Choose any memory locations for loading STA 2000h instruction)

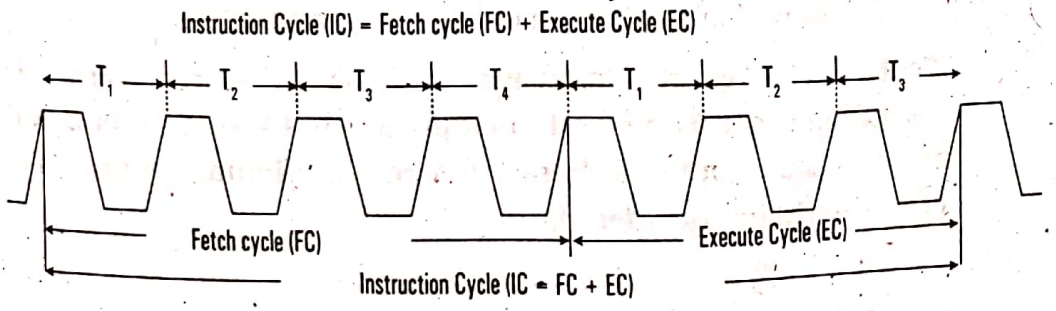
Ans: Machine cycle:

It is defined as the time required to complete one operation of accessing memory i/p, o/p or acknowledging and external request. This cycle may consists 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.

Instruction cycle:

The necessary steps that the CPU carries out to fetch an instruction and necessary data from the memory and to execute it constitute and instruction cycle 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 fetch an opcode from memory constitute a fetch cycle. 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 and execute cycle. The total time required to execute an instruction given by IC = Fc+ Ec the 8085 consists of 1-6 machine cycles or operations.



Timing diagram of STA 2000H

- STA means Store Accumulator- The contents of the accumulator is stored in the specified address(2000).
- The opcode of the STA instruction is said to be 32H. It is fetched from the memory 41FFH.
- Then the lower order memory address is read(00). - *Memory Read Machine Cycle*
- Read the higher order memory address (20). - *Memory Read Machine Cycle*
- The combination of both the addresses are considered and the content from accumulator is written in 2000. - *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 2000.

Address	Mnemonics	Op code
41FF	STA 2000	32H
4200		00H
4201		20H

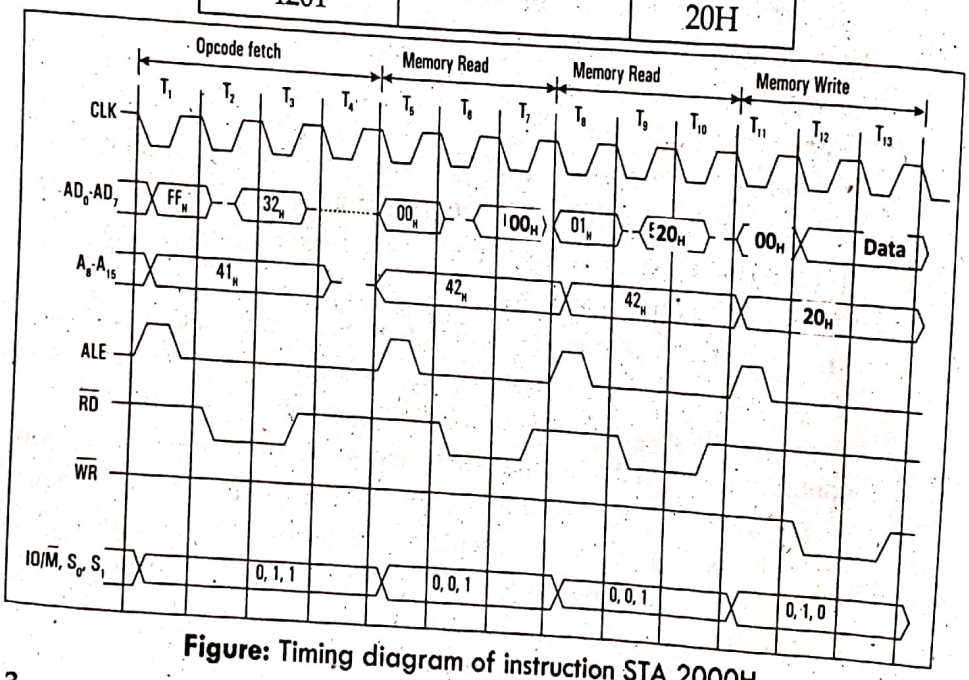


Figure: Timing diagram of instruction STA 2000H.

3. **Write an assembly language program to sort an array in ascending order using 8 bit microprocessor. (Assume appropriate array data and address where minimum array size of 10 should be considered)**

Ans:

	LXIH, 8D10H	8C20	21	10	8D	Address for count
	MOV C, M	8C23	4E			Count in C register
	DCR C	8C24	OD			Count for number of passes in C register i.e. count-1
BACK	MOV D, C	8C25	51			Count for number of comparisons in D register
	LXIH, 8D11H	8C26	21	11	8D	Starting address of the array of data in H-L register pair
	MOV A, M	8C29	7E			1st number in accumulator
LOOP:	INX H	8C2A	23			Address of next number
	MOV B, M	8C2B	46			Next number in B register
	CMP B	8C2C	B8			Compare next number with the previous greatest number in ACC
	JNC AHEAD	8C2D	D2	36	8C	If previous greatest number> next number, go to AHEAD
	DCX H	8C30	2B			Address of previous memory location
	MOV M, A	8C31	77			Place smaller of the two compared numbers in memory
	MOV A, B	8C32	78			Place greater of the two numbers in accumulator
	JMP GO	8C33	C3	38	8C	Branch to step labeled GO.
AHEAD:	DCXH	8C36	2B			Address of ,previous memory location
	MOV M, B	8C37	70			Place smaller of the two number in memory
GO:	IN X H	8C38	23			Address of next memory location
	DCR D	8C39	15			Decrease the count for comparisons
	JNZ LOOP	8C3A	C2	2A	8C	Branch to stop labeled LOOP if the count is not equal to 0.
	MOV M, A	8C3D	77			Place the greatest number after a pass in the memory
	DCR C	8C3E	OD			Decrease the count for passes
	JNZ BACK	8C3F	C2	25	8C	If count for passes = 0, go the BACK
	HLT	8C42	76			Other wise, stop.

Group B (Short Answer Question Section)

Attempt any EIGHT questions. (8x5=40)

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

Ans: PIN Configuration of 8085

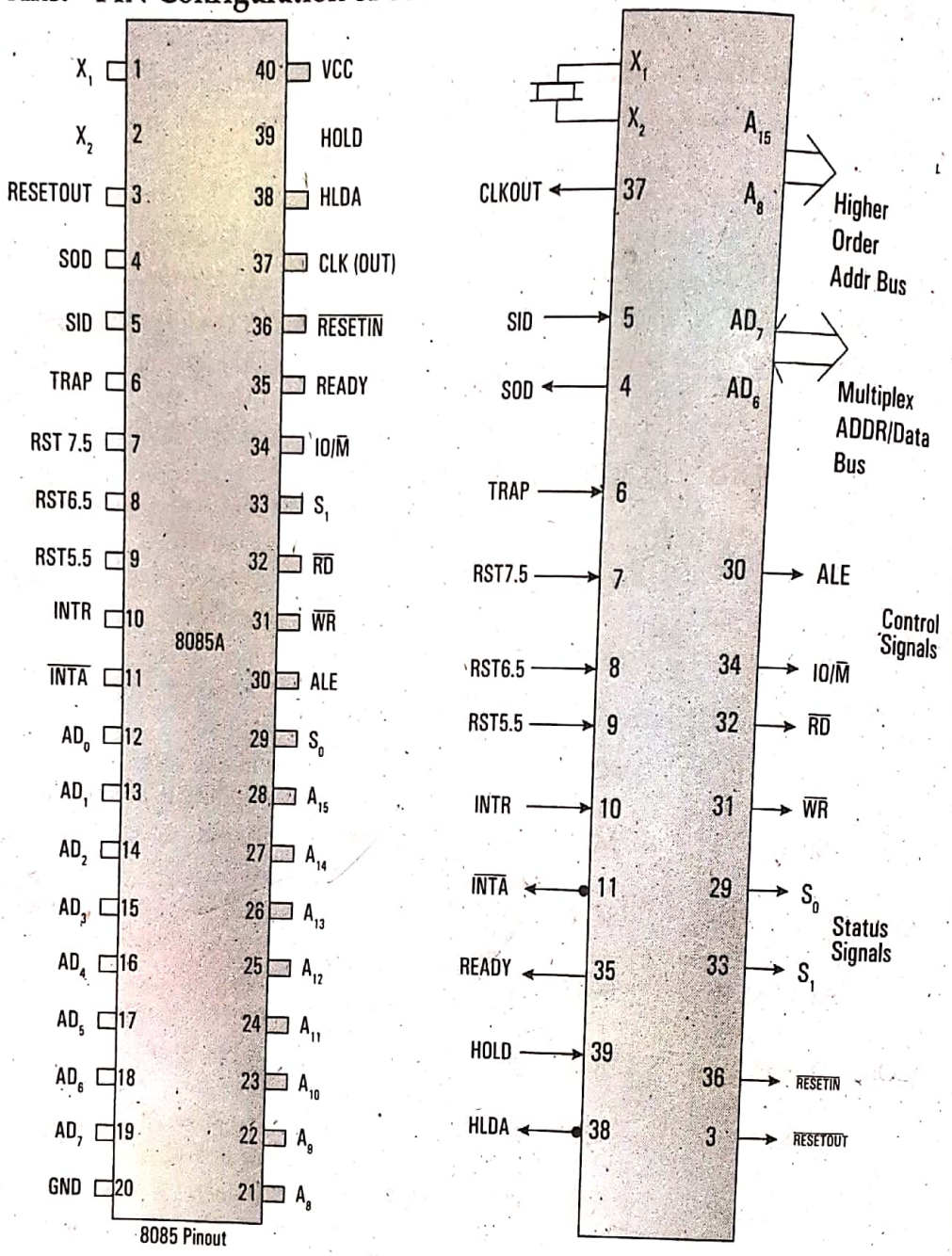


Figure: PIN Configuration of 8085

- The microprocessor 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.
- The microprocessor is capable of performing various computing functions and making decisions to change the sequence of program execution.

- In large computers, a CPU implemented on one or more circuit boards performs these computing functions.
- The microprocessor is in many ways similar to the CPU, but includes the logic circuitry, including the control unit, on one chip.
- The microprocessor can be divided into three segments for the sake clarity, arithmetic/logic unit (ALU), register array, and control unit.
- 8085 is a 40 pin IC, DIP package. The signals from the pins can be grouped as follows:
  1. Power supply and clock signals
  2. Address bus
  3. Data bus
  4. Control and status signals
  5. Interrupts and externally initiated signals
  6. Serial I/O ports

5. **Specify the output in PORT1 after the execution of the following program. Writecomments for each instruction.**

```
MVI A, AAH
MOV B, A
RRC
XRA B
OUT PORT1
HLT
```

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

Ans: The data transfer technique in which peripherals manage the memory buses for direct interaction with main memory without involving the CPU is called direct memory access (DMA). Using DMA technique large amounts of data can be transferred between memory and the peripheral without severely impacting CPU performance.

During the DMA transfer, the CPU is idle and has no control of the memory buses. A DMA controller takes over the buses to manage the transfer directly between the I/O device(s) and main memory.

The 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 after placing the address bus, data bus and read and write lines into high impedance state (which behave like open circuit).

CPU initializes the DMA by sending following information through the data bus.

1. Starting address of memory block for read or write operation.

2. The word count which is the no. of words in the memory block.
3. Control to specify the mode of transfer such as read or write.
4. A control to start the DMA transfer.

The DMA takes control over the buses directly interacts with memory and I/O units and transfers the data without CPU intervention. When the transfer completes, DMA disables the BR line. Thus CPU disable BG line, takes control over the buses and return to its normal operation.

The DMA transfer operation is illustrated below:

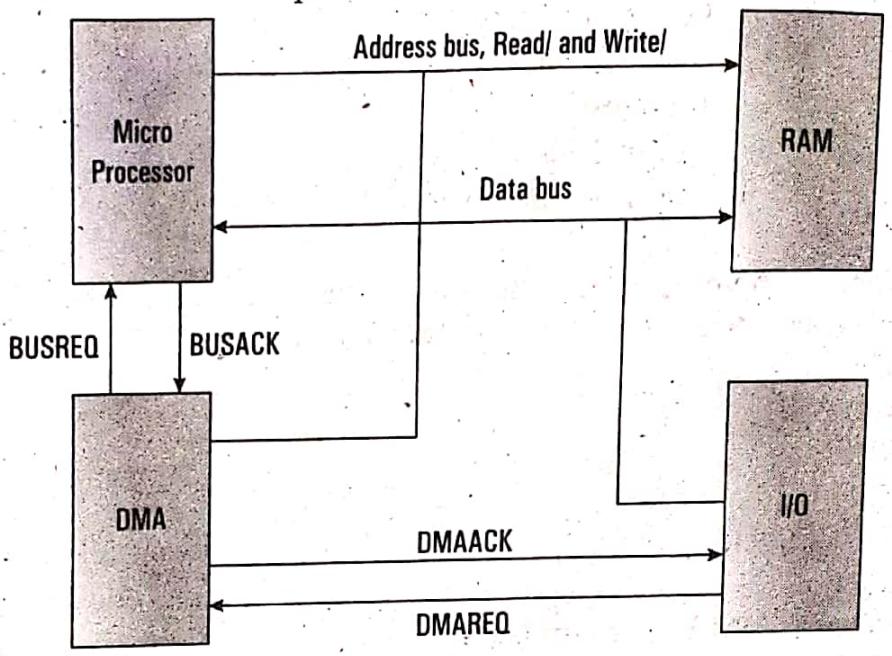


Figure: Illustration for DMA transfer in a computer system

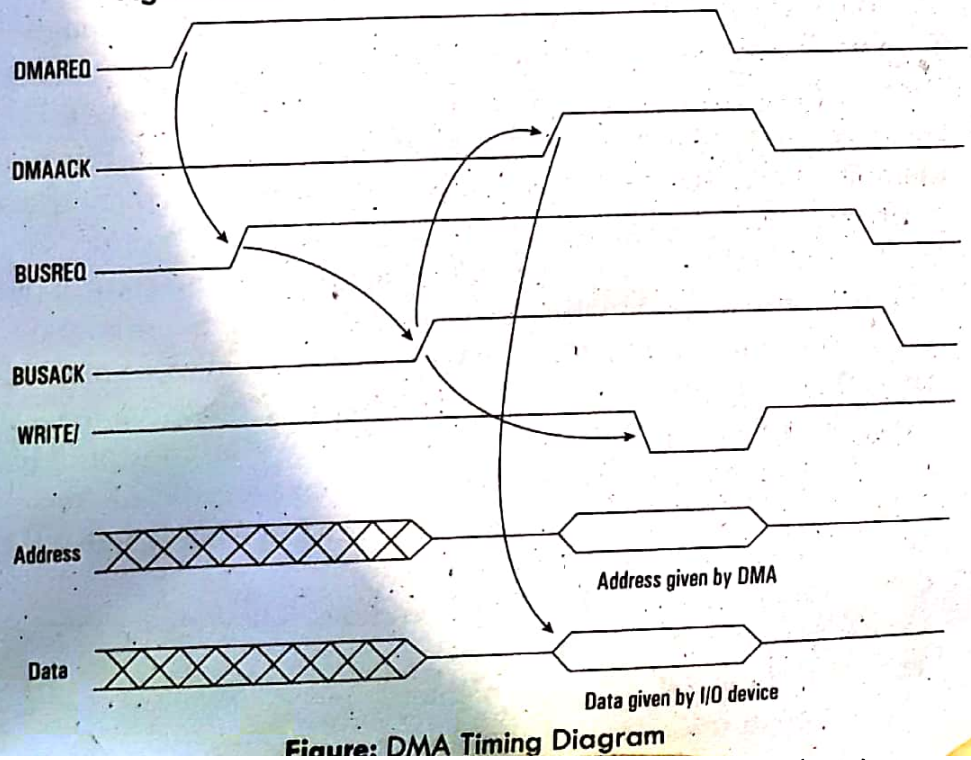


Figure: DMA Timing Diagram

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

Ans: **Addressing modes:**

Instructions are command to perform a certain task in microprocessor. The instruction consists of op-code and data called operand. The operand may be the source only, destination only or both of them. In these instructions, the source can be a register, a memory or an input port. Similarly, destination can be a register, a memory location, or an output port. The various format (way) of specifying the operands are called addressing mode. So addressing mode specifies where the operands are located rather than their nature. The 8085 has 5 addressing mode:

1) **Direct addressing mode:**

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

E. g. LDA 9500H A ← [9500]
IN 80H A ← [80]

This type of addressing is called absolute addressing.

2. **Register Direct addressing mode:** This mode specifies the register or register pair that contains the data.
E g. MOV A, B
Here register B contains data rather than address of the data.
Other examples are: ADD, XCHG etc.

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 in this type of addressing mode, it is the address of the address rather than address itself. (One operand is register)
e. g. MOV R, M
STAX, LDAX etc.
STAX B
B= 95 C =00
[9500] ← A

4. **Immediate addressing mode:** In this mode, the operand position is the immediate data. For 8-bit data, instruction size is 2 bytes and for 16 bit data, instruction size is 3 bytes.
E.g. MVI A, 32H
LXI B, 4567H

5. **Implied or Inherent addressing mode:** The instructions of this mode do not have operands.
E.g. NOP: No operation
HLT: Halt
EI: Enable interrupt
DI: Disable interrupt

8. Write a program to reverse a given a string using 16 bit microprocessor.

Ans:

```
.MODEL SMALL
.STACK
.DATA
STRING DB '!EMOC-LEW'
REVERSE DB 9 DUP(' ')
.CODE
;-------------------------------------------------------
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX

  LEA SI,STRING  ;Load Effective Address of STRING  into SI
    LEA DI,REVERSE ;Load Effective Address of REVERSE into DI
    ADD DI,9      ;Add the Address of DI With 9
    MOV CX,9      ;Initialize Counter as 9
    TOP:
        MOV AL,[SI] ;Content of SI to AL
        MOV [DI],AL ;Content of AL to Content of DI
        INC SI      ;Increment SI
        DEC DI      ;Decrement DI
        LOOP TOP    ;Loop Until  CX!=0
    ADD DI,10  ;Add DI With 10 to Locate to End
    MOV AL,'$'         ;Add String Termination Character
    MOV [DI],AL
    MOV AH,09H     ;AH=09 Specifies String
    LEA DX,REVERSE ;Load Effective Address of REVERSE into DX
    INT 21H            ;DOS Interrupt Function
    MOV AH,4CH
    INT 21H
MAIN ENDP
END MAIN
```

9. Explain memory interfacing in 8085 microprocessor along with appropriate diagram.

Ans: The programs and data that are executed by the microprocessor have to be stored in ROM/EPROM and RAM, which are basically semiconductor memory chips.

Microprocessor need to access memory quite frequently to read instructions and data stored in memory, the interface circuit enables that access.

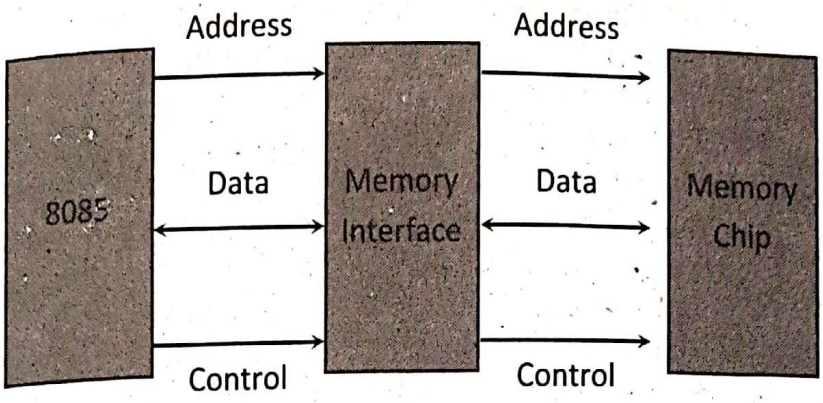


Figure: 8085 interfacing with memory chips

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 (AO – A15), hence a maximum of 64 KB (= 216 bytes) of memory locations can be interfaced with it.
- The memory address space of the 8085 takes values from 0000H to FFFFH.
- The 8085 initiates set of signals such as IO/M, RD' and WR' when it wants to read from and write into memory.
- Similarly, each memory chip has signals such as CE or CS (chip enable or chip select), OE or RD' (output enable or read) and WE or WR' (write enable or write) associated with it.

Generation of Control Signals for Memory

When the 8085 wants to read from and write into memory, it activates IOM, RD and WR signals as shown. Status of IQ/M, RD' and WR' signals during memory read and write operations

IOM'	RD'	WR'	Operation
0	0	1	8085 reads data from memory
0	1	0	8085 writes data into memory

Using IO/M, RD and WR signals, two control signals MEMR (memory read) and MEMW (memory write) are generated. Figure shows the circuit used to generate these signals.

Generating Control Signals

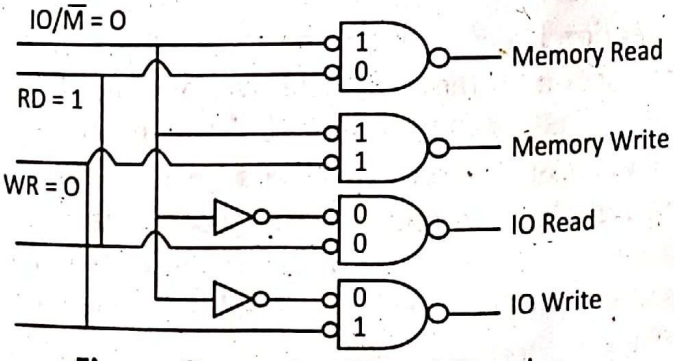


Figure: Generating Control Signals

During the execution the partial results of the previously executed portions are again fetched into the physical memory, if required for further execution. The procedure of fetching the chosen program segments or data from the secondary storage into physical memory is called swapping. The procedure of storing back the partial results or data back on the secondary storage is called unswapping. The virtual memory is allotted per task.

The 80286 is able to address 1 G byte (230 bytes) of virtual memory per task. The complete virtual memory is mapped on to the 16Mbyte physical memory. If a program larger than 16Mbyte is stored on the hard disk and is to be executed, if it is fetched in terms of data or program segments of less than 16Mbyte in size into the program memory by swapping sequentially as per sequence of execution.

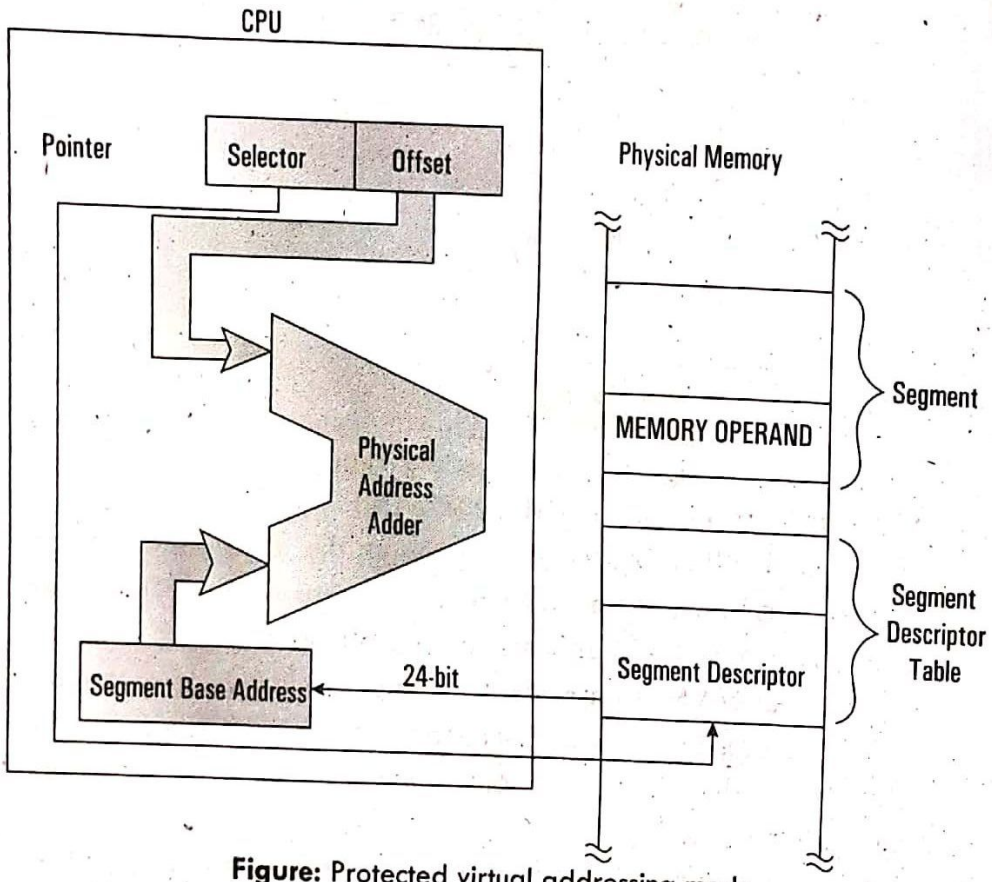


Figure: Protected virtual addressing mode

Whenever the portion of a program is required for execution by the CPU, it is fetched from the secondary memory and placed in the physical memory is called swapping in of the program. A portion of the program or important partial results required for further execution, may be saved back on secondary storage to make the PM free for further execution of another required portion of the program is called swapping out of the executable program.

80286 uses the 16-bit content of a segment register as a selector to address a descriptor stored in the physical memory. The descriptor is a block of contiguous memory locations containing information of a segment, like segment base address, segment limit, segment type, privilege level, segment availability in physical memory, descriptor type and segment use another task.

11. **"Interrupt based I/O is efficient compared to polled I/O". Justify this statement withgeneral working mechanism in both methods.**

Ans: **Polled interrupt:**

Polled interrupt are handled using software and are there fores lower compared to vectored (hardware) interrupts. In this method there is one common branch address for all interrupts. The program that takes care of interrupts begins at the branch address and polls the interrupts sources in sequence. The order in which they are tested determines the priority of each interrupt. The highest priority source is tested first, and if its interrupt signal is on, control branches to a servicer outline for this source. Otherwise the next lower priority source is tested, and so on. Thus, the initial service routine for all interrupts consists of a program that test the interrupt sources in sequence and branches to one of many possible service routines.

Polled interrupts are very simple. But or large number of devices, the time required to poll each device may exceed the service to the device. In such case, the faster mechanism called chained interruptisused.

Chained interrupt:

This is hardware concept to f handling the multiple interrupts. In this technique, the devices are connected in a chain fashion as shown in figure below for setting up the priority system.

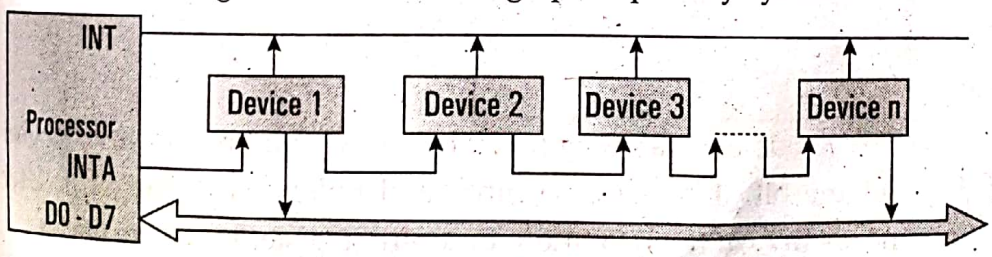


Figure: Chained interrupt handling.

Here the device with the highest priority placed in the first position, followed by lower priority devices. Suppose that one or more devices interrupt the process or at a time. In response, the process or saves its current status and then generates an interrupt acknowledge (INTA) signal to the highest priority device, which is

device 1 in our case. If this device has generated the interrupt it will accept the INTA signal from the processor; otherwise, it will pass INTA on to the next device until the INTA is accepted by the interrupting device.

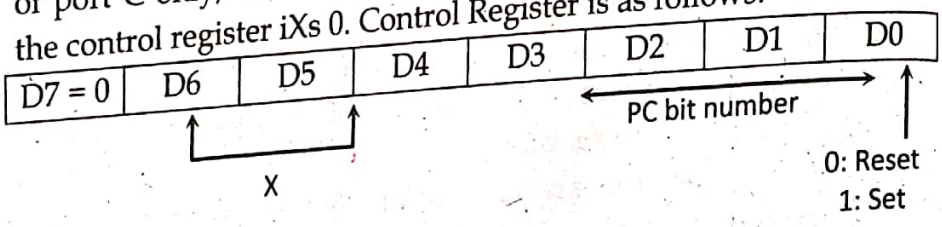
Once accepted, the device provides a means to the process or for finding the interrupt address vector using external hardware. Usually there questing device responds by placing a word on the data lines. With the help of hardware it generates interrupts vector address. This word is referred to as vector, which the process or used as a pointer to the appropriate device service routine. This avoids the need to execute a general interrupt service routine first. So this technique is also referred to as vectored interrupts.

12. Write Short Notes (Any Two):

a) Macro Assembler

b) BSR Mode

Ans: **Bit set reset (BSR) mode –** This mode is used to set or reset the bits of port C only, and selected when the most significant bit (D7) in the control register iXs 0. Control Register is as follows:



D3	D2	D1
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	0
1	1	0
1	1	1

This mode affects only one bit of port C at a time because, as user set the bit, it remains set until and unless user changes it. User needs to load the bit pattern in control register to change the bit.

c) System Bus

Ans: **System Bus:** It is a communication path between the microprocessor and peripherals; it is nothing but a group of wires to carry bits.

Data Bus : A collection of wires through which data is transmitted from one part of a computer to another is called Data Bus. Data Bus can be thought of as a highway on which data travels within a computer. This bus connects all the computer components to the CPU and main memory. The size (width) of bus determines how much data can be transmitted at one time.

E.g.:

- A 16-bit bus can transmit 16 bits of data at a time.
- 32-bit bus can transmit 32 bits at a time.

Address Bus

A collection of wires used to identify particular location in main memory is called Address Bus. In other words, the information used to describe the memory locations travels along the address bus. The size of address bus determines how many unique memory locations can be addressed.

E.g.:

- A system with 4-bit address bus can address 24 = 16 Bytes of memory.
- A system with 16-bit address bus can address 216 = 64 KB of memory.
- A system with 20-bit address bus can address 220 = 1 MB of memory.

Control Bus: The connections that carry control information between the CPU and other devices within the computer is called Control Bus. The control bus carries signals that report the status of various devices.

E.g.:

- This bus is used to indicate whether the CPU is reading from memory or writing to memory.

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Bachelor Level/ First Year/ Second Semester/ Science Full Marks: 60

Microprocessor (CSC 162) Pass Marks: 24

Time: 3 hours.

TU QUESTIONS-ANSWERS 2075

Candidates are required to give their answers in their own words as for as practicable.

The figures in the margin indicate full marks.

Group A (Long Answer Question Section) (2x10=20)

Attempt any TWO questions.

1. Draw the block diagram of 80286 microprocessor and explain its functional units.

Ans: The functional block diagram of 80286 microprocessor is as given below:

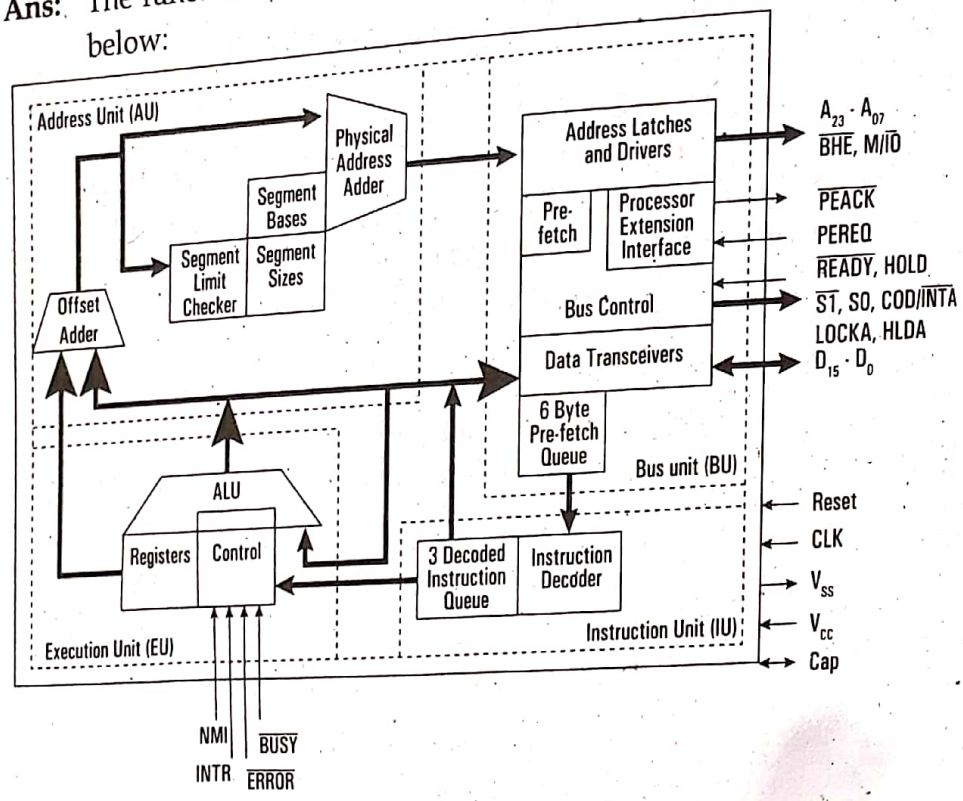


Figure: Internal Block Diagram of 80286

The CPU contain four functional blocks

1. Address Unit (AU)
2. Bus Unit (BU)
3. Instruction Unit (IU)
4. Execution Unit(EU)

The address unit is responsible for calculating the physical address of instructions and data that the CPU wants to access. Also the address lines derived by this unit may be used to address different peripherals. The physical address computed by the address unit is handed over to the bus unit (BU) of the CPU. Major function of the bus unit is to fetch instruction bytes from the memory. Instructions are fetched in advance and stored in a queue to enable faster execution of the instructions. The bus unit also contains a bus control module that controls the prefetcher module. These prefetched instructions are arranged in a 6-byte instructions queue. The 6-byte prefetch queue forwards the instructions arranged in it to the instruction unit (IU). The instruction unit accepts instructions from the prefetch queue and an instruction decoder decodes them one by one. The decoded instructions are latched onto a decoded instruction queue. The output of the decoding circuit drives a control circuit in the execution unit, which is responsible for executing the instructions received from decoded instruction queue. The decoded instruction queue sends the data part of the instruction over the data bus. The EU contains the register bank used for storing the data as scratch pad, or used as special purpose registers. The ALU, the heart of the EU, carries out all the arithmetic and logical operations and sends the results over the data bus or back to the register bank.

2. **Explain instruction cycle, machine cycle and T-states? Draw a timing diagram for STA instruction. Make necessary assumptions.**

Ans: **Instruction cycle**

The necessary steps that the CPU carries out to fetch an instruction and necessary data from the memory and to execute it constitute and instruction cycle 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 fetch an opcode from memory constitute a fetch cycle. 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 and execute cycle. The total time required to execute an instruction given by IC = Fc+ Ec the 8085 consists of 1-6 machine cycles or operations.

Fetch cycle:

The first byte of an instruction is its opcode. The program counter keeps the memory address of the next instruction to be executed in

the beginning of fetch cycle the content of the program counter, which is the address of the memory location where opcode is available, is send to the memory. The memory places the opcode on the data bus so as to transfer it to CPU. The entire process takes 3 clock cycle.

Execute cycle/Operation:

The opcode fetched from the memory goes to IR from the IR it goes to the decoder which decodes instruction. After the instruction is decoded execution begins.

- If the operand is in general purpose register, execution is performed immediately. I,e in one clock cycle.
- If an instruction is contains data or operand address, then CPU has to perform some read operations to get the desired data.
- In some instruction write operation is performed. In write cycle data are sent from the CPU to the memory of an o/p device.
- In some cases execute cycle may involve one or more read or write cycle or both.

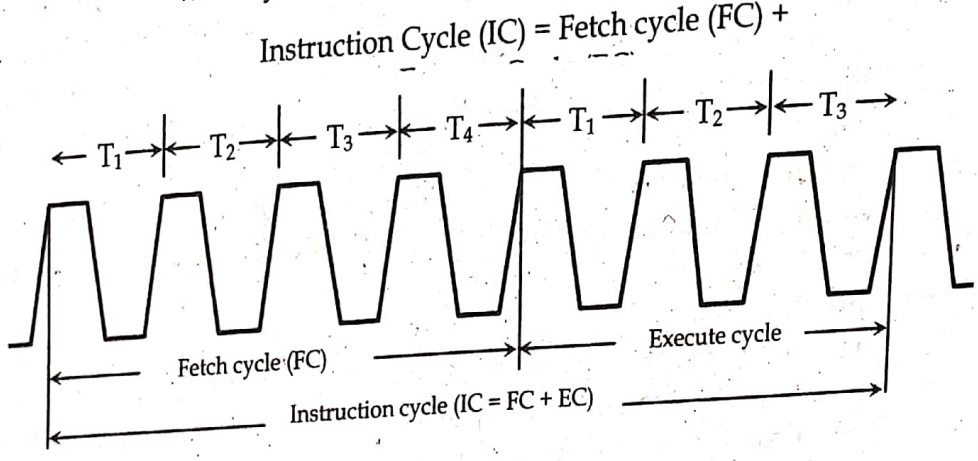


Figure: (a) Processor cycle

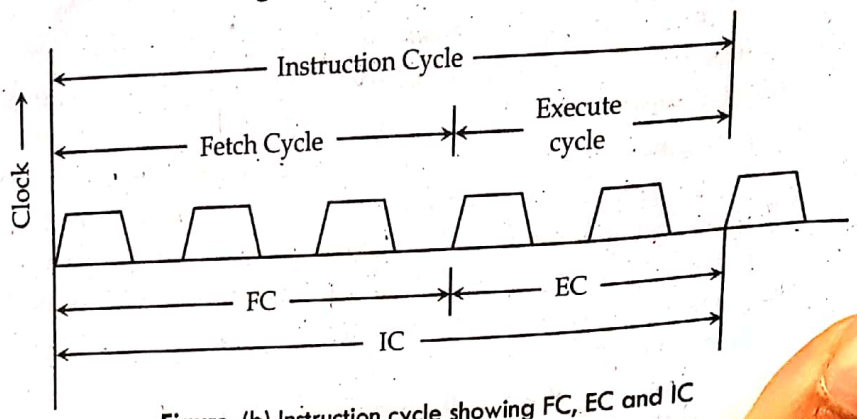


Figure: (b) Instruction cycle showing FC, EC and IC

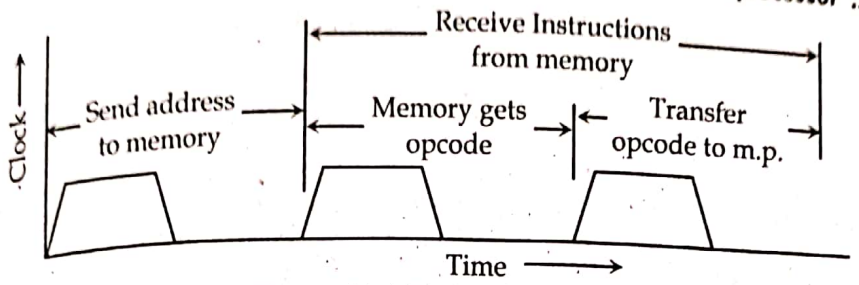


Figure: (c) A Typical Fetch cycle

Machine cycle

It is defined as the time required to complete one operation of accessing memory i/p, o/p or acknowledging and external request. This cycle may consists 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.

Machine cycles of 8085

The 8085 microprocessor has 5 (seven) basic machine cycles. They are

- ✓ Opcode fetch cycle (4T)
- ✓ Memory read cycle (3 T)
- ✓ Memory write cycle (3 T)
- ✓ I/O read cycle (3 T)
- ✓ I/O write cycle (3 T)
- ✓ Interrupt

Time period, T = 1/f; where f = Internal clock frequency

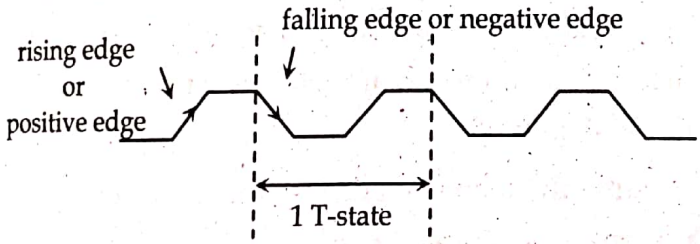


Figure: Clock Signal.

Machine Cycle Status and Control Signals

Table 5.1 (a) Machine cycle status and control signals

	Status			Controls		
Machine cycle	IO/	S1	S0	RD	WR	INTA
Opcode Fetch (OF)	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read (I/OR)	1	1	0	0	1	1
I/O Write (I/OW)	1	1	1	1	0	1
Acknowledge of INTR (INTA)	1	0	1	1	1	0
BUS Idle (BI) : DAP	0	1	0	1	1	1
ACK of RST, TRAP	1	1	1	1	1	1
HALT	Z	1	0	Z	Z	1
HOLD	Z	0	X	Z	Z	1

X ⇒ Unspecified, and Z ⇒ High impedance state

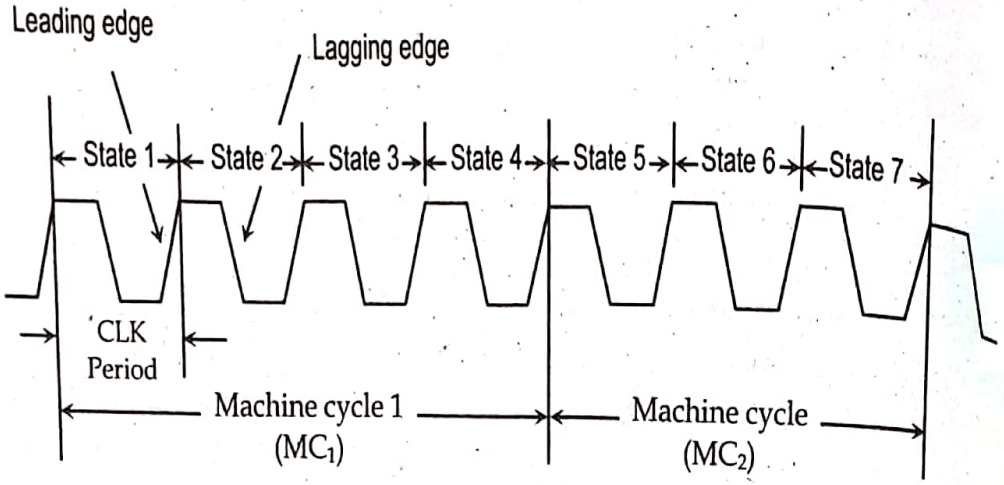
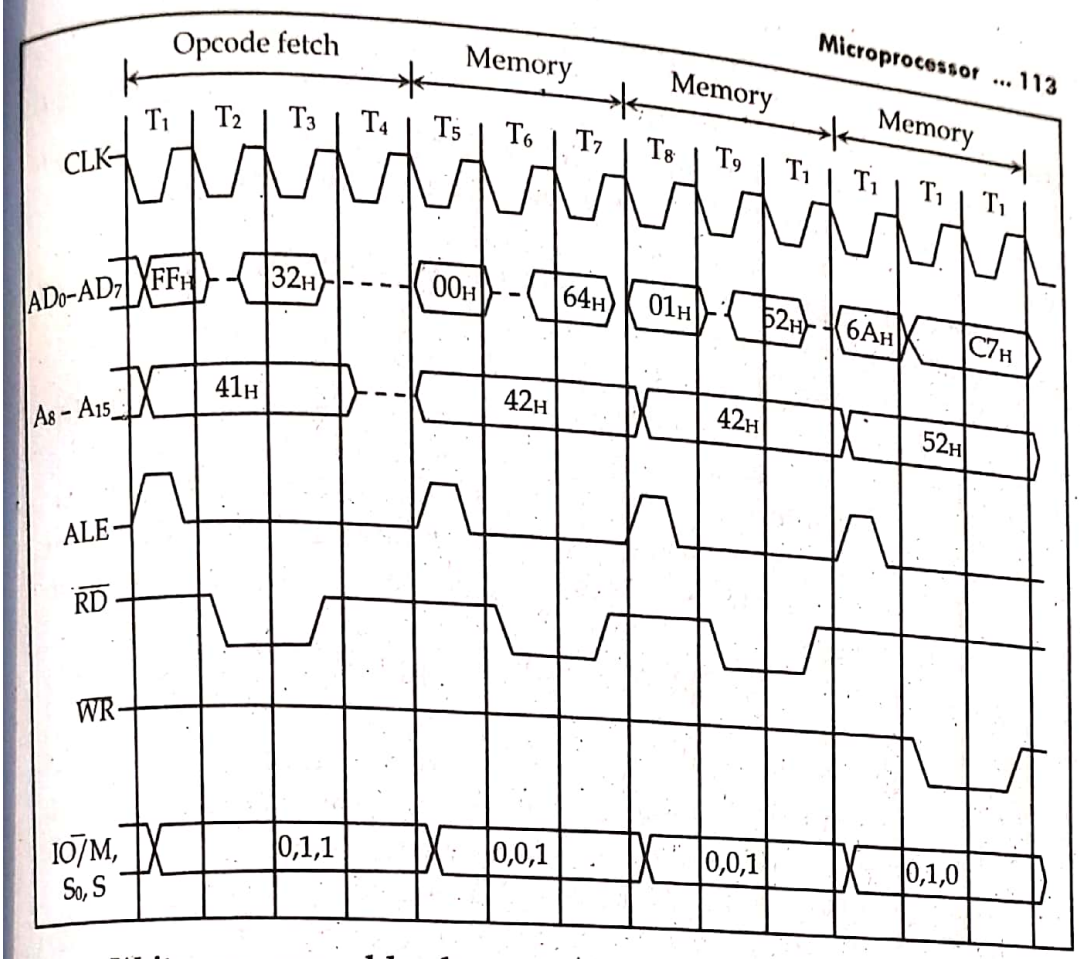


Figure: Machine cycle showing clock periods.

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	Memonics	Op code
41FF	STA 526 AH	32_H
4200		$6A_H$
4201		52_H



3. Write an assembly language program to find the smallest number in an array using 8 bit microprocessor. (Assume appropriate array data and address where minimum array size of 15 should be considered.)

Ans:

```
LXI H 8D01      ; LOAD COUNTER TO MEMORY ADDRESS 8D01
MOV B M         ; ASSIGNS B AS COUNTER
INX H           ; POINTS WHERE DATA ARE STORED
MOV A M         ; CONTENT TRANSFERRED TO ACCUMULATOR
DCR B           ; COUNTER VALUE DECREASED
LOOP: INX H     ; POINTS TO ANOTHER MEMORY ADDRESS
CMP M           ; ADDRESSED MEMORY AND
                  ACCUMULATOR ARE COMPARED
JNC/JC AHEAD    ; JUMPS IF ACCUMULATOR IS GREATER
MOV A M         ; ACCUMULATOR CARRIES GREATER VALUE
AHEAD: DCR B    ; DECREASE OF COUNTER
JNZ LOOP        ; IF COUNTER IS NOT ZERO REPEAT THE
                  PROCESS
STA 8D00 H      ; STORES IN MEMROY ADDRESS 8D00 HLT
```

Group B (Short Answer Question Section)

Attempt any EIGHT questions. (8x5=40)

4. Differentiate between vectored and non-vectored interrupt. Where and how 8259 PIC can be used to handle interrupts.

Ans: Vectored Interrupt

In vectored interrupts, the processor automatically branches to the specific address in response to an interrupt.

Non-Vectored Interrupt

But in non-vectored interrupts the interrupted device should give the address of the interrupt service routine (ISR).

In vectored interrupts, the manufacturer fixes the address of the ISR to which the program control is to be transferred. The vector addresses of hardware interrupts are given in table above in previous page.

- The TRAP, RST 7.5, RST 6.5 and RST 5.5 are vectored interrupts.
- The INTR is a non-vectored interrupt. Hence when a device interrupts through INTR, it has to supply the address of ISR after receiving interrupt acknowledge signal.

The 8259 Programmable Interrupt Controller

The 8259A programmable interrupt controller designed to work with Intel microprocessors 8085, 8086 and 8088. The 8259A interrupt controller can

1. Manage eight interrupts according to the instructions written into its control registers. This is equivalent to proving eight interrupt pins on the processor in place of one INTR (8085) pin.
2. Vector can interrupt request anywhere in the memory map. However, all eight interrupts are spaced at the interval of either four or eight locations. This eliminates all the major drawback of the 8085 interrupts in which all interrupts are vectored to memory locations on page 00H.
3. Resolve eight levels of interrupt priorities in a variety of modes, such as fully nested mode, automatic rotation mode, and specific rotation mode.
4. Mask each interrupt request individually.
5. Read the status of pending interrupts, in-service interrupts, and masked interrupts.
6. Be set up to accept either the level-triggered or the edge-triggered interrupt request
7. Be expanded to 64 priority levels by cascading additional 8259As.
8. Be set up to work with either the 8085 microprocessor mode or the 886/8088 microprocessor mode.

The 8259A is upward-compatible with its predecessor, the 8259. The main difference between the two is that the 8259A can be used with Intel's 8086/88 16-bit microprocessor. It also includes additional features such as the level-triggered mode, buffered mode, and automatic-end-of interrupt mode. To simplify the explanation of the 8259A, illustrative examples will not include the cascade mode or the 8086/88 mode and will be limited to modes continuously used with the 8085.

5. Explain addressing modes of 8085 microprocessor with examples.

Ans: Addressing modes:

Instructions are command to perform a certain task in microprocessor. The instruction consists of op-code and data called operand. The operand may be the source only, destination only or both of them. In these instructions, the source can be a register, a memory or an input port. Similarly, destination can be a register, a memory location, or an output port. The various format (way) of specifying the operands are called addressing mode. So addressing mode specifies where the operands are located rather than their nature. The 8085 has 5 addressing mode:

1) **Direct addressing mode:**

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

E. g. LDA 9500H A ← [9500]

IN 80H A ← [80]

This type of addressing is called absolute addressing.

2) **Register Direct addressing mode:**

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

E g. MOV A, B

Here register B contains data rather than address of the data.

Other examples are: ADD, XCHG etc.

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 in this type of addressing mode, it is the address of the address rather than address itself. (One operand is register)

e. g. MOV R, M

MOV M, R

STAX, LDAX etc.

STAX B B= 95 C =00 [9500] ← A

4) **Immediate addressing mode:**

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

E.g. MVI A, 32H

LXI B, 4567H

5) **Implied or Inherent addressing mode:**

The instructions of this mode do not have operands.

E.g. NOP: No operation

HLT: Halt

EI: Enable interrupt

DI: Disable interrupt

6. **Write an ALP to read a string and display the string in uppercase.**

```
;Program To Change The String Into Toggle Case
;-------------------------------------------------------------
.MODEL SMALL
.STACK
.DATA
    STRING DB 'Welcome'
    CASE DB 7 DUP(' ')
.CODE
;-------------------------------------------------------------
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX
    LEA SI,STRING ;Load Effective Address of STRING into SI
    LEA DI,CASE  ;Load Effective Address of CASE
    MOV CX,7     ;Load Counter with 7
    TOP:
        CMP CX,0000H ;Compare CX with 0
        JE EXIT ;If CX=0 then Goto Exit
        MOV AH,[SI]  ;Load Content of SI into AH
CMP AH,60H  ;Compare AH with 60H i.6 96
        JA ISSMALL
        CMP AH,5AH  ;Compare AH with 5AH i.e 90
        JB ISCAP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;TO UPPERCASE
```

```
ISSMALL:
        AND AH,11011111B  ;Mask With 11011111B
        MOV [DI],AH
        INC SI
        INC DI
DEC CX
        JMP TOP
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;TO LOWERCASE
ISCAP:
        OR AH,00100000B ;Mask With 00100000B
MOV [DI],AH
        INC SI
        INC DI
        DEC CX
        JMP TOP
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    EXIT:
    MOV AL,'$'      ;Add String Terminator.
      MOV [DI],AL  ;Pass the Content of AL to DI.
      MOV DX,OFFSET CASE
      MOV AH,09H
      INT 21H
      MOV AH,4CH
      INT 21H
MAIN ENDP
END MAIN
                              with DI
;---------------------------------------------------------------
```

7. **What is system bus? Explain different types of system bus in detail.**

Ans: **System Bus:**

It is a communication path between the microprocessor and peripherals; it is nothing but a group of wires to carry bits.

Data Bus: A collection of wires through which data is transmitted from one part of a computer to another is called Data Bus. Data Bus can be thought of as a highway on which data travels within a computer. This bus connects all the computer components to the CPU and main memory. The size (width) of bus determines how much data can be transmitted at one time.

E.g.:

- A 16-bit bus can transmit 16 bits of data at a time.
- 32-bit bus can transmit 32 bits at a time.

Address Bus:

A collection of wires used to identify particular location in main memory is called Address Bus. In other words, the information used to describe the memory locations travels along the address bus. The size of address bus determines how many unique memory locations can be addressed.

E.g.:

- A system with 4-bit address bus can address 24 = 16 Bytes of memory.
- A system with 16-bit address bus can address 216 = 64 KB of memory.
- A system with 20-bit address bus can address 220 = 1 MB of memory.

Control Bus

The connections that carry control information between the CPU and other devices within the computer is called Control Bus. The control bus carries signals that report the status of various devices. E.g.: This bus is used to indicate whether the CPU is reading from memory or writing to memory.

8. **How DTE and DCE are wired using RS 232 cable? Explain the process of double handshaking I/o.**

Ans: RS 232C interface with DTE and DCE:

The figure below shows a interfacing with minimum lines.

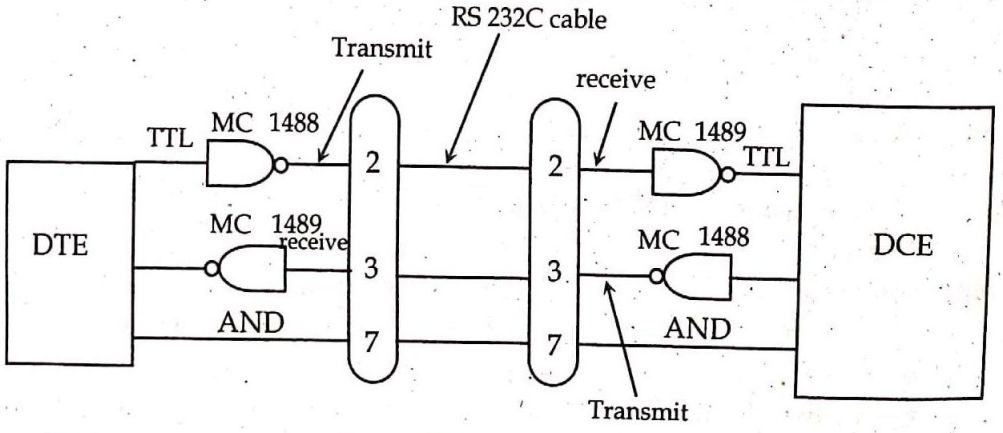


Figure: RS 232C interface

The signaling in RS-232C is not compatible with the TTL logic level. For TTL 0 v to 0.2V is considered a logic 0 and 3.4 v to 5v as logic 1. But RS-232C works in a negative logic -3 to -15v considered as logic 1 and +3 to +15v as logic 0. Because of this incompatibility of the data lines with the TTL logic, voltage translators called line

E.g.:

- A 16-bit bus can transmit 16 bits of data at a time.
- 32-bit bus can transmit 32 bits at a time.

Address Bus:

A collection of wires used to identify particular location in main memory is called Address Bus. In other words, the information used to describe the memory locations travels along the address bus. The size of address bus determines how many unique memory locations can be addressed.

E.g.:

- A system with 4-bit address bus can address 24 = 16 Bytes of memory.
- A system with 16-bit address bus can address 216 = 64 KB of memory.
- A system with 20-bit address bus can address 220 = 1 MB of memory.

Control Bus

The connections that carry control information between the CPU and other devices within the computer is called Control Bus. The control bus carries signals that report the status of various devices.

E.g.: This bus is used to indicate whether the CPU is reading from memory or writing to memory.

8. **How DTE and DCE are wired using RS 232 cable? Explain the process of double handshaking I/o.**

Ans: RS 232C interface with DTE and DCE:

The figure below shows a interfacing with minimum lines.

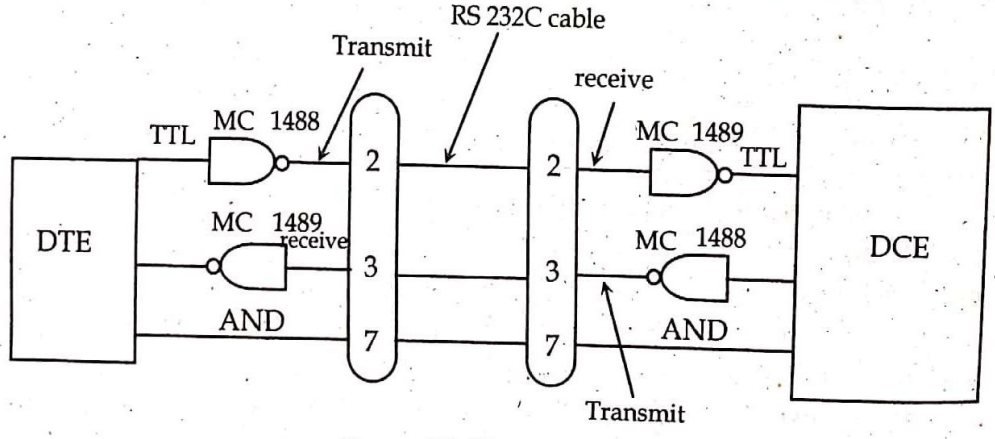


Figure: RS 232C interface

The signaling in RS-232C is not compatible with the TTL logic level. For TTL 0 v to 0.2V is considered a logic 0 and 3.4 v to 5v as logic 1. But RS-232C works in a negative logic -3 to -15v considered as logic 1 and +3 to +15v as logic 0. Because of this incompatibility of the data lines with the TTL logic, voltage translators called line

drivers and line receivers are required to interface TTL logic with RS-232C signals. The line driver MC 1488 converts logic 1 into approx. -9V and logic 0 into +9v. Before it is received by the DCE it is again converted by the line receiver MC 1489 into TTL-Compatible logic.

The minimum interface required both a computer and a peripheral device requires three lines; pin 2, 3 and 7. These lines are defined in relation to the DTE; the terminal transmits on pin 2 and receives as pin 3. On the other hand the DCE transmits on pin 3 and receives on pin 1. Pin 7 is ground pin.

DOUBLE HANDSHAKE I/O DATA TRANSFER

For data transfers where coordination is required between sending system and the receiving system, a double handshake is used.

- The sending device asserts its STB low to ask **„Are you ready?"**
- The receiving system raises its ACK line high to say **„I'm ready".**
- The peripheral device then sends the data byte and raises its STB signal high to say **"Here is valid data for you".**
- When the receiving system finishes to read the data, receiving system drop its ACK line low to say **"I have data than you and I await your request to send the next byte of data".**

What is instruction set? Explain different kinds of instruction set used in 8085 microprocessor.

Ans: **THE 8085 INSTRUCTION SETS**

An instruction is a binary pattern designed inside a microprocessor to perform a specific function (task). The entire group of instructions called the instruction set. The 8085 instruction set can be classified into 5- different groups:

- Data Transfer Instructions
- Arithmetic Instructions
- Branching Instructions
- Logical Instructions
- Control Instructions

DATA TRANSFER INSTRUCTIONS

It is the longest group of instructions in 8085. This group of instruction copy data from a source location to destination location without modifying the contents of the source. The transfer of data may be between the registers or between register and memory or between an I/O device and accumulator. None of these instructions changes the flag.

ARITHMETIC INSTRUCTIONS

The 8085 microprocessor performs various arithmetic operations such as addition, subtraction, increment and decrement.

The arithmetic operation add and subtract are performed in relation to the contents of accumulator. The features of these instructions are

- They assume implicitly that the accumulator is one of the operands.
- They modify all the flags according to the data conditions of the result. They place the result in the accumulator.
- They do not affect the contents of operand register or memory. But the INR and DCR operations can be performed in any register or memory. These instructions

✓ Affect the contents of specified register or memory.

✓ Affect the flag except carry flag.

BRANCHINGINSTRUCTIONS

The microprocessor is a sequential machine; it executes machine codes from one memory location to the next. The branching instructions instruct the microprocessor to go to a different memory location and the microprocessor continues executing machine codes from that new location.

LOGICAL INSTRUCTIONS

A microprocessor is basically a programmable logic chip. It can perform all the logic functions of the hardwired logic through its instruction set. The 8085 instruction set includes such logic functions as AND, OR, XOR and NOT (Complement).

10. **What is mean by memory interfacing? Explain the address decoding process in 8085 microprocessor with 3 X 8 decoder.**

Ans: Interfacing a microprocessor is to connect it with various peripherals to perform various operations to obtain a desired output.

Memory Interfacing and I/O Interfacing are the two main types of interfacing.

Memory Interfacing is used when the microprocessor needs to access memory frequently for reading and writing data stored in the memory. It is used when reading/writing to a specific register of a memory chip.

I/O Interfacing is achieved by connecting keyboard (input) and display monitors (output) with the microprocessor.

The method use by the system to select the correct location on the correct chip is called addressing decoding. In another way, it is the process of generating a chip select (CS) or enable signals from the address bus for each device in the system. The address bus lines are split into two sections:

- The N, MSB are used to generate a chip select (CS) signal for different device.
- The M, LSB are passed in the device as address to the different memory cells or internal registers.

ADDRESS BUS

N- bits to decoder(MSB) | M- bits to memory (LSB)

Address decoding can be done using 3 to 8 decoder.

- Assume a microprocessor with 10 address lines that give total of mapping 1 KB memory.
- Let us consider that we want to implement all its memory space and we use 128*8 memory chips.
- So we have 8 memory chips with 128 bytes storage, so we need 3 address lines connected with the decoder to generate 8 different chip select signals and select one of the 8 chips.
- And 7 address lines to select a particular memory word of the selected chip.

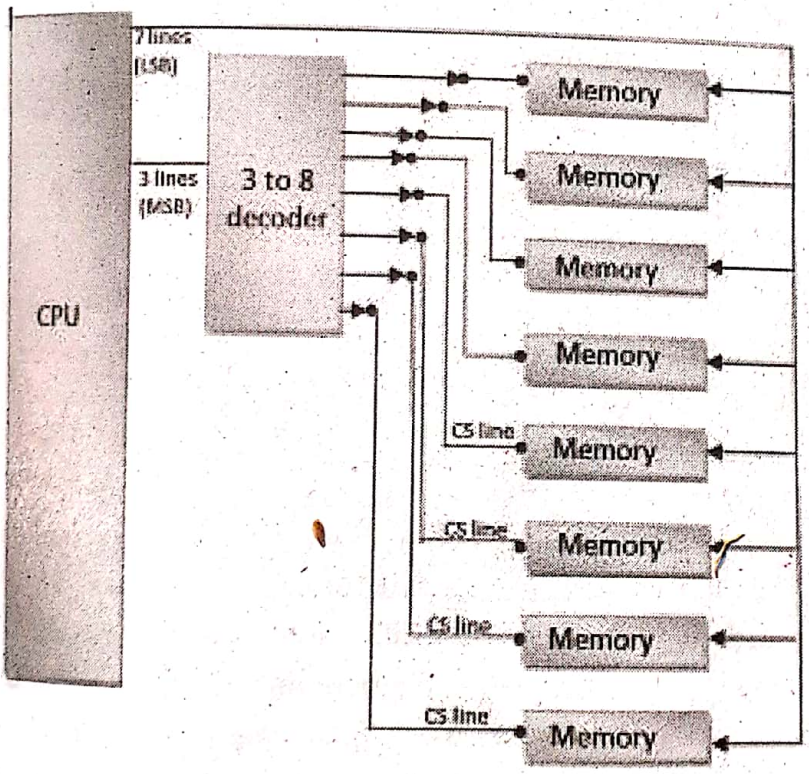


Figure: 3 to 8 address decoding.

11. **Explain how pipelining is achieved in 8086 microprocessor.**

Ans: **Pipelining:**

Pipelining is an implementation technique where multiple instructions are overlapped in execution. The computer pipeline is divided in**stages**. Each stage completes a part of an instruction in parallel. The stages are connected one to the next to form a pipe - instructions enter at one end, progress through the stages, and exit at the other end.

Pipelining does not decrease the time for individual instruction execution. Instead, it increases instruction throughput. The **throughput** of the instruction pipeline is determined by how often an instruction exits the pipeline.

Because the pipe stages are hooked together, all the stages must be ready to proceed at the same time. We call the time required to move an instruction one step further in the pipeline ***a machine cycle***. The *length* of the machine cycle is determined by the time required for the slowest pipe stage.

The pipeline designer's *goal is to balance the length of each pipeline stage*. If the stages are perfectly balanced, then the time per instruction on the pipelined machine is equal to

Time per instruction on nonpipelined machine

Number of pipe stages

Under these conditions, the speedup from pipelining equals the number of pipe stages. Usually, however, the stages will not be perfectly balanced; besides, the pipelining itself involves some overhead.

Instruction Pipeline:

Any architecture can be pipelined by making each clock cycle into a pipe stage.

Clock#	1	2	3	4	5	6	7
Instruction i	Fetch	Decode	Execute				
Instr. i+1		Fetch	Decode	Execute			
Instr. i+2			Fetch	Decode	Execute		
Instr. i+3				Fetch	Decode	Execute	
Instr. i+4					Fetch	Decode	Execute

Here instruction i is fetched in clock #1. After it has been fetched it is decoded in clock #2 and at the same time next instruction i.e. instr. i+1 is fetched. At Clock#3, instruction i is executed, instr. i+1 is decoded and instr. i+2 is fetched and so on.

Hence at the end of clock #3, instruction i is executed. Sometime later at the end of clock #4 instruction i+1 is executed.

If we assume that unpipelined architecture took 3ns then this pipelined architecture will take 1ns to finish a stage (fetch, decode and execute).

Advantages of pipelining:

- The execution unit always reads the next instruction byte from the queue in BIU. This is faster than sending out an address to the memory and waiting for the next instruction byte to come.

- In short pipelining eliminates the waiting time of EU and speeds up the processing. The 8086 BIU will not initiate a fetch unless and until there are two empty bytes in queue. 8086 BIU normally obtains two instruction bytes per fetch.

12. Write short notes on: (any two)

a. Von Neumann architecture

Ans: 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 task of entering and altering the programs for ENIAC was tedious. It could be facilitated if the program could be represented in a form suitable for storing in memory alongside the data. So the computer could get its instructions 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' was first adopted by John Von Neumann and such architecture is named as von-Neumann architecture and shown in figure below.

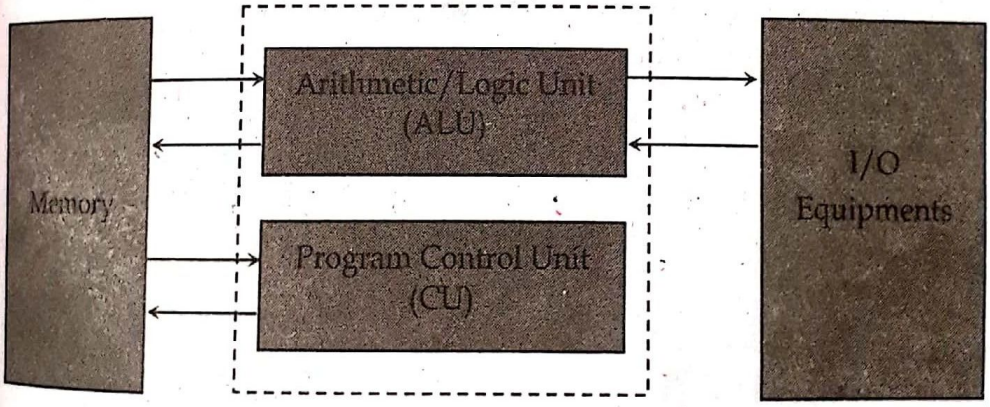


Figure: Von–Neumann Architecture.

The main memory is used to stare both data and instructions. The arithmetic and logic unit is capable of performing arithmetic and logical operation on binary data. The program control unit interprets the instruction in memory and causes them to be executed. The I/O unit gets operated from the control unit.

The Von-Neumann architecture is the fundamental basis for the architecture of modern digital computers. It consisted of 1000 storage locations which can hold words of 40 binary digits and both instructions as well as data are stored in it. The storage location of control unit and ALU are called registers and the

various models of registers are:

MAR – memory address register – contains the address in memory of the word to be written into or read from MBR.

MBR – memory buffer register – consists of a word to be stored in or received from memory.

IR – instruction register – contains the 8-bit op-code instruction to be executed.

IBR – instruction buffer register – used to temporarily hold the instruction from a word in memory.

PC - program counter - contains the address of the next instruction to be fetched from memory.

AC & MQ (Accumulator and Multiplier Quotient) - holds the operands and results of ALU after processing.

b. Macro assembler

Ans: A macro assembler is able to generate a program segment, which is defined by a macro, when the name of the macro appears as an opcode in a program. The macro assembler is still capable of regular assembler functioning, generating a machine instruction for each line of assembly language code; but like a compiler, it can generate many machine instructions from one line of source code. Its instruction set can be expanded to include new mnemonics, which generate these program segments of machine code. The following discussion of how a macro works will show how this can be done.

A frequently used program segment can be written just once, in the macro definition at the beginning of a program. For example, the

macro

```
LFEED MACRO
MOV AH, 06H
MOV DL, 0AH
INT 21H
MOV DL, 0DH
INT 21H
ENDM
```

Is used to put down the cursor in next line.