Electrical Engineering

Microprocessors

Comprehensive Theory

with Solved Examples and Practice Questions

MADE EASY
Publications

MADE EASY Publications

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016
E-mail: infomep@madeeasy.in
Contact: 011-45124660, 8860378007
Visit us at: www.madeeasypublications.org

Microprocessors

First Edition : 2015
Second Edition : 2016
Third Edition : 2017
Fourth Edition : 2018

Contents

Microprocessors

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

■■■■

CHAPTER 1

Introduction to 8085 and Its Functional Organization

1.1 Introduction

The most important technological invention of modern times is the "microprocessor". A microprocessor is a multiple purpose programmable clock driven, register based electronic device that reads binary instructions from memory, accepts binary data as input and processing this data according to the instructions written in the memory. The microprocessor is capable of performing computing functions and making decisions to change the sequence of program execution. The microprocessor can be embedded in a larger system, and can function as the CPU of a computer called a microcomputer.

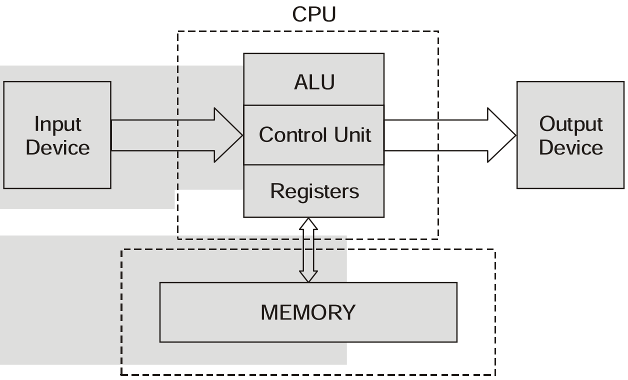

***Figure*-1.1 :** *Block diagram of microcomputer*

The Figure 1.1 shows the basic block diagram of a microcomputer which processes binary data and traditionally represented by four blocks i.e. CPU, memory, input device and output device.

Here, input device is a device that transfers information from outside world to the computer for example: Key board, mouse, webcam, microphone, scanner, electronic white boards, etc. The output device transfers information from computer to the outside world like monitor, printers (all types), speakers, headphones, projector, plotter, Braille embosser, LCD projection panel, computer output microfilm (COM) etc. Memory is an electronic medium that stores binary information.

Central Processing Unit (CPU) is the heart of computer systems. The microprocessors in any microcomputer act as a CPU. The CPU can be made up with ALU + CU + Registers. Where ALU is the group of circuits that performs arithmetic and logical operations. Control Unit (CU) is a group of circuits that provide timings and signals to all the operations in the computer and controls the data flow.

Microcontroller is a programmable device that includes microprocessor, memory and I/O signal lines on a single chip, fabricated using VLSI technology. Microcontrollers are also known as single chip microcomputers. They are mostly used to perform dedicated functions such as automatic control of equipment, machines and process in industries and consumer appliances.

System Bus

A bus is a group of wires/lines used to transfer data (bits) between components inside a computer or between computers. In most simple form, they are communication path used to carry the signals between microprocessor and peripherals.

The system bus of a microprocessor is of three types:

1. **Address Bus**
   - It is a group of lines that are used to send a memory address or a device address from the Microprocessor Unit (MPU) to the memory or the peripheral.
   - The address bus is always uni-directional i.e address always goes out of the microprocessor.
   - If the address line are 'n' for a MPU then its addressing capacity is 2^n.
2. **Data Bus**
   - It is group of lines used to transfer data between the microprocessor and peripherals and/or memory.
   - Data bus is always bi-directional.
3. **Control Bus**
   - Control bus provides signals to control the flow of data.

Do You Know: The internal architecture of the microprocessor unit depends on the data bus width, which is equal to the bit-capacity of the microprocessor.

1.2 History of Microprocessors

A brief review of certain microprocessors were given in the Table 1.1. Intel introduced its first 4-bit PMOS microprocessor 4004 in the year 1971. It has 16 pins, 640-bytes of memory addressing capability and 10 address lines. After this enhanced version of 4004, a 4-bit, Intel 4040 was developed. In 1972, Intel introduced its first 8-bit processor Intel 8008, which also uses PMOS technology. The PMOS technology processors were slow and not compatible with TTL logic. These microprocessors could not survive as general purpose microprocessor due to design limitations. In 1974, Intel introduced its more powerful and faster 8 bit NMOS microprocessor Intel 8080. These processors were faster and compatible with TTL logic. Intel 8085 followed 8080 microprocessor. The main limitations of 8 bit microprocessors tempted the designers to go for more powerful processors in terms of advanced architecture, more processing capability, larger memory addressing capability and more powerful instruction set. The Intel 8086 was the result, launched in 1978 . The technology used was HMOS, high speed and high performance MOS technology.

Microprocessor	Word length	Memory capacity
Intel 4004 (PMOS)	4-bit	640 B
Intel 8008	8-bit	16 kB
Intel 8080 (NMOS)	8-bit	64 kB
Intel 8085 (NMOS)	8-bit	64 kB
Intel 8086 (HMOS)	16-bit	1 MB
Intel 8088	8/16-bit	1 MB
Intel 80186	16-bit	1 MB
Intel 80286	16-bit	16 MB real, 4 GB virtual
Intel 80386	32-bit	4 GB real, 4 GB virtual
Intel 80486	32-bit	4 GB real, 64 TB virtual
Pentium-II	64-bit	64 GB real
Z-80	8-bit	64 kB
Z-800	8-bit	500 kB

Table 1.1 : *A brief review of various microprocessors*

NOTE: Most of the general purpose microprocessors used in the modern world computers are the family of 8086.

1.3 Computer Language

- Scale of integration:
  - **SSI (Small Scale Integration):** The term refers to the technology used to fabricate discrete logic gates on a chip.
  - **MSI (Medium Scale Integration):** The process of designing few tens of gates on a single chip.
  - **LSI (Large Scale Integration):** The process of designing hundreds of gates on a single chip similarly terms VLSI (very large scale integration), ULSI (ultra large scale integration) are used to indicate the scale of integration.
- **Digital computer:** A programmable machine that process binary data. It is traditionally represented by five components: CPU, ALU, CU, memory, input and output.
- **Instruction:** a command in binary that is recognized and executed by the computer in order to accomplish a task. Some instructions are designed with one word, and some require multiple words.
- **Mnemonic:** a combination of letters to suggest the operation of an instruction.
- **Program:** a set of instructions written in a specific sequence for the computer to accomplish a given task.
- **Machine Language:** the binary medium of communication with a computer through a designed set of instructions specific to each computer.
- **Assembly Language:** a medium of communication with a computer in which programs are written in mnemonics. An assembly language is specific to a given computer.
- **Low-Level Language:** a medium of communication that is machine-dependent or specific to a given computer. The machine and the assembly languages of a computer are considered low-level languages. Programs written in these languages are not transferrable to different types of machines.
- **High-Level Language:** a medium of communication that is independent of a given computer. Programs are written in English-like words, and they can be executed on a machine using a written translator (a compiler or an interpreter).
- **Compiler:** a program that translates English-like words of a high-level language into the machine language of a computer. A compiler reads a given program, called a source code, in its entirety, and then translates the program into the machine language which is called an object code. (Ex. *C*, *C*++)
- **Interpreter:** a program that translates the English-like statements of a high-level language into the machine language of a computer. An interpreter translates one statement at a time from a source code to an object code. (Ex. BASIC)
- **Assembler:** a computer program that translates an assembly language program from mnemonics to the binary machine code of a computer and these machine codes are called object programme .

  Difference between compiler and interpreter: Interpreter reads one line at a time, converts it into object code, executes and then reads next line. Whereas compiler reads whole program at a time and convert it into the object code and then execute.
- **Bit:** a binary digit, 0 or 1.
- **Byte:** a group of eight bits.
- **Nibble:** a group of four bits.
- **Word:** a group of byte the computer recognizes and processes at a time.

Summary

- The 8085 microprocessor (μP) is an improved version of 8080 A.
- 8085 μP has 74 instruction sets.
- The programming of 8085 μP is done in Assembly language.
- There are 27 pins (16 + 1 + 1 + 9) for output in a 8085 μP.
- There are 21 pins for input in a 8085 μP.
- In 8085 μP, memory it contains 2^{16} address line or 64 K or 65536 memory locations and each location can stores 8 bit. So, we can say the memory capacity of 8085 μP equals to 64 k × 8 bit ≈ 64 k bytes.
- A "TRISTATE DEVICE" has 3 states, two logic states (1 or 0) and one high impedance state.

  When device is disabled, it remains in high impedance state and doesn't draw any current from the system.
- To interconnect peripherals with the microprocessor, additional logic circuitry (Buffers, Decoders, Encoders and Latches) are needed.
- Performance of "Cache Memory" are measured in "Hit ratio".
- An I/O processor controls the flow of information between main memory and I/O devices.
- "Cache Memory" is a small high-speed memory placed between the CPU and the main memory (RAM).
- When a CPU is interrupted, it acknowledges interrupt and branches to a subroutine.
- The reference bit is used for the purpose of implementing NRU (Not recently used) algorithm.
- The larger the RAM of a computer, the faster is its speed, since it eliminates frequent disk I/Os.
- An "Assembler" is used for translation of a program from assembly language to Machine language.

Student's Assignments 1

Q.1 The number of flip-flops in a flag register of INTEL 8085 are ___________.

Q.2 Maximum memory that can be connected to INTEL 8085 is __________ bytes.

Q.3 Explain the difference between a compiler an interpreter.

Q.4 Explain the functions of the ALE and $IO/\overline{M}$ signals of the 8085 μP.

Answer: (Conventional)

1. (8 or 5) 2. (65536)

Student's Assignments 2

Q.1 In 8085 microprocessor unit scratch pad memory comprises of

(a) *B*, *C*, *D*, *E*, *H* and *L* Registers

(b) *W*, *Z*, *B*, *C*, *D*, *E*, *H* and *L* Registers

(c) *W*, *Z*, *B*, *C*, *D* and *E* Registers

(d) *W*, *Z*, *B*, *C*, *D*, *E*, *H*, *L* and status Registers

Q.2 An interrupt in which the external device supplies its address as well as the interrupt request is known as
(a) vectored interrupt
(b) maskable interrupt
(c) polled interrupt
(d) non-maskable interrupt

Q.3 **Assertion (A):** The data bus and address bus of 8085 microprocessor are multiplexed.
Reason (R): Multiplexing reduces number of pins.
(a) Both A and R are correct and R is correct explanation of A.
(b) Both A and R are correct but R is not correct explanation of A.
(c) Only A is correct.
(d) Only R is correct.

Q.4. P : Program counter is the register which stores the address of the next instruction to be executed.
Q : Stack pointer stores the address of the top of the stack.
Out of these two statements, which statement/s is/are true?
(a) Only P (b) Only Q
(c) Both P and Q (d) None of them

Q.5 How many instructions does microprocessor 8085 has
(a) 255 (b) 256
(c) 246 (d) 250

Q.6 How many nibbles are there in 1 kbyte data?
(a) 500 (b) 1024
(c) 2048 (d) none of these

Q.7 Match **List-I (Interrupt)** with **List-II (Property):**

	List-I		List-II
P.	RST 7.5	1.	Non-maskable
Q.	RST 6.5	2.	Edge sensitive
R.	INTR	3.	Level sensitive
S.	TRAP	4.	Non-vectored

Codes:

	P	Q	R	S
(a)	1	3	4	2
(b)	2	4	3	1
(c)	1	4	3	2
(d)	2	3	4	1

Q.8 For fetch machine cycle the status signal S_1 and S_0 are respectively
(a) 0 and 0 (b) 0 and 1
(c) 1 and 0 (d) 1 and 1

Q.9 In INTEL 8085, while executing a program non maskable interrupt occurs. The data present on data line is
(a) 00 H (b) 24 H
(c) 36 H (d) can't be predicted

Q.10 Consider the table given below.

$IO/\overline{M}$	S_1	S_0	Machine cycle
0	1	1	X
1	0	1	Y
1	1	1	Z

Here S_0, S_1 are status signals.
X, Y, Z are respectively.
(a) Interrupt acknowledgment, I/O read, opcode fetch.
(b) Interrupt acknowledgment, I/O write, opcode fetch.
(c) Opcode fetch, I/O read, Interrupt acknowledgment.
(d) Opcode fetch, I/O write, Interrupt acknowledgment.

Q.11 A stack is
(a) an 8-bit register in the microprocessor
(b) an 16-bit register in the microprocessor
(c) a set of memory location in R/W memory reserved for storing information temporarily during the execution of a program.
(d) A 16-bit memory address stored in the program counter

Answer Key :

1. (a)	**2.** (c)	**3.** (a)	**4.** (c)	**5.** (c)
6. (c)	**7.** (d)	**8.** (d)	**9.** (b)	**10.** (d)
11. (c)				

CHAPTER 4

Peripheral Devices

4.1 Development of Data Transfer Schemes

The microprocessor can't just issue an address on the address bus following it up either by sending data or be ready to receive data from any of the peripherals at only instant of time as it may be already engaged. For this a signal is sent to the peripheral to know its status, common status signals are BUSY and READY.

A microprocessor based system or a computer may have several I/O devices of different speed. A slow I/O device can not transfer data when microprocessor issues instruction for the same because it takes some time to get ready. To solve this problem of speed mismatch between a microprocessor and I/O devices a number of data transfer techniques have been developed.

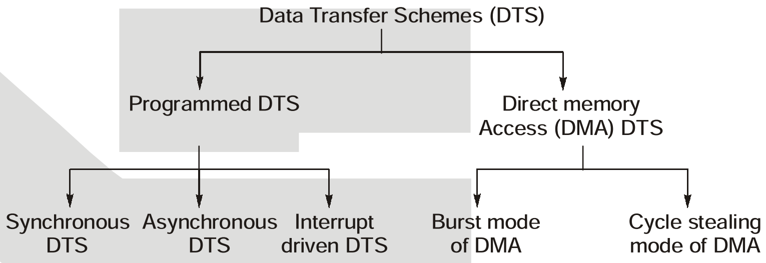
Data Transfer Schemes (DTS)
Programmed DTS
Direct memory Access (DMA) DTS
Synchronous DTS
Asynchronous DTS
Interrupt driven DTS
Burst mode of DMA
Cycle stealing mode of DMA

***Figure*-4.1**

Programmed DTS

- Programmed data transfer schemes are controlled by the CPU.
- Data is transferred from an I/O device to the CPU (or to the memory through the CPU) or vice versa under the control of programs which reside in the memory.
- Programmed DTS are employed when small amount of data are to be transferred.

Synchronous DTS

- Synchronous means "at the same time". In this DTS the device which sends data and the device which receives data are synchronised with the same clock.
- The data transfer with I/O devices is performed executing IN or OUT instructions for I/O mapped I/O devices or using memory read/write instructions for memory mapped I/O devices.
- The I/O devices compatible with microprocessors in speed are usually not available. Hence, this technique of data transfer is rarely used for I/O devices.

Asynchronous DTS

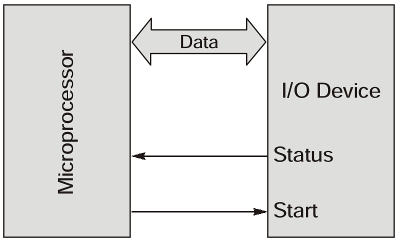

***Figure-* 4.2 :** *Asynchronous data transfer*

- Asynchronous means "at irregular intervals". In this DTS, data transfer is not based on predetermined timing pattern.
- This technique of data transfer is used when the speed of an I/O device does not match the speed of the microprocessor, and the timing characteristic of I/O device is not predictable.
- In this technique the status of the I/O device i.e. whether the device is ready or not, is checked by the microprocessor before the data are transferred.
- Asynchronous data transfer is used for slow I/O devices. This technique is an inefficient technique because the precious time of the microprocessor is wasted in waiting.

Two ways to achieve asynchronous DTS are:

(*i*) Strobe control: This method solves the problems by introducing a "2nd control signal" that provides a reply to the unit which indicates the transfer.

(*ii*) Hand shaking method of DTS: The "Handshaking Control" operates as follows:

1. The CPU interrogates the I/O module to check the status of the required device (peripheral).
2. The I/O module replies to the device status.
3. If free, the peripheral sends the data and a DATA READY signal to the I/O port.
4. The CPU determines that the DATA READY signal is active. A latch holds the DATA READY signal until the CPU reads it.
5. The CPU accepts the data and sends an Data Accepted Signal (DACC) by way of acknowledgment.
6. DATA READY line is reset to be ready to receive further data.

- For sending out data, the peripheral READY signal is sent to I/O module which is detected by the CPU.
- After sending an OUTPUT READY signal, data is fed on data bus to the peripheral at the completion of which acknowledgment is sent.

DMA Data Transfer Scheme (DMA-DTS)

- In DMA-DTS, MPU doesn't participate because data is directly transferred from an I/O device to the memory or vice-versa.
- In DMA, the MPU releases the control of the buses to a device called "DMA Controller". The controller manages data transfer between memory and a peripheral under its control, thus by- passing the MPU.
- The MPU communicates with the controller by using the chip select line, Buses and Control signals. Examples of DMA controller chips are: Intel 8237A, 8257 etc.
- DMA-DTS is a faster scheme as compared to programmed DTS.
- It is used to transfer data from "mass storage devices" (hard disks, floppy disks etc). It is also used for high-speed printers.
- Once the controller has gained the control, it plays the role of a processor for data transfer as:
  1. The DMA controller chip puts the MPU in a HOLD state by means of the HOLD control signal. This is an active high input signal.
  2. The MPU then stops executing the program and disconnects the address, data and memory control lines from its bus by placing them on a high impedance state. The microprocessor is totally disabled during DMA.

Example - 4.3 Assertion (A): DMA is faster than either Interrupt initiated I/O or Polling based I/O for very large data transfers.

Reason (R): DMA takes control of the system buses and needs no processor intervention during the data transfer.

(a) Both A and R are true and R is the correct explanation of A
(b) Both A and R are true but R is NOT the correct explanation of A
(c) A is true but R is false
(d) A is false but R is true

Answer : (a)

Example - 4.4 Match List-I with List-II and select the correct answer using the codes given below the lists:

	List-I		List-II
A.	Immediate addressing	1.	LDA 3000
B.	Implied addressing	2.	MOV A, B
C.	Register addressing	3.	LXI H, 2050
D.	Direct addressing	4.	RRC

Codes:

	A	B	C	D
(a)	3	4	2	1
(b)	2	1	3	4
(c)	3	1	2	4
(d)	2	4	3	1

Solution : (a)

A → 3 , B → 4, C → 2, D → 1.

4.2 Interfacing Devices

In order to communicate with the outside world microprocessor uses peripherals or I/O devices. The peripherals are connected to the microprocessor through interfacing circuits which convert the data available from an input device into compatible format for the microprocessor. The interface associated with the output device convert the output of the microprocessor into the desired peripheral format. To simplify the work interfacing devices are used.

Intel 8155 : Multipurpose Programmable Peripheral Interface

- Intel 8155 is multipurpose programmable device which is compatible to 8085 microprocessor.
- The ALE, $IO/\overline{M}$, $\overline{RD}$ and $\overline{WR}$ signals from 8085 can be directly connected to the device. This eliminate the need for external demultiplexing of lower order buses AD_0 - AD_7 and generation of control signals such as $\overline{MEMR}$, $\overline{MEMW}$, $\overline{IOR}$ and $\overline{IOW}$.
- The 8155 includes 256 bytes of RD/WR memory i.e. RAM, 3 I/O ports and a 16-bit timer.
- The timer of 8155 is two 8-bit register in which 14-bits are used for the counter and two bits are used for the timer mode.

CHAPTER 5

Introduction to Microprocessor 8086

5.1 Introduction

Intel's 16-bit processors are 8086, 80186 and 80286. These microprocessors has 16-bits internal architecture and contain 16 data lines. The 8088 and 80188 have their internal architecture of 16-bit but their data lines are only 8. All these microprocessor come under Intel's 8086 family. The Intel 8086, 80186, 80286, 8088 microprocessor have the same basic set of register, instruction and addressing modes.

The 8086 is a 16-bit, N-channel, HMOS microprocessor. The term HMOS is used for "high-performance MOS". Its clock frequencies for its different versions are 5 MHz, 8 MHz and 10 MHz. It contains an electronic circuitry of 29000 transistors. It is built on a single semiconductor chip with 40 pins packed in dual-in line packaging.

5.2 8086 Pin Diagram & Architecture

The pin diagram of 8086 is shown in Figure 5.1. The 8086 uses 20 address lines and 16 data lines. It can address 2^{20} = 1 Megabytes of memory. AD_0 - AD_{15} are 16 lower order address lines multiplexed with data lines and it operates in two modes.

From the above pin diagram, the pin description of 8086 is

GND : There are two pins connected to ground.

V_{CC} : This pin is connected to power supply +5 V DC.

RESET : This is active high input signal to reset the system.

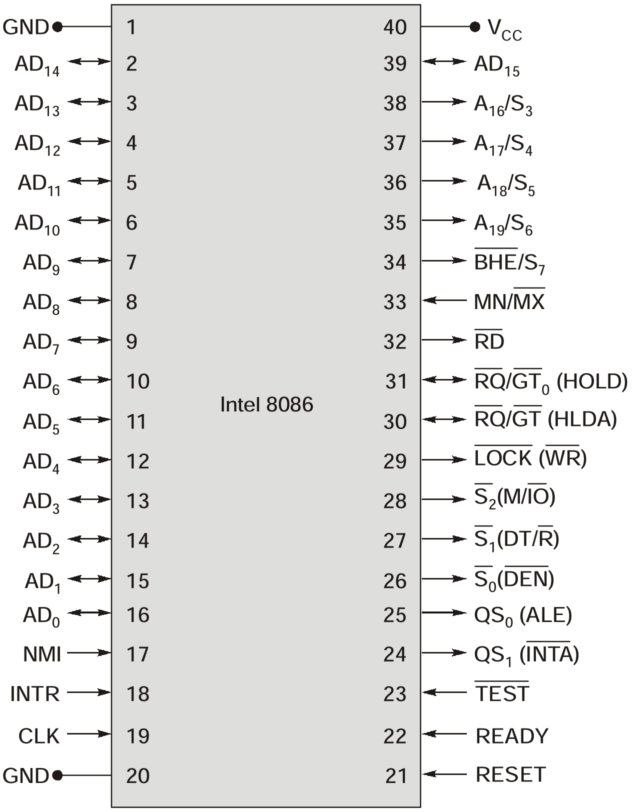

***Figure-5.1 :** Pin diagram of 8086 microprocessor*

READY : This is an active high input signal indicating that peripheral is ready to transfer data.

$\overline{RD}$: This is an active low output signal used for read operation.

AD_0 - AD_{15} : Bidirectional latched address data lines.

A_{16} - A_{19} : Higher order address bus.

$\overline{BHE}/S_7$: Bus high enable/status.

$MN/\overline{MX}$: This pin decide whether the processor is operating in minimum mode or maximum mode.

NMI: Non maskable interrupt request

$\overline{TEST}$: It is an active low signal. When it is low microprocessor continues execution and when it is high microprocessor goes to wait state.

S_1 - S_6 : These are the status signals multiplexed with higher order address lines.

Operating Modes of 8086

8086 can operate in two modes:

1. MN or Min mode: In a single processor system 8086 operates in min mode. In this, processing unit issues control signals required by memory and I/O devices.
2. MX or Max mode: In multiprocessor system, 8086 operates in max mode. In this, bus controller issues control signals.

Min Mode and Max Mode Signals

Pin No.	Min Mode	Max Mode
24.	$\overline{INTA}$ - Interrupt acknowledge	QS_1 - Queue status
25.	ALE - Address latch enable	QS_0 - Queue status
26.	$\overline{DEN}$ - Output data enable	$\overline{S}_0$ - Status signal
27.	$DT/\overline{R}$ - Data transmit/receive	$\overline{S}_1$ - Status signal
28.	$M/\overline{IO}$ - Memory or I/O access	S_2 - Status signal
29.	$\overline{WR}$ - Write signal	$\overline{LOCK}$ - LOCKS peripherals off the system
30.	HLDA - HOLD acknowledgment	$\overline{RQ}/\overline{GT}_1$ - Local bus priority control
31.	HOLD - HOLD request for system bus	$\overline{RQ}/\overline{GT}_0$ - Local bus priority control

Table-5.1

5.3 Internal Architecture of 8086

The internal architecture of 8086 is divided into two functional unit:

1. Bus interface unit [BIU]
2. Execution unit [EU]

Example - 5.7 If CS = 0000 H, DS = 1000 H, ES = SS = 2000 H, BP = BX = SI = DI = A000 H and SP = 0000 H, then explain and indicate the locations from which data required for the following instructions would be fetched:

MOVSB
JMP [BP + 10H]
IN AX, DX
IDIV [12134]
CALL FARPTR [2000 H]

Solution:

1. MOVSB

The following instruction will perform the following function

$$[ES : DI] \leftarrow [ES : SI]$$

i.e. the content at location IA 000 will be moved to location 2A000.

$$\begin{array}{lll} DS & : & 1000 \\ SI & : & \mathrm{A000} \\ \hline & & \mathrm{1A000H} \end{array} \qquad \begin{array}{lll} ES & : & 2000\ \mathrm{H} \\ DI & : & \mathrm{A000} \\ \hline & & \mathrm{2A000H} \end{array}$$

Since it is a 20 bit address instruction thus the address will be generated as shown above.

2. JMP [BP + 10H]

JMP is related to transfer of program control. This instruction will cause program to jump from current location to the location given by adding the data stored in the *BP* plus the offset.

$$\text{i.e} \qquad \begin{array}{lr} \mathrm{BP} : & \mathrm{A000H} \\ + & \mathrm{0010H} \\ \hline & \mathrm{A010H} \\ \hline \end{array}$$

JMP is related to program control.

$$\text{i.e} \qquad \begin{array}{lr} \mathrm{CS} : & \mathrm{0000H} \\ + & \mathrm{A010H} \\ \hline & \mathrm{A010H} \\ \hline \end{array}$$

3. IN AX, DX

When this instruction is executed the input word will be from 16 bit part address of DX to AX.

4. IDIV [12134]

This is used for the purpose of divide operation. In the dividend which is available at AX register will be divided by the number stored at the location of 1214H.

AL contains the result after division of AX by [12134]

i.e. AL contains the signed quotient

AH contains the remainder (signed).

5. CALL FARPTR [2000 H]

Step 1 : This instruction decrements the stock pointer by 2 and copies the content of cs register to the stock. i.e. 2 memory locations.

Step 2 : It again decrements the stack pointer by 2 and copies the offset of the instruction after the CAL instruction to the stack. i.e. value of instruction pointer.

Step 3 : It loads CS with the segment base of the segment that contains the procedure.

i.e. CS stores with 2000 H.

Step 4 : It also stores the IP with offset of the first instruction in the procedure.

Student's Assignment

Q.1 The interrupt vector table IVT of 8086 contains
(a) the contents of CS and IP of the main program address to which the interrupt has occurred.
(b) the contents of CS and IP of the main program address to which the control has to come back after the service routine.
(c) the starting CS and IP values of the interrupt service routine.
(d) the starting address of the IVT.

Q.2 The 8086 arithmetic instructions work on
1. signed and unsigned numbers
2. ASCII data
3. unpacked BCD data.

Select the correct answer using the codes given below:

(a) 1 and 2 (b) 2 and 3
(c) 1 and 3 (d) 1, 2 and 3

Answer Key :

1. (c) **2.** (c)