Victoria University
of Bangladesh

## Assessment Topic:

**Final Assessment**

## **Course Title:** Microprocessor

**Course Code:** CSE-413

## Submitted To:

## Umme Khadiza Tithi

**Lecturer, Department of Computer Science & Engineering**

Victoria University of Bangladesh

## Submitted By:


## Ruhul Amin

**ID:** 2120180051

**Department:** CSE

**Semester:** Fall-2022

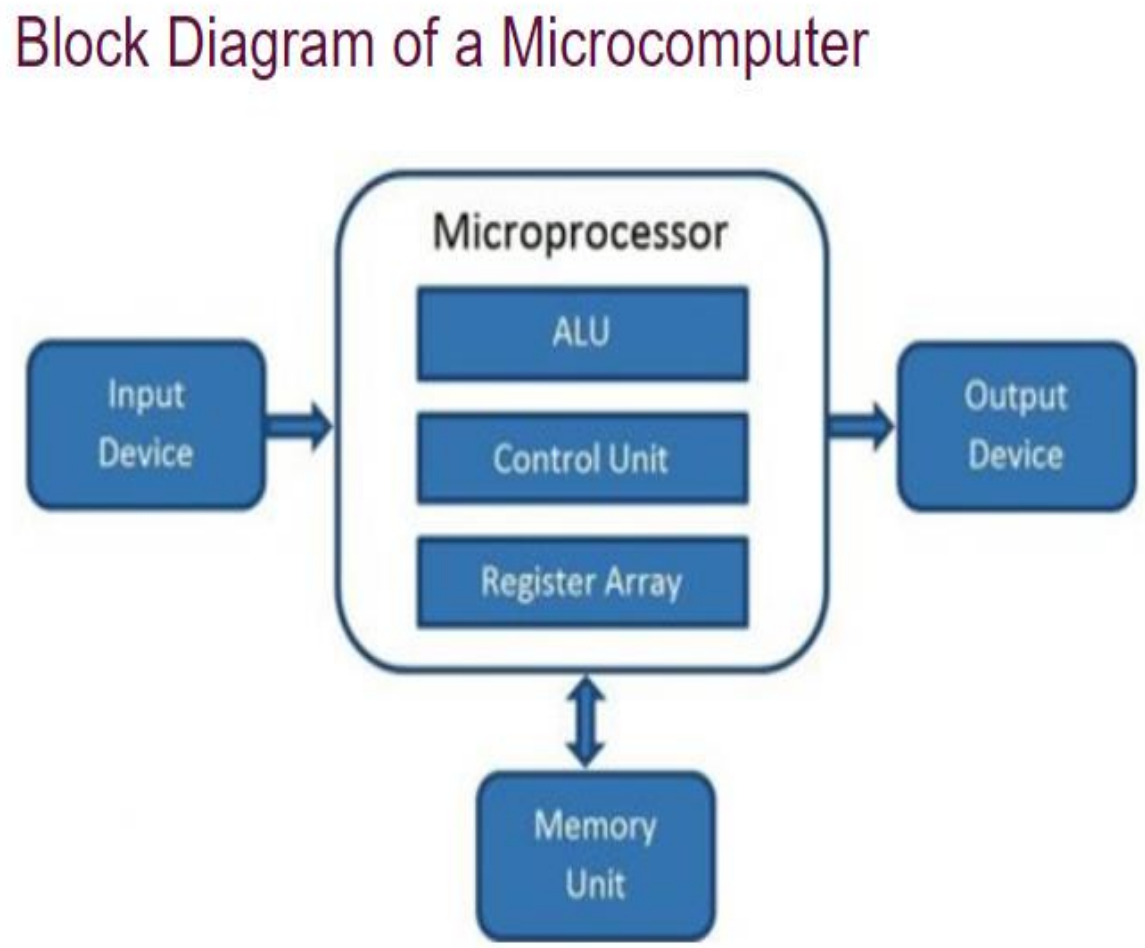
**Batch:** 18th


Submission Date: 05th February 2023

## Answer to question no 1(a)

**# Write down a block diagram of a basic Microcomputer and the features of a Microprocessor.**

**Answer:** A digital computer with one microprocessor which acts as a CPU is called a microcomputer. It is a programmable, multipurpose, clock-driven, register-based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions, and provides results as output.
The microprocessor contains millions of tiny components like transistors, registers, and diodes that work together.



**Features of Microprocessor**

- **Low Cost** - Due to integrated circuit technology microprocessors are available at a very low cost. It will reduce the cost of a computer system.

- **High Speed** - Due to the technology involved in it, the microprocessor can work at very high speed. It can execute millions of instructions per second.

- **Small Size** - A microprocessor is fabricated in a very less footprint due to very large scale and ultra-large-scale integration technology. Because of this, the size of the computer system is reduced.

- **Versatile** - The same chip can be used for several applications; therefore, microprocessors are versatile.

- **Low Power Consumption** - Microprocessors are using metal oxide semiconductor technology, which consumes less power.

- **Less Heat Generation** - Microprocessors use semiconductor technology which will not emit much heat as compared to vacuum tube devices.

- **Reliable** - Since microprocessors use semiconductor technology, therefore, the failure rate is very less. Hence it is very reliable.

- **Portable** - Due to their small size and low power consumption microprocessors are portable.

## Answer to question no 1(b)

**Answer:** Intel 8085 is fabricated as a 40-pin DIP IC. DIP stands for 'dual inline package'. It means the package will have pins on only two sides, 20 on each side in this case.

Intel manufactures 8085 in several versions, like 8085A, 8085AH, 8085AH-2, and 8085AH-1. The 8085A is fabricated using NMOS technology. It is a variant of MOS (metal oxide semiconductor) technology. It uses n channel silicon-gate process. The AH series are more expensive processors, which use high-density MOS (HMOS) for fabrication. They typically consume 20% less power compared to the A series.

# Pin Configuration

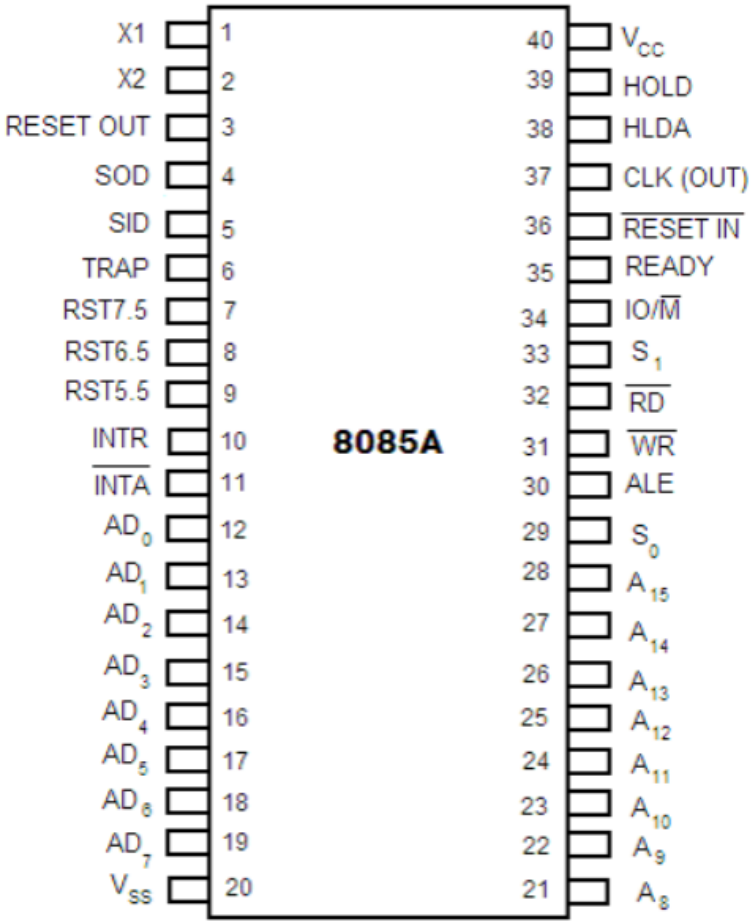



**Fig: Pin diagram of Intel 8085 microprocessor**

## Answer to question no 1(c)

**# Define the classification of microprocessors.**

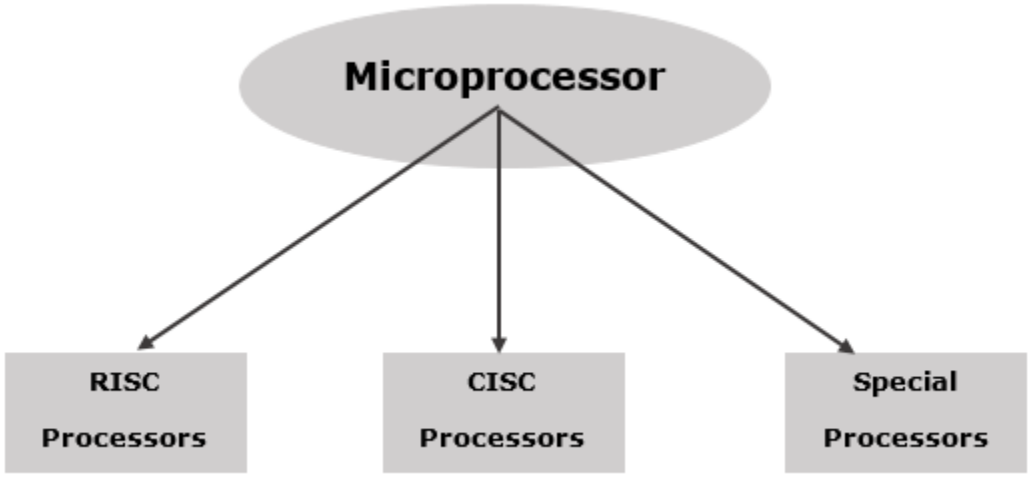
**Answer:**

Microprocessors can classify based on their application & architecture of microprocessors. Based on the application of the processors, they are classified as-

- General Purpose Processors
- Microcontrollers
- Special Purpose Processors

Based on the architecture of the processor, they are classified as-

- RISC (Reduced Instruction Set Computer)
- CISC (Complex Instruction Set Microprocessors)




**General Purpose Processor**

- The use of general-purpose processors by the programmer for any application is used in general computer system integration.
- Common microprocessors such as Intel 8085 to Intel Pentium processors are an example of general-purpose processors.

**Microcontroller**

- Microcontrollers are microprocessors designed especially for control applications.
- Microcontrollers contain memory units and I/O ports inside a chip in addition to the CPU part.
- Some of the machines using microcontrollers are microwave ovens, washing machines, computer printers, fax machines, etc.

**Special Purpose Processor**

- Special-purpose processors are designed specifically to handle special functions required for an application.
- Digital signal processors are examples of special-purpose processors and these have special instructions to handle signal processing.
- The application-specific integrated circuit chips are also examples of this category of microprocessors.

**RISC (Reduced Instruction Set Computer)**

- RISC stands for Reduced instruction set computer. RISC chips evolved around mid -1980 as a reaction to CISC chips.
- It is designed to reduce the execution time by simplifying the instruction set of the computer.
- Using RISC processors, each instruction requires only one clock cycle to execute results in uniform execution time.
- This reduces the efficiency as there are more lines of code, hence more RAM is needed to store the instructions.
- The compiler also has to work more to convert high-level language instructions into machine code.
- Because of the Simpler and faster instructions, RISC chips have more simple instructions which require fewer transistors, which makes them easier to design and cheaper to produce.

**CISC (Complex Instruction Set Microprocessors)**

- CISC stands for Complex Instruction Set Computer.
- It is designed to minimize the number of instructions per program, ignoring the number of cycles per instruction.
- The focus is on building complex instructions directly into the hardware.
- The compiler has to do very little work to translate a high-level language into assembly-level language/machine code because the length of the code is relatively short, so very little RAM is required to store the instructions
- Most PC use CPU based on architecture. For instance, intel and AMD CPUs are based on CISC architectures.

## Answer to question no 2(a)

**# How DMA operations are performed?**

**Answer:** DMA stands for Direct Memory Access. It is designed by Intel to transfer data at the fastest rate. It allows the device to transfer the data directly to/from memory without any interference from the CPU.

Using a DMA controller, the device requests the CPU to hold its data, address, and control bus, so the device is free to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving the HLDA signal from the CPU.

**Following is the sequence of operations performed by a DMA −**

- Initially, when any device has to send data between the device and the memory, the device has to send a DMA request (DRQ) to the DMA controller.
- The DMA controller sends a Hold request (HRQ) to the CPU and waits for the CPU to assert the HLDA.
- Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves control over the bus and acknowledges the HOLD request through the HLDA signal.
- Now the CPU is in a HOLD state and the DMA controller has to manage the operations over buses between the CPU, memory, and I/O devices.

## Answer to question no 2(b)

**# Define 8257 pin description.**

**Answer:**

**Address Bus (A0-A3 and A4-A7):** The four least significant lines A0-A3 are bi–directional tri–state signals. In the idle cycle, they are inputs and used by the CPU to address the register to be loaded or read. In the Active cycle, they output the lower 4 bits of the address for DMA operation. A4-A7 are unidirectional lines, that provide 4 bits of addresses during DMA service.

**Address Strobe (ADSTB):** This signal is used to demultiplex higher byte address and data using the external latch.

**Address Enable (AEN):** This active high signal enables the 8-bit latch containing the upper 8-address bits onto the system address bus. AEN can also be used to disable other system bus drivers during DMA transfers.

**Memory Read and Memory Write (MEMR, MEMW):** These are active low tri-state signals. The MEMR signal is used to access data from the addressed memory location during a DMA read or memory-to-memory transfer and the MEMW signal is used to write data to the addressed memory location during a DMA write or memory-to-memory transfer.

**I/O Read and I/O Write (IOR and IOW):** These are active low bi-directional signals. In the idle cycle, these are input control signals used by the CPU to read/write the control registers. In the active cycle IOR signal is used to access data from a peripheral and IOW signal is used to send data to the peripheral.

**Chip Select (CS):** This is an active low input, used to select the 8257 as an I/O device during the idle cycle. This allows the PU to communicate with the 8257 Pin Diagram.

**Reset** This active high signal clears, the command, status, request, and temporary registers. It also clears the first/last flip-flop and sets Master Register. After resetting, the device is in the idle cycle.

**Ready**: This input is used to extend the memory read and write signals from the 8257 to interface slow memories or I/O devices.

Hold request (HRQ): Any valid DREQ causes 8257 to issue the HRQ. It is used for requesting the CPU to get the control of system bus.

**Hold Acknowledge (HLDA):** The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system bus.

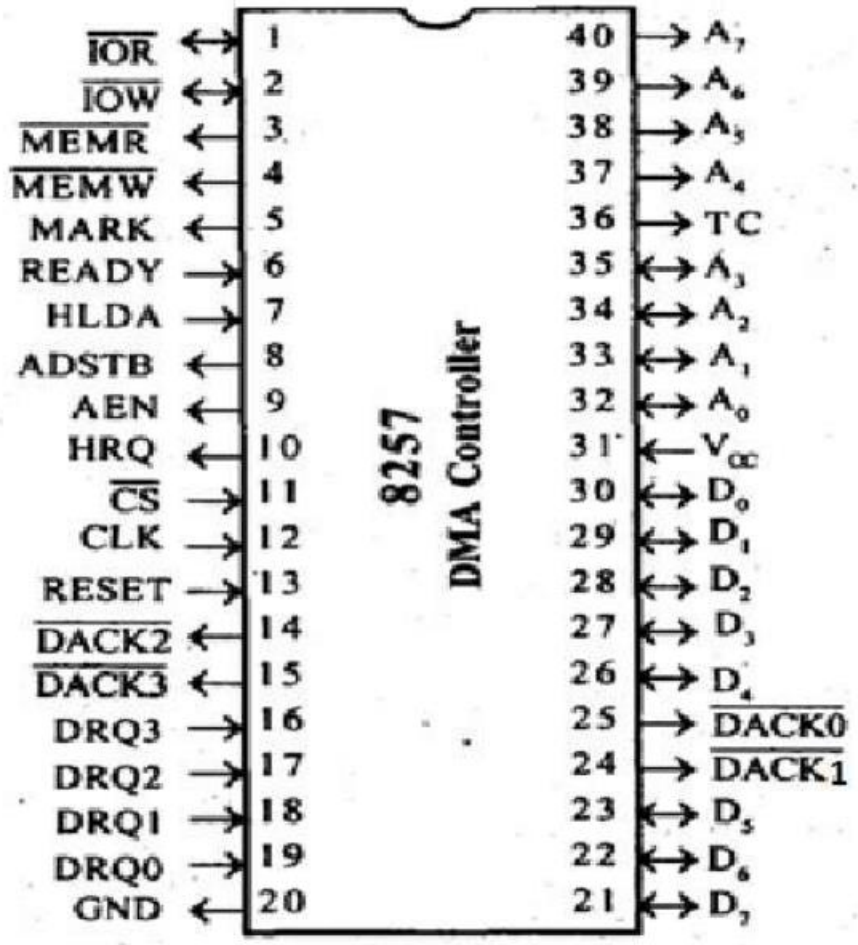
**DREQ0-DREQ3:** These are DMA request lines, which are activated to obtain DMA service, until the corresponding DACK signal goes active.

DACK0-DACK3: These are used to indicate peripheral devices that the DMA request is granted.

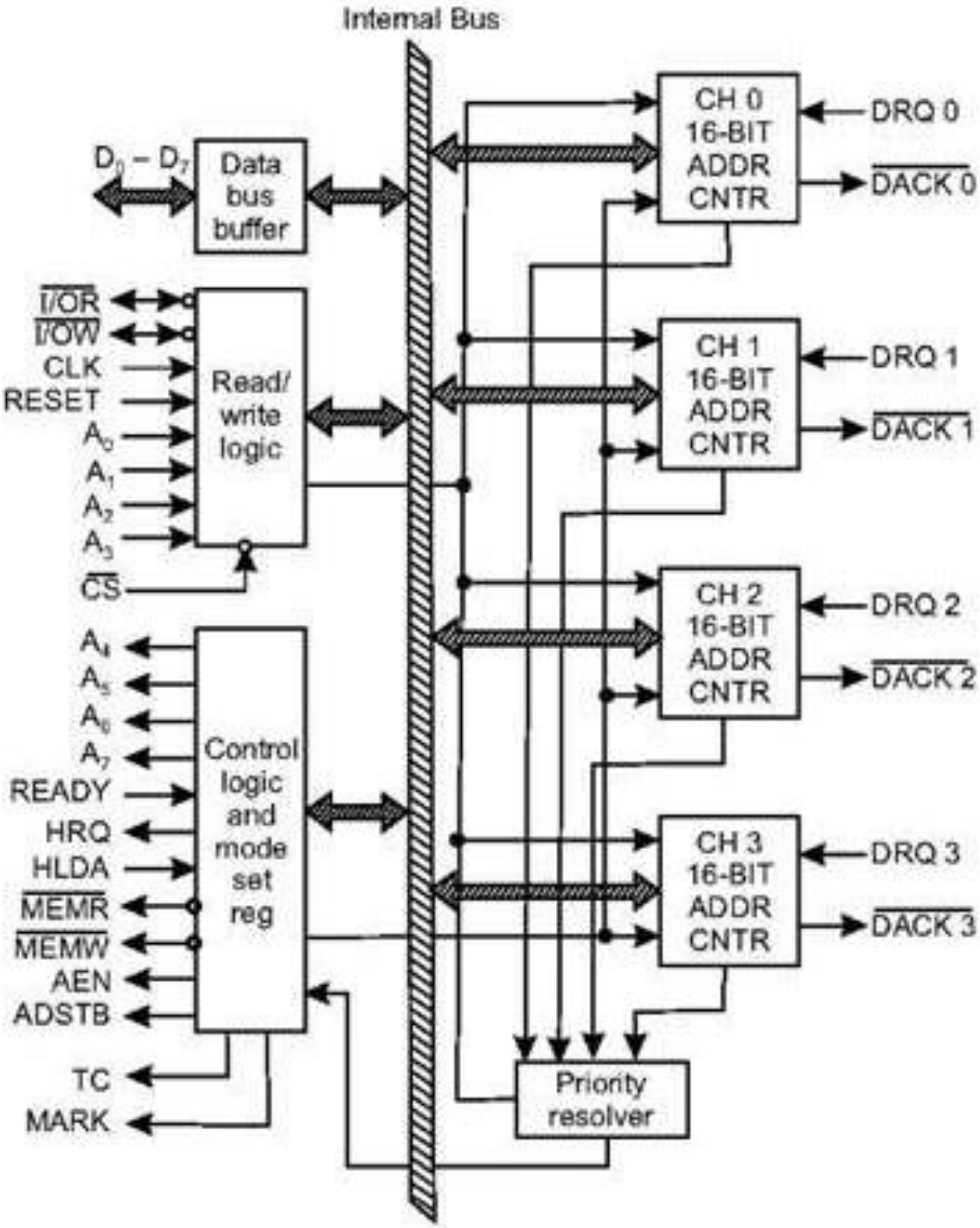
**Terminal Count (TC):** This is an active high signal concern with the completion of the DMA service. The TC output signal is activated at the end of the DMA service, i.e., when the present cycle is the last cycle for the current data block.

**MARK**: This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (all multiplies of 128) cycles from the end of the data block.

The following image shows the pin diagram of a 8257 DMA controller −



The following image shows the architecture of 8257 –



## Answer to question no 3(a)

**# Describe Rotate, Shift, and Branch Instruction set of 8086.**

**Answer:**

As we know that any machine (system) works on machine language, which consists of binary numbers. In the 8086 microprocessors, we have 16-bit registers to handle our data.

### Rotate Instructions

The following instructions come under this category:

| Instruction | Description |
|---|---|
| **RCL** | Rotate all bits of the operand left by specified number of bits through carry flag. |
| **RCR** | Rotate all bits of the operand right by specified number of bits through carry flag. |
| **ROL** | Rotate all bits of the operand left by specified number of bits. |
| **ROR** | Rotate all bits of the operand right by specified number of bits. |

### Shift Instructions

The following instructions come under this category:

| Instruction | Description |
|---|---|
| **SAL or SHL** | Shifts each bit of operand left by specified number of bits and put zero in LSB position. |
| **SAR** | Shift each bit of any operand right by specified number of bits. Copy old MSB into new MSB. |
| **SHR** | Shift each bit of operand right by specified number of bits and put zero in MSB position. |

### Branch Instructions

It is also called program execution transfer instruction. Instructions of this group transfer program execution from the normal sequence of instructions to the specified destination or target. The following instructions come under this category:

| Instruction | Description |
|---|---|
| **JA or JNBE** | Jump if above, not below, or equal i.e. when CF and ZF = 0 |
| **JAE/JNB/JNC** | Jump if above, not below, equal or no carry i.e. when CF = 0 |
| **JB/JNAE/JC** | Jump if below, not above, equal or carry i.e. when CF = 0 |
| **JBE/JNA** | Jump if below, not above, or equal i.e. when CF and ZF = 1 |
| **JCXZ** | Jump if CX register = 0 |
| **JE/JZ** | Jump if zero or equal i.e. when ZF = 1 |
| **JG/JNLE** | Jump if greater, not less or equal i.e. when ZF = 0 and CF = OF |
| **JGE/JNL** | Jump if greater, not less or equal i.e. when SF = OF |
| **JL/JNGE** | Jump if less, not greater than or equal i.e. when SF ≠ OF |
| **JLE/JNG** | Jump if less, equal or not greater i.e. when ZF = 1 and SF ≠ OF |
| **JMP** | Causes the program execution to jump unconditionally to the memory address or label given in the instruction. |
| **CALL** | Calls a procedure whose address is given in the instruction and saves their return address to the stack. |
| **RET** | Returns program execution from a procedure (subroutine) to the next instruction or main program. |
| **IRET** | Returns program execution from an interrupt service procedure (subroutine) to the main program. |
| **INT** | Used to generate software interrupts at the desired point in a program. |
| **INTO** | Software interrupts to indicate overflow after an arithmetic operation. |
| **LOOP** | Jump to the defined label until CX = 0. |
| **LOOPZ/LOOPE** | Decrement CX register and jump if CX ≠ 0 and ZF = 1. |
| **LOOPNZ/LOOPNE** | Decrement CX register and jump if CX ≠ 0 and ZF = 0. |

Here, CF = Carry Flag

ZF = Zero Flag

OF = Overflow Flag

SF = Sign Flag

CX = Register

## Answer to question no 3(b)

**# What is Memory and I/O interfacing?**

**Answer:**

**Memory** refers to the storage units where data and instructions are temporarily stored for processing by the microprocessor. There are two main types of memory in microprocessors:

1. Random Access Memory (RAM): RAM is volatile memory that can be read from and written to by the microprocessor. It is used for temporary storage of data and instructions, and the contents of RAM are lost when the power is turned off.
2. Read-Only Memory (ROM): ROM is non-volatile memory that can only be read by the microprocessor. It is used to store permanent data and instructions, such as the program that starts up when the computer is turned on.

There are various types of RAM and ROM, including Dynamic RAM (DRAM), Static RAM (SRAM), Programmable ROM (PROM), and Erasable Programmable ROM (EPROM). Each type of memory has different characteristics, such as cost, speed, and capacity, which make them suitable for different applications.

**I/O (Input/Output) interfacing** refers to the communication between the microprocessor and the external devices such as keyboards, displays, storage devices, etc. The microprocessor communicates with these devices by sending and receiving data and control signals through an I/O interface.

There are two main methods of I/O interfacing:

**Memory-Mapped I/O:** In this method, I/O devices are assigned memory addresses, just like memory. The microprocessor reads and writes data to these memory addresses, which are then passed on to the corresponding I/O device. This method is simple and efficient, but it uses valuable memory space.

**Port-Mapped I/O:** In this method, I/O devices are assigned port addresses, which are separate from the memory addresses. The microprocessor reads and writes data to these port addresses, which are then passed on to the corresponding I/O device. This method uses less memory space, but it is more complex and slower than memory-mapped I/O.

The type of I/O interface used depends on the specific requirements of the system and the devices being interfaced. In general, memory-mapped I/O is used for high-speed devices, while port-mapped I/O is used for low-speed devices.