Name	**Miandad Khan**
ID	**14130**
Subject	**MAL**

Q.1 Give answers to each of the following: (1.5 x 5 = 7.5)

a) Discuss the Virtual machine concept using examples

Ans:-

A virtual machine (or "VM") is an emulated computer system created using software. It uses physical system resources, such as the CPU, RAM, and disk storage, but is isolated from other software on the computer. It can easily be created, modified, or destroyed without affecting the host computer.

Virtual machines provide similar functionality to physical machines, but they do not run directly on the hardware. Instead, a software layer exists between the hardware and the virtual machine.

b) Explain different registers used in x86 32-Bit processors.

Ans:-

The x86 architecture has 8 General-Purpose Registers (GPR), 6 Segment Registers, 1 Flags Register and an Instruction Pointer. 64-bit x86 has additional registers.
Accumulator register (AX). Used in arithmetic operations
Counter register (CX). Used in shift/rotate instructions and loops.
Data register (DX). Used in arithmetic operations and I/O operations.
Base register (BX). Used as a pointer to data (located in segment register DS, when in segmented mode).
Stack Pointer register (SP). Pointer to the top of the stack.
Stack Base Pointer register (BP). Used to point to the base of the stack.
Source Index register (SI). Used as a pointer to a source in stream operations.
Destination Index register (DI). Used as a pointer to a destination in stream operations.

c) Discuss different features of Intel P965 Express Chipset.

Ans:-

- Support Intel® next generation 45nm Multi-core CPU
- Intel LGA775 Platform
- Intel® P965 chipset
- Intel® Core™2 Quad / Core™2 Extreme / Core™2 Duo / Pentium® Extreme / Pentium® D / Pentium® 4 Processors
- Dual-channel DDR2 800/667/533 MHz
- 5 SATA on board/ 1 eSATA
- 10 USB
- 6-channel HD Audio
- All High-quality Conductive Polymer Capacitors

d) Elaborate different I/O levels involved in displaying a string of characters.

Ans:-

Management of I/O devices is a very important part of the operating system - so important and so varied that entire I/O subsystems are devoted to its operation. (Consider the range of devices on a modern computer, from mice, keyboards, disk drives, display adapters, USB devices, network connections, audio I/O, printers, special devices for the handicapped, and many special-purpose peripherals.

I/O devices can be roughly categorized as storage, communications, user-interface, and other

Devices communicate with the computer via signals sent over wires or through the air.

Devices connect with the computer via ports, e.g. a serial or parallel port.

A common set of wires connecting multiple devices is termed a bus.

Buses include rigid protocols for the types of messages that can be sent across the bus and the procedures for resolving contention issues.

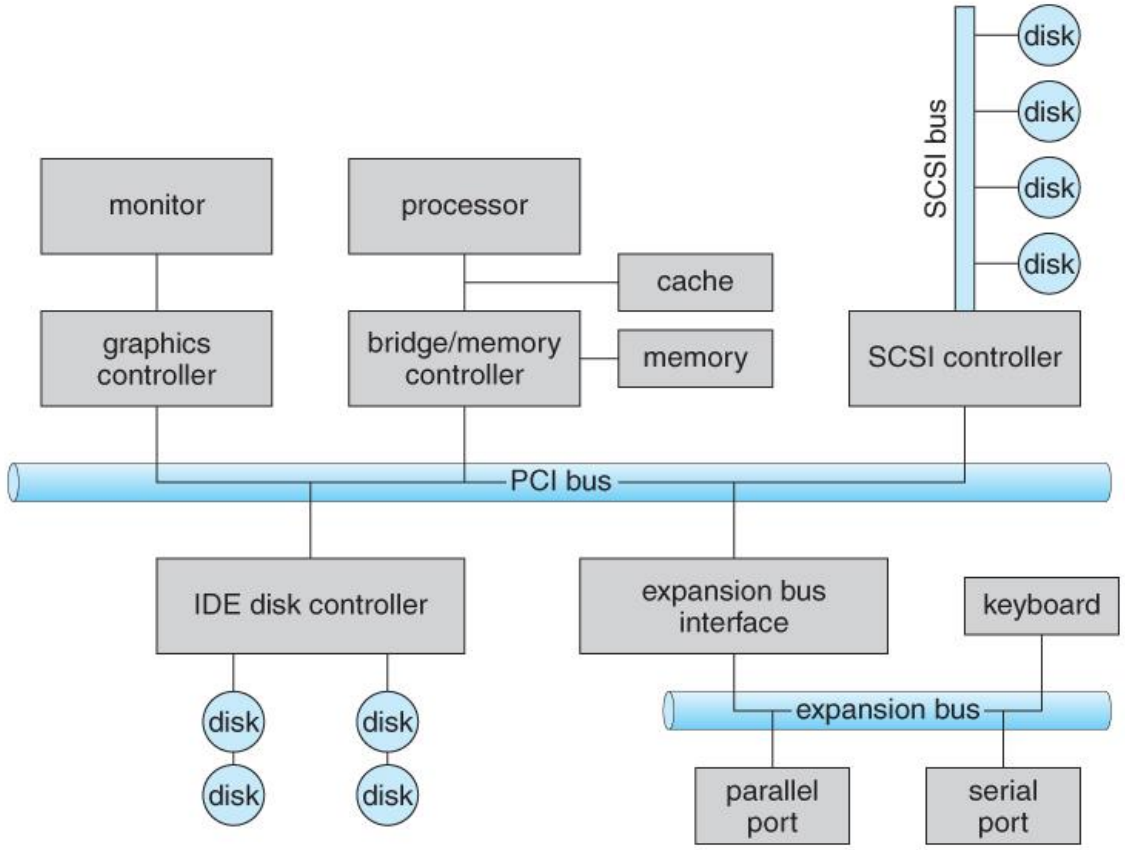
Figure 13.1 below illustrates three of the four bus types commonly found in a modern PC:

The PCI bus connects high-speed high-bandwidth devices to the memory subsystem (and the CPU.)

The expansion bus connects slower low-bandwidth devices, which typically deliver data one character at a time (with buffering.)

The SCSI bus connects a number of SCSI devices to a common SCSI controller.

A daisy-chain bus, (not shown) is when a string of devices is connected to each other like beads on a chain, and only one of the devices is directly connected to the host.



e) Explain different instruction mnemonics having zero, one, and two operands.

Ans:-

Instruction Formats (Zero, One, Two and Three Address Instruction
Computer perform task on the basis of instruction provided. An instruction in computer comprises of groups called fields. These field contains different information as for computers every thing is in 0 and 1 so each field has different significance on the basis of which a CPU decide what to perform. The most common fields are:
An instruction is of various length depending upon the number of addresses it contain. Generally CPU organization are of three types on the basis of number of address fields:

Single Accumulator organization
General register organization
Stack organization
In first organization operation is done involving a special register called accumulator. In second on multiple registers are used for the computation purpose. In third organization the work on stack basis operation due to which it does not contain any address field. It is not necessary that only a single organization is applied a blend of various organization is mostly what we see generally.

On the basis of number of address, instruction are classified as:

Note that we will use X = (A+B)*(C+D) expression to showcase the procedure.

Zero Address Instructions –

A stack based computer do not use address field in instruction.To evaluate a expression first it is converted to revere Polish Notation i.e. Post fix Notation.

Expression: X = (A+B)*(C+D)
Postfixed : X = AB+CD+*
TOP means top of stack
M[X] is any memory location

PUSH	A	TOP = A
PUSH	B	TOP = B
ADD		TOP = A+B
PUSH	C	TOP = C
PUSH	D	TOP = D
ADD		TOP = C+D
MUL		TOP = (C+D)*(A+B)
POP	X	M[X] = TOP

One Address Instructions –

This use a implied ACCUMULATOR register for data manipulation.One operand is in accumulator and other is in register or memory location.Implied means that the CPU already know that one operand is in accumulator so there is no need to specify it.

Expression: X = (A+B)*(C+D)
AC is accumulator
M[] is any memory location
M[T] is temporary location

```
LOAD   A      AC = M[A]
ADD    B      AC = AC + M[B]
STORE  T      M[T] = AC
LOAD   C      AC = M[C]
ADD    D      AC = AC + M[D]
MUL    T      AC = AC * M[T]
STORE  X      M[X] = AC
```

Two Address Instructions –

This is common in commercial computers.Here two address can be specified in the instruction.Unlike earlier in one address instruction the result was stored in accumulator here result cab be stored at different location rather than just accumulator, but require more number of bit to represent address.

Here destination address can also contain operand.

Expression: X = (A+B)*(C+D)

R1, R2 are registers
M[] is any memory location

```
MOV    R1, A    R1 = M[A]
ADD    R1, B    R1 = R1 + M[B]
MOV    R2, C    R2 = C
ADD    R2, D    R2 = R2 + D
MUL    R1, R2   R1 = R1 * R2
MOV    X, R1    M[X] = R1
```

Q.2 Differentiate each of the following: (1.5 x 5 = 7.5)

a) Real address mode and protected mode

Ans:-

Protected Mode

Protected mode is the native state of the processor, in which all instructions and features are available. Programs are given separate memory areas named segments, and the processor prevents programs from referencing memory outside their assigned segments

Real-Address Mode

Real-address mode implements the programming environment of an early Intel processor with a few extra features, such as the ability to switch into other modes. This mode is useful if a program requires direct access to system memory and hardware devices.

b) Instruction and directive?

Ans:-

Directives and instructions are two terms that the often confused. A person may be needed to follow instructions or a directive. So, what exactly is the difference between the two and which should be used when?

Directive:

Serving to direct; directing: a directive board.
Psychology. Pertaining to a type of psychotherapy in which the therapist actively offers advice and information rather than dealing only with information supplied by the patient.
An authoritative instruction or direction; specific order: a new directive by the President on foreign aid.

Instructions:-

The act or practice of instructing or teaching; education.
Knowledge or information imparted.
An item of such knowledge or information.
Usually, instructions. orders or directions: The instructions are on the back of the box.
The act of furnishing with authoritative directions.
Computers. A command given to a computer to carry out a particular operation.
The main difference between directions or instruction is the fact that a directive is mainly an order, usually issued by an authority. A directive may establish policy, assign responsibilities, define objectives and delegate authority to those working in and with the authoritative figure.

c) Equal-sign directive and EQU directive

Ans:-

Equal-Sign Directive
The equal-sign directive associates a symbol name with an integer expression. The
syntax is
name = expression
5
- Expression is a 32-bit integer (expression or constant)
- may be redefined
- Name is called a symbolic constant

EQU Directive
🕐 Define a symbol as either an integer or text expression
🕐 Cannot be redefined
🕐 There are three formats:

o name EQU expression SIZE EQU 10*10
o name EQU <text> pressKey EQU <"Press any key to continue...",0>
o name EQU symbol : symbol is an existing symbol name, already defined with
= or EQU
d) Data label and code label?
Ans:-
Status Flags
The status flags reflect the outcomes of arithmetic and logical operations performed by the CPU. They are the Overflow, Sign, Zero, Auxiliary Carry, Parity, and Carry flags. Their abbreviations are shown immediately after their names:

• The Carry flag (CF) is set when the result of an unsigned arithmetic operation is too large to fit into the destination.

• The Overflow flag (OF) is set when the result of a signed arithmetic operation is too large or too small to fit into the destination.

• The Sign flag (SF) is set when the result of an arithmetic or logical operation generates a negative result.

• The Zero flag (ZF) is set when the result of an arithmetic or logical operation generates a result of zero.

A flag is set when it equals 1; it is clear (or reset) when it equals 0.

• TheAuxiliary Carry flag (AC) is set when an arithmetic operation causes a carry from bit 3 to bit 4 in an 8-bit operand.

• The Parity flag (PF) is set if the least-significant byte in the result contains an even number of 1 bits. Otherwise, PF is clear. In general, it is used for error checking when there is a possibility that data might be altered or corrupted.

Q.3 Solve each of the following: (1.5 x 4 = 6)

a) If W = 11101100, X = 00010011, and Y = 00111100, then find Z = W ∨ X ∧ ¬Y?

W	x	Y	¬y	w ∨ x	w ∨ x ∧ ¬y
1	0	0	1	1	1
1	0	0	1	1	1
1	0	1	0	1	0
0	1	1	0	1	0
1	0	1	0	1	0
1	0	1	0	1	0
0	1	0	1	1	1
0	1	0	1	1	1

b) Create a truth table for the Boolean function described by ¬A ∧ ¬B

A	¬A	B	¬B	¬A ∧ ¬B
0	1	0	1	1
0	1	0	1	1
1	0	1	0	0

1	0	1	0	0

d) Write the real number -3.7×107 as a real number literal using MASM syntax

Q.4 Attempt each of the following: $(1.5 + 1.5 + 2 + 4 = 9)$?

a) Show the order of individual bytes in memory for the following doubleword variable using little endian order: dVal DWORD 87654321h

Ans:-

b) Write a program that calculates the following expression, using registers:
$D = (C + B) - A$ Assign integer values to the EAX, EBX, and ECX registers

Programm solution:-

```
.386
.model flat,stdcall
.code
Main PROC
        mov ecx,3h
        mov ebx,8h
        move ax,5h

        add ecx,ebx
        sub eax,ecx
INVOKE ExitProcess,0
Main ENDP
END main
```

d) Write a program that performs arithmetic operations on different 32-bit memory operands and stores the result in memory. Give stepwise explanation of each statement.

ANS:-

The INC Instruction

The INC instruction is used for incrementing an operand by one. It works on a single operand that can be either in a register or in memory.

Syntax
The INC instruction has the following syntax —

INC destination
The operand destination could be an 8-bit, 16-bit or 32-bit operand.

Example

```
INC EBX          ; Increments 32-bit register
INC DL           ; Increments 8-bit register
INC [count]    ; Increments the count variable
```

The DEC Instruction

The DEC instruction is used for decrementing an operand by one. It works on a single operand that can be either in a register or in memory.

Syntax

The DEC instruction has the following syntax –

DEC destination

The operand destination could be an 8-bit, 16-bit or 32-bit operand.

Example

```
segment .data
    count dw    0
    value db    15

segment .text
    inc [count]
    dec [value]

    mov ebx, count
    inc word [ebx]

    mov esi, value
    dec byte [esi]
```

The ADD and SUB Instructions

The ADD and SUB instructions are used for performing simple addition/subtraction of binary data in byte, word and doubleword size, i.e., for adding or subtracting 8-bit, 16-bit or 32-bit operands, respectively.

Syntax

The ADD and SUB instructions have the following syntax –

```
ADD/SUB         destination, source
```

e)