Session: 2018-2019

1.

a)

I. **How can you determine the microprocessor as 8-bit or 16-bit or 32-bit or 64-bit?**

To determine the microprocessor's bit architecture, check its data bus width. 8-bit has an 8-bit data bus, 16-bit has 16 bits data bus, 32-bit has 32 bits data bus, and 64-bit bus has 64 bits data bus.

II. **Distinguish between 8085 and 8086.**

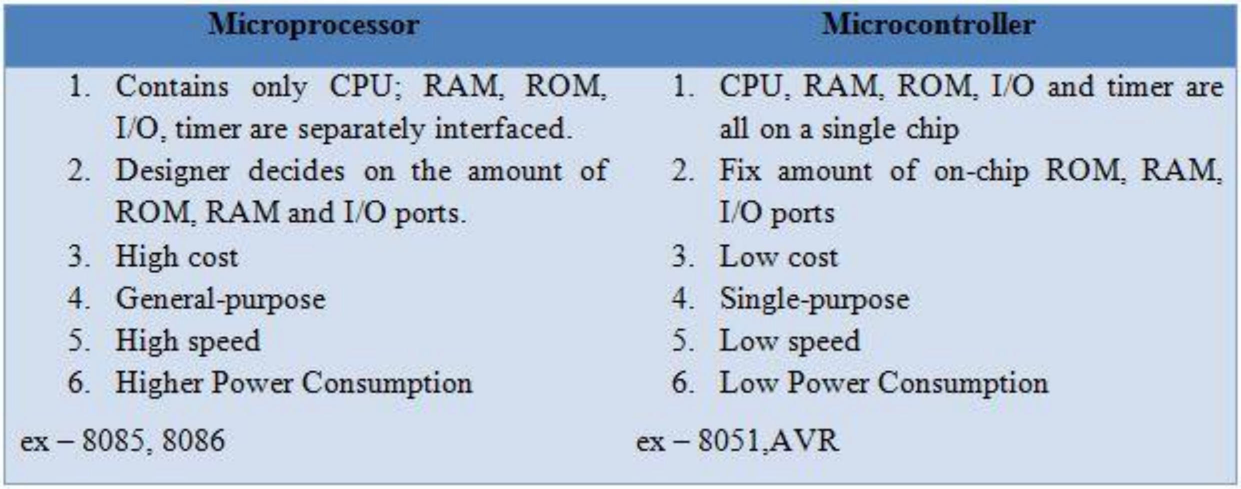
Feature	**Intel 8085**	**Intel 8086**
Architecture	8-bit	16-bit
Data Bus	8-bit	16-bit
Address Bus	16-bit	20-bit (later versions supported 16-bit mode)
opcodes	74 instructions, 74 opcodes	246 instructions, 191 opcodes
Instruction Set	Limited set of instructions	Rich set of instructions for complex tasks
Registers	Accumulator, BC, DE, HL, SP, PC	General Purpose Registers (AX, BX, CX, DX, SI, DI, BP, SP), IP, Flags

b)

i. Mention the role of segment register during protected mode operation.

In protected mode, segment registers define memory segments. They play a crucial role in providing access control and memory protection by specifying the base address of the segments.

ii. Distinguish microcontroller and microprocessor.



Microprocessor	Microcontroller
1. Contains only CPU; RAM, ROM, I/O, timer are separately interfaced.	1. CPU, RAM, ROM, I/O and timer are all on a single chip
2. Designer decides on the amount of ROM, RAM and I/O ports.	2. Fix amount of on-chip ROM, RAM, I/O ports
3. High cost	3. Low cost
4. General-purpose	4. Single-purpose
5. High speed	5. Low speed
6. Higher Power Consumption	6. Low Power Consumption
ex – 8085, 8086	ex – 8051,AVR

c)

i. Shortly describe 82C55 PPI with its operational modes.

82C55 PPI is a programmable peripheral interface. It has three operational modes: Mode 0 (basic I/O), Mode 1 (strobed I/O), and Mode 2 (bi-directional I/O).

ii. Why is memory decoding necessary in computer system?

Memory decoding ensures proper addressing of memory locations, preventing conflicts. It's necessary for the CPU to access specific memory locations in a computer system.

d) Criticize the statement "More registers integration produces faster CPU".

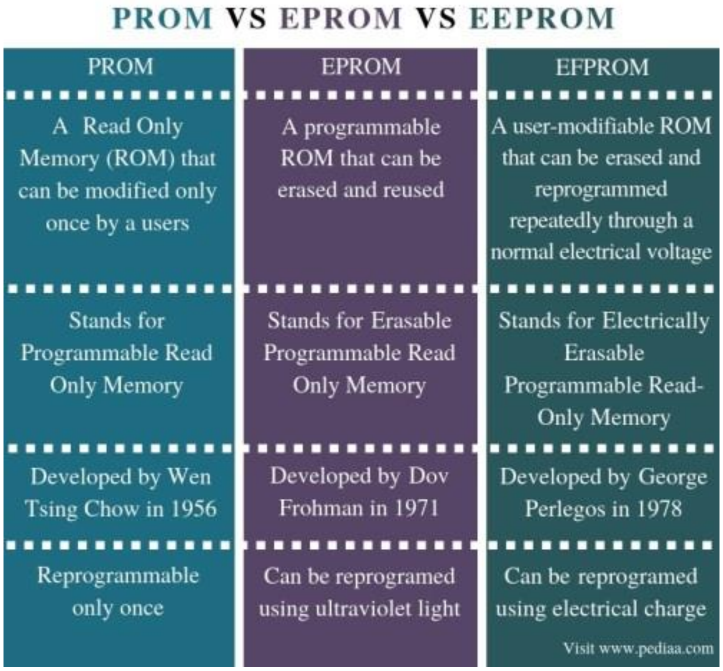
The statement is not entirely accurate. More register integration alone doesn't always guarantee a faster CPU. While registers are essential for storing and quickly accessing data, other factors like architecture, clock speed, and overall design also play a crucial role in determining CPU performance.

Example: Imagine a car with a bigger trunk (registers) but a slower engine (clock speed). Even though you have more space, the car may not necessarily go faster. It depends on the engine's speed and efficiency, just as CPU performance relies on various factors, not just the number of registers.

2.

a)

i) **Compare PROM, EPROM and EEPROM.**

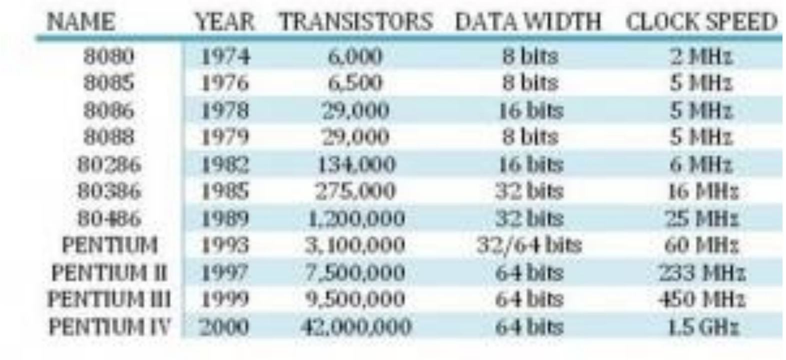


PROM VS EPROM VS EEPROM

PROM	EPROM	EFPROM
A Read Only Memory (ROM) that can be modified only once by a users	A programmable ROM that can be erased and reused	A user-modifiable ROM that can be erased and reprogrammed repeatedly through a normal electrical voltage
Stands for Programmable Read Only Memory	Stands for Erasable Programmable Read Only Memory	Stands for Electrically Erasable Programmable Read-Only Memory
Developed by Wen Tsing Chow in 1956	Developed by Dov Frohman in 1971	Developed by George Perlegos in 1978
Reprogrammable only once	Can be reprogramed using ultraviolet light	Can be reprogramed using electrical charge

Visit www.pediaa.com

ii) **Give the evolution of microprocessor from mechanical era to present (with important advancement)**



NAME	YEAR	TRANSISTORS	DATA WIDTH	CLOCK SPEED
8080	1974	6,000	8 bits	2 MHz
8085	1976	6,500	8 bits	5 MHz
8086	1978	29,000	16 bits	5 MHz
8088	1979	29,000	8 bits	5 MHz
80286	1982	134,000	16 bits	6 MHz
80386	1985	275,000	32 bits	16 MHz
80486	1989	1,200,000	32 bits	25 MHz
PENTIUM	1993	3,100,000	32/64 bits	60 MHz
PENTIUM II	1997	7,500,000	64 bits	233 MHz
PENTIUM III	1999	9,500,000	64 bits	450 MHz
PENTIUM IV	2000	42,000,000	64 bits	1.5 GHz

Apple M1	2020	Approximately 16 billion	64-bit	Not disclosed in the same way as traditional CPUs
Intel Core i9-11900K	2021	Not specified	64-bit	3.5 GH

b)

i) How does the DMA speed up CPU performance?

DMA (Direct Memory Access) speeds up CPU performance by allowing devices (like hard drives or network interfaces) to transfer data directly to or from memory without involving the CPU. This frees up the CPU to perform other tasks, improving overall system efficiency and speed.

Imagine you're copying files from a USB drive to your computer. Without DMA, the CPU would have to manage each step of the transfer, taking time and effort. With DMA, the USB drive can send data directly to the computer's memory, letting the CPU focus on more important tasks like running applications. This speeds up the overall process and makes your computer more responsive.

ii) Distinguish between SRAM and DRAM.

SRAM	DRAM
It can store data as long as electricity is available.	It saves data for as long as the power is on or for a few moments if the power is turned off.
SRAM has a storage capacity of 1 MB to 16 MB in most cases.	DRAM, which is often found in tablets and smartphones, has a capacity of 1 GB to 2 GB.
The storage capacity of SRAM is low.	The storage capacity of DRAM is higher than SRAM.
SRAM is more expensive than DRAM.	DRAM is less expensive than SRAM.
It is comparatively faster.	It is comparatively slower.
SRAM is used in cache memories.	DRAM is used in main memories.

c)

i) Why is stepper motor so called?

Stepper motors are so called because each pulse of electricity turns the motor one step. Stepper motors are controlled by a driver, which sends the pulses into the motor causing it to turn.

ii) “All coprocessors are peripherals but all peripherals are not coprocessors”-Explain the statement.

Imagine your computer is like a team, and devices you connect to it are players.

- "All coprocessors are peripherals": This is like saying, "All goalkeepers are players." A coprocessor is a special kind of player that helps with specific tasks, like a math coprocessor assisting in calculations.
- "All peripherals are not coprocessors": Now, think of other players on the team, like a striker (printer) or a defender (keyboard). They're players too, but they don't do the same job as the goalkeeper (coprocessor).

Example:

- Peripheral (Not a Coprocessor): A printer is like a striker. It's part of the team (connected to the computer), but it doesn't assist in the same way a coprocessor (goalkeeper) does with calculations.
- Coprocessor (Also a Peripheral): Now, add a math coprocessor to the team. It's like a specialized goalkeeper, helping with specific tasks related to calculations while still being part of the team.

d) "CPU actually works on binary digits"- Explain this statement.

Think of binary digits like a secret language that the CPU understands. In this language, 0 means "off" or "no," and 1 means "on" or "yes." Just like a light switch that can be either on or off.

Example:

If you want the CPU to understand the number 5, you express it using 0's and 1's. In binary, 5 is 101. The CPU reads this binary code and knows you're talking about the number 5.

3.

a)

i) Suppose you would like to transfer data from your disk drive to flash drive by using DMA controller. Explain whole procedure in details to complete the activities.

Step-by-step guide for transferring data from a disk drive to a flash drive using a DMA controller:

- **Prepare DMA Controller:** Set up the DMA controller for data transfer.
- **Create Buffers:** Allocate space in the computer's memory for temporary storage.
- **Configure DMA Channels:** Specify where data comes from (disk) and goes to (flash drive).
- **Activate DMA Channels:** Start the DMA controller to handle the transfer.
- **Copy from Disk to RAM:** Let the DMA controller move data from the disk to the allocated memory.
- **Move to Flash Drive:** Transfer data from the memory to the flash drive.
- **Wait for Finish:** Pause and wait for the DMA controller to complete the transfer.
- **Clean Up:** Disable DMA, release memory, and finish the process**.**

ii) Describe overlapping data movement mechanism of DMA.

Overlapping data movement in DMA (Direct Memory Access) means that while the DMA controller is transferring data between two devices (like a disk drive and a flash drive), the CPU can do other tasks simultaneously.

1. **Start Data Transfer:** DMA controller begins moving data from the source (e.g., disk drive) to the destination (e.g., flash drive).
2. **CPU Continues Processing:** While the DMA is handling the data transfer, the CPU is free to perform other operations or tasks.
3. **Overlapping Operations:** The key is that the CPU's work and the DMA data transfer happen at the same time, overlapping each other.
4. **Efficient Resource Utilization:** This overlap makes the overall system more efficient because both the CPU and DMA controller are utilized simultaneously.

Example: Imagine you're copying files from a USB drive (DMA is handling this) while your computer's CPU is running a software update in the background. This overlapping allows the update to happen without waiting for the file transfer to finish.

b)

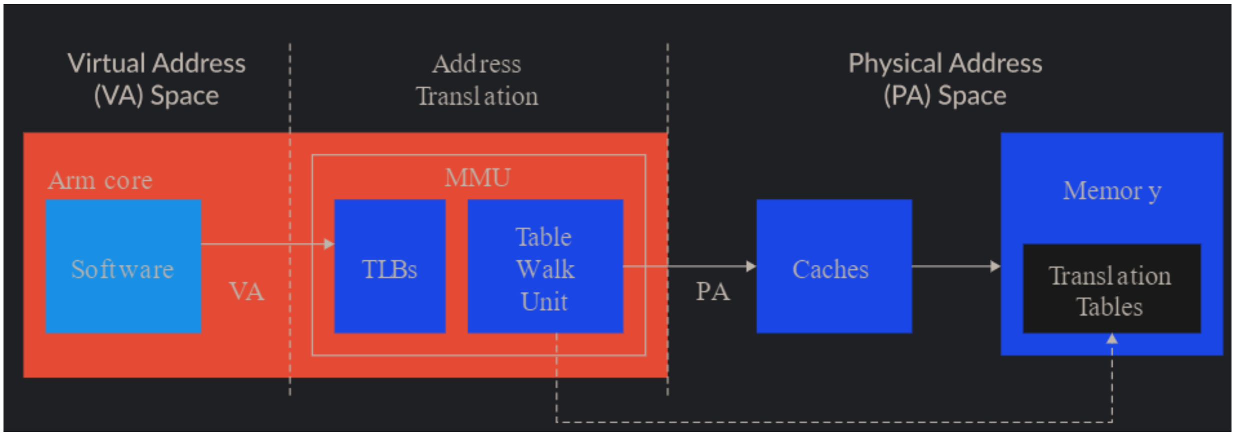
i) Describe the responsibility of memory management unit in computer system.

The Memory Management Unit (MMU) is responsible for the translation of virtual addresses used by software to physical addresses used in the memory system.

The MMU contains the following:

- ✓ The table walk unit, which contains logic that reads the translation tables from memory.
- ✓ Translation Lookaside Buffers (TLBs), which cache recently used translations.

All memory addresses that are issued by software are virtual. These memory addresses are passed to the MMU, which checks the TLBs for a recently used cached translation. If the MMU does not find a recently cached translation, the table walk unit reads the appropriate table entry, or entries, from memory, as shown here:



ii) Calculate the number of page table entries that are needed for following combinations of virtual address size(n) and page size (p)

n	**$P=2^p$**	**#PTE**
16	**4K**	16 entries
64	**16k**	2^{50} entries

The number of page table entries (PTE) needed can be calculated using the formula:

Number of PTE = Virtual Address Size/Page Size

Let's calculate the number of PTE for the given combinations:

Problem-1: First row of table,

Given virtual address size 2^{16}

Given page size is 4K = 4x1024 bytes=2^{12} bytes.

Therefore, the number of page table entries (PTEs) = $2^{16}/2^{12}$ = 2^4 =16 entries.

Problem-2: Second row of table,

Given virtual address size 2^{64}

Given page size is 16K = 16x1024 bytes=2^{14} bytes.

Therefore, the number of Page Table Entries (PTEs) = $2^{64}/2^{14}$ = 2^{50} entries

c)

i) Define handshaking.

Handshaking is a I/O control method to synchronize I/O devices with the microprocessor.

ii) Explain page fault handling mechanism in virtual memory system.

Page Fault Defined:

Imagine you have a big book (virtual memory) but can only read a few pages at a time (physical memory, like RAM).

Page fault happens when you need a page that's not in the part you can currently read (not in RAM).

Fixing a Page Fault:

Check if the needed page is already in the part you can read (in RAM).

If yes, great! If not, go to the big book (virtual memory) and bring the needed page to the part you can read (load it into RAM).

Update and Continue:

After bringing in the page, update your notes (page table) to remember where the pages are. Now, you can keep reading your book smoothly with the new page in the part you can read.

In simple terms, a page fault is like needing a page from a big book that's not currently in front of you. The computer checks where the needed page is, brings it if necessary, and updates its notes so it remembers for next time. This way, you can work with a big book even if you can only read a few pages at once.

d) "CPU actually works on binary digits"-Explain this statement.

Already noted.

4.

a) which Intel microprocessor addresses 1T of memory? Draw the block diagram of a computer system.

80486 through the Core2 Intel Microprocessor addressed 1T of memory.

CHAPTER 1

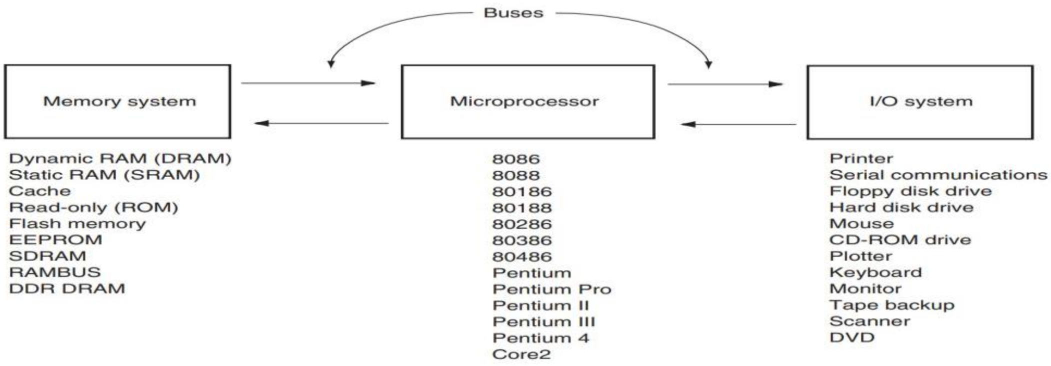


FIGURE 1–6 The block diagram of a microprocessor-based computer system.

b) What is displacement? How does it determine the memory address in a MOV DS: [2000H], AL instruction?

It is an 8 bit or 16 bit immediate value given in the instruction.

In the assembly instruction MOV DS: [2000H], AL:

- ✓ MOV: Move instruction.
- ✓ DS: Data Segment register, specifying the segment where the memory address is located.
- ✓ [2000H]: Square brackets indicate an absolute memory address, in this case, 2000H.
- ✓ AL: The content of the AL register will be moved to the specified memory address.

So, this instruction moves the content of the AL register to the memory location specified by the combination of the Data Segment register (DS) and the absolute address 2000H. It is a way to store data in a specific memory location within the specified segment.

c) **Explain the difference between the MOV BX, DATA instruction and the MOV BX, OFFSET DATA instruction.**

In x86 assembly language, both MOV BX, DATA and MOV BX, OFFSET DATA instructions are used for moving data, but they have different implications:

MOV BX, DATA:

- ✓ This instruction moves the value of the label DATA into the BX register.
- ✓ If DATA is a constant or a memory address, the actual value (or address) is loaded into BX.

Example:

```
DATA DW 1234
...
MOV BX, DATA ; Moves the value 1234 into BX
```

MOV BX, OFFSET DATA:

- ✓ This instruction moves the offset of the label DATA into the BX register.
- ✓ It's often used in the context of calculating addresses.

Example:

```
DATA DW 5678
...
MOV BX, OFFSET DATA; Moves the offset of DATA into BX
```

If DATA is located at, for example, address 0x1000 in the code segment, then MOV BX, OFFSET DATA would load 0x1000 into BX. The actual value stored at 0x1000 is not loaded into BX.

d) **What, if anything is wrong with a MOV AL, [BX][SI] instruction? Suppose that DS = 1200H, BX = 0100H, and SI = 0250H. Determine the address accessed by each of the following instructions, assuming real mode operation:**

(a) MOV [100H],DL

(b) MOV [SI+100H],EAX

(c) MOV DL,[BX+100H]

Here's the correct version:
MOV AL, [BX + SI]
This instruction means "move the contents of the memory location whose address is the sum of BX and SI into the AL register.

(a) DSx10+100H=12000H+100H=12100H

(b) DSx10+SI+100H=12000H+250H+100H=12350H

(c) DSx10+BX+100H=12000H+100H+100H=12000H

e) **How many bytes are stored on the stack by a PUSH AX? Show which JMP instruction assembles (short, near, or far) if the JMP THERE instruction is stored at memory address 10000H and the address of THERE is:**

a) **10020H**
b) **11000H**
c) **0FFFEH**
d) **30000H**

when a PUSH AX instruction is executed, 2 bytes are stored on the stack.

a **<u>If the memory address of THERE is 10020H</u>**,

10020H
-10000H
00020H → 0000 0000 0000 0010 0000
1 byte
1 byte displacement → short jump

b **<u>If the memory address of THERE is 11000H</u>**,

11000H
-10000H
01000H → 0000 0001 0000 0000 0000
2 byte
2 byte displacement → near jump

(c) short (d) far (short 1 byte, near 2 byte,far>near

5.

a) Write down the formats of the 8086-Core2 instructions. (a) The 16 bit form and (b) the 32 bit form.

16-bit instruction mode

Opcode 1–2 bytes	MOD-REG-R/M 0–1 bytes	Displacement 0–1 bytes	Immediate 0–2 bytes

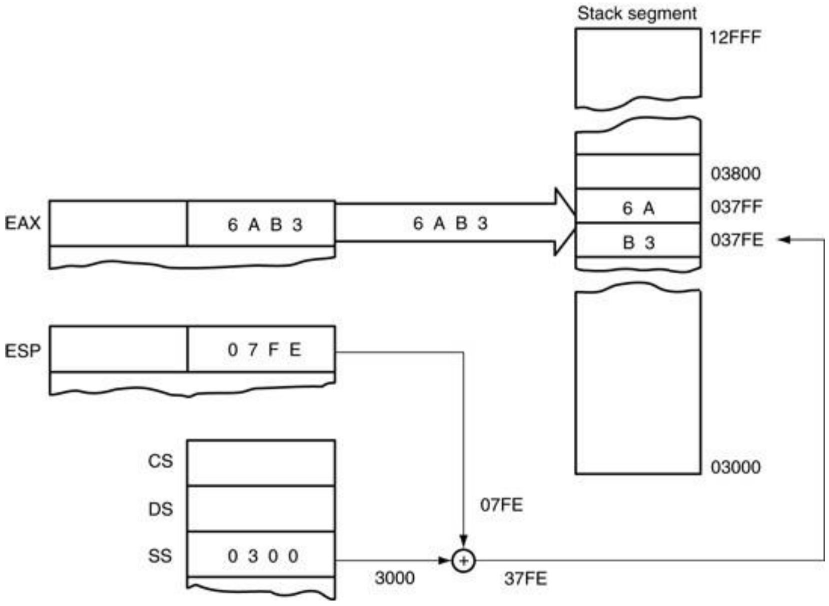
(a)

32-bit instruction mode (80386 through Pentium 4 only)

Address size 0–1 bytes	Register size 0–1 bytes	Opcode 1–2 bytes	MOD-REG-R/M 0–1 bytes	Scaled-index 0–1 bytes	Displacement 0–4 bytes	Immediate 0–4 bytes

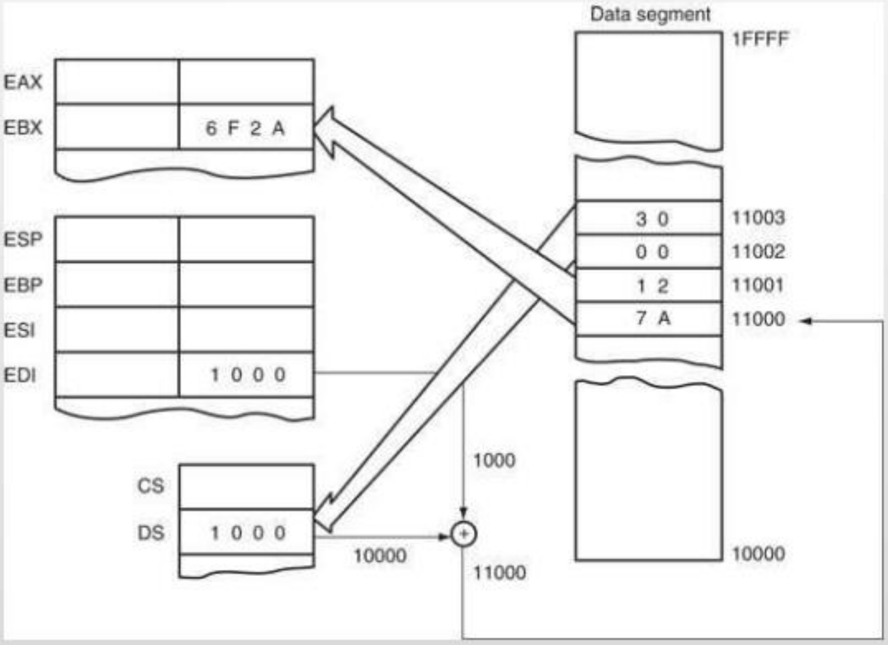
(b)

b) The effect of the PUSH AX instruction on ESP and stack memory locations 37FFH and 37FEH. [Assume SS=0300, ESP=07FE]

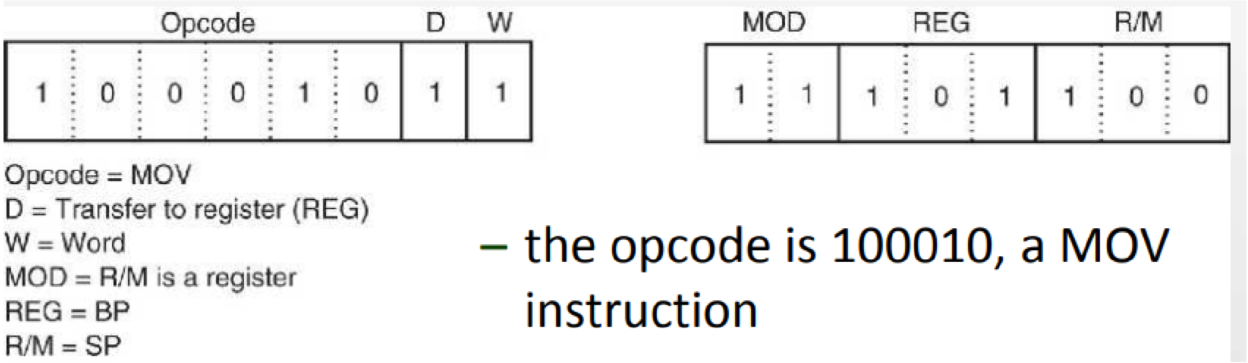


c) **Draw a diagram and show the LDS BX, [DI] instruction loads register BX from addresses 11000H and 11001H and register DS from locations 11002H and 11003H. This instruction shows at the point just before DS changes to 3000H and BX changes to 127AH. The initial value of DS=1000 and EDH 1000.**

Figure 4–17 The LDS BX,[DI] instruction loads register BX from addresses 11000H and 11001H and register DS from locations 11002H and 11003H. This instruction is shown at the point just before DS changes to 3000H and BX changes to 127AH.



d) **Convert machine code 8BEC to equivalent assembly instruction.**



Opcode						D	W
1	0	0	0	1	0	1	1

MOD		REG			R/M		
1	1	1	0	1	1	0	0

Opcode = MOV
D = Transfer to register (REG)
W = Word
MOD = R/M is a register
REG = BP
R/M = SP

– the opcode is 100010, a MOV instruction

MOV ESP, EBP

e) **If the start of a segment Is identified with DATA, what type of memory organization in in effect? What values appear in SP and SS if the stack is addressed at memory location 02200H?**

If the start of a segment is identified with "DATA," it indicates a memory organization known as "Data Segment" or "Segmented Memory Organization."

One possibility is 200H in both registers.

6.

a) **Define DAA and DAS. Show the process of addition with carry, show the carry flag (C) links the two 16-bit additions into one 32-bit addition.**

Decimal adjust after addition and decimal adjust after subtraction.

b) **What is wrong with the ADD RCX, AX instruction? If AX=1001H and DX=20FFH, list the sum and the contents of each flag register bit (C, A, S, Z and O) after the ADD AX, DX instruction executes.**

RCX is 64 bit and AX is 16 bit. We can't used mixed register for instructions.

AX=3100H, C=0, A=1, S=0 and Z=0.

c) **What is the difference between the NOT and the NEG instruction? List the number of data items stored in each of the following memory devices and the number of bits in each datum:**

i. **2K x 4**

ii. **1k x 1**

iii. **4k x 8**

iv. **16k x 1**

v. **64k x 4**

NOT is the 1's complement and **NEG** is the 2's complement.

- 2K*4 2K memory location and 4 bits are stored in each datum.
- 1K*1 1K memory locations and 1 bit stored in each datum.
- 4K*8 4K memory location and 8 bits stored in each datum.
- 16K*1 16K memory location and 1 bit stored in each datum.
- 64K*4 64K memory location and 4 bits stored in each datum.

d) **Which type of JMP instruction (short, near or far) assembles for the following:**

i. **If the distance is 0210H bytes.**

Near

ii. **If the distance is 0020H bytes**

Short

iii. **If the distance is 10000H byte.**

Far

e) **Contrast minimum and maximum 8086/8088 operation. Explain the operation of the pin.**

Minimum mode involves a single 8086 microprocessor, while maximum mode supports multiple processors like the 8087 and 8089.

Operation of pin in 8086 Architecture:

- **Address and Data Bus:** Transmit address and data information.
- **Control Pins:** Manage operations (read, write, etc.).
- **Power and Ground Pins**: Provide electrical power and ground.
- **Clock Pin**: Synchronizes internal operations.
- **Interrupt Pins:** Handle external interrupts.
- **Segment Register Pins:** Manage code, data, stack segments.
- **Flag Pins:** Hold status flags (zero, carry, etc.).
- **Reset Pin:** Resets the microprocessor.

Session: 2017-2018

1.

a) Define von Neumann machine. Write short report detailing the features of the Itanium 2 microprocessor.

A machine that stores the instructions of a program in the memory system.

b) What are program-visible registers?

Program visible register are the registers that are directly used in an instruction

c) Which registers are used as an offset address for the string instruction destination in the microprocessor?

Destination index register is used as an offset register for the string instruction destination.

d) Protected mode memory addressing allows access to which area of the memory in the 80286 Microprocessor?

above the first 1M byte of memory, as well as within the first 1M byte of memory.

2.

c) if near jump uses a signed 16-bit displacement, how can it jump to any memory location within the current code segment?

Intra-segment jump.

3.

c) contrast the operation of a JMP DI and JMP[DI].

The JMP DI instruction jumps to the current code segment location addressed by the contents of DI, and the JMP [DI] instruction jumps to the current code segment location addressed by the contents of the data segment location addressed by DI.

d)What conditions do the QS1 and QS0 pins indicate about the 8086/8088?

These two are output pins are used to indicate the *instruction queue status.*

4.

b) which conditional jump instructions test both Z and C flag bits?

JBE