19-20

1 (a)

(i) How large is windows application program area?

(Ans.) For modern windows system the user mode application adress space depends on the OS and architecture:

- 32 bits windows
- 32 bit with/3 GB switch:
- 64 - bit windows

(ii) Distinguish between microcontroller and microprocessor:

• Microcontroller:

• CPU only — no on chip RAM, ROM, or I/O

• Requires external memory and peripheral chips

• used where higher performance and flexible memory/peripherals are needed.

Example: Intel 8085, 8086 (are used with external peripheral/memory)

• microcontroller:

• CPU + on chip memory (RAM/ROM or I/O)

• Designed for embedded control application small, low cost, low power.

• Easier to deploy for single board embedded system

Example: 8051 - family, PIC, AVR, ARM

Feature of 8051 microcontroller:

- 8 bit CPU/ALU
- On-chip memory: 4KB on chip ROM and 128 byte on chip RAM
- I/O ports: 4 parallel 8 bit I/O ports → 32 I/O lines
- Register
- Timer
- Serial communication
- Interrupts
- Bit adressable memory.
- special function registers
- on chip oscillator.
- Boolean
- Low cost widely used in embedded system

1(b)

program visible registers are those that can be directly accessed and modified ~~and modified~~ by a programmer using instruction in assembly language.

Example: AX BX CX DX - general purpose data register

CS DS SS ES - segment register

IP → (instruction pointer) → next instruction adress.

Flag register → status bits (zero, carry, sign.

Purpose of IP / EIP register:

- IP (instruction pointer) in 8086
- It holds the offset adress of the next instruction to be executed
- Together with the code segment register it forms the physical address.

physical adress = (CS × 10H) + IP

calculate Real-mode memory location:

(i) DS = 1000H, DI = 2000H

(1000H × 10H) + 2000H

Physical adress = 12000H

(ii) DS = 2000H EAX = 00003000H

∴ (2000H × 10H) + 3000H

= 23000H

(iii) SS = 8000H ESP = 00009000H

∴ (8000H × 10H) + 9000H

= 89000H

Physical adress = (segmement × 10H) + offset

segment = block of memory

CS, DS, SS, ES

offset = position inside the segment

1.9

Flat mode memory system.

In flat mode, the entire memory is treated as one continuous block.

There are no visible adress to the programmer - all addresses look like simple linear address

Example: DS, CS SS all point to the same base 0..

The memory looks like 0 to 4 GB continuous.

protected mode:

- The 80286 introduced protected mode for memory management
- In this mode the processor can access up to 16 MB of physical memory
- Each memory segment can be upto 64 KB.

1. (D)

Difference between Register and memory:—

	Register	memory
Location	inside CPU	outside CPU
speed	very fast	slower
size	very small	Large
Access	Directly by CPU	indirectly via adress bus

Special Function of AX, BX, CX DX

1. Accumulator
2. Base Register
3. count Register
4. Data register

2 (a)

ⓐ Ans: wrong with mov. BL, CX

mov BL, CX - is invalid because

operands are registers of different sizes.

BL → 8 bit

CL → 16 bit.

correct: mov BL, CL → 8 bit to 8 bit

mov BX, CX → 16 bit to 16 bit

segment registers with mov, push. pop

segment register → CS, DS, SS, ES, FS GS.

mov AX, DS

PUSH DS.

pop ES

Q.2(b)

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

Displacement means or offset value added to a base or index register to form an effective address.

For MOV DS: [2000H], AL:

- Segment register = DS

Offset = 2000H

So, the physical address = DS X 10H + 2000H

2 (c)

What do the symbol indicate ?

Suppose DS = 0200H, BX = 0300H,

DI = 0400H

Find memory address for–

(i) mov AL, [1234H]

(ii) mov EAX [BX]

(iii) mov [DI], AL

Ans. The symbol [ ] means contents of memory address.

(i) mov AL, [1234H]

$= DS \times 10H + 1234H$

$= (0200H \times 10H) + 1234H$

$= 2234H$

(ii) mov EAX [BX]

$$= (DS \times 10H) + BX$$

$$= (0200H \times 10H) + 0300H$$

$$= 2300H$$

(iii) mov [DI], AL

$$(DS \times 10H) + DI$$

$$= (0200H \times 10H) + 400.$$

$$= 2400H$$

CamScanner

2 (d)

which base register addresses data in the stack segment

DS = 1300H, SS = 1400H, BP = 1500H

SI = 0100H

(I) mov EAX [BP+200H]

(II) mov AL, [BP + SI - 200H]

Ans: Base register address data in the Stack segment is the Base pointer

Determined the adress access:

Physical Adress = (SS ×10H) + Effective Adress

(i) mov EAX ; [BP + 200H]

$= (1400H \times 10) + (1500H + 200H)$

$= 14000H + 1700H$

$= 15700H$

(ii) mov . AL [BP + SI - 200H]

$= (1400 \times 10H) + (1500H + 0100H + 0200H)$

$= 14000H + 1400H$

$= 15400H$

2 (e)

Intra-segment jump: Jump to another location within the same code segment

Example: JMP, SHORT,
JMP NEAR

Inter-segment jump: Jump to a location in a different code segment.

Ex: JMP FAR

3.(a)

Describe the purpose of the D and w bits found in some machine language instruction

Ans

When an instruction like MOV, ADD, SUB is converted into machine language The CPU uses special bits to understand how the instruction work. This two bits are D and W.

D bits show the Direction of data transfer between the register and memory.

D	Meaning	
0	Data moves from register → memory	MOV [BX], AX
1	Data moves from memory → register	MOV AX, [BX]

W bit

W	Data size	Example
0	8 - bit data	AL BL CL DL
1	16 bit data	AX BX CX DX

* W bits decides the size of the data

* D bits decides who is source and who is destination.

REG Field Table

REG code	W=0	W=1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX

Given REG = 010 and W=0

The selected register is DL

3 (b) Identify the default segment

Register:

Register	Default segment
SP (stack pointer)	SS (stack segment)
EBX / BX (Base Register)	DS (Data segment)
DI (Destination index)	DS (Data segment)
SI (Source index)	DS (Data) segment

3(c)

① .DATA segment:

.DATA tells the assembler that this part of the program stores variables and constants

It means program is using segmented memory organization

The (DS) (Data segment) register points to this segment automatically

So, segmented memory organization is in effect. specially Data segment machine code conversion:

8B → opcode for mov r16.
(move 16 bit data)

07 → mod R/m byte means

[BX] as memory operand

and AX as the register

8B07H → mov AX, [BX]

means: move data from memory

at adress [BX] into AX

3.d

what directive indicate the start and end of a procedure? Explain what happens when the PUSH BX instruction executes:

Ans: A procedure is a small subprogram used for performing a specific task

Two assembler directive mark its start and end.

Directive	meaning
PROC	Start of procedure
ENDP	End of procedure.

Example:

```
SUM PROC
MOV AX, BX
ADD AX, CX
RET
SUM ENDP
```

Execution of PUSH BX:

when PUSH BX is executed:

- $SP = SP - 2 \rightarrow 0100H - 2 = 00FEH$

- BX contents are saved in memory [SS:SP]:

  BH → 0200: 00FEH

  BL → 0200: 00FFH

- SP now = 00FEH

BX is stored in the stack. stack grows downward, and sp decreases by 2.

3(e)

(i) Display a "?"

Ans.
```
mov DL, '?'
Mov AH, 02H
INT 21 H
```

output: ?

(ii) Read two decimal digits whoose sum is less than 10.

Ans:
```
MOV AH, 01H
INT 21H

MOV BL, AL
SUB BL, '0'

MOV AH, 01H
INT 21H
MOV BH, AL
SUB BH, '0'
```

ADD BL, BH

(iii) Display them and their sum on the next line with an appropriate message:

Ans:

```
LEA DX, MSG1
MOV AH 09H        ; Display message the sum of
INT 21 H

MOV DL '2'
MOV AH, 02H
INT 21H

LEA DX, MSG2
MOV AH, 09H       ; Display "AND"
INT 21 H
```

```
MOV DL, '7'

MOV AH, 02H
INT 21H

LEA DX MSG3
MOV AH, 09H        ; Display "IS"
INT 21 H

MOV DL, '9'
MOV AH , 02H
INT 21 H     ; show Sum
```

Output:

?27

The sum of 2 and 7 is 9.

and save the sum in DH

4 Develop a short sequence that Adds AL, BL, CL, DL, ~~AH~~.

4(a) Q whats wrong with ADD RCX AX instruction

RCX is a 64 bit register

and AX is 16 bit

in 8086 and 8088 both operands of ADD must be the same size.

So, ADD RCX AX is invalid.

Develop short sequence of Register: -

```
mov DH, 00H
ADD DH, AL
ADD DH, BL
ADD DH, CL
ADD DH, DL
ADD DH, AH
```

Now DH contain the sum of AL, BL, CL, DL and DH

4(b)

Instruction	operation	store resut.
Sub	performs $A \leftarrow A-B$	yes
CMP	compares A and B (by A-B)	No

① multiply AL by 8 :-

shifting left by 3 bits $= \times 8$

```
MOV CL, 3
SHL AL, CL.
```

(ii) Divided 32142 by 4

```
mov AX, 32142
mov BL, 4
DIV BL
```

Quotient = 8035, stored in AX

4(c)

Explain what JMP AX instruction accomplishes.
Identify it as near or far Jump and list five flag bits tested by conditional jump instruction.

Ans.

Transfers program control to the adress stored in AX. The next instruction execute from that address.

Because only IP changes and CS remains the same. It is a near Jump.

Five main flag bits used (by conditional Jump: —

Flag	Example Jump
ZF (zero Flag)	jump if zero (J/Z)
CF (carry Flag)	JC - jump if carry
SF (sign Flag)	JS - jump if sign set
OF (overflow Flag)	JO - jump if overflow
PF (parity Flag)	JP