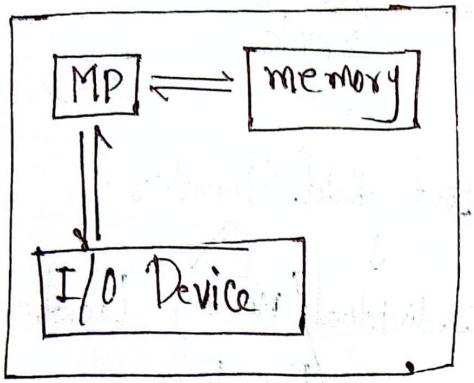
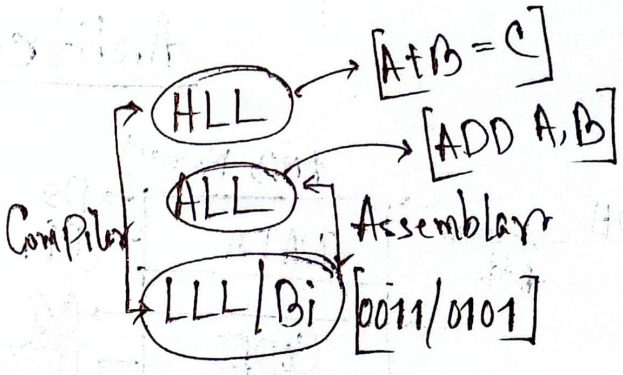
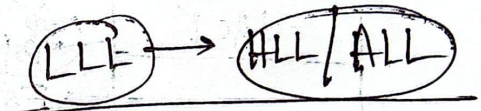
MicroProcessor 8086

Date - 25.02



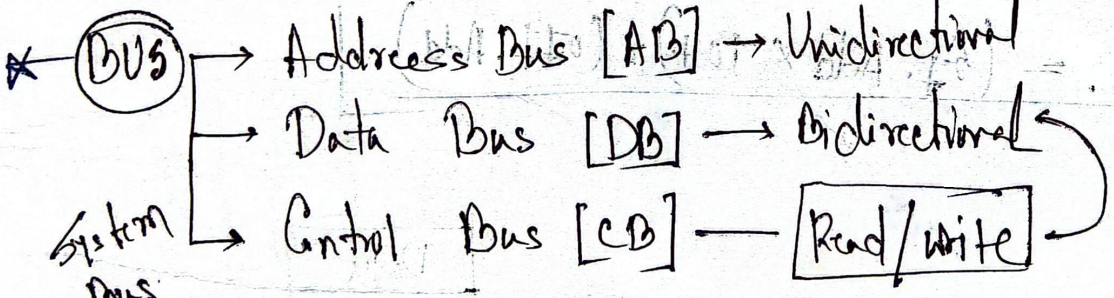
Computer System





Decoding

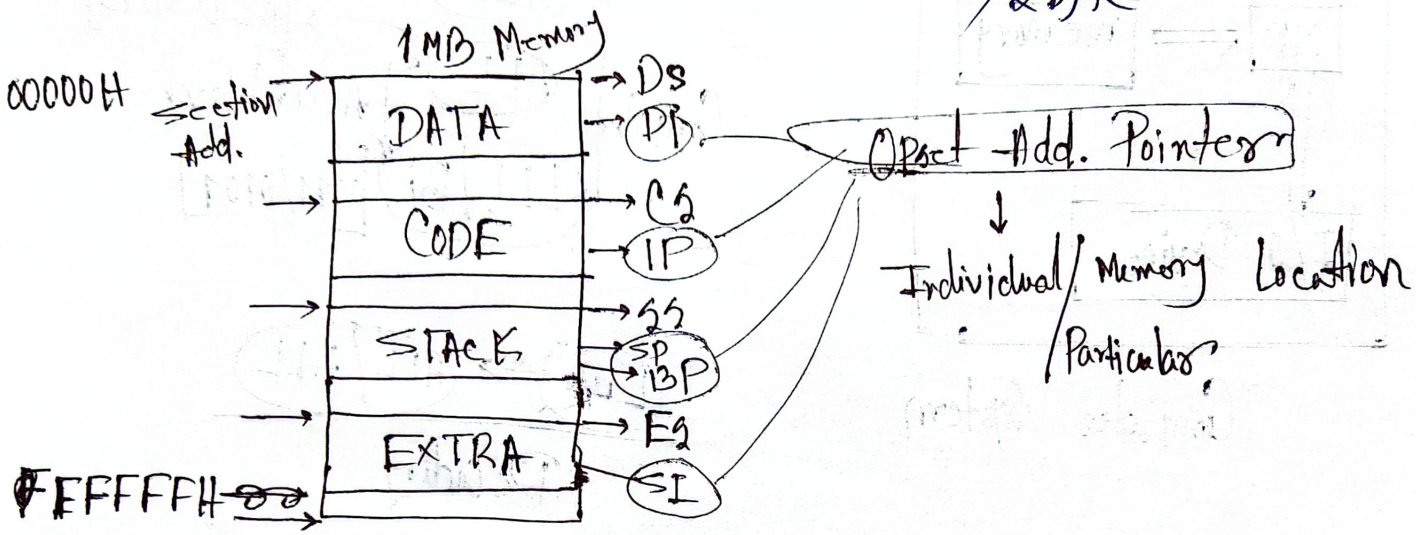
(#) MP instruction Cycle have 3 Part → Fatch, Decode, Execute



Support [20 bit] Address Bus

Architecture [16 bit MP] (8086)

2 byte



Section / Add → segment → [DS / CS / SS / ES]

* [Physical Add. = (Seg. Add × 10H) + (Off. Add)]

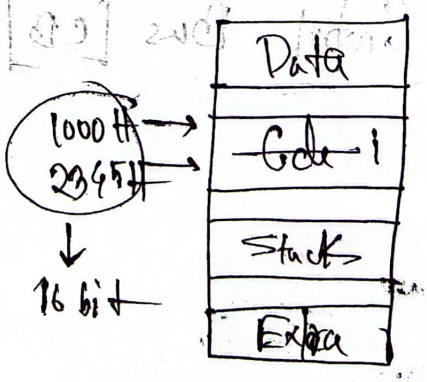
Example:

PA = (1000H × 10H) + 2345H

= 10000H + 2345H

= [12345H]

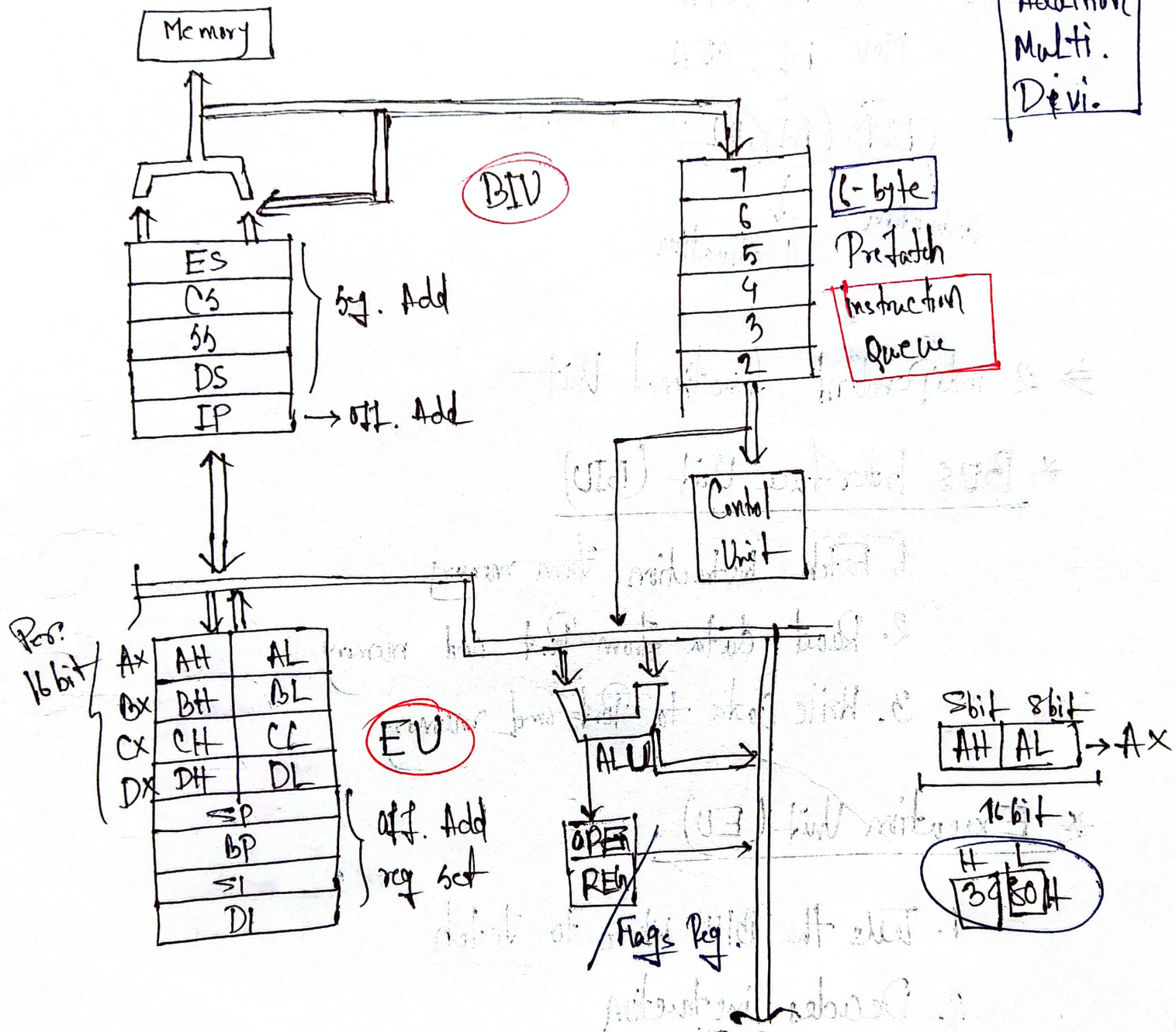
20 bit



(*) 8086 support maximum 6 byte instruction.

Aritmatic Operat^{n}

- Sub.
- Addition
- Multi.
- Divi.



Ax, Bx, Cx, Dx } Data Register

SP, BP } Pointer Reg.

SI, DI } index Reg.

* Example: $\underset{1\,byte}{04H} + \underset{1\,byte}{05H} = \underset{1\,byte}{09H}$

Ans:
```
MOV AL, 04H
MOV BL, 05H
```

⇒ 2 independent functional Unit

* Bus Interface Unit (BIU)

1. Fatch instruction from memory
2. Read data from Port and memory
3. Write data to Port and memory

* Execution Unit (EU)

1. Tells the BIU where to fetch
2. Decodes instruction
3. Executes instruction.

(*) **ALU():**

Add, subtract, AND, OR, XOR, increment, decrement, Complement, shift binary numbers.

(#) Total (14) Register [16 bit]

(1.) Data Reg → AX, BX, CX, DX } Total Four

(2) ADDRess Reg.
- Seg. Reg.
- Pointer Reg.
- Index Reg.

(3) Status Register / Flags Reg. [Hold Current status] → Status Flags, Control Flags

(4) Temporary Register [Holding Operands]

For starting Add. — Val × 10

Ending Add. — (Starting add. + OFFFFh)

~~Physical Add. Calculation~~

Ex: (i) 1000H
(ii) 1234H

Soln:

Seg. Reg	Str. Add (Val × 10)	End Add. (St. Add + 0FFFFH)
1000H	10000H	1FFFFH
1234H	12340H	12233FH ?

Ex: Physical Add for > A9FB:9872

Segment Add | Offset Add

PA = [Seg A × 10H + Off Add.]

= A9FBh × 10H + 9872h

= A9FB0h + 9872h

= A9822h

A9FB0
9872
A9822

1×16+(x)

1+F+8 = 15+8 = 23
= (1×16+8)
C

B+7
11+7 = 18 − 16
= 2

1 + 4 + 4 = (9)

Flag Register

16 bit — 9 Active flag

OF — Overflow Flag
DF — Direction "
IF — Interrupt "
TF — Trap Flag
SF — Sign "
ZF — Zero Flag
AF — Auxillary Carry "
PF — Parity "
CF — Carry "

Segment Register :

Memory Seg	Sg register	Off. Reg.
Data Seg. —	DSR —	source index (SI)
Code " —	CSR —	Instruction Pointer (IP)
Stack seg. —	SSR —	Stack Pointer (SP) / Base Pointer (BP)
Extra Seg. —	ESR —	Destination Index (DI)

Low level Langug

→ Machine langug.

→ Assembly "

* Assembly Langu → SUB AX, BX
* machine " → 000101 00001 00100

Program Statement :

1. Name Field [Colon(:)]
2. Operation " [MOV, SUB --]
3. Operand " [ADD AX,2 → Two Operand field]
4. Comment "

MOV CX,0 ; move 0 to CX

; → Comment

Assembler ignores any of things after (;)

MOV Word1, Word2
XCHG Word1, Word2
} illegal [Memory to memory move is illegal]

MOV AX, Word1;
XCHG Word1, AX
} Legal

MOV AX, Byte1 (Illegal)

Ex: NEG BX [BX = 0002]

Soln:

Before	After
0002	FFFE
BX	BX

```
  0    0    0    2
0000 0000 0000 0010
1111 1111 1111 1101
                1    1
          10000 1110
```

0000
1

Statement

1. B = A

2. A = 5 - A

Translation

```
MOV AX, A;
MOV B, AX;
```

```
MOV AX, 5 ; Put 5 in AX
SUB AX, A ; AX .... 5 - A
MOV A, AX ; Put it in A
```

③ $A = B - 2*A$

```
MOV AX, B  ; AX has B
SUB AX, A  ; AX has B-A
SUB AX, A  ; AX has B-2*A
MOV A, AX  ; make results to B
```

Different interface used in 8086 microprocessor:

1. Memory interface [DRAM/SRAM/ROM]
2. Input/output "
3. Interrupt "
4. Direct Memory Access Interface [Memory → I/O Device]
5. Bus interface

Addressing Mode:

1. Immediate Add. mode
2. Register " "
3. Direct " "
4. Indirect " "
5. Indexed " "
6. Based Add. mode

MicroProcessor Vs MicroController :

(1) MP is the heart of ComPuter System	(1) MC is the heart of an embedded system.
(2) It is only a Processor, so memory and i/o devices need to be Connected externally	(2) MC has a Processor along with internal memory and i/o devices.
(3) Memory and i/o device need to be Connect externally, so the circuit becomes larger.	(3) Memory and i/o are already Present and the internal circuit is small
(4) You Can't use it in Compact system.	(4) You Can use it in Compact sys.
(5) Cost of the entire system high	(5) Cost of the entire sys. low

FAHIM REZA | C211046
fahim.bdctg51@gmail.com