

## Syllabus :

- 1) Microprocessor, definition, elements, use and necessity (From first class)
- 2) Lecture #1 (8086 Architecture)
- 3) Lecture #2 (8086 Registers)
- 4) Lecture #3 (8086 Flag Registers)
- 5) Lecture #4 (8086 Memory Segmentation)
- 6) Lecture #5 (8086 Memory Bank Concept)
- 7) Assembly Language From Manual (chapters 3, 4, 5) the topics discussed and problem solved.

(17) (2)

## Microprocessor

"The Microprocessor is the integrated circuit which contains all the function of the CPU (Central Processing Unit) of a computer."

Microprocessor any of a type of miniature electronic device that contains the arithmetic, logic and control circuitry necessary to perform the functions of a digital computer's central processing unit. The production of inexpensive microprocessors enable computer engineers to develop microcomputers.

(Q.T.M.)

Example: A laptop is the best example where microprocessor is used.

\* What are the elements/components of microprocessor?

Structure of microprocessor will

with its functions and their



With the help of 3904 chip to construct

microprocessor to find

ALU

Control Unit

Register Array

memory units

(P.T.O)

b32u

(3)  
ALU: The arithmetic and logic unit (ALU) performs math computations, such as subtraction, addition, division and Boolean functions. It is a type of logic used for circuit designs. The ALU also executes comparisons and logic testing. The processor transmits signals to the ALU to perform appropriate operations on the data.

Control Unit: Control unit (CU) receives signals from the CPU, which instructs the control unit to move data from microprocessor to microprocessor. The control unit also directs the arithmetic and logic unit. It consists of multiple components such as decoders, clock and control logic circuits.

(6)

These device transmit signals to

certain locations on the microprocessor.

These devices are called

to system bus. (bus means bus)

ATI example: timer not been used

Registers:

Microprocessor has temporary data

holding places called Registers. These

Memory areas maintain data, such

as computer instructions, storage

addresses and other data. Each

register has a specific function,

such as instruction register,

program counter, memory address

Registers also have different

functions.

Example: A program register hold the

address of instructions taken

from random access memory.

Scanned with CamScanner

(5)

Memory: That means Cache Memory.  
Mainly Discuss RAM. Because ROM are  
fixed. Some advanced microprocessors  
have memory caches, which retain  
the last data used by the CPU. Memory  
caches speed up the upcoming computing  
process, because the CPU does not have  
to go to the slower RAM to retrieve  
data. Many computers have level 1 or  
level 2 caches; Some systems have level  
3 caches. The cache level indicates the  
order in which the CPU checks  
for data, starting with level 1.

Manufacturers often integrate level  
2 and level 3 caches into the  
microprocessor, which enhances  
processing speed.

(6)

## \* Life cycle of Microprocessor

→ It has two phases - Fetch & Execute.

Fetch operation is to get instruction from memory and execute operation is to perform the instruction.

main → URG soft → how is Bob fast soft

Registers, peripherals, bus → badge culture

Send a fetch request → Fetch phase

bit of address → work stop

to L1 cache send configuration from fetch

local and peripheral → Execution phase

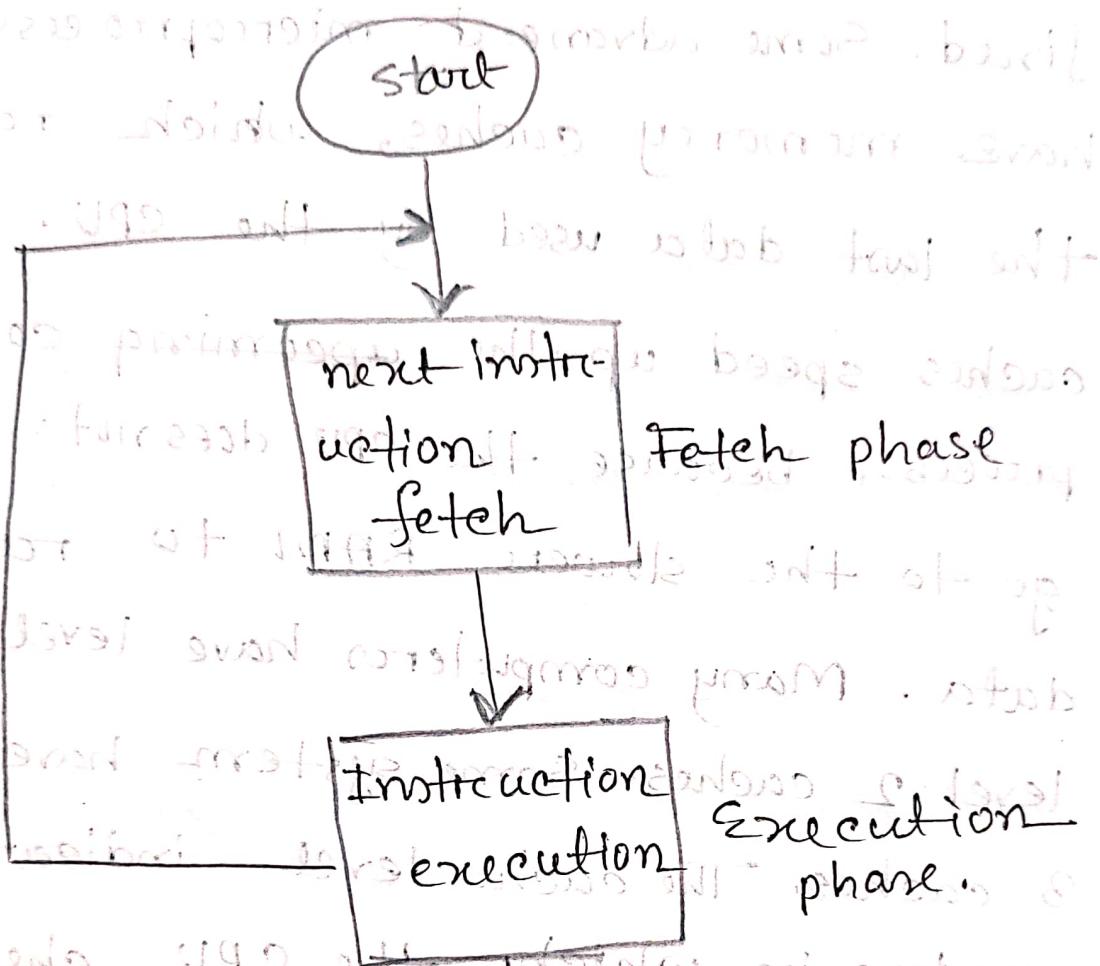
If no instruction → no instruction

→ address → result in registers

→ local config sent to cache stop

→ address → result in registers

→ local config sent to cache stop



→ The instruction cycle (also known

(3) (7)

as the fetch - decode - execute cycle or simply the fetch execute cycle) is

the cycle that the central processing unit (CPU) follows from boot - up

until the computer has shut down

in order to process instructions.

Then execute. It fetches the instruction

from the main memory and

executes them. This is done

repeatedly from when the computer

is booted up to when it is

shut down.

stop 1 ← fid 3 ← stop

stop 2 ← fid 1 ← bus

stop 3 ← ~~1000~~ three address

stop 4 ← fid 0 ← bus

stop 5 ← fid 2 ← bus

stop 6 ← fid 1 ← bus

stop 7 ← fid 0 ← bus

(x) (8)

no steps of binary addition instead add to no

## Lecture 1

at (steps of binary addition with pipeline)

### "8086 Architecture"

division of instruction into - fetch - decode - execute - fixed width - parallel (340) bytes

16000b bytes word 16 bit address 20 bit - bytes

8086 microprocessor: The 8086 is a 16 bit

microprocessor chip designed by Intel

between early 1976 and mid-1978,

when it was released. The 8086

gave rise to the x86 architecture

of Intel's future processors.

# Registers of 8086

# Bit → Binary Digit → 0/1

Byte → 8 bit → 1 byte.

word → 16 bit → 2 bytes.

Double word → 32 bit → 4 bytes.

Quad word → 64 bit → 8 bytes.

(9)

## 8086 Features / Characteristics

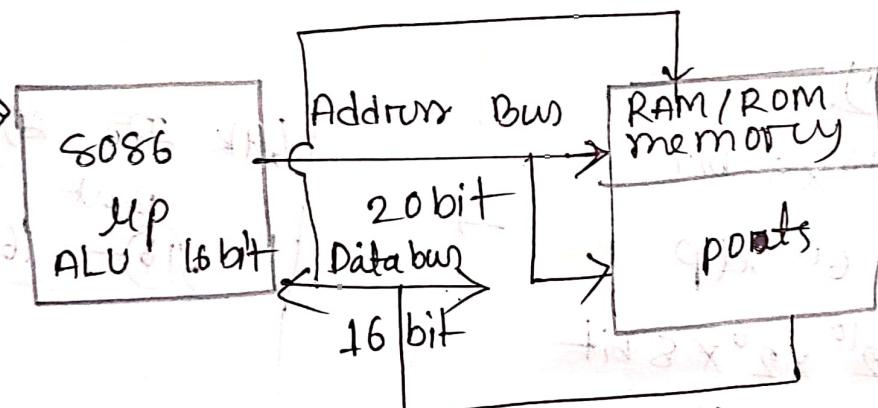
1) 8086 microprocessor is usually the 16

bit version of 8080 microprocessor.

2) ALU and Internal registers are all 16-bit.

3) Data Bus is 16-bit

4) Address bus is 20-bit.



20 bit are

unique  
address

Hex to digit

4 unit - 1 hex digit

RAM

8 bit  
8 bit

location

$$8+4+2+1 = 15$$

address Range

$00000 H = FFFFF H$

$2^3 \rightarrow 2^2 \rightarrow 2^1 \rightarrow F H$

(X) (8)

## Lecture 1

### "8086 Architecture"

Food worth swallows (99%) fine

8086 microprocessor: The 8086 is a 16 bit

microprocessor chip designed by Intel

between early 1976 and mid-1978,

when it was released.

gave rise to the x86 architecture

of Intel's future processors.

# Bit → Binary Digit → 0/1

Byte → 8 bit → 1 byte.

word → 16 bit → 2 bytes.

Double word → 32 bit → 4 bytes.

Quad word → 64 bit → 8 bytes.

## 8086 features / characteristics

1) 16 bit processor

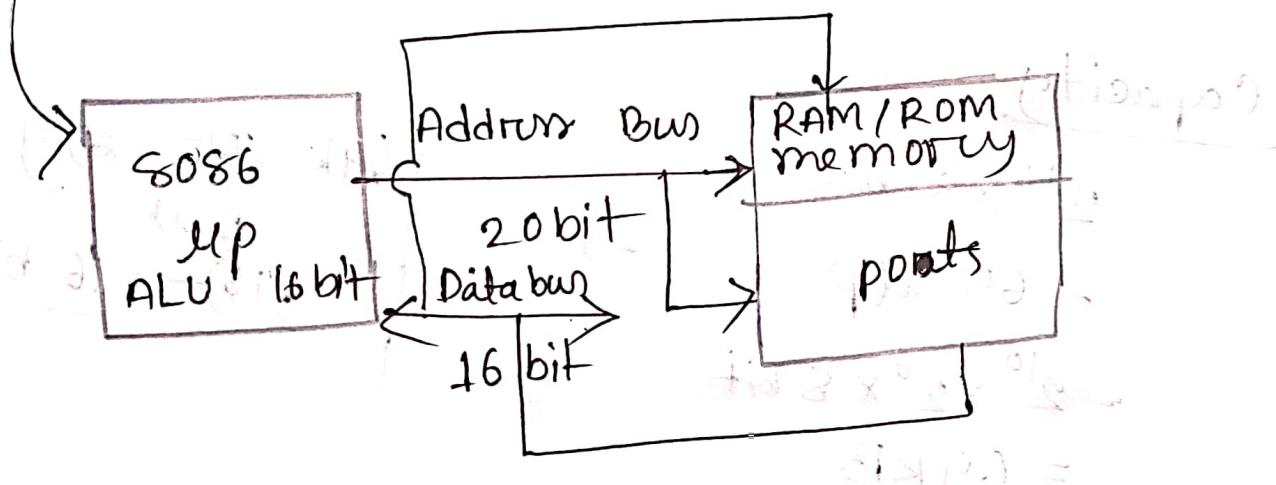
(processor for personal)

1) 8086 microprocessor usually the 16 bit + + bit version of 8080 microprocessor.

2) ALU and Internal registers set all are 16-bit.

3) Data Bus is 16-bit

4) Address bus is 20-bit

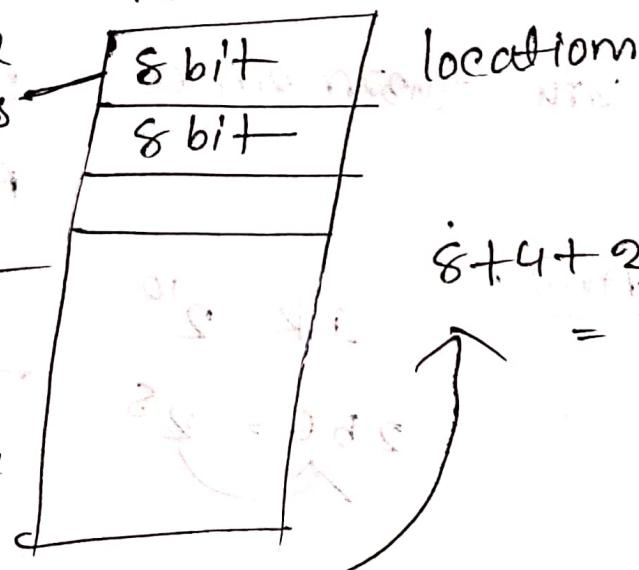


20bit are

unique address  
Hex to digit

4 unit - 1 hex digit

RAM



address Range

$00000 H = FFFFF H$

$2^9 2^8 2^7 2^6 \rightarrow F H$



(1)

(2)

~~2<sup>8</sup> × 8~~ bays having bidirectional bus (1)  
power 2<sup>8</sup> bays having Address Bus (2)  
Address Bus 8 bits at address 8 bits.  
possible address Bus 8 bits  
possible address Bus 8 bits  
possible address Bus 8 bits

5.11 Direct Addressing Capacity 1MB ( $2^{20}$ ) of Memory.

6) It can support up to 64K ( $2^{16}$ ) I/O ports.

7) It provides 14 16-bit registers.

8) It has multiplexed address and data bus ADD-AD15 and A16-A19.

9) Memory is organized as an array of bytes.

10) Memory is segmented.

11) Architecture: Designed for Powerful Assembly Language and Efficient High level Languages.

(12)

(13)

12) 8 and 16-bit Signed and Unsigned  
Arithmetic in Binary or Decimal  
is applicable including multiplication  
and division.

13) All options are possible.

13) Power supply is 5V

14) External clock signal generator  
8284 is used.

15) Word size is 16 bits and double  
word size is 4 bytes and 4F

16) CPU-DMA has DMA-0 and  
DMA-1

16) Range of clock Rates;

5 MHz for 8086; output to

8 MHz for 8086-2;

10 MHz for 8086-3;

17) MULTIBUS system Compatible

18) Interface Bus

(3)

(P)

- 18) The CPU is implemented in N-channel, depletion load, silicon gate technology (NMOS), and packaged in a 40-pin DIP or plastic package.
- 19) The 8086 operates in both single processor and multiple processor configurations to achieve high performance levels.
- 20) 8086 is designed to operate in two modes, Minimum and Maximum.
- 21) It can prefetches up to 6 instruction bytes from memory and queues them in order to speed up instruction execution.
- 22) A 40 pin dual in line package.
- 23) Address range from 00000H to FFFFFH

(14)

(a)

→ 8086 Architecture has two blocks  
for BIU and EU

- i) BIU (Bus Interface Unit)
- ii) EU (Execution Unit)

Sample and best in referring books off :-

BIU:  
• It is bus controller, which

→ i) The BIU handles all transactions  
with memory and I/O devices.

thus transferring data and  
addresses on the buses to load  
store instruction and variable.

2) It fetches instructions also  
Prefetches for empty space in  
queue.

3

at H00000 most upper 22000000  
and 111111

(15)

(a)

3) Calculates 20-bit PA (Physical Address)

4) Generates the bus Control signals to execute bus cycle.

bus control signals addressed to

### BIU Elements

i) Instruction queue (Q)

ii) Segment registers. (CS, DS, ES, SS).

iii) Instruction pointer (IP)

iv) Address adder. ( $\Sigma$ )

v) Bus Control Logic.

bus control logic

partitioned into

subblocks forward (if

if

configuration sequencing forward (if

configuration selection been finished (if

not input port (if

(16)

Number (16) is hit on instruction (P)

EU:

- EU means Execution Unit  
- ALU and Registers

- 1) Decodes the instruction fetched by BIU.
- 2) Generates the control signals for execution.
- 3) Executes the instructions.
- 4) Reflects the processor's status on flags.

EU Elements:

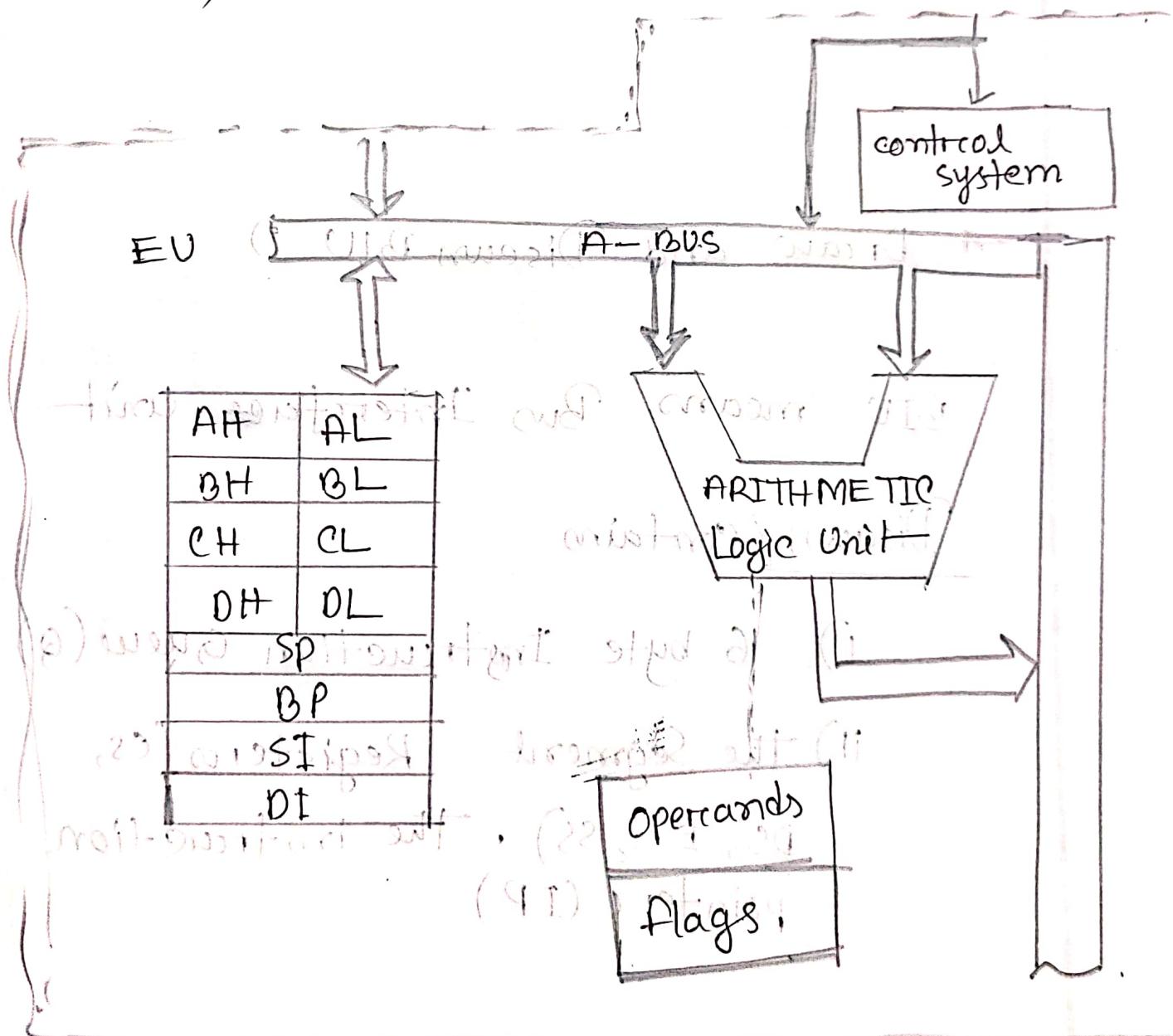
- i) Control circuitry;
- ii) Instruction decoder;
- iii) ALU;
- iv) General purpose registers.
- v) Pointer and Index registers
- vi) Flag register.

(17)

## Draw and Discuss EU

EU is execution unit

- i) Decodes Instructions Fetched by the BIU
- ii) Generate control signals.
- iii) Executes instructions.



The main parts are the system word

- i) Control Circuity
- ii) Instruction decoder
- iii) ALU
- iv) Registers.

→ Draw and Discuss BIU ?

BIU means Bus Interface unit

Discuss / Contains

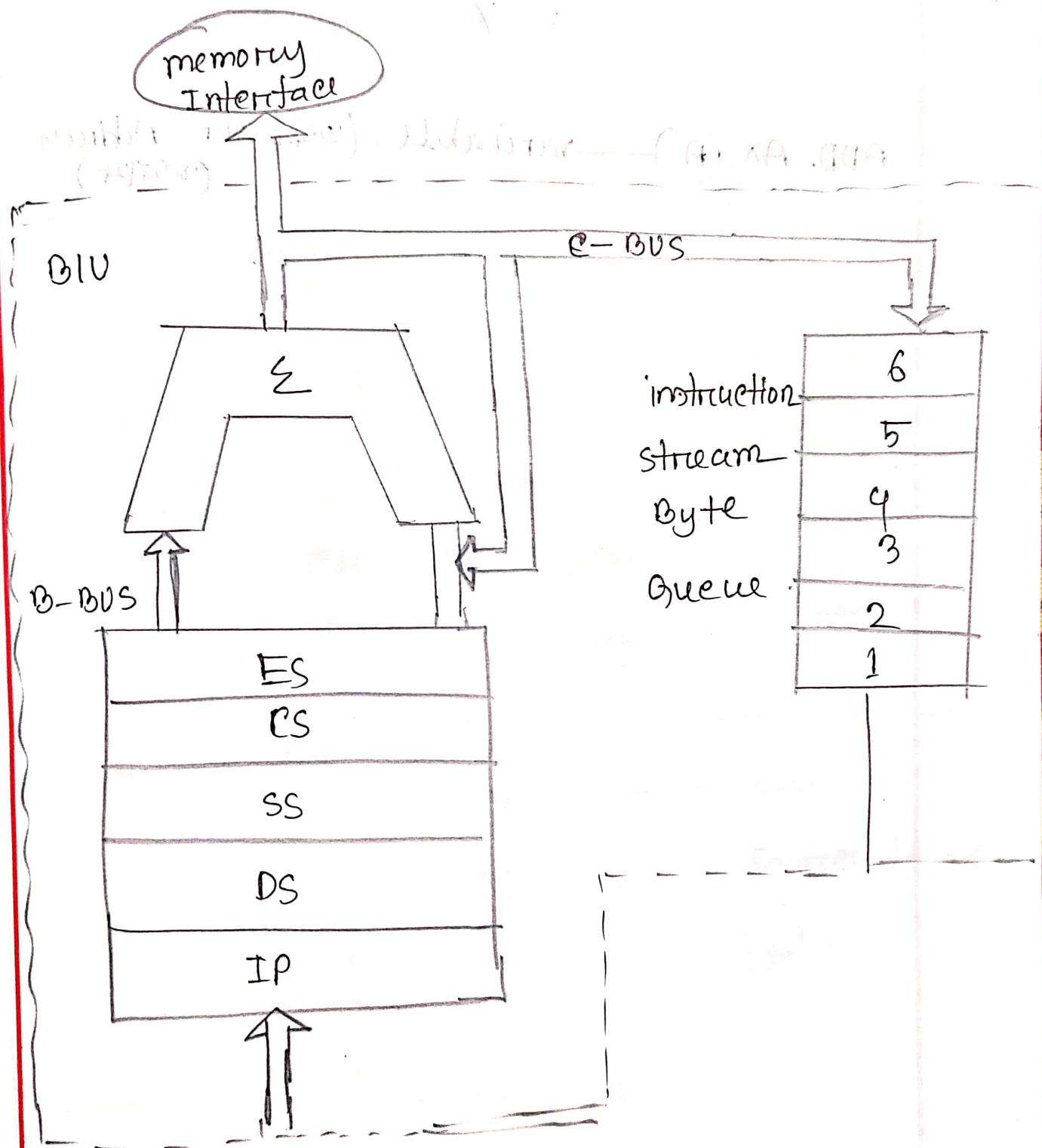
- i) 6 byte Instruction Queue (Q)
- ii) the Segment Registers (CS, DS, ES, SS). The instruction pointer (IP)

(19)

(20)

- The Address Summing block ( $\Sigma$ )

- Bus control logic



(20)

(a)

# ADD, (AX) (2) construct variable add

$$\begin{aligned} AX &= \frac{AX + 2}{5 + 2} \\ &= 7 \end{aligned}$$

ADD, AX, (A) variable. (start Address current)

(start address)

Memory

E

C

R

M

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

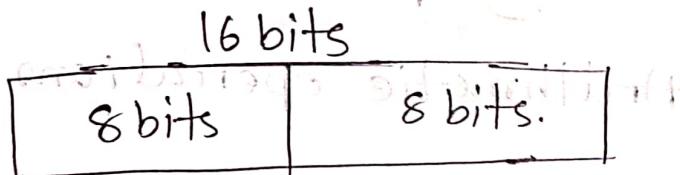
(21)

## 8086 Registers (chip #1) shall

### Lecture #2

#### General Purpose Registers.

Establish basic addressing modes (I)



AX AH → AL → Accumulator  
BX BH → BL → Base

CX CH → CL → Count

DX DH → DL → Data

SP → Stack Pointer

BP → Base

SI → Source Index

DI → Destination

Note: ~~move~~ AX, A 16 bit  
^  
Move AX, A" 8 bit

(22)

## Data Register

AX - Accumulator Register - ~~24(2110)~~ ~~125~~ use ~~0011 225~~ 1  
1000 10000000000000000000000000000000

I) Carry operands and results,

Carry operations

11) word ~~if multiply~~  $\rightarrow$  MUL

3) word divide → DIU

iv) Word I/O data transfer

• ~~is also~~) can be accessed as  $AT/AT_{xD}$

Port

IN AX, Port

Port AX

#  $(3) \times (2) = (6)$  — result ২টি product

- 2020-21 वर्ष 23

~~of~~ multiplier verbaal

↳ ၂၀၇၁။ ၂၂

~~tid 3)~~ multiplicand ~~multipliziert~~ mit : Stelle

10

8 11 AM 24A 3000W 10

Suppose: The linear function  $y = mx + b$

1101  
X 10

0 0 0 0 → product

1101

(f) 11010 → multiplicand / 1st step.

11010 — product

~~17-6200~~ - ~~XOTD 03307~~ Final  
~~17-6200~~ - result

MUL Multiplies (Reg / memory / Immediate)

Multiplicand  $\rightarrow$  AX, AL,

16bit  $\rightarrow$  8bit

Mov AX, multipli cand

MOV BX / CX / DX, multiplier

$$16 \times 16 = 32 \text{ bit}$$

Lower world 2740-AX

"higher world" DX

Scanned with CamScanner

(24)

(Ans)

multiplication result এবং প্রসেসর 16 bit word,  
 32 bit lower 3 higher words /

Assume  $A = 01010101$   
 Multiplication:  $8 \times 8 = 16$  (fixed point)  
 mov AL, multiplicand  
 mov BL, multiplier  
 MUL, BL  
 Product =  $01010101 \times 01010101 = 1000000000000000$   
 AL/BL/MUL

Product/Result - AX

Instruction: ADD AX, BX (addition), SUB AX, BX (subtraction)

~~Division~~

AX (XA) + b most digit-left

BX - Base Registers

i) Store data and Result.

ii) Carry offset address for

Data Segment.

iii) Accessed as BL/BH also.

XA = offset by 1000 word

XA = 4 byte word

Ex-Counter Registers

(25)

## Ex-Counter Registers:

Total - 4

Following 4 bits of YA

looped

i) Data Registers;

ii) String operation

iii) Loops instruction

iv) Repeated shift and rotate.

v) Can be accessed as CL/CH or

bit field.

bit 4 to bit 3

bit 2 to bit 1

## CL - Lower Byte EX :

i) Data Registers;

ii) Byte string operation;

iii) 8-bit counter loops

iv) Byte shift, rotate.

## IN/OUT

IN - read operation processor → memory

Data @MRA |

out → processor after data MRA

000 → X0

010 → X1

(26)

processor 214

port 214

IN → Destination      Source  
 AX - 16 bit operation      AL 8bit  
 $\hookrightarrow \boxed{000}$

OUT → Destination      Source  
 port      processor  
 16 bit " AX "      8 bit " AL "  
 (data register)

16 bit " AX "      8 bit " AL "  
 fixed AX  
 $\rightarrow \boxed{10000000}$

multiplication, division      AL  $\rightarrow \boxed{10000000}$

AH  $\rightarrow \boxed{0000}$       AH  $\rightarrow \boxed{1111}$

In/out operation      AL  $\rightarrow \boxed{10000000}$ , AH  $\rightarrow \boxed{0000}$

or 1 octet 8bit (8bit)

But normal data operating

① AH AH  $\rightarrow \boxed{0000}$  1. 10000000

AX  $\rightarrow \boxed{000}$

CX  $\rightarrow \boxed{001}$

DX  $\rightarrow \boxed{010}$

BX  $\rightarrow \boxed{011}$

## multiplication

(28)

$8 \times 8$

→

16 Result

AL R/M/Imi

AX

Registers

Memory

Immediate

fixed 270

$16 \times 16$

→

32 bit (result)

AX R/M/Imi

Registers AX, DX

8086 lower word  
AX, Higher word  
DX

## Division

$16 \times 8$

→ result

8 bit

8 bit

32 bit  
lower  
Higher

Quotient

Remainder

AX

AX

DX

AL: Lower byte of AX

i) Byte data transfer and store result, Arithmetic operation.

ii) Byte Multiply → MUL, 8 bit

iii) Byte Divide → DIV 8 bit

iv) Byte I/O data transfer → IN AL, port  
out port, AL

AH - Higher byte of AX

i) Higher byte AH → Port

ii) Data Registers

iii) Store data and arithmetic operation

( 29 )

( 30 )

CX =

for (i=0; i<=10; i++)

{  
    // loop body  
    // do something  
}

MOV CX, 1110 → Rightmost bit is 0

↓ Down counter for (i=0 to 9) and

value = 1024

Label

ADD AX, 1100 → 2+2=4

loop + 2

CX = CX - 1

Loop until back to 01 (initial value)

then decrement (0=CX) and

then add 1024

CX = 0 Instruction

Shift left

AL, 1

loop over CX=0

AL = 0001000

SHIFL AL = 0001000

Mov CL, 4, — 24 or just immediate

SHIFL, AL, CL

(30)

## D<sub>x</sub> - Data Registers

CHIPS (and) just  
i) Data carry and storing result  
for arithmetic operation.

ii) word multiply  $\rightarrow 16 \times 16 \rightarrow 32$

word operand multiplication result  
 $\downarrow$   
Higher w lower w  
DX AX

iii) word divide;

$P = S + Q \rightarrow 32/16 \rightarrow 2$   
 $\downarrow$  Higher w lower w  
DX AX

iv) Indirect I/O for port accessing.

Port ( $X0 = 0$ ) by command from CPU

v) Can be accessed as DL/DH

multiple for  $L = 0 = X0$

$X0 = 0$  more quick

DL DH [ + ] + link

00010000 = JA

00010000 = JA + HHR

16 bits total, 16 bits = P, 16 bits = R

(31)

(31)

## IP - Instruction Pointer:

- i) IP holds the offset address of the instruction to be executed.  
instruction add sp = 8x
  - ii) This is the program counter of the processor unit.
  - iii) During JUMP/ INT/ LOOP execution IP is modified by ~~some~~ displacement.  
if it is backward jump that means negative, and if it is forward jump that means label value is positive.
  - iv)  $\text{JMP } L \quad (\text{IP} = \text{IP} + L)$   
if it is backward jump that means negative, and if it is forward jump that means label value is positive.
- $I_1, I_2, I_3, \dots$  step by step sequence  
It takes ip to sequence step by step instruction

## Instruction Queue:

Instruction Queue is the element of Instruction Fetch off chip QI (i.e. of DIV).

6 bytes of 48 bit register.

↳ length of Queue.

to refer to memory and CPU itself.

1) The instruction Register of

8086

(i) The queue is 6 bytes, i.e. length

(ii) The maximum size of an

instruction of 8086 is size

of 1 byte.

• Address of instruction

at

upper address of queue

purpose of queue is

not to store queue of

(33)

(Pn)

## SP (STACK POINTER)

i) SP is used to point the stack top.

ii) Used by PUSH and POP instructions

iii) SP decrements by 2 after PUSH

iv) SP increments by 2 before POP.

v) Stack allows word operation

## BP (BASE POINTER)

i) BP is used to access stack using base addressing mode.

ii) MOV AX, A[BP]

iii) SP and BP the default reference to form a physical address is the

## Stack Segment (SS- Register)

- iv) Can be used to access data in other segments.

int 909 bios H209 pd bad (ii)

## Source INDEX REGISTER (SI) (iii)

- i) It can be for string or array operation. (v)
- ii) When string operations are performed the SI register points to memory locations in the data segment which is addressed by the DS register. (vi)
- Thus, SI is associated with the DS in string operations. (vii)

int 909 bios H209 pd bad (iii)

int 909 bios H209 pd bad (iii)

## DI : DESTINATION INDEX REGISTER

- i) This is also required for some string operations.
- ii) When string operations are performed, the DI register points to memory locations in the data segment which is addressed by the ES register. Thus, DI is associated with the ES in string operations.
- iii) The SI and DI registers may also be used to access data stored in arrays.
- iv) Stored in arrays.

## The Queue (Q)

- The BIU uses a mechanism known as an instruction stream queue to implement a pipeline architecture.
- This queue permits pre-fetch of up to 6 bytes of instruction code. Whenever the queue of the BIU is not full, it has room for at least two more bytes and at the same time the EU is not requesting it to read operands from memory, the BIU is free to look ahead in the program by pre-fetching the next sequential instruction.

(36)

(36)

## Segment Registers

- In 8086 / 88 the processor have 4 segments registers.
- Code Segment register (CS), Data Segment register (DS), Extra segment register (ES) and Stack Segment (SS) register.
- All are 16 bit registers.
- Each of the segment registers store the upper 16 bit address of the starting address of the corresponding segments.

+ Segment and Address register

## Combination's

CS:IP ~~consistency enforcement~~

150: (SS): SP-~~elp~~ SS: BP ~~emp~~ (260)

DS : BX DS : SI

safe base (2) solution - than strong  
DS: DI (for other operations)  
• note best (difficult) approach

ES : DI (fore string operation)

visiting • going off to work.

o 22000 bits/sec. at 0.1 sec. rate with 20% noise

244 to 2200bb0 p01b0fz 011-

• 2 hours ago 9:25 AM 100

## Lecture #3

8086 Flag registers

- top of programming by hand

Flag registers & conditions

→ 8086 Flag Register is 16-bit. Each bit indicates a flag. Each flag is a flip flop which indicates some conditions (0/1) produced by the execution of an instruction or controls certain operation of the EU.

i) The EU contains Flag Registers.

ii) It is 16 bit register.

iii) Among the 16 flags, 9 are active flags and remaining 7 are undefined.

c - l - f) 9 undefined flags

iv) 6 flags indicates some conditions -

(39)

(24)

- status flags, those set/ reset auto.

v) 3 flags are controlled. Flags, set/ reset by programmer to get certain facilities.

part. detail of principle part 2826 & if part 2827 part is also given error detection facility will give us

left part branching (LW) condition

Conditional Flags / Status Flag.

To control various instructions

• CF off

1.) CF - Carry Flag.

• If part of part condition CF off (i)

Holds the carry after addition

• If part of part condition CF off (ii)

Or the borrow after subtraction,

This is used by unsigned arithmetic.

If the result exceeds the

size of destination CF-1 ,

if bits are overflowed part of (iii)

(40)

(1P)

otherwise 0. CF becomes set if the unsigned scale range is exceeded.

### 2) PF - Parity Flag:

Usually parity is an error detection procedure. This counts no. of ones in result. If the no. of one's count in result is even the  $PF=1$ , otherwise  $PF=0$ ; odd parity,  $PF=1$ ; even parity.

### 3) AF - Auxiliary Flag:

This carry flag is used in BCD arithmetic operation. Holds the carry (half-carry) after addition or borrow after subtraction. Usually if a carry generates from bit position 3 towards bit position 4 of the result, then  $AF=1$ , otherwise 0.

(91)

(91)

4) ZF - Zero Flag: It is a 16-bit register.

Shows the result of the arithmetic or logic operation. If the result becomes zero, then  $ZF=1$ ; otherwise  $ZF=0$ .

Has four variants forms of previous question

5) SF - Sign Flag:

The flag is used in signed arithmetic operation. This holds the sign of the result after an arithmetic logic instruction execution.  $SF=1$ ; for negative result;  $SF=0$ ; for positive result, while the MSB (Most Significant Bit) of the result

indicating all the sign bits is 0 indicating no possible overflow (unless溢出) a溢出溢出

Overflow or underflow will happen if

Message with 32-bit length + 32-bit

• 0 segments,  $L = FA$

(92)

(93)

6) OF - overflow flag : (for add & sub op.)

Overflow occurs when signed numbers are added or subtracted. This is used by signed arithmetic operations.

An overflow indicates the result has exceeded the capacity of sign range. In expression

$$OF = Cout \oplus Cout - 1$$

for  $x + y = z$  if  $x, y, z$  are binary

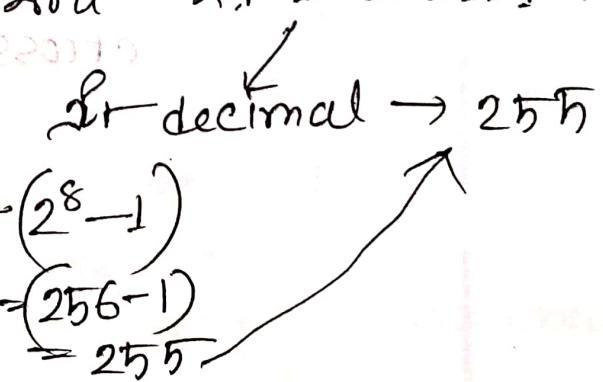
# CF  $\rightarrow$  unsigned numbers ~~will~~ all positive.

suppose: 8 bit positive numbers scale  
(unsigned).

Binary minimum combination is

always all zero and  $00000000$

maximum will  $11111111$ .



range  $2^8$

$$0 - (2^8 - 1)$$
$$\rightarrow (256 - 1)$$
$$= 255$$

(43)

16 bit 數字範圍： $(0 \sim (2^{16} - 1))$

reference sample number (0-65, 536-1)

$$\text{Point } P(65, 535) \text{ is located in the second quadrant.}$$

ପ୍ରମାଣ,

प्रा तो पुरुष २५५ - २५२ बाबुलोनीज विद्या

Orange ৩৮ (০-২৫৫), range br  
orange ওজ ২৫৭ গ্র CF ২(৫)

## Binary वृत्ति अंक (२)

$$\begin{array}{r}
 \text{Binary 2's Comp} \\
 \hline
 \text{BCD} \quad 0111 \ 1111 \ 1111 - 8610 \\
 \text{BCD} \quad 0110 \ 1001 \ 0010 - 8610 \\
 \hline
 \text{BCD} \quad 1000 \ 0001
 \end{array}$$

Si existen más convicciones, <sup>o</sup> bien

result fit 19 bit, 2 or 8 bit (20 bits)

CROSS - ফেরিটি ৭২ CF $\phi$  = 1 ২/৩

15881588-13

carry.

卷之三

1920-1921

(44) (iii)

ବେଳାକୁ 255, 255 ରୁ ପରିମାଣ କରିବାକୁ ପାଇଁ

$$253 + 2 = 255$$

$111111101 \rightarrow$  ନିର୍ଦ୍ଦିତ ଫର୍ମ୍‌ଯୁକ୍ତି

$111111111$

$$\begin{array}{r} 128 \\ 64 \\ 32 \\ 16 \\ 8 \\ 4 \\ 2 \\ 1 \end{array}$$

ଅଗ୍ରମ ଫର୍ମ୍‌ଯୁକ୍ତି 255

Digit 2

0000010

$11111101$

$00000010$

$$\begin{array}{r} 11111101 \\ 00000010 \\ \hline 11111111 \end{array}$$

$\rightarrow 255$

କିମ୍ବା Range କିମ୍ବା କ୍ଷେତ୍ରରେ ୧୨ ( $CF=0$ ) 255

carry flag.

$\Rightarrow CF \rightarrow ADD / SUB \rightarrow Adder / Subtractor$ .

HS

$00000101 \rightarrow 5$

(-)  $00000010 \rightarrow 2$

$$\begin{array}{r} 00000010 \\ - 00000010 \\ \hline 00000010 \end{array}$$

ଅଗ୍ରମ ଫର୍ମ୍‌ଯୁକ୍ତି

୩

Sub ଫର୍ମ୍‌ଯୁକ୍ତି  
(-)

$$\begin{array}{r} 1 \\ 0 \\ - 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ 0 \\ - 1 \\ \hline 0 \end{array}$$

ବେଳାକୁ ମାନ୍ୟ

ଚାହୁଁ ଦ୍ୱାରା ୧ ବାବୁ କିମ୍ବା ୨ ମାନ୍ୟ ମାନ୍ୟ

ଏ କାହାରେ କିମ୍ବା କିମ୍ବା କାହାରେ ୧

ମୋଟାକୁ 0 ହେଉ ଲୋକାର୍ଥୀ କେତେ ଏହା ବେଳାକୁ ମାନ୍ୟ

ଅଗ୍ରମ ଫର୍ମ୍‌ଯୁକ୍ତି

(45) (45)

• 2 (ମାତ୍ର) 1-ବାଦ ଦିଏ । ଲିଖିବା 2 (୨୫ ଟଙ୍କା) ଦେଖିବା  
ପାଇଁ ସହିତ 2୦୨୯ ଜାନ୍ମୀ ବିଷୟରେ ।

পরব্রহ্ম ধারা ২০১৫ বর্ষাপে ১৫/১০/১৫ ১০' ৭৩৮  
০। ১ ০ (২-১)

THE PAPER WAS READ IN

then  $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  onto

$$\begin{array}{r}
 1000000 \\
 \text{मुद्रा रूपान} \\
 \hline
 25 - 1-\text{प्रियंका} \\
 \hline
 \end{array}$$

0 0 0 0 1 0 1 0 0 0 0 0 0  
1 0 0 0 0 0 1 0 1 1 1 1 1

Last bit =  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$  for 2nd column (Step 1)

• ৰাষ্ট্ৰ নিৰ্মাণ পথ (১) ১৯৭৫

( $\text{CF} = 1$ ) always 1 ( $\text{NEGR CF already}$   
 not on initial bubble +  $\text{CF} = 1$ ) ( $\text{CF} = \frac{1}{1}$ )

અમારી કાંઈ જો હોય તો આપણે એવી વિધિ

(46)

10

## Binary

$$0 - .0V$$

your friend

PF 1:39 ~~negative~~ ~~positive~~ ~~negative~~ ~~positive~~

parity method 21 अप्रैल

Dodd

11) even

odd ~~is~~  $\oplus$  one count = odd  
even  $\oplus$  one " = even.

parity method 2 msb<sup>0</sup> bit for depend  $\leftarrow$

odd seven purity use soft off

msb ← 0 0 0 0 1 1 0 1  
first bit

soft odd parity use soft

~~one with even method use ans~~

1 0101010  
22 bit count

$$\begin{array}{r} 0 \mid 0 \mid 0 \mid 0 \\ | \quad | \quad | \quad | \\ 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1 \end{array} \rightarrow 32 + 16 + 8 + 2 = 58 \text{ sum} \\ \text{even} \\ PF = 1 \quad 127 \text{ sum} \quad 57 \text{ 27} \\ PF = 0$$

$$F=1 \quad | 27\text{मि} \quad 57\text{ग्र} \\ PF=0$$

(47)

- $PF = 0$   $\rightarrow$  odd parity.
- $\rightarrow$  count no. of 1's
  - even  $\rightarrow$  odd  $\rightarrow PF = 1$
- finding the bad bit position

bbs (i)

more (ii)

A.F.:

$BED \xrightarrow{bbs} 0 \rightarrow$  odd  $\rightarrow$   $BED \xrightarrow{more}$

0 - 0000 bit

0 = 1001 bit

bbs & more  $\rightarrow$  finding a bad bit position

$5+9 \rightarrow 0110 \rightarrow 9$  bbs

1001 - 5

1011  $\rightarrow$  1100  $\rightarrow$  1010

is fault.

01010 } 9 bbs  $\rightarrow$  RMA Code Generation

6 more  $\rightarrow$  25 1

10011  $\rightarrow$  finding bad bit positions  $\rightarrow$  10011

10011  $\rightarrow$  101010 (P.I.O.)

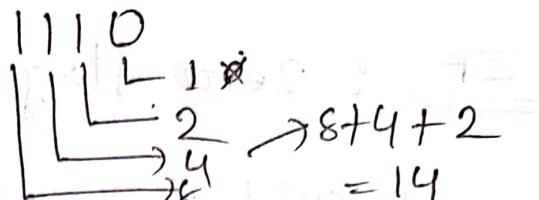
10011  $\rightarrow$  finding bad bit positions  $\rightarrow$  10011

$25 = 1 + 3 + 5 + 7 + 9 = 1 + 01010$

$more = 9 \rightarrow 1 + 01010$

(48)

$$9+5 = 14 \rightarrow$$



$$L = 7.5 \text{ ms} \quad T_{\text{off}} = 7.5 \text{ ms} \quad (0) \text{ min} + 14 \text{ ms} = 14$$

$$\begin{array}{r} 1110 \rightarrow 14 \\ 0110 \rightarrow 6 \\ \hline 10000 \rightarrow 20 \end{array} \quad \text{the same flow as } 1$$

$\rightarrow$  Binary  $\rightarrow$  Bit 3 - Bit 4  $\Rightarrow$  2<sup>3</sup>  
 $\rightarrow$   $\frac{1}{2}$   $\rightarrow$  carry 2<sup>3</sup> 1

$$\begin{array}{r} \text{d} - \text{d} + \text{d} \\ \text{d} - \text{d} + \text{d} \\ \hline \text{d} \end{array} \quad \begin{array}{r} 10010 \\ 0000 \\ \hline 0110 \end{array} \quad \begin{array}{l} \text{Bit 3} \\ \text{Bit 4} \end{array}$$

$$\begin{array}{r} \text{d} - \text{d} + \text{d} \\ \text{d} - \text{d} + \text{d} \\ \hline \text{d} \end{array} \quad \begin{array}{r} 10100000 \\ 21011111 \\ \hline 11011111 \end{array} \quad \begin{array}{l} \text{Bit 3} \text{ is } 0 \text{ (AF=0)} \\ \text{Bit 4} \text{ is } 1 \text{ (AF=1)} \\ \text{Bit 3} - \text{Bit 4} \\ \text{d} - \text{d} + \text{d} \end{array}$$

$$\begin{array}{r} \text{d} - \text{d} + \text{d} \\ \text{d} - \text{d} + \text{d} \\ \hline \text{d} \end{array} \quad \begin{array}{r} 1010 \\ 0001 \\ \hline 1010 \end{array} \quad \begin{array}{l} \text{d} - \text{d} + \text{d} \\ \text{d} - \text{d} + \text{d} \\ \hline \text{d} \end{array}$$

ZF: zero flag

$$\text{result} + \text{sum}(0) \leftarrow 92(n) - ZF_1$$

$$\text{result } 2\bar{m} = 1 \quad 2\bar{m} - \frac{1}{2} = \frac{1}{2} \quad 2F = 0$$

ମେଲାନ

5-5 →

08-1963

K R P file = e file → 0000 0101

$$1 \text{ टक्के } \frac{2}{5} \text{ रुपये } 20\text{रु} (5-5) \text{ रुपये}$$

$$\text{গুরু (-5)} \quad +5 -5 \quad \text{গুরু 2 মুক্ত}$$

~~মন এবং ক্রেতে ইন্ডিপেন্ডেন্সের রেভেন্যু~~

reverse

00000101 → 5

11111010 → 115

+ 1 → 2's

11111111 → -1

# O Grande Verso

1 23 November

3

- Then 275

## compliment

ମାନ୍ୟ ୧୯

115 ③ 215 ମହିଳାଙ୍କ ପ୍ରତି ଏକ ଲକ୍ଷୀ

279-270 Then 1900 1010

compliment—

1 + 28 28

ମୋହନ୍ତି ୨୦

~~life~~ at ~~work~~ + money P.T.O

$\theta = 75^\circ$

(50)

(14)

प्राप्ति का अनुदान निम्नलिखित है।  
 $+5 \rightarrow 0000\ 0101$

$-5 \rightarrow \begin{array}{r} 1111\ 1011 \\ 0000\ 0000 \\ \hline \end{array}$  अवधारणा।  
 +5 का अनुदान  
 1111 1011  
 0000 0000  
 +5 का अनुदान

उत्तर CF=1

ZF=1  $\rightarrow$  8-bit form में result

अन्य 9 bit  $\rightarrow$  00000000 form 228 bit  
 $(32\text{bit}) \rightarrow 00000001$  मिले  $\rightarrow$  228 bit

$(32\text{bit}) \rightarrow 11111111$  का 1 वाला 0 था ZF=1.

AF=1, 8-bit 3-Bit 4 जो 1 के लिए

PF=1  $\rightarrow$  उत्तर इसका 0 भी 1-201 0 even number 1

बहुत से 8-bit 0, 1

SF:

unsigned  $\rightarrow$  00000000  $\rightarrow$  mini  $\rightarrow$  0

11111111  $\rightarrow$  255  $\rightarrow$  maximum

$(0 - 255)$   $(0 - (2^8 - 1))$

signed: msb (sign bit)  $\rightarrow$

msb = 0  $\rightarrow$  +ve number

msb = 1  $\rightarrow$  -ve "

negative number 20125 or  $(216)^{-1}$   
fast 00000000000000000000000000000000

msb=0 +ve SF=0

00000000000000000000000000000000

msb 2T SF 0-275 1

127 7 bits

msb=1, SF=1

fl 00000000000000000000000000000000

min 10000000000000000000000000000000

max 01111111111111111111111111111111

$\in (2^7)$  to  $+ (2^7 - 1)$

$+ (2^7 - 1)$  to  $+ (2^7 - 1)$  fl 0-275 1 = 7A

128 0 127 + 127 = 255 + 127 = 382

1 27 bits

OF: Sign overflow:

01111111111111111111111111111111

sign scale

0 + 00000000000000000000000000000000

00000000000000000000000000000000

00000000000000000000000000000000

+127 to -128

(+127) + 0 (+127 - 0) = -

127  $\rightarrow$  (01111111111111111111111111111111) 2am

00000000000000000000000000000000

00000000000000000000000000000000  $\rightarrow$  -128

0  $\rightarrow$  OF = 1 down

(52)

$$\begin{array}{r} 0111111 \rightarrow 127 \\ 0000011 \rightarrow 3 \\ \hline 10000010 \end{array} \quad \text{Rightmost Entry Count}$$

$\frac{1}{0} \Rightarrow 1$

$OF = 1$

#  $Cout \oplus Cout - 1 = OF$

00	$\rightarrow 0$
01	$\rightarrow 1$
10	$= 1$
11	$= 0$