

microprocessors

calculator $\xrightarrow{\text{for}}$
 4004, 8008, 8080, 8085 8086 - 16 bit first 16-31 bit
 only for specific application

80286

80386

80486

4004 → calculator

8008 → traffic controller

8080 → temp controller

8085 → temp (d)

8086 → CPU

80286

80386

80486

8088 → pentium - pro

→ P-II

→ P-III

→ P-IV

→ P-V

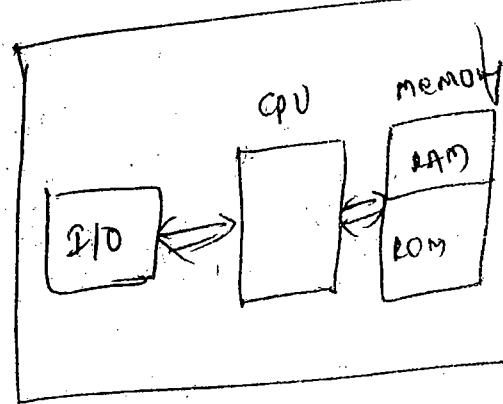
→ P-6

→ P-7

→ P-8

→ P-9

→ P-10



8 bit ALU

→ width of 8 bit ALU.

most of 16 internal registers available as 8-bit.

Temporary storage of data (d) variable @ constant
during execution.

Fetch Decode Execute.

↓
Read data from memory

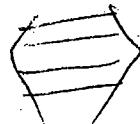
The width of ALU → Arithmetic logic unit.

→ Arithmetic, logic & bit manipulation operations
on bits (or nibbles or) bytes

No bit manipulation operations : RRC, RAR, REC, AAL.

It has 8 data lines

D0
:
D7



parallel and bidirectional.

direction from either CPU to memory (d),
memory to CPU

(a) either CPU to MP (d), CPU to I/O devices

$$2^8 = 256 - 16$$

- I/O devices

IN

Some

eq

8085

data

0

9

OUT

→ trans

data for

16

28

M0V

M0V

GT

③

It has

AC

AX

AF

AK

IP - 00H

OP - 00

- FFH

FFH

8-bit

(a) Constant

IN \rightarrow transfer 16 data from UP to MP.

Source is 8-bit port Address

eg IN 40H→ 8085 is accumulator based UC.
format — ASCII

0	— 30	A — 41	A — 61
9	— 39	Z — 5A	Z — 7A

PC unit

I/O Operations

RAR, RLC, AAL.

OUT d

→ transfer data from MP to I/O device.

Data Port→ It is NO. of instruction based on data lines
 $2^8 = 256$ — opcodes (01) instructions are possible.

at

memory (d)

MOV A, B

MOV B, A

→ It can read (d) write 8-bit data at a time.

MP to UP devices

(3) It has 16 Address linesA0 } Lower order Address lines
A15 }A8 } Higher order Address lines
A15 }A0  parallel and unidirectional

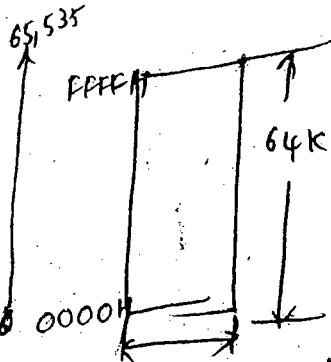
~~always~~
processor from MP to memory.

$$\text{memory} = 2^{16} = 2^6 \cdot 2^{10}$$

$$\rightarrow 65536$$

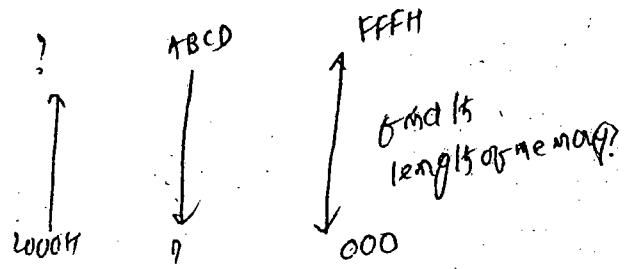
$$= 64 \times 1024$$

$$= 64 \text{ KB}$$



64K = length
of memory

+ words of memory indicates 1 byte \rightarrow width of memory
i.e. at once it can read or write ~~any~~ 1 byte data.



, 64KB \rightarrow Physical memory

64KB indicate the size of the RAM and ROM.

why up based system?

Binary

4

2

1

0 1 1

1

$2^0 = 1$

Decimal

11

8

6

0-9

5

4

$2^4 = 16$

0000

Octal

Hexa

11

16

14

0-9 & A-F

14

16

$2^4 = 16$

0000 = 0

!

1001 - 9

1010 - A

1011 - B

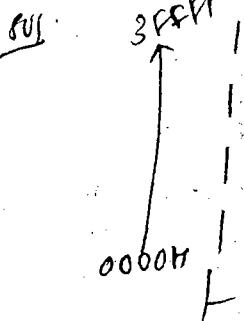
!

1111 - F

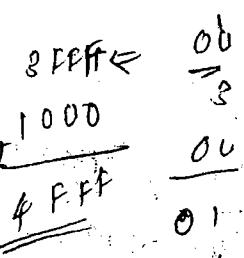
→ Hex

Q. 16

1000 H

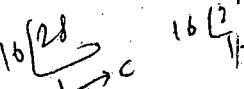


Q. 16



Q. 16

ABCD H



1000 H

ABCD H

BBCC

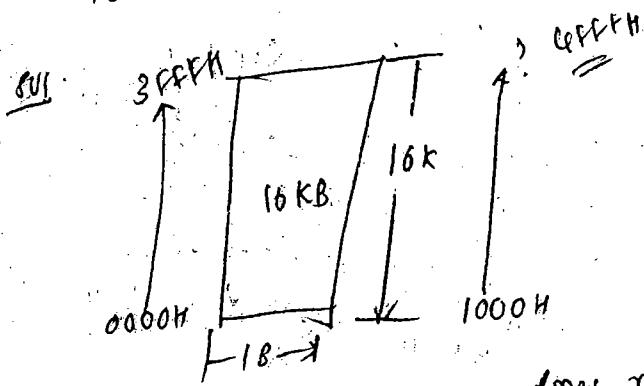
1111 H

1111 H

NewtonDesk.com

T
 $54K = \text{length}$
 of memory
word of memory
1 byte

→ Hexa decimal utilized all the no. so.
 Q: At starting memory location of 16KB memory at
 $1000H$ what is last memory location?



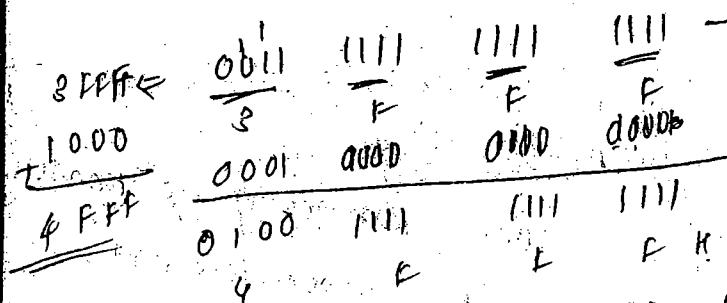
S1: No. of Address lines require

$$= 16K \times 8$$

$$= 2^4 \times 2^{10} \times 8$$

$= 2^{14} \rightarrow$ Address lines

$\rightarrow 14$ Address lines



used in it

Q: On 4KB of External RAM at starting memory loca

ABCDH what is last memory location

No. of Address lines = 4×8

$$= 2^4 \times 2^{10} \times 8$$

$$= 2^{12} \rightarrow 12 = \text{Address lines}$$

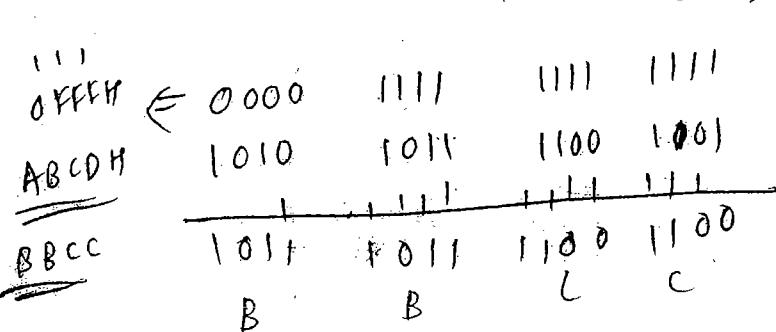
$0000 \quad 0001 \quad 0010 \quad 0011$
 $0 + 4 + 4 + 4 = 12$

A-F



H
 $00 = 0$

11 - 9
 10 - A
 1 - B



..... for memory locations

2. At 8KB of external RAM it last memory location

PS ABCD H what ps/tk starting memory location.

Address (mes) = 2 kB

2010x8

$\approx 2 \times 10^8$

- 11 x 8

211 Address 1 fm 4

$$\begin{array}{cccc}
 0110 & 1011 & 1100 & 1101 \\
 0000 & 0111 & 1111 & 1111 \\
 \hline
 1010 & 0011 & 1100 & 1110 \\
 & 3 & & \\
 \end{array}
 \Rightarrow \text{ABCEH} = 02FFH$$

16 16

OFFICE

$$\begin{array}{r}
 16 \\
 13 - D \\
 \hline
 29 \\
 -15 - F \\
 \hline
 14 - B \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 16 \\
 11 - B \\
 \hline
 9 \\
 -15 - F \\
 \hline
 12 - C
 \end{array}$$

14-B
Q. Qn 1KB of external RAM. last memory location is
Ans 1000?

13FFH last what is its starting memory location?

Address times = $1K \times B$

$$= 2^0 \cdot 2^{10} \times 8$$

2 10 Address

0001 0011 1111 1111 \Rightarrow 13FF

0000 0011 1111 1111 → 03FF

0001 0000 0000 0000 1000H

1 0 0 0

$$0+2+9+4=15$$

⑤ clock

16

off location

on.



$$\begin{array}{l} 01111111 \\ + 3 + 4 + 4 = 11 \end{array}$$

Q. If 16 starting memory location 8000H last memory location 9FFFH what is the lengths of the memory

- a) 8000 b) 1191 c) 8192 d) none of the above.

8000H

9FFFH

8000H

1FFFH

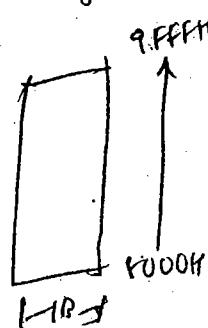
1444

→ 13 → Address 11 bits

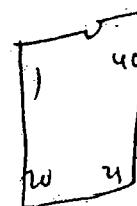
→ $2^3 \times 2^{10}$

→ 8×1024

→ 8192



Q. It is a standard 40-pin IC
It is available dual in-line package



location is
cation?

+5V DC single
There are 3-type of 3 IC packages

- y Dual in-line
- y metal can
- y flat top.

Q. clock freq $f_c = f_0$ oscillating freq (i) crystal freq
(ii) clock freq dividers dividing two

$$f_c = 3.07 \text{ MHz} \rightarrow \text{Theoretically}$$

$$= 3 \text{ MHz} \Rightarrow \text{practically}$$

$$> 2 \text{ MHz} \Rightarrow \text{operating freq} \Rightarrow 0.5 \text{ MHz}$$

$$f_0 = 6.14 \rightarrow T$$

Q. If 16 oscillating freq of 8085 MP is 5 MHz what is

16 clock time.

- a) 0.5 μs b) 0.25 μs c) 0.45 μs d) none of the above

$f_c = \frac{10}{2} = 2.5 \text{ MHz}$

$$\text{clock time} = \frac{1}{f_c} = \frac{1}{2.5 \times 10^6} = 0.4 \text{ μs}$$

⑧ 8086 \Rightarrow 40 pin IC

20 \rightarrow add lines $\left\{ \begin{array}{l} \underline{\text{86}} \\ \text{86} \end{array} \right.$ Vcc = 1

16 \rightarrow data lines $\left\{ \begin{array}{l} \underline{\text{86}} \\ \text{86} \end{array} \right.$ GND = 1

CLK = 1

INT = 1

$$\frac{1}{4 + 36} = \underline{0.025}$$

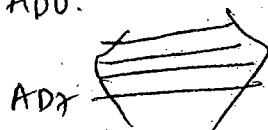
⑨ Lower order add and data lines are multiplexed

because to reduce its pins and increase its application.

\rightarrow GND \rightarrow provide return path to its power supply.

\rightarrow idle
control
device

ADD.



ALE \rightarrow Address latch enable

8085 take min 4-clock cycles

max 18-clock cycle

for execution of instructions.

- ⑩
y. Add
y. Data
y. Control

No. of pins

→ It is basically serial to parallel to converter.

→ It acts as parallel communication.

8257: DMA Controller (Direct memory access)

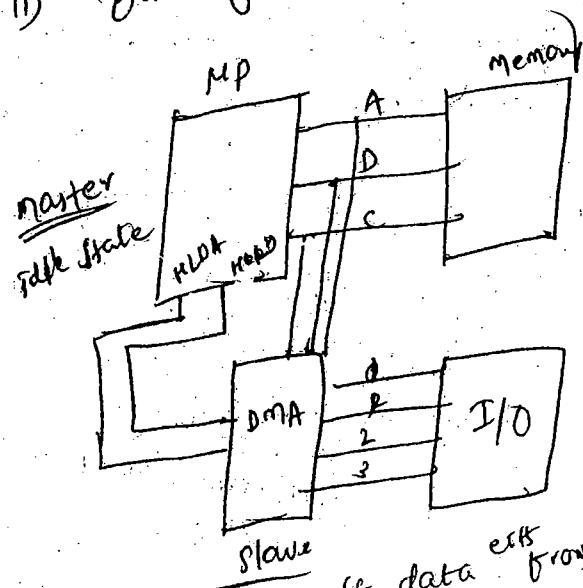
MOV, M, M

i) memory to memory data transfer is not possible
in MP.

ii) for large data transmission MP is not suitable.

are various

p. to p. devices



⇒ DMA transfer if data exits from memory to I/O device

(a) I/O Devices to memory without using MP.

⇒ Hold → request from DMA to MP.

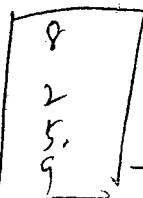
⇒ HLDA → Acknowledgement from MP to DMA

, through

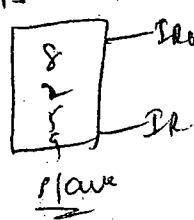
8259: Programmable interrupt controller (PIC).

→ Provide

multiple
master



interrupts and control functions.



using and

Programming

model 8085 P.P.

Note

W.	Z
A	F
B	C
D	E
H	L
SP	
PC	

A is by default 8-bit Accumulator

NOTE: 8085 is a accumulator based μ.p.

③ Flag

→ Flag

the



ADD S

Destination is by default Accumulator.

$$d = d + s$$

$$A = A + s$$

→ 8086 is a "Register based" μ.p.

Add DS

Add BL, CL

$$BL = BL + CL$$

→ accumulator based mean after arithmetic operation it result stored in the accumulator.

purpose of Registers:-

General purpose registers

Six - 8-bit General purpose registers

B, C, D, E, H, L

These are used to temporary storage of

data or variable or constant

(a) Three - 16-bit G.P.R

BC

DE

HL

8085 → 5

8086 → 51

i) Call

y pair.

j) Aux

y zero

g) sign 0

8086

j) carry

from

Accumulator

Accumulator based

Note: By default HL is the memory pointer

③ Flag register

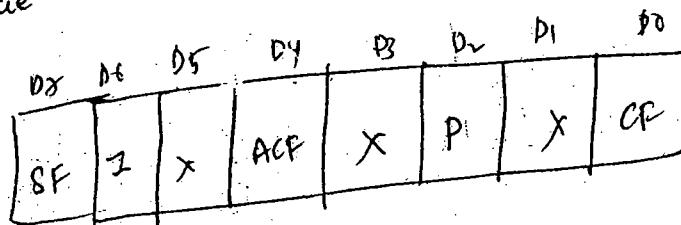
Accumulator + Flag \Rightarrow program status word (PSW)

(or) processor status word

\rightarrow Flag register is always indicates the current status

Accumulator

See ALU.



8085 \rightarrow 5 flags

8086 \rightarrow 5+4 \Rightarrow 9 flags

ACTION 16

1) Carry flag

2) parity flag

3) Auxiliary carry flag

4) Zero flag

5) sign flag

8086 \rightarrow Status flag + control flag.

8086 \rightarrow Status flag + control flag. said to be there is a carry

6) Carry flag: carry flag from just of MSB.

overflow
trap
direction
interrupts
clock

$$A = 89 \Rightarrow 1000\ 1000$$

$$B = 94 \Rightarrow 1001\ 0100$$

$$\begin{array}{r} 1000\ 1000 \\ + 1001\ 0100 \\ \hline 1001\ 1101 \end{array}$$

$$\text{ADD } B \Rightarrow CP = 1.$$

If the result is too large to fit in the destination

e.g. of set 1.

$D_A = 1 \quad D_B = 0$	$D_A = 0 \quad D_B = 1$	$D_A = D_B = 1$	$D_A = D_B = 0$
$CF = 1$	$CF = 0$	$CF = 1$	$CF = 0$
$OF = 1$	$OF = 1$	$OF = 0$	$OF = 0$

$$\boxed{OF = D_A \oplus D_B}$$

$$CF = D_A$$

Q: after

what

g) 02H

0000 0090

PP=0

(ii) Axialaf

→ There

ACF

A = 08H

B = 49H

1. Parity Flag:

Parity flag is said to 1 if result contains even no. of 1's otherwise reset to 0.

no. of 1's in micro controller this process is Reverse.

Q5

In the micro controller this process is Reverse.

A = 48H → 0100 1000

B = 24H → 0010 0100

$\underline{\underline{0110 \ 1100}}$

PP=1

A = 98H

B = 65H

SR = -55 → 0110

2) 65

T

(i) subtraction

$$A = 80H \quad 1000\ 0000$$

$$B = 80H \quad 1000\ 0000$$

$$\underline{1000\ 0000}$$

even or even = 0.

$$OF = 0$$

$$CF = 0$$

$$OF = 0$$

$$CF = 1 \quad PF = 1$$

Q: after execution of arithmetic operation PF is set to
what is content in the accumulator?

- a) 02H b) AAH c) 70H d) none

$$0000\ 0000 \quad 1010\ 1010 \quad 0111\ 0000$$

$$PF = 0 \quad CF = 1 \quad OF = 0$$

(ii) Auxiliary carry flag :-

→ There is a carry from lower nibble to higher nibble otherwise reset to 0.
ACF is set to 1

$$A = 08H \rightarrow 0000\ 1000$$

$$B = 49H \rightarrow \underline{0100\ 1001} \\ 0101\ 0001$$

$$ACF = 1$$

$$SUB B$$

(i) borrow from higher nibble to lower nibble

$$A = 45H \rightarrow \underline{0100\ 0101} \\ B = 28H \rightarrow \underline{1001\ 0000} \quad 1101$$

5. Reverse

$$A = 984 \rightarrow 1001\ 1000$$

$$B = 654$$

$$\begin{array}{r} 1001\ 1000 \\ 1001\ 0111 \\ \hline 1001\ 1011 \end{array}$$

$$CF = 1 \\ ACF = 1$$

$$\begin{array}{r} 1001\ 1000 \\ 0110\ 0101 \\ \hline 0011\ 0011 \end{array}$$

$$ACF = 0$$

(ii) complement

$$\begin{array}{r} 1001\ 1000 \\ 0110\ 0101 \\ \hline 0011\ 0011 \end{array}$$

$$CF = 0 \quad ACF = 0$$

take complement

$$CF = 0 \\ ACF = 0$$

Find 2's complement of +18 & -18.

$$\begin{array}{r} \text{+18} \\ \text{-18} \\ \hline \end{array} \rightarrow \begin{array}{l} 0001\ 0111 \\ 1000 \\ \hline 01111 \end{array} \left\{ \begin{array}{l} \text{2's comp.} \\ \text{X} \end{array} \right.$$

(b) +ve no then 95 is no complement. that is same as 1's
complement no.

2's complement range

$$\begin{array}{r} 2^{n-1} \text{ to } -2^{n-1} \\ \hline \end{array} \quad \begin{array}{l} n=6 \\ =2^6-1 = 63 \\ =32-1 \\ =31 \end{array}$$

$$\begin{array}{r} 100 \\ \hline 2^{n-1}-1 \\ 2^{n-1} \\ \hline 15 \\ \hline \end{array} \quad \begin{array}{l} +18 \rightarrow 0\ 1000 \\ -18 \rightarrow 1\ 01111 \end{array} \left\{ \begin{array}{l} \text{✓} \end{array} \right.$$

v) 8192

A

B

SUB

A =

2nd Method

A = 4

B = 55

55 → 0101

$\overline{2^6} 55 \Rightarrow 1011$

in MP

$$+18 \rightarrow 00010001$$

$$-18 \rightarrow 11101111$$

v) Zero Flag: zero flag is set to 1 if result is 0 otherwise

→ ACF = 1

Cases: A,

add +

A = 10

B = 4

5B

MJ

$$\begin{array}{r} A = 80H \rightarrow 1000\ 0000 \\ B = 80H \rightarrow 1000\ 0000 \\ \hline 0000\ 0000 \end{array}$$

CF = 1

ZP = 0

=

v) Sign flag: If result of MSB is 1, its sign bit
is set '1' otherwise reset to '0'.

(same as 1st)

$$\begin{array}{l} A = 49H \rightarrow 0100\ 1001 \\ B = 55H \rightarrow \underline{\underline{0\ 101}}\ \underline{\underline{0\ 101}} \\ \text{SUB B} \\ A = A - B \end{array}$$

$\overline{0\ 111}\ \underline{\underline{0\ 100}}$

SF = 1
CF = 1

ACF = 0
PF = 0
ZF = 0.

 $I = 159$ $= 821$ $= 31$ $=$ \checkmark 2nd Method

$$\begin{array}{l} A = 49H \rightarrow 0100\ 1001 \\ B = 55H \rightarrow \cancel{0\ 101}\ \cancel{0\ 101} \\ \cancel{2\ 1\ 000} \quad \cancel{1\ 010}\ \cancel{1\ 011} \\ 55 \rightarrow 0101\ 0101 \quad \cancel{1\ 11}\ \underline{\underline{0\ 100}} \\ \cancel{2\ 1\ 55} \rightarrow 1010\ 1011 \end{array}$$

SF = 1 → 1
CF = 0 [carry] → 0
ACF = 1 → 0
PF = 0 → 0
ZF = 0 → 0

if SF = 01 otherwise

→ ACF is not available to user, it is for BCD add
Case 2: If the result of lower nibble is invalid BCD then
 add +6 to lower nibble. Convert to

$$\begin{array}{l} A = 19H \rightarrow 0001\ 1000 \\ B = 42H \rightarrow \underline{\underline{0\ 100}}\ \underline{\underline{0\ 010}} \\ \underline{\underline{5\ 8H}} \quad \underline{\underline{0\ 101\ 1011}} \\ \text{FMU} \quad \underline{\underline{5\ B}} \quad \rightarrow \text{invalid BCD} \\ \underline{\underline{0\ 110}} \\ \underline{\underline{0\ 110\ 000}} \quad \rightarrow \text{BCD} \end{array}$$

11/12/10

~~case i)~~ If result of lower nibble is invalid BCD no.
but ACf is set 1.

$$\begin{array}{r}
 A = 88H \rightarrow 1000 \quad 1000 \\
 B = 09H \rightarrow 0000 \quad 1001 \\
 \hline
 91H \quad 10010001 \\
 \hline
 ACf = 1 \quad 0110 \\
 \hline
 98H \quad = 10010111
 \end{array}$$

case ii) If result of higher nibble is invalid BCD no.
add 6 to higher nibble.

$$\begin{array}{r}
 A = 91H \rightarrow 1001 \quad 0001 \\
 B = 24H \rightarrow 0010 \quad 0100 \\
 \hline
 B5H \quad = 10110101 \\
 \hline
 0110 \\
 \hline
 05H \quad = 00010101
 \end{array}$$

case iii) If result of higher nibble is valid BCD.
But CF is set 1. Add +6 to higher nibbles.

$$\begin{array}{r}
 A = 88H \rightarrow 1000 \quad 1000 \\
 B = 90H \rightarrow 1001 \quad 0000 \\
 \hline
 18H \quad 0001 \quad 1000 \\
 \hline
 0110 \\
 \hline
 0111 \quad 1000
 \end{array}$$

push S →

Source 3)

BC →

DE →

HL →

PSW →

→ parallel
and higher

Jr

1. BCD no.

11112110Stack

i) Stack is outside IC part
and part of RAM.

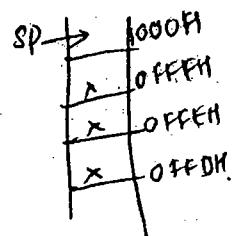
BCD no.

Stack pointer

i) If it is a 16-bit register &
it is inside IC MP.

ii) It always points to indicate the
top of the stack.

iii) SP either incremented (or)
decremented (as depends upon
IC read and write operation).



push S → write data into stack.

BC, DE, HL (or), PSW.

id BCD.

b61e,

BC - 1234H

SP = SP - 1

DE - 5678H

SP = SP - 1

HL - 9834H

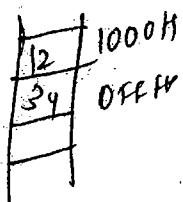
push

PSW → ABEDH

lower byte to lower memory location

→ transferred of data always
and higher byte higher location

push B



SP = SP - 1

SP = SP - 1

SP = SP - 1
operation

if SP is decremented

⇒ Un IC part

if S

PUSH D

$$D = 5678H$$

12
34
56 0FFE8H
91 0FF0X
SP X 0FFC9H
X

The fn of push instruction is write its data into memory during this operation SP is decremented.

Q2:

first \rightarrow decrement SP
and write data (predecrement)

\rightarrow read data from memory.

d

DP (Register pair)

$$BC = 1234H$$

$$DE = 5678H$$

$$HL = 9123H$$

$$PSW = ABC0FF$$

> POP B

SP = SP + 1 and read data from

$$SP = SP + 1$$

$$B = 9134$$

fn of pop instruction is read data SP \rightarrow

from memory during this operation SP is prem

post incremented SP

12
34
56
28
91
34

$$SP = SP - 1$$

and write

$$SP = SP - 1$$

and write

$$SP = OFFE$$

Q. ff fe sp

instruction

i) what is 16

ii) where is 16

$$SP = 2000H$$

$$PSW = ABC$$

$$SP =$$

let con

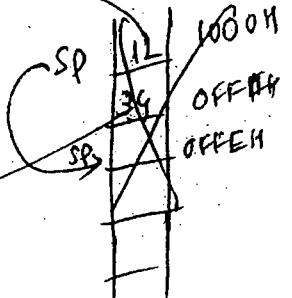
→ first read data from memory and sp increases

off 2

Q. If SP is initialized with 1000 memory location and push B instruction is executed its content bc register pair is data into 1234H. While 98H what is the content SP. Where is it decremented?

1234H available

~~best 1234H~~



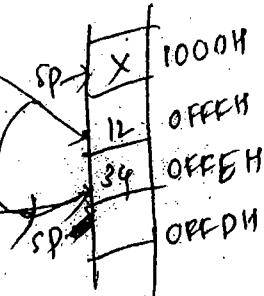
on.

push B

SP = SP + 1
and write

SP = SP + 1
and write

SP = OFFEH

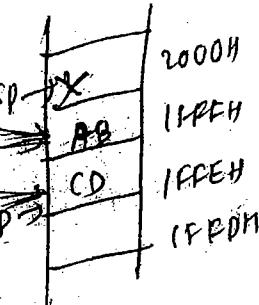


content available from OFFEH & OFFEH
Q. If SP initialized with 2000 memory location and push PSW
instruction is executed the content in the PSW is ABCDH.

i) what is 16 content in ESP
ii) where is 16 data available

SP = 2000H

PSW = ABCDH



SP = 1FFEH

(the content available PD)

1FFFH, & 1FFEH

SP is prem

Program Counter (PC)

→ If it is 16-bit register it always holds the address
of next instruction to be fetched.

→ 8085 can hold instruction address

min -1 byte
max 3-byte instructions (length)

PC: $\frac{4000H}{4001H}$ MVI A, 45H - 2

PC = 4002H. BMOV B, A - 1

PC = 4003H ADD B - 1

PC = 4004H HLT - 1

→ PC = 4005.
→ In 8085 PC & IP are same.

Q. After the execution of the following program what is the

content in the PC.

PC = 2000 MVI C, 40H - 2

2001 ← DCR C - 1

2002 ← JNZ X - 3

2003 ← 2004 ← RET - 1

2005 ← 2006 ← RET - 1

PC = 2007.

Q. After the execution of the following program what is the content

in the PC.

PC = 2000H MVI A, 29H - 2

2001 ← MOV B, A - 1

2002 ← ADD B - 1

2003 ← CALL 4000H - 3

2004 ← HLT - 1

8. (i) Call

(ii) C

JNT

and

selvis

16

(iii) Call

PC

the address

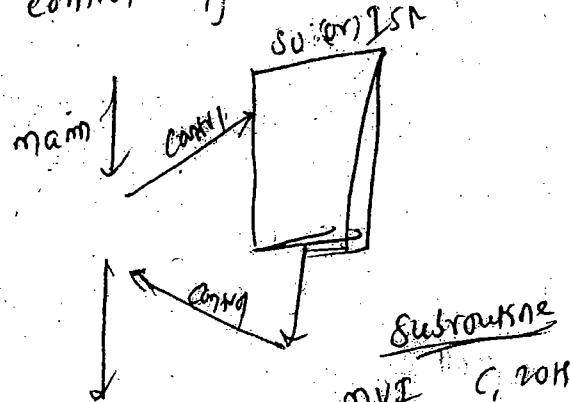
~~PC + 0000~~
 $2004 \rightarrow 0000$
 $2005 \rightarrow 00$ } 3 bytes.
 $2006 \rightarrow 40$

(i) Call is a 3-byte instruction

(ii) Call is a type of interrupt.

→ Interrupt or breaks the normal sequence of the execution and control transfer to either subroutine or interrupt service routine after the execution of subroutines or after the control again refers to the main program.

What is the



(the content

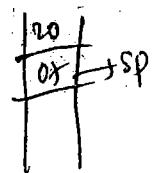
4000
4001
4002 ← DCR C
4003 ← JN2 X
4004
4005
4006 RET
PC = 4007

equivalent

to push instruction.

(iii) Call is

w	2
40	00



PC is loaded into SP and

→ RET is equivalent to the pop instruction

→ End of sub routine

- Q) After execution of the following program what is the content in the accumulator.

1000 | MVI A, 1000H
main | call sub

sub: INR A
RET

- a) 00H b) 01H c) 02H d) none of these

Case 1: main & sub

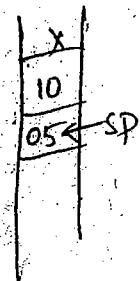
main → 1000H
sub available → 2000DH

1000: MVI A, 1000H
1001: ← Call 2000H
1002 ←
1003 → 00
1004 → 20
PC = 1005 ~~1006~~ ~~1007~~ ~~1008~~ ~~1009~~ ~~100A~~
RET

PC = 2000: INR A, A → 01H
2001: RET

PC = 2002
after expansion of the subroutine
PC = 1005

A = 01A



Interrupt

→ Out

allocate

hardware
interrupts



TRAP

RST 2-5

Ve

am what is

Case (ii) 1000 MVI A, 00H $\Rightarrow A = 00H$. Club routine can access
next memory location

1002 CALL 1005H

1003-05

1004-10

PC = 1005 INR A $\rightarrow 01H$

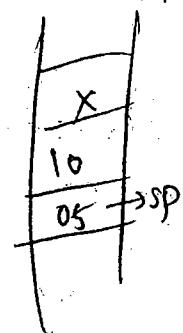
1006 RET

PC = 1007

after ret if SP in decremented by two.

so (i) PC increased at 1005

so (ii) PC = 1008 INR A $\rightarrow 01+01 \Rightarrow 02H$



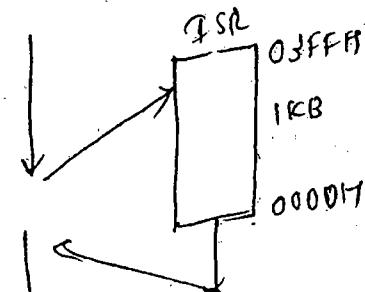
Interrupts

→ out of 64KB of External memory starting 1KB 25

allocated for ISR.

Hardware
Interrupts

(8)



Software
interrupt

TRAP RST 3-5 RST 6-5 RST 5-5 INTL

TRAP

RST 3-5 RST 6-5 RST 5-5 INTL

Vectored Interrupts

Non vectored

TRAP (4-5)

INTL

RST 7-5

RST 6-5

RST 5-5

structure

→ TRAP is internally 4.5

RST 2.5
X8 → (8 bits per) 10 multiplies with 15

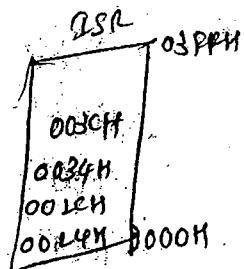
60.0

60 - Decimal Value

16(60)

3-12(C)

003CH is 16 vector location for RST 2.5



RST 4.5
X8

36.0

6(36)
2-4

002CH is 16 vector location for TAP.

RST 0 → 0000H

RST 1 → 0008H

RST 2 → 0010H

RST 3 → 0018H

RST 4 → 0020H

RST 5 → 0028H

RST 6 → 0030H

RST 7 → 0038H

RST 6.5
X8

52.0 → 9

16(5L)

3-4 → 00034H

TAP.

RST 5.5
X8

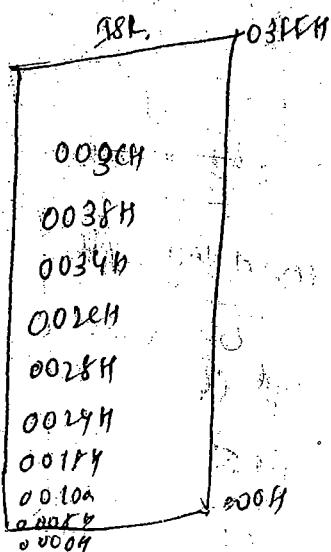
44.0

16(44)

2-C

vector 002CH

2X8 3X8 4X8
16(16) 16(14) 16(32)
1-0 1-8 2-0



6) max

RST
AST 6.
AST 5.
INT.
→ TRAP

We can

→ EI

or Disabl.
g) Priority

1

fixed (a)
normal

TRAP

RST 2

RST 6

RST 5

INTR

fixed -1

orders 95 / 28

RST 8.5 TRAP

6.5

5.5

TRAP
TRAP

R
03PPH
H
1
0000H

RST 5.5

XX

44.0

16(44)

2-C

new 002CH

48x8

16(32)
2-D

CH

b) maskable

RST 7.5
RST 6.5
RST 5.5
INTR.

→ TRAP is

Non Maskable

TRAP

Non maskable interrupt by any mean

We can not enable or disable the interrupt.

→ EI & DI using these instruction

or disable the interrupt.

c) Priority

fixed (0)
Normal priority system

rotating priority system.

RST

TRAP

RST 7.5

RST 6.5

RST 5.5

INTR

fixed → Enable

orders	1st	2nd	3rd
RST 7.5	RST 7.5		
TRAP	TRAP	RST 6.5	
6.5			
5.5			
TRAP			
INTR			

Rotating Priority Enable

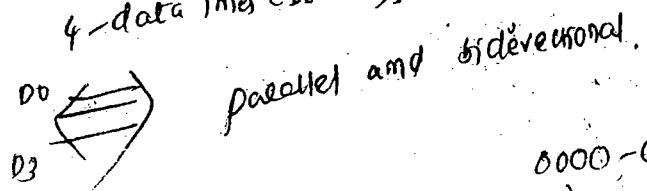
1st	2nd	3rd
RST 7.5	RST 6.5	RST 5.5
6.5	RST 5.5	TRAP
5.5	TRAP	INTR
TRAP	INTR	RST 7.5
INTR		

PLD: It is a programmable device or it is a multipurpose clock driven programmable logic devi → 8-bit

4004 → 4-bit PLD It is operate on 4-bit data

→ It was released by Intel in 1971

→ It has 4-data lines (D0 - D3).



0000 - 0

1001 - 9

1111 - F

4 = 16⁴ devices

= 16-bit devices

= 16-instruction or opcodes possible.

→ 10 Address lines



$2^{10} = 1024$ bytes possible
but ~ 640 bytes available

→ 16 pins IC

Dual in line package

power supply -5V, +5V & ±12V

clock freq 103 KHz

2800 (PMOS) It has

calculator, basic application

8080: It is 8-bit μP operation on 8-bit data

→ Intel by 1972

→ 8-bit μP

→ 8-bit

→ 14

→ 2

→ 3

→ 4

→ 5

→ 6

→ 7

→ 8

→ 9

→ 10

→ 11

→ 12

→ 13

→ 14

→ 15

→ 16

→ 17

→ 18

→ 19

→ 20

→ 21

→ 22

→ 23

→ 24

→ 25

→ 26

→ 27

→ 28

→ 29

→ 30

→ 31

→ 32

→ 33

→ 34

→ 35

→ 36

→ 37

→ 38

→ 39

→ 40

→ 41

→ 42

→ 43

→ 44

→ 45

→ 46

→ 47

→ 48

→ 49

→ 50

→ 51

→ 52

→ 53

→ 54

→ 55

→ 56

→ 57

→ 58

→ 59

→ 60

→ 61

→ 62

→ 63

→ 64

→ 65

→ 66

→ 67

→ 68

→ 69

→ 70

→ 71

→ 72

→ 73

→ 74

→ 75

→ 76

→ 77

→ 78

→ 79

→ 80

→ 81

→ 82

→ 83

→ 84

→ 85

→ 86

→ 87

→ 88

→ 89

→ 90

→ 91

→ 92

→ 93

→ 94

→ 95

→ 96

→ 97

→ 98

→ 99

→ 100

→ 101

→ 102

→ 103

→ 104

→ 105

→ 106

→ 107

→ 108

→ 109

→ 110

→ 111

→ 112

→ 113

→ 114

→ 115

→ 116

→ 117

→ 118

→ 119

→ 120

→ 121

→ 122

→ 123

→ 124

→ 125

→ 126

→ 127

→ 128

→ 129

→ 130

→ 131

→ 132

→ 133

→ 134

→ 135

→ 136

→ 137

→ 138

→ 139

→ 140

→ 141

→ 142

→ 143

→ 144

→ 145

→ 146

→ 147

→ 148

→ 149

→ 150

→ 151

→ 152

→ 153

→ 154

→ 155

→ 156

→ 157

→ 158

→ 159

→ 160

→ 161

→ 162

→ 163

→ 164

→ 165

→ 166

→ 167

→ 168

→ 169

→ 170

→ 171

→ 172

→ 173

→ 174

→ 175

→ 176

→ 177

→ 178

→ 179

→ 180

→ 181

→ 182

→ 183

→ 184

→ 185

→ 186

→ 187

→ 188

→ 189

→ 190

→ 191

→ 192

→ 193

→ 194

→ 195

→ 196

→ 197

→ 198

→ 199

→ 200

→ 201

→ 202

→ 203

→ 204

→ 205

→ 206

→ 207

→ 208

→ 209

→ 210

→ 211

→ 212

→ 213

→ 214

→ 215

→ 216

→ 217

→ 218

→ 219

→ 220

→ 221

→ 222

→ 223

→ 224

→ 225

→ 226

→ 227

→ 228

→ 229

→ 230

→ 231

→ 232

→ 233

→ 234

→ 235

→ 236

→ 237

→ 238

→ 239

→ 240

→ 241

→ 242

→ 243

→ 244

→ 245

→ 246

→ 247

→ 248

This is a

single logic device

275

8-bit data lines

288 → 160 bytes

→ 0F

→ opcodes

16 Address lines

$$2^{14} = 2^4 \times 2^{10}$$

⇒ 16 KB (ROM & RAM)

standard 18 pin IC

→ Dual in line package (DIP)

→ -5V, +5V & +12V power supply

→ clock freq 200 kHz

→ 3500 NMOS

→ basic application is traffic light controller

→ basic

8080

→ Synthesized in 1974

→ 8-bit MP

→ 8-data lines

→ 16 address lines

→ 916 = 64 KB (ROM & RAM)

→ 40-pin IC

→ DIP

→ +5, -5 & +12V

→ fc = 2 MHz

→ EP had 6000 NMOS on it.

→ microprocessor temp controller

1986

+ 1986
, gr is built up required only +5V DC power supply.

, FC 20 MHz

, 6500 (NMOS)

, Temp / pressure controlled

108B → CPU

It is first general purpose 16-bit mc.

, pp granted from 1988

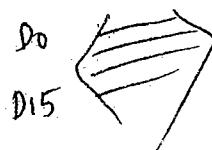
(16-bit mc) 4 general purpose registers

AX, BX, CX, DX → 4 general purpose registers

SI, DI, BP, SP, IP → 5 special purpose registers

+ base index → SI
DS, ES, CS, SS → 4 segment registers

16-data lines

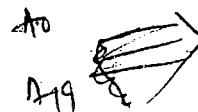


$$2^16 = 64 \text{ KB - RAM devices}$$

- OPI

- opcode(8), instructions are available

) It has 20 Address lines



$$2^{20} \rightarrow 2^{10} \times 2^{10}$$

$$1024 \times 1024$$

$$1K \times K = \underline{\underline{1 \text{ MB}}}$$

→ Built

only 6

one

1024

256

→ 192 KB mem

in 16 bit

in 16

Instead

64 KB OS

gr is

available

→ +5 V

fc = 5 to

→ gr oper

→ gr is v

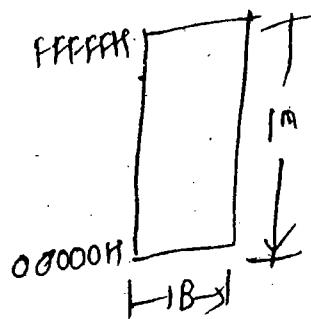
gr is v

gr is v

Input

1 MB (RAM/ROM)

→ But at most it can access
only 64 KB, remaining is free
space.

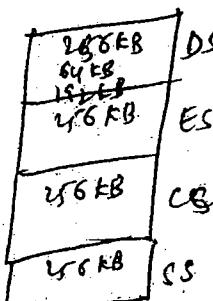


(1 MB) → divided into 4 segments

$$\frac{1024 \times 1024}{16} = 64$$

= 256 KB

→ 192 KB memory ~~available~~
in free space available
in each segment.



→ Instead of these, it is divided into 16 segments

$$\frac{1024 \times 1024}{16} = 64 \text{ KB}$$

it can access whole size of

64 KB of memory.

→ It is a 40 pin IC

→ available in DIP

→ +5 V DC single supply

is available.

→ $f_C = 5 \text{ to } 10 \text{ MHz}$

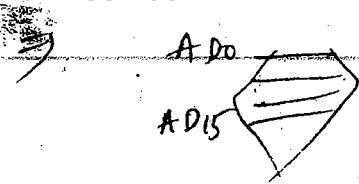
→ It operates in min mode & max mode.

→ used for small system
with one procIV
medium & large system

with one or more processors

→ It is operated in min mode if $f_C = 5 \text{ MHz}$
or in max mode if $f_C = 10 \text{ MHz}$

B.



(lower order Address and data lines
are multiplexed)

10) psw 0

comparator

29000 (NMOS)

base application 8-bit CPU

% duty cycle is only 33%.

8085 MP

i) 8-bit MP

ii) 8-data lines

iii) 16-address lines

iv) 64 KB (ROM/RAM)

$$f_C = 2 \text{ MHz} - 3 \text{ MHz}$$

v) ADD, ALE multiplexed

vi) ADX

vii) No. of hardware interrupts - 5.

TRAP, RST 7.5, RST 6.5, RST 5.5, INT

viii) No. of software interrupts - 8

RST 0 - RST 8

ix) flag register - 8-bit

5 active flags

CF, PF, AC, ZF, SF

8086

i) 16-bit MP

ii) 16-data lines

iii) 20-Address lines

iv) 1MB (ROM/RAM)

v) 5MHz - 10MHz

vi) ADD - A15

vii) only 2 I/O multiplexed

HARDWARE → non maskable

INT → maskable

viii) 256 I/O interrupt

Type 0 - Type 256

ix) flag registers - 16-bit

5 active flags

6 → status flags CF, PF, AC, ZF, SF
OF

3 → control flags TRAP, interrupt,

Direction

x) SFT → ?

B, C, D

ly OR is acc
+ micro

A

64KB

No. of

256

6500

No. of

No. of

CPU &

NewtonDesk.com

i) data buses

ii) psw - 16-bit
combination of A + flag register
higher byte towards left

iii) STX → general purpose register
B, C, D, E, H, L.

iv) If it is accumulator based
in microprocessor.

ADD B

$$A = A + B$$

64 KB

No segmentation

256 - instruction

6500 (NMD)

No - parallel processing

17

w/ interrupt
non maskable
maskable
interrupts

18 bits

- 16 bits

g

F, FF, ACF, ZF, SF

OF

TRAP, interrupt

19. No - physical address concept

20. CPU is not divided

10. psw - 16 bit
only flag register.

11. 4 - general purpose 16-bit

AX, BX, CX, EDX

12. 8 - first general purpose
AH, AL, BH, BL, CH, CL, DH, DL

13. register based MP.

ADD BL, CL

$$BL = BL + CL$$

14. 1MB

15. 4 - segments (utilization
memory)

16. 64 KB - instruction

17. 29000 (NMD)

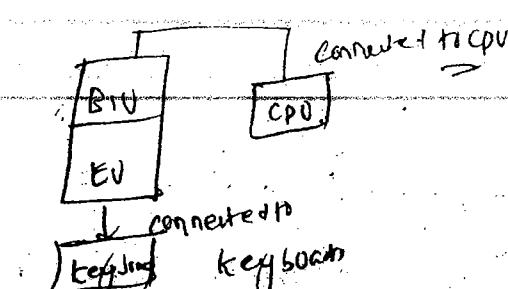
18. Parallel processing is possible
because of instruction queue by
lengths of every byte is 6 bytes and
width is 16.

19. Physical Address Concept
(How to generate 20-bit physical
address by using 16-bit register)

20. CPU is divided into 3 unit

i) Bus Interfacing Unit (BIU)

ii) Execution Unit (EU).



Instruction

I. Data

II. Address

III. logic

IV. Branch

V. Memory

VI. General

VII. Source & Dest.

VIII. How many

IX. which

X. No. of

XI. Data

① Mov d,

Copy

or write

Source →

destination

Ex.

b.

a.

m

B1U functions

i) fetch instruction from memory

ii) read

iii) write

iv) handle all transfer of data for execution unit.

EU functions

i) it tells B1U where to fetch instruction

physical Add. = Total Base Address + Effective Address

MOV AL, (4000)

EA = 4000

DS = 1234H

ES = 5638H

SS = 1239H

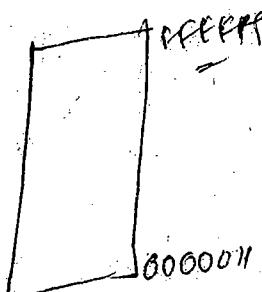
GS = 5639H

BA = 1234H add 0

TBA = 12340H ~~+ 0000~~= ~~1234~~

DA = 12340 + 40000H

PA = 16340H



revert to CPU

Infection set 95 8085 :-

5yt 5-types. ($2^8 = 256$ opcodes
analogous)

apprenſomſ

from memory

fer of Data box

where to get

ve Address

Data transfer instructions

Algorithmic Instructions

Logical & Bit manipulation instructions

mechanical control / process control

General format

function?

source & destination

flowing only by the gravitation
of water.

which flags obstructed

which flags are negated
No. of clock cycles are required
instructions:-

6) No. 08 Date _____
I. Data transferred instructions:-

~~SECRET~~

MOV d, R

Copy data from source to destination with source information or with out altering it. Store information in 8-bit register or memory location.

Source → Bitfield 8-bit Register or memory location , R

decrementing \rightarrow B, C, D, E, H, K, M, A \rightarrow 8-bit register
 before m - memory pointed

MOV B,C

EY. before $B = 49 H$ ~~$688 H$~~

before $B = 6 \text{ G}$
after $\mu\text{G/cm}^2$ $B = 6 \text{ G}$, $C = 6 \text{ PH}$

MOV D, M

D, " (m-mem) for or always takes
rooter,

MOV D, M

D = 40

AC = 4000H (40H)

D is loaded with [HL] = 40H

Ex: MOV M, E E = 60H

transferred (i.e. data from reg. to memory.)

E = 60H

MC: 3000H

Ex 4 MOV M, M

Memory to memory data is not possible

M → takes off content of R[pc]

→ in copy M → (the length of instruction) min 1 byte max
3 byte instructions

Instruction = opcode + operand

↓

min - 8 bit

8-bit register

max - 16 bit

16-bit register

memory pointer

(a) 8-bit port address

(b) 16-bit Address

(c) 8-bit immediate data

(d) 16-bit immediate data

MOV B, A
 &
 opcodes operands.

D = 000

C = 001

B = 010

E = 011

H = 100

L = 101

M = 110

A = 111

byte max

	S	B	C	D	E	H	L	M	A
d									
B	40	41	42	43	44	45	46	47	
C	48	49	4A	4B	4C	4D	4E	4F	
D	50	51	52	53	54	55	56	57	
E	58	59	5A	5B	5C	5D	5E	5F	
H	60								6F
L	68								
M	70	71	72	73	74	75	76	77	
A	78								7F

MOV C, E
01 001, 011
4 BH

MOV M, M

01 110 110
26H

26H is not possible

so 26H is allocated for HLT
 instruction

~~No flags are affected during data transfer~~

Instructions

(2) MVI d, S

→ 8-bit immediate data

d → either 8-bit register or a memory location.

MVI B, 49H

B = 49H

MVI A, 88H

16bit → 1 byte
 D add 8 bits → 1 byte
 5 bits Data 8 bits → 1 byte
 → op code 9 bits → 1 byte

(3)

LXI d, S

↓
 16-bit Immediate data

Register

pair

(PC, DE, AC, SP, PSW)

Q: write
pair w/
off. P

Q1

Q. push the content of 16 accumulator on to stack then

what are sequence of instructions

push AC

push PSW

push BC

push None

push A

8-bit gathering

but Stack is 16-bit Register

It is already operated on 16-bit.

(1)

(4) 1K

refer

LXI B, 1234H

↳ no need to write BC

Load 16 specified destination with 16 bit immediate

data

$$BC = \frac{1234H}{H} - 1234H$$

data 16 → 2 byte

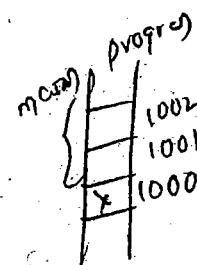
opcode → 1 byte

LXI B, 1234H

00111
00

LXI SP, 1000H

SP = 1000H

Data available in 0FFFH and OFFEH

misuse - DE register

Q: Write a Assembly language program

pair with 5678H

→ 2 byte - 2 memory location

(Q1) P1 MVI D, 56H

→ 2 byte - 2 memory loc

MOV E, 78H

→ 2 byte - 4 memory

HLT

DE = 5678H

K them

(Q2) LXI D, 5678H → 8 byte - 4 memory

HLT

DE = 5678H

(Q3)

IN S

IV

8-bit port Address

IN 38H

Prog 8 An Accelerator \Rightarrow 30H

B

 \Rightarrow 38H

A IN 20H

1 byte 1 byte

→ It is a 2-byte instruction or takes 2 memory locations.

⑦ LDAX

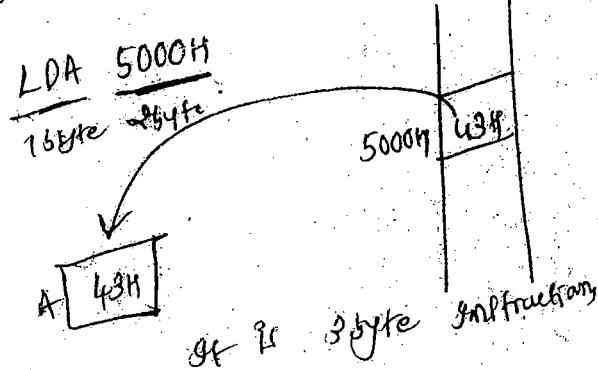
LDP

⑥ Out of \downarrow \rightarrow Transfer of data from MP to IO device.
 \Rightarrow 1 8-bit port address

LDA

000111Out 29H
1 byte 1 byte \Rightarrow 2 bytes

⑤ LDA \downarrow 16-bit Address \rightarrow Content of 16-bit Address
 \rightarrow Load PC Accelerator

LDA 5000 \rightarrow AddressLXI 5000 \rightarrow data

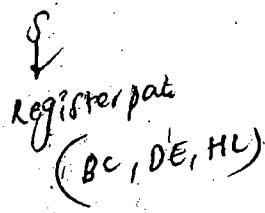
30H

MVI A, 40H \rightarrow AddressIN 40H \rightarrow Port Address \rightarrow It is

⑧ STAX

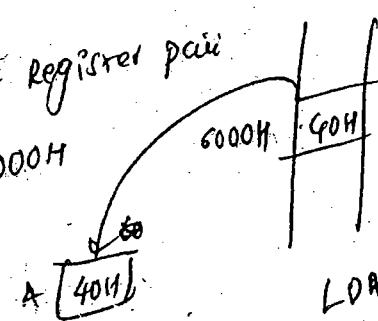
STA

 \rightarrow StoreSTA
1 byte

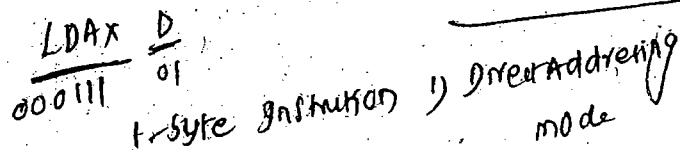
⑦ LDAX 
 register pair
 (BC, DE, HL)

memory location
 6000H

reg deccr.

LDAX D 
 DE register pair
 upper DE = 6000H
 A [40H] LDR

area

LDAX D 
 000111 01
 1-byte instruction i) Direct Addressing mode

ii) 3-byte instruction
 iii) 3-memory locations

LDAX
 i) indirect addressing mode.
 ii) 1-byte instruction
 iii) 1-memory location for execution

memory location.

⑧ STA d → (16 bit Address)
 content of in 16-bit address
 → store

STA 9000H
 1-byte 9000 → 40H
 A [40H]
 3-byte instruction and direct addressing mode.

⑨ STAX d
 register pair

Q) write a Assembly language program to transfer data
from 2000 to 5000H memory location

80L LXI D, 5000H

LXI H, 2000H

MOV C, 03H

X: MOV A, M (w) LODA M

STAX D

(NR H → HL 200)

(NXD → D + 500)

DEC C

JNZ X

HLT

2000	11	5000	00
2001	12	5001	00
2002	13	5002	00
2003	14	5003	00
2004	15	5004	00
2005	16	5005	00
2006	17	5006	00

Q) after

Content

3

3

1

1

1

a) 8F8F

⑩ XCHG HL ↔ DE

→ Both (E) SORG and destination are implied

HL = 1234H

DE = 5678H

XCHG

HL = 5678H

DE = 1234H

It is 1 byte instruction. It take only one memory location.

⑫ PCHI

→ load

⑪ XTHL

Content of HL to Content

→ It Exchange content of stack.

→ 1 byte

⑬ SPHL

→ load

7 bytes
ram free data

HL = 1234H

HL = 0304H

	1000
01	0FFFH
02	0FFEH
03	0FFDH
04	0FFCH

Q. after 16 execution of following instruction what will be
Content in HL register pair

3 LXI H, 2095H

3 LXI B, 8FBFH

1 push B

1 XTHL

1 POP H

1 HLT

a) 8FBFH b) 2095H c) 1234H d) None

HL = 2095

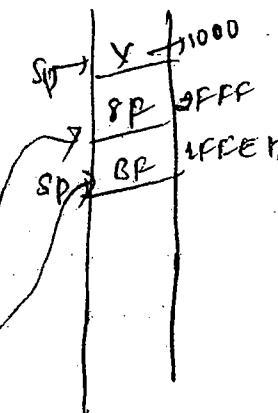
BE = 8FBF

XTHL.

HL = 8FBF

SP = 2095

HL = 2095



⑫ PCHL → load program counter to content of HL register pair.

HL = ABCDH

PC = 9000H

PCHL

PC = ABCDH

HL = ABCDH

off location.

→ 1 byte to instruction.

Content

⑬ SPHL → load SP to content of HL register pair.

$HL = 1122H$ $SP = 100011$ $SP \leftarrow$ $SP = 1122H$ $HL = 1122H$

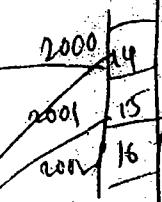
② LHLD

16-bit Address

→ Load HL register pair the content of that 16-bit Address

LHLD 2000H

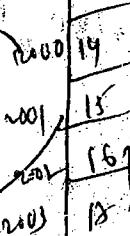
HL



Lower Address

HL

15 14



Higher Address

LHLD 2002

HL = 1234H

3 byte instruction

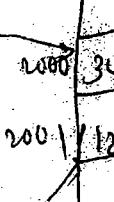
SHLD

16-bit Address

HL = 1234H

SHLD 2000

12 34



→ 3 byte instruction

⑥ PUSH

→ write data

→ 4 byte size

i) 95 1F

ii) 8-6

2 bytes

iii) 80 1E

16-bit A

iv) 98 1E

16 long

⑦ POP

Ref

→ read data

SP = SP

SP = S

(16) push 5

BC 2112.24

$$DE = 334411$$

$$MC = 55 \times 10^4$$

register pain

BC, PE, HC, PSW

BC, PE, HC, PSW

→ write data into memory.

~~PLATE B~~ LTB SP. 1000 H

push B

$$sp-1 \quad \left\{ \quad sp-2$$

→ If byte instruc⁹ S_{P-2} J^{P-2}
then HOD 8-bit immediate data

→ It is part of instruction

(a) 8-bit port Address them 16 bytes.

2-3y res

in 95 it part of instruction

either 16-bit immediate or
length of the instruction is 3-ff
hex

16-514 Address

them

strength of the construction is 50% higher

16-51
iv) go 15 part 15 instruction

neither 16 SIC nov 16 SIC

iv) It is 1 byte
It length

(17) pop ♂
register pair

read data from memory

read and (too) increment
by

$SP = SP + 1$ ($SP + 1$) post increment by 2

$$S_P = S_I + 1$$

Arithmetic Instructions

① ADD S

$$A = A + [S]$$

8-bit register/memory location

ADD B

$$A = 49H \rightarrow 0100\ 1001$$

$$B = 94H \rightarrow \begin{array}{r} 1001\ 0100 \\ 1101\ 1101 \\ \hline \end{array}$$

$$DDH$$

$$\rightarrow OF=0, PF=1, SF=1, AC=0, ZF=0$$

$\frac{\text{ADD } B}{1000\ 00\ 00} \rightarrow 80H$, ADD instruction is 100000
 → If excute opcode of ADD instruction is 100000
 + 1 byte instruction.

ADC S \Rightarrow 16-bit Addition
 8-bit register/memory location

$$A = A + [S] + C$$

$$BC = 1299H$$

$$DE = 4585H$$

$$MVI A, 94H \rightarrow A = 99 + 1001\ 0100$$

$$MVI E, 85H \rightarrow E = 85 \rightarrow \begin{array}{r} 1000\ 0101 \\ 1001\ 1101 \\ \hline \end{array}$$

ADD E

CF=1

$$MVI A, 12H$$

$$MVI D, 45H$$

ADC D

$$A = A + D + C$$

$$A = 12 \rightarrow 0001\ 0010$$

$$D = 45 \rightarrow 0100\ 0101$$

$$\begin{array}{r} 0001\ 0010 \\ 0100\ 0101 \\ \hline 1101\ 1111 \end{array}$$

CF=0

③ : ADI S

OF=0

$\rightarrow 91\ 11$

$$A = 11$$

ADI S

$$A = 99H$$

$$+ 92H$$

④ ACI S

8-bit

$$A = A -$$

⑤ SUB S

8-bit

$$A =$$

SUB B

method 4 A
B

CF=0

ACF=1

$CF=0, SF=0, PF=0, ZF=0$

(3) ADD S → 8-bit Addition
↓
8-bit immediate data

→ 8-bit instruction

$$A = A + 8\text{-bit data}$$

ADD 42H

$$A = 99H \rightarrow 1001 1001$$

$$42H \rightarrow 0100 0010$$

$$\underline{1011011}$$

$CF=0, ZF=0, SF=1, PF=1, ACF=0$.

(4) ACI S → 16-bit Addition
↓
8-bit immediate data

$$A = A + 8\text{-bit } + C$$

(5) SUB S
↓
8-bit register/memory location

→ 15-bit instruction

$$A = A - [S]$$

SUB B

method 4: $A = 92H \rightarrow 1001 0010$

$$B = 65H \rightarrow 0110 0101$$

$$\underline{0101101}$$

method 2

$$A = 92 \rightarrow 1001 0010$$

$$B = 65 \rightarrow \begin{array}{r} 1001 1011 \\ + 0010 1101 \\ \hline 100101101 \end{array}$$

$CF=1, ACF=0, PF=0$

$SF=0, ZF=0$

↓ compare take Invert operation
∴ $ACF=10, ACF=1$

0010
0101
1111

A A

$CF=0, PF=1$

$ACF=1, ZF=0$

$PF=0$

8) SDB

$$A = A - (S) \rightarrow B$$

$S \rightarrow$ 8-bit/mem off location
(9)

\rightarrow 1 byte instruction

⑦ SUI S

↓
8-bit immediate data

$$A = A - 8\text{-bit immediate data}$$

⑧ SBI S

↓
8-bit Im/Memory

⑨ INC/INR d

8-bit Register/Memory

$$B = FFFH$$

INC B

$$B = 00$$

1111 1111

+1

17 0000 0000

→ either increment (or) decrement depends on result.
Other bits are effected.

$$INR M$$

\rightarrow we can't directly access content of memory.
So better to read data from memory and then store it.

Load Sack to memory

⑩ DEC/DCR

9

8-bit Regt

$$D = 00H$$

DCR?

$$D = ?$$

\rightarrow CF NOT

1 byte

IMP

DCX B

$$HL = 1^r$$

$$HL = 20C$$

DCX H

$$HL = 1^r$$

Notes NO 6

XRA A

LXI B

X: DCX B

JNE X

HLT

a) notes

LXI B,

X: DCX B

MOV A,C

ORA B

JNL X;

HLT

one more 97KM

$D = 00H$

DCR D

 $D = FFH$

\rightarrow CF not effected, others are effected

~~JMP~~
1 byte instruction
II DCX d
register pair

 $HL = 2000H$

DCX H

 $HL = 1FFF$

Note: No flags are effected during increment or decrements

XRA A

 $A = 55 \rightarrow 01010101$

LXI B, 0000H

~~55~~ $\overline{01010101}$ $\overline{00000000} 2F-1$

not effected

X: DCX B

JNZ X ~~2F-1~~a) none
b) x-times
c) infinite
d) none

HLT

in memory.

and moreover

LXI B, 0000H

 $BC = 0000H$

X: DCX B

 $\rightarrow 0000H$ $0005 \quad 0004 \quad 0003$

MOV A, C

 $\rightarrow A = 00 \quad 05 \quad 04 \quad 03$

ORA B

 $00 \quad 00 \quad 00 \quad 00$

JNZ X;

 $00 \quad 00 \quad 00 \quad 00$

HLT

 \rightarrow none

108 8,000H

III 109

① DCR B

JNL X

HLT

- g) AAC one b) 7 times c) subtract -d) none -

② INX d
register pair

INX H

$$HL = 2000H$$

$$HL = 2001H$$

③ DAA

f) decimal adjustment after addition
If after addition invalid BCD is there this
converts to valid BCD

→ If is

② AND

AND 0

99H

00H

PF=1

ZF=0

SF=0

OF

④ DAD S
register pair

$$HL = HLT [RPL]$$

DAD SP

$$HL = 2000H$$

$$SP = 3000H$$

DAD SP

$$HL = 5000H$$

?

III logical & Bit manipulation instructions :-

① ANA S

→ Register/memory location

- 8 bit register / memory
→ used to mask 16 required bits/nibble / byte.
→ used to reset it system.

ANA B

$$A = 48\pi \rightarrow 0100 \begin{smallmatrix} 1000 \\ 0000 \end{smallmatrix}$$

$$B = FOK \rightarrow \frac{1111}{100000} 0000$$

$$A_{AB} =$$

→ talk to lower middle

* 81

imp
→

$$CF = 0$$

$$ACF = 1$$

$ACF = 1$ ~~never~~ always (no po

$CF = 0$
 $ACF = 1 \Rightarrow$ always (no possibility to do else)

→ It is a life-gastrulation.

② ANS ↓

↓
8-bit immediate dat

ANI 00K

NI 001
99H → 1001 1001

$$\begin{array}{r} 994 \\ \times 100 \\ \hline 99400 \end{array}$$

00
1
15 the nibbles are marked

$$\therefore PF = 1$$

$$2F = \emptyset$$

SF = 0

→ It is 2-byte instruction.

③ XRA S

8-bit register / memory

- It is used for comparisons, and it is used clear the content in the accumulator.

XRA B

$$A = 48H \rightarrow 0100\ 1000$$

$$B = 08H \rightarrow \begin{array}{r} 0000\ 1000 \\ - 0100\ 0000 \\ \hline 0100\ 0000 \end{array}$$

SF = P

P = 0

Z = 0

XOR (Y₁ Y₂)Y₁Y₂

Y

(MP) ACF = 0, CF = 0 always set to 0. (no possible)

XRA A

$$A = 89H \quad 1000\ 1001$$

$$= 89H \quad 1000\ 1001$$

$$\begin{array}{r} 1000\ 1001 \\ - 0000\ 0000 \\ \hline 0000\ 1001 \end{array}$$

P = 1 SF = 1

SF = 0

Z = 1

to content in 15

XOR (Y₁ Y₂)Y₂Y₁Y₂

accumulator.

it is a 1-type instruction.

- It is a 1-type instruction.
→ If 8085 MP executes XOR(Y₁ Y₂) and place the result into specified destination. To the following sequence of instructions
as expected

XOR (Y₁ Y₁)XOR (Y₁ Y₂)XOR (Y₂ Y₁)What is the content now in Y₁ Y₂.

a) $Y_1 = \frac{Y_1 + Y_2}{2}$

b) $Y_2 = Y_1 + Y_2$ e. Y₁ & Y₂ swapped

c) none of the above

④ XRI S

Y = 61H

XRI 48H

A = 29

= 4

→ It is a

write a

and

C1

P1

X

P2

Y

Z

clear R

$$\begin{array}{r}
 03H \rightarrow 0000\ 0011 \\
 09H \rightarrow 0000\ 1001 \\
 \hline
 0000\ 1010 \\
 \xrightarrow{\quad} 0AH \\
 \xrightarrow{\quad}
 \end{array}$$

(i) r2)

$$\begin{array}{r}
 r_1 = 03H \rightarrow 0000\ 0011 \\
 r_2 = 0A H \rightarrow 0000\ 1010 \\
 \hline
 0000\ 1001 \\
 r_1 \Rightarrow 09H
 \end{array}$$

ste)

content of R

$$\begin{array}{r}
 XRI(r_2, r_1) \\
 r_1 = 0000\ 1001 \\
 r_2 = 0000\ 1010 \\
 \hline
 0000\ 0011 \\
 r_2 \Rightarrow
 \end{array}$$

$$\begin{array}{r}
 r_1 = 09H \\
 r_2 = 03H
 \end{array}
 \quad \left. \begin{array}{l} \{ r_1 \text{ & } r_2 \text{ are swapped.} \\ \text{it result from} \\ \text{of instruction} \end{array} \right.$$

XRI S
 \downarrow
 $S = 51E$ immediate data.

$$\begin{array}{r}
 XRI 48H \\
 A = 28H \rightarrow 0010\ 1000 \\
 = 48H \rightarrow 0100\ 1000 \\
 \hline
 0110\ 0000
 \end{array}$$

wrapped

above.

and

P1.

XRA A - 1 byte

MUL A, 00H

It is 2-byte instruction.
 write a program to clear 16 content in Accumulator
 clear the content in B-register.

Content in Accumulator

B-register.

~~A = 001~~~~py: MUL B,00H~~

⑤ ORA S
 \downarrow
 8bit register/m

ORA B

$$\begin{array}{r} A = 48H \rightarrow 0100 1000 \\ B = 06H \rightarrow 0000 0110 \\ \hline 0100 1110 \end{array}$$

~~and~~
 $\boxed{CF=0}$
 $ACF=0$

→ AF always set by system.

→ AF is a byte instruction.
 → AF is a byte instruction.

⑥ ORI S
 \downarrow
 8bit immediate data

AF is a byte instruction.

⑦ CMP S
 \downarrow
 8bit of memory location

A - {S}

→ AF not covered only check AF conditions.

Op S not covered only check AF conditions.

A < CS CF = 1 ZF = 0

A > CS CF = 0 ZF = 0

A = CS CF = 0 ZF = 1

CMP B
 $A = ?$
 $B = ?$

CF =

AC =

AF =

ZF =

CF =

ACF =

CMP D

A = 80H

D = 7DH

SUB S

A = A - {S}

→ destination is not needed

⑧ CPI D

CF =

⑨ STC

→ only

→ AF

CMP B

$$A = 40H \rightarrow \begin{array}{r} 111 \\ 0100 \\ 0000 \end{array}$$

$$B = 49H \rightarrow \begin{array}{r} 11110111 \\ \hline 01001001 \end{array}$$

$$\begin{array}{l} CF=1 \\ AF=0 \\ ACF=1 \\ SF=1 \\ ZF=0 \end{array}$$

in comp

$$A = 40H \rightarrow 0100 0000$$

$$B = 49H \rightarrow \begin{array}{r} 10110111 \\ \hline 11110111 \end{array}$$

$$PF=0 \quad ZF=0$$

$$\begin{array}{l} CF=1 \\ Z \end{array}$$

$$\begin{array}{l} CF=0 \\ ACF=0 \end{array} \quad \begin{array}{l} \text{in comp} \\ CF=1 \\ ACF=1 \end{array}$$

CMP D

$$A = 80H \rightarrow \begin{array}{r} 111 \\ 1000 \\ 0000 \end{array}$$

$$D = 70H \rightarrow \begin{array}{r} 0111 \\ 0000 \\ \hline 00010000 \end{array}$$

$$CF=0$$

$$ZF=0$$

$$CF=0 \quad ZF=0$$

so it is not equal

⑧ CPI 000's
J-set immediate data.

$$A = 40H \rightarrow 0100 0000$$

$$B = 40H \rightarrow \begin{array}{r} 0100 \\ 0000 \end{array}$$

CF=0, ZF=1

⑨ STC (Set TC carry flag) CF=1
only carry flag is effected after flags not effected
i.e. a 1-byte instruction.

Complement (for carry)

$CF = 0$ $CF = 1$ { carry cc is affected
 $CF = 1$

1897e instruction

CMA Complement (if content in 16 accumulator)

no flags are affected

$A = 00H \rightarrow 0000\ 0000$

$\overline{1111\ 1111}$

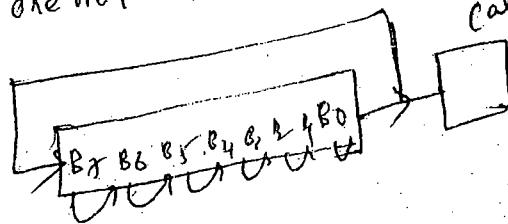
$\leftarrow F\ F\ A$

RRC (rotate eight without carry)

(6015 source and destination are implied)

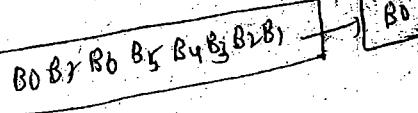
d 8
 ↓
 one notation

A,

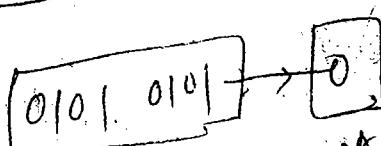
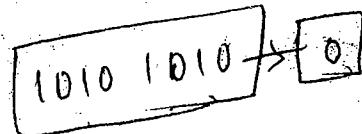


1st rot=100

content of counter.



$A = AAH\ CF = 0$



After 3rd rotation 55H

→ if IS
 → if IS
 → exchange
 nibble to 10

$A = 99H$

$\boxed{100}$

$\boxed{01}$

$\boxed{0}$

$\boxed{1}$

$\boxed{1}$

\rightarrow after 4th

⑬ ROR/RRR C

→ if IS 9-

→ move

\boxed{B}

\boxed{A}

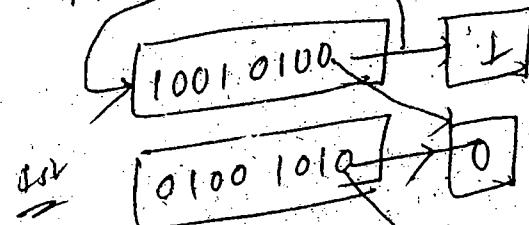
→ 8F acts as Application of shifting:-

→ 8F is for 8-bit rotation

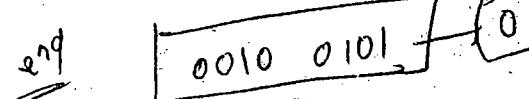
→ 8F acts as divided by 2

→ Exchange the lower nibble to higher nibble and higher nibble to lower nibble.

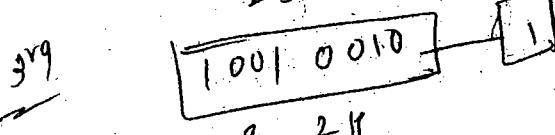
$$A = 94H \xrightarrow{8F} 1001\ 0100$$



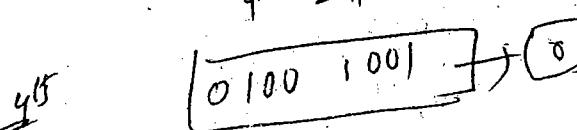
$$A = 94AH$$



$$25H$$



$$92H$$



$$49H$$

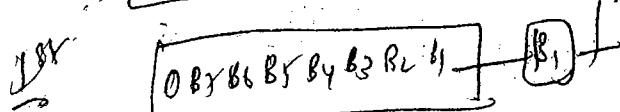
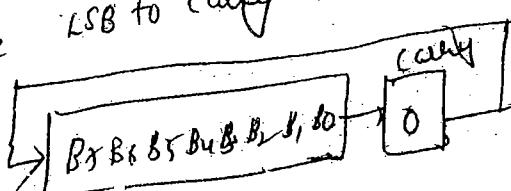
→ After 4F rotation exchange lower nibble to higher nibble via carry

(13) ROR/RRC (rotate right through carry)

→ 8F is 9-bit ~~8-bit~~ rotation

→ 8F acts as divided by 2

→ move LSB to carry and carry to MSB



After execution of the following instruction what is it contents
to the accumulator.

O: 90

MVI A, 45H

MOV B, A

STC :

CNC

SHR

XRA B

A 45H

B 54H

A = 22H

B = 45H

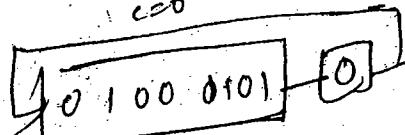
Y 63H

Z 84H

A = 45
B = 45

c = 1

c = 0



$$A = \boxed{0\ 0\ 1\ 0\ 0\ 0\ 1\ 0} \text{ (1)}$$

$$\begin{array}{r} 0\ 4\ 5\ H \\ \times 0\ 1\ 0\ 0 \\ \hline 0\ 1\ 1\ 0\ 0\ 1\ 1 \end{array}$$

⑫ RAL /

→ 8F i

⑬ RL

Rotate left without carry

It is for 8-bit rotation

MSB to carry and MSB to LSB.

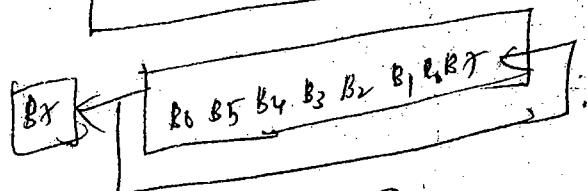
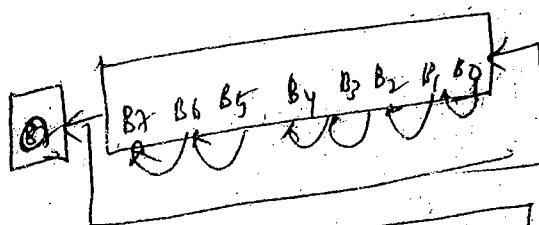
→

MOVE

⑭ Branch

J

jmp

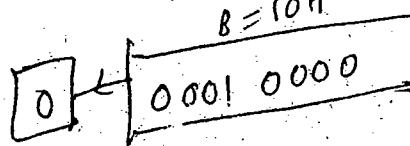
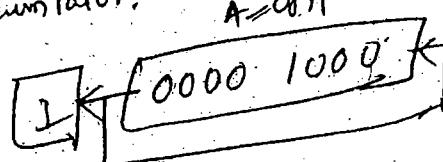


It acts as multiplied by 2
lower nibble to higher nibble and vice versa
it exchange

JC
f85 con.
otherwise

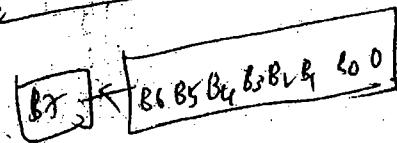
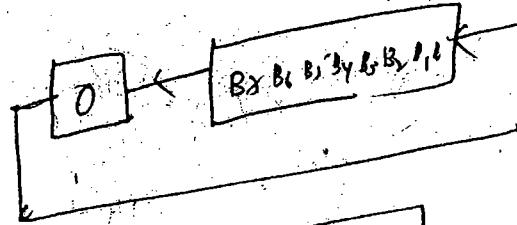
What is its consequences?

Q: If the content of R0 is A=0FH after rotation left & content of accumulator.



~~Q. RAL / ROL (rotate left with or without carry)~~

→ It is 9-bit rotation



II Branch (or) program execution transfer control instruction
4 types

↓
jump ↓
call ↓
RET ↓
RST

JMP
↓
16-bit Address

↓
unconditional
jump (①)
JMP

↓
jump

↓
jump (①)

↓
JMP 8000H

↓
Jump conditions jump
② goto

JC → CF=1

JNC → CF=0

JZ → ZF=1

JNE → PF=1

JM → SF=1

JP → SF=0

JM → SF=1

JHE → PF=1

Notes and references

* JC 4000
for condition
Otherwise

otherwise goes to other programs

it is a 1-byte instruction

JNL → ZF=0

JNP → SF=0

JNM → SF=1

JHE → PF=1

Q. Off all available jump instructions which are executed only once from the total memory required ms after IC? 8/10

87: 9 jump instructions so of 2x3 bytes
→ 27 bytes.

Q. Write a Assembly language program to find no. of even number and no. of odd number from a given n bytes number

88: If LSB is 1, ie odd no.
0, ie even number.

2000 MVI C, 08H

2002 MVI D, 00H D → to count odd no.

2004 MVI B, 00H B → to count even no.

2006 LXI H, 1000

2009 Y: MOV A, M

RRC

JC X;

2010 INC B

2011 TMP Z

2012 X: INC D

2013 Z: INX H

2014 DCR C

2015 JNL -Y;

2016 HLT

PC = 2019H

1000	29
1001	42
1002	34
1003	67
1004	89
1005	46
1006	35
1007	65

Q. Content

Address

203A

203B

203C

203D

203E-2

203F & C

2040. F

PC = 2041

→ (to find no. of +ve and no. of -ve numbers)

98 JMP 2 if Note that
B = 03H
D = 08H

B = 03H
D = 05H

Q. Content

program

and are extracted
from bytes 10.

to find no.05
from given bytes

9
L
4
3F
19
46
35
85

10F - ve
in hex)

=

If there

= 03H

= 08H

=

→ 815 Replace 16 RRE with RLC if used to take
no.0F +ve and no.0F -ve number

Q: After execution of the following program what is 16

Content in the accumulator.

<u>Address</u>	<u>Code</u>	<u>Mnemonics</u>
203A - 3E	3E, 20	MVI A, 20H
203B - 20	2A, 3A, 20	LHLD 203AH
203C - 24	86	ADD
203D - 3A	76	HLT
203E - 20		
203F - 86		
2040 - 86		

PC = 2041

$$AC = \boxed{203E} \rightarrow 20$$

$$M = 203EH \rightarrow 20$$

$$A = 20H$$

$$R = 20H$$

$$X = 40H$$

$$1068H$$

Q: Content of memory location after execution of PP8

program.	LXI B, 2080H	B = 2080H
	MVI A, 8FH	A = FFH
	MVI C, 68H	C = 68H
	SUB C + A = 8H	A = 8H → 1000 1111
	ANI 0FH + A = 0FH	C = 68 → 0110 1000 00100111 ↓ A = 8H
	STAX B →	
	HLT	

$A = 2FH \rightarrow 0\ 010\ 0111$

$20FH \rightarrow \begin{array}{r} 0000\ 1111 \\ -0000\ 0111 \\ \hline 03H \end{array}$

$A = 07H$

$BC = 2068 \quad \boxed{08H}$

2. If on execution of subroutine \$BX is given below, is executed
if on POPS BP the value in the accumulator after
execution of the following program.

\$BX: MVI A, 99H $\rightarrow A=99$

ADD 11H $\Rightarrow A=AAH$

MUV C, A $\Rightarrow C=AAH$

RET

$$\begin{array}{r} A = 1001\ 1001 \\ -0001\ 0001 \\ \hline 1010\ 1000 \end{array}$$

a) 00H b) 11H

g) 99H d) AAH

2. If on POPS BP

carry flag may be set by the instruction

a) PUB

b) INX

d) No flags

are affected

f) RST

c) CMX

d) ANA

e) CF=0

f) CF=1

are affected

1. If the following instructions are executed sequentially after the execution of following program what will be content of AC pair

XRA A

MOV LR

MUV H

INX H

DAD H

a) 0000H b)

g) 0001H b)

Q. On a)

B DE RF

Code is exa
following

LXI H

PUSH H

PUSH H

POP B

DAD SP

XCHG

HLE

HL=

DE=

- Q. A memory which of the accumulators

$\text{XRA A, } \Rightarrow A = 00H$

$\text{MOV L,A } \Rightarrow L = 00H$

$\text{MOV H,L } \Rightarrow H = 00H$

$\text{INX H } \Rightarrow H = 00H$

$\text{DAD H } \Rightarrow H = 00H$

$HL = 0000H$

0000H	0000 0000 0000 0000
0001H	0000 0000 0000 0001

i. is excited
or after

a) $0000H$ b) $0001H$

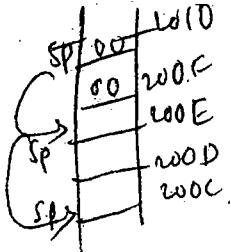
c) $0001H$ d) $0002H$

Q. On an 8085 CPU Value of 16 SP is $2010H$ and that
B DE register pair loaded with $1234H$ before 16 ex
code is excited. 16 value of DE register pair after 16 ex

$1001\ 1001$
 $0001\ 0001$
 \hline
 10101000

following program:

LXI H, $0000H \Rightarrow HL = 0000H$



push H

push H

pop B $\Rightarrow B = 0000H$

DAD SP $\Rightarrow HL = 0000H$

XCHG $\Rightarrow SP = 200EH$

$SP = 200EH$

HL > 200EH

HL = 1234H

DE = 200EH

By the instruction

a) ANI

CF \Rightarrow
always

ref

excited
program what

Q. A memory mapped I/O device as an address of $00F0H$.
Which of the following 8085 instruction will content of
the accumulator to the I/O device.

a) ~~box~~ I HOOFOY

MOV R1, A

b) 4×1 H, 00K0H p

out my first post Address

9 LFI H,00F04

our ~~folk~~

d) LFI H, OOF^D

Moving (~~go~~ going to
up)

Fig. 5E (20 mppd)
20 mm.

→ memory mapped I/O instructions

Q: The following is returned for memory mapped I/O ports 80P5H to add address 1FPEH and 1FFFH
 and completion of the above program, the result of addition
 two bytes located at memory

is found.
in the register.

a) $\text{at } 1000 \text{ K}$
 b) 1500 K c) 2000 K

$$y = \frac{1}{2} \cdot \text{IFP}$$

$$y = H_1 \cdot \text{IFCEH} \Rightarrow H_1 = \frac{y}{\text{IFCEH}}$$

$$\begin{array}{c} \text{LXII} \\ \text{H}_1 \\ 14 \\ \text{B}_1 \text{ N} \end{array} \Rightarrow$$

$$\text{MOV } B_1 \rightarrow L \Rightarrow L = FF$$

$$\text{PNR} \quad L \quad \Rightarrow \quad \text{ML} = \text{PEEP}$$

$$m_{UV} \stackrel{M}{\sim} A_1 M \Rightarrow$$

MUR ADD B AS OOH

ADD INR L \Rightarrow 00H

INR L →
MOV A, R →

MOV A, A ;
XRA A, A ;

1988.14
1988.14

KCZ/KOOG
7

CALL

calld:
↓ must be 16 bit Address

G TO

(P)

P TO ADD:

EH AND IPFFH

FOR ADDITION

ex: call 5000;

→ 3-byte instruction

→ call - it's type of interrupt; breaks normal sequence of execution.

→ call = push → Decrement stack pointer by 2
and push to current content of PC to stack.

ex 4000: call 5000

4001

4002

4003



Now PC = 5000

PC

W/Z

A/F

B/C

After completing call, it is

returned back to main program.



HLZ (R009)

unconditional conditional

call 3000:

CF=1

CF=0

ZF=0

ZF=1

SF=1

SF=0

PF=1

PF=0

...

1-bit register
- accumulator.

→ So at any time GR can serve one interrupt only.

→ If $D_7 = 1$, Set up data is enable.

Q. After execution of the RIM instruction, if content in the accumulator is E9H then what is the status of the recall data, and status of the maskable interrupt.

GR	$A = E9H$
$D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$	1 1 1 0 1 0 0 1

bit 8 interrupt
P data.

RST 5.5 is enable

→ If $D_3 = 1$ and $D_0 = 1$

RST 6.5 is disable

→ If $D_3 = 1$ and $D_1 = 0$

RST 7.5 is disable

maskable

and $D_2 = 0$

→ If $D_3 = 1$ and $D_4 = 0$

RST 5.5

is pending

locating priority system

otherwise

→ If $D_3 = 1$ and $D_5 = 1$

RST 6.5

is pending

provide disable

→ If $D_3 = 1$ and $D_6 = 1$

RST 7.5

is pending

RST 8.5

is pending

RST 9.5

is pending

RST 10.5

is pending

RST 11.5

is pending

RST 12.5

is pending

RST 13.5

is pending

RST 14.5

is pending

RST 15.5

is pending

RST 16.5

is pending

RST 17.5

is pending

RST 18.5

is pending

RST 19.5

is pending

RST 20.5

is pending

RST 21.5

is pending

RST 22.5

is pending

RST 23.5

is pending

RST 24.5

is pending

RST 25.5

is pending

RST 26.5

is pending

RST 27.5

is pending

RST 28.5

is pending

RST 29.5

is pending

RST 30.5

is pending

RST 31.5

is pending

RST 32.5

is pending

RST 33.5

is pending

RST 34.5

is pending

RST 35.5

is pending

RST 36.5

is pending

RST 37.5

is pending

RST 38.5

is pending

RST 39.5

is pending

RST 40.5

is pending

RST 41.5

is pending

RST 42.5

is pending

RST 43.5

is pending

RST 44.5

is pending

RST 45.5

is pending

RST 46.5

is pending

RST 47.5

is pending

RST 48.5

is pending

RST 49.5

is pending

RST 50.5

is pending

RST 51.5

is pending

RST 52.5

is pending

RST 53.5

is pending

RST 54.5

is pending

RST 55.5

is pending

RST 56.5

is pending

RST 57.5

is pending

RST 58.5

is pending

RST 59.5

is pending

RST 60.5

is pending

RST 61.5

is pending

RST 62.5

is pending

RST 63.5

is pending

RST 64.5

is pending

RST 65.5

is pending

RST 66.5

is pending

RST 67.5

is pending

RST 68.5

is pending

RST 69.5

is pending

RST 70.5

is pending

RST 71.5

is pending

RST 72.5

is pending

RST 73.5

is pending

RST 74.5

is pending

RST 75.5

is pending

RST 76.5

is pending

RST 77.5

is pending

RST 78.5

is pending

RST 79.5

is pending

RST 80.5

is pending

RST 81.5

is pending

RST 82.5

is pending

RST 83.5

is pending

RST 84.5

is pending

RST 85.5

is pending

RST 86.5

is pending

RST 87.5

is pending

RST 88.5

is pending

RST 89.5

is pending

RST 90.5

is pending

RST 91.5

is pending

RST 92.5

is pending

RST 93.5

is pending

RST 94.5

is pending

RST 95.5

is pending

RST 96.5

is pending

RST 97.5

is pending

RST 98.5

is pending

RST 99.5

is pending

RST 100.5

is pending

③ SIM: [Set Interrupt Mask]

→ 1-bit instruction

→ multipurpose

→ always operate on current accumulator.

Dx	D6	D5	D4	D3	D2	D1	D0
SOD	SDE	EST 7.5	X	MSE	M5.5	M6.5	M5.5

→ MSE - mask set Enable

D4 → UD

SDE → serial data Enable

SOD → serial output data.

, 15 D3 = 0 There is no control on D2-D1-D0.
S0 RST 7.5, 6.5, 5.5 are disable or maskable

RST 5.5 is enable

→ 90 D3=1 & D0=1

RST 6.5 is enable

→ 90 D3=1 & D1=1

RST 7.5 is enable

→ 90 D3=1 & D2=1

RST 7.5 is enable

→ 90 D3=0, but by using D5=1 RST 7.5 is enable

(There is no control on D2-D1-D0. If defined SOD is

→ 90 D6=0

disable

→ 90 D6=1 & D7=1 SOD is enable

→ 90 D6=1 & D7=0 SOD is disable

→ EE → Enable all /K maskable instructions

RST 7.5, 6.5, 5.5, INT

→ GF is 1-byte instruction

④ DI

→ (D)

→ 1-byte

→ EI on

⑤ NOP - r

→ 1 byte

→ It is

Note: NOP

→ In pr

fetched b.

because opcode of

⑥ HLT

→ 1-byte

→ exactly

→ GF STOP

at end

system

Instruction

execute

GF

(all the
possible ~~as~~ ~~most~~ gateway H)

byte instruction.

and DI be used for individual set or reset

NOP - No operation

1 byte
it is used to generate delay

~~NOP~~

in 8085 if NOP is the only instruction it is fetched from memory and it is decoded but it is not executed
cause opcode of NOP is - 00H.

Enable

every SLD is

⑥ HLT

→ 1-byte instruction

→ exactly opcode → 76H

MOV M, M
01 110 110 (X)

not possible 76H is allocated to HLT instruction

→ It stops the further process and keep the system in
the high impedance state (current flowing through
system 0)

Instruction cycle - It is defined as the time required to
execute one complete instruction.

It takes ^{min} 1 - machine cycle

and may 6 - machine cycles.

machine cycle

gt is defined as time required to execute

one complete operation.

- either read data from memory
- or write data to memory
- or read data from I/O device
- or write data into I/O devices.

It is take min 3 clock cycle and
max 6 clock cycles

(read lots
from
memory)

Note 1 1st machine cycle must be to opcode fetch

gt take min 4-clock cycles
max 6-clock cycles.

Note 2 During T₁, T₂, T₃ MP pre-fetch instruction from memory during T₄ to MP decode and execute

(i) If it decide how many byte instruction items what is its function of instruction.

MOV A,B

→ It takes 3-clock cycle for fetching

(ii) 1-clock cycle for config

for gt to run internal operations so it takes

clock cycle: clock cycle is its on and off time of 1st instruction

T₂ TON+TOFF

(or) Sub division of T-state

power s

address

data, IF

Av.

0	for	mm	?
1	to	16	?
2	prog	1	?
3	1	1	?
4	1	1	?
5	5	5	?
6	MVI	1	?
7	X: DCR	1	?
8	JNZ	1	?
9	REF	1	?
Y	Address	1	+
2	Data, IF	1	+
3	Av.	1	+

ed to crate

for the execution of instruction i.e. 4055 up take
 min 4-clock cycle and max 18-clock cycles.
 the no. of instruction cycles required for the following
 programs

code fetch
 (read data from memory)

instruction from
 and execute
 using terms

i) $MV A \#05H$ → 1 instruction cycle
 $X: DCR C$ → 5 times so 5 instruction cycles
 $JNZ X$ → 5 times so 5 instruction cycles
 RET → 1 instruction cycle
 12 instruction cycles

Address bus (d), pins 8

$A_8 - A_{15}$

it takes

line of 16

state

② Data lines / multiplexed 8

$A_0 - A_7$

power supply and clock signal

$V_{CC} = +5V DC$ (from)

GND — return power supply

Clock → provide to internal synchronization

RESETN or RS active low signal, get issued to reset READY

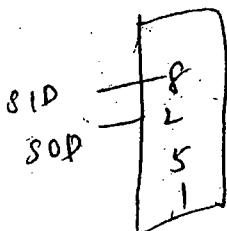
ie. 8085 MP

→ after Reset the content in PC is "0000H"

↳ Serial I/O Data Signals:

SIO — serial up data

SOD — serial down data



initiated signal

5) Externally

TRAP

RST 3.5

RST 6.5

RST 5.5

INT R

INTA

HOLD } for DMA controller.

HOLDA

KCD is for the priority signal

TRAP

RST 2.5

RST 6.5

5.5

INTR

RESET OR

RE

6) local

AL

D9

SO

10/10

0

0

0

1

1

1

To resetREADY: provide 1a. synchronization

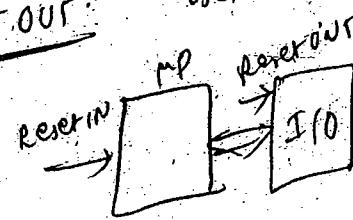
bln & I/O and

basic op.

RESET OUT

used to RESET

16 I/O devices.



(6) Sett Status & Control Signals:

ALE (Address latch enable)

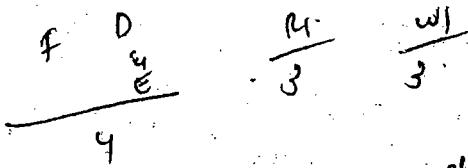
Distinguish & decide Address (d) Data.

so, si → decide machine cycle

IO/M = 0 → memory operation

= 1 → I/O operation

$\overline{RD}, \overline{WR} \rightarrow$	$\overline{RD}, \overline{WR} \rightarrow$	$\overline{RD}, \overline{WR} \rightarrow$	Control Signal	Name of 16 machine op.	No. of clock cycle req.
$IO/M = 0$	0 1	$\overline{WR} = 0$	\overline{MEMW}	memory write	3
0	1 0	$\overline{RD} = 0$	\overline{MEMR}	memory read	3
0	1 1	$\overline{RD} = 0$	\overline{IOW}	opcode fetch	4-min 6-max
1	0	$\overline{WR} = 0$	\overline{IOW}	I/O write	3
1	1 0	$\overline{RD} = 0$	\overline{IOW}	I/O read	3
1	1 1	$\overline{INTA} = 0$	-	Interrupt Acknowledge	6



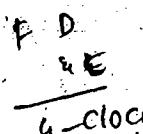
$\Rightarrow 10 \text{ clock cycles}$

Ex: $10000 \text{ op JMP } 8000$ (Control Operation)

$1001-00$ 3-byte

$1002-80$

$PC = 1003$



Need data from 1001 memory location $(M) - 3$

Need data from 1002 "

$4+3+3 = 10 \text{ clock cycles}$

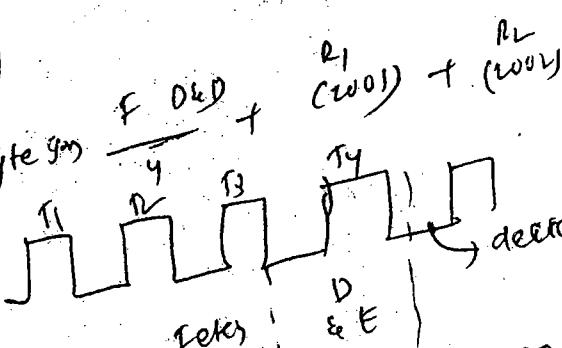
$3: 1000 JC 9000H$

$1001-00$

$1002-90$

$PC = 2003$

3 byte op



after $T_4 \rightarrow B_F$ is unconditional jump

$$\frac{R_1}{4} + \frac{R_2}{3} = 10 \text{ clock}$$

B_F (condition) is not satisfied. It will go to

next instruction so it will take 7 clock cycles

$10/2$ clock cycles

09

= PM to
memory

eg: XTHL

1000-OP 1+ byte instruction HC = 1284H

F D&E
4 clock

$$\frac{R_1(05) + R_2(06)}{3} + \frac{W(09) + W(09)}{3} = 16$$

16
15 OT
14 OS
13 OS
SP 12 OT

= 16 clock

eg: 1000-OP RET

F & D&E
4 clock

$$\frac{R_1}{3} \frac{R_2}{3}$$

= 10 clock

RET = POP

eg: 2000
2001
2002
PC = 20

eg: ORA B → internal operation

80 4 - clock cycles

→ internal operation

eg: MVI

C, 40H

5000-OP

5001 - 40H

PC = 5002

F D&E
4

R1(5001)

= 7 clock cycle

→ internal (6 opp)

LXI

B, 1234H

eg: 6000-OP

6001 - 34

6002 - 12

F D&E
4R1(6000)
R2(6001)

= 10

eg: 1000-OP DAD B

F D&E
4

HC = 1289H

BC = 1585H

Bus idle statBus idle stat

+3

+3

= 10 clocks

eg:

Q: 10
of 16 :

I

S.

4 E

eg: 4

6000

6001 -

T6
 15 OR
 14 OR
 13 OR
 12 OR

DAA → 4 clock
4 byte

SBB B → internal operation
→ 4 clock cycle

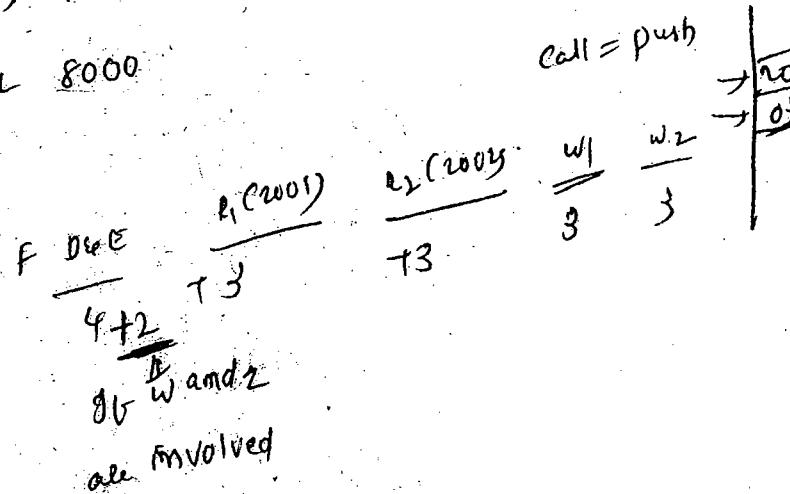
ADC B → 4 clock cycle.

e.g. 2000-00 CALL 8000

2001-00

2002-80

PC = 2003



e.g:

DCX B

16-58

DCL B

8.68

6 clock cycle

Q: (a) No. of memory cycles required for the execution of the following instruction.

I. LD A 4000H

II. IN 20H

(a) 4 for I 9-2 8W II 5) 3 over I & 3 over II

4 for I

3 BW II

4 over II

4 over II

6000H IN 20H

6001-20H + DCE

4
3

4
3

f2 (read data from key board to MP)

MP

→ 11 clocks

calculate the approximate delay generated by 16 loop instructions, exact delay generated by 16 loop instructions and total delay generated by its program.

Step 2:

MVI C, 40H — execute only one — 2 clock cycles

X: DCR C] 100p instruction
IN2 X → 4 clock cycles

LET — execute only one → 12 clock cycles

$T_{LA} = \text{approximate delay generated by loop instruction}$

Si:

Step 3:

$T_{LA} = n \times T_{N10}$

$n = \text{no. of clock cycles required for execution}$

of loop for one

cycles

$n = 14 \text{ one}$

$T \rightarrow \text{system clock time}$

$= 0.5 \mu s$ [8085 operate @ MHz freq.]

$N_{10} = \text{equivalent decimal value of counter.}$

$$= 4 \times 16^1 + 0 \times 16^0$$

$$\Rightarrow 64$$

$$T_{LA} = (14 \times 0.5 \mu s \times 64) \mu s$$

$$\Rightarrow (7 \times 64) \mu s$$

$$\Rightarrow 448 \mu s$$

Now

Q: 8085
el 1

81:

Ans:

83)

generated by 16
2nd by 16 loop
by 16 program.

1a

loop instruction

2d for execution

clock
free]

of counter.

Step 2: exact delay generated by loop instruction T_L

$$T_L = (T_{EA} - 3 \times 0.5) \mu s$$

JNZ → 63 fine grfs
only one if is
conditioned
on condition
~~C10-7~~ clock → 3 clocks

$$= (448 - 1.5)$$

$$= 446.5 \mu s$$

Step 3: total delay $T_D = T_p + T_L$

generated outside
loop instruction

[MVI C, 40H]
NET

$$T_p = (18 \times 0.5) \mu s = 8.5 \mu s$$

$$T_D = (8.5 + 446.5) \mu s = 455 \mu s = 0.4 \text{ ms}$$

Q: How to get 16 max delay.
max value $c = FFFF$ (forget max delay)

80: M1 load C with

$$T_{LA} = n \times T_p N10$$

81: $n = 14 \quad T = 0.5 \mu s$

$$N10 = 15 \times 16^1 + 15 \times 16^0 \quad (\text{This is only charge})$$

$$= 240 + 15$$

$$= 255$$

$$T_{LA} = 1885 \mu s$$

$$T_L = (1885 - 1.5) = 1883.5 \mu s$$

$$T_D = T_p + T_L = 1798 \mu s$$

max delay 1.5 μs

83

→ To increase delay Add 1-NOP instruction

MVI C, ~~PFH~~ → 7 clock cycles

X: DCR C] → 4
 NOP] 3 loop - 9
 JNZ X] 8 memory - 10/8
 RET] → 10

81: $T_{LA} = n \times T \times N_{10}$
 $n = 18 \quad T = 0.5 \mu s$

$N_{10} = 255$

$T_{LA} = 29.95 \mu s =$

$T_L = 29.95 + 29.95 \times 5.18 = 230.7 \mu s = 2.3 ms$

$T_D = 8.5 + 22.93.5 \mu s = 23.02 \mu s = 2.3 ms$

Diff = $23.02 - 2.3 = 20.72 \mu s = 0.5 ms$ delay.

So one NOP can generate 0.5 ms delay.

after adding 10-NOP delay = $1.8 + 0.5 \times 10 = 6.8 \mu s$

They are two methods to increase 10x delay.

y) register pair

y) loop ~~instruction~~ within the loop.

y) register pair method

LXI B, 1234H → 10

DCX B → 6

MOV A, C → 4

ORA B → 7

39

$$jw_2 - i \leftarrow -10/2 \\ \text{NET} = 70$$

89:

$$TLA = N \times T \times NIO =$$

$$N = 24 \quad T = 0.5 \mu s$$

$$NIO = 1 \times 16^3 + 2 \times 16^2 + 3 \times 16^1 + 4 \times 16^0 = 4660$$

$$NIO = 55920 \mu s = 55.92 \mu s = 55 \mu s$$

$$TLA = 55920 \mu s = 55918.5 \mu s$$

$$TL = (55920 - 1.5) = 55918.5 \mu s$$

$$TD = T0 + TL = 55928.5 = 56 \mu s$$

$$T0 = (20 \times 0.5) \mu s = 10 \mu s$$

$T0$ has max value (PPH)

→ Load BC register pair with

LXI B, FFFFH

$$NIO = 65535$$

$$TLA = 786420 \mu s$$

$$TL = 786428.5$$

$$TD = 786428.5 = 785.5 \mu s$$

delay

$$0.5 \times 10$$

dy.

→ odd

1-NOP instruction

$$\text{Now } N = 28 \quad T = 0.5$$

In 100P. to increase delay.

$$NIO = 65,535$$

$$TLA = 918490 \mu s$$

$$TL = 918498.5 \mu s$$

$$TD = 918489.5 \mu s$$

$$918489.5 \mu s - 786428.5 \mu s = 131000 \mu s = 131 \mu s$$

or 131 ns are generated by 1-NOP 8 ns

After Adding 10-NOPS $(10 \times 131 \mu s) + 786.5$

$$2096.5 \mu s$$

10

6

4

7

II - 10pp with 16 loop method:-

29, 28, 27 - steps (arrange ascending order)

$B = (n-1) = 0FF$ — outer counter

$C = 02H$ — inner counter.

1000 1001 1002

29, 28, 27
compared = 01

28, 29, 27
 $c = 00$

28, 27, 29
if $c = 00H$ (then only B decrements)
 $B = 01H$ so again loaded with same value
 $c = 02$

C202H 1000 1001 1002

e.g. 28, 27, 29
 $c = 00H$

28 28 29
 $c = 00H$

28 28 29
 $c = 00H$ (from)

$B = 00H$

MVI B, 3FH

$y = \text{MVI } C, 0FH$

X : DCR C] inner loop
JNZ X inner loop

DCR B — q

JNZ Y \rightarrow 10/8 q 100 ns

NET

Outer loop
total delay generated by
per program

b) 100.5 ms y 1000 ns

c) none of these

82:

83:

Q. A 6

below

Delta

t

ts

nding order)

approximate delay generated by 16 minor loop instructions

$$T_{LA1} = N \times T_{XN10}$$

$$\eta = 14 \quad T = 0.5 \text{ ms} \quad N_{10} = 255$$

$$T_{LA1} = 1785 \text{ ns}$$

$$T_L = (1785 - 1.5) = 1783.5 \text{ ms}$$

82. exact delay $T_L = 1783.5 \text{ ms}$

approximate delay generated by out loop instruction

83:

$$T_{LA} = 56 (T_{L1} + 21 \times 0.05 \mu\text{s})$$

$$3 \times 16 + 8 \times 16^{10}$$

$$= 56 [1783.5 + 10.5] \mu\text{s}$$

$$= 4818 = 56$$

$$= 100464 \mu\text{s}$$

$$T_L = 100462.5 \text{ ns}$$

$$T_D = [100462.5 + 8.5] \mu\text{s} = 100471 \mu\text{s}$$

$$= 100.5 \text{ ms}$$

Q. A software delay subroutine is written as given below. How many time deereament instruction excited

Delay: MVI H, 255D = PPH

MVI L, 255D = PPL

loop1: MVI

loop2: DCR L

JNZ loop

255×25

$\Rightarrow 63750.5$ times

DCR H

JNZ loop1;

ns (1) 1000 ns

of the asse

An 8085 Assembly language program is given as follow.

- Q. The execution time of each instruction is given against the instruction in terms of T-states. Find the execution time of the program in terms of T-states?

Instruction	T-states
MVI B, 0AH	3T → 1 Byte 7T × 10 (Any)
LOOP: - MVI C, 05H	4T × 10
DCR C	4T × 10
DCR B	4T × 10
ZNZ2 LOOP	(10/2)T
	3T → 1 Byte 7T + 30T + 40T + 40T + 9 × 10T + 1 × 25 = 274T
	254T

a) 245T b) 250T c) 254T d) 258T

$$T_{LA} = M \times T \times NIO$$

$$= (25 \times 10)T$$

↓
0AH

$$T_L = (25 \times 3) = 75T$$

$$T_D = M \times T$$

$$= (2 \times 25)T$$

→ 254T answering language program after

Q. Consider the following 8085 program. What is the result of executing it below?

MVI B, 89H → B 89

MOV A,B → A = 89

Start: JMP Next

MVI B, 00H

Addresing

~~left~~: 16

effed 64

immmedia

→ 8 or 16-bit

or 6-bit

or 6-bit

mem as known
given against
to execute it

~~XRA B~~ \Rightarrow ~~out port 1~~

Next: XRA B \Rightarrow A20CH

JP start
out port 2
HLT.

a) an old op for out port 1.

b) an old op for out port 2

c) immediate of looping as program execution with

accumulator data remaining at 00H

d) immediate looping of program execution with accumulator data

e) immediate looping of alternating 0/1 00H and 80H

$$\begin{array}{r} 00 \\ \oplus 80 \\ \hline 1000 \text{ 1111} \end{array}$$

so
 \Rightarrow out port 2

op at port 2

Addressing modes:

↳ In different ways the processor can access data either from memory or I/O devices.

Immediate Addressing mode:

→ In this addressing mode source must be either 8-bit or 16-bit immediate data and destination must be either 8-bit or 16-bit internal register.

ex: MVI A, 40H

LXI B, 1234H

SUB 2017

ADI 40H

NRI 30H

Implied (a) Implicitly:

→ In this Addressing mode both source and destination are implied.

eg. STC (CF=1)	D1	RET	RAR
CMC (CF=0)	E1	XCHG	RET
CMA	SIM	XTHL	LLC
	RIM	DAD	NOP

III Registered Addressing mode:

→ In this Addressing mode both source and destination must be either 8-bit or 16-bit internal register.

eg. A MOV A1 B
ADD C
OR A B

SUB C
Direct (a) Registered Addressing mode:

→ In this Addressing mode either source or destination must be either 8-bit pur 16-bit Address.

eg. IN 50H
OUT 60H
CDR 4000H
STA 9000H
LHD 300

CALL 4000

JMP 8000

IV Implied Addressing mode (a) Register R. A. D:

→ either source (a) destination can be specified by using Register pair

Q. Q.

staller

extenar

such

a way no

mas

No?

M?

what

to MF

Memory

mapping

GF is no

for Decr

Data

MOV A

ON

ADD

Q3

Q. after Execution of the following program what is the

Content in the accumulator:

3000H MVI A, 00H

3000H call sub1 (3005)

3005 806: INL A → 01, 02H

RET

after last instruction pc load with 3005

at 3005 (NN) so A is incremented

one program so $A = 02H$

RET

RET

on conditional

①

Conditional

⑧ graph from C

CF=1

RC

CF=0

RNC

ZF=1

RZ

ZF=0

RNZ

ZF=0

RPE

PF=1

RPO

PF=0

RP

SF=0

SP=1

RET is a 1 byte instruction

RET = pop instruction

in 605 Call and RET instruction

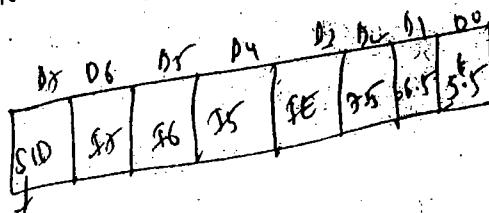
vice and definitions also

implies those 12 "program counter" am

implies

Machine (a) processor instructionsD RIM (Read interrupt mask)

- no operands only opcode so it is a 1 byte instruction
- if it is for multipurpose
- if it is always operate off the content of the accumulator



serial up data serial up data (markable interrupt)

multipurpose: either enable or disable

IE - interrupt Enable

S1D → serial up data

no control on 7.5, 6.5, 5.5

→ 01 03=0

RST 7.5, 6.5, 5.5 are disable or markable

if default

RST 5.5 is enable otherwise disable

→ 01 03=1

and D0=1

RST 6.5 is enable otherwise disable

→ 01 03=1 and D1=1

RST 6.5 is enable

→ 01 03=1 and D2=1

RST 7.5 is pending

→ 01 04=1

RST 5.5 is pending

→ 01 05=1

RST 6.5 is pending

→ 01 06=1

RST 7.5 is pending

→ 01 03

→ 01 04

→ 01 05

→ 01 06

→ 01 07

→ 01 08

→ 01 09

→ 01 0A

→ 01 0B

→ 01 0C

and destination

eg:

LADX D	push
STAX B	pop
MOV A, M	
MOV M, B	

and destination
+ register.

Q. In a MP the service routine for a certain interrupt starts from a fixed location of memory which can not be externally set but if interrupt can be delayed or rejected such interrupt is.

a) Non maskable - Non vectored memory

b) maskable - "

c) Non maskable - Vectored

d) maskable - vector

Q. what are different ways in which I/O and I/O devices are connected to MP.

i) memory mapped I/O

Assign Address to

memory mapped I/O

or is required 16-bit address lines

for decoding

Data transfer instructions

MOV A, M, MOV M, B

LDI 4000H STA 8000H

- - - - -

ii) I/O mapped I/O.

memory or I/O devices

iii) I/O mapped I/O

or address lines for decoding

iv) IN/OUT.

5 bit/16 bit

register pair

clock speed

 $m\mu \rightarrow$ clock cyclemax \rightarrow 13 clock cycles

4) control signals generated

 \overline{MEMR} & \overline{MEMW}

5) Direct Arithmetic operation

is possible in 16 memory

mapped I/O.

e.g. ADD^M, ORA^M.6) transferred as data $\&$ only
registered and memory7) more hardware complexity
required to decode

16 - Address line

more power consumption
at RAM

10 - clock cycles.

4) $\overline{IEN} \& \overline{IOW}$

5) it is not possible.

6) only accumulator and
I/O device.

7) less hardware complexity.

less power consumption

4P 200

AT -

AT -

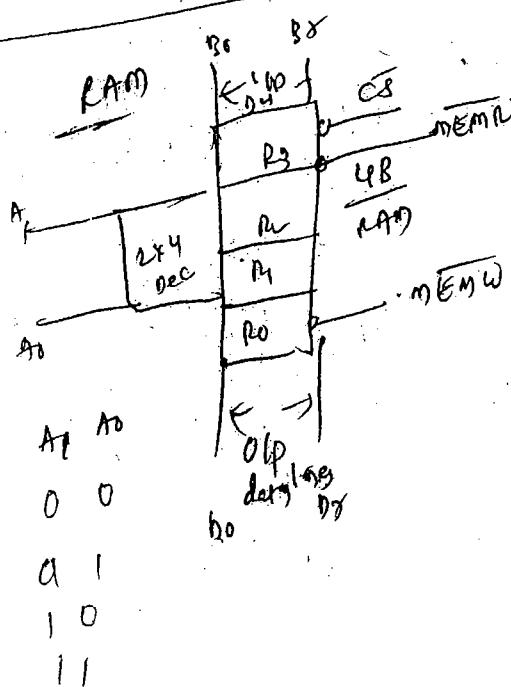
char

complex

8. Design

step 1

Step 2



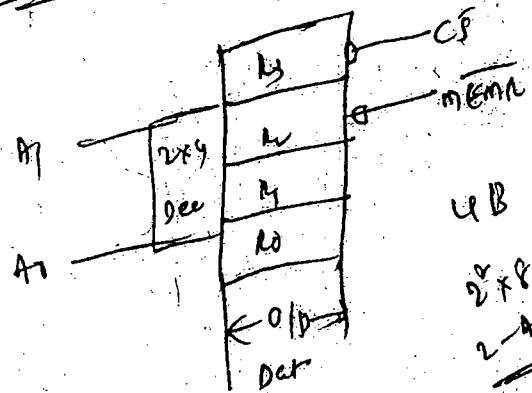
MB

N X 8

2 Address

4P ROM

read only so only op data lines no update

so there is no MEMW

A1	A2	A3	A4
0 0	→ R0		
0 1	→ R1		
1 0	→ R2		
1 1	→ R3		

4B
2x8
2 Address

change the memory mapping by changing the hard

factor and

circuitry.

ie.

Q. Design 256 Bytes of ROM and find its memory mapping

circuitry.

Q1) Step1: find no. of Address lines

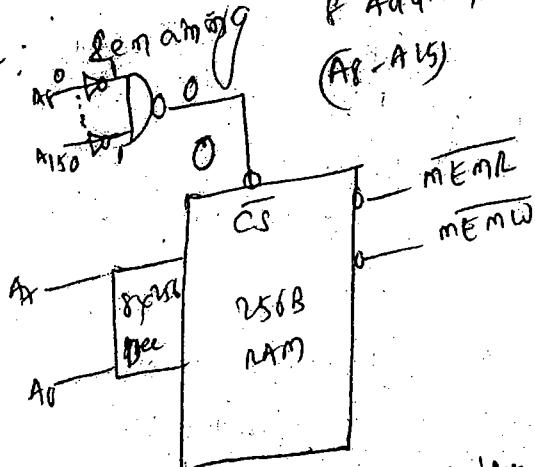
$$256 \times 8 = 2^8 \times 8$$

8 Address lines $(A_0 - A_7)$ + Decode

8 Address lines are used to Select Ch1

excepts

Step 2: Segmenting



A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

on for CS

= 0000H

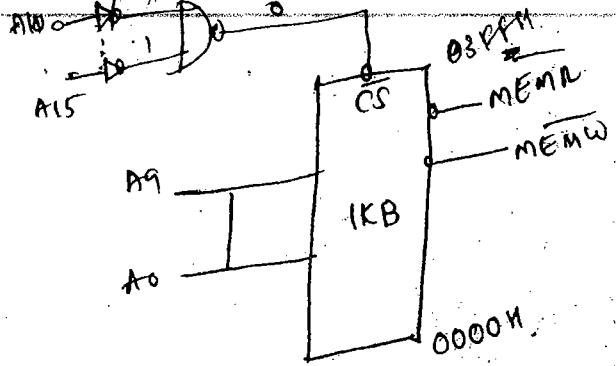
all are A8 to A15

= FFFFH

0 256 RAM =

= 0000H

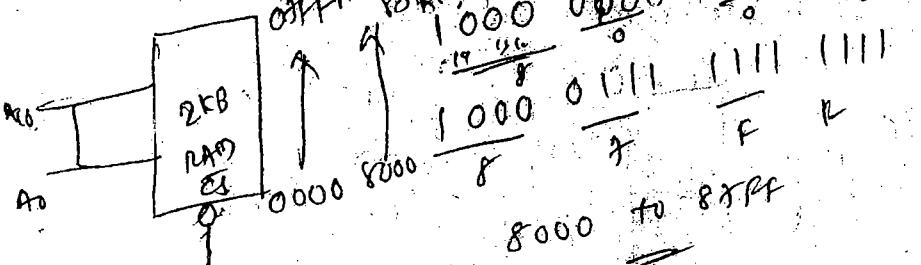
Design, 1 KB of memory and find its memory mapping



S: 1 . . . $2 \times 2^{10} \times 8$ 16-to-8 decoder the Ada. (A₀-A₉)

3) 10 Address
correct 16 chip.

Q. for the following okr find the answer



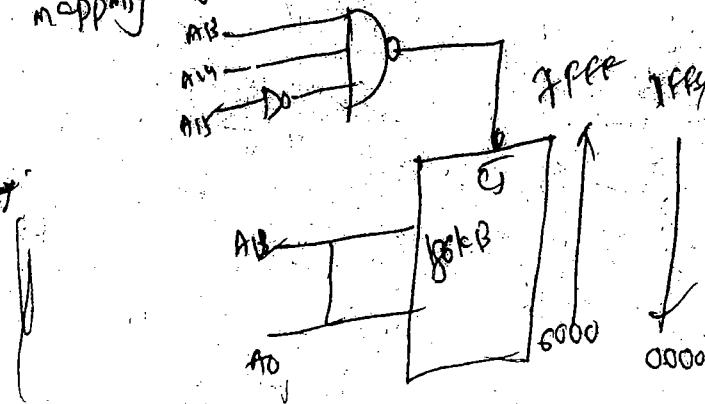
A₁^o Do
A₂ to
~~A₃~~ S
A₁₅ (overhead)
carried

carroted
found it ready
in RAM 16KB

$$P_k = 2^3 \times 2^{10} \times P_{key}$$

213 x 8

19 address file



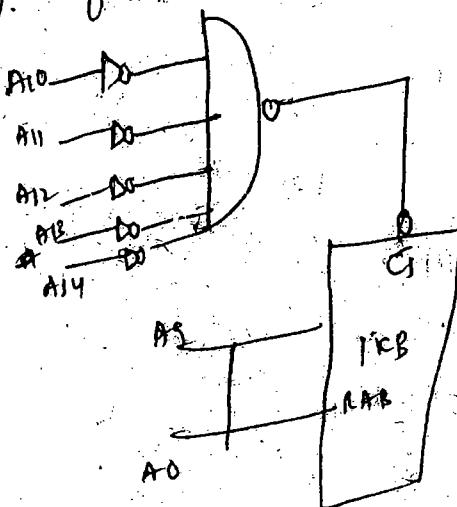
16 memory mapping

$\overline{A_15}$

$0\ 110$	0000	0000	$0000 \rightarrow 0000$
$0\ 111$	1111	1111	$1111 \rightarrow FFFFH$

Q. 6-bit $A_{10} - A_4$ memory mapping of the following ckt.

Add. ($A_0 - A_9$)



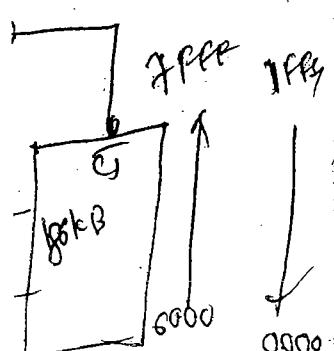
EA, M
000

$\overline{111}$
 $03FFH$

off mapping

$0000\ 0000$
 $0\ 0$
 $1111\ 1111$
 $F\ F$
 $8FFF$

say out char



$\rightarrow A_{15}$ is not connected. Assume $A_{15} = \text{don't care}$. either (0, or 1)

$\overline{A_{15}=0}$ $A_{15}=A_{14}, A_{13}, A_{11}, A_{10}, A_9, A_8$
 $0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$ $0000\ 0000 \rightarrow 0000H$
 $0\ 0\ 0\ 0\ 0\ 0\ 0\ 1$ $1111\ 1111 \rightarrow 03FFH$.

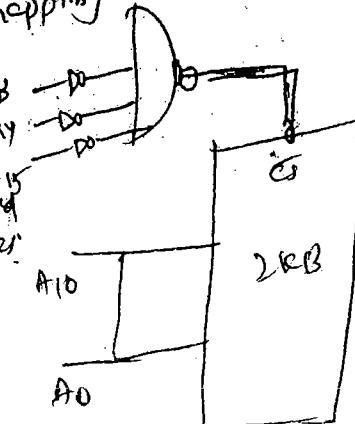
$\overline{A_{15}=1}$ $1\ 0\ 0\ 0\ 0\ 0\ 0\ 0$ $0000\ 0000 \rightarrow 8000H$.
 $1\ 0\ 0\ 0\ 0\ 0\ 1\ 1$ $0011\ 1111 \rightarrow 83FFH$

0000 to 03FFH (a), 8000 to 83FFH, such is not a decoding technique

decoding - it is a linear decoding mapping of the following ckt. show

memory 2kB
 $A_{11} \& A_{12}$ are not connected

$A_{11} \& A_{12}$ are don't care
 00
 01
 10



~~A15 A14 A13 A12 A11~~

$$\begin{array}{r} 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\ \underline{0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0} \end{array} \quad \begin{array}{l} 0000 \\ 0000 \end{array} = 0000H$$

~~if $A_11 = 0 \ A_{12} = 1$~~

$$\begin{array}{r} 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \\ \underline{0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0} \end{array} \quad \begin{array}{l} 0000 \ 0000 \\ 0000 \ 0000 \end{array} = 0800H$$

$$\begin{array}{r} 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \\ \underline{0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0} \end{array} \quad \begin{array}{l} 0000 \ 0000 \\ 0000 \ 0000 \end{array} = 0FFFH$$

~~if $A_{11} = 1 \ A_{12} = 0$~~

$$\begin{array}{r} 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \\ \underline{0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0} \end{array} \quad \begin{array}{l} 0000 \ 0000 \\ 0000 \ 0000 \end{array} = 1000H$$

$$\begin{array}{r} 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \\ \underline{0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0} \end{array} \quad \begin{array}{l} 0000 \ 0000 \\ 0000 \ 0000 \end{array} = 1000H$$

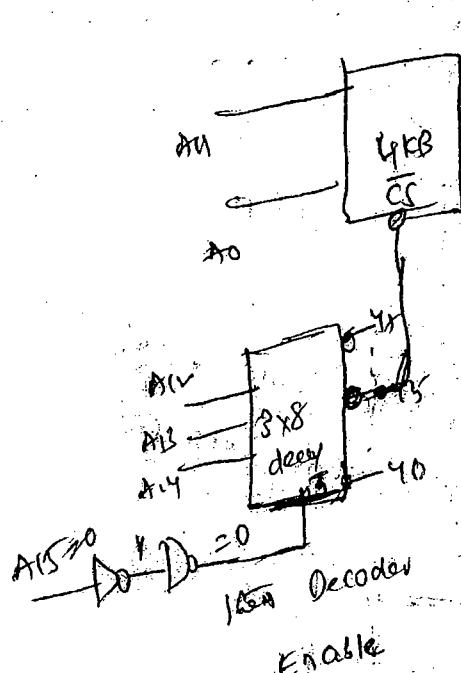
~~if $A_{11} = 1 \ A_{12} = 1$~~

$$1800 \rightarrow 91FFFH$$

min 0000 and max range

mapping of the following ckt

Q. find the memory



1800 \rightarrow 91FFFH
 $5000 \rightarrow 5000H \rightarrow 5FFFH$
 as enable when q5 selected

A15 A14 A13 A12

0	0	0	Y0
0	0	1	Y1
0	1	0	Y2
1	0	1	Y3
1	1	1	Y4

1800 A15

A14 -
A13 -

min A13

A13
0
0
0
0

Q. find

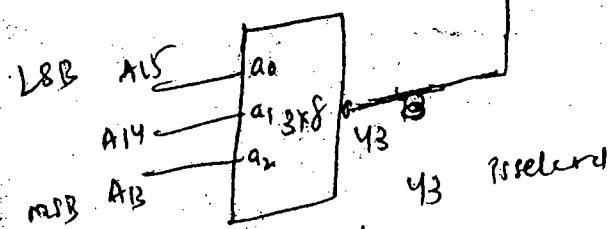
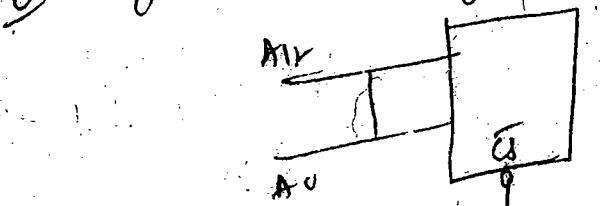
A15
A14
A13
A12

00001

A15
0
0
A12

Q. find

Q. Find 16 memory mapping of the following ckt.



when

$$A_{13} A_{14} A_{15} = 011$$

$$\begin{array}{cccccc} & & & & & \\ A_{15} & A_{14} & A_{13} & A_{12} & 0 & 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\ 1 & 1 & 0 & 1 & 1 & 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 \\ \end{array}$$

$$\Rightarrow A_{12} 0000 \text{ to } 0 FFFH$$

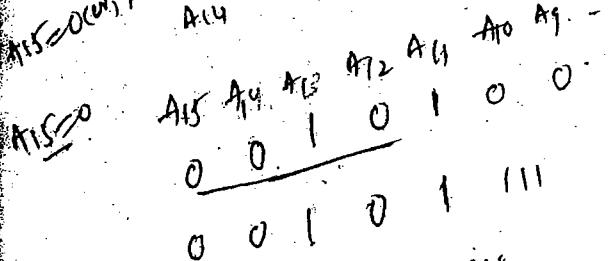
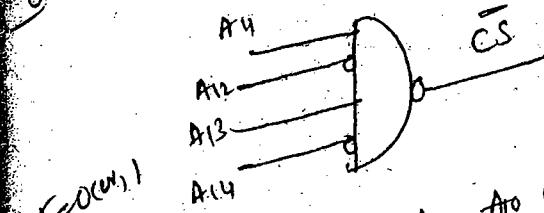
of the following ckt

(we have to assume A0-A10
as don't decoding)

A0
0000 0000
1111 1111

FFFF
1111 0000
selected

Q. Find 16 memory mapping



$$\begin{array}{cccccc} & & & & & \\ A_{15} & A_{14} & A_{13} & A_{12} & 0 & 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \\ 1 & 1 & 0 & 1 & 1 & 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 \\ \end{array}$$

10.

11.

2

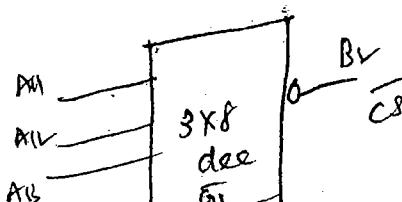
15

17.

~~A15=0~~ A800 to AF_{FFH}
mapping of 16 following ckt

Q. Find 16 memory mapping

of 16 following ckt



10 ready

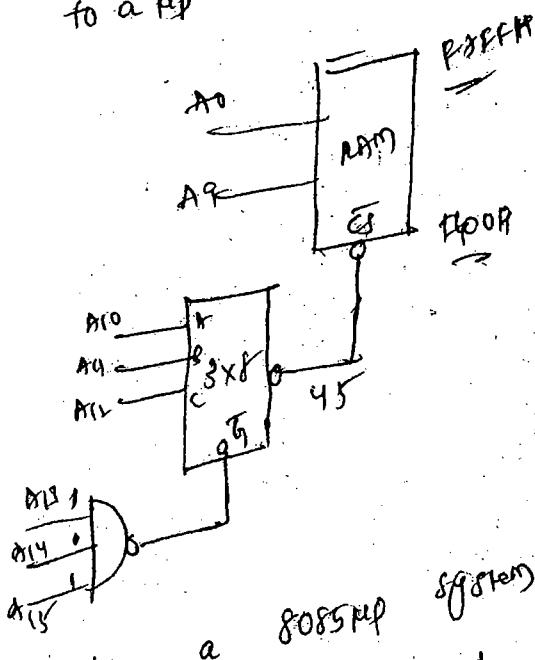
11 ready

A₀ - A₁₀ decode the address.

A ₉	A ₁₂	A ₁₃	
0	0	0	B ₀
0	0	1	B ₁
0	1	0	B ₂
1	1	1	B ₃

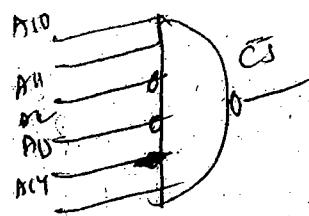
A₁₅ A₁₄ A₁₃ A₁₂ A₁₁
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 = 1000
= FFFFH

- Q. The range of P₁ as shown in the fig which is interfaced to a 4K RAM.

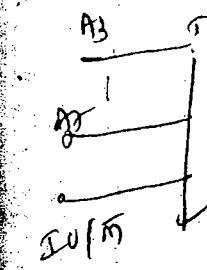


A₁₅ A₁₄ A₁₃ A₁₂ A₁₁ A₁₀
1 1 1 0 1 00 = F400
1 1 1 1 0 1 00 = FFFFH

- Q. An a 8085MP system is generated by passing Address 1659 A₁₀-A₁₅ through a 64P NAND gate for selecting 16K range of NAND gates. The active low chip select signal is generated through a 64P NAND gate. The output of NAND gates are:



1100 1100 0000 0000 → CC00
1100 1100 1111 1111 → CCFF



Q A2=1

Q Draw

Q 16 CPU

0 0000 0 1000
= FFEE
=

Each IC interfaced

$\Sigma_0 = F400$
FFEH
=

chip select
in 16 bits A10-A15

using 16 Address
gates are
10 bits A10-A13 A14-A15
10 bits A10-A13 A14-A15

A10-A13 A14-A15
10 bits A10-A13 A14-A15

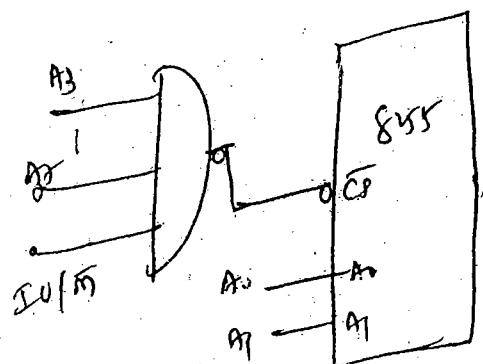
:00

:FF

An 8255 chip is interfaced to an 8085MP as
also 80 mapped IO as shown in (E69) 16 Address
off all A0-A7 of 8255 connected to Address 80 lines.
New Address lines are used to select 16 ports
8085MP control word register. 16 Address lines A3 to A7
as well as 16 signal are used

Address decoding. 16 range
16 Address for which 8255 will
get select is?

A7 A6	
0 0	port A
0 1	port B
1 0	port C
1 1	port CWR



- a) F8-FBH
- b) F8H-FCH
- c) F8H-FFH
- d) F0H-FTH

A2 is not connected

A7 A6 A5 A4 A3 A2 A1 A0

1 1 1 1 1 0 0 0 = F8H to max is FFH
1 1 1 1 1 0 1 1 = FBH

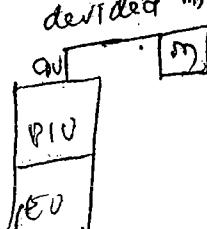
so 16 range is
F8-FBH

1 1 1 1 1 1 0 0 = FC

1 1 1 1 1 1 1 = FF

Q Draw 16 BLOCK Diagram for 8085 and explain.

by 16 CPU 015 8086



→ Devide to work b/w two units and increase the speed of the processor.

for BIOS to fetch instruction from memory & read

data from memory or I/O device.

(ii) write data into memory or I/O device.

(iii) get handle of transfer of Data for the exceptions

unit

BUS EU:-

it tells where to fetch the instruction.

MOV AC(4000) most of register is 16-bit

physical Address = 2031t
4000 is effective address

Physical Address = Total Base Address + effective Register

content in it segment is base address

→ 1st content

D₀ = 4000

place value of address to D₀ D₁ D₂ D₃ = 400000
effective = 4000
PA = 440000

and execute

it decodes the instruction

EU:-

y control

AC

operand

flag

General

Special

and

of word

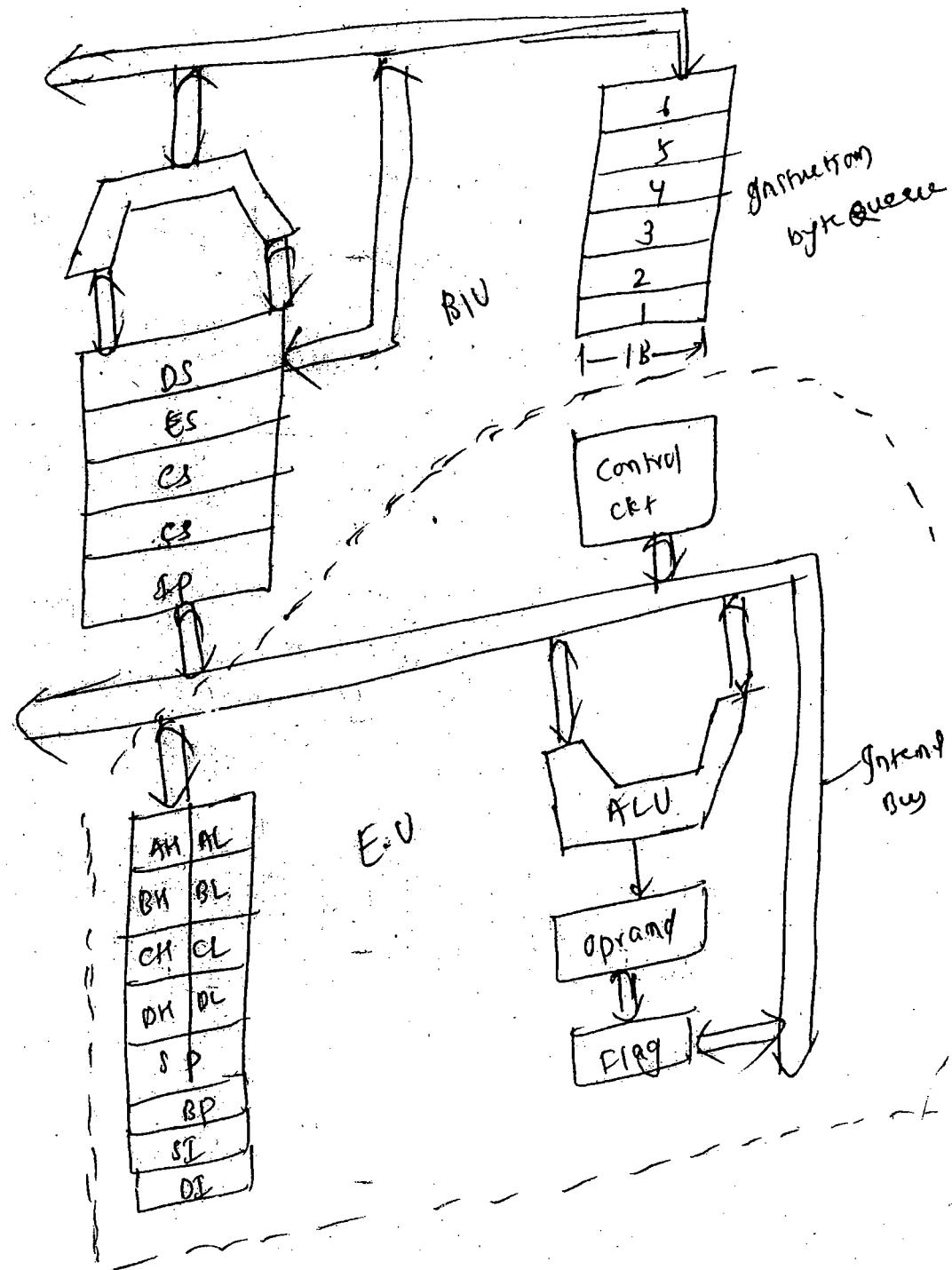
for example

instruction

in 16-bit bus

effective
use Address

10.
500
000M



EU:

- y Control system
- y ALU
- y Operand
- y Flag register
- y General purpose regn
- y Numerical regn

y) Control Clr direct & internal operation

3) ALU: It can perform arithmetic, logical and shift operations on bits, nibble, bytes or words.

③ Operand: It is 16-bit register. It holds the immediate data.

16-bit Data

MOV AX, 4000 → immediate data

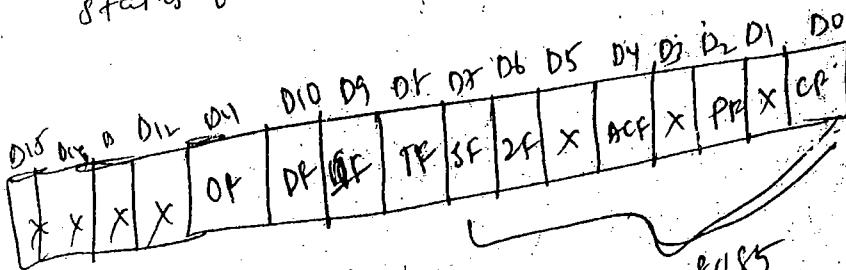
MOV AX [4000] → address

data from memory [4000]

④ Flag Register: It is 16-bit register. It is also called PSW register. It always indicates the current

status of ALU. It has 9-active flags, 6 are condition

& 3 are control flags



Conditional (C) Status

control flag

trap flag

interrupt flag

Direction flag

y) CF

y) PF

3) ACF

y) ZF

y) SF

y) OF

DF

Trap flag

8 - 8

CF → from DS

OF → OF & DS

CY

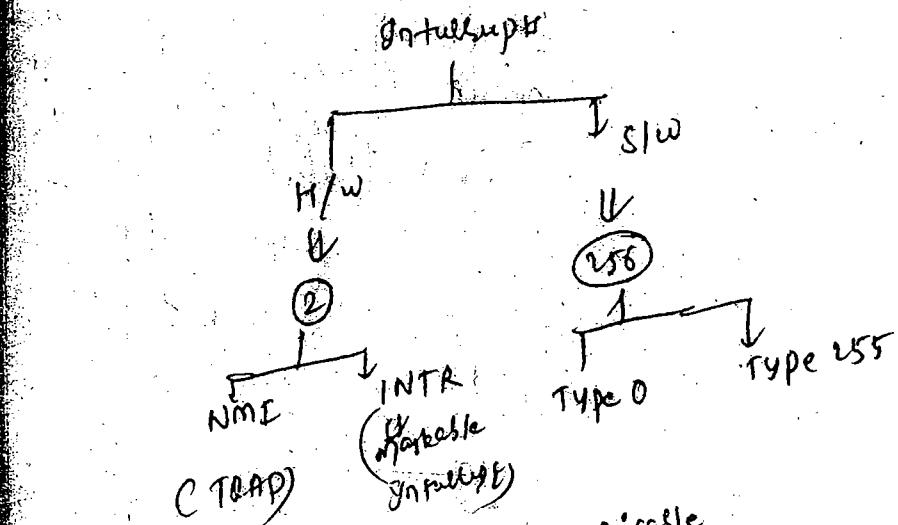
nd SIFMANS

WDS.

PC ~~simonidek~~

2^{ms} [6000]

110 called
current
as condition
not flags



\rightarrow IF = 0 INTR is disable
IF = 1 INTR is enable

Direction flag get using STD & CLD instructions. This is
mainly used for string instructions

CLD (Clear direction flag)

get (to direction flag) STD
DF = 1
IF

SI ↓ increment
DI ↓ decrement
one (or) two
byte transfer
depends on — word transfer

SI ↑ increment
one (or) two
byte
word

\rightarrow DF for auto increment (or) Decrement to SI & DI.

Trap flag: Trap is for step by step execution.

8-bit general purpose register for temporary storage of data

AH, AL, BH, BL, CH, CL, DH, DL

Car, Row 4 16-bit register AX, BX, CX, DX

AX — data is 16-bit accumulator

BX — base pointer

CX — counter

DX — data pointer

NewtonDesk.com

→ 48 bit special purpose register. Itself hold 16 effective address. → free
 SP - stack pointer Address (or) offset - Address can pair as cursor
 BP - Base pointer → effective
 SI - source index → mem
 DI - destination index → easy
 ex: mov AL, [5000]
 ↓
 Effective Address
 (or) offset Address
 → 8b
 → 8b

(B10):

i) Segment Register

ii) IP

iii) Hardware zero.

iv) Instruction Byte Queue

v) It has 20 Address bits

Addressing capability = 2^{20}
 $= 2^{10} \times 2^{10}$

$$\approx 1000 \times 1000$$

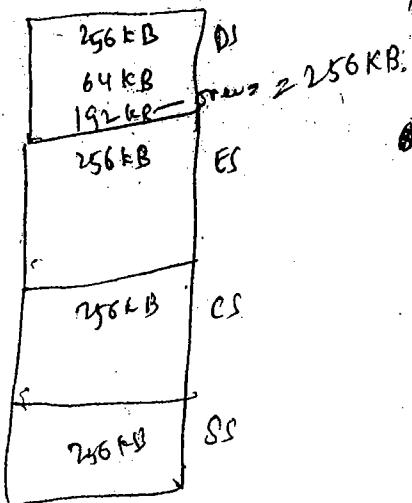
$$\approx 1 \text{ MB}$$

1 MB of memory is divided into 4 segments.

$$\approx 1024 \times 1024$$

4

256 kB available



Or, the most Jr. can access only 64 kB.

192 kB is free space available

In 16 each segmentation.

DS = 480

16 CO

(Q10):
 i) cont.
 ex: DS -
 ES -
 CS -
 SS -

→ hold 16 eff.
can point 16

free space used for:-

negative utilization of memory

multitasking possible

free Address

free Address

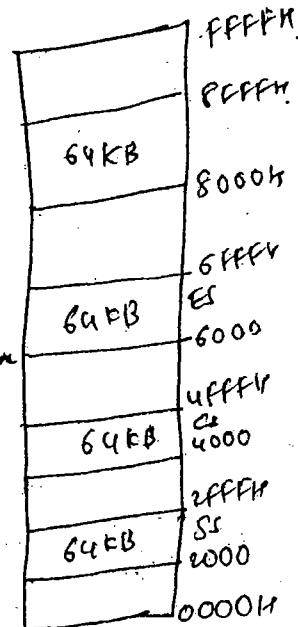
easy access of memory.

for data below 16 256KB use 16 Data segment
16 256KB use Extra 16 segment

$$\begin{matrix} & 64 \\ = & 1024 \times 1024 \\ & 16 \end{matrix}$$

$$\begin{matrix} & 64 \\ = & 64 \times 16 \end{matrix}$$

Instead of 16 this gr is divided
into 16 logical segments. Each segment
is divided into 4 parts.



entation.

The content in the segment is the base Address.

ex: DS = 4000
ES = 5000
SS = 6000
CS = 7000

Base Address.

P.A = T.B.A + E.A (only offset address)

$$DS = 4000 \quad \text{if } B.A = 4000$$

$$TBA = 40000$$

Content in 16 SI, DS + BP, SP, IP is effective Address

$$SI = 1034H \quad B$$

$$ABCDH \quad BP = 5678H \quad SP = 1234H$$

→ In function IP is save as PC ~~if always help~~
16 Address of next memory location.

Note :- DS associated with some indent (SI) (DI), BP (CR) BX

- i) ES → DI \Rightarrow DS \downarrow SF
 $pA = 40000 + 1234H$
hardware zero
- ii) CS → IP
- iii) SS → SP \Rightarrow 41234H
- iv) DS → BX

Eg. $ES = 5000H$ $BP = 5000$
 $TBA = 50000$

Then effective address $DI = ABCDH$

$$\begin{array}{c} pA = 50000 \rightarrow ES \\ \hline ABCD \rightarrow DI \\ \hline 5ABCDH \end{array}$$

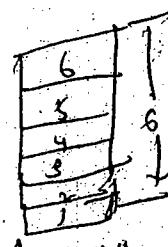
10 P.

Challenger's o

④ Instruction Queue Byte is

The basic principle of this is First IN First Out

The length of instruction Q-byte is 8 byte

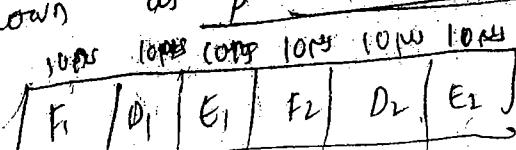


1 byte
total = 6 byte.

In BIU fetches 6 instruction byte at head of time for execution. These unit: pipeline free fetched instruction stored in a group of register is called 'queue'.

It helps to fetch 16 next instruction while the current instruction is executing.

This is known as parallel processing or pipelining



INT0 →
INT1 →
INT2 →
INT3 →

always hold
for

Total time required to fetch & execute = 60ns

, BP (or) BX

F1	F2	F3	F4	F5	
D1	E1	D2	E2		

10ns 10ns 10ns 10ns 10ns

D1 E1 D2 E2

E1 E2

D2 E2

E2

MAY — overflow result.

Instruction Set of 8086: $16 = 64\text{KB}$ opcodes are possible

Divided into 6 categories:

- 1) Data transfer instructions
- 2) Arithmetic instructions
- 3) Logical & bit manipulation
- 4) String instructions
- 5) Branch control (of processor control)
- 6) Machine control (of processor control)

Data transfer instructions

MOV d, s
 R → R-bit (d, 16-bit Register or memory location or)
 M → data
 data → 8-bit or 16-bit immediate data.

e.g.: MOV AL, 40H

MOV AX, 4000H
 16-bit data

MOV BX, [4000]
 Address

MOV [5000], AL

MOV AX, BX

MOV AL, [8F]

MOV [5000], [4000] X

MOV DS, 4000H X

MOV AX, 4000H

MOV DS, AX

Direc. Addressing mode

Indirect Addressing mode.

Memory transfer or copying

We can't make any

segment reg directly.

IP

C

IP

C

possible

→ we can't ~~mix~~ code segment directly.

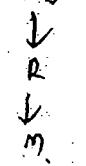
$MOV CS, 4000$ X

$MOV AX, 5000$

$MOV CS, AX$

→ no flags are affected.

(2) $XCHG d, S$ (not implied)



(not immediate data)

$AL = 29H \quad BL = 30H$

$XCHG AL, BL$

$XCHG [5000], [4000]$ X

$XCHG [8E], AL$

(3) Push ↓
8-bit or memory location
→ R - 16-bit or memory location
memory to memory is possible by using push &

↓ mode

↓
push instruction. $SP = SP - 2$

(4) POP ↓
16-bit reg or memory location.

using mode.

$SP = SP + 2$

+ don't know

(5) 8086 UP If code segment register content / FA
and IP = 10AH from what is its effective memory address (P.B.)

here any direct.

$CS = BA = 1FABH \Rightarrow EA = 1FAB0H$

$DP = EA = 10AH$

$PS = IRAB0$

$10AH$

16 bit
1-5

a) 2085H ✓ b) 504CFH
(ans)

c) FBC0H ✓ d) F0B5H × (is 8bit)

(16bit)

Q. On 8086 which macro instructions works on
a) signed & unsigned data

ASCII Data

b) unpacked BCD

182 b) 2E3 a) 1E3. d) 1,243.

Q. what is the following sequence of instruction excited by an INT3
what are content of SP and HC register pair

1000: LXI SP, 1FF2H

1001
jov

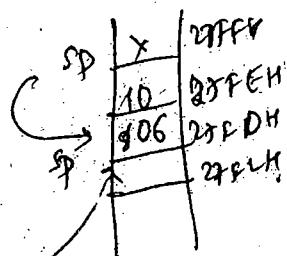
1003: call 1006

1004
1005

pc. 1006
POP H

→ PC = 1006
HL = 1006

SP = 23FFH



Ans - 2003:

Q. 3-memory chips are size 1KB, 2KB & 4KB this
Address bus is 10-bits what are data bus sizes of these chips

a) 8, 16 & 24-bits b) 8, 16 & 32 c) 8, 16, & 64 d) 8, 16 & 128-bits.

1KB = $2^{10} \times 8 \Rightarrow 10$ Address bits (only 10 Address)
given

$$2KB = 2^1 \times 2^{10} \times 8 \Rightarrow 11$$

$$\Rightarrow 2^{10} \times 16 \quad (16 \text{ Ad. Addressing})$$

$$4KB \Rightarrow 2^2 \times 2^{10} \times 8$$

$$\Rightarrow 2^{10} \times 32 \Rightarrow (32 \text{ Ad. Addressing})$$

Q. What is the Address Base of 8086 CPU.

- a) 1MB b) 256KB c) 1KB d) 64KB

Ans: consider the following which
of above flags in 8085

- a) SF b) ZF c) CF d) PF

get affected by the instruction SUB B.

- a) I & 2 b) I & 3 c) 3 & 4 d) 1, 2, 3, & 4

Ans: The content of the accumulator in 8085 CPU are altered

after the execution of the following instruction.

- a) CMP C b) CPI 3A c) AND 40H d) ORA A

Ans: 8085 CPU after the execution XRA R.

- a) CF = 1 b) A = FFFF c) jifed if \neq 0, ZF = 1

Q. What machine cycles are no. of machine cycles "n" and types of memory cycles carried out for push B instruction.

Ans: 1 KB this
is size of stack
16 & 128 bits.
10 address lines
given.

- a) n=2

- b) n=3

- c) n=2

- d) n=3

A)

IMM A-M

B)

implied A-M

C)

Register

D)

Direct

Fetch & memory write

Fetch & memory write & memory read

Fetch & memory write & memory read

Fetch & memory read

LDA 3040H

MOV A,B

LXI H, 2050H

RRC

Arithmetic Instructions:-

LOGIC

⑨ ADD d, S or d=d+d

⑩ ADC d, S \rightarrow d+d+S+R=Post/16-bit

⑪ SUB d, S or d=d-C

⑫ SBB d, S \rightarrow d-d-S

not mmrd data

 \rightarrow all flags are effected.

⑬ MUL S

8-bit/16-bit register or memory.

eg:

implied

stack action

re Accumulator.

⑭ MUL BL

$$AX = AL \times BL$$

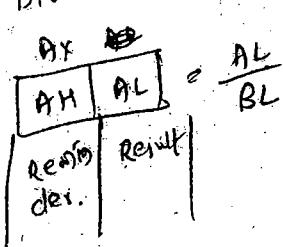
MUL BX

(Ans) $DX, AX = AX * BX$
 Extended Accumulator size is 32-bit

⑮ IMUL S
 unsigned multiplication.

⑯ DIV S

eg: B DIV BL



eg: DIV BX

$$DX, AX = \frac{AX}{BX}$$

↓ ↓
Quotient Result

\rightarrow Extended Accumulator
in 32-bit

⑰ I DIV S

ANS

ORA

XOR

NOT

NEG

ROR d,

RCR d,

Right shift

ROL d

RCL d,

calculator

period

Delay

Logical & Bit manipulation:

AND d, S

ORA d, S

XOR d, S

~~NOT~~ & is complement \Rightarrow q - memory
 NOT d — is complement

NEG d — is comp.

eg: NOT A
 $A = 40 \Rightarrow 01000000$
 $A' \Rightarrow 10011111$
 $\rightarrow A = BFH$

NEG BL
 $BL = 45H \Rightarrow 01000101$
 $\Rightarrow 10111001$
 $\rightarrow BL = BBH$

ROR d, S

RCR d, S
 Right through carry

S
 \downarrow
 first counter

d
 \downarrow
 affect least L/mem

ROL d, S

RCL d, S

Calculate delay time in the following code when it system clock

period $T = 0.3 \text{ ns}$

LXI B, 12FFH

- 10 $\Rightarrow BC = 12FFH$ - 6 $\Rightarrow BC = 12FEH$

- 16

- 4

- 4

- 4

- 4

- 407 $\Rightarrow TLA = 70027.2 \text{ ns}$

Delay: DCX B

XTHL

NOP

NOP

MOV A,C

ORA B

→ TN2 delay

8) $TLA = NMTR \times NIO$ $T = 48$ $NIO = 1 \times 16^3 \times 2 \times 10^3 \times 16^3 \times FXI$ $= 4863H$

$$T_L = (70027.2 - 8703) \mu s = 6132 \mu s$$

$$\Rightarrow (70027.2 - 0.9) \mu s = 70026.3 \mu s$$

$$T_D = T_0 + T_L$$

$$\Rightarrow (10 \times 803) + 70026.3 \mu s$$

$$\Rightarrow 70029.3 \mu s = 70.03 \mu s$$

Q. What is status of carry and zero flag after execution

of following program.

MVI A, 85H $\rightarrow A = 85H$

MVI B, 0EH $\rightarrow B = 0EH$

XRI B, 69H $\rightarrow B = 6AH$

ADD B, 8AH $\rightarrow A = 8AH$

ANL B, 98H $\rightarrow A = 8AH$

CPL B, 98H $\rightarrow A = 8AH$

SHR B, 3010H \rightarrow

HLR $\overline{CF} = 2F = 0$

Q. what is content of AX

push a

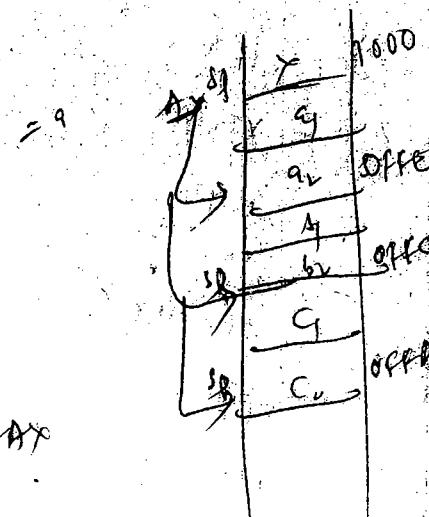
a) $a+b-c$ b) $C+b-a$

push b

c) $C+b+a$ d) $C-b-a$

push c

pop AX $\rightarrow C$



pop BY $\rightarrow B$

SUB AX, BX $\rightarrow C-b$

pop BX $\rightarrow a$

ADD AX, BX $\rightarrow C-b+a. zAX$

STANG

D

SI

E02

REP/C

REP

REP2

PEP

Consider the following instructions executed in a 8086

(a) value stored in G1 would be.

push AX \rightarrow 20H is in it.

push BX \rightarrow 34H is in it

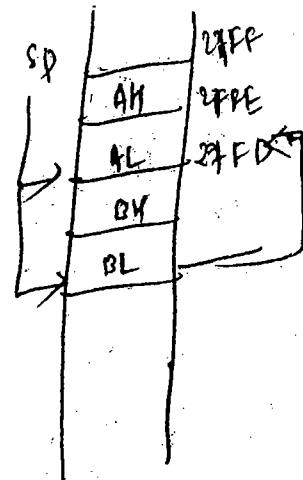
pop AX \rightarrow 34H

ADD AX,BX \rightarrow $AX = 34 + 34H = 68H$

pop G1 \rightarrow 20H

20H b) 34H

34H d) 68H



after execution

Consider (b) following

1. MOV M,A

✓ 2. ADD C

✓ 3. AND A, FFH

✓ 4. CMP M, C

which

of (yes cause change

in status of flag.

+b-a

C-b-a

String Instruction

DP=0

ZF=1

SI ↑ increment
EDI ↓ decrementing STD
by using CLD

81] Decreases by using STD

until Counter equal to zero)

REP / C Repeated
REPE

until ZF=1

REPZ E

until ZF=0

1	X	1000
2	g	
3	a1	0FFE
4	A1	0FFC
5	b1	0FFC
6	G1	0FFD
7	C1	0FFD

(ii) REPZNE

until ZF=0

4) MOV S / MOVS / MOUSB / MOFSW

→ Both S & D are implied. Transfer Data from

DS to ES.

d ↓
↓ S
ES DS

MOV SI, 4000

MOV DI, 5000

MOV CX, 0004H

x: MOV BL, [SI]
MOV [DI], BL
INC SI
INC DI
DEC CX
JNZ x
HLT

DS=0010H ES=0000H

41	4000
4E	4001
43	4002
44	4003

41	5000
501	5001
502	5002
503	5003
504	5004

STOS / STC

↓
STC
HLT

MOV SS, 4000

MOV DI, 5000

MOV CX, 0004H

CLD

REP MOUSB

HLT

Transfer

counter
int[rel] mov

counter x: mov

data 01H

↓

DT

J1

*

SCAS / SC

↓
Scan string sc

b

MOV

MOV

MOV

y: comp

JE -

INC S

DEC L

JNZ

5) LODS / LODSB / LODSW

Load string byte word

DS to accumulator (AL)

Transfer Data from

MOV SI, 6000

MOV CX, 0004H

x: MOV AL, [SI]

INC SI

11	6000
12	6001
13	6002
14	6003

From

DEC CX | Legend to
JNZ X: LODSB. MOV SI, 6000
HLT MOV CX, 0004H

CLD — Increases SS

REP LODSB
HLT

STOS / STOSB | STOSW |
↓ Store string
store byte store word

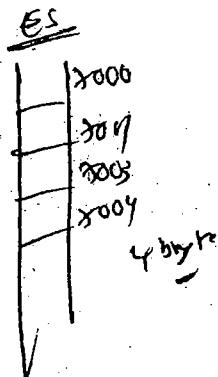
Transfer data from up to ES

MOV DI, 7000

Counter initially MOV CX, E0004

counter: MOV (DX), AL
data 01H INC DI
DEC CX JNZ X

MOV DI, 7000
MOV CX, 0004
CLD
REP STOSB
HLT



SCAS / SCASB | SCASW |
↓ Scan string scan string
byte

MOV SI, 1000
MOV CX, 0008H
MOV AL, "1"
y: CMP AL, [SI]

JNE -X

INC SI

DEC CX

S	
10	100%
20	10%
30	100%
4	100%
12	100%
14	100%
16	100%
18	100%

MOV SI, 1000

MOV CX, 0008H

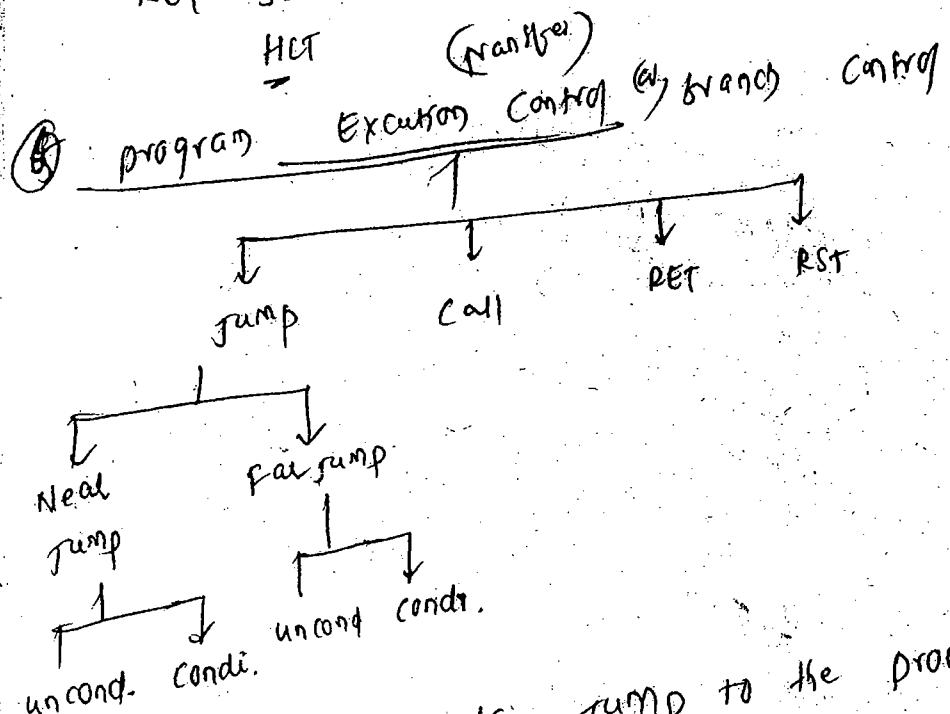
MOV AL, 11H

CLD

REP SCASB

HLT

(branch)



→ Near jump is jump to the procedures (sub routines), which is in the same segment.

→ Far jump is jump to the procedure which is in different code segment.

Q2 4000 MOV BX, 2000

4003 MOV DI, 9000

4006 MOV CX, 8000H

4009 MOV AL, (82)

400C MOV (0E), AL

400F INC BX

4010 INC DI

4011 DCE ex₀₀₀₀JN2 X

42
85
27
31
49
51
64

→ If length of program is not greater than 16 bytes
 code segment so, it is a near jump.

JMP X
 JZ, JNO, JC, JNC, JL, JLE, JG, JGE,
 JZ, JNZ, JPE, JPO, JF, overlow

JB, JBE, JA, JAE, JS, JNS → SF=0
 write a Assembly language program to jump ie mnemonics
 from a given byte number.

42	2000
85	2001
23	2002
21	2003
49	2004
51	2005
69	2006

eduled (Sub)

t.

which is

1-byte data

00 - mnemo.

FF - max no.

initialise DS

n, dB 42H, 35H, 29H, 31H, 49H, 52H, 64H

(Variable byte type)

Data ends

code segment

Assume CS:code, DS:Data

start: MOV AX, data

MOV DS, AX

MOV BX, offset n

MOV CX, 0008H

MOV AL, OFFH

Y: CMP AL, [SI]

AL <= SB

JB X

MOV AL, 42

FFC42 X

DCX CX

JNL Y

MOV SI, AL

HLT INT3H

code ends

end start

INT

LDA

JMP

POP

SPL

16 6

01P at 1

CALL

→ There are two types of call instruction.

i) Near call ii) far call

→ Near call is call to procedure which is in the same code segment after w15 far call.

5000 MOV AX, 1234H

MOV BX, 5678H

ADD AX, BX

call 8000H

8000
5000
3000H → < 64 KBSuggesting let
same code
segment.

HLT

MUL

MUL

ADD

OUT

HL

Q. Give meaning of operation for the following

i) operand ii) 00-0F bytes iii) machine cycles

No. of 00-0F states

Instructions are

LDA

JMP

POP

PC CC

shown

memory

Addr

30

300C

3002

3003

<u>Instruction</u>	<u>opercnd</u>	<u>No. of bytes</u>	<u>No. of M cycle</u>	<u>No. of clc</u>
LDA	4000/first address	3	4	13
JMP	16-bit address	3	3	10

Q. Specify the register content and the flag states after executing the following instructions also indicate the O/P at port 01.
 Ans: It's same

A	B	SF	ZF	CF
00	FF	0	1	0

0. \Rightarrow 164KB
 His 16 bits
 no. code
 ment.
 169
 cycles

MVI A, FAH \Rightarrow A = FAH

MVR B, ZAH \Rightarrow B = ZAH

ADD B \Rightarrow A = 6CH

out port 0 \Rightarrow 6CH

HLT \Rightarrow A = 6CH

B = ZAH

1110010
 01111010
 11111111
 01001100
 6 C

SF = 0

CF = 1

ZF = 0

Q. The content of some memory location after an 8085 MP. instruction will be content of KC register pair

memory Address (Hex)	content (Hex)
3000	02 = C
3001	30 = X
3002	00
3003	30

after execution of following program.

LHD 3000 \Rightarrow HL = 3002H

MOV B, M \Rightarrow E = 00H

INX H \Rightarrow HL = 3003H

MOV D, M \Rightarrow D = 30H

LBAX D \Rightarrow DE = 3000H

MOV C, B \Rightarrow L202H

INX D \Rightarrow 3001

Q. A set of 16-bit reading is stored in memory location starting at 2050H, each reading occupy memory location 2050H i.e. lower byte stored first followed by its higher byte and add 16-bit content of 2052 & 2053 after execution of the following program where is the result.

~~801~~: EHLQ 2050 \Rightarrow HL = 1514H

~~DE~~ \Rightarrow HL: XCHG \rightarrow DE = 1514H

LADD 2052 \Rightarrow HL = 1A16H

DAD D \Rightarrow DE + HL acc.

MOV AL

STA [0509H] } offset

MOV BH

STA 2055H

~~801~~

2050	19
2051	15
2052	16
2053	10
2054	1A
2055	16

HL = 1514H
DE = 1514

2C2A H

HL
DE

2C2A H

Q. Content of registers pair and accumulator of 8085 H.P. are 50H and 3BH respectively at content in the accumulator and status of the carry flag & sign flag after execution of the following instruction.

SUB B.

A = 0011 1011

B = 0101 0000

~~801~~: ~~Sub B = 1011 0000~~

~~11~~
11101011

0110000

X = EBH

CX = 0 | CF = 1 AF04

SF = 1 |

all P?
cc
2
D
L
W
eo clear
801 : A

AN
of mem

2000	1
2001	
2002	2

in memory location
memory locations
by 16 bit words
1024 of them.

for 16 bit instruction given below how many memory operations
performed during 16 bit execution in an 8085 MP

call 2000H - F, R₁, R₂, W₁, W₂ - 5

LDA 2000H → F, R₁, R₂, R₃ - 4

Write am instruction which take min possible time
to clear 16 bit Accumulator.

* MVI A, 00H - 7 clock cycle

XRA A - 4T ✓

ANI 00H - 7T

An 8085 Assembly language program is given below. 16 bit
memory locations at the end of 16 following program
what is 16 bit content of A, B, C, H and L?

2000	18H
2001	10H
2002	28H

i) Status of CF, ZF & SF

ii) Content of 2000, 2001, & 2002!

0001 0000
1111 000
1000 100

iii) MVI C, 03H ⇒ C = 03H

LXI H, 2000H ⇒ HL = 2000H

MOV A, M ⇒ A = 18H

DCR C ⇒ C = 02

0001 1001
1111 000
1000 100

0001 1001
1111 000
1000 100

HL = ~~1000 1001~~ 2000H, ii) 2002

0001 1001
1111 000
1000 100

0001 1001
1111 000
1000 100

(loop1): INX H

MOV B, M

CMP B

JNC loop2:

MOV A, B

loop2: DCR C

TZ2 loop1

⇒ ...

B = 10H, 28H

A = 08H, ED, and CF = 1

2BH ←

C = 01H

E = 200H

STA 2100H ~~28H~~ → 28H

HLT

~~A=0~~

y A=28H

B=28H

C=2004

H=20

L=02H

y CF=1

SF=0

PF=1

2000	18
2001	10
2003	2B
2004	2B

write

1. Sync
output com

2.

inner count

control instructions

y Proces Control / machine control

y CLD (Clear It direction flag)

DF=0 SI ↑ increment
DI ↓

(Set It direction flag)

2) STD (Set It direction flag)

DF=1 SI ↓ decrement
DI ↑3) CLD (Clear It interrupt flag)
IF=0 → INTR is disabled

IF=1 → INTR is enable

STC CR1

addressing

one

CF=0 → CF=1

CF=1 → CF=0

4) CLC → clear (E) carry flag
CF=0

different

memof

memof

memof

write a Assembly language program to arrange given

7-byte numbers into a) Ascending b) Descending order.

Outer Counter
MOV B, 0005H (C-15)

Inner Counter
Z: MOV SI, 4000H
MOV CX, 0005H

MOV A, [SI]

INC SI → 4001H

Y: CMP AL, [SI] → AL < (SI)

JBE → X.

X: XCHG AL, [SI] → AL = 95 (SI = 15)

DEC SI

MOV (SI), AL

80C

X: INC SI

DEC CX → 0104H

JNZ → Y

DEC BX → 0004H

80C become zero.

B is decremented by 1

(here C loaded with same value)

~~JNZ~~

JNZ → Z

HLT

Addressing modes of 8086 MP

Different ways in processor can access data either

memory or I/O devices.

memory Direct A^m

memory Indirect A^m

Memory I/O A.M.

Memory Direct A.M.

(4)

Ex: In this both the source & D. are 16-bit, after
register A.M.: Registers.

internal registers.

es: MOV AX, BX ADD AL, BL

CMP CX, DX

Immediate A.M.:

Source must be either 8-bit/16-bit immediate
data and di must be 8bit/16bit internal Registers

B: MOV AL, 40H ADD BL, 20H
MOV CX, 4000H CMP CX, 5000H

Implied @ Implicitly A.M.:

DAA RAS (Ass Adjustment after sub)
DAS AAM
AAA APD -

Direct A.M.: In this Am either source or destination must be

16-bit Address.

MOV AL, \$4000

$$\begin{aligned} 4000 - EA \\ DS = 1000H \rightarrow BA \Rightarrow BA = 10000 \\ PB = 14000H \end{aligned}$$

Memory Indirect A.M.:

In this addressing mode either

Index A.M.: In this source/destination must be specified by either

SJ (SI) DI.

Ex:

Based

destination

base phs

es:

mc

m

Based

by us

mc

m

es:

mc

m

Based

A

either

32

Ex: `MOV AL, [SI]`

`MOV (DI), BL`

16-bit offset

Based Addressing mode: In this A.M. either source or destination must be specified either by using BX register or

base pointer (BP).

$BX \neq BP$

16-bit immediate

Registers

Ex:

`MOV AL, [BX]`

`MOV BP, BL`

In this A.M.

either S/D specified

Based Indexed A.M.:

(i) using base BP and SI/ DI.

`MOV DL, [SI+BP]`

Ex:

`MOV [DI+BX], BL`

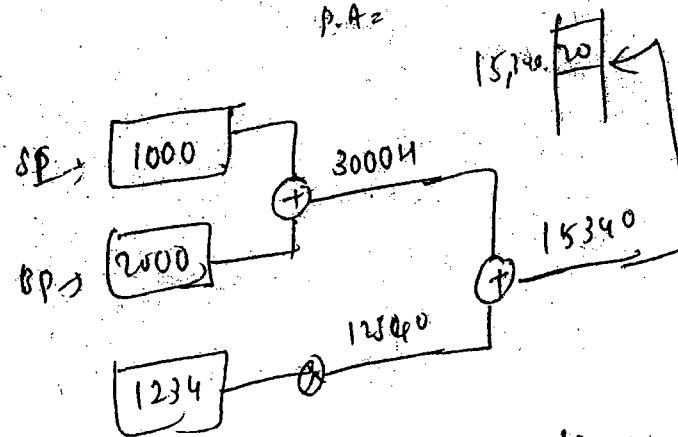
$$\begin{array}{l} SI \Rightarrow 1000 \rightarrow EA \\ BP - 2000 \rightarrow EP \end{array}$$

$$1000 + 2000 = 3000 = E.A$$

$$DS = B.A. = 1234H \Rightarrow 12340 \Rightarrow R.B.F$$

action must be

P.A. =



either

either

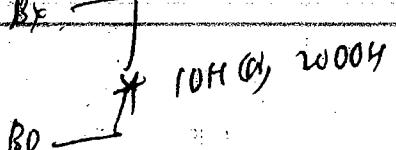
Based Addressing mode + Displacement data: (a), relative based A.M.

either S/D must be BX & BP

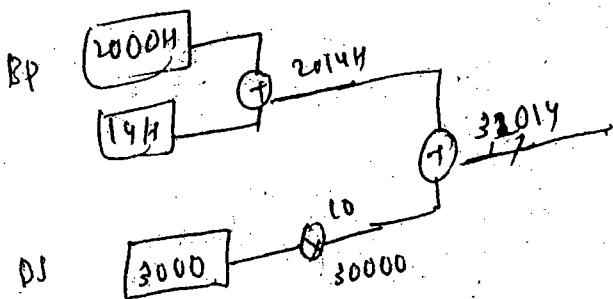
→

either S/D must be BX & BP

BF



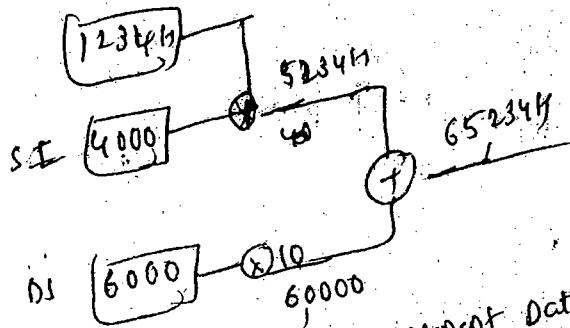
Eg: $MOV AL, [BP + 10H]$



3) Indexed Address + Displacement Data:

SI

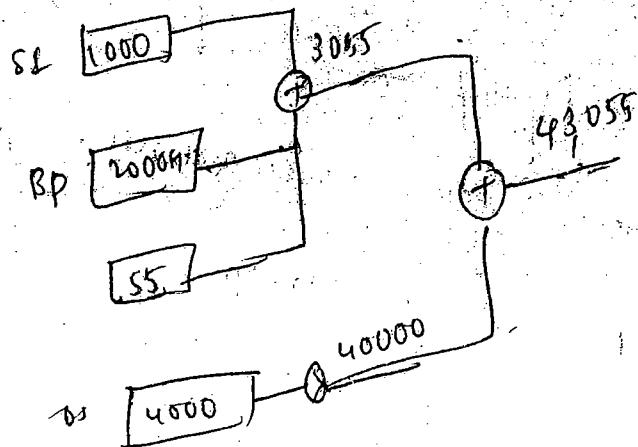
Eg: DR
 $MOV AL, [SI + 1234H]$



Boxed Indexed Address + Displacement Data:

Boxed
(relative based A+n + D)

$MOV AL, [SI + BP + 55H]$

Memory

IN

ALU
A

IN AL

IN AX

OUT d

RS/16 bit

Variable A

MOV BX,

OUT DX

A MP

16 bit

Q: (a) Con.

(b) If SP =

effective RS

Memory I/O A-M

bits S&D are specified.

i) Fixed

ii) Variable I/O

IN d, S
 ↓ ↓
 ALU PSH
 AX @, 1615F] Port Address

IN AL, 40H] fixed

IN AX, 11234H] I/O A-M.

OUT d, S

↓
PSH 1615F AX W AL.Variable A-M:

MOV DX, 4000H

OUT DX, AX

Q: A MP ran 24 address lines and 32-data lines what is

it's memory.

memory = $2^4 \times 32$ bits

$$\approx 2^4 \times 2^{20} \times 4 \times 8$$

$$\approx 2^{26} \times 2^{20} \times 8$$

$$\approx 64 MB$$

Q: If content in the code segment is CS = 12BFH from the content

if DS = 1A02H, SP = 2000H BP = 4000H calculate the

effective physical address.

$$PA = 12BF \Rightarrow TA = 12BF$$

$$EA = 1A02$$

$$AO = 141E2$$

calculate P.A of the following specifications.

$BP = DS = 4000H \quad SP = 2000H \quad SP = 6000H \quad DI = 3000H \quad SS = 9000H$

$BP = 5000H$

$PA = DS = 4000 \Rightarrow PB = 40000$

EASL = 9000

PA = 49000

for the following instruction calculate 16 bit E.A.

MOV [BX][SI], AL

$BX = 6000 \quad] \quad 8000 = T.E.A \quad (\text{length} 16bit)$

$SI = 2000$

$DS = 3000 \Rightarrow TB = 30000$

$+ = 38000$

MOV AL, [BP+42H]

write a Assembly language program to reverse the string.

Data segment

(HYDERABAD)

11 dB

12 dB

Data ends

Code segment

Assume CS:CODE, DS:DATA

Start: MOV AX, DATA

Mov DS, AX

Mov SE, 0000H

Mov DS, 0000H

Mov DI, 0000H

Mov CX, 0009H

SI
H
1000
9001
D
8002
A
7003
B
6004
G
5005
R
4006
E
3007
D
2008
M
1009
Y
000A

start?

Y: C

X

10H SF = 90004

ADD DI, 09H

X: MOV AL[SI]
MOV [DI], AL

INC SI

DEC DI

DEC CX

JNZ X

HLT

Q. Write a Assembly language program to print 10 strings (length 10 each)

string

Data segment

n, dB ('GATE \$')

Data ends

Code segment

Assume CS:code, DS:Data

G	4000
A	4001
C	4002
E	4003
S	4004

4000

4001

4002

4003

4004

4005

4006

4007

4008

start: MOV AX, Data

MOV DS, AX

MOV SI, DS[00H]

MOV CX, 0000H

MOV AL, \$

Y: CMP AL[SI]

JE X

INC CX

INC SI

JMP Y

X INT 34 (temporally stop (G program))

W.A.P. for to follow my specifications.

MOV SI, 9000H

(convert unpacked BCD

MOV DI, 4000H

to packed BCD)

MOV CX, 004H

X: MOV AL, [SI]

INC SI

MOV BL, [SI]

MOV CL, 04H
Persons

ROR BL, CL BC=04H
after 4-rotations
→ BL = 40H.

OR AL, BL → AL = 42H

MOV [DI], AL

INC SI

INC DI

DEC CXH

JNZ X

INT 3H,

8255 (PPPI): (40-pin 2e) (General purpose)
210 series
parallel interface.

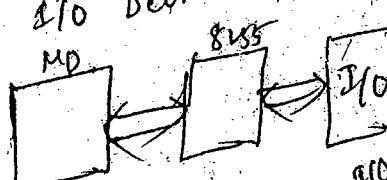
→ It is a programmable peripheral interface.

→ It is used for parallel data communications.
SIO devices are connected to it via through

Note: Any

XP+LP

8255.



Only
Parallel connection!

If it is general purpose programmable I/O
device.

If it can transfer data from simple I/O to intelligent
I/O.

	4P	1P
02	5000	4000
03	5001	4001
04	5002	4002
05	5003	4003
06	5004	4004
07	5005	4005
08	5006	4006
09	5007	4007

8 bits
Address

→ 16-bit
register

	6000
24	6001
35	6002
DF	6003
93	6004

port A/86

Port A:
One

Port B: One

Port C: One
(A)

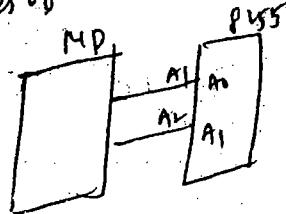
	01P
5000	42 4000
5001	65 4001
5002	88 4002
5003	39 4003
5004	4007
5005	4005
5006	4006
5007	4002

③ It has 8-data lines D0-D7, then are used to interface i.e. S/M buses with internal buses of the 8255.

④

⑤

It has 8 address lines A0, A1. These are connected to 16 Address lines of i.e. A1, A2 of 8086/8088 MP.



→ These are used to decode port address and control word

register

A1	A0	port
0	0	port A
0	1	port B
1	0	port C
1	1	CWR

i.e. Port A = 0130H

(thus) Port B = 01302H

Port C = 01304H

CWR = 01306H

PWR

PA to

11

0130H

01

0132H

10

0134H

11

0136H

→ A15 A14 to all Address lines A1-A4 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 → 8086/8088

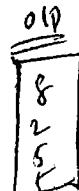
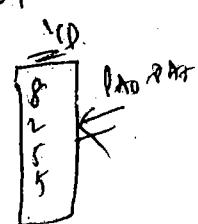
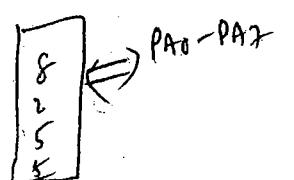
→ 3-8-bit ports Port A, Port B, Port C.

008.

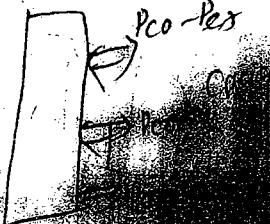
→ to MP through

PORT A:

one 8-bit - Bidirectional I/O port

PORT B: one 8-bit

Bidirectional I/O port (PB0 PB7)



PB to PB full

PORT C:

one 8-bit - Bidirectional I/O

(a) two 4-bit - Bidirectional I/O.

8) each port has unique address to read data from / write data into I/O devices.

port A - 8000H $\xrightarrow{\text{PORT}}$

port B - 8002H

port C - 8004H

CWR - 8006H

PORT

port A - 00H

port B - 02H

port C - 04H

CWR - 06H

→ 85 D1 = 1

→ 85 D1 = 1

9) It has 24 of bidirectional I/O lines i.e. driving capability of each I/O lines is 2.5 mA.

9) Basically 8255 operate in either mode (i) BSR mode.

8255

BSR mode (Bit set/reset)

I/O mode

II

used to programming

Set / Reset individual bits in

port C

mode 0 mode 1 mode 2 mode 3

operations are selected by using it.

9) all three mode off operations are selected by using it. internal register is known as control word register (CWR).

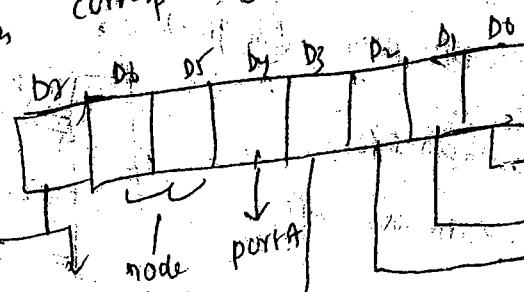
→ S/I
→ M/A
→ ACT

BSR Register

CWR:

It is a 8-bit programmable register. It decides I/O modes of operation, corresponding ports which acts as I/O (A, C/P port).

- Decides



0

CWR

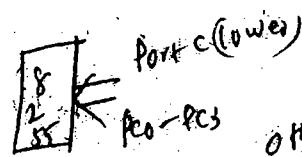
BSR formatate

C

BSR

data from /

85 D0=1 port C lower act as I/O, port



otherwise acts as I/O port.

→ 85 D1=1 → Port C acts as I/O otherwise I/O:

D2	mode selection for port B	
0	mode 0	
1	mode 1	

mod

D6 D5	mode selection for port C	
0 0	mode 0	
0 1	mode 1	
1 X { 1 0 } 1 1 }	mode 2	

Ex. → S/I mode \Rightarrow D7=1 \rightarrow mode 0 \rightarrow acts I/O port (all ports)

I/O mode 0's

BSR register

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0

I/O port.

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

8 0H

Operated from BSR mode.

D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X				

 \rightarrow CWR = 00H when F255

BSR format

D7	D6	D5	D4	D3	D2	D1	D0
0	X	X	X				

BSR mode.

Set or Reset

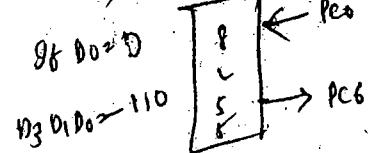
Bit positions in port C

when for port B

etc

D ₃	D ₂	D ₁	BTF POSITION (ms)	Port C
0	0	0		Pc0
0	0	1		Pc1
0	1	0		Pc2
0	1	1		Pc3
1	0	0		Pc4
1	0	1		Pc5
1	1	0		Pc6
1	1	1		Pc7

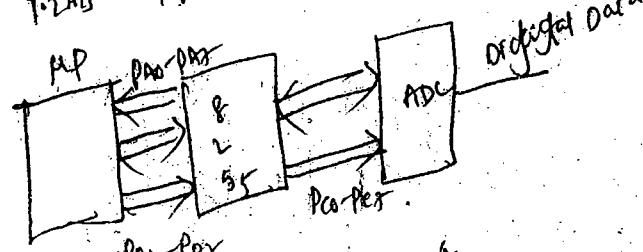
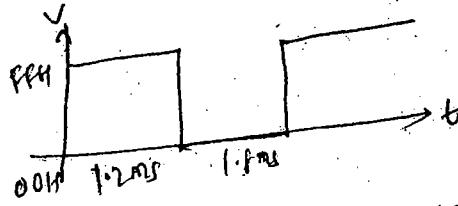
00 001 \Rightarrow Pco acts as S/P port
 $D_3 D_2 D_1 \rightarrow 000$



design

Q:

W.A.P for the following pr. waveform.



D ₇	D ₆	D ₅ , D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	0

A \rightarrow 82H

MVC A, 82H

out port CWR

MVR A, 0FFH

out port B

call delay1

MVC A, 00H

out port B

call Del942

15W

off exm

moi

sample

3-8-bit

avai

port B, PC

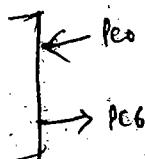
2-4-5

available

port C (U)

$\rightarrow 000$

sets all S/P port

delay: $MVI C, 82H$

DCR C

NOP

JNZ X

$$TCA = n \times 9 \times N10$$

$$1.2ms = 18 \times 0.8ms \times N10$$

$$\text{Counter value} = \frac{1.2 \times 10^3}{18 \times 0.5 \times 10}$$

$$= \frac{12 \times 10^3}{9 \times 3}$$

$$= (180)_{10}$$

 $= 82H$ delay

$$\text{Counter value} = \frac{1.5 \times 10^3}{18 \times 0.5 \times 10^6}$$

$$= (166)_{10} \quad \frac{166 \times 10^6}{10^{-6}} \\ = A6H$$

delay: $MVI C, A6H$

X: DCR C

JNZ X

RET

Z: $MVI A, FCH$ for generating waveomp Z:Offering 6/10mode 0

simple I/O mode

mode 1y strobed I/O (only)
Hand shakingmode 2y bidirectional I/O
(a) Hand shaking

3-8-bit ports are available port A,

NPB, port C

y two-8-bit port are available port A & port B
port C signals are used to control I/O opreations of A & B.

y one - 8-bit port is available ie port A & port C signals are used to control I/O operations of port A.

- 4-bits are

available port C (lower)

MPC (upper)

No 4-bit ports

No 4-bit ports

No. of I/O pins

— No —

2 - 1/0 (μs)

PC₃ & PC₂

5 - 1/0 (μs)

PC₃ to PC₂

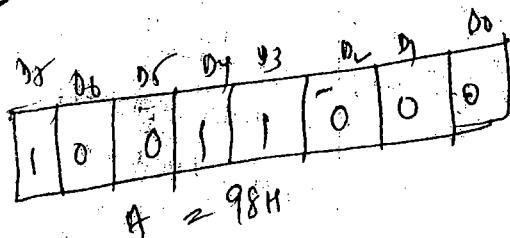
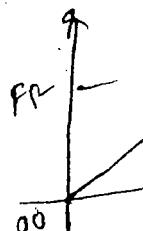
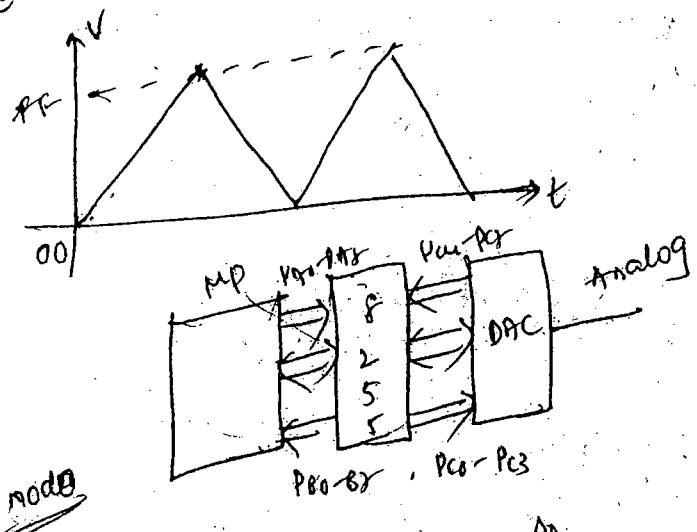
5) UPS are not latched
But O/Ps are latched
(holding data)

6) Both UP & O/Ps are
latched

7) Both UP & O/Ps
are latched

Grayscale
waveform

Q. W.A-P for the following waveforms



MOV A, 98H

OUT COM

MVS A, 00H

~~MOV A, 00H~~

DX: OUT PORTA

INRLA

AND PFH

JC X

OUT PORTA

DCR A

MOV

DW

Y: MVI

X: OUT

GRAP

CMO

JC

JMR

Emissions

DMA control

purpose

it is

+5V DC

it is a

memory

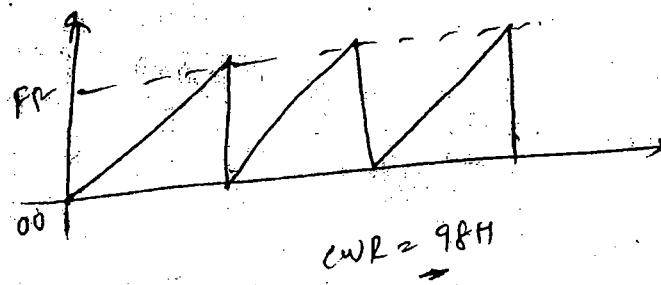
and

5 - I/O pins.

Poj to Poj.

It is type of operation
we discuss

CMP 00H

JNE Yagainst
wiring TMPQ: w. A interfacing program for it following to generate
the following waveform.MOV A, 98H }
out CWR

Y: MVI A, 00H

X: out port A

~~ENHANCED~~ (NCA)

CMP PFH

JC X] [local increment and
specifications] sudden fall to 00'
JMP Y]DMA controller 8253

purpose Direct memory Access

→ It is a standard 40 pin IC available dual inline PC

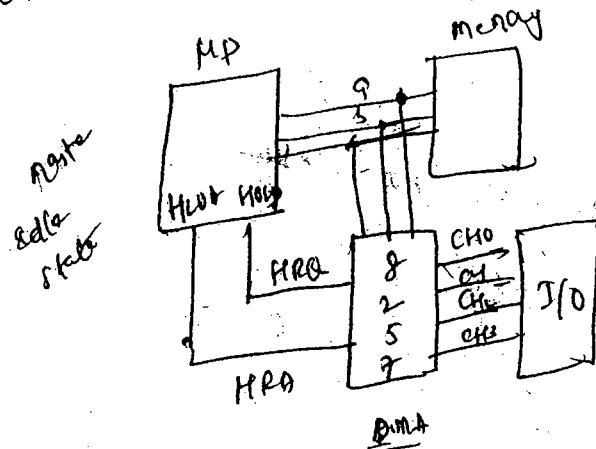
→ +5V DC single supply

→ It is a programmable I/O Device

→ Memory to memory data transfer is possible

→ Data transfer rate of DMA Controller is 100

8f has 4 identical channels CH0, CH1, CH2, CH3



CH0 → higher priority

CH1

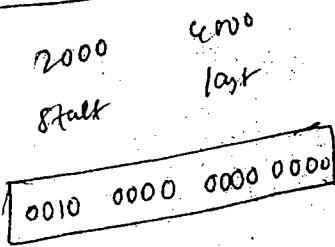
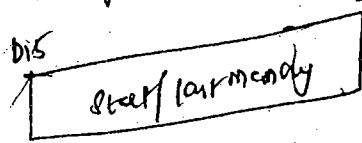
CH2

CH3

6) Each channel has 2 16-bit registers, a DMA Address register & a terminal count register.

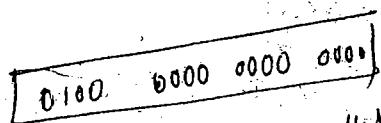
a) DMA Address Register

→ It is 16-bit register (D0-D15) & holds the either the starting memory location or last memory location.



After transmission of each data byte automatically incremented &

2001
2002
2003



FFF → automatically decremented

b) Terminal count register

out of 16 bits 14 bits used for counting and remain 2 bits D14 & D15 used for DMA type of operation

8f has

- mode
- status

mode set

8f is 5mm
8f is 8
it is derived
8f is

1₂ CH₃

r priority

RA Address

		<u>DMA type or op</u>
015	014	
0 0		Verify DMA cycle (whether system burst AND write operation) - I/O to M.
0 1		Read operation - M to I/O w.r.t
1 0		X
1 1		

so only

14-bits for counting $2^{14} = 2^{4 \cdot 2^10}$

• 16 EB.

it can count from 0000 to FFFFH.

it has two 8-bit register common to all 4 channels.

after the

a) mode set register

b) status register

mode set register

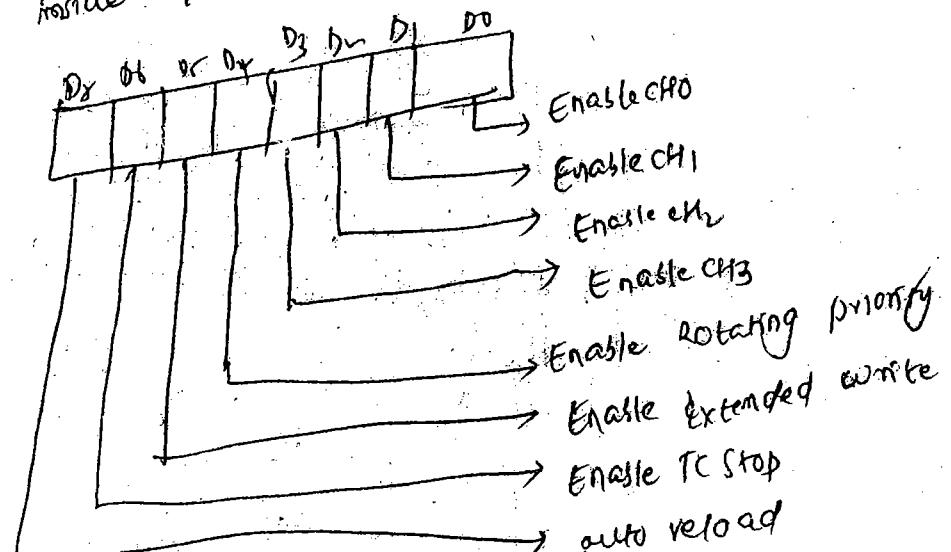
→ it is 8-bits CCR in 8255

→ it is sufficient to control mode of operations

it is 8-bit programming register decide if mode of operations

it decides whether corresponding channels are Enable or Disable

it is inside the DMA controller.



if D0 = 0 → channel is disable

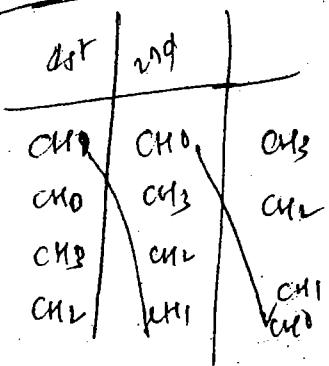
= 1 → channel is enable

~~if D4=1 rotating priority is enable otherwise fixed priority~~

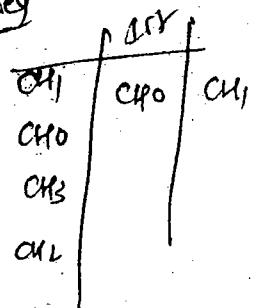
writer

or Normal priority is Enable

Rotating



fixed



channel

memory locan

step 1:

step 2: TC RE

→ 10 D5=1

Extended write

IOW (Extended modulation of IOW)

TC stop is enable of corresponding channel is disable

→ 10 D6=1

TC stop is disable

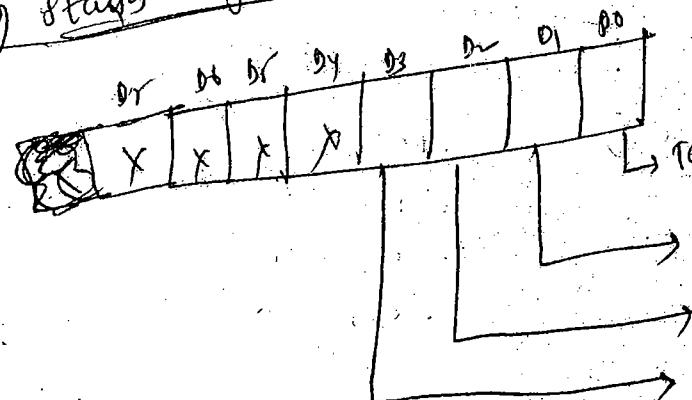
even though TC is received in corresponding

channel is enable

auto reload is enable

→ 01 D7=1

→ Status Register



TC Status for CH0

TC → CH1

CH2

CH3

step 3 mod

by r

E.

step 4 sti

DY DS
0 n

MU

out

MD

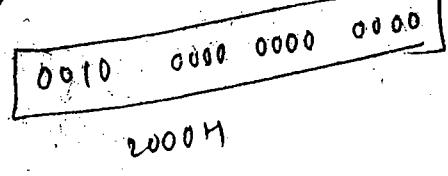
DI

MI

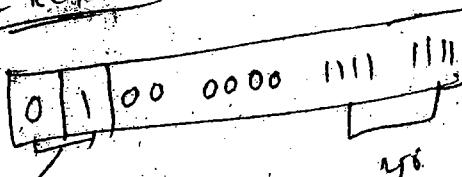
Fixed priority

Q. Write A. Program Transfer 256 bytes of data through channel zero from I/O device to memory and setting memory location of text is 2000H.

01: step 1: DMA Address register content



Step 2: TC Register



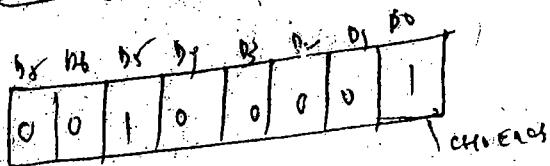
9 channel is
selected

write
operation

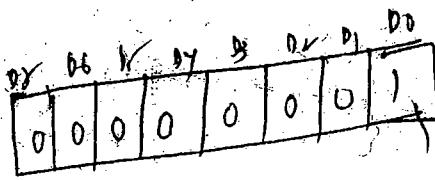
$$256 \Rightarrow N = N + 1$$

$$= 256$$

mod 8 set register



Step 3: Status Register content



TC
CH0 → busy enable

MVI A, 21H
out MSR } { MSR.

MAS A, 0H

out S-R] S.R
MUL A, 00

out port 1
MOV A, 20H
out 0MA

DMA

Q what is O/P at port 1. when it following instruction sequence

expected

MOV A, 8FH $\Rightarrow A = 8FH$

ADD 32H $\Rightarrow A = 00H$

JC - Display

out port 1

HLT

Display XRA A $\Rightarrow A = 00$
out port 1 $\Rightarrow 00H$

HLT

Q. find if range of the bytes that will displayed at
a) 00 to 49H b) 51 to 89H
c) 80 to 99H d) None

MOV A, bytes

MOV B, A

SUI 50H

JC delete

MOV A, B

8VL 80H

JC - Display

delete: XRA A
out port 2

HLT

Display MOV A, B
out port 2

HLT

$$\begin{array}{r}
 10001111 \\
 01110090 \\
 +111111 \\
 \hline
 00000000
 \end{array}$$

A=00H
ext-1

MOV A
out 1

A

OUT P1

HLT

display

display EP

LN1 H

MOV B

MOV A

before: MOV N

INR N

INT N

DCR 1

JNZ

HLT

Calculate

clock period

if there is carry so o/p port 2.
if there is no value at port 2.

if range must be
 $80 \leq A \leq 50$

$\Rightarrow 50 \leq A \leq 89$ (50-89)
there is carry

o/p port 2.

Q. Find the type of no. that can be displayed at 100P.

MUL A, Byte[2¹⁰⁰⁻¹⁰¹] 100 to 3FH

ORA A

JP out-port (Jump 1¹⁰⁰)
LSB=0

XRA A

100 111
111 0090
+ 111 0000

= 000 0090

Output: outport → 00, 90.

HLT

Q. Specify so for content of memory location XX30H to XX74H after execution of following program.

LXI H, XX30H ⇒ HL = XX30H

MUL B05H ⇒ B = B05H

MUL A10H ⇒ A = A10H

M = 01

stores: MOV M,A ⇒

INRA

INT A1

DCR B

JNZ stores

HLT

A = 20H

HL = XX31

B = 04H

01H	XX30
02H	XX31
03	XX32
04	XX33
05	XX35

so 000 part.

part 2.

will be

(50-79)
as carry

part 2.

Q: Calculate its delay in the following 100P

Assuming that the system

Clock period is 2.33 ns

Table	Mnemonics	T-states
	LXI B, 0008H	10

Delay: DCR B

RET HL

XTL

NOP

NOP

MOV A, C

ORA B

6	9
16	
16	
4	
4	
4	
4	
4	

68 tstates

$$TCA = \text{NET CYCLES}$$

$$n = (64 \times 0.33 \times 10^6) \mu s \quad M10, 6 \times 10 + 8 \times 10^0 \\ \rightarrow 10^4$$

$$TCA = 2196.48 \mu s$$

$$T_L = (2196.48 - 0.33 \times 8)$$

$$\approx 2195.49$$

$$T_D = T_0 + T_L = [10 \times 0.33 + 2195.49]$$

$$\approx 2198.79 = 2.2 \text{ ms}$$

(K no. 0.08 time 16 following loops are executed)

loop: MUL A, 17H

OR A

JNC loop

$$\begin{array}{r} A = 17H \rightarrow 0001\ 0111 \\ 0001\ 0111 \\ \hline 0 \end{array}$$

$$\overline{0001\ 0111}$$

(CF = 0)

PRO There is no carry

(0 program executed) ~~for write A~~

Q. No. 0.08 times 16 loops are executed.

CF = 0

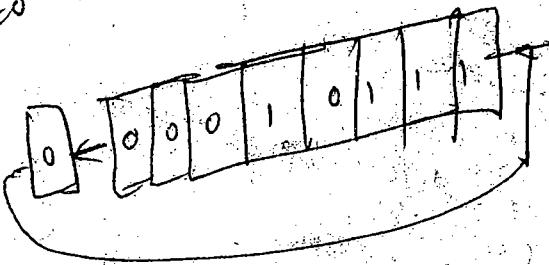
MUL A, 17H

OR A

loop

JNC loop

2.2 = 0



program expected ~~for write A~~

0 < 0010 1110 0E
0 < 0101 1100
0 < 1011 1000

1. The following program executed how many no. 0.08 times?

loop: DCR B \rightarrow 000000

NOP

JNL loop

~~DCR, INR~~

does not affect

surf carry

XOR only once

MUL A, 00 } NT do V

OR A } do V

Q. ORA A
MVI B, C
DCR B
JNC loop
INX }
DCX }

Q. MVI A
loop. ORA A
RNE

JNC - loop

only "

Calculate

MUL
NOP

loop: NOP

DCR B

JNZ con

Q:

NewtonDesk.com

Loop : MVI B, 64H

NOP

DCR B

JNL loop.

ORA A $\rightarrow A = 20H$ CF = 0

MVI B, 64H $\rightarrow B = 64$ CF = 0

DCR B $\rightarrow B = 63$ CF = 0

JNC loop ~~red~~ "multiple times"

INX } no flags
DCX } effected

JNL } CF not affected
DCX } after will effect

MVI A, 13H

ORA A

RRE

JNC -loop

only JNC

Calculate (B) counter

MVI B, count

NOP

DCR B

JNL loop

No. of times?

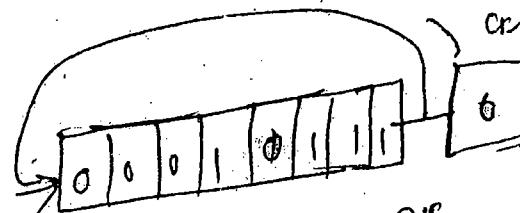
~~for only one~~

MVI A, 90H

DCR B

~~not effect~~

if. carry



1000 1000 -1 = CF

value to obtain 100μs delay

$$TCA = n \times T \times NIO$$

$$NIO = \frac{100 \times 10^6}{nT} = \frac{\text{Delay time}}{n \times T}$$

$$\text{Counter Required} = \frac{100 \times 10^6}{18 \times 0.5 \times 10^{-6}}$$

$$\frac{100}{1} = 11$$

$$= 084$$

~~z~~

Q. What is the memory space of 8086 CPU - 1MB 20

Q. Which of the following is used as interface chip for Data Transf.

a) 8251 b) 8253 c) 8255 d) none (8255)

Q. If an 8086 CPU has clock freq 5MHz, 18 Kbytes required to execute an instruction of 18T is.

a) 8 μs b) 3.6 μs c) 4 μs d) 6 μs

$T_C = 0.2 \mu s$

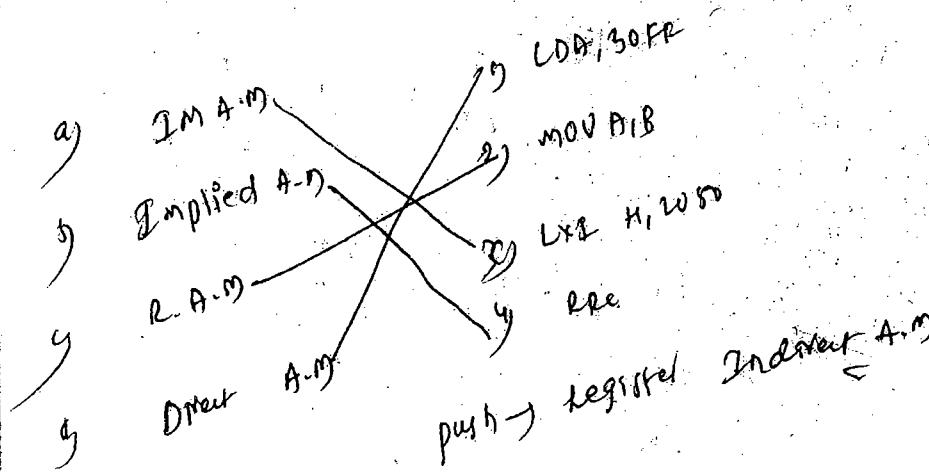
$$= 18 \times 0.2 = 3.6 \mu s.$$

Q. A memory of system size 82KB is required to be designed by using memory chips which have 12 Address lines and 4 data lines each. What are the no. of such chips required to design 16M memory system?

No. of memory chips required = Required capacity / capacity of one chip

$$\frac{82 \text{ KB}}{32 \times 4} = \frac{8}{32 \times 10 \times 8}$$

$$= 16$$



a) 4 b) 8

Consider
data bus
memory

$\approx 2^{\text{st}}$

for Data Transfer

if stack pointer or 8085 RSP is loaded with ABCD4 at the execution of the required instruction what will be the value of S.P.

as required to

push Bsw

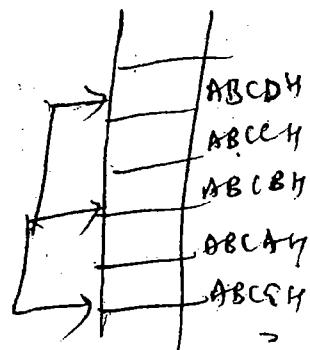
XTHL → change of only

push D

JMP FCTOH

a) ABCD4 b) ABCAH

c) ABC9H d) ABC8H



program flow

use designed

int 4-data

required to

Q. Atm 8085 RSP is executing. If following times the instruction NOP will be executed.

MVI A, 10H → A=10H

MVI B, 20H → B=20H

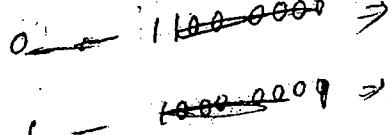
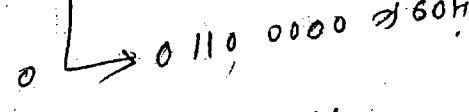
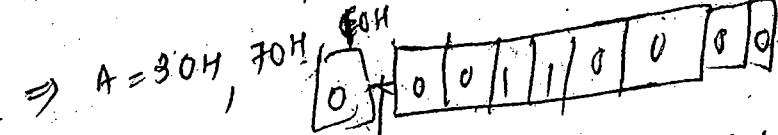
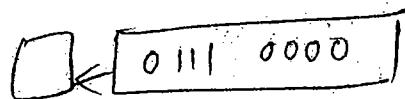
BACR: NOP 1

ADD B

RCL

INC BACK

RET

0 110 0000 \Rightarrow E0H

a) 4 b) 3 c) 2 d) none

Q. Consider the following program given below. It transfers a block of data from one place to another place. What is the total no. of memory access including instruction fetches carried out.

MVI C, 0BH - 1 X2 \rightarrow ^{memory}
cycles

LXI H, 2400H - 1 X3

LXI D, 3400H - 1 X3

(1 = mov A,D - 1 X2

SFAT D → H X X

INL L → H X X

INR E → H X X

DCR C → H X X

JNZ U → H X X → 3V

10X3 + 1XV =

10³ memory cycles

w.A-P: for multi digit

Data segments

n₁ dB, 01H, 02H, 03H, 04H, 05H
 n₂ dB, 01H, 02H, 03H, 04H, 05H
 n₃ dB 05H dup<00>

Data ends

Code segment

```
start: MOV AX, Data
       MOV DS, AX
       MOV CX, 00005H
       MOV SI, offset n1
       MOV DI, offset n2
       MOV BX, offset n3
```

Addition

3000	11	1000		
3001	12	1001		
3002	13	1002		
3003	14	1003		
3004	15	1004		

01	2000
02	2001
03	2002
04	2003
05	2004

BX → B-P

DX → Data pointer

x MOV AL, [BX]

ADD AL, [DI]

MOV [BX], AL

INC SI

INC DI

INC BX

DCR CX

JNZ >

(NT 3)

A
B
C
D
E

B) find character 'R' from string and delete that
 and re write next value

Data S

n₁, 0B (Hyderabad 9)

Data ends

DD

→ It is USA!

→ It is F

3 types of com

j Simplex

y Half du

o Full du

T

It is

8 Data

C/F support

start : mov SI, offset

mov AL, [SI]

mov AL, RI

2: cmp AL, [SI]

JE - X

cmp AH, [SI]

JE - ④

INC SI

jmp ②

④ INC SI

MOV BL, [SI]

DEC SI

MOV [SI], BL

jmp w

⑦ INT 14H

8251 Specifications

- It is USART (Universal Synchronous and Asynchronous Ex and Rx)
- It is for serial communication

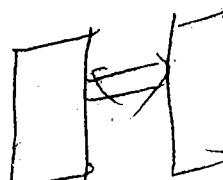
3-types of comms → Rx

- ↳ Simplex → Only one way of comm. Re from Tx to Rx) Ex: Computer to printer
- ↳ Half duplex → Two way of comm is possible but not simultaneous Ex: walky-talky.
- ↳ Full duplex → Two way of comm is possible simultaneously.

i) It is stand alone IC, Dual in line package, +5V DC

j) 8 Data Lines (D0-D7)

↳ used to form bidirectional buses



k) It supports Synchronous and Asynchronous Ex and Rx

H
Y
D
E
R
A
B
I
D

