

Microprocessor

- 1 Byte = 8 bits / 2 Hex
- Word = 16 bits / 4 Hex

Ques. Why hex preferred in Computer System?

Ans:

Binary has hexadecimal digits ($0-F$) → $0000-1111$ → all 16 possibilities utilized in hex
 decimal digits ($0-9$) → $0000-1001$ → only 10 possibilities utilized in decimal No. system.
 i.e., binary 1010 to 1111 are not utilized in decimal No. system.

Also, decimal required conversion from decimal → binary to understand Computer language easily.

- * 10 Address lines have 2^{10} . Address = 1K
- * 20 Address lines have 2^{20} . Address = 1M
- * 30 Address lines have 2^{30} . Address = 1G

Address:

Eg. $(8K \times 8)$

$2^{3} \times 2^{10}$ and 8 Data lines.

Address lines:

Eg. $(32K \times 8)$

$2^{5} \times 2^{10}$ and 8 Data lines. ⇒ 15 address lines & 8 Data lines.

Eq $32M \times 8$ mean 25 address and Data lines.
 $25 \times 2^9 = 8$ Data
 $\underline{25}$
25 address lines

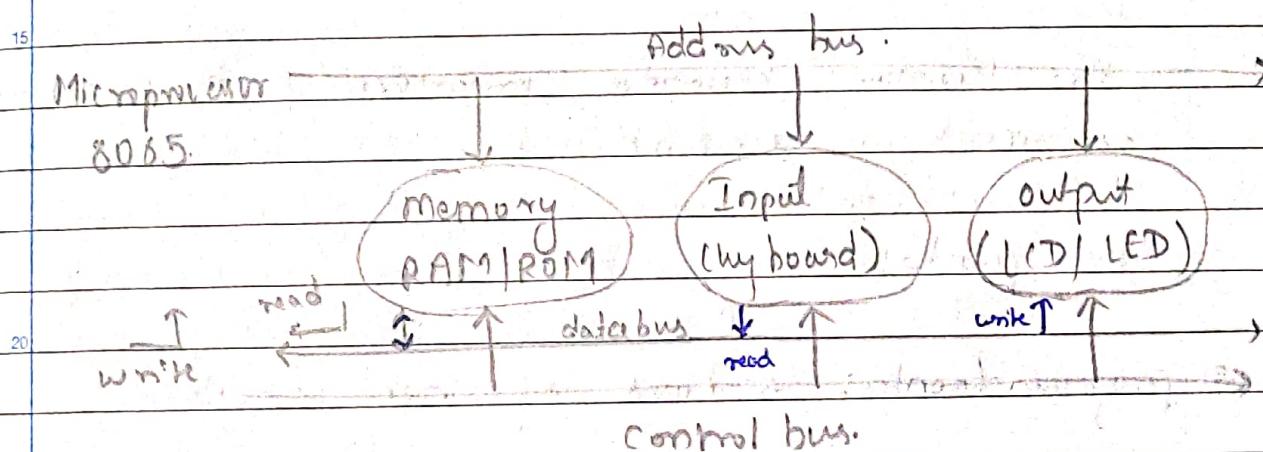
• Buses in 8085

connection b/w, processor, memory, I/O device are done using buses.

• Buses:

J J

Address bus & data bus & control bus.



- **Address bus:** address information will not be sufficient. There are total 16 bits address buses in 8085 microprocessor. Those buses are used to identify the address of memory and I/O peripherals.

(A₀-A₁₅) unidirectional buses.

• Control bus:

- provide control to I/O, memory devices unidirectional. provide memory read, memory write, I/O read and I/O write operation.

* Data bus

- 1) There are D₀-D₇ data bus in 8085. (8 data bus)
 - 2) Bidirectional buses. direction of buses is based on read and write.
- 5 RDM (read)
RAM (r/w)
I/O (r)
Output (w) Microprocessor → Output peripheral.

* Architecture of 8085 microprocessor:-

1) General purpose registers:

A, B, C, D, E, H, L uses for storing the digit input given by user.
Or Programming can do programming by using two 8 bit registers (Size = 8 bits)

2) Stack pointer: indicates top of stack (Size = 16 bits)

3) program Counter: address of next instruction.

4) Incrementor / Decrementer : I → program Counter to jump on next instruction.
Address latch 16
J/D → for stack pointer.

5) W/Z (Temp Reg): Not available for programmer.
Utilized by 8085 microprocessor itself.

30 (Refer video No. 8)

Draw Diagram of 8085 microprocessor

Refer ushah notes

* Flag Registers in 8085 (5 Flags), [Important for making program]

1) Z helps in get to know the status of the program after every instructions

2) Flag and ALU are directly connected
3) after execution of each instruction Flag Register are updated.

S Z AC P Cy
D7 D6 D5 D4 D3 D2 D1 D0

Ex. MVI A, CCH
= A = 5CH 1100 1100
MVI B, E6H
= B = E6H 1110 0110
ADD B
A = B2H 1011 0010

① Cy = 1, carry is generated after execution. Cy = 0 (Then not generated)

② P = 1, Result has even 1's
P = 0, Result have odd 1's

③ AC = 1, when result have carry from D3 to D4 bit. (Mibble to Mibble carry)

④ Zeroflag = 1 Result is zero after execution
Zeroflag = 0 Result is non-zero after execution.

⑤ S (sign flag) → Represent D7 bit | D7 = 1 (-ve) No. / S = 1
| D7 = 0 (+ve) No. / S = 0

* Programming Model of 8085

* Different Registers of 8085

- 1) (A) This main register of 8085, default Operand of ALU.
o all processes are performed by default with accumulator register. (8) bit

(2) Flag Register (8) bit : (S, Z, P, AC, P, CY)

(3) general purpose registers: (A, B, C, D, E, H, L) used for programming
also, BC, DE & HL in pair for programming.

(4) Program Counter: (16bit): PC $\xrightarrow{\text{Points}}$ to address where next byte to be fetched
or
stores next instruction going to execute.

(5) Stack pointer (SP) (16bit):

points to Read / write memory called stack.
Top of stack represented by SP. (16bit)
as known as memory pointer.

Summary:

- 1) Total 7 registers: used for programmatic programming.
2) PC: program counter increment by one execution of each instruction. (automatically by one).

* Control Signals of 8085

① I/O / M

if flag is 1 then, IO operations should done by 8085.

if flag is 0 then, Memory operation should done by 8085.

② RD (Read)

active low signal.

read operation are performed by I/O or memory.

③ WR (Write)

active low signal

write operation are performed

using these 3 control signal we can generate four control signals.

mem-read, mem-write, IO-read & IO-write.

I/O/M

RD

WR

Control Signals.

0

0

1

memory - Read MEMR

0

1

0

memory - write.

1

0

1

IO - Read IOR

1

1

0

IO - write. IOW

NAND gate.

o1 (memory - write)

active low

I/O M

RD

WR

o2 (I/O - read)

Wrong

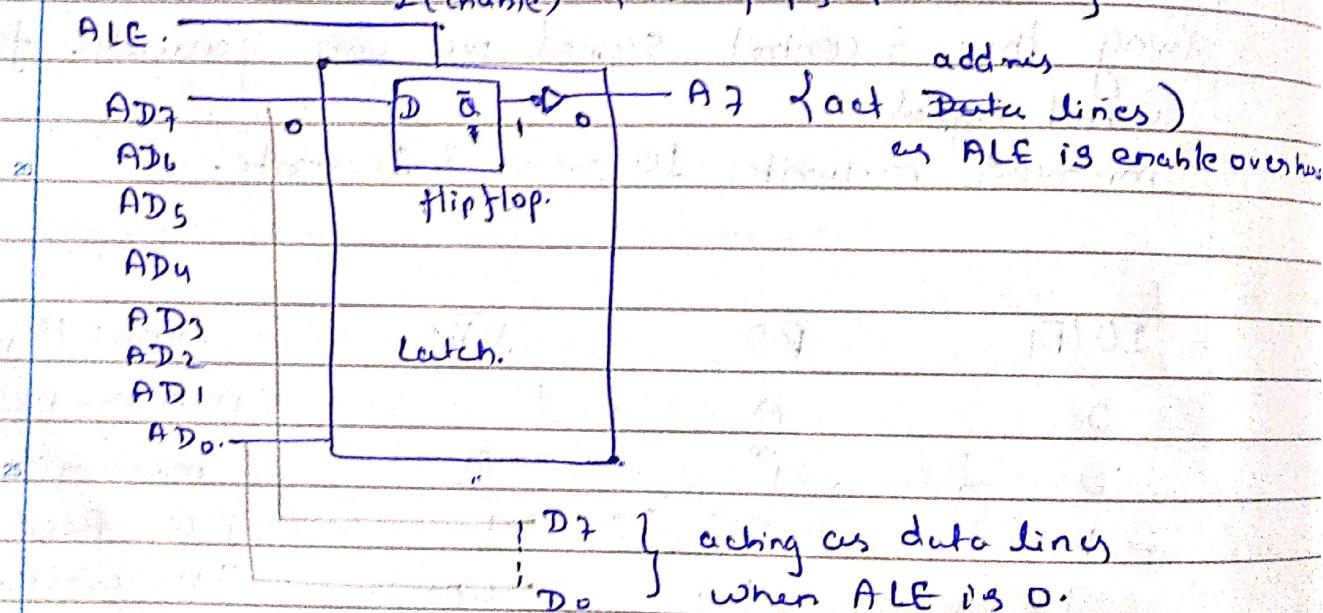
Signal of I/O M should remain same through circuit using an and gate.

• Address Data Demultiplexing (Time multiplexing of Address & Data)

- 8085 have 16 address lines & 8 data lines.
- higher bit byte of address ($A_8 - A_{15}$) right
- lower byte address ($A_0 - A_7$) left side
- using ALE (address latch enable) → Separate data & address lines.
- $ALE \rightarrow 1$ (address lines) $A_{D0} - A_{D7}$
- $ALE \rightarrow 0$ (data lines) $A_{D0} - A_{D7}$

A₁₅
A₁₄
A₁₃
A₁₁
A₁₀
A₉
A₈

2 (Enable) {Here flip flop is enable}

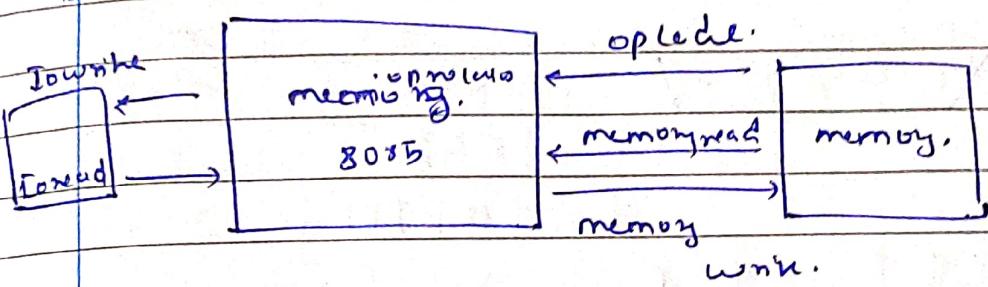


- ① we're going pulse give by ALE at that time $AD_7 - AD_0$ represent address lines.
- ② when ALE act as active low at that $AD_0 - AD_7$ act data lines.

* Machine Cycle of 8085 Microprocessor

→ There are 5 different cycle with micro processor.
each machine cycle, microprocessor transfer one byte.

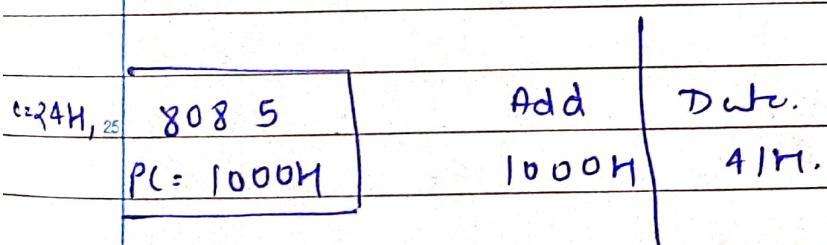
- ① Opcode fetch (9T/ GT states)
- ② Memory read (3T states)
- ③ memory write (3T states)
- ④ IO read (3T states)
- ⑤ IO write (3T states)



	IOM	91	90	Control Signals
opcode fetch.	0	1	1	$\overline{RD} = 0$

* Timing Diagram of MOV instructions:-

MOV B,C (opcode = 41H)



MOV B,C execution need only
opcode fetch machine cycle.

After execution $B = 24H$.

MOV A, E LHL D 1050H
XCHL

L = [1050H]
H = [1051H] 4956

Page:
Date:

Ques. addition 16 bit Number

$$1st \text{ No} = 1050 - 5EH 1051H$$

$$2nd \text{ No} = 1052 - 1053H$$

HL = DE pair

$$D = [1051H] [1050H] E = 1054 - 1055H$$

$$H = [1053H] L = [1052H]$$

DE

=

1051H (HL)
XCHL

(1050H)

LHL D 1052H

LHL D

XCHL

1051H

H = [1051H]

L = [1050H]

This simulation
will exchange.

HL with DE
pair

MOV A, E ; A = E

ADD L ; A = A + L H = [1053H]

E L = [1052H]

L = A → final value

D = [1051H]

E = [1050H]

HL DE

Ques.

division Operation.

divisor = 1050H

dividend d : 1051H.

divisor = 1050H
dividend d : 1051H.

Store Quotient : 1052H

Remainder: 1053H

Ans.

LDA 1050H

A = [1050H]

MOV B,A

B = A (divisor)

LDA 1051H

A = [1051H] (divident)

Storage
Register

\leftarrow MVI C, 00H

C = 00H

loop:

CMP B

(divident < divisor)

Compare $A < B, A > B, A = B$.

A compare with B.

JNZ exit

$B > A \quad A \neq B \quad 00 \rightarrow \text{com.}$

JNC exit

exit $A \neq B \quad 00 \quad 1$

ADD B

E

$A = A + B \quad 00 \quad 010$

JC exit / END

~~B = B~~

SUB B

$A = A - B$

IMR C

12-5

JMP loop

Quotient
should be
by 1.

At Remainder will be in A as we are subtracting
divisor from dividend in loop.
divident.

STA 1053H \rightarrow Store remainder.

MOV A,C

$A = C$

STA 1052H \rightarrow Store Quotient

HLT

H=10514

1050H

10514

MOV D, 00H

LXI H, 1050H

MOV B, M

INR A, JMX

MOV C, M

MOV A, C

loop SUB ③

divisor

divident

B

C

divisor

divident

C

D

short 3 times

5 from 17.

CMP B

A Compare divisor B

17

-15

Divisor < divident

divident

②

Divisor > divident

5 > 17 when

B becomes greater than A.

Stop subtraction

If ②

(B) < A

divisor < divident

MVI C, 00H

LXI H, 1050H

MOV B, M

LXI H, 10514 → digit 10

Divident

10/2

10 < 2

divident < divisor

2/10

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

A=0

D=0 → constant Subtraction
LWIPB
divident
divident

1050H

10514

CMP B

A < B

1 A < B

True

SUB 2

10

8

2

6

-2

4

-2

2

-2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

1/2

30

Camlin

Program 4 Multiplication

Page:
Date:

5) 3 times
17

```

LDA 1050H (Input divisor)
MOV B,A
LDA 1051H (Input dividend)
MVI C,00H
for Quotient
    CMP B
    JC exit
    SUB B : A = A - B
    INR C
    JNE loop
exit: STA 1053H
      MOV A,C
      STA 1052H
      HLT
  
```

dividend > divisor

dividend < divisor

A > B 0

A < B 1

(exit) dividend < divisor

B > A

A < B

Ques Find mux of 5 digit stored at 1050H to 1054H
store mux at 1060H?

1 digit from.

LXI H, 1050H
MOV A, M

H = 10H L = 50H

MVI C, 04H } → 1 data already consider.

loop: INX H
CMP M

HL = 1051
L(M)

A > M 0
A < M 1

JNC EXIT SKJP

MOV A,M

DCR C

JNZ loop

Camlin

STA 1060H HLT

Store data in reverse order.

Q1 Array of 5 data given at 1050H to 1054H
Store it in reverse order at 1060H to 1064H.

MN

Pointers of HL for 1050H array

LXI H, 1050H

H=10H, L=50H

= Pointers of DE for 1060H array

LXI D, 1064H

D=10H, E=64H

MVI C, 05H

→ Store at lower more

10

Loop:

MOV A, M
IMXH
STAX D
DCX D

|| Taking data from HL pointer

HL=1051H || Increment HL pointer.

DE=A || Store Accumulator into DE

|| Decrnt DE pair

15

DCR C

Important decrement loop.

JNZ loop

HLT

20

MVI C, 05H

This is only worn

loop: 1050H 1051H

MOV D, M

EDAX C

↓ point

HL

DCX H

point. 0H

STAX D

HL

DCX D

pair

DCR C;

for HL
pair loop

JNZ loop

MOV A, M

30

HLT

10 data 1050H to 1059H

Page :

Date :

Addition of Array.

store at 1060H & 1061H
+
ans
↓
carry

1050H

[23 | 33 | 54 | 65 | 78 | 99 | 97 | 74]

↑ ↑

INX H

points of HL pair

lower byte

Answer in ans

LXI H, 1050H

H = 10H L = 50H

MOV A, M

A = [1050H], 23

store all
carry

MVI B, 00H

B = 00

MVI C, 09H

C = 09

Counts

loop: INX H H = [1051H]

ADD M

A = (A) + (H)

(JNC) skip

23 + 33
 =

INR B

DCR C

JNZ loop

→ Skip:

STA 1060H

↓ lower byte of add.

MOV A, B

→ carry

STA 1061H

↓ high byte of

addition.

HLT

MCU

30

Camlin