Date ___/___/___

* Microprocessor :- Its a controlling unit of micro-computer capable of performing Arithmatic & Logical operations, fabricate on a small chip.

It consists of an ALU, Register Array and control unit.

* ALU → Performs Arithmatic & logical operation on the data recieved from the memory or on the Input device.

* CU → Controls the flow of data & instructions within the computer

* Register array → It is used to execute the program

* Registers are made of flip-flops

* WORKING OF μP
(1) fetch Instructions (2) decode (3) execute

Size of Memory.

(1) 16 bit Address bus } in 8085 μP
8 bit data bus

formula :- $2^A \times D$
A = address bus size
D = data bus size.

2^{10} = kilo
2^{20} = Mega
2^{30} = Giga
2^{40} = Tera

Date __/__/____

Architecture

* features of µP :- (1) Cost effective
(2) Reltability (3) Verstalid (4) Low power consumption

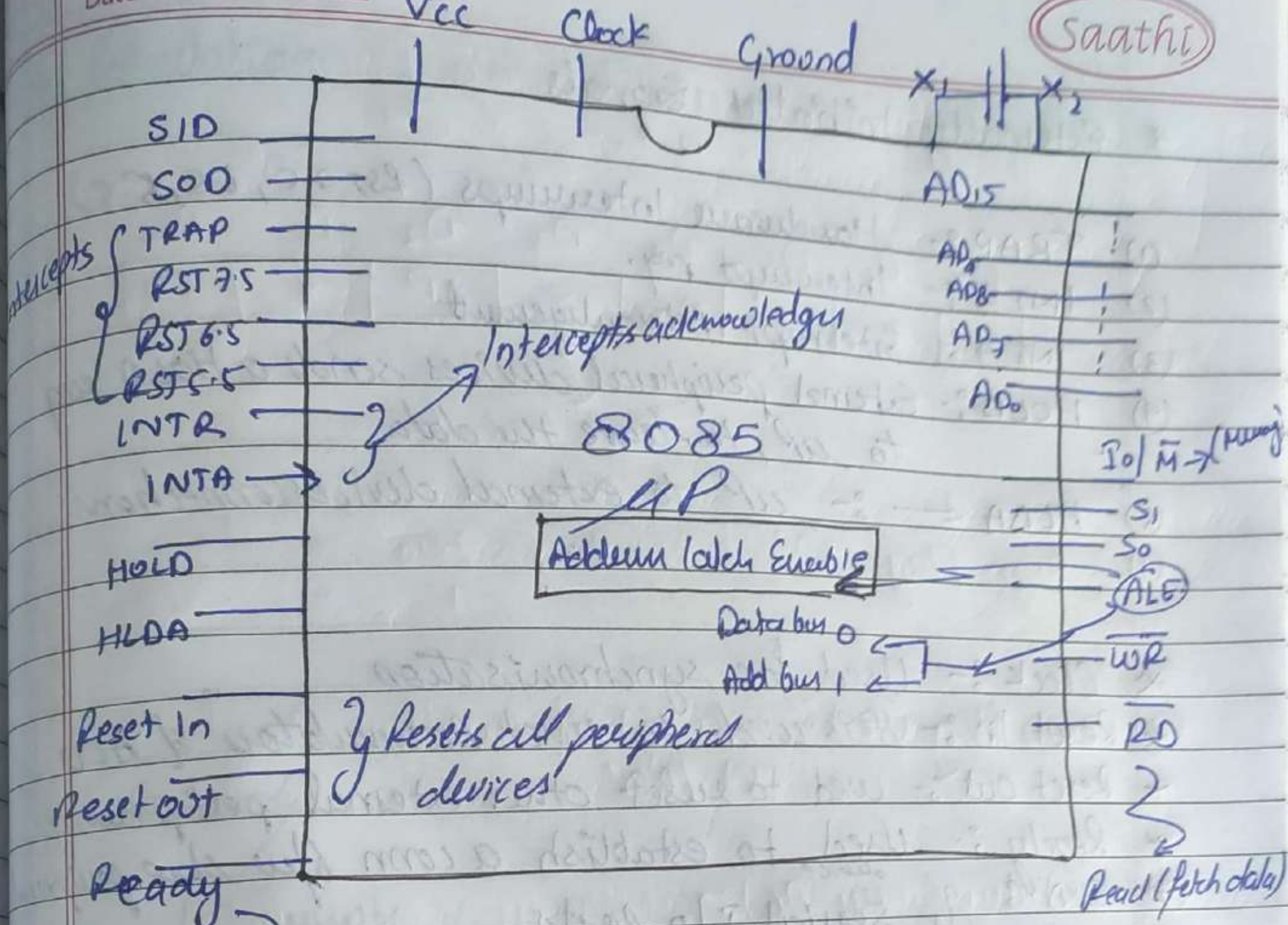
* features of 8085 µP

(1) 40 pin IC
(2) 16 bit Adderess bus
(3) 8 bit data bus
(4) NMOS technology
(5) frequency - (3 MHz - 5 MHz)
(6) hexadecimal number system

Architecture

External Architecture → Internal Architecture

External Architecture
- → 8085 µP (40 pin)
- → Data bus (AD_0 to AD_7)
- → Adderess bus (AD_0 to AD_{15})
- → Control & status Signal ($\overline{RD}$, $\overline{WR}$, $AL\bar{E}$, S_0, S_1, $I_0/\bar{M}$)
- → Power supply & frequency Signal (V_{cc}, Clock, Ground, X_1 & X_2)
- → Externally Initiated signal



→ Used to establish connections of sync b/w slow peripheral devices & itself

* Data bus :- (AD_0 - AD_7) Carries least Significant 8 bit

* Address bus :- (AD_8 - AD_{15}) Carries most Significant 8 bit (I/O)

* Status Signal :- Used to Identify the nature of signal

* Read ($\overline{RD}$) :- Read operation (fetch)

* Write ($\overline{WR}$) :- Write "

* $IO/\bar{M}$:- if 1 → I/O
0 → Memory

* S_0, S_1 :- Tells the status of current operation.

* Externally Initiated Signals

(1) TRAP :- Hardware Interrupts (RST 7.5, 6.5, 5.5)
(2) INTR :- Interrupt req.
(3) INTA :- Interrupt Acknowlegement
(4) HOLD :- External peripheral devices sends a HOLD req to uP, to leave the data
(5) HLDA ← :- uP to external devices connections.

* Clock :- Used for synchronisation
* Reset In :- It is used to reset Accumulator & ALU
* Reset Out :- used to reset other external perip devices
* Ready :- Used to establish a conn b/w slow peripheral device
* SID :- } serial I/o ports.
* SOD :-

Interrupts
* Maskable
* Non-Maskable
* Vectored
* Non-vectored
* Based on trigger
* Based on Priority.

Interrupts	Based on Priority	Based on Masking	Based on Vector add
TRAP (RST 4.5)	high	Maskable	0024H
RST 7.5	↓	Non-Maskable	003CH
6.5		"	0034H
5.5		"	002CH
INTR	low	"	Non-vectored

Date __/__/__

* Calculation of vector address

RST 7.5

= 7.5 × 8

= 60.0

then convert 60 to hexadecimal

16	60	3
	12	

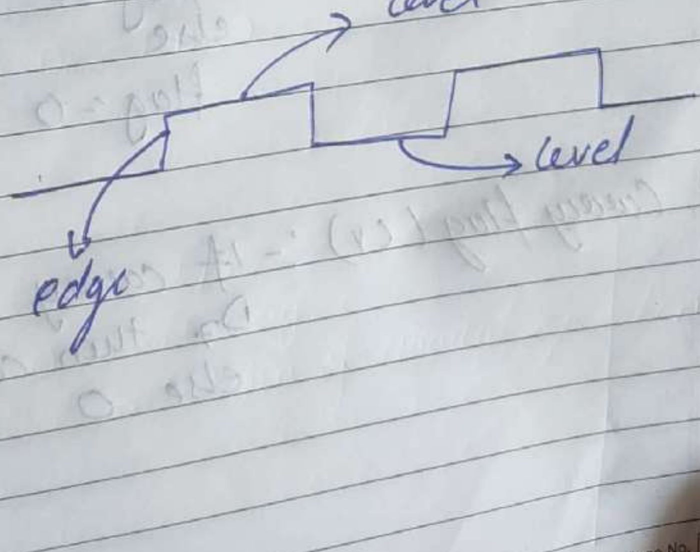
12 = C

⇒ 3C

Vector add of RST 7.5 = 003CH

Similarly, Vector add of RST 6.5 = 0034H

Interrupts	based on trigger
TRAP	Level & edge
RST 7.5	Edge
6.5	Level
5.5	" ~~edg~~
INTR	"



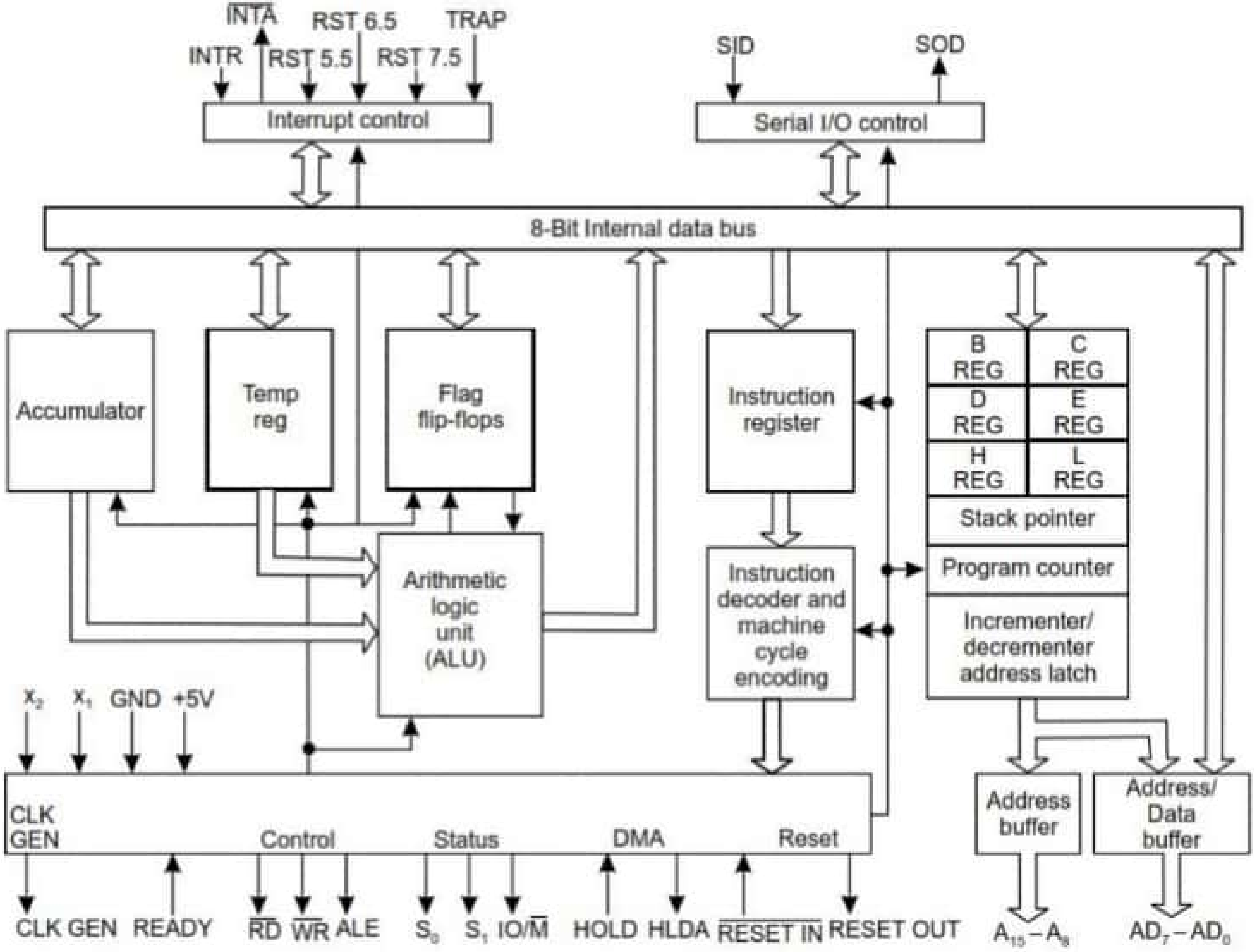


Fig 1.1 Hardware Architecture of 8085

Date __/__/____

* flag Registers :- used to store of control status
→ It is of 8 bit

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z	x	AC	x	P	x	CY

Don't care

(1) Sign flag (S):- If M.S.B is zero, result = +ve
" " " 1, " = -ve

(2) zero flag (Z) :- sets to 1, if result is stored in Accumulator
set to '0'

(3) Auxillary carry (AC) :- If carry is generated from any bit its transferred from D_4 to D_3, the Aux carry = 1 otherwise 0.

(4) Parity flag (P) :- If no. of 1's = even then
flag = 1
else
flag = 0

(5) Carry flag (CY) :- If carry is generated in D_7 then carry flag = 1
else 0

__/__/__

Interrupts /

Book
Gaonkar

Internal Architecture

(*) Instructions :- Commands to the microprocessor to perform task on specific data.

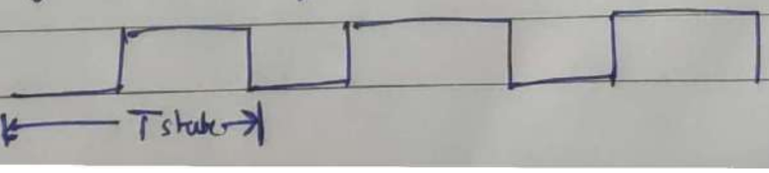
- ⊙ Opcode :- Task which is to be performed
- ⊙ operand :- Data which is to be operated upon.

- ⊙ Instruction Word Size :- No. of byte taken by the instruction to execute the operation.

- ⊙ Machine Cycle :- It is the time taken by the μP to execute an instruction

→ Fetch Machine Cycle.
→ Memory Read Machine cycle
→ Memory write Machine Cycle
→ I/O machine cycle
→ Bus Idle Machine cycle

* 1 Machine cycle consist of a 3T to 6T



- Fetch Machine Cycle :- Consist of 4T state
- Memory Read M/C :- Consist 3T state
- Memory write M/C :- Consist 3T state
- I/O Read M/C :- Consist 3T state.
- Bus Idle M/C :- Only used for DAD instruction ⟷ double Add
  :- 3T state

Special Instruction (consists 6T state)

C :- Call & Return
R :- Restart
I :- INX & DNX
S :- SPHL & PCHL
P :- PUSH

* Total time to execute Instruction

$$\boxed{\text{Total time} = n \times t_{clk}}$$

$$t_{clk} = \frac{1}{f}$$

Ques The clk freq of 8085 µP is 5MHz. If the time required to execute an instruction is 1.4 µSec then no. of T state needed for executing the instruction is

a) 1

b) 6

[✓ c) 7]

d) 8

$f = 5 MHz$

$T = 1.4 \mu Sec.$

~~1.4×10^{-6} t~~

⑥ Addressing Mode

The method by which the address of source of data or address of decimation of result are given in instructions are called addressing.

⇒ Register Addressing : Data is stored in register of name of a register are given in instruction as operand.

MOV A B

⇒ ~~Direct~~ Immediate Addressing :- • No register is required
• Directly data given in the instruc as operand.
Ex :- MVI A, 24H

⇒ Direct Addressing mode :- Address of data directly given in instruction as operand

Ex :- LDA 2   LD A 2500

(Load into accumulator)

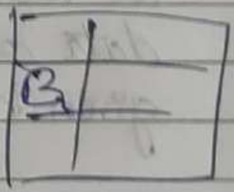
⇒ Indirect Addressing Mode :- Address of data is stored in register pair, and name of a register given in instruction as operand.

Ex :- MOV B, M

Mem will alway point in HL pair of will move to B

H = 90
L = 20
D = ?

9020



⇒ Implicit Add Mode :- No operand is required

Ex :: NOP
HLT

Mov D, M   2 bytes
M/C → 3 (F + R + W)
(4T) (3T(3T)

Instruction	Iws (Instruction word size)
Register Register Pair No operand	1 Byte
8 bit data 8 bit address	2 Byte
16 bit data 16 bit address	3 Byte

① 8 Bit data transfer Instruction.

1 1 4

(1) MOV Ro, Rs (Inst / MC / T-state)

→ WS ⇒ 1 byte

→ operation ⇒

→ Addressing Mode ⇒ Register addressing

~~T2 M/C ⇒ 4T~~ T state → 4T

② MVI R, 8 bit data → { data is directly transfered }

- WS → 2 byte
- Operation →
- Addressing Mode → Immediate
- T state → 4T ~~+3T~~ (F ~~+R~~)

③ MVI B, 28H

- WS → 2 byte
- op →
- Add mode → Immediate
- T state → 4T + 3T (F + R)

③ In 8 bit port address

~~WS~~

~~Operation:-Content of Accumulator~~ will display

~~Add mode~~

WS

Operation:- When this is executed the content of Input code whose add is given is Instruction as operand will be

Stored in accumulator.

Add mode :- Direct.

T-state :- 4T + 3T + 3T (F + R + R)

④ Out, 8 bit Port address

ws - 2 byte

op -

Add mode - direct

T-state - 4T + 3T + 3T (F + R + R)

⑤ HLT

ws - 1 byte

M/C - 1 (4T) fetch

⑥ NOP

ws - 1

MC → 1 (4T) fetch

__/__/__

AIM :- To exchange the contents of 2050 (add) & store the result. (Swapping)

*(Iske aage ka abhay se mangleना)

Program

```
LHLD 2050   [Load HL Pair Load]

      (00)  (00)
MOV    A     H                 L → 2050 → 03
      (03)      08
MOV    H     L                 H → 2051 → 04
      (04)      00
MOV    L     A


SHLD 3050
HLT
```

Instructions	M|C	T-state	add mode
MOV B, C	1 (	4 T	Register
MVI D, 20H	2	7 T (F+R)	Immediate
MVI M, 40H	3	10 T (F+R+W)	Indirect
NOP	1	4T	
IN 20H	3	10T (F+R+R)	Direct
OUT 20 H	3	10T (F+R+L)	Direct.

Scanned with CamScanner

* Write an instruction to display the content of register H at output 00H

```
MOV A, H
~~SHLD~~
OUT 00H
```

Qus

A = 97H B = A9H

~~ADD A, B~~ AND A, B

Carry = 1
Sign

10010111
10101001
~~11111111~~
10000001
8 1 H

AND A, B
81 H

Sign = 1
Z = 0
AC = 0
PF = 1
C = 0