

MICROPROCESSOR AND MICROCONTROLLERS

PAGE NO. ①

DATE :

MICROPROCESSOR -

A programmable device similar to CPU of a computer which is responsible for computing and taking logical decisions.

- * Semiconductor device * Manufactured using LSI or VLSI techniques

Application: ① in office automation and word processing

② Computer Graphics (2D, 1D & 3D)

③ Airlines and Railways (Reservations, tracking) LAN MAN satellite

EVOLUTION: from SSI to VLSI technology

① 4 bit - Intel 4004 (1st), PMOS technology,

[8085] ② 8 bit - intel 8008 (1st 8bit) PMOS, Powerful, fast, compact

③ 16 bit - Intel 8086

④ 32 bit - Intel 80386 (used → data Processing, LG IS)

⑤ 64 bit - SUN's SPARC, Intel's Itanium

(super fast, most powerful, super comp).

8085 MICROPROCESSOR

* ARCHITECTURE:

reason

① Intel 8085 is **8 bit** Microprocessor.

② It is capable of addressing **64K** of Memory

③ It is manufactured on a single LSI chip using NMOS technology.

④ It is a 40 Pin IC Package

⑤ uses +5 V Single D.C. Power Supply

⑥ Operate with 3.14 MHz Single phase clock.

⑦ Clock cycle is of $320 \mu\text{sec}$.

⑧ Divided within 3 blocks

① ALU ② Registers ③ Timing control

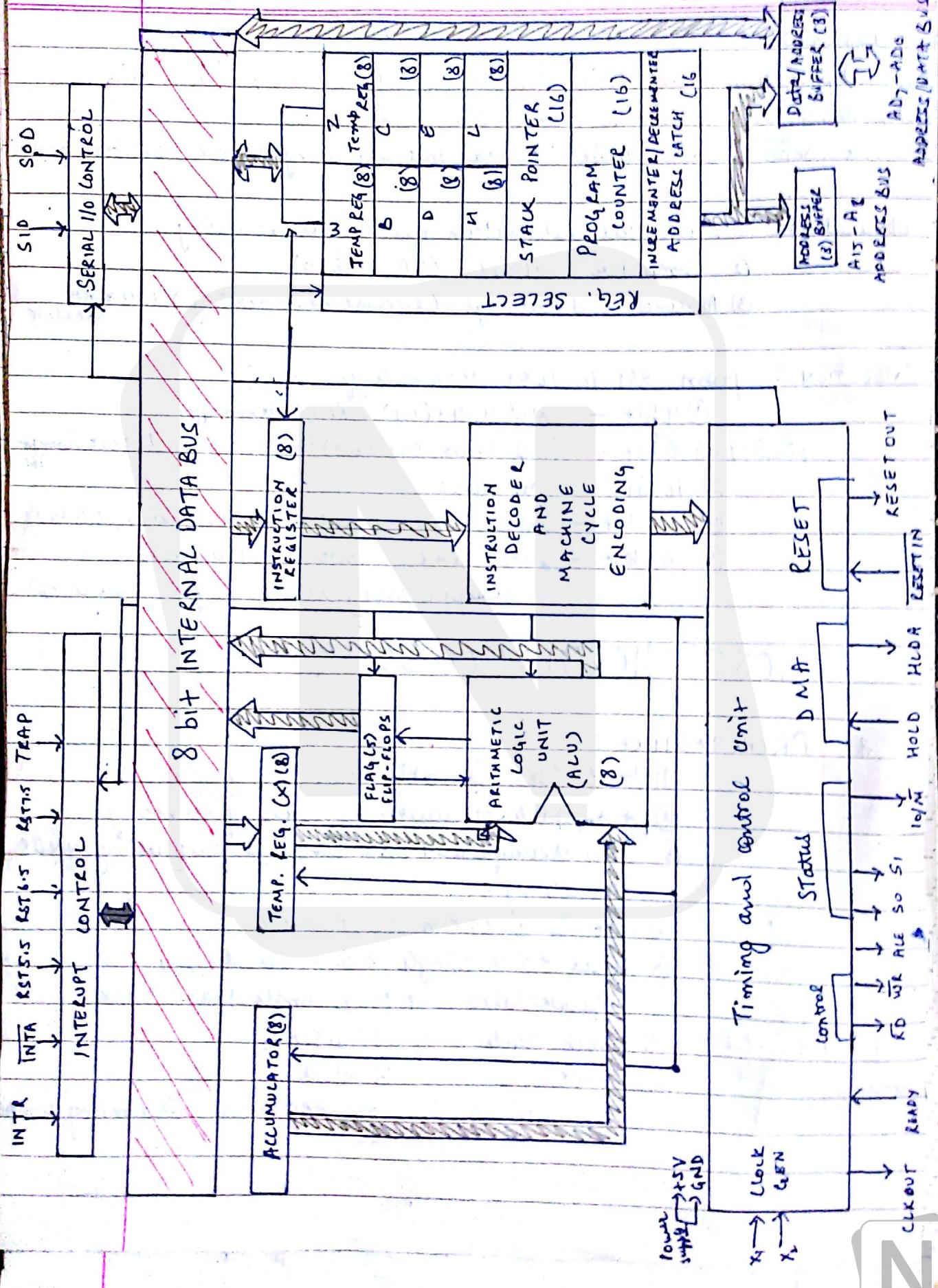
320 μsec



8085

PAGE NO. (2)

DATE :



① Buses :

Group of lines used for interconnecting various building blocks within a computing system. These lines transmit electrical signals and provide common means of communication. Each is capable of transmitting one electric pulse, represent (1 bit). Bus can be unidirectional as well as bidirectional.

→ ADDRESS BUS :

→ 16 Address lines A₁₅ - A₈ upper address line
A₇ - A₀ lower address line

→ If no. of address lines are 'n' then the microprocessor is capable of addressing ' 2^n ' memory locations.

→ Each memory location is identified by a binary no. (address)

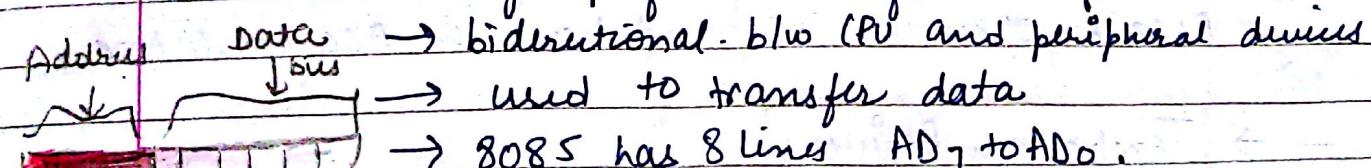
imp → 8085 → 16 bit → $2^{16} = 65536 \rightarrow 64\text{K (approx) memory locations}$

→ $(65536)_{16} \rightarrow (FFFF)_{16}$
so, 0000H to FFFFH Address Range of Memory

whenever an address of 16 bit is transferred then, 8 bits are transferred to A₁₅ to A₈ and 8 bits are transferred to A₇ to A₀, so, the address memory location is selected.

→ DATA BUS :

→ group of 8 lines for data flow



→ As add. and Data are transferred at diff time, same 8 pins AD₇ to AD₀ are used for transferring 8 LSB's of 16 bit add as well as 8 bit data.

→ AD₇ - AD₀ are called time multiplexed and time shared add. data bus.

4 nibble \Rightarrow 1 nibble
 8 nibble \Rightarrow 2 nibble \Rightarrow 1 byte.
 1024 bytes \Rightarrow 1KB
 1024 KB \Rightarrow 1MB \Rightarrow

PAGE NO. (4)
DATE

- 8085 has 8 data pins, so it can transfer Max. 8bit data in parallel. So, transferred from 00H to FFH.
- largest No. 1111 1111.
- Determines word length, register size of microprocessor.
- 8085 has 8 data lines, so 8085 is 8 bit MP.

INTERNAL DATA BUS

8 bit data bus is provided to transfer data b/w outer data lines and inner registers in Bidirectional direction.

General purpose Registers:

- 6 Eight bit Registers BC, DE, HL.
- Each Register contains 8 Flip-Flops.
- Each Register can store 8 bit of data (1 Byte).
- To store large data upto 16 bits, they are used in Pairs BC, DE, HL Pair. \rightarrow each can store 2bytes.

Accumulator

8bit AC
8bit Regi

- 8bit Register contains 8 F/F, so can store 8 bit of data (Max).

Importance: ALU operations, MP take first 8 bits from Accumulator, operation is performed, then 8 LSB's of result is stored back in Accumulator

[FACT]

Temporary Registers

\rightarrow 3 registers are present X, W, Z each of 8bit and can store Max 8bit memory data.

- Not accessed/used by programmers, only used to store temporary data while in operation upto 8bit each.
- For 16 bit storage WZ pair can be used to minimize storage.

→ Arithmetic and Logic Unit (ALU)

- 8 bit ALU

- Can perform operations like :

AO
 $\begin{cases} \rightarrow 8\text{ bit binary subtraction, BCD addition} \\ \rightarrow 8/16\text{ bit binary addition} \\ \rightarrow Increment/Decrement of 8/16 bit data \end{cases}$

LO
 $\begin{cases} \rightarrow AND, OR, EX-OR of two 8bit no. \\ \rightarrow Inverting 8bit no. \\ \rightarrow Comparison of 2 - 8bit no. \\ \rightarrow Rotating 8bit no towards Left/Right with/without carry. \end{cases}$

→ FLAG Registers (Program Status word)

S	Z		AC	.	P		CF
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

- 8 bit Register \rightarrow 8 F/F,

- D₀, D₂, D₄, D₆, D₇ \rightarrow are called status flags

- D₁, D₃, D₅ \rightarrow Not used

- Status of result is stored in flag register.

① Carry Flag (D₀) (CF) :

When MP performs addition of 8/16 bits no., result will be max of 9/17 then, last carry is stored in CF !

Similarly for Subtraction (x-y) of 8 bit no.

a) $x > y$, additional borrow = 0, CF = 0

b) $x < y$, additional borrow = 1, CF = 1



* (2) Auxiliary Carry (AC) / Half carry Flag :

Status of AC flag is similar to carry flag except that AC flag will indicate the status of 4 LSB's

* (3) Parity flag (P) :

$P=0$, even
 $P=1$, odd

- Shows the status of 8 LSB's of result in ALU
- $P=1$, if there is odd Parity
- $P=0$, if there is even Parity

(4) Zero flag (Z) :

- $Z=0$, if any one or more bit of 8 bit no, is non-zero. i.e from 01H to FFH.
- $Z=1$, if all the bits of a 8 bit no are zero, i.e 00H

(5) Sign flag (S) :

Sign flag copy MSB of 8 bit LSB's result of ALU

- Result is unsigned, flag is of no use
- Result is out of Range $(+127)_D$ to $(-128)_D$, then flag is of no use.
- Result is in Range $(+127)_D$ to $(-128)_D$, if then
 - If $S=0$, then result is Positive
 - If $S=1$, then result is negative

INSTRUCTION REGISTER (IR) AND INSTRUCTION DECODER (ID)

- Instruction Register (IR) \Rightarrow 8bit register is applied to Inst. Decoder (ID)
- ID has 8 I/Os and 2^8 ID O/Ps (y_0 to $y_{7,ss}$). Each O/P is connected to one controlling clk. \Rightarrow 256 controlling clk. each clk generate signal to perform the particular operation. So, 8085 have 256 diff. operations.

When 8 bit IR is given as I/P to ID, only one ID O/P becomes active (Logic 1) and rest 255 ID O/P are inactive. So controlling ckt connected to the inactive ID O/P generate control signal and a particular operation is executed.

→ PROGRAM COUNTER (PC)

- It acts as a pointer to the next instruction to be executed and always contains the 16 bit address of the memory location of next instruction.
- After the completion of instruction, PC is updated by Processor and points to the next instruction.

→ Stack Pointer (SP)

- Stack is an area of RAM (read/write memory) in which temp. info. is stored in A/F/O basis.
- Add. is assigned in RAM area to first stack entry position, afterwards stack writing in instruction fill memory positions in progressively decreasing addresses.
- SP holds the address of last byte written onto the Stack. Called Top of stack. ~~Top is dt~~
- SP is decremented as data is read out of memory

→ Interrupt Control

- Responsible for enabling and disabling of interrupts. There are five hardware interrupts

INTR →

PS T S.S →

RST 6.5 →

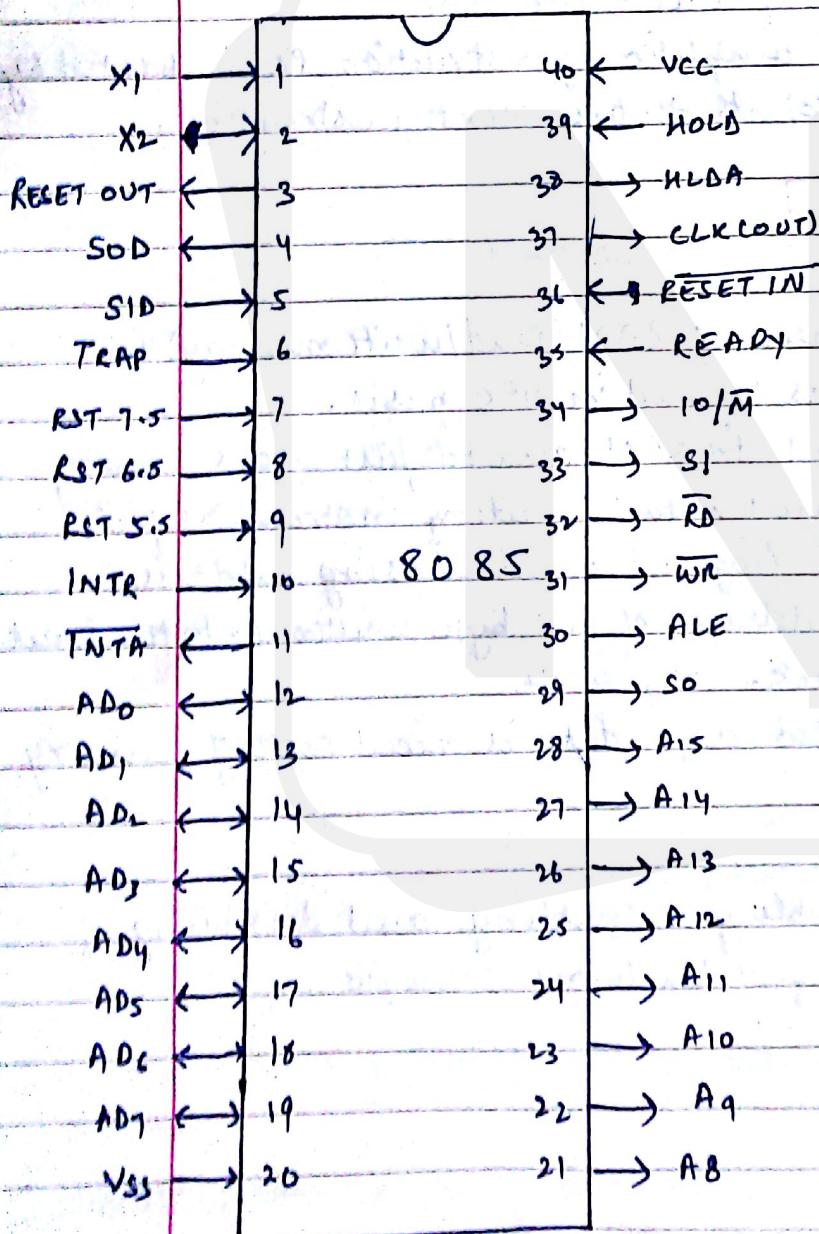
RST 7.5 →

TRAP →

→ Serial I/O Control:

In 8085, Parallel data transfer is possible, but if we want to send out serially/serial data in, this can be done using SIN, SOD pins.

PIN DIAGRAM OF 8085



① X_1 and X_2 (Pin 1 & 2)

→ I/P Pins

→ terminals are connected to external oscillator which drives the internal circuitry of the MP to produce a suitable clock.

→ Crystal frequency of 8085 is 6.28 MHz,
Operating " of 8085 is 3.14 MHz

② Resetout → O/P pin, signal is given out to reset all external devices.

③ SOD → O/P Pin (Serial data out), data sent out serially using SIM instruction

④ SID → I/P Pin, used for I/P data serially. data is in serially using RIM instruction.

* ⑤ TRAP → Non-Maskable Interrupt (Highest Priority) also known as RST4.5
vector location 0024H

RSTS.5 RST6.5 RST7.5 → I/P Pins of MP, (Restart interrupt).

when interrupt is recognised all next instructions will be executed from a fixed location in the memory.

RSTS.5 LOC = 0026H $\leftarrow 8 \times 5.5$

RST 6.5 LOC = 0034H $\leftarrow 8 \times 6.5$

RST 7.5 LOC = 003CH $\leftarrow 8 \times 7.5$

* INTR → (Interrupt-request) Signal, used as general purpose interrupt (lowest Priority), when it goes high the PC does not increment its content, suspends normal sequence of instruction and MP executes interrupt service routine.

INTA → (Interrupt acknowledge bar) → o/p pin of MP, In response to interrupt request an active low signal is sent on this pin to acknowledge interrupt.

AD₇-AD₀ → Address / Data bus, LSB

VSS → Ground Reference $\frac{1}{3}$

AS-AS₅ → Address pin / Bus USB.

S₀, S₁ → S₁, S₀ are status Signal, used to distinguish the various types of operations

<u>S₀</u>	<u>10/4</u>	<u>S₀</u>	<u>S₁</u>	Operation
0	1	1		opcode fetch
0	1	0		memory read
0	0	1		memory write
1	1	0		I/O read
1	0	1		I/O write
1	0	0		interrupt acknowledge.

ALE → Address latch enable, o/p pin of MP, goes high during first clock cycle of a Machine cycle and enables the lower 8 bits of the address to be latched either into memory or external latch.

WE (write bar) → write control signal (active low), o/p pin of MP, when it goes low the data on the data bus is written into selected memory or I/O device

RD (read bar) → ~~■~~ read control signal, o/p pin of MP, This signal indicates that the data is to be read from the selected I/O or memory.



PAGE NO. (11)

DATE :

$I/O/M$ ($I/O/P$ / O/P / Memory), \rightarrow O/P Pin of MP, Status signal which distinguishes whether address is for memory or I/O . When it goes high the address on the address bus is for I/O devices, when it goes low the address on the address bus is for the memory.

If $I/O/M = 1$ I/O device selected.

$I/O/M = 0$ memory selected

Ready \rightarrow pin is used by the processor to check whether the I/O device is ready to send or receive data. If ready is high peripheral is ready and if it is low MP waits till it goes high. MP uses this signal for synchronisation with the slow devices.

RESET IN \rightarrow I/P pin of MP, when active low signal is applied to this pin the MP is reset. The contents of program counter becomes $0000H$, it also disables interrupts.

Ck \rightarrow (Clock O/P) O/P pin of MP, used as the system clock for other devices.

HOLD \rightarrow A high on this pin suspends normal CPU operations. It ~~suspends~~ indicates that another device such as DMA controller is requesting the use of address and data bus. After receiving the HOLD request the MP relinquishes the use of the buses as soon as the current machine cycle is completed. The Processor requires the buses as soon as the HOLD signal is removed.

* NLDA (Hold acknowledgement) \rightarrow O/P of HOLD acknowledgement. It indicates that HOLD request has been received.

Vcc \rightarrow +5V Power Supply.

A → accumulator

PAGE NO. 13

DATE :

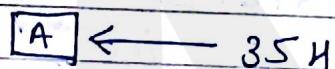
ADDRESSING MODES OF 8085

MP requires data, data is stored at some address, along with instruction, we have to give address of source of data. The method by which address of source of data in a instruction is called as addressing modes of source.

5 Modes in 8085:

1) IMMEDIATE (IAM)

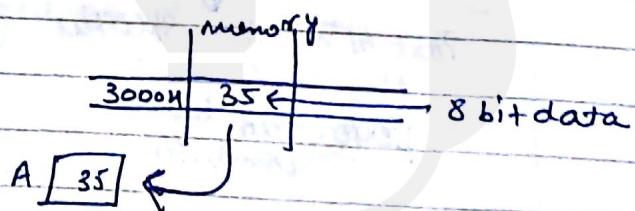
If 8/16 bit data required for executing the inst. is given along with the instruction, then such instructions are Immediate AM instructions. (Generally end with I)
Ex → MVI (move immediately)



2) Direct (DAM)

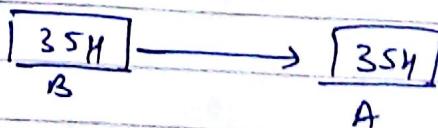
If 8/16 bit data reqd for executing the inst. is present in memory locations and 16 bit address of this memory location is given along with the instruction,

Ex → LDA, STA



3) Register Direct (RDM)

If 8/16 bit data reqd for executing the instruction is present in register/R pair and the name of the R/R pair is given along with the instruction,
Ex MOV,



4) Register Indirect (RIAM)

If 8/16 bit regd for executing the inst. is present in memory location, the 16 bit address of this memory location is present in R pair and the Name of register pair is given along with the inst.

Ex → LDAX, STAX

5) Implicit AM

If the address of source of data as well as address of destination of result is fixed, then there is no need to give any operand alongwith the instruction, such are Implicit AM.

Ex → CMA, STC.

TYPES OF INSTRUCTION

instruction	
opcode	operand

Part of inst that specifies the task to be performed by the computer.

2nd Part, is the data on which the instruction is to be operated.

Ex MOV, MVI

Ex 10H, 30H

types.

* Single * Double * triple.

DATE :

INSTRUCTION SET

8085 → can perform 256 operation at once.

instructions :-

(1) Data transfer : transfers data from source to destination

(R)	MOV Rd, Rs	(Move)	MOV A, B	B → A
(I)	MVI Rd, data	(Move)	MVI A, M	M → A
(I)	LXI Rp, data(16 bit)	(Load)	LXI H	H, L ← data
(RI)	MOV M, Rs	(Move)	MOV M, B	B → M
(R)	MOV Rd, M	(Move)	MOV A, M	M → A
(I)	MVI M, data	(Move)	MVI A, DS	Data → A
(R)	LDAX Rp	(load)	LDAX B	M → A
(Im)	STAX Rp	(Save)	STAX D	A → M
(D)	LDA address(16 bit)	(move)	LDA 5000H	M → A
(D)	STA address(16 bit)	(Save)	STA 5001H	A → M
(D)	LHLD address	(Move)	LHLD 5000H	M → R _P , H
(D)	SHLD address	(Save)	SHLD 5001H	(H, L) → M
(Implicit)	XCHG	(Exchange)	HL ≈ DE Pair	H ↔ D L ↔ L

(2) Arithmetic :

8bit add	ADD R	Register	A ← (A+R)
	ADI data(8 bit)	Immediate	A ← A+data
	ADD M	R Indirect	A ← A+M
8bit add with carry	ADC R	R	A ← A+R+CY
	ACI data(8)	I	A ← A+data+CY
	ADC M	RI	A ← A+M+CY
8bit sub	SUB R	R	A ← A-R
	SBI data(8)	I	A ← A-data
	SBB M	IM	A ← A-M+CY



8 bit sub with borrow	SBB R (R) SBI data (I) SBB M (RI)	$A \leftarrow A - R - Cy$ $A \leftarrow A - \text{data} - Cy$ $A \leftarrow A - M - Cy$
16 bit add decimal adjust	DAD RP (R) DAA (immediate)	$HL \leftarrow HL + RP$ $8\text{bit} \rightarrow BCD \rightarrow ACC$
8 bit increment	INRR (R) INRM (RI)	$R \leftarrow R + 1$ $M \leftarrow M + 1$
16 bit increment	INX RP (R)	$RP \leftarrow RP + 1$
8 bit decrement	DCR R (R) DCRM (RI)	$R \leftarrow R - 1$ $M \leftarrow M - 1$
16 bit decrement	DCXR RP R.	$RP \leftarrow RP - 1$

③ Logical :

AND	ANA R (R) ANI data (I) ANAM (RI)	
OR	ORA R (R) ORI data (I) ORAM (RI)	$CMA \rightarrow \text{complement A}$ $CMC \rightarrow \text{complement C}$
X-OR	XRA R (R) XRI data (I) XRAM (RI)	
compare	CPI (R) (R) CPI data (I) CMP M (RI)	(A) - (R) and affect flags (A) - data " " " (A) - M " " "
complement bit manipulation	CMA implicit STC implicit CMC implicit	$A \leftarrow A^1$ $CF \leftarrow 1$ (Set carry to 1) $CF \leftarrow CP^1$ carry flag comple



(4) Stack Related:

Load	LXI SP, 16 bit data	(I)	data → SP data	
increment	INX SP	(RD)	$\boxed{X} + 1 \rightarrow \boxed{X+1}$	
decrement	DCX SP	(RD)	$\boxed{X} - 1 \rightarrow \boxed{X-1}$	
Double add	DAD SP.	(RD)	$\boxed{SP} \rightarrow \boxed{HL}$	$\boxed{HL} + \boxed{SP} \rightarrow \boxed{HL}$ $SP \downarrow$

Transfer SP HL

PUSH RP (Implicit)

POP RP (RI)

$\boxed{HL} \rightarrow \boxed{SP}$

Exchange SP to HL

$\boxed{SP} \rightarrow \boxed{HL}$

(5) Branching is used to transfer 16 bit add to PC

branch
PC to PC
uncond
Jump

PC HL

HL $\boxed{X} \rightarrow \boxed{X} PC$

JMP add (16) unconditional $\boxed{add} \rightarrow \boxed{add} PC$

JC	Jump if carry	CF=1	conditional jump
JNC	" if No carry	CF=0	
JZ	" if Zero	Z=1	
JNZ	" if No zero	Z=0	
JPE	" if Even Parity	P=1	
JPO	" if Odd Parity	P=0	
JM	" if Minus	S=1	

JP " if Positive S=0



PAGE NO 18
DATE :

(6)

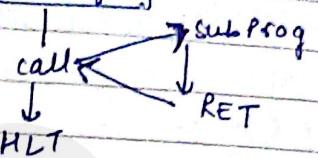
Subroutine

uncond

call address (16 bit)

Subprograms / Service - Routine

main Program

unconditional
return

RET

* (Return)

call

CC

CNC

CZ

CNZ

CPE

CPO

CM

CP

conditional

if carry

if no carry

if zero

if NO zero

if Even parity

if odd parity

if negative

if positive

return

RC

RNC

RZ

RNZ

RPE

RPO

RM

RP

(7)

Machine Control.

D1	disable interrupt	[]	NO flag affected
E1	enable interrupt	[]	

(8)

Other

NOP

No operation

HLT

Halt

IN

I/P data to acc from a Port

OUT

O/P " " " " "

ANAL

PORTB

PORTB

— —

PAGE NO. 19

DATE :

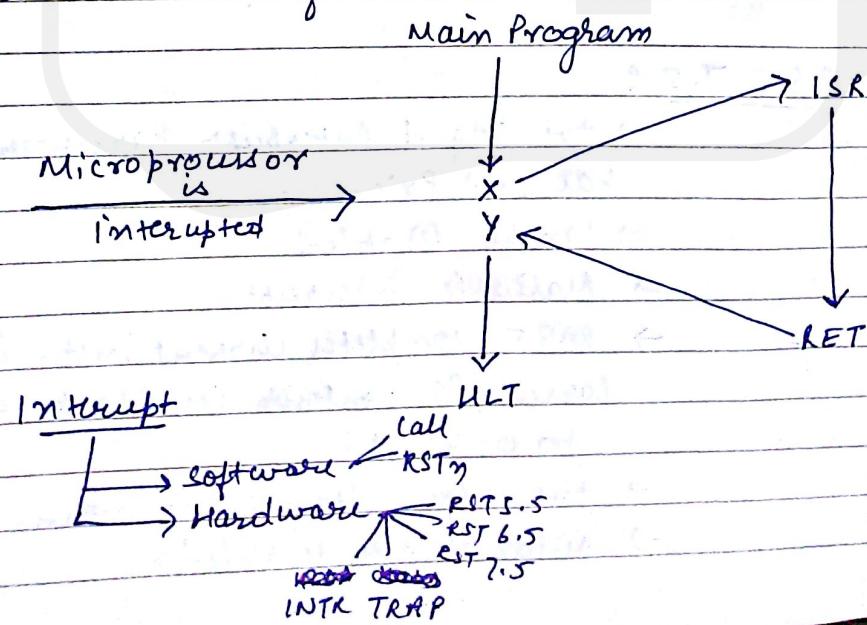
RAL	Rotate A left through carry
RAR	" A Right " "
RLC	" A left "
RRC	" A Right
RIM	Read interrupt mask
RSTn	Restart from address (n*8)H
SUI	Subtract immediate from A
XCHG	Exchange HL with DE

~~RETI~~~~DI~~

INTERRUPTS OF 8085

When Microprocessor executes any program, then it is called Main Program. In b/w the execution of Main Program, if microprocessor is interrupted, then it will branch from main program 'X' to sub program called Interrupt Service Routine (ISR).

MP executes ISR, when RET comes in ISR, MP returns back from ISR to Main Program inst. 'Y' from where it was 'left'.



Software Interrupt :

- ① Call add(16bit)
- ② RST_n Restart from address (8x n)H

Hardware Interrupt :

① TRAP : (RST 4.5) (\nearrow)

- +ve edge, level triggered hardware interrupt
- If signal on TRAP pin has a leading edge and a sustained high level, the MP completes the current instruction, pushes the Program Counter in the stack, and branches to location 0024H.
- In order to accept another interrupt on the trap line, the prev. Trap interrupt can be disabled by the falling edge. This avoids multiple interrupts from same device.
- Trap is used for power failures and emergency shut off.
- highest priority, can't be disabled
- Non-Maskable interrupt (NMI)

② RST 7.5 :

- +ve edged interrupt hardware interrupt has 2nd Priority.
- can be disabled.
- Maskable interrupt.
- 8085 completes current inst, Push current PC contents in stack & branches to 003CH.
- +ve edge also set an internal A F/F
- automatically disabled.



enable/disable through SIM1 net for RST 6.5, 7.5, 5.5

PAGE NO. 21

DATE :

(2) RST 6.5 :

- level triggered ($\overline{I} \uparrow L$) hardware interrupt
- can be disabled so maskable
- after completion of current inst., Push PC contents in stack and branches to 0034H
- automatically disabled.

(4) RST 5.5 :

- level triggered ($\overline{I} \uparrow L$) H.I.
- disabled, maskable
- after completion, Push PC contents in stack and branches to 002CH.
- enable/disable through SIM1 net.

(5) INTR →

- level triggered ($\overline{I} \uparrow L$), least Priority
- can be disabled, Maskable
- after completion, push the PC into stack, generates an interrupt acknowledge (\overline{INTA}) low pulse on the control bus.
- This instruction must be provided with external hardware..



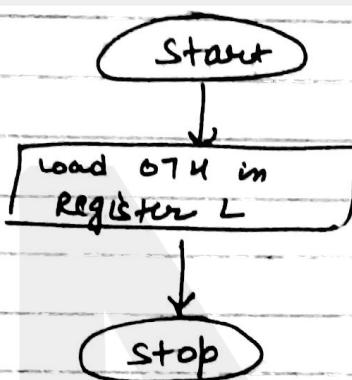
PAGE NO. 22

DATE :

PROGRAMMING IN MICROPROCESSOR.

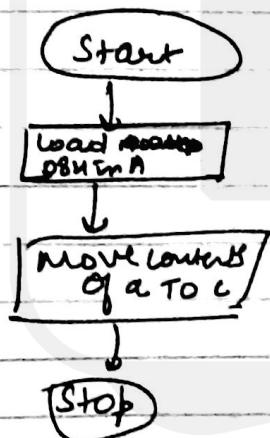
Loading Register , Memory with data:

① transfer 07H in register L



M-Ad	Mnemonics	operands
2000H	MVI	L, 07
2002H	HLT	-

② load A Register with 08H ,then move to C



M-Ad	Mnemonics	operands
2000H	MVI	A, 08
2001H	MOV	C, A
2002H	HLT	-

