Timing and Control Unit in 8085

⇒ Serial IO segment.

- Mp 8085 is able for serial communication and parallel communication.
- In general, Mp perform parallel communication.
- Mp can perform serial communication by using serial IO segment.
- Parallel communication is faster than serial communication.

There are two pins available for serial communication.

i) SID (Serial In Data)
⇒ to receive serial data.

ii) SOD (serial out data)
⇒ Mp use this pin to transfer serial data.

11010110 —serially→ 11010110
totally 8 clock pulses

11010110
|||||||| data lines ⇒ 1 clock pulse
11010110

Instruction decoder segment

i) Instruction Register (IR)
ii) Instruction Decoder ⇒ fetching

IR ⇒ 8 bit special purpose register which is used to store the OPCODE which is next to be going to fetch.

⇒ It is the only register which cannot be accessed by user.

⇒ There is no any instruction available for IR.

Instruction Decoder ⇒ used to decode the opcodes so that Mp can perform the given operation.

Timing Segment → X_1, X_2, clock out

Power Supply → +5V, GND

Control Signals → READY, $\overline{RD}$, $\overline{WR}$, ALE

Status Signal → S_0, S_1, IO/$\overline{M}$

DMA Signal → HOLD, HLDA

Reset Signal → RESET IN, RESET OUT

Timing Segment

→ X_1, X_2, CLK OUT

Control segment

i) control signals
→ READY, $\overline{RD}$, $\overline{WR}$, ALE

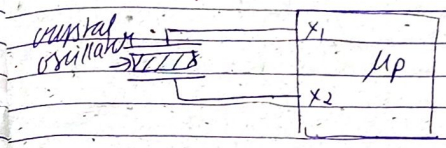
ii) Status signal
→ S_0, S_1, IO/$\overline{M}$

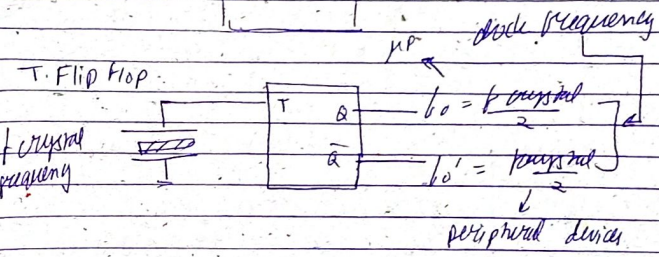
iii) DMA signal
→ HOLD, HLDA

iv) RESET signal.
→ RESET IN, RESET OUT.

Timing Segment

- This segment is basically used to provide clock signal to µp and peripheral devices.





T FF (Toggle)

I/p	O/p
0	Q_n (previous)
1	$\bar{Q}_n$ (toggle)

⇒ Crystal frequency ($f_{crystal}$) is divided in two equal frequencies by using T.F.F.

$f_1 = f_2 = \frac{f_{crystal}}{2}$

⇒ these frequencies are provided to µp & peripheral devices for maintaining the synchronization b/w them.

Control segment

1.) Control Signals
i.) ALE 2.) $\overline{RD}$ 3.) $\overline{WR}$ 4.) READY

- ALE (Address latch Enable)
⇒ this signal is used to enable the address latch.
⇒ this signal is used for demultiplexing of multiplexing buses.

- Read signal ($\overline{RD}$)

$\overline{RD}$ → 0 → µp will perform read operation
→ 1 → µp does not perform read operation.

- Write signal ($\overline{WR}$)

→ 0 → will perform write operation
→ 1 → does not perform write operation

- READY
⇒ used by slower peripheral devices.
⇒ this signal impose wait state to µp.

READY → 0 → µp is in wait state
→ 1 → slower peripheral devices are ready to transmit is data

2.) Status signal
1. IO/$\overline{M}$, 2. S_0, 3. S_1.

IO/$\overline{M}$ → 0 → µp will operate with memory
→ 1 → µp will operate with I/O devices.

S_0	S_1	Operation
0	1	read
1	0	write
1	1	fetch

IO/M	S_0	S_1	RD	WR	Operation
0	0	1	0	1	Memory read
0	1	0	1	0	Memory write
0	1	1	0	1	Memory fetch
1	0	1	0	1	I/O read
1	1	0	1	0	I/O write
1	1	1	1	1	Interrupt acknowledge

3) RESET Signals.

⇒ this signal is used to reset (making 0) the µp or peripheral device.

i) RESET IN ⇒ used for µp for resetting µp.

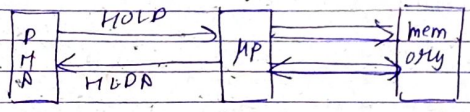
⇒ it reset the content of accumulator, SP, PC, flag register, general purpose registers.

2) RESET OUT ⇒ used to reset the peripheral devices

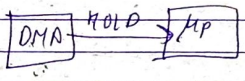
4) DMA signals

⇒ direct memory access

the device which has higher speed than µp is called DMA device.



HOLD ⇒ when DMA device want to communicate with µp, it sent HOLD signal to µp.



when µp recive HOLD signal, it performs following task

i) µp suspend all its operations and save the intermediate result in stack.

ii) µp release its command from all bus systems.

iii) it sends HLDA signal to DMA device.

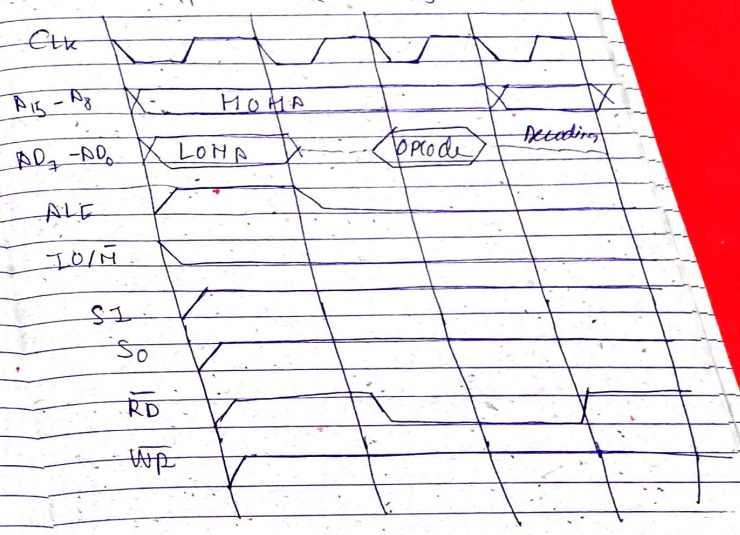
HLDA (hold acknowledgement)
→ sent by up to DMA device.

When DMA device received HLDA signal, it transmit its data directly in the memory. Here this data transfer operation is also known as "Input to memory data transfer operation".

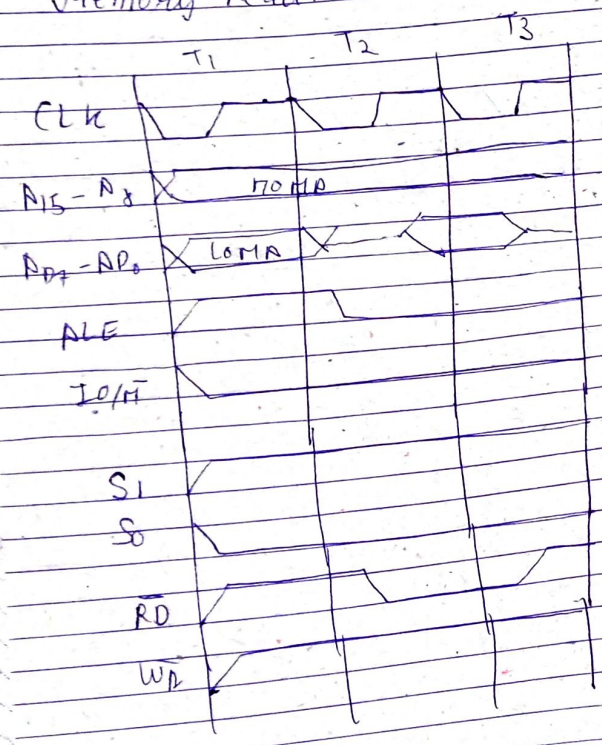
Timing Diagram of 8085

1) opcode fetch
2) Memory read
3) Memory write (4T states)
4) I/O read (3T)
5) I/O write

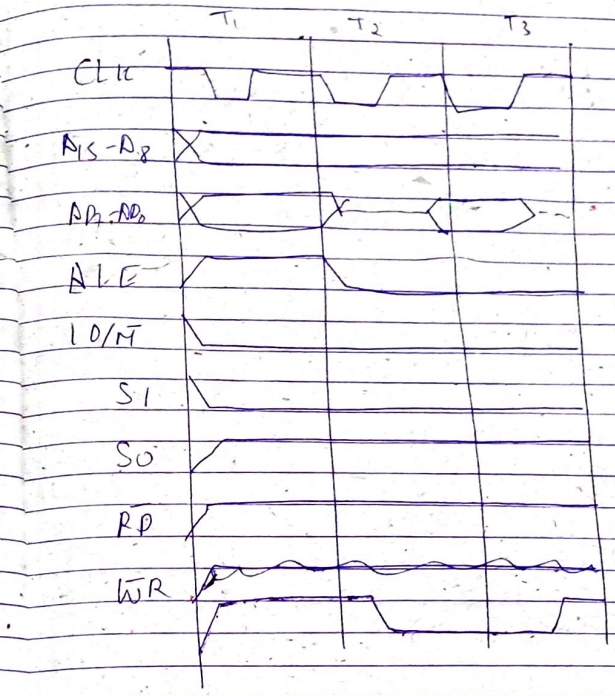
Opcode fetch



Memory Read



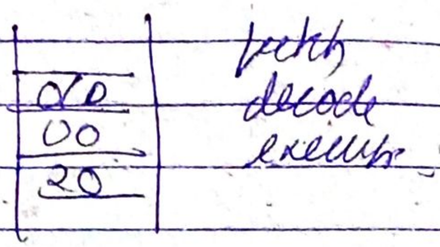
Memory Write



Timing Diagram of Machine Cycles

A ← [2000H]

i) LDA 2000h



1. Opcode fetch (4T)
2. Memory read (3T)
3. Memory Read (3T)
4. Memory Read (3T)

ii) STA 2000H;

1. Opcode fetch
2. Mem read
3. Mem read
4. Mem write

iii) MVI B, 25H

1. O/P fetch (4T)
2. Mem Read ← 3(T)

(iv) LHLD 5000h

i) O/P fetch
(i) Mem Read
ii) Mem. Read
(iv) Mem. Read
(v) Mem. Read