

Microprocessor 8085

8085 Block diagram includes two more important things

- ① ALU (Arithmetic & logical unit): It performs various arithmetic & logical operations. That are addition, subtraction, Increment & decrement.
- ② TCU (Timing & control unit):
 - 1) Act as brain
 - 2) It generates TCS (Timing & control signals) that are required by the processor, memory and I/O devices. for the operation of
 - 3) It controls data flow b/w processor, memory, other I/O peripheral devices.
 - 4) TCU is connected with crystal oscillator responsible generating 6 MHz frequency on which 8085 operates. TCU
oscillator
- ③ Registers: Registers are used for temporary storage for and manipulation of data & instructions.
8085 has following set:
 - ① Special purpose Register.
 - a) Accumulator: (8 bit)
 - 8 bit Register associated with ALU.
 - It holds one of the arithmetic operand for AL operation.
 - Other operand is stored inside in the memory or in the general purpose Register.
 - The final result AC operation is placed inside accumulator.
 - b) Program Counter: (16 bit) address
 - It is to store the address not data.
 - It holds the address of next instruction to be executed.
 - This register takes care the sequencing of the program flow & control.

c) Stack pointer: (16 bit).

- SP is 16 bit register that used point the memory.
- The memory this register points to a special area called stack.
- Stack stores the content of A, PC, flags, GPR register during the execution of program.
- Stack is area of memory used to hold data that will be retrieved soon.
- Any portion of the memory can be used as stack.
- The stack is usually accessed in LIFO (Last In First Out) order.
- Content of accumulator is stored when subroutine is used.

d) Flag Register. (8 bit) only 5 bits are used by microprocessor and rest 3 bits are unused.

- Flag Register directly associated with ALU.
- [After execution of each instructions flag registers are updated] For after execution of each every single instruction flag register's status are updated.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S	Z	X	AC	X	P	X	CY

① CY flag: 1 if carry is generated after execution of arithmetic instruction.

0 if carry is not generated after execution of arithmetic instruction.

MVI A, CCH

MVI B, E6H

$\begin{matrix} & \text{cc4} \\ \oplus & \text{E6H} \end{matrix}$

$$\begin{array}{r} 1100 \\ 1110 \\ \hline 1011 \end{array} \quad \begin{array}{r} 1100 \\ 0110 \\ \hline 0010 \end{array}$$

ADD B.

$$\boxed{CY = 1}$$

② P flag: (Parity) flag $\boxed{P=1}$

1 if No. of binary one's in accumulator are even.

0 if No. of binary one's in accumulator are odd.

③ AC flag: (Auxiliary carry)

- If carry is generated from D₃ to D₄, AC=1
- 0 if carry is not generated from D₃ to D₄.

Eg MVI A, CCH

MVI B, E6H

ADD B.

$$\begin{array}{r} \textcircled{1} & 1100 \\ & 1110 \\ \hline & 0010 \end{array}$$

carry generated from D₃ to D₄.

- AC=1 when nibble to nibble carry is generated.

④ Z flag: (Zero flag)

- Z=1 if result is zero after execution of instruction.
- Z=0 if result is non-zero after execution of instruction.

⑤ SF flag: (Signed flag)

- SF=1 if D₇=1 (If result is -ve after execution AL instruction)
- SF=0 if D₇=0 (If result is +ve after execution AL instruction)
- Sign flag 8085 is judged according D₇ bit Accumulator.

⑥ Temporary Registers (8bit).

- These temporary registers are not accessible by the programmers.
- It is only used by microprocessor in some instructions.
- It holds data during AL operations.
- W & Z are example of 8bit temporary registers.

⑦ General purpose Registers. (8bit).

- B, C, D, E, H, L

- used by programmers.

- can be used in combination of two register pairs.

Control Signals in 8085

- ① \overline{RD} → It is signal sent by microprocessor to I/O device/memory to control the read operation. When \overline{RD} bus goes down/selected memory & I/O devices are read.
- ② \overline{WR} → It is a signal sent from processor to I/O device/memory to control write operation. When it goes low that data is written to selected memory or sent to I/O devices.
- ③ $I/O/\overline{M} = 1$ is used to distinguish whether the address is of memory or I/O devices.

When, $I/O/\overline{M} = 1$, I/O operation's should be performed.
when, $I/O/\overline{M} = 0$, memory operation's should be performed.

By using above 3 signals we can generate 4 control signals. i.e., M1 Read, M1 Write, I/O read, I/O write.

			control signals.
$I/O/\overline{M}$	\overline{RD}	\overline{WR}	
0	0	1	memory-read \overline{MEMR}
0	1	0	memory-write \overline{MEMW}
1	0	1	I/O-read \overline{IOR}
1	1	0	I/O-write \overline{IOW}

Later. These control signals are used for interfacing with 8085 microprocessor with help of decoder. (3×8).

Status Signals in 8085

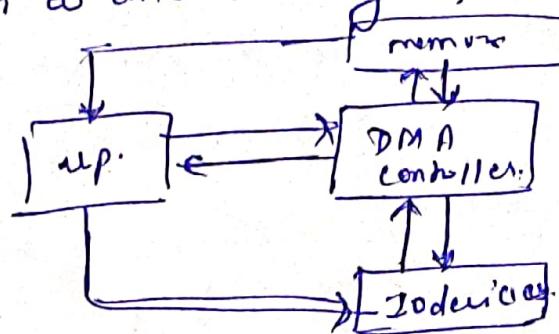
- ① S0 and S1 (Status lines): use distinguish b/w H/W R/F. operation.

S0	S1	Operation.
0	0	HALT
0	1	Write
1	0	Read
1	1	Fetch

* CPU and Bus Control Lines.

① **HOLD:** When another device of computer system requires address or data bus lines of data transfer, it sends HOLD signal to supervisor ($\text{IO} \rightarrow \text{supervisor}$)

② **NLDA:** (Acknowledgment) Signal sent from (supervisor $\rightarrow \text{IO}$) indicating signal has been received by supervisor and currently upon being held to and waiting of receiving data from IO device.

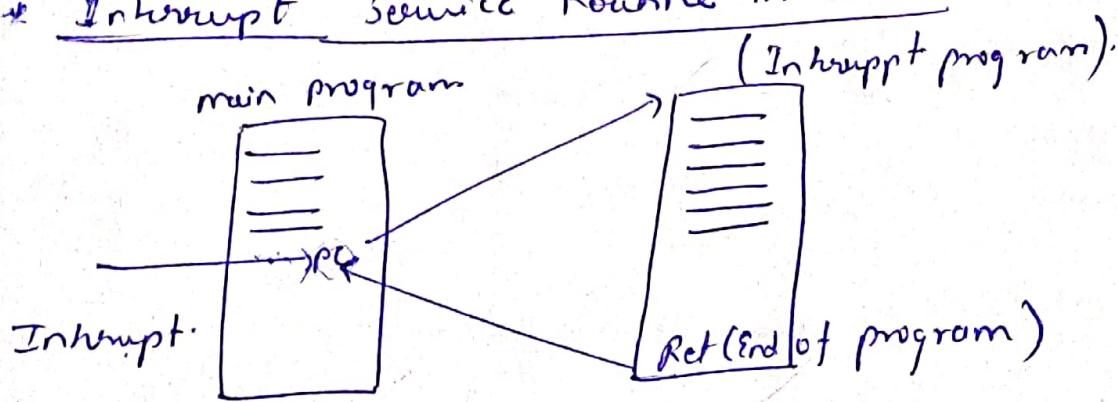


③ **DMA:** Control Not external devices that Control Data flow b/w memory & I/O devices.

④ **Reset In:** Reset the PC, Interrupt enable, NLDA Flip Flop & IR.

⑤ **Reset Out:** indicates that CPU is being reset.

* Interrupt Service Routine in 8085.



→ Interrupt are the external or internal signals or hardware or software signals that distract or alter the current execution of the program.

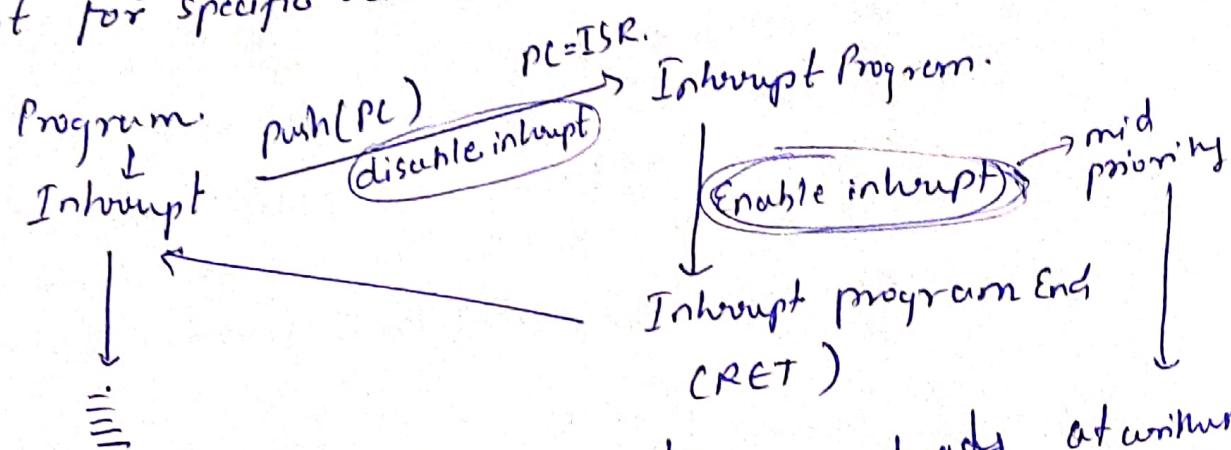
→ Interrupt is the way through which I/O devices communicate with the processor.

→ I/O devices send's the signals to processor that may require's the services of ^{the} processor.

It is method through which IO devices inform the processor that they required the services of processor.

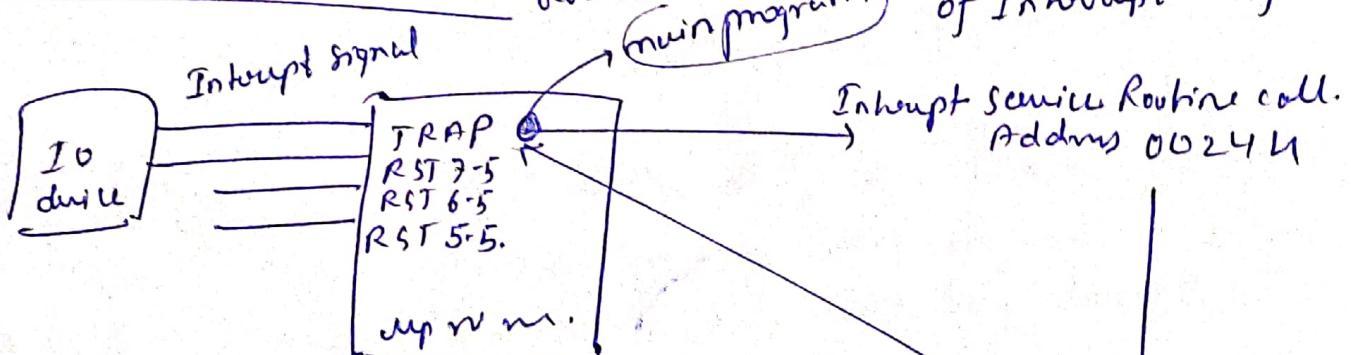
Flow of interrupt

- ① Micro. checks the interrupt after / during every instructions.
 - ② When interrupt occurs from IO devices / memory. processor firstly completely execute current instruction.
 - ③ After that processor will push program counter on the stack.
 - ④ Reset INTE flip flop so that no more interrupt can recognize.
 - ⑤ That flow is transferred to ISR (Interrupt Service Routine).
- ISR: is nothing but contains step by step instructions to resolve the interrupt for specific address.



Hardware Interrupt: (ISR address given by processor only well available) at written address already in mid.

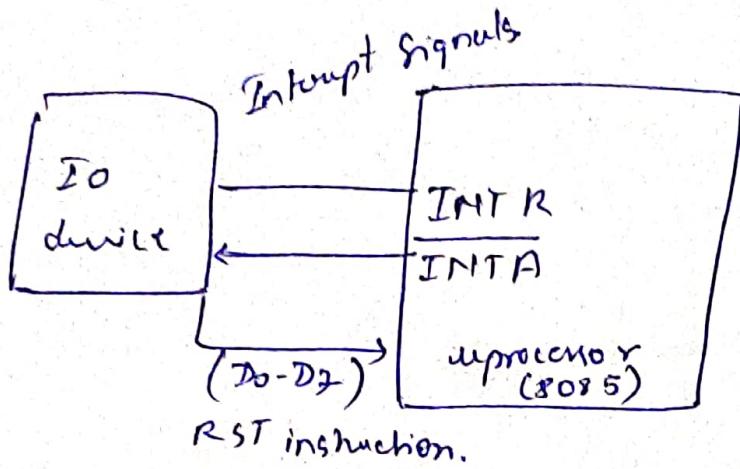
High priority written end of Interrupt Program.



In execution hardware interrupt each hardware pin have well defined memory location.

be served. because address of ISR ~~is~~ received from interrupting device

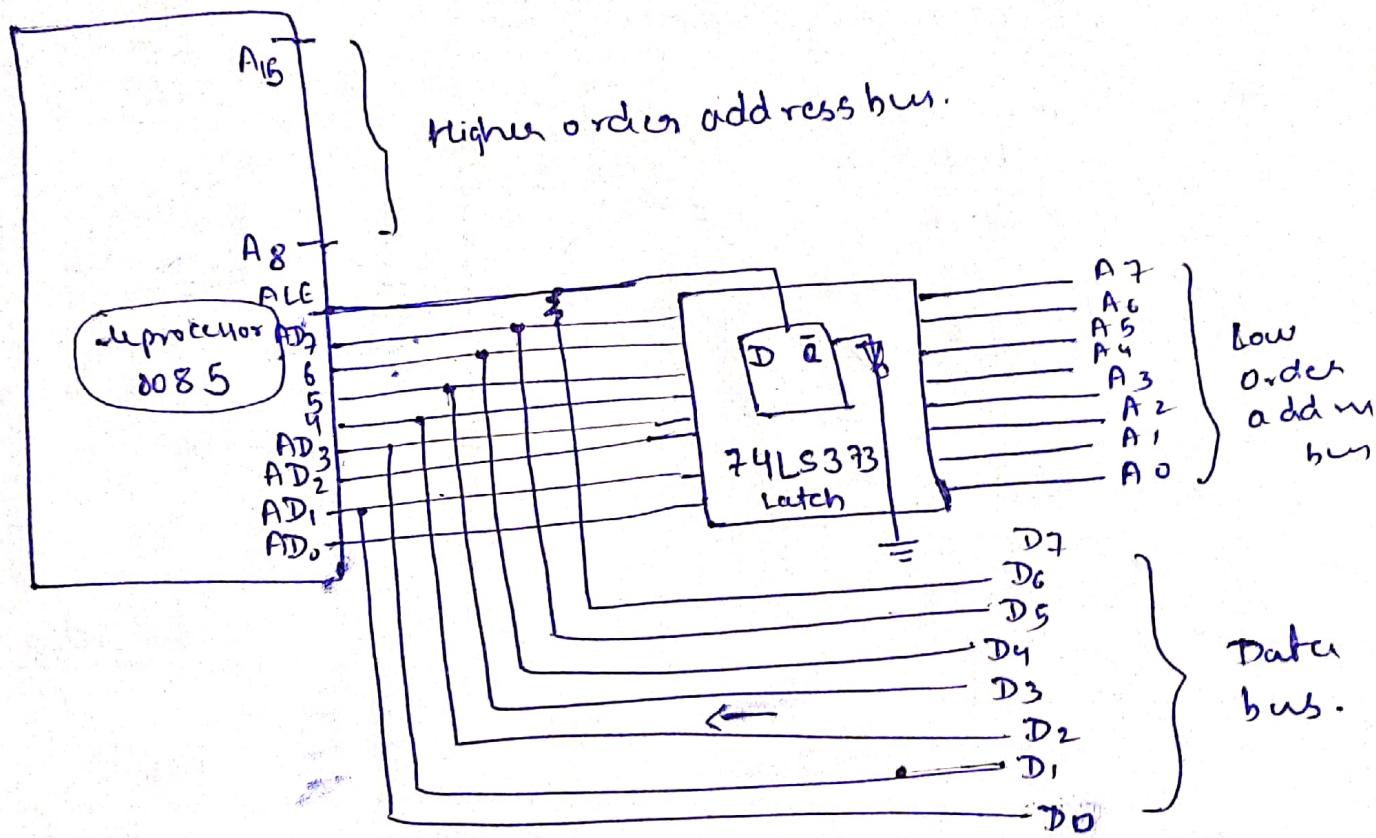
* Software Interrupt: (ISR address given by IO only in form of instruction).



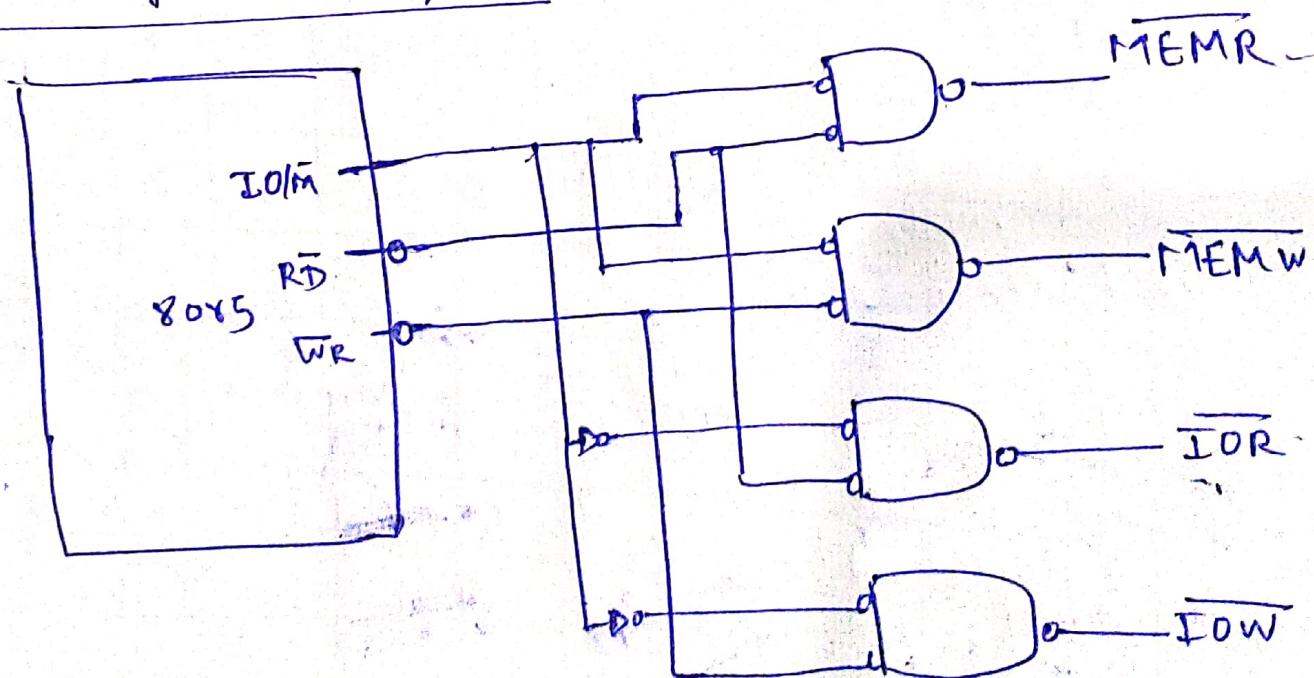
In software Interrupt:

- ① When Microprocessor receives software interrupt from IO devices.
- ② microprocessor $\xrightarrow{\text{send}} \overline{\text{INTA}}$ signal to IO devices. to indicates that interrupt has been received.
- ③ Then IO device again gives instruction to microprocessor. Then whatever instruction is given by IO device microprocessor will transfer control ISR (Interrupt Service Routine).
- ④ Maskable interrupt: These type of interrupt can be ignored.
- ⑤ Non maskable interrupt: This type of interrupt can't be avoided or ignored.
- ⑥ Vectorized interrupt: vectorized I_O are those type of interrupt can contain's specified well defined address for ISR (Interrupt Service Routine) and they doesn't want to depend on peripheral devices. It contain's unique code typically of size 4bits to 8bits. When interrupt is generated this unique code will transferred to processor using data bus. Then processor will identify which ^{interrupt} service routine is to be executed.
- ⑦ Non-vectorized: M-V doesn't contain's well defined unique code. In M-V interrupt peripheral devices directly send's ISR instruction to processor. This required more time interrupt to be served because address of ISR ~~interrupt~~ received from interrupting

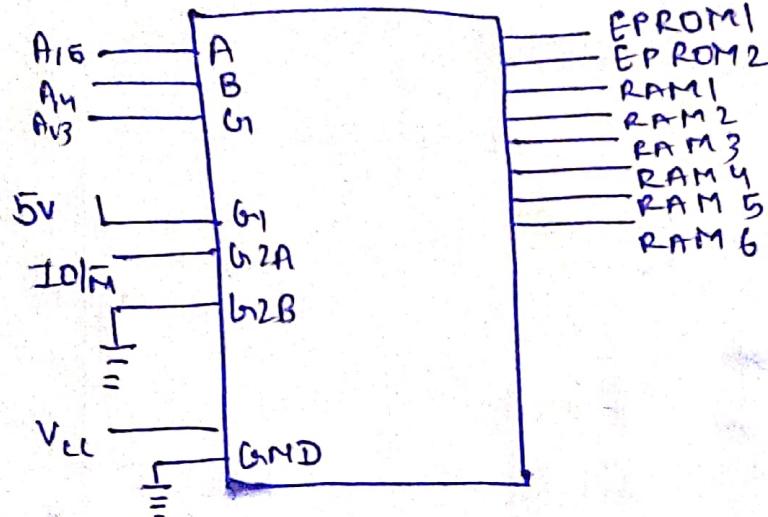
* Schematics for demultiplexing low order address bus & data bus.



* Schematics for Control signals (\overline{IOM} , \overline{RD} , \overline{WR})



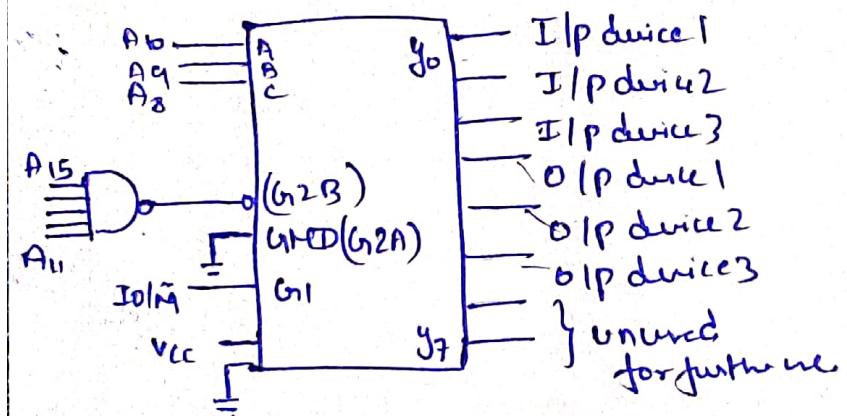
* Memory Interacing Using 743L138 Decoder.



Truth table

Decoder output	Memory Device	Zones	Memory address range
Y ₀	EPROM1	Zone 0	0000 - FFFF
Y ₁	EPROM2	Zone 1	2000 - 3FFF
Y ₂	RAM1	Zone 2	4000 - 5FFF
Y ₃	RAM2	Zone 3	
Y ₄	RAM3	Zone 4	
Y ₅	RAM4	Zone 5	
Y ₆	RAM5	Zone 6	
Y ₇	RAM6	Zone 7	E000 - FFFF

* Interface of I/O device Using 74LS138



A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	O/p lines selected	Corresponding address	I/O connected device
1	1	1	1	1	0	0	0	Y ₀	F8	I/O device 1
1	1	1	1	1	0	0	1	Y ₁	F9	I/O device 2
1	1	1	1	1	0	1	0	Y ₂	FA	I/O device 3
1	1	1	1	1	0	1	1	Y ₃	FB	O/P device 1
1	1	1	1	1	:	:	:	:	FC	O/P 2
1	1	1	1	1	:	:	:	:	FD	O/P 3
1	1	1	1	1	:	:	:	:	FE	Unused
1	1	1	1	1	1	1	1	Y ₇	FF	Unused

* Data Transfer Group 8085

- ① MOV } → extension
- ② MVI } → z extension.
- ③ LXJ }
- ④ LDA
- ⑤ STA
- ⑥ LHLD
- ⑦ SHLD
- ⑧ LDAX
- ⑨ STAX
- ⑩ XCHL

* Arithmetic Instruction Group

- ① ADD r
- ② ADD M
- ③ ADC r
- ④ ADC M
- ⑤ ADI data $\rightarrow A \leftarrow A + \text{data}$.
- ⑥ ACI data $\rightarrow A \leftarrow A + \text{data} + C$
- ⑦ DAD rp $\rightarrow [W] - [H] + [rP]$
- ⑧ SUB r
- ⑨ SUB M
- ⑩ SBB r
- ⑪ SBB M
- ⑫ SBI data $A \leftarrow A - \text{data}$
- ⑬ SBI data. $A \leftarrow A - \text{data} - C$
- ⑭ INC r

⑯ INC M

⑰ DCR r

⑱ DCR M

⑲ JNX rp $\rightarrow rP - [rP] +$

⑳ DCX rp $\rightarrow rP - [rP] +$

㉑ DAA \rightarrow mode

Correct hen \rightarrow
demirel

* Arithmetic Group of 8085

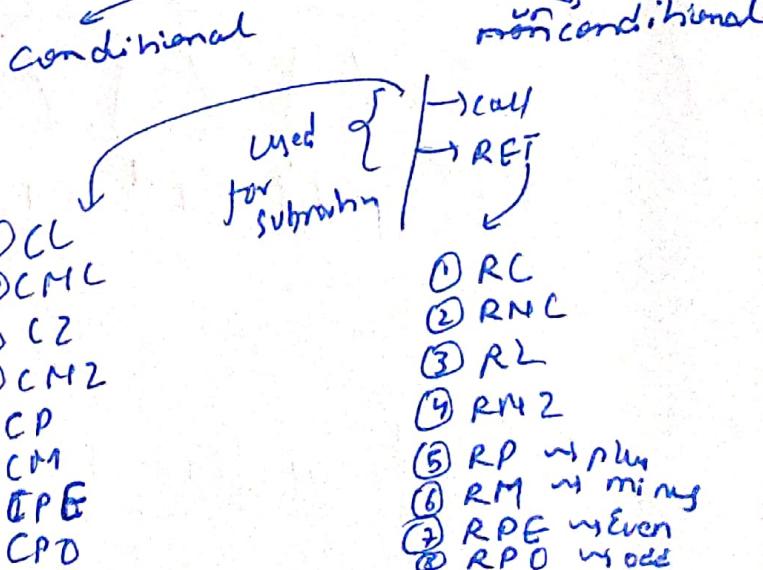
- 1) ANA } → two combinations.
 - 2) ANI
 - 3) ORA }
 - 4) ORI
 - 5) XRA }
 - 6) XRI
 - 7) CMA → complement of accumulator
 - 8) CMC → complement of carry.
 - 9) CMP R → compare for A with r ($r-A$) result set according to result.
 - 10) CMP M
 - 11) RLC → left (without) carry.
 - 12) RRC
 - 13) RAL → Left with carry shift along with carry.
 - 14)RAR
- $O = [A] - [r]$ carry played
 $+1 = [A] > [r]$ carry play = 0
 $-1 = [A] < [r]$ carry play = 1
- This includes carry flag

* Branch Instruction Group of 8085

- Jump statement
 - call & return statement
 - Restart { RST0 } ^{0000N} directly use RST 0-7 for RST (Restart instruction)
 - RST7 } ^{0008N}
 - RST8 } ^{0008N}
- (A) JUMP Statement : used to jump to specific memory location directly that is they are also known as unconditional JUMP.

Branch statements

- ① JC
- ② JNC
- ③ JZ
- ④ JMZ
- ⑤ JP → result plus
- ⑥ JMO → result minus
- ⑦ JPE → even parity
- ⑧ JPO → odd parity.



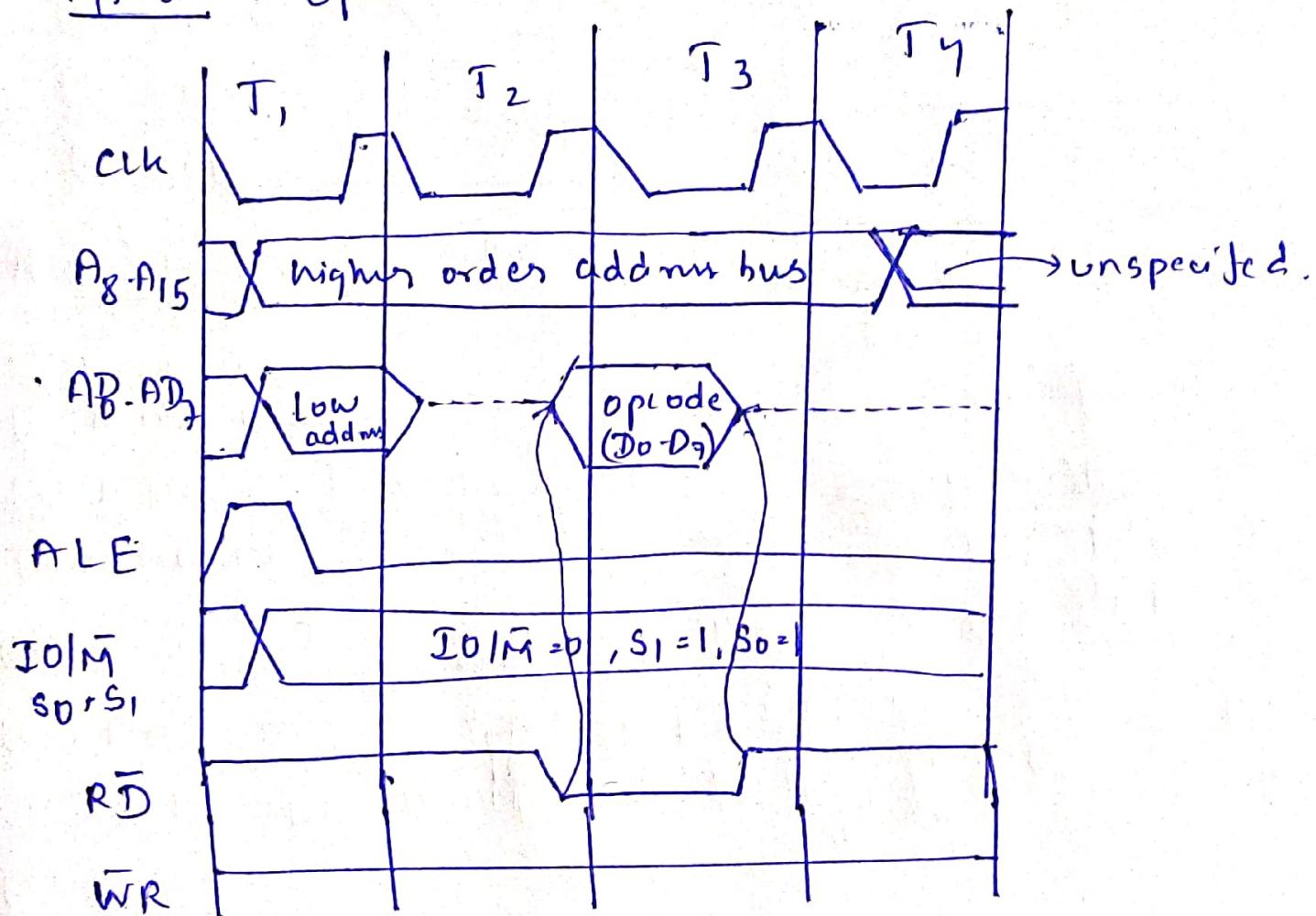
* Timing Diagrams for opcode fetch cycle.

T₁ State → supervisor sends address location where op code is available.

T₂ State → control unit will generate RD signal to enable read operation from memory. The content of address/op code is placed on the data bus.

T₃ State → The op code is placed on IR.

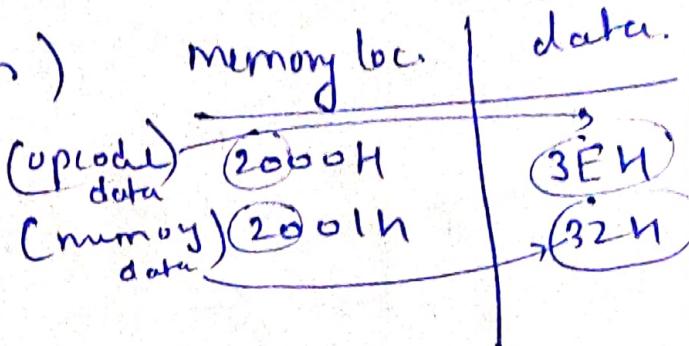
T₄ State → op code is decoded.



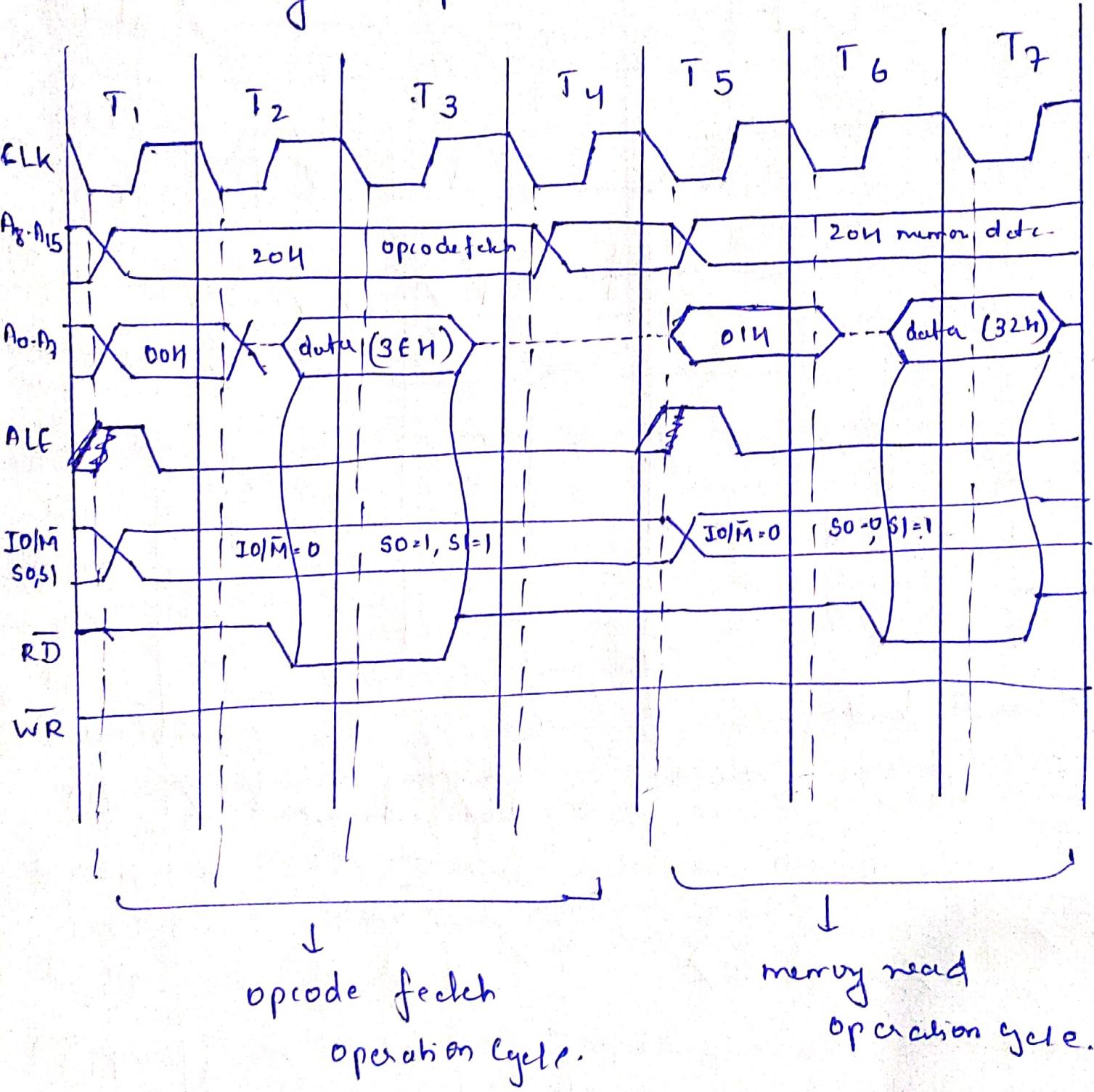
* Timing Diagram of MVI instruction.

MVI A, 32H (2 byte instruction)

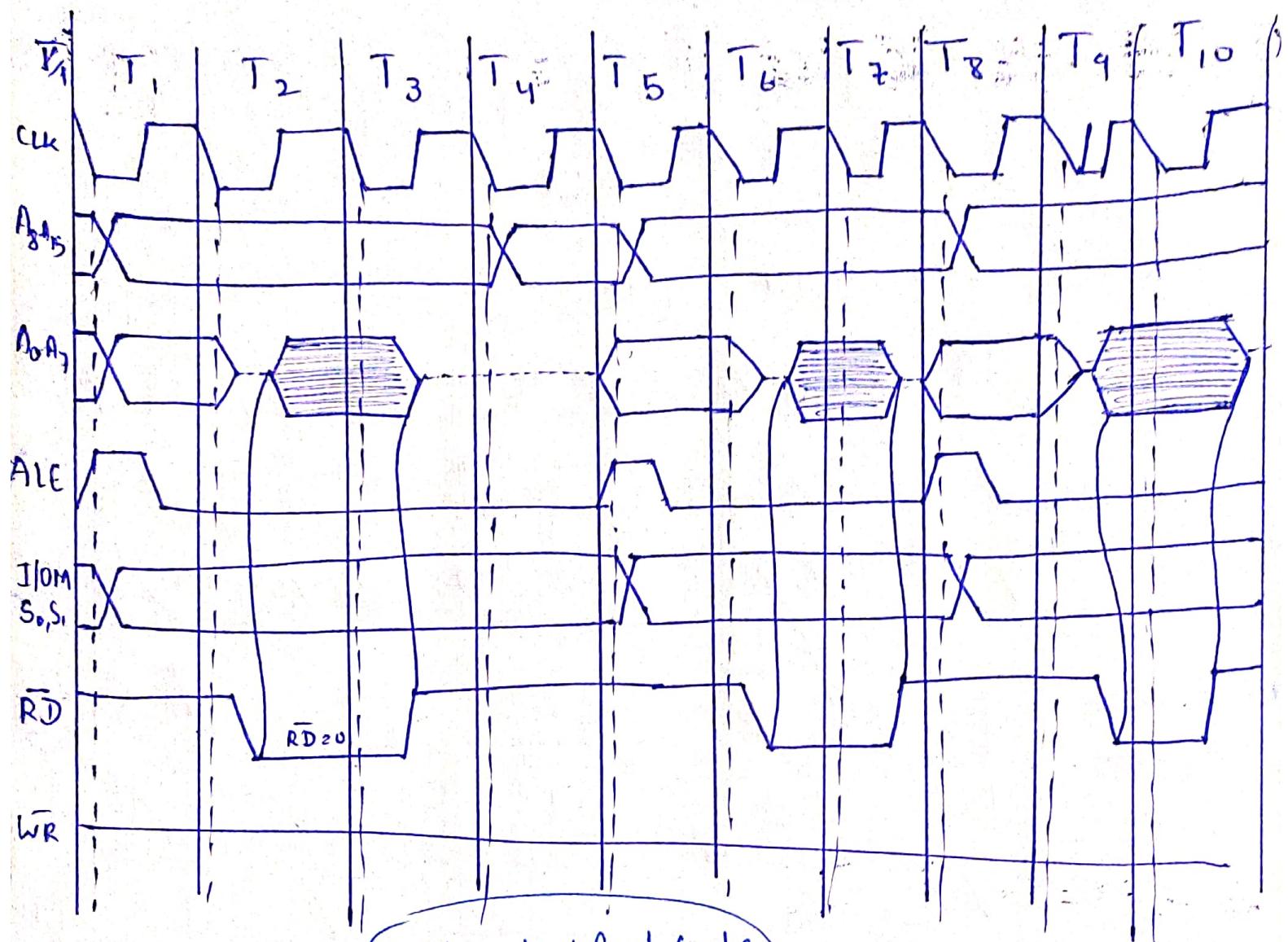
Name: Hitendra Bisodha
SAPid: 500091910



- ① opcode fetch operation.
- ② Memory read operation



* Timing Diagram for opicode fetch + Memory Read + Memo I/O Read.



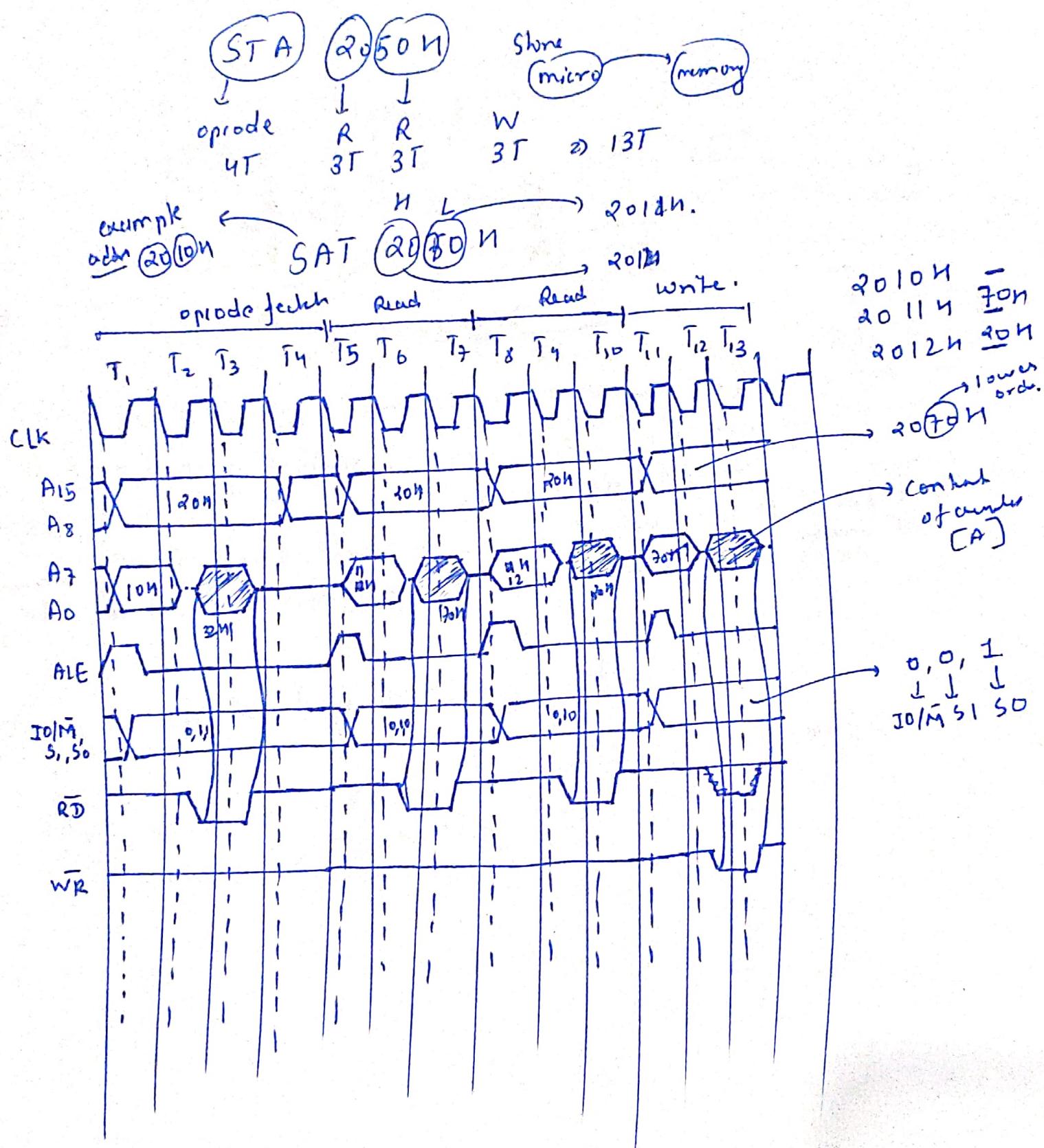
contains Fetch cycle + Read cycle
execute.

* **Instruction Cycle:** Instruction Cycle is the time taken by microprocessor to complete the execution of one instruction. For one instruction cycle there can be multiple machine cycles.

* **Machine Cycle:** machine cycle can consist of multiple T-cycles Time required to complete execution of single operation

* **T-state:** T-state is the time corresponding to one clock period. Unit which is used to calculate the time taken by one instruction.

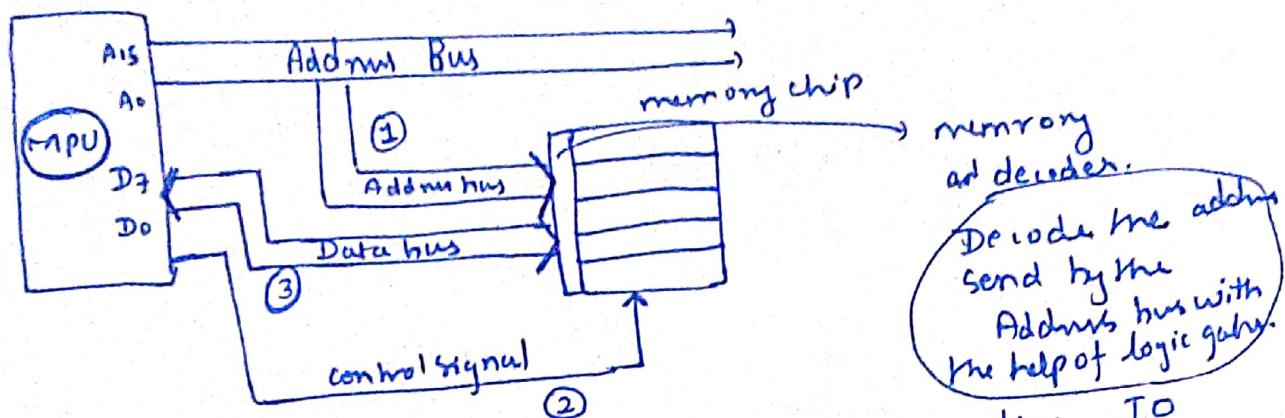
* Timing Diagram for [STA 9030H] of 3 byte instruction!



* Address instruction flow inside 8085 (Data Flow)

(Read Operation)

Steps: Microprocessor places 16 bit address to address bus.



Step 2: MPU sends control signal to memory chip to activate IO and/or memory operation. (memory I/O chip is activated with the help of control signal.)

Step 3: Once the memory chip is activated with the help of control signal, present at address that is decoded by logic gates are placed on address bus data.

{ For Write Operation some instruction are followed. }

Step 3: Microprocessor will send the to memory using the Data bus.