Timing Diagram 8085 Microprocessor

A time diagram is a graphical representation where the 8085 instruction timing diagram represents the execution time of each instruction in graphical format. In this article, we will be Going Through the Timing Diagram, We will go through Different Machine Cycles Such as Opcode Fetch, Memory Read, Memory Write, I/O Read, and I/O Write, Then we will go through the Timing Diagram of Some Instructions such as MVI, STA, and INR, At last.

What is a Timing Diagram?

A time diagram is a graphical representation. The [8085](#) instruction timing diagram represents the execution time of each instruction in graphical format. Execution time is given in T-states. The 8085 microprocessor has a set of control signals and data signals that play an important role in the execution of instructions. In this article, I will explain in detail what a timing diagram is and how to draw a timing diagram of different instructions

- **Clock Signal:** The time required to execute an instruction is called a clock cycle.

- **Machine Cycle:** The time required to access memory or input/output devices is called a machine cycle. The 8085 has 5 basic machine cycles i.e., load opcode, read from memory, write to memory, read I/O, and write I/O.

- **T-State**: A machine cycle and an instruction cycle take several clock periods. The portion of an operation performed in one system clock period is called a T-state.

- **Control Signals:** The control signal controls the operations. Common signals are [ALE](#) (address block

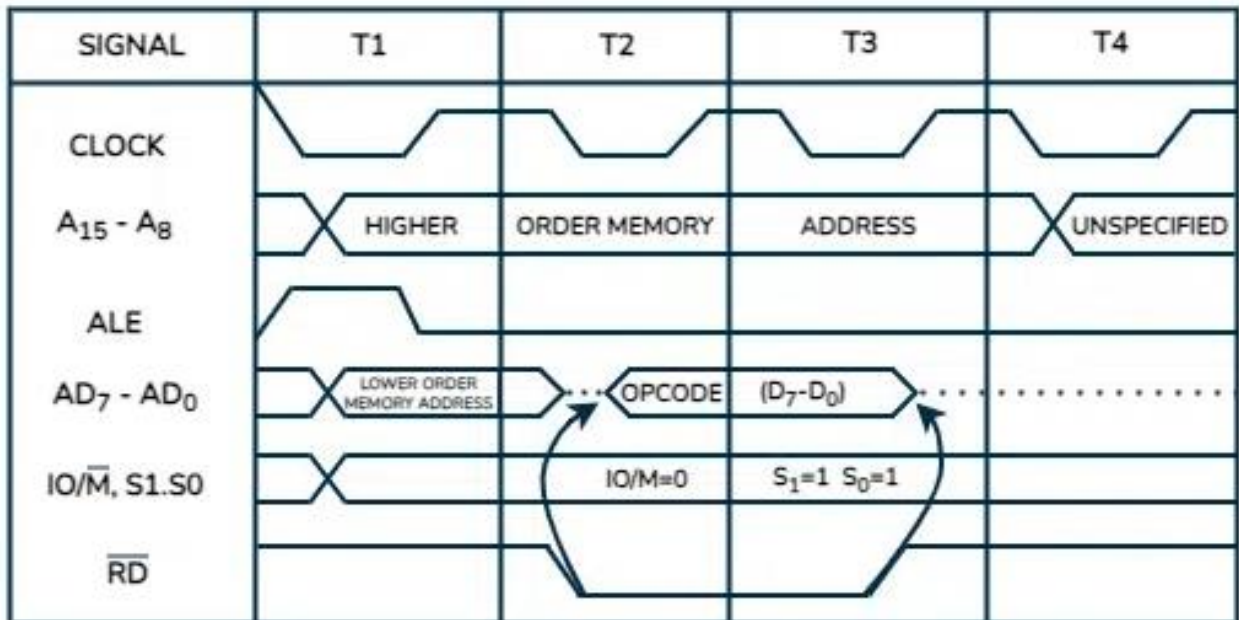enable), RD (read), WR (write), and IO/M (input/output) memory.

Machine Cycle of 8085

The 8085 microprocessor has 5 basic machine cycles. They are :

- Opcode Fetch {4T- state}

- Memory Read {3T- state}

- Memory Write {3T- state}

- I/O Read {3T- state}

- I/O Write {3T- state}

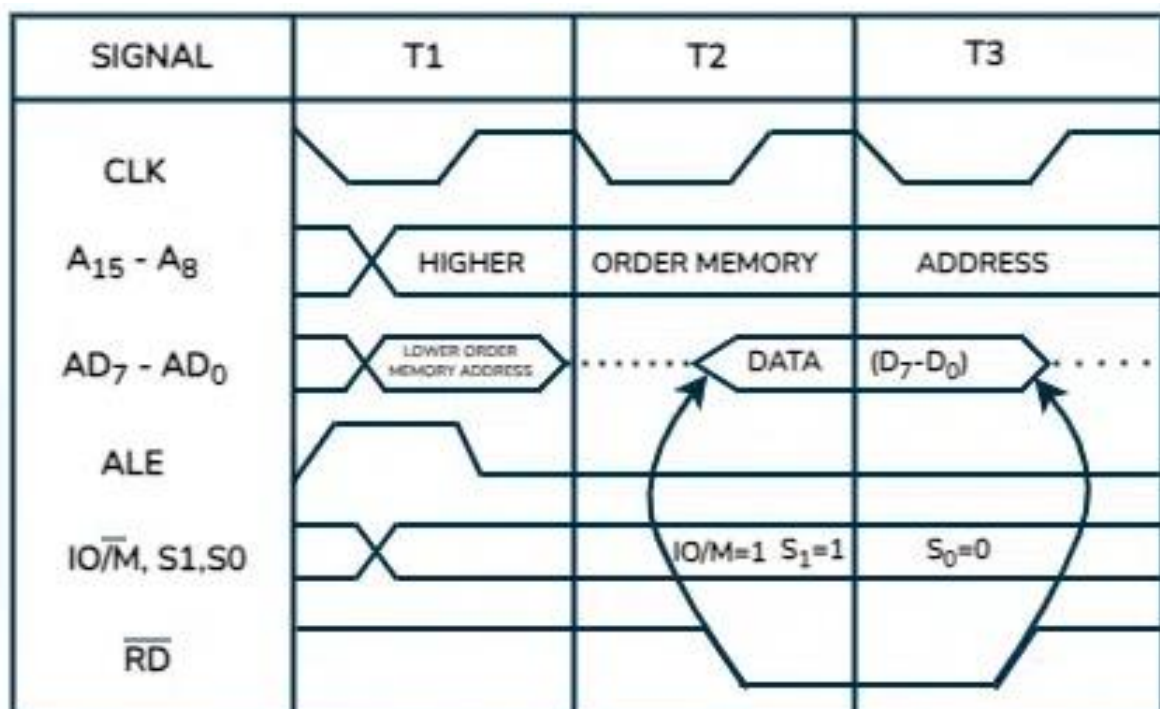**Opcode Fetch Machine Cycle of 8085**

- Each processor instruction has a one byte opcode.

- Operation codes are stored in memory. Thus, the processor performs an opcode load machine cycle to load the opcode from memory.

- Thus, each instruction begins with a machine cycle of opcodes.

- The time the processor takes to perform an opcode load cycle is 4T.

- At this time, the first, 3T-states are used to load the opcode from memory, and the remaining T-states are used for internal opcodes.

SIGNAL	T1	T2	T3	T4
CLOCK				
$A_{15} - A_8$	HIGHER	ORDER MEMORY	ADDRESS	UNSPECIFIED
ALE				
$AD_7 - AD_0$	LOWER ORDER MEMORY ADDRESS	OPCODE	$(D_7 - D_0)$	
IO/$\overline{M}$, S1.S0		IO/M=0	S_1=1 S_0=1	
$\overline{RD}$				

Opcode Fetch Cycle

Memory Read Machine Cycle of 8085
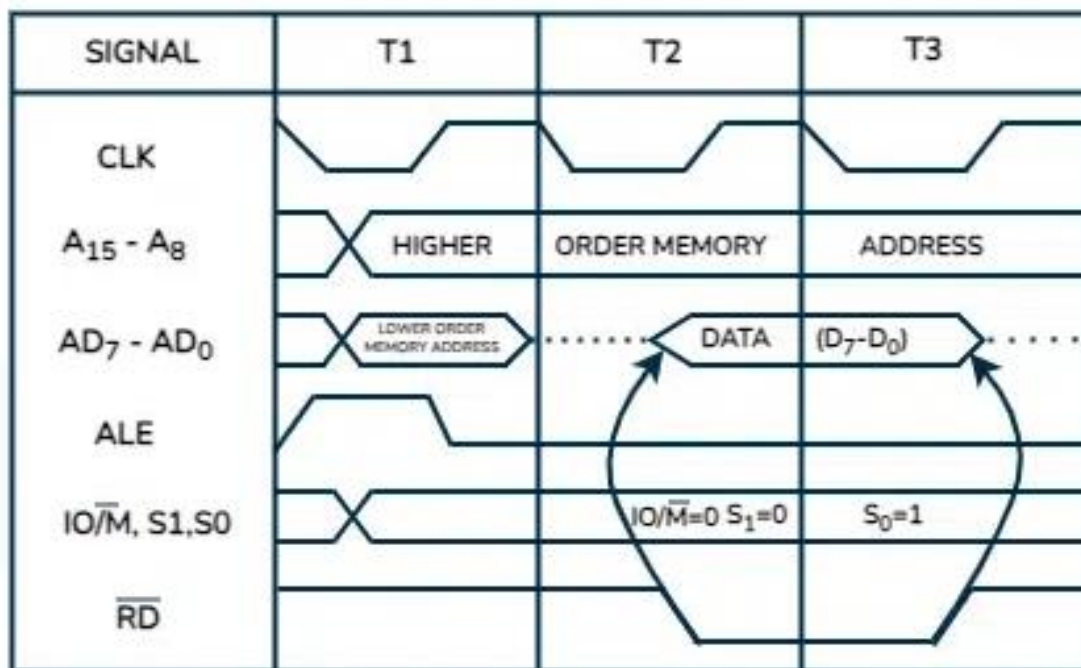
- A memory read machine cycle is executed by the processor to read a data byte from memory.

- The processor takes 3T states to perform this cycle.

- Instructions that have more than one byte word will use machine cycle after machine cycle to load the opcode.

SIGNAL	T1	T2	T3
CLK			
$A_{15} - A_8$	HIGHER	ORDER MEMORY	ADDRESS
$AD_7 - AD_0$	LOWER ORDER MEMORY ADDRESS	DATA	$(D_7 - D_0)$
ALE			
IO/$\overline{M}$, S1,S0		IO/M=1 S_1=1	S_0=0
$\overline{RD}$			

Memory Write Machine Cycle of 8085
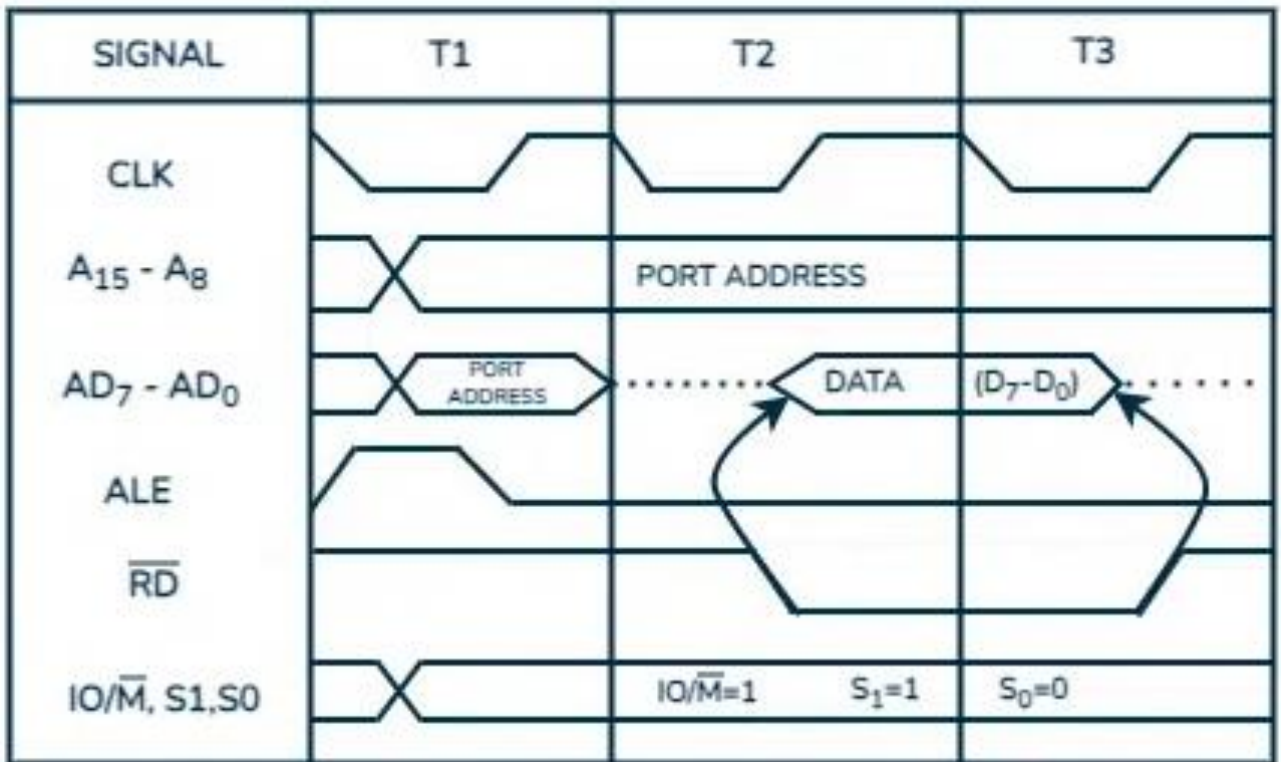
Memory Write Machine Cycle of 8085

- A write-to-memory machine cycle is executed by the processor to write a data byte to memory.

- The processor takes 3T states to perform this cycle.

- Instructions that have more than one byte word will use machine cycle after machine cycle to load the opcode.

SIGNAL	T1	T2	T3
CLK			
$A_{15} - A_8$	HIGHER	ORDER MEMORY	ADDRESS
$AD_7 - AD_0$	LOWER ORDER MEMORY ADDRESS	DATA	(D_7-D_0)
ALE			
IO/$\overline{M}$, S1,S0		IO/$\overline{M}$=0 S_1=0	S_0=1
$\overline{RD}$			

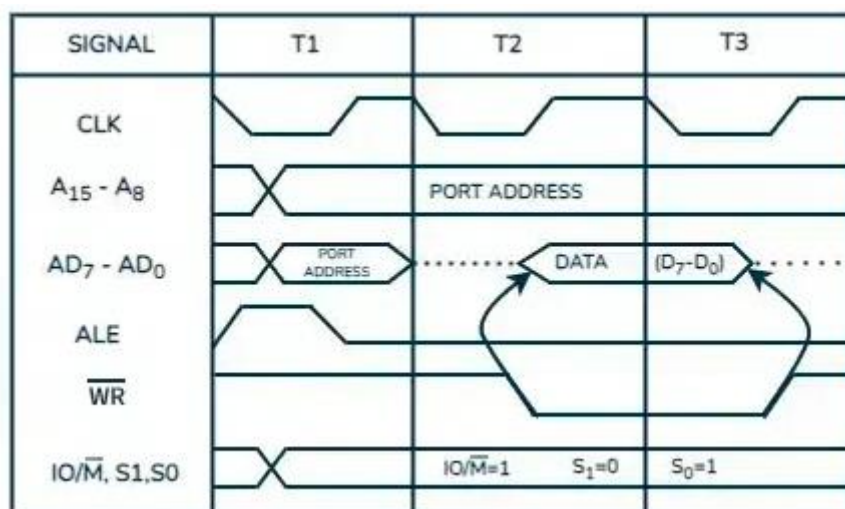**Memory Write Machine Cycle**

I/O Read Machine Cycle

- A reader I/O cycle is performed by the processor to read a data byte from an I/O port or peripheral that is I/O mapped in the system.

- The processor needs 3T states to execute this machine cycle.

SIGNAL	T1	T2	T3
CLK			
$A_{15} - A_8$		PORT ADDRESS	
$AD_7 - AD_0$	PORT ADDRESS	DATA	(D_7-D_0)
ALE			
$\overline{RD}$			
$IO/\overline{M}$, S1,S0		$IO/\overline{M}=1$	$S_1=1$ $S_0=0$

I/O Read Machine Cycle

I/O Write Machine Cycle

- A writer's I/O cycle is executed by the processor to write a data byte to an I/O port or peripheral that is I/O mapped in the system.

- The processor needs 3T states to execute this machine cycle.

SIGNAL	T1	T2	T3
CLK			
$A_{15} - A_8$		PORT ADDRESS	
$AD_7 - AD_0$	PORT ADDRESS	DATA	(D_7-D_0)
ALE			
$\overline{WR}$			
$IO/\overline{M}$, S1,S0		$IO/\overline{M}=1$	$S_1=0$ $S_0=1$

I/O Write Machine Cycle
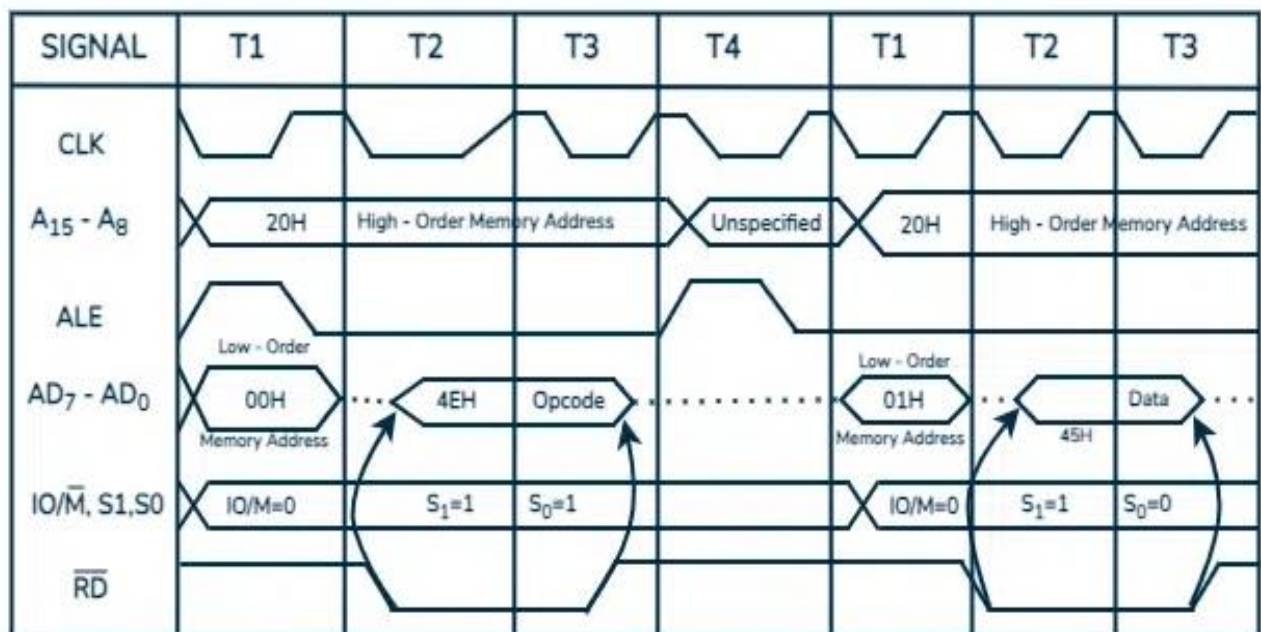
Timing Diagram of Some Instructions

- MVI A,45

- STA 8000H

- INR M

Timing Diagram of MVI A,45

- Loading opcode 4EH from memory 2000H. (Operation Code Loading Machine Cycle)

- Read (move) data 45H from memory 2001H. (memory reading).

Address	Mnemonics	Opcode
2000H	MVI A,45H	4EH
2001H		45H

| SIGNAL	T1	T2	T3	T4	T1	T2	T3
CLK							
$A_{15} - A_8$	20H	High - Order Memory Address		Unspecified	20H	High - Order Memory Address	
ALE							
$AD_7 - AD_0$	Low - Order 00H Memory Address	4EH	Opcode		Low - Order 01H Memory Address	45H	Data
IO/$\overline{M}$, S1,S0	IO/M=0	$S_1=1$	$S_0=1$		IO/M=0	$S_1=1$	$S_0=0$
$\overline{RD}$							

Timing Diagram of MVI Instructions

Timing Diagram of STA 8000H

- STA stands for Store Accumulator - the contents of the accumulator are stored at (1000H).

- The transaction code of STA indicates 32H. It is provided by 1000H memory (see picture). - In the machine cycle.

- Then read the sub address (00). - Memory reading machine cycle

- Read higher memory location (80) – Memory read machine cycle.

- Consider the combination of the two addresses and write the details of the accumulator for 8000. - Don't forget to write the machine cycle

- Assume the memory address for the instruction and let the content of accumulator is C7H. So, C7H from accumulator is now stored in 8000H.

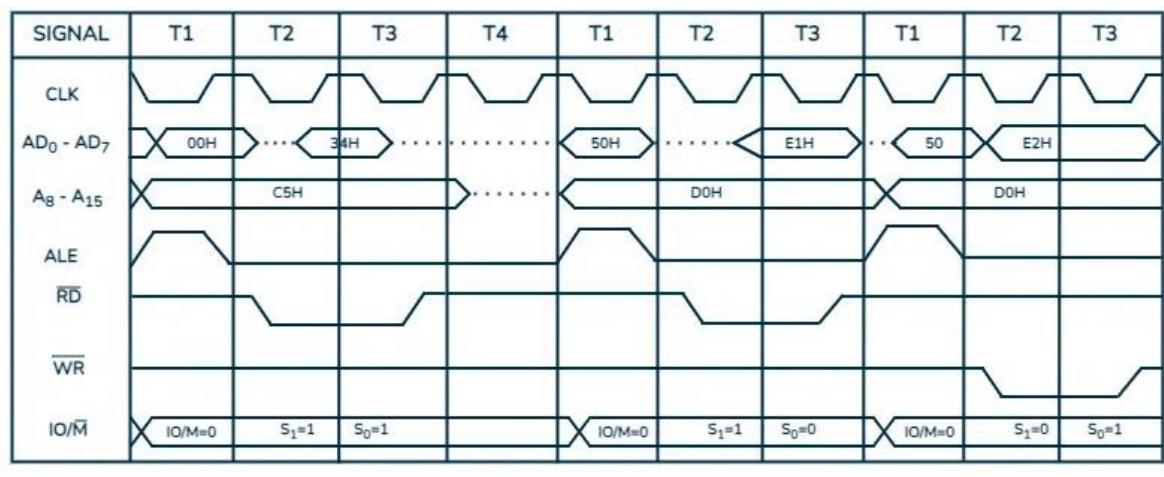| Address	Mnemonics	Opcode
1000H	STA 8000H	32H
1001H		00H
1002H		80H
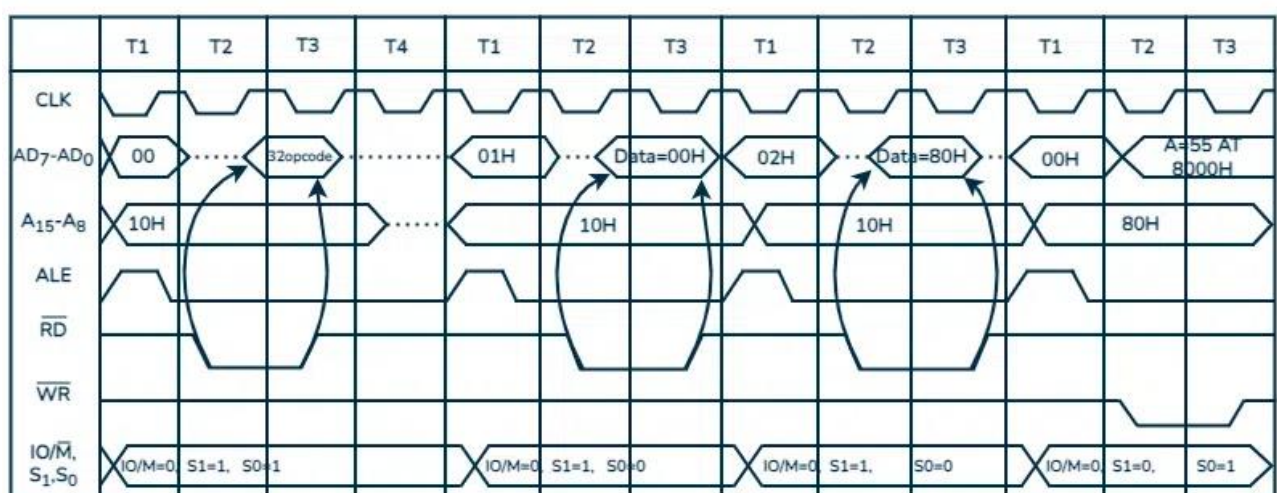
Timing Diagram of STA Instructions

Timing Diagram of INR M Instructions

- C500H INR M 34H

- D050 as memory address
- E1H and E2H as opcode

SIGNAL	T1	T2	T3	T4	T1	T2	T3	T1	T2	T3
CLK										
$AD_0 - AD_7$	00H	34H			50H		E1H	50	E2H	
$A_8 - A_{15}$		C5H				D0H			D0H	
ALE										
$\overline{RD}$										
$\overline{WR}$										
IO/$\overline{M}$	IO/M=0	$S_1=1$	$S_0=1$		IO/M=0	$S_1=1$	$S_0=0$	IO/M=0	$S_1=0$	$S_0=1$

Timing Diagram of INR Instructions

Applications 0f 8085

- System charts keep track of all changes occurring in the system.

- Timelines provide a more accurate representation of physical activity.

- The timing table of 8085 instructions graphically represents the time required to execute each instruction.

- Timing Diagram is a special form of sequence diagram.

- Timing Diagram keep track of every change that occur in the system.

- The timing diagram is important for planning the clock frequency of the microprocessor.

- The timing diagram helps design memory circuits that meet the required set and hold times during read and write operations.

Conclusion

The timing diagram of the 8085 microprocessor looks like a brief overview of how the microprocessor works during

operation. It helps engineers solve problems, create good memories and interactions between devices, make quick decisions, and do good, on-time work. For students, it simplifies the learning process by providing a visual representation of the inner workings of a microprocessor. Essentially, it serves as a guide for understanding, troubleshooting and optimizing the performance of the 8085 microprocessor.

Problem – Draw the timing diagram of the given instruction in 8085, MOV B, C

Given instruction copy the contents of the source register into the destination register and the contents of the source register are not altered.

Example:

MOV B, C

Opcode: MOV

Operand: B and C

B is the destination register and C is the source register whose contents need to be transferred to the destination register.

Algorithm – The instruction MOV B, C is of 1 byte; therefore the complete instruction will be stored in a single memory address. For example:

2000: MOV B, C

Only opcode fetching is required for this instruction and thus we need 4 T states for the timing diagram.

For the opcode fetch the IO/M (low active) = 0, S1 = 1 and S0 = 1. The timing diagram of MOV instruction is shown below:

- 00 – lower bit of address where the opcode is stored, i.e., 00.

- 20 – higher bit of address where the opcode is stored, i.e., 20.

- ALE – provides signal for multiplexed address and data bus. Only in t1 is it used as an address bus to fetch a lower bit of address otherwise it will be used as the data bus.

- RD (low active) – signal is 1 in t1 & t4 as no data is read by the microprocessor. Signal is 0 in t2 & t3 because here the data is read by a microprocessor.

- WR (low active) – signal is 1 throughout, no data is written by a microprocessor.

- IO/M (low active) – signal is 1 throughout because the operation is performing on memory.

- S0 and S1 – both are 1 in case of opcode fetching.

In Opcode fetch (t1-t4 T states):