5

TIMING DIAGRAM OF 8085

5.1 INTRODUCTION

Timing diagram is the display of initiation of read/write and transfer of data operations under the control of 3-status signals $IO/\overline{M}$, S_1, and S_0. As the heartbeat is required for the survival of the human being, the CLK is required for the proper operation of different sections of the microprocessors. All actions in the microprocessor is controlled by either leading or trailing edge of the clock. If I ask a man to bring 6-bags of wheat, each weighing 100 kg, he may take 6-times to perform this task in going and bringing it. A stronger man might perform the same task in 3-times only. Thus, it depends on the strength of the man to finish the job quickly or slowly. Here, we can assume both weaker and strong men as machine. The weaker man has taken 6-machine cycle (6-times going and coming with one bag each time) to execute the job where as the stronger man has taken only 3-machine cycle for the same job. Similarly, a machine may execute one instruction in as many as 3-machine cycles while the other machine can take only one machine cycle to execute the same instruction. Thus, the machine that has taken only one machine cycle is efficient than the one taking 3-machine cycle. Each machine cycle is composed of many clock cycle. Since, the data and instructions, both are stored in the memory, the µP performs fetch operation to read the instruction or data and then execute the instruction. The µP in doing so may take several cycles to perform fetch and execute operation. The 3-status signals : $IO/\overline{M}$, S_1, and S_0 are generated at the beginning of each machine cycle. The unique combination of these 3-status signals identify read or write operation and remain valid for the duration of the cycle. Table-5.1(*a*) shows details of the unique combination of these status signals to identify different machine cycles. Thus, time taken by any µP to execute one instruction is calculated in terms of the clock period.

The execution of instruction always requires read and writes operations to transfer data to or from the µP and memory or I/O devices. Each read/ write operation constitutes one machine cycle (MC_1) as indicated in Fig. 5.1 (*a*). Each machine cycle consists of many clock periods/ cycles, called **T-states**. The heartbeat of the microprocessor is the clock period. Each and every operation inside the microprocessor is under the control of the clock cycle. The clock signal determines the time taken by the microprocessor to execute any instruction. The clock cycle shown in Fig. 5.1 (*a*) has two edges (leading and trailing or lagging). State is defined as the time interval between 2-trailing or leading edges of the clock. Machine cycle is the time required to transfer data to or from memory or I/O devices.

Table 5.1(*a*) Machine cycle status and control signals

Machine cycle	*Status*			*Controls*		
	IO / $\overline{M}$	S_1	S_0	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch (OF)	0	1	1	0	1	1
Memory Read	0	1	0	0	1	1
Memory Write	0	0	1	1	0	1
I/O Read (I/OR)	1	1	0	0	1	1
I/O Write (I/OW)	1	0	1	1	0	1
Acknowledge of INTR (INTA)	1	1	1	1	1	0
BUS Idle (BI) : DAD	0	1	0	1	1	1
ACK of RST, TRAP	1	1	1	1	1	1
HALT	Z	0	0	Z	Z	1
HOLD	Z	X	X	Z	Z	1
X ⇒ Unspecified, and Z ⇒ High impedance state						

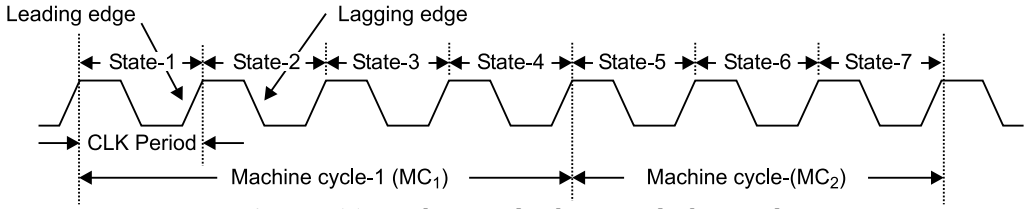

Fig. 5.1 (*a*) Machine cycle showing clock periods

5.2 PROCESSOR CYCLE

The function of the microprocessor is divided into fetch and execute cycle of any instruction of a program. The program is nothing but number of instructions stored in the memory in sequence. In the normal process of operation, the microprocessor fetches (receives or reads) and executes one instruction at a time in the sequence until it executes the halt (HLT) instruction. Thus, an instruction cycle is defined as the time required to fetch and execute an instruction. For executing any program, basically 2-steps are followed sequentially with the help of clocks

- Fetch, and
- Execute.

The time taken by the μP in performing the fetch and execute operations are called fetch and execute cycle. Thus, sum of the fetch and execute cycle is called the instruction cycle as indicated in Fig. 5.2 (*a*).

Instruction Cycle (IC) = Fetch cycle (FC) + Execute Cycle (EC)

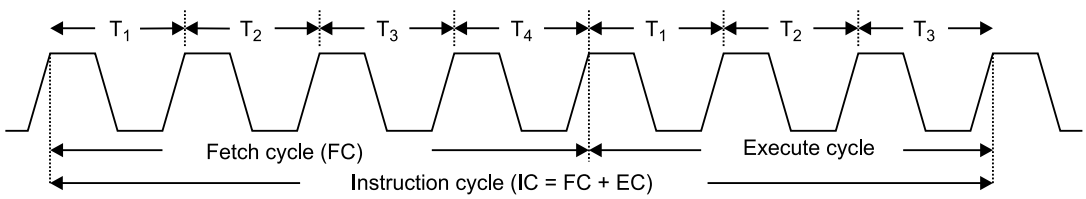

Fig. 5.2 (*a*) Processor cycle

These cycles have been illustrated in Figs. 5.2(*a*) and (*b*). Each read or writes operation constitutes a machine cycle. The instructions of 8085 require 1–5 machine cycles containing 3–6 states (clocks). The 1st machine cycle of any instruction is always an Op. Code fetch cycle in which the processor decides the nature of instruction. It is of at least 4-states. It may go up to 6-states.

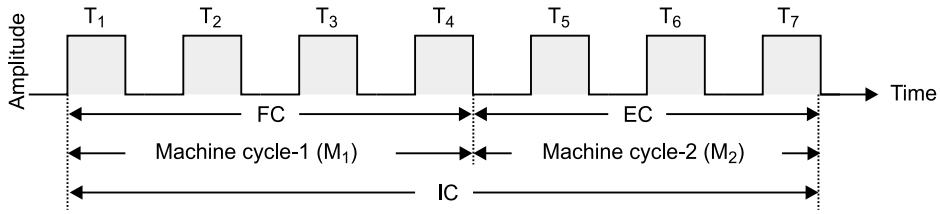

Fig. 5.2 (*b*) Ideal wave shape relationship for FC, EC, MC, and IC.

It is well known that an instruction cycle consists of many machine cycles. Each machine cycle consists of many clock periods or cycles, called T-states. The 1st machine cycle (M_1) of every instruction cycle is the **opcode fetch** cycle. In the opcode fetch cycle, the processor comes to know the nature of the instruction to be executed. The processor during (M_1 cycle) puts the program counter contents on the address bus and reads the opcode of the instruction through **read process**. The T_1, T_2, and T_3 clock cycles are used for the basic memory read operation and the T_4 clock and beyond are used for its interpretation of the opcode. Based on these interpretations, the µP comes to know the type of additional information/data needed for the execution of the instruction and accordingly proceeds further for 1 or 2-machine cycle of memory read and writes.

The Op. code fetch cycle is of fixed duration (normally 4-states), whereas the instruction cycle is of variable duration depending on the length of the instruction. As an example, STA instruction, requires opcode fetch cycle, lower-order address fetch cycle and higher order fetch cycle and then the execute cycle. Thus opcode fetch cycle is of one machine cycle in this example. A particular microprocessor requires a definite time to performing a specific task. This time is called **machine cycle**. Thus, one machine cycle is required each time the µP access I/O port or memory. A fetch opcode cycle is always 1-machine cycle, whereas, execute cycle may be of one or more machine cycle depending upon the length of the instruction.

Instruction Fetch (FC) ⇒ An instruction of 1 or 2 or 3-bytes is extracted from the memory locations during the fetch and stored in the µP's instruction register.

Instruction Execute (EC) ⇒ The instruction is decoded and translated into specific activities during the execution phase. Thus, in an instruction cycle, instruction fetch, and instruction execute cycles are related as depicted in Fig. 5.2 (*a*). Every instruction cycle consists of 1, 2, 3, 4 or 5-machine cycles as indicated in Fig. 5.2 (*c*). One machine cycle is required each time the µP access memory or I/O port. The fetch cycle, in general could be 4 to 6-states whereas the execute cycle could of 3 to 6-states. The 1st machine cycle of any instruction is always the fetch cycle that provides identification of the instruction to be executed.

The fetch portion of an instruction cycle requires one machine cycle for each byte of instruction to be fetched. Since instruction is of 1 to 3 bytes long, the instruction fetch is one to 3-machine cycles in duration. The 1st machine cycle in an instruction cycle is always an **opcode**

fetch. The 8-bits obtained during an opcode fetch are always interpreted as the Opcode of an instruction. The machine cycle including wait states is shown in Fig. 5.2 (*c*).

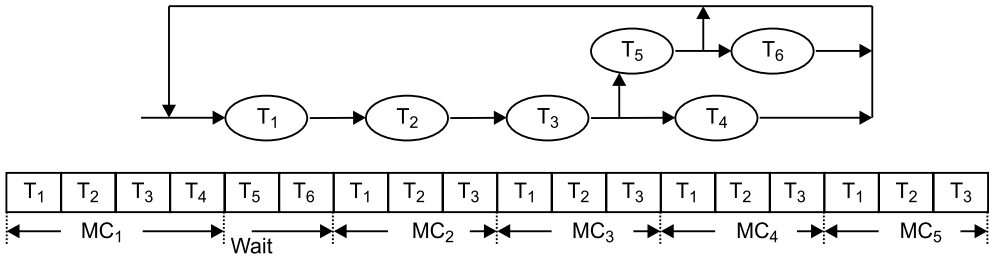

Fig. 5.2 (*c*) Machine cycle including wait states

Note : Some instructions do not require any machine cycle other than that necessary to fetch the instruction. Other instructions, however, require additional machine cycles to write or read data to or from memory or I/O devices.

A typical fetch cycle is explained in Fig. 5.2 (*d*). In Fig. 5.2 (*d*) only two clock cycles have been shown as the requirement to read the instruction. Since the access time of the memory may vary and it may require more than 2-clock cycles, the microprocessor has to wait for more than 2-clocks duration before it receives the opcode instruction. Hence, most of the microprocessors have the provisions of introducing wait cycle within the fetch cycle to cope up with the slow memories or I/O devices.

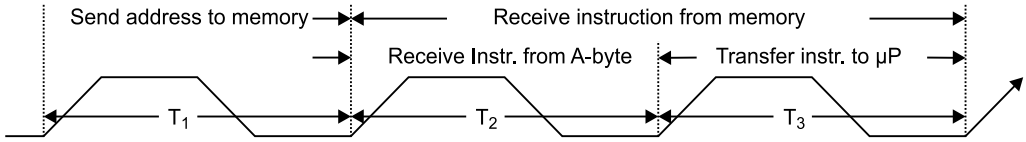

Fig. 5.2 (*d*) Fetch cycle

Opcode Fetch

A microprocessor either reads or writes to the memory or I/O devices. The time taken to read or write for any instruction must be known in terms of the µP clock. The 1st step in communicating between the microprocessor and memory is reading from the memory. This reading process is called opcode fetch. The process of opcode fetch operation requires minimum 4-clock cycles T_1, T_2, T_3, and T_4 and is the 1st machine cycle (M_1) of every instruction.

In order to differentiate between the data byte pertaining to an opcode or an address, the machine cycle takes help of the status signal $IO/\overline{M}$, S_1, and S_0. The $IO/\overline{M}$ = 0 indicates memory operation and $S_1 = S_0 = 1$ indicates Opcode fetch operation.

The opcode fetch machine cycle M_1 consists of 4-states (T_1, T_2, T_3, and T_4). The 1st 3-states are used for fetching (transferring) the byte from the memory and the 4th-state is used to decode it.

Thus, thorough understanding about the communication between memory and microprocessor can be achieved only after knowing the processes involved in reading or writing into the memory by the microprocessor and time taken w.r.t. its clock period. This can be explained by examples.

The process of implementation of each instruction follows the fetch and execute cycles. In other words, first the instruction is fetched from memory and then executed. Figs. 5.2 (*e*) and (*f*) depict these 2-steps for implementation of the instruction ADI 05H. Let us assume that the accumulator contains the result of previous operation i.e., 03H and instruction is held at memory locations 2030H and 2031H.

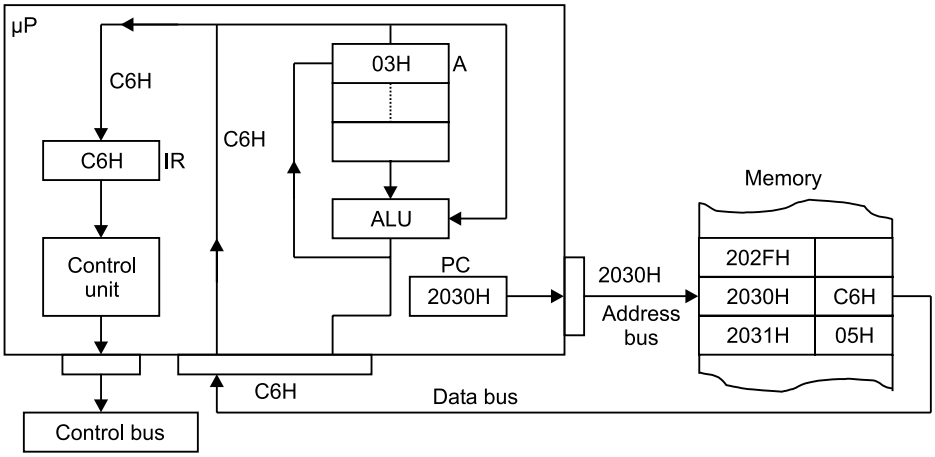

Fig. 5.2 (*e*) Instruction fetch : reads 1st byte (Opcode) in instruction register (IR)

The fetch part of the instruction is the same for every instruction. The control unit puts the contents of the program counter (PC) 2030H on the address bus. The 1st byte (opcode C6H in this example) is passed to the instruction register. In the execute cycle of the instruction, the control unit examines the opcode and as per interpretation further memory read or write operations are performed depending upon whether additional information/ data are required or not. In this case, the data 05H from the memory is transferred through the data bus to the ALU. At the same time the control unit sends the contents of the accumulator (03H) to the ALU and performs the addition operation. The result of the addition operation 08H is passed to the accumulator overriding the previous contents 03H. On the completion of one instruction, the program counter is automatically incremented to point to the next memory location to execute the subsequent instruction.

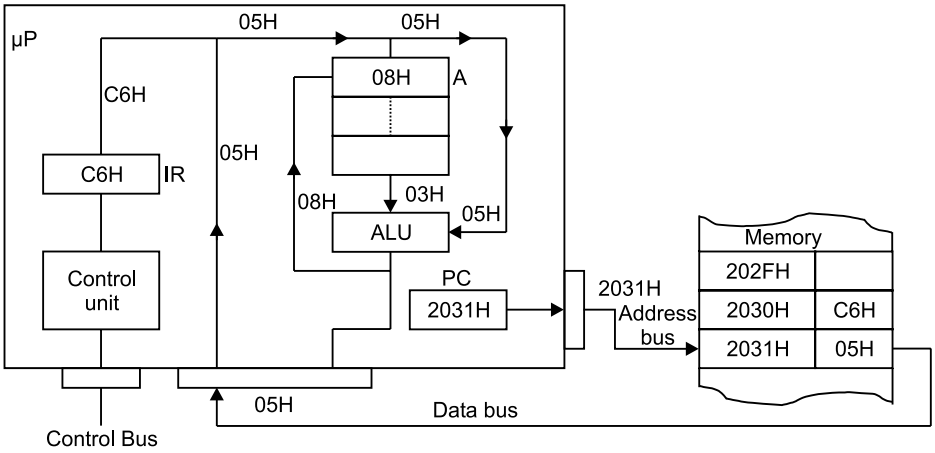

Fig. 5.2 (*f*) Instruction execute : reads 2nd byte from memory and adds to accumulator

Note : The slope of the edges of the clock pulses has been shown to be much exaggerated to indicate the existence of rise and fall time.

5.3 TIMING DIAGRAM OF OPCODE FETCH

The process of opcode fetch operation requires minimum 4-clock cycles T_1, T_2, T_3, and T_4 and is the 1st machine cycle (M_1) of every instruction.

Example

Fetch a byte 41H stored at memory location 2105H.

For fetching a byte, the microprocessor must find out the memory location where it is stored. Then provide condition (control) for data flow from memory to the microprocessor. The process of data flow and timing diagram of fetch operation are shown in Figs. 5.3 (*a*), (*b*), and (*c*). The µP fetches opcode of the instruction from the memory as per the sequence below

- A low $IO/\overline{M}$ means microprocessor wants to communicate with memory.
- The µP sends a high on status signal S_1 and S_0 indicating fetch operation.
- The µP sends 16-bit address. AD bus has address in 1st clock of the 1st machine cycle, T_1.
- AD_7 to AD_0 address is latched in the external latch when ALE = 1.
- AD bus now can carry data.
- In T_2, the $\overline{RD}$ control signal becomes low to enable the memory for read operation.
- The memory places opcode on the AD bus
- The data is placed in the data register (DR) and then it is transferred to IR.

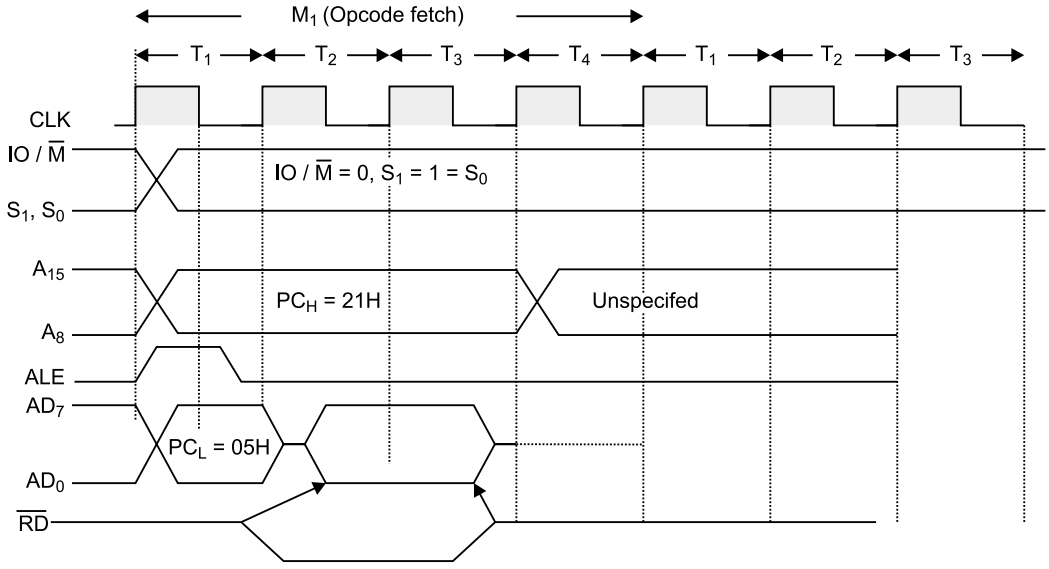

Fig. 5.3 (*a*) Opcode fetch

- During T_3 the $\overline{RD}$ signal becomes high and memory is disabled.
- During T_4 the opcode is sent for decoding and decoded in T_4.
- The execution is also completed in T_4 if the instruction is single byte.
- More machine cycles are essential for 2- or 3-byte instructions. The 1st machine cycle M_1 is meant for fetching the opcode. The machine cycles M_2 and M_3 are required either to read/ write data or address from the memory or I/O devices.

Example

Opcode fetch MOV B,C.

T_1 : The 1st clock of 1st machine cycle (M_1) makes ALE high indicating address latch enabled which loads low-order address 00H on $AD_7 \Leftrightarrow AD_0$ and high-order address 10H simultaneously on $A_{15} \Leftrightarrow A_8$. The address 00H is latched in T_1.

T_2 : During T_2 clock, the microprocessor issues $\overline{RD}$ control signal to enable the memory and memory places 41H from 1000H location on the data bus.

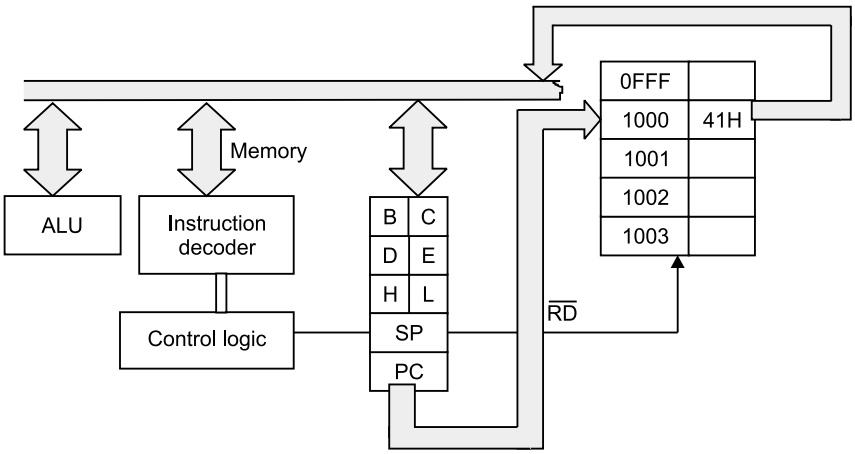

Fig. 5.3 (*b*) Data flow from memory to microprocessor

T_3 : During T_3, the 41H is placed in the instruction register and $\overline{RD}$ = 1 (high) disables signal. It means the memory is disabled in T_3 clock cycle. The opcode cycle is completed by end of T_3 clock cycle.

T_4 : The opcode is decoded in T_4 clock and the action as per 41H is taken accordingly. In other word, the content of C-register is copied in B-register. Execution time for opcode 41H is

Clock frequency of 8085 = 3.125 MHz

Time (T) for one clock = 1/3.125 MHz = 325.5 ns = 0.32 μS

Execution time for opcode fetch = 4T = 4*0.32 μS = 1.28 μS

Explain the execution of MVI B,05H stored at locations indicated below

Mnemonics	Machine code	Memory Locations
MVI B, 05H	06H	1000H
	05H	1001H

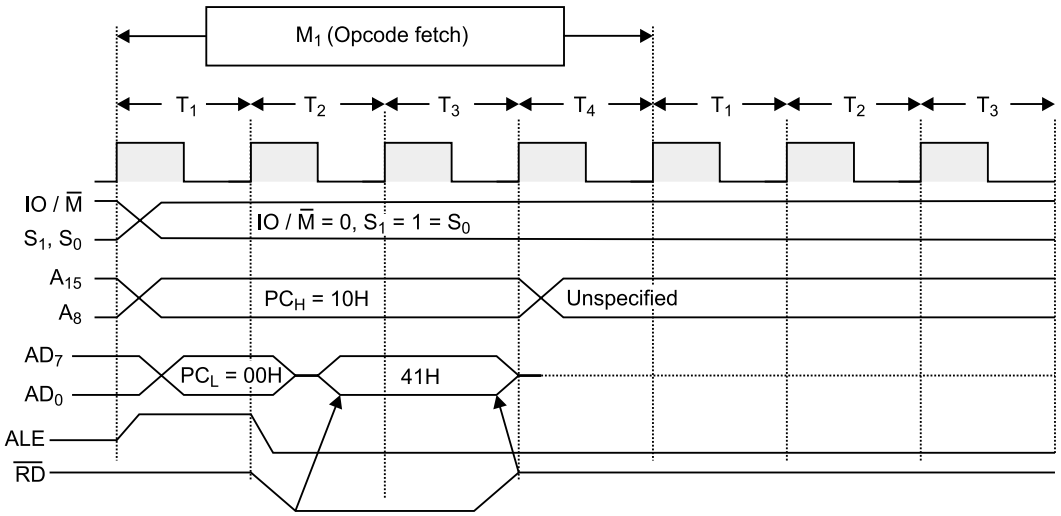

Fig. 5.3 (*c*) Opcode fetch (MOV B,C)

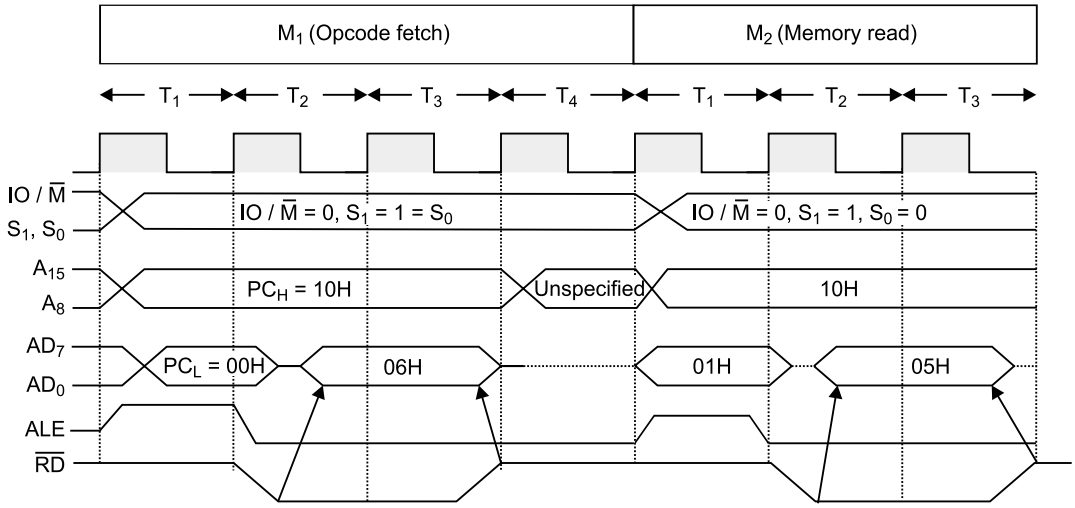

Fig. 5.3 (*d*) Timing diagram for MVI B,05H

The MVI B,05H instruction requires 2-machine cycles (M_1 and M_2). M_1 requires 4-states and M_2 requires 3-states, total of 7-states as shown in Fig. 5.3 (*d*). Status signals $IO/\overline{M}$, S_1 and S_0 specifies the 1st machine cycle as the op-code fetch.

In T_1-state, the high order address {10H} is placed on the bus $A_{15} \Leftrightarrow A_8$ and low-order address {00H} on the bus $AD_7 \Leftrightarrow AD_0$ and ALE = 1. In T_2-state, the $\overline{RD}$ line goes low, and the data 06H from memory location 1000H are placed on the data bus. The fetch cycle becomes complete in T_3-state. The instruction is decoded in the T_4-state. During T_4-state, the contents of the bus are unknown. With the change in the status signal, $IO/\overline{M}$ = 0, S_1 = 1 and S_0 = 0, the 2nd machine cycle is identified as the memory read. The address is 1001H and the data byte

[05H] is fetched via the data bus. Both M_1 and M_2 perform memory read operation, but the M_1 is called op-code fetch i.e., the 1st machine cycle of each instruction is identified as the opcode fetch cycle. Execution time for MBI B,05H i.e., memory read machine cycle and instruction cycle is

Mnemonic	*Instruction Byte*	*Machine Cycle*	*T-sstates*
MVI B,05H	Opcode	Opcode Fetch	4
	Immediate Data	Read Immediate Data	3
			7

Clock frequency of 8085 = 3.125 MHz

Time (T) for one clock = 1/3.125 MHz = 0.32 μS

Time for Memory Read = 3T = 3*0.320 μS = 0.96 μS

Total Execution time for Instruction = 7T = 7*0.320 μS = 2.24 μS

Read Cycle

The high order address ($A_{15} \Leftrightarrow A_8$) and low order address ($AD_7 \Leftrightarrow AD_0$) are asserted on 1st low going transition of the clock pulse. The timing diagram for $IO/\overline{M}$ read are shown in Fig. 5.3 (*e*) and (*f*). The $A_{15} \Leftrightarrow A_8$ remains valid in T_1, T_2, and T_3 i.e. duration of the bus cycle, but $AD_7 \Leftrightarrow AD_0$ remains valid only in T_1. Since it has to remain valid for the whole bus cycle, it must be saved for its use in the T_2 and T_3.

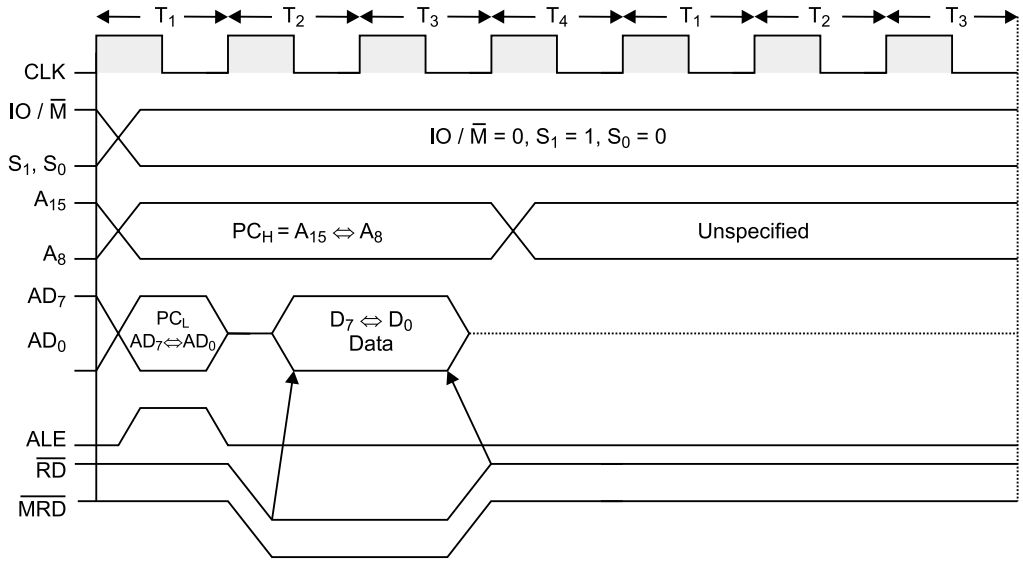

Fig. 5.3 (*e*) Memory read timing diagram

ALE is asserted at the beginning of T_1 of each bus cycle and is negated towards the end of T_1. ALE is active during T_1 only and is used as the clock pulse to latch the address ($AD_7 \Leftrightarrow AD_0$) during T_1. The $\overline{RD}$ is asserted near the beginning of T_2. It ends at the end of T_3. As soon as the $\overline{RD}$ becomes active, it forces the memory or I/O port to assert data. $\overline{RD}$ becomes inactive towards the end of T_3, causing the port or memory to terminate the data.

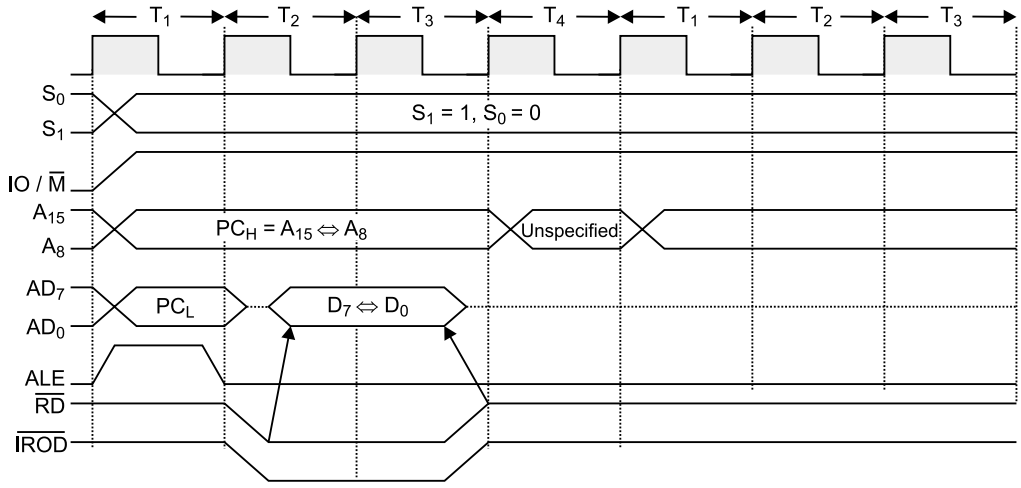

Fig. 5.4 (*f*) I/O Read timing diagram

Write Cycle

Immediately after the termination of the low order address, at the beginning of the T_2, data is asserted on the address/data bus by the processor. $\overline{WR}$ control is activated near the start of T_2 and becomes inactive at the end of T_3. The processor maintains valid data until after $\overline{WR}$ is terminated. This ensures that the memory or port has valid data while $\overline{WR}$ is active.

It is clear from Figs. 5.3 (*g*) and (*h*) that for READ bus cycle, the data appears on the bus as a result of activating $\overline{RD}$ and for the $\overline{WR}$ bus cycle, the time the valid data is on the bus overlaps the time that the $\overline{WR}$ is active.

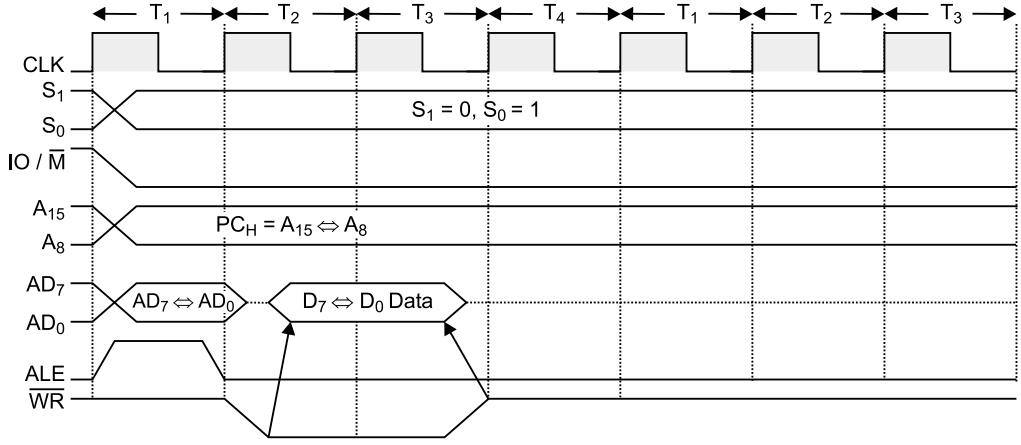

Fig. 5.3 (*g*) Memory write timing diagram.

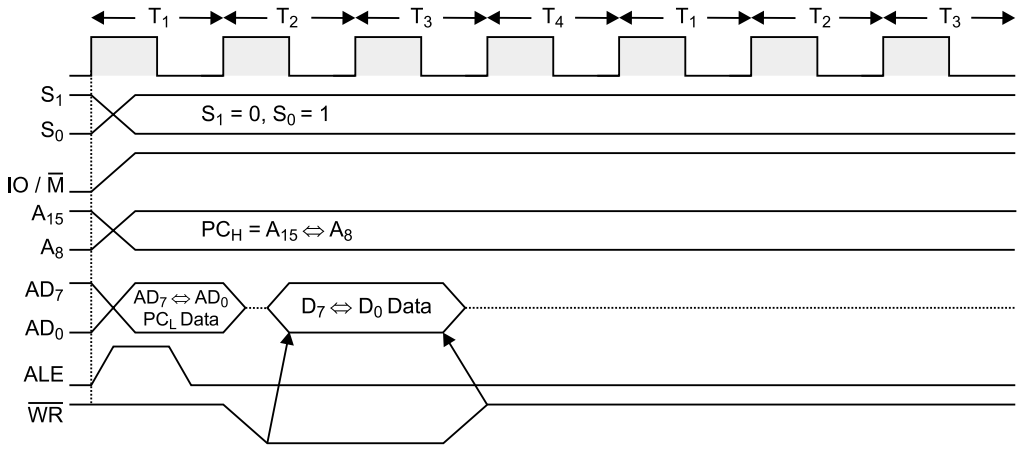

Fig. 5.3 (*h*) I/O write timing diagram

STA

The STA instruction stands for storing the contents of the accumulator to a memory location whose address is immediately available after the instruction (STA). The 8085 have 16-address lines, it can address 2^{16} = 64 K. Since the STA instruction is meant to store the contents of the accumulator to the memory location, it is a 3-byte instruction. 1st byte is the opcode, the 2nd and 3rd bytes are the address of the memory locations. The storing of the STA instruction in the memory locations is as

Opcode	1st byte
Low address	2nd byte
High address	3rd byte

Three machine cycles are required to fetch this instruction : opcode Fetch transfers the opcode from the memory to the instruction register. The 2-byte address is then transferred, 1-byte at a time, from the memory to the temporary register. This requires two Memory read machine cycles. When the entire instruction is in the microprocessor, it is executed. The execution process transfers data from the microprocessor to the memory. The contents of the accumulator are transferred to memory, whose address was previously transferred to the microprocessor by the preceding 2-Memory Read machine cycles. The address of the memory location to be written is generated as

Mnemonic	*Instruction Byte*	*Machine Cycle*	*T-states*
STA	Opcode	Opcode Fetch	4
	LOW Address	Memory Read	3
	HIGH Address	Memory Read	3
		Memory Write	3
			13

The high order address byte in the temporary register is transferred to the address latch and the low order address byte is transferred to the address/data latch. This data transfer is affected

by a Memory Write machine cycle. Thus 3-byte STA instruction has four machine cycles in its instruction cycle.

The timing and control section of the microprocessor automatically generates the proper machine cycles required for an instruction cycle from the information provided by the opcode. The timing diagram of the instruction STA is shown in Fig. 5.3 (*i*). The status of $IO / \overline{M}$, S_1 and S_0 for 4-machine cycles are obtained from Table 5.1. The condition of $IO / \overline{M}$, S_1 and S_0 would be 0, 1 and 1 respectively in MC_1. The status of ALE is high at the beginning of 1st state of each machine cycle so that $AD_7 \Leftrightarrow AD_0$ work as the address bus. $\overline{RD}$ remains high during 1st state of each machine cycle, since during 1st state of each machine cycle $AD_7 \Leftrightarrow AD_0$ work as address bus. It remains high during 4th state of the 1st machine cycle also as the 4th state is used to decode the op code for generating the required control signals.

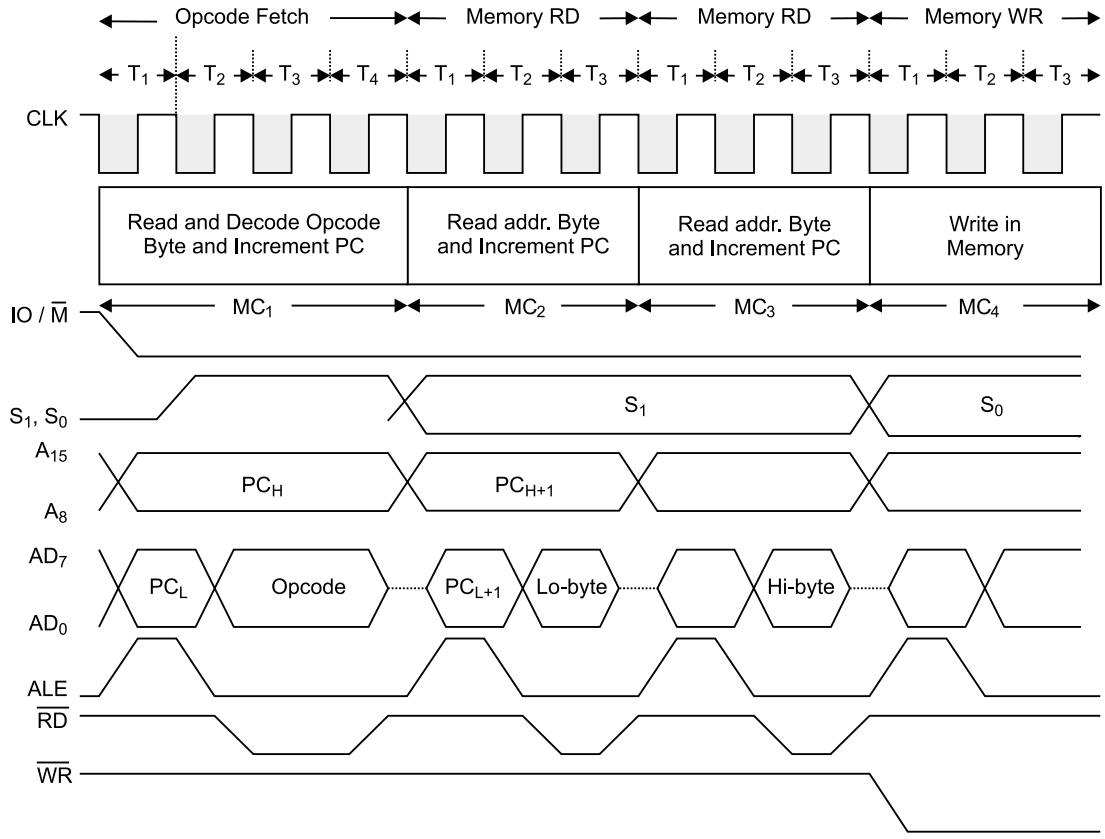

Fig. 5.3 (*i*) STA timing diagram

The opcode fetch of STA instruction has 4-states (clock cycles). Three states have been used to read the opcode from the main memory and the 4th to decode it and set up the subsequent machine cycle.

The action of memory read or write cycles containing 3-states *i.e.*, T_1, T_2, and T_3 are explained as

T_1 : During this period the address and control signals for the memory access are set up.

T_2 : The µP checks up the READY and HOLD control lines. If READY = 0, indicating a slow memory device, the µP enters in the wait state until READY = 1, indicating DMA request, then only the µP floats the data transfer lines and enters into wait until HOLD = 0.

T_3 : In memory read cycles the µP transfers a byte from the data bus to an internal register and in memory write cycle the µP transfers a byte from an internal register to the data bus.

Thus STA instruction requires 4-machine cycles containing 13-states (clock cycles). With a typical clock of 3 MHz (= 330 ns), the STA instruction requires 13*330 ns = 4.29 ms for its execution.

REVIEW QUESTIONS

1. Calculate the execution time required for executing the instructions with the system frequency of

   3 MHz.

   MOV A,B

   MOV C,D

   MOV A,M

   MVI A,05H

   MVI B,05H
2. With relevant diagram, explain the role of timing and control unit in the operation of microprocessors.
3. Define (*a*) Instruction cycle, (*b*) Machine cycle, (*c*) Clock cycle.
4. Draw and explain the timing diagram for the memory read instruction.
5. Draw and explain the timing diagram for the I/O write instruction.