

Instruction Timing and Operation of 8085 Microprocessor

INTRODUCTION

In general, microprocessor performs two operations, viz., instruction fetch operation and instruction execution operation. In 8-bit microprocessor, these operations are performed separately. In pipelined microprocessor, these operations are performed, simultaneously.

The instruction cycle, (also called *fetch-and-execute cycle* and *fetch-decode-execute cycle (rDX)*, can refer to either the time period during which one instruction is fetched from memory and executed when a computer receives a machine language instruction; or the sequence of actions that a MPU performs to execute each machine code instruction in a program. The name is quite literal. The instruction, along with any data to be worked on, must be fetched from the main memory and then executed by the MPU.

In this chapter, we will discuss fetch and execute cycle, T-state, machine cycle and instruction cycle, basic state diagram, timing diagram of opcode fetch cycle, memory read cycle, memory write cycle, IO read cycle, IO write cycle and idle machine cycle. Instruction cycles of all the 8085 instructions are also part of this chapter.

5.1 INSTRUCTION CYCLE

Instruction cycle is defined as the time taken by the processor to execute an instruction. Each processor has different cycles based on different instruction sets. Typically all the processor utilizes the following five stage cycles:

Stage 1: Fetch the instruction from the main memory

The processor presents the value of the program counter (PC) on the address bus. The processor then fetches the instruction from the main memory via the data bus into the memory data register (MDR). The value from the MDR is then placed into the instruction register (IR), a register that holds the instruction so that it can be decoded and executed.

Stage 2: Decode the instruction

The instruction decoder decodes what is to be done in response of an opcode.

Stage 3: Fetch data from the main memory

Read the required data from the main memory to be processed and placed into registers.

Stage 4: Execute the instruction

From the instruction register, the data forming the instruction is decoded by the instruction decoder machine cycle encoder unit of microprocessor. This unit then sends the decoded information to the control unit. The control unit then passes the decoded information as a sequence of control signals to the relevant function units of the MPU to perform the actions required by the instruction such as reading values from registers, passing them to the arithmetic logic unit (ALU) to add them together and writing the result back to a register. A condition signal is sent back to the control unit by the ALU if it is involved.

Stage 5: Store results

The result generated by the operation is stored in the accumulator (in most of the cases). Based on the condition feedback from the ALU, the PC is either incremented to address the next instruction or updated to a different address where the next instruction will be fetched. The cycle is then repeated.

Stages 1 and 2 of the instruction cycle are called the *fetch cycle*. These stages are the same for each instruction. During this operation the microprocessor fetches all the bytes of the current instruction from the program memory. The first byte of instruction is always opcode and is always fetched into instruction register only. Non-pipelined microprocessor executes a special fetch cycle to fetch opcode of the instruction and this cycle is called *opcode fetch cycle*. The length of opcode fetch cycle is larger than the normal fetch cycle in a non-pipelined microprocessor. After decoding opcode in opcode fetch cycle, the microprocessor decides whether to execute next fetch cycle or not. It is one of the read operations.

Stages 3 and 4 of the instruction cycle are part of the execute cycle. These stages will change with each instruction. After instruction fetch operation microprocessor performs instruction operation. In this operation, the microprocessor executes fetched instruction. There are two types of execution operation, internal and external operations. If the operands are available in memory or I/O module then the microprocessor performs external execution. Internal operations are arithmetic, logical, decision making and internal data transfer operations. External executions are data memory read, data memory write, stack read, stack write and I/O write operations.

5.2 BASIC STATE TRANSITIONS

The microprocessor operates sequentially into two states, viz., fetch and execution. In fetch state it fetches all current instruction bytes and enters into execution states. In execution state, it executes a fetched instruction and enters into the fetch state again. In execution state, the microprocessor performs arithmetic, logic, data transfer and decision-making operations.

When the reset signal is activated in fetch state, the microprocessor enters into reset state. In the reset state, it does not perform any internal and external operations and does not accept interrupt and HOLD signals. Hence, reset signal has the highest priority. When the reset signal is activated in execution state then the microprocessor enters into the reset state. When the reset state is disabled, the microprocessor comes out of the state and enters into the fetch state. It does not enter into the execution state. When the HALT flip-flop is encountered in set state, in the fetch cycle, the microprocessor enters into the HALT state. In the HALT state the microprocessor performs any external and internal operations, but it accepts Reset, Interrupt and Hold signals. To get out of the HALT state the halt flip-flop must be reset. When halt flip-flop is reset, the microprocessor enters only into the fetch state not the execute state. When HOLD is activated in the HALT state the microprocessor enters into the HOLD state. After disabling the HOLD signals, in case, it enters into the HALT state again.

When the HOLD signal is activated, in the fetch state, the microprocessor enters into the HOLD state and does not perform external operations, however, it may perform internal execution operations. In this state it accepts only Reset signal, hence HOLD signals has the second priority and Interrupt has the third priority.¹ When HOLD signal is activated in the execution state, microprocessor enters into HOLD state when hold signal is removed.¹

5.3 INTRODUCTION TO MACHINE CYCLES

A program is a set of instructions written in a sequence. The instructions are stored sequentially in memory. The opcode are executed by microprocessor. To execute an instruction the microprocessor first takes the opcode from memory and this is called *opcode fetch*. Then this instruction is decoded by instruction decoder and microprocessor performs the operation specified. If an instruction is a 2-byte instruction then the microprocessor executes this instruction in two steps:

- First it fetches the opcode and decodes it. After decoding the instruction, the microprocessor comes to know about the second byte of the instruction.
- To get the second byte from memory, microprocessor will perform the memory read cycle. Then the microprocessor executes the instruction.

Similarly, if the instruction is of 3-bytes then the microprocessor will execute another operand read (memory read) cycle.

All these operations are performed sequentially with respect to the clock. Microprocessor performs an operation in a specific time period, i.e., specific clock cycles.

5.3.1 T-state

T-state is the time period of a single cycle of the clock frequency as shown in Figure 5.1.

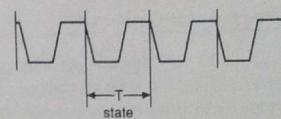


Figure 5.1 T-state.

5.3.2 Machine Cycle

The number of T-state required to access a peripheral is called a *machine cycle*. Access a peripheral means to perform a read or a write operation either from memory or an I/O. A machine cycle may consist of three to six T-states. In general, microprocessor 8085 require only three T-states to access a peripheral. But in case of fetch cycle, microprocessor requires some additional time so as to decode an instruction and, hence, a fetch cycle requires 4 to 6T-states.

5.3.3 Instruction Cycle

Instruction cycle is the total number of machine cycles required to execute a complete instruction. Or the number of machine cycles required to fetch and execute an instruction is called an *instruction cycle*. In 8085 an instruction cycle consists of one to five machine cycles.

Machine cycles are classified on the basis of the operations performed by the microprocessor. There are eight different operations and hence there are eight machine cycles. These are as follows (Table 5.1):

Table 5.1 Machine Cycles

Opcode fetch cycle
Operand fetch cycle
Memory read cycle
Memory write cycle
I/O read cycle
I/O write cycle
Interrupt acknowledge cycle
Idle machine cycle

Opcode fetch cycle

The microprocessor uses this cycle to take the opcode of an instruction, i.e., fetch opcode. In this case the address of the memory location where the opcode is stored, is given by the program counter. This is always the first cycle of any instruction cycle. The opcode is taken from the memory and transferred to the instruction register for decoding and execution. The status of IO/M¹ line is low. The time required to complete this cycle is 4 to 6T-states.

Operand fetch cycle

In 2-and 3-byte instructions taking opcode does not complete the instruction fetching so we require operand fetch cycles. For the 2-byte instructions we require one operand fetch cycle and

for 3-byte instructions we require 2 operand fetch cycles. In this cycle the addresses/data given in an instruction are fetched. But the contents are not transferred to instruction register instead these are stored in temporary registers. The status of IO/M⁻ line is low. The time required to complete the operand fetch cycle is 3T-states.

Memory read cycle

The microprocessor executes these cycles to read data from memory. The address of the memory location is given by instructions. In this case the program counter is not used, instead instruction will give address or will specify where the address is present. The status of IO/M⁻ line is low. The time required to complete the memory read cycle is 3T-states.

Memory write cycle

The microprocessor executes these cycles to write data of memory. The address of memory is given by instructions. The status of IO/M⁻ line is low. The time required to complete the memory write cycle is 3T-states.

IO read

The microprocessor executes these cycles to read data from I/O device instead of memory. The status of IO/M⁻ line is high. The address of the port is given by instruction. The time required to complete the IO read cycle is 3T-states.

IO write

The microprocessor executes these cycles to write or send data to I/O device. The status of IO/M⁻ line is high. The address of the port is given by instruction. The time required to complete the IO write cycle is 3T-states.

Interrupt acknowledge cycle

In response to interrupt request input INTR, the microprocessor executes these cycles to get information from the interrupting device. The time required to complete the interrupt acknowledge cycle is 3T-states.

Idle machine cycle

The microprocessor executes these cycles for internal execution. During this cycle, all data address and control lines are tri-stated, i.e., microprocessor does not perform any external operations. The time required to complete the idle machine cycle is 3T-states. For instance, DAD instruction.

5.4 TIMING DIAGRAM

The graphical representation of the status of the various signals involved during a machine cycle, with respect to time, i.e., clock, is called *timing diagram*. This gives the basic idea of what is happening in the system when the instruction is getting fetched and executed, at what instant which signal is getting activated.

The signals involved during a machine cycle are CLK, A₁₅–A₈, AD₇–AD₀, IO/M⁻, RD⁻, WR⁻ and S₁S₀. The timing diagrams of 8085 instructions are drawn by combining the timing diagrams of the various machine cycles. The status of all the signals associated in a machine cycle are given in a tabular form in Table 5.2. In this table:

0 = Logic "0"

1 = Logic "1"

TS = High impedance

X = Unspecified or depends on the status, data, address values.

Here, in this table, it should be noted that:

1. ALE not generated during 2nd and 3rd machine cycles of DAD instruction.

2. IO/M⁻ = 1 during T₄–T₆ of INTA⁻ machine cycle.

Table 5.3 shows the machine cycle chart.

Table 5.2 8085 Machine State Chart

Machine state	Status and buses				Control			
	S ₁ , S ₀	IO/M ⁻	A ₈ –A ₁₅	AD ₀ –AD ₇	RD ⁻ , WR ⁻	INTA ⁻	ALE ⁻	
T ₁	x	x	x	x	x	1	1	1
T ₂	x	x	x	x	x	x	x	0
T _{WAIT}	x	x	x	x	x	x	x	0
T ₃	x	x	x	x	TS	1	1	0
T ₄	1	0	x	x	TS	1	1	0
T ₅	1	0	x	x	TS	1	1	0
T ₆	1	0	x	x	TS	TS	1	0
T _{RESET}	x	TS	TS	TS	TS	TS	1	0
T _{HALT}	0	TS	TS	TS	TS	TS	1	0
T _{HOLD}	x	TS	TS	TS	TS	TS	1	0

Table 5.3 8085 Machine Cycle Chart

Machine cycle	Status and buses			Control		
	IO/M ⁻	S ₁	S ₀	RD ⁻	WR ⁻	INTA ⁻
Opcode Fetch (OF)	0	1	1	0	1	1
Memory Read (MR)	0	1	0	0	1	1
Memory Write (MW)	0	0	1	1	0	1
I/O Read (IOR)	1	1	0	0	1	0
I/O Write (IOW)	1	0	1	1	0	1
Acknowledge of INTR (INA)	1	1	1	1	1	0
BUS idle (BI): DAD acknowledge of RST, TRAP, HALT	0 1 TS	1 1 0	0 1 0	1 1 TS	1 1 TS	1 1 1

The following section discusses the timing diagram of the various machine cycles.

5.4.1 Timing Diagram of Opcode Fetch Cycle

The 8085 uses this cycle to fetch or to take instruction opcode from memory. In this case, address of the memory is always given by the program counter. There are two types of opcode fetch cycles one using 4T-states to complete the operation and the other using 6T-states to complete the operation. Generally, instruction uses 4T-states but some instructions performing additional operation in opcode fetch requires 6T-states. Figure 5.2 shows a machine cycle having 4T-states and Figure 5.3 shows a machine cycle having 6T-states. The various internal and external operations performed during an opcode fetch cycle are explained in the following steps:

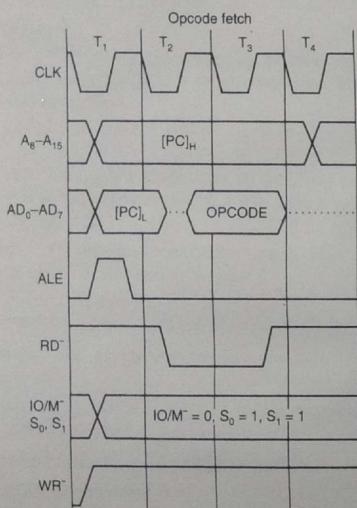


Figure 5.2 Opcode fetch cycle having 4T-states.

Step 1 (T₁-state)

(a) **External operation:** In the T₁-state, 8085 sends 16-bit address on A₈ – A₁₅ and AD₀ – AD₇. The high order byte of program counter is placed on the A₈ – A₁₅ lines and it remains there up to T₃-state. The low order byte of program counter is placed on the AD₀ – AD₇ lines, which remains there only for T₁-state. During this state address latch enable (ALE) gives a positive pulse which represents the contents of AD₀ – AD₇ as address. The ALE signal is

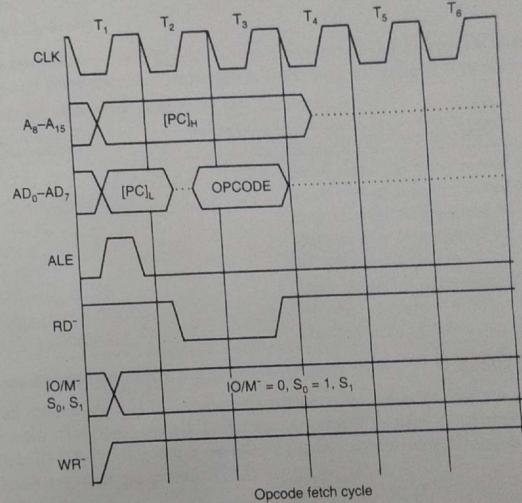


Figure 5.3 Opcode fetch cycle having 6T-states.

used to latch the address from A₀ to A₇. At the end of the T₁-state ALE goes down. During this state no control signals are generated. In the T₁-state, 8085 also sends the status signals IO/M⁺, S₁ and S₀. IO/M⁺ specifies whether the operation is memory related or I/O related. S₁ and S₀ status lines specifies the operation, i.e., IO/M⁺ = 0, S₀ = 1 and S₁ = 1 (opcode fetch).

(b) **Internal operation:** The microprocessor samples its Reset in input line. When it finds Reset low, it completes T₁-state and then enters into reset state. The microprocessor samples its internal HALT flip-flop. When it finds flip-flop set, it completes T₁ and then enters into the HALT state.

Step 2 (T₂-state):

(a) **External operation:** At the end of the T₁-state ALE goes down and, hence, the content of AD₀ – AD₇ disappears. At the beginning of the T₂-state these lines are ready to be used as data lines. The contents from A₀ to A₇ are still available to memory as they are latched in the latch IC. The control signal RD⁺ is made low by the processor which enables the read circuit of addressed memory device. The memory device then sends the content on data bus, i.e., AD₀ – AD₇.

(b) **Internal operation:** During the T₂-state microprocessor performs the following internal operations:

- (i) The microprocessor increments program counter by 1 so as to point towards the next byte of instruction.
- (ii) The microprocessor samples its ready input at the rising edge of T₂. When it finds ready low, it adds "wait" states between T₂ and T₃. Otherwise it executes T₃ after T₂.
- (iii) The microprocessor samples its HOLD inputs. When it finds HOLD high it sets internal HLDA flip-flop and completes the machine cycle.

Step 3 (Wait state)

(a) **External operation:** It is an optional T-state. During this state the microprocessor does not perform any external operation. Hence, T₂-state levels of address, data and control lines are maintained for this state also.

(b) **Internal operation:** The microprocessor samples its ready input at the rising edge of the "wait" state. When it finds Ready low, it adds another wait state. Thus, microprocessor goes on adding states until Ready goes high.

Step 4 (T₃-state)

(a) **External operation:** During this clock cycle on the rising edge the data from memory, i.e., opcode is transferred to instruction register and RD⁻ control signal is made high. This RD⁻ disables the memory device.

(b) **Internal operation:** The microprocessor takes opcode from data bus, i.e., fetch opcode into instruction register and it is provided to decoder.

Step 5 (T₄-state)

(a) **External operation:** The microprocessor performs only internal operation. Data and address buses are tri-stated.

(b) **Internal operation:** The opcode is decoded by the microprocessor and after decoding 8085 knows all the information about

- (i) Whether it should enter T₅ and T₆-states or not?
- (ii) How many bytes of instruction it is?

If the instruction does not require T₅ and T₆-states, it will start or go to the next machine cycle. If it is a multi-byte instruction, the operand fetch cycle are executed to complete the instruction fetching.

During T₄ of T₆ opcode fetch cycle, the microprocessor samples its HOLD input. When it finds hold set it sets internal HLDA flip-flop, otherwise it does not change the state of an internal HLDA flip-flop.

During T₄ of 4T opcode fetch cycle, the microprocessor samples its internal HLDA flip-flop. If HLDA flip-flop is set, it completes T₄, disconnects itself from the system bus, activates HLDA signal and then enters into the HOLD state.

Step (T₅ and T₆-state)

(a) **External operation:** Their T₅ and T₆-states are required just to provide time to complete the decoding and some operation inside 8085. It depends on instruction to instruction. The following instruction require T₅ and T₆-states.

Instruction	Operations performed
CALL	Stack pointer is decremented by 1
Call conditional	Stack pointer is decremented by 1
DCX R _p	Register pair decremented by 1
INX R _p	Register pair incremented by 1
PCHL	HL pair transferred to PC
PUSH R _p	Stack pointer is decremented by 1
SPHL	HL pair transferred to SP
RET conditional	The condition of flags checked

All other instructions except the above instructions require T₁ to T₄.

(b) **Internal operation:** (i) Microprocessor performs the operation on stack pointer, registers pair and checks conditions. During T₆ the microprocessor samples its internal HLDA flip-flop when it finds HLDA flip-flop set; it completes T₆, disconnects itself from system bus, activates HLDA signals and then enters into HOLD state.

5.4.2 Operand Fetch or Memory Read Cycle

The 8085 uses this cycle to fetch the operand of an instruction, if it is a multi-byte instruction, to read the data from memory. These operations require 3T-states, i.e., T₁ to T₃ but if the IO devices are not ready, then the microprocessor inserts some wait T-states (Tw) in between T₂ and T₃ states. Figure 5.4(a) shows the memory read cycle with wait states whereas Figure 5.4(b) shows the normal memory read cycle, i.e. without any wait state. The various steps involved during this cycle are:

Step 1 (T₁-state)

In this T₁-state, 8085 sends appropriate status signals IO/M⁺, S₀ and S₁, i.e., IO/M⁺ = 0, S₀ = 0, S₁ = 1 (memory read). The 16-bit address is transferred on A₈ – A₁₅ and AD₀ – AD₇. The address is dependent on the operation. In operand fetch cycle, address is given by the program counter and then incremented.

In memory read cycle, address is given or indicated by instructions. ALE is generated by 8085 to indicate the availability of address A₀ – A₇ on AD₀ – AD₇. The ALE is used to latch A₀–A₇.

All other operations are similar to step 1 of opcode fetch cycle except that HALT flip-flop is sampled.

Step 2 (T₂-state)

In T₂-state, the content of AD₀ – AD₇ which contains address A₀ – A₇, are removed but the content of A₀ – A₇ is still available for memory in the latch. The lines AD₀ – AD₇ will be

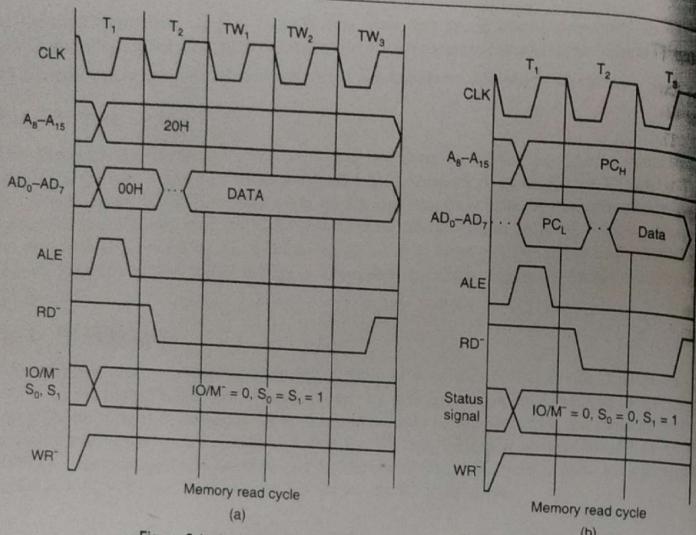


Figure 5.4 Memory read cycle with (a) and without (b) wait state.

used as D₀ - D₇ to transfer data. The control signal RD is made low to enable the memory selected.

Step 3 (Wait state)

Similar to Step 3 of opcode fetch cycle.

Step 4 (T₃-state)

In this T₃-state, data from memory is transferred to 8085. 8085 accepts this data and transfers instruction for which this machine cycle is used. It is generally stored in general purpose registers. If it is operand fetch cycle, then it is loaded in temporary registers for further use. All other operations are similar to Step 4 of opcode fetch cycle except that data is fetched into registers except instruction register and internal HLDA flip-flop is sampled.

In operand fetch cycle the PC contents are incremented by one to point to the next location.

5.4.3 Memory Write Cycle

The 8085 uses this operation to store data in memory location and require 3T-states, i.e., diagram of memory write cycle. If the IO devices are not synchronized with the speed of the

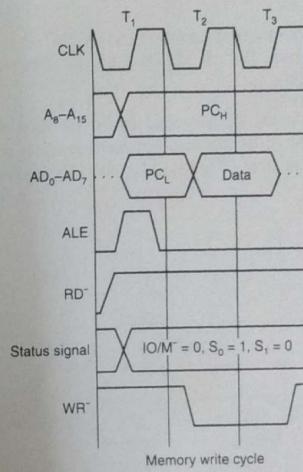


Figure 5.5 Memory write cycle.

microprocessor, i.e., the IO devices are slower, then microprocessor will insert the wait states between T₂ and T₃ states.

Step 1 (T₁-state)

In this T₁-state, 8085 sends the appropriate status signals IO/M*, S₀ and S₁, i.e., IO/M = 0, S₀ = 1, S₁ = 0. The 16-bit address is transferred on A₈ - A₁₅ and AD₀ - AD₇ lines and ALE is generated which indicates availability of address on AD₀ - AD₇. All other operations are same as Step (1) of memory read cycle.

Step 2 (T₂-state)

(a) External operation: In this T₂-state the address on AD₀ - AD₇, i.e., A₀ - A₇ is removed and the data to be stored in memory is transferred on these lines. The control signal WR⁺ is made low, to make memory active and to accept the contents of the data bus and store at selected location.

(b) Internal operation: All operations are similar to memory read cycle.

Step 3 (Wait state)

Similar to memory read cycle.

Step 4 (T_3 -state)

(a) **External operation:** In this T_3 -state, data from data bus is stored at memory location. WR^+ is made high which deactivates memory and microprocessor completes the machine cycle.

(b) **Internal operation:** All internal operations are similar to operand fetch cycle.

5.4.4 IO Read Cycle

The microprocessor executes these cycles to read data from I/O device instead of memory. The address of the port is given by instructions. The time required to complete the IO read cycle is 3T-states. In the IO read machine cycle all the operations are same as that of memory read cycle with the difference that the IO have 8-bit port address instead of 16-bit address. This 8-bit port address is carried by the multiplexed $AD_0 - AD_7$ lines and the higher order bus $A_8 - A_{15}$ are XXH. The IO/M^+ signal is high in IO related operations. Figure 5.6 shows the IO read cycle.

5.4.5 IO Write Cycle

The microprocessor executes these cycles to write or send data to I/O device. The status of the IO/M^+ line is high. The address of port is given by instruction. The time required to complete the IO write cycle is 3T-states. In IO read machine cycle all the operations are same as that of memory write cycle with the difference that the IO have 8-bit port address instead of 16-bit address. This 8-bit port address is carried by the multiplexed $AD_0 - AD_7$ lines and the higher order bus $A_8 - A_{15}$ are XXH. Figure 5.6 shows the IO write cycle.

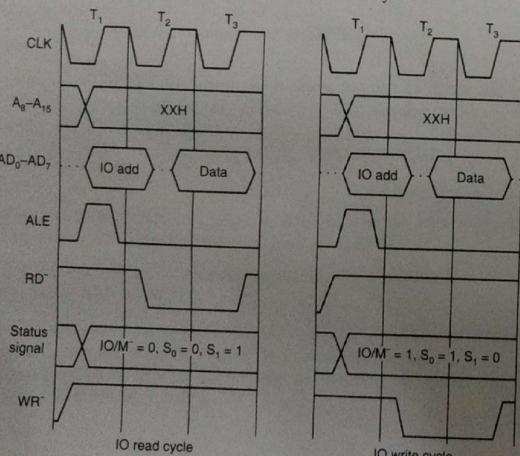


Figure 5.6 IO read cycle and IO write machine cycle.

5.5 TIMING DIAGRAM OF 8085 INSTRUCTIONS**MOV A, C**

This is a one-byte instruction. This instruction does not require any data from memory and does not store any data to memory. So, only opcode fetch is required to fetch and execute the instruction.

Opcode fetch: For this cycle the higher order and the lower order address is given by the program counter and then program counter is incremented by one to point to next instruction. This instruction does not require T_5 and T_6 -states but T_1 to T_4 -states only. Figure 5.7 shows the timing diagram of MOV A, C.

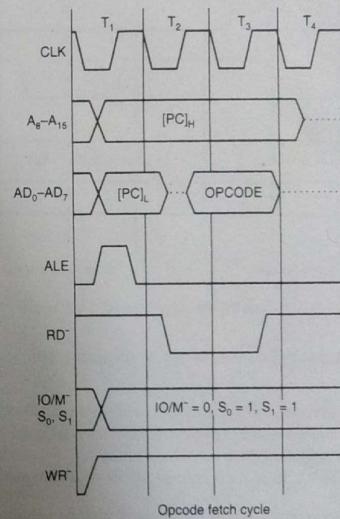


Figure 5.7 Timing diagram MOV A, C.

ADD B: This is a one-byte instruction. This instruction does not require any data from memory and does not store any data to memory. So only opcode fetch is required to fetch and execute the instruction.

The timing diagram of SUB R, ADC R, SBB R, ORA R, XRA R, ANA R, CMA, CMP R and ADD R are same as that of ADD B. Figure 5.8 shows the timing diagram of ADD B.

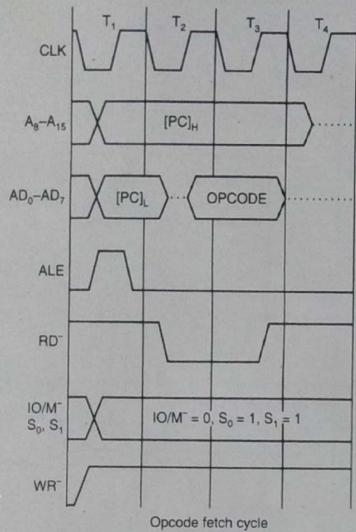


Figure 5.8 Timing diagram of ADD B.

ADD M: This instruction adds the contents of the memory location M whose address is in register pair HL. The HL pair gives the address of memory. Two machine cycles are required to execute this instruction. They are as follows:

1. Opcode fetch,
2. Memory read

The instructions like SUB M, ANA M, ORA M, XRA M have the same timing diagram of that of ADD M. Figure 5.9 shows the timing diagram of ADD M.

INX H: This is a one-byte instruction, does not require data from memory or store data into memory. So only opcode fetch is required.

Opcode fetch: For this cycle the higher order and lower order address is given by the program counter and then the program counter is incremented by one to point to next instruction. The opcode fetch of INX need more time for decoding the opcode and internal operations so T₅- and T₆-states are required.

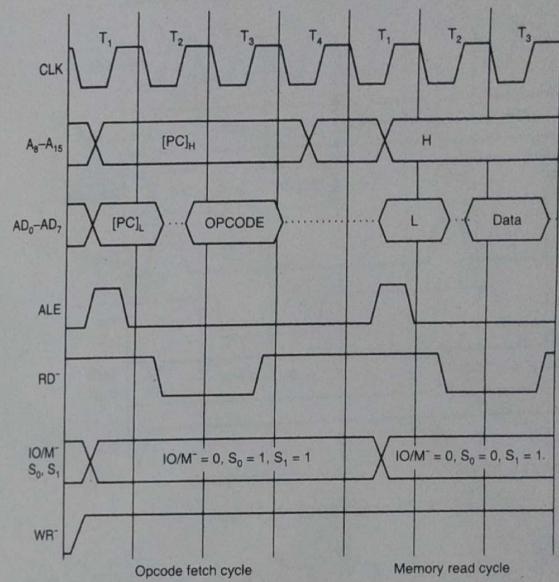


Figure 5.9 Timing diagram of ADD M.

DCX R_p also has the same timing diagram as that of INX H. Figure 5.10 shows the timing diagram of INX H.

MVI A, 50 H: This is a two-byte instruction so it requires two machine cycles to fetch the instruction: opcode fetch and operand fetch machine cycles. To execute this instruction, no data from the memory is required so only 2 machine cycles are sufficient. This instruction does not require T₅- and T₆-states of opcode fetch so the opcode fetch is of T₁- to T₄-states.

Opcode fetch: For this cycle the address is given by the program counter and then the PC is incremented by one. Now it points where the operand is stored.

Operand fetch cycle: The program counter contents are used to take the operand and again increment by one to point to the next instruction after MVI.

The instructions such as SBI data, SUI data, ADI data, ACI data, ANI data, ORI data, XRI data and CPI data have the same timing diagram as that of MVI A, 50H. Figure 5.11 shows the timing diagram of MVI A, 50.

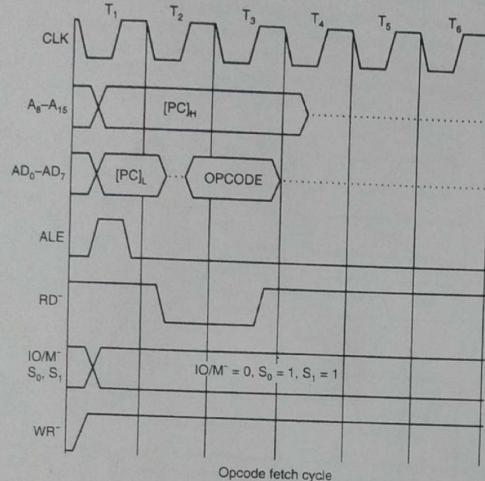


Figure 5.10 Timing diagram of INX H.

MVI M, 25H

This is a two-byte instruction so it requires 2 machine cycles to fetch the instruction: opcode fetch and operand fetch machine cycles. For execution of this instruction it is required to write data to memory so one more machine cycle is required. Therefore, in all 3 machine cycles are required. This instruction does not require T_5 - and T_6 -states of opcode fetch so the opcode fetch is of T_1 - to T_4 -states. For the last machine cycle, i.e., the memory write cycle, the address is given by HL pair and PC remains unchanged. Figure 5.12 shows the timing diagram of MVI M, 25H.

LXI H, 2500H

This is a three-byte instruction. It requires 3 machine cycles to fetch the instruction:

1. Opcode fetch
2. Operand read
3. Operand read

For execution of this instruction it does not require any data transfer to or from memory so 3 machine cycles will be sufficient to execute this instruction. The opcode fetch does not require T_5 - and T_6 -states so opcode fetch is of T_1 - to T_4 -states. For opcode fetch the address

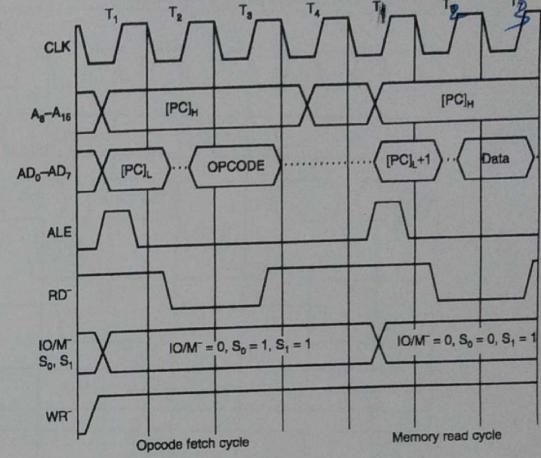


Figure 5.11 Timing diagram MVI A, 50H.

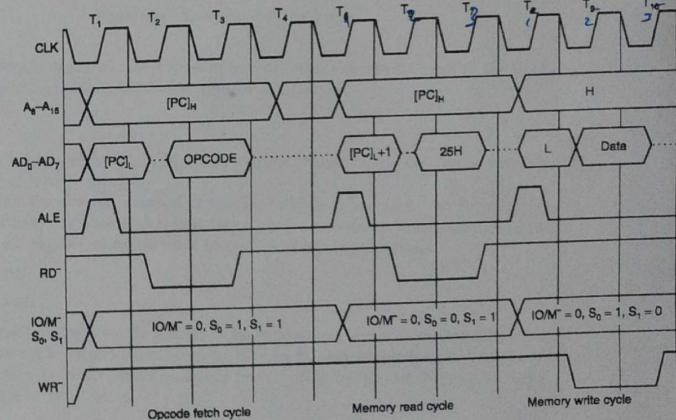


Figure 5.12 Timing diagram MVI M, 25H.

* Please Count T from T_1 in every new Machine cycle.

is given by PC and then it is incremented by one to point to lower order data of instruction stored after opcode. The lower order operand is fetched from the next memory location and PC is again incremented by one so now it points to the higher order operand. The higher order operand is fetched using PC and PC is again incremented by one to point to the next instruction after LXI H, 2500H. Figure 5.13 shows the timing diagram of LXI H, 2500H.

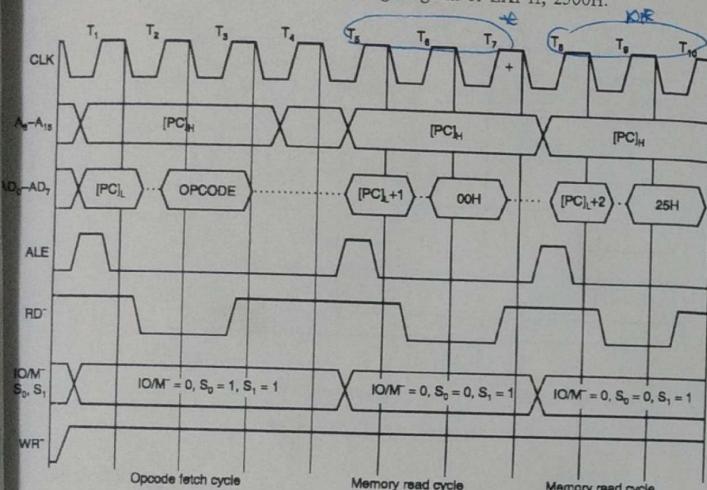


Figure 5.13 Timing diagram of LXI H, 2500H.

MOV A, M

This is a one-byte instruction so it requires one machine cycle to fetch the opcode. This instruction transfers the contents of the memory to the specified register. The HL register pair is the address of memory. To take the data from the memory, memory read operation is required. To execute this instruction 2 machine cycles are required: opcode fetch and memory read. For opcode fetch, address is given by PC; PC is then incremented by one. For memory read cycle the address is given by HL and PC remains unchanged. The opcode fetch does not require T_5 and T_6 -states so opcode fetch is of T_1 - to T_4 -states. Figure 5.14 shows the timing diagram of MOV A, M.

MOV M, A

This instruction is same as MOV A, M with one change. In MOV A, M instruction the address is given by HL whereas in LDAX B the address is given by BC register pair. The timing diagram of LDAX B will remain same as that of MOVA, M with only one change in memory

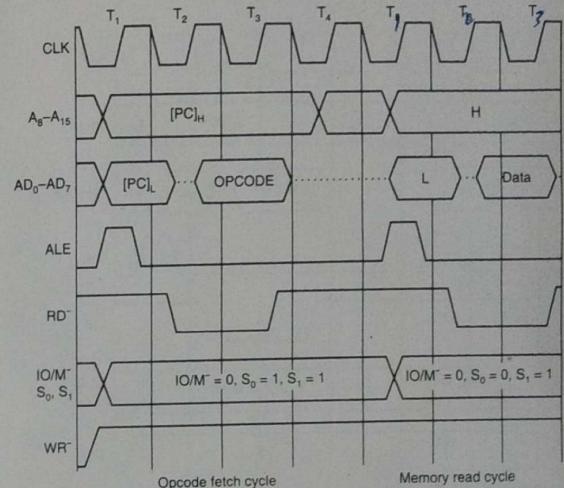


Figure 5.14 Timing diagram of MOV A, M.

read cycle. In memory read cycle the address of memory is provided by BC register pair instead of HL pair. Figure 5.15 shows the timing diagram of LDAX B.

MOV M, A

This instruction is same as MOV A, M but one major change is that the data is written to memory instead of reading the data.

The instruction transfers the contents of specified register to memory. The HL pair gives the address of memory. To transfer data to memory it requires memory write operation, so two machine cycles are required to execute this instruction. They are as follows:

1. Opcode fetch
2. Memory write.

For opcode fetch the PC gives the address of the memory and then it is incremented by one. For memory write cycle the address is given by HL and PC remains unchanged. The opcode fetch does not require T_5 and T_6 -states so opcode fetch is of T_1 - to T_4 -states. Figure 5.16 shows the timing diagram of MOV M, A.

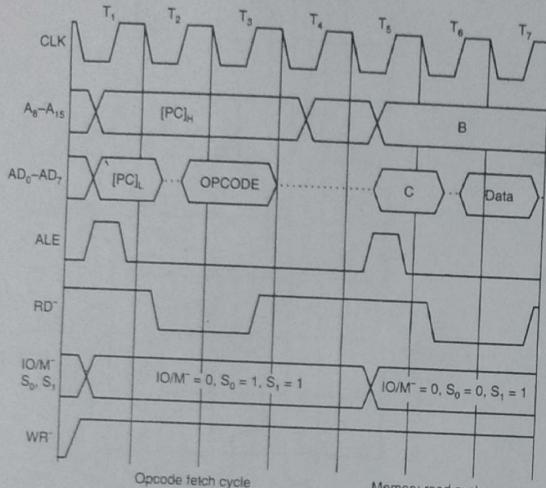


Figure 5.15 Timing diagram of LDAX B.

LHLD 2500 H

This is a three-byte instruction so it requires 3 machine cycles to fetch the instruction. These cycles are opcode fetch, operand read and operand read. The LHLD instruction loads the contents of memory 2500H and 2501H into L and H registers. To load these memory contents into HL 2 more memory read cycles are required. So, in all, 5 machine cycles are required.

Opcode fetch: To take opcode of LHLD instruction, address is given by PC and then PC is incremented by 1. T₅ and T₆-states are not required so opcode fetch is of T₁- to T₄-states only.

Operand fetch: This cycle is performed to take lower order address. PC gives address for this and then PC is incremented by 1. The lower address is stored in temporary register W.

Operand fetch: This cycle is performed to take higher order address. PC gives address for this and then PC is incremented by 1. The higher order address is stored in temporary register Z.

Memory read: This cycle is performed to take data from memory. This data will be stored in L register. Address for this is given by instruction, i.e., above 2 memory reads has read the address with instruction and stored in temporary registers W and Z. The WZ will give the address; address will be incremented by 1.

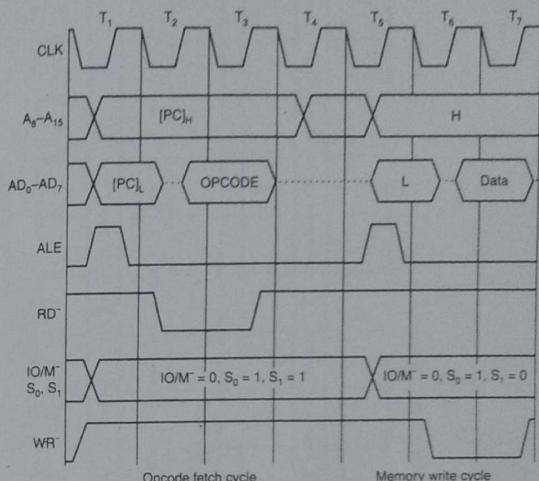


Figure 5.16 Timing diagram of MOV M, A.

Memory read: This cycle is performed to take data from memory, this data will be stored in H register. Address for this is given by instruction, in the above state the address is incremented by 1 so that the address is used to take data.

Figure 5.17 shows the timing diagram of LHLD 2500H.

SHLD 2500H: This instruction is similar to LHLD except that the data is stored in memory from H and L registers. So the last two machine cycles will be memory write and data will be given by H and L registers.

Total 5 machine cycles are used:

1. Opcode fetch
 2. Operand fetch
 3. Operand fetch
 4. Memory write
 5. Memory write.
- or
m-r
m-r
m-w
m-w*

Figure 5.18 shows the timing diagram of SHLD 2500H.

DAD B

This is a one-byte instruction. So to fetch the instruction only machine cycle is required, i.e., opcode fetch. The instruction adds the contents of HL with the contents of BC and the result

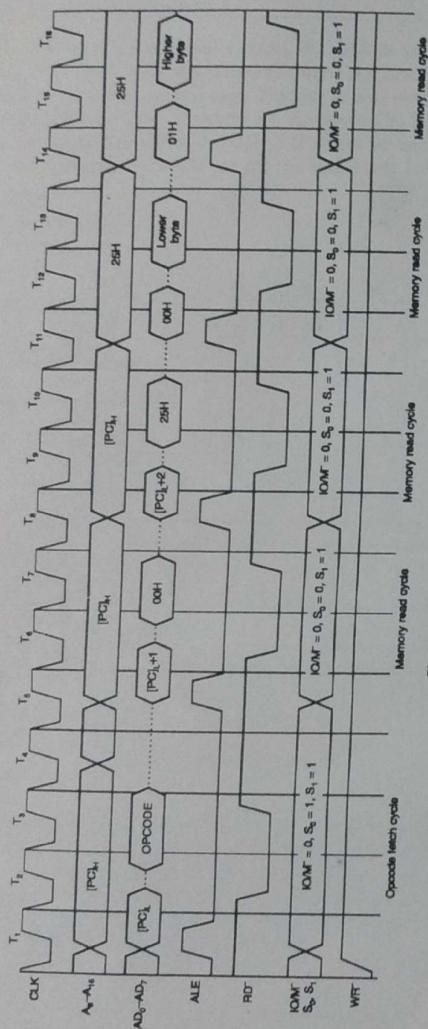


Figure 5.17 Timing diagram of LHLD 2500H.

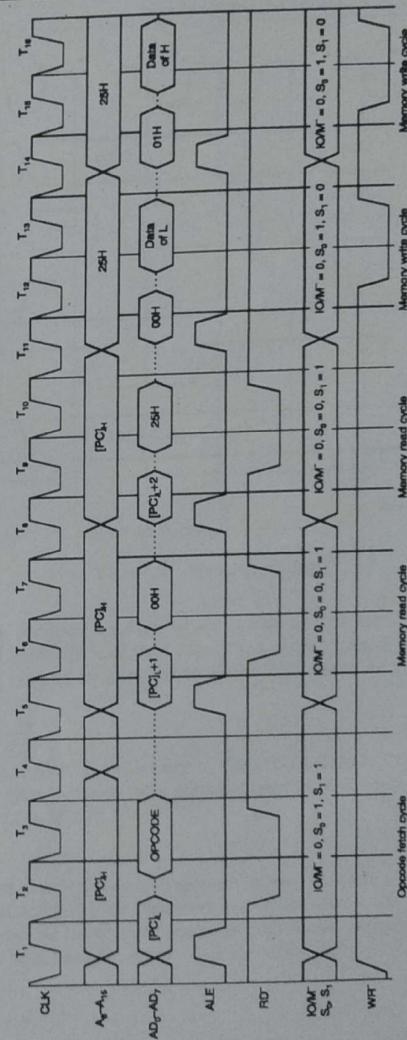


Figure 5.18 Timing diagram of SHLD 2500H.

is placed in HL. The operation is 16-bits addition. But as 8085 is 8-bit processor hence it requires additional time to complete the execution. During that time 8085 is busy in internal execution so it executes bus idle machine cycle. Fetching of next instruction is stopped by not giving control signal RD⁺ and ALE. PC is also not incremented by 1. It is similar to memory read without control signal RD⁺ and ALE. Figure 5.19 shows the timing diagram of DAD B.

Three machine cycles are required to execute this instruction. They are as follows:

1. Opcode fetch
2. Bus idle machine cycle
3. Bus idle machine cycle.

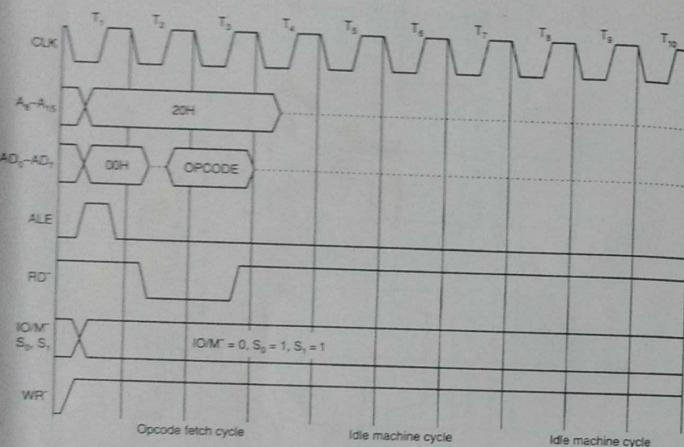


Figure 5.19 Timing diagram of DAD B.

HLT

This is a one-byte instruction, so only opcode fetch is required to take the instruction. In response to the execution of HLT instruction microprocessor sets its HALT flip-flop. Meanwhile the PC contents are incremented by 1 and microprocessor starts the next machine cycle. The flip-flop is checked in T₂-state so that all buses will be tri-stated and the microprocessor will go on adding T HALT states. To come out of this HALT state a reset is required. Figure 5.20 shows the timing diagram of HLT.

STA 2500H

This instruction consists of 4 machine cycles. The first machine cycle is opcode fetch and the second and third machine cycles are memory read cycles. Memory write cycle is the last machine cycle.

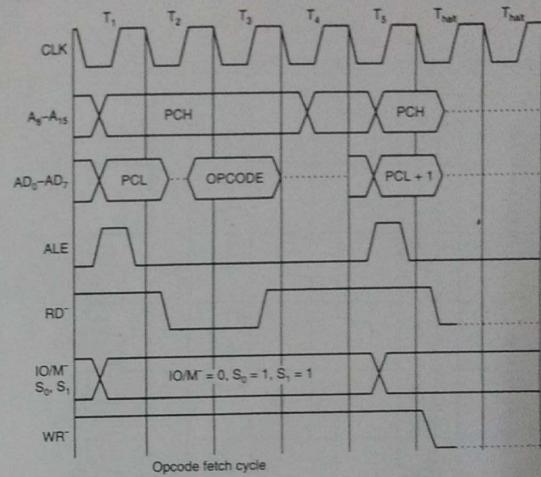


Figure 5.20 Timing diagram of HLT.

In opcode fetch microprocessor fetches the opcode of STA and in the next two machine cycles first it will read 00H and then 25H. Then microprocessor will write the data of STA accumulator in the memory location 2500H. Figure 5.21 shows the timing diagram of STA 2500H.

LDA 2500H

The timing diagram of LDA 2500H is same as that of STA 2500H. The only difference is that the fourth machine cycle is memory read instead of memory write cycle. Figure 5.22 shows the timing diagram of LDA 2500H.

INR M

This instruction increments the content of memory location M whose address is specified by HL pair. In this instruction first the microprocessor fetches the opcode and then gets the data from the memory and stores the incremented data/result back in the same memory location. So, this instruction requires three machine cycles. They are as follows:

- (i) Opcode fetch
- (ii) Memory read.
- (iii) Memory write.

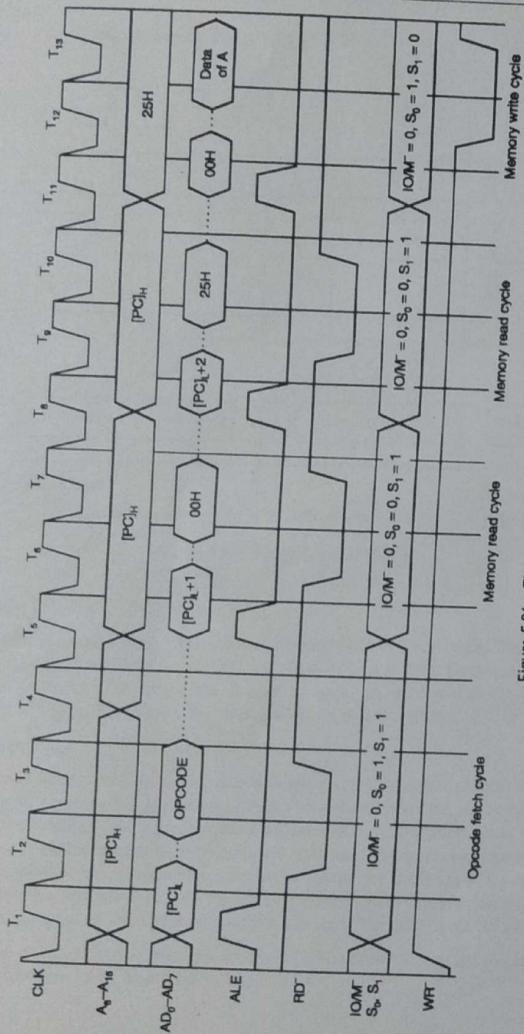


Figure 5.21 Timing diagram of STA 2500H.

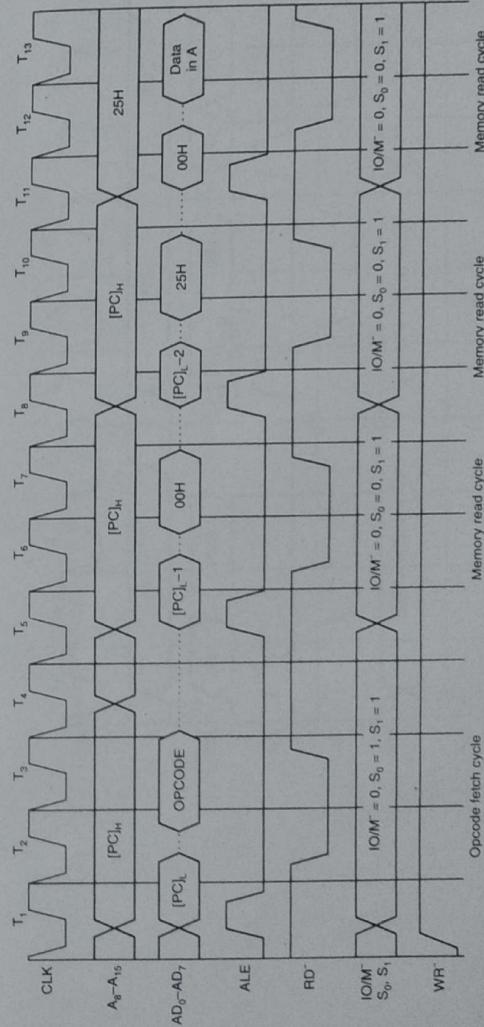


Figure 5.22 Timing diagram of LDA 2500H.

In the memory read and memory write cycle the address of the memory is given by HL register pair.

Figure 5.23 shows the timing diagram of INR M.

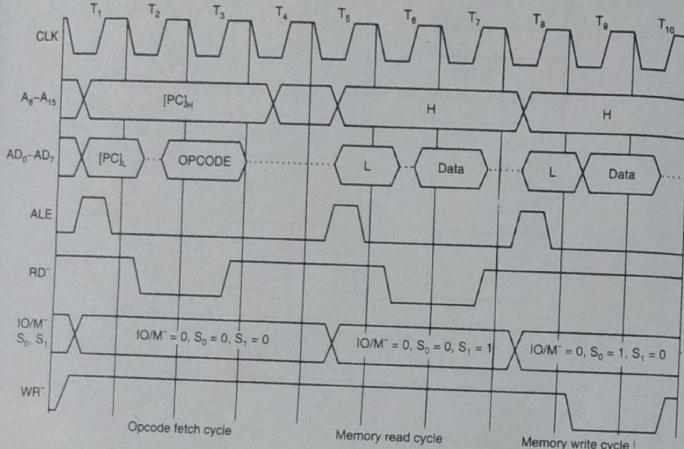


Figure 5.23 Timing diagram of INR M.

JMP 2100H

This is a control transfer instruction and this instruction is of three bytes. The first byte is the opcode of JMP and the second byte is 00H and the third byte is 21H. So this instruction requires three machine cycles. The first cycle is opcode fetch, the second and the third cycles are operand fetch. Figure 5.24 shows the timing diagram of JM 2100H.

Conditional jump

This is a conditional control transfer instruction and this instruction is of three bytes. When the microprocessor executes this instruction, it has to decide whether the condition is true or not. To check the condition, the microprocessor requires some extra (more than 4T) time. After opcode fetch cycle the microprocessor continues with the next cycle, i.e., operand fetch cycle. And in the meantime the microprocessor internally checks the condition. If the condition is true the microprocessor will continue with the remaining third cycle, i.e., operand fetch cycle. If the condition is not true the microprocessor will not execute the next cycle. In short:

1. If the condition is true then the process of execution will remain the same as that of unconditional jump, as shown in Figure 5.24.

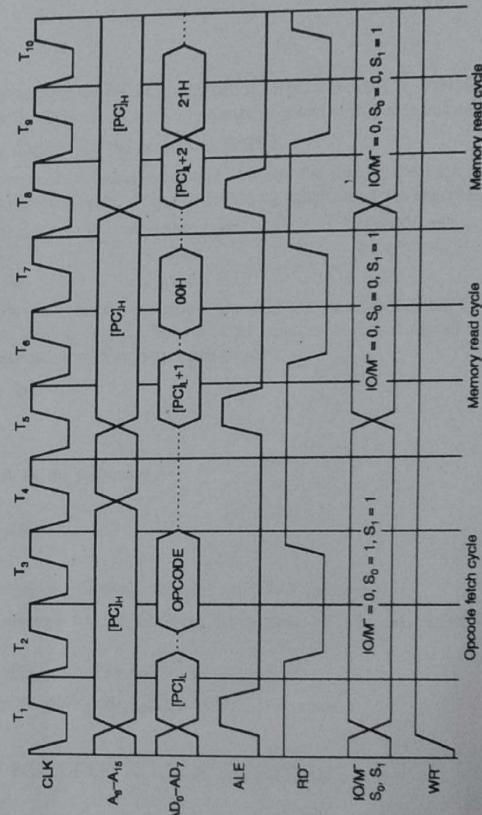


Figure 5.24 Timing diagram of JMP 2100H.

2. If the condition is not true the microprocessor will execute only the first two machine cycles.

Figure 5.25 shows the timing diagram of JM 2100H.

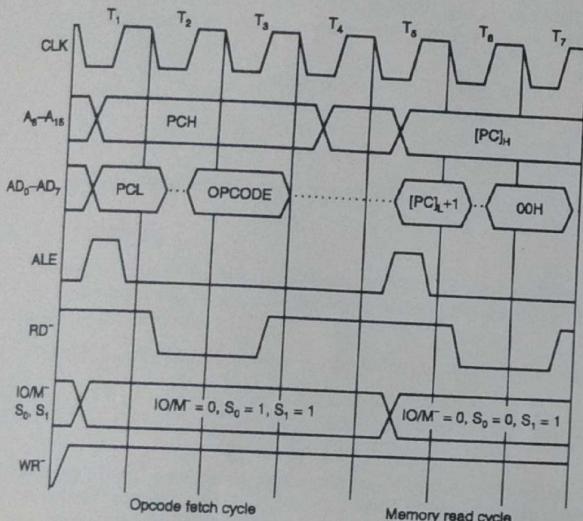


Figure 5.25 JNC 2100H if condition is not true.

EXERCISES

- 1. Draw and explain the timing diagram of memory read cycle.
- 2. Draw and explain the timing diagram of memory write cycle.
- 3. Draw and explain the timing diagram of opcode fetch cycle.
- 4. Explain how STA 2800H is executed. Draw its timing diagram.
- 5. Draw the timing diagram of INRM.
- 6. Discuss instruction cycle, machine cycle and T-state in conjunction with 8085 microprocessor.
- 7. The instruction MOV B, M copies the contents of the memory location in register B. It is a 1-byte instruction with two machine cycles and seven T-states. Identify the second machine cycle and its control signals.

- 5.8 Calculate the time required to execute the following instructions if the system clock frequency is 700 kHz.

- MOV C, B
- JMP 205
- ADI, F2H

- 5.9 Draw the timing diagram for the execution of the instruction MVI A, 40H. If the clock frequency is 5 MHz, how much time is required to execute this instruction?

- 5.10 Draw the timing diagram for the instruction SBBM.

- 5.11 Illustrate the contents of the address bus and data bus and timing of the control signals in 8085 microprocessor system, when the following instructions are executed?

Memory address	Machine code	Mnemonics
2000	C6	ADI, F2 H
2001	F2	

- 5.12 If the program counter is always one count ahead of the memory location from which the machine code is being fetched, how does the microprocessor change the sequence of the program execution with a jump instruction?

- 5.13 Draw the timing diagrams of

- ANA B
- MOV M, A

- 5.14 Write a short note on the following:

- Fetch cycle
- Instruction cycle
- Execute cycle

- 5.15 Explain the functions and timings associated with RST instruction.

- 5.16 Draw the timing diagram for the following instructions: MVI D, data, LDA 8000 and RST 7.

- 5.17 Draw the timing diagram of I/O read cycle in 8085 and explain it.

- 5.18 Draw and explain the timing diagram for XCHG instruction.

MULTIPLE CHOICE QUESTIONS

- Number of t-states required to access a peripheral is called a _____.
 - Machine cycle
 - Instruction cycle
 - Both (a) and (b)
 - Neither (a) nor (b)
- Maximum machine cycles that can be there in instruction cycle is
 - 4
 - 5
 - 6
 - 7

7.1 USE OF STACK FOR PROGRAMMER

The programmer can use instruction to store data in stack. 16-bit data is stored by a single instruction. The instruction to store data is PUSH and to take back the contents is POP, with these PUSH and POP instructions a register pair is always specified. The examples are as follows:

PUSH B	POP B
PUSH D	POP D
PUSH H	POP H
PUSH PSW	POP PSW

B, D, H, PSW specifies register pairs BC, DE, HL, A and flags.

7.1.1 PUSH R_p

Push the contents of the specified register pair onto the stack memory. The storing pattern used by PUSH instruction is that first the higher order register contents are stored then the lower order register contents.

EXAMPLE 7.1 Write comments on the following set of instructions:

```
LXI SP, 2500H
LXI D, F3C0H
PUSH D
```

Solution In this example the first instruction is used to initialize the stack pointer (SP). With the second instruction F3C0H immediate data is loaded in the register pair DE. By third instruction the contents of the DE register pair are loaded on the top of the stack memory. The contents of the stack memory are shown in Figure 7.1.

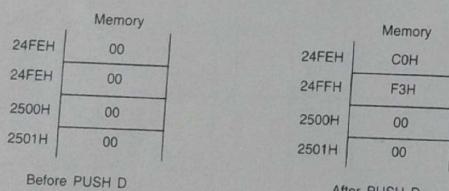


Figure 7.1 Stack operation.

The execution of PUSH D instruction will be as follows:

1. The contents of SP register are decremented by one.
2. The contents of higher order register, i.e., D register are stored at memory location specified by SP register.

3. The contents of SP register are again decremented by one.

4. The contents of lower order register, i.e., E register are stored at memory location specified by SP register. Steps 1-4 can be represented as follows:

Similarly, other register pairs such as DE, HL and PSW can be stored by using PUSH D, PUSH H, PUSH PSW instructions.

EXAMPLE 7.2 Write comments on the following set of instructions:

```
LXI SP, 2500H
LXI B, 2050H
LXI D, 9285H
:
PUSH D
PUSH B
```

Solution In this example, the stack pointer is initialized at 2500H, so SP = 2500H. The BC register pair is initialized with 2050H data. So B = 20H and L = 50H. The DE pair is initialized with 9285H data so D = 92 and E = 85.

When PUSH D instruction is executed, the SP is decremented by 1 and the contents of the D register are stored at the memory location specified by SP register contents. The stack pointer register is again decremented by 1 and the contents of E register are stored at memory location specified by SP register contents. So D register contents are stored at address 24FFH and E register contents are stored at address 24FEH. After the execution of this instruction the stack pointer contents will be 24FEH.

When PUSH H instructions are executed, the SP is decremented by 1 and H register contents are stored at the memory location, specified by SP register contents. The SP again is decremented by 1 and L register contents are stored at memory location specified by SP register contents. So at address 24FDH, H register contents are stored and at addresses 24FCH, L register contents are stored. The stack pointer contents will be 24FCH.

7.1.2 Timing Diagram of the PUSH R_p

PUSH is a one byte instruction so microprocessor requires one machine cycle to fetch the PUSH instruction, i.e., opcode fetch. The PUSH H instruction stores HL register pair contents onto the stack memory. To perform this storing operation, 8085 requires two more MEMW machine cycles. So, in all three machine cycles are required. The timing diagram of PUSH H is given in Figure 7.2.

1. Opcode fetch (M₁): Opcode fetch is used to take opcode of PUSH H instruction, address is given by the program counter (PC) and then PC is incremented by 1. The opcode fetch operation of PUSH instruction requires 6T states. During T₄, T₅ and T₆ states, the microprocessor performs internal operation of decoding PUSH and decrementing stack pointer register contents by 1.

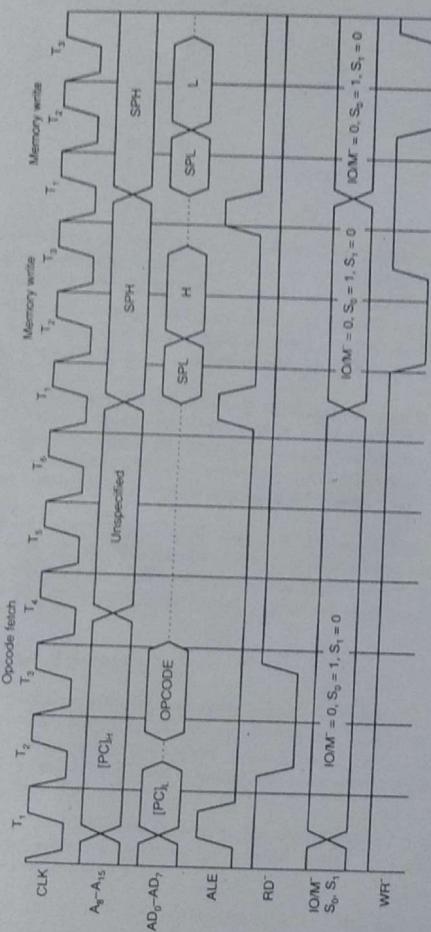


Figure 7.2 Timing diagram of the BRISI H

2. Memory write (M₂): This machine cycle is used to store data from higher order register onto stack. Address for this is given by stack pointer register. The contents are stored and the stack pointer is again decremented by 1.

3. Memory write (M_3): This cycle is used to store data from lower order register onto the stack. Address for this is given by the stack pointer register. The contents are stored and the microprocessor will start opcode fetch machine cycle for the next instruction.

7.1.3 POP R_p

Pop off the contents of the specified register pair from the top of the stack memory. The reading of the contents of the stack memory in the specified register pair are summarized as follows:

1. The contents of stack are read from memory location whose address is specified by SP register, these contents are transferred to lower order register of register pairs.
 2. The contents of SP register are incremented by one.
 3. The contents of stack are read from memory location whose address is specified by SP register, these contents are transferred to higher order register of register pairs.
 4. The contents of SP register are again incremented by one.

Similarly, other register pairs can be popped by using POP D, POP H and POP PSW instructions.

EXAMPLE 7.3 See Figure 7.3. If the present content of the SP is 24FEH. Find out the contents of the register pair BC after the following set of instructions:

LXI SP 2500H		
POP B		
	Memory	
24FEH	00	Memory
24FFH	20	24FEH 00
2500H	00	24FFH 20
2501H	00	2500H 00
		2501H 0

Figure 7.3 Stack operation corresponding to Example 7.3.

Solution Memory and register contents

Before execution of instruction

$$B = 00, C = 00$$

SP = 24FEH

Memory and register contents

Memory and register contents
After execution of instruction

$B = 20H$, $C = 00H$

SP = 2500H

EXAMPLE 7.4 Describe the operation performed by the following set of instructions:

```

LXI      SP, 2500H
LXI      H, 5535H
LXI      D, FFDDH
:
:
PUSH D
LXI      D, 0000H
:
POP D

```

Solution In this example the stack pointer is initialized at 2500H, so the contents of SP = 2500H. The HL pair is initialized with 5535H. So the contents of H = 55H and L = 25H. The DE pair is initialized with FFDDH data so D = FFH and E = DDH.

When a PUSH D instruction is executed, the contents of D and E registers are pushed on to top of the stack. The contents of the D register are stored at 24FEH and the contents of the E register are stored at C6FD. E register contents are stored and SP will be decremented by 2, so SP = 24FEH.

When LXI D, 0000H instruction is executed, D and E register contents are changed.

When a POP D instruction is executed, it takes back or reads contents of the stack and loads in DE register pair. The memory location whose address is specified by the contents of the SP register is transferred to E register. SP is incremented by 1 and the contents of memory location specified by SP are transferred to D register and SP is again incremented by 1, therefore, SP = 2500H.

The contents stored by PUSH instruction are taken back by using the POP instruction. After execution of one PUSH and one POP instruction, the stack pointer again comes back to the initial condition. So while working with stack, every PUSH must have a POP instruction and structure.

7.1.4 Timing Diagram of the POP H

POP H is a one-byte instruction so the microprocessor requires one machine cycle to fetch the opcode. The POP H instruction reads top of the stack contents and loads in HL register pair, first lower order register then higher order register. To perform this, 8085 requires two more memory read machine cycles. So, in all, three machine cycles are required. The timing diagram of POP H is given in Figure 7.4.

1. **Opcode fetch (M_1):** The M_1 machine cycle is used to take opcode of POP H instruction from the memory whose address is given by the program counter (PC) and then PC is incremented by 1. It is a 4T machine cycle.

2. **Memory read (M_2):** This cycle is used to read data from top of the stack and transfer it to the lower order register (L). The address of the top of the stack is given by the stack pointer register. The contents are read and the stack pointer is again incremented by 1.

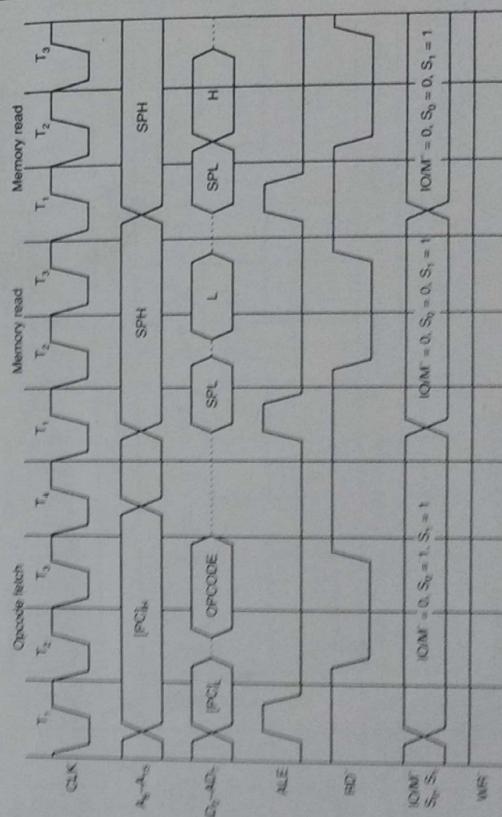


Figure 7.4 Timing diagram of the POP H

The main advantage of this SPHL instruction is that we can change the SP register contents by taking the decisions.

XTHL: Exchange the contents of the top of the stack with the contents of the HL register pair. Top of the stack term specifies the contents of the memory location which is addressed by SP register and the next memory location whose address is $(SP + 1)$.

$$L \leftrightarrow (SP)$$

$$H \leftrightarrow (SP + 1)$$

The following steps are involved during the execution of the XTHL instruction:

1. The content of memory location specified by SP register is transferred to the temporary register Z.
2. The content of next memory location specified by $SP + 1$ is transferred to the temporary register W.
3. The content of H register is transferred to the memory location specified by $(SP + 1)$.
4. The content of L register is transferred to the memory specified given by SP.
5. The data from Z temporary register is transferred to L register and data from W register is transferred to H register.

The operation of the XTHL is explained in Figure 7.5:

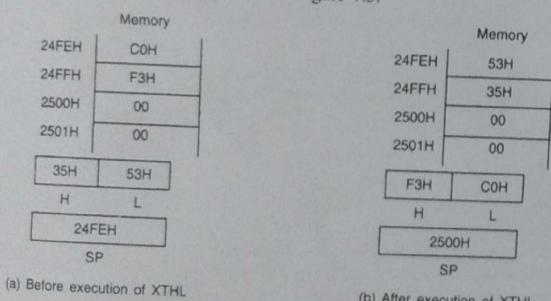


Figure 7.5 Pictorial representation of XTHL.

7.2.2 Timing Diagram of XTHL

This instruction interchanges the contents of HL pair with the contents of top of the stack. This is a one-byte instruction so the microprocessor requires one machine cycle for opcode fetch. To perform this instruction, the microprocessor has to first read the top of the stack contents into the temporary registers W and Z and then it stores the contents of H and L at the top of the stack and then transfers the W and Z contents to H and L registers. The timing diagram of XTHL is as shown in Figure 7.6.

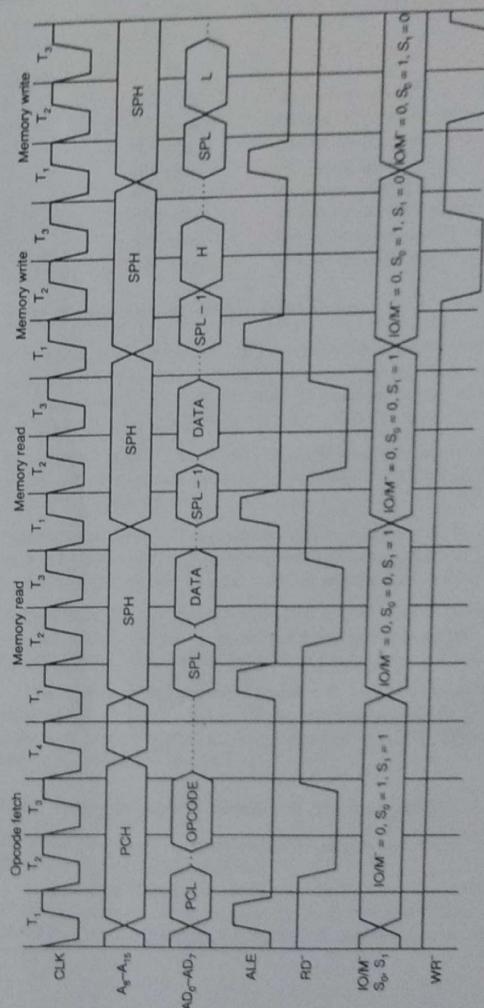


Figure 7.6 Timing diagram of XTHL.

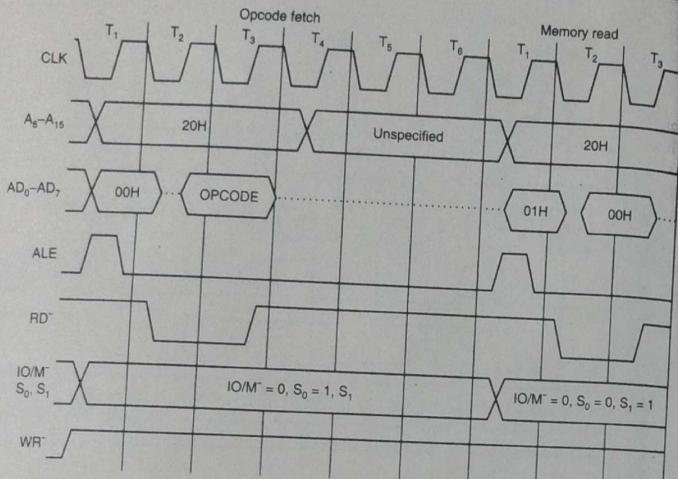


Figure 7.10 2000H CC2500H if condition is not true.

During the M₂ machine cycle, the microprocessor internally checks the condition of the flag register and finds out that the condition is not true and, hence, microprocessor does not execute the next machine cycles.

7.4.3 RET

It is a 1-byte instruction used to transfer the program control from the subroutine program to the main program. To execute this instruction the microprocessor takes back the stored contents of PC from stack top, i.e., the contents of the stack top pointed by SP are read and stored in the temporary register Z, SP is incremented by 1 and the contents of stack top pointed by SP are read and stored in the temporary register W, SP is again incremented by 1. W and Z register's contents are then transferred on address lines and are incremented by 1 and then loaded in the program counter. So the next instruction executed will be from the main program and will be the next instruction after CALL.

Return from subroutine conditionally

The program sequence is transferred from the subroutine to the calling program based on the specified flag of the PSW as described below. The two bytes from the top of the stack are copied into the program counter and program execution begins at the new address. All these

instructions have the same addressing mode, same number of bytes, and same flag conditions as that of unconditional RET. But the number of machine cycles will be different when the condition is true (3 machine cycles and 10T states) and when the condition is not true (1 machine cycle and 6T states).

Example RZ

RC none	Return on carry CY = 1	D8H
RNC none	Return on no carry CY = 0	D0H
RP none	Return on positive S = 0	C8H
RM none	Return on minus S = 1	C0H
RZ none	Return on zero Z = 1	F0H
RNZ none	Return on no zero Z = 0	F8H
RPE none	Return on parity even P = 1	E0H
RPO none	Return on parity odd P = 0	E8H

7.4.4 Timing Diagram of RET Instructions

RET is a one-byte instruction. So microprocessor requires only one machine cycle, i.e., fetch cycle. The RET instruction reads PC contents from the stack and transfers the control back to the main program. To perform this, the microprocessor requires to memory read (stack) machine cycles. So, in all, three machine cycles are required.

Opcode fetch (M₁): This machine cycle is used to fetch opcode of RET instruction from the memory whose address is specified by PC and then PC is incremented by 1. The opcode fetch cycle of RET is of 4T states.

Memory read (M₂): This machine cycle is used to read the lower order PC contents from stack and transfer that operand to internal temporary register (Z). The address for this is given by the stack pointer, the contents are read and the stack pointer is incremented by 1.

Memory read (M₃): This machine cycle is used to read the higher order PC contents from the stack and transfer that to the internal temporary register (W). The address for this is given by stack pointer. The contents are read and stack pointer is incremented by 1.

The contents of the temporary registers W and Z are transferred on address lines and are incremented by 1 and loaded to PC. So the program control is transferred back to the main program at address C008, i.e., next instruction CALL C 200. The timing diagram of RET instruction is shown in Figure 7.11.

7.4.5 Timing Diagram of Conditional RET Instructions

The timing diagram of the conditional RET is shown in Figure 7.12 (if condition is not true) and Figure 7.13 (if condition is true).

In conditional RET instruction the opcode fetch cycle is of 6T states. The T₅ and T₆ states are used to internally check the status of the flag register. If the microprocessor finds that the condition is not true then the microprocessor will not execute the second and third machine cycles.

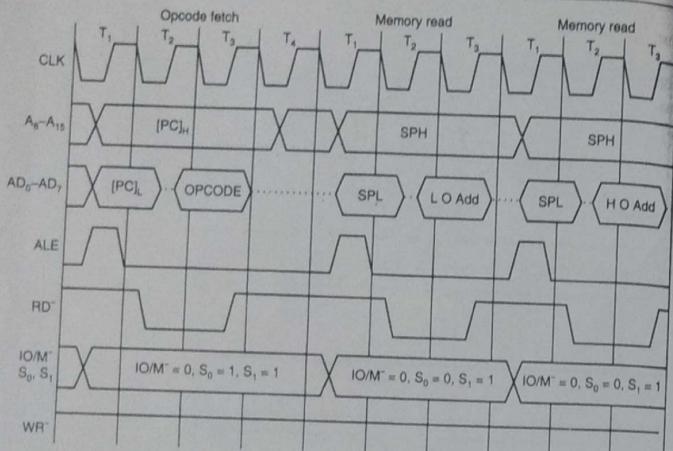


Figure 7.11 Timing diagram of RET.

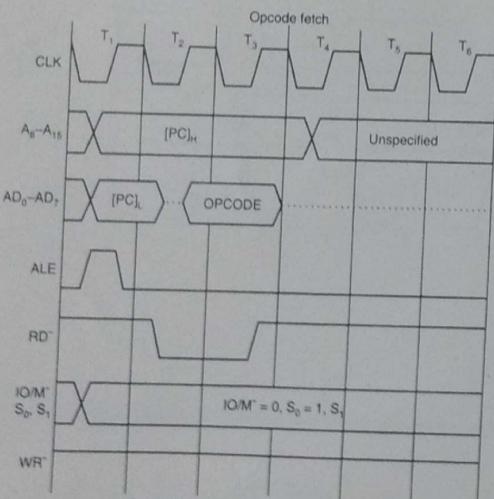


Figure 7.12 Timing diagram of conditional RET (if condition is not true).

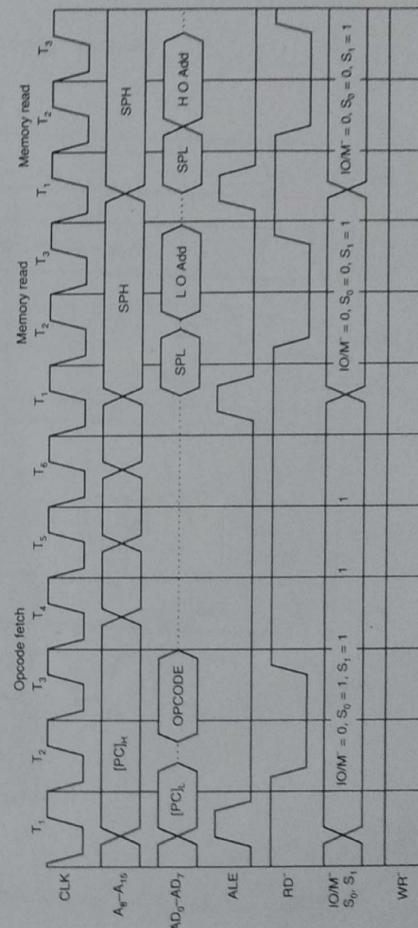


Figure 7.13 Timing diagram of conditional RET (if condition is true).

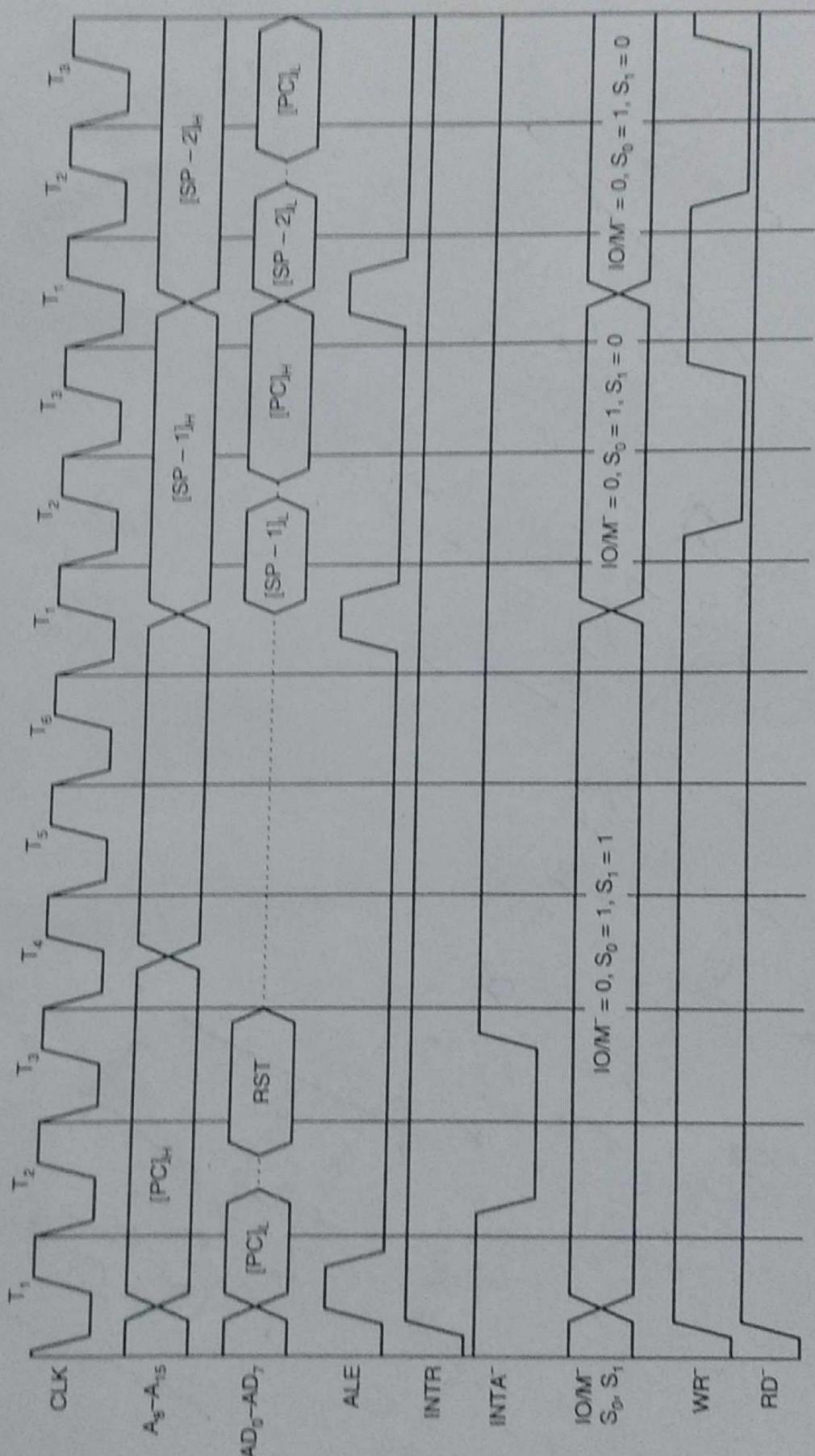


Figure 8.5 Timing diagram of interrupt acknowledges machine cycle.