ONE BYTE INSTRUCTIONS

TIMING DIAGRAM FOR MOVE DATA (MOV R_d , R_s)

Here $\mathbf{R_d}$ represents the destination register where data is to be moved and $\mathbf{R_s}$ represents the source register from where data is to be moved.

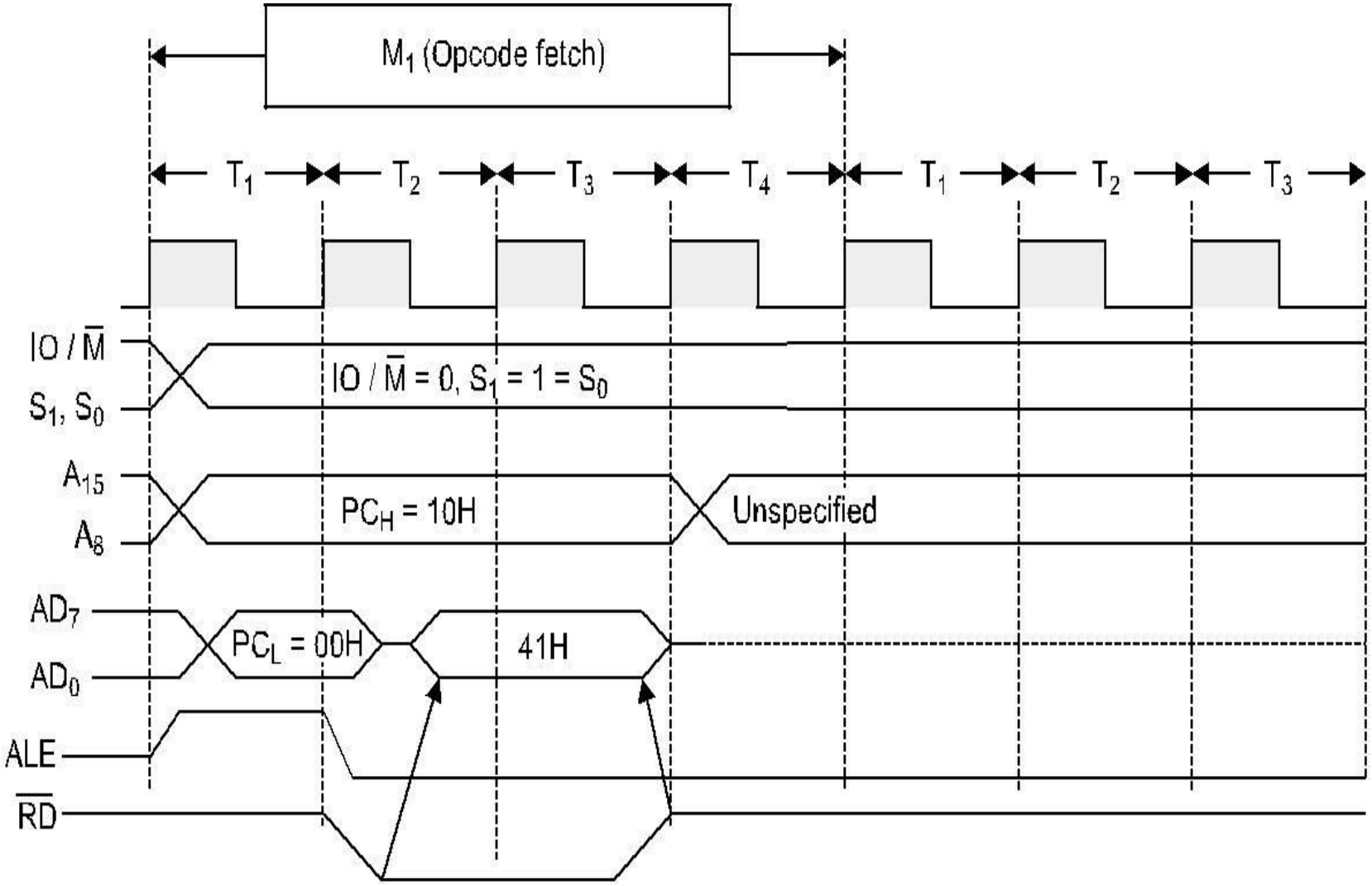
ADDRESS	MNEMONICS	OP-CODE
1000H	MOV B,C	41_H

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 RD control signal to enable the memory and processor places 41H from 1000H location on the data bus.

T_3 : During T_3, the 41H is placed in the instruction register and 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 op-code is decoded in T_4 clock and the action as per 41H is taken accordingly. In other words, the content of C-register is copied in B-register.


M1 (Opcode fetch)
T1
T2
T3
T4
T1
T2
T3
IO / M
S1, S0
IO / M = 0, S1 = 1 = S0
A15
A8
PCH = 10H
Unspecified
AD7
AD0
PCL = 00H
41H
ALE
RD

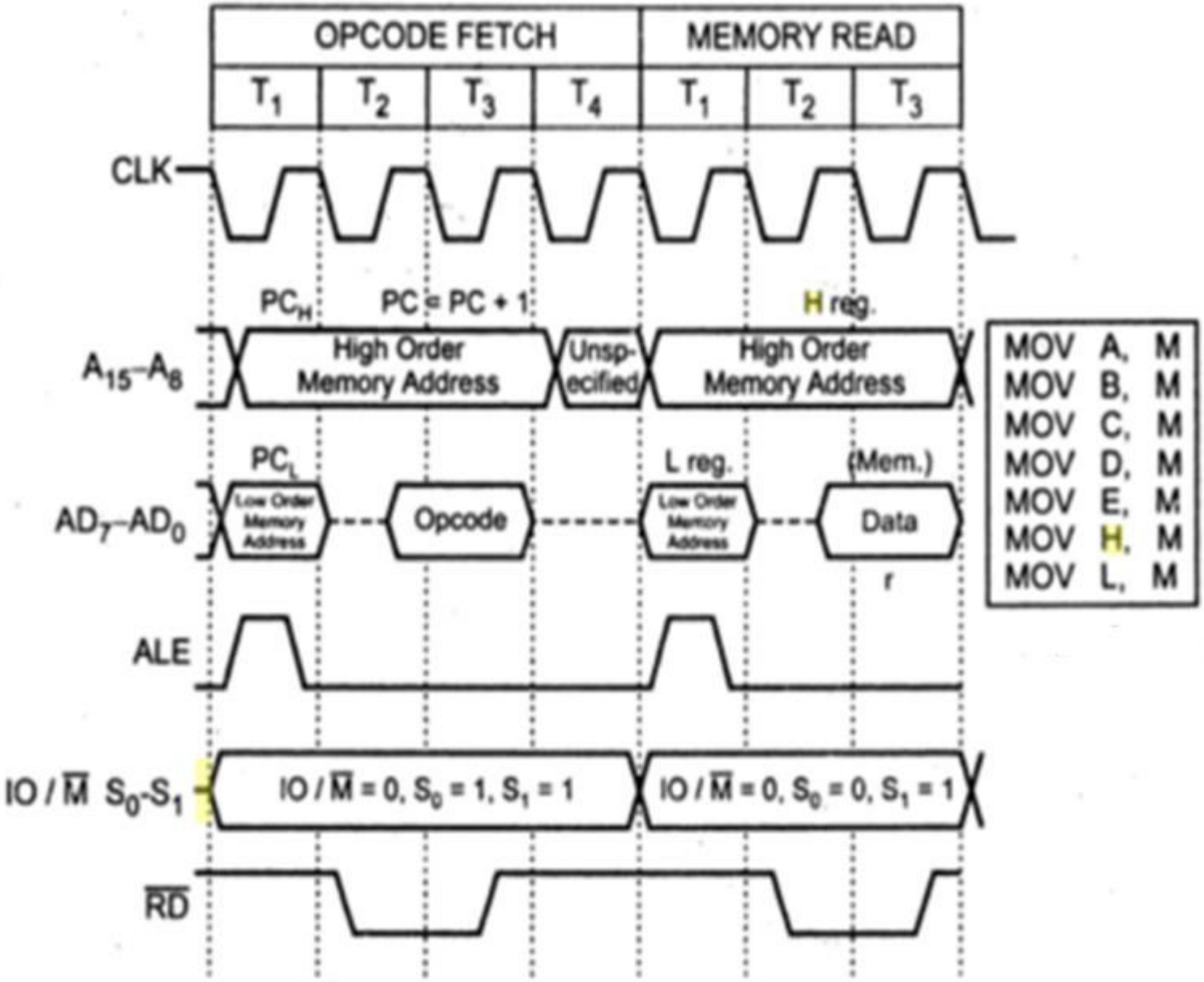
TIMING DIAGRAM FOR MOV r , M

MOV r, M

This instruction copies the contents from memory location pointed by HL register pair into the register specified within the instruction. This instruction requires two machine cycles.

1) **Opcode fetch** : Program counter places address on low order and high order address bus. The opcode at this memory location (e.g. 7EH of MOV A, M) is read into the microprocessor and is decoded. 4 T-states are required for this machine cycle. Program counter is incremented by one.

2) **Memory read** : HL register pair gives address on low order and high order address bus. The data at this addressed memory location is read into the specified register of the microprocessor. Program counter is incremented by one.


OPCODE FETCH
MEMORY READ
T1
T2
T3
T4
T1
T2
T3
CLK
PCH
PC = PC + 1
H reg.
A15–A8
High Order Memory Address
Unspecified
High Order Memory Address
PCL
L reg.
(Mem.)
AD7–AD0
Low Order Memory Address
Opcode
Low Order Memory Address
Data
r
ALE
IO / M S0-S1
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
RD
MOV A, M
MOV B, M
MOV C, M
MOV D, M
MOV E, M
MOV H, M
MOV L, M

TIMING DIAGRAM FOR MOV M , r

EXAMPLE :-

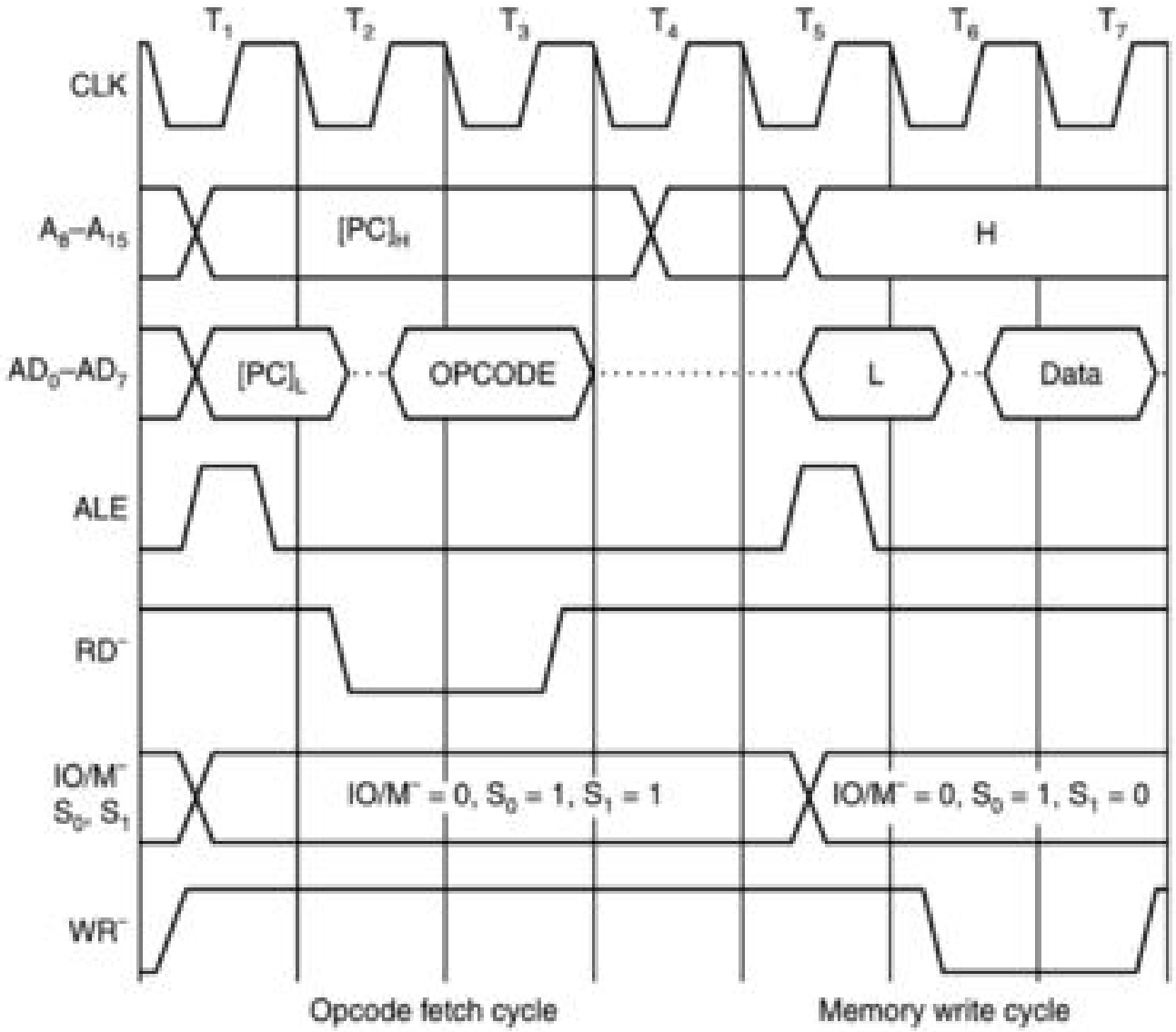
MOV M, A

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.

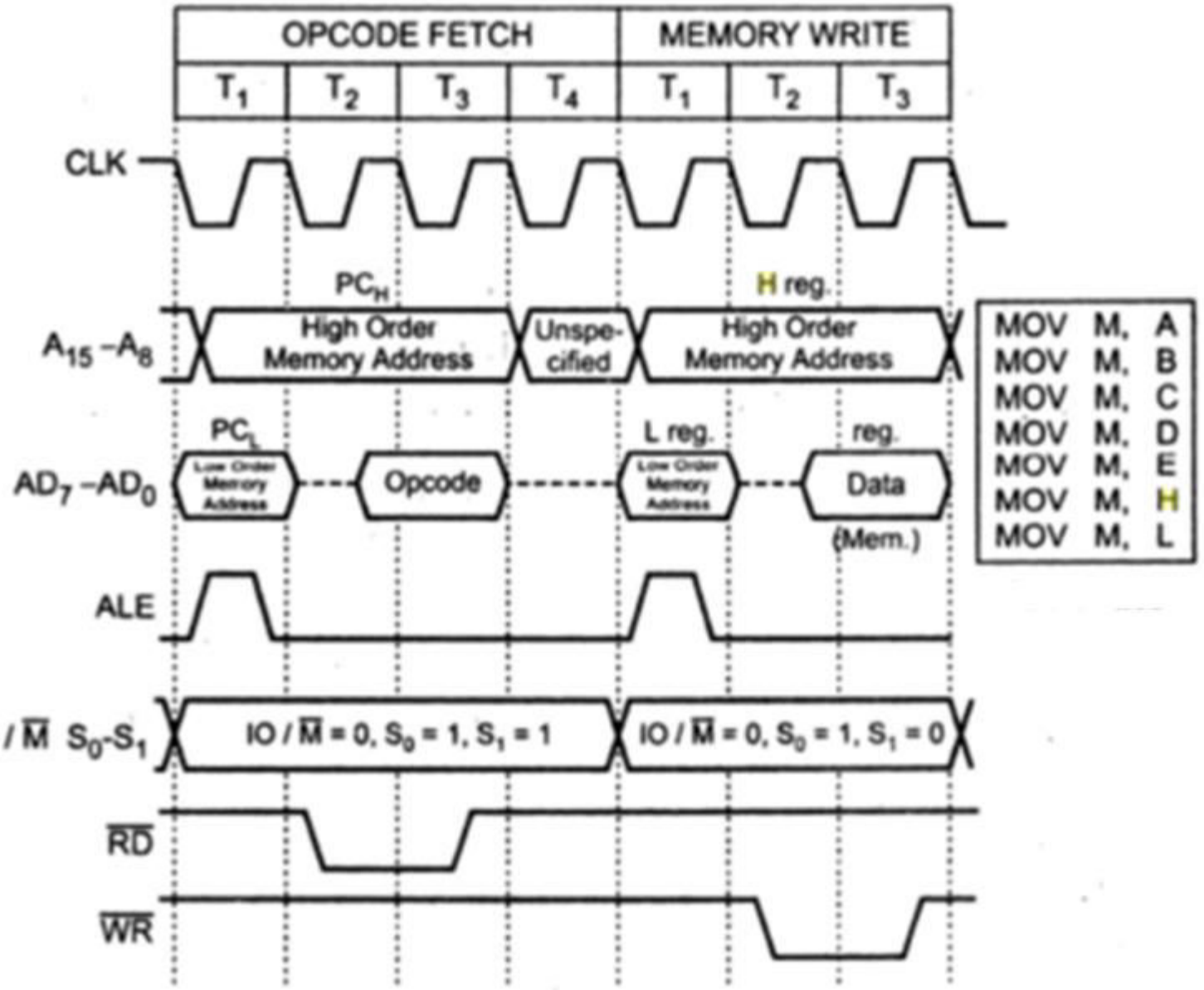
The instruction transfers the contents of specified register to memory. The HL pair gives the address of memory. To transfer data to memory it requires memory write operation, so two machine cycles are required to execute this instruction. They are as follows:

1. Opcode fetch
2. Memory write.

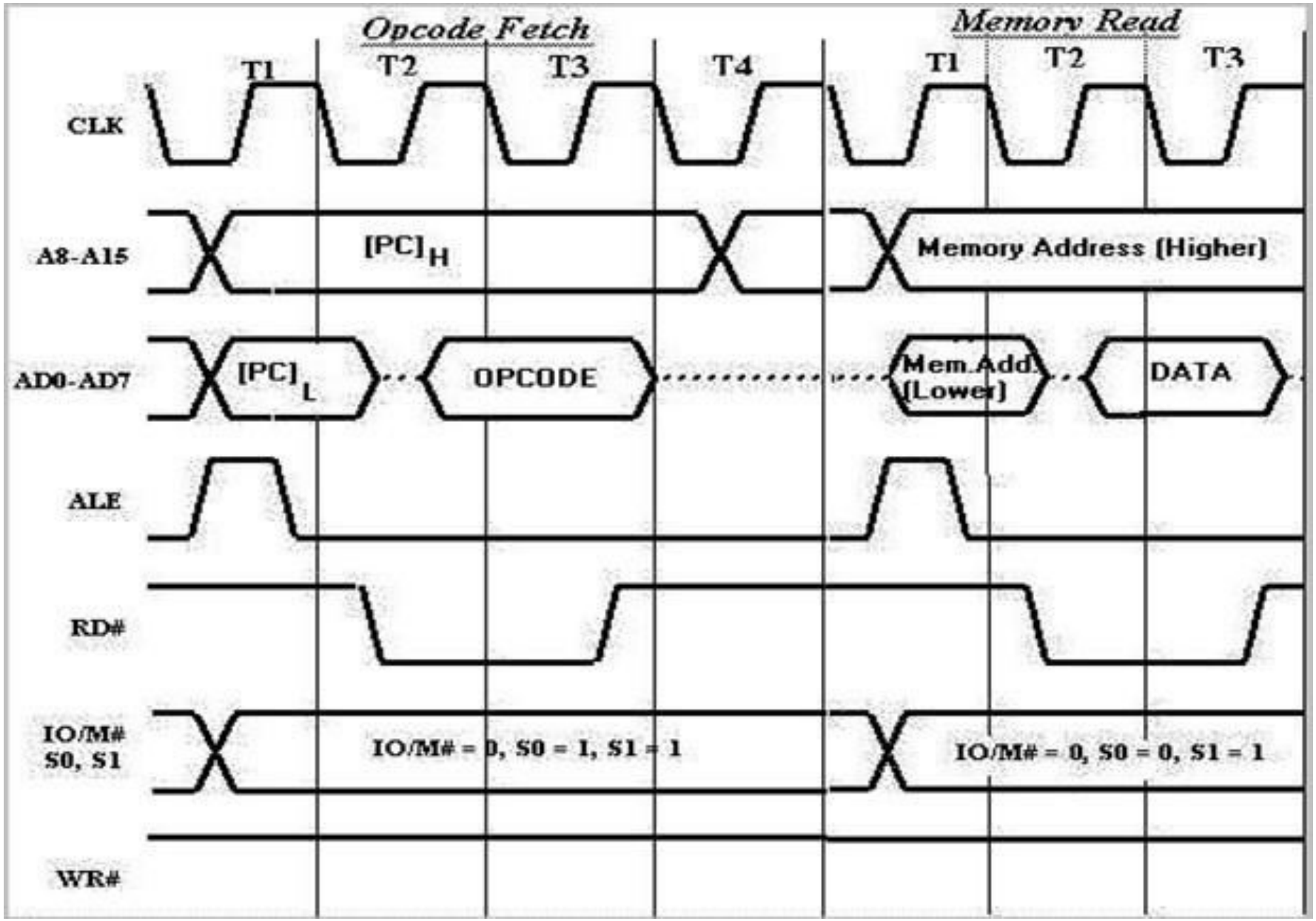
For opcode fetch the PC gives the address of the memory and then it is incremented by one. For memory write cycle the address is given by HL and PC remains unchanged. The opcode fetch does not require T_5- and T_6-states so opcode fetch is of T_1- to T_4-states.



Timing diagram of MOV M, A.


OPCODE FETCH
MEMORY WRITE
T1
T2
T3
T4
T1
T2
T3
CLK
PCH
H reg.
A15 –A8
High Order Memory Address
Unspe-cified
High Order Memory Address
PCL
L reg.
reg.
AD7 –AD0
Low Order Memory Address
Opcode
Low Order Memory Address
Data
(Mem.)
ALE
IO / M S0-S1
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 1, S1 = 0
RD
WR
MOV M, A
MOV M, B
MOV M, C
MOV M, D
MOV M, E
MOV M, H
MOV M, L

TIMING DIAGRAM FOR ADD M



TIMING DIAGRAM FOR ADD R

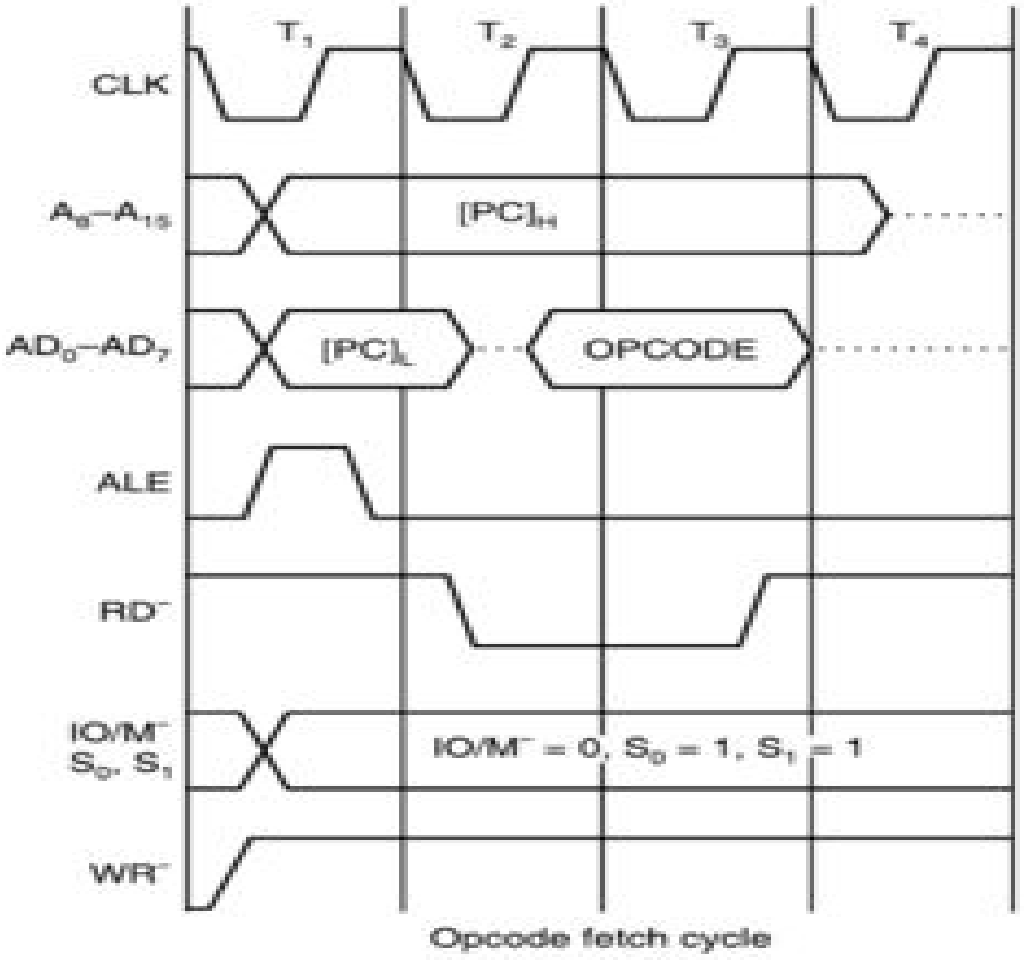
EXAMPLE :- ADD B

ADDRESS	MNEMONICS	OP-CODE
1000H	ADD B	80_H

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.

EXAMPLE :- ADD B

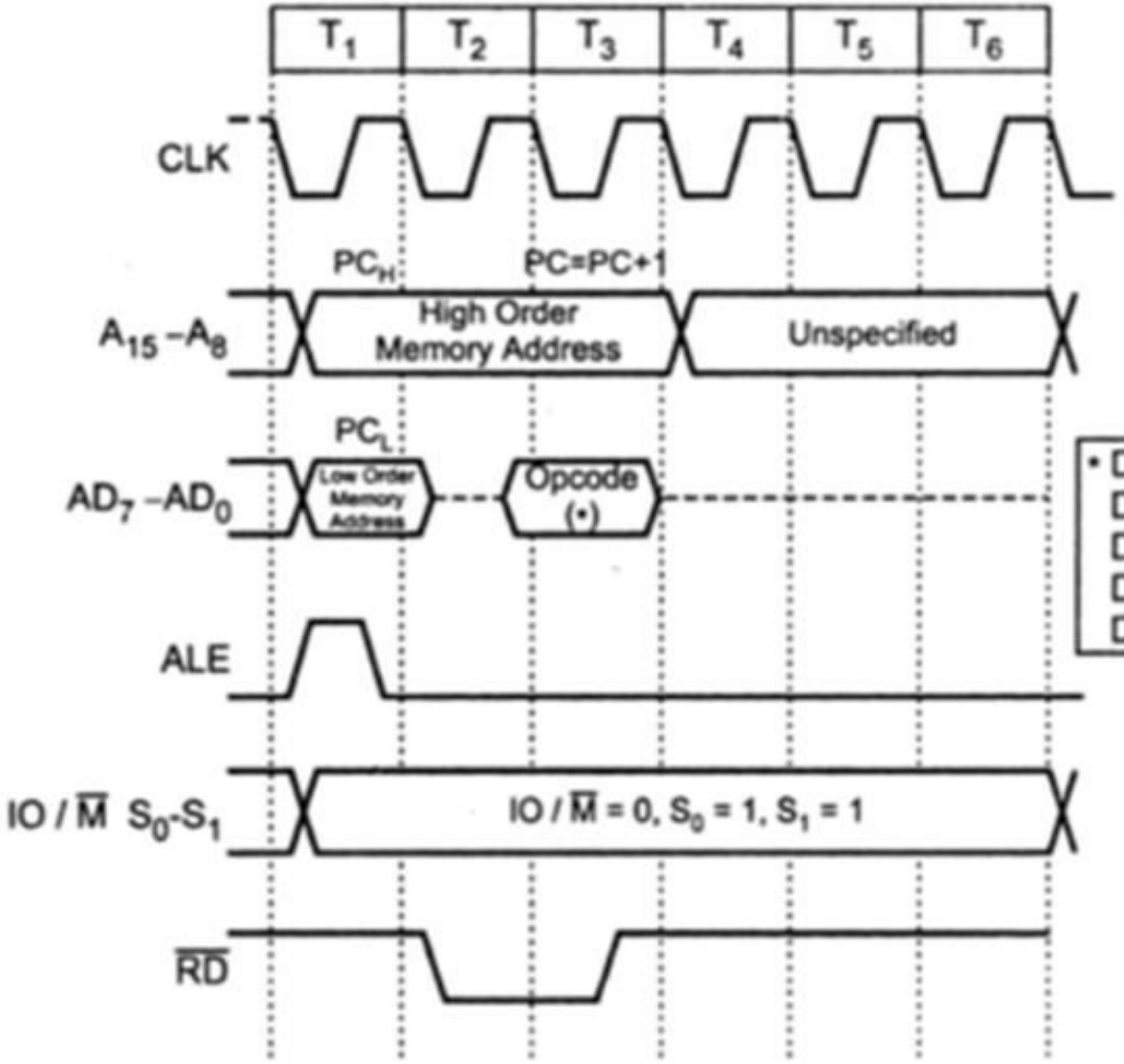


Timing diagram of ADD B.

TIMING DIAGRAM FOR INX R_P and DCX R_P

DCX rp and INX rp

These instructions decrement/increment the contents of register pair (rp) specified within the register by one and store result in the same register pair. It requires only opcode fetch machine cycle. In this cycle, program counter gives address on low order and high order address bus. The opcode of DCX rp (e.g. 0B of DCX B) is read into the microprocessor and it decodes it. This machine cycle requires 6 T-states. Fig. A.36 gives the timing diagram of DCX rp. The instruction INX rp also has same timing diagram, the only difference is the opcode.



• DCX		rp	INX		rp
DCX	B	0B	INX	B	03
DCX	D	1B	INX	D	13
DCX	H	2B	INX	H	23
DCX	SP	3B	INX	SP	33

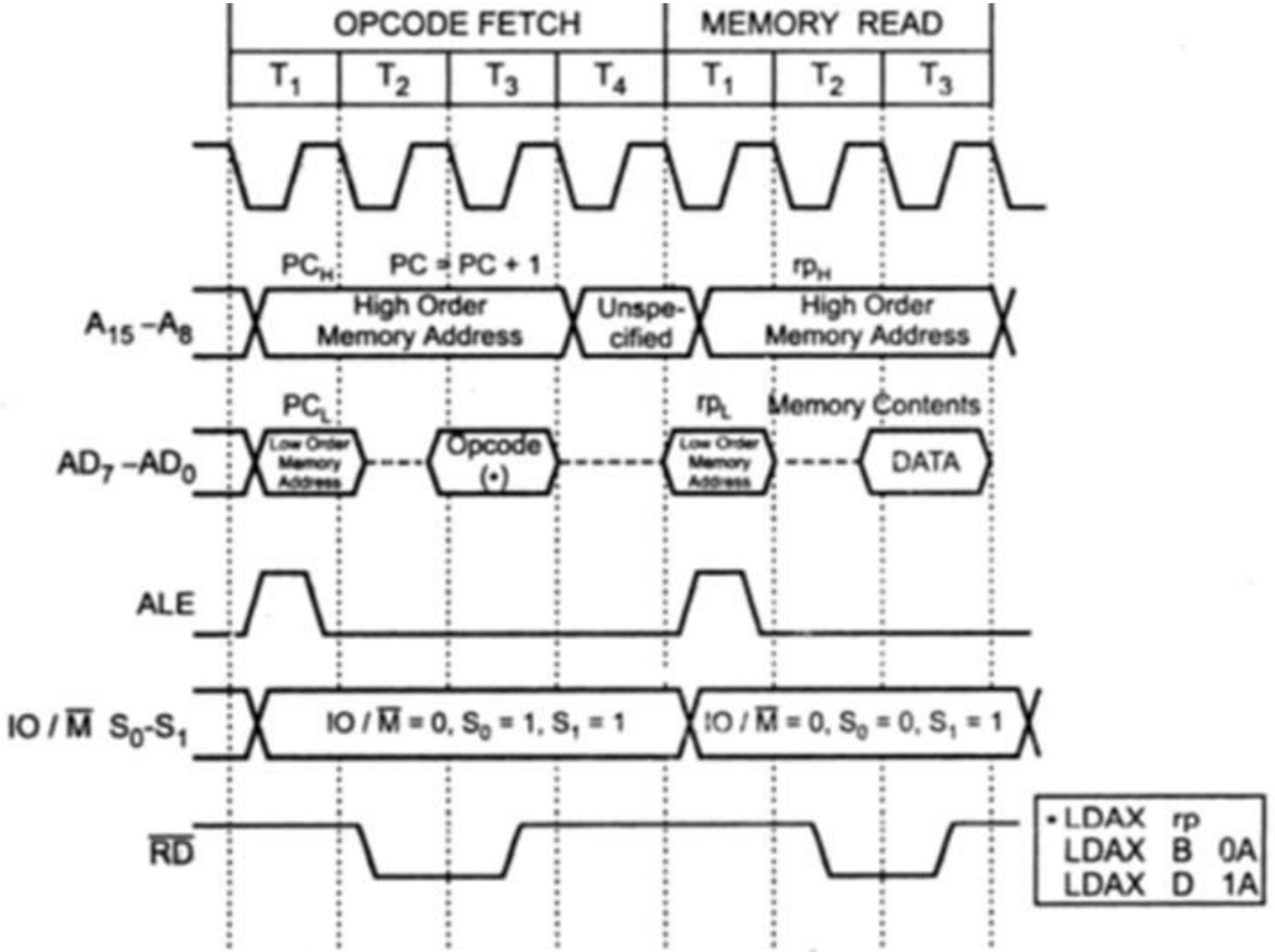
TIMING DIAGRAM FOR LDAX R_P

LDAX rp

This instruction loads A register with the contents of memory location whose address is specified by register pair (BC or DE). It requires the following machine cycles.

1) Opcode fetch : Program counter places the memory address on low-order and high-order address bus. This machine cycle is required for reading the opcode of LDAX rp (eg. 0A for LDAX B) into the microprocessor and decode it.

2) Memory read : This machine cycle is required for reading the data into the accumulator. The address at which the data is stored is obtained from register pair specified within the instruction. In this machine cycle higher order register contents are kept on higher-order address bus and lower order register contents are kept on lower-order address bus. The data is read into the microprocessor (register A) from the addressed memory location.


OPCODE FETCH
MEMORY READ
T1
T2
T3
T4
T1
T2
T3
PCH
PC = PC + 1
rpH
A15 –A8
High Order Memory Address
Unspe-cified
High Order Memory Address
PCL
rpL
Memory Contents
AD7 –AD0
Low Order Memory Address
Opcode (•)
Low Order Memory Address
DATA
ALE
IO / M S0-S1
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
RD
• LDAX rp
LDAX B 0A
LDAX D 1A

TIMING DIAGRAM FOR DCR M

INR M and DCR M

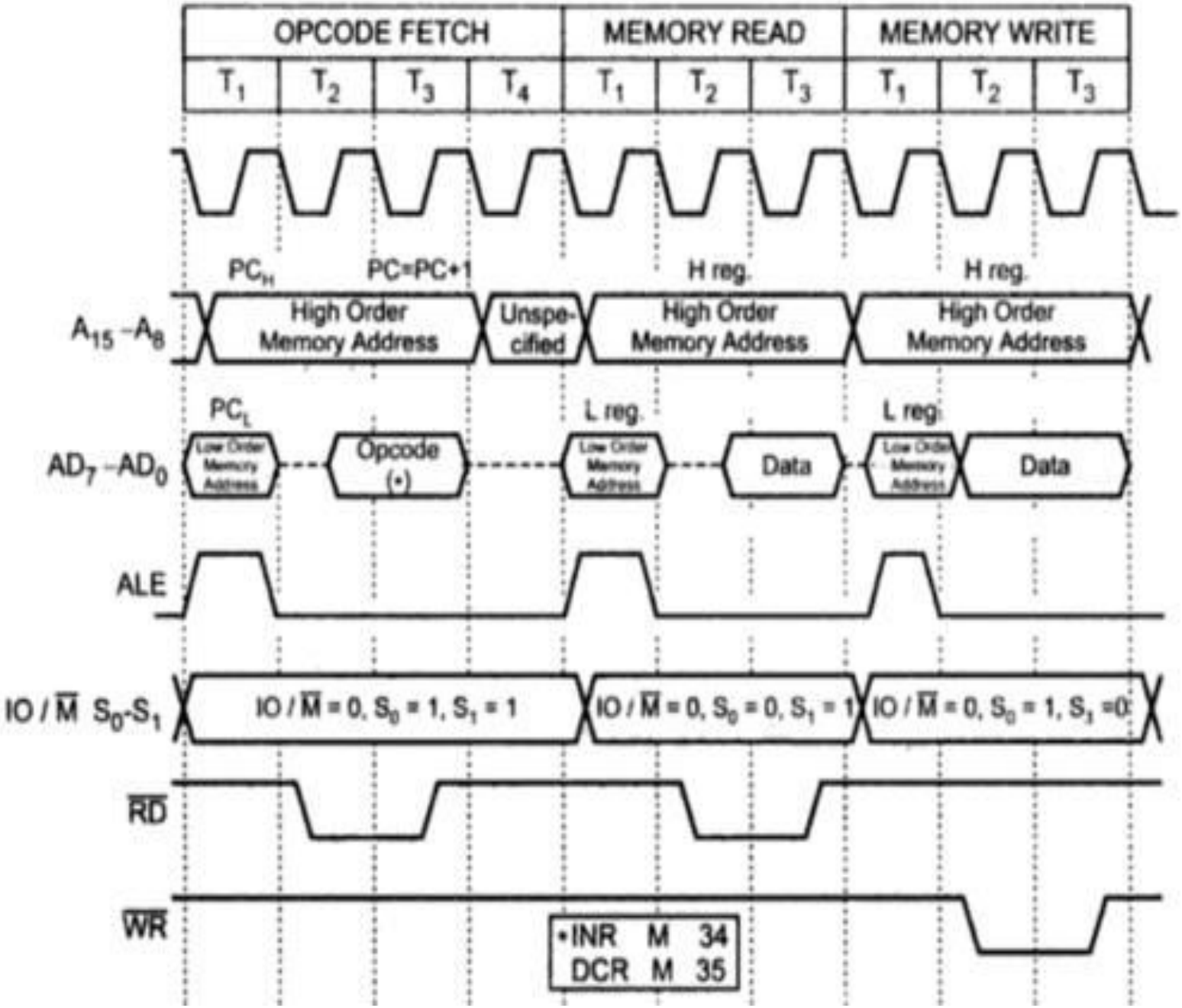
These instructions increment/decrement the contents of memory by one where memory address is specified by HL register pair. It requires three machine cycles as explained below.

1) Opcode fetch : Program counter gives the memory address on low-order and high-order address bus. This type of machine cycle is required for reading the opcode (e.g. 34H of INR M) into the microprocessor and to decode it.

2) Memory read : The contents of HL register pair give the address of memory where the data (which is to be incremented) is stored. In this machine cycle data is read into the microprocessor from this memory location.

3) Memory write : The microprocessor writes the incremented data at the memory location given by the contents of HL register pair.

Fig. gives the number of T-states and timing required for each of the above operation. The DCR M instruction decrements the contents of memory by one where memory is specified by HL register pair. The timing diagram for DCR M is same as for INR M. Only difference is in opcode.


OPCODE FETCH
MEMORY READ
MEMORY WRITE
T1 T2 T3 T4 T1 T2 T3 T1 T2 T3
PCH
PC=PC+1
H reg.
H reg.
A15 –A8
High Order Memory Address
Unspe-cified
High Order Memory Address
High Order Memory Address
PCL
L reg.
L reg.
AD7 –AD0
Low Order Memory Address
Opcode (•)
Low Order Memory Address
Data
Low Order Memory Address
Data
ALE
IO / M S0-S1
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
IO / M = 0, S0 = 1, S1 =0
RD
WR
•INR M 34
DCR M 35

Timing diagram for INR M (INR 4250)

- Fetching the Opcode 34H from the memory 4105H. (OF cycle)

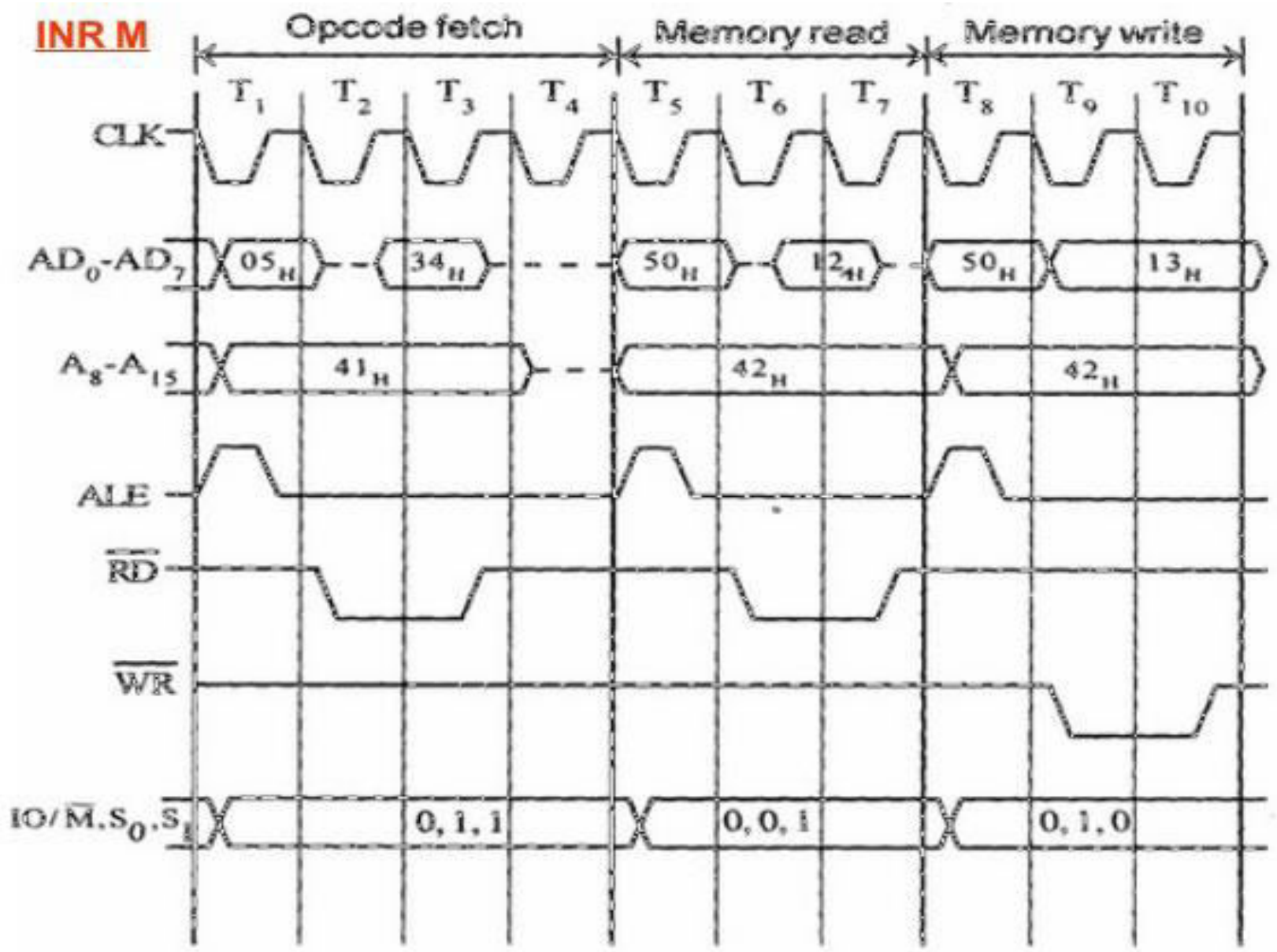
- Let the memory address (M) be 4250H. (MR cycle -To read Memory address and data)

- Let the content of that memory is 12H.

- Increment the memory content from 12H to 13H. (MW CYCLE)

Address	Mnemonics	Opcode
4105	INR M	34_H

INR M


Opcode fetch
Memory read
Memory write
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10
CLK
AD0-AD7
05H
34H
50H
42H
50H
13H
A8-A15
41H
42H
42H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
0, 1, 0

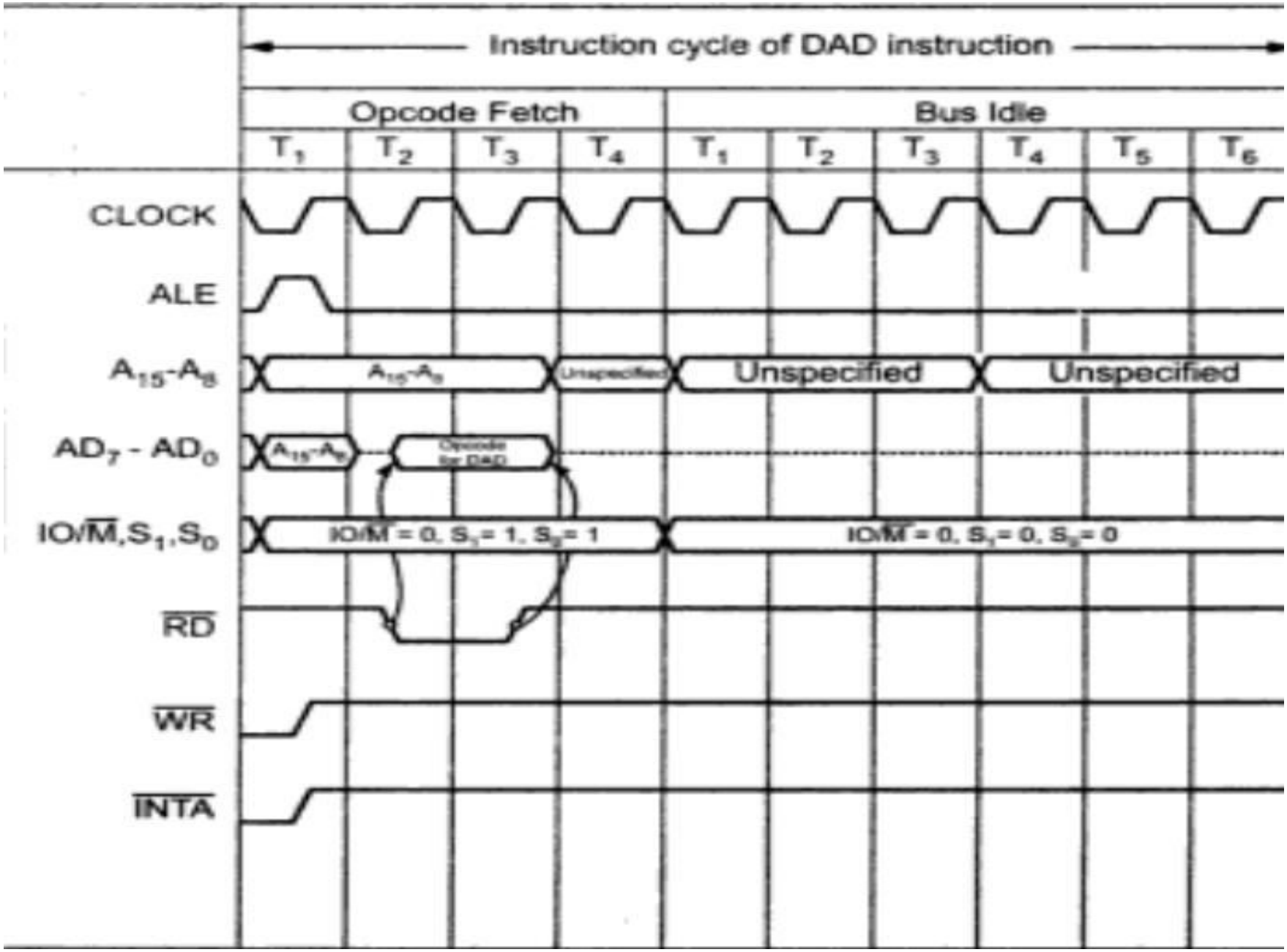
TIMING DIAGRAM FOR DAD (DAD R_P)

1. For execution of DAD instruction (this instruction adds the contents of a specified register pair to the contents of HL register pair) ten T states are required. This means that after execution of opcode fetch machine cycle, DAD instruction requires 6 extra T-states to add 16 bit contents of a specified register pair to the contents of HL register pair. These extra T-states which are divided into two machine cycles do not involve any memory or I/O operation. These machine cycles are called BUS IDLE machine cycles.

Example :- DAD B adds the content of register pair BC with content of register pair HL.

In the case of DAD, these Bus Idle cycles are similar to memory read cycles, except $\overline{RD}$ and ALE signals are not activated.

ADDRESS	MNEMONICS	OP-CODE
4000_H	DAD B	09_H


Instruction cycle of DAD instruction
Opcode Fetch
Bus Idle
T1
T2
T3
T4
T1
T2
T3
T4
T5
T6
CLOCK
ALE
A15-A8
A15-A8
Unspecified
Unspecified
Unspecified
AD7 - AD0
A15-A8
Opcode for DAD
IO/M, S1, SD
IO/M = 0, S1 = 1, S0 = 1
IO/M = 0, S1 = 0, S0 = 0
RD
WR
INTA

TIMING DIAGRAM FOR POP (POP R_P)

POP rp

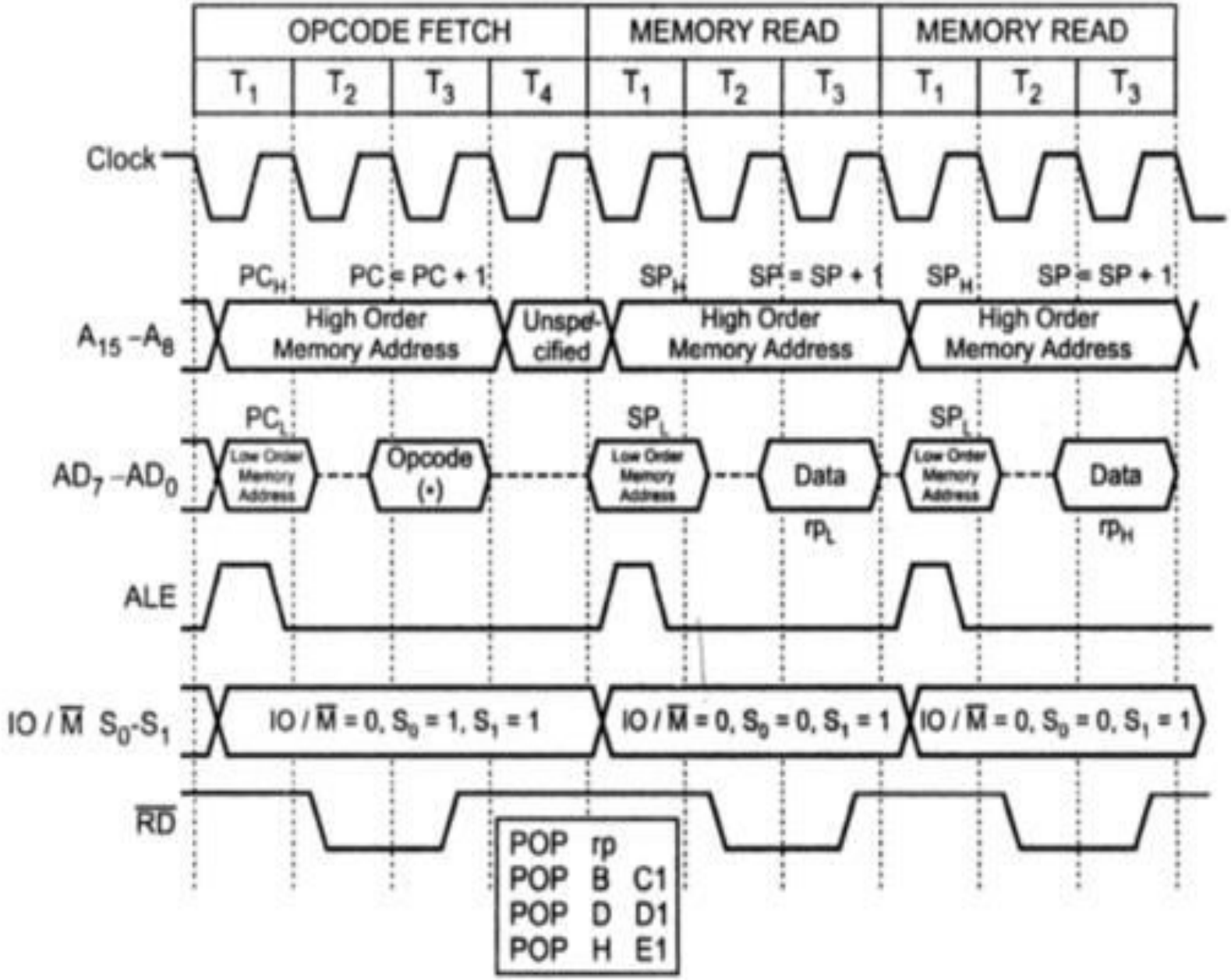
Fig. A.46 shows the timing diagram of POP rp instruction. This instruction copies the contents of memory location pointed by the stack pointer into the lower byte of the specified register pair and increments the stack pointer by one. It then copies the contents of memory location pointed by stack pointer into the higher byte of the specified register pair and increments the stack pointer again by one.

This instruction requires the following machine cycles.

1) Opcode fetch : Program counter places address on low order and high order address bus. The opcode of POP rp (e.g. C1 of POP B) is read into the microprocessor and is decoded in this machine cycle. The program counter is incremented by one.

2) Memory read : The stack pointer gives address on low-order and high-order address bus. The data present at this address is loaded into the lower byte of the specified register pair (e.g. in C register for BC register pair). The stack-pointer is incremented by one.

3) Memory read : The stack pointer gives address on low-order and high-order address bus. The data present at this address is loaded into high order byte of the specified register pair (e.g. in B register for BC register pair).The stack pointer is again incremented by one.


OPCODE FETCH
MEMORY READ
MEMORY READ
T1
T2
T3
T4
T1
T2
T3
T1
T2
T3
Clock
PCH
PC = PC + 1
SPH
SP = SP + 1
SPH
SP = SP + 1
A15 –A8
High Order Memory Address
Unspe-cified
High Order Memory Address
High Order Memory Address
PCL
SPL
SPL
AD7 –AD0
Low Order Memory Address
Opcode (•)
Low Order Memory Address
Data
rpL
Low Order Memory Address
Data
rpH
ALE
IO / M S0-S1
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
RD
POP rp
POP B C1
POP D D1
POP H E1

TIMING DIAGRAM FOR PUSH (PUSH R_P)

PUSH rp

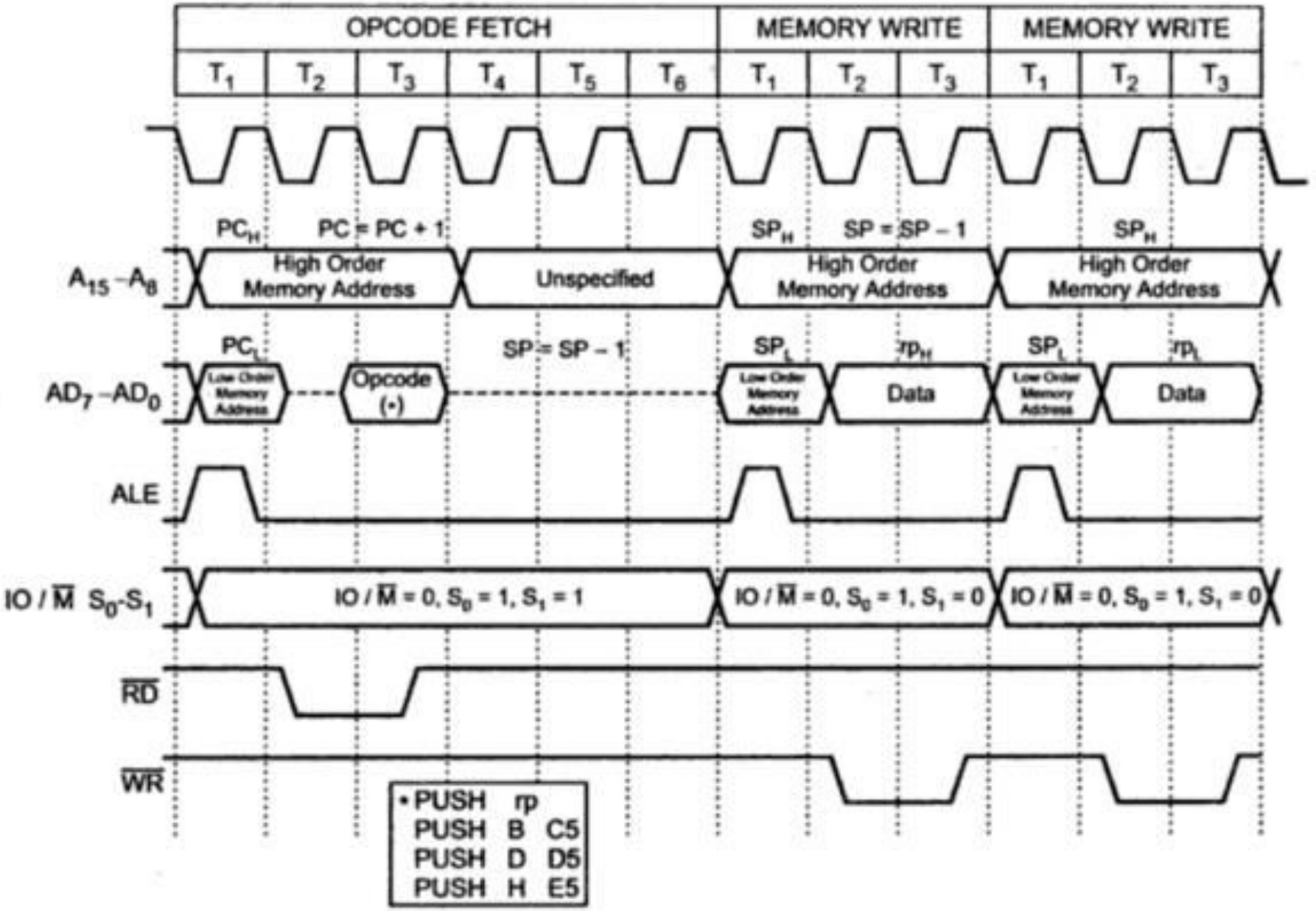
This instruction decrements stack pointer by one and copies the higher byte of the register pair (rp) into the memory location pointed by stack pointer. It then decrements the stack pointer again by one and copies the lower byte of the register pair into the memory location pointed by stack pointer. The machine cycles required for this instruction are explained below.

1) Opcode fetch : Program counter places address on low order and high order address bus. The opcode of PUSH rp. (e.g. C5H of PUSH B) is read into the microprocessor. It decodes it. Program counter is incremented by one. Stack pointer is decremented by one.

2) Memory write : Stack pointer places address on low order and high order address bus. Microprocessor writes contents of high order register (e.g. B in BC) at this address. Stack pointer is again decremented by one.

3) Memory write : Stack pointer gives address on low order and high order address bus.

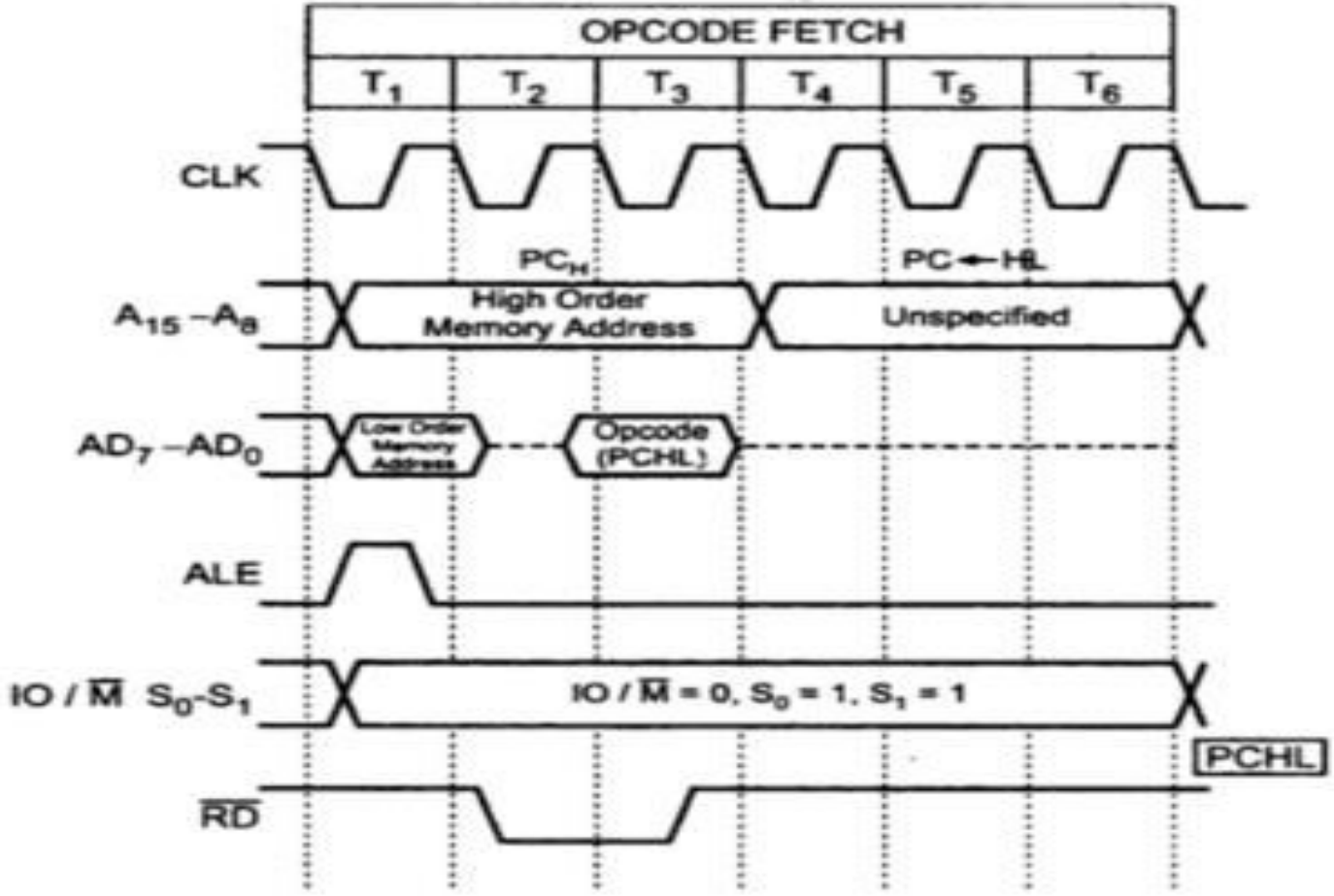
Microprocessor writes contents of low order register (e.g. C in BC) at this address.


OPCODE FETCH
MEMORY WRITE
MEMORY WRITE
T1 T2 T3 T4 T5 T6
T1 T2 T3
T1 T2 T3
PCH
PC = PC + 1
SPH
SP = SP – 1
SPH
A15 –A8
High Order Memory Address
Unspecified
High Order Memory Address
High Order Memory Address
PCL
SP = SP – 1
SPL
rpH
SPL
rpL
AD7 –AD0
Low Order Memory Address
Opcode (•)
Low Order Memory Address
Data
Low Order Memory Address
Data
ALE
IO / M S0-S1
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 1, S1 = 0
IO / M = 0, S0 = 1, S1 = 0
RD
WR
• PUSH rp
PUSH B C5
PUSH D D5
PUSH H E5

TIMING DIAGRAM FOR PCHL

PCHL

This instruction loads the contents of HL register pair into the program counter. It requires only opcode fetch machine cycle. In this, the program counter gives address on low order and high order address bus. Microprocessor reads the opcode of PCHL (E9H) from this memory address and decodes it. This requires 6 T-states.



TIMING DIAGRAM FOR RET

Return from subroutine unconditionally

RET

This instruction POPs the return address (address of the instruction next to call in the main program) from the stack and loads program counter with this return address. Thus transfers program control to the instruction next to CALL in the main program. It requires three machine cycles as explained below.

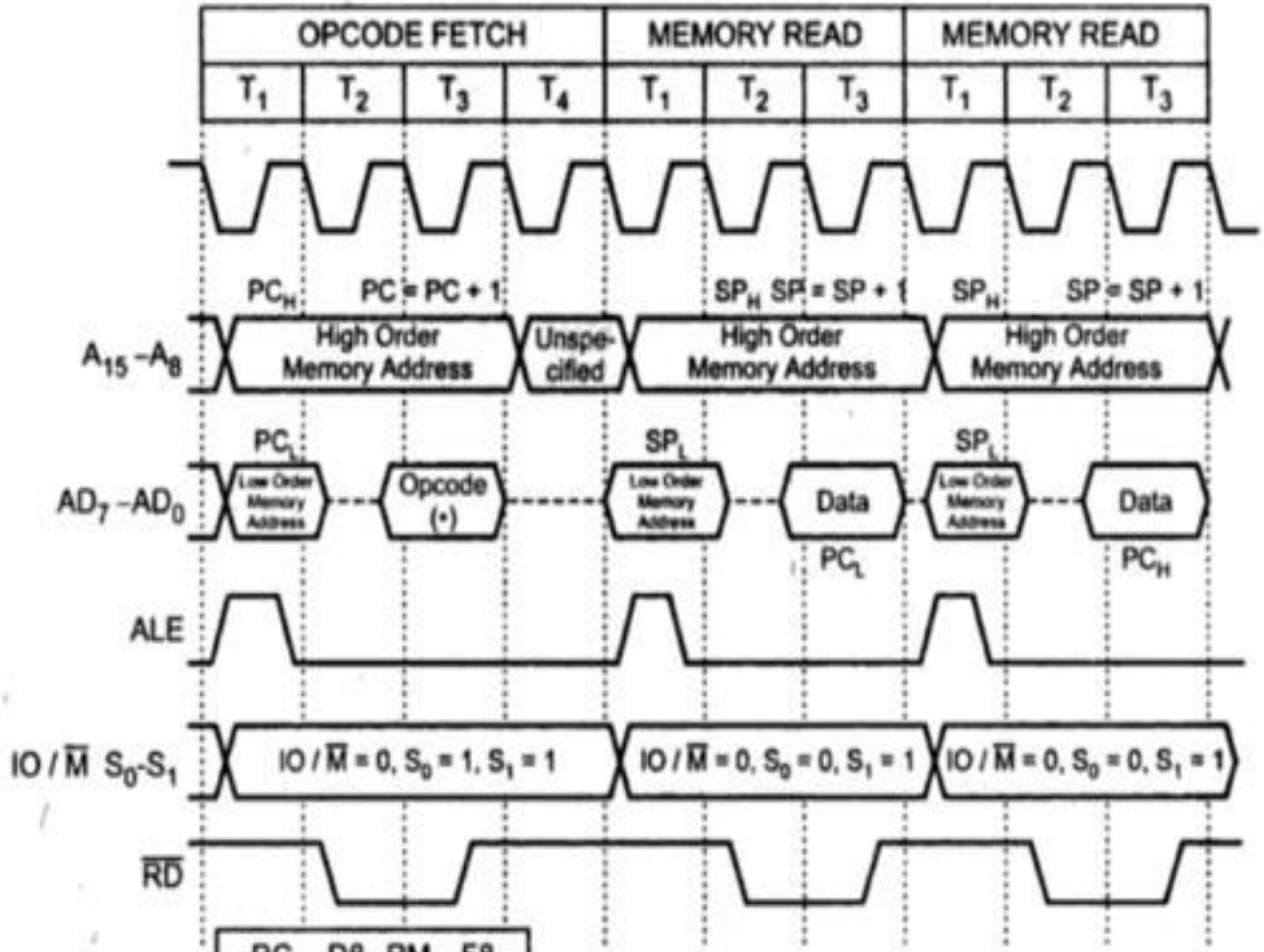
1) Opcode fetch : Program counter places address on low order and high order address bus. From this address microprocessor reads the opcode of RET (C9H) and decodes it. Program counter is incremented by one.

2) Memory read : This machine cycle reads the "data" into the microprocessor from the memory whose address is the contents of stack pointer. This "data" is the low order byte of the address to which the program control is to be transferred. Stack pointer is incremented by one.

3) Memory read : This machine cycle reads the "data" in the microprocessor from the memory whose address is the contents of stack pointer. This "data" is the high order byte of the address to which the program control is to be transferred, Stack pointer is again incremented by one.

NOTE :-

The timing diagram of $R_{condition}$, when condition is valid is same as that of RET instruction.



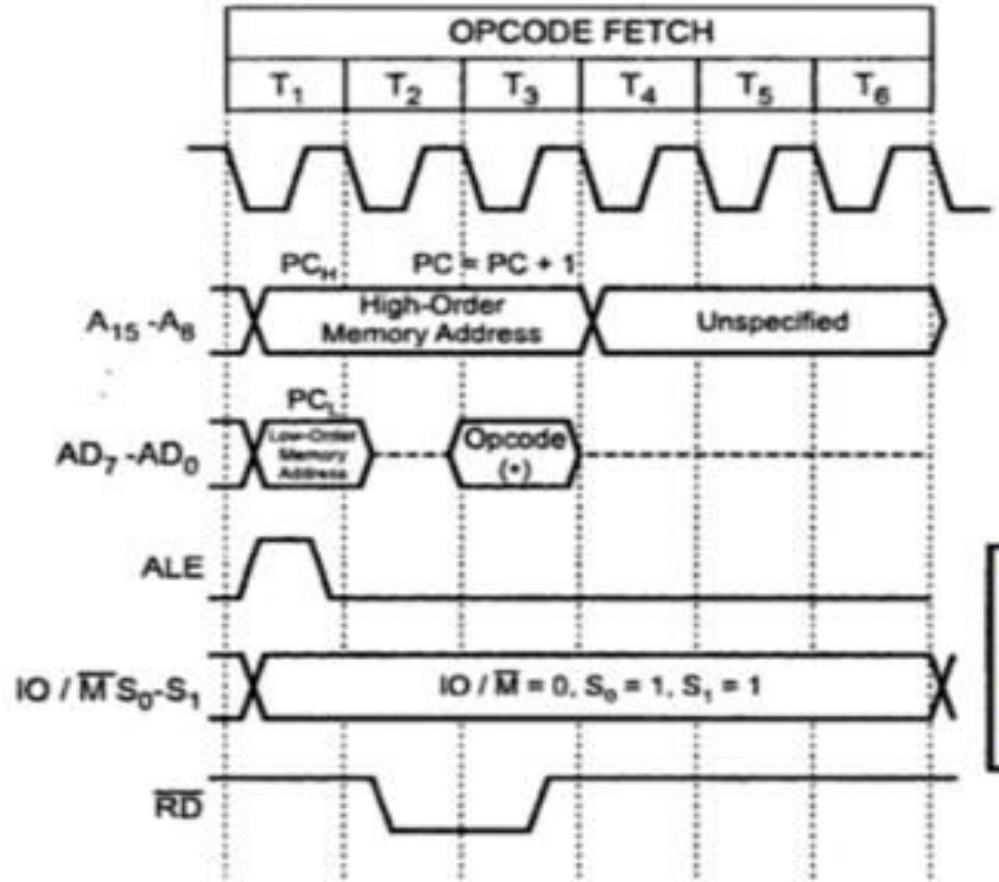
RC	D8	RM	F8
RNC	D0	RD	F0
RNZ	C0	RPE	E8
RZ	C8	RPO	E0

RET R conditional (when condition is valid)
* for instruction RNZ, if zero flag is not set at the time of execution of instruction condition is valid

R conditional (When condition is not valid)

This instruction requires opcode fetch machine cycle.

6 T-states are required for this opcode fetch machine cycle. This instruction transfers program control to next instruction written after this instruction. It requires opcode fetch machine cycle. In this cycle, program counter places address on low-order and high-order address bus. Opcode of Recondition (e.g. C0 of RNZ) is read into the microprocessor from the addressed memory location. Program counter is incremented by one.



RC	D8
RNC	D0
RNZ	C0
RZ	C8
RM	F8
RD	F0
RPE	E8
RPO	E0

R conditional (• when condition is not valid) and for instruction RNZ, if zero flag is set at the time of execution of instruction condition is not valid

TWO BYTE INSTRUCTIONS

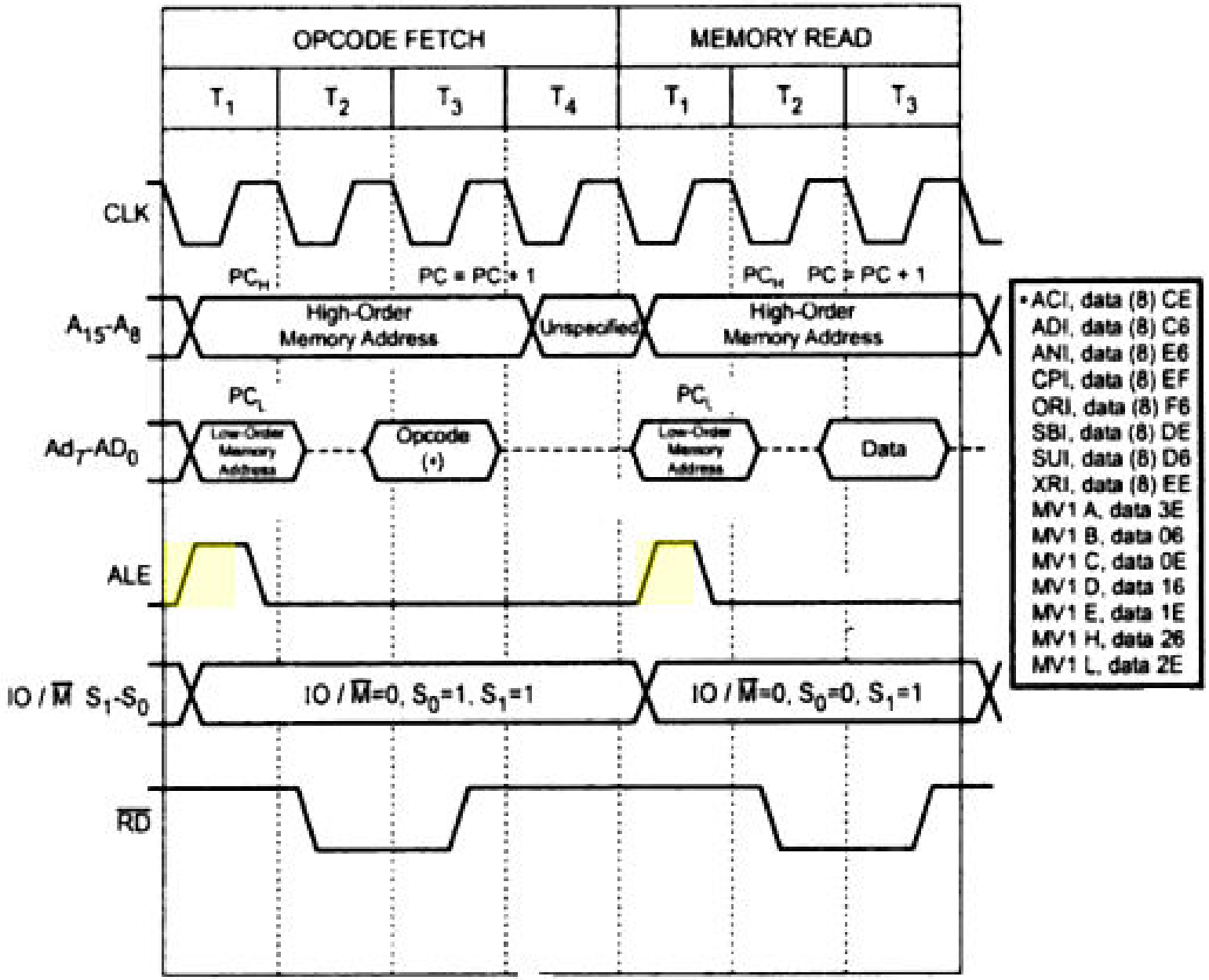
TIMING DIAGRAM FOR ACI Data, ADI Data, ANI Data, CPI Data, ORI Data, SBI Data, SUI Data, XRI Data

ACI, data.. ADI, data.. ANI, data.. CPI, data.. ORI, data.. SBI, data.. SUI, data.. XRI, data..

These instructions perform logical operation specified in the instruction with the contents of accumulator and the data within the instruction. They require the following machine cycles.

1) Opcode fetch : Program counter gives the memory address on low-order and high order address bus. This machine cycle is required for reading the opcode into the microprocessor and decode it. Program counter is incremented by one.

2) Memory read : This machine cycle reads the data from addressed memory location and after performing specified logical operation result is stored in the accumulator.


OPCODE FETCH
MEMORY READ
T1
T2
T3
T4
T1
T2
T3
CLK
PCH
PC = PC + 1
PCH
PC = PC + 1
A15-A8
High-Order Memory Address
Unspecified
High-Order Memory Address
PCL
PCL
AD7-AD0
Low-Order Memory Address
Opcode (•)
Low-Order Memory Address
Data
ALE
IO / M S1-S0
IO / M=0, S0=1, S1=1
IO / M=0, S0=0, S1=1
RD
• ACI, data (8) CE
ADI, data (8) C6
ANI, data (8) E6
CPI, data (8) EF
ORI, data (8) F6
SBI, data (8) DE
SUI, data (8) D6
XRI, data (8) EE
MVI A, data 3E
MVI B, data 06
MVI C, data 0E
MVI D, data 16
MVI E, data 1E
MVI H, data 26
MVI L, data 2E

TIMING DIAGRAM FOR MVI R, DATA

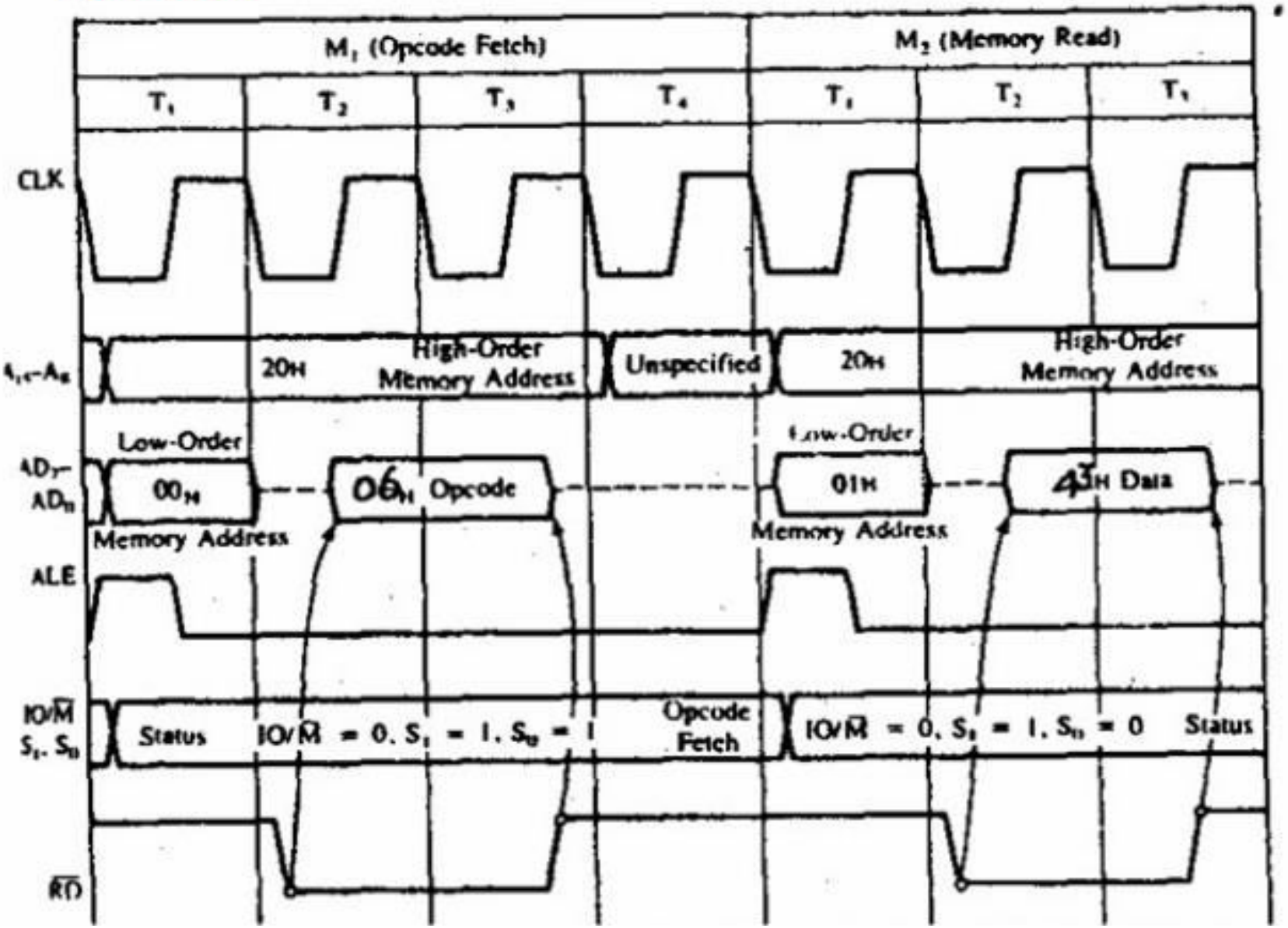
Timing diagram for MVI B, 43H.

Fetching the Opcode 06H from the memory 2000H. (OF machine cycle)

Read (move) the data 43H from memory 2001H. (memory read)

Address	Mnemonics	Opcode
2000	MVI B, 43_H	06_H
2001		43_H

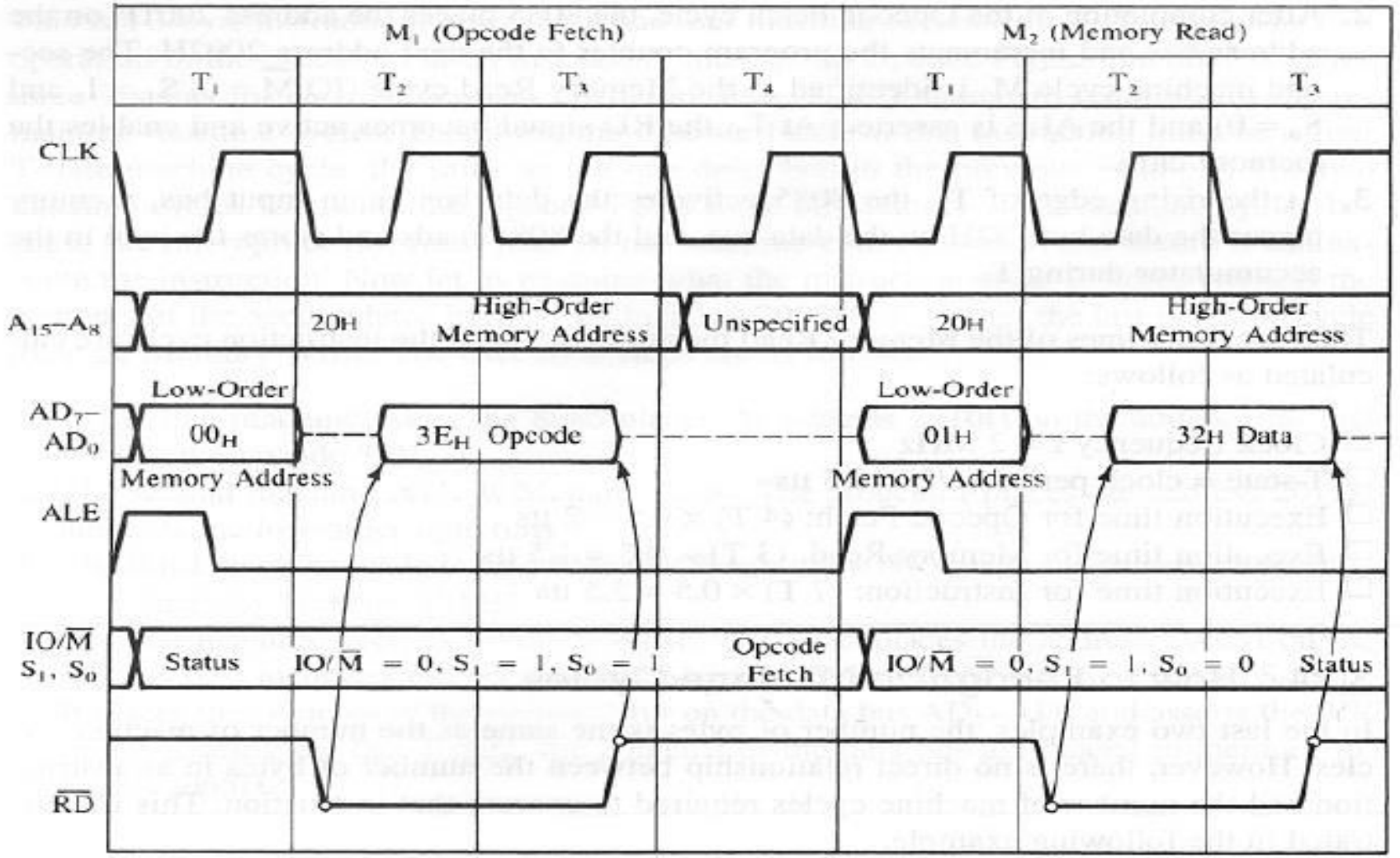
MVI B, data

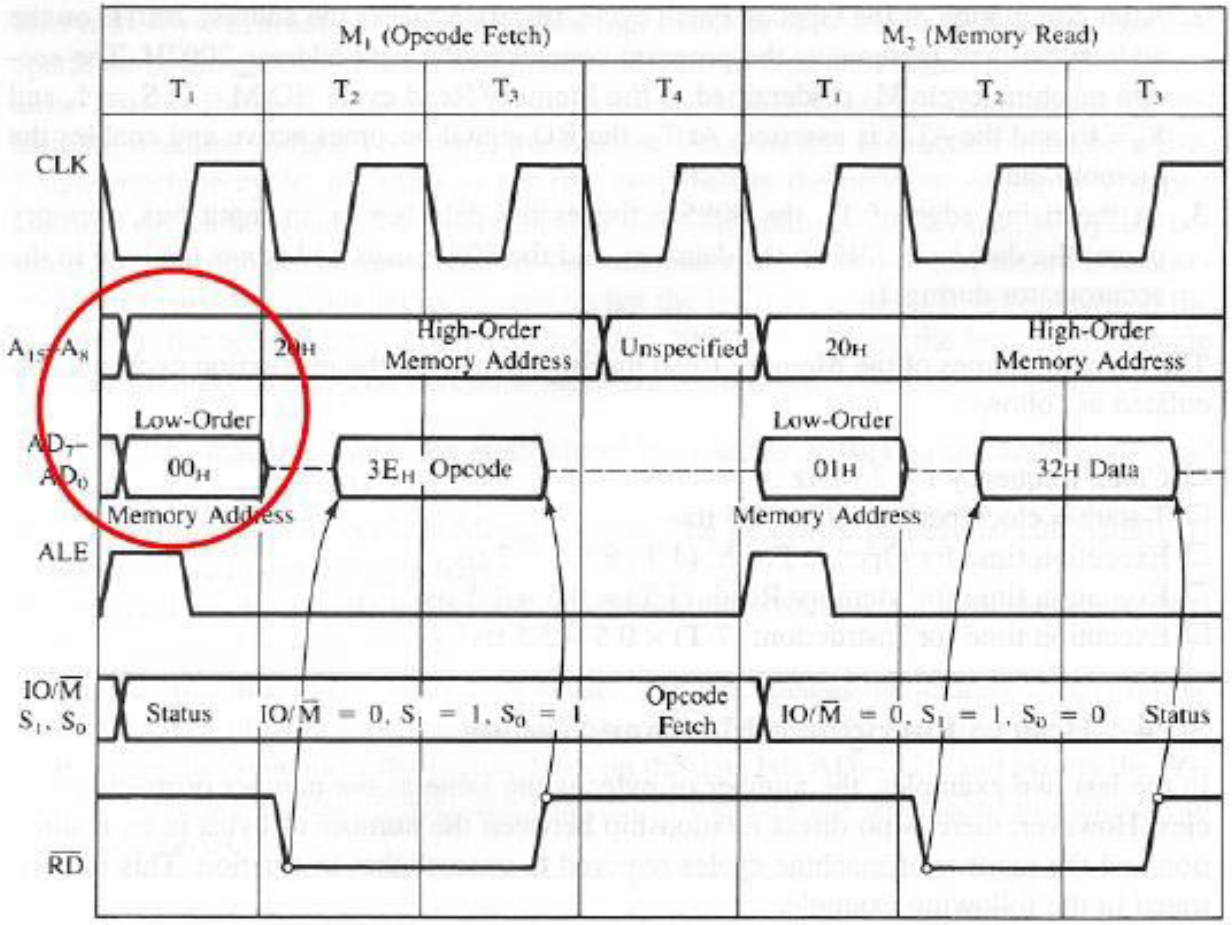

M1 (Opcode Fetch)
M2 (Memory Read)
T1
T2
T3
T4
T1
T2
T3
CLK
A15–A8
20H
High-Order Memory Address
Unspecified
20H
High-Order Memory Address
AD7–AD0
Low-Order Memory Address
00H
06H Opcode
Low-Order Memory Address
01H
45H Data
ALE
IO/M S1, S0
Status
IO/M = 0, S1 = 1, S0 = 1
Opcode Fetch
IO/M = 0, S1 = 1, S0 = 0
Status
RD

Example :- MVI A, DATA

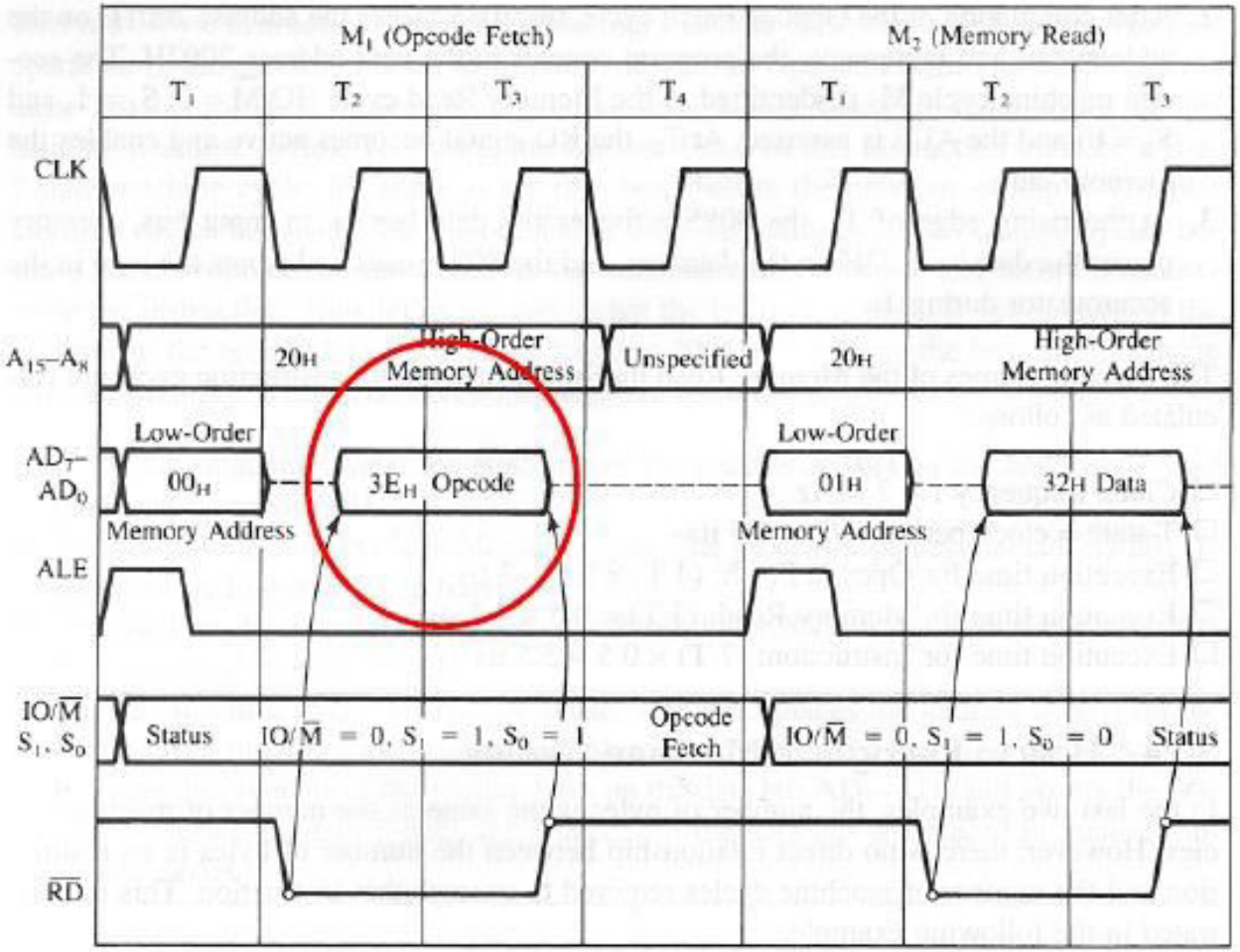
ADDRESS	MNEMONICS	OP-CODE
2000	MVI A,32_H	3E
2001		32

TIMING DIAGRAM OF MVI A,32 H

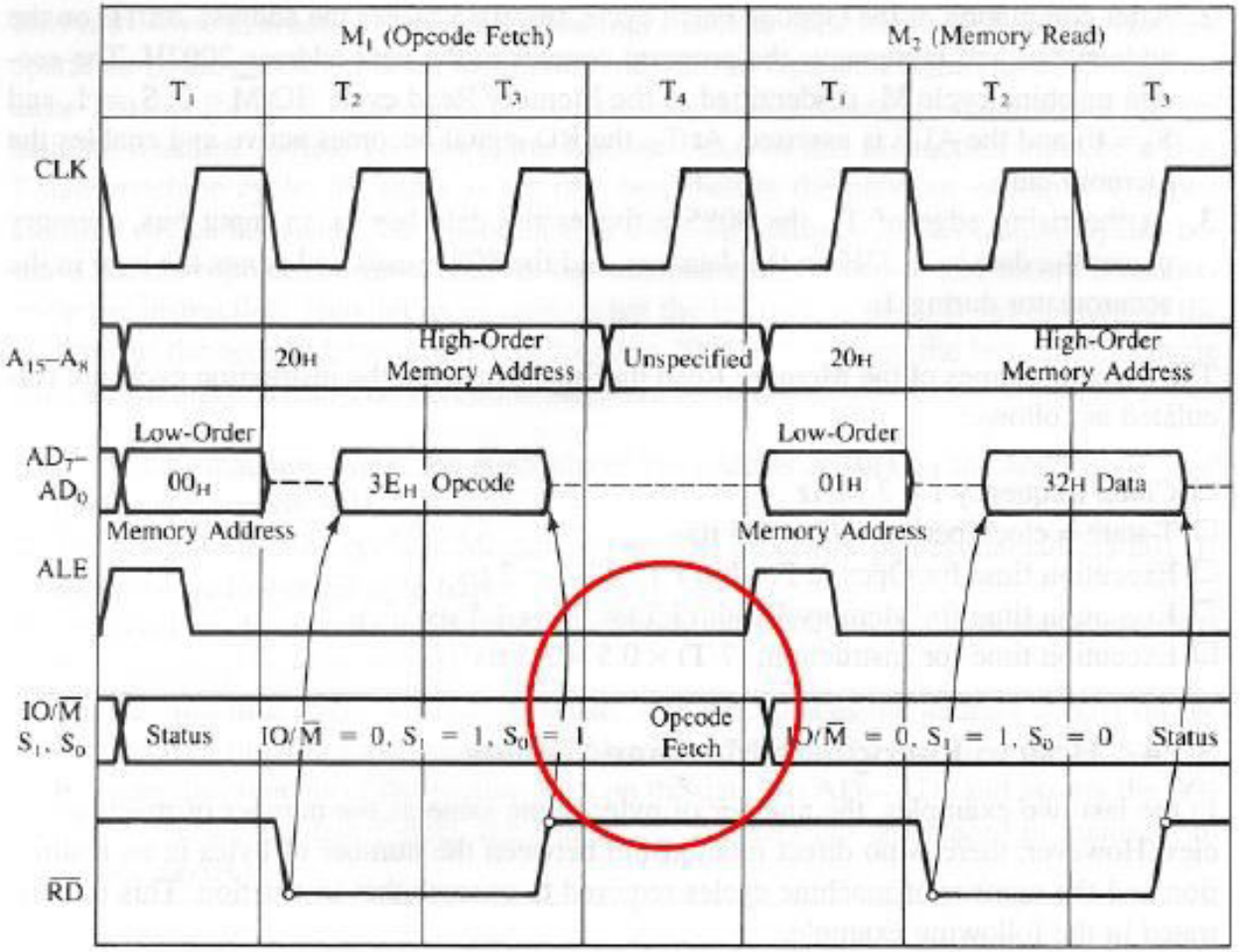




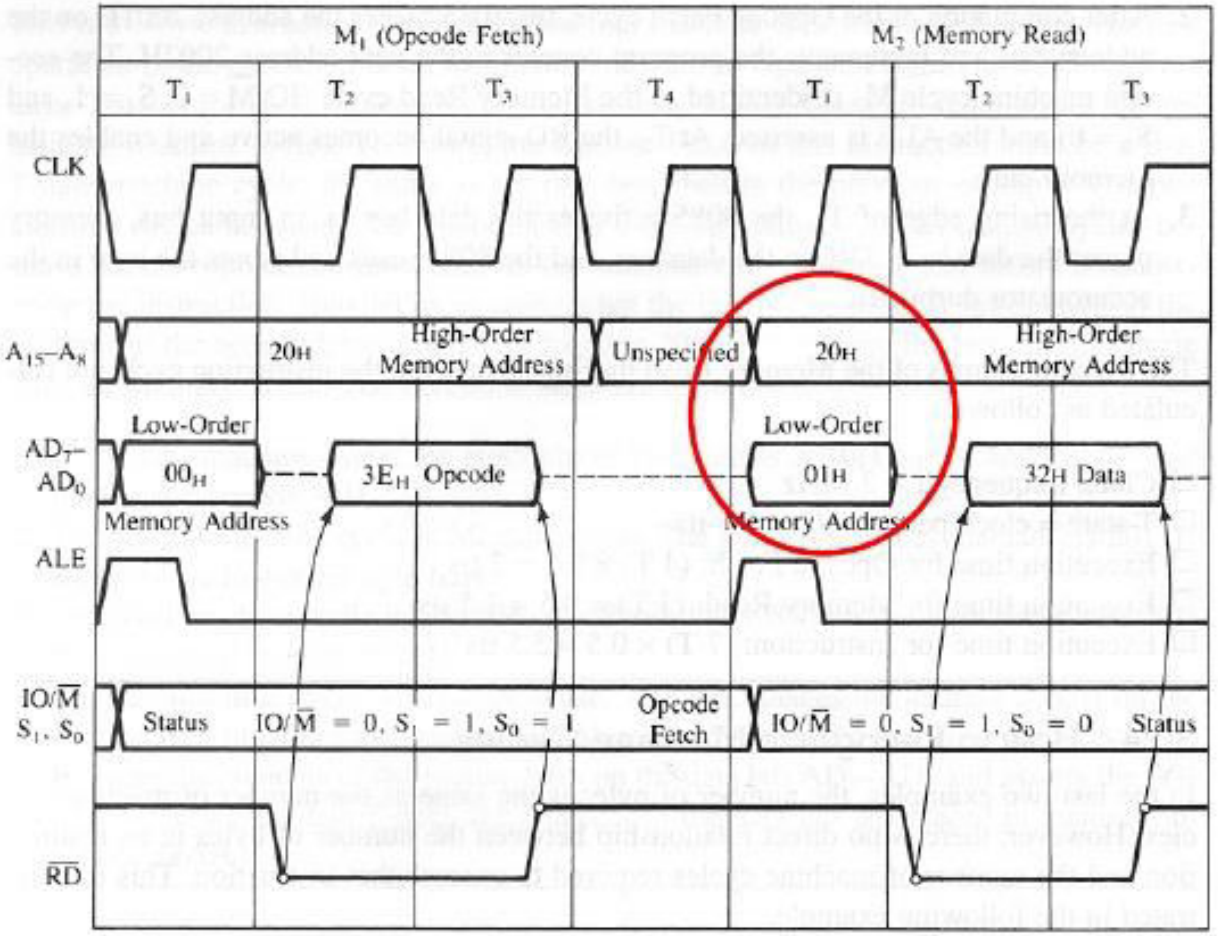
Put the first memory Location on the address bus(2000 H)



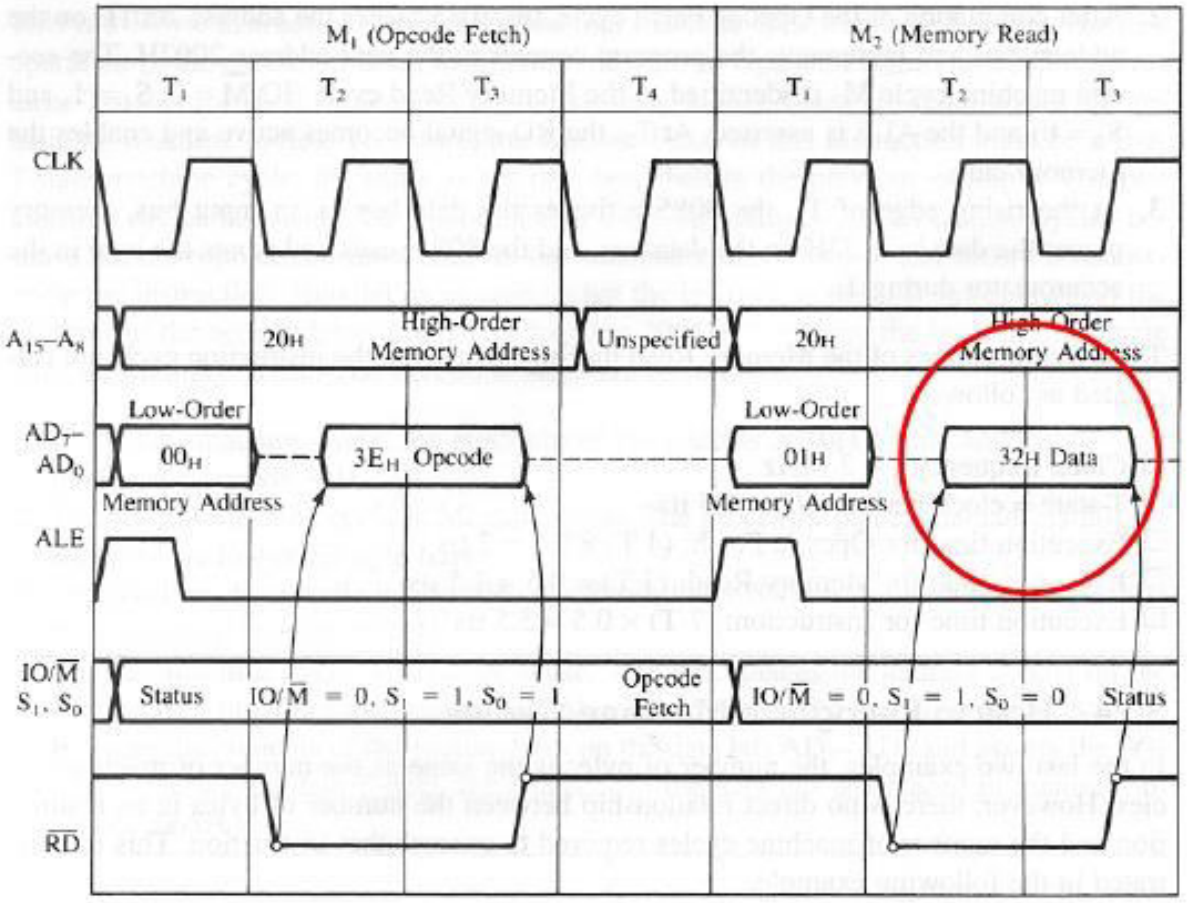
Get the instruction(op-code) byte from memory



Interpret the Instruction : Wait for the data byte



Put the next memory location on the address bus (2001 H)



Get the data byte from the memory

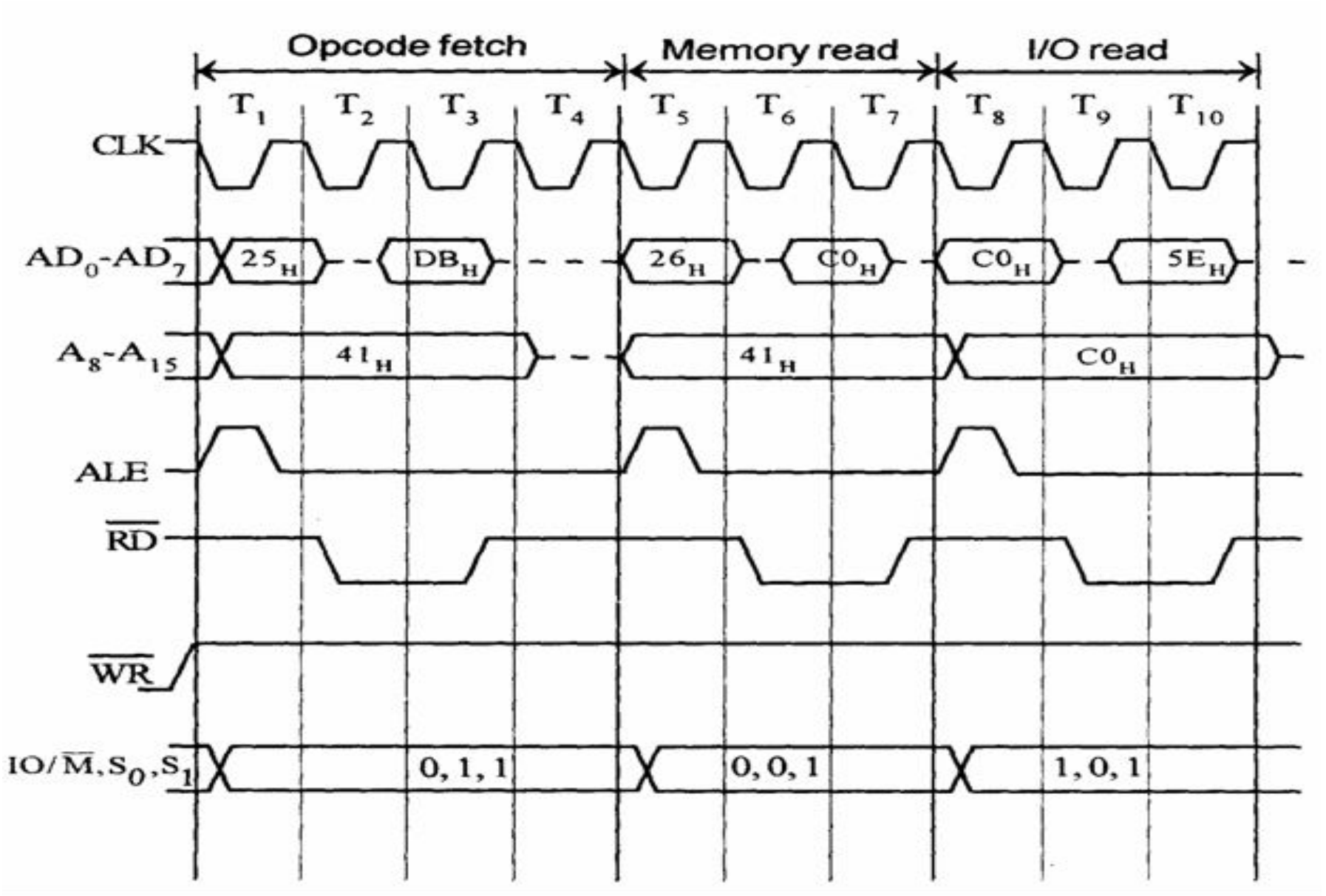
Through data lines
This will be finally placed into accumulator

TIMING DIAGRAM FOR IN port address.

1. Fetching the Op-code DBH from the memory 4125H.
2. Read the port address C0H from 4126H.
3. Read the content of port C0H and send it to the accumulator.
4. Let the content of port is 5EH.

ADDRESS	MNEMONICS	OP-CODE
4125	IN $C0_H$	DB_H
4126		$C0_H$

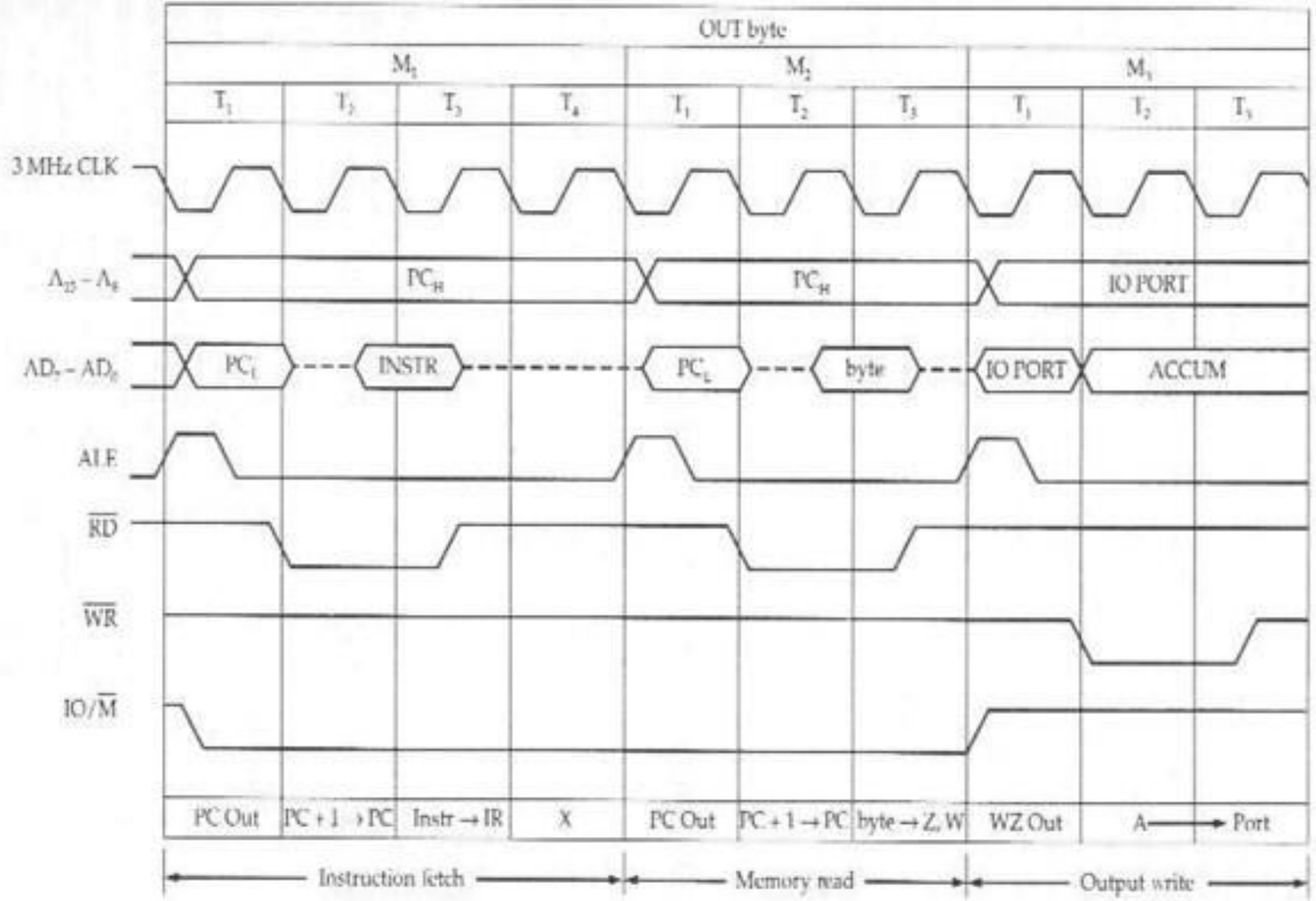
IN C0H


Opcode fetch
Memory read
I/O read
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
CLK
AD0-AD7
25H
DBH
26H
C0H
C0H
5EH
A8-A15
41H
41H
C0H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
1, 0, 1

Timing diagram for OUT 8-bit port address.

ADDRESS	MNEMONICS	OP-CODE
4125	OUT $C0_H$	$D3_H$
4126		$C0_H$

TIMING DIAGRAM FOR OUT 8-bit port address



THREE BYTE INSTRUCTIONS

Timing diagram for STA 526AH:

- STA means Store Accumulator -The contents of the accumulator is stored in the specified address (526A).

- The opcode of the STA instruction is said to be 32H. It is fetched from the memory 41FFH (OF machine cycle)

- Then the lower order memory address is read(6A). - Memory Read Machine Cycle

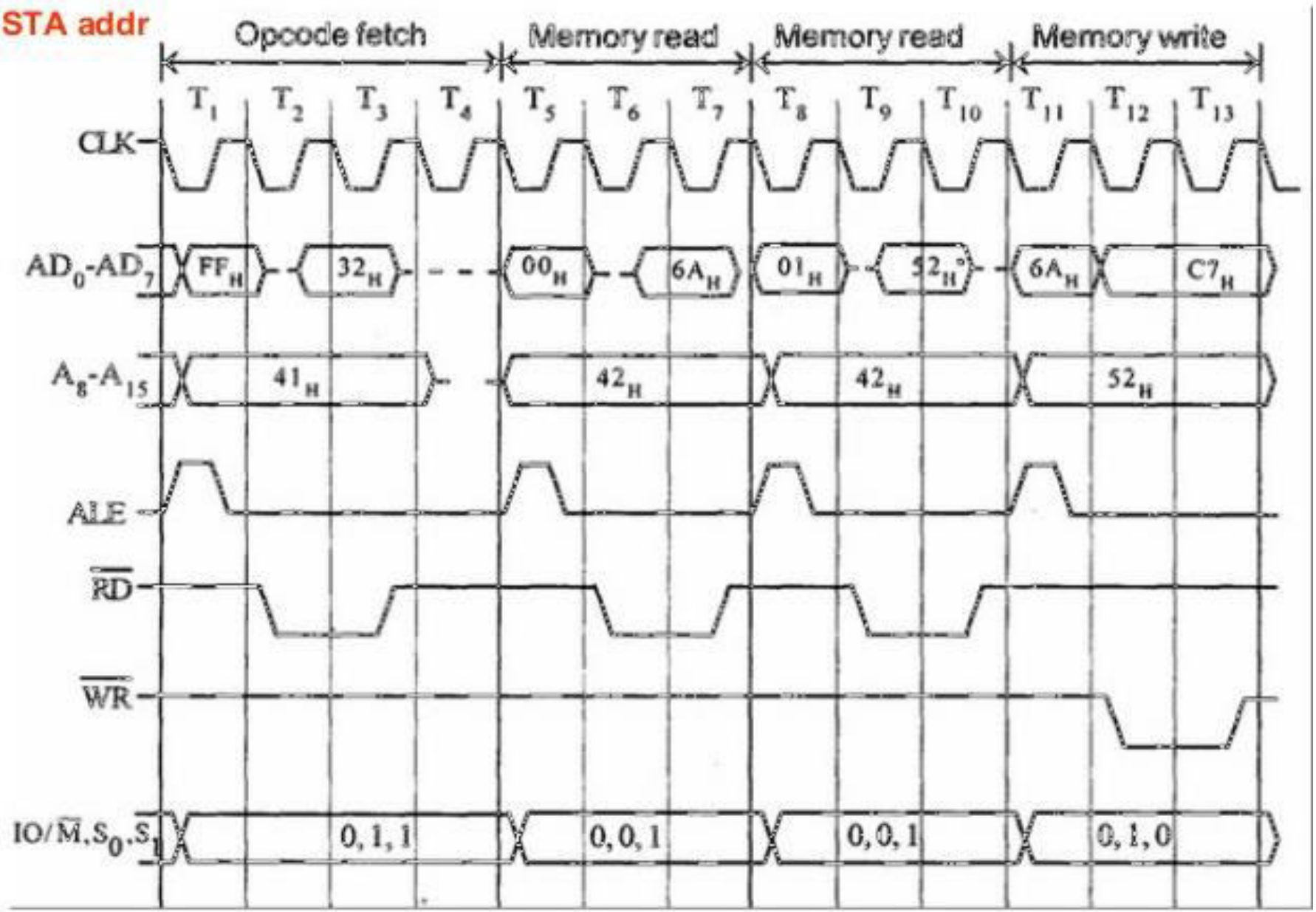
- Read the higher order memory address (52).- Memory Read Machine Cycle

- The combination of both the addresses are considered and the content from accumulatoris written in 526A. - Memory Write Machine Cycle

- Assume the memory address for the instruction and let the content of accumulator isC7H. So C7H from accumulator is now stored in 526A

ADDRESS	MNEMONICS	OP-CODE
41FF	STA $526A_H$	32_H
4200		$6A_H$
4201		52_H

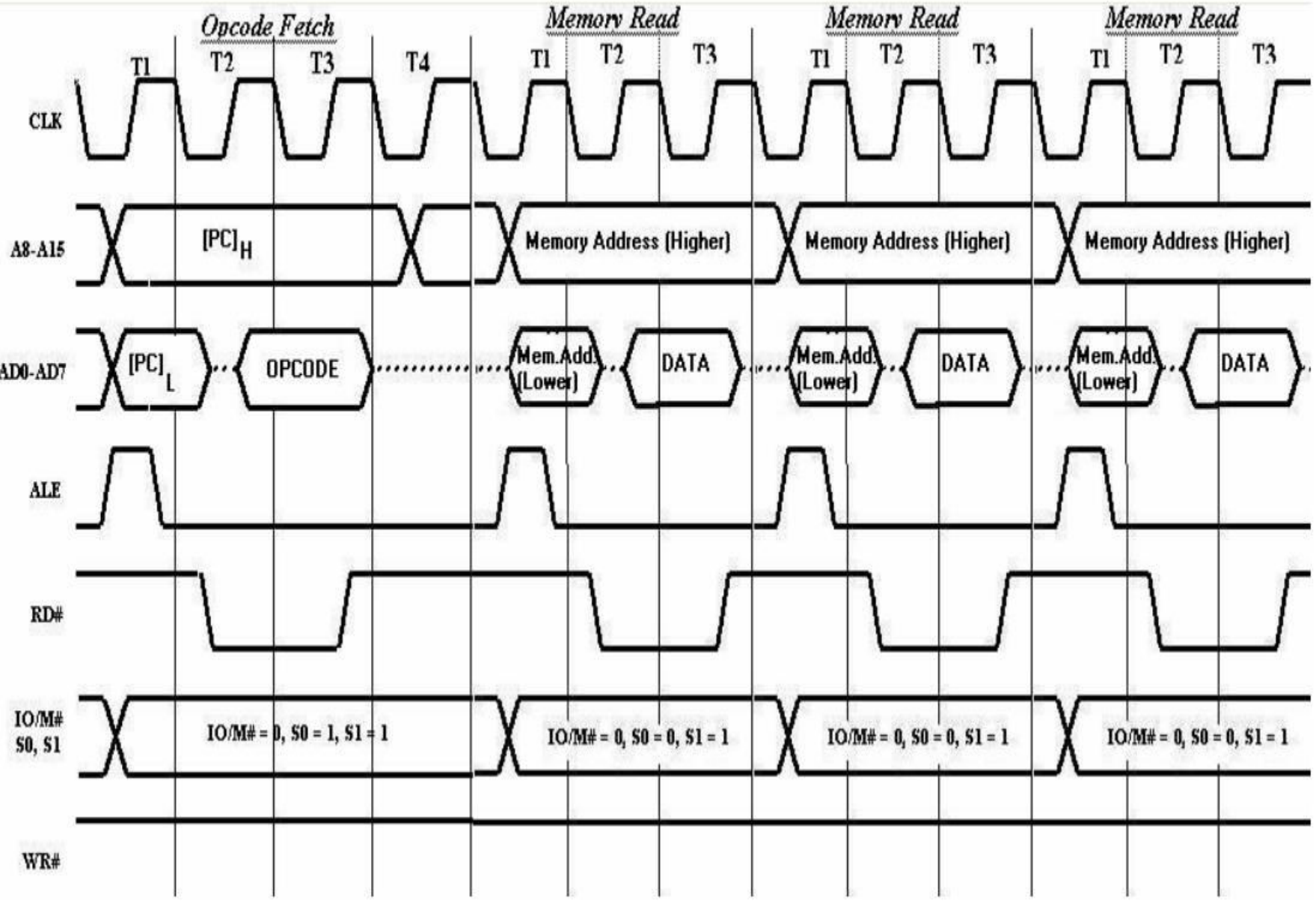
STA addr


Opcode fetch
Memory read
Memory read
Memory write
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13
CLK
AD0-AD7
FFH 32H 00H 6AH 01H 52H 6AH C7H
A8-A15
41H 42H 42H 52H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
0, 0, 1
0, 1, 0

Timing diagram for LDA 16-BIT ADDRESS

ADDRESS	MNEMONICS	OP-CODE
4000	LDA 2000	3A
4001		00
4002		20

LDA 16bit Addr



Timing diagram for CALL ADDRESS

CALL address

This instruction is used to transfer program control to a subprogram or subroutine. This instruction pushes the current program counter contents onto the stack and loads the given address into the program counter. It requires five machine cycles as explained below.

1) Opcode fetch : Program counter places address on low order and high order address bus. Microprocessor reads the opcode of CALL (CDH) from this memory address and decodes it. Program counter is incremented by one. This machine cycle requires 6 T-states.

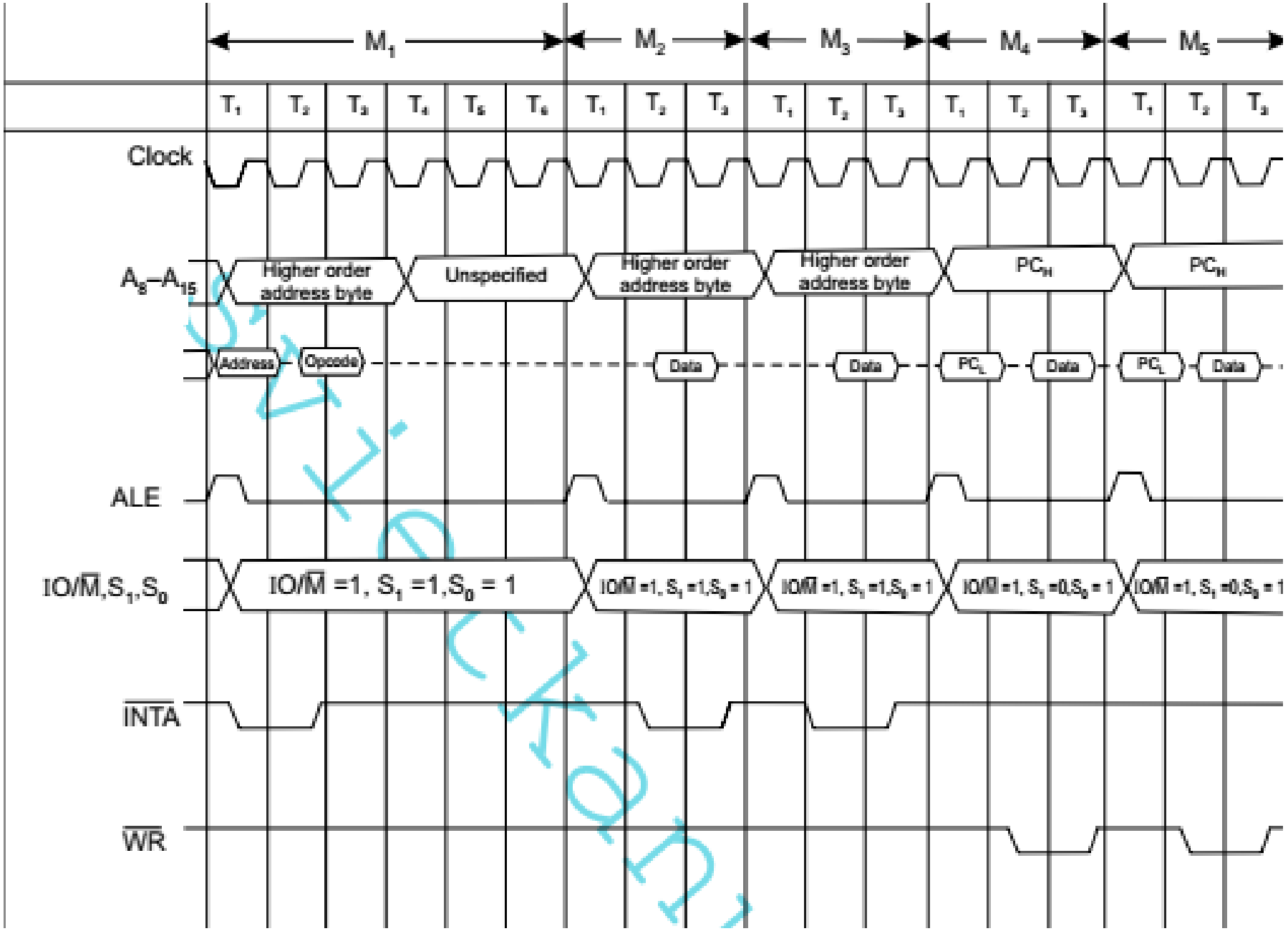
2) Memory read : Program counter gives address on low order and high order address bus. Microprocessor reads the lower byte of address specified within the instruction from the addressed memory location. Program counter is incremented by one.

3) Memory read : Program counter gives address on low order and high order address bus. Microprocessor reads the higher byte of address specified within the instruction from the addressed memory location. Program counter is incremented by one. Stack pointer is decremented by one.

4) Memory write : Stack pointer places address on low order and high order address bus. Microprocessor writes the high order byte of program counter at this addressed memory location. Stack pointer is again decremented by one.

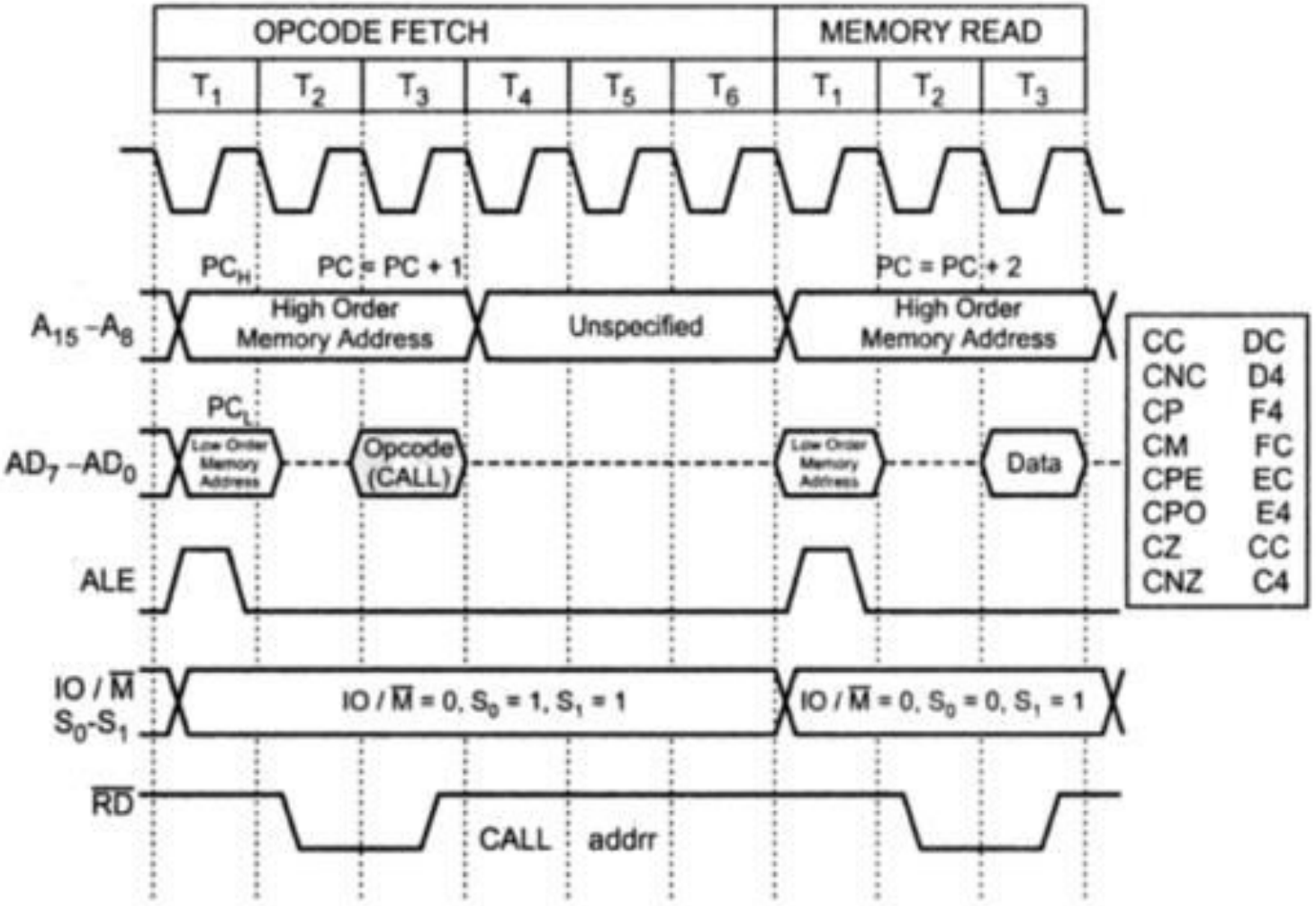
5) Memory write : Stack pointer places address on low order and high order address bus. Microprocessor writes the low order byte of program counter at this addressed memory location.

ADDRESS	MNEMONICS	OP-CODE
4000	CALL 2000	CD
4001		00
4002		20


M1
M2
M3
M4
M5
T1 T2 T3 T4 T5 T6 T1 T2 T3 T1 T2 T3 T1 T2 T3 T1 T2 T3
Clock
A8–A15
Higher order address byte
Unspecified
Higher order address byte
Higher order address byte
PCH
PCH
AD0–AD7
Address
Opcode
Data
Data
PCL
Data
PCL
Data
ALE
IO/M,S1,S0
IO/M =1, S1 =1,S0 = 1
IO/M =1, S1 =1,S0 = 1
IO/M =1, S1 =1,S0 = 1
IO/M =1, S1 =0,S0 = 1
IO/M =1, S1 =0,S0 = 1
INTA
WR

C condition (when condition is not valid)

This instruction transfers program control to the next instruction written after this instruction. It is similar to J condition when condition is not valid.



Timing diagram for LHLD 16-BIT ADDRESS

LHLD address

This instruction loads L register with the contents of memory location given within the instruction and loads H register with the contents of memory location at address next to it. It requires the following five machine cycles.

---

1) Opcode fetch : Program counter places address on low order and high order address bus. Microprocessor reads the opcode of LHLD (2AH) from this memory location and decodes it. Program counter is incremented by one.

2) Memory read : Program counter gives address on low order and high order address bus. Microprocessor reads data from the addressed memory location. This data is the low order byte of the address specified within the instruction. Program counter is incremented by one.

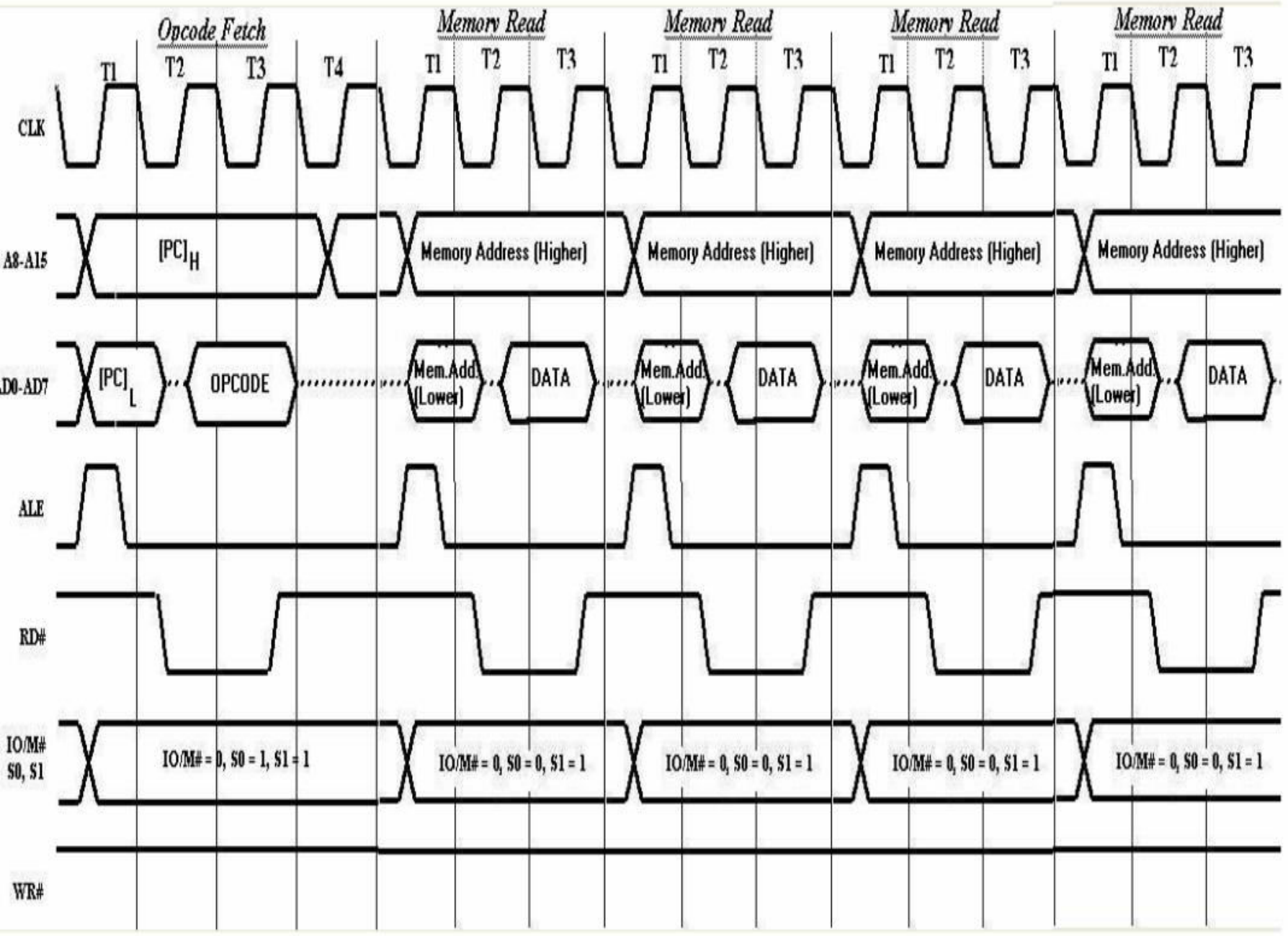
3) Memory read : Program counter gives address on low order and high order address bus. Microprocessor reads data from the addressed memory location. This data is the high order byte of the address specified within the instruction.

4) Memory read : The data read in the previous two memory read cycles is placed on the address bus. Microprocessor reads the contents of memory location and loads it in L register. This memory address is incremented by one.

5) Memory read : Now the incremented address is present on the address bus. Microprocessor reads the contents of memory location and loads it in H register.

ADDRESS	MNEMONICS	OP-CODE
4000	LHLD 2000	2A
4001		00
4002		20

LHLD 16bit Addr



Timing diagram for SHLD 16-BIT ADDRESS

ADDRESS	MNEMONICS	OP-CODE
4000	SHLD 2000	22
4001		00
4002		20

SHLD address

This instruction stores the contents of L register in the memory location given within the instruction and contents of H register at address next to it. It requires the following five machine cycles.

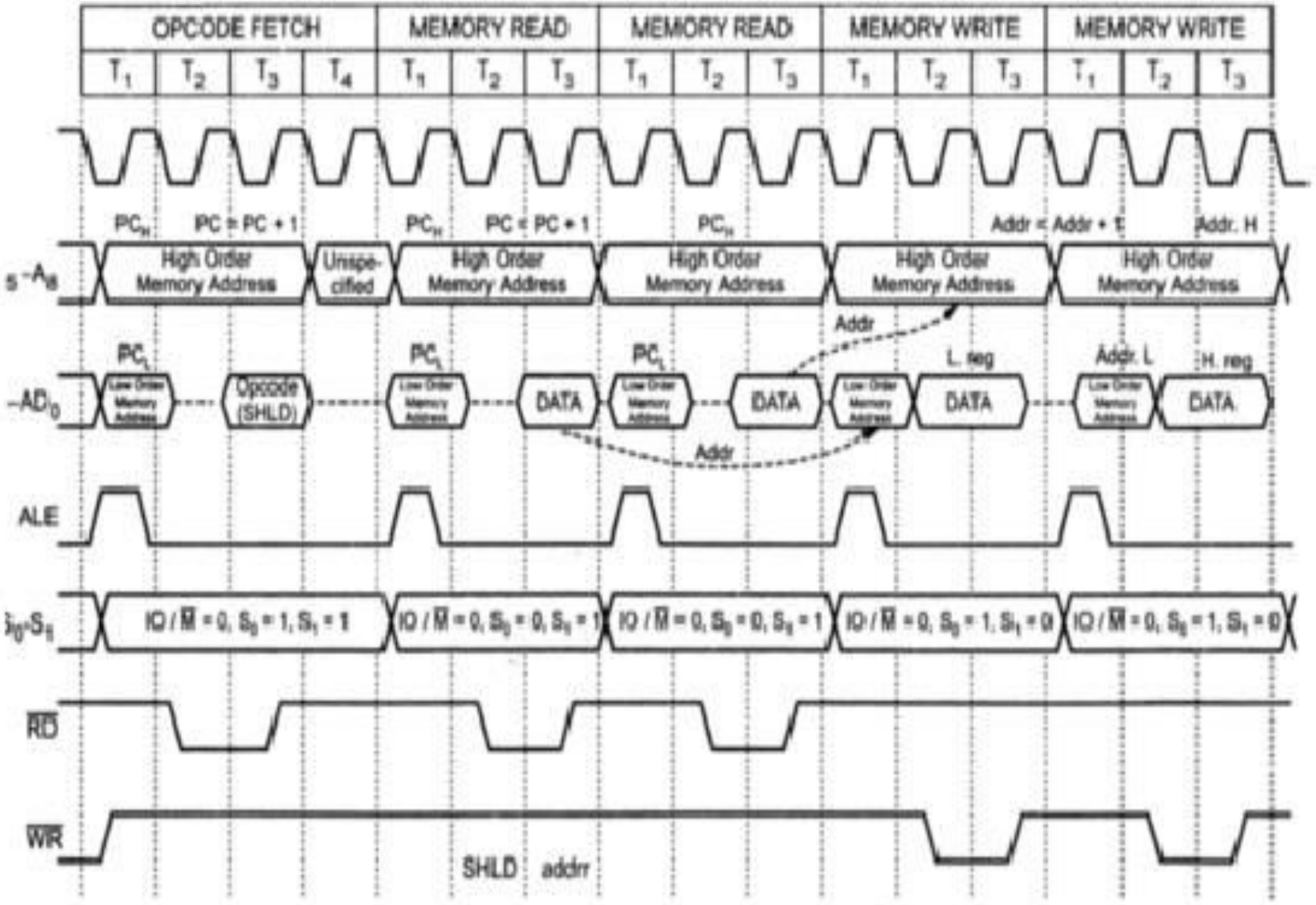
1) Opcode fetch : Program counter places address on low order and high order address bus. Microprocessor reads the opcode of SHLD (22H) from this memory location and decodes it. Program counter is incremented by one.

2) Memory read : Program counter gives address on low order and high order address bus. Microprocessor reads data from the addressed memory location. This data is the low order byte of the address specified within the instruction. Program counter is incremented by one.

3) Memory read : Program counter gives address on low order and high order address bus. Microprocessor reads data from the addressed memory location. This data is the high order byte of the address specified within the instruction.

4) Memory write : The data read in the previous two memory read cycles is placed on the address bus. Microprocessor writes the contents of L register at this memory address. This memory address is incremented by one.

5) Memory write : Now the incremented address is present on the address bus. Microprocessor writes the contents of H register at this memory address.


OPCODE FETCH
MEMORY READ
MEMORY READ
MEMORY WRITE
MEMORY WRITE
T1 T2 T3 T4 T1 T2 T3 T1 T2 T3 T1 T2 T3 T1 T2 T3
PCH
PC = PC + 1
PCH
PC = PC + 1
PCH
Addr = Addr + 1
Addr. H
High Order Memory Address
Unspecified
High Order Memory Address
High Order Memory Address
High Order Memory Address
High Order Memory Address
PCL
Opcode (SHLD)
PCL
DATA
PCL
DATA
Addr
L. reg
DATA
Addr. L
H. reg
DATA
Addr
ALE
IO / M = 0, S0 = 1, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
IO / M = 0, S0 = 0, S1 = 1
IO / M = 0, S0 = 1, S1 = 0
IO / M = 0, S0 = 1, S1 = 0
RD
WR
SHLD addrr

Timing diagram for JMP 16-BIT ADDRESS

JMP address

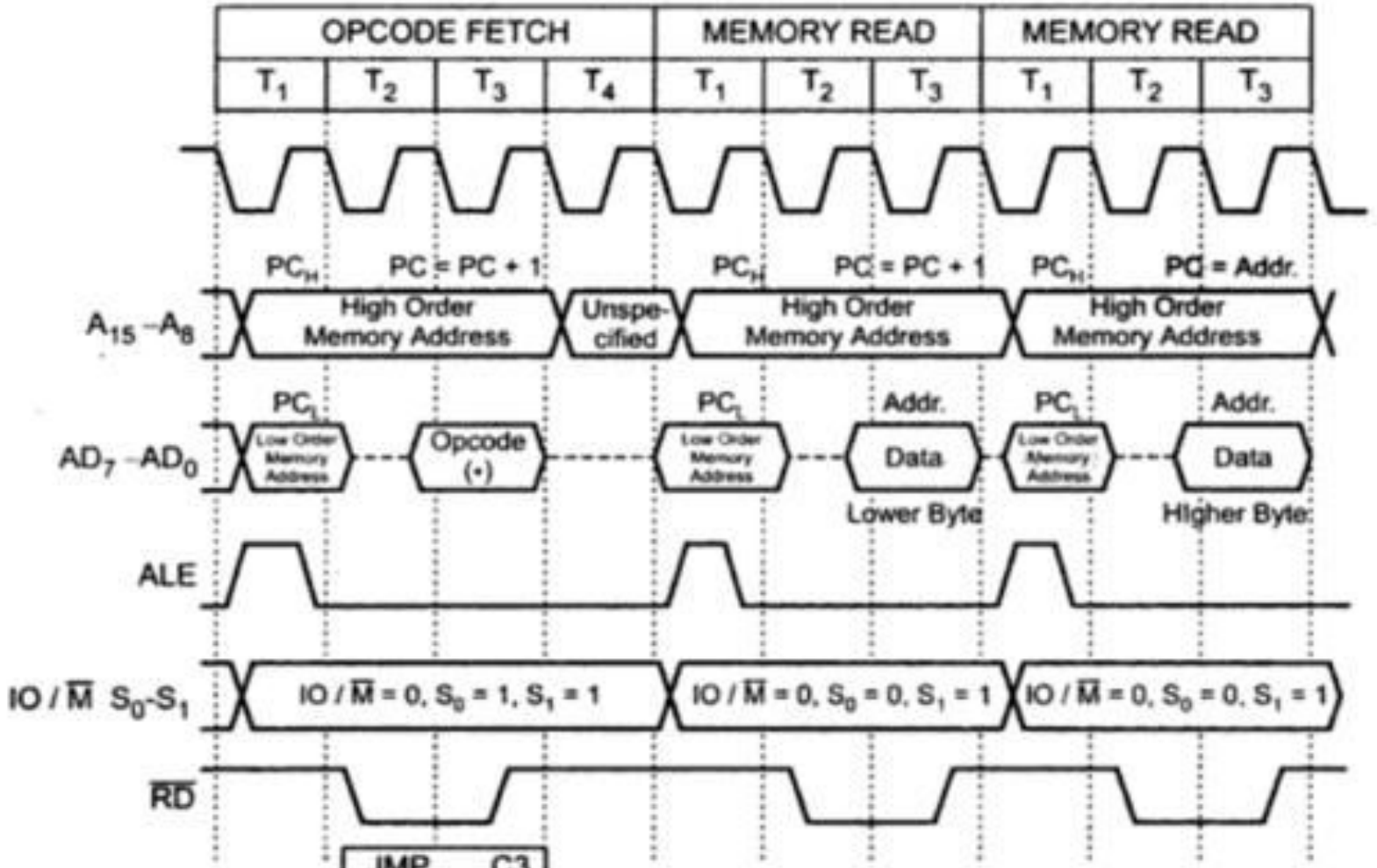
This instruction loads the program counter with the address given within the instruction and resumes the program execution from this location. It requires three machine cycles as explained below.

1) Opcode fetch : Program counter places address on low order and high order address bus. The opcode of JMP (C3H) is read into the microprocessor from this address and is decoded. Program counter is incremented by one.

2) Memory read : Program counter gives address on low order and high order address bus. The data at this addressed memory location is read into the microprocessor. This data is nothing but the low order byte of the address specified within the instruction. Program counter is incremented by one.

3) Memory read : Program counter gives address on low order and high order address bus. The data at this addressed memory location is read into the microprocessor. This data is nothing but the high order byte of the address specified within the instruction. The data read into the microprocessor in these two memory read cycles is loaded into the program counter. So the program execution starts from the address specified within the instruction.

The timing diagram of J condition, when condition is valid is same as that of JMP address.

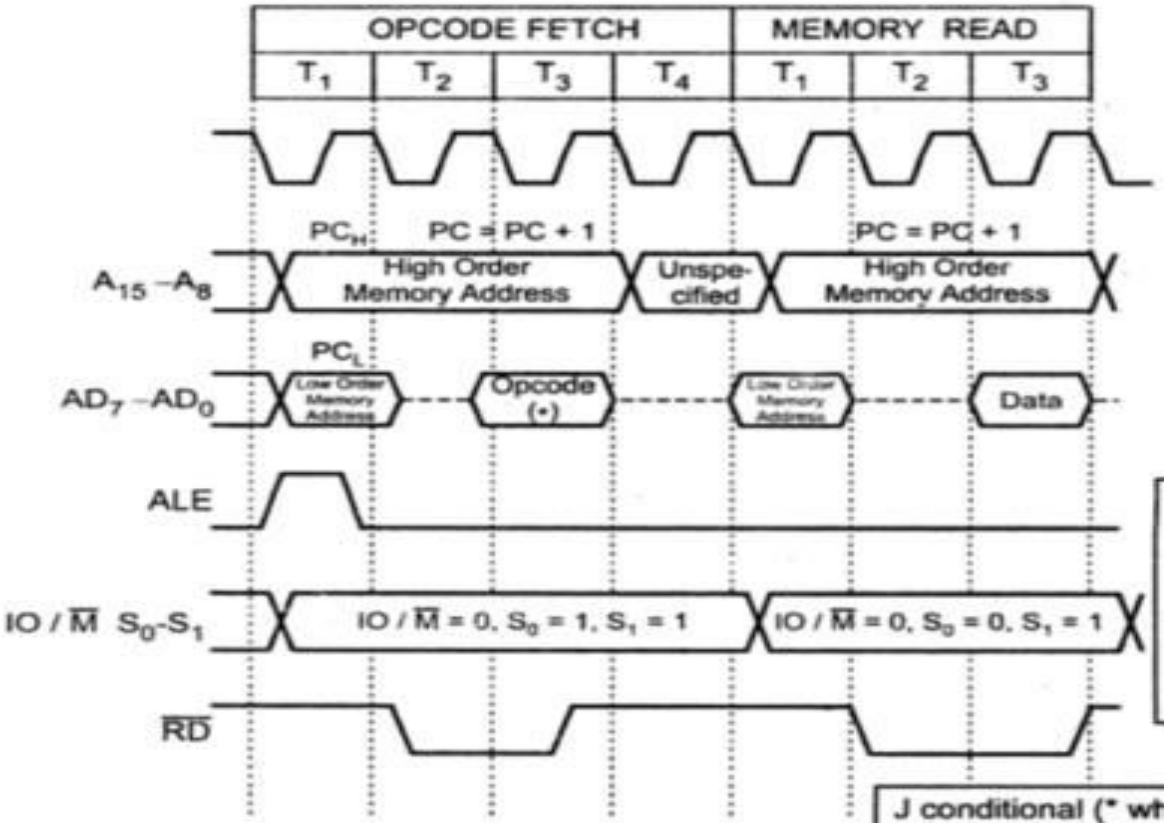


JMP	C3
JZ	CA
JNZ	C2
JC	DA
JNC	D2
JP	F2
JM	FA
JPE	EA
JPO	E2

JMP J conditional (when condiiton is valid) * for instruction JNZ addr, if zero flag is not set at the time of execution of instruciton condition is valid.

J condition (when condition is not valid)

This instruction transfers program control to the next instruction written after this instruction. Due to fetch execution overlap flag are checked in the T-state of the next machine cycle. Hence it is not possible for 8085 to decide whether to jump or not at the given address after opcode fetch machine cycle. As a result, it executes memory read machine cycle. At T1 of this machine cycle 8085 checks the necessary flag and goes to the next instruction if condition is not true.



*JZ	CA
JNZ	C2
JC	DA
JNC	D2
JP	F2
JM	FA
JPE	EA
JPO	E2

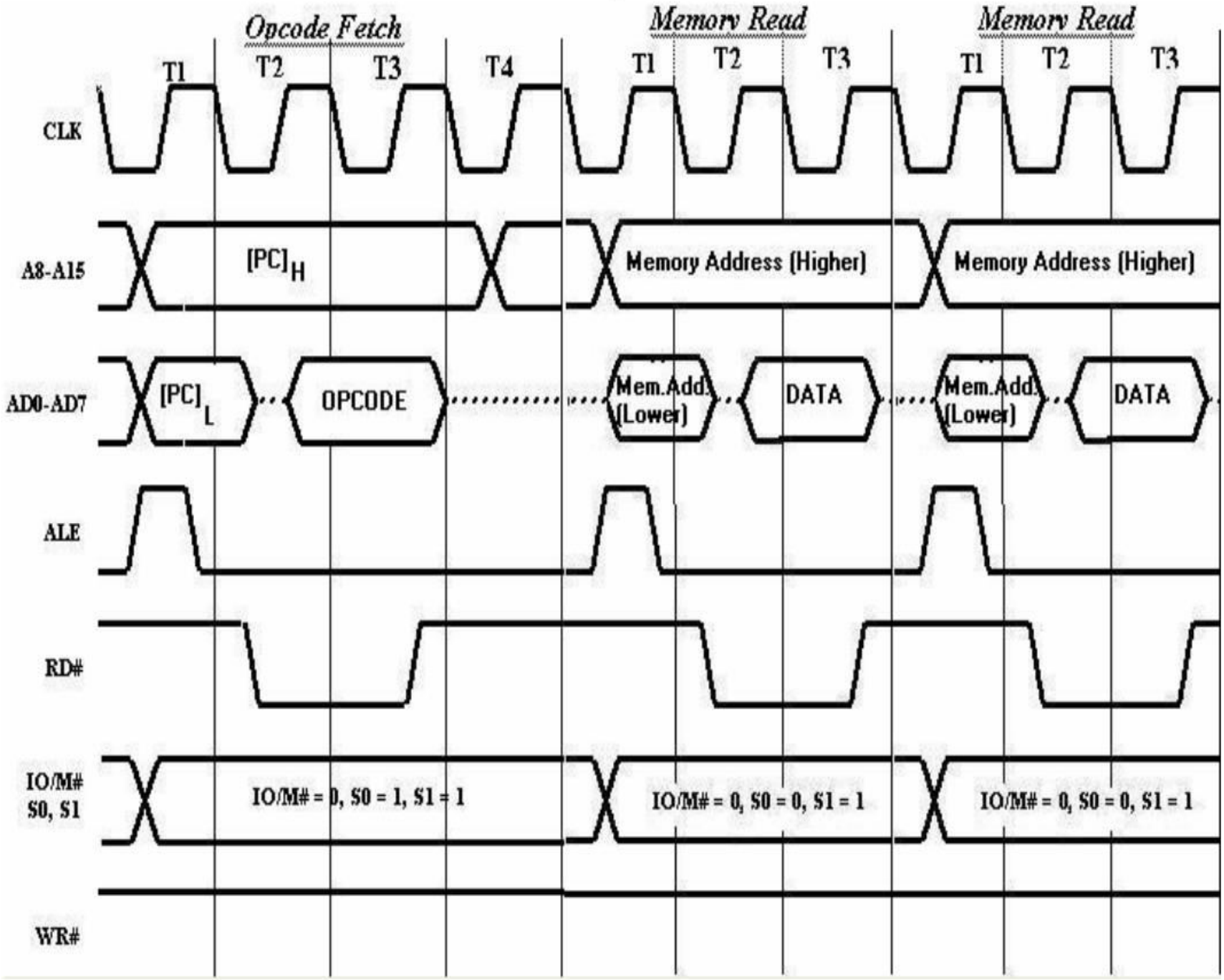
Table A.12

J conditional (* when condition is not valid)* for instruction JNZ address, if zero flag is set at the time of execution of inst-ruction condition is not valid.

EXAMPLE :-

ADDRESS	MNEMONICS	OP-CODE
4000	JMP 2000	C3
4001		00
4002		20

JMP 16bit Addr



Timing diagram for LXI R_P, 16-BIT DATA

ADDRESS	MNEMONICS	OP-CODE
4000	LXI B, 2050	01
4001		50
4002		20

three machine cycles are required to fetch the instruction. As it is an immediate instruction, operand i.e. immediate 16-bit data is given within the instruction, no further machine cycle is required.

Machine cycles required for LXI instruction.