Microprocessor

Chapter 3: Instruction Cycle

Prepared by
Er. Sachita Nand Mishra

Contents

- Instruction Cycle, Machine Cycle and T-states
  - ✓Machine Cycle of 8085 Microprocessor: op-code fetch, memory read, memory write, I/O read, I/O write, interrupt
- Fetch and Execute Operation, Timing Diagram
  - ✓Timing Diagram of MOV, MVI, IN, OUT, LDA, STA
- Memory Interfacing and Generation of Chip Select Signal

Instruction cycle

- Instruction: A command given to the microprocessor to perform an operation
- Program : A set of instructions given in a sequential manner to perform a particular task

The CPU fetches one instruction from memory at a time & executes it.

Instruction cycle

- Time required to execute and fetch an entire instruction is called ***instruction cycle***.
- It consists:

✓ **Fetch cycle:** The next instruction is fetched by the address stored in program counter (PC) and then stored in the instruction register.

✓ **Decode instruction:** Decoder interprets the encoded instruction from instruction register.

✓ **Reading effective address:** The address given in instruction is read from main memory and required data is fetched. The effective address depends on direct addressing mode or indirect addressing mode.

✓ **Execution cycle:** Consists memory read (MR), memory write (MW), input output read (IOR) and input output write (IOW)

- In the normal process of operation, the microprocessor fetches (receives or reads) and executes one instruction at a time in the sequence until it executes the halt (HLT) instruction.

Instruction cycle

- Thus, an instruction cycle is defined as the time required to fetch and execute an instruction.

  Instruction cycle = Fetch cycle + Execute cycle

Fetch Cycle : The steps taken by CPU to fetch the opcode from the memory

✓ The time taken for fetch cycle is fixed.

Execute Cycle : The steps taken by CPU to perform the operation specified in the instruction

✓ The time taken for execute cycle is variable which depends on the type of instruction ,i.e. 3 –byte , 2-byte & 1-byte instruction.

Machine cycle

- The time required by the micro processor to complete the operation of accessing memory or I/O device .
- Each Read/Write operation constitutes one machine cycle.

Operations like :

- Opcode fetch
- Memory read
- Memory write
- I/O read
- I/O write

T - states

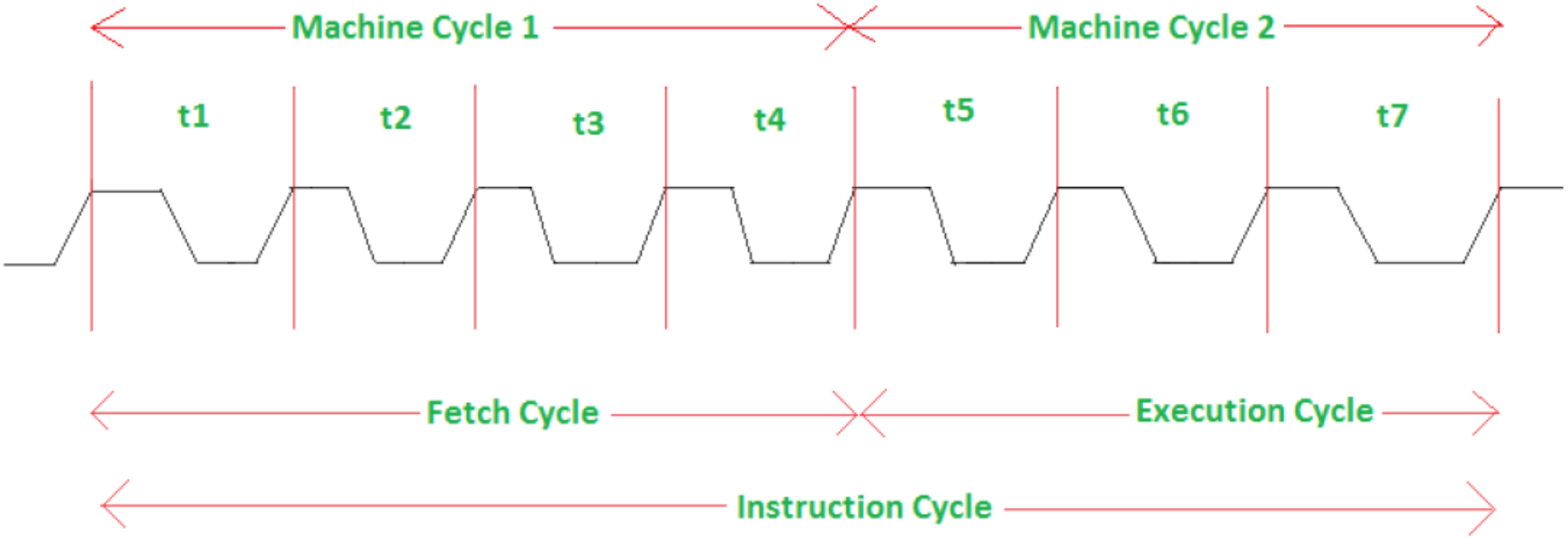
- Microprocessor performs an operation in specific clock cycles.

Each clock cycle is called as T –States.

- The number of T – states required to perform an operation is called Machine Cycle .

T - States

- One-time period of frequency of microprocessor is called **T-State**.
- A t-state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse.
- Fetch cycle takes four t-states and execution cycle takes three t-states. It is shown below:



T - States

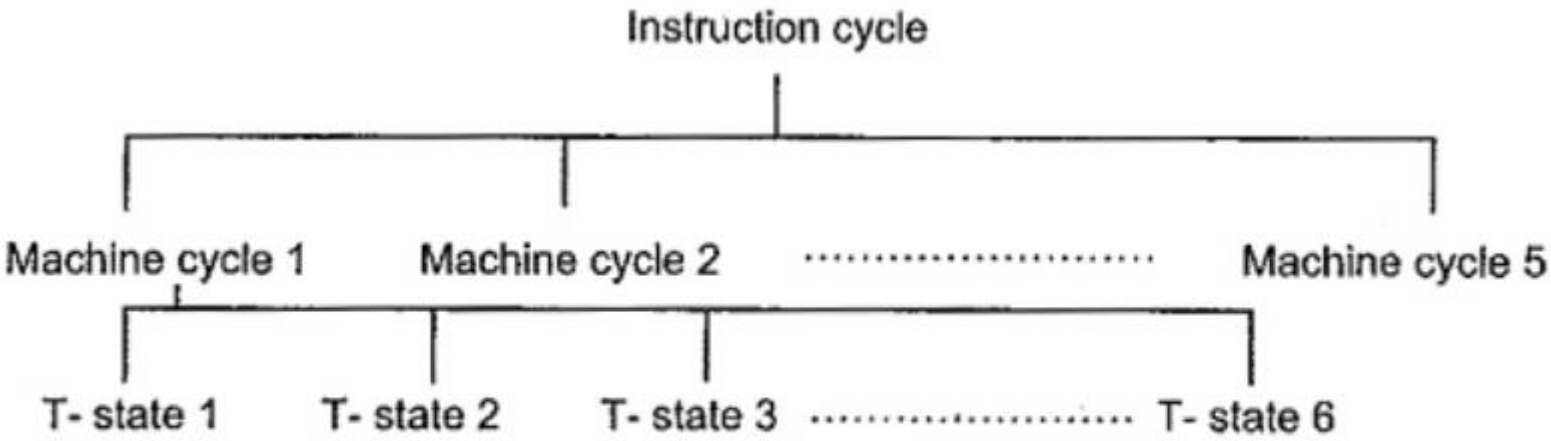


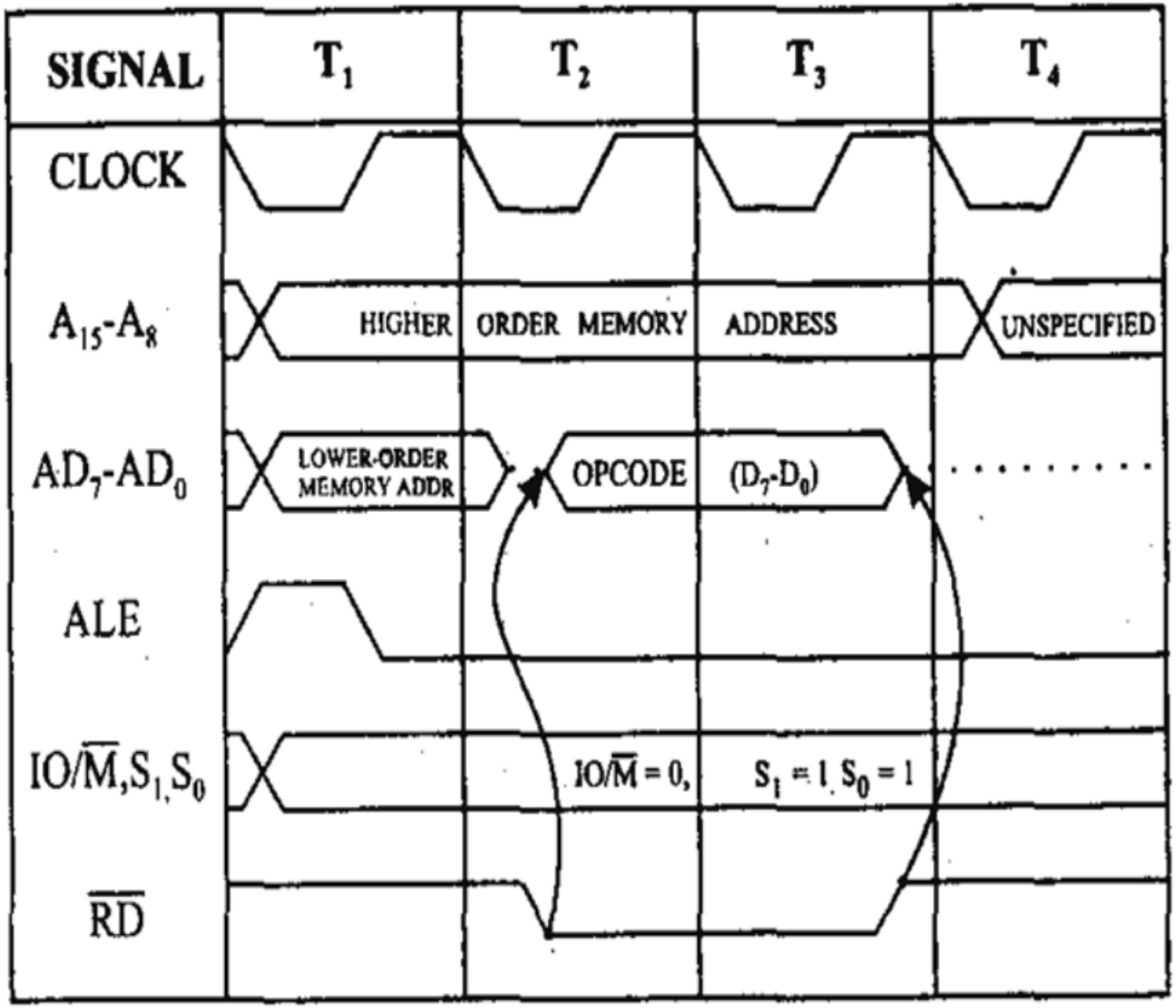
Fig. Relation between instruction cycle, machine cycle and T-state

Machine Cycles of 8085 Microprocessor

Opcode Fetch Cycle

- The first machine cycle of every instruction is **opcode fetch cycle** in which the 8085 finds the nature of the instruction to be executed.
- Each instruction of the processor has one byte opcode.
- The opcodes are stored in memory. So, the processor executes the opcode fetch machine cycle to fetch the opcode from memory. Hence, every instruction starts with opcode fetch machine cycle.
- The time taken by the processor to execute the opcode fetch cycle is 4T.
- In this time, the first, 3 T-states are used for fetching the opcode from memory and the remaining T-states are used for internal operations by the processor.

Opcode Fetch Cycle



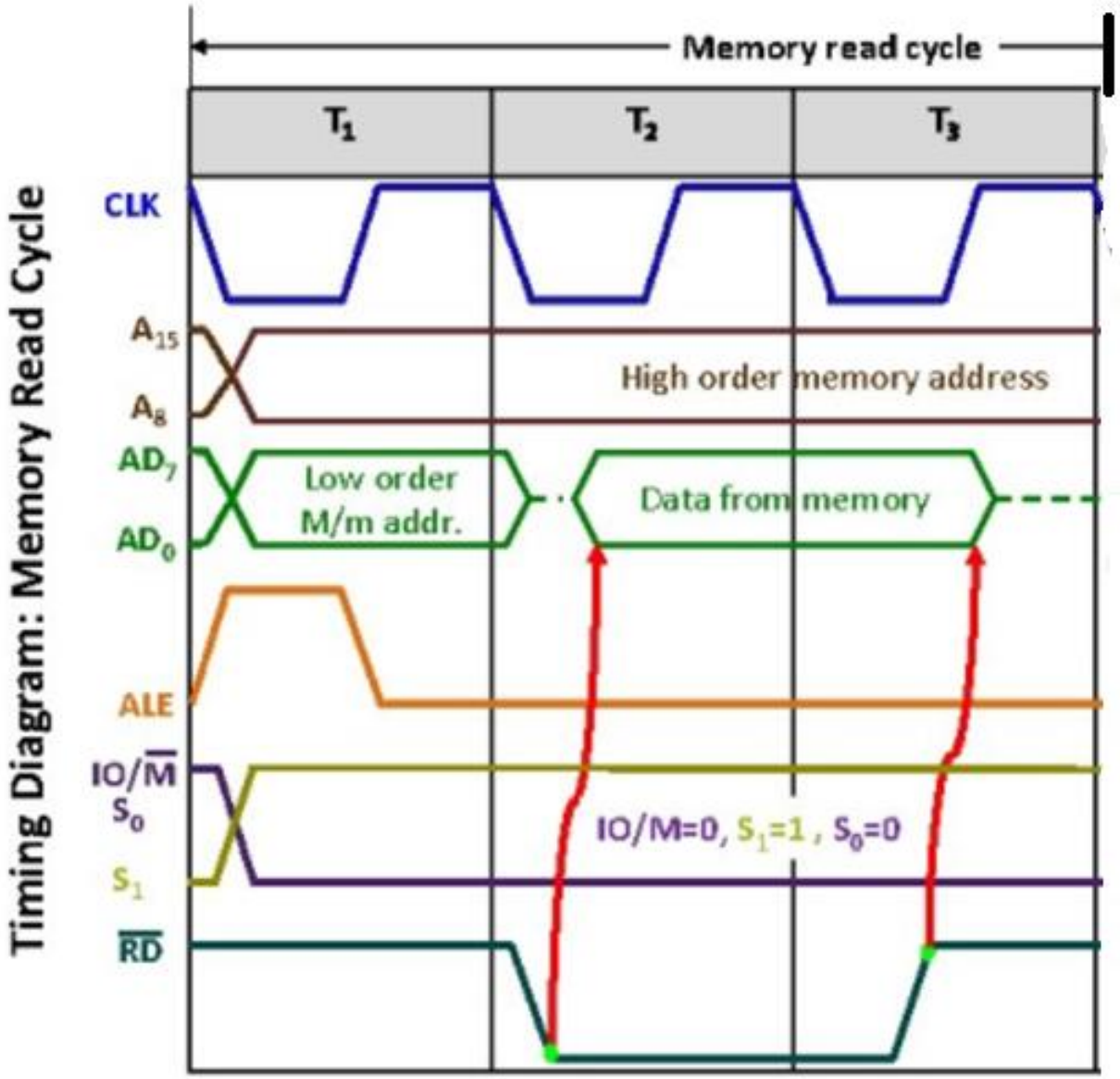
Opcode Fetch Cycle

- The op-code fetch timing diagram can be explained as below:
- The MP places the 16-bit memory address from the program counter on address bus. At time period T1, the higher order memory address is placed on the address lines A15 – A8. When ALE is high, the lower address is placed on the bus AD7 – AD0. The status signal IO/M(bar) goes low indicating the memory operation and two status signals S1 = 1, S0 = 1 to indicate op-code fetch operation.
- At time period T2, the MP sends RD(bar) control line to enable the memory read. When memory is enabled with RD(bar) signal, the op-code value from the addressed memory location is placed on the data bus with ALE low.
- The op-code value is reached at processor register during T3 time period. When data (op-code value) is arrived, the RD(bar) signal goes high. It causes the bus to go into high impedance state.
- The op-code byte is placed in instruction decoder of MP and the op-code is decoded and executed. This happens during time period T4.

Memory Read Cycle

- The 8085 executes the memory read cycle to read the contents of R/W memory or ROM.
- The length of this machine cycle is 3-T states (T1 – T3).
- Single byte instructions require only Opcode Fetch machine cycles. But, 2-byte and 3-byte instructions require additional machine cycles to read the operands from memory. The additional machine cycle is called Memory Read machine cycle.
- For example, the instruction MVI A, 50H requires one OF machine cycle to fetch the operand from memory and one MR machine cycle to read the operand (50H) from memory..
- In this machine cycle, processor places the address on the address lines from the stack pointer, general purpose register pair or program counter, and through the read process, reads the data from the addressed memory location.

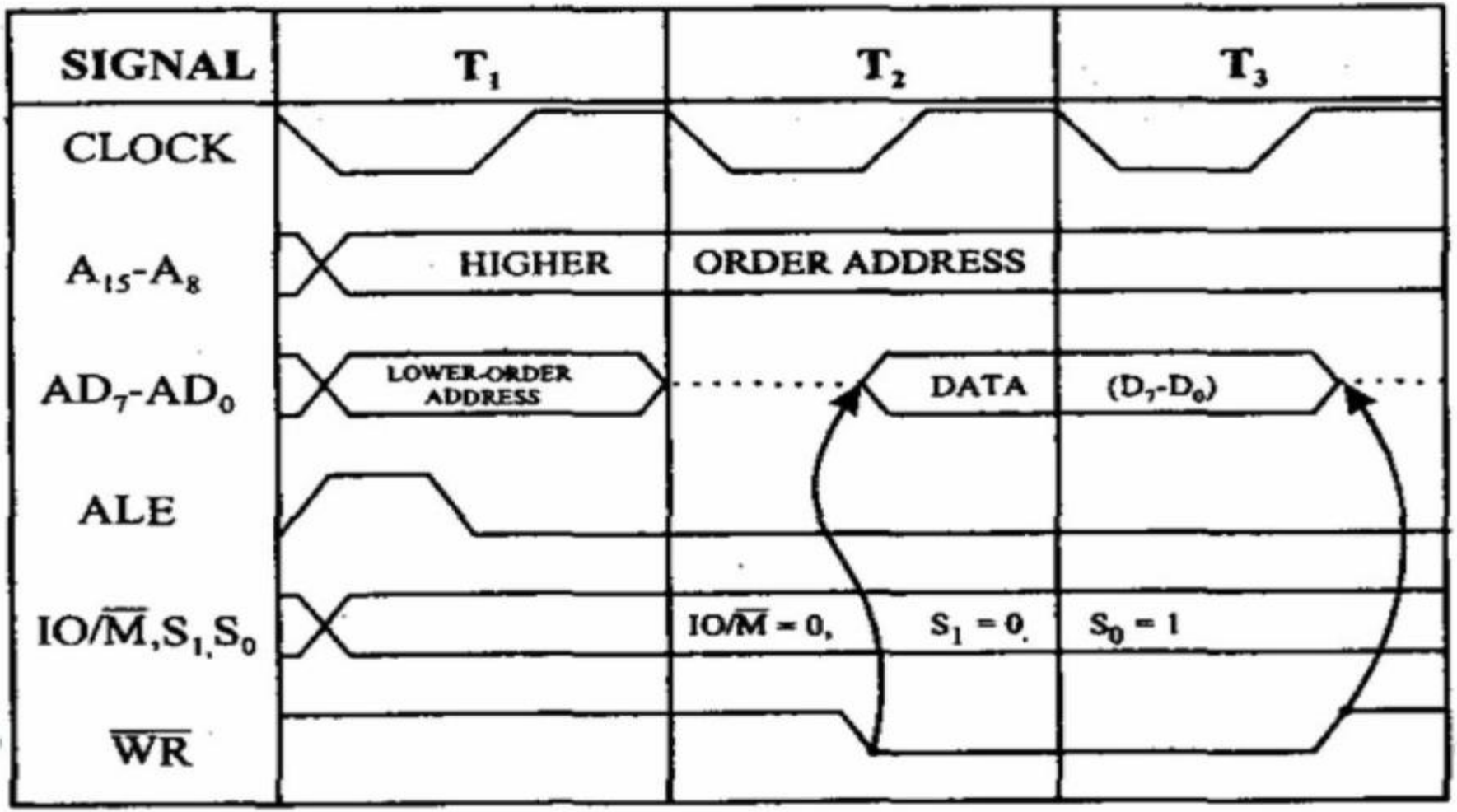
Memory Read Cycle



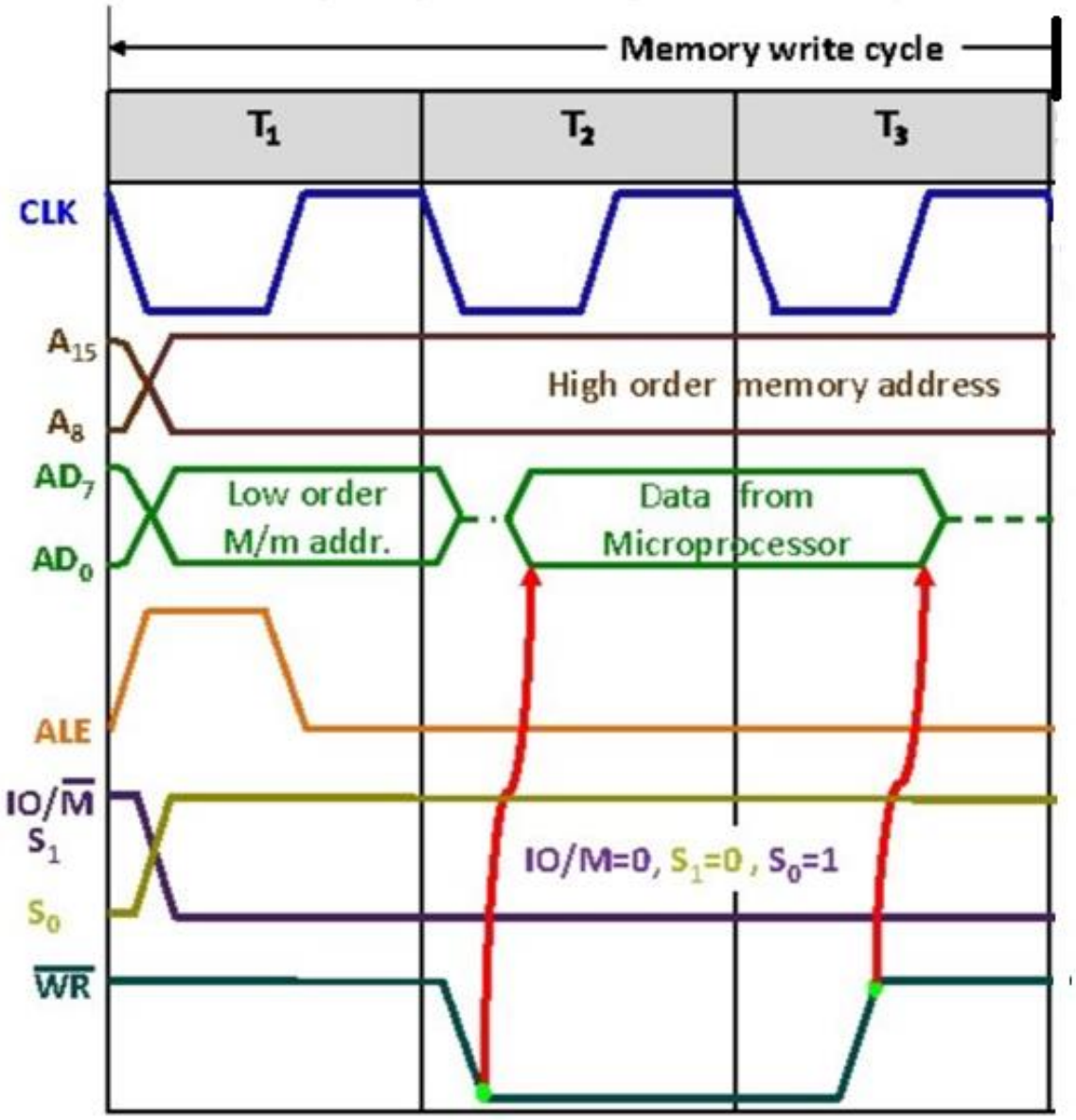
Memory write cycle

- The MP executes memory write machine cycles to write data in a memory location.
- The address of memory is given by instructions.
- The time required to complete the memory write cycle is 3 T-states.

Memory write cycle



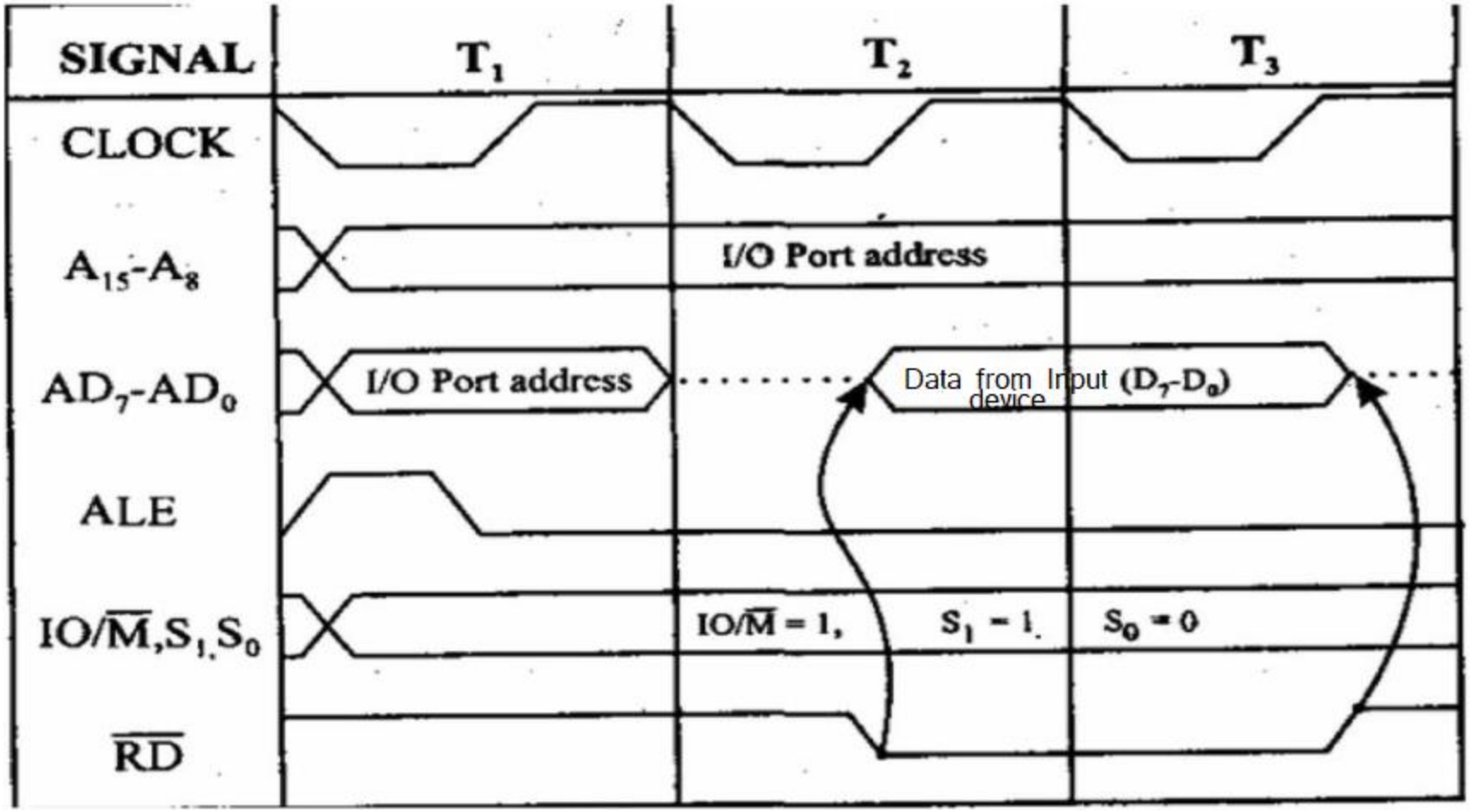
Memory write cycle



I/O Read Cycle

- The MP executes I/O Read cycles to read data from I/O devices or from peripheral.
- The time required to complete the I/O read cycle is 3 T-states.
- Microprocessor uses the I/O Read machine cycle for receiving a data byte from the I/O port or from the peripheral in I/O mapped I/O systems.
- The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
- The data loaded on the address / data bus is moved to the microprocessor i.e., to the accumulator.
- The IN instruction uses this machine cycle during execution.

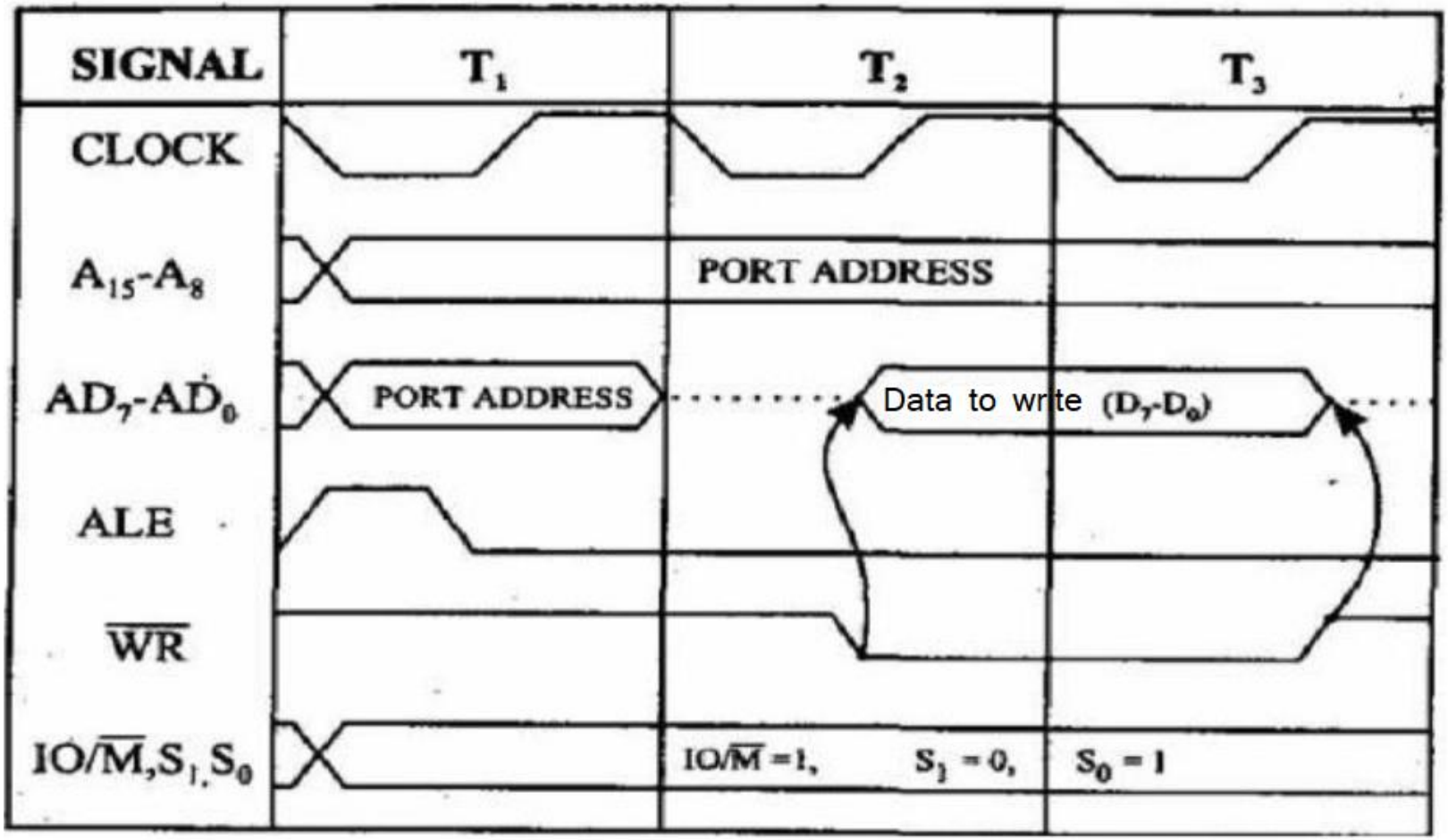
I/O read cycle



I/O Write Cycle

- Microprocessor uses the I/O Write machine cycle for sending a data byte to the I/O port or to the peripheral in I/O mapped I/O systems.
- The IOR machine cycle takes 3 T-states
- The microprocessor places the address of the I/O port specified in the instruction on A15 – A8 address bus and also on AD7 – AD0 address / data bus.
- The data placed on the address / data bus is transferred to the specified I/O port.
- The OUT instruction uses this machine cycle during execution.

I/O Write Cycle



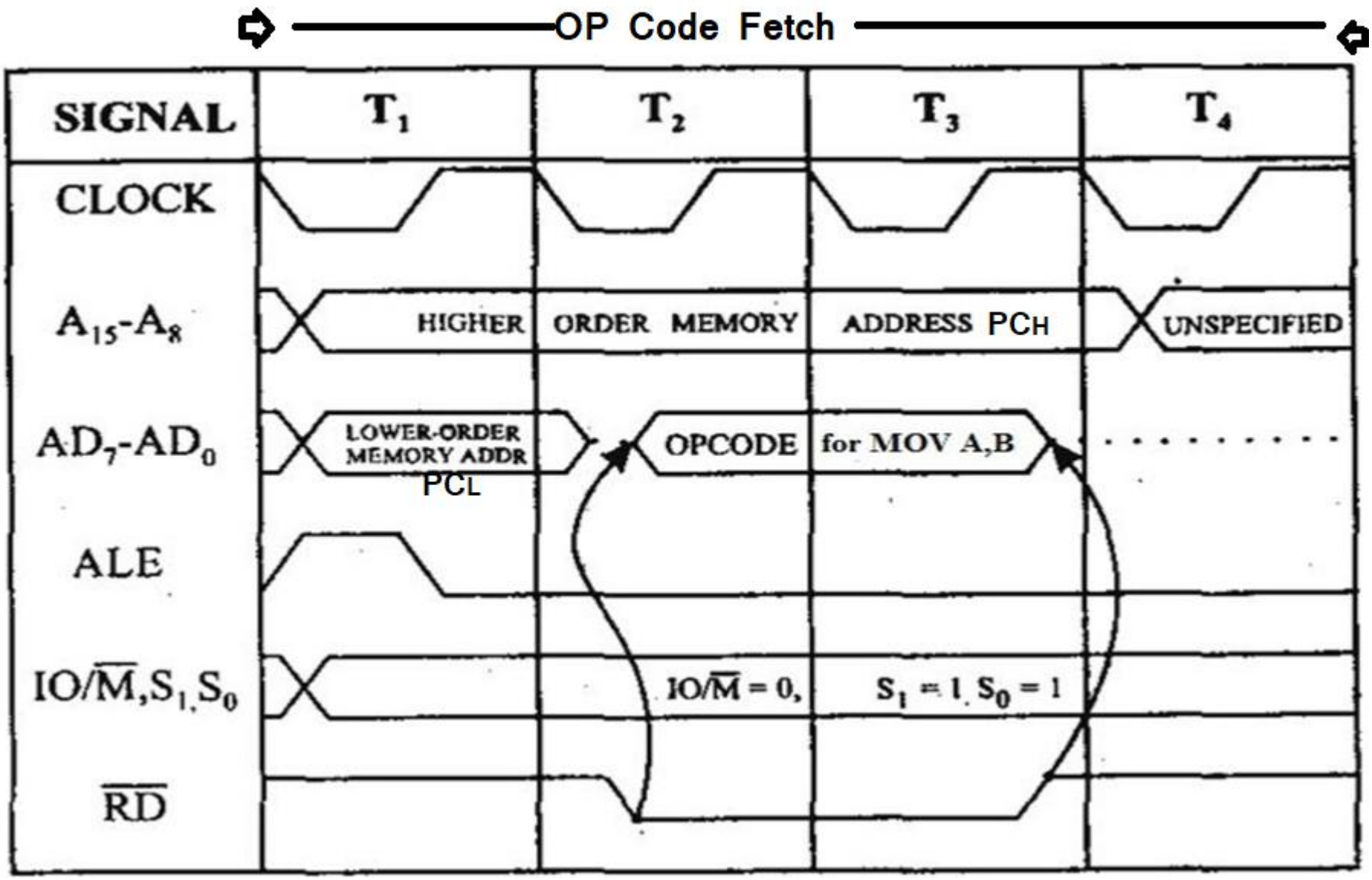
Timing Diagram of Instructions

- MOV regd, regs
- MVI reg, 8bit data
- LDA 16 bit address
- STA 16 bit address
- IN 8 bit port
- OUT 8 bit port

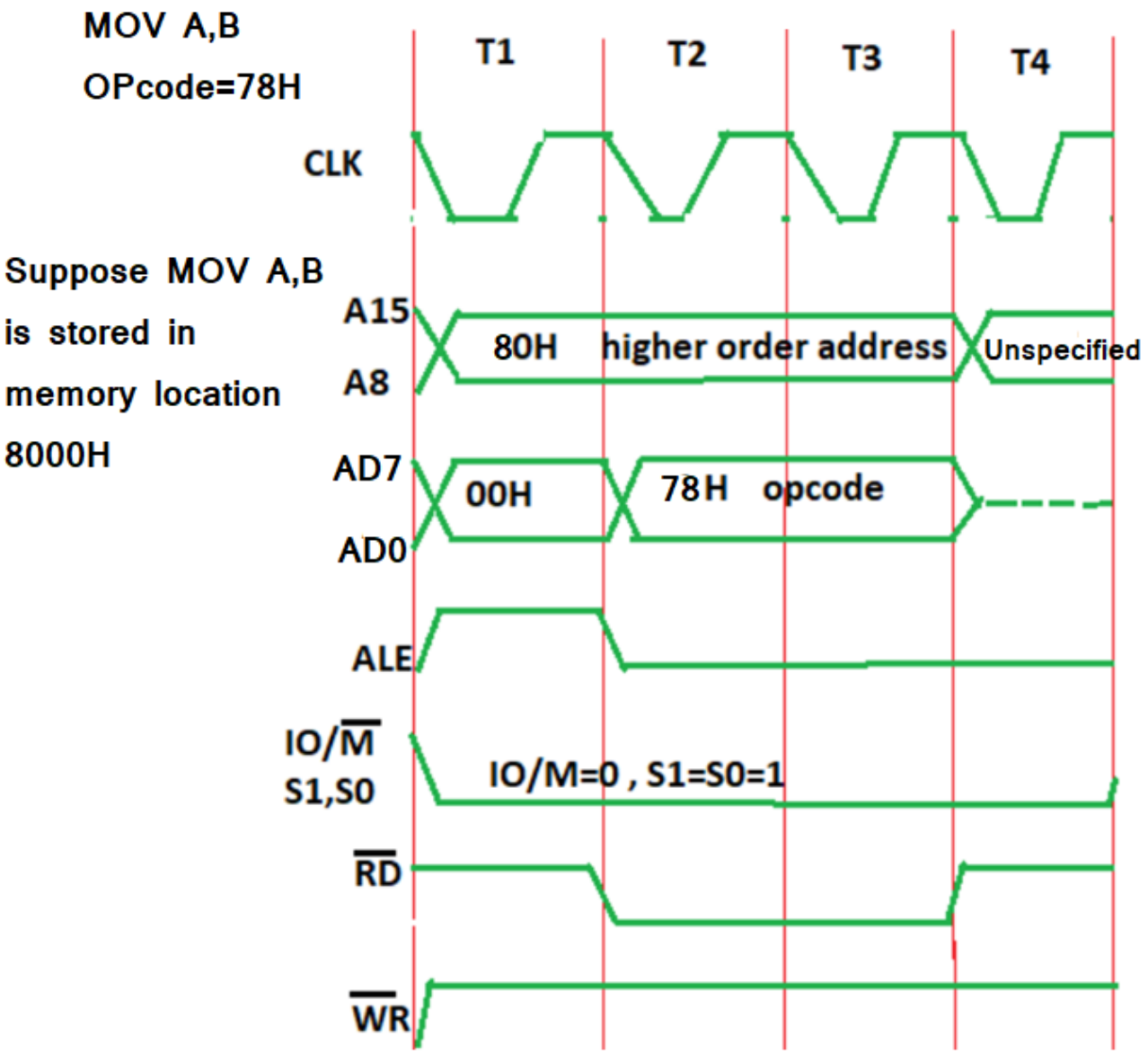
Timing diagram for MOV Rd, Rs (or MOV r1, r2) Instruction

- MOV Rd, Rs instruction moves (copies) the contents of the source register (Rs) into the destination register (Rd).
- The instruction MOV A, B is of 1 byte; therefore the complete instruction will be stored in a single memory address.
- It has only Opcode Fetch machine cycle.
- Some examples for MOV Rd, Rs instruction:
  a. MOV A, B
  b. MOV C, L
- The time taken by the processor to execute the Opcode Fetch cycle is 4T (T- states).
- The first 3 T-states are used for fetching the Opcode from memory and the remaining T-state is used for internal operations by the microprocessor.
- The timing diagram for MOV Rd, Rs (Opcode Fetch machine cycle) is shown in figure.
  ✓ It has 4 T states.

Timing Diagram for MOV A,B



Timing Diagram for MOV A,B



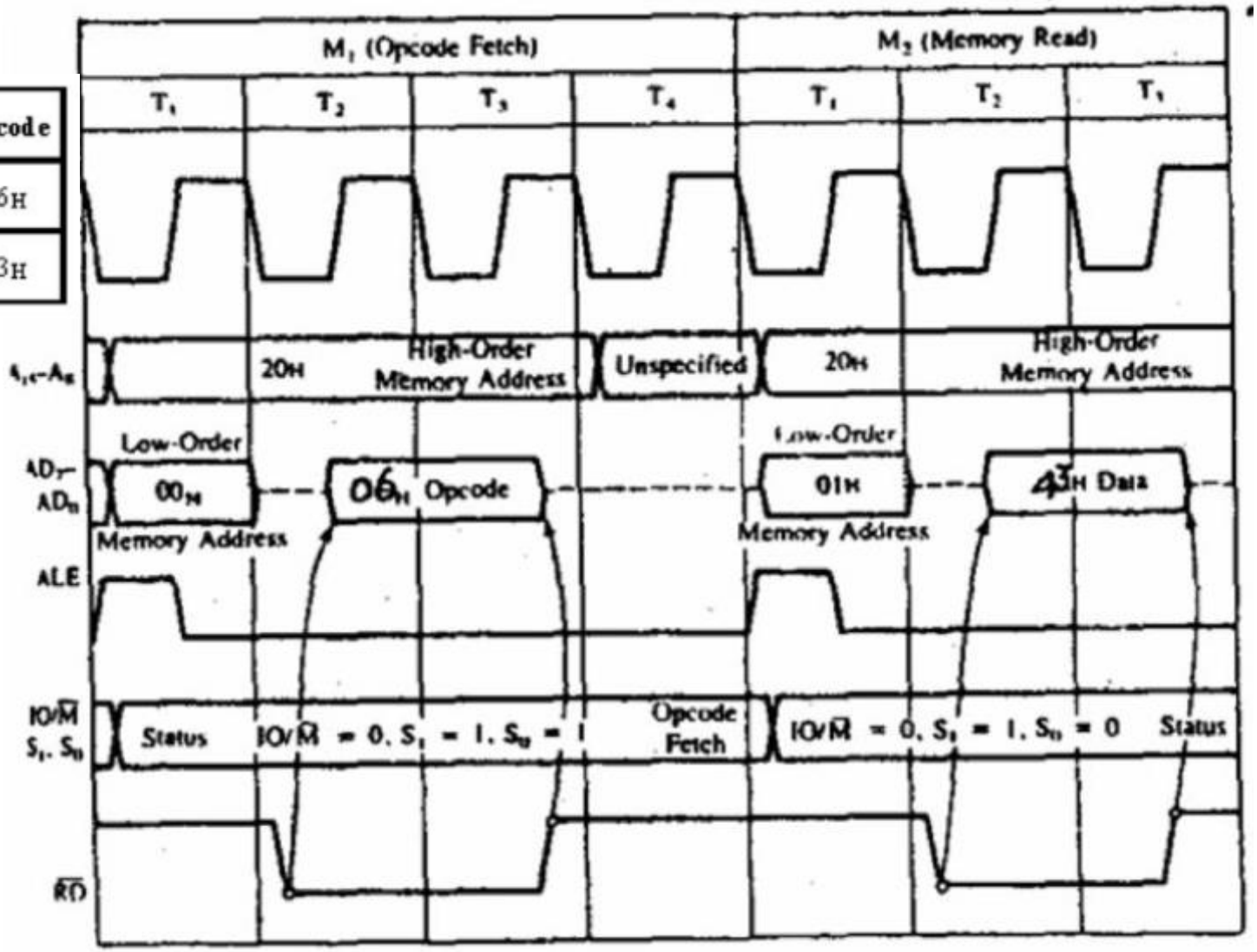
Timing diagram of MVI instruction

- Two byte instruction
- Two machine cycle = Op code Fetch(4-T State) + Memory Read (3-T State)=7-T State
- Decide what is the opcode and what is the data. Here, opcode is ‘MVI B’ and data is 43.
- Assume the memory address of the opcode and the data
- The opcode fetch will be same in all the instructions.
- Only the read instruction of the opcode needs to be added in the successive T states.

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)

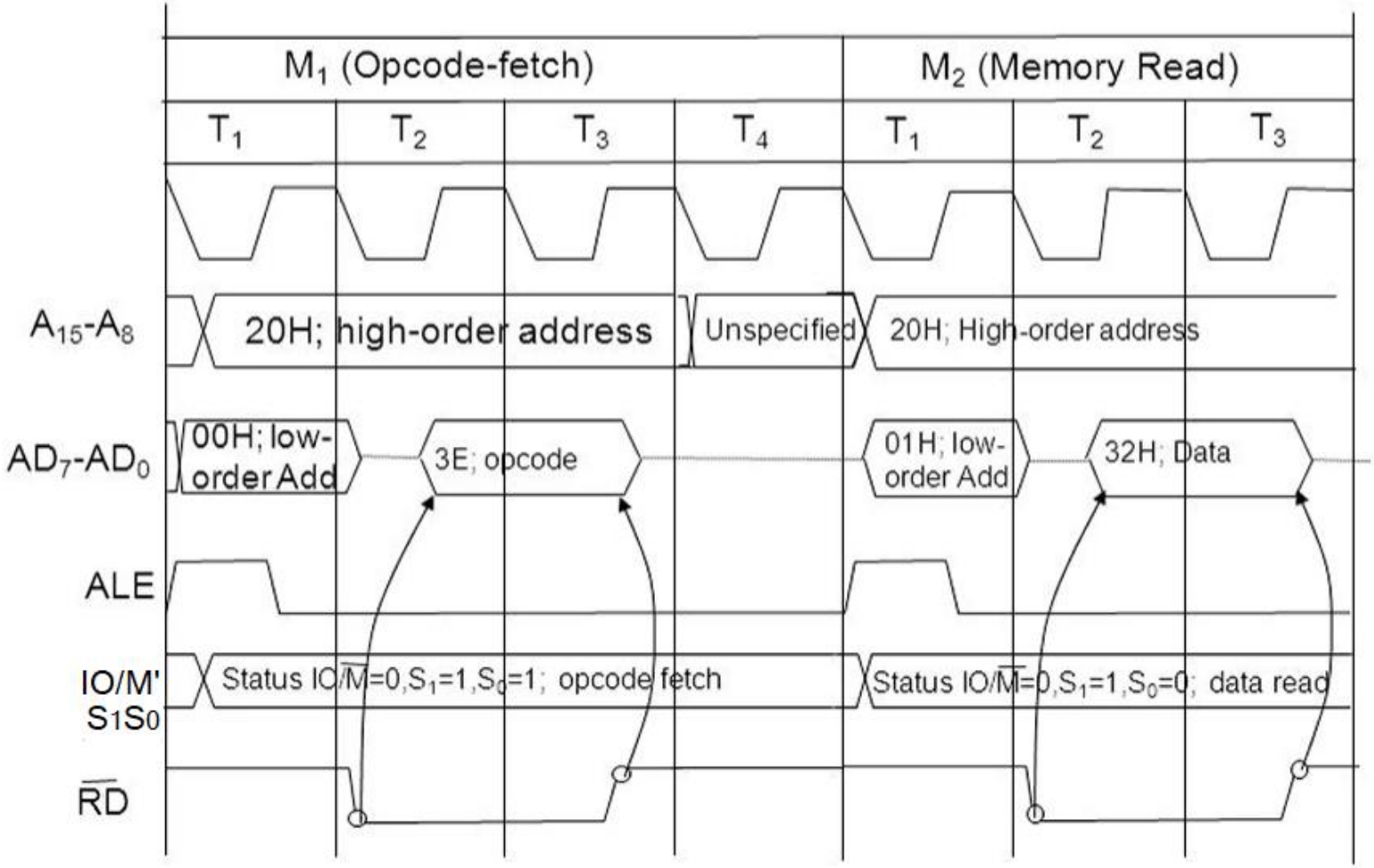
Address	Mnemonics	Opcode
2000	MVI B, 43H	06H
2001		43H



Timing Diagram for MVI A,32H

2000H 3EH ;MVI A, 32H

2001H 32H



LDA 3000h

- LDA: **LDA** is a mnemonic that stands for Load Accumulator with the contents from memory.
  - ✓ 3 Byte Instruction
- Syntax: LDA 16 bit memory Address
- Suppose the instruction LDA 3000h (3 byte instruction) is stored from memory location.

  8000 =3A

  8001= 00

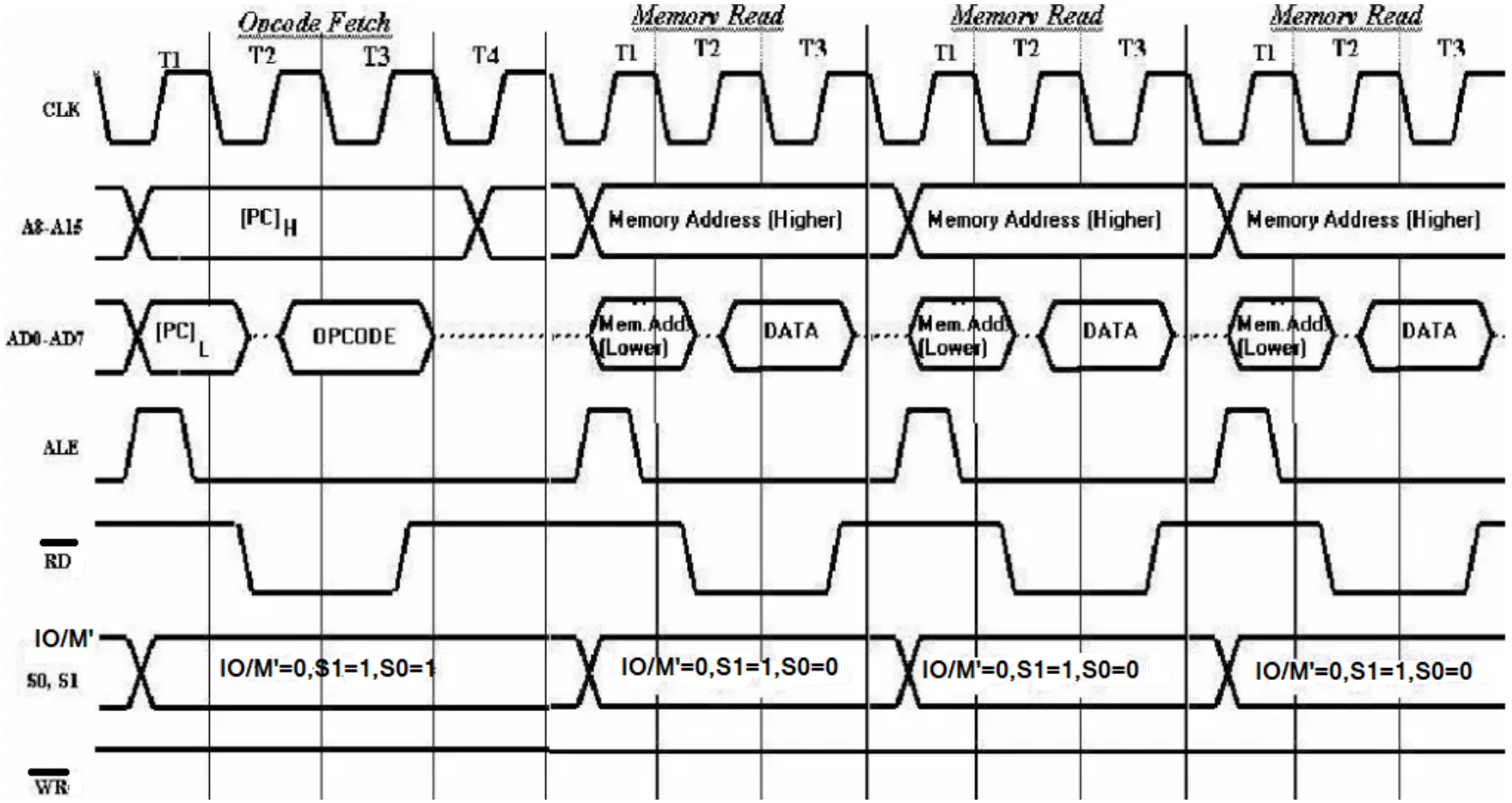
  8002= 30

  IC=OF(4-T State) + MR(3-T State) + MR(3-T State) + MR(3-T State)

Hints:

OP Code Fetch	Memory Read	Memory Read	Memory Read
80	80	80	30
00 opcode=3A	01 00	02 30	00 Data

LDA 16 bit address



STA 3000h

- STA: Store Accumulator
- 3 Byte instruction(1 byte op code;2 byte memory Address)
- Syntax: STA 16 bit memory Address
- Suppose the instruction STA 3000h (3 byte instruction) is stored from memory location starting from 8000h

  8000 =32

  8001= 00

  8002= 30

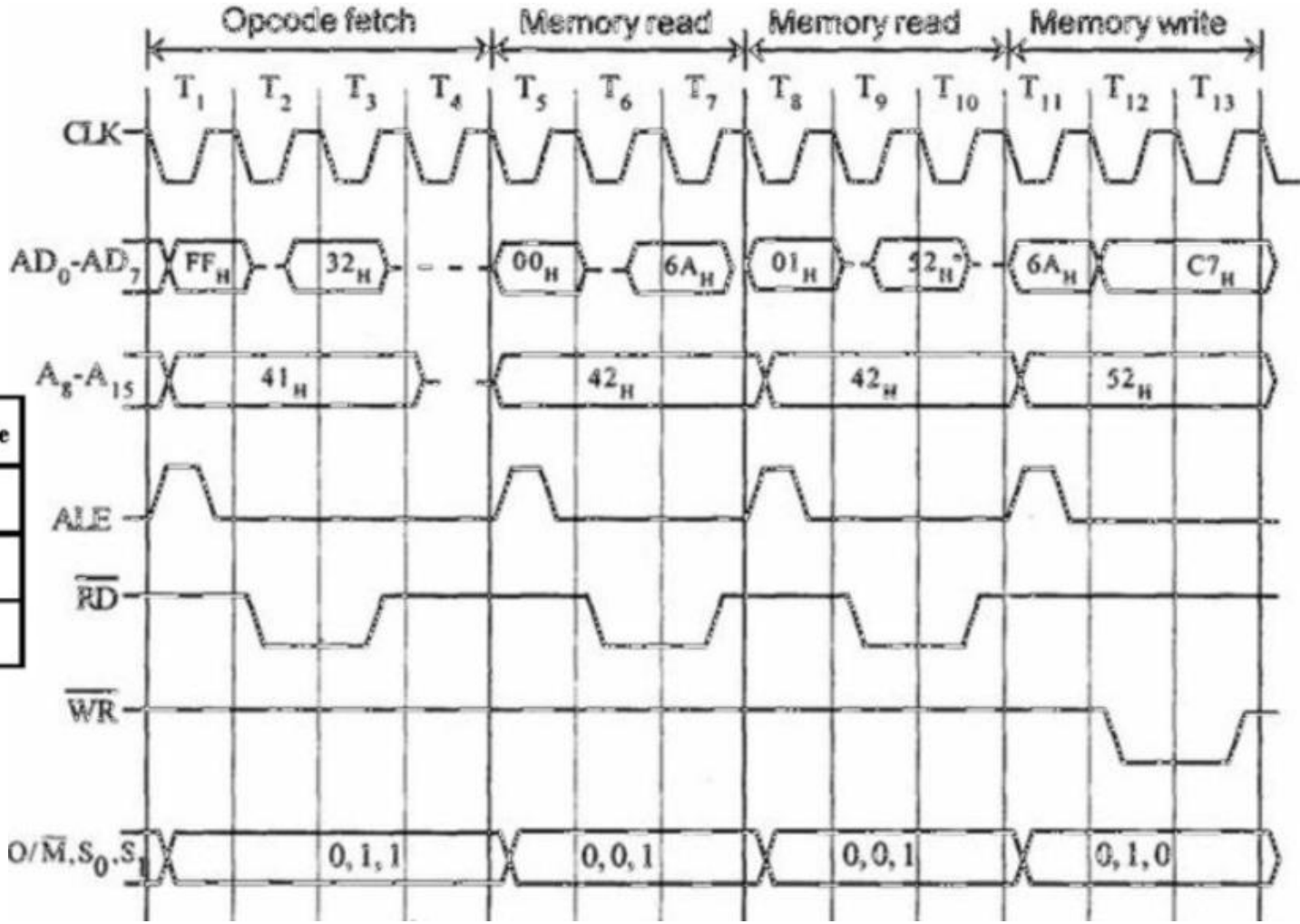
  IC=OF + MR + MR + MW

Hints:

OP Code Fetch	Memory Read	Memory Read	Memory Write
80	80	80	30
00 opcode=32	01 00	02 30	00 Data

STA 526A

Address	Mnemonics	Opcode
41FF	STA 526A$_H$	32$_H$
4200		6A$_H$
4201		52$_H$



IN instruction

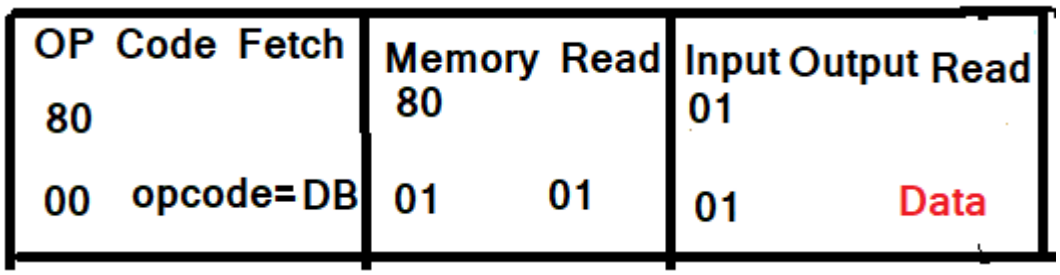
- Syntax: IN 8 bit port address
- IN 01h (2 byte Instruction)
  - ✓ It occupies 2 byte of memory.
- Suppose the instruction is stored in memory location starting from 8000h

8000 DB

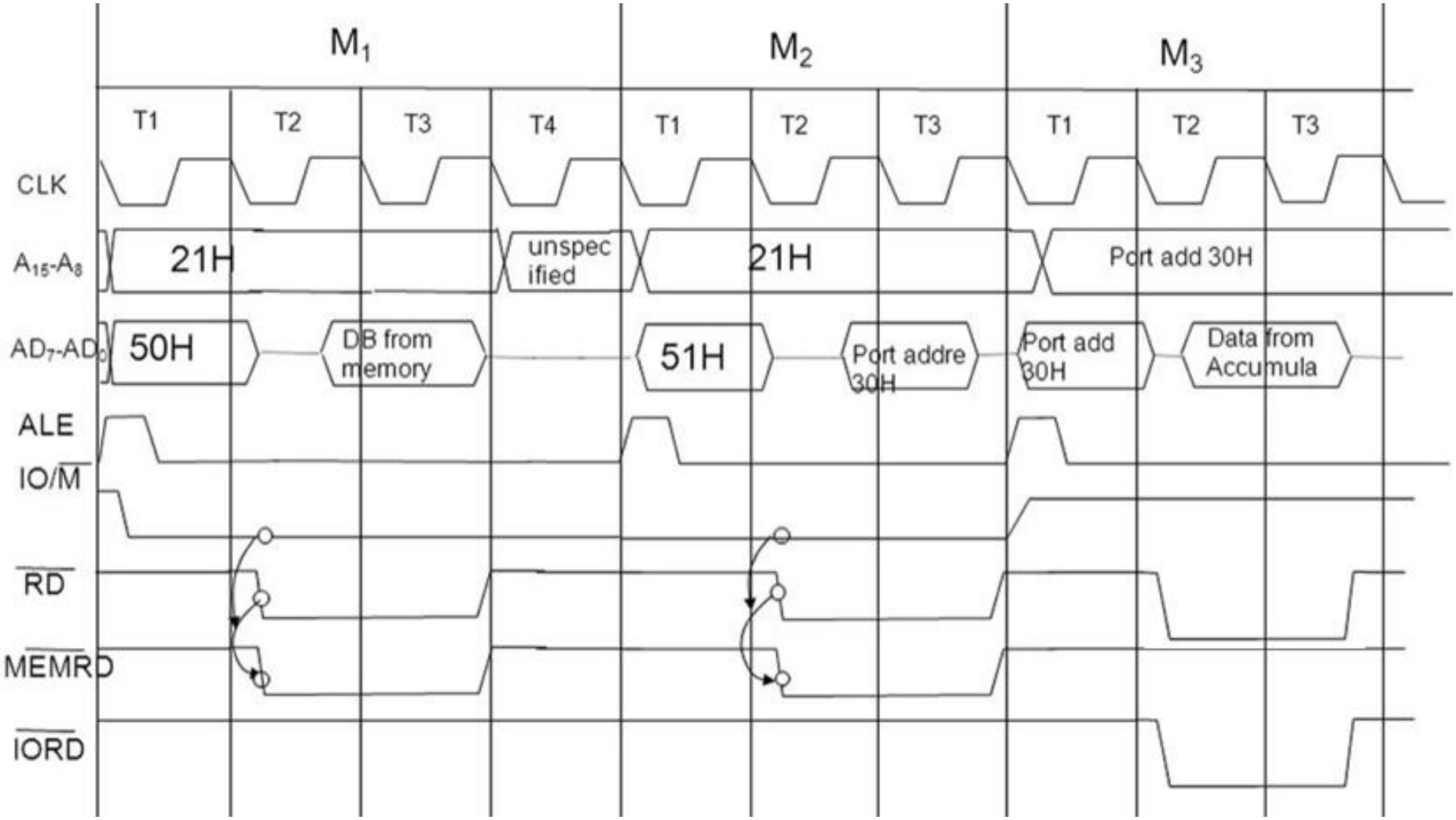
8001 01 (port address)

IC= OF + MR + IOR

Hints:



IN 30H Instruction



OUT Instruction

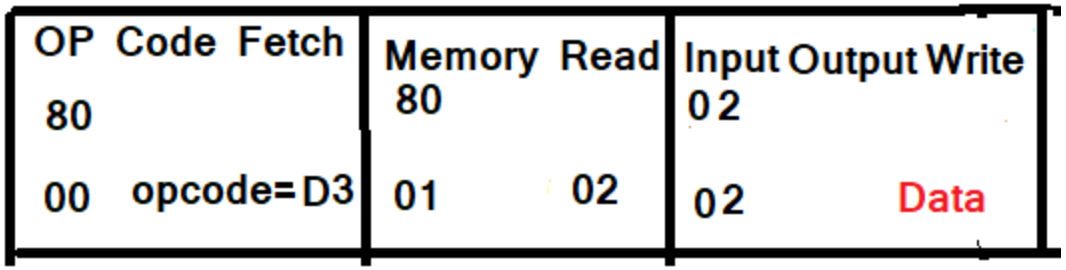
- Syntax: OUT 8 bit port address
- OUT 02h (2 byte Instruction)
  - ✓ It occupies 2 byte of memory.
- Suppose the instruction is stored in memory location starting from 8000h

  5000 D3

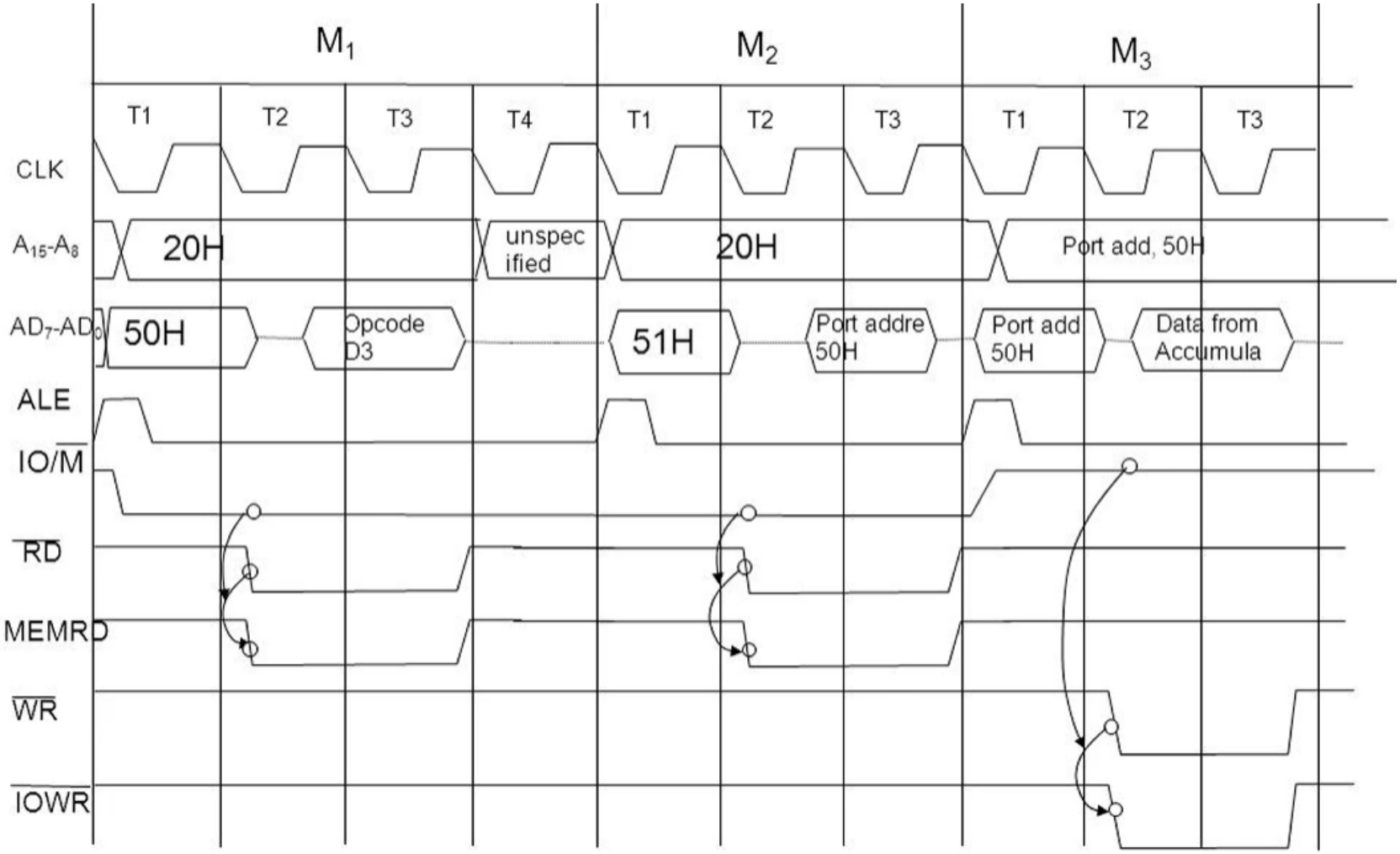
  5001 02 (port address)

  IC= OF + MR + IO W

Hints:



OUT 50H


M1
M2
M3
T1
T2
T3
T4
CLK
A15-A8
20H
unspec
ified
20H
Port add, 50H
AD7-AD0
50H
Opcode
D3
51H
Port addre
50H
Port add
50H
Data from
Accumula
ALE
IO/M
RD
MEMRD
WR
IOWR

Assignment

- Draw timing diagram of following instructions:
- MOV
  MVI
  LDA
  STA
  IN
  OUT

Memory Interfacing and Generation of Chip Select Signal

What is Interface?

- **Interface** is the path for communication between two components.
- Interfacing is of two types, memory interfacing and I/O interfacing.

Memory Interfacing

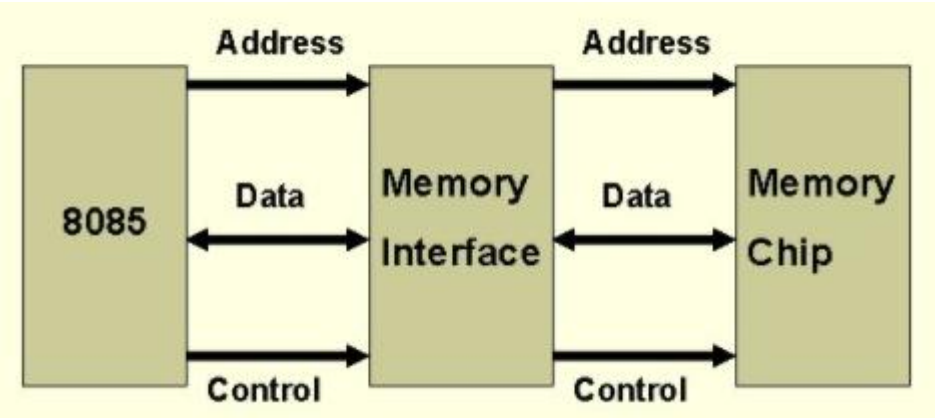
- While executing an instruction, there is a necessity for the microprocessor to access memory frequently for reading various instruction codes and data stored in the memory.
- The interfacing circuit aids in accessing the memory.
- Memory requires some signals to read from and write to registers. Similarly the microprocessor transmits some signals for reading or writing a data.

Purpose of interfacing circuit

- The interfacing process involves matching the memory requirements with the microprocessor signals.
- The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.
- For example for carrying out a READ process, the microprocessor should initiate a read signal which the memory requires to read a data.
- In simple words, the primary function of a memory interfacing circuit is to aid the microprocessor in reading and writing a data to the given register of a memory chip.

Basic concepts of memory Interfacing

- The primary function of memory interfacing is to allow the microprocessor to read from and write into a given register of memory chip.
  - ✓ Be able to select the chip
  - ✓ Identify the register
  - ✓ Enable the appropriate buffer.
- Microprocessor need to access memory quite frequently to read instructions and data stored in memory; the interface circuit enables that access.

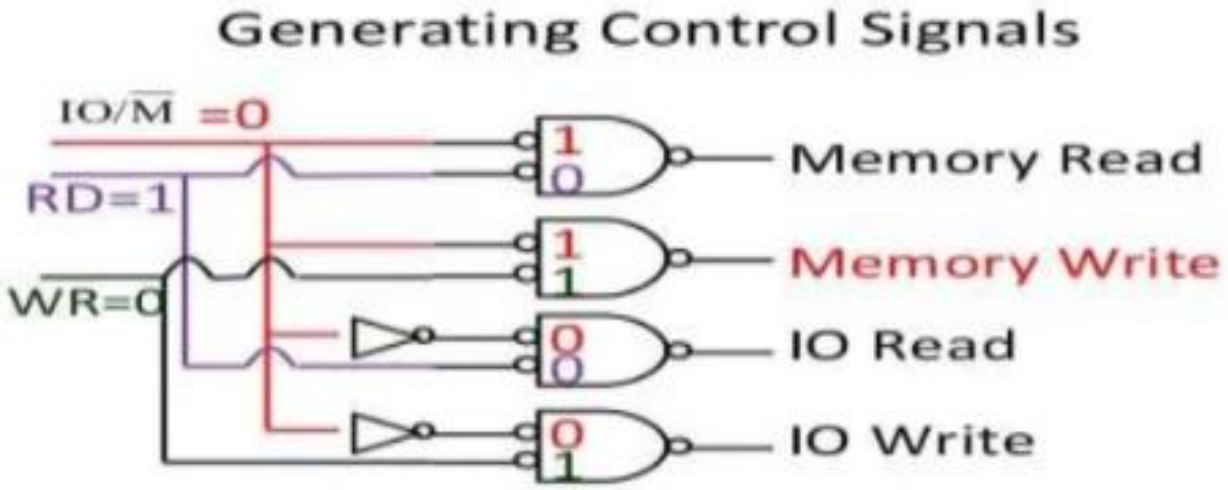


Basic concepts of memory Interfacing

- The interface process involves designing a circuit that will match the memory requirements with the microprocessor signal.
- Memory has certain signal requirements to read from and write into memory.
- Similarly Microprocessor initiates the set of signals when it wants to read from and write into memory.
  - ✓ 8085 has 16 address lines (A0 - A15), hence a maximum of 64 KB (= 2^16 bytes) of memory locations can be interfaced with it.
  - ✓ The memory address space of the 8085 takes values from 0000H to FFFFH.
  - ✓ The 8085 initiates set of signals such as IO/M', RD' and WR' when it wants to read from and write into memory.
  - ✓ Similarly, each memory chip has signals such as CS' (chip select), OE'/ RD' (output enable or read) and WE'/ WR' (write enable or write) associated with it.

Generation of Control Signals for Memory

- When the 8085 wants to read from or write into memory, it activates IO/M', RD' or WR' signals.
- Using IO/M', RD' and WR' signals, two control signals MEMR' (memory read) and MEMW' (memory write) are generated.



Function of Memory Interfacing:

- Primary Function of memory interfacing is that the microprocessor should be able to read from and write into a given register of a memory chip:
  - ✓ Select the Chip
  - ✓ Identify the register
  - ✓ Enable the appropriate buffer.

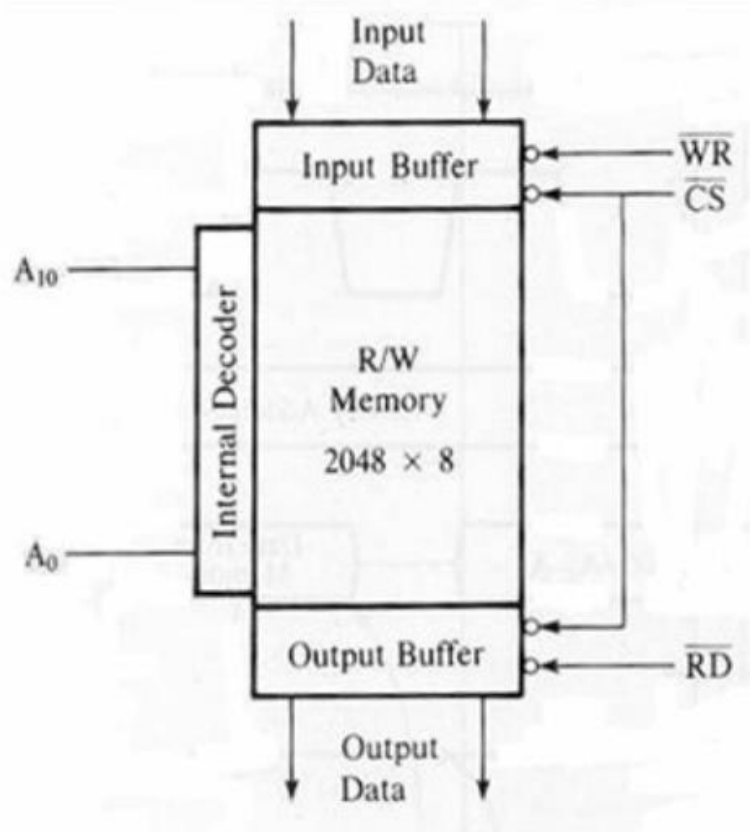
Requirement and Memory structure

- RAM: Read and Write
- ROM: Read Only

Structure of R/W Memory (2KB RAM)

- 2048 (2K) Size
- Each Register store 8-bits
- 8 input, 8-output lines
- 11 address lines (A10-A0)
- 1 chip select
- 2 control lines to enable input and output buffer.
  - RD': Enable output buffer
  - WR': Enable input buffer
- Internal decoder to decode internal memory address lines

❖ MEMR and MEMW are given to RD and WR pins of Memory chip.

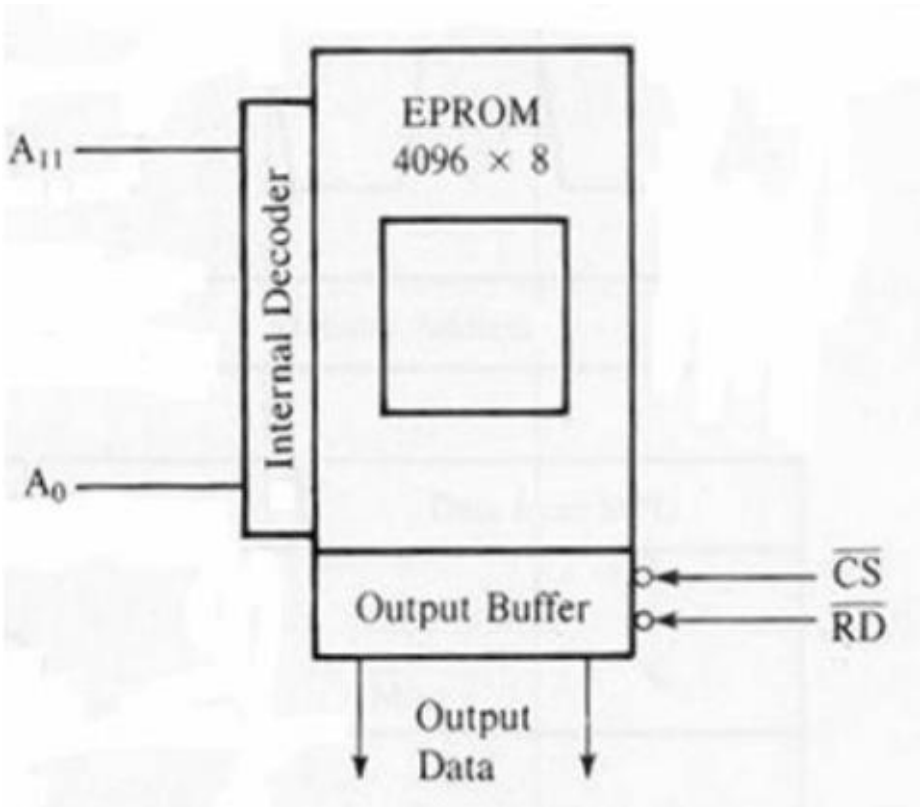


Structure of ROM

Generally EPROMS are used as program memory and RAM as data memory.

EPROM: Chip must be programmed before it can be used as ROM.

- 4096 (4K) registers.
- Register store 8-bits
- Internal decoder to decode address lines
- 12 address lines (A11-A0)
- 1 chip select
- 1 Read control Signal lines to enable output buffer



8085 Interfacing with Memory:

- The following are the steps involved in interfacing memory with 8085 processor.

✓ First decide the size of memory requires to be interfaced. Depending on this we can say how many address lines are required for it.

  ➢ For example if you want to interface 4KB memory it requires 12 address lines. Remaining 4 address lines can be used in address decoding.

✓ Depending on the size of memory required and given address range, construct address decoding circuitry.

  ➢ This address decoding circuitry can be implemented with NAND gates or decoders.

✓ Connect data bus of memory to processor data bus.

✓ Generate the control signals required for memory using IO/M', WR', RD' signals of 8085 processor.

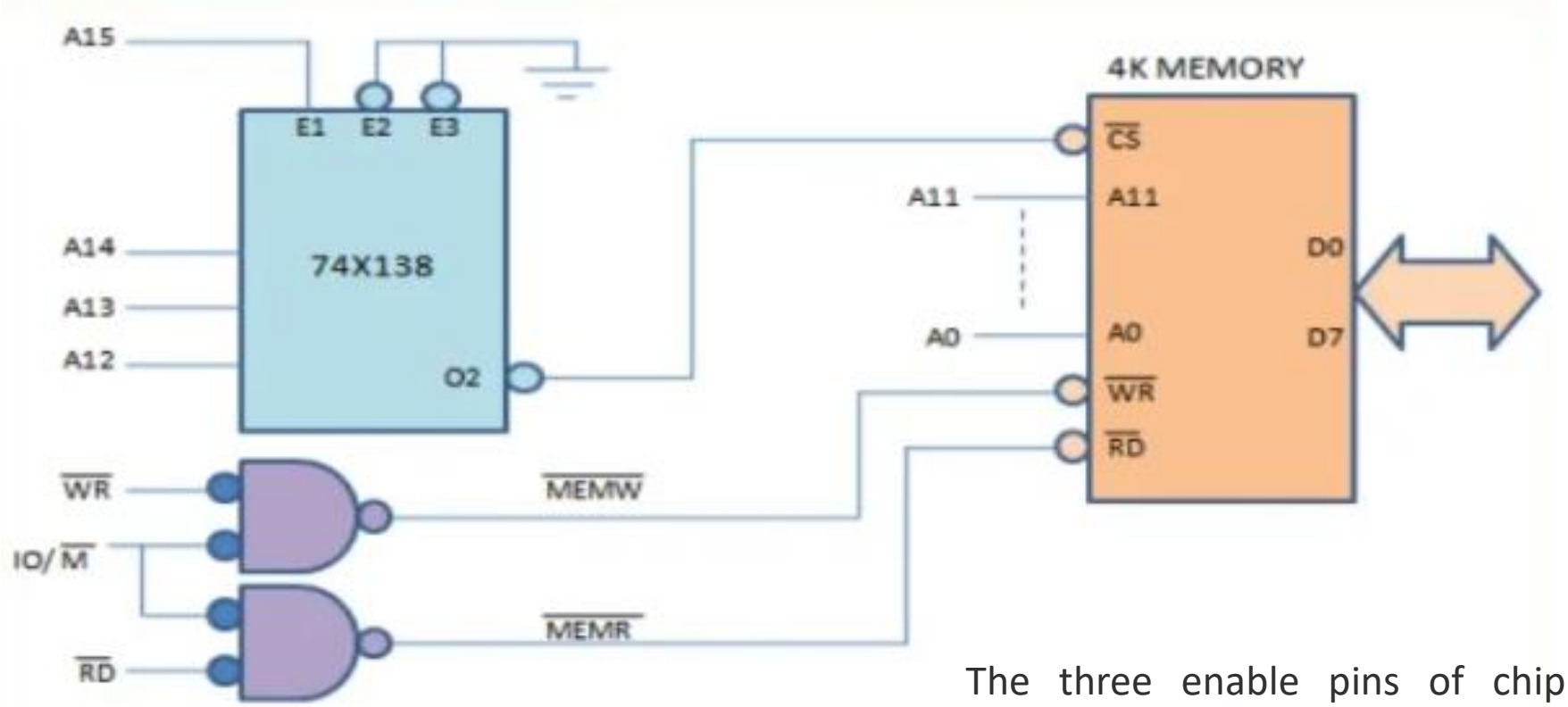
Example: Interface 4KB memory to 8085 with starting address A000H.

- 4KB memory requires 12 address lines for addressing and 4 address lines are used for generating chip select signal.
  - ✓ Given that starting address for memory is A000H.
  - ✓ So for 4KB memory ending address becomes A000H+0FFFH (4KB) = AFFFH.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

- A0-A11 address lines are directly connected to address bus of memory chip.
- A12 - A15 are used for generating chip select signal for memory chip.

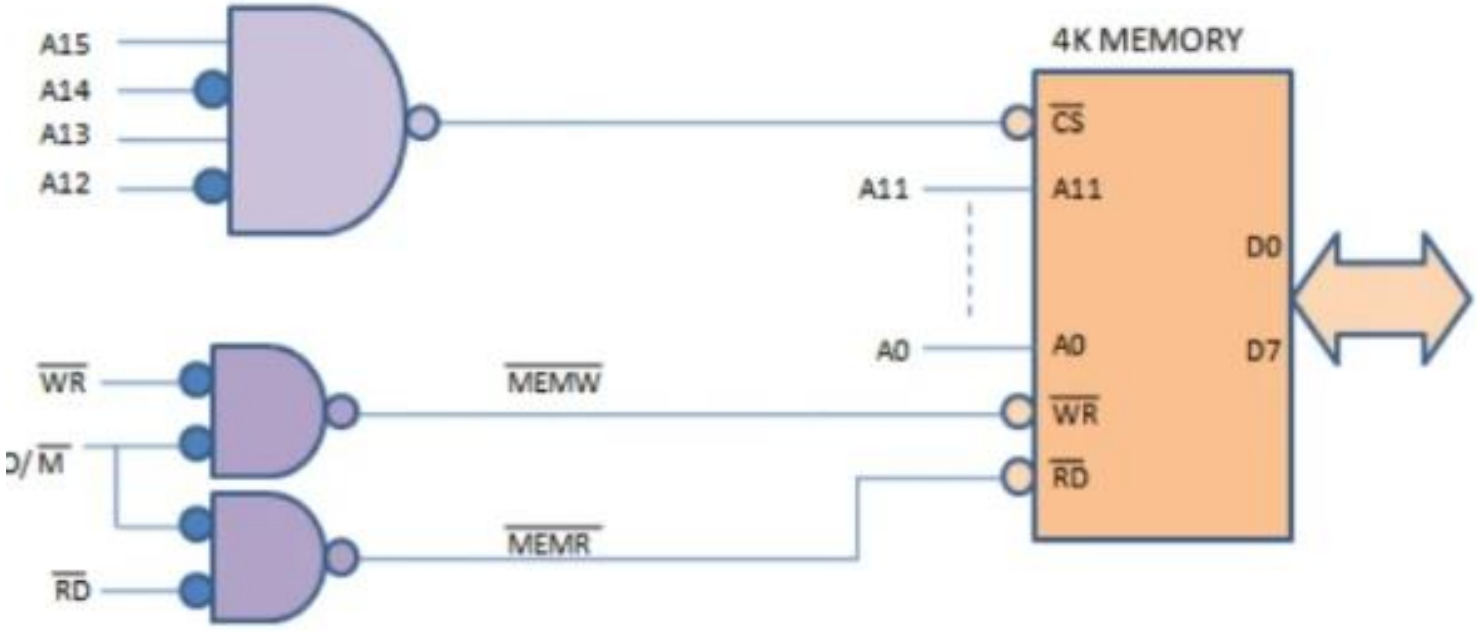
Address decoding circuit using 3X8



The three enable pins of chip (in which Two active-low and one active-high)

- A15 line is use for enabling 74x138 the decoder chip. A12, A13, A14 lines are connected to the 74X138 chip as inputs.
- When theses lines are 010 output should be ‘0’. This is provided at O_2 pin of the 74X138 chip.

Address decoding circuit using only NAND gates:



A15, A14, A13, A12 inputs should be 1010, for enabling the chip.
So the circuit for this is as shown above.

Types of address decoding:

- There are two types of address decoding mechanism, based on address lines used for generating chip select signal.
- Absolute decoding.
- Partial decoding

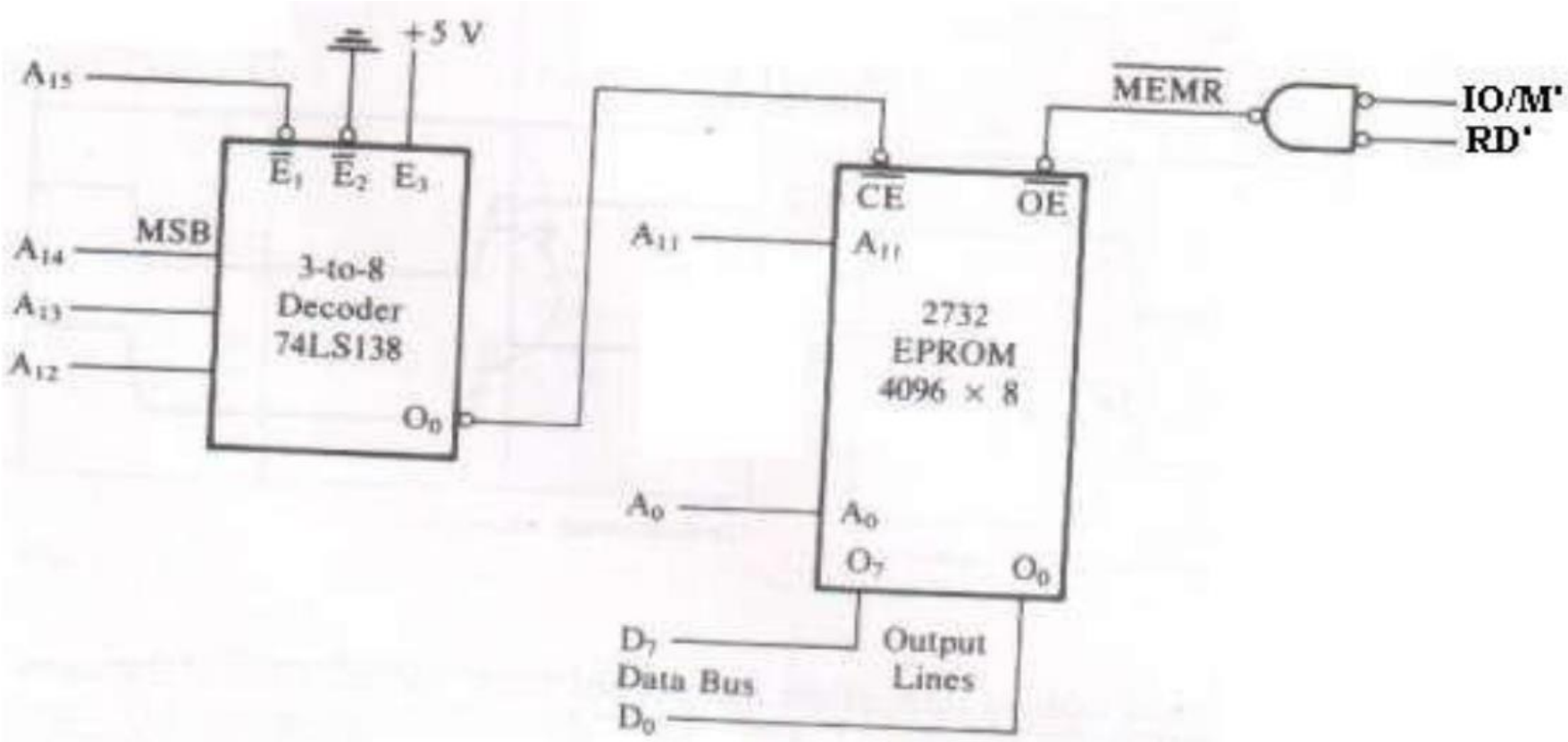
Absolute decoding

- All the higher order lines of microprocessor, left after using the required signals for memory are completely used for generating chip select signal as shown in above example.
- This type of decoding is called absolute decoding

Partial decoding

- Only some of the address lines of microprocessor left after using the required signals for memory are used for generating chip select signal.
- Because of this multiple address ranges will be formed. If total memory space is not required for the system then, this type of address decoding can be used.
- The advantage of this technique is fewer components are required for memory interfacing because of this board size reduces and in turn cost reduces

8085 Interfacing Circuit to interface EPROM: (using 3x8 Decoder)



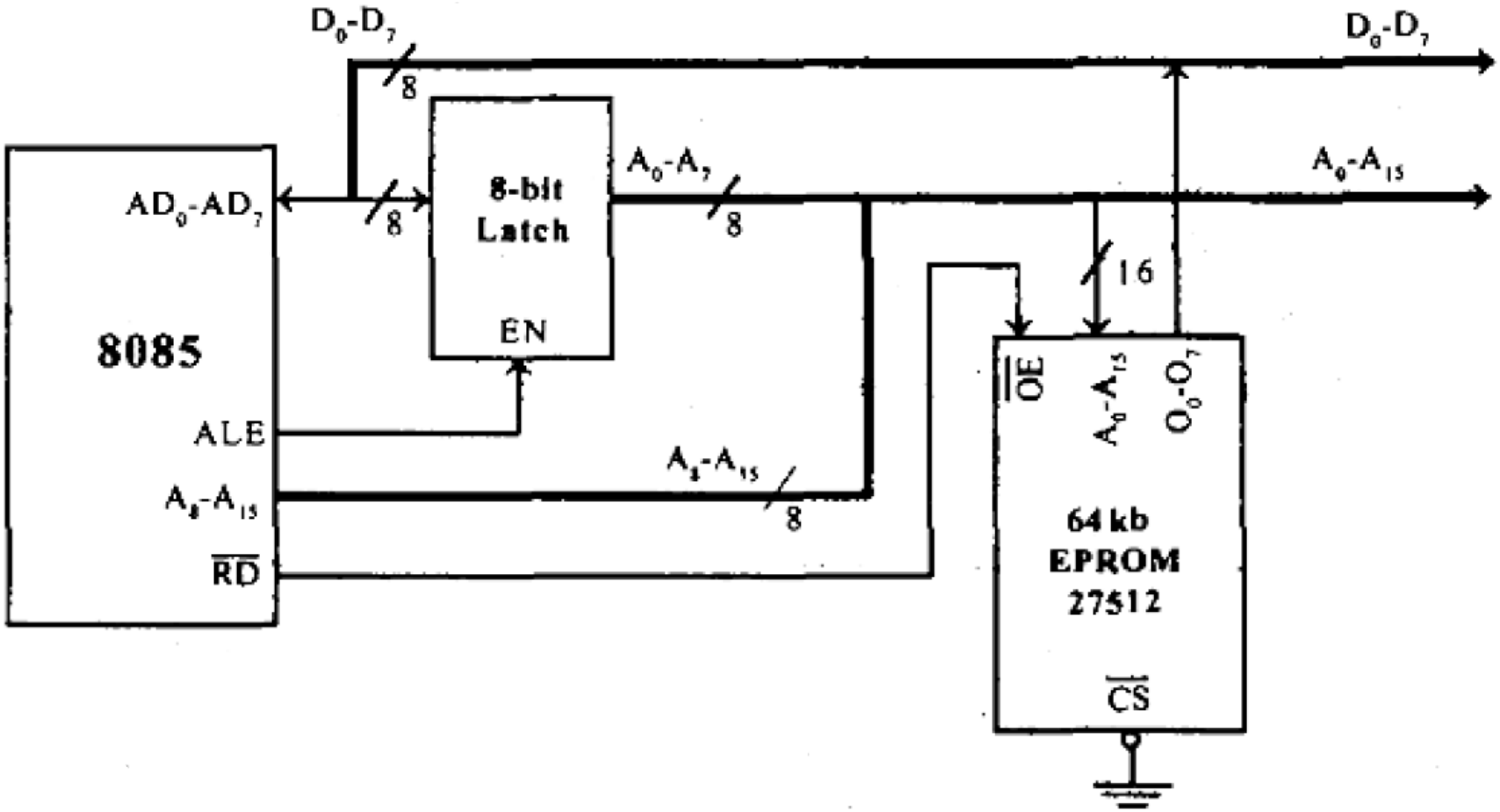
8085 Interfacing Circuit to interface EPROM: (using 3x8 Decoder)

- The 8085 address lines A11-A0 are connected to the pins A11-A0 of the memory chip.
  - ✓ Decoder decodes A15-A12 and output O_0 is connected to CE' which is asserted only when A15-A12 is 0000.
  - ✓ One control signal MEMR' is connected to OE' to enable output buffer

Example: Interfacing 64Kb EPROM with 8085:

- Consider a system in which the full memory space 64kb is utilized for EPROM memory.
- ✓ The memory capacity is 64 Kbytes. i.e. 2^n = 65536 bytes where n = 16 (address lines).
- ✓ In this system the entire 16 address lines of the processor are connected to address input pins of memory IC in order to address the internal locations of memory.
- ✓ The chip select (CS) pin of EPROM is permanently tied to logic low (i.e., tied to ground).
- ✓ Since the processor is connected to EPROM, the active low RD pin is connected to active low output enable pin of EPROM.
- ✓ The range of address for EPROM is 0000H to FFFFH
- ✓ Latch has been used to separate the data and address bus

Example: Interfacing 64Kb EPROM with 8085:



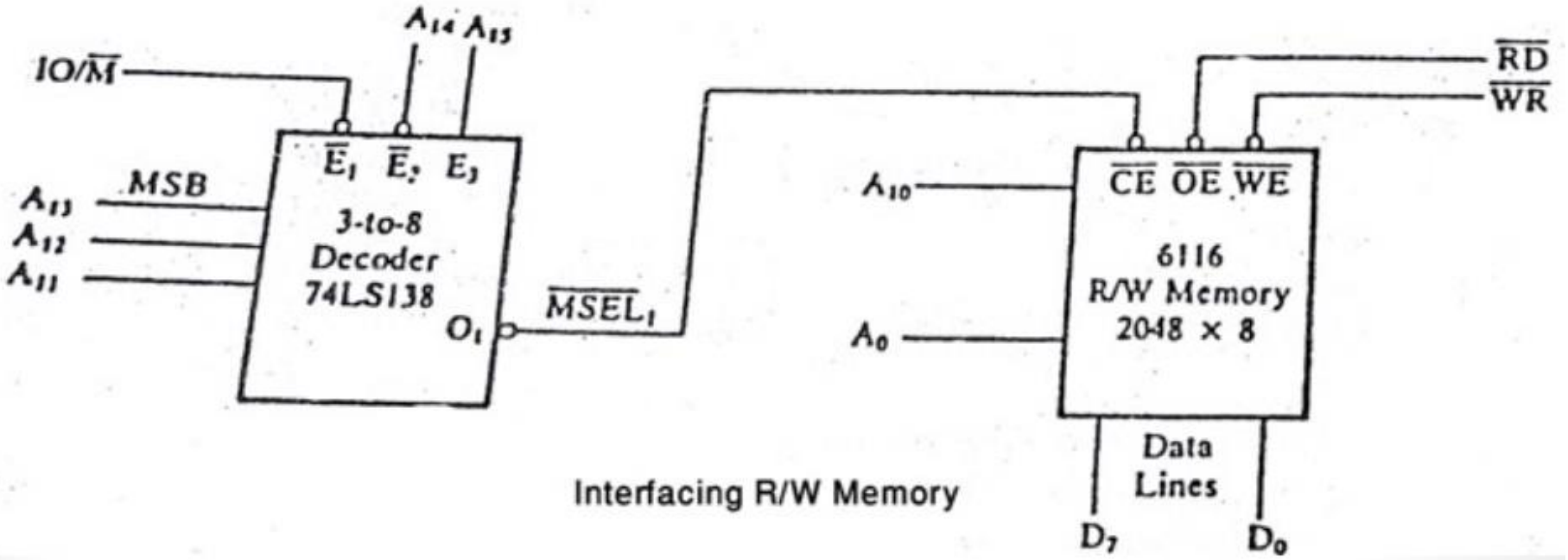
8085 Interfacing 6116 RAM Memory (2KB) Chip:

- 11 Address lines A10-A0 to decode 2048K registers.
- ✓ Address lines A15-A11 are connected to decoder (which is enabled by IO/M' signal in addition to the address lines A15 and A14).
- RD' and WR' signals are directly connected to memory chip.
- MEMR' and MEMW' need not to be generated separately (this technique save two gates).
- Memory Address Ranges from 8800H to 8FFFH.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	8800H
Chip Select				1	1	1	1	1	1	1	1	1	1	1	1	8FFFH

- A13-A11 (001) activate output O_1 of decoder which is connected to CE' of memory chip and it is asserted only when IO/M' is low

8085 Interfacing 6116 RAM Memory (2KB) Chip:



Interfacing R/W Memory

8085 Interfacing: Example

- Example: Interface two 6116 ICs with the 8085 using 74LS138 decoder such that the starting addresses assigned to them are 8000H and 9000H, respectively.
- Specification of IC 6116: 2 KB RAM
  - 2 K x 8 RAM
  - 2 KB = 2^11 bytes
  - 11 address lines
- 6116 has 11 address lines and since 2 KB, therefore ending addresses of 6116 chip 1 and chip 2 are 87FFH and 97FFH, respectively.
- Table shows the address range of the two chips.

8085 Interfacing: Example

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Address
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	87FFH (RAM chip 1)
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	9000H
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	97FFH (RAM chip 2)

8085 Interfacing: Example

- Fig. shows the interfacing.
- A0 – A10 lines of 8085 are connected to 11 address lines of the RAM chips.
- Three address lines of 8085 having specific value for a particular RAM are connected to the three select inputs (C, B and A) of 74LS138 decoder.
- Table shows that A13=A12=A11=0 for the address assigned to RAM 1 and A13=0, A12=1 and A11=0 for the address assigned to RAM 2.
- Remaining lines of 8085 which are constant for the address range assigned to the two RAM are connected to the enable inputs of decoder.
- When 8085 places any address between 8000H and 87FFH in the address bus, the select inputs C, B and A of the decoder are all 0. The Y0 output of the decoder is also 0, selecting RAM 1.
- When 8085 places any address between 9000H and 97FFH in the address bus, the select inputs C, B and A of the decoder are 0, 1 and 0.
- The Y2 output of the decoder is also 0, selecting RAM 2.

8085 Interfacing: Example

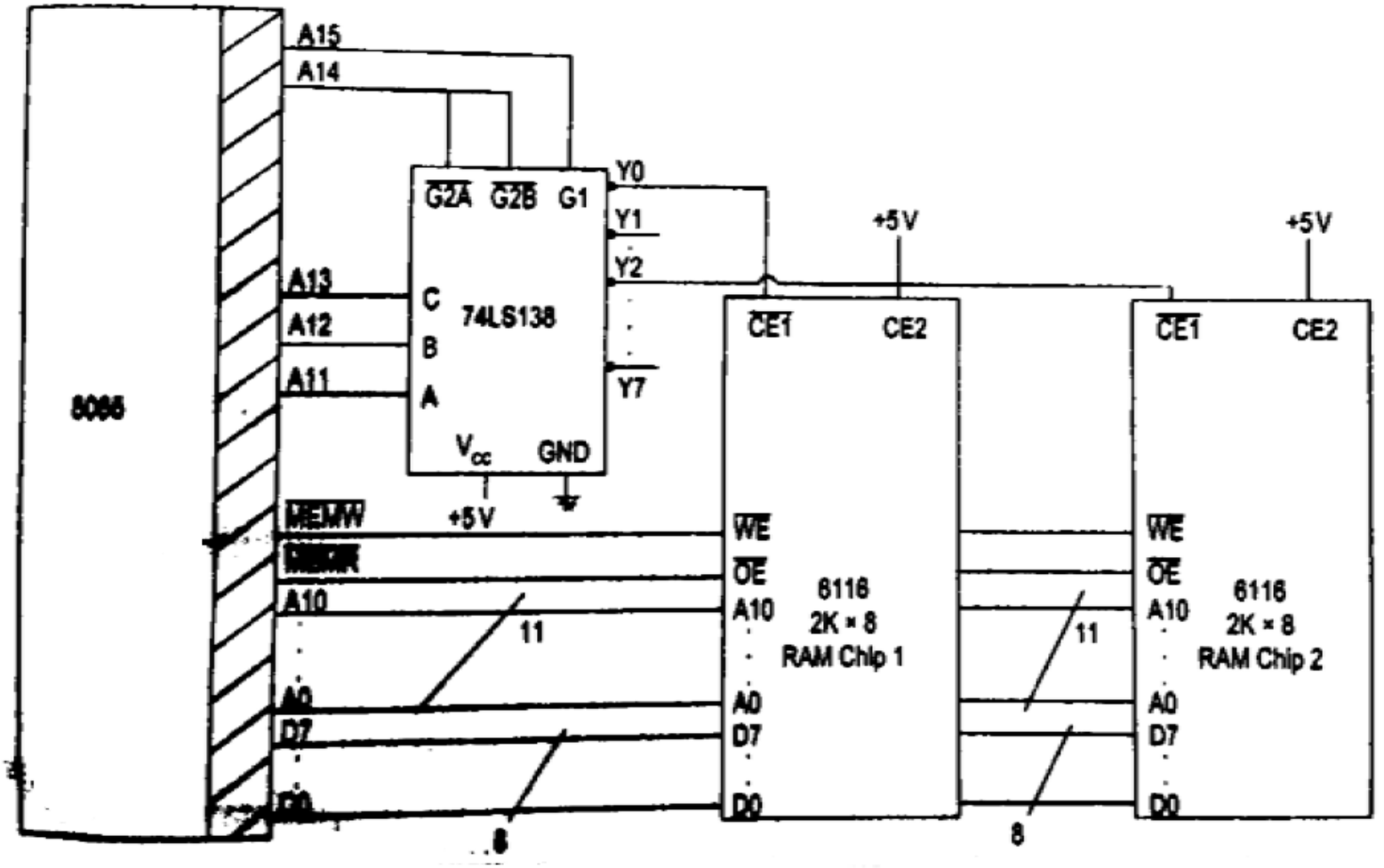


Fig. Interfacing two 6116 RAM chips using 74LS138 decoder

Address decoding: Example

- Design an address decoding circuit for two RAM chips each of 256 bytes at address 5300H.
- Solution:

256 bytes requires 8 address lines.

$2^x=256$, $x=8$

So to address one of 256 bytes in each RAM requires 8 address lines A_0-A_7

Block		Address	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	Start	5300H	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0
	End	53FFH	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1
2	Start	5400H	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
	End	54FFH	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1

Address decoding: Example