Microprocessor

BCA 3rd Semester 2020

Lecture- 11

SUBHADIP MUKHERJEE

Department of computer science

Kharagpur college

TIMING DIAGRAM OF 8085

TIMING DIAGRAM

- Timing Diagram is a graphical representation.
- It represents the execution time taken by each instruction in a graphical format.
- The execution time is represented in

  T-states.

CONTROL SIGNALS

IO/M(Active Low)	S1	S2	Data Bus Status(Output)
0	0	0	Halt
0	0	1	Memory WRITE
0	1	0	Memory READ
1	0	1	IO WRITE
1	1	0	IO READ
0	1	1	Opcode fetch
1	1	1	Interrupt acknowledge

INSTRUCTION CYCLE

- The time required to execute an instruction is called instruction cycle.

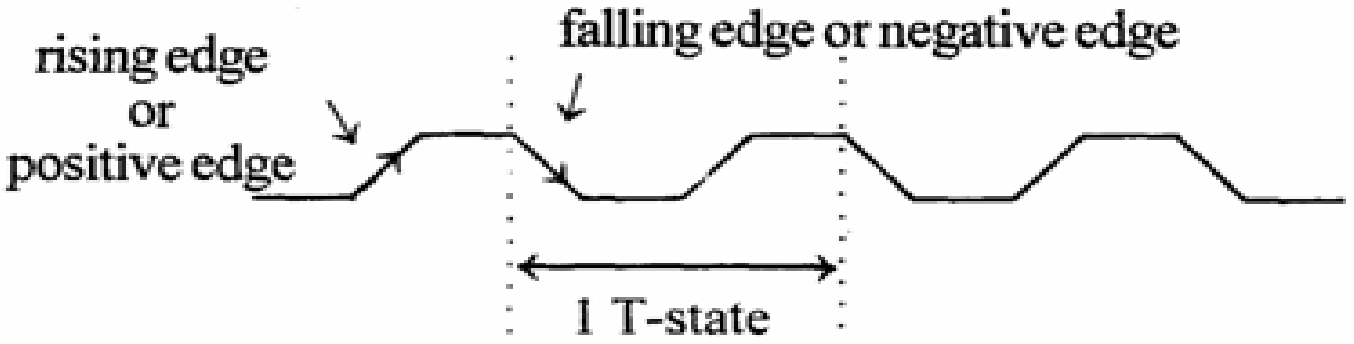
MACHINE CYCLE

- The time required to access the memory or input/output devices is called machine cycle.

T-STATE

- The machine cycle and instruction cycle takes multiple clock periods.
- A portion of an operation carried out in one system clock period is called as T-state.

Note : *Time period, $T = 1/f$; where f = Internal clock frequency*



MACHINE CYCLES OF 8085

The 8085 microprocessor has 5 basic machine cycles.

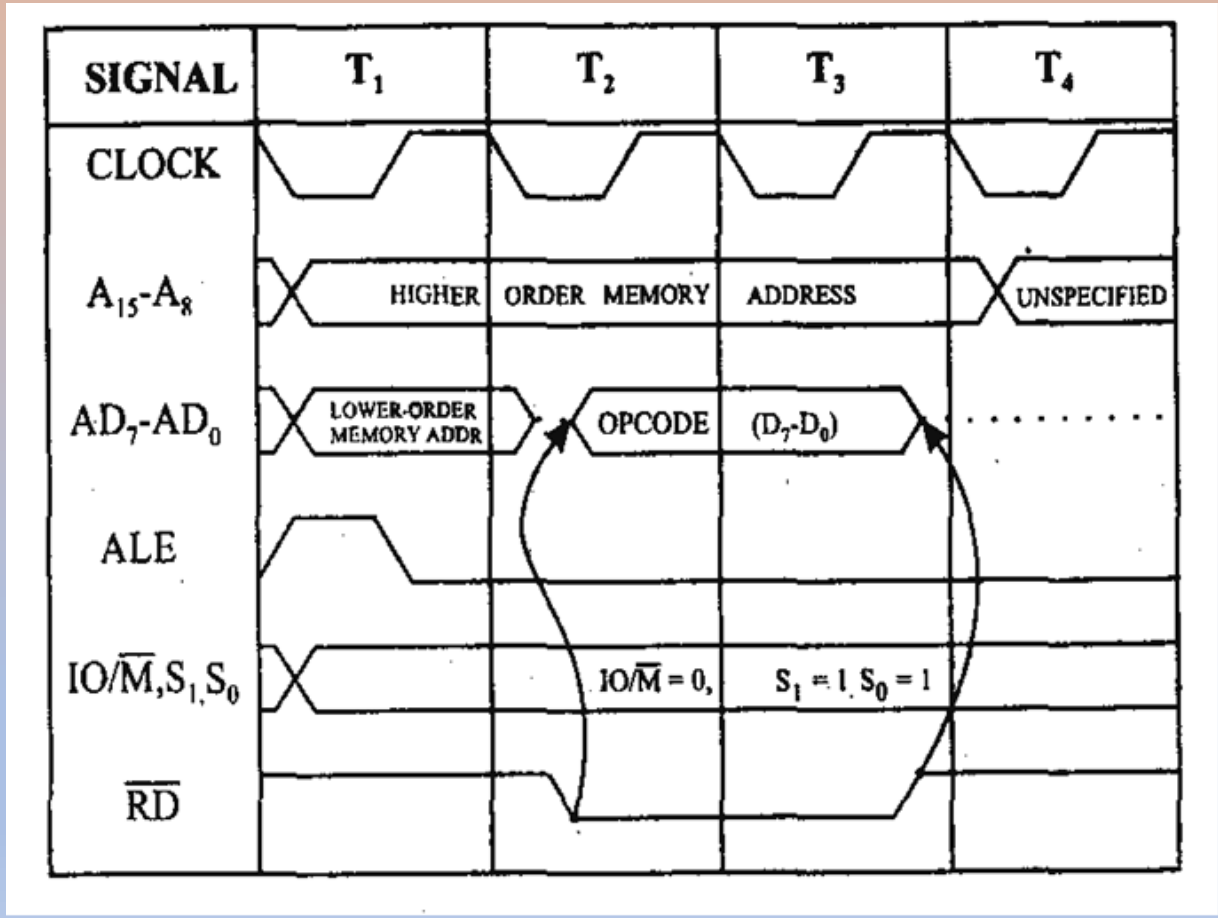
They are

1. Opcode fetch cycle (4T)
2. Memory read cycle (3 T)
3. Memory write cycle (3 T)
4. I/O read cycle (3 T)
5. I/O write cycle (3 T)

MACHINE CYCLES OF 8085

- The processor takes a definite time to execute the machine cycles. The time taken by the processor to execute a machine cycle is expressed in T-states.
- One T-state is equal to the time period of the internal clock signal of the processor.
- The T-state starts at the falling edge of a clock.

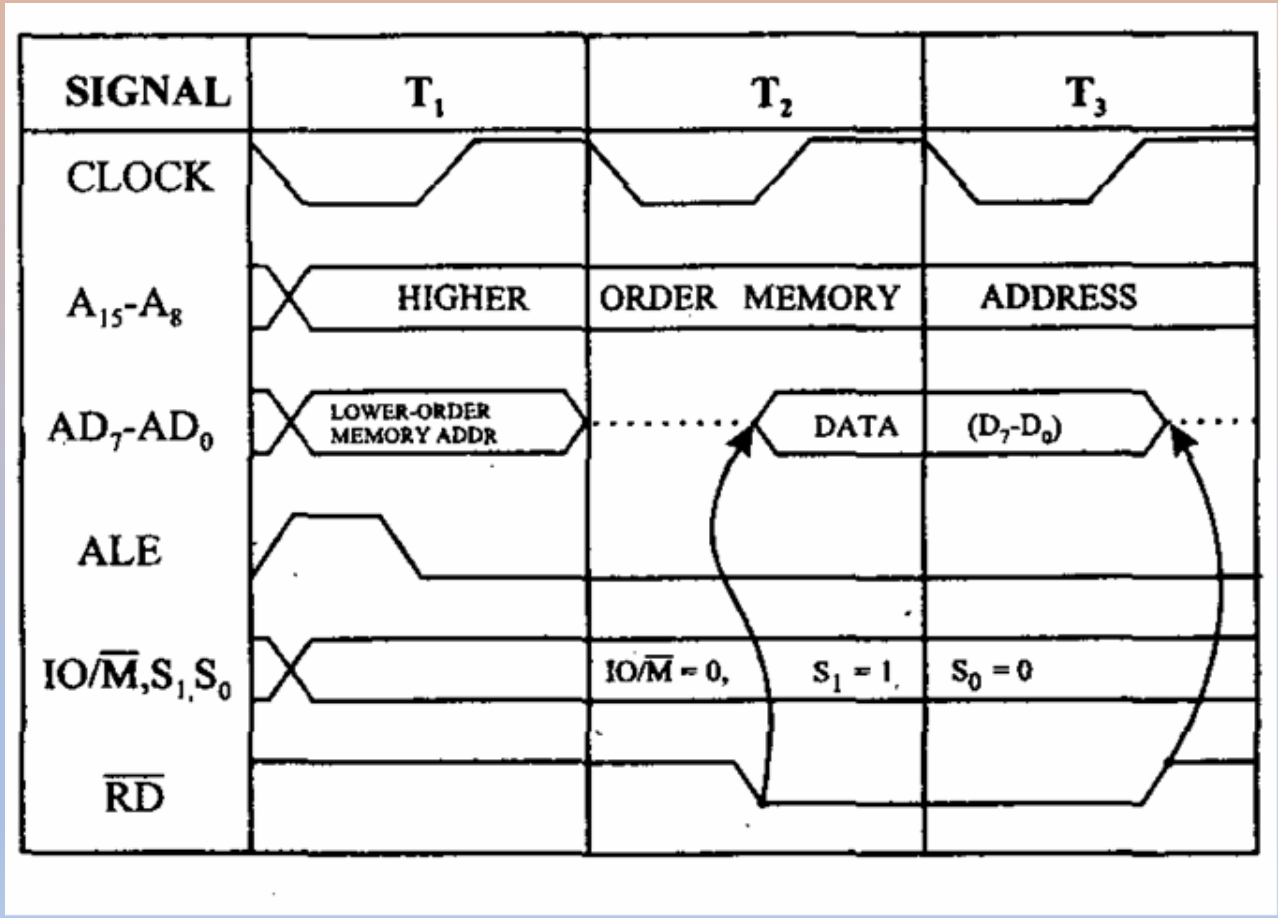
OPCODE FETCH MACHINE CYCLE OF 8085



OPCODE FETCH MACHINE CYCLE OF 8085

- 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.

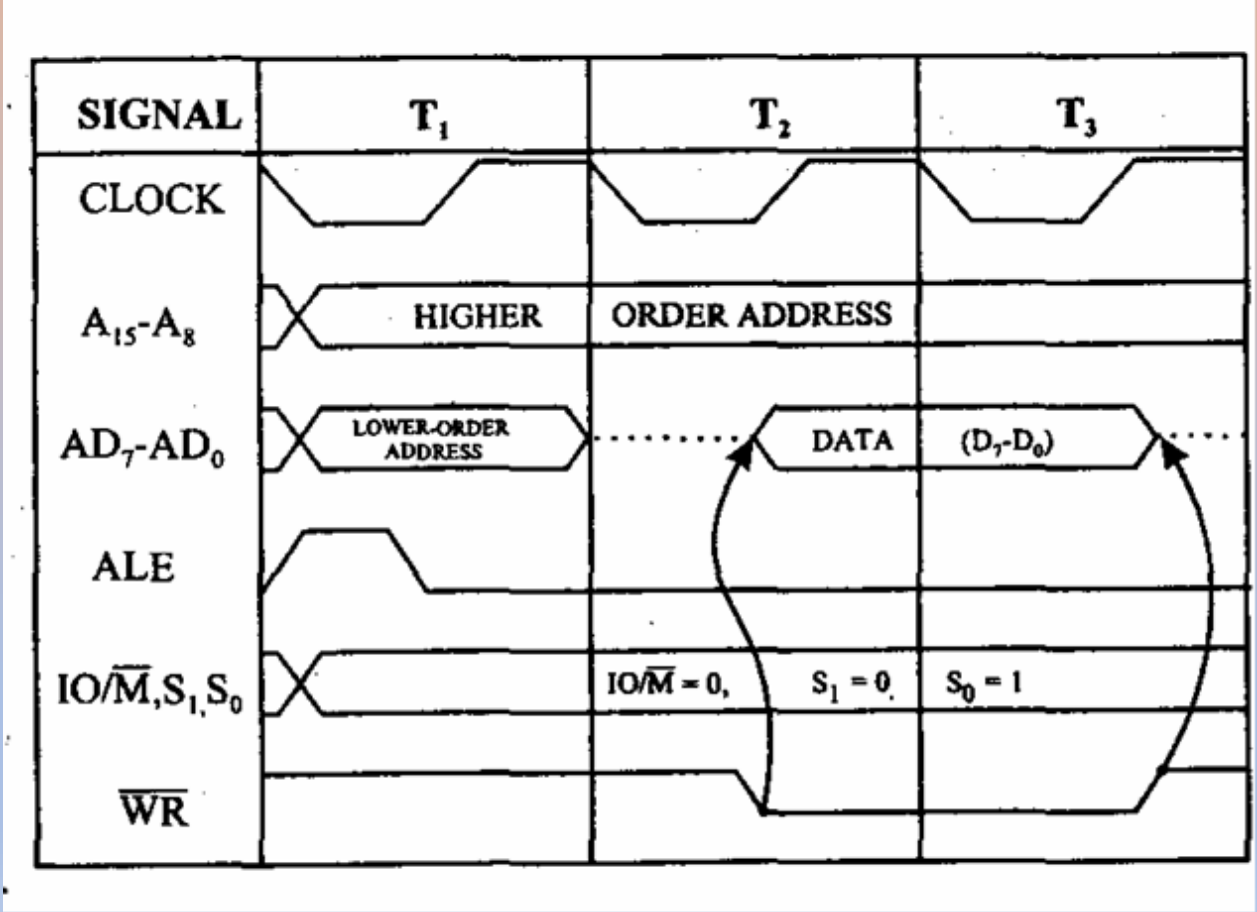
MEMORY READ MACHINE CYCLE OF 8085



MEMORY READ MACHINE CYCLE OF 8085

- The memory read machine cycle is executed by the processor to read a data byte from memory.
- The processor takes 3T states to execute this cycle
- The instructions which have more than one byte word size will use the machine cycle after the opcode fetch machine cycle.

MEMORY WRITE MACHINE CYCLE OF 8085



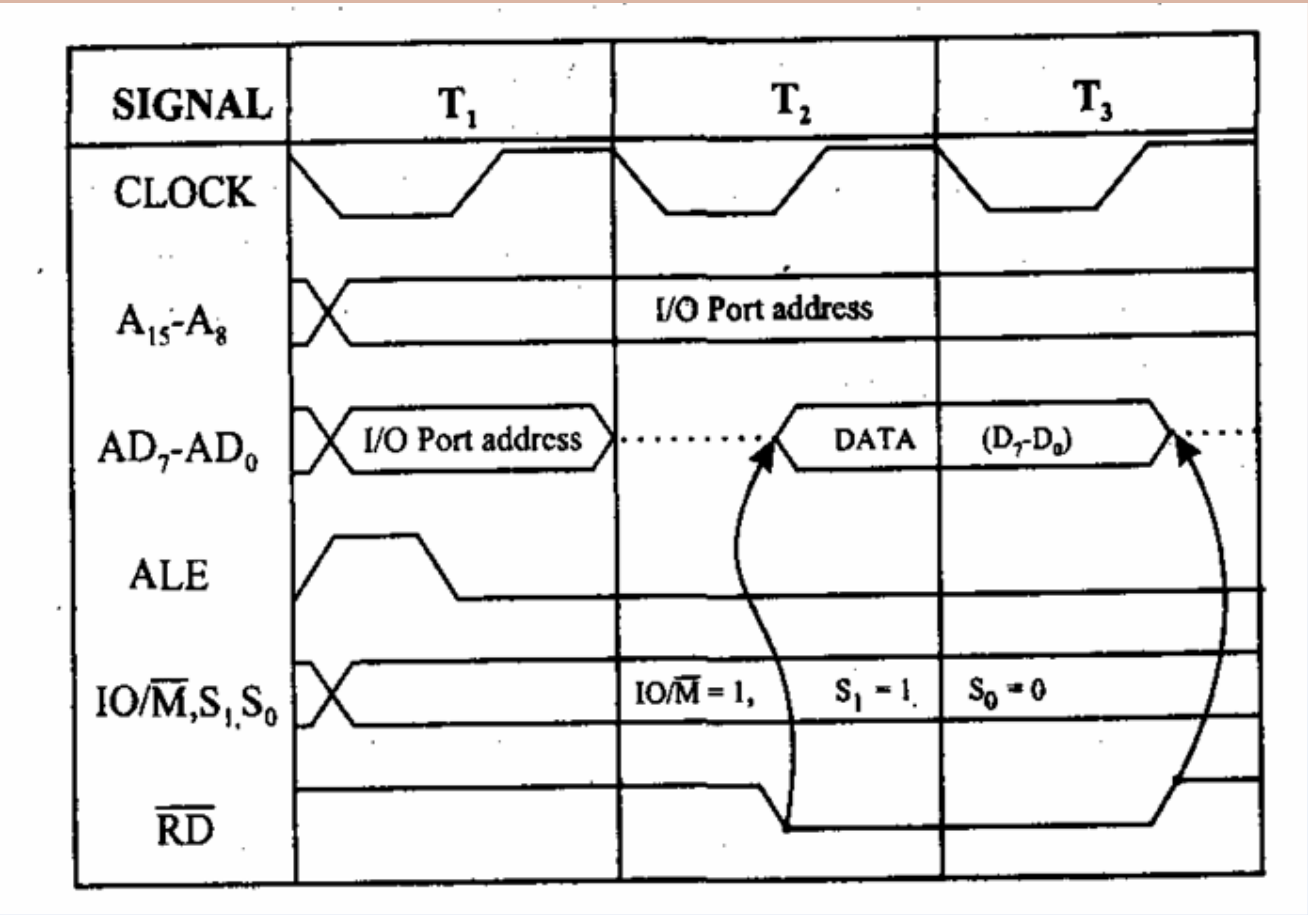
MEMORY WRITE MACHINE CYCLE OF 8085

- The memory write machine cycle is executed by the processor to write a data byte in a memory location.
- The processor takes, 3T states to execute this machine cycle

I/O READ CYCLE OF 8085

- The I/O Read cycle is executed by the processor to read a data byte from I/O port or from the peripheral.
- The processor takes 3T states to execute this machine cycle.
- The IN instruction uses this machine cycle during the execution.

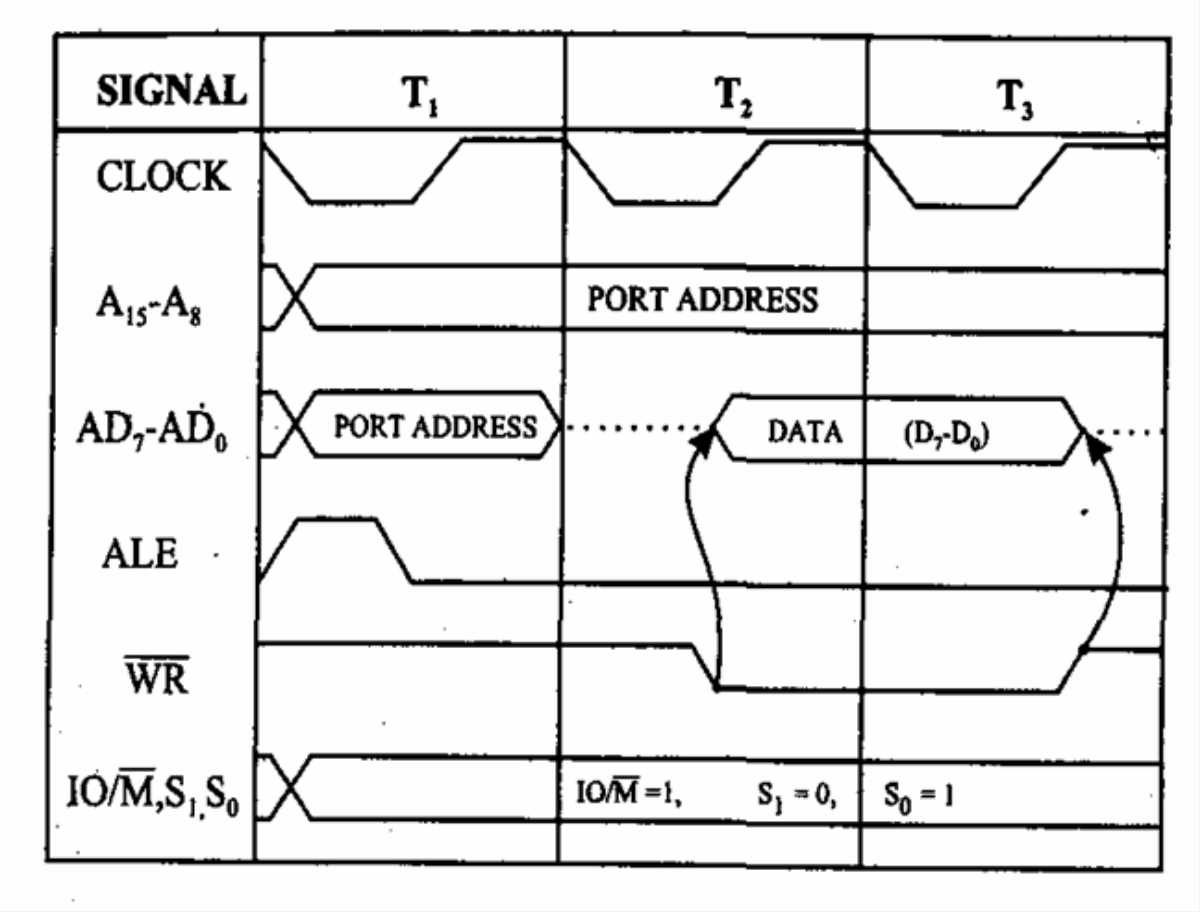
I/O READ CYCLE OF 8085



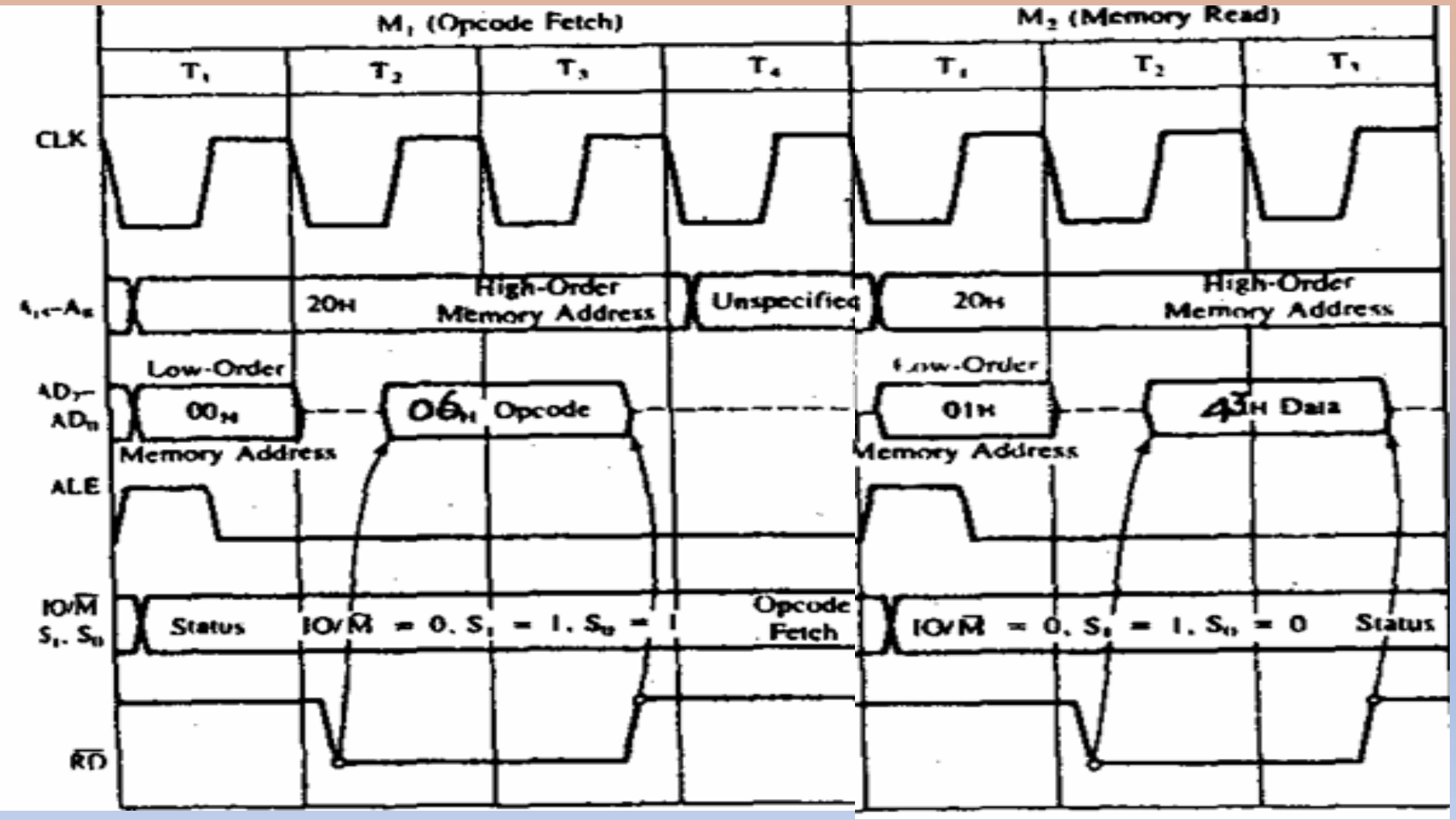
I/O WRITE CYCLE OF 8085

- The I/O write machine cycle is executed by the processor to write a data byte in the I/O port or to a peripheral, which is I/O, mapped in the system.
- The processor takes, 3T states to execute this machine cycle.

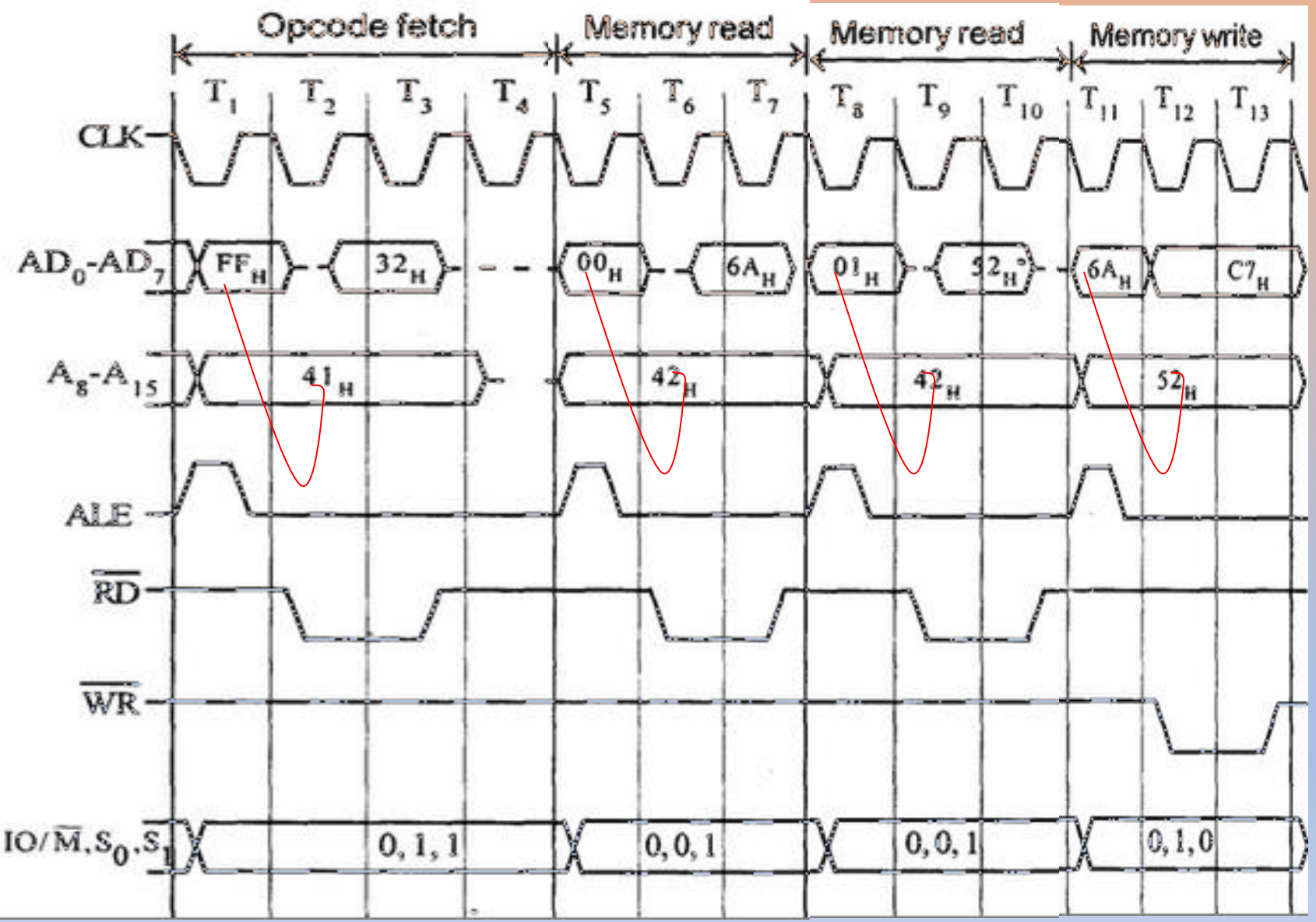
I/O WRITE CYCLE OF 8085



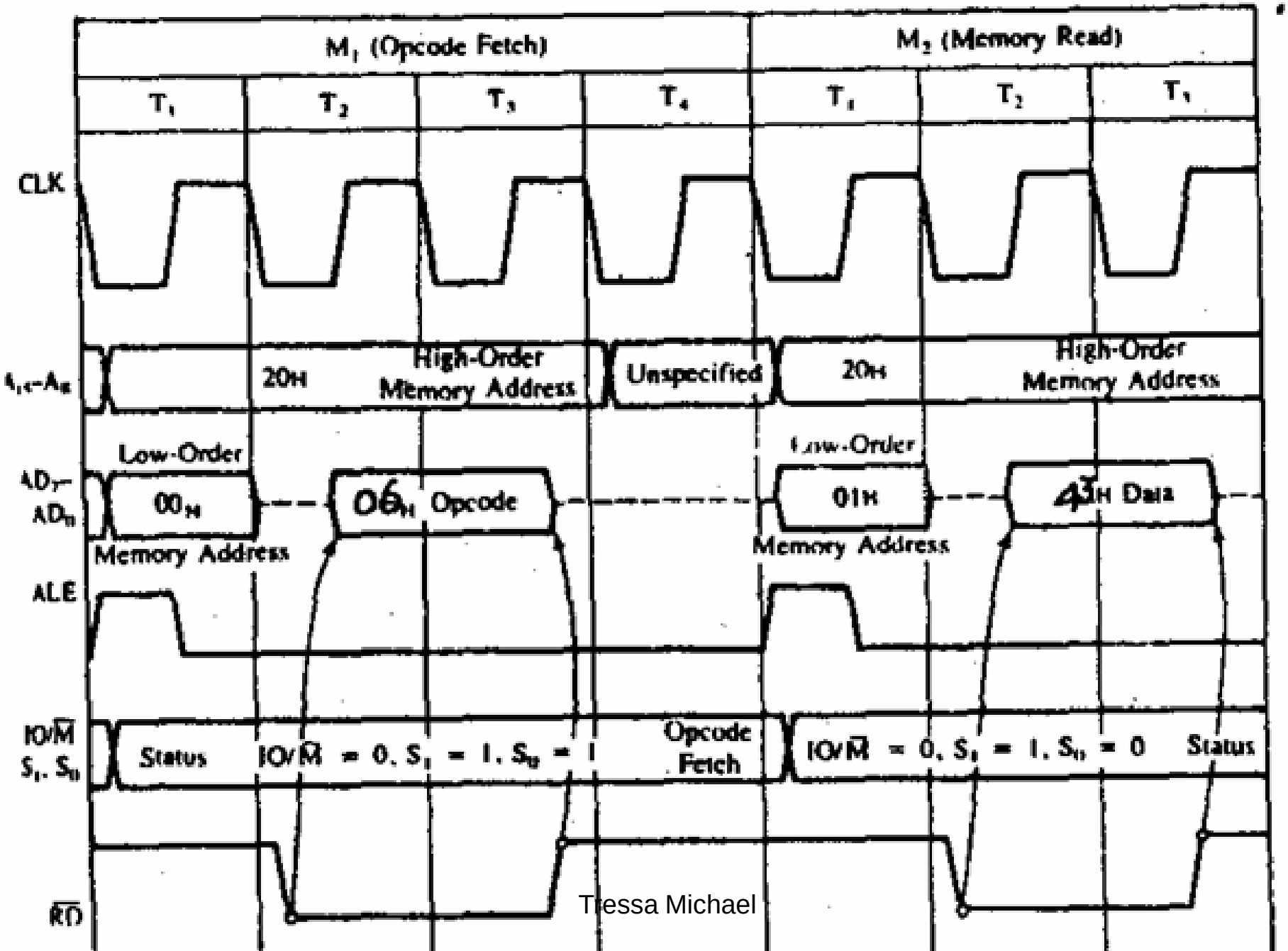
EXAMPLE INSTRUCTION : MVI B, 43



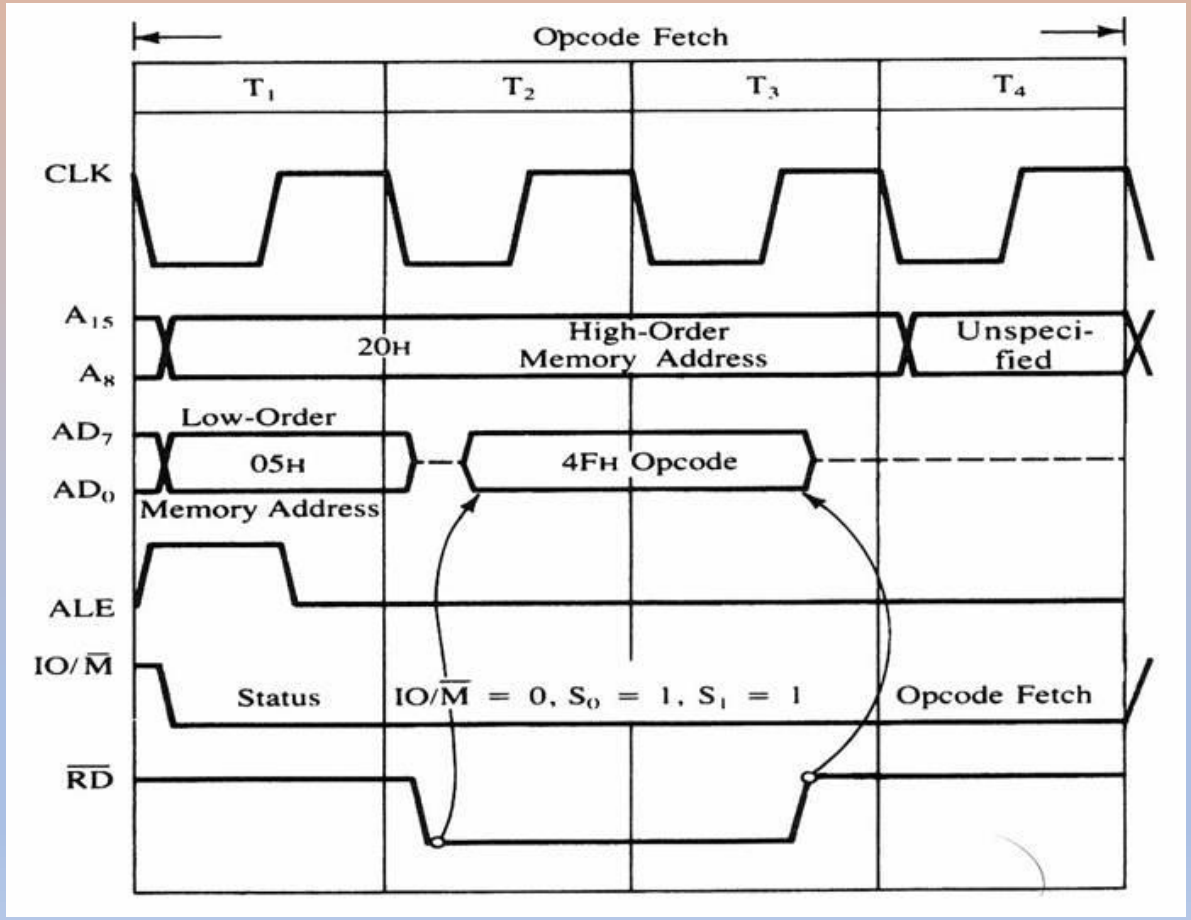
EXAMPLE INSTRUCTION : STA 526A



MVI B, data

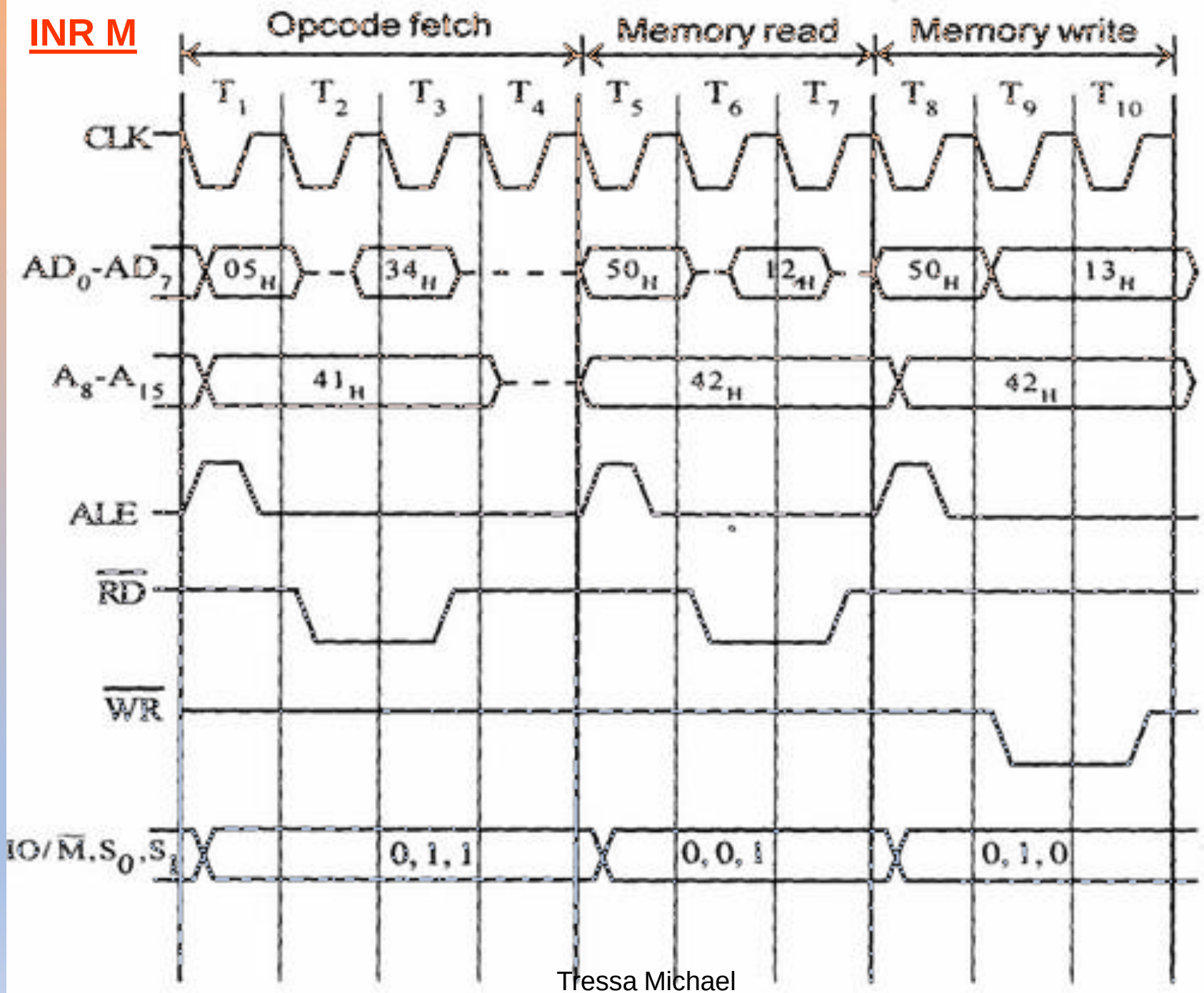

M1 (Opcode Fetch)
M2 (Memory Read)
T1
T2
T3
T4
T1
T2
T3
CLK
A15–A8
20H
High-Order Memory Address
Unspecified
20H
High-Order Memory Address
AD7–AD0
Low-Order Memory Address
00H
06H Opcode
Low-Order Memory Address
01H
43H Data
ALE
IO/M S1, S0
Status
IO/M = 0, S1 = 1, S0 = 1
Opcode Fetch
IO/M = 0, S1 = 1, S0 = 0
Status
RD

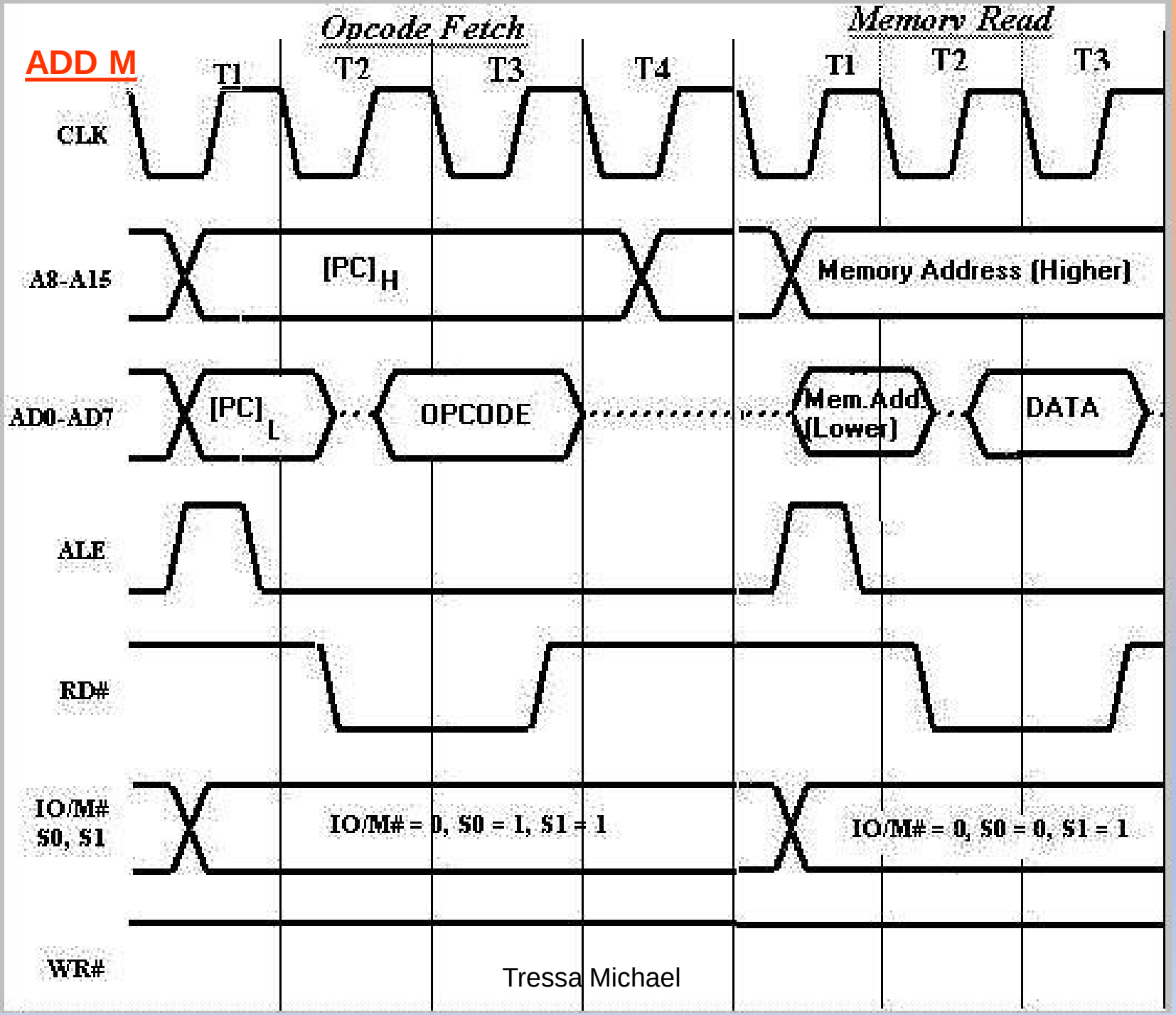
8085 timing diagram for Opcode fetch cycle for MOV C, A .



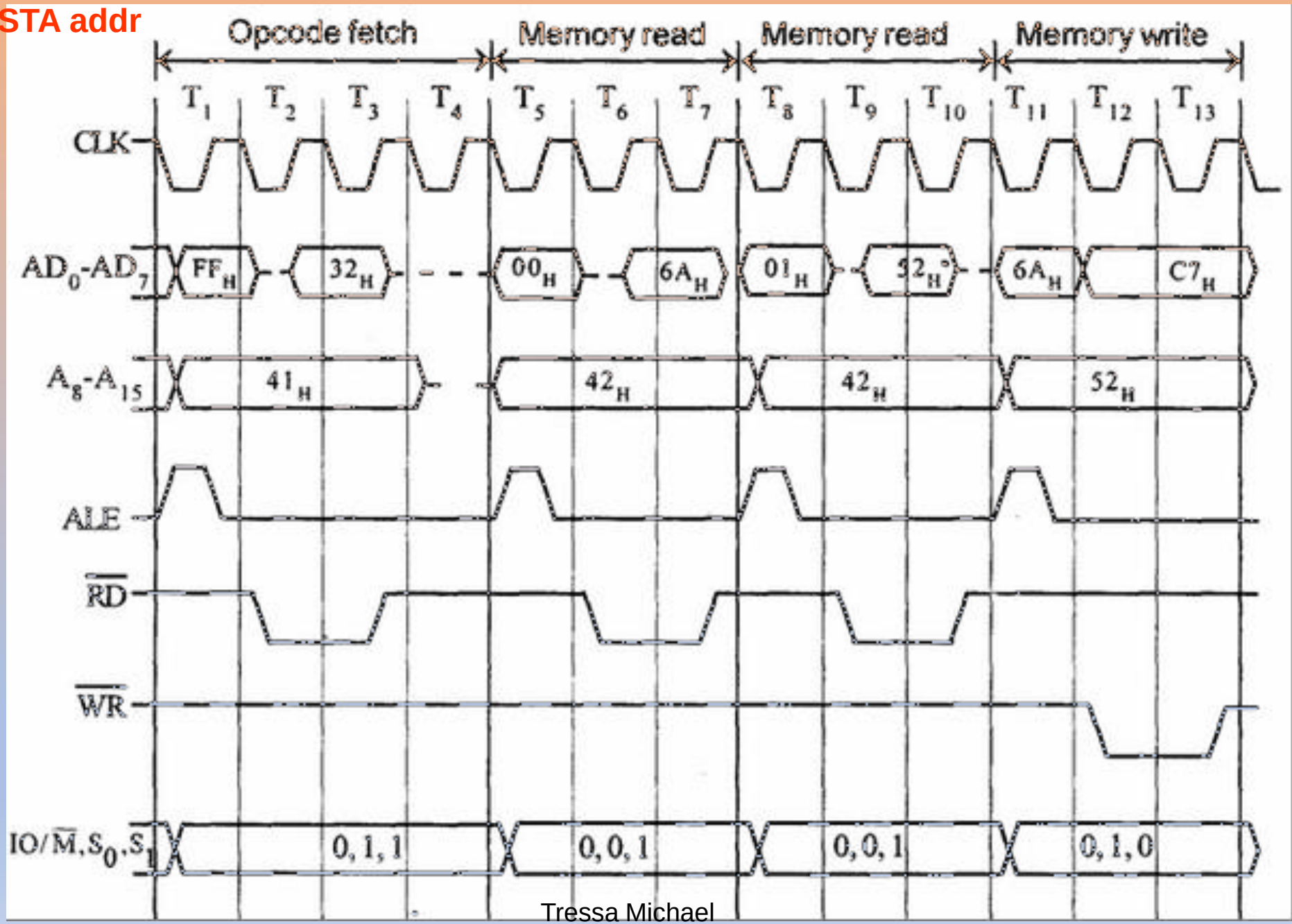
8085 timing diagram for Opcode fetch cycle for MOV C, A .

INR M

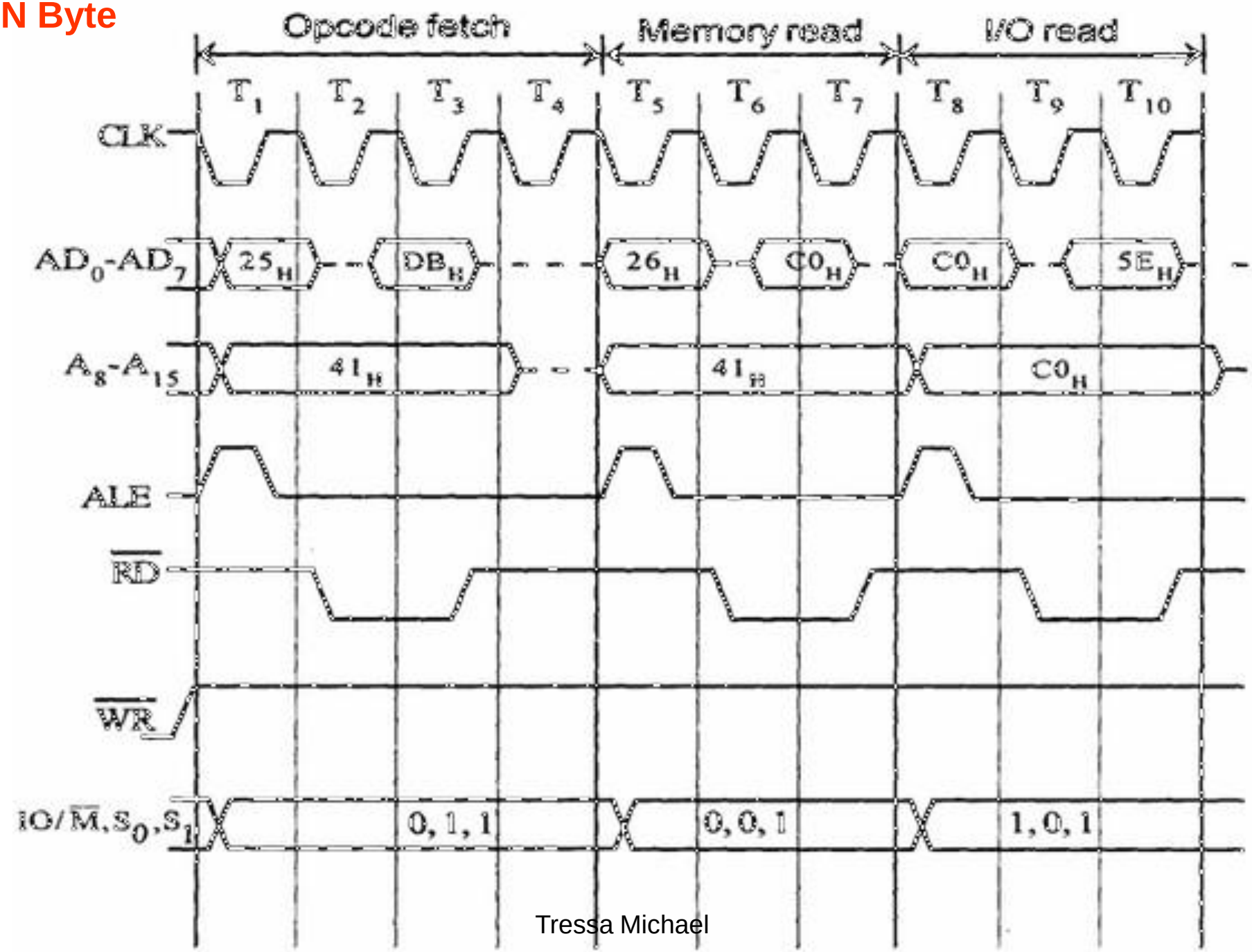

Opcode fetch
Memory read
Memory write
T1 T2 T3 T4 T5 T6 T7 T8 T9 T10
CLK
AD0-AD7
05H
34H
50H
12H
50H
13H
A8-A15
41H
42H
42H
ALE
RD
WR
IO/M, S0, S1
0, 1, 1
0, 0, 1
0, 1, 0


ADD M
Opcode Fetch
Memory Read
T1
T2
T3
T4
T1
T2
T3
CLK
A8-A15
[PC]H
Memory Address (Higher)
AD0-AD7
[PC]L
OPCODE
Mem.Add.
(Lower)
DATA
ALE
RD#
IO/M#
S0, S1
IO/M# = 0, S0 = 1, S1 = 1
IO/M# = 0, S0 = 0, S1 = 1
WR#

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 32H 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

IN Byte


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

Thank You