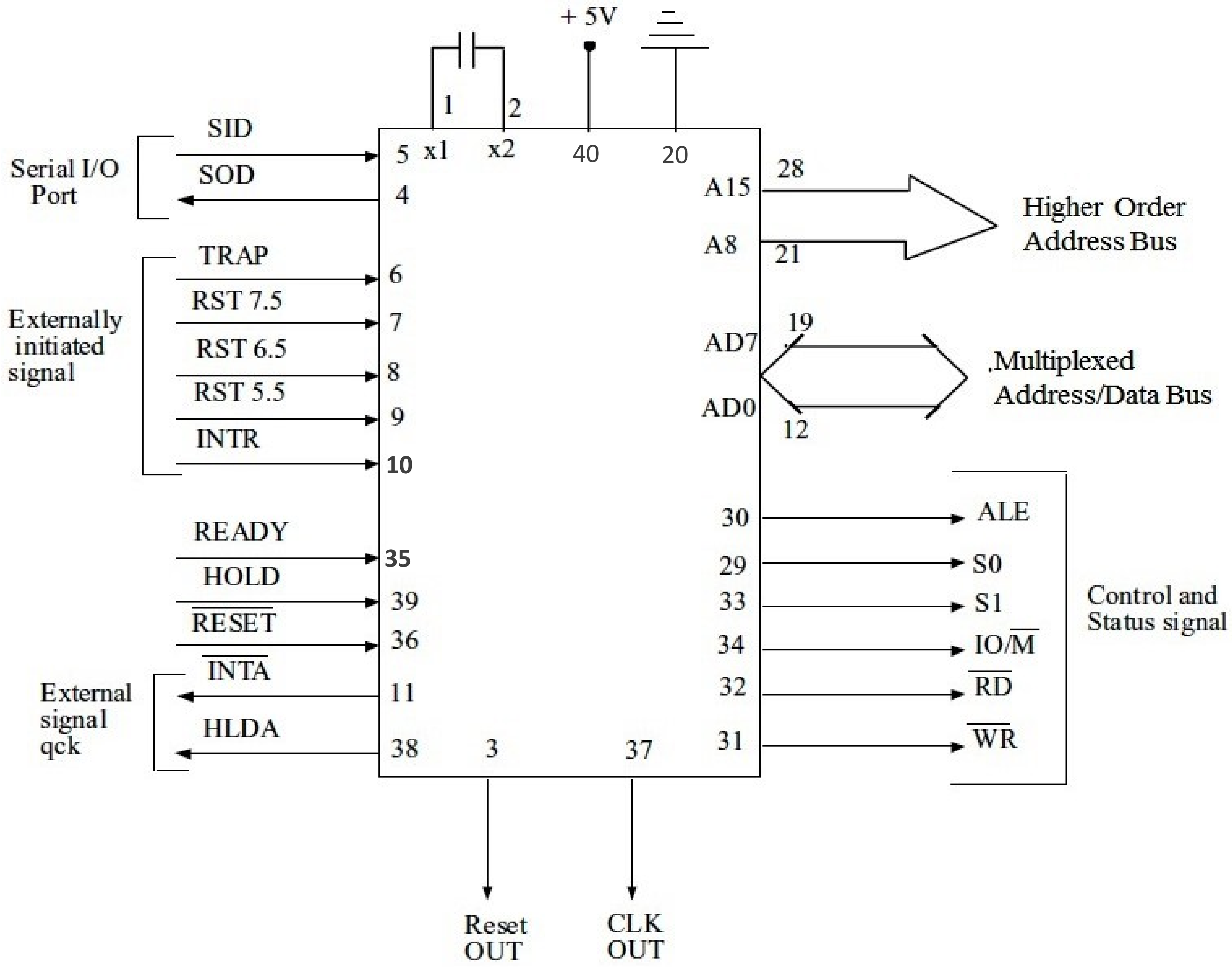
Chapter 4

Microprocessor System

8085 Microprocessor Pin Diagram



8085 Microprocessor Pin Diagram

- Among the 40 pins of 8085 μp, they are categorized into following types:
  - Unidirectional higher order address pins. (A_8 – A_{15}) – 8
  - Bidirectional multiplexed Address/Data pins (AD_0 –AD_7) – 8
  - Control input pins (11):
    - 8 externally initiated signals
    - 2 clock inputs XI & X2
    - 1 SID
  - Control output pins (11):
    - 6 control and status pins
    - 2 external acknowledgement signals
    - 1 CLK OUT +
    - 1 RESET OUT
    - 1 SOD
  - power supply & ground pins. – 2

8085 Microprocessor Pin Diagram

- ***Clock inputs (X_1 & X_2)***

  These pins are used as source clock by the 8085A μp to perform its internal operation. A crystal oscillator of 6.06 MHz is used a frequency generator source so that the 8085A μp works at half frequency 3.03MHz
- ***Power Supply (Vcc) and Ground (Vss)***

  Vcc: A TTL compatible +5V power supply to 8085A up.

  Vss: A ground reference.
- ***Higher order Address Pins (A8 –A15):***

  These are the unidirectional pins that always carry the higher byte of 16-bits memory address.
- ***Lower multiplexed Address/Data Bus (AD0 – AD7):***

  These are the bidirectional pins that carry either lower byte of 16-bits memory address, or an 8-bits I/O address or an 8-bits data in/out of microprocessor.
  - If ALE = 0, then AD_0 – AD_7 acts as a bidirectional data bus D_0 – D_7.
  - If ALE = 1, then AD_0 – AD_7 acts as a unidirectional address bus A_0 – $A_{7.}$

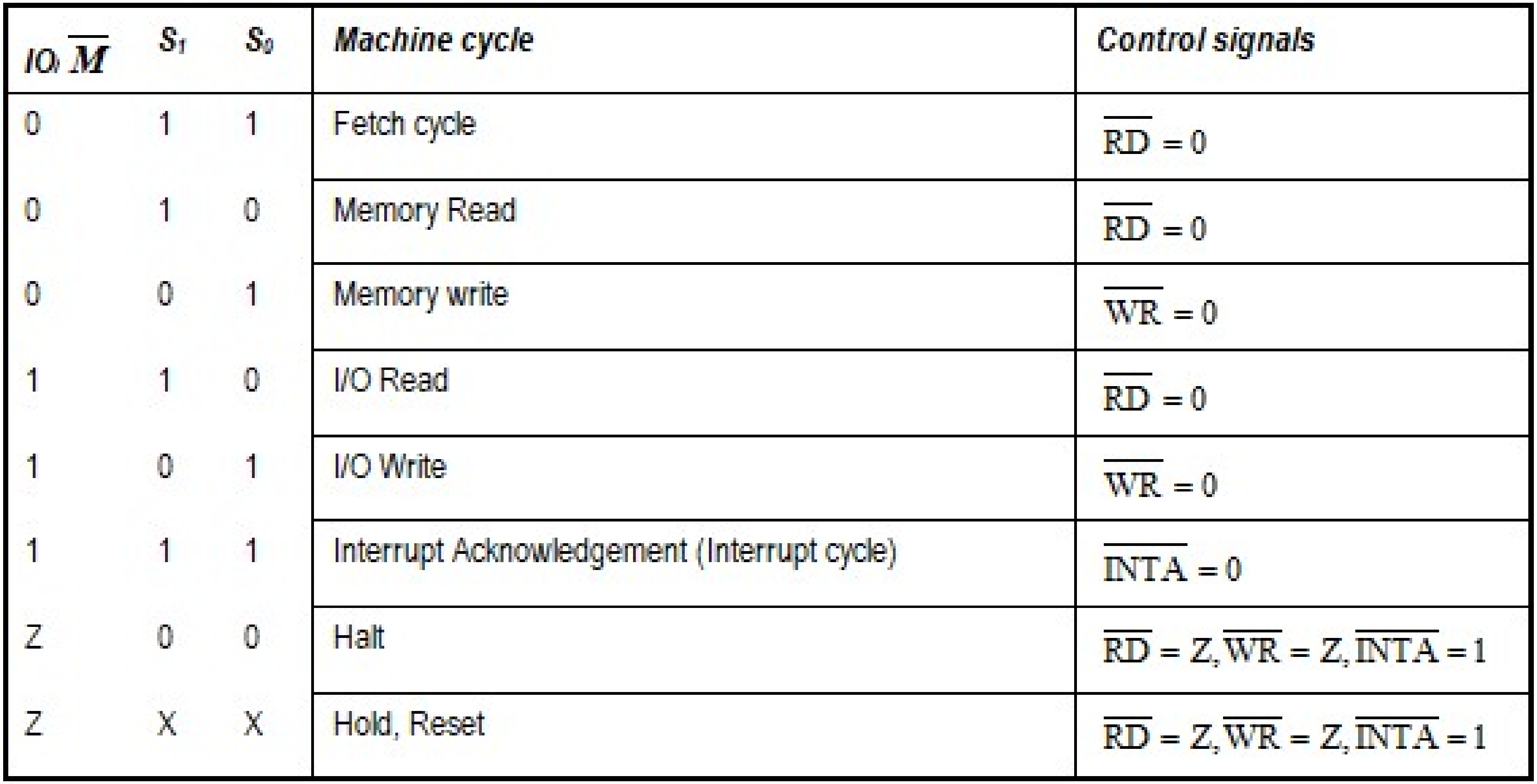
8085 Microprocessor Pin Diagram

- ***Control and status signals:***
  - ***ALE (Address Latch Enable) :*** is used to enable the multiplexed AD_0 – AD_7 bus as an address bus
  - ***$\overline{RD}$ (Read):*** It is a tri-state active low output signal and is made low to read a data from I/O device a memory unit.
  - ***$\overline{WR}$ (Write):*** It is a tri-state active low output signal and is made low to write or display the data present in data bus to I/O device or memory unit.
  - ***$IO/\overline{M}$:*** It is a tri-state output signal and holds the status of selection of I/O device or memory unit.

    If $IO/\overline{M}$ = 0 then memory device is selected.

    If $IO/\overline{M}$ = 1 then I/O device is selected.
  - ***S_1 & S_0:*** They hold the status of machine cycle during the internal operation of microprocessor as shown in table

8085 Microprocessor Pin Diagram



IO/$\overline{M}$	S_1	S_0	*Machine cycle*	*Control signals*
0	1	1	Fetch cycle	$\overline{RD} = 0$
0	1	0	Memory Read	$\overline{RD} = 0$
0	0	1	Memory write	$\overline{WR} = 0$
1	1	0	I/O Read	$\overline{RD} = 0$
1	0	1	I/O Write	$\overline{WR} = 0$
1	1	1	Interrupt Acknowledgement (Interrupt cycle)	$\overline{INTA} = 0$
Z	0	0	Halt	$\overline{RD} = Z, \overline{WR} = Z, \overline{INTA} = 1$
Z	X	X	Hold, Reset	$\overline{RD} = Z, \overline{WR} = Z, \overline{INTA} = 1$

8085 Microprocessor Pin Diagram

- ***Serial I/O pins:***
  - ***SID (Serial Input Data):*** It is used to accept the serial data from external device through the MSB of accumulator
  - ***SOD (Serial Output Data):*** It is used to output the data present in accumulator starting from MSB serially to external device
- ***Externally Initiated Signals:***
  - Interrupt signals: Interrupt signals are issued to the microprocessor by the external devices for CPU time for their service. Interrupt is a kind of disturbance to the microprocessor created by an external device for it to be served by microprocessor.

    INTR: This pin is used to let the microprocessor know whether it has received any interrupts. The INTR is a maskable interrupt and goes high when one or more external devices send the interrupts (s) to microprocessor.

    RST 5.5, RST 6.5, RST 7.5: They are maskable interrupt pins used by different devices for different purposes.

    TRAP: It is a highest-priority non-maskable interrupt and is used for operating the microprocessor in single-step mode.

8085 Microprocessor Pin Diagram

- ***Externally Initiated Signals:***
  - ***READY:*** It is used by the slower devices to indicate their readiness while communicating with a faster device microprocessor. It is made high by an external device informing microprocessor that it is ready to send/receive a data.
  - ***HOLD:*** It is used to take hold or control of system bus by external device (S/O or memory) for direct memory access (DMA). It is done by issuing a high HOLD signal to the CPU (microprocessor)
  - ***$\overline{\text{RESET IN}}$:*** It is made low to reset all the CPU registers to their default/random values, to reset different interrupts and to place data bus, address bus & control lines in a high impedance state.

8085 Microprocessor Pin Diagram

- ***External Acknowledgement Signals.***
  - ***$\overline{\text{INTA}}$ (Interrupt Acknowledge):*** When an external device sends an interrupt to the microprocessor, it sends an active low INTA signal to that device as a response indicating that the microprocessor is coming to serve that particular external device.
  - ***HLDA (Hold Acknowledge):*** It is an answer to the HOLD signal. The CPU issues an active high HLDA signal to can external device (which has issued a signal as a request for system bus) saying that the system bus is granted for DMA process.
- ***RESET OUT and CLK OUT signals:***
  - ***RESET OUT:*** This pin is used by microprocessor to reset the peripheral chips or devices connected to the microprocessor by making it HIGH. An active low RESET OUT pin indicates that the microprocessor is in reset condition.
  - ***CLK OUT:*** It is used to give clock input to the other peripheral devices or chips connected to microprocessor.

Bus Structure

- ***Synchronous Bus***

In a synchronous bus, the occurrence of the events on the bus is determined by a clock . The clock transmits a regular sequence of 0’s & 1’s of equal duration. All the events start at beginning of the clock cycle.

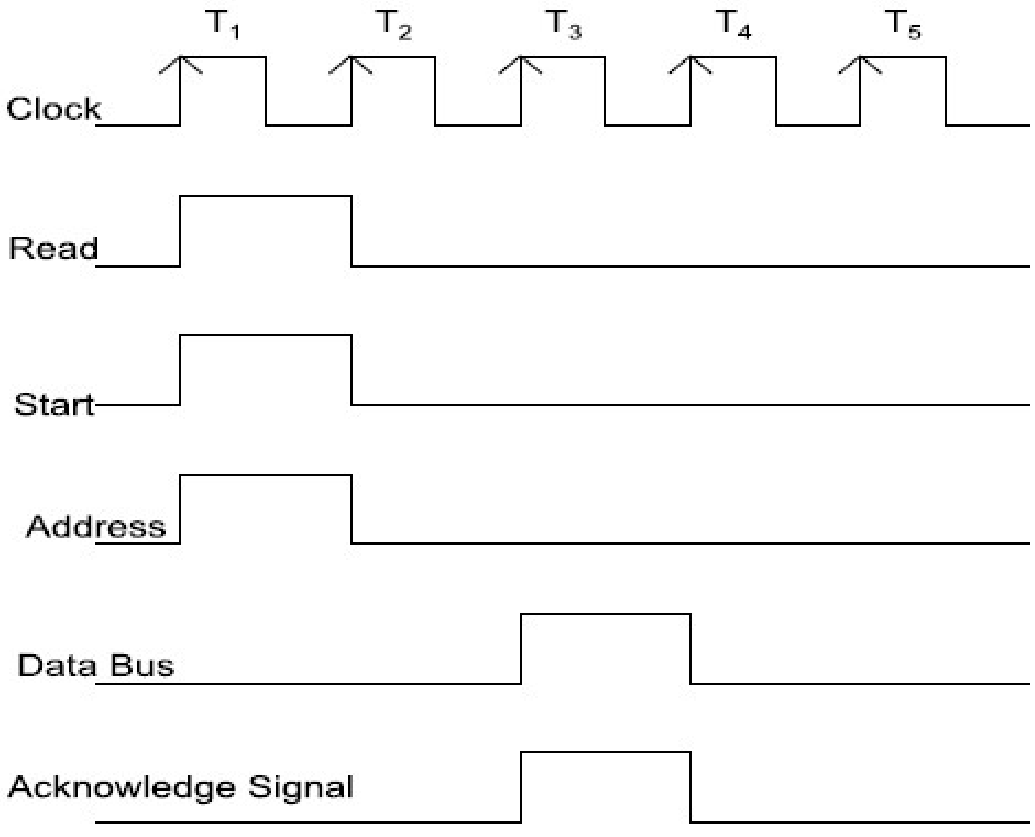


Fig: Synchronous Read Operation

Bus Structure

- ***Synchronous Bus***
  - Here the CPU issues a START signal to indicate the presence of address and control information on the bus.
  - Then it issues the memory read signal and places the memory address on the address bus.
  - The addressed memory module recognizes the address and after a delay of one clock cycle it places the data and acknowledgment signal on the buses.
  - In synchronous bus, all devices are tied to a fixed rate, and hence the system can not take advantage of device performance but it is easy to implement.

Bus Structure

- ***Asynchronous Bus***
  - In an asynchronous bus, the timing is maintained in such way that occurrence of one event on the bus follows and depends on the occurrence of previous event.

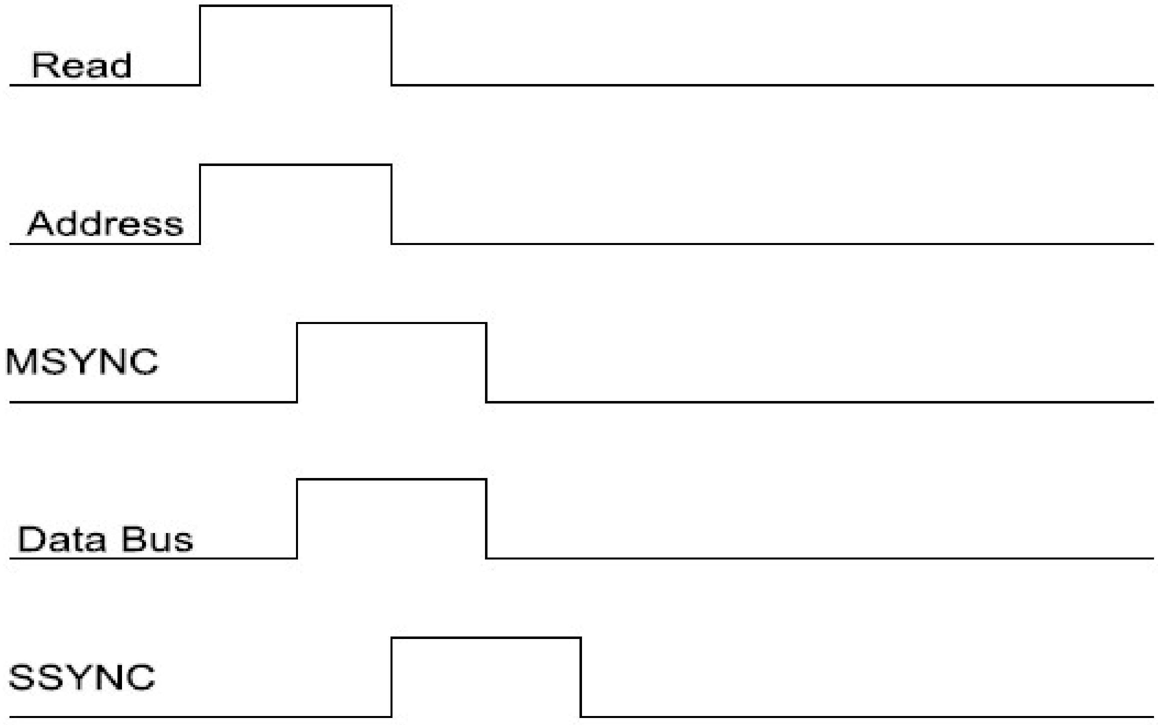


Fig: Asynchronous Read Operation

Bus Structure

- ***Asynchronous Bus***
  - Here the CPU places Memory Read (Control) and address signals on the bus.
  - Then it issues master synchronous signal (MSYNC) to indicate the presence of valid address and control signals on the bus.
  - The addressed memory module responds with the data and the slave synchronous signal (SSYNC)

Register Transfer Language

- The language, which is basically used to express the transfer of data among the registers, is called Register Transfer Language (RTL). In such transfer, one of the source or destination should be register (not necessarily both).
- The RTLs are basically used to represent the transfer of data in different operations and micro-operations. Since each micro-operation specifies the transfer of data into or out of a register, the RTLs are best used to represent the different micro-operations of an instruction.

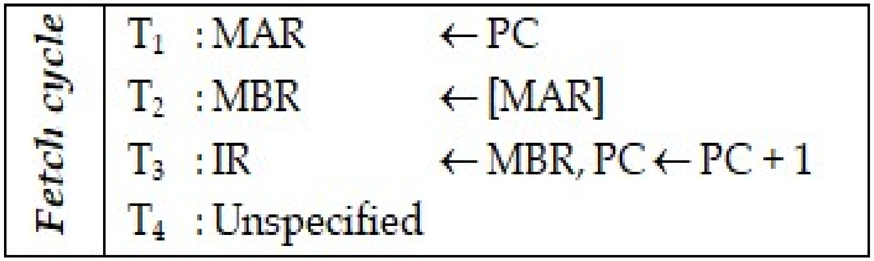
Register Transfer Language

- **While writing RTLs you should have a prior knowledge on the following registers.**
  - ***Memory Address Register (MAR)*** *is a register that always holds the memory address before reading a data from memory or before writing a data into memory. That means, it acts as a memory address bridge between memory and microprocessor.*
  - ***Input Output Address Register (IOAR)*** *holds the I/O address which is same as that of MAR in working principle.*
  - ***Memory Buffer Register (MBR)*** *is a register that always holds the data after reading from the memory or before writing into the memory. That means, it acts as a data bridge between memory and microprocessor.*
  - ***Input Output Buffer Register (IOBR)*** *holds the data before/after writing/reading to/from I/O which is same as that of MBR in working principle.*
  - ***Program Counter (PC)*** *is a 16-bits register that always holds the address of next instruction.*
  - ***Instruction Register (IR)*** *is an 8-bits register that always holds an opcode of an instruction after fetching it from memory.*

RTL and Timing Diagram of Fetch Cycle

- Fetch cycle is the total time required to read on opcode from memory, place that opcode into instruction register (IR) and decode that to recognize the corresponding instruction. It requires 4 T-states.

Address	opcode
8000	3E



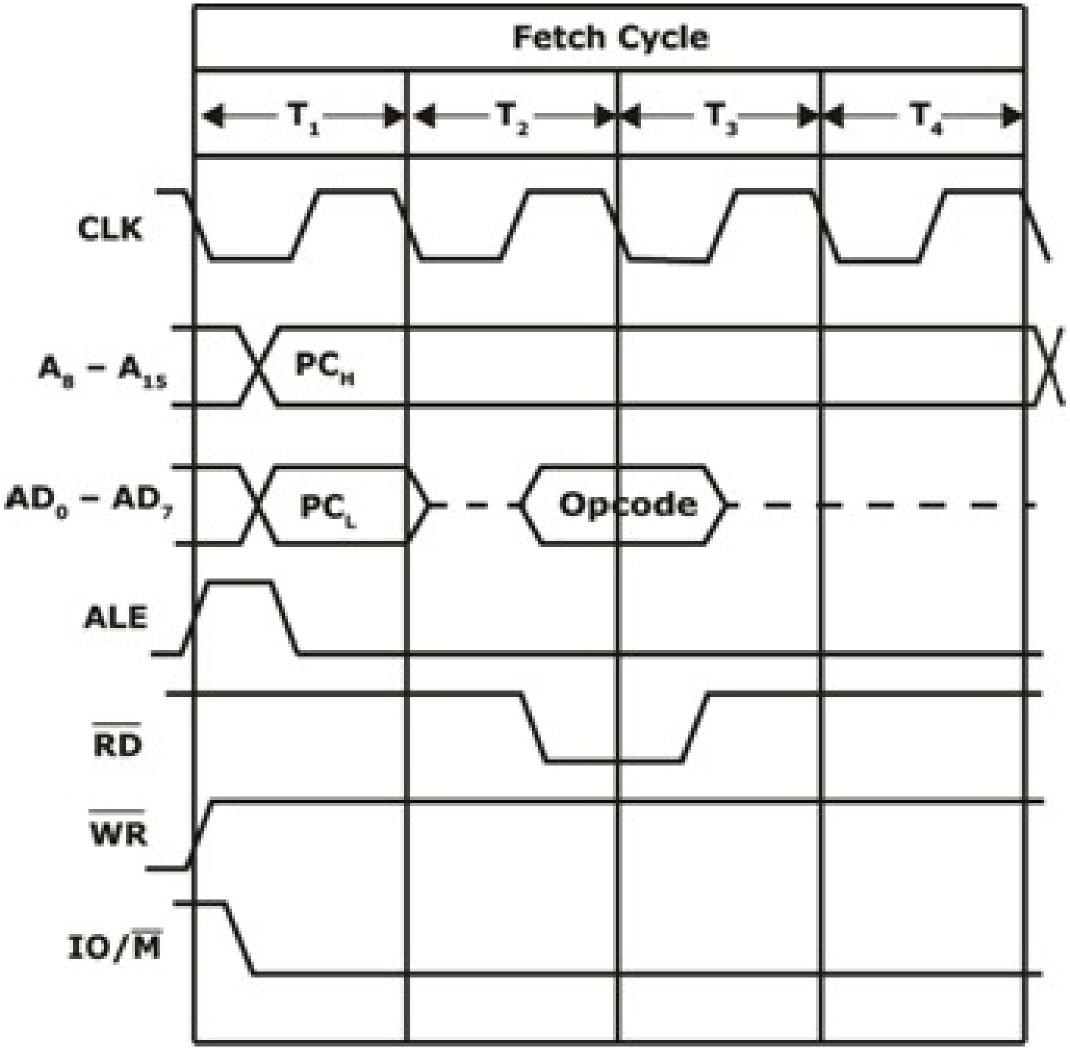
Fetch cycle

T_1 : MAR ← PC

T_2 : MBR ← [MAR]

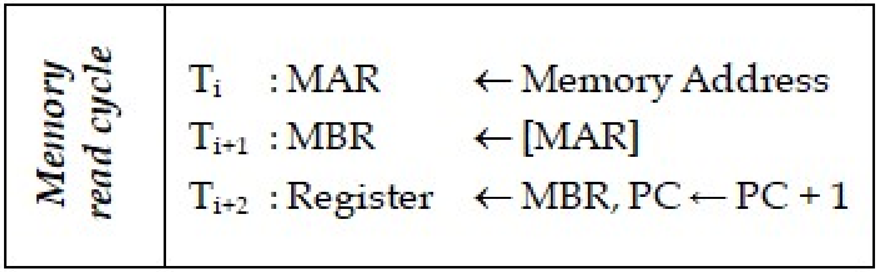
T_3 : IR ← MBR, PC ← PC + 1

T_4 : Unspecified

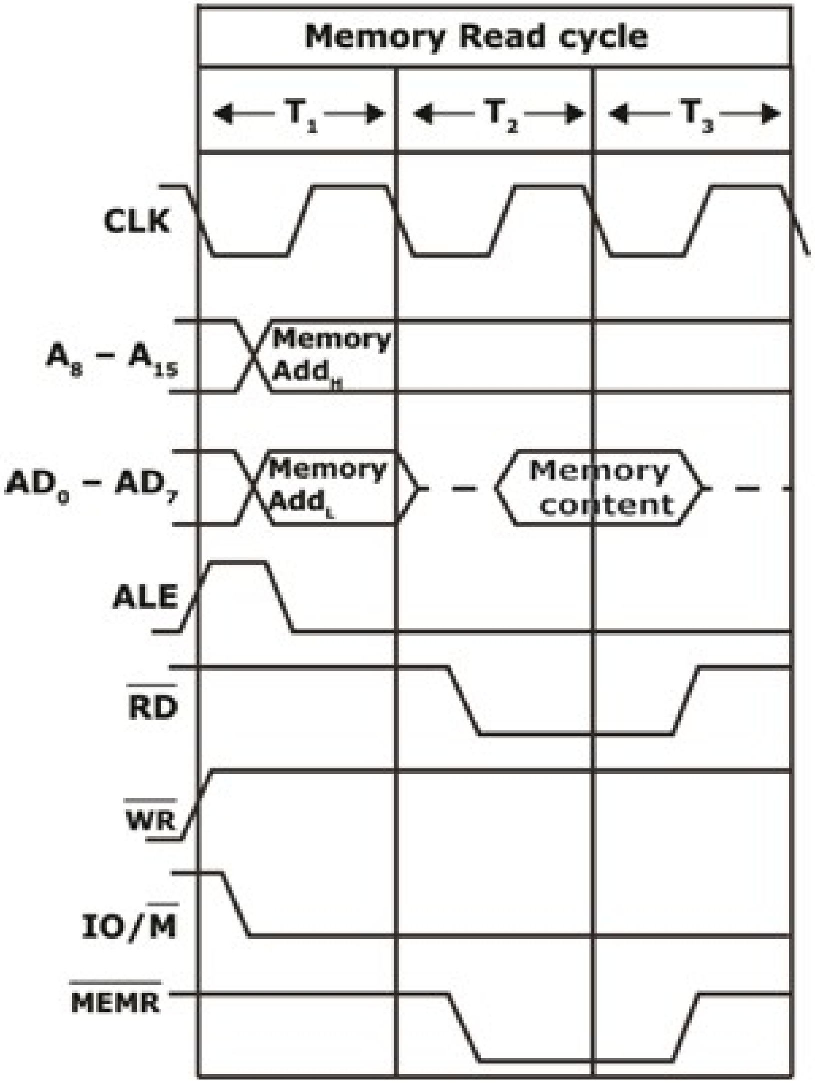


RTL and Timing Diagram of Memory Read Cycle

- Memory read cycle is the total time required to read a data (byte) from memory and place that into a register within the microprocessor. It requires 3 T-states as shown below using RTLs.

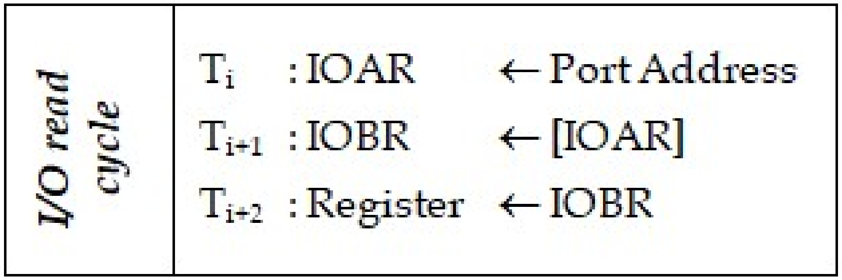


Memory read cycle	T_i	: MAR	← Memory Address
	T_{i+1}	: MBR	← [MAR]
	T_{i+2}	: Register	← MBR, PC ← PC + 1

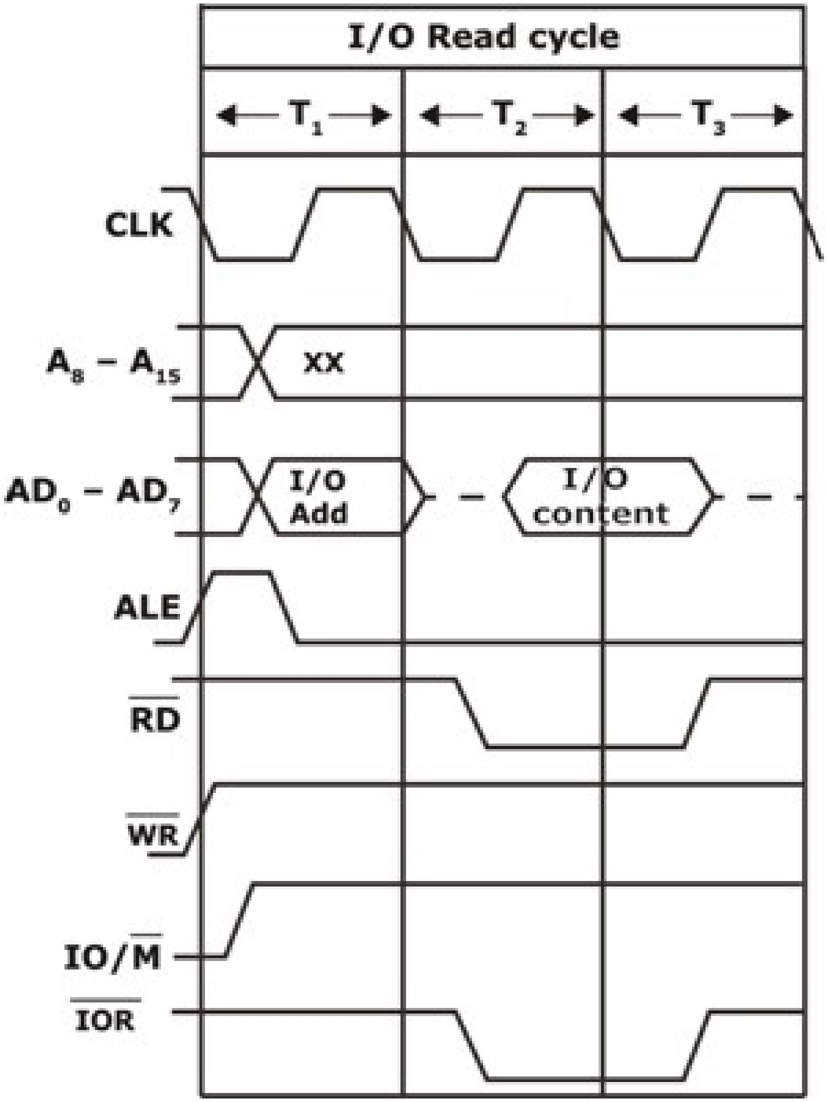


RTL and Timing Diagram of I/O Read Cycle

- I/O read cycle is the total time required to read a data (byte) from an input port (device) . It takes 3 T-states as shown below using RTLs.

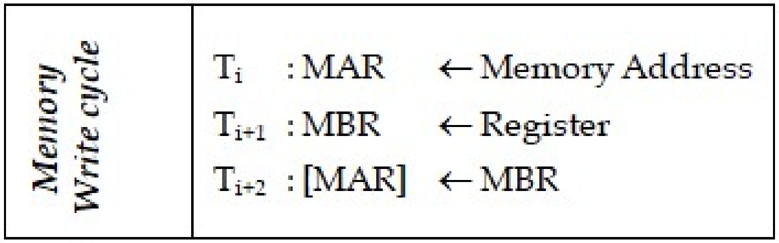


I/O read cycle	T_i	: IOAR	← Port Address
	T_{i+1}	: IOBR	← [IOAR]
	T_{i+2}	: Register	← IOBR

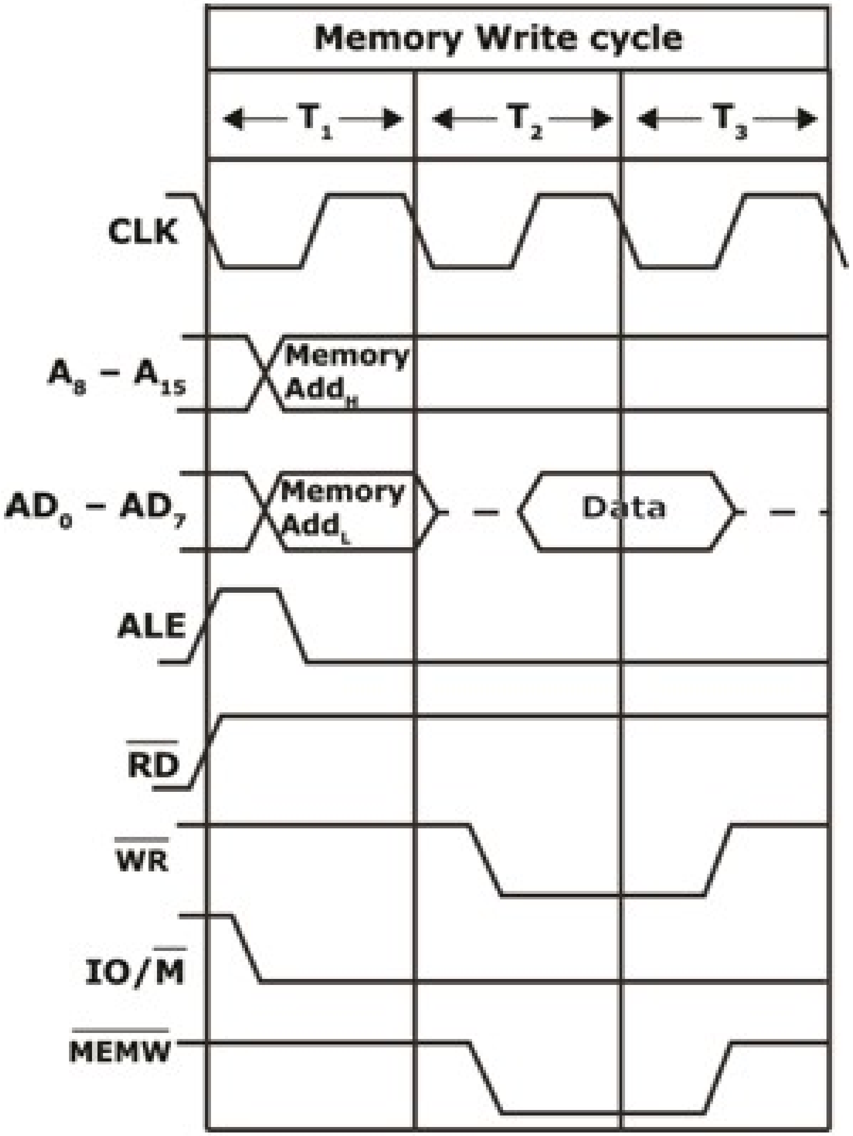


RTL and Timing Diagram of Memory Write Cycle

- Memory write cycle is the total time required to write a data (byte) into a memory location. It takes 3 T-states as shown below using RTLs

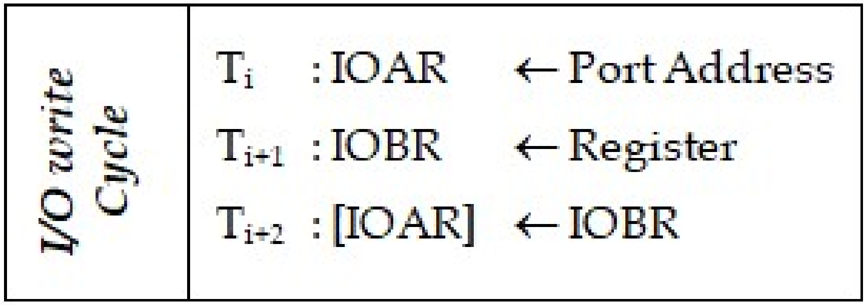


Memory Write cycle	T_i	:MAR	← Memory Address
	T_{i+1}	:MBR	← Register
	T_{i+2}	:[MAR]	← MBR

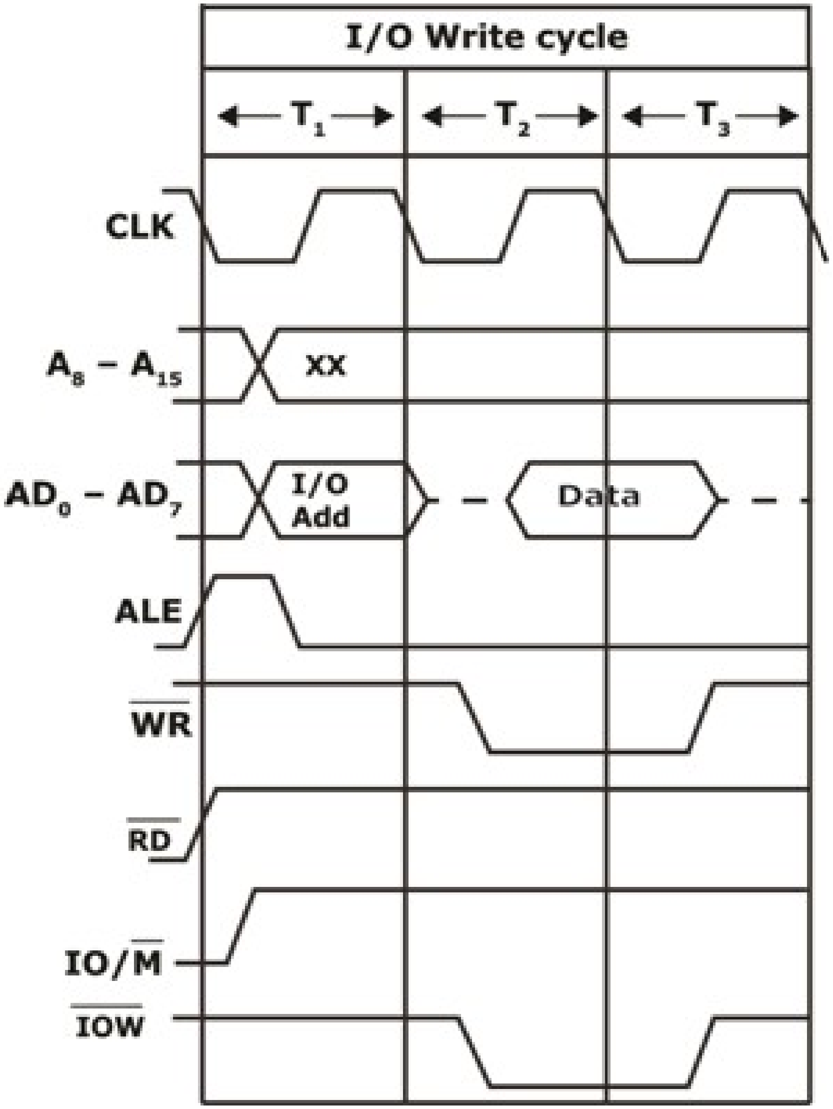


RTL and Timing Diagram of I/O Write Cycle

- I/O write cycle is the total time required to write or display a data (byte) into an output port (device). It takes 3 T-states as shown below using RTLs.



I/O write Cycle	T_i	: IOAR	← Port Address
	T_{i+1}	: IOBR	← Register
	T_{i+2}	: [IOAR]	← IOBR


I/O Write cycle
T_1
T_2
T_3
CLK
$A_8 - A_{15}$
XX
$AD_0 - AD_7$
I/O Add
Data
ALE
$\overline{WR}$
$\overline{RD}$
$IO/\overline{M}$
$\overline{IOW}$