Segmented Memory

Reason for Segmented Memory:

- ✓ 8086 has a 20-bit address bus. So it can address a maximum of 1MB of memory and each memory location is addressed by a 20 bit address.
- ✓ To hold a 20-bit address there must be a 20-bit address register available within processor but 8086 only has 16-bit registers. So 20-bit address can't be stored inside the 16-bit register. To avoid this problem segmented memory is used in 8086.

Total 1MB memory can be divided into some equal size segments each of having capacity 64 KB.
So max no of segments is 16. (1mb/64 kb=16)
8086 can work with only four 64KB segments at a time within this 1MB range.
Each location in a particular segment can be expressed by two addresses.

i) ***Segment Address (16 bit):*** It refers the starting address of a segment and it is fixed for whole of the segment.
ii) ***Offset or Displacement Address (16 bit):*** It refers the individual location in that segment and it is varied location wise.

By using these two addresses the 20 bit physical address can be calculated as below:

Physical address (20 bit) = [Segment Address (16 bit) * 10]$_H$ + Offset Address(16 bit)

According to this formula segment address is multiplied by 10 and is added to offset. This is equivalent to shifting of segment register content towards left 4 times so that four zero are added to right side (MSB) of the segment address and added with the offset address to get the physical address which is 20 bit.

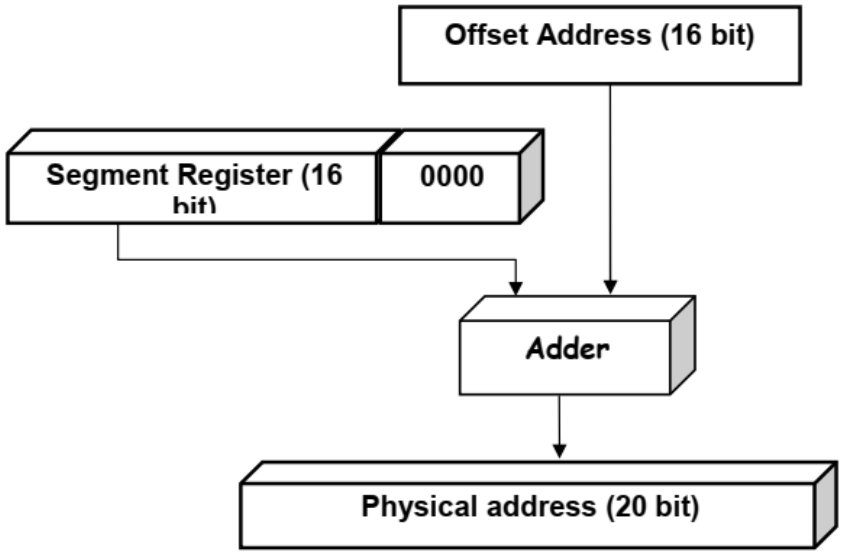


Figure 1 Fig: Physical address calculation

EX:-
Given Segment Address=3578H, Offset Address =6676H
So Physical address = $[\text{Segment Address} * 10]_H + \text{Offset Address}$
$= [3578 * 10]_H + 6676H$
$= 35780+6676$
$= 3BDF6H$

Types of Segments

There are four types of memory segments defined in 8086:

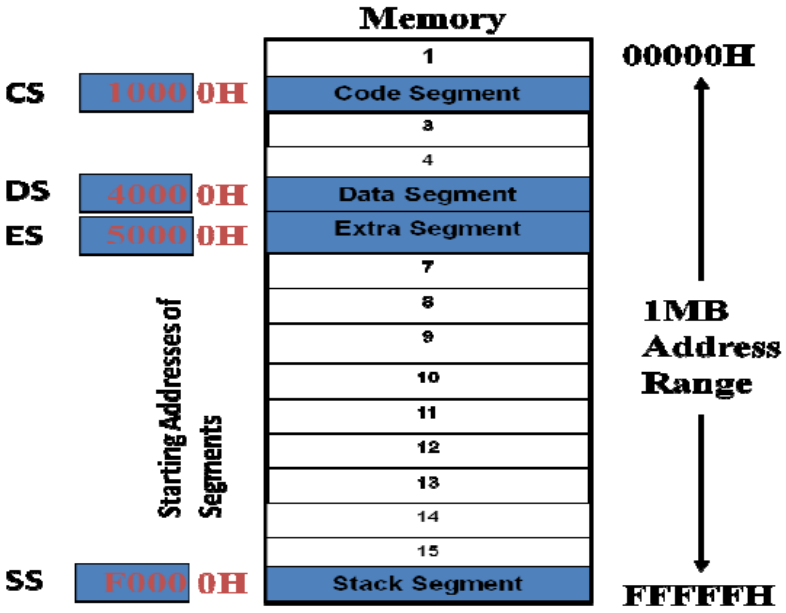
- Code segment(CS)
- Data segment (DS)
- Stack segment(SS)
- Extra segment(ES)

Code segment (CS): This segment is used to store code/program instructions.
Data and Extra segment (DS&ES): This segment is used to store data used in the program.
Stack segment (SS): This segment is used to store the stack contents.

Types of Segments Registers:

To hold the upper 16-bits of the starting address for each of the segments, there are four segment registers:

- **CS (Code Segment register)**
- **DS (Data Segment register)**
- **SS (Stack Segment register)**
- **ES (Extra Segment register)**



Advantage of memory Segmentation:

- Allows the memory capacity to be 1 Mbytes although the actual addresses to be handled are of 16-bit size.
- Allows the placing of code, data and stack portions of the same program in different parts (segments) of memory, for data and code protection.
- Permits a program and/or its data to be put into different areas of memory each time the program is executed. i.e., provision for relocation is done.

PIN Diagram

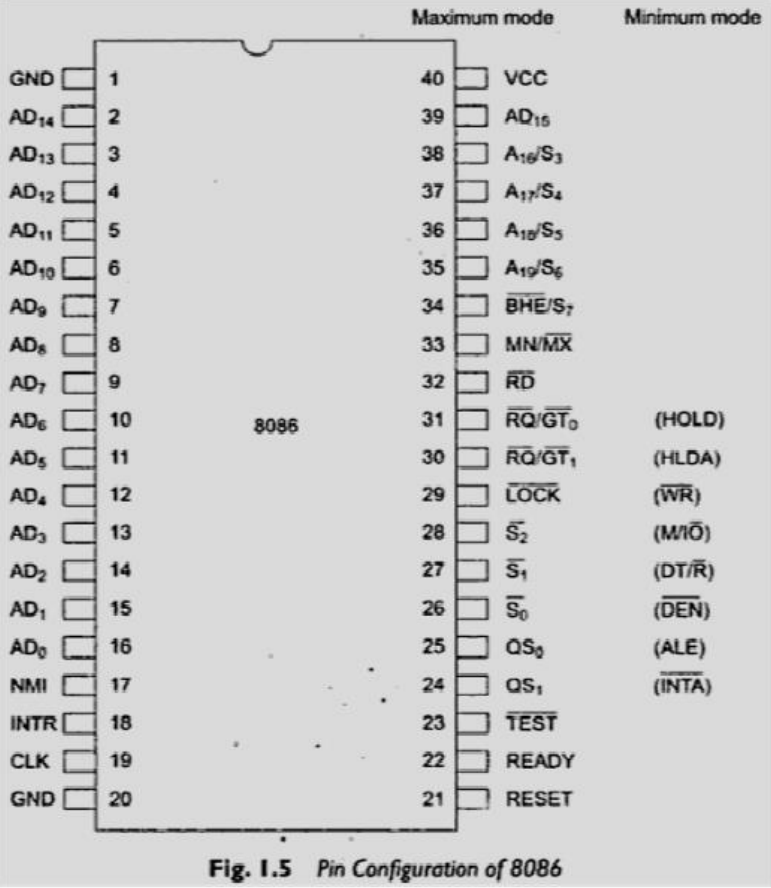


Fig. 1.5 *Pin Configuration of 8086*

The following signal descriptions are common for both modes.

AD15-AD0:

These are the time multiplexed memory I/O address and data lines. Address remains on the lines during T1 state, while the data is available on the data bus during T2, T3, Tw and T4.

A19/S6, A18/S5, A17/S4, A16/S3:

These are the time multiplexed address and status lines. During T1 these are the most significant address lines for memory operations. During I/O operations, these lines are low. During memory or I/O operations, status information is available on those lines for T2, T3, Tw and T4.

- **A16/S3,A17/S4-**

  A16,A17 are multiplexed with segment identifier signals S3 and S4 which combinedly indicate which segment register is presently being used for memory accesses as in below fig..

S4	S3	Indication
0	0	Extra segment(ES)
0	1	Stack segment(SS)
1	0	Code segment(CS)
1	1	Data segment (DS)

- **A18/s5:** A18 is multiplexed with status S5 of the interrupt enable flag bit which is updated at the beginning of each clock cycle.
- **A19/s6:** A18 is multiplexed with status S6.

$\overline{\text{BHE}}$ / S7: (Bus High enable)

- The bus high enable is used to indicate the transfer of data over the higher order (D15-D8) data bus as shown in table.
- It goes low for the data transfer over D15-D8 and is used to derive chip selects of odd address memory bank or peripherals. BHE is low during T1 for read, write and interrupt acknowledge cycles, whenever a byte is to be transferred on higher byte of data bus.
- The status information is available during T2, T3 and T4. The signal is active low.

$\overline{\text{BHE}}$	A0	Indication
0	0	Whole Word
0	1	Upper byte from or to odd address
1	0	Lower byte from or to even address
1	1	None

$\overline{RD}$ (Read):

- This signal on low indicates the peripheral that the processor is performing s memory or I/O read operation. The signal is active low.

READY:

- This is the acknowledgement from the slow device or memory that they have completed the data transfer. The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086. This signal is active high.
- enter into wait states and remain idle : READY = 0
- no effect on the operation of μ : READY = 1

INTR (Interrupt Request):

- This is a level triggered input and hardware interrupt pin.
- If any interrupt request is pending, the processor enters the interrupt acknowledge cycle. This can be internally masked by resulting the interrupt enable flag.

NMI : non-maskable interrupt

- This is a edge triggered input and hardware interrupt pin which causes Type 2 interrupt.

$\overline{TEST}$:

- This input is examined by a 'WAIT' instruction.
- If the TEST pin goes low, execution will continue, else the processor remains in an idle state.

CLK (Clock Input):

- The clock input provides the basic timing for processor operation and bus control activity.

V_{CC} (power supply) : +5.0V, ±10%

RESET:

- μ : reset if RESET is high

GND(Ground) : Two pins labeled GND(0 voltage

MN/$\overline{MX}$:

- The logic level at this pin decides whether the processor is to operate in either minimum or maximum mode.
- if **MN/$\overline{MX}$**= 1; Minimum Mode else Maximum Mode

Pin functions for the minimum mode operation of 8086

1. **$M/\overline{IO}$**
    - This is a status line logically equivalent to $\overline{S2}$ in maximum mode. When it is low, it indicates the CPU is having an I/O operation, and when it is high, it indicates that the CPU is having a memory operation.

2. **$\overline{INTA}$ (Interrupt Acknowledge):**
    - when this signal it goes low, the processor has accepted the interrupt.
3. **ALE (Address Latch Enable):**
    - This output signal indicates the availability of the valid address on the address/data lines, and is connected to latch enable input of latches.
4. **$DT/\overline{R}$ (Data Transmit/Receive):**
    - This output is used to decide the direction of data flow through the transreceivers (bidirectional buffers).
    - When the processor sends out data, this signal is high and when the processor is receiving data, this signal is low.
5. **$\overline{DEN}$ Data Enable:**
    - This signal indicates the availability of valid data over the address/data lines. It is used to enable the transreceivers (bidirectional buffers) to separate the data from the multiplexed address/data signal.
    - It is active from the middle of T2 until the middle of T4.
6. **HOLD, HLDA**- Acknowledge:
    - When the HOLD line goes high, it indicates to the processor that another master is requesting the bus access.
    - The processor, after receiving the HOLD request, issues the hold acknowledge signal on HLDA pin, in the middle of the next clock cycle after completing the current bus cycle.
7. **$\overline{WR}$ (Write):**
    - When it is low the processor perform memory or Io write .

Pin functions for the maximum mode operation of 8086

1. **$\overline{S2}$, $\overline{S1}$, $\overline{S0}$ – Status Lines:**

- These signals are connected to 8288.These are the status lines which reflect the type of operation according to the below table, being carried out by the processor.

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Indication
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
0	0	0	Code access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive State

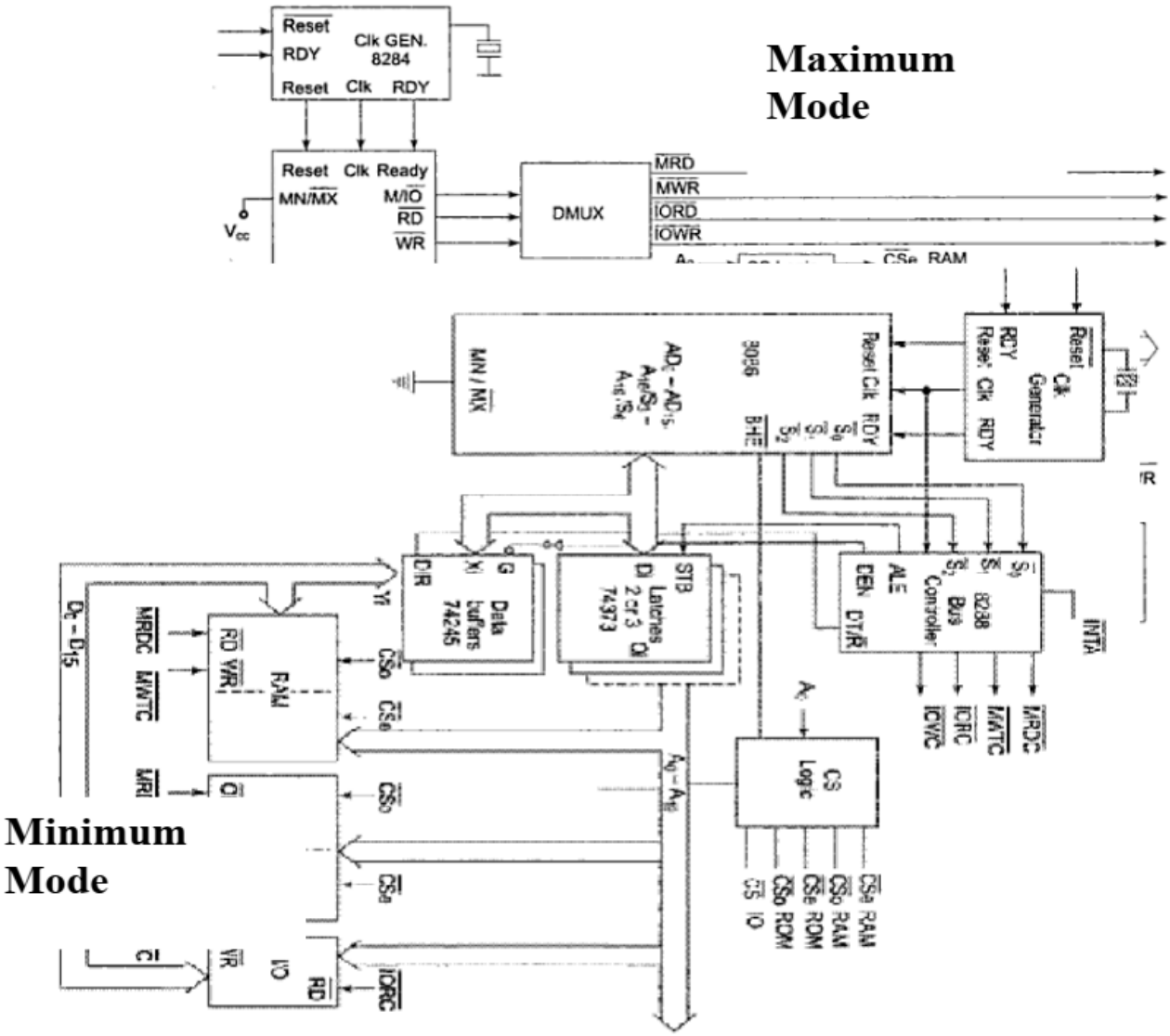
2. **$\overline{LOCK}$** :
   - This output pin indicates that other system bus master will be prevented from gaining the system bus, while the LOCK signal is low.
   - The LOCK signal is activated by the 'LOCK' prefix instruction and remains active until the completion of the next instruction. When the CPU is executing a critical instruction which requires the system bus, the LOCK prefix instruction ensures that other processors connected in the system will not gain the control of the bus.
3. **QS1, QS0 (queue status)**
   - These lines give information about the status of the code-prefetch queue. These are active during the CLK cycle after while the queue operation is performed.

QS1	QS0	Indication
0	0	No operation
0	1	First byte of opcode from the queue
1	0	Empty Queue
1	1	Subsequent byte from queue

4. **$\overline{RQ/GT1}$,$\overline{RQ/GT0}$ (Request/Grant)**

- These pins are used by the other local bus master in maximum mode, to force the processor to release the local bus at the end of the processor current bus cycle.
- Each of the pin is bidirectional with RQ/GT0 having higher priority than RQ/GT1.

In maximum mode of operation signals like **$\overline{WR}$, ALE, $\overline{DEN}$, DT/ $\overline{R}$** etc are not available directly from the processor.
These signals are available from the controller 8288.



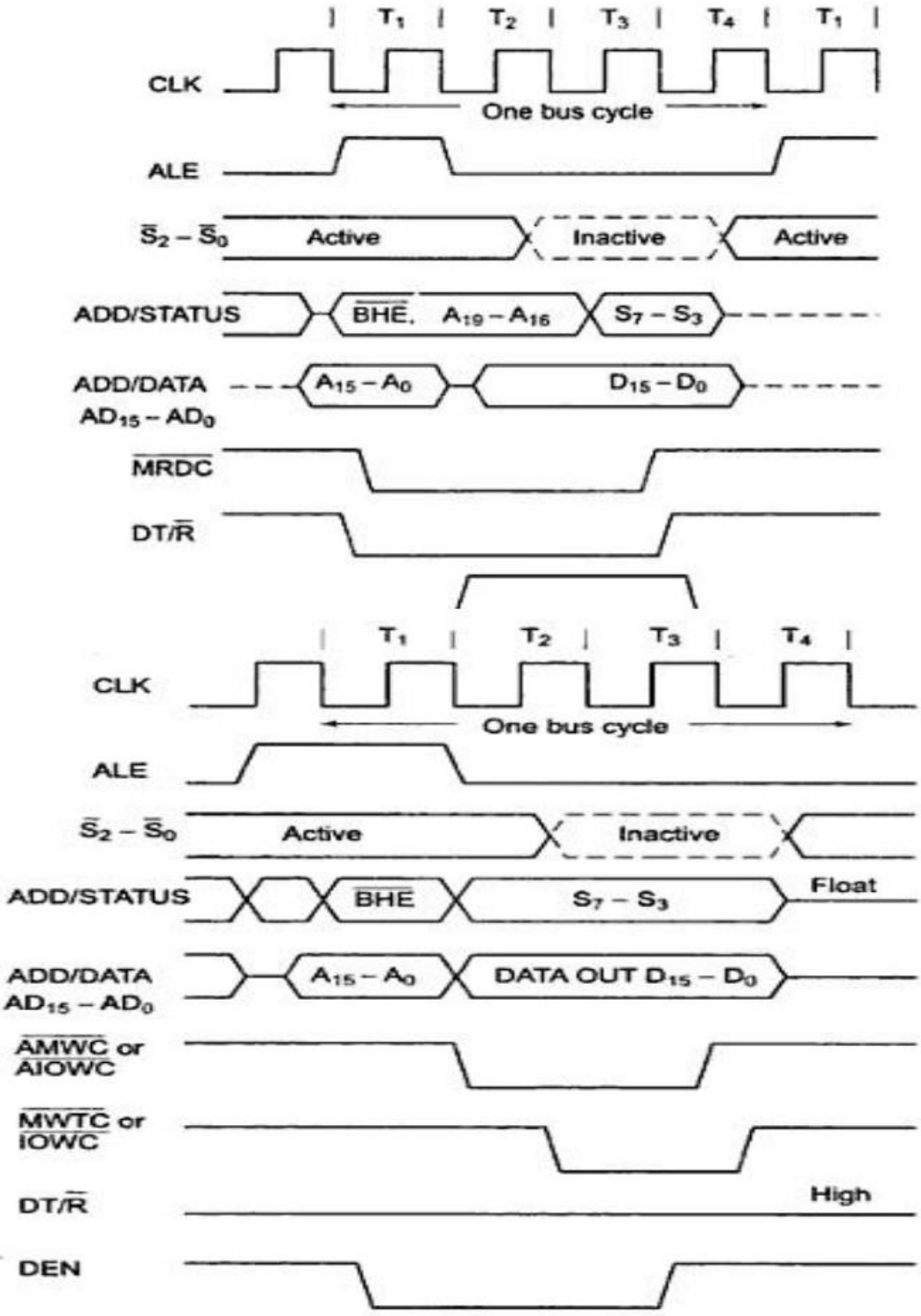


Fig. 1.16(b) *Memory Write Timing in Maximum Mode*

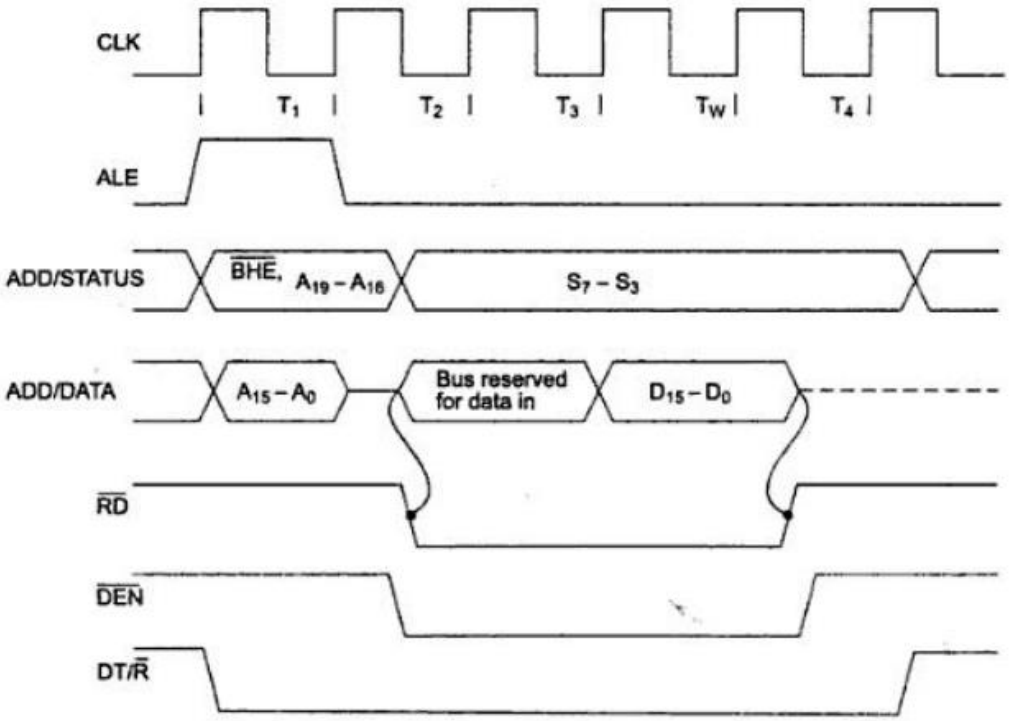


Fig.1.14(a) *Read Cycle Timing Diagram for Minimum Mode*

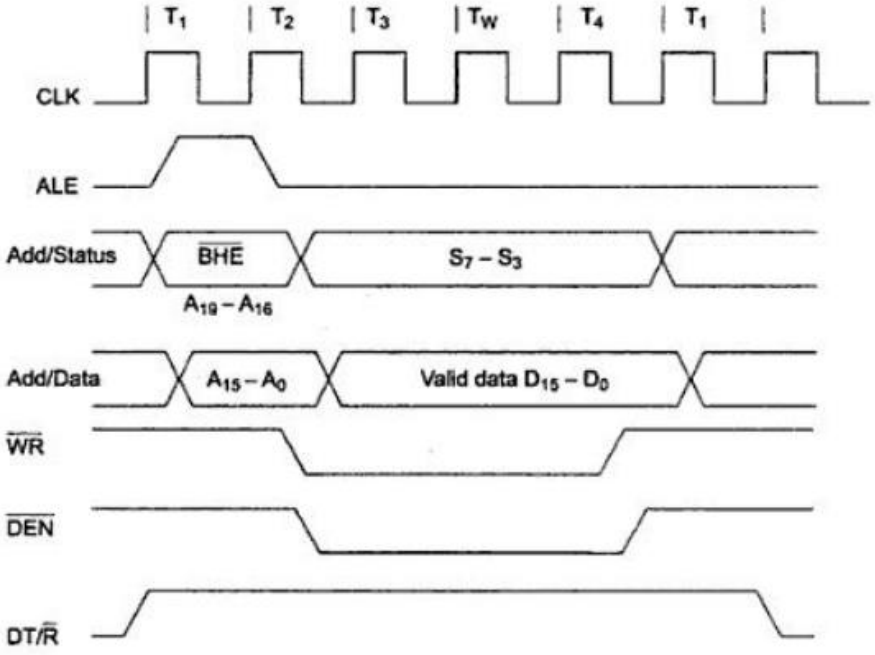


Fig. 1.14(b) *Write Cycle Timing Diagram for Minimum Mode Operation*