Microprocessors and Microcontrollers

Text Books

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.

2. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, Pearson Education.

3. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education

MODULE 1

Syllabus:

8085 microprocessor (-Basic Architecture only). 8086 microprocessor – Architecture and

signals, Physical Memory organization, Minimum and maximum mode of 8086 system

and timings. Comparison of 8086and 8088.Machine language instruction format

Microprocessor

- A **microprocessor** is a computer processor that incorporates the functions of a computer’s central processing unit on a single integrated circuit.
- Wherever programming is involved there is a microprocessor present.
- Eg: Remote Control, Traffic light, Computers etc.

Evolution of Microprocessors

- A microprocessor with a 4-bit ALU is referred to as a 4-bit microprocessor.

- Intel 4004 is the first commercially available 4-bit microprocessor introduced in 1971 and Intel 8008 is the first 8-bit microprocessor introduced in 1972.
  - These microprocessors could not survive as general purpose microprocessors due to the limitations in their design and performance.

- Intel 8080 an 8-bit microprocessor introduced in 1974 is considered as the first major stepping stone towards development of advanced processors.

Evolution of Microprocessors cont..

- The microprocessor 8085 followed 8080 with few added features in its architecture. This resulted in a functional complete microprocessor.
  - However the limitations where low speed, low memory addressing capability, limited number of general purpose registers and less powerful instruction set.

- Intel’s 8086 was the first 16-bit processor launched in 1978. It has a more powerful instruction set along with architectural developments.

- 8088 is the next microprocessor that has all the programming facilities as that of 8086 along with some hardware features and external interface like 8085 so that all existing circuits built around 8085 can work like before.

Evolution of Microprocessors cont..

- 80186 was the microprocessor with integrated peripherals.

- 80286 is the first member of the family of advanced microprocessors with memory management and protection abilities(1982)

- 80386 was the first 32 bit processor which tried to overcome the limitations of 80286.

- The 32-bit 80486 from Intel is the first processor with an inbuilt floating point.

- Next Pentium was introduced. It is superscalar, superpipelined architecture.(Pentium, Pentium II, Pentium III, Pentium IV)

8085 architecture

- 8085 is an 8-bit microprocessor designed by Intel in 1977
- It has the following configuration –
- 8-bit data bus
- 16-bit address bus, which can address upto 64KB
- A 16-bit program counter
- A 16-bit stack pointer
- Six 8-bit registers arranged in pairs: BC, DE, HL
- Requires +5V supply to operate at 3.2 MHZ single phase clock

8085 Microprocessor – Functional Units

- 8085 consists of the following functional units –

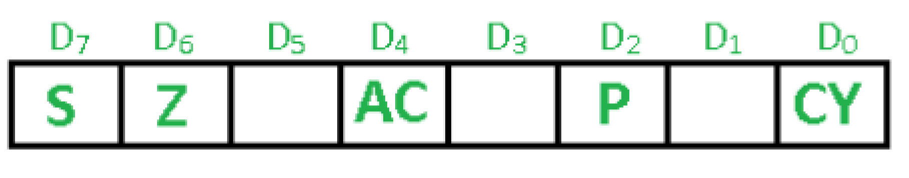
Accumulator

- It is an 8-bit register used to perform arithmetic, logical, I/O & LOAD/STORE operations. It is connected to internal data bus & ALU.

Arithmetic and logic unit

- As the name suggests, it performs arithmetic and logical operations like Addition, Subtraction, AND, OR, etc. on 8-bit data.

- The **Flag register** is a Special Purpose Register. Depending upon the value of result after any arithmetic and logical operation the flag bits become set (1) or reset (0). In 8085 microprocessor, flag register consists of 8 bits and only 5 of the
- m are useful.



D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		CY

General purpose register

- There are 6 general purpose registers in 8085 processor, i.e. B, C, D, E, H & L.
- Each register can hold 8-bit data.

- These registers can work in pair to hold 16-bit data and their pairing combination is like B-C, D-E & H-L.

Program counter

It is a 16-bit register used to store the memory address location of the next instruction to be executed.

Microprocessor increments the program whenever an instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

Instruction register and decoder

It is an 8-bit register. When an instruction is fetched from memory then it is stored in the Instruction register.

Instruction decoder decodes the information present in the Instruction register.

Timing and control unit

It provides timing and control signal to the microprocessor to perform operations. Following are the timing and control signals, which control external and internal circuits –

- Control Signals: READY, RD’, WR’, ALE
- Status Signals: S0, S1, IO/M’
- DMA Signals: HOLD, HLDA
- RESET Signals: RESET IN, RESET OUT

Interrupt control

As the name suggests it controls the interrupts during a process.

When a microprocessor is executing a main program and whenever an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program

There are 5 interrupt signals in 8085 microprocessor: INTR, RST 7.5, RST 6.5, RST 5.5, TRAP.

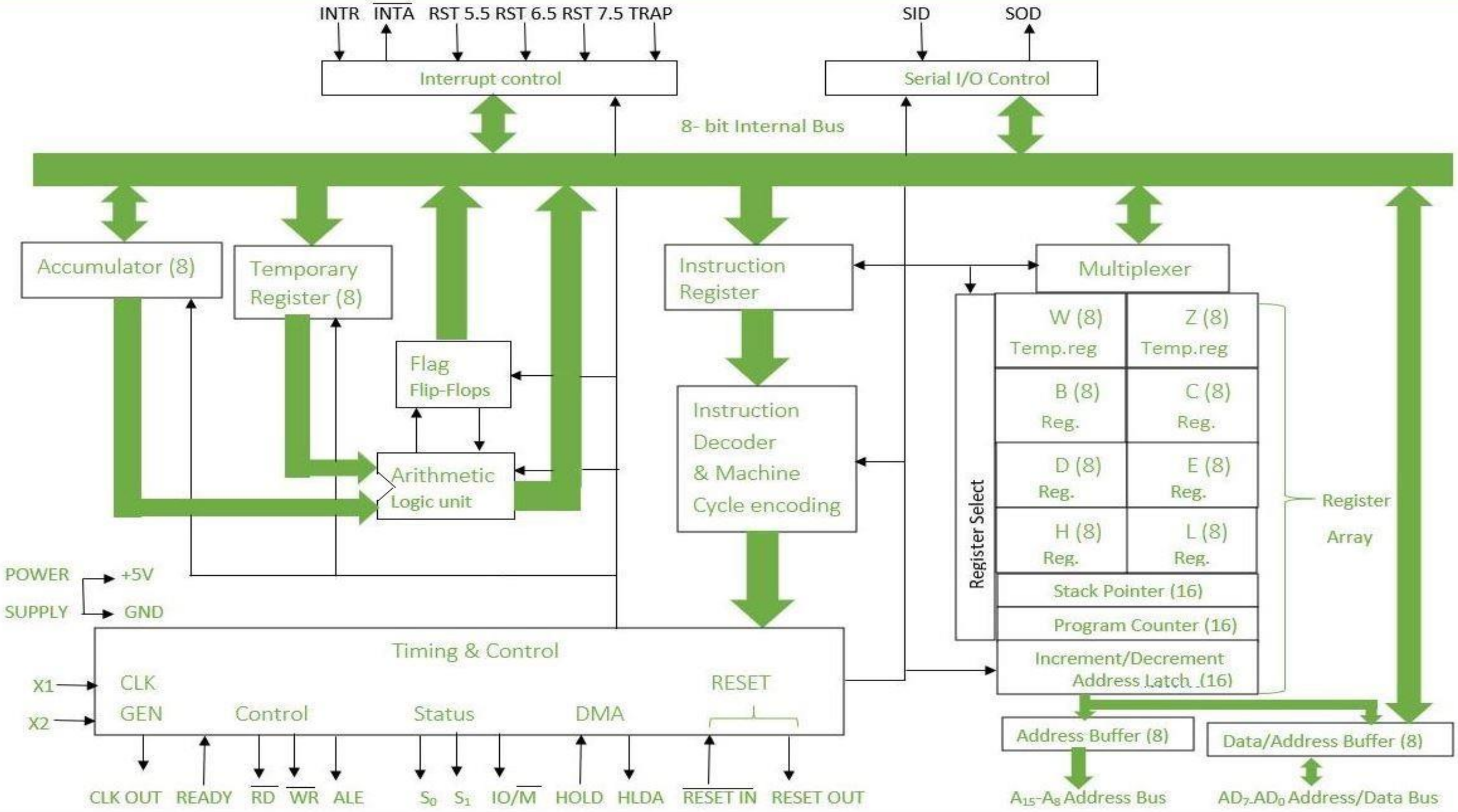
Serial Input/output control

It controls the serial data communication by using these two instructions: SID (Serial input data) and SOD (Serial output data).

- **Address buffer and address-data buffer**
- The content stored in the stack pointer and program counter is loaded into the address buffer and address-data buffer to communicate with the CPU.
- The memory and I/O chips are connected to these buses; the CPU can exchange the desired data with the memory and I/O chips.

- **Address bus and data bus**
- Data bus carries the data to be stored. It is bidirectional, whereas address bus carries the location to where it should be stored and it is unidirectional.
- It is used to transfer the data & Address I/O devices.-lower order buses are multiplexed

8085 ARCHITECTURE



8086 Microprocessor Architecture

- 8086 processor supports a 16-bit ALU, a set of 16 bit registers and provides segmented memory addressing capability.

- It has a rich instruction set, powerful interrupt structure, fetched instruction queue for overlapped fetching and execution etc.

- The 8086 architecture is divided into two independent functional parts,
  - Bus Interface Unit (BIU)
  - Execution Unit (EU).

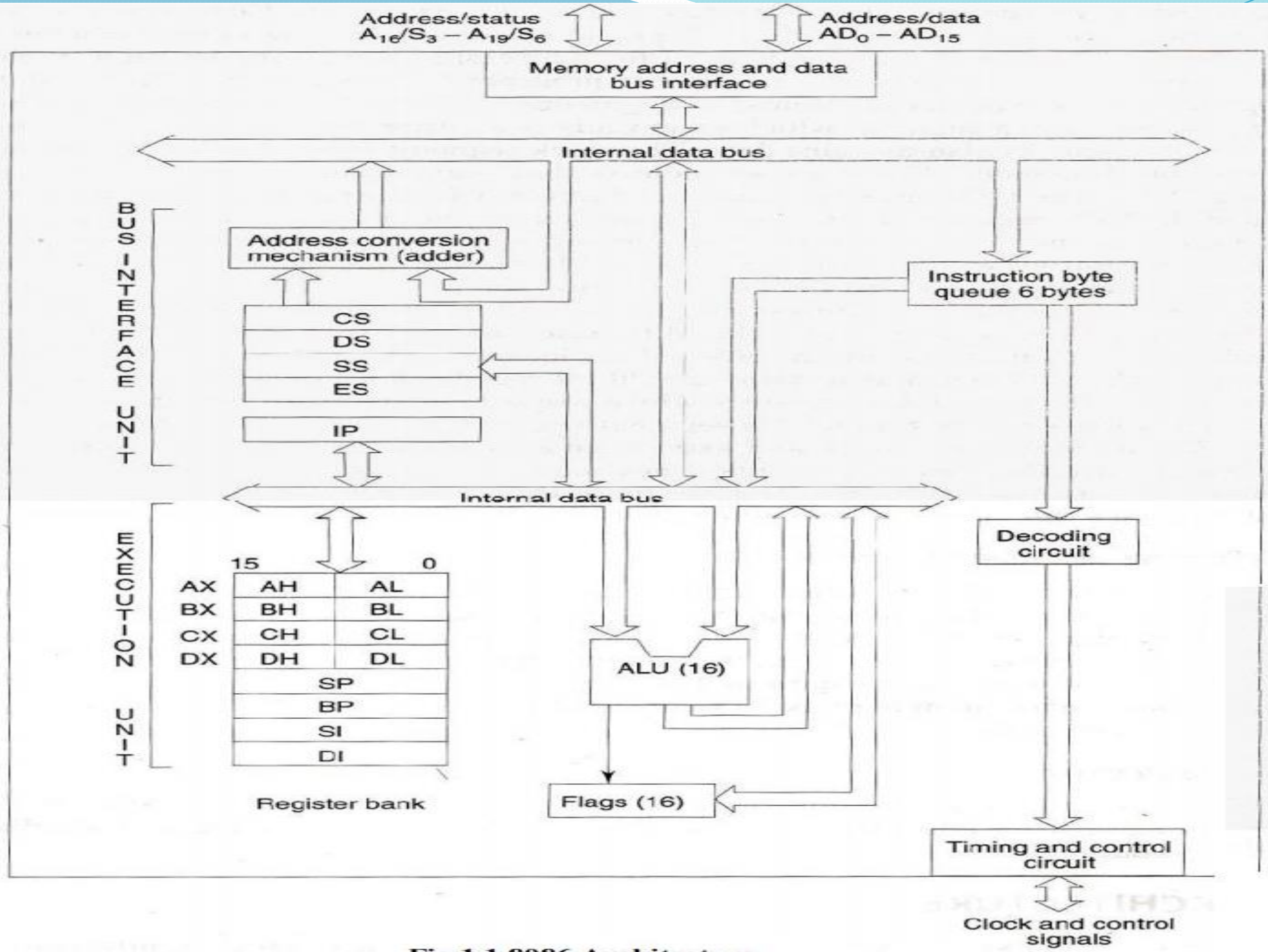


Fig.1.1 8086 Architecture

A) Bus Interface Unit (BIU)

- The Bus Interface Unit (BIU) manages the data, address and control buses

- The BIU functions in such a way that it:
  - Fetches the sequenced instruction from the memory,
  - Finds the physical address of that location in the memory where the instruction is stored and
  - Manages the 6-byte pre-fetch queue where the pipelined instructions are stored.

- 8086 addresses a segmented memory. There are four segment registers.

Code Segment Register(CS)	Data Segment Register(DS)
Extra Segment Register(ES)	Stack Segment Register(SS)

- Code Segment Register(CS) contains the starting address of code segment of the memory where the executable program is stored.

- Data Segment Register(DS) points to the data segment of the memory where the data is resided.

- Extra Segment Register(ES) refers to another data segment of memory which stores extra data.

- Stack Segment Register(SS) is used for addressing stack segment of memory which is used to store stack data.

- Physical address should be calculated for addressing any location in the memory.

- Physical address is calculated from two parts: Segment address and offset address

- Segment Registers contain 16-bit segment base addresses related to different segments.

- Instruction Pointer Register(IP) contains the 16-bit offset which indicates the distance of the required memory location from the base address.

- The advantage of this scheme of calculating physical address is that the processor does not require a 20-bit register to store physical address. Instead it just need to maintain two 16-bit registers which are in capacity of the machine.

- Thus physical address(20 bits) is generated using segment and offset register which are each of 16-bit length.

- For generating physical address, the segment address is shifted left bitwise 4 times and to this result the offset address is added to produce 20 bit physical address.

- Shifting a number 4 times is equivalent to multiplying by 10H.

  Thus PA=Seg*10H+Offset

- Example:

If the segment address is 1005Hand the offset is 5555H, then the physical address can be calculated as below:

Segment address (bitwise) -> 0001 0000 0000 0101

After shifting by 4 bits -> 0001 0000 0000 0101 0000 -----(1)

Offset address (bitwise) -> 0101 0101 0101 0101 -----(2)

(1) + (2) = 0001 0101 0101 1010 0101

1 5 5 A 5

<u>Going by second method</u>

i.e. Physical Address=Segment Address*10H+Offset

Segment address*10 = 10050 --------(1)

Offset address = 5555 --------(2)

Physical Address (1)+(2) = 155A5

▪The address conversion mechanism(adder) in the bus interface unit performs this procedure to obtain physical address.

- **6-byte pre-fetch queue**: This queue is used in 8086 in order to perform pipelining.
  - i.e. at the time the previously decoded instruction is executed in the execution unit(EU), the BIU fetches the sequential upcoming instructions and stores it in this queue. Thus BIU along with EU forms a pipeline.

- The size of this queue is 6-byte. This means at maximum a 6-byte instruction can be stored in this queue.

- The queue exhibits **FIFO** behaviour., first in first out.

B) Execution Unit(EU)

- The Execution Unit (EU) performs the execution of the instructions that are being fetched from the desired memory location.

- EU contains the registers AX, BX, CX and DX which are known as general purpose 16 bit registers. Usually the letters 'L' and 'H' specify lower and higher bits of a particular register and 'X' denotes the complete 16 bit register.
  - AX : used as 16 bit accumulator. Lower 8 bits is in AL and higher 8 bits in AH
  - BX: used as an offset storage for forming physical addresses in case of certain addressing modes.
  - CX: used as default counter in case of string and loop instructions.
  - DX: is a general purpose register which may be used as an implicit operand or destination

- SP and BP are pointers which usually contain the offset within the stack segments

- The register SI is used to store the offset of source data in data segment.

- DI is used to store the offset of destination data in extra segment. Index registers are useful for string manipulations.

- The decoding circuit decodes the opcode bytes issued from the instruction byte queue.

- The timing and control unit in 8086 microprocessor produces control signal after decoding the opcode to inform the general purpose register to release the value stored in it. And it also signals the ALU to perform the desired operation.

- **ALU**: The 16-bit arithmetic and logic unit carries out the arithmetic and logical tasks according to the signal generated by the Control Unit. The result of the operation is stored in the desired register.

- **Flag**: The 16-bit flag register holds the status of the result generated by the ALU. It has several flags that show the different conditions of the result.

- Finally, EU may pass the results to the BIU for storing them in memory.

Memory Segmentation

- **Segmentation** is the process in which the main memory of the computer is logically divided into different segments and each segment has its own base address.

- It is basically used to enhance the speed of execution of the computer system, so that the processor is able to fetch and execute the data from the memory easily and fast.

- The memory in 8086 is organised as segmented memory i.e. the complete physically available memory is divided into a number of segments.

- Each segment has maximum 64K bytes in size and minimum 16 bytes size. It is addressed by one of the segment registers.

- The content of segment register points to the starting location of a particular segment.

- To address a specific memory location within a segment, an offset address is needed.

- 8086 processor is able to address 1MBytes of physical memory. The complete 1 MBytes memory can be divided into 16 segments, each of 64Kbytes size.

- There are two types of segmentation
  - **Overlapping and Non-Overlapped Segment**

- **Overlapping Segment –** A segment starts at a particular address and its maximum size can go up to 64Kbytes. But if another segment starts along with this 64Kbytes location of the first segment, then the two are said to be *Overlapping Segment*.
  - The area of memory from the start of second segment to the possible end of the first segment is called an overlapped segment area.
  - In the overlapping area locations physical address = CS1+IP1 = CS2+IP2.

- **Non-Overlapped Segment –** A segment starts at a particular address and its maximum size can go up to 64Kbytes. But if another segment starts before this 64Kbytes location of the first segment, then the two segments are said to be *Non-Overlapped Segment*.

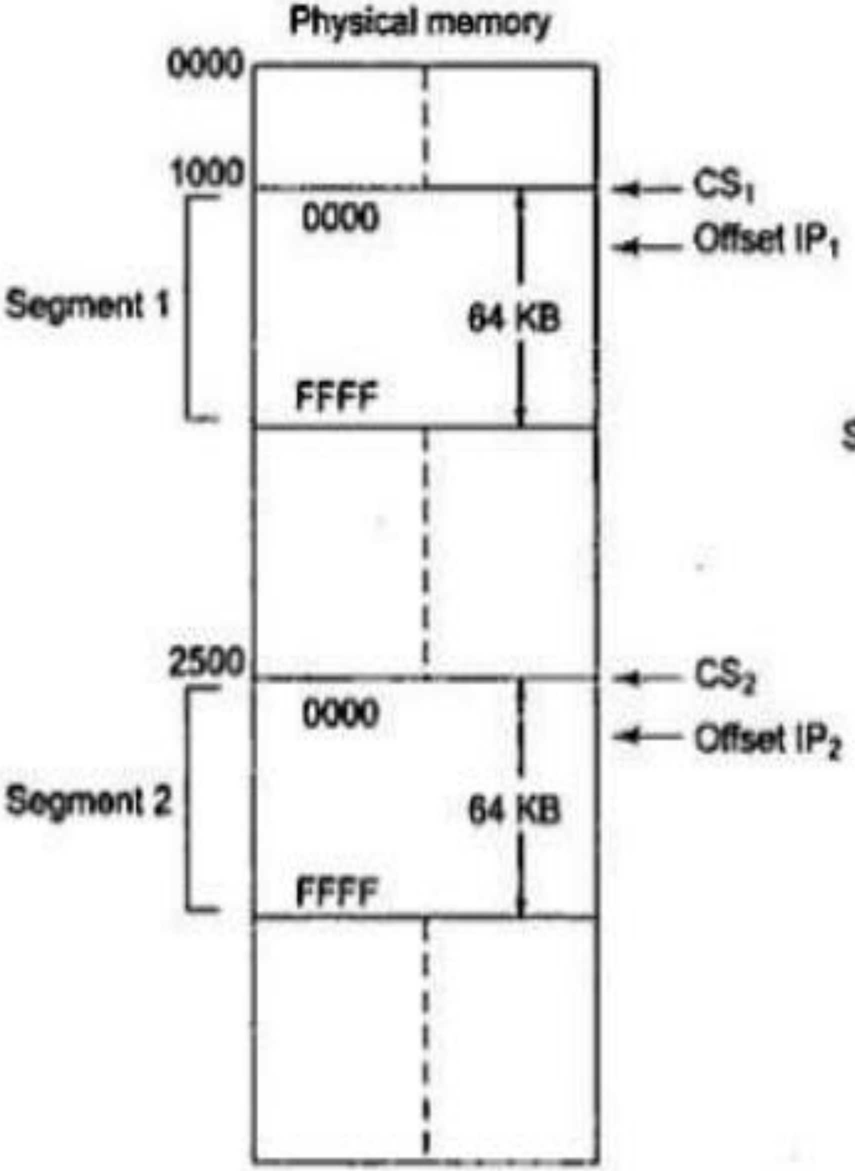


Fig. 1.3(a) *Non-overlapping Segments*

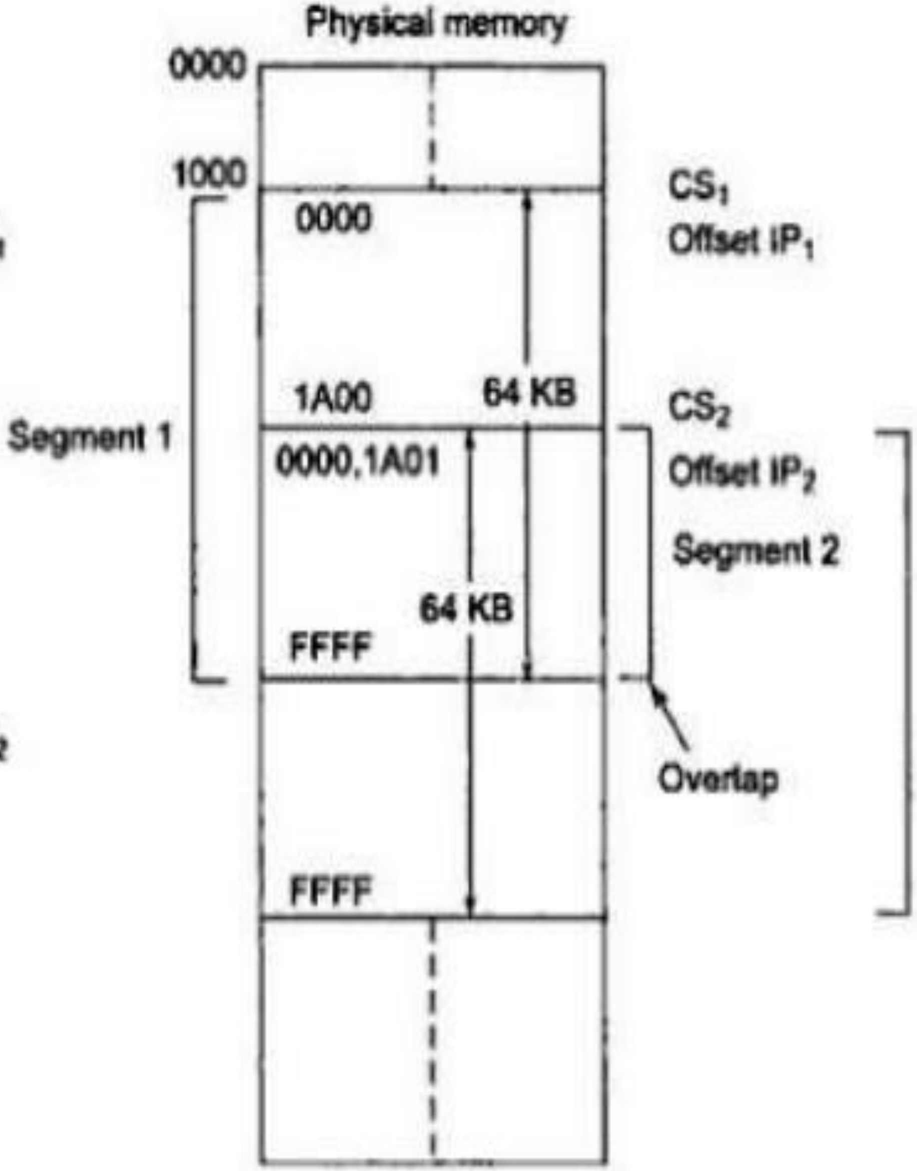


Fig. 1.3(b) *Overlapping Segments*

- The main advantages of the segmented memory scheme are as follows:

1. Allows the memory capacity to be 1MB although the actual addresses to be handled are of 16-bit size.

2. Allows the placing of code, data and stack portions of the same program in different parts (segments) of memory, for data and code protection.

3. 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.

Flag Register

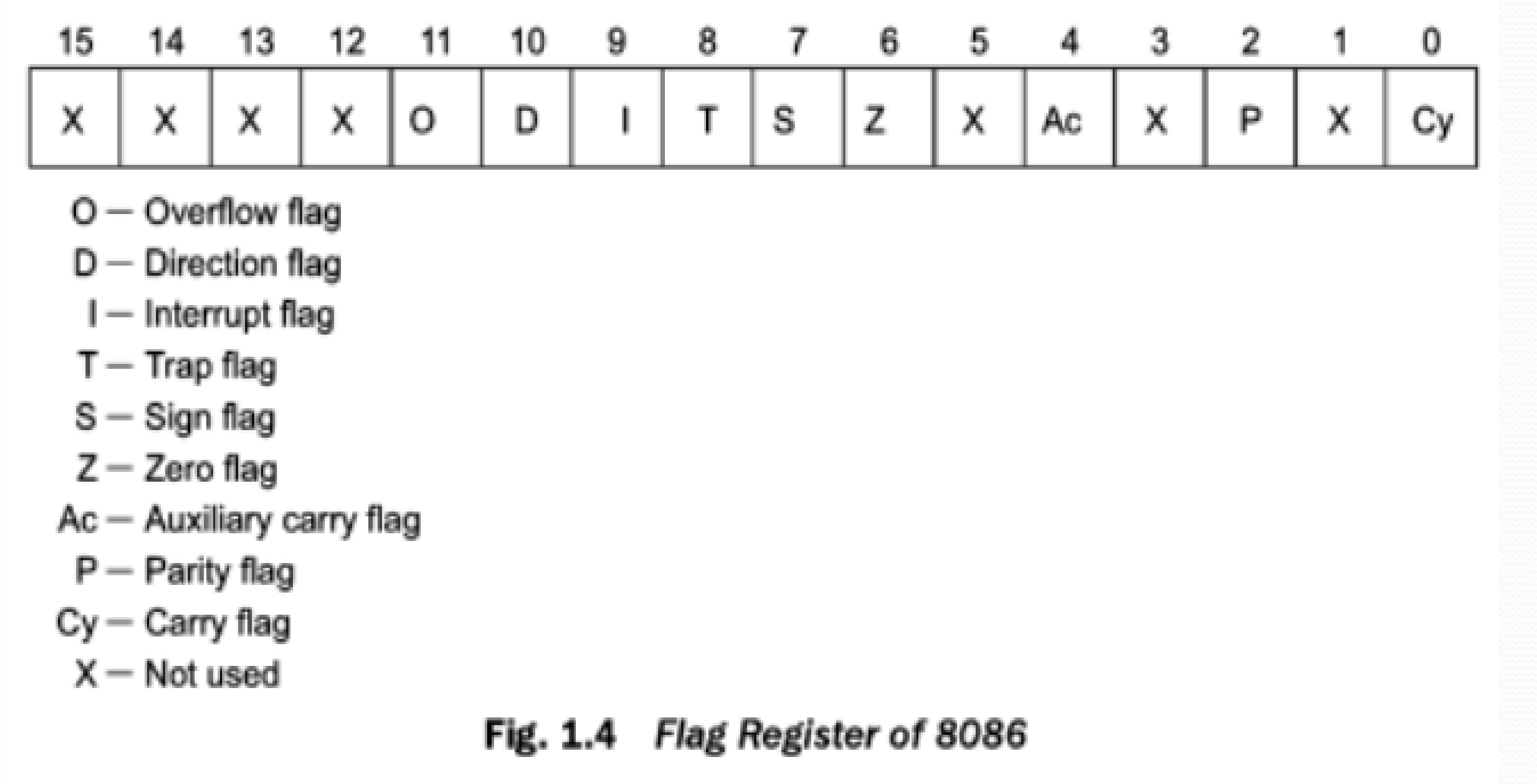
- The 8086 flag register contents indicate the results of computation in the ALU.

- A 16 bit flag register is used in 8086. It is divided into two parts
  - Condition code or status flags
  - Machine control flags

- The condition code flag register is the lower byte of the 16-bit flag register along with the overflow flag. This register reflects the results of the operations performed by ALU.

- The control flag register is the higher byte of the flag register. It contains three flags namely Direction flag (D), Interrupt flag (I) and Trap flag (T). It is used by programmers to control different operations.

Flag register configuration



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	O	D	I	T	S	Z	X	Ac	X	P	X	Cy

O — Overflow flag
D — Direction flag
I — Interrupt flag
T — Trap flag
S — Sign flag
Z — Zero flag
Ac — Auxiliary carry flag
P — Parity flag
Cy — Carry flag
X — Not used

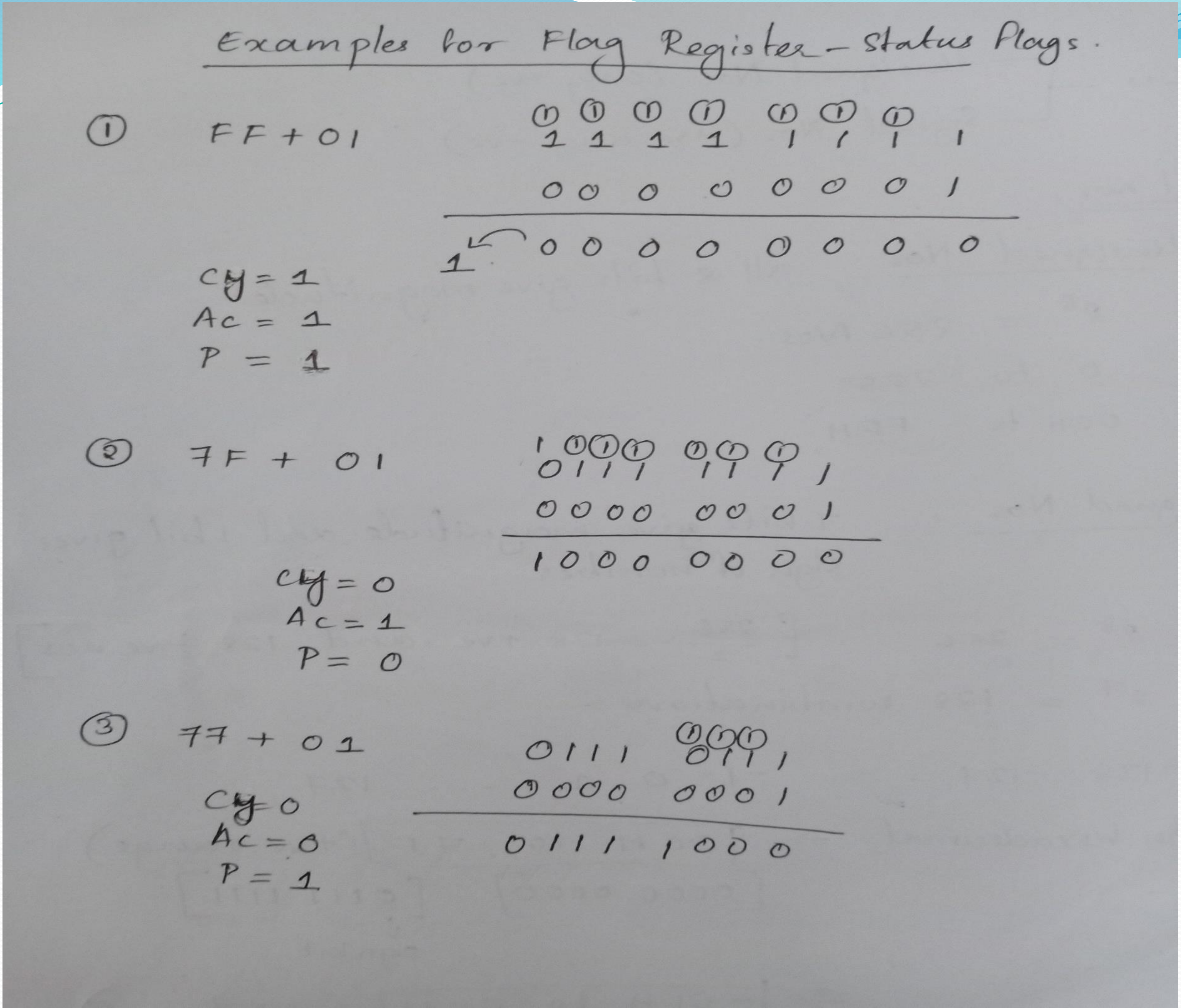
Fig. 1.4 *Flag Register of 8086*

- **Carry Flag(Cy)–** This flag is set when there is a carry out of MSB in the case of addition or borrow in the case of subtraction.

- **Parity Flag(P)** – This flag is set to 1 if the lower order 8 bits of the result contains even number of 1’s otherwise it is reset to 0.

- **Auxiliary carry flag(**Ac) - This flag is set if there is a carry from the lowest nibble (i.e. carry from third bit of lowest position) during addition or borrow for the lowest nibble during subtraction.

- **Zero Flag(Z)** – This flag is set if the result of the computation or comparison performed by the previous instruction is zero.



Examples for Flag Register - Status Flags.

① FF + 01

```
    1 1 1 1  1 1 1
    1 1 1 1  1 1 1 1
    0 0 0 0  0 0 0 1
  -----------------
  1 0 0 0 0  0 0 0 0
```

Cy = 1
Ac = 1
P = 1

② 7F + 01

```
  1 1 1 1  1 1 1
  0 1 1 1  1 1 1 1
  0 0 0 0  0 0 0 1
  ----------------
  1 0 0 0  0 0 0 0
```

Cy = 0
Ac = 1
P = 0

③ 77 + 01

```
            1 1 1
  0 1 1 1  0 1 1 1
  0 0 0 0  0 0 0 1
  ----------------
  0 1 1 1  1 0 0 0
```

Cy = 0
Ac = 0
P = 1

- **Sign Flag(S)** – This flag contains the value of MSB of result. This flag is set to 1 when the result of any computation is negative for signed numbers.

- **Overflow flag(O)** -This flag is set if an overflow occur i.e. if the result of signed operation is large enough to accommodate in a destination register.

  - Example: In case of addition of two signed numbers, if the result overflows onto the sign bit i.e the result is of more than 7-bits in size in case of 8-bits signed operations and more than 15-bits in size in case of 16-bits signed operations, then overflow flag will be set.

- **Trap Flag(T)** – If this flag is set, a trap interrupt is generated after execution of each instruction. The processor executes the current instruction and the control is transferred to the trap interrupt service routine.
  - So when trap flag is set, program can be run in single step mode. Thus it allows user to execute one instruction of a program at a time for debugging.

- **Interrupt Flag(I)** – If this flag is set the mask-able interrupts are recognized by the CPU.

- **Direction Flag(D)** – This is used by string manipulation instruction.
  - If this flag bit is 0, the string is processed beginning from the lowest address to the highest address.
  - If this flag bit is 1, the string is processed beginning from the highest address to the lowest address.

TUTORIAL 1

I) What is the importance of 6-Byte Prefetch Queue in 8086 Architecture

II) Find the value of Cy, P, Ac, Z, S, O flags for the following questions.

a) 42H+23H

b) 37H+29H

c) 42H+43H

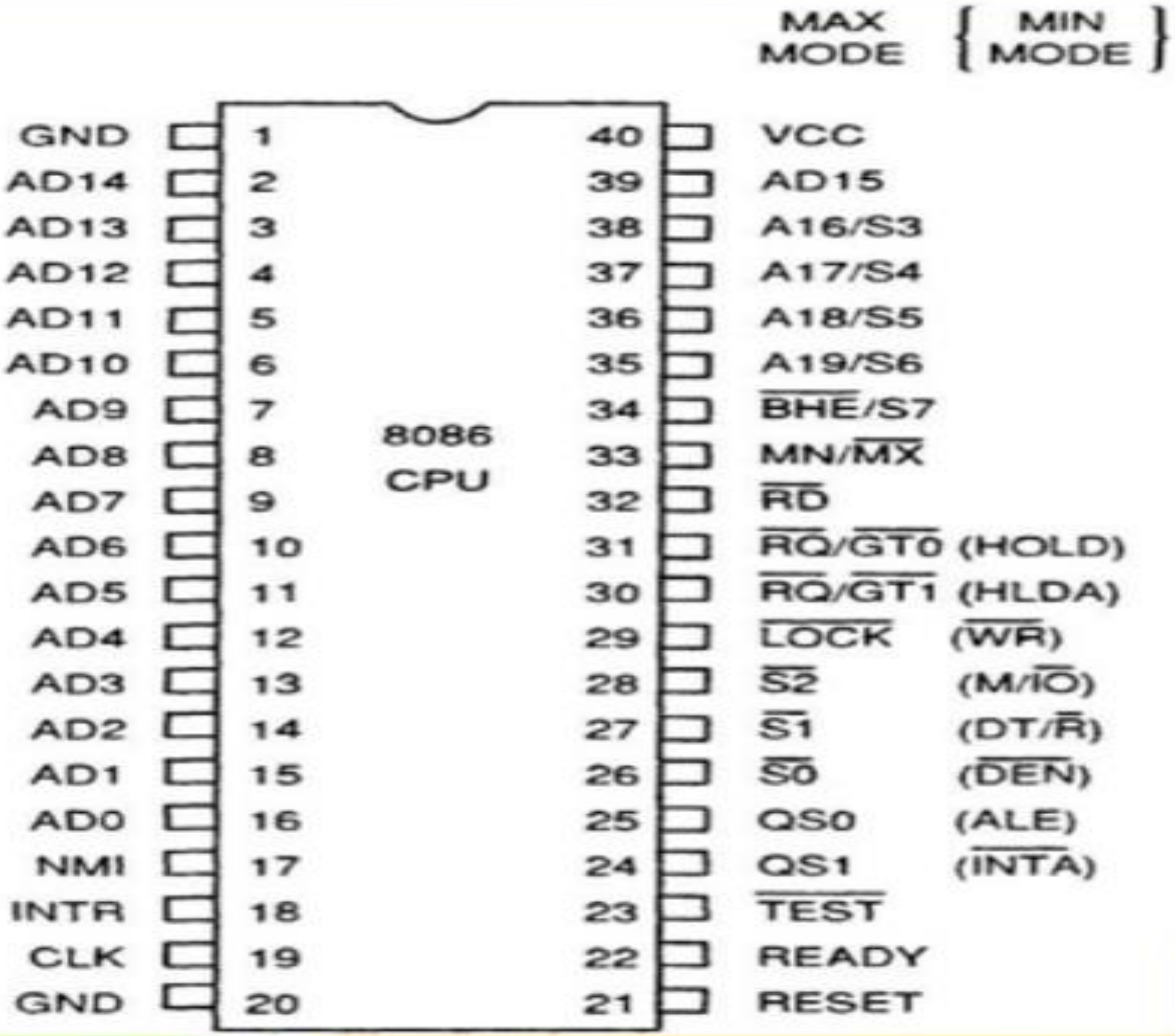
Signal Description of 8086

- The 8086 is a 16-bit microprocessor. This microprocessor operates in single processor or multiprocessor configurations to achieve high performance.

- It is Packaged in a 40 pin CERDIP (CERamic Dual In-line Package)

- Some of the pins serve a particular function in minimum mode (single processor mode) and others function in maximum mode (multiprocessor mode).

Pin Configuration of 8086

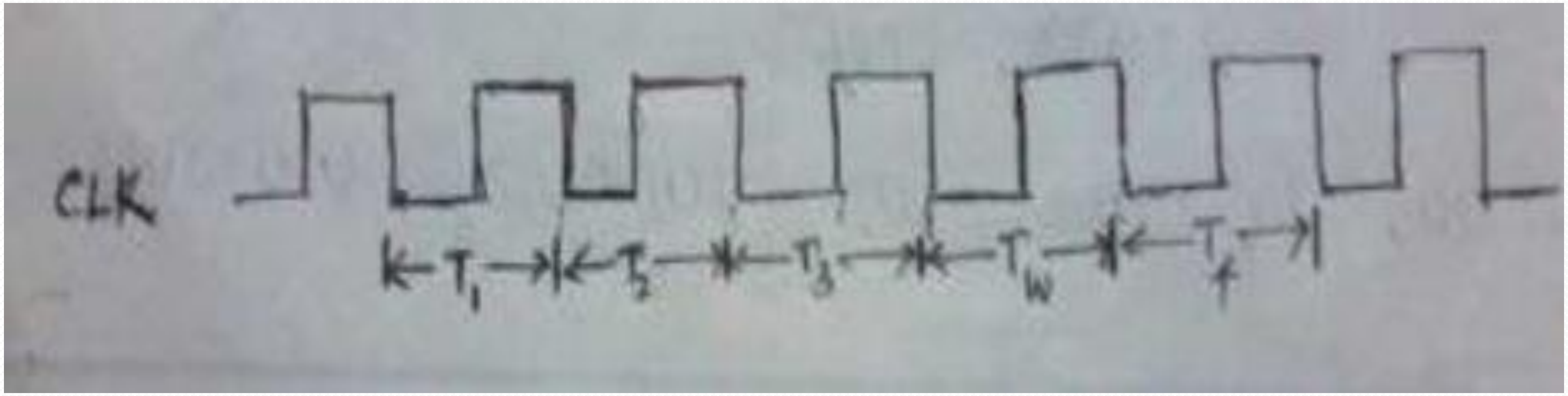


- 8086 signals can be categorized in three groups
  - Signals having common functions in minimum as well as maximum mode.
  - Signals having special functions for minimum mode.
  - Signals having special functions for maximum mode.

Pins common for both minimum and maximum nodes:

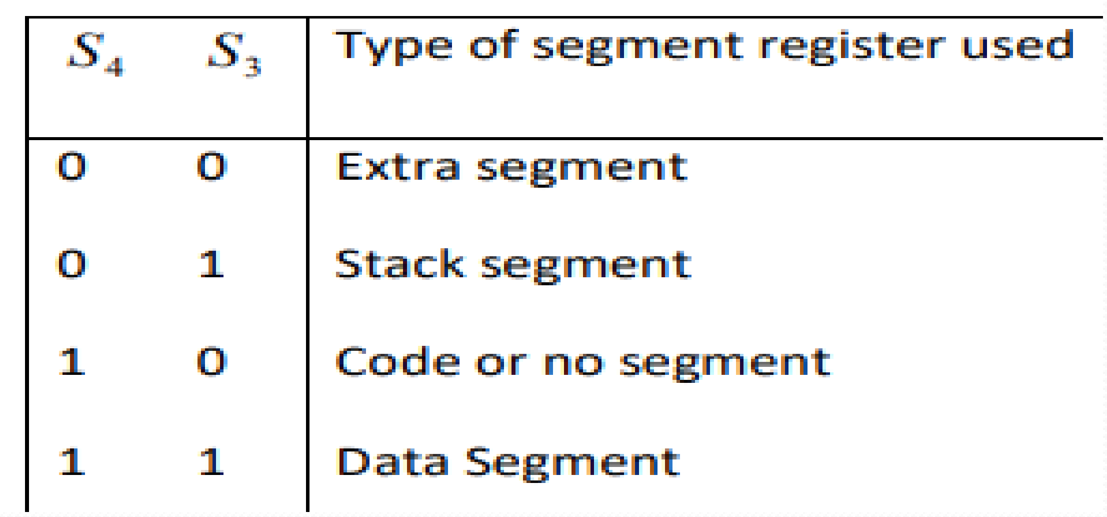
AD15 – AD0:-

- These are the time multiplexed memory I/O address and data lines
- T1,T2,T3,T4 and Tw are the clock states of a machine cycle.
- Tw is a wait state.
- Address remains on the lines during T1 state, while data is available on the data bus during T2 ,T3 , Tw and T4.



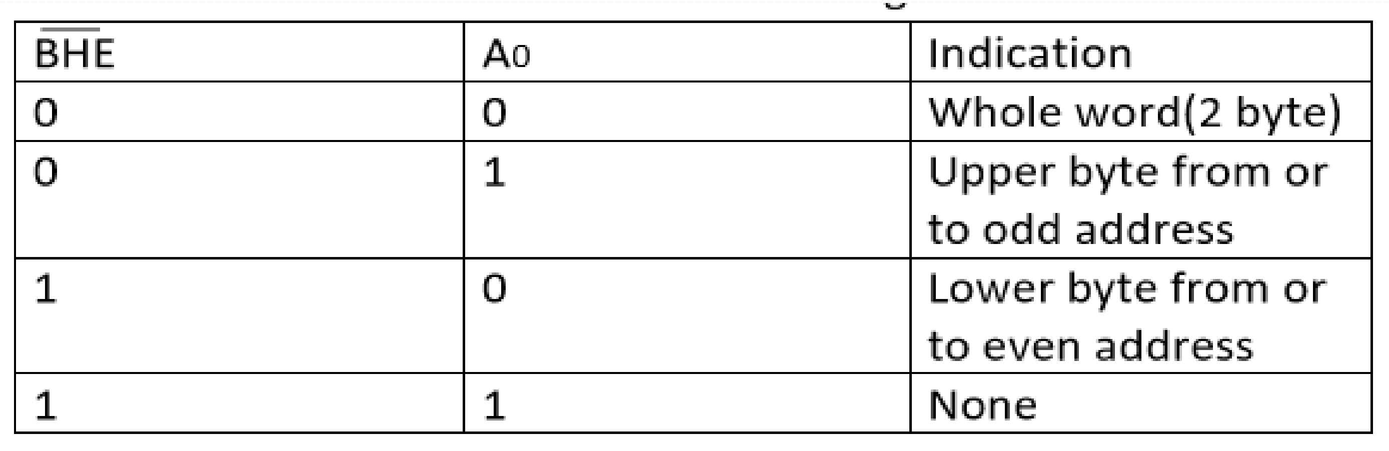
A19/S6 ,A18/S5, A17/S4, A16/S3 : These are time multiplexed address and status lines.

- During T1 , these are the most significant address lines for memory operations.
- During memory or I/O operations, status information is available on those lines for T2,T3,Tw and T4.
- S6 is always low(logic 0).
- S5 denotes the status of interrupt enable flag bit and is updated at the beginning of each clock cycle.
- S4 and S3 together indicate which segment register is presently being used for memory accesses.



S_4	S_3	Type of segment register used
0	0	Extra segment
0	1	Stack segment
1	0	Code or no segment
1	1	Data Segment

- **$\overline{\text{BHE}}$ / S7:- Bus High Enable / Status** :
- It indicates the transfer of data over the higher order (D15 - D8) data bus
- It goes low for the data transfers over D15-D8 and is used to derive chip select of odd address memory bank or peripherals.
- $\overline{\text{BHE}}$ is low during T1 for read ,write and interrupt acknowledge cycles, whenever a byte is to be transferred on the higher byte of the data bus.
- The status information is available during T2,T3,and T4.



$\overline{\text{BHE}}$	A_0	Indication
0	0	Whole word(2 byte)
0	1	Upper byte from or to odd address
1	0	Lower byte from or to even address
1	1	None

RD – Read :-

- Read signal, when low, indicates the peripheral that the processor is performing a memory or I/O read operation.

READY:-

- This is the acknowledgement from the slow devices or memory.
- When high, it indicates that the device is ready to transfer data, else the microprocessor is in wait state.

INTR:- Interrupt Request

- This is a level triggered input.
- This is sampled during last clock cycle of each instruction to determine the availability of the request.
- If any interrupt request is pending, the processor enters the interrupt acknowledge cycle

TEST:-

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

NMI:-Non Maskable Interrupt

- This is an edge triggered input which causes Type-2 interrupt.
- It is not maskable internally by software.
- A transition from low to high initiates the interrupt response at the end of each instruction

- **RESET:-**
- This input causes the processor to terminate the current activity and start execution from FFFFOH.
- The signal is active high for atleast 4 clock cycles.
- It restarts execution when the RESET returns low.

CLK- Clock input:-

- Provide the basic timing for processor operation and bus control activity.

VCC:-

- +5V power supply for the operation of the internal circuit.

GND:-

- ground for the internal circuit.

MN/$\overline{\text{MX}}$:-

- The logic level at this pin decides whether the processor is to operate in either minimum or maximum mode.
- When the pin is 1: minimum mode and 0: maximum mode

Pins for the minimum mode operation

M/I/O:- Memory I/O

- It indicates that the microprocessor address bus contains either a memory address or an I/O port address .
- Low indicates that CPU is having an I/O operation.
- High indicates that CPU is having a memory operation.

INTA:- Interrupt Acknowledge

- The interrupt acknowledge signal is a response to the INTR input signal.
- Low means processor has accepted the interrupt.

ALE:- Address Latch Enable

- This output signal indicates the availability of the valid address on address/data lines. If pin is set high, AD0 to AD15 pins will carry address, else they will carry data.

DT/$\overline{R}$:- Data Transmit/Receive

- This output is used to decide the direction of data flow through the transreceivers(bidirectional buffers).
- Signal is high when processor sends out data and low when processor receives data.

$\overline{DEN}$:-Data Enable

- This signal indicates the availability of valid data over the address/data lines.
- It is used to enable the data transreceivers to separate the data from the multiplexed address/data signal. It is active from middle of T2 to T4.

HOLD, HLDA:- Hold/Hold Acknowledge

- When the HOLD line goes high, it indicates to the processor that another master is requesting the bus access.

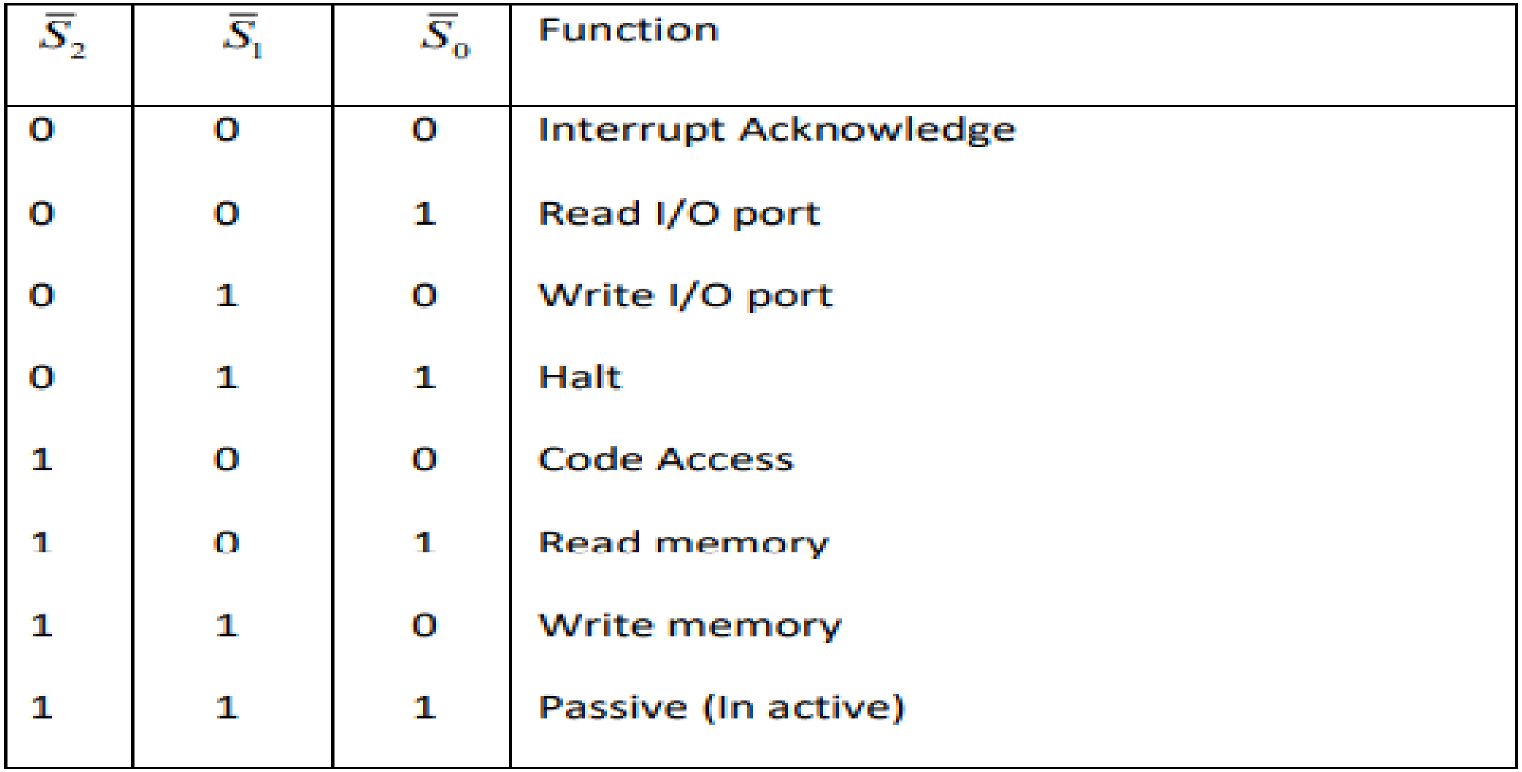
- The processor, after receiving the HOLD request, issues the hold acknowledgement signal on HLDA pin, in the middle of the next clock cycle after completing the current bus cycle.

Pins for the maximum mode operation

$\overline{S2}$,$\overline{S1}$,$\overline{S0}$ – status lines

- It indicates the type of operation carried out by the processor.

- Status lines are encoded as follows:



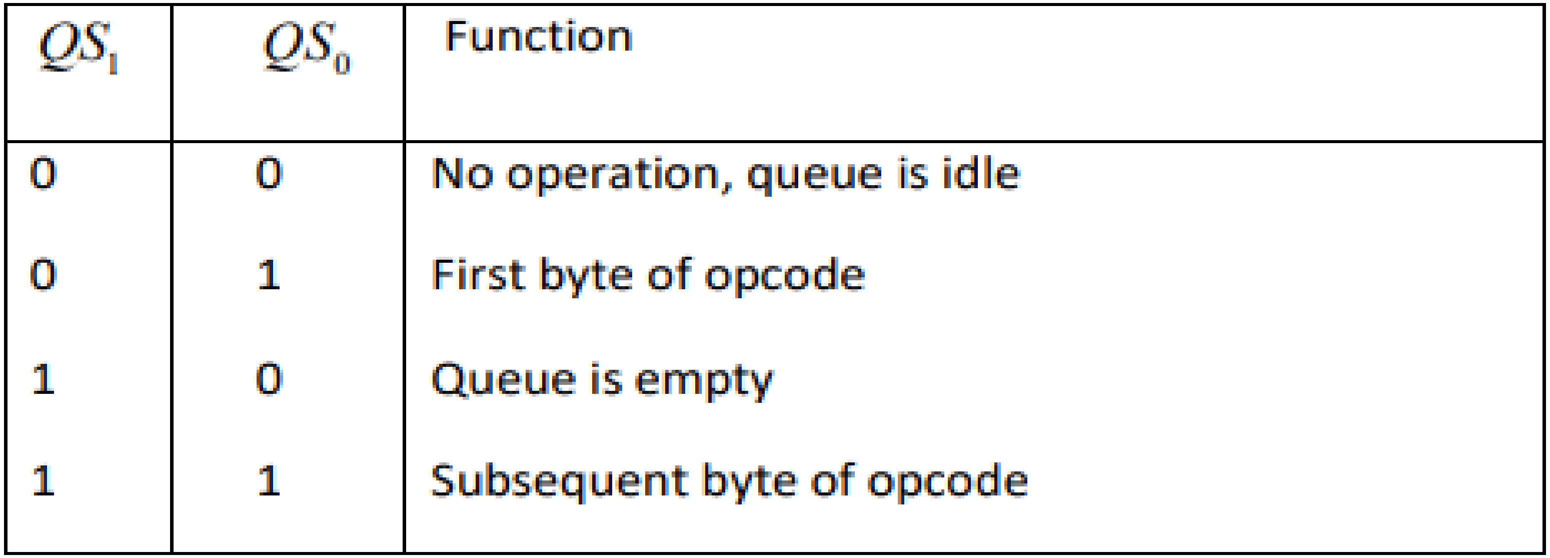
$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	Function
0	0	0	Interrupt Acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code Access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive (In active)

LOCK :

- It indicates that other system bus masters will be prevented from gaining the system bus, while the lOCK signal is low.

QS1, QS0- queue status

- These lines give information about the status of the code-prefetch queue.
- The encoding of the signals are as follows:



QS_1	QS_0	Function
0	0	No operation, queue is idle
0	1	First byte of opcode
1	0	Queue is empty
1	1	Subsequent byte of opcode

$\overline{RQ}$/$\overline{GT0}$, $\overline{RQ}$/$\overline{GT1}$: Request/grant

- These pins are used by other local bus masters, to force the processor to release the local bus at the end of the processors current bus cycle.

Physical Memory Organisation

- In 8086 based system, 1MB memory is physically organized as an odd bank and an even bank, each of 512 Kbytes, addressed in parallel by the processor.

- Byte data with an even bank address is transferred on D7-D0, while the byte data with an odd address is transferred on D15-D8 bus lines.

- The processor provides two enable signals, $\overline{\text{BHE}}$ and A0 for selection of either even or odd or both the banks.

- Commercially available memory chips are only 1 Byte size that is they can store only one byte in one memory location.

- To store 16 bit data, two successive memory locations are used and the lower byte of 16 bit data can be stored in first memory location while the second byte is stored in next location.

- In a 16 bit read or write operation both of these bytes will be read or written in a single machine cycle.

- Bits D0-D7 of a 16 bit data will be transferred over D0 - D7 (lower byte) of 16 bit data bus to/from 8bit memory.
- Bits D8-D15 of the 16 bit data will be transferred over D8 - D15 (higher byte)of the 16 bit data bus to/from 8 bit memory.

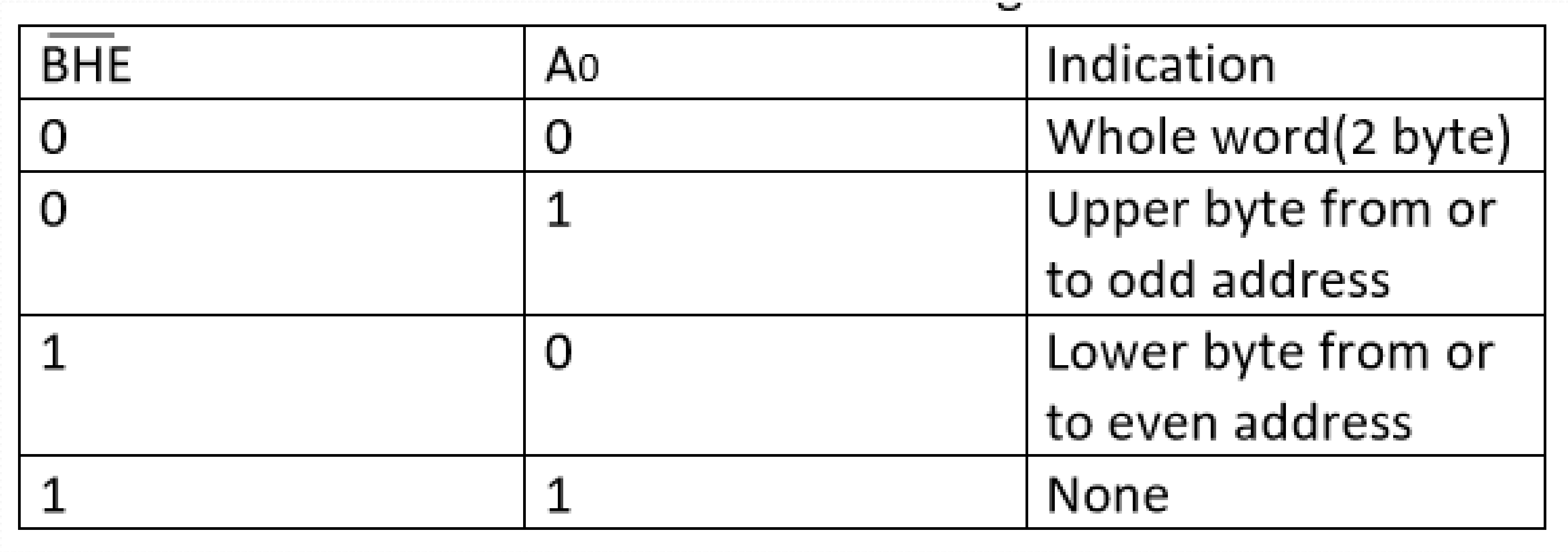
- The lower byte of a 16 bit data is stored at the first address of the map 00000H and is to be transferred over D0-D7 of the microprocessor bus.
- Higher byte of the 16 bit data is stored in the next address 00001H is to be transferred over D8-D15 of the microprocessor bus.

- For next 16 bit data, lower byte will be stored at the next address 00002H while the higher byte will be stored at 00003H.

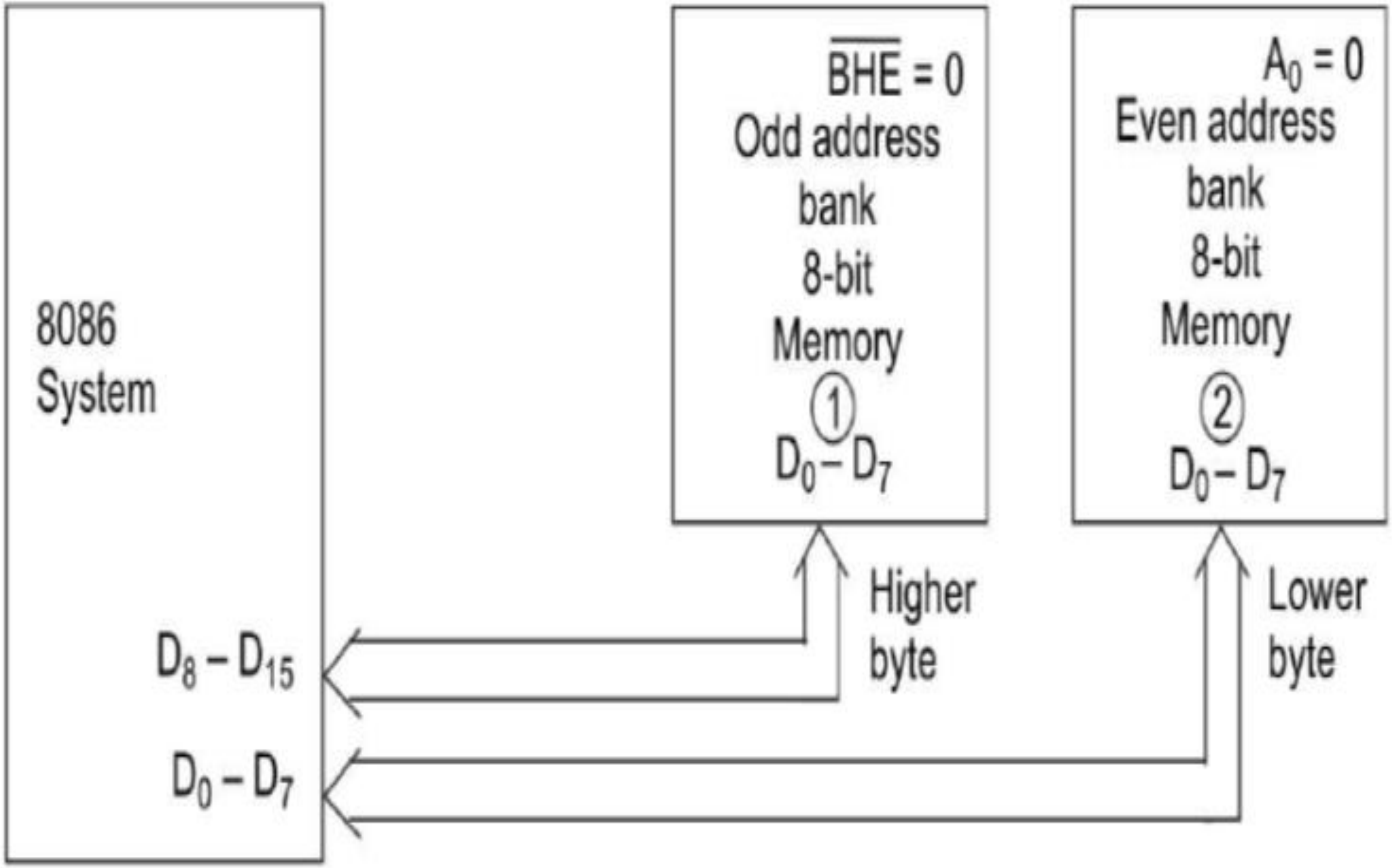
- Thus all lower bytes of each 16 bit data have to be stored at even address and all the higher bytes have to be stored at odd address. Hence the complete memory map of 8086 system is divided into even and odd address memory banks.

- If 8086 transfer a 16 bit data to/from memory, both of these banks must be selected for 16 bit operation.

- To maintain the upward compactibility with 8085, 8086 must be able to implement 8 bit operations.(i.e. 8 bit operations with even/odd address banks) The two signals A0 and $\overline{\text{BHE}}$ solve the problem of selection of appropriate memory banks.

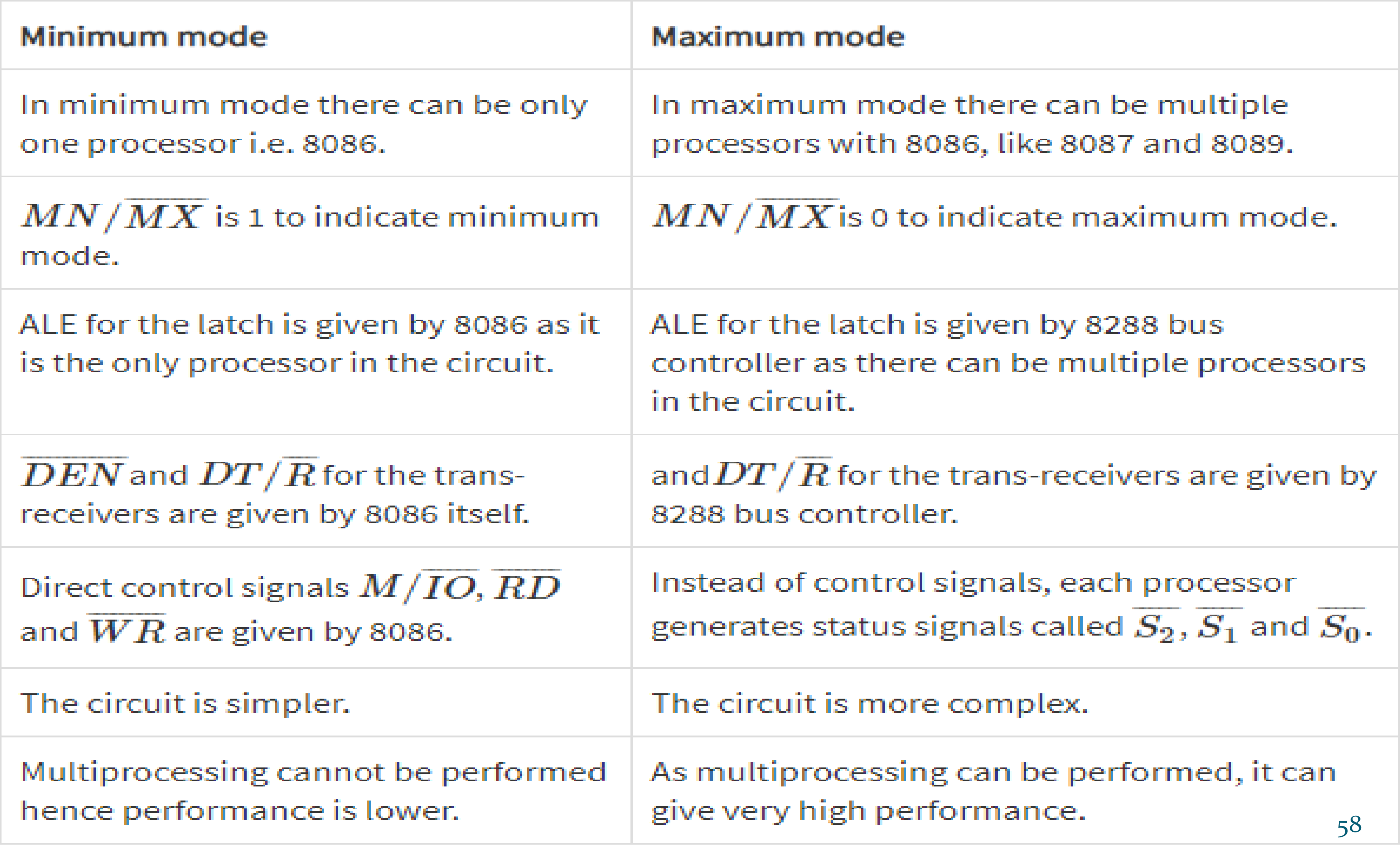


$\overline{\text{BHE}}$	A_0	Indication
0	0	Whole word(2 byte)
0	1	Upper byte from or to odd address
1	0	Lower byte from or to even address
1	1	None



Physical Memory Organisation

Minimum and Maximum Mode of Operation



Minimum mode	Maximum mode
In minimum mode there can be only one processor i.e. 8086.	In maximum mode there can be multiple processors with 8086, like 8087 and 8089.
$MN/\overline{MX}$ is 1 to indicate minimum mode.	$MN/\overline{MX}$ is 0 to indicate maximum mode.
ALE for the latch is given by 8086 as it is the only processor in the circuit.	ALE for the latch is given by 8288 bus controller as there can be multiple processors in the circuit.
$\overline{DEN}$ and $DT/\overline{R}$ for the trans-receivers are given by 8086 itself.	and $DT/\overline{R}$ for the trans-receivers are given by 8288 bus controller.
Direct control signals $M/\overline{IO}$, $\overline{RD}$ and $\overline{WR}$ are given by 8086.	Instead of control signals, each processor generates status signals called $\overline{S_2}$, $\overline{S_1}$ and $\overline{S_0}$.
The circuit is simpler.	The circuit is more complex.
Multiprocessing cannot be performed hence performance is lower.	As multiprocessing can be performed, it can give very high performance.

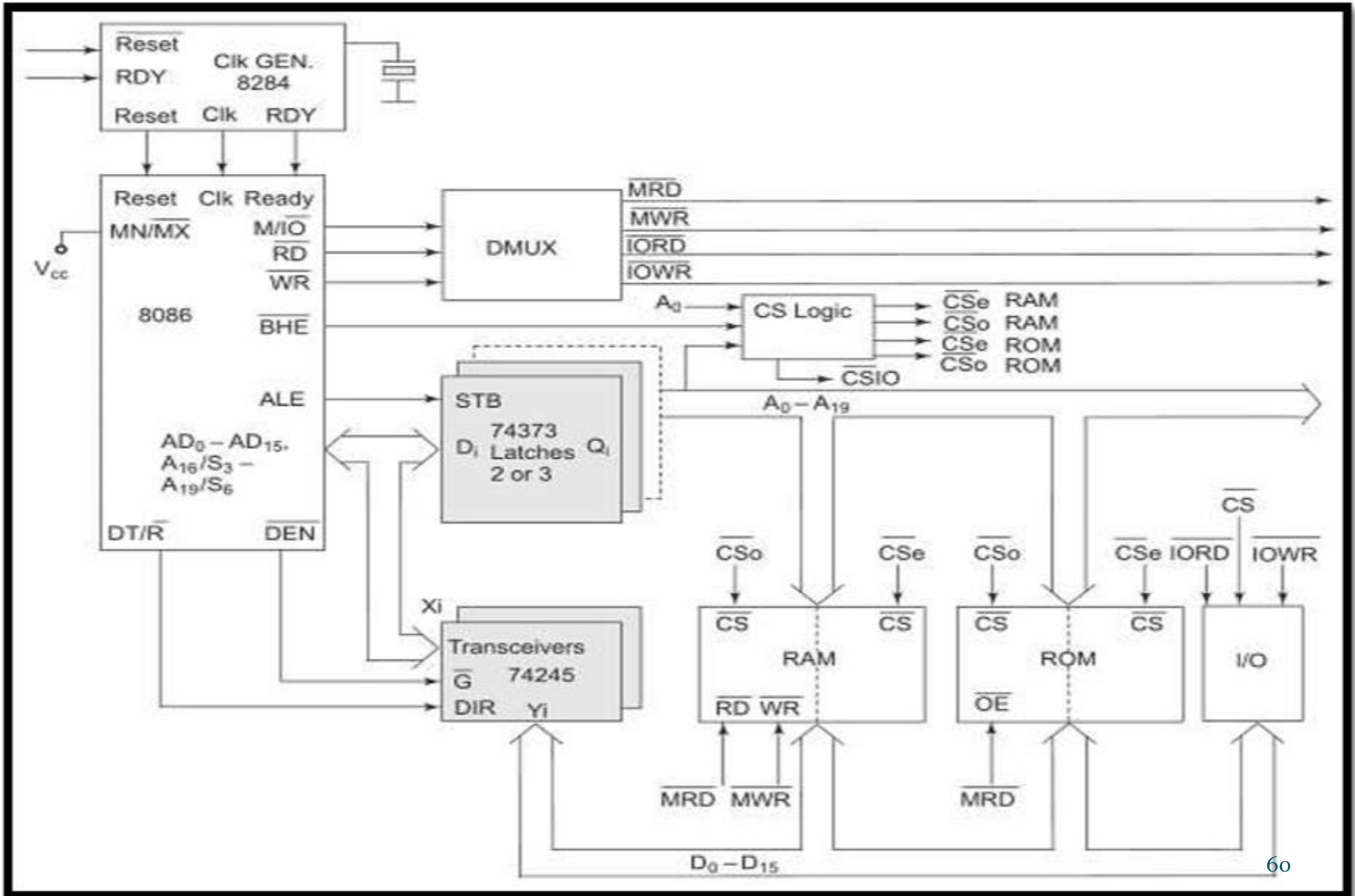
Minimum Mode System

- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its $MN/\overline{MX}$ pin to logic 1.

- In this mode, all the control signals are given out by the microprocessor chip itself. There is a single microprocessor i.e. 8086 in the minimum mode system.

- The remaining components in the system are latches, transreceivers, clock generator, memory and I/O devices.

Minimum Mode 8086 System

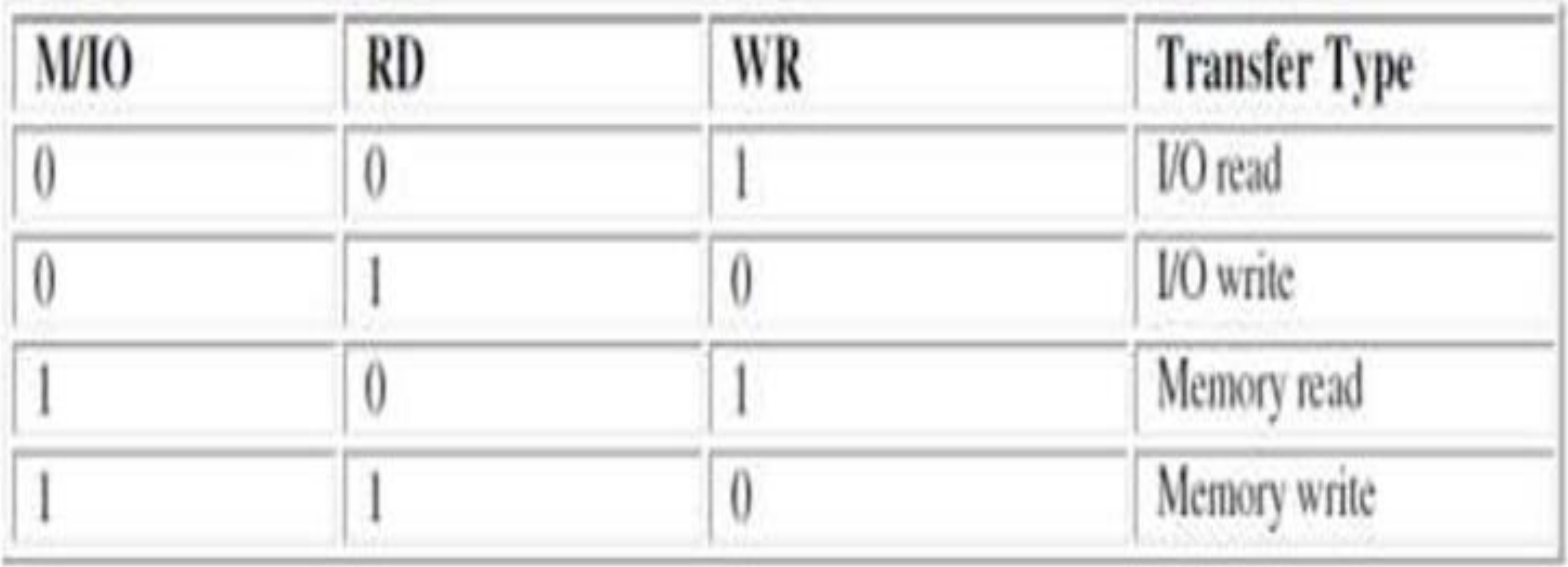


- **Latches** : They are generally buffered output D-type flip-flops like 74LS373 or 8282. They are used for separating the valid address from the multiplexed address/data signals and are controlled by the ALE signal generated by 8086.

- **Trans-receivers** are the bidirectional buffers and sometimes they are called as data amplifiers. They are required to separate the valid data from the time multiplexed address/data signals.
  - They are controlled by two signals namely, $\overline{DEN}$ and DT/$\overline{R}$.
  - The $\overline{DEN}$ signal indicates the availability of valid data over the address/data lines. The DT/R signal indicates direction of data, i.e. from or to the processor.

- Since it has 20 address lines and 16 data lines, the 8086 CPU requires three octal address latches and two octal data buffers for the complete address and data separation.

- The $M/\overline{IO}$, $\overline{RD}$ and $\overline{WR}$ signals indicate the types of data transfer as specified in the below table



M/IO	RD	WR	Transfer Type
0	0	1	I/O read
0	1	0	I/O write
1	0	1	Memory read
1	1	0	Memory write

- The system contains **memory** for the monitor and users program storage.
  - Usually, EPROM are used for monitor storage, while RAM for users program storage.

- A system may contain I/O devices for communication with the processor as well as some special purpose i/o devices.

- Clock generator generates the clock from the crystal oscillator, and it can be used as an accurate timing reference for the system. It also synchronizes some external signals with the system clock.

Timing diagram of Minimum Mode

- The timing diagram can be categorized in two parts,
  - the Timing diagram for read cycle
  - the Timing diagram for write cycle.

- The **read cycle** begins in T1 with the assertion of address latch enable (ALE) signal and also M / $\overline{IO}$ signal.

- During the negative going edge of this signal, the valid address is latched on the local bus.

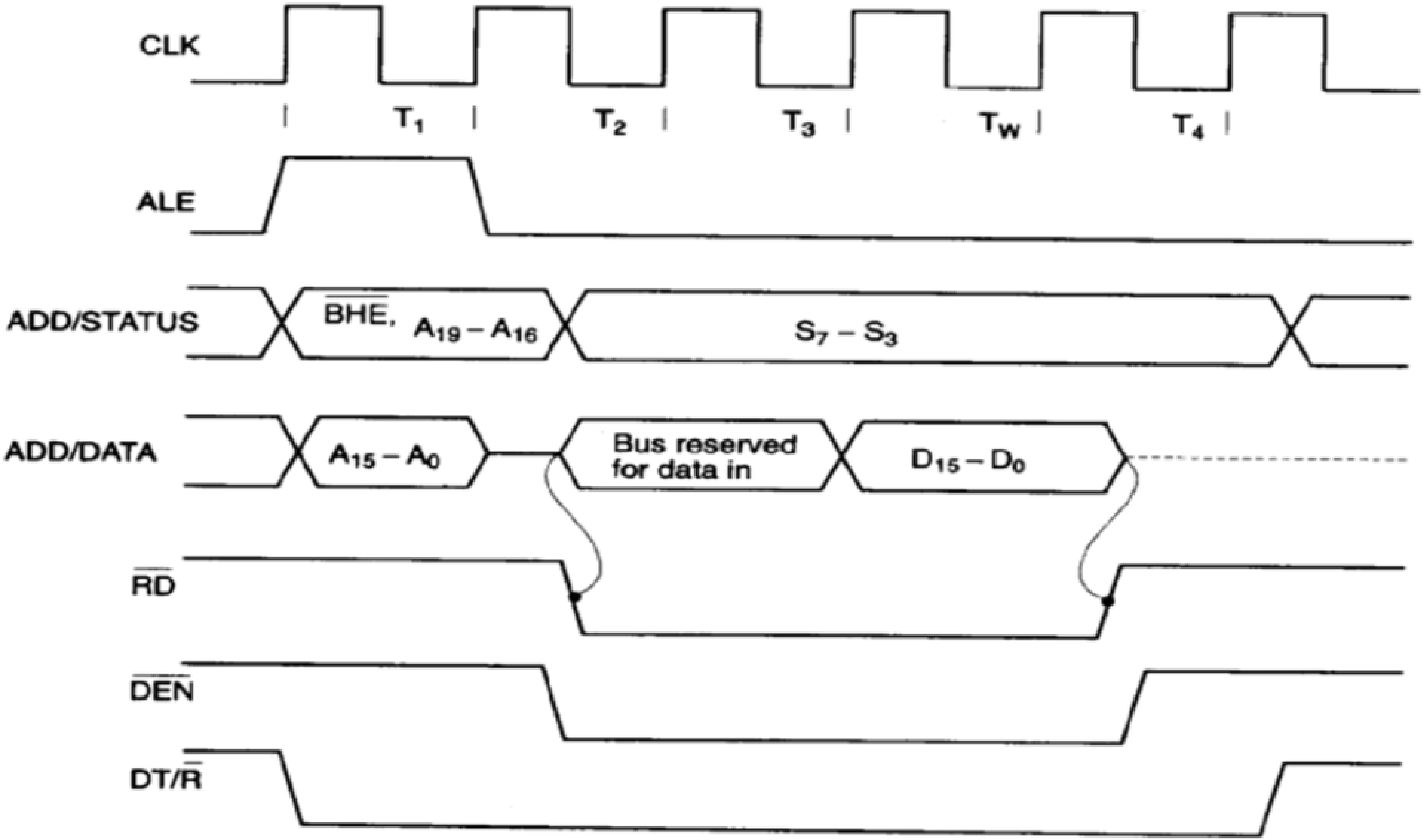
- The $\overline{\text{BHE}}$ and A0 signals address low, high or both bytes. From T1 to T4, the M/IO signal indicates a memory or I/O operation.

- At T2, the address is removed from the local bus and is sent to the output. The bus is then tri-stated. The read ($\overline{\text{RD}}$) control signal is also activated in T2.

- The read ($\overline{\text{RD}}$) signal causes the address device to enable its data bus drivers. After $\overline{\text{RD}}$ goes low, the valid data is available on the data bus.

- The addressed device will drive the READY line high. When the processor returns the read signal to high level, the addressed device will again tri-state its bus drivers.

Read cycle timing diagram for minimum mode



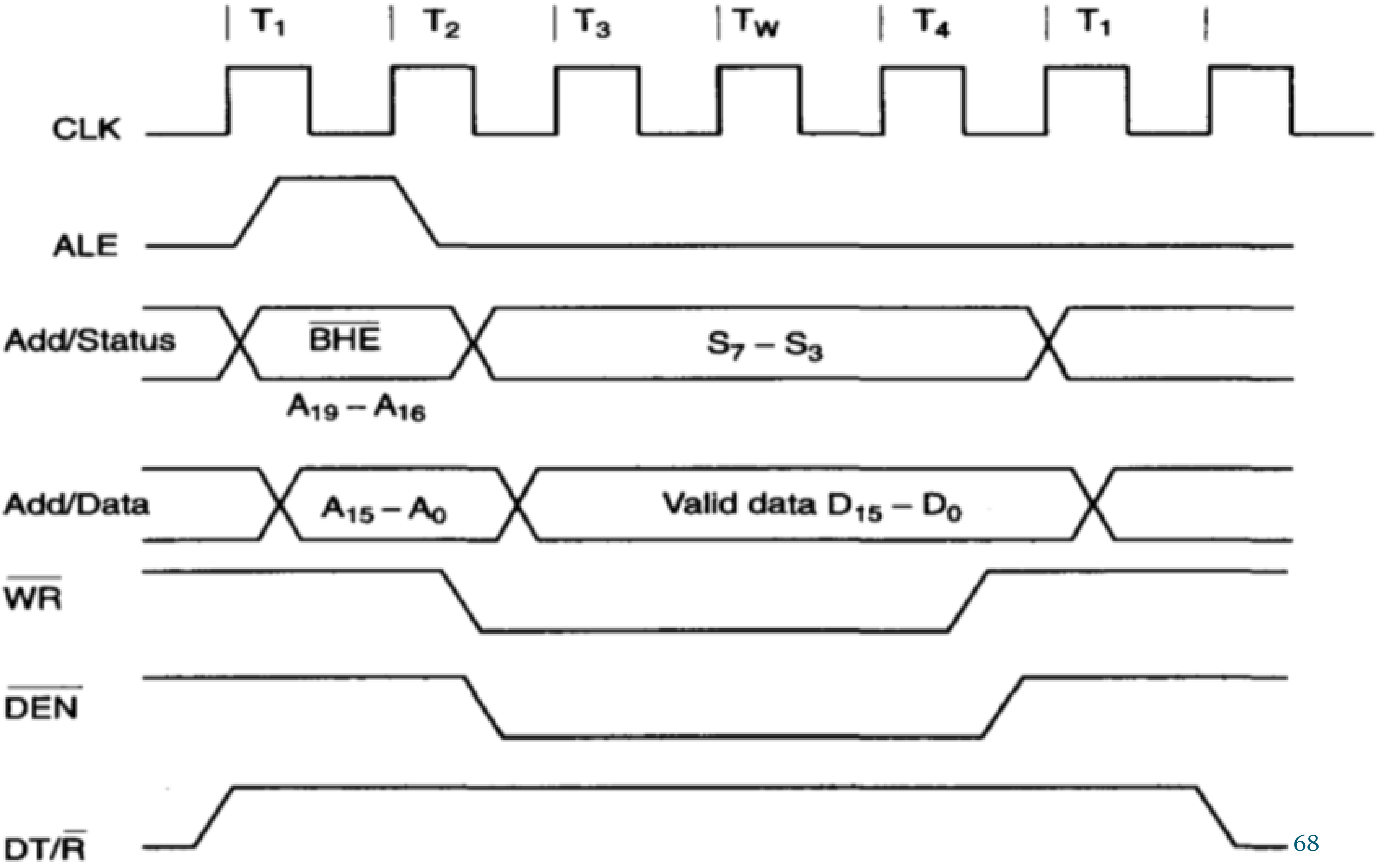
- A **write cycle** also begins with the assertion of ALE and the emission of the address. The M/$\overline{\text{IO}}$ signal is again asserted to indicate a memory or I/O operation.

- In T2, after sending the address in T1, the processor sends the data to be written to the addressed location.

- The data remains on the bus until middle of T4 state. The $\overline{\text{WR}}$ becomes active at the beginning of T2 (unlike $\overline{\text{RD}}$ which is somewhat delayed in T2 to provide time for floating).

- The $\overline{\text{BHE}}$ and A0 signals are used to select the proper byte or bytes of memory or I/O word to be read or write.

Write cycle timing diagram for minimum mode



Comparison of 8086 and 8088

8086	8088
In case of 8086 MPU the data bus is of 16 bits and the address bus is of 20 bits.	In 8088 MPU the data bus is of 8 bits and the address bus is of 20 bits.
In 8086, all address & data Buses are multiplexed.	In 8088, address bus; AD7- AD0 buses are multiplexed.
It needs two IC 74343 for de-multiplexing AD0-AD19.	It needs one IC 74343 for de-multiplexing AD0-AD7.
In 8086, memory divides into two banks even or lower bank and odd or higher bank	The memory in 8088 does not divide into two banks.
It can read or write either 8-bit or 16-bit word at the same time.	It can read only 8-bit word at the same time.
The instruction Queue is 6 byte long. BIU will fetch next instruction only when atleast 2 bytes of queue is free.	The instruction Queue is 4 byte long. BIU will fetch next instruction each time when 1 byte of queue is free.
It has BHE (bar) signal on pin no. 34 & there is no SSO (bar) signal.	It does not have BHE (bar) signal on pin no. 34 & has only SSO (bar) signal. It has no S7 pin.
Control pin in 8086 is M/IO (bar).	Control pin in 8088 is IO/M (bar).