



Microprocessor Architecture and Interfacing: Intel x86

Module 2

FEATURES OF 8086 MICROPROCESSOR

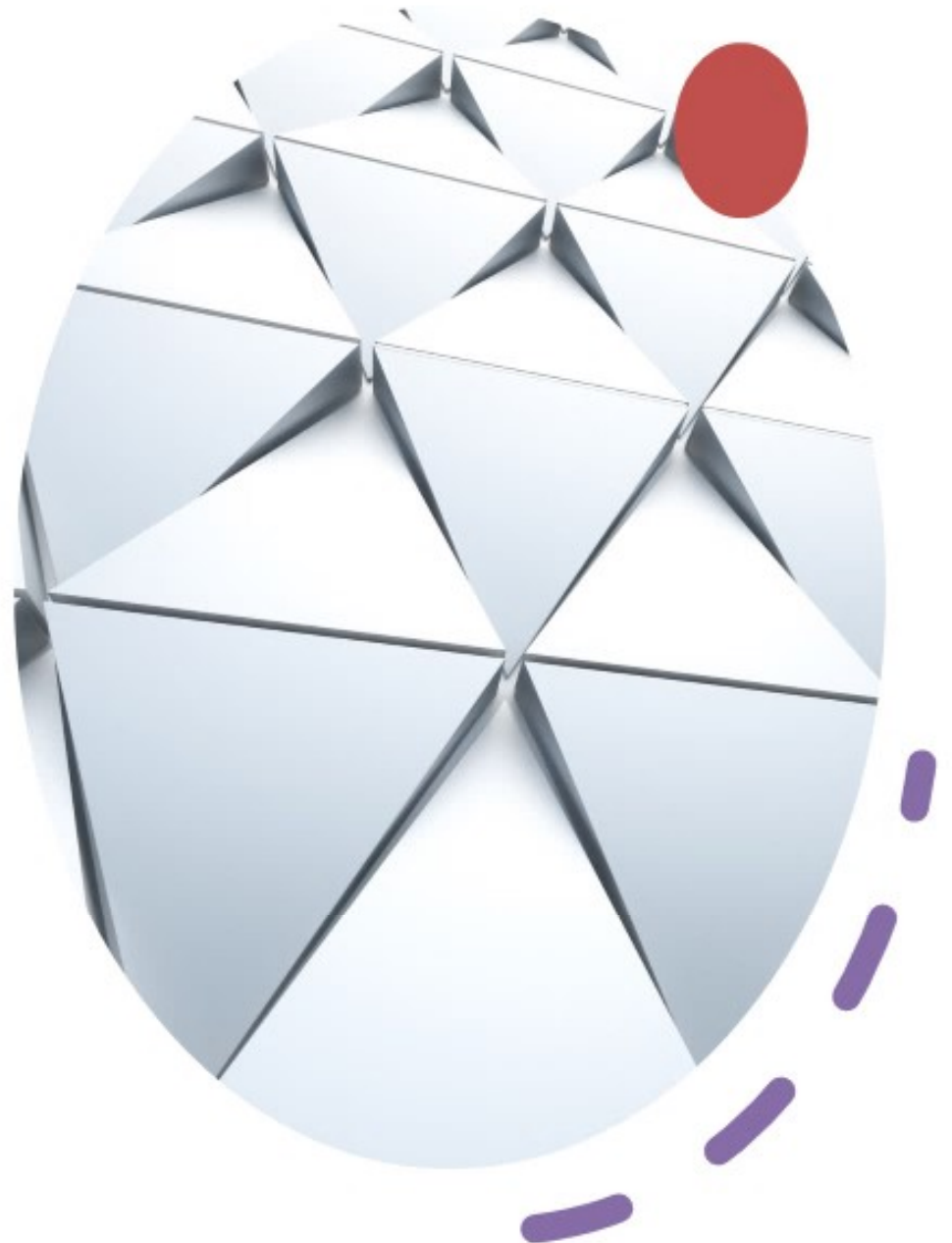
- It is a 16-bit μ p
- Single +5V power supply
- Clock speed range of 5-10MHz capable of executing about 0.33 MIPS (Millions instructions per second)
- 8086 has a 20 bit address bus can access up to 2^{20} memory locations (1 MB).
- It can support up to 64K I/O ports.
- It is 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It uses two stages of pipelining. It can pre-fetches up to 6 instruction bytes from memory and queues them in order to speed up instruction execution
- It has 256 interrupts.
- 8086 is designed to operate in two modes, Minimum and Maximum.
- 40 pin dual in line package.

Architecture of 8086

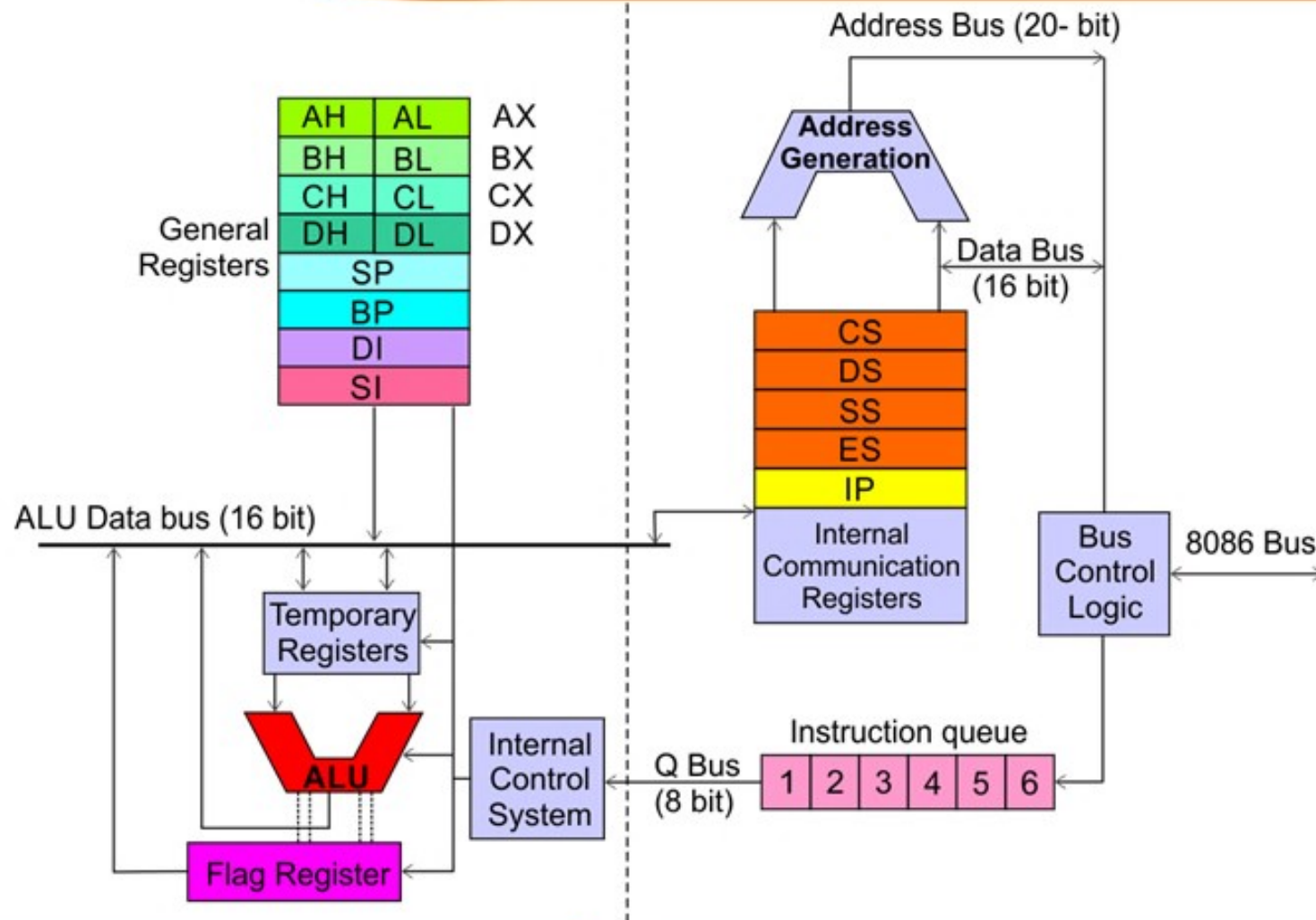
The 8086 CPU is divided into two independent functional units:

1. Bus Interface Unit (BIU)

2. Execution Unit (EU)



Architecture



Execution Unit (EU)

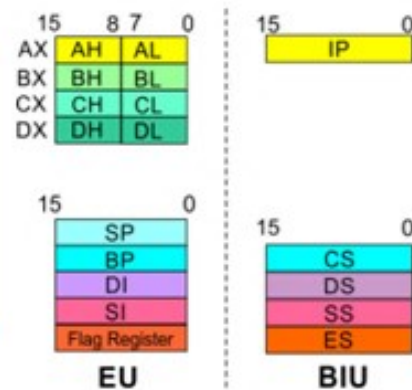
EU executes instructions that have already been fetched by the BIU.

BIU and EU functions separately.

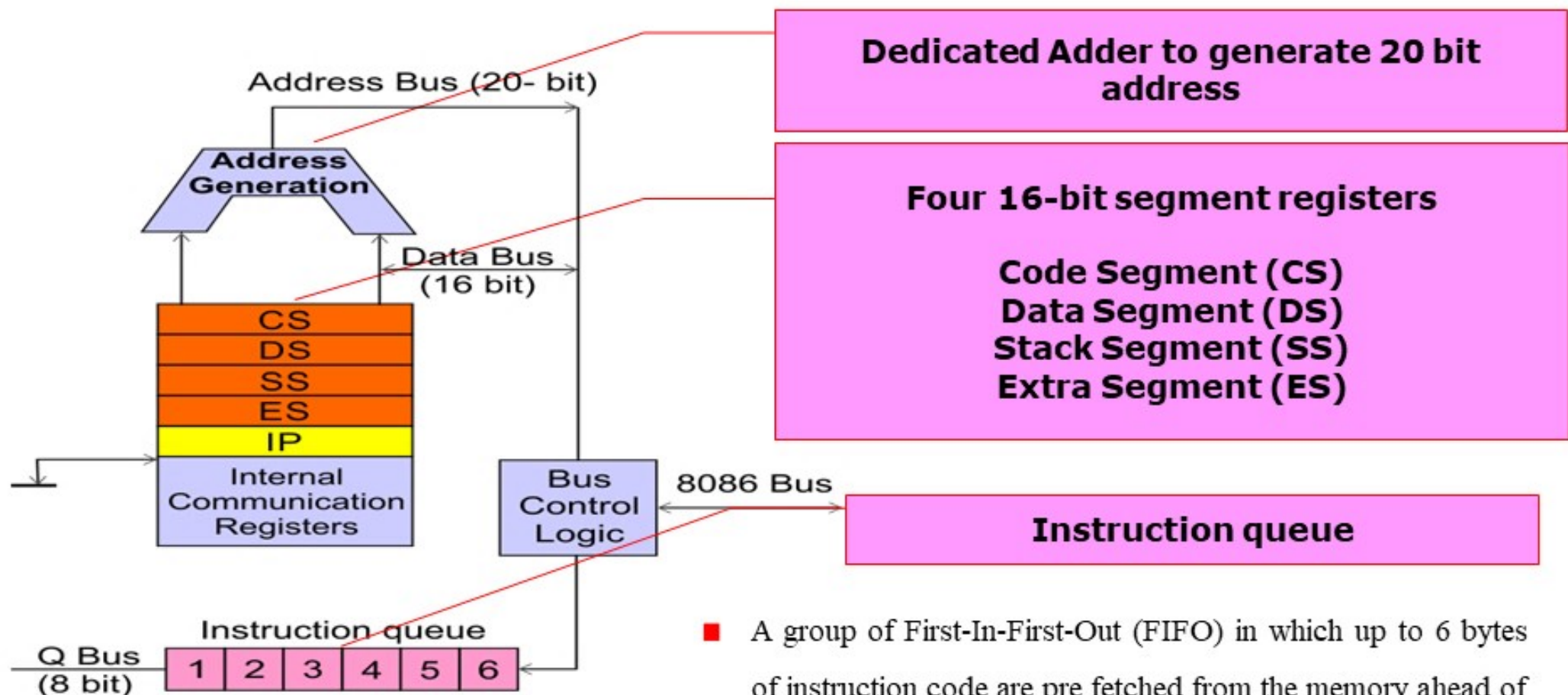
Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/O ports.

**8086 registers
categorized
into 4 groups**



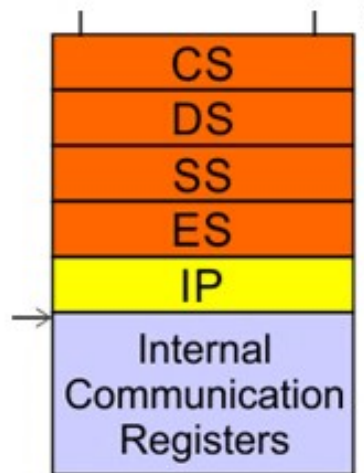
Sl.No.	Type	Register width	Name of register
1	General purpose register	16 bit	AX, BX, CX, DX
		8 bit	AL, AH, BL, BH, CL, CH, DL, DH
2	Pointer register	16 bit	SP, BP
3	Index register	16 bit	SI, DI
4	Instruction Pointer	16 bit	IP
5	Segment register	16 bit	CS, DS, SS, ES
6	Flag (PSW)	16 bit	Flag register



Bus Interface Unit (BIU)

- A group of First-In-First-Out (FIFO) in which up to 6 bytes of instruction code are pre fetched from the memory ahead of time.
- This is done in order to speed up the execution by overlapping instruction fetch with execution.
- This mechanism is known as pipelining.
- It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.

Segment Registers



- **8086's 1-megabyte memory is divided into segments of up to 64K bytes each.**
 - **The 8086 can directly address four segments (256 K bytes within the 1 M byte of memory) at a particular time.**
 - **Programs obtain access to code and data in the segments by changing the segment register content to point to the desired segments.**
-
- **CS (Code Segment Register)**
 - **DS (Data Segment Register)**
 - **SS (Stack Segment Register)**
 - **ES (Extra Segment Register)**
 - **IP (Instruction Pointer)**

Segment Registers

Code Segment Register

- **16-bit**
- **CS** contains the base or start of the current code segment; **IP** contains the distance or offset from this address to the next instruction byte to be fetched.
- **BIU** computes the 20-bit physical address by logically shifting the contents of **CS** 4-bits to the left and then adding the 16-bit contents of **IP**.
- That is, all instructions of a program are relative to the contents of the **CS** register multiplied by 16 and then offset is added provided by the **IP**.

Data Segment Register

- **16-bit**
- Points to the current data segment; operands for most instructions are fetched from this segment.
- The 16-bit contents of the **Source Index (SI)** or **Destination Index (DI)** or a 16-bit displacement are used as offset for computing the 20-bit physical address.

Segment Registers

Stack Segment Register

- 16-bit
- Points to the current stack.
- The 20-bit physical stack address is calculated from the Stack Segment (SS) and the Stack Pointer (SP) for stack instructions such as **PUSH** and **POP**.
- In based addressing mode, the 20-bit physical stack address is calculated from the Stack segment (SS) and the Base Pointer (BP).

Extra Segment Register

- 16-bit
- Points to the extra segment in which data (in excess of 64K pointed to by the DS) is stored.
- String instructions use the ES and DI to determine the 20-bit physical address for the destination.

Segment Registers

Instruction Pointer

- **16-bit**
- **Always points to the next instruction to be executed within the currently executing code segment.**
- **So, this register contains the 16-bit offset address pointing to the next instruction code within the 64Kb of the code segment area.**
- **Its content is automatically incremented as the execution of the next instruction takes place.**

EU decodes and executes instructions.

A decoder in the EU control system translates instructions.

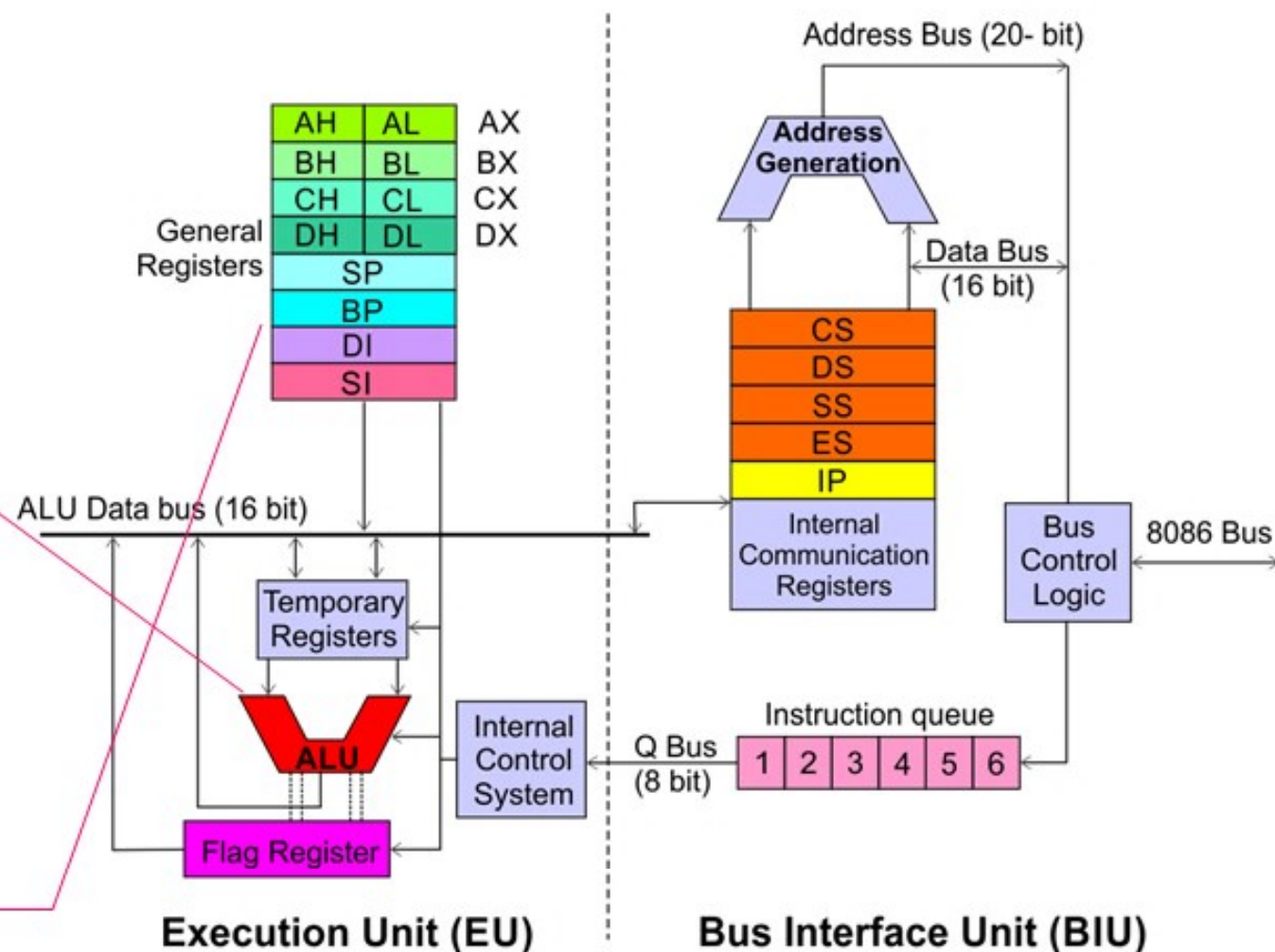
16-bit ALU for performing arithmetic and logic operation

Four general purpose registers (AX, BX, CX, DX);

Pointer registers (Stack Pointer, Base Pointer);

and

Index registers (Source Index, Destination Index) each of 16-bits



Some of the 16 bit registers can be used as two 8 bit registers as :

AX can be used as AH and AL
 BX can be used as BH and BL
 CX can be used as CH and CL
 DX can be used as DH and DL

General Purpose/ EU Registers

Accumulator Register (AX)

- Consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX.
- AL in this case contains the low order byte of the word, and AH contains the high-order byte.
- The I/O instructions use the AX or AL for inputting / outputting 16 or 8 bit data to or from an I/O port.
- Multiplication and Division instructions also use the AX or AL.

Base Register (BX)

- Consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX.
- BL in this case contains the low-order byte of the word, and BH contains the high-order byte.
- This is the only general purpose register whose contents can be used for addressing the 8086 memory.
- All memory references utilizing this register content for addressing use DS as the default segment register.

General Purpose/ EU Registers

Counter Register (CX)

- Consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX.
- When combined, CL register contains the low order byte of the word, and CH contains the high-order byte.
- Instructions such as **SHIFT**, **ROTATE** and **LOOP** use the contents of CX as a counter.

Data Register (DX)

- Consists of two 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX.
- When combined, DL register contains the low order byte of the word, and DH contains the high-order byte.
- Used to hold the high 16-bit result (data) in 16 X 16 multiplication or the high 16-bit dividend (data) before a $32 \div 16$ division and the 16-bit remainder after division.

Special Purpose EU Registers

Stack Pointer (SP) and Base Pointer (BP)

- **SP** and **BP** are used to access data in the stack segment.
- **SP** is used as an offset from the current **SS** during execution of instructions that involve the stack segment in the external memory.
- **SP** contents are automatically updated (incremented/ decremented) due to execution of a **POP** or **PUSH** instruction.
- **BP** contains an offset address in the current **SS**, which is used by instructions utilizing the based addressing mode.

Source Index (SI) and Destination Index (DI)

- Used in indexed addressing.
- Instructions that process data strings use the **SI** and **DI** registers together with **DS** and **ES** respectively in order to distinguish between the source and destination addresses.

PSW or Flag Register

Flag Register

Sign Flag

This flag is set, when the result of any computation is negative

Auxiliary Carry Flag

This is set, if there is a carry from the lowest nibble, i.e, bit three during addition, or borrow for the lowest nibble, i.e, bit three, during subtraction.

Carry Flag

This flag is set, when there is a carry out of MSB in case of addition or a borrow in case of subtraction.

Zero Flag

This flag is set, if the result of the computation or comparison performed by an instruction is zero

Parity Flag

This flag is set to 1, if the lower byte of the result contains even number of 1's ; for odd number of 1's set to zero.



Over flow Flag

This flag is set, if an overflow occurs, i.e, if the result of a signed operation is large enough to accommodate in a destination register. The result is of more than 7-bits in size in case of 8-bit signed operation and more than 15-bits in size in case of 16-bit sign operations, then the overflow will be set.

Tarp Flag

If this flag is set, the processor enters the single step execution mode by generating internal interrupts after the execution of each instruction

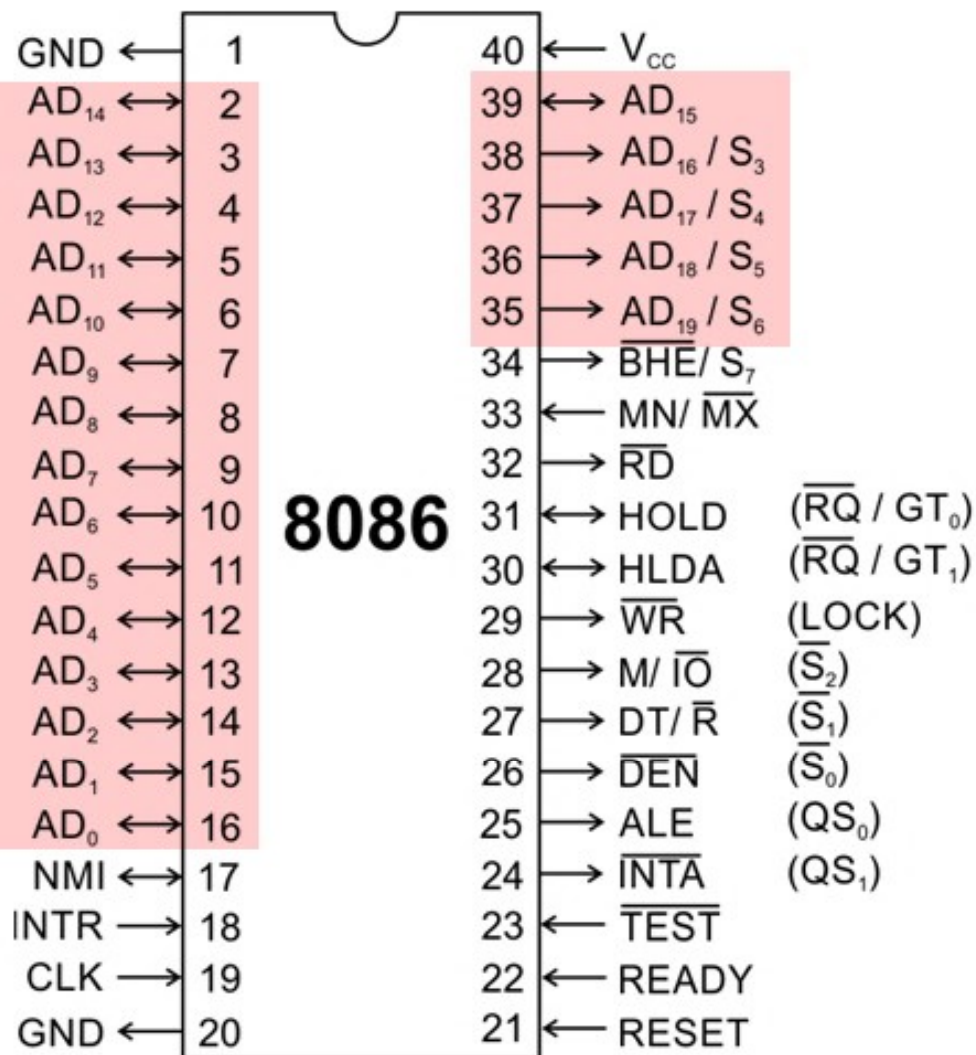
Direction Flag

This is used by string manipulation instructions. If this flag bit is '0', the string is processed beginning from the lowest address to the highest address, i.e., auto incrementing mode. Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto decrementing mode.

Interrupt Flag

Causes the 8086 to recognize external mask interrupts; clearing IF disables these interrupts.

Pins and signals



Power supply

It uses 5V DC supply at VCC pin 40, and uses ground at VSS pin 1 and 20 for its operation.

AD₀-AD₁₅ (Bidirectional)

Address/Data bus

Low order address bus; these are multiplexed with data.

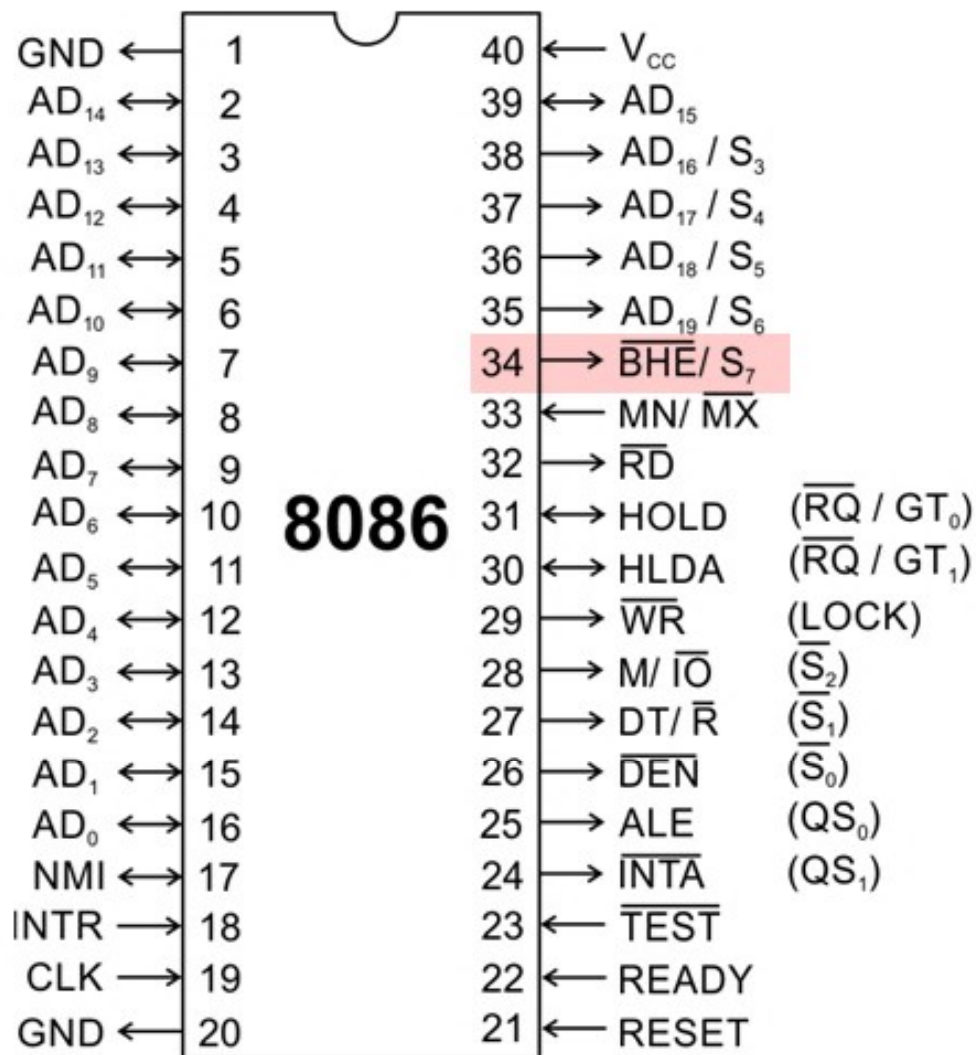
When AD lines are used to transmit memory address the symbol A is used instead of AD, for example A₀-A₁₅.

When data are transmitted over AD lines the symbol D is used in place of AD, for example D₀-D₇, D₈-D₁₅ or D₀-D₁₅.

Address/ status bus

A₁₆/S₃, A₁₇/S₄, A₁₈/S₅, A₁₉/S₆

High order address bus. These are multiplexed with status signals



BHE (Active Low)/ S_7 (Output)

Bus High Enable/Status

It is used to enable data onto the most significant half of data bus, D_8-D_{15} . 8-bit device connected to upper half of the data bus use **BHE (Active Low)** signal. It is multiplexed with status signal S_7 .

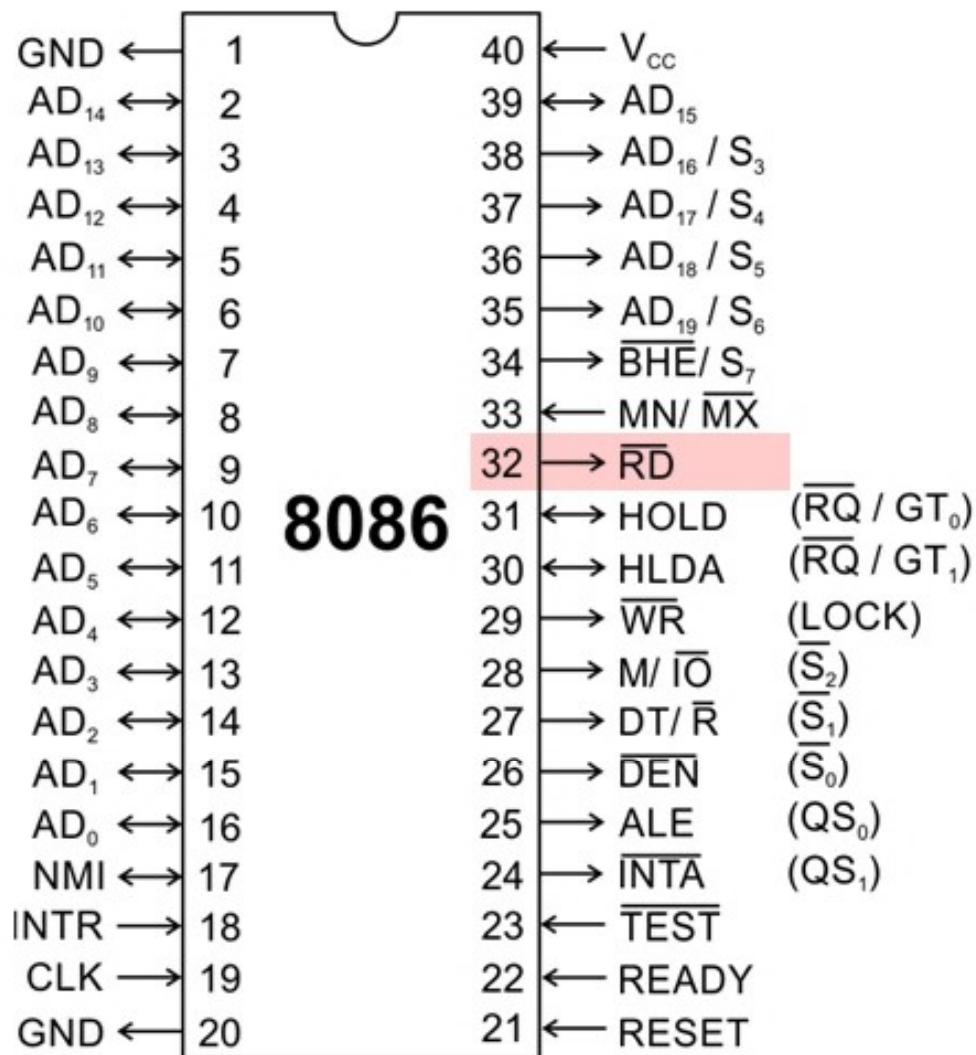
MN/ MX

MINIMUM / MAXIMUM

This pin signal indicates what mode the processor is to operate in.

RD (Read) (Active Low)

The signal is used for read operation.
It is an output signal.
It is active when low.



TEST

$\overline{TEST}$ input is tested by the 'WAIT' instruction.

8086 will enter a wait state after execution of the WAIT instruction and will resume execution only when the $\overline{TEST}$ is made low by an active hardware.

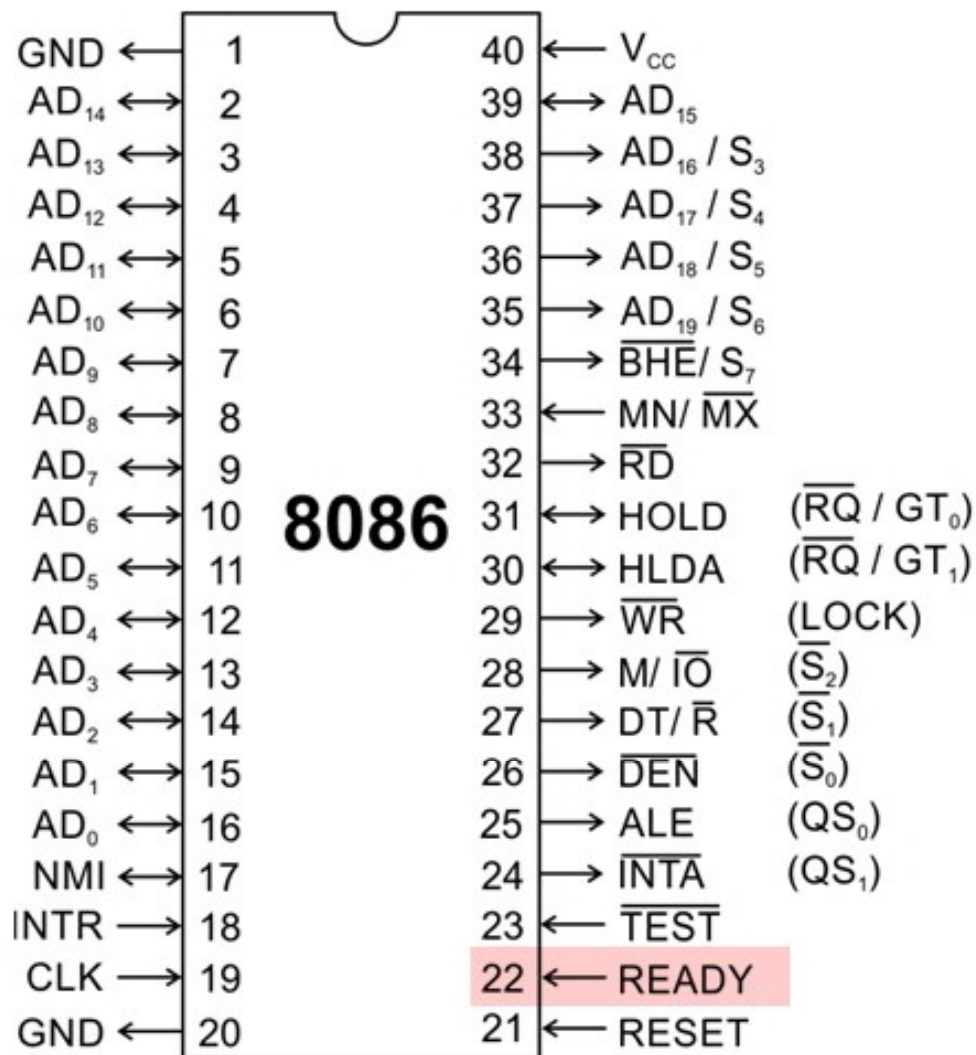
This is used to synchronize an external activity to the processor internal operation.

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**.

The signal is active high.



RESET (Input)

Causes the processor to immediately terminate its present activity.

The signal must be active **HIGH** for at least four clock cycles.

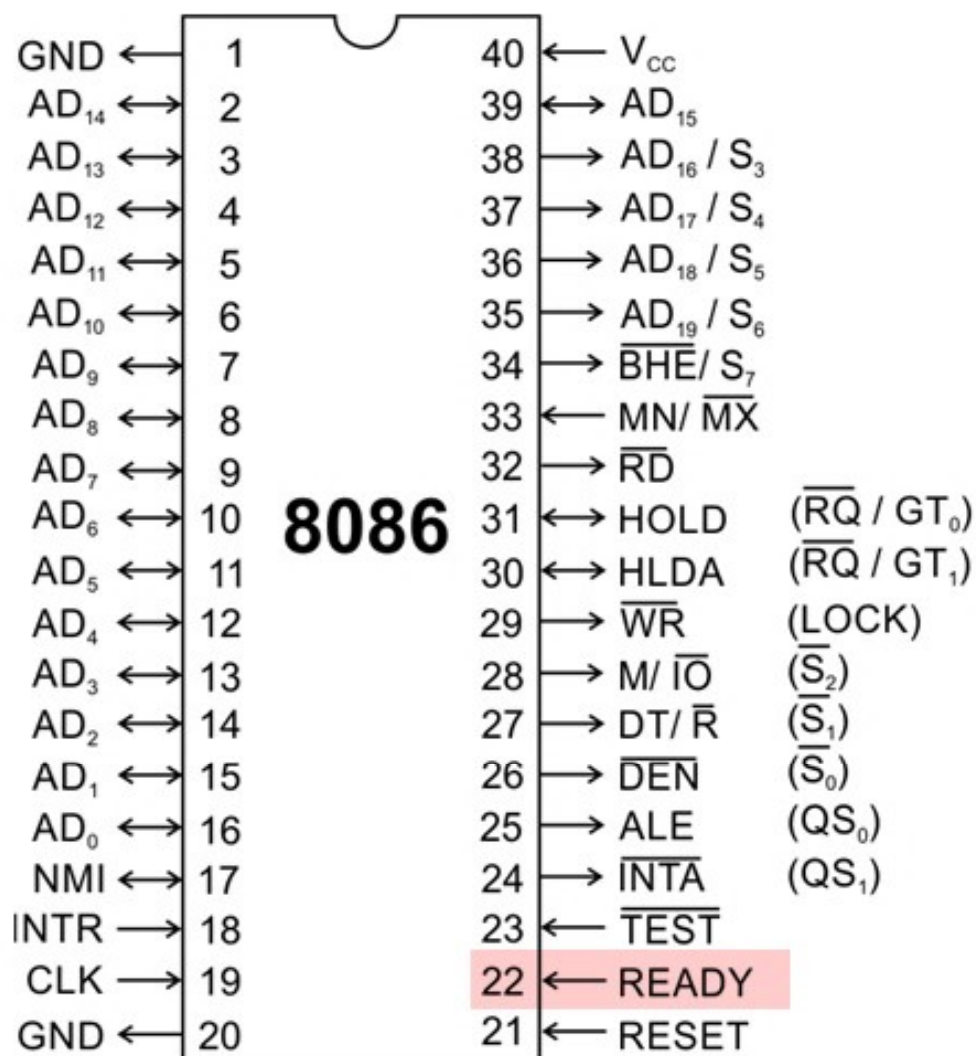
CLK

The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with **33%** duty cycle.

INTR Interrupt Request

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

This signal is active high and internally synchronized.

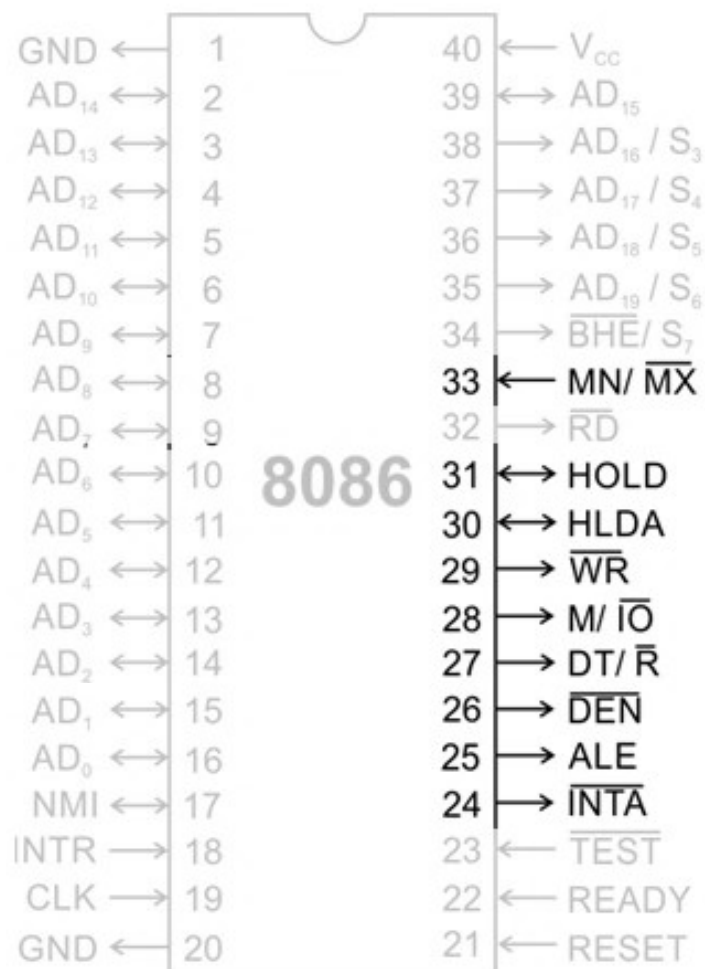


NMI

It stands for non-maskable interrupt and is available at pin 17. It is an edge triggered input, which causes an interrupt request to the microprocessor.

INTA

It is an interrupt acknowledgement signal and is available at pin 24. When the microprocessor receives this signal, it acknowledges the interrupt.



The **8086** microprocessor can work in two modes of operations : **Minimum mode** and **Maximum mode**.

In the minimum mode of operation the microprocessor do not associate with any co-processors and can not be used for multiprocessor systems.

In the maximum mode the **8086** can work in multi-processor or co-processor configuration.

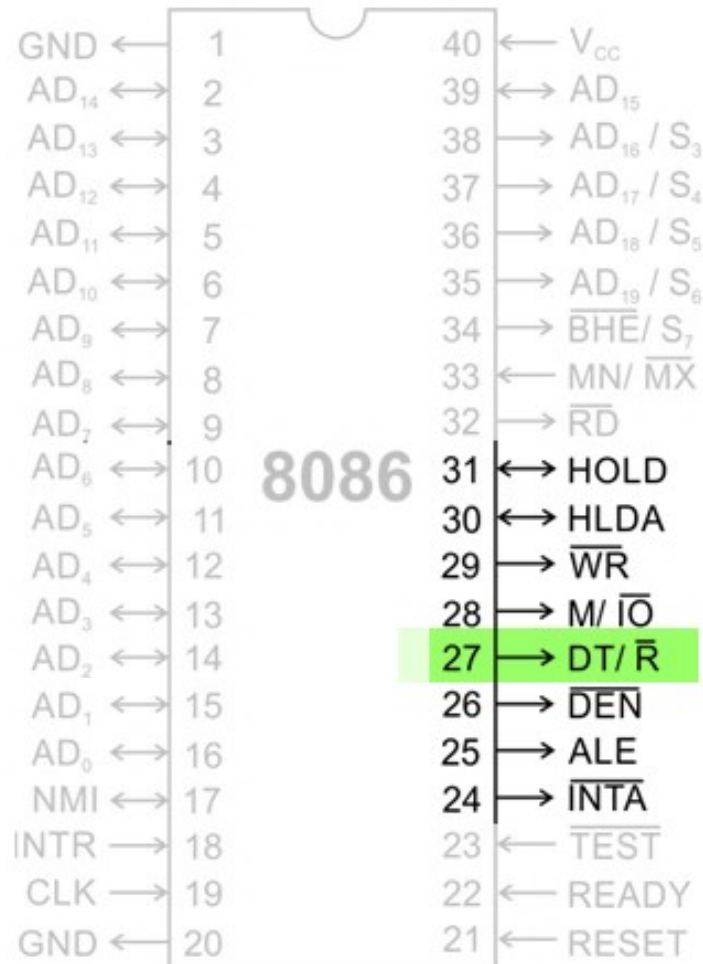
Minimum or maximum mode operations are decided by the pin MN/ MX(Active low).

When this pin is high **8086** operates in minimum mode otherwise it operates in **Maximum mode**.

Pins 24 -31

For minimum mode operation, the $\overline{MN}/\overline{MX}$ is tied to VCC (logic high)

8086 itself generates all the bus control signals



DT/ $\overline{R}$ (**Data Transmit/ Receive**) Output signal from the processor to control the direction of data flow through the data transceivers

$\overline{DEN}$ (**Data Enable**) Output signal from the processor used as out put enable for the transceivers

ALE (**Address Latch Enable**) Used to demultiplex the address and data lines using external latches

M/ $\overline{IO}$ Used to differentiate memory access and I/O access. For memory reference instructions, it is **high**. For IN and OUT instructions, it is **low**.

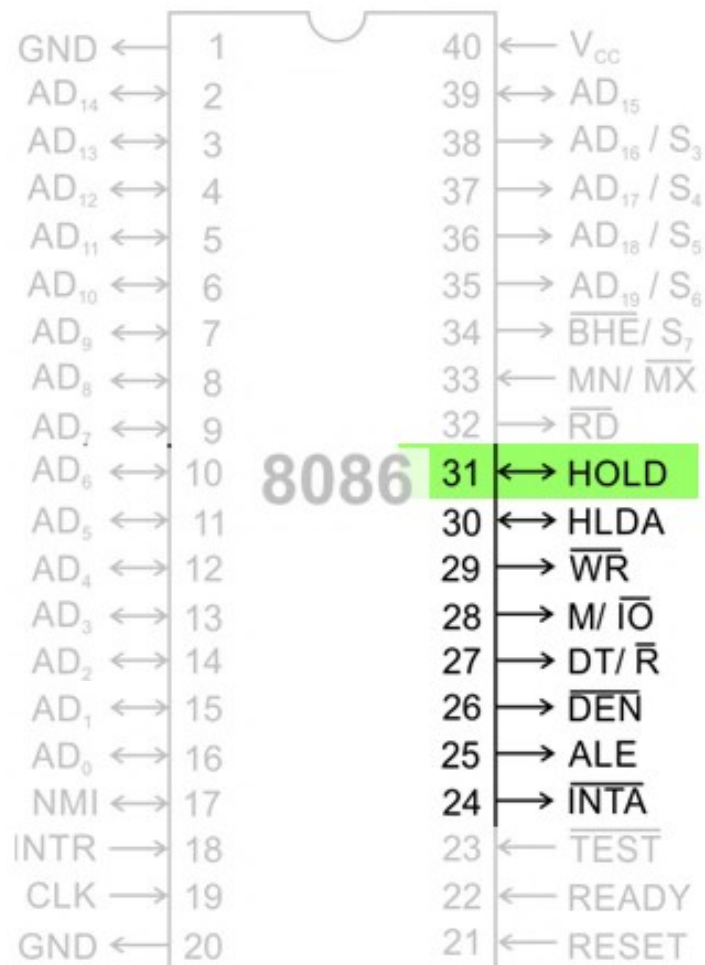
$\overline{WR}$ Write control signal; asserted **low** Whenever processor writes data to memory or I/O port

$\overline{INTA}$ (**Interrupt Acknowledge**) When the interrupt request is accepted by the processor, the output is **low** on this line.

Pins 24 -31

For minimum mode operation, the $\overline{MN}/\overline{MX}$ is tied to VCC (logic high)

8086 itself generates all the bus control signals

**HOLD**

Input signal to the processor from the bus masters as a request to grant the control of the bus.

Usually used by the DMA controller to get the control of the bus.

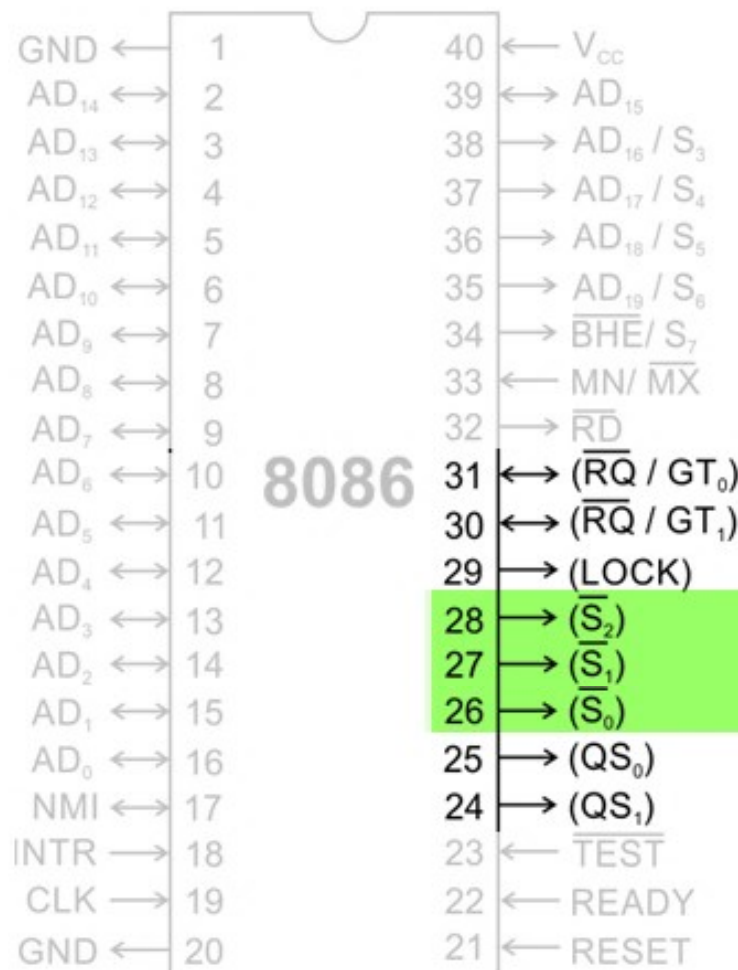
HLDA

(Hold Acknowledge) Acknowledge signal by the processor to the bus master requesting the control of the bus through HOLD.

The acknowledge is asserted high, when the processor accepts HOLD.

During maximum mode operation, the $\overline{MN}/\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



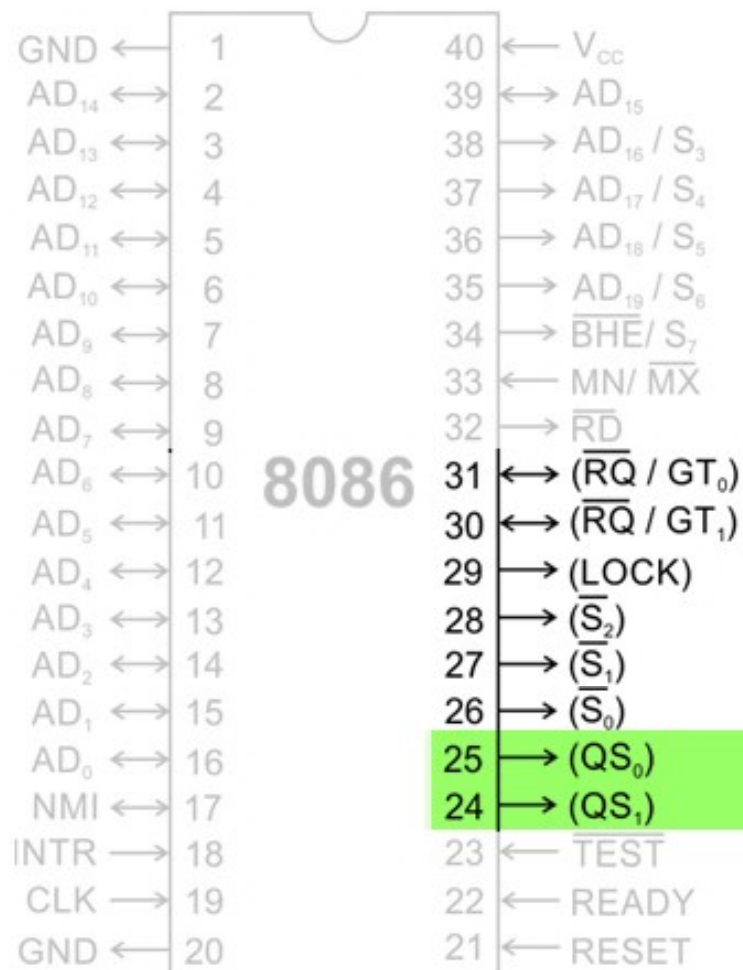
$\overline{S_0}, \overline{S_1}, \overline{S_2}$

Status signals; used by the 8086 bus controller to generate bus timing and control signals. These are decoded as shown.

Status Signal			Machine Cycle
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	
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/Inactive

During maximum mode operation, the $\overline{MN}/\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



$\overline{QS_0}, \overline{QS_1}$

(Queue Status) The processor provides the status of queue in these lines.

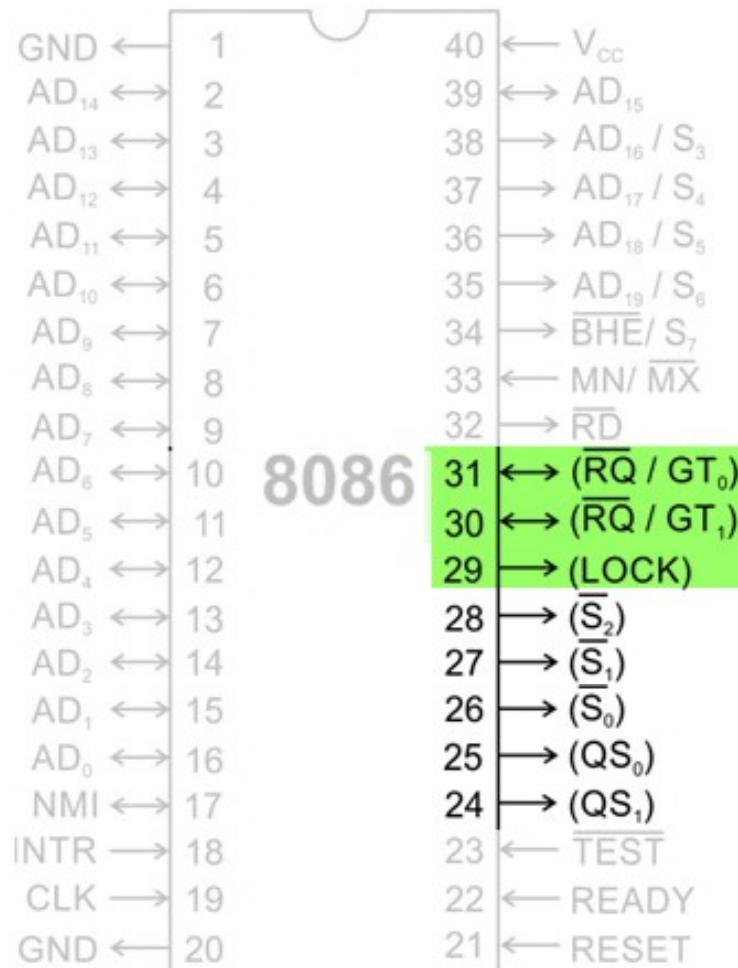
The queue status can be used by external device to track the internal status of the queue in 8086.

The output on QS_0 and QS_1 can be interpreted as shown in the table.

Queue status		Queue operation
QS_1	QS_0	
0	0	No operation
0	1	First byte of an opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

During maximum mode operation, the $\overline{MN}/\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



$\overline{RQ}/\overline{GT}_0$
 $\overline{RQ}/\overline{GT}_1$

(**Bus Request/ Bus Grant**) These requests are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle.

These pins are bidirectional.

The request on $\overline{GT}_0$ will have higher priority than $\overline{GT}_1$

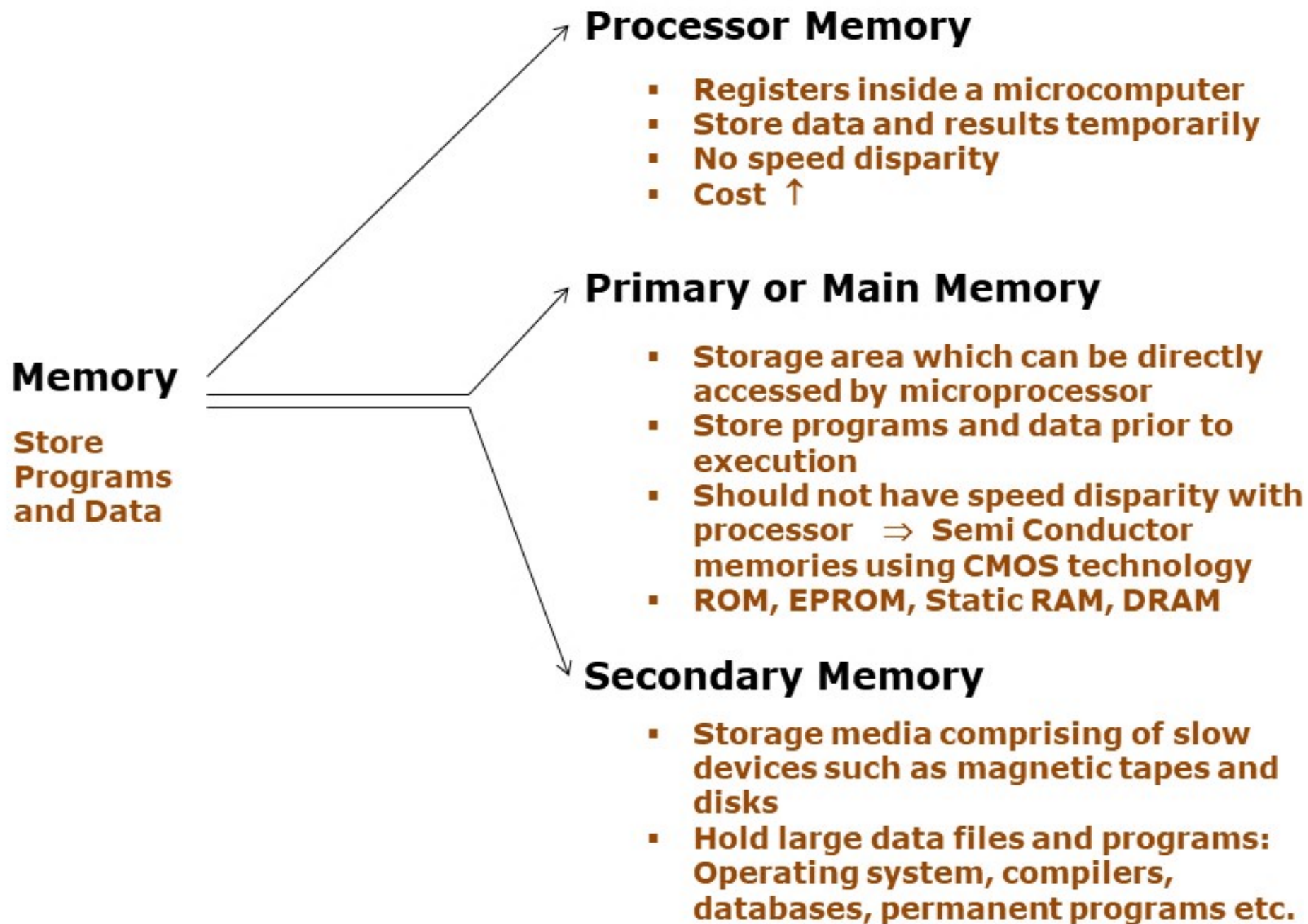
LOCK

An output signal activated by the LOCK prefix instruction.

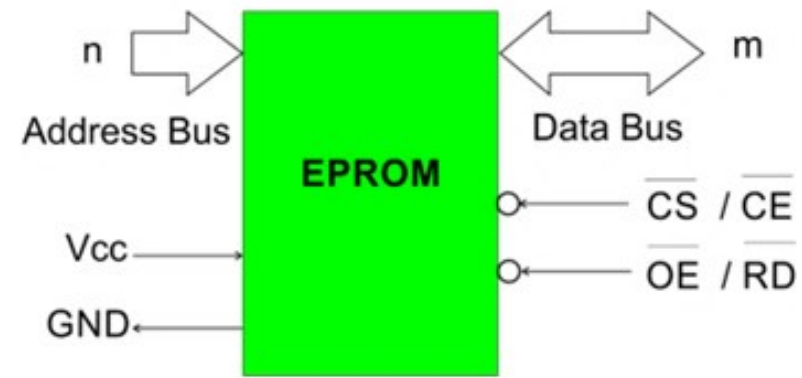
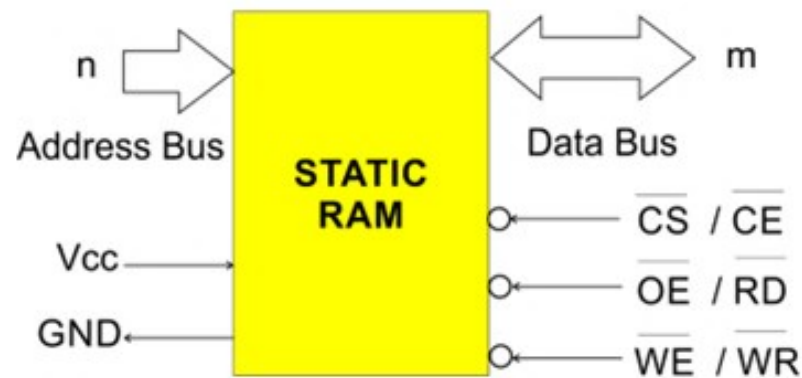
Remains active until the completion of the instruction prefixed by LOCK.

The 8086 output low on the **LOCK** pin while executing an instruction prefixed by LOCK to prevent other bus masters from gaining control of the system bus.

Memory organization

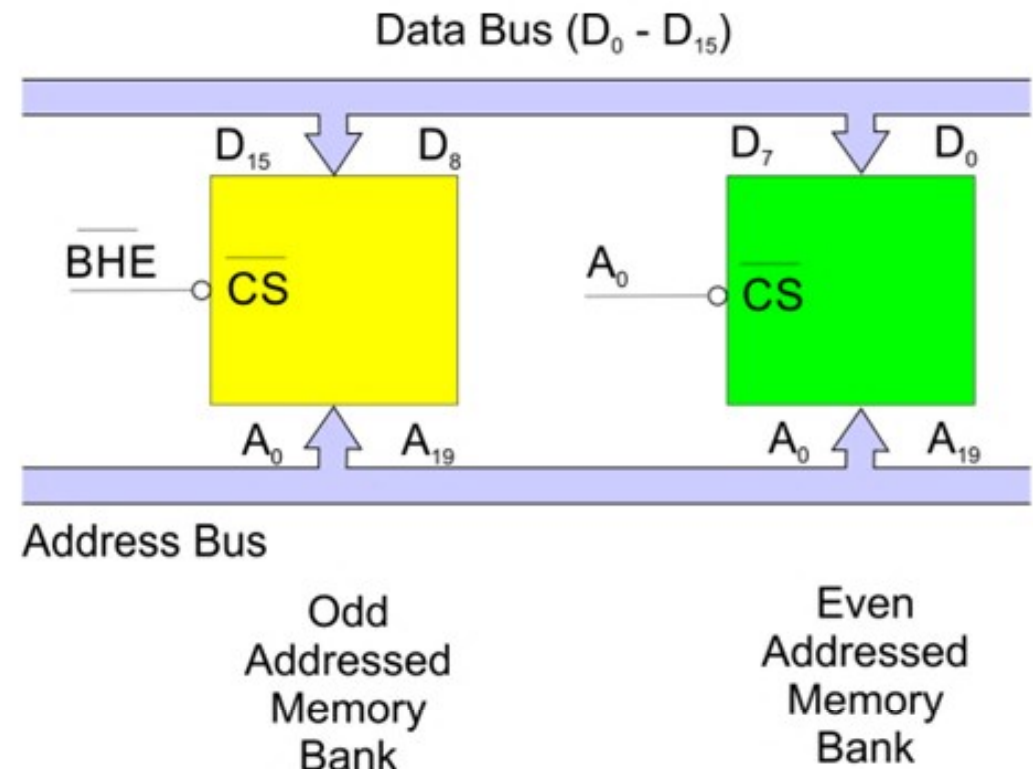


■ Typical Semiconductor IC Chip

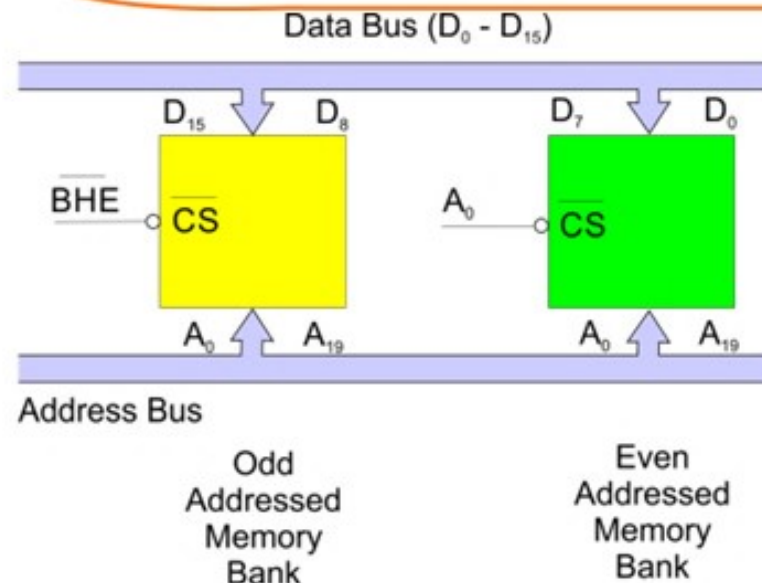


No of Address pins	Memory capacity			Range of address in hexa
	In Decimal	In kilo	In hexa	
20	$2^{20} = 10,48,576$	1024 k = 1M	100000	00000 to FFFFF

- Memory IC's : Byte oriented
- 8086 : 16-bit
- Word : Stored by two consecutive memory locations; for LSB and MSB
- Address of word : Address of LSB
- Bank 0 : $A_0 = 0 \Rightarrow$ Even addressed memory bank
- Bank 1 : $\overline{BHE} = 0 \Rightarrow$ Odd addressed memory bank



Memory organization in 8086



	Operation	$\overline{BHE}$	A_0	Data Lines Used
1	Read/ Write byte at an even address	1	0	$D_7 - D_0$
2	Read/ Write byte at an odd address	0	1	$D_{15} - D_8$
3	Read/ Write word at an even address	0	0	$D_{15} - D_0$
4	Read/ Write word at an odd address	0	1	$D_{15} - D_0$ in first operation byte from odd bank is transferred

Reserved Memory for specific operations

FFFF0H to FFFFFH :

Reserved memory for operation including jump to initialization programme and I/O- processors initialization.

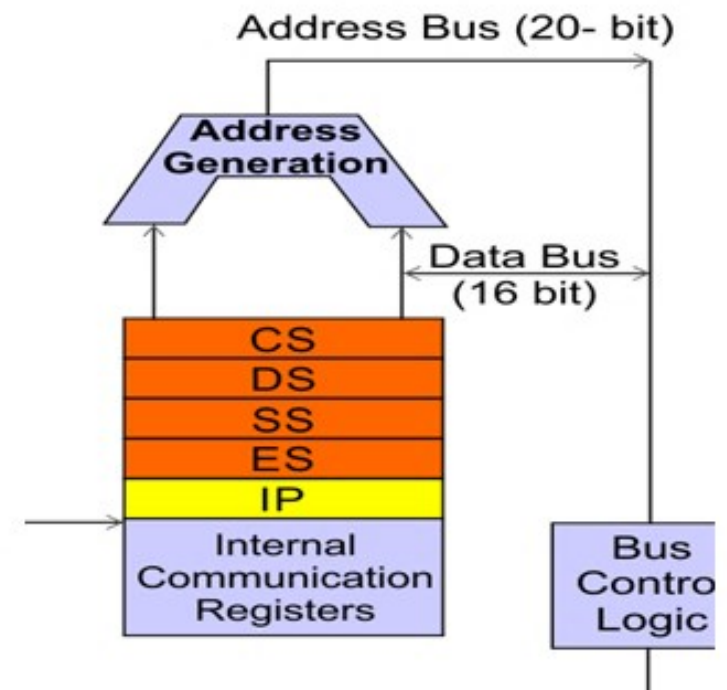
00000H to 003FFH:

Reserved memory for Interrupt Vector Table.

Memory Segmentation in 8086

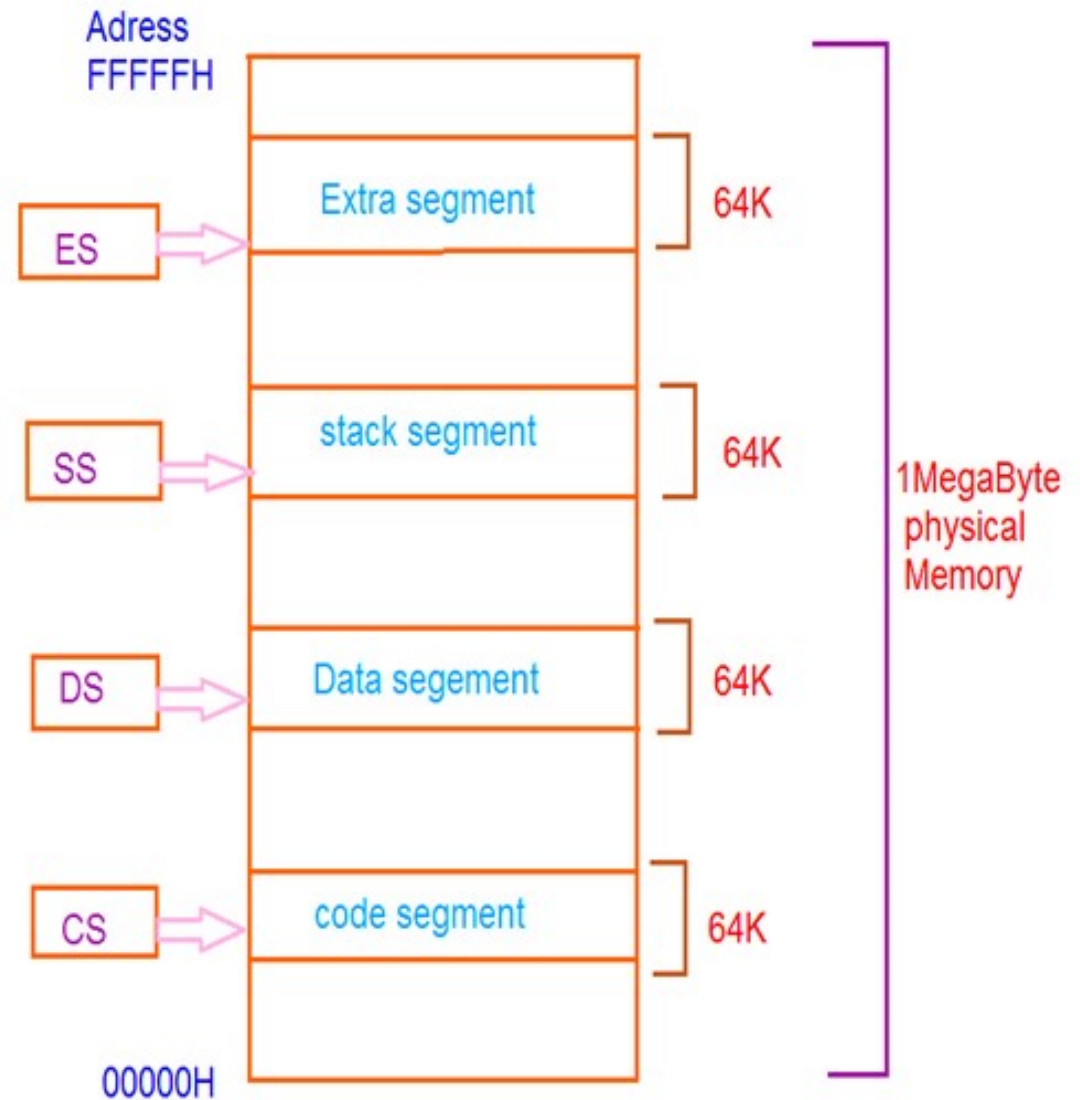
Memory Segmentation

- The number of address lines in 8086 is 20, 8086 BIU will send 20bit address, so as to access one of the 1MB memory locations.
- But the register used in 8086 (segment registers in BIU) are 16- bit registers.
- So, to generate a 20-bit address using 16-bit segment registers the concept of memory segmentation is added in 8086
- 8086 has a segmented memory.
- Complete 1 MB memory can be divided into 16 segments each of 64KBytes.
- The 16 bit content of segment register actually point to starting location of a particular segment.
- To address a specific memory location within a segment we need an offset address. The offset address is also 16 bits long.
- To generate a 20- bit physical address, the content of segment register and offset address are used.



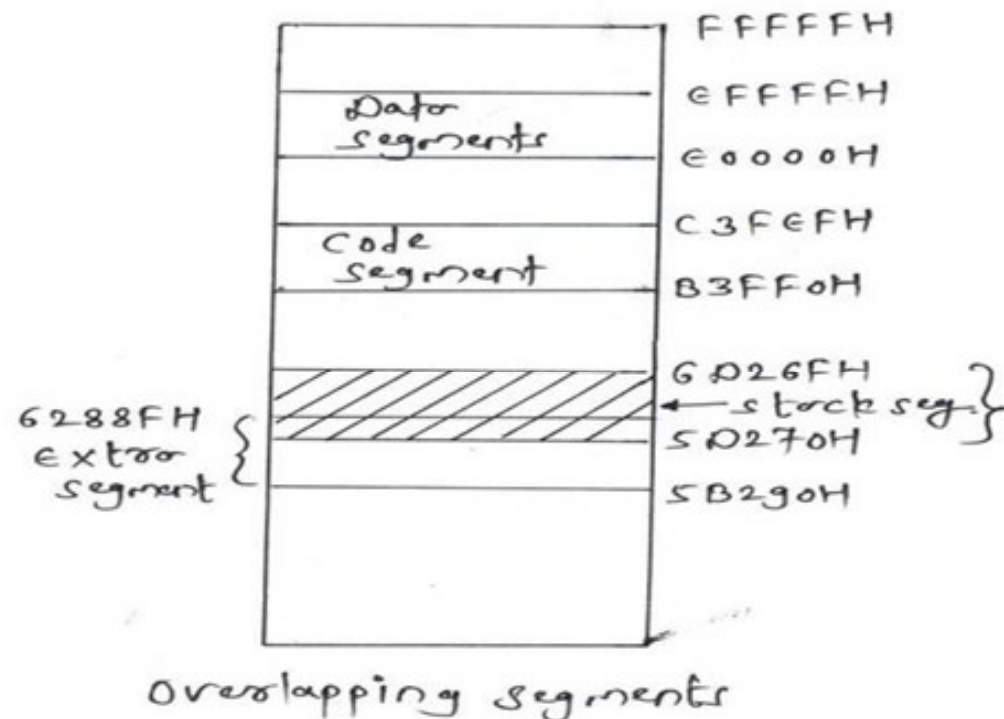
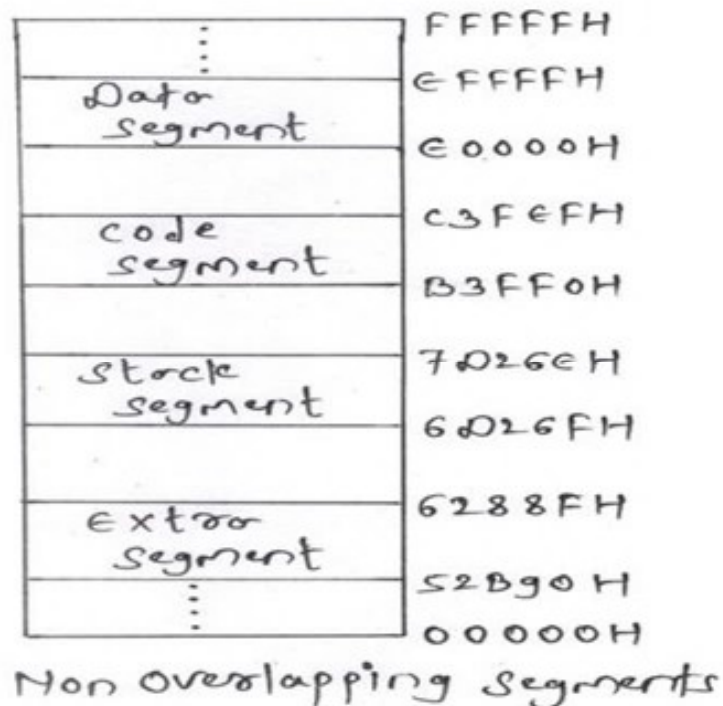
Memory Segmentation

- **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.

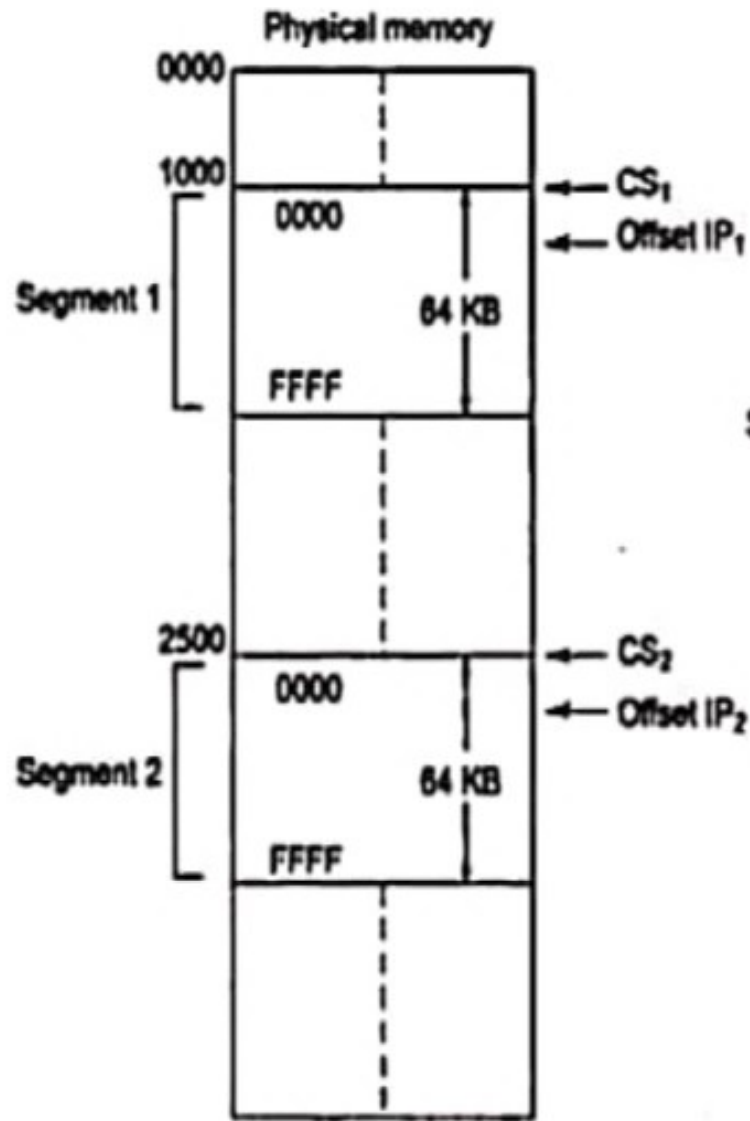


Types Of Segmentation

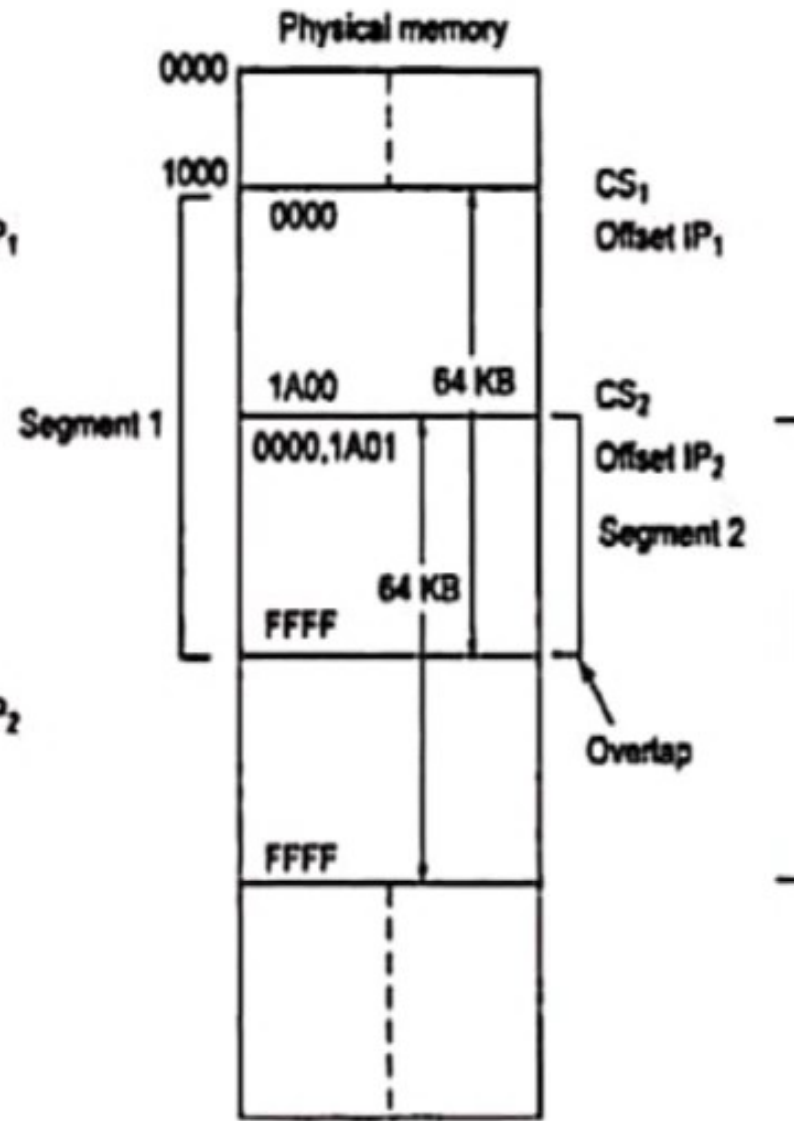
- 1. Overlapping Segment** – A segment starts at a particular address and its maximum size can go up to 64kilobytes. But if another segment starts along with this 64kilobytes location of the first segment, then the two are said to be *Overlapping Segment*.
- 2. Non-Overlapped Segment** – A segment starts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



Types Of Segmentation



Non Overlapping



Overlapping

Physical Address Calculation

4 Segments in 8086

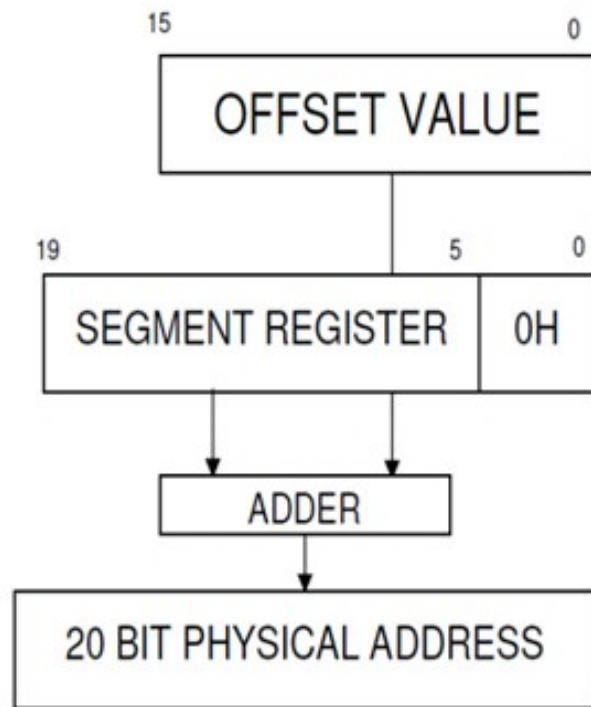
- Code Segment (CS)
- Data Segment (DS)
- Stack Segment (SS)
- Extra Segment (ES)

SEGMENT	SEGMENT REGISTER	OFFSET REGISTER
Code Segment	CSR	Instruction Pointer (IP)
Data Segment	DSR	Source Index (SI)
Extra Segment	ESR	Destination Index (DI)
Stack Segment	SSR	Stack Pointer (SP) / Base Pointer (BP)

Seg Reg	Contents/ Address	Pointer	Contents
ES	7000	DI	Min=0000, max=FFFF
CS	3000	IP	Min=0000, max=FFFF
SS	5000	SP	Min=0000, max=FFFF
DS	2000	SI	Min=0000, max=FFFF

Physical address =
Segment content x 10H
+
Offset address

Physical Address Calculation



Example:

The code segment register holds the segment address which is 4569 H and Instruction pointer IP holds the offset address which is 10A0 H. Calculate a 20-bit physical address.

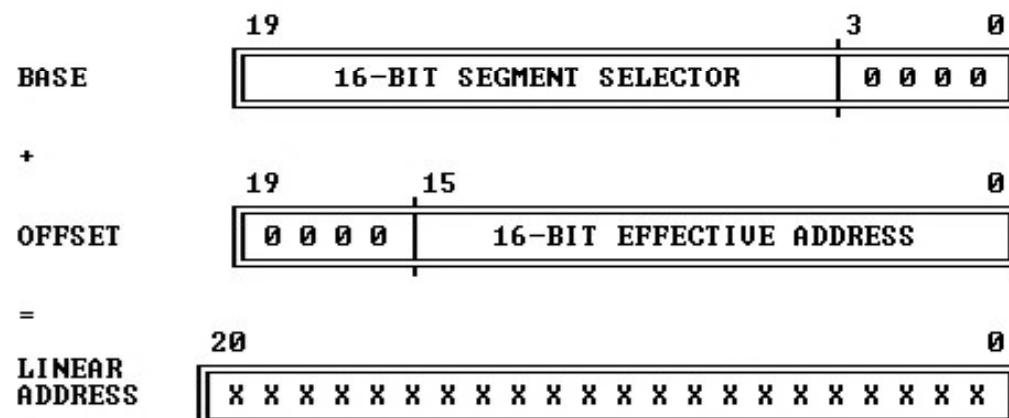
Sol:

$$4569 \text{ H} \times 10 \text{ H} = 45690 \text{ H}$$

$$\text{Physical address} = 45690 \text{ H}$$

$$+ 10A0 \text{ H}$$

$$46730 \text{ H}$$



Example: A memory location has physical address 80FD2h. In what segment does it have offset BFD2h ?

Solution:

$$\text{physical address} = \text{segment} * 10h + \text{offset}$$

$$\Rightarrow \text{segment} * 10h = \text{physical address} - \text{offset}$$

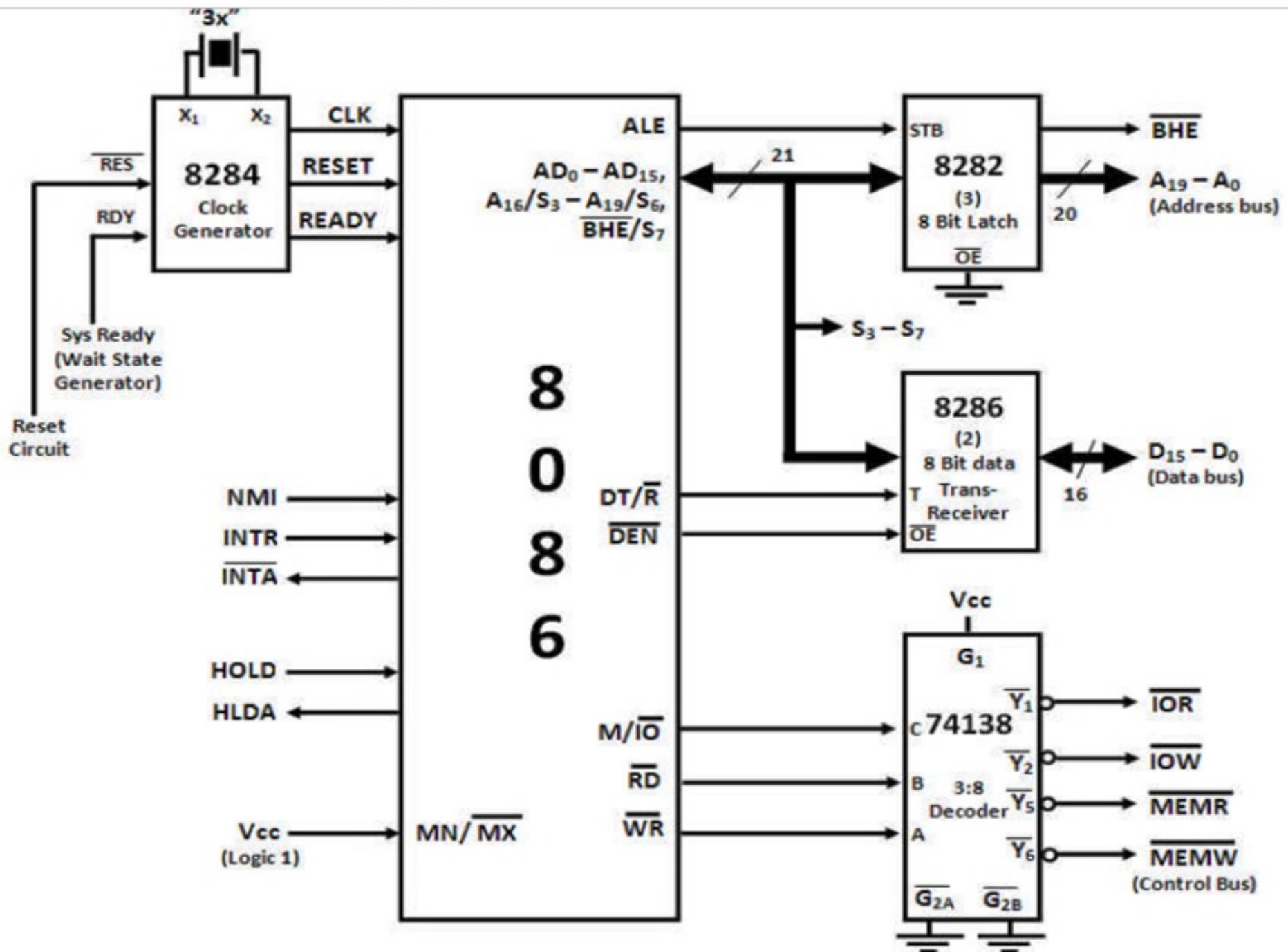
Hence:

$$\begin{array}{r} \text{physical address} = 80FD2h \\ - \quad \text{offset} = \text{BFD2h} \\ \hline \quad \quad \quad 75000h \end{array}$$

Thus the segment is 7500h

Minimum Mode

Minimum Mode



Minimum Mode

- In a minimum mode 8086 system, the microprocessor 8086 is operated in minimum mode by strapping its **MN//MX pin to logic 1**.
- In minimum mode, 8086 is the only processor in the system which provides all the control signals which are needed for memory operations and I/O interfacing.
- Here the circuit is simple but it does not support multiprocessing.
- The other components which are transceivers, latches, 8284 clock generator, 74138 decoder, memory and i/o devices are also present in the system.
- Latches 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**.
- The address bus of 8086 is 20 bits long. By this we can access 220 byte memory i.e. 1MB . Out of 20 bits, 16 bits A0 to A15(or 16 lines) are multiplexed with a data bus. By multiplexing, it means they will act as address lines during the first T state of the machine cycle and in the rest, they act as data lines. A16 to A19 are multiplexed S3 to S6 and BHE' is multiplexed with S7.

Minimum Mode

Control signals provided by 8086 for memory operations and i/o interfacing :

- 8282 (8 bits) latch :

The latches are buffered D FF. They are used to separate the valid address from the multiplexed Address/data bus by using the control signal ALE, which is connected to strobe(STB) of 8282. The ALE is active high signal. Here three such latches are required because the address is 20 bits.

- 8286 (8 bits) transceivers :

They are bidirectional buffers and also known as data amplifiers. They are used to separate the valid data from multiplexed add/data bus. Two such transceivers are needed because the data bus is 16 bits long. 8286 is connected to DT/R' and DEN' signals. They are enabled through the DEN signal .The direction of data on the data bus is controlled by the DT/R' signal. DT/R' is connected to T and DEN' is connected to OE'.

Minimum Mode

Control signals provided by 8086 for memory operations and i/o interfacing :

- 8284 clock generator is used to provide the clock.
- $M/\overline{IO}' = 1$, then I/O transfer is performed over the bus. and when $M/\overline{IO}' = 0$, then I/O operation is performed.
- The signals $\overline{RD}'$ and write $\overline{WR}'$ are used to identify whether a read bus cycle or a write bus cycle is performing. When $\overline{WR}' = 0$, then it indicates that valid output data on the data bus.
- $\overline{RD}'$ indicates that the 8086 is performing a read data or instruction fetch process is occurring. During read operations, one other control signal is also used, which is $\overline{DEN}$ (data enable) and it indicates the external devices when they should put data on the bus.
- Control signals for all operations are generated by decoding $M/\overline{IO}'$, $\overline{RD}'$, $\overline{WR}'$. They are decoded by 74138 3:8 decoder.

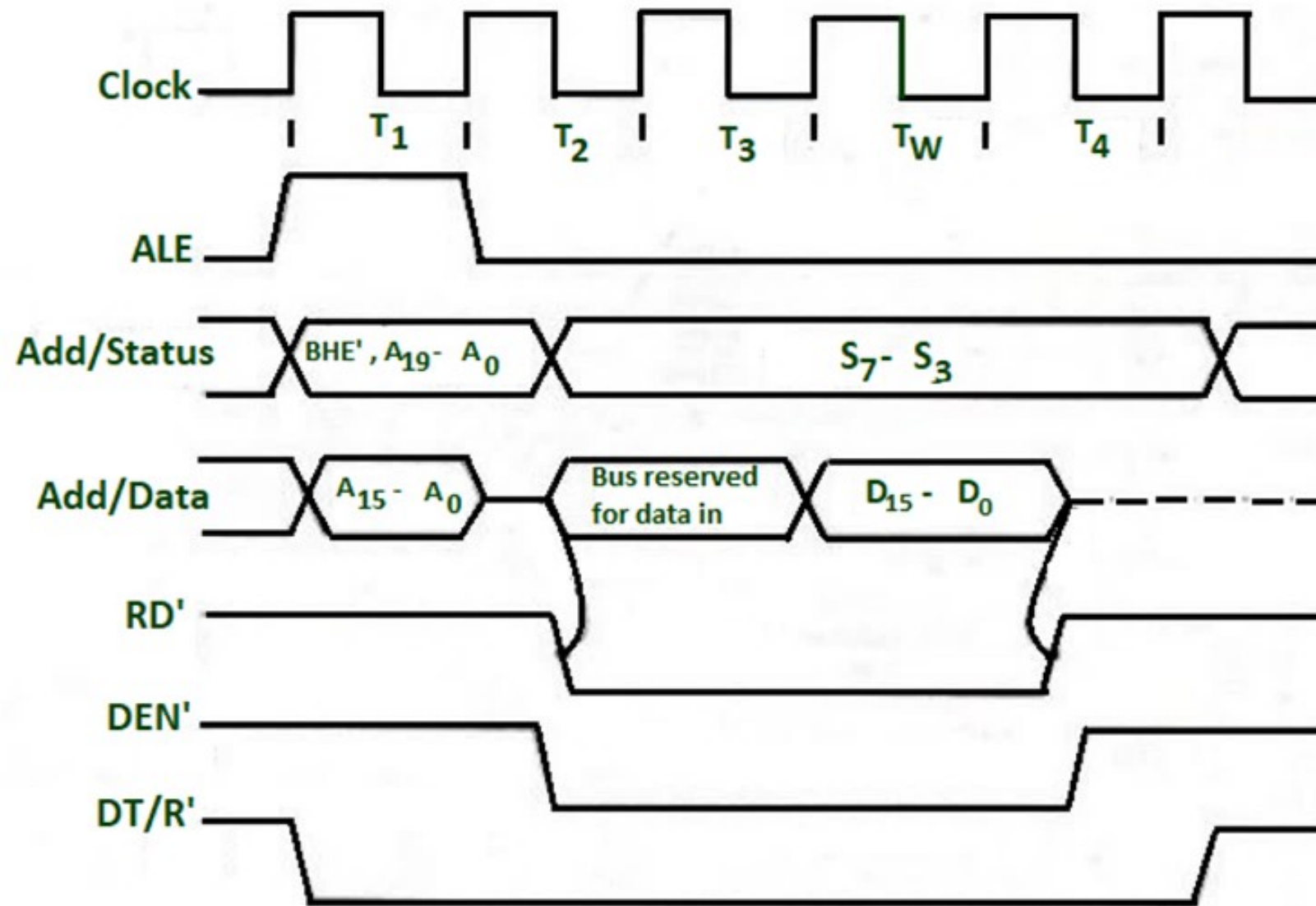
Minimum mode

- Usually, EPROMs are used for monitor storage, while RAM for users program storage. A system may contain I/O devices.
- The opcode fetch and read cycles are similar. Hence the timing diagram can be categorized in two parts, the first is the timing diagram for read cycle and the second is the timing diagram for write cycle.

The working of min mode can be easily understood by timing diagrams.

- All processors bus cycle is of at least 4 T-states (T1,T2,T3,T4) .The address is given by processor in the T1 state. It is available on the bus for one T-state.
- In T2, the bus is tristated for changing the direction of the bus(in the case of a data read cycle.)
- The data transfer takes place between T3 and T4.
- If the addressed device is slower, then the wait state is inserted between T3 and T4.

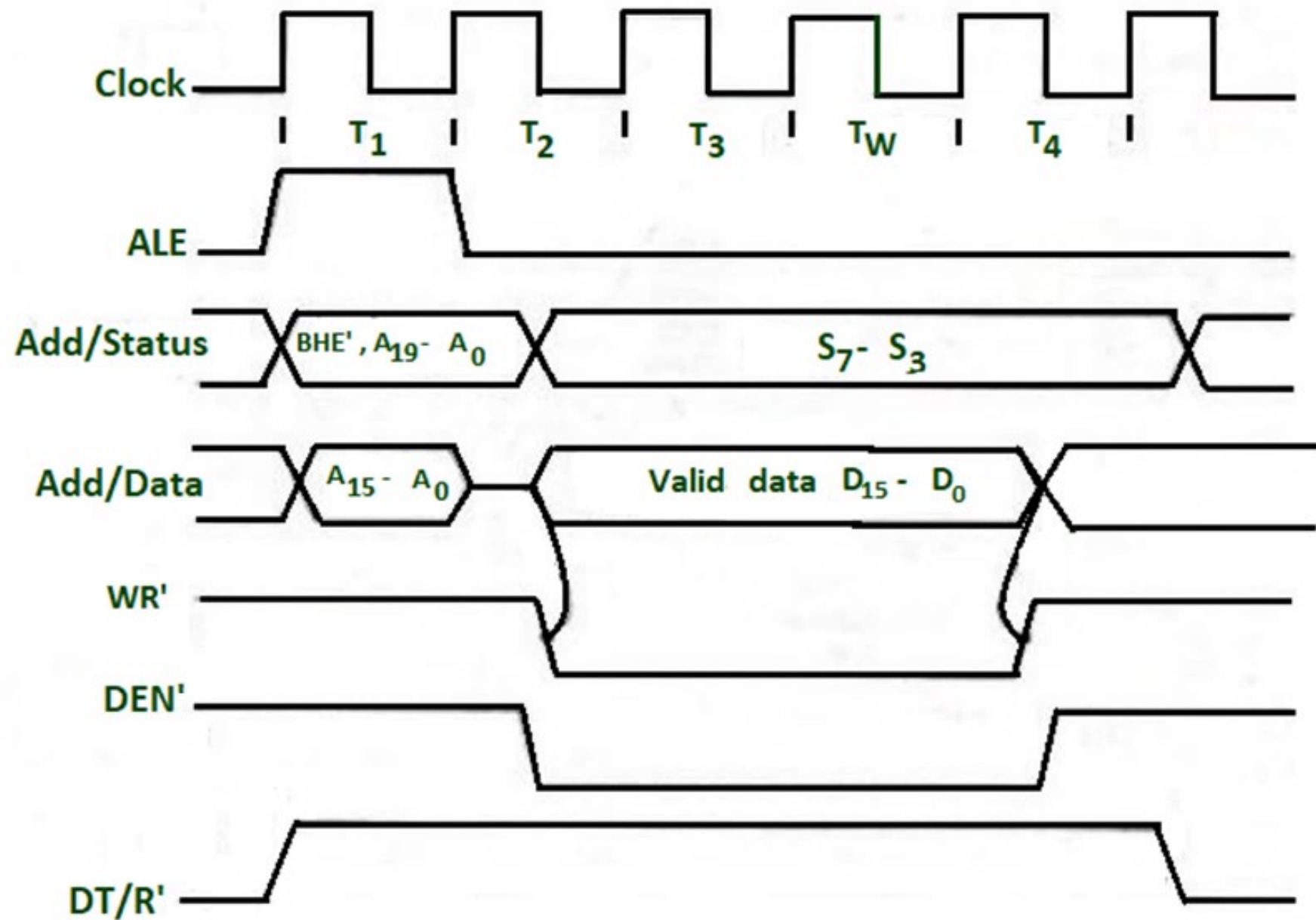
Timing diagram for Read Cycle



Read Cycle in minimum mode

- At T_1 state $ALE = 1$, this indicates that a valid address is latched on the address bus and also $M / IO' = 1$, which indicates the memory operation is in progress.
- The read cycle begins in T_1 with the assertion of address latch enable (ALE) signal and also M / IO signal. During the negative going edge of this signal, the valid address is latched on the local bus.
- The BHE and A0 signals address low, high or both bytes. From T_1 to T_4 , the M/IO signal indicates a memory or I/O operation.
- In T_2 , the address is removed from the local bus and is sent to the addressed device. Then the bus is tristated.
- When $RD' = 0$, the valid data is present on the data bus.
- During T_2 $DEN' = 0$, which enables transceivers and $DT/R' = 0$, which indicates that the data is received.
- During T_3 , data is put on the data bus and the processor reads it.
- The output device makes the **READY line high**. This means the output device has performed the data transfer process. When the processor makes the read signal to 1, then the output device will again tristate its bus drivers.

Timing diagram for Write Cycle

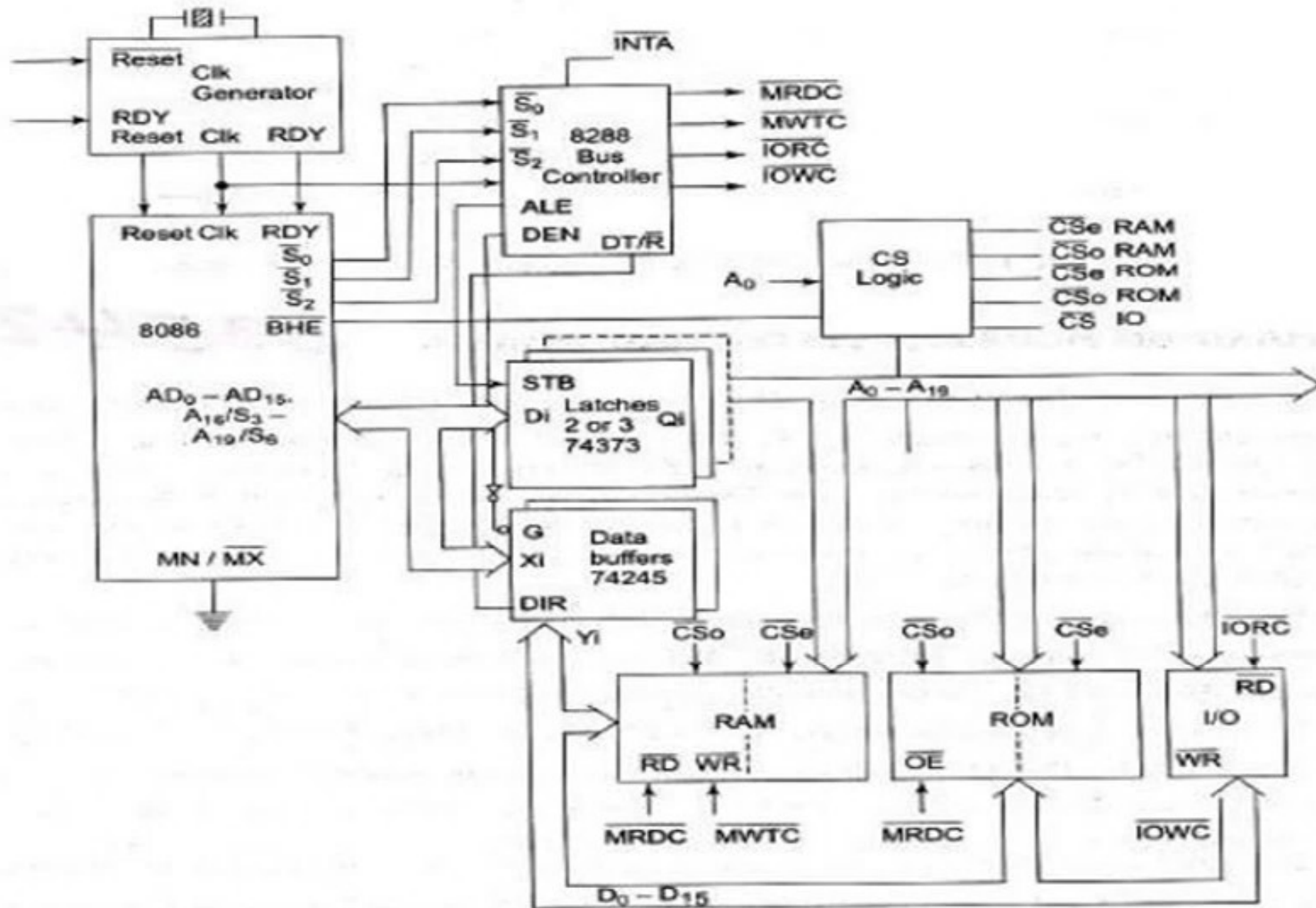


Write Cycle in minimum mode

- At T1 state $ALE = 1$, this indicates that a valid address is latched on the address bus and also $M / IO' = 1$, which indicates the memory operation is in progress.
- The BHE' and A0 signals are used to select the byte or bytes of memory or I/O word.
- In T2, the processor sends the data to be written to the addressed location.
- During T2, $DEN' = 0$, which enables, transceivers and $DT/R' = 1$, which indicates that the data is transferred by the processor to the addressed device.
- The $WR' = 0$ becomes at the beginning of T2.
- The WR becomes active at the beginning of T2 (unlike RD is somewhat delayed in T2 to provide time for floating).
- The data is buffered on the bus until the middle of T4 state.

Maximum Mode

Maximum Mode



Maximum Mode

- In this we can connect more processors to 8086 (8087/8089).
- 8086 max mode is basically for implementation of allocation of global resources and passing bus control to other coprocessor(i.e. second processor in the system), because two processors can not access system bus at same instant.
- All processors execute their own program.
- The resources which are common to all processors are known as global resources.
- The resources which are allocated to a particular processor are known as local or private resources.

Maximum Mode

- When $MN/\overline{MX}' = 0$, 8086 works in max mode.
- Clock is provided by **8284 clock generator**.
- **8288 bus controller**- Address from the address bus is latched into 8282 8-bit latch. Three such latches are required because **address bus is 20 bit**. The ALE(Address latch enable) is connected to STB(Strobe) of the latch. **The ALE for latch is given by 8288 bus controller.**
- The data bus is operated through 8286 8-bit transceiver. Two such transceivers are required, because **data bus is 16-bit**. The transceivers are enabled the DEN signal, while the direction of data is controlled by the DT/R signal. DEN is connected to **OE'** and **DT/ R'** is connected to T. **Both DEN and DT/ R' are given by 8288 bus controller.**
- In this mode, the processor drives the status signal **S'2, S'1 and S'0 using 8288 bus controller.**

Maximum Mode

- The basic functions of the bus controller chip IC8288, is to derive control signals like RD* and WR* (for memory and I/O devices), DEN*, DT/R*, ALE, etc. using the information made available by the processor on the status lines.
- The bus controller chip has input lines S2*, S1* and S0* and CLK. These inputs to 8288 are driven by the CPU.
- It derives the outputs ALE, DEN*, DT/R*, MWTC*, AMWC*, IORC*, IOWC* and AIOWC*.
- The AEN*, IOB and CEN pins are especially useful for multiprocessor systems. AEN* and IOB are generally grounded. CEN pin is usually tied to +5V.
- The significance of the MCE/PDEN* output depends upon the status of the IOB pin.
- If IOB is grounded, it acts as master cascade enable to control cascaded 8259A; else it acts as peripheral data enable used in the multiple bus configurations.

Status pin in maximum mode

S'_2	S'_1	S'_0	Processor State (What the μP wants to do)	8288 Active Output (What Control signal should 8288 generate)
0	0	0	Interrupt Acknowledge	INTA'
0	0	1	Read I/O Port	IORC'
0	1	0	Write I/O Port	IOWC' and AIOWC'
0	1	1	Halt	None
1	0	0	Instruction Fetch	MRDC'
1	0	1	Memory Read	MRDC'
1	1	0	Memory Write	MWTC' and AMWTC'
1	1	1	Inactive	None

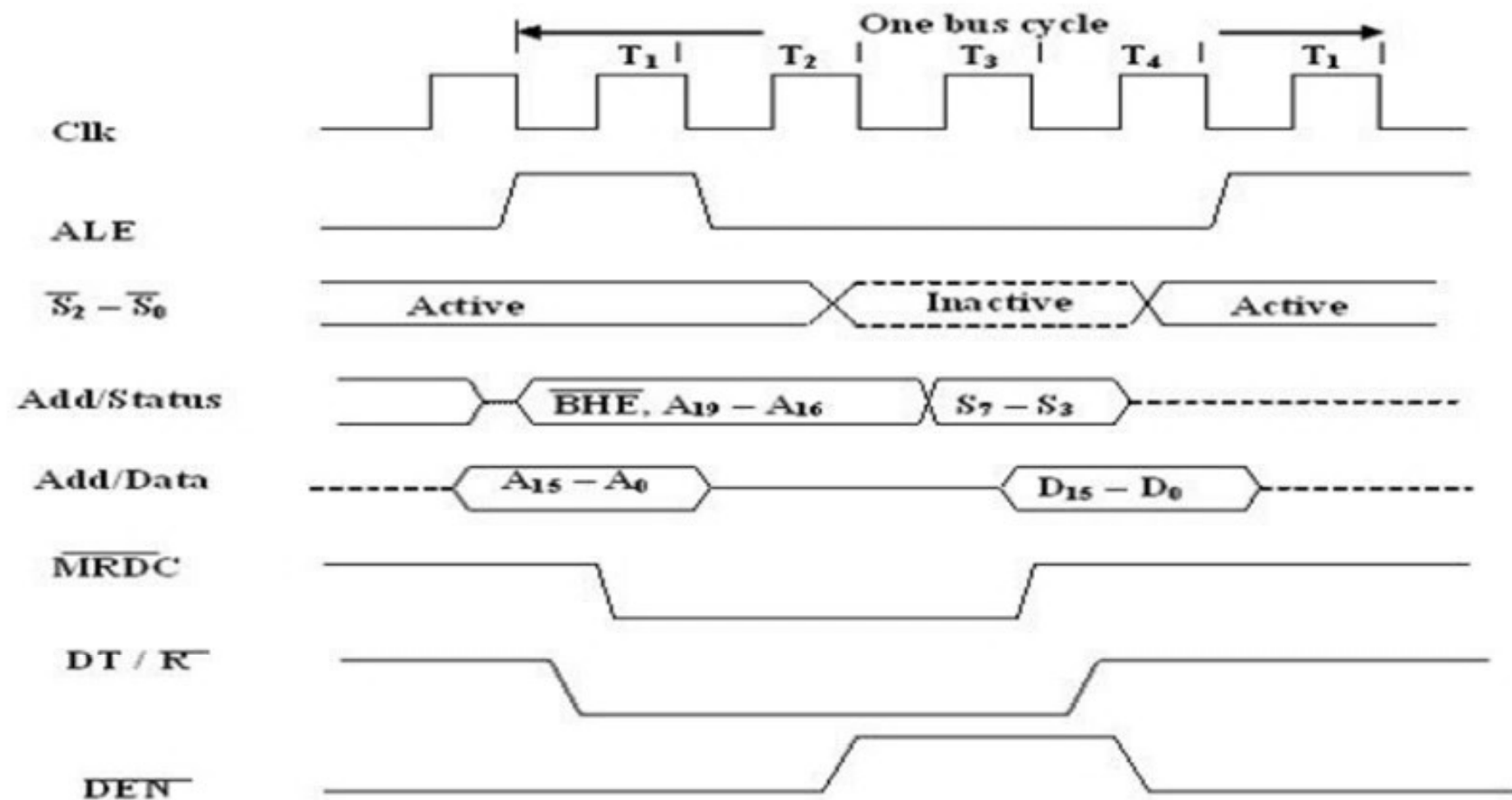
Maximum Mode

- INTA pin used to issue two interrupt acknowledge pulses to the interrupt controller or to an interrupting device.
- IORC, IOWC are I/O read command and I/O write command signals respectively.
- These signals enable an IO interface to read or write the data from or to the address port.
- The MRDC, MWTC are memory read command and memory write command signals respectively and may be used as memory read or write signals.
- All these command signals instructs the memory to accept or send data from or to the bus.
- For both of these write command signals, the advanced signals namely AIOWC and AMWTC are available.
- Here the only difference between in timing diagram between minimum mode and maximum mode is the status signals used and the available control and advanced command signals.

Timing diagrams in maximum mode

- The maximum mode system timing diagrams are also divided in two portions as read (input) and write (output) timing diagrams.
- The address/data and address/status timings are similar to the minimum mode.
- ALE is asserted in T1, just like minimum mode.
- The only difference lies in the status signals used and the available control and advanced command signals.
- S0, S1, S2 are set at the beginning of bus cycle.
- 8288 bus controller will output a pulse as on the ALE and apply a required signal to its DT / R pin during T1.
- In T2, 8288 will set DEN=1 thus enabling transceivers, and for an input it will activate MRDC or IORC. These signals are activated until T4.
- For an output, the AMWC or AIOWC is activated from T2 to T4 and MWTC or IOWC is activated from T3 to T4.
- The status bit S0 to S2 remains active until T3 and become passive during T3 and T4.
- If reader input is not activated before T3, wait state will be inserted between T3 and T4.

Timing diagram for Read Cycle



Memory Read Timing in Maximum Mode

Timing diagram for Write Cycle

