VIGNAN'S
FOUNDATION FOR SCIENCE,
TECHNOLOGY AND RESEARCH
(Deemed to-be University U/s 3 of the UGC Act, 1956)

Micro Processors & Interfacing

16CS307

Unit-2

Mr. M Krishna Chennakesava Rao,
Asst. Professor, Dept. of ECE,
VFSTR University

Micro Processor & Interfacing

Syllabus Overview

UNIT 1 -   Introduction to 8086microprocessor

UNIT 2 - Hardware features of 8086 :

UNIT 3 -   Advanced Processors

UNIT 4 -   Introduction to 8051Microcontroller

UNIT 5 -   8051 Microcontroller Hardware :

Micro Processor & Interfacing

Syllabus Overview

UNIT 1 - Introduction to 8086microprocessor

- **History** of Micro processors
- **Architecture** of 8086 → Register model
- **Memory Segmentation**
- **Software Aspects** of 8086 → Addressing modes

  → Instruction set

  → Interrupts → Hardware Interrupts

  Software Interrupts

UNIT 2 - Hardware features of 8086 :

- Pins
- Max / Min modes
- Memory Interfacing

UNIT - 2

Hardware features of 8086 :

- Pin diagram of 8086,
- Multiplexed ADD/DATA and ADD/STATUS buses, control bus,
- Minimum and maximum modes,
- Memory READ/WRITE and I/O READ/ WRITE machine cycles,
- Machine cycle with WAIT states.
- Physical Memory organization &
- Memory interfacing to 8086.

8086 - Pins and Signals

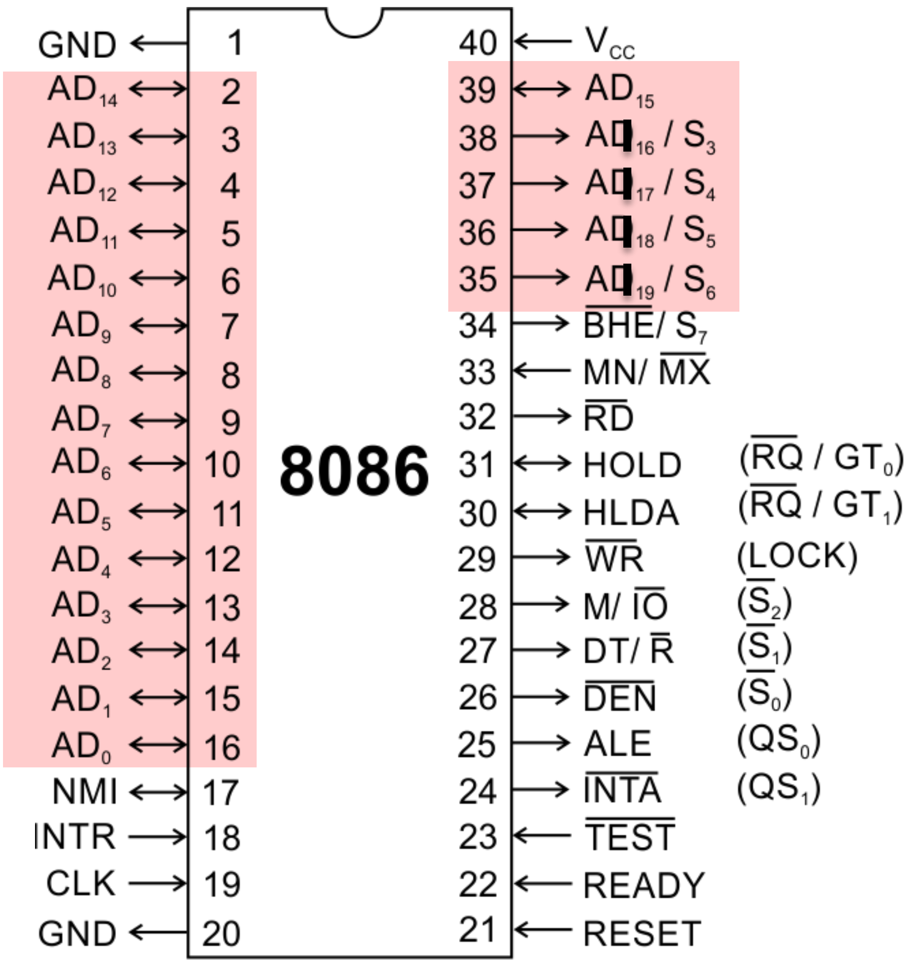
8086 - Pins and Signals

- ❑ 8086 is available in 40 pin DIP.
- ❑ It operates in single processor (**MIN**) mode and

  multi processor (**MAX**) mode configurations.
- ❑ 8086 signals can be categorized into **3** groups.
  - ✓ Signals with **common functions** in MIN & MAX modes
  - ✓ Signals which have special functions for **MIN** mode.
  - ✓ Signals which have special functions for **MAX** mode.

Pins and Signals

Common signals



AD_0-AD_{15} (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 remains on the lines during T1 state. Data is available during T2, T3, TW & T4 CLOCK States.

A_{16}/S_3, A_{17}/S_4, A_{18}/S_5, A_{19}/S_6

High order address bus. These are multiplexed with status signals

Pins and Signals

Common signals

A_{16}/S_3, A_{17}/S_4, A_{18}/S_5, A_{19}/S_6

High order address bus. These are multiplexed with status signals.

The address bits are separated from the status bits using Latches controlled by ALE signal.

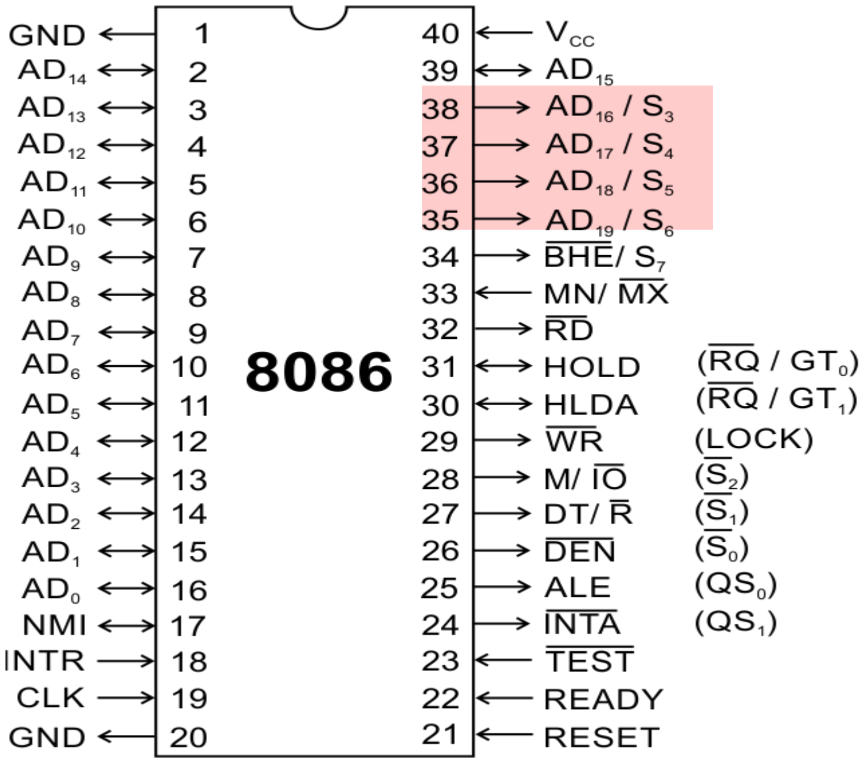
During T_1 , these are the Address lines.

During T_2, T_3, T_W & T_4 CLOCK States status information is available on these lines.

S_6 is always LOW (Logical).

S_5 displays the status of IF (Interrupt enable flag bit), at the beginning of each clock cycle.

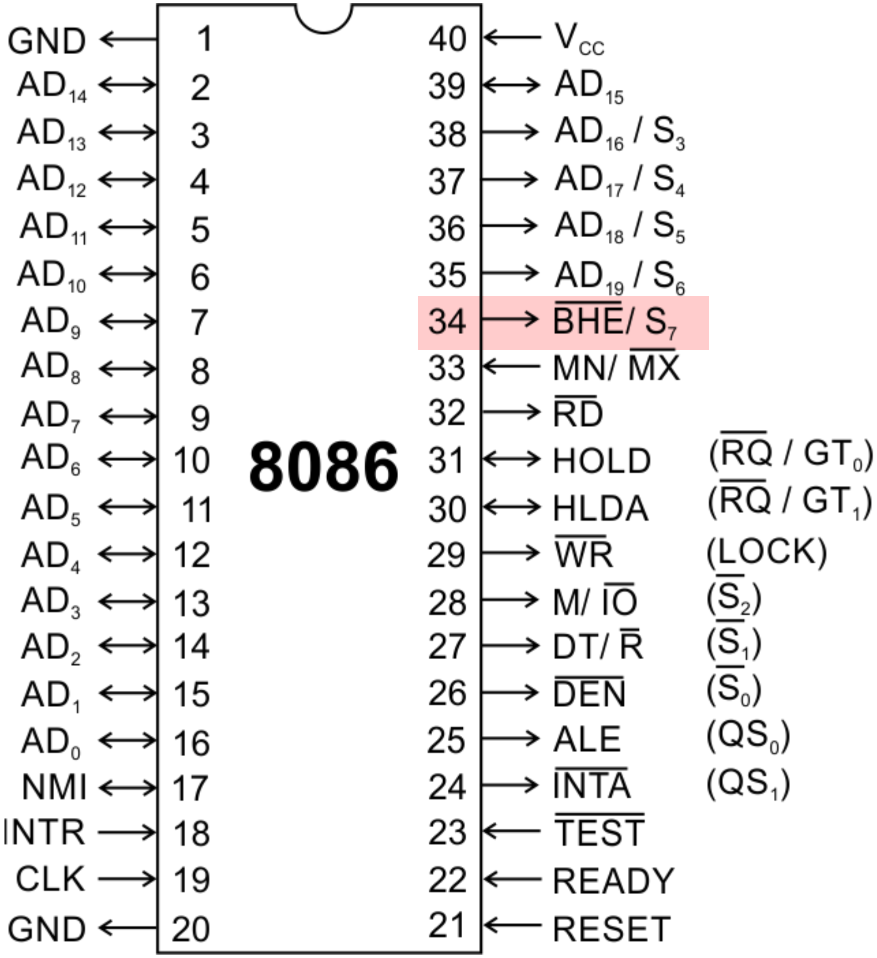
S_4 & S_3 combindely indicates, which segment register is being used presently, for memory access.



S4	S3	Segment Reg Indication
0	0	Extra (ES)
0	1	Stack (SS)
1	0	Code (CS) or None
1	1	Data (DS)

Pins and Signals

Common 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.

Bus High Enable/Status Signal $\overline{BHE} / S_7$ 1024KB [1MB] addressable memory of 8086 is divided into two 512KB even and odd memory banks.

Even memory bank has addresses from 00000H to 0FFFFEH it contains only Even Addresses.

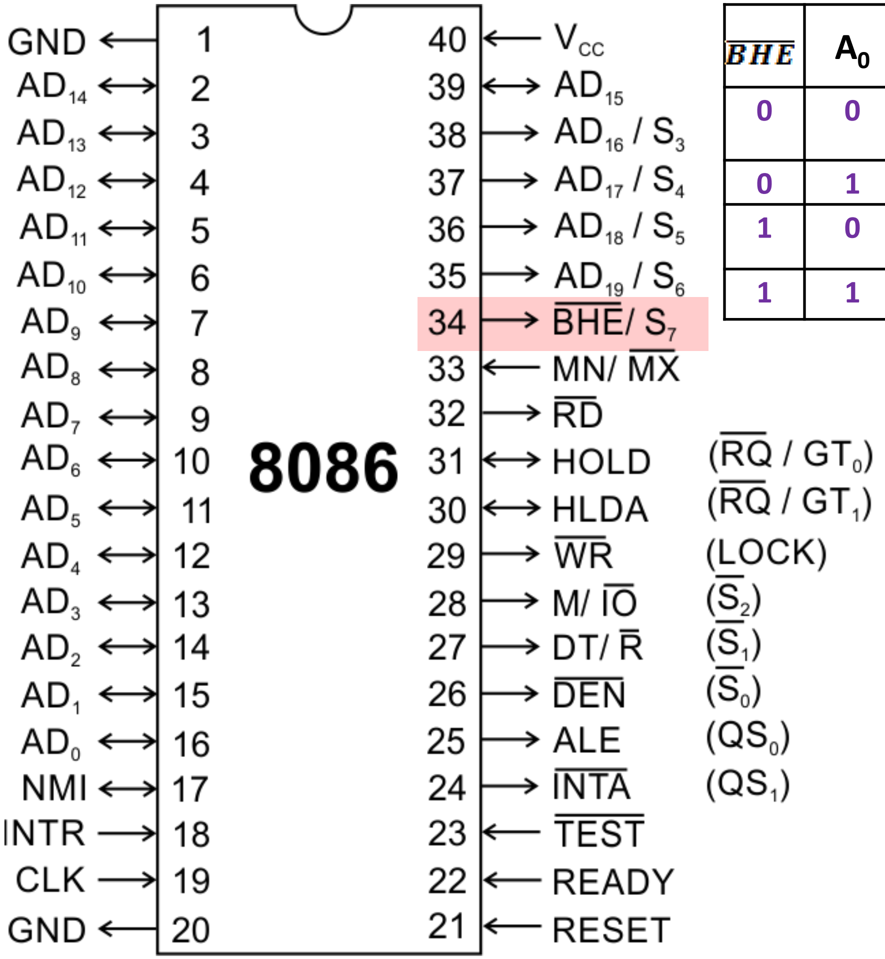
Odd memory bank has addresses from 00001H to 0FFFFFH it contains only Odd Addresses.

BHE is active low signal which is used as enable signal of Odd memory bank. (D_8-D_{15})

The address line A0 is used as Enable signal for the Even Memory Bank. (D_0-D_7)

Pins and Signals

Common signals



BHE (Active Low)/S_7 (Output)

$\overline{BHE}$	A_0	Indication
0	0	Whole word is received/transmitted [D15-D0 data lines are active]
0	1	Byte from Odd memory bank [D15-D8 data lines are active]
1	0	Byte from Even memory bank [D7-D0 data lines are active]
1	1	Passive

MN/ MX*

MINIMUM / MAXIMUM

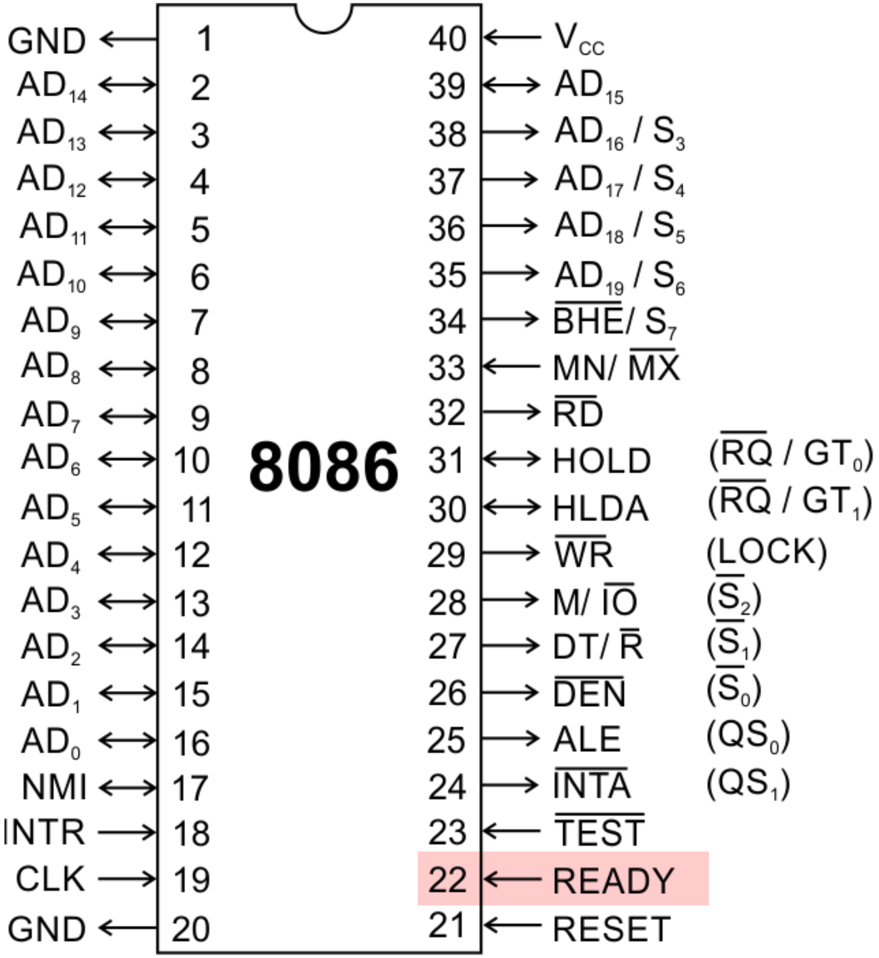
This pin signal indicates what mode the processor is to operate in.

RD* (Read) (Active Low)

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

Pins and Signals

Common signals



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 level 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.

READY :

- It is the acknowledgement from the addressed memory or I/O device .
- If **READY=0** then the processor will insert **WAIT** states between T_3 and T_4.
- If **READY=1**, the external device is ready to communicate with the processor.

TEST*

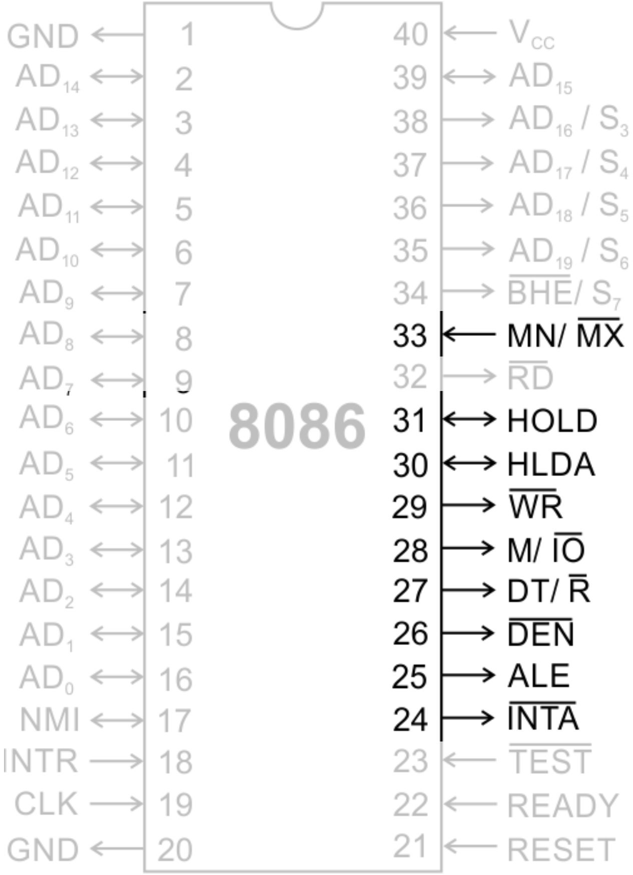
It is the input signal which is examined by the "wait" instruction.

If **TEST*** is **LOW**, **execution continues**, otherwise **the processor waits in an "idle" state.**

Pins and Signals

Min/ Max Pins

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.

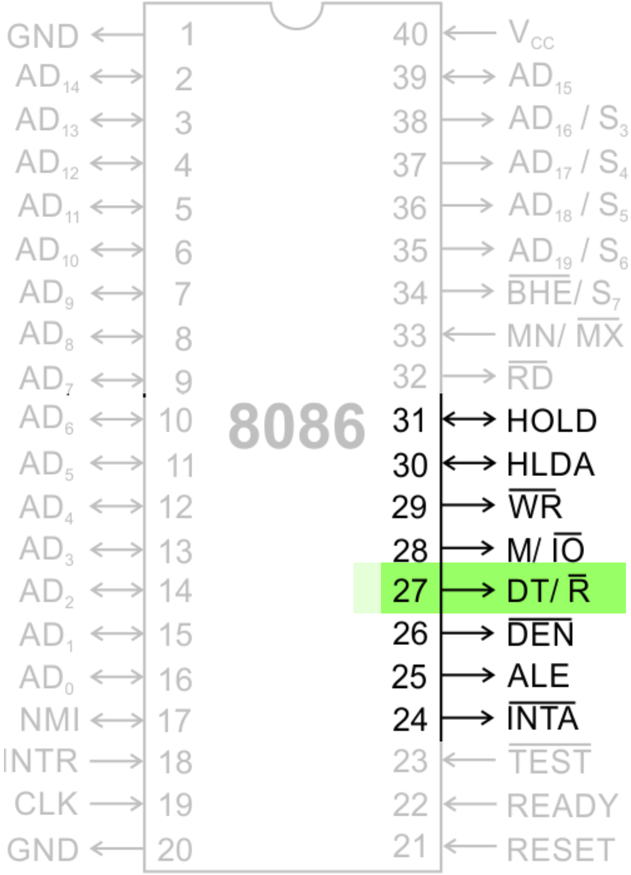
Pins and Signals

Minimum mode signals

Pins 24 -31

For minimum mode operation, the 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 to enable the transceivers. It indicates the availability of data over address / data lines.**

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.**

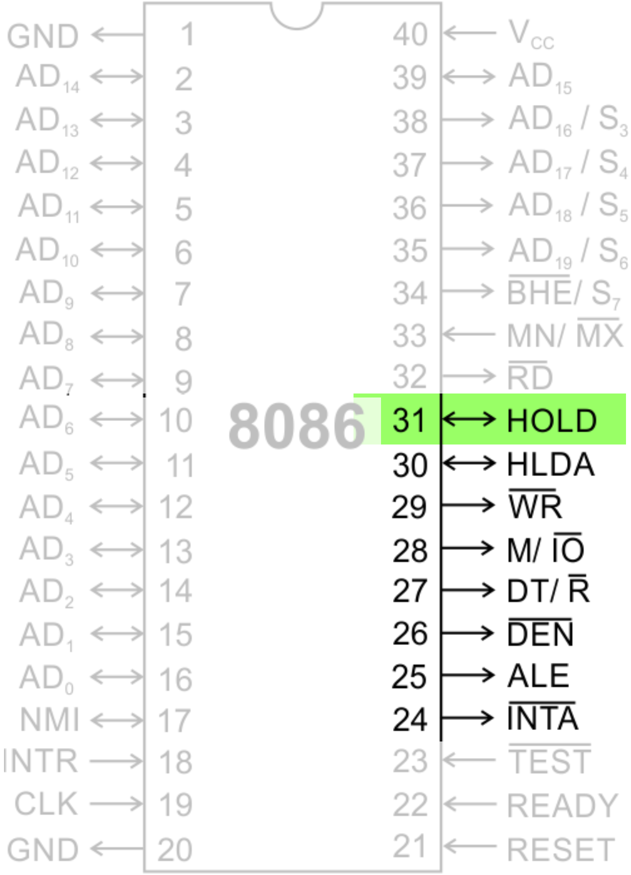
Pins and Signals

Minimum mode signals

Pins 24 -31

For minimum mode operation, the 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.**

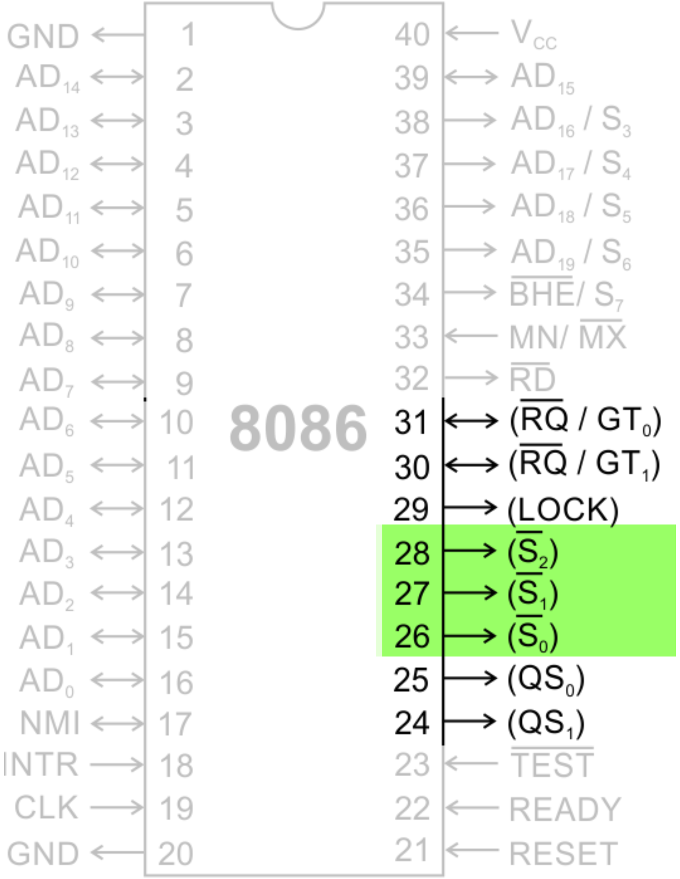
The acknowledge is asserted high, when the processor accepts HOLD.

Pins and Signals

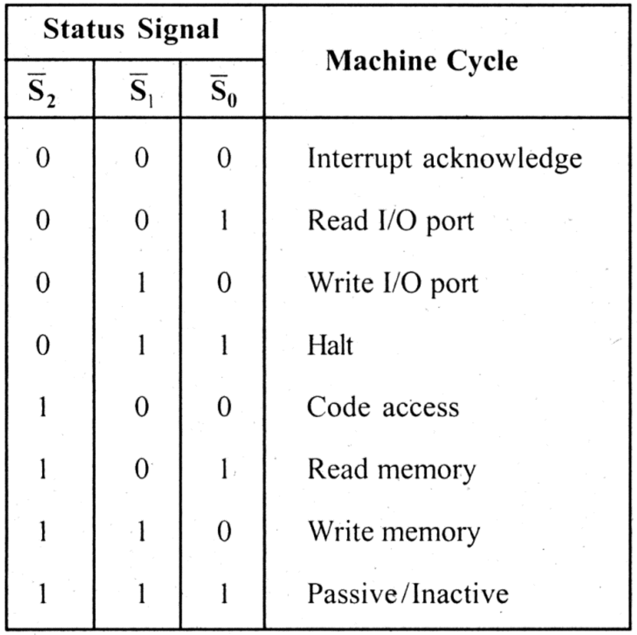
Maximum mode signals

During maximum mode operation, the 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.



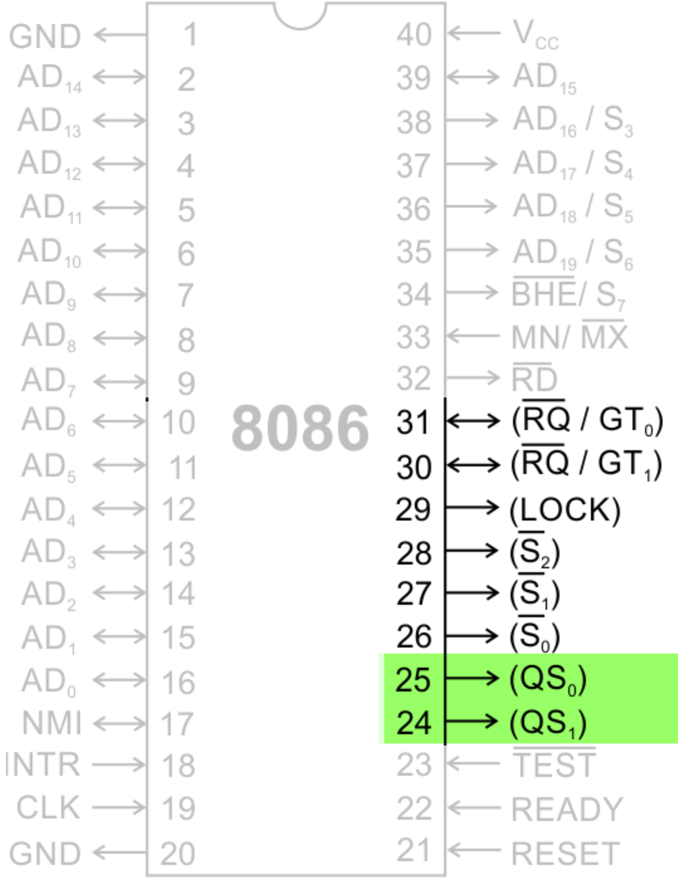
Status Signal			Machine Cycle
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive/Inactive

Pins and Signals

Maximum mode signals

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

Pins 24 -31 are reassigned

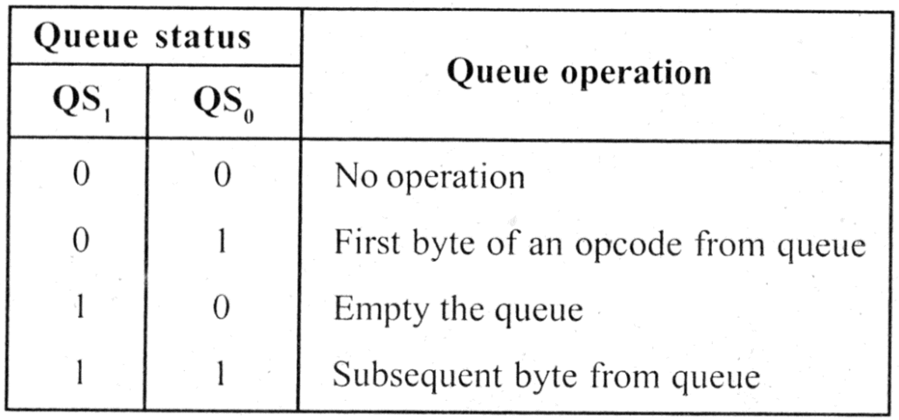


$\overline{QS_0}$, $\overline{QS_1}$

(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.



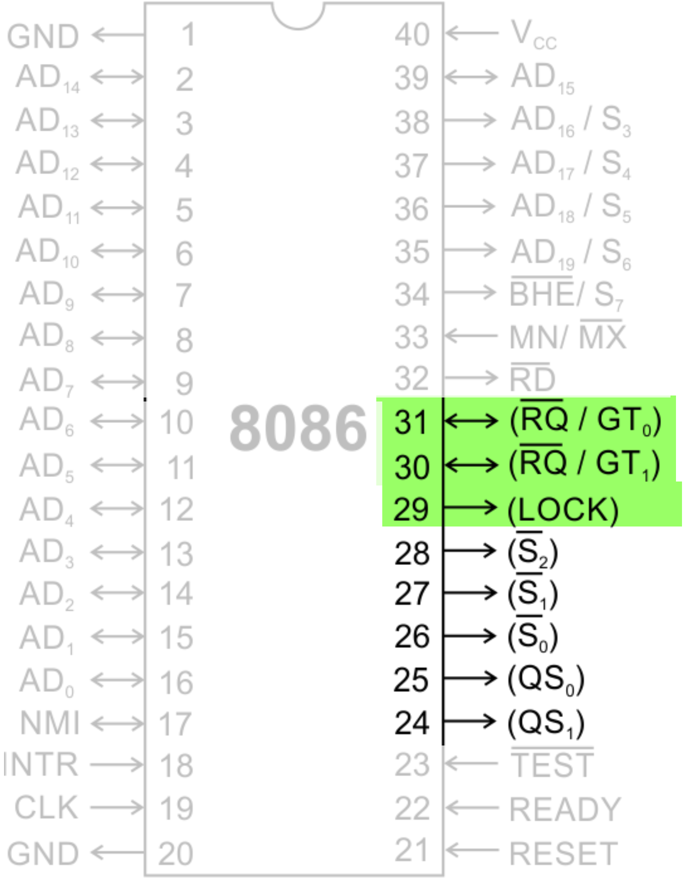
Queue status		Queue operation
QS_1	QS_0	
0	0	No operation
0	1	First byte of an opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

Pins and Signals

Maximum mode signals

During maximum mode operation, the 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 $\overline{GT_1}$

$\overline{LOCK}$

An output signal activated by the LOCK prefix instruction.

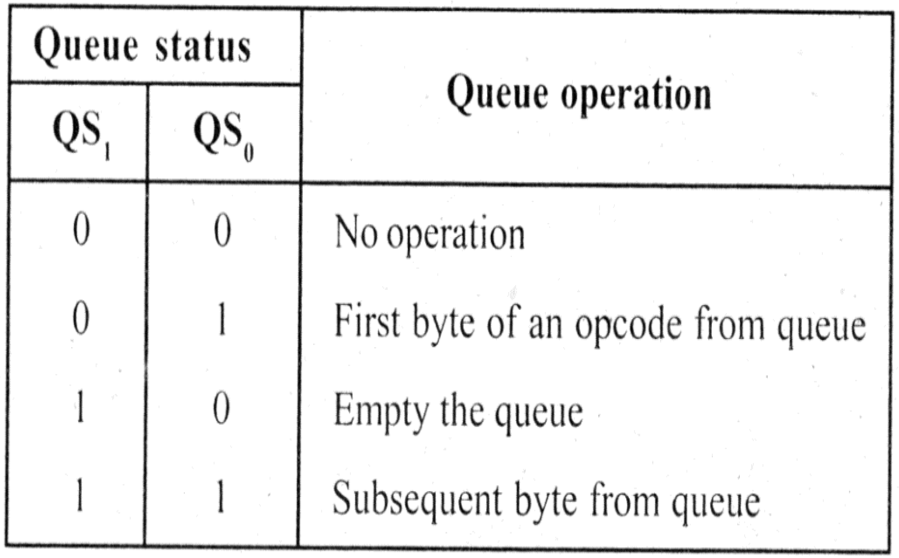
Remains active until the completion of the instruction prefixed by LOCK.

The 8086 output low on the $\overline{LOCK}$ pin while executing an instruction prefixed by LOCK to <u>prevent other bus masters from gaining control of the system bus.</u>

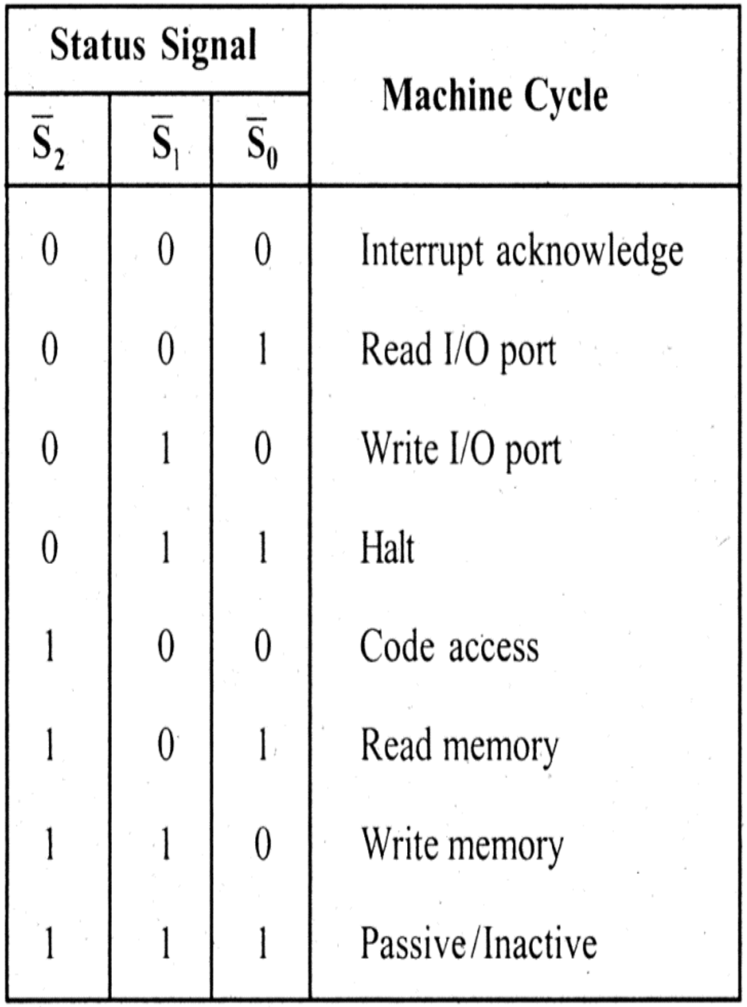
$\overline{BHE}$	A_0	Indication
0	0	Whole word is received/transmitted [D15-D0 data lines are active]
0	1	Byte from Odd memory bank [D15-D8 data lines are active]
1	0	Byte from Even memory bank [D7-D0 data lines are active]
1	1	Passive

M/ $\overline{IO}$	$\overline{RD}$	$\overline{WR}$	Transfer Type
0	0	1	I/O read
0	1	0	I/O write
1	0	1	Memory read
1	1	0	Memory write

S4	S3	Segment Reg Indication
0	0	Extra (ES)
0	1	Stack (SS)
1	0	Code (CS) or None
1	1	Data (DS)



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

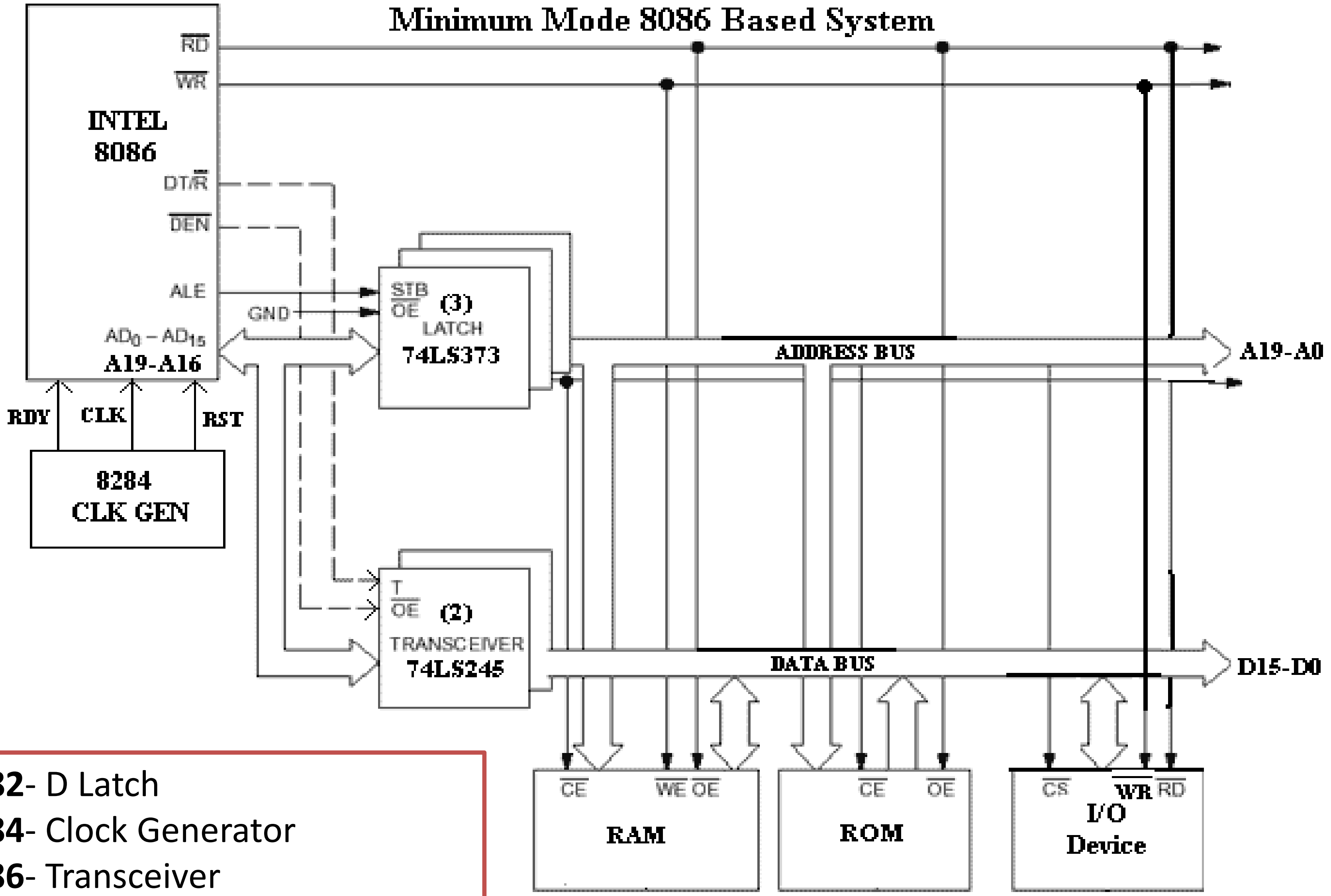


Status Signal			Machine Cycle
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive/Inactive

TRY IT…

1) What are Minimum mode pins?
2) What are maximum mode pins?
3) What is INTR?
4) What is the importance of ALE?
5) Why two ground pins for 8086?
6) Why address and data pins are multiplexed?
7) How many address pins in total?
8) List out all active low signals of 8086.

Minimum Mode 8086 System



8282- D Latch
8284- Clock Generator
8286- Transceiver
8288- Bus Controller (Max mode)

Minimum Mode 8086 System

- Turns MN/MX* pin to logic 1
- Control signals are given by microprocessor chip itself
- Single microprocessor with **latches**, **transrecievers**, **clock generator**, **memory** & IO devices.

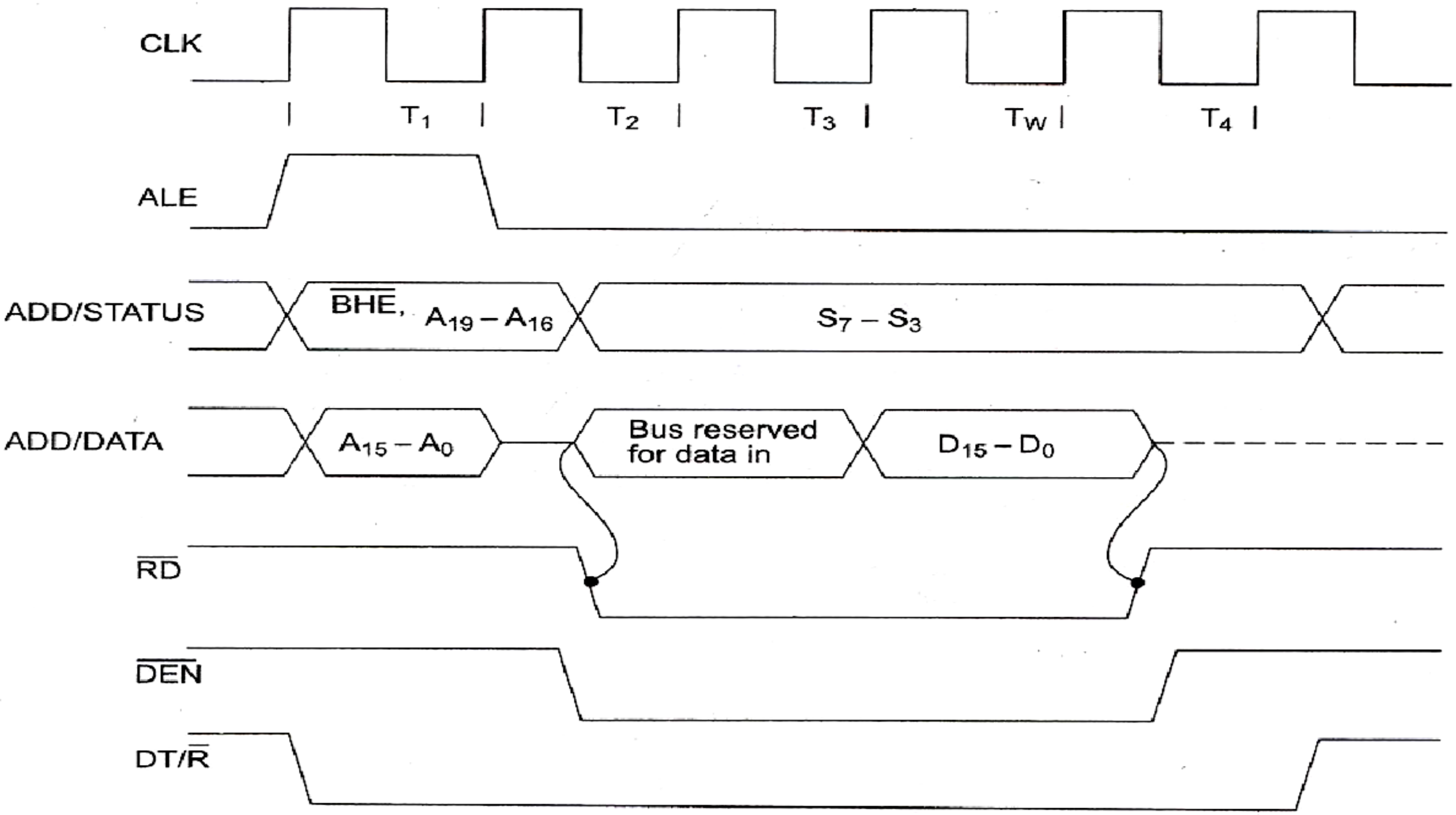
- **Latches** are used **to separate valid address** from address/data signals controlled by ALE.
- The **latches** are generally buffered output **D-type flip-flops**, like, **74LS373 or 8282**.

- **Transrecievers** are bidirectional buffers , also termed as **data amplifiers.**
- **Transrecievers** are used **to separate the valid data** from multiplexed address/data signal.
- Controlled by DEN* or DT/R*
- The **DEN*** signal indicates that **the valid data is available** on the data bus.
- **DT/R*** indicates the **direction of data**, i.e. from or to the processor.

- The system contains **memory** , for the **monitor** and **users program storage**.
- **EPROMS** are used **for monitor storage**
- **RAMs** for **users program** storage.

- The clock generator (**8284**) synchronizes some external signals with the system clock.

Read Cycle Timing Diagram for Minimum Mode

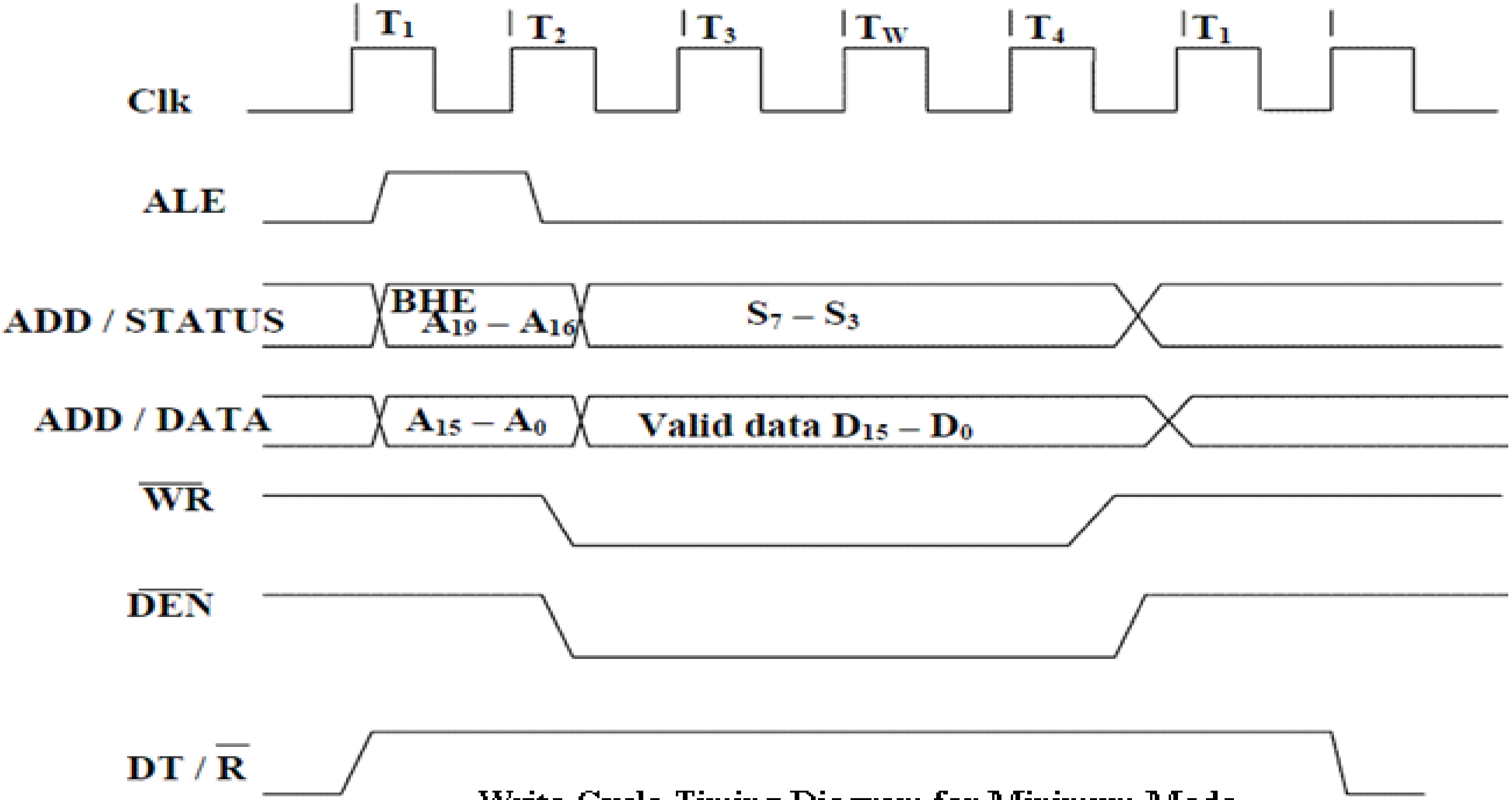


Read Cycle Timing Diagram for Minimum Mode

- The read cycle begins in T_1 with the assertion of the address latch enable (ALE) signal
- During the negative going edge of this signal, the valid address is latched on the local bus.
- The BHE* and A_0 signals address low, high or both bytes.
- From T_l to T_4, 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 tristated.
- The read $\overline{RD}$ control signal is also activated in T_2 .
- The read $\overline{RD}$ signal causes the addressed device to enable its data bus drivers.
- After 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 tristate its bus drivers.

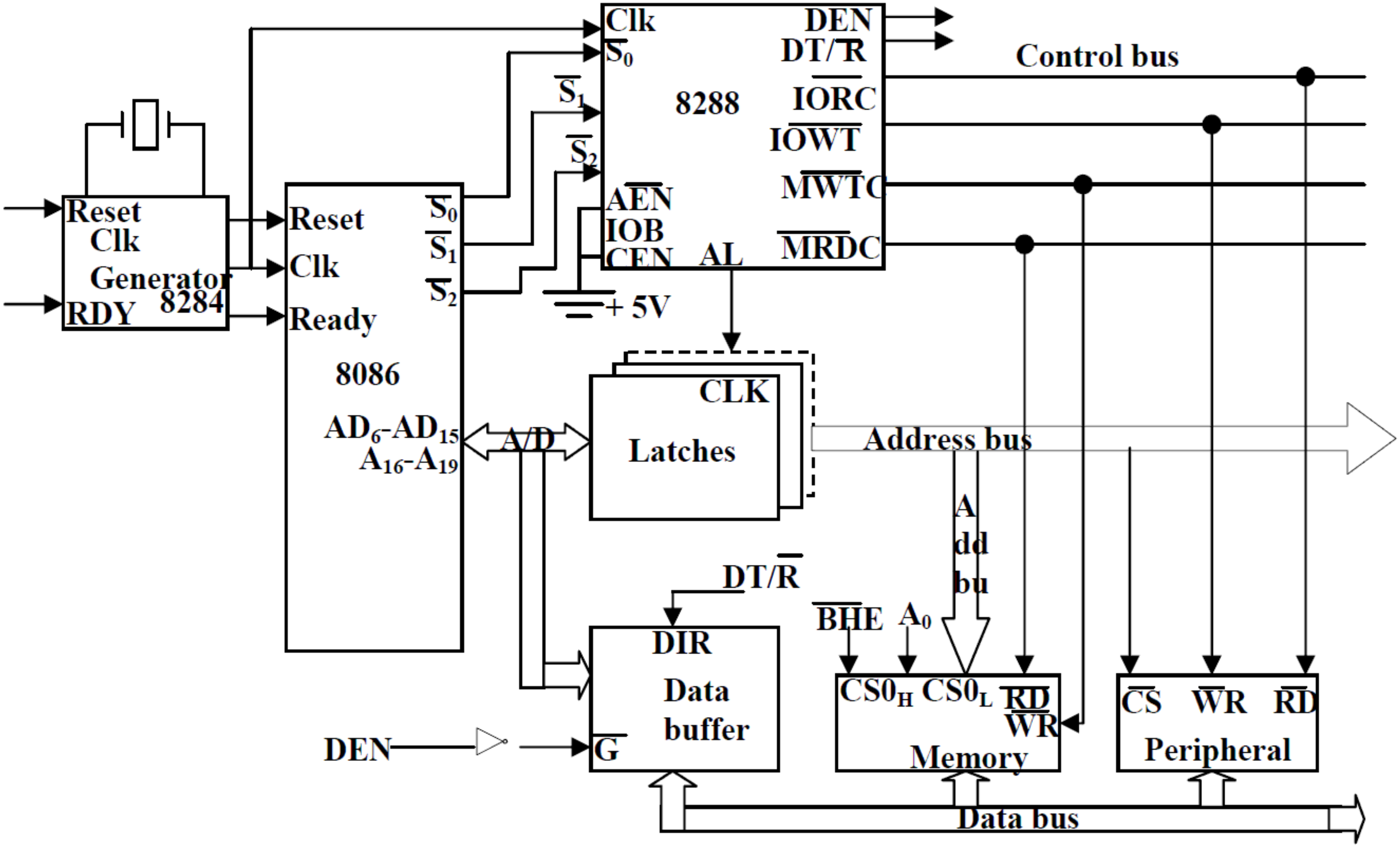
Read Timing Diagram

Write Cycle Timing Diagram for Minimum Mode



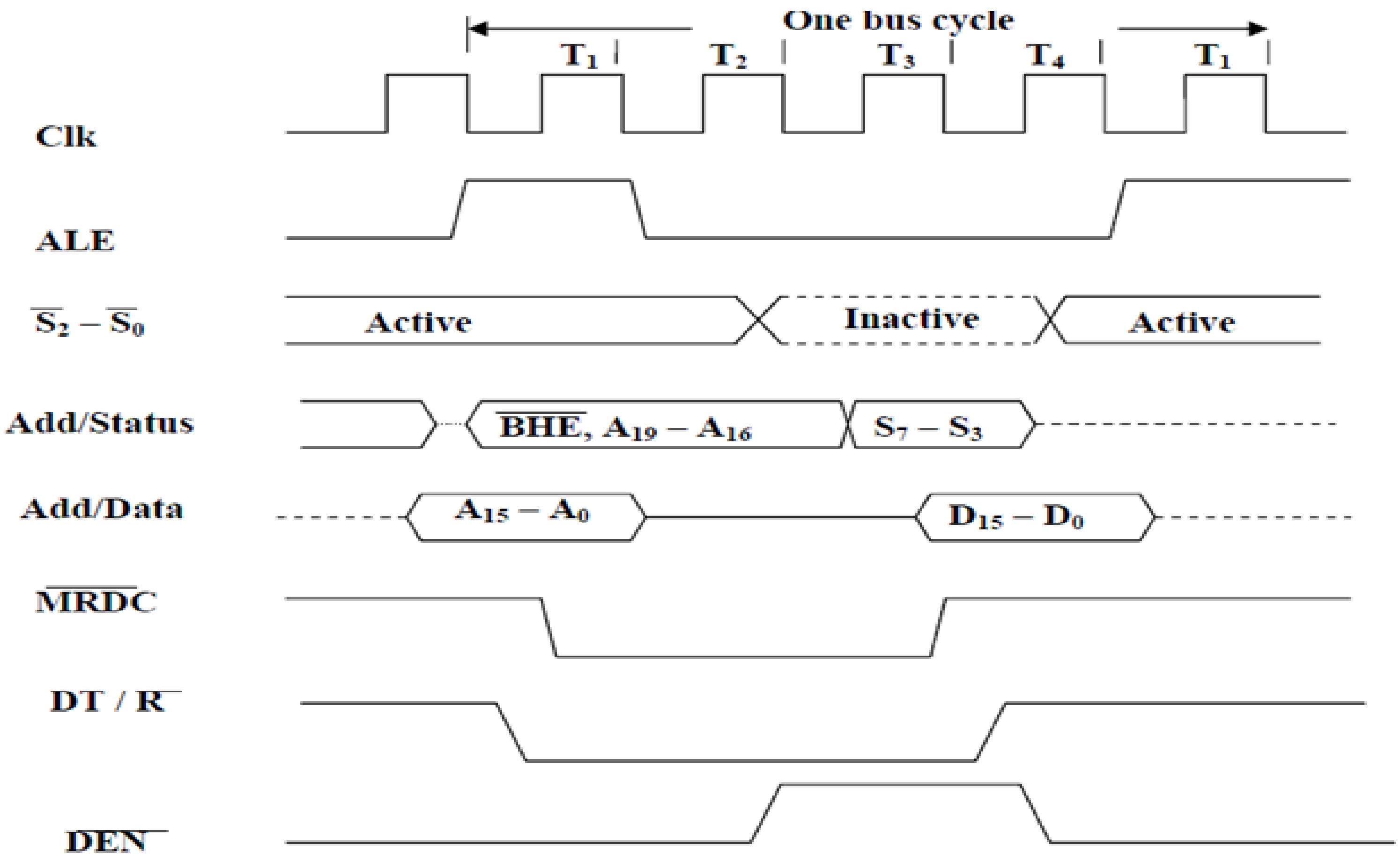
Write Cycle Timing Diagram for Minimum Mode

Maximum Mode 8086 system



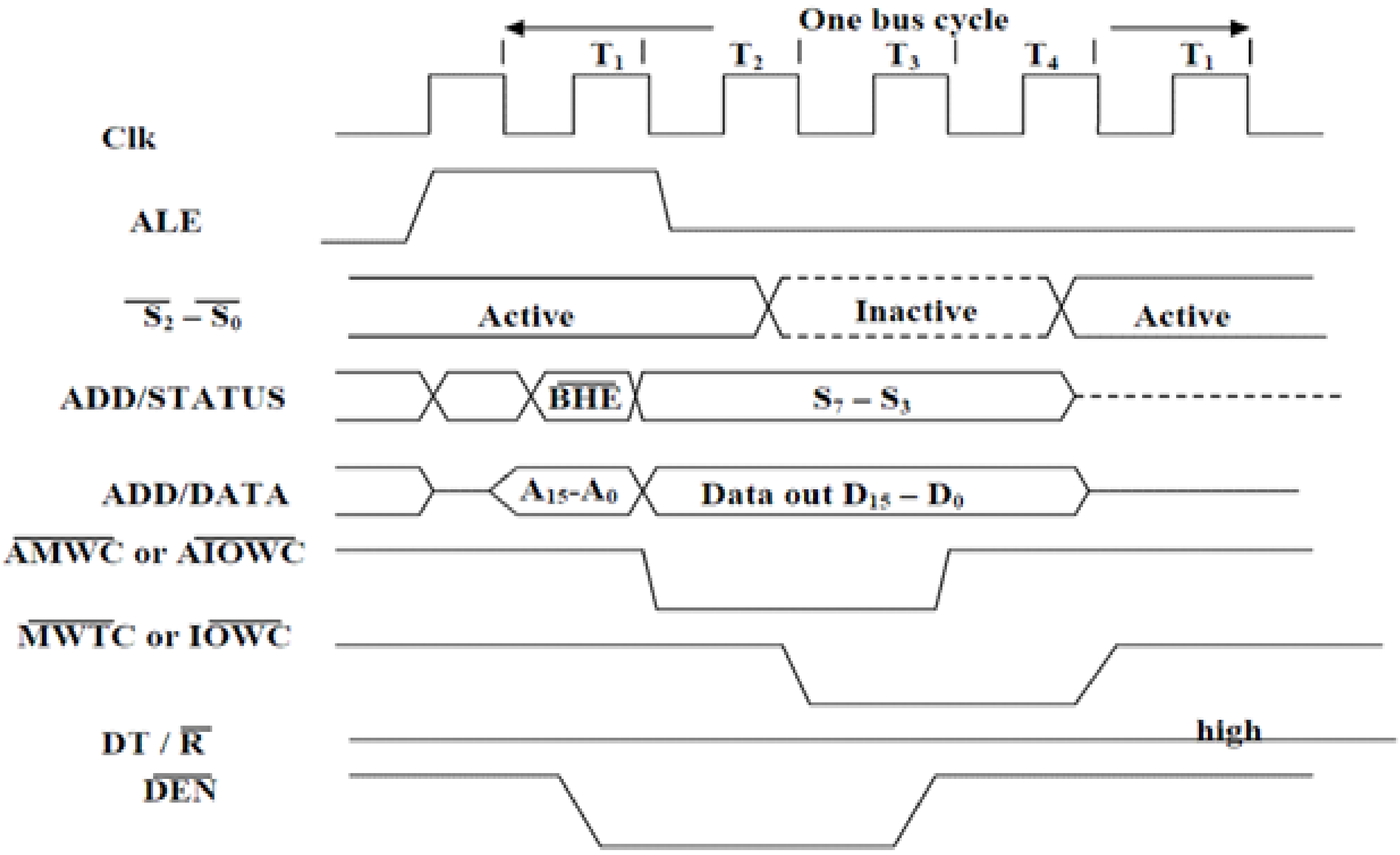
Maximum Mode 8086 System.

Memory Read Cycle Timing Diagram for Maximum Mode



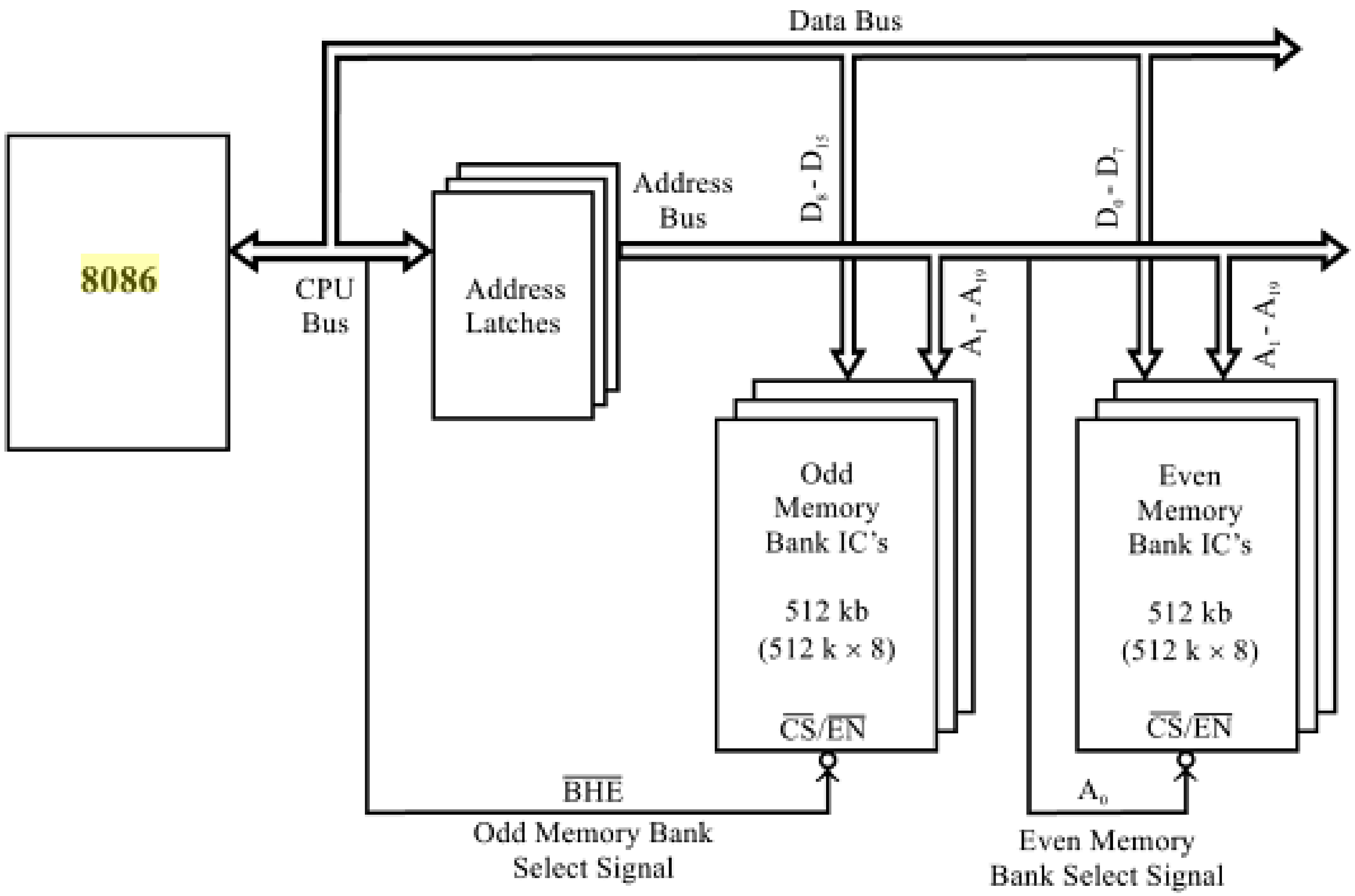
Memory Read Timing Diagram for Maximum Mode

Memory Write Cycle Timing Diagram for Maximum Mode

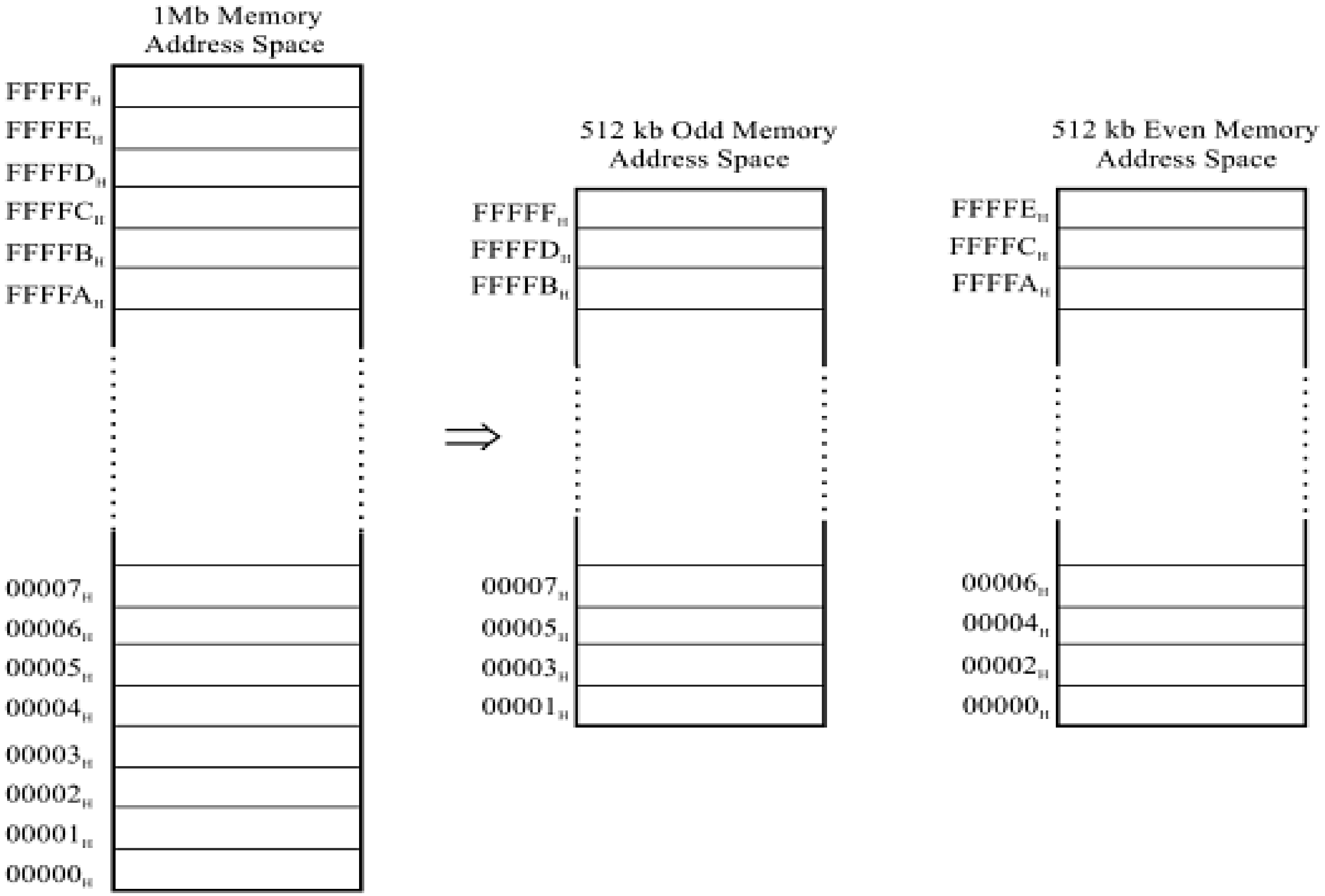


Maximum Mode Timing Diagram for Memory Write

PHYSICAL MEMORY ORGANIZATION in 8086-BASED SYSTEM



PHYSICAL MEMORY ORGANIZATION cont'd...



Organization of even and odd memory banks in 8086-based system.

PHYSICAL MEMORY ORGANIZATION cont'd...

- The 8086 microprocessor provides a 20-bit address to memory.

- The memory is organized as a linear array of up to 1 MB, addressed from **00000H to FFFFFH**.

- The 1MB of 8086 addressable memory space is divided into 2 banks : **Even memory bank** and **Odd memory bank**.

- Each bank will have an addressable space of 512 kB.

- In 8086-based system the **lower 8** lines of data bus, **Do- D_7**, are connected to **even bank** memory ICs and

- The **upper 8** lines of data bus, D_8- D_{15} are connected to **odd bank** memory .

PHYSICAL MEMORY ORGANIZATION cont'd...

- The processor provides two enable signals $\overline{BHE}$ and A_0.

- These two signals selectively allow reading from or writing into either an odd byte location, even byte location, or both.

- $\overline{BHE}$ is used to enable the odd bank and

- The A_0 address line is used to enable the even bank.

$\overline{BHE}$	A_0	Indication
0	0	Whole word is received/transmitted [D15-D0 data lines are active]
0	1	Byte from Odd memory bank [D15-D8 data lines are active]
1	0	Byte from Even memory bank [D7-D0 data lines are active]
1	1	Passive

PHYSICAL MEMORY ORGANIZATION cont'd...

- A microprocessor-based system requires both EPROM and RAM.

- Hence, the **available memory space** has to be **divided** between **EPROM and RAM.**

- This choice depends on the system designer as well as on the application for which the system is designed.

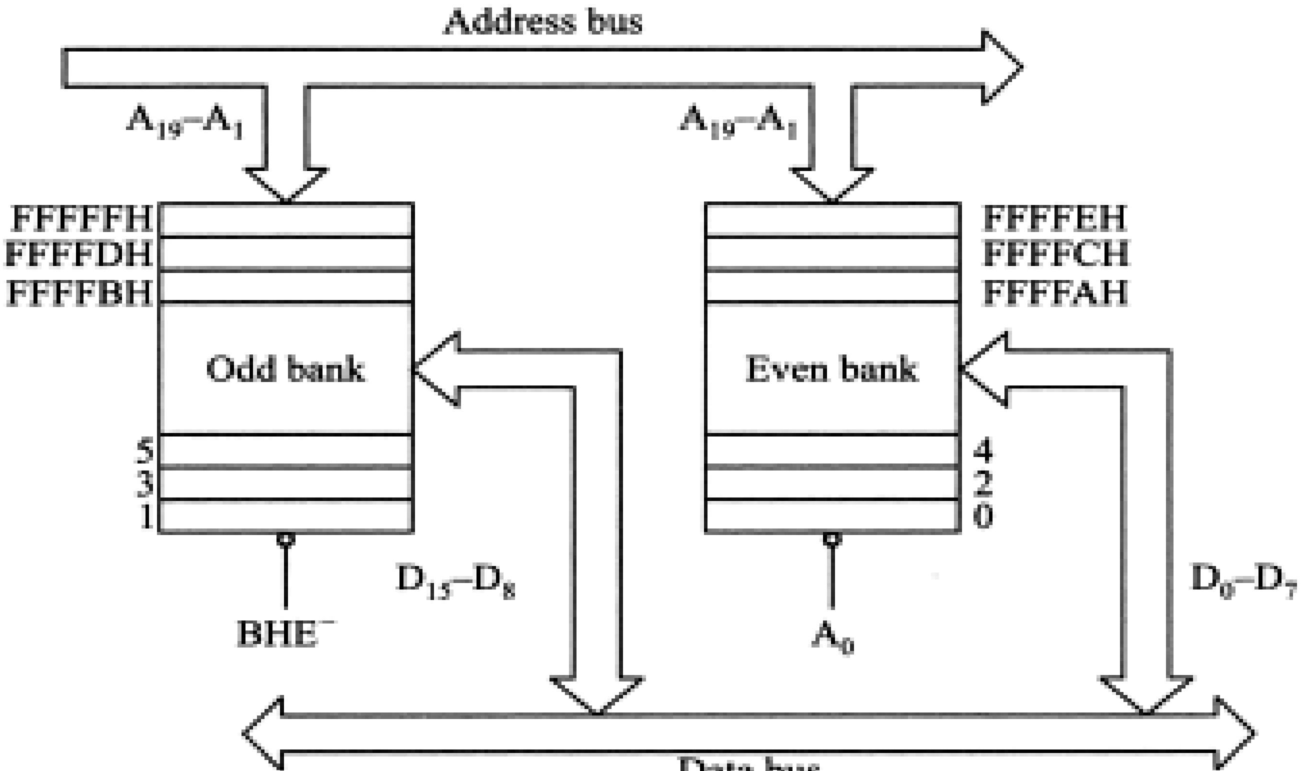
- The system designer should **allot equal address space** in odd and even bank for both **EPROM and RAM**.

- The required **EPROM** memory capacity can be implemented in **two** Ics (one for even and the other for odd bank) or in multiple ICs.

- Similarly, the RAM capacity of the system can be implemented in two ICs or in multiple ICs.
- This choice depends on the availability of memory IC and the system designer.

PHYSICAL MEMORY ORGANIZATION

cont'd...



Memory banks.

PHYSICAL MEMORY ORGANIZATION **cont'd…**

- Some systems may require large memory space and so full memory space is utilized.

- But in some systems, the memory requirement may be less and in this case the full memory space is not utilized.

- When full memory space is not utilized for memory, then the **unused memory addresses can be used for addressing I/O devices.**

- Such I/O devices are called **memory-mapped I/O devices** and **they can be accessed similar to that of memory device.**

Interfacing

1. I/O Interfacing

2. Memory Interfacing

I/O Addressing or I/O Interfacing

- The I/O devices in 8086 system may be interfaced with 8086 in **two** different ways.

  i) **I/O mapped I/O** and ii) **memory mapped I/O.**

- These two methods of I/O interfacing give rise to two different I/O address spaces.

I/O ADDRESSING cont'd..

I/O Mapped I/O

- The **I/O Mapped I/O** is also known as **Isolated I/O**.
- Here the I/O devices are treated as I/O devices only.
- The I/O Mapped I/O interfacing may be of **two types**, viz.

  Direct Addressing : I/O devices having 8-bit port addresses and

  Indirect Addressing : I/O devices having 16-bit addresses.
- In the first case, microprocessor 8086 can address up to 256 (2^8) input devices as well as 256 output devices.
- In the second case, microprocessor 8086 can address up to 64K (2^{16}) input devices as well as 64K (2^{16}) output devices.
- The address of the I/O device is called the port address.
- The control signals used in this space are $\overline{IOR}$ and $\overline{IOW}$.
- In this technique of interfacing, there are only two instructions for data transfer between an I/O device and microprocessor.
- These instructions are **IN and OUT**.

Memory Mapped I/O

- In memory mapped I/O, the **I/O devices are allotted 20-bit addresses**.
- This can be understood as that a memory register have been replaced by an I/O device.

- Now the I/O device has the address of a memory.

- In memory mapped I/O we can interface 1M input/ output devices with 8086.
- The **I/O location** in memory mapped I/O **is considered as a memory location.**

- The memory related control signals ($\overline{MEMR}$ and $\overline{MEMW}$) are used to read and write from the I/O.

- All the memory related instructions are used to access an I/O.

Difference between I/O mapped I/O and memory mapped I/O

I/O mapped I/O	Memory mapped I/O
8- or 16-bit port address.	20-bit port address.
Maximum 256 I/O ports can be interfaced using direct addressing and 64K I/O ports can be interfaced using indirect addressing.	1M I/O ports can be interfaced.
Less, **address decoding** hardware is required as only 8 address bits are to be decoded.	More address decoding hardware is required because in this case 16 address bits are to be decoded.
• $\overline{IOR}$ & $\overline{IOW}$ will be the control signals. • IN and OUT are the only instructions for data communication between the microprocessor and the I/O devices.	$\overline{MEMR}$ & $\overline{MEMW}$ are the control signals. Any memory related instructions, such as MOV etc. can be used for data communication between the microprocessor and the I/O devices.
I/O can communicate only with the accumulator.	I/O can communicate with any of the registers, just like a memory register.
I/O and memory can have the same address.	I/O and memory cannot have the same address.
There is no effect on memory size.	Memory size will reduce.
Arithmetic and logic operations cannot be performed directly on the data at I/O.	Arithmetic and logic operations can be performed directly on the data at I/O.

Memory Interfacing

- Semiconductor memories are organized as two dimensional arrays of memory locations.

- For example, 4K x 8 or 4K byte memory contains 4096 locations, where each location contains 8-bit data and only one of the 4096 locations can be selected at a time.

- Once a location is selected, all the bits in it are accessible using a group of conductors called 'data bus'.

- For addressing 4K bytes of memory, 12 address lines are required.

  1K (2^{10}) → **10** Address lines   4K (2^{12}) → **12** Address lines

- In general, to address a memory location out of **N memory locations** we require at least n bits of address. i.e. **n address lines** where **$n = \log_2 N$**.

- Thus if the microprocessor has n address lines, then it is able to address at the most N locations of memory, where **$2^n = N$**.

Memory Interfacing

Cont'd...

- The **least significant** p address lines out of the available n lines can be **directly connected** from the microprocessor **to the memory chip .**
- While the remaining **(n-p) higher order** address lines may be **used** for **address decoding** (as inputs to the **chip selection** logic).
- The memory address depends upon the hardware circuit used for decoding the chip select
- The output of the decoding circuit is connected with the $\overline{CS}$ pin of the memory chip.

Memory Interfacing

Cont'd...

The general **procedure of static memory interfacing** with 8086 is briefly described as follows:

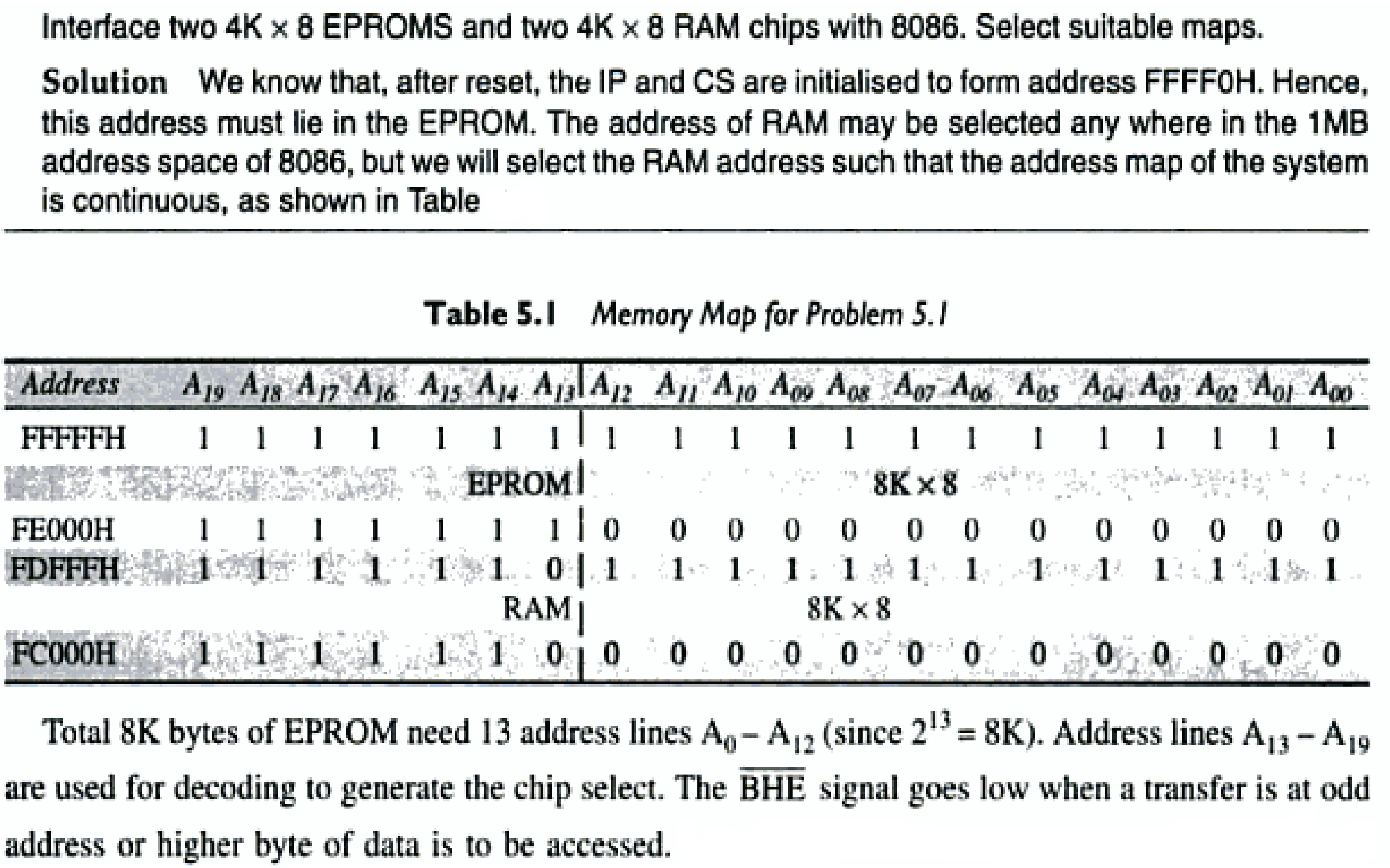
1. **Arrange the available memory chips** so as **to obtain 16-bit data bus width**. Connect the upper 8-bits of data bus to 'odd address memory bank' and then Connect the lower 8-bits of data bus to 'even address memory bank' .

2. Connect the 16-bit data bus of the memory bank with that of the 8086.

3. **Connect** available memory **address lines** of memory chips with those of the microprocessor and also connect the memory $\overline{RD}\,\&\,\overline{WR}$ inputs to the corresponding processor control signals.

3. The remaining address lines of the microprocessor , $\overline{BHE}$ and A_0 are used for decoding the required chip select signals for the odd and even memory banks. The $\overline{CS}$ of memory is derived from the O/P of the decoding circuit.

Problem 1:



Interface two 4K × 8 EPROMS and two 4K × 8 RAM chips with 8086. Select suitable maps.

Solution We know that, after reset, the IP and CS are initialised to form address FFFF0H. Hence, this address must lie in the EPROM. The address of RAM may be selected any where in the 1MB address space of 8086, but we will select the RAM address such that the address map of the system is continuous, as shown in Table

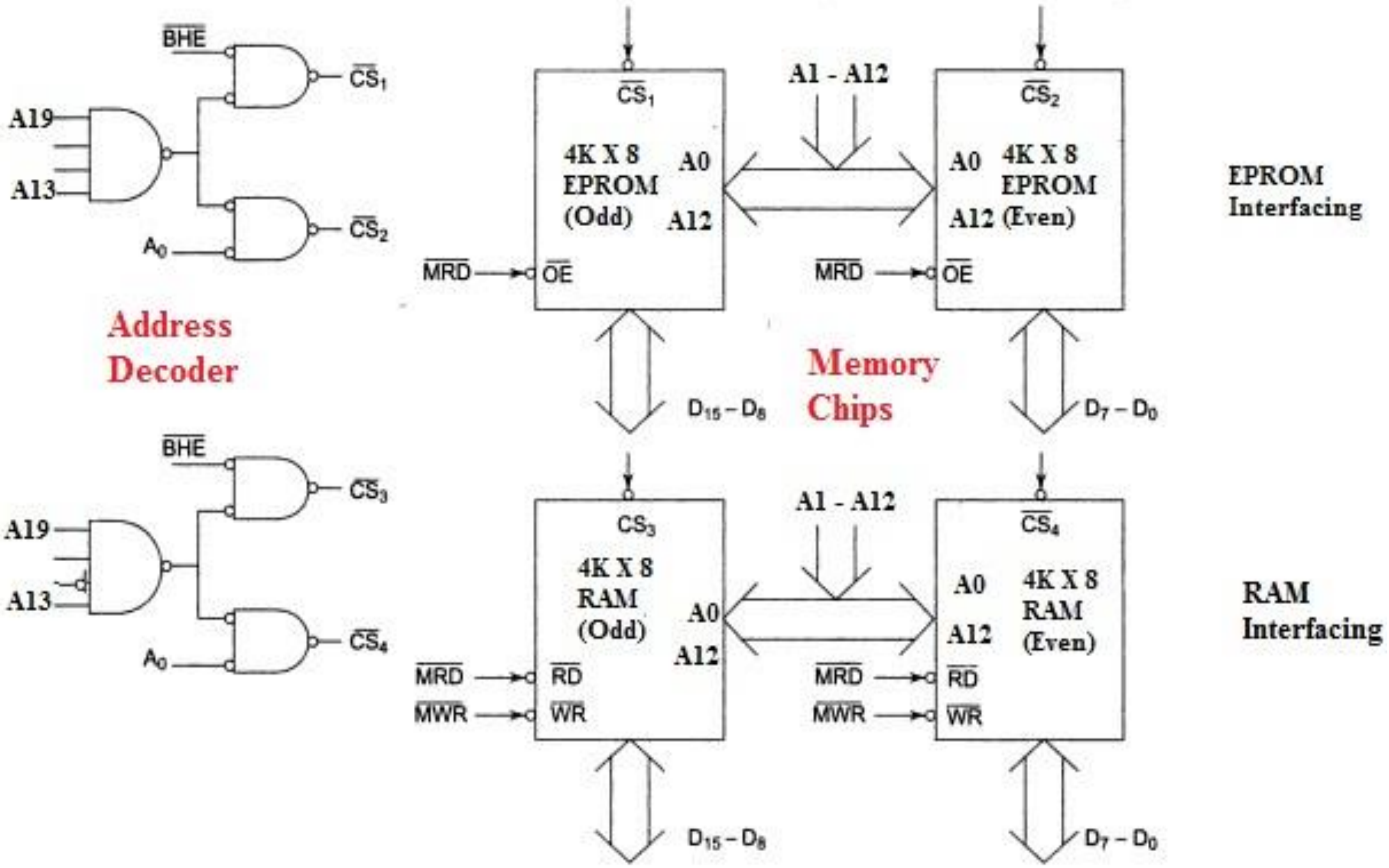
Table 5.1 *Memory Map for Problem 5.1*

Address	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
							EPROM						8K × 8							
FE000H	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
FDFFFH	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
							RAM					8K × 8								
FC000H	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

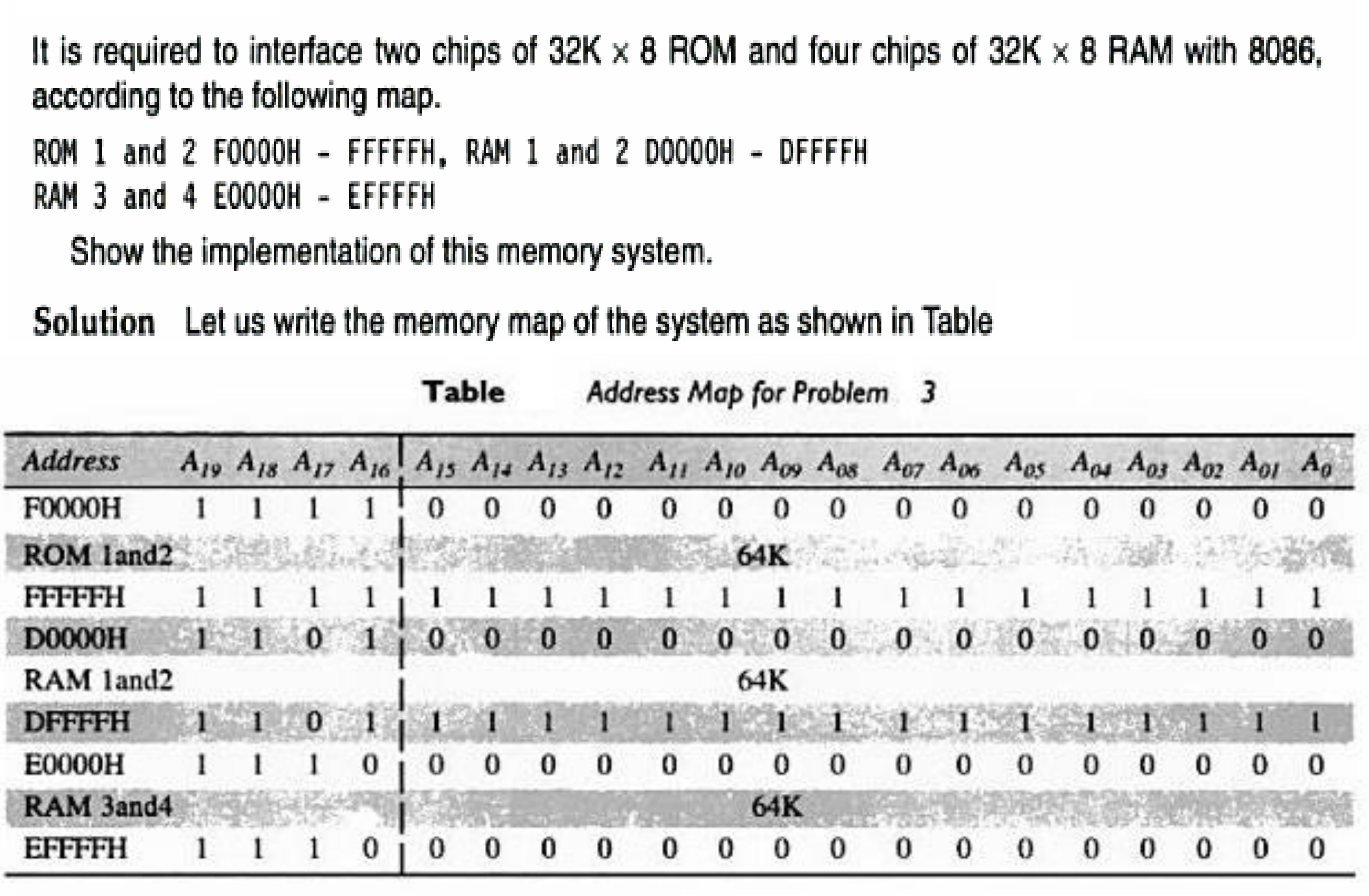
Total 8K bytes of EPROM need 13 address lines $A_0 - A_{12}$ (since $2^{13} = 8K$). Address lines $A_{13} - A_{19}$ are used for decoding to generate the chip select. The $\overline{BHE}$ signal goes low when a transfer is at odd address or higher byte of data is to be accessed.

Problem 1: cont'd…

All the address, data and control signals are assumed to be readily available.



Problem 2 :



It is required to interface two chips of 32K × 8 ROM and four chips of 32K × 8 RAM with 8086, according to the following map.

ROM 1 and 2 F0000H - FFFFFH, RAM 1 and 2 D0000H - DFFFFH
RAM 3 and 4 E0000H - EFFFFH

Show the implementation of this memory system.

Solution Let us write the memory map of the system as shown in Table

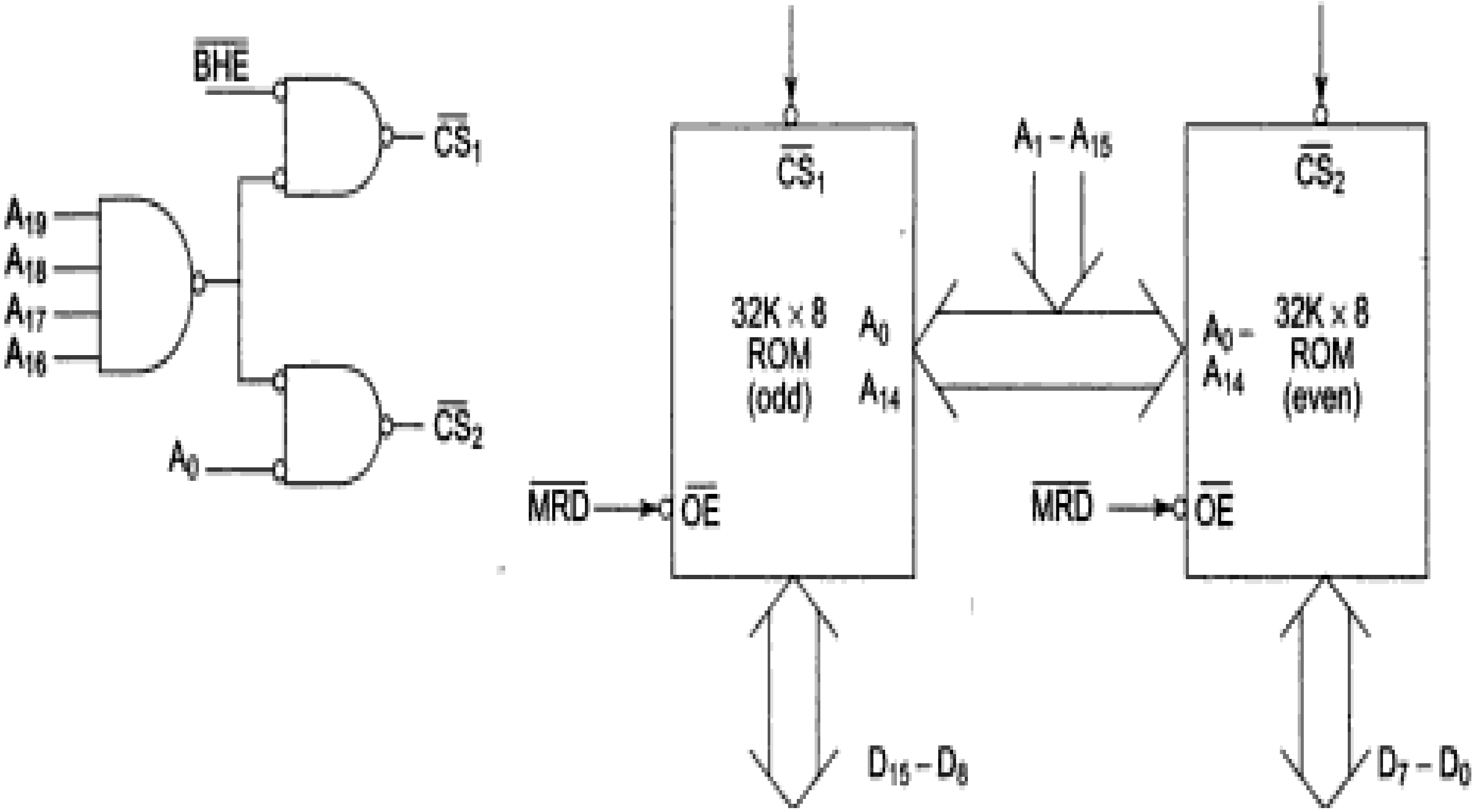
Table *Address Map for Problem 3*

Address	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{0}
F0000H	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ROM 1and2										64K										
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D0000H	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RAM 1and2										64K										
DFFFFH	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E0000H	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RAM 3and4										64K										
EFFFFH	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Problem 2 : cont'd…

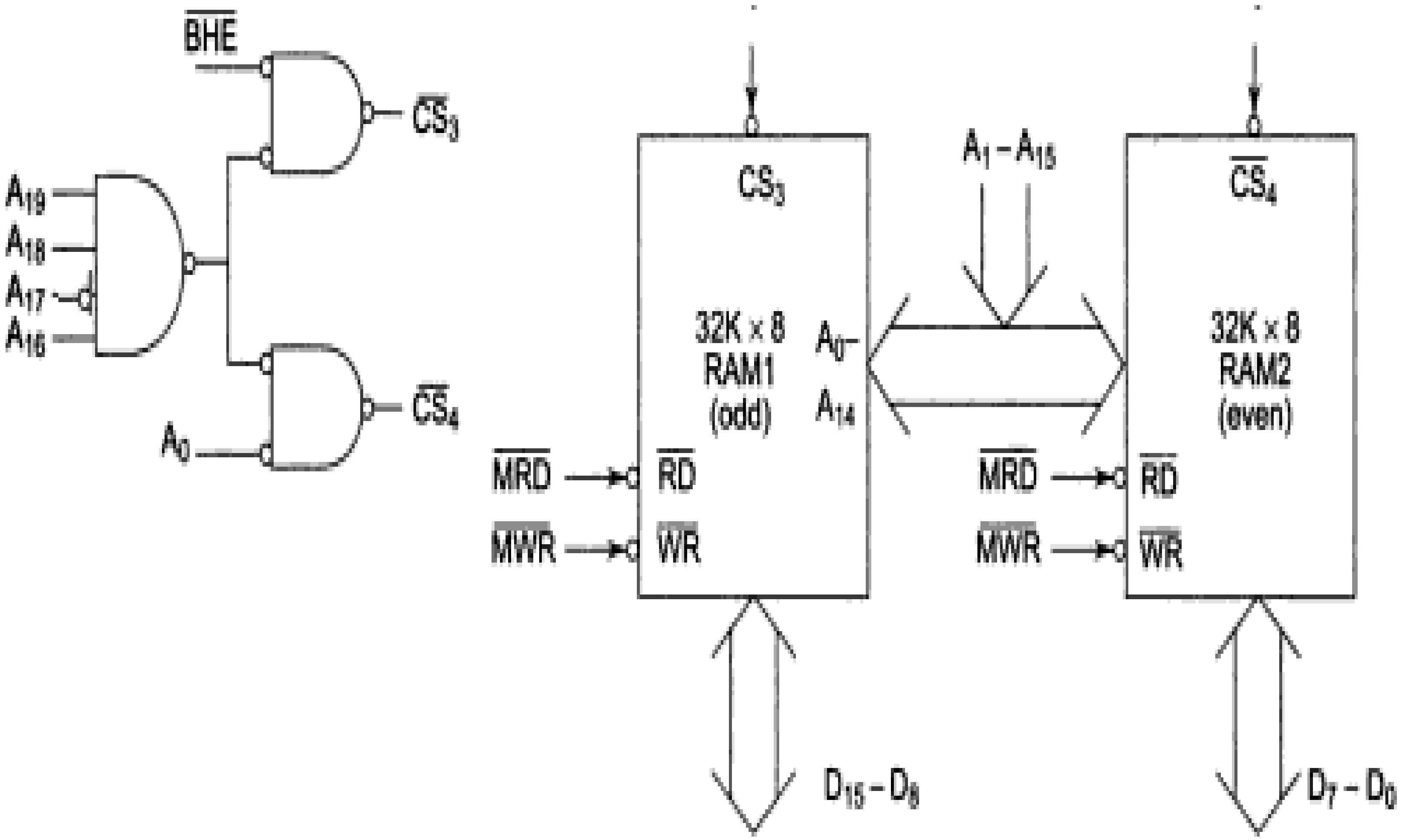
ROM Interfacing

All the address, data and control signals are assumed to be readily available.



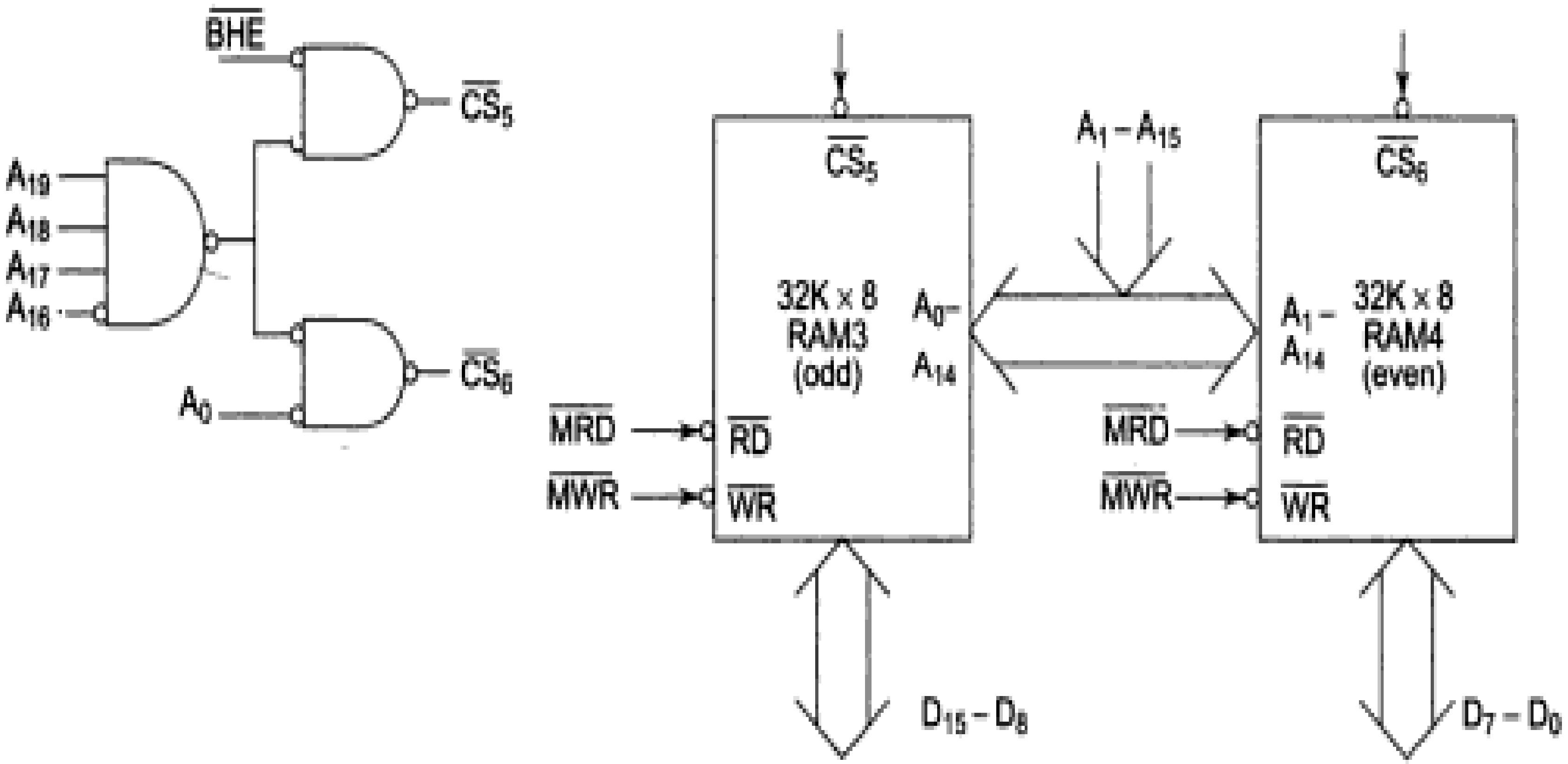
Problem 2 : cont'd…

RAM 1 & 2 Interfacing



Problem 2 : cont'd…

RAM 3 & 4 Interfacing


$\overline{BHE}$
$\overline{CS_5}$
A_{19}
A_{18}
A_{17}
A_{16}
A_0
$\overline{CS_6}$
$\overline{CS_5}$
32K x 8
RAM3
(odd)
A_0–
A_{14}
$\overline{MRD}$
$\overline{RD}$
$\overline{MWR}$
$\overline{WR}$
$A_1 - A_{15}$
$\overline{CS_6}$
A_1 –
A_{14}
32K x 8
RAM4
(even)
$\overline{MRD}$
$\overline{RD}$
$\overline{MWR}$
$\overline{WR}$
$D_{15} - D_8$
$D_7 - D_0$