Basic Peripherals and their interfacing with 8086/88

→ We discuss general peripheral devices and their interfacing techniques with the microprocessor.

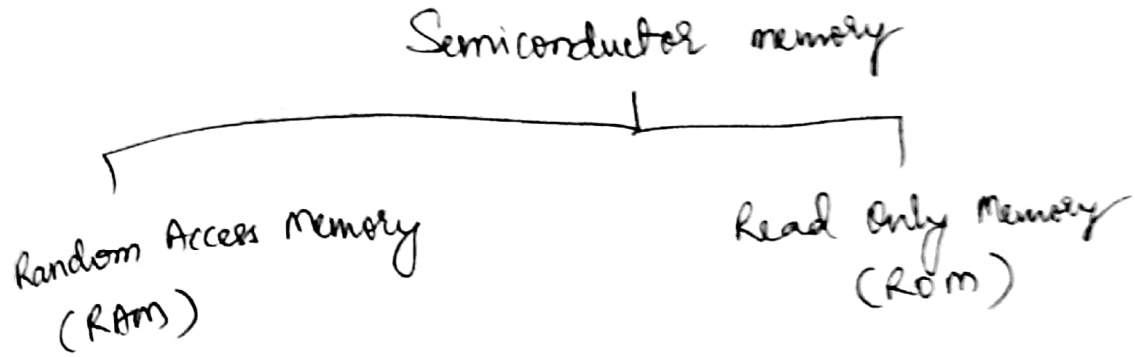
→ In the minimal working system configuration of a general microprocessor, we consider a keyboard, display system, memory system and I/o ports along with the CPU.

All the above devices are called peripherals or peripheral devices.

→ There are also some additional dedicated peripheral devices like
PIC (programmable interrupt controller)
DMA (direct memory access) controller
CRT (cathode ray tube) controller etc.

→ Interfacing of the above devices with microprocessor will be dealed here.

Semiconductor memory Interfacing:



Static RAM Interfacing:

→ Here, we consider the interfacing of Static RAM and ROM with 8086/8088

→ Semiconductor memories are organized as 2D arrays of memory locations.

Ex:- 4K×8 or 4K byte memory contains 4096 locations, where each location can store 8 bit data. and only one of 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'.

→ To address 4k bytes of memory, twelve address lines are required.

To address a memory location out of N memory locations we require

$$n = \log_2 N \text{ lines.}$$

However, if out of N locations only p memory locations are to be interfaced, then the least significant p address lines out of the available n lines can be directly connected to the memory chip, while the remaining (n-p) higher order address lines can be used for address decoding (as inputs to the chip selection logic)

The memory address depends on the hardware circuit used for decoding the chip select ($\overline{CS}$).

The output of the decoding circuit is connected with the $\overline{CS}$ pin of the memory chip.

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

(1.) Arrange the available memory chips so as to obtain 16 bit data bus widths.

The upper 8 bit bank is called the "odd address memory bank"

The lower 8 bit bank is called the "even address memory bank".

(2) Connect the available memory address lines of the memory chip with those of the microprocessor and also connect the memory $\overline{RD}$ and $\overline{WR}$ inputs to the corresponding processor control signals.

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

(3) The remaining address lines of the microprocessor, $\overline{BHE}$ and A_0 are used for decoding the required chip select signals for odd and even memory banks.

Example:

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

Sol

Note:

We know that, after reset, the IP, CS are initialized to form the address FFFF0H.

Hence, this address must lie in the EPROM.

The address of RAM may be selected anywhere in the 1MB address space of 8086, but we will select RAM address such that the address map of the system is continuous.

8K bytes of EPROM requires

$8K = 2^3 \cdot 2^{10} \Rightarrow$ 13 address lines ($A_0 - A_{12}$).

Address lines $A_{13} - A_{19}$ can be 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.

→ Let us assume that the latched address, $\overline{BHE}$ and demultiplexed data lines are readily available for interfacing.

Memory map :

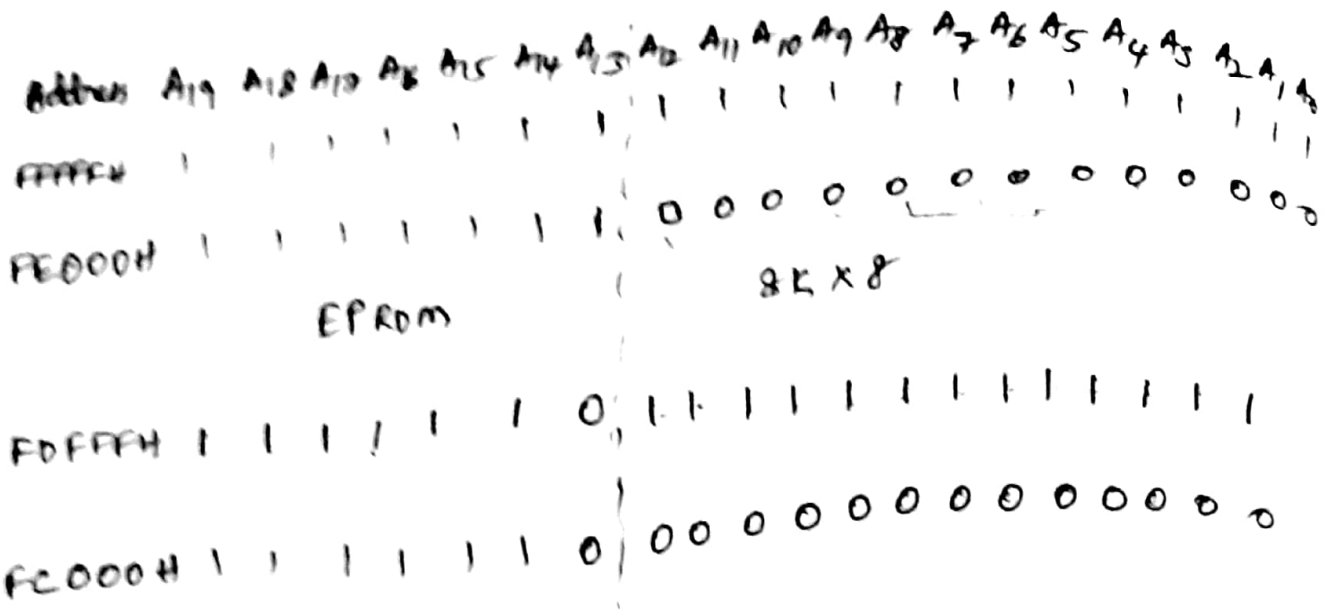
8086 has 20 address lines.

ROM size = 8k × 8

RAM size = 8k × 8

∴ no. of address lines need for ROM and RAM $8k = 2^3 \cdot 2^{10}$.

∴ 13 address lines.



Note: If we observe the memory map, we see that address locations in RAM and ROM are consecutive locations.

Chip Selection logic: 3 x 8 decoder

Decoder I/P Address / $\overline{BHE}$	A_2 A_1 A_0			Selection
	A_{13}	A_0	$\overline{BHE}$	
Word transfer $D_0 - D_{15}$	0	0	0	Even and odd addresses in RAM
Byte transfer $D_7 - D_0$	0	0	1	Only even address in RAM
Byte transfer $D_8 - D_{15}$	0	1	0	only odd address in RAM
NOT used	0	1	1	NOT used
Word transfer $D_0 - D_{15}$	1	0	0	Even and odd addresses in ROM
Byte transfer $D_7 - D_0$	1	0	1	Only Even address in ROM
Byte transfer $D_8 - D_{15}$	1	1	0	only odd address in ROM
NOT used.	1	1	1	NOT need

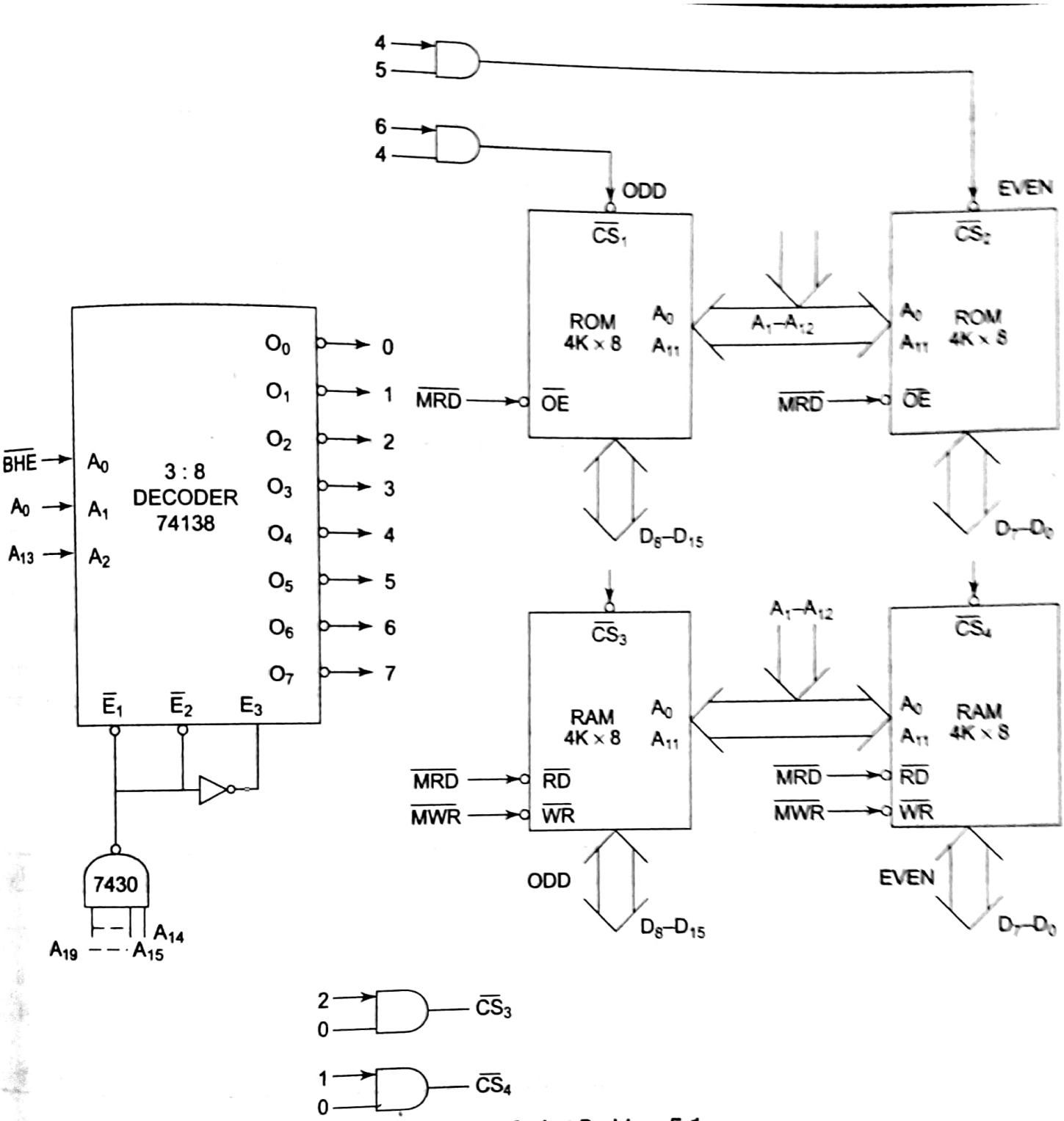


Fig. 5.1 *Interfacing Problem 5.1*

Ex:-

Design an interface between 8086 CPU and two chips of 16K x 8 EPROM and two chips of 32K x 8 RAM.

Select the starting address of EPROM suitably.

The RAM address must start at 00000H.

Sol

Note:

The last address in the memory map of 8086 is FFFFFH. After resetting, the processor starts from FFFF0H. Hence, this address must lie in the address range of EPROM.

Memory map:

ROM: 32K × 8 (Two 16K × 8 chips)

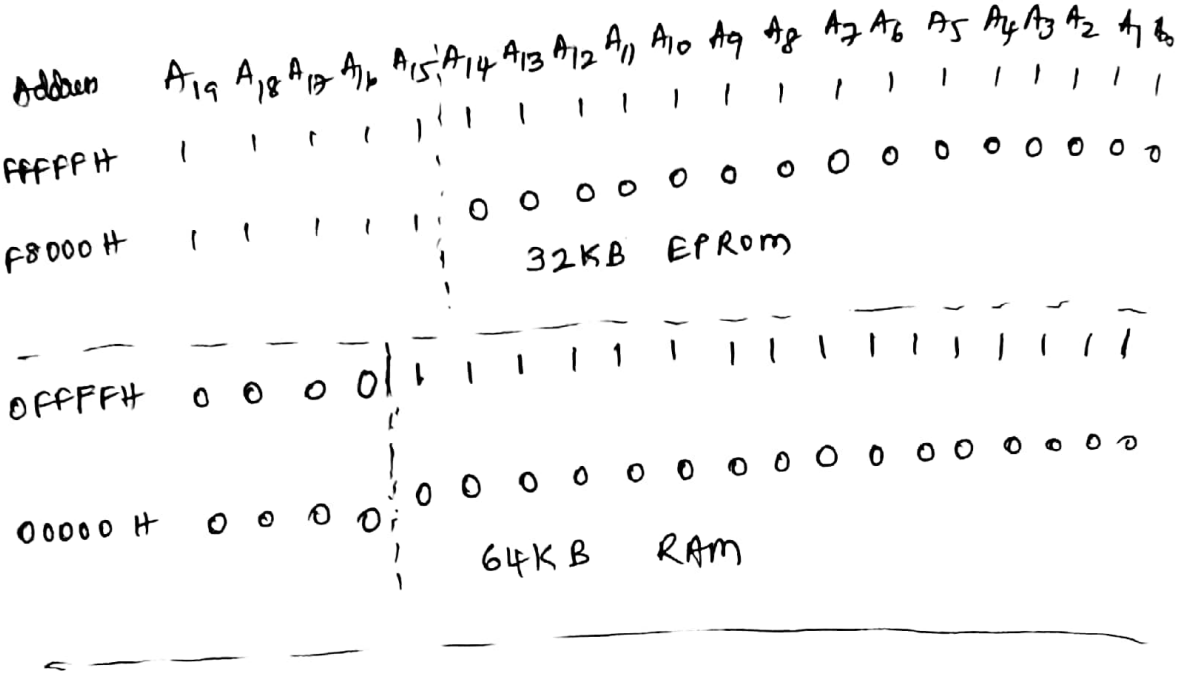
$32K = 2^5 \times 2^{10}$

∴ 15 Address lines are required for ROM

RAM: 64K × 8 (Two 32K × 8 chips)

$64K = 2^6 \times 2^{10}$

∴ 16 Address lines are required for RAM.



∴ chip selection logic uses the address lines $A_{16} - A_{19}$ for RAM

chip selection logic uses the address line $A_{15} - A_{19}$ for ROM.

Note :-

In this case, we will not use a decoder logic to determine the chip selection logic as the memory map is not continuous. There is some unused address space between the last RAM address (0FFFFH) and the first EPROM address (F8000H).

Hence, the chip selection logic is implemented using logic gates.

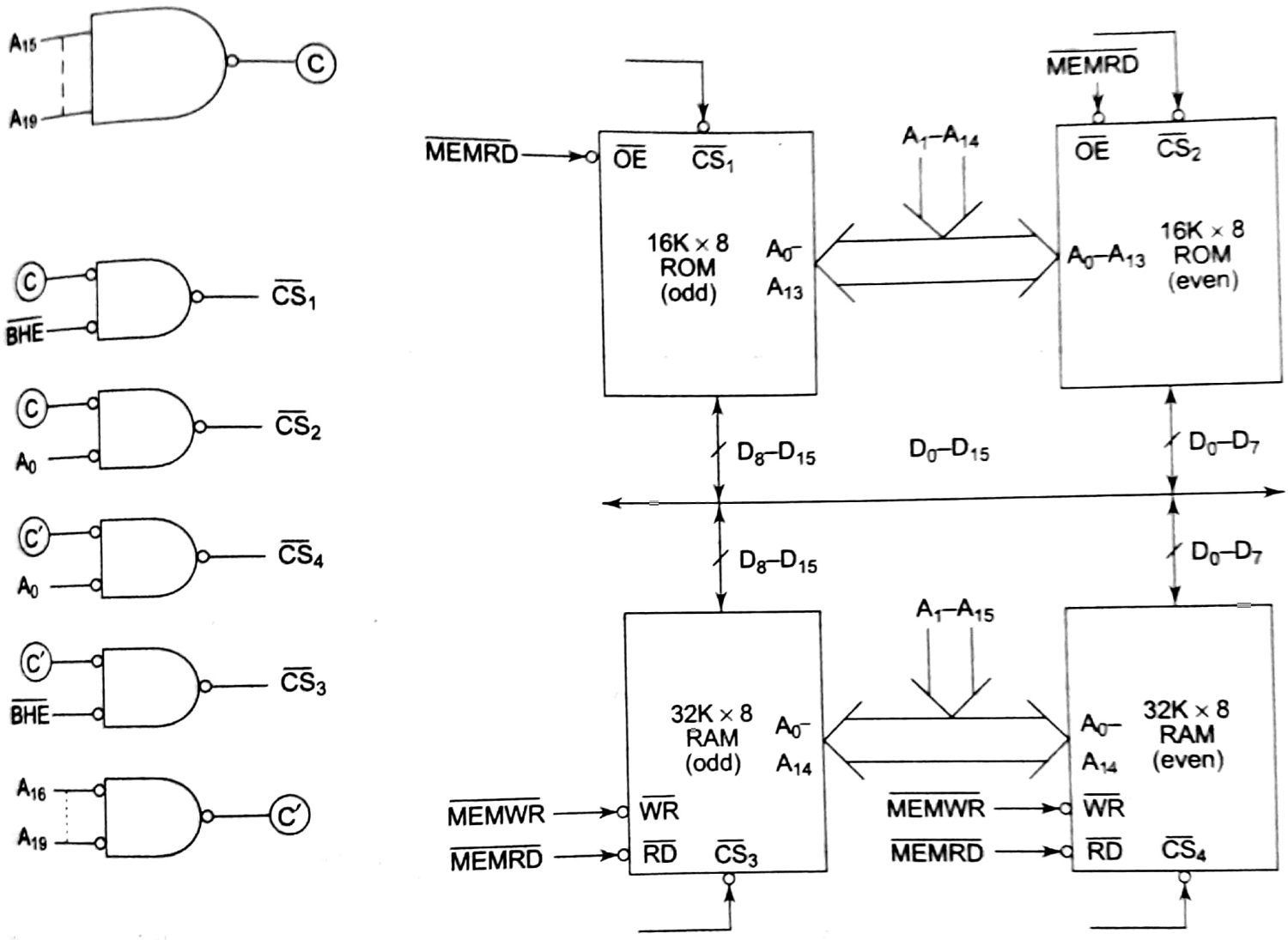


Fig. 5.2 *Interfacing Problem 5.2*

Let us select a variable C for memory address pulse, i.e. output of 6-input NAND gate. $\overline{BHE}$ is abbreviated as B. The chip selection logic can be designed as shown in Table 5.4.

Table 5.4

I/P		*O/P*	
A_0	$B(\overline{BHE})$	C_1	C_2
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

$C_2 = A_0$

$C_1 = \overline{BHE}$

To find out $\overline{CS_1}$ and $\overline{CS_2}$ we will have to combine C_1 and C_2 with C.

Table 5.5

I/P			O/P	
C_1	C_2	C	$\overline{CS_1}$	$\overline{CS_2}$
0	0	0	0	0
1	0	0	1	0
1	1	0	1	1
0	1	0	0	1

Table 5.5 shows that $\overline{CS_1} = C + C_1 = C + \overline{BHE}$ and $\overline{CS_2} = C + C_2 = C + A_0$

Similarly we can find out CS_3 and CS_4.

Q:-

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

ROM 1 and 2 : F0000H – FFFFFH

RAM 1 and 2 : D0000H – DFFFFH

RAM 3 and 4 : E0000H – EFFFFH

Sol

$32K = 2^5 \times 2^{10} = 15$ address lines for each chip

Memory map

Address	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
F0000H	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ROM 1 and 2					64K ROM																
EFFFFH	1	1	1	0	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 3 & 4					64K RAM																
DFFFFH	1	1	0	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 1 & 2					64K RAM																

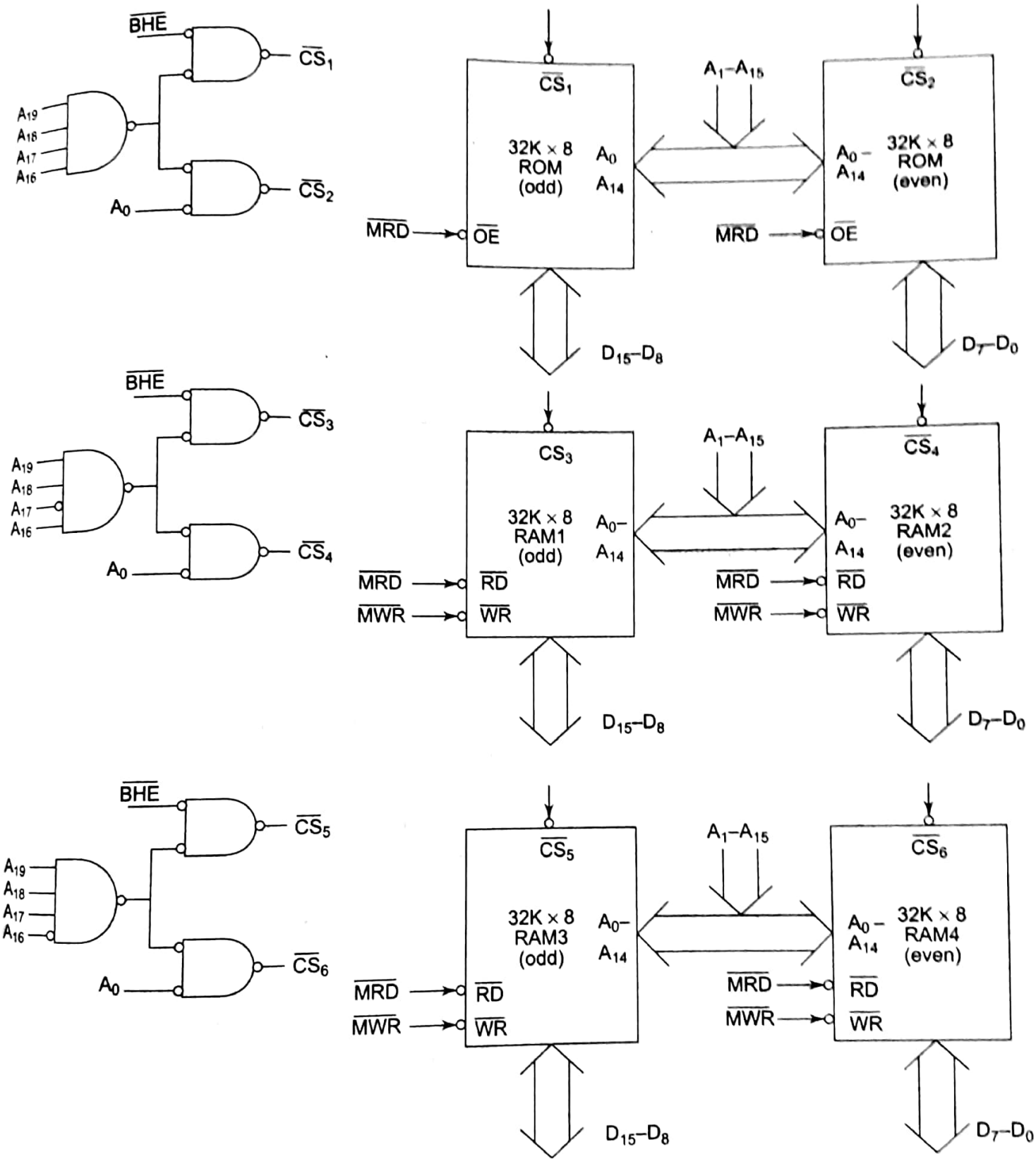


Fig. 5.3 *Interfacing Problem 5.3*

Dynamic RAM Interfacing :

→ Whenever a large memory is required in a microcomputer system, the memory subsystem is generally designed using dynamic RAM as it has various advantages eg. higher packaging density, lower cost, and less power consumption.

→ A typical static RAM cell may require six transistors while dynamic RAM cell requires only a transistor along with a capacitor.
Hence, it is possible to obtain higher packaging density and hence low cost units are available.

→ There are some drawbacks of dynamic RAMs.
The basic dynamic RAM cell uses a capacitor to store the charge as a representation of data. This capacitor is manufactured as a diode that is reverse biased so that storage capacitance comes into picture.

→ This storage capacitance is utilized for storing the charge representation of data, but the reverse biased diode has a leakage current that tends to discharge the capacitor giving rise to the possibility of data loss.

→ To avoid this data loss, the data stored in a dynamic RAM cell must be refreshed after a fixed time interval regularly.
This process of refreshing the data in the RAM is known as refresh cycle.

→ During this refresh period, all other operations related to memory subsystem are suspended. Hence the refresh activity causes loss of time, resulting in reduced system performance.

→ But, because of the advantages like low power consumption, higher packaging density and low cost, most computer systems are designed using dynamic RAM, at the cost of operating speed.

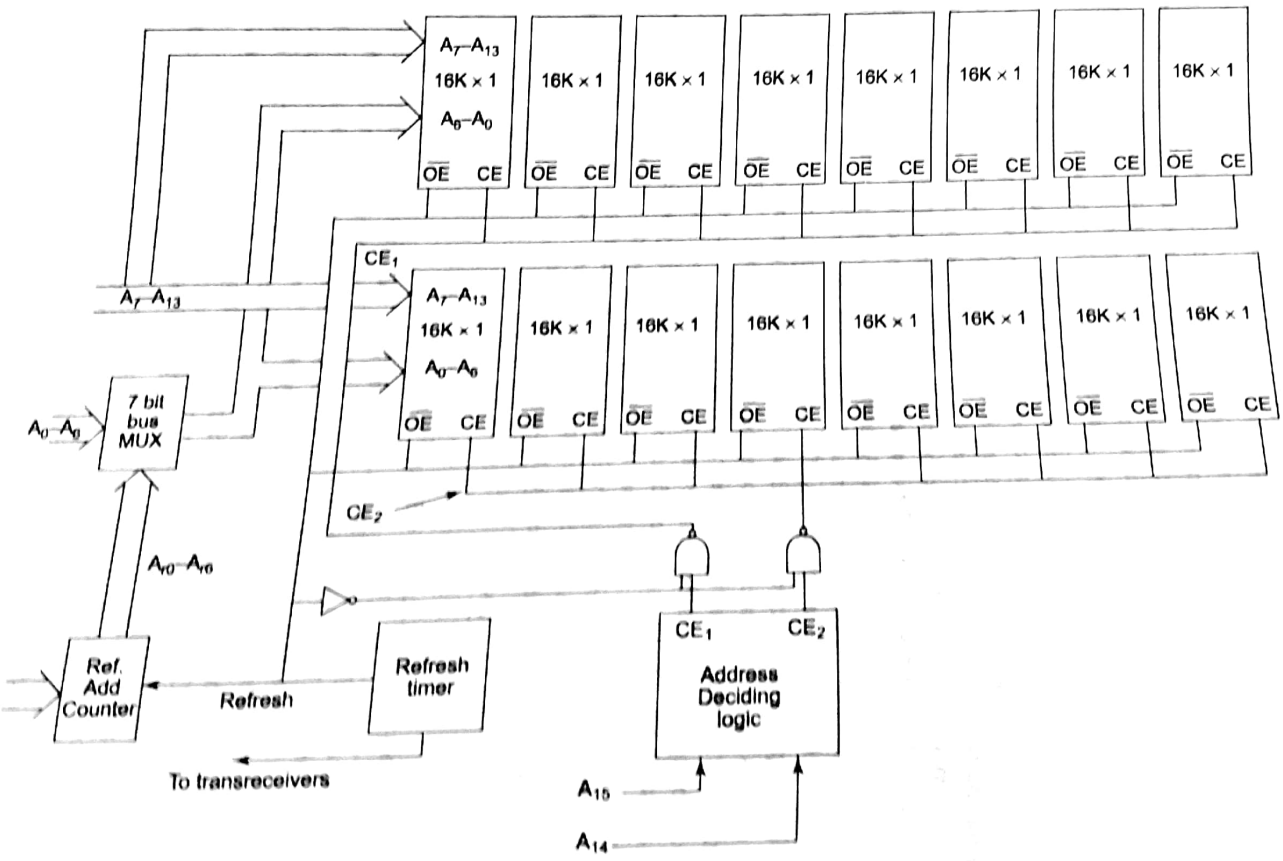


Fig. 5.7 *Dynamic RAM Refreshing Logic*

→ The above block diagram, explains the refreshing logic and 8086 interfacing with dynamic RAMs.

→ Each used chip is a 16K x 1 bit dynamic RAMs cell array.

→ The system contains two 16K byte dynamic RAMs units.

→ All the address and data lines are assumed to be available from an 8086 microcomputer system.

→ The $\overline{OE}$ pin controls the output data buffers of memory chips.

→ The CE pins are active high chip selects of memory chips.

→ The refresh cycle starts, if the refresh output of refresh timer goes high and $\overline{OE}$, CE also tend to go high.

→ The high $\overline{CE}$ enables the memory chip for refreshing, while high $\overline{OE}$ prevents the data from appearing on the data bus.

→ The 16K x 1 bit dynamic RAM has an internal array of 128 x 128 cells, requiring 7 bits for row addresses.

The lower order seven lines $A_0 - A_6$ are multiplexed with the refresh counter output $A_{10} - A_{16}$.

→ The pin assignments for 2164 dynamic RAMs are shown below.

$\overline{RAS}$ and $\overline{CAS}$ are row and columns address strobes and are driven by the dynamic RAM controller outputs.

$A_0 - A_7$ lines are row or column address lines driven by $OUT_0 - OUT_7$ outputs of the controller.

$\overline{WE}$ pin indicates the memory write cycle.

→ The D_{IN} and D_{OUT} pins are data pins for write and read operations, respectively.

→ In practice, the refreshing logic is integrated inside dynamic RAM controller chips like 8203, 8202, 8207 etc.

→ Intel 8203 is a dynamic RAM controller that supports 16K or 64K dynamic RAM chips. This selection is done using pin 16K/64K. If this pin is high, 8203 is configured to control 16K dynamic RAM, else it controls 64K dynamic RAM.

→ The address inputs of 8203 controller accept address lines A_1 to A16 on line $AL_0 - AL_7$ and $AH_0 - AH_7$. A0 line is used to select even or odd bank.

→ $\overline{RD}$, $\overline{WR}$ signals indicate whether the cycle is memory read or memory write cycle and are accepted as inputs to 8203 from the microprocessor.

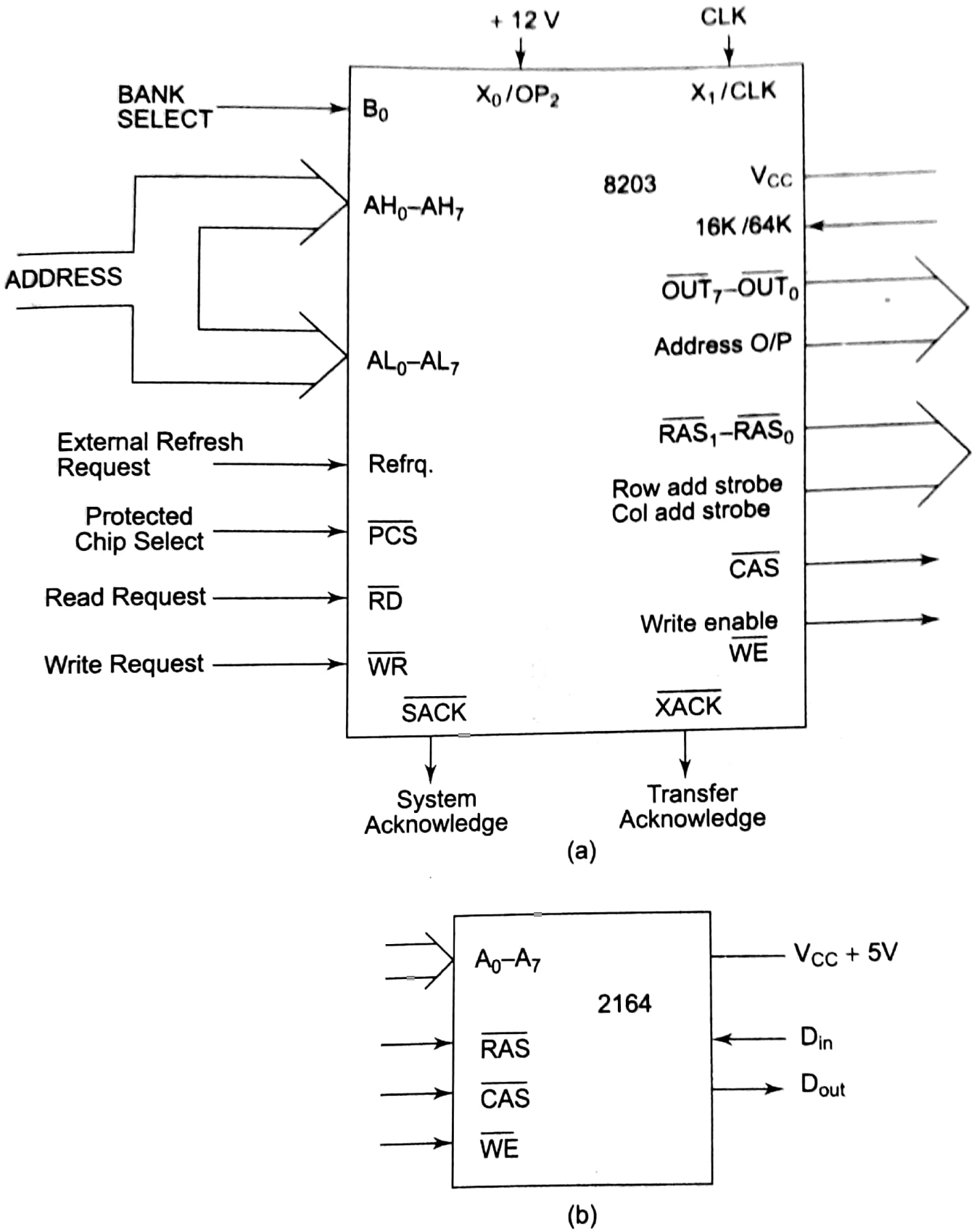


Fig. 5.8 *(a) Dynamic RAM controller (b) 1-bit Dynamic RAM*

→ $\overline{WE}$ signal specifies the memory write cycle and are accepted as output from 8203 that drives the $\overline{WE}$ input of the dynamic RAM memory chip.

→ The $\overline{OUT_0}$ - $\overline{OUT_7}$ set of eight pins is an 8 bit output bus that carries the multiplexed row and column addresses of a bit locations in a dynamic RAM chip.

→ The $\overline{CAS}$ signal strobes the column addres on $\overline{OUT_0}$ - $\overline{OUT_7}$.

→ The $\overline{RAS_1}$ - $\overline{RAS_0}$ pins strobe the row address on $\overline{OUT_0}$ - $\overline{OUT_7}$, for at the most two banks of 2164 dynamic RAM chips.

→ An external crystal may be applied between X_0 and X_1 pins, otherwise, with the OP_2 pin at +12V, a clock signal may be applied at the pin CLK.

→ The $\overline{PCS}$ pin accepts the chip select signal derived by an address decoder.