Memory Interfacing

Interfacing :-

Although microprocessors are very powerful and can carry out many functions, in order for it to be useful it has to be built into a system such as a microcomputer, along with other peripherals such as memory and input/output units. The technique of connecting these peripherals to the microprocessor is called interfacing.

µP has only one set of address and data lines to be used by all the peripherals. This means that the interfacing technique should include some sort of scheme to select the peripheral device that is required to communicate with the µP while at the same time isolating all other peripherals, even though all the peripherals stay connected to the µP all the time.

8086 Memory Designing/Interfacing

8086 has 1 MB memory 00000
:
FFFFF

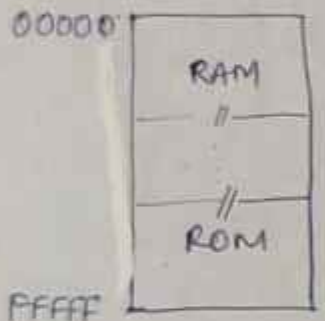
Designing a mly interface for a µP consists of the following steps:

1) Decide the size of the mly required and the address range it should occupy (Memory Map).
2) Choose the type of mly device to be used based on cost and other conditions such as speed, availability etc (RAM, EPROM etc)
3) Based on this information, we now know how many chips of the chosen device are needed to give the required mly size. Then design the circuitry (Memory Address Decoder) that will enable the mly to communicate with the µP whenever µP initiates either a read or write cycle and disables it at other times.
If the mly consists of more than one chip, only the appropriate chip or chips should be enabled.

Note:-
1) While interfacing mly to 8086 ensure that atleast 4K of ROM is available in the beginning locations - starting at 00000H as IVT is stored in this location.

2) While interfacing m/y to 8086 ensure that atleast 4K of ROM is available in the last locations – ending at location FFFFFH. This is due to the fact that on RESET the first instruction is executed from location FFFF0H.

[The 2K restriction is because the smallest m/y chip available is 2K. 2K E and 2K O adds upto 4K.]



The starting address of RAM is known. But for ROM, the starting address is to be calculated.

How?
Subtract the size of ROM from the ending address of ROM.

For eg. 128 KB EPROM using 64 KB chips

$$\text{No. of chips} = \frac{\text{Required}}{\text{Available}} = \frac{128\ KB}{64\ KB} = 2 \text{ chips.}$$

$$128\ KB = 128 \times 1K$$
$$= 2^7 \times 2^{10} = 2^{17} \quad (0001\ 1111\ 1111\ 1111\ 1111)$$
$$1\ F\ F\ F\ F$$

Starting address = FFFFF
− 1FFFF
E0000

Subject: Microprocessor **Semester: IV**

(1) Design a 8086 based maximum mode system having 32 KB EPROM using 16 KB chip's and 128 KB RAM using 32 KB chips.

EPROM

Required = 32 KB
Available = 16 KB
No of chips = 2

Starting address of EPROM : FFFFFH − Size of EPROM (32 KB)

32 KB = $2^5 \times 2^{10} = 2^{15}$ Put 15 ones and make them into sets of 4

0111 1111 1111 1111 = 7FFF

```
  FFFFF
- 7FFFF
-------
  F8000
```

Size of a single EPROM chip = 16 KB = $16 \times 1K = 2^4 \times 2^{10} = 2^{14}$

∴ 14 address lines ($A_{14} \cdots A_1$)

Note :- µP never gives A_0 inside any chip (because it is used for banking). So address lines are from $A_{14} \cdots A_1$.

RAM

No of chips = $\frac{\text{Required}}{\text{Available}} = \frac{128}{32} = 4$ chips.

Size of a single RAM chip = 32 KB = $2^5 \times 1K = 2^{15}$

∴ 15 Address lines ($A_{15} - \cdots A_1$)

Subject: Microprocessor

Memory Map

Semester: IV

Memory Chip	Address Bus																				Memory Address
	A19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	
RAM 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
(LB)	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0FFFEH
RAM 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001H
(HB)	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0FFFFH
RAM 3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10000H
(LB)	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1FFFEH
RAM 4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10001H
(HB)	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFFH
EPROM 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F8000H
(LB)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH
EPROM 2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	F8001H
(HB)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH

Note:- Memory map is a graphical representation of the total available mly in a µP system and how it is used ie, it shows all the chips, their starting and ending addresses.
MAP always begins with RAM.

Subject: Microprocessor **Semester: IV**

RAM1 begins at 00000 H.

RAM1 ends ? ⇒ make $A_1 - A_{15}$ all ones (keep others as it is).

Keeping $A_0 = 0$ (even bank), and doing this, all the addresses in this chip are even addresses.

So in the next RAM chip, all addresses are odd.

So RAM2 has $A_0 = 1$.

RAM2 begins at 00001 H

RAM2 ends : 0FFFF H

Together RAM1 and RAM2 make a band.

A band is a corresponding set of lower bank (LB) and higher bank (HB).

Now the next band will start from the next address. ie, we finished at 0FFFF H.

```
  0FFFF H
+     1
-------
  10000   ← next address.
```

We have total 6 chips in the interfacing circuit.

At time only 2 chips (one band) will be selected.

The higher byte of the address identifies which band is selected.

00000 H } 1st band
0FFFFF H }

10000 } 2nd band
1FFFF }

F0000 } 3rd band
FFFFF }

Address

09234 belongs to 1st band

19234 belongs to 2nd band.

F9234 belongs to 3rd band.

So to identify that the address belongs to which band, consider the higher 4 bits of the address.

Since we are using a 3:8 decoder, we need to take 3 consecutive lines from the higher byte such that they have different patterns for different banks.

So A_{18}, A_{17}, A_{16} is selected.

When $A_{18}\ A_{17}\ A_{16} = 000$, Y_0 is activated and 1st bank will be selected if we give Y_0 to $\overline{CS}$ (Chip select) of first bank. So 16 bit operation can be performed.

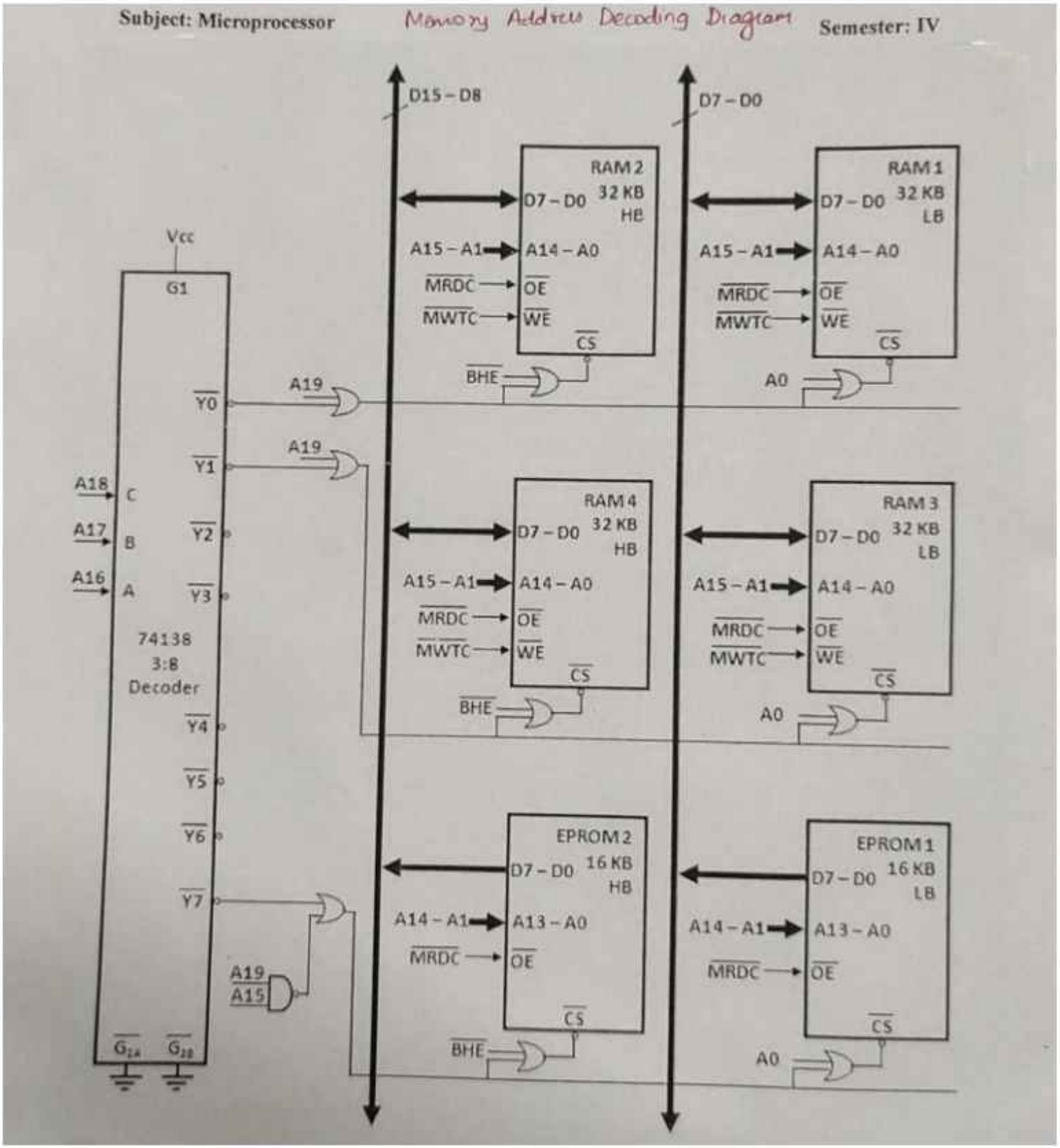
But at times, we want to perform 8 bit operations. ie, 3 options are required.

1) Select only lower bank
2) Select only higher bank
3) Select both banks.

∴

$\overline{BHE}$	A_0	Action
0	0	Both
0	1	Higher Bank
1	0	Lower Bank
1	1	None

Subject: Microprocessor — Memory Address Decoding Diagram — Semester: IV



Subject: Microprocessor **Semester: IV**

Memory address decoding circuit utilises the address lines of the µP to generate the signals that enable the appropriate memory chips.

Note:-

For RAM → A_1-A_{15} have gone inside the chip. } Hence these cant be used for decoder.
For EPROM → A_1-A_{14} have gone inside the chip }

Only if you get a condition where you dont find common lines which can differentiate the bands, use the lines which go inside the chip.

Sometimes, you dont find different patterns for all bands.

- Sometimes, we get different decoder o/p's for all bands
- Sometimes, we get one decoder o/p for 2 bands. Then split it for the 2 bands.
- Sometimes you get 2 decoder o/p for the same band, then combine them for one band.

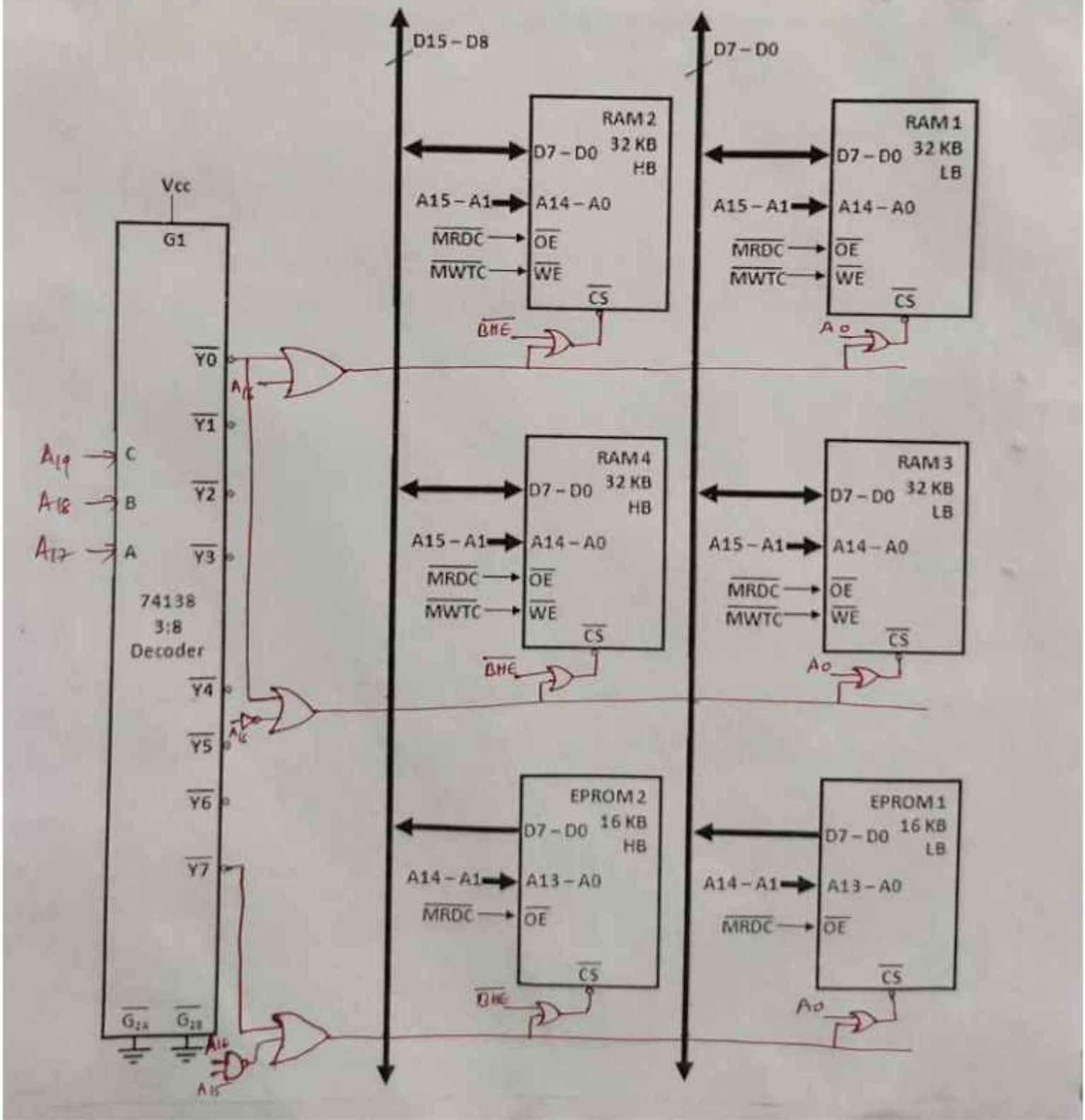
Note:- If only A_{18}, A_{17} and A_{16} are used in the generation of chip select (ie, only few bits of higher address line), it is called partial decoding.

If A_{16}, A_{17}, A_{16}, A_{19} and A_{15} are also used in the generation of chip select (ie, all higher address lines are used), then it is called absolute decoding.

Subject: Microprocessor **Semester: IV**

How to split a common decoder o/p to select a bank?

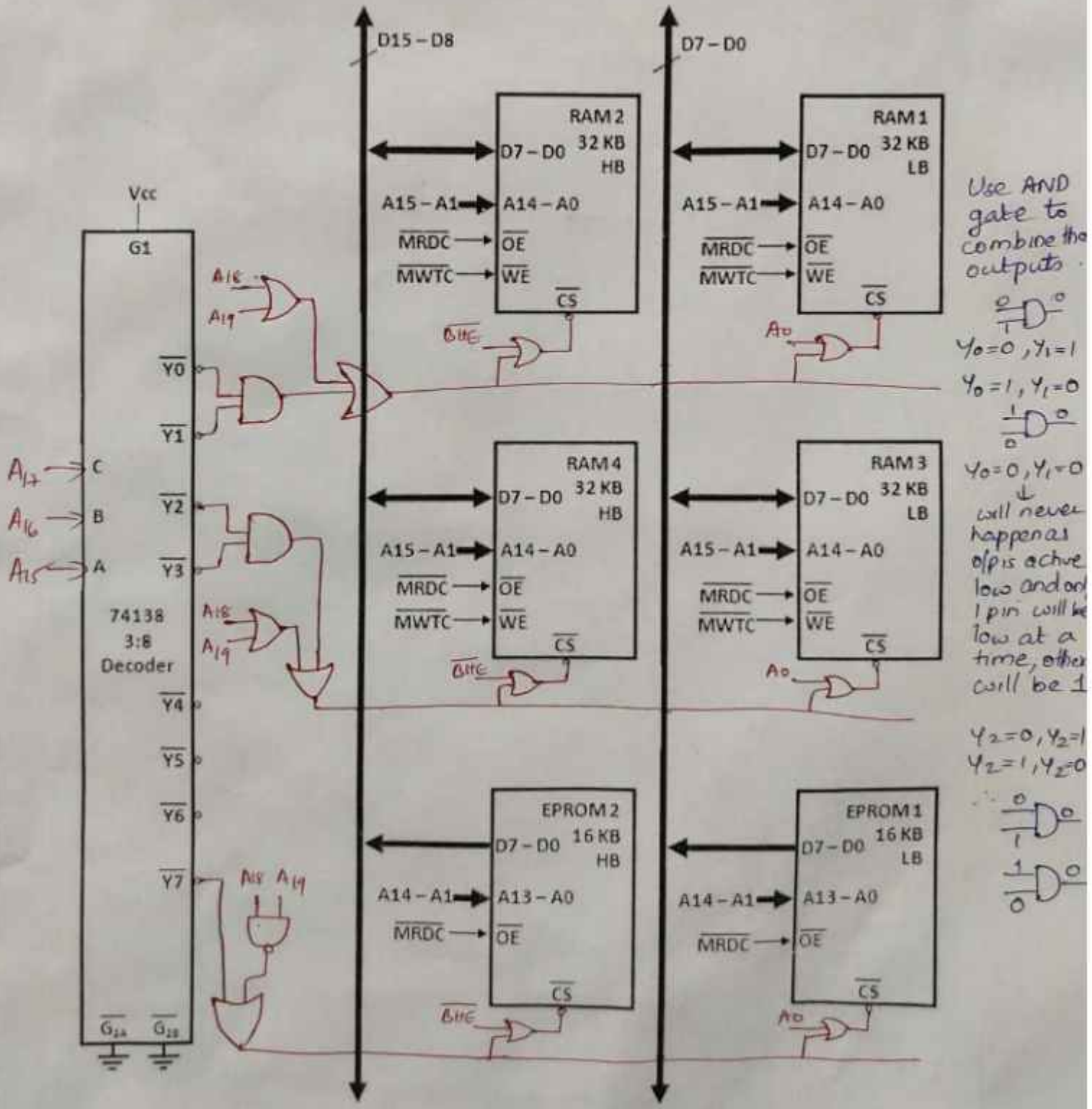
- A_{19}, A_{18}, A_{17} are common for bank1 and bank2. So look for neighbouring lines that differentiate the 2 banks. Here A_{16} differentiates bank1 and bank2.



Subject: Microprocessor Semester: IV

How to combine 2 decoder o/p to select a bank ?

A_{17}	A_{16}	A_{15}	
0	0	0	$\rightarrow Y_0$ } Bank 1 will be selected for both Y_0 & Y_1
0	0	1	$\rightarrow Y_1$ }
0	1	0	$\rightarrow Y_2$ } Bank 2 will be selected for both Y_2 & Y_3
0	1	1	$\rightarrow Y_3$ }



Use AND gate to combine the outputs.

$Y_0=0, Y_1=1$

$Y_0=1, Y_1=0$

$Y_0=0, Y_1=0$ ↓ will never happen as o/p is active low and only 1 pin will be low at a time, other will be 1

$Y_2=0, Y_2=1$

$Y_2=1, Y_2=0$

Subject: Microprocessor **Semester: IV**

On the same problem.

Design a 8086 based max mode system working at 6MHz having the following:

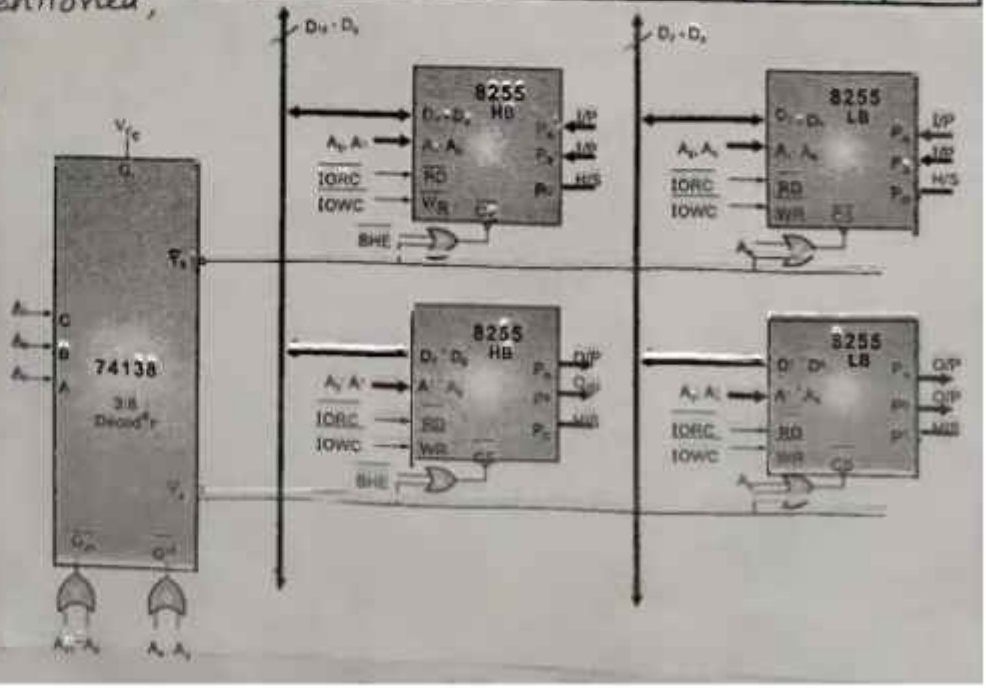
- 32 KB EPROM using 16 KB chips.
- 128 KB RAM using 32 KB chips.
- Two 16 bit i/p & Two 16 bit o/p ports, all interrupt driven

I/O Map →

I/O Port	Address Bus																I/O Address
	A15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	
8255 LB																	
Port A	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0060H
Port B	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0062H
Port C	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0064H
C Word	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0066H
8255 HB																	
Port A	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0061H
Port B	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0063H
Port C	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0065H
C Word	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0067H
8255LB																	
Port A	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0080H
Port B	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0082H
Port C	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0084H
C Word	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0086H
8255 HB																	
Port A	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0081H
Port B	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0083H
Port C	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0085H
C Word	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0087H

Note:- If direct addressing / fixed addressing is asked, then use 8 bit address like 80H ($A_0 - A_7$)

If Indirect addressing / Variable port addressing is mentioned, then we 16 bit address like 0080H ($A_{19} - A_0$)



Subject: Microprocessor **Semester: IV**

Note:- Address decoding is of 2 types - [Partial, Absolute

Absolute / Exhaustive Decoding	Partial Decoding
In this type of scheme all the 20 bits of 8086 address bus are used to select a particular location in mly chip.	In this scheme minimum no of address lines are used as required to select a mly location in a chip (ie, all lines are not used)
Advantages :-1) Complete address utilization 2) Ease in future expansion 3) No bus contention, as all addresses are unique.	Advantages - Simple, cheap and fast.
Disadvantages :- 1) Increased hardware and cost 2) Speed is less due to increased delay.	Disadvantages - 1) Unutilized space & multiple mapping
Used in large systems.	Used in small systems. In small systems, the hlw for the decoding logic can be eliminated by using individual higher order address lines to select mly chips.

(2) Design an 8086 based minimum mode sln working at 6MHz having the following:
128 KB EPROM using 32 KB chips.
128 KB RAM using 64 KB chips.

RAM

Required = 128 KB

Available = 64 KB

No. of chips = $\frac{128}{64}$

= 2 chips

Size of single RAM chip

= 64 KB

= 64 × 1K

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

= 2^{16}

∴ 16 Address lines

(A1 – A16)

EPROM

Required = 128 KB

Available = 32 KB

No. of chips = $\frac{128}{32}$

= 4 chips

Size of single EPROM chip

= 32 KB

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

= 2^{15}

∴ 15 Address lines

(A1 – A15)

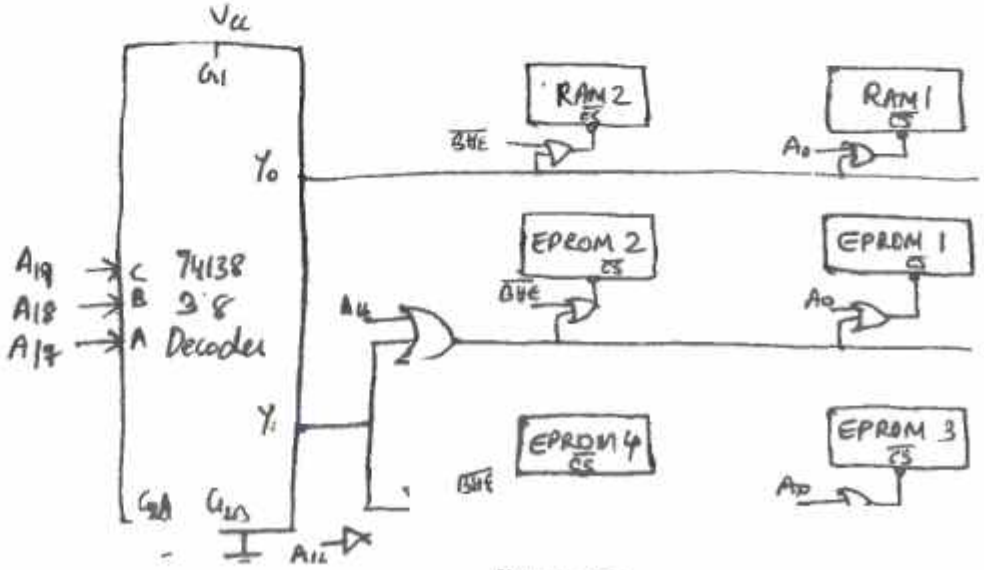
Starting Address

FFFFF

– 1FFFF

E0000

	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Address
RAM 1 Begin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
RAM 1 End	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1FFFEH
RAM 2 Begin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001H
RAM 2 End	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFFH
EPROM 1 Begin	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E0000H
EPROM 1 End	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	EFFFEH
EPROM 2 Begin	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	E0001H
EPROM 2 End	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	EFFFFH
EPROM 3 Begin	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F0000H
EPROM 3 End	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH
EPROM 4 Begin	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	F0001H
EPROM 4 End	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH



Subject: Microprocessor

Semester: IV

(3) Design an 8086 based minimum mode sm working at 6 MHz having the following: 64 KB EPROM using 16 KB chips.
256 KB RAM using 64 KB chips.

RAM

Required = 256 KB
Available = 64 KB
No of chips = $\frac{256}{64}$ = 4 chips.

Size of a single RAM chip = 64 KB
$= 2^6 \times 1K$
$= 2^6 \times 2^{10}$
$= 2^{16}$

∴ 16 Address lines ($A_1 - A_{16}$)

EPROM

Required = 64 KB
Available = 16 KB
No of chips = 64/16 = 4 chips.
Size of a single EPROM chip = 16 KB = $2^4 \times 2^{10}$
$= 2^{14}$

∴ 1 Address lines ($A_1 - A_{14}$)

Starting Address:

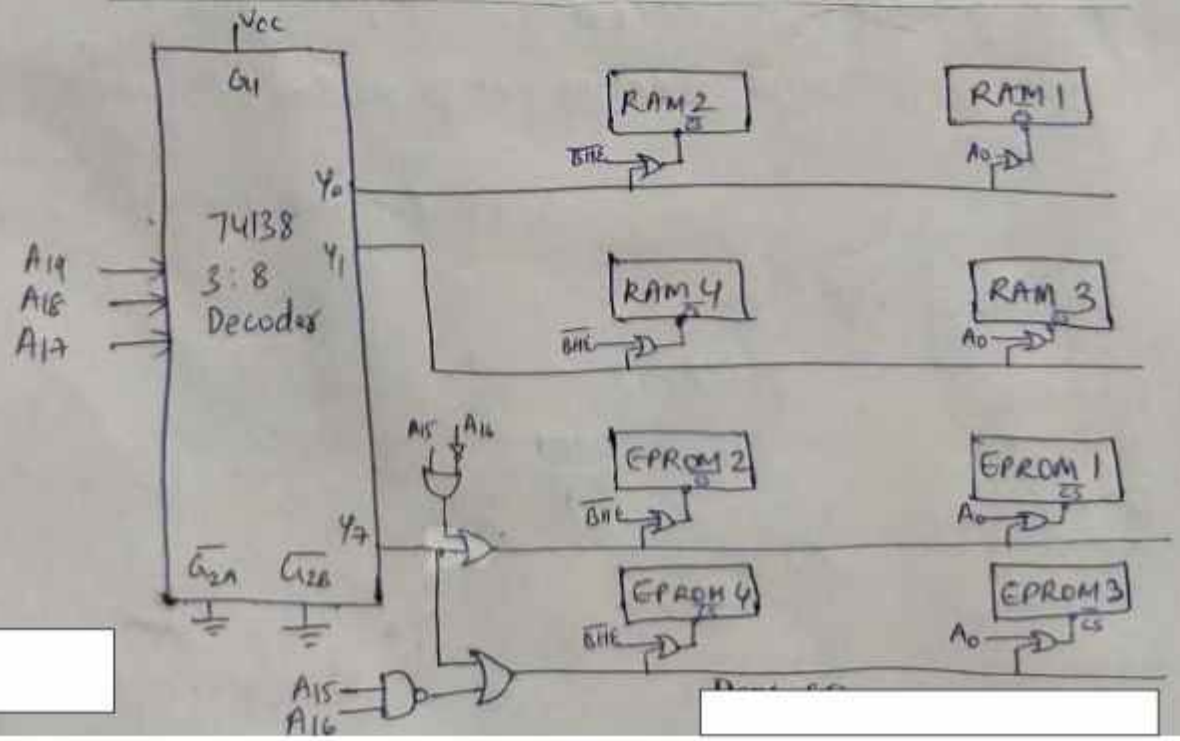
F FFFF
FFFF
F0000

64 KB = $2^6 \times 2^{10} = 2^{16}$

1111 1111 1111 1111
F F F F

Subject: Microprocessor **Semester: IV**

Memory CHIP	Address Bus 19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	Memory Address
RAM 1 Begin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
RAM 1 End	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1FFFE
RAM 2 Begin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001
RAM 2 End	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFF
RAM 3 Begin	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20000
RAM 3 End	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	3FFFE
RAM 4 Begin	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	20001
RAM 4 End	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFF
EPROM 1 Begin	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F0000
EPROM 1 End	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	F7FFE
EPROM 2 Begin	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	F0001
EPROM 2 End	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	F7FFF
EPROM 2 Begin	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F8000
EPROM 3 End	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFE
EPROM 4 Begin	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	F8001
EPROM 4 End	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFH



Subject: Microprocessor

Semester: IV

(4) Design an 8086 based minimum mode system working at 10 MHz having the following:
16 KB EPROM using 4 KB chips.
64 KB RAM using 8 KB chips

EPROM

Required = 16 KB
Available = 4 KB
No. of chips = $\frac{16\ KB}{4\ KB} = 4$.

Starting address of EPROM

$$\begin{array}{r} FFFFF \\ -\ 3FFF \\ \hline FC000 \end{array}$$

$16\ KB = 2^{14}$

0011 1111 1111 1111
3 F F F

Size of 1 EPROM chip = 4 KB = 2^{12}
∴ 12 Address lines (A1 – A12)

RAM

Required = 64 KB
Available = 8 KB
No. of chips = 8
Address lines:
Size of 1 RAM chip = 8 KB = 2^{13}

∴ 13 Address lines (A1 – A13)

Subject: Microprocessor

Semester: IV

Memory CHIP	Address Bus																				Memory Address
	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	
RAM 1 Begin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
RAM 1 End	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	03FFEH
RAM 2 Begin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001H
RAM 2 End	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	03FFFH
RAM 3 Begin	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	04000H
RAM 3 End	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	07FFEH
RAM 4 Begin	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	04001H
RAM 4 End	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	07FFFH
RAM 5 Begin	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	08000H
RAM 5 End	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0BFFEH
RAM 6 Begin	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	08001H
RAM 6 End	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0BFFFH
RAM 7 Begin	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0C000H
RAM 7 End	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0FFFEH
RAM 8 Begin	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0C001H
RAM 8 End	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0FFFFH
EPROM 1 Begin	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FC000H
EPROM 1 End	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	FDFFEH
EPROM 2 Begin	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	FC001H
EPROM 2 End	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	FDFFFH
EPROM 3 Begin	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	FE000H
EPROM 3 End	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH
EPROM 4 Begin	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	FE001H
EPROM 4 End	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH

Subject: Microprocessor

Semester: IV

Two Equally Correct Decoding Solutions