INTERFACING MEMORY CHIPS WITH 8086

- The 8086 microprocessor provides a 20 bit memory address that allows up to 1 MB MM.
- Out of these, several address lines are unused (i.e not connected to the memory chip)
- The extra lines determine the range of addresses the memory interface occupies.
- Address decoder circuit determines these extra address lines and enables the memory for a specific range of addresses.

Address Decoding Techniques in 8086 Microprocessor

- Depending up on number of lines used for decoder, we get
  - **Full Decoding (Absolute Decoding):** All the unused lines are used.
  - **Partial Decoding (Linear Decoding):** All the unused lines are not used.
  - **Block Decoding**

Absolute decoding

- In this technique the memory chip is selected only for the specified logic level on the address lines; no other logic levels can select the chip.

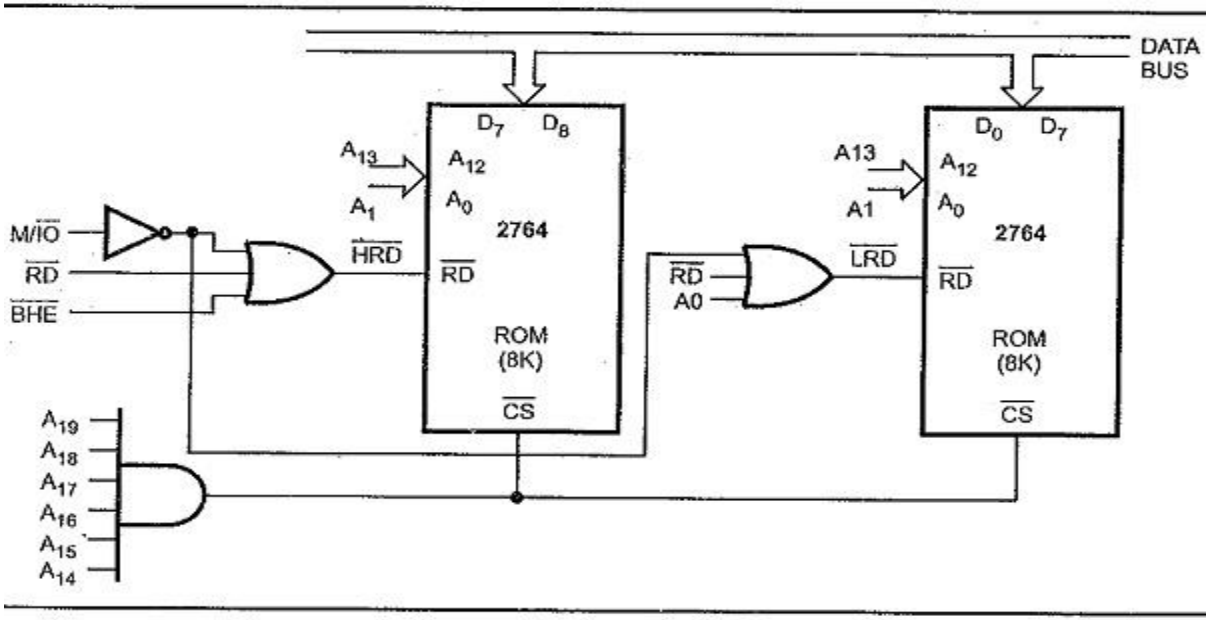


Fig. 10.12 Absolute decoding

Advantages of Absolute decoding

- Each Memory location has only one address, i.e there is no duplication in the address.
- Memory can be placed contiguously in the address space of the processor.
- Future expansion can be easily done without disturbing the existing circuitry.

Disadvantages:

- Extra decoders are necessary.
- Delay will be produced by these extra decoders.

Linear Decoding

- In small systems, hardware for the decoding logic can be eliminated, by using only required number of addressing lines (not all). Other lines are simply ignored.

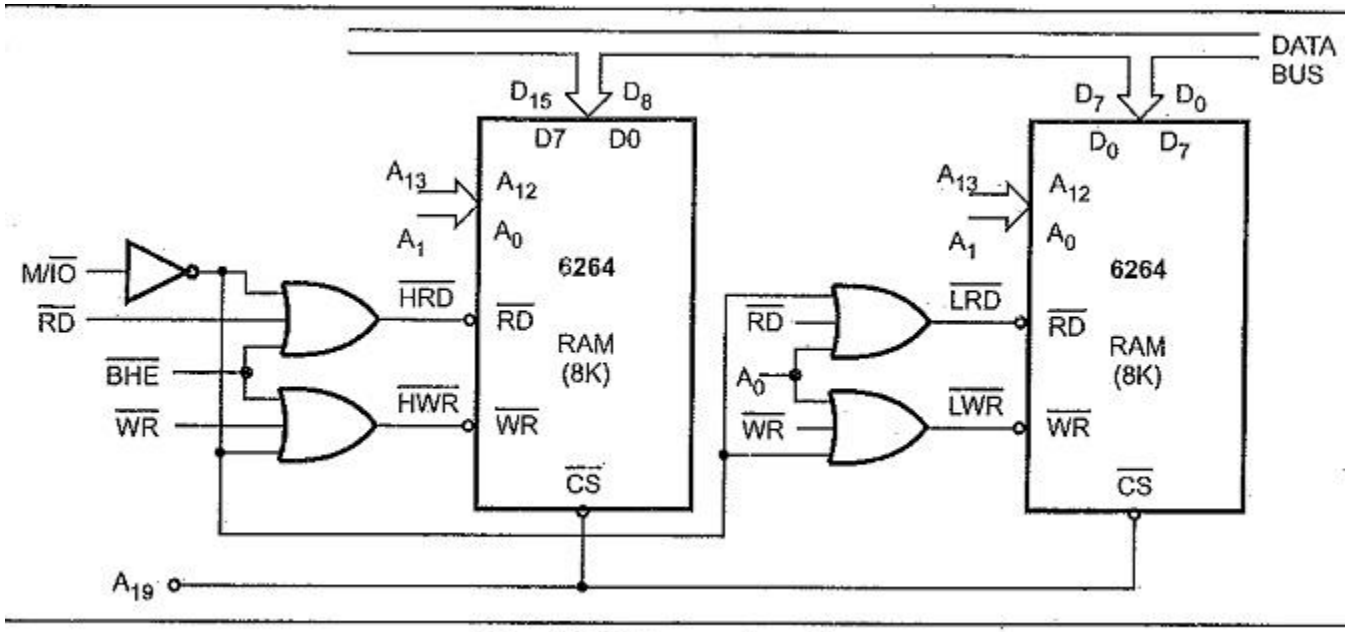


Fig. 10.13 Linear decoding

Adv and Disadv of Linear decoding

Simplified decoding circuit

Disadv:

- Multiple addresse are provided for the same location.
- Complete memory space is not efficiently used.
- Adding or interfacing ICs with already existing circuitry is difficult.

Block Decoding

- In a microcomputer system the memory array is often consists of several blocks of memory chips. Each block of memory requires decoding circuit. To avoid separate decoding for each memory block special decoder IC is used to generate chip select signal for each block.

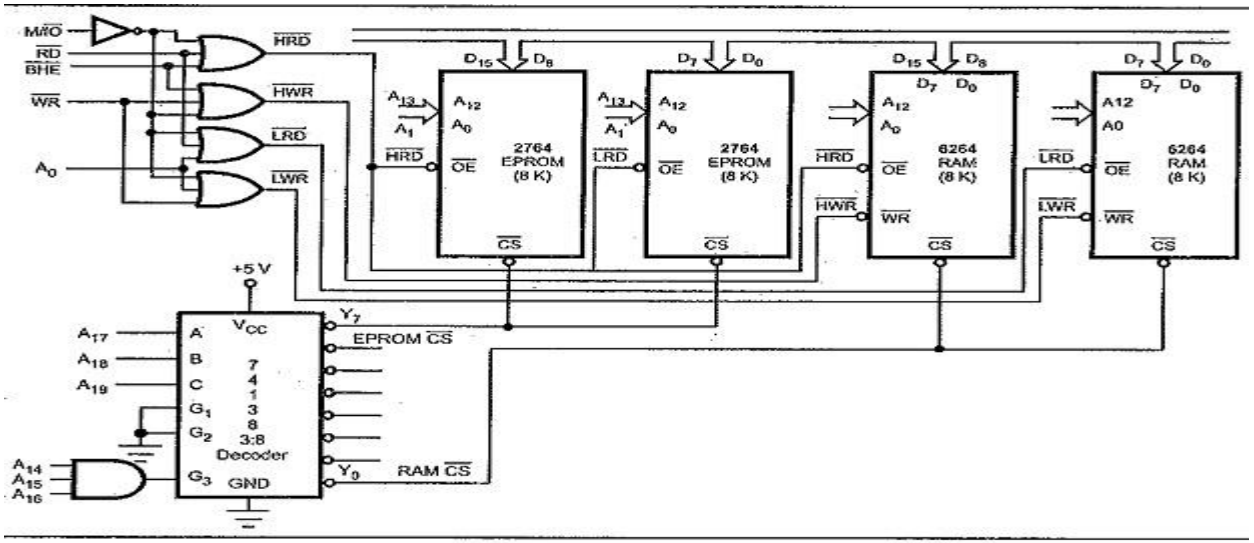


Fig. 10.14 Block decoding

- Arrange the available memory chips so as to obtain 16-bit data bus width. The upper 8-bit bank is called as the "odd address memory bank" and the lower 8-bit bank is referred to as the "even address memory bank".
- Now connect the available memory address lines of memory chips with those of the 8086 microprocessor and connect the memory RD and WR inputs to the corresponding processor control signals.

- Connection of the 16-bit data bus of the memory bank with that of the microprocessor is to be done.
- BHE , A0 and the rest of the address lines left are used for decoding the required chip select signals for the odd and even memory banks.

Steps for interfacing of 8086 with SRAM and ROM

1. Calculate the number of memory devices(IC Required).

– Num. of mem. Devices= $\frac{\text{Reqd. Memory size}}{\text{Size of Mem. Chip.}}$

2. Arrange the given memory chips as even and odd memory bank so as to obtain 16- Bit data bus width.

3. Determine the address lines of 8086 microprocessor to be connected directly to the address pins of memory devices.

4. Since two memory banks are to be interfaced, so that two decoders are required one for each bank.
   - The M/IO’ and A0 pins are used to enable the even bank decoder
   - The M/IO’ and BHE’ pins are used to enable the odd bank decoder.

EPROM Interfacing with 8086

- Whenever the 8086 CPU is reset, its value is set to FFFFH and IP value is set to 0000H that corresponds to physical address FFFF0H which is always a part of ROM. This means FFFF0H to FFFFFH should be always included in the ROM.

Example 1

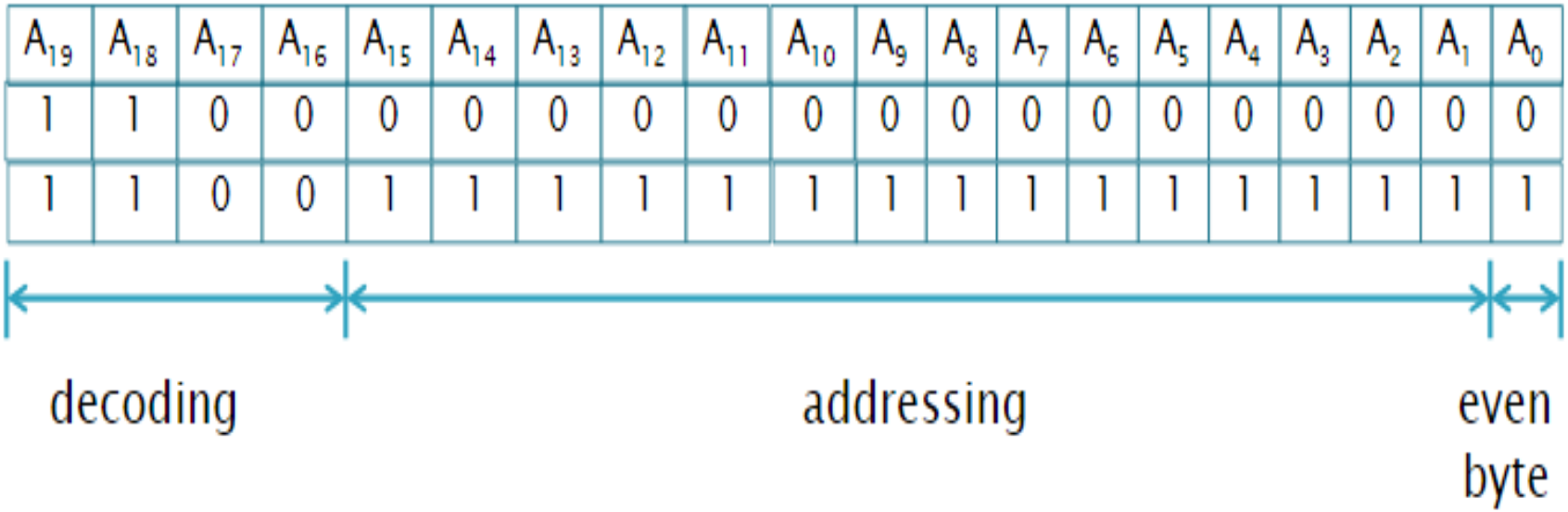
Interface a 64KB RAM module to an 8086 processor starting at address C0000H.

Soln: 64 KB RAM can be split into 2 banks of 32 KB each.

Hence each memory chip can be addressed using 15 address lines.

The address bus of the 8086 (A19 -A0) will be interfaced to each memory bank as follows:

- The least significant address lines A 15 -A 1 are used to address memory locations in memory chip
- The most significant address lines A 19 -A 16 are used to select the memory module
- The address line A 0 and BHE’ signal are used to select even byte, odd byte or both

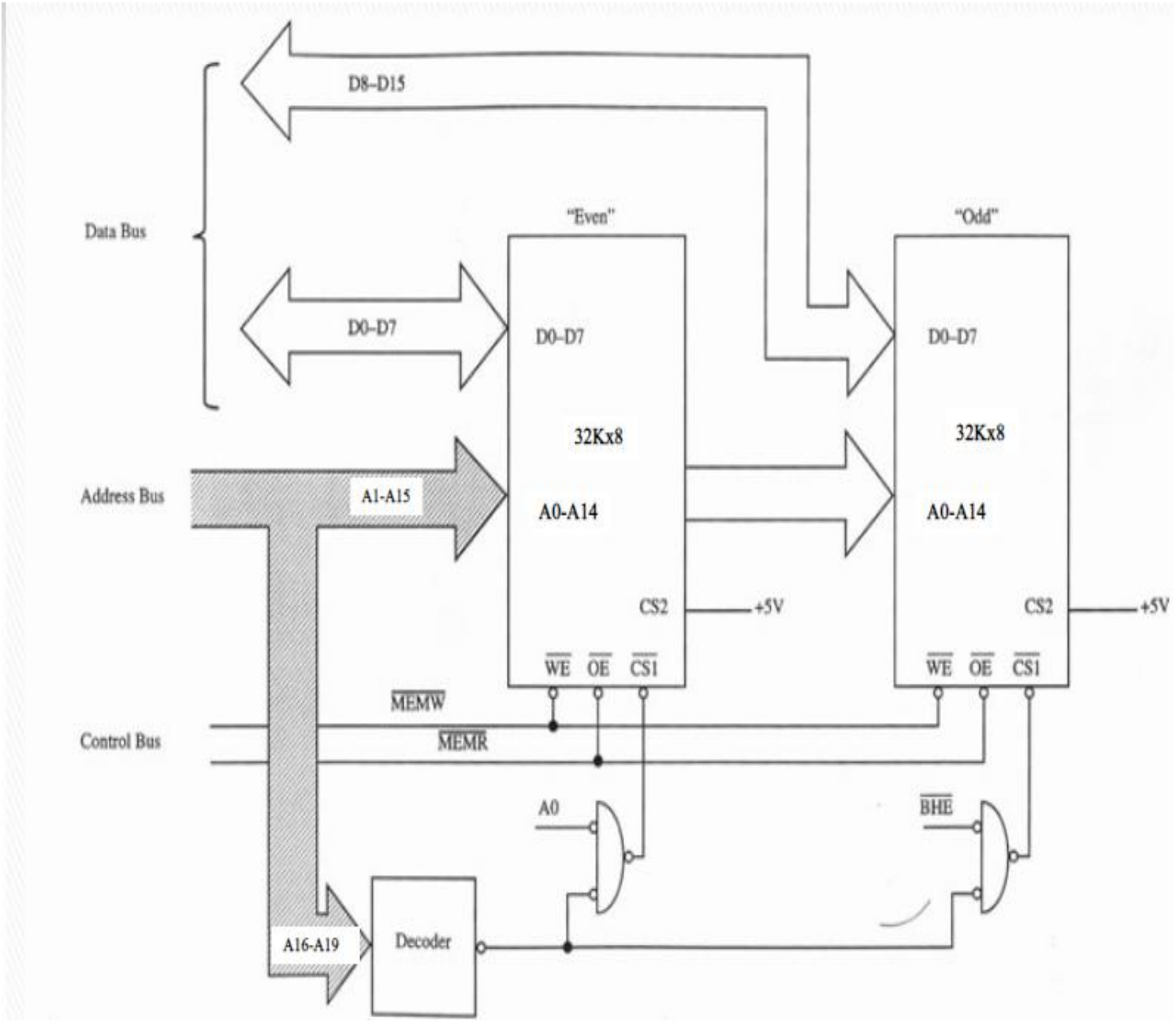

A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
decoding
addressing
even byte

- The control-bus of the 8086 is interfaced to the SRAM as follows:

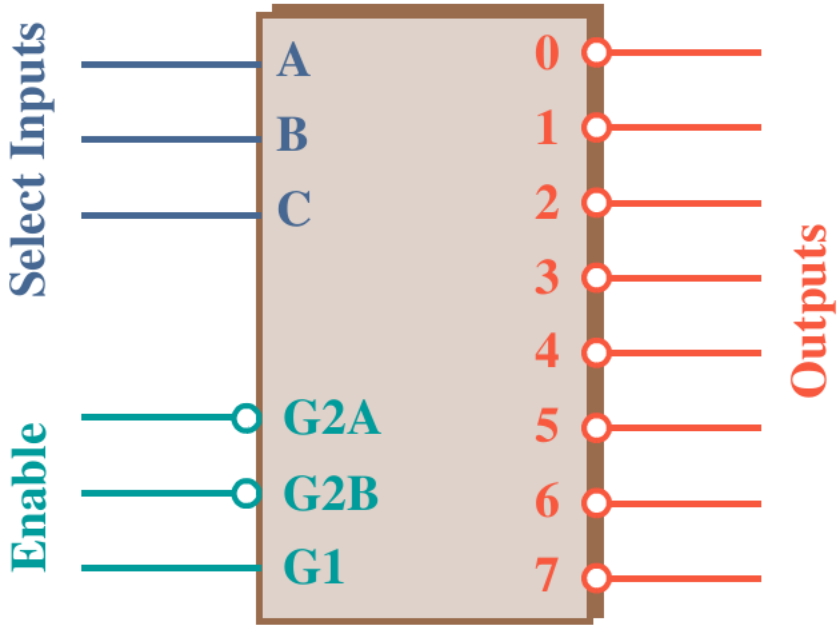
  Decode the M/IO', RD', and WR' signals to generate 4 signals Memory Read (MEMR'), Memory Write (MEMW'), I/O Read (IOR') and I/O Write (IOW').

  Connect MEMR' to OE' and MEMW' to WE'

- The data-bus of the 8086 is interfaced to the SRAM by connecting:
  - The buffered data lines D 7 -D 0 to input/output lines (D 7 -D0) of the even SRAM.
  - The buffered data lines D 15 -D 8 to input/output lines (D 7 -D0) of the odd SRAM.


D8–D15
Data Bus
"Even"
"Odd"
D0–D7
D0–D7
D0–D7
32Kx8
32Kx8
Address Bus
A1-A15
A0-A14
A0-A14
CS2
+5V
CS2
+5V
WE
OE
CS1
WE
OE
CS1
MEMW
Control Bus
MEMR
A0
BHE
A16-A19
Decoder

74LS138 Decoder



Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	**0**	1	1	1	1	1	1	1
0	0	1	0	0	1	1	**0**	1	1	1	1	1	1
0	0	1	0	1	0	1	1	**0**	1	1	1	1	1
0	0	1	0	1	1	1	1	1	**0**	1	1	1	1
0	0	1	1	0	0	1	1	1	1	**0**	1	1	1
0	0	1	1	0	1	1	1	1	1	1	**0**	1	1
0	0	1	1	1	0	1	1	1	1	1	1	**0**	1
0	0	1	1	1	1	1	1	1	1	1	1	1	**0**

Example 2

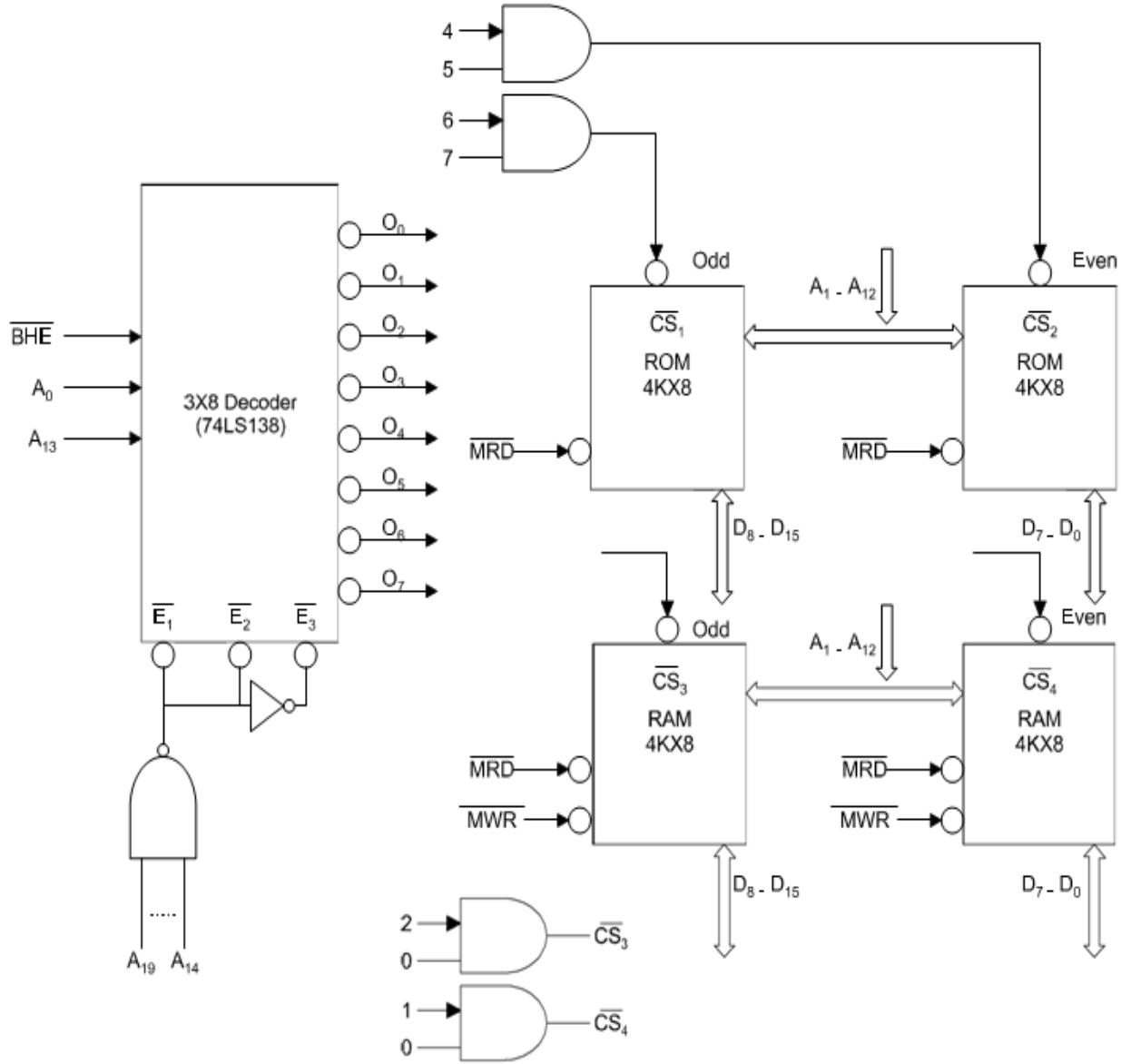
- **Interface two 4K X 8 EPROMs and two 4K X 8 RAM chips with 8086, select suitable maps.**

- Since after reset, the IP and CS are initiated to form the address FFFF0H and this address must lie in the EPROM.
- The address of RAM may be selected anywhere in the 1MB space of 8086, but we will select the RAM address such that the address map of the system is continuous.

- Total 8K of EPROM/RAM need 13 address lines A0-A12 (Since 2^{13} = 8K).
- Address lines A13-A19 are used for decoding to generate the chip select.
- The BHE' signal goes low when a transfer is at odd address or higher byte of data is to be accessed.

Address	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	EPROM						8K X 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 X 8													
FC000H	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

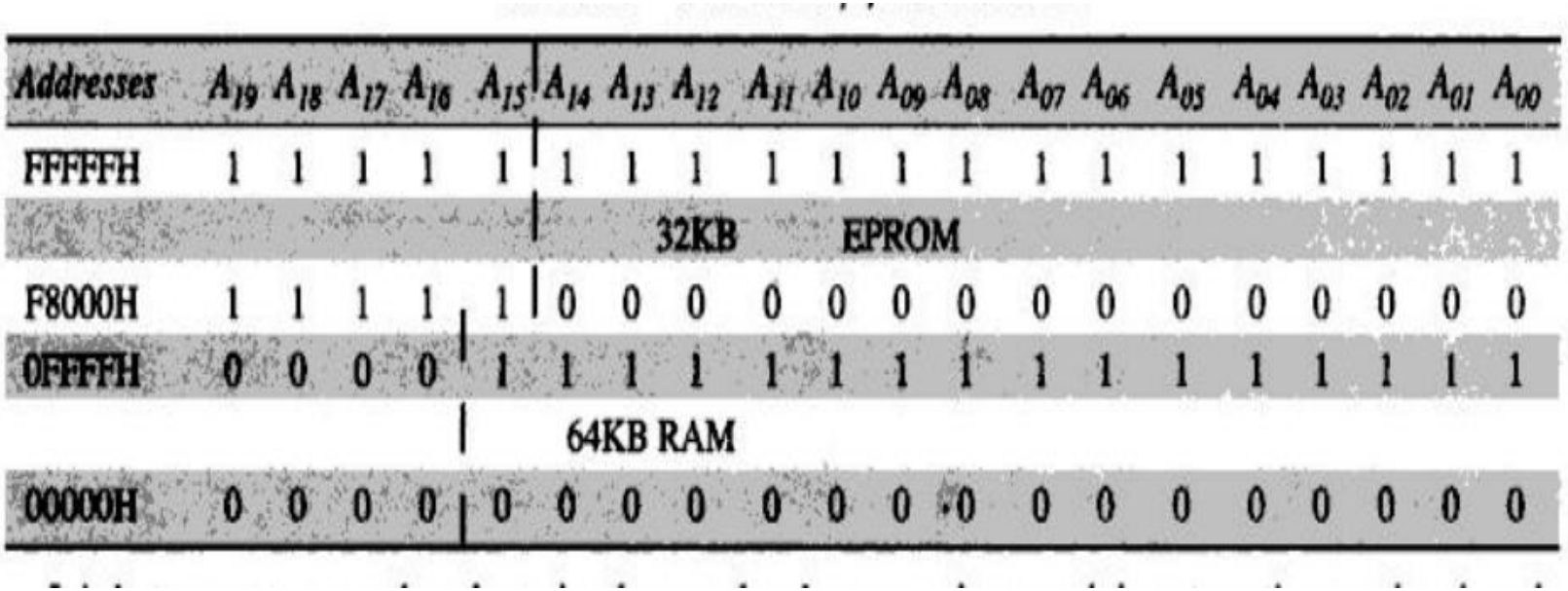
Decoder I/P Address / BHE'	**A2**	**A1**	**A0**	**Selection / Comment**
	A13	**A0**	**BHE'**	
Word transfer on D0-D15	0	0	0	Even and Odd address in RAM
Byte transfer on D0-D7	0	0	1	Only even address in RAM
Byte transfer on D8-D15	0	1	0	Only odd address in RAM
Word transfer on D0-D15	1	0	0	Even and Odd address in ROM
Byte transfer on D0-D7	1	0	1	Only even address in ROM
Byte transfer on D8-D15	1	1	0	Only odd address in ROM


4
5
6
7
Odd
Even
A_1 - A_{12}
$\overline{CS_1}$
ROM
4KX8
$\overline{CS_2}$
ROM
4KX8
$\overline{MRD}$
$\overline{MRD}$
D_8 - D_{15}
D_7 - D_0
O_0
O_1
O_2
O_3
O_4
O_5
O_6
O_7
$\overline{BHE}$
A_0
A_{13}
3X8 Decoder
(74LS138)
$\overline{E_1}$
$\overline{E_2}$
$\overline{E_3}$
Odd
Even
A_1 - A_{12}
$\overline{CS_3}$
RAM
4KX8
$\overline{CS_4}$
RAM
4KX8
$\overline{MRD}$
$\overline{MWR}$
$\overline{MRD}$
$\overline{MWR}$
D_8 - D_{15}
D_7 - D_0
2
0
$\overline{CS_3}$
1
0
$\overline{CS_4}$
.....
A_{19}
A_{14}

Example 3

- Design an interface between 8086 CPU and two chips of 16KB EPROM and two chips of 32KB RAM. Select the starting address of EPROM suitably. The RAM address must start at 00000H.

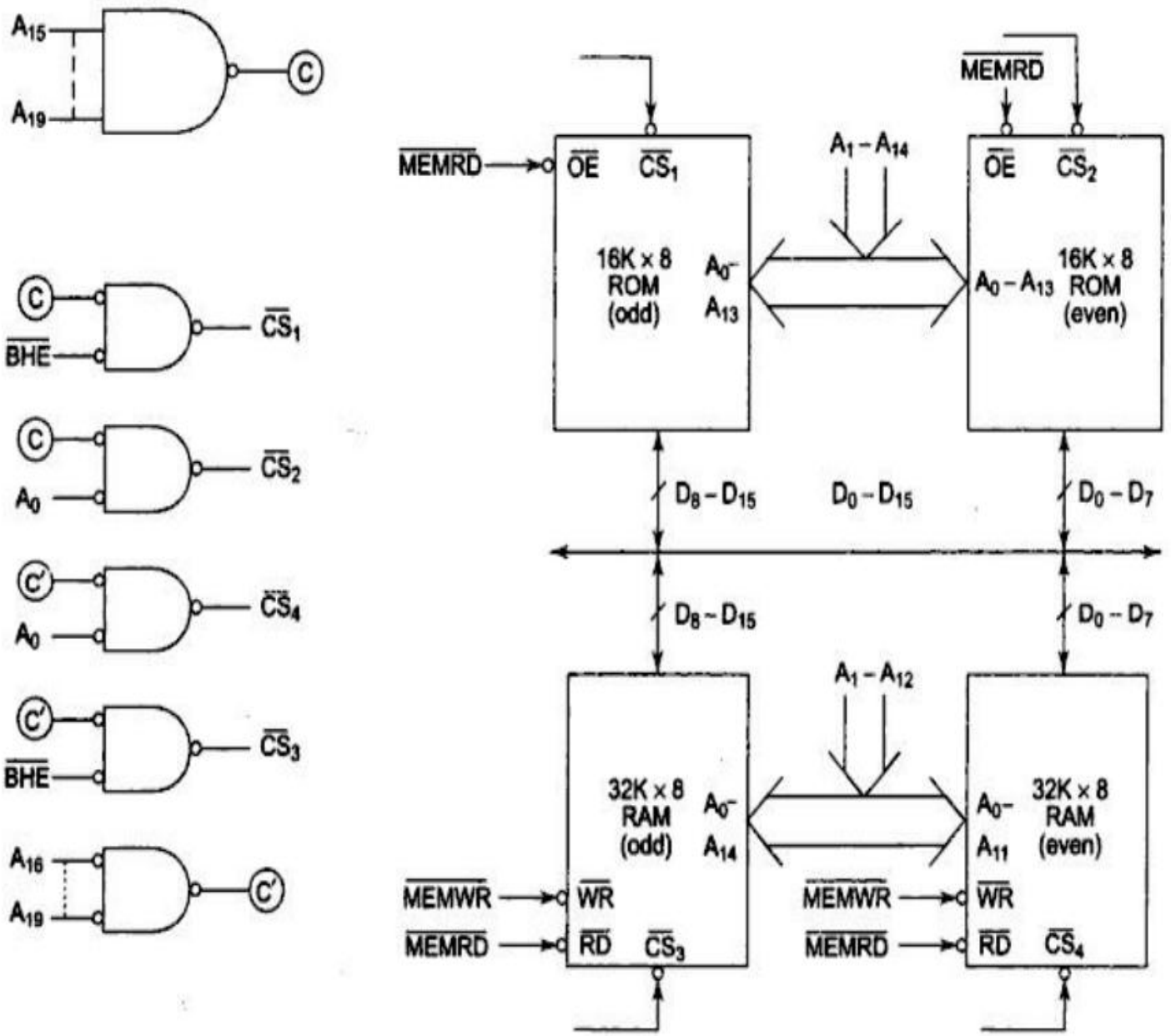
Address Map of EPROM and RAM

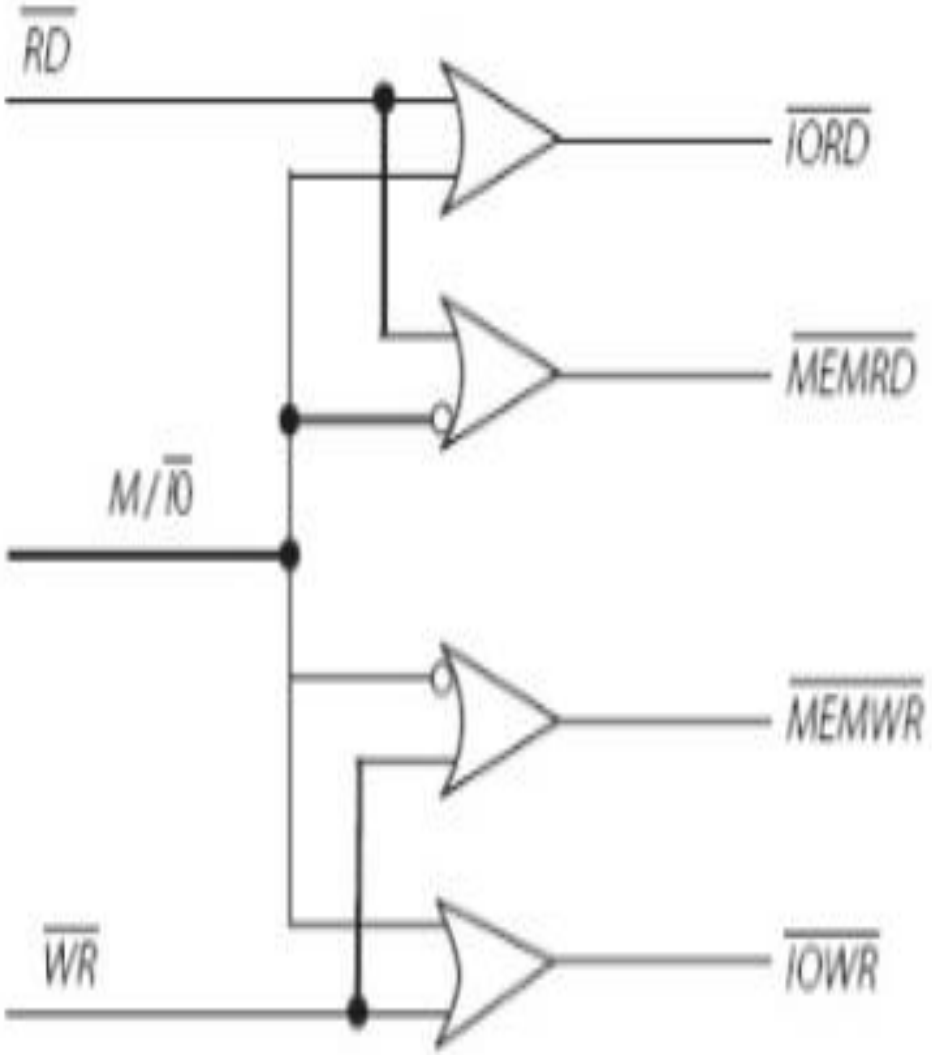


Addresses	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_{09}	A_{08}	A_{07}	A_{06}	A_{05}	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}
FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						32KB EPROM														
F8000H	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0FFFFH	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
						64KB RAM														
00000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Last address of EPROM should be FFFFFH. Since after resetting the processor starts from FFFF0H.

Here decoder is not preferred as the addresses are not continuous. Hence it can be implemented using logic gates.


A15
A19
C
C
BHE
CS1
C
A0
CS2
C'
A0
CS4
C'
BHE
CS3
A16
A19
C'
MEMRD
OE
CS1
A1 – A14
MEMRD
OE
CS2
16K x 8
ROM
(odd)
A0 –
A13
A0 – A13
16K x 8
ROM
(even)
D8 – D15
D0 – D15
D0 – D7
D8 – D15
D0 – D7
A1 – A12
32K x 8
RAM
(odd)
A0 –
A14
A0 –
A11
32K x 8
RAM
(even)
MEMWR
MEMRD
WR
RD
CS3
MEMWR
MEMRD
WR
RD
CS4


RD
IORD
MEMRD
M/IO
MEMWR
WR
IOWR

Example 4

Design a 8086 based system with following specifications

- CPU at 10MHz in minimum mode operation
- 32 KB SRAM using 8 KB devices
- 64 KB EPROM using 16 KB devices

Step 1: Total EPROM required = 64 KB

Chip size available = 16 KB

∴Number of chips required=64KB/16KB=4

∴Number of sets required=Numberofchips/Numberofbanks=4/2=2

SET 1: Ending address of SET 1 = FFFFFH

SET size = Chip size x 2 = 16 KB x 2 = 32 KB

i.e. 0000 0111 1111 1111 1111 =07FFFH

Starting address = Ending address – SET size=FFFFFH–07FFFH=F8000H

SET 2: Ending address of SET 2 = F7FFFH (previous ending - 1)

SET size = Chip size x 2 = 16 KB x 2 = 32 KB

i.e. 0000 0111 1111 1111 1111=07FFFH

Starting address = Ending address – SET size=F7FFFH–07FFFH=F0000H

		Even bank	Odd bank
ROM SET 1	Starting Address	F8000H	F8001H
	Ending Address	FFFFEH	FFFFFH
ROM SET 2	Starting Address	F0000H	F0001H
	Ending Address	F7FFEH	F7FFFH

Step 2: Total RAM required = 32 KB

Chip size available = 8 KB

∴Number of chips required=32KB/8KB=4

∴Number of sets required=Numberofchips/Numberofbanks=4/2=2

SET 1: Starting address = 00000H

SET size = Chip size x 2 = 8 KB x 2 = 16 KB

i.e. 0000 0011 1111 1111 1111=03FFFH

Ending address = Starting address +SET size=00000H+03FFFH=03FFFH

SET 2: Starting address = 04000H (previous ending +1)

SET size = Chip size x 2 = 8 KB x 2 = 16 KB

i.e. 0000 0011 1111 1111 1111=03FFFH

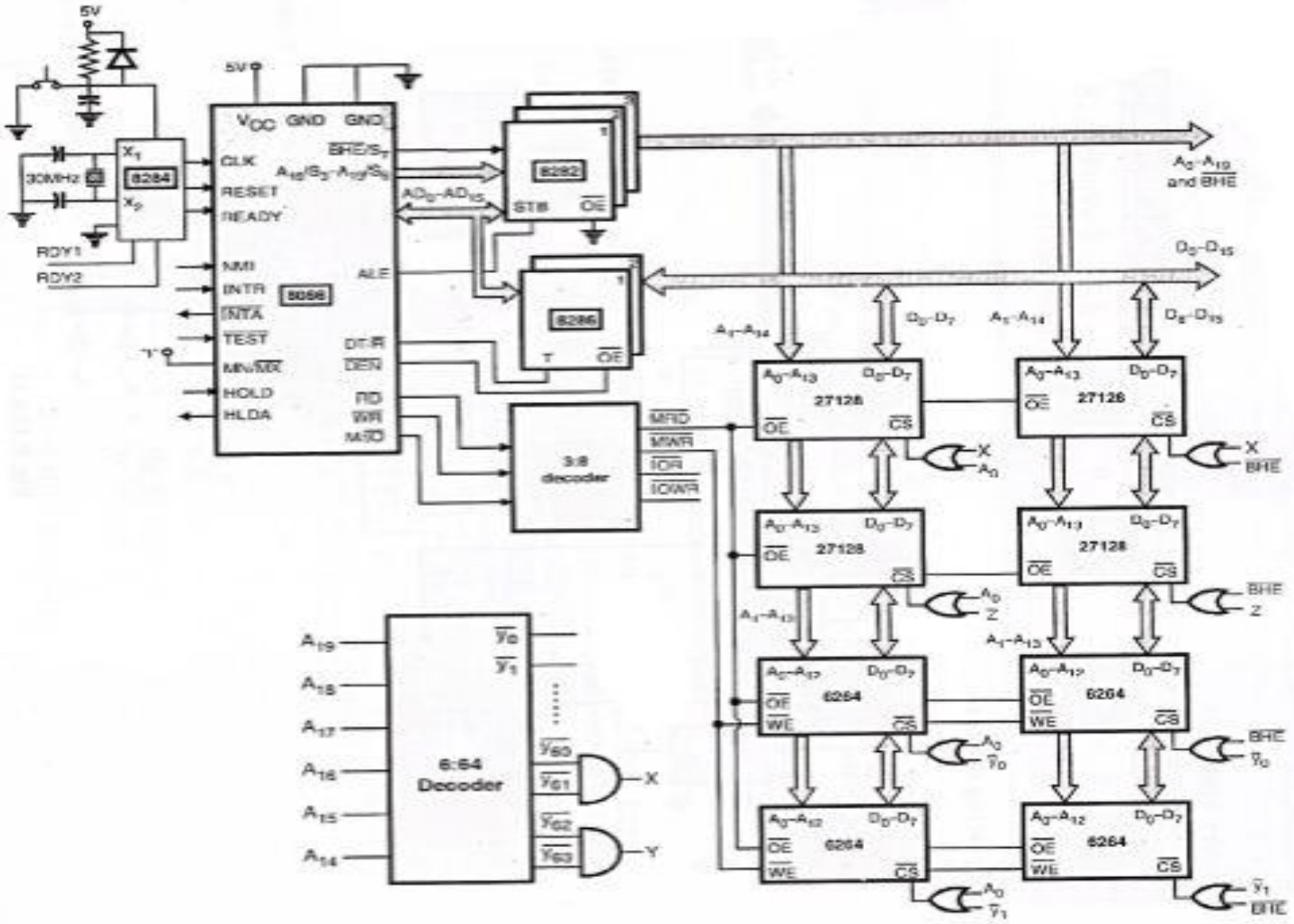
Ending address = Starting address +SET size=04000H+03FFFH=07FFFH

		Even bank	Odd bank
RAM SET 1	Starting Address	00000H	00001H
	Ending Address	07FFEH	07FFFH
RAM SET 2	Starting Address	04000H	04001H
	Ending Address	07FFEH	07FFFH

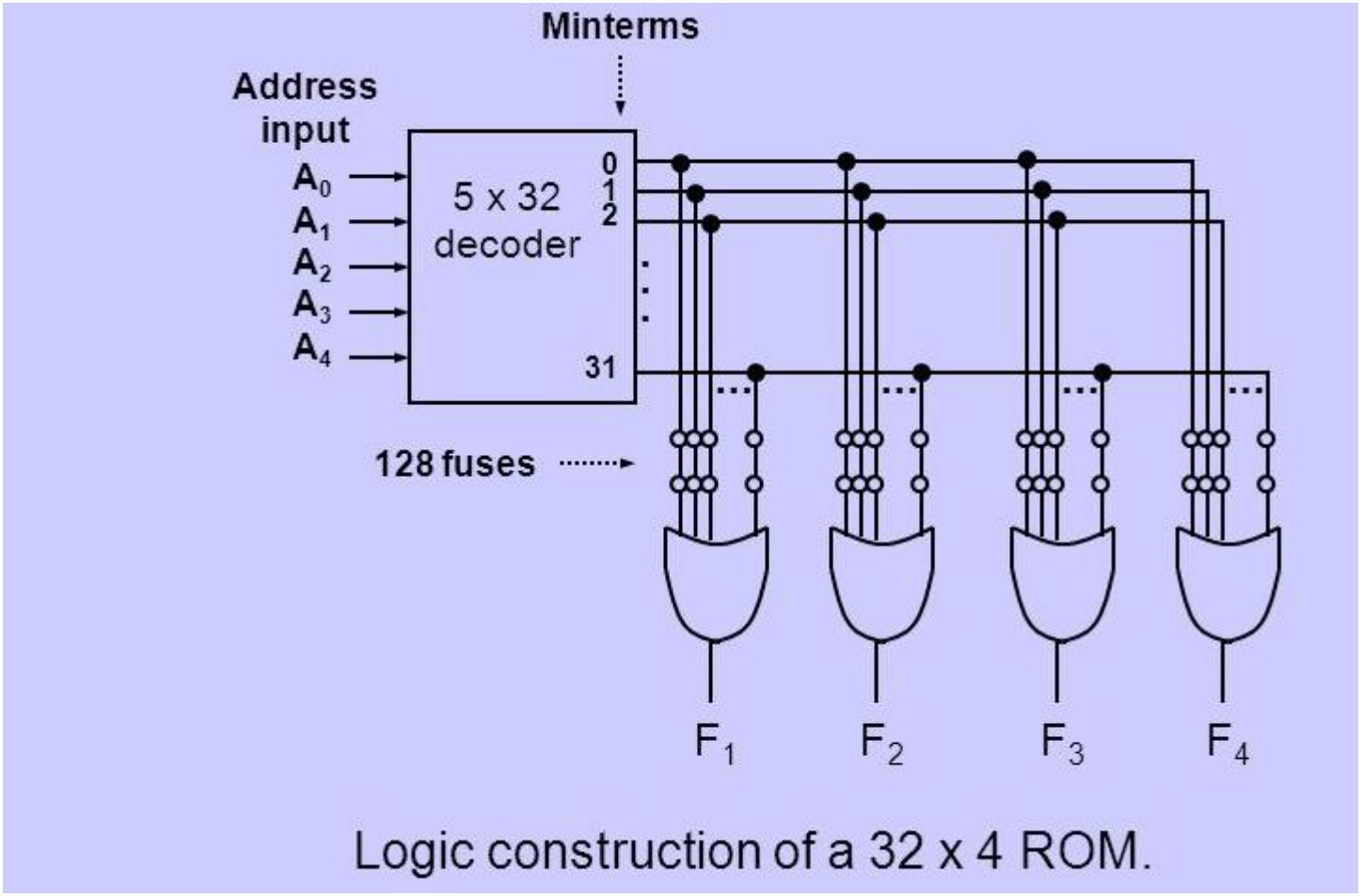
Step 3 : Memory Map :

EA ⇒ Ending address, SA ⇒ Starting address, EB ⇒ Even Bank, OB ⇒ Odd Bank

			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
RAM Set-1 ($\overline{y}_0$)	EB	SA = 00000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		EA = 03FFEH	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	OB	SA = 00001H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		EA = 03FFFH	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RAM Set - 2 ($\overline{y}_1$)	EB	SA = 04000H	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		EA = 07FFEH	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	OB	SA = 04001H	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		EA = 07FFFH	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM Set - 2 ($\overline{y}_{60}$ - $\overline{y}_{61}$)	EB	SA = F0000H	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		EA = F7FFEH	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	OB	SA = F0001H	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		EA = F7FFFH	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM Set - 1 ($\overline{y}_{62}$ - $\overline{y}_{63}$)	EB	SA = F8000H	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		EA = FFFFEH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	OB	SA = F8001H	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
		EA = FFFFFH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1


5V
30MHz
8284
RDY1
RDY2
8086
8282
8286
3:8 decoder
27128
6264
6:64 Decoder
MRD
MWR
IOR
IOWR
RD
WR
M/IO
DT/R
DEN
ALE
NMI
INTR
INTA
TEST
HOLD
HLDA
READY
RESET
CLK
STB
OE
CS
WE

Decoding using PROM



Logic construction of a 32 x 4 ROM.

Advantages of using PROM as decoder

- PROM are programmable.(i.e the memory address can be moved to new addresses by simply programming a new PROM).
- Large number of inputs on the PROM allows us to select a specific area of memory without using external gates.

Program control Instructions

Mnemonics	Explanation
STC	Set CF ← 1
CLC	Clear CF ← 0
CMC	Complement carry CF ← CF/
STD	Set direction flag DF ← 1
CLD	Clear direction flag DF ← 0
STI	Set interrupt enable flag IF ← 1
CLI	Clear interrupt enable flag IF ← 0
NOP	No operation
HLT	Stops the execution of software.To exit from halt cane be done either from interrupt/hardware reset.
WAIT	Wait for TEST pin active
ESC opcode mem/ reg	Used to pass instruction to a coprocessor which shares the address and data bus with the 8086
LOCK	Lock bus during next instruction

ORG and EVEN directives

- **ORG (Origin) : is used to assign the starting address (Effective address) for a program/ data segment**
- **Even: Informs the assembler to store program/ data segment starting from an even address**