Computer Interface
Part 1: Memory

1. **What types of connections are common to all memory devices?**

1- Address connections (Address Lines):

- used to select a memory location within the device
- Always labeled from A_0 (least significant address input), to A_n (Most significant address input)
- determined by the number of memory locations found within it

2- **Data connections**.

- Are points at which data are entered for storage (WRITE) or extracted for (READ).
- are labeled D_0 through D_n

3- Control connections.

- Are inputs that selects or enables the memory device like CS or CE.

4- Selection connections:

- allow data to move from memory to CPU and form CPU to Memory
- Like RE or WE or OE

2. **What is the purpose of CS and CE pin on memory?**

These pins are used to select or enable device so that it can perform read or write operation.

3. **What is the purpose of OE pin on memory devices?**

The OE enables or disable a set of tri-state buffers located within memory devices & must be active to read data

4. **What is the purpose of WE pin on a RAM?**

It causes the memory to perform write operation provided the CS or CE pin is also active

5. **For the following EPROM devices: (i) 2708 (ii) 2716 (iii) 2732 (iv)2764 Determine:** Capacity – Number of address lines – Number of data lines.

6. **List the number of words in bytes or KB found in each memory device for the following no of address connections?**

8	256
11	1024
12	4096
13	8192

1. **List number of data items stored in each of the following memory devices and number of bits in each datum?**

2K*4	2K memory location and 4 bits are stored in each datum.
1K*1	1K memory locations and 1 bit stored in each datum
4K*8	4K memory location and 8 bits stored in each datum
16K*1	16K memory location and 1 bit stored in each datum.
64K*4	64K memory location and 4 bits stored in each datum.

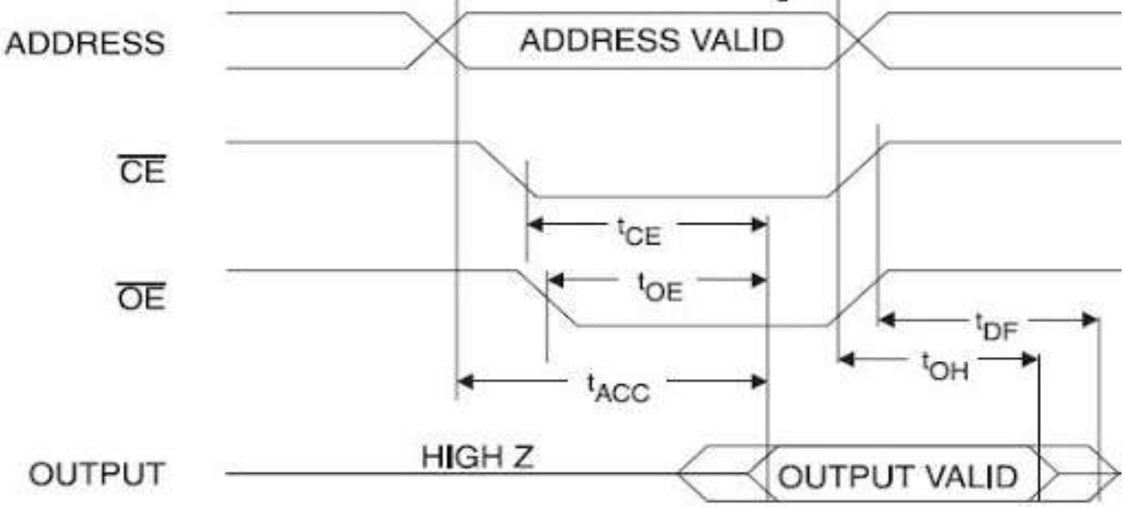
2. How many bits of storage do the following EPROM memory devices contain?

2708	(1K *8) =1024 bytes
2716	(2K*8)=2048 bytes
2732	(4K*8)=4096 bytes
2764	(8K*8)=8192 bytes
27128	(16k*8)=16384 bytes

7. Specify the number of necessary address lines, data items, and the total number of memory cells in 32 M X 64 memory chip.

8. The memory address of the last location of a 2k-byte memory chip is DFFF H. find the starting address?

9. Using the following trimming diagram discuss how the memory read is performed?



10. What is the different Type of Memory does a memory hierarchy have? And what are the main parameters that characterize each type?

A computer have several types of memory, ranging from fast, expensive internal registers, to slow, inexpensive Hard Magnetic Disks.

- Registers:

  Are matched in speed to the CPU
  Consume a significant amount of power.
  Only a small number of registers in a processor
  More expensive.

- Secondary storage:

  Such as hard magnetic disks.
  The cost per stored bit is small in terms of money and electrical power.
  Access time is very long when compared with registers.

- Between the registers and secondary storage there are a number of other forms of memory that bridge the gap between the two.

11. The 4016 memory has a G pin as S pin and W pin what are these pins used for RAM?

G pin work as OE and S pin as CS pin and W pin as WE

12. How much memory access time is required by the slowest 4016?

250 ns

13. **How much time is required to refresh a typical DRAM?**

2-4 ms

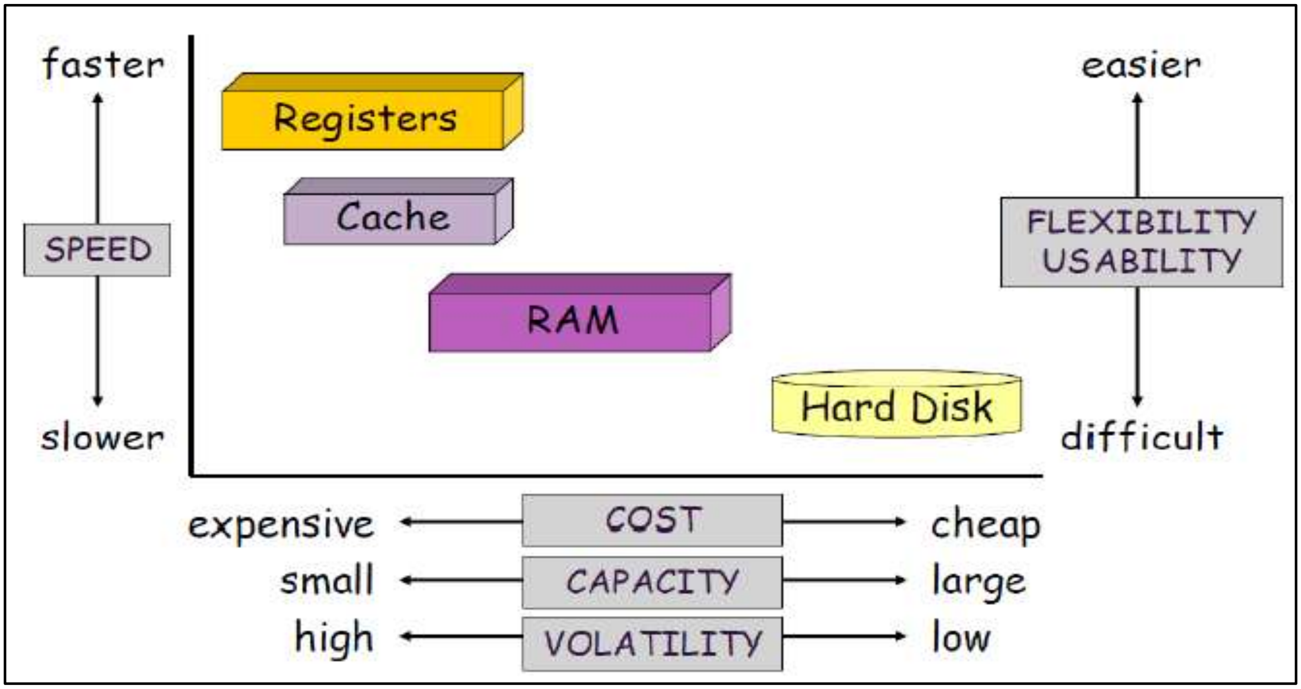
14. **How much time is allocated for memory access when the 8088/8086 microprocessor are operated with 5 MHz clock?**

460 ns

15. **What is the purpose of BHE and A0 pins on 8086 microprocessor?**

The BHE select the upper memory bank and A0 select lower memory bank.

16. **Draw a diagram that indicates the memory hierarchy in a computer system and show the effect of these parameters on each memory level?**

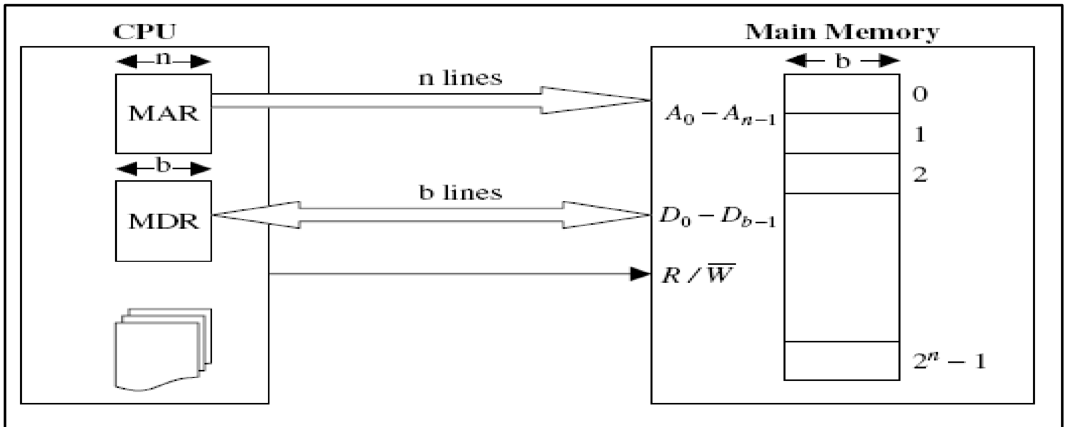


17. **Define Each of the following:**
Hit – Miss – Hit rate – Miss Rate – Hit Time

- **Hit:** The requested data resides in a given level of memory.
- **Miss:** The requested data is not found in the given level of memory.
- **Hit rate:** The percentage of memory accesses found in a given level of memory.
- **Miss rate:** The percentage of memory accesses not found in a given level of memory**.**

  Miss Rate = 1 - Hit Rate.
- **Hit time:** The time required to access the requested information in a given level of memory.

18. Draw a block diagram that represent the basic interface between the CPU and main memory?

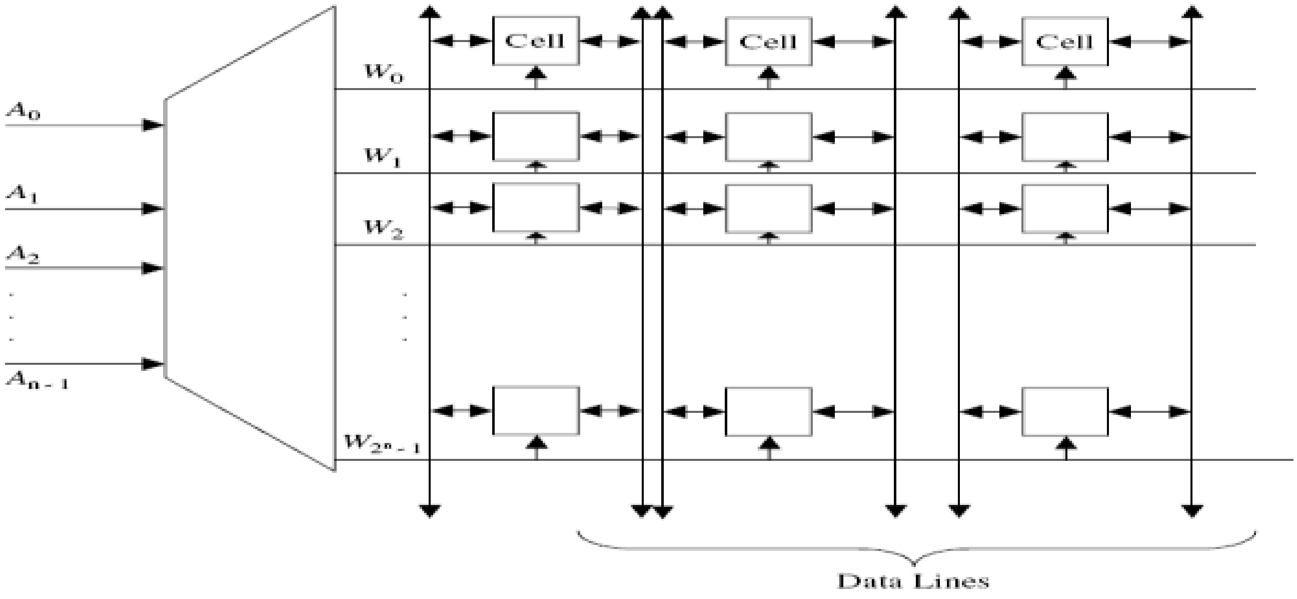


A typical CPU and main memory interface

19. Describe the internal organization of a memory chip? And draw block diagram that represent the internal organization of a memory chip?

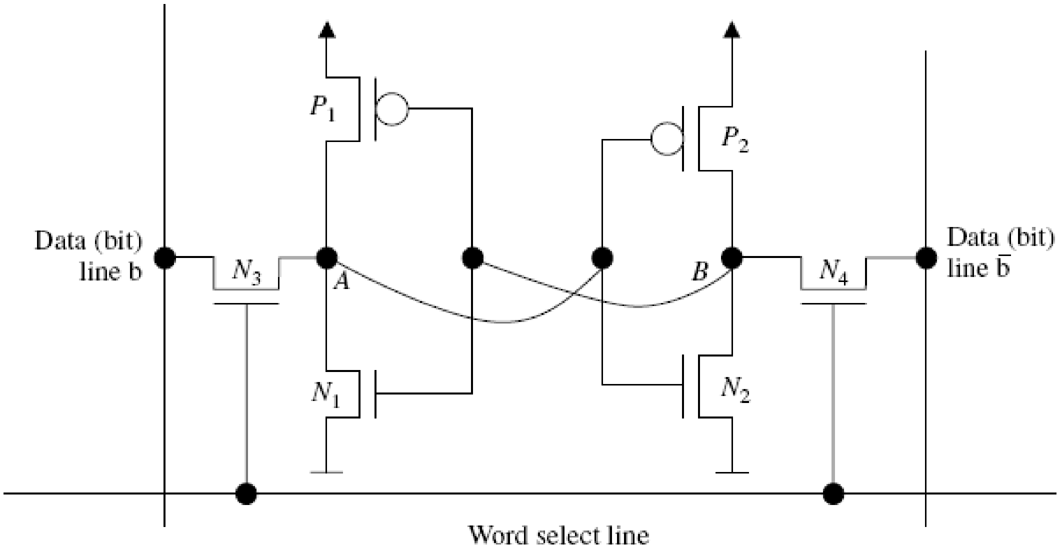
Main memory consists of:

- Rows and columns of basic cells that store one bit of information.
- Cells in one row can be used to form a memory word.
- Address lines $A_{n-1}A_{n-2}$ . . A_1A_0 are used as inputs to the address decoder in order to generate the word select lines $W_{2^{n}-1}$ . . .W_1W_0.
- A given word select line is common to all memory cells in the same row and is used to enable all cells in a row for read or write.
- Data (bit) lines are used to input or output the contents of cells. Each memory cell is connected to two data lines. A given data line is common to all cells in a given column.

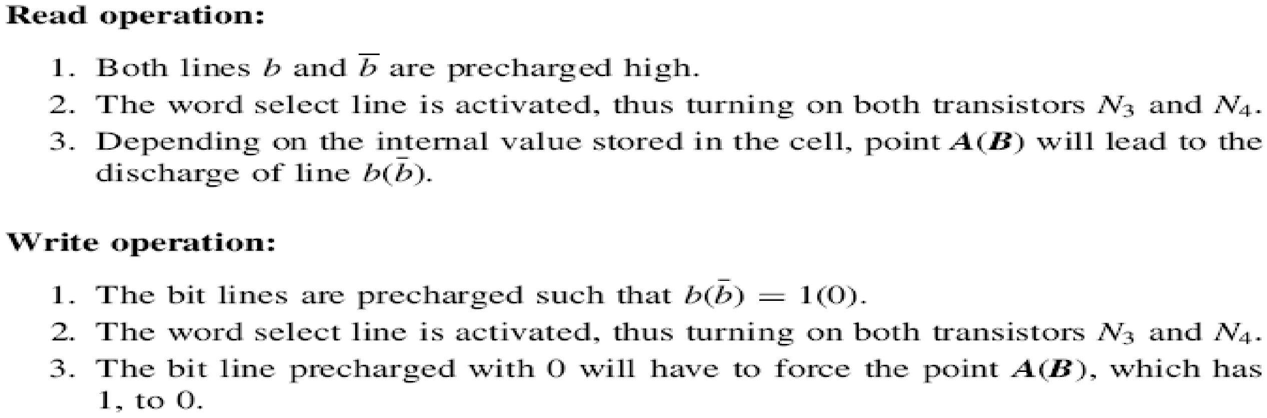


A conceptual internal organization of a memory chip

20. **Implement a static main memory cell using CMOS transistors? And describe how READ and WRITE operations are performed?**



Static CMOS memory cell



Read operation:

1. Both lines b and $\overline{b}$ are precharged high.
2. The word select line is activated, thus turning on both transistors N_3 and N_4.
3. Depending on the internal value stored in the cell, point $\boldsymbol{A}(\boldsymbol{B})$ will lead to the discharge of line $b(\overline{b})$.

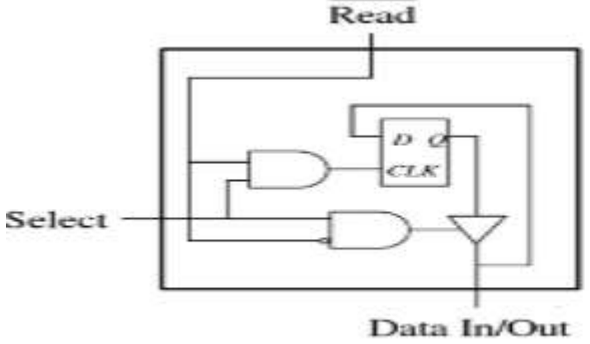
Write operation:

1. The bit lines are precharged such that $b(\bar{b}) = 1(0)$.
2. The word select line is activated, thus turning on both transistors N_3 and N_4.
3. The bit line precharged with 0 will have to force the point $\boldsymbol{A}(\boldsymbol{B})$, which has 1, to 0.

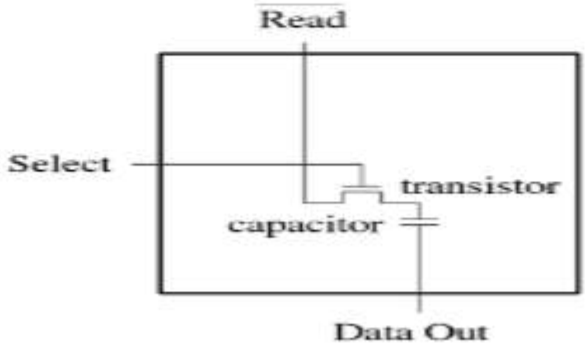
21. **Specify the number of address connections, data connections, the data items, and the total number of memory cells in 8192 X 8 memory chip, and if the end address is 9FFFH, find the start address.**

22. **compare between the following: SRAM – DRAM**

- **SRAM (Static RAM):** Is a RAM chips that are based upon flip-flops, because the contents of each location persist as long as power is applied to the chips.
- **DRAM (Dynamic RAM) chips**,
  - Employ a capacitor, which stores a minute amount of electric charge, in which the charge level represents a 1 or a 0.
  - Capacitors are much smaller than flip-flops, so a capacitor based DRAM can hold much more information in the same area than an SRAM.
  - Since the charges on the capacitors dissipate with time, the charge in the capacitor storage cells in DRAMs must be restored, or refreshed frequently.



Static RAM cell



dynamic RAM cell

23. SRAM is acronym for what type of device?

SRAM (Static Random Access Memory) is devices that retain data for as long as power is applied to memory devices.

24. DRAM is an acronym for what type of device?

Dynamic Random Access Memory (DRAM) contains data for only a short period, usually 2-4 ms. it must be refreshed periodically.

25. compare between the following: ROM – PROM – EPROM – EEPROM

26. compare between the following: SIMM – DIMM

27. Why won't a 450 ns EEPROM work directly with 8088 5MHz?

This type of component requires wait states to operate properly with 8086/8088 because of its rather long access time.

28. What can be stated about the amount of time to erase or write in flash memory?

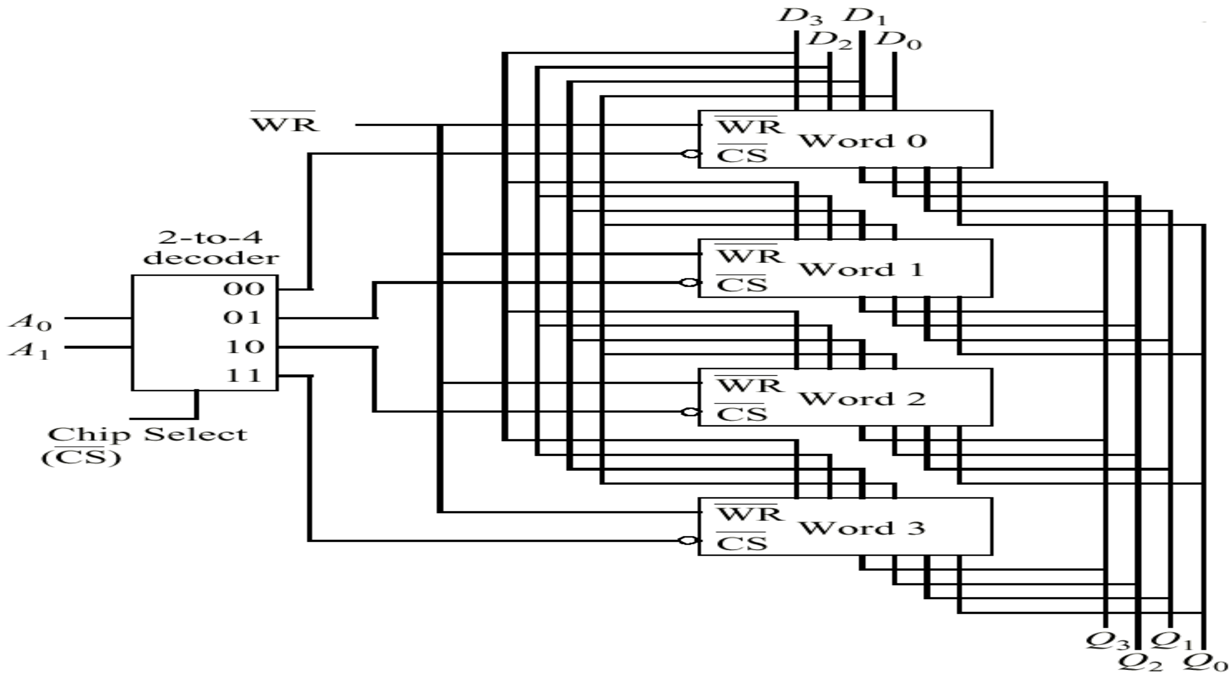
The difference between Flash Memory and EEPROM are, EEPROM erases and rewrite its content one byte at a time or in other words, at byte level. Whereas Flash memory erases or writes its data in entire blocks, which makes it a very fast memory compared to EEPROM. Flash memory can't replace DRAM and SRAM because the speed at which the DRAM/SRAM can access data and also their ability to address at byte level can't be matched by Flash.

A write operation in any type of flash device can only be performed on an empty or erased unit. So in most cases write operation must be preceded by an erase operation and this takes much time.

29. What are the purposes of CAS and RAS inputs of a DRAM?

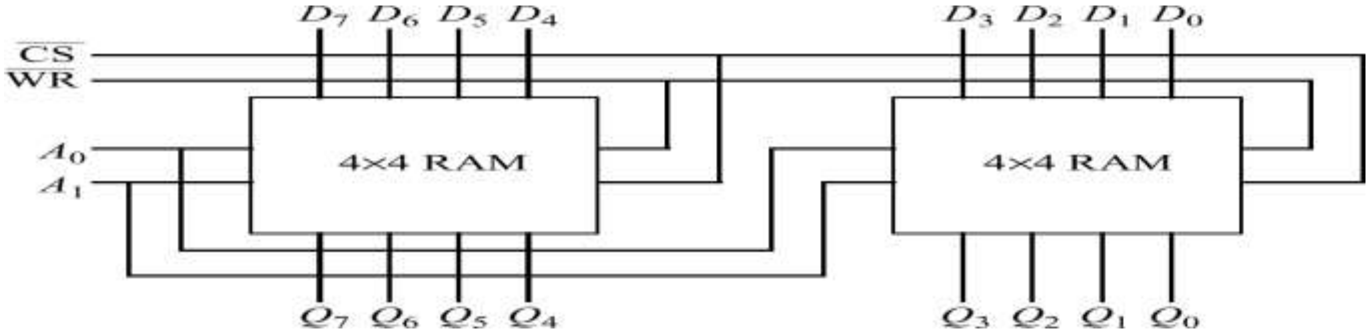
- First, A0–A7 are placed on the address pins and strobed into an internal row latch by RAS as the row address .
- Next, address bits A8-A15 are placed on the same eight address inputs and strobed into an internal column latch by CAS as the column address
- The 16-bit address in the internal latches addresses the contents of one of the 4-bit memory locations .
- CAS also performs chip selection input to DRAM

30. Design A Four-Word Memory with Four Bits per Word?

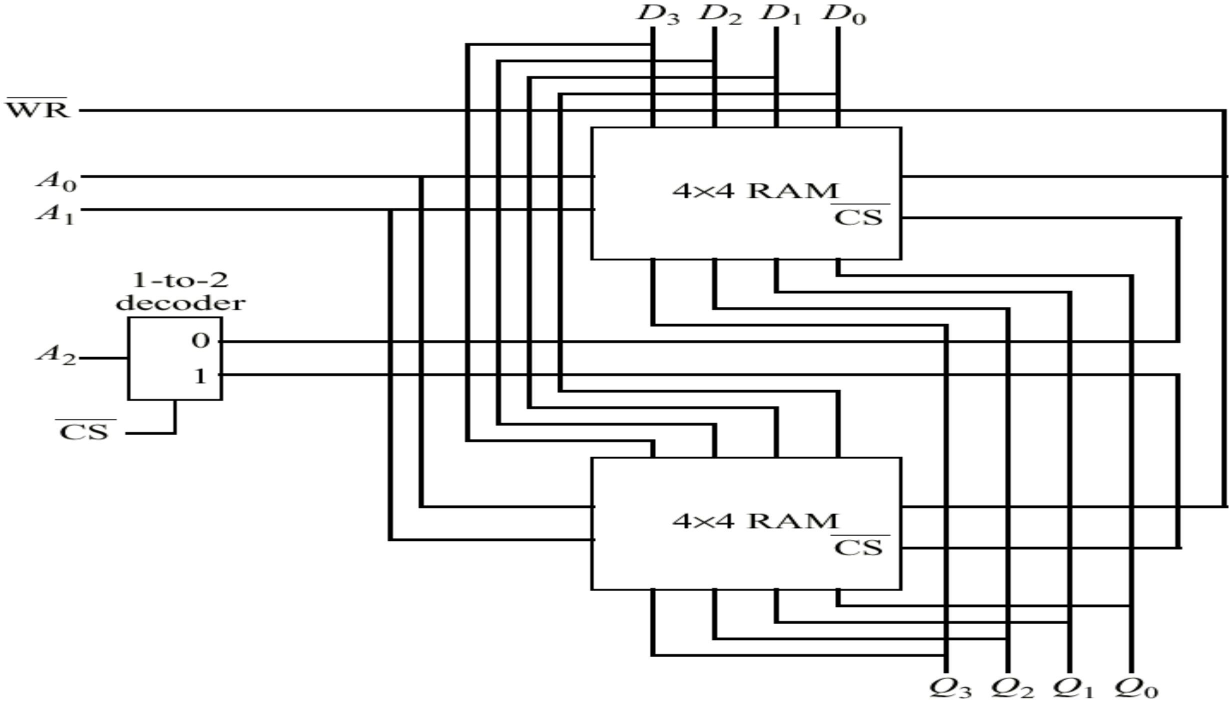


D_3 D_1
D_2 D_0
$\overline{WR}$
2-to-4
decoder
00
01
10
11
A_0
A_1
Chip Select
$(\overline{CS})$
$\overline{WR}$ Word 0 $\overline{CS}$
$\overline{WR}$ Word 1 $\overline{CS}$
$\overline{WR}$ Word 2 $\overline{CS}$
$\overline{WR}$ Word 3 $\overline{CS}$
Q_3 Q_1
Q_2 Q_0

31. Use a Two Four-Word by Four-Bit to design one word of 8-bit?



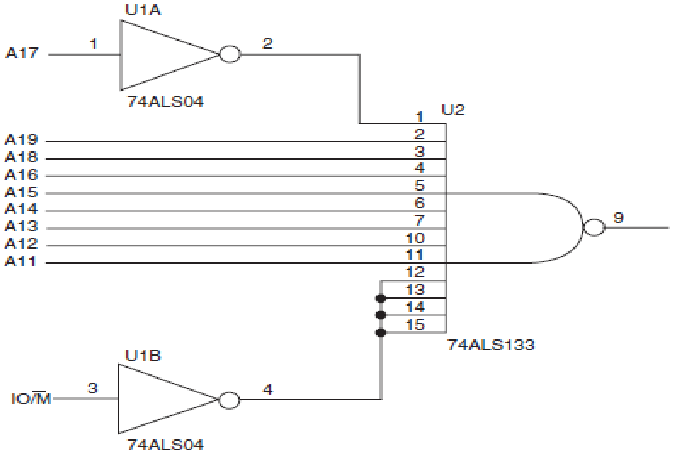
32. Use two Four-Word by Four-Bit RAMs to make up an Eight-Word by Four-Bit RAM?



Part 2:
Memory Address Decoding

1. **Why are memory address decoder important?**

The 8088 has 20 address connections and the 2716 EPROM has 11 connections. The 8088 sends out a 20-bit memory address whenever it reads or writes data because the 2716 has only 11 address pins, there is a mismatch that must be corrected The decoder corrects the mismatch by decoding address pins that do not connect to the memory component.

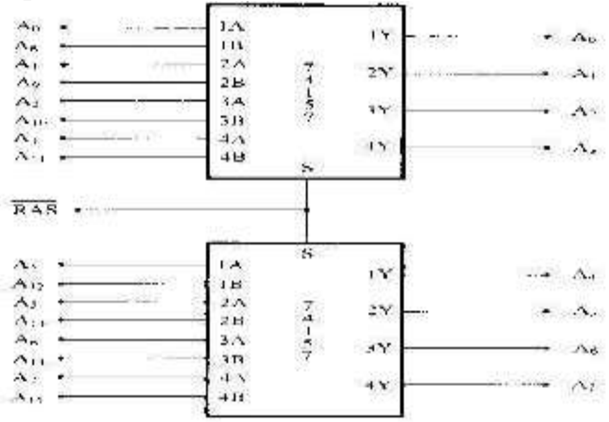


2. **The TMS4464 has 8 address inputs, yet it is a 64 k DRAM, explain how a 16-bit memory address can be forced into eight address input?**

The addresses inputs to most DRAMS are multiplexed. This allows the 16 bit address to be sent to DRAM through 8 address input pins.

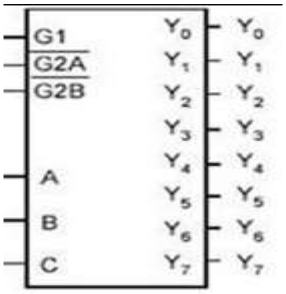
Address bits can be forced into eight address pins in two 8-bit increments.

This requires two special pins: the column address strobe (CAS) and row address strobe (RAS)



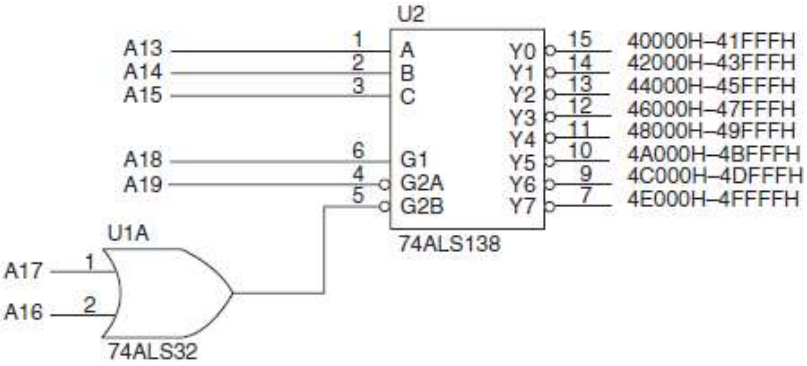
3. **For The 74LS138 3X8 decoder in figure**

When the G1 input is high and both $\overline{G2A}$ and $\overline{G2B}$ are low, what happens to the outputs of the 74HCT138 3-to-8 line decoder?



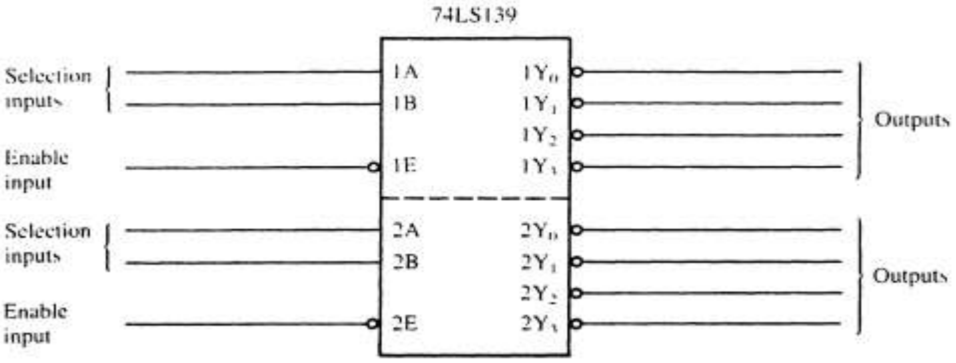
4. **When the G1 input is high and both $\overline{G2A}$ and $\overline{G2B}$ are low, what happens to the outputs of the 74HCT138 3-to-8 line decoder?**

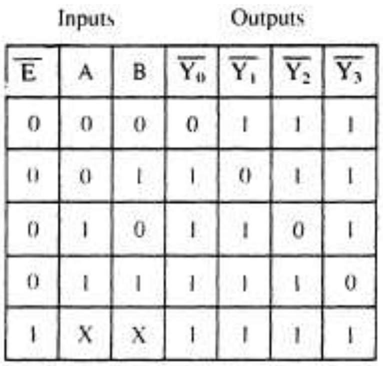
One of the eight outputs becomes a logic zero as dictated by the address inputs.



5. **Describe 74LS139 decoder (Dual 2 X 4 Decoder)?**

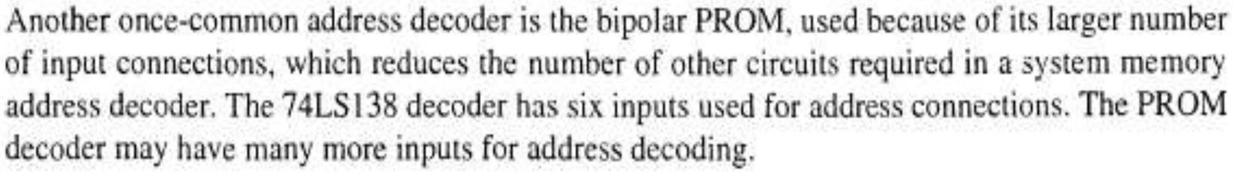
The 74LS139 is a chip which contains two decoders (2X4) each.
Each decoder has two inputs and four outputs, one output active at a time
According to the following table





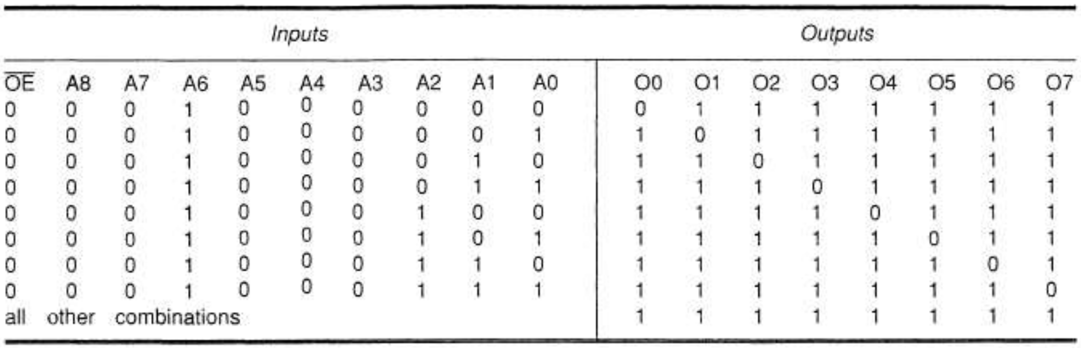
Inputs			Outputs			
$\overline{E}$	A	B	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	X	X	1	1	1	1

6. **Why is a PROM address decoder often found in a memory system?**

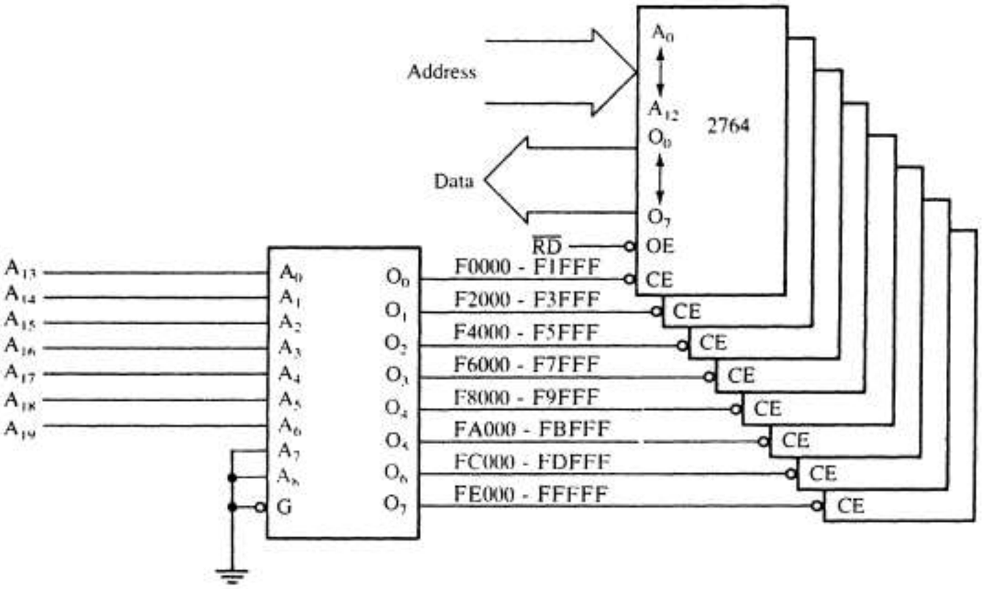


Another once-common address decoder is the bipolar PROM, used because of its larger number of input connections, which reduces the number of other circuits required in a system memory address decoder. The 74LS138 decoder has six inputs used for address connections. The PROM decoder may have many more inputs for address decoding.

When using PROM as an address decoder we must first modify the table
Which written in thee PROM

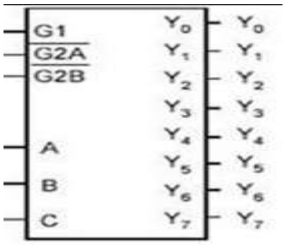


Inputs										Outputs							
$\overline{OE}$	A8	A7	A6	A5	A4	A3	A2	A1	A0	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	0	0	0	0	0	1	1	0	1	1	1	1	1	1
0	0	0	1	0	0	0	0	1	0	1	1	0	1	1	1	1	1
0	0	0	1	0	0	0	0	1	1	1	1	1	0	1	1	1	1
0	0	0	1	0	0	0	1	0	0	1	1	1	1	0	1	1	1
0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	1	1
0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1
0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	0
all other combinations										1	1	1	1	1	1	1	1

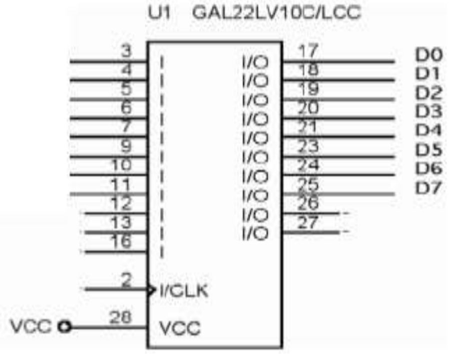


A memory system using 512 X8 PROM as an address decoder.

7. **Use the NAND gate decoder to select the EPROM memory addresses started from 0F800H–0FFFFH**
8. **For The 74LS138 3X8 decoder in figure:**

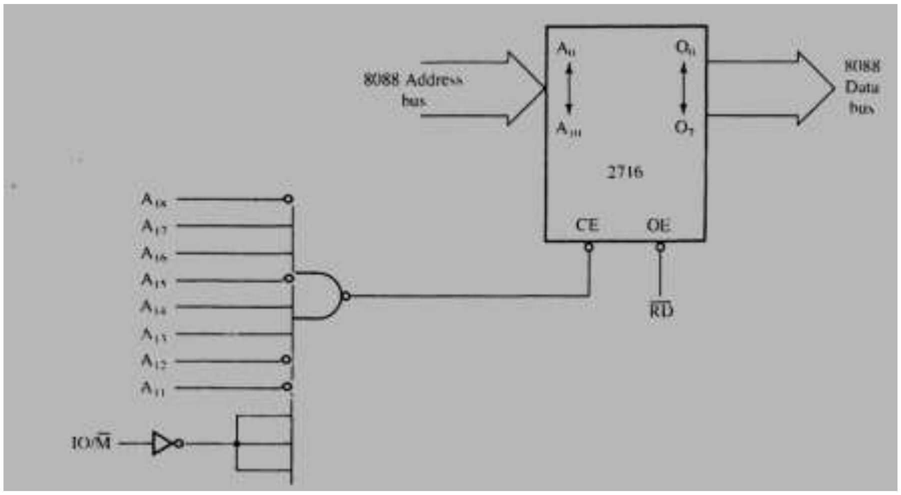


a. Find the truth table of this decoder?
b. Use this decoder to interface three 2764 EPROMs (64K × 8) at Y2, Y4 and Y6.
c. Find the address range of these 3 Memory chips?



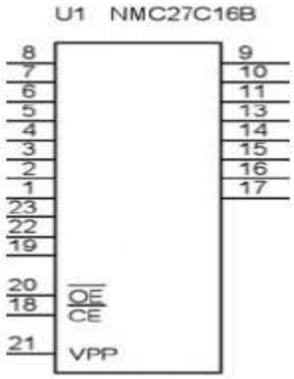
9. **Design an interfacing circuit to interface eight 2732 EPROM (4K X 8) with a 8088 microprocessor so that the total memory map is C8000H – CFFFFH**
10. **Design an interfacing circuit to interface eight 2732 EPROM (8K X 8) with a 8088 microprocessor so that the total memory map is C0000H – CFFFFH.**

11. **Identify the memory map of the following figure:**



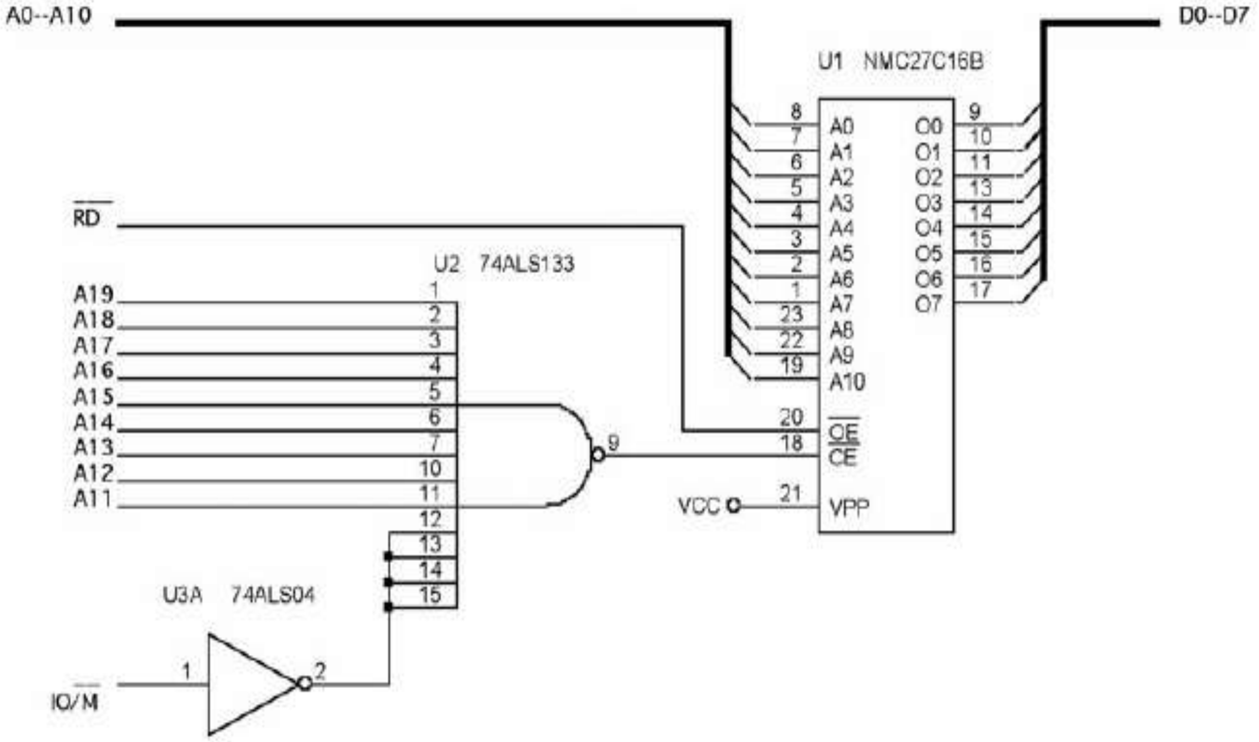
Then modify the circuit so that the address range become DF800H – DFFFFH

Question (2) **(15 Marks)**

The NMC27C16B in figure have 16,384-Bit (2048 x 8) CMOS EPROM a. How many address lines are required? **Log_2 2048 = 11 Address Line** b. How many data lines are required? **Required data lines are 8 lines.** c. What is the purpose of the $\overline{OE}$ and $\overline{CE}$ pins. **$\overline{\text{CE}}$: These pins are used to select or enable device so that it can perform read or write operation.** **$\overline{OE}$: enables or disable a set of tri-state buffers located within memory devices & must be active to read data.**	 U1 NMC27C16B 8 7 6 5 4 3 2 1 23 22 19 20 18 21 9 10 11 13 14 15 16 17 OE CE VPP

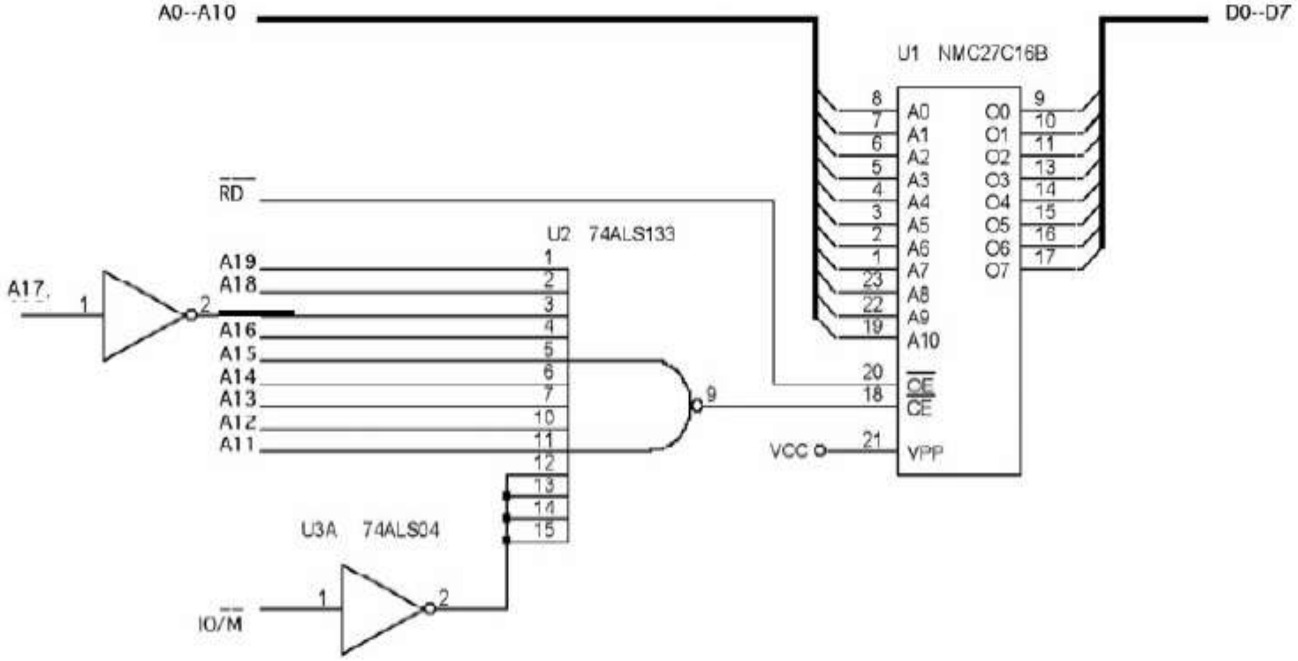
d. Use the NAND gate decoder to select the EPROM memory addresses started from FF800H–FFFFFH

Start	**F**	**F**	**8**	**0**	**0**	H
Start	**1111**	**1111**	**1**000	0000	0000	Binary
End	**1111**	**1111**	**1**111	1111	1111	Binary
End	**F**	**F**	**F**	**F**	**F**	H



e. Modify the NAND gate decoder to select the memory for address range DF800H–DFFFFH.

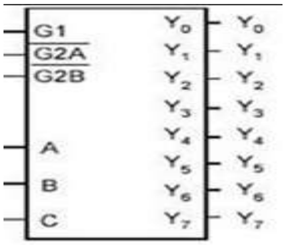
Start	**D**	**F**	**8**	**0**	**0**	H
Start	**1101**	**1111**	**1**000	0000	0000	Binary
End	**1101**	**1111**	**1**111	1111	1111	Binary
End	**D**	**F**	**F**	**F**	**F**	H

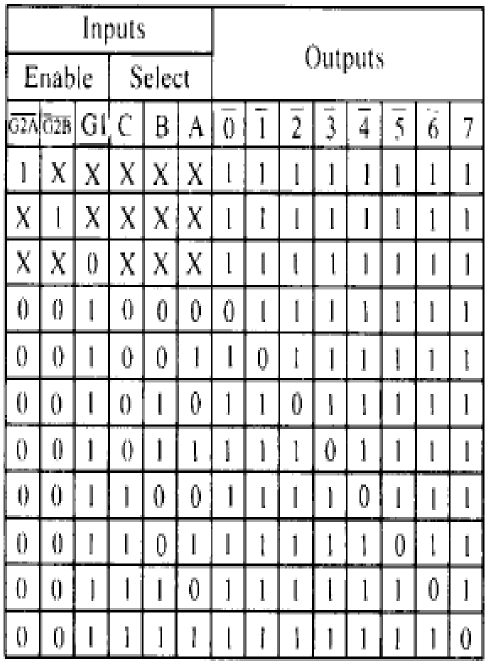


Question (3) (15 Marks)

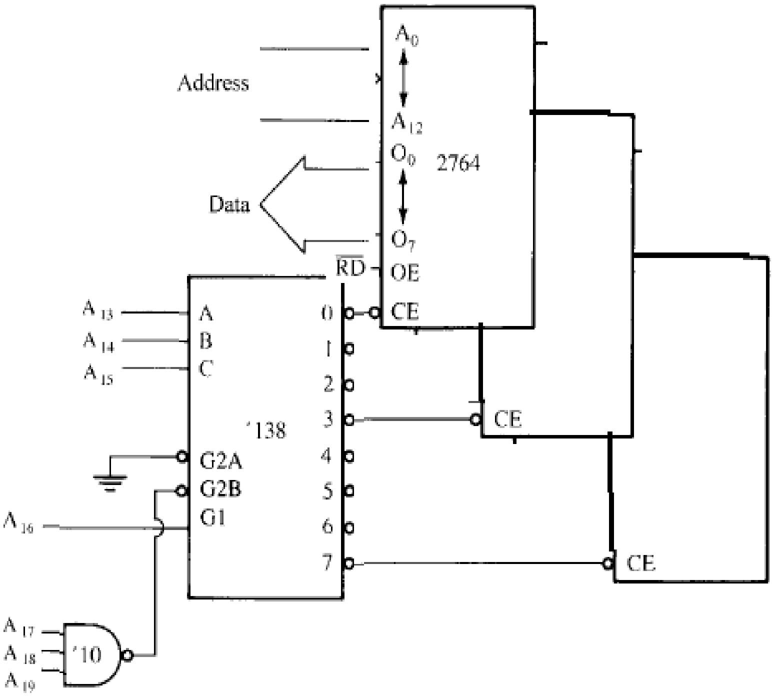
For The 74LS138 3X8 decoder in figure:

d. Find the truth table of this decoder?
e. Use this decoder to interface **three** 2764 EPROMs (64K × 8) at Y0, Y3 and Y7.
f. Find the address range of these 3 Memory chips?





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



Start	**F**	**0**	**0**	**0**	**0**	H
Start	**1111**	**000**0	0000	0000	0000	Binary
End	**1111**	**000**1	1111	1111	1111	Binary
End	**F**	**1**	**F**	**F**	**F**	H

Start	**F**	**6**	**0**	**0**	**0**	H
Start	**1111**	**011**0	0000	0000	0000	Binary
End	**1111**	**011**1	1111	1111	1111	Binary
End	**F**	**7**	**F**	**F**	**F**	H

Start	**F**	**E**	**0**	**0**	**0**	H
Start	**1111**	**111**0	0000	0000	0000	Binary
End	**1111**	**111**1	1111	1111	1111	Binary
End	**F**	**F**	**F**	**F**	**F**	H

Part 3
VHDL

1. **What is VHDL?**
2. **Draw the circuit layout the represent the effect of running this VHDL code?**

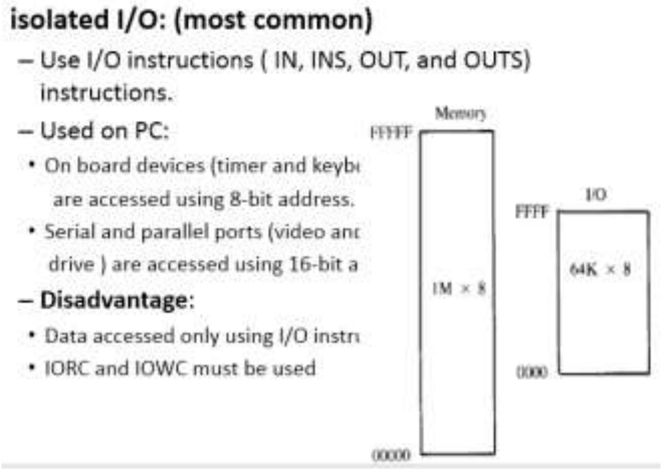
```
port (   A19, A18, A17, MIO: in STD_LOGIC;
         ROM, RAM, AX19: out STD_LOGIC );
end;
architecture V1 of DECODER_10_19 is
begin
ROM <= A19 or A18 or A17 or MIO;
RAM <= not (A18 and A17 and (not MIO));
AX19 <= not A19;
end V1;
```

3. **Describe the effect of running the following VHDL code?**

```
-- VHDL code for the decoder
library ieee;
use ieee.std_logic_1164.all;
entity DECODER_10_19 is
port (   A19, A18, A17, MIO: in STD_LOGIC;
         ROM, RAM, AX19: out STD_LOGIC );
end;
architecture V1 of DECODER_10_19 is
begin
ROM <= A19 or A18 or A17 or MIO;
RAM <= not (A18 and A17 and (not MIO));
AX19 <= not A19;
end V1;
```

Part 4
I/O Interface

1. Contrast a memory-mapped I/O system with an isolated I/O system.



isolated I/O: (most common)

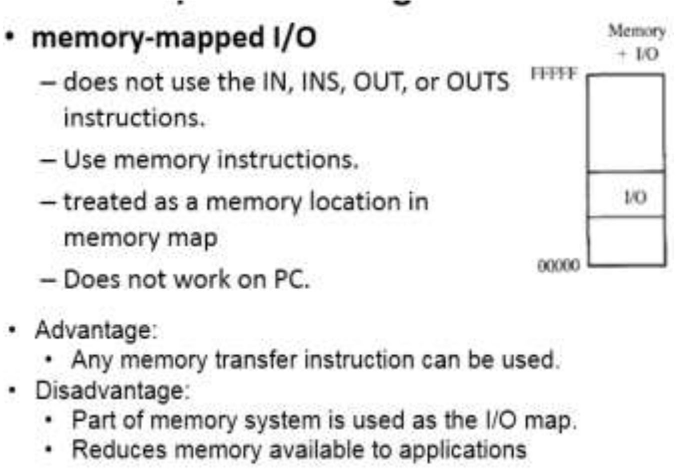
– Use I/O instructions (IN, INS, OUT, and OUTS) instructions.

– Used on PC:

- On board devices (timer and keyb
  are accessed using 8-bit address.
- Serial and parallel ports (video anc
  drive) are accessed using 16-bit a

– **Disadvantage**:

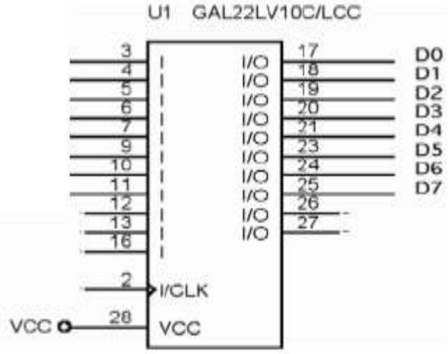
- Data accessed only using I/O instr
- IORC and IOWC must be used



- **memory-mapped I/O**
  – does not use the IN, INS, OUT, or OUTS instructions.
  – Use memory instructions.
  – treated as a memory location in memory map
  – Does not work on PC.
- Advantage:
  - Any memory transfer instruction can be used.
- Disadvantage:
  - Part of memory system is used as the I/O map.
  - Reduces memory available to applications

2. What is the basic input interface?
3. What is the basic output interface?

4. Write a VHDL code to design the PLD version, using a GAL22V10 (a low-cost device in figure 3) for decoding two 8-Bit I/O Port Addresses using Fixed address located at ports F0,F1?
5. What is the basic output interface, and how to assign an address to an output port?
6. Compare between the following Fixed addressing and variable addressing
7. Develop an I/O port decoder, using a 74ALS138 (3X8 decoder), to interface 8 I/O devices using the fixed address for the following 8-bit I/O port addresses: E0H, E1H, E2H, E3H, E4H, E5H, E6H, and E7H.
8. Develop an I/O port decoder, using a PLD (in figure 2), that interface two I/O devices using fixed addresses for the following I/O port addresses: D0H, D1H.

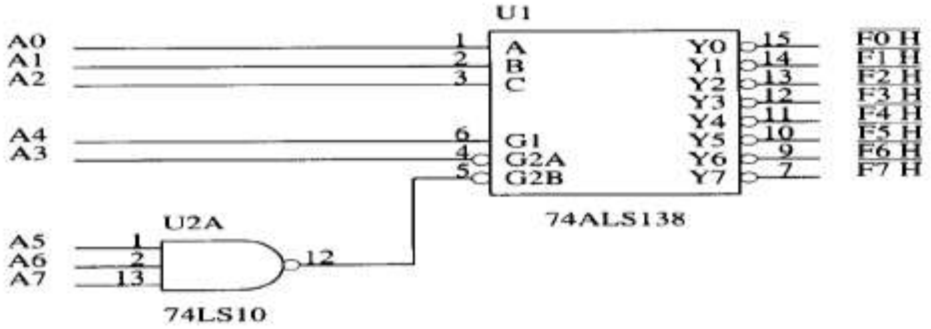


9. **What is the different between memory address decoding and memory mapped I/O address decoding?**

Memory mapped I/O address decoding is the same as memory address decoding (except that the $\overline{IOWC}$ and $\overline{IORC}$ are not used because there is no IN or OUT or INS or OUTS instruction). The I/O is treated as normal memory location.

10. **What is the different between memory address decoding and Isolated I/O address decoding?**
    - The number of address pins connect to the decoder is different:
        - In memory: decode $A_0 - A_{31}$, $A_0 - A_{23}$, or $A_0 - A_{19}$ for memory,
        - **In I/O if fixed address** is used it decode $A_0 - A_7$ and from $A_8 - A_{15}$ are zeros, and from $A_{16} - A_{19}$ are undefined. If **variable address** are used it decode $A_0 - A_{15}$ and from $A_{16} - A_{19}$ are undefined.
    - $\overline{M}/IO$ and $\overline{RD}$ control are used to activate I/O device for reading ($\overline{IORC}$) and $\overline{M}/IO$ and $\overline{WR}$ control are used to activate I/O device for writing operation($IOWC$).

11. **How to Decode 8-Bit I/O Port Addresses using Fixed address?**
    - Only address connections $A_0 - A_7$ will be used and the other will be ignored.
    - Figure Q3 illustrates a 74ALS138 decoder that decodes 8-bit I/O ports F0H through F7H.



The address at Y4 will be F4H according to the following table.

A7	A6	A5	A4(G1)	A3(G2A)	A2(C)	A1(B)	A0(A)
1	1	1	1	0	1	0	0
F				4			

The address at Y1 will be F1H according to the following table.

A7	A6	A5	A4(G1)	A3(G2A)	A2(C)	A1(B)	A0(A)
1	1	1	1	0	0	0	1
F				1			

12. **Write a VHDL code to design the PLD version, using a GAL22V10 (a low-cost device) for decoding 8-Bit I/O Port Addresses using Fixed address?**
    - The VHDL program for the PLD is

```
-- VHDL code for the decoder of Figure Q4
library ieee;
use ieee.std_logic_1164.all;
entity DECODER_Q4 is
port (        A7, A6, A5, A4, A3, A2, A1, A0: in STD_LOGIC;     ---List Here the inputs
                    D0, D1, D2, D3, D4, D5, D6, D7: out STD_LOGIC );  ---List Here the outputs
end;
architecture V1 of DECODER_Q4 is
begin
D0 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and not A1 and not A0 ); -- ≈ F̅0̅
D1 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and not A1 and A0 ); -- ≈ F̅1̅
D2 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and A1 and not A0 );
D3 <= not( A7 and A6 and A5 and A4 and not A3 and not A2 and A1 and A0 );
D4 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and not A1 and not A0 );
D5 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and not A1 and A0 );
D6 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and A1 and not A0 );
D7 <= not( A7 and A6 and A5 and A4 and not A3 and A2 and A1 and A0 );
end V1;
```

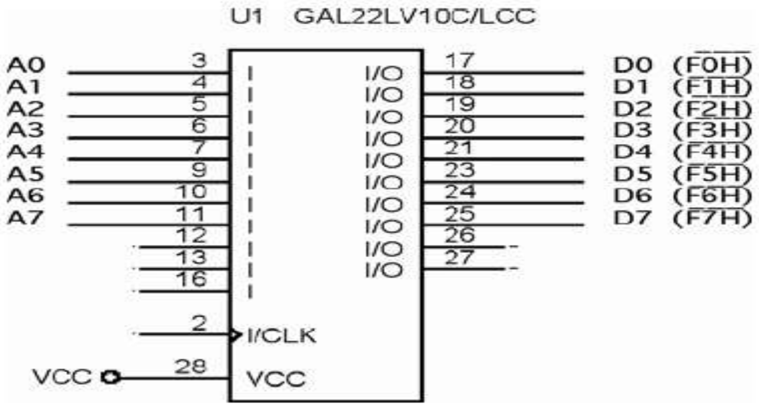


FIGURE Q4: A PLD that generates part selection signals F0H–F0H.

13. **How to Decode 16-Bit I/O Port Addresses using variable address using a GAL22V10 (write VHDL code)?**

- Figure Q5 illustrates a circuit that contains a PLD and a 4-input NAND gate used to decode I/O ports EFF8H–EFFFH. The NAND gate decodes part of the address (A15, A14, A13, and A11) because the PLD does not have enough address inputs. The output of the NAND gate connects to the Z input of the PLD and is decoded as a part of the I/O port address. The PLD generates address strobes for I/O ports – .
- The program for the PLD is listed in Example 11–3.

```
    -- VHDL code for the decoder of Figure Q-5
    library ieee;
    use ieee.std_logic_1164.all;
    entity DECODER_Q5 is
    port (     Z, A12, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1, A0: in STD_LOGIC;
               D0, D1, D2, D3, D4, D5, D6, D7: out STD_LOGIC);
    end;
    architecture V1 of DECODER_Q5 is
    begin
D0 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and not A2 and not A1 and not A0 );
D1 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and not A2 and not A1 and A0 );
D2 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and not A2 and A1 and not A0 );
D3 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and not A2 and A1 and A0 );
D4 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and A2 and not A1 and not A0 );
D5 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and A2 and not A1 and A0 );
D6 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                    and A2 and A1 and not A0 );
D7 <= not ( not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3
                  and A2 and A1 and A0 );
end V1;\
```

To find the address of the output D4 for example

1. get the code for D4

D4 <= not (not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3 and A2 and not A1 and not A0);

2. remove the first NOT

D4 <= (not Z and not A12 and A10 and A9 and A8 and A7 and A6 and A5 and A4 and A3 and A2 and not A1 and not A0);

<u>not Z</u> means (inputs to the NAND A11=1,A13=1,A14=1 A15=1)
<u>and not A12</u> means (A12=0) , <u>and A10</u> means (A10=1)
<u>and A9</u> means (A9=1) , <u>and A8</u> means (A8=1) , <u>and A7</u> means (A7=1)
<u>and A6</u> means (A6=1), <u>and A5</u> means (A5=1), <u>and A4</u> means (A4=1)
<u>and A3</u> means (A3=1), <u>and A2</u> means (A2=1), <u>and not</u> A1 means (A1=0)
<u>and not</u> A0 means (A0=0)

3. fill in the following table

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0
E				**F**				**F**				**C**			

From the table the address that decode the I/O device connected to the output pin D4 will be EFFE H

By the same method find the address that decode the I/O device connected to the output pins D1, D2, D3, D4, D5, D6 (It will be as the following figure)

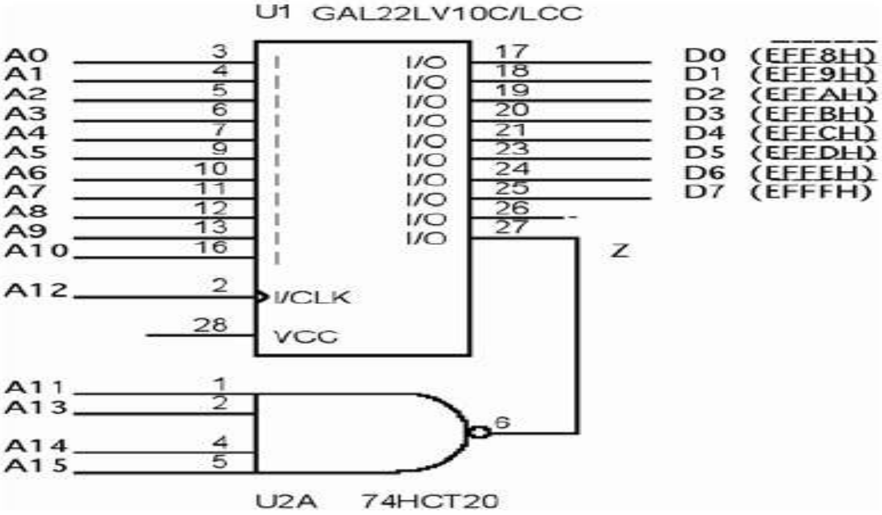


FIGURE Q5: A PLD that decodes 16-bit I/O ports EFF8H through EFFFH.

<u>Important Note</u>: any change in the VHDL code will be change the address at the output of the PLD device.

14. What is the different of <u>I/O Port address</u> and <u>I/O port wide</u>?

- **I/O Port address** are the address the decode the I/O device to enable it for read or write operation
- **I/O port wide** is the wide of data read or written from the I/O device and it may be 8-bit wide or 16-bit wide or 32-bit wide or 64-bit wide.

15. **How the data can be transferred between microprocessor and I/O device?**

- **In case of 8-bit wide I/O devices**: Data transferred to an 8-bit I/O device exist in one of the I/O banks in a 16-bit microprocessor such as the 80386SX. There are 64K different 8-bit ports, because a 16-bit port uses only one 8-bit ports (one of the I/O banks).
- **16-bit wide I/O devices**: The I/O system on such a microprocessor contains two 8-bit memory banks, just as memory does. This is illustrated in Figure, which shows the separate I/O banks for a 16-bit system such as the 80386SX.
- In a 16-bit microprocessor such as the 80386SX. There are only 32K different 16-bit ports because a 16-bit port uses two 8-bit ports (the two I/O banks).

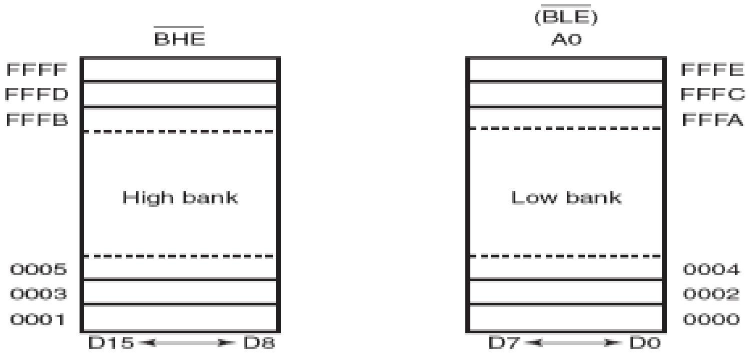


Figure: The I/O banks found in the 8086, 80186, 80286, and 80386SX.

16. **How to connect two different 8-bit (port-wide) I/O devices using fixed address method (8-bit I/O address) located at 40H and 41H for write operation?**

- Because these are 8-bit output devices and because they appear in different I/O banks, separate I/O write signals are generated to clock a pair of latches that capture port data.
- Note that all I/O ports use 8-bit addresses.
- Thus, ports 40H and 41H can each be addressed as separate 8-bit ports, or together as one 16-bit port.

Port	A7	A6	A5	A4	A3	A2	A1	A0
40 H	0	1	0	0	0	0	0	0
VHDL	Not A7	And A6	And Not A5	And Not A4	And Not A3	And Not A2	And Not A1	And Not A0

Port	A7	A6	A5	A4	A3	A2	A1	A0
41 H	0	1	0	0	0	0	0	1
VHDL	Not A7	And A6	And Not A5	And Not A4	And Not A3	And Not A2	And Not A1	And Not A0

- According to the above tables, the program for the PLD decoder used will be:

```
    -- VHDL code for the decoder of Figure Q8
    library ieee;
    use ieee.std_logic_1164.all;
    entity DECODER_Q8 is
    port (
    BHE, IOWC, A7, A6, A5, A4, A3, A2, A1, A0: in STD_LOGIC;
    D0, D1: out STD_LOGIC );
    end;
    architecture V1 of DECODER_Q8 is
    begin
D0 <= not BHE and not IOWC and A7 and not A6 and A5 and A4 and A3 and A2 and A1 and A0;
D1<= BHE and not IOWC and A7 and not A6 and A5 and A4 and A3 and A2 and A1 and not
A0;
    end V1;
```

- The design will be as shown in figure

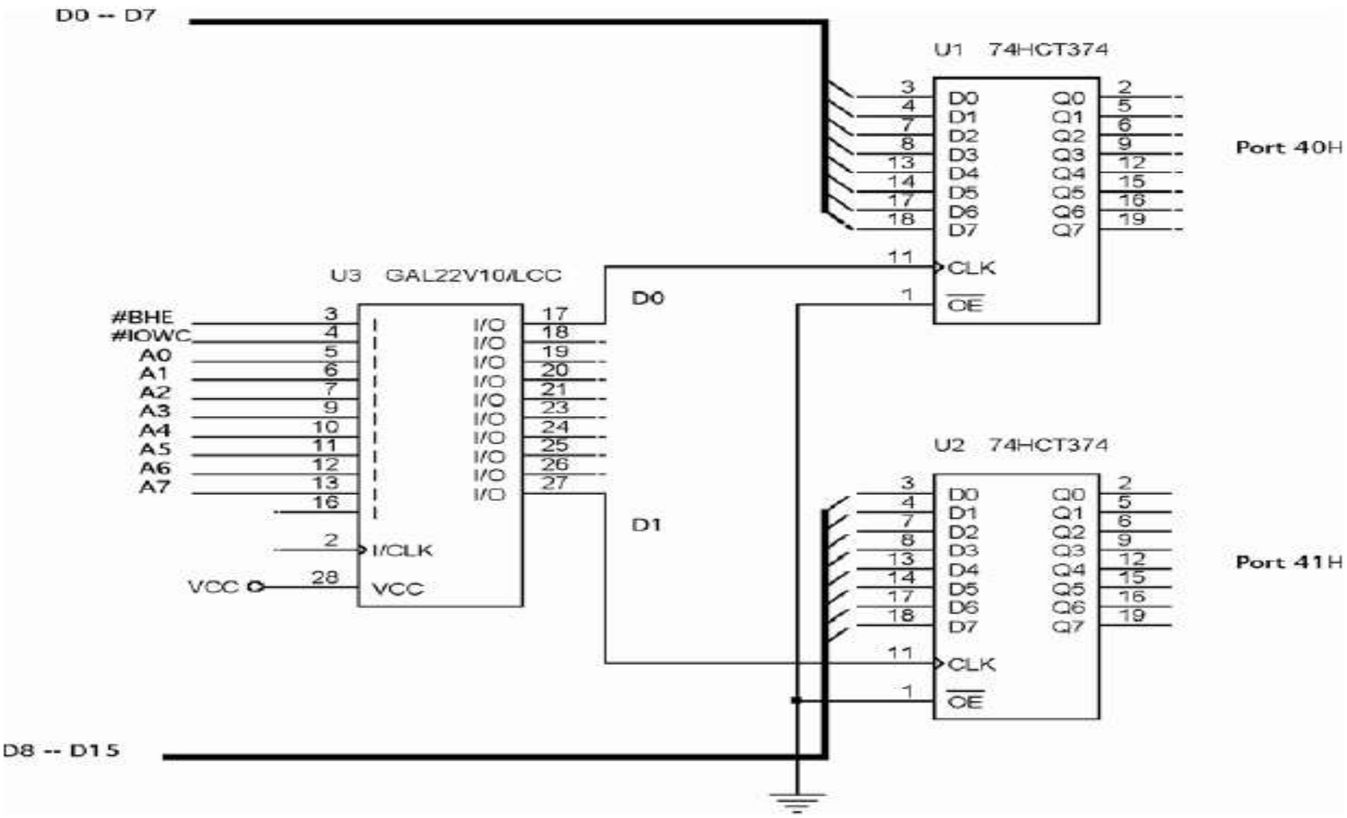


FIGURE: An I/O port decoders that select ports 40H and 41H for output data.

17. **How to connect 16-bit I/O device using fixed address method (8-bit I/O address) located at 64H and 65H for read operation?**
    - The PLD decoder does not have a connection for address bits BHE because these signals do not apply to 16-bit-wide I/O devices.
    - Only port number 64 will be decoded and 65 will be generated automatically.

Port	A7	A6	A5	A4	A3	A2	A1	A0
64 H	0	1	1	0	0	1	0	0

VHDL	Not A7	And A6	And A5	And Not A4	And not A3	And A2	And Not A1	And Not A0

- The program for the PLD

```
-- VHDL code for the decoder of Figure Q9
library ieee;
use ieee.std_logic_1164.all;
entity DECODER_Q9 is
port (   IORC, A7, A6, A5, A4, A3, A2, A1: in STD_LOGIC;  D0: out STD_LOGIC );
end;
architecture V1 of DECODER_Q9 is
begin
D0 <= not IORC and not A7 and A6 and A5 and not A4 and not A3 and not A2 and not A1 and
          not A0;
end V1;
```

- As illustrated in Example, shows how the enable signals are generated for the three-state buffers (74HCT244) used as input devices.

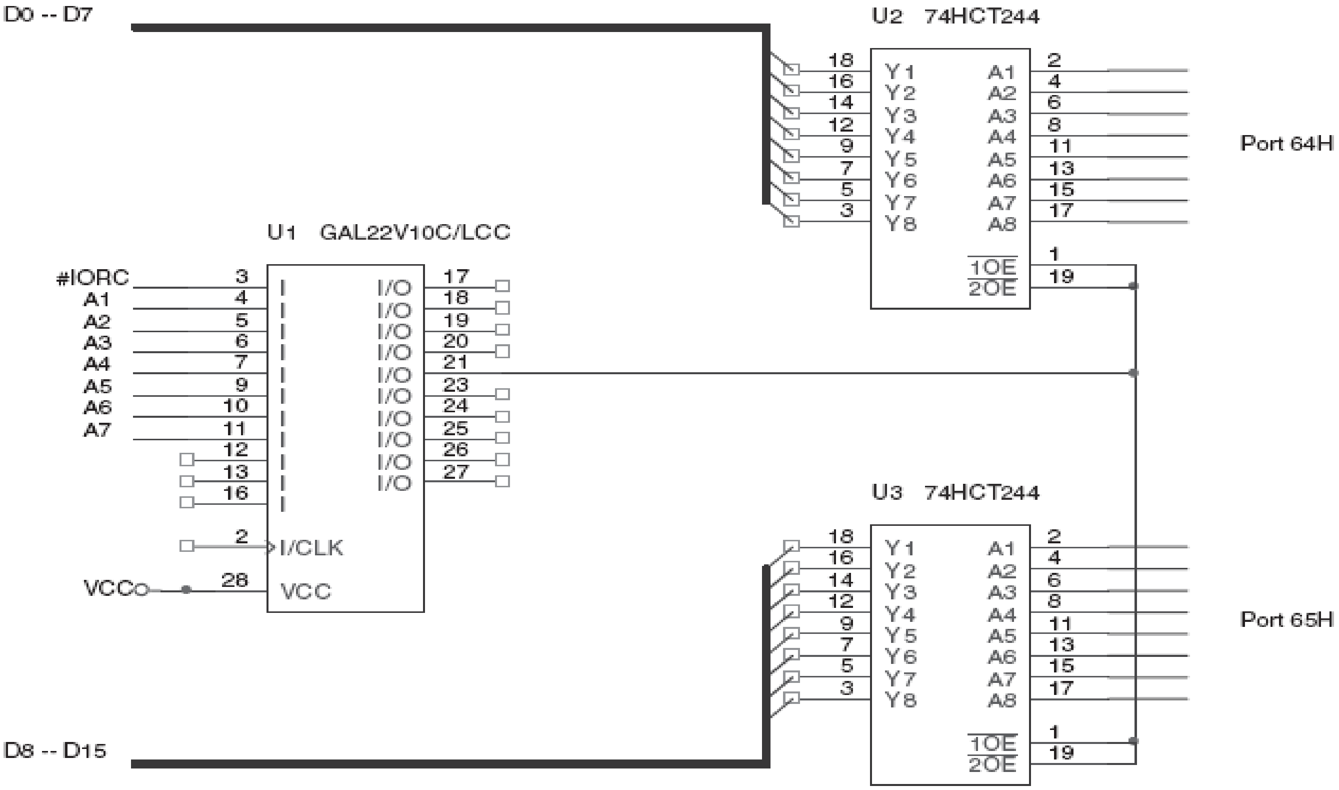


FIGURE: A 16-bit-wide port decoded at I/O addresses 64H and 65H.

18. What is the programmable Peripheral Interface (PPI)?

The 82C55 programmable peripheral interface (PPI) is:

- A very popular, low-cost interfacing component found in many applications.
- The PPI, which has 24 pins for I/O that are programmable in groups of 12 pins,
- Have groups that operate in three distinct modes of operation.
- The 82C55 can interface any TTL-compatible I/O device to the microprocessor.

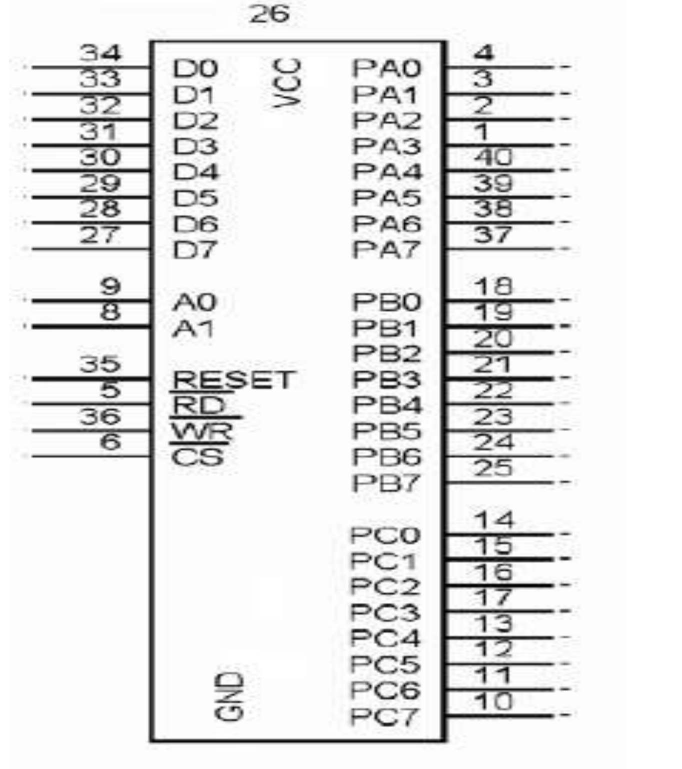
- The 82C55 (CMOS version) requires the insertion of wait states if operated with MP using higher than 8 MHz
- The 82C55 is used for interface to the keyboard and the parallel printer port in PC.

19. The 82C55 PPI has how many programmable I/O PIN CONNECTION. List the pins that belong to group A & group B in the 82C55. Also mention two pins which accomplish internal I/O port section.

The 82C55 has 24 programmable I/O.

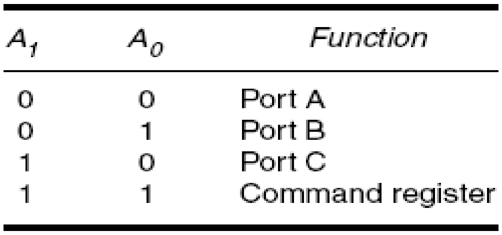
- Group A consist:
  1- port A
  2- upper half of port C (PC7-PC4)
  3- A1,A0,WR
- Group b consist from :
  1- port B (PB7-PB0)
  2- lower half of port c(PC3-PC0_
  3- RD
  4- RESET
  5- CK
  6- VCC
  7- GND

. Two pins accomplish internal I/O port address is A0 and A1



20. What three modes of operation are available to the 82C55 PPI? Mention the purpose of STB signal is strobes input operation of the 82C55

Mode 0: Latched I/O
Mode 1: strobes I/O
Mode2: bi-directional I/O.



A_1	A_0	Function
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Command register

21. How to connect 82C55 PPI to a microprocessor

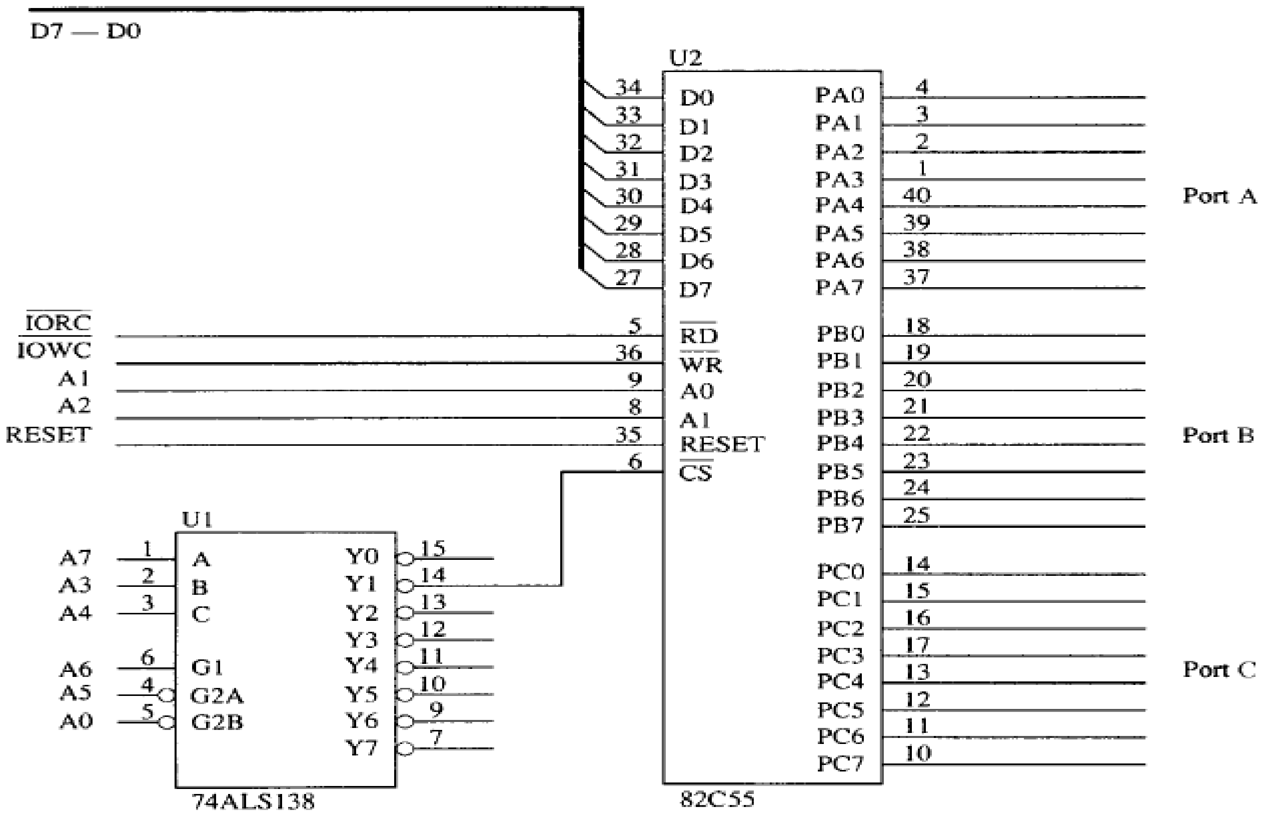
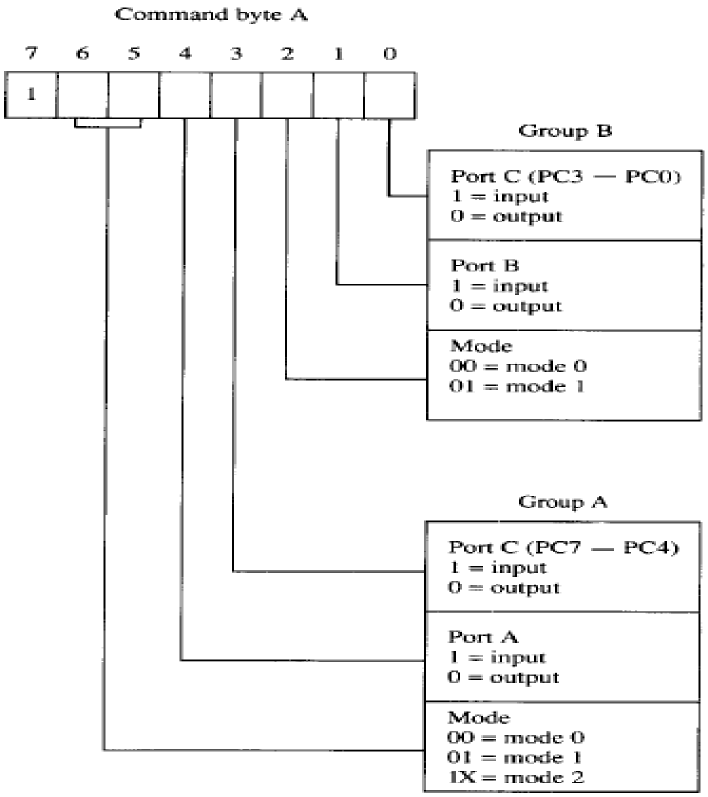
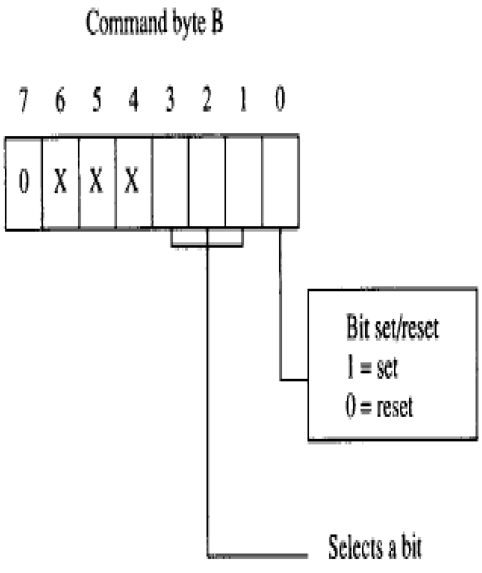


Figure: The 82C55 interfaced to the low bank of the 80386SX microprocessor.

It functions at 8-bit I/O port addresses C0H (port A), C2H (port B), C4H (port C), and C6H (command register).

22. How to program 82C55 PPI?


Command byte A
7 6 5 4 3 2 1 0
1
Group B
Port C (PC3 — PC0)
1 = input
0 = output
Port B
1 = input
0 = output
Mode
00 = mode 0
01 = mode 1
Group A
Port C (PC7 — PC4)
1 = input
0 = output
Port A
1 = input
0 = output
Mode
00 = mode 0
01 = mode 1
1X = mode 2


Command byte B
7 6 5 4 3 2 1 0
0 X X X
Bit set/reset
1 = set
0 = reset
Selects a bit