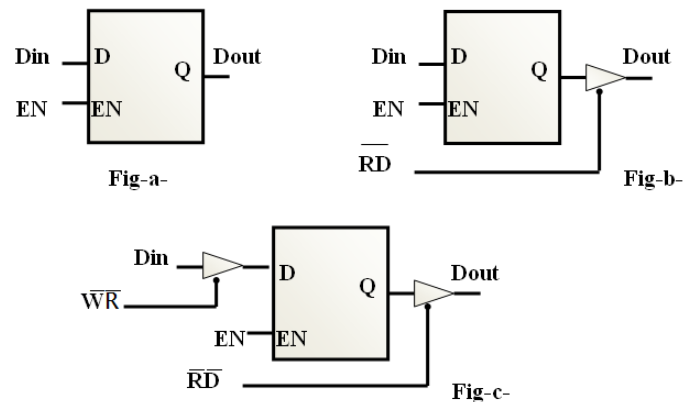- **Memory: -**

Is a group of registers, arranged in a sequence, to store bit. The number of bits stored in a register is called memory word.

Two- type of memory R/WM & ROM

Fig-a-   Fig-b-

Fig-c-

A flip- flop or latch is a basic element of memory to write or store a bit in the latch, we need an input data bit and an enable signal fig-a- in this latch, the stored bit is always available on the O/P line if a tri-state buffer is connected to the O/P of the latch as shown in fig-b- the stored bit can be read only when the buffer is enabled

Similarly, we can also use a tri-state buffer on the I/P of the latch, now we can write into the latch fig-c- by enabling the I/P buffer and read from it by enabling the O/P buffer. This latch, which can store one binary bit is called a memory cell.
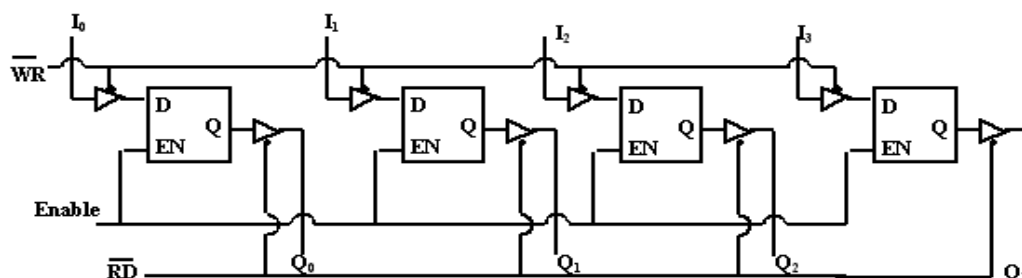

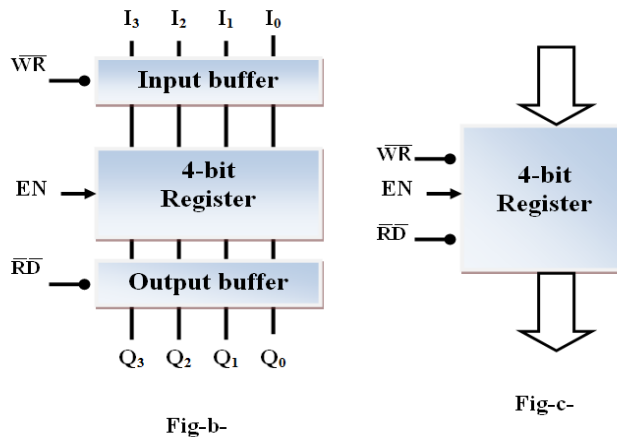Fig-a- four latches as a 4-bit register

Fig-b-

Fig-c-

Fig-b- & fig-c- shows simplified block diagram of 4-bit register.

Fig-a- shows four cells or latches grouped together to form a register which has 4 input lines and 4 output lines and can store 4-bits. The size of this register specified as either 4-bits or 1x4 bit, which indicate one register with four cells or four I/O lines.

The number of bits stored in a register is called a memory word

- **4x8 bit register: -**
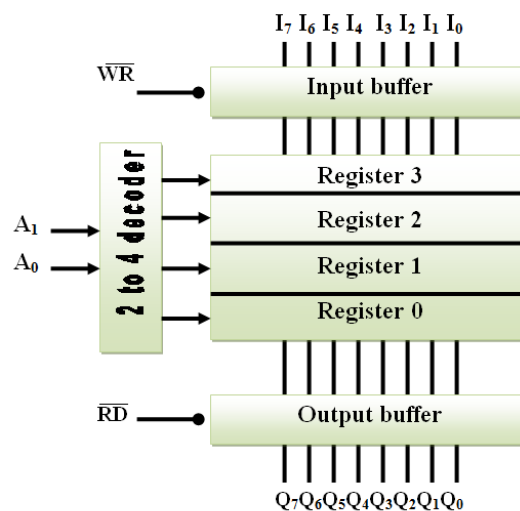
This fig shown 4 register with 8 cells or 8 bits memory word are arranged in a sequence to write into or read from any one of the registers, a specific register should be identified or enabled.

This is a simple decoding function a 2-to-4 decoder can perform the function, however, two input lines A_1 and A_0 called address lines are required to the decoder, these two

I/P lines can carry 4 different bit combinations (00,01,10,11), and each combination can identify or enable one of the registers named as register 0 through register 3

No. of register=2^n ,n= no. of line

	No. of line	Number of register
A1A0	2	4
A2A1A0	3	8
A3A2A1A0	4	16
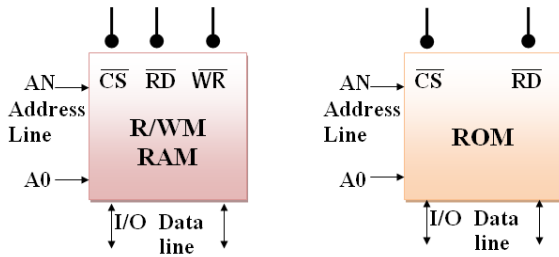
A model of typical memory chip

Fig-a- represent the read write memory chip, Fig-b- represent the read only memory, the only difference between a & b is that ROM dose not need $\overline{WR}$ signal.

- **To communicate with memory, the mp should be able to**

  1- Select the chip
  2- Identify the register
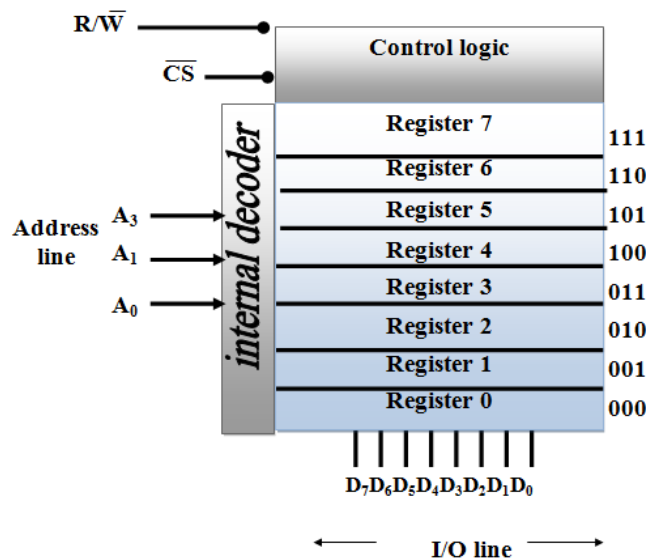 3- Read from or write into the register

Fig-a- memory chip with 8 register

Fig-a- shows memory chip of 8 registers with three address line, one chip select ($\overline{CS}$) line one read/write ($R/\overline{W}$) line, and eight I/O lines.

The mp uses the $\overline{CS}$ line to select the chip, and the $R/\overline{W}$ line to control data flow.

The register are arranged sequentially and numbered 000 to 111. These numbers are called memory addresses, identifying each register as a memory location.

To identify each register. The mp would require three address line to place eight different addresses from 000 to 111.

The microprocessor with its 16 address line is capable of identifying or addressing 65,536 (64K) such memory register or locations and the microprocessor with its 20 address line is capable of addressing 1,048,576 (1Mbyte) memory location

The size of this chip can be specified either as 8 byte, 8x8 bit or 64 bit.

- A memory chip with 256 register (or location) with 4 I/O lines is specified as 256x4 bit or 1024 bit, for an 8-bit mp, two such memory chips (256x4) would be necessary to form the 8 bit memory word size, resulting 256 bytes of memory.
- The chip select $\overline{CS}$ line, also known as chip enable ($\overline{CE}$), is necessary to select one particular memory chip from among several memory chips in a system.

- **Memory map: -**

   Is defined as the assignment of addresses to memory register in various memory chips in system. The assignment of memory addresses is done through the chip select logic.

EX:- illustrate the memory map of the chip with 256 bytes of memory shown in fig-a-
and explain how the memory map can be changed by modifying the hardware of the
chip select ($\overline{CS}$) line in fig-b-
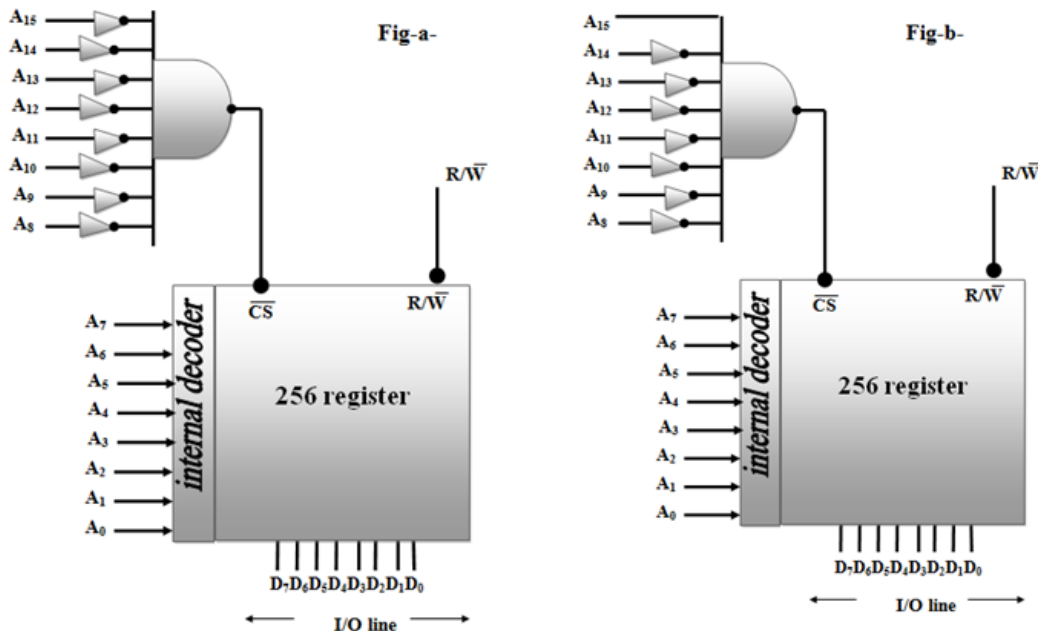
Solution: -

Fig-a- shows a memory chip with 256 registers and 8 I/O lines, the memory size of the
chip is expressed as 256 X 8.

It has eight address lines, one chip select ($\overline{CS}$) line (active low) and read/write ($R/\overline{W}$)
line.

The 8 address lines (A_7-A_0) of the mp are required to identify 256 (2^8) memory registers.

Eight other addresses lines (A_{15}-A_8) are connected to the chip select ($\overline{CS}$) line through
inverters and NAND gate.

The memory chip is enabled or selected when the ($\overline{CS}$) goes low, therefore, to select
the memory chip, the address line (A_{15}-$A_{8)}$ should be at logic 0, which will cause the
output of NAND gate to go low.

No other logic levels on the lines (A$_{15}$-A$_{8)}$ can select the chip. Once is selected (enabled).

The address lines (A$_7$-A$_{0)}$ can assume any combination from 00 H to FF H, and identify any one of the 256 memory locations through its decoder.

The control signal $R/\overline{W}$ enables data flow therefore; the memory addresses of the chip in fig-a- will range from 0000 H to 00FF H as shown below:

$$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8 \qquad A_7A_6A_5A_4A_3A_2A_1A_0$$

$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \qquad 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = 0000\ H$$

$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \qquad 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 = 00FF\ H$$

chip enable or chip select addresses   **register select addresses**

The entire range of the memory addresses from 0000 to 00FF is known as the memory map of the chip in fig-a-.

The chip select addresses are determined by the hardware (the inverter & NAND gate) therefore, the memory map of chip can be changed by modifying the hardware. If the inverter on line A$_{15}$ is removed as shown in fig-b-. The address required on A$_{15}$-A$_8$ to enable the chip will be as follows

$$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9A_8$$

$$1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = 80\ H$$

The memory map of the chip (fig-b-) range from 8000H to 80FF H.

By changing the hardware of the chip select logic, the memory map can be changed and memory can be assigned addresses in various locations over the entire map

- **Type of memory:**

```
                        ┌──────────┐
                        │  Memory  │
                        └──────────┘
                  ┌──────────┴──────────────┐
        ┌───────────────────┐      ┌──────────────────┐
        │ Prime or main     │      │ Storage memory   │
        │ Memory            │      │                  │
        └───────────────────┘      └──────────────────┘
          ┌────────┴────────┐                │
  ┌──────────────┐  ┌──────────────┐  ┌──────────────────┐
  │ Read/write   │  │ Read only    │  │ • Magnetic tape  │
  │ Memory       │  │ Memory       │  │ • Cassette tape  │
  │ RAM          │  │ ROM          │  │ • Floppy disk    │
  └──────────────┘  └──────────────┘  │ • Hard disk      │
    ┌─────┴─────┐         │           └──────────────────┘
┌─────────┐ ┌─────────┐   │
│ Static  │ │ Dynamic │   │
│ RAM     │ │ RAM     │   │
└─────────┘ └─────────┘   │
                   ┌───────┴────────┐
            ┌──────────────┐ ┌──────────────┐
            │ Erasable     │ │ Permanent    │
            │ • EPROM      │ │ • MASKROM    │
            │ • EEPROM     │ │ • PROM       │
            └──────────────┘ └──────────────┘
```
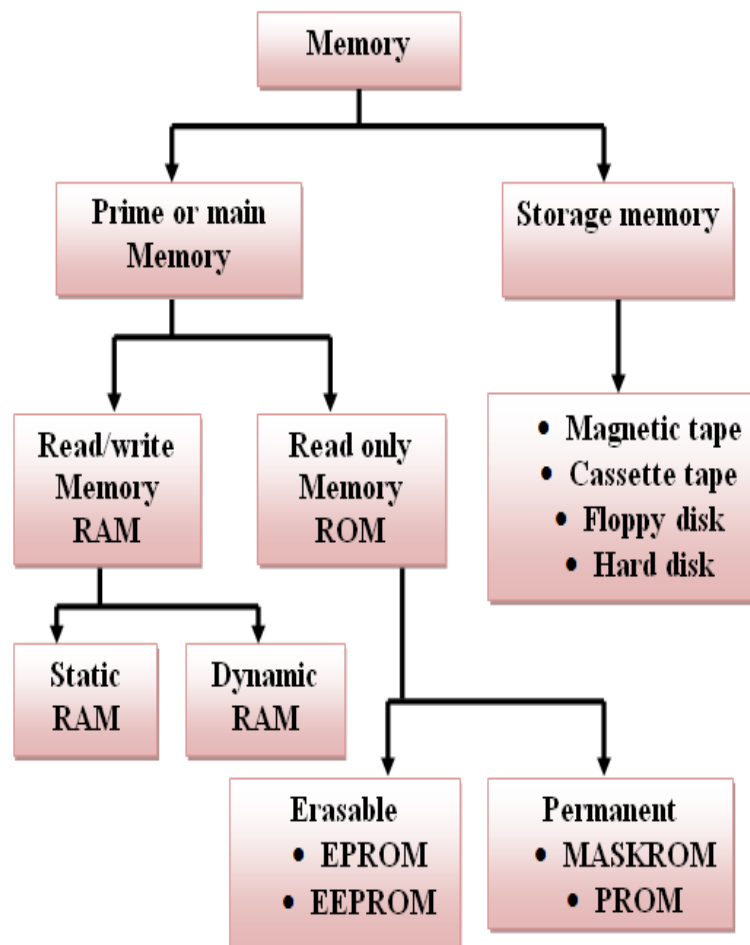
- RAM is volatile, meaning that when the power is turn off, all the contents are destroyed.
- ROM is nonvolatile, meaning that it retains the stored information even if the power is turn off.

Ex: suppose the ROM memory has 15 address lines and eight data line. How many bytes of information can be stored in the ROM? What is its total storage capacity?

Solution:

With 8 data lines, the number of bytes is equal to the number of locations, which is

2^{15} = 32,768 bytes

This gives a total storage of

32,768 * 8 = 262,144 bits

- **memory address lines:**

we need to know the relationship between the number of registers in memory chip and the number of address line

for a chip with 256 register we need 256 binary numbers to identify each register, Each address line can assume only two logic states (0 and 1), therefore we need to find the power of 2 that will give us 256 combinations, the problem can be restated as follows

find x where $2^x=256$ by taking the log of both side, we get

$\log 2^x =\log 256$

$x \log 2= \log 256$

$x=\log 256/\log 2=8$

the x represents the number of address lines needed to obtain 256 binary numbers.

- **Memory expansion**

In many applications, the microcomputer system requirement for memory is greater than what is available in a single device. There are two basic reasons for expanding memory capacity:

1. The byte-wide length is not large enough

2. The total storage capacity is not enough bytes.

Both of these expansion needs can be satisfied by interconnecting a number of ICs.
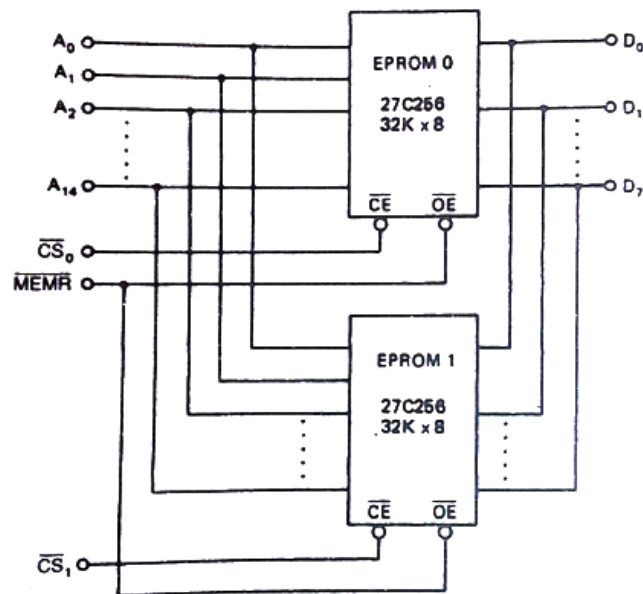
Ex: show how to implement 32K× 16 EPROM using two 32K×8 EPROM?
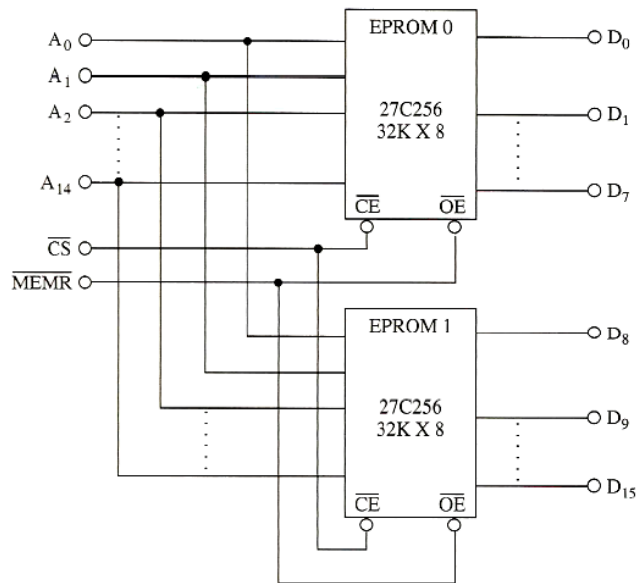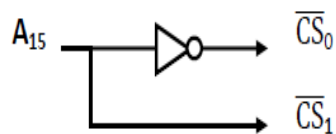
Solution:

Ex: show how to implement 64K× 8 EPROM using two 32K×8 EPROM?

Solution:

The $\overline{CS_1}$ and $\overline{CS_0}$ Signals could be implemented as follow:

- **Memory design:**

Example:

Design an interfacing memory circuit with a microprocessor have 16 addresses lines to meet the following specifications:

1. 74LS138: 3-to-8 decoder

2. 2732 (4K X 8): EPROM—address range should begin at 0000H and additional 4K memory space should be available for future expansion.

3. 6116 (2K X 8): CMOS R/W memory

Solution:

Example analysis:

o Given one 3-to-8 decoder without any additional gates, the decoder must be enabled using the $IO/\overline{M}$, and the $\overline{RD}$ and $\overline{WR}$ signals must be connected directly to the memory chips as necessary.

o The 2732 EPROM is a 4K memory chip that requires 12 address lines (A_{11}-A_0) therefore, only four high-order address lines (A_{15}–A_{12}) must be decoded. The 74LS138 decoder has three input lines and three enable lines (two active low and one active high). However, *we* have four high-order address lines and one $IO/\overline{M}$ signal. Therefore, three address lines can be connected as inputs, one address line and $IO/\overline{M}$ to the active low enable lines of the decoder, and the remaining enable line of the *decoder* must be connected high (+ 5V). The EPROM once programmed, needs only the $\overline{RD}$ signal. Therefore, the $\overline{RD}$ signal of the mp must be connected to the Output Enable ($\overline{OE}$) of the EPROM.

o To assign the starting address, 0000H, the address lines should have the following logic levels:

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

  0   0   0   0   0   0   0   0 0   0 0   0   0 0   0   0   = 0000 H

The logic levels of A_{15}–A_{12} should be zero. Therefore. A_{15} must be connected to the active low enable signal of the decoder. If the output Q_0 of the decoder is used to select the EPROM memory chip, address lines A_{14}–A_{12} will be at logic 0.

o   The 6116 is a 2K R/W memory chip and requires 11 address lines from A_{10} to A_0. To use the same decoder, address line A_{11} must be left as don't care. There is no specific requirement for the starting address for this memory except that 4K memory space must be left for additional expansion. Therefore, the output Q_1 of the decoder must be left for future expansion. We can use Q_2 to select the 6116 by connecting it to the $\overline{CE}$ signal. The $\overline{RD}$ and $\overline{WR}$ signals of the mp should be connected to the Output Enable ($\overline{OE}$)) and Write Enable ($\overline{WE}$) signals. respectively, of the 6116 R/W memory chip.

Circuit analysis:
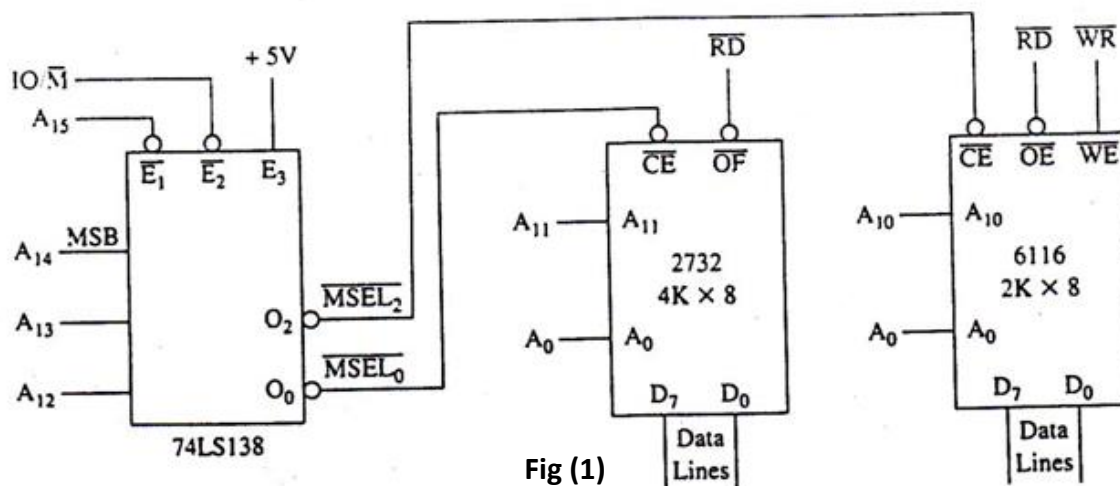
Fig (1)

Fig (1) shows the schematic of interfacing circuit based on the example analysis

1. When the logic levels of A_{15}–A_{12} are all 0's, and the processor asserts IO/$\overline{M}$ to read from memory the output Q_0 goes active and selects the chip. The address range of the EPROM is as follows:
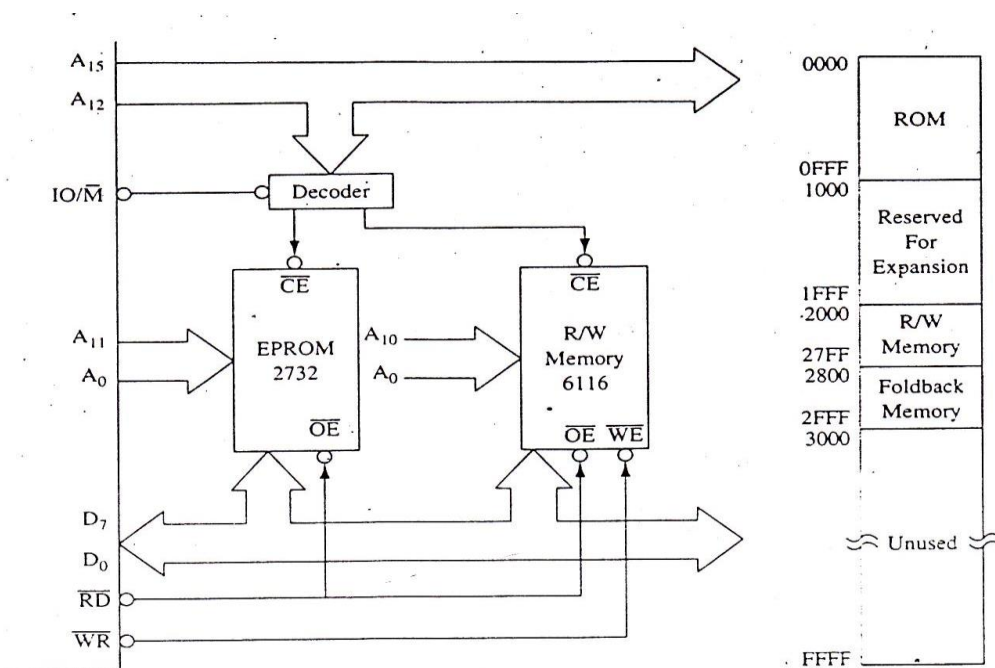
A_{15} A_{14} A_{13} A_{12}	A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0	
0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	=0000 H
MSEL$_0$	1 1 1 1 1 1 1 1 1 1 1 1	= 0FFF H

2. The 6116 R/W memory is selected by the output signal of the decoder Q_2, In this circuit the address line A_{11} is at don't care logic. Assuming A_{11} is at logic 0, the address range is as follows:

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	
0 0 1 0	0 0 0 0 0 0 0 0 0 0 0 0	=2000 H
MSEL$_2$	0 1 1 1 1 1 1 1 1 1 1 1	= 27FF H

Assuming A_{11} is at logic 1, the foldback (or mirror) address range is as follows:

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	
0 0 1 0	1 0 0 0 0 0 0 0 0 0 0 0	=2800 H
MSEL$_2$	1 1 1 1 1 1 1 1 1 1 1 1	= 2FFF H

Fig(2) –a System memory for this example -
fig (2)-b- system memory map

Ex: Design a 8086 memory system consisting of 1Mbytes, Using 64K×8 memory