



Memory Organization Interfacing and Memory Map

Second Stage Microprocessor Computer Engineering Department Lecture No. 4

Prepared by: Asst. Lect. Eng.
Hamza Waleed Hamza

Review of Memory Pin Connections Requirement

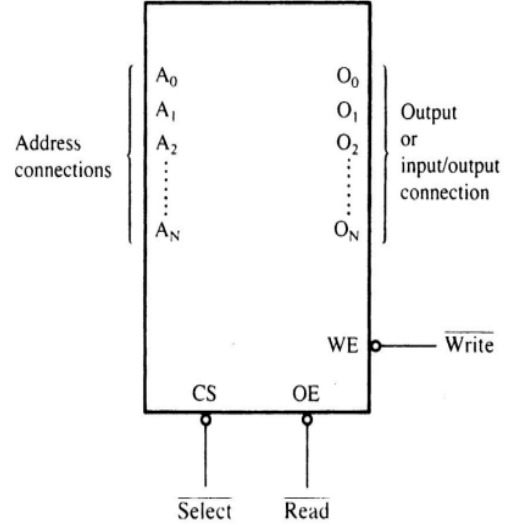
– Address inputs; Memory devices have address inputs to select a memory location within the device. Almost always labeled from $A_0 \dots A_N$

Ex: A 1K memory device has 10 address pins. therefore, 10 address inputs are required to select any of its 1024 memory locations.

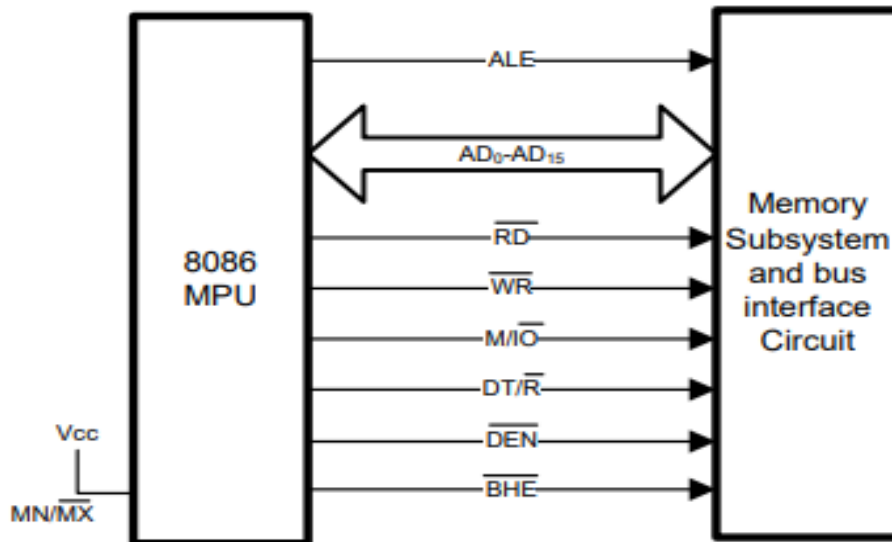
– Data outputs or input/outputs; Data pins on memory devices are labeled D0 through D7 for an 8-bit-wide memory device.

– Some type of selection; each memory device has an input that selects or enables the memory device. Sometimes more than one. This type of input is most often called a chip select (CS) chip enable.

_ Control Connections; All memory devices have some form of control input or inputs. Write enable must be active to perform memory write, and OE active to perform a memory read



8086 Minimum Mode Memory interfacing

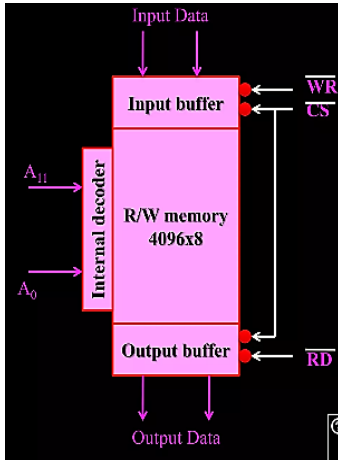


Address Decoding

• In order to attach a memory device to the microprocessor, it is necessary to decode the address sent from the microprocessor. The address line are divided into 2 groups;

A) The first group is usually connect directly to the memory chip through the address pins (usualy the lower address lines starting from A0) and the number of address lines is related to the memory capacity. Each memory chip has an internal decoder to select a unique memory location according to the combination of group address lines.

EX: A MP with 16 address lines A0-----A15 connected to a 4K byte of RAM

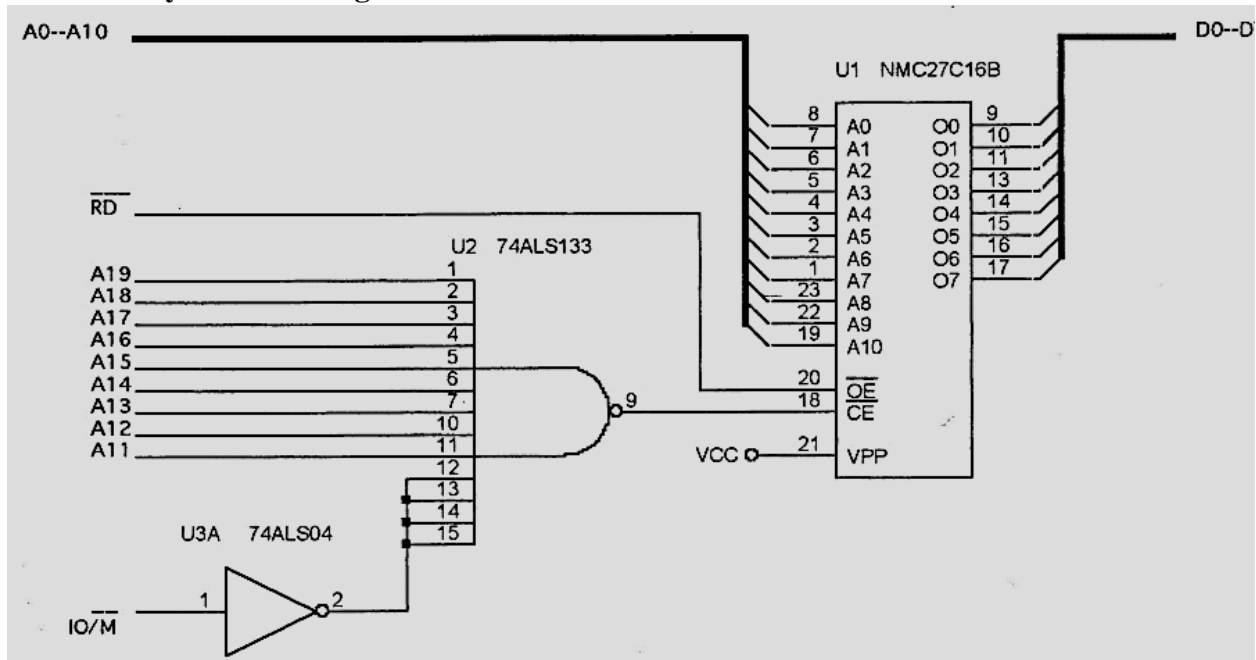


| | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|--------------------------|-----|-----|----|----|----|----|----|----|----|----|----|----|
| 1 st location | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 nd location | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 rd location | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ⋮ | | | | | | | | | | | | |
| Last location | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

The first group $2^{12} = 4096$

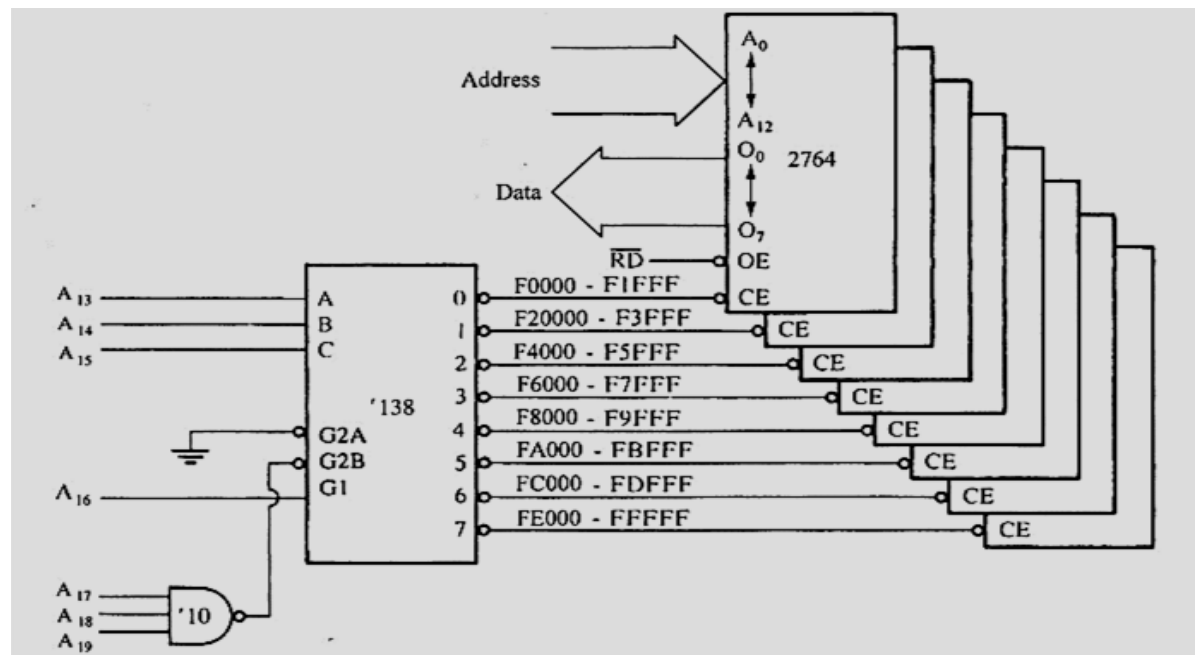
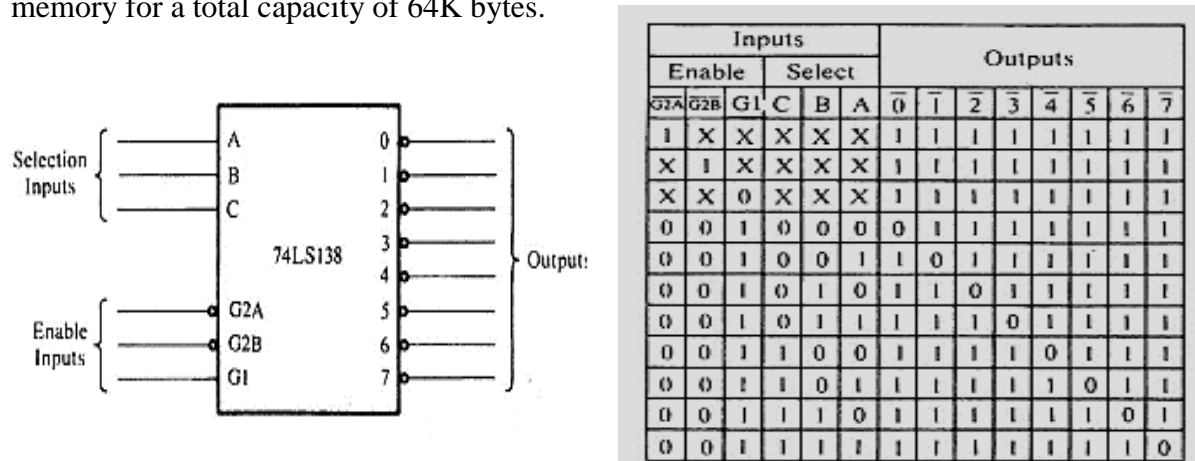
B) The second group (the higher address lines) which are the address lines remaining from address bus (16-12=4, A12-A15) are used to generate the chip select logic. Each memory chip is selected for the specified logic levels on these lines through an external decoder.

EX: A simple NAND gate decoder that selects 2K byte of EPROM connected to 8086 MP
The memory location range of addresses is FF800H–FFFFFH.



EX: Interfacing eight 2764 EPROMs memory chip to 8088 MP (8 bit data bus) , so that each chip is 8K byte.

A common integrated circuit decoder found in many systems is the 74LS138 3-to-8 line decoder. The outputs of the decoder in figure below are connected to eight different 2764 EPROM memory devices (select one at a time). The decoder selects eight 8K-byte blocks of memory for a total capacity of 64K bytes.



The three-input NAND gate is connected to address bits A19–A17. When all three address inputs are high, the output of this NAND gate goes low and enables input G2B of the 74LS138. Input G1 is connected directly to A16. In order to enable this decoder, the four address connections (A19–A16) must all be high for all memory chips.

- Accordingly the combination of A13-A15 will determine which chip to be selected. And A0-A12 will determine which location within the selected chip to be activated.

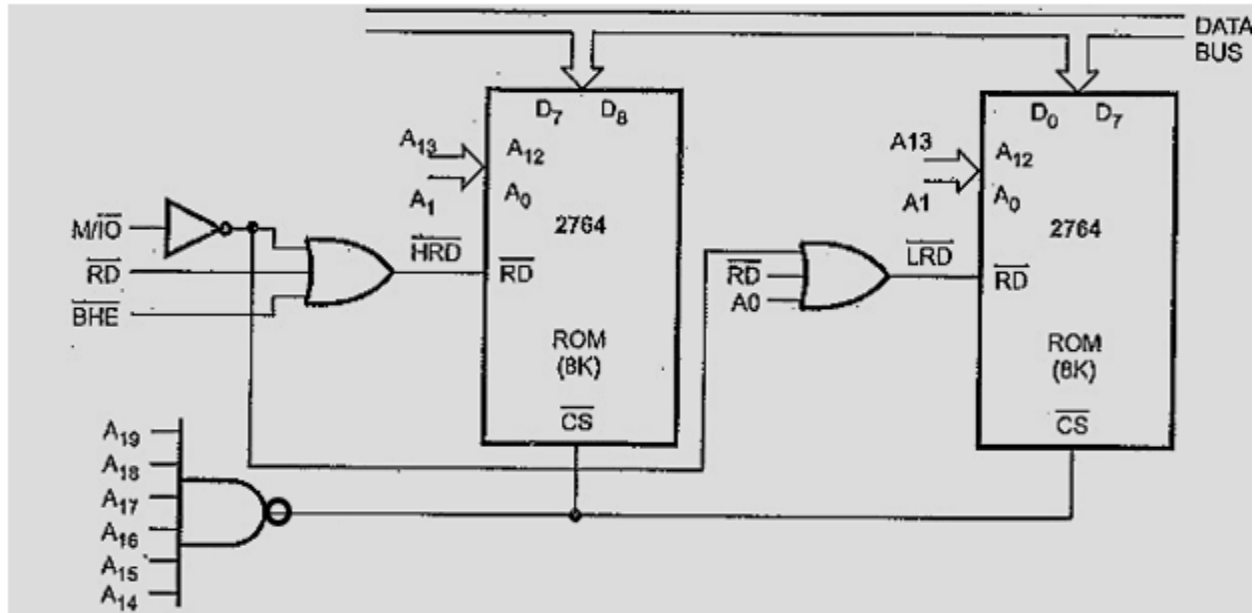
H/W: Determine the range of addresses for each EPROM if we connect A15 to G1 and A16 to the decoder input C

16 bit Data Bus Memory Interfacing

Bank selection is accomplished in two ways:

1. Separate decoders are used to activate each bank.

EX: 8086 MP interfacing with low and high bank of 8K byte ROM



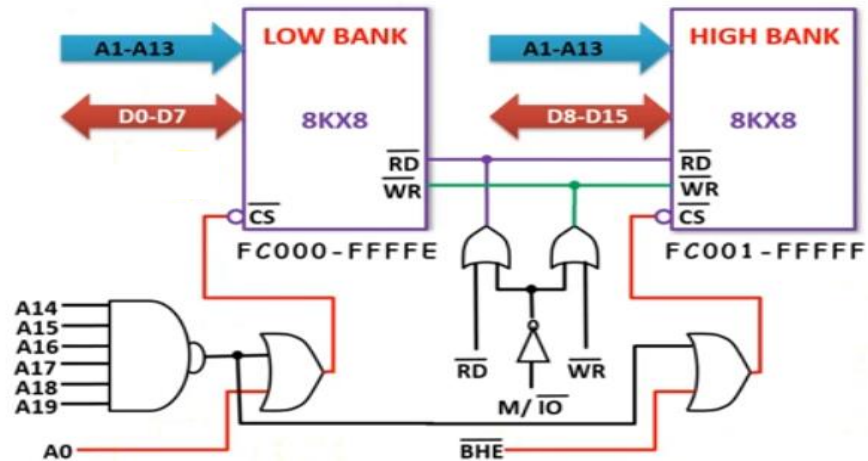
IF data is read from both bank, then $\overline{A0}$ and \overline{BHE} must be (0). Also the combination of A14-A19 must be set to activate the chip select of both banks.

| A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

The rang of addresses= FC000-----FFFFE

This type of decoding is called absolute decoding. Two 8K EPROMs (2764) are used to provide even and odd memory banks. Control signals \overline{BHE} and $\overline{A0}$ are used to enable outputs of odd and even memory banks respectively. As each memory chip has 8K memory locations, thirteen address lines are required to address each locations, independently starting from A1...A13. All remaining address lines are used to generate unique chip select signal. This addressing technique is normally used in large, memory systems.

Another connections



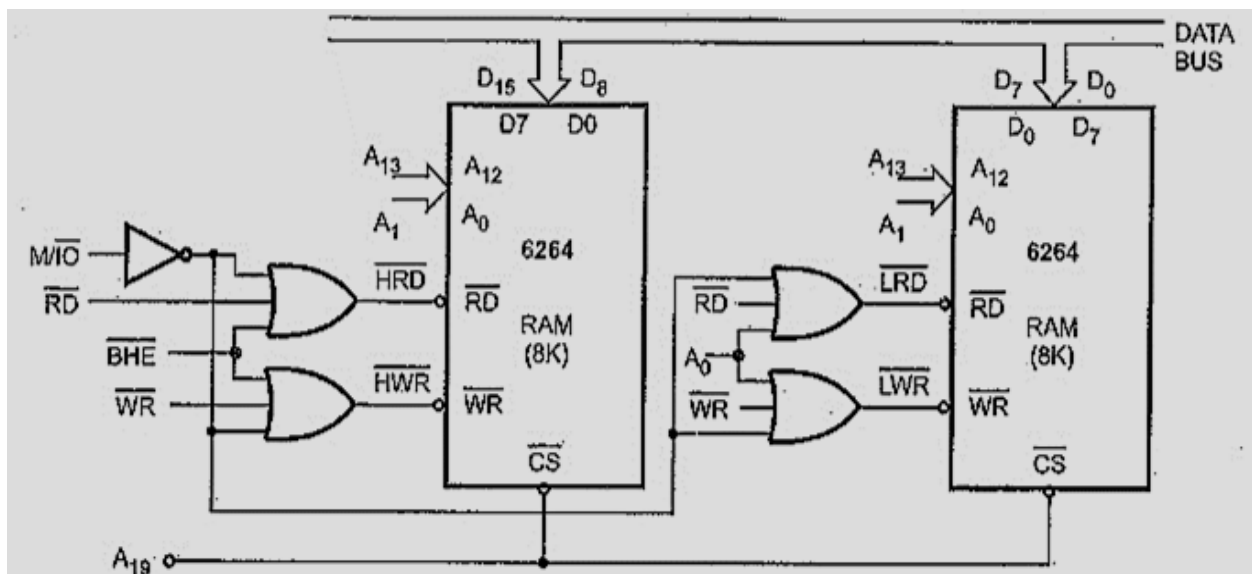
Absolute Decoding:

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

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. This technique is referred as Linear Decoding or Partial Decoding.

EX: 8086 MP interfacing with low and high bank of 8K byte RAM



IF data is read from both bank, then A0 and BHE must be (0). Also A19 must be (0) to activate the chip select of both banks. A14-A18 are not connected, accordingly any logic state of these line

will not effect. Hence these line are considered don't care conditions. This type of decoding is known as Linear Decoding. This gives you multiple addresses (shadow addresses). This technique reduces the cost of Address Decoding Techniques in 8086 Microprocessor circuit, but it has drawback of multiple addresses.

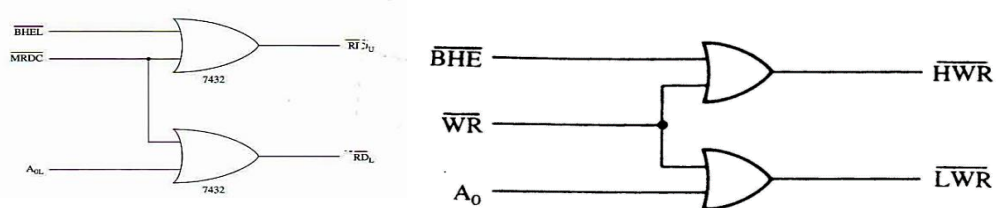
| | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|----|
| A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| 0 | x | x | x | x | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | x | x | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

H/W : Determine the rang of addresses

H/W : Determine the rang of addresses if A16 is connected to both CS instead of A19

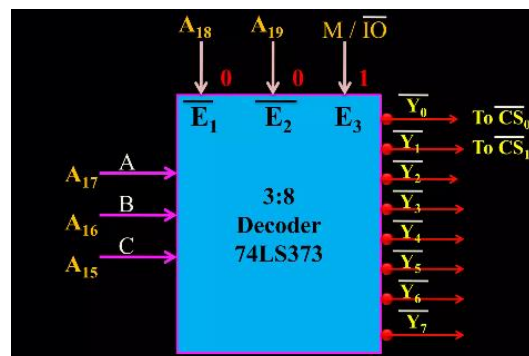
H/W : Determine the rang of addresses if both memory chip are reduced to 4KBytes

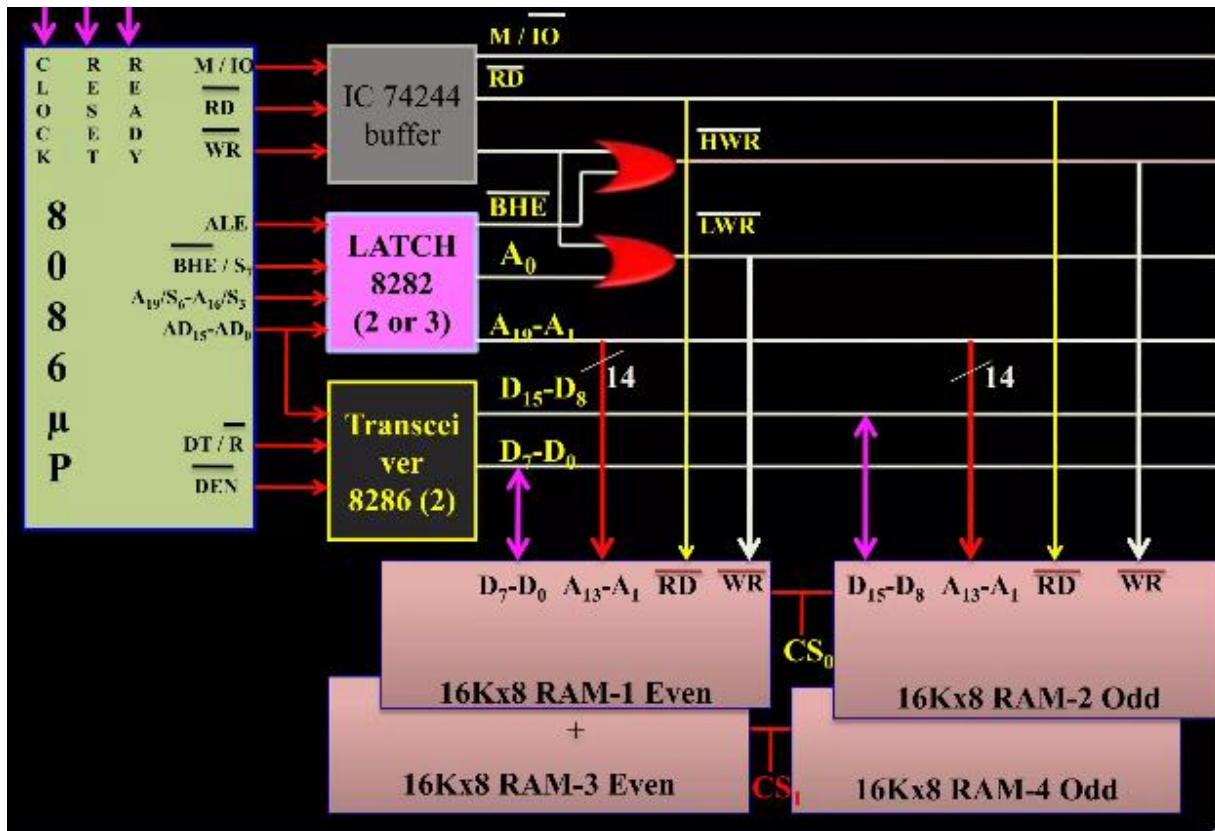
2. Separate control signal is developed to select a write/read to each bank of the memory.



Ex: Interface 32Kword of memory to 8086 MP system, where the avialable chips =16x8bit RAM
 Total memory=32K word= 32x2K= 64K
 Number of IC required=64/16 = 4 ICs, and each IC required 15 address lines

| Even bank | Odd bank |
|-------------|-------------|
| RAM_1 (16K) | RAM_2 (16K) |
| RAM_3 (16K) | RAM_4 (16K) |





The range of addresses is as below:

| MEMORY IC | HEX ADDRESS | BINARY 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 |
| 16 K x 8 RAM-(1) | 00000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 07FFE | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 16 K x 8 RAM-(3) | 08000 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0FFFE | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

Ex: Design a 8086 memory system consisting of 1Mbytes, Using 64Kx8 memory

