



Computer Engineering

Electrical Engineering Department

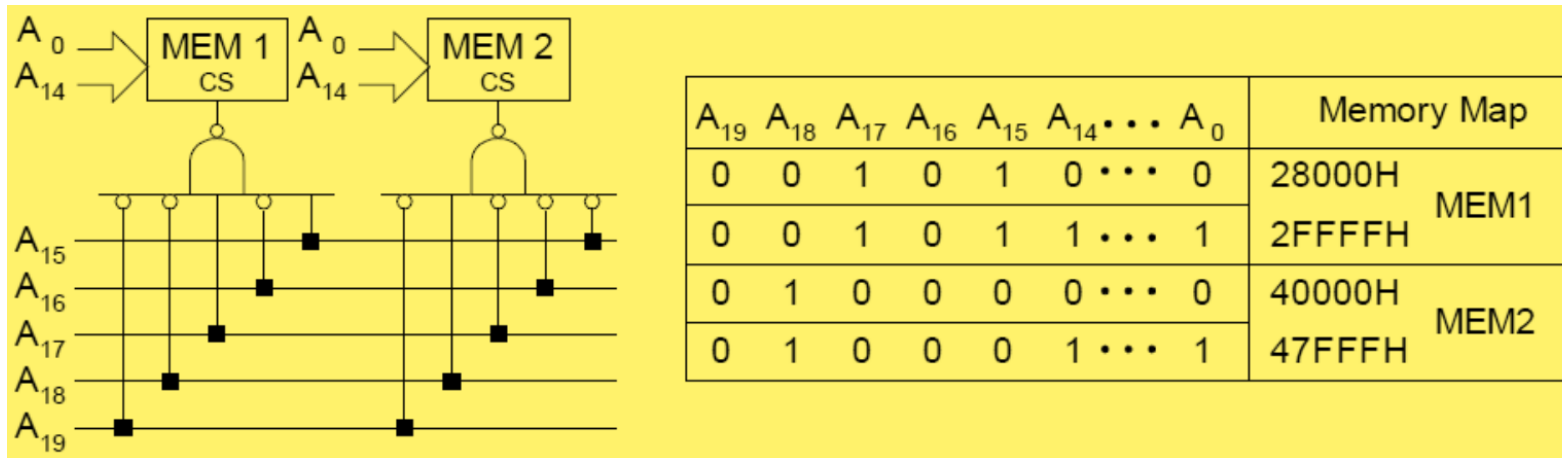
Third Stage

Instructor: Asst. Lecturer

Adnan Ali Abdullah

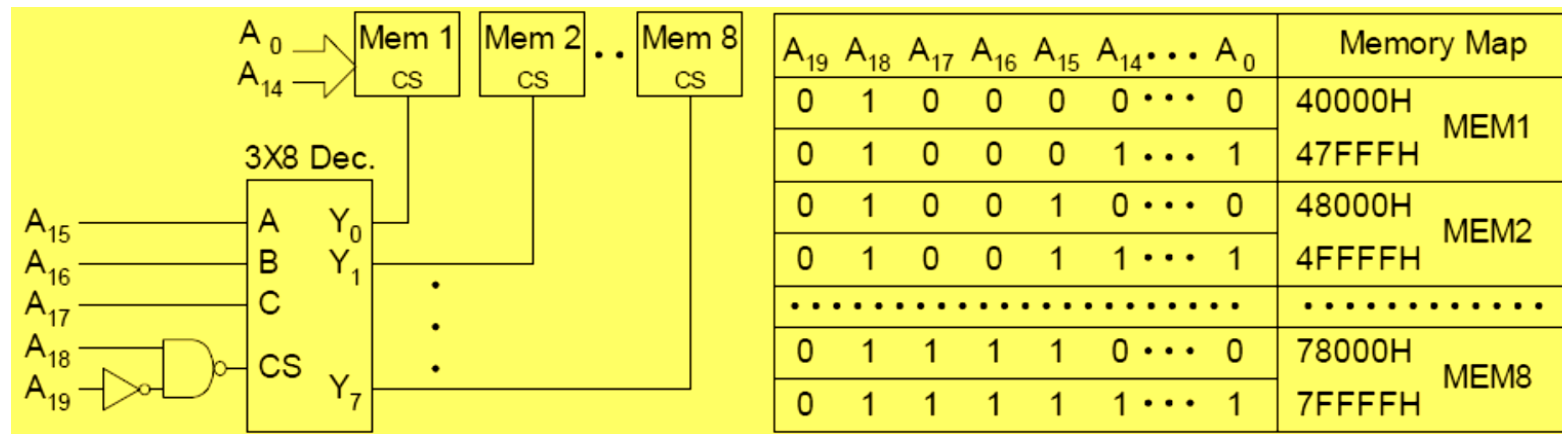
Address Decoding Circuits Using Only NAND Gates

- A single NAND gate is used to decode each memory device. The inputs of the NAND gate can be connected on the address lines either directly, or through inverters, according to the required memory map.
- This method has the advantage that it adds a short time delay in the memory path.
- The disadvantage of this circuit is that too many gates (NAND and NOT) are needed for memory systems that have a few memory chips. This increases the cost of the system, adds to the complexity of the PCB board (too many chips and lines).
- The figure below shows a memory map of two memory chips. Lines A0 – A14 are connected to each chip and lines A15 – A19 are connected to NAND gates to generate the CS signal.



Address Decoding Circuits Using decoders and a NAND Gate

- One or more line decoders such as the 74LS139 (2 x 4 decoder) or the 74LS138 (3 x 8 decoder) are used to decode(enable) one out of a number of memory device. The CS inputs of the decoders are enabled by a NAND gate, according to the required memory map.
- This decoding circuit has the disadvantage that it adds at least three gate delays in the memory path.
- The advantage of this circuit is that less gates (NAND, NOT and decoders) are needed for memory systems that have a number memory chips.
- The figure below shows a memory map of 8 memory chips. The decoding circuit has one decoder, one inverter and one NAND gate.



Example: Show how a 128 Kbyte RAM module can be connected on an 8088 μ P using SRAM chips 32K x 8, occupying the address range starting from the address C0000H. Use the following address decoding circuits:

1. NAND gates 2. Line decoders

Solution:

Number of chips needed: $128\text{K} / 32\text{K} = 4$ chips

Number of address lines: $32\text{K} = 2^5 * 2^{10} = 2^{15} \rightarrow 15$ address lines ($A_0 - A_{14}$)

Lines $A_{15} - A_{19}$ are used to generate the CS signals

The final address = the starting address + memory size – 1

RAM1 : final address = $\text{C0000H} + 8000\text{H} - 1\text{H} = \text{C7FFFH}$

RAM2 : final address = $\text{C8000H} + 8000\text{H} - 1\text{H} = \text{CFFFFH}$

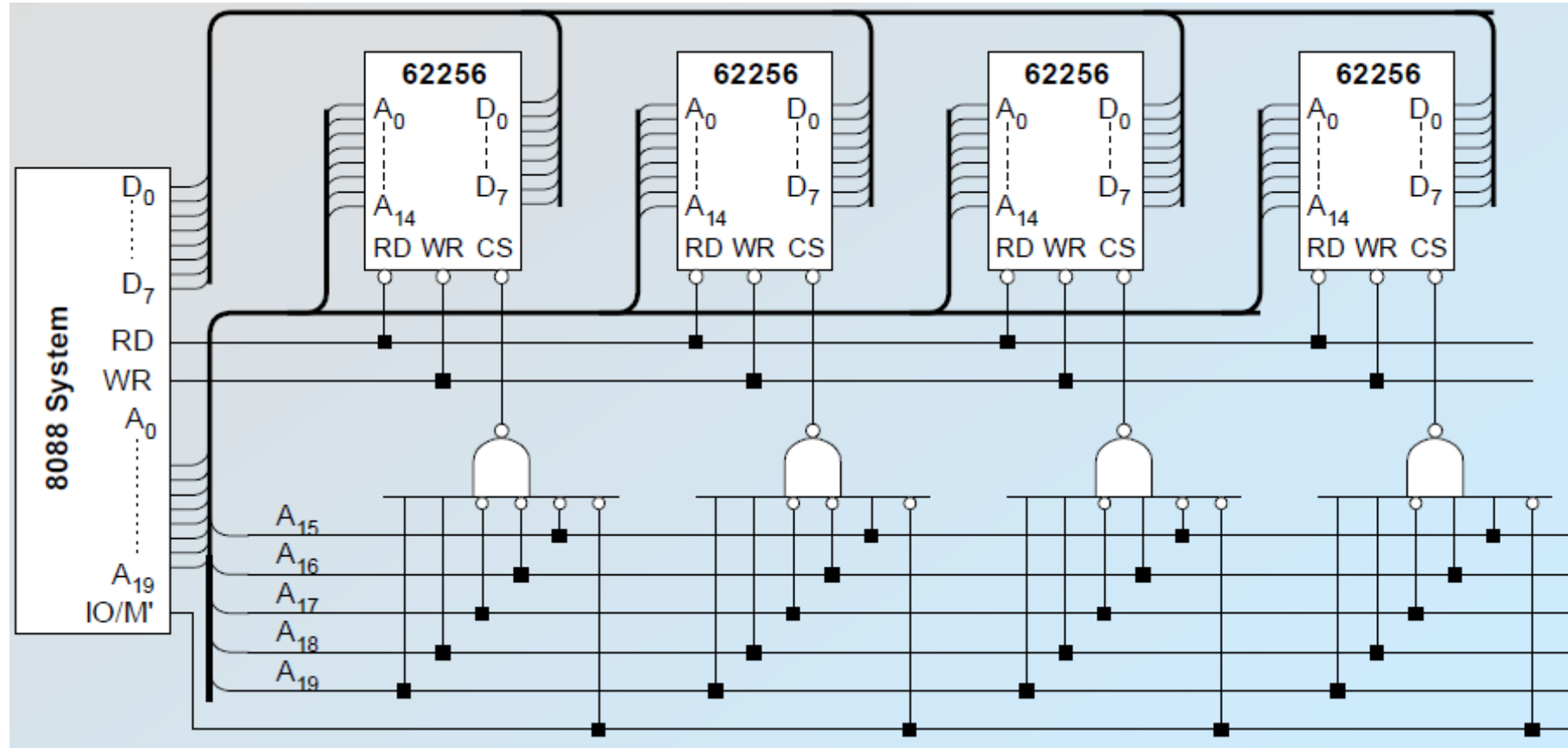
RAM3 : final address = $\text{D0000H} + 8000\text{H} - 1\text{H} = \text{D7FFFH}$

RAM4 : final address = $\text{D8000H} + 8000\text{H} - 1\text{H} = \text{DFFFFH}$

A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	$\dots$	A_0	Memory Map
1	1	0	0	0	0	0	$\dots$	0	C0000 C7FFF RAM1
1	1	0	0	0	1	1	$\dots$	1	
1	1	0	0	1	0	0	$\dots$	0	C8000 CFFFF RAM2
1	1	0	0	1	1	1	$\dots$	1	
1	1	0	1	0	0	0	$\dots$	0	D0000 D7FFF RAM3
1	1	0	1	0	1	1	$\dots$	1	
1	1	0	1	1	0	0	$\dots$	0	D8000 DFFFF RAM4
1	1	0	1	1	1	1	$\dots$	1	

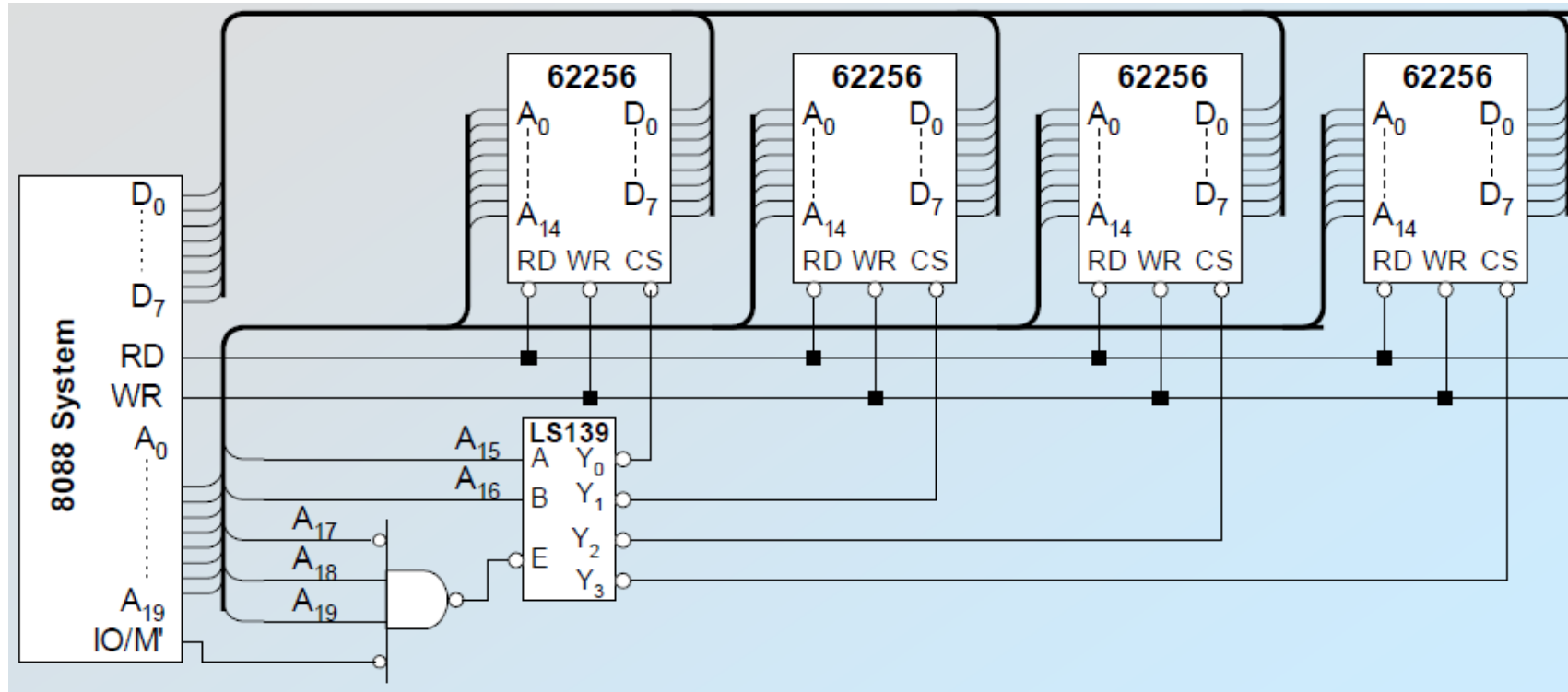
1. Using NAND gates

In this case the lines $A_{15} - A_{19}$ are connected to four NAND gates with $IO/\bar{M}$ control signal sent by the 8088 μ P to generate the CS signal as shown below.



2. Using a line decoder and a NAND GATE

In this case, the lines A_{15} and A_{16} are input to the decoder to select one of the four chips, while the lines $A_{17} - A_{19}$ and $IO/\overline{M}$ signal are connected to a NAND gate to enable the decoder.



H.W.

Show how a 32Kbyte ROM module can be connected on an 8088 system using 2764 EPROM chips (8K x 8), occupying the address range starting from the address E0000H. Use the following address decoding circuits:

1. NAND gates
2. A line decoder and a NAND gate

Example:

Design an 8K x 8 RAM module using 2K x 8 RAM chips. The module should be connected on an 8-bit processor with a 16-bit address bus, and occupy the address range starting from the address A000. Show the circuit and the memory map.

Solution:

Number of memory devices needed = $8K / 2K = 4$

Decoder needed = 2×4

Number of address lines on each 2K x 8 memory chip = 11

$$2^m = 2K = 2^1 \times 2^{10} = 2^{11} \rightarrow (A_0 - A_{10})$$

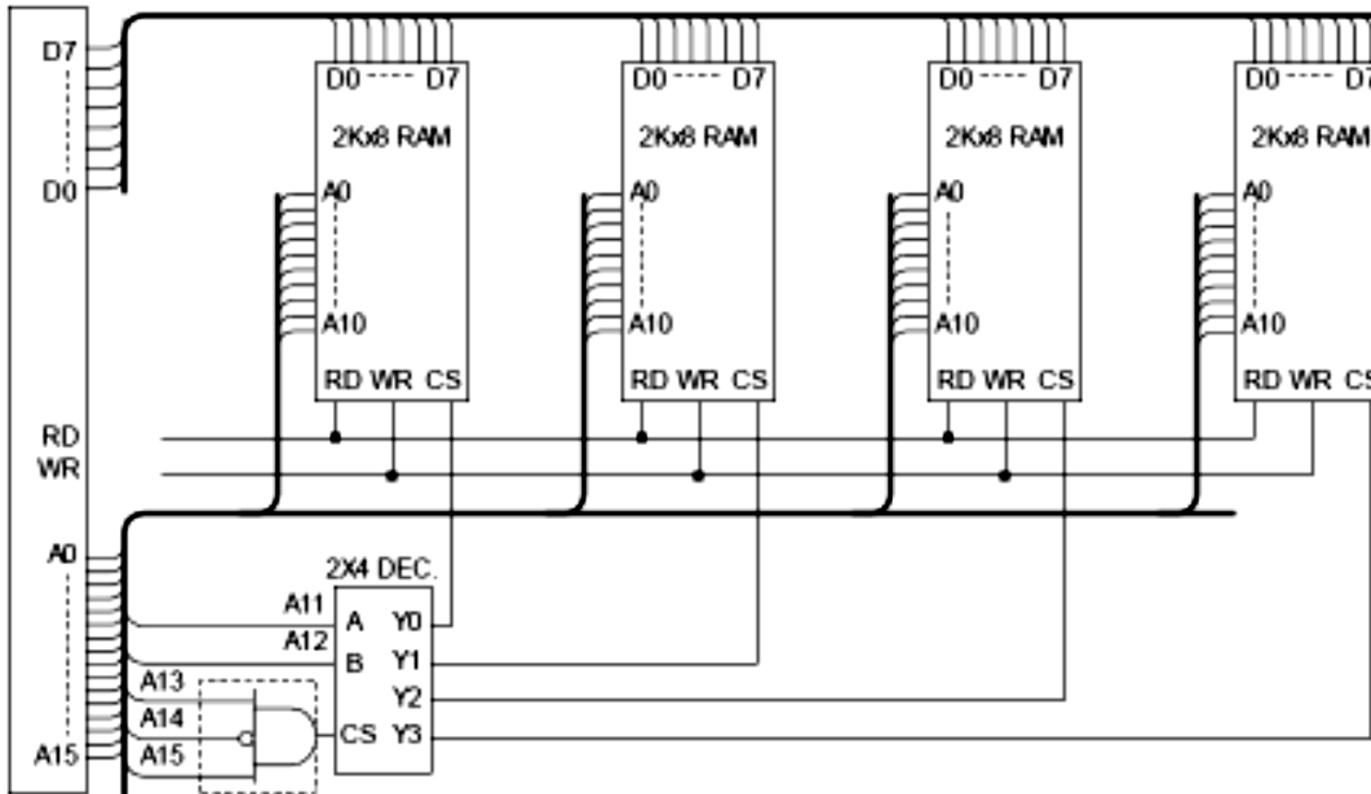
Two address lines are needed for the decoder $\rightarrow (A_{11}, A_{12})$

Number of address lines needed for the address selection circuit
 $= 16 - 11 - 2 = 3 \rightarrow (A_{13}, A_{14}, A_{15})$

Starting Address = A000 = 1010-0000-0000-0000
 $\Rightarrow A_{15} = 1, A_{14} = 0$ and $A_{13} = 1$



A15	A14	A13	A12	A11	A10	...	A0	Mem. Map
1	0	1	0	0	0	...	0	A000
1	0	1	0	0	1	...	1	A7FF
1	0	1	0	1	0	...	0	A800
1	0	1	0	1	1	...	1	AFFF
1	0	1	1	0	0	...	0	B000
1	0	1	1	0	1	...	1	B7FF
1	0	1	1	1	0	...	0	B800
1	0	1	1	1	1	...	1	BFFF



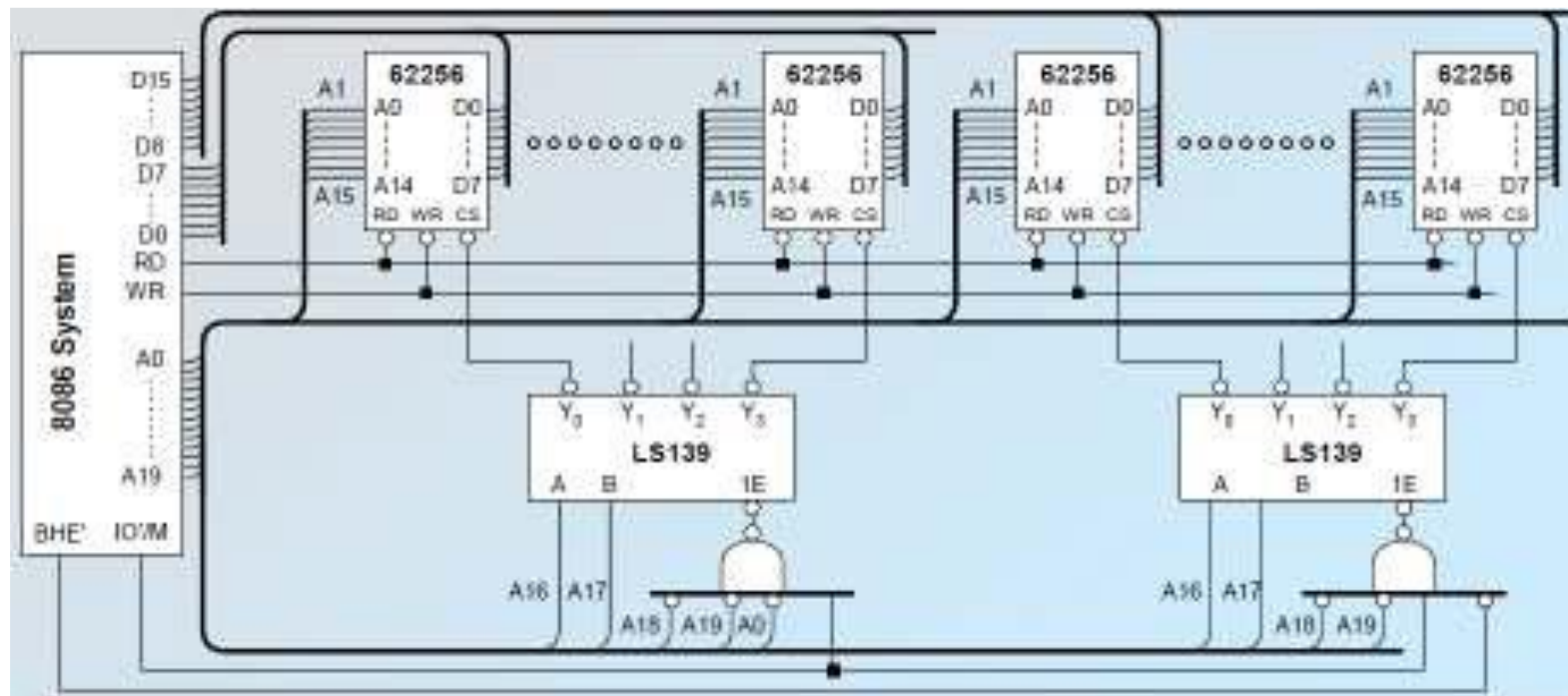
16-bit Microprocessors Memory Interfacing (8086μP)

To design memory interface circuit with the 8086μP, bank selection is accomplished in two ways:

1. Separate decoders are used for each bank of the memory.
2. Separate write signal is generated to select a write signal to each bank of the memory.

1. 16-bit Memory Interfacing Using Separate Bank Decoders

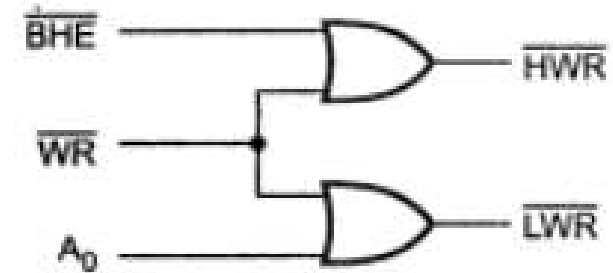
The figure below shows a memory map with 16-bit microprocessor (8086 μ P). Four memory chips of 32k x 8 are used, two of them represent the low bank and the other represent the high bank. The first decoder (left side) is enabled when A_0 is zero, thus it is enable with even addresses. Thus the data lines of the memory devices decoded by this decoder must be connected on the processor's data lines $D_0 - D_7$. The second decoder (right side) is enabled when $\overline{BHE}$ is zero, thus it is enable with odd addresses. Thus the data lines of the memory devices decoded by this decoder must be connected on the processor's data lines $D_8 - D_{15}$



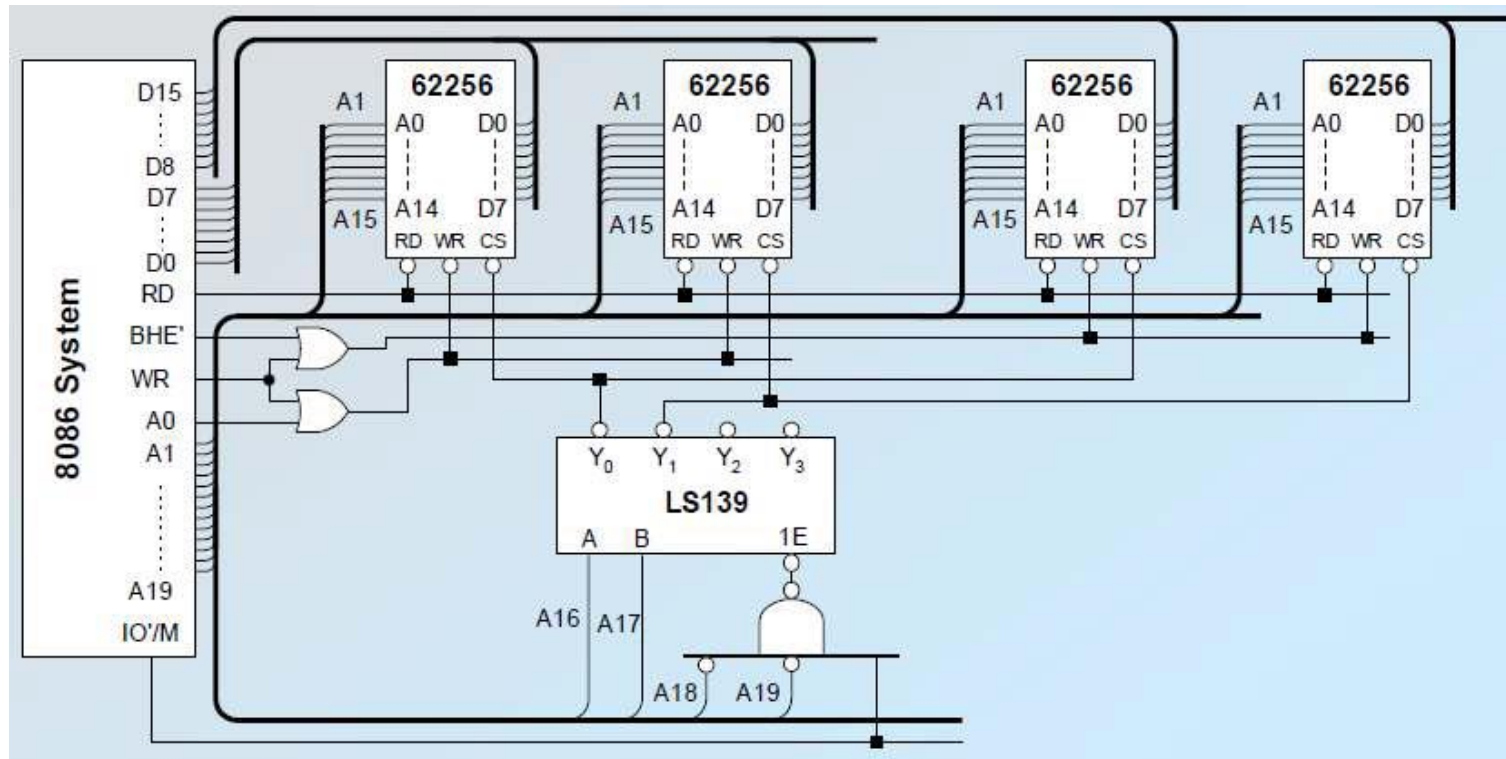
2. 16-bit Memory Interfacing Using Separate Write Signals

In this method the decoder always enables both banks. On a memory read operation, the data from both banks is loaded on the data bus. The microprocessor selects internally the appropriate bank, according to the instruction being executed. On a memory write operation, only the $\overline{WR}$ signal of the appropriate bank is enabled, thus data is copied only in the appropriate memory chip.

By providing separate write signals, we require one decoder to select a 16-bit wide memory. The signal $\overline{HWR}$ is activated only when $\overline{BHE}$ and $\overline{WR}$ both are activated, thus it is used with memory chips from odd memory bank. The signal $\overline{LWR}$ is activated only when A_0 and $\overline{WR}$ both are activated, thus it is used with memory chip from even memory bank.



Note: It is not necessary to generate separate read signals for two memory banks. Read memory signal always put 16-bit data on the data bus, but the CPU reads data according to 8-bit read operation or 16-bit read operation.



Example: Design an 8086 μ P based system 128Kbyte memory (64 Kbyte EPROM and 64 Kbyte RAM). Draw the complete schematic of the design indicating the address map. Assume the starting Address of the EPROM is 30000H and for the RAM is 00000H.

Solution:

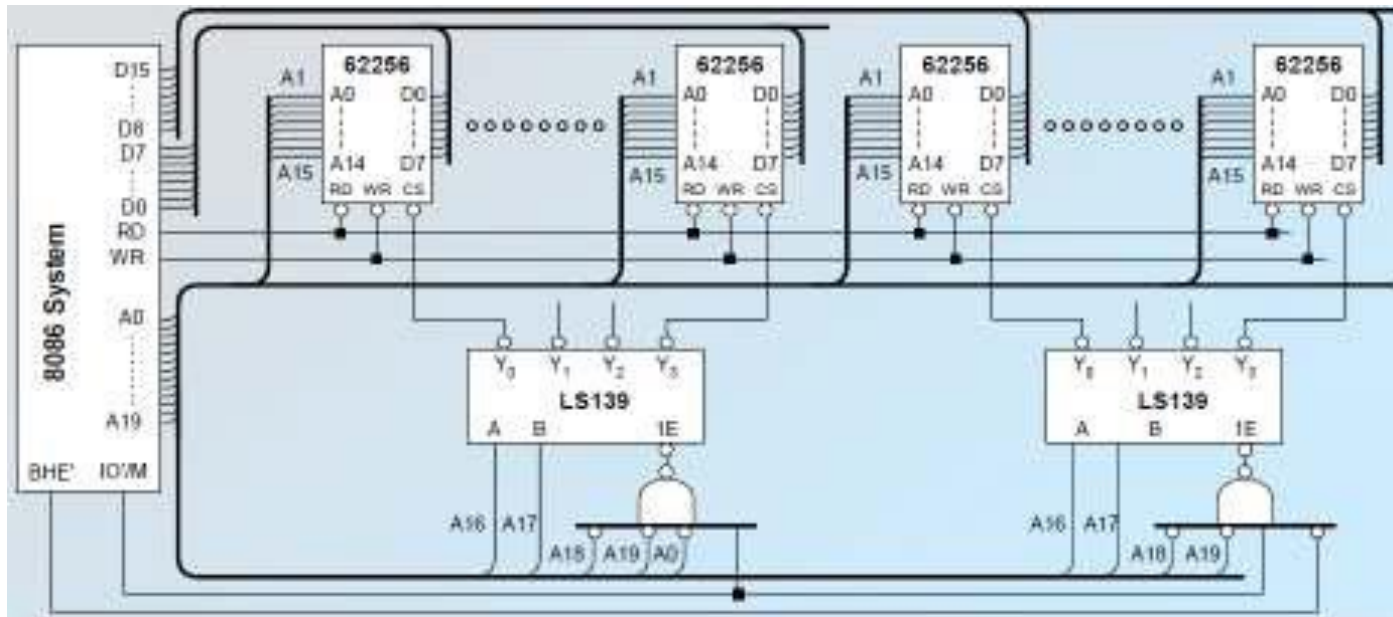
The design can be achieved by using two 32Kbyte EPROMs (one for odd bank and another for the even bank) and two 32Kbyte RAMs (one for odd bank and another for the even bank).

Each RAM and EPROM chip needs 15 address lines $A_1 - A_{15}$ because $32\text{Kbyte} = 2^5 * 2^{10} = 2^{15}$. A_0 and $\overline{BHE}$ are used to select even and odd memory banks respectively.

The memory map is as shown in the table below :

The schematic diagram is as shown below.

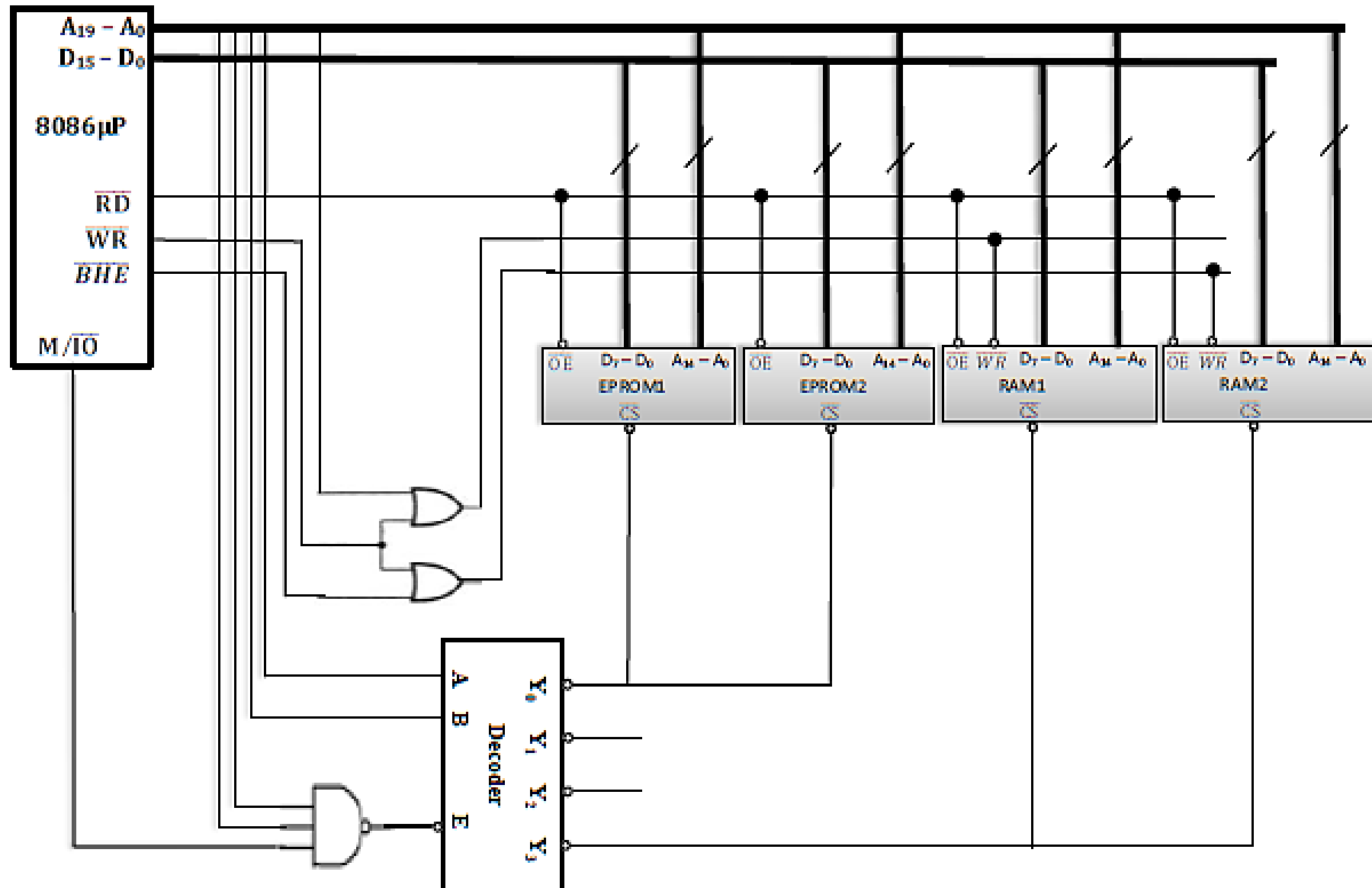
$\overline{BHE}$	$A_{19}A_{18}A_{17}A_{16}$	$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$	Address	Memory
1	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	30000H	Even(Low)
1	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0	3FFFEH	EPROM1
0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	30001H	Odd(High)
0	0 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	3FFFFH	EPROM2
1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	00000H	Even(Low)
1	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0	0FFFEH	RAM1
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	00001H	Odd(High)
0	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	0FFFFH	RAM2



Example: Design an 8086 μ P based system 128Kbyte memory (64 Kbyte EPROM and 64 Kbyte RAM). Draw the complete schematic of the design indicating the address map. Assume the starting Address of the EPROM is F0000H and for the RAM is C0000H.

Each RAM and EPROM chip needs 15 address lines $A_1 - A_{15}$ because $32\text{Kbyte} = 2^5 * 2^{10} = 2^{15} A_0$ and $\overline{BHE}$ are used to select even and odd memory banks respectively. The memory map is as shown in the table below :

$\overline{BHE}$	$A_{19}A_{18}A_{17}A_{16}$	$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$	Address	Memory
1	1 1 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	F0000H	Even
1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0	FFFFEH	EPROM1
0	1 1 1 1	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	F0001H	Odd
0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	FFFFFH	EPROM2
1	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	C0000H	Even
1	1 1 0 0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0	CFFFEH	RAM1
0	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	C0001H	Odd
0	1 1 0 0	1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1	CFFFFH	RAM2



H.W: Design an 8086 μ P based system 256Kbyte memory (128 Kbyte EPROM and 128 Kbyte RAM), the size of the RAM and EPROM chips is 32 Kbyte. Draw the complete schematic of the design indicating the address map. Assume the starting Address of the EPROM is F0000H and for the RAM is 30000H.