
Lecture XIV 8086 Memory Interface

Wisam I Hasan

Objectives:

Upon completion you will be able to:

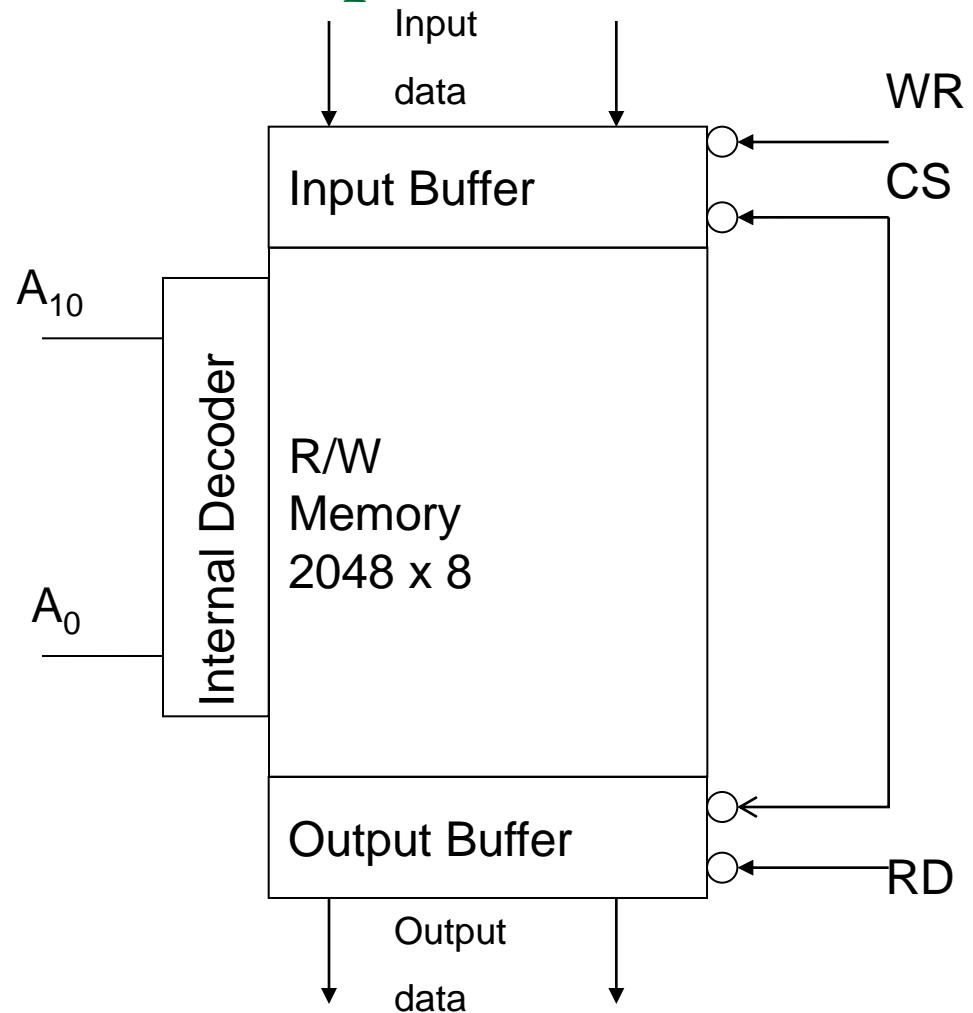
1. How to interface a memory device with μp
2. How to access memory in READ or WRITE
3. Describe memory structure with all its type: RAM, EPROM etc...
4. Memory requirements
5. Address Decoding types
6. Interfacing examples

Terminology and Operations

- Memory are made up of (**registers**).
- Each register consists of one storage location
- Each location consists of an **address**
- The number of storage locations from few hundreds to several mega or giga locations
- The total number of memory storage is called memory **capacity** and measured in Bytes
- Each register consists of storage element (FF, capacitor for semiconductor)
- A storage element is called **cell**
- The data could be read from or written to memory

Memory Structure and its requirements

- As mentioned earlier, read/write memories consist of an array of registers, in which each register has unique address
- The size of the memory is $N \times M$ as shown below where N is the number of registers and M is the word length, in number of bits¹



Logic Diagram for RAM

Example 1

If memory is having 12 address lines and 8 data lines, then Number of registers/
memory locations (capacity) = $2^N = 2^{12}$
= 4096

Word length = M bit
= 8 bit

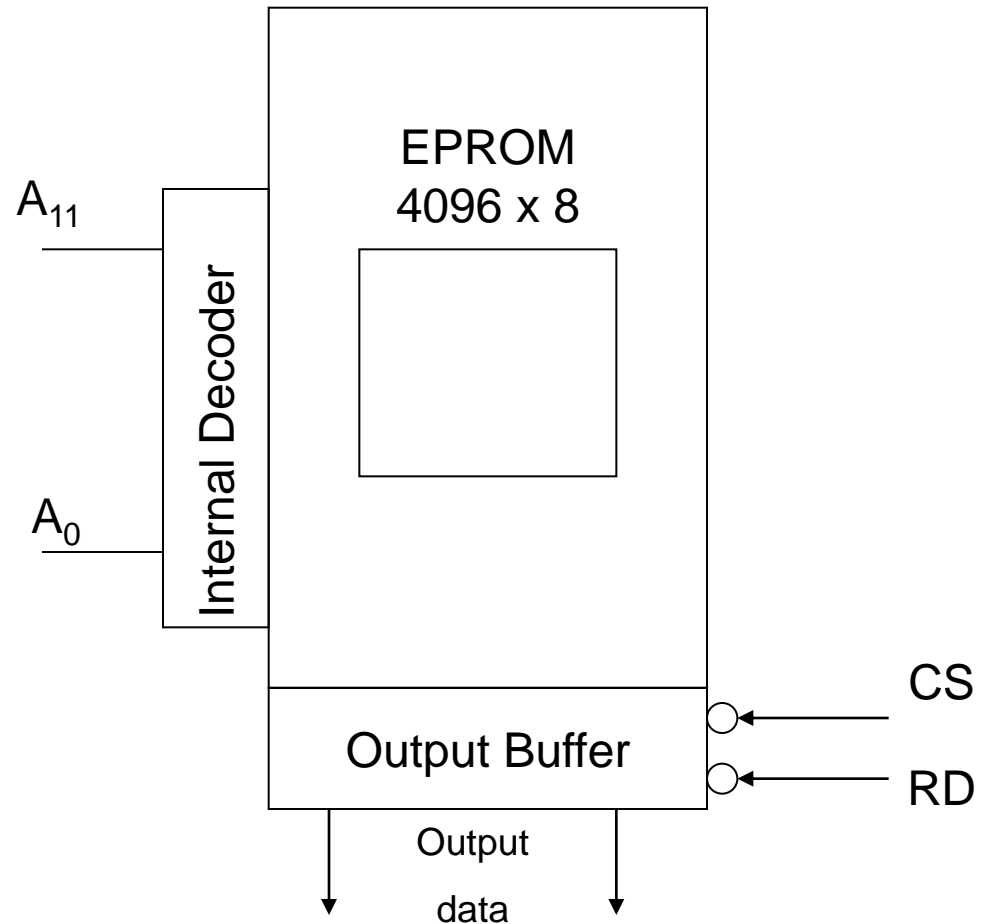
Example 2: if memory has 8192 memory locations, then it has 13 address lines.
(How?)

Table 14.1 summarizes capacity with address

Memory Capacity	Address lines required
1k = 1024 memory locations	10
2k = 2048 memory locations	11
4k = 4096 memory locations	12
8k = 8192 memory locations	13
16k = 16384 memory locations	14
32k = 32768 memory locations	15
64k = 65536 memory locations	16

EPRM layout

- Shows the logic diagram of typical EPROM (Erasable Programmable Read-Only Memory) with 4096 (4k) registers
- It has 12 address lines (A0-A11), one chip select (CS), one Read control signal.
- No WR signal, why?



Basic Memory Interfacing with 8085

- For interfacing memory devices to μp 8085, keep the following points in your mind:
- μp 8085 can access 64KB memory since address bus is 16-bit.¹
- Generally EPROM (or EPROMs) is used as a program memory and RAM (or RAMs) as data memory.²
- The capacity of program memory and data memory depends on the application.
- It is not always necessary to select 1 EPROM and 1 RAM. We can have multiple EPROMs and multiple RAMs as per the requirement of application

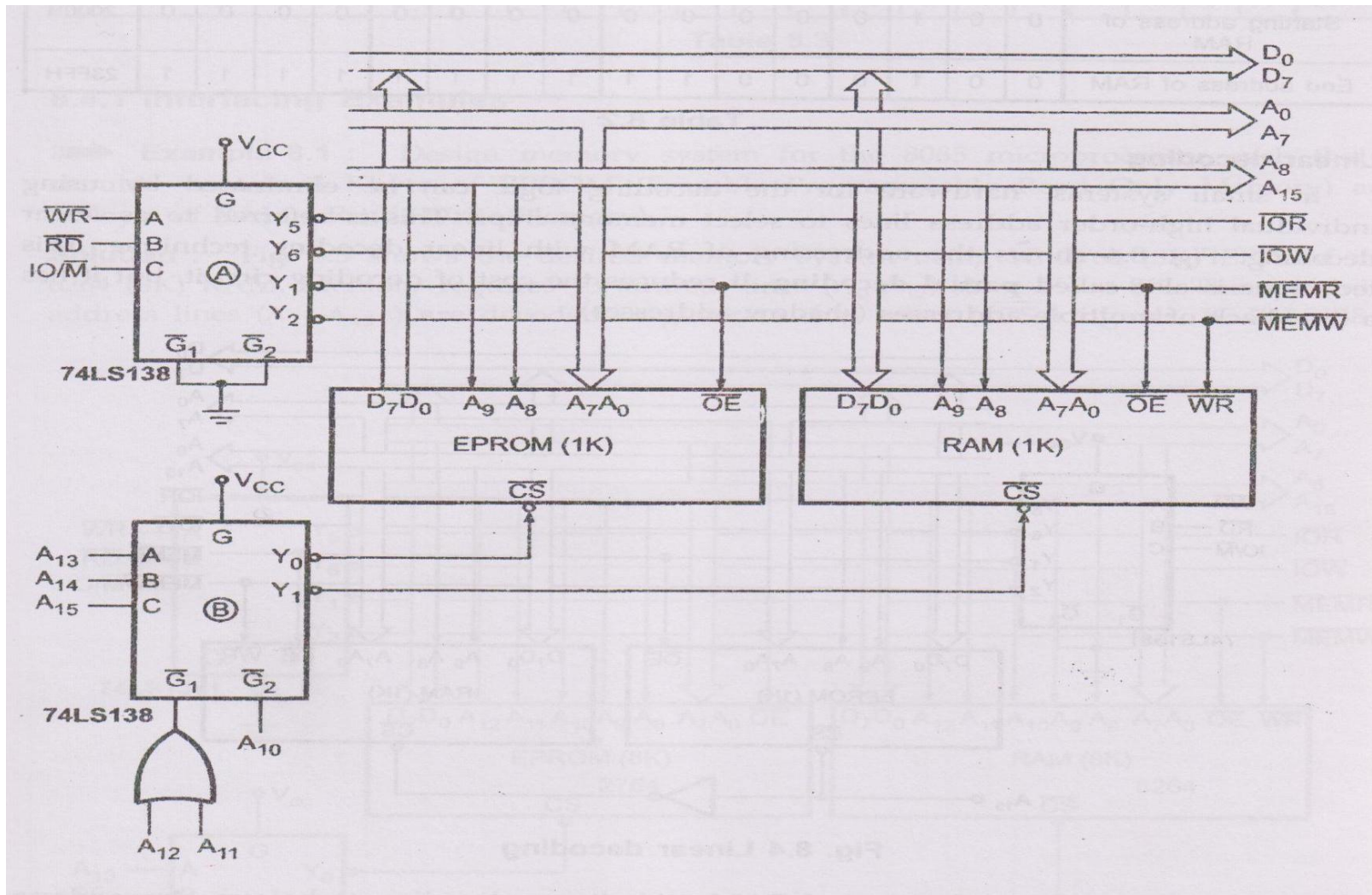
For Example

- We have to implement 32 KB of program memory and 4KB EPROMs are available. In this case we can connect 8 EPROMs in parallel.¹
- We can place EPROM/RAM anywhere in full 64 KB address space. But program memory (EPROM) should be located from address 0000 H.²
- It is not always necessary to locate EPROM and RAM in consecutive memory address.³
- The memory interfacing requires to:
 - Select the chip
 - Identify the register
 - Enable the appropriate buffer.
- μ p system includes memory and I/O devices.
- It is important to note that μ p can communicate (read/write) with only one device at a time, so address decoding needed.

Address Decoding techniques

- There are two main techniques:
 - Absolute decoding/ Full Decoding
 - Linear decoding / Partial Decoding
- Absolute Decoding:
- All the higher address lines are decoded to select the memory chip, and the memory chip is selected only for the specified logic level on these high-order address, no other logic levels can select the chip.
- The following figure shows the memory interface with absolute decoding. This addressing technique is normally used in large memory systems.

Absolute Decoding Technique



Memory map

Memory Map :

Memory ICs	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	Address
Starting address of EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End address of EPROM	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	03FFH
Starting address of RAM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
End address of RAM	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	23FFH

Table 8.2

Linear Decoding

- In small systems, h/w for the decoding logic can be eliminated by using individual high-order address lines to select memory chips.
- This is referred to as linear decoding.
- The figure below shows the addressing of RAM with linear decoding technique.
- This technique is also called partial decoding.
- It reduces the cost of the decoding cct., but it has a drawback of multiple address (shadow addresses)

Continued

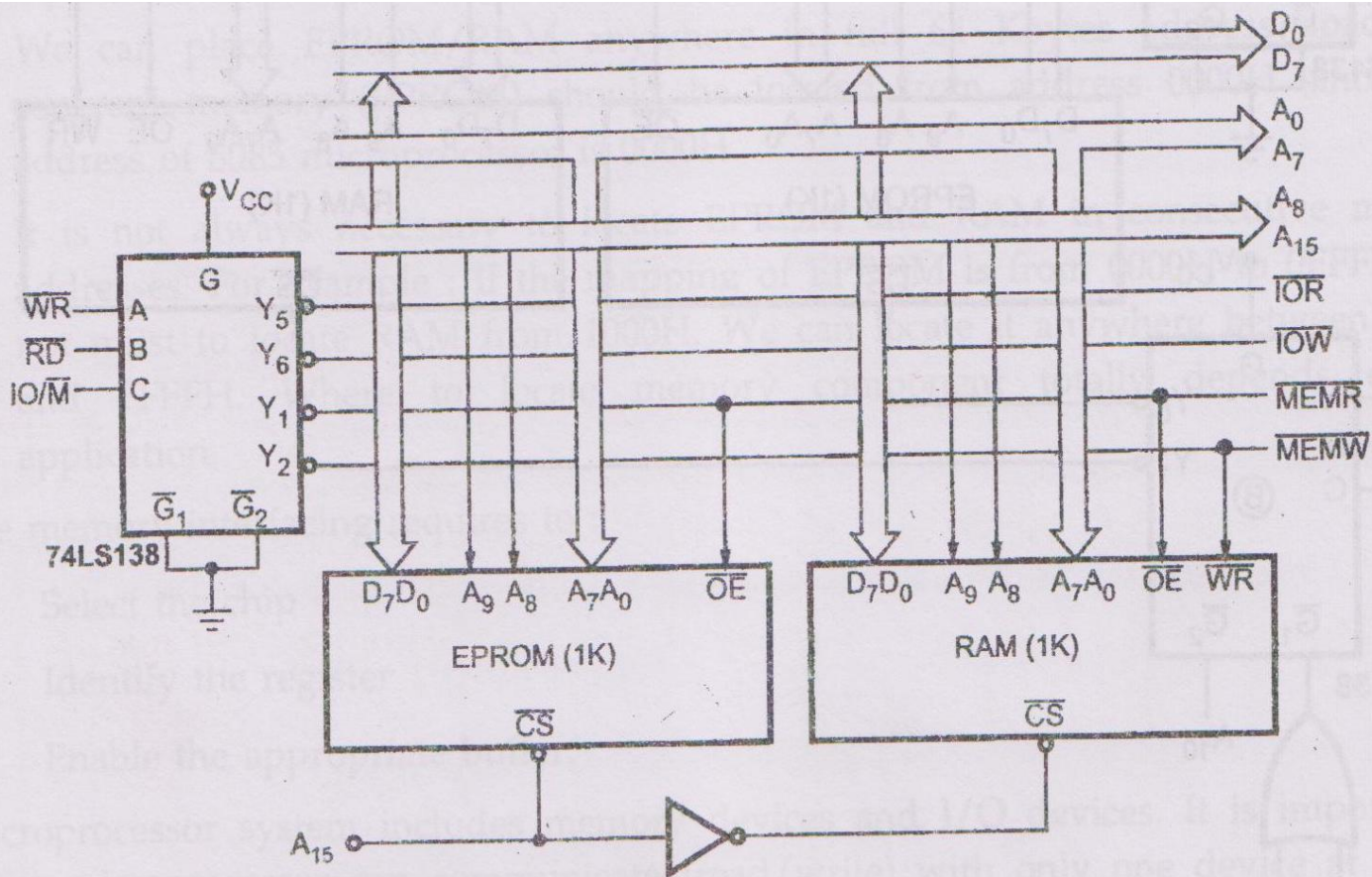


Fig. 8.4 Linear decoding

What about memory map?

Memory ICs	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	Address
Starting address of EPROM	0	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0000H
End address of EPROM	0	X	X	X	X	X	1	1	1	1	1	1	1	1	1	1	03FFH
Starting address of RAM	1	X	X	X	X	X	0	0	0	0	0	0	0	0	0	0	8000H
End address of RAM	1	X	X	X	X	X	1	1	1	1	1	1	1	1	1	1	83FFH

Table 8.3

Interfacing example1:

- Design memory system for 8085 μ p such that it should contain 8 KB of EPROM and 8 KB of RAM.
- Sol: the figure below shows the desired memory system using IC 2764 (8K) EPROM and 6264 (8k) RAM. Memory requires 13 address lines (A0-A12)¹.
- The remaining address lines (A13-A15) are decoded to generate chip select (CS) signals.
- IC 74SL138 is used as decoder, when (A13-A15) address lines are 0 the Y0 output of decoder goes low and select the EPROM. other line to select the particular memory location.
- When these lines are 001, the Y1 output of decoder goes low and selects the RAM.

Memory system

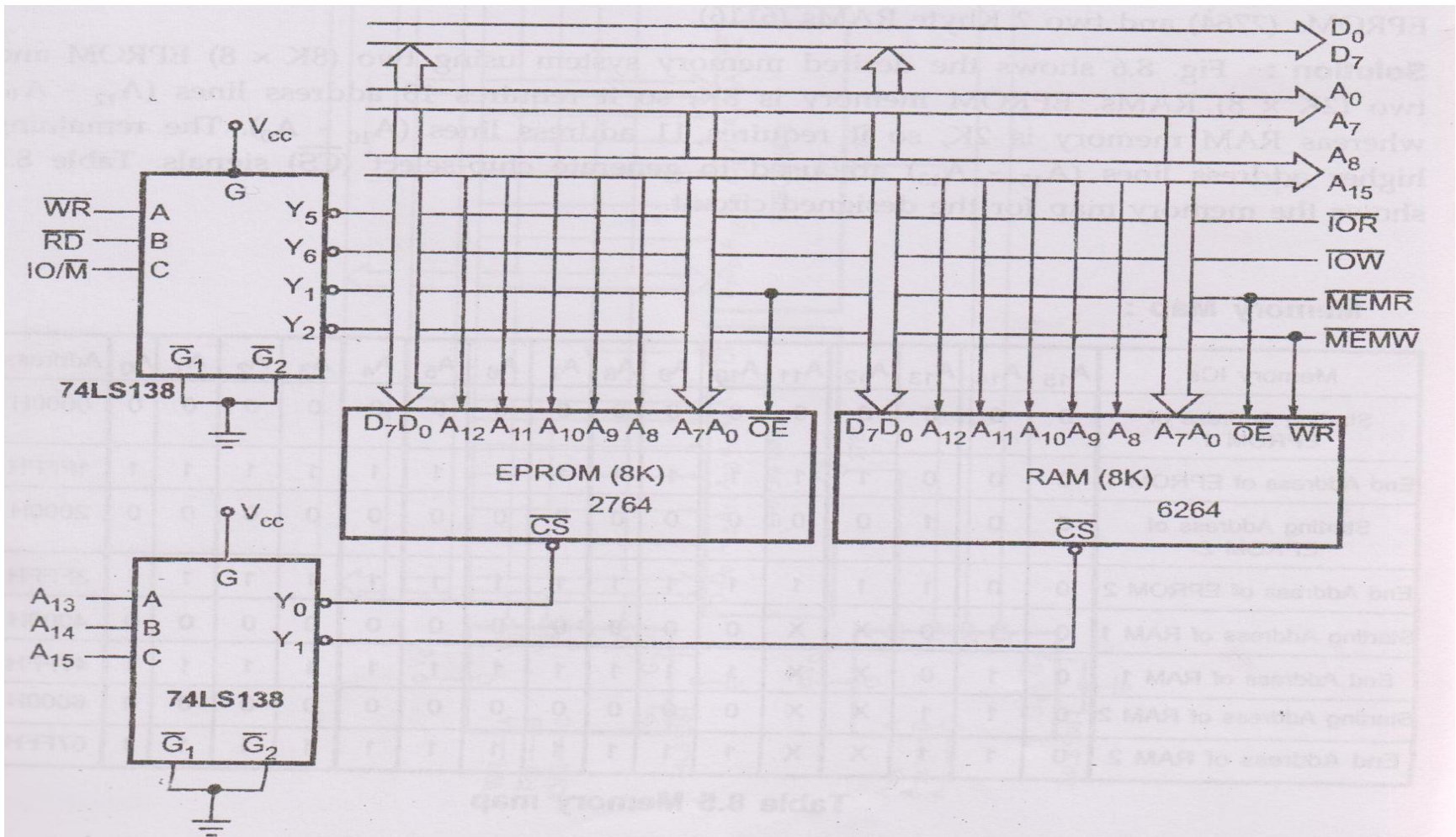


Fig. 8.5 Memory system using IC 2764 (8K) EPROM and 6264 (8K) RAM

Memory map

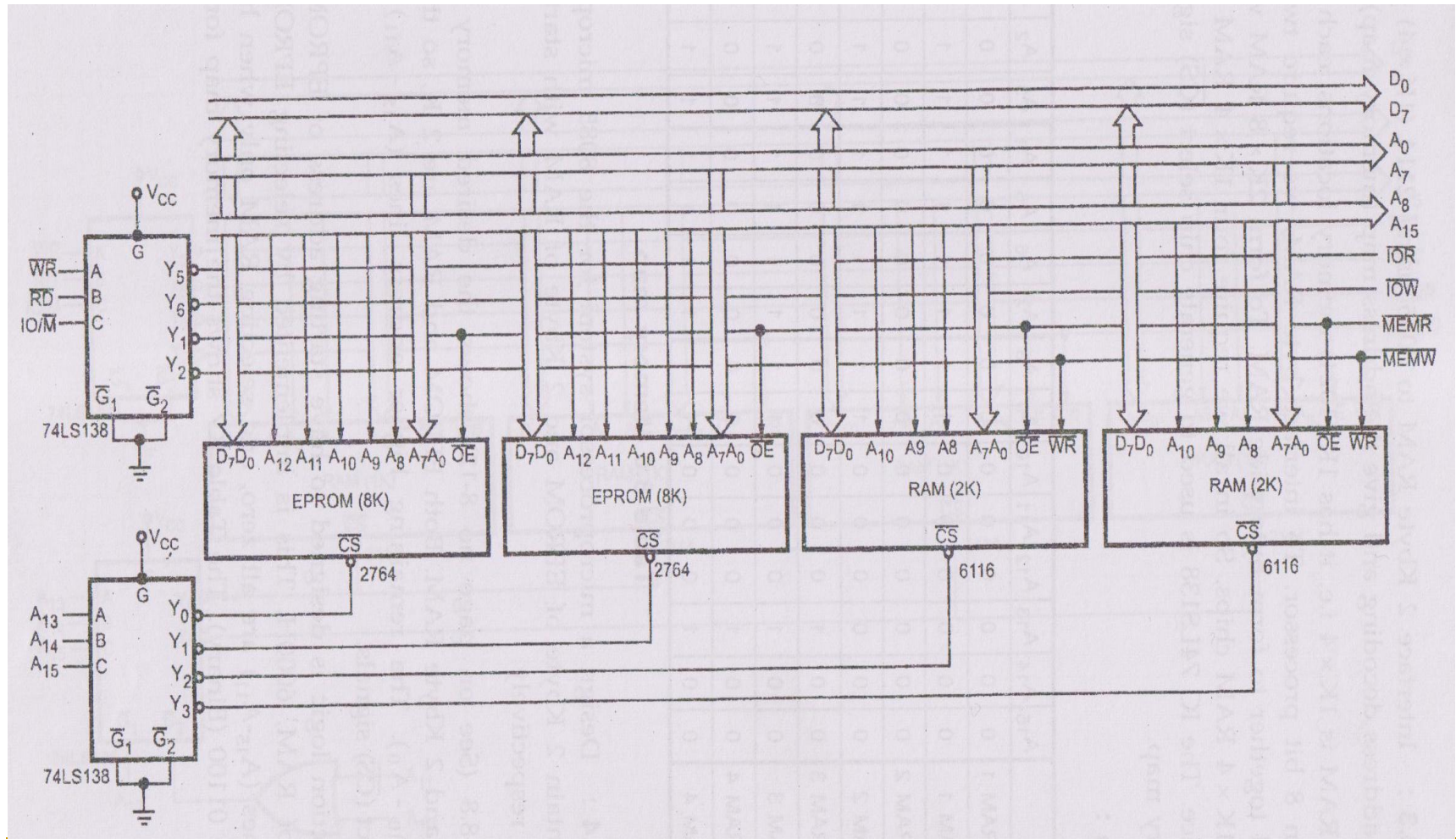
Memory ICs	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	Address
Starting address of EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End address of EPROM	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFH
Starting address of RAM	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
End address of RAM	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH

Table 8.4 Memory map

Example 2

- Design a μp system for 8085 such that it should contain 16 KB of EPROM and 4 KB of RAM using two 8 KB EPROMs (2764) and 2 KB RAMs (6116).
- Sol: the following figure shows the desired memory system using two (8K x 8) EPROM and two (2K x 8) RAMs.
- EPROM memory is 8K, so it requires 13 address lines (A_{12} - A_0) whereas RAM memory is 2K, so it requires 11 address lines (A_{10} - A_0). The remaining address lines (A_{15} - A_{13}) are used to generate chip-select (CS) signals. Also the memory map is shown

Memory system solution



Memory map

memory map .

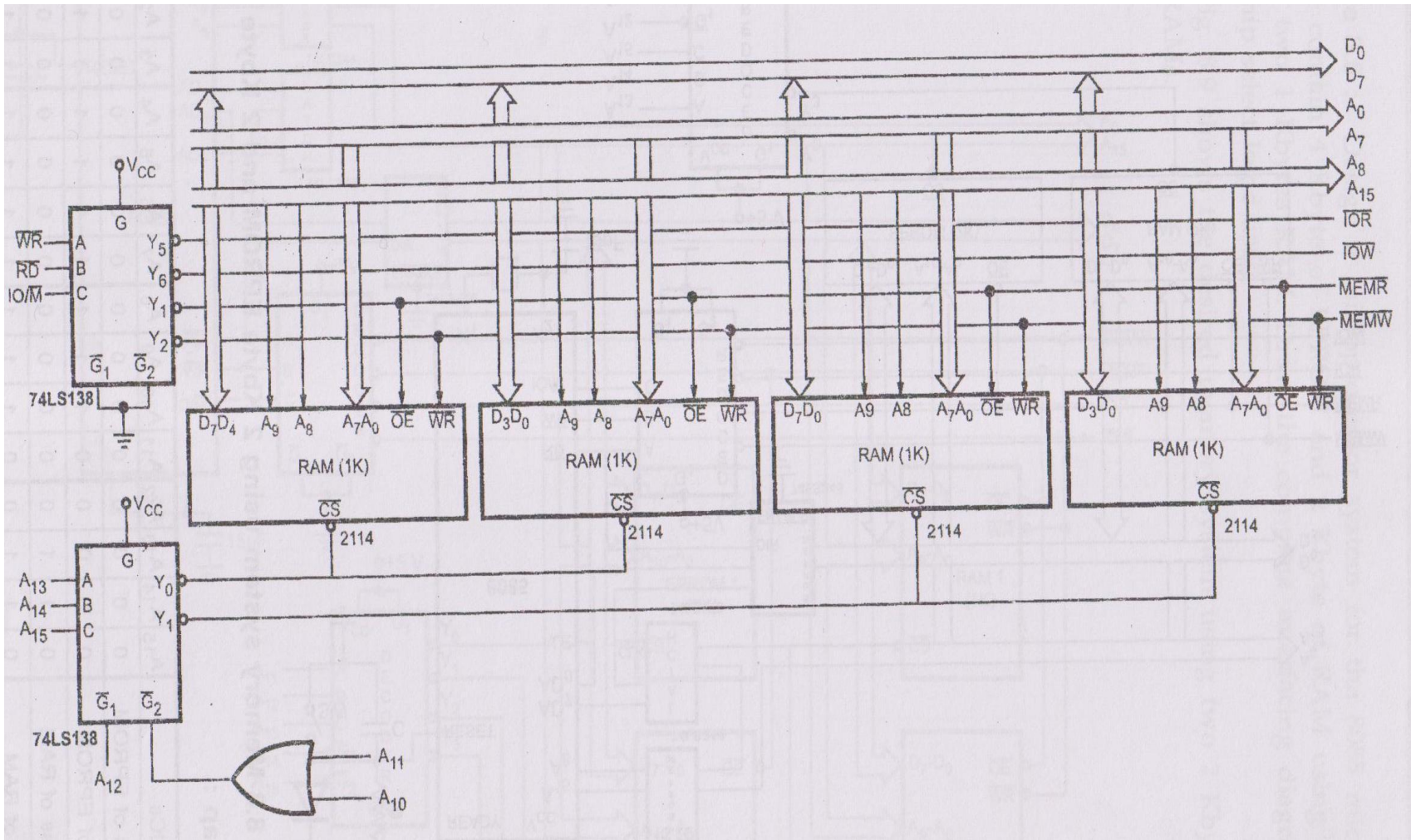
Memory ICs	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	Address
Starting Address of EPROM 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End Address of EPROM 1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFH
Starting Address of EPROM 2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000H
End Address of EPROM 2	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH
Starting Address of RAM 1	0	1	0	X	X	0	0	0	0	0	0	0	0	0	0	0	4000H
End Address of RAM 1	0	1	0	X	X	1	1	1	1	1	1	1	1	1	1	1	47FFH
Starting Address of RAM 2	0	1	1	X	X	0	0	0	0	0	0	0	0	0	0	0	6000H
End Address of RAM 2	0	1	1	X	X	1	1	1	1	1	1	1	1	1	1	1	67FFH

Table 8.5 Memory map

Example 3:

- Interface 2 KB RAM to 8085 using 2114 (1k x 4) chips, 74LS138 decoder and full address decoding and give the address map.
- Sol: 2114 RAM is 1K x 4 i.e. it has 1K (1024) memory locations, each of which is 4 bits. 8085 is an 8 bit processor. To interface byte RAM, we require two nibble wide RAMs, connected together to form byte wide RAM.
- To form 2K x 8 RAM we require two sets of 1Kx4 + 1Kx4 RAM chips.
- so in all we require four 1K x 4 RAM chips. the figures below shows both the interface and map.

Memory System using 2114 (1K x 4) chips



Memory Map

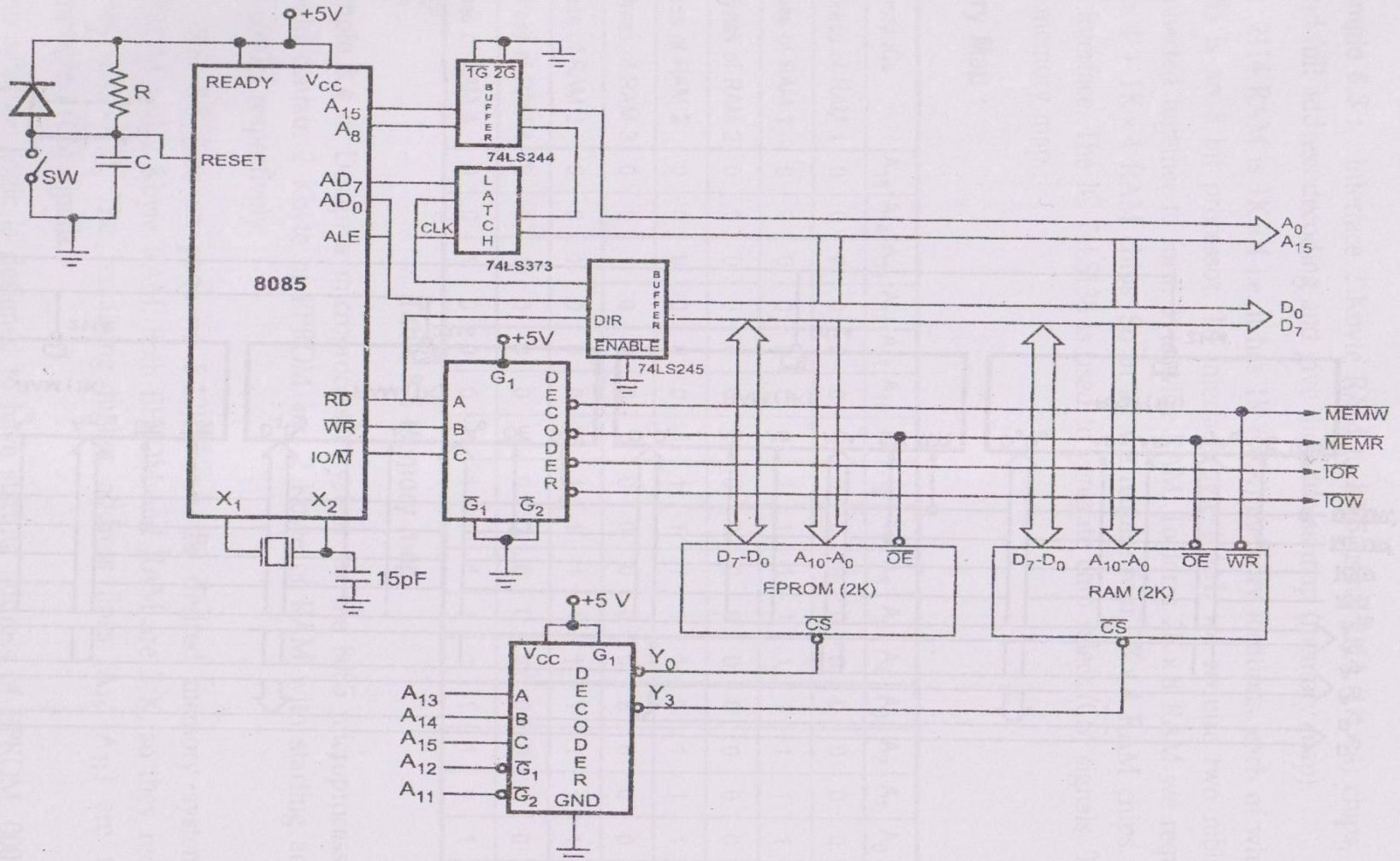
Mem IC	A ₁₅	A ₁₄				A ₉							A ₁	A ₀	Address
SA RAM1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
EA RAM1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	03FFH
SA RAM2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
EA RAM2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	03FFH
SA RAM3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2000H
EA RAM3	0	0	1	0	0	1	1	1	1	1	1	1	1	1	23FFH
SA RAM4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2000H
EA RAM4	0	0	1	0	0	1	1	1	1	1	1	1	1	1	23FFH

Example 4

- Design a μp system for 8085 μp such that it should contain 2 KB of EPROM and 2 KB of RAM with starting address 0000H & 6000H.
- Sol: the figure below shows the desired memory system using 2 KB EPROM and 2 KB RAM. Both are 2 KB so they need 11 lines (A_0 - A_{10}) and the remaining higher addresses (A_{15} - A_{11}) is used to generate the chip select (CS).
- Since EPROM starts from 0000H and RAM from 6000H; EPROM selected when all higher address = 0000 and RAM selected when (A_{15} - A_{11}) = 01100 B

Memory system using 2K EPROM & RAM

Fig. 8.8 Memory system using 2 Kbyte EPROM and 2 Kbyte RAM



Memory map

Memory ICs	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	Address
Starting address of EPROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H
End address of EPROM	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FFH
Starting address of RAM	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000H
End address of RAM	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	67FFH

Table 8.7 Memory map

Decoding type comparison

Full Address Decoding	Partial Address Decoding
All higher lines are decoded	Few higher address lines are decoded
More H/W is required	H/W design is less or eliminated
Higher cost	Less cost
No multiple addresses	It has multiple address (shadow address)
Used in large system	Used in small system

Basic concepts in memory interface

- μp 8086 can access 1 MB memory (since address lines are 20)
- Same as 8085 when EPROM is used for program memory and RAM is used for data the space is shared between those two.
- Both RAM and EPROM capacities depends on application
- Multiple RAM and EPROM can be used.
- EPROM/RAM can be placed any where, but EPROM should be located at last memory page¹
- It is not always to locate EPROM and RAM in consecutive locations. (it is advised to)
- Even bank is selected when $A_0=0$ and odd if $BHE=0$

Memory interface requirements

- Select the chip
- Identify the register
- Enable the appropriate buffer.
- μ p system includes memory and I/O devices
- μ p can communicate with only one device¹
- So decoding needed to communicate with memory and I/O
- So each device can be accessed independently.
- Three type of decoding:
 - Absolute
 - Linear
 - block

Absolute decoding

- The memory chip is selected only for the specified logic level on address lines; no other logic level can select the chip as shown below.
- Two 8K EPROMs (2764) are used to provide even and odd memory banks (16 bits).
- Control signals BHE and A_0 are used to enable O/P of odd and even memory banks.
- 13 address lines are required to address 8K locations. (A_1 - A_{13})
- All remaining address lines (A_{14} - A_{19}) to generate chip select
- This technique is used in large memory system.

Absolute addressing

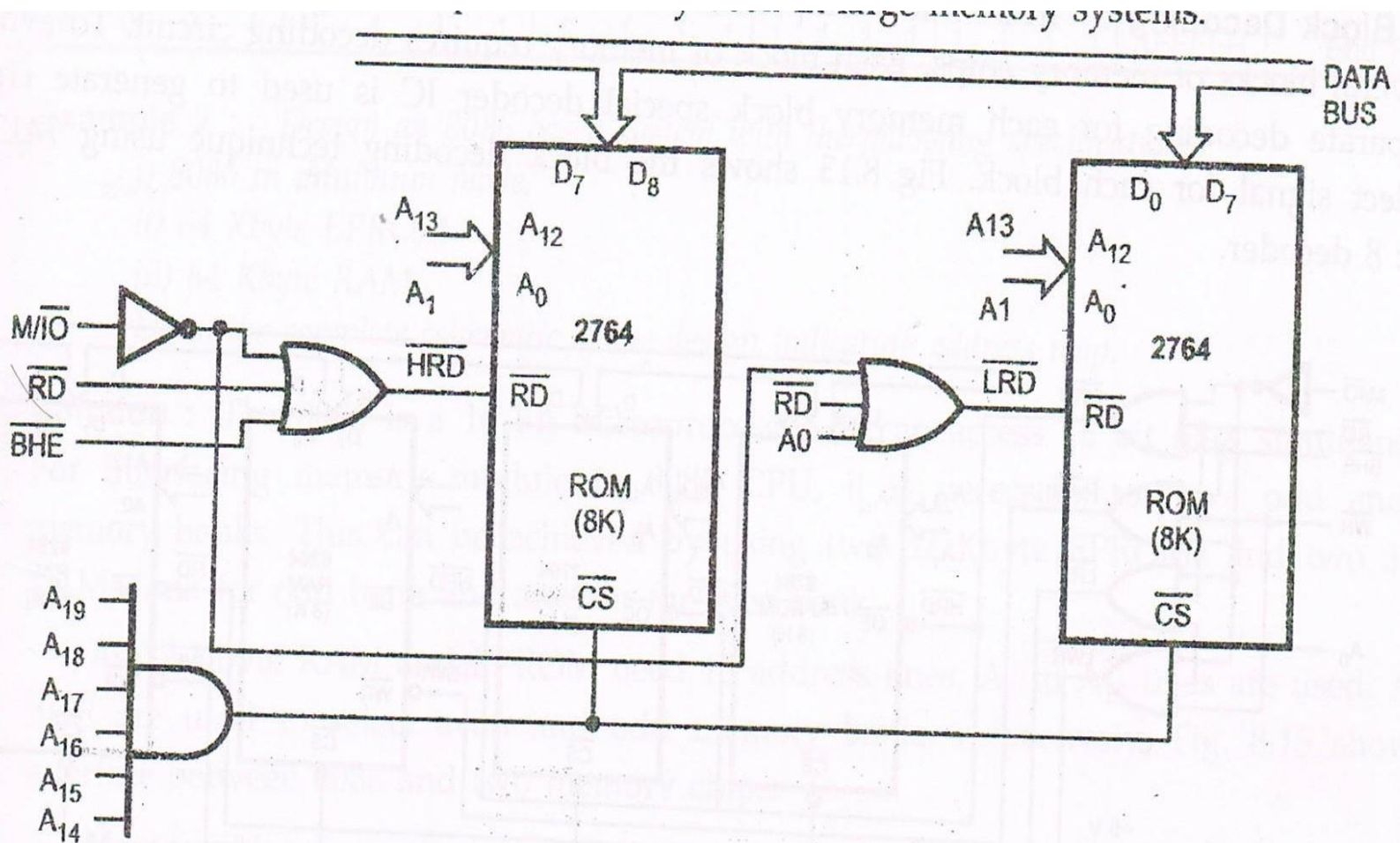


Fig. 8.11 Absolute decoding

Linear Decoding (Partial Decoding)

- In small system, H/W can be eliminated by using the required addressing lines¹.
- The figure below shows the addressing of 16K RAM (6264) with linear decoding
- BHE & A0 are used to enable odd & even banks.
- A19 is used to select the RAM².
- A14 to A18 are not affect the chip selection
- This method reduces cost
- But it gives multiple (shadow) addresses

Linear Decoding

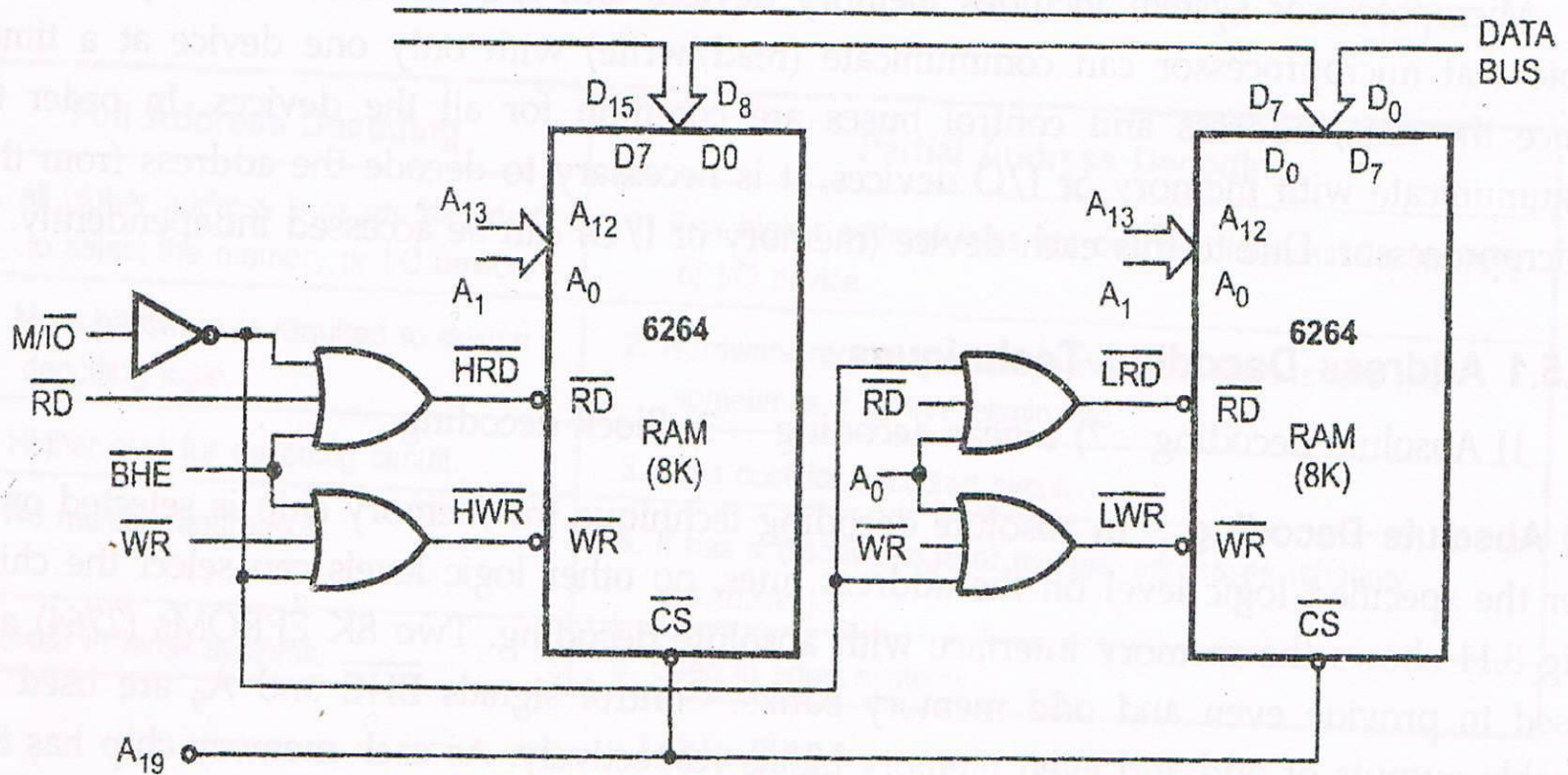


Fig. 8.12 Linear decoding

2) Block Decoding

Block Decoding

- In a μp system the memory array is often consists of several blocks of memory chips
- Each block of memory requires decoding cct.
- To avoid separate decoding for each memory block special decoder IC is used to generate chip select signal for each block.
- The figure below shows the block decoding using IC 74138 \Rightarrow 3:8 decoder

Block Decoding

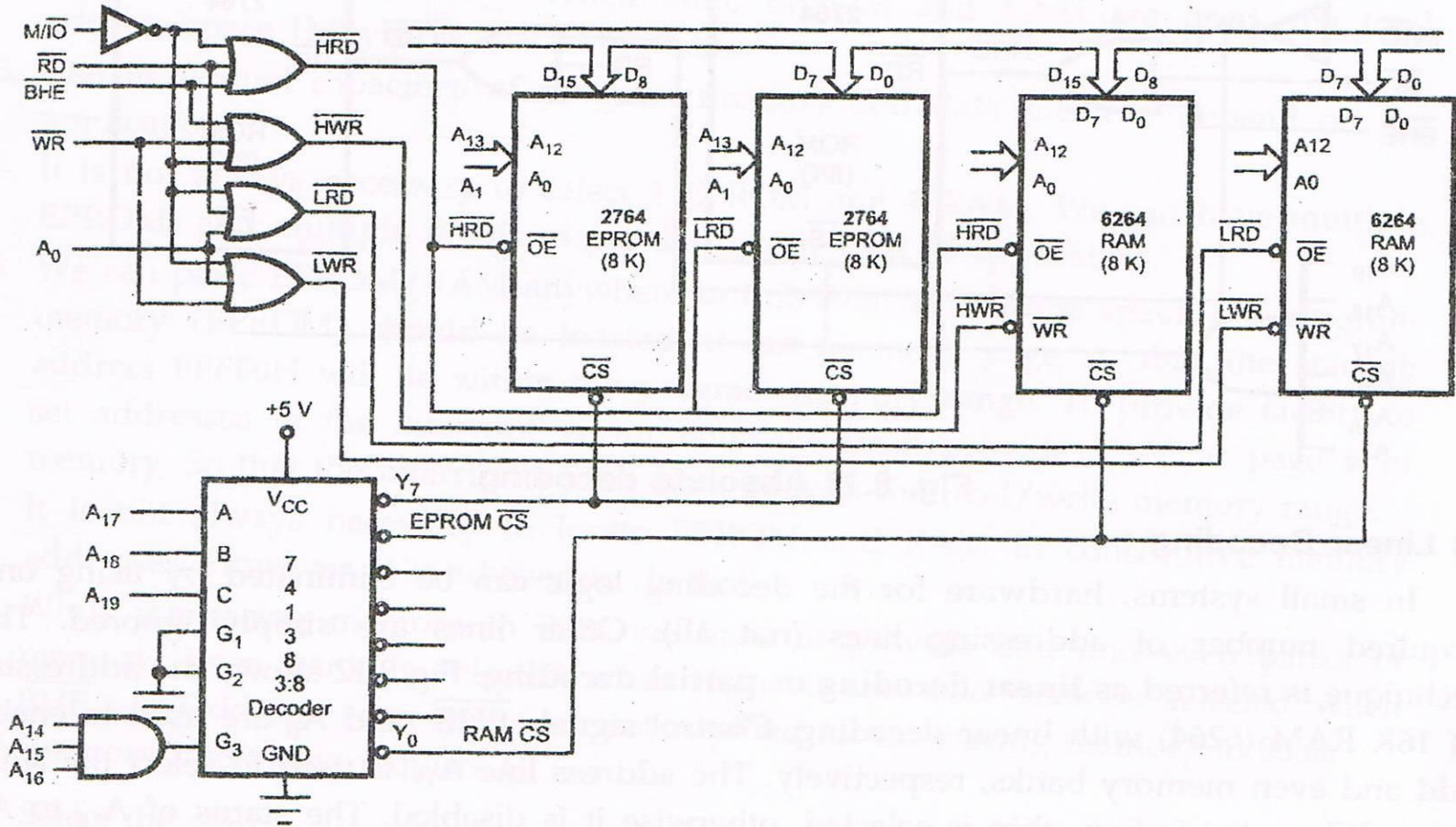


Fig. 8.13 Block decoding

Example 1:

- Design an 8086 based system with the following specifications:
 - 8086 in minimum mode
 - 64 KB EPROM
 - 64 KB RAM
- Draw the complete schematic of the design indicating address map.
- Sol: 8086 is 16 bit μ p so it is necessary to have odd and even memory banks.
- Two 32 KB EPROMs and two 32 KB RAMs¹
- For 32 KB RAM & EPROM need 15 address lines (A1-A15)
- A0 & BHE are used to select even and odd banks

Memory Map

Memory Map :

BHE	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	Address	Memory
1	1	1	1	1	1	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	FFFFFFH	EPROM1
0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F0001H	Odd
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH	EPROM2
1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30000H	Even
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	3FFFEH	RAM1
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	30001H	Odd
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFFH	RAM2

