Lecture 13 Chapter 7: Memory

Overview

- Memory definitions
- Random Access Memory (RAM)
- Static RAM (SRAM) integrated circuits

Memory Definitions

Memory – A storage device (a collection of storage cells) for read and write data

Two types of memories:

- Random Access Memory (**RAM**)
  -- Read and write at any time and any address
  -- *Volatile* – loses stored information when power turned off

- Read-only Memory (**ROM**)
  -- perform only read operation
  -- *Non-volatile* – retains information when power turned off

Memory Definitions

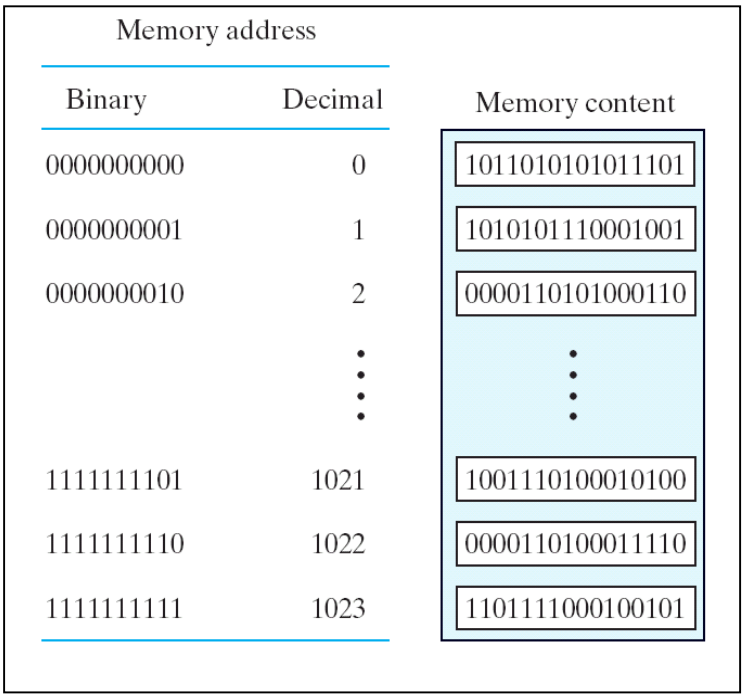
- Typical data elements are:
  - bit — a single binary digit
  - byte — a collection of eight bits accessed together
  - word
    — an entity of bits that move in and out of storage as a unit
    — typically multiple of 8 bits (bytes)
    Example: 8-bit word, 16-bit word, 32-bit word

Capacity of a memory —usually stated as the total number of **bytes** that the unit can store

MEMORY ORGANIZATION EXAMPLE

Memory address:

- Value of the index for each word
- 16 bit per word
  $1024=2^{10}$ words



Memory Block Diagram

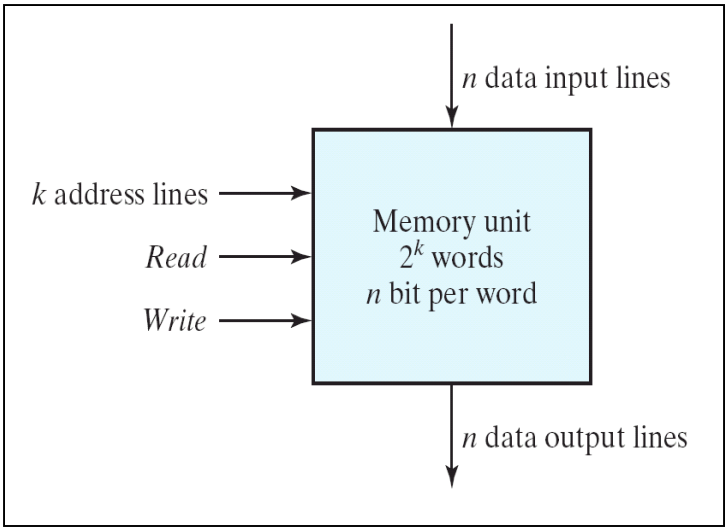
- Data lines: input or output

  n-bit word: 8-bit, 16-bit, 32-bit

- Address bus:
  k address lines:
  --address space is 2^k words
  --n bits per word
  $2^k \times n$ memory

  example: 64K x 10 memory

- Control:
  Read and Write
  Enable



RAM

Data lines

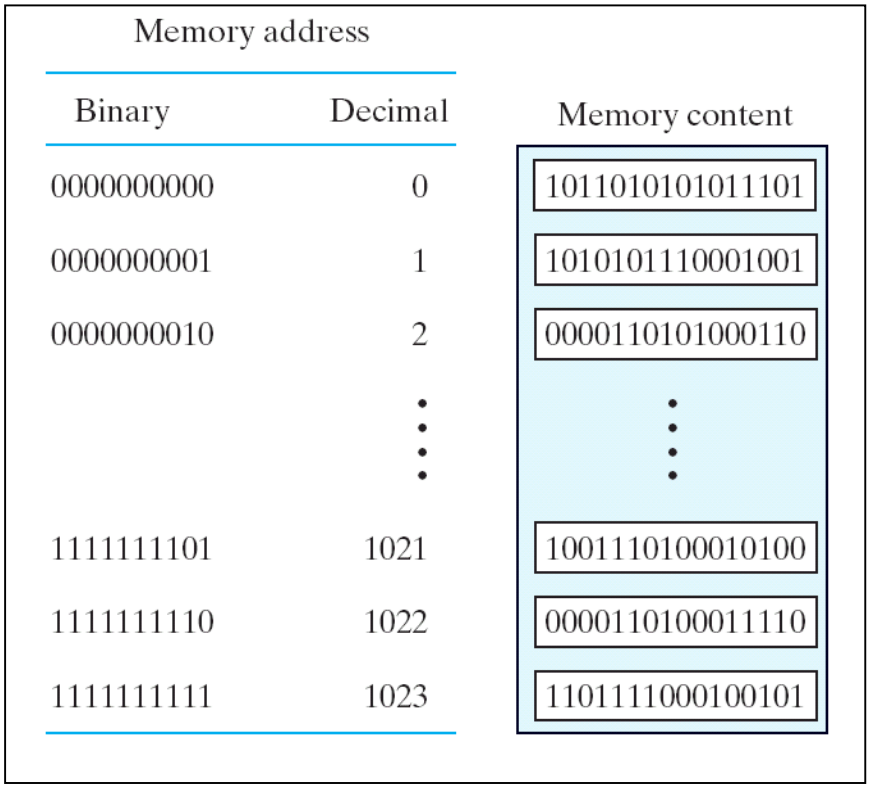
- 16- bit

Address lines

- 10- bit

So, it is a

- 1024×16 memory



Basic Memory Operations

Write

- Apply the binary address of the desired word to the address lines
- Apply the data bits that to be stored in memory to the data input lines
- Activate write

Read

- Apply the binary address of the desired word to the address lines
- Activate read

Table 7.1
Control Inputs to Memory Chip

Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

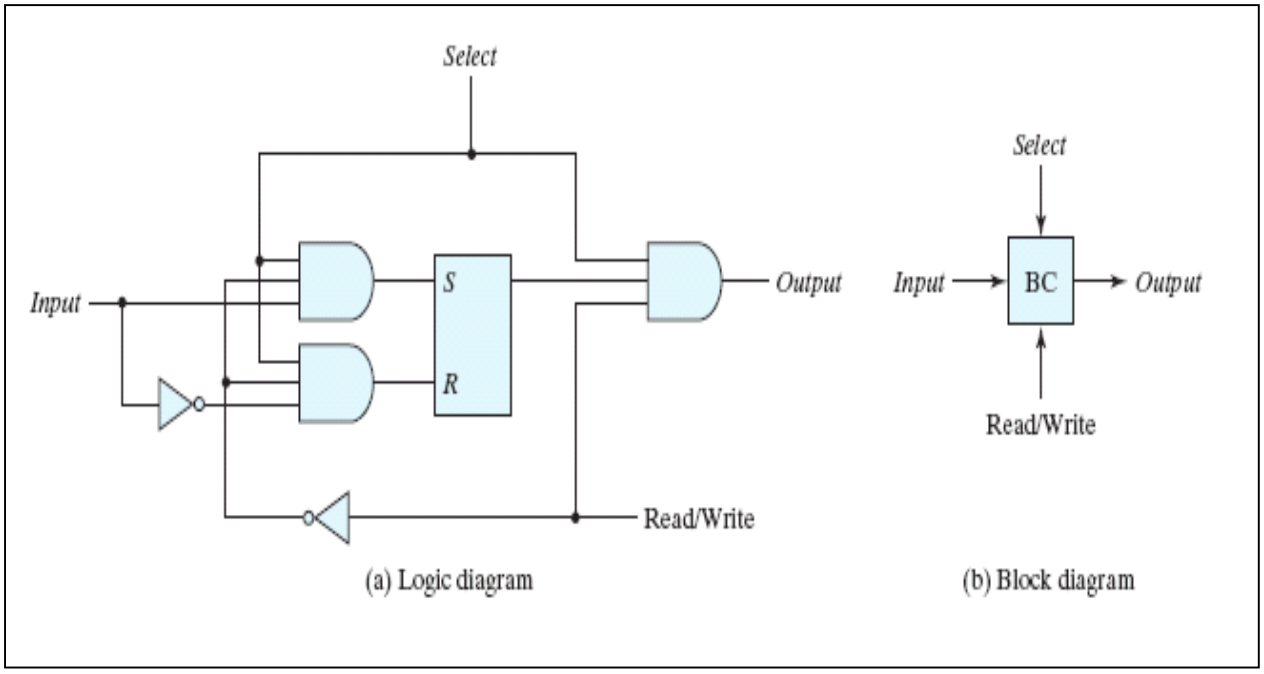
Types of RAM

Types of random access memory

- SRAM: *Static* – bits stored in latches and flipflops
- Information is valid as long as power is on
- Easier to use and faster to read and write

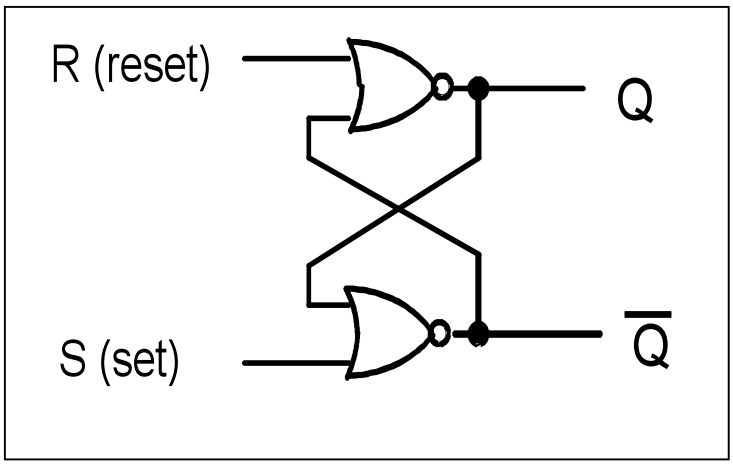
Memory Cell

- If select=0, ???
- If select=1 and r/w=1, ???
- If select=1 and r/w=0, ???



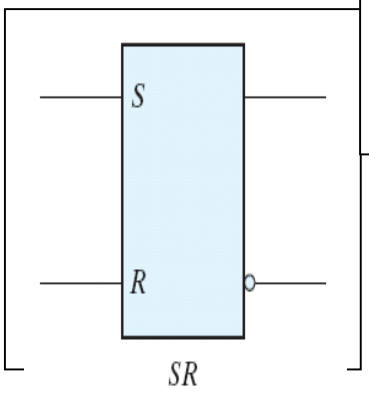
Basic (NOR) S – R Latch

- Cross-coupling two NOR gates gives the S – R Latch:
- Which has the time sequence



behavior:

S = 1, R = 1 is <u>forbidden</u> as input pattern

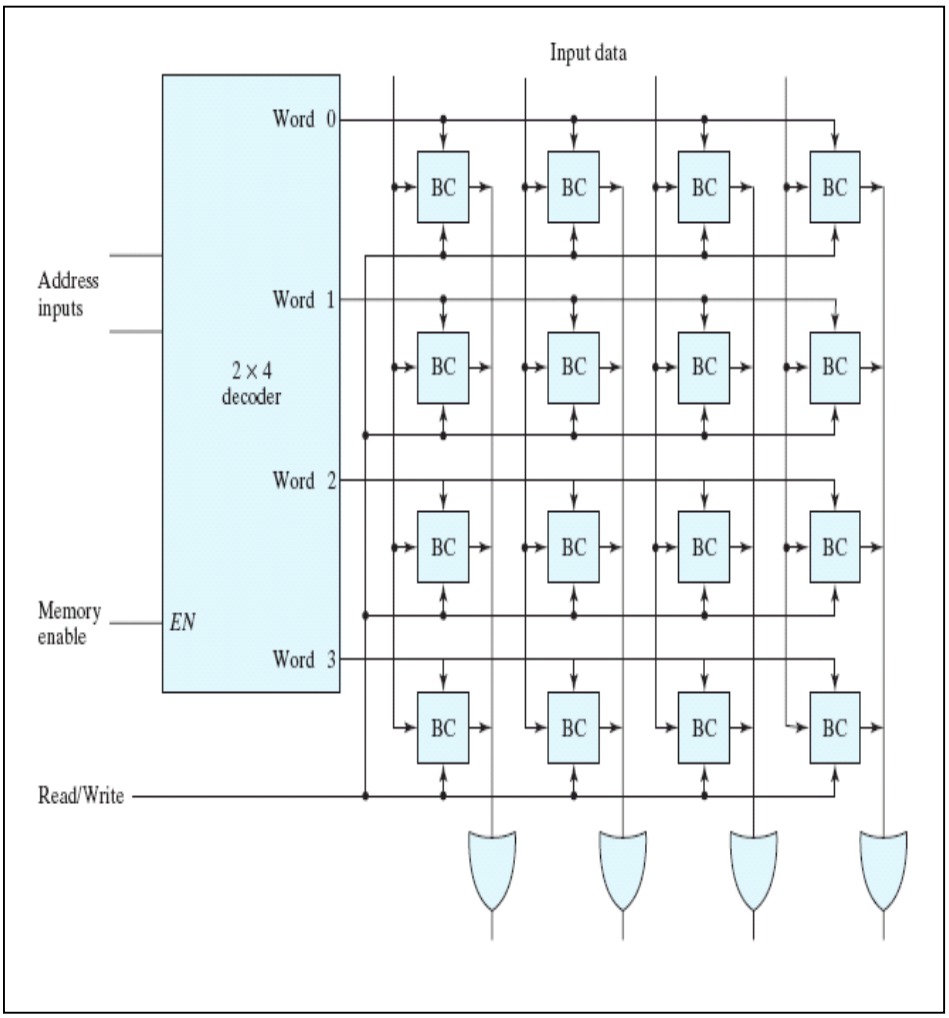


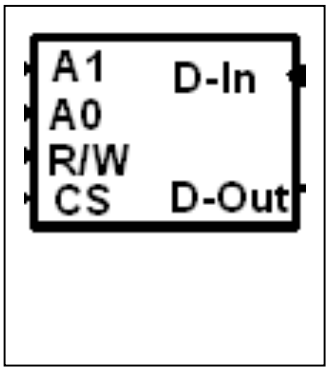
Time

R	S	Q	$\overline{Q}$	Comment
0	0	?	?	Stored state unknown
0	1	1	0	"Set" Q to 1
0	0	1	0	Now Q "remembers" 1
1	0	0	1	"Reset" Q to 0
0	0	0	1	Now Q "remembers" 0
1	1	0	0	forbidden

BUILDING A 4×4 RAM

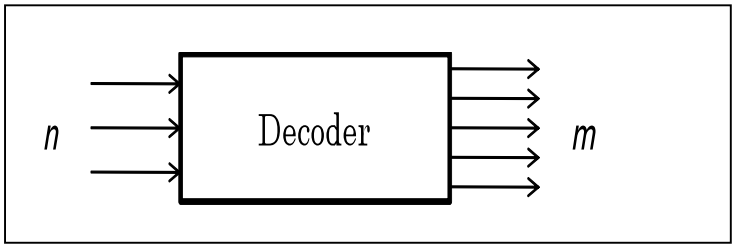
Decoder : decodes the n address lines to 2^n word select lines





Decoder

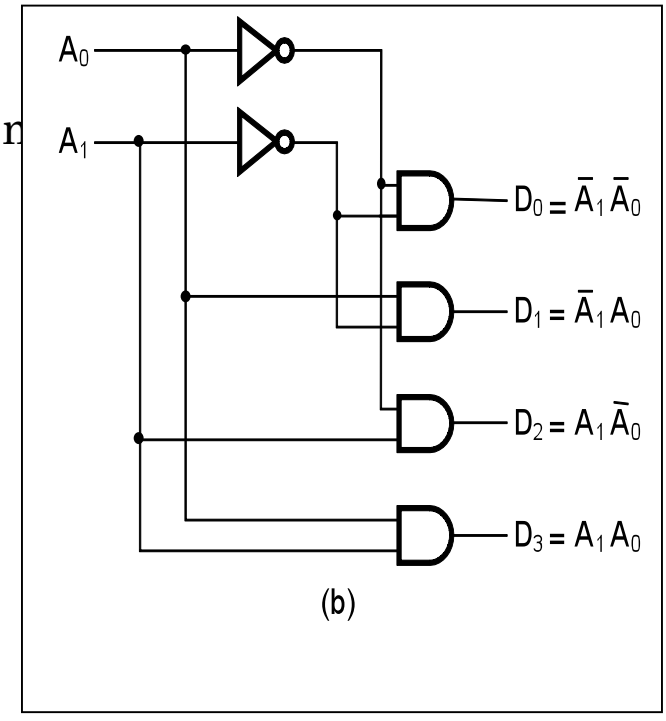
- Decoder – converts n-bit input lines to a maximum of 2^n unique output lines



- n-to-m line decoders, where $n <= m <= 2^n$
- Function: generate 2^n (or fewer) minterms for the n input variables
- 2-to-4-Line Decoder
- Each line D_i is a minterm n

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

(a)

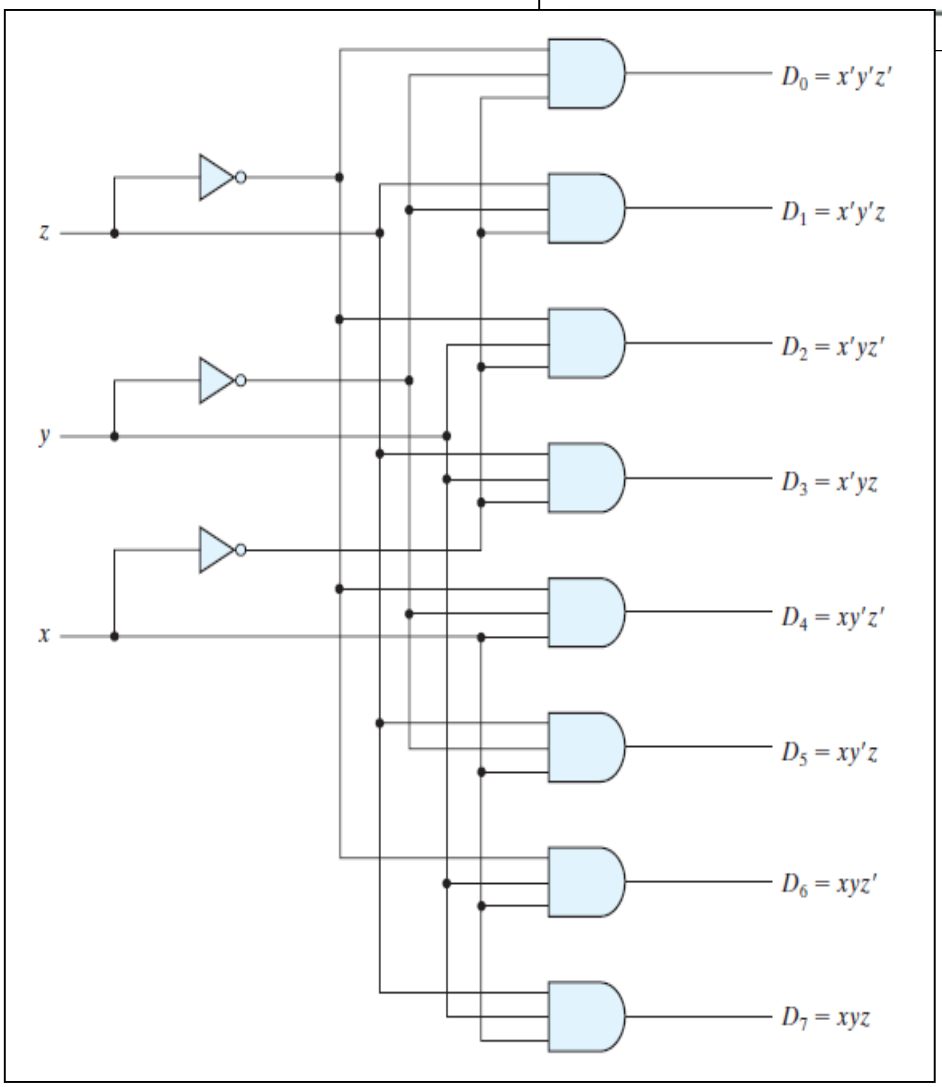


(b)

Decoder Examples

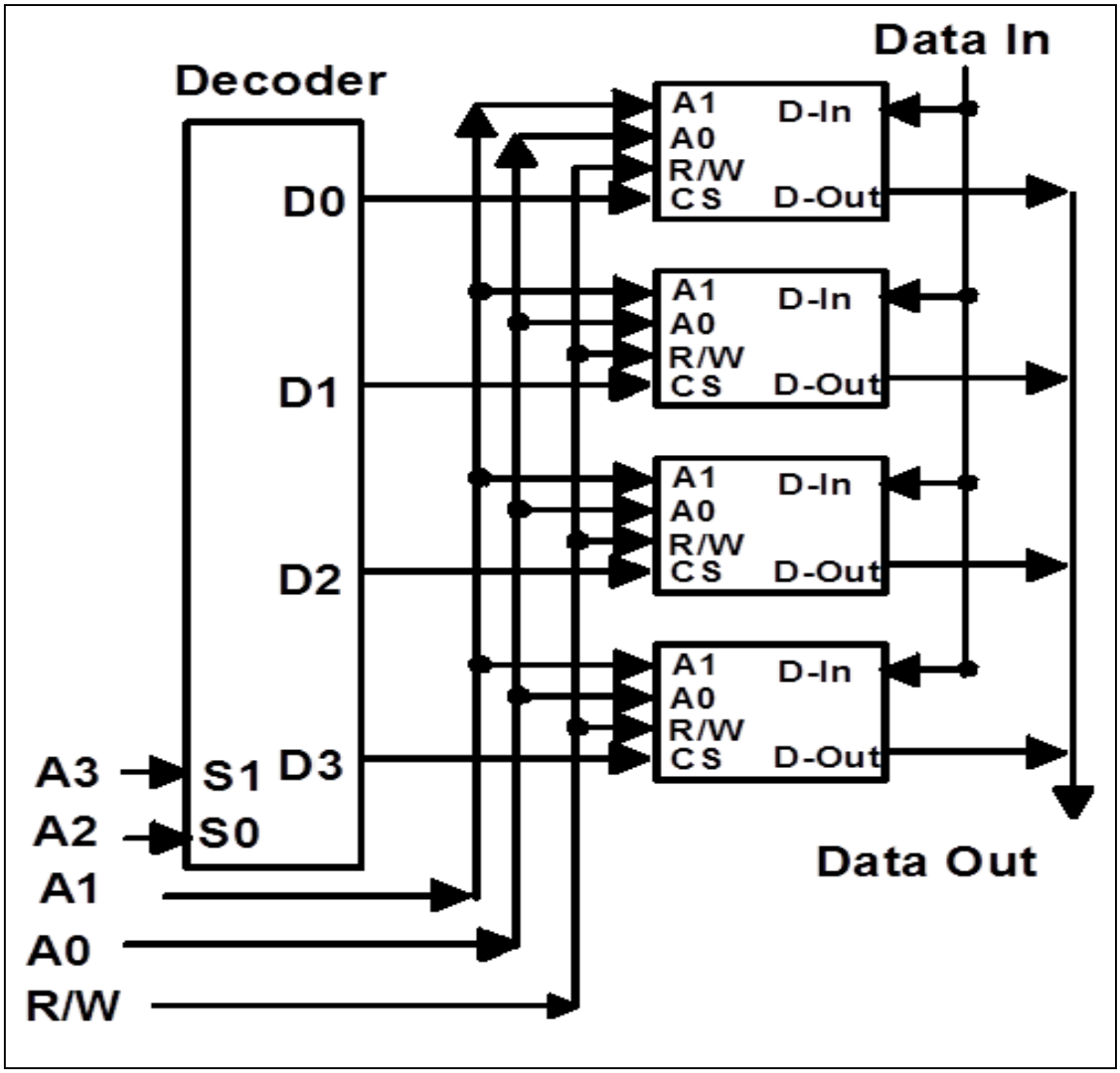
3-to-8-Line Decoder

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Making Larger Memories

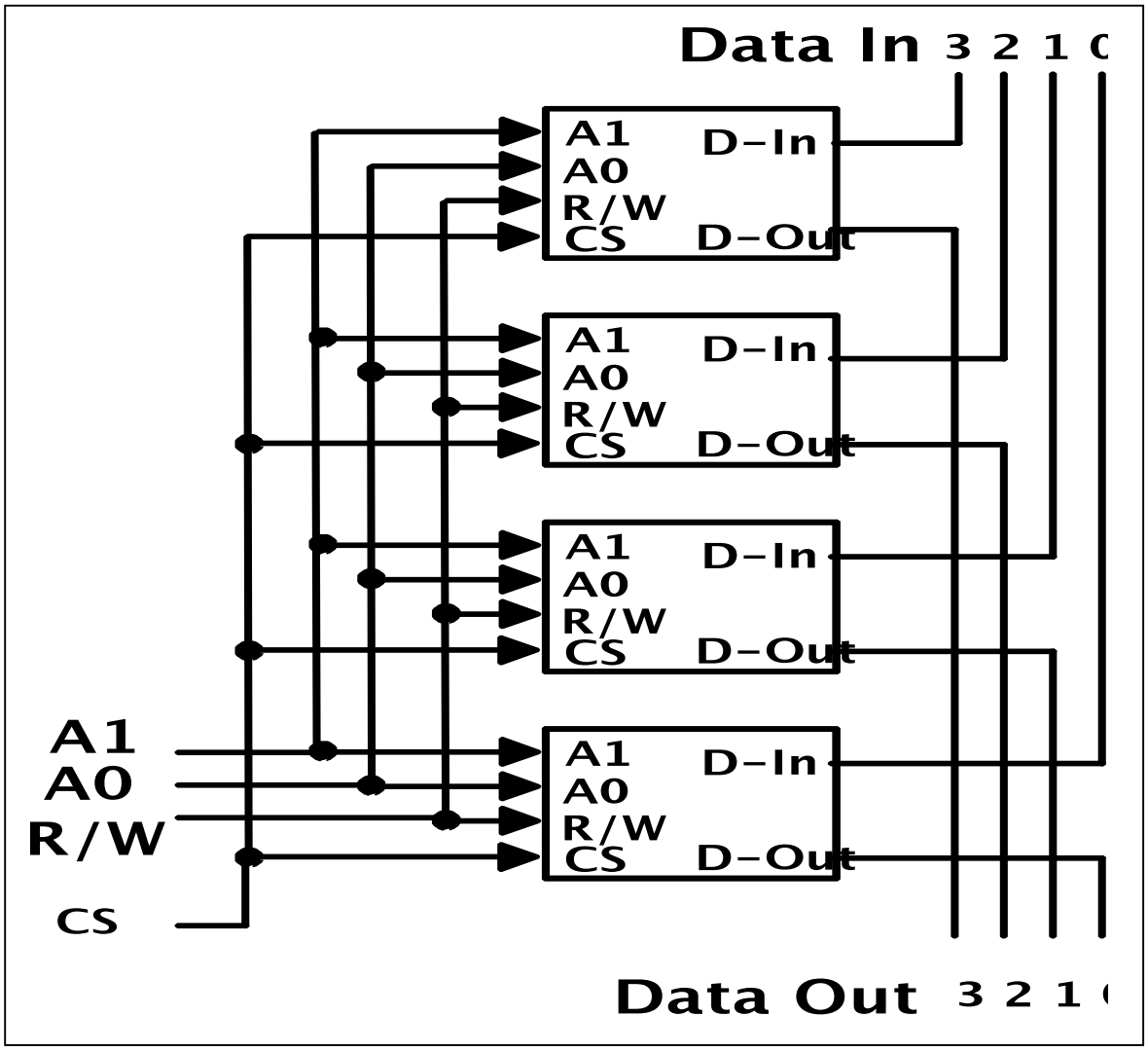
- To make larger memories from smaller ones, we tie all address, data, and R/W lines in parallel, and using the decoded higher order address bits to control CS.



- Using the 4x1 memory chips to construct a 16x1 memory.

Making Wider Memories

- To construct wider memories from narrow ones, we tie the address and control lines in parallel and keep the data lines separate.



- Using the 4x1 memory chips to build a 4x4 memory.

Making larger memories

- Make a 256kx8 memory from 64kx8 memory chips.
    - How many chips?
    - What decoder?
    - How to arrange address lines, data lines, and read/write control?

Making Wider Memories

- Make a 64Kx16 memory from 64Kx8 memory chips.
    - How many chips?
    - What decoder?
    - How to arrange address lines, data lines, and read/write control?

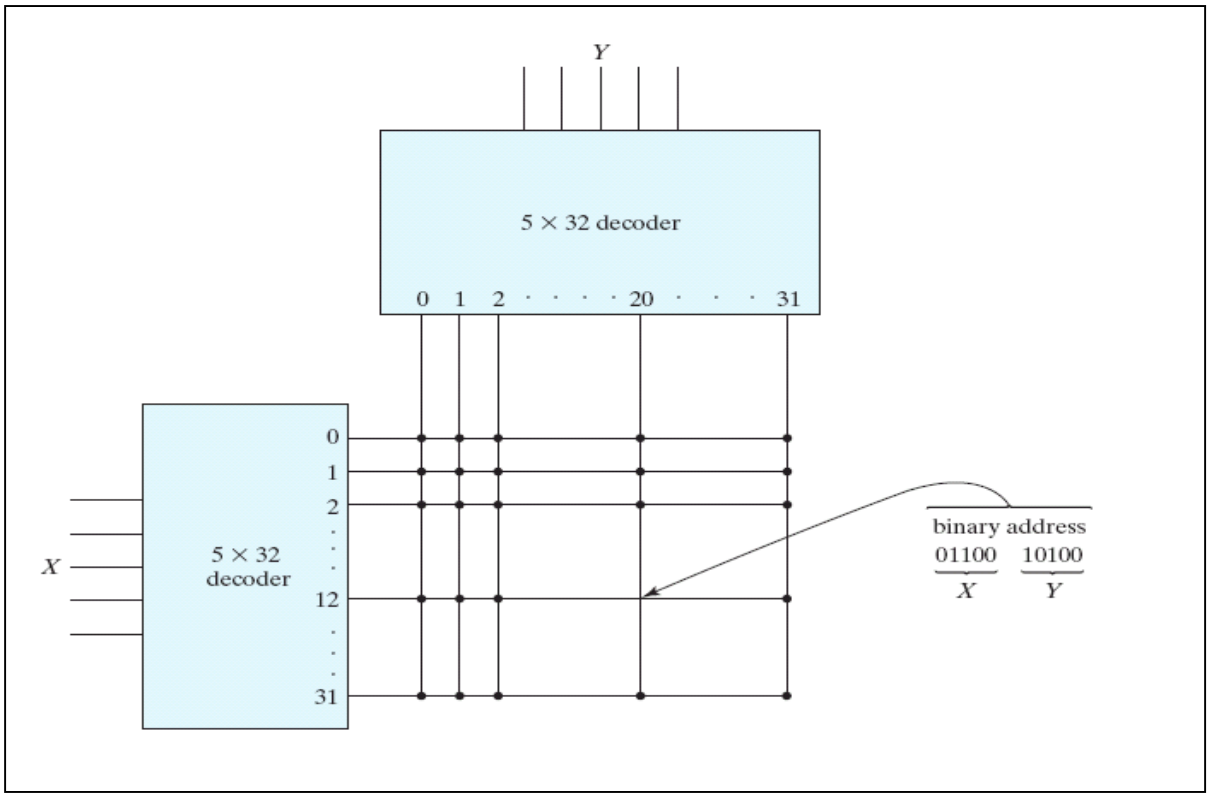
Making Wider and Larger Memory

- Make a 256Kx16 memory from 64Kx8 memory chips.
    - How many chips?
    - What decoder?
    - How to arrange address lines, data lines, and read/write control?

Coincident Decoding

- Problem of a single decoder
  - For 1Kx1 memory, a huge 10-2^{10} decoder is needed

- Solution: coincident decoding
  - Two 5-2^5 decoders are used.
  - One decoder decodes higher bits in address lines.
  - The other decodes lower bits in address lines.
  - Memory units are arranged in an array.
  - A memory unit is selected if both decoders select it.



Coincident Decoding

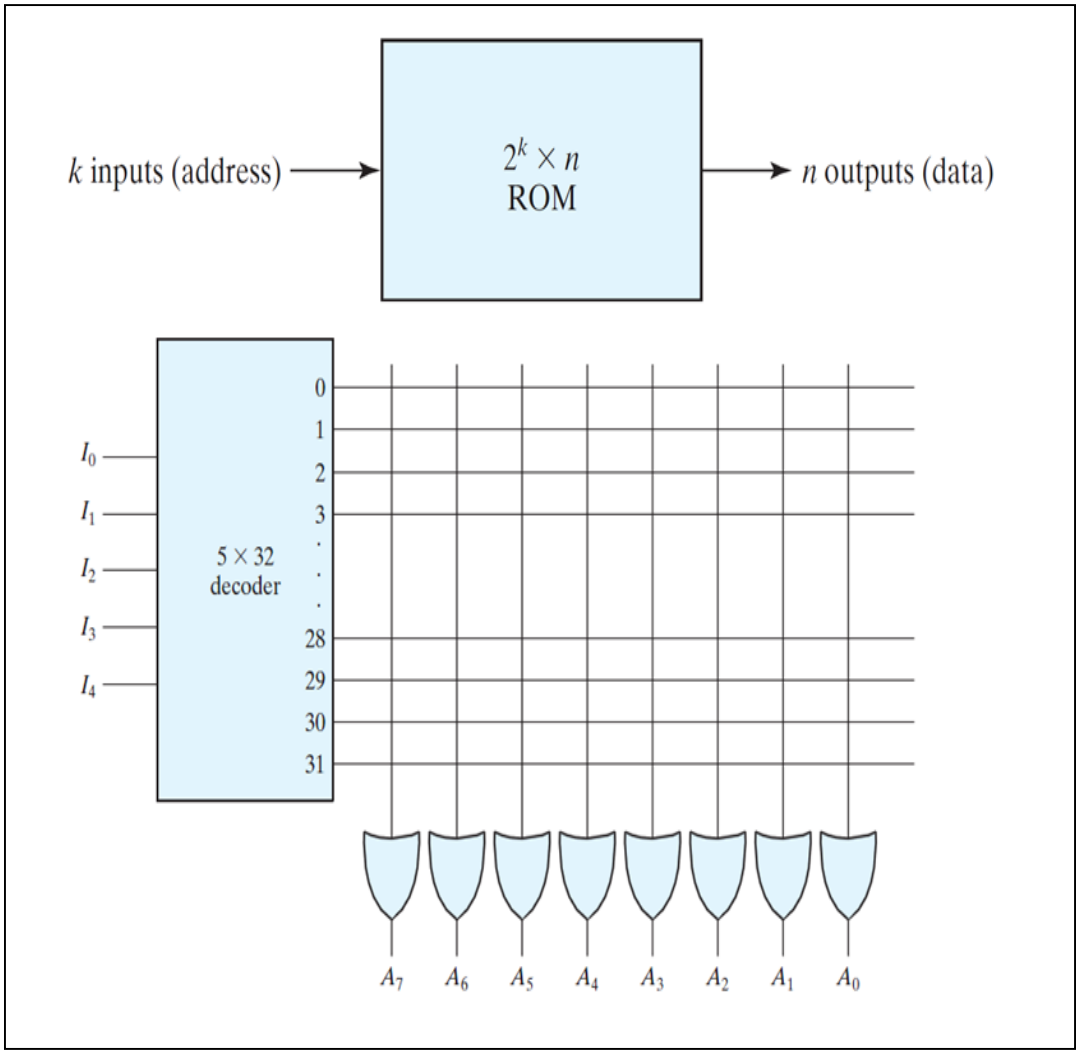
- Given the 1Kx4 memory with coincident decoding.
    - X takes the higher 5 bits of an address
    - Y takes the lower 5 bits of an address

- Write 1011 to address 0110000010
    - X=?
    - Y=?
    -

Read from address 0000110100

ROM

Similar to RAM, but no data input lines.

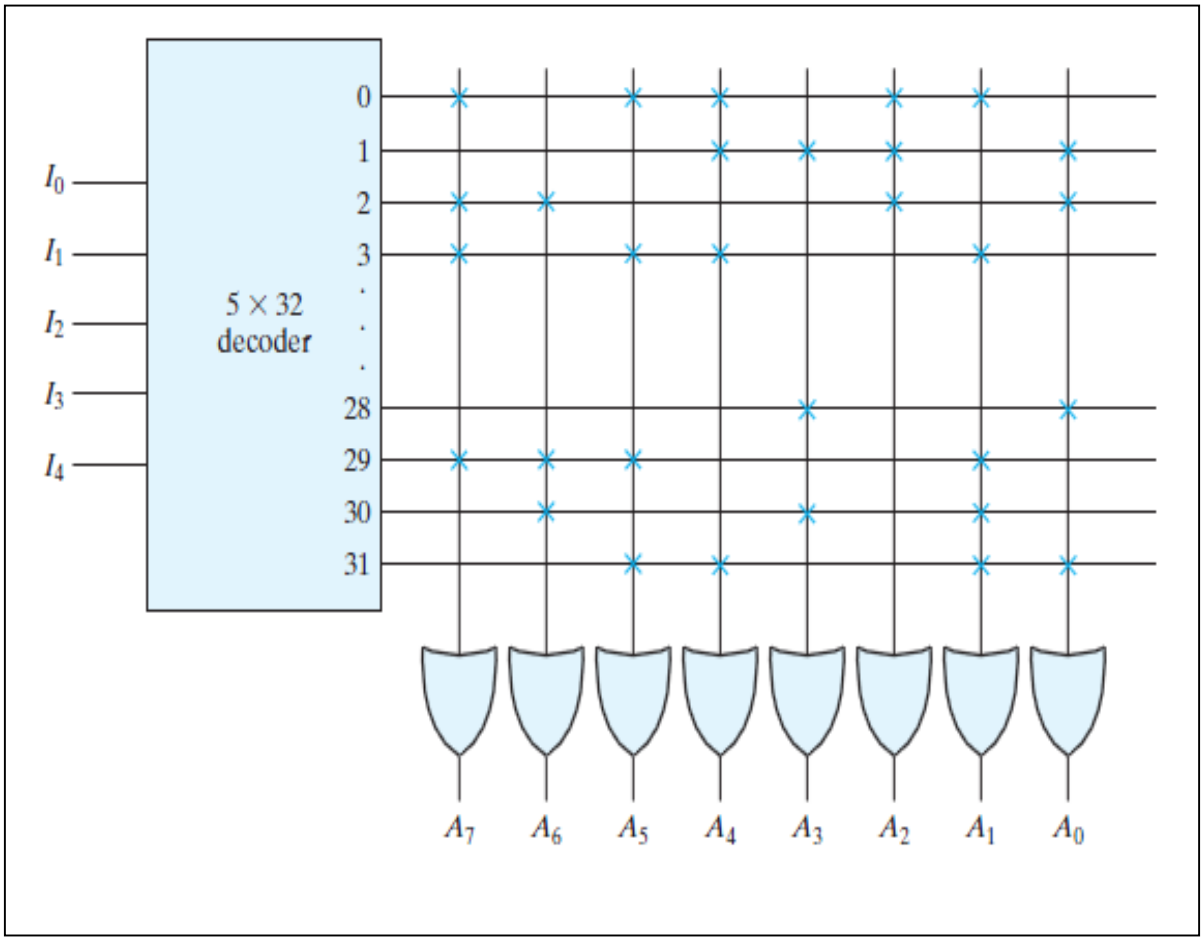


- Stored information in a ROM is represented by a truth table.

Table 7.3
ROM Truth Table (Partial)

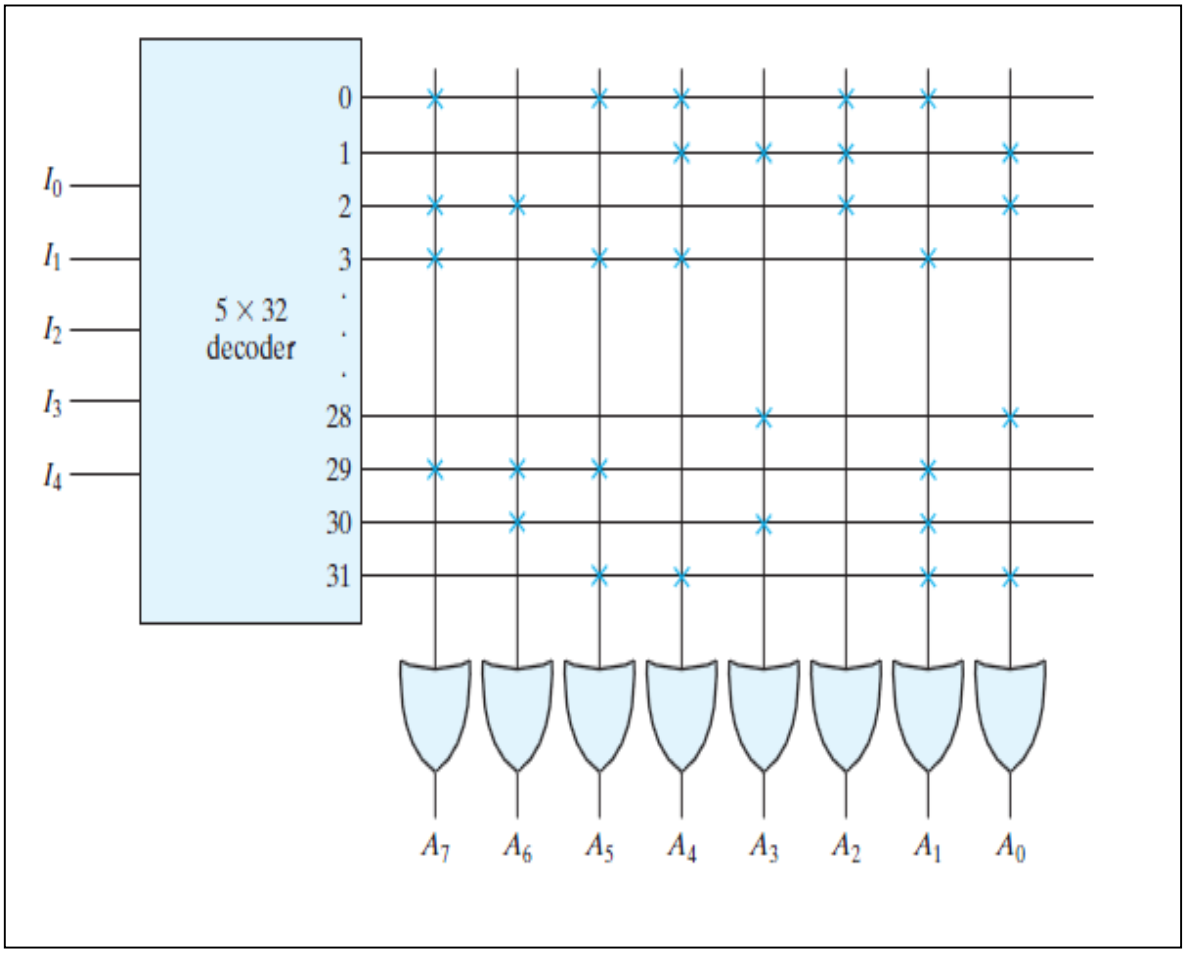
Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮							⋮			
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

- 1 (x) means a connection; 0 means no connection.
- Each gate is a OR gate with inputs from all addresses.



ROM

- What data is at address
- 0 : 10110110
- 1 : 00011101
- 2
- 3
- 28
- 29
- 30
- 31

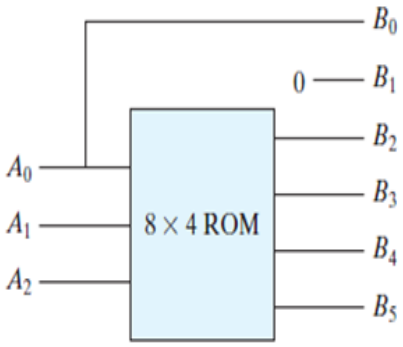


Applications of ROM

- Design a circuit to take a 3-bit number and output its square.

Truth Table for Circuit of Example 7.1

Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49



(a) Block diagram

Types of ROM

- MROM : Mask programming ROM
    - Programmed when being manufactured
- PROM : Programmable ROM
    - Fuses intact when being manufactured
    - Allow programming once, irreversible
- EPROM : Erasable PROM
    - Erase with special ultraviolet, and then reprogram
- EEPROM : Electronically EPROM
    - Erase with special charge, and then reprogram