Combinational Logic Design using ROM Array

Read-Only Memory (ROM)

- Combinational circuits are often referred to as memoryless circuits, because their output depends only on their current input and no history of prior inputs is retained.
- There is a type of memory that is implemented with combinational circuits, namely **read-only memory (ROM).**

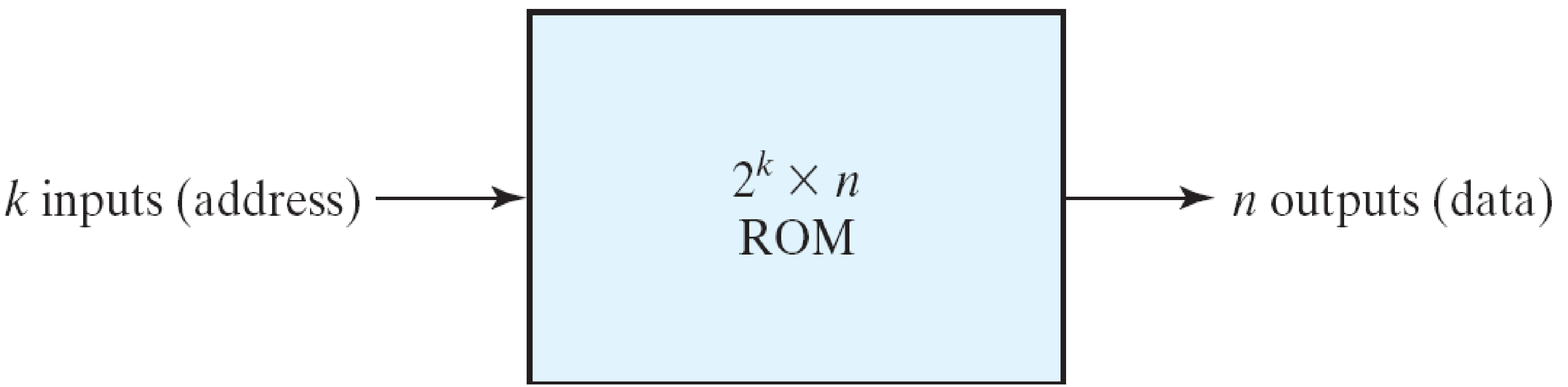
Read-Only Memory (ROM)

- ROM is a memory unit that performs only the read operation
- Binary information stored in a ROM is permanent and is created during the fabrication process
- A given input to the ROM (address lines) always produces the same output (data lines)
- Because the outputs are a function only of the present inputs, ROM is a combinational circuit

- ROM is a **programmable logic device** (PLD). The binary information that is stored within such a device is specified in some fashion and then embedded within the hardware in a process is referred to as programming the device.
- Other such units of PLD are the **programmable logic array** (PLA), **programmable array logic** (PAL), and the **field-programmable gate array** (FPGA).

- A **word** is the basic unit that moves in and out of memory
- The length of a word is often multiples of a byte (=8 bits)
- Memory units are specified by its **number of words** and the **number of bits** in each word

- A ROM is essentially a memory device in which permanent binary information is stored



- Ex: 1024(words) x 16(bits)
  - Each word is assigned a particular **address**, starting from 0 up to 2^k -1 (k = number of address lines)

Read-Only Memory (ROM)

- A ROM can be implemented with a decoder and a set of OR gates.

For a 2^k x n ROM,
it consists of

- k inputs (address line) and n outputs (data)
- 2^k words of n-bit each
- A k x 2^k decoder (generate all minterms)
- n OR gates with 2^k inputs
- Initially, all inputs of OR gates and all outputs of the decoder are fully connected

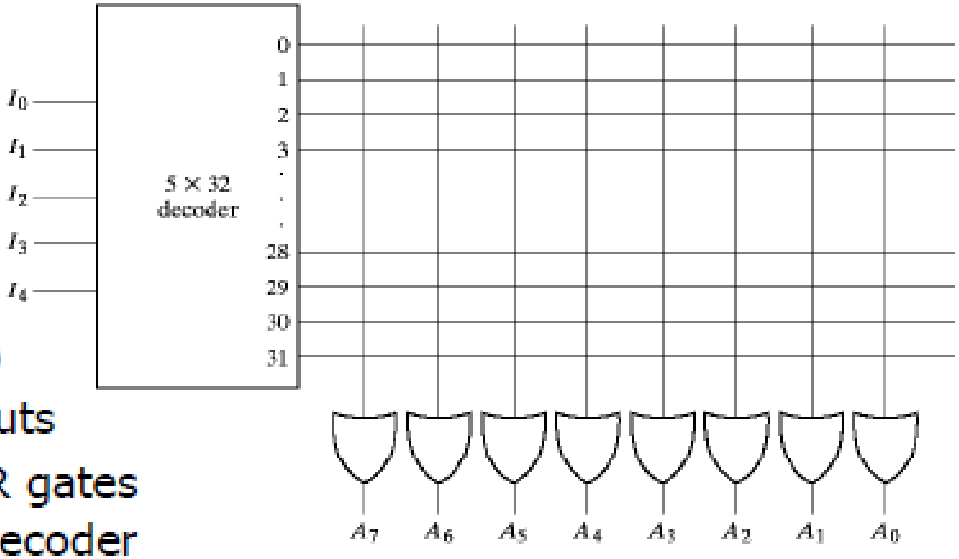
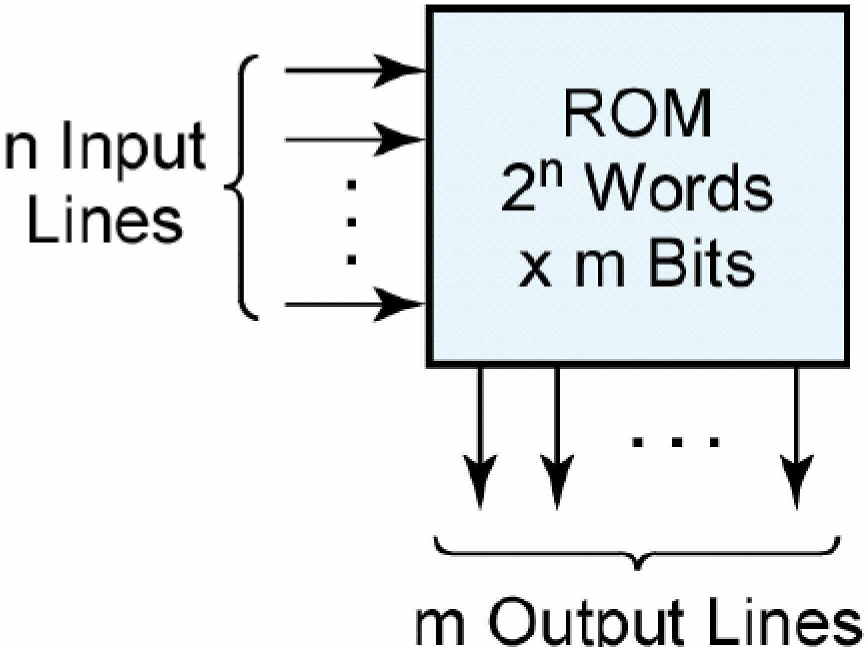


Fig. 7-10 Internal Logic of a 32 × 8 ROM

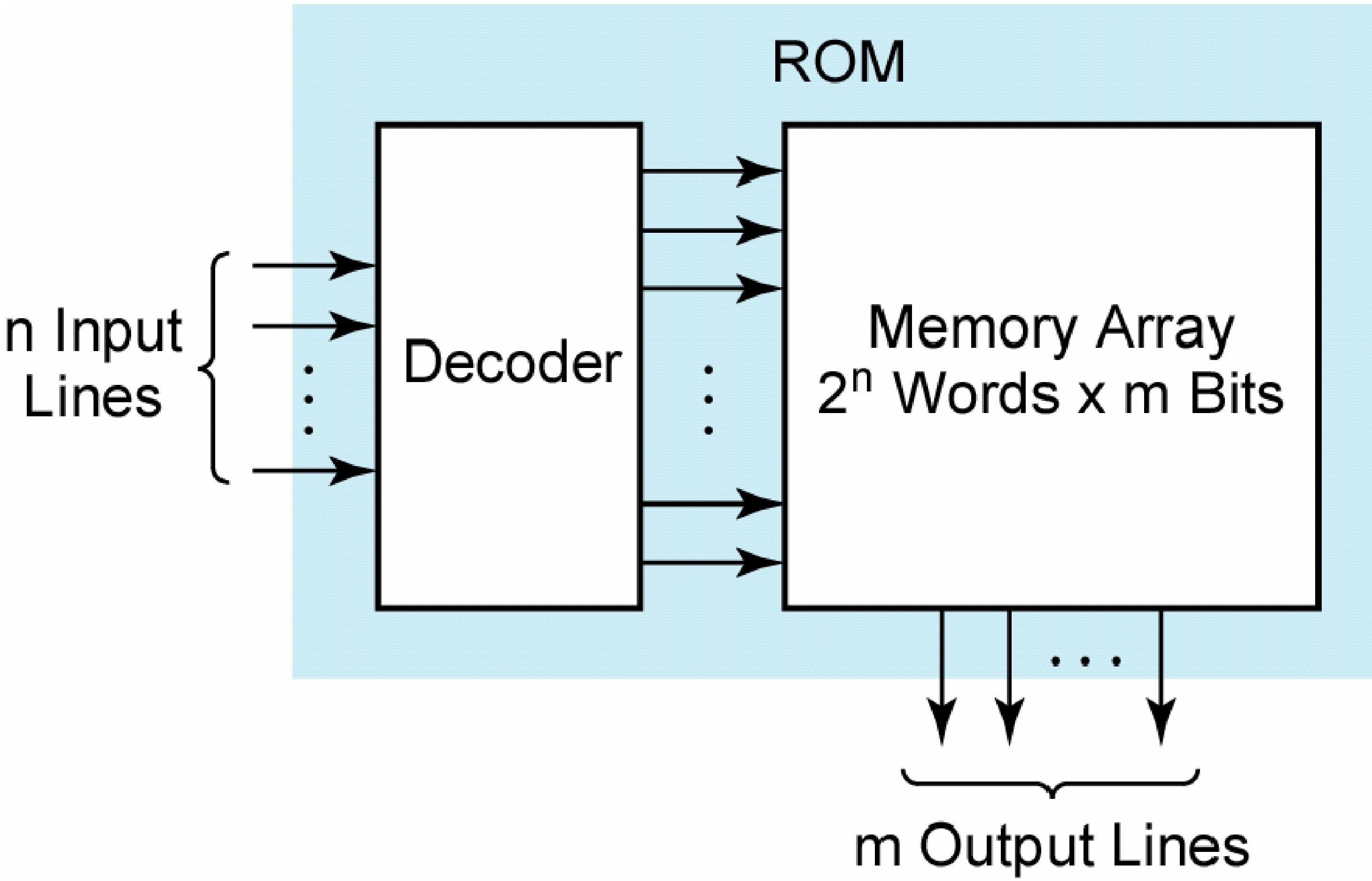
Programming the ROM

- Each intersection (crosspoint) in the ROM is often implemented with a fuse.
- Blow out unnecessary connections according to the truth table
  - 1 means connected (marked as X)
  - 0 means unconnected
- Cannot recovered after programmed

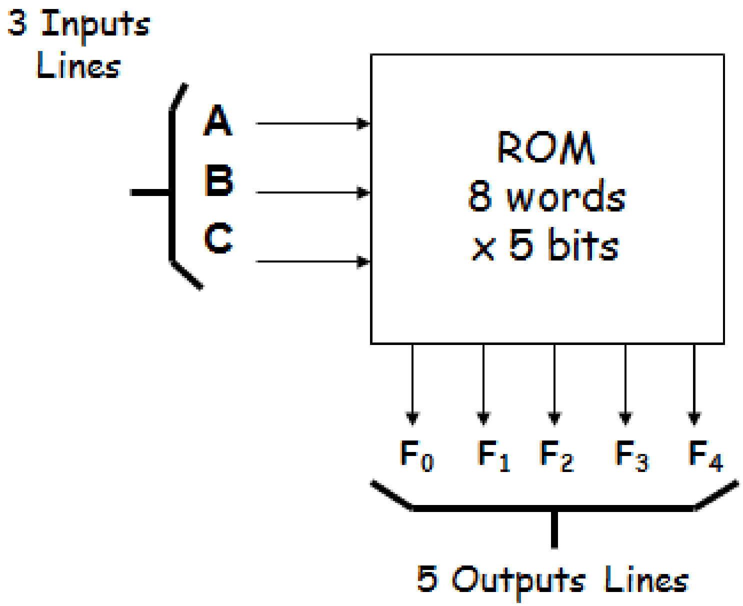


n Input Variables	*m* Output Variables
00 · · · 00	100 · · · 110
00 · · · 01	010 · · · 111
00 · · · 10	101 · · · 101
00 · · · 11	110 · · · 010
⋮	⋮
11 · · · 00	001 · · · 011
11 · · · 01	110 · · · 110
11 · · · 10	011 · · · 000
11 · · · 11	111 · · · 101

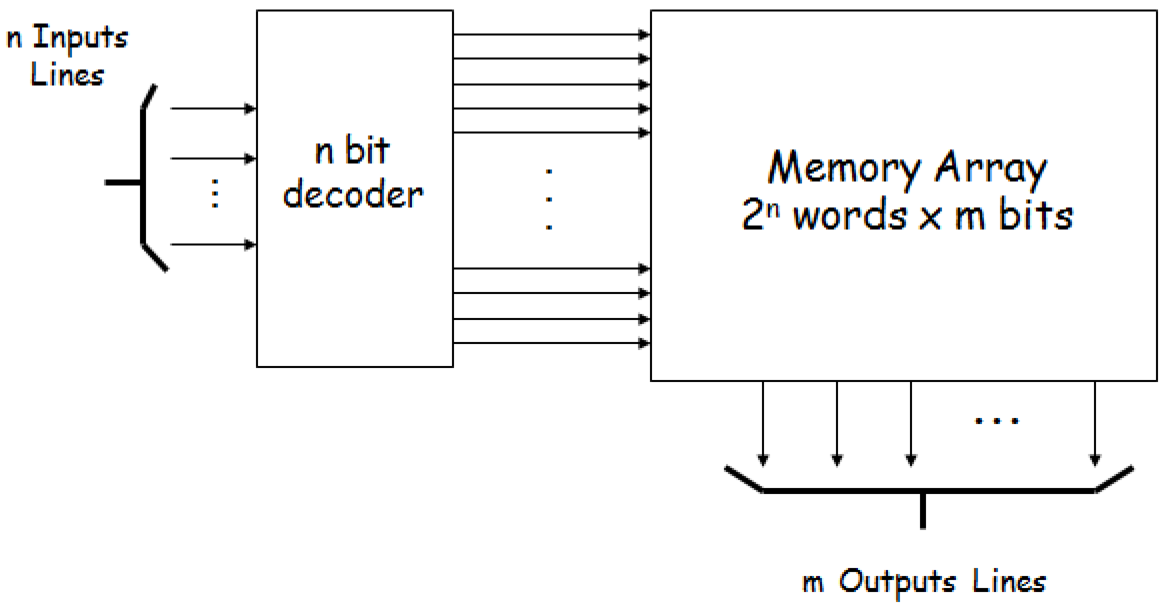
ROM Structure



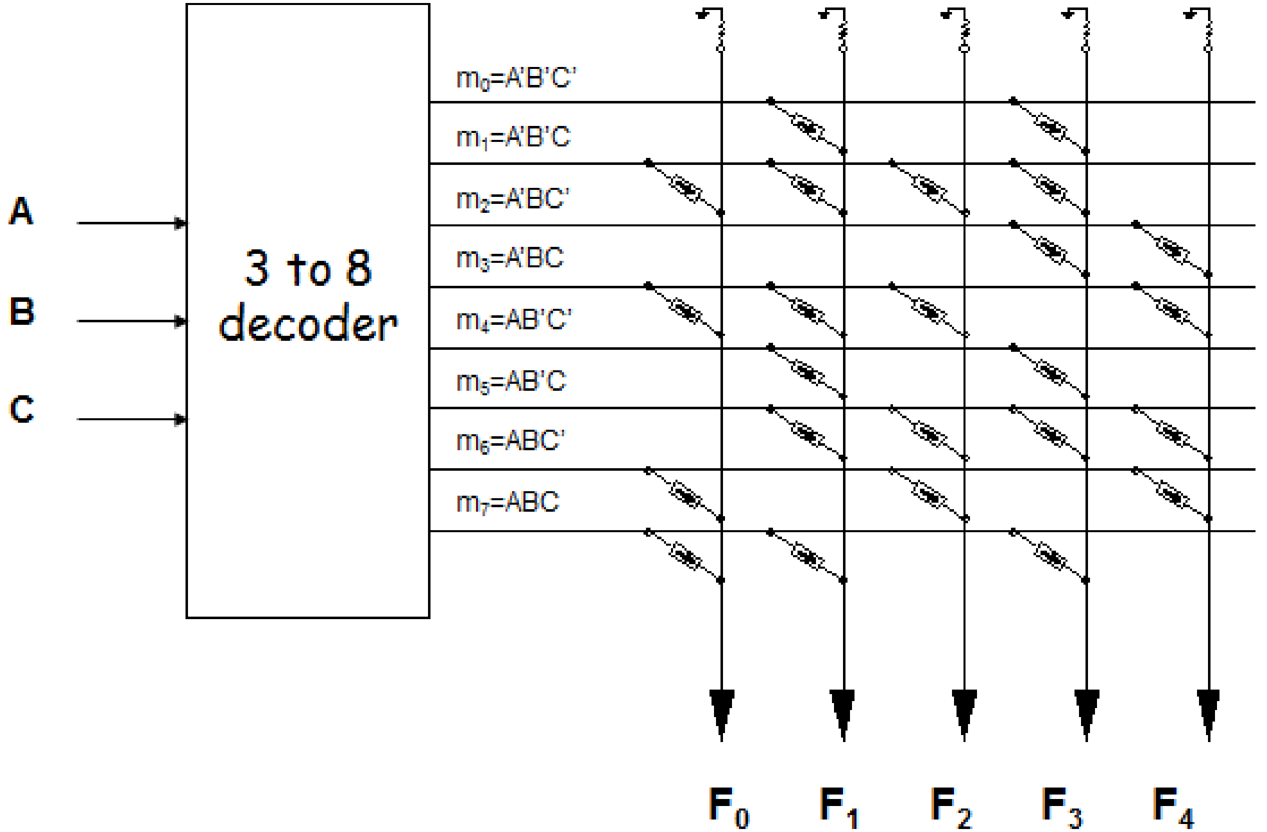
Example



A	B	C	F_0	F_1	F_2	F_3	F_4
0	0	0	0	1	0	1	0
0	0	1	1	1	1	1	0
0	1	0	0	0	0	1	1
0	1	1	1	1	1	0	1
1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	0	1
1	1	1	1	1	0	1	0

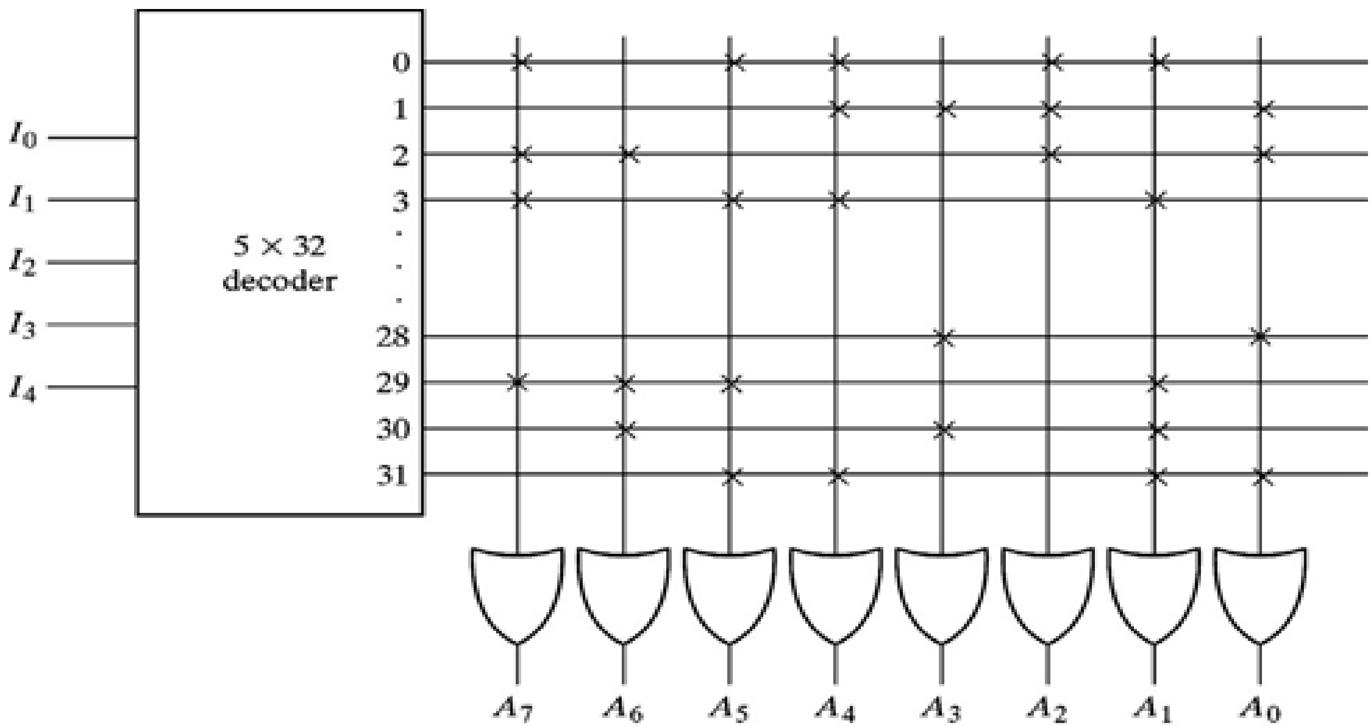

n Inputs Lines
n bit decoder
Memory Array
2^n words x m bits
m Outputs Lines

ROM Memory Array



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


I_0
I_1
I_2
I_3
I_4
5 × 32
decoder
0
1
2
3
28
29
30
31
A_7
A_6
A_5
A_4
A_3
A_2
A_1
A_0

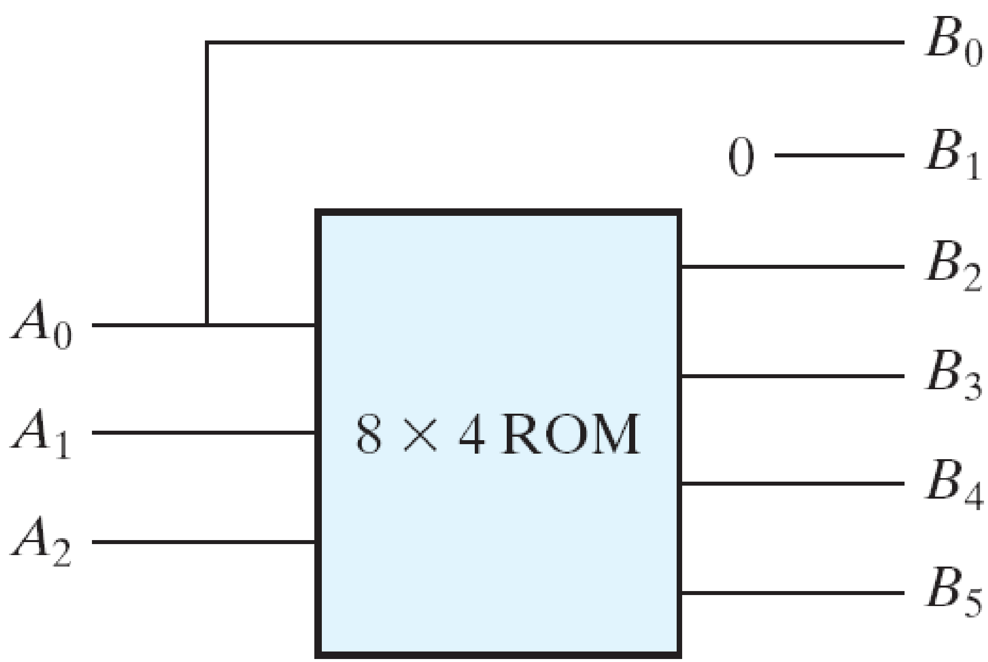
Design Comb. Circuit with ROM

- Derive the truth table of the circuit
- Determine minimum size of ROM
- Program the ROM

- Design a combinational circuit using a ROM. The circuit accepts a three-bit number and outputs a binary number equal to the square of the input number.

Example

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

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table