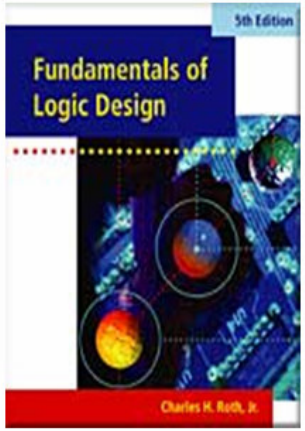
FIGURES FOR

CHAPTER 9

MULTIPLEXERS, DECODERS, AND PROGRAMMABLE LOGIC DEVICES



This chapter in the book includes:

Click the mouse to move to the next page.
Use the ESC key to exit this chapter.

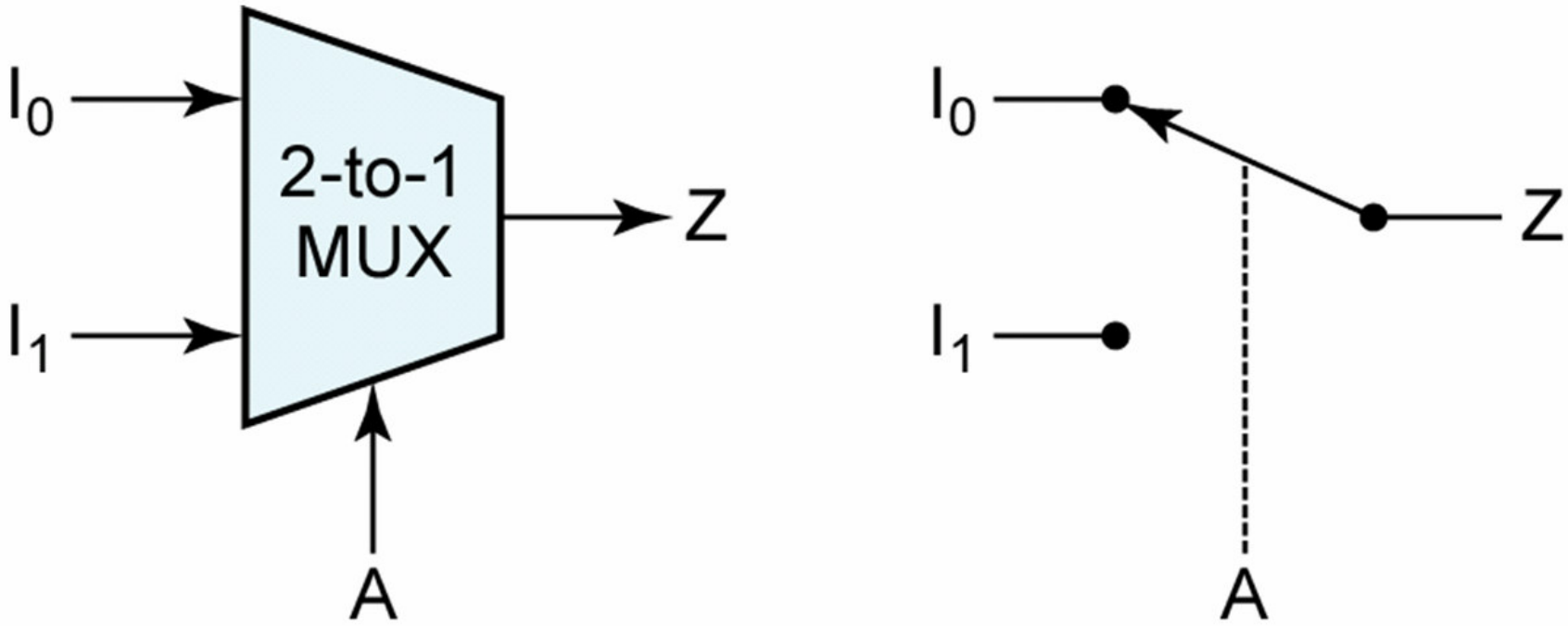


Figure 9-1: **2-to-1 Multiplexer and Switch Analog**

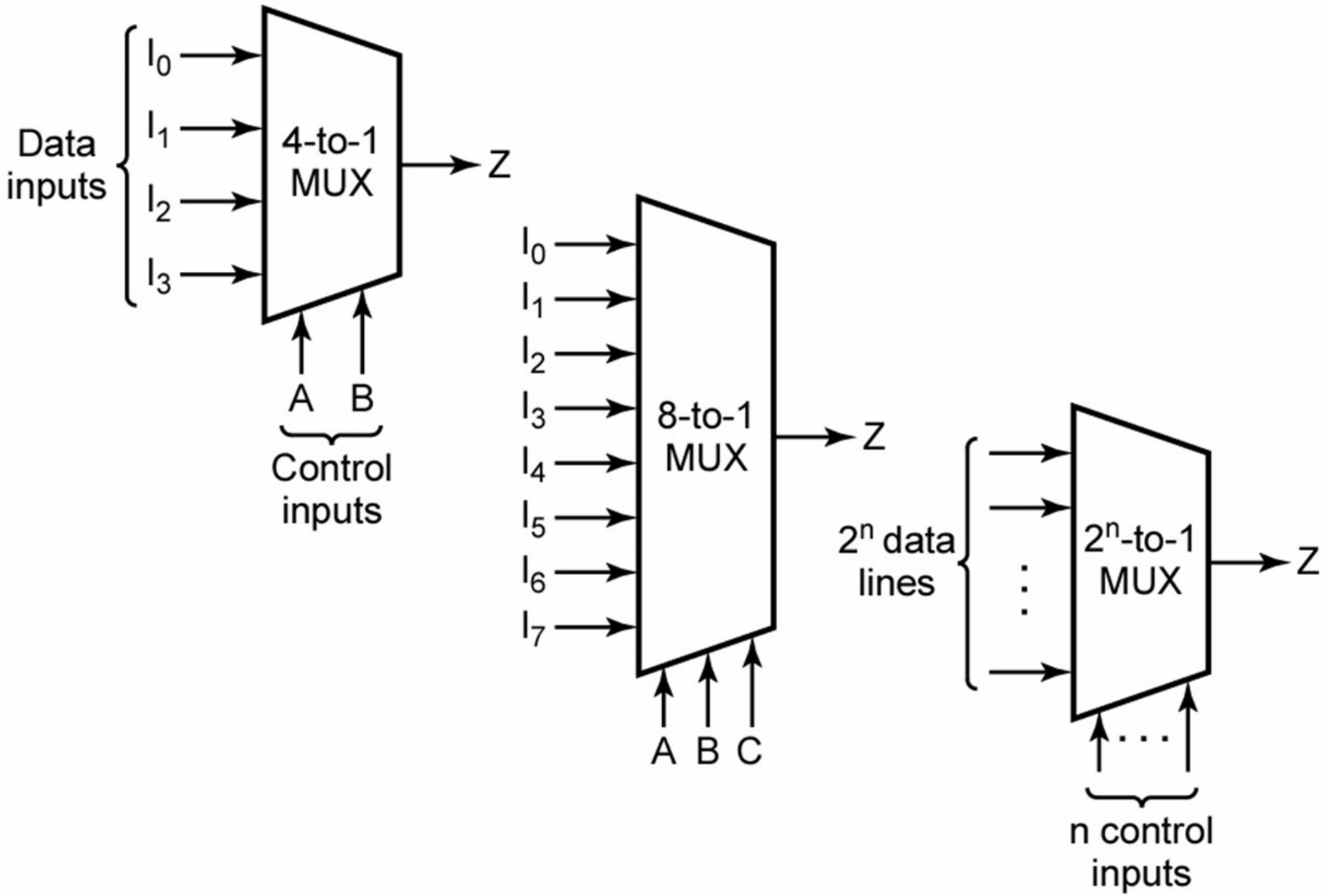


Figure 9-2: **Multiplexers**

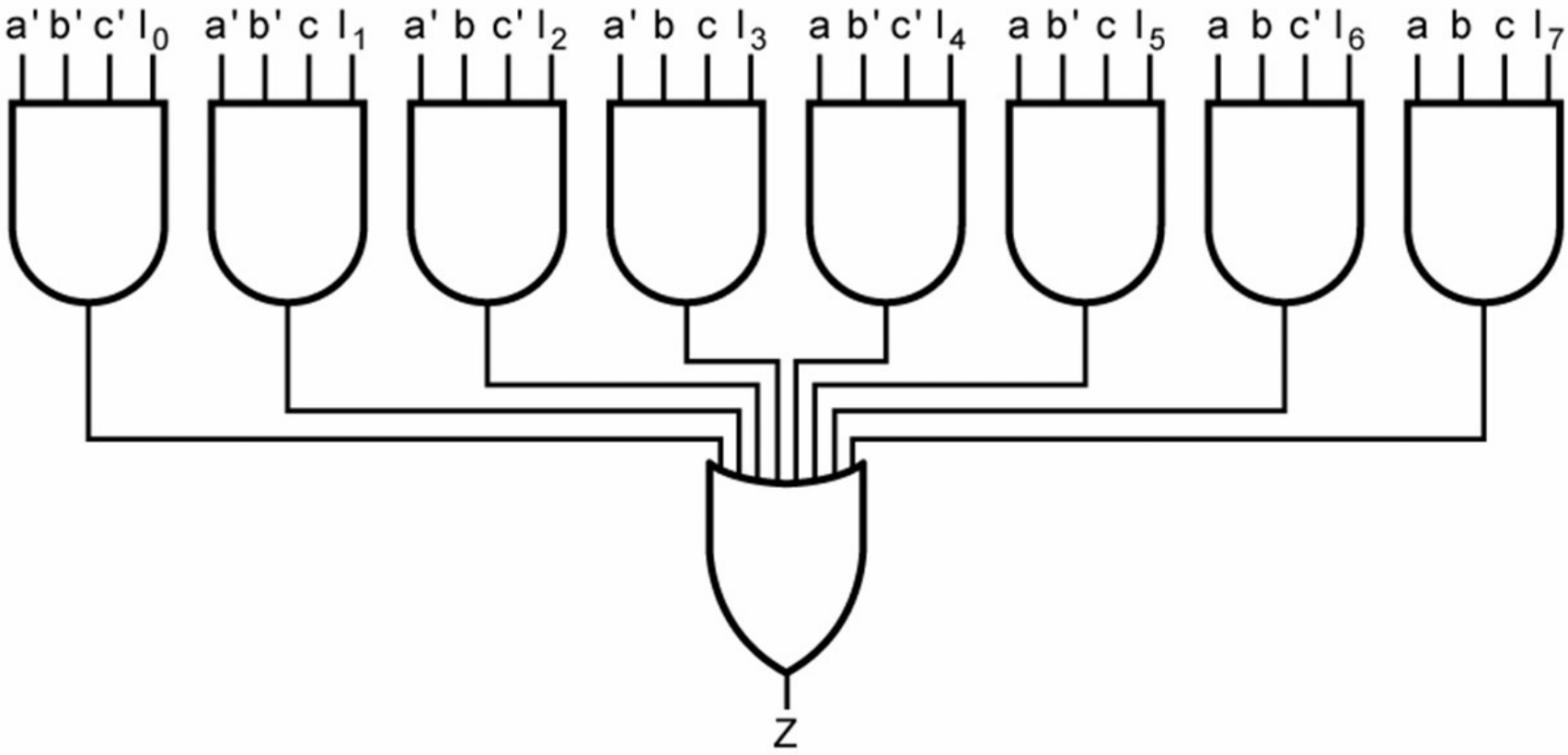


Figure 9-3: **Logic Diagram for 8-to-1 MUX**

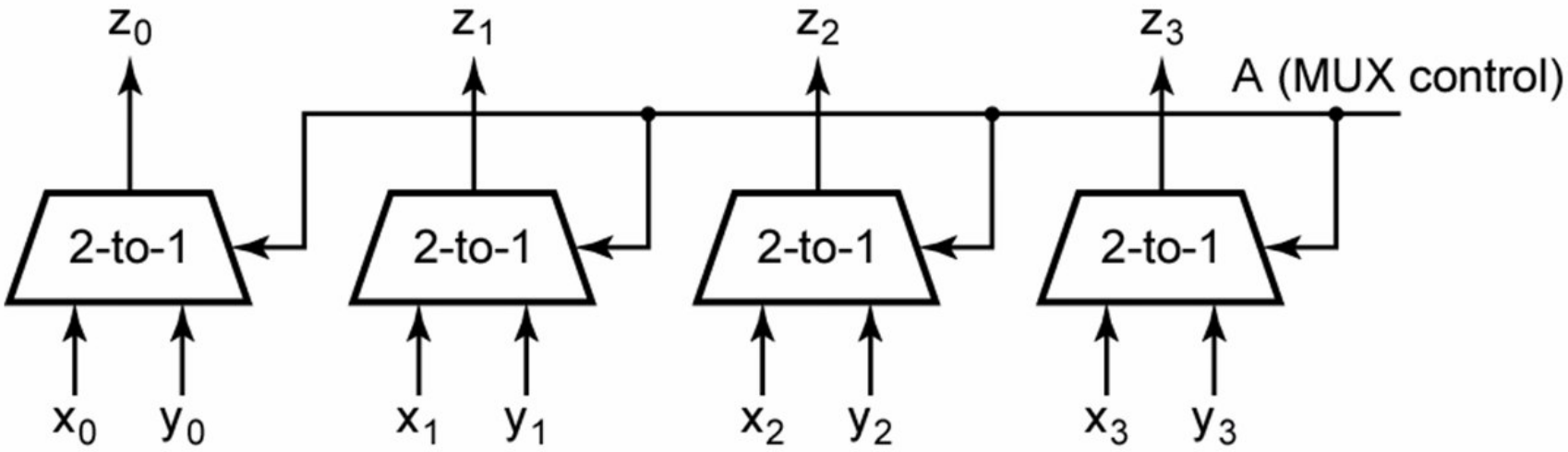


Figure 9-4: **Quad Multiplexer Used to Select Data**

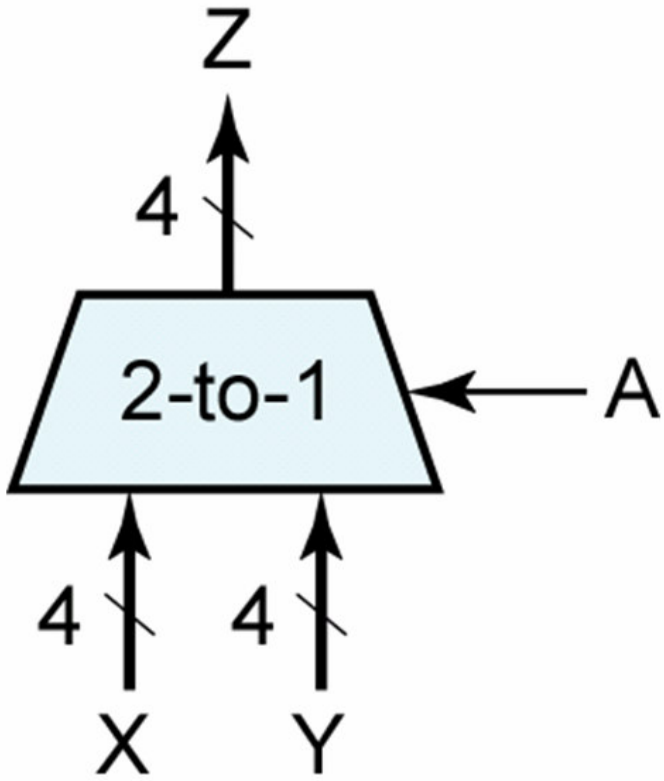


Figure 9-5: **Quad Multiplexer with Bus Inputs and Output**

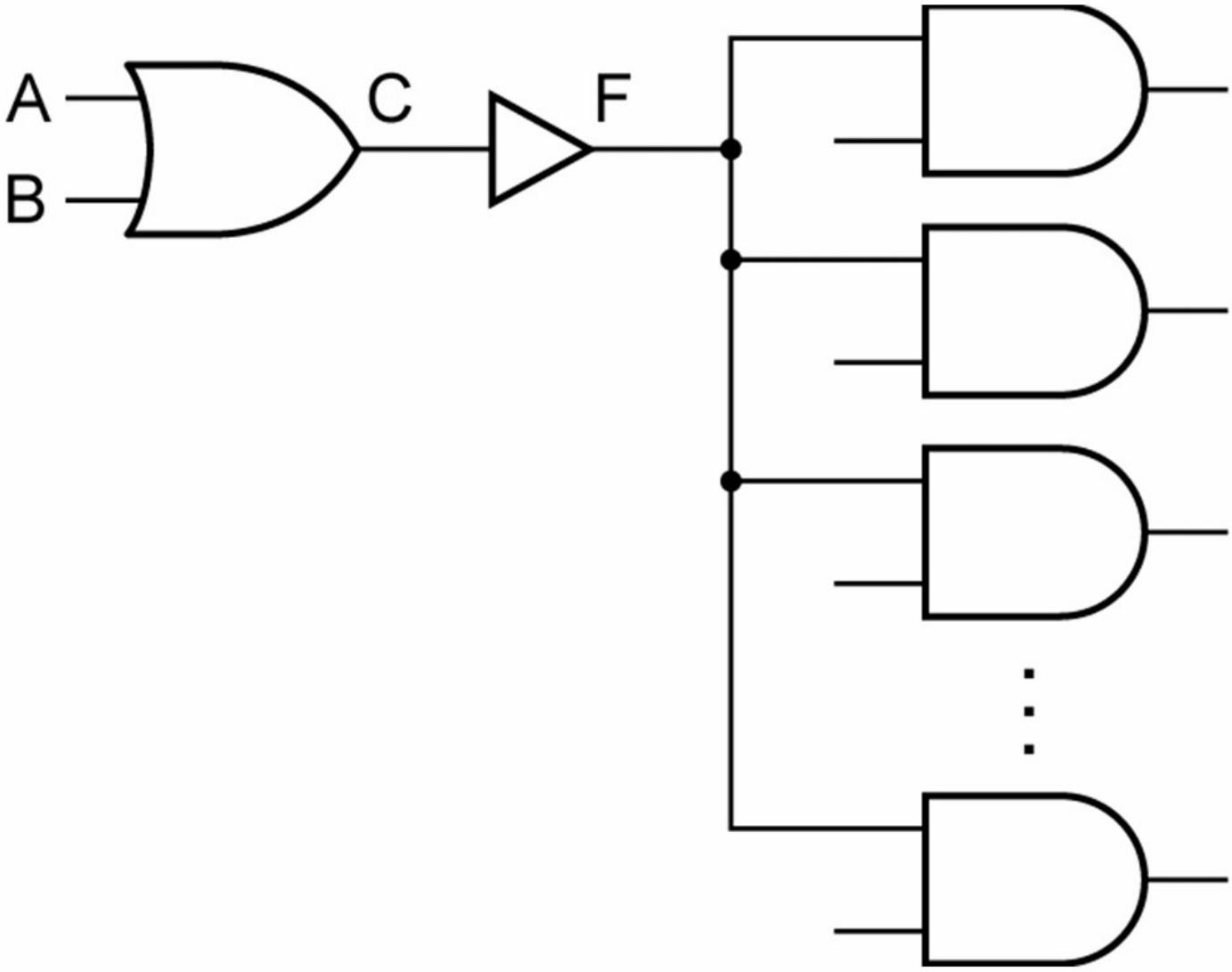


Figure 9-6: **Gate Circuit with Added Buffer**

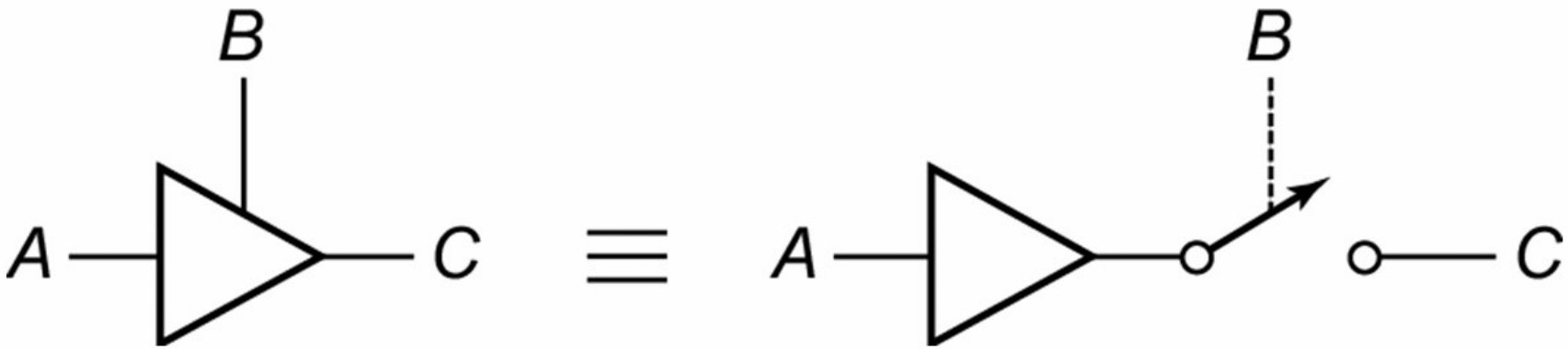
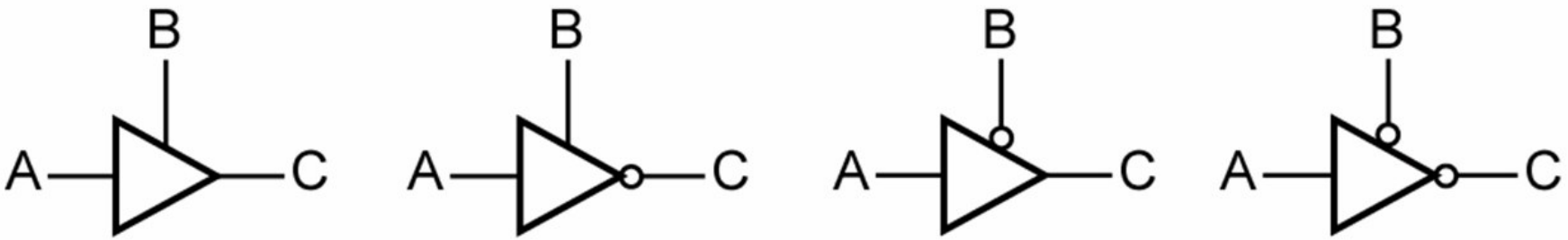


Figure 9-7: **Three-State Buffer**



B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

(a)

B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

(b)

B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

(c)

B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

(d)

Figure 9-8: **Four Kinds of Three-State Buffers**

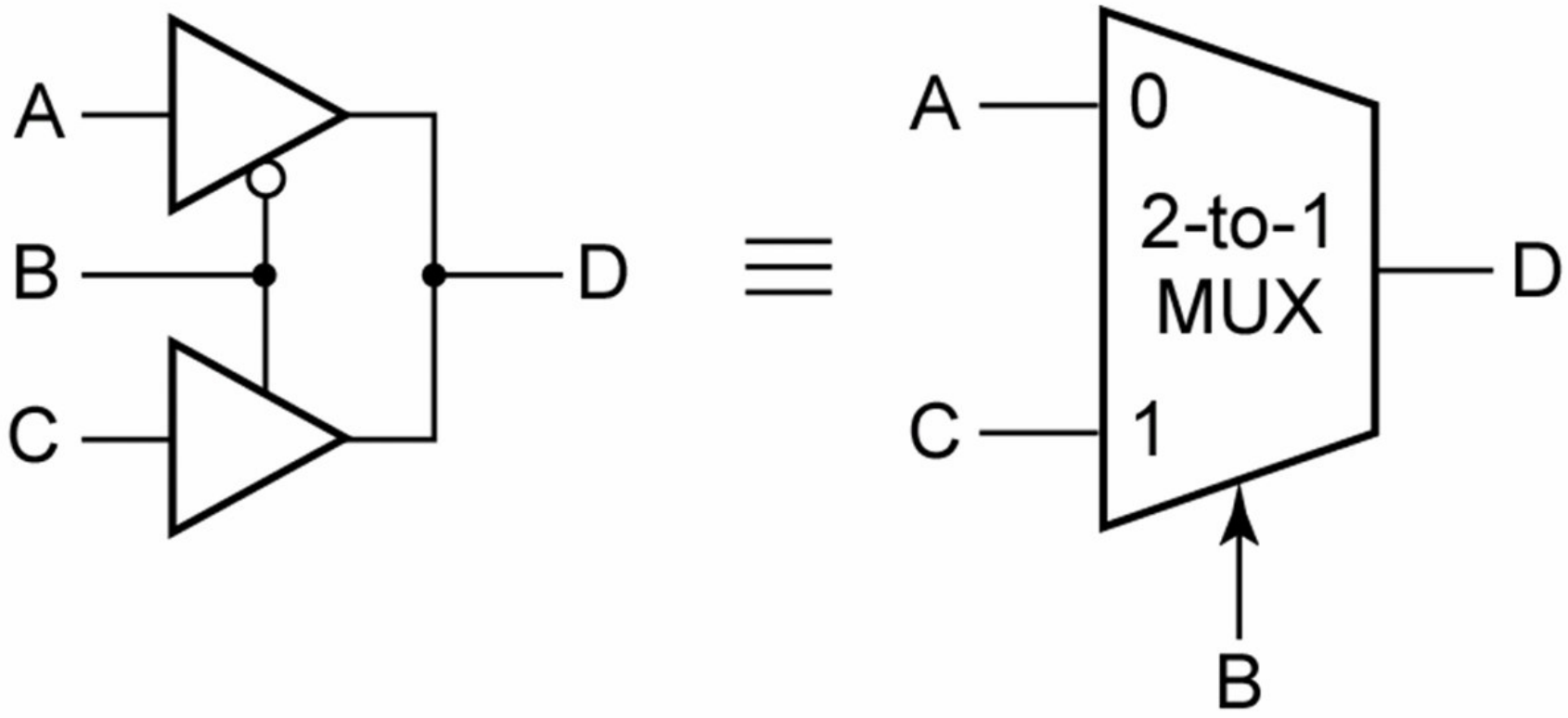
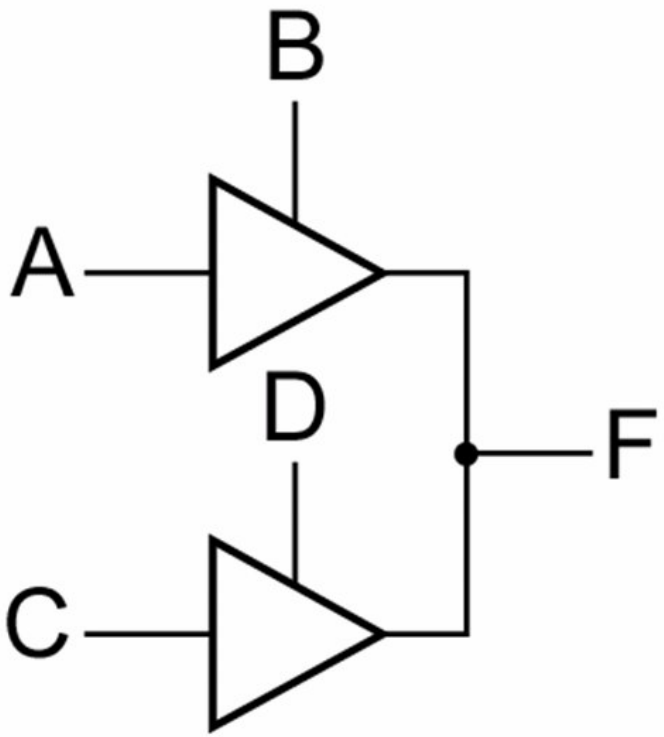


Figure 9-9: **Data Selection Using Three-State Buffers**



S1 \ S2	X	0	1	Z
X	X	X	X	X
0	X	0	X	0
1	X	X	1	1
Z	X	0	1	Z

Figure 9-10: **Circuit with Two Three-State Buffers**

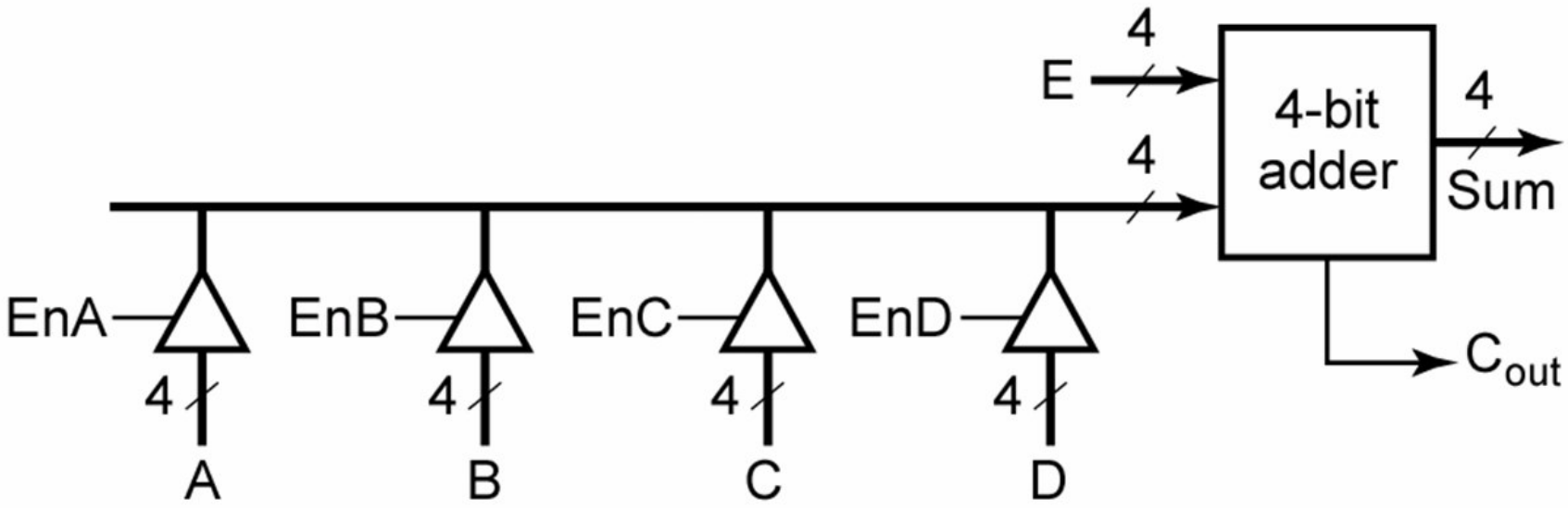


Figure 9-11: **4-Bit Adder with Four Sources for One Operand**

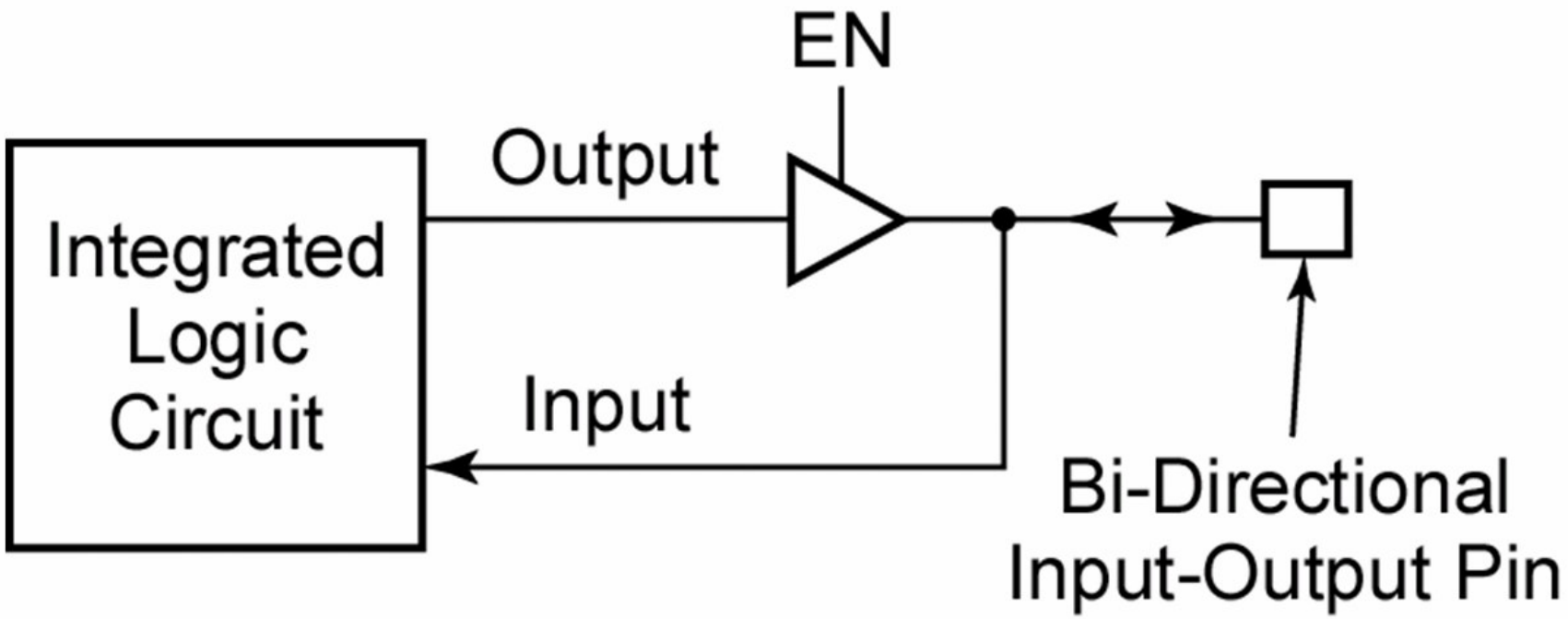


Figure 9-12: **Integrated Circuit with Bi-Directional Input/Output Pin**

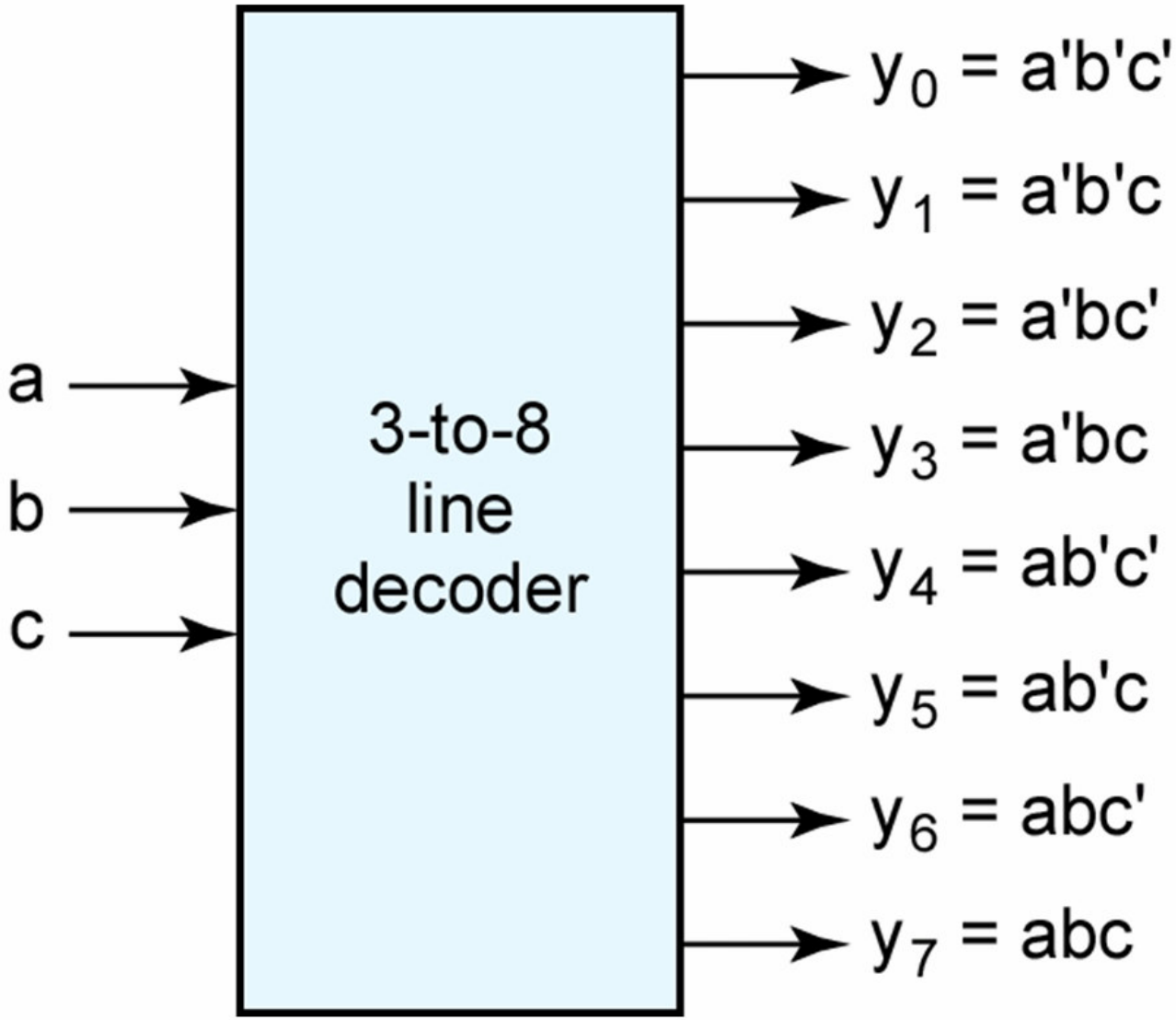
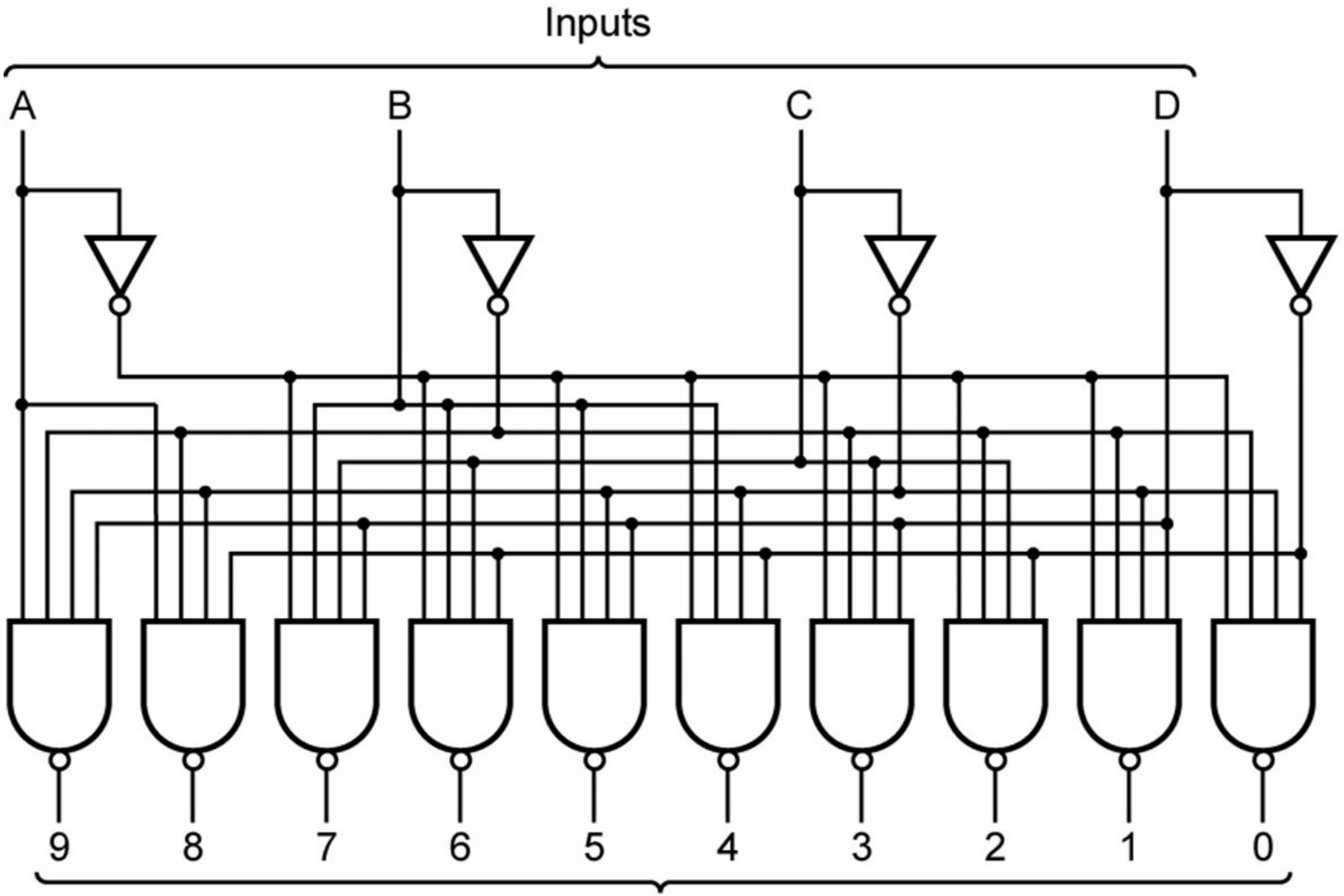


Figure 9-13: **3-to-8 Line Decoder**

a	b	c	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_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

Figure 9-13: **3-to-8 Line Decoder**



(a) Logic diagram

Figure 9-14a:
A 4-to-10 Line Decoder

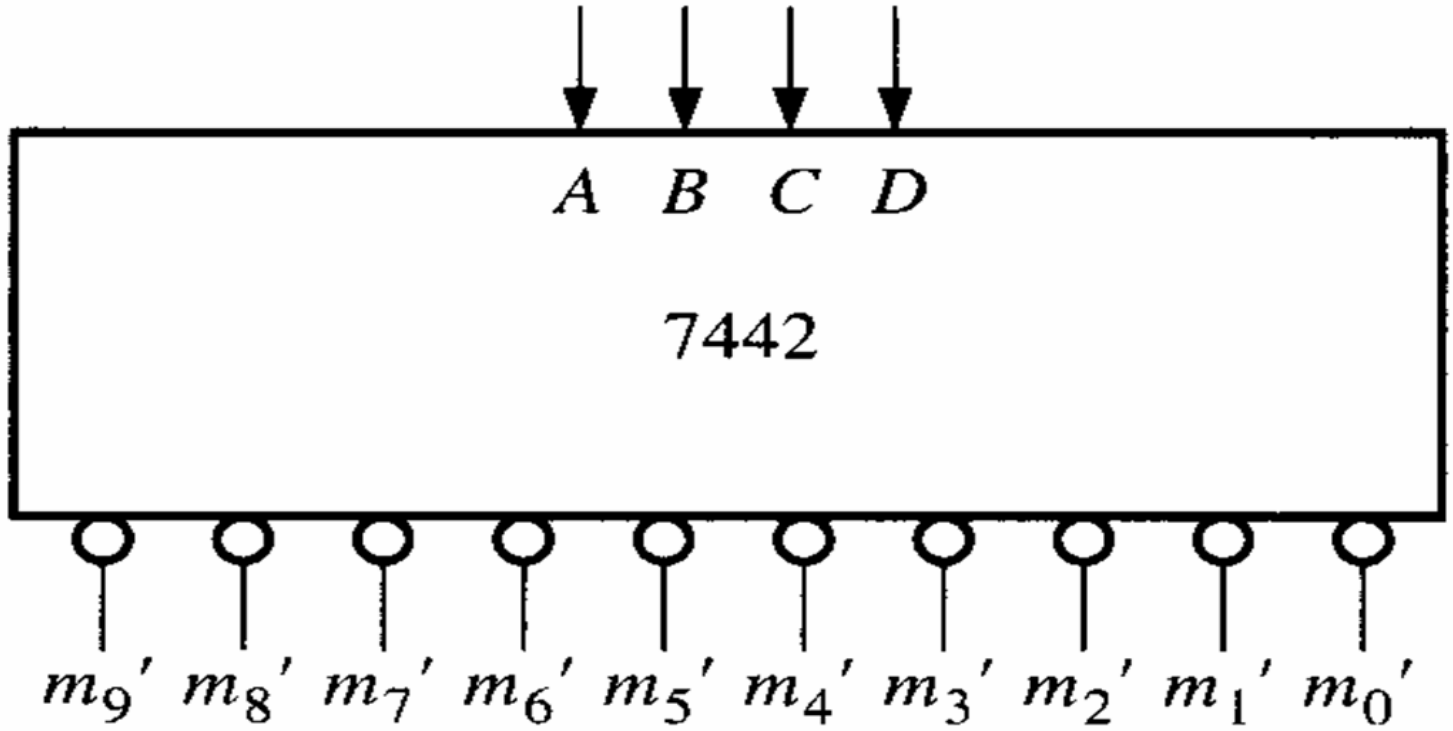


Figure 9-14b:
A 4-to-10 Line Decoder

(b) Block diagram

BCD Input				Decimal Output									
A	*B*	*C*	*D*	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1

Figure 9-14c:
A 4-to-10 Line Decoder

(c) Truth Table

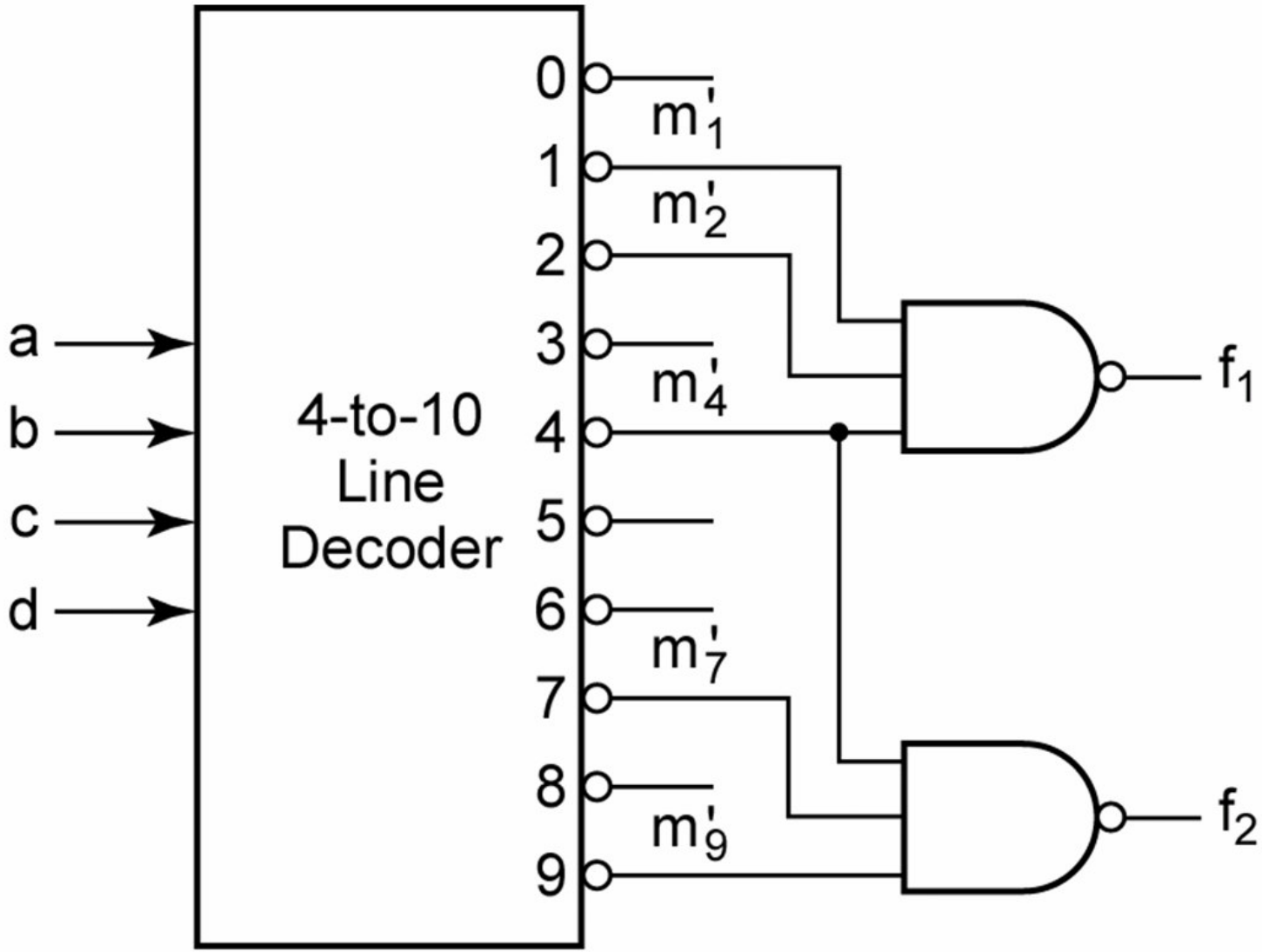
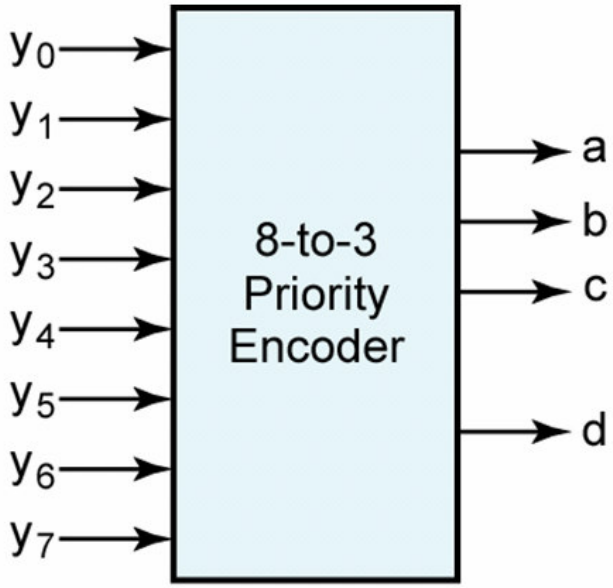
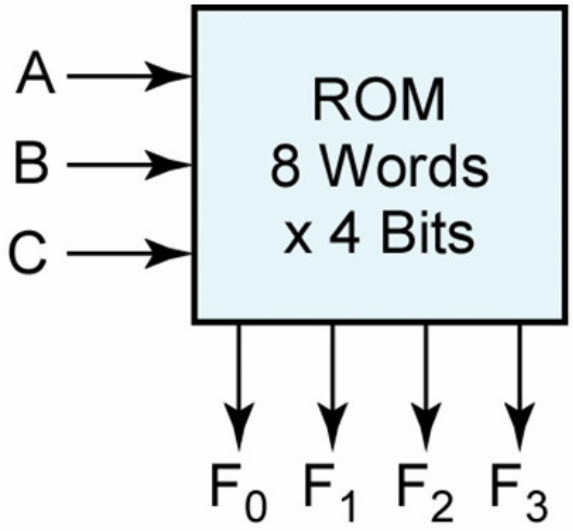


Figure 9-15: **Realization of a Multiple-Output Circuit Using a Decoder**



y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
X	1	0	0	0	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0	1
X	X	X	1	0	0	0	0	0	1	1	1
X	X	X	X	1	0	0	0	1	0	0	1
X	X	X	X	X	1	0	0	1	0	1	1
X	X	X	X	X	X	1	0	1	1	0	1
X	X	X	X	X	X	X	1	1	1	1	1

Figure 9-16: **8-to-3 Priority Coder**



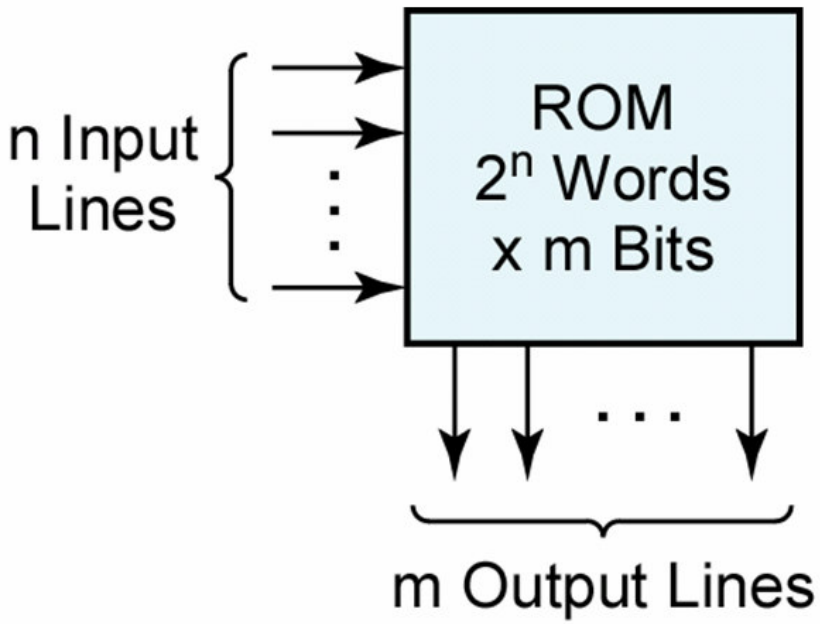
(a) Block diagram

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

typical data stored in ROM (2^3 words of 4 bits each)

(b) truth table for ROM

Figure 9-17: **An 8-Word x 4-Bit ROM**



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

Figure 9-18: **Read-Only Memory with *n* Inputs and *m* Outputs**

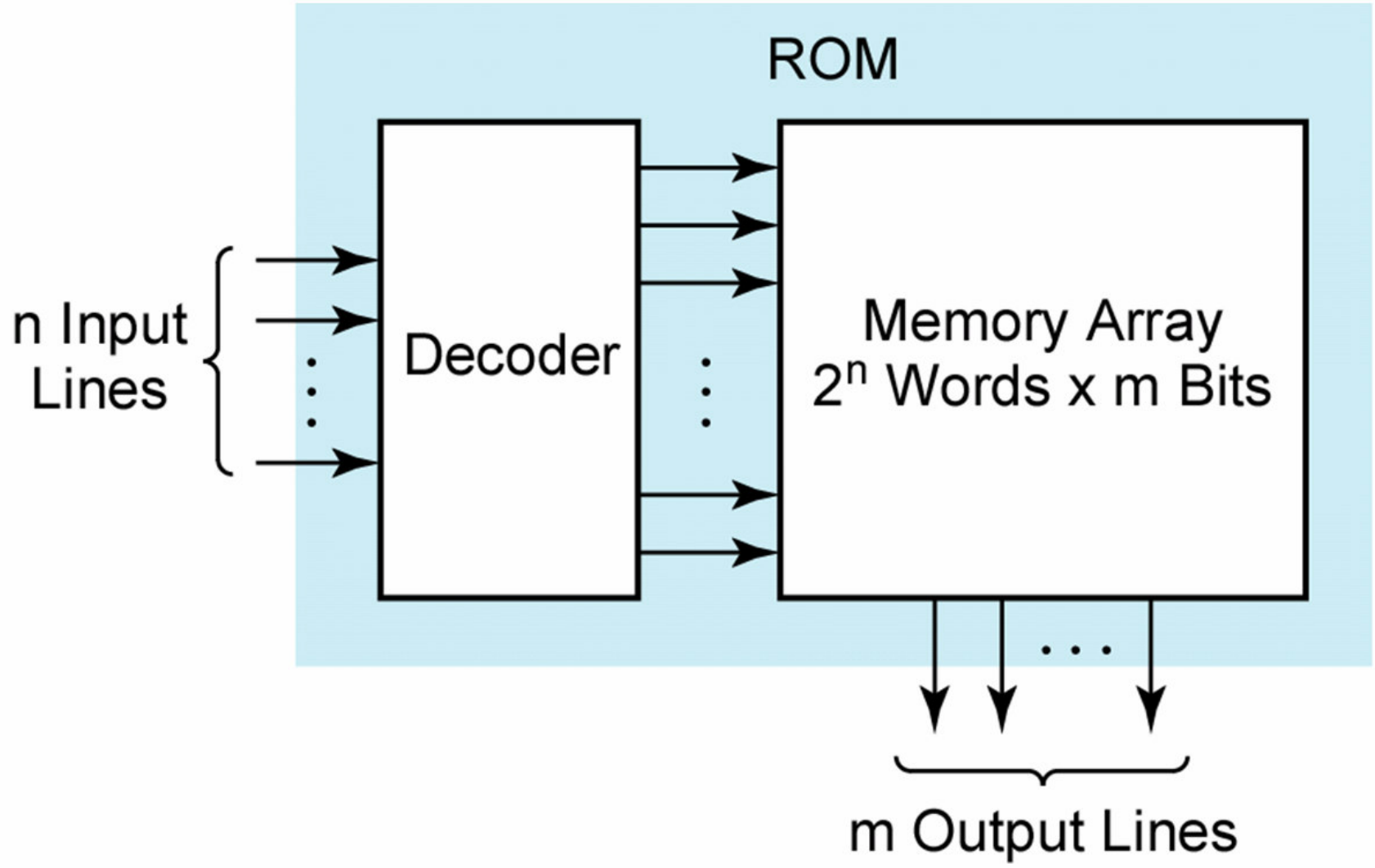


Figure 9-19: **Basic ROM Structure**

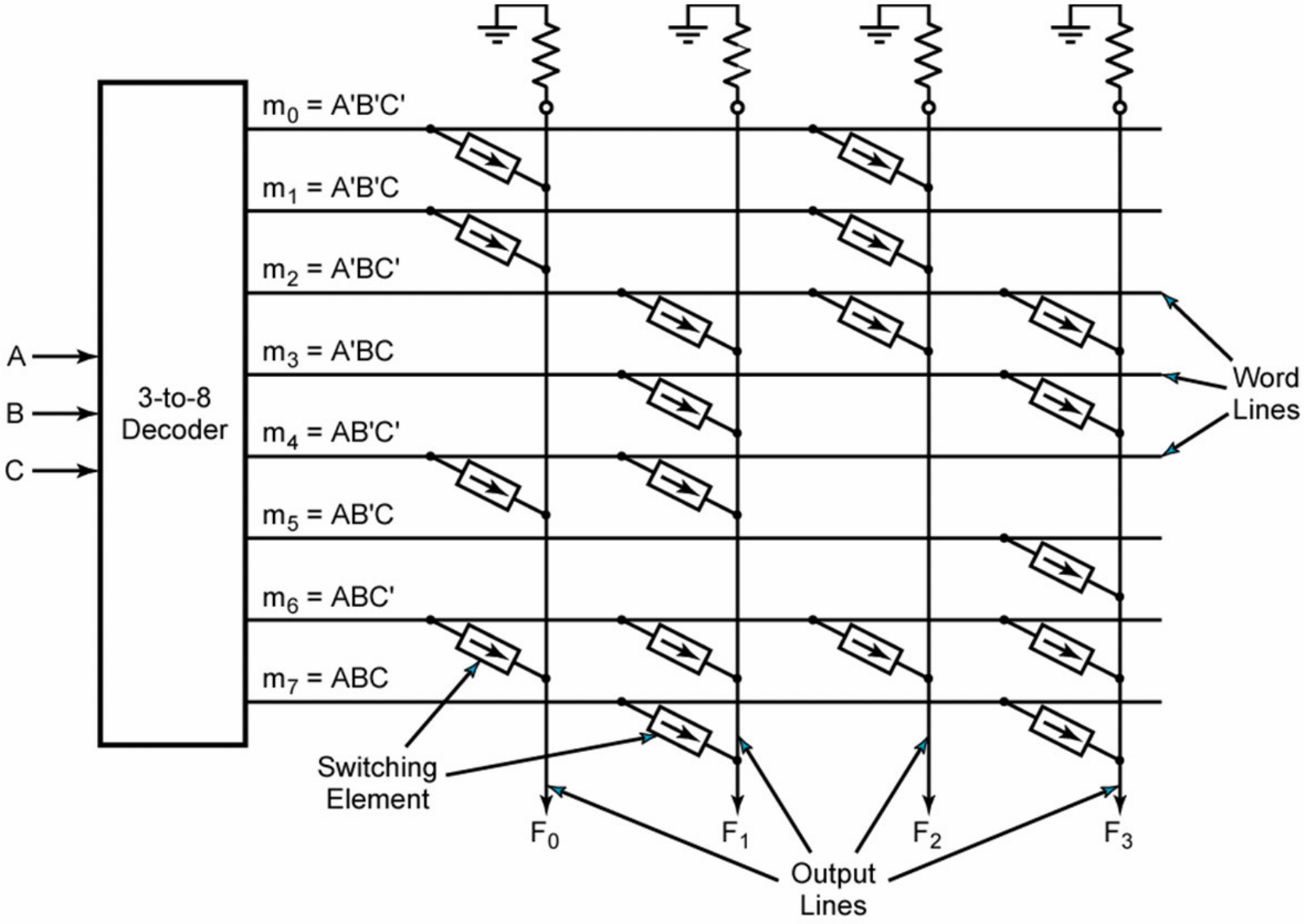


Figure 9-20: **An 8-Word x 4-Bit ROM**

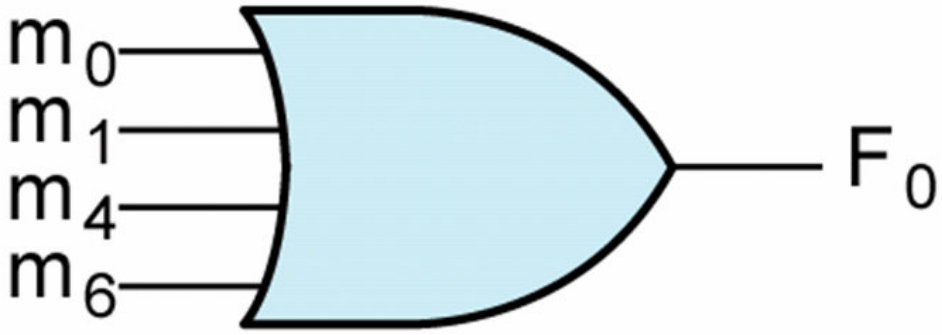


Figure 9-21: **Equivalent OR Gate for F_0**

Input				Hex	ASCII Code for Hex Digit						
W	X	Y	Z	Digit	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	1	1	0	0	0	1
0	0	1	0	2	0	1	1	0	0	1	0
0	0	1	1	3	0	1	1	0	0	1	1
0	1	0	0	4	0	1	1	0	1	0	0
0	1	0	1	5	0	1	1	0	1	0	1
0	1	1	0	6	0	1	1	0	1	1	0
0	1	1	1	7	0	1	1	0	1	1	1
1	0	0	0	8	0	1	1	1	0	0	0
1	0	0	1	9	0	1	1	1	0	0	1
1	0	1	0	A	1	0	0	0	0	0	1
1	0	1	1	B	1	0	0	0	0	1	0
1	1	0	0	C	1	0	0	0	0	1	1
1	1	0	1	D	1	0	0	0	1	0	0
1	1	1	0	E	1	0	0	0	1	0	1
1	1	1	1	F	1	0	0	0	1	1	0

Figure 9-22: **Hexadecimal to ASCII Code Converter**

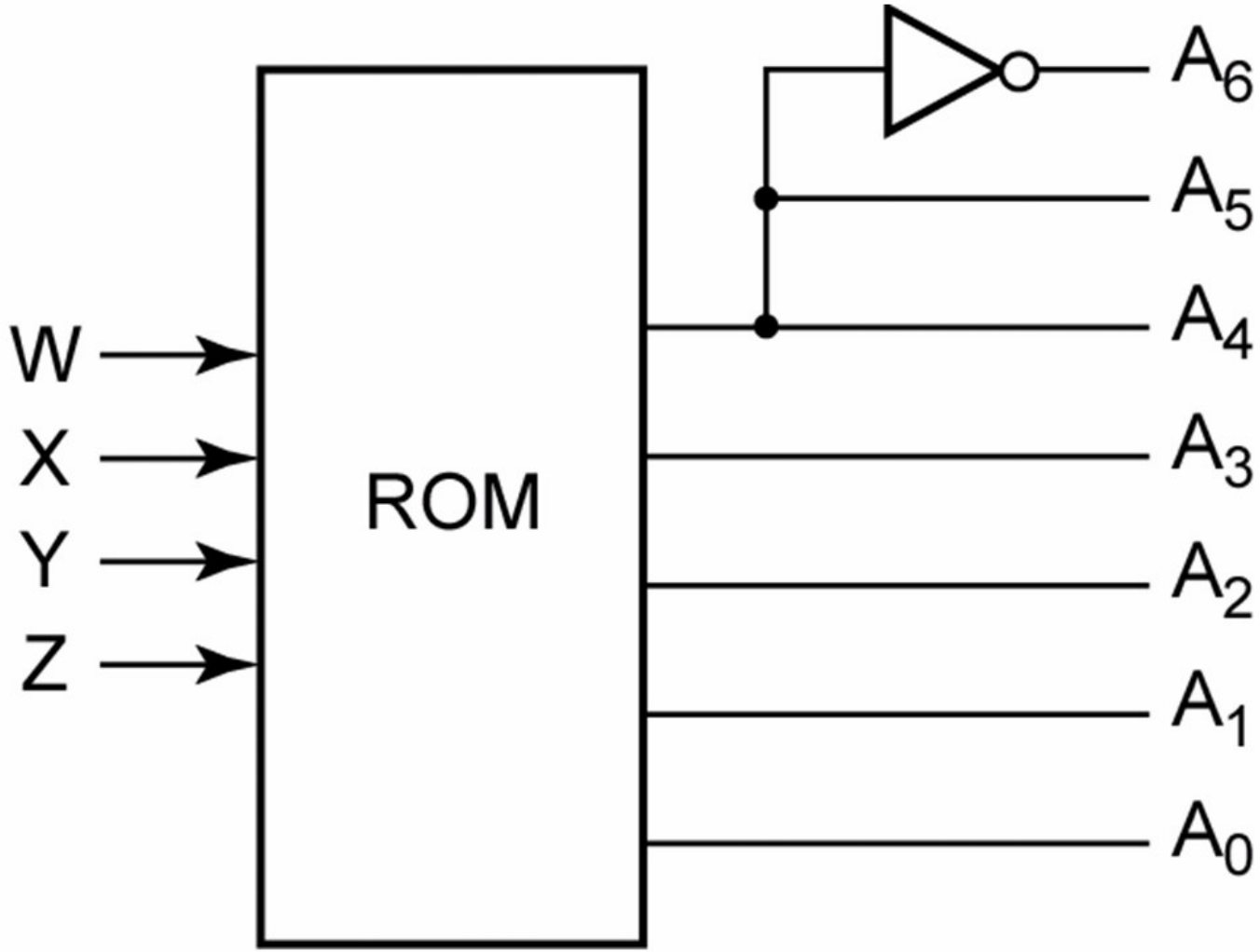


Figure 9-22: **Hexadecimal to ASCII Code Converter**

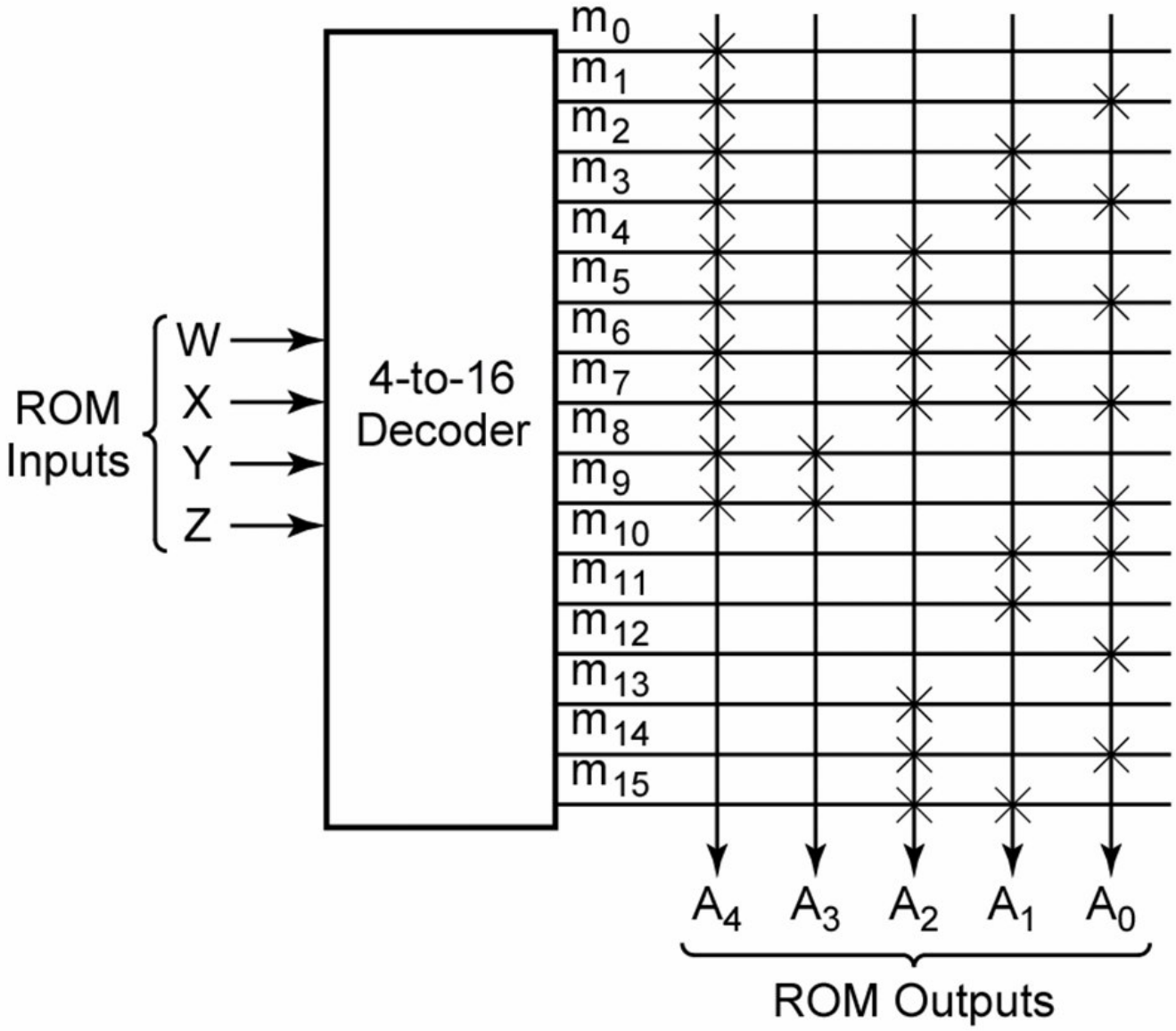


Figure 9-23: **ROM Realization of Code Converter**

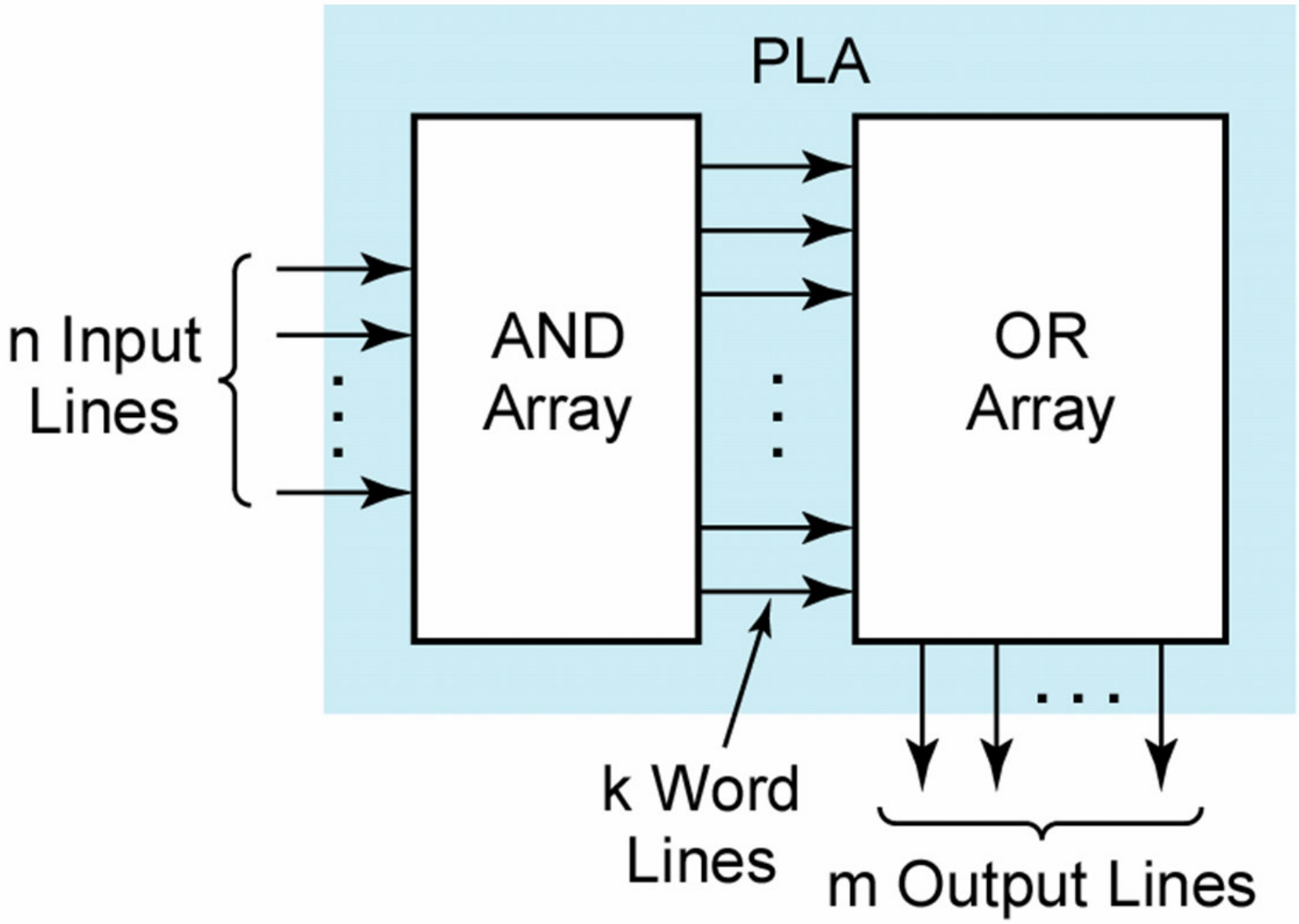


Figure 9-24: **Programmable Logic Array Structure**

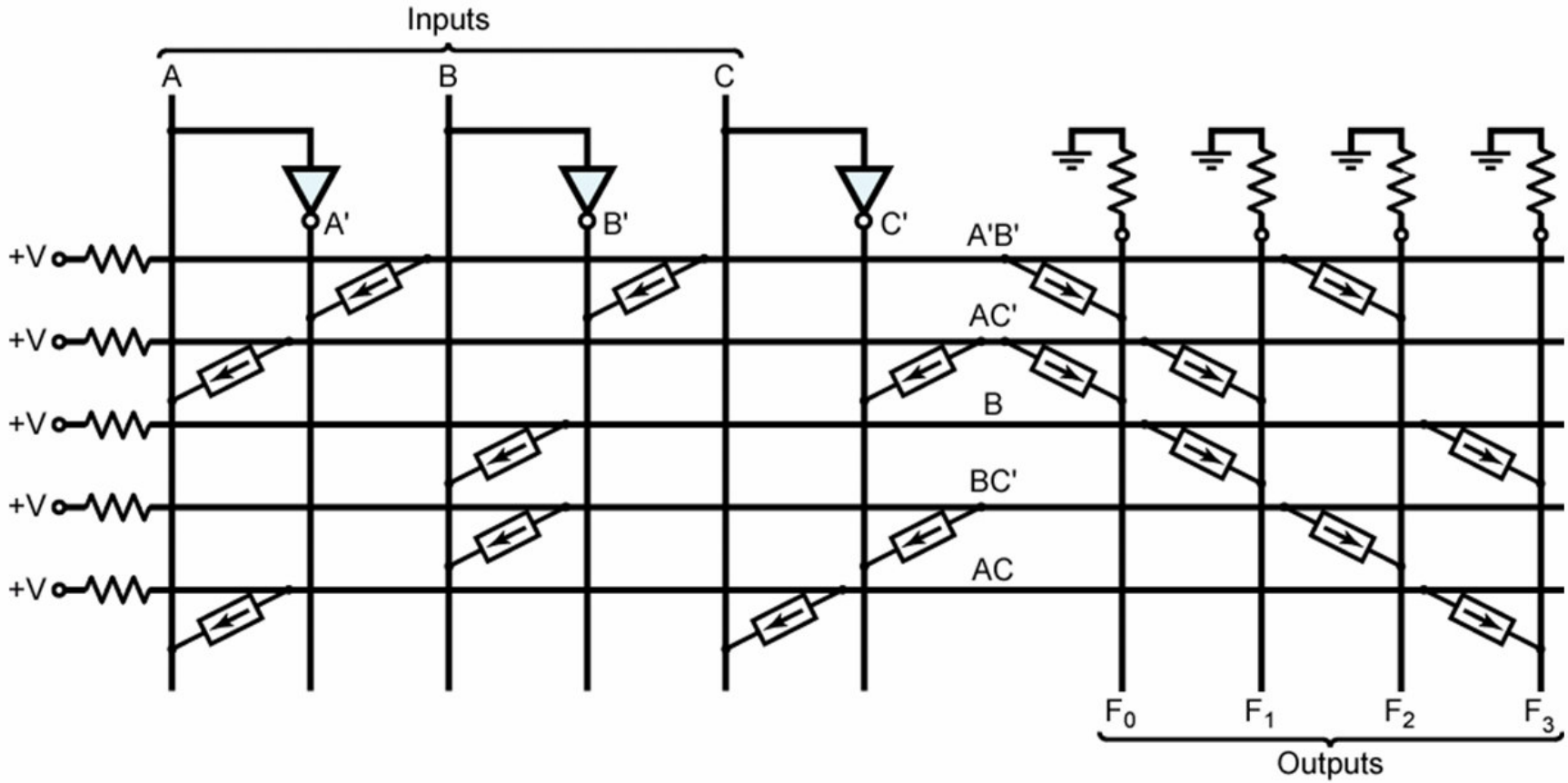


Figure 9-25: **PLA with Three Inputs, Five Product Terms, and Four Outputs**

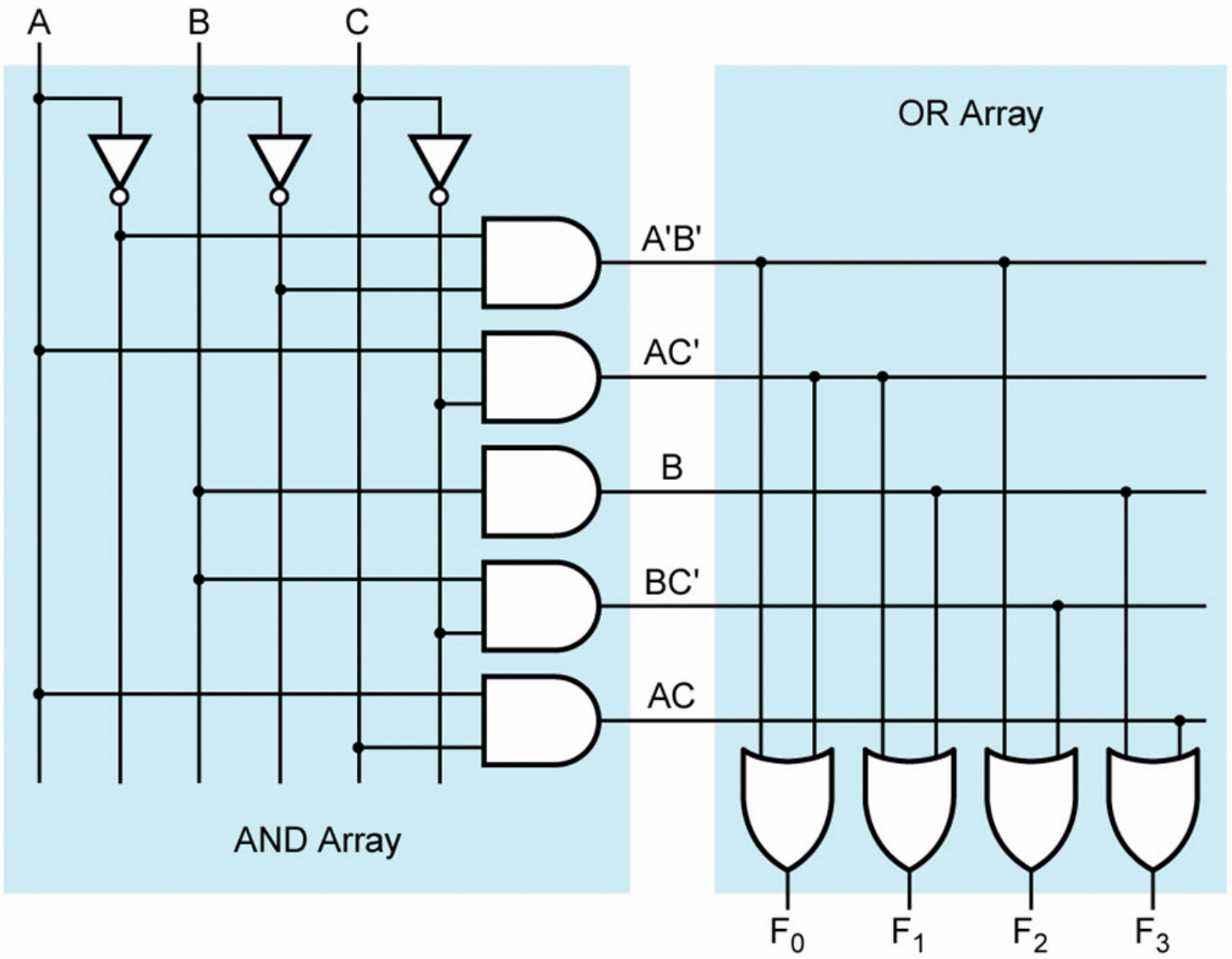


Figure 9-26: **AND-OR Array Equivalent to Figure 9-25**

Table 9-1. PLA Table for Figure 9-25

Product Term	Inputs A B C	Outputs F_0 F_1 F_2 F_3
$A'B'$	0 0 –	1 0 1 0
AC'	1 – 0	1 1 0 0
B	– 1 –	0 1 0 1
BC'	– 1 0	0 0 1 0
AC	1 – 1	0 0 0 1

$$F_0 = A'B' + AC'$$
$$F_1 = AC' + B$$
$$F_2 = A'B' + BC'$$
$$F_3 = B + AC$$

(a) PLA table

a	b	c	d	f_1	f_2	f_3
0	1	–	1	1	1	0
1	1	–	1	1	0	1
1	0	0	–	1	0	1
–	0	1	–	1	0	0
–	–	1	–	0	1	0
–	1	1	–	0	0	1

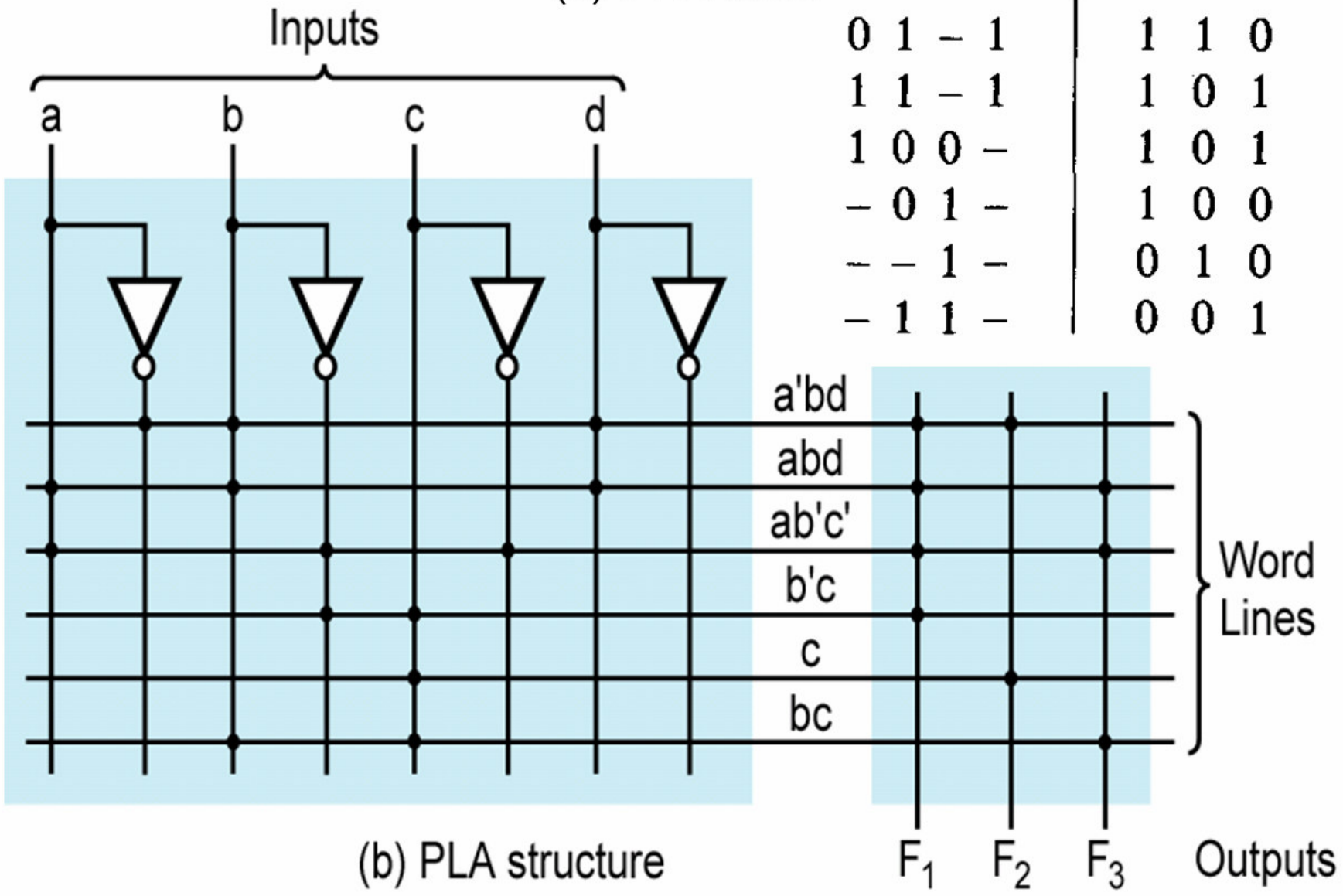
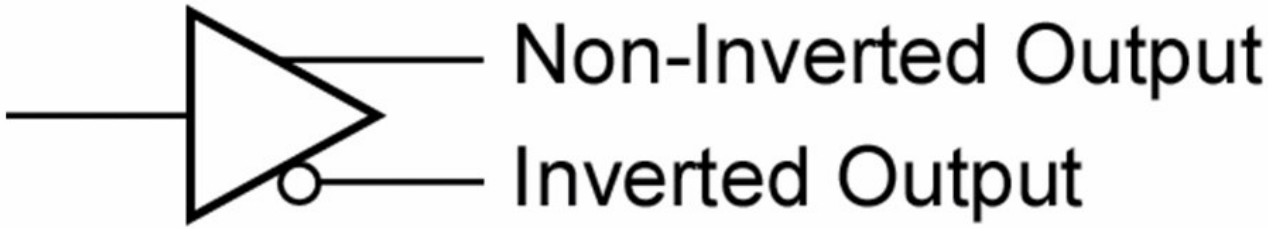
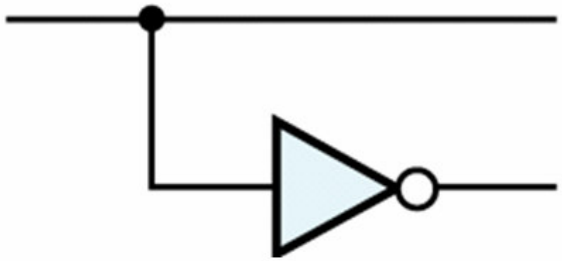


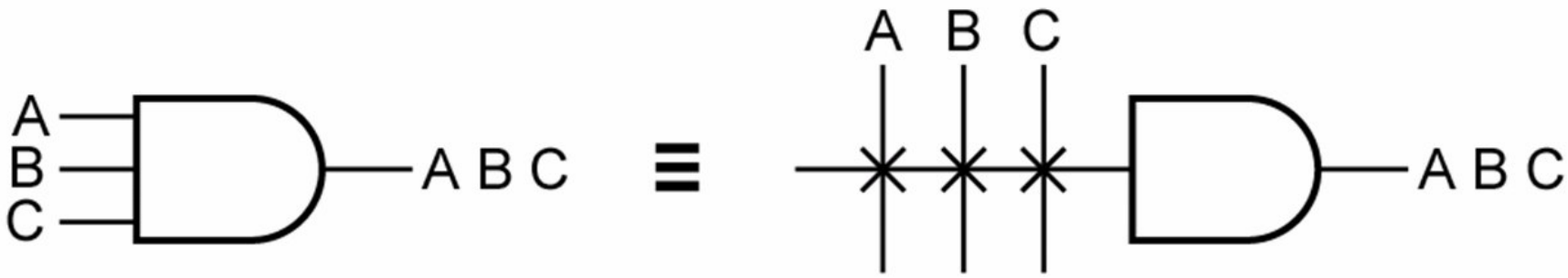
Figure 9-27b: **PLA Realization of Equations**



Buffer logically equivalent to



Section 9.6, p. 245



Section 9.6, p. 246

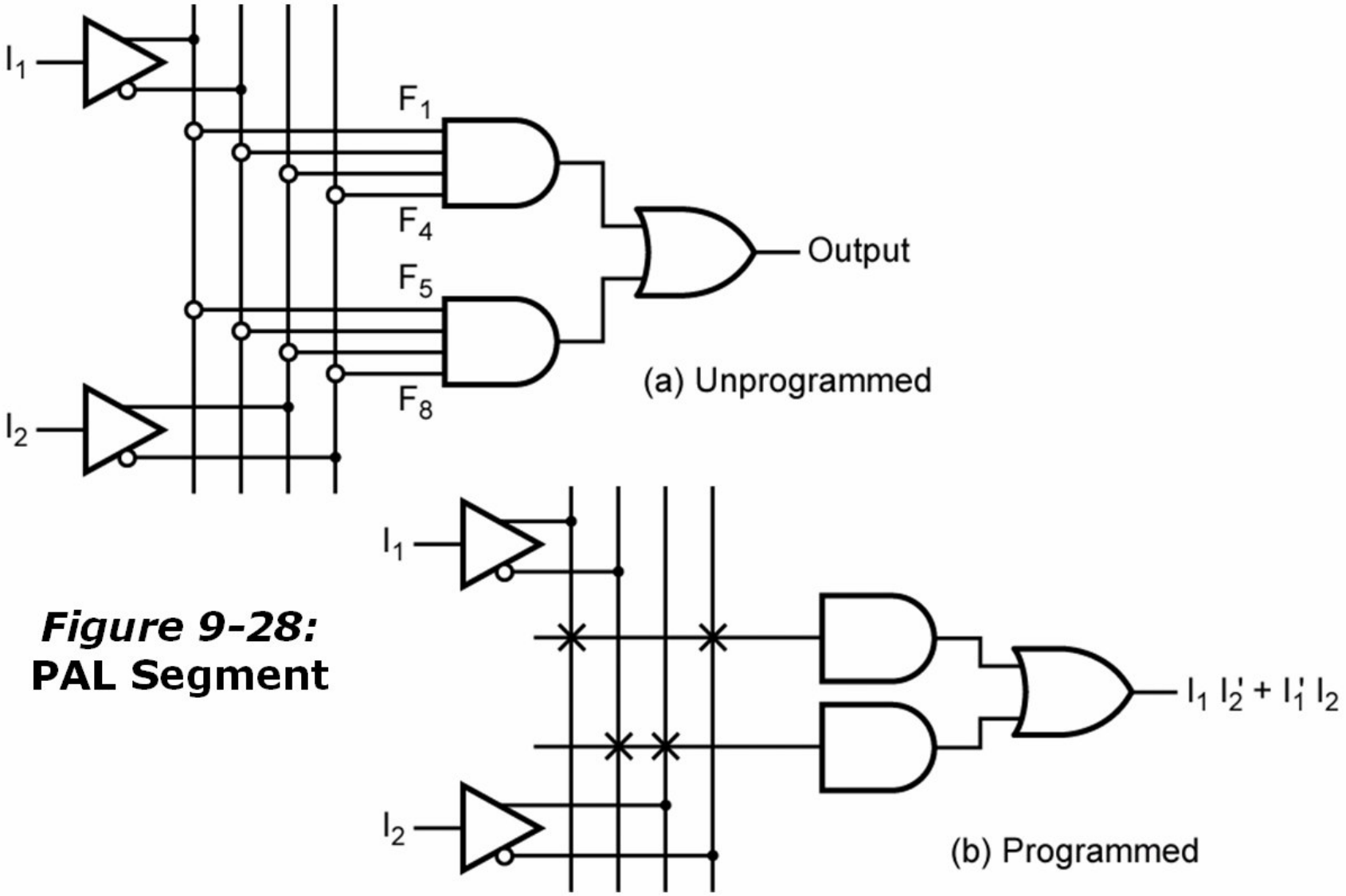


Figure 9-28:
PAL Segment

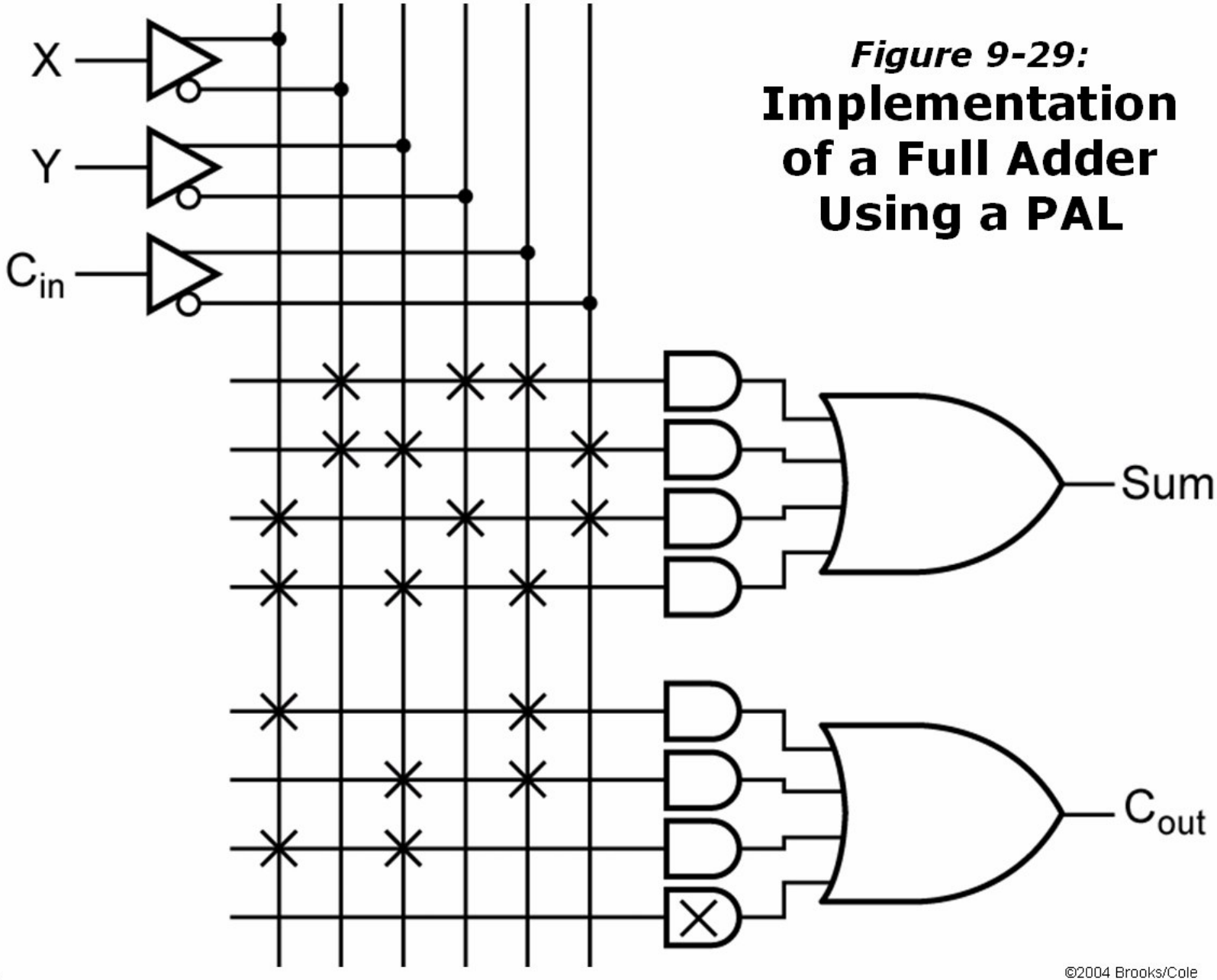


Figure 9-29:
Implementation of a Full Adder Using a PAL

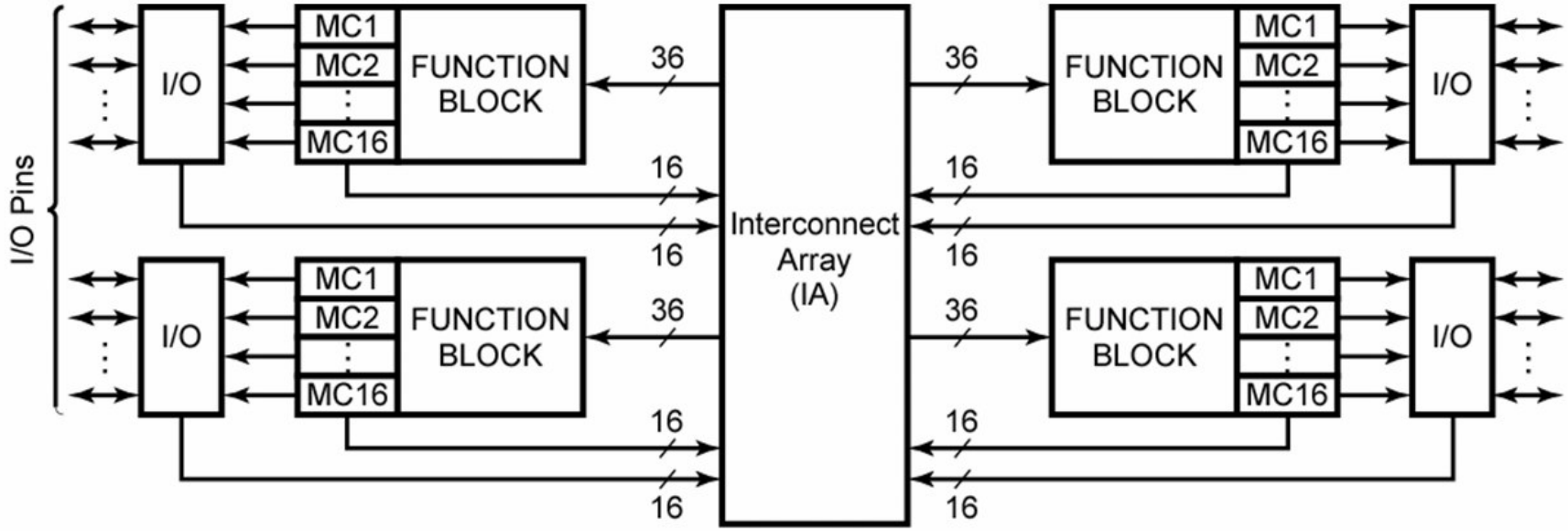


Figure 9-30: **Architecture of Xilinx XCR3064XL CPLD**

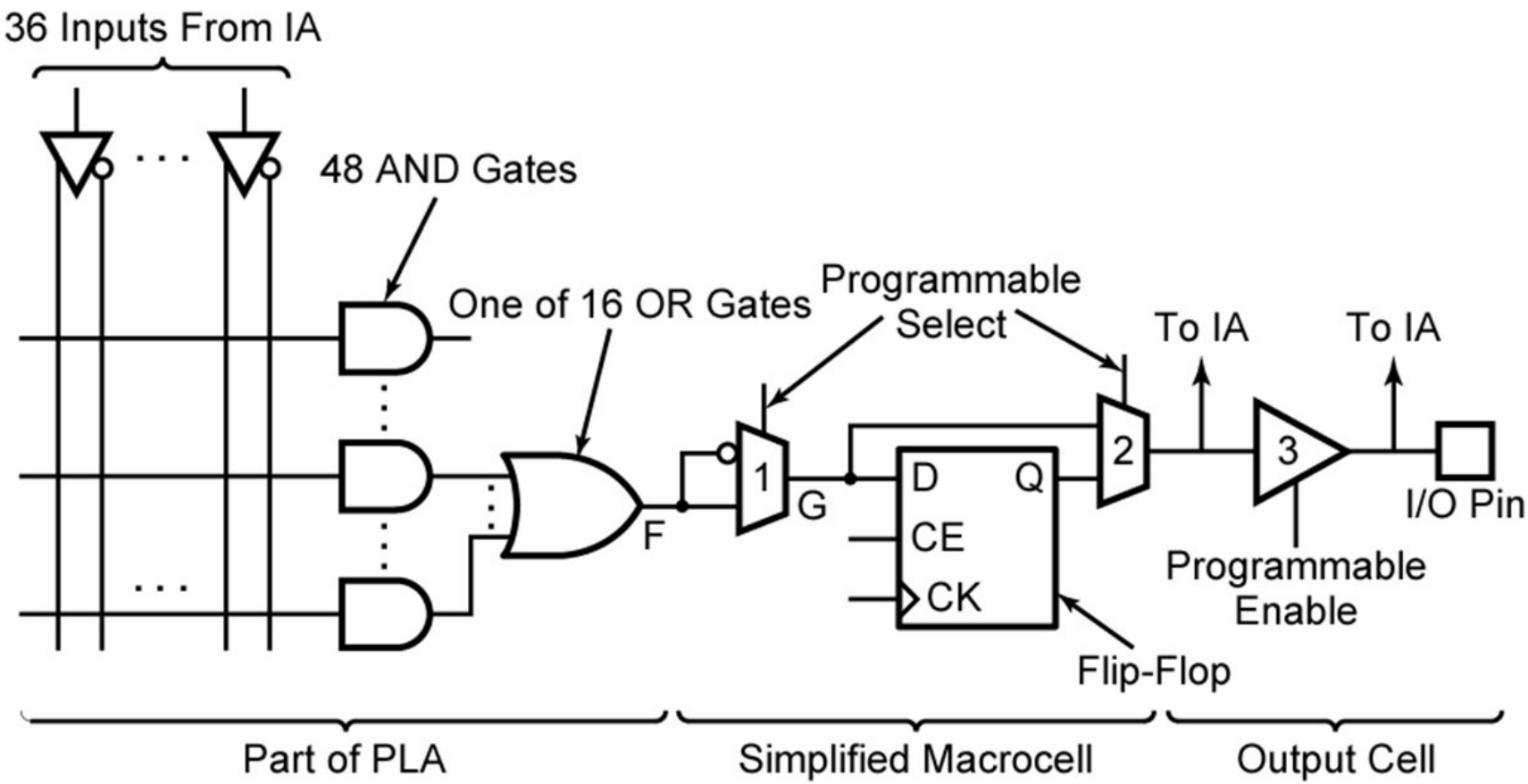


Figure 9-31: **CPLD Function Block and Macrocell**
(A Simplified Version of XCR3064XL)

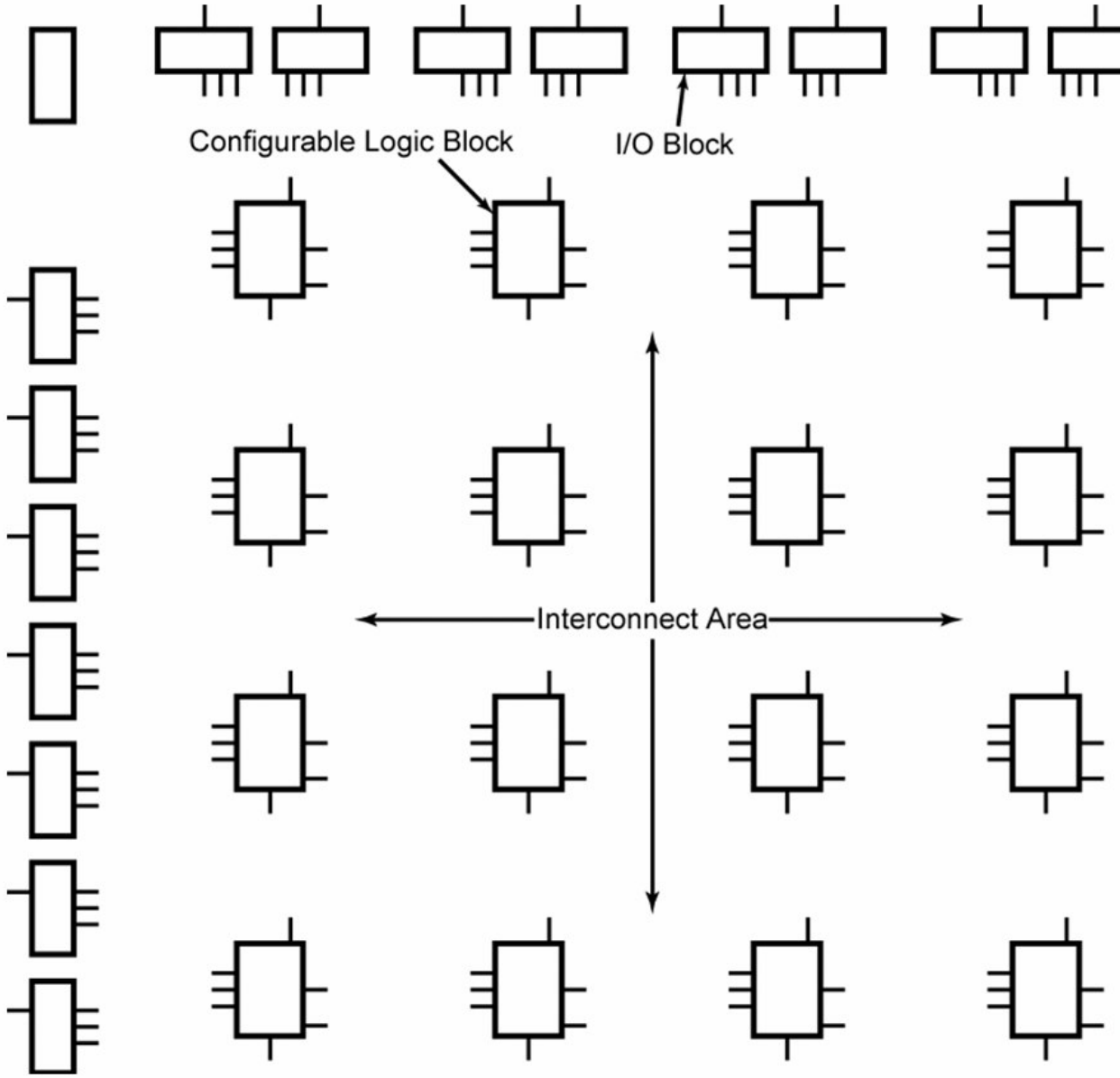


Figure 9-32: **Layout of a Typical FPGA**

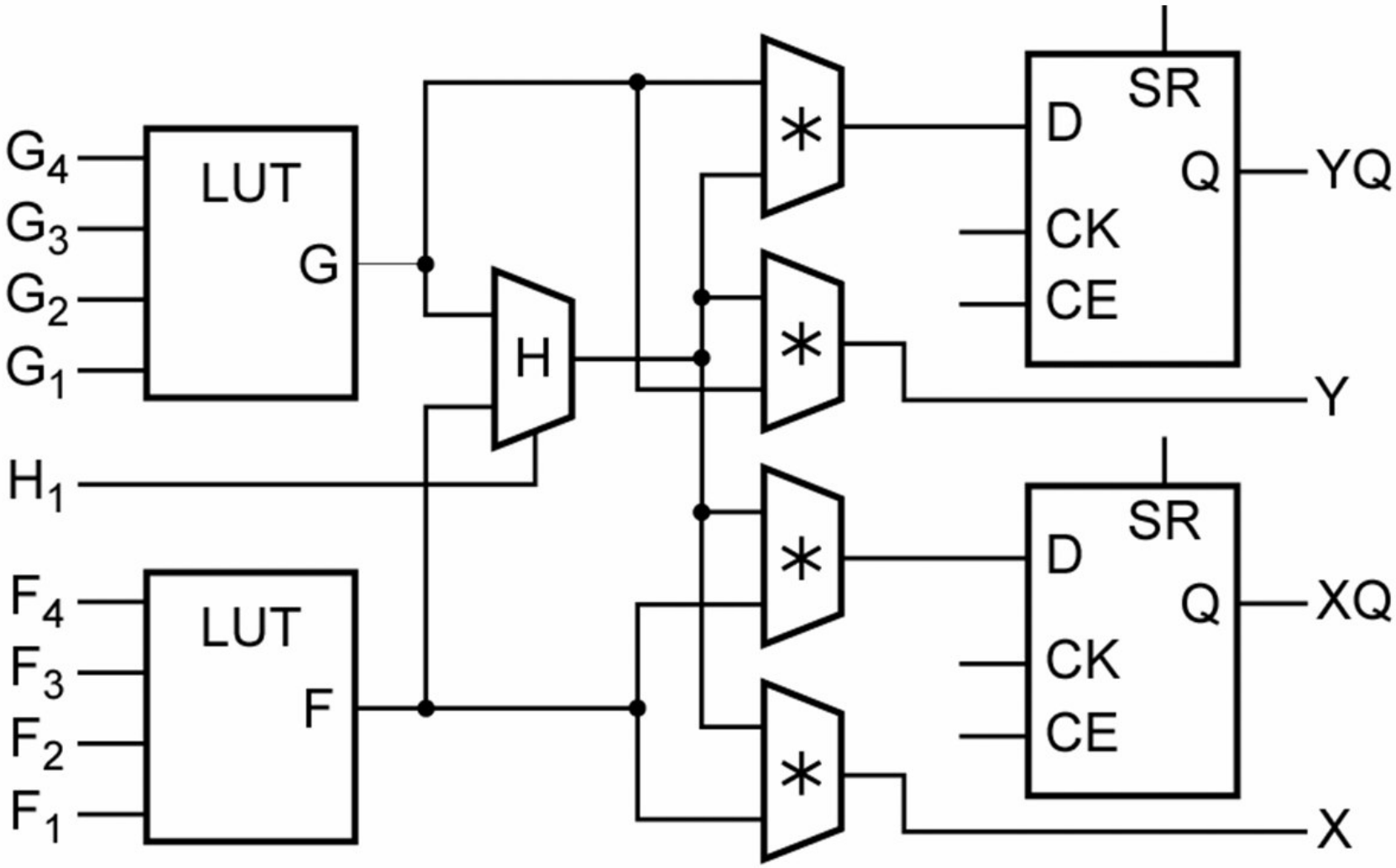


Figure 9-33: **Simplified Configurable Logic Block (CLB)**

abcd	F
0000	0
0001	1
.	.
.	.
.	.
1111	1

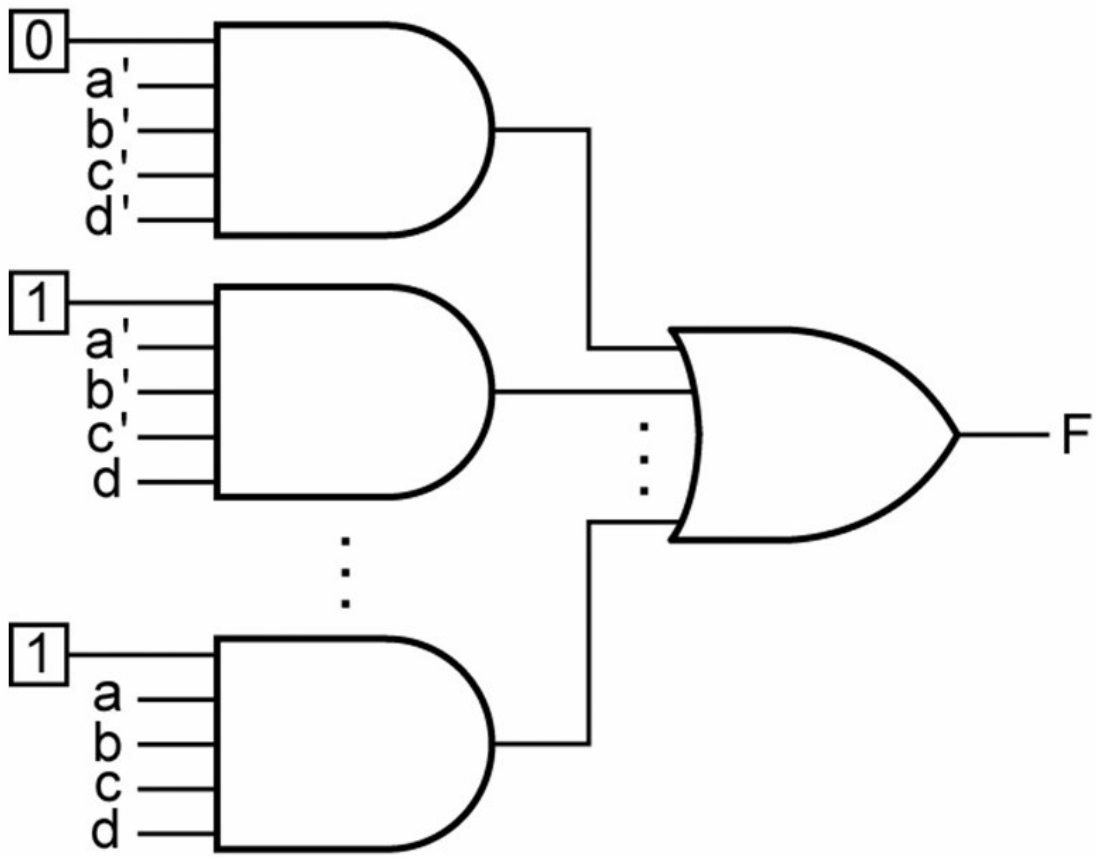


Figure 9-34:

Implementation of a Lookup Table (LUT)

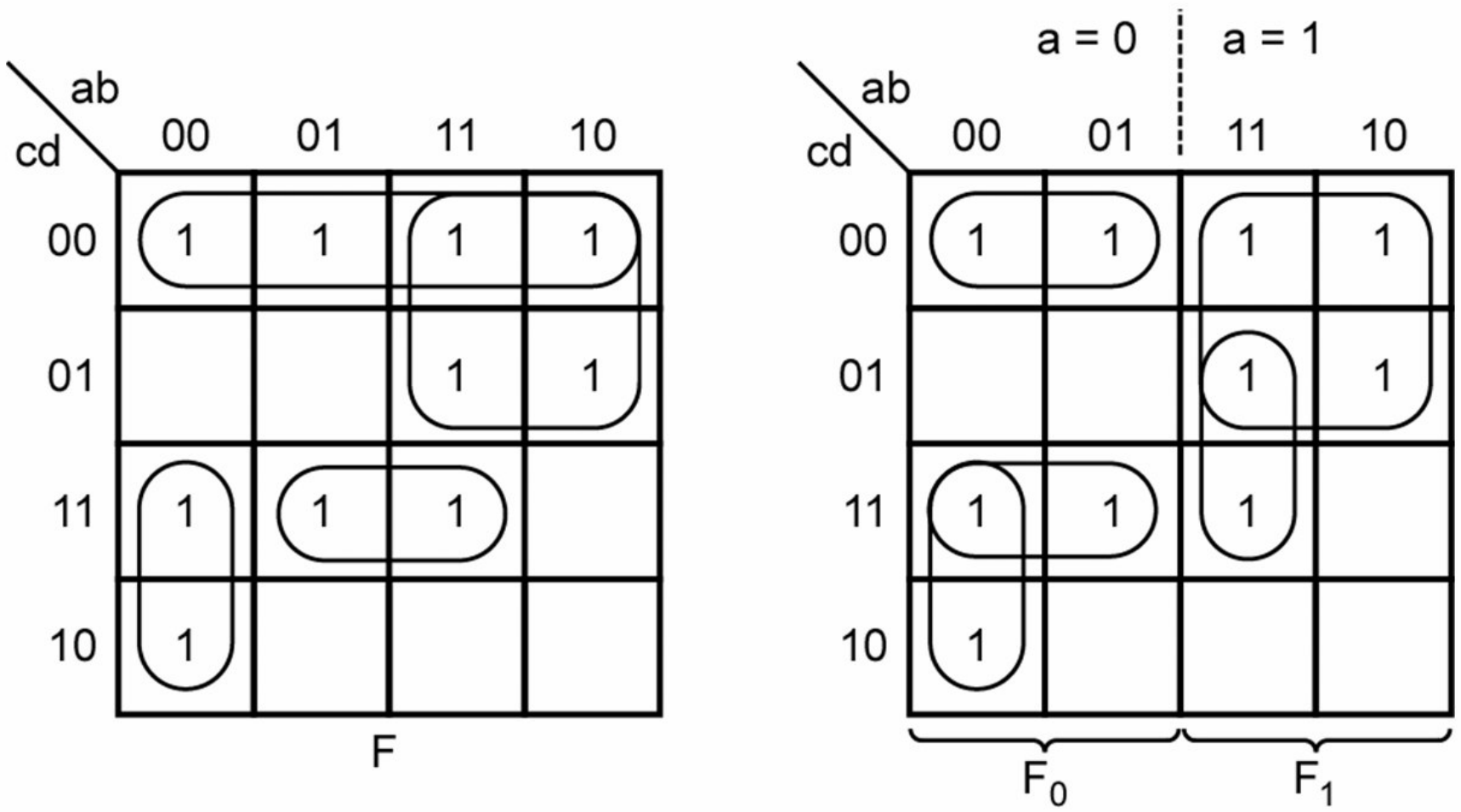


Figure 9-35: **Function Expansion Using a Karnaugh Map**

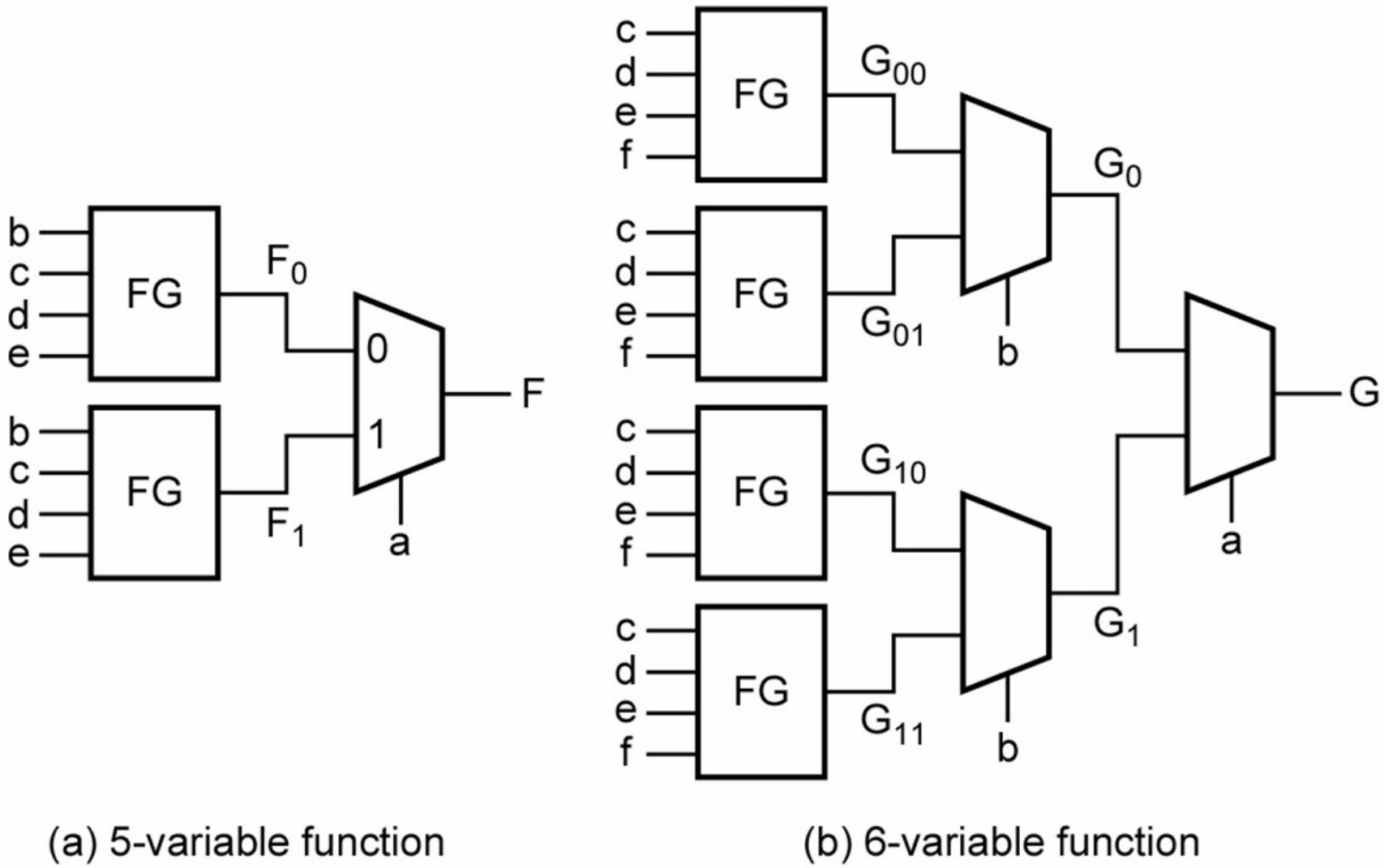


Figure 9-36: **Realization of Five- and Six-Variable Functions with Function Generators**