แก้ไขครั้งล่าสุดเมื่อวันที่ 28 สิงหาคม 2568

Chapter 4

Combinational Logic

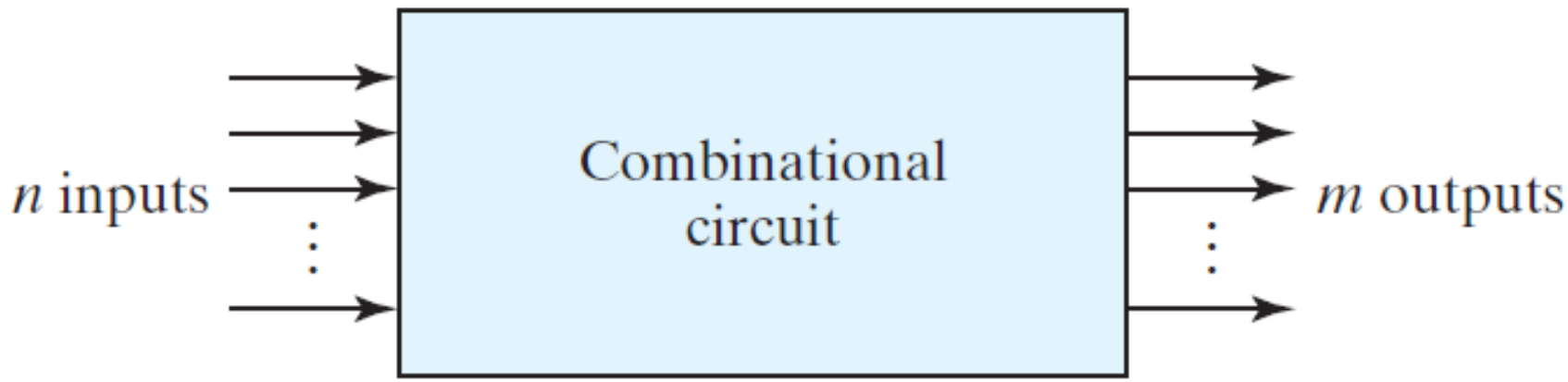


FIGURE 4.1
Block diagram of combinational circuit

Half Adder (ไม่มีตัวทดจากหลักก่อนหน้า)

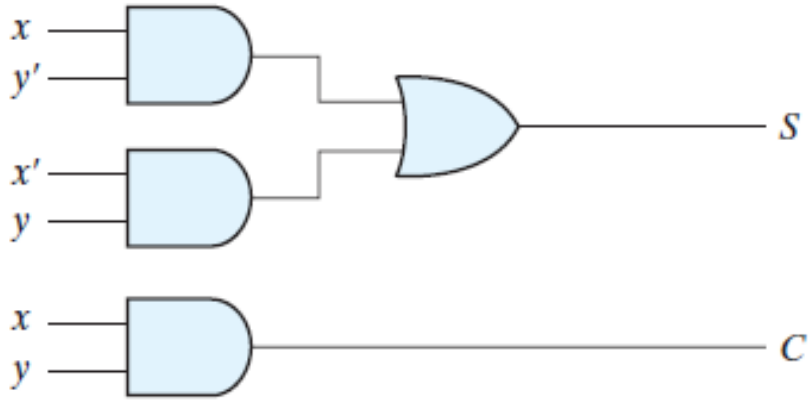
Table 4.3
Half Adder

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

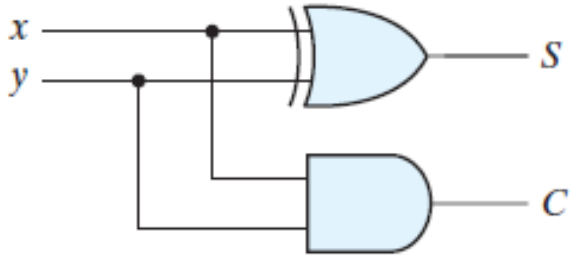
```
  11
 0101  +
 0011
------
 1000
```

Carry In
ตัวทดจากหลักก่อนหน้า

Carry Out
ตัวทดไปหลักถัดไป



(a) $S = xy' + x'y$
$C = xy$



(b) $S = x \oplus y$
$C = xy$

FIGURE 4.5
Implementation of half adder

Full Adder (มีตัวทดจากหลักก่อนหน้า)

Table 4.4
Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

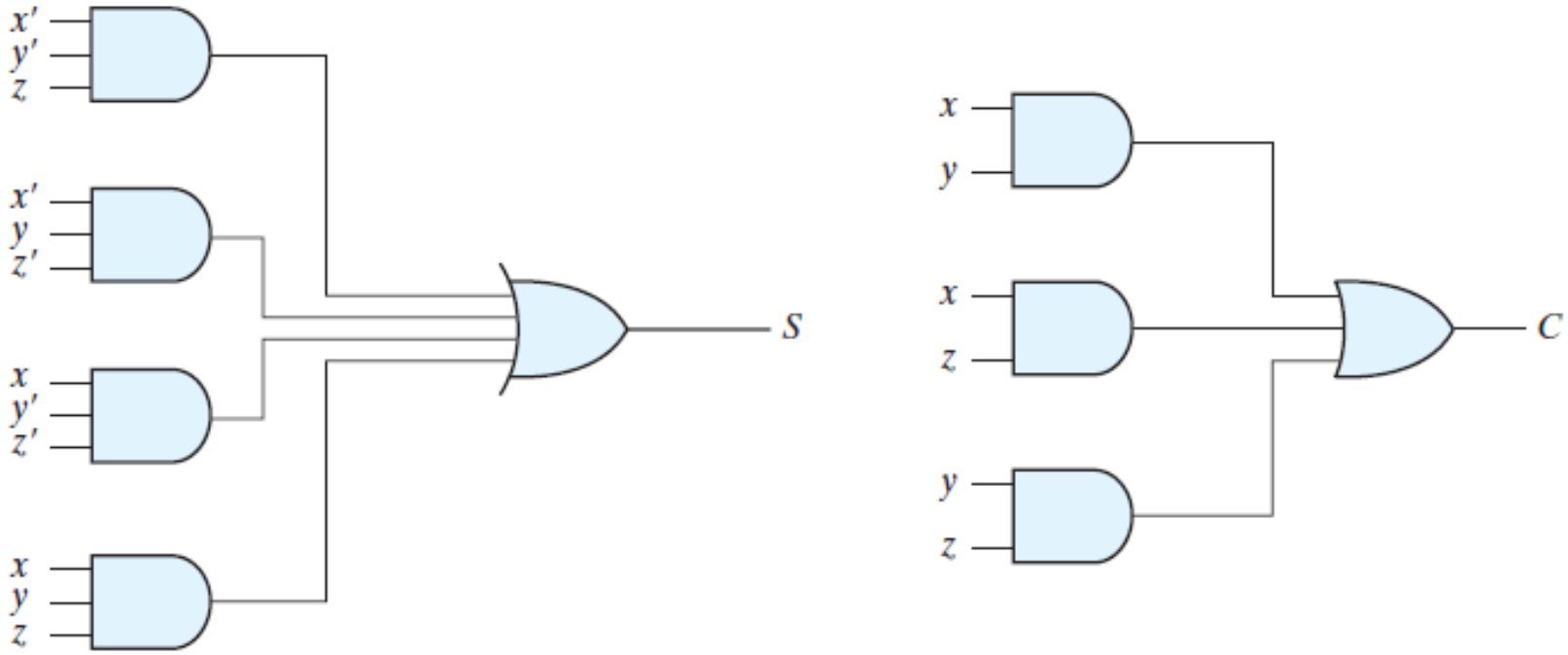


FIGURE 4.7
Implementation of full adder in sum-of-products form

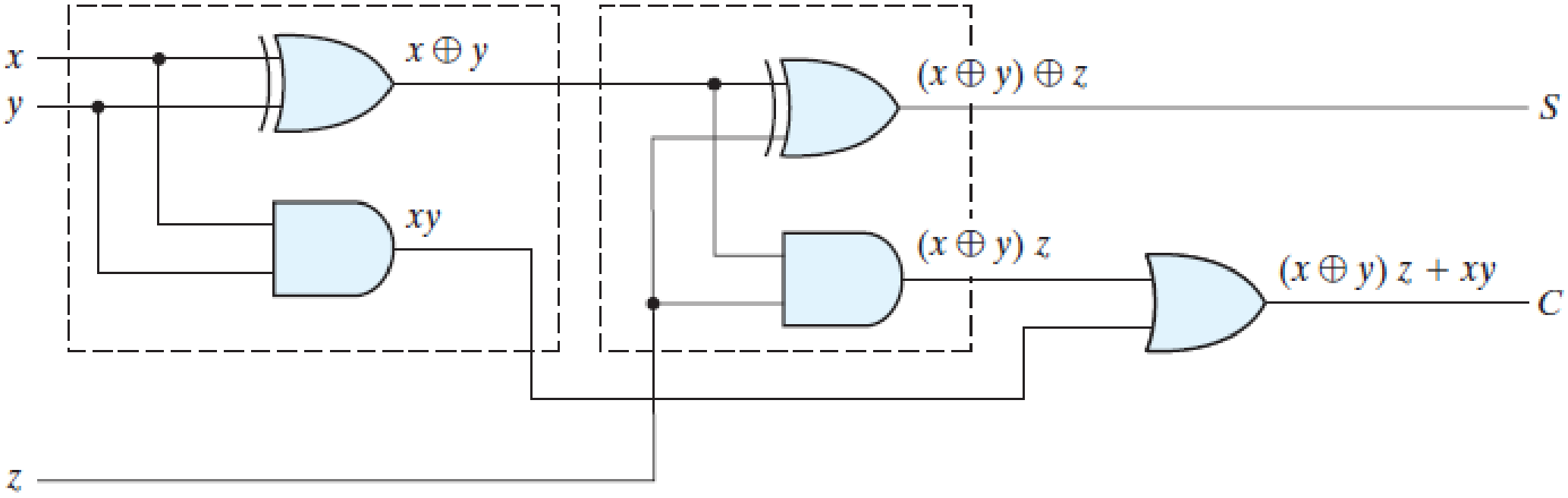


FIGURE 4.8
Implementation of full adder with two half adders and an OR gate

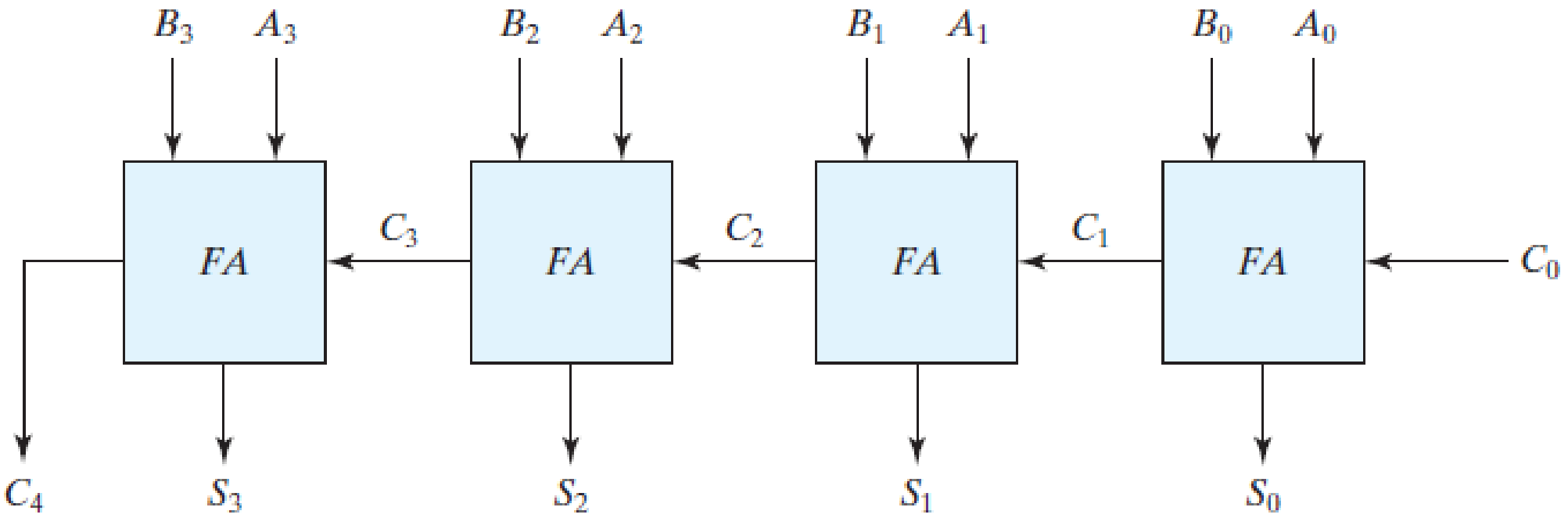


FIGURE 4.9
Four-bit adder

Most Significant Bit (MSB)

Least Significant Bit (LSB)

นิยมเขียนสองแบบ $A = A_3A_2A_1A_0$

$A = A_8A_4A_2A_1$

หลัก 1, 2, 4, 8 ในฐานสอง ก็เหมือน
หลักหน่วย สิบ ร้อย พัน ในฐานสิบ

Carry Propagation

P คือ Carry Propagate
G คือ Carry Generate

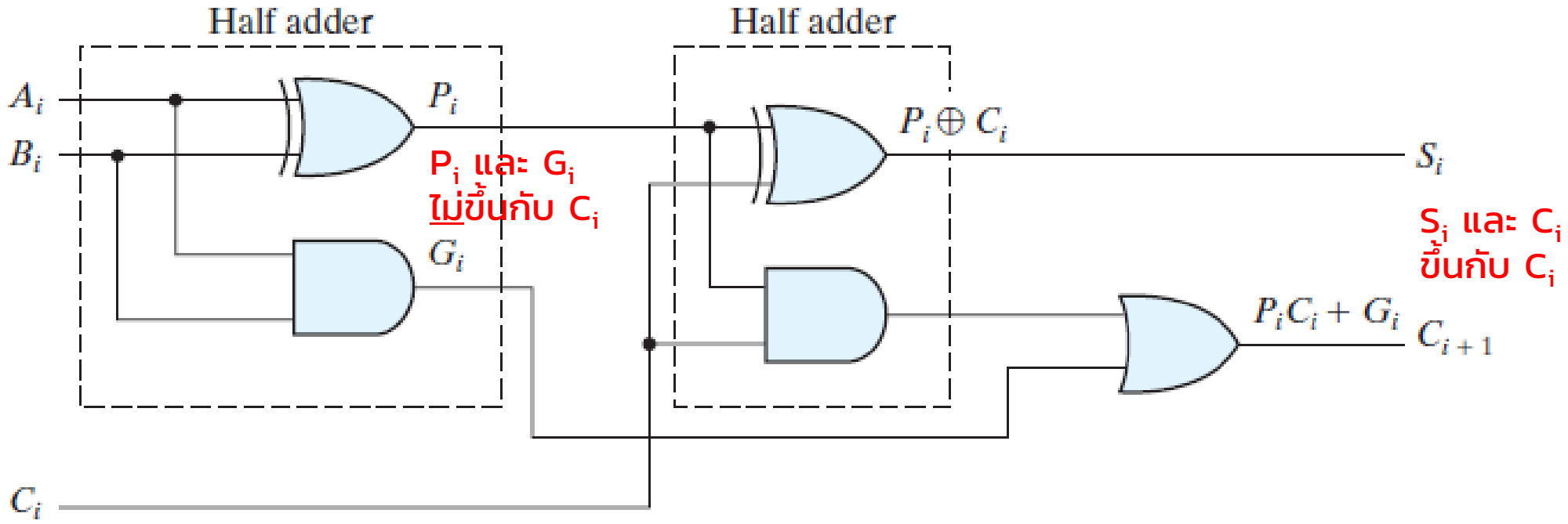


FIGURE 4.10
Full adder with *P* and *G* shown

Gate Delay มาก

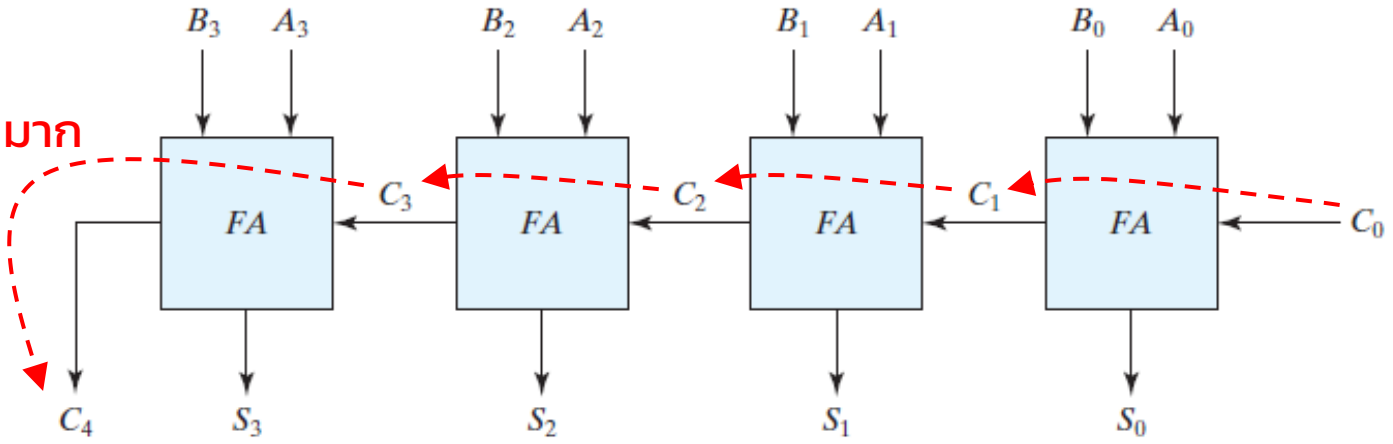


FIGURE 4.9
Four-bit adder

$$P_i = A_i \oplus B_i \qquad S_i = P_i \oplus C_i$$

$$G_i = A_i B_i \qquad C_{i+1} = G_i + P_i C_i$$

$C_0 =$ input carry

$C_1 = G_0 + P_0 C_0$

เขียน C_i ให้เป็น sum of products ให้หมด (ดีเลย์ = 2 เกต)
วงจรทำงานได้เร็ว แต่เปลืองจำนวนเกต

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

เป็นตัวอย่างการออกแบบที่ใช้ resource (จำนวน gate) ไปแลกกับความเร็วในการคำนวณ ได้ gate delay คงที่ ไม่ว่าจะบวกเลขกี่บิต

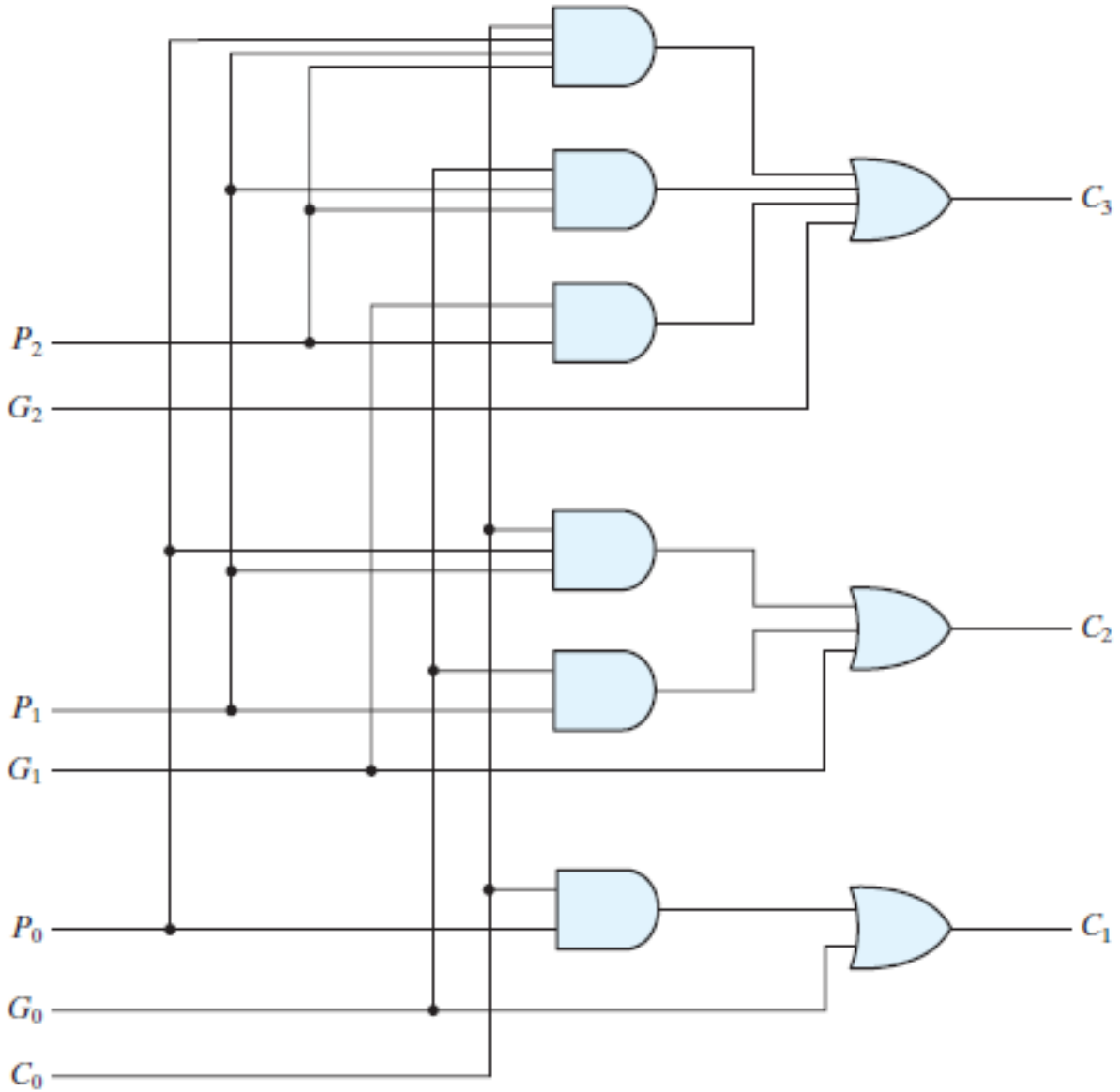


FIGURE 4.11
Logic diagram of carry lookahead generator

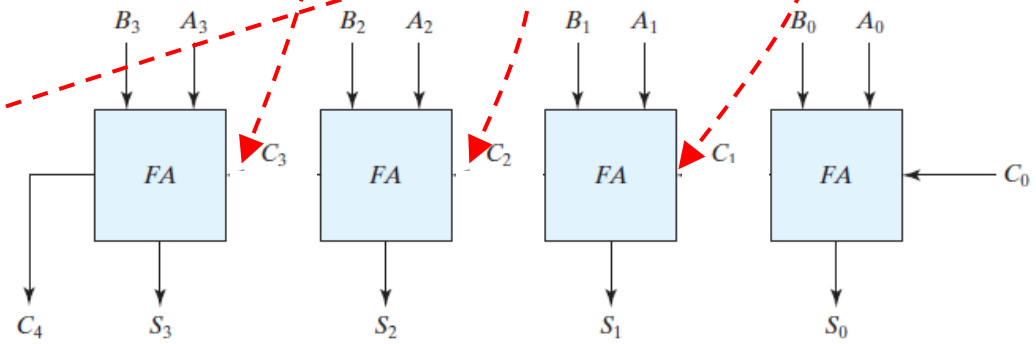
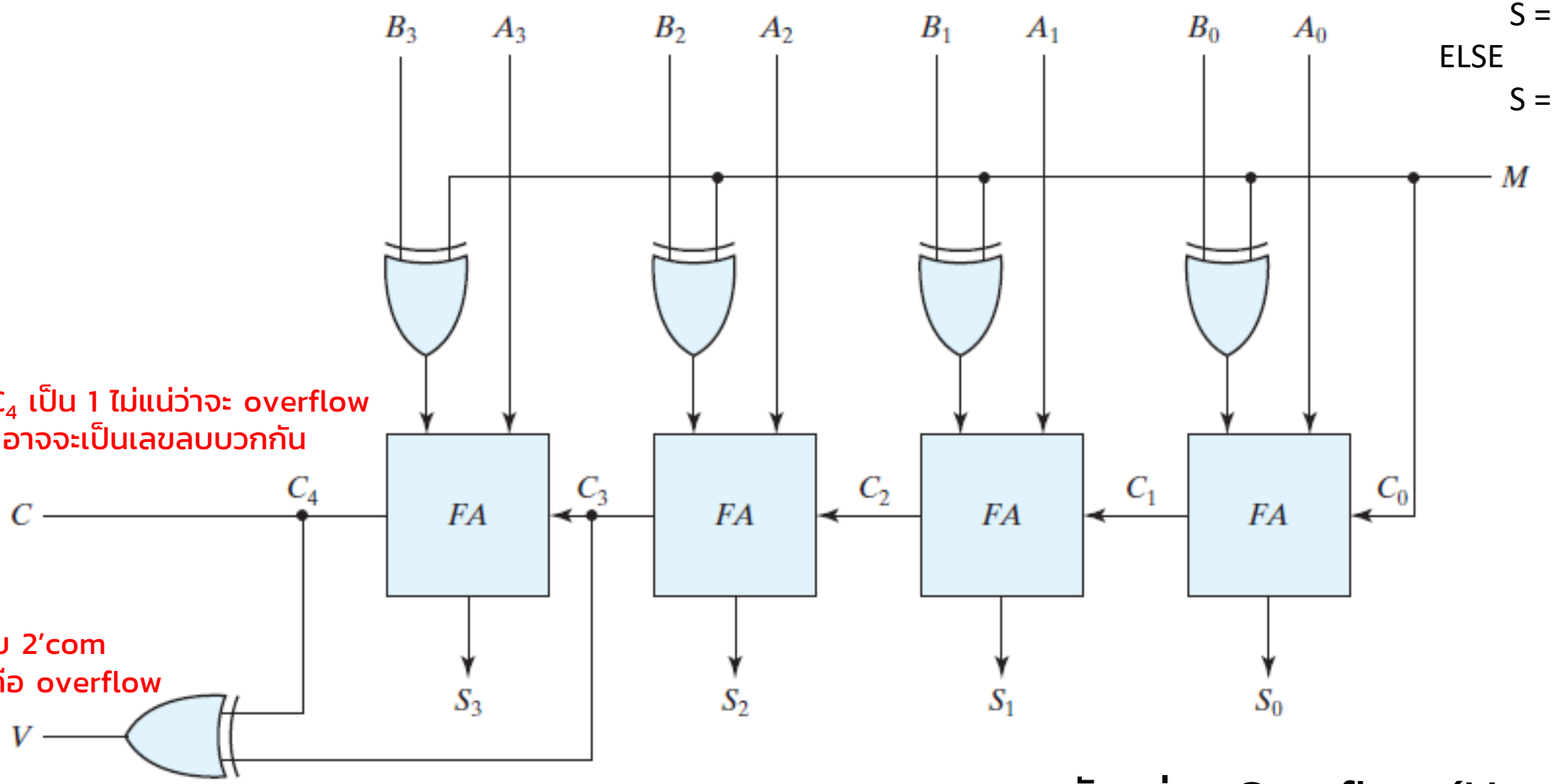


FIGURE 4.9
Four-bit adder

IF M == 0:
S = A + B
ELSE
S = A – B



แค่ C_4 เป็น 1 ไม่แน่ว่าจะ overflow
อาจจะเป็นเลขลบบวกกัน

ในระบบ 2'com
ถ้า V = 1 คือ overflow

FIGURE 4.13
Four-bit adder–subtractor (with overflow detection)

<u>ตัวอย่าง Overflow (V = 1)</u>

0110 (6) + 0100 (4) = 01010

1011 (–5) + 1011 (–5) = 10110

Overflow

The binary adder–subtractor circuit with outputs C and V is shown in Fig. 4.13. If the two binary numbers are considered to be unsigned, then the C bit detects a carry after addition or a borrow after subtraction. If the numbers are considered to be signed, then the V bit detects an overflow. If $V = 0$ after an addition or subtraction, then no overflow occurred and the n-bit result is correct. If $V = 1$, then the result of the operation contains $n + 1$ bits, but only the rightmost n bits of the number fit in the space available, so an overflow has occurred. The $(n + 1)$ th bit is the actual sign and has been shifted out of position.

BCD Adder

Table 4.5
Derivation of BCD Adder

จะใช้วงจรบวก Binary มาบวก BCD

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

ไม่ต้องทำอะไร
Binary = BCD

ที่วงไว้คือ
กรณีที่
ต้องแก้
โดย +6

แก้กลับไป

+6

เช่น 5 + 5, 7 + 3 ได้ 0 ทด 1

เช่น 9 + 9 กับตัวทดอีก 1 ได้ 9 ทด 1

$$C = K + Z_8Z_4 + Z_8Z_2$$

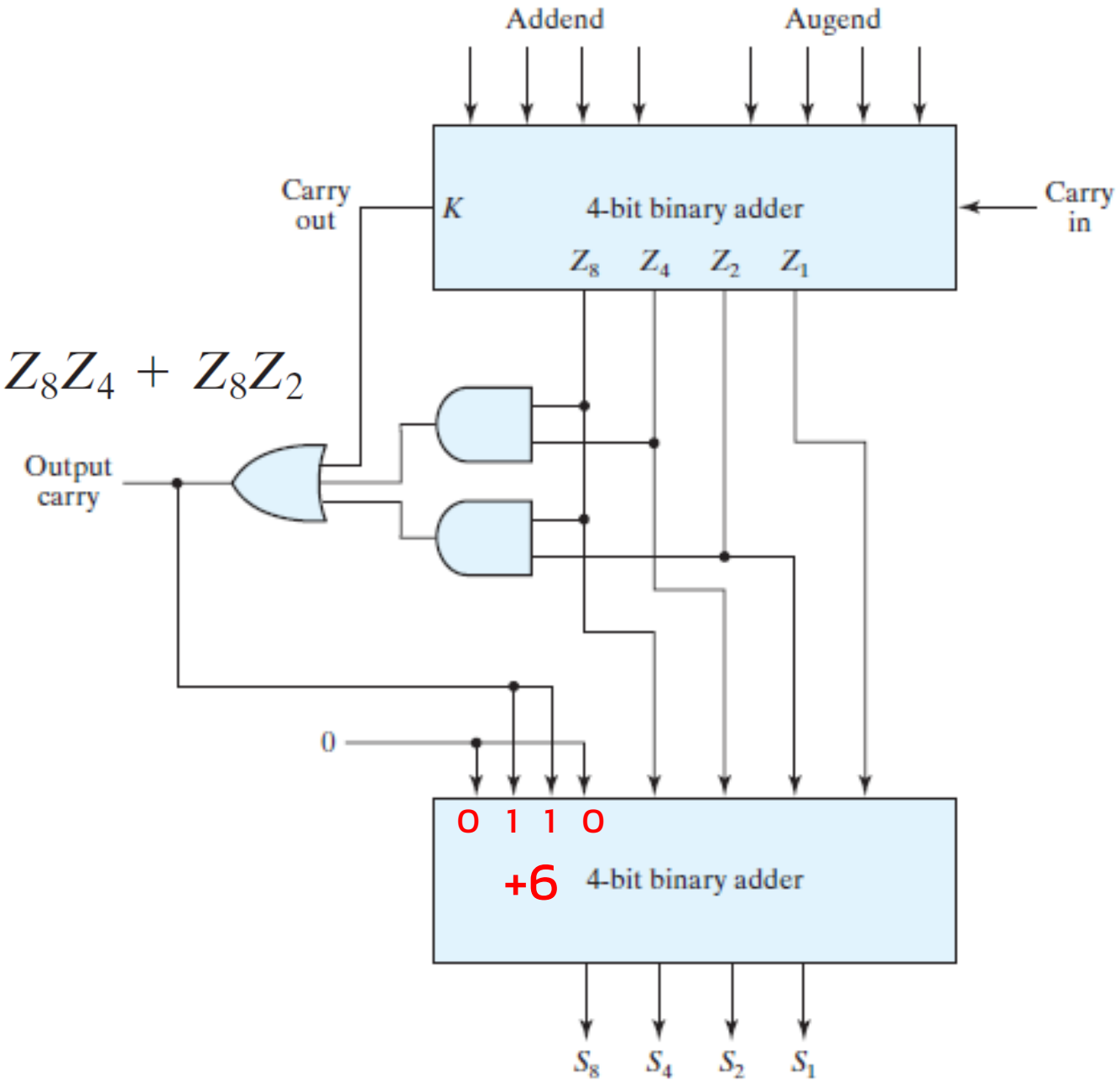
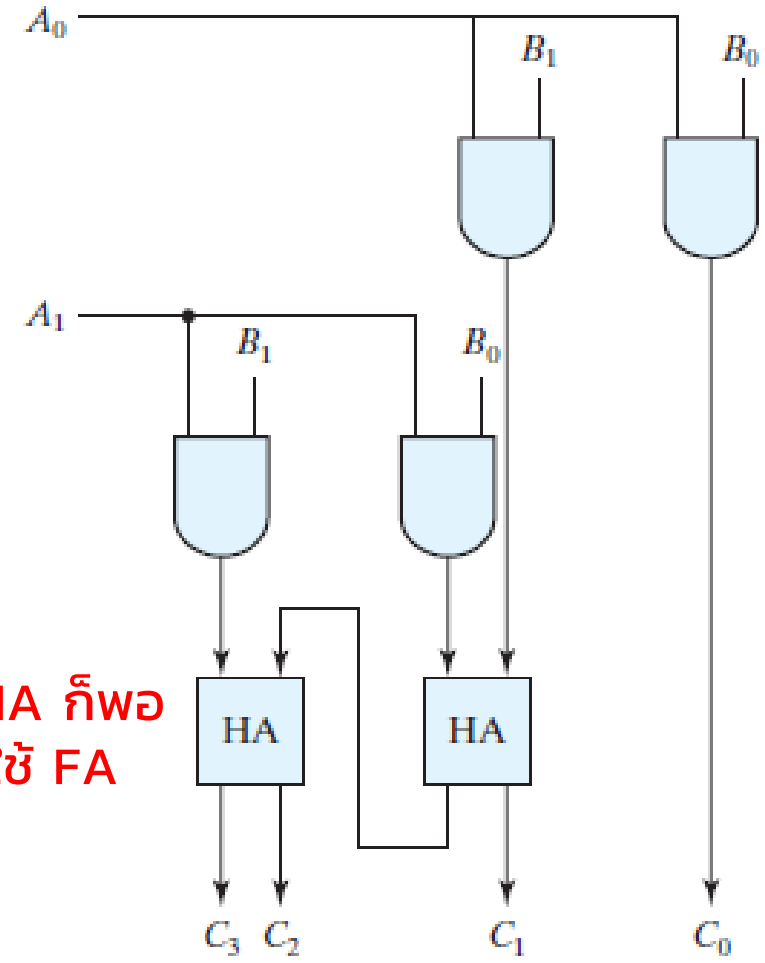


FIGURE 4.14
Block diagram of a BCD adder

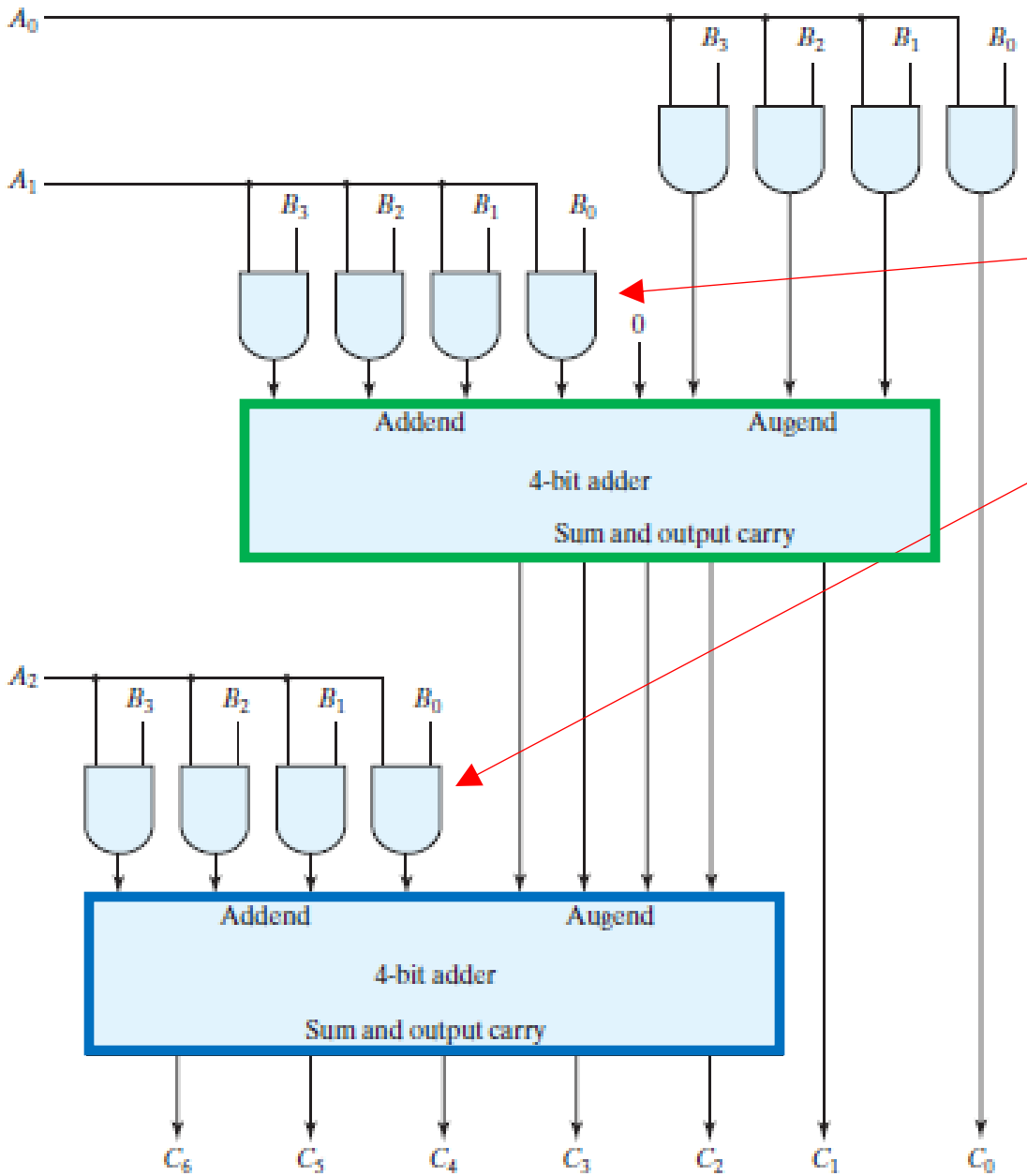
Binary Multiplier

$$
\begin{array}{cccc}
 & & B_1 & B_0 \\
 & & A_1 & A_0 \\
\hline
 & & A_0B_1 & A_0B_0 \\
 & A_1B_1 & A_1B_0 & \\
\hline
C_3 & C_2 & C_1 & C_0
\end{array}
$$



ใช้แค่ HA ก็พอ
ไม่ต้องใช้ FA

FIGURE 4.15
Two-bit by two-bit binary multiplier



			B_3	B_2	B_1	B_0
				A_2	A_1	A_0
			A_0B_3	A_0B_2	A_0B_1	A_0B_0
		A_1B_3	A_1B_2	A_1B_1	A_1B_0	
	A_2B_3	A_2B_2	A_2B_1	A_2B_0		
C_6	C_5	C_4	C_3	C_2	C_1	C_0

FIGURE 4.16
Four-bit by three-bit binary multiplier

Comparator (unsigned)

$$A = A_3\, A_2\, A_1\, A_0$$

$$B = B_3\, B_2\, B_1\, B_0$$

$$x_i = A_i B_i + A_i' B_i' \quad \text{for } i = 0, 1, 2, 3$$

$$(A = B) = x_3 x_2 x_1 x_0$$

ฐานสิบ	2'com (unsigned)
+3	011 (3)
+2	010 (2)
+1	001 (1)
0	000 (0)
-1	111 (7)
-2	110 (6)
-3	101 (5)
-4	100 (4)

$$(A > B) = A_3 B_3' + x_3 A_2 B_2' + x_3 x_2 A_1 B_1' + x_3 x_2 x_1 A_0 B_0'$$

$$(A < B) = A_3' B_3 + x_3 A_2' B_2 + x_3 x_2 A_1' B_1 + x_3 x_2 x_1 A_0' B_0$$

Textbook เขียนผิด

วิธีนี้ใช้ได้กับ unsigned integer เท่านั้น !!!
ใช้กับ 2'com ไม่ได้

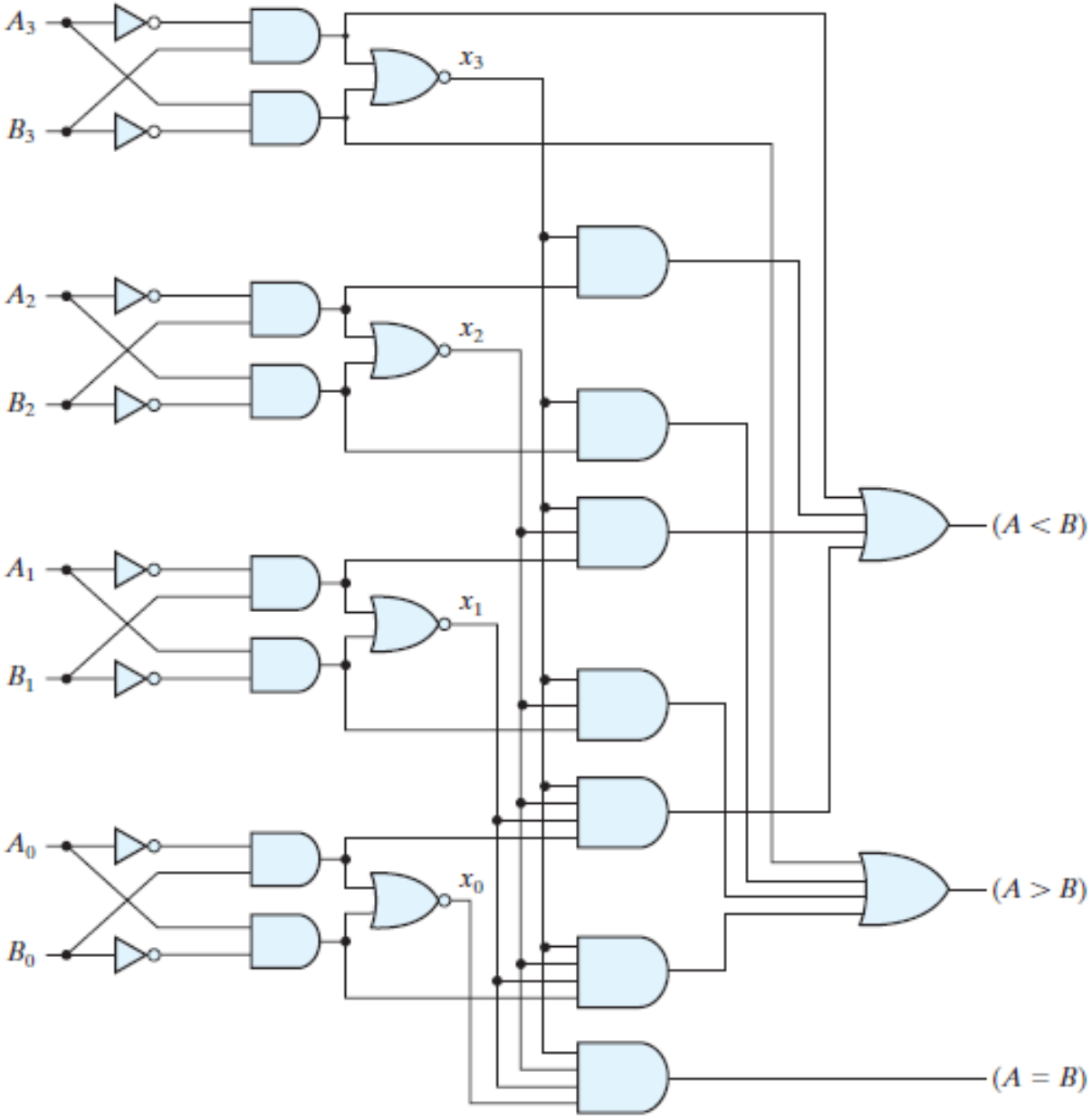
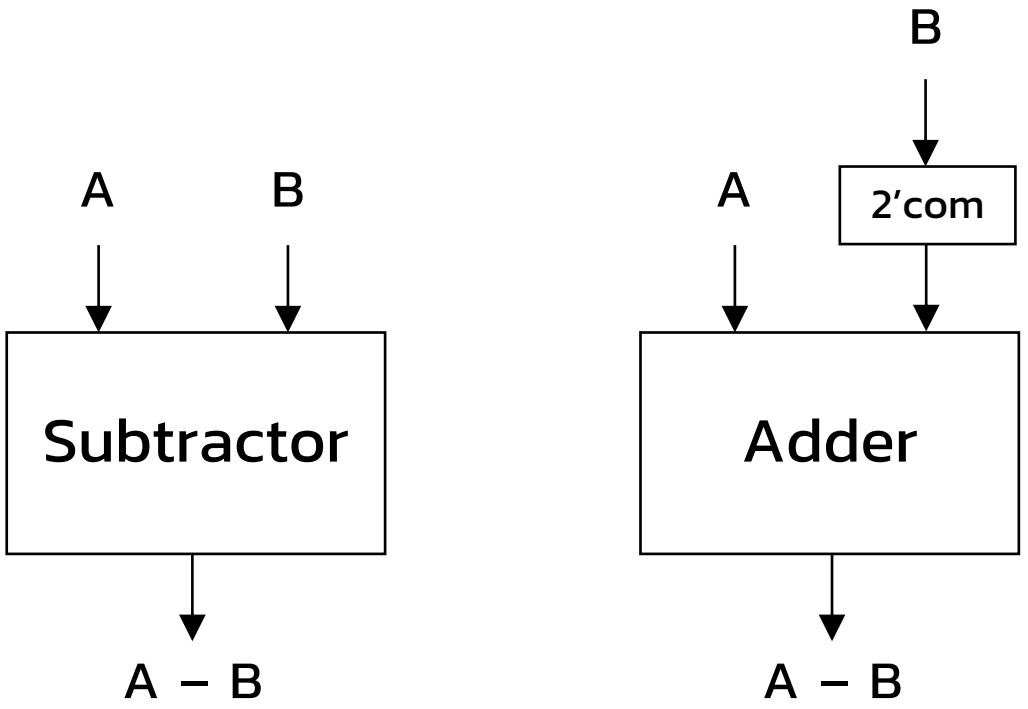


FIGURE 4.17
Four-bit magnitude comparator

วิธีนี้ใช้ได้กับ unsigned integer เท่านั้น !!!
ใช้กับ 2'com ไม่ได้

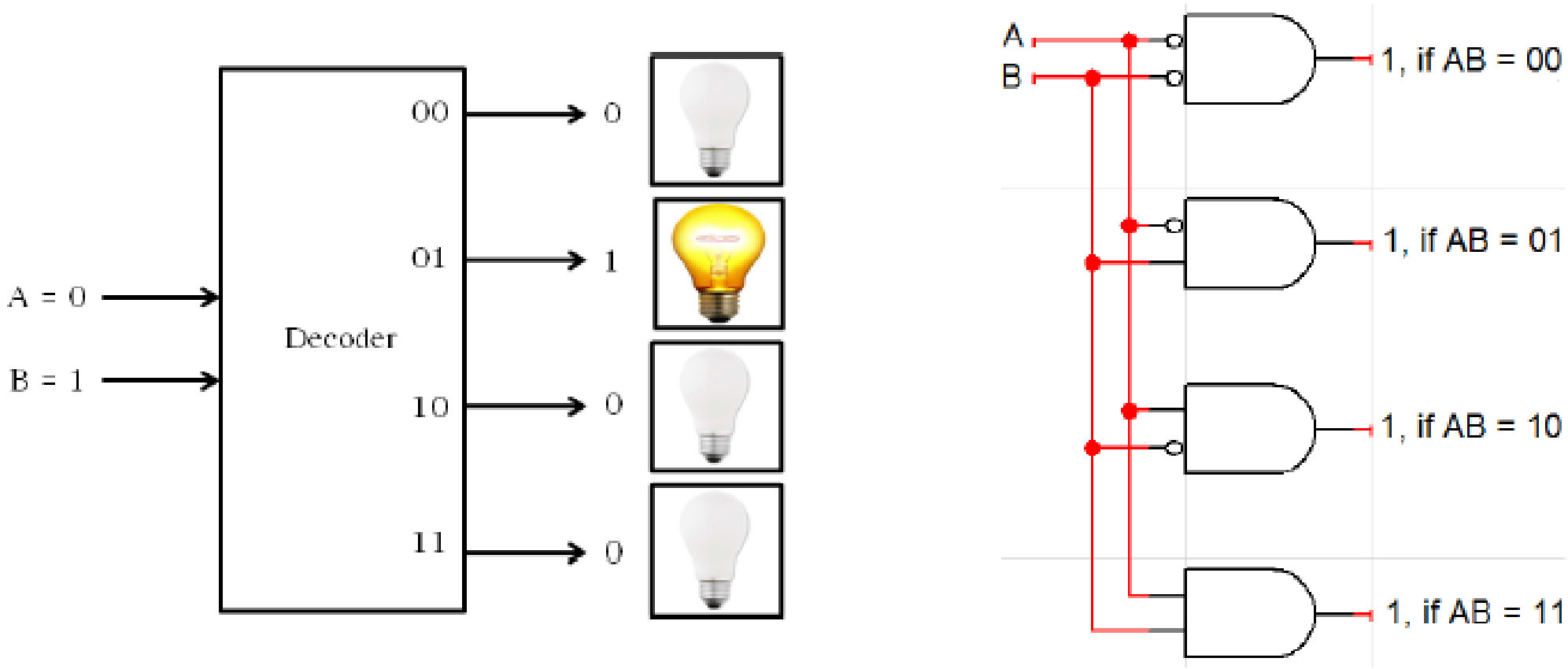
Comparator (2'com)

A และ B เป็น signed integer (ระบบ 2'com)
A และ B อาจจะมีค่าเป็นบวกหรือลบก็ได้



MSB = 0	บิตที่เหลือ = 0	สรุปว่า A = B
	บิตที่เหลือ ≠ 0	สรุปว่า A > 0
MSB = 1	A < B	

Decoder



รูปที่ 3.6 ตัวถอดรหัส 2-to-4

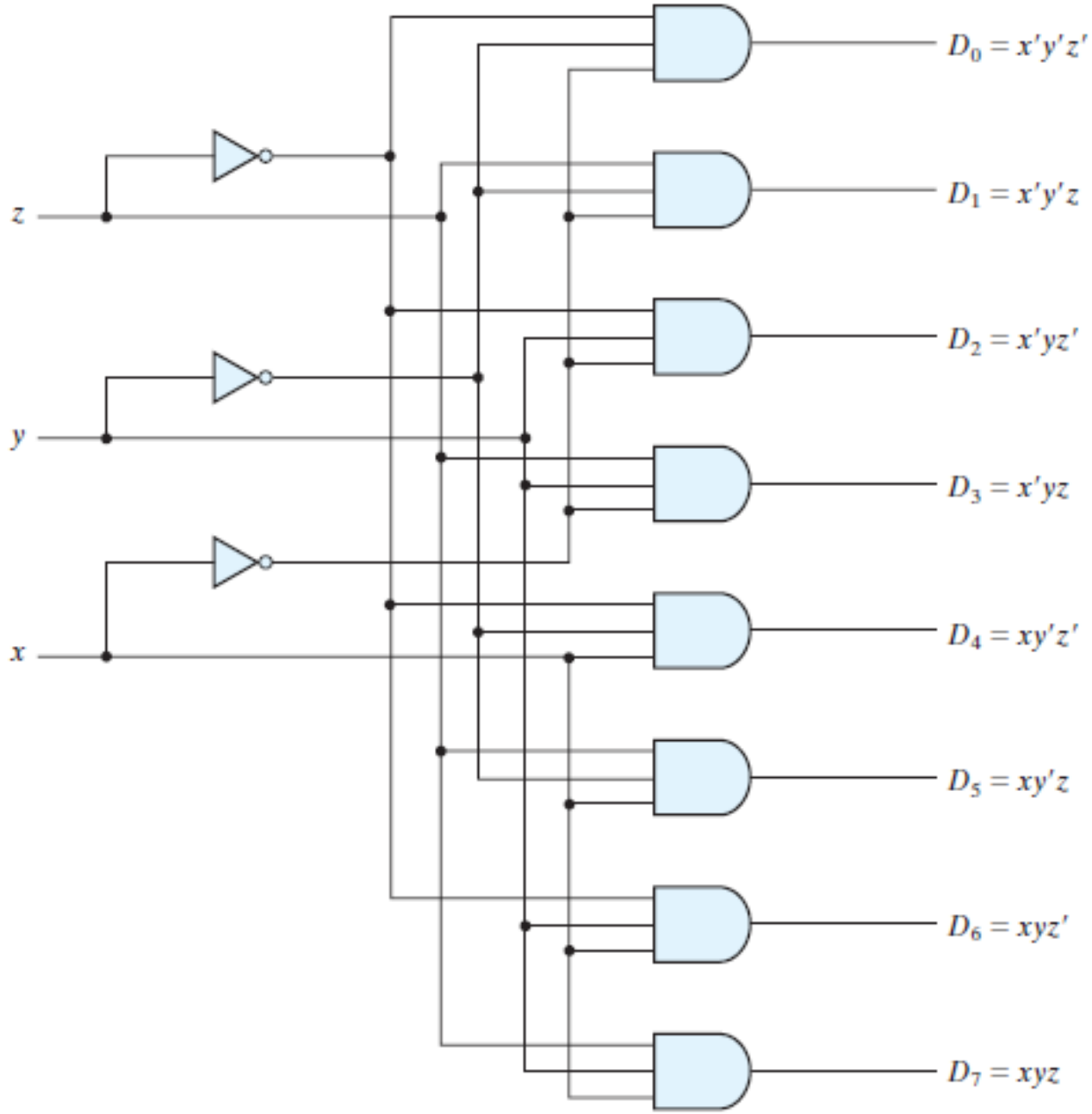
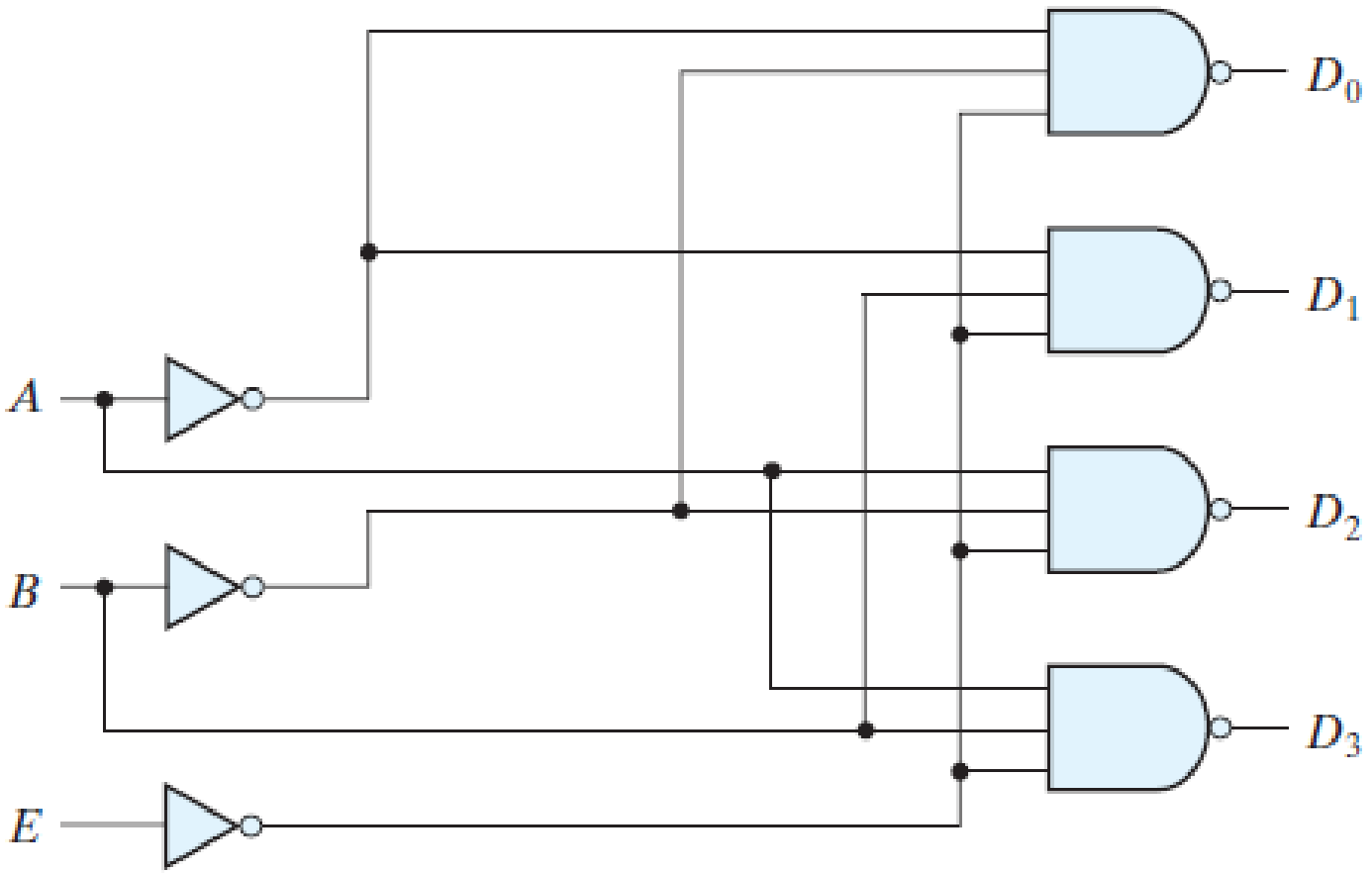


FIGURE 4.18
Three-to-eight-line decoder



(a) Logic diagram

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0 แก้

(b) Truth table

D_i = 0 คือให้หลอดไฟติด
E = 0 คือให้ทำงาน

FIGURE 4.19
Two-to-four-line decoder with enable input (E)

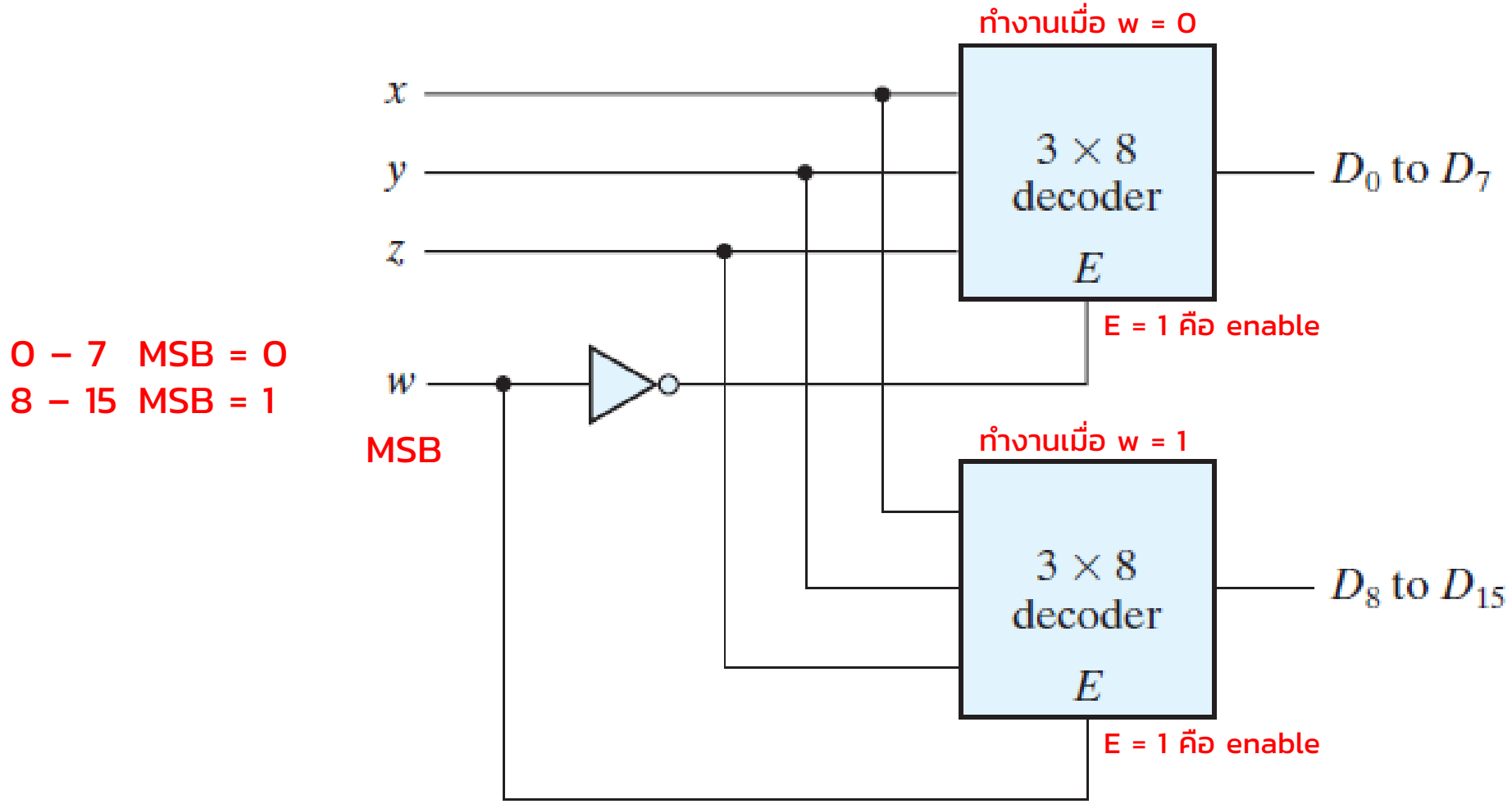


FIGURE 4.20
4 × 16 decoder constructed with two 3 × 8 decoders

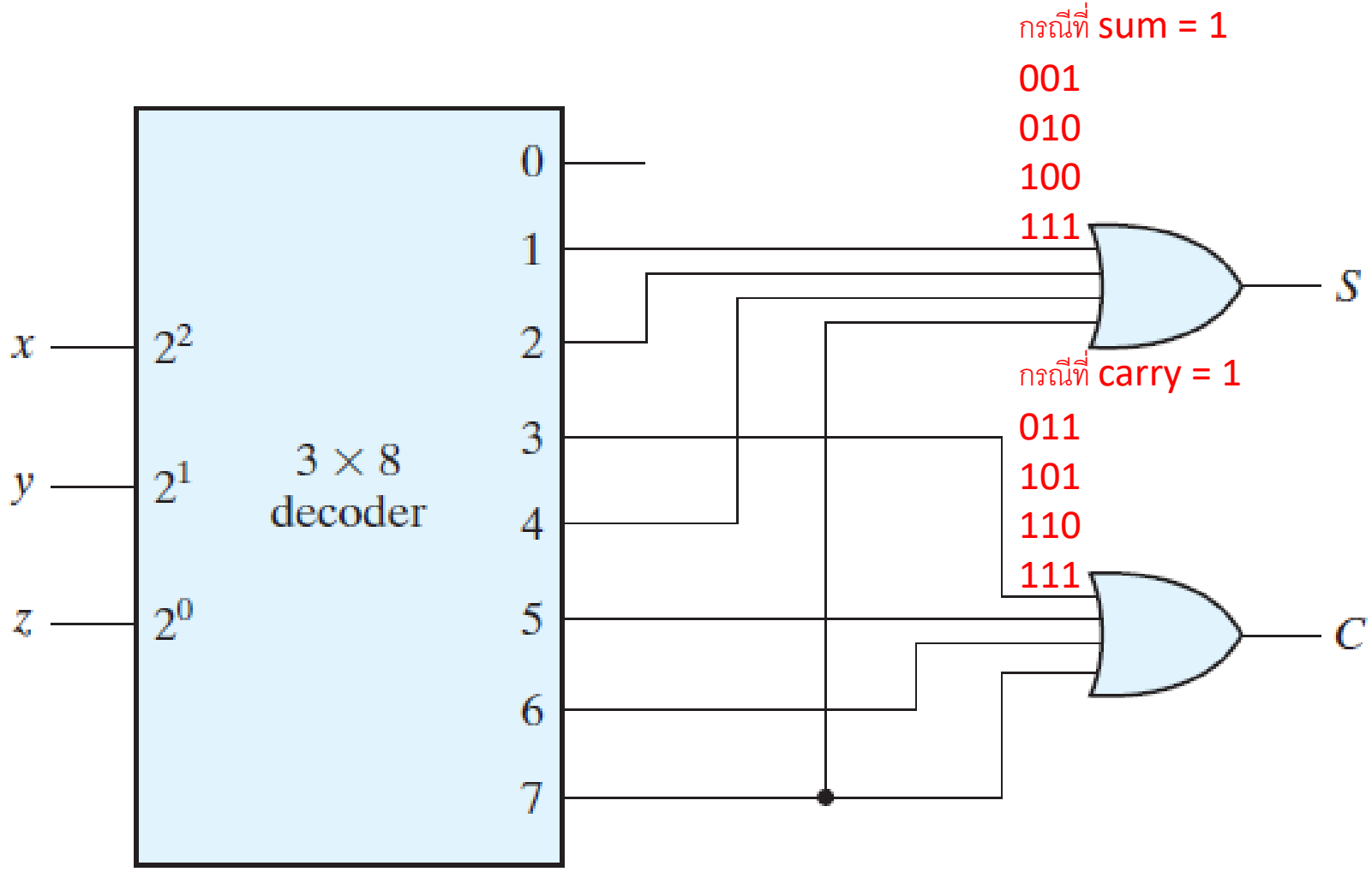


FIGURE 4.21
Implementation of a full adder with a decoder

Encoder (Inverse function ของ Decoder)

Table 4.7
Truth Table of an Octal-to-Binary Encoder

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

$$z = D_1 + D_3 + D_5 + D_7$$
$$y = D_2 + D_3 + D_6 + D_7$$
$$x = D_4 + D_5 + D_6 + D_7$$

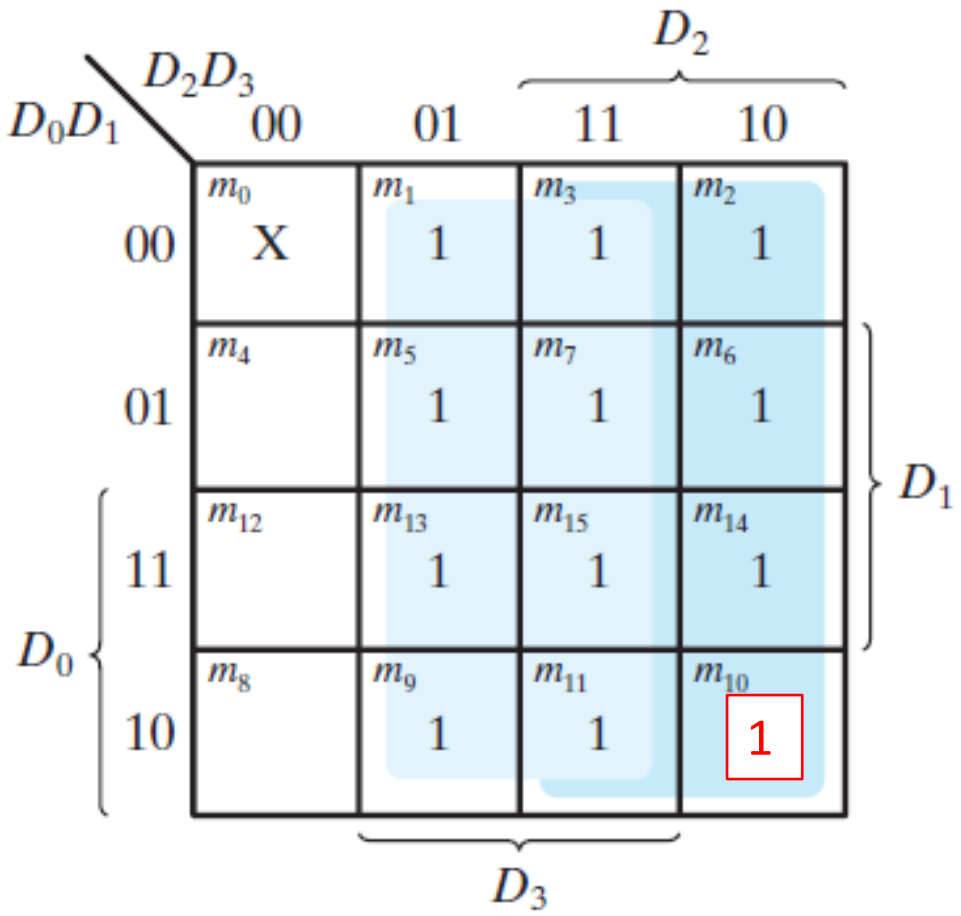
Highest Priority
เช่น หลอดไฟติดพร้อมกันมากกว่า 1 ดวง
จะให้ตอบว่าดวงไหน

Table 4.8
Truth Table of a Priority Encoder

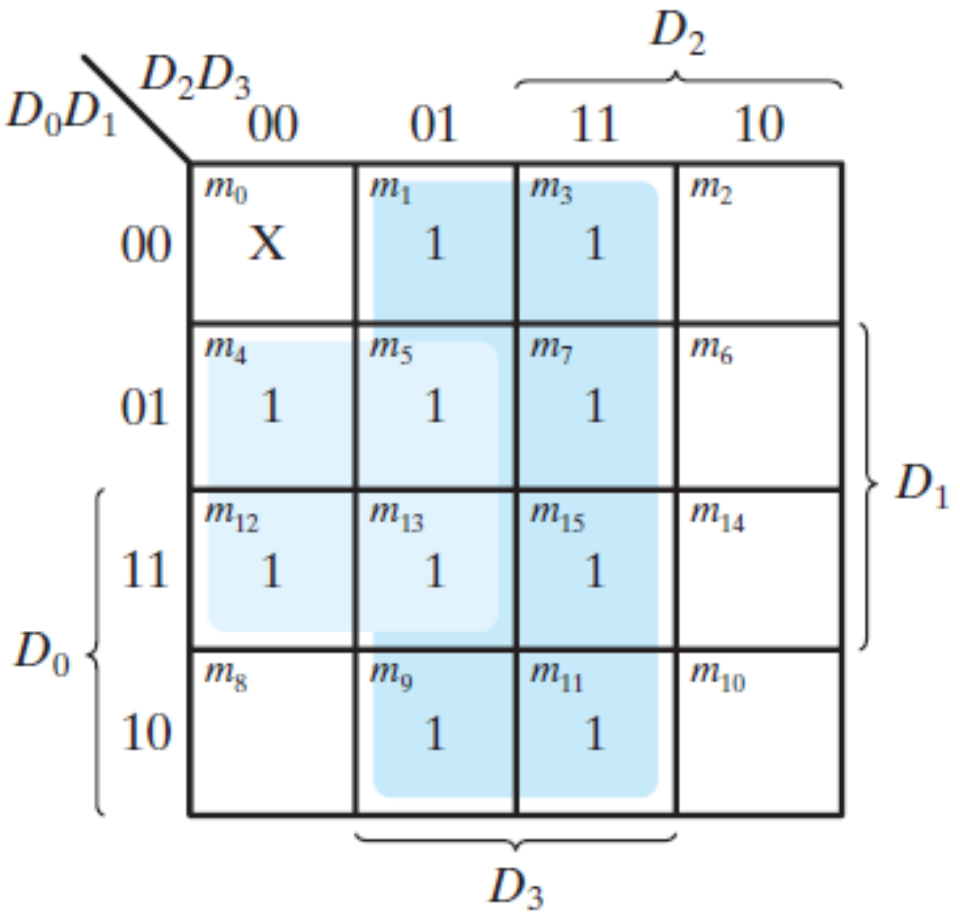
Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

มีหลอดไฟติด
อย่างน้อย 1 ดวง

X คือ don’t care



$x = D_2 + D_3$



$y = D_3 + D_1D'_2$

FIGURE 4.22
Maps for a priority encoder

$V = D_1 + D_2 + D_3 + D_4$

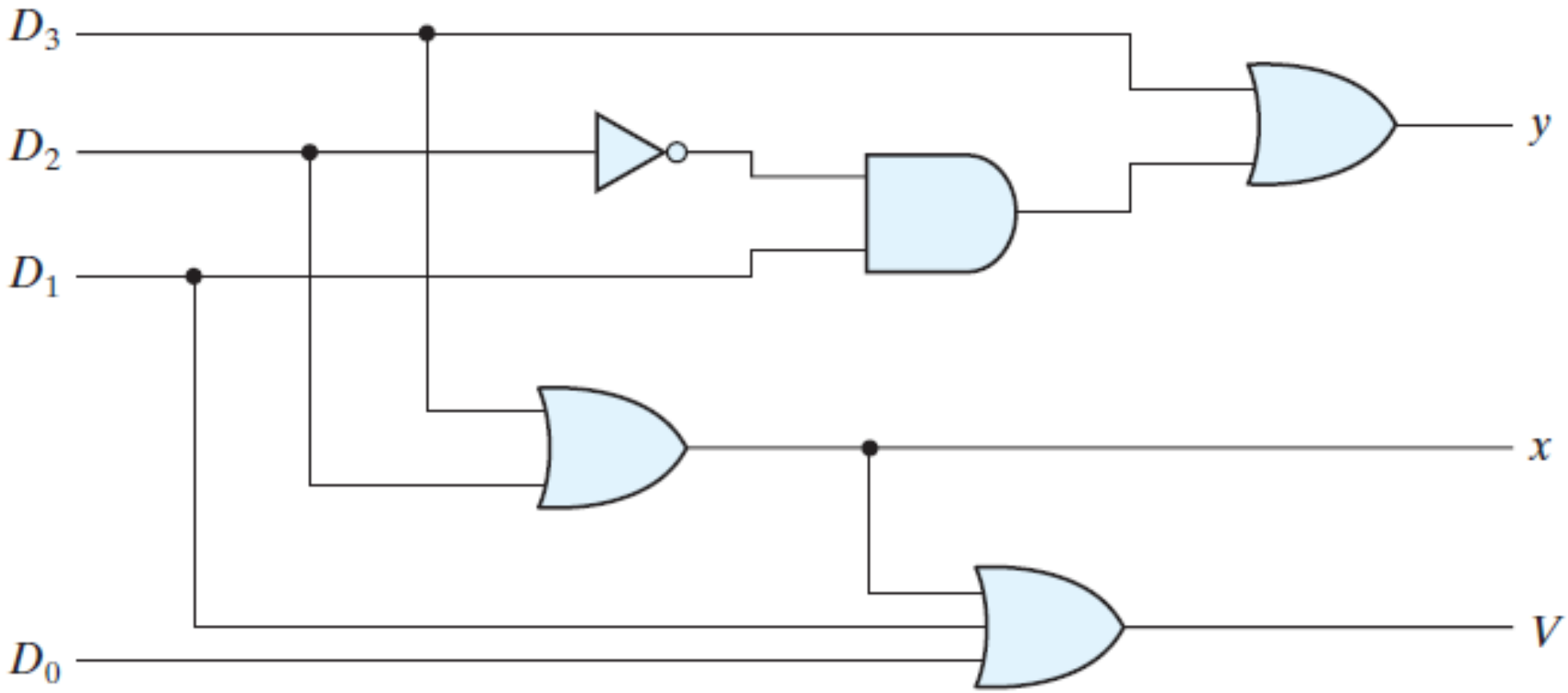
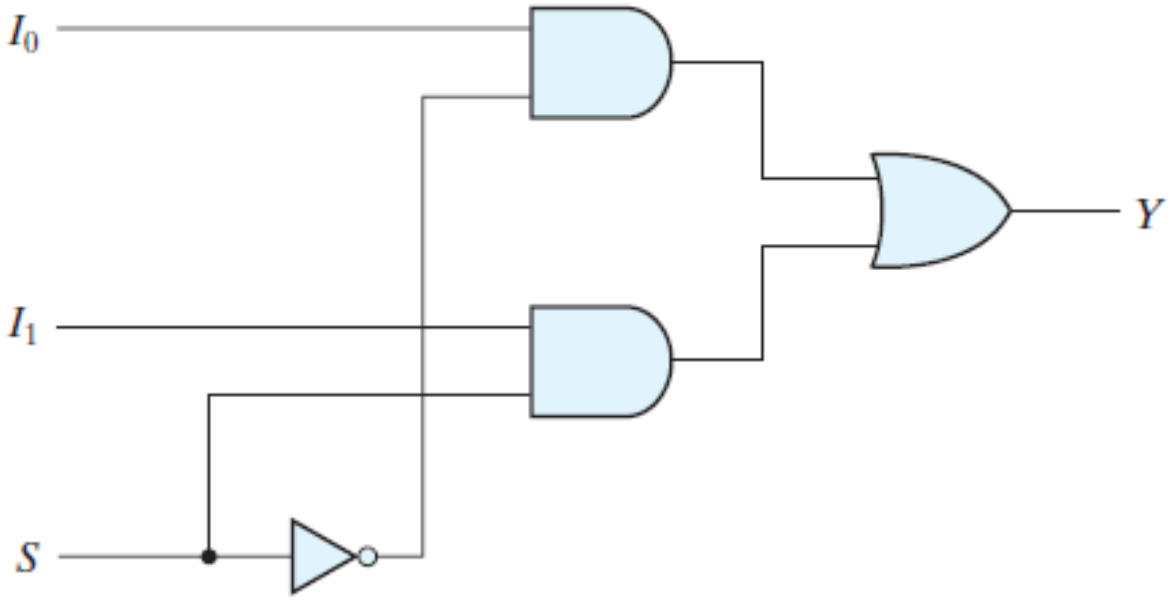


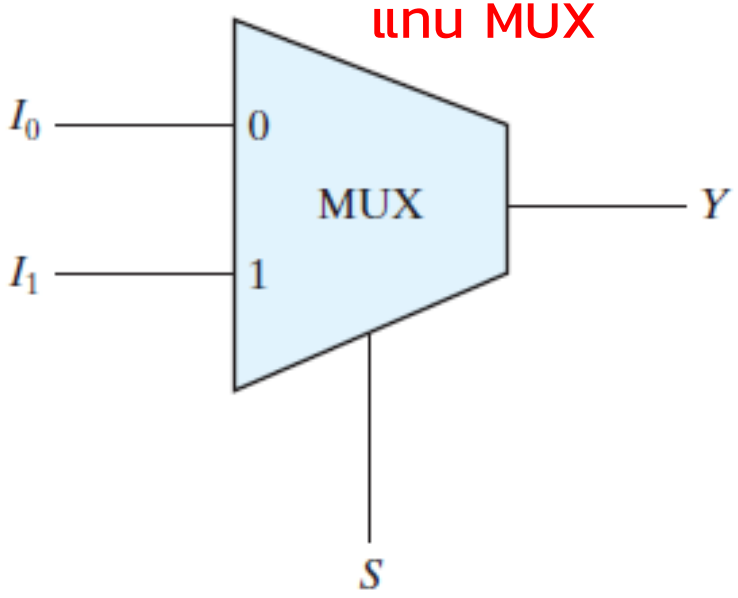
FIGURE 4.23
Four-input priority encoder

Multiplexor (if-else)

ใช้รูปสี่เหลี่ยมคางหมู
แทน MUX

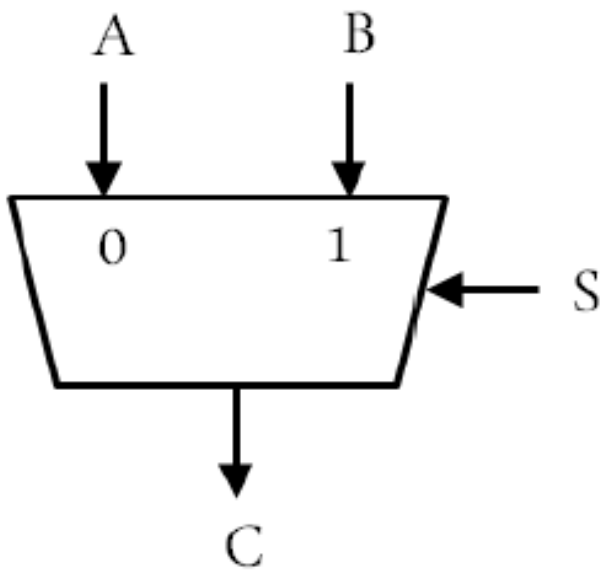


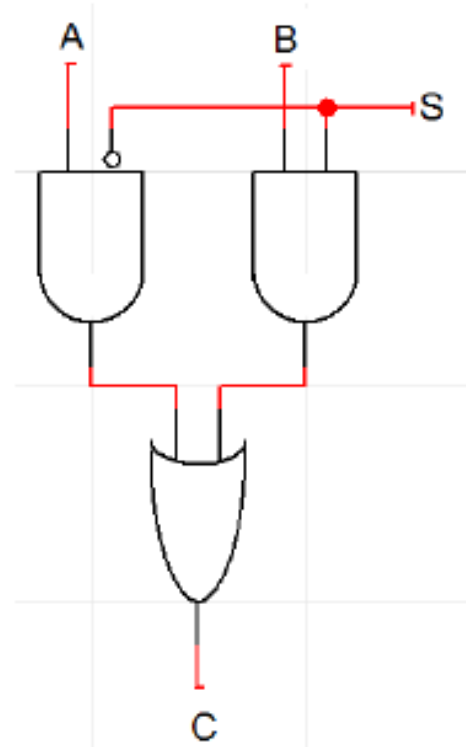
(a) Logic diagram



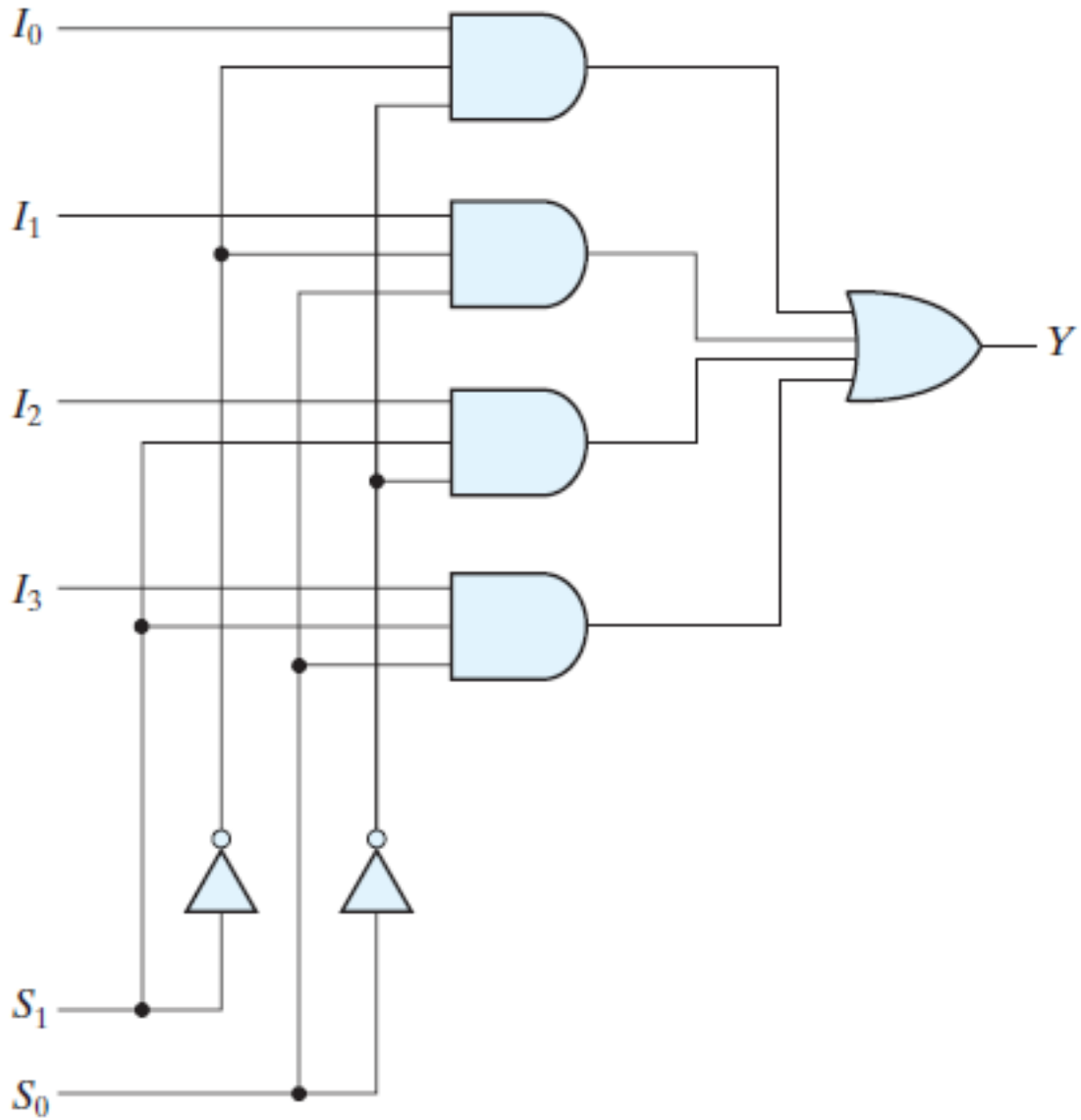
(b) Block diagram

FIGURE 4.24
Two-to-one-line multiplexer





(ก) มักซ์ 2-to-1

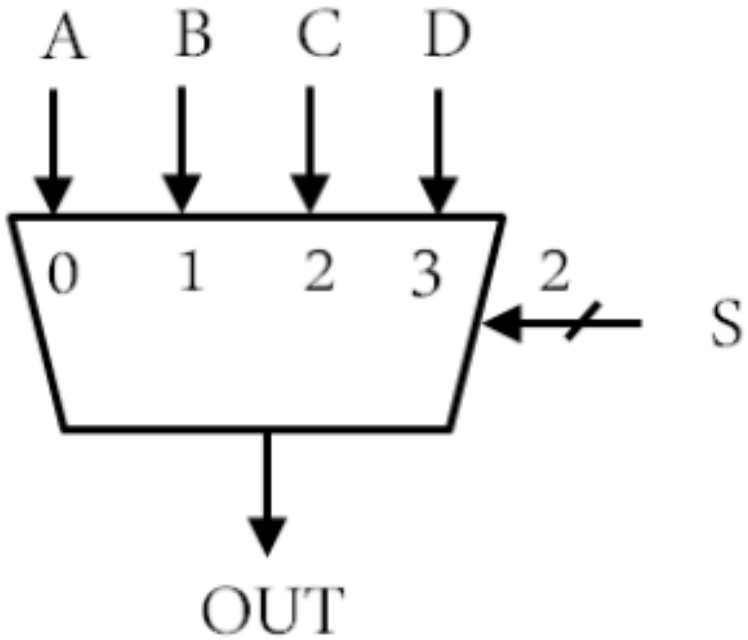


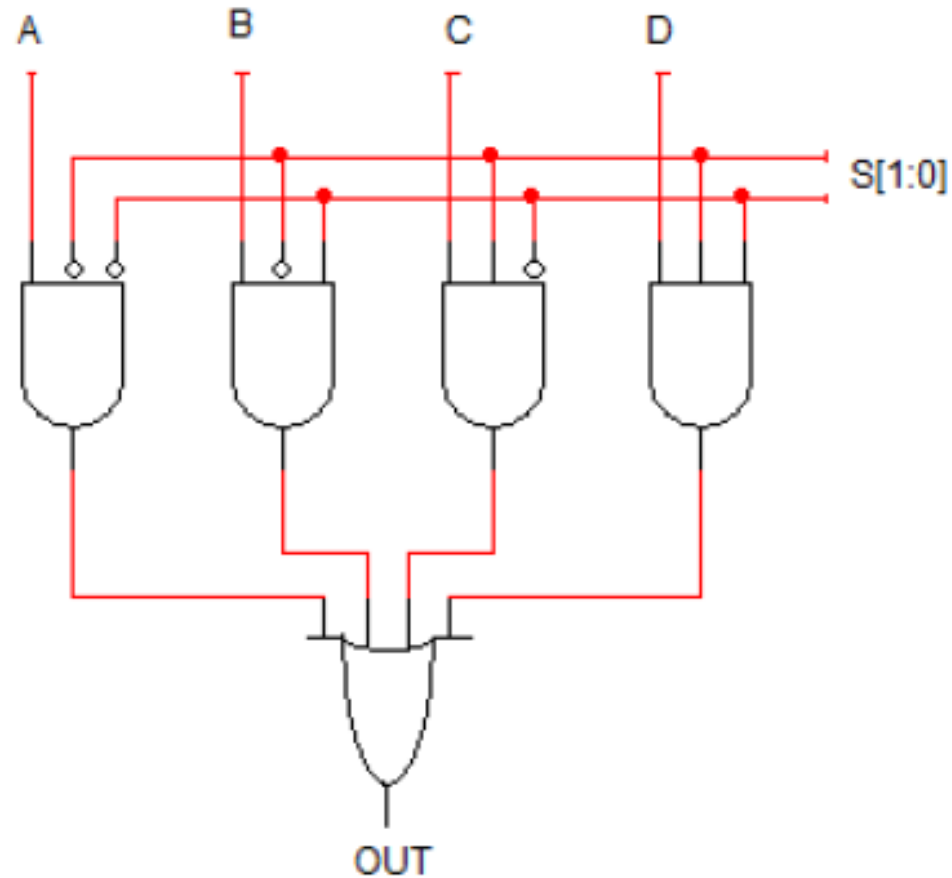
(a) Logic diagram

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

FIGURE 4.25
Four-to-one-line multiplexer





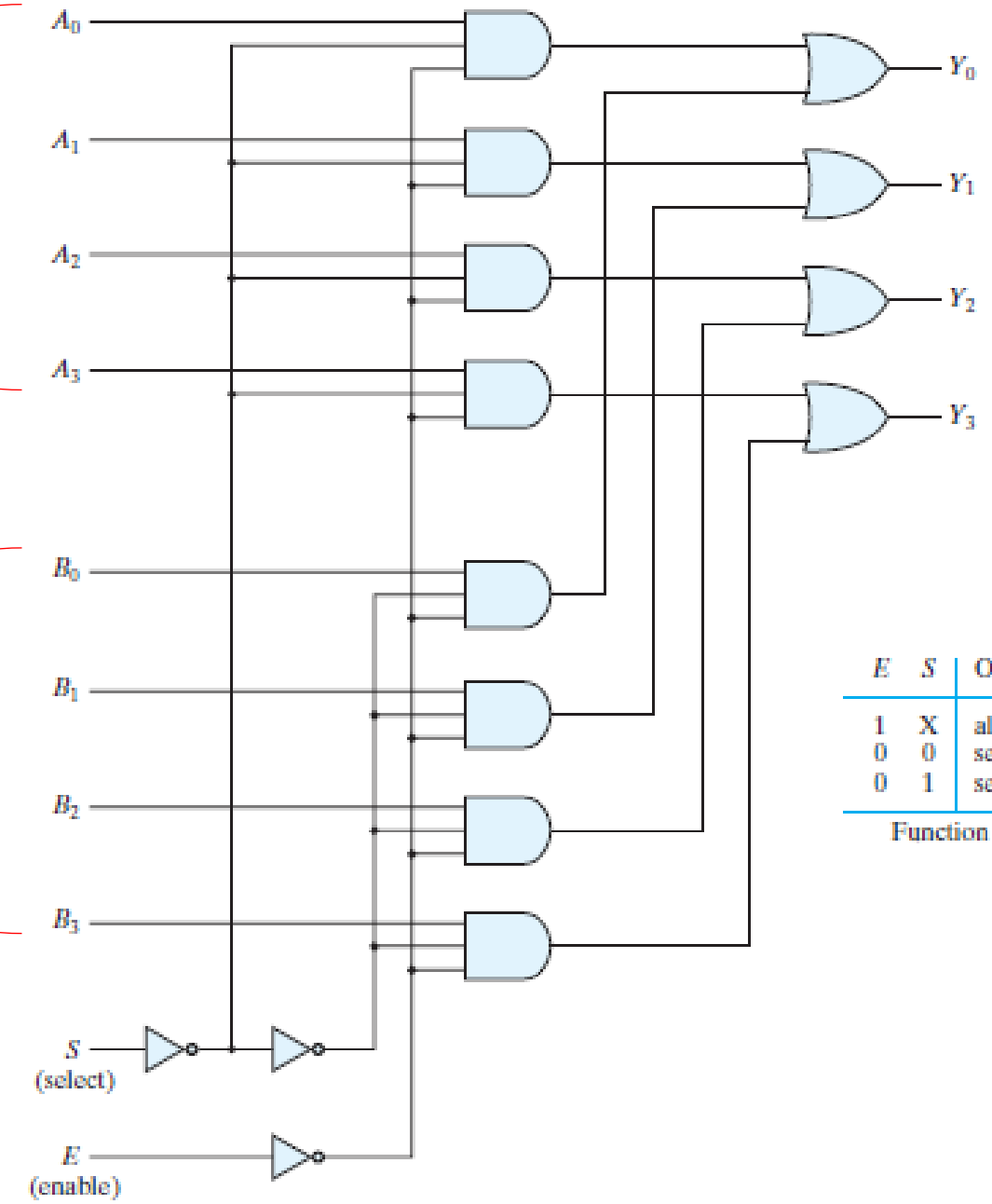
(ข) 4-to-1 MUX

$A_3A_2A_1A_0$

Quadruple

$B_3B_2B_1B_0$

Quadruple



E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

Function table

FIGURE 4.26
Quadruple two-to-one-line multiplexer

Three-State Gates หรือ Tri-State Buffers

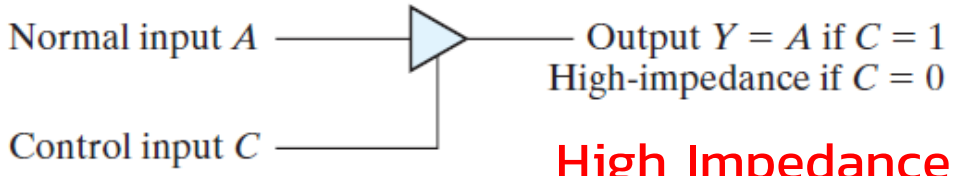
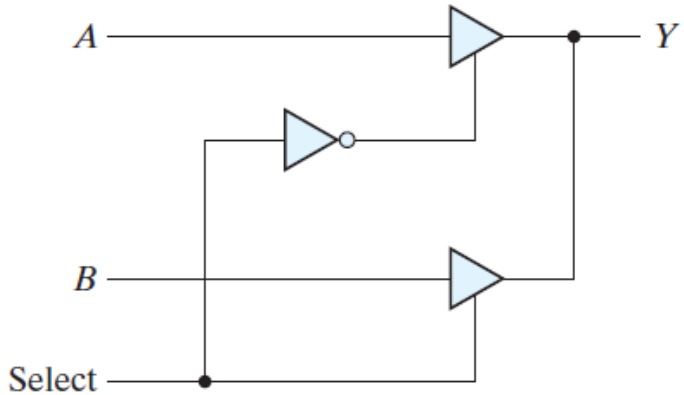


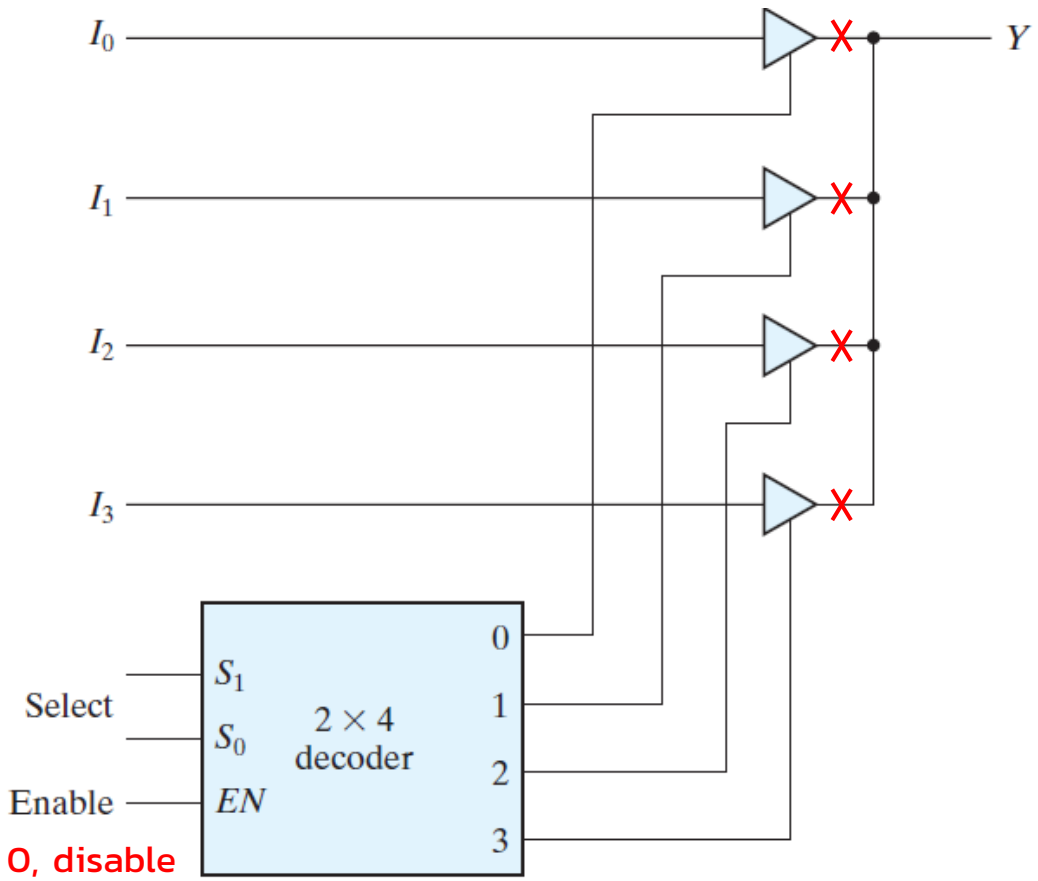
FIGURE 4.29
Graphic symbol for a three-state buffer

High Impedance (Z)
ความต้านทานสูงมาก
Open Circuit

EN = 0 จะตัดสายอินพุตทั้งหมดออกไป
EN = 1 จะเอาเฉพาะอินพุตที่ถูก decoded



(a) 2-to-1-line mux

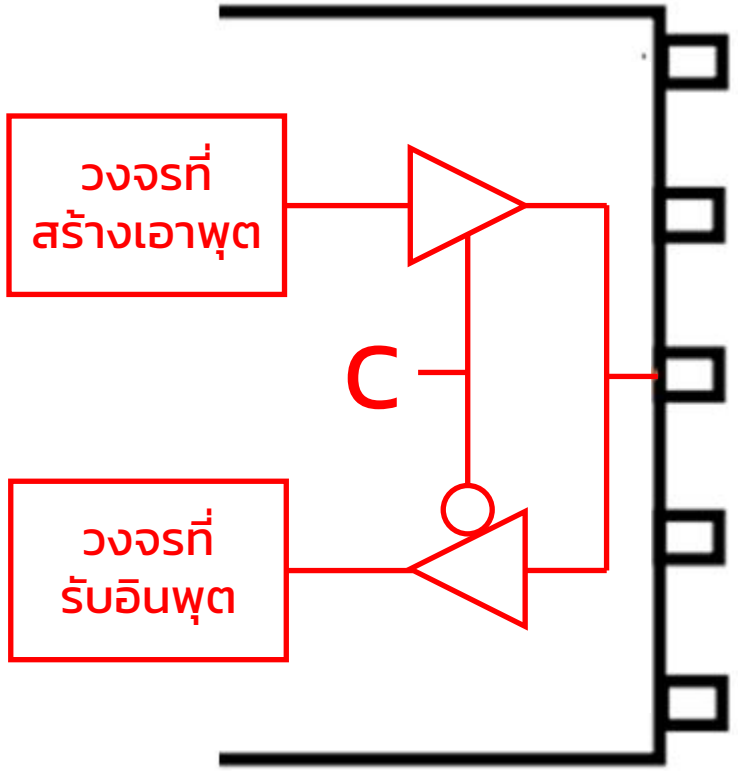


En = 0, disable
En = 1, enable

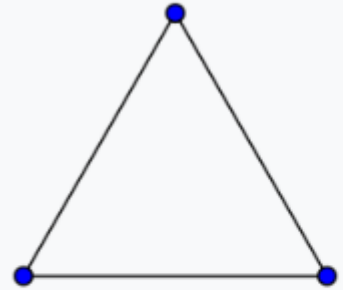
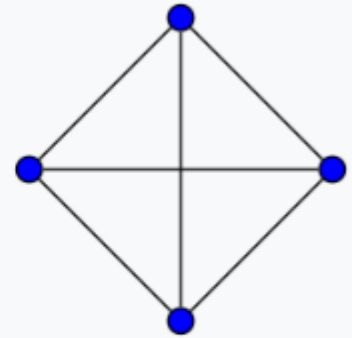
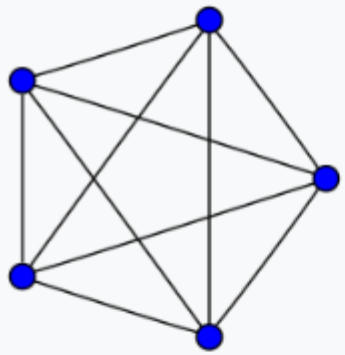
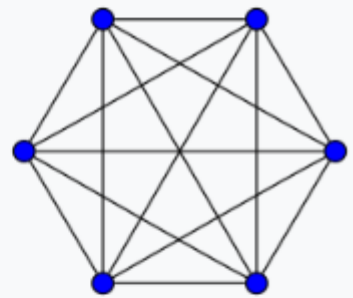
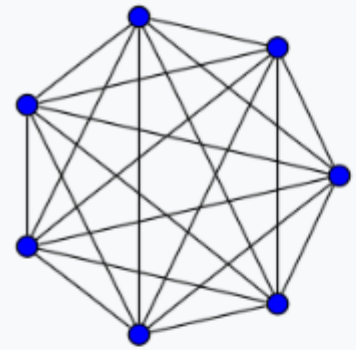
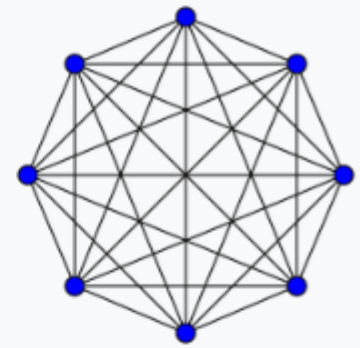
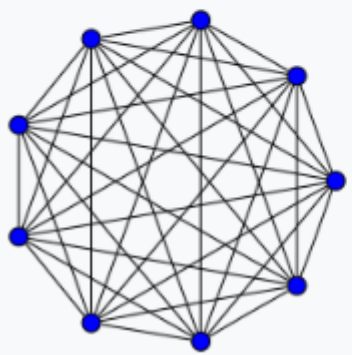
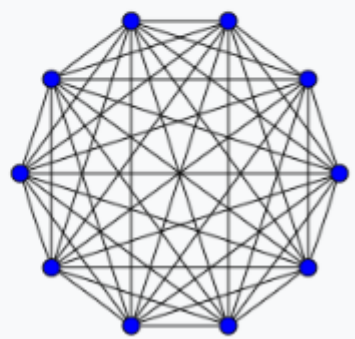
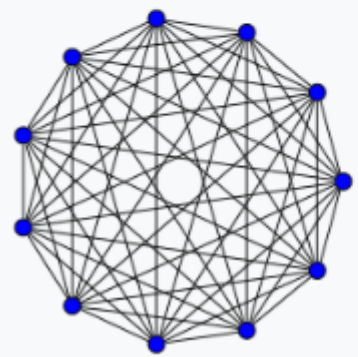
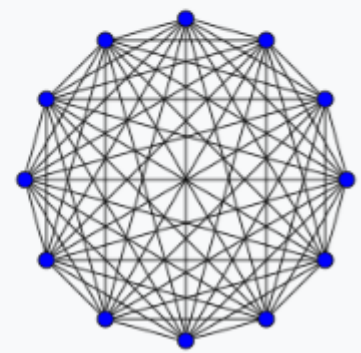
(b) 4-to-1-line mux

FIGURE 4.30
Multiplexers with three-state gates

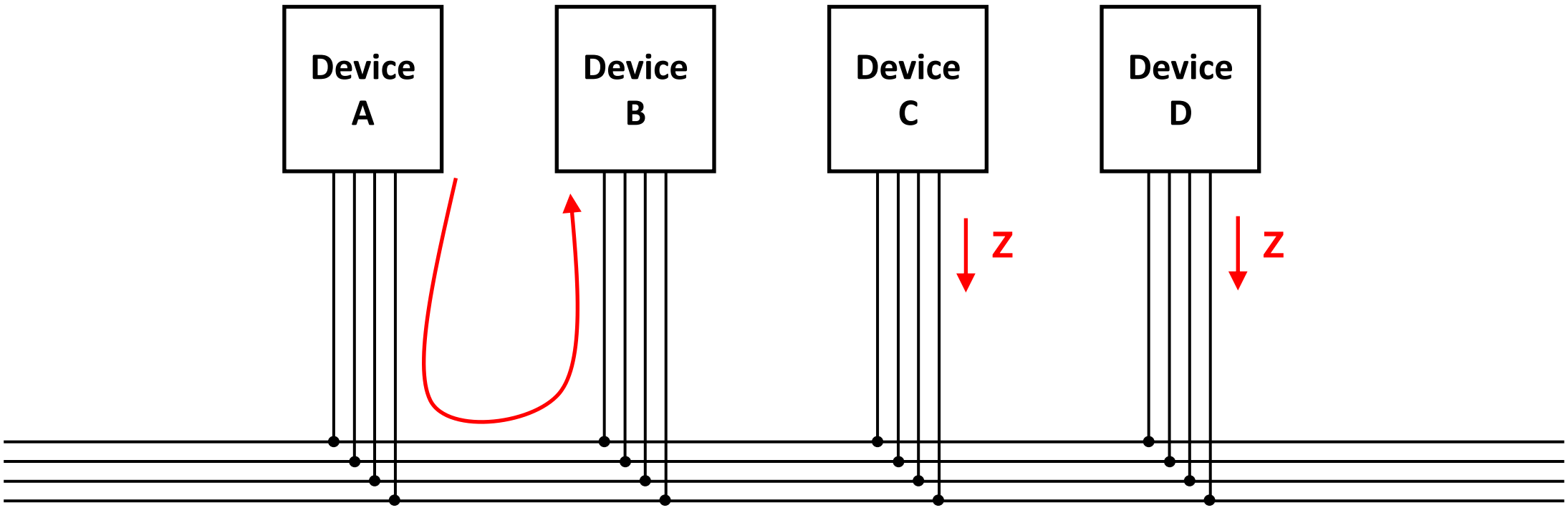
Bidirectional Ports



C = 0 ขานี้จะเป็น input port
C = 1 ขานี้จะเป็น output port

K_1: 0	K_2: 1	K_3: 3	K_4: 6
			
K_5: 10	K_6: 15	K_7: 21	K_8: 28
			
K_9: 36	K_{10}: 45	K_{11}: 55	K_{12}: 66
			

Data Bus


Device A
Device B
Device C
Device D
Z
Z
ผลัดกันรับส่งข้อมูลผ่าน data bus ทีละคู่
4-bit data bus

PROBLEMS

(Answers to problems marked with * appear at the end of the text. Where appropriate, a logic design and its related HDL modeling problem are cross-referenced.)

4.1 Consider the combinational circuit shown in Fig. P4.1. (HDL—see Problem 4.49.)

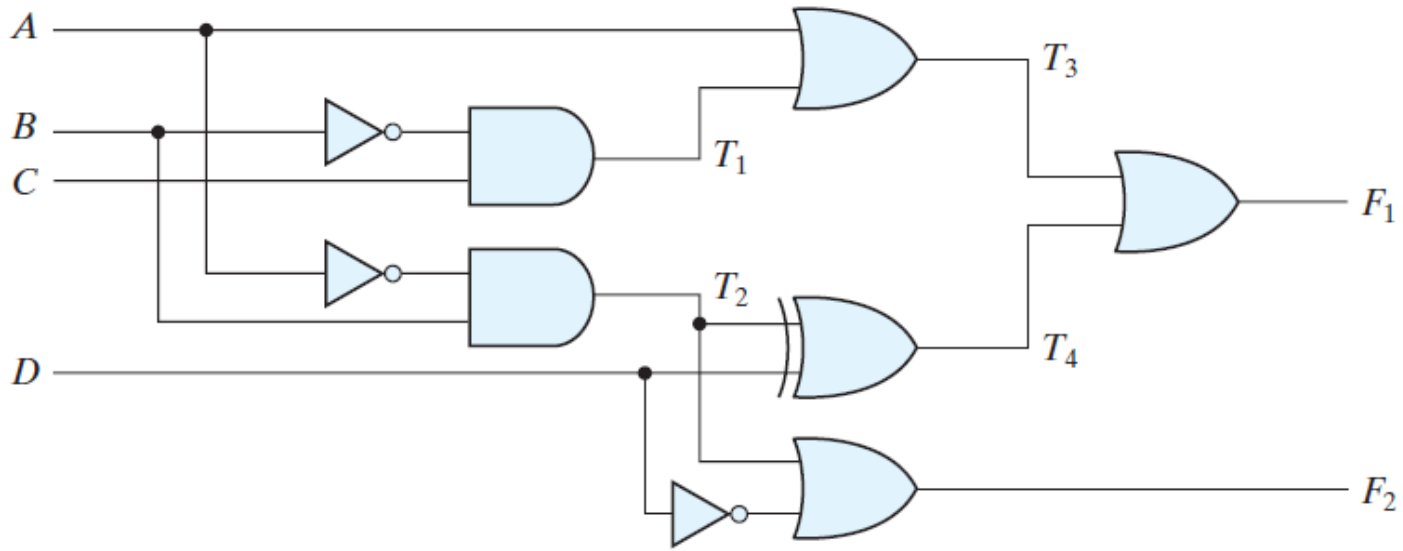


FIGURE P4.1

(a)* Derive the Boolean expressions for T_1 through T_4. Evaluate the outputs F_1 and F_2 as a function of the four inputs.

(b) List the truth table with 16 binary combinations of the four input variables. Then list the binary values for T_1 through T_4 and outputs F_1 and F_2 in the table.

(c) Plot the output Boolean functions obtained in part (b) on maps and show that the simplified Boolean expressions are equivalent to the ones obtained in part (a).