جامعة فلسطين التقنية - خضوري
Palestine Technical University - Kadoorie

Digital Electronics and Logic Design

Combinational Logic Analysis

Dr. Jafar Saifeddin Jallad

Dept. of Electrical Engineering

Palestine Technical University

Tulkaram, Palestine

Code Converters

BCD-to-Binary Conversion

The binary numbers representing the weights of the BCD bits are summed to produce the total binary number.

Let's examine an 8-bit BCD code (one that represents a 2-digit decimal number) to understand the relationship between BCD and binary. For instance, you already know that the decimal number 87 can be expressed in BCD as

$$\underbrace{1000}_{8} \quad \underbrace{0111}_{7}$$

	Tens Digit				Units Digit			
Weight:	80	40	20	10	8	4	2	1
Bit designation:	B_3	B_2	B_1	B_0	A_3	A_2	A_1	A_0

Binary representations of BCD bit weights.

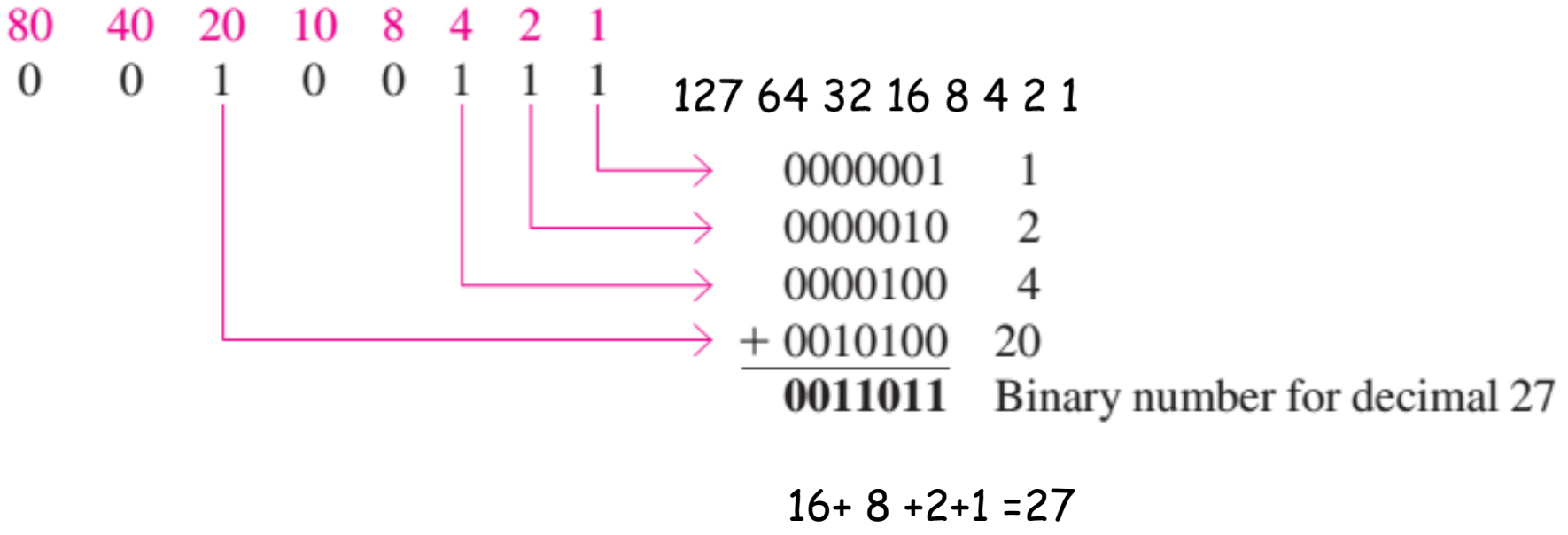
BCD Bit	BCD Weight	(MSB) Binary Representation						(LSB)
		64	32	16	8	4	2	1
A_0	1	0	0	0	0	0	0	1
A_1	2	0	0	0	0	0	1	0
A_2	4	0	0	0	0	1	0	0
A_3	8	0	0	0	1	0	0	0
B_0	10	0	0	0	1	0	1	0
B_1	20	0	0	1	0	1	0	0
B_2	40	0	1	0	1	0	0	0
B_3	80	1	0	1	0	0	0	0

EXAMPLE

Convert the BCD numbers 00100111 (decimal 27) to binary.

Solution

Write the binary representations of the weights of all 1s appearing in the numbers, and then add them together.

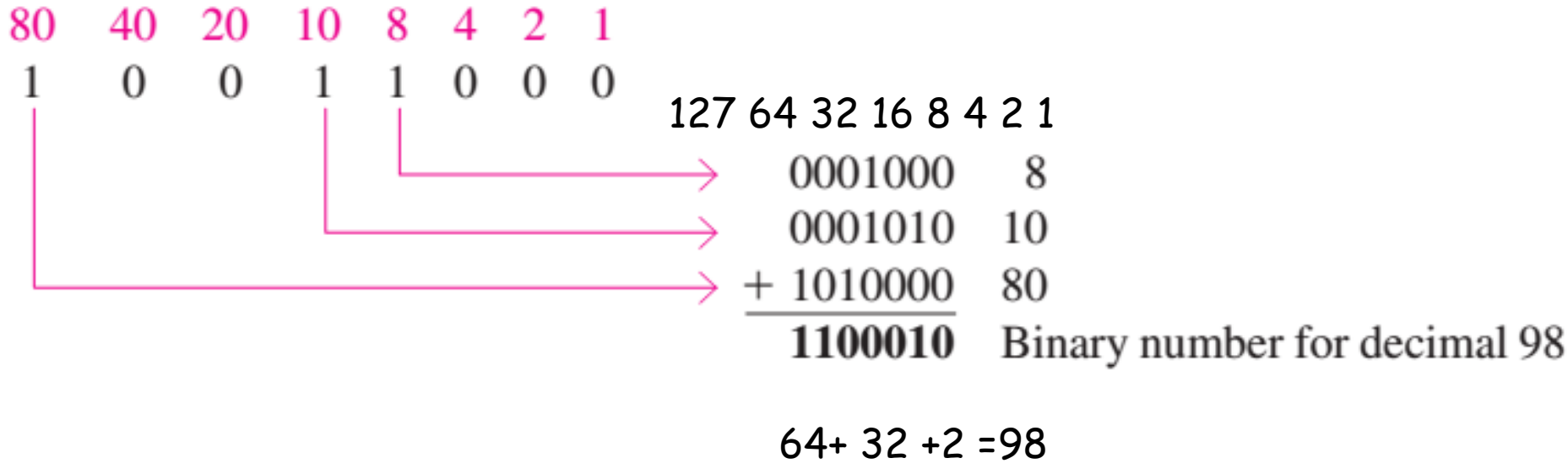


EXAMPLE

Convert the BCD numbers 10011000 (decimal 98) to binary.

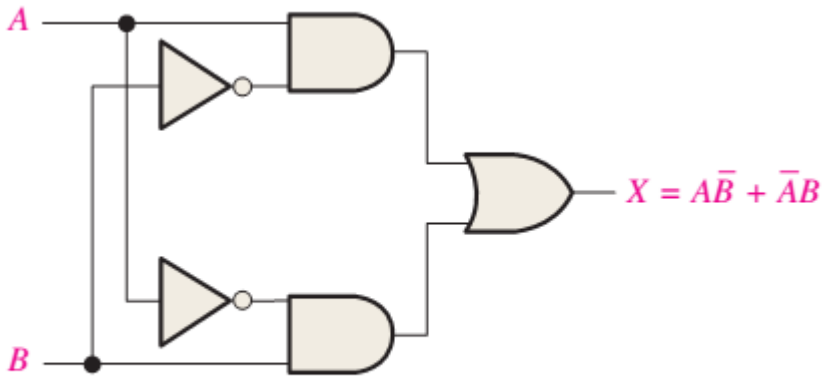
Solution

Write the binary representations of the weights of all 1s appearing in the numbers, and then add them together.



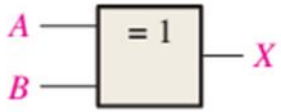
64+ 32 +2 =98

Exclusive-OR Logic



(a) Logic diagram

(b) ANSI distinctive shape symbol



(c) ANSI rectangular outline symbol

$$X = A \oplus B$$

$$X = A\overline{B} + \overline{A}B$$

Truth table for an exclusive-OR.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

$$x \oplus 0 = x$$

$$x \oplus 1 = x'$$

$$x \oplus x = 0$$

$$x \oplus x' = 1$$

$$x \oplus y' = x' \oplus y = (x \oplus y)'$$

$$A \oplus B = B \oplus A$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$$

$$X \oplus 0 = X$$

$$X \oplus 1 = \overline{X}$$

$$X \oplus X = 0$$

$$X \oplus \overline{X} = 1$$

$$\overline{X} \oplus \overline{X} = 0$$

$$\overline{X} \oplus \overline{X} = X \oplus X$$

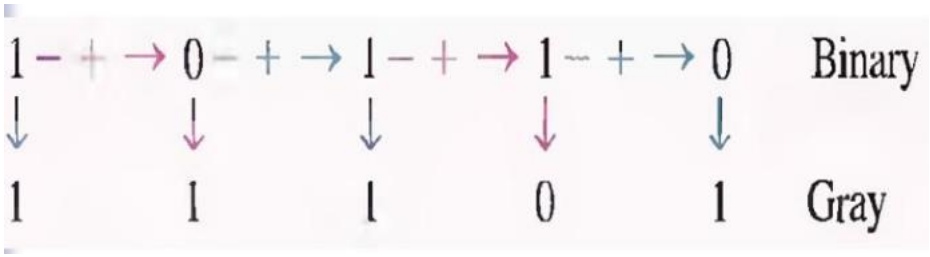
$$A \oplus AB = A\overline{B}$$

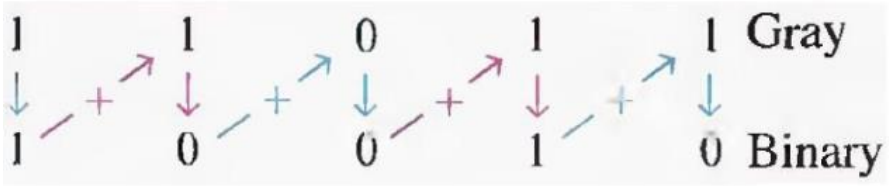
$$A \oplus A\overline{B} = AB$$

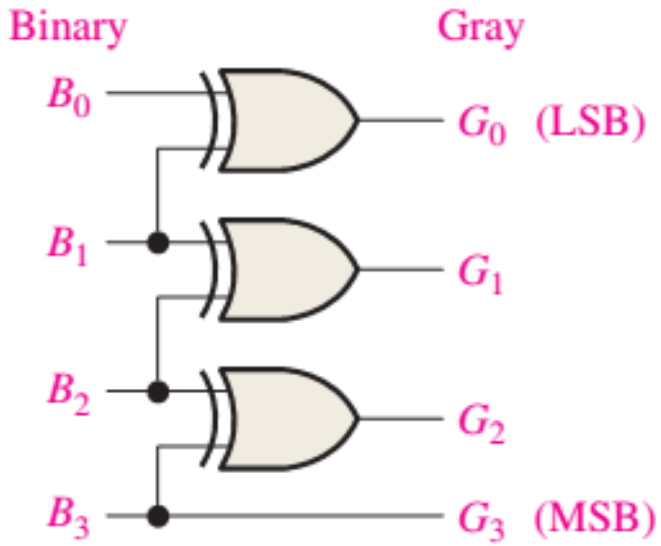
$$A \oplus \overline{A}B = A + B$$

$$(A \oplus B).(A \oplus C) = \overline{A}BC + A\overline{B}\overline{C}$$

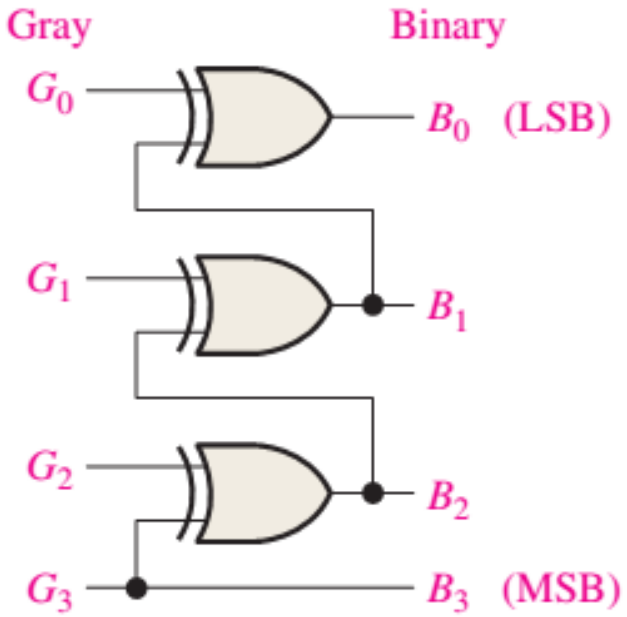
Binary-to-Gray and Gray-to-Binary Conversion







Four-bit binary-to-Gray conversion logic. Open file



Four-bit Gray-to-binary conversion logic. Open file

The Half-Adder

A half-adder adds two bits and produces a sum and an output carry.

Recall the basic rules for binary addition as stated in Chapter 2.

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

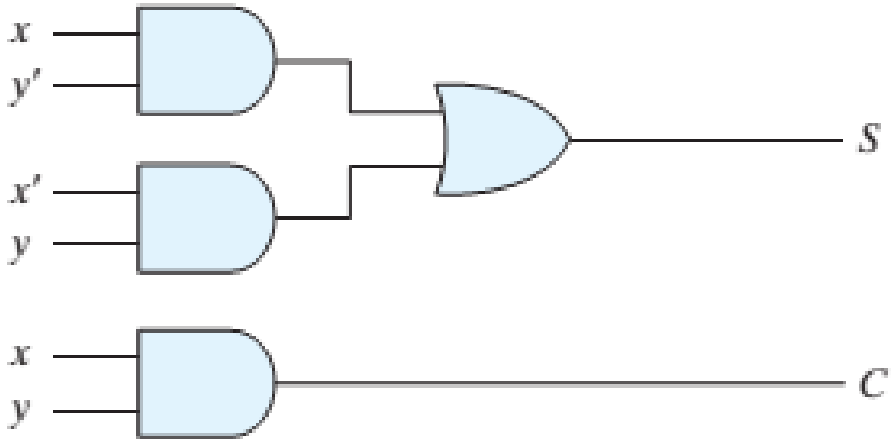
$1 + 1 = 10$

Half Adder

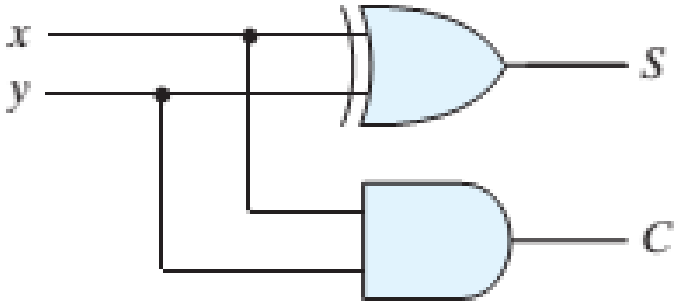
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = x'y + xy'$$

$$C = xy$$

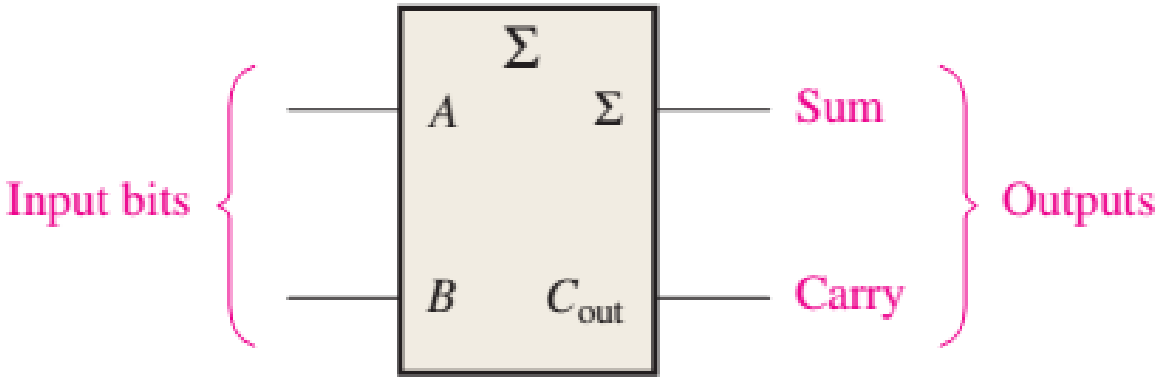


(a) $S = xy' + x'y$
$C = xy$



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

A half-adder is represented by the logic symbol in Figure



The Full-Adder

A full-adder has an input carry while the half-adder does not.

Full Adder

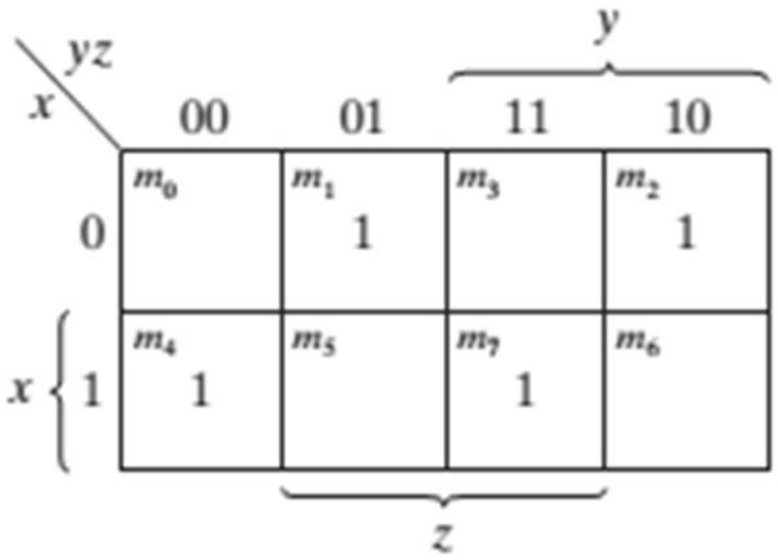
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Carry bit from right column

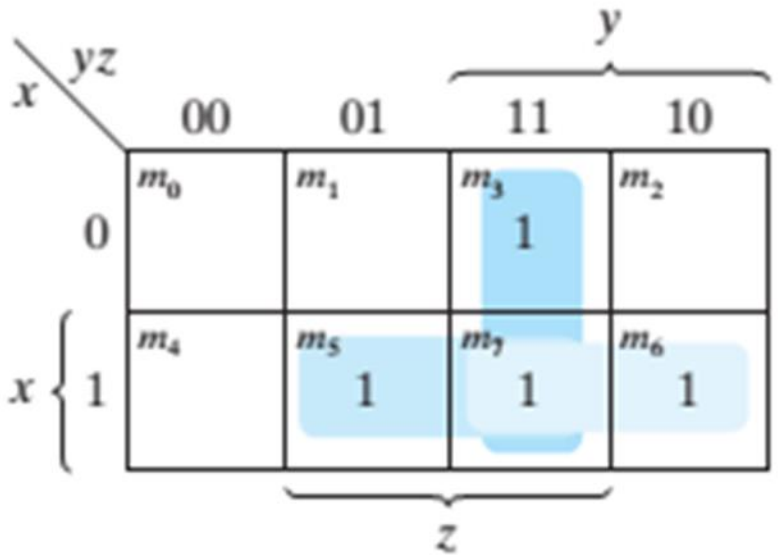
$$\begin{array}{r} {\scriptstyle 1} \\ 11 \\ +\,01 \\ \hline 100 \end{array}$$

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$C = xy + xz + yz$$



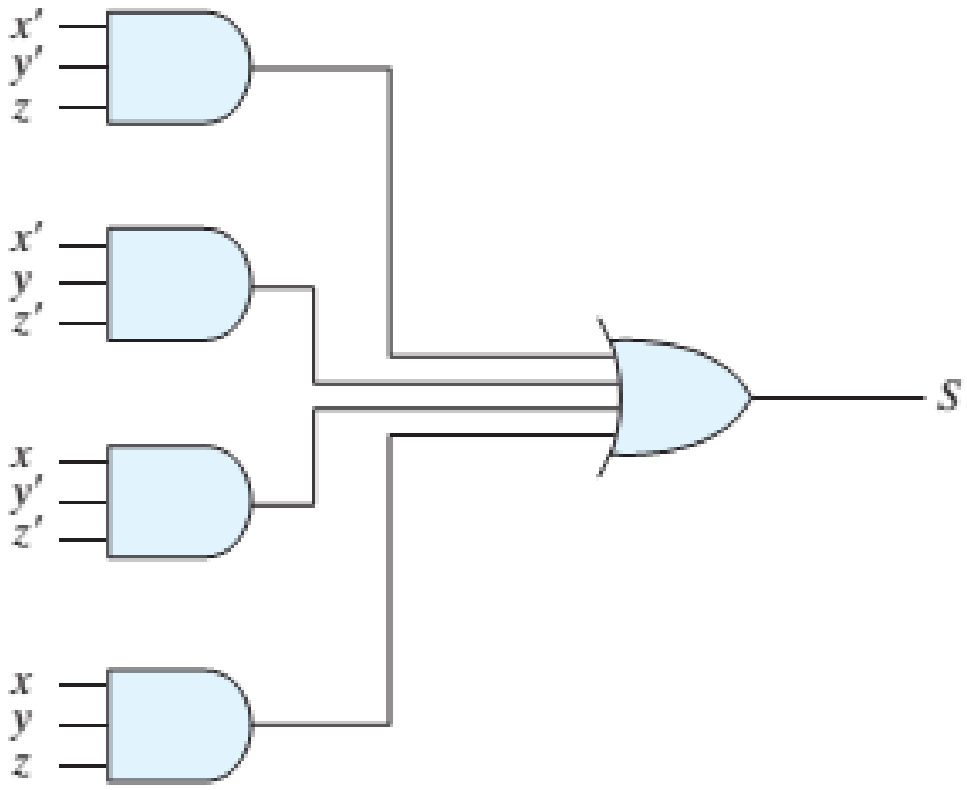
(a) $S = x'y'z + x'yz' + xy'z' + xyz$

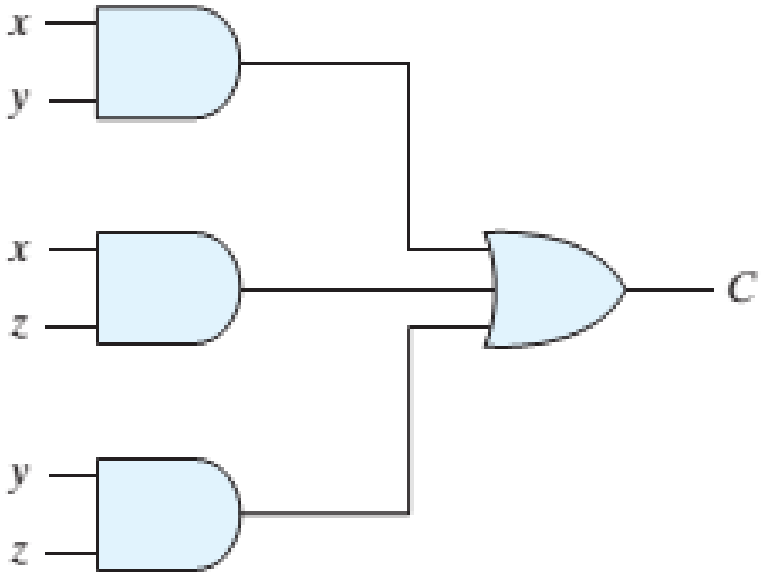


(b) $C = xy + xz + yz$

$$S = x'y'z + x'yz' + xy'z' + xyz$$

$$C = xy + xz + yz$$



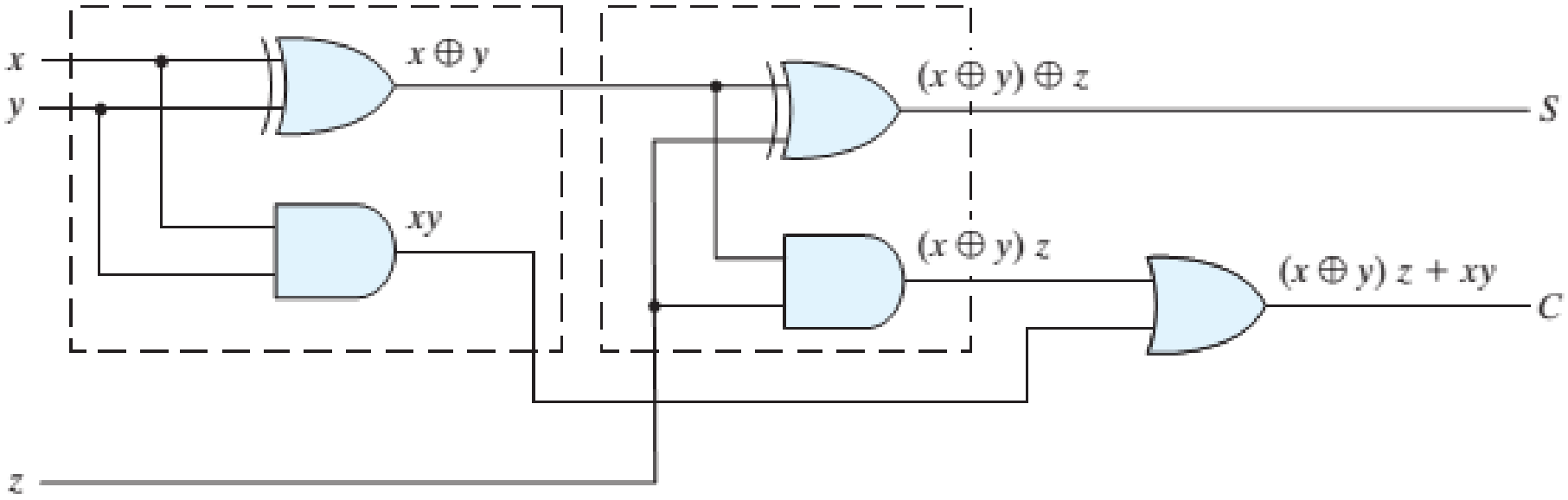


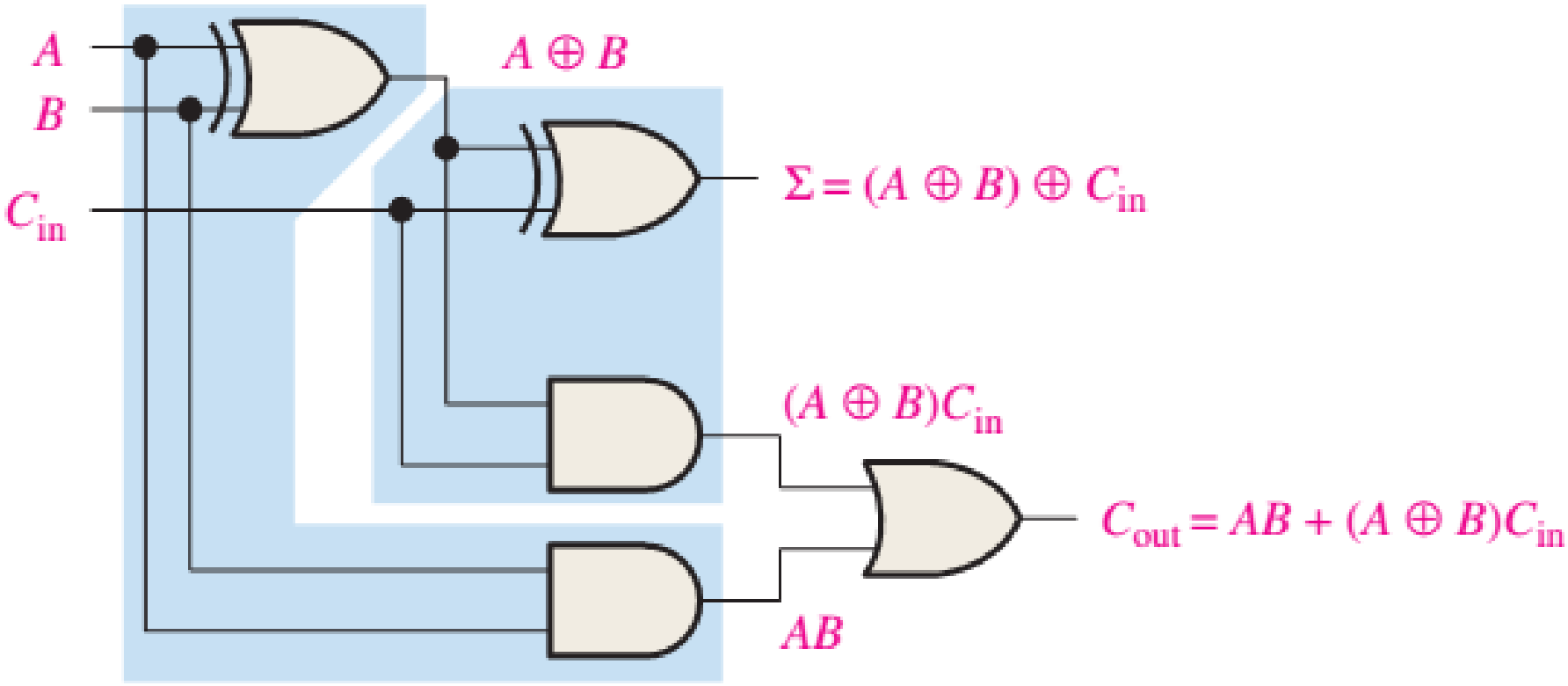
Full Adder

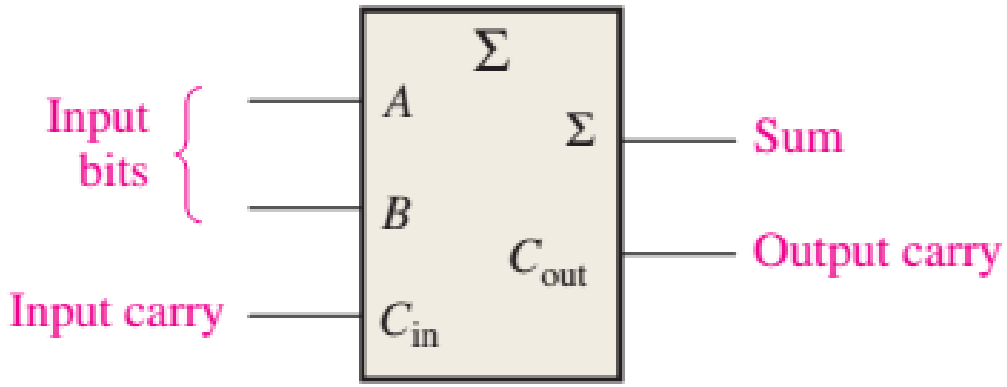
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$
\begin{aligned}
S &= xy'z' + x'yz' + xyz + x'y'z \\
&= z'(xy' + x'y) + z(xy + x'y') \\
&= z'(xy' + x'y) + z(xy' + x'y)' \\
&= z \oplus (x \oplus y)
\end{aligned}
$$

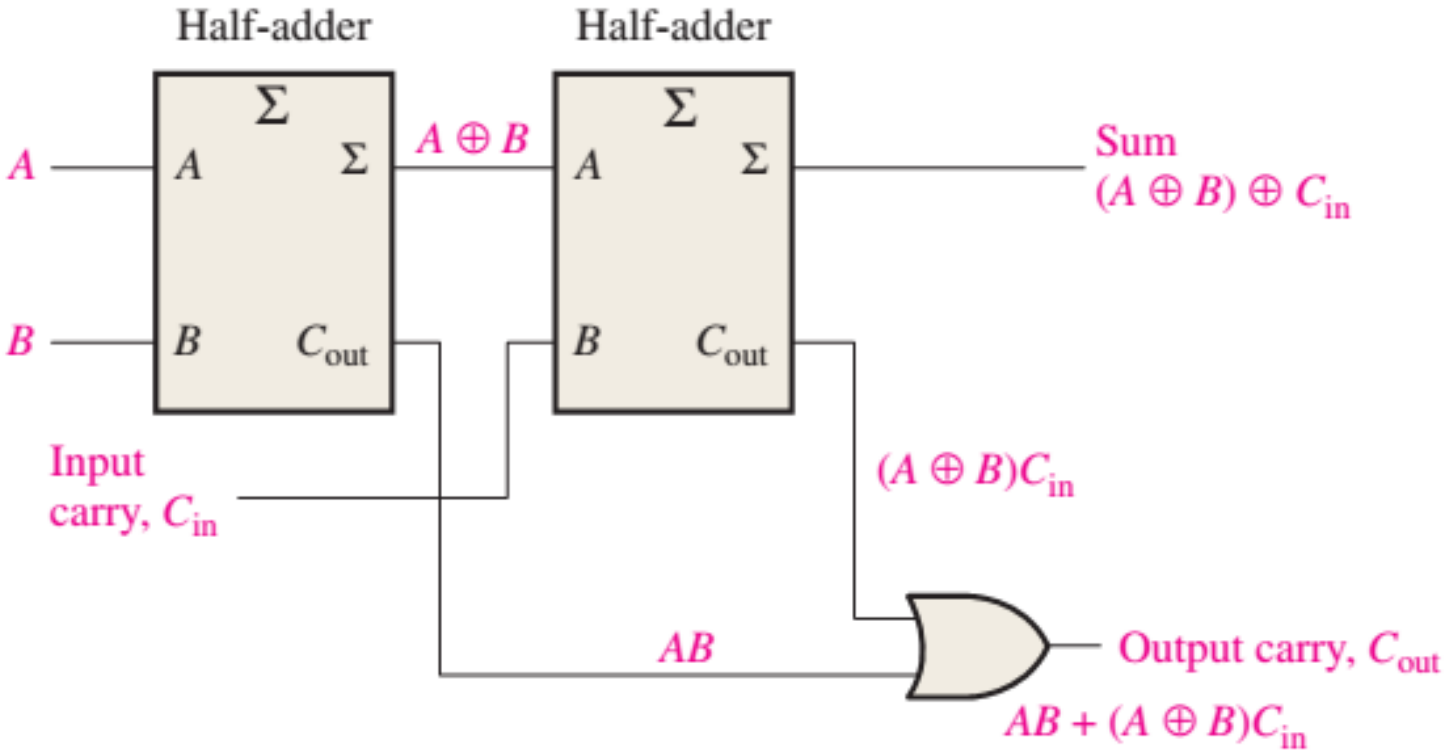
$$
\begin{aligned}
C &= xy'z + x'yz + xy \\
&= z(xy' + x'y) + xy \\
&= z(x \oplus y) + xy
\end{aligned}
$$




A
B
C_{in}
$A \oplus B$
$\Sigma = (A \oplus B) \oplus C_{in}$
$(A \oplus B)C_{in}$
$C_{out} = AB + (A \oplus B)C_{in}$
AB



Logic symbol for a full-adder.



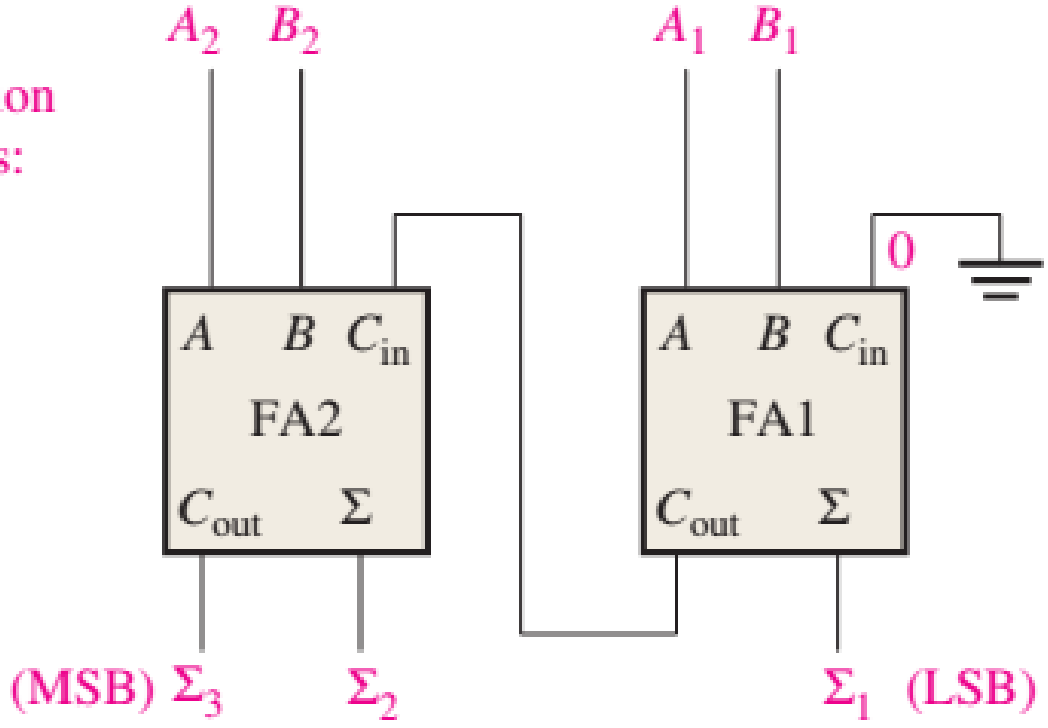
Carry bit from right column

$$\begin{array}{r} {\scriptstyle 1} \\ 11 \\ +\ 01 \\ \hline 100 \end{array}$$

In this case, the carry bit from second column becomes a sum bit.

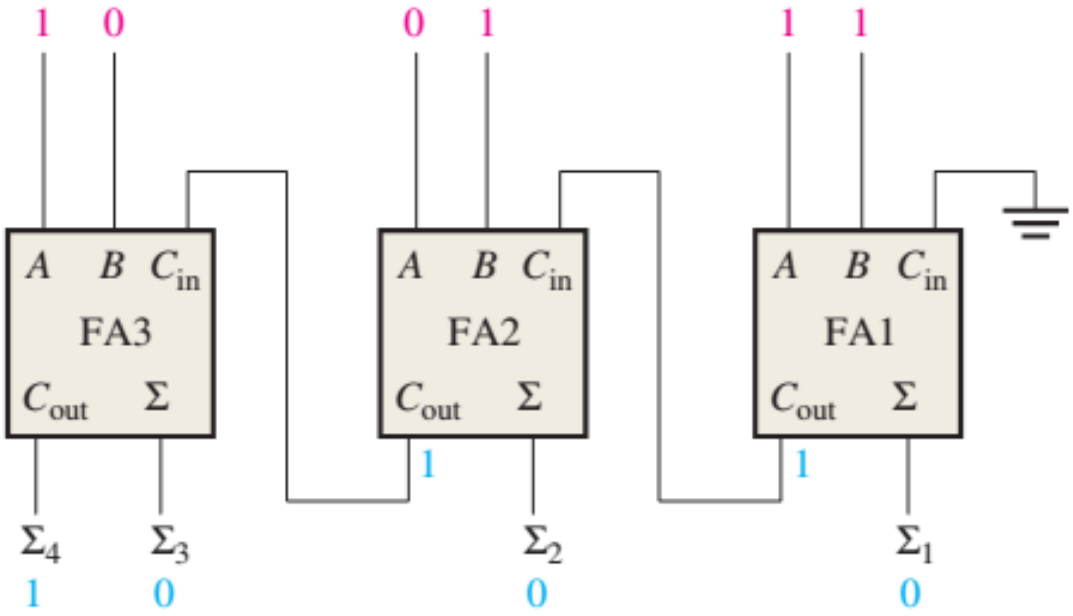
General format, addition of two 2-bit numbers:

$$\begin{array}{r} A_2A_1 \\ +\ B_2B_1 \\ \hline \Sigma_3\Sigma_2\Sigma_1 \end{array}$$

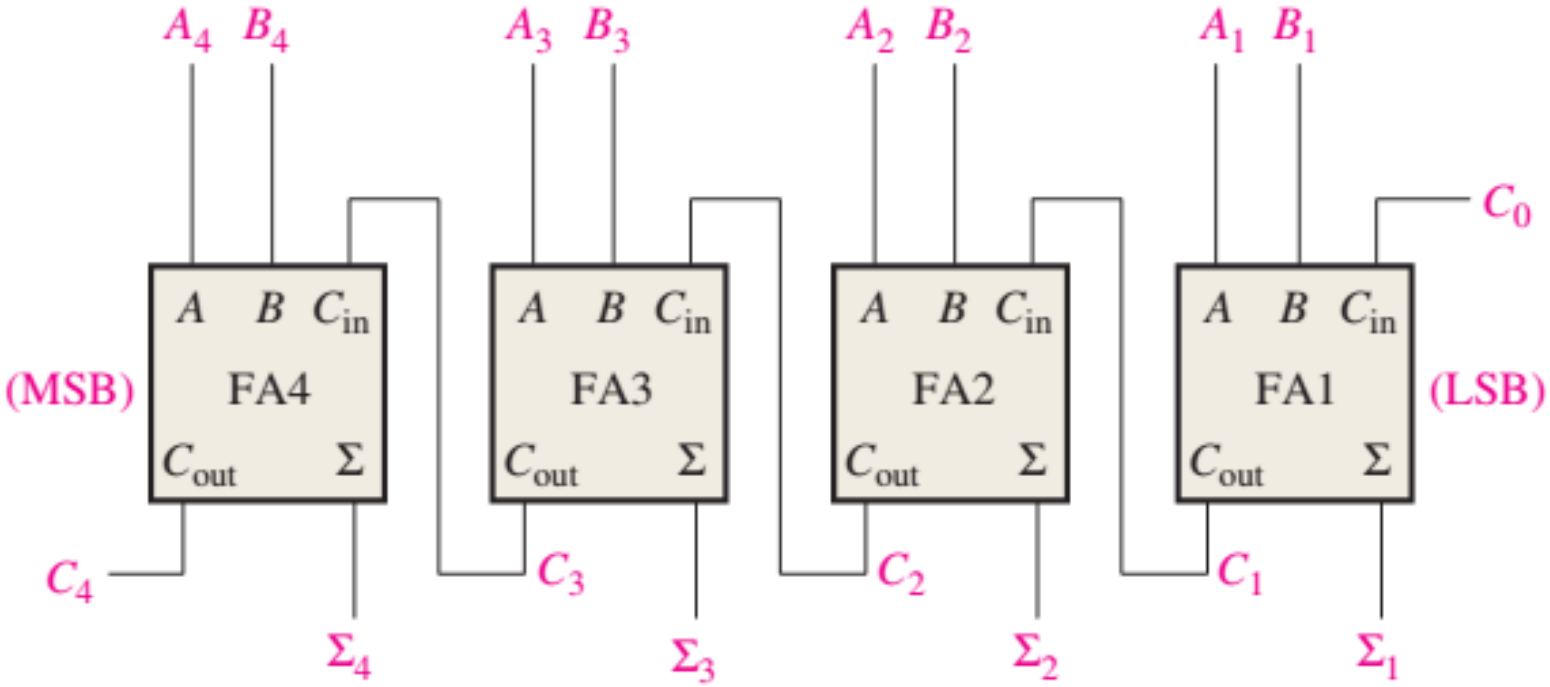


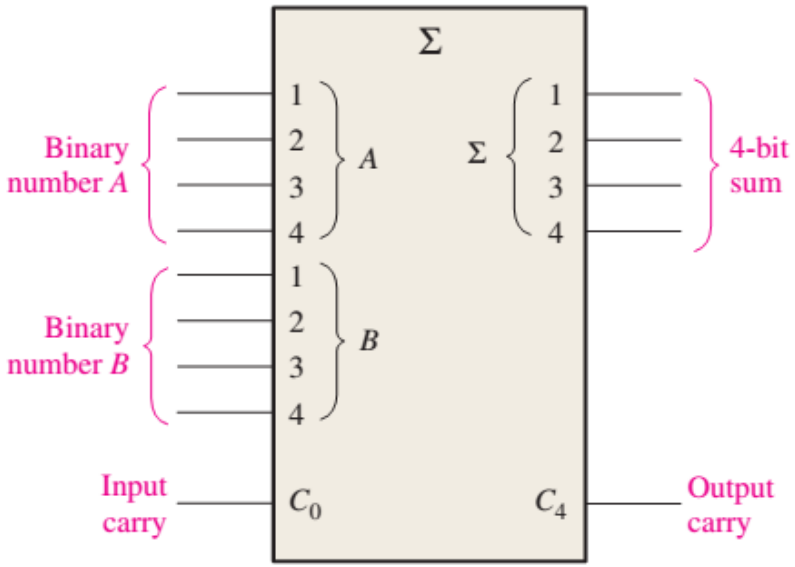
Block diagram of a basic 2-bit parallel adder using two full-adders.

Determine the sum generated by the 3-bit parallel adder in Figure and show the intermediate carries when the binary numbers 101 and 011 are being added.



A 4-bit parallel adder.

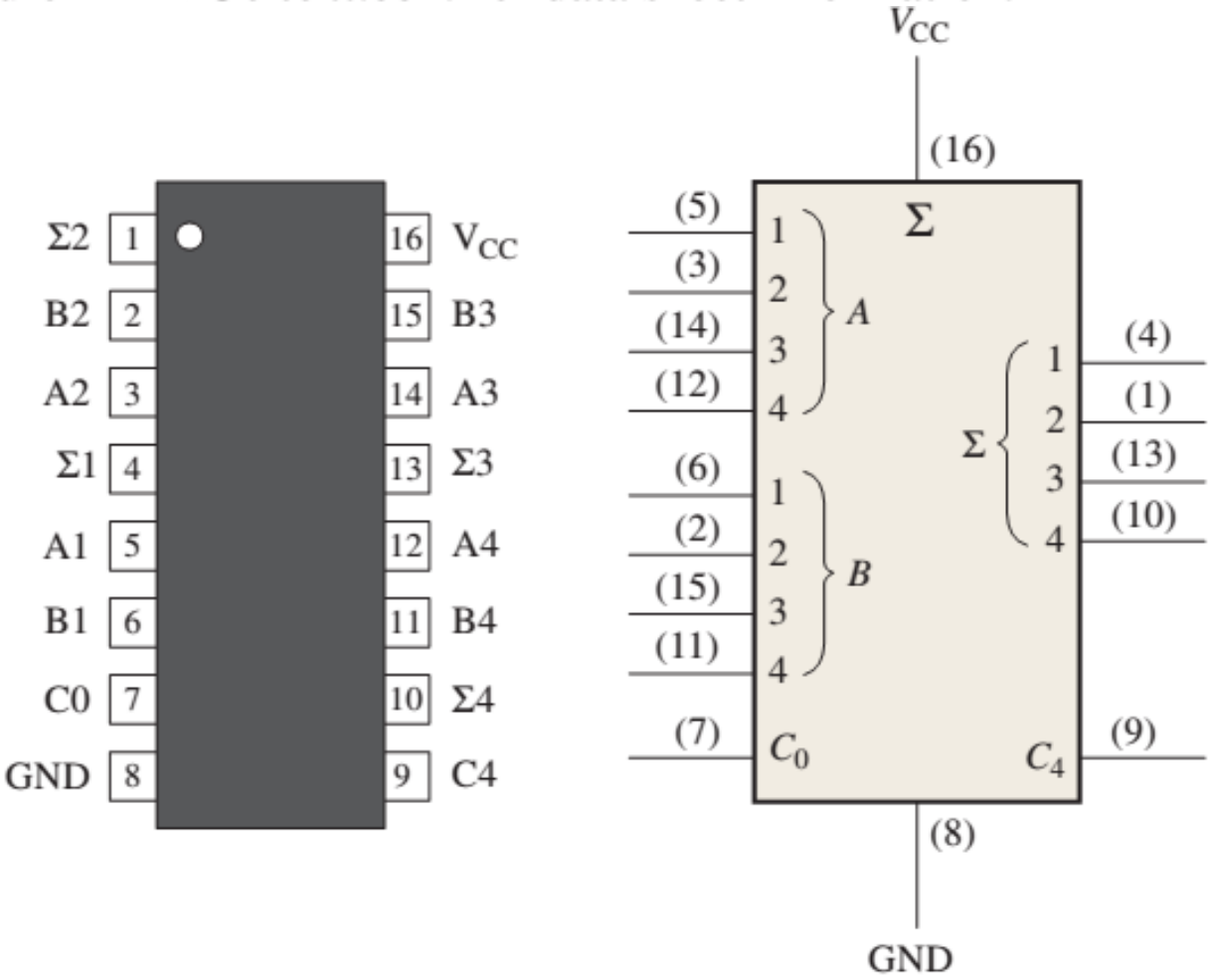




(b) Logic symbol

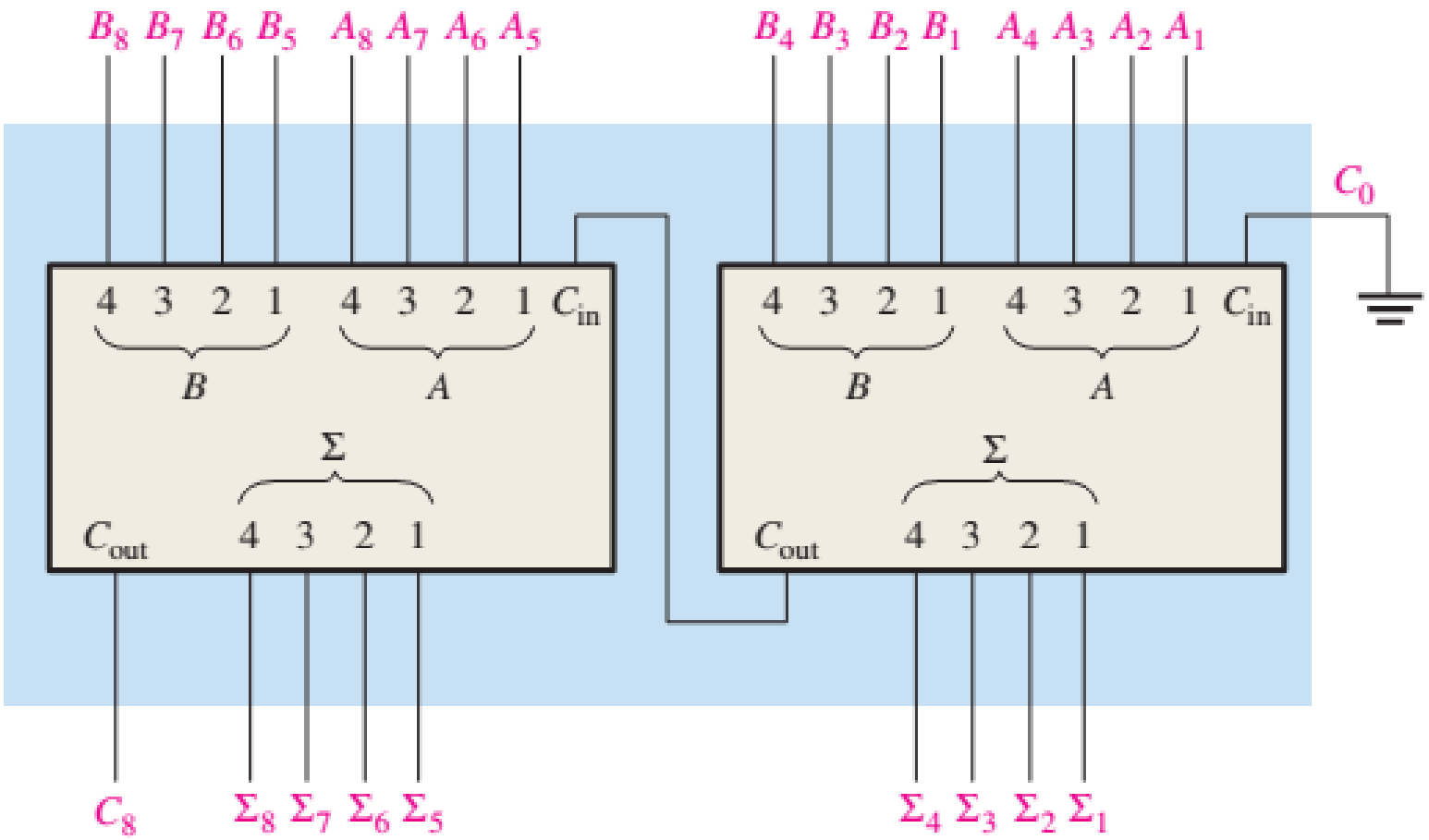
4-BIT PARALLEL ADDER

Fixed-Function Device The 74HC283 and the 74LS283 are 4-bit parallel adders with identical package pin configurations. The logic symbol and package pin configuration are shown in Figure Go to *ti.com* for data sheet information.



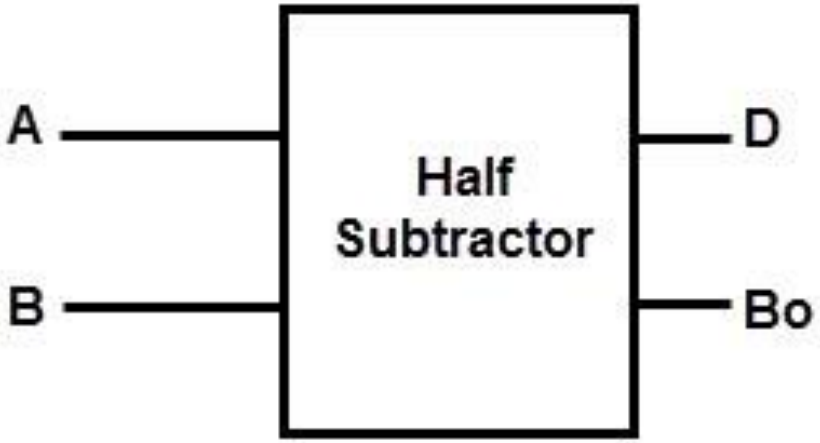
(a) Pin diagram (b) Logic symbol

The 74HC283/74LS283 4-bit parallel adder.



Cascading of two 4-bit adders to form an 8-bit adder.

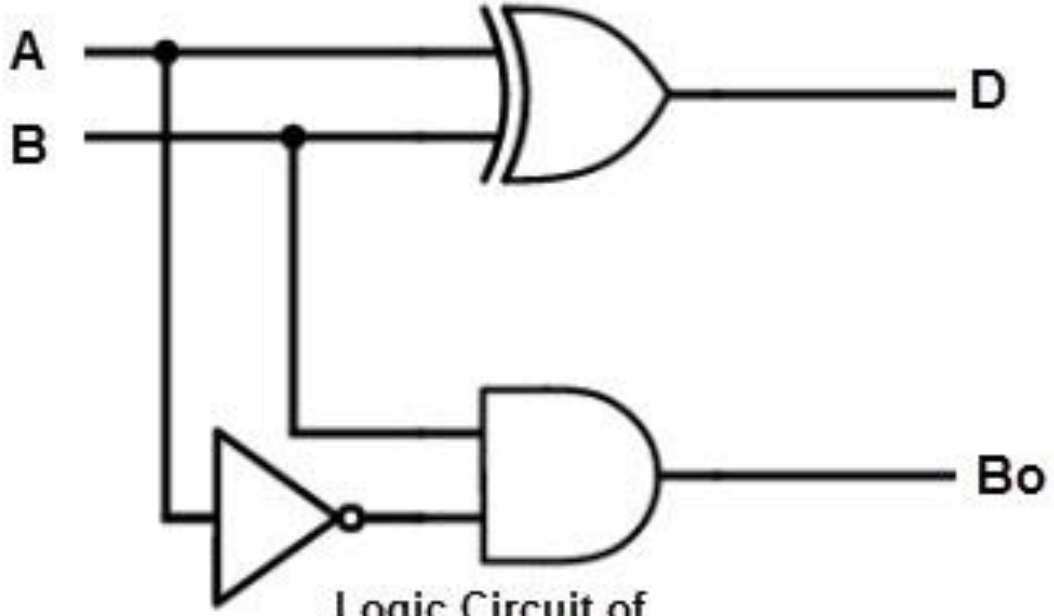
Half Subtractor



Block Diagram

A	B	D	B_o
0	0	**0**	**0**
0	1	**1**	**1**
1	0	**1**	**0**
1	1	**0**	**0**

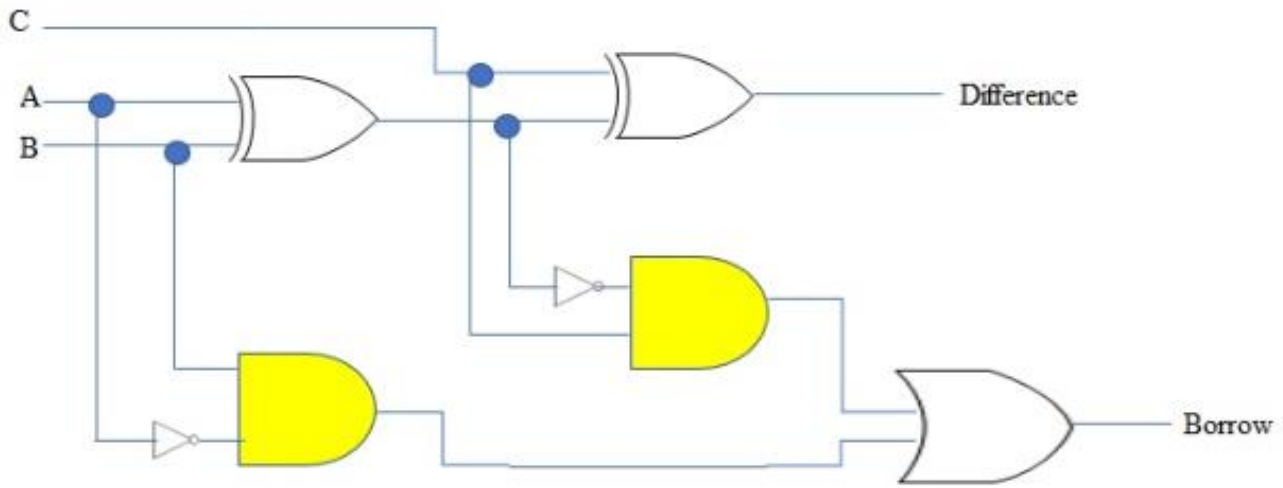
Truth Table



Logic Circuit of
Half Subtractor

Full Subtractor:

INPUT			OUTPUT	
A	B	C	DIFF.	BORROW
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

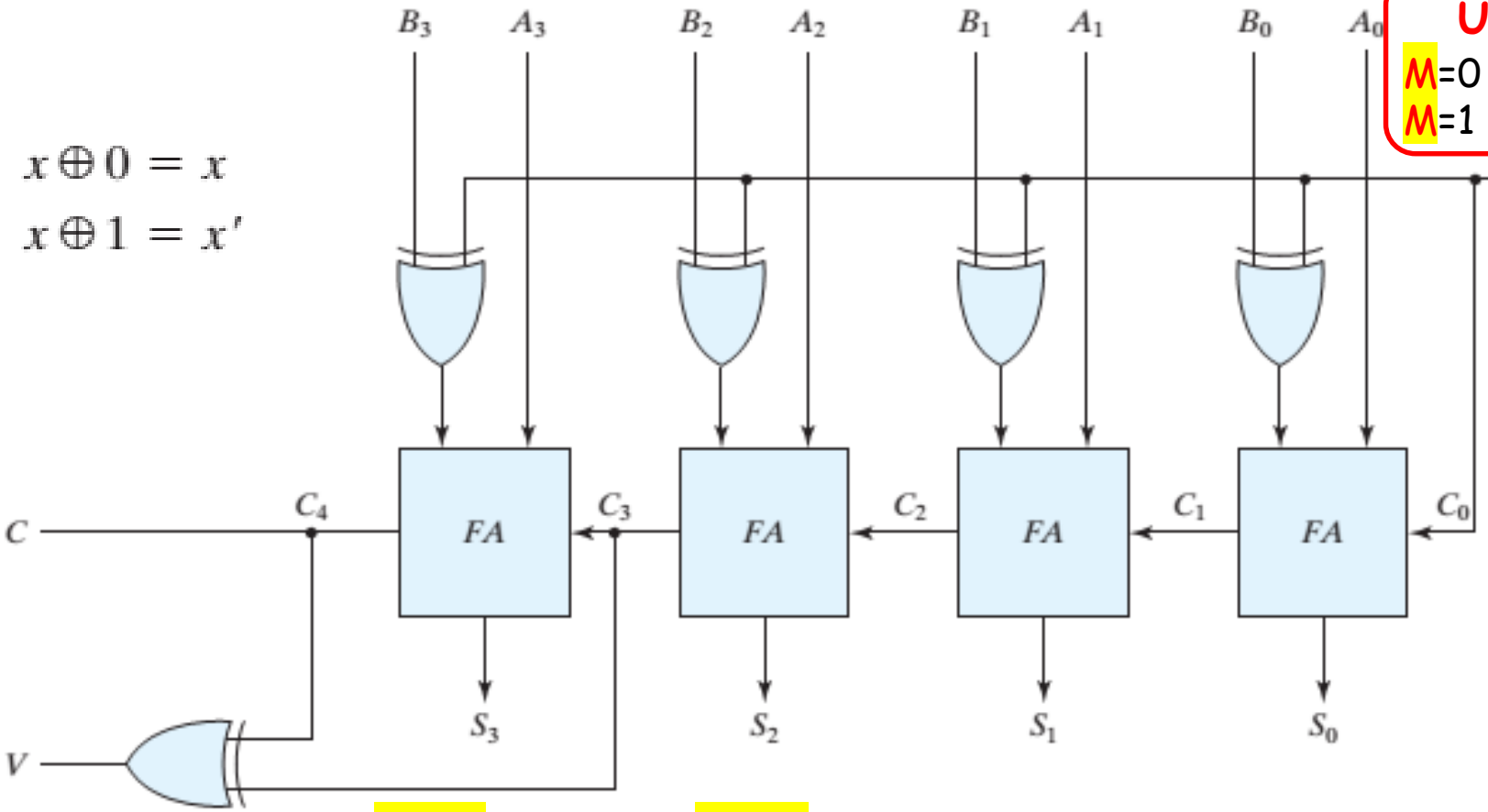


Four-bit adder–subtractor (with overflow detection)

$$x \oplus 0 = x$$
$$x \oplus 1 = x'$$

Unsigned NO.
M=0 ➔ Adder
M=1 ➔ Subtractor

Signed NO
M=0



Unsigned NO. ➔ C bit detects a **carry** after addition or a **borrow** after subtraction.
Signed NO. ➔ V bit detects an **overflow**.
If V = 0 after an addition or subtraction, then no overflow occurred and then 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.

signed numbers

V=1 C4 C3

carries:	0 1
+70	0 1000110
+80	0 1010000
+150	1 0010110

binary +127 to binary −128

V=1 C4 C3

carries:	1 0
−70	1 0111010
−80	1 0110000
−150	0 1101010

-128 64 32 16 8 4 2 1
0 1 0 0 0 1 1 0 =+70

-128 64 32 16 8 4 2 1
0 1 0 1 0 0 0 0 =+80

-128 64 32 16 8 4 2 1
1 1 0 1 1 1 0 0 =-70

-128 64 32 16 8 4 2 1
1 0 1 1 0 0 0 0 =-80

C

-256 128 64 32 16 8 4 2 1
0 1 0 0 1 0 1 1 0

+127 + 16 + 4 + 2 = +150

V=1

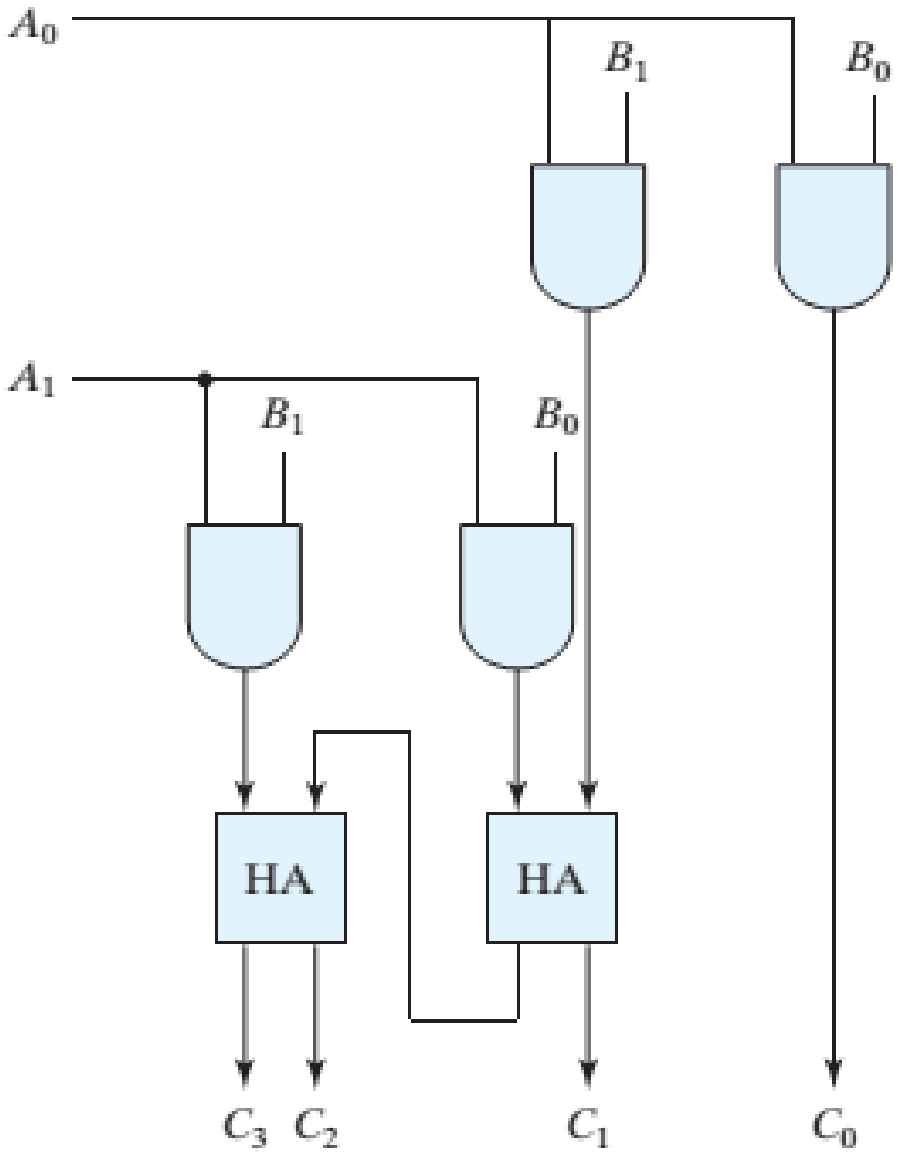
C

-256 128 64 32 16 8 4 2 1
1 0 1 1 0 1 0 1 0

-256 + 64 + 32 + 8 + 2 = -150

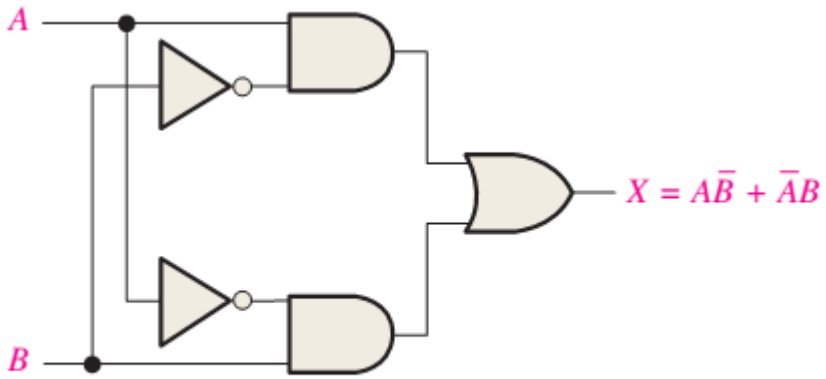
V=1

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



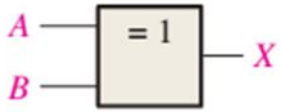
Two-bit by two-bit binary multiplier

Exclusive-OR Logic



(a) Logic diagram

(b) ANSI distinctive shape symbol



(c) ANSI rectangular outline symbol

$$X = A \oplus B$$

$$X = A\overline{B} + \overline{A}B$$

Truth table for an exclusive-OR.

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

$$x \oplus 0 = x$$

$$x \oplus 1 = x'$$

$$x \oplus x = 0$$

$$x \oplus x' = 1$$

$$x \oplus y' = x' \oplus y = (x \oplus y)'$$

$$A \oplus B = B \oplus A$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$$

$$X \oplus 0 = X$$

$$X \oplus 1 = \overline{X}$$

$$X \oplus X = 0$$

$$X \oplus \overline{X} = 1$$

$$\overline{X} \oplus \overline{X} = 0$$

$$\overline{X} \oplus \overline{X} = X \oplus X$$

$$A \oplus AB = A\overline{B}$$

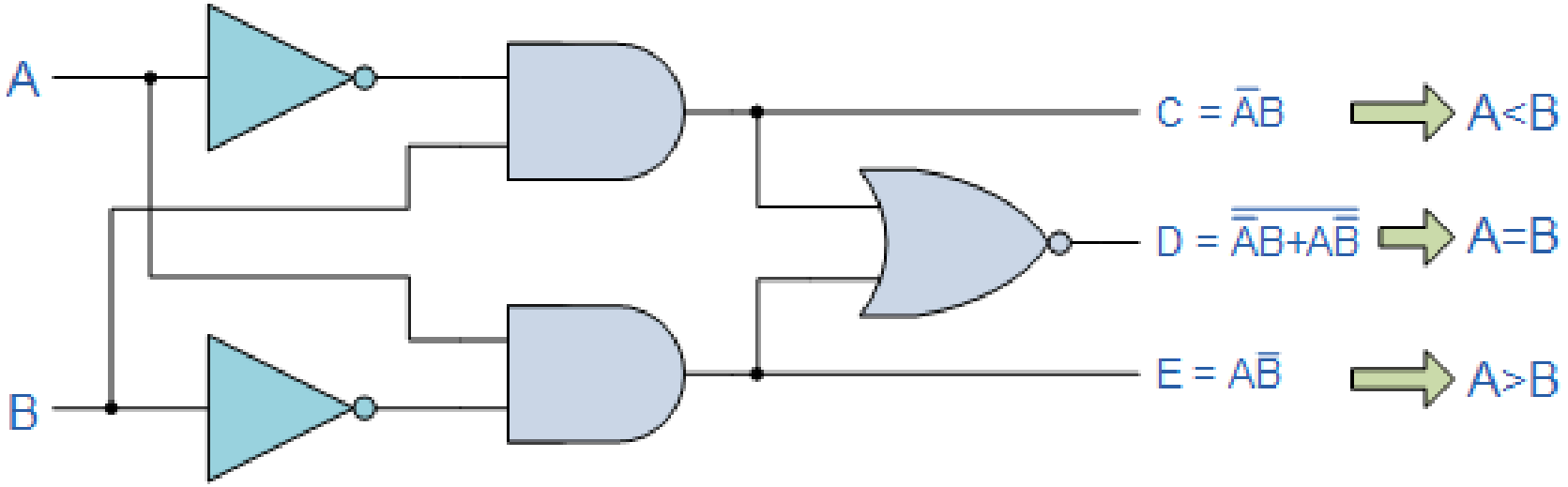
$$A \oplus A\overline{B} = AB$$

$$A \oplus \overline{A}B = A + B$$

$$(A \oplus B).(A \oplus C) = \overline{A}BC + A\overline{B}\overline{C}$$

Comparators

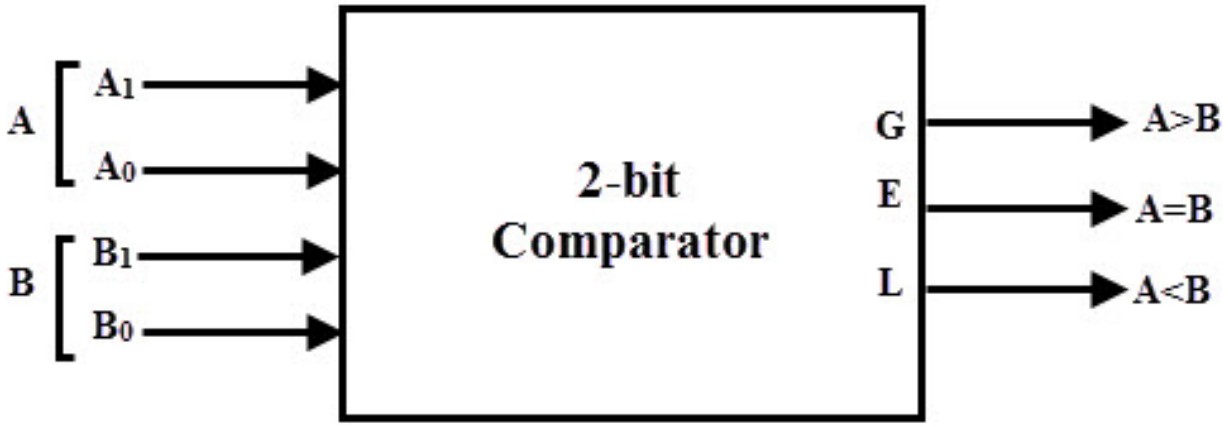
Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



2-Bit Comparator

A 2-bit comparator compares two binary numbers, each of two bits and produces their relation such as one number is equal or greater than or less than the other. The figure below shows the block diagram of a two-bit comparator which has four inputs and three outputs.

The first number A is designated as A = A1A0 and the second number is designated as B = B1B0. This comparator produces three outputs as G (G = 1 if A>B), E (E = 1, if A = B) and L (L = 1 if A<B).



The truth table of this comparator is shown below which depicting various input and output states.

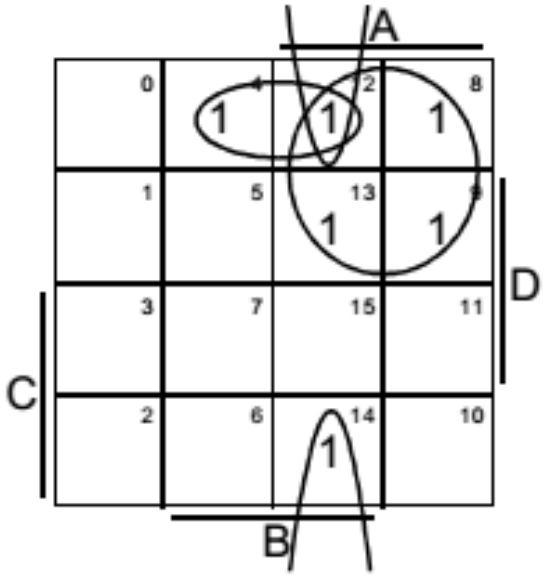
Inputs				Outputs		
Number P		Number Q		P > Q	P = Q	P < Q
A	B	C	D	Y1	Y2	Y3
0	0	0	0	***0***	***1***	***0***
0	0	0	1	***0***	***0***	***1***
0	0	1	0	***0***	***0***	***1***
0	0	1	1	***0***	***0***	***1***
0	1	0	0	***1***	***0***	***0***
0	1	0	1	***0***	***1***	***0***
0	1	1	0	***0***	***0***	***1***
0	1	1	1	***0***	***0***	***1***
1	0	0	0	***1***	***0***	***0***
1	0	0	1	***1***	***0***	***0***
1	0	1	0	***0***	***1***	***0***
1	0	1	1	***0***	***0***	***1***
1	1	0	0	***1***	***0***	***0***
1	1	0	1	***1***	***0***	***0***
1	1	1	0	***1***	***0***	***0***
1	1	1	1	***0***	***1***	***0***

$$Y1 = \bar{A}\,B\,\bar{C}\,\bar{D} \vee A\,\bar{B}\,\bar{C}\,\bar{D} \vee A\,\bar{B}\,\bar{C}\,D \vee A\,B\,\bar{C}\,\bar{D} \vee A\,B\,\bar{C}\,D \vee A\,B\,C\,\bar{D}$$

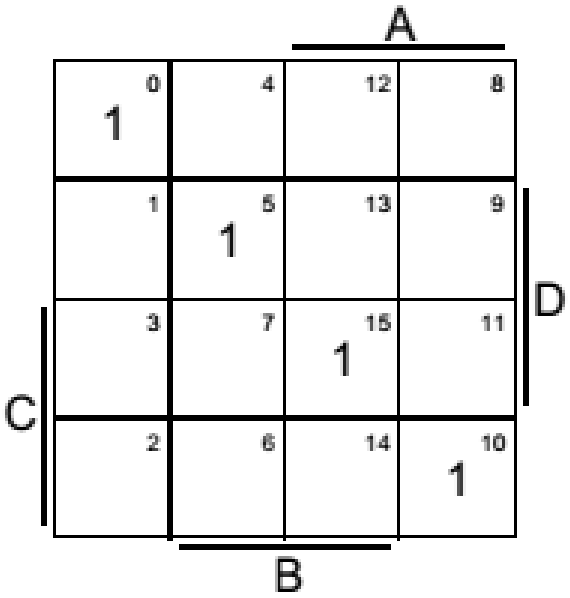
$$Y2 = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D} \vee \bar{A}\,B\,\bar{C}\,D \vee A\,\bar{B}\,C\,\bar{D} \vee A\,B\,C\,D$$

$$Y3 = \bar{A}\,\bar{B}\,\bar{C}\,D \vee \bar{A}\,\bar{B}\,C\,\bar{D} \vee \bar{A}\,\bar{B}\,C\,D \vee \bar{A}\,B\,C\,\bar{D} \vee \bar{A}\,B\,C\,D \vee A\,\bar{B}\,C\,D$$

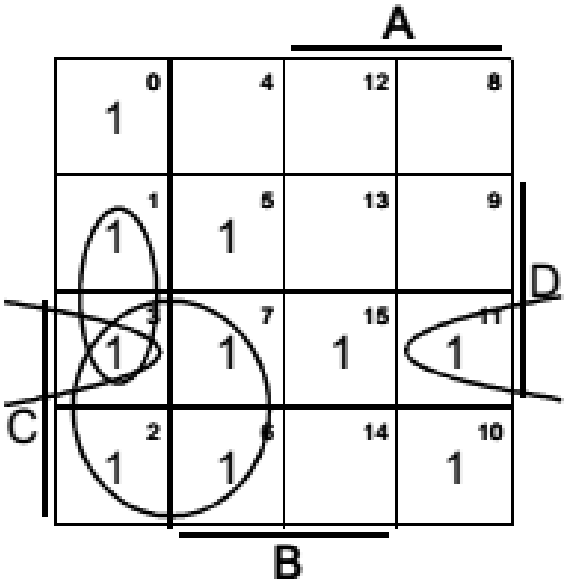
The k-map simplification for the above truth table is as follows.



KV diagram for Y1



KV diagram for Y2

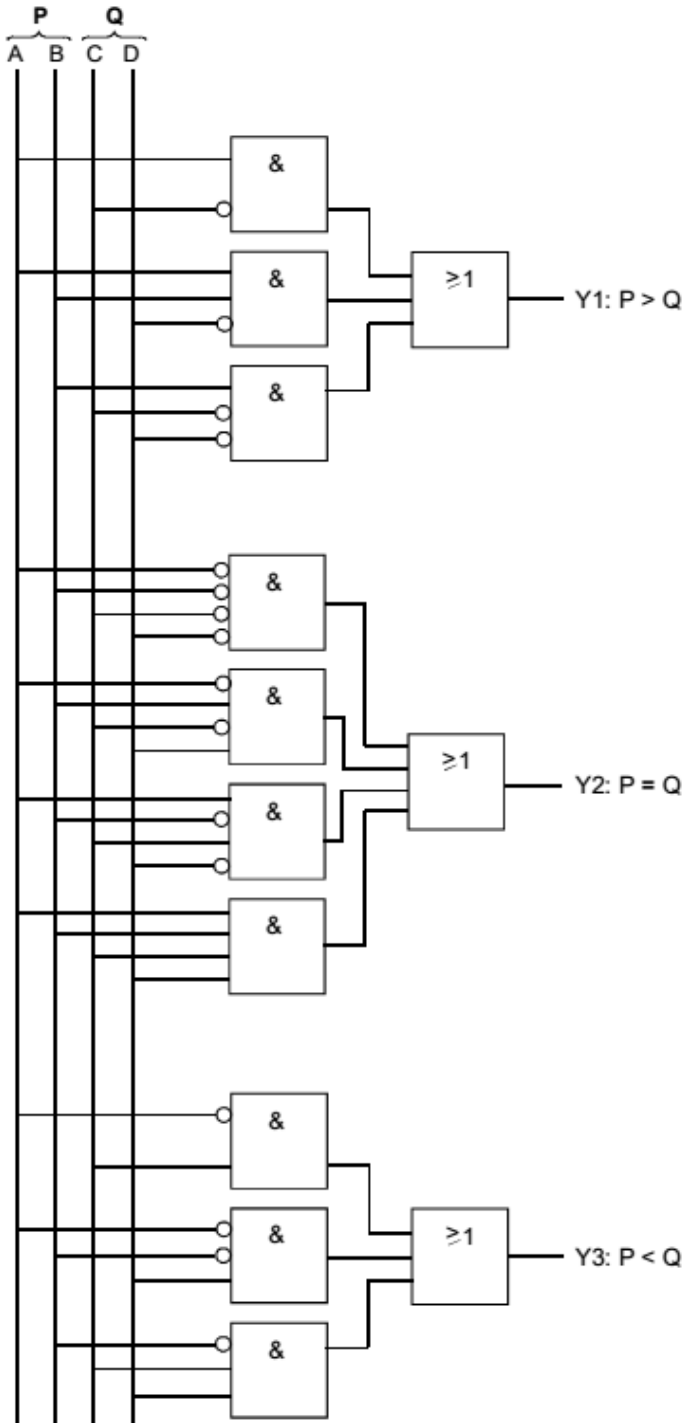


KV diagram for Y3

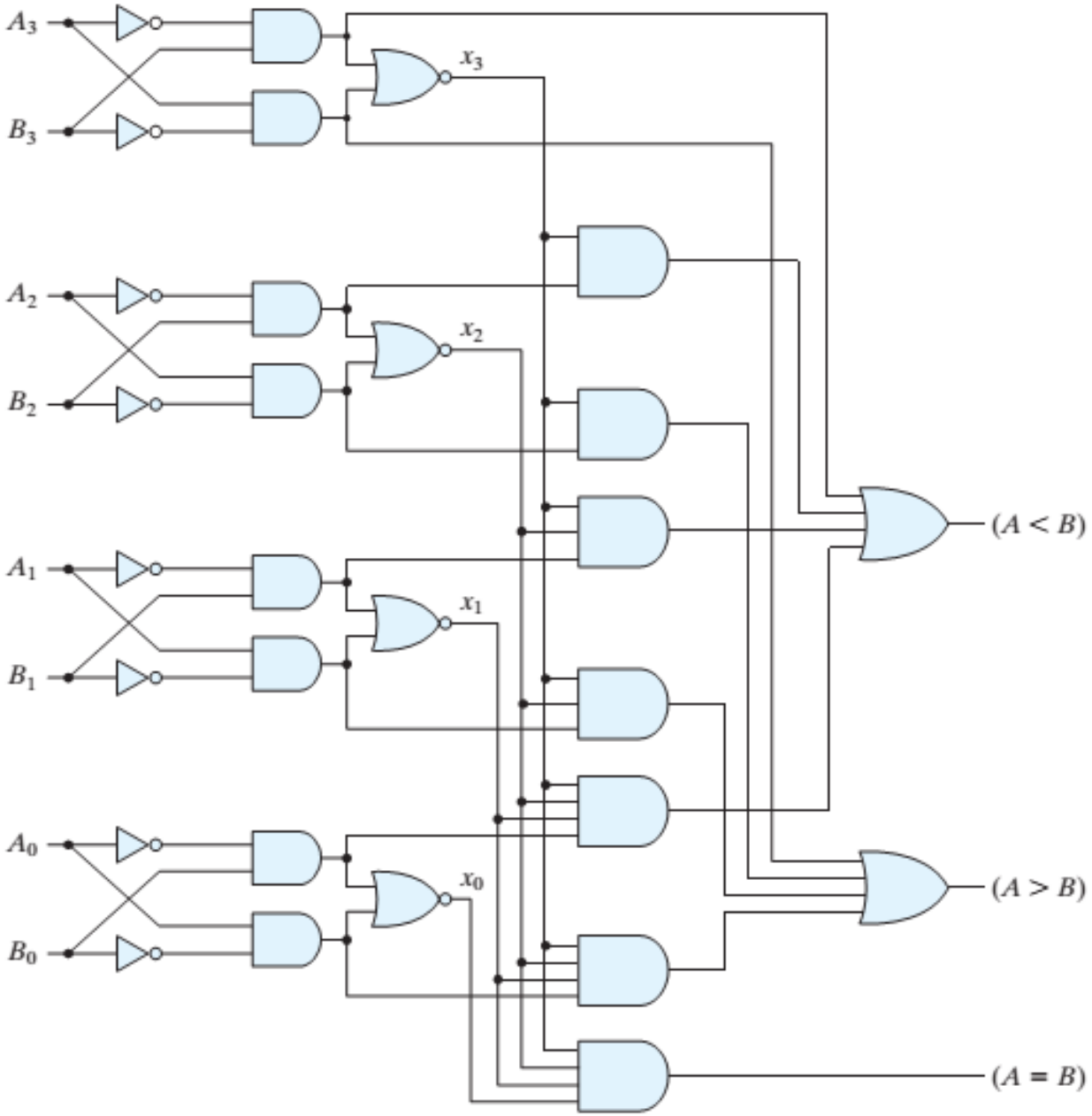
Y1 = A $\overline{C}$ ∨ A B $\overline{D}$ ∨ B $\overline{C}$ $\overline{D}$

No minimization is possible with Y2!

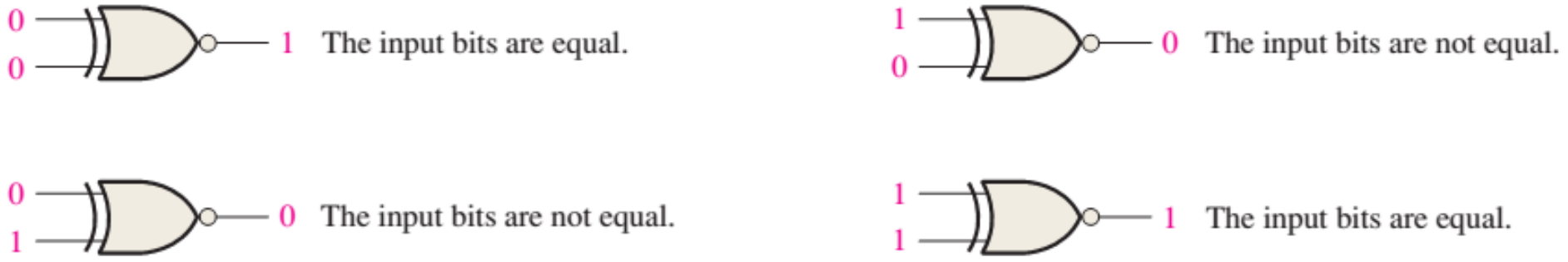
Y3 = $\overline{A}$ C ∨ $\overline{A}$ $\overline{B}$ D ∨ $\overline{B}$ C D


P
Q
A
B
C
D
&
&
&
≥1
Y1: P > Q
&
&
&
&
≥1
Y2: P = Q
&
&
&
&
≥1
Y3: P < Q

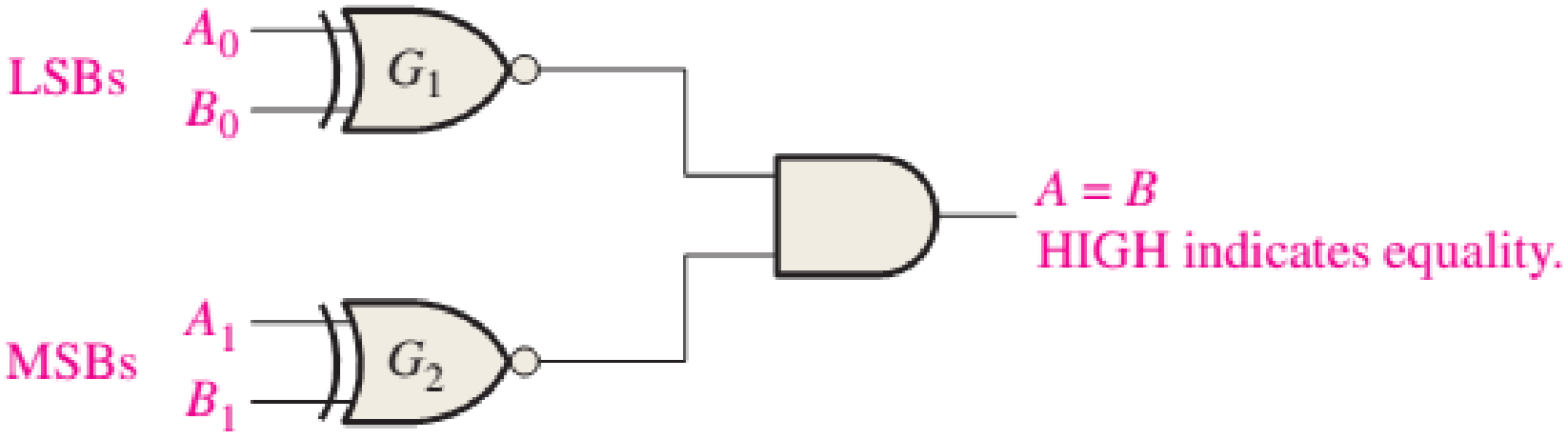
$$A = A_3\, A_2\, A_1\, A_0$$
$$B = B_3\, B_2\, B_1\, B_0$$



Four-bit magnitude comparator

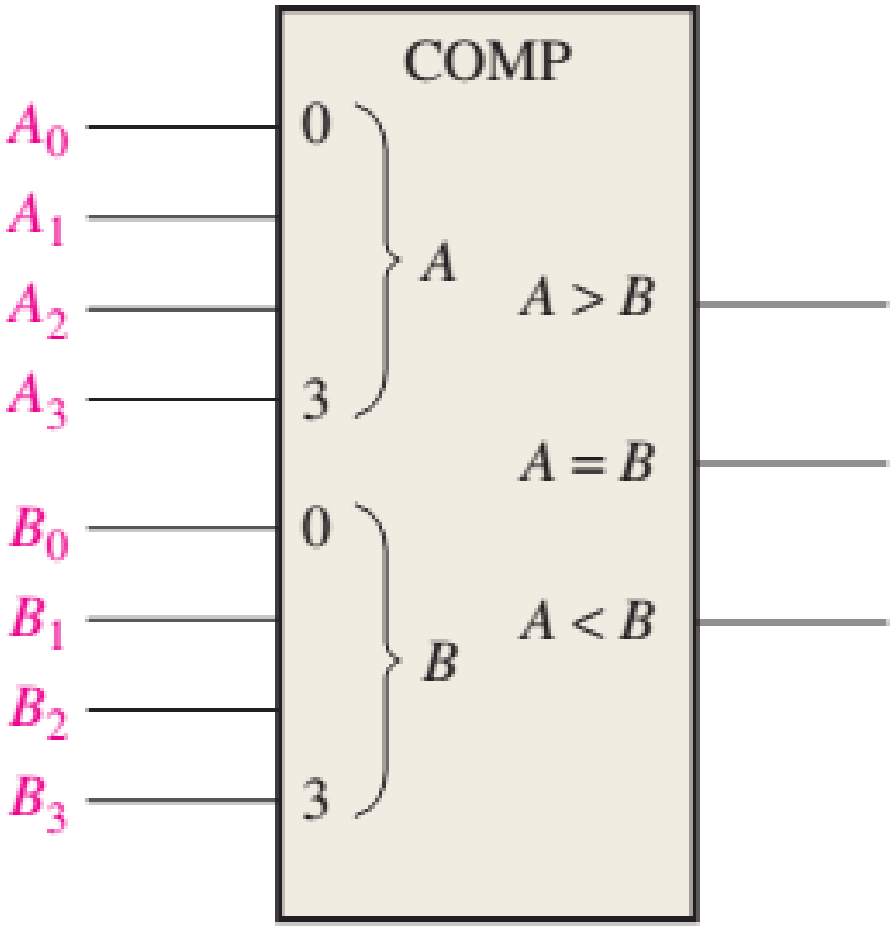


Basic comparator operation.

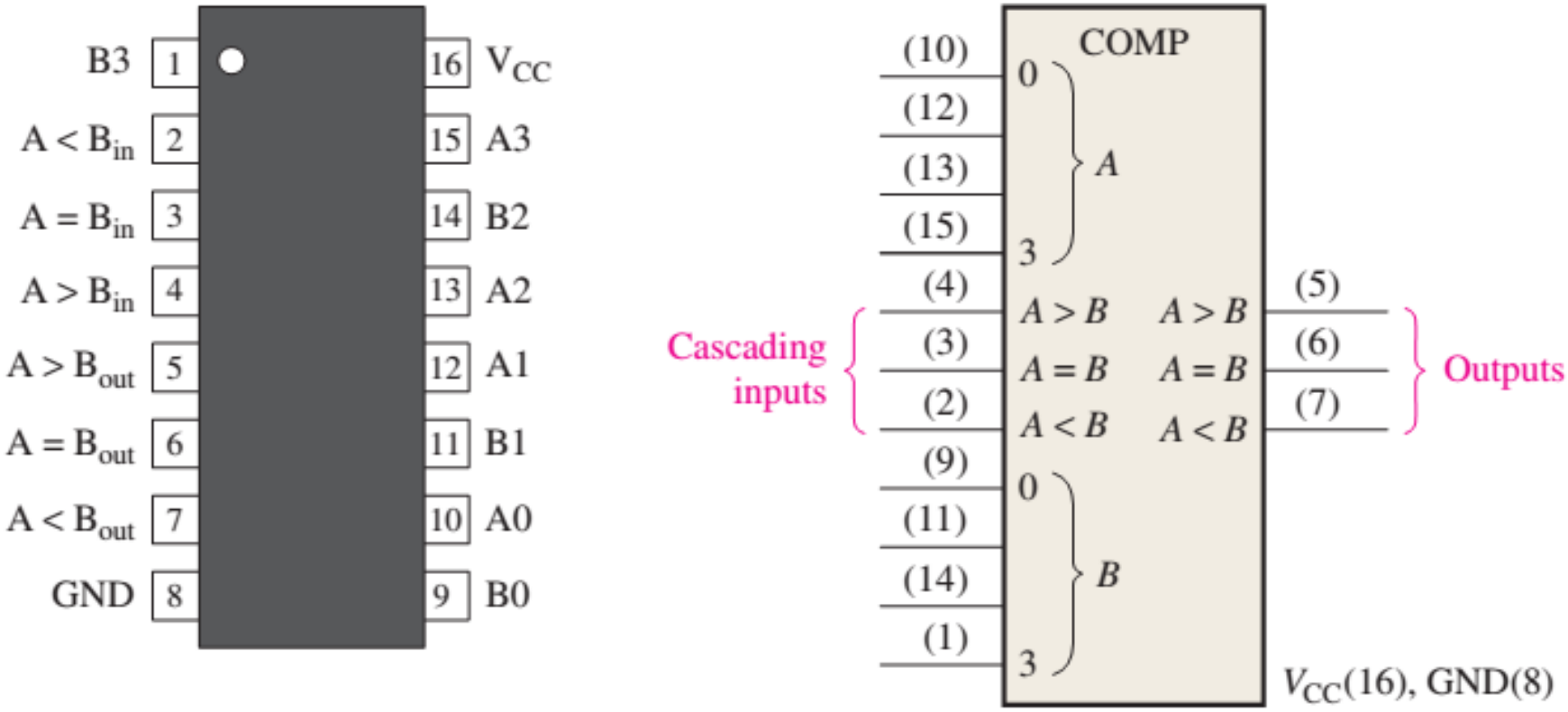


General format: Binary number $A \rightarrow A_1A_0$
Binary number $B \rightarrow B_1B_0$

Logic diagram for equality comparison of two 2-bit numbers.



Logic symbol for a 4-bit comparator with inequality indication.

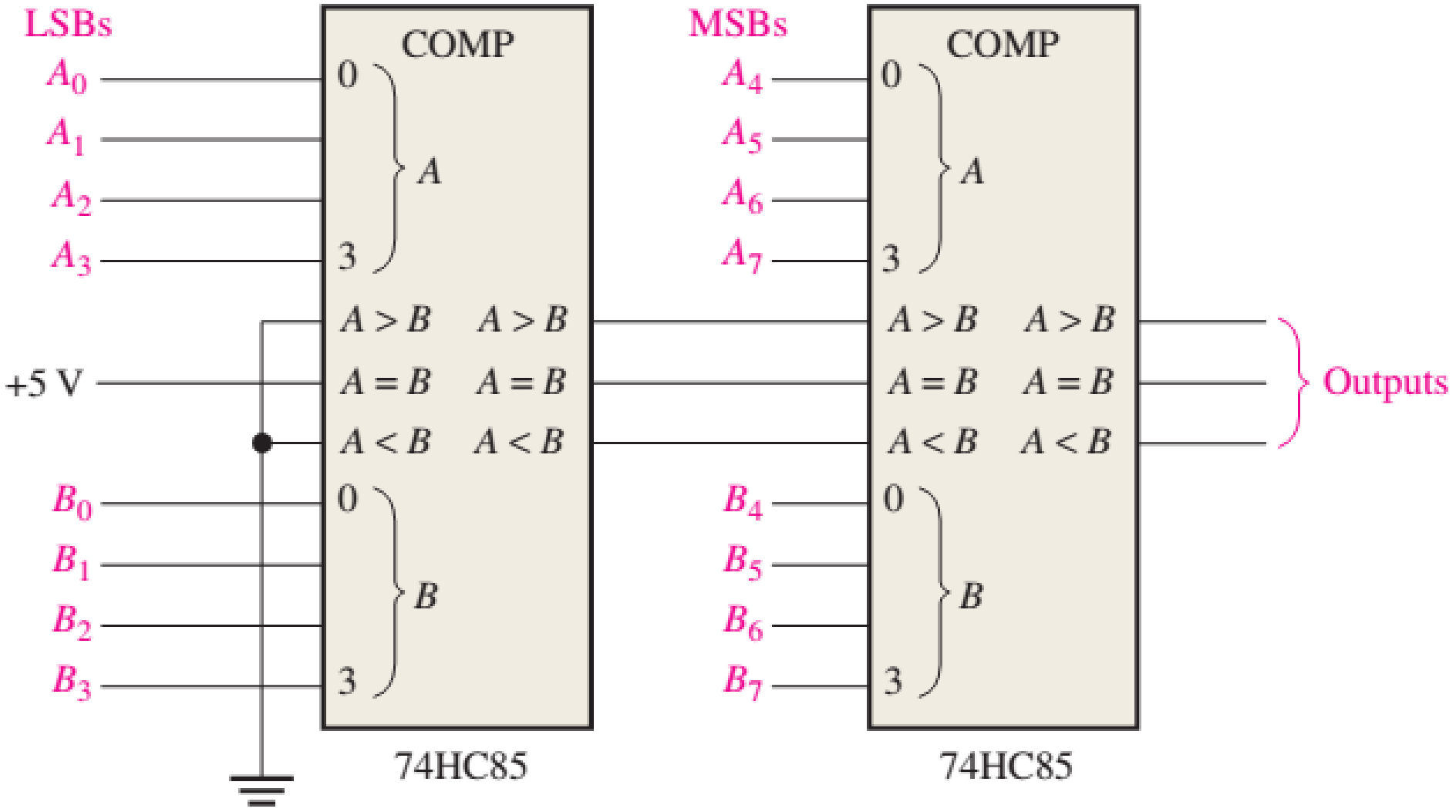


(a) Pin diagram

(b) Logic symbol

The 74HC85/74LS85 4-bit magnitude comparator.

Use 74HC85 comparators to compare the magnitudes of two 8-bit numbers. Show the comparators with proper interconnections.



An 8-bit magnitude comparator using two 74HC85s.


Thank
you!