DIGITAL LOGIC DESIGN EE(1005)

LECTURE-19

Chapter No:6
Function of combinational logic

TOPICS

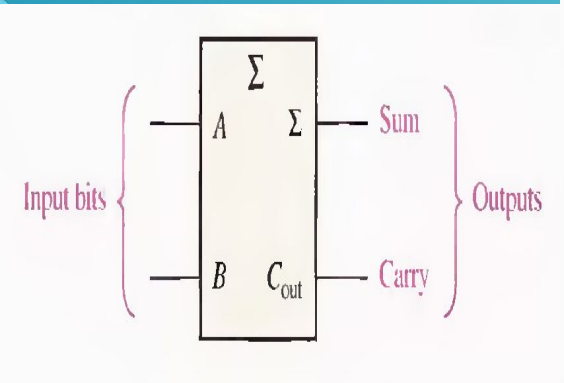
1. Half and Full Adders
2. Parallel Binary Adders
3. Comparators
4. Decoders
5. Encoders
6. Code Converters
7. Multiplexers (Data Selectors)
8. Demultiplexers

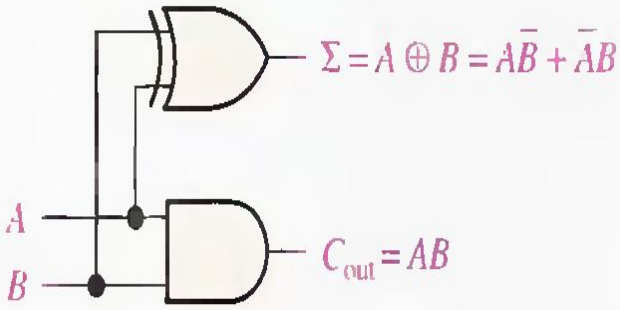
BASIC ADDERS

The Half-Adder

The operations are performed by a logic circuit called a **half-adder.**

The half-adder accepts two binary digits on its inputs and produces two binary digits on its outputs, a sum bit and a carry bit.





A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Σ = sum

C_{out} = output carry

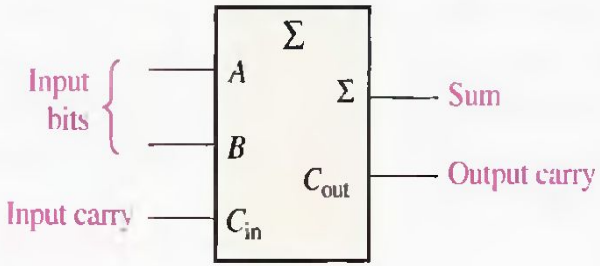
A and *B* = input variables (operands)

$$\Sigma = A \oplus B$$

$$C_{out} = AB$$

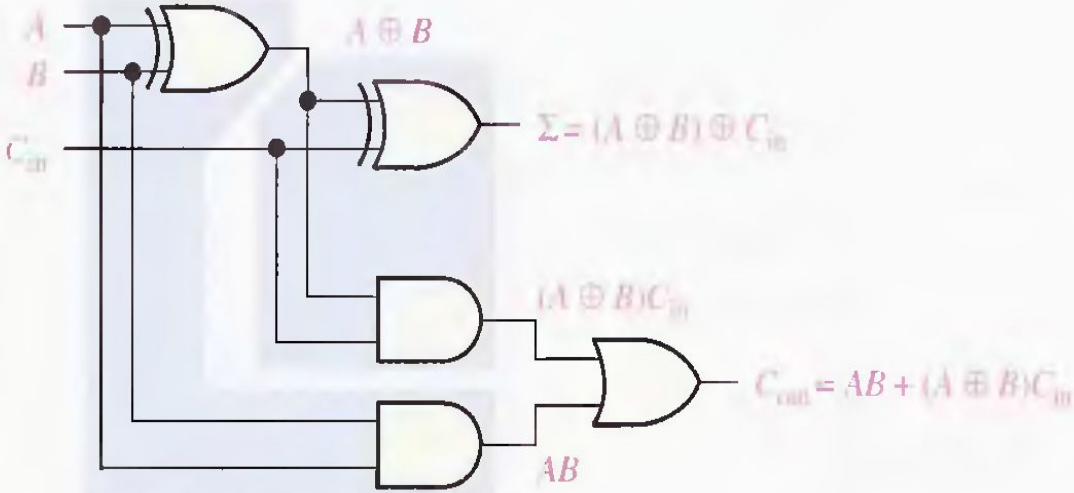
The Full-Adder

The full-adder accepts two input bits and an input carry and generates a sum output and an output carry.



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

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



A	B	C_{in}	C_{out}	Σ
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

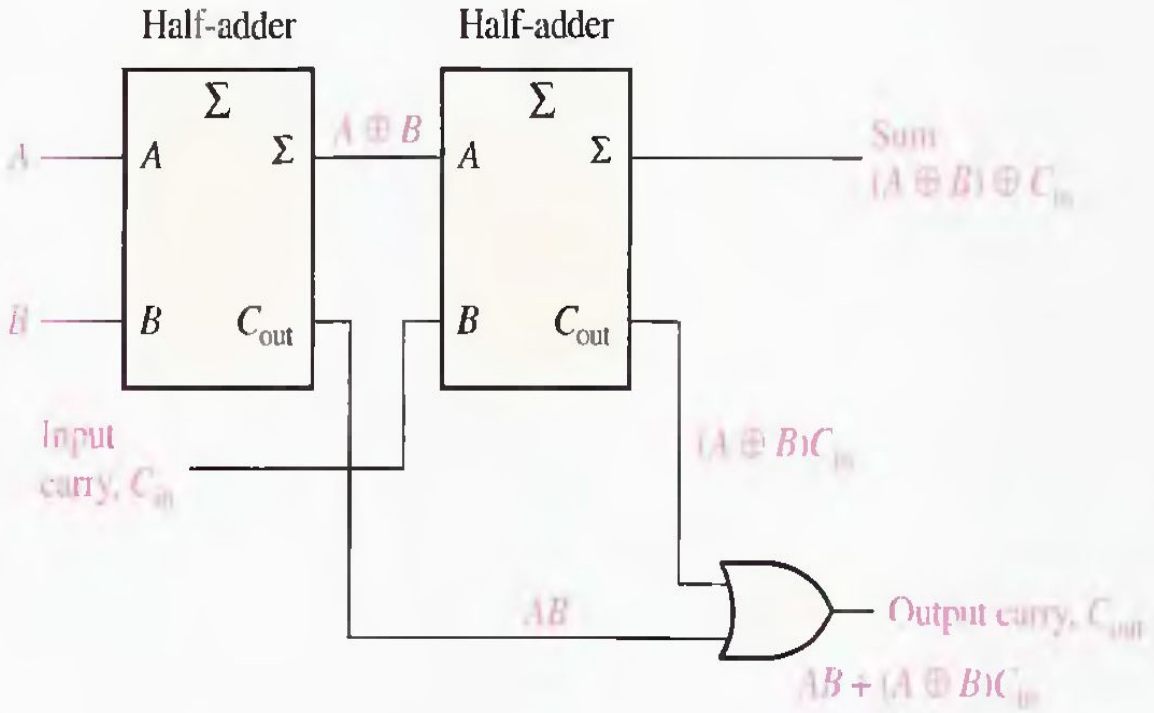
C_{in} = input carry, sometimes designated as *CI*

C_{out} = output carry, sometimes designated as *CO*

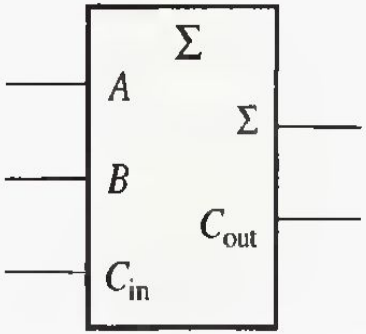
Σ = sum

A and B = input variables (operands)

The Full-Adder



(a) Arrangement of two half-adders to form a full-adder

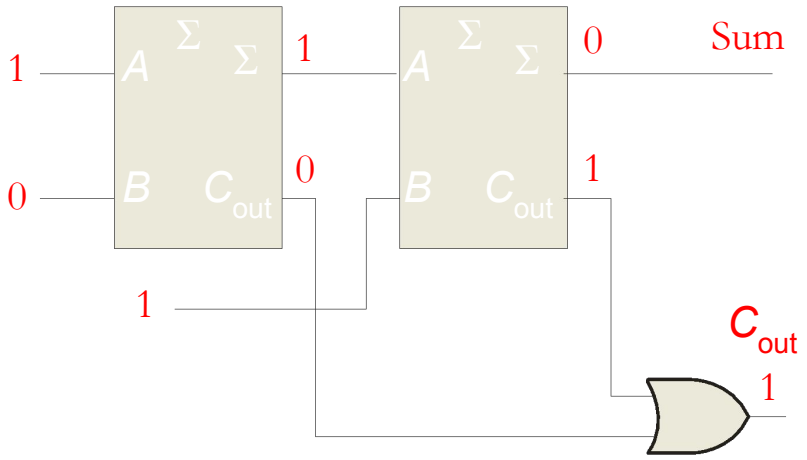


(b) Full-adder logic symbol

Full-Adder

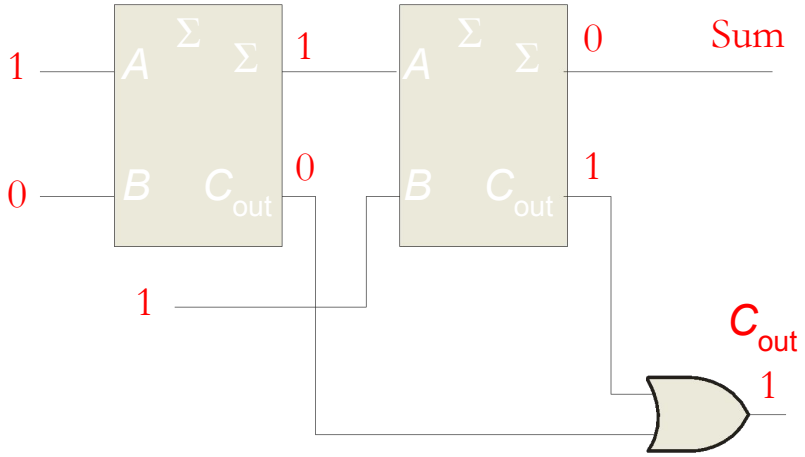
Example

Solution



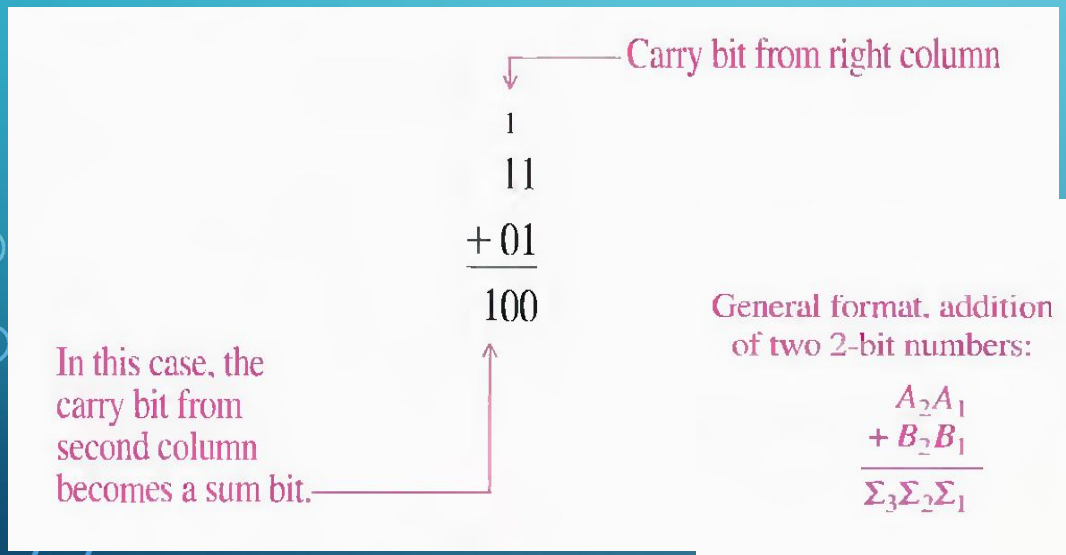
Full-Adder

Inputs			Outputs	
A	B	C_{in}	C_{out}	Σ
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



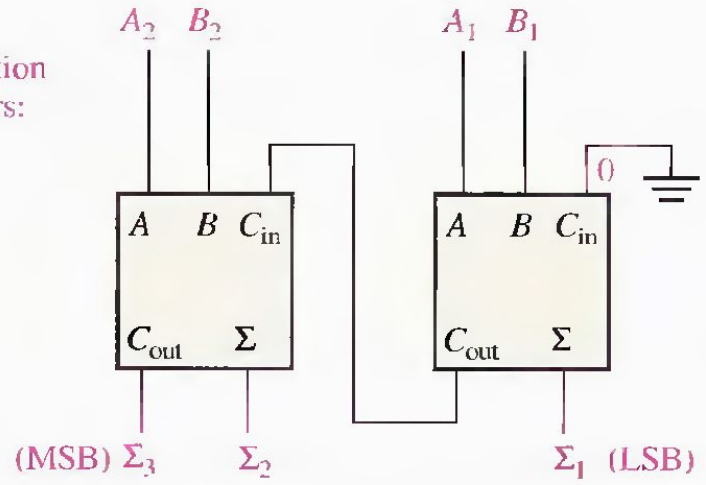
PARALLEL BINARY ADDERS

Two or more full-adders are connected to form parallel binary adders. A single full-adder is capable of adding two, I-bit numbers and an input carry. To add binary numbers with more than one bit, you must use additional full-adders.



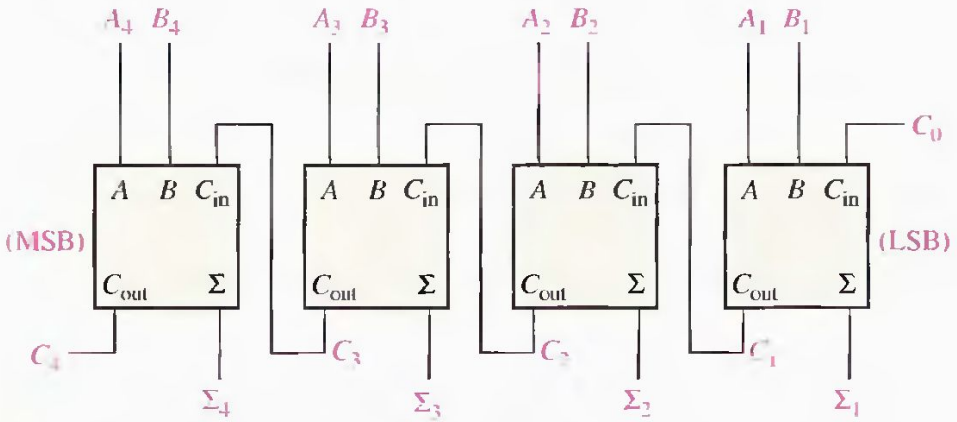
General format, addition of two 2-bit numbers:

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

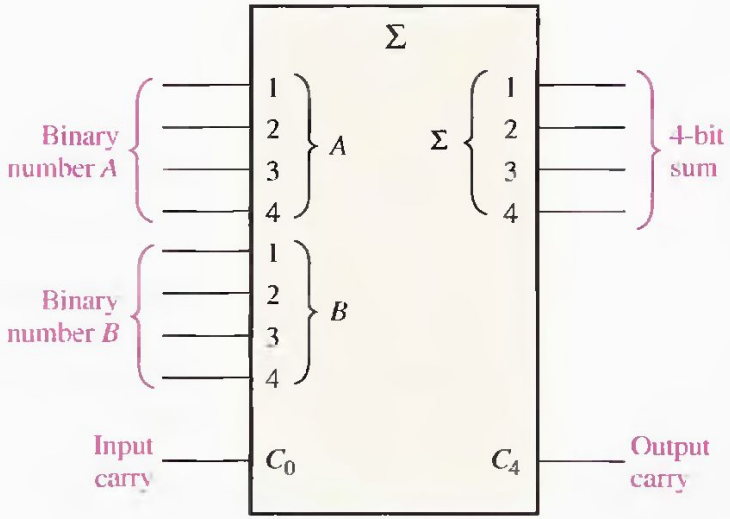


Four-Bit Parallel Adders

A group of four bits is called a nibble. A basic 4-bit parallel adder is implemented with four full-adder stages .



(a) Block diagram

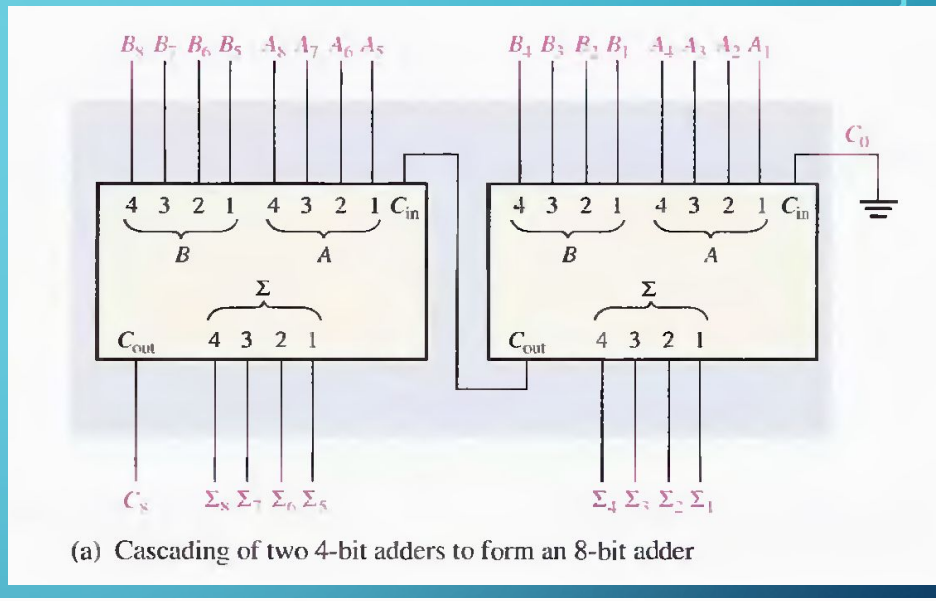


(b) Logic symbol

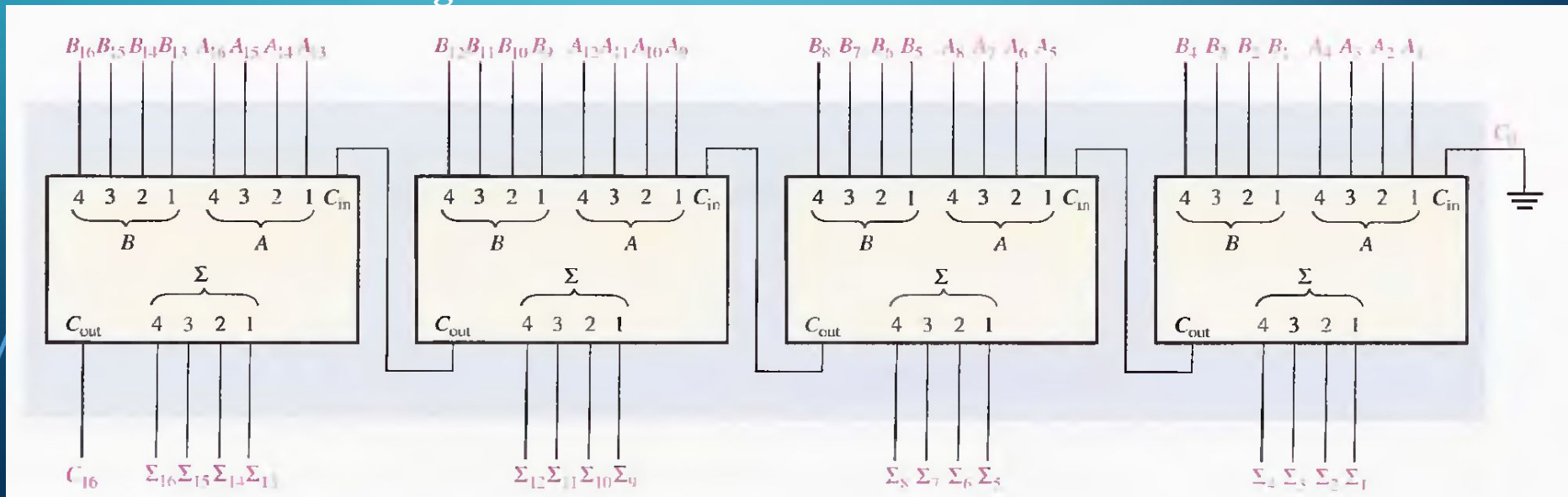
C_{n-1}	A_n	B_n	Σ_n	C_n
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Adder Expansion

Cascading: The 4-bit parallel adder can be expanded to handle the addition of two 8-bit numbers by using two 4-bit adders. The carry input of the low-order adder (Co) is connected to ground because there is no carry into the least significant bit position, and the carry output of the low-order adder is connected to the carry input of the high-order adder, as shown in Figure . This process is known as ***cascading.***



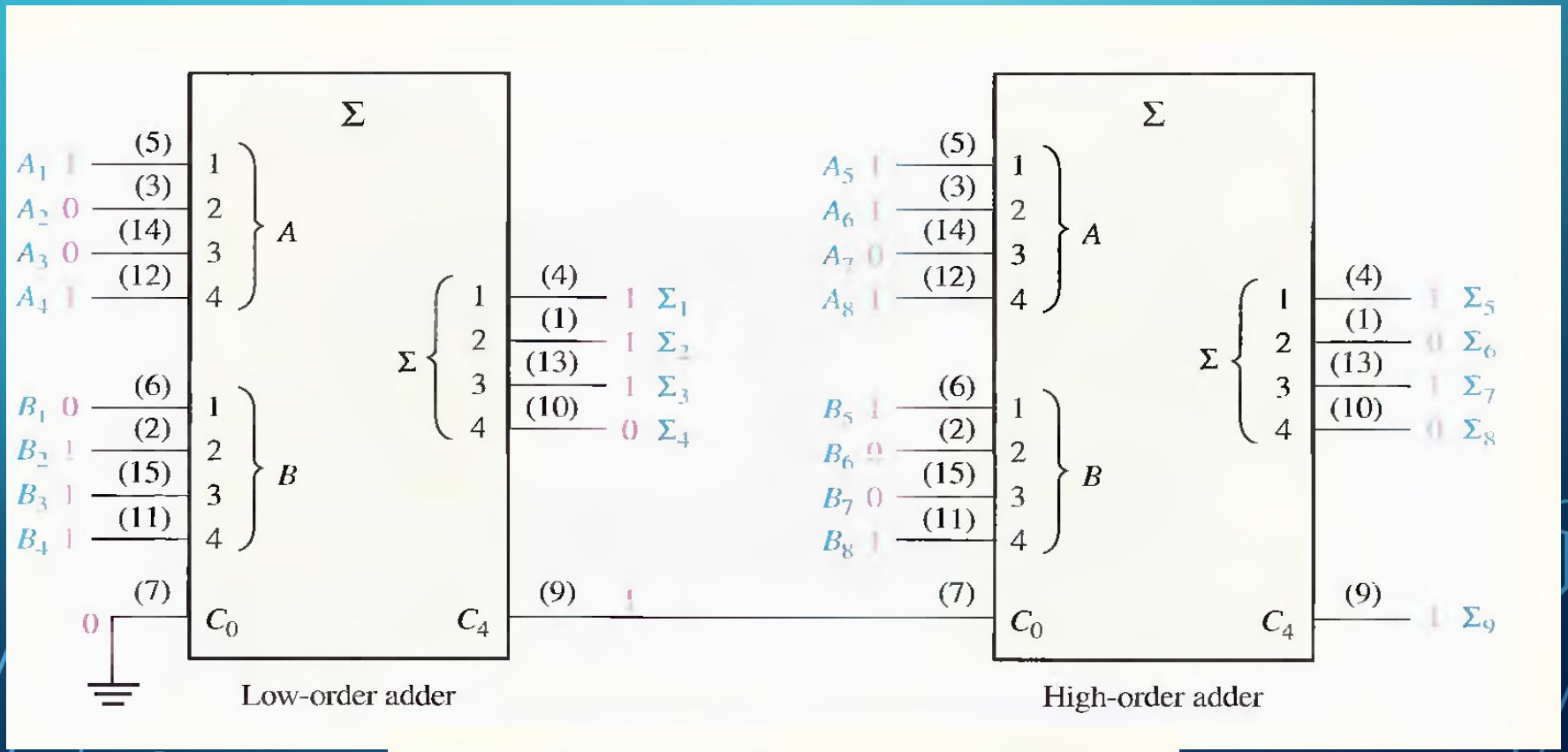
(a) Cascading of two 4-bit adders to form an 8-bit adder



EXAMPLE

Show how two 74LS283 adders can be connected to form an 8-bit parallel adder. Show output bits for the following 8-bit input numbers:

$$A_8A_7A_6A_5A_4A_3A_2A_1 = 10111001 \quad \text{and} \quad B_8B_7B_6B_5B_4B_3B_2B_1 = 10011110$$



$$\Sigma_9\Sigma_8\Sigma_7\Sigma_6\Sigma_5\Sigma_4\Sigma_3\Sigma_2\Sigma_1 = 101010111$$

Use the 4-bit parallel adder truth table (Table 6–3) to find the sum and output carry for the addition of the following two 4-bit numbers if the input carry (C_{n-1}) is 0:

$$A_4A_3A_2A_1 = 1100 \quad \text{and} \quad B_4B_3B_2B_1 = 1100$$

For $n = 1$: $A_1 = 0$, $B_1 = 0$, and $C_{n-1} = 0$. From the 1st row of the table,

$$\Sigma_1 = \mathbf{0} \quad \text{and} \quad C_1 = 0$$

For $n = 2$: $A_2 = 0$, $B_2 = 0$, and $C_{n-1} = 0$. From the 1st row of the table,

$$\Sigma_2 = \mathbf{0} \quad \text{and} \quad C_2 = 0$$

For $n = 3$: $A_3 = 1$, $B_3 = 1$, and $C_{n-1} = 0$. From the 4th row of the table,

$$\Sigma_3 = \mathbf{0} \quad \text{and} \quad C_3 = 1$$

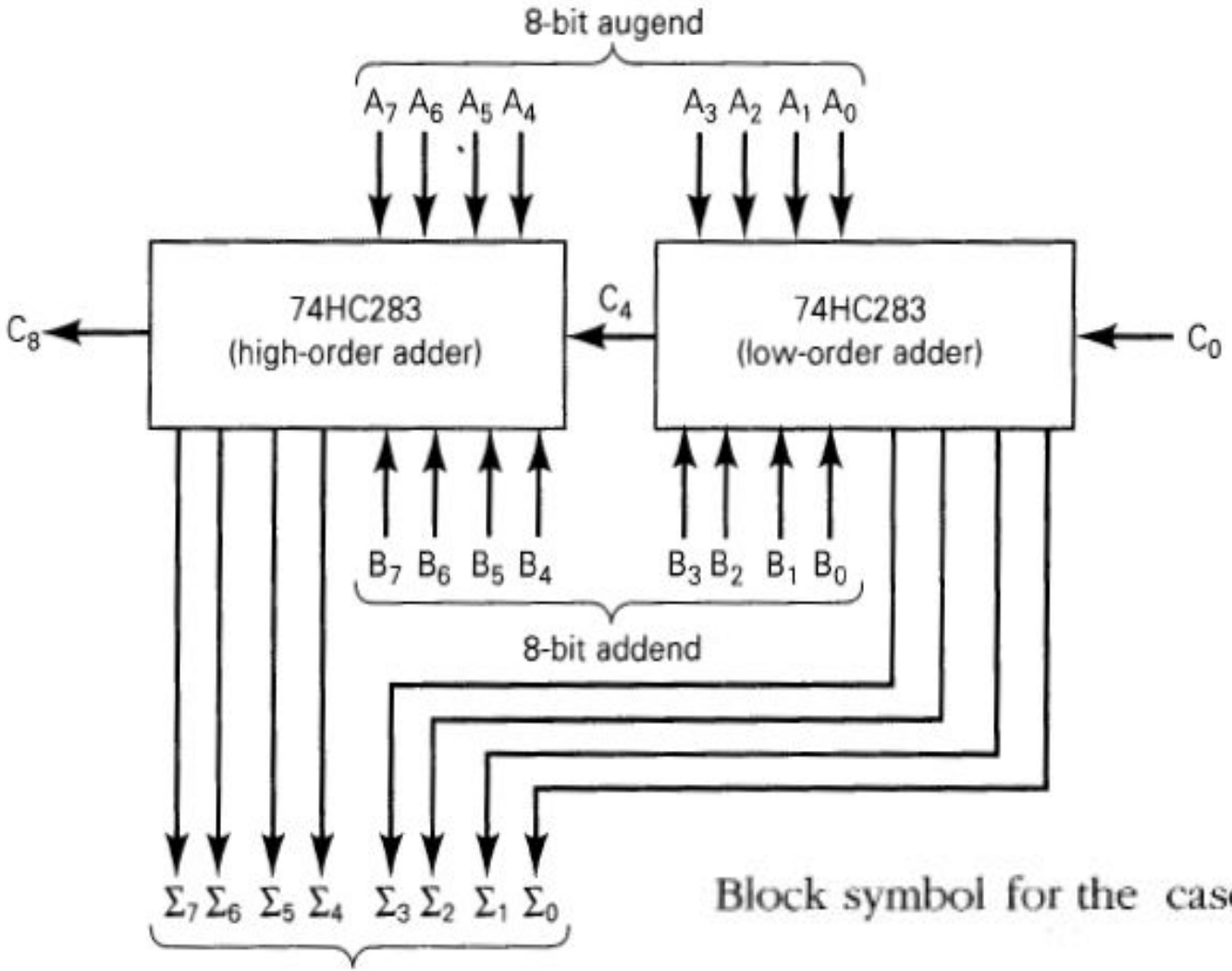
For $n = 4$: $A_4 = 1$, $B_4 = 1$, and $C_{n-1} = 1$. From the last row of the table,

$$\Sigma_4 = \mathbf{1} \quad \text{and} \quad C_4 = \mathbf{1}$$

C_4 becomes the output carry; the sum of 1100 and 1100 is 11000.

Cascading Parallel Adders

Two or more IC adders can be connected together (cascaded) to accomplish the addition of larger binary numbers. Figure shows two 74HC283 adders connected to add two 8-bit numbers $A_7\ A_6\ A_5\ A_4\ A_3\ A_2\ A_1\ A_0$ and $B_7\ B_6\ B_5\ B_4\ B_3\ B_2\ B_1\ B_0$.



Block symbol for the cascading two 74HC283s.

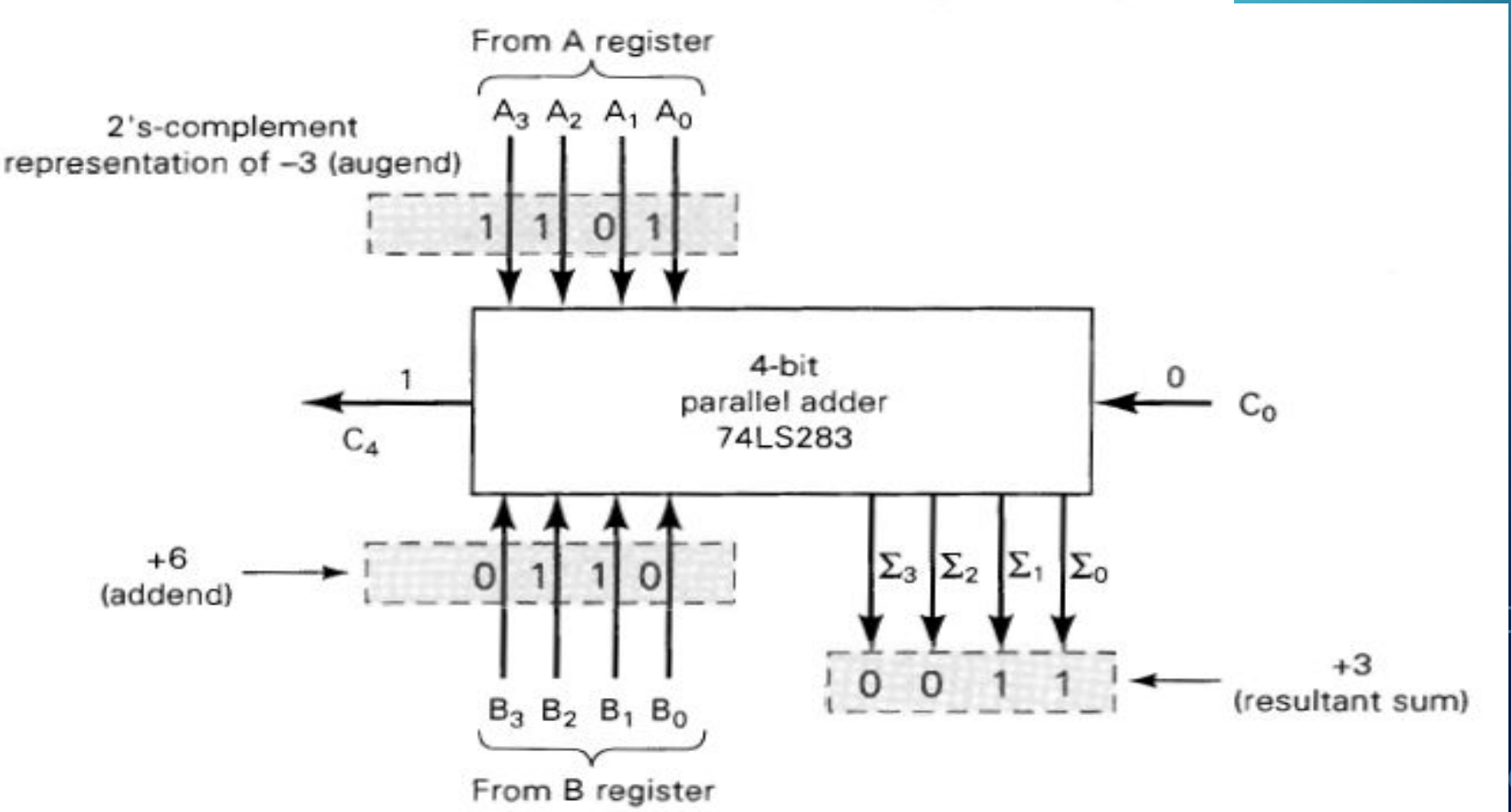
2's-COMPLEMENT SYSTEM

Addition

Positive and negative numbers, including the sign bits, can be added together in the basic parallel-adder circuit when the negative numbers are in 2's-complement form.

for the addition of −3 and +6. The −3 is represented in its 2's-complement form as 1101

Parallel adder used to add + and − numbers in 2's-complement system.



COMPARATOR

The basic function of a comparator is to compare the magnitudes of two binary quantities to determine the relationship of these quantities

Basic comparator operation.

The input bits are equal.

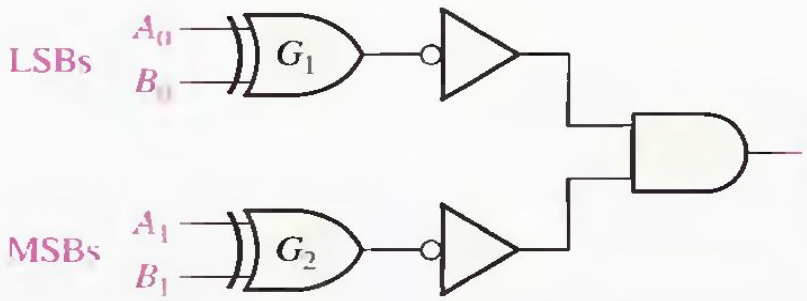
The input bits are not equal.

The input bits are not equal.

The input bits are equal.

In order to compare binary numbers containing two bits

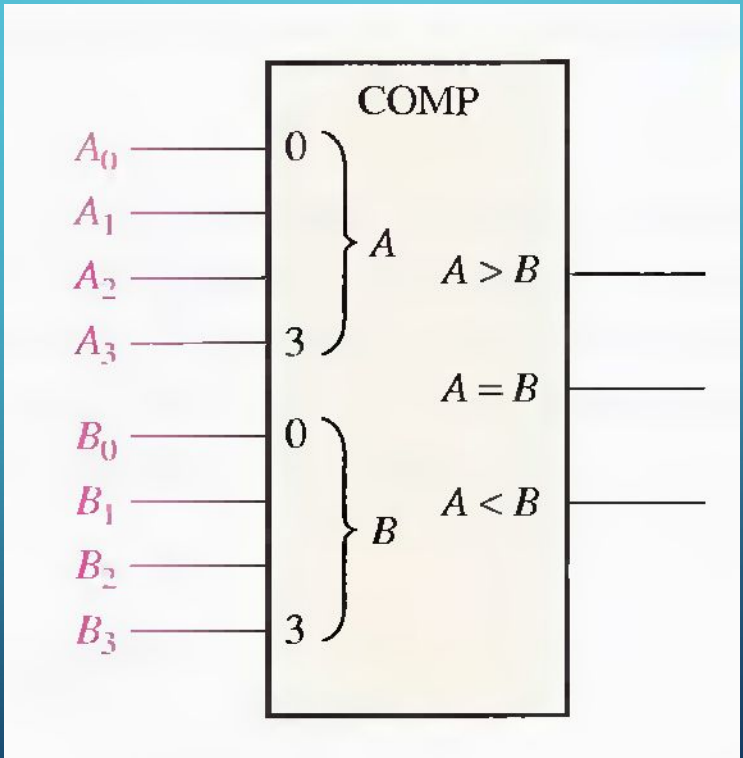
Equality



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

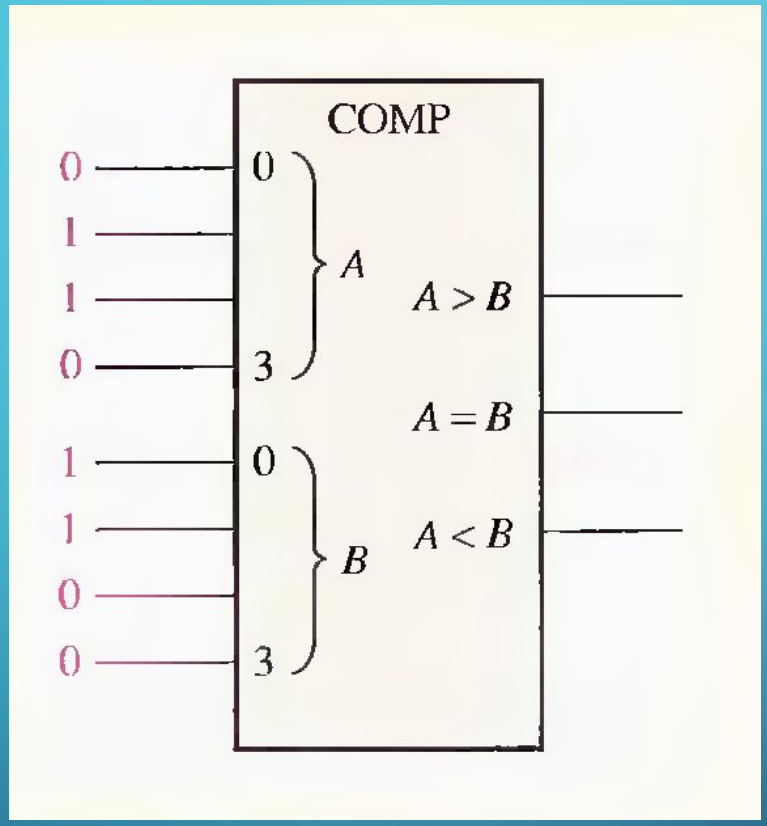
Inequality

1. If $A_3 = 1$ and $B_3 = 0$, number A is greater than number B.
2. If $A_3 = 0$ and $B_3 = 1$, number A is less than number B.
3. If $A_3 = B_3$, then you must examine the next lower bit position for an inequality.



The highest-order indication must take precedence.

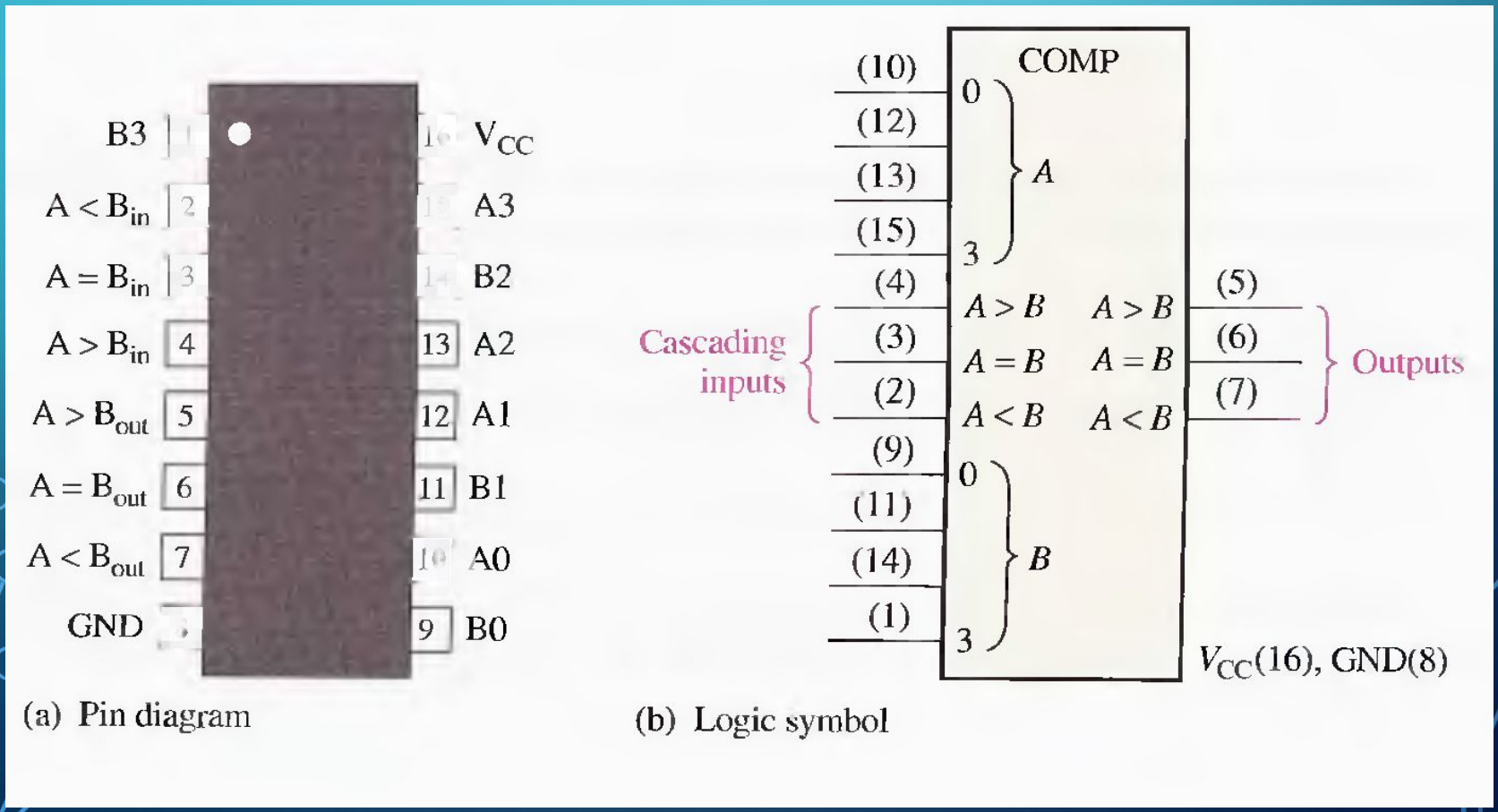
Determine the A = B, A > B, and A < B outputs for the input numbers shown on the comparator in Figure .



The number on the *A* inputs is 0110 and the number on the *B* inputs is 0011. The ***A* > *B* output is HIGH and the other outputs are LOW.**

THE 74HC85 4-BIT MAGNITUDE COMPARATOR

The 74HC85 is a comparator that is also available in other IC families. The pin diagram and logic symbol are shown in Figure 0-24. Notice that this device has all the inputs and outputs of the generalized comparator previously discussed and, in addition, has three cascading inputs: A < B, A = B, A > B. These inputs allow several comparators to be cascaded for comparison of any number of bits greater than four.



Compare two 8-bit numbers:

Two 74HC85s are required to compare two 8-bit numbers. They are connected as shown in Figure in a cascaded arrangement.