Digital Logic & Design

Functions of combinational logic comparators

Hazara University Mansehra
Department of Computing & IT

Comparators

- The basic function of a comparator is to compare the magnitudes of two binary quantities to determine the relationship of those quantities.
- In its simplest form, a comparator circuit determines whether two numbers are equal.

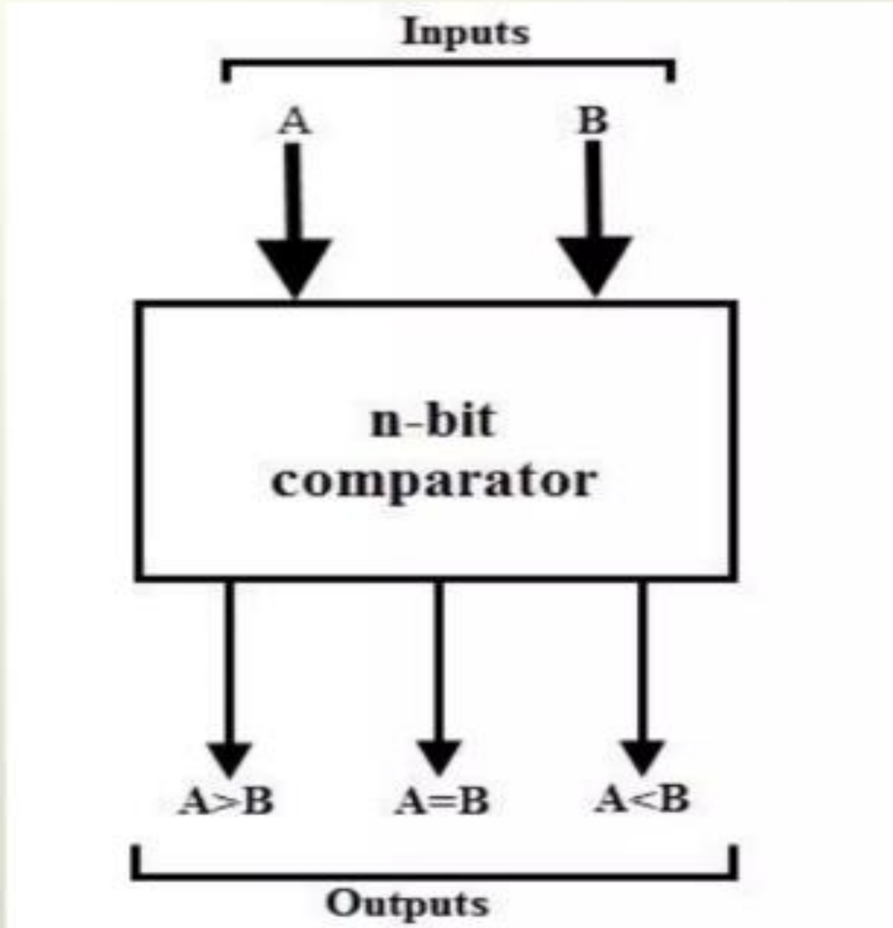
Magnitude Comparator

- Three binary variables are used to indicate the outcome of the comparison as

  $A>B$, $A<B$, or $A=B$.

- The below figure shows the block diagram of a n-bit comparator which compares the two numbers of n-bit length and generates their relation between themselves.

Digital Comparator



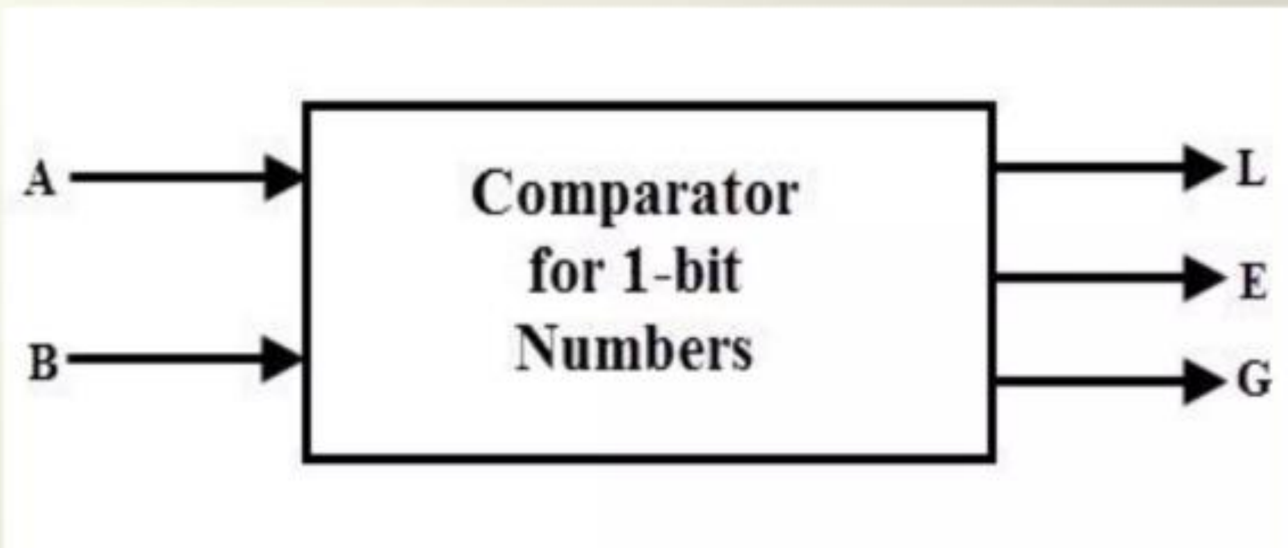
Comparators

- 1-Bit Comparator
- 2-Bit Comparator
- 4-Bit Comparator

1 bit Magnitude comparator

- A comparator used to compare two numbers each of single bit is called single bit comparator.
- It consists of two inputs for allowing two single bit numbers and three outputs to generate less than, equal and greater than comparison outputs.
- The figure below shows the block diagram of a single bit magnitude comparator.
- This comparator compares the two bits and produces one of the 3 outputs as L (A<B), E (A=B) and G (A>B).

Block Diagram (Single Bit Comparator)



- There are two different types of output relationship between the two binary quantities;

- **Equality output** indicates that the two binary numbers being compared is equal ($A = B$) and

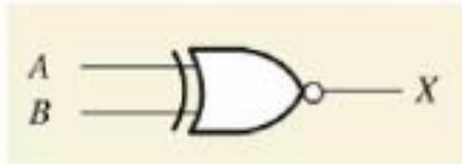
- **Inequality output** that indicates which of the two binary number being compared is the larger.

- That is, there is an output that indicates when A is greater than B ($A > B$) and an output that indicates when A is less than B ($A < B$).

Equality

- XNOR gate can be used as a basic comparator
- Output is a 0 if the two input bits are not equal and 1 if the input bits are equals

- 1-Bit Comparator

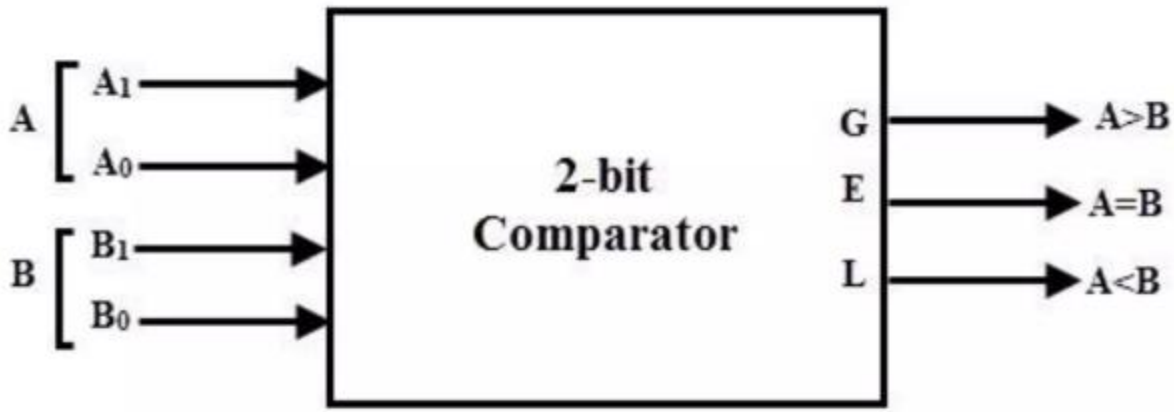


A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

The output is 1 when the inputs are equal

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.

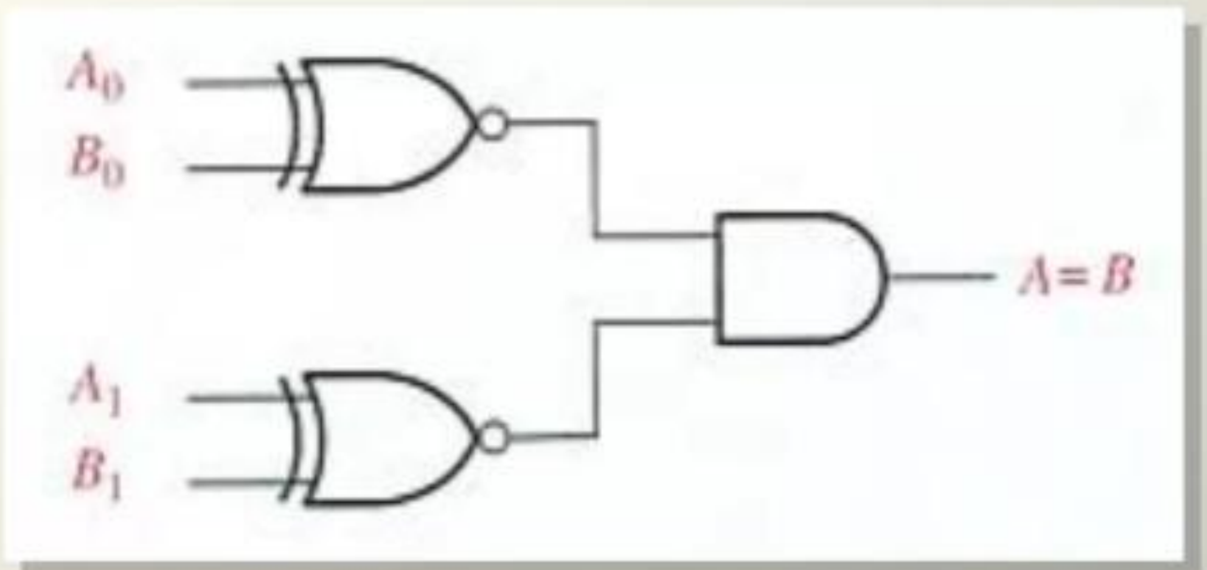


A
A1
A0
B
B1
B0
2-bit
Comparator
G
E
L
A>B
A=B
A<B

- In order to compare binary numbers containing two bits each, an additional XNOR gate is necessary
- **2 LSB** of two numbers are compared by gate G1
- **2 MSB** of two numbers are compared by gate G2
- **1 AND** gate can be used

- If 2 numbers are equal, their corresponding bits are same and the output of each X-NOR gate is 1.
- If the corresponding sets of bits are not equal, a 0 occurs on that exclusive –NOR gate output.

- 2-Bit Comparator



The output is 1 when $A_0 = B_0$ AND $A_1 = B_1$

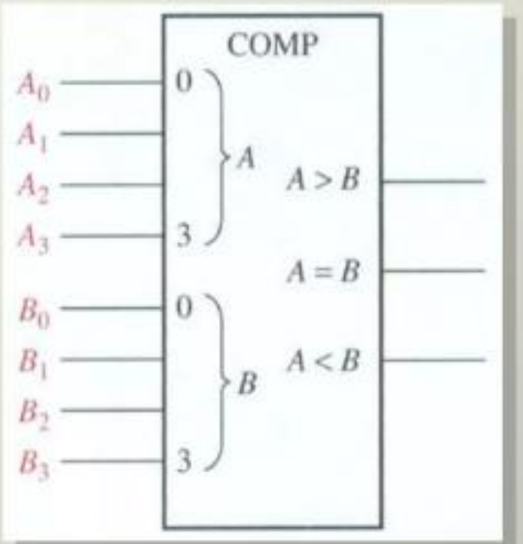
Example

- Apply each of the following sets of binary numbers to the comparator inputs and determine the output by the following logic levels through the circuit.
- 10 and10
- 11 and 10

- Repeat the process for binary inputs of 01 and 10.

In-Equality

- In addition to the equality output, fixed function comparators can provide additional outputs that indicate:
- Which of the two binary numbers being compared is the larger.
- i.e. An output that indicates when number A is greater than number B. (A>B)
- An output that indicates when number A is less than number B (A<B) as shown in logic symbol for 4-bit comparator.



4-Bit Comparator

- It can be used to compare two four-bit words.
- The two 4-bit numbers are A = A3 A2 A1 A0 and B3 B2 B1 B0 where A3 and B3 are the most significant bits.

It has **three active-HIGH outputs**

Start with most significant bit in each number to determine the inequality of 4-bit binary numbers A and B

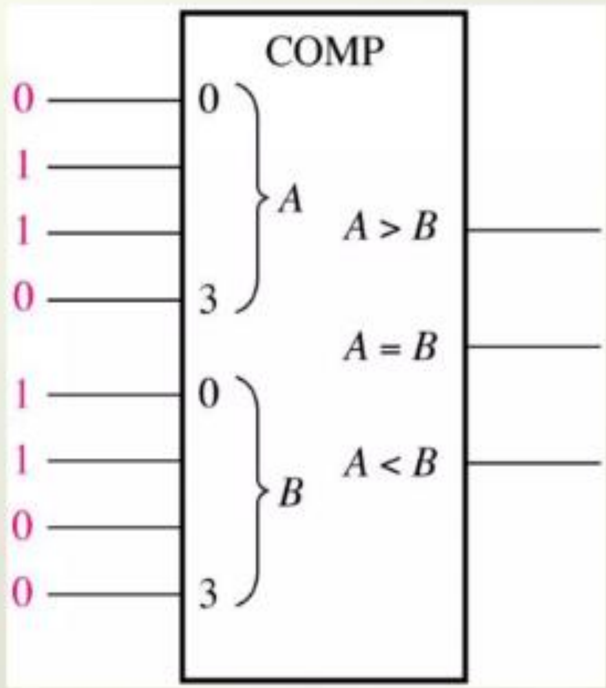
- Output $A<B$ will be HIGH if $A_3=0$, and $B_3=1$
- Output $A>B$ will be HIGH if $A_3=1$, and $B_3=0$

- If $A_3=0$, and $B_3=0$ or $A_3=1$, and $B_3=1$, then examine the next lower order bit position for an inequality, Only when all bits of $A=B$, output $A=B$ will be HIGH

The general procedure used in comparator:

- Start with the highest-order bits (MSB)
- When an inequality is found, the relationship of the 2 numbers is established, and any other inequalities in lower-order positions must be ignored
- THE HIGHEST ORDER INDICATION MUST TAKE **PRECEDENCE**

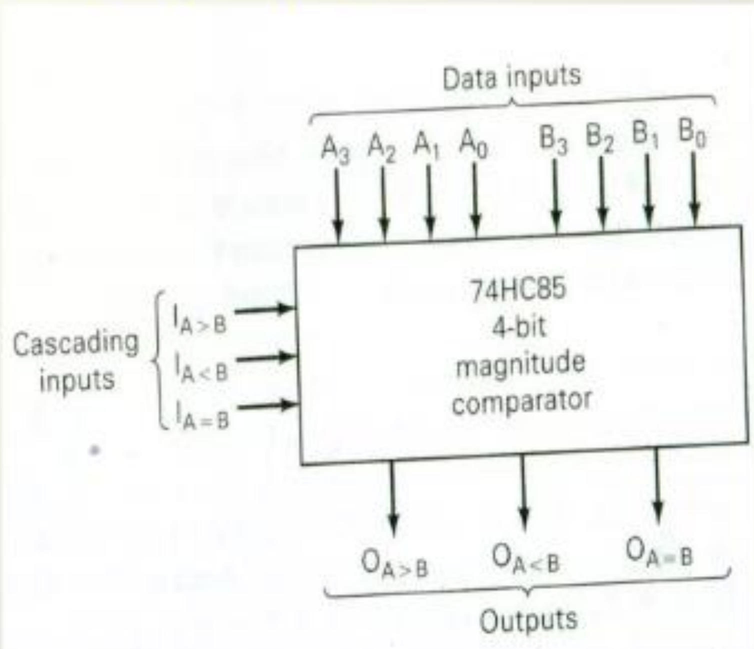
Example: Determine the A=B, A>B, and A<B outputs for the input numbers shown on the 4-bit comparator as given below.

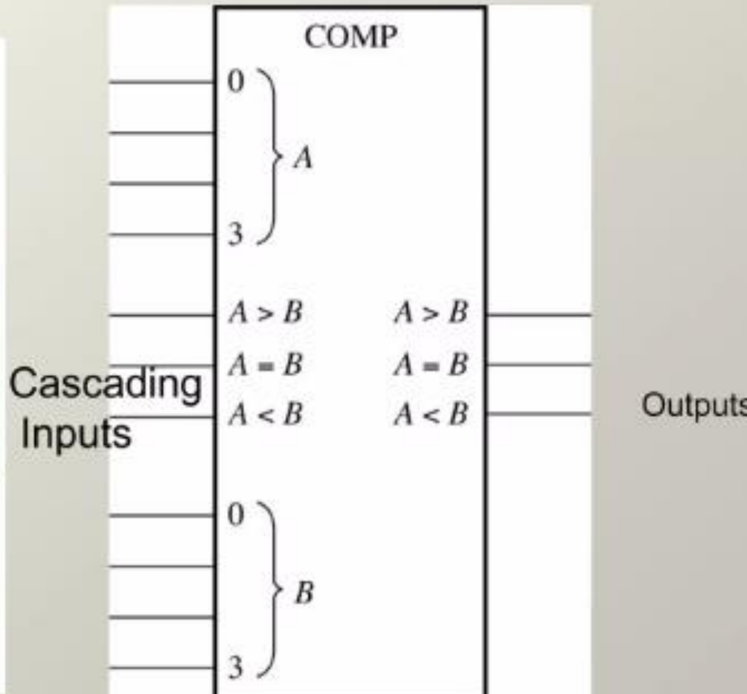


Solution: The number on the A inputs is 0110 and the number on the B inputs is 0011.
The A > B output is HIGH and the other outputs (A=B and A<B) are LOW

74LS85 (4–bit magnitude comparator)

The 74LS85 **compares two unsigned 4-bit binary** numbers , the unsigned numbers are A_3, A_2, A_1, A_0 and B_3, B_2, B_1, B_0.





Comparator Expansion (Cascading Comparator)

- In addition, it also has three cascading inputs:

- These inputs provides a means for expanding the comparison operation by cascading two or more 4–bit comparator.

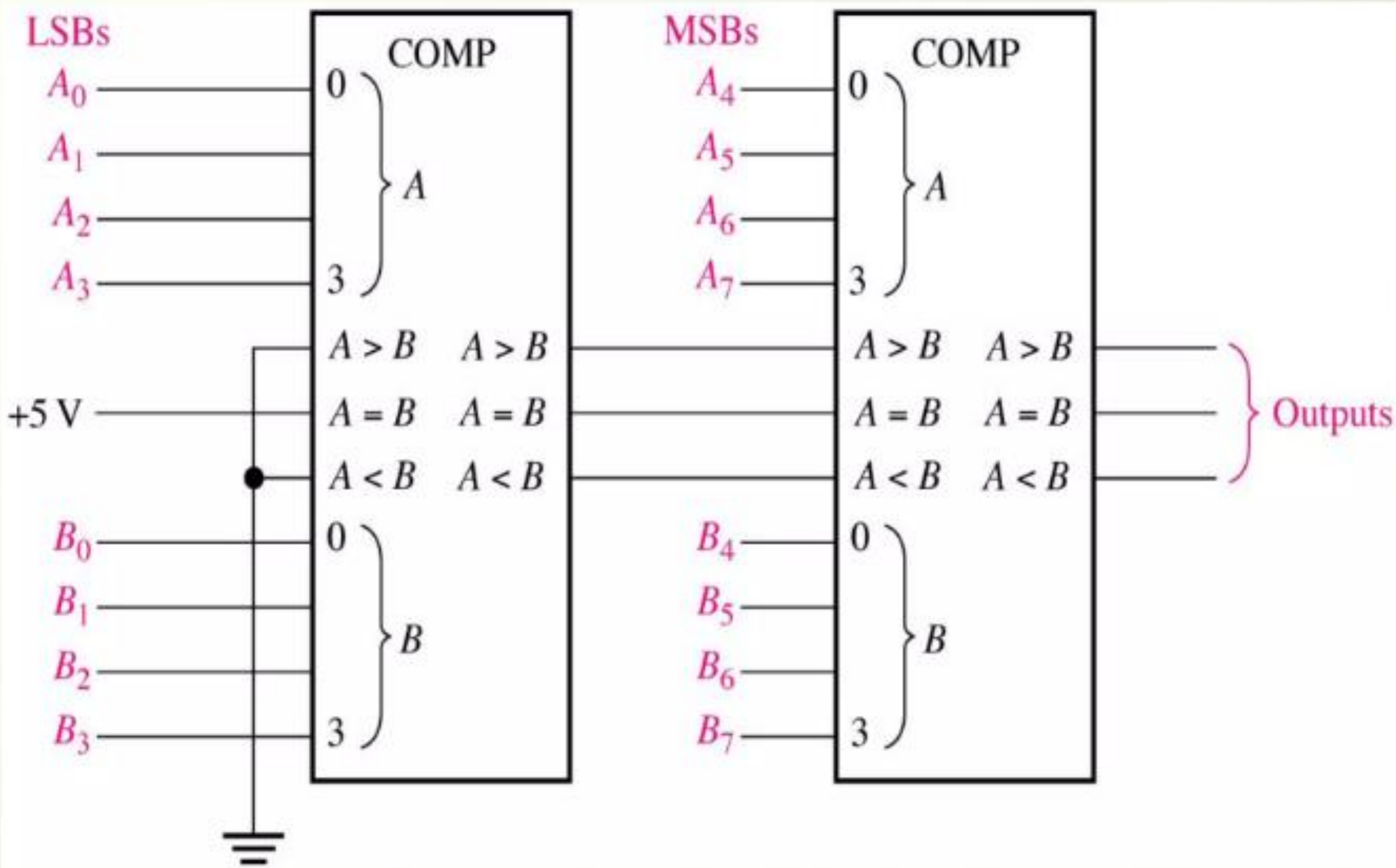
- To expand the comparator, the $A<B$, $A=B$, and $A>B$ outputs of the lower–order comparator are connected to the corresponding cascading inputs of the next higher–order comparator.

- The lowest-order comparator must have a HIGH on the A=B, and LOWs on the A<B and A>B inputs as shown in next slide.
- The comparator on the left is comparing the lower-order 8–bit with the comparator on the right with higher–order 8–bit .

- The outputs of the lower–order bits are fed to the cascade inputs of the comparator on the right, which is comparing the high-order bits.

- The outputs of the high-order comparator are the final outputs that indicate the result of the 8–bit comparison.

An 8-bit magnitude comparator using two 4-bit comparators.



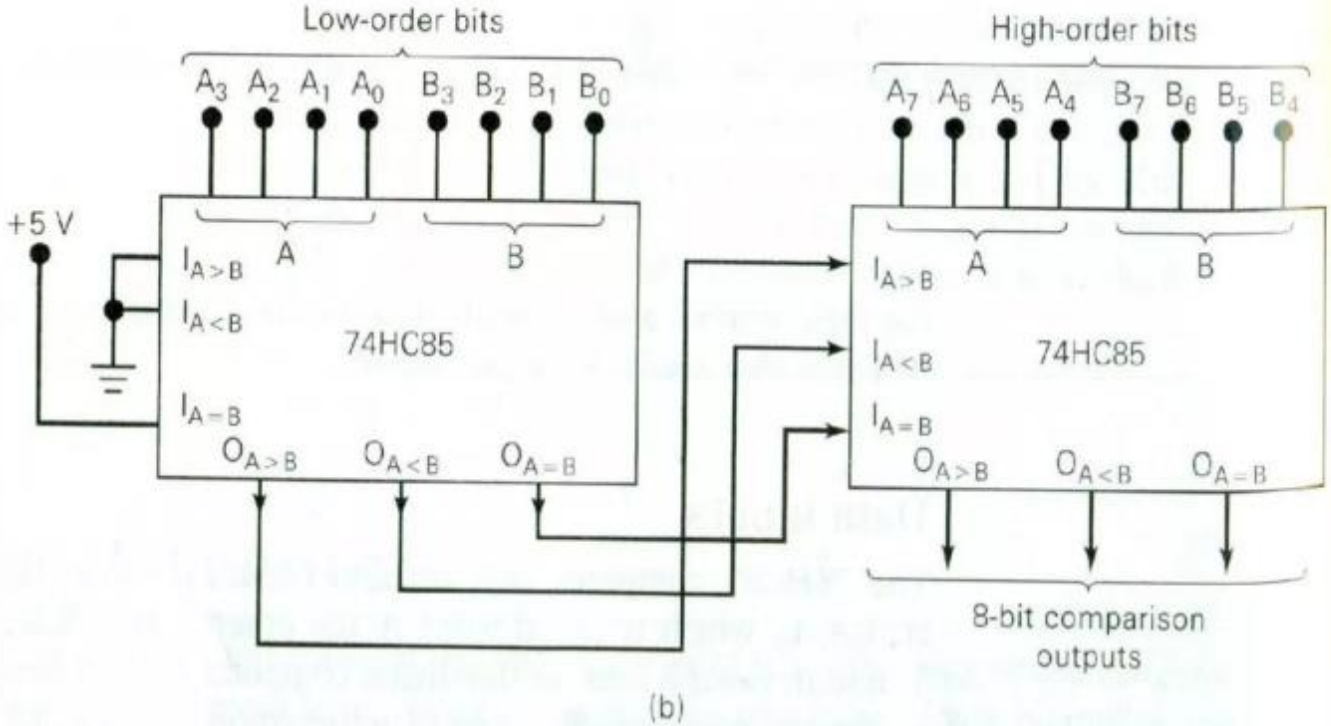


FIGURE 9-37 (a) 74HC85 wired as a four-bit comparator; (b) two 74HC85s cascaded to perform an eight-bit comparison.

Describe the operation of the eight-bit comparison arrangement in Figure 9-37(b) for the following cases.

(a) $A_7A_6A_5A_4A_3A_2A_1A_0 = 10101111$; $B_7B_6B_5B_4B_3B_2B_1B_0 = 10110001$

(b) $A_7A_6A_5A_4A_3A_2A_1A_0 = 10101111$; $B_7B_6B_5B_4B_3B_2B_1B_0 = 10101001$

(b) The high-order comparator sees $A_7A_6A_5A_4 = B_7B_6B_5B_4 = 1010$, so it must look at its cascade inputs to see the result of the low-order comparison. The low-order comparator has $A_3A_2A_1A_0 = 1111$ and $B_3B_2B_1B_0 = 1001$, which produces a 1 at its $O_{A>B}$ (low 4bit) output and the $I_{A>B}$ input of the high-order comparator. The high-order comparator senses this 1, and since its data inputs are equal, it produces a HIGH at its $O_{A>B}$ to indicate the result of the eight-bit comparison. ✓