MAGNITUDE COMPARATOR

A ***magnitude comparator*** is a combinational circuit that compares two given numbers (A and B) and determines whether one is equal to, less than or greater than the other. The output is in the form of three binary variables representing the conditions A = B, A>B and A<B, if A and B are the two numbers being compared.

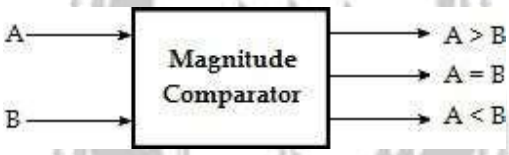


Fig : 2.21 - Block diagram of magnitude comparator

For comparison of two *n*-bit numbers, the classical method to achieve the Boolean expressions requires a truth table of 2^{2n} entries and becomes too lengthy and cumbersome.

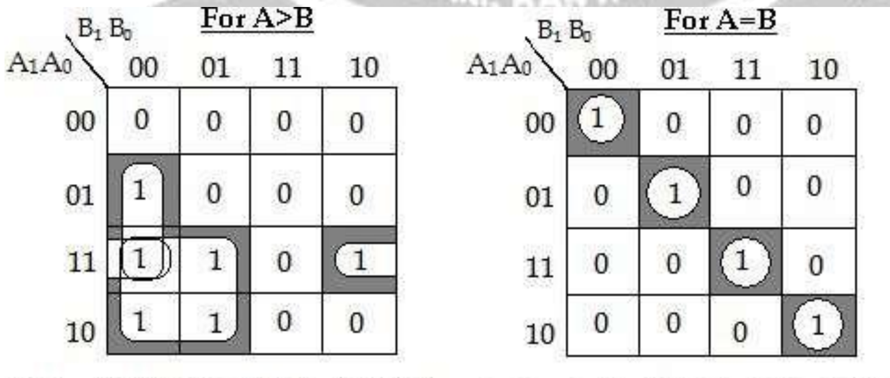
2-bit Magnitude Comparator:

The truth table of 2-bit comparator is given in table below—Truth table:

Inputs				**Outputs**		
A3	**A2**	**A1**	**A0**	**A>B**	**A=B**	**A<B**
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

K-map Simplification**:**



For A>B

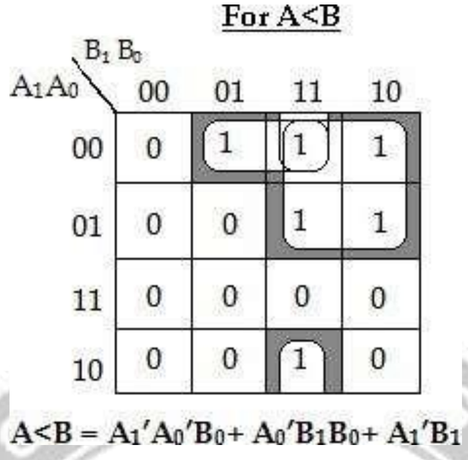
$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

For A=B

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$A>B = A_0B_1'B_0' + A_1B_1' + A_1A_0B_0'$

$$\begin{aligned} A=B &= A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0B_1B_0 + A_1A_0'B_1B_0' \\ &= A_1'B_1'(A_0'B_0' + A_0B_0) + A_1B_1(A_0B_0 + A_0'B_0') \\ &= (A_0 \odot B_0)(A_1 \odot B_1) \end{aligned}$$

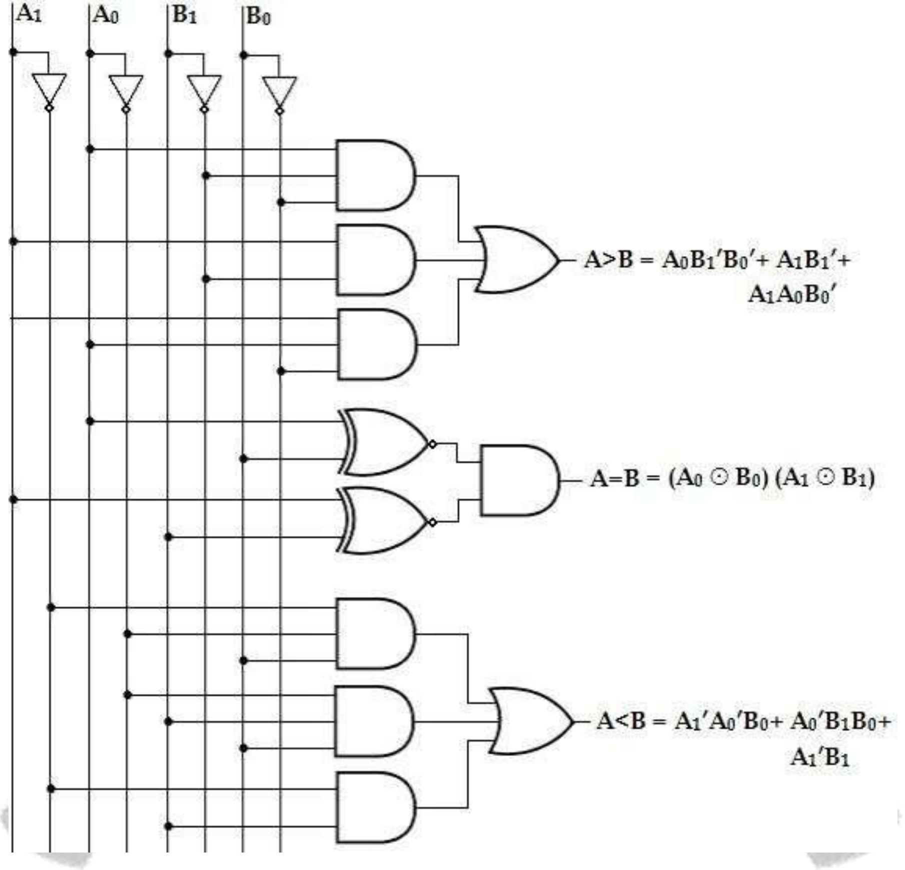


For A<B

A_1A_0 \ B_1B_0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$A<B = A_1'A_0'B_0 + A_0'B_1B_0 + A_1'B_1$

Logic Diagram:



A_1
A_0
B_1
B_0
$A>B = A_0B_1'B_0' + A_1B_1' + A_1A_0B_0'$
$A=B = (A_0 \odot B_0)(A_1 \odot B_1)$
$A<B = A_1'A_0'B_0 + A_0'B_1B_0 + A_1'B_1$

Fig : 2.22 - 2-bit Magnitude Comparator4-bit Magnitude Comparator

Let us consider the two binary numbers A and B with four digits each. Write the coefficient of the numbers in descending order as,

A = A3A2A1A0

B = B3 B2 B1 B0,

Each subscripted letter represents one of the digits in the number. It is observed from the bit contents of two numbers that A = B when $A_3 = B_3$, $A_2 = B_2$, $A_1 = B_1$ and $A_0 = B_0$. When the numbers are binary they possess the value of either 1 or 0, the equality relation of each pair can be expressed logically by the equivalence function as

$\mathbf{X}_i = \mathbf{A}_i\mathbf{B}_i + \mathbf{A}_i'\mathbf{B}_i'$ $\mathbf{X}_i$ for i = 1, 2, 3,

Or, = (**A** **B**)′. 4. or, **X*i* ′ = A**

B

Or, $\mathbf{X}_i = (\mathbf{A}_i\mathbf{B}_i' + \mathbf{A}_i'\mathbf{B}_i)'$.

where, X*i =1* only if the pair of bits in position i are equal (ie., if both are 1 or both are0). To satisfy the equality condition of two numbers A and B, it is necessary that all X*i* must be equal to logic 1. This indicates the AND operation of all X*i* variables. In other words, we can write the Boolean expression for two equal 4-bit numbers.

$$(\mathbf{A} = \mathbf{B}) = \mathbf{X}_3\mathbf{X}_2\mathbf{X}_1\,\mathbf{X}_0.$$

The binary variable (A=B) is equal to 1 only if all pairs of digits of the two numbers are equal.

To determine if A is greater than or less than B, we inspect the relative magnitudes of pairs of significant bits starting from the most significant bit. If the two digits of the most significant position are equal, the next significant pair of digits is compared. The comparison process is continued until a pair of unequal digits is found. It may be concluded that A>B, if the corresponding digit of A is 1 and B is 0. If the corresponding digit of A is 0 and B is 1, we conclude that A<B. Therefore, we can derive the logicalexpression of such sequential comparison by the following two Boolean functions,

$$\mathbf{(A>B) = A_3B_3' + X_3A_2B_2' + X_3X_2A_1B_1'}$$
$$\mathbf{+ X_3X_2X_1A_0B_0' \; (A<B) = A3'B3 + X3A2'B2}$$
$$\mathbf{+ X3X2A1'B1 + X3X2X1A0'B0}$$

The symbols (A>B) and (A<B) are binary output variables that are equal to 1 when A>B or A<B, respectively.

The gate implementation of the three output variables just derived is simpler than it seems because it involves a certain amount of repetition. The unequal outputs can use the same gates that are needed to generate the equal output. The logic diagram of the 4-bit magnitude comparator is shown below,

The four x outputs are generated with exclusive-NOR circuits and applied to an AND gate to give the binary output variable (A=B). The other two outputs use the x variables to generate the Boolean functions listed above. This is a multilevel implementation and has a regular pattern.

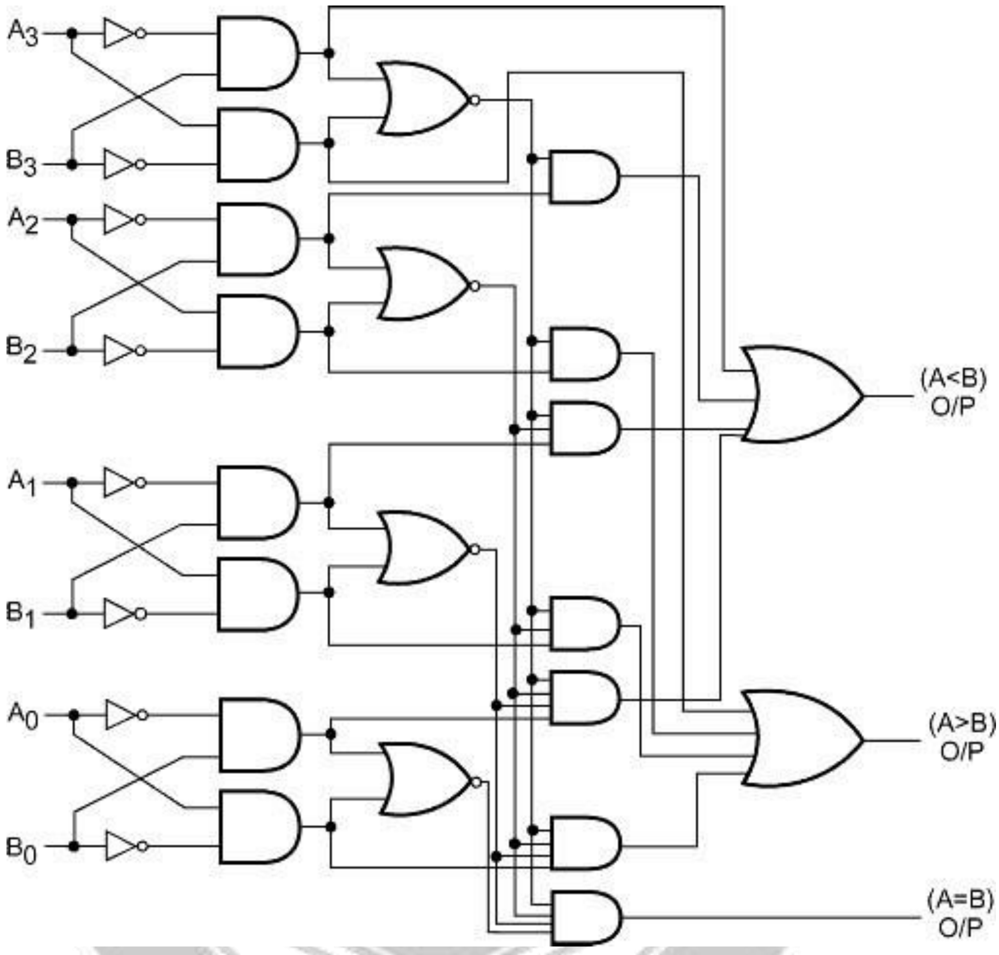


Fig : 2.23 - 4-bit Magnitude Comparator