CLASS: S.E. COMPUTER **SUBJECT: DEL**

EXPT. NO.: 5 **DATE:**

TITLE **:** To Verify the truth table of two-bit comparators using logic gates

OBJECTIVE **:** 1 bit, 2-bit Comparator.

APPARATUS **:** Digital-Board, GP-4Patch-Cords, Comparator IC-7485 and Required Logic gates if any.

THEORY :

Another common and very useful combinational logic circuit is that of the **Digital Comparator** circuit. Digital or Binary Comparators are made up from standard AND, NOR and NOT gates that compare the digital signals present at their input terminals and produce an output depending upon the condition of those inputs.

For example, along with being able to add and subtract binary numbers we need to be able to compare them and determine whether the value of input A is greater than, smaller than or equal to the value at input B etc. The digital comparator accomplishes this using several logic gates that operate on the principles of *Boolean algebra*. There are two main types of **Digital Comparator** available and these are

1. **Identity Comparator** – an *Identity Comparator* is a digital comparator that has only one output terminal for when A = B either -HIGH‖ A = B = 1 or -LOW‖ A = B = 0

2. **Magnitude Comparator** – a *Magnitude Comparator* is a digital comparator which has three output terminals, one each for equality, A = B greater than, A > B and less than A < B.

The purpose of a **Digital Comparator** is to compare a set of variables or unknown numbers, for example A (A1, A2, A3… An, etc) against that of a constant or unknown value such as B (B1, B2, B3… Bn, etc) and produce an output condition or flag depending upon the result of the comparison. For example, a magnitude comparator of two 1-bits, (A and B) inputs would produce the following three output conditions when compared to each other. Which means: A is greater than B, A is equal to B, and A is less than B

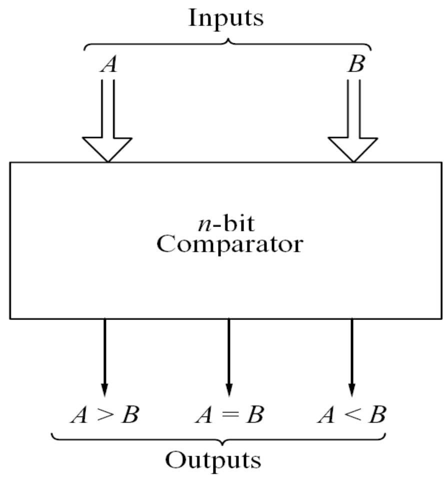


Fig. 6.22 ***Block Diagram of***

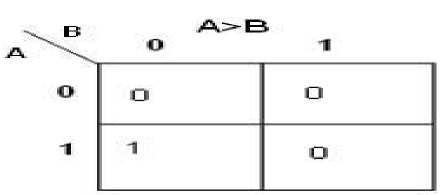
This is useful if we want to compare two variables and want to produce an output when any of the above three conditions are achieved. For example, produce an output from a counter when a certain count number is reached. Consider the simple 1-bit comparator below

1. 1-bit comparator

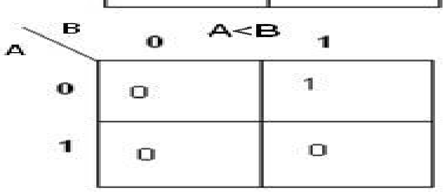
- Truth Table: -

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

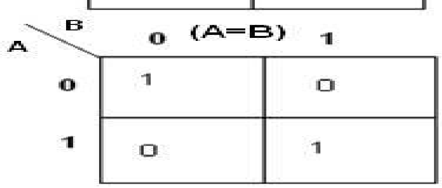
- K-Map



Equation is $A>B = A.\overline{B}$



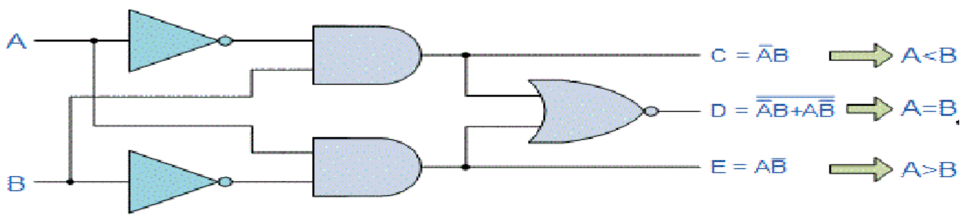
Equation is $A<B = \overline{A}.B$



The equation is $f(A=B) = \overline{A}.\overline{B} + A.B$
$= A \text{ XNOR } B$

or we can write the equation for $f(A=B)$ as $\overline{A.\overline{B} + \overline{A}.B} = \overline{f(A>B) + f(A<B)}$

- **Logic Diagram of 1 bit Comparator**



---

2 Bit Comparator:-

- **Truth Table:-**

Inputs				Outputs		
A_1	A_0	B_1	B_0	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

- **<u>K-map :-</u>**

1. For A>B:

f(A>B)

A_1A_0 \ 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

We get the equation as f(A>B)

$$= A_1\overline{B_1} + A_0\overline{B_1}\,\overline{B_0} + A_1A_0\overline{B_0}$$

2. For A=B

f(A=B)

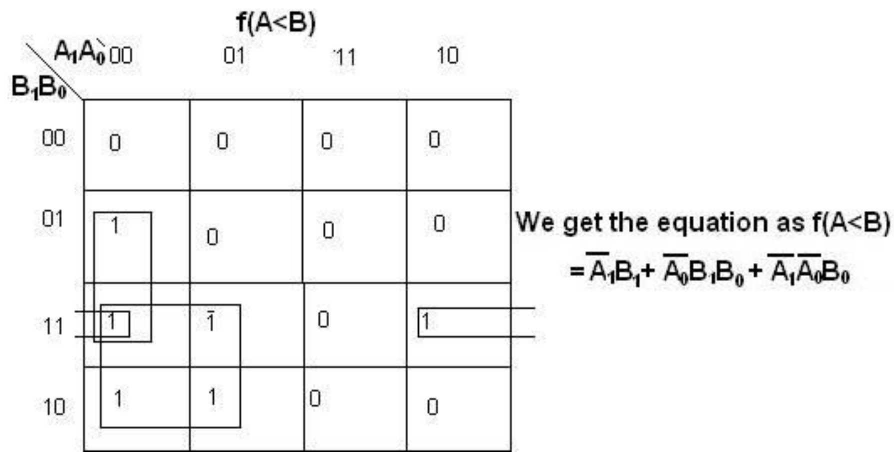
A_1A_0 \ B_1B_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

We get the equation as f(A=B)

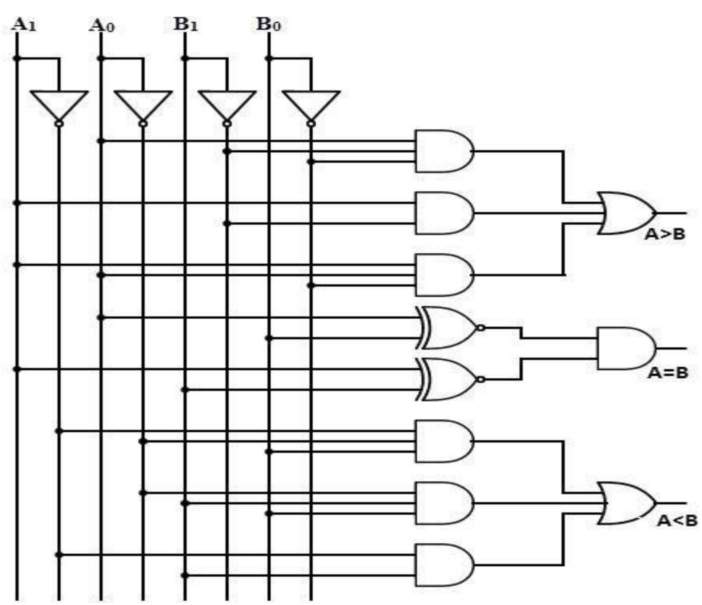
$$=(A_1 \text{XOR } B_1).(A_0 \text{XOR } B_0)$$

or we can write the equation for f(A=B) as $= \overline{f(A>B)+f(A<B)}$

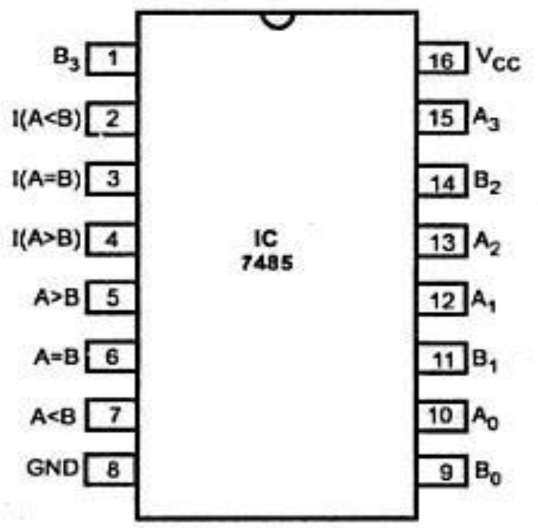
3. For A<B :-



- **Circuit Diagram:-**



PIN DIAGRAM



(a) Pin diagram (IC 7485)

- **For n bit Comparator :-**

Digital comparators actually use Exclusive-NOR gates within their design for comparing their respective pairs of bits. When we are comparing two binary or BCD values or variables againsteach other, we are comparing the ―magnitude‖ of these values, a logic ―0‖ against a logic ―1‖ which is where the term **Magnitude Comparator** comes from.

As well as comparing individual bits, we can design larger bit comparators by cascading togethern of these and produce a n-bit comparator just as we did for the n-bit adder in the previous tutorial. Multi-bit comparators can be constructed to compare whole binary or BCD words to produce an output if one word is larger, equal to or less than the other.

Outcome:

Up and down counters are successfully implemented, the comparators are studied & o/p are checked. The truth table is verified.

REFERENCE:

1. R.P.Jain "Modern Digital Electronics" TMH 4^{th} Edition
2. D.Leach,Malvino,Saha,"Digital Principles and Applications",TMH