Module -9

Combinational Circuits and Comparator

Learning Objectives

After learning this module you will be able to

1. Understand the concept of Combinational Circuits
2. Study the classification of Combinational Circuits
3. Learn design and analysis procedure of Combinational Circuits
4. Get familiar with comparators and its applications

1. Introduction

Digital logic circuits are classified into two main categories namely Combinational and sequential circuits. Combinational logic is a type of digital logic which is implemented using logic gates. The output of combinational circuit is the function of combination of present inputs only. Whereas in sequential circuit the output depends not only on the present input but also on the previous inputs i.e. history of inputs. In other words, a sequential circuit has memory while combinational circuit does not. Combinational circuit is able to generate an output simply from knowing what current input values are. Combinational circuits do not need to know the history of past inputs.

Let us take real life example of combinational circuit e.g. Car security system. In this example, the siren is turned on when the master switch is on and someone opens the door. If you close the door then the siren will turn off immediately. With this setup, the output, which is the siren, is depend only on the present inputs, which are the master and door switches.

This module deals with studying basics of combinational circuits and their classification. Further this module is also aimed at to use the knowledge acquired in previous units and formulate systematic design and analysis procedures of combinational circuits. The main focus is on learning the basics, types, designing and applications of comparators.

2. Combinational Circuit basics

A combinational circuit consists of input variables, logic gates, and output variables. The logic gates accept signals from the inputs and generate signals to the outputs. This process transforms binary information from the given input data to the required output data. Obviously, both input and output data are represented by binary signals, i.e. they exist in two possible values, one representing logic-1 and the other logic-0. A block diagram of a combinational circuit is shown in Fig. 1.

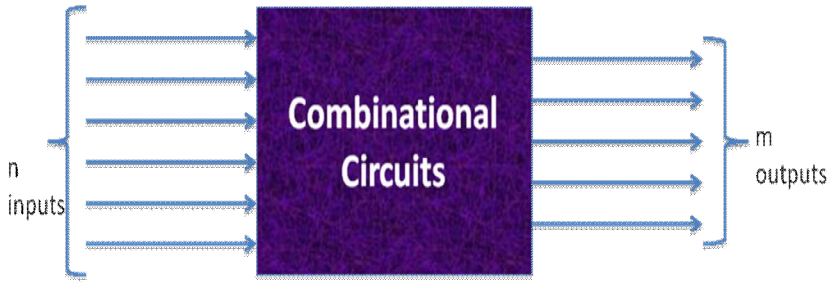


Figure 1: Combinational logic circuit

Combination Logic Circuits are made up from basic gates (AND, OR, NOT) or universal gates (NAND, NOR) gates that are "combined" or connected together to produce more complicated logic circuits. These logic gates are the building blocks of combinational logic circuits. In these circuits ñ**the outputs at any instant of time depends on the inputs present at that instant only**.ò

For ***n*** input variables, there are 2^n possible combinations of binary input values. For each possible input combination, there is one and only one possible output combination. A combinational circuit can be described by ***m*** Boolean functions, one for each output variable. Each output function is expressed in terms of the ***n*** input variables.

3. Types of combinational circuits

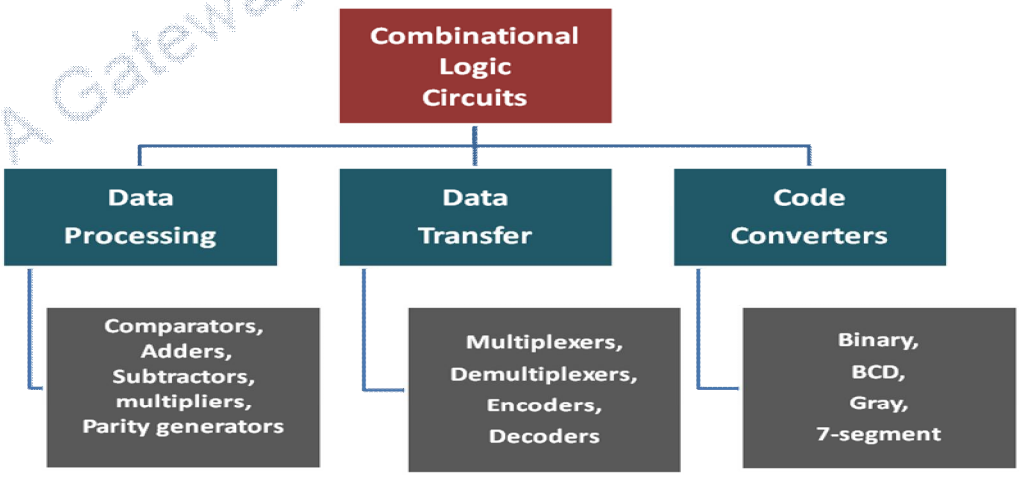


Figure 2: Classification of Combinational circuits

At any time a combinational logic circuitôs outputs depends only upon the combination of its inputs at that time. It is important to note that output is not influenced by previous inputs. Fig. 2 shows the classification scheme of Combinational circuits.

There are three major types of combinational circuits:

(a) **Data processing** circuits that process or transform data e.g. Adder, subtractor, comparator, multiplier etc.

(b) **Data transfer** circuits to control the flow of logic e.g. Multiplexer, demultiplexer, encoder, decoder etc.

(c) **Code converters** for conversion from binary to any other code and vice versa e.g. Binary, BCD, Gray, 7-segment converters

4. Design procedure of combinational circuits

For the design of Combinational digital circuits basic gates (AND, OR, NOT) or universal gates (NAND, NOR) are used. The design of combinational circuits begins with problem definition and ends with a logic circuit or set of Boolean expressions.

1. State the problem clearly.
2. Determine the available inputs and required outputs from problem.
3. Assign a letter symbols to input and outputs of the circuits.
4. Derive the truth table that relates inputs and outputs.
5. Write the Boolean functions for the output and further simplify to get reduced number of variables.
6. Draw the logic diagram.

5. Analysis procedure of combinational circuits

In many applications, logic diagram is available but correlation between input and output is not known. The analysis of combinational circuits is the reverse process.

1. Take a closer look at given combinational circuit.
2. Identify the available inputs and outputs and label them appropriately.

3. Find the output Boolean function for each gate.
4. Obtain the output Boolean functions and / or truth table for the given circuits in terms of input variables.

Next few modules will describe the design and analysis of the combinational circuits with special attention to simplification of Boolean expressions leading to minimization of logic circuits.

6. Comparator

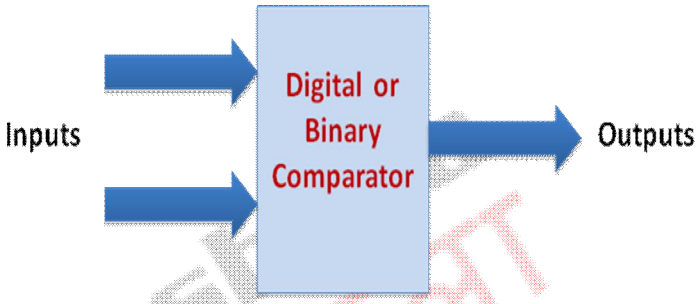


Figure 3: Digital Comparator

Comparator is a combinational logic circuit that compares the magnitudes of two binary quantities to determine which one has the greater magnitude. In other word, a comparator determines the relationship of two binary quantities. A exclusive–OR gate can be used as a basic comparator.

Digital comparators are also called as binary or logic comparators. Comparator is a very useful combinational circuit capable of comparing two numbers as input in binary form and determines whether one number is greater than, less than or equal to other number. This logic circuit is used for testing whether the binary number at one input is greater than or less than or equal to another binary number.

There are two types of comparators:

1. Equality or Identity Comparator
2. Magnitude or inequality comparator.

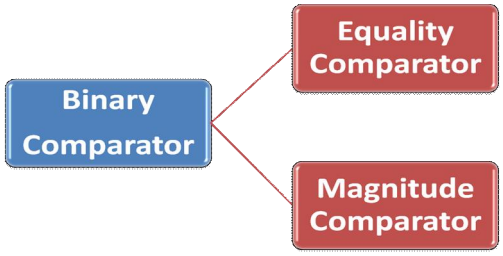


Figure 4: Types of Binary comparators

6.1 Equality (or Identity) comparator

Equality comparator requires two input binary numbers A and B for comparison as shown in fig. 5. There is only one output indicating whether the input numbers are equal or not.

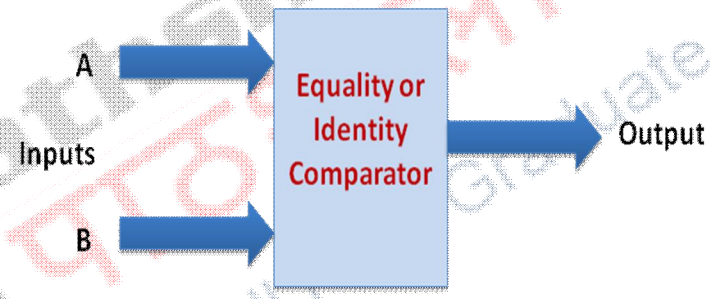
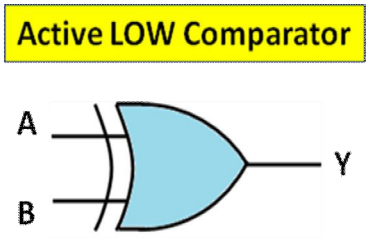


Figure 5: Logic symbol of equality comparator

Output of the comparator is at logic ó1ô if the two input words A and B match otherwise the output remains at logic ó0ô. Out of the 2^n combinations, there is only one input combination that is correct and other combinations will result in inequality.

This comparator can be used for simple circuits like electronic lock and security systems where binary password consisting of n bits compared with another preset code. Let us discuss the implementation of simple 1-bit equality comparator.

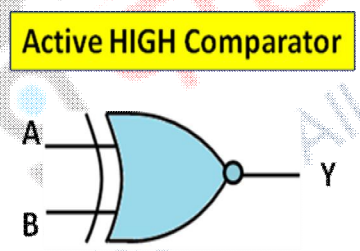
1-bit Equality comparator



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Figure 6: 1 bit Equality comparator (Active LOW)

Let us study 1-bit equality comparator using simple XOR circuit. The truth table of XOR indicates the operation of Identity or Equality comparator. When both the inputs of the comparator are equal then the output is active LOW indicating matching of the input code, other wise output is at logic ó1ô indicating the mismatch.



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Figure 7: 1 bit Equality comparator (Active HIGH)

Let us study ACTIVE HIGH 1-bit equality comparator using simple XNOR circuit as shown in fig. 7. The truth table of XNOR indicates the operation of Identity or Equality comparator. If two inputs are identical then the output of comparator is ó1ô else provides ó0ô indicating mismatch in the input code.

2-bit Equality comparator

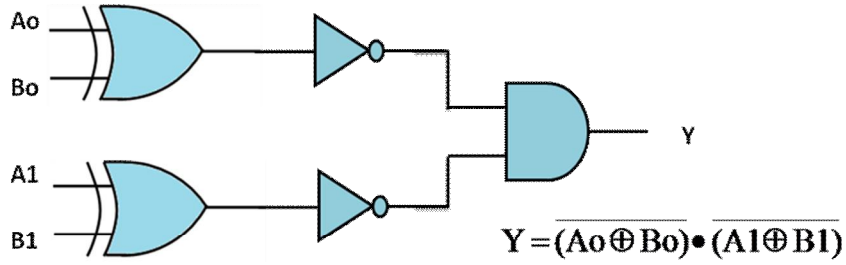


Fig. 8: 2-bit Equality comparator

Inputs				Outputs
A1	A0	B1	BO	A=B
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

In two bit equality comparator as shown in fig.8, two bit binary numbers A and B represented as A1A0 and B1B0 are compared. Here A1 and B1 are the most significant bits, and A0 and B0 are the least significant bits of numbers A and B respectively. A1 is compared with B1 and A0 is compared the B0. If A1 is equal to B1 and A0 is equal to B0 output of a two bit comparator will be HIGH. If A1 is not equal to B1 or A0 is not equal to B0 output will be LOW as shown in the table.

6.2 Magnitude comparator

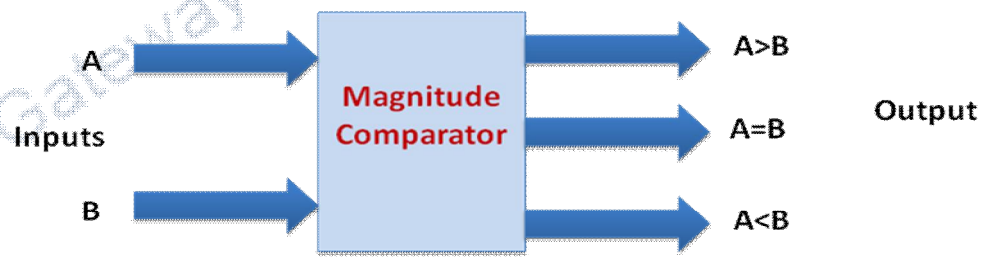
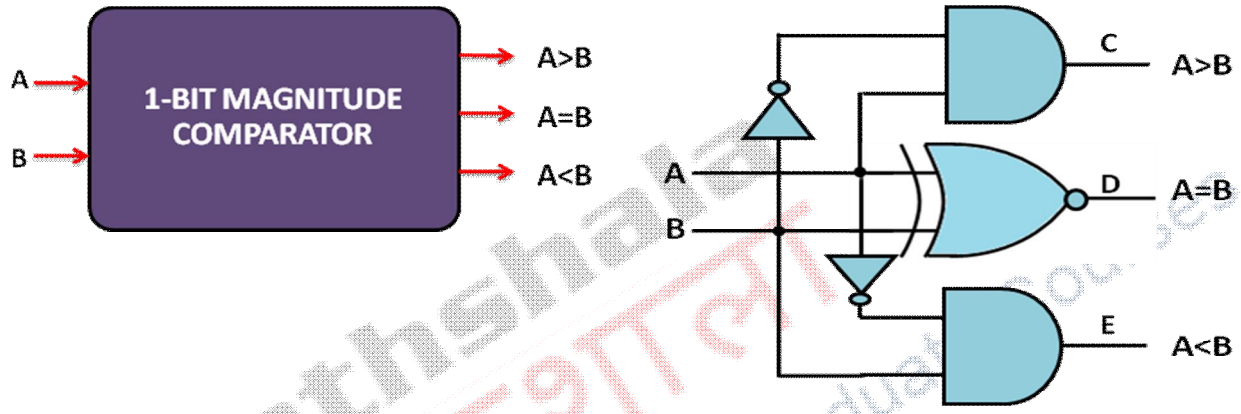


Fig. 9: Magnitude comparator.

Magnitude comparator as the name suggest compares only the magnitude of the two binary numbers and not the sign. It can also be used to indicate the equality of the input numbers. In addition, the magnitude comparators has two additional outputs to indicate whether input A is greater that input B and whether input A is less than input B as shown in fig.9.

1-bit magnitude comparator



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

Figure 10: 1-bit magnitude comparator

In 1-bit mag. Comparator, A, B are two 1 bit number and there are three outputs corresponding to A>B, A=B and A<B respectively as shown in fig. 10. The Boolean expressions for 1-bit magnitude comparator are

$$C = A \bullet \overline{B}$$

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

$$E = \overline{A} \bullet B$$

For implementation of this comparator, XNOR gate, AND gates and inverters are also used. It is very easy to derive the Boolean expressions from the truth table by visual inspection.

2-bit magnitude 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.

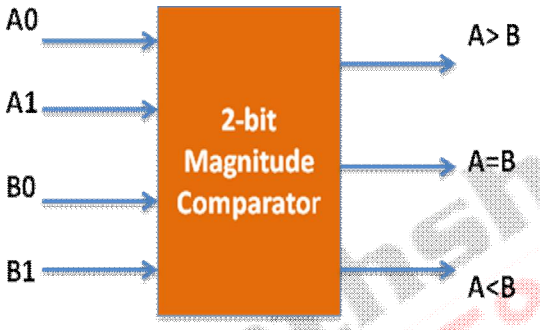


Figure 11: 2-bit magnitude comparator

The fig. 11 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 (A>B), (if A = B) and (A<B).

Let us prepare the K-map for each output so as to implement the 2-bit magnitude comparator.

Inputs				Outputs		
A1	A0	B1	BO	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

The K-map and Boolean expression for the outputs A>B, A=B and A>B of comparator are shown in fig. 12.

	B1'B0'	B1'B0	B1B0	B1B0'
A1'A0'				
A1'A0	1			
A1A0	1	1		1
A1A0'	1	1		

A>B: = A1B1' + A1'A0B1'B0'+A1A0B1B0'
= A1B1' + A0B0' (A1'B1'+A1AB1)
= A1B1' + A0B0' Y1

	B1'B0'	B1'B0	B1B0	B1B0'
A1'A0'	1			
A1'A0		1		
A1A0			1	
A1A0'				1

A=B: = A1'A0'B1'B0'+A1'A0B1'B0+A1A0B1B0+A1A0'B1B0'
= (A1'B1'+A1B1) (A0'B0'+A0B0)
= Y1Y0

	B1'B0'	B1'B0	B1B0	B1B0'
A1'A0'		1	1	1
A1'A0			1	1
A1A0				
A1A0'			1	

A<B: = A1'B1+A1'A0'B1'B0+A1A0'B1B0
= A1'B1'+A0'B0 (A1'B1'+A1B1)
= A1'B1'+A0'B0 Y1

Figure 12: Obtaining Boolean expression for 2-bit magnitude comparator

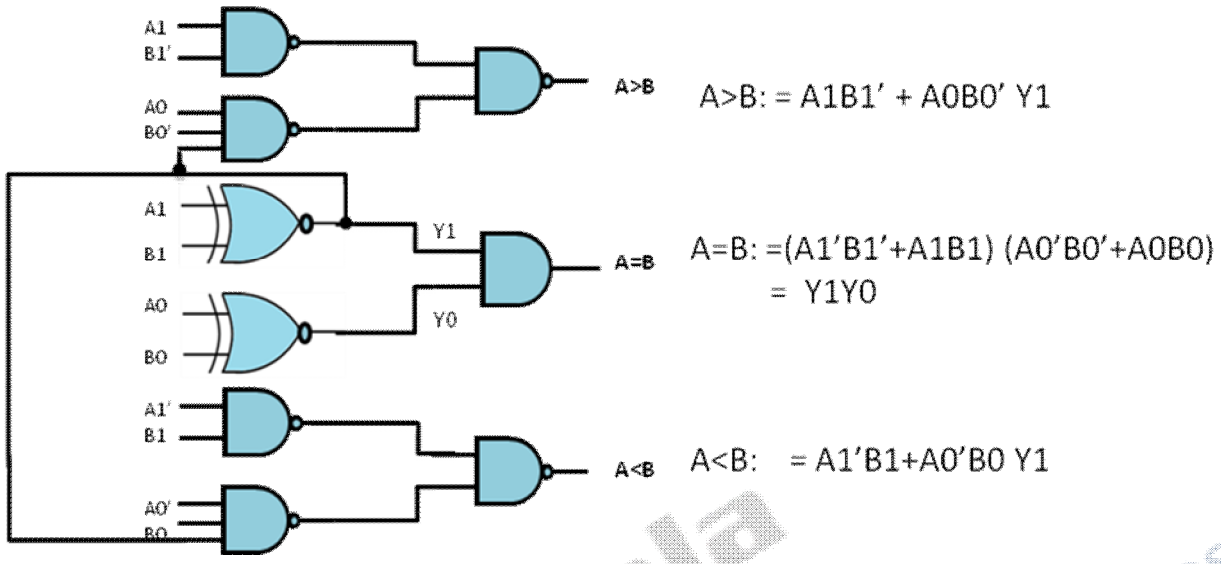


Figure 13: Logic diagram for 2-bit magnitude comparator

By using above obtained Boolean equation for each output, the logic diagram can be implemented by using four NOT gates, seven AND gates, two OR gates and two Ex-NOR gates. Figure 13 shows the logic diagram of a 2-bit comparator using basic logic gates. It is also possible to construct this comparator by cascading of two 1-bit comparators.

6.3 A 4-bit magnitude comparator

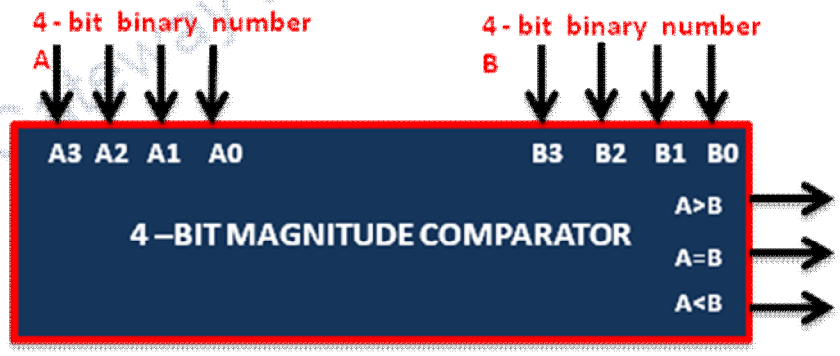


Figure 14: Logic symbol of 4-bit magnitude comparator

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

compares each of these bits in one number with bits in that of other number and produces one of the following outputs as A = B, A < B and A>B. The output function table of 4-bit magnitude comparator is shown in Figure 15.

INPUT				OUTPUTS		
A3 B3	A2 B2	A1 B1	A0 B0	A>B	A=B	A<B
A3 >B3	X	X	X	1	0	0
A3 <B3	X	X	X	0	0	1
A3 =B3	A2>B2	X	X	1	0	0
A3 =B3	A2<B2	X	X	0	0	1
A3 =B3	A2=B2	A1>B1	X	1	0	0
A3 =B3	A2=B2	A1<B1	X	0	0	1
A3 =B3	A2=B2	A1=B1	A0>B0	1	0	0
A3 =B3	A2=B2	A1=B1	A0<B0	0	0	1
A3 =B3	A2=B2	A1=B1	A0=B0	0	1	0

Figure 15: Function table of 4-bit magnitude comparator

If A3 = 1 and B3 = 0, then A is greater than B (A>B), Or If A3 and B3 are equal, and if A2 = 1 and B2 = 0, then A > B. Or If A3 and B3 are equal & A2 and B2 are equal, and if A1 = 1, and B1 = 0, then A>B. Or If A3 and B3 are equal, A2 and B2 are equal and A1 and B1 are equal, and if A0 = 1 and B0 = 0, then A > B.

IC 7485

ICs are available in TTL as well as CMOS logic family ICs and some of these ICs include IC 7485 (4-bit comparator), IC 4585 (4-bit comparator in CMOS family) and IC 74AS885 (8-bit comparator). Fig. 16 indicates an IC for 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.

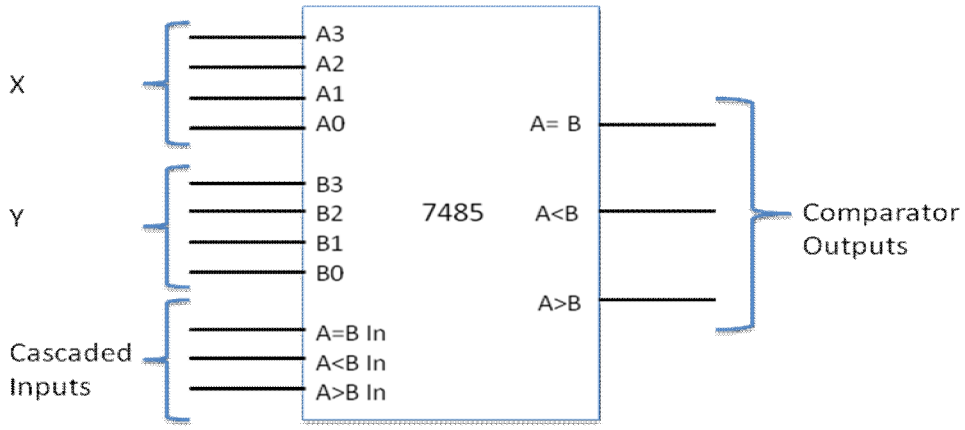


Figure 16: IC 7485- a 4 bit magnitude comparator.

6.4 Cascading comparators

Cascading inputs provide a means for expanding the comparison operation to more than four bits by cascading two or more 4-bit comparators. When two comparators are to be cascaded, the outputs of the lower-order comparator are connected to corresponding inputs of the higher-order comparator. The comparator on the left is comparing lower 4 bits and right comparator compare higher order nibble.

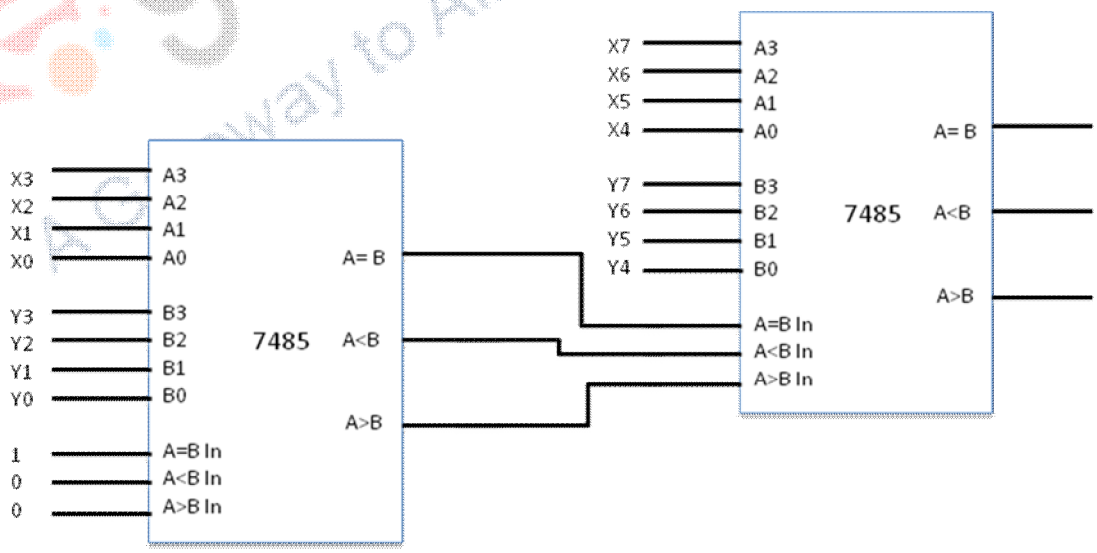


Figure 17: Cascading o0f Magnitude comparator

6.5 Applications of comparators

Comparators are used in control applications in which the binary numbers representing physical variables such as temperature, position, etc. are compared with a reference value. Then the outputs from the comparator are used to drive the actuators so as to make the physical variables closest to the set or reference value. In addition, comparators are also used in following applications.

1. Arithmetic logic unit (ALU)
2. Compare two binary words in microprocessors, computers for interfacing the external devices.
3. Analog to digital converters (ADC)
4. In Process Control Applications & for Servo-motor control
5. Password verification, biometric applications, for searching and surfing

Summary

Combinational circuits are the logic circuits in which **the outputs at any instant of time depend on the inputs present at that instant only**. In Combinational circuit, for n input variables, there are 2^n possible combinations of binary input values. There are 3 types of combinational circuits: Data processing, Data transfer and code converters.

The design of combinational circuits begins with problem definition and ends with a logic circuit or set of Boolean expressions. In many applications, logic diagram is available but correlation between input and output is not known. The analysis of combinational circuits is the reverse process of design.

Comparator is a combinational logic circuit that compares the magnitudes of two binary quantities to determine which one has the greater magnitude. An exclusive–OR gate can be used as a basic comparator. There are two types of comparators: Equality or Identity Comparator and Magnitude or inequality comparator.

Cascading inputs provide a means for expanding the comparison operation to more than four bits by cascading two or more 4-bit comparators.