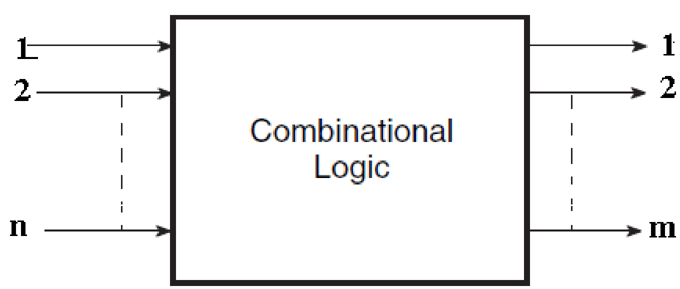
Combinational Circuit

By

Dr. P. S. Jadhav

Introduction to combinational circuits:

- A combinational circuit is the digital logic circuit in which the output depends on the combination of inputs at that point of time with total disregard to the past state of the inputs.
- The digital logic gate is the building block of combinational circuits.
- The function implemented by combinational circuit is depend upon the Boolean expressions.
- On the other hand, sequential logic circuits, consists of both logic gates and memory elements such as flip-flops.
- Figure below shows the combinational circuit having n inputs and and m outputs. The n number of inputs shows that there are 2^n possible combinations of bits at the input. Therefore, the output is expressed in terms m Boolean expressions.



3.2 MULTIPLEXER

Multiplexer means many to one. A multiplexer (MUX) is a combinational circuit which is often used when the information from many sources must be transmitted over long distances and it is less expensive to multiplex data onto a single wire for transmission.

Following Fig. 3.1 shows the general idea of a multiplexer with n input signals, m control signals and one output signal.

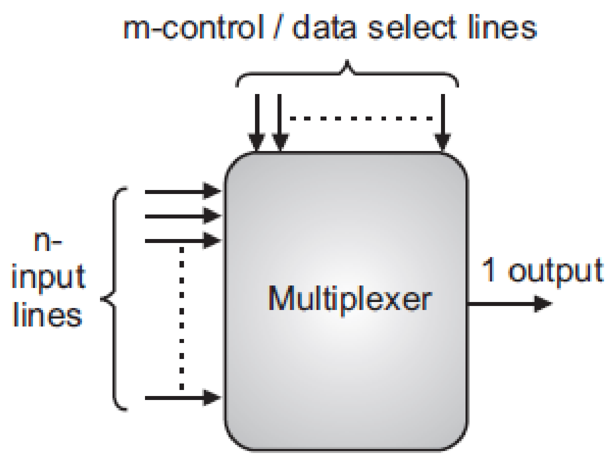


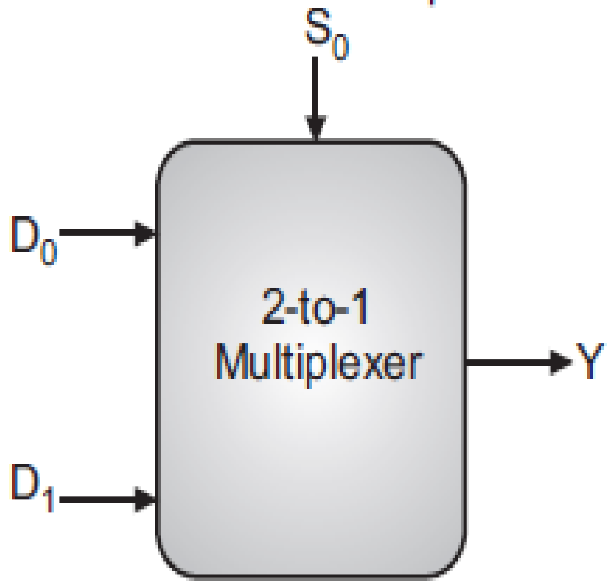
Fig. 3.1 : Logic symbol for a multiplexer

Multiplexers are also called as DATA Selector or router because it accepts several data inputs and allows only one of them to get through to the output at a time. The basic multiplexer has 'n' input lines and single output line. It also has 'm' select or control lines. The relation between number of select lines and number of data inputs are $2^m = n$.

As multiplexer selects one out of many, it is often called as 2^m to 1 line converter.

3.3 A 2-to-1 Multiplexer

A 2-to-1 multiplexer consists of two inputs D_0 and D_1, one select input S and one output Y. Depending on the select signal, the output is connected to either of the inputs. Since there are two input signals only two ways are possible to connect the inputs to the outputs, so one select is needed to do these operations.



Truth Table

Select input S_0	Output Y
0	D_0
1	D_1

$$Y = \bar{S}_0 D_0 + S_0 D_1$$

Fig. 3.2 : Logic symbol and Truth table of 2-to-1 Multiplexer

Fig. 3.2 indicates the logic symbol and truth table of 2-to-1 multiplexer. In this multiplexer, D_0 and D_1 are the data inputs, S_0 is the select line and Y is the output of the multiplexer. In a truth table, select line S_0 is shown as the input and Y is the output. When $S_0 = 0$ then the data D_0 appears at output Y and when $S_0 = 1$, the output Y receives the data D_1.

From the truth table the Boolean expression of the output is given as

$$Y = \overline{S}_0 D_0 + S_0 D_1$$

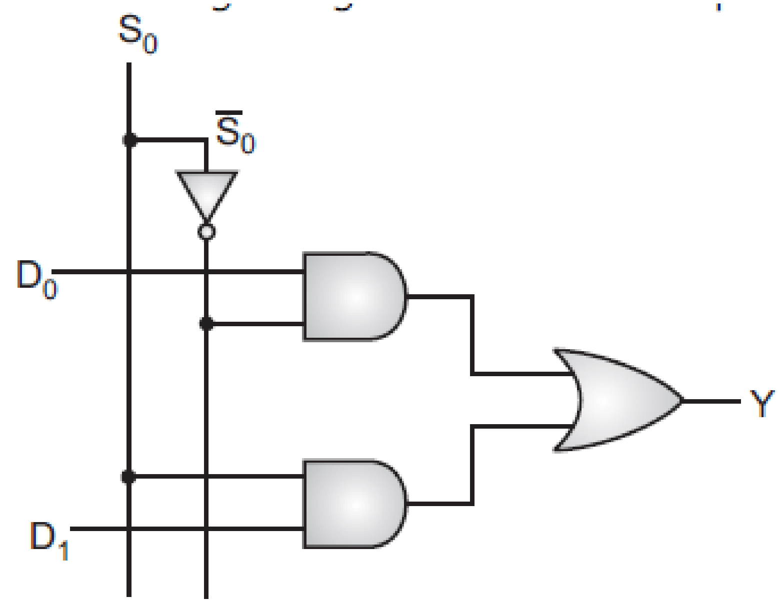


Fig. 3.3 : Logic diagram of 2-to-1 multiplexer

3.4 A 4-TO-1 MULTIPLEXER

A 4-to-1 multiplexer consists of four data input lines as D_0 to D_3, two select lines as S_0 and S_1 and a single output line Y. The select lines S_1 and S_2 select one of the four input lines to connect the output line. The particular input combination on select lines selects one of input (D_0 through D_3) to the output.

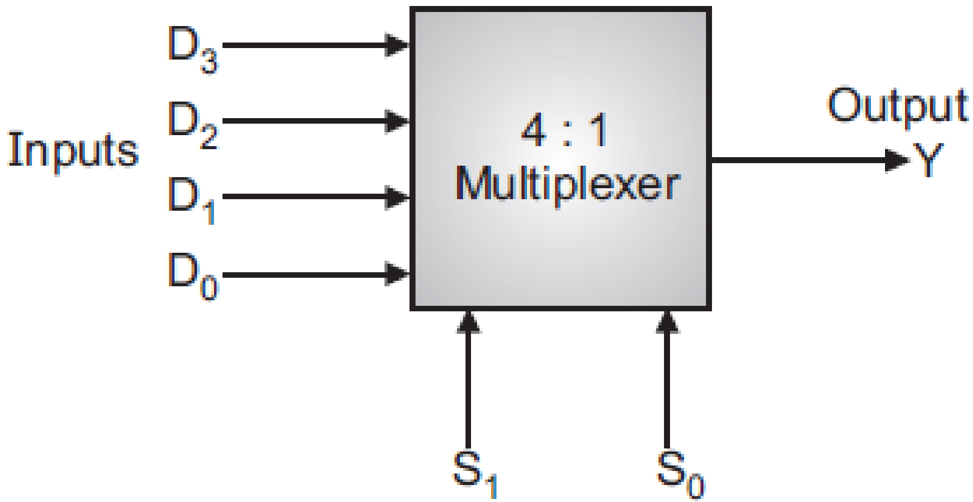


Fig. 3.4 (a) : Symbol of 4-to-1 multiplexer

Select Data Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Fig. 3.4 (b) : Truth table of 4-to-1 multiplexer

The output Y receives D_0 only when S_1 = 0 and S_0 = 0. Similarly, output Y receives D_1 only when S_1S_0 = 01. Output Y receives D_2 only when S_1S_0 = 10. Output Y receives D_3 only when S_1S_0 = 11.

From the function table, the Boolean expression can be written in SOP form

$$Y = \boxed{\bar{S}_1\,\bar{S}_0\,D_0 + \bar{S}_1\,S_0\,D_1 + S_1\,\bar{S}_0\,D_2 + S_1\,S_0\,D_3}$$

Select Data Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Fig. 3.4 (b) : Truth table of 4-to-1 multiplexer

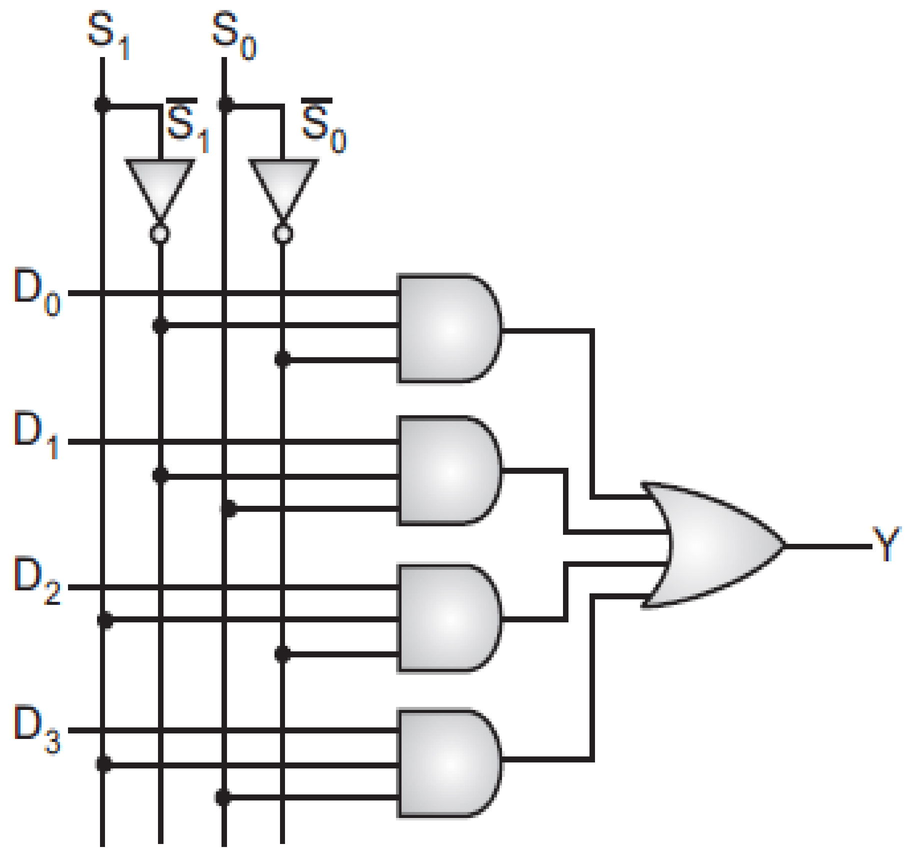


Fig. 3.5 : Logic diagram of 4-to-1 multiplexer

3.5 A 8-TO-1 MULTIPLEXER

An 8-to-1 multiplexer consists of eight data inputs D_0 through D_7, three input select lines S_2 through S_0 and a single output line Y. Depending on the select lines combinations, multiplexer decodes the inputs.

Fig. 3.6 (a) shows the block diagram of an 8-to-1 multiplexer. Since the number data bits given to the MUX are eight then 3 bits ($2^3 = 8$) are needed to select one of the eight data bits.

The truth table for an 8-to-1 multiplexer is given below with eight combinations of inputs so as to generate each output corresponding to input.

For example, if $S_2 = 0$, $S_1 = 1$ and $S_0 = 0$ then the data output Y is equal to D_2. Similarly the data outputs D_0 to D_7 will be selected through the combinations of S_2, S_1 and S_0 as shown in Fig. 3.6 (a).

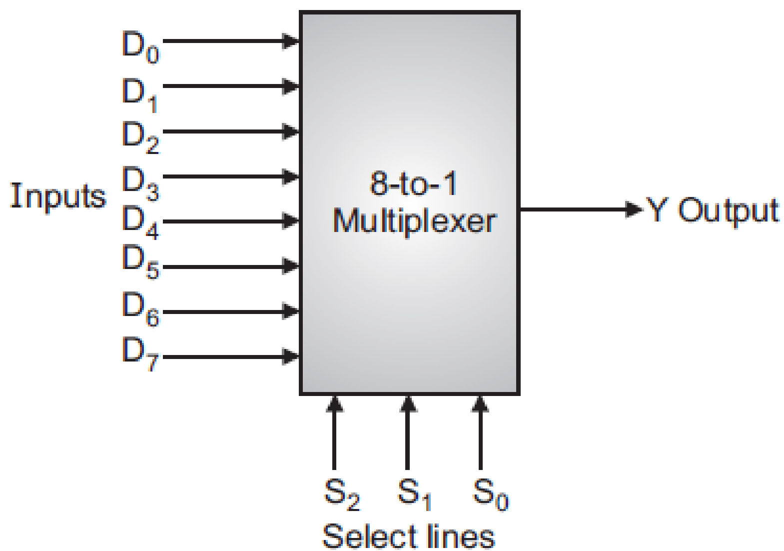


Fig. 3.6 (a) : Symbol of 8-to-1 multiplexer

Select Data Inputs			Output
S_2	**S_1**	**S_0**	**Y**
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Fig. 3.6 (b) : Truth table of 8-to-1 multiplexer

From the above truth table, the Boolean equation for the output is given as

$$Y = D_0 \overline{S_2}\,\overline{S_1}\,\overline{S_0} + D_1 \overline{S_2}\,\overline{S_1}\,S_0 + D_2 \overline{S_2}\,S_1\,\overline{S_0} + D_3 \overline{S_2}\,S_1\,S_0$$

$$D_4 S_2\,\overline{S_1}\,\overline{S_0} + D_5 S_2\,\overline{S_1}\,S_0 + D_6 S_2\,S_1\,\overline{S_0} + D_7 S_2\,S_1\,S_0$$

From the above Boolean equation, the logic circuit diagram of an 8-to-1 multiplexer can be implemented by using 8 AND gates, 1 OR gate and 7 NOT gates as shown in Fig. 3.7.

S_2 S_1 S_0

Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Fig. 3.6 (b) : Truth table of 8-to-1 multiplexer

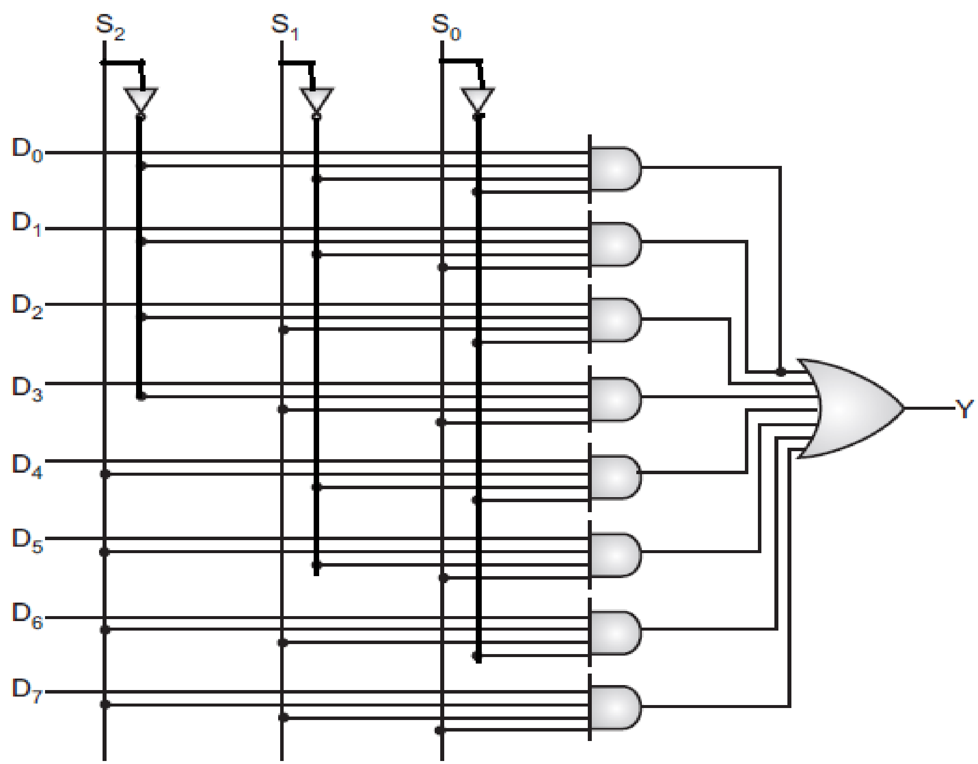


Fig. 3.7 : Logic diagram of 8-to-1 multiplexer

3.6 APPLICATIONS OF MULTIPLEXERS

Multiplexer or data selectors are combinational circuits which transfer data from many sources to output under the control of data select lines.

1. Data routing
2. Data bussing
3. Cable TV signal distribution
4. Telephone network
5. Sharing printer/resources

3.7 DEMULTIPLEXER

Demultiplexer means one into many. A demultiplexer reverses the multiplexing operation. In other words, the demultiplexer takes one data input source and selectively distributes it to a given number of output lines. For this reason, demultiplexer is also known as **data distributor**. It also has '**m**' select lines for selecting the desired output for the input data as shown in Fig. 3.8. The mathematical relationship between select lines and '**n**' output lines is : $2^m = n$

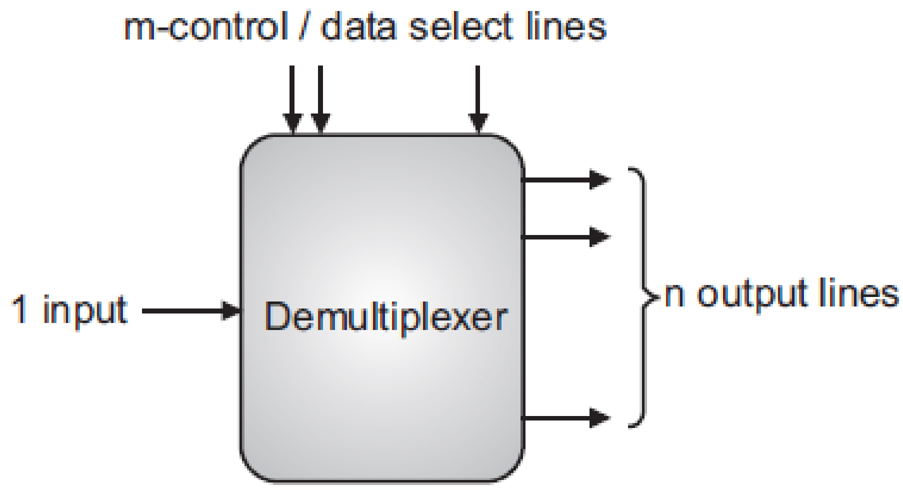


Fig. 3.8 : Logic symbol of basic demultiplexer

As a demultiplexer takes data from one input line and distributes over a 2^m output line, hence it is often referred to as 1 to 2^m **line converter**. There are four basic types of demultiplexers : 1-to-2 demultiplexer, 1-to-4 demultiplexer, 1-to-8 demultiplexer and 1-to-16 demultiplexer.

3.8 A 1-TO-2 DEMULTIPLEXER

A 1-to-2 demultiplexer consists of one input line, two output lines and one select line. The signal on the select line helps to switch the input to one of the two outputs. Fig. 3.9 (a) shows the block diagram of a 1-to-2 demultiplexer.

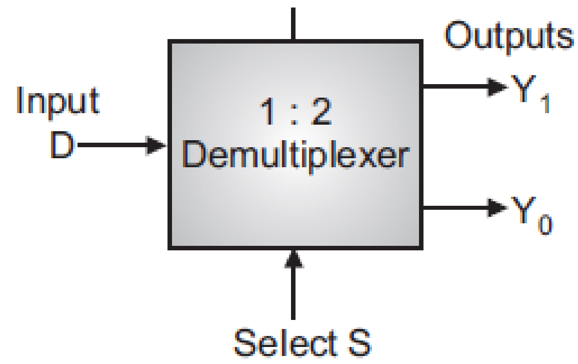


Fig. 3.9 (a) : Symbol of 1-to-2 demultiplexer

Select input	Outputs	
S_0	Y_0	Y_1
0	D	0
1	0	D

Fig. 3.9 (b) : Truth table of 1-to-2 demultiplexer

The Boolean expressions for the outputs are

$$Y_0 = \bar{S}_0 D$$
$$Y_1 = S_0 D$$

In 1-to-2 demultiplexer, with $S_0 = 0$ the Y_0 output of demultiplexer receives the input data. Similarly when S_0 becomes '1', the Y_1 output of demultiplexer receives the input data. Thus the select or control line selects the desired output to which the input data is transferred or distributed. Hence, demultiplexer is also known as **data distributor**.

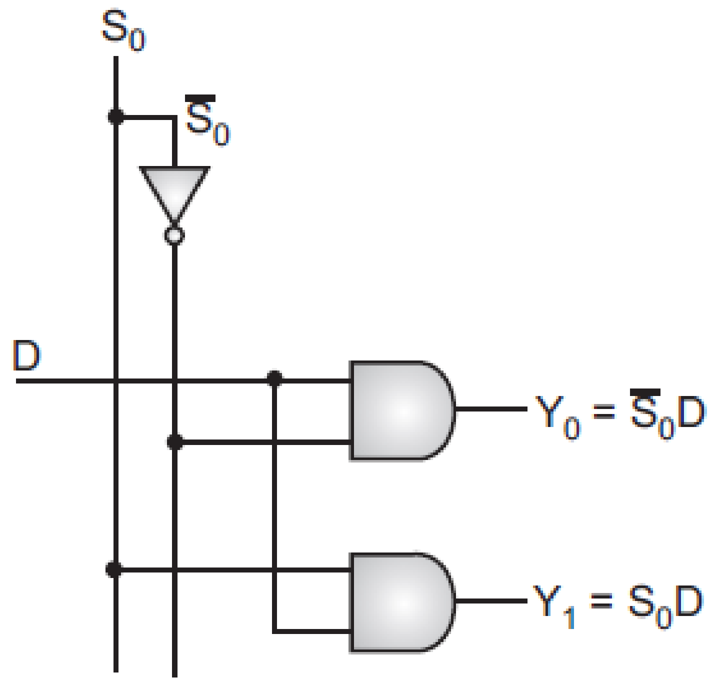


Fig. 3.10 : Logic diagram of 1-to-2 demultiplexer

3.9 A 1-TO-4 DEMULTIPLEXER

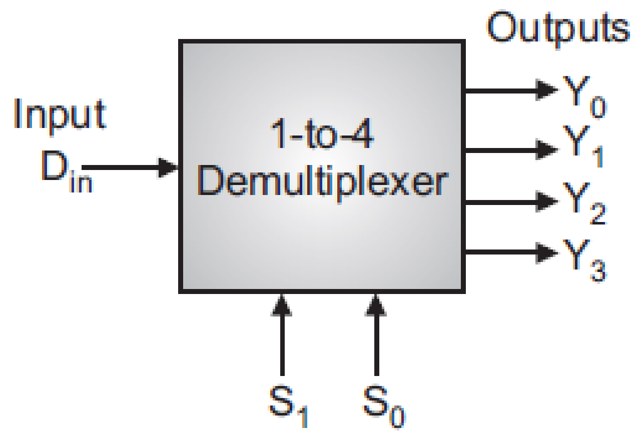


Fig. 3.11 (a) : Symbol of 1-to-4 demultiplexer

Data input	Select input		Outputs			
D	S_1	S_0	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0

Fig. 3.11 (b) : Truth table of 1-to-4 demultiplexer

The Boolean expressions for the outputs are :

$$Y_0 = \overline{S_1}\,\overline{S_0}\,D$$

$$Y_1 = \overline{S_1}\,S_0\,D$$

$$Y_2 = S_1\,\overline{S_0}\,D$$

$$Y_3 = S_1\,S_0\,D$$

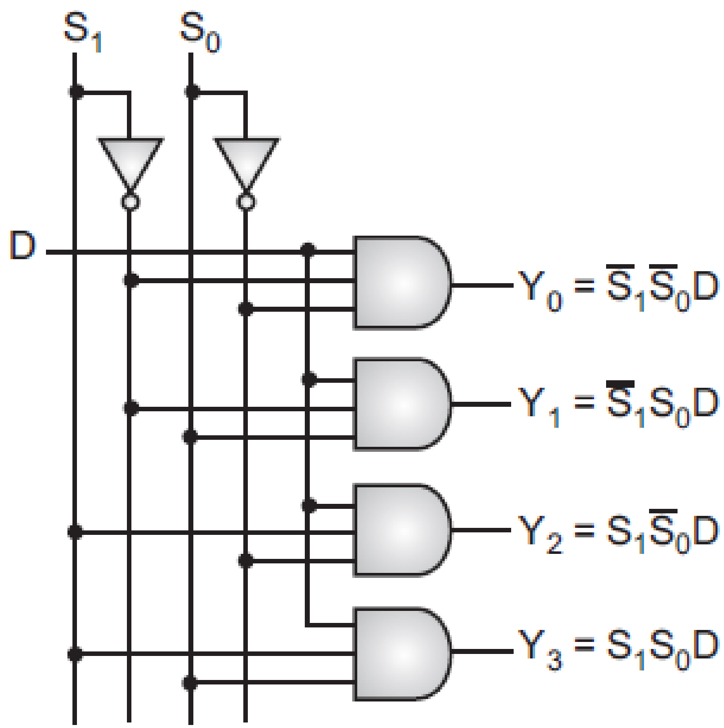


Fig. 3.12 : Logic diagram of 1-to-4 demultiplexer

3.10 1-TO-8 DEMULTIPLEXER

Fig. 3.13 (a) shows the block diagram of a 1-to-8 demultiplexer that consists of single input D, three select inputs S_2, S_1 and S_0 and eight outputs from Y_0 to Y_7.

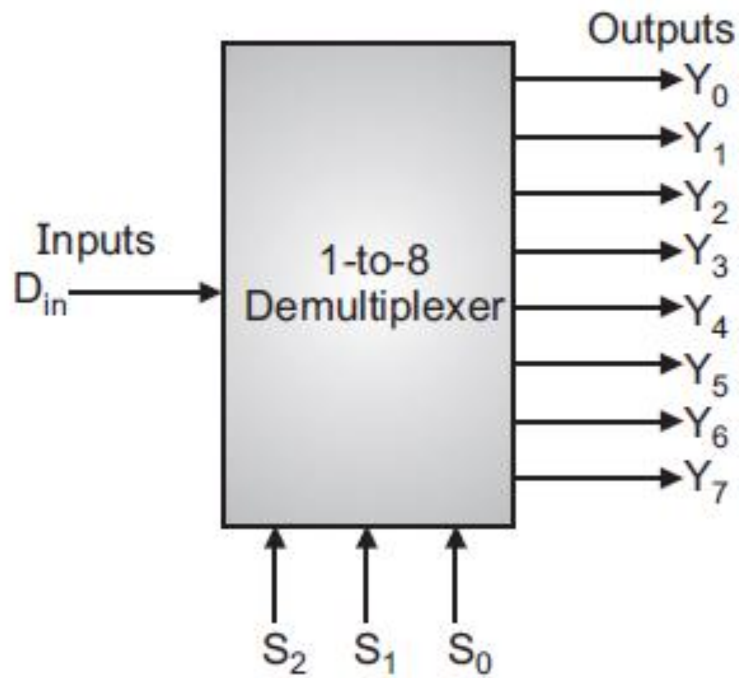


Fig. 3.13 (a) : Symbol of 1-to-8 demultiplexer

Data input	Select inputs			Outputs							
D	**S_2**	**S_1**	**S_0**	**Y_7**	**Y_6**	**Y_5**	**Y_4**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
D	0	0	0	0	0	0	0	0	0	0	D
D	0	0	1	0	0	0	0	0	0	D	0
D	0	1	0	0	0	0	0	0	D	0	0
D	0	1	1	0	0	0	0	D	0	0	0
D	1	0	0	0	0	0	D	0	0	0	0
D	1	0	1	0	0	D	0	0	0	0	0
D	1	1	0	0	D	0	0	0	0	0	0
D	1	1	1	D	0	0	0	0	0	0	0

Fig. 3.13 (b) : Truth table of 1-to-8 demultiplexer

From this truth table, the Boolean expressions for all the outputs can be written as follows.

$$Y_0 = D\,\overline{S_2}\,\overline{S_1}\,\overline{S_0} \qquad Y_4 = D\,S_2\,\overline{S_1}\,\overline{S_0}$$

$$Y_1 = D\,\overline{S_2}\,\overline{S_1}\,S_0 \qquad Y_5 = D\,S_2\,\overline{S_1}\,S_0$$

$$Y_2 = D\,\overline{S_2}\,S_1\,\overline{S_0} \qquad Y_6 = D\,S_2\,S_1\,\overline{S_0}$$

$$Y_3 = D\,\overline{S_2}\,S_1\,S_0 \qquad Y_7 = D\,S_2\,S_1\,S_0$$

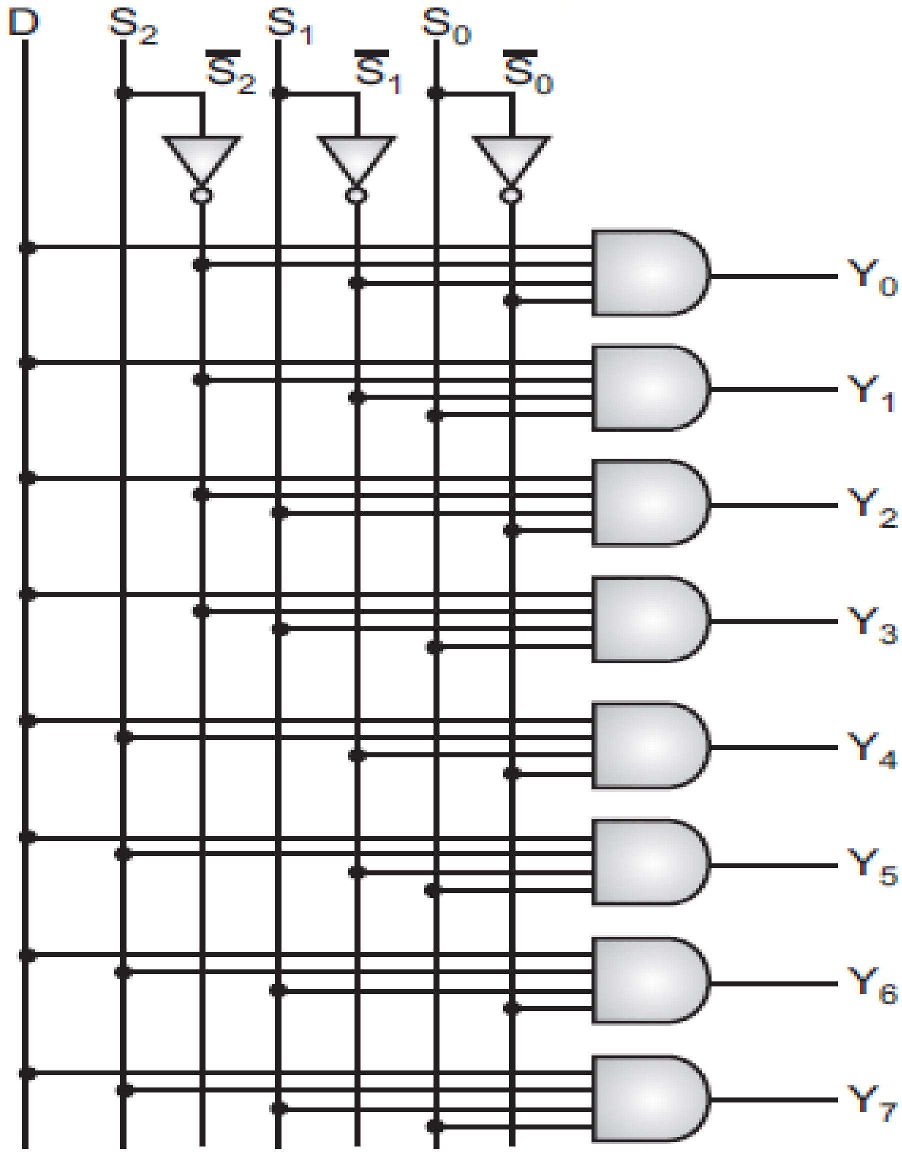


Fig. 3.14 : Logic diagram of 1-to-8 demultiplexer

3.11 BASICS OF ENCODER

The process of converting from human readable code to machine readable code i.e. binary is known as **encoding**. An encoder is a combinational circuit that converts more familiar numbers, symbols or characters into binary code. Encoders are widely used in many applications such as calculator, mobile phones, computer, ATM etc. An encoder has a number of input lines but only one of them is activated at a time representing a digit or character and produces a binary code depending on which input is activated. Fig. 3.15 is the logic symbol of encoder with 'm' inputs and 'n' outputs.

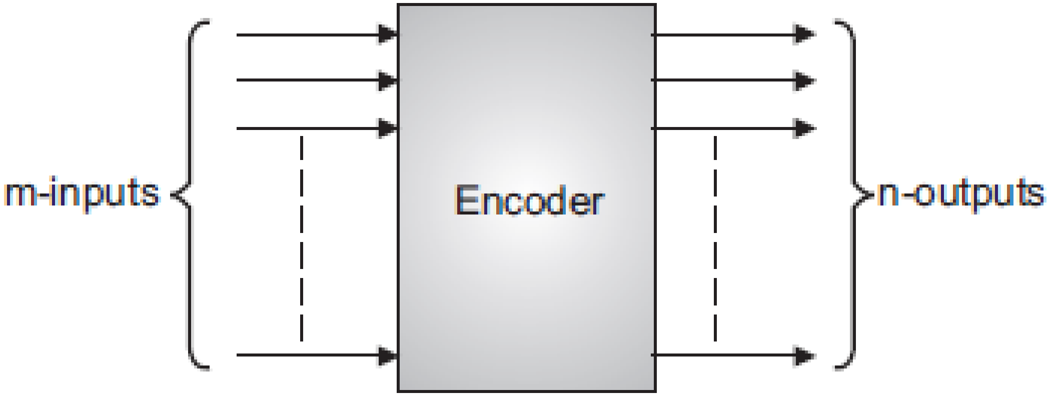


Fig. 3.15 : Symbol of basics of encoder

3.12 DECIMAL TO BCD ENCODER

A decimal to BCD (binary coded decimal) encoder is also known as 10-line to 4-line encoder. It accepts 10 inputs and produces a 4-bit output corresponding to the activated decimal input. Fig. 3.16 (a) shows the logic symbol of decimal to BCD encoder. The BCD (8421) code is listed in Fig. 3.16 (b).

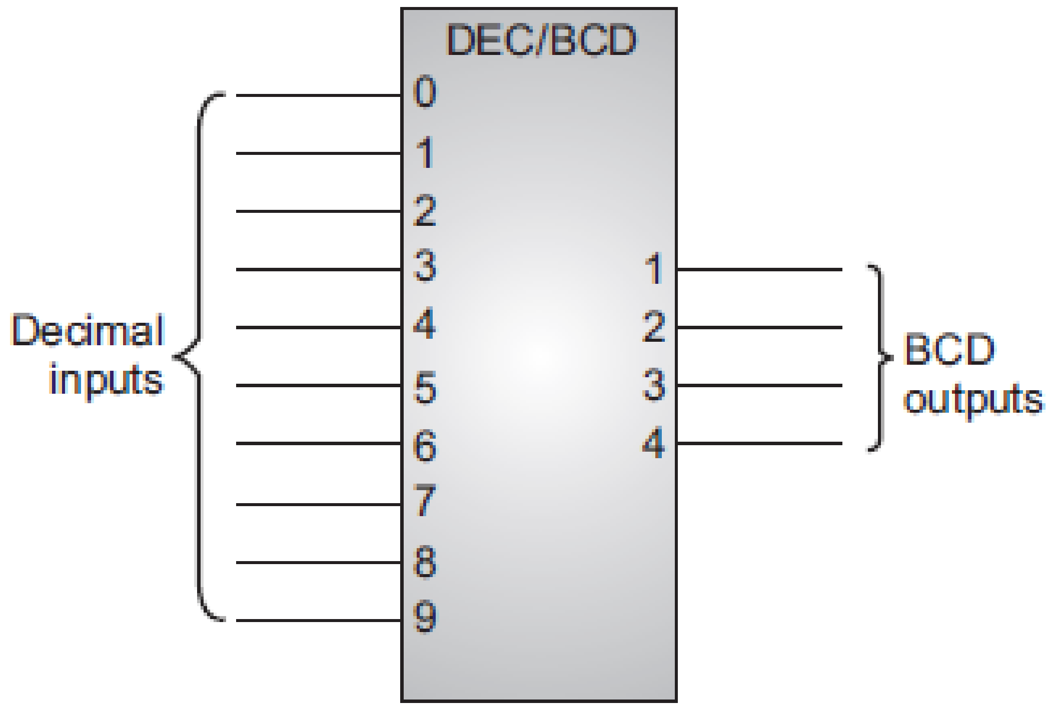


Fig. 3.16 (a) : Symbol of decimal to BCD encoder

Decimal Digit	BCD Code			
	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Fig. 3.16 (b) : Truth table of decimal to BCD encoder

From this table you can determine the relationship between each BCD bit and the decimal digits in order to analyze the logic. For instance, the most significant bit of the BCD code A_3 is always A_1 for decimal digit 8 or 9. An OR expression for bit A_3 in terms of the decimal digits can therefore be written as :

$$A_3 = 8 + 9$$

Bit A_2 is always A_1 for decimal digits 4, 5, 6 or 7 and can be expressed as an OR function as :

$$A_2 = 4 + 5 + 6 + 7$$

Bit A_1 is always A_1 for decimal digits 2, 3, 6 or 7 and can be expressed as :

$$A_1 = 2 + 3 + 6 + 7$$

Finally, A_0 is always A_1 for decimal digits 1, 3, 5, 7 or 9. The expression for A_0 is :

$$A_0 = 1 + 3 + 5 + 7 + 9$$

Now let us implement the logic circuitry required for encoding each decimal digit to a BCD code by using the logic expressions just developed. It is simply a matter of ORing the appropriate decimal digit input lines to form each BCD output. The basic encoder logic resulting from these expressions is shown in Fig. 3.17.

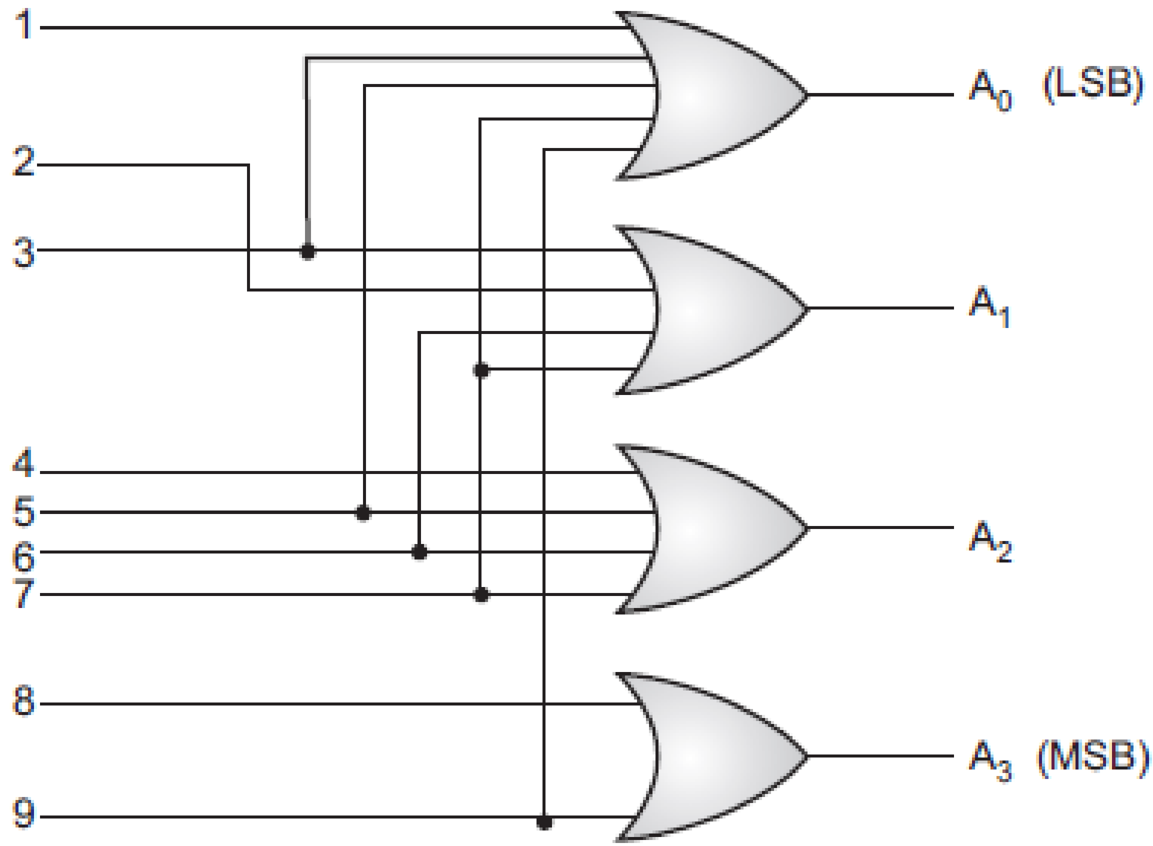


Fig. 3.17 : Logic diagram of decimal to BCD encoder

When a HIGH appears on one of the decimal digit input lines, the appropriate levels occur on the four BCD output lines. For instance, if input line 9 is HIGH, assuming all other input lines are LOW, this condition will produce a HIGH on outputs A_0 and A_3 and LOW on outputs A_1 and A_2, which is the BCD code, 1001 for decimal 9.

3.13 BASICS OF DECODER

Decoder is a combinational logic circuit that converts a binary code into the desired output signals. It is called decoder because it performs the reverse process of encoder. The process of converting binary input code into desirable output is known as **decoding**.

Fig. 3.18 is the logic symbol of decoder with 'n' inputs and 'm' outputs. In short, it is multiple input and multiple output device with proper conversion system. Note that decoder performs the reverse operation of the encoder.

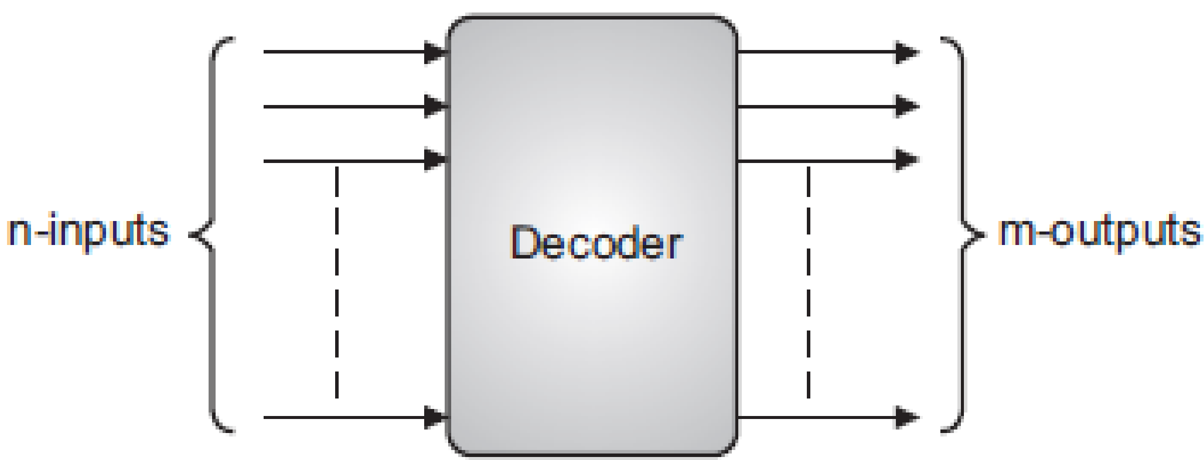
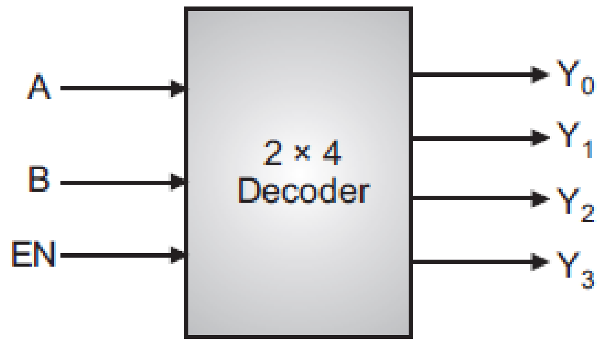


Fig. 3.18 : Symbol of basics of decoder

The basic function of a decoder is used to detect the presence of a specified combination of bits (code) on its inputs and to indicate the presence of that code by a specific output level.

3.14 2-TO-4 BINARY DECODER

In a 2-to-4 binary decoder, two inputs are decoded into four outputs, hence it consists of two input lines and 4 output lines. Only one output is active at any time while the other outputs are maintained at logic 0 and the output which is held active or high is determined by the two binary inputs A and B.



(a)

Fig. 3.19 (a) : Symbol of 2-to-4 binary decoder

Inputs			Outputs			
EN	**A**	**B**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Fig. 3.19 (b) : Truth table of 2-to-4 binary decoder

From the above truth table we can obtain Boolean expression for each output as :

$$Y_0 = \bar{A}\,\bar{B}$$

$$Y_1 = \bar{A}\,B$$

$$Y_2 = A\,\bar{B}$$

$$Y_3 = AB$$

These expressions can be implemented by using basic logic gates. Thus, the logic circuit design of the 2-to-4 line decoder is shown below in Fig. 3.20 which is implemented by using NOT and AND gates.

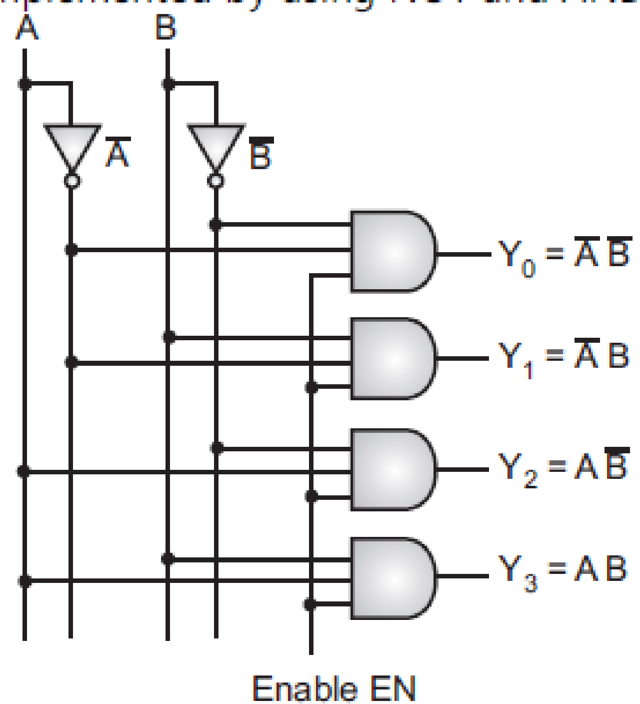


Fig. 3.20 : Logic diagram of 2-to-4 decoder

3.15 3-TO-8 DECODER

In a 3-to-8 decoder, three inputs are decoded into eight outputs. It has three inputs as A, B and C and eight outputs from Y_0 through Y_7. Based on the combinations of three inputs, only one of the eight outputs is selected.

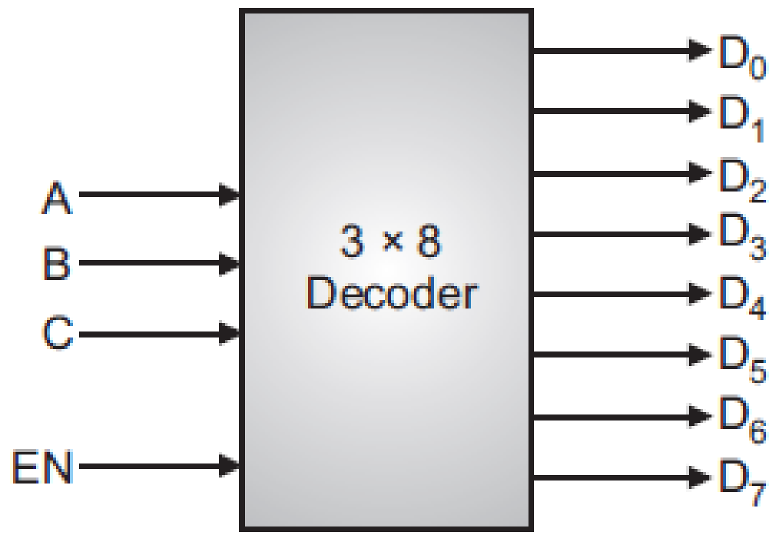


Fig. 3.21 (a) : Symbol of 3-to-8 binary decoder

Inputs				Outputs							
EN	**A**	**B**	**C**	**Y_7**	**Y_6**	**Y_5**	**Y_4**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Fig. 3.21 (b) : Truth table of 3-to-8 decoder

$Y_0 = \bar{A}\,\bar{B}\,\bar{C}$

$Y_1 = \bar{A}\,\bar{B}\,C$

$Y_2 = \bar{A}\,B\,\bar{C}$

$Y_3 = \bar{A}\,B\,C$

$Y_4 = A\,\bar{B}\,\bar{C}$

$Y_5 = A\,\bar{B}\,C$

$Y_6 = A\,B\,\bar{C}$

$Y_7 = A\,B\,C$

Also enable input activates the decoded output depending on the input data. The logic diagram of this decoder is shown in Fig. 3.22.

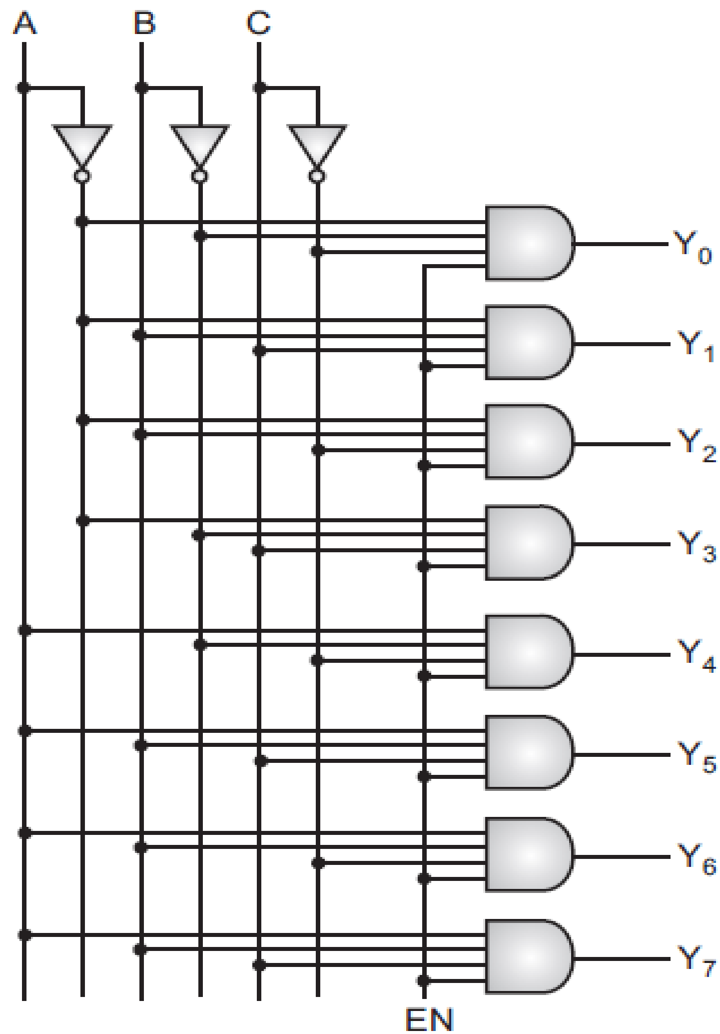


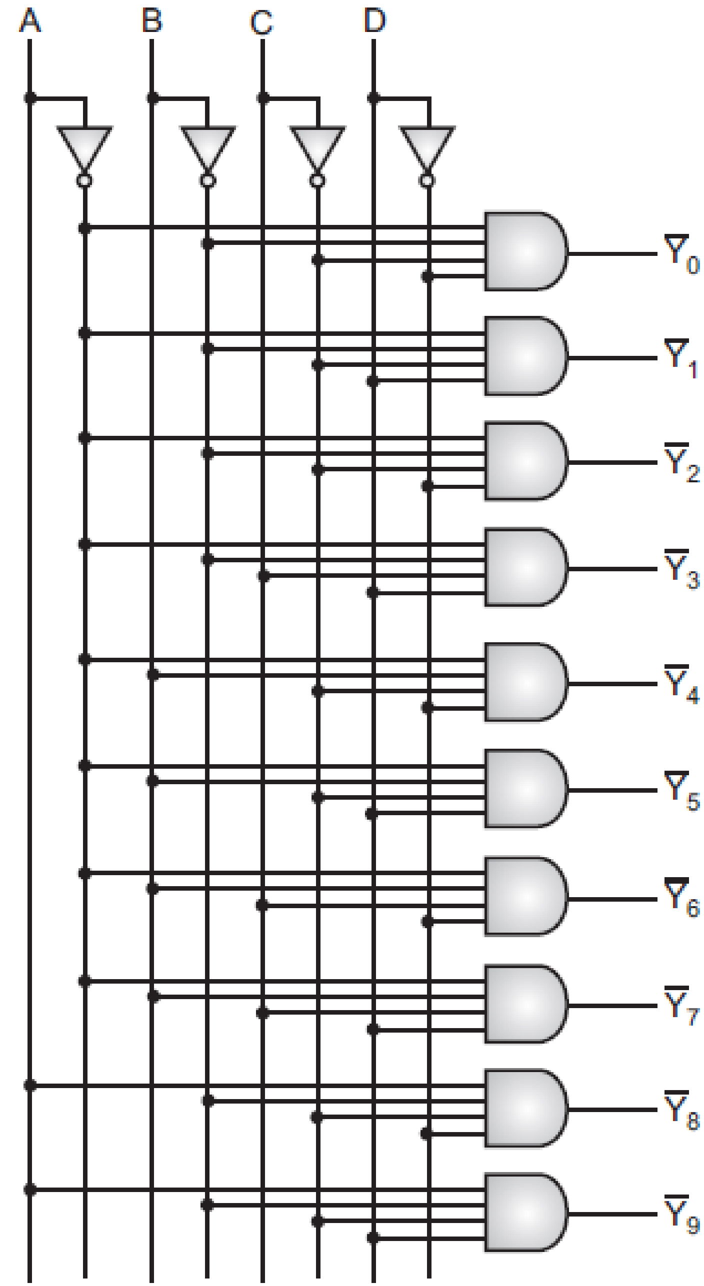
Fig. 3.22 : Logic diagram of 3-to-8 decoder

Also, 3-bit binary numbers at the input are converted to eight digits at the output (which is equivalent to octal number system), that is how it is also called as a binary-to-octal decoder.

3.16 BCD TO DECIMAL DECODER

The BCD to decimal decoder converts each BCD code (8421 code) into its decimal digit. It is frequently referred as 4 line to 10 line decoder. This decoder is similar to 4 line to 16 line, except only ten lines are used and remaining six lines are eliminated. A list of 10 BCD codes for 0 to 9 decimal numbers is given in Table 3.1. The necessary circuit is shown in Fig. 3.23. If the output AND gates are connected, we get active high output and if NAND are connected, we get active low output.

IC 7442 is a BCD to Decimal Decoder :



3.17 BCD TO SEVEN SEGMENT DECODER

BCD (Binary Coded Decimal) is an encoding scheme which represents each of the decimal numbers by its equivalent 4-bit binary pattern. Many digital types of equipment require some means for displaying information. One of the simplest and most popular methods for displaying numeric digits utilizes seven segment configuration for numbers 0 to 9 and sometimes the hex characters A to F.

Special decoders are designed to drive the seven segment display. Each segment is made up of materials that emit light when current is passed through it. Most commonly used devices include light emitting diode (LEDs). The seven segment display has 7 LEDs for the segments and one additional LED for dot i.e. used as decimal point.

There are two configurations for seven segment display. These are common anode and common cathode display.

A common cathode display [Fig. 3.24 (a)] has all the cathode terminals of its LED segments tied together. Further, this is grounded and hence is considered to be at logic 0 state. This means that in order to light up (ON) an LED, one needs to drive it high. On the other hand, a common anode display shown by Fig. 3.24 (b) has all its anode terminals connected together which is further driven high by connecting it to a positive supply voltage . Hence for this kind of display to work, one has to drive low on the cathode terminals of the individual LED segments.

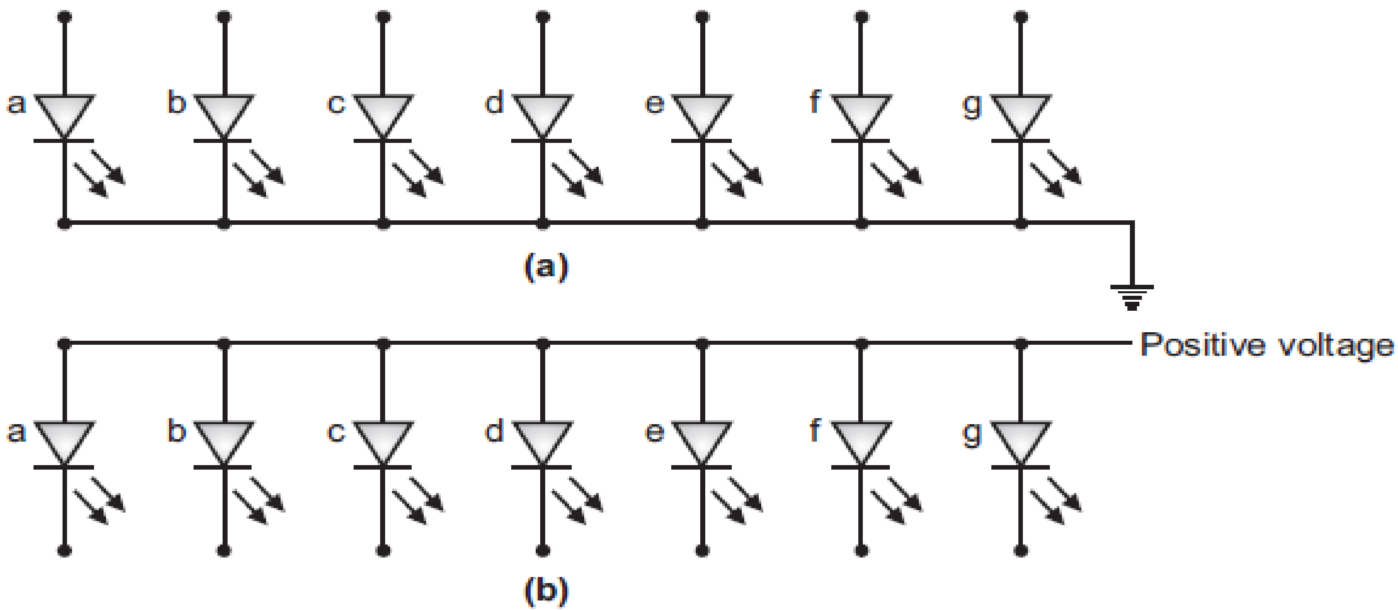


Fig. 3.24 : (a) Common cathode type display,

(b) Common anode type display

BCD to seven segment decoder is a circuit used to convert the input BCD into a form suitable for the display. It has four input lines (A, B, C and D) and 7 output lines (a, b, c, d, e, f and g) as shown in Fig. 3.25. Considering common cathode type of arrangement, the truth table for the decoder can be given as in Table 3.1.

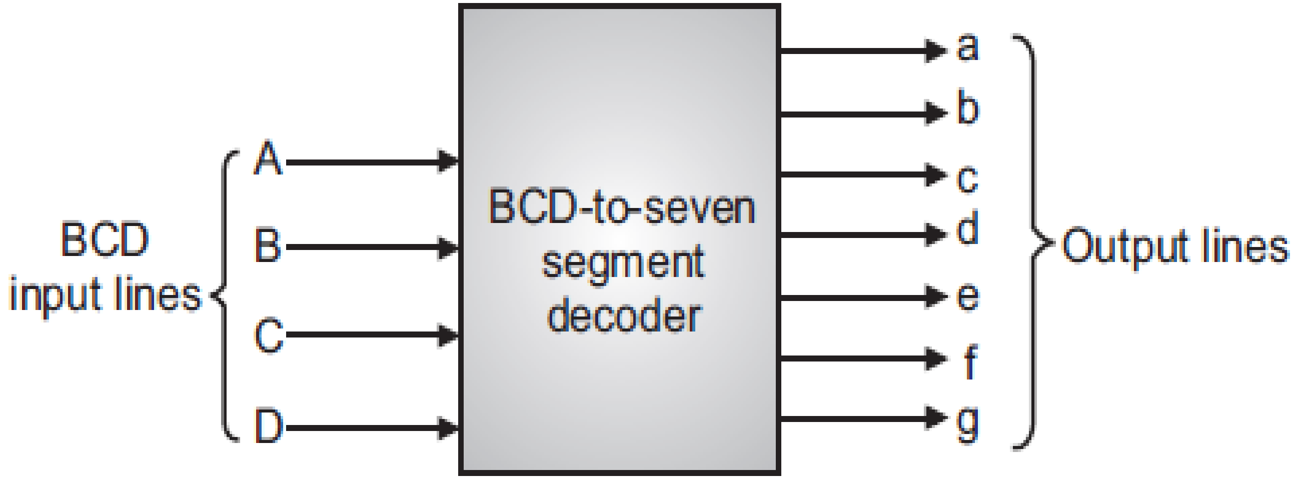


Fig. 3.25 : BCD to seven segment decoder

3.18 IC 7447

Interfacing of BCD to seven segment decoder/driver with the seven segment display is shown in Fig. 3.26. Here limiting resistors are connected between them. The resistor value decides the brightness of the segment.

IC 7447 is driving common anode displays. In addition to 4 BCD inputs and seven segment outputs, this decoder has 3 additional pins. These are used for special purpose.

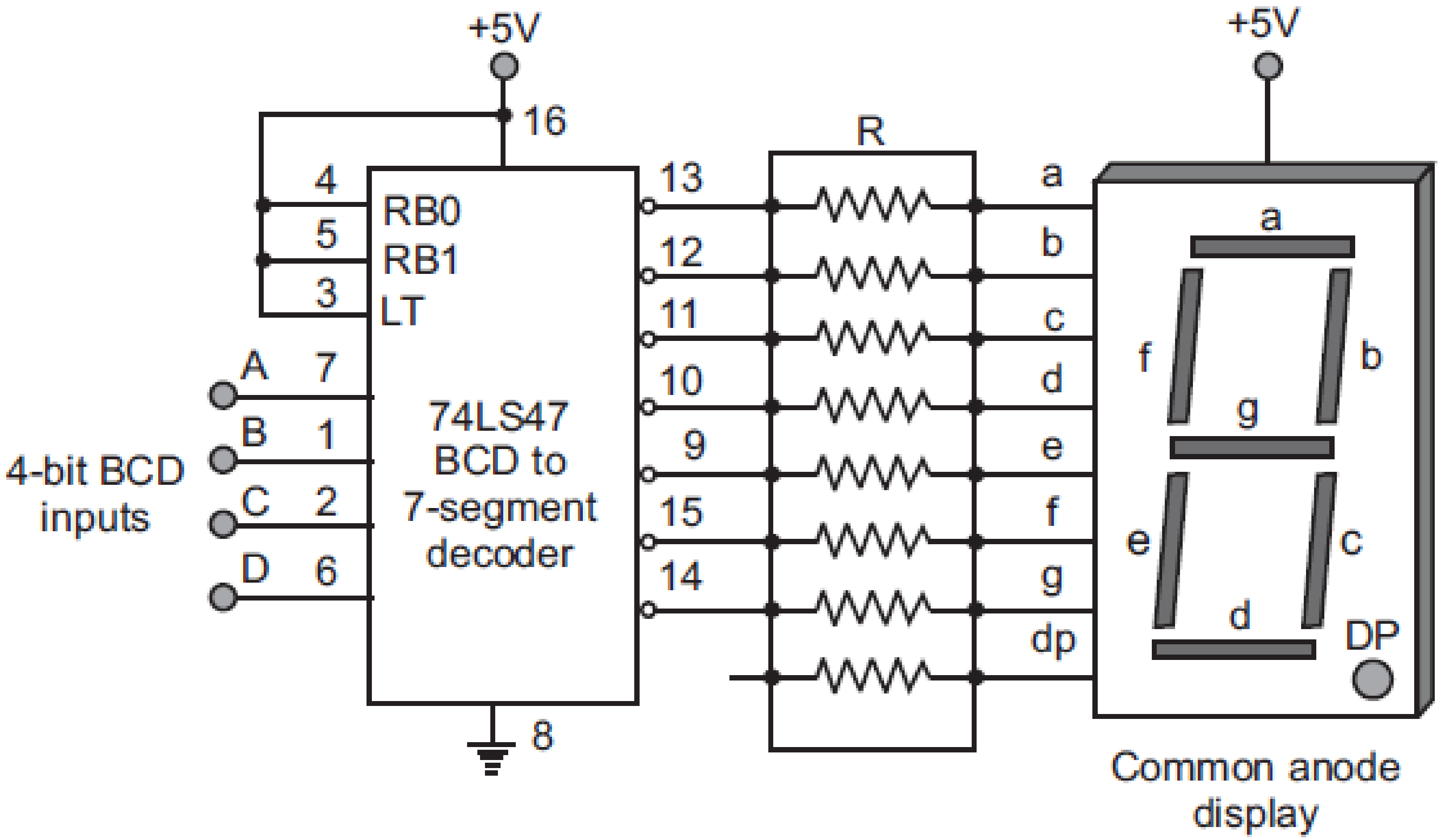


Fig. 3.26 : Practical interfacing of BCD to seven segment decoder with the 7-segment display

Common Anode

Decimal digit	Input lines				Output lines							Dispay
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0	1	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0	2
3	0	0	1	1	0	0	0	0	1	1	0	3
4	0	1	0	0	1	0	0	1	1	0	0	4
5	0	1	0	1	0	1	0	0	1	0	0	5
6	0	1	1	0	0	1	0	0	0	0	0	6
7	0	1	1	1	0	1	0	1	1	1	1	7
8	1	0	0	0	0	0	0	0	0	0	0	8
9	1	0	0	1	0	0	0	0	1	1	0	9

Thank You