Module -10

Encoder

Table of Contents

1. Introduction
2. Code converters
3. Basics of Encoder
    - 3.1 Linear encoders
        - 3.1.1 Octal to binary encoder
        - 3.1.2 Decimal to BCD encoder
        - 3.1.3 Hexadecimal to binary encoder
    - 3.2 Priority encoder
    - 3.3 Keyboard encoder
4. Applications of Encoders
5. Summary

Learning outcome –

After studying this module, you will be able to:

1. Recognize the need for code converters
2. Understand the operations of encoders
3. Design priority encoders
4. Apply the knowledge of encoders in different digital systems.

1. Introduction

Numbers are usually coded in one form or another so as to represent or use it as required. In modern communication system, data communication involves a process encoding and decoding. Encoding is the process of converting human readable characters into binary information. Decoding is the reverse process of conversion of binary or encoded format into human readable code. This process is also referred to as code conversion.

There are numerous encoding and decoding methods exist. The oldest code of all, originally employed in the landline communication for telegraph application was the Morse code. The code used by almost all computers is the ASCII (American Standard Code for Information Interchange) code. ASCII utilizes uppercase, lowercase alphabetic characters, numeral, punctuation marks and common symbols. In addition, digital system designers use Binary, BCD, Octal Hexadecimal, Decimal, Gray, Excess-3 code and several other codes. In data communication, a special form of encoding is used to convert binary digits to high to low or low to high transitions.

Encoding and decoding are also used in context with the process of analog to digital conversion or digital to analog conversion. In this sense, code conversion term can apply to any form of data including text, images, audio, video, multimedia, computer codes, sensor signals. Encoding should not be confused with encryption. In this module, the focus is on understanding the principles of encoding, designing simple encoders and studying some important applications.

2. Code conversion

Modern digital system like calculators, mobile phones, fax machines, computers and other digital gadgets use binary numbers 1 and 0 for processing and storage. However, our language of communication is in the form of decimal numbers and alphabetic characters. It is difficult for human being to understand information represented in binary. Therefore, there is a need of a group of devices to convert human readable language to machine or binary language. To achieve this task, several binary codes are developed. The process of generating binary codes is known as encoding and the reverse process is known as decoding. Let us consider a simple example to demonstrate the role of encoder and decoder. Figure-1 indicates simple block diagram of generalized digital system.

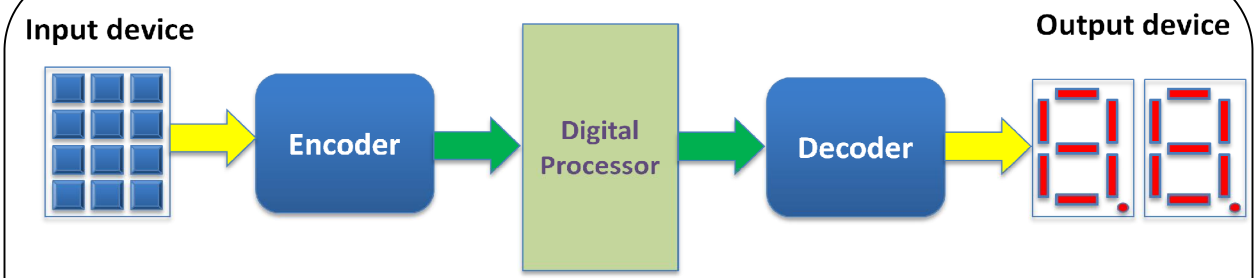

Fig. 1: Simple block diagram of Digital system

A digital system cannot be complete without encoders and decoders. It contains five basic building blocks. Keyboard and seven segment display are the input and output devices respectively. The encoders essentially receive the input from the keyboard. Many keyboards use encoders to produce a machine recognizable code i.e. binary and decoder accepts the processed binary code and converts back in the format useful for output devices like seven segment display, CRT, Printers etc.

BCD, binary, Octal, Hexadecimal, ASCII and decimal are the most common codes used in the digital systems. Thus encoders and decoders play an important role in code conversion process in digital system. One can make comparison of the above system with the functional blocks of hand held calculator. In the next section, the concept of encoder is explained in details.

3. 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 coverts more familiar numbers, symbols or character into binary code. 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. Figure 2 is the logic symbol of encoder with ómô inputs and ónô outputs. In short, it is multiple inputs and multiple outputs device with proper conversion system. Note that encoder performs the reverse operation of the decoder.

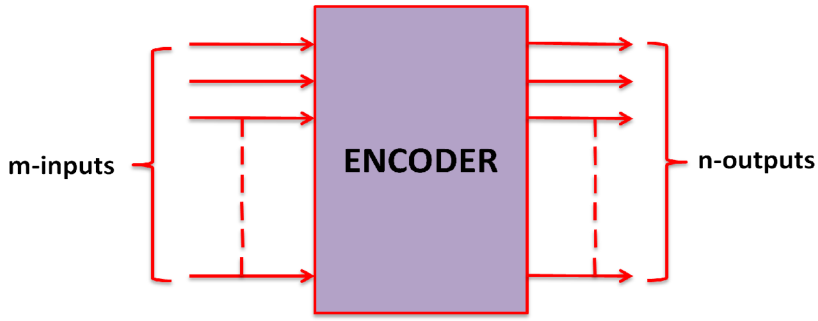

Fig. 2: Logic Symbol of Encoder

An encoder has ómô number of input lines and ónô number of output lines. The numbers of outputs (n) are always less than number of inputs (m). Here the number of inputs can have following relation

$$m \leq 2^n$$

Thus, the encoder accepts an m-bit input digital word and converts it into an n-bit another digital word. Encoders can also be devised to encode various symbols and characters for the keypads and keyboards. Some of the most commonly used encoders are ï (1) **Linear encoders** are octal to binary, Decimal to BCD and Hexadecimal to binary where normal encoding is implemented and (2) **Priority encoders**.

3.1 Liner encoders

Linear / normal encoders accepts m inputs, only one is activated at a given time and produces n-bit output code. As the numbers of output lines are ónô, it is possible to encode maximum up to 2^n inputs. In many digital systems, octal, decimal or hexadecimal numbers are used as inputs. Let us now discuss the working principal and designing of these encoders.

3.1.1 Octal to binary encoder

An octal to binary encoder is also known as 8-line to 3-line encoder. It accepts 8- inputs and produces a 3-bit output corresponding to the activated octal input. Figure-3 shows the logic symbol of Octal to binary encoder.

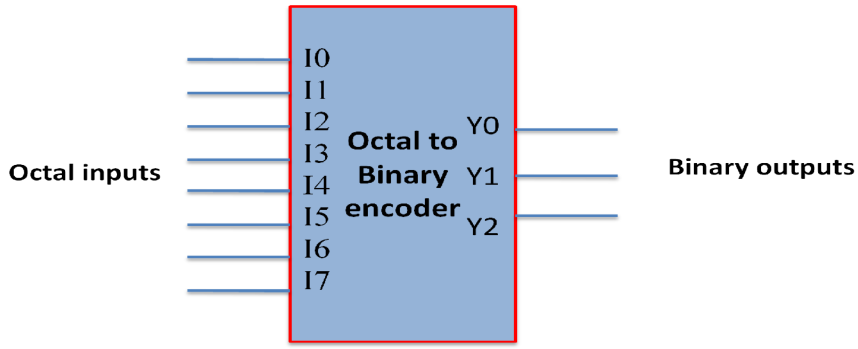

Fig. 3: Logic symbol of Octal to binary encoder.

The truth table of octal to binary encoder is shown in Table-1 There are eight inputs corresponding 8 octal inputs (I0, I1, I2, I3, I4, I5, I6, I7) and 3 Outputs (Y0, Y1, Y2). Let us look at the table carefully. Note that in encoders only one input is activated (logic 1) and other inputs are not activated (i.e. at logic 0).

TABLE-1: Truth – Table for Octal Binary Encoder

Octal Inputs								Binary Outputs		
I0	**I1**	**I2**	**I3**	**I4**	**I5**	**I6**	**I7**	**Y2**	**Y1**	**Y0**
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

In the truth table, the input variable I0 represents the least significant digit (LSD) and and I7 represents most significant digit (MSD). Similarly, in the outputs Y0 represents least significant bit (LSB) and Y2 is the most significant bit (MSB). The truth table includes only all valid combinations of the inputs. The valid combinations are those which have exactly one input equal to logic 1 while all other inputs are logic 0‟s.

As the number of inputs are 8, K-maps cannot be used to derive the Boolean expressions. The Boolean expression can be directly derived from the truth table by visual inspection. Let us obtain the Boolean expression for each output.

Output Y0 is 1 if any of the inputs I1 or I3 or I5 or I7 is 1. Then, the Boolean expression

Y0 = I1 +I3 +I5 +I7

Similarly,

Y1 = I2 +I3 +I6 +I7

and

Y2 = I4 +I5 +I6 +I7

From these Boolean expressions, the Octal to Binary Encoder can be implemented by using simply three 4 input Or gates. Figure-4 indicates the logic diagram for Octal to Binary Encoder.

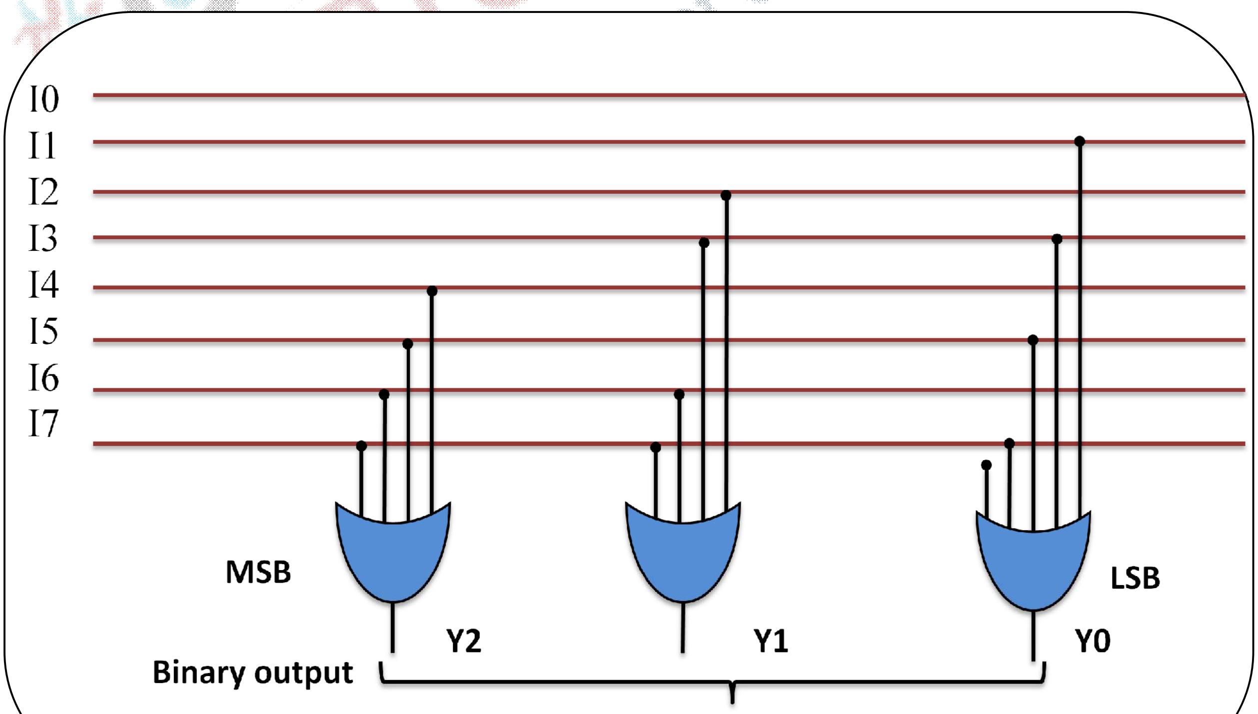

Fig. 4: Octal to Binary Encoder

This octal to binary encoder design has few drawbacks. In the encoder, at a time only one input must be active. (1) If number of active inputs is greater than one then the output is incorrect. (2) Similarly, when no inputs are active then all outputs are low which is also similar to the output generated by active input I0. To resolve first problem, one must design and use priority encoder where the output corresponding to input with higher weightage is generated. The solution to the second drawback is to generate additional output signal to check the validity of inputs. If no inputs are active, data is invalid.

3.1.2 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 a4-bit output corresponding to the activated decimal input. Figure-5 shows the logic symbol of decimal to BCD encoder.

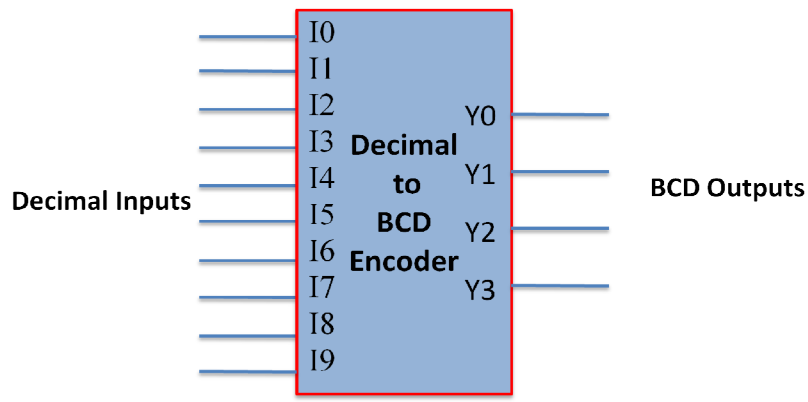

Fig. 5: Decimal to BCD Encoder

The truth table of Decimal to BCD encoder is shown in Table-2 There are ten inputs corresponding 10 decimal inputs (I0, I1, I2, I3, I4, I5, I6, I7, I8, I9) and 4 Outputs (Y0, Y1, Y2, Y3). Let us look at the table carefully. Note the encoder assumption that only one of the inputs is activated (logic 1) and other inputs are not activated (i.e. at logic 0).

TABLE-2: Truth – Table for Decimal to BCD Encoder

Decimal Inputs										BCD Outputs			
I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	Y3	Y2	Y1	Y0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

In the truth table, the input variable I0 represents the least significant digit (LSD) and and I9 represents most significant digit (MSD). Similarly, in the outputs Y0 represents least significant bit (LSB) and Y3 is the most significant bit (MSB). The truth table includes only all valid combinations of the inputs. The valid combinations are those which have exactly one input equal to logic 1 while all other inputs are logic 0ǿs.

As the number of inputs is 10, K-maps cannot be used to derive the Boolean expressions. The Boolean expression can be directly derived from the truth table by visual inspection. Let us obtain the Boolean expression for each output.

Output Y0 is 1 if any of the inputs I1 or I3 or I5 or I7 is 1. Then, the Boolean expression

Y0 = I1 +I3 +I5 +I7 +I9

Similarly,

Y1 = I2 +I3 +I6 +I7

Y2 = I4 +I5 +I6 +I7

Y3 = I8 +I9

From these Boolean expressions, the Decimal to BCD Encoder can be implemented by using simply three 4 OR gates. Figure-6 indicates the logic diagram for Decimal to BCD Encoder.

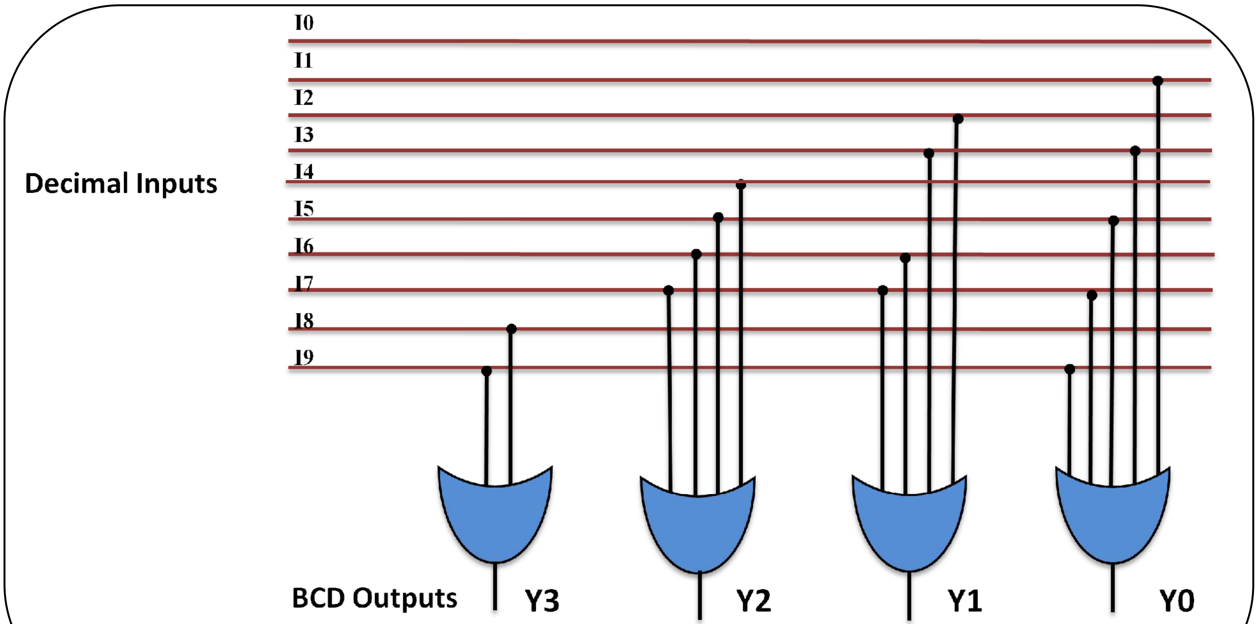

Fig.-6: Decimal to BCD Encoder

This decimal to BCD encoder design also provides few drawbacks as mentioned in Octal to Binary encoder. The solution to this problem is to use priority encoder and use of additional valid output in the encoder design.

3.2.3 Hexadecimal to Binary encoder –

It accepts sixteen inputs (0, 1, 2, é ..9, A,B,C,D,E,F) and produces 4-bit binary code corresponding to the activated Hexadecimal input. The logic diagram for Hexadecimal to Binary encoder can be developed on the similar line of Decimal to BCD encoder. The truth table for the same encoder can be prepared by simply writing four bit binary code for each activated hexadecimal input as shown in Table-3

Table- 3: Truth table for Hexadecimal to Binary encoder

Hexadecimal Inputs	Binary Outputs			
	Y3	Y2	Y1	Y0
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
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

3.2 Priority Encoder

Standard encoders generate the wrong output code when there is more than one input present at logic 1. This is one of the major disadvantages of the linear encoders. One of the simplest ways to overcome this problem is to prioritize the inputs. If there are more than one input at logic ó1ô then the actual code correspond to the highest designated priority are produced and all other inputs with lower priority will be ignored. This type of encoder is known as priority encoder.

4 to 2 priority encoder

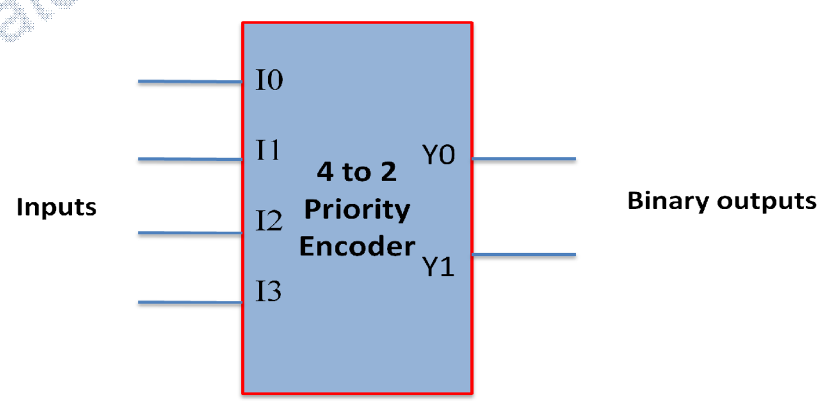

Fig. 7: 4 to 2 priority encoder

Figure ï 7 shows the logic symbol of 4 to 2 priority encoder. Priorities are given to the input lines. In this case input I3 is having highest priority than other inputs, I2 has second highest priority as compare to I1 and I0 and so on. If two or more inputs are at logic ó1ô at the same time, the input line with highest priority will be considered for generating binary output.

Table -4: Truth table of 4 to 2 priority encoder

Inputs				Outputs	
I3	I2	I1	I0	Y1	Y0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

There are four inputs I0 through I3 and two outputs Y0 and Y1. With these four inputs, the actual truth table would have been with 16 possible input conditions. But for simplicity a truth table is prepared wisely and contain only few conditions with the help of X ï do not care condition. As shown in table-4. Out of the four inputs I3 has highest priority and I0 has lowest priority. That means when I3=1 then Y1 Y0 = 11 irrespective of the other inputs.

	I1'I0'	I1'I0	I1I0	I1I0'
I3'I2'	X	0	0	0
I3'I2	1	1	1	1
I3I2	1	1	1	1
I3I2'	1	1	1	1

I2

I3

Output Y1= I3 +I2

	I1'I0'	I1'I0	I1I0	I1I0'
I3'I2'	X	0	1	1
I3'I2	0	0	0	0
I3I2	1	1	1	1
I3I2'	1	1	1	1

I2'I1

I3

Output Y0= I3 + I2'I1

Fig. 8: K-maps for the outputs of 4 to 2 line encoder

The k-maps for two outputs Y1 and Y0 are shown in figure-8. The logic ó1ô is mapped for each output for all input conditions. For output Y1, it is possible to group 8 neighboring cells to form the octet. It is possible to form two octets. From these octet, we can eliminate three variables each. The remaining unchanged variables are written in the simplified for. Thus

Output Y1= I3 +I2

Similarly we can obtain Boolean expression for output Y0 using another K-map. After mapping the ó1ôs in the map, it is possible to group neighboring cells to form Octet and Quad. Thus, we get

Output Y0= I3 + I2'I1

Using these Boolean expressions, it is possible to obtain the logic diagram for 4 to 2 priority encoder. The two binary outputs are obtained with the help of two OR gates, one AND gate and a NOT gate. The logic diagram for 4 to 2 priority encoder is indicated in figure ï 9.

If more than one input is active, the higher order input has priority over the lower order input. The higher value is encoded at the output. Further, one can introduce one more output for the priority encoder to indicate whether the output is valid or not. This output is also called as validity indicator. It is generated by simply ORing all possible inputs. If all the inputs are ó0ô, the output is invalid. The output is valid (logic-1) only when at least one input is active.

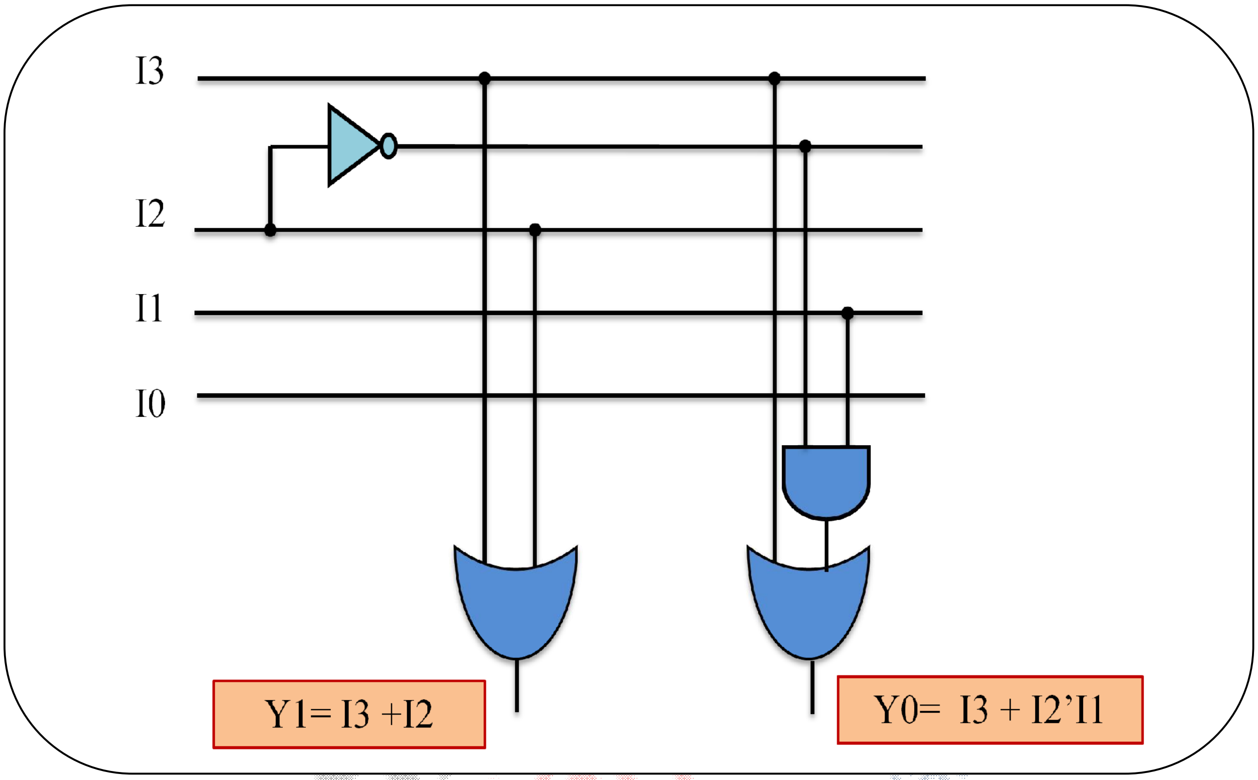

Fig. 9: Logic diagram for 4 to 2 Priority Indicator

Priority Encoders are also available in the Integrated Circuit form. The TTL priority encoders are:

IC 74148 ï Octal to Binary priority encoder
IC 74147 ï Decimal to BCD priority encoder

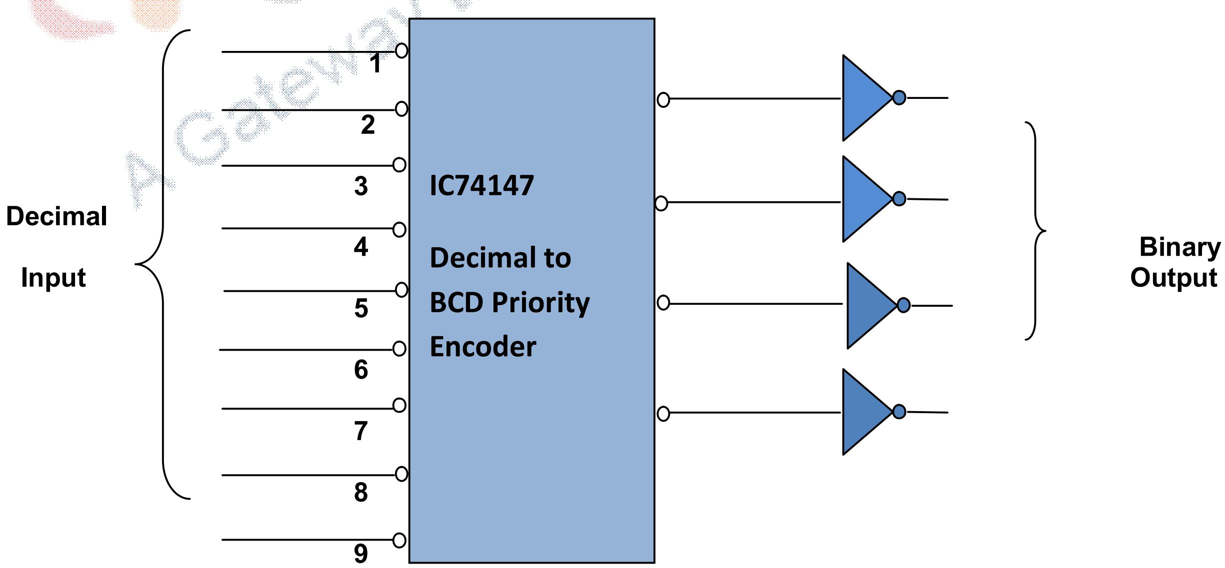

Fig. 10 Logic Symbol of Decimal to BCD Priority encoder IC 74147

Keyboard Encoder

Keyboards are generally used for data entry into the computer and digital systems. The Keyboards are either hexadecimal or ASCII type. For electronic calculators there are 10 switches (Keys) for decimal number 0 to 9 and some function keys. When a digit key is pressed the internal circuit understands which key is pressed and Keyboard encode will give a code corresponding to key pressed. A typical Keyboard encoder using IC 74147 a priority Encoder is shown in below figure 11.

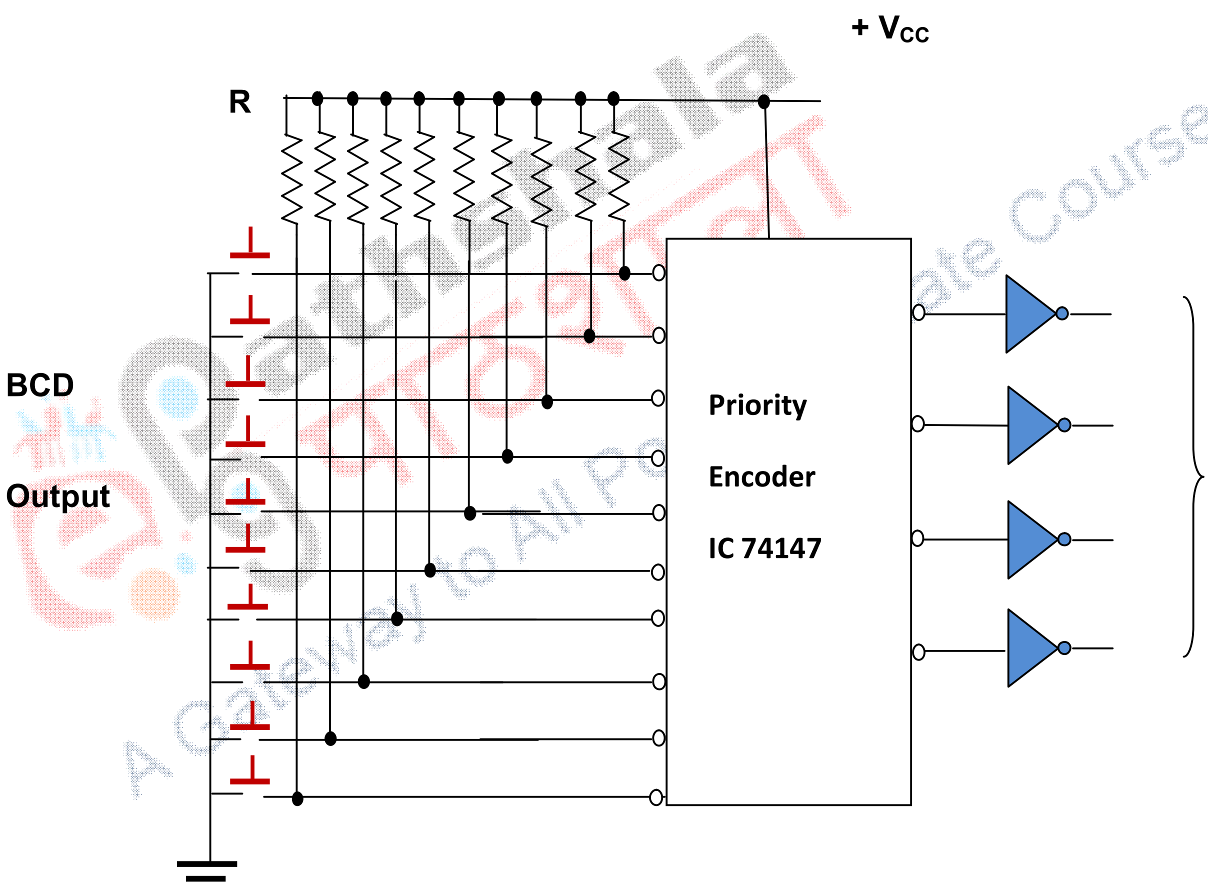

Fig.11 Keyboard Encoder

There are 10 Push button switches with pull-up resistors connected to the positive Supply. The pull up resistor make the input line high when the switch is not closed. When a switch is closed the corresponding line will be connected to ground. This provides low level to the corresponding input to the encoder. When key corresponding to decimal digit say 3 is pressed, the encoder IC generates a BCD code of 3 in complemented form. The generated binary code can be converted to proper BCD by using inverter at the output of the encoder. In this encoder, there are nine inputs with highest priority to number 9 and least to number 1.

4: Applications of Encoder

Encoders are normally preferred where there is a need to convert information of code from one format to another. In addition to linear and priority encoders, encoders have many applications as listed below.

1. Keyboard encoder for computers
2. Optical encoders ï linear or rotary
3. Interfacing peripherals to microprocessors
4. Audio/video coding and transmission

5. Summary

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 coverts more familiar numbers, symbols or character into binary code. 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.

The encoder accepts an m-bit input digital word and converts it into an n-bit another digital word. Commonly used encoders are ï (1) **Linear encoders** are octal to binary, Decimal to BCD and Hexadecimal to binary where normal encoding is implemented and (2) **Priority encoders**.

Encoders can also be used in keyboard and other peripheral interfacing, converting linear or angular position into binary through optical encoders and audio/video coding and transmission.