Unit-IV

Combinational Circuits, Multiplexer-IC 74150 and IC 44151, De multiplexer-IC 74154, Decoder-IC 74139, BCD to Seven segment De-coder IC 7446/7447 IC 7448/7449 Decimal to BCD Priority Encoder-IC 7447, parity Checker-IC 741 80, Magnitude Comparator IC 7485.

Chapter

4 Combinantional Circuits

1. Introduction

Logic circuit for digital systems may be combinational or sequential. A combinational circuit consists of logic gates whose output at any time are determine from the present combination of inputs. A boolean circuit performs an operation that can be specified logically by a set of Boolean function.

A combinational circuit consists of input variables, logic gates and output variables. The logic gates accept signals from the inputs and generate signal to the outputs. This process transforms binary information from the given input date to a required output date. A block diagram of a combinational circuit is shown in Fig. 4.1. The n-input binary variable come from an external source, the m output variable go to an external destination.

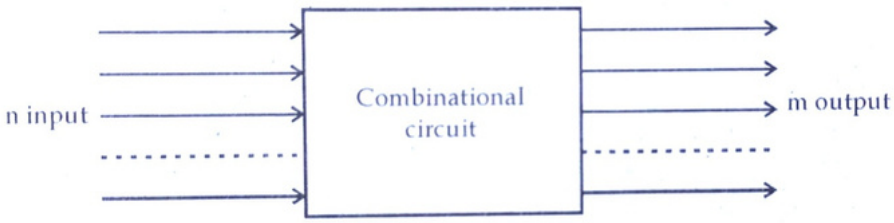


Fig. 4.1 : Block diagram of Combinational Circuit

Each input and output variable exists physically as a binary signal that represent logic 1 and logic 0. In many applications the source and destination are storage registers. It the registers are included with the combinational gates, then the total circuit considered as sequential circuit.

For n-input variables, there are 2^n possible binary input combinations. For each possible input combination, there is one possible output value, Thus, a combinational circuit can be specified with a truth table that lists the output value for each combination of input variables. A combinational circuit also can be described by m boolean functions, one for each output variable. Each output function is expressed in terms of the n input variables.

There are several combinational circuit that are employed extensively in the design of digital systems. These circuits are available in interpreted circuits and are classified as standard components. They perform specific digital functions commonly needed in the design of digital systems. In this chapter, we introduce the most important standard combinational circuits such as multiplexers, Demultiplexers, Decoder

Comparator. These components are available in interpreted as MSI (medium scale intepration) circuits. They are also used as standard cells in complex VLSI circuits. Such as application specific interprated circuits (ASIC). The standard cell functions are interconnected within the VLSI circuit in the same ay as they are used in multiple IC-MSI design.

2. Multiplexers - (Data Selectors)

The term 'multiplex' means "many into one" the multiplexer is combinational circuit that is one of the most widely used standard circuits in digital design. Multiplexing is the process of transmitting a large number of information over a single line. The multiplexers or (data selectors) is a logic circuit that gates one and of several inputs to a single output. The input selected is controlled by a set of select inputs. Fig. 4.2 shows the block diagram of a multiplexer with n input lines and one output line. For selecting one out of n inputs for connection to the output, a set of m select inputs one out of n date sources is selected and transmitted to a single output channel. Normally a strobe (or enable) input (a) is incorporated which helps in cascading and it is generally active law, which means performs its intended operation when it is low.

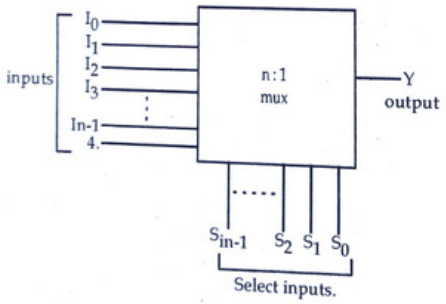


Fig. 4.2 : Block Diagram of a digital multiplexer.

Some standard ICs are available, which are shown in table below, for the 2:1, 4:1, 8:1 and 16:1 multiplexers.

Table 1 : Available multiplexer IC's

IC No.	Description	Output
74157	Quad 2:1 Multiplexer	Same as input
74158	Quad 2:1 Multiplexer	Inverted input
74153	Dual 4:1 Multiplexer	Same as input
74352	Dual 4:1 Multiplexer	Inverted input
74151A	8:1 Multiplexer	Complementary outputs
74152	8:1 Multiplexer	Inverted input
74150	16:1 Multiplexer	Inverted input

Basic four input one output multiplexer.

The table shown below, the truth table of 4:1 multiplexer. The $4(2^2=4)$ represents number of inputs and 1 represents output line. The 2 select lines are required to construct 4:1 multiplexer.

Table 2 : Truth table of 4 : 1 mux

Enable input E	Select lines S_1	S_0	Output 4. (Selected inputs)
0	X	X	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

When the enable input is equal to 0, the multiplexer is disable and when the enable is equal to 1 the multiplexer is enabled and is operated based on the above truth table.

The data output is equal : to I_0 when $S_1 = S_0 = 0$ Y, $= I_0 S_1 S_0$

Data output is equal to I_1 when $S_1 = 0$ and $S_0 = 1$ Y, $= I_1 S_1 S_0$

Data output is equal to I_2 when $S_1 = 1$ and $S_0 = 0$ Y, $= I_2 S_1 S_0$

Data output is equal to I_3 when $S_1 = 1$ and $S_0 = 1$ Y, $= I_3 S_1 S_0$

The Boolean function of 4 to 1 multiplexer is written as follows.

$$Y = I_0 \overline{S_1 S_0} + I_1 \overline{S_1} S_0 + I_2 S_1 \overline{S_0} + I_3 S_1 S_0$$

The hardware implementation of the above equation four 3 input AND gates and a 4 input OR gate and two inverters to generate the complements of S_1 and S_0 Fig. 4.3 (a) shows the logic diagram of 4:1 multiplexer and Fig 4.3 (b) shows the logic Symbol of 4:1 multiplexer.

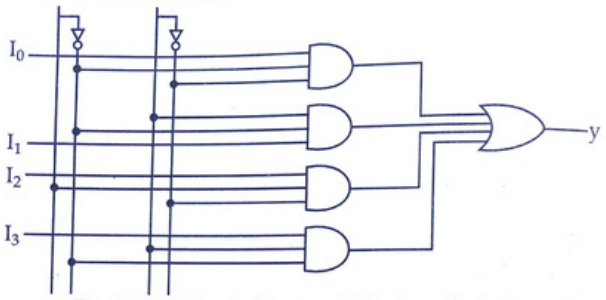


Fig. 4.3 : (a) Logic diagram of 4 to 1 multiplexer

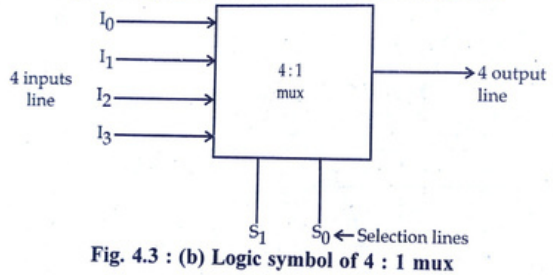


Fig. 4.3 : (b) Logic symbol of 4 : 1 mux

IC 74151 - 8 to 1 multiplexer -

IC 74151 is an 8-to-1 multiplexer with eitht data inputs (D_0 - D_1), three select input lines (S_2 - S_0) and a single output (Y). It also has n enable input $\overline{E}$ and provides both normal and inverted outputs (i.e. Y and $\overline{Y}$). Since $2^3 = 8$, three bits are required to select any one of the eight data bits. When $\overline{E} = 0$, the select inputs $S_2S_1S_0$ will select one of the data input to pass through the output Y. When $\overline{E} = 1$, the multiplexer is disabled. The logic symbol of IC 74151 is shown is shown in fig. 4.4 (a) and its logic diagram is shown in Fig. 4.4 (b). The operation of this IC is summarized in truth Table.

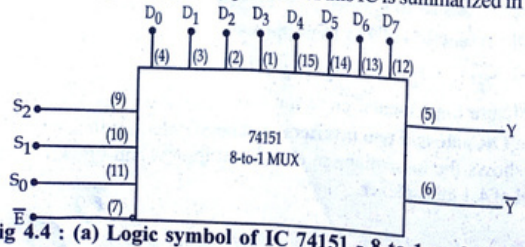


Fig 4.4 : (a) Logic symbol of IC 74151 - 8-to-1 multiplexer

Table-3 : Truth table of IC 74151 - 8-to-1 multiplexer

Inputs				Outputs	
$\overline{E}$	S_2	S_1	S_0	Y	$\overline{Y}$
1	X	X	X	1	0
0	0	0	0	D_0	$\overline{D}_0$
0	0	0	1	D_1	$\overline{D}_1$
0	0	1	0	D_2	$\overline{D}_2$
0	0	1	1	D_3	$\overline{D}_3$
0	1	0	0	D_4	$\overline{D}_4$
0	1	0	1	D_5	$\overline{D}_5$
0	1	1	0	D_6	$\overline{D}_6$
0	1	1	1	D_7	$\overline{D}_7$

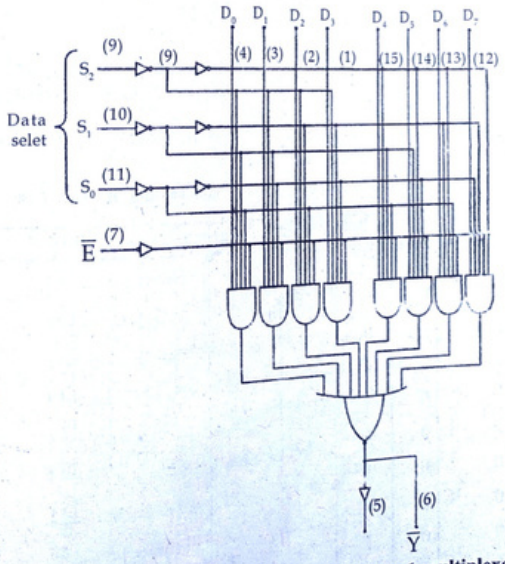


Fig. 4.4 : (b) : Logic diagram of IC 74151 - 8-to-1 multiplexer

IC- 74150 - 16 to 1 multiplexer-

IC 74150 is a 16-to-1 TTL multiplexer. Its pinout diagram is shown in Fig. 4.5 (a). It has 16 inputs (D_6 - D_{15}), a single output (Y) and four select inputs (S_3 - S_0). Pins 1 to 8 and 16 to 23 are the input pins and the pins 11, 13, 14 and 15 are the select inputs $S_3S_2S_1S_0$. Pin 10 is the output and it equals the complement of the selected data input. Pin 9 is for the strobe, an input signal that disables or enables the multiplexer. A low strobe enables the multiplexer, so that output Y equals the complement of the input data bit (i.e. $Y = \overline{D_n}$). The truth table of IC 74150 is shown in Table. 4. The logic diagram of a typical 16-to-1 multiplexer is shown in Fig. 4.6.

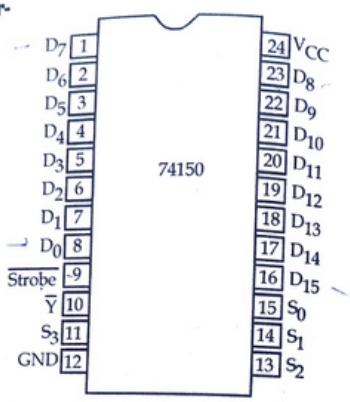


Fig. 4.5 (a) Pinout diagram of IC 74150 - 16-to-1 multiplexer

Table- 4 : Truth table of IC 74150 - 16-to-1 multiplexer

Strobe	Select inputs				Output
$\overline{S}$	S_3	S_2	S_1	S_0	Y
0	0	0	0	0	$\overline{D}_0$
0	0	0	0	1	$\overline{D}_1$
0	0	0	1	0	$\overline{D}_2$
0	0	0	1	1	$\overline{D}_3$
0	0	1	0	0	$\overline{D}_4$
0	0	1	0	1	$\overline{D}_5$
0	0	1	1	0	$\overline{D}_6$
0	0	1	1	1	$\overline{D}_7$
0	1	0	0	0	$\overline{D}_8$
0	1	0	0	1	$\overline{D}_9$
0	1	0	1	0	$\overline{D}_{10}$

0	1	0	1	1	$\overline{D}_{11}$
0	1	1	0	0	$\overline{D}_{12}$
0	1	1	0	1	$\overline{D}_{13}$
0	1	1	1	0	$\overline{D}_{14}$
0	1	1	1	1	$\overline{D}_{15}$
1	X	X	X	X	1

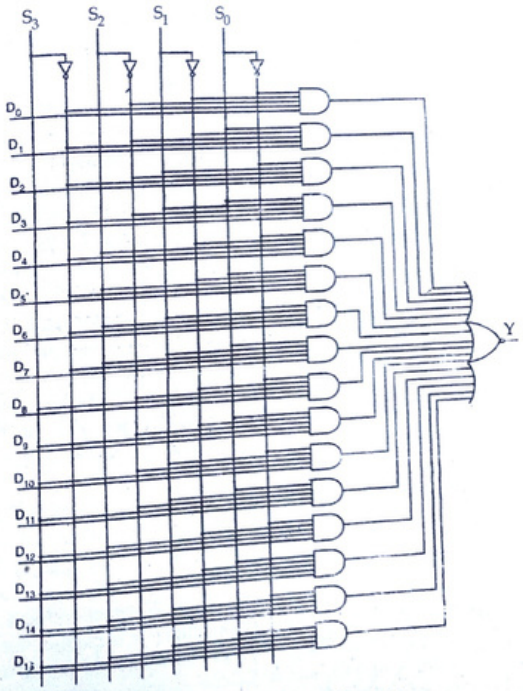


Fig. 4.6 : Logic diagram of 16-to-1 multiplexer

3. Demultiplexer : (Data Distributors)

Demultiplexer accept single input and distributes it over several outputs. Thus demultiplexer performs the reverse operation of a multiplexer. Block diagram of

demultiplexer is given below in Fig. 4.7.

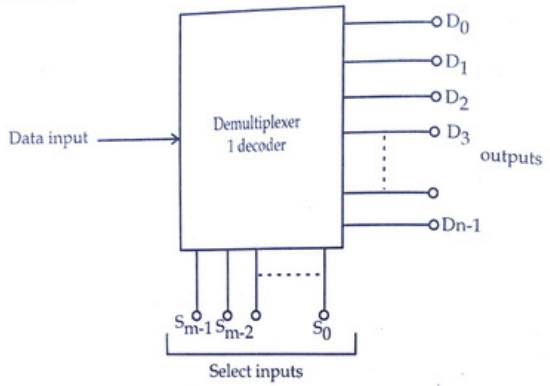


Fig. 4.7 : Block diagram of a 1 : 4 Demultiplexer

As we can see a single input is given select input lines determines to which output line. the input is transmitted let number of output lines are n. And number of select lines is m. Then there is a relation $n = 2^m$. So this circuit can also be used as binary to decimal decoder with binary inputs applied at the select input lines and output is obtained on corresponding output line. Input data lines are connected to logic 1. The circuit can be designed using gates DEMULTIPLEXER is also available as an MSI IC and can be used for the design of combinational circuit. DEMULTIPLEXER is used if multiple output combinational circuit is to be designed, because it requires minimum package count. DEMULTIPLEXER may be available in many forms like-2-line to 4-line, 3-line to 8-line and 4-line to 16-line decoder output of all above demultiplexers are active low. There is a active low enable data input terminal available.

To realize some of the boolean expressions in the standard SOP form, decoder requires some gates. In combinational logic design using multiplexer, additional gates are not required. But if we are designing combinational circuit using demultiplexer, then some additional gates are required. Inspite of this drawback, decoder is more ecomomicalin those cases where nontrivial, multiple output expressions of the same input variables required. In such case 1-multiplexer is required for each output, whereas it is likely that only one decoder will be required supported with few gates. So in some cases using a decoder could have advantages over using a multiplexer. Demultiplexer are available in IC form. Some of the available demultiplexer ICs are given below.

Table : 5 Available demultiplexer ICs

IC No.	Description Output	
74139	Dual 1:4 Demultiplexer (2-line-to-4-line decoder)	Inverted input
74155	Dual 1:4 Demultiplexer (2-line-to-4-line decoder)	1Y—Inverted input 2Y—Same as input
74156	-do-	Open-collector 1Y—Inverted input 2Y—Same as input
74138	1:8 Demultiplexer (3-line-to-8-line decoder)	Inverted input
74154	1:16 Demultiplexer (4-line-to-16-line decoder)	Same as input
74159	-do-	Same as input Open-collector

1-to-4 Demultiplexer

In 1:4 demultiplexer, there is single input line (D). There are 4 output lines (y_0, y_1, y_2, y_3). Two select lines (S_0 and S_1). Truth table of 1 to 4 demultiplexer is shown below in Fig..

Table 6 : Truth table for 1 : 4 demultiplexer.

Data input	Select input		Output			
	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
D						

We can see from the above table that if select lines are $S_1 = 0$ and $S_0 = 0$ then output to is selected similarly when $S_1 = 0$ and $S_0 = 1$ then y_1 output is selected. If $S_1 = 0$ and $S_2 = 1$ then y_3 output line is selected. The output expression from the truth table can be written as-

$$Y_0 = \overline{S_1}\,\overline{S_0}\,D$$
$$Y_1 = \overline{S_1}\,S_0\,D$$
$$Y_2 = S_1\,\overline{S_0}\,D$$
$$Y_3 = S_1 S_0 D$$

So from above expression it is clear that 1:4 demultiplexer can be designed complemented using four 3-input AND gates and two NOT gates. The logic diagram of 1 to 4 demultiplexer is shown in Fig. 4.8.

In logic diagram the input data lines are connected to all AND gates. Two select lines S_1S_0 enables only one gate at a time and the data that appears on the input lines passes through the selected gate to the associated output line.

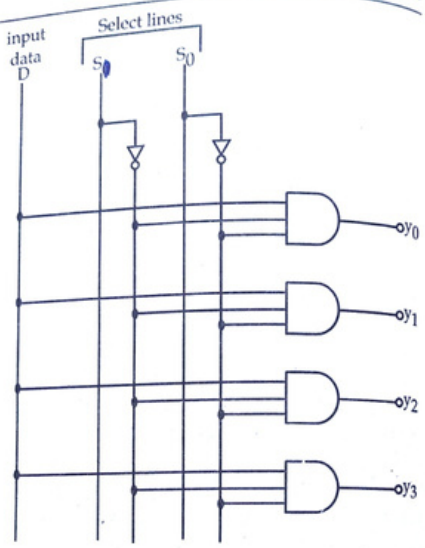


Fig. 4.8 : logic diagram of 1:4 demultiplexer.

1-to-8 demultiplexer

In 1 to 8 demultiplexer there is single input line (D). There are 8 output lines (y_0, y_1, y_3, y_4, y_5, y_6, y_7). There are three select lines (S_0, S_1, S_2). The select line actually select the output lines on which input is to be transmitted.

Truth table of 1 to 8 demultiplexer is shown below.

Table 7 : Truth table of 1 to 8 demultiplexer.

Data input	Select input			Output							
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

From the table it is clear that input data line is connected to one of the output line based on select input. Expressions of 8 output lines can be expressed as follow from the truth table.

$$Y_0 = \overline{S_2}\,\overline{S_1}\,\overline{S_0}\,D$$
$$Y_1 = \overline{S_2}\,\overline{S_1}\,S_0D$$
$$Y_2 = \overline{S_2}\,S_1\,\overline{S_0}\,D$$
$$Y_3 = \overline{S_2}\,S_1S_0D$$
$$Y_4 = S_2\,\overline{S_1}\,\overline{S_0}\,D$$
$$Y_5 = S_2\,\overline{S_1}\,S_0D$$
$$Y_6 = S_2S_1\,\overline{S_0}\,D$$
$$Y_7 = S_2S_1S_0D$$

So 1 to 8 demultiplexer can be designed using eight 4-input AND gates. and 3 NOT gates. So logic diagram of 1 to 8 demultiplexer is shown below in Fig. 4.9.

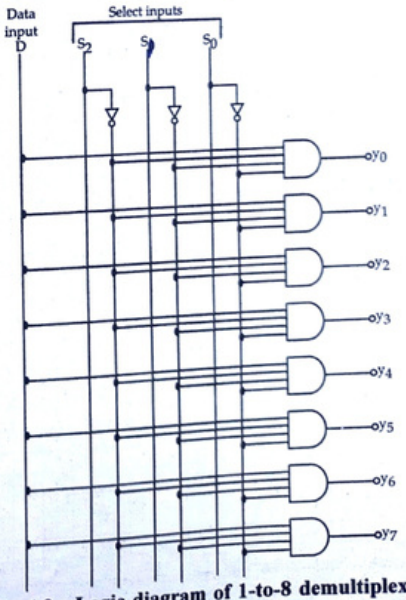


Fig. 4.9 : Logic diagram of 1-to-8 demultiplexer

IC- 74154 - 1 to 16 Demultiplexer

The 74154 is a 1 to 16 demultiplexer with the pin diagram of Fig. 4.10 (a) The pin 18 is for the input Data 0. and pins 20 to 23 are for the controls bits. ABCD Pin 1 to 11 and 13 to 17 are for the output bits $y_0 y_{15}$. Pin 19 is for the strobe. again an active low input finally. Pin 24 is for Vcc and Pin 12 for ground.

Table 8 shows the truth table of 74154. The STROBE input must be low to activate the 74154. When STROBE input is low. then the control input ABCD determines which output lines are low when the DATA input is low. When the DATA input is high. all output lines are high. And when STROBE is high, all output lines are high.

Fig. 4.11 shows low to draw a 74154 on a schematic diagram. There is one input data bit under the control of nibble ABCD. The DATA bit is automatically steered to the output line whose subscript is the decimal equivalent of ABCD. Bubble on the STROBE Pin indicates an active low input. DATA is inverted at the input and again on any output. With this bubble inversion, DATA pass through the 74154 unchanged.

Table-8 : Truth table of IC 74154 1-to-16 DEMUX

Strobe	Data	S_3	S_2	S_1	S_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

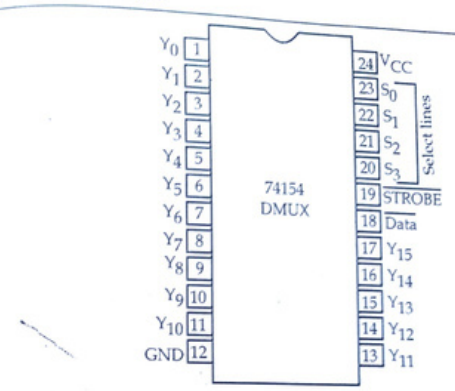


Fig. 4.10 : Pinout diagram of IC 74154—1-to-16 DEMUX

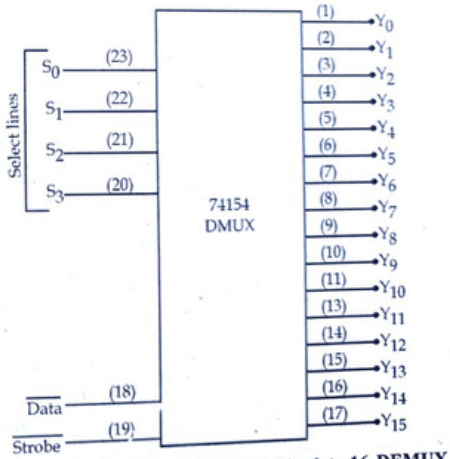


Fig. 4.11 : Logic symbol of IC 74154—1-to-16 DEMUX

4. Decoder

A decoder is similar to demultiplexer but without any date input. The digital systems require the decoding of date. Decoding is necessary in applications such as data Demultiplixing, digital display, digital to analog converters and memory addressing. A decoder is a logic circuit that coverts an n-bit binary input code into 2^n output lines.

Such that each output line will be activated for only one of the possible combinations of inputs.

In a decoder the number of output is greater than the number of inputs. A binary code of n bits is capable of representing up to 2^n distinct elements of coded information. A decoder is a combinational circuit that converts binary information form n input line to a maximum of 2^n unique output lines. If the n-bit coded information has unused combinations, the decoder may have fewer than 2^n outputs.

The decoders presented here are called n to m line decoders, where $m \leq 2^n$. Their purpose is to generate the 2^n minterms of n input variables. The name decoder is also used in conjuction with other code converters such as a BCD-to-seven-segment decoder.

Basic Binary decoder

An AND gate can be used as the basic decoding element because its output is HIGH only when all the inputs are HIGH. For example if the input binary number is 1001, then, to make all the inputs to the AND gate HIGH, the two middle bits (O_s) must be inverted by using two not gates as shown in Fig. 4.12.

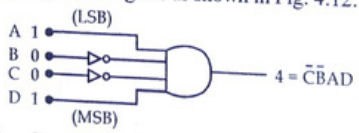


Fig. 4.12 : Decoding logic for input 1001 with an active HIGH input

If a NAND gate is used in place of the And gate, then a Low output is generated to indicate the presence of the proper binary code.

3 to 8 decoder

Three input lines i.e. n = 3 [A, B, C] and m = 2^3 = 8 output lines [y_0, y_1, y_2..... y_7]. The block diagram of 3 to decoder is shown in Fig. 4.13.

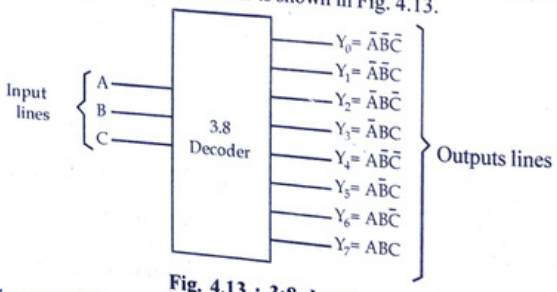


Fig. 4.13 : 3:8 decoder

Each expression on output line corresponds to a single minterm. For example, when input ABC = 000 is applied, the output line y_0 will get enabled. It means the output line y_0 will give logic 1 and output of other lines will be zero. So minterm

$\bar{A}\bar{B}\bar{C}$ [i.e. ABC = 000] us obtained at output line y_0. the truth table of 3 to 8 line decoder is shown in table below.

Table 9 : Truth Table for 3:8 decoder

Input			Outputs							
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

From the truth table the boqlem expression of each output is as follows :

$y_0 = \bar{A}\bar{B}\bar{C}, y_1 = \bar{A}\,\bar{B}C, y_2 = \bar{A}\,B\bar{C}, y_3 = ABC,$

$y_4 = A\,\bar{B}\,\bar{C}, y_5 = A\,\bar{B}C, y_6 = AB\bar{C}, y_7 = ABC$

The logic diagram of these boolem expression can be implemented using AND and Not gates as shown in Fig. 4.14.

Therefore, the n to 2^n line decoder can be called as minterm generator. Any boolean function can be implemented with decoder and logic gates. We know each output line of the decoder represents the partialar minterm of corryponding inputs applied to the selected lines for example consider the pair of boolean functions.

$Y_1 = F_1 [A, B, C] = \sum m[0, 1, 4, 6]$

$Y_2 = F_2 [A, B, C] = \sum m[2, 5, 7]$

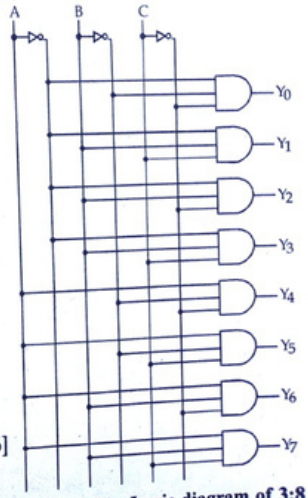


Fig. 4.14 : Logic diagram of 3:8 decoder

The above boolean function F_1 and f_2 can be obtained by 3:8 decoder and two OR gates. The number of input terminals is each OR gate will be the number of minterms in the boolem function. The implemenation of boolem function y_1 and y_2 using is shown in Fig 4.15.

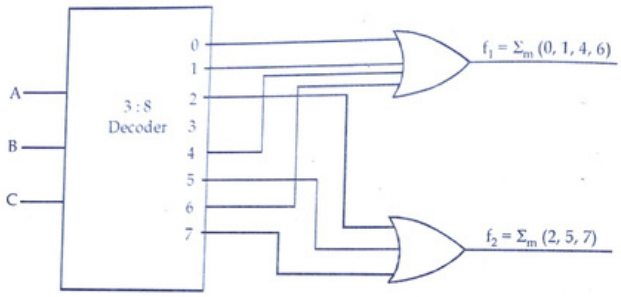


Fig. 4.15 : 3 : 8 Decoder with two OR code

The Fig. 4.15 shows that both the function have same three inputs A, B, C since function f_1 has minterms 0, 1, 4, 6 and function f_2 has 2, 5, 7 minterms the output of f_1 and f_2 will be 1 when input combination A, B, C is applied.

For example when ABC = 010 equivalent to $[2]_{10}$ is applied, then second output line of decoder will select and output will be high or logic 1.

These boolean expressions can be implemented using a 3.8 decoder and NOR gate also. The boolem function can be expressed as

$$\overline{f_1} = [A, B, C] = \overline{\sum_m [0, 1, 4, 6]} = \Sigma_m [2, 3, 5, 7]$$

$$\overline{f_2} = \overline{\sum_m [2, 5, 7]} = \Sigma_m [0, 1, 3, 4, 6]$$

so, $$f_1 [A, B, C,] = \overline{f_1} [A, B, C] = \overline{\sum_m [2, 3, 5, 7]}$$

$$f_2 [A, B, C] = \overline{f_2} [A, B, C] = \overline{\sum_m [0, 1, 3, 4, 6]}$$

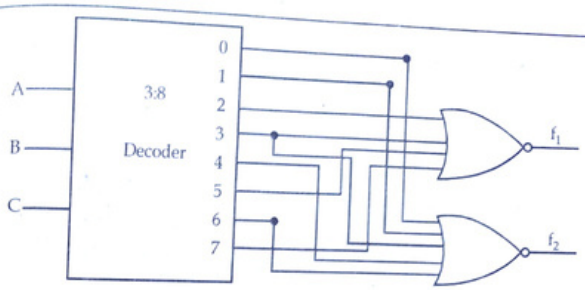


Fig. 4.16 : 3 : 8 Decoder with two NOR gate

The complement of the function will be the sum of minterms which gives the output equal to zero. If complement is taken of the complemented function, it will be equal to the function of itself. So it can be obtined by NOR gate.

If the function is given in the product of sum [POS] from then the function can be implemented using decoder and AND gates. For examples : F

$$f_1 [A, B, C] = \pi_m [2, 3, 5, 7]$$

$$f_2 [A, B, C] = \pi_m [0, 1, 3, 4, 6]$$

The implemenation of the above function is shown in Fig. 4.17.

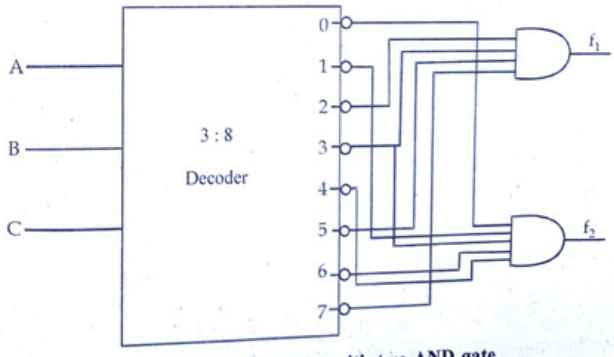


Fig. 4.17 : 3:8 Decoder with two AND gate

IC 74139-Dual 2 to 4 decoder

IC 74139 consists of two individual 2-to-4 decoder demultiplexers in a single pack-age. The logic symbol of IC 74139 is shown in Fig 4.18 . Each decoder has two inputs, four active LOW outputs and one active LOW enable input. This active LOW enable inputs can be used as the data input in demultiplexing applications.

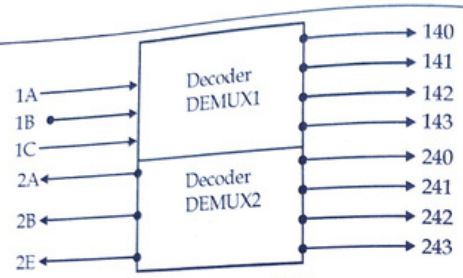


Fig. 4.18 : IC 74139

IC 74154-4 to 16 Decoder

IC 74154 is calleda 4-to-16 decoder demultiplexer because it is capable of decoding as wall as demultiplexing. In order to use this IC 74154 as a decoder, the inputs data and strobe must be grounded as shown in Fig. 4.19

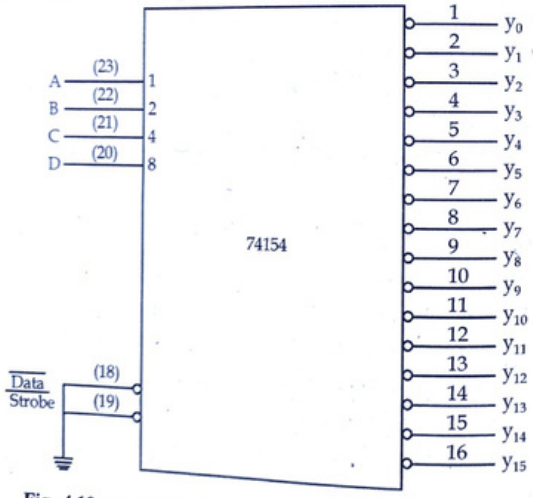
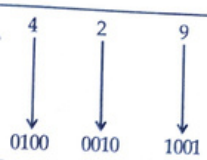


Fig. 4.19 : Logic symbol of IC 74154 used as decoder

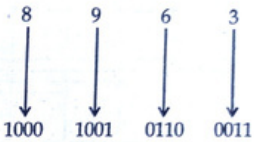
5. BCD to Decimal Decoder

BCD is an abbraviation for binary-coded decimal. The BCD code expresses each digit in a decimal number by its nibble equivalent. For instance, decimal number 429 is changed to its BCD from as follows.



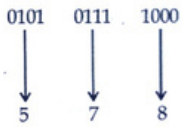
To anyone using the BCD code, 0100 0010 1001 is equivalent to 429.

As another example here is how to convert the decimal number 8963 to its BCD from.



Again , we have changed each decimal digit to its binary equivalent some early computers processed BCD numbers. Which the converted from BCD back to decimal numbers.

Here is an example of how to convert from the BCD from back to the decimal number.



So that 578 is decimal equivalent of 0101, 0111, 1000.

One final point should be considered. Notice that BCD digits are from 0000 to 1001. All combinatiors above this [1010 to 1111] cannot in the BCD code because highest decimal digit bing coded is 9.

BCD to Decimal Decode-IC-7445 :

The TTL IC 7445 is a BCD-to-decimal decoder/driver. The term driver is added to its decription because this IC has open collector outputs that can operate at higher current, and voltage limits than normal TTL output. The output of 7445 can sink upto 80mA in the low state and can be raised up to 30V in the HIGH state. This makes it possible for them to drive loads such as indicator LEDs or lamps, relays or DC motors. This IC is functionally equivalent to 1-of -10 decoder, shown in Fig 4.20 except that the outputs are in the active low state.

For a valid BCD code, the corresponding output will be in the LOW state while all other output liny are in the HIGH state. It is important to note that an invalid BCD

input [1010 to 1111] forces all output liny into the HIGH state. The Pinout diagram IC 7445 is shown in Fig. 4.21

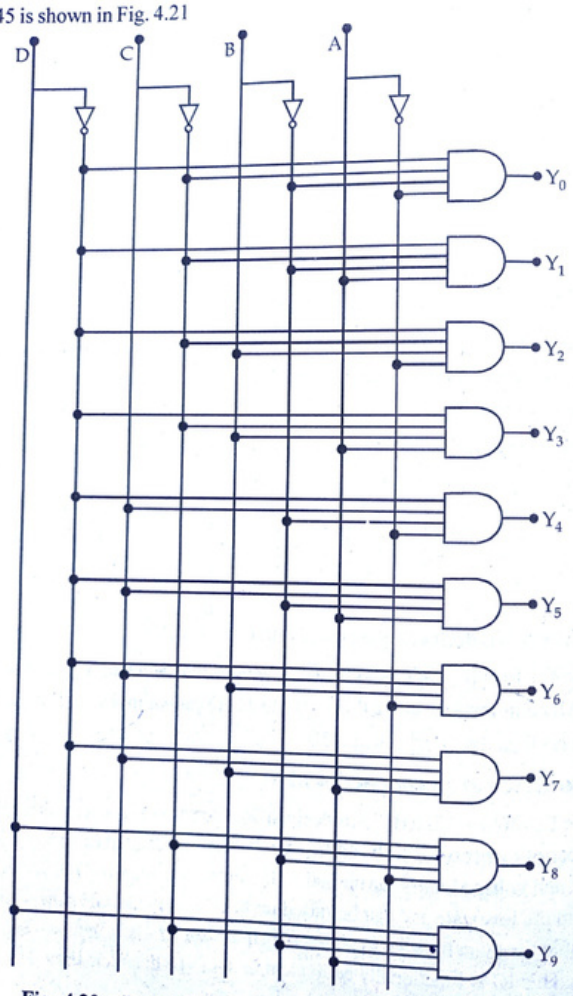


Fig. 4.20 : Logic diagram of BCD-to-decimal decoder

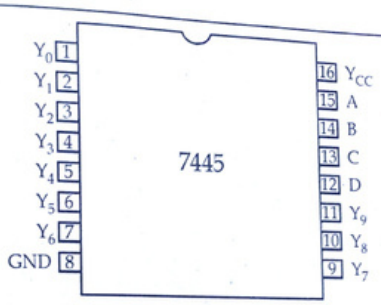


Fig. 4.21 : Pinout diagram of IC 7445-BCD-to-decimal decoder

A list of BD-to-decimal decoder ICs is given decoder driver ICs

Table 10 : BCD-to-decimal decoder driver ICs

IC No.	Output circuits	Applications
7441	Open-collector	Nixie tube driver
7442	Totem-pole	LED driver
7445	Open-collector	Indicator/relay driver
74141	-do-	Nixite tube driver
74115	-do-	Indicator/relay driver
74445	-do-	Indicator/relay driver

6. BCD to seven segment decoder :

A seven segment display is used to display the decimal digits from 0 to 9. The BCD to seven segment decoder accepts BCD input and displays the corrosponding decimal digits. The block diagram of BCD to 7-segment decoder is shown in Fig. 4.22 (a). The seven segment display is composed of seven elements or segment. Each segment is small light emitting diode (LED), which glows when electrical current passes through it. As shown in Fig.4.22 (b) the segments are labelled from a to g.

The letters a, b, c, d, e, f and g run clockwise from the top of each segment from which we candisply the digits 0 through 9 as shown in Fig. 4.22 (c) for example, to display 9, we need light up segment a, b, c, d, f and g.

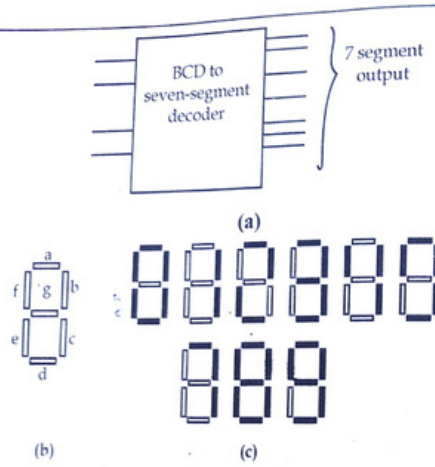


Fig. 4.22 : (a) Block diagram BCD to segment diagram, (b)Seven-segment display, (c) Display of decimal digits in a seven segment display

Design of BCD of seen segment decoder

The truth table for BCD to seven segment decoder is given in table below.

Table - 11 : Truth table of BCD-to-7-segment decoder

BCD inputs				Seven segment outputs						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

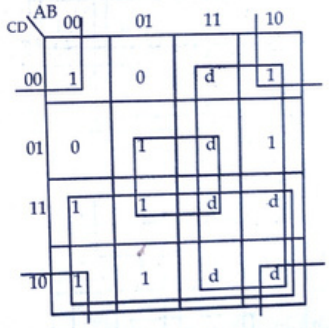
According to this table the decoder accepts the 4-bit BCD inputs A, B, C, D and he seven outputs are set to display the appropriate BCD digit for example, if BCD

input ABCD = 0111 is applied to decoder the output will display decimal seven. It means the segment a, b, and c will glow and other segments will be off. The output a, b, c, will be on logic and other will be on logic. Similarly, all the conditions can be verified from the truth table . Since the input is in BCD from, so the 4-bit inputs from 0000 to 1001 (0 to 9) will be valid and other six combinations (1010 to 1111) will be invalid for the invalid combinations, the output will be considered as don't care (d) condition.

To implement the logic diagram of BCD to seven segment decoder the boolean expression of the outputs in terms of inputs are obtained by solving K-map as follows.

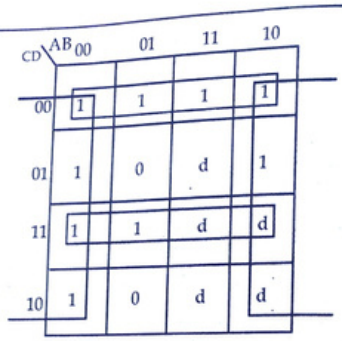
$$a = \Sigma_m (0, 3, 4, 5, 6, 7, 8, 9) + \Sigma_d (10, 11, 12, 13, 14, 15)$$
$$b = \Sigma_m (0, 1, 2, 3, 4, 7, 8, 9) + \Sigma_d (10, 11, 12, 13, 14, 15)$$
$$c = \Sigma_m (0, 1, 3, 4, 5, 6, 7, 8, 9) + \Sigma_d (10, 11, 12, 13, 14, 15)$$
$$d = \Sigma_m (0, 2, 3, 5, 6, 8, 9) + \Sigma_d (10, 11, 12, 13, 14, 15)$$
$$e = \Sigma_m (0, 2, 6, 8) + \Sigma_d (10, 11, 12, 13, 14, 15)$$
$$f = \Sigma_m (0, 4, 5, 6, 8, 9) + \Sigma_d (10, 11, 12, 13, 14, 15)$$
$$g = \Sigma_m (2, 3, 4, 5, 6, 8, 9) + \Sigma_d (10, 11, 12, 13, 14, 15)$$

The above expressions can be simplified using K-map method as shown Fig. 4.23.



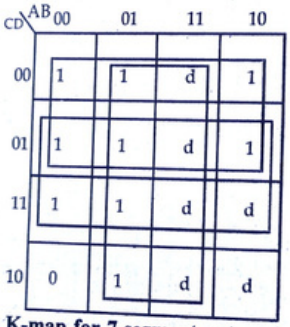
(a) K-map for 7-segment output 'a'

From the above K-map, $a = A + C + \underline{BD} + \bar{B}\,\bar{D} = A + C + \overline{B \oplus D}$



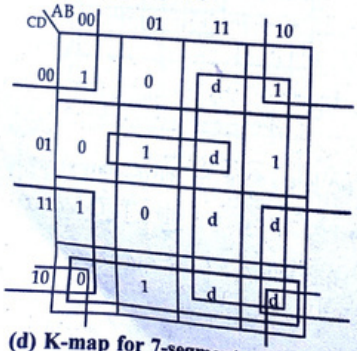
(b) K-map for 7 segment output 'b'

From the above, $b = \bar{B} + CD + \bar{C}\,\bar{D} = \bar{B} + \overline{C \oplus D}$



(c) K-map for 7-segment output 'c'

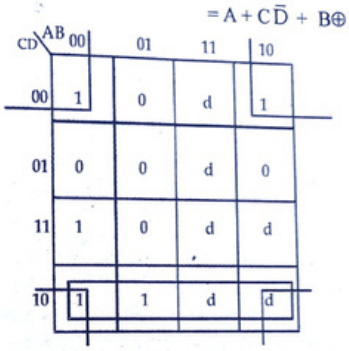
From the above K-map, $c = B + \bar{C} + D$



(d) K-map for 7-segment output 'd'

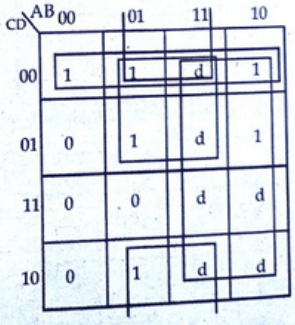
From the above K-map,

$$d = A + \bar{B}\bar{D} + C\bar{D} + \bar{B}C + B\bar{C}D = A + C\bar{D} + \bar{B}(C + \bar{D}) + B\bar{C}D$$
$$= A + C\bar{D} + B \oplus (C + \bar{D})$$



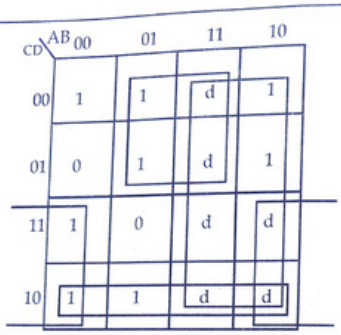
(e) K-map for 7-segment output 'e'

From the above K-map, $e = \bar{B}\bar{D} + C\bar{D} = \bar{D}(\bar{B} + C)$



(f) K-map for 7-segment output 'f'

From the above K-map, $f = A + \bar{C}\,\bar{D} + B\bar{C} + B\bar{D} = A + \bar{C}\,\bar{D} + B(\bar{C} + \bar{D})$



(g) K-map for 7-segment output 'g'

From the above K-map, $g = A + C\bar{D} + B\bar{C} + \bar{B}C = A + C\bar{D} + (B \oplus C)$

Fig. 4.23 K-map simplification for BCD-to-7 segment decoder

Now, using the above simplification expressions for seven-segment outputs, we implement the following logic gates shown in Fig. 4.24.

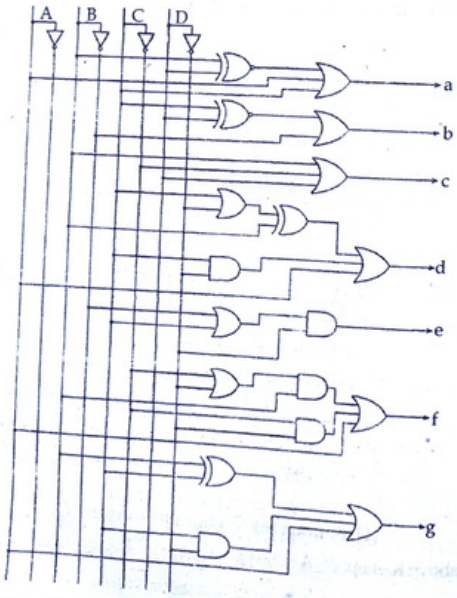


Fig. 4.24 : Logic diagram of BCD-to-7-segment decoder

IC-7446/7447, BCD to seven segment Decoder

A seven-segment decoder-driver is an IC decoder that can be used to drive a seven-segment indicator. There are two types of decoder-drivers, corresponding to the common-anode and common-cathode indicators. Each decoder-driver has 4 input pins [the BCD input] and 7 output pins [the a through g segments].

Fig. 4.25 shows a 7446 driving a common anode indicator. Logic writs inside the 7446 convert the BCD input to the required output. For instance, if the BCD input is 0111, the internal logic [not shown] of the 7446 will force [LED a, b, and c to conduct. As a result, digit 7 will applar on the seven-segment indicator.

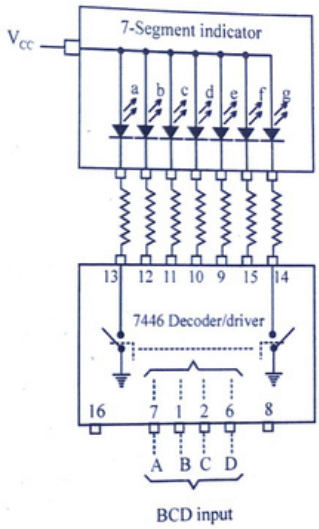


Figure 4.25 : 7446 decoder driver

Notice the current-limiting resistors between the seven-segment indicator and the 7446 of Fig. 4.25 you have to connect these external resistors to limit the current in each sigment to a safe value between 1 and 50mA, depending on how bright you want the deeply to be.

IC-7448/7449-BCD to seven segment Decoder :

Fig. (b) is the alternative decoding approach. Here a 7448 drives a common-cathode indicator. Again logic converts the BCD input to the required output. For example, when a BCD input of 0100 is used, the internal logic forces LEDs b, c, F, and g to conduct. The seven segment indicator then deeplays a unlike the 7446 that requires external current-limiting resistors, the 7448 has its own current-limiting resistors on the chip. A switch 7448 in Fig (4.26) 8 switching in the actual IC is of courage accomplished using bipolar junction transistors.

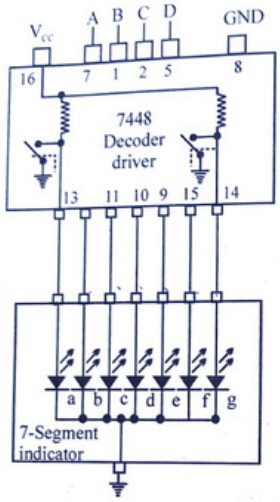


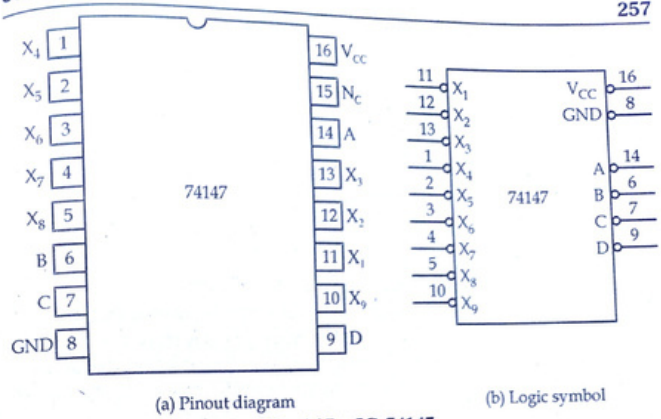
Fig.. 4.26 : 7448 decoder driver

7. IC-74147 Decimal to BCD priority encoder :

The decimal to BCD priority encoder, pinout diagram IC 74147 shown in fig 4.27.

When all the inputs ($X_1 - X_9$) are HIGH, all the outputs are HIGH [i.e. 1111] which is the inverse of 0000, the BCD code for 0. When X_g is low, the ABCD output is 0110, which is the inverse of 1001, the BCD code for 'g'. When X_8 is low, the ABCD output is 0111, the inverse of 1000, the BCD code for 8. Hence the outputs of 74147 will normally be HIGH when none of the inputs is activated. This corresponds to the decimal 0 input condition. Since there is no X_0 input the encoder assumes the decimal 0 input state when all the inputs are HIGH.

The 74147 is called a priority encoder because it gives priority to the highest-order input. For example at a particular instant, if both the inputs X_3 and X_5 are activated, then the highest priority of these two inputs [i.e. X_5] is encoded as 1010 which is the inverse of 0101.



(a) Pinout diagram (b) Logic symbol

Fig. 4.27 : IC 74147

8. Parity generators and checkers :

Even parity means an-n-bit input has an even number of 1s. For instance, 110011 has even parity because it contains four 1s. Odd parity means an-n-bit input has an odd number of 1s. For example, 110001 has odd parity because it contains 1s.

Here are two more examples :

1111 0000 1111 0011 even parity
1111 0000 1111 0111 odd parity

The first binary number has even parity because it contains ten 1s : the second binary number has odd parity because it contains eleven 1s. In identally, longer binary numbers are much easier to read if they are split into nibbly, or groups of four, as done here.

9. Parity checker

Exclusive-OR gates are ideal for checking the parity of a binary number because they produce on 1 when the input has an odd number of 1s. Therefore, on even-parity input to an exclusive-OR gate produces a low output, while an odd parity input producy a high output.

For instance, Fig. 7.28 shows a 16-input exclusive-OR gate. A 16 big number drives the input. The exclusive-OR gate produces on output 1 because the input has odd parity [an odd number of 1s] If the 16-bit input changes to another value, the output become 0 for even-parity numbers and 1 for odd-parity numbers.

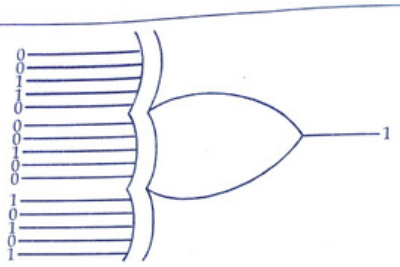


Fig. 4.28 : : EX-OR gate with 16 input

In a computer, a binary number may represent an instraction that tells the computer to add subtract, and so on ; the binary number may represent data to be produssed like 0 number letter etc. In either case, you sometimes will see an extra bit added to the original binary number to produce a new binary number with even or odd parity.

For instance, Fig. 4.28 shows this 8-bit binary number $X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0$.

Suppose they number equals 0100 0001. Then, the number has even parity, which means the exclusive-OR gate produces an output of 0 because of the inverter.

$$X_8 = 1$$

and the final 9-bit output is 1 0100 0001. Notice that this has odd parity.

Suppose we change the 8-bit input to 0110 0001. Now, it has odd parity. In this case the exclusive-OR gate produces an output 1. But the inverter produces a 0, So that the final 9-bit output is 0 0110 0001. Again the final output has odd parity.

The circuit given in Fig. 7.28 is called an odd-parity generator because it always produces a-9-bit output number with odd party. If the 8-bit input has even parity, a 1 comes out of the inverter to produces a final output with odd parity. On the other hond, if the 8-bit input has odd parity a 0 comes out of the inverter, and the final 9-bit output again has odd parity.

10. The 74180-parity generator-checker :

Fig. 4.29 shows the pinout diagram for a 74180, which is a TTL parity generator-checker. The input data bits are X_7 to X_0 ; these bits have a even or odd parity. The even input [Rin 3] and the odd input '[Rin 4] control the operation of the chip os shown in Table. The symbol Σ stands for summation. In the left input coloum of Table. Σ of H's [high]refers to the parity of the input data X_1 to X_0 . Depending on how you set up the value of the even and odd inputs, the Σ even and Σ odd outputs may be low or high.

For instance, suppose even input is high and odd input is low. When the input date has been even parity [the first entry of table], the Σ even output is high and the

Σ odd output is low. When the input date has odd parity. The Σ even output is low and the Σ odd output is high.

If you change the control inputs, you change the operation. Assume that the even input is low and the odd input is high. When the input data has even parity, the Σ even output is low and the Σ odd output is high. When the input data has odd parity, the Σ even output is high and the Σ odd output is low.

The can be used to detect even or odd parily. It can also be set up to generate even or odd parily.

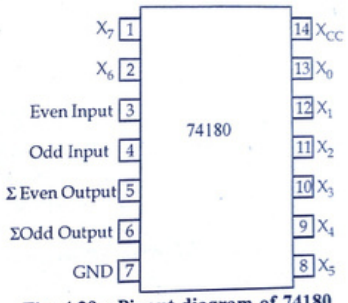


Fig. 4.29 : Pinout diagram of 74180

Table - 74180 Truth Table

Input			Output	
Σ of H's at X_7 to X_0	Even	Odd	Σ Even	Σ Odd
Even	H	L	H	L
Odd	H	L	L	H
Even	L	H	L	H
Odd	L	H	H	L
X	H	H	L	L
X	L	L	H	H

Questions

Very Short Answer Type Questions

1. Define combinational circuits. **[Raj. 2008]**
2. What is a multiplexer ?
3. What is the function of a multiplexer select inputs ?
4. What are the applications of multiplexers.

5. What is a demultiplexer ? Write its uses. **[Raj. 2009]**
6. What is the different multiplexer and demultiplexer ?
7. What is meant by a decoder ? **[Raj. 2008]**
8. Draw the block diagram of decoder.
9. What is seven segment display ?
10. What is encoder ?
11. How does an encoder differ from a decoder ?
12. What is the function of a decoder's enable input (s) ?
13. Can more than one decoder output be activated at one time ?
14. What is priority encoder ?
15. Define parity.
16. How does a priority encoder differ from an ordinary encoder ?
17. What is parity checker ?
18. What is meant by a magnitude comparator ?

Long Answer Type Questions

1. (a) Explain the construction and working of a 1 : 16 demultiplexer make a truth table and draw a logic diagram for a 74154 IC demultiplexer.
   (b) Illustrate the concept of chip expression to get a 1 : 32 decoder using two 74154 IC's **[Raj. 2007]**
2. (a) Explain the working of IC-7485, magnitude comparater.
   (b) Draw the truth table for IC-74180 parity generator and show how to connect a 74180 to generate a 9-bit output with odd parity. **[Raj. 2007].**
3. (a) Explain the construction and working of a 16 : 1 multiplexer.
   (b) Draw the truth table for IC74180 parity generator.
4. Draw pinout and logic diagram of 74147 encoder. **[Raj. 2010]**
5. (a) Explain the workingof Schmitt-trigger.
   (b) Explain the working of IC 7485, magnitude comparater.
6. Describe the decimal to BCD encoder. **[Raj. 2005]** **[Raj. 2005]**
7. Sketch the segments of a seven-segment indicator and explain the function of a seven-segment decoder drive. **[Raj. 2005]**
8. Design a excess-3 to BCD code converter using 4-bit adder.
9. A combination circuit is defined by the following functions.
   $F_1 = x'y' + xyz'$ $F_2 = x' + y$
   $F_3 = xy + x'y'$
   Design the circuit with a decoder and external gates.

10. Write a short note on 'Combinational Circuit Binary Multiplier'.
11. An 8 : 1 MUX has inputs A, B and C connected to selection inputs S_2, S_1 and S_0 respectively. The data input I_0 through I_7 are follows. Determine the Boolean Expression that MUX implements.
    $D_1 = D_2 = D_7 = 0$
    $D_3 = D_5 = 1$
    $D_4 = D_0 = D$
    $D_6 = D$
12. What is the function of a multiplexer's select inputs ?
13. Design a 32-to-1 multiplexer using 8-to-1 multiplexer IC's.
14. Explain BCD to seven segment decoder ?
15. Define a code converter logic circuit.
16. What is meant by a magnitude comparator.
17. Show how a 16-input MUX such as IC 74150 is used to generate the function. $Y = \overline{A}\,\overline{B}\,\overline{C}\,D + BCD + A\overline{B}\,\overline{C} + AB\overline{C}\,D$.
18. How many IC 74154 4 to 16 decoders are necessary to decode a six-bit binary number.
19. Design a 4-bit ADDER/SUBTRACTOR circuit with ADD/SUB control line.
20. Design a BCD-to-Gray code. Converter using
    (a) 8 : 1 multiplexers
    (b) dual 4 : 1 multiplexers and some gates
    (c) NAND gate only.
    (d) quad 2 : 1 multiplexers and some gates.

■■■